

**Witold Jaworski**

# ***Wirtualne modelarstwo***

**Tworzenie realistycznych,  
cyfrowych modeli samolotów**

**Wyciąg IV: Detale i renderowanie**



**wersja 2.0**

# ***Wirtualne modelarstwo***

Mojej żonie

Ta książka powstała dzięki jej wyrozumiałości



**Witold Jaworski**



# ***Wirtualne modelarstwo***

**Tworzenie realistycznych,  
cyfrowych modeli samolotów**

**wersja 2.0**

Copyright Witold Jaworski, 2009 - 2013.

wjaworski@samoloty3d.pl

<http://www.samoloty3d.pl>



This work is licensed under a [Creative Commons Attribution-NonCommercial-NoDerivs 2.0 License](http://creativecommons.org/licenses/by-nc-nd/2.0/).

W szczególności, publikowanie przekładu tej książki na inne języki wymaga pisemnej zgody Autora.

Niniejsza książka może być kopiowana i rozpowszechniana na następujących warunkach:

- Każda kopia powinna zawierać informację o Autorze (copyright);
- Nie wolno używać tej książki do celów komercyjnych;
- Nie wolno zmieniać tekstu ani przekształcać w jakikolwiek sposób układu tej publikacji;

Pełen tekst zasad udostępnienia tej publikacji (licencja) znajduje się na str. 509.

ISBN: 978-83-931754-9-9

Ilustracja na poprzedniej stronie:

Para P-40B z 47 Pursuit Squadron USAAC, której udało się wystartować 7 grudnia 1941r do walki z japońskim nalotem na Pearl Harbor. Pilotują je porucznicy: Kenneth Taylor (samolot z nr „316”) i George Welch (samolot z nr „160”).

47 PS był w tym czasie oddelegowany na ćwiczenia strzeleckie w ustronny, północny zakątek wyspy Oahu, na polowe lądowisko Haleiwa. To obszar nie był blokowany przez japońskie samoloty. (Pearl Harbor leżało na przeciwnym brzegu wyspy). Pas startowy biegł wzdłuż brzegu morza. Tak ta para samolotów mogła wyglądać zaraz po starcie. Fotografia tła, uzyskana z [www.hawaii.gov](http://www.hawaii.gov), przedstawia autentyczny fragment brzegu w pobliżu lotniska Haleiwa.

## SPIS TREŚCI

Po co ta książka została napisana? .....	8
Co zawiera ta publikacja? .....	10
Konwencje zapisu .....	11
Jak czytać tę książkę? .....	12
Budowa modelu .....	15
Rozdział 1. Przygotowanie "warsztatu pracy" .....	16
1.1 Instalacja Blendera .....	17
1.2 Instalacja GIMP .....	20
1.3 Instalacja Inkscape .....	21
1.4 Przetłumaczenie Inkscape i GIMP na język angielski .....	22
Rozdział 2. Pozostałe detale samolotu .....	24
2.1 Korekta kształtu kadłuba .....	25
2.2 Kółko ogonowe .....	38
2.3 Mechanizacja kółka ogonowego .....	51
2.4 Podwozie główne .....	62
2.5 Podwozie — wykończenie .....	75
2.6 Wnętrze kabiny .....	83
2.7 Figura pilota .....	102
2.8 Pozostałe szczegóły zewnętrzne .....	108
2.9 Podsumowanie .....	125
Rozdział 3. Kompozycja sceny .....	128
3.1 Realistyczna scena w locie .....	129
3.2 Realistyczna scena na ziemi .....	143
Szczegóły obsługi programów .....	156
Rozdział 4. GIMP — szczegóły obsługi .....	157
4.1 Zaznaczanie — wg wskazanego koloru .....	158
4.2 Zmniejszanie/zwiększanie obszaru zaznaczenia .....	159
4.3 Zaznaczenie — poprawianie za pomocą <i>Quick Mask</i> .....	160
4.4 Zaznaczenie — zachowywanie obszaru w zakładce <i>Channels</i> .....	162
4.5 Rozmycie metodą Gaussa .....	163
4.6 Nanoszenie przetarć i odprysków farby .....	164
Rozdział 5. Inkscape — szczegóły obsługi .....	173
5.1 Dokładna transformacja obiektu ( <i>Transform</i> ) .....	174
5.2 Wektoryzacja bitmap .....	175
Rozdział 6. Blender — ogólne .....	179
6.1 Ustawienie środowiska pracy .....	180
6.2 Pliki konfiguracji Blendera .....	187
6.3 Kopia bezpieczeństwa i odtwarzanie .....	189
6.4 Ustalenie tła widoku (planów samolotu) .....	191
6.5 Układ ekranu ( <i>Screen layout</i> ) — zarządzanie .....	199
6.6 Instalacja i usuwanie dodatków ( <i>add-ons</i> ) .....	201
6.7 Sceny — zarządzanie .....	203
6.8 Kontrolka wyboru barwy .....	204
6.9 Edytor węzłów ( <i>Node Editor</i> ) .....	207
6.10 Przykład wykorzystania drugiej sceny .....	213
Rozdział 7. Blender — edytor obiektów (Object Mode) .....	217
7.1 Wstawienie pustego obiektu ( <i>Empty</i> ) .....	218

7.2	Powielanie obiektu ( <i>Duplicate</i> ) .....	219
7.3	Wyznaczenie krawędzi przecięcia dwóch powłok .....	221
7.4	Zmiana położenia środka obiektu ( <i>Set Origin</i> ) .....	224
7.5	Chwilowe ukrycie obiektu ( <i>Hide Selected</i> ) .....	225
7.6	Wyrównanie widoku do orientacji obiektu ( <i>Align View to Selected</i> ) .....	226
7.7	Przypisanie materiału do obiektu .....	227
7.8	Opcje wyświetlania obiektu .....	228
7.9	Dodanie armatury ( <i>Armature</i> ) .....	229
7.10	Przypisanie ograniczenia <i>Track To</i> .....	235
7.11	Przypisanie ograniczenia <i>Locked Track</i> .....	237
7.12	Przypisanie ograniczenia <i>Limit Location</i> .....	241
7.13	Przypisanie ograniczenia <i>Transformation</i> .....	243
7.14	Przypisanie ograniczenia <i>Stretch To</i> .....	249
7.15	Przypisanie ograniczenia <i>Limit Distance</i> .....	254
7.16	Odbicie lustrzane ( <i>Mirror</i> ) .....	257
7.17	Wstawienie krzywej ( <i>Curve</i> ) .....	260
7.18	Panel sterowania ruchem obiektów ( <i>HandlePanel</i> ) .....	265
7.19	Dokładne umieszczanie obiektów w zadanym miejscu .....	269
Rozdział 8.	Blender — edytor siatki ( <i>Edit Mode</i> ) .....	272
8.1	Przypisanie materiału do siatki .....	273
8.2	„Zwykle” rozwijanie siatki ( <i>Unwrap</i> ) .....	276
8.3	Rozwijanie poprzez rzutowanie siatki ( <i>Project from View</i> ) .....	277
8.4	Definiowanie grupy wierzchołków ( <i>Vertex Group</i> ) .....	278
8.5	Zaznaczenie krawędzi jako szwu ( <i>Seam</i> ) .....	280
8.6	Dodanie alternatywnego rozwinięcia UV ( <i>UV Map</i> ) .....	282
8.7	Operacje na bryłach ( <i>Boolean</i> ) .....	284
8.8	Wyginanie wzdłuż krzywej ( <i>Curve</i> ) .....	288
8.9	Malowanie wag ( <i>Weight paint</i> ) .....	291
8.10	Fazowanie i zaokrąglanie krawędzi ( <i>Bevel</i> ) .....	294
8.11	Powielenie siatki ( <i>Array</i> ) .....	297
8.12	Deformacja według tekstury ( <i>Displace</i> ) .....	299
8.13	Rzutowanie na powierzchnię ( <i>Shrinkwrap</i> ) .....	302
8.14	Sterowanie kierunkiem normalnych wzdłuż krawędzi siatki .....	304
8.15	Deformacja za pomocą pomocniczej siatki ( <i>Mesh Deform</i> ) .....	309
8.16	Deformacja wielu obiektów za pomocą siatki (dodatek <i>MDeform</i> ) .....	312
Rozdział 9.	Blender — edytor materiałów ( <i>Cycles</i> ) .....	316
9.1	Grupowanie węzłów .....	317
9.2	Wykorzystanie grupy węzłów .....	322
9.3	Wykorzystywanie informacji o typie śledzonego promienia ( <i>Light Path</i> ) .....	324
9.4	Użycie prostego obrazu nieba ( <i>Sky Texture</i> ) .....	327
9.5	Sonda obrazu otoczenia .....	330
9.6	Transformacja współrzędnych ( <i>Mapping</i> ) .....	332
9.7	Dodanie obrazu tekstury ( <i>Image Texture</i> ) .....	334
9.8	Dodanie panoramy otoczenia ( <i>Environment Texture</i> ) .....	336
9.9	Użycie tekstury gradientu ( <i>Gradient Texture</i> ) .....	340
9.10	Proceduralne tekstury „szumu” ( <i>Noise Texture</i> , <i>Voronoi Texture</i> , <i>Musgrave Texture</i> ) .....	343
9.11	Podstawienie obrazu na tło renderingu .....	347
9.12	Komponowanie otoczenia sceny .....	350
9.13	Skonfigurowanie materiału: szkło ( <i>Plexiglas</i> ) .....	360

9.14	Skonfigurowanie materiału: dural i inne powierzchnie ( <i>Gloss Paint</i> ) .....	368
9.15	Węzły pomocnicze .....	376
9.16	Wykorzystanie alternatywnych rozwinięć UV ( <i>Attribute</i> ) .....	380
9.17	Wykorzystanie współrzędnych tekstury ( <i>Texture Coordinate</i> ) .....	383
9.18	Wykorzystanie spektrum barw ( <i>Color Ramp</i> ) .....	386
9.19	Proceduralna tekstura pasków ( <i>Wave Texture</i> ) .....	388
9.20	Stworzenie tekstury „ogólnych zabrudzeń” .....	390
9.21	Skonfigurowanie materiału: stal .....	396
9.22	Skonfigurowanie materiału: guma .....	402
9.23	Skonfigurowanie materiału: skóra .....	408
9.24	Skonfigurowanie materiału: tkanina (pasów) .....	415
9.25	Skonfigurowanie materiału: plastik .....	418
Rozdział 10. Blender — pozostałe .....		420
10.1	Linie pomocnicze ( <i>Grease Pencil</i> ) .....	421
10.2	Animacja ruchu śmigła .....	425
10.3	Rozmycie ruchu śmigła .....	429
10.4	Kompozycja rezultatu ( <i>Composite Nodes</i> ) .....	432
10.5	Zmiana odcienia obrazu ( <i>Color Balance</i> ) .....	438
10.6	Tworzenie masek ze współrzędnych Z pikseli .....	443
10.7	Rozmycie w jednym kierunku ( <i>Directional Blur</i> ) .....	446
10.8	Trawa ( <i>Particles</i> ) .....	452
Dodatki .....		460
Rozdział 11. Dodatkowe wyjaśnienia .....		461
11.1	Struktura danych modelu i sceny w Blenderze .....	462
11.2	Arytmetyka barw .....	468
Skorowidz .....		473
Słownik .....		507
Bibliografia .....		508
Szczegółowe zasady udostępnienia tej publikacji .....		509



## Po co ta książka została napisana?

Bo zawsze chciałem stworzyć model samolotu w komputerze. A gdy już to zrobiłem — stwierdziłem, że to dobra zabawa, i że warto tym doświadczeniem podzielić się z innymi.

Dawno temu byłem zwykłym modelarzem i robiłem redukcyjne modele samolotów. To hobby dość specyficzne. Ślęczyśmy miesiącami nad naszym dziełem, starając się nanieść na kawałek materiału maleńkie nity, złącza blach, czy drobne detale tablic przyrządów. Sądzę, że postronnym obserwatorom zajęcia wędkarzy wydadzą w porównaniu z modelarzami mniej męczące, choć podobnie monotonne. W dodatku, zamiast tworzyć ładne, błyszczące miniaturki, nanosimy na ich powierzchnię (o zgrozo!) zabrudzenia i przetarcia, jakie powstawały podczas intensywnego używania. Szpecimy je, aby wyglądały jak rzeczywiste, zużyte i gdzieś tam pordzewiałe maszyny. Całą naszą nagrodą jest pokazanie naszych dzieł komuś, kto potrafi docenić ich finezję. Zdarza się to raz na jakiś czas. Mówiąc szczerze, dziwnym trafem są to zazwyczaj inni modelarze. A może ukrytą nagrodą jest możliwość nawiązania dyskusji na tak ezoteryczne tematy, jak wyższość Spitfire'a IX nad FW 190 A4? Albo paląca kwestia, jak był pomalowany P-40, na którym Witold Urbanowicz latał nad Chinami w 1943r?

Otóż chciałbym teraz zaoferować kolegom modelarzom (i nielicznym koleżankom – jakoś tak się te proporcje układają) zupełnie nowy materiał i narzędzia. Zamiast własnego, czasami nieco pobrudzonego aerografem biurka — okno na inny świat, w głębi ekranu.

Jest tu niezwykle plastyczny materiał, który można "wytłaczać" w dowolne powłoki, bez obawy że go zabraknie. Są tu farby, dla których można dokładnie ustalić odcień i zasady nakładania. Jest tu możliwa do osiągnięcia dokładność, której nie uzyskasz nigdzie indziej, ani w skali 1:24, ani w 1:18. Tu, gdy stwierdzisz, że okapotowanie silnika samolotu, który zrobiłeś rok temu, powinno mieć inny kształt, zawsze możesz ten błąd poprawić. I to nie raz! Tu nigdy Twój model nie obrośnie kurzem. Nie będziesz wysłuchiwał narzekań domowników na to, że nie ma już gdzie zmieścić Twojej kolekcji. Możesz tu powielić swoje dzieło – choćby po to, by odwzorować je w kilku różnych wersjach malowania. I możesz przesłać swój model innym hobbystom, takim jak Ty, bez obawy że coś się z nim stanie podczas transportu. Z obiektów, które w ten sposób stworzysz, można szybko budować całe sceny.

Jedyne, do czego trzeba się przyzwyczaić, to to, że niczego w tym świecie nie można dotknąć. Możesz tylko patrzeć, i łapać wszystko myszką. Przynajmniej na obecnym etapie rozwoju technologii, urządzenia dotykowe (ang. *haptic devices*) są nadal drogie i prymitywne. Nie ma co liczyć, że pojawią się w sklepie za rogiem. Na pocieszenie mogę powiedzieć, że ich obecne modele pozwoliłyby Ci dotknąć wszystkiego tylko j e d n y m punktem – końcówką takiego „niby-długopisu”.

Ten świat wirtualnego modelarstwa stał niepostrzeżenie dostępny. W istocie każdy komputer, kupiony po 2005r, to silna stacja graficzna, o której w latach 90-tych można było tylko pomarzyć. Około 15 lat temu „ruszył z miejsca” nowy model matematyczny, służący do odwzorowania powierzchni. Mam na myśli powierzchnie podziałowe (ang. *subdivision surfaces*). Pozwoliły stworzyć Shreka i dziesiątki innych postaci z filmów animowanych za pomocą komputera. Ten nowy model matematyczny skierował do lamusa powierzchnie NURBS, wykorzystywane od lat 70-tych. Powierzchnie podziałowe pozwalają łatwo uzyskiwać naprawdę złożone kształty. W miarę dobrze radzą sobie ze zmorą NURBS – otworami i wycięciami.

Co więcej – odpowiednie programy, które potrafią to wszystko wykorzystać, stały się dostępne za darmo! Stworzyły je dziesiątki programistów *Open Source*. Ludzie ci chcą pokazać, że stać ich co najmniej na to samo, co twórców najlepszych programów komercyjnych. I to zaczęło im się udawać! Być może, trochę w tym zdrowej ambicji („ja to zrobię lepiej!”), oraz innego rytmu powstawania takich produktów. Podczas pracy nad nimi nie ma, typowego dla komercyjnych projektów pośpiechu, związanego z napiętymi terminami. (A pośpiech rodzi błędy). W efekcie pracy odpowiednio dużej grupy entuzjastów powstaje dobry, stabilny program.

Podsumowując – kupując do domu komputer do gier, kupiłeś wszystko, co potrzeba do wejścia w świat wirtualnego modelarstwa. Nie będę oszukiwał, że nie ma tu tego, co jest nieodłączną cechą pracy każdego modelarza: wielotygodniowego wysiłku. Mam jednak nadzieję, że ta książka ułatwi Ci, drogi Czytelniku, jak najszybsze osiągnięcie pożądaných rezultatów. Potem możesz pójść dalej i zrobić to samo jeszcze lepiej, niż tu proponuję. Przeczytanie dalszych stron, oszczędzi Ci dużo czasu i – niekiedy – frustracji. Ta ostatnia bywa nieodłącznym składnikiem pracy z „tą głupią maszyną” – komputerem. Postaram się nie zanudzać i pokazywać dużo obrazków, więc mam nadzieję, że "Wirtualne modelarstwo" Cię zainteresuje.

Witold Jaworski

## Co zawiera ta publikacja?

„Wirtualne Modelarstwo” uczy „od zera”, jak tworzyć takie modele samolotów, jak pokazany na okładce. Aby szybciej udostępniać Czytelnikom uaktualnione fragmenty tej książki, zdecydowałem się stworzyć i opublikować wyciągi z jej tekstu, związane z określoną dziedziną. Są to: „Wyciąg I: Przygotowania”, „Wyciąg II: Modelowanie”, „Wyciąg III: Materiały i tekstury”, oraz „Wyciąg IV: Detale i renderowanie”. Przebieg pracy prezentuję na przykładzie modelu myśliwca Curtiss P-40B. Maszyna ta zawiera klasyczne rozwiązania, stosowane w większości samolotów tego okresu. Z całym rozmysłem nie wybrałem Spitfire, Mustanga, Thunerbolta, Focke-Wulfa czy Messerschmita. Nie chciałem zabierać Ci przyjemności samodzielnego odwzorowania tych słynnych samolotów.

- Możesz tworzyć model każdego myśliwca z okresu II wojny światowej<sup>1</sup>, czytając jednocześnie kolejne rozdziały z tej książki. To wcale nie musi być P-40. Wszystkie te samoloty wykonuje się podobnie. W trakcie pracy na pewno zetkniesz się ze wszystkimi zagadnieniami, które są tu opisane.

Ta książka ma służyć zarówno tym, którzy dopiero zaczynają swoją przygodę „w 3D”, jak i tym, którzy mają już w tej dziedzinie pewne doświadczenie. W związku z tym zdecydowałem się ją podzielić na dwie części:

- część pierwsza ("Budowa modelu") to tekst podstawowy, który koncentruje się na tym, **co** trzeba zrobić;
- część druga ("Szczegóły obsługi programów") to szczegółowe opisy, **jak** posłużyć się odpowiednim programem, by osiągnąć efekt, pokazany w części pierwszej.

Cały tekst ma dużo stron, z czego "Szczegóły obsługi programów" zajmują ok. 40%. Układ tej części przypomina tekst systemu pomocy (*Help*) do programu. To krótkie (na jedną lub dwie strony), nie zawierające więcej niż kilka ilustracji, opisy pojedynczych poleceń.

Dzieląc materiał na część podstawową i „szczegółową”, starałem się uniknąć niepotrzebnych, wydłużających niezmiernie tekst, wskazówek w rodzaju "kliknij tu, a potem naciśnij tamto". W odpowiednich miejscach części pierwszej umieściłem odnośniki do części drugiej. Jeżeli nie znasz programu, o którym akurat piszę, korzystaj z tych odnośników i czytaj umieszczone w nich informacje szczegółowe<sup>2</sup>. Przygotowałem je tak, abyś nauczył się posługiwać wszystkimi narzędziami "od zera", w trakcie czytania głównego tekstu. Gdy już będziesz wiedział, **jak** zrobić to, co opisuję — po prostu przestaniesz do nich zaglądać.

Książka zawiera jeszcze jedną część: "Dodatki". To materiały, które mogą być ciekawe dla co dociekliwszych Czytelników. Tematyka "Dodatków" przypomina trochę "groch z kapustą": od pewnych zagadnień optyki (deformacja obrazu na fotografii), do odwzorowania profili lotniczych. A oprócz tego: metody zaawansowanej weryfikacji rysunków samolotu (na poziomie tworzenia planów modelarskich), oraz szczegółowy opis właściwości powierzchni podziałowych. Nie musisz do „Dodatków” zaglądać, choć sądzę, że niektórzy mogą w tych materiałach znaleźć dla siebie coś interesującego.

<sup>1</sup> No, może tych z silnikami rzędowymi. Nie opisałem tu, jak modelować widoczne z zewnątrz elementy chłodzonych powietrzem silników gwiazdowych — cylindry, popychacze, karter, przewody. Może w kolejnym wydaniu książki podjąłbym się modelu P-36?

<sup>2</sup> Zakładam, że korzystanie z części drugiej będzie zawsze wyrzutek. Stąd większość z zagadnień, które tam się znajdują, jest omówione bez dalszych odnośników. Takie podejście powoduje, że np. opis skalowania w edytorze siatki Blendera jest niemal dosłowną kopią opisu z edytora obiektów. Różnią się tylko ilustracjami i paroma zdaniami. Po prostu nie wiem, który z tych tematów otworzysz jako pierwszy, a w każdym chciałbym dostarczyć pełną informację.

## Konwencje zapisu

Wskazówki dotyczące klawiatury i myszki oparłem na założeniu, że masz standardowe:

- klawiaturę — w normalnym układzie amerykańskim, 102 klawisze (dodam także parę uwag o klawiaturze notebooka, bo sam na takiej pracuję);
- myszkę — wyposażoną w dwa przyciski i kółko przewijania (które daje się także naciskać: wtedy działa jak trzeci, środkowy przycisk).

Wywołanie polecenia programu będę zaznaczał następująco:

**Menu → Polecenie** - taki zapis oznacza wywołanie z menu „Menu” polecenia „Polecenie”. W przypadku bardziej zagnieżdżonych menu może wystąpić więcej strzałek!

**Panel: Przycisk** - taki zapis oznacza naciśnięcie w oknie dialogowym lub panelu "Panel" przycisku „Przycisk”. Czasami mogę także w ten sposób napisać o przełączniku lub liście rozwijalnej. ("Panel" to pojęcie związane z ekranem Blendera, wyjaśnienia — patrz „Wyciąg II: Modelowanie”)

Naciśnięcie klawisza na klawiaturze:

**Alt-K**

- myślnik pomiędzy znakami klawiszy oznacza jednoczesne naciśnięcie obydwu klawiszy na klawiaturze. W tym przykładzie trzymając wciśnięty **Alt**, naciskasz **K**;

**G, X**

- przecinek pomiędzy znakami klawiszy oznacza, że je naciskasz (i zwalniasz!) po kolei. W tym przykładzie najpierw **G**, a potem **X** (tak, jak gdybyś chciał napisać wyraz „gx”).

Naciśnięcie klawisza myszki:

**LPM**

- lewy przycisk myszy

**PPM**

- prawy przycisk myszy

**SPM**

- środkowy przycisk myszy (**naciśnięte** kółko przewijania)

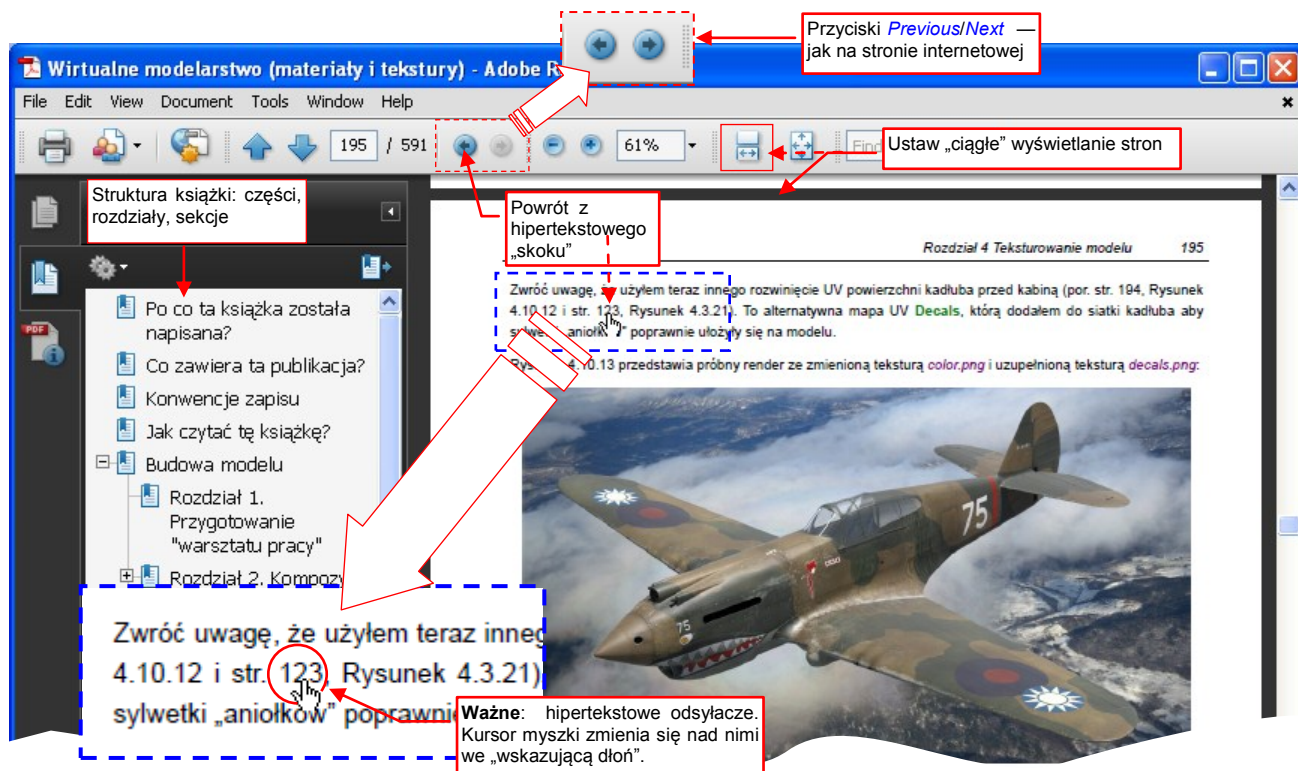
**KM**

- kółko przewijania (pełni tę rolę, gdy jest **obracane**)

Na koniec — „w kwestii formalnej”: jak mam się do Ciebie zwracać? Zazwyczaj w poradnikach używa się formy bezosobowej („teraz należy zrobić”). To jednak, mówiąc szczerze, czyni czytany tekst mniej zrozumiałym. Aby ta książka była jak najbardziej czytelna, zwracam się do Czytelnika w krótkiej, drugiej osobie („teraz zrób”). Czasami używam także osoby pierwszej („teraz zrobiłem”, „teraz zrobimy”). Tak jest mi łatwiej. Podczas pracy nad modelem traktowałem nas — czyli Ciebie, drogi Czytelniku, i siebie, piszącego te słowa — jako jeden zespół. Może trochę wyimaginowany, ale w jakiś sposób prawdziwy. Przecież pisząc tę książkę ja także wiele się uczyłem, bo wiedziałem, że każde zagadnienie mam Ci porządnie przedstawić!

## Jak czytać tę książkę?

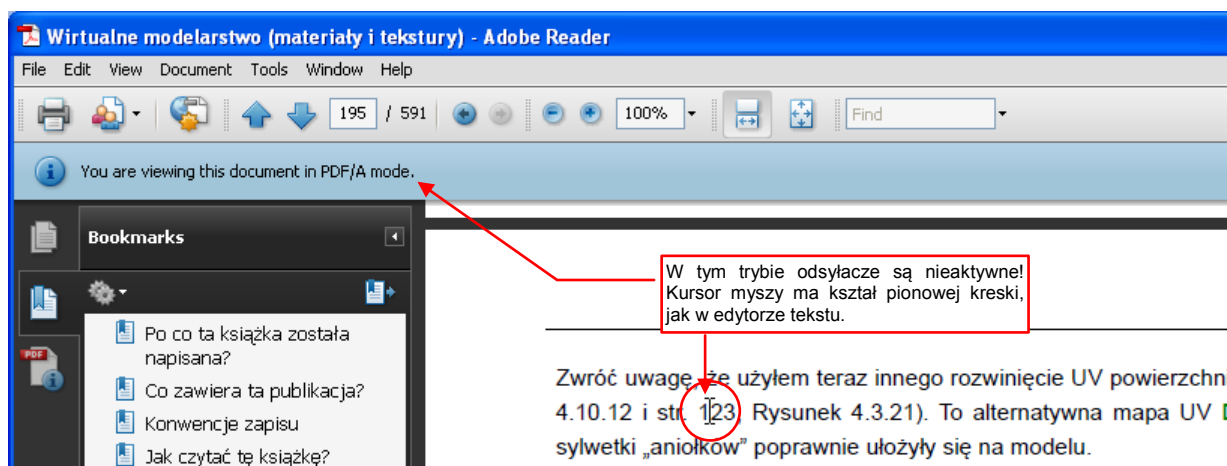
Starając się, by ta publikacja elektroniczna była bardziej czytelna niż typowe „papierowe” poradniki, przenieśliśmy tu większość szczegółowych opisów z głównego tekstu do pozdodziałów. W ich miejscu pozostawiłem odsyłacze („linki”, takie same jak w stemach podpowiedzi czy stronach internetowych). Aby nie „zaśmieszczać” tekstu nie wyróżniałem specjalnie w tych miejsc kolorem lub podkreśleniem. Możesz je jednak rozpoznać na podstawie kontekstu („por. str. ...”) a także po zmianie kształtu kursora myszki na „wskazującą dłoń” (Rysunek 1.1.1):



Rysunek 1.1.1 Czytanie książki w przeglądarce (Adobe Reader)

Gdy klikniesz w taki skrót, przejdziesz na stronę, na którą odsyła Cię tekst. Aby powrócić w poprzednie miejsce książki, wykorzystaj przycisk *Previous* umieszczony u góry ekranu (Rysunek 1.1.1). Działają tak samo, jak w przeglądarce internetowej. Aby podział stron nie stanowił przeszkody w czytaniu, możesz dodatkowo przełączyć wyświetlanie w tryb „ciągły” (Rysunek 1.1.1). Wydaje mi się, że tak wygodniej jest czytać „online”.

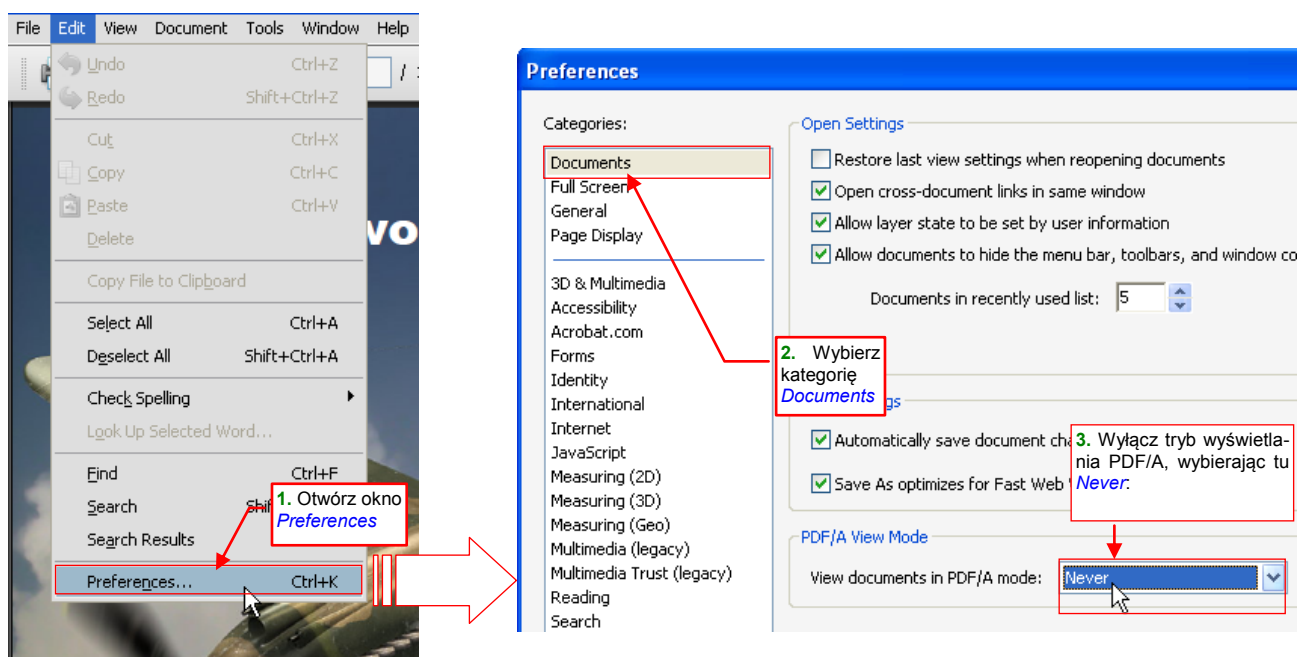
Aby ten dokument PDF można było poprawnie odczytać na różnych urządzeniach, jest zapisany w standardzie PDF/A. Jeżeli przeglądasz go za pomocą popularnego programu **Adobe Acrobat**, zwróć uwagę że domyślnie wszystkie hipertekstowe odsyłacze są w nim wyłączone (Rysunek 1.1.2):



Rysunek 1.1.2 Nieaktywne odsyłacze w trybie PDF/A (Adobe Reader)



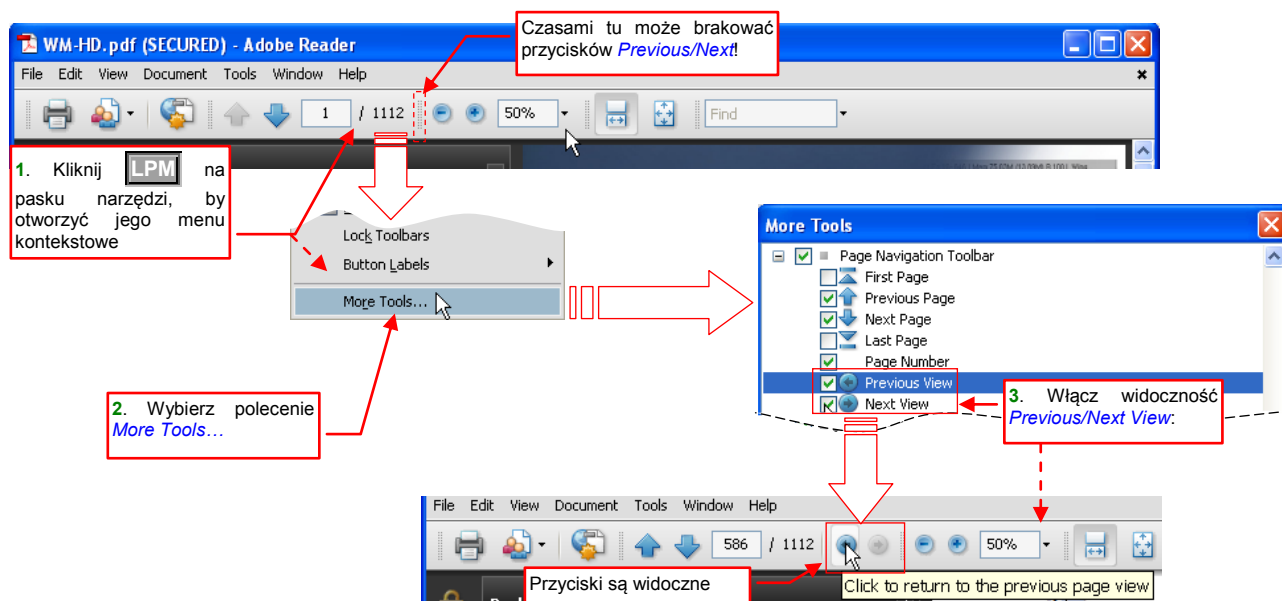
Aby uaktywnić hipertekstowe odsyłacze w dokumencie PDF/A, musisz zmienić pewien domyślny szczegół w konfiguracji **Adobe Acrobat** (Rysunek 1.1.3):



Rysunek 1.1.3 Uaktywnienie wewnętrznych odsyłaczy dla dokumentów PDF/A (Adobe Reader, wersja 9.0)

Z menu **Edit** otwórz okno dialogowe **Preferences**. Z listy kategorii wybierz **Documents**. Spowoduje to pojawienie się po prawej stronie okna kontrolki, jak na ilustracji (Rysunek 1.1.3). Wyłącz tryb **PDF/A View Mode** wybierając **Never** z jego listy rozwijalnej.

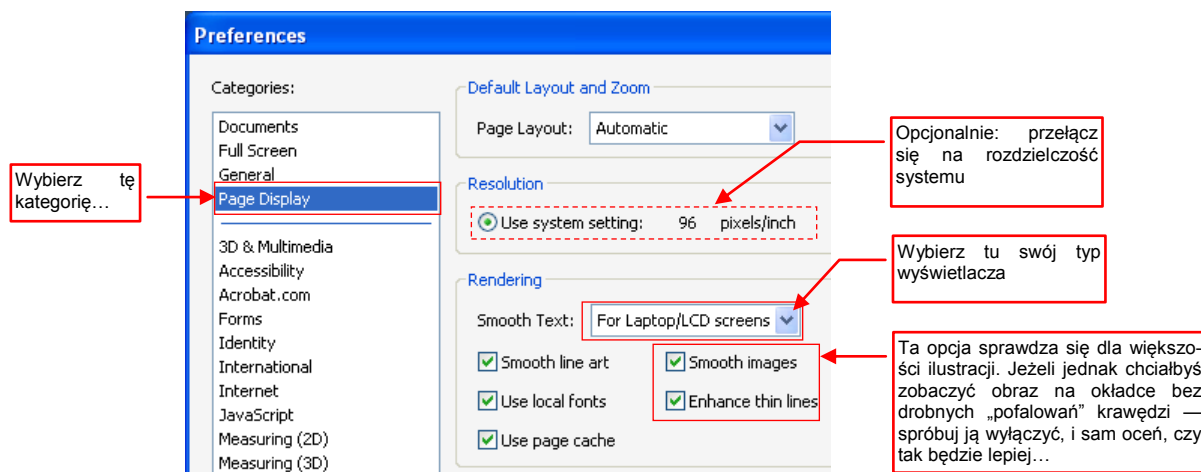
Kolejne wersje **Adobe Acrobat** mogą się od siebie różnić. Na przykład pasek narzędzi może się pojawić u dołu ekranu (jak to się przydarzyło wersji 8.0). Co więcej, w domyślnej konfiguracji przyciski **Previous/Next**, przydatne podczas posługiwania się odsyłaczami, mogą być ukryte (Rysunek 1.1.4):



Rysunek 1.1.4 Dodanie do paska narzędzi przycisków **Previous/Next** (Adobe Reader, wersja 9.0)

Choć zawsze można użyć skrótów klawiszowych (**Previous**: **Alt** + **←**, **Next**: **Alt** + **→**), to jakoś różniej jest mieć te przyciski „pod ręką”. Kliknij **LPM** na pasku narzędzi, i z menu kontekstowego wywołaj polecenie **More Tools...**. Znajdź i zaznacz w oknie **More Tools** polecenia **Previous View** i **Next View** (Rysunek 1.1.4).

W tej książce bardzo ważne są ilustracje — wypełniają prawie każdą stronę. (Właściwie to można ten poradnik określić jako „półkomiks”). Niestety, takie obrazy istotnie zwiększają rozmiar pliku PDF. Aby nie był zbyt duży, musiałem poddać ilustracje kompresji, która pogarsza ich jakość. To, co widzisz, to wynik kompromisu pomiędzy rozmiarem publikacji a ostrością detali obrazów. W przypadku przeglądarki **Adobe Acrobat** chciałbym zasugerować przestawienie paru parametrów, które mogą poprawić dokładność wyświetlania ilustracji. Nie są to ustawienia domyślne, więc powinieneś je teraz zmienić tak, jak pokazuje to Rysunek 1.1.5:



Rysunek 1.1.5 Parametry programu Adobe Reader (wersji 9.0), odpowiednie dla ilustracji w tej książce

Z listy kategorii okna **Preferences** wybierz **Page Display**. Spowoduje to pojawienie się po prawej stronie okna kontrolki, jak na ilustracji. W sekcji **Resolution** przełącz się na opcję **Use system settings**. (Domyślnie jest wybrana ta druga — a to może pogarszać jakość wyświetlania obrazów. W razie czego nie przejmuj się, jeżeli zobaczysz przy ustawieniach systemowych inną liczbę piks/cale niż tą, którą pokazuje Rysunek 1.1.5 — to zależy od tego, jak sobie ustawiłeś rozmiar tekstów w systemie Windows).

Oprócz tego, możesz zmienić jeszcze dwa, mniej istotne ustawienia. Pierwsze z nich to metoda wygładzania tekstu (lista rozwijalna **Smooth Text**). Acrobat ma domyślnie wybrany tryb **Monitor**. Zapewne używasz jakiegoś „cienkiego” ekranu LCD — wtedy warto ją zmienić na **For Laptop/LCD screens**. I wreszcie przełączniki **Smooth images** i **Enhance thin lines**. W większości przypadków lepiej jest, gdy pozostaną włączone (ustawienie domyślne). Jeżeli jednak samolot na okładce tej książki ma lekko pofalowane krawędzie skrzydeł, to możesz spróbować wyłączyć wygładzanie obrazów. Sam oceń, czy bez tej opcji wygląda lepiej, czy nie.

\* \* \*

Zawsze możesz wydrukować egzemplarz „Wirtualnego modelarstwa” dla swoich potrzeb<sup>1</sup>. Możesz go także nie drukować i czytać podczas pracy nad modelem wprost z ekranu. Oszczędzisz w tym przypadku trochę papieru — zawsze to o skrawek lasu więcej. Co więcej, możesz wtedy korzystać z hipertekstowych odsyłaczy i widzieć ilustracje w pełnym kolorze. Oczywiście, jeżeli wolisz częste kartkowanie prawdziwego papieru — drukuj całość. Nawet jeżeli Twoja drukarka obsługuje wydruk dwustronny, zużyjesz całą ryzę A4! Zapewniam, że nie było to moim celem ☺.

<sup>1</sup> Nie chciałem w tej elektronicznej książce rezygnować z kolorowych ilustracji. W związku z tym wiele z nich na wydruku wydaje się być mało kontrastowa. Aby temu w jakimś stopniu zaradzić, na niektórych ilustracjach zmieniałem kolory odnośników (na ciemniejsze lub jaśniejsze). Dzięki temu na wydruku są w miarę czytelne.

## Budowa modelu

W tej części kontynuujemy pracę nad uformowanym modelem P-40B. Jego stworzenie jest opisane we wcześniejszych wyciągach z „Wirtualnego modelarstwa” (patrz „Wyciąg I: Przygotowania”, „Wyciąg II: Modelowanie”).

## Rozdział 1. Przygotowanie "warsztatu pracy"

W tym rozdziale omówimy pokrótce instalację oprogramowania, z którego będziemy korzystać. W tym wyciągu to:

- **Blender**: program podstawowy;
- **GIMP**: pomocniczy, do edycji obrazów rastrowych;
- **Inkscape**: pomocniczy, do edycji obrazów wektorowych;

To oprogramowanie jest udostępniane w zasadach licencji GPL. Oznacza to m.in., że nie można pobierać za nie żadnych opłat. (Oczywiście, jeżeli masz gest, a program Ci się bardzo spodoba, możesz wspomóc twórców dotacją — ale na zasadzie pełnej dobrowolności. Szczegółowy opis, jak to zrobić, znajdziesz na stronie internetowej każdego z tych projektów.)

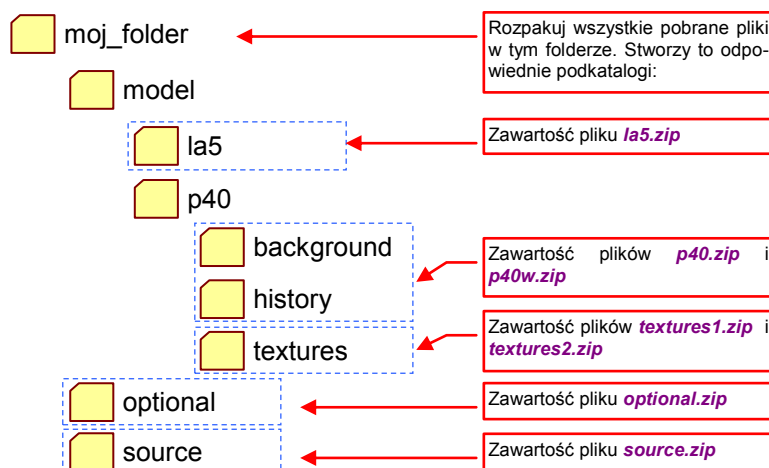
Przypuszczam, że większość czytelników tej książki używa komputera z Windows. Podczas opisu instalacji koncentruję się wyłącznie na tym środowisku. Sam go używam. Nie mam żadnych doświadczeń z Linuksem, ani z Mac OS, więc nie będę pisał o instalacjach, których nie wykonałem. Jeżeli używasz systemów innych niż Windows — poszukaj wskazówek na stronach tych programów (podam ich adresy w kolejnych sekcjach rozdziału).

Następujące pliki są ważną częścią książki:

- **source.zip**: zawiera skrypty Pythona, plany P-40B, udostępnione przez producenta fragmenty rysunków technicznych (P-40E), i inne pomocnicze materiały;
- **p40.zip**, **p40w.zip**: Zawierają historię zmian (pliki *\*.blend*) modelu P-40, oraz kilka obrazów tła, wykorzystanych w tej książce. Każdej sekcji tej książki odpowiada plik ze spakowanego folderu *history*!;
- **textures1.zip**, **textures2.zip**: folder, zawierające kolejne wersje tekstur odpowiadające poszczególnym modelom z pliku *p40.zip*, i robocze pliki, wykorzystywane do ich stworzenia. (Podzieliłem te dane na dwie części ze względu na znaczny rozmiar zawartych w nich plików rastrowych);

- Pobierz powyższe pliki ze strony: [http://samoloty3d.pl/wm2\\_p.xml](http://samoloty3d.pl/wm2_p.xml). Utrzymuję tam ich aktualne wersje<sup>1</sup>.

Rozpakuj wszystkie pobrane pliki *\*.zip* do tego samego folderu. W trakcie ich rozpakowania stworzą odpowiednie podkatalogi (pliki nie wymienione powyżej nieżą do innych części tej publikacji):



<sup>1</sup> Na tej stronie znajdziesz także odpowiedzi na najczęstsze pytania zadawane przez Czytelników, oraz informacje o zmianach w opisywanym oprogramowaniu, jakie się pojawiły po publikacji tej książki.

## 1.1 Instalacja Blendera

Ta książka wymaga użycia Blendera w wersji **2.6**. Gdy zaczynałem ją pisać, aktualną wersją była 2.59, więc na jej przykładzie pokażę, jak można zainstalować ten program. (Sposób instalacji nie uległ zmianie).

Najpierw pobierz pliki programu ze strony Fundacji Blendera: — <http://www.blender.org>. Od lat pliki do pobrania są dostępne pod adresem: <http://www.blender.org/download/get-blender> (Rysunek 1.1.1). Nie mogę jednak gwarantować, że za jakiś czas (rok, dwa) może to się trochę zmienić, więc w razie czego zacznij od strony głównej<sup>1</sup>.



Rysunek 1.1.1 Strona, z której można pobrać pliki Blendera (na przykładzie wersji dla Windows, 32 bit)

Pod wskazanym adresem znajdziesz skróty do wersji programu skompilowanych dla różnych systemów operacyjnych (Windows, Linux, Mac OS, ...). Na ilustracji widać, że wybrałem wersję dla Windows, 32 bity. Czynię to wyłącznie ze względu na mój system operacyjny (posługuję się jeszcze starym Windows XP). Na potrzeby tego, co będziemy robić w tej książce, ta wersja jest wystarczająca. Jeżeli jednak używasz Windows 7, lepiej pobierz 64-bitową wersję Blendera<sup>2</sup>.

Każda z wersji dla Windows jest udostępniona w dwóch postaciach: programu instalacyjnego i spakowanego pliku \*.zip.

<sup>1</sup> Na stronach projektu można znaleźć także programy instalacyjne wcześniejszych wersji programu. Wspominam o tym, gdyż być może będziesz czytał te słowa już po opublikowaniu nowszych wersji. Do pracy z „WM” zalecam wersję 2.6, której używałem pisząc tę książkę.

<sup>2</sup> W wersji 64-bit dla program ma do dyspozycji więcej pamięci RAM, co pozwala mu na wykorzystanie tekstury o rozmiarach większych niż 4000x4000 pikseli. Ta właściwość może być bardzo przydatna, gdy będziesz chciał dokładniej odwzorować powierzchnię modelu. W tej książce największa tekstura, jakiej użyjemy, będzie miała rozmiar 4096x4096 pikseli. Gdybyś jednak zaczął eksperymentować z generowaniem bardzo dużego obrazu — o bokach nie krótszych niż 6 tys. pikseli, z teksturami o rozmiarach 8192x8192 px — to wersja 32-bitowa zakończy generację z błędem, a wersja 64-bitowa wykona ją poprawnie do końca. (Obraz o bokach 6000 x 4000 px, przy „posterowej” rozdzielczości 300 dpi, odpowiada karcie o rozmiarach 50x34 cm)



Jeżeli jesteś bardzo przyzwyczajony do programów instalacyjnych Windows, pobierz go i uruchom. Po wskazaniu folderu, w którym należy umieścić Blendera, będziesz miał po chwili zainstalowany program.

Ja jednak należę do ludzi, którzy wolą dokładnie wiedzieć, co się dzieje z ich komputerem<sup>1</sup>. Rzecz w tym, że Blender prawie „nie dotyka” reszty systemu. Dlatego najlepszym (jak uważam) sposobem instalacji Blendera jest rozpakowanie pliku \*.zip. W ten sposób po prostu wiesz dokładnie, co robisz. Wystarczy:

1. pobrać ze strony BF plik \*.zip (Rysunek 1.1.1), zawierający folder z jego plikami;
2. rozpakować zawartość tego pliku na lokalny dysk, np. do folderu *C:\Program Files\Blender*;
3. dodać skrót do *C:\Program Files\Blender\blender.exe* na pulpit i/lub do menu *Start*;
4. opcjonalnie: skojarzyć rozszerzenie \*.blend z tym programem (np. przy okazji otwierania przykładowego pliku);

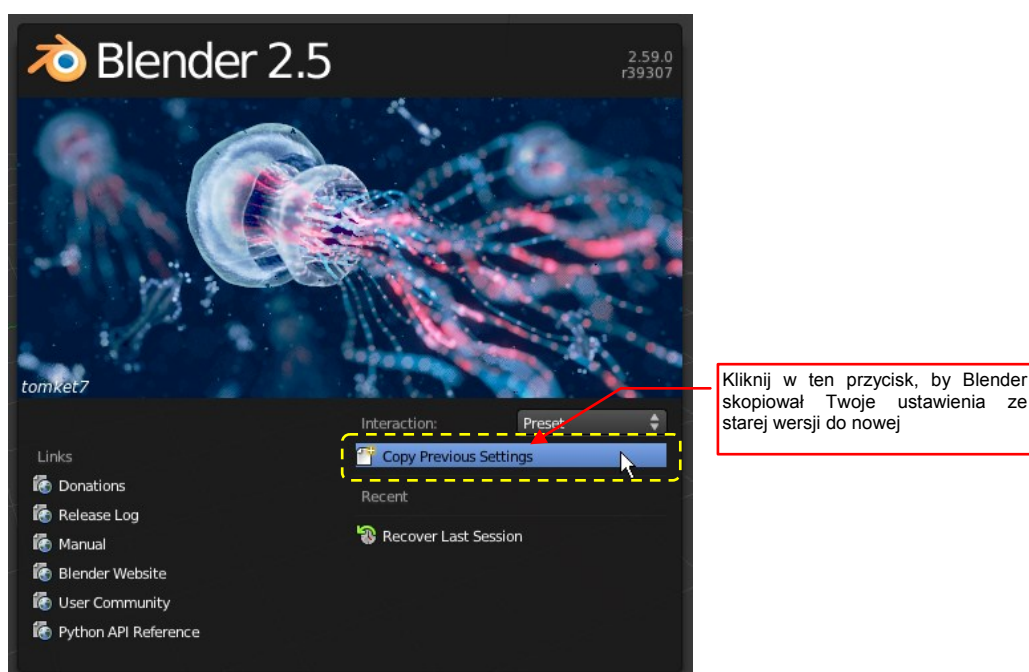
Jeżeli masz jakiegokolwiek wątpliwości co do przebiegu tej nieskomplikowanej procedury, szczegółowy opis, jak to robię znajdziesz w Wyciągu II.

Gdy już używasz Blendera, i chcesz wgrać nową wersję — nie musisz wcale po raz kolejny uruchamiać programu instalacyjnego. (Choć, oczywiście, możesz to zrobić). Moje *upgrade* do nowej wersji polega na:

1. zmianie nazwy aktualnego folderu *Blender* na "*Blender-XX.X*", gdzie XX.X to numer starej wersji. Na przykład: przed wgraniem wersji 2.59 zmieniłem nazwę dotychczasowego folderu z *Blender* na *Blender-2.58*;
2. pobraniu ze strony BF pliku \*.zip z nową wersją Blendera;
3. rozpakowaniu plików Blendera do folderu o takiej samej nazwie, jaką zmieniliśmy w pkt 1.

(W ten sposób wszystkie skróty do Blendera w menu *Start* i skojarzenie rozszerzenia plików \*.blend samoczynnie przechodzą na nową wersję — nie trzeba niczego więcej zmieniać).

Przy pierwszym uruchomieniu Blender sam zaproponuje mi przeniesienie konfiguracji ze starej wersji (Rysunek 1.1.2) — wystarczy nie przeoczyć tego momentu i kliknąć przycisk *Copy Previous Settings*:



Rysunek 1.1.2 Okno powitalne (*splash screen*) przy pierwszym uruchomieniu nowej wersji Blendera

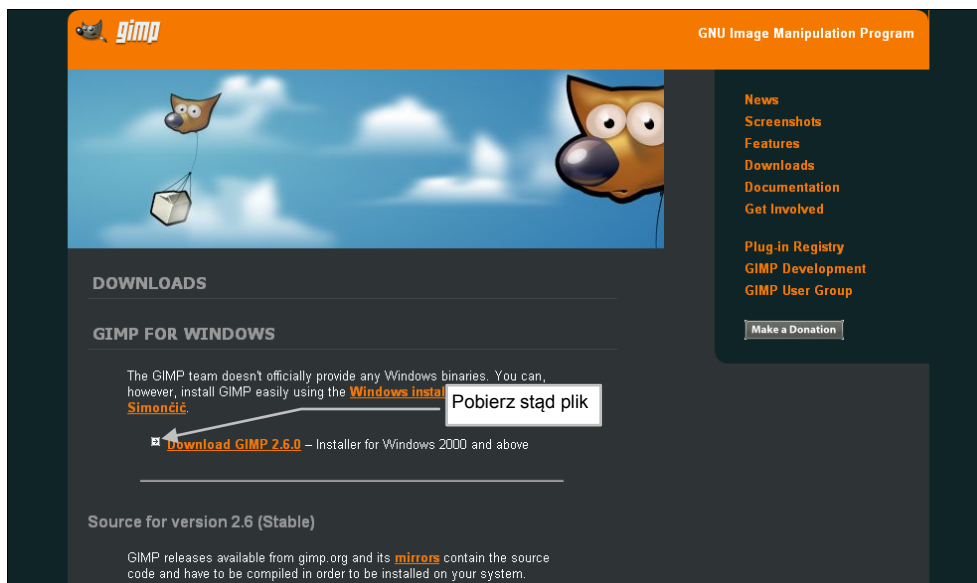
<sup>1</sup> Dzięki temu przez lata używam tej samej instalacji Windows — nigdy nie musiałem ich instalować na nowo! I nawet z upływem czasu nie spowalniają swojego działania, jak to nieraz widziałem na komputerach innych użytkowników...

Dzięki zachowywaniu poprzednich wersji Blendera w folderze o zmienionej nazwie, możesz zawsze z niej skorzystać. Wystarczy uruchomić plik *blender.exe* z jej katalogu.

- Możesz mieć na swoim komputerze równocześnie używać wielu różnych wersji Blendera. Wystarczy, że każda z nich znajduje się w innym katalogu.
- Do poprawnego działania Blender nie potrzebuje żadnej instalacji Pythona. (Możesz o niej napotkać wzmianki w Internecie. Interpreter Pythona był opcjonalnym dodatkiem do jego wcześniejszych wersji — np. 2.4). Blender od wersji 2.5 korzysta z własnego, wbudowanego interpretera tego języka. Jest on używany np. w obsłudze interfejsu użytkownika (okienek, przycisków, i menu programu).

## 1.2 Instalacja GIMP

Aby zainstalować GIMP, zajrzyj na stronę tego projektu — <http://www.gimp.org>. W chwili, gdy piszę ten rozdział, plik do pobrania jest dostępny pod adresem: <http://www.gimp.org/downloads> (Rysunek 1.2.1)<sup>1</sup>:



Rysunek 1.2.1 Strona, z której można pobrać program instalacyjny GIMP

Jak wynika z opisu, GIMP jest programem Open Source w każdym calu: wymaga, by użytkownik sam pobrał źródło programu i je skompilował! Na szczęście znalazła się dobra dusza (Jernej Simončič), która to zrobiła za nas. W dodatku spakowała to wszystko w plik instalacyjny Windows. A twórcy Gimp'a, niby tego oficjalnie nie popierają, ale umieścili na tej stronie poręczny skrót...

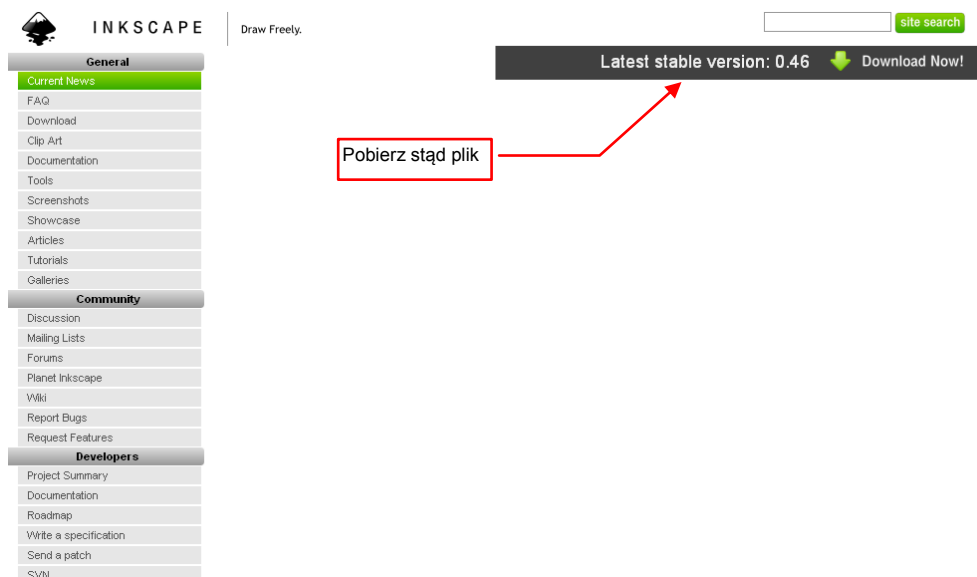
Proces instalacji Gimp'a nie zawiera żadnych "haczyków". Wystarczy, że będziesz naciskał wszystkie przyciski domyślne. Gdybyś miał jakieś wątpliwości — szczegółowy opis instalacji znajdziesz w Wyciągu I.

Podczas pracy nad tą książką używałem GIMP v. 2.66.

<sup>1</sup> Wersję 2.66, której używałem, możesz także znaleźć pod <http://www.samoloty3d.pl/downloads/wm/install/gimp/gimp-2.6.6-i686-setup.exe>. Znajdziesz tam także pliki pomocy (po angielsku): <http://www.samoloty3d.pl/downloads/wm/install/gimp/gimp-help-2.2-eng-setup.exe>

### 1.3 Instalacja Inkscape

Zajrzyj na stronę projektu Inkscape — <http://www.inkscape.org> (Rysunek 1.3.1)<sup>1</sup>:



Rysunek 1.3.1 Strona projektu Inkscape

Plik instalacyjny jest bezpośrednio dostępny z głównej strony projektu.

Podczas instalacji Inkscape wystarczy, gdy będziesz naciskał wszystkie przyciski domyślne. Gdybyś miał jakieś wątpliwości — szczegółowy opis instalacji znajdziesz w Wyciągu III.

W tej książce używałem Inkscape w wersji 0.47

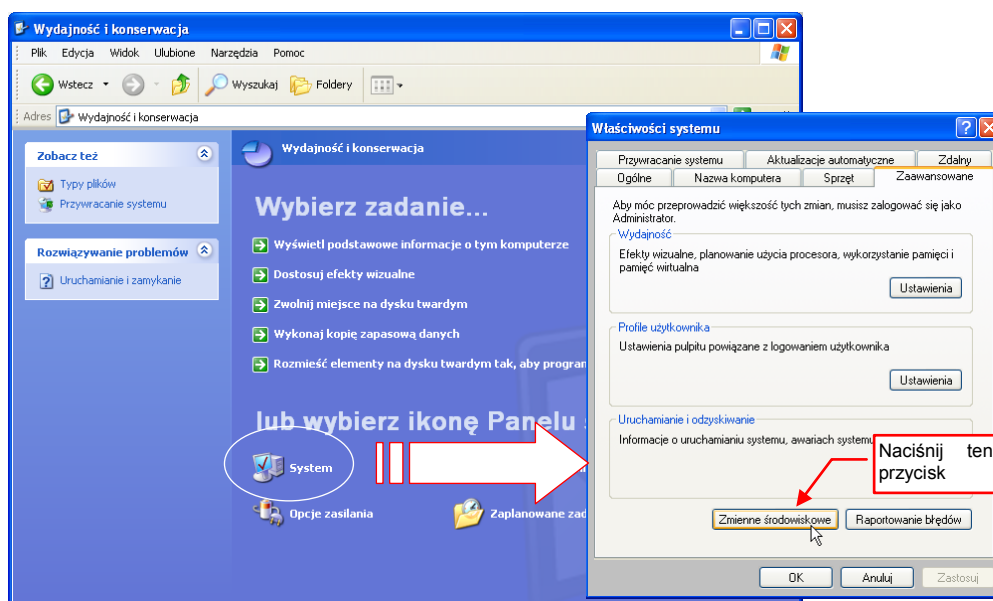
<sup>1</sup> Wersję instalacyjną Inkscape 0.47 znajdziesz także na <http://www.samoloty3d.pl/downloads/wm/install/inkscape/Inkscape-0.47-3.exe>

## 1.4 Przetawienie Inkscape i GIMP na język angielski

Podczas instalacji w polskiej wersji Windows, GIMP i Inkscape są na tyle "uprzejme", że samoczynnie wybierają polską wersję językową. Mimo, że ich tłumaczenia na nasz język są naprawdę dobre, nie mogę tego samego powiedzieć o tłumaczeniu Blendera. Zdecydowanie wolę w nim pracować w języku angielskim. Zresztą — wszystkie polskie poradniki opisują właśnie wersję anglojęzyczną. W tej sytuacji sądzę, że lepiej, aby dwa pomocnicze programy — GIMP i Inkscape — także były po angielsku. W ten sposób można uniknąć czasami wątpliwości, czy np. komunikaty w Blenderze i Gimpie na pewno mówią o tym samym.

GIMP i Inkscape zostały napisane za pomocą tej samej biblioteki okienek - GTK+. Przełączenie ich języka, jakiego dokonamy, polega na ustawieniu tzw. zmiennej środowiskowej o nazwie **LANG**. Dla uszu kogoś, kto nie jest informatykiem brzmi to, być może, strasznie. Spokojnie, rzecz jest w istocie prosta, i już pokazuję, jak to zrobić.

Wybierz z menu **Start** Windows polecenie **Panel Sterowania**. Jeżeli używasz tzw. "układu klasycznego" (wszystkie ikony panelu sterowania w jednym folderze) — wybierz ikonę o nazwie **System**<sup>1</sup>. Jeżeli używasz domyślnego układu Windows XP — ikonę tę znajdziesz w grupie **Wydajność i konserwacja** (Rysunek 1.4.1):



Rysunek 1.4.1 Otwarcie okna **Właściwości systemu**.

W oknie **Właściwości systemu** przejdź do zakładki **Zaawansowane**. Naciśnij tam przycisk **Zmienne środowiskowe** (Rysunek 1.4.1).

<sup>1</sup> To samo okno możesz także otworzyć, wywołując z menu kontekstowego ikony **Mój Komputer** polecenie **Właściwości**.



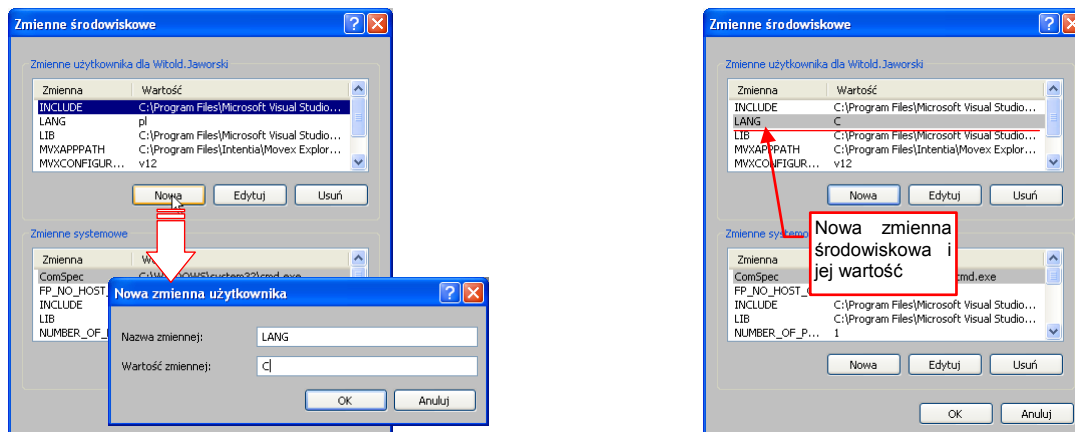
Spowoduje to otworenie okna o tej samej nazwie (*Zmienne środowiskowe*).

Naciśnij przycisk **Nowa** w sekcji *Zmienne użytkownika* (Rysunek 1.4.2). Spowoduje to otwarcie formularza *Nowa zmienna użytkownika*. Wpisz w jego pola:

- **Nazwa zmiennej:** **LANG**;
- **Wartość zmiennej:** **C**.

i naciśnij **OK** (Rysunek 1.4.2).

To wszystko: nowa zmienna środowiskowa została dodana do systemu (Rysunek 1.4.3):



Rysunek 1.4.2 Windows: dopisywanie nowej zmiennej środowiskowej

Rysunek 1.4.3 Dodana zmienna środowiskowa **LANG**

Gdybyś się chciał za jakiś czas przełączyć z powrotem na język polski — możesz to zrobić w oknie *Zmienne środowiskowe* dwoma drogami:

- zmieniając (przyciskiem **Edytuj**) wartość zmiennej **LANG** z "C" na "pl";
- usuwając (przyciskiem **Usuń**) zmienną środowiskową **LANG**;

## Rozdział 2. Pozostałe detale samolotu

W tym rozdziale zakończymy formowanie naszego samolotu. Zaczniemy od typowej sytuacji — skorygowania tego, co zrobiliśmy wcześniej na podstawie nowych materiałów referencyjnych. Jeżeli robisz dokładny model, praca zawsze posuwa się po takiej „spirali”: od czasu do czasu trzeba wrócić do rzeczy, które już wykonałeś poprzednio, i je poprawić. Pokażę to w sekcji 2.1 na przykładzie poważnej modyfikacji kształtu kadłuba.

Następnie dodamy do modelu mnóstwo drobnych części — na przykład elementy podwozia. Nie będziemy ich rozwijać i tekstuować, jak to robiliśmy z podstawowymi powierzchniami samolotu w poprzednim rozdziale. Zamiast tego do ich pokrycia użyjemy prostszych, jednorodnych materiałów, opartych o tekstury proceduralne.

W tym rozdziale nie będę opisywał wykonania każdego detalu. Skupię się raczej na tych, które wymagają zastosowania jakiejś nowej, nie omawianej do tej pory metody. Aby nabrać wprawy, pracę nad szczegółami zaczniemy od stworzenia kółka ogonowego. To dość prosty mechanizm, więc łatwo pokazać na jego przykładzie wszelkie niuanse nowych technik. Potem przejdziemy do bardziej złożonych całości: najpierw podwozia głównego, a później — wnętrza kabiny pilota.

Gdy skończysz kabinę pilota, cała reszta pracy wyda Ci się już łatwa i bardzo prosta!

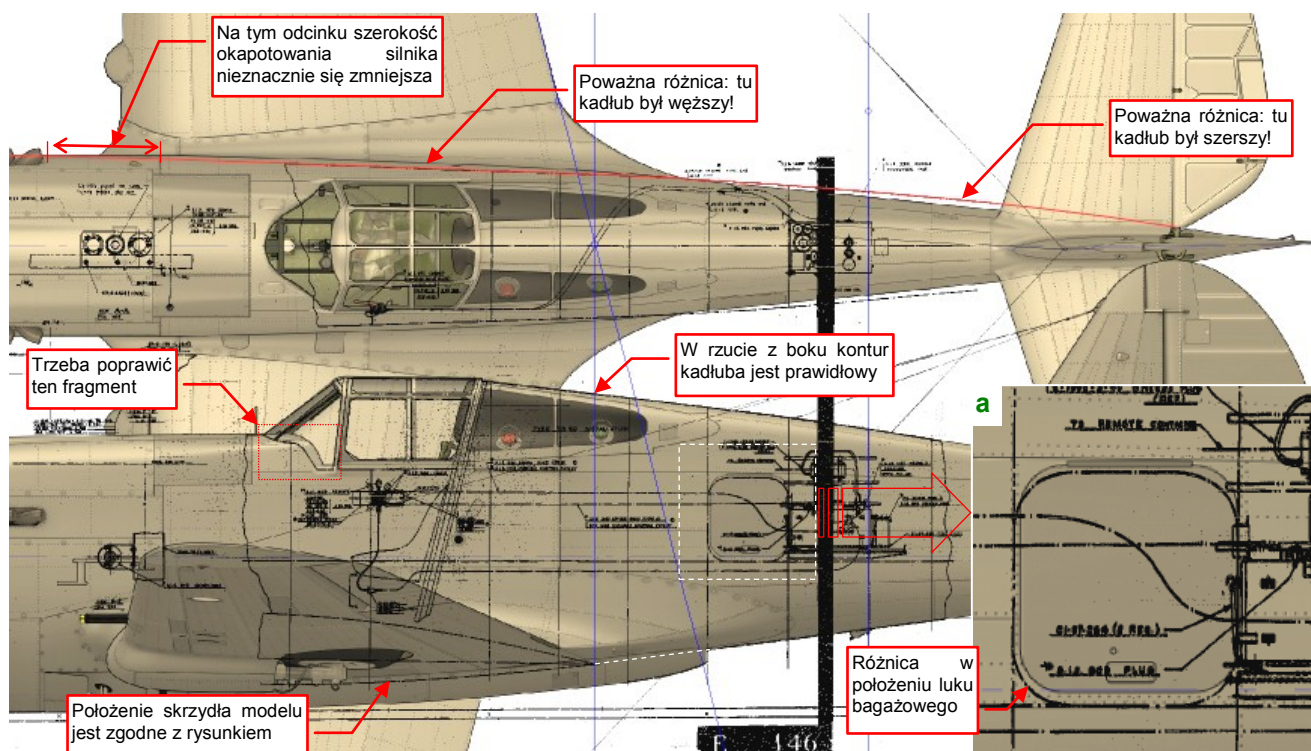
- W tym rozdziale będziemy wykorzystywać materiały z towarzyszącego tej książce pliku [source.zip](#). Pliki [\\*.blend](#) odpowiadające rezultatom poszczególnych sekcji rozdziału znajdziesz w pliku [p40.zip](#), a foldery z obrazami ich tekstur — w pliku [textures2.zip](#). (Adres miejsca, z którego można pobrać te pliki, znajdziesz na str. 16).

Numery rozdziałów w tym wyciągu są inne niż w kompletnej książce, dlatego pliki z rezultatami sekcji tego rozdziału noszą nazwy zaczynające się od [P40B-8.\\*](#), a foldery z teksturami — [08.\\*](#).

## 2.1 Korekta kształtu kadłuba

Podczas pracy nad modelem, prędzej czy później natrafisz na nowe materiały o samolocie, który tworzysz. (Czasami możesz natrafić na nie wtedy, gdy cała praca jest już skończona). Analizując takie nowe zdjęcia czy rysunki można znaleźć błędy w kształcie modelu. Poprawa takich niezgodności wiąże się często z dużym nakładem pracy. Jest to jednak konieczne, jeżeli chcesz jak najdokładniej odtworzyć pierwowzór. W tej sekcji pokażę jak możliwie prosto wprowadzić taką poważną zmianę kształtu.

Podczas pracy nad drugą edycją książki otrzymałem dzięki uprzejmości Hume Bates'a dwa nowe rysunki P-40B/C. Dotyczą wnętrza kabiny, ale widać na nich także kontur tylnej części kadłuba. Czym prędzej więc zestawiałem je z naszym modelem (Rysunek 2.1.1):



Rysunek 2.1.1 Porównanie otrzymanych rysunków fabrycznych z modelem

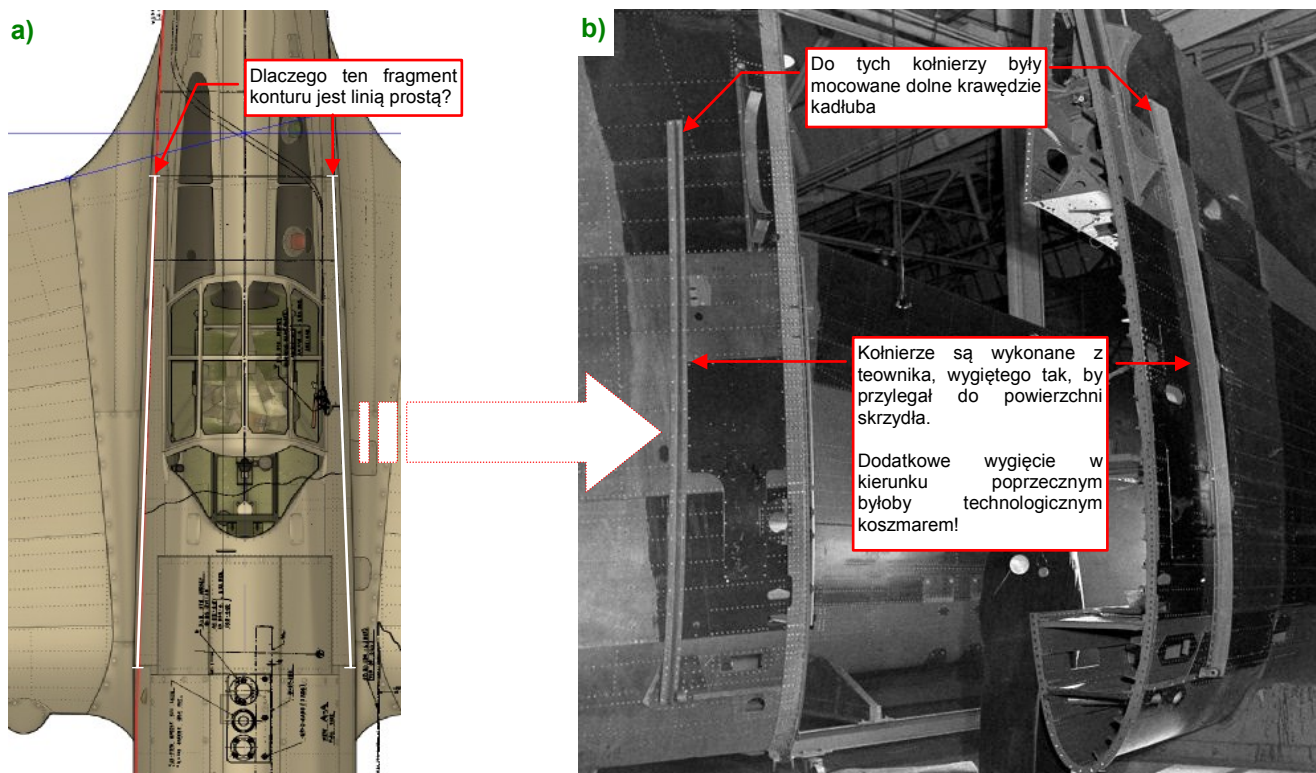
Użyłem tu rzutów z boku i z góry moich planów, które odpowiadają modelowi stworzonemu w pierwszej edycji „Wirtualnego modelarstwa”. (Kształt tego modelu jest taki sam jak ten, którego używaliśmy do tej pory).

Rysunek 2.1.1 pokazuje, że w rzucie z boku kadłub ma w zasadzie poprawny kształt. Zgadza się także położenie skrzydła. Do poprawki jest dolna część wiatrochronu kabiny pilota. (Widać źle zinterpretowałem zniekształcenie perspektywiczne na zdjęciach). Długo zastanawiałem się nad położeniem luku bagażowego, który na rysunkach fabrycznych jest odrobinę przesunięty w stosunku do tego na modelu (Rysunek 2.1.1a). Problem polegał na tym, że jednocześnie jego obramowanie (wręgi i podłużnice) są zgodne z dokumentacją. Po analizie dostępnych zdjęć doszedłem do wniosku, że to może być efekt jakiegoś przesunięcia kalki na desce kreślarskiej. (Na zdjęciach górna krawędź luku leży bliżej podłużnicy, niż widać to na tych rysunkach).

Za to w rzucie z góry widać poważne różnice pomiędzy kształtem modelu i konturem z dokumentacji technicznej. W zasadzie nie powinno mnie to dziwić: do tej pory nie było z czym go zweryfikować. (Jak to opisuje Wyciąg I, odpowiednie zdjęcia z góry praktycznie nie istnieją). Błędny kształt kadłuba w tym rzucie występuje na najwcześniejszych planach modelarskich, jakie odnalazłem<sup>1</sup>.

<sup>1</sup> Jest to rysunek P-36 z albumu Paula R. Matta z 1969r. Sądję, że korzystał z niego Jacek Jackiewicz, opracowując w 1999r. plany P-36 i P-40 dla monografii AJ-Press. Te rysunki były z kolei wykorzystywane przez wielu innych autorów, w tym Mariusza Łukasika.

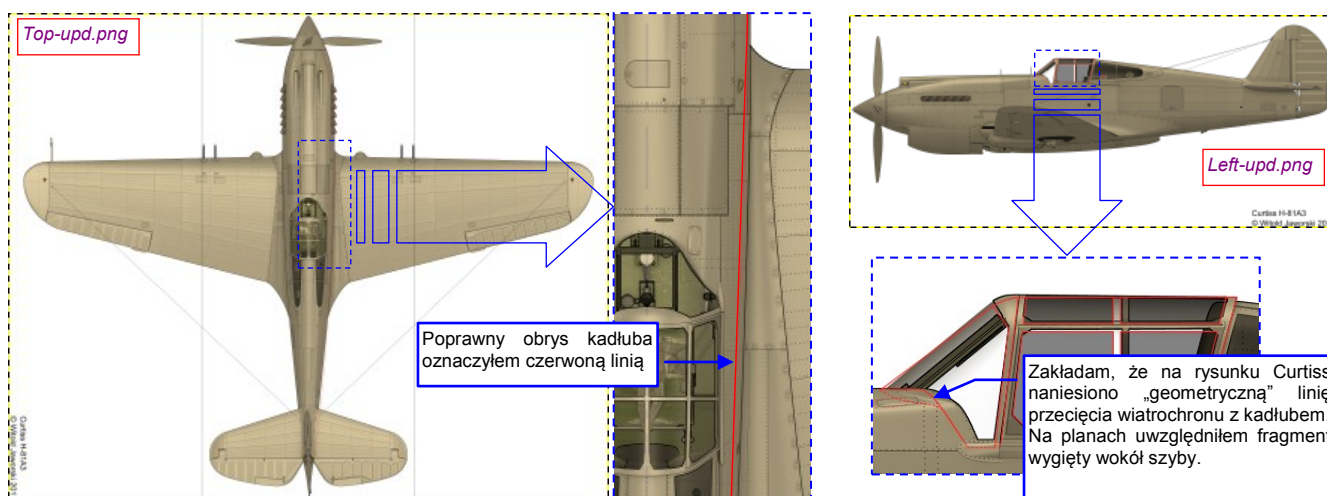
Szczególnie zainteresował mnie długi, prosty fragment konturu kadłuba ponad skrzydłem (Rysunek 2.1.2a):



Rysunek 2.1.2 Przyczyna płaskich boków kadłuba: kołnierze mocujące skrzydła (zdjęcie: [Life](#))

Okazało się, że taki kształt nie był przypadkowy, i wynikał ze względów technologicznych: kadłub był mocowany do płata za pomocą kołnierza, ciągnącego się wzdłuż burt kabiny (Rysunek 2.1.2b). Z załączonego zdjęcia wynika, że ten kołnierz był łagodnie wygięty fragment przekroju o kształcie litery „T”. Przypuszczam, że dodatkowe wygięcie w kierunku poprzecznym, które odwzorowałoby jakąś krzywiznę kadłuba byłoby bardzo kłopotliwe (to znaczy: drogie). Stąd łatwiej było dopasować na tym odcinku kadłub do kołnierza, pozostawiając proste krawędzie burt kabiny w płaszczyźnie poziomej. I choć na wszystkich innych planach P-40 ten fragment jest lekko wygięty, to ostatecznie przekonało mnie do poprawności konturu z rysunków fabrycznych.

Naniosłem odpowiednie poprawki czerwonymi liniami na plany samolotu (Rysunek 2.1.3):



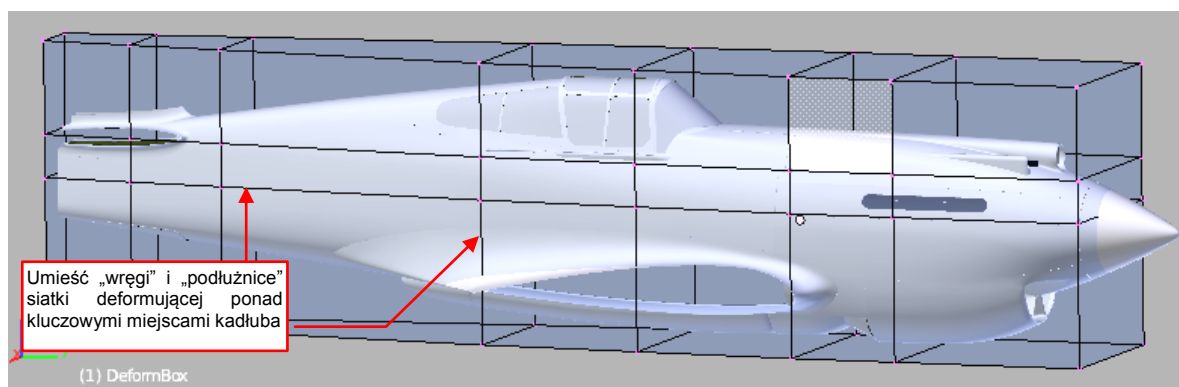
Rysunek 2.1.3 Korekty, naniesione na plany samolotu

Ten skorygowany rzut z lewej (plik [Left-upd.png](#)) i z góry (plik [Top-upd.png](#)) znajdziesz w materiałach pomocniczych (w folderze [source\plans\p40b-highres](#) — por. str. 16).



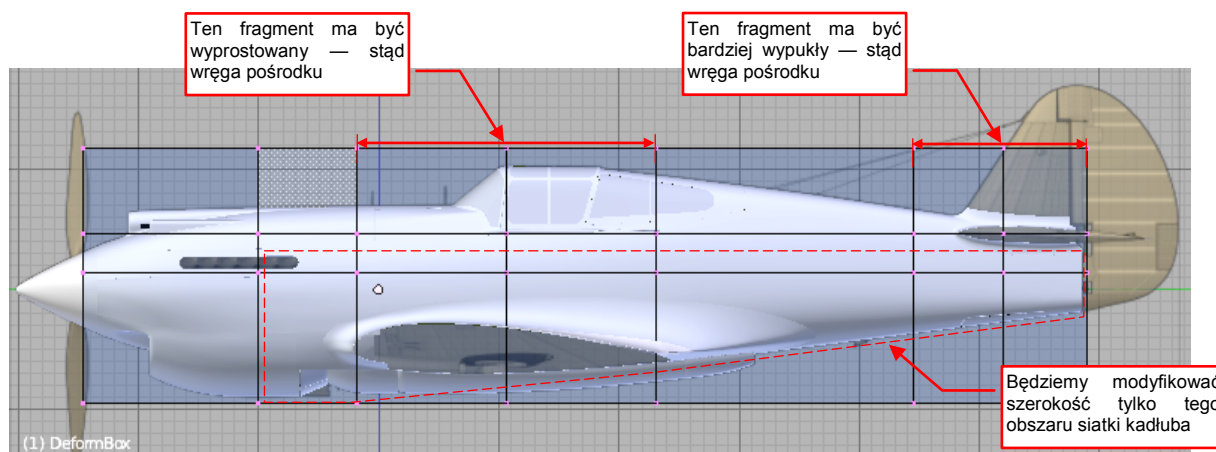
Jak teraz dopasować kadłub samolotu do tych zmodyfikowanych planów? Zmiany dotyczą wielu siatek naraz. Ręczna modyfikacja każdej z nich byłaby pracochłonna. W dodatku wiąże się z ryzykiem powstania jakichś szczelin pomiędzy poszczególnymi częściami. Co gorsza, modyfikowane siatki są stosunkowo gęste. Trudno będzie uniknąć na ich powierzchni drobnych pofałdowań. Jedynym ułatwieniem jest to, że te zmiany dotyczą tylko kształtu siatek — nie musimy (na szczęście) modyfikować tekstur. Dopóki nie usuniemy z poprawianych siatek żadnej ściany, dopóty współrzędne mapowania tekstur także pozostaną nienaruszone.

W takiej sytuacji dobrym rozwiązaniem wydaje się zmiana wszystkich tych części równocześnie, za pomocą modyfikatora *Mesh Deform* (por. str. 309). Zaczniemy od obudowania kadłuba odpowiednim „pudełkiem” (Rysunek 2.1.4):



Rysunek 2.1.4 Pomocnicza siatka, której użyjemy do zmiany kształtu kadłuba

Nazwijmy tę tymczasową siatkę *DeformBox*. Wstaw do niej (*Loop Cut*) na poziome i pionowe krawędzie ponad kluczowymi miejscami kadłuba. Gdzie dokładnie trzeba je umieścić? W zasadzie tam, gdzie w rzucie z boku biegną odpowiednie węż i podłużnice. Musisz jednak z tym trochę sam poeksperymentować (Rysunek 2.1.5):

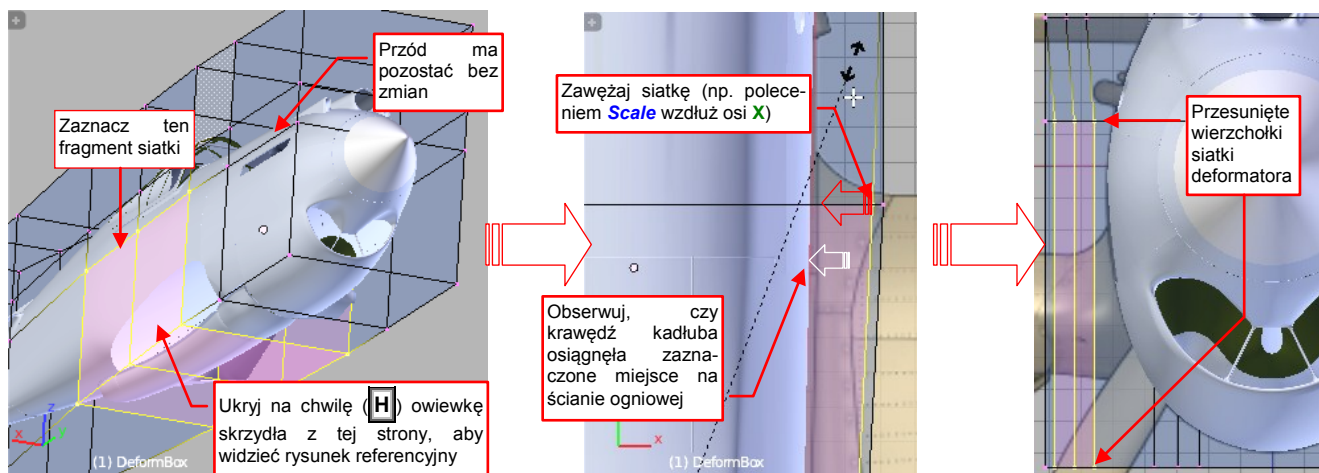


Rysunek 2.1.5 Rozmieszczenie krawędzi siatki pomocniczej

I właśnie przy okazji tych „eksperymentów” natkniesz się na mankamenty użycia modyfikatorów *Mesh Deform*. Choć na czas prób wystarczy je przypisać do głównych elementów kadłuba, to i tak ich „przypinanie” (przycisk *Bind* — por. str. 309) i późniejsze „odpinanie” (*Unbind* — por. str. 310) od deformatora jest pracochłonne. W dodatku po każdej próbie siatka deformująca pozostaje zmieniona, i trzeba ją ręcznie przywracać do poprzedniego kształtu... Tu potrzeba jakiegoś skryptu *Pythona*, który zautomatyzuje te kroki!

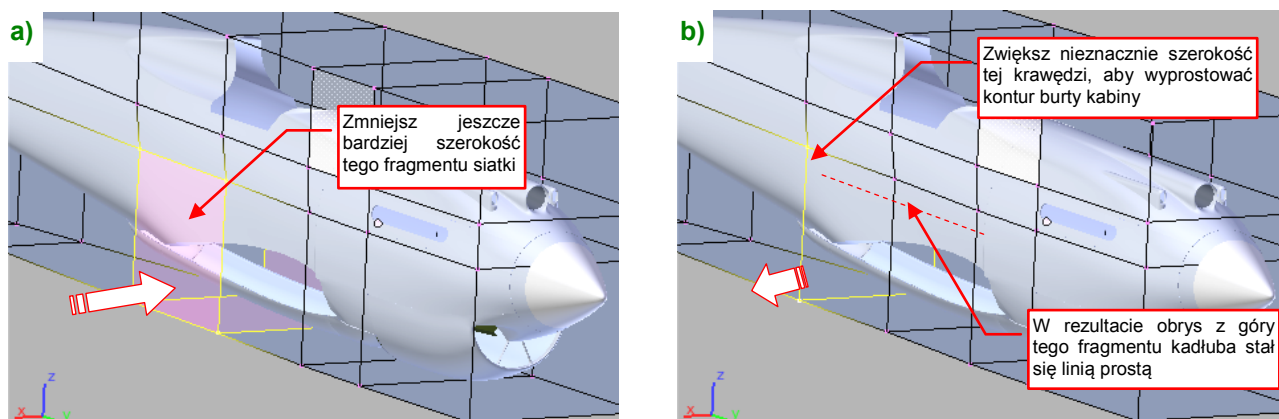
Stworzyłem więc taki dodatek do zarządzania zespołem modyfikatorów *Mesh Deform*. (Chodzi o zespół wielu deformowanych obiektów i jednej siatki deformującej). Nazywa się *MDeform* i znajdziesz go w oknie *User Preferences*, zakładce *Addons*, dziale *Object*. Jego plik jest dołączony do innych skryptów, towarzyszących tej książce. Szczegółowy opis jego użycia umieściłem na str. 312. Tutaj skoncentruję się bardziej na tym, jak przygotować kadłub do tej deformacji.

Zacznijmy od „przymiarki”: przypisz poleceniem **Set Deformer** do siatki **DeformBox** główne powierzchnie kadłuba: część centralną i tylną (**B.300.Fuselage**) oraz okapotowanie silnika (obiekty **B.350.\*.Cowling.\***). Następnie uaktywnij deformator (**Deformer:Bind objects** — por. str. 313) i zacznij stopniowo zmieniać szerokość kadłuba, zaczynając od przodu (Rysunek 2.1.6):



Rysunek 2.1.6 Zmniejszenie szerokości przedniej części kadłuba

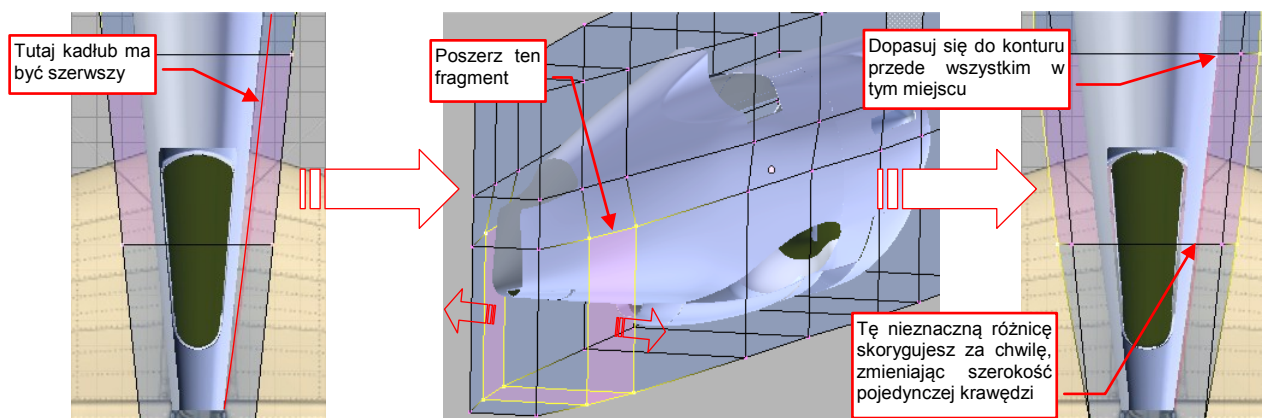
Gdy fragment wokół ściany ogniowej osiągnie już założoną szerokość, ogranicz deformowany obszar siatki do dwóch krawędzi i ponownie zmniejsz jego szerokość (Rysunek 2.1.7a):



Rysunek 2.1.7 Zmniejszenie szerokości burt kabiny

Dopasowujesz w ten sposób do nowego konturu burty kabiny. Na koniec wyeliminuj ostatnie ślady po wygięciu tego fragmentu poszerzając nieco pojedynczą krawędź za kabiną pilota (Rysunek 2.1.7b).

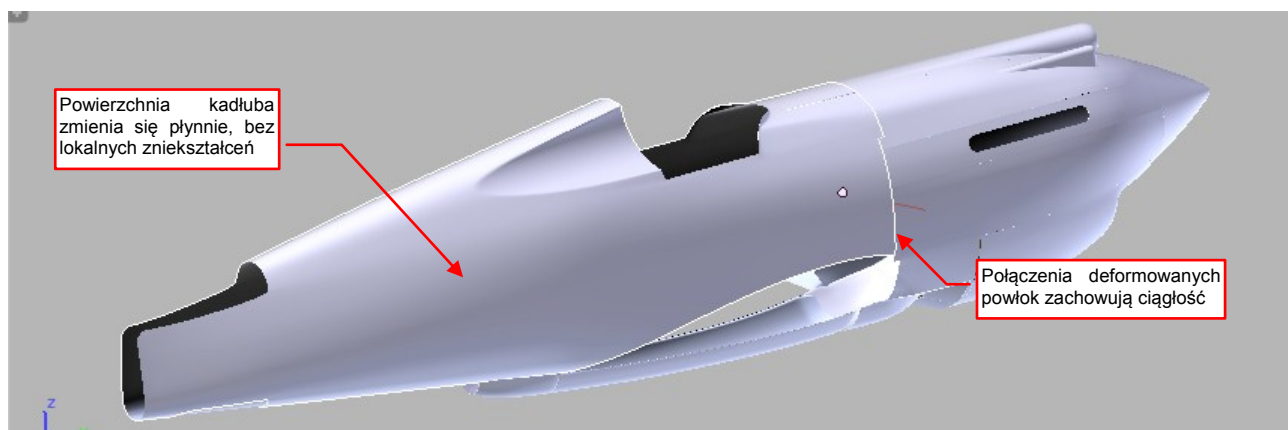
W ten sam sposób zmień szerokość ogona samolotu (Rysunek 2.1.8):



Rysunek 2.1.8 Poszerzanie ogona

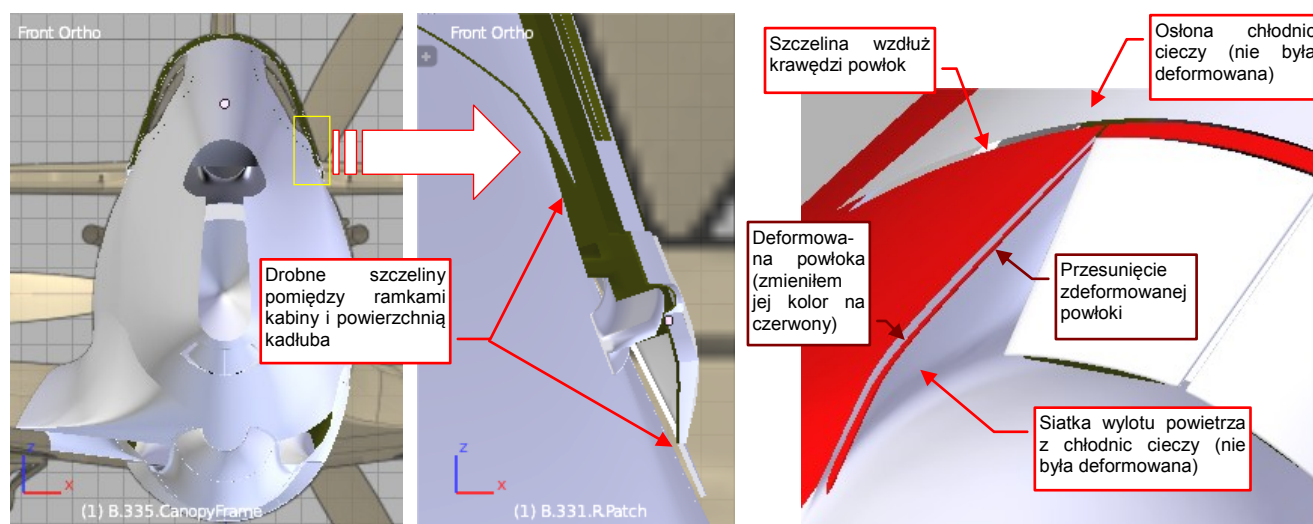


W ten sposób upewniliśmy się, że siatka **DeformBox** ma wystarczającą liczbę krawędzi do właściwego przekształcenia kadłuba. Obejrzyj teraz uważnie rezultat z każdej strony (Rysunek 2.1.9):



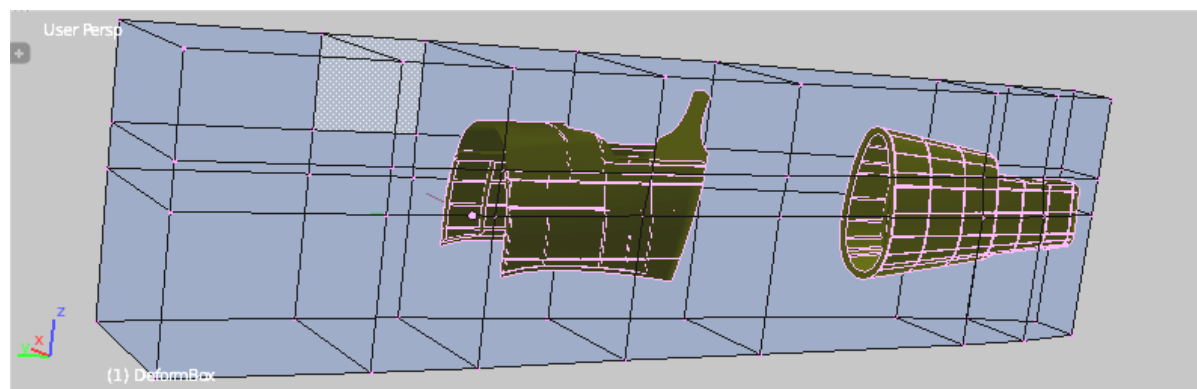
**Rysunek 2.1.9 Weryfikacja rezultatu**

Pierwsze wrażenia są pozytywne: cienie na kadłubie nie wykazują żadnych lokalnych zaburzeń kształtu. Połączenia deformowanych powłok zachowały ciągłość. Koniecznie jednak włącz inne warstwy i sprawdź wszystkie zakamarki — bo na pewno znajdziesz takie szczeliny, jak te które pokazuje Rysunek 2.1.10:



**Rysunek 2.1.10 Szczeliny i przesunięcia pomiędzy odkształconymi i nieodkształconymi obiektami**

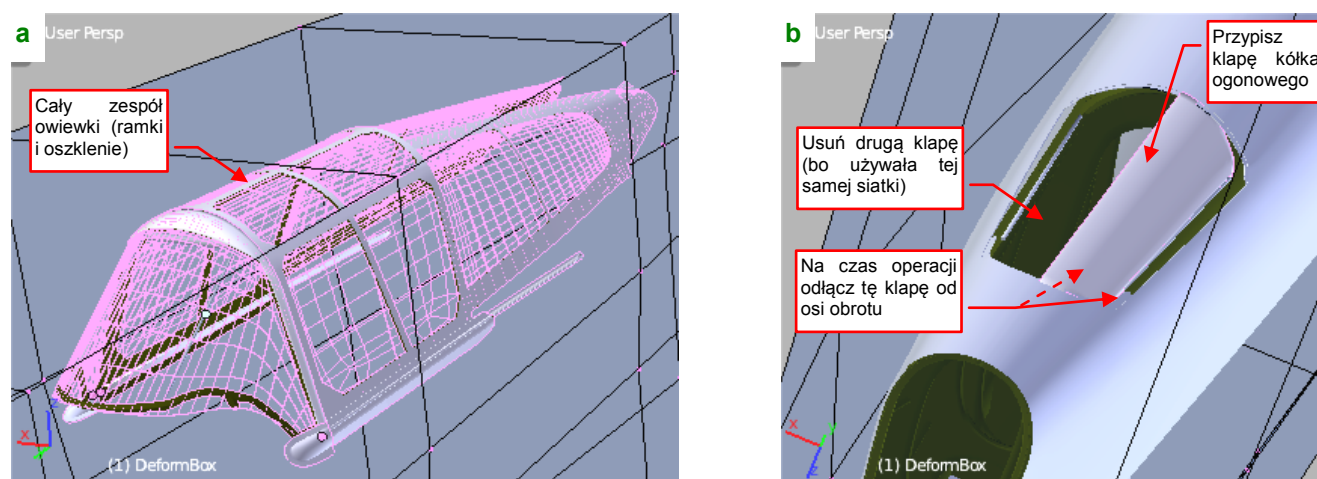
Zapamiętaj albo zanotuj, gdzie znajdują się szczeliny lub przesunięcia. Większość z nich wyeliminujemy, przypisując do deformatora obiekty, wzdłuż których się pojawiły. Zaczniemy od przypisania do **DeformBox** wewnętrznych powierzchni kadłuba, wraz z podłużnicami i wręgami (Rysunek 2.1.11):



**Rysunek 2.1.11 Przypisanie do deformatora wnętrza kadłuba**

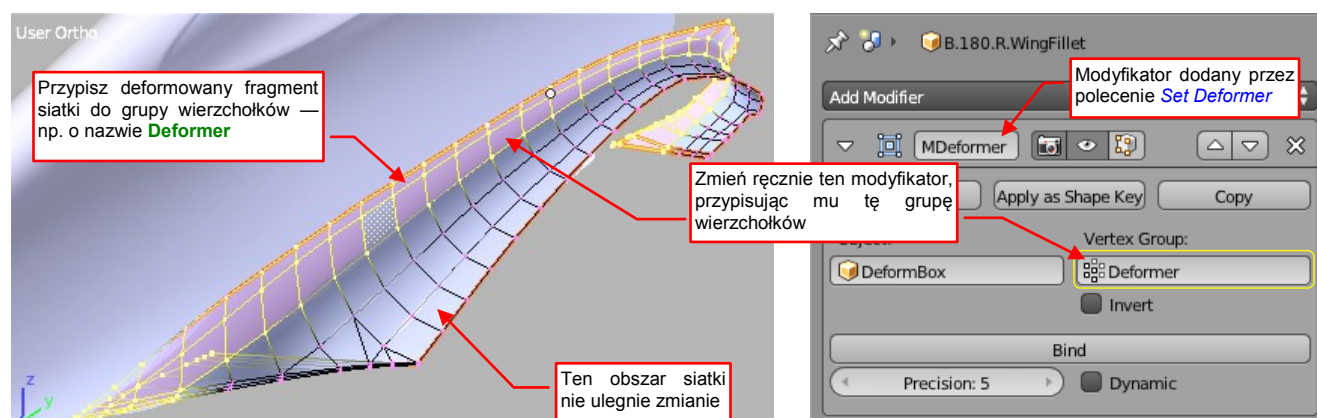
(Nie będę już pokazywać, jak bardzo te obiekty wystają spod zdeformowanej powierzchni kadłuba).

Aby podczas finalnej deformacji kadłuba nie powstały szczeliny wzdłuż ramek kabiny (por. Rysunek 2.1.10), przypisz je do deformatora (Rysunek 2.1.12a):



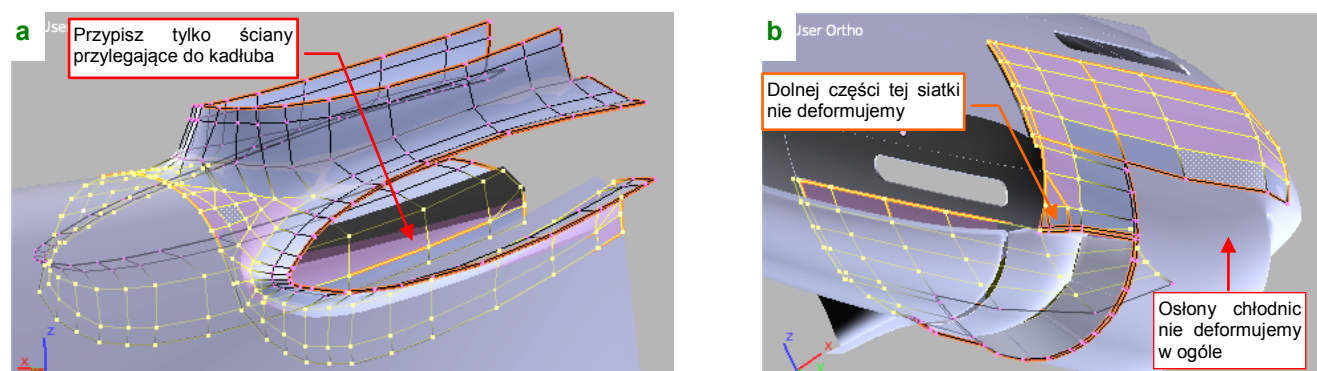
Rysunek 2.1.12 Przepisanie do deformatora dalszych elementów odstających od zmodyfikowanego kadłuba

Przypisz do deformatora jedną z klap kółka ogonowego (Rysunek 2.1.12b). Pamiętaj tylko, aby na czas tej deformacji odłączyć ją od jej osi obrotu (*Clear Parent: Clear and Keep Transformation*). (Te osie dosuniemy do nowych krawędzi otworu po zakończeniu całej operacji). Każda z pokryw kółka ogonowego używa tej samej siatki, więc przed przypisaniem usuń jedną z nich lub skopiuj jej siatkę (inaczej polecenie *Apply* nie zadziała). Aby zmienić tylko część owiewki związaną z kadłubem, zdefiniuj na niej deformowany obszar jako grupę wierzchołków (*Vertex Group* — str. 278) o nazwie **Deform**. Po przypisaniu obiektu do deformatora wpisz ręcznie tę grupę we właściwościach modyfikatora **MDformer** (Rysunek 2.1.13):



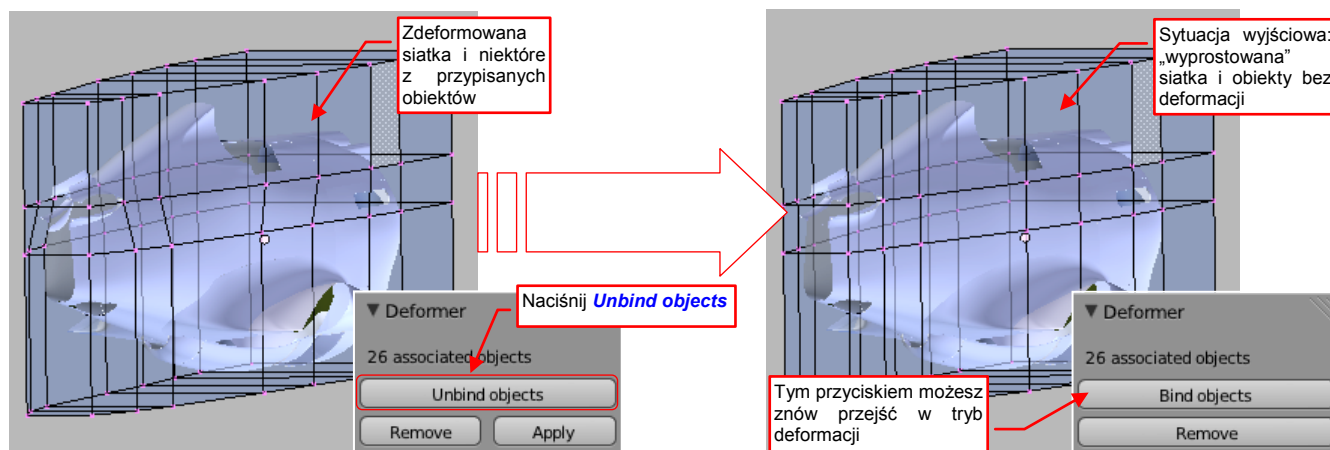
Rysunek 2.1.13 Szczegóły przypisania owiewki skrzydła do deformatora

W podobny sposób ogranicz deformowany obszar na owiewce usterzenia (Rysunek 2.1.14a) i niższej części osłony silnika (Rysunek 2.1.14b):



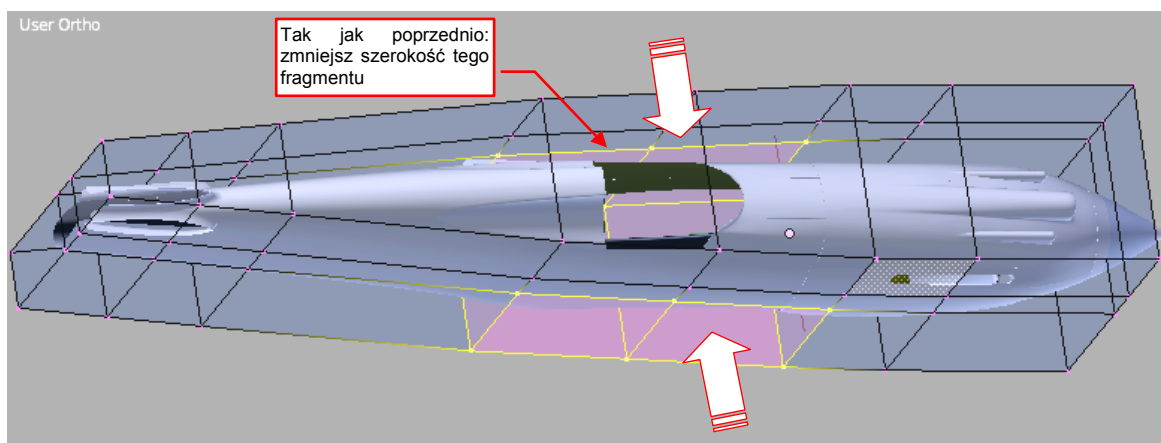
Rysunek 2.1.14 Inne siatki, częściowo przypisane do deformatora

Gdy wszystkie drobne obiekty są już przypisane, wyłącz (*Deformer:Unbind objects*) wprowadzone zmiany w siatce deformatora. W ten sposób wracasz z deformacją do sytuacji początkowej, gdy ani żaden z obiektów, ani siatka deformatora nie były odkształcone (Rysunek 2.1.15):



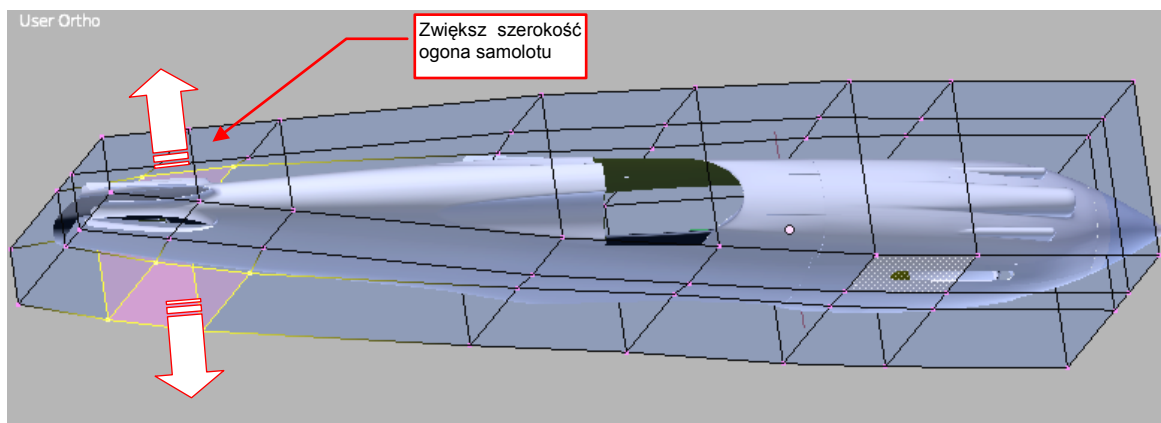
Rysunek 2.1.15 Wyłączenie deformacji siatki

Potem uaktywnij z powrotem deformację, naciskając *Deformer:Bind objects* (tym razem jest więcej obiektów, więc potrwa to jeszcze dłużej). Gdy siatka stanie się aktywna, powtórz przekształcenia opisane na str. 28. Zaczynaj tak jak poprzednio, od przodu (Rysunek 2.1.16):



Rysunek 2.1.16 Deformacja kompletnej siatki kadłuba — część przednia

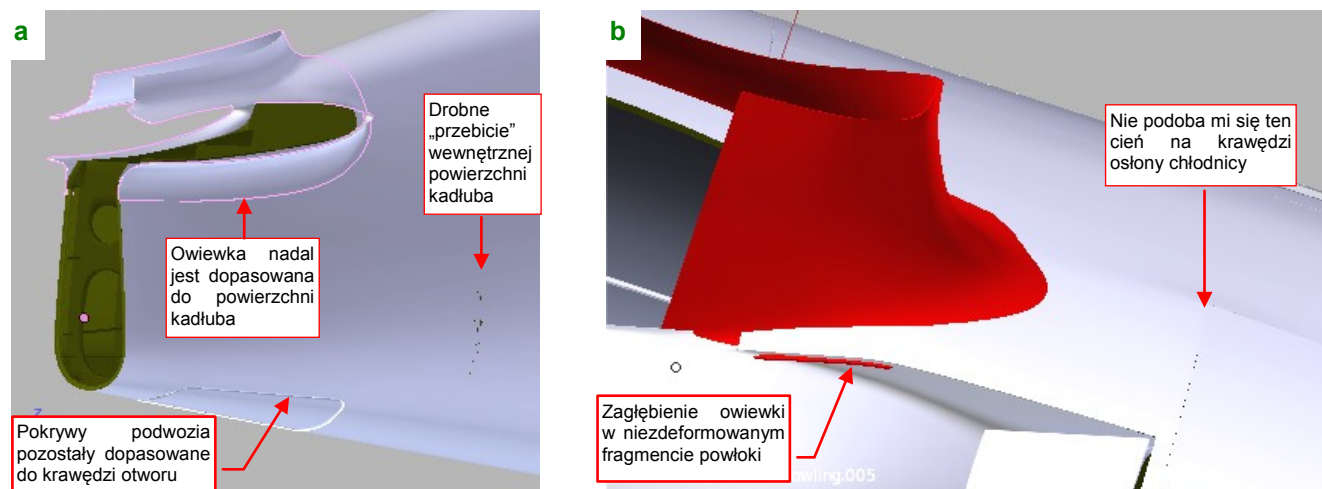
Potem poszerz ogon (Rysunek 2.1.17). (Traktuj te ilustracje tylko jako przypomnienie głównych kroków):



Rysunek 2.1.17 Deformacja kompletnej siatki kadłuba — ogon

Pamiętaj, że część warstw możesz ukryć, a to, co przeszkadza na warstwie aktualnej — także schować (**H**), by nic nie zasłaniało rysunku referencyjnego. Na przykład — owiewkę usterzenia (na ostatnim etapie deformacji).

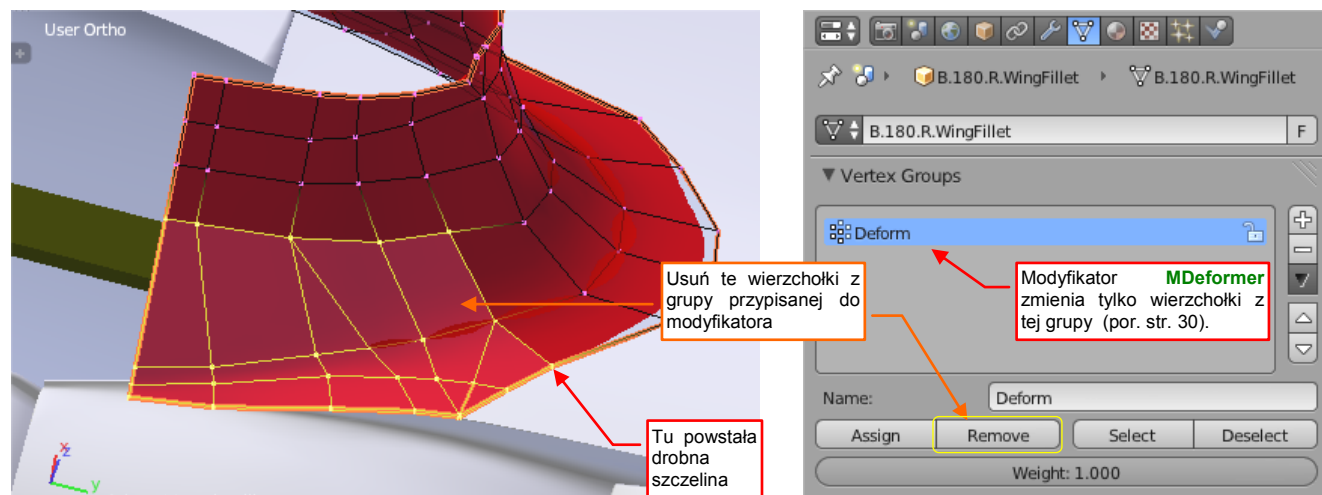
Teraz znów przyjrzyj się uważnie wynikowi deformacji. Rysunek 2.1.18a) pokazuje ogon — wygląda na to że tu, poza drobnym przebicciem krawędzi powierzchni wewnętrznej wszystko jest poprawnie. Za to owiewka skrzydła zagłębiła się nam w niezdeformowaną część kadłuba! (Na chwilę nadałem jej kolor czerwony, aby Rysunek 2.1.18b) lepiej pokazał, co się stało):



Rysunek 2.1.18 Weryfikacja rezultatu

W dodatku nie podoba mi się cień, wskazujący na drobną nieregularność kształtu w okolicy narożnika osłony chłodnicy cieczy (Rysunek 2.1.18b).

Na szczęście do modyfikatora przypisaliśmy tylko część owiewki skrzydła (por. str. 30, Rysunek 2.1.13). Pozwala to na wprowadzenie korekty „na bieżąco”, bez wyłączania całej deformacji. Wystarczy w trybie edycji tej siatki wykluczyć z deformowanej grupy odpowiednie wierzchołki (Rysunek 2.1.19):



Rysunek 2.1.19 Korygowanie obszaru deformacji na owiewce skrzydła

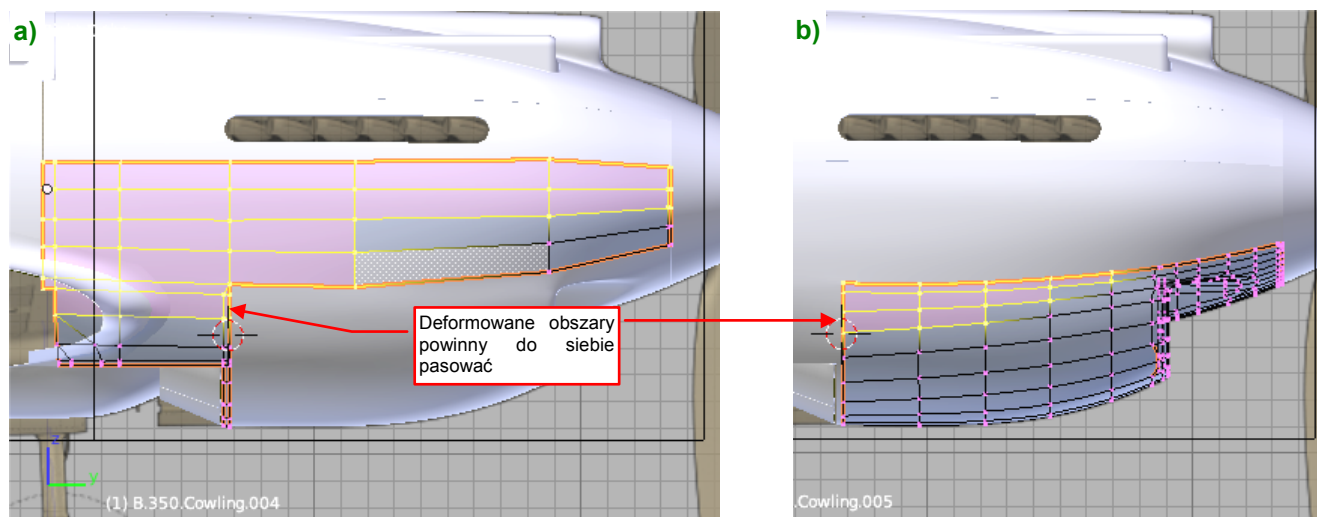
Po zmianie przypisania wystarczy przełączyć się na **Object Mode**, by zmiana odniosła skutek. Gdy do modyfikatora masz przypisaną grupę wierzchołków, możesz także sterować intensywnością efektu (za pomocą kontrolki **Vertex Groups:Weight**). Można nawet przypisywać indywidualne wagi poszczególnym wierzchołkom, „malując” je na siatce (por. str. 291). Pozwala to na płynne sterowanie kształtem. W ten sposób można wyeliminować np. jakieś drobne szczeliny, która pojawiła się po wykluczeniu z deformacji zaznaczonego fragmentu siatki. Tak się jednak złożyło, że tu jeszcze nie musimy ich stosować. Malowanie wag wykorzystamy następnym rozdziale, używając go do sterowania gęstością trawy na powierzchni lotniska.

O wiele gorzej przedstawia się sprawa „cienia” na narożniku chłodnicy cieczy (Rysunek 2.1.18b). Aby go wyeliminować, musimy jednak włączyć do deformacji także ten fragment osłony silnika. A to oznacza, że trzeba wycofać cały obecny efekt (znów wywołując **Deformer:Unbind objects**)!



- Zawsze warto dodać do deformatora więcej siatek niż mniej — przy czym od razu przypisz tym, co do których masz wątpliwości, odpowiednie grupy wierzchołków.

Po wyłączeniu deformatora zmodyfikowałem obszar siatki **B.350.Cowling.004** (Rysunek 2.1.20a), oraz przypisałem narożnik osłony chłodnicy cieczy (**B.350.Cowling.005** — Rysunek 2.1.20b):

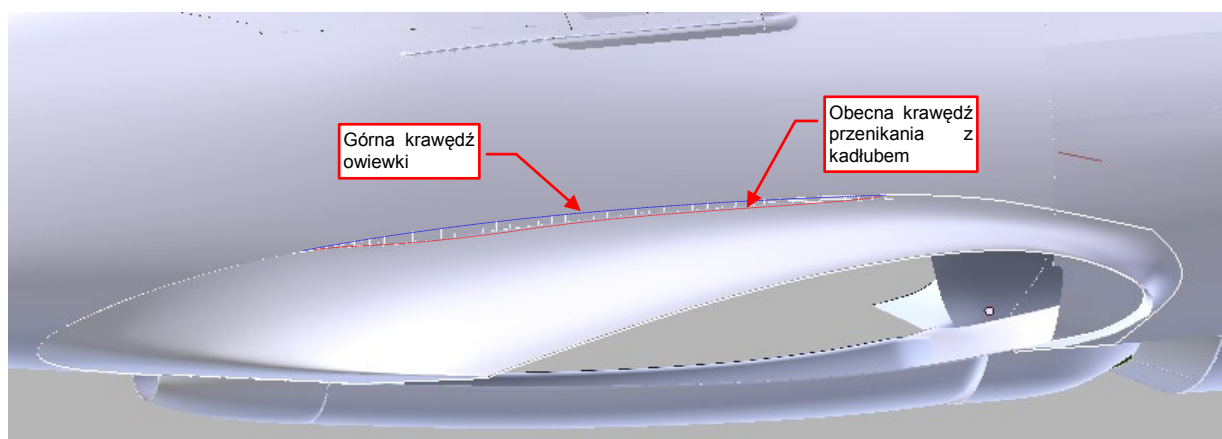


Rysunek 2.1.20 Zmodyfikowane obszary deformacji okapotowania silnika

Potem jeszcze raz wywołaj *Deformer:Bind objects* i poczekaj chwilę (na moim laptopie ta operacja zajmuje Blenderowi około 4 minut). Potem jeszcze raz zdeformuj siatkę — tak jak to robiliśmy poprzednio (por. str. 28 i str. 31).

Za trzecim razem zdeformowane elementy kadłuba pasują do siebie całkiem dobrze. Pozostały do poprawienia:

- „przebiecie” wewnętrznej powłoki kadłuba na ogonie (Rysunek 2.1.18b)
- nieznaczne zagłębienie owiewki skrzydła w burtę kabiny (Rysunek 2.1.21):



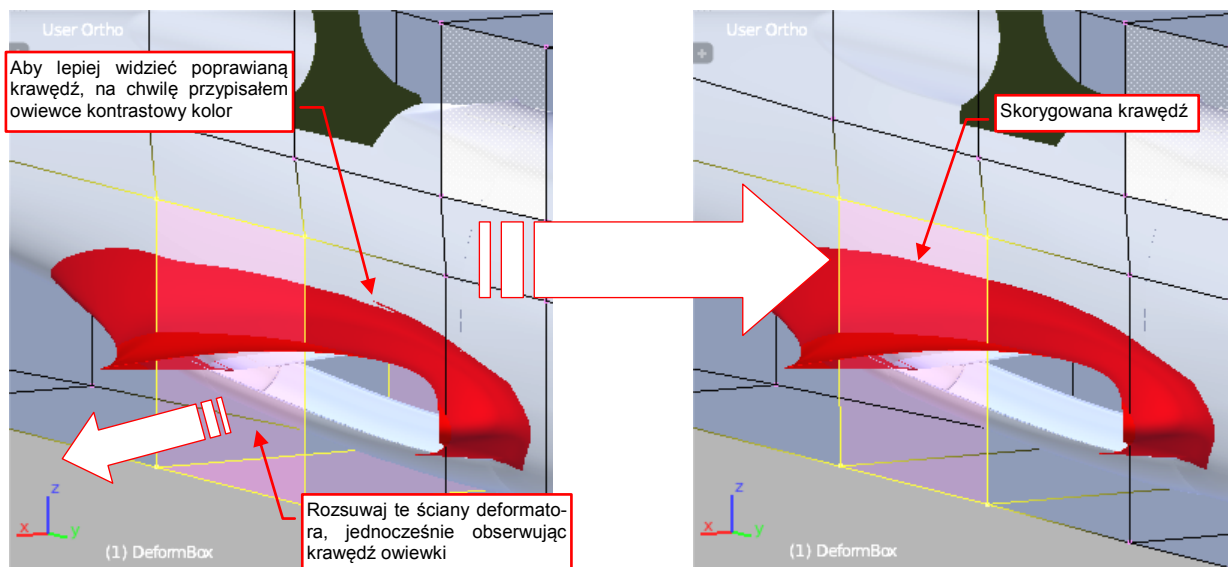
Rysunek 2.1.21 Ostatni defekt — niewielkie zagłębienie owiewki skrzydła

Obydwu przypadków nie można już skorygować żadną sztuczką z deformatorem<sup>1</sup>.

<sup>1</sup> A można by było, gdyby ulepszyć działanie dodatku *MDeformer*. Doświadczenia z formowania tego kadłuba uświadomiły mi, że skrypt mógłby w każdym przypisanym obiekcie tworzyć grupę ze wszystkich wierzchołków siatki i przypisywać ją do dodawanego modyfikatora *MDeformer*. Każda z grup miałaby przypisaną wagę (tzn. wpływ deformowanej siatki) na poziomie 0.5. Siatkę deformującą trzeba byłoby deformować jeszcze bardziej, by uzyskać obecne odkształcenie. Jednak takie rozwiązanie pozwala na regulację „siły deformacji” za pomocą indywidualnych wag wierzchołków. Umożliwia skorygowanie kształtu różnych szczególnych elementów, takich jak krawędź tej owiewki. Postaram się umieścić taką ulepszoną wersję dodatku wśród innych skryptów na [www.samoloty3d.pl](http://www.samoloty3d.pl). W razie czego sprawdź tam w jego opisie, czy wspominam tam o tej nowej funkcjonalności!

W takim razie zatwierdźmy deformację naciskając **Deformer:Apply**<sup>1</sup>. A gdy kadłub jest przekształcony „na stałe”, zajmijmy drobnymi poprawkami, które trzeba nanieść po tej operacji.

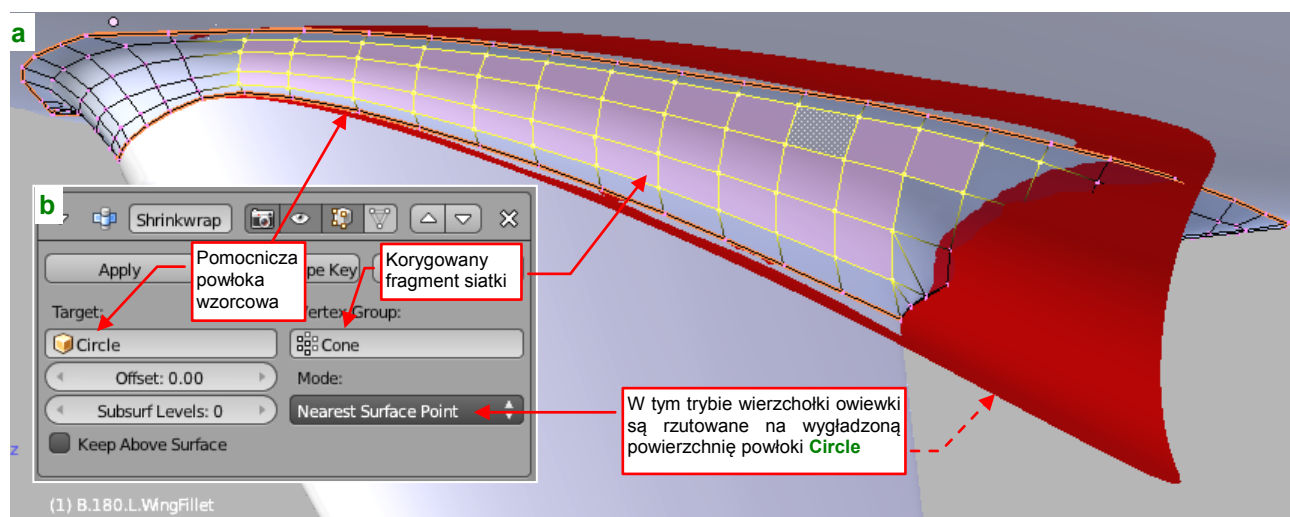
Krawędź obydwu owiewek możemy szybko skorygować. Przypisz je jeszcze raz do deformatora (tą samą grupę wierzchołków — por. str. 30) i lekko rozsuń odpowiednie ściany jego siatki (Rysunek 2.1.22):



Rysunek 2.1.22 Wysunięcie krawędzi owiewki skrzydła spod kadłuba

W podobny sposób dokonaj niewielkiej korekty szerokości elementów wewnątrz ogona (por. str. 32).

Wewnętrzne ściany owiewki skrzydła musimy poddać jeszcze korekcie, bo w wyniku tych wszystkich deformacji ich przekrój poprzeczny nie jest już okrągły. To wymagało trochę pracy, bo trzeba było przygotować nowy wzorcowy „stożek”, dostosowany do zmienionego kształtu narożnika kadłuba i skrzydła (por. Wyciąg II). Potem przeniosłem zaznaczone ściany siatki owiewki na tę nową powłokę (Rysunek 2.1.23a). Aby to zrobić szybko, użyłem modyfikatora **Shrinkwrap** (Rysunek 2.1.23b):



Rysunek 2.1.23 Poprawienie kształtu wewnętrznych ścian owiewki (za pomocą modyfikatora **Shrinkwrap**)

(Szczegółowy opis modyfikatora **Shrinkwrap** znajdziesz na str. 302).

<sup>1</sup> Może się zdarzyć, że za pierwszym razem otrzymasz komunikat, że dla któregoś obiektu to się nie udało (jak na str. 961, Rysunek 8.16.9). Podczas pisania tej sekcji przytrafiło się to i mnie: okazało się że główna siatka kadłuba (**B.300.Fuselage**) jest używana przez więcej niż jeden obiekt. Użyłem okna **Outliner** (w trybie **All Scenes**) by znaleźć ją w pomocniczej kopii kadłuba, którą „chomikowałem” w scenie **Tools**. Stworzyłem jej kopię (**Shift-D**), usunąłem oryginał, i powtórzyłem operację **Deformer:Apply**.



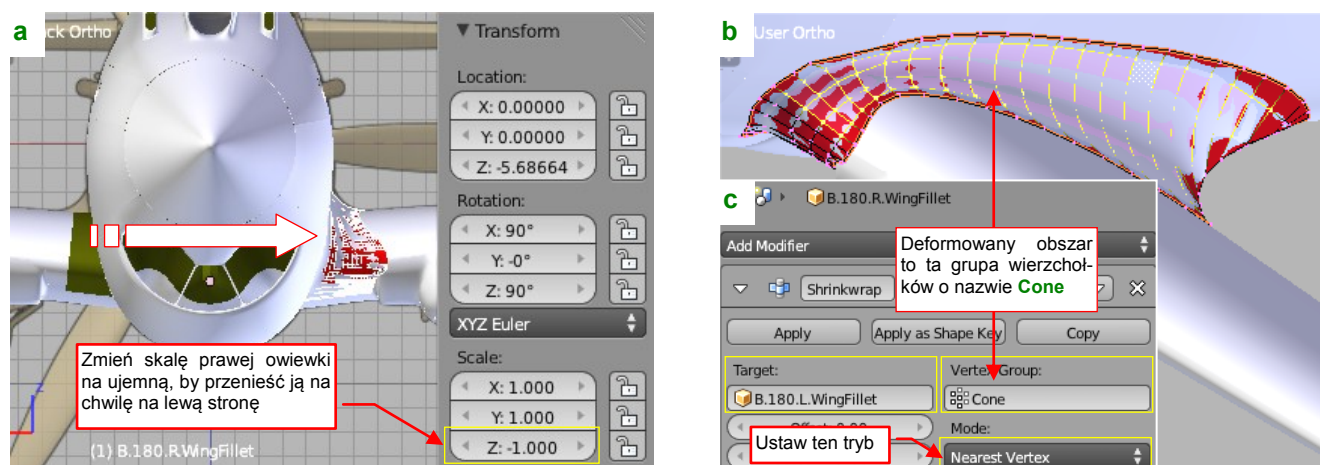
„Utrwal” modyfikator **Shrinkwrap** pokazywany przez Rysunek 2.1.23b) przyciskiem **Apply**. Po tej operacji trzeba jeszcze przesunąć (ręcznie) trzy wierzchołki z przodu siatki, przy krawędzi natarcia skrzydła (Rysunek 2.1.24). (W tym obszarze przekrój owiewki już nie był okrągły, i dlatego wzorcowa powłoka nie sięga do tego miejsca).



Rysunek 2.1.24 Wierzchołki do ręcznej korekty

Uzyskaliśmy w ten sposób poprawny kształt lewej owiewki. Teraz jednak przed nami pewien problem: jak te wszystkie zmiany szybko i dokładnie przenieść na prawą owiewkę? Proponuję także do tego użyć modyfikator **Shrinkwrap**. Zrób to w następujący sposób:

- wyłącz (na czas tej operacji) w lewej „wzorcowej” owiewce (**B.180.L.WingFillet**) modyfikator **Subdivision Surface**;
- zmień skalę prawej owiewki (**B.180.R.WingFillet**) wzdłuż jej lokalnej osi **Z** na ujemną. To spowoduje „nałożenie” tej powłoki na lewą owiewkę (Rysunek 2.1.25a);
- zdefiniuj na prawej owiewce grupę wierzchołków, których współrzędne chcesz uzgodnić (nazwałem ją **Cone** — por. Rysunek 2.1.25b)
- dodaj do obiektu **B.180.R.Wing.Fillet** modyfikator **Shrinkwrap** (na początek stosu modyfikatorów!) i przypisz go tak jak pokazuje to Rysunek 2.1.25c). Koniecznie ustaw tryb na **Nearest Vertex**!

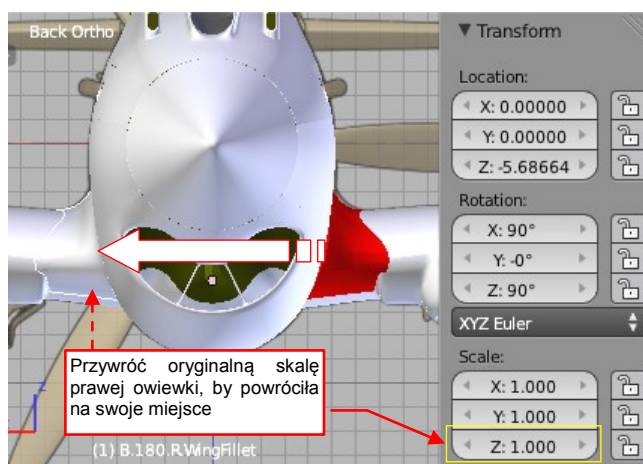


Rysunek 2.1.25 Kopiowanie zmian w kształcie siatki za pomocą modyfikatora **Shrinkwrap**

Abyś mógł odróżnić każdą z tych powłok, na ilustracjach lewa owiewka ma kolor czerwony. Gdybym pozostawił w niej modyfikator **Subsurf**, to wierzchołki prawej owiewki dopasowałyby się do wygładzonych wierzchołków lewej.

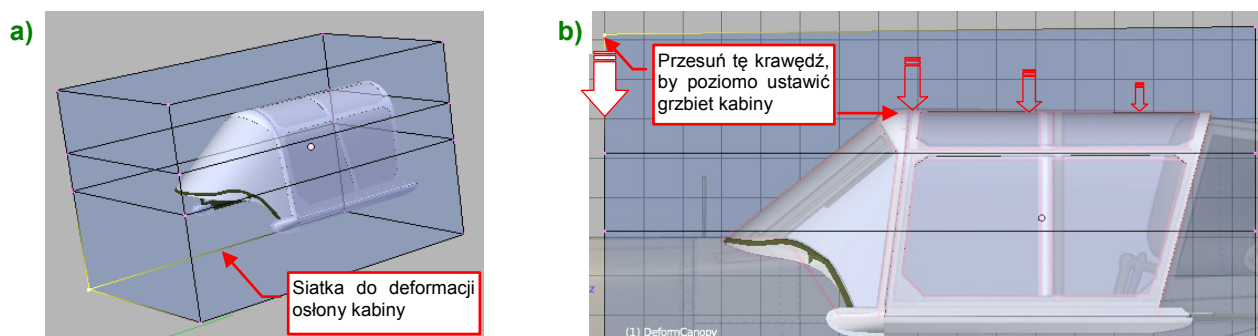
Gdy wszystko wygląda poprawnie, zatwierdź zmianę prawej owiewki, naciskając przycisk **Apply** w panelu jej modyfikatora **Shrinkwrap**. Potem przywróć oryginalną skalę tego obiektu, by wrócił na swoje miejsce (Rysunek 2.1.26).

Na koniec dodaj z powrotem do lewej owiewki modyfikator **Subsurf**, by była wygładzona tak jak prawa.



Rysunek 2.1.26 Przywrócenie prawej owiewce oryginalnej skali

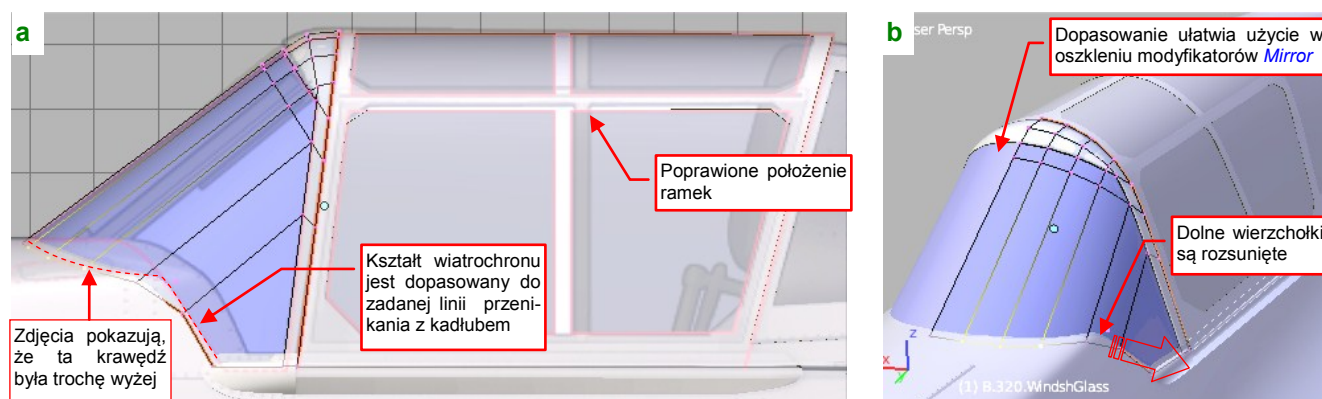
Korektę owiewki kabiny zaczynam od obniżenia jej grzbietu (do linii poziomej) . To także można szybko wykonać za pomocą odpowiedniego „pudełka” i dodatku **MDeform** (Rysunek 2.1.27a):



Rysunek 2.1.27 Obniżenie grzbietu owiewki za pomocą dodatku **MDeform**

Korektę wprowadzamy przesuwaną górną przednią krawędź siatki do dołu (Rysunek 2.1.27b). (Poziome linie na „pudełku” deformatora zapobiegają deformacji niższych partii osłony kabiny).

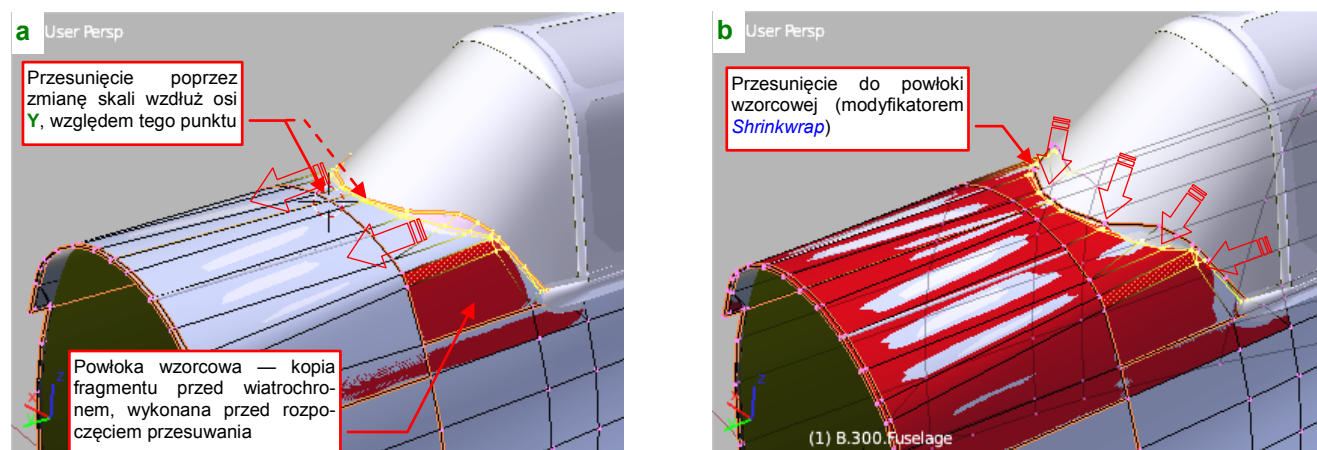
Więcej pracy będziemy mieli ze zmianą wiatrochronu. Zmieńmy najpierw kształt tej siatki, dopasowując ją do zadanej krawędzi przenikania (Rysunek 2.1.28a):



Rysunek 2.1.28 Dalsze korekty wiatrochronu i owiewki

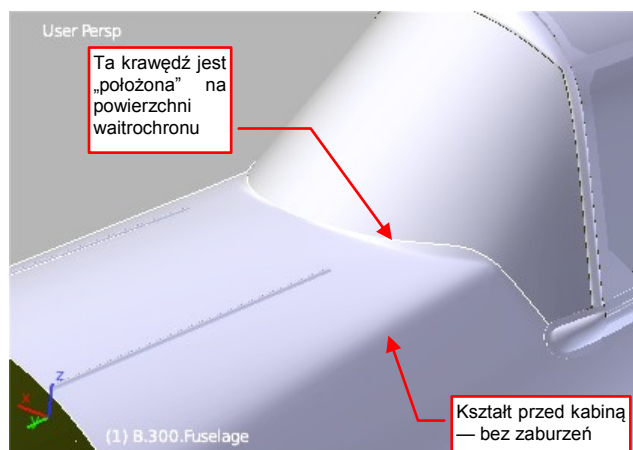
Przednia krawędź wiatrochronu jest niemal zgodna, więc pozostaje poszerzyć podstawę tej powłoki (zmiana skali wzdłuż osi **X**). Na czas tej operacji usunąłem tylne ściany tej siatki, bo ich rozwinięcie UV nie było wykorzystywane w materiale **B.Canopy.Glass**. Dodamy je później, chociażby modyfikatorem **Solidify**.

Potem trzeba przesunąć krawędź kadłuba, otaczającą wiatrochron. Aby nie zdeformować przypadkiem tej części modelu, skopiowałem ją najpierw jako wzorec. Potem przesunąłem odpowiednie wierzchołki (zmianą skali wzdłuż osi **Y** — Rysunek 2.1.29a), i „położyłem” je na wzorcu (**Shrinkwrap** — Rysunek 2.1.29b):

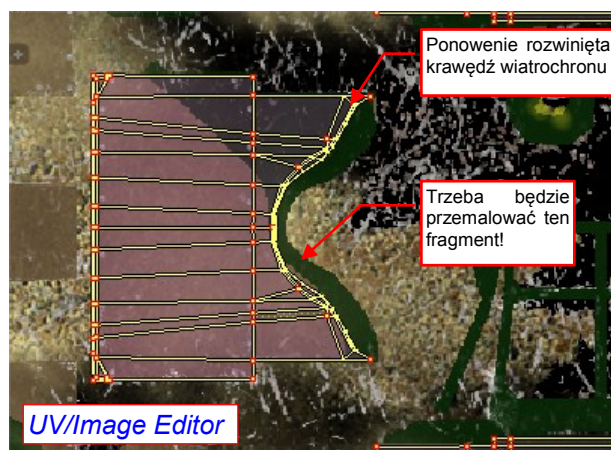


Rysunek 2.1.29 Przesunięcie krawędzi poszycia kadłuba

Formowanie krawędzi kadłuba wokół wiatrochronu zawsze zajmuje trochę czasu, bo to dość złożony fragment, który musi być dobrze przylegać do oszklenia. Posługując się przedstawionymi metodami udało mi się zachować oryginalny kształt kadłuba, bez żadnych nieregularności (Rysunek 2.1.30):



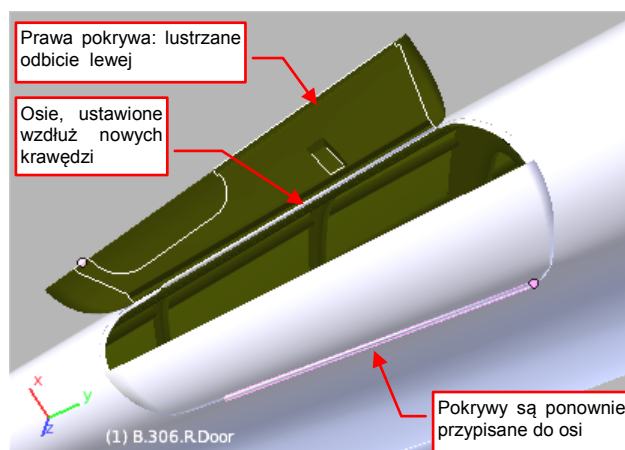
Rysunek 2.1.30 Przesunięta krawędź kadłuba



Rysunek 2.1.31 Skorygowane rozwinięcie UV tego fragmentu

Po tak poważnej zmianie kształtu trzeba także uaktualnić rozwinięcie tej siatki w [UV/Image Editor](#) (Rysunek 2.1.31). To na szczęście dość proste: wystarczy „odpiąć” (**Alt-P**, [Unpin](#)) odpowiednie wierzchołki, i wywołać [UVs → Unwrap](#) (**E**). Blender ułoży ten fragment sam, odpowiednio do nowych proporcji. Pozostanie potem przenieść to rozwinięcie do pliku [uv.svg](#) i uaktualnić poszczególne rysunki tekstur ([skin.svg](#), [color.xcf](#), i pochodne). W przeciwnym razie nie będą pasować do siebie plamy kamuflażu!

Jeżeli chodzi o modelowanie, to na koniec można uzupełnić usuniętą (por. str. 30) symetryczną prawą osłonę kółka ogonowego. Potem przesunąć ich osie obrotu ponad nowe krawędzie otworu w kadłubie, i powiązać z powrotem odpowiednimi relacjami [Parent](#) (Rysunek 2.1.32):



Rysunek 2.1.32 Uzupełnione pokrywy kółka ogonowego

### Podsumowanie

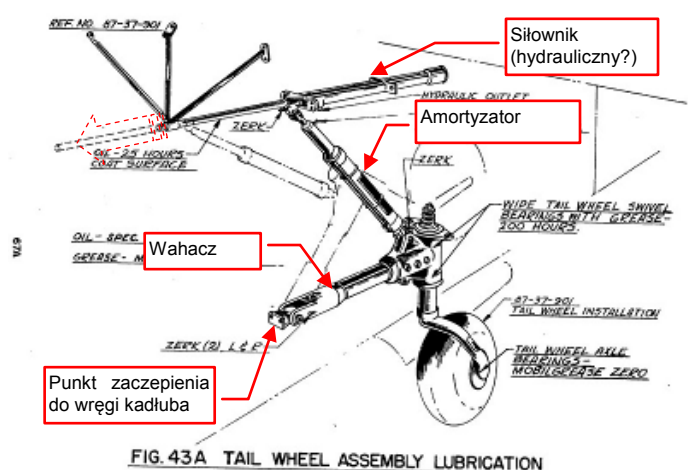
- Postęp prac nad modelem redukcyjnym przypomina bardziej „spiralę”, w której co jakiś czas powracasz do tego, co zrobiłeś, aby to poprawić. Tylko w wyniku takiego ciągłego doskonalenia można uzyskać naprawdę dokładny model. Ta sekcja opisuje jeden z takich „nawrotów” — wprowadzenia poważnej zmiany kształtu kadłuba (str. 25);
- Do zmiany kształtu jednego lub wielu złożonych obiektów można wykorzystać deformację za pomocą pomocniczej siatki. Umożliwia to dodatek [MDeform](#) (str. 27);
- Aby dopasować jedną powłokę do drugiej (str. 34, 36), lub skopiować zmiany wykonane w jednej siatce do jej symetrycznej kopii (str. 35) możesz wykorzystać modyfikator [Shrinkwrap](#);
- Aby nie naruszyć mapowań [UV](#) tekstur, żadna z opisanych w tej sekcji modyfikacji nie spowodowała usunięcia (ani stworzenia) jakiegokolwiek ściany w siatkach modelu.



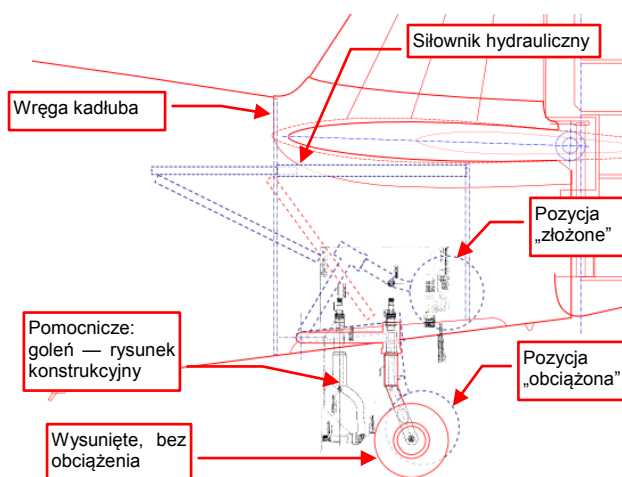
## 2.2 Kółko ogonowe

Szczegóły samolotu zaczniemy od kółka ogonowego. Wybrałem je na początek tego rozdziału, gdyż nie jest zbyt skomplikowane. Jednocześnie wykonanie tego elementu pozwala pokazać kilka technik, które zastosujemy także w innych miejscach modelu.

Nim zabierzesz się za modelowanie podwozia, postaraj się zgromadzić jak najwięcej zdjęć tego elementu. Czasami są to dość skomplikowane mechanizmy! Na szczęście kółko ogonowe P-40 do takich nie należy<sup>1</sup>. Wśród dokumentacji, opublikowanej w Internecie, można znaleźć całkiem przejrzysty rysunek tego zespołu (Rysunek 2.2.1 — w istocie to schemat okresowego smarowania). Dodatkowo znalazłem tam także rysunek fabryczny goleni. Przygotowałem na tej podstawie „szkic wykonawczy” kółka ogonowego, wkomponowany w plany samolotu (Rysunek 2.2.2, patrz także [source\plans\p40b-highres\Left \(gear\).png](#)):

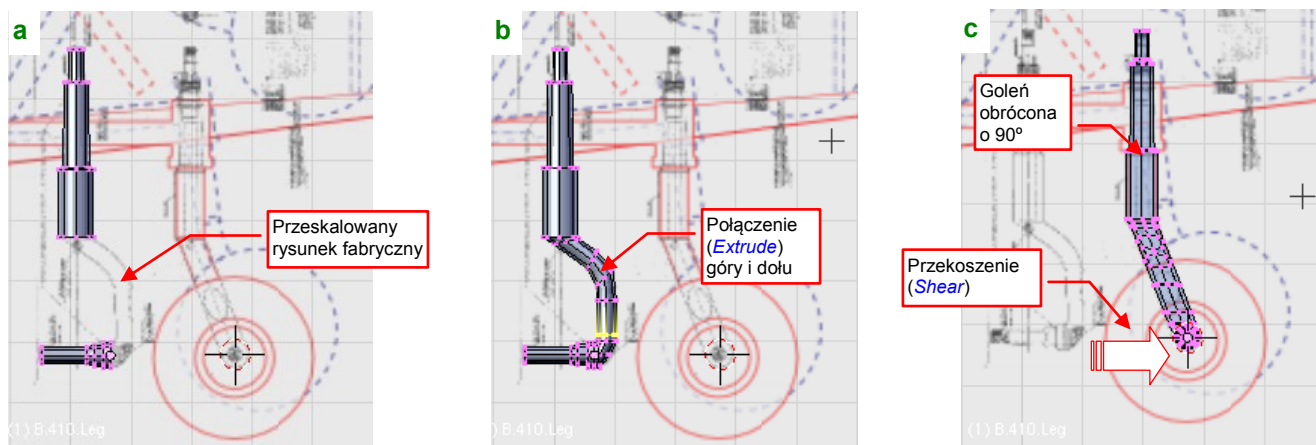


Rysunek 2.2.1 Oryginalny schemat zespołu kółka ogonowego



Rysunek 2.2.2 Rysunki wykonawcze dla modelu

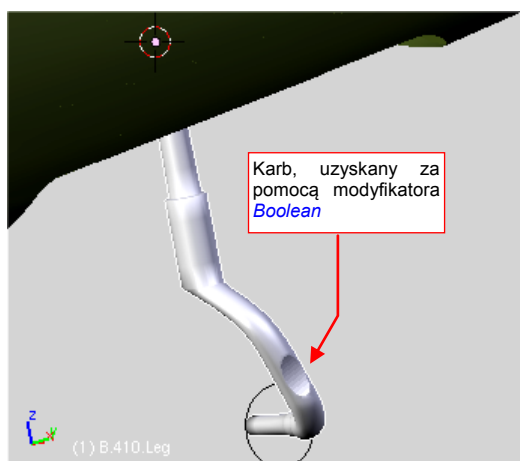
Cały zespół zaczniemy od goleni. W oparciu o rysunek fabryczny, zmniejszony do odpowiedniej skali, uformowałem najpierw dwa elementy cylindryczne: oś koła i górną część goleni (Rysunek 2.2.3a):



Rysunek 2.2.3 Formowanie goleni kółka ogonowego

Połączyłem te dwie „rury” wytłoczeniem, wygiętym jak na rysunku fabrycznym (Rysunek 2.2.3b). Następnie odwróciłem całą goleń o 90°, w położenie docelowe, i przesunąłem (przekoszeniem — *Shear*) oś koła w położenie zgodne z rzutem z boku (Rysunek 2.2.3c).

<sup>1</sup> Warto zauważyć, że w odróżnieniu od wielu innych konstrukcji z tego okresu (np. Bf-109G, Ła-5) mechanizm chowania kółka ogonowego w P-40 nie sprawiał problemów podczas eksploatacji z lotnisk polowych. Nie zauważyłem nigdzie przypadków blokowania tego kółka przez warsztaty w pozycji „wysunięte” — bardzo częstych sytuacji w myśliwcach Luftwaffe i WWS.



Rysunek 2.2.4 Uformowana goleń (B.510.Leg)

rzucić do rezultatu dodatkowe ściany. Trzeba mu wtedy takie „twórcze podejście” wyperswadować, dodając do siatki dodatkowe krawędzie. (Chodzi o to, by zmienić jej topologię, uzyskując inny przebieg linii wzajemnego przenikania powłok).

Od razu przypiszmy wykonany element do odpowiedniego materiału. Na zachowanych kolorowych zdjęciach z lat 40-tych XX w. podwozie P-40 ma wyraźny, metaliczny połysk. Nie wygląda to na farbę, tylko na inny rodzaj powłoki (np. ocynk<sup>1</sup>). Takie pokrycie miałyby w tym miejscu sens. W końcu podwozie jest bardziej narażone na korozję, niż reszta samolotu.

Stworzyłem odpowiedni materiał, który będzie wyglądał jak lekko chropowata stal, pokryta jakąś warstwą zabezpieczającą w kolorze metalu. Szczegóły, jak to zrobić, znajdziesz na str. 396. Materiałowi stali nadałem nazwę **B.Steel.Rough**, i przypisałem do siatki **B.510.Leg**. Rysunek 2.2.5 pokazuje, jak wygląda goleń podwozia na próbnym renderze.



Rysunek 2.2.5 Materiał B.Steel.Rough.

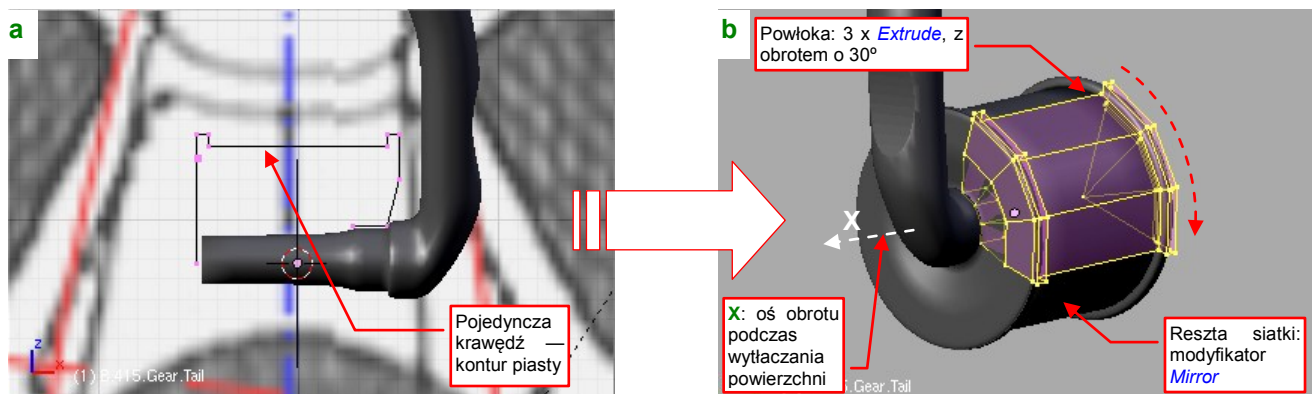
- Dla wszystkich drobnych części samolotu, za wyjątkiem paru elementów kabiny pilota, nie stosuj pracochłonnych rozwinięć w UV i „ręcznie malowanych” tekstur. Na typowych ujęciach i tak będą za małe, by takie detale można było dostrzec.

Zamiast tego, wykorzystaj do pokrycia drobnych części takie materiały jak **B.Steel.Rough**. Użyto w nich co najwyżej tekstur proceduralnych, nałożonych inną metodą (innymi współrzędnymi tekstury) niż pracochłonne UV (np. **Generated** lub **Object**).

Oczywiście, jeżeli planujesz robić jakieś ujęcie podwozia z bliska, to wtedy możesz się postarać. Trzeba na każdą siatkę nanieść szwy, rozwinąć, sprawdzić „jak leży” testowy obraz na poszczególnych detalach. Potem na obrazie tekstury namalować wszystkie zabrudzenia, plamy oleju, zacieki... Licz się jednak z tym, że zajmie Ci to dzień lub dwa. To poniekąd powtórzenie, tylko w trochę mniejszej skali, tego, co opisywał Wyciąg III. Mówiąc krótko: masa dodatkowej roboty!

<sup>1</sup> Mam na myśli pokrycie elementów stalowych warstwą cynku. To popularna, niezbyt droga technologia, zapewniająca lepsze zabezpieczenie przed korozją niż pomalowanie. Możesz ją dostrzec na co dzień: prawie wszystkie kolanka, złączki, a także rury instalacji gazowych wykonuje się ze stali ocynkowanej.

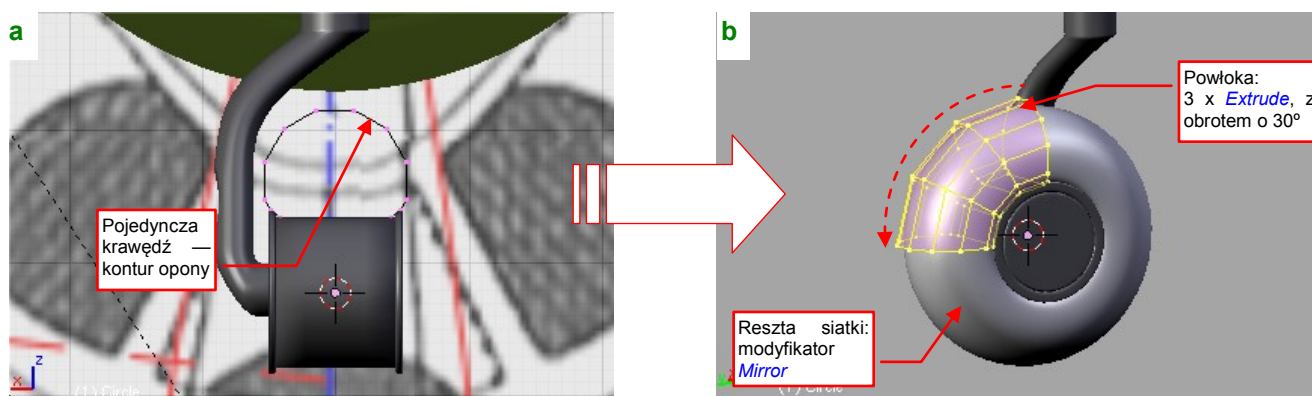
Piastę koła najszybciej jest uformować, zaczynając od jej przekroju poprzecznego (Rysunek 2.2.6a):



**Rysunek 2.2.6 Formowanie piasty kółka ogonowego**

Potem wystarczy wytłoczyć ten kontur poprzez kolejne obroty o 30° wokół lokalnej osi **X** obiektu (Rysunek 2.2.6b). Na koniec dodałem do siatki modyfikator **Mirror**, wzdłuż osi **Y** i **Z**. Piastę nazwałem **B.515.Gear.Tail** i przypisałem jej ten sam materiał, co goleni (**B.Steel.Rough**).

Tak samo uformowałem oponę (**B.520.Tire.Tail**). Zacząłem od linii konturu (Rysunek 2.2.7a), a potem wytłoczyłem ją w „ćwiartkę” koła (Rysunek 2.2.7b):



**Rysunek 2.2.7 Formowanie opony kółka ogonowego**

Oponie kółka ogonowego przypisałem właściwości „zakurzonej gumy” — materiał **B.Rubber.Tire**. Gdybyś miał wątpliwości, jak powinna wyglądać, po prostu przyjrzyj się uważnie kołom jakiegoś niezbyt czystego samochodu (lub takiego, który przejechał jakąś długą trasę).

Rysunek 2.2.8 przedstawia próbny rendering tego zespołu. Muszę przyznać, że uzyskanie właściwego efektu zużycia koła nie było łatwe. Wymagało wykorzystania wielu tekstur proceduralnych. Zacząłem od maski, która osłabia wszelkie tekstury poza obszarem bieżnika. (Bieżnik jest zawsze bardziej wytarty, dlatego efekty chropowatości i zabrudzeń występują na nim w pełnej skali). Następnie naniosłem wokół opony wzór bieżnika (podłużne paski). Potem ustaliłem bazowe własności materiału opony, takie jak barwa (tylko zupełnie nowa guma jest czarna!), szorstkość, i połyskliwość. Na koniec za pomocą tekstur proceduralnych naniosłem efekty chropowatości i nieznacznego zakurzenia. Dokładny opis, jak przygotować taki materiał dla opony, znajdziesz na str. 402.



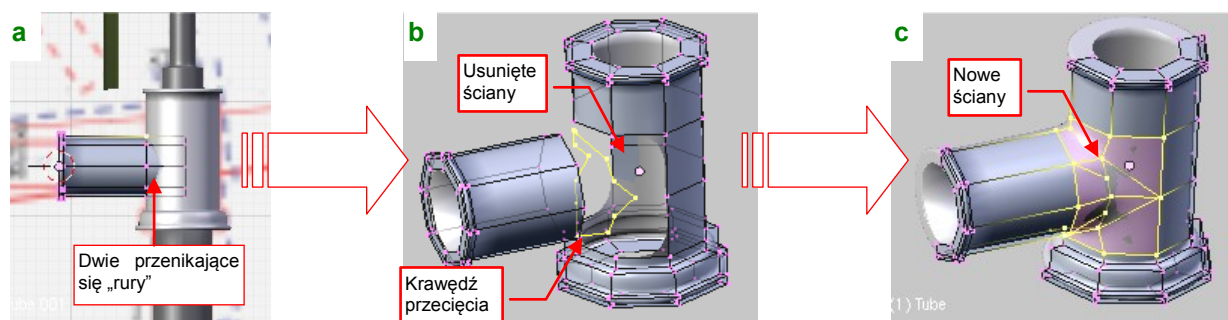
**Rysunek 2.2.8 Próbny rendering z materiałem **B.Rubber.Tire****



Mówiąc szczerze, zawsze zaczynam modelowanie takich mechanizmów, jak podwozie, od odręcznego szkicu. To taka moja metoda na dogłębne poznanie rzeczy, którą mam odtworzyć. Konieczność narysowania poszczególnych detali (nawet niestarannie!) zmusza mnie do zastanowienia się nad ich kształtem i rolą. W czasie szkicowania decyduję, na jakie części podzielę cały zespół, oraz jakie szczegóły pominę. Po zakończeniu pracy taki szkic jest już niepotrzebny i ląduje w koszu. Przypuszczam, że możesz wypracować na własny użytek inne metody „rozumienia” modelowanego zespołu. Najważniejsze w każdym razie, by najpierw ułożyć sobie w głowie plan pracy, a potem go zrealizować. Nie zaczynaj takich rzeczy impulsywnie, nie kieruj się naszym tradycyjnym (narodowym?) podejściem „jakoś to będzie” — bo zmarnujesz niepotrzebnie swój czas.

Po bliższej analizie stwierdzisz zapewne, że nawet najprostsze podwozie samolotu składa się z kilkunastu części, o różnym kształcie. Oznacza to, że trzeba tworzyć je co najmniej w tempie kilku dniennie, aby całość prac nie rozciągnęła się w tygodnie. Nie zawsze jest to takie łatwe. Muszę przyznać, że wymyślenie w miarę prostej siatki dla każdego niebanalnego detalu zajmuje mi trochę czasu. Na tej i następnych stronach pokażę formowanie szczegółów dwóch części: wahacza i amortyzatora. W ten sposób będziesz mógł skorzystać z moich doświadczeń, i podpatrzeć typowe metody tworzenia takich detali. Resztę elementów zespołu kółka ogonowego omówię bardzo pobieżnie — po jednej ilustracji na każdy obiekt.

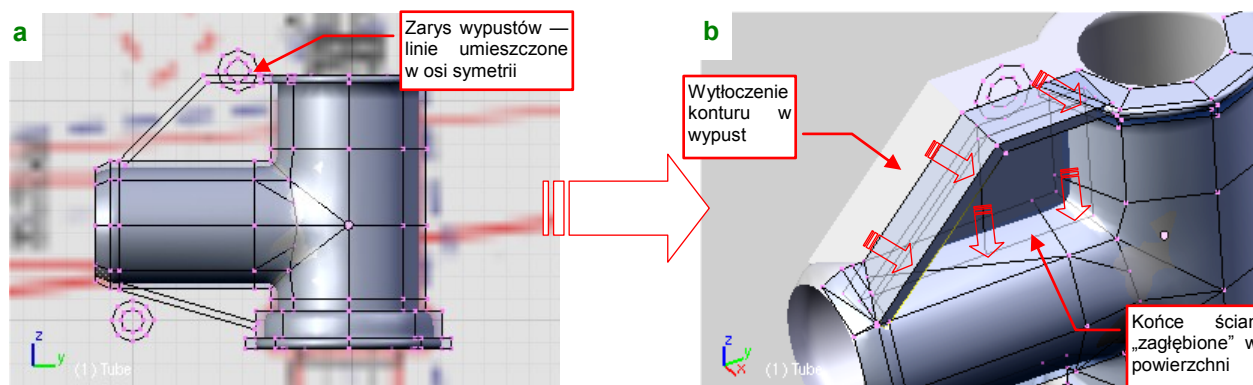
Zacznijmy od końcówki wahacza, w której ma być osadzona goleń podwozia (Rysunek 2.2.9):



Rysunek 2.2.9 Początek formowania wahacza — miejsce na goleń podwozia

Podstawowy kształt końcówki to dwie rury, połączone w poziome „T” (Rysunek 2.2.9a). Następnie wyznaczam ich krawędź przecięcia (za pomocą dodatku *Intersection* — por. str. 221). Po wyznaczeniu tej krawędzi usunąłem ściany otaczające linię przenikania powłok (Rysunek 2.2.9b). Utworzyłem w ich miejsce nowy układ ścian, włączając w siatkę krawędź przecięcia (Rysunek 2.2.9c). Przy okazji usunąłem jedną z połówek siatki i włączyłem modyfikator *Mirror*.

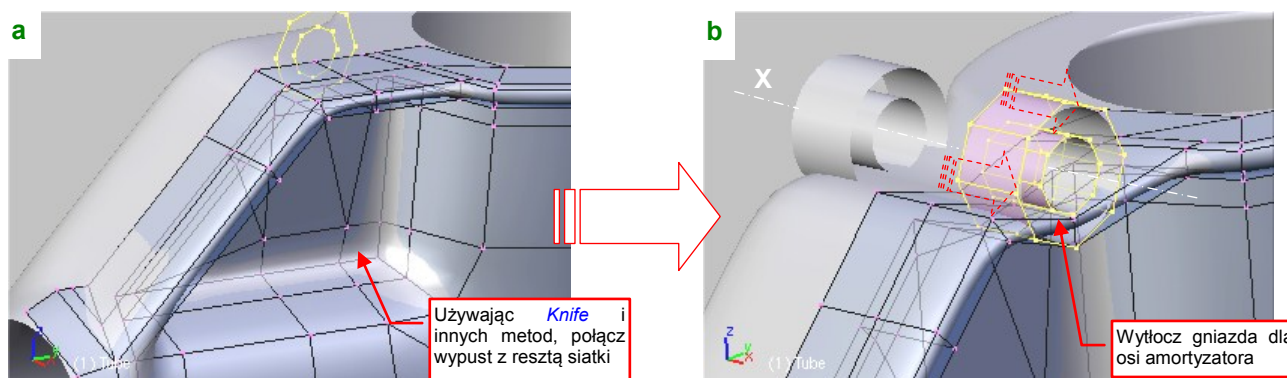
Kolejnym detalem tej siatki są wypusty, umieszczone u góry i dołu końcówki. Zacząłem ich formowanie od zarysu w rzucie z boku. Do odwzorowania okręgów użyłem ośmioboków (Rysunek 2.2.10a):



Rysunek 2.2.10 Wahacz — początek formowania wypustu

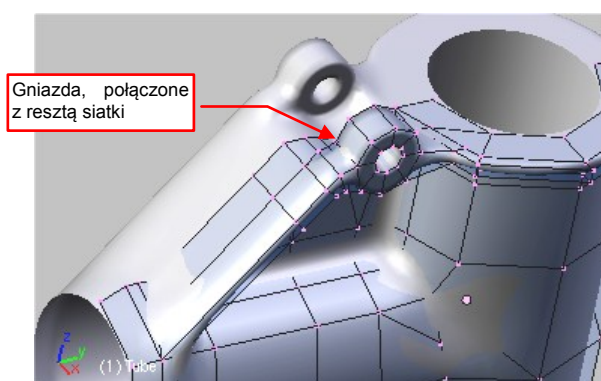
Potem wytłoczyłem kontury wypustów na boki i do dołu, by uformować ich zasadniczy kształt (Rysunek 2.2.10b). Zrobiłem to tak, by wytłoczone ściany zagłębiły się wszędzie w powierzchni „rur” końcówki.

Kolejnym krokiem jest scalenie siatki „nur” i wypustu. Zrobiłem to, tworząc nacięcia *Knife*, a potem zmieniając układ niektórych krawędzi i ścian (Rysunek 2.2.11a):

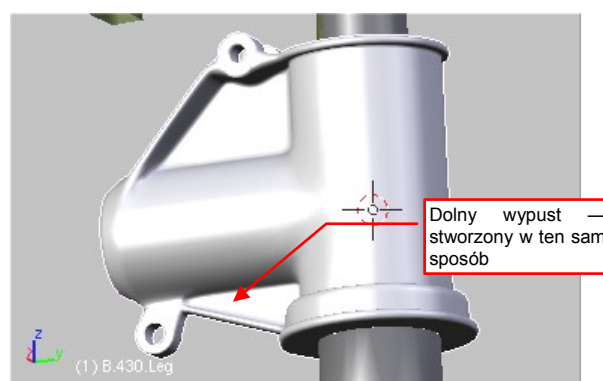


Rysunek 2.2.11 Wahacz — formowanie wypustu

Następnie przesunąłem w odpowiednie miejsce zarys gniazda osi amortyzatora, i wyłoczyłem je wzdłuż osi **X** (Rysunek 2.2.11b). Tymi samymi metodami, których użyłem poprzednio do scalenia wypustu z resztą siatki, scalałem teraz z nią cylindry gniazd osi (Rysunek 2.2.12):



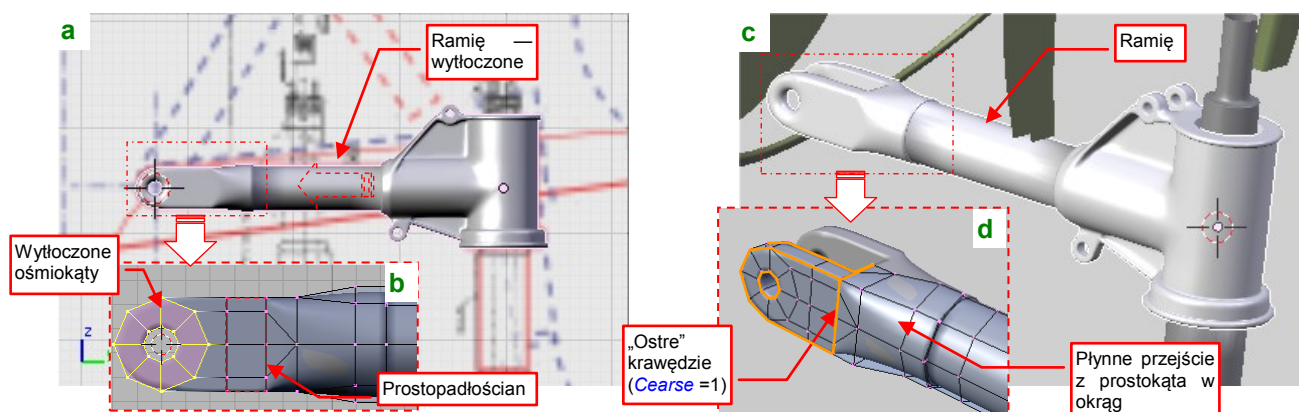
Rysunek 2.2.12 Gotowy wypust górny



Rysunek 2.2.13 Wahacz — gotowa końcówka

I to wszystko — górny wypust jest gotowy. W ten sam sposób uformuj drugi, u dołu wahacza (Rysunek 2.2.13).

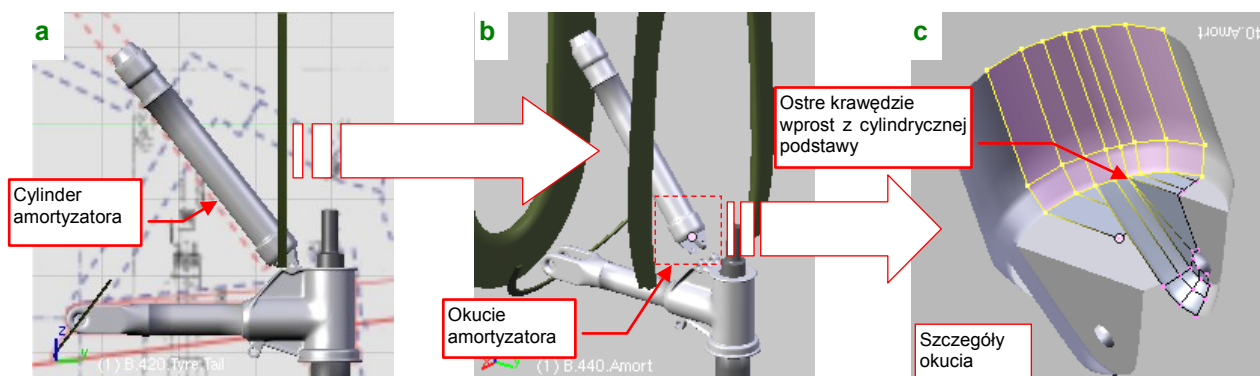
Do wykonania pozostała jeszcze reszta wahacza: ramię, które wyłoczę z końcówki, oraz okucie osi obrotu (Rysunek 2.2.14a):



Rysunek 2.2.14 Wahacz — okucie osi obrotu

Okucie osi obrotu wahacza wyłoczyłem z dwóch współosiowych ośmiokątów (Rysunek 2.2.14b). Potem połączyłem je z prostokątem, który z kolei płynnie łączy się z okrągłym ramieniem wahacza (Rysunek 2.2.14d). Takie końcówki napotkasz w wielu elementach tego modelu. Temu wahaczowi nadałem nazwę **B.530.SwingArm**.

Amortyzator wydaje się na pierwszy rzut oka walcem, zupełnie łatwym do uformowania (Rysunek 2.2.15a). Ot, wystarczy stworzyć cylinder, rozciągnąć, dodać kilka wytłoczeń w odpowiednich miejscach, a na koniec pochylić, i gotowe. A jednak, gdy przyjrzyś się bliżej jego dolnej (tej lepiej widocznej z zewnątrz!) części (Rysunek 2.2.15b), napotkasz tam niby proste okucie, którego wykonanie może przysporzyć niespodziewanych kłopotów (Rysunek 2.2.15c):

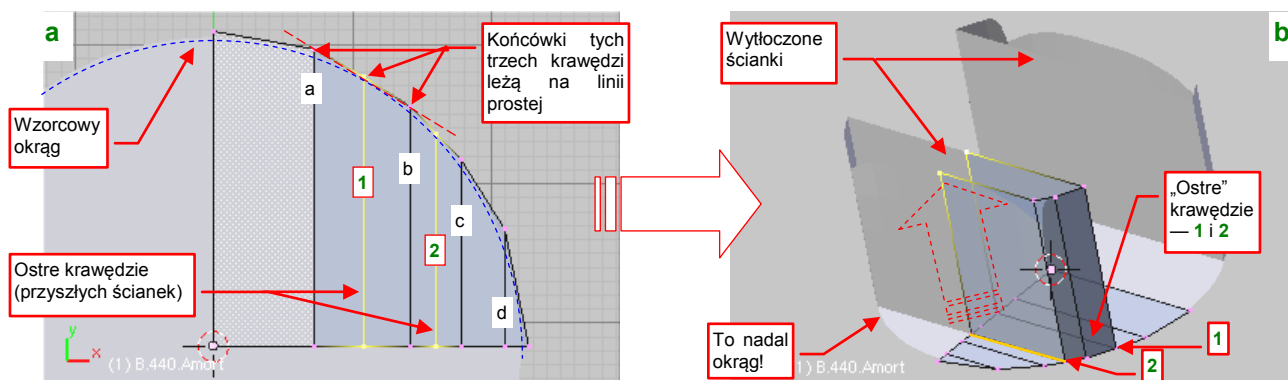


Rysunek 2.2.15 Amortyzator i jego okucie

Problem z kształtem tego okucia polega na tym, że jego ścianki „wyrastają” wprost z okrągłej podstawy (Rysunek 2.2.15c). Takie ostre załamania psują zawsze kształt wygładzonych powierzchni podziałowych. Jak więc uzyskać w tym miejscu taki „kant”, a jednocześnie zachować wygładzony, okrągły kształt krawędzi?

Można to zrobić całkiem łatwo, pod warunkiem że zaczniemy formowanie całego amortyzatora właśnie od tego okucia. Wstaw do rysunku jakiś okrąg, który będzie nam służył za wzorec kształtu. Potem wstaw drugi okrąg (np. ośmiokąt) — to początek amortyzatora. Możesz od razu dodać do siatki odpowiednie modyfikatory *Mirror* w kierunkach **X**, **Y**, i usunąć niepotrzebne wierzchołki, abyśmy dalej pracowali na „ćwiartce” okręgu. (Im mniej punktów do zmiany, tym lepiej!). Wytłocz (*Extrude* → *Edges*) krawędź ośmiokąta, zmieniając jej skalę wzdłuż osi **Y** do zera, by wypełnić „ćwiartkę” koła (Rysunek 2.2.16a).

Następnie wstaw w naszą „ćwiartkę” dodatkowe krawędzie przyszłych ścianek okucia: **1** i **2**. Obok nich dodaj kolejne, pomocnicze krawędzie: **a** i **c** (b już istniała — to był wierzchołek oryginalnego ośmiokąta) :

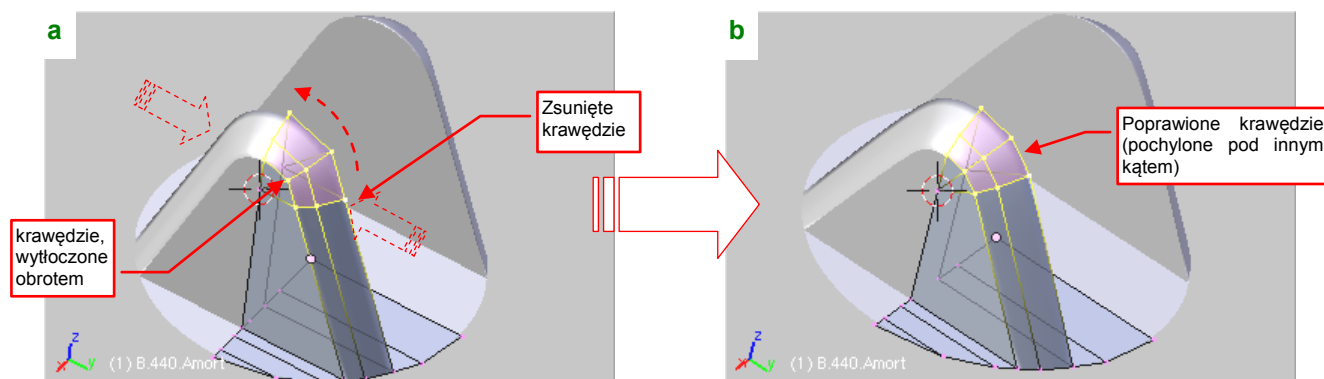


Rysunek 2.2.16 Końcówka amortyzatora — początki formowania

Jest taka własność powierzchni podziałowych: jeżeli końcówki „ostrej” krawędzi (np. **1**) i jej „sąsiadów” (**a**, **b**) leżą na linii prostej — to powłoka pozostanie w tym miejscu gładka (pokazuje to Rysunek 2.2.16a). Wystarczy teraz poprzesuwać trochę wierzchołki, by dopasować kształt obrysu do wzorcowego okręgu. Aby jak najlepiej odwzorować łuk, dodaj do siatki ostatnią, pomocniczą krawędź **d**. Całe dopasowanie zajmuje tylko chwilę. Potem możesz zaznaczyć krawędzie **1** i **2** jako „ostre” (*Crease* = 1). Sam zobaczysz, że nie zaburzy to w żaden sposób kształtu okręgu obwodu.

Teraz spokojnie możesz już wytłoczyć (*Extrude*) powierzchnię ograniczoną krawędziami **1** i **2** w ścianki okucia (Rysunek 2.2.16b).

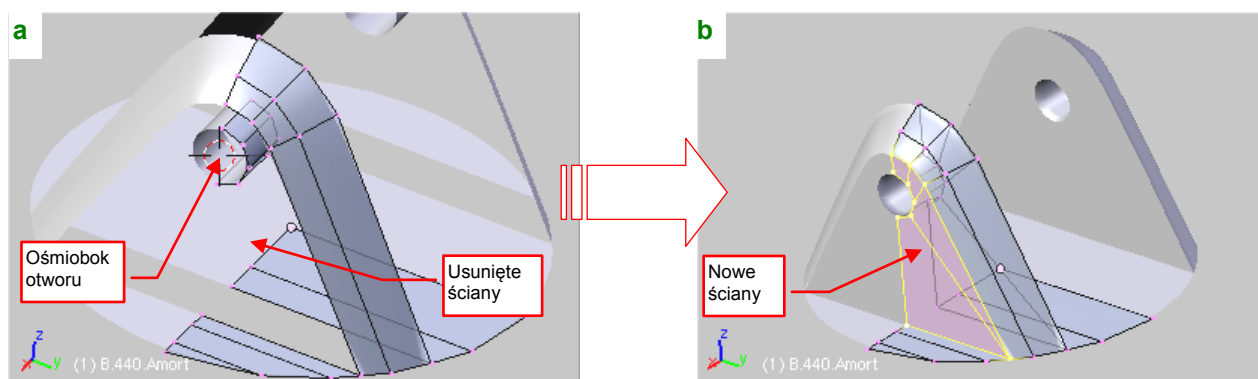
Końcówki wytłoczonych ścianek „zsuń” wzdłuż osi **Y** do środka, by nadać im kształt trójkąta. Następnie ustaw kursor 3D w wierzchołku leżącym w na płaszczyźnie symetrii — tu będzie biegła oś otworu. Wytłocz krawędź końcówek w dwóch obrotach w półokrągłe zakończenie (Rysunek 2.2.17a):



**Rysunek 2.2.17 Końcówka amortyzatora — formowanie wypustu**

Potem obróć nieco te krawędzie końcówki jeszcze raz, aby dopasować je lepiej do pochylenia ścianek okucia (Rysunek 2.2.17b).

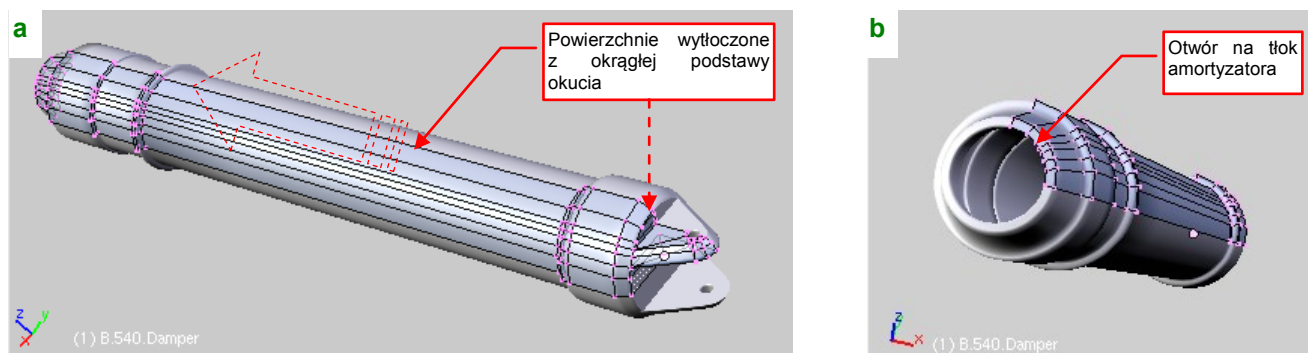
Następnie usuń dwa wierzchołki z centrum ścianki, by znikły dotychczasowe ściany boczne, i wstaw w ich miejsce kolejny, ośmioboczny cylinder (Rysunek 2.2.18a):



**Rysunek 2.2.18 Końcówka amortyzatora — formowanie wypustu**

Połącz krawędzie ścian otworu ze zewnętrznymi ścianami siatki, uzyskując w ten sposób nowe powierzchnie boczne (Rysunek 2.2.18b). Potem pozostaje tylko zaznaczyć je jako ostre, oraz zmienić tryb cieniowania na **Solid**, by okucie było gotowe.

Resztę siatki amortyzatora uzyskasz przez wytłoczenie okrągłej krawędzi podstawy okucia (Rysunek 2.2.19a):

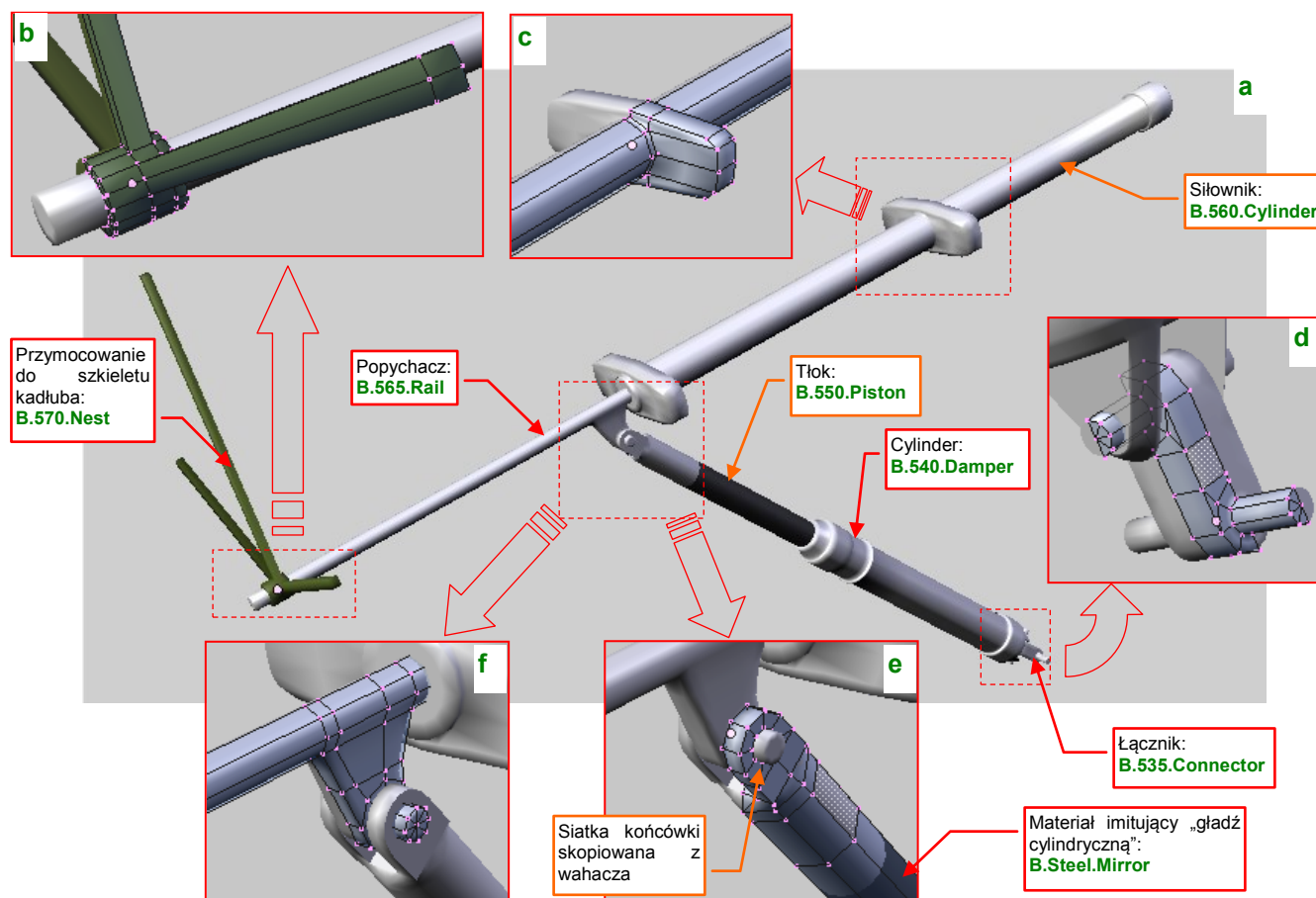


**Rysunek 2.2.19 Gotowy cylinder amortyzatora (B.540.Damper)**

Na drugim końcu tego cylindra pozostaw otwór na tłok (Rysunek 2.2.19b). Nadaj tej części nazwę „z serii 500”, np. **B.540.Damper**.



Rysunek 2.2.20 przedstawia elementy podzespołu amortyzatora (por. str. 38, Rysunek 2.2.1):



Rysunek 2.2.20 Podzespół amortyzatora

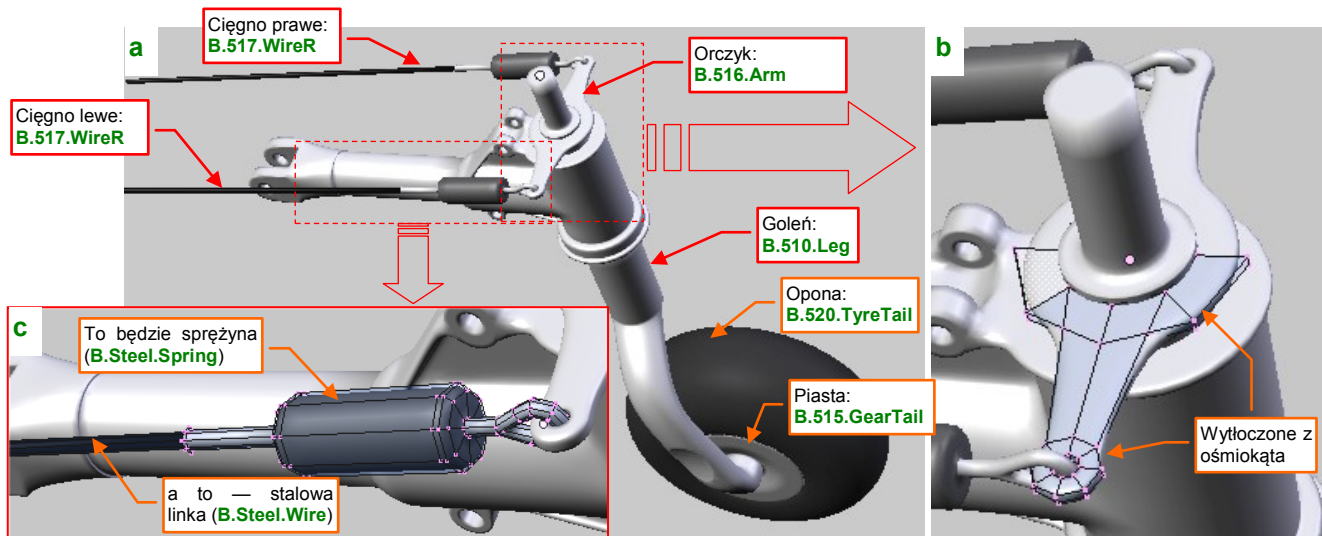
Cylinder amortyzatora był połączony z wahaczem poprzez dodatkowy łącznik (**B.535.Connector** — Rysunek 2.2.20d). Taki element umożliwiał dodatkowo zginanie się cylindra na boki. (Przypuszczam, że czasami wahaczem mogło całkiem porządnie „machnąć” w poprzek kadłuba, szczególnie na jakimś nierównym lądowisku). Siatka **B.335.Connector** powstała z połączenia dwóch wytłoczonych w cylindry ośmiokątów. Element jest niewielki, więc jest tylko wygładzona „zgrubnym” modyfikatorem *Subsurf* (*Level* = 1). Aby oszczędzić sobie pracy, zamodelowałem tylko „ćwiartkę” tej części, a resztę załatwiły za mnie modyfikatory *Mirror* wzdłuż lokalnych osi **X** i **Z** (Rysunek 2.2.20d). Wydaje mi się, że takie elementy były pozostawione w kolorze „surowej” stali — więc zastosowałem tu gładzy materiał **B.Steel.Smooth** (to **B.Steel.Rough** bez tekstur — p. str. 396).

Tłok amortyzatora (**B.550.Piston**) to prosty cylinder, zakończony „widelcem” z otworem na oś (Rysunek 2.2.20e). Siatkę widelca skopiowałem (**P** — *Separate*) z wykonanej wcześniej końcówki wahacza. Potem wystarczyło ją odpowiednio ustawić, zmniejszyć, i złączyć (**Ctrl-J** — *Join*) z siatką cylindra. Zwróć uwagę, że obszar roboczy cylindra ma bardzo gładką powierzchnię. Odzworowałem to, stosując w tym miejscu inny materiał: **B.Steel.Mirror** (to taka trochę ciemniejsza odmiana **B.Steel.Smooth**, bez tekstury zabrudzeń).

Siłownik (**B.560.Cylinder**) to walec, z którego w miejscu mocowań wytłoczyłem poprzeczne „skrzydełka” (Rysunek 2.2.20c). Grubość tych wypustów jest dopasowana do szerokości siatek żeber — tak, aby razem wyglądały jak zupełnie cienkie, blaszane wzmocnienia. Popychacz siłownika (**B.565.Rail**) to także cylinder, z którego w odpowiednim miejscu wytłoczyłem wypust z osią (Rysunek 2.2.20f). Przyłączyłem do niej cylinder amortyzatora. Ostatnim, najmniej widocznym z zewnątrz elementem całego zespołu, są wysięgniki (**B.570.Nest**). Mocują koniec popychacza do szkieletu kadłuba. W odróżnieniu od pozostałych części nie zastosowałem tu żadnego wygładzenia — bo i tak ten element będzie tylko majaczył w głębi kadłuba.

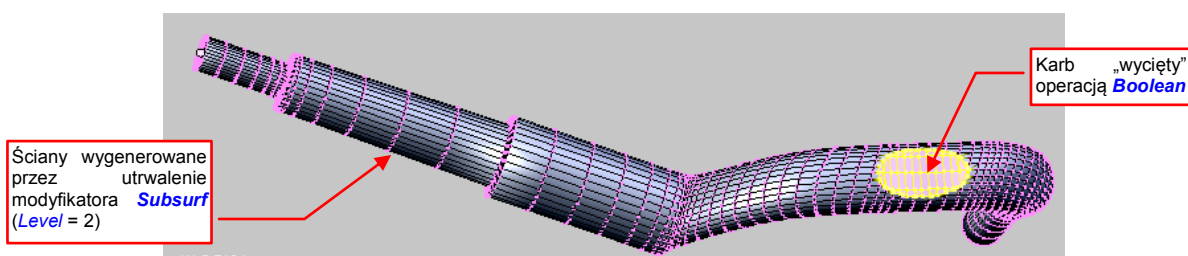
**B.535.Connector** jest „rodzicem” (*Parent*) cylindra amortyzatora (**B.540.Damper**). Siłownik (**B.560.Cylinder**) jest „rodzicem” popychacza (**B.565.Rail**), a ten z kolei — „rodzicem” tłoka amortyzatora (**B.550.Piston**).

Rysunek 2.2.21a) przedstawia zespół wahacza, wraz z cięgnami służącymi do skręcania kółka ogonowego (były sprężone ze sterem kierunku):



Rysunek 2.2.21 Podzespół wahacza

Siatki wahacza (**B.530.SwingArm**), opony (**B.520.TyreTail**), piasty (**B.515.GearTail**) i goleni (**B.510.Leg**) prezentowałem już w tej sekcji, więc nie ma co ich tu drugi raz pokazywać. Chociaż co do goleni, to muszę odnotować, że zdecydowałem się już ostatecznie „utrwalić” jej karb, uzyskany za pomocą modyfikatora **Boolean** (por. str. 39 i 284). Dlaczego teraz? Bo wydaje mi się, że nie będzie już trzeba poprawiać jej kształtu<sup>1</sup>. Aby utrwalić modyfikator **Boolean**, trzeba utrwalić także poprzedzający go modyfikator **Subsurf**. To dodaje goleni tyle wierzchołków, że można się pożegnać z myślą o jakichkolwiek dalszych poprawkach tej siatki (Rysunek 2.2.22):



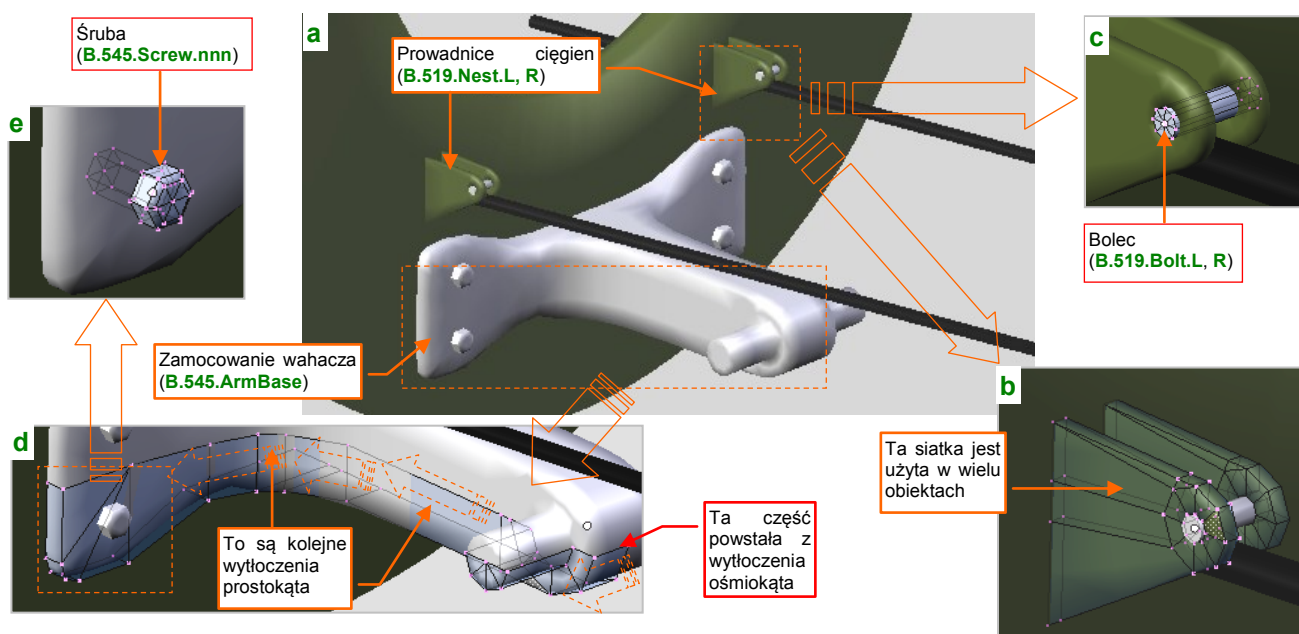
Rysunek 2.2.22 Siatka goleni (**B.510.Leg**) po „utrwaleniu” modyfikatorów **Subsurf** i **Boolean**

Elementem zespołu pokazywanego przez Rysunek 2.2.21a), którego do tej pory jeszcze nie omawialiśmy, jest orczyk (**B.516.Arm**), przymocowany do końca goleni. Wytłoczyłem go z dwóch ośmiokątów, które następnie połączyłem (Rysunek 2.2.21b). Zastosowałem tu symetrię wzdłuż osi **X** i **Z** (można było i wzdłuż **Y**, ale nim o tym pomyślałem, miałem już element gotowy). Podobnie jak łącznik wahacza z amortyzatorem (**B.535.Connector**), orczyk jest wykonany z „surowej” stali (materiał **B.Steel.Smooth**). Cięgna — lewe i prawe (**B.517.Wire.L** i **B.517.Wire.R**) używają tej samej siatki **B.517.Wire**. Sama siatka to odpowiednie wytłoczone cylindry (Rysunek 2.2.21c). „Rurki”, modelujące cięgna, mają przypisany specjalny materiał **B.Steel.Wire**. Na renderze wyglądają dzięki niemu jak plecione stalowe linki (szczegóły — str. 397) Zwróć uwagę na zgrubienia przy orczyku — za pomocą odpowiedniego materiału (**B.Steel.Spring**) zmienię je na renderingu w dość realistyczne sprężyny (szczegóły — p. str. 398).

Jeżeli chodzi o hierarchię: wahacz (**B.530.SwingArm**) jest „rodzicem” goleni (**B.510.Leg**). Goleń jest z kolei „rodzicem” piasty koła (**B.515.GearTail**) i orczyka (**B.516.Arm**). Piasta jest „rodzicem” opony (**B.520.TyreTail**), a orczyk — cięgien (**B.517.Wire.R** i **L**).

<sup>1</sup> Gdyby jednak trzeba było je jeszcze poprawić, skopiowałem (**Shift-D**) oryginalne: goleń z włączonymi modyfikatorami i obiekt „tnącym”, i przeniósłem te kopie (**Object→Make Links→To Scene...**) do sceny **Tools**. To nasz schowek na różne elementy, które mogą się jeszcze przydać.

Jako kolejny zespół można wyróżnić elementy, przymocowane „na sztywno” do wzmocnionej wręgi kadłuba (Rysunek 2.2.23a) :

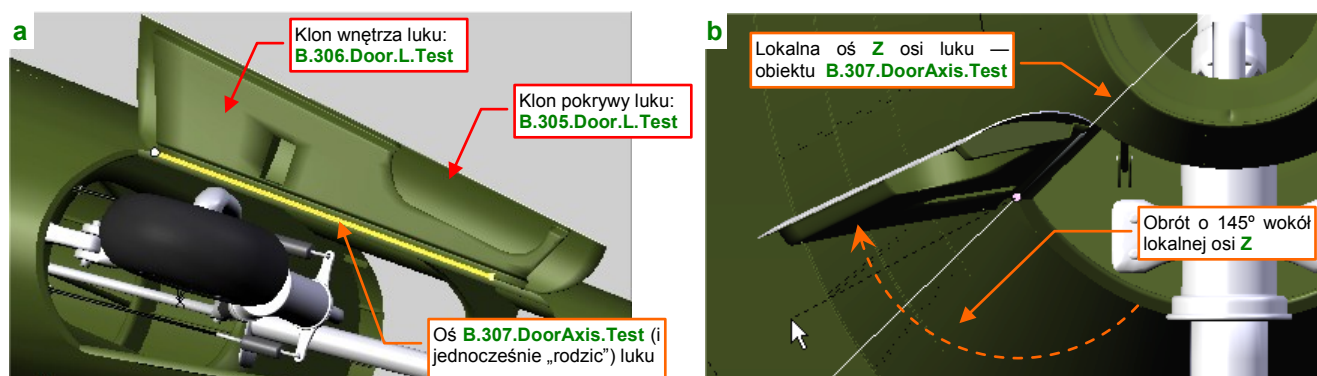


Rysunek 2.2.23 Elementy przymocowane do wręgi ogonowej

Najważniejszą częścią jest tu okucie **B.545.ArmBase**, mocujące wahacz do wręgi. Oś obrotu wahacza wytłoczyłem z ośmiokąta, a resztę — za pomocą kolejnych wytłoczeń sekcji prostokątnej (Rysunek 2.2.23d). Formowałem tylko symetryczną „ćwiartkę” okucia, a potem ją powieliłem za pomocą modyfikatora *Mirror* wzdłuż lokalnych osi **X** i **Z**. W tym miejscu były bardzo widoczne łby czterech śrub, więc nie pozostało mi nic innego, jak je zamodelować (**B.545.Screw.001 - .004**). Śruby przydadzą się w dziesiątkach innych miejsc, więc utworzyłem je jako cztery klony tej samej siatki — **B.Screw**. Sama siatka jest bardzo prosta (Rysunek 2.2.23e). Zwróć uwagę na położenie punktu centralnego obiektu — jest w płaszczyźnie podstawy łba śruby, aby najwygodniej ustawić ich podstawy. „Rodzicem” wszystkich czterech śrub jest oczywiście okucie **B.545.ArmBase**.

Kolejnymi składnikami zespołu pokazywane przez Rysunek 2.2.23a są prowadnice ciągien (**B.517.Nest.L** i **.R**). Takie trójkąty z otworem na końcu przydadzą się jeszcze w innych miejscach, więc, podobnie jak śruby, są to klony jednego wzorca (Rysunek 2.2.23b). Łączy je bolec (**B.519.Bolt.L** i **.R**). Siatka tego cylindra także będzie zawsze powieleniem jednego pierwowzoru: **B.Bolt**. Jeżeli chodzi o hierarchię, to bolec (**B.519.Bolt.L** lub **.R**) i prawa ścianka (**B.519.Nest.002.L** lub **.R**) są przypisane do lewych ścianek (**B.519.Nest.001.L** i **.R**), a te — do wręgi kadłuba.

Nim zabierzemy się za formowanie popychaczy luków podwozia, sklonujmy (**Alt-D**) do testów jeden luk wraz z osią (**B.307.DoorAxis.R**) (Rysunek 2.2.24a):



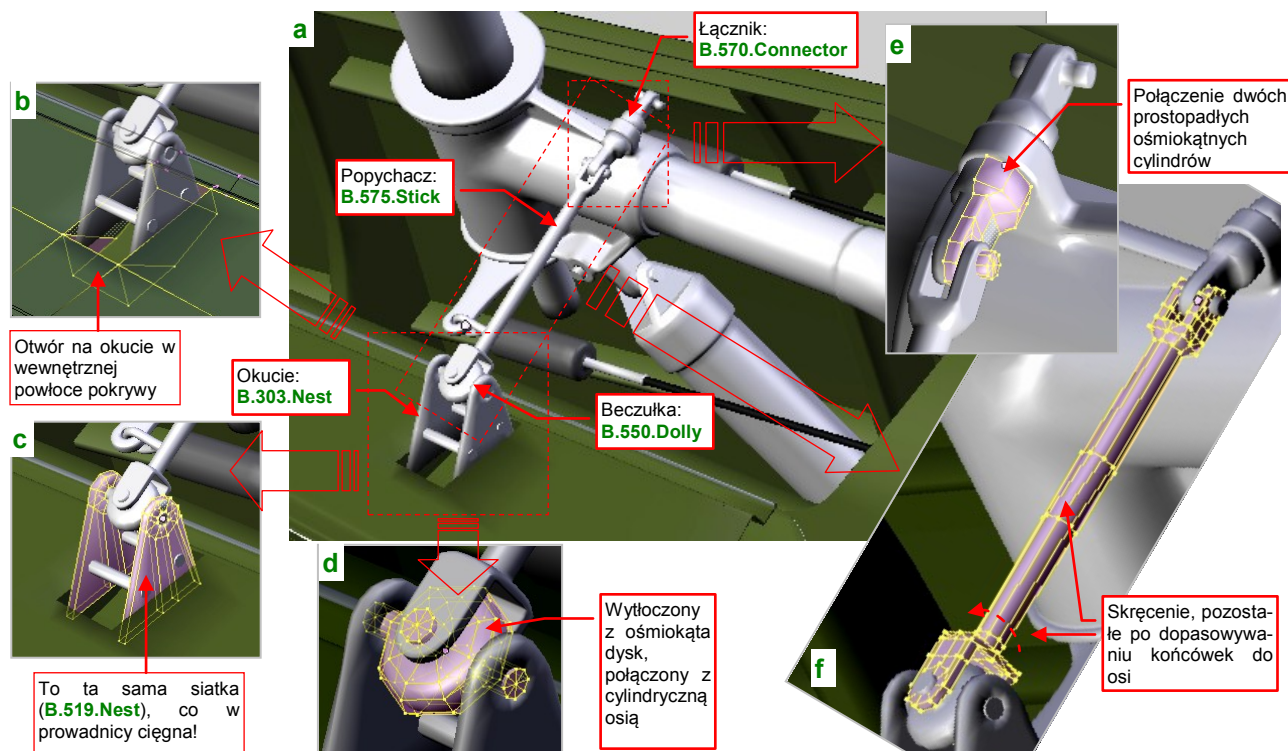
Rysunek 2.2.24 Robocza kopia luku podwozia i jego osi

Luk po skopiowaniu otwórz, obracając oś **B.307.DoorAxis** o jakiś „równy” kąt — np. 145° (Rysunek 2.2.24b).



Jeżeli obrócisz zespół luku podwozia o jakiś przypadkowy kąt, będziesz musiał potem go z powrotem pracownice dopasowywać do pozycji „zamkniętej”. Dlatego — na wszelki wypadek — lepiej jest stworzyć robocze klony pokryw i do nich przymierzać otwarte podwozie. Ustaw taki klon luku w pozycji „otwarte” i dopasuj go do kółka ogonowego. Przekonasz się, że to bardzo wygodne. Możesz np. zmieniać kształt jego siatki, a zmiany są od razu widoczne na pierwowzorach pokryw. Konieczność takiej zmiany przytrafiła się i mi: podczas formowania zapomniałem wykonać w wewnętrznej powierzchni pokrywy podwozia otworu na okucie popychacza (Rysunek 2.2.25b). Musiałem ją uzupełnić teraz, oczywiście przymierzając od razu okucie do właściwych rozmiarów.

Rysunek 2.2.25a) pokazuje szczegóły popychacza pokrwy kółka ogonowego:



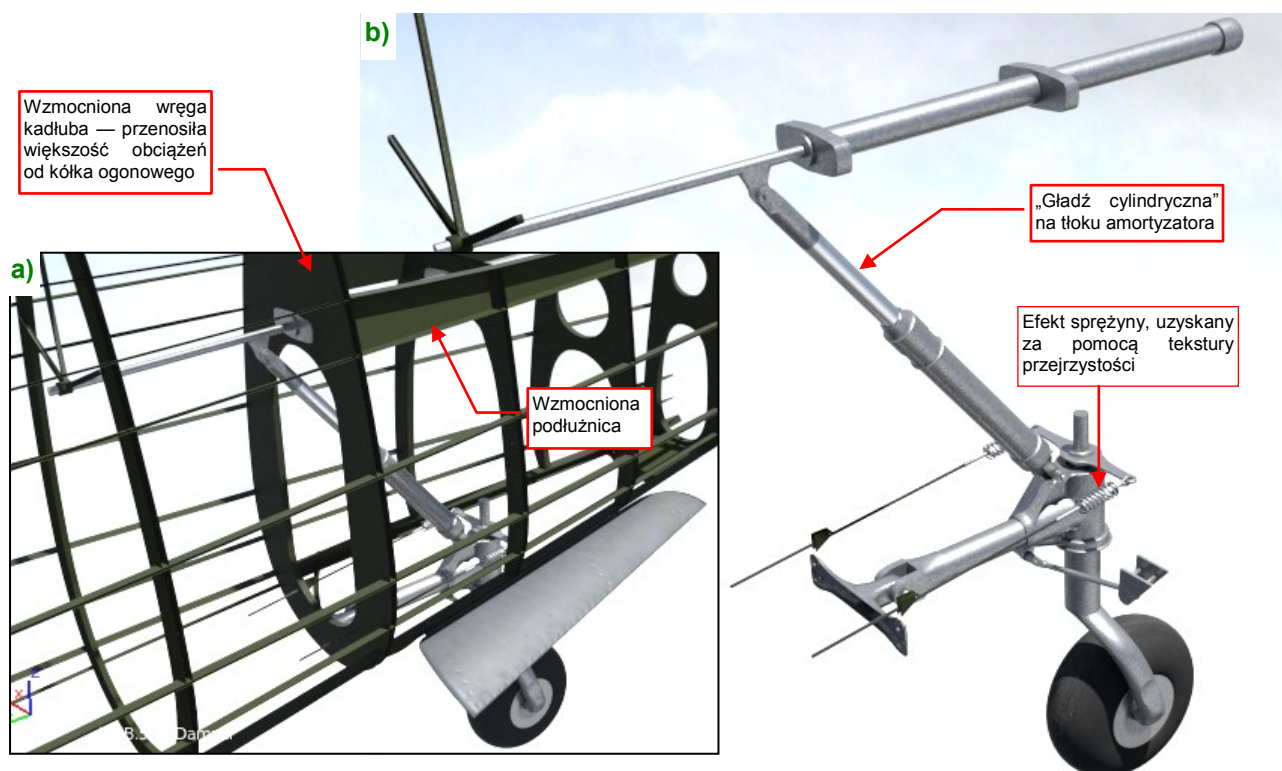
Rysunek 2.2.25 Zespół popychaczy luku podwozia

Popychacze lewy i prawy (**B.575.Stick.L** i **.R**) są połączone ze sobą poprzez otwór w wahaczu łącznikiem **B.570.Connector** (Rysunek 2.2.25e). Ten łącznik może się obracać — bo inaczej popychacze by się zablokowały podczas chowania podwozia. Każda z końcówek popychacza **B.575.Stick** ma inny kształt i jest pochylona pod innym kątem, by dopasować się do „swojej” osi. Uzyskałem to poprzez obrót części siatki wokół lokalnej osi **Z** obiektu. Śladem p tych zabiegach jest skręcona siatka pośrodku popychacza (Rysunek 2.2.25f).

Okucie pokrywy podwozia zbudowałem z klonów tej samej trójkątnej siatki „blaszki” (**B.519.Nest**), co prowadnice ciągnię (por. str. 47, Rysunek 2.2.23b). Skoro wszystkie te obiekty używają tej samej siatki, to jak udało mi się przypisać im inny kolor? To nic trudnego: na okuciach pokryw przełączyłem tryb przypisania materiału z **Data** na **Object** (szczegóły — str. 227). W trybie **Object** renderer używa materiału przypisanego do obiektu, a nie siatki. Dzięki temu każdy z klonów może mieć inny kolor, bo jest odrębnym obiektem. Do każdego z okuc **B.303.Nest.L** i **.R** dodałem u dołu dwa dodatkowe bolce (**B.303.Bolt.L.nnn** lub **.R.nnn**) — bo są widoczne na zdjęciach. W osi okucia umieściłem „beczulkę” (**B.550.Dolly**). Ten element umożliwia obrót końcówki popychacza w dwóch płaszczyznach. Mówiąc szczerze, nie jestem pewien kształtu tego elementu — na wszystkich fotografiach, które posiadam, jest zasłonięty przez blaszki okucia.

Hierarchia tego zespołu wygląda następująco: centralny łącznik (**B.570.Connector**) jest „rodzicem” popychacza (**B.575.Stick**). („Rodzicem” centralnego łącznika jest wahacz — **B.530.SwingArm**). Wewnętrzna część pokrywy luku podwozia (**B.306.Door**) jest „rodzicem” okucia (**B.303.Nest**). Okucie jest „rodzicem” beczulki (**B.550.Dolly**) oraz bolców wzmacniających.

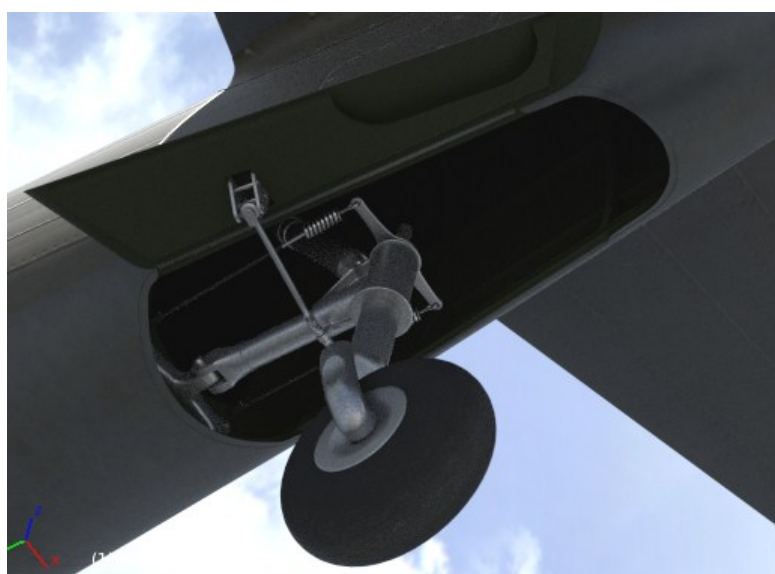
Rysunek 2.2.26a) przedstawia uformowany mechanizm kółka ogonowego, wkomponowany w szkielet kadłuba, a Rysunek 2.2.26b) — bez szkieletu (widać wtedy więcej szczegółów):



**Rysunek 2.2.26 Mechanizm kółka ogonowego — „w szkielecie” i osobno**

Teraz konstrukcja ogona kadłuba stała się bardziej zrozumiała. Wręga przed lukiem podwozia musiała być taka szeroka (wzmocniona), bo przenosiła większość obciążeń pochodzących od kółka ogonowego. Skrzynkowe, wzmocnione podłużnice pod statecznikiem poziomym tworzyły wraz z tą wręgą „ramę”, w której zamocowany był siłownik.

Rysunek 2.2.27 przedstawia, jak teraz wygląda wnętrze luku podwozia. Przyznasz, że prezentuje się już bardziej realistycznie, prawda?



**Rysunek 2.2.27 Widok do wnętrza komory kółka ogonowego**

Myślę, że nie będziemy już dodawać tutaj dalszych detali. Oczywiście, w następnej sekcji uzupełnimy jeszcze tylko brakujący, prawy popychacz pokrywy, wraz z okuciem.

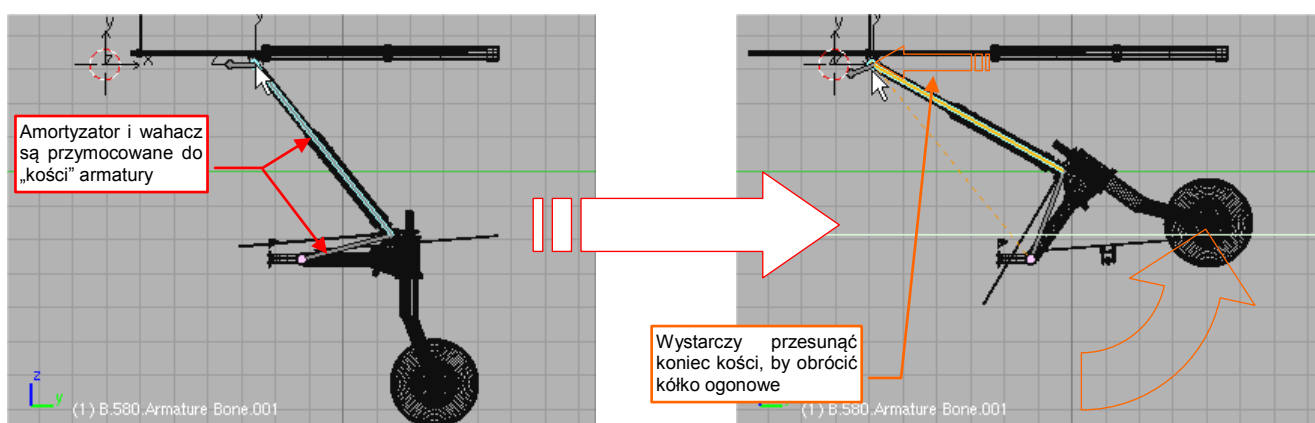
### Podsumowanie

- W tej sekcji starałem się pokazać ze wszystkimi szczegółami, jak modeluje się podwozie samolotu. Zrobiłem to na przykładzie stosunkowo prostego zespołu kółka ogonowego. Gdy w dalszych sekcjach będę omawiał podwozie główne, pominę już szczegóły formowania każdej śrubki czy popychacza. Po zapoznaniu się z tą sekcją, powinieneś być w stanie tworzyć je szybko, bez żadnych podpowiedzi;
- Podwozie na zwykłych planach samolotu jest zazwyczaj narysowane w zbyt małej skali, a przez to — niewystarczająco dokładnie. Aby jak najlepiej odwzorować je w modelu, zgromadź możliwie dużo zdjęć tego zespołu. Nie zaszkodzi także przygotować jakiegoś schematycznego rysunku (str. 38);
- Przed rozpoczęciem modelowania podwozia dobrze przemyśl, co i jak odwzorujesz. Musisz zdecydować, co należy pominąć, a co — uprościć (str. 41). Mnie pomaga w tym wykonanie odręcznych szkiców modelowanego mechanizmu. Na pewno istnieją także inne metody ☺;
- Najtrudniejszymi do modelowania fragmentami są elementy z wieloma wypustami i otworami. Przykładem tak skomplikowanej części jest np. wahacz kółka ogonowego. Jego formowanie składa się z wielu etapów (str. 41 i dalsze);
- Kształty prawie wszystkich części mechanizmów można „wyprowadzić” z początkowej kombinacji cylindrów i prostopadłościanów. Dobre przybliżenie cylindra uzyskuje się z ośmiokąta, „wygładzonego” modyfikatorem *Subsurf* (z *Level* = 2 dla większych obiektów, lub *Level* = 1 dla mniejszych). Tylko w wyjątkowych wypadkach należy stosować operacje typu *Boolean* (str. 39, 46);
- Inne problemy mogą się kryć w prostych z pozoru przypadkach, jak wyprowadzenie prostopadłej ścianki z okrągłej podstawy (str. 43). Aby coś takiego uzyskać z powierzchni podziałowej (*Subsurf*), musisz przygotować odpowiedni układ wierzchołków. Jeżeli rozpoczniesz siatkę, od razu uwzględniając powstanie takich ostrych krawędzi, nie jest to nawet specjalnie trudne (str. 43);
- Zaraz po stworzeniu nowego obiektu przypisz mu odpowiedni materiał. Do drobnych części używaj prostszych materiałów, których tekstury nie wymagają rozwinięcia siatki w przestrzeni UV. Zazwyczaj taka „stal”, „guma”, czy „ciągną”, będą się później powtarzać w innych zespołach samolotu (str. 39, 40);
- Drobne sprężyny można modelować za pomocą zwykłych walców, pokrytych specjalnym, półprzezroczystym materiałem (str. 46, 49);
- Jako elementy, które się powtarzają (śruby, bolce, płaskowniki, ...) warto wykorzystywać klony pojedynczej siatki (np. wzorcowa śruba — str. 47);
- Czasami trzeba w jakimś konkretnym klonie zmienić materiał (chodzi o sytuacje, gdy ma być inny niż ten, przypisany do wzorcowej siatki). Wystarczy wówczas przełączyć taki element z wykorzystania materiału siatki (*Data*) na materiał obiektu (*Object*) — str. 48;
- Przy wzajemnym dopasowywaniu pokryw luków i mechanizmu podwozia, warto pozostawić oryginalne pokrywy w pozycji „zamkniętej” (tak, jak je wcześniej zamodelowaliśmy). Stwórz za to dodatkowy, roboczy klon pokrywy i jej osi, i ustaw je w pozycji „otwarte” (str. 47);

### 2.3 Mechanizacja kółka ogonowego

Jak już zdążyłeś się przekonać w poprzedniej sekcji, każde podwozie to mniej lub bardziej złożony zespół kół, goleni, amortyzatorów i różnorodnych popychaczy. Większość samolotów z okresu II Wojny Światowej miała w dodatku podwozie chowane. Wyobraź sobie, że robisz dwie sceny ze swoim modelem: jedną na lotnisku, a drugą w locie. W pierwszej podwozie musi być wysunięte, a w drugiej — schowane. Oczywiście, można je chować i wysuwać element po elemencie. Zapewniam Cię jednak, że za drugim razem podobna zabawa staje się już bardzo nużąca. Po co zresztą się tak męczyć, skoro Blender udostępnia nam różnorodne środki animacji? Wystarczy z nich skorzystać, by związać te wszystkie części w jeden mechanizm, uruchamiany przesunięciem pojedynczego uchwytu.

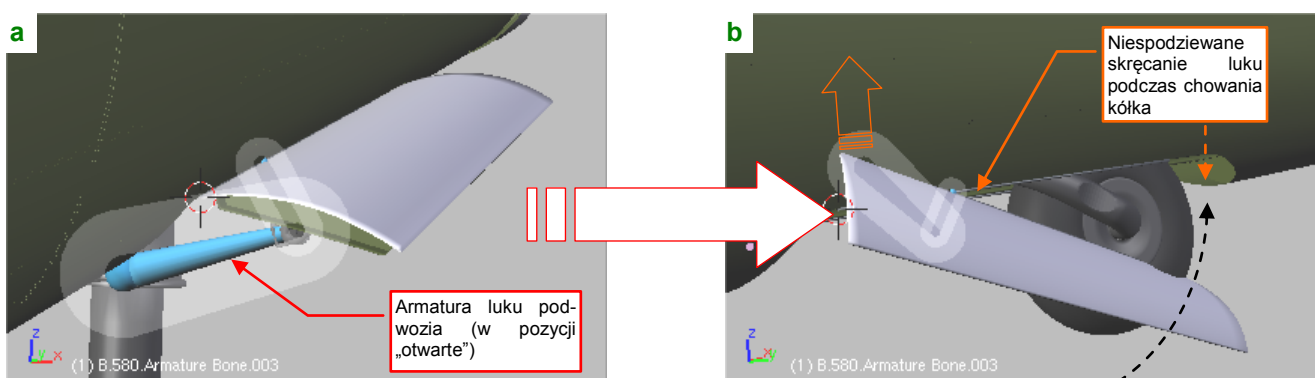
Pierwszym rozwiązaniem, które się w takiej sytuacji nasuwa, jest wykorzystanie takich samych szkieletów, jakich używa się w Blenderze do animacji postaci ludzi czy zwierząt. Te szkielety nazywane są w grafice komputerowej *armaturami* (*armatures*). Składają się (a jakże by inaczej!) — z kości (*bones*). Wstawiający do rysunku nieskomplikowaną armaturę, złożoną z dwóch połączonych kości, można szybko uzyskać podstawowy efekt chowania i wysuwania kółka ogonowego (Rysunek 2.3.1):



Rysunek 2.3.1 Chowanie kółka ogonowego za pomocą armatury

(Wszystko na temat budowy takiej armatury znajdziesz na str. 229 i następnych). Niestety, tak prosto można okiełznać tylko płaski zespół elementów. W przypadku ruchu w przestrzeni kości zaczynają się skręcać wokół osi podłużnej, co prowadzi do zaskakujących efektów.

Zobacz, na przykład, układ dwóch kości, modelujący mechanizm zamykania luku podwozia (Rysunek 2.3.2a):



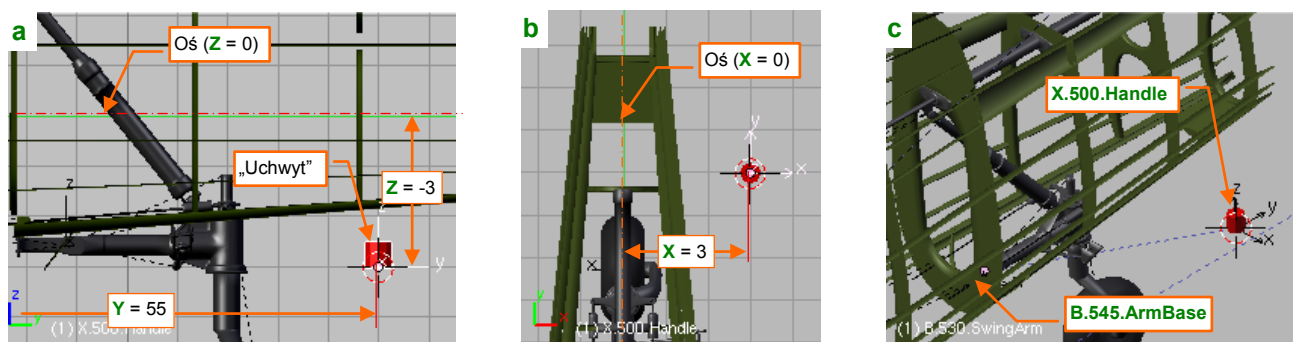
Rysunek 2.3.2 Przykład problemów z odwrotną kinematyką kości (armatura luku podwozia)

Wydaje się równie prosty jak ten poprzedni, prawda? A jednak, gdy koniec ramienia zaczyna wędrować, wraz z wahaczem, do góry, cała pokrywa także się składa, ale jednocześnie — skręca (Rysunek 2.3.2b). Co prawda, można zapanować nad takim efektem ubocznym za pomocą różnych ograniczeń (*Constraints*), ale cała praca zamienia się w dość męczące zajęcie.



Sądzę, że taką „niesformą” armaturę można okiełznać, ale, mówiąc szczerze, nie chciało mi się poświęcać na to czasu. Przy okazji prób radzenia sobie z jej obrotami poznałem system ograniczeń (*constrains*) Blendera. Szybko doszedłem do wniosku, że do osiągnięcia zamierzonego efektu — otwierania i zamykania całego podwozia za pomocą jednego przesunięcia. — kości właściwie nie są potrzebne. Wystarczy tylko zamienić miejscami przyczynę ze skutkiem. To nie wahacz będzie się chował, bo jest ciągnięty w górę przez amortyzator, tylko amortyzator podniesie się do góry, bo jest popychany przez obrót wahacza. Gdy to dobrze rozegramy, zewnętrzny obserwator nawet nie zwróci uwagi, jakie są prawdziwe zależności pomiędzy poszczególnymi częściami zespołu. A gdy odwrócisz przyczynę i skutek, możesz cały mechanizm zamodelować za pomocą kilku ograniczeń. Nie trzeba do tego żadnej armatury, ani *Pose Mode*!

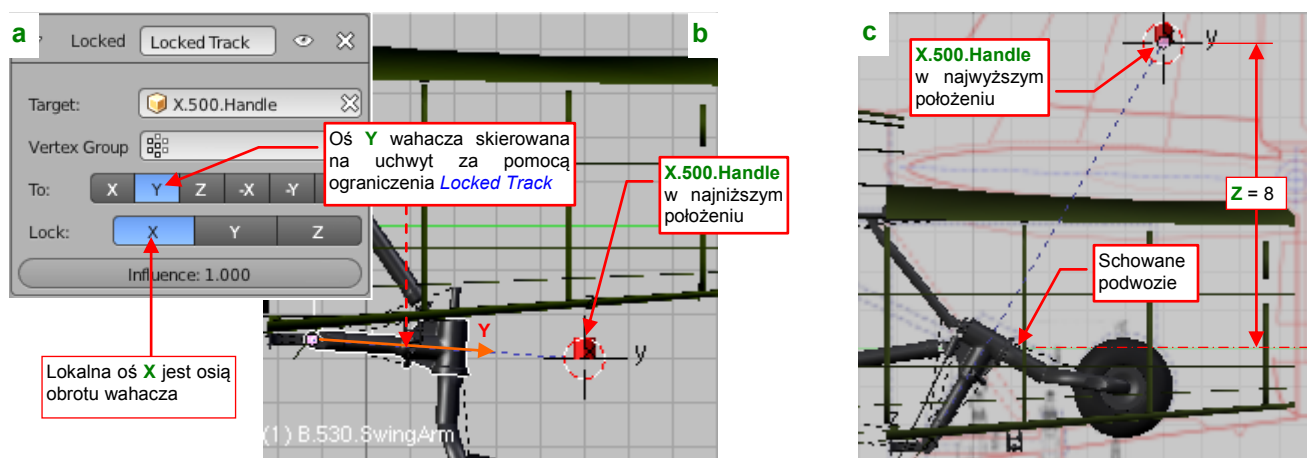
Zacznijmy od wstawienia obiektu — uchwytu, którego przemieszczenie będzie sterować wysuwaniem i chowaniem podwozia (Rysunek 2.3.3):



Rysunek 2.3.3 Uchwyt, który będzie poruszał całym zespołem kółka ogonowego

Osobiście używam na takie uchwyty zwykłych, ośmiobocznych cylindrów. Przypisałem mu materiał **Artificial** i umieściłem na warstwie **10** (tej samej, na której znajduje się „cel” kamery — por. Wyciąg II). Uchwyt położony jest nieco z boku kadłuba (Rysunek 2.3.3a,b) by w razie potrzeby łatwo go było dostrzec. Postaraj się dobrać jego położenie tak, by jego współrzędne były liczbami całkowitymi. Nie jest to niezbędne, ale bardzo ułatwia dalszą pracę. Nadajmy mu nazwę z innym przedrostkiem, by odróżniał się od „zwykłych części”. Niech to będzie **X.500.Handle**. Włącz uchwyt do hierarchii modelu, przypisując go (relacją *Parent*) do okucia wahacza (**B.545.ArmBase** — por. Rysunek 2.3.3c, a także str. 47).

Aby wahacz (**B.530.SwingArm**) podążał za ruchem uchwytu, przypisz mu ograniczenie *Locked Track* (Rysunek 2.3.4, więcej na temat tego ograniczenia — str. 237):

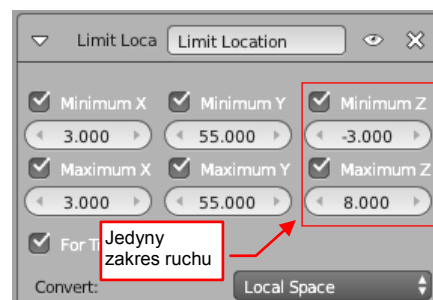


Rysunek 2.3.4 Przypisanie do wahacza ograniczenia *Locked Track*.

W parametrach ograniczenia wskaż jako cel obiekt **X.500.Handle** (Rysunek 2.3.4a). Ustal **X** jako oś obrotu, a **Y** niech celuje w środek uchwytu (Rysunek 2.3.4b). Gdy przesuniesz uchwyt do góry (do pozycji **Z = 8.0**), koło podwozia znajdzie się w pozycji „schowane” (Rysunek 2.3.4b).

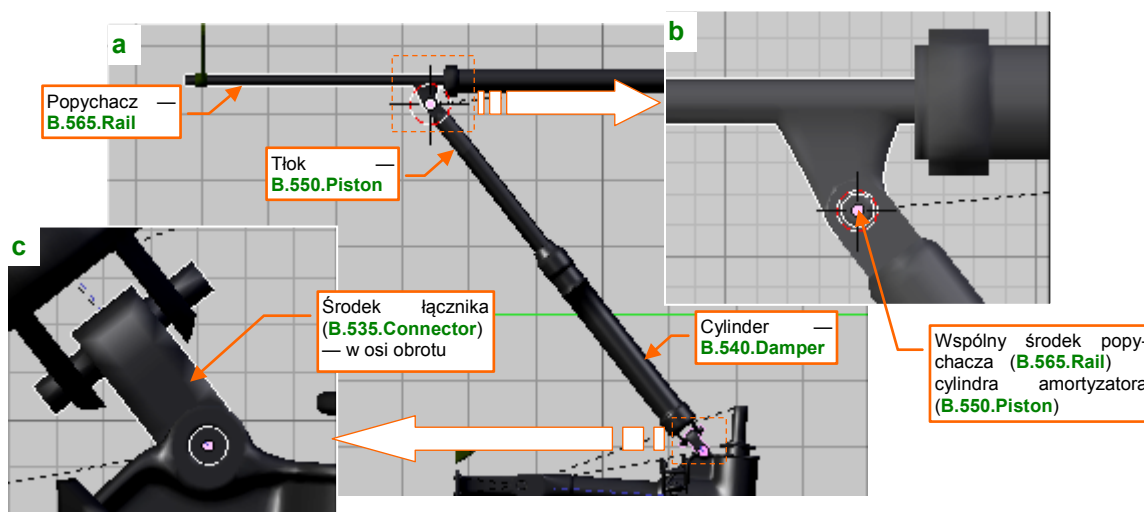


Dolne i górne położenia uchwytu musisz dopasować tak, by odpowiadały właściwemu położeniu kółka ogonowego. (Położenie dolne powinno odpowiadać podwoziu całkowicie wysuniętemu, a górne — schowanemu). W razie czego możesz lepiej dobrać to położenie, zmieniając odległość uchwytu od środka obrotu wahacza (zmieniając współrzędną **Y**). Gdy zakres ruchu uchwytu jest już ustalony, „utrwal” go za pomocą odpowiedniego ograniczenia **Limit Location** (Rysunek 2.3.5). (Więcej na temat tego rodzaju ograniczenia i jego parametrów znajdziesz na str. 241).



Rysunek 2.3.5 Ograniczenie ruchu uchwytu.

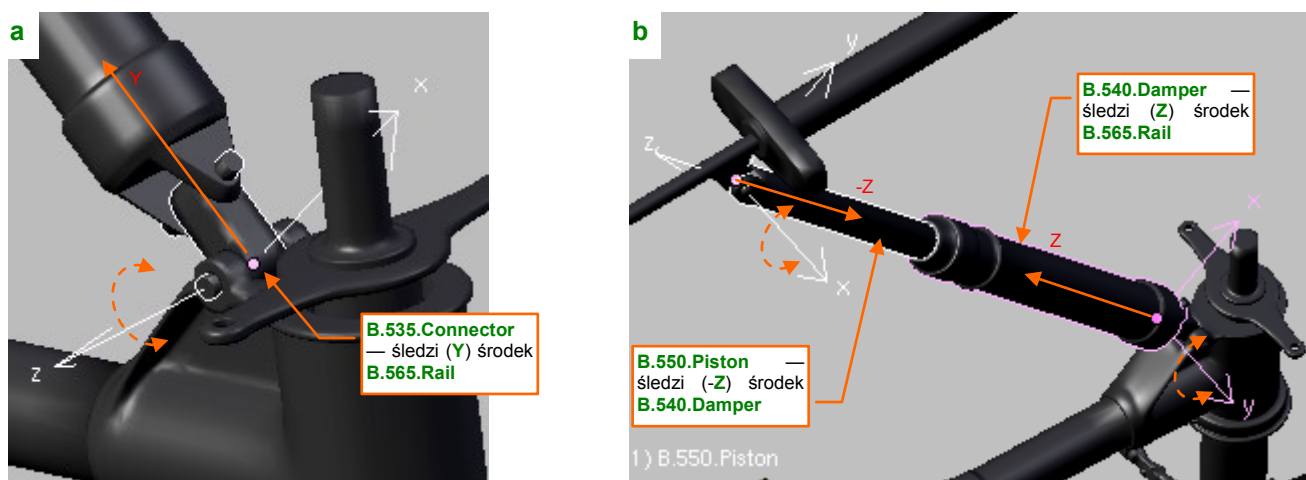
Zajmijmy się teraz górną częścią podwozia: popychaczem i amortyzatorem. Bardzo ważne jest odpowiednia pozycja środków i hierarchii tych elementów (Rysunek 2.3.6):



Rysunek 2.3.6 Położenie środków poszczególnych części zespołu amortyzatora

Środek popychacza (**B.565.Rail**) umieść (jak — zob. str. 224) w osi obrotu tłoka amortyzatora (Rysunek 2.3.6b). Jeżeli chodzi o hierarchię, to przypisz (relacją **Parent**) tłok (**B.550.Piston**) do popychacza **B.565.Rail**. Cylinder amortyzatora (**B.540.Damper**) przypisz (relacją **Parent**) do niepozornego łącznika (**B.535.Connector** — Rysunek 2.3.6c), a sam łącznik — do wahacza (**B.530.SwingArm**).

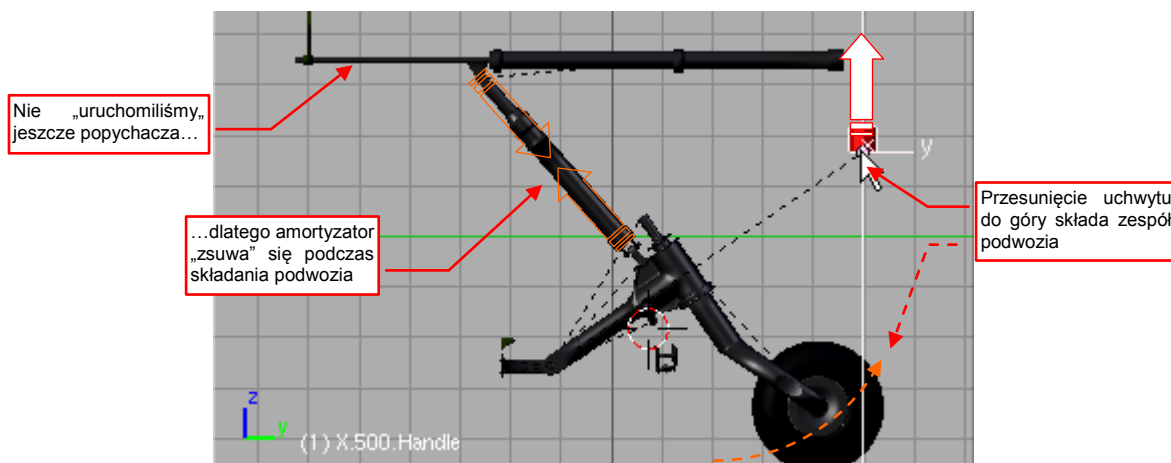
Teraz przypisz łącznikowi **B.535.Connector** ograniczenie **Locked Track** tak, by jego oś **Y** „celowała” w środek popychacza (Rysunek 2.3.7a):



Rysunek 2.3.7 Ograniczenia **Locked Track**, przypisane do elementów amortyzatora

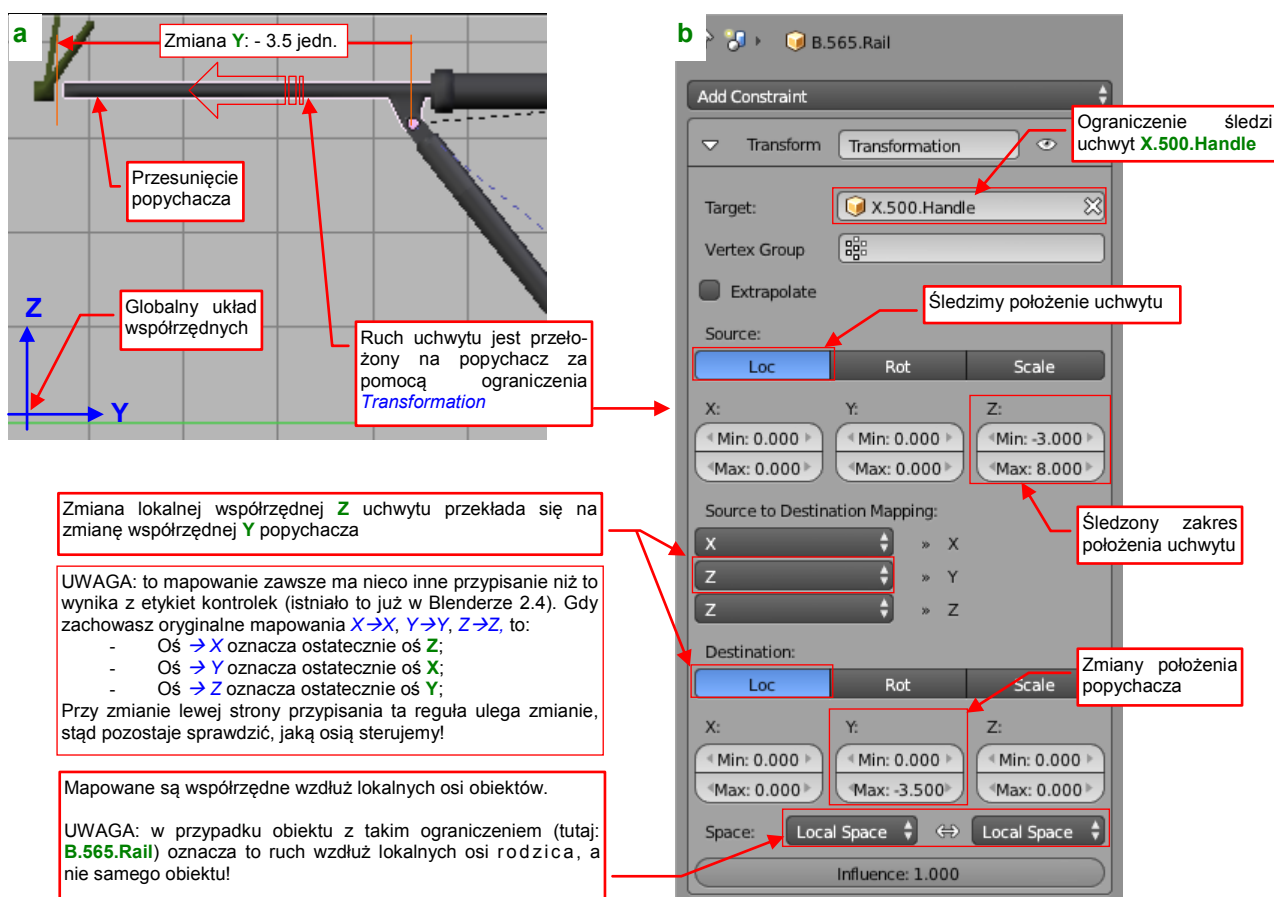
Jednocześnie ogranicz cylinder **B.550.Piston** tak, by jego oś **Z** śledziła środek cylindra **B.540.Damper** (Rysunek 2.3.7b). Na wszelki wypadek możesz także dodatkowo ograniczyć cylinder, by „celował” w tłok.

Na razie nasze ograniczenia powodują, że w trakcie obrotu wahacza amortyzator jest niemiłosiernie ściskany (Rysunek 2.3.8):



Rysunek 2.3.8 Rezultat działania ograniczeń: ugięcie amortyzatora

To dlatego, że nie uruchomiliśmy jeszcze popychacza. Gdy będziesz przesuwiał uchwyt **X.500.Handle** do góry, popychacz powinien się przesunąć w lewo (Rysunek 2.3.9a), o 3.5 jednostki Blendera. Najprościej ten efekt uzyskać za pomocą „uniwersalnej przekładni” — ograniczenia **Transformation** (Rysunek 2.3.9b):

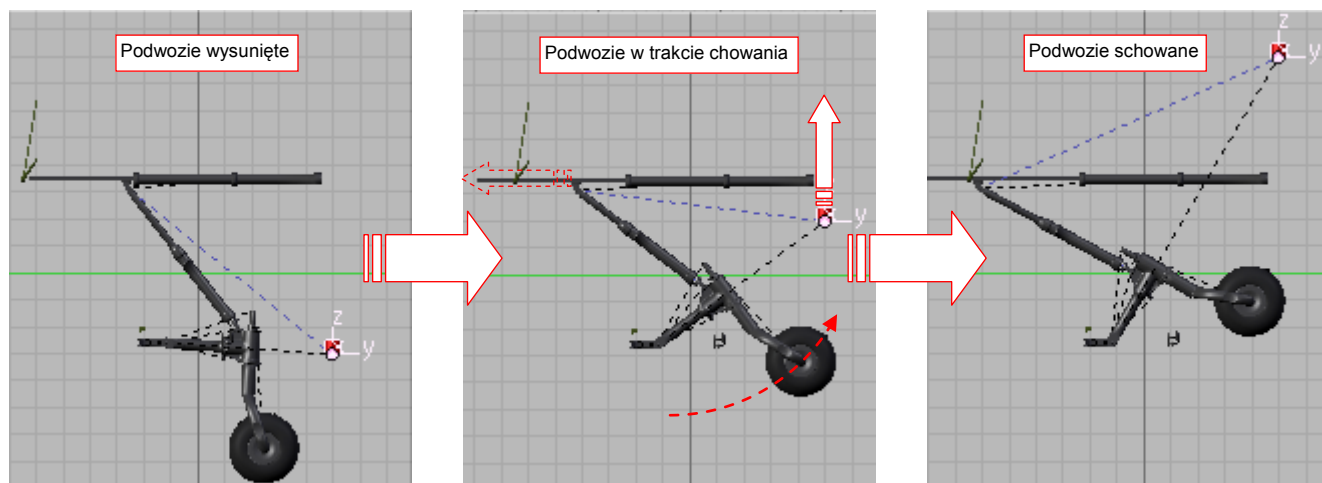


Rysunek 2.3.9 Przełożenie ruchu uchwytu na przesunięcie popychacza (ograniczenie **Transformation**)

Jak widzisz, panel tego ograniczenia ma wyjątkowo duże rozmiary. Ustala się w niej obiekt sterujący (określany jako **Target** lub **Source**), oraz przełożenie jego współrzędnych na obiekt do którego przypisano ograniczenie. Jak widać z opisu na ilustracji, to w zasadzie proste zagadnienie wiąże się z paroma uwagami. Szczegółowy opis parametrów i działania ograniczenia **Transformation** znajdziesz na str. 243.

Dystans przesunięcia popychacza — 3.5 jedn. Blendera — dopasowałem po kilku „przymiarkach”.

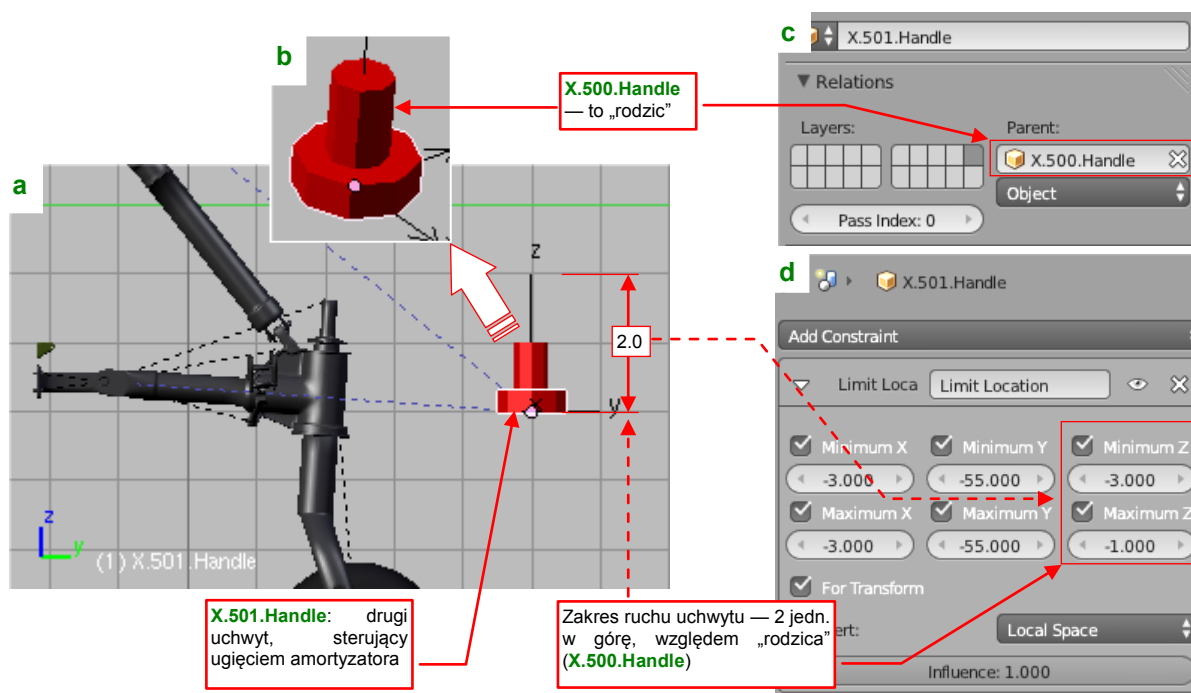
Uzyskaliśmy w ten sposób całkiem poprawny ruch mechanizmu podwozia (Rysunek 2.3.10):



Rysunek 2.3.10 Chowanie podwozia za pomocą uchwyty **X.500.Handle**.

Do tej pory ustawialiśmy koło w pozycji wysuniętej tak, jak wyglądało tuż przed lądowaniem — gdy amortyzator był zupełnie bez obciążenia. Jednak wtedy, gdy samolot stoi na ziemi, amortyzator „zsuwa” się trochę. Powodem jest to odchylenie wahacza.

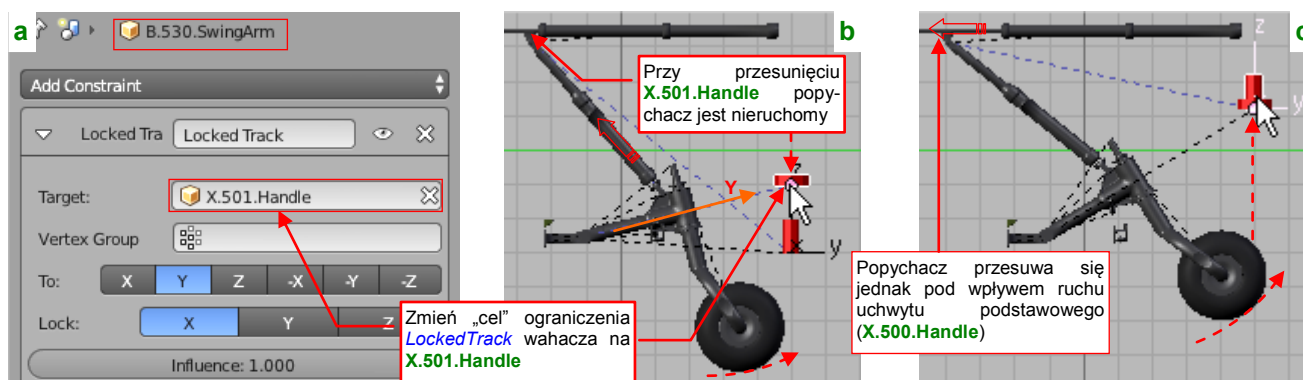
Aby odwzorować takie ugięcie podwozia, „nałóż” na uchwyt **X.500.Handle** drugi, mniejszy uchwyt (ale za to szerszy! — Rysunek 2.3.11b). Nadajmy mu nazwę **X.501.Handle** (Rysunek 2.3.11a):



Rysunek 2.3.11 Dodatkowy uchwyt, sterujący ugięciem amortyzatora (**X.501.Handle**).

Przypisz (relacją **Parent**) uchwyt **X.501.Handle** do podstawowego uchwyty (**X.500.Handle**) (Rysunek 2.3.11c). Dzięki temu uchwyt pomocniczy będzie „przyczepiony” do **X.500.Handle**, gdy zaczniesz nim poruszać, by schować podwozie. Aby ugięcie amortyzatora nie było „przegięciem”, dodaj do uchwyty **X.501.Handle** ograniczenie **Limit Location** (Rysunek 2.3.11d). Pozwól przesuwaniu mu się wyłącznie w pionie (wzdłuż lokalnej osi **Z**), na dystans 2 jednostki (od -3.0 do -1.0). (Po kilku przymiarkach zdecydowałem, że taki zakres będzie odpowiedni — powoduje „zsuniecie” amortyzatora o około 1 jednostkę Blendera. Odpowiada to 10 cm na rzeczywistym samolocie).

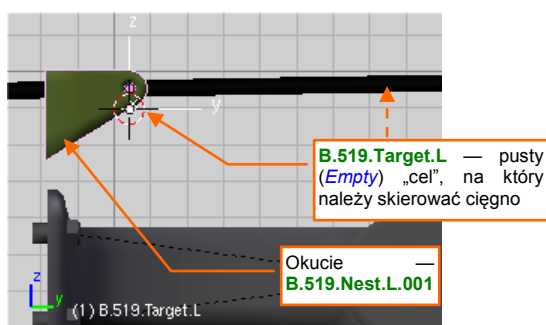
Teraz zmodyfikuj „cel” ograniczenia *Locked Track* wahacza (**B.530.SwingArm**). Wpisz tam, w miejsce nazwy uchwytu podstawowego, nazwę uchwytu pomocniczego (**X.501.Handle**) (Rysunek 2.3.12a):



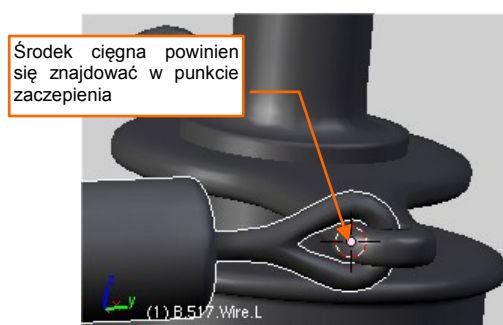
Rysunek 2.3.12 Sterowanie ugięciem amortyzatora.

Początkowo nie zmienia to niczego, gdyż środki obydwu uchwytów się pokrywają (por. Rysunek 2.3.11a). Gdy jednak zaczniesz przesuwając **X.501.Handle** do góry, zauważysz że ten ruch wsuwa cylinder amortyzatora — tak, jak gdyby był obciążony (Rysunek 2.3.12b). Popychacz zacznie się wysuwać dopiero wtedy, gdy przesuwniesz do góry uchwyt podstawowy (**X.500.Handle**) (Rysunek 2.3.12c). **X.501.Handle** będzie się poruszać wraz z nim (bo są związane relacją *Parent*). Musisz tylko pamiętać, by przed zamknięciem podwozia przesunąć uchwyt pomocniczy zupełnie do dołu (tj. ustawić koło w pozycji „swobodnej”).

Kolejnym elementem „do ograniczania” są cięgna. Zaczniij od wstawienia w każde z okuc — lewe i prawe — pomocniczych obiektów typu *Empty* (por. str. 218). Nadaj im nazwę **B.519.Target.L** i **.R** (Rysunek 2.3.13):



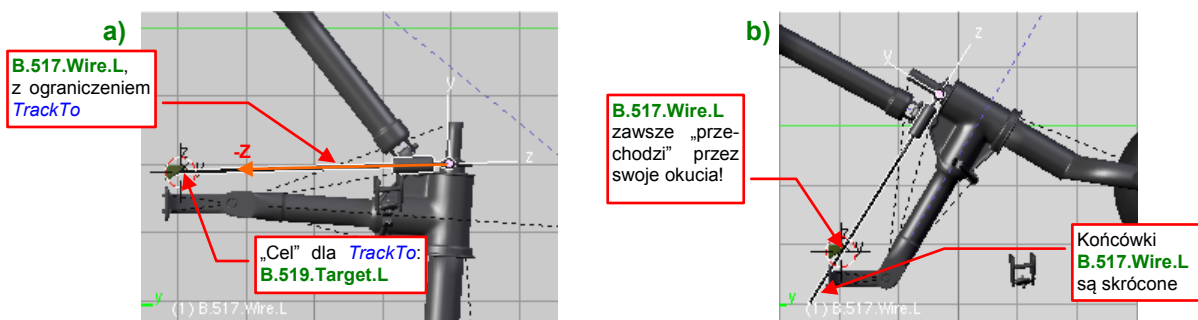
Rysunek 2.3.13 Pusty obiekt (*Empty*) — cel dla cięgna



Rysunek 2.3.14 Położenie środka cięgna

Jeżeli chodzi o strukturę modelu — przypisz (relacją *Parent*) każdy z tych „celów” do odpowiedniego okucia. (**B.519.Target.L** do **B.519.Nest.L.001**, a **B.519.Target.R** do **B.519.Nest.R.001**)

Środki cięgien powinny się znajdować w miejscu ich zaczepienia (Rysunek 2.3.14). Gdy już tam są, możesz do każdego cięgna dodać ograniczenie *Track To* (por. str. 235). Jako „cel” (*Target*) tych ograniczeń wpisz odpowiedni z obiektów **B.519.Target** (**.L** lub **.R**) (Rysunek 2.3.15a).



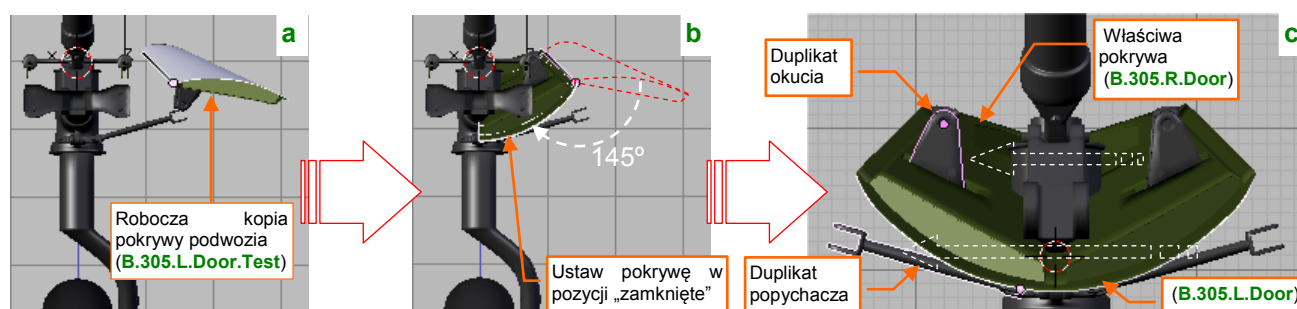
Rysunek 2.3.15 „Układanie się” cięgien podczas ruchu podwozia

- W przypadku cięgien należy stosować ograniczenie **Track To**, a nie **Locked Track**, gdyż mogą się obracać względem dwóch osi. (**Locked Track** imituje obrót wokół jednej osi)

Za pomocą ograniczeń **Track To** sprawiłeś, że w każdej pozycji podwozia cięgna będą „przechodzić” przez własne okucia. W przyszłości, aby podnieść realizm modelu, możesz przełożyć ruch steru kierunku na skręcenie goleni kółka ogonowego. (Oryginalnie cięgna **B.517.Wire** służyły właśnie do tego). W naszym modelu nie będzie to przyczyną, a skutkiem ruchu: w wyniku obrotu goleni obróci się przypisany do niej orczyk. To z kolei spowoduje niesymetryczne przesunięcie środków cięgien (jedno się cofnie, drugie przesunie do przodu). Obydwa zbliżą się nieznacznie do osi kadłuba (poruszają się po okręgu). Dzięki ograniczeniu **Track To**, nawet przy takim przemieszczeniu ich końce będą przechodzić przez okucia, przymocowane do wręgi kadłuba.

Na koniec warto skrócić końcówki tych drutów, aby przy całkowicie schowanym podwoziu nie „sterczały” z poszycia (Rysunek 2.3.15b). Na szczęście można dobrać ich długość tak, by jednocześnie przy zupełnie wysuniętym kole „dotykały” powierzchni wręgi. Kłopot może się zacząć, gdy obrócisz orczyk — wtedy końcówka jednego z nich stanie się widoczna! Może trzeba tu będzie potem „dosztukować” jakiś pomocniczy „drut”? Udałabyś dalszą część cięgna, ale poruszałaby się tylko poziomo. Byłby wysuwany tylko przy obrocie orczyka.

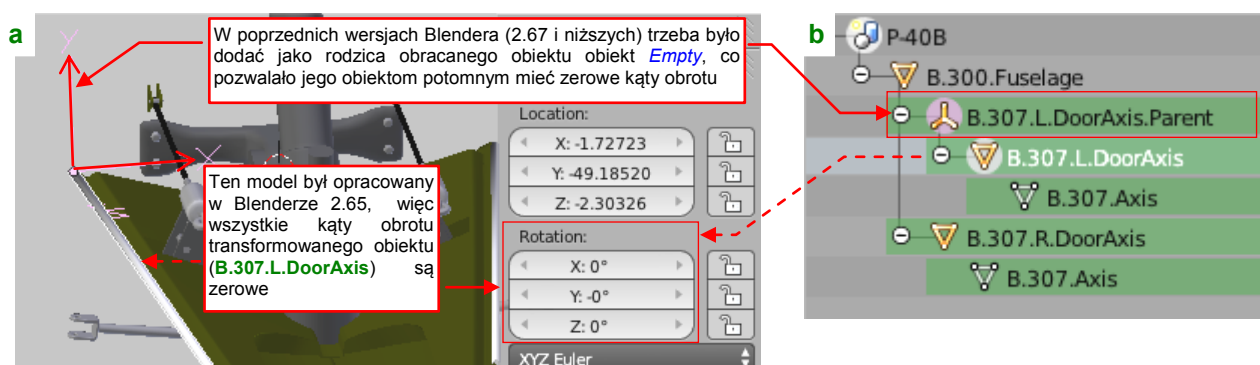
Ruch wahacza powodował w P-40 otwieranie lub zamykanie pokryw podwozia. Włączmy je więc w wirtualny mechanizm, który tu budujemy. W poprzedniej sekcji zamodelowaliśmy okucie i popychacz tylko dla lewej pokrywki. Umieściliśmy je na jej „roboczej” kopii — **B.305.L.Door.Test** (por. str. 47). Włącz teraz widoczność warstwy, na której znajduje się ten obiekt (Rysunek 2.3.16a):



Rysunek 2.3.16 Powielenie okuc i popychaczy pokryw podwozia

Obróć pokrywę wokół lokalnej osi **Z** jej zawiasów, tak, by przyjęła pozycję „zamkniętą”. W ten sposób okucie znalazło się w położeniu dopasowanym do „właściwych” pokryw: **B.305.L.Door** i **B.305.R.Door**. Tych pokryw nigdy dotąd nie ruszaliśmy z pozycji „zamkniętej”. Odlącz okucie od roboczej kopii (**Alt-P** — **Clear Parent**) i przypisz do pokrywki **B.305.L.Door**. Jego lustrzane odbicie przypisz do **B.305.R.Door**. (Opis tej operacji — jak stworzyć lustrzane odbicie okucia, jak przypisywać je do pokryw — znajdziesz na str. 257). Potem możesz już usunąć cały zespół roboczej pokrywki **B.305.L.Door.Test** — nie będziemy go więcej potrzebować.

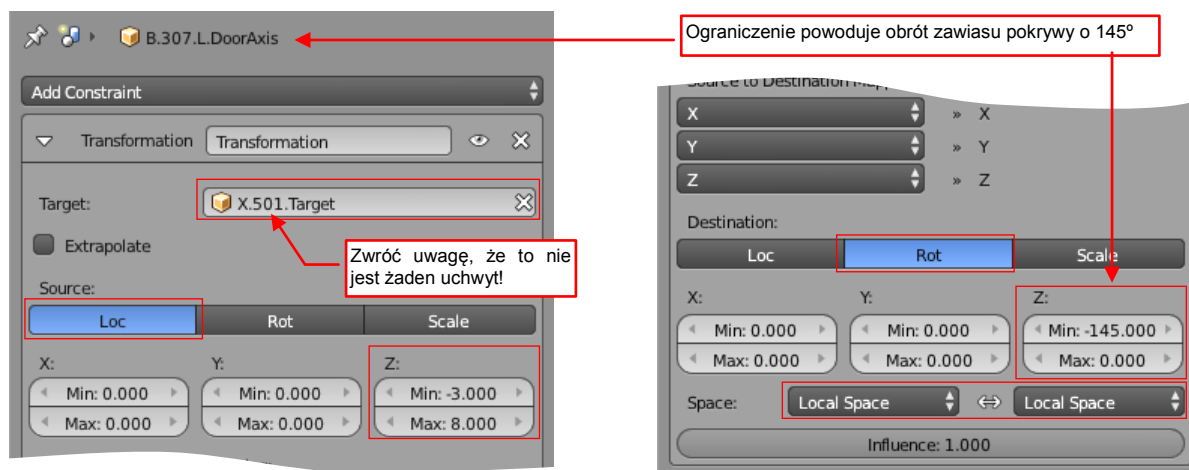
Ograniczenie **Transformation** w poprzednich wersjach Blendera (2.67 i wcześniejszych) miało pewien błąd pojawiający się przy obrocie. Aby nie wystąpił, musiałem tę oś poprzedzić obiektem **Empty** (Rysunek 2.3.17):



Rysunek 2.3.17 Pusty (**Empty**) rodzic zawiasu — obejście problemu, występującego w Blenderze 2.67 i wcześniejszych



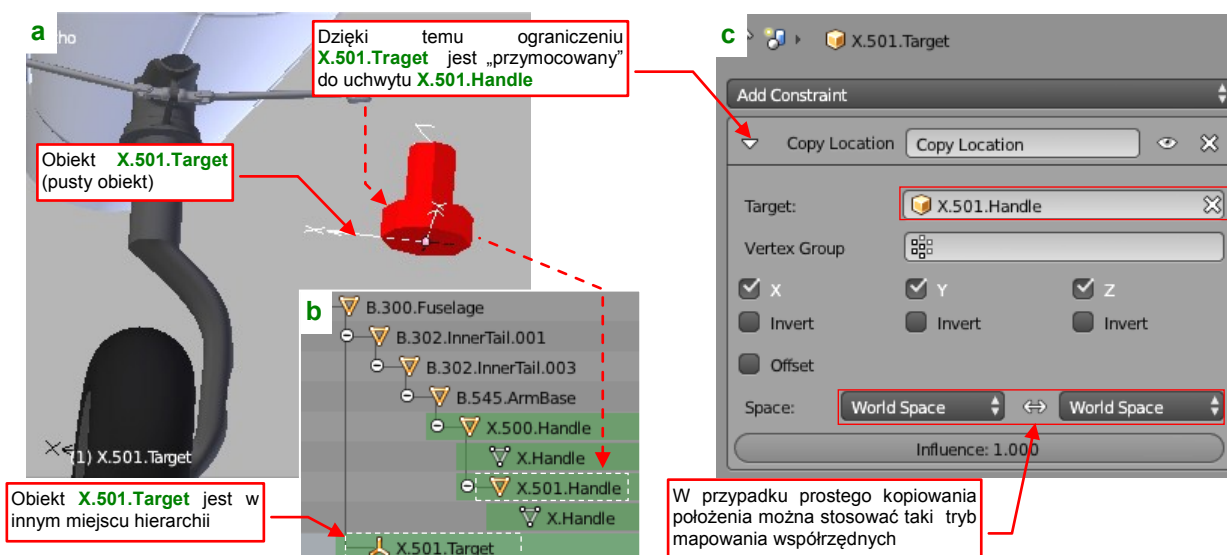
W Blenderze 2.67 i wcześniejszych obrót za pomocą ograniczenia **Transformation** wymagał wyzerowania kątów obrotu transformowanego obiektu. Tylko w ten sposób mogłem uzyskać poprawny obrót wokół jednej osi. Dlatego do prawego i lewego zawiasu dodałem nowego rodzica — obiekt **Empty** (por. Rysunek 2.3.17b). Aby to ułatwić stworzyłem polecenie **Add Empty Parent** (por. str. 247). Szczegółowe wyjaśnienie tej operacji znajdziesz na str. 245. Ustawienia ograniczenia wymuszającego obrót zawiasu przedstawia Rysunek 2.3.18:



Rysunek 2.3.18 Ograniczenie **Transformation**, przypisane do zawiasów pokrywy osłony kółka ogonowego

Zwróć uwagę na obiekt, wpisany w polu **Target** na panelu tego ograniczenia. Początkowo próbowałem uzależnić kąty obrotu zawiasów (**B.307.DoorAxis**) od przesunięcia uchwytu **X.501.Handle**. Okazuje się jednak, że przesunięcie opisane w sekcji **Transformation:Source** jest związane z przesunięciem obiektu względem jego bezpośredniego „rodzica”. W efekcie pokrywy przymykały się poprawnie wtedy, gdy uginałem amortyzator. Gdy poruszałem jego „rodzicem” — podstawowym uchwytom **X.500.Handle** — drzwi komory pozostawały nieruchome.

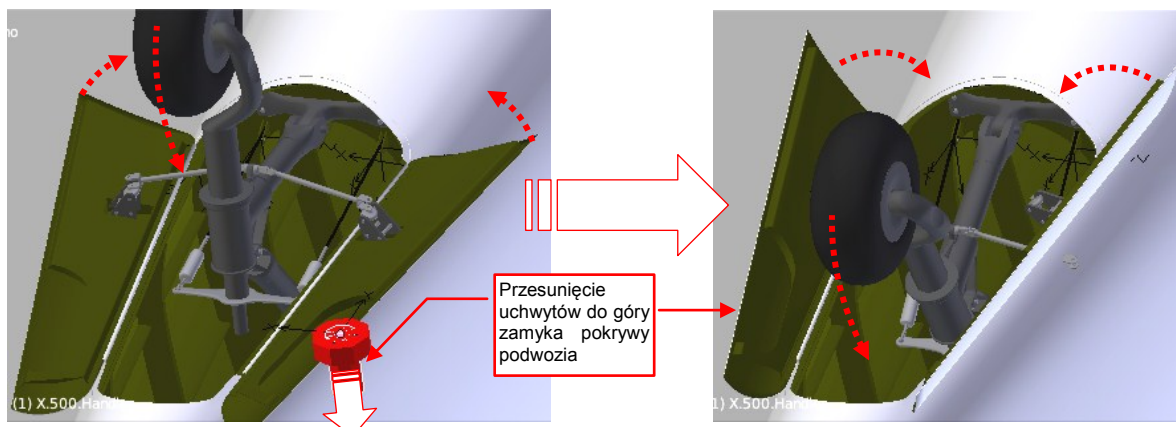
Aby obejść ten problem, musiałem stworzyć dodatkowy obiekt (typu **Empty**) który podąża za jakimkolwiek ruchem każdego z dwóch uchwytów (Rysunek 2.3.19a):



Rysunek 2.3.19 **X.501.Target** — pomocniczy obiekt sterujący obrotem pokryw podwozia

Nadałem mu nazwę **X.501.Target**. W hierarchii modelu zajmuje inne miejsce (por. Rysunek 2.3.19b). Aby jednak „przymocować” go do obydwu uchwytów, powiązałem go z uchwytom **X.501.Handle** ograniczeniem **Copy Location** (por. Rysunek 2.3.19c). Dzięki **Copy Location** obiekt **X.501.Target** podąża za każdym przesunięciem uchwytów. I dlatego to ten obiekt steruje obrotem zawiasów.

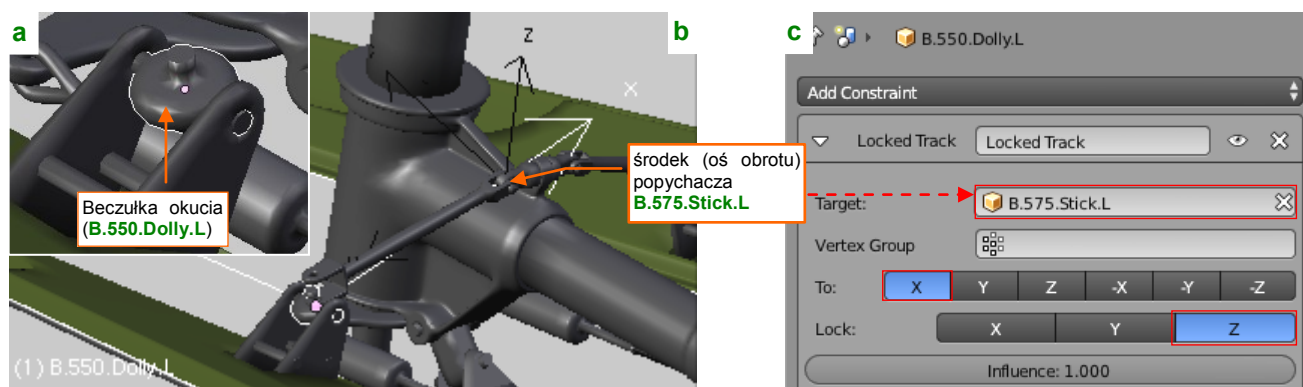
Tak przygotowane pokrywy zamykają się, gdy przesuniesz główny uchwyt do góry (Rysunek 2.3.20):



Rysunek 2.3.20 Chowanie kółka ogonowego

Pokrywy reagują także na ugięcie amortyzatora, wywołane przesunięciem pomocniczego uchwytu **X.501.Handle**.

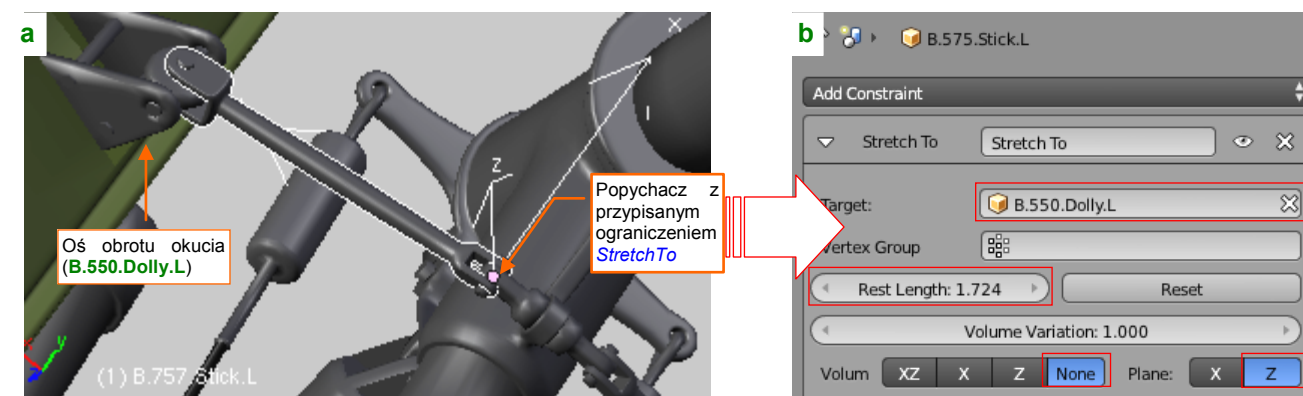
Ostatnim elementem, który pozostał nam do mechanizacji, są popychacze pokryw podwozia. Zaczniemy od beczulek okucia, do których są przymocowane (Rysunek 2.3.21a):



Rysunek 2.3.21 Skierowanie beczulki okucia na oś popychacza (**Locked Track**)

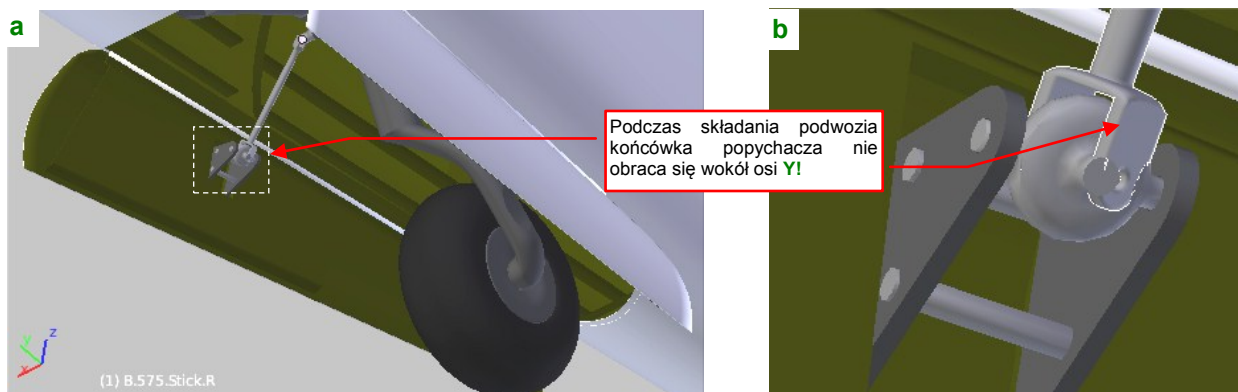
Przypisz beczułkom ograniczenia **Locked Track**. Skieruj ich osie **X** na środki (**origin**) popychaczy (Rysunek 2.3.21c). (Te punkty znajdują się na wewnętrznych krańcach popychaczy — por. Rysunek 2.3.21b).

Jednocześnie skieruj osie **Y** popychaczy na beczułki okuć (**B.550.Dolly.L** lub **.R**). Tutaj jednak nie posłużymy się ograniczeniem **Locked Track**. Gdybyśmy to zrobili, to pomiędzy końcami popychaczy i beczułkami pojawiałyby się wyraźne odstępy (por. str. 249). (Jest to rezultat przybliżenia rzeczywistego ruchu pokryw podwozia za pomocą zwykłego obrotu). Zamiast tego wykorzystamy inny rodzaj ograniczenia: **Stretch To** (Rysunek 2.3.22):



Rysunek 2.3.22 Dopasowanie popychacza do okucia (**Stretch To**)

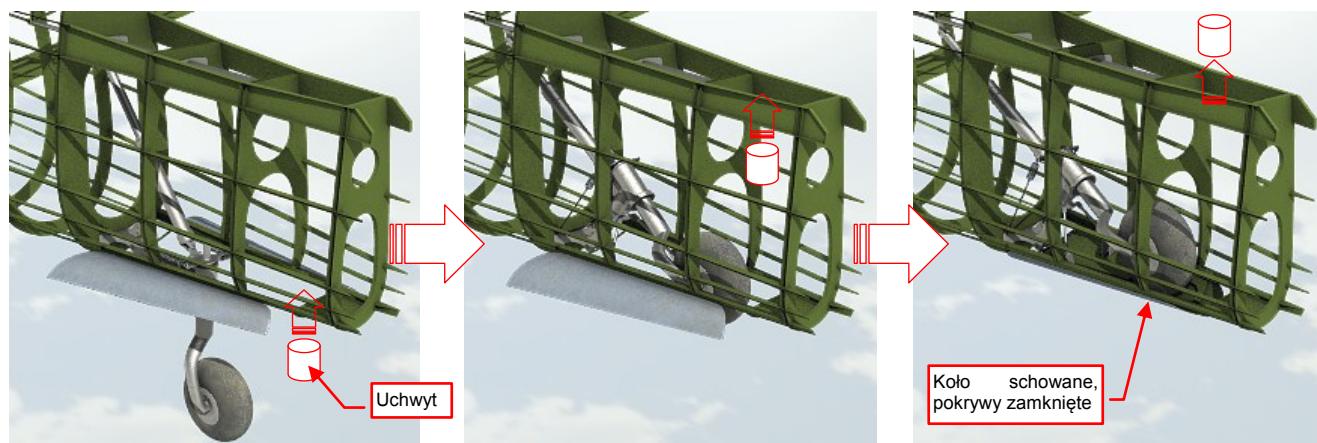
Szczegółowe informacje o działaniu i parametrach tego ograniczenia znajdziesz na str. 249. Dzięki **Stretch To** końce popychaczy będą się trzymać beczulek „jak przyklejone”. Co prawda, ograniczenie **Stretch To** nie obsłuży nam wszystkich niuansów ruchu prawdziwego popychacza. Rysunek 2.3.23 pokazuje, jak skręca się jego końcówka podczas chowania podwozia:



Rysunek 2.3.23 Skręcona końcówka popychacza podczas chowania podwozia

W dodatku w niektórych stadiach składania kółka ogonowego popychacze rozciągają się jak gdyby były z gumy. To kompromis, z którym musiałem się pogodzić<sup>1</sup>. Najważniejsze jest, by podwozie wyglądało poprawnie w pozycji „otwarte”. W pozycji „zamknięte” nie powinno razić, choć może odbiegać od oryginału w niewielkich szczegółach (nikt pod poszycie nie będzie zaglądał).

Rysunek 2.3.24 przedstawia ostateczne działanie mechanizmu, który stworzyliśmy w tej sekcji:



Rysunek 2.3.24 Składanie kółka ogonowego — jednym przesunięciem uchwytu!

<sup>1</sup> Aby końcówka popychacza **B.575.Stick** zawsze poprawnie przylegała do beczulki **B.550.Dolly**, musisz podzielić popychacz pośrodku długości na dwie części, skierowane na siebie jak tłok i cylinder amortyzatora. Przypisz część wewnętrzną relacją **Parent** do wahacza **B.530.SwingArm**, a zewnętrzną — do beczulki pokryw. Skieruj obydwie części na siebie: zewnętrzną ograniczeniem **Locked Track**, a wewnętrzną — **Stretch To**. Dzięki temu że w miejscu połączenia popychacz jest okrągły, nie będzie widać wzajemnego obrotu tych części. Są to jednak tak małe części, że zrezygnowałem z tych szczegółów...

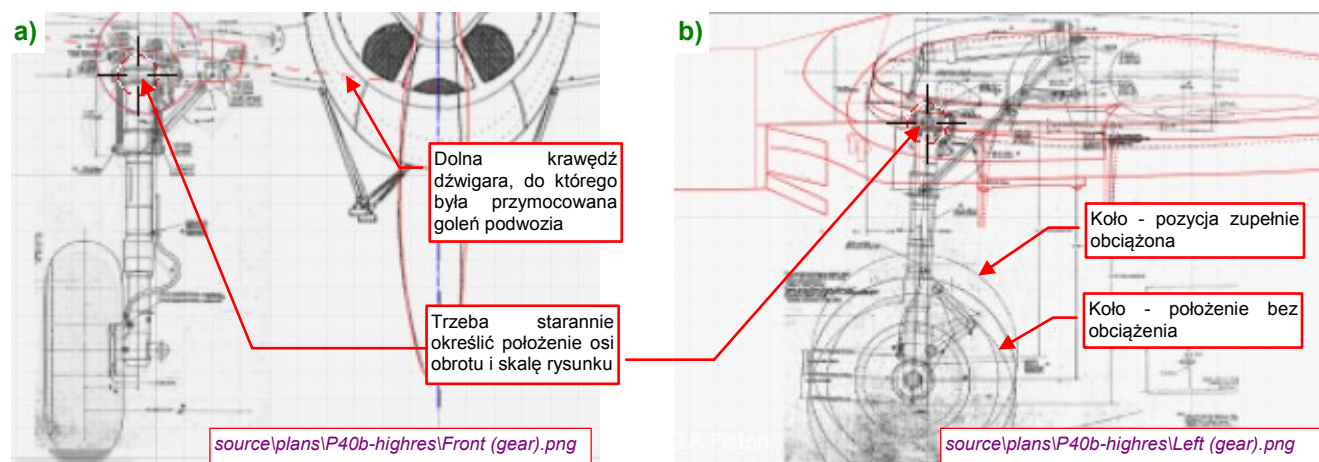
### Podsumownie

- „Mechanizacja” opisana w tej sekcji polega na połączeniu poszczególnych części modelu w jedną całość (mechanizm). Chodzi o to, by można było łatwo uruchomić model jakiegoś zespołu — np. podwozia — za pomocą jednego przesunięcia jakiegoś pomocniczego uchwytu (str. 51);
- Ruch i wzajemne zależności części mechanizmu można najwierniej odwzorować za pomocą tzw. armatury (*Armature*), zawierającej zespół tzw. „kości” (*bones*). Niestety, w przypadku takich elementów jak podwozie, próba zastosowania kości może się wiązać z poważnymi trudnościami (str. 51);
- W tej sekcji przedstawiam metodę alternatywną: zamiast używać armatury i jej kości, można zbudować cały „wirtualny” mechanizm przypisując elementom zespołu różnego typu ograniczenia ruchu (*constraints*);
- Pracę zaczynamy od stworzenia na warstwie **10** łatwo dostępnego „uchwytu”. Ruch tej części będzie poruszał całym mechanizmem (str. 52);
- Podstawowym ograniczeniem, imitującym zawiasy i inne połączenia osiowe, jest *Locked Track* (str. 52). Przed zastosowaniem tego ograniczenia warto jest zgrubnie ustawić osie obiektu w odpowiednim kierunku — by zastosowanie ograniczenia zmieniło je tylko trochę (str. 59);
- Ograniczenie *Limit Location* pozwala ustalić zakres ruchu uchwytu — a w rezultacie całego mechanizmu — do zakresu położenia odpowiadającego rzeczywistości (str. 53);
- Ograniczenie *Transform* można traktować jako „uniwersalną przekładnię” ruchu jednego przedmiotu na inny. Można jej użyć, by powiązać obrót, czy nawet zmianę skali, z przesunięciem jakiegoś wzorcowego obiektu (str. 54). Obroty obiektu docelowego najlepiej jest ograniczyć do jednej osi — tak jak to zrobiliśmy dla zawiasów pokryw podwozia (str. 57);
- „Zsuwający” się amortyzator należy zamodelować z dwóch części: cylindra i tłoka. Każda z nich powinna mieć środek (*origin*) na przeciwległym krańcu, i innego „rodzica” (*parent*). Tłok powinien być „celem” (*Target*) ograniczenia *Locked Track* cylindra, a cylinder — tłoka (str. 53);
- Ugięcie amortyzatora pod obciążeniem jest osobnym ruchem mechanizmu. Dlatego można je uzależnić od dodatkowego uchwytu (str. 56);
- Ograniczenia *Track To* należy użyć do obiektów, które mogą się obracać w dwóch kierunkach — na przykład cięgien (str. 56);
- Ograniczenie *Stretch To* można użyć do „przyłączenia” do mechanizmu popychaczy. Unikniemy w ten sposób „odrywania” ich końcówek od okuć, występującego w niektórych pośrednich pozycjach mechanizmu podwozia (str. 59);



## 2.4 Podwozie główne

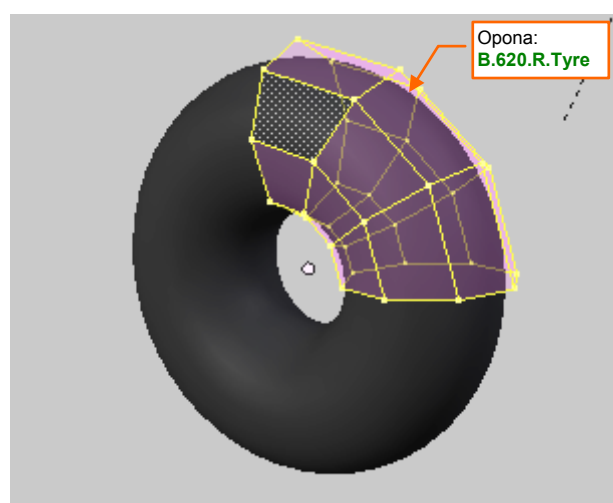
Jak przekonałeś się w poprzednich sekcjach, podwozie składa się z dziesiątek drobnych elementów. Zazwyczaj są one zbyt małe, by zostać prawidłowo nakreślone na planach modelarskich. O ile autor nie załączył jakiegoś szczegółowego rysunku, podczas wykonywania podwozia jesteś skazany na analizę dostępnych zdjęć. W przypadku P-40 jesteśmy w tej szczęśliwej sytuacji, że oprócz udostępnionej w Internecie dużej liczby fotografii, można znaleźć także oryginalny rysunek złożeniowy podwozia głównego (widok z przodu i z boku). Pozostaje wkomponować go w rzuty samolotu (Rysunek 2.4.1):



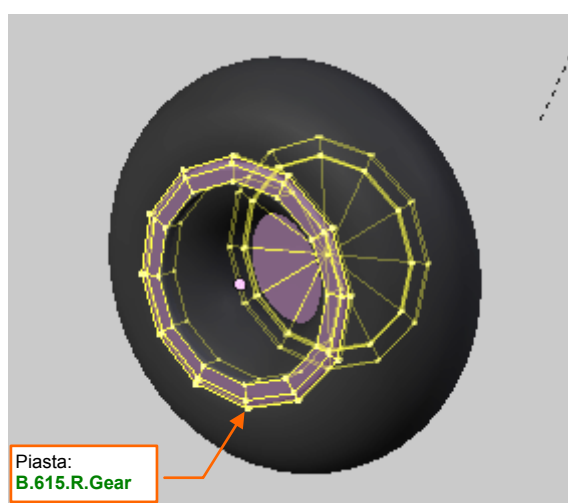
Rysunek 2.4.1 Fabryczne rysunki zespołu koła głównego

Oryginalny rysunek fabryczny podzieliłem na rzut z przodu i rzut z boku. Dopasowywanie zacząłem od rzutu z przodu, gdyż zawiera zarys dolnej krawędzi przedniego dźwigara. (Było do niej przymocowane okucie osi obrotu goleni). Na siatce skrzydła modelu zidentyfikowałem podłużnicę, która odpowiada tej krawędzi, i przenieśliem jej zarys na plany (Rysunek 2.4.1a). W oparciu o tę linię, można było poprawnie „zawiesić” podwozie pod skrzydłem. Potem dla tej samej współrzędnej **Z** i skali ustawiłem rzut z boku (Rysunek 2.4.1b). Aby nie wierzyć ślepo planom, położenie goleni wzdłuż osi samolotu (ich współrzędną **Y**) ustaliłem na podstawie zdjęć.

Modelowanie podwozia zaczynam od opony. Wykonałem ją tą samą techniką, co kółko ogonowe — jako „ćwiartkę” torusa, odbitą modyfikatorem *Mirror* względem dwóch osi (Rysunek 2.4.2).



Rysunek 2.4.2 Siatka opony

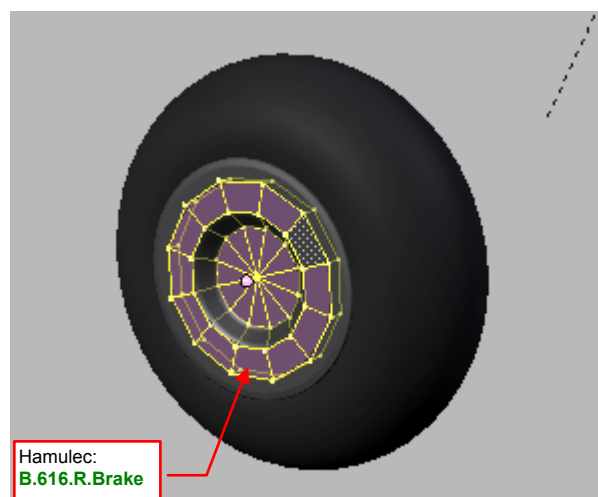


Rysunek 2.4.3 Siatka piasty koła

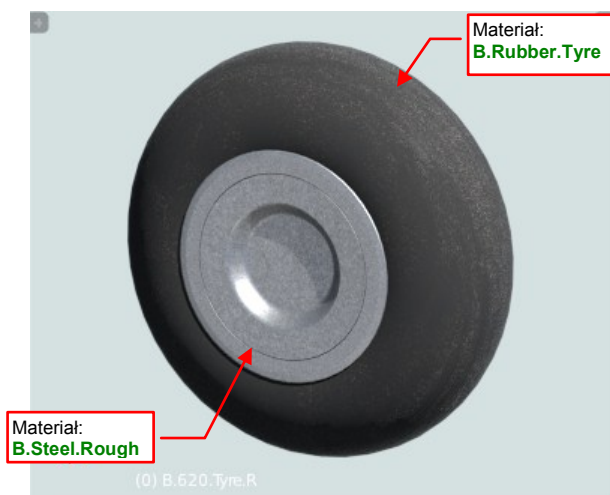
Ze zdjęć wynika, że opony wczesnych wersji P-40 zupełnie nie miały bieżnika. W razie czego — bieżnik można zamodelować lub odwzorować za pomocą tekstury nierówności. Podczas wykonywania piasty koła nie skorzystałem z modyfikatora *Mirror* (Rysunek 2.4.3). Tarcze piast były często na P-40 malowane w fantazyjne wzory, więc jako jedyny element podwozia jej siatka zostanie rozwinięta w płaszczyźnie UV.



W bardzo podobny sposób wykonałem tarczę hamulca (Rysunek 2.4.4). W odróżnieniu od piasty była „na sztywno” przymocowana do osi koła, więc musi być odrębnym obiektem.



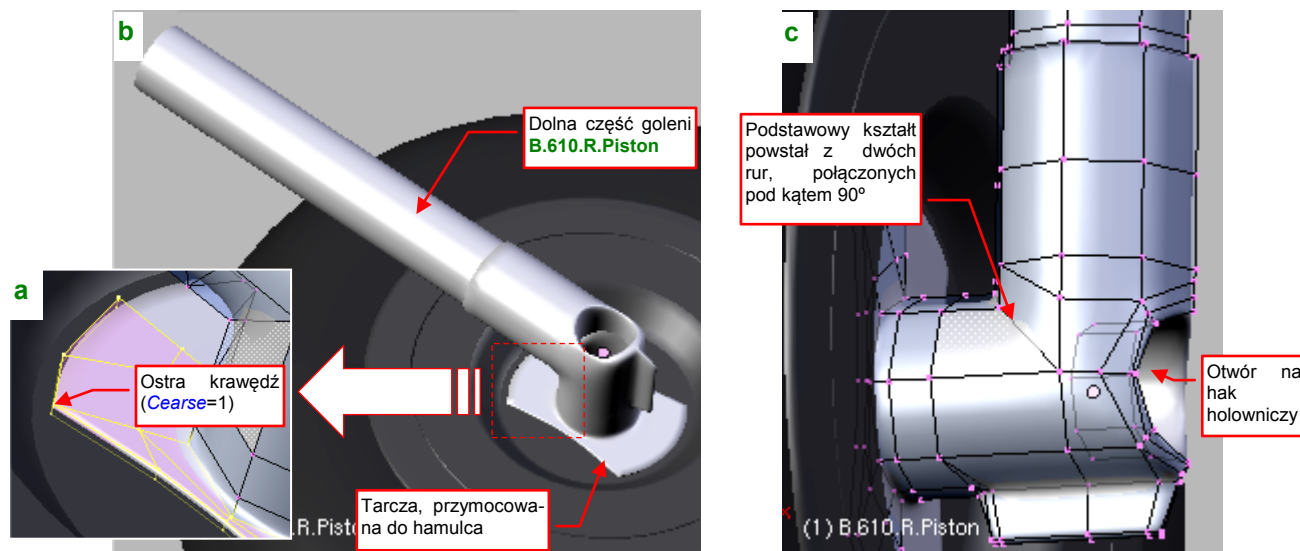
Rysunek 2.4.4 Siatka tarczy hamulcowej



Rysunek 2.4.5 Koło główne — obiekty i ich materiały

Nadałem tej oponie nazwę **B.620.R.Tyre**, piaście — **B.615.R.Gear**, a tarczy hamulca — **B.616.R.Brake**. Zbudowałem z nich prostą hierarchię, przypisując oponę do piasty, a piastę do hamulca. Zaraz po skończeniu modelowania warto także przypisać poszczególnym siatkom odpowiednie materiały, by już do nich nie wracać. Przy okazji pracy nad kółkiem ogonowym „dorobiliśmy” się już odpowiedniego zestawu wykończeń. Przydzieliłem oponie koła głównego ten sam materiał, jaki wykorzystaliśmy na oponę kółka ogonowego: **B.Rubber.Tyre**. Pozostałym elementom przypisałem typową powłokę metalu, czyli **B.Steel.Rough** (Rysunek 2.4.5).

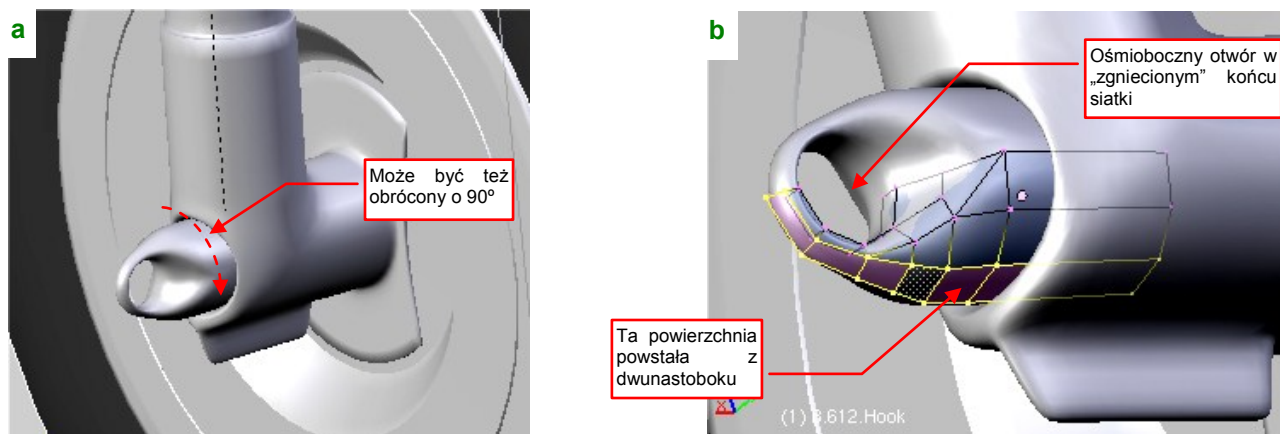
Teraz wykonamy dolną część goleni, pełniącą jednocześnie rolę tłoka amortyzatora (Rysunek 2.4.6):



Rysunek 2.4.6 Dolna część goleni — tłok amortyzatora

Podstawowy kształt tej siatki powstał ze złożenia pod kątem  $90^\circ$  dwóch rur (Rysunek 2.4.6c). Następnie wyznaczyłem na tylnej ścianie okrągłą krawędź otworu na zaczep (hak) holowniczy. (Zrobiłem to za pomocą dodatku *Intersection* — por. str. 221). W oparciu o tę krawędź wkomponowałem w siatkę goleni odpowiedni otwór. Z przeciwnej strony dodałem do siatki okrągłą tarczę, do której w prawdziwym samolocie był przykręcony zespół hamulca. Przy odpowiednim przesunięciu i zagęszczeniu wierzchołków, udało mi się przekształcić ten ośmiokąt w kształt przypominający klepsydrę (Rysunek 2.4.6a). Oczywiście, uzyskanie takich ostrych krawędzi było tylko możliwe poprzez zwiększenie ich wartości *Cearese* do 1.0. Zamodelowałem tylko jedną stronę siatki — reszta jest wygenerowana przez modyfikator *Mirror*. Nadałem tej części goleni nazwę **B.610.R.Piston** i ustaliłem materiał na **B.Steel.Rough**. W hierarchii modelu ta goleń jest „rodzicem” tarczy hamulca.

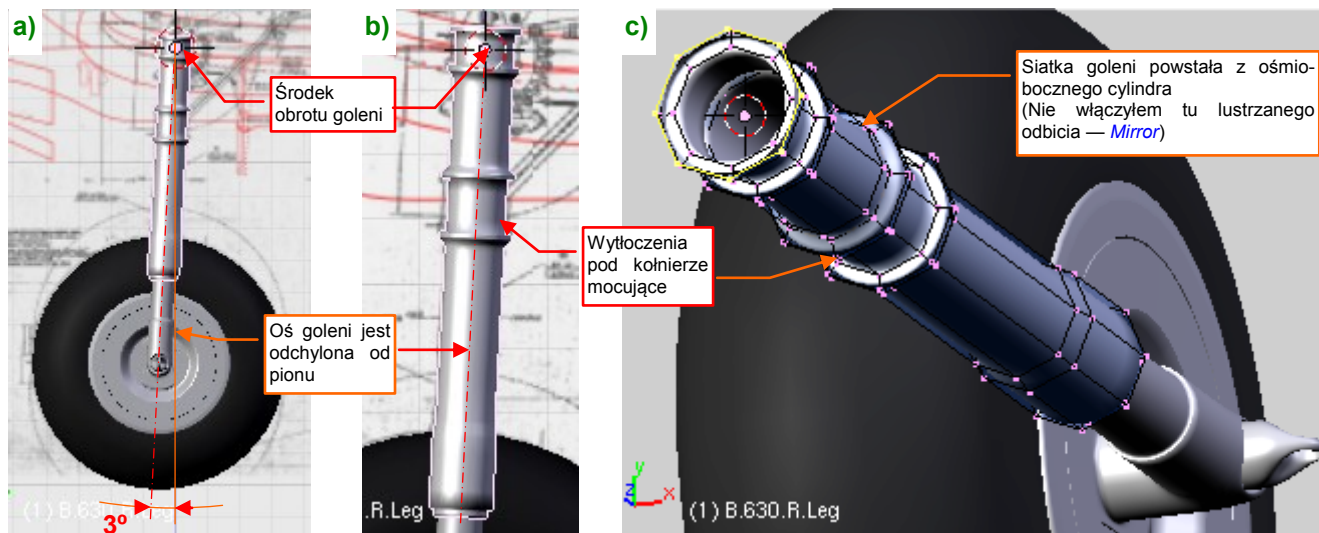
Kolejnym fragmentem podwozia jest zaczep (hak) holowniczy (Rysunek 2.4.7):



Rysunek 2.4.7 Hak do holowania

Według rysunków fabrycznych, hak powinien być ustawiony pionowo (Rysunek 2.4.7a), choć na zdjęciach odręstaurowanych samolotów najczęściej jest obrócony o 90°. (W tym położeniu najprawdopodobniej łatwiej jest przymocować do każdego z kół końcówkę jakiegoś pomocniczego „zaprzęgu” holowniczego). Siatka tej części to dwunastoboczny cylinder, któremu „zgniotłem” końcówkę, wkomponowując w nią otwór, oparty na ośmioboku (Rysunek 2.4.7b). Aby się nie przemęczać, modelowałem tylko „ćwiartkę” tej powierzchni — reszta to rezultat użycia modyfikatora *Mirror*, generującego odbicie względem dwóch osi. Nadałem tej części nazwę **B.612.Hook**, i umieściłem w hierarchii obiektów jako „potomka” goleni (**B.610.R.Piston**). Tak jak pozostałym elementom podwozia, przypisałem do siatki **B.612.Hook** materiał **B.Steel.Rough**.

Jak się powiedziało „A”, trzeba powiedzieć i „B”, więc skoro mamy dolną część goleni (tłok amortyzatora), dodajmy zaraz część górną, zawierającą jego cylinder (Rysunek 2.4.8):

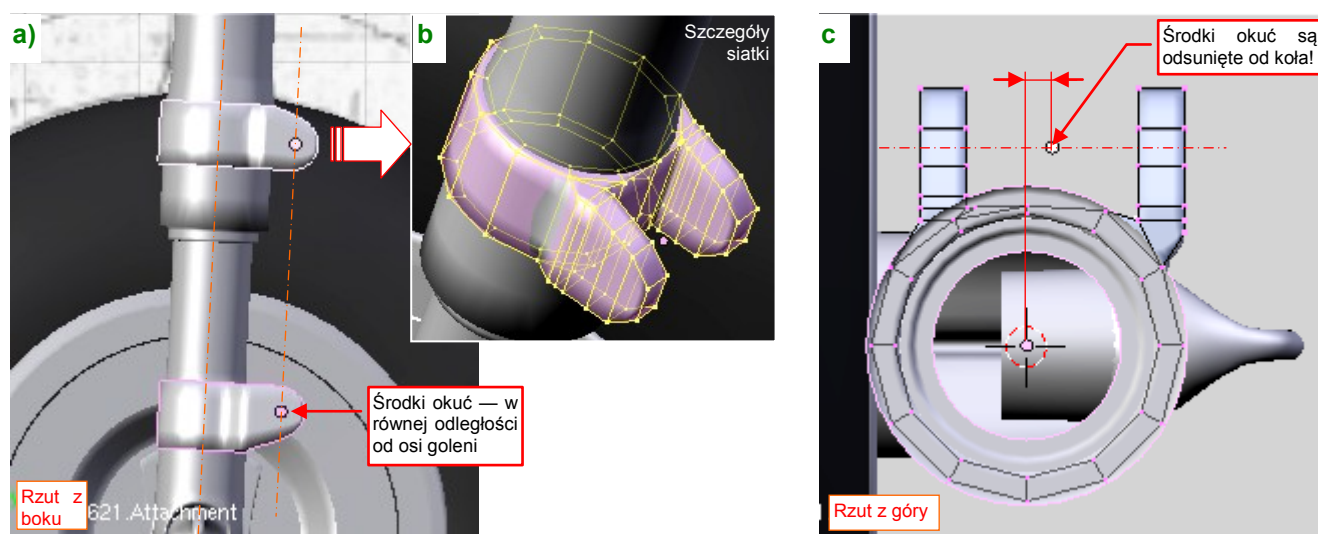


Rysunek 2.4.8 Druga część goleni — cylinder amortyzatora

Cylinder amortyzatora stanowił integralny fragment górnej części goleni podwozia. Siatka tego elementu powstała z ośmioboku (Rysunek 2.4.8c). To stosunkowo prosty kształt, jak z tokarki. Nie redukowałem tu liczby oryginalnych ścian za pomocą modyfikatora *Mirror* — bo z jednej strony ta siatka będzie miała „wypustkę” (por. str. 70). Goleń podwozia P-40 ma lekko pochyloną oś. Podczas pracy będziesz wielokrotnie ustawiał ją na chwilę do pionu, więc warto ustalić jej pochYLENIE na jakiś określony kąt, by łatwo było w to położenie wrócić. Po sprawdzeniu planów i zdjęć, zdecydowałem się na pochYLENIE osi goleni dokładnie o 3° (Rysunek 2.4.8a).

Podczas składania podwozia goleń obraca się wokół swojej osi o ponad 90°. Jest utrzymywana przez specjalny kołnierz. Stąd na powierzchni **B.630.R.Leg** przygotowałem małe wytłoczenia dla tego elementu.

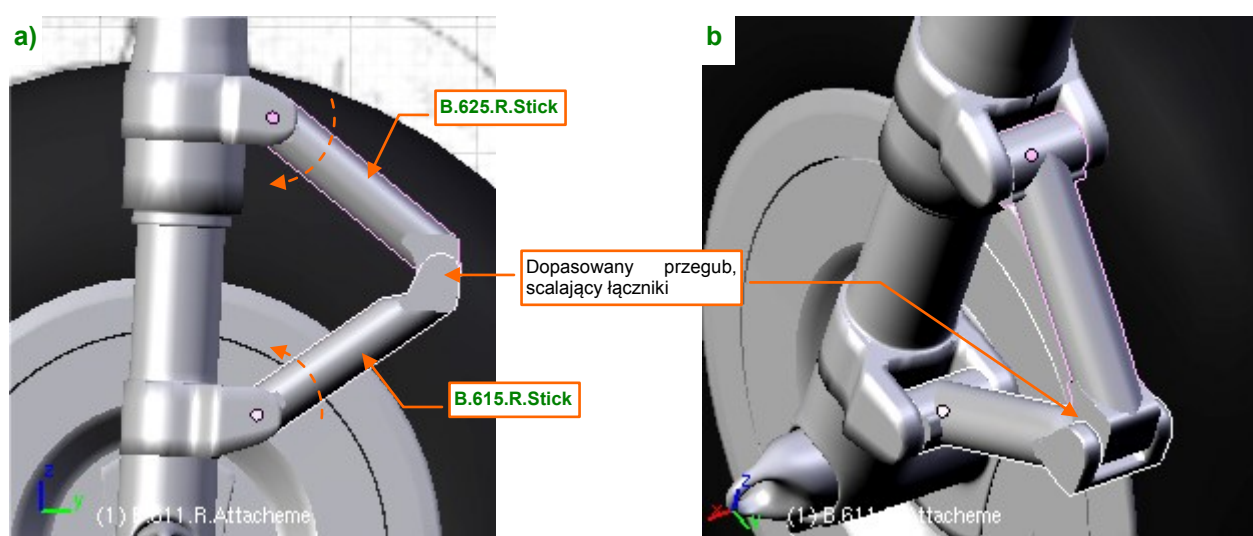
Aby zapobiec przypadkowemu obrotowi, cylinder i tłok amortyzatora są powiązane za pomocą dwóch przegubowych łączników (Rysunek 2.4.10). Modelowanie tego zespołu zaczniemy od ich okuć (Rysunek 2.4.9):



Rysunek 2.4.9 Okucia łączników

Okucia łączników w P-40 wyglądają jak dwa paski, wygięte wokół goleni (Rysunek 2.4.9a). Wydaje mi się, że takie rozwiązanie znajdziesz na większości samolotów z okresu II Wojny Światowej. (Technologicznie nie opłacało się wykonać „w jednym kawałku” goleni razem z okuciami). Rysunek 2.4.9b) pokazuje szczegóły siatki okuć — to w zasadzie cylinder z dołączonymi „uszami” na oś obrotu łącznika. Świadomie zrezygnowałem tu z modelowania otworu osi (por. np. 44, Rysunek 2.2.18) — bo i tak jedna z jego ścian zostanie zupełnie zasłonięta przez łeb śruby a druga — przez nakrętkę. Zwróć uwagę, że „uszy” okuć są odsunięte w bok od koła (Rysunek 2.4.9c). Tak wynika ze zdjęć (gdyby płaszczyzna symetrii łączników leżała na osi goleni, wówczas nakrętka śruby zawadzałaby o oponę). Środki okuć umieściłem w osiach obrotu łączników (są równoległe, ale odsunięte od goleni — por. Rysunek 2.4.9a). Takie położenie będzie w przyszłości bardzo przydatne do zbudowania poprawnego „mechanizmu” dla tego zespołu. Górnemu okuciu nadałem nazwę **B.621.R.Attachment**, a dolnemu — **B.611.Attachment**.

Gdy mamy już w czym „osadzić” łączniki, można je teraz wykonać (Rysunek 2.4.10):

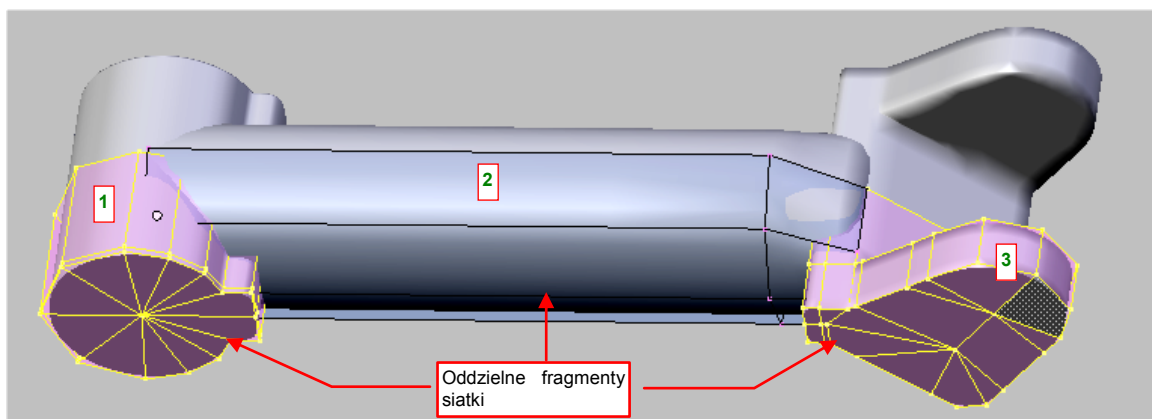


Rysunek 2.4.10 Łączniki tłoka i cylindra amortyzatora

Środki łączników znajdują się w tym samym punkcie, co środki ich okuć (por. Rysunek 2.4.10a i Rysunek 2.4.9a). Łączniki nie są identyczne — dolny jest zakończony „widelcem”, który zachodzi na końcówkę górnego (Rysunek 2.4.10b).



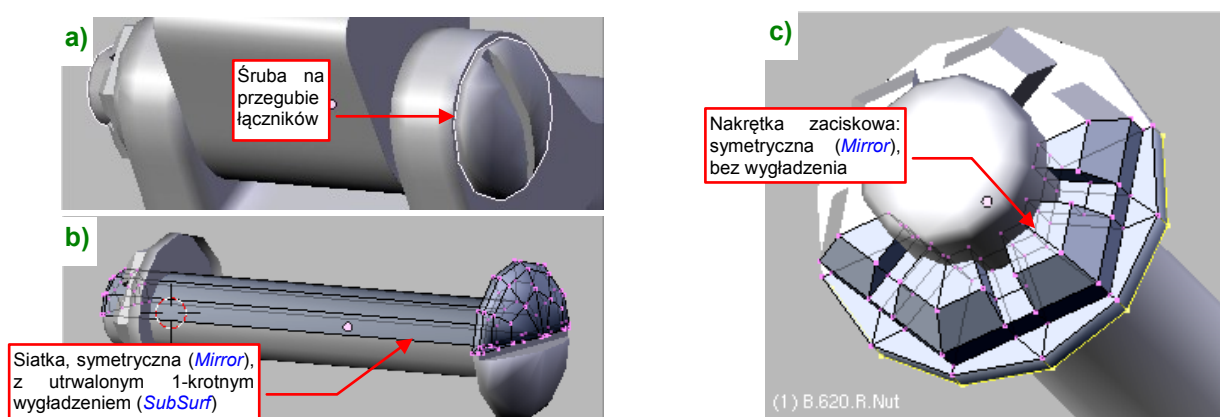
Sam łącznik wygląda na solidną, krótką sztangę. Jest na tyle drobny, że tworząc jego siatkę zrezygnowałem z pracochłonnego łączenia poszczególnych zespołów (Rysunek 2.4.11):



Rysunek 2.4.11 Szczegóły siatki łącznika amortyzatora

Siatka łącznika powstała ze złożenia trzech oddzielnych powłok: cylindra wokół osi okucia (1), właściwego ramienia (2), oraz widelca (3) (w górnym łączniku ten fragment ma inny kształt). Mimo ostrych krawędzi przenikających się powierzchni, element w **Object Mode** i na renderze wygląda to całkiem normalnie. Trzeba tylko dokładnie dosunąć do siebie wszystkie trzy powłoki, by nigdzie nie było widać jakichś szczelin.

Wreszcie doszliśmy do śrub (Rysunek 2.4.12a). Tak, tak, zdecydowałem się zamodelować co większe egzemplarze, choć sam wcześniej ostrzegałem przed popadaniem w przesadę. (I zamiast modelować np. śruby mocujące owiewki połączeń skrzydła i kadłuba, odwzorowałem je za pomocą tekstury nierówności — por. Wyciąg III). Tym razem jednak robię to z całą premedytacją. Po pierwsze — śruby, wykorzystane do połączenia elementów goleni miały większy rozmiar od śrub używanych do mocowania blach płatownia. Po drugie — do modelowania podwozia stosujemy tylko tekstury proceduralne, więc nie ma gdzie narysować odpowiednich łbów i nakrętek. Po trzecie — aby poprawnie „zmechanizować” ruch łączników towarzyszący ugięciu amortyzatora, potrzebujemy oddzielnego obiektu w każdej z trzech osi obrotu tego zespołu. Po co wstawiać w to miejsce jakieś „puste” (**Empty**) twory, skoro możemy w naturalny sposób skorzystać ze śrub, umieszczonych tam przez konstruktora? Siatkę śruby wygenerowałem z połówki cylindra, z włączonym modyfikatorem **Mirror** (Rysunek 2.4.12b):

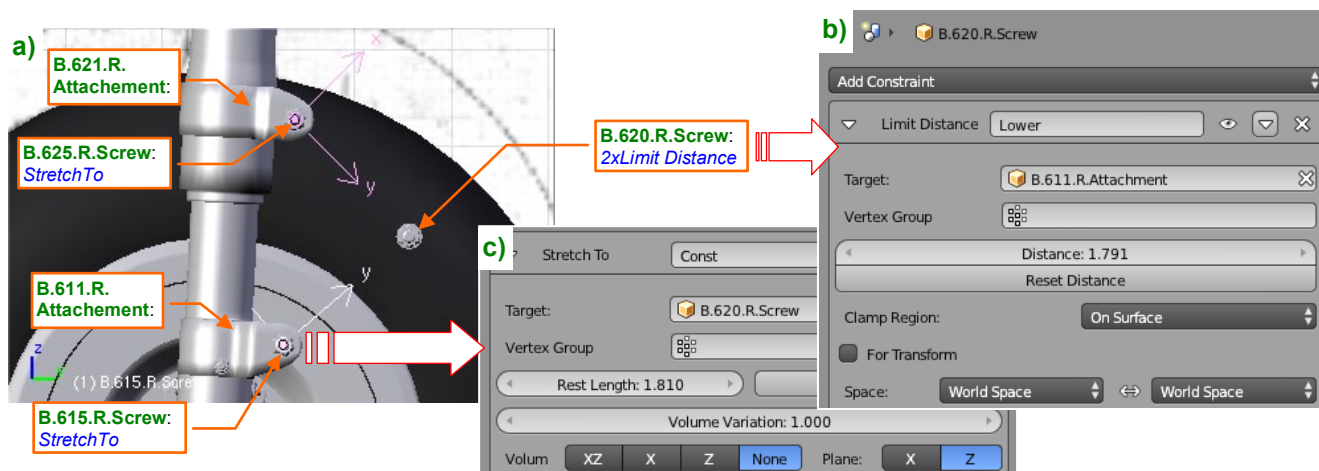


Rysunek 2.4.12 Śruba i nakrętka

Kulisty łeb śruby wytłoczyłem z końcówki walca. Po włączeniu umiarkowanego wygładzenia (modyfikator **Subsurf** z **Subdivisions** = 1) utrwaliłem je (**Apply**). Zrobiłem to po to, by wygodnie naciąć we łbie szczelinę. Śruby wykorzystamy w wielu miejscach podwozia, więc przygotowałem kilka alternatywnych siatek, różniących się długością gwintu (**B.Screw.Flat.008**, **B.Screw.Flat.010**, **B.Screw.Flat.011** — końcówki oznaczają długość, w 1/10 jednostki Blendera). Zastosujemy je w wielu klonach. Nakrętka zaciskowa („bezpieczna” — **B.Nut.Safe**) ma siatkę na bazie dwunastoboku, bez żadnych wygładzeń (Rysunek 2.4.12c).

Nie wiem, czy w oryginalnym P-40 stosowano takie zaciskowe nakrętki, jakie pokazuje Rysunek 2.4.12c). Zamodelowałem ją na podstawie zdjęć jednego z odrestaurowanych P-40E.

Śruby umieściłem w osiach ramion łącznika. Same ramiona można na chwilę ukryć (**H** — *Hide*). Przypisałem im takie ograniczenia, jakie pokazuje Rysunek 2.4.13:

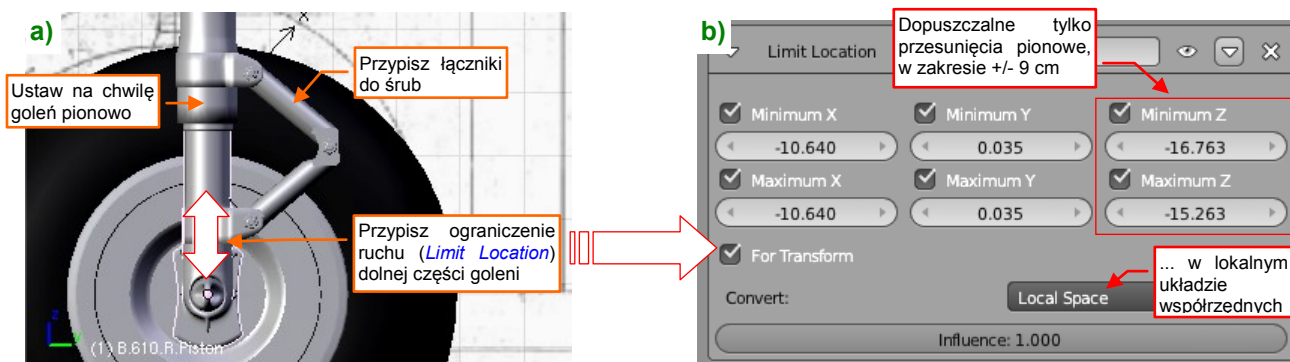


Rysunek 2.4.13 Budowa mechanizmu przegubowego połączenia goleni.

Śrubie umieszczonej w osi przegubu nadałem nazwę **B.620.R.Screw** (Rysunek 2.4.13a). Przypisałem jej dwa ograniczenia *Limit Distance* (Rysunek 2.4.13b). Każde z nich wymusza utrzymanie przez ten obiekt stałej odległości (*Clamp Region* = *Surface*) od środków okuć (leżą w tym samym miejscu, co osie obrotu ramion). Śruby umieszczone w tych osiach — **B.625.R.Screw** i **B.615.R.Screw** — przypisałem (relacją *Parent*) do okuć. Nadałem im ograniczenia *Stretch To*, by ich osie *Y* były zawsze skierowane na oś przegubu. Potem możesz z powrotem „odsłonić” (**Alt-H**) schowane przed chwilą ramiona. Przypisz każdą z nich relacją *Parent* do odpowiedniej śruby (**B.625.R.Screw** dla górnego ramienia, **B.615.R.Screw** dla dolnego). Powinieneś uzyskać rezultat jaki pokazuje Rysunek 2.4.14a).

- Szczegółowy opis budowy tego mechanizmu znajdziesz w sekcji dotyczącej ograniczenia *Limit Distance*, na str. 254. Wykorzystałem tam go jako przykład.

Aby wymusić ruch dolnej części amortyzatora tylko w założonych granicach (max. skok tłoka —  $\pm 9$  cm), przypiszemy mu ograniczenie *Limit Location* (Rysunek 2.4.14):

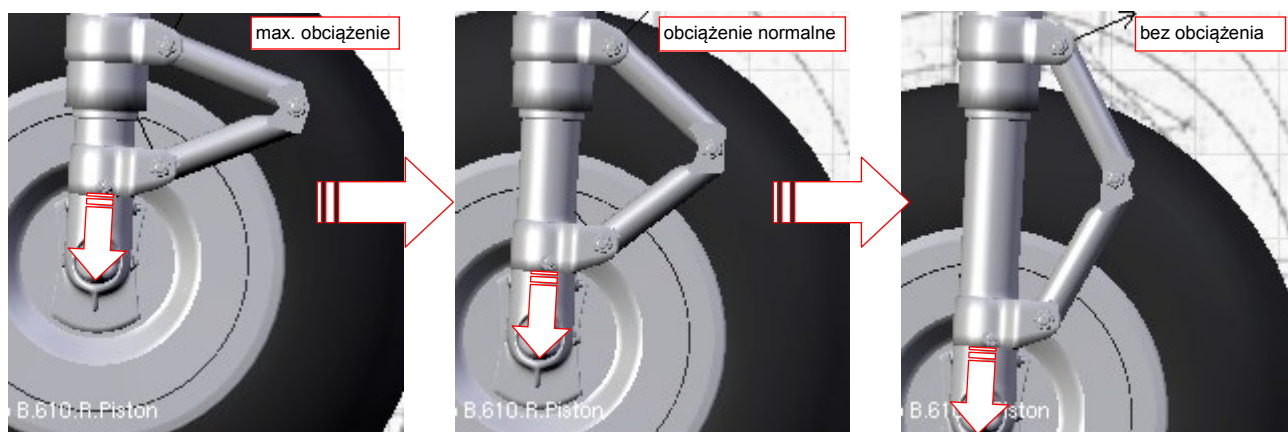


Rysunek 2.4.14 Ograniczenie ruchu amortyzatora.

Uważaj: przed przypisaniem tłokowi ograniczenia *Limit Location*, ustaw na chwilę całą goleń zupełnie pionowo (Rysunek 2.4.14a). (To dlatego poprzednio podkreślałem, że powinna być pochylona o jakąś „okrągłą” wartość kąta, by łatwo było ją na chwilę przywrócić do pionu — por. str. 64). Tylko w takim położeniu Blender poprawnie „zrozumie”, co masz na myśli pozostawiając obiektowi swobodę ruchu tylko wzdłuż osi *Z* (Rysunek 2.4.14b). Potem możesz z powrotem pochylić całą goleń — dzięki przełączeniu *Convert* na *Local Space*, kierunki działania ograniczeń obrócą się wraz z nią.



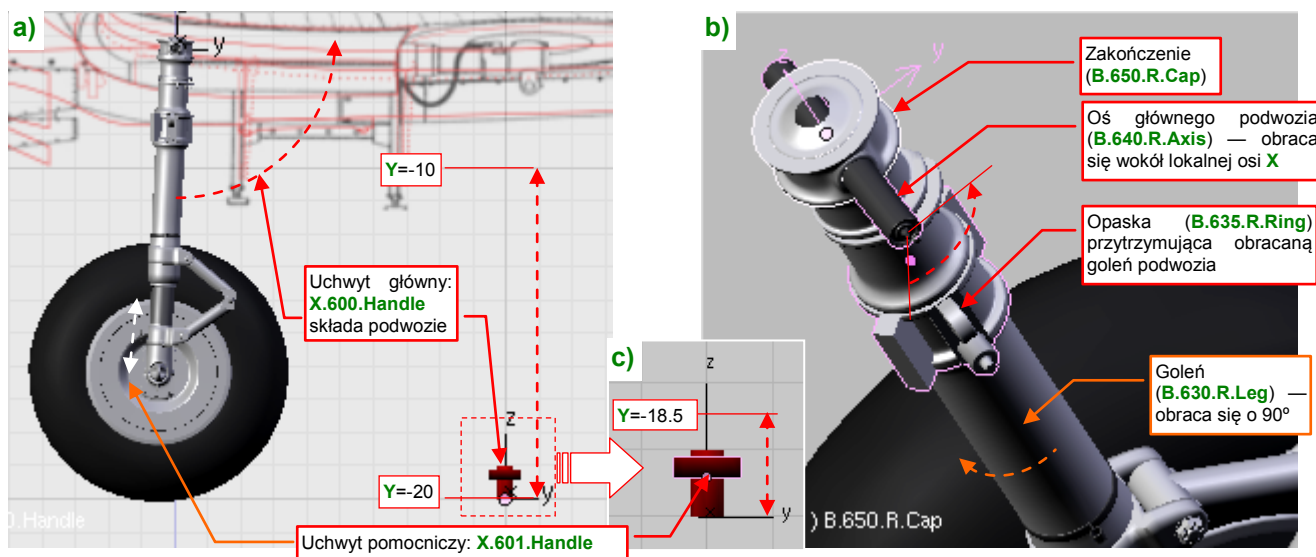
Rysunek 2.4.15 przedstawia działanie mechanizmu, który zbudowaliśmy. Wraz z ruchem tłoka ramiona łączników poprawnie się składają i rozkładają:



Rysunek 2.4.15 Działanie mechanizmu przegubowego połączenia goleni.

Zapewne, ten sam efekt można było uzyskać za pomocą armatury, złożonej z dwóch kości (por. str. 229). Jednak w poprzedniej sekcji zdecydowałem się wyrugować armatury z mechanizmu kółka ogonowego. Aby zachować jednorodność rozwiązań w modelu, muszę teraz konsekwentnie wykonać każdy mechanizm wyłącznie za pomocą ograniczeń. Zresztą — to nie jest aż takie trudne, jak mogłoby się wydawać.

Czas pomyśleć o dalszej „mechanizacji” naszego podwozia. Zaczniemy od przygotowania dwóch uchwytów, działających w sposób podobny do uchwytów kółka ogonowego (Rysunek 2.4.16a):

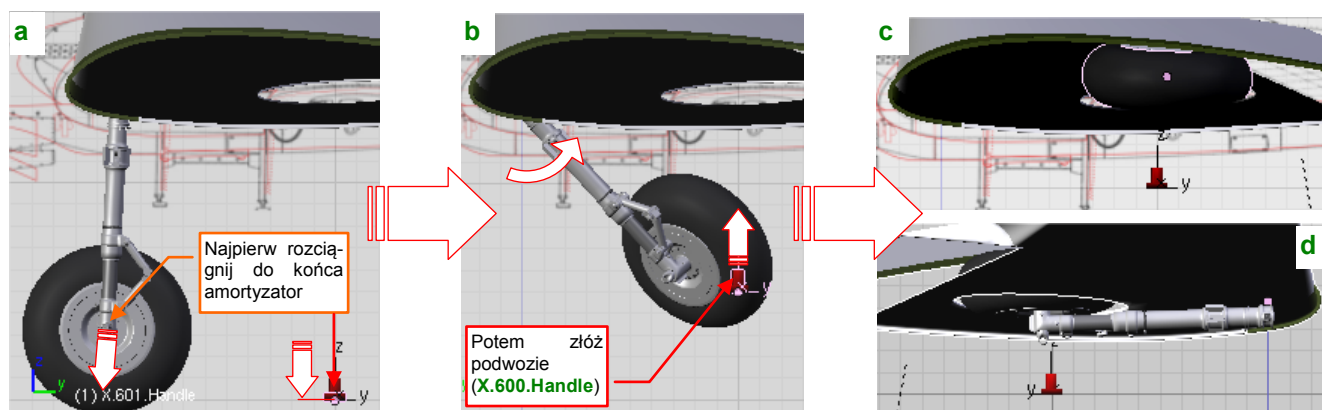


Rysunek 2.4.16 Podstawowe elementy animacji goleni

Podstawowym uchwytem, odpowiedzialnym za złożenie podwozia, jest pomocniczy obiekt **X.600.Handle**. Jego ruch ograniczyłem do osi **Y**, od -10 (podwozie schowane) do -20 (zupełnie wysunięte). **X.600.Handle** jest „rodzicem” pomocniczego uchwytu **X.601.Handle** (Rysunek 2.4.16c). To uchwyt sterujący ugięciem amortyzatorów. Ograniczyłem jego ruch do 1.5 jednostki Blendera (15 cm), by dokładnie odpowiadał zakresowi ugięcia podwozia głównego. (Z rysunków fabrycznych wynika, że dla normalnej masy startowej ugięcie wynosiło 6cm. Gdy zakresy ruchu amortyzatora i sterującego uchwytu są takie same, ustawienie go w odpowiedniej pozycji nie nastręcza problemów).

Goleń podwozia P-40 nie tylko obracała się do tyłu, ale także wokół swojej osi. Aby dobrze odwzorować ten ruch, odtworzyłem elementy oryginalnego podwozia (Rysunek 2.4.16b). Dodałem oś — **B.640.R.Axis** — obracaną za pomocą uchwytu **X.600.Handle** (poprzez ograniczenie **Transform**). Bezpośrednio do osi przypisane są trzy elementy: opaska (**B.635.R.Ring**) i „czepiec” (**B.650.R.Cap**), oraz sama goleń (**B.630.R.Leg**). Podczas składania podwozia goleń dodatkowo obraca się wokół swojej osi (w wyniku ograniczenia **Transform**) o 90°.

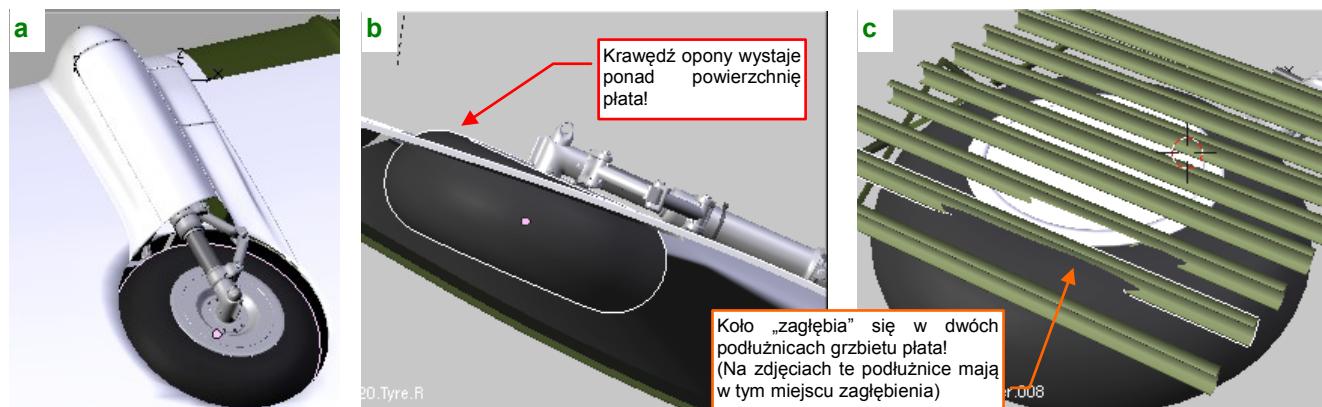
A jak to wygląda w działaniu? Składanie podwozia zacznij od rozciągnięcia amortyzatora (za pomocą uchwytu **X.610.Handle**) do pozycji swobodnej (Rysunek 2.4.17a):



Rysunek 2.4.17 Składanie podwozia głównego

(Inaczej koło nie trafi w swój otwór!). Następnie złap za główny uchwyt (**X.600.Handle**) i zacznij go przesuwając do góry. Spowoduje to złożenie goleni, połączone z jednoczesnym obrotem (Rysunek 2.4.17b) — tak jak w prawdziwym P-40. Zakres obrotu **B.640.R.Axis** wokół lokalnej osi **X** — od  $-3^\circ$  do  $+88^\circ$ . Zakres obrotu goleni **B.630.R.Leg** wokół lokalnej osi **Z** — od  $0^\circ$  do  $+96^\circ$  (trochę więcej niż  $90^\circ$ , by płaszczyzny kół leżały na powierzchni płata o wzniosie  $+6^\circ$ ). Rysunek 2.4.17c, d) pokazuje podwozie główne w pozycji złożonej.

Dopasowanie odpowiednich kątów złożenia wymaga paru iteracji, podczas których należy uważnie sprawdzać położenie koła względem poszycia płata i ścian komory. Te, które podałem, dobrałem na podstawie rysunków fabrycznych. Pozwalają złożyć podwozie tak, jak to pokazuje Rysunek 2.4.18a):



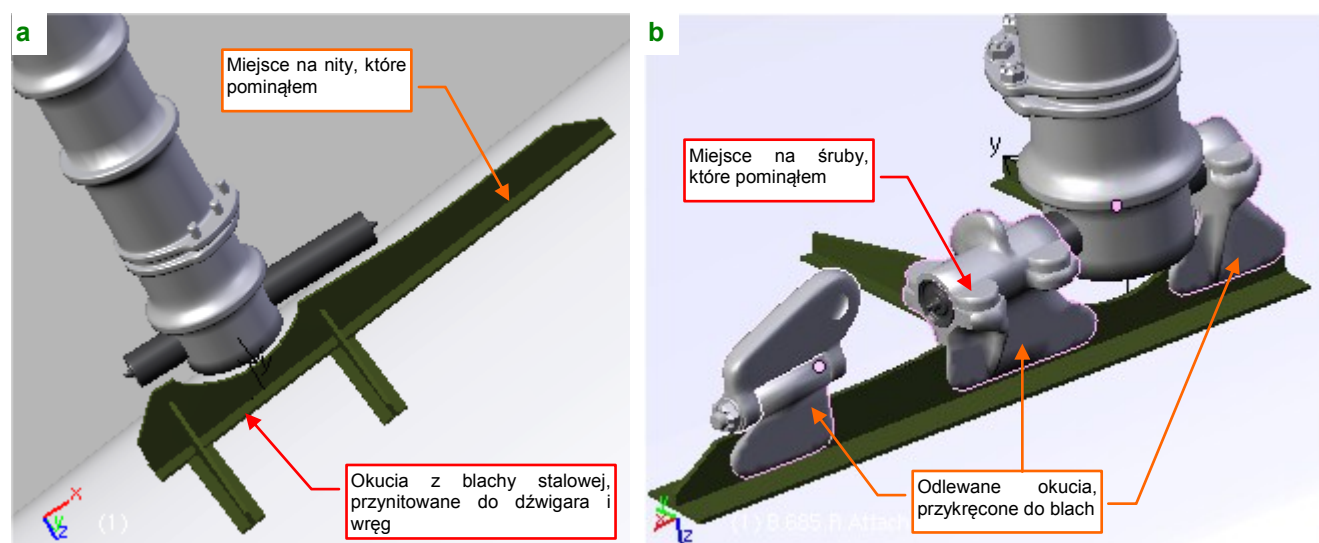
Rysunek 2.4.18 Dopasowanie kół do komory podwozia

Wydaje się być całkiem poprawne — tak samo złożone podwozie wygląda na dostępnych zdjęciach P-40, wykonanych podczas lotu. Przy okazji dopasowywania wyszły na jaw ciekawe szczegóły:

- Tylna krawędź opony wystaje z dolnej powierzchni skrzydła (Rysunek 2.4.18b)! Sprawdzając trzy razy rozmiar opony, kąt pochylenia złożonej goleni, i doszedłem do wniosku, że tak było naprawdę<sup>1</sup>. Jak pokazuje Rysunek 2.4.18a), w większości ujęć zupełnie tego nie widać, gdyż otwór komory jest nieco większy od koła. Dzięki temu czarna, matowa, wystająca opona nie rzuca się w oczy;
- Wewnętrzna powierzchnia koła zagłębia się w dwóch tylnych podłużnicach górnego pokrycia płata (Rysunek 2.4.18c)! Dzięki temu zrozumiałem, dlaczego te dwie podłużnice na zdjęciach mają wyraźne wgłębienia wzdłuż komory podwozia. Nie zamodelowałem ich, bo mało kto w to miejsce modelu będzie zaglądał. Oczywiście, jeżeli chcesz wykonać dokładniejszy model, nie zapomnij o tym szczególe;

<sup>1</sup> P-36 Hawk i nawet prototyp XP-40 miały mniejsze koła, które dokładnie mieściły się w płacie. To większe podwozie wprowadzono podczas przygotowywania produkcji seryjnej. Z tego powodu został wygięty tylny dźwigar. Przy tak poważnej zmianie parę centymetrów gumy pod skrzydłem nic nie znaczyło! Prawdopodobnie to z tego powodu zrezygnowano z pokryw kół, które miał P-36.

Uruchomiliśmy już podwozie, ale czas je wzbogacić o resztę „żelastwa”, które widać na zdjęciach. Zaczniemy od przygotowania okuć, które mocowały oś do skrzydła (Rysunek 2.4.19a):

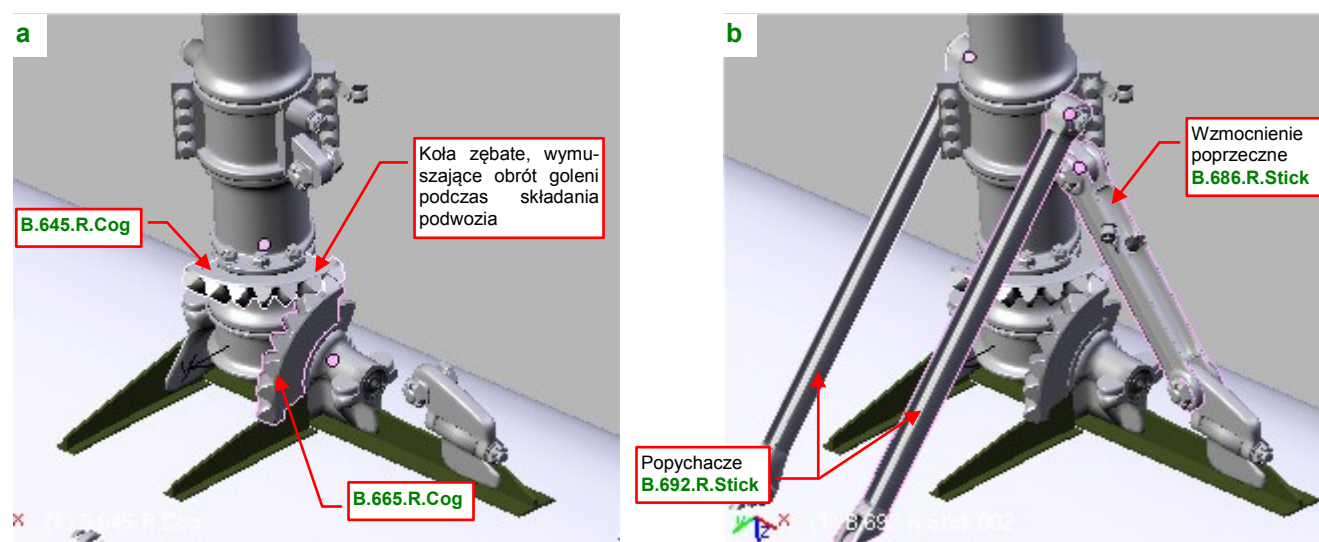


Rysunek 2.4.19 Okucia osi podwozia

Na zdjęciach i rysunkach konstrukcyjnych P-40 widać blachy (lub odkuwki) o profilu w kształcie „T”, przynitowane do dźwigara i wręg. Zamodelowałem je jako jeden obiekt (**B.660.R.Frame**), starannie dopasowując do powierzchni skrzydła. Gdybyś chciał robić jakiś bardzo dokładny model, należałoby odwzorować — np. za pomocą mapy nierówności — wypukłe nit, którymi przymocowano półki tych elementów do płata. Ja jednak przypisałem tym wzmocnieniom standardowy materiał **B.Skin.Inner**, bez nitów. I tak nie będzie ich prawie widać.

Do blach okuć były przykręcone (za pomocą wielu śrub) kolejne elementy, mocujące oś podwozia (Rysunek 2.4.19b). Mówiąc szczerze, mimo rysunków i paru zdjęć, jakie miałem do dyspozycji, trudno było zdeterminować ostateczny kształt tych części. Kilka razy musiałem je szkicować na kartce papieru, by uświadomić sobie ich wszystkie zakamarki. I w tym przypadku pominąłem kilkanaście śrub, którymi były przykręcone do blach okuć — ich także prawie nie widać.

Następnie dodałem dwie zębátky, które wymuszały obrót goleni podczas składania podwozia (Rysunek 2.4.20a):

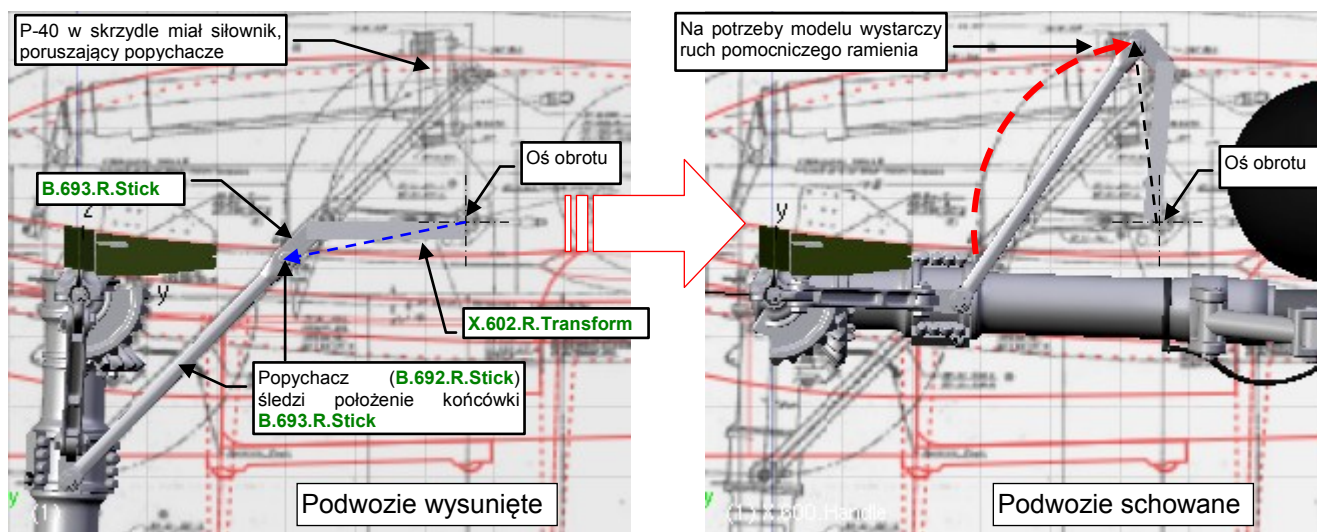


Rysunek 2.4.20 Dalsze szczegóły podwozia

Zębátka **B.645.R.Cog** jest przypisana do obracającej się goleni **B.630.R.Leg**. (W tym momencie przydało się zamodelowanie goleni jako pełnej siatki: można było z niej łatwo wytłoczyć niesymetryczną wypustkę do umocowania zębátky). Zębátkę **B.665.R.Cog** przypisałem do nieruchomych okuć osi. Potem dodałem pozostałe elementy: popychacze (**B.692.R.Stick**) i wzmocnienie poprzeczne (**B.686.R.Stick**) (Rysunek 2.4.20b). Każde z nich jest przypisane (relacją **Parent**) do opaski **B.635.R.Ring**, by składały się wraz z resztą podwozia.



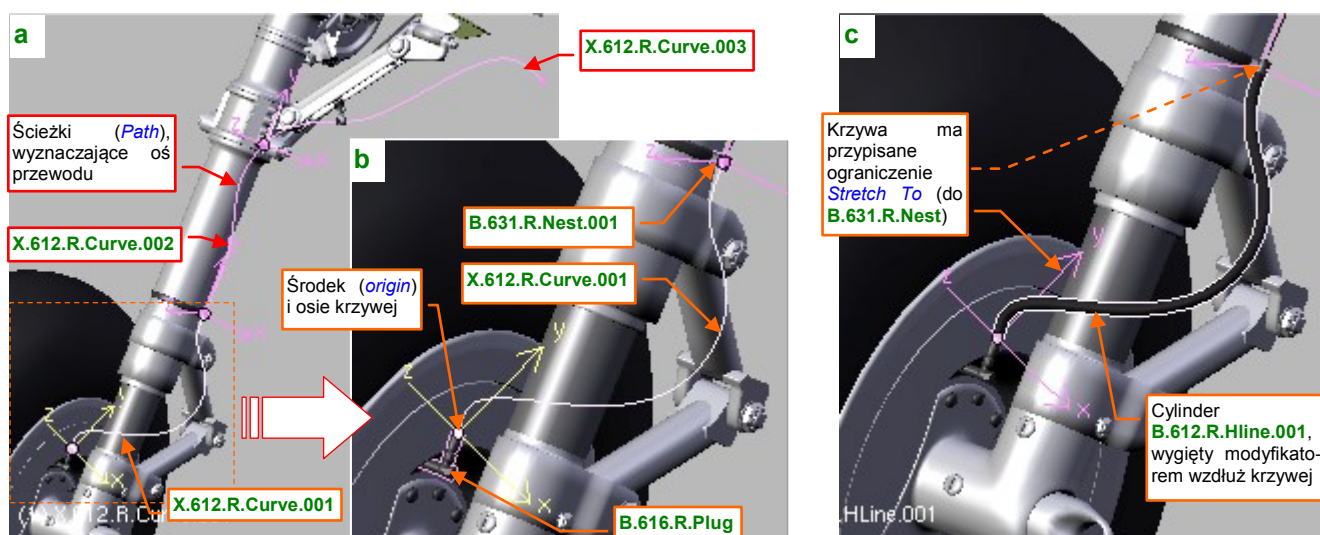
Aby popychacze wykonywały ruch przypominający rzeczywisty, dodałem na ich krańcach końcówki (**R.693.R.Stick**). Są one przypisane do zupełnie niewidocznej ramki, która je obraca (**X.602.R.Transform**) (Rysunek 2.4.21):



Rysunek 2.4.21 Mechanizm popychaczy

Ramka **X.602.R.Transform** ma przypisane ograniczenie **Transformation**. Dzięki temu podczas składania podwozia wykonuje obrót o taki sam kąt, jak ramka popychaczy. Do ramki są przypisane (relacją **Parent**) końcówki **R.693.R.Stick**, więc obracają się wraz z nią. Popychacze **B.692.R.Stick** mają ustawione ograniczenia **Locked Track** na swoje końcówki, więc podczas chowania podwozia poruszają się po trajektorii bardzo zbliżonej do rzeczywistej. Mechanizmu, ukrytego wewnątrz skrzydła, nie modelowałem. I tak jest zupełnie niewidoczny.

Wzdłuż goleni P-40 (i większości samolotów) biegnie elastyczny przewód z płynem hamulcowym. Ten detal zamodelowałem za pomocą krzywej (**Curve**: szczegóły — str. 260), wokół której „owinałem” rurkę (cylinder) przewodu (za pomocą polecenia **Parent→Curve Deform**, opisanego na str. 288). Gdyby chodziło o statyczny model, to by wystarczyło. Nasza goleń potrafi się jednak obrócić, i może się w niej ugiąć amortyzator. To zmusiło mnie do zamodelowania przewodu za pomocą trzech oddzielnych krzywych (Rysunek 2.4.22a):

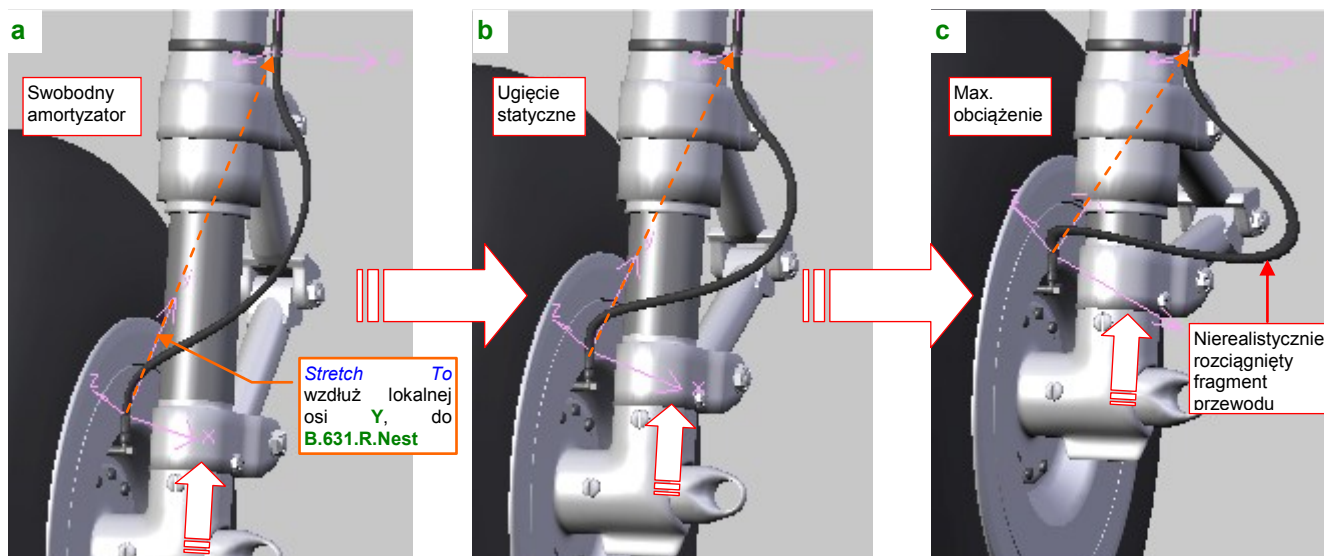


Rysunek 2.4.22 Odzworowanie przewodu hamulcowego

Krzywe — **X.612.R.Curve.001** .. **003** — są łączone wewnątrz dwóch opasek, mocujących przewód do goleni. Na przykład pierwszy segment — **X.612.R.Curve.001** (Rysunek 2.4.22b) — ma swój początek (i punkt odniesienia) we wtyczce do tarczy hamulca. Drugi koniec dotyka opaski **B.631.R.Nest.001**. Aby przewód ugiął się elastycznie przy ruchu amortyzatora, krzywa jest powiązana z opaską ograniczeniem **Stretch To**. Wzdłuż krzywej wygięłem długi i cienki cylinder przewodu **B.612.R.HLine.001** (Rysunek 2.4.22c)



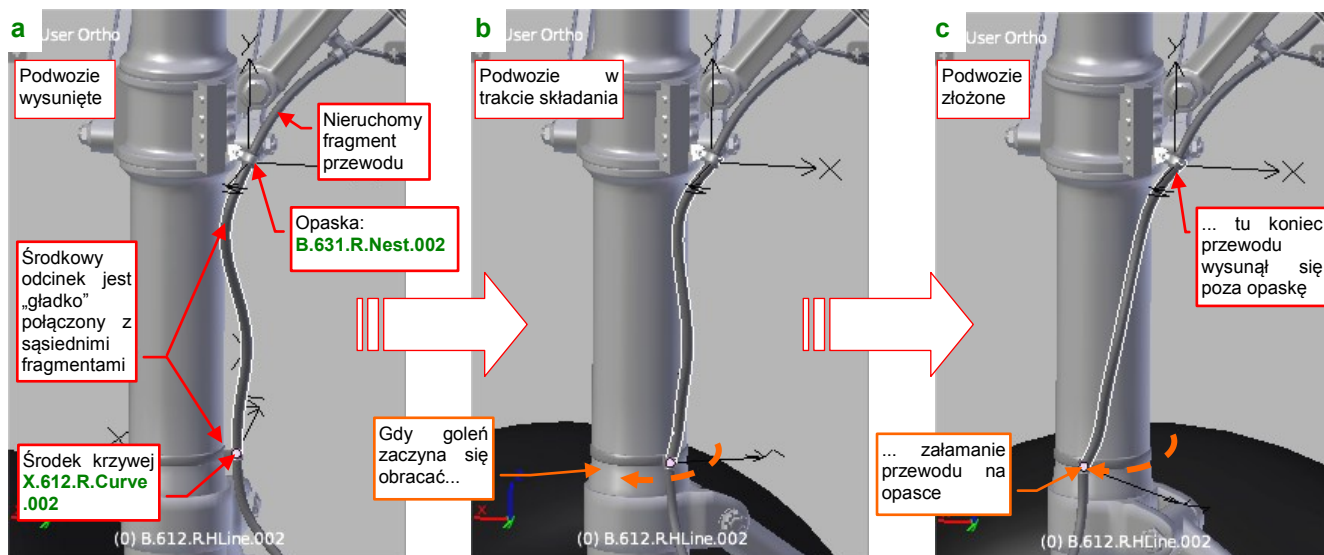
Sprawdźmy, jak „pracuje” przygotowany w ten sposób zespół podczas ruchu amortyzatora (Rysunek 2.4.23):



Rysunek 2.4.23 Ugięcie dolnej części przewodu hamulcowego

Gdy amortyzator jest zupełnie rozciągnięty (bez obciążenia — Rysunek 2.4.23a), ograniczenie **Stretch To** wydłuża krzywą **X.612.R.Curve.001**, wraz z powiązany z nią przewodem. Przekrój poprzeczny przewodu staje się wtedy w niektórych miejscach eliptyczny. Przy ugięciu statycznym amortyzatora (Rysunek 2.4.23b) wszystko wygląda normalnie (bo jest to pozycja „neutralna” modyfikatora **Stretch To**). Przy maksymalnym ugięciu (Rysunek 2.4.23c) na przewodzie pojawiają się nierealistyczne zgrubienie. Zdecydowałem się jednak tak to pozostawić, gdyż w tej pozycji amortyzator będzie się znajdował bardzo rzadko, i tylko przez chwilę. Zresztą w ugięciach z większych odległości ten defekt w ogóle nie będzie widoczny.

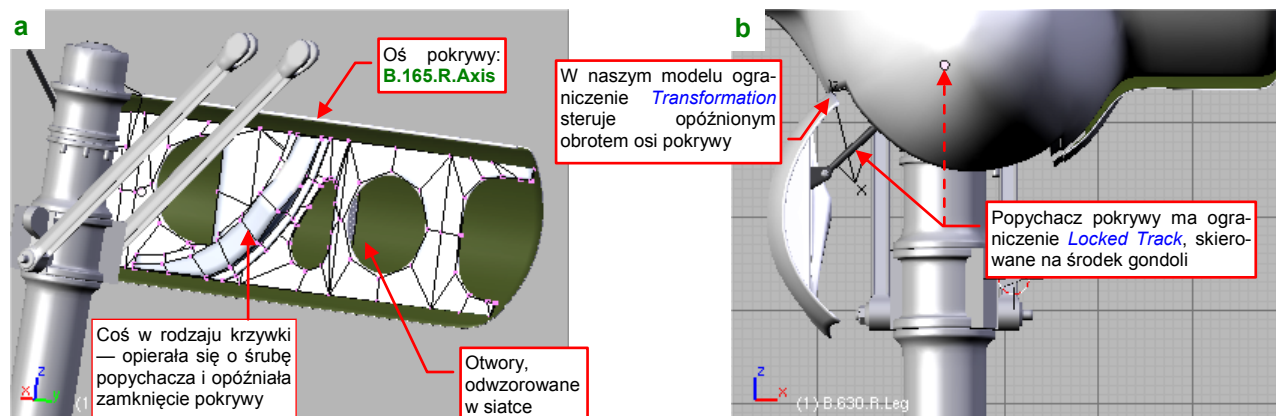
Podobnie zamodelowałem środkową część przewodu (**X.612.R.Curve.002**). Jej górny kraniec był przymocowany do nieruchomej „opaski”, a dolny — obracał się razem z gołenią (Rysunek 2.4.24):



Rysunek 2.4.24 Skręcenie środkowej części przewodu

Końcówki przewodu powinny zawsze przechodzić „gładko” w odcinki sąsiednie. Punkt odniesienia krzywej **X.612.R.Curve.002** leży na jej dolnym krańcu. Krzywa jest „przymocowana” (relacją **Parent**) do obracanej części gołeni. Podobnie jak poprzedni fragment, jest „rozciągana” ograniczeniem **Stretch To**, wzdłuż osi **Y**, do nieruchomej opaski **B.631.R.Nest.002** (Rysunek 2.4.24a). Takie rozwiązanie powoduje załamanie przewodu na opaskach w końcowych fazach składania podwozia (porównaj Rysunek 2.4.24b i c). Jednak gołeń jest wówczas już zasłonięta pokrywą i ten defekt nie jest widoczny.

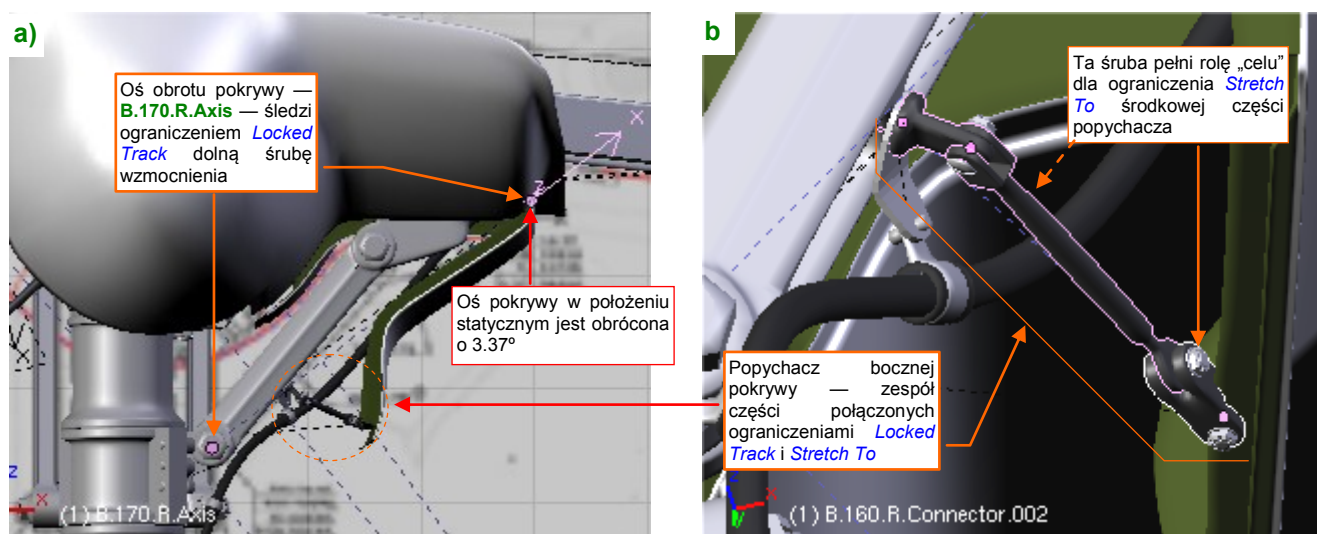
Kolejnym elementem zespołu podwozia jest pokrywa goleni (Rysunek 2.4.25):



Rysunek 2.4.25 Pokrywa goleni wraz z popychaczem

Przy otwartym podwoziu jej otwory są dość widoczne, więc zdecydowałem się je wykonać nie teksturą przejrzystości, a „w siatce” (Rysunek 2.4.25a). Najważniejszą częścią pokrywy jest oś (**B.165.R.Axis**). Przypisałem jej ograniczenie **Transformation**, by zamykała się w odpowiedni sposób (szczegóły ustawień tego ograniczenia znajdziesz na str. 79). Popychacz i jego okucie (Rysunek 2.4.25b) to kopie elementów z pokryw kółka ogonowego. Popychacz ma przypisane ograniczenie **Locked Track**.

Oś pokrywy bocznej obraca się w trakcie wysuwania goleni dzięki ograniczeniu **Locked Track**, kierującemu oś **X** na jedną ze śrub<sup>1</sup> (Rysunek 2.4.26a):

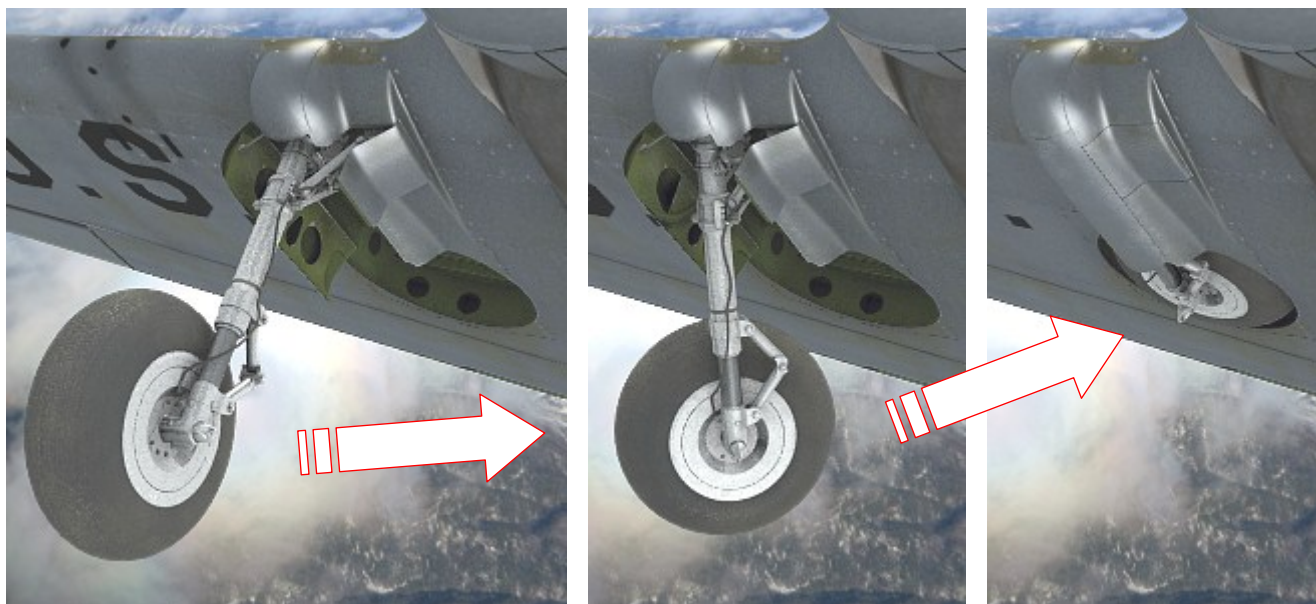


Rysunek 2.4.26 Pokrywa boczna i jej popychacz

Wraz z osią obraca się połączona z nią „na sztywno” (relacją **Parent**) pokrywa. Pomiędzy pokrywą a golenią podwozia znajdował się mały popychacz (Rysunek 2.4.26b). Powstał ze złożenia trzech śrub i tyluż elementów „podłużnych”. (To znacznie przerobiony fragment popychacza pokryw kółka ogonowego — por. str. 59, Rysunek 2.3.21). Śruby w tym elemencie nie są tylko „upiększeniem”. Pełnią także rolę „celów” dla poszczególnych ograniczeń **Locked Track** i **Stretch To**. W ten sposób unikamy komunikatów w konsoli Blendera o wykryciu zapętlenia w zależnościach pomiędzy obiektami. (Z tego samego powodu dodałem podobne śruby do mechanizmu kółka ogonowego).

<sup>1</sup> Gdyby na przedłużeniu osi **X** nie leżał środek żadnej ze śrub, odłączyłbym ją na chwilę od pokrywy, i odpowiednio obróciłem, kierując oś **X** na właściwy cel. Należy to robić, gdy podwozie jest złożone, a pokrywa — domknięta. Cel musi być tak dobrany, by przy wysuniętym podwoziu oś **X** odchylała się o taki kąt, o jaki obracała się pokrywa. Gdyby w modelu nie byłoby na co wycelować osi — dodałbym odpowiedni obiekt **Empty**. Potem ponownie przypisałbym pokrywę do osi relacją **Parent**.

Rysunek 2.4.27 prezentuje ostateczny efekt — podwozie główne w kolejnych fazach składania:



Rysunek 2.4.27 Podwozie główne podczas wciągania

### Podsumownie

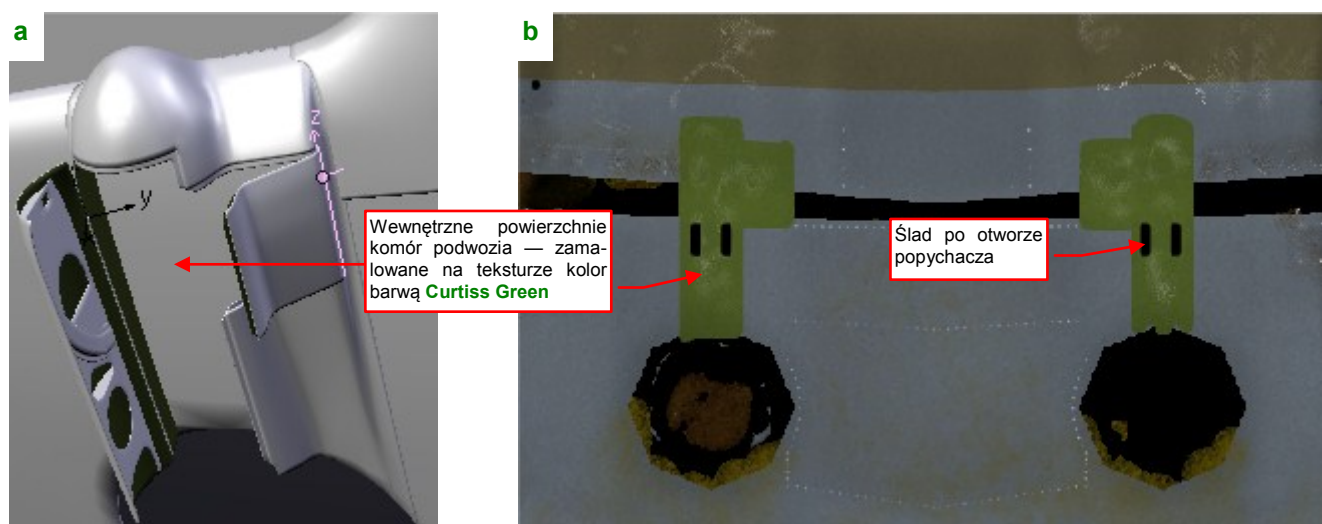
- Dokładne rysunki podwozia samolotu często nie są dostępne. Należy wówczas posłużyć się zdjęciami. Zazwyczaj najlepszej jakości, kolorowe ujęcia pochodzą ze współcześnie odrestaurowanych egzemplarzy. Takie „maszyn z odzysku” powinienś traktować z pełną rezerwą: w wielu szczegółach mogą się różnić od oryginału! (Zazwyczaj dlatego, że nie zachowały się odpowiednie detale, a ich wierne odtworzenie było zbyt drogie);
- Opony koła głównego i kółka ogonowego można pokryć tym samym materiałem (w tym modelu jest to **B.Rubber.Tyre** str. 62). Gdyby koło główne miało bieżnik (a tak jest w późniejszych wersjach P-40) — trzeba byłoby stworzyć nowy materiał, z odpowiednią teksturą nierówności. Można byłoby jednak w nim powtórnie wykorzystać przynajmniej większość tekstur proceduralnych, które opracowaliśmy w poprzedniej sekcji;
- Podwozie główne jest modelowane w ten sam sposób, co kółko ogonowe (str. 63 - 64). Czasami, by uniknąć kłopotliwego łączenia siatek, „łączyłem” dwie powłoki poprzez zwykłe złożenie (str. 66);
- Do uzyskania realistycznego ruchu ramion łącznika amortyzatora można użyć ograniczenia **Limit Distance** (str. 67);
- Do „uruchomienia” podwozia używamy zespołu dwóch uchwytów — pomocniczego do ugięcia amortyzatora i głównego do chowania i otwierania (str. 68). Takie samo rozwiązanie zastosowaliśmy w poprzedniej sekcji, do wymuszenia ruchu kółka ogonowego;
- Koło podwozia musi być prawidłowo dopasowane do swojej komory (str. 69). Ten proces wymaga zazwyczaj kilku przybliżeń. Czasami można odkryć, że opona jest zbyt szeroka, lub amortyzator miał w pozycji „swobodnej” trochę inną długość;
- Okucia, mocujące podwozie, i inne detale mogą przysporzyć dużo pracy, jeżeli chcesz je wykonać dokładnie. Lepiej uważaj, i naucz się w odpowiednim momencie rezygnować z części drobnych, mało widocznych szczegółów!
- Do odtworzenia przewodu, doprowadzającego płynem hamulcowym do tarczy koła, najlepiej jest skorzystać z krzywej (**Path**) i modyfikatora **Curve** (str. 71);



## 2.5 Podwozie — wykończenie

W poprzednich sekcjach uformowaliśmy i uruchomiliśmy kółko ogonowe i jedno z kół głównych. W tej sekcji uzupełniamy wszystko, czego jeszcze w podwoziu brakuje, by definitywnie zakończyć ten zespół.

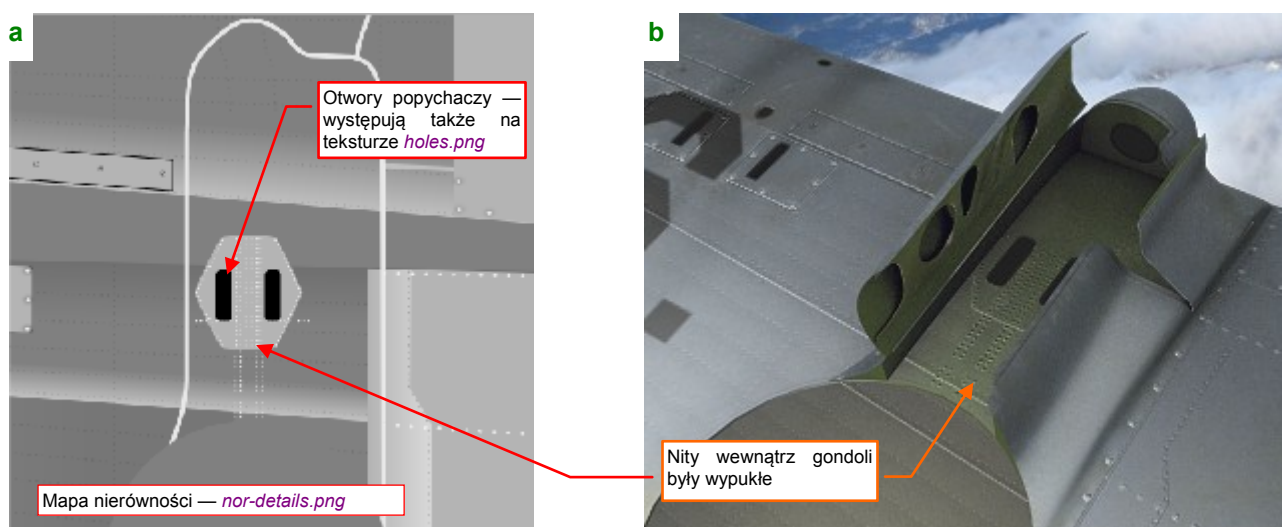
Zacznijmy od wnętrza komory podwozia. W przypadku kółka ogonowego wypełniliśmy ją wręgami już na etapie modelowania (por. Wyciąg II). Uważam, że ten fragment jest niemal na każdym ujęciu niewidoczny, więc nie trzeba go o nic dodatkowego wzbogacać. Co innego z wnętrzem gondoli koła głównego (Rysunek 2.5.1a):



Rysunek 2.5.1 Zamalowane fragmenty płata wewnątrz owiewek gondoli podwozia

W P-40 była to wzmocniona powierzchnia płata, pokryta taką samą farbą jak inne powierzchnie wewnętrzne: **Curtiss Green**. Uznałem, że nie trzeba w tym miejscu zagęszczać siatki skrzydła tylko po to, by przypisać odpowiednim ścianom materiał **B.Skin.Inner**. Tak więc ten fragment jest pokryty materiałem **B.Skin.Camouflage**, tak jak cała reszta płata. Po prostu zamalowałem odpowiednie miejsca tekstury barwy (*color.png*) odpowiednim kolorem (Rysunek 2.5.1b). Zrobiłem to na nowej warstwie — **Color-Inner** — w źródłowym pliku *skin.xcf*.

Poza barwą, warto wzbogacić ten fragment poszycia o zarys blach, naniesionych w charakterze wzmocnienia, nity, oraz małe otwory (Rysunek 2.5.2):



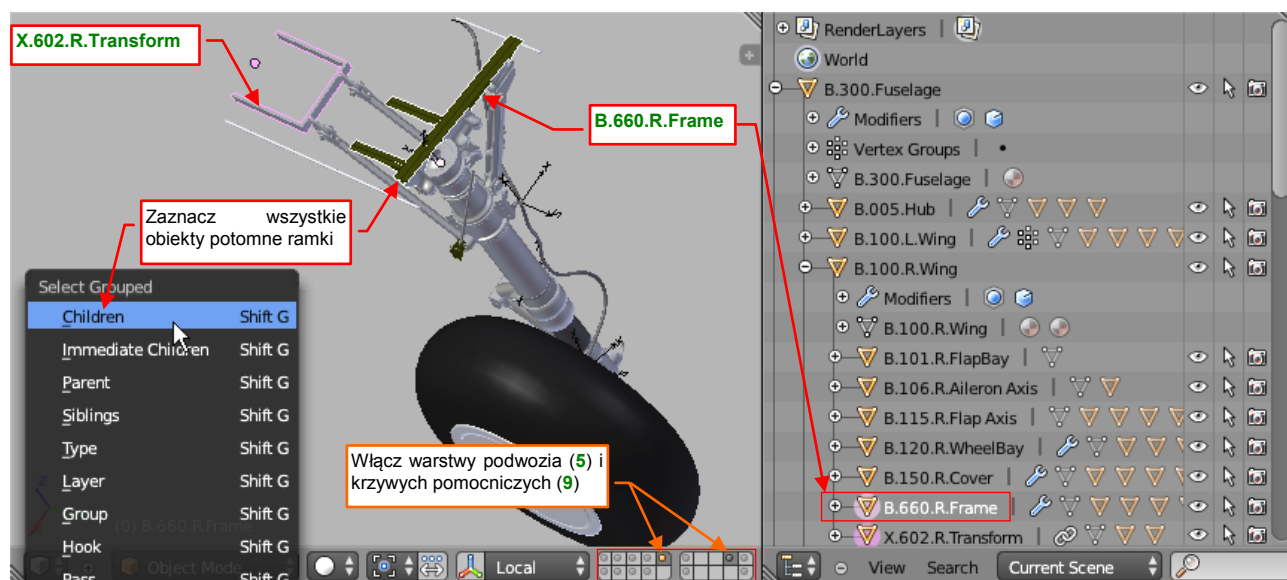
Rysunek 2.5.2 Odzworowanie szczegółów powierzchni płata wewnątrz owiewki podwozia

Nity na wszystkich powierzchniach wewnętrznych były wypukłe (bo naniesienie takiego szwu było mniej pracochłonne i tańsze, a sam szew — bardziej wytrzymały). Naniosłem więc zespół białych „kółek” na odpowiednią warstwę rysunku *skin.svg*. Na tym rysunku wektorowym znalazło się także wzmocnienie wokół otworów w



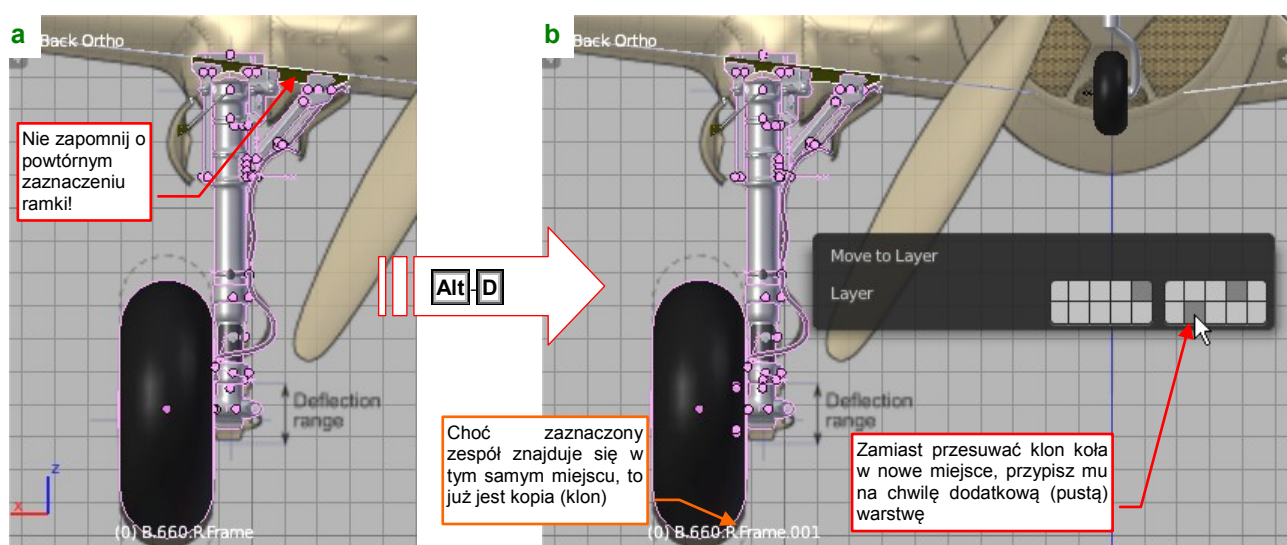
skrzydle (Rysunek 2.5.2a), oraz same otwory. Z pliku *skin.svg* powtórnie wygenerowałem ponownie wszystkie pliki rastrowe, by uzyskać nowe obrazy tekstur (zgodnie ze schematem z Wyciągu III). Rezultat zastosowania tych poprawionych tekstur pokazuje Rysunek 2.5.2b).

Teraz można skopiować zespół prawego koła w koło lewe. Takich całości lepiej nie powielać poleceniem *Mirror*, bo potem poprawianie rezultatu może zająć dużo czasu. Lewe koło stworzymy jako „klon” prawego (w ten sposób obiekty wchodzące w skład lewego koła będą wykorzystywać te same siatki, co koło prawe). Przed zaznaczeniem elementów do skopiowania pamiętaj, aby oprócz warstwy **5** (podwozie) włączyć także warstwę nr **9**, z pomocniczymi obiektami (zawiera pomocnicze krzywe, „wyginające” przewód hydrauliczny) (Rysunek 2.5.3):



Rysunek 2.5.3 Przygotowanie do skopiowania podwozia

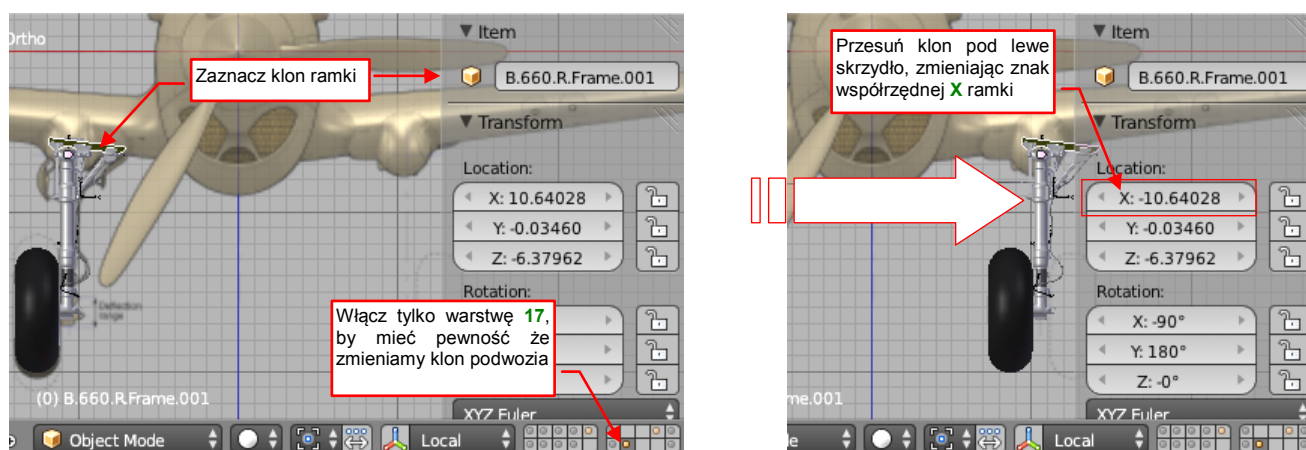
Obiektem nadrzędnym prawie całego zespołu podwozia jest ramka **B.660.R.Frame** (por. Rysunek 2.5.3). Zaznacz wszystkie obiekty potomne tej ramki (Najlepiej poleceniem *Select → Grouped → Children*). Tylko pamiętaj, że po wywołaniu tego polecenia należy włączyć do zbioru selekcji także samego „rodzica” (Rysunek 2.5.4a)! Potem sklonuj podwozie (**Alt-D**):



Rysunek 2.5.4 Skopiowanie (sklonowanie) podwozia

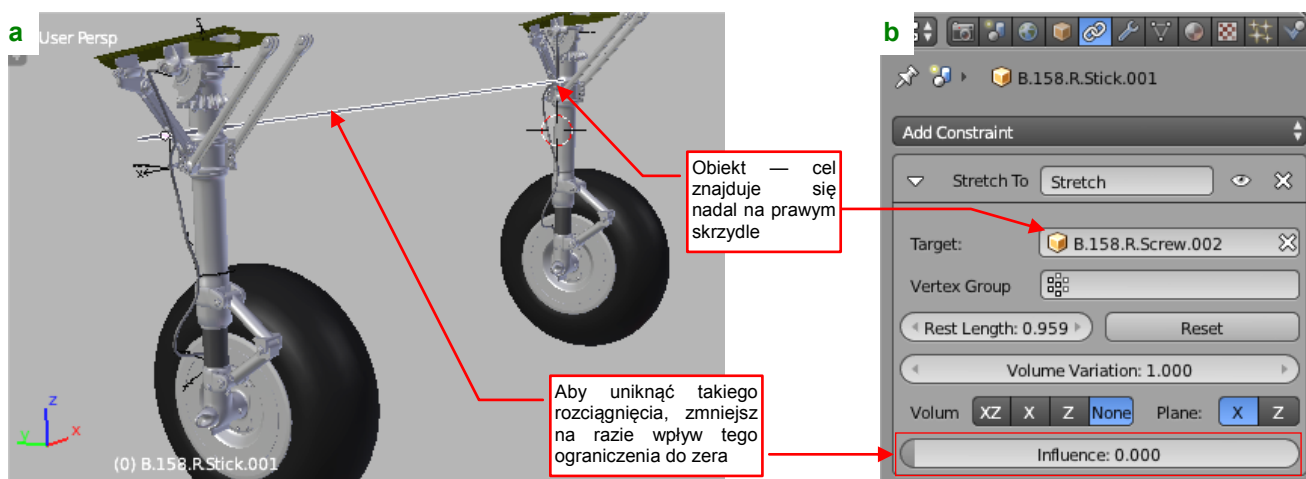
Po wywołaniu polecenia nie przesuwaj klona podwozia w nowe miejsce. Zamiast tego przypisz na czas pracy wszystkie zaznaczone obiekty do dodatkowej warstwy (Rysunek 2.5.4b). Powinna to być jakaś pusta warstwa, której w tej chwili nie używamy. Proponuję użyć do tego celu warstwę **17**.

Pozostaw teraz włączoną tylko warstwę **17** (Rysunek 2.5.5a). Zaznacz ponownie ramkę i przenieś ją na lewą stronę modelu, zmieniając znak jej współrzędnej **X** na ujemny (Rysunek 2.5.5b):



Rysunek 2.5.5 Zmiana położenia lewego podwozia

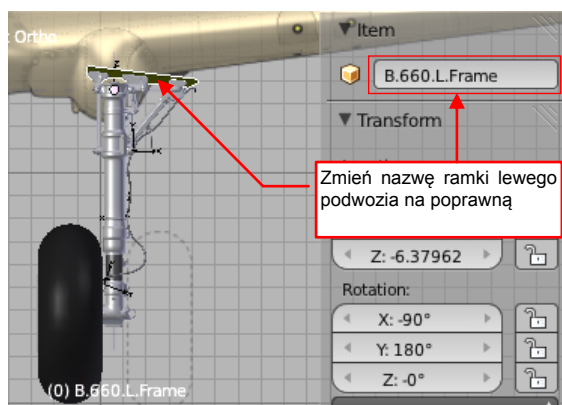
Teraz możesz już z powrotem włączyć widoczność warstwy **5** (podwozia). Gdy przyjrzyś się rezultatowi, zauważysz w lewym kole potwornie rozciągnięty popychacz (Rysunek 2.5.6a):



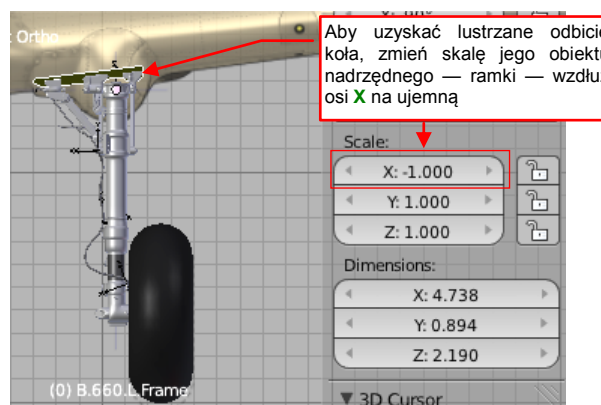
Rysunek 2.5.6 Efekty uboczne zmiany położenia

To rezultat przypisanego ograniczenia, które pozostało skierowane na obiekt przypisany do pokrywy prawego podwozia. Dopóki go nie skopiujemy, zmniejsz wpływ tego efektu do zera (**Influence** = 0 — Rysunek 2.5.6b).

Następnie zmień nazwę ramki lewego koła na **B.660.L.Frame** (Rysunek 2.5.7) i przerzuć cały zespół w poziomie, zmieniając skalę tego obiektu wzdłuż osi **X** na ujemną (Rysunek 2.5.8):



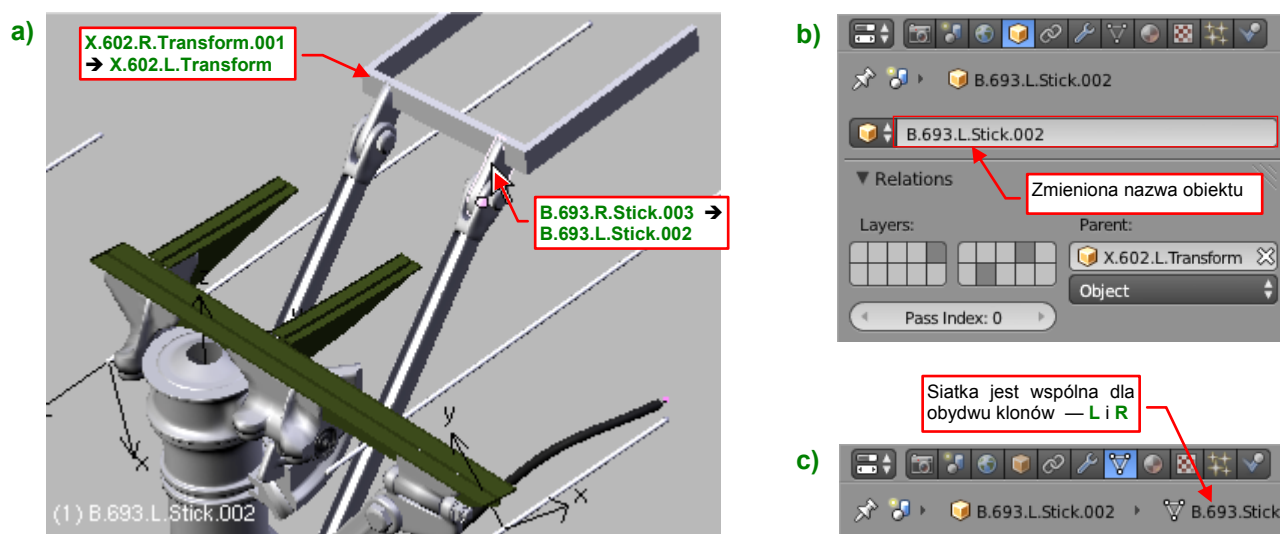
Rysunek 2.5.7 Zmiana nazwy ramki



Rysunek 2.5.8 Zmiana skali ramki wzdłuż osi **X** na ujemną

Na koniec przypisz ramkę lewego koła (obiekt **B.660.L.Frame**) do lewego skrzydła (poleceniem **Set Parent To→Object (Keep Transform)**). W podobny sposób sklonuj i przerzuć na lewą stronę popychacz **X.606.R.Transform** (por. Rysunek 2.5.3). To element jest symetryczny, więc nie trzeba mu zmieniać skali na ujemną. Pamiętaj tylko, że ma przypisane ograniczenie **Transformation** (powiązane z uchwytem sterującym podwoziem — por. str. 71, Rysunek 2.4.21). Przed przypisaniem popychacza do lewego skrzydła poleceniem **Set Parent** zmniejsz wpływ tego ograniczenia (**Influence**) do zera. Potem z powrotem możesz przestawić je na 1. Jeżeli o tym zapomnisz, Blender w wyniku jakiegoś błędu ustawi go w przestrzeni pod dziwnym kątem.

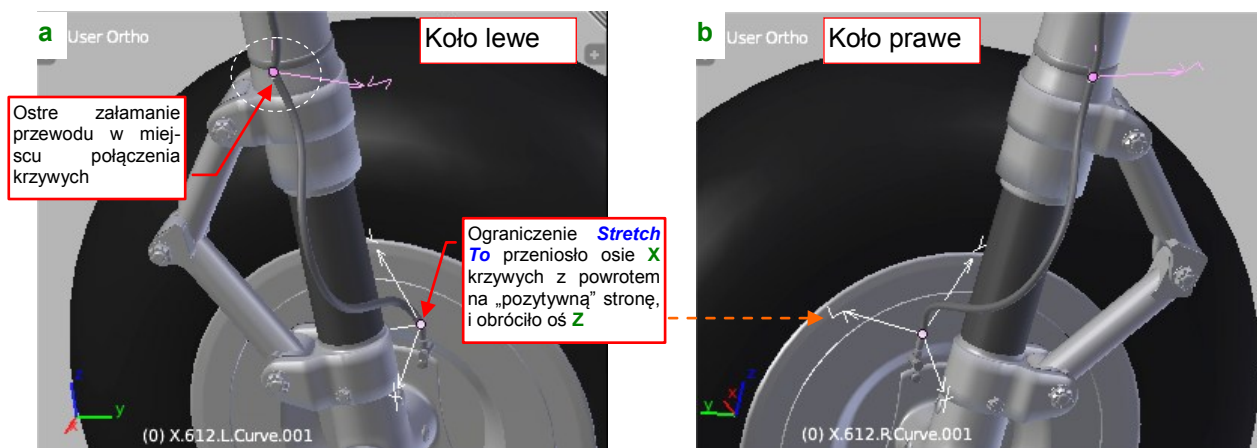
Teraz czeka nas trochę żmudnej pracy „porządkowej”: zmiana nazw wszystkich obiektów, wchodzących w skład skopiowanego zespołu. Blender domyślnie dodał do nich końcówki „.001”, „.002” (albo niektórym z nich takie końcówki usunął). By mieć porządek w modelu, musimy poprawić wszystkie nazwy, zmieniając w nich symbol strony z „R” na „L”. Aby niczego nie przeoczyć, rób to metodycznie, zaczynając np. od góry (Rysunek 2.5.9):



Rysunek 2.5.9 Porządkowanie nazw

Przesuwaj się sukcesywnie do dołu, aż nie poprawisz nazw całego zespołu. Na koniec sprawdź, czy o czymś nie zapomniałeś, przeglądając ten fragment w oknie **Outliner** (por. Wyciąg I).

Zmiana nazw wszystkich części lewego koła jest okazją do starannego przejrzania wszystkich jego zakamarków. Zwróć szczególną uwagę na niesymetryczne obiekty z przypisanymi ograniczeniami **Stretch To**. W naszym przypadku były to dwie krzywe, odpowiedzialne za wygięcie przewodów hydraulicznych. Gdy się im przyjrzyysz, zauważysz że na lewym kole mają inny, nieprawidłowy kształt (Rysunek 2.5.10):



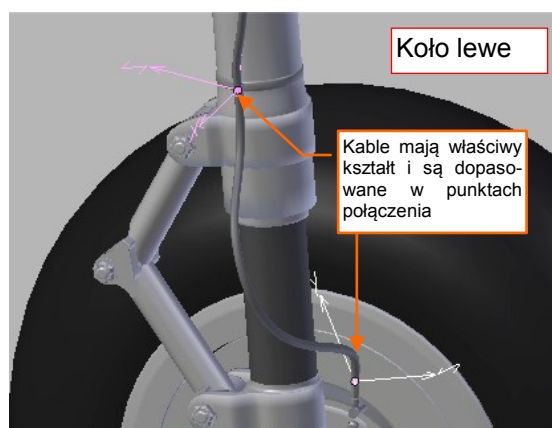
Rysunek 2.5.10 Wpływ ograniczenia **Stretch To** na kopie krzywych

Wynika to z faktu, że wszystkie obiekty lewego koła mają przestawiony zwrot osi **X** (to skutek ujemnej skali ramki — por. Rysunek 2.5.8). Ograniczenie **Stretch To** przestawia ten kierunek z powrotem na pozytywny...



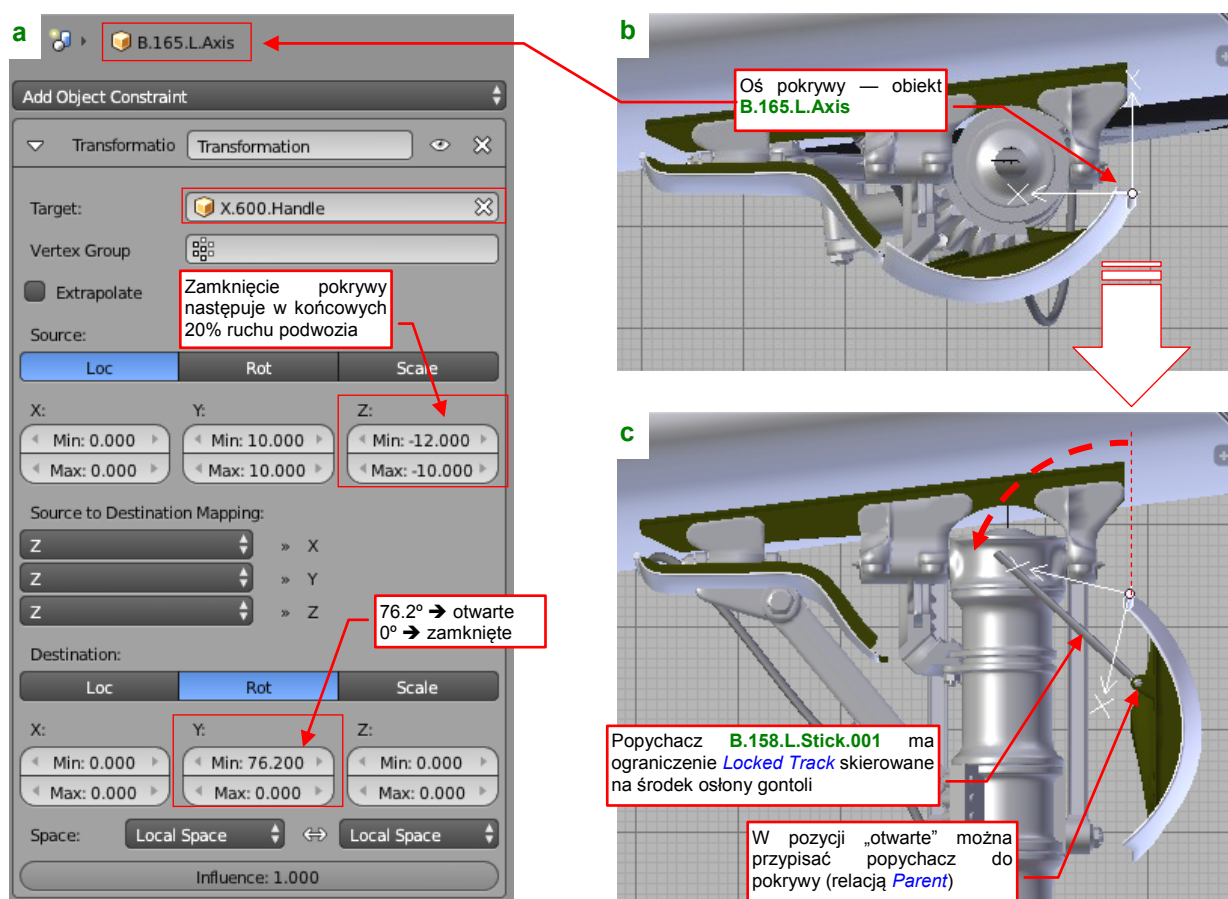
Takie problem może się pojawić dla obiektów, które używają ograniczenia *Stretch To* i są niesymetryczne względem lokalnej osi **Y**.

Aby sobie z nim poradzić, trzeba na chwilę wyłączyć taki obiekt z hierarchii zespołu, „utrwalić” (*Apply*→*Scale*) jego krzywej taką ujemną skalę, a potem skorygować orientację obiektu i krzywej (lub siatki). Szczegółowy opis jak to zrobić znajdziesz na str. 251. Podczas tej operacji konieczne jest m. in. utworzenie indywidualnych instancji krzywych dla lewego i prawego podwozia. Skorygowane krzywe przewodów z płynem hamulcowym przedstawia Rysunek 2.5.11:



Rysunek 2.5.11 Skorygowane krzywe na lewego koła

Gdy podstawowy zespół lewego koła jest gotowy, zajmiemy się jego pokrywami. W przypadku pokrywy głównej wystarczy przypisać jej osi (**B.165.L.Axis**) takie same ograniczenie *Transformation* jakie przypisaliśmy osi z prawej gondoli (Rysunek 2.5.12a):

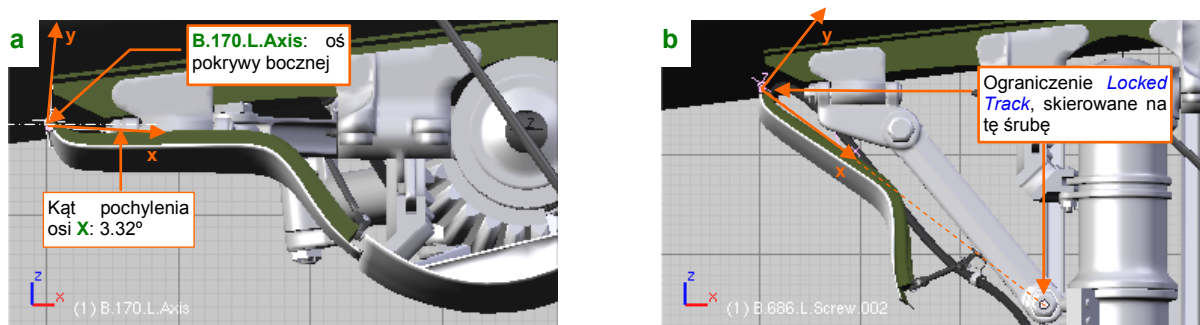


Rysunek 2.5.12 Mechanizacja głównej pokrywy podwozia

To ograniczenie przekaże przesunięcie głównego uchwytu (**X.600.Handle**) na obrót osi pokrywy. Zwróć uwagę na „śledzony” zakres współrzędnych **Z** (*Source:Z*). Choć pełen zakres ruchu uchwytu w tym kierunku to od **Z=-20** do **Z=-10**, to obrót pokrywy rozpoczyna się dopiero w ostatniej fazie składania goleni. (W ten sposób unikam kolizji pokrywy z golenią). Gdy pokrywa jest w pozycji otwartej, można klonować (**Alt-D**) z prawego podwozia popychacz pokrywy i jego okucie. Przed skopiowaniem warto zmniejszyć wpływ ograniczenia *Locked Track*, przypisanego popychaczowi **B.158.R.Stick.001**, do zera. „Rodzicem” kopiowanego zespołu jest obiekt **B.157.L.Nest.001** (to nazwa po już zmianie). Przekształć go tak samo jak zrobiliśmy to z ramką lewego koła (por. str. 77). Przypisz ten zespół do lewej pokrywy gdy jest w pozycji otwartej (Rysunek 2.5.12c).

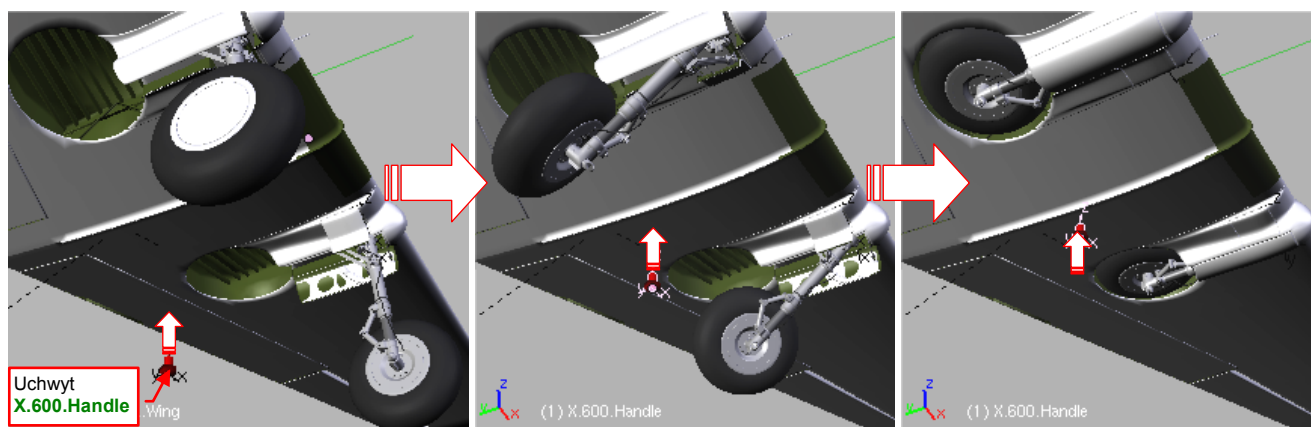


Analogicznie jak pokrywę goleni, „uruchom” pokrywę boczną. Zaczynij od dopasowania „spoczynkowego” kąta osi obrotu (gdy podwozie jest zamknięte) (Rysunek 2.5.13a), a potem przypisz jej odpowiednie ograniczenie *Locked Track* (Rysunek 2.5.13b):



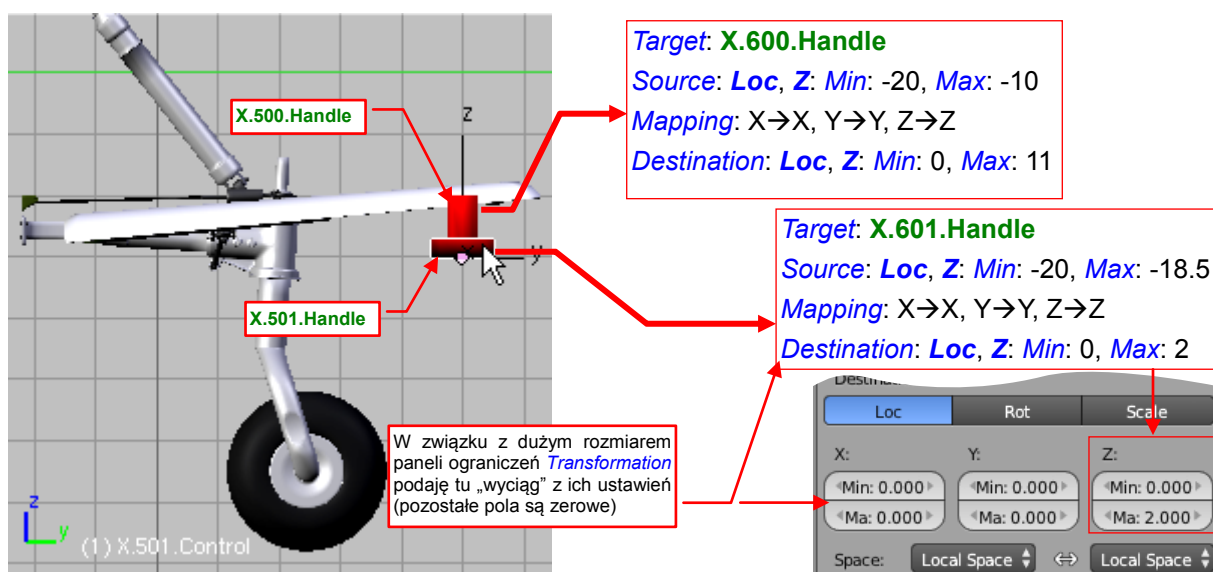
Rysunek 2.5.13 Dopasowanie zakresu kątów pokrywki bocznej

Gdy przypiszesz już jej wszystkie ograniczenia i popychacz, można sprawdzić działanie pełnego zespołu podwozia głównego (Rysunek 2.5.14):



Rysunek 2.5.14 Test składania całego podwozia głównego

Składa się całkiem realistycznie, nieprawdaż? Skoro tak, to po co nam dwie pary uchwytów — jeden do kół głównych, a drugi do kółka ogonowego? Przypisz im odpowiednie ograniczenia *Transformation*, wiążące ich ruch z ruchem uchwytów *X.600.Handle* i *X.601.Handle* (ich szczegóły podaje Rysunek 2.5.15):

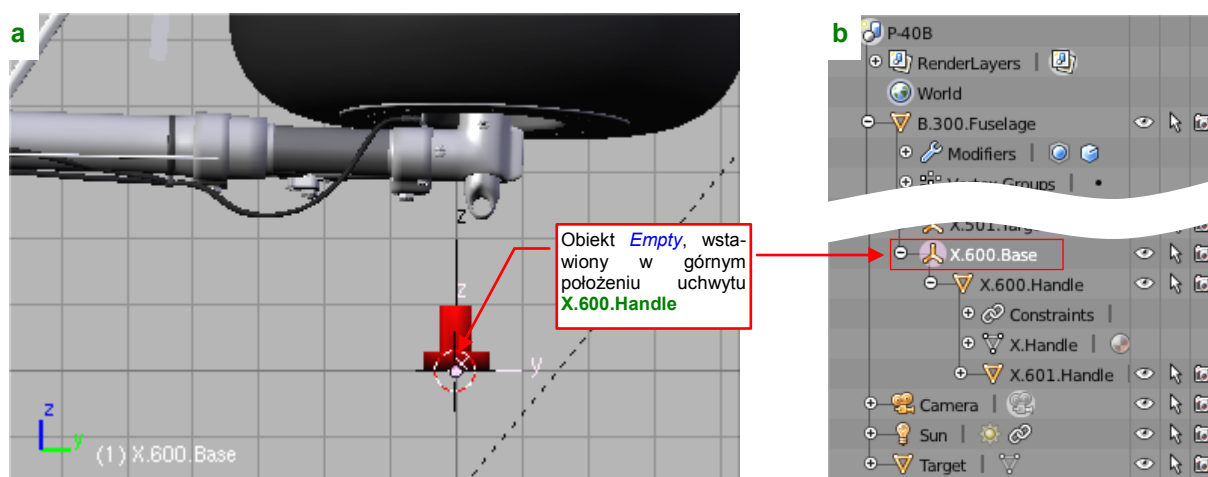


Rysunek 2.5.15 Parametry ograniczeń *Transformation* przypisanych do uchwytów kółka ogonowego

W ten sposób chowaniem i wysuwaniem podwozia naszego modelu steruje jeden uchwyt — *X.600.Handle*, a ugięcie wszystkich amortyzatorów można ustawić za pomocą uchwytu pomocniczego — *X.601.Handle*.

Nim ukończymy ten model, takich uchwytów jak podwozia będzie potrzeba więcej: do sterów, lotek, klap... Przydałaby się do nich jakaś „panel sterowania”, aby nie trzeba było ich szukać po przestrzeni modelu.

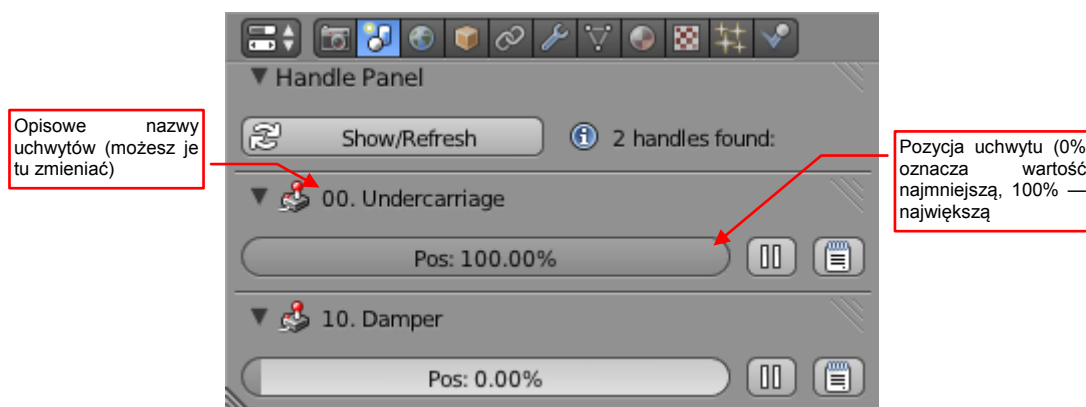
Przygotowując sobie warsztat pracy w Blenderze, parę lat temu stworzyłem właśnie taką uniwersalną panel sterowania modelem. Jest to dodatek **Handle Panel** (dołączony do innych dodatków Blendera, towarzyszących tej książce). Szczegółowy opis, jak się nim posługiwać, znajdziesz na str. 265. Skrypt jest uniwersalny, bo samoczynnie wykrywa w pliku modelu wszystkie uchwyty. Muszą to być po prostu obiekty o nazwie zakończonej przyrostkiem **\*.Handle**, z przypisanym ograniczeniem **Limit Location** (por. str. 266). Muszą być także przypisane (relacją **Parent**) do jakiegoś obiektu nadrzędnego („rodzica”). Środek takiego obiektu jest traktowany przez skrypt jako „położenie domyślne” uchwytu. Załóżmy, że domyślnie podwozie ma być schowane. Stwórz w górnym położeniu uchwytu **X.600.Handle** nowy, pusty (**Empty**) obiekt (Rysunek 2.5.16a), i nadaj mu nazwę **X.600.Base**. Przypisz **X.600.Handle** do **X.600.Base**<sup>1</sup>, a **X.600.Base** — do kadłuba (**B.300.Fuselage**). Hierarchia tego fragmentu powinna wyglądać tak, jak to pokazuje Rysunek 2.5.16b):



Rysunek 2.5.16 Przypisanie do uchwytu „punktu odniesienia”

Z kolei „rodzicem” pomocniczego obiektu, sterującego ugięciem amortyzatora (**B.601.Handle**) jest główny uchwyt podwozia (**B.600.Handle**). W efekcie domyślnym położeniem amortyzatora jest pełne rozciągnięcie (w tej pozycji były rzeczywiście chowane).

Położenie obiektów **X.500.Handle** i **X.501.Handle**, sterujących położeniem kółka ogonowego, uzależniliśmy od uchwytów podwozia głównego. Aby te obiekty nie zostały zidentyfikowane przez skrypt jako uchwyty (ich położenie jest związane z **X.60\*.Handle**), musisz zmienić im nazwę — np. na **X.500.Control** i **X.501.Control**. Na razie **Handle Panel** tworzy tylko dwa panele (Rysunek 2.5.17):



Rysunek 2.5.17 Panel sterowania modelem (na razie tylko dwa uchwyty podwozia)

<sup>1</sup> Przed przypisaniem obiektowi **B.600.Handle** rodzica wyłącz jego ograniczenie **Limit Location**. Potem powoli włącz je z powrotem (zwiększając stopniowo współczynnik **Limit Location:Influence**), sprawdzając czy nie powoduje to jakichś przesunięć uchwytu w nowe położenie.

Dzięki panelowi sterowania nie musisz się zastanawiać, do czego służy konkretny uchwyt. (Parę miesięcy po zakończeniu modelu możesz już mieć taki dylemat). W [Handle Panel](#) udostępnia wszystko czytelnie poopisywane. Z czasem okno tego skryptu zapełni się panelami różnorodnych uchwytów.

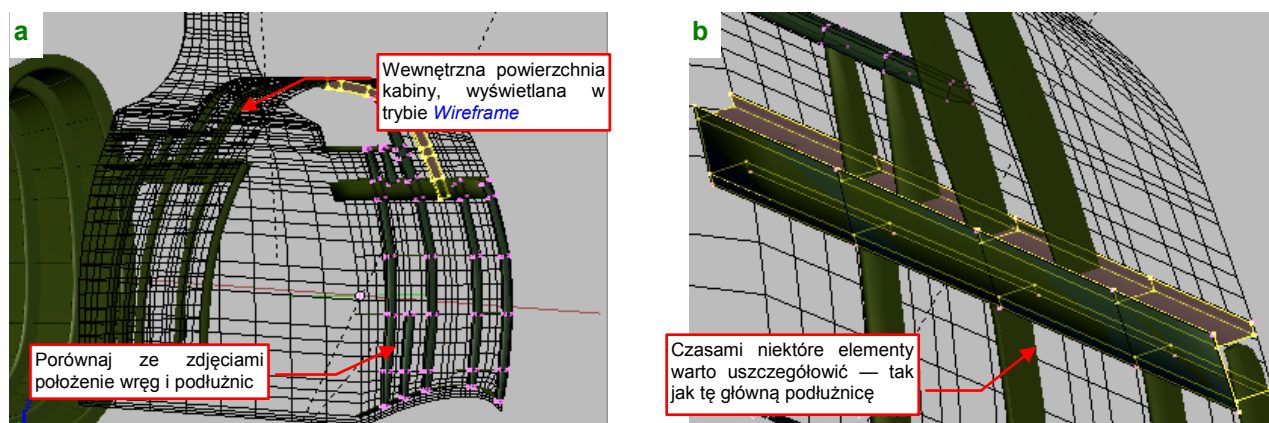
### Podsumownie

- Aby nie komplikować niepotrzebnie siatki, część powierzchni wewnętrznych można namalować wprost na teksturze kamuflażu (str. 75);
- Zespół drugiego koła (prawego, lewego — w zależności od tego, które wykonałeś jako pierwsze) najlepiej jest stworzyć poprzez sklonowanie (str. 77). Potem rezultat tej operacji wymaga jeszcze trochę pracy, m.in. poprawienia nazw skopiowanych obiektów (str. 78);
- „Lustrzane odbicie” zespołu podwozia najprościej uzyskać, zmieniając skalę na ujemną dla kilku wybranych obiektów (str. 77). (Chodzi zazwyczaj o jeden lub dwa elementy, znajdujące się „na szczycie” hierarchii całego zespołu);
- Pokrywy podwozia powstają zazwyczaj wcześniej niż koła i golenie — już podczas modelowania powierzchni samolotu (por. Wyciąg II). Najlepiej dopasować je do reszty zespołu podwozia w pozycji „złożonej” (str. 79 - 80);
- Całe podwozia — główne i ogonowe — można „sprząć” ze sobą tak, by było sterowane pojedynczym zespołem uchwytów (str. 80);
- Na koniec warto uporządkować nazwy wszystkich pomocniczych obiektów. Tylko nazwy „prawdziwych” uchwytów mogą mieć przyrostek „**.Handle**”. Dobrze jest także przypisać je (relacją [Parent](#)) do elementów, których środki wyznaczają „położenie neutralne” (str. 81). To wszystko pozwoli później na prawidłową identyfikację zakresu ruchu całego zespołu przez specjalną „paneli sterowania” ([Handle Panel](#) — str. 81);

## 2.6 Wnętrze kabiny

Budując wnętrze kabiny, trzeba polegać głównie na zdjęciach – jakiegokolwiek rysunki na planach modelarskich są niemal zawsze za mało dokładne. Podczas budowy modelu stosowaliśmy zasadę „jeżeli czegoś nie odwzorowujesz, to przynajmniej przygotuj tak, aby łatwo można było zrobić to później”. Dzięki niej podczas formowania kadłuba przygotowaliśmy wewnętrzną powłokę kokpitu, wraz z wręgami i podłużnicami (por. Wyciąg II).

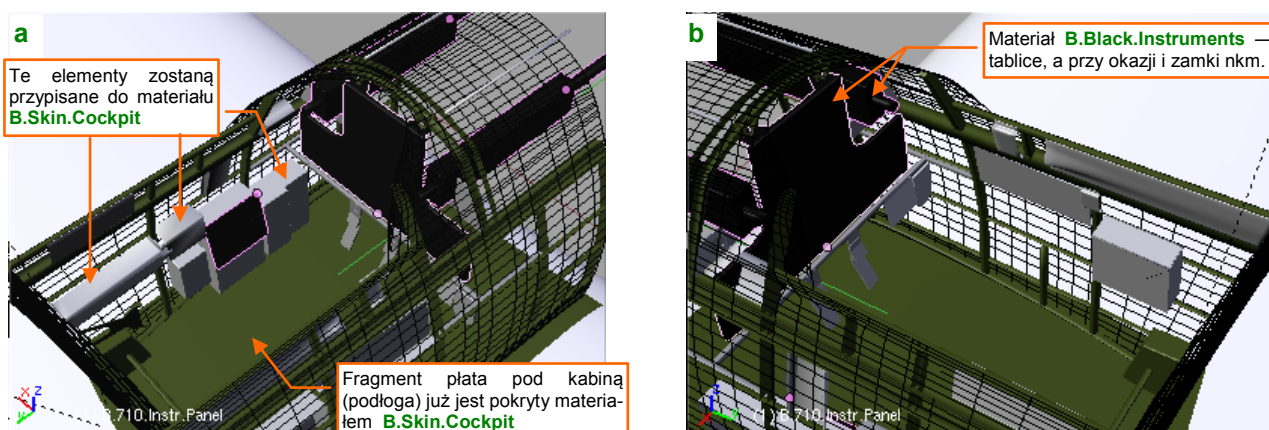
Pracę nad kabiną najlepiej jest zacząć od przełączenia wewnętrznej powłoki z domyślnego trybu wyświetlania *Textured* w tryb *Wire* (zestaw *Object*, panel *Display*). Ułatwia to znacznie dalsze prace, bo możesz teraz widzieć na wskroś poprzez powierzchnię kadłuba (Rysunek 2.6.1):



Rysunek 2.6.1 Przygotowanie szkieletu kadłuba i poszycia kabiny

Kolejnym krokiem jest powtórne zweryfikowanie kształtu i rozłożenia elementów szkieletu wewnątrz kabiny (Rysunek 2.6.1a). Porównaj je starannie z dostępnymi zdjęciami. To bardzo ważne, bo potem krzyżujące się wręgi i podłużnice będą służyły Ci za coś w rodzaju „siatki odniesienia”. Analizując fotografie, będziesz mógł na ich podstawie szacować wielkość pozostałych elementów wyposażenia. Przy okazji możesz zdecydować się na jakieś uszczegółowienia ich kształtu — na przykład zdecydowałem się zmienić przekrój głównej podłużnicy z „C” na „H”, bo taki widać na zdjęciach (Rysunek 2.6.1b).

Zdjęcia kabiny przytłaczają liczbą szczegółów. Nie przejmuj się tym tak bardzo! Przyjrzyj się fotografiom i wyodrębnij na nich elementy podstawowe (te, do których są przymocowane pozostałe) (Rysunek 2.6.2):

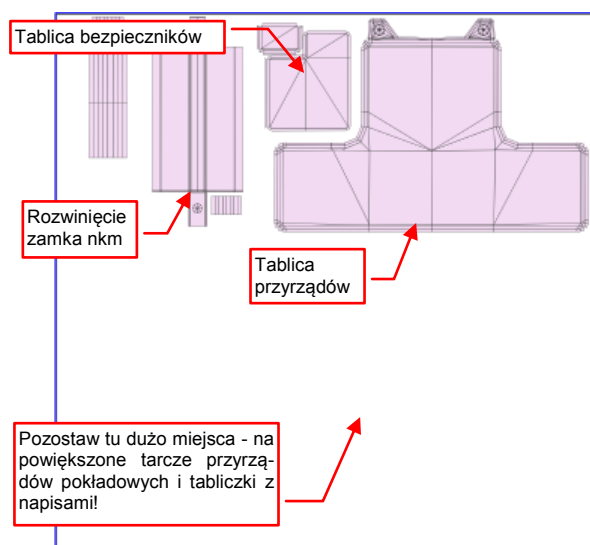


Rysunek 2.6.2 Podstawowe „bryły” wyposażenia kabiny

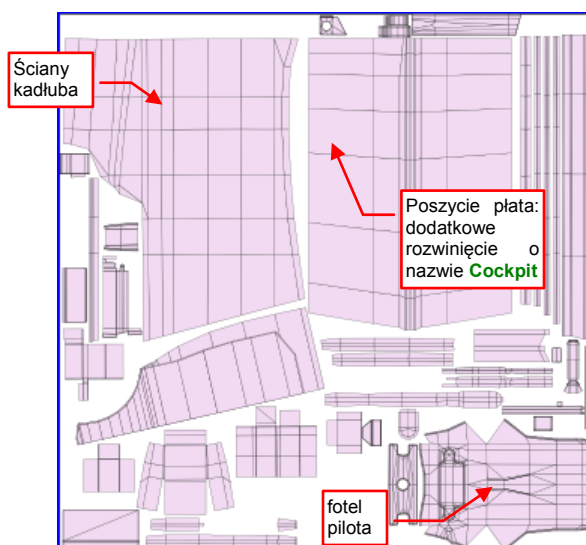
Zazwyczaj zidentyfikujesz i stworzysz tablicę przyrządów (wraz ze stelażem), oraz parę „pudełek” i „płytek”, ciągnących się wzdłuż lewej (Rysunek 2.6.2a) i prawej (Rysunek 2.6.2b) strony kabiny. Krawędzie „pudełek” warto fazować (poleceniem *Bevel* — por. str. 294). Przygotuj także dwa nowe materiały (możesz je skopiować z *B.Inner.Details*). Pierwszym: *B.Skin.Cockpit*, pokryj „wszystko, co nie jest czarne”. Drugim — *B.Black.Instruments* — pokryjesz „wszystko, co czarne” (tablicę przyrządów, pokrętła, itp.).



Materiał **B.Black.Instruments** będzie miał przypisane oddzielne tekstury, na których odwzorujemy wszystkie drobne szczegóły: tabliczki z napisami, tarcze przyrządów pokładowych. Trochę „na doczepkę” zaliczyłem do nich także wystające w kabinie zamki „półcalówek” Browninga. (Na zdjęciach także są czarne i mają taki sam połysk, co tablica przyrządów). Przygotuj rozwinięcia siatek należących do tego materiału. Aby uzgodnić ich położenie w UV, przypisz wszystkie do pomocniczego obrazu o nazwie **Test-Instruments** (Rysunek 2.6.3):



Rysunek 2.6.3 Rozłożenie w UV siatek przypisanych do materiału **B.Black.Instruments** (obraz **Test-Instruments**).

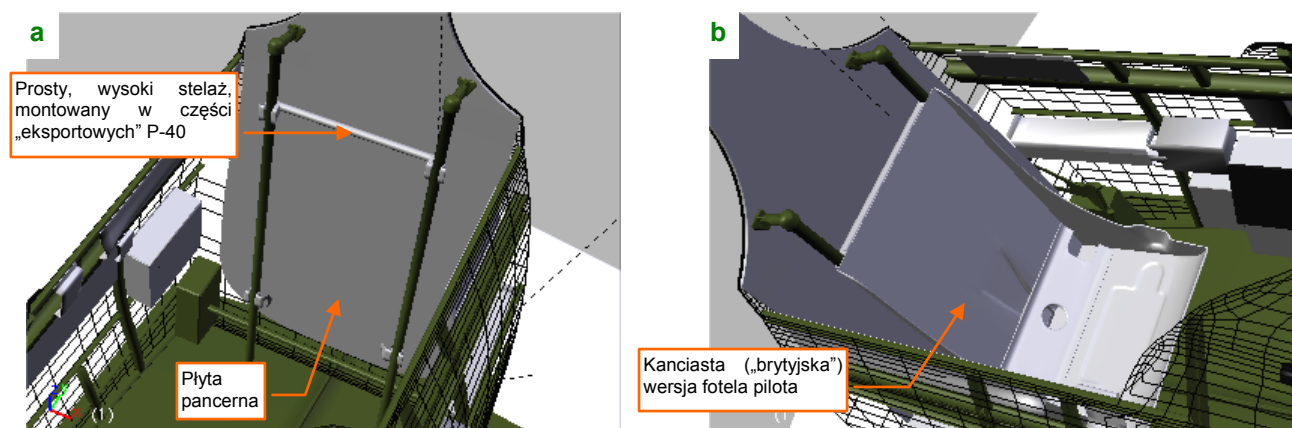


Rysunek 2.6.4 Rozłożenie w UV siatek przypisanych do materiału **B.Skin.Cockpit** (obraz **Test-Cockpit**).

Rozwinięcia ułóż tak, by zostawić na teksturze dużo wolnego miejsca — umieścimy tam powiększone obrazy tarcz instrumentów pokładowych, i różnych innych „opisanych” elementów. (Na przykład: bloku sterowania przepustnicą, mieszanką i skokiem śmigła, pokręteł trymerów, itp.).

Należy oczekiwać, że ciekawscy obserwatorzy będą częściej zaglądać do kabiny niż do luków podwozia. Warto więc zastosować w materiale **B.Skin.Cockpit** odpowiednią mapę nierówności, odwzorowującą łączenia blach i nity, a także mapę barw, odwzorowującą różnorodne przetarcia, zadrapania i zabrudzenia. Aby to było możliwe, rozwiń w UV odpowiednie powierzchnie (Rysunek 2.6.4). By nie było ich zbyt dużo, możesz pominąć drobniejsze elementy, przypisując je do materiału **B.Inner.Details**. Dla uproszczenia rozwinąłem tylko symetryczną połowę kadłuba (zabrudzenia będą lustrzanym odbiciem, ale to nie powinno razić).

Zdecydowałem się odwzorować już na tym etapie fotel pilota (Rysunek 2.6.5b). (Jest najbardziej „wyeksponowany”, i na pewno będzie miejscami zabrudzony, więc trzeba go także wkomponować w rozwinięcie UV).

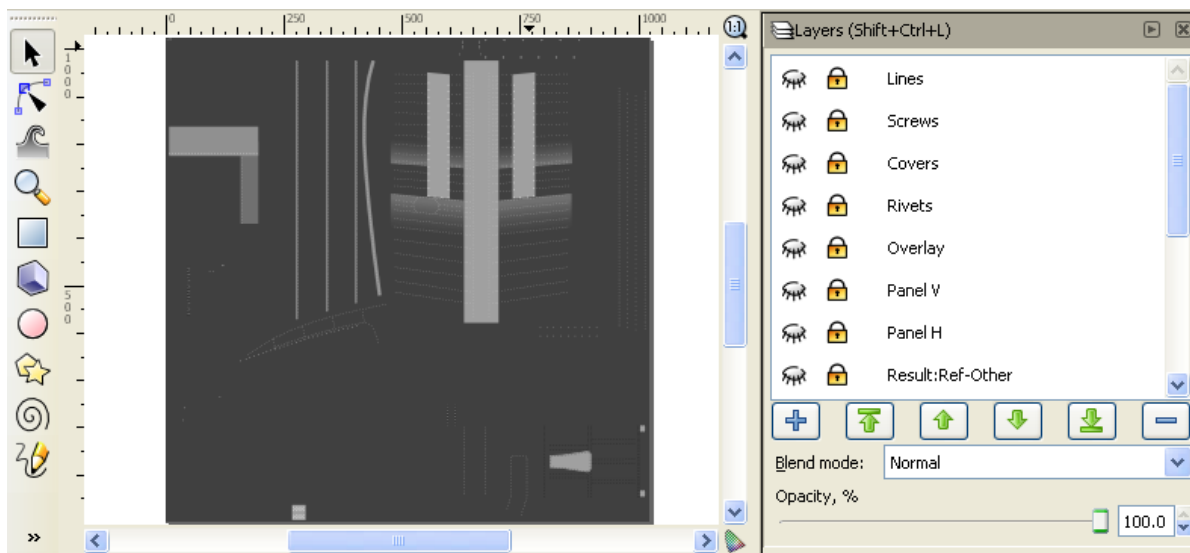


Rysunek 2.6.5 Stelaż i fotel pilota

W P-40 stosowano dwa rodzaje foteli pilota i stelaży, w zależności od odbiorcy: „brytyjski” i „amerykański”. Samoloty AVG miały, wg Hume Batesa ([7]), kanciasty model „brytyjski”, na prostym stelażu (Rysunek 2.6.5a).

Tekstury dla materiału **B.Skin.Cockpit** uzyskamy w ten sam sposób, jak tekstury powierzchni zewnętrznych samolotu (por. Wyciąg III). Przygotujemy je w oddzielnych plikach, w podkatalogu o nazwie **cockpit**. (Uważam, że oddzielny folder jest wygodniejszym do rozróżniania plików z obrazami tekstur, niż dodawania jakichś przedrostków do ich nazw).

Najpierw wyeksportuj z Blendera rozwinięcia siatek (tych, które pokazywał Rysunek 2.6.4) w formacie SVG, i złóż je w Inkscape (por. Wyciąg III), w pliku **cockpit\cockpit.svg**. Obraz **cockpit\cockpit.svg** będzie mniejszy od obrazu powierzchni zewnętrznych — nadaj mu rozmiar, powiedzmy, 1024x1024px. Potem przygotuj podobne warstwy, jak w pliku **skin.svg** (por. Wyciąg III). (Niektóre możesz pominąć, bo ten rysunek nie będzie miał takich elementów, jak cienie pod nitami lub zagłębienia pod śrubami). Narysuj na nich obraz nierówności wnętrza kabiny (Rysunek 2.6.6):

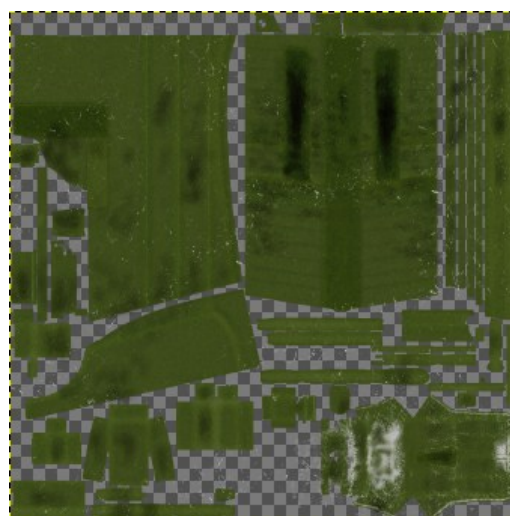


Rysunek 2.6.6 Przygotowywanie w Inkscape tekstury nierówności (plik **cockpit\cockpit.svg**)

Sprawdź w Blenderze, jak ta mapa nierówności „leży” na siatkach, i nanieś odpowiednie poprawki. Gdy będzie gotowa — połącz w Inkscape poszczególne warstwy w grupy (por. Wyciąg III). Skomponuj z klonów tych grup na oddzielnej warstwie obraz map nierówności (**Result:Nor**). Używając odpowiednich filtrów (por. Wyciąg III) przygotuj z klonów tych samych warstw mapę odbić (**ref.png**). Podobnie jak w pliku **skin.svg**, będzie złożenie trzech warstw: **Result:Ref-Blur**, **Result:Ref-Grunge**, **Result:Ref-Other** (Rysunek 2.6.7):



Rysunek 2.6.7 Tekstura odbić (**cockpit\ref.png**)



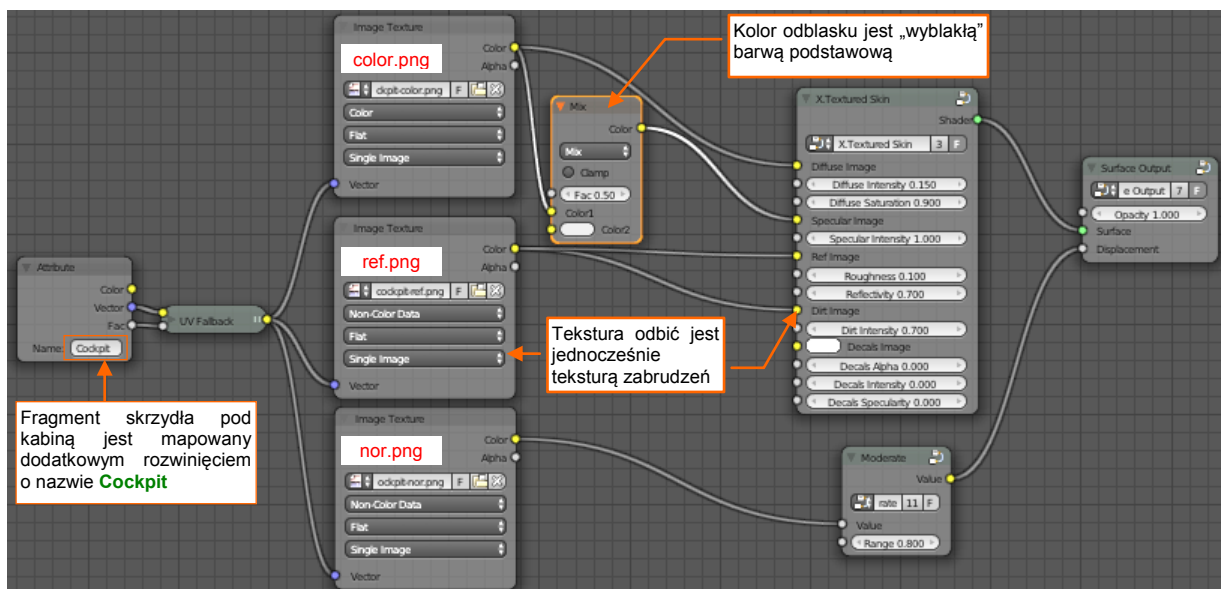
Rysunek 2.6.8 Tekstura barwy (**cockpit\color.png**)

Obraz rozwinięć UV wyeksportuj z Inkscape do pliku rastrowego, i wczytaj do Gimpa (jako plik **cockpit\cockpit.xcf**). Posługując się tymi samymi technikami, co w sekcji „Tekstura barwy” (Wyciąg III) stwórz obraz tekstury barwy. Rezultat wyeksportuj do pliku **cockpit\color.png**.

Przygotowaliśmy w ten sposób trzy obrazy dla odpowiednich tekstur materiału **B.Skin.Cockpit**:

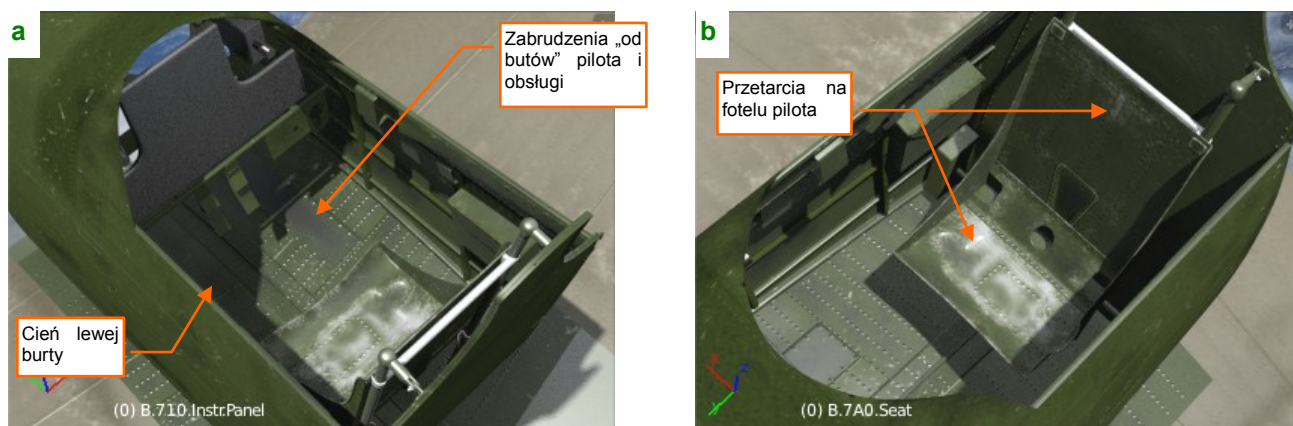
- plik *cockpit\nor.png* zawiera obraz dla tekstury nierówności (**cockpit-nor**);
- plik *cockpit\ref.png* zawiera obraz dla tekstury odbić (**cockpit-ref**);
- plik *cockpit\color.png* zawiera obraz dla tekstury barwy (**cockpit-color**);

Na razie użyłem tych tekstur tak samo jak w materiale **B.Skin.Camouflage**, podłączając je do komponentu **X.Textured Skin**. Powstała w ten sposób jego uproszczona wersja — **B.Skin.Cockpit** (Rysunek 2.6.9):



Rysunek 2.6.9 Schemat materiału **B.Skin.Cockpit** (uproszczona wersja **B.Skin.Camouflage**)

Rysunek 2.6.10 przedstawia render, stworzony z zastosowaniem tego materiału. (Aby rozproszyć nieco cienie, oświetlam kabinę pomocniczym światłem typu **Sun** o **Strength** = 5, **Size** = 1, **Cast Shadows** = Off):



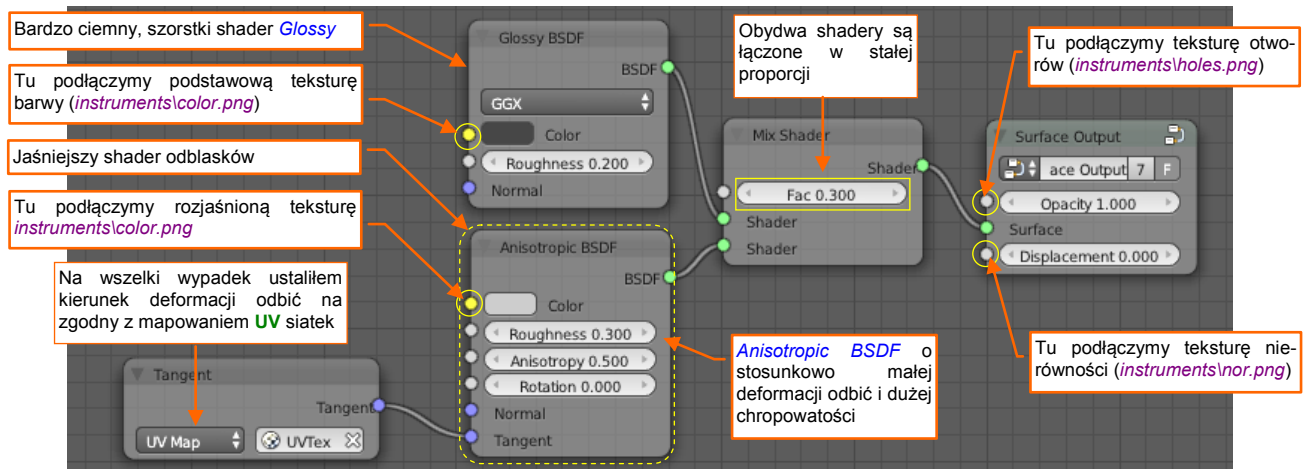
Rysunek 2.6.10 Rezultat zastosowania materiału **B.Skin.Cockpit**

Podłogą kabiny jest górna powierzchnia płata (Rysunek 2.6.10a). Obszary zakryte, tak jak ten, były nitowane za pomocą zwykłych nitów z łbami wypukłymi (a nie wpuszczanymi, jak na powłokach zewnętrznych). Jak przystało na podłogę, musi być zabrudzona, szczególnie tam, gdzie najczęściej stawiano stopy: w okolicach pedałów orczyka. Dodatkowo wzbogaciłem ją serią innych niewielkich zabrudzeń i zadrapań.

Fotel pilota (Rysunek 2.6.10b) należy zawsze do najbardziej „wytartych” elementów kabiny. Ten sam typ siedzenia był wykorzystywany w innym samolocie, produkowanym dla Wielkiej Brytanii: North American „Mustang”. Dokumentacja tego detalu to kilka zdjęć z restauracji P-51A, które znalazłem w Internecie. Wygląda na to, że o ile siedzenia pilota P-40 w USAAF pozostawiano w kolorze naturalnego duralu, to te fotele były malowane. W przypadku „Mustanga” były szaroniebieskie (farba **Dull Grey**)? Mogę tylko spekulować, w jakim kolorze były w samolotach AVG, więc pozostawiłem je w **Curtiss Green**, tak jak resztę kabiny.



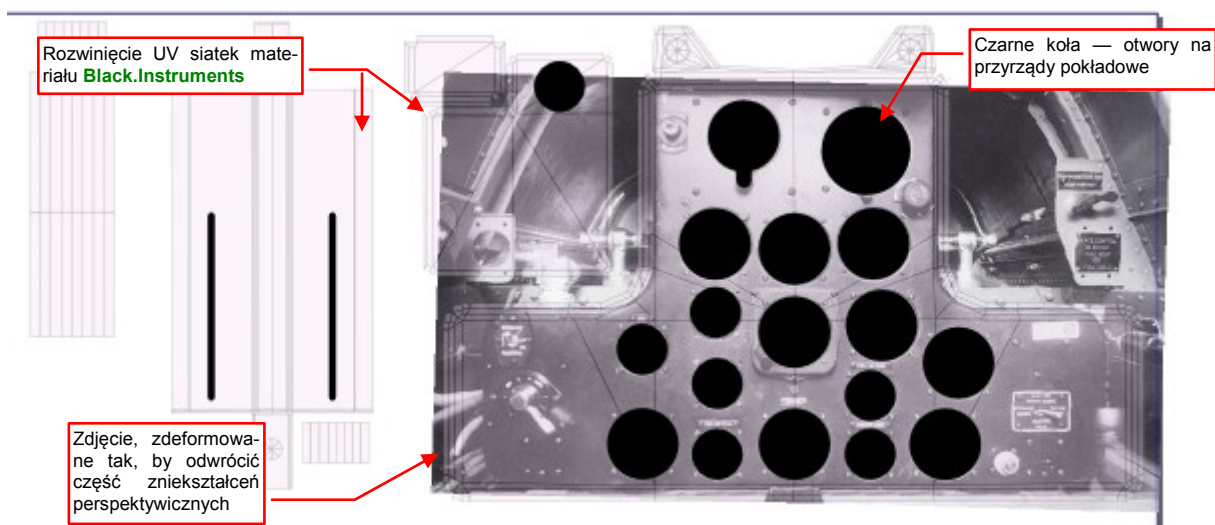
Materiał **B.Black.Instruments** odpowiada niemal czarnej, połyskliwej powierzchni tablicy przyrządów. Dla tak ciemnej barwy nie można zastosować naszego uniwersalnego shadera *Glossy Paint* (jest w tym przypadku za mało połyskliwy). Musiałem poszukać innego rozwiązania. Ostatecznie zdecydowałem się wykorzystać połączenie 70% szorstkiej lecz połyskliwej powierzchni (shader *Glossy*) i 30% jeszcze bardziej szorstkiej lecz jaśniejszej anizotropowej<sup>1</sup> powierzchni metalu (shader *Anisotropic* — Rysunek 2.6.13):



Rysunek 2.6.11 Wstępny schemat materiału **B.Black.Instruments**

- Ten schemat **B.Black.Instruments** to moja propozycja rozwiązania dla takiego materiału. Nie traktuj go jako czegoś ostatecznego, być może uda Ci się uzyskać lepszy efekt w odmienny sposób.

Tablica przyrządów jest najbardziej wyeksponowanym fragmentem kabiny pilota, więc warto jej poświęcić nieco więcej uwagi. Zaczniemy od naniesienia w Inkscape miejsc na tarcze przyrządów (Rysunek 2.6.12):



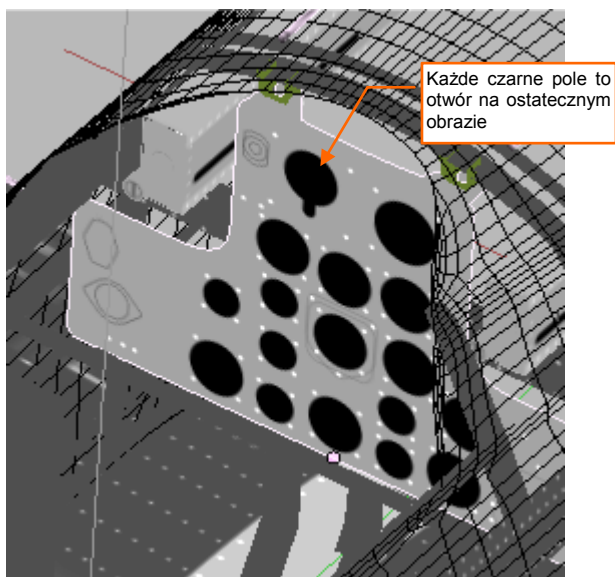
Rysunek 2.6.12 Przygotowanie tekstury otworów (głównie z myślą o tablicy przyrządów)

Pod rozwinięcia *UV* podstawilem zdjęcie prawdziwej tablicy, i naniosłem w odpowiednich miejscach czarne koła. Stworzony w Inkscape rysunek wyeksportuj do pliku rastrowego *instruments\holes.png*.

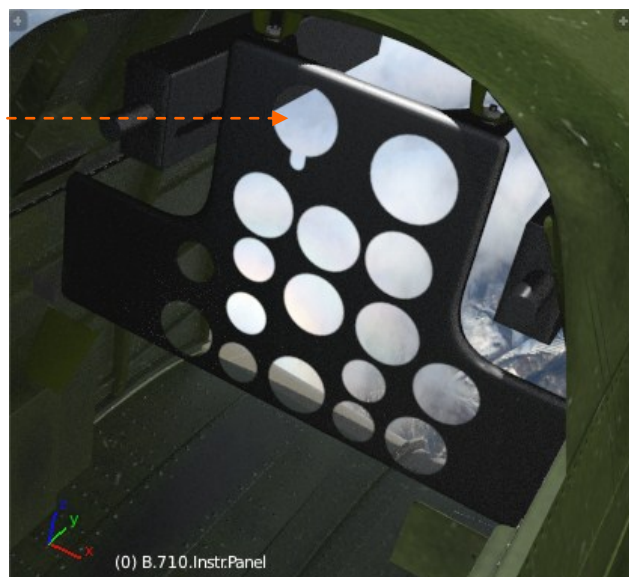
<sup>1</sup> Powierzchnia anizotropowa to taka, która zniekształca obicie światła w określonym kierunku. Taką właściwość ma np. dno garnka wytłoczonego ze stali nierdzewnej. (Jest to rezultatem zastosowanej obróbki tego fragmentu metalu). W Cycles przygotowano specjalny shader który odpowiada takiej powierzchni: to *Anisotropic BSDF*. Intensywnością deformacji steruje parametr *Anisotropy*. Gdy jest równy 0, żadna deformacja nie występuje. Gdy *Anisotropy* = 1, odbicia są już zupełnie zdeformowane. Anizotropowa deformacja odbić zwiększa się także wraz ze wzrostem szorstkości (*Roughness*) shadera. Gdy *Roughness* = 0, deformacja pojawia się dopiero dla *Anisotropy* = 1.0 (skokowo). Dla większych szorstkości pojawia się wcześniej, np. przy *Roughness* = 0.1 jest już widoczna dla *Anisotropy* = 0.5. Kierunkiem deformacji odbić steruje wektor podłączony do jego wejścia *Tangent*. Podłącza się tam zazwyczaj węzeł *Tangent*, który potrafi wytworzyć albo deformację radialną (taką jak na dnie kuchennego garnka), albo zorientowaną zgodnie z podanym w parametrach węzła rozwinięciem UV.



W trybie wyświetlania **Textured** podstaw go w Blenderze pod rozwinięcia siatek (jako plik przypisany do obrazu **Test-Instruments** — Rysunek 2.6.13). Sprawdź, czy nie są zdeformowane:



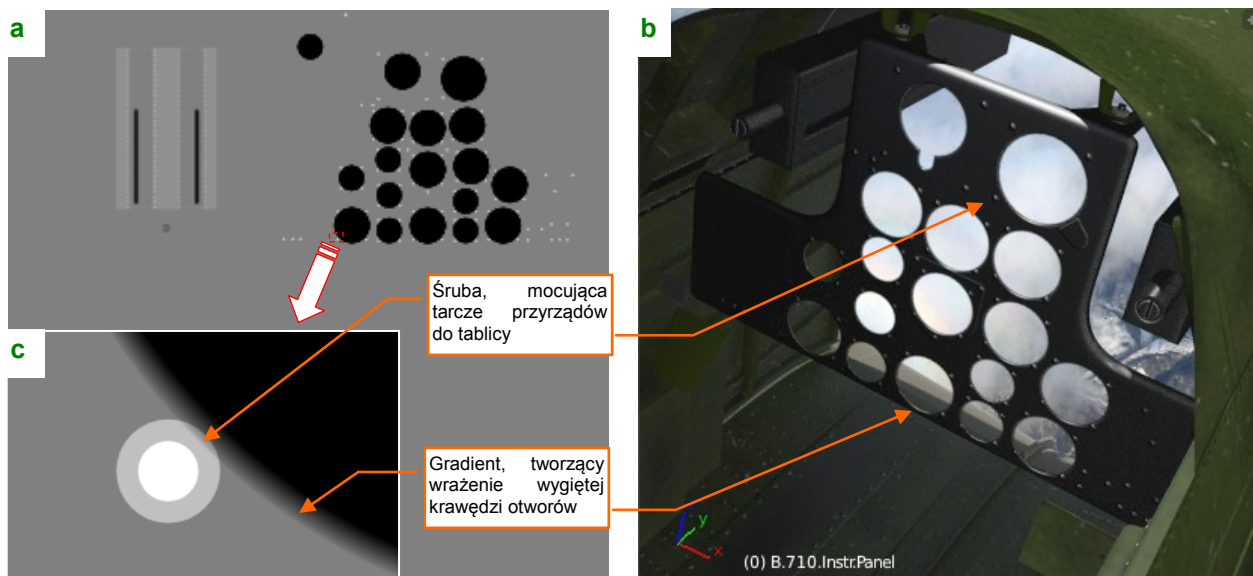
Rysunek 2.6.13 Dopasowanie obrazu otworów do rozwinięcia siatek (**3D View**, tryb **Textured**)



Rysunek 2.6.14 Test zastosowania tekstury otworów

Tarcze każdego wskaźnika, umieszczonego na tablicy, znajduje się w odpowiednim otworze. Przy krawędziach tych otworów widać okrągłe łby śrub, mocujących przyrządy. Śruby odwzorujemy za pomocą mapy nierówności. Otwory na tarcze przyrządów można byłoby zamodelować w siatce (wielu modelarzy tak robi). Ja jednak proponuję tu alternatywne rozwiązanie: zamiast wycinać dziury w siatce — uzyskajmy je za pomocą tekstury przejrzystości (Rysunek 2.6.14). Efekt jest równie dobry jak przy modelowaniu, nawet przy użyciu obrazu o umiarkowanej rozdzielczości. (Użyłem tu pliku *instruments/holes.png*, który ma rozdzielczość 1024x1024 pikseli. Odpowiada takiemu rozwinięciu UV siatek, jakie pokazuje Rysunek 2.6.3 ze str. 84).

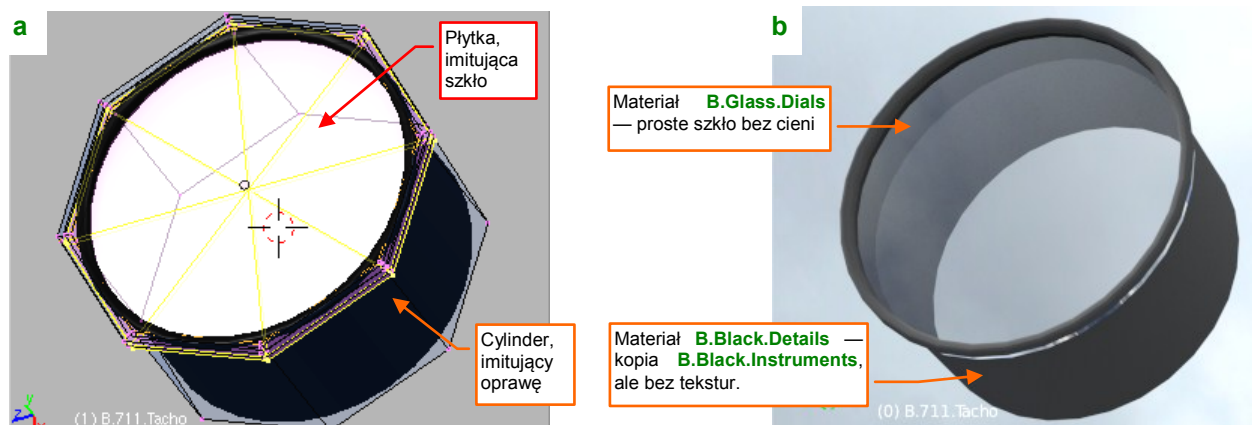
Blacha na krawędziach otworów w tablicy była wygięta do środka (typowy efekt technologii tzw. cięcia gumą). Aby odtworzyć ten szczegół, przygotowałem odpowiednią mapę nierówności (Rysunek 2.6.15a):



Rysunek 2.6.15 Szczegóły tekstury nierówności

Otwory na obrazie (*instruments/holes.png*) tekstury nierówności są otoczone obrysem, wypełnionym odpowiednim gradientem kołowym (Rysunek 2.6.15c). Daje to całkiem poprawny rezultat (Rysunek 2.6.15b).

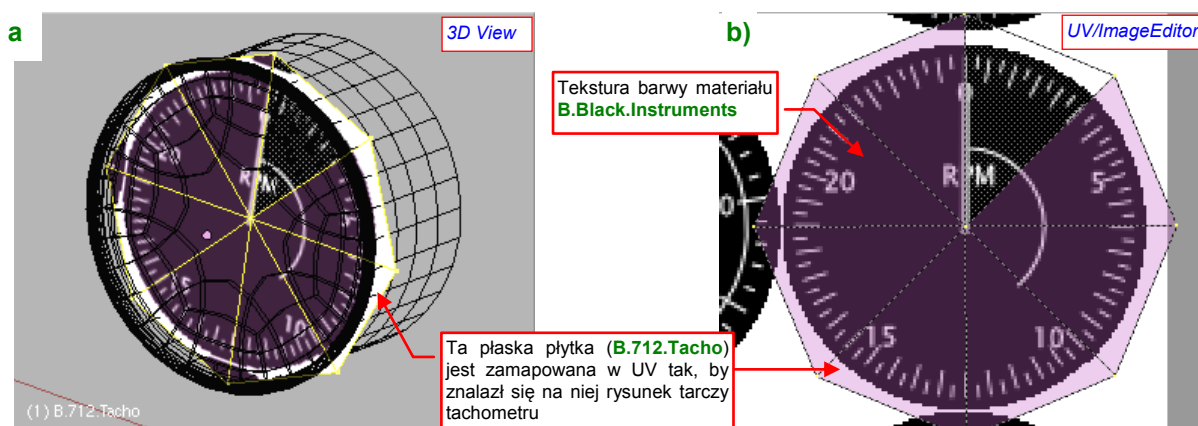
W tak przygotowane otwory wstawimy przyrządy. Większość z nich miała niemal identyczne oprawy (na tablicy występują w zasadzie tylko dwie średnice tarcz — wyjątkiem jest sztuczny horyzont). W tej sytuacji proponuję stworzyć większość opraw jako klony jednej siatki **B.711.Dial** (Rysunek 2.6.16):



Rysunek 2.6.16 „Oprawa” wskaźnika

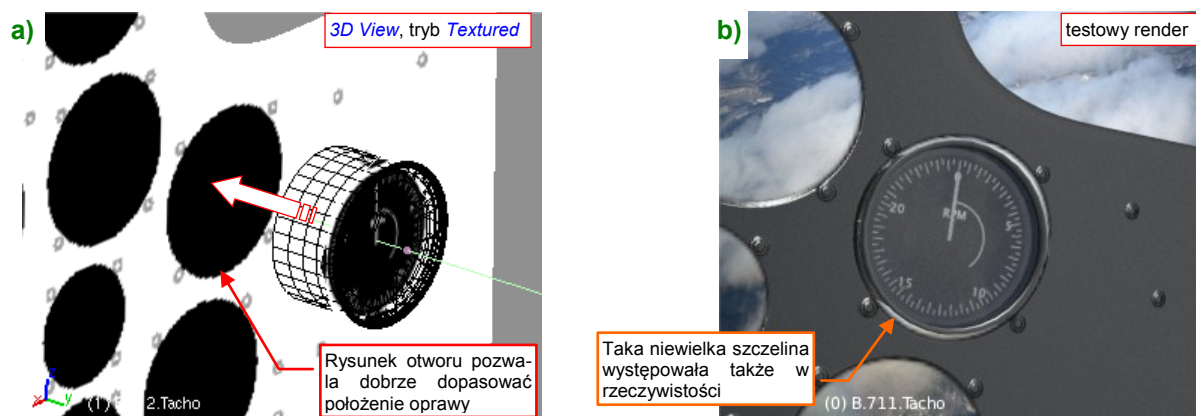
Oprawa składa się z cylindra, pokrytego czarnym, połyskliwym materiałem (**B.Black.Details**). W środek cylindra wstawiłem płytkę imitującą szkło (Rysunek 2.6.16a). Ta część jest pokryta materiałem **B.Glass.Dials** — prostym szkłem o przejrzystych cieniach (por. str. 326). Testowy render tej części przedstawia Rysunek 2.6.16b).

W środku oprawy umieściłem oddzielny obiekt. W tym przypadku to **B.712.Tacho** — płaska płytką, na której znajduje się rysunek tarczy przyrządu (pokryta materiałem **B.Black.Instruments**) (Rysunek 2.6.17):



Rysunek 2.6.17 Tarcza wskaźnika

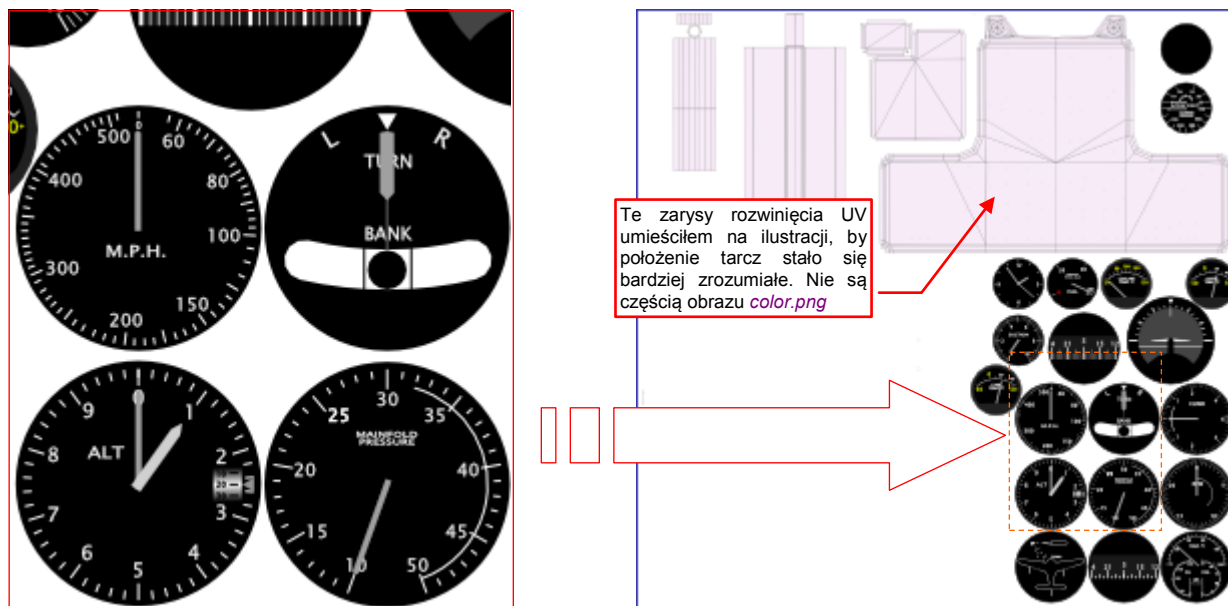
Potem pozostaje tylko umieścić oprawę w tablicy (najlepiej w trybie **Textured**, by trafić dobrze w otwór — Rysunek 2.6.18a). Rezultat na renderze pokazuje Rysunek 2.6.18b):



Rysunek 2.6.18 Osadzenie wskaźnika w tablicy

Zwróć uwagę na szczelinę pomiędzy krawędzią otworu w tablicy przyrządów a oprawą tarczy wskaźnika (Rysunek 2.6.18b). Takie same szczeliny widać na zdjęciach z prawdziwych kabin P-40. W tej chwili nieco rażą, bo prześwituje przez nie obraz otoczenia. Przestaną być widoczne gdy włączysz warstwy z nosem kadłuba.

Wykonaj wszystkie wskaźniki tablicy przyrządów w ten sam sposób co **B.712.Tacho**: jako zespół klon siatki oprawy (**B.712.Dial**) i „indywidualnej” tarczy. Muszę przyznać, że najbardziej pracochłonną czynnością było ich narysowanie. (Wskazówki dot. szybkiego rysowania skali na tarczy przyrządu znajdziesz na str. 174). Stworzyłem je w Inkscape, na podstawie dostępnych zdjęć (Rysunek 2.6.19):



Rysunek 2.6.19 Tekstura barwy materiału **B.Black.Instruments** z narysowanymi tarczami wskaźników

Tarcze są dość duże — ich rozmiar dobrałem tak, by po eksporcie do obrazu rastrowego *instruments\color.png*, o wymiarach 1024x1024, nadal wyglądały w miarę dokładnie. Stworzenie takiego rysunku wymagało kilku godzin pracy. Rozdzielczość większości zdjęć kabiny, którymi dysponowałem, nie pozwalała na dokładne odczytanie cyfr i opisów na tarczach wskaźników. Czasami musiałem szukać podobnych przyrządów w konstrukcjach pokrewnych, np. w P-51. Rezultat wyszedł jednak dość zadowolająco (Rysunek 2.6.20):



Rysunek 2.6.20 Render tablicy przyrządów, po wstawieniu wszystkich wskaźników

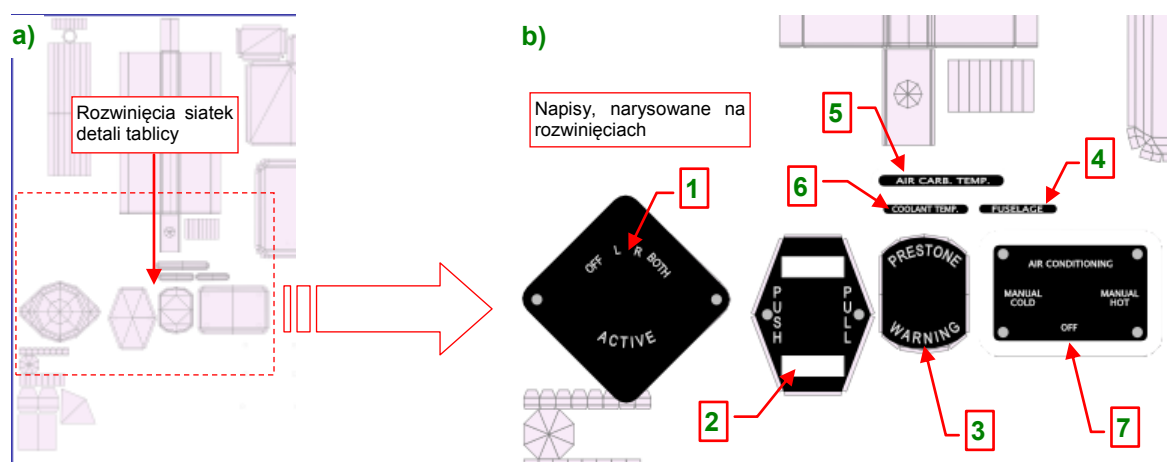


Teraz pozostaje uzupełnić tablicę przyrządów o kilka ostatnich detali (Rysunek 2.6.21):



Rysunek 2.6.21 Tablica przyrządów, uzupełniona o pozostałe szczegóły

Umieszczono na niej kilka plakietek z napisami (4, 5, 6, 7), wskaźnik ostrzegawczy przegrzania silnika (3), oraz parę przełączników (1, 2, 7). Siatki elementów, na których znajdował się tekst lub inny znak graficzny warto nanieść na teksturę barwy *instruments\color.png* (Rysunek 2.6.22a):



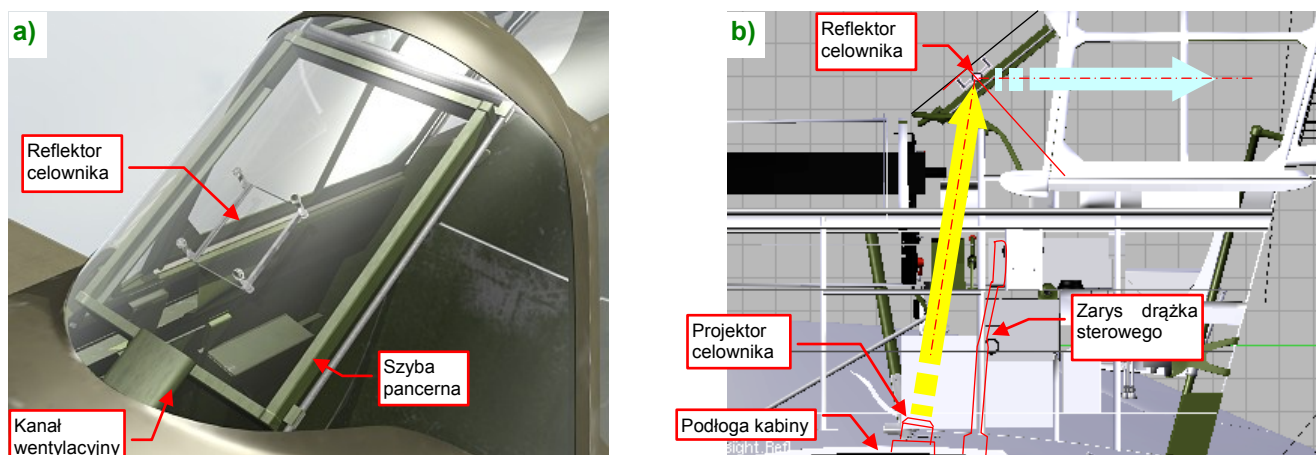
Rysunek 2.6.22 Napisy na detalach tablicy przyrządów — kolejny fragment obrazu *instruments\color.png*

(Nie będę tu opisywał tworzenia i rozwijania siatek, bo są trywialne — sądzę, że sam już dobrze wiesz, jak je wykonać). Rozwinięcia dodałem do warstwy UV w pliku *instruments\instruments.svg*. Posługując się Inkscape, naniosłem na nie odpowiednie napisy i inne elementy (np. białe krawędzie). Na warstwie **Screws** dodałem także do grupy **#Screws** śruby, umieszczone w narożnikach plakietek. Po wygenerowaniu kolejnej wersji mapy kolorów (*instruments\color.png*) i nierówności (*instruments\nor.png*) uzyskałem efekt jak na renderze, który pokazuje Rysunek 2.6.21.

Zwróć może uwagę (por. Rysunek 2.6.21), że w niektórych elementach tablicy przyrządów wykorzystałem inne materiały, niż **B.Cockpit.Instruments**. Uchwyt przełącznika iskrowników (1) był czerwony, więc cały jest w materiale **B.Red.Details**. (To kopia materiału **B.Skin.Inner** ze zmienionym kolorem — dla drobnych detali bez rozwinięć UV). Podobnie przełączniki na plakietkach 2 i 7 to klony tej samej siatki, przypisane do materiału **B.Steel.Smooth** (bo ten materiał zapewnia im najbardziej odpowiedni „odblask”).

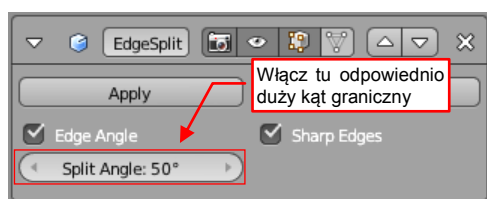


Tablica przyrządów jest już gotowa. To jednak dopiero „przygrywka” do dalszej pracy nad wnętrzem kabiny. Czeką nas jeszcze wykonanie mnóstwa drobiazgów. Rysunek 2.6.23a) przedstawia wykończone wnętrze wiatrochronu:



Rysunek 2.6.23 Szczegóły wnętrza wiatrochronu

Pomiędzy prętami owiewki umieściłem ramkę z szybą pancerną. Szyba miała grubość 1.5 cala (38 mm). Nie byłem pewien, czy miała taki sam zielonkawy odcień, jak na P-40E. Przypisałem jej więc ten sam materiał „szkła”, jaki został użyty w pozostałych szdach owiewki (**B.Glass.Canopy**).



Rysunek 2.6.24 Sposób na zaokrąglenia drobnych siatek — duża wartość **Split Angle**

istocie przez cztery ściany. Cały trick polega na doborze w modyfikatorze **Edge Split** tego obiektu odpowiedniej wartości granicznego kąta pomiędzy ścianami **Split Angle** — na nieco większą niż 45° (Rysunek 2.6.24).

W maszynach produkowanych dla USAAC przed szybą pancerną był umieszczony reflektor celownika (Rysunek 2.6.23a). Projektor celownika była montowany nietypowo, na podłodze kabiny (Rysunek 2.6.23b). To rozwiązanie odziedziczone po P-36. Samoloty produkowane dla Wielkiej Brytanii miały klasyczny, jednoczęściowy celownik refleksyjny za szybą pancerną. Kłopoty mieli za to zbrojmistrze AVG, otrzymawszy brytyjskie samoloty bez celowników. Po wielu „podchodach” udało im się uzyskać amerykańskie projektory podłogowe (takie, jakie pokazuje Rysunek 2.6.23b). Musieli jednak sami do nich dorobić prowizoryczne reflektory (Rysunek 2.6.25).

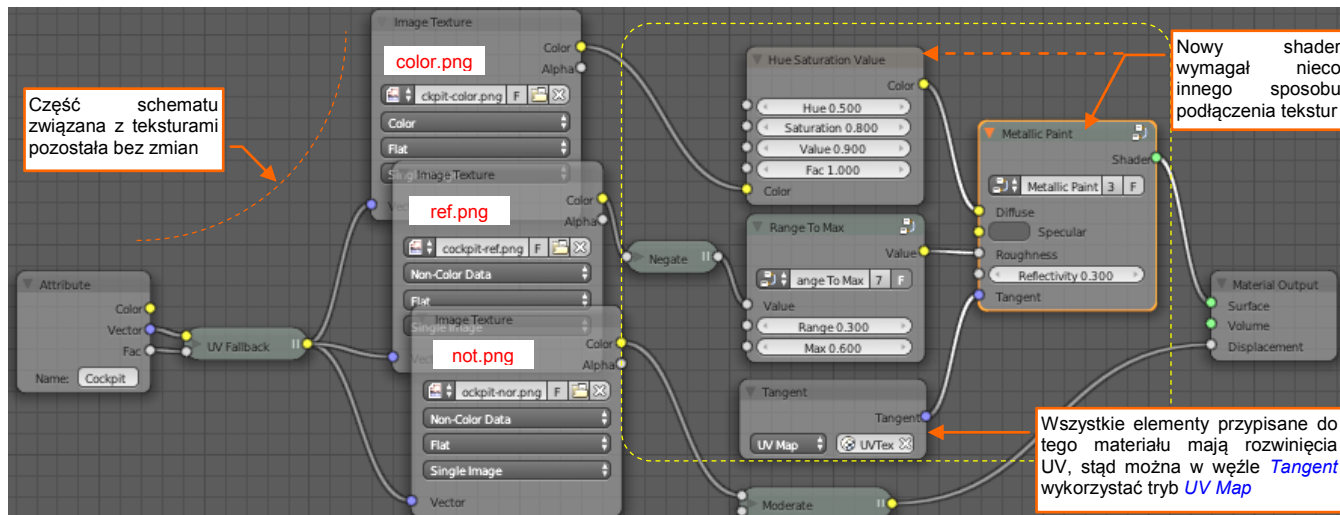


Rysunek 2.6.25 Reflektor celownika, dorabiany dla samolotów AVG





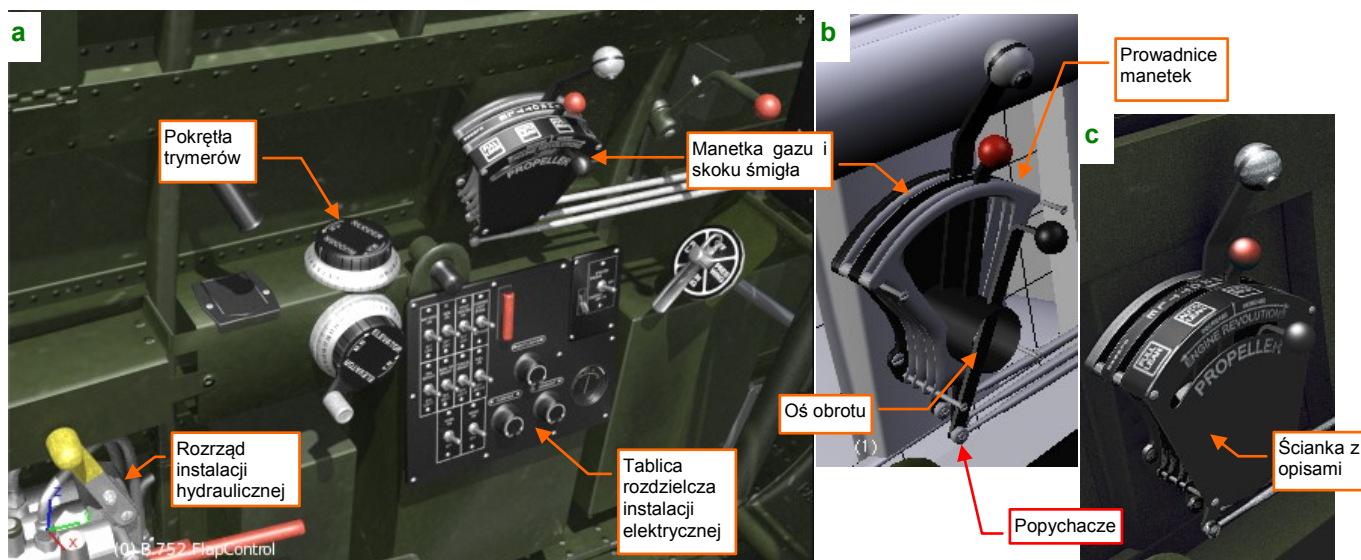
Tak przygotowany shader użyłem w materiale **B.Skin.Cockpit**. Wcześniej wykorzystywaliśmy tu grupę **X.Textured.Skin** (por. str. 86, Rysunek 2.6.9), która wewnętrznie była oparta na shaderze **Gloss Paint**. Uzyskany w ten sposób efekt był zbyt matowy. Na przykład, mimo zastosowania pomocniczego źródła oświetlenia, nie było widać wyraźnie napisu wytłoczonego na powierzchni pedałów orczyka (w takim ujęciu jak to, pokazane na str. 93, Rysunek 2.6.26). Dlatego, aby podkreślić podobne detale, spróbowałem wstawić w miejsce grupy **X.Textured.Skin** nowy zespół węzłów, oparty o shader **Metallic Paint** (Rysunek 2.6.27):



Rysunek 2.6.28 Alternatywna wersja **B.Skin.Cockpit**, wykorzystująca shader **Metallic Paint**

Uważam, że ta wersja materiału wygląda lepiej. W każdym razie odbija światło w inny sposób. Wykorzystałem ją do uzyskania widoku pokazywanego przez Rysunek 2.6.26, i będę wykorzystywał w dalszej części tej sekcji.

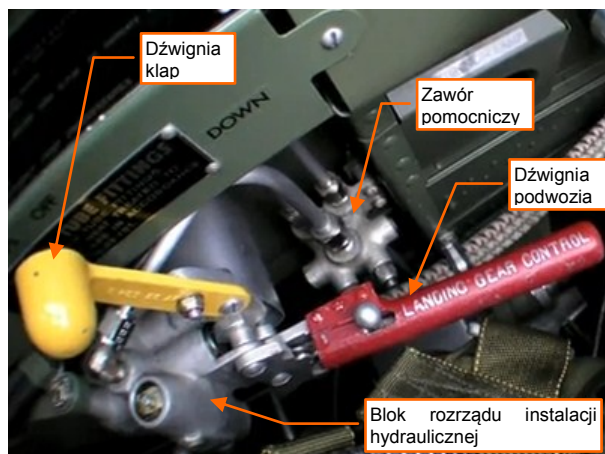
W tym momencie mamy gotowy cały przód kabiny, do wysokości tablicy przyrządów. Czas zabrać się za kolejny obszar — lewą burtę (Rysunek 2.6.29a).



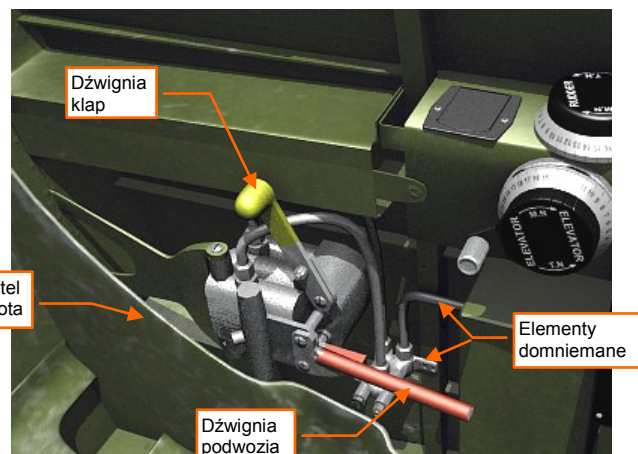
Rysunek 2.6.29 Detale lewej burty.

Na lewej burcie znajduje się większość różnorodnych dźwigni i pokręteł. W szczególności zespół manetek silnika (przepustnicy i mieszanek) oraz skoku śmigła (Rysunek 2.6.29b,c). Modelarze zazwyczaj odwzorowują obudowę tego zespołu jako trójkątne „pudełko”. Ze zdjęć wynika jednak, że była to raczej grupa blaszanych prowadnic (Rysunek 2.6.29b), bez jakichkolwiek ścianek bocznych. Z przodu była do niej tylko przykręcona „maskownica”, z napisami eksploatacyjnymi (Rysunek 2.6.29c). Innym szczegółem lewej burty jest tablica rozdzielcza instalacji elektrycznej (Rysunek 2.6.29a). Wykorzystałem na niej wiele klonów przełącznika, użytego po raz pierwszy na tablicy przyrządów (por. str. 91).

Kolejnym detalem jest rozrząd instalacji hydraulicznej, z dźwigniami: klap i podwozia (Rysunek 2.6.30):



Rysunek 2.6.30 Zawór, sterujący klapami i podwoziem (P-40E)



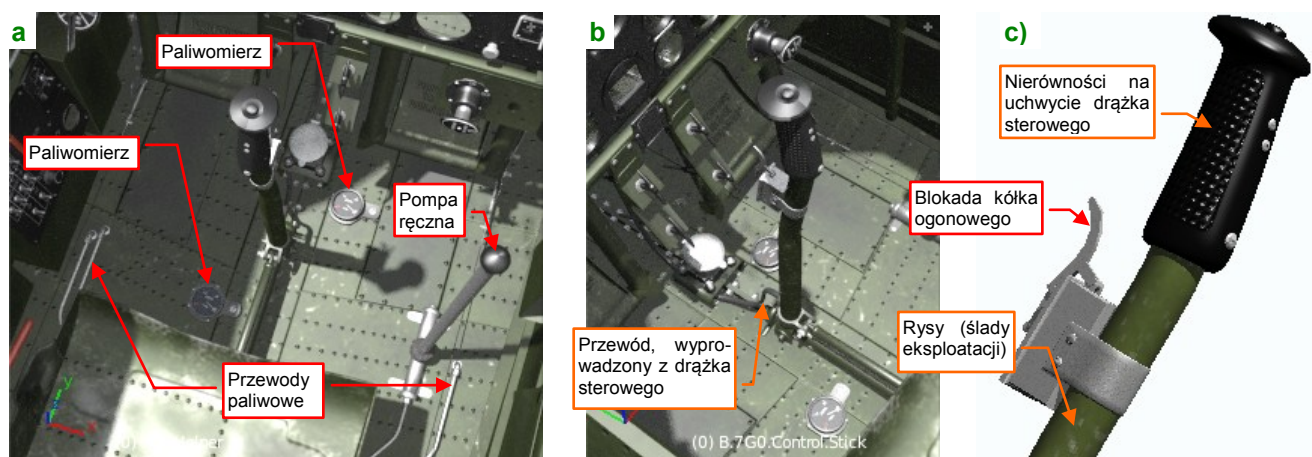
Rysunek 2.6.31 Odwzorowanie zaworu w naszym modelu

Tylko na dwóch zdjęciach pewnego P-40C w jednym z muzeów w USA dostrzegłem wystającą zza fotela dźwignię sterującą klapami skrzydłowymi. W dodatku ten egzemplarz nie ma całkiem oryginalnego wyposażenia: wygląda na to, że zespół sterowania silnikiem zaadaptowano z P-40E. Nie widać tam też śladu po przewodach, zasilających ten rozrząd. (Tak w ogóle, to w jego kabinie brak jeszcze wielu innych szczegółów).

Korpus rozrządu był na pewno odlewany — świadczą o tym obecne wszędzie zaokrąglenia i wyoblenia. Dokładne odtworzenie takiego kształtu byłoby bardzo pracochłonne. Ten szczegół jest tak mało widoczny, że zdecydowałem się odwzorować go w sposób uproszczony, bez zaokrągłeń (Rysunek 2.6.31). Przebieg jednego z przewodów i sposób zamocowania pomocniczego zaworu są oparte wyłącznie na domysłach, które mogą okazać się fałszywe. Pomiąłem za to wiele innych przewodów, kryjących się za fotelem.

Kolejnym uproszczeniem jest także pominięcie napisów eksploatacyjnych, biegnących wzdłuż burt samolotu. Wprawdzie wykonanie napisów „DOWN” lub „OFF”, widocznych na fotografii (Rysunek 2.6.30) nie stanowiłoby większego problemu, ale co jest napisane na tej czarnej tabliczce obok?

Kolejnym obszarem do wypełnienia jest podłoga kabiny (Rysunek 2.6.32). W P-40 tę funkcję pełniła górna powierzchnia płata. Na szczęście nie ma tu dużo detali do wykonania. Najważniejszymi elementami na tym obszarze są: zespół drążka sterowego i projektor celownika (Rysunek 2.6.32b):

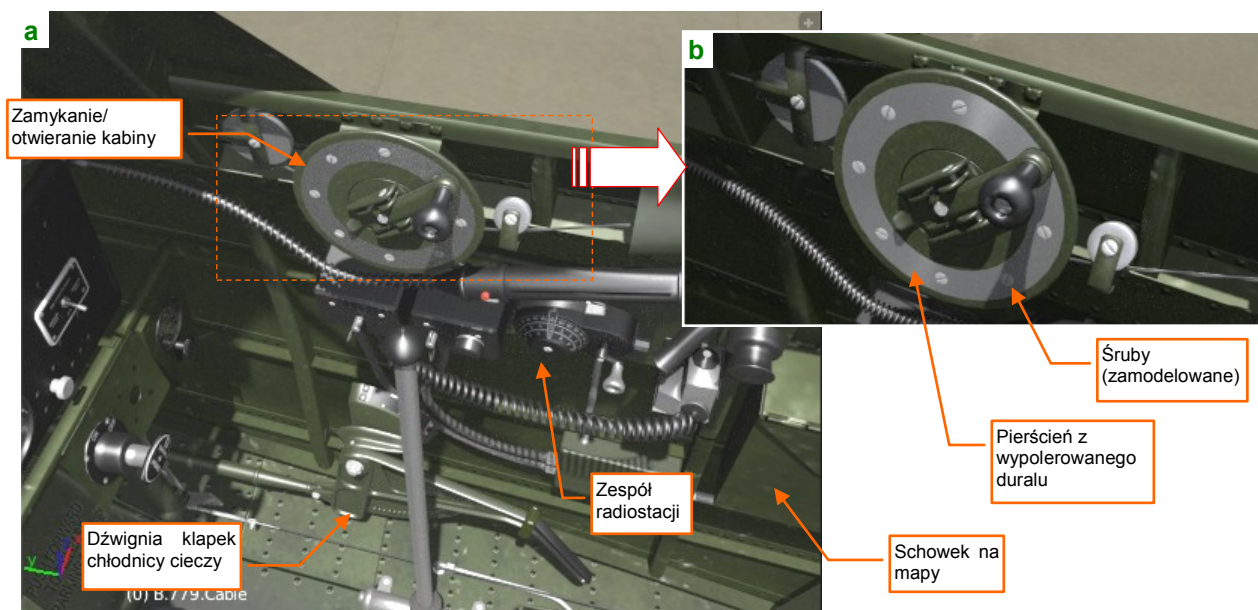


Rysunek 2.6.32 Szczegóły podłogi kabiny pilota

Drążek był najbardziej narażony na rysy i zadrapania farby, więc rozwinąłem ten fragment siatki w przestrzeni **UV** materiału **B.Skin.Cockpit**. Zwróć uwagę na przewód, wychodzący z drążka. To szczegół dość często pomijany przez modelarzy. Rysunek 2.6.32c) przedstawia szczegóły górnej części drążka sterowego. Wytlóczenie na uchwycie uzyskałem za pomocą mapy nierówności (materiału **B.Black.Instruments**). Specyficzne dla tej konstrukcji są przymocowane do na podłogi paliwomierze zbiorników skrzydłowych (Rysunek 2.6.32a).



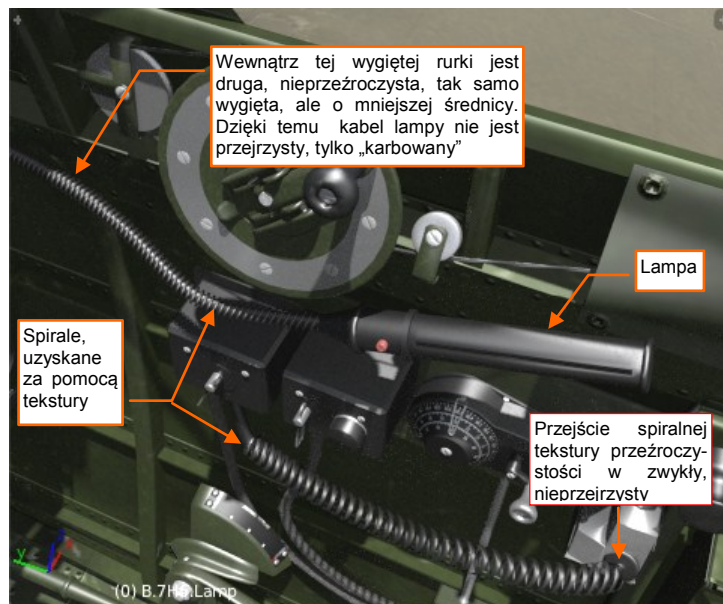
Wreszcie prawa burta kabiny (Rysunek 2.6.33). Konstruktorzy umieścili z tej strony „korbkę” służącą do odsuwania i zasuwania osłony, kontrolki radiostacji, oraz dźwignię sterowania chłodnicy (Rysunek 2.6.33a):



Rysunek 2.6.33 Szczegóły prawej burty kabiny pilota

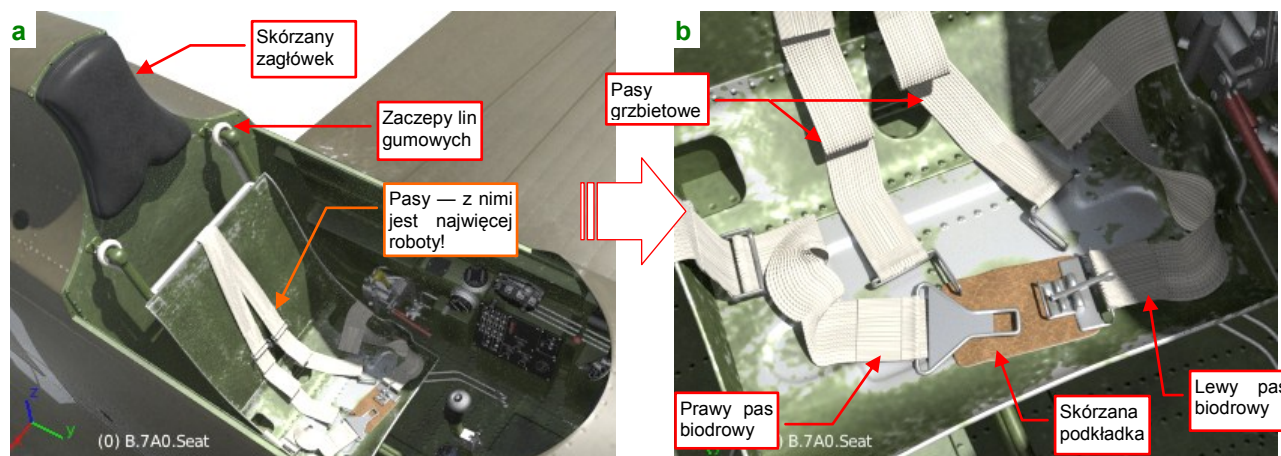
Tarcza korby osłony kabiny miała fragment powierzchni zachowany w naturalnej barwie duralu, wypolerowanego niemal jak lustro (Rysunek 2.6.33b). Powinienem właściwie go pokryć materiałem **B.Skin.Duraluminum**, takim jak poszycie samolotu (por. Wyciąg III). Przygotowanie dla tego tak małego fragmentu odpowiedniej tekstury nierówności byłoby jednak zbyt pracochłonne. Zrobiłem więc to „na skróty”: pokryłem tę powierzchnię zmodyfikowaną wersją materiału **B.Steel.Rough: B.Alumin.Smooth**. Łby ośmiu śrub, które są na niej widoczne, to siatki, a nie tekstura (Rysunek 2.6.33b). (Wykorzystałem klony standardowych śrub, często używanych w innych miejscach kabiny i podwozia). Tak po prostu było najszybciej, a wpływ na komplikację modelu tych kilkudziesięciu dodatkowych ścian jest minimalny.

Na prawej burcie występują dwa spiralne przewody (Rysunek 2.6.34). Pierwszy to elastyczny kabel lampy (spotyka się takie nadal w niektórych lampkach biurowych). Drugi — to zawijany kabel od wtyczki (mikrofonu?) radiostacji, przypominający kable słuchawek starych, przewodowych telefonów. Są to na tyle drobne elementy, że zostały odwzorowane za pomocą tekstury. Już raz taką sztuczkę wykonaliśmy, przy sprężynach kółka ogonowego (por. str. 49). Wykorzystałem do tego celu, podobnie jak wtedy, odpowiednio spłaszczoną teksturę proceduralną typu **Wave Texture:Bands**. Każdy z kabli to po prostu wygięty wzdłuż krzywej cylinder. Pewną niedogodnością jest tu konieczność indywidualnego dopasowania mapowania zwojów tekstury. Zwróć uwagę np. na zakończenie przewodu od radiostacji (Rysunek 2.6.34) — „kabel”, imitowany przez teksturę dyskretnie przekształca się w nieprzezroczystą końcówkę. Wymagało to „trafienia” końcem zwoju w początek nieprzezroczystego kabla. W rezultacie dla każdego z tych przewodów trzeba było stworzyć indywidualny materiał. Dla dwóch elementów, jak w tej kabine, to jeszcze nie był problem. Pamiętaj jednak o tym ograniczeniu, gdybyś chciał wykorzystywać to rozwiązanie częściej. Zbyt dużo materiałów może bardzo utrudnić wprowadzanie zmian w modelu.



Rysunek 2.6.34 Szczegóły spiralnych kabli

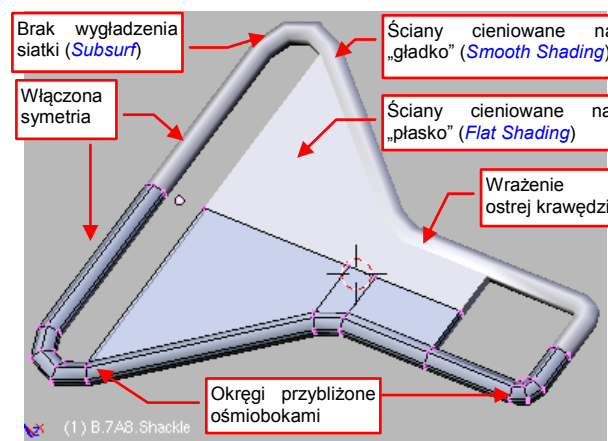
Ostatnim obszarem kabiny, który „uszczegółowiłem”, była tylna ściana, wraz z fotelem pilota (Rysunek 2.6.35):



Rysunek 2.6.35 Szczegóły tylnej ściany kabiny pilota

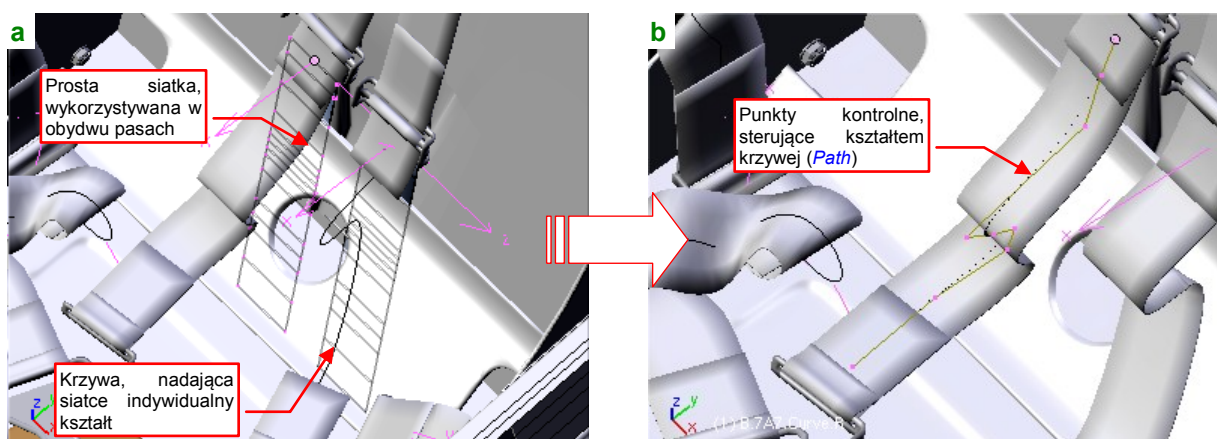
Wykonanie tego fragmentu wydaje się prostsze od pozostałych — bo zawiera mniej detali. Ot, zagłówek, pasy, i jakieś inne drobiazgi (Rysunek 2.6.35a). A jednak modelowanie „luźno rzuconej” uprząży pilota (Rysunek 2.6.35b) wymagało istotnego wysiłku. I to wcale nie z powodu zróżnicowanych materiałów!

Tworzenie tego zespołu zacząłem od wykonania kompletu sprzączek. Są to stosunkowo małe detale, więc nie trzeba było tu wygładzać siatki modyfikatorem *Subsurf*. Do uzyskania odpowiedniego efektu wystarczyło użyć uproszczeń — np. okręgu ośmiokątem (Rysunek 2.6.36). Wrażenie zaokrągleń uzyskałem wyłącznie przez oznaczanie cieniowania odpowiednich ścian jako „gładkie” (*Smooth Shading*). Ewentualne ostre krawędzie powstawały samoistnie na granicy ścian, różniących się tym ustawieniem. Zastosowałem także inne uproszczenia, jak lustrzane odbicie siatki. Rezultaty na renderze wyglądają całkiem realistycznie, nawet w zbliżeniach (Rysunek 2.6.35b).



Rysunek 2.6.36 Siatka sprzączki

Same pasy to bardzo płaskie prostopadłościany (Rysunek 2.6.37a). Ich grubość odpowiada 1 mm w rzeczywistym samolocie. Te siatki są wygięte wzdłuż odpowiedniej krzywej (modyfikatorem *Curve*, tak jak wyginałymi kable i rurki). To te krzywe nadają im indywidualny kształt (Rysunek 2.6.37b):

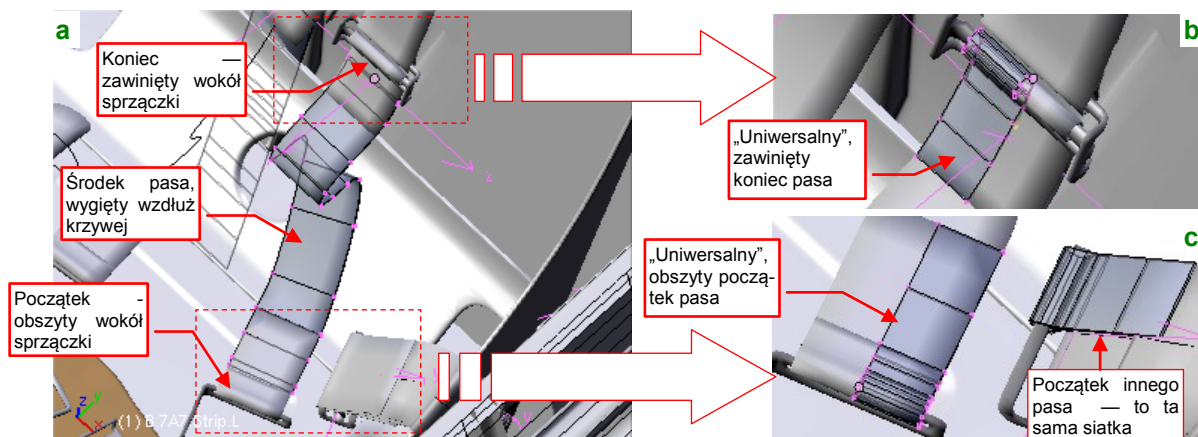


Rysunek 2.6.37 Pasy — płaskie siatki, wygięte wzdłuż krzywej

Do stworzenia lewego i prawego pasa użyłem dwóch klonów tej samej siatki (por. Rysunek 2.6.37a).



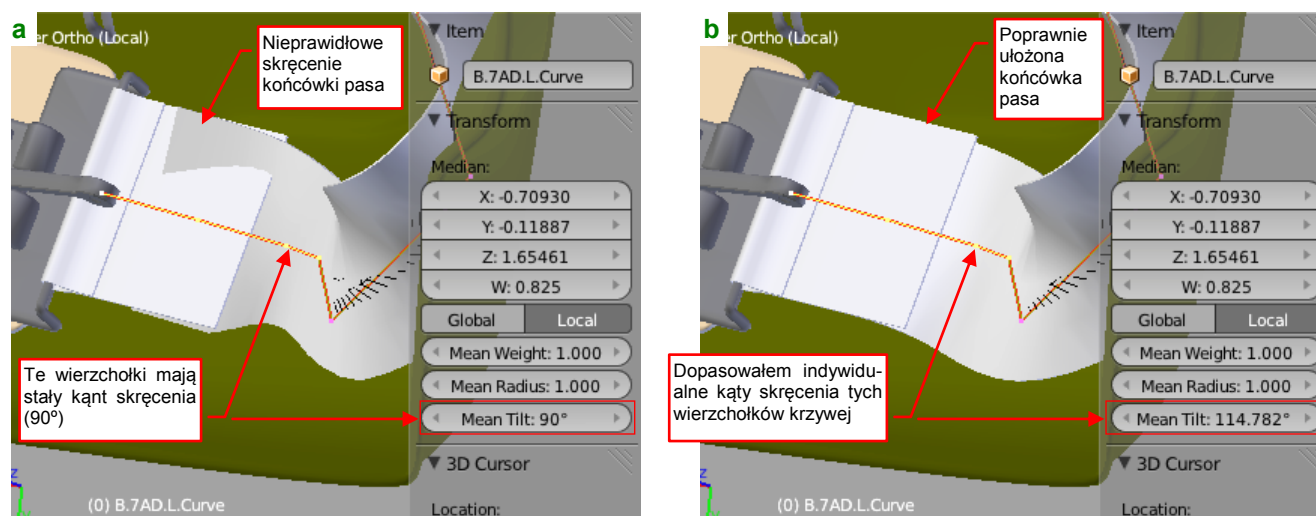
W zasadzie uprząż pilota składa się z czterech pasów: dwóch biodrowych i dwóch grzbietowych (choć te ostatnie są właściwie połączone w jeden długi odcinek — por. Rysunek 2.6.35a). W środku każdego pasa jest sprzączka, dzieląca go na dwie części. Początek każdej z tych części jest obszyty wokół jednej ze sprzączek, a koniec — zawinięty wokół poprzeczki drugiej (Rysunek 2.6.38a):



**Rysunek 2.6.38** Trzy elementy każdego pasa: wygięty środek i dwie nie zdeformowane końcówki

Zamodelowanie takich końcówek poprzez wygięcie wokół krzywej jest możliwe, ale wymagałoby powtórzenia wiele razy tej samej, żmudnej roboty. (Mamy tu siedem fragmentów upręży, z których każdy ma swój indywidualny kształt — i swoją własną krzywą). Można jednak to zdecydowanie uprościć, gdy w charakterze obszycia użyjemy klonu jednej, uniwersalnej siatki. Dosztukujemy ją, jako oddzielny obiekt, do początku każdego pasa (Rysunek 2.6.38c). Podobnie każde owinięcie tkaniny wokół sprzączki to klon innej, uniwersalnej siatki, nakładany na koniec wygiętego odcinka pasa. Wystarczy tak skomponować krzywe, modelujące fragmenty upręży, by płaskie końcówki wydawały się być ich naturalnym przedłużeniem. (Obszycia, przez to, że łączą „na zakładkę” dwie warstwy tkaniny, rzeczywiście układają się płasko. Nie można tego powiedzieć o zawiniętych końcach pasów. Te jednak wystarczy pochylić pod różnymi kątami, by także nie raziły swoją „sztywnością”). Dzięki temu, że ani obszycie, ani zawinięcie nie są wyginane wzdłuż krzywej, można je jeszcze bardziej uprościć, modelując tylko połowę ich siatek — a drugą uzupełnić modyfikatorem *Mirror* (Rysunek 2.6.38b,c).

Przy bardziej „pokręconych” fragmentach pasów możesz mieć problemy z ich dopasowaniem do „sztywnych” końcówek (Rysunek 2.6.39a):



**Rysunek 2.6.39** Korygowanie niewłaściwego skrócenia pasa wokół osi

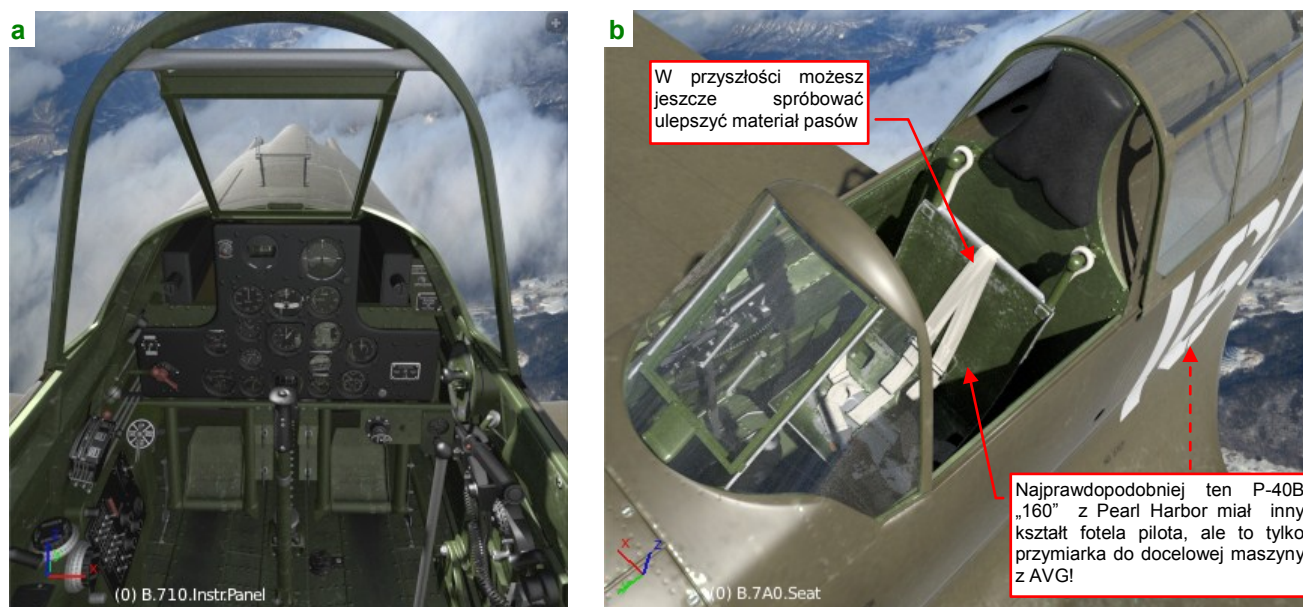
W takich sytuacjach musisz zmienić kąty poprzecznego skrócenia (*Tilt*) w poszczególnych wierzchołkach krzywej (W przyborniku *Property* — por. Rysunek 2.6.39b, więcej na ten temat znajdziesz na str. 290).

- Gdy masz zamodelować jakiś specjalnie skręcony środkowy fragment pasa, nie zapomnij przygotować dla tego obiektu indywidualnej siatki! Na pewno będziesz musiał ją lokalnie zagęścić, wstawiając więcej poprzecznych sekcji (poleceniem [Loop Cut](#)). Takie modyfikacji nie powinny być powielane w pozostałych częściach (np. w pasie lewym i pasie prawym).

Mówiąc szczerze, układanie pasów uprząży tak, by wyglądały jak swobodnie rzucone na siedzenie, uważam za jeden z trudniejszych elementów modelowania samolotu. Gdy się z tym uporamy, nadchodzi pora, by przypisać poszczególnym obiektom odpowiednie materiały. Skórzanemu zagłówkowi przypisałem materiał **B.Black.Leather**, a podkładce pod „spinką” pasów (por. Rysunek 2.6.35b, str. 97) **B.Brown.Leather** (szczegóły konfiguracji — str. 408). Pasom uprząży przypisałem materiał **B.White.Belt** (szczegóły konfiguracji — str. 415), gdyż na zdjęciach i w filmach te pasy wyglądają na białe. Liny gumowe, zaczepte za fotelem, także wydawały się białe, więc przypisałem im ten sam materiał, co pasom (to uproszczenie, ale nie razi).

Przy okazji wprowadziłem pewną modyfikację do podstawowego materiału wnętrza kabiny (**B.Skin.Cockpit**). Podstawiłem we wnętrzu przetarc metaliczną powierzchnię (w charakterze maski użyłem intensywności tekstury barw — por. str. 170). Otrzymałem w ten sposób ostre krawędzie przetarc, które lepiej wyglądają w zbliżeniach (np. str. 97, Rysunek 2.6.35). Rezultaty były na tyle ciekawe, że dodałem podobne rozwiązanie także do materiału powierzchni zewnętrznych (**B.Skin.Camouflage** — por. str. 171).

Wnętrze naszej kabiny jest ukończone. Nadal nie dorównuje najlepszym symulatorom tego samolotu. Pomiąłem wiele przewodów i napisów eksploatacyjnych, których nie byłem w stanie odcyfrować. (Zawsze można je dodać później, gdy natkniesz się na nowe, dokładniejsze materiały). Tym niemniej osiągnęliśmy całkiem przyzwoite przybliżenie oryginału. Rysunek 2.6.40 przedstawia dwa ujęcia gotowej kabiny pilota:



Rysunek 2.6.40 Gotowa kabina (na razie bez pilota)


Rysunek 2.6.40a) to ujęcie z wnętrza kadłuba<sup>1</sup>. Rysunek 2.6.40b) przedstawiaabinę widzianą z zewnątrz. Mam wrażenie że w takim zbliżeniu pasy na oparciu fotela pilota wydają się trochę sztuczne. (W przyszłości możesz spróbować ulepszyć ich materiał). Zazwyczaj będziemy jednak oglądać ten szczegół z większej odległości, więc pozostawiłem je w takim stanie.

<sup>1</sup> Aby rozjaśnić głębokie cienie wewnątrz kabiny, przed stworzeniem tego renderu umieściłem za zagłówkiem pilota pomocnicze źródło światła typu [Sun](#), skierowane na tablicę przyrządów. Ustawienia: **Size**: 1.0, **Cast Shadows**: Off, **Color**: biały, **Strength** = 3.0.



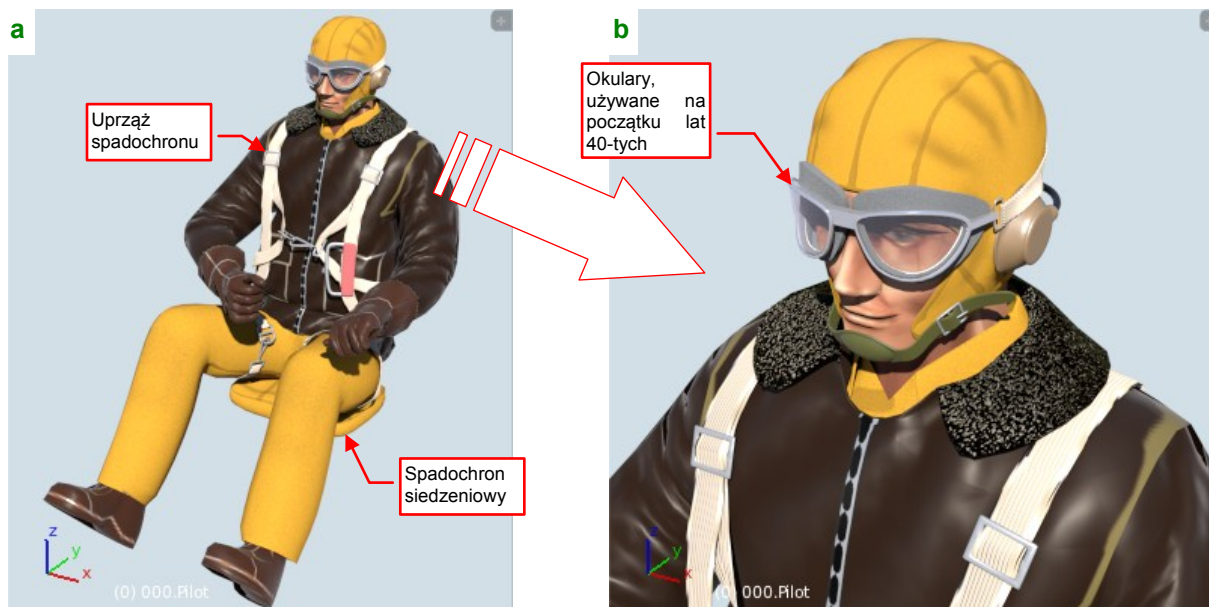
## Podsumowanie

- Wnętrze kabiny samolotu odtwarza się głównie na podstawie zdjęć (str. 83, 95). Jest to zazwyczaj bardzo skomplikowany zespół. Dlatego dobrze jest zawsze przed rozpoczęciem tej pracy założyć odpowiedni poziom uproszczeń — na przykład rezygnację z pomniejszych przewodów czy części napisów eksploatacyjnych;
- Rysunki w trzech rzutach przydają się głównie do dokładnego ustalenia położenia wręg i podłużnic wewnętrznych kabiny. Te elementy szkieletu pełnią później rolę czegoś w rodzaju „układu współrzędnych”. Pomagają lepiej oszacować rozmiary i położenie poszczególnych elementów wyposażenia (str. 83);
- Pracę zaczynamy od identyfikacji i wymodelowania podstaw, do których są przymocowane pozostałe elementy wyposażenia (str. 83). Sprowadza się to zazwyczaj do umieszczenia z przodu tablicy przyrządów, a jakiegoś zespołu „pudełek” po lewej i prawej burcie. Do „podstaw” zaliczam także fotel pilota (str. 84);
- W kabinie użyłem dwóch podstawowych materiałów (str. 83):
  - **B.Cockpit.Skin**: do „pomalowanych” powłok, czyli wewnętrznej strony poszycia samolotu, szkieletu, a także fotela pilota i innych większych elementów;
  - **B.Black.Instruments**: do wszystkich „czarnych” części: tablicy przyrządów, tabliczek z napisami eksploatacyjnymi;Dla każdego z tych materiałów trzeba przygotować rozwinięcia w **UV** odpowiednich siatek (str. 84). Posłużyłem się nim do mapowania tekstur nierówności (str. 85), odbić i barwy;
- Otwory w tablicy przyrządów najlepiej jest uzyskać za pomocą odpowiedniej tekstury przejrzystości (str. 87 - 88), oraz tekstury nierówności (str. 88);
- Każdy wskaźnik można wykonać jako zespół dwóch części:
  - szkła i oprawy, wspólnego dla wszystkich przyrządów. (Każdy ze wskaźników wykorzystuje klon tej siatki, w razie potrzeby — w zmienionej skali);
  - prostej tarczy zegara, która jest tworzona indywidualnie dla każdego wskaźnika (nanoszony jest na nią odpowiedni obraz wskazówek i skali);Tak przygotowany wskaźnik umieszczam w odpowiednim otworze tablicy przyrządów (str. 89 - 90);
- Tabliczki znamionowe i inne napisy eksploatacyjne to proste siatki, pokryte materiałem **B.Black.Instruments**. W ich rozwinięciach **UV** są umieszczone odpowiednie fragmenty tekstury barwy (str. 91);
- Szczegóły wyposażenia kabin mogą się różnić nawet w obrębie tej samej wersji samolotu. Widać to na przykładzie odmiennych reflektorów celowników w samolotach należących do USAAC i AVG (str. 92);
- Aby nie pogubić się w detalach, proponuję podzielić pracę nad kabiną na 5 stref:
  - przód, z tablicą przyrządów i wiatrochronem;
  - lewa burta;
  - podłoga;
  - prawa burta;
  - ściana tylna, wraz z fotelem pilota;Każdą z nich konsekwentnie wypełniamy wszystkimi szczegółami. Pracę nad następną strefą zaczynamy dopiero wtedy, gdy poprzednia została zupełnie skończona (str. 93);
- Drobne elementy, których siatek nie opłacało się rozwijać w przestrzeni UV pokrywam materiałem **B.Skin.Inner** (str. 91, 92). Stosowaliśmy już go na wewnętrznych powierzchniach wnętrza podwozia i klap skrzydła. Powinien mieć taką samą barwę, co wnętrze kabiny (**B.Skin.Cockpit**). Dla pokrycia detali w innych kolorach powstały z niego (zmieniłem tylko barwę podstawową) inne materiały: **B.Red.Details**, **B.Black.Details**;
- Wiele szczegółów trzeba wykonać w sposób uproszczony, o ile nie chcesz spędzić roku nad samą kabiną. Przykładem takiej części jest odlewany korpus sterowania hydraulicznych klap i podwozia (str. 95). To niewielki, mało widoczny element, wciśnięty w kąt kabiny. Próba dokładnego odwzorowania jego kształtu pochłonięłaby tyle samo pracy, co całe skrzydło!

- Elastyczne, spiralne przewody można łatwo uzyskać za pomocą odpowiedniej tekstury przejrzystości (str. 96). Mankamentem tego rozwiązania jest to, że trzeba stworzyć odrębny materiał dla każdego rodzaju przewodu. Dobrze się za to nadaje się do wykonania kilku takich elementów w kabinie;
- Pasy fotela pilota najlepiej zbudować z odcinków, połączonych sprzączkami. Każdy odcinek pasa składa się z trzech elementów:
  - obszycia początkowej sprzączki (jedna, uniwersalna siatka, której klony są użyte w każdym odcinku);
  - środka (prostokątna siatka, wygięta wzdłuż krzywej w indywidualny kształt);
  - zawinięcia wokół sprzączki końcowej (uniwersalna siatka, której klony są użyte w każdym odcinku);Taka metoda pozwala uniknąć pracochłonnego „obszywania” każdej sprzączki (str. 97, 98);
- Pasy wygięte wzdłuż bardzo skomplikowanej krzywej mogą wymagać ręcznych korekt kąta skręcenia poprzecznego (*Tilt*). Wprowadzasz je w panelu właściwości () pojedynczego punktu (lub punktów) sterującego krzywej (str. 98);

## 2.7 Figura pilota

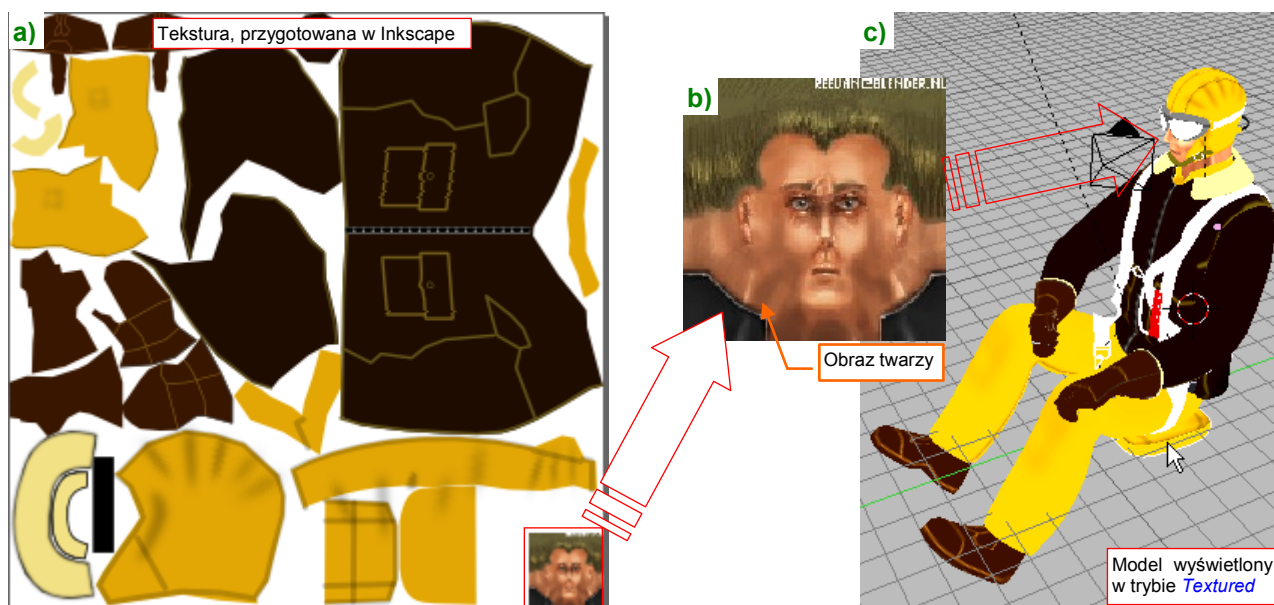
Kabina jest już gotowa, ale w scenach w locie dziwnie by wyglądała bez pilota. Tak się miło składa, że tę samą figurkę (byleby tylko była ubrana odpowiednio dla właściwych sił powietrznych) można wstawić do wielu modeli. Przygotowałem więc „uniwersalnego” pilota USAAC, w hełmofonie, goglach, kombinezonie i kurtce używanych w latach 1941-42. Założyłem mu także spadochron siedzeniowy, razem z uprzężą (Rysunek 2.7.1):



Rysunek 2.7.1 Figurka pilota USAAC/AVG (render)

Najbardziej szczegółowo przygotowałem górne partie tego modelu (Rysunek 2.7.1b) — gdyż te, po wsadzeniu pilota wabinę, będą najbardziej widoczne. Elementy mało widoczne, jak buty i spodnie, wykonałem bardzo pobieżnie (np. pominąłem na spodniach kieszenie, znajdujące się pod kolanami) (Rysunek 2.7.1a).

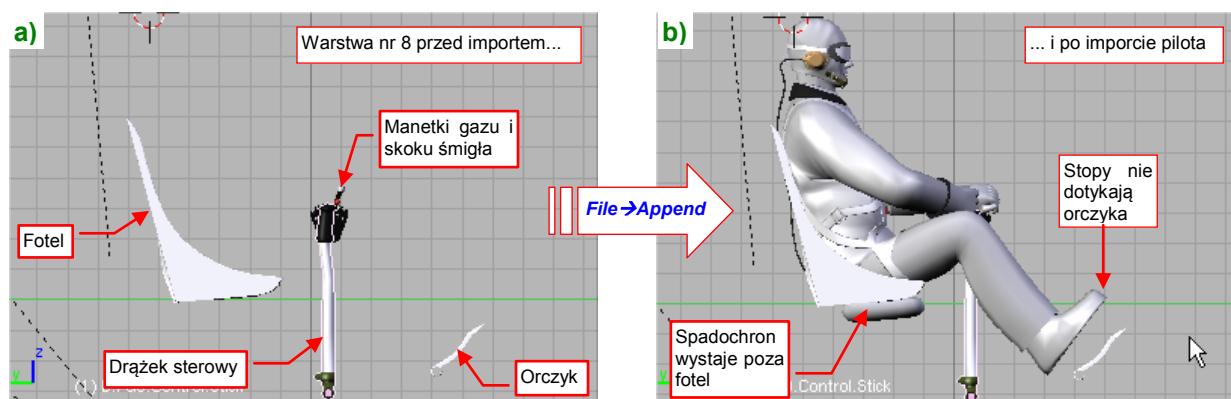
Plik z pilotem znajdziesz w folderze [model/p40/](#), pod nazwą [pilot.blend](#). (Zdjęcia pokazują, że lotnicy AVG ubierali się podobnie jak piloci USAAC). W pliku jest także „zapakowany” obraz tekstury, wykorzystywanej w modelu (Rysunek 2.7.2a). W prawym dolnym rogu umieściłem obraz twarzy (Rysunek 2.7.2b), zapożyczony z *Blender Art Magazine* (sam nie potrafiłbym tak namalować). Dla porządku zachowałem na nim podpis autora:



Rysunek 2.7.2 Tekstura i jej rozłożenie na modelu

Źródłowy plik obrazu (wykonany w Inkscape) tekstury znajdziesz w [p40/textures/8.07/pilot.src/000.svg](#).

Jeżeli sprawdzisz wcześniej sam plik *pilot.blend* w Blenderze, przekonasz się, że pilot znajduje się w nim na warstwie **8**, niewidoczne elementy pomocnicze (krzywe) na **9**, a armatura („kości” — por. str. 229) — na warstwie **10**. Na tych samych warstwach pojawiają się po wczytaniu do modelu P-40. Warto więc włączyć elementom, do których pilot musi się „dopasować” — dodatkową obecność na warstwie **8** (Rysunek 2.7.3a). Potem wystarczy wczytać z pliku *pilot.blend* figurkę (poleceniem **File→Append**, szczegóły str. 180) :

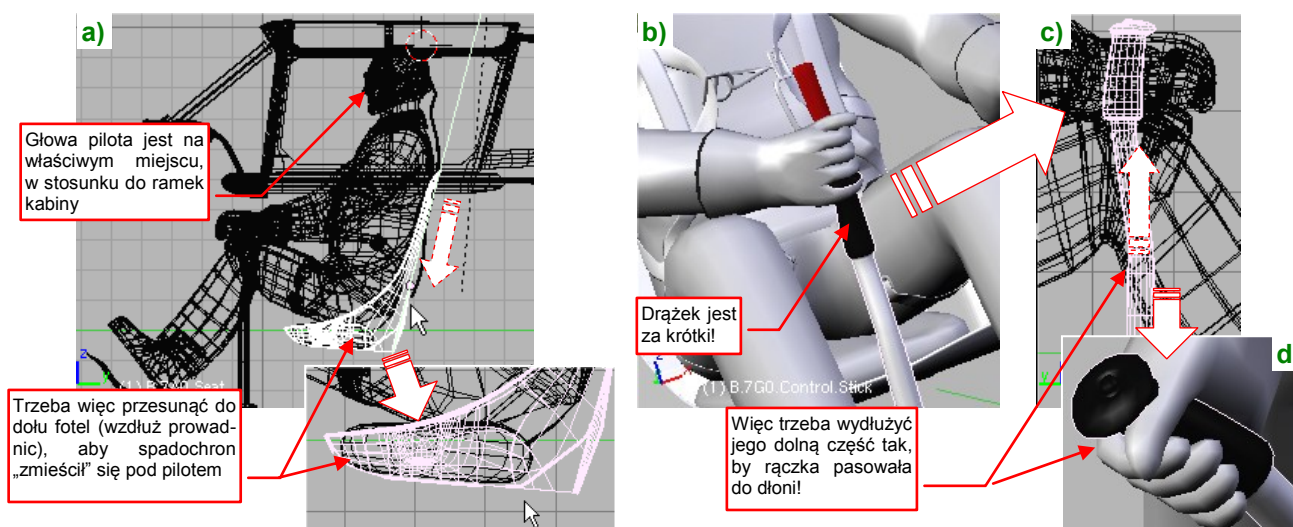


Rysunek 2.7.3 Import pilota do naszego modelu

Rysunek 2.7.3b) pokazuje sytuację zaraz po wczytaniu: zazwyczaj, trzeba dokonać kilku dopasowań. Nie inaczej jest w tym przypadku: spadochron wystaje za dno fotela, a stopy nie sięgają orczyka. Dodatkowo włączyłem widoczność ramek kabiny (warstwa **4**) (Rysunek 2.7.4a). Wygląda na to, że głowa pilota znajduje się na prawidłowym poziomie, więc trzeba obniżyć fotel, by dopasować go do spadochronu.

- Gdybyś miał chciał przesunąć pilota, „łap” go za pakunek spadochronu (**000.Parachute**). To do niego są przypisane wszystkie pozostałe elementy (także te na innych warstwach: krzywe i armatura) .

Rysunek 2.7.4a) przedstawia dopasowywanie do „poziomego siedzenia” pilota. Tak się dobrze składa, że lokalna oś **Y** fotela biegnie wzdłuż prowadnic — więc wszystko, co trzeba zrobić, to przesunąć go do dołu wzdłuż tej osi. Wystarczy obserwować (w trybie *Wireframe*), kiedy spód siedzenia pilota obejmie cały spadochron:

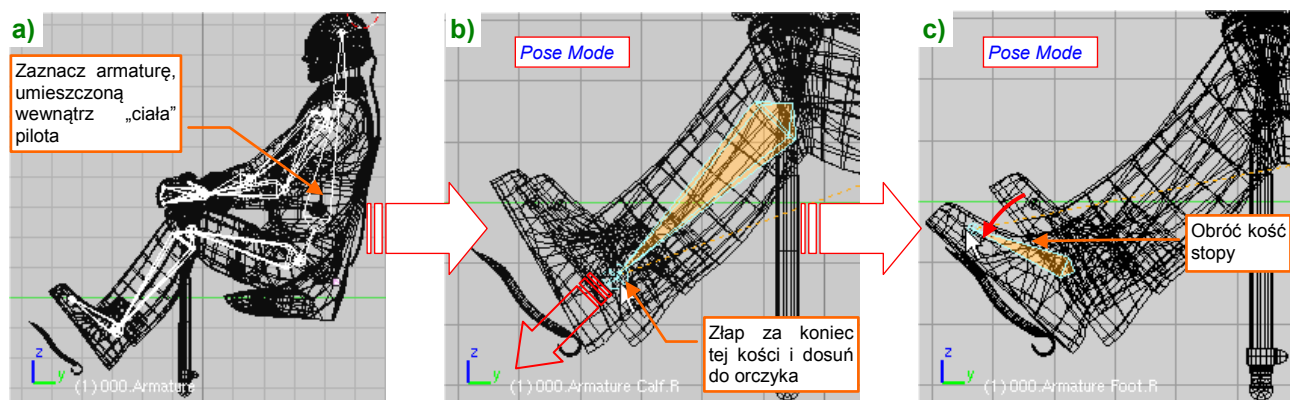


Rysunek 2.7.4 Dopasowanie elementów kabiny do pilota

Teraz przyjrzyj się uważnie rękom pilota. Lewa pewnie spoczywa na manetkach gazu i obrotów — nie wymaga żadnych poprawek. Za to z prawą jest kłopot: drążek sterowy jest za krótki o jakieś 10 cm (licząc w jednostkach rzeczywistych) (Rysunek 2.7.4b). To raczej wygląda na mój błąd podczas modelowania — nieprawidłowo oceniłem długość tego elementu na podstawie zdjęć. Na szczęście nie jest to jakiś duży problem: wydłużyłem siatkę w dolnej części drążka (tam, gdzie była prosta rura) (Rysunek 2.7.4c). Zrobiłem to tak, by zachować proporcje pomiędzy częścią prostą i wygiętą. Potem pozostało jeszcze przesunąć do góry parę elementów związanych z rączką, i całość znalazła się na właściwym miejscu (Rysunek 2.7.4d).



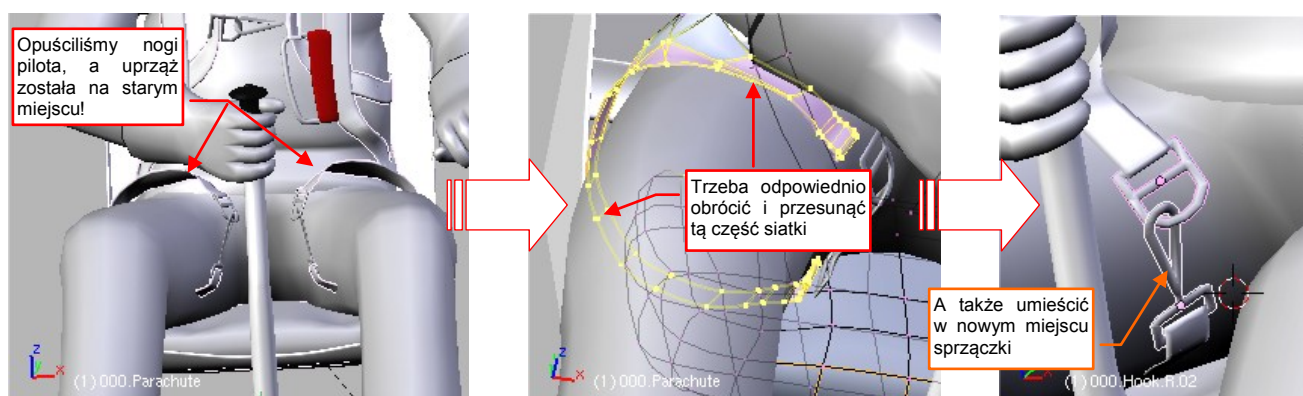
Teraz trzeba wyprostować nogi pilota, by spoczęły na pedalach orczyka. Model, który wczytałeś, ma siatkę przypisaną do armatury („kości”). Wystarczy włączyć dodatkowo widoczność warstwy 10, na której je umieściłem. W związku z tym, że armatura znajduje się wewnątrz „ciała” lotnika, najwygodniej ją zaznaczyć po przełączeniu trybu wyświetlania na *wireframe* (Rysunek 2.7.5a):



**Rysunek 2.7.5** Dosuwanie stóp pilota do pedalów orczyka

Następnie przełącz się w *Pose Mode* (por. str. 231), i zaznacz koniec kości goleni nogi (Rysunek 2.7.5b). Przesuń ją tak, by obcas oparł się o pedał. Następnie zaznacz kość stopy (Rysunek 2.7.5c), i obróć ją, by cała podszwa buta spoczęła na orczyku. Tak samo zmień położenie drugiej nogi.

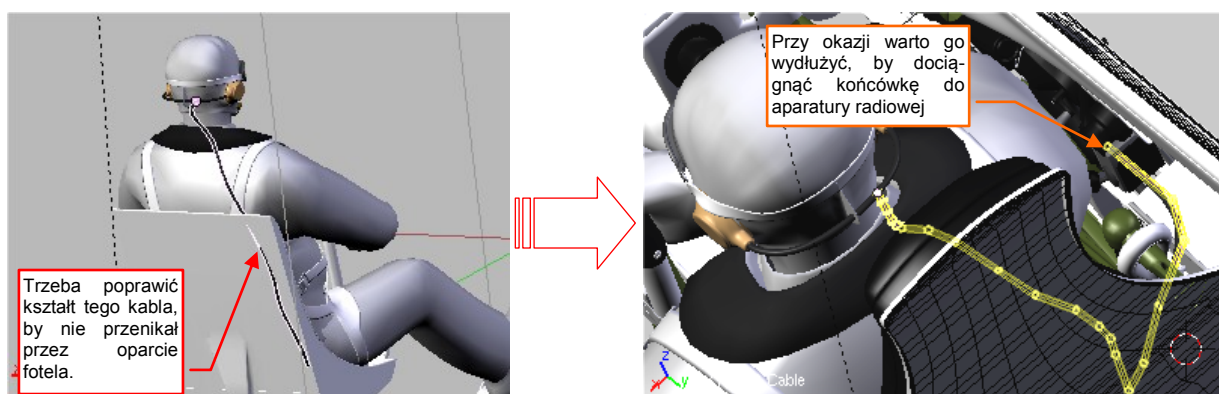
Czasami po zmianie pozycji modelu trzeba poprawić pasy uprząży (Rysunek 2.7.6). (Choć też są przypisane do armatury, to nie deformują się w ten sam sposób).



**Rysunek 2.7.6** Poprawianie uprząży spadochronu

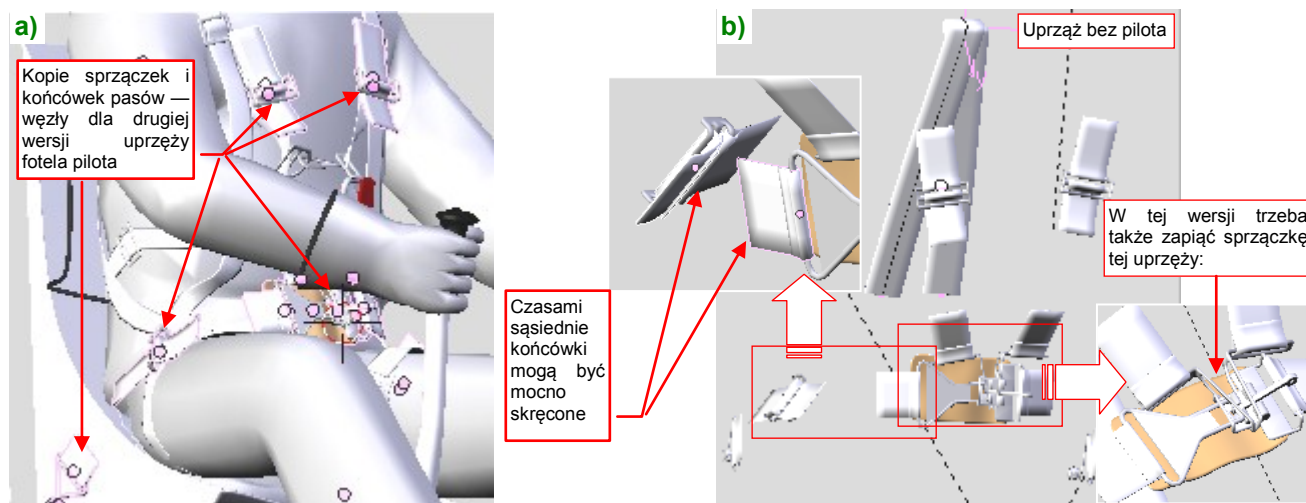
W takim przypadku trzeba przejść do edycji ich siatki i obrócić lub przesunąć odpowiednie grupy wierzchołków. Na koniec można dopasować sprzączki do zmienionego położenia końcówek pasów.

Ostatnim elementem, który na pewno będzie wymagał poprawienia, jest kabel radiowy, biegnący z hełmofonu pilota (Rysunek 2.7.7). Trzeba go inaczej wygiąć, by nie przechodził przez oparcie fotela, i wydłużyć:



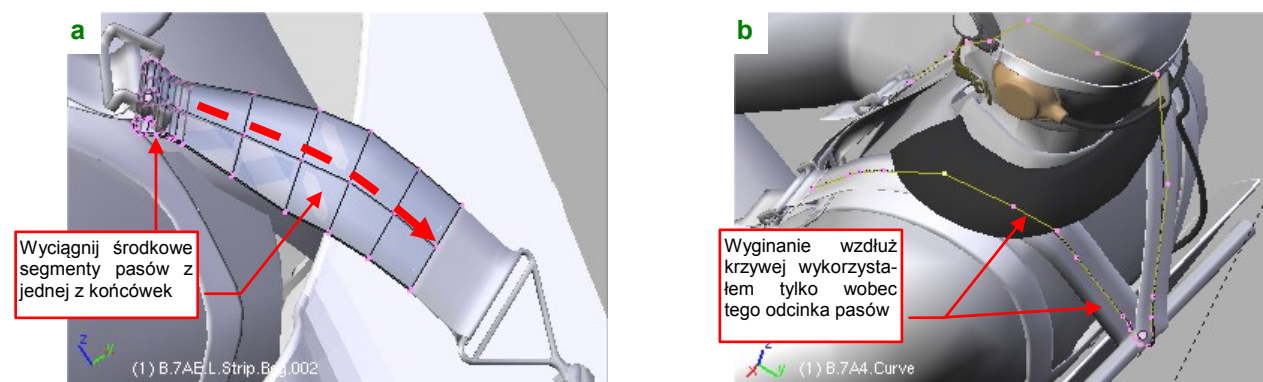
**Rysunek 2.7.7** Poprawianie przebiegu kabla hełmofonu

Pozostaje „przypiąć” pilota do fotela. Umieść w tym celu na warstwie 8 wszystkie sprzączki i końcówki pasów uprząży, którą przygotowaliśmy w poprzedniej sekcji (por. str. 97). Rozmieść je w nowych położeniach — tam, gdzie powinny się znaleźć na ciele pilota (Rysunek 2.7.8a). To drugi zestaw pasów, do scen w locie:



Rysunek 2.7.8 Ponowne rozłożenie uprząży fotela pilota

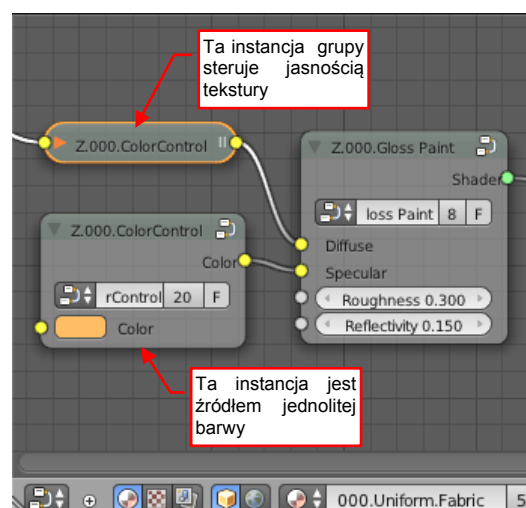
Przy okazji „zapnij” wszystkie elementy sprzączki (Rysunek 2.7.8b), i ustaw odpowiadające sobie węzły tak, by mimo skręcenia miały szansę zostać poprawnie połączone. Tym razem pasy nie są tak bardzo „pokręcone”, więc najlepiej jest je wyciągnąć z jednej z końcówek (Rysunek 2.7.9a). Krzywej użyłem tylko dla najdłuższego odcinka na plecach (Rysunek 2.7.9b):



Rysunek 2.7.9 Szczegóły wykonania pasów

Wczytany model pilota ma swoje własne materiały. Możesz je łatwo odróżnić od innych materiałów modelu po przedrostku nazwy: wszystkie zaczynają się od „000.”. Na przykład materiał kombinezonu nosi nazwę **000.Uniform.Fabric**, a skórzanej kurtki — **000.Uniform.Leather**. Wykorzystałem w nich kopie różnych grup pomocniczych stworzonych na potrzeby tego modelu: *Moderate*, *Negate*, *Gloss Paint*. Aby jednak po imporcie nie mylić ich ze sobą, te dodatkowe grupy mają przedrostek „Z.000.”. Dzięki temu pojawią się na samym końcu menu *Add → Group*.

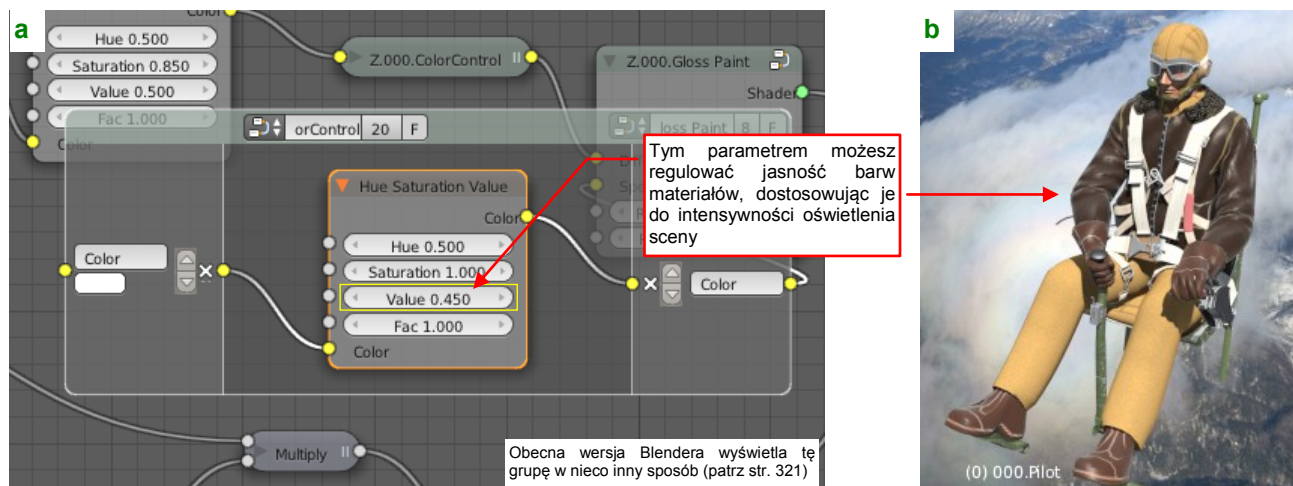
Model pilota będzie można wykorzystać w wielu scenach o różnej intensywności oświetleniu. Dlatego każda barwa w schematach jego materiałów jest wprowadzona za pomocą specjalnej grupy o nazwie **Z.000.ColorControl**. Jej instancje znajdziesz w wielu miejscach (Rysunek 2.7.10):



Rysunek 2.7.10 Użycie grupy **Z.000.ColorControl** w materiale uniformu pilota



Wystarczy że przejdziesz do edycji jakiegokolwiek z tych instancji grupy **Z.000.ColorControl** i zmienisz wartość parametru **Value** w węźle **Hue Saturation Value** (Rysunek 2.7.11a). Definicja grupy jest tylko jedna, więc spowoduje to natychmiastową rozjaśnienie lub przyciemnienie wszystkich materiałów wykorzystanych w modelu pilota (Rysunek 2.7.11b):



Rysunek 2.7.11 Dostosowanie barwy modelu pilota do oświetlenia sceny

- Aby nie pokazać na tym samym renderze dwóch wersji pasów fotela pilota, umieściłem na warstwie **18** ich opracowane w poprzedniej sekcji „luźne” wersje. Tak więc pamiętaj — gdy chcesz pokazać pustą kabinę (np. w scenie na lotnisku), włącz warstwy **15** (wnętrze kabiny) i **18**, a gdy potrzebujesz sceny w locie — warstwy **15** i **8**. Dla ujęć w których widać tylko głowę pilota wystarczy włączyć samą warstwę **8**.

Rysunek 2.7.12a) przedstawia naszego pilota w kabinie w ujęciu z przodu, a Rysunek 2.7.12b) — z tyłu:



Rysunek 2.7.12 Ukończona kabina, wraz z pilotem

Uważam, że efekt uzyskany na tych testowych renderach jest całkiem zadowalający. Być może dla takiego ujęcia warto zastosować na zewnętrznych powierzchniach samolotu teksturę barw o większej rozdzielczości. Taką decyzję trzeba było jednak podjąć wcześniej — teraz obraz poszycia jest już namalowany w GIMP, i trzeba by-łoby go tworzyć od nowa. Tak więc pamiętaj o tym na przyszłość, jeżeli planujesz zbliżenia!

**Podsumownie**

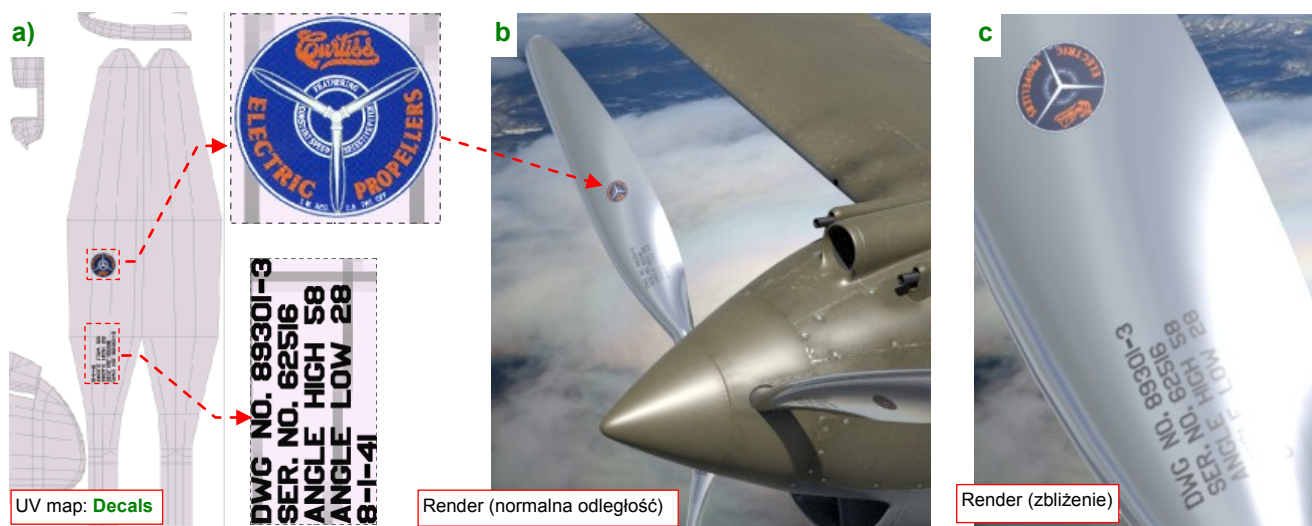
- Figurę pilota masz przygotowaną w odrębnym pliku Blendera (by można ją było wykorzystać w wielu modelach). Jest przypisana do warstwy 8. Zaimportuj ją do modelu p-40 (str. 102);
- Model pilota trzeba dopasować do kabiny. Czasami oznacza to poprawki w szczegółach jej wyposażenia (str. 103). Czasami jednak trzeba zmienić „pozę” pilota (str. 104). Do zmiany pozycji figurki służy umieszczona wewnątrz armatura („kości”);
- Po dopasowaniu pilota do kabiny, trzeba poprawić parę szczegółów, takich jak uprząż spadochronu (str. 105), czy kabel radiowy hełmofonu;
- Ostatnim elementem, jakim pozostał, jest druga wersja pasów fotela pilota (do scen „w locie”). Skopiuj w tym celu na warstwę 8 wszystkie sprzączki i końcówki pasów, stworzone w poprzedniej sekcji (str. 97). Należy je rozmieścić w nowych miejscach na figurce pilota (str. 105). Potem wystarczy połączyć je ze sobą nowymi segmentami pasa. Te fragmenty są zazwyczaj dość proste, i wtedy nie trzeba wykorzystywać do ich formowania krzywych.



## 2.8 Pozostałe szczegóły zewnętrzne

Do wykończenia pozostało jeszcze trochę drobiazgów. Aby o żadnym nie zapomnieć, proponuję zacząć je uzupełniać od nosa samolotu. Zasada jest podobna do tej, którą stosowaliśmy w kabinie (por. str. 93): sukcesywnie posuwamy się w kierunku ogona, tworząc po drodze wszystkie brakujące detale. Każdy z nich należy wykończyć nie pomijając materiałów i tekstur, by już nie trzeba było do niego więcej wracać. To także dobra okazja, by po kolei sprawdzić wygląd wszystkich zakamarków naszego modelu.

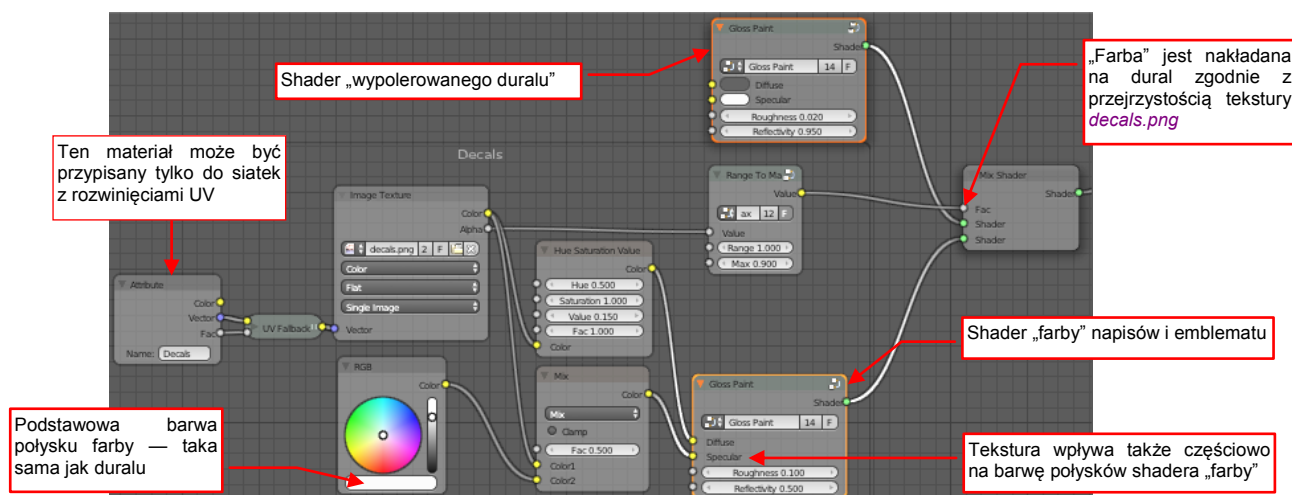
Zacznijmy od śmigła: na łopaty należy nałożyć teksturę z metryczką części i znakiem firmowym Curtiss. Jak pozostałe elementy tego typu, wkomponujemy ją w teksturę *decals.png* (Rysunek 2.8.1a,b, por. Wyciąg III):



Rysunek 2.8.1 Drobne oznaczenia na śmigle (tekstura *decals.png*)

Okrągły emblemat „Curtiss Electric Propellers” możesz zeskanować z jakiegoś schematu malowania, pomniejszyć i wstawić do rysunku *decals.svg*. Ten symbol jest tak mały, że dla domyślnego mapowania UV łopaty śmigła (**UVTex**) wygląda przekonująco tylko z większej odległości. Stworzyłem więc dla tej siatki alternatywną mapę UV **Decals**, na której łopata i emblemat są dużo większe (Rysunek 2.8.1a). Dzięki temu mapowaniu napisy i symbole wyglądają dobrze nawet na zbliżeniach (Rysunek 2.8.1c). Dla łopat w kolorze naturalnego duralu symbol Curtiss miał niebieskie tło, a dla łopat pomalowanych na czarno (np. samolotów z AVG) — czarne tło. Aby łatwo zmieniać te tła, zdecydowałem się go „zwektoryzować” (jak — patrz str. 175).

Materiał łopaty nosi nazwę **B.Duralumin**. Aby uzyskać te napisy na śmigle, dołączyłem do niego teksturę *decals.png* (Rysunek 2.8.2):

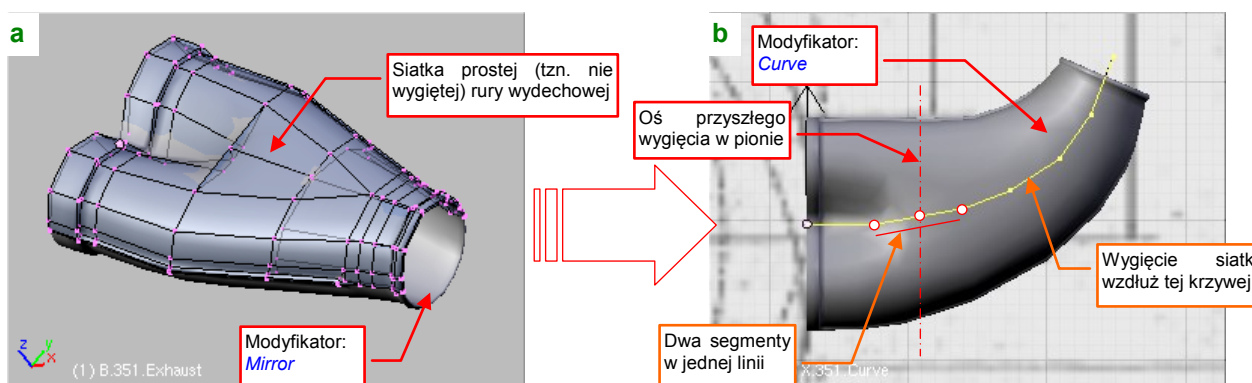


Rysunek 2.8.2 Materiał duraluminiowej powierzchni łopaty

W związku z użyciem tekstury mapowanej we współrzędnych UV, materiału **B.Duralumin** nie można wykorzystać do mniejszych szczegółów. Użyłem go jeszcze we wnętrzu wlotu powietrza do sprężarki (ta siatka także posiada rozwinięcie UV).

Teraz można przejść do kolejnego obszaru: okapotowania silnika. Do tej pory brakowało na nim kolektora spalin, teraz wreszcie nadszedł czas, by go wykonać.

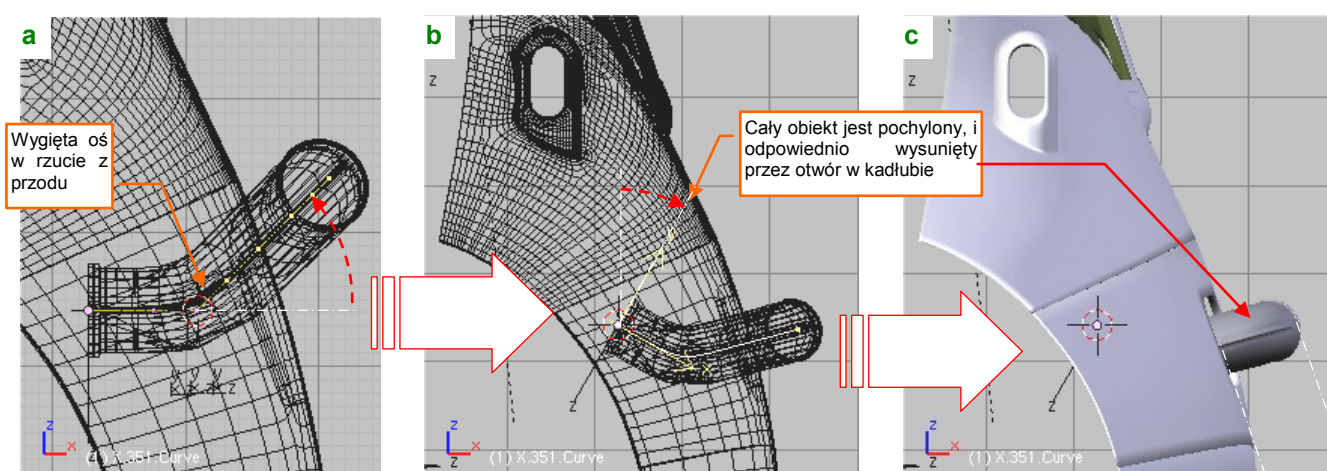
Kolektor zaczynam od uformowania pojedynczej rury wydechowej. Takie rury miały dość „pogięty” kształt. Aby nie namęczyć się nad nim bez potrzeby, przygotowałem najpierw „wyprostowaną” wersję tego elementu (Rysunek 2.8.3a):



**Rysunek 2.8.3 Formowanie pojedynczej rury wydechowej**

Na takiej siatce zdecydowanie łatwiej jest kształtować odpowiednie przejścia pomiędzy kolejnymi sekcjami. Aby sprawę jeszcze bardziej uprościć, wykonuję tylko górną połowę siatki — dolną generuje mi modyfikator **Mirror** (Rysunek 2.8.3a). Potem wystarczy dodać modyfikator **Curve**, związany z odpowiednią krzywą (**Path**), by wygiąć rurę w kształt, który widać na zdjęciach i rzutach z góry (Rysunek 2.8.3b).

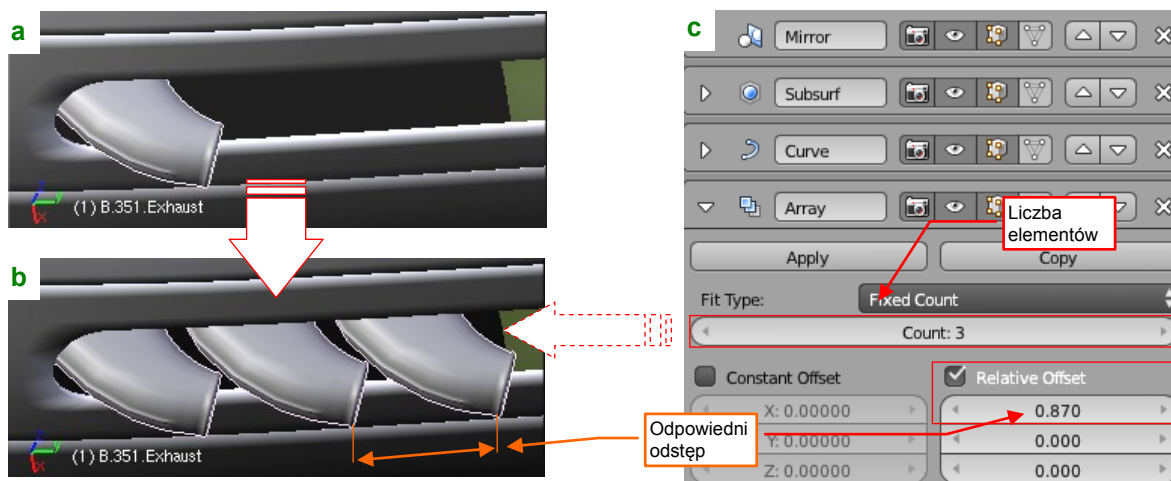
Na tym jednak formowanie się nie kończy, bo rura była wygięta także w płaszczyźnie prostopadłej (Rysunek 2.8.4a). To lokalne zagięcie, na przestrzeni dwóch segmentów wieloboku sterującego krzywą. Aby modyfikator **Curve** nie skręcił nam siatki wokół tak „pogiętej” osi, w rzucie z góry ustawiłem te dwa odcinki w jednej linii (por. Rysunek 2.8.3b).



**Rysunek 2.8.4 Kolektor spalin — ustalenie położenia w kadłubie**

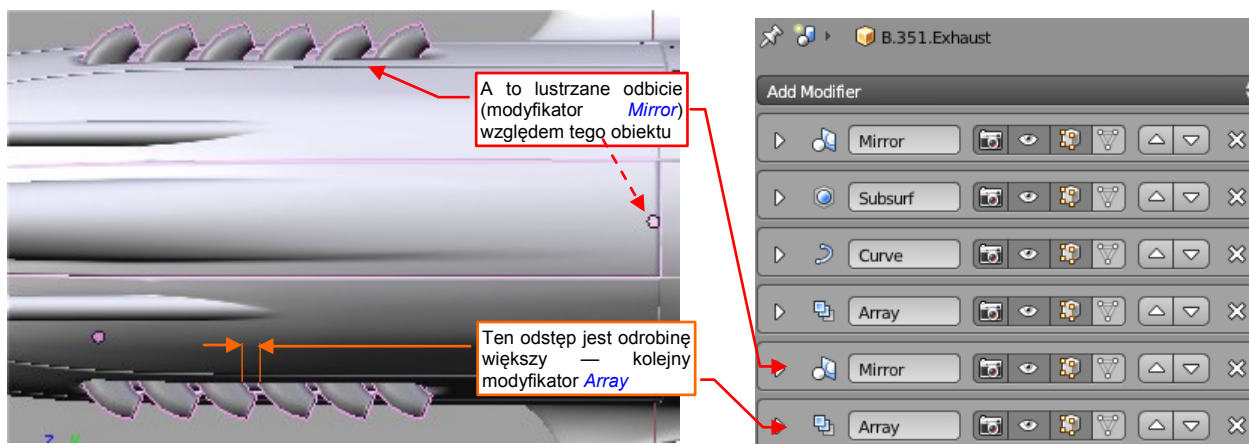
Gdy siatka jest już powyginana na wszystkie strony, pozostaje ją ustawić we właściwym miejscu. Zacząłem od odpowiedniego pochylenia rury (tym razem jako obiektu, a nie siatki) (Rysunek 2.8.4b). Obróciłem ją o 30°, bo przypuszczałem że pod takim kątem była pochylona boczna ścianka cylindrów silnika, do której był przymocowany kolektor. Potem pozostało odpowiednio go ustawić wewnątrz kadłuba, by wystawał z niego na właściwą odległość. (Jaką? Co tu dużo gadać, na planach była za duża, dobrałem ją na podstawie zdjęć).

Mamy już pierwszą rurę, teraz należy ją powielić. Użyłem tu innego modyfikatora: **Array** (Rysunek 2.8.5a,b):



Rysunek 2.8.5 Powielenie rury wydechowej (modyfikator **Array**)

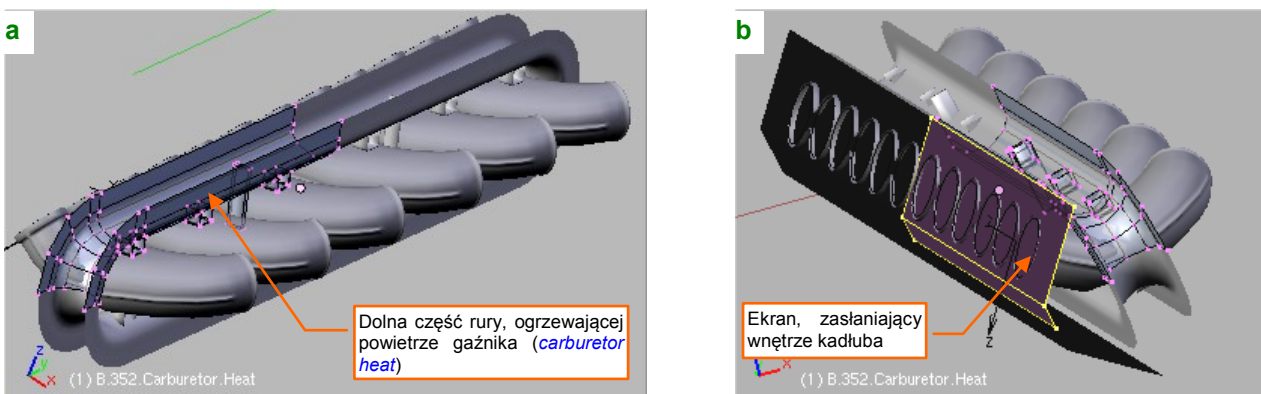
Wystarczyło go dodać i wpisać odpowiedni odstęp (**Offset** — por. Rysunek 2.8.5c), by pojawiły się kolejne rury! Drugi rząd rur (z prawej strony kadłuba) można wygenerować za pomocą kolejnego modyfikatora **Mirror** (Rysunek 2.8.6). Wystarczy podać jako punkt odniesienia środek innego obiektu (np. okapotowanie silnika):



Rysunek 2.8.6 Formowanie całości kolektora (wyłącznie za pomocą modyfikatorów!)

Dlaczego w poprzednim kroku wygenerowałem tylko trzy, a nie sześć rur? Bo pozostało coś dla koneserów: odstęp pomiędzy trzecim i czwartym cylindrem silnika Allison był nieco większy! Załatwi to kolejny modyfikator **Array** (Rysunek 2.8.6, a szczegóły o modyfikatorze **Array** — str. 297 i następna).

Pozostało już tylko „owinać” wokół kolektora połowę rury ogrzewającej powietrze gaźnika (Rysunek 2.8.7a):

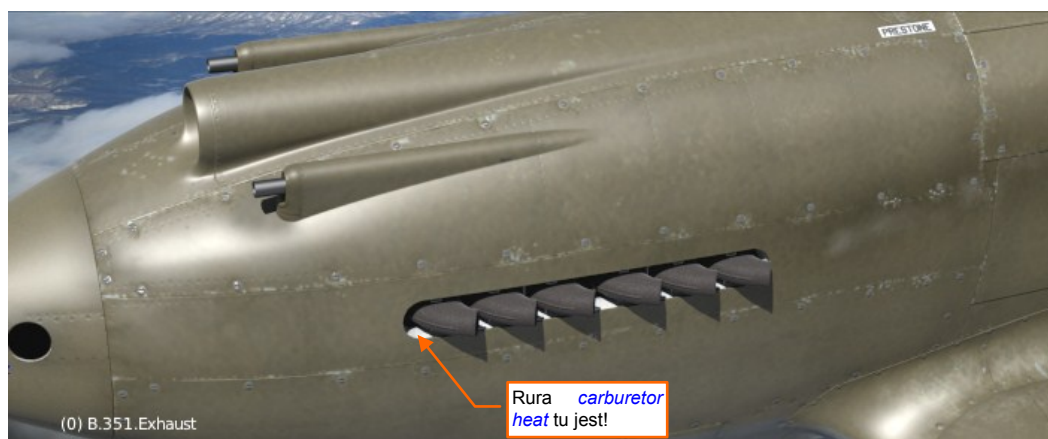


Rysunek 2.8.7 „Obudowa” kolektora spalin — ogrzewanie powietrza dla gaźnika

Na koniec dodałem z tyłu czarny „ekran”, by nie straszyć prześwitami w modelu (Rysunek 2.8.7b).



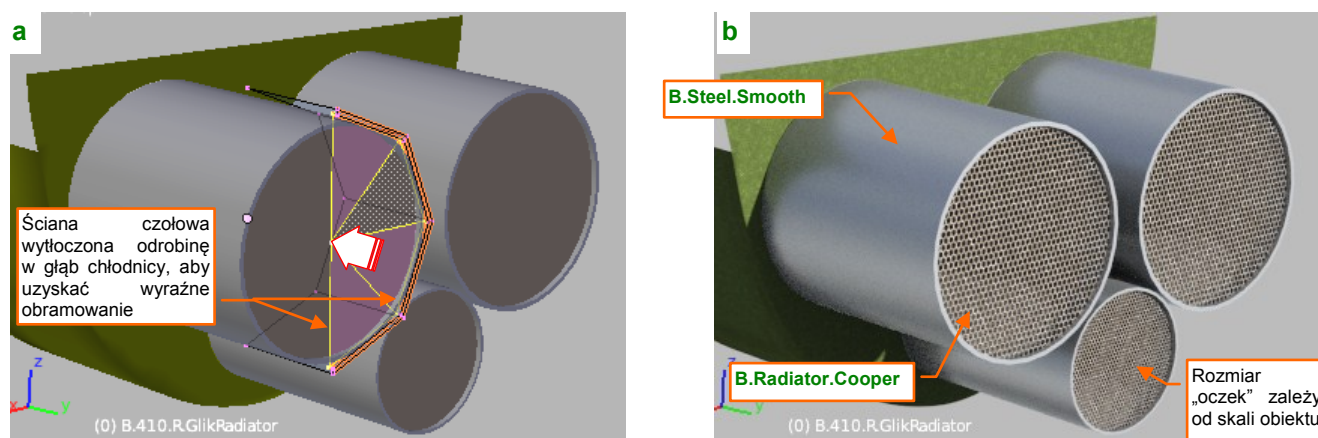
Kolektor pokryłem nowym materiałem **B.Steel.Exhaust**. To przerobiony **B.Steel.Rough**, w którym nałożyłem, za pomocą kolejnej tekstury proceduralnej „brudno-rdzawe” naleciałości (Rysunek 2.8.8):



Rysunek 2.8.8 Gotowy kolektor spalin — testowy render

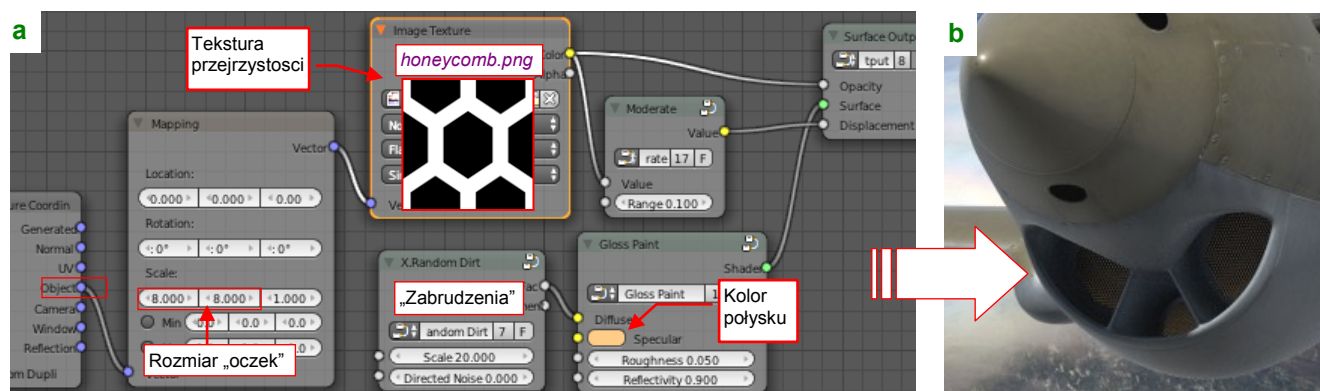
Rura z powietrzem dla gaźnika biegnie wewnątrz kadłuba, tuż przy krawędzi otworu. Nie jest na tej ilustracji specjalnie widoczna. Musisz mi uwierzyć na słowo, że pokryłem ją zwykłą stalą — materiałem **B.Steel.Rough**.

Kolejnym elementem, który wykończymy, jest zespół chłodnic. To ta sama siatka, powielona trzy razy. Ściankę czołową wgłębiłem nieco do wewnątrz, uzyskując wyraźne obramowanie (Rysunek 2.8.9a):



Rysunek 2.8.9 Siatka i materiał chłodnic

Przypisałem temu obszarowi perforowany materiał **B.Radiator.Cooper** (Rysunek 2.8.9b). Jest to „zażółcona” odmiana **B.Steel.Smooth** (zmieniłem barwę połysku — *Specular*). Perforację uzyskałem za pomocą odpowiedniej tekstury przejrzystości (wystarczy mały fragment jej wzoru — por. Rysunek 2.8.10a):



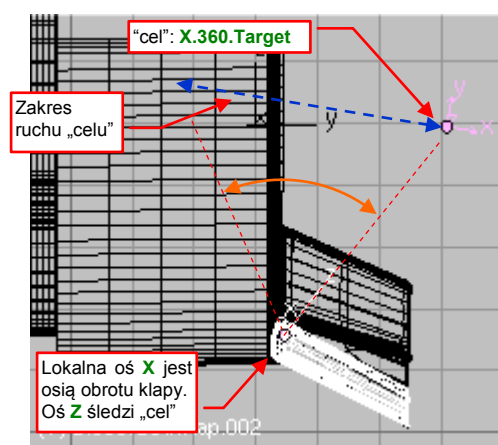
Rysunek 2.8.10 Schemat materiału **B.Radiator.Cooper**

Jako barwę podstawową (*Diffuse*) wykorzystałem teksturę ogólnych zabrudzeń **X.Random.Dirt**.

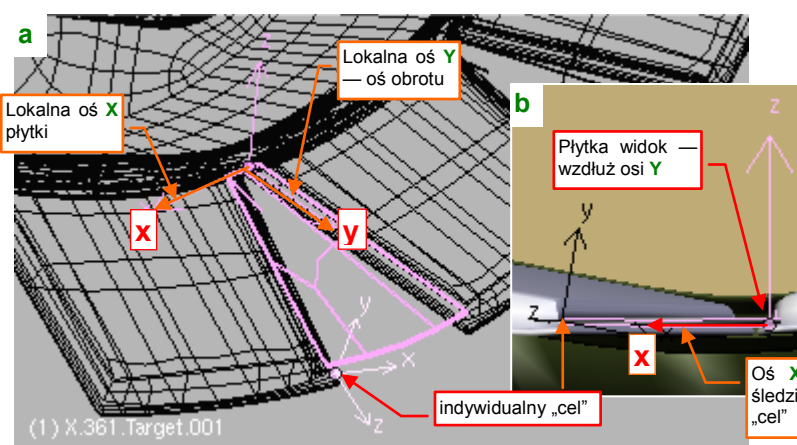


Materiał wlotu i wylotu chłodnic otrzymał „miedziany” połysk (por. Rysunek 2.8.9b), gdyż chłodnice Prestone, stosowane w P-40 aż do wersji „M”, były miedziane lub mosiężne<sup>1</sup>. Miały ścianki ułożone w sześciokątny wzór „plastra miodu”, który odtworzyliśmy za pomocą prostego obrazka [honeycomb.png](#)<sup>2</sup>. Mapowałem ten obraz we współrzędnych **Object**. Co prawda spowodowało to mniejszy rozmiar „oczek” na chłodnicy oleju (to ta w środku, o nazwie **B.415.OilRadiator**), bo ten element ma inną skalę niż chłodnice cieczy. Pozostałem jednak to w tym stanie, bo w normalnych warunkach oświetlenia ta różnica nie jest widoczna (Rysunek 2.8.10b).

Skoro chłodnice są już gotowe, czas popracować nad ich tunelem wylotowym. Na jego obwodzie umieszczono cztery kłapy. „Zmechanizowałem” je za pomocą ograniczenia **Locked Track**, śledzącego pojedynczy obiekt - cel **X.360.Target** (Rysunek 2.8.11):



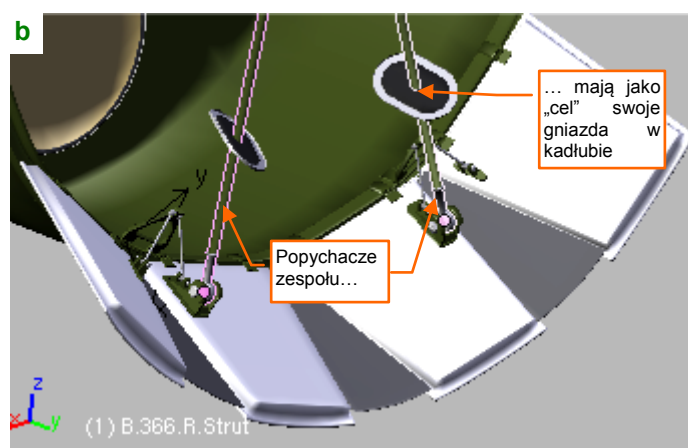
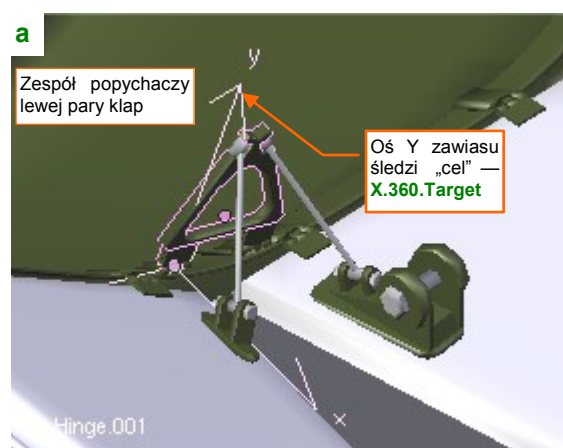
Rysunek 2.8.11 „Mechanizacja” kłap chłodnic



Rysunek 2.8.12 Dopasowanie „uszczelnień” pomiędzy kłapami

Aby pojawiające się w szczelinach trójkątne „zasłonki” nie rozjeżdżały się podczas ruchu kłap, musiałem także dodać im ich własne ograniczenia **Locked Track** (Rysunek 2.8.12a). Ich osie **X** śledzą indywidualne „cele”, umieszczone na krawędzi sąsiedniej kłapy (Rysunek 2.8.12b).

Podobnie jak w przypadku podwozia, ruch obiektu-celu (**X.360.Target**) wymusza, za pomocą kolejnych ograniczeń **Locked Track**, ruch popychaczy. Zamodelowałem dwie pary takich elementów, by udawały, że to one poruszają kłapami (Rysunek 2.8.13):

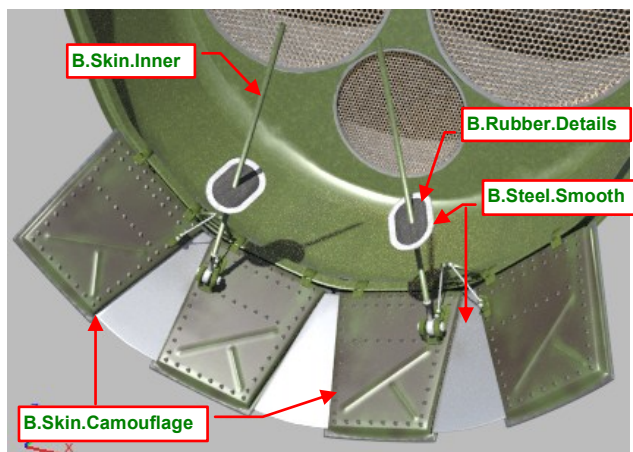


Rysunek 2.8.13 Odwzorowanie mechanizmu sterującego kłapami

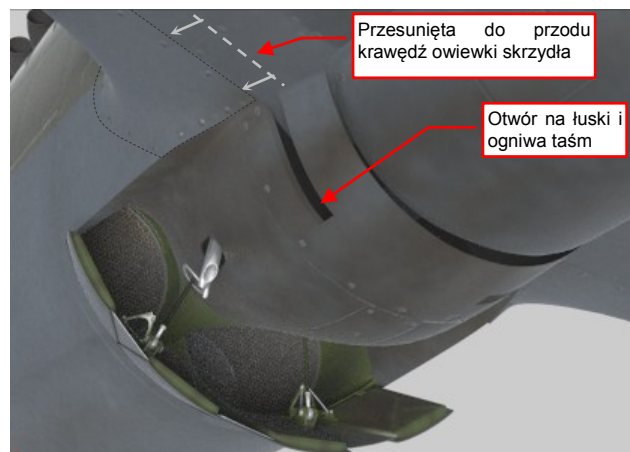
<sup>1</sup> W późniejszych konstrukcjach, jak choćby P-51 „Mustang”, czy nawet P-40N, napotkasz lżejsze chłodnice aluminiowe. Te chłodnice miały wzór ścianek zbliżony do tego, który do dziś jest używany w silnikach samochodowych, chłodzonych cieczą. Tak więc, gdy będziesz robił ich modele, w poszukiwaniu referencji wystarczy zajrzeć pod maskę własnego auta!

<sup>2</sup> Plik jest tak prosty i mały — 128x128 pikseli — że „spakowałem” go do pliku Blendera, a nie zostawiłem na dysku, jak inne obrazy. Wątpliwe, by trzeba było go w przyszłości zmieniać. W razie czego — źródłowy, wektorowy plik [honeycomb.svg](#) znajdziesz w materiałach źródłowych, folderze [model/p40/textures/8.08/src](#).

Jeżeli chodzi o dobór materiałów, to ostatecznie wnętrze kanału wylotowego pokryłem standardowym **B.Skin.Inner** (Rysunek 2.8.14)<sup>1</sup>. Można tak zrobić, gdyż te powierzchnie są mało widoczne. Tym samym materiałem są pokryte części mechanizmu poruszającego całym zespołem. Otwory w kadłubie były zakryte gumowymi uszczelkami (**B.Rubber.Details**) z metalowym obramowaniem (**B.Steel.Rough**) — przynajmniej na rekonstrukcji z Duxford! W kolorze stali pozostawiłem także trójkąty zamykające szczeliny pomiędzy klapami. Za to same klapki, jako elementy najlepiej widoczne, pokryte zostały materiałem **B.Skin.Camouflage**, z odpowiednimi teksturami nierówności, odblasku i barwy. Pobrudziłem je na brązowawo, zakładając jakieś drobne wycieki instalacji chłodnicy oleju. Ostateczny rezultat przedstawia Rysunek 2.8.15:



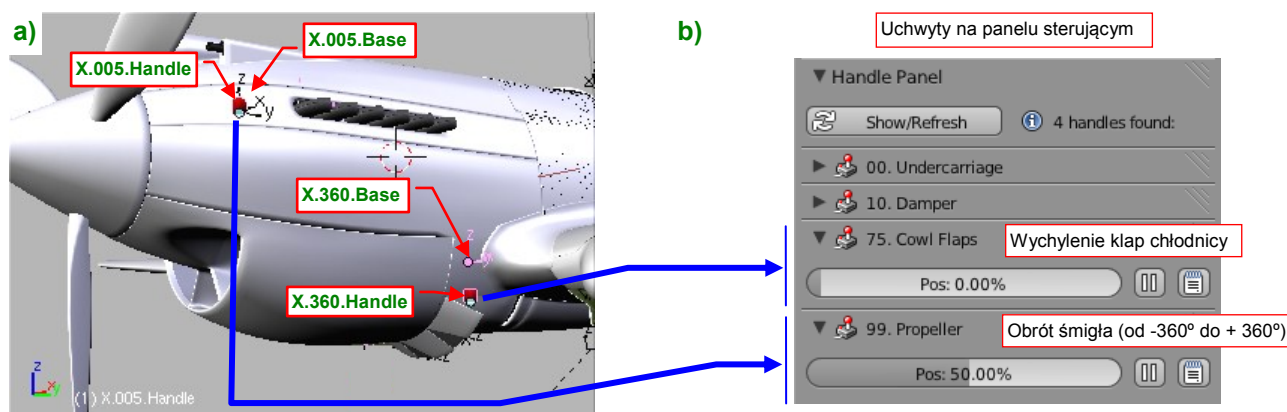
Rysunek 2.8.14 Wylot z chłodnicy — dobór materiałów



Rysunek 2.8.15 Ukończony wylot z chłodnicy

Zwróć uwagę na otwory na łuski, umieszczone za chłodnicą (Rysunek 2.8.15). Dodałem ten szczegół po zebraniu i przeanalizowaniu większej liczby zdjęć. Wymagało to kilku modyfikacji, m.in. przesunięcia do przodu dolnych krawędzi owiewek skrzydeł, razem z ich śrubami na teksturze nierówności!

Do paneli sterowania modelem (*Handle Panel* — por. str. 265) dodałem dwa zestawy kontrolki, związane z nosem samolotu: jedną dla otwierania i zamykania klapy, a drugą — dla obrotu śmigła (Rysunek 2.8.16b) :

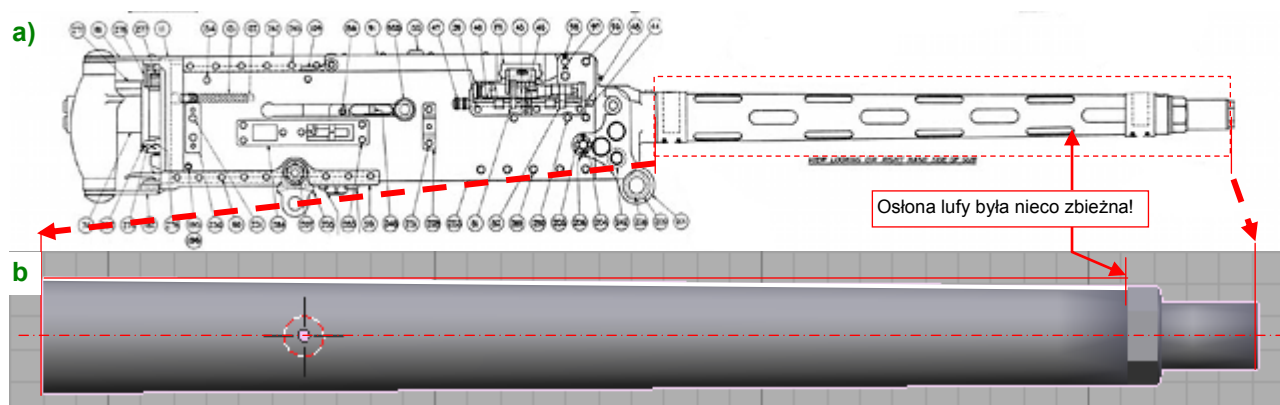


Rysunek 2.8.16 Nowe kontrolki, sterujące obrotem śmigła i klapami chłodnicy

Kontrolki tworzę tak samo, jak to robiliśmy dla podwozia (por. str. 52 i dalsze). Najpierw wstawiam na warstwę 9 „punkt odniesienia”, przymocowany na trwałe (relacją *Parent*) do kadłuba. Dla obrotu śmigła to pusty obiekt (*Empty*) **X.005.Base**, a dla klapy — **X.360.Base** (Rysunek 2.8.16a). Na warstwę 10 dodałem uchwyty **X.005.Handle** i **X.350.Handle**. Są powiązane z *\*.Base* relacjami *Parent* i ograniczeniami *Limit Location*. Z tymi uchwytami, za pomocą ograniczeń *Transfrom*, jest skojarzony ruch elementów modelu. Dla śmigła to po prostu obrót kołpaka (**B.005.Hub**), a dla klapy — przesunięcie wzdłuż lokalnej osi **X** obiektu **B.360.Target** (por. str. 112).

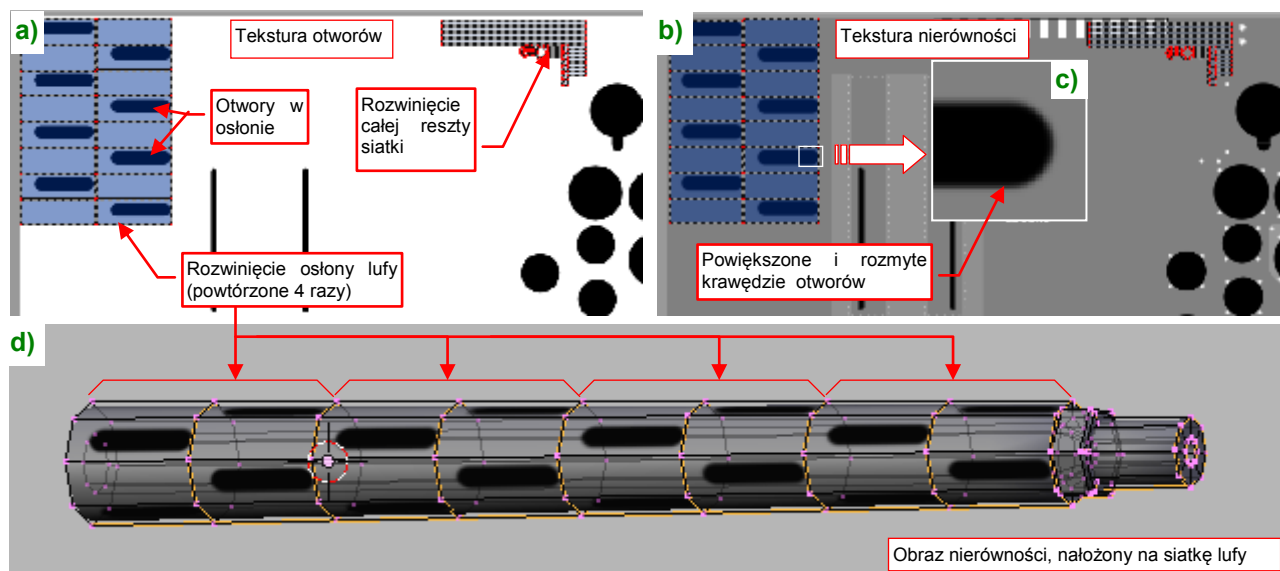
<sup>1</sup> Wcześniej był przewidziany dla **B.Skin.Camouflage**. Dzięki temu zwolniło mi się miejsce na mapie **UV**. Wykorzystałem je na wstawienie większego rozwinięcia tarczy koła. (W oryginalnej, „małej” wersji, śruby na jej powierzchni wyglądały brzydko)

Przygotowując się do odwzorowania luf karabinów, zamontowanych w skrzydłach, ściągnąłem ze strony poświęconej karabinom Browning rysunek wersji lotniczej, kalibru 0.303 cala (Rysunek 2.8.17a) :



Rysunek 2.8.17 Odwzorowanie kształtu lufy karabinu Browning 0.303 cala

Z zaskoczeniem stwierdziłem, że zewnętrzna osłona lufy była lekko stożkowa! Oczywiście zaraz to odwzorowałem (Rysunek 2.8.17b). Lufy wykorzystują ten sam materiał co zamki w kabinie pilota: **B.Black.Instruments**, bo jest „odpowiednio czarny”.



Rysunek 2.8.18 Przygotowanie otworów w osłonie lufy

Pokazywane przez Rysunek 2.8.18 otwory w osłonie wykonałem za pomocą dwóch tekstur: otworów (*instruments-holes.png*, Rysunek 2.8.18a) i nierówności (*instruments-nor.png*, Rysunek 2.8.18b)

Rysunek 2.8.19 pokazuje uzyskany rezultat — myślę, że dośkonuje otworom modelowanym w siatce.

- Przy modelowaniu konkretnej maszyny zwróć uwagę, czy końce luf karabinów nie były zaklejone jakąś jaskrawą taśmą klejącą. Tak często zabezpieczano je przed kurzem, wzbijanym przez samolot podczas startu i lądowania. To już indywidualny szczegół samolotu, który w razie czego musisz uzupełnić.

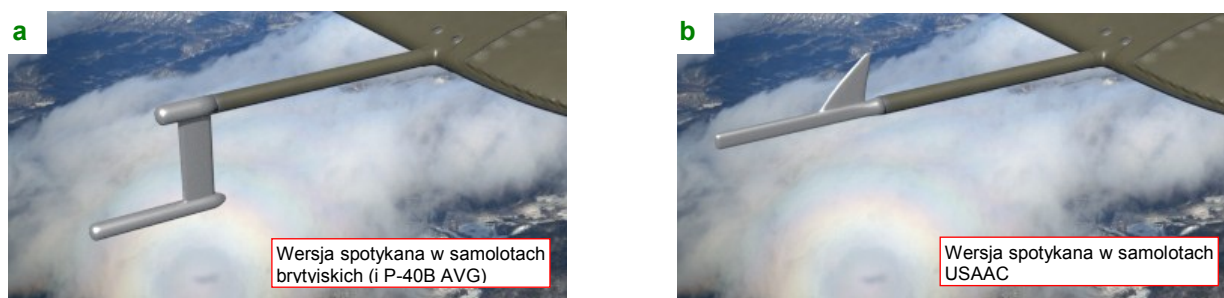
Właśnie z tego powodu nie cyelowiałem specjalnie końcówek luf karabinów — i tak były zazwyczaj zasłonięte...



Rysunek 2.8.19 Rezultat zastosowanych tekstur.



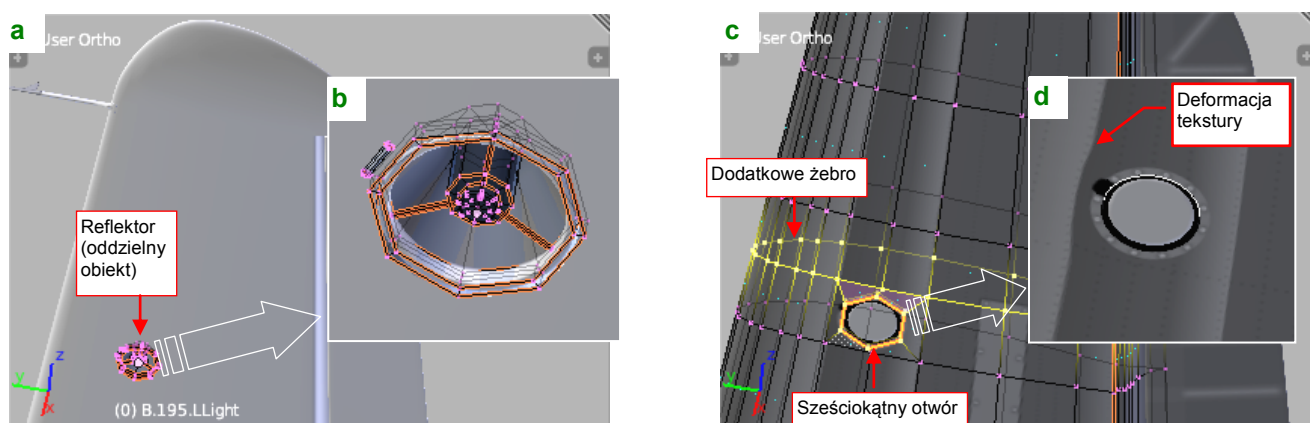
Kolejną „urką” sterzącą z krawędzi natarcia samolotu jest czujnik prędkościomierza, nazywany od nazwiska wynalazcy „urką Pitota”. Egzemplarze P-40 produkowane dla Wielkiej Brytanii miały inną końcówkę tego przyrządu (Rysunek 2.8.20a), niż samoloty wytwarzane na zamówienie USAAC (Rysunek 2.8.20b):



Rysunek 2.8.20 Dwie wersje rurki Pitota

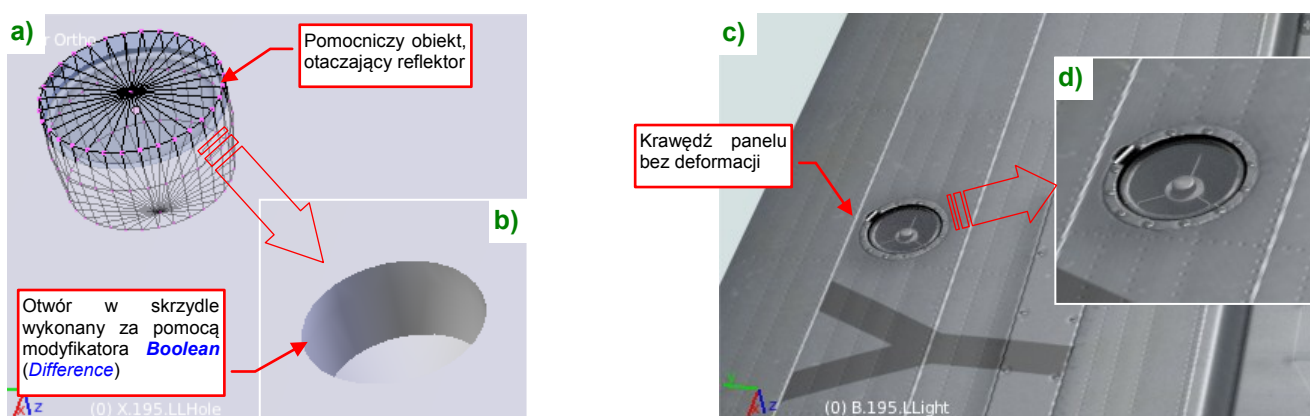
Podobnie jak w przypadku reflektora celownika (por. str. 92), przygotowałem obydwie wersje, i umieściłem je na razie na różnych warstwach (7 i 17). W następnym rozdziale podzielimy ten plik Blendera na różne sceny: jedną z Pearl Harbor, inną z AVG. Umieścimy wtedy odpowiednią wersję w odpowiedniej scenie.

Innym szczegółem jest wysuwany reflektor do lądowania w nocy. Wbudowano go w lewe skrzydło. Uformowanie samego reflektora nie było jakąś specjalną filozofią: ot, stożek z jakimś cylindrem (pełniącym rolę „żarówki”) i szybką (Rysunek 2.8.21a,b):



Rysunek 2.8.21 Składany reflektor do lądowania (w lewym płacie)

Zdecydowałem się wykonać w siatce tego płata rzeczywisty otwór, w oparciu o sześciokąt (Rysunek 2.8.21c). Niestety, okazało się że w pobliżu tego otworu brzydko wygina się tekstura nierówności (Rysunek 2.8.21d). Próbowałem różnych metod, ale w końcu stwierdziłem że pozostaje ten otwór wycięty modyfikatorem *Boolean* (Rysunek 2.8.22a,b). Wstaw ten modyfikator do obiektu lewego skrzydła po *Subdivision Surface*, a uzyskasz poprawny efekt (Rysunek 2.8.22c,d):

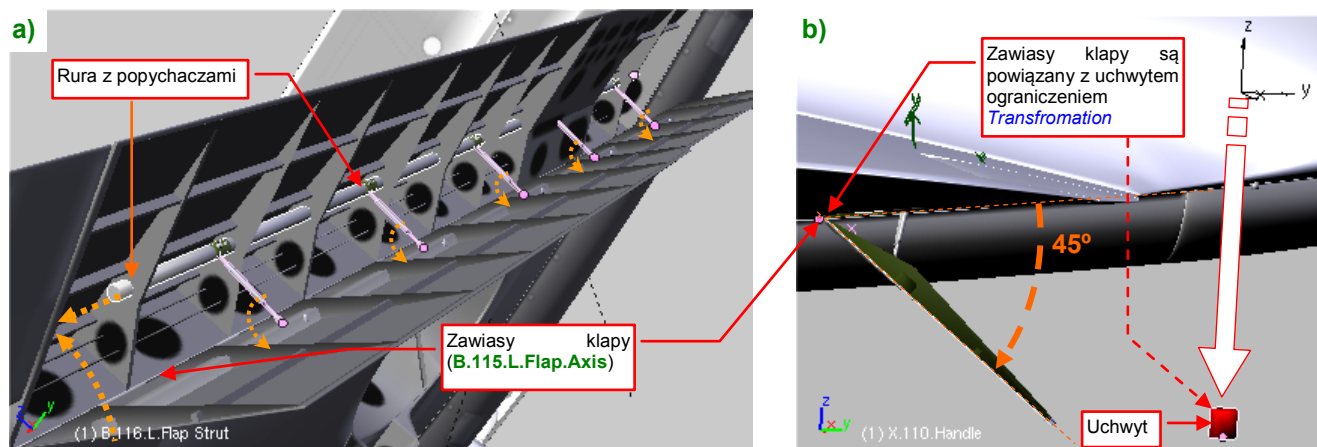


Rysunek 2.8.22 Inna metoda wykonania otworu na reflektor



Kolejnym elementem płata jest mechanizm klap. W P-40 zastosowano klapy „krokodylowe”. Po otwarciu takich klap wewnątrz skrzydła staje się widoczne, a w nim — mechanizm, który nimi porusza, oraz jakieś elementy szkieletu płata. Na szczęście już na etapie modelowania odwaliliśmy połowę roboty, umieszczając gdzie trzeba wszystkie żebra i podłużnice (por. Wyciąg II). Teraz pozostał do dorobienia sam mechanizm.

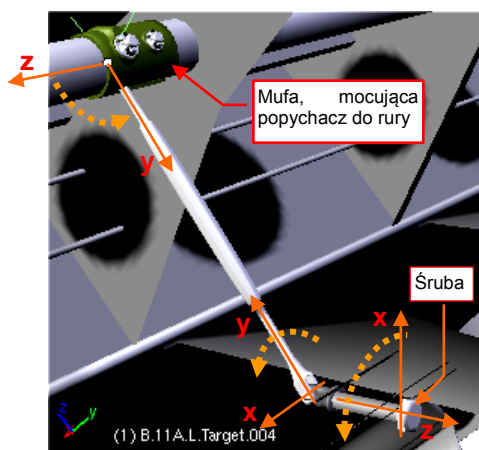
Mechanizm klap krokodylowych działał w każdej konstrukcji w ten sam sposób: wzdłuż skrzydła biegła rura, w której były przypięte popychacze (Rysunek 2.8.23a). Drugi koniec każdego popychacza były przymocowany przegubowo do szkieletu klapy:



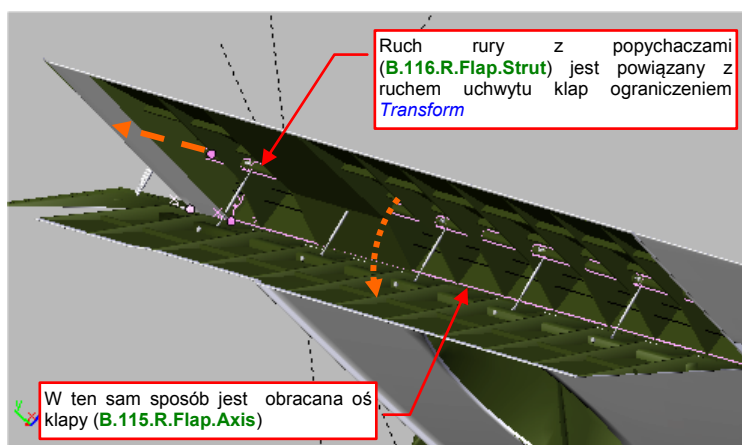
Rysunek 2.8.23 Mechanizm otwierania klapy

Aby złożyć klapę, wystarczyło by siłownik hydrauliczny przesunął tę rurę wzdłuż jej osi podłużnej. Przesuwająca się rura „ciągnęła” za sobą popychacze, a te — klapę. Przy otwieraniu wszystko poruszało się w przeciwną stronę. Mechanizując nasz model jak zwykle zamieniłem przyczynę na skutek: tu klapę poruszają obrót jej zawiasów, przeniesiony z ruchu odpowiedniego uchwyty (Rysunek 2.8.23b). Początkowo zamierzałem powiązać oś z uchwytem ograniczeniem *Locked Track*. Wymaga to jednak precyzyjnego ustalenia położenia uchwyty (por. str. 238). Dlatego ostatecznie zdecydowałem się użyć tu ograniczenia *Transformation* (przypisując każdą z osi klapy do „pustego” obiektu-rodzica poleceniem *Add Empty Parent* (por. str. 247). W ten sam sposób steruję przesunięciem rury z popychaczami.

Rysunek 2.8.24 przedstawia zespół pojedynczego popychacza (aby go dobrze pokazać, ukryłem wewnątrz klapy). Na rurę nałożona jest mufa. W środku mufy umieściłem niewidzialny obiekt (*Empty*) — cel dla ograniczeń *Locked Track* popychacza i mocującej go śruby. Śledzi go, obracając się wokół osi *Z*, oś *X* śruby. Śledzi go także oś *Y* popychacza, obracając się wokół jego osi *X*. Cała rura tylko się przesuw, ale sama mufa również obraca, śledząc osi *Y* śruby na drugim końcu popychacza.



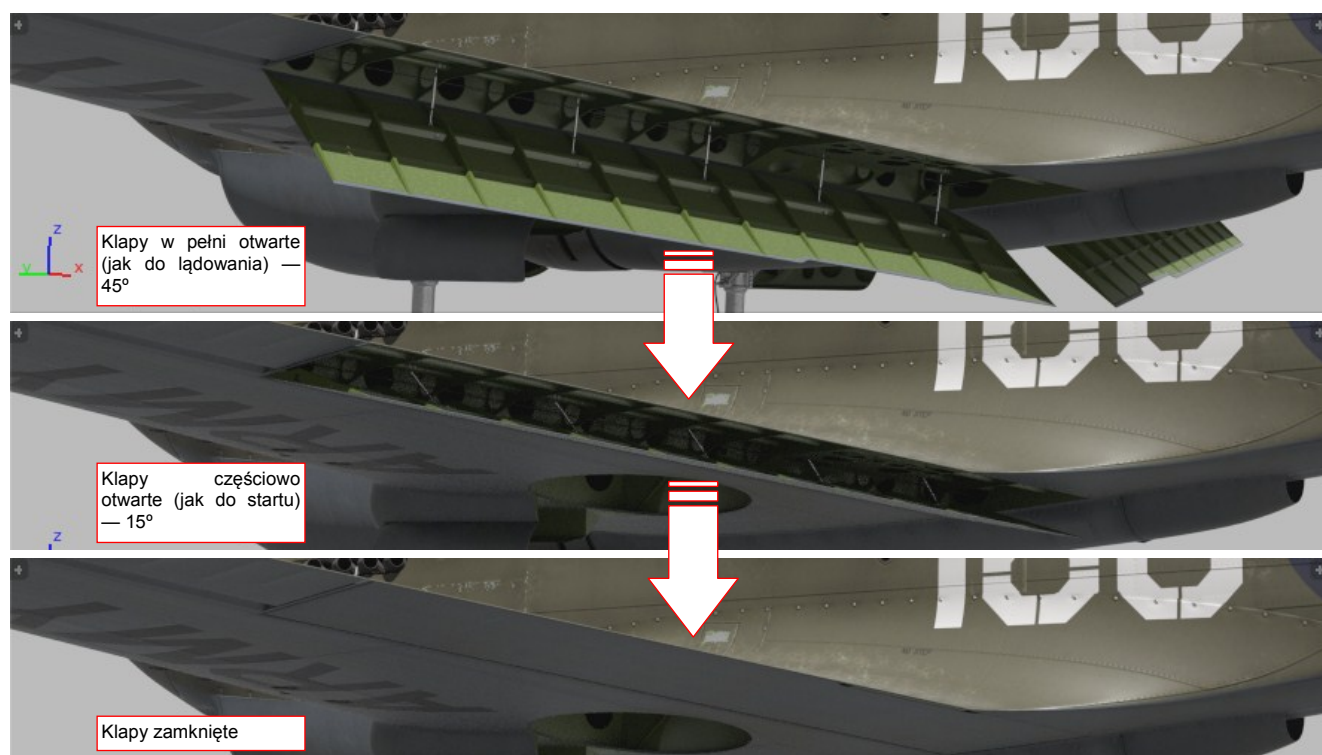
Rysunek 2.8.24 Zespół popychacza



Rysunek 2.8.25 Ograniczenia prawej klapy

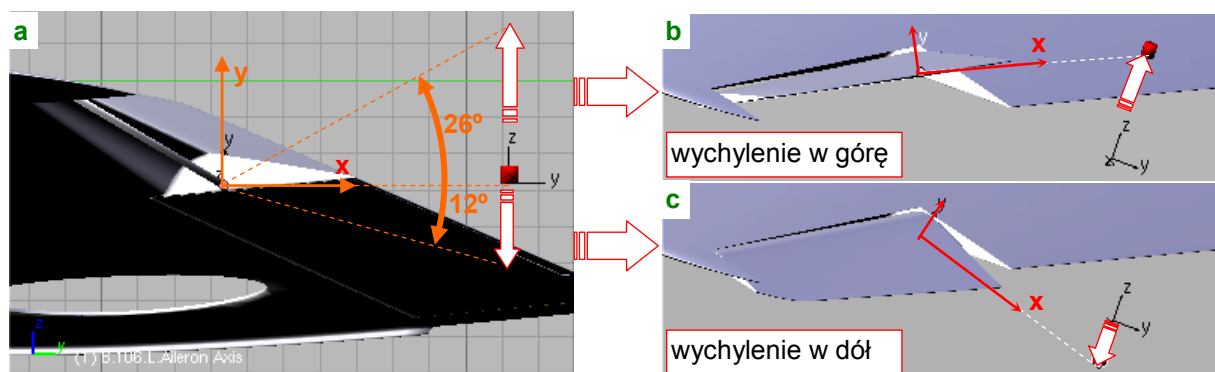
Gdy mechanizm lewej klapy zaczął działać, skopiowałem go do klapy prawej (Rysunek 2.8.25).

Rysunek 2.8.26 przedstawia kolejne fazy chowania klapy P-40. Zwróć uwagę na charakterystyczne „składanie” popychaczy na boki:



Rysunek 2.8.26 Kolejne fazy chowania klapy

Lotki także mają swoje osi obrotu. Dodałem do lewej uchwyt **X.105.Handle**, by nimi sterować (Rysunek 2.8.27):



Rysunek 2.8.27 Sterowanie lewą lotką

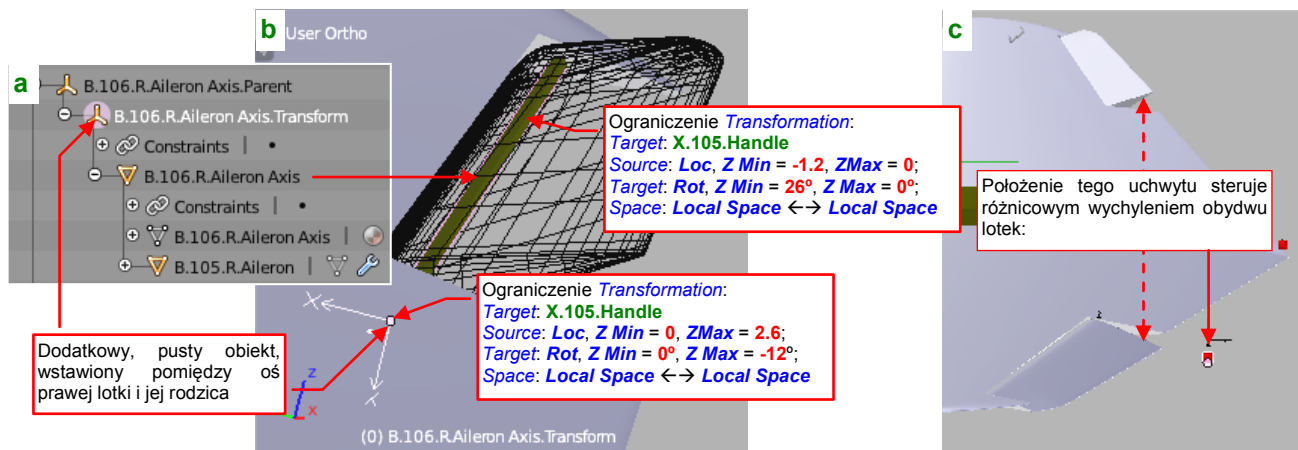
Podobnie jak oś klapy, oś lotki poprzedziłem pomocniczym obiektem „rodzica” (za pomocą dodatku [Add Empty Parent](#)). Pozwala to przypisać do osi lotki odpowiednie ograniczenie [Transformation](#), wiążące jej obrót z położeniem uchwytu (w granicach pokazanych przez Rysunek 2.8.28):



Rysunek 2.8.28 Sterowanie lewą lotką za pomocą ograniczenia [Transformation](#)

Położenie uchwytu **X.105.Handle** jest mierzone względem jego „rodzica” — **X.105.Base**. (Z kolei rodzicem **X.105.Base** jest lewe skrzydło).

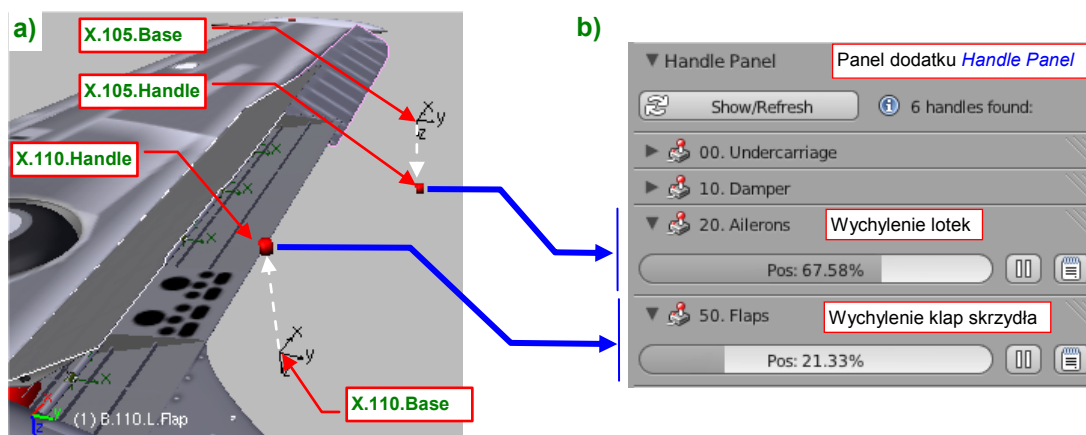
Lotki wychylają się różnicowo: gdy jedna porusza się w górę, to druga — w dół. (Gdy prawa jest podniesiona do góry o  $26^\circ$ , to lewa jest obrócona w dół o  $12^\circ$ , i na odwrot!)<sup>1</sup>. Najprościej byłoby przypisać osi prawej lotki takie ograniczenie **Transformation** jak dla lewej lotki. Jednak gdy w takim rozwiązaniu ustawisz uchwyt w pozycji „neutralnej”, to lewa lotka będzie także w pozycji neutralnej, a prawa — obrócona o  $14^\circ$ . To byłby błąd. Próbowaliśmy przypisać do tego obiektu dwa uzupełniające się ograniczenia **Transformation** (jedno na obrót w górę, drugie na obrót w dół). Niestety, w wyniku jakiegoś błędu w obecnej wersji Blender uwzględnia tylko ostatnie z tych ograniczeń. Pozostało więc dodać kolejny obiekt pomiędzy rodzica i oś prawej lotki (Rysunek 2.8.29a), i przypisać temu nowemu obiektowi obrót w górę, a osi — obrót w dół (Rysunek 2.8.29b):



Rysunek 2.8.29 Różnicowe wychylenia lotek, uzyskane za pomocą dwóch ograniczeń **Transformation**.

Dzięki tak przypisanym ograniczeniom różnicowym wychyleniem obydwu lotek steruje pojedynczy uchwyt (Rysunek 2.8.29c).

Na koniec pozostaje nadać uchwytom lotek i klap odpowiednie oznaczenia na naszym panelu sterowania modelem (**HandlePanel**) (Rysunek 2.8.30):

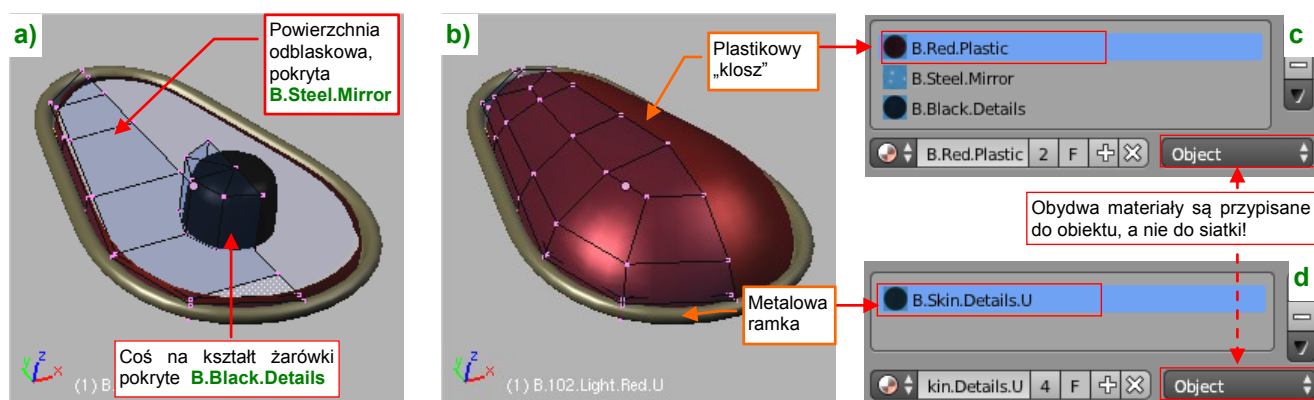


Rysunek 2.8.30 Nowe kontrolki, sterujące klapami i lotkami

Przedrostki etykiet kontrolki („20.”, „50.” — por. Rysunek 2.8.30b) ustalam, starając się je ułożyć w logicznej kolejności. (Pomiędzy lotkami i klapami zostawiłem miejsce w numeracji na stery kierunku i wysokości).

<sup>1</sup> Lotki wychylają się o różne kąty w górę i w dół, aby zapewnić (choć w przybliżeniu) podobne siły aerodynamiczne. Wynika to z niesymetrycznego profilu płata (górna część jest bardziej wypukła niż dolna) oraz faktu, że skrzydło jest zazwyczaj ustawione pod pewnym kątem do toru lotu samolotu. (Ten kąt, nazywany „kątem natarcia”, podczas lotu z dużą prędkością jest bardzo mały, ale może dochodzić do  $13-15^\circ$  podczas lądowania!)

Ostatnim szczegółem, który pozostał do ukończenia obszaru płatów, są światła pozycyjne. Element niewielki, ale wyeksponowany na końcówkach skrzydeł. Uformowałem najpierw ich podstawę (Rysunek 2.8.31a):

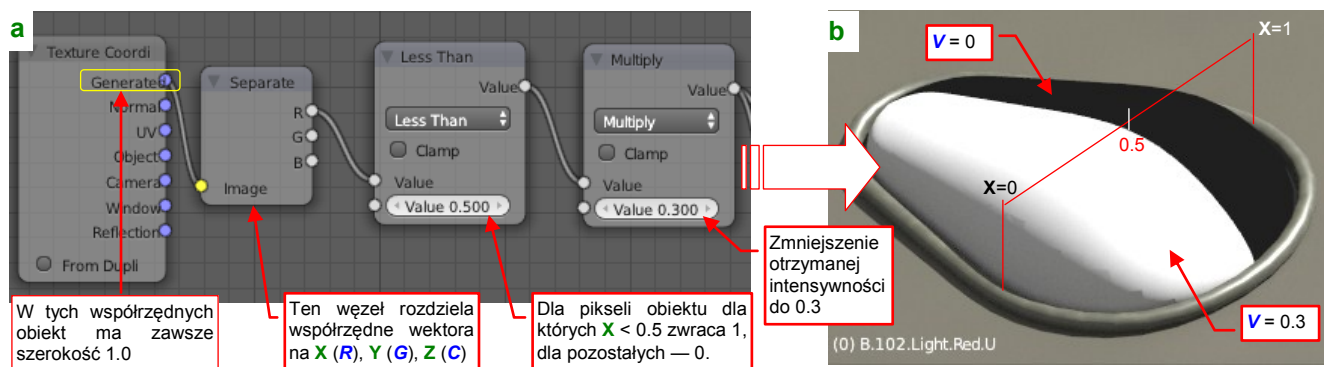


Rysunek 2.8.31 Światło pozycyjne — szczegóły

Potem zakryłem tę lampę „kloszem”, pokrytym materiałem **B.Red.Plastic** (Rysunek 2.8.31b). (Opis tego materiału znajdziesz na str. 418). Żadnej z tych siatek nie rozwijałem w **UV**, tylko przypisałem im „uproszczone” wersje barw powierzchni zewnętrznych samolotu: **B.Skin.Details.L** (**Neutral Gray**) i oliwkowozieloną **B.Skin.Details.U** (**Olive Drab**). (Obydwa materiały używają grupy **X.Dirty.Color**, której użyliśmy już we wnętrzu kabiny). Wykorzystamy je także do pokrycia innych detali zewnętrznych.

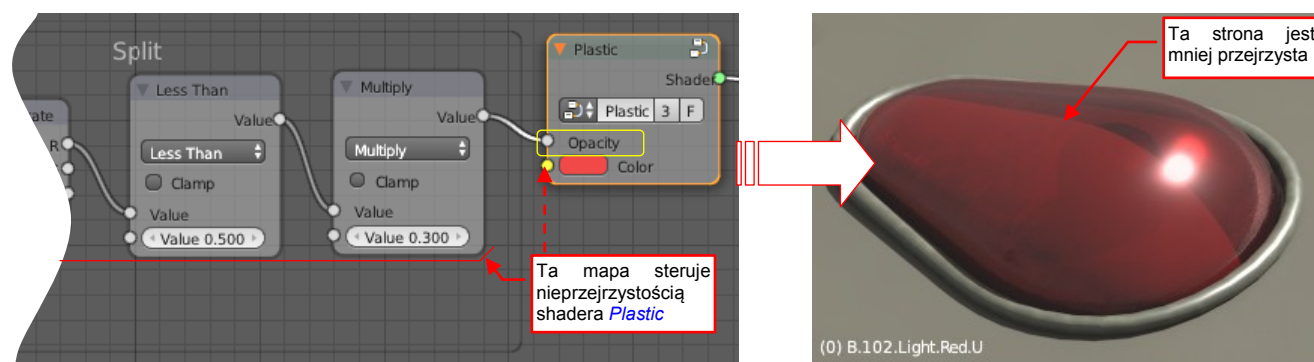
P-40 miał sześć światel pozycyjnych — po dwa z każdej strony skrzydeł i statecznika. Materiały plastiku i ramki pochodzą z przypisanego obiektu, a nie z siatki (Rysunek 2.8.31c,d) — by jej nie powielać.

Oslony światel pozycyjnych były z jednej strony nieprzezroczyste, by nie oślepić pilota. Aby ten efekt uzyskać, przygotowałem odpowiednią mapę nieprzezroczystości (Rysunek 2.8.32):



Rysunek 2.8.32 Mapa nieprzezroczystości

Potem podłączyłem ją do parametru **Opacity** shadera **Plastic** (Rysunek 2.8.34):

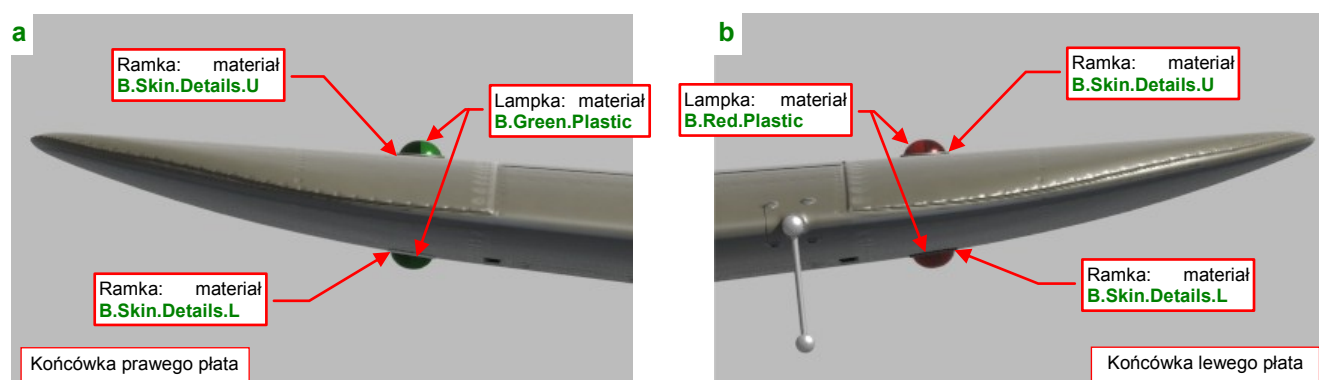


Rysunek 2.8.33 Mapa nieprzezroczystości

Zwróć uwagę, że materiał od strony pilota zachował resztkę przejrzystości (myślę że tak wygląda lepiej).



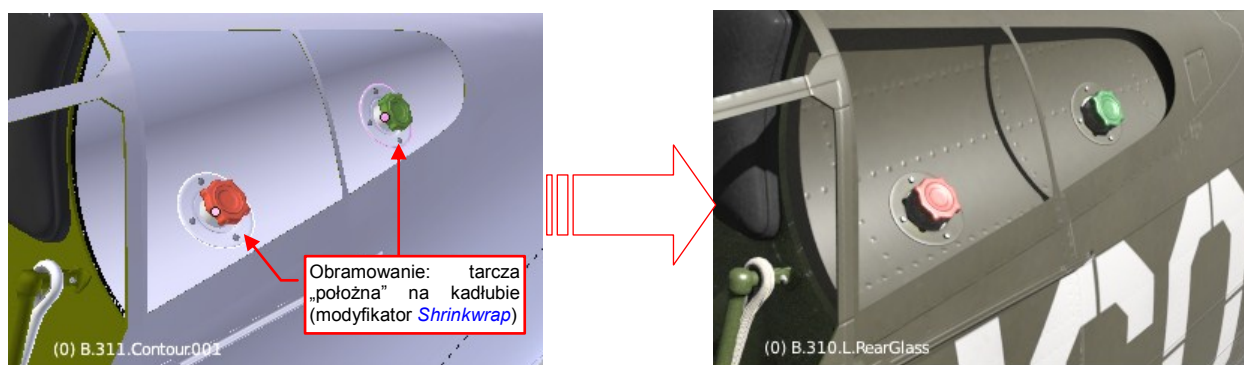
Inna sprawa, że cyzelowanie takiego szczegółu, jak lampka pozycyjna, wynika wyłącznie z czystej modelarskiej pasji. Na wszystkich renderach nie będzie tego widać — bo ten detal jest po prostu za mały (Rysunek 2.8.34):



Rysunek 2.8.34 Materiały, przypisane do poszczególnych obiektów (klonów) lampek pozycyjnych

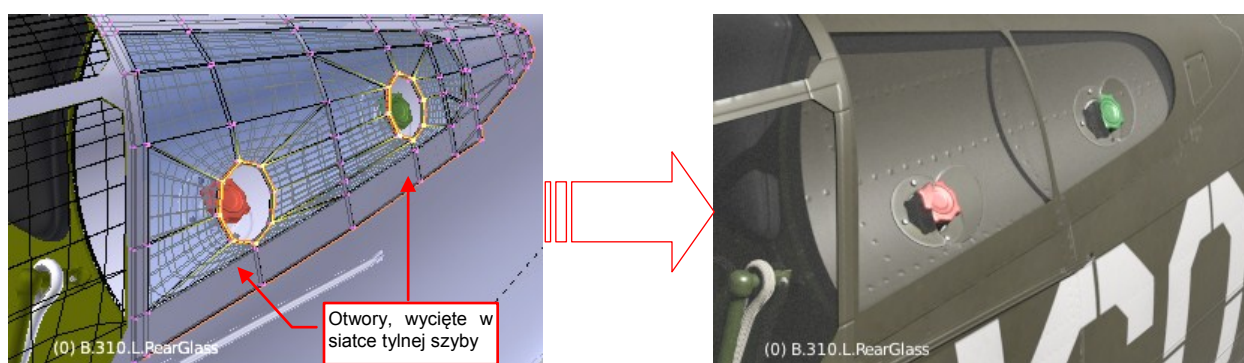
Rysunek 2.8.34a) pokazuje, jak wyglądają lampki na prawej końcówce płata, a Rysunek 2.8.34b) — na lewej. Prawda, że nie są duże? Zwróć za to uwagę, że mają różne barwy — mimo tego, że wszystkie cztery wykorzystują tę samą siatkę! To dzięki przypisaniu materiału do obiektu. (Dopóki jest to możliwe, staram się nie powiełać siatek. To bardzo się przydaje, gdy za jakiś czas okaże się, że ich kształt trzeba poprawić).

Wlew paliwa do zbiornika kadłubowego znajdował się w P-40 za kabiną pilota. W wersjach B/C były to dwa „korki”, umieszczone w ciągnących się za nią wyżłobieniach (Rysunek 2.8.35):



Rysunek 2.8.35 „Korki” tylnego zbiornika paliwa

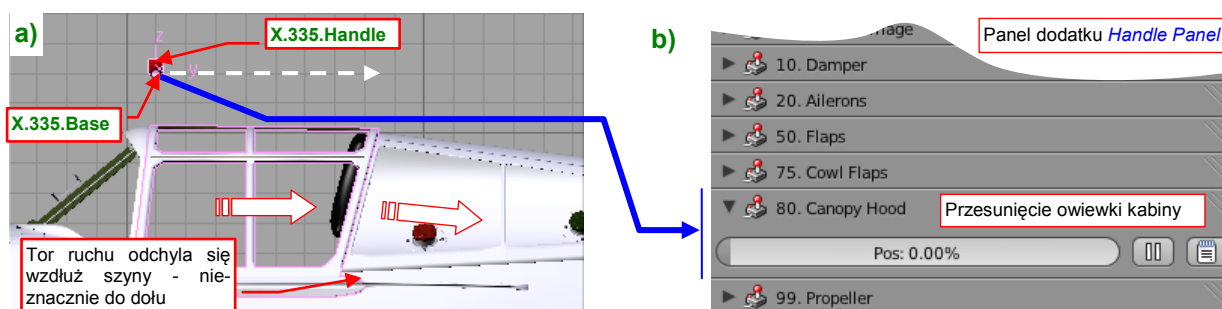
Było to dość dziwne rozwiązanie: w lewej szybie ponad „korkami” były wycięte dwa otwory (Rysunek 2.8.36):



Rysunek 2.8.36 Modyfikacja lewej tylnej szyby kabiny

Okrągłe wzmocnienia wokół „korków”, oraz umieszczone na nich śruby to wymodelowane siatki. Właściwie to można było je uzyskać także za pomocą tekstury nierówności. Zamodelowałem je trochę „z rozpędu”, a pozostawiłem jako przykład zastosowania ciekawej techniki modelowania. Każdy z tych kołnierzy to zwykły płaski pierścień. Ułożyłem je „jak naleśniki” na powierzchni kadłuba za pomocą modyfikatora **Shrinkwrap** (por. str. 302). Uzyskanie takiego nie zdeformowanego kształtu na teksturze rozwiniętej w **UV** mogło być trudne.

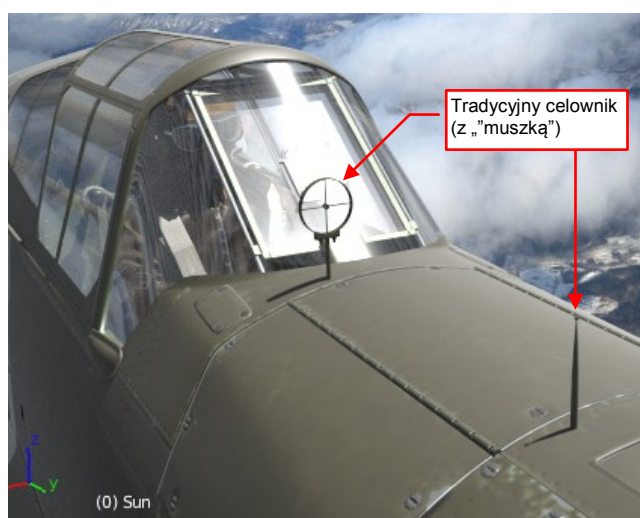
Proponuję dodać do naszego zestawu kontrolki sterujących modelem kolejną, odsuwającą owiewkę kabiny pilota (Rysunek 2.8.37). W domyślnym położeniu (0%) jest zamknięta (Rysunek 2.8.37b):



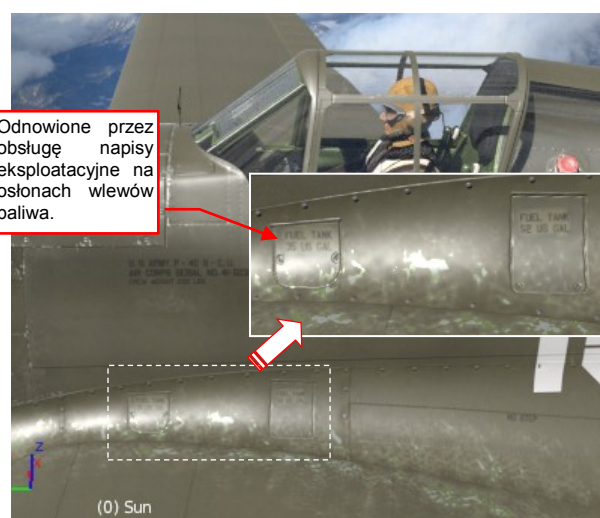
Rysunek 2.8.37 Nowa kontrolka, sterująca przesunięciem osłony kabiny

Szyny owiewki P-40 były nieznacznie wygięte do dołu (Rysunek 2.8.37a), stąd ruch owiewki jest powiązany z przesunięciem uchwytu (**X.335.Handle**) za pomocą dwóch ograniczeń *Transformation*. Jedno z nich wywołuje przesunięcie w poziomie, a drugie — to niewielki obrót (o  $1^\circ$ ).

Obok celownika odbłaskowego, P-40 miały także drugi, tradycyjny. Umieszczono go z boku (Rysunek 2.8.38):

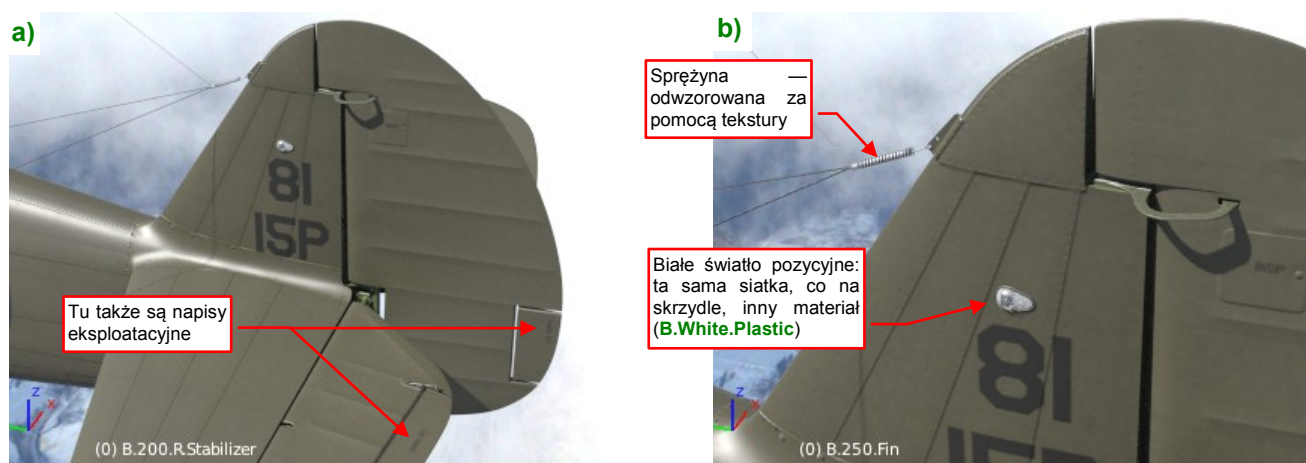


Rysunek 2.8.38 Dodatkowy celownik



Rysunek 2.8.39 Odnawiane napisy eksploatacyjne

Nawet w AVG, mimo „połowych” warunków eksploatacji, mechanicy odnawiali co ważniejsze napisy eksploatacyjne. Naniosłem na tekstury teksty z osłon wlewów paliwa do zbiorników skrzydłowych (Rysunek 2.8.39). Inne napisy, które potwierdzają zdjęcia, umieściłem na usterzeniu (Rysunek 2.8.40a):

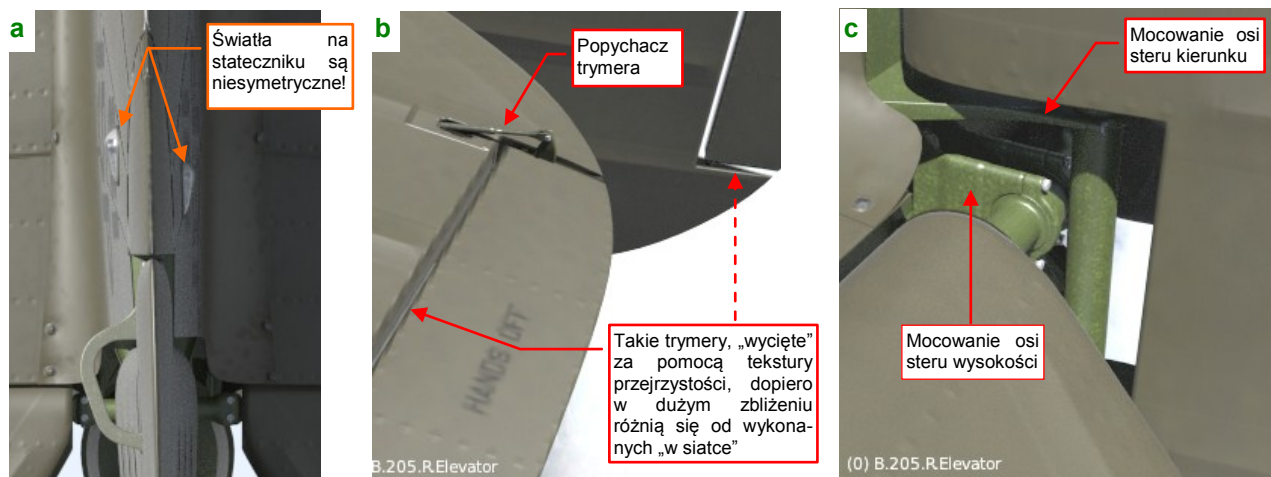


Rysunek 2.8.40 Usterzenie, wraz z anteną

Na tym etapie prac wiele części masz już gotowych: na przykład światło na usterzeniu (Rysunek 2.8.40b).

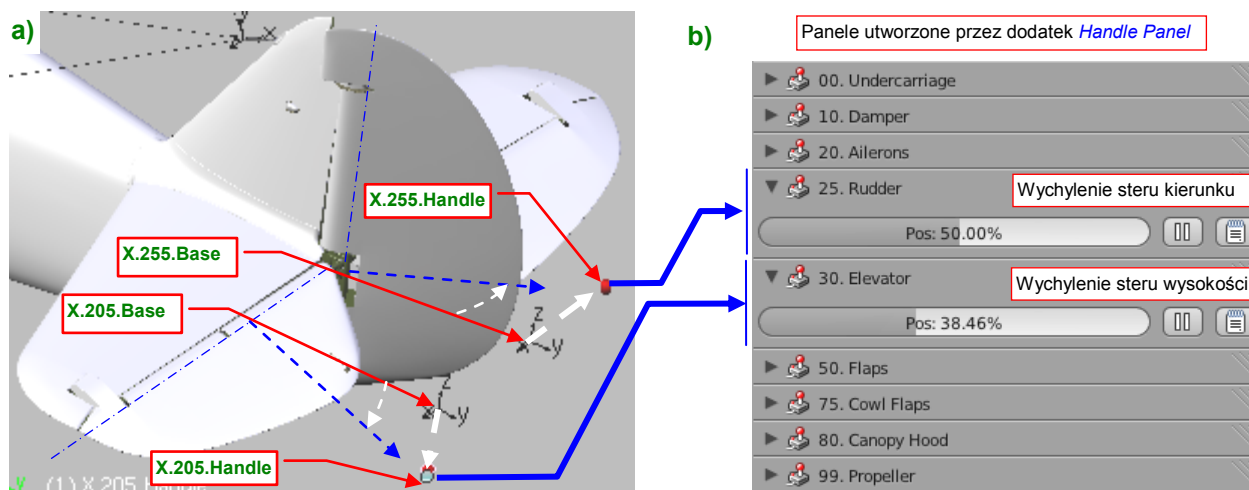
Innym elementem „z odzysku” jest sprężyna, naciągająca druty anteny<sup>1</sup> (Rysunek 2.8.40b). Odzworowałem ją za pomocą odpowiedniej tekstury przejrzystości. (Wykorzystałem materiał **B.Steel.Spring**, który stworzyliśmy w zespole kółka ogonowego — por. str. 46 i 49).

Jeżeli chodzi o światła sygnalizacyjne, to rysunek fabryczny statecznika pionowego P-40 pozwolił mi wyśledzić pewien szczegół: były umieszczone niesymetrycznie (Rysunek 2.8.41a). Kolejnym przydatnym drobiazgiem okazały się popychacze, wykorzystane w klapach. Mogłem ich użyć powtórnie, wraz ze śrubką i nakrętką, do zamodelowania napędu trymera (Rysunek 2.8.41b). W porównaniu z tym szczegółem krawędź trymera w dużych zbliżeniach wygląda trochę nieładnie — bo jest „wcięta” za pomocą tekstury. Jednak z nieco większych odległości (por. np. Rysunek 2.8.40a) wygląda zupełnie naturalnie.



Rysunek 2.8.41 Szczegóły usterzenia ogonowego

W przedniej części steru kierunku P-40 znajdowały się dwa otwory. W dolnym umieściłem orczyk na osi steru. Do końców orczyka przymocowane są linki. Do ich zamodelowania wykorzystałem tę samą siatkę, co w kabine pilota (w istocie to końce tych samych linek). W górnym odzworowałem okucia, mocujące osie sterów do stateczników (Rysunek 2.8.41c). Na koniec przygotowałem odpowiednie mechanizmy poruszania powierzchniami sterowymi (Rysunek 2.8.42) :



Rysunek 2.8.42 Nowe kontrolki, sterujące sterem kierunku i wysokością

<sup>1</sup> Swoją drogą, ta postać anteny była odziedziczona po P-36, i pochodziła z poprzedniej epoki rozwoju lotniczych urządzeń radiowych. Musiała stawiać poważny opór. Dla porównania: prototyp MiG-3 miał tylko jedną taką pochyłą „nitkę”, biegnącą od statecznika pionowego do osłony kabiny. Zastąpienie jej masztem z zupełnie poziomą anteną zwiększyło prędkość maksymalną o 3 km/h. A P-36 i P-40 B/C/D/E miały aż trzy takie druty, i to o wiele dłuższe! Myślę, że mogły spowalniać samolot nawet o jakieś 10 km/h. W P-40 maszt antenowy zaczęto stosować od wersji F.



W tej sekcji przebyliśmy drogę „od nosa do ogona” samolotu, uzupełniając po drodze wszystkie szczegóły zewnętrzne. Rysunek 2.8.43 przedstawia testowy render sceny w locie (o tym, jak „puścić w jego ruch” śmigło napiszę w następnym rozdziale):



Rysunek 2.8.43 Gotowy model — w powietrzu

Rysunek 2.8.44 przedstawia ten sam model stojący na abstrakcyjnej, białej płaszczyźnie<sup>1</sup>. W następnym rozdziale pokażę, jak go dopasować do jakiegoś przekonującego otoczenia.



Rysunek 2.8.44 Ten sam model postawiony na białej płaszczyźnie

<sup>1</sup> Model na tym ujęciu ma 870 tys. elementarnych ścian. Okazało się jednak że to zbyt wiele dla mojego 32-bitowego Blendera na starym Windows XP (2 GB RAM, zintegrowana grafika Intel) – program zgłaszał błąd po uruchomieniu renderowania (brakowało mu pamięci). Gdy się zorientowałem co jest przyczyną, wykonałem ten render na innym komputerze z Windows 7, używając 64-bitowej odmiany Blendera. Alternatywnie możesz w tej scenie wyłączyć warstwy z wnętrzem kabiny, bo i są tu prawie nie jest widoczne.



### Podsumownie

- Aby uzyskać w zbliżeniach modelu bardzo wyraźne napisy i oznaczenia, możesz powiększyć rozwinięcia UV wybranych siatek (str. 108);
- Praca nad detalami oznacza także konieczność poprawiania fragmentów tekstur. Może to być, na przykład, tekstura barwy (str. 121), czy nierówności (str. 120);
- Kolektor spalin najprościej jest wykonać z pojedynczej, prostej „rury wydechowej”, stosując kilka modyfikatorów. Modyfikatorów [Curve](#) użyłem do wygięcia siatki w odpowiedni kształt (str. 109). Potem powieliłem siatkę pojedynczej rury za pomocą modyfikatorów [Array](#) (str. 110);
- Ściany chłodnic cieczy można odwzorować za pomocą odpowiedniej tekstury przejrzystości (str. 111);
- W czasie prac nad ostatnimi szczegółami może się zawsze pojawić konieczność poprawy jakiegoś dawno uformowanego fragmentu poszycia. Tak się stało z otworami wyrzutników łusek karabinowych, które umieściłem na spodzie kadłuba (str. 113). Takie poprawki są często wymagają zmian w rozwinięciach UV, a te z kolei pociągają za sobą konieczność modyfikacji podstawowych tekstur;
- Do paneli sterujących modelem (dodatek [HandlePanel](#)) warto dodać także uchwyt odpowiedzialny za obrót śmigła (str. 113). To ułatwi „pozowanie” modelu na ziemi i uzyskanie efektu rozmycia łopat w locie;
- Otwory w osłonach luf karabinów maszynowych najłatwiej jest uzyskać za pomocą tekstury przejrzystości (str. 114);
- Do zestawu paneli sterowania modelem warto dodać, oprócz sterów kierunku, lotek i klap, także owiewkę kabiny (str. 121);

## 2.9 Podsumowanie

W tej sekcji zakończyliśmy pracę nad samym modelem. Rysunek 2.9.1 przedstawia jak wygląda uzyskany efekt (to inne ujęcie modelu „na białym tle”, pokazanego na koniec poprzedniej sekcji):



Rysunek 2.9.1 Obecny stan naszego modelu

W takim momencie jak ten warto jest przejrzeć plik Blendera, zawierającym naszą pracę. Zazwyczaj zauważysz w nim wiele zapomnianych zakamarków, wymagających uporządkowania. Na przykład — przejrzyj listę obrazów (w oknie [UV/Image Editor](#)). Często znajdziesz w niej wiele niepotrzebnych obiektów (Rysunek 2.9.2):



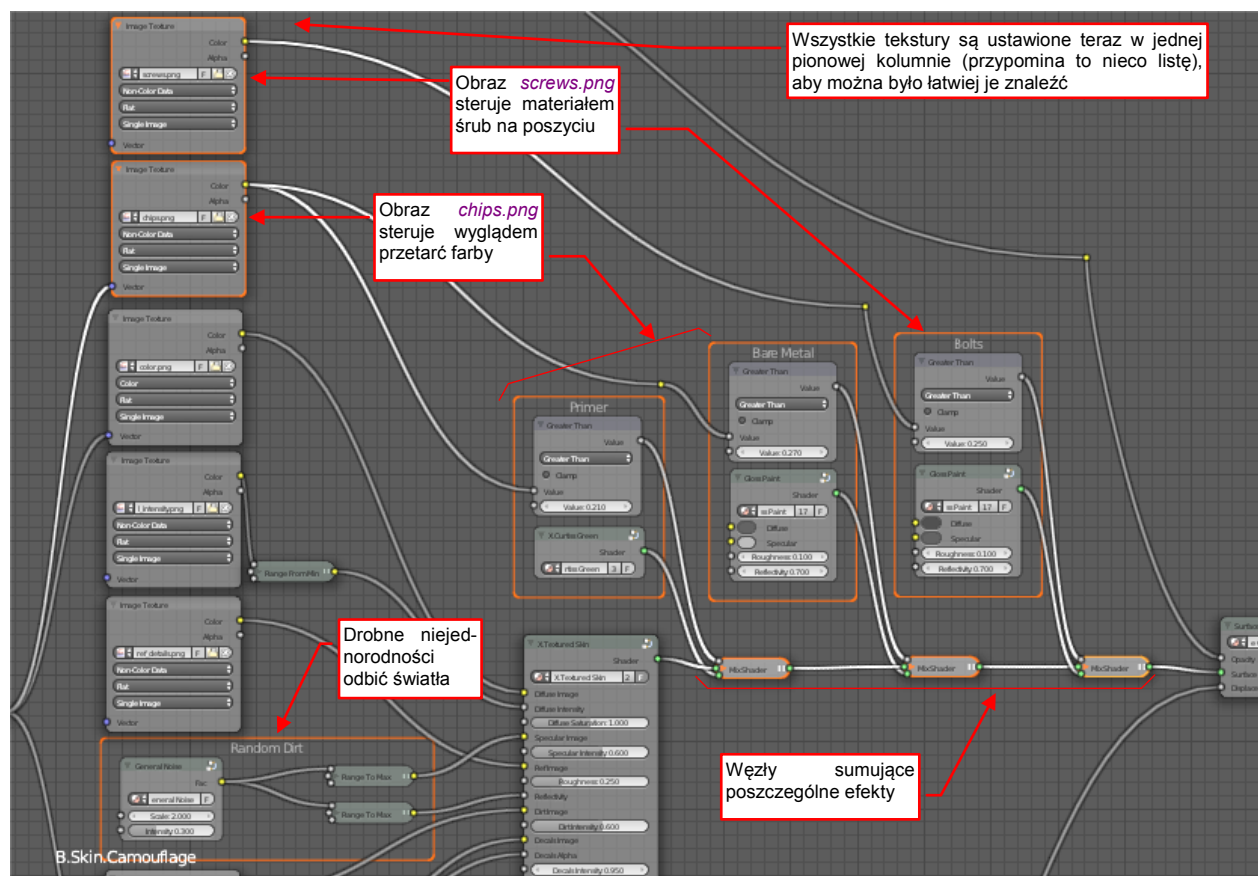
Rysunek 2.9.2 Usuwanie z pliku referencji do niepotrzebnych obrazów

Nie używane obrazy dość opornie znikają podczas kolejnych zapisów pliku *\*.blend* na dysk. Dzieje się tak dlatego, że ich „licznik użycia” (*reference count* — por. str. 462) jest zwiększany przez każde wyświetlenie w jakimkolwiek oknie [UV/Image Editor](#). Takie okna z załadowanym i zapomnianym obrazem mogą się kryć pod każdym z widoków *3D View*. (Wystarczy że kiedyś przełączyłeś je na [Image Editor](#), otworzyłeś w nim obraz, a potem przełączyłeś z powrotem na *3D View*). Tych przypisań nie znajdziesz posługując się [Outlinarem](#). W oknie [Outliner](#) nie znajdziesz także powiązań obrazu ze ścianami siatki (co prawda wykorzystywaliśmy do tego obraz *Test* — por. Wyciąg III — ale mogły się gdzieś zapodziać i inne) .

- Najszybszym sposobem na pozbycie się niepotrzebnych obrazków jest odłączyć je w [UV/Image Editor](#) z wciśniętym klawiszem **Shift** (Rysunek 2.9.2).

Taka operacja „zrywa” wszelkie połączenia bloku danych z resztą sceny. W efekcie ściany jakiejś siatki mogą stracić przypisanie do tego obrazu (w trybie [Blender Render](#)) ale to nie przeszkadza w renderowaniu.

W trakcie prac nad szczegółami modelu stworzyliśmy wiele nowych materiałów: dla wnętrza kabiny (str. 83) i dla różnorodnych detali (gumy, skóry, tkaniny pasów, plastiku). Rozbudowaliśmy także podstawowy materiał powierzchni zewnętrznych (**B.Skin.Camouflage**) o dodatkowe efekty (Rysunek 2.9.3):



Rysunek 2.9.3 Nowe węzły w materiale **B.Skin.Camouflage**

W stosunku do schematu omówionego w Wyciągu III, **B.Skin.Camouflage** zawiera teraz kilka nowych elementów poprawiających jego wygląd w zbliżeniach. Są to:

- efekt przetarc „do gołego metalu” (grupa **Bare Metal** i sterująca nią tekstura **chips.png**);
- efekt przetarc „do farby podkładowej” (grupa **Primer**, także sterowana przez tę samą teksturę **chips.png**);
- podkreślenie śrub za pomocą odrębnego shadera (grupa **Bolts** i sterująca nią tekstura **screws.png**);
- zróżnicowanie odbić światła od powierzchni modelu (zespół **Random Dirt**, wykorzystujący teksturę proceduralną **General Noise**);

Szczegółowy opis efektów przetarc (zespoły **Bare Metal**, **Primer** i **Bolts**) znajdziesz na str. 170-172. Opis odwzorowania drobnych nieregularności w połyskliwości powierzchni (zespół **Random Dirt**) za pomocą tekstury proceduralnej **General Noise** znajdziesz na str. 394-395.

W materiale **B.Skin.Camouflage** zostały wykorzystane dwa dodatkowe obrazy:

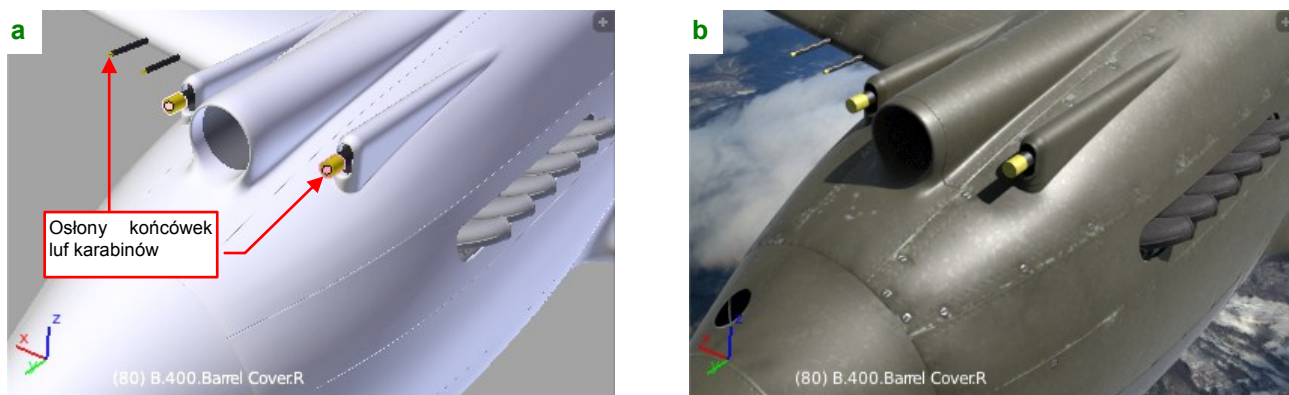
- **screws.png**: wyeksportowana z GIMP (*skin.xcf*) warstwa **Color-Screws.png** (na czarnym tle);
- **chips.png**: wyeksportowana z GIMP (*skin.xcf*) warstwa **Color-Chips** (na czarnym tle);

Jednocześnie użyte w materiale obrazy zostały podzielone na grupy:

- niezależne od kamuflażu: przechowywane w folderze głównym (np. *model/p40/textures/8.08*);
- zależne od kamuflażu: przechowywane w podfolderach odpowiednich dla wersji (**USAAC** lub **AVG**);

Zarówno obraz przetarc (**chips.png**) jak i śrub (**screws.png**) umieściłem w podfolderach zależnych od kamuflażu (bo każda maszyna nosi nieco inne ślady zużycia eksploatacyjnego).

Na koniec przypomniałem sobie o pewnym drobnym szczególe: obsługa naziemna przed każdym startem zalepiała lufy karabinów taśmą klejącą. Ta osłona chroniła przed kurzem, który mógł spowodować późniejsze zacięcie broni. Przypuszczam że kolory tych „zaklejek” bywały różne — czerwone, zielone, białe, żółte... Aby ten detal odwzorować, umieściłem na każdej z luf mały cylinder (Rysunek 2.9.4a) w kolorze żółtym (użyłem materiału **B.Yellow.Details** — Rysunek 2.9.4b):



**Rysunek 2.9.4 Osłona luf karabinów przed kurzem (żółta taśma klejąca)**

To oczywiście element opcjonalny — po strzelaniu z broni pokładowej resztki tych taśm mogły być zerwane przez strumień powietrza.



## Rozdział 3. Kompozycja sceny

W tym rozdziale ostatecznie kończę pracę nad naszym modelem, wkomponowując go w jakiś przekonujący obrazek.

Ten rozdział zawiera tylko dwie sekcje. Pierwsza sekcja opisuje zagadnienia związane z przygotowaniem sceny w locie. Pokażę jak uzyskać poprawny obrazu rozmytych ruchem łopat śmigła, lepiej dopasować model do zdjęcia podstawionego jako tło, czy nadać scenie dynamikę poprzez rozmycie otoczenia.

Druga sekcja pokazuje, jak „postawić samolot na ziemi”, którą widać na podstawionej za modelem fotografii. W tym przypadku kluczową sprawą jest właściwe ustawienie samolotu i uzyskanie jego cienia na podłożu. Model postawimy na lotnisku polowym, więc pokażę jak stworzyć trawę i inne dodające realizmu szczegóły otoczenia.

- Podstawowe modele opisane w tym rozdziale znajdziesz w plikach *\*.blend* z pakietu *p40.zip*, a foldery z obrazami ich tekstur — w pakiecie *textures2.zip*. (Adres miejsca, z którego można pobrać te pliki, znajdziesz na str. 16). Pliki *\*.blend* z dodatkowymi scenami pokazanymi w tym rozdziale znajdziesz w pakiecie *p40w.zip*.

Numery rozdziałów w tym wyciągu są inne niż w kompletnej książce, dlatego pliki z rezultatami sekcji tego rozdziału noszą nazwy zaczynające się od *P40B-9.\**, a foldery z teksturami — *09.\**.

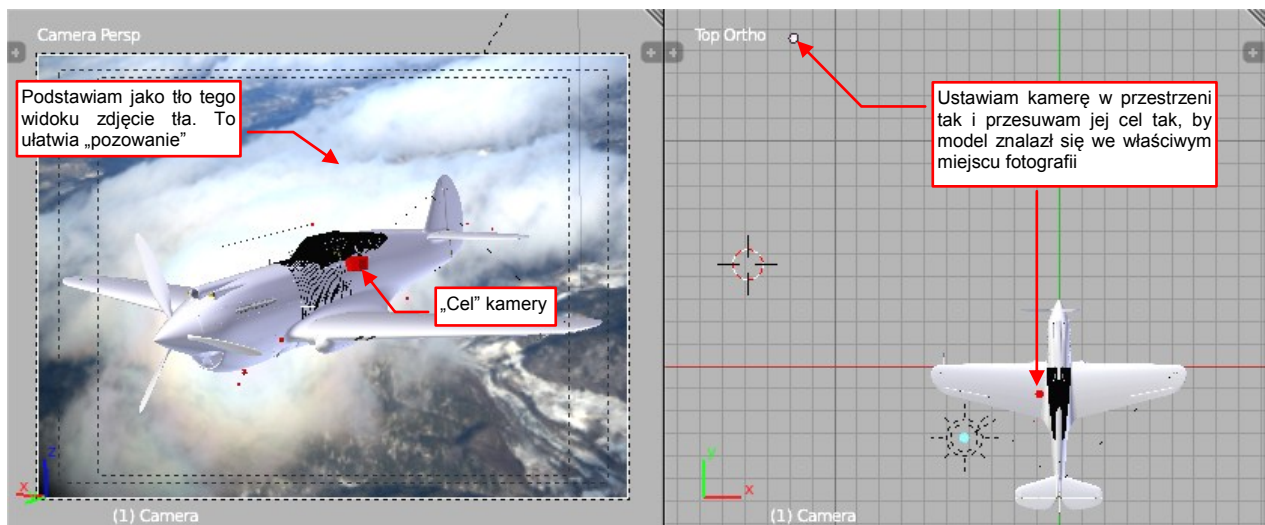
Renderując przykłady z tego rozdziału w Cycles na CPU natknąłem się na pewne ograniczenie: 32-bitowa wersja Blendera potrafi obsłużyć sceny zawierające do 400 – 420 tys. ścian (800-900 tys. trójkątów). Nasz model w różnych konfiguracjach ma ich zazwyczaj więcej — około 600 tys. ścian. Stąd podczas przygotowania danych do renderowania 32-bitowa wersja Blendera zgłosi błąd. (Program nie jest w stanie otrzymać od systemu operacyjnego Windows żądanej ilości pamięci RAM).

- Zalecam renderowanie przedstawionych w tym rozdziale przykładów w 64-bitowej wersji Blendera.

### 3.1 Realistyczna scena w locie

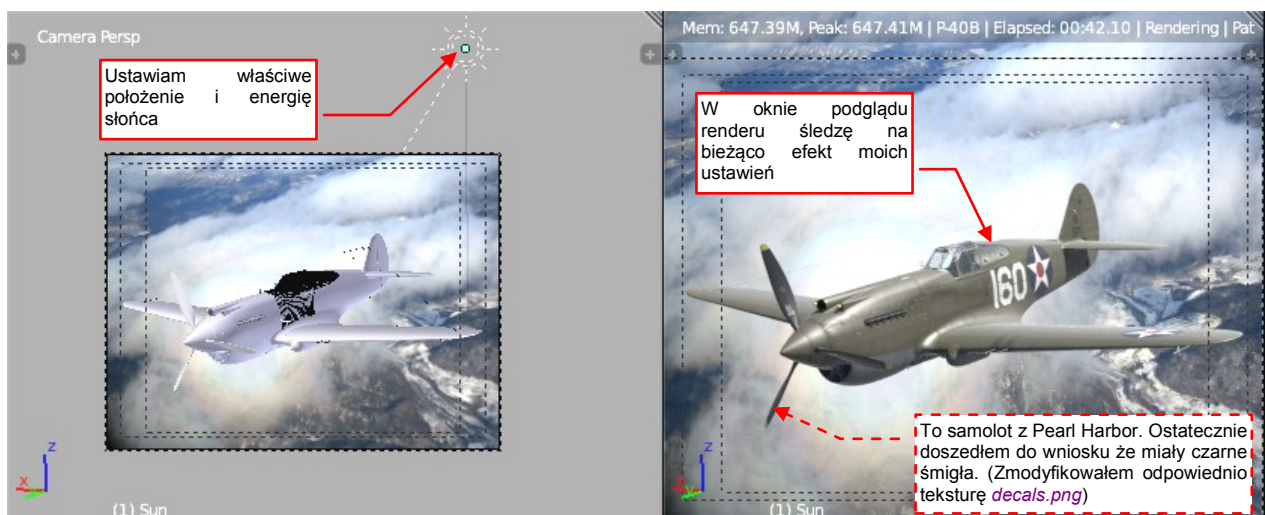
W tej sekcji pokażę, jak przygotować scenę w locie. Omówię poszczególne zagadnienia, posługując się przykładami z różnych scen<sup>1</sup>. Zaczynam od wyszukania w Internecie jakiegoś darmowego zdjęcia o odpowiedniej rozdzielczości. Potem podstawiam je za modelem, komponując prostą scenę. Jest to dla mnie swoisty „ostateczny test” realizmu: jeżeli mój samolot będzie dobrze wyglądał na tle fotografii, oznacza to że udało mi się go dobrze odwzorować. Specjalnie zresztą wyszukuję zdjęcia okolic, nad którymi naprawdę latała konkretna maszyna, której barwy nosi mój model. Chcę w ten sposób odtworzyć jej „naturalne otoczenie”.

Pracę nad sceną zaczynam od podstawienia za widok z kamery obrazu tła. Rysunek 3.1.1 pokazuje takie ustawienie (jest to *Background Image*, przypisany do osi *Camera* — por. str. 198). Potem w widokach z przodu i z góry przesuwam odpowiednio kamerę, „pozując” model w tym tle:



Rysunek 3.1.1 „Pozowanie” — dopasowanie pozycji kamery (a w efekcie – modelu) do tła

Następnie włączam podgląd renderu i ustalam pozycję i energię (*Strength*) słońca *Sun* i otoczenia *World*. Robię to tak, by światło padało na model z tej samej strony, co na fotografii (Rysunek 3.1.2):

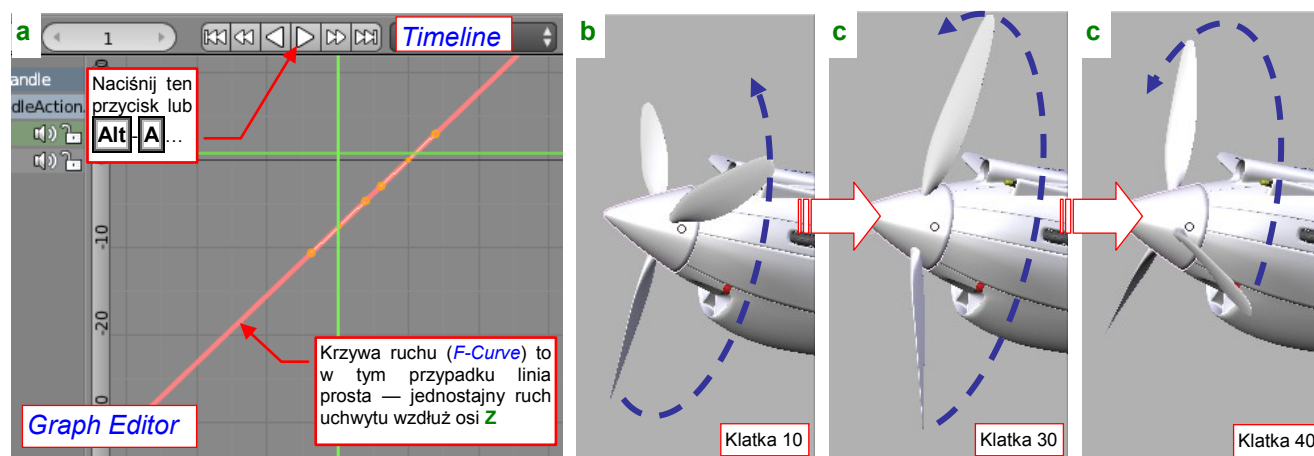


Rysunek 3.1.2 Dopasowanie położenia światła słońca i energii otoczenia

Dla tego zdjęcia ustawiłem *Sun:Strength* = 25 i *Background:Strength* = 2;

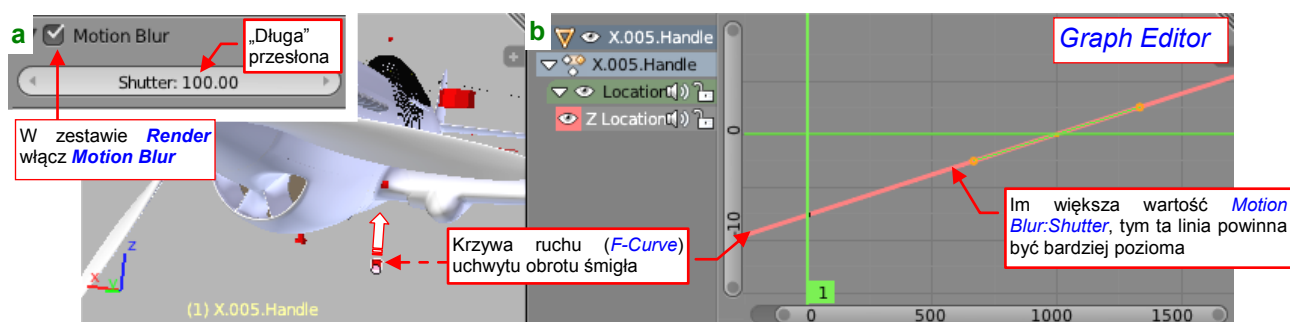
<sup>1</sup> W związku z tym, trochę nietypowo, w folderze *p40\history1* znajdziesz nie jeden, a kilka plików Blendera związanych z tą sekcją książki (oprócz *P40B-8.01.blend* są tu kolejne, oznaczone *p40B-8.01w1.blend* ... *p40B-8.01w6.blend*). Każdy z nich to pojedyncza scena, z innym tłem. Źródła zdjęć, które wykorzystałem jako tło, podaję w podpisach pod ilustracją z ostatecznym renderem każdej sceny.

Na razie na renderze nie są także widoczne łopaty śmigła, a sam kołpak jest nieruchomy. Czas już, by zaczęły się kręcić! Przypisałem sekwencję ruchu (tzw. „krzywą F” — *F-Curve*) do uchwytu, poruszającego kołpakiem (Rysunek 3.1.3). (Szczegóły tworzenia tej animacji są opisane na str. 425). Naciśnij **Alt-A** w którymś z okien *3D View*, by zobaczyć jak to działa — Rysunek 3.1.3b,c,d):



Rysunek 3.1.3 Przypisanie krzywej ruchu do uchwytu sterującego obrotem śmigła

Gdy animacja śmigła jest już gotowa, możemy uzyskać w naszej scenie efekt rozmycia ruchu. Wystarczy w zestawie *Render* włączyć opcję *Motion Blur*. Aby obraz wirujących łopat śmigła miał odpowiedni kształt, zastosuj jakąś dużą wartość parametru *Shutter* (Rysunek 3.1.4a) i odpowiednio spowolnij obrót śmigła (Rysunek 3.1.4b). (Więcej na temat tych ustawień znajdziesz na str. 429).



Rysunek 3.1.4 Dobór wartości przesłony i prędkości obrotu śmigła

Potem wystarczy wcisnąć przycisk *Render* (lub klawisz **F12**) by otrzymać rezultat (Rysunek 3.1.5):

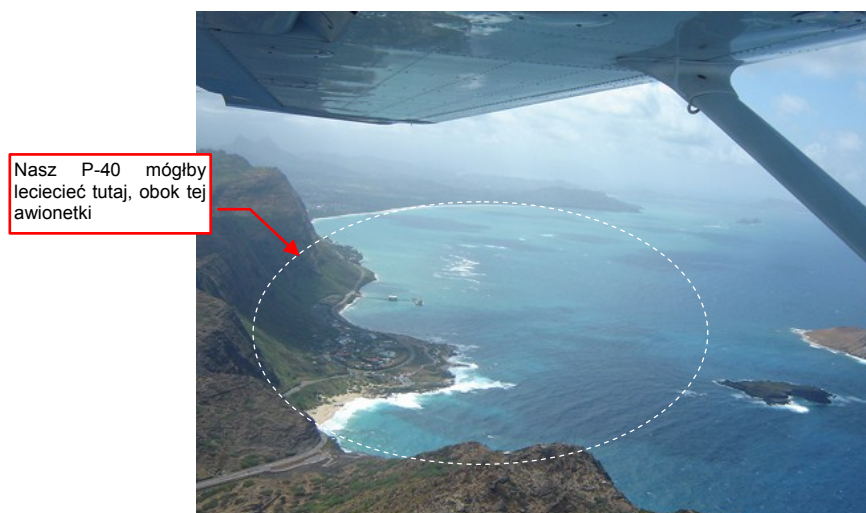


Rysunek 3.1.5 Efekt „rozmycia” wirujących łopat śmigła (tło: © Tomo Yun, [www.yunphoto.net](http://www.yunphoto.net))

Efekt jest całkiem zadowalający, w dodatku nie wydłużył znacznie czasu renderowania.

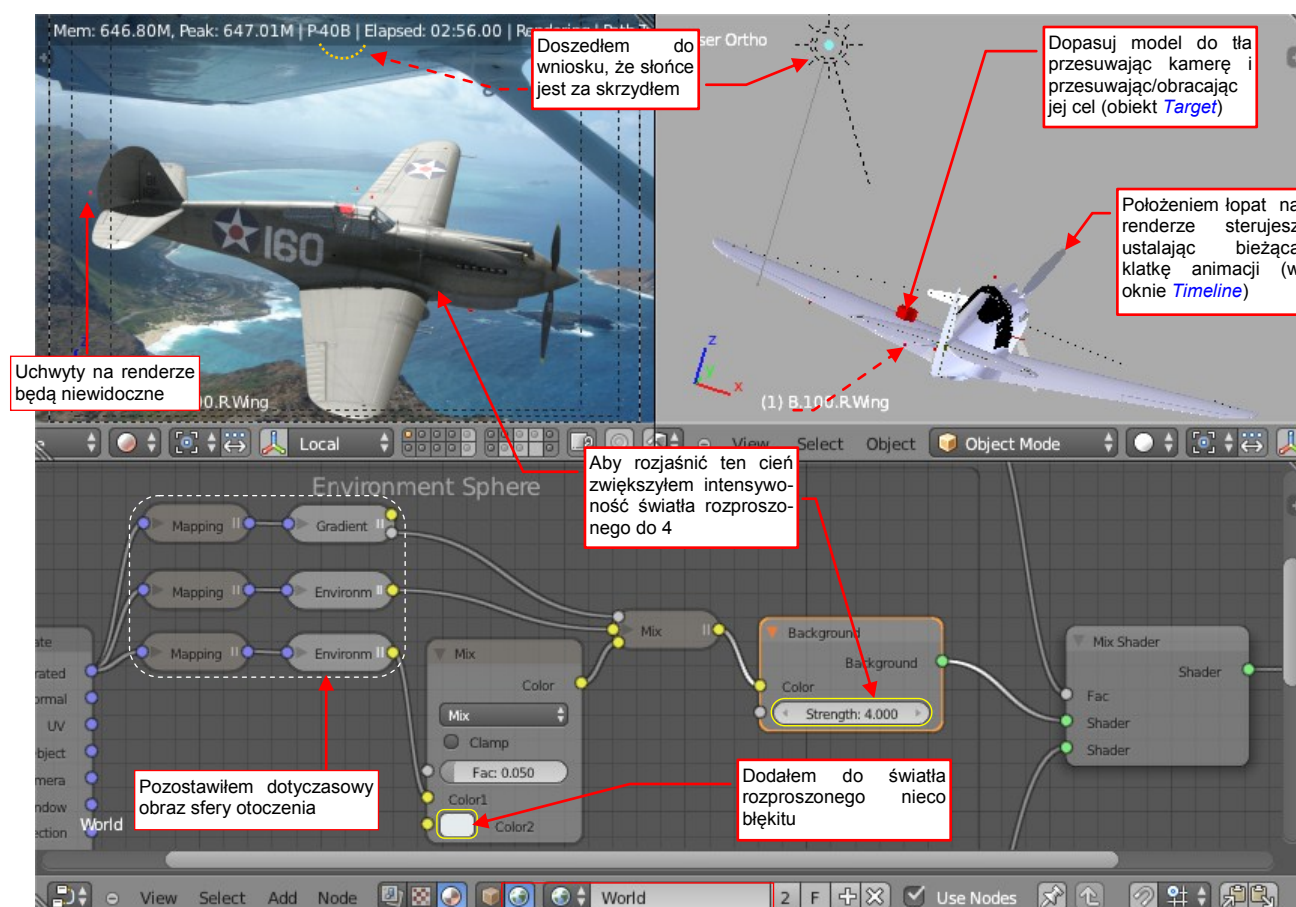


Co prawda model, tło i oświetlenie przedstawione przez Rysunek 3.1.5 dopasowywaliśmy już w Wyciągu III. W istocie te góry tworzą uniwersalne tło, pasujące do wielu maszyn. Jednak ta „biała 160” to maszyna stacjonująca w 1941r w Pearl Harbor, i na pewno nikt nie miał okazji zrobić jej zdjęcia ponad szczytami pokrytymi śniegiem (w dodatku te konkretne góry znajdują się w Japonii). Stąd spróbowałem poszukać obrazu na jakieś bardziej adekwatne tło. W Internecie znalazłem ciekawe ujęcie wyspy Oahu wykonane „spod skrzydła” awionetki (Rysunek 3.1.6):



Rysunek 3.1.6 Fragment wybrzeża Hawajów (wyspa Oahu, [www.archerbravo.com](http://www.archerbravo.com))

Spróbujmy wykorzystać to tło do stworzenia nowej sceny (Rysunek 3.1.7):



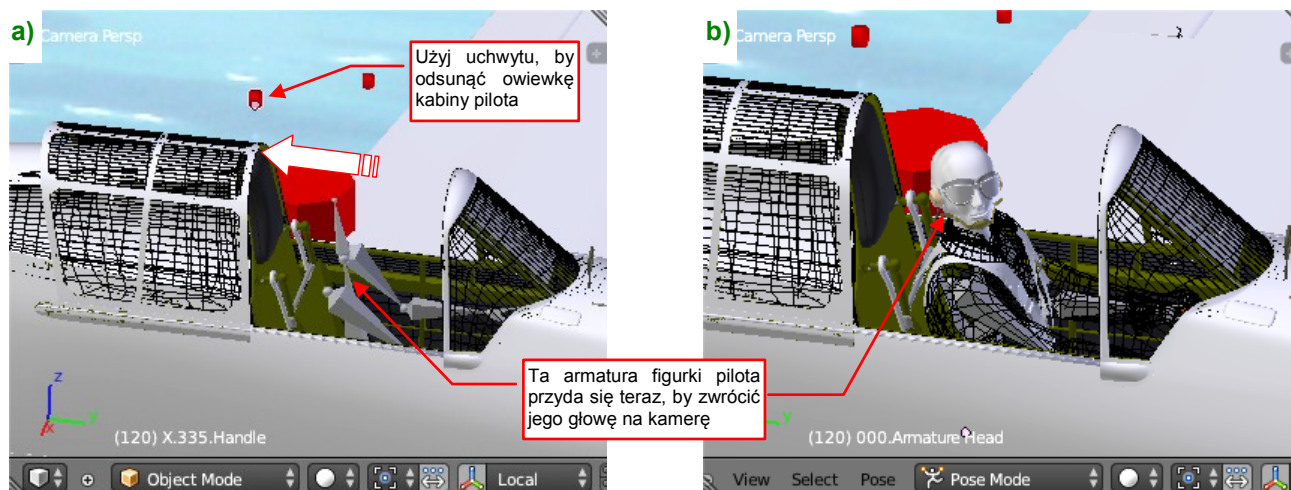
Rysunek 3.1.7 Wkomponowanie modelu w nowe tło

Wykorzystałem tu dotychczasowy obraz otoczenia (por. Wyciąg III) — to metoda mniej poprawna, ale szybsza. Zwiększyłem tylko jego energię (do 4) i zmieniłem odcień nieba na bardziej błękitny (por. str. 357).



(Taka adaptacja światła rozproszonego jest niezbędna, by je zbliżyć do warunków oświetlenia widocznych na zdjęciu). Dla tak silnego światła otoczenia musiałem zmniejszyć wartość *Reflectivity* szkła owiewki do 0.1.

Aby lecieć w pobliżu awionetki, P-40 musiał bardzo zwolnić (być może wysunął kalpy, zasłonięte w tym ujęciu przez skrzydła). W takich warunkach piloci często odsuwają owiewki kabiny. Dlatego zdecydowałem się odsunąć osłonę kabiny pilota (Rysunek 3.1.8a). Można to zrobić używając dodatku *Handle Panel* (por. str. 265), lub uchwytu z warstwy 10. Jeżeli chodzi o samego pilota — pomyślałem, że powinien patrzeć na samolot, do którego się zbliża. Przełączyłem się więc w *Pose Mode* i obróciłem odpowiednio „kość” *Head* z armatury umieszczonej wewnątrz tej figurki (Rysunek 3.1.8b):



Rysunek 3.1.8 „Pozowanie” pilota

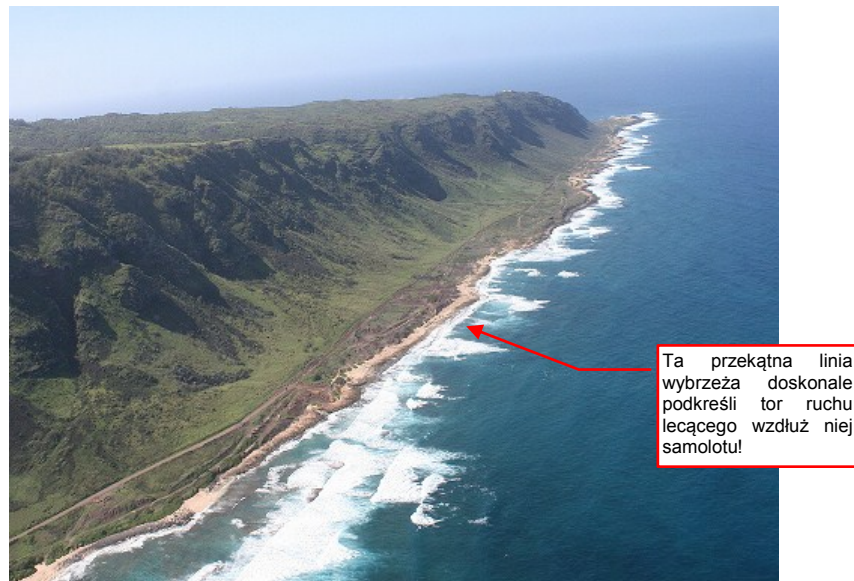
Rysunek 3.1.9 przedstawia uzyskany rezultat:



Rysunek 3.1.9 Rezultat — model wstawiony w fotografię

Zwróć uwagę ze stery nie są w położeniu neutralnym, a kłapy chłodnicy są szeroko otwarte.

W podobnej morsko – zielonej tonacji jest kolejne zdjęcie (Rysunek 3.1.10). To półwysep Kaena, na północy wyspy Ohau. Nieco dalej wzdłuż tego wybrzeża znajdowało się polowe lotnisko, które 7 grudnia 1941r uniknęło ataku japońskich myśliwców. Nasza „biała 160” jest jedną z maszyn, której udało się stamtąd wystartować.



Rysunek 3.1.10 Kolejne tło — półwysep Kaena ([www.hawaii.gov](http://www.hawaii.gov))

Wybrałem to zdjęcie, gdyż przekątna linia brzegu dobrze się komponuje z modelem, podkreślając przestrzeń i tor lotu maszyny (Rysunek 3.1.11):



Rysunek 3.1.11 Pierwszy rezultat

Tak jak poprzednio, dobrałem odpowiednie położenie słońca (sugerowałem się cieniami na zboczach wzgórz). Energia słońca jest nieco większa niż poprzednio (*Strength* = 30). Światło rozproszone ma umiarkowaną intensywność (2.5) i kolor biały. Jednak obraz, który otrzymaliśmy jest lekko rozmyty. Czy można to jakoś poprawić?



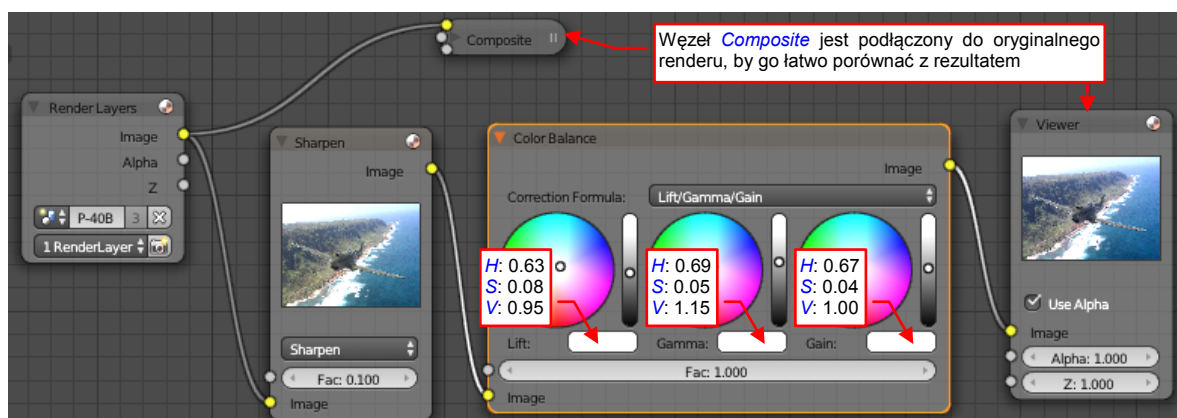
Tak, do końcowych poprawek obrazu lub filmu służy w Blenderze tzw. kompozytor. Budujesz w nim z węzłów odpowiedni schemat końcowego przetwarzania wyrenderowanego obrazu (to tzw. *postprocessing*). Wprowadzenie do budowania schematów kompozytora znajdziesz na str. 432.

W tym konkretnym przypadku wykorzystałem węzeł *Filter* → *Sharpen*, o natężeniu 10% (Rysunek 3.1.12):



Rysunek 3.1.12 Wyostrzenie renderowanego obrazu (podgląd z *Node Editor*)

Potem dodałem węzeł *Color Balance* (por. str. 438). Przesunąłem w nim barwy obrazu w kierunku błękitu, zmniejszając odrobinę intensywność odbłyśków i zwiększając intensywność średnich odcieni (Rysunek 3.1.13):



Rysunek 3.1.13 Zmiana tonacji obrazu za pomocą węzła *Color Balance*

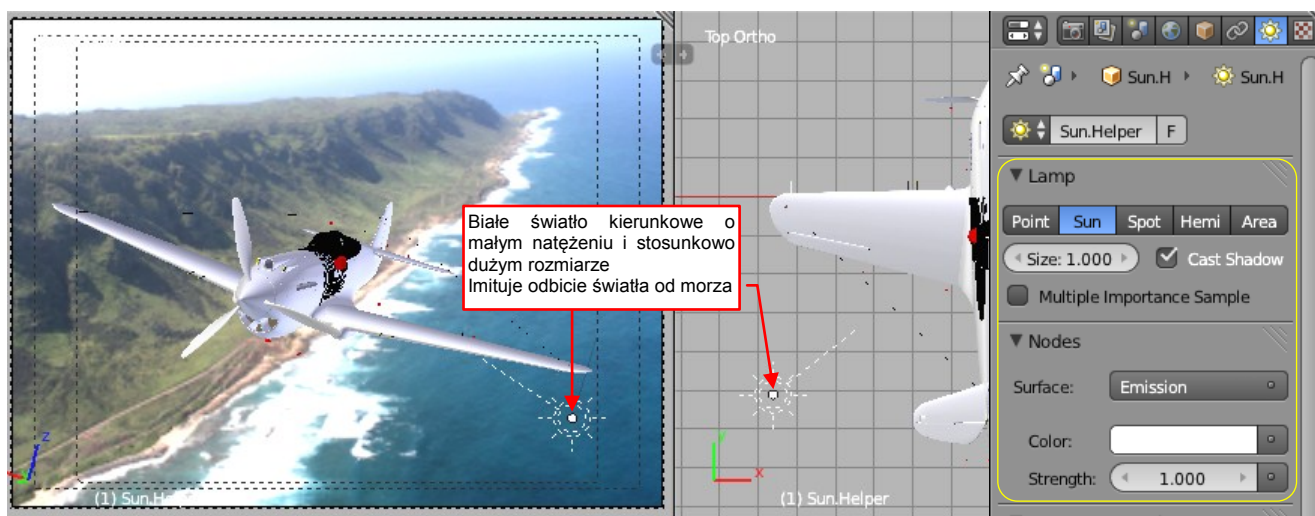
Rysunek 3.1.14 przedstawia uzyskany rezultat:



Rysunek 3.1.14 Rezultat zastosowania obydwu węzłów



Nic tak nie ożywia cieni jak odrobina światła. Cycles nie ma jeszcze (Blender 2.67) prawdziwego światła sferycznego (*Hemi*). Dlatego przybliżyłem wpływ światła odbitego od powierzchni morza za pomocą słabego światła kierunkowego **Sun.Helper** (Rysunek 3.1.15):



Rysunek 3.1.15 Dodatkowe źródło światła

To światło ma włączoną opcję *Cast Shadow*, aby nie rozjaśniało zbyt głębokich cieni w zakamarkach tuneli chłodnicy czy wlotu powietrza. Jednak jednocześnie jego duży rozmiar (*Size* = 1) i niewielkie natężenie sprawia, że cienie rzucane przez **Sun.Helper** są tak rozmyte, że praktycznie niedostrzegalne. Tworzy za to delikatne podświetlenie na zacienionej powierzchni kadłuba (Rysunek 3.1.16):



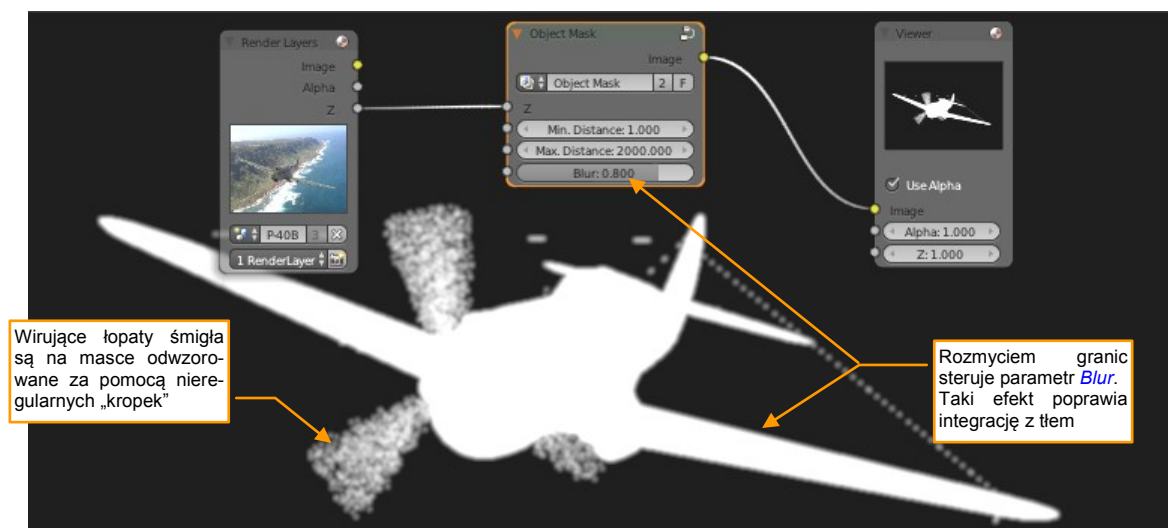
Rysunek 3.1.16 Wpływ dodatkowego źródła światła

Choć początkowo sądziłem, że światło odbite od powierzchni morza powinno być zielonkawe lub błękitne, to po kilku próbach na podglądzie renderu zdecydowałem się ostatecznie na białe. Dzięki zastosowaniu **Sun.Helper** wyraźniej widać podziały blach, zadrapania i zabrudzenia na zacienionych powierzchniach kadłuba.

Na tym etapie zmieniliśmy za pomocą węzła *Color Balance* odcień całego obrazu — zarówno tła jak i samolotu. Niestety, nie mogłem rozjaśnić całości tak bardzo jak bym chciał, ze względu na „przepalania” bieli obrazu (por. str. 442). Czy można wydzielić z obrazu sam samolot, aby chociaż go nieco rozjaśnić?

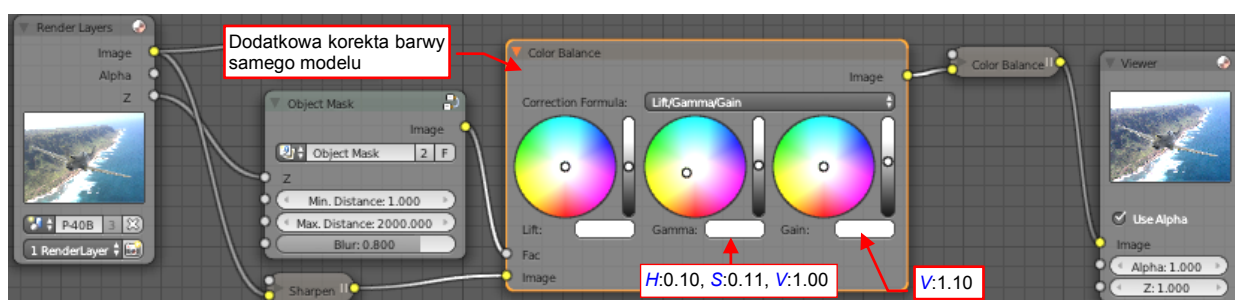


Na szczęście można stworzyć na podstawie współrzędnych **Z** maskę samolotu (Rysunek 3.1.17):



Rysunek 3.1.17 Maska, zbudowana na podstawie współrzędnych **Z** pikseli

Wykorzystuję tu do tego celu gotową grupę o nazwie **Object Mask**. Zasadę działania i szczegóły budowy tego elementu znajdziesz na str. 443. Tę maskę podłączyłem do wejścia **Fac** dodatkowego węzła **Color Balance**. Zmieniam za jego pomocą odcienie i jasność samego samolotu (Rysunek 3.1.18):



Rysunek 3.1.18 Zmodyfikowany schemat

W ten sposób udało się nam znacząco poprawić oryginalny render (Rysunek 3.1.19):

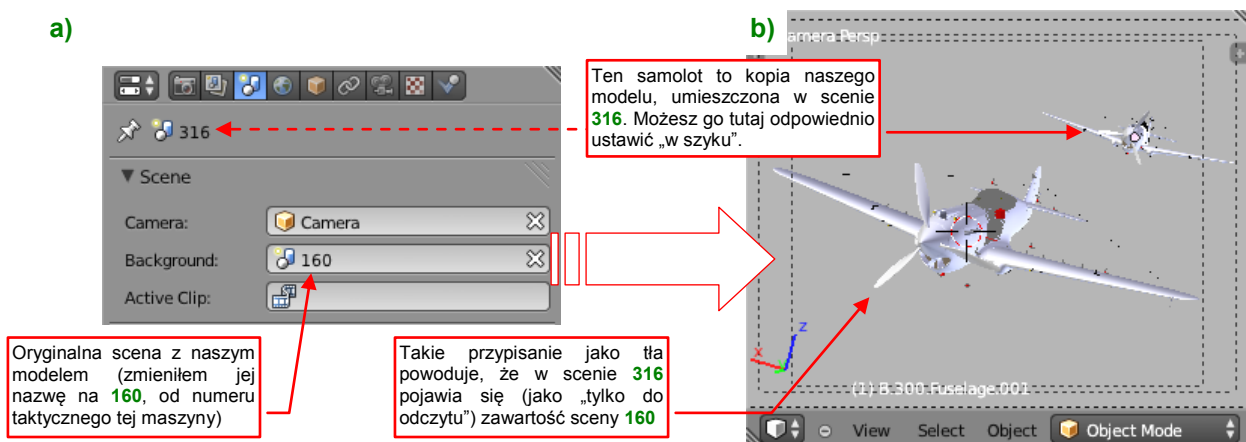


Rysunek 3.1.19 Ostateczny rezultat

Zwróć uwagę, że na obrazie przedstawionym przez Rysunek 3.1.19 samolot ma nieco inny odcień niż na poprzednim renderze. To wynik drobnego przesunięcia ku żółci w kontrolce **Gamma** jego dodatkowego węzła **Color Balance** (por. Rysunek 3.1.18). Ten model znajdziesz w pliku *history\p40\p40B-9.01w3.blend*. Możesz sam poeksperymentować z jego odcieniem, tworząc własną wariację na temat „jak w tym oświetleniu wyglądałby średnio zużyty kolor **Olive Drab**”.

- Możliwość wyodrębnienia samego modelu za pomocą grupy **Object Mask** umożliwia łatwiejszą integrację z tłem: możesz swobodnie zmieniać odcień samego modelu, aby lepiej pasował do otoczenia

Wydaje mi się, że na tej scenie brakuje drugiego samolotu. 7 grudnia 1941 roku wystartowały z lotniska Haleiwa dwie maszyny, o numerach bocznych „160” (to nasz model) i „316”. Zmieniłem więc nazwę podstawowej sceny na **160** i dodałem nową (wybierając opcję **Link Objects**), o nazwie **316**. Potem przekształciłem jej referencje (**links**) w kopie obiektów (**Object → Make Single User → Object**). Wreszcie przypisałem scenę **160** jako „tło” sceny **316** (Rysunek 3.1.20a):



**Rysunek 3.1.20 Dodanie drugiego samolotu (jako drugiej sceny)**

Szczegóły budowy takiej złożonej sceny znajdziesz na str. 213. Rysunek 3.1.21 przedstawia rezultat:

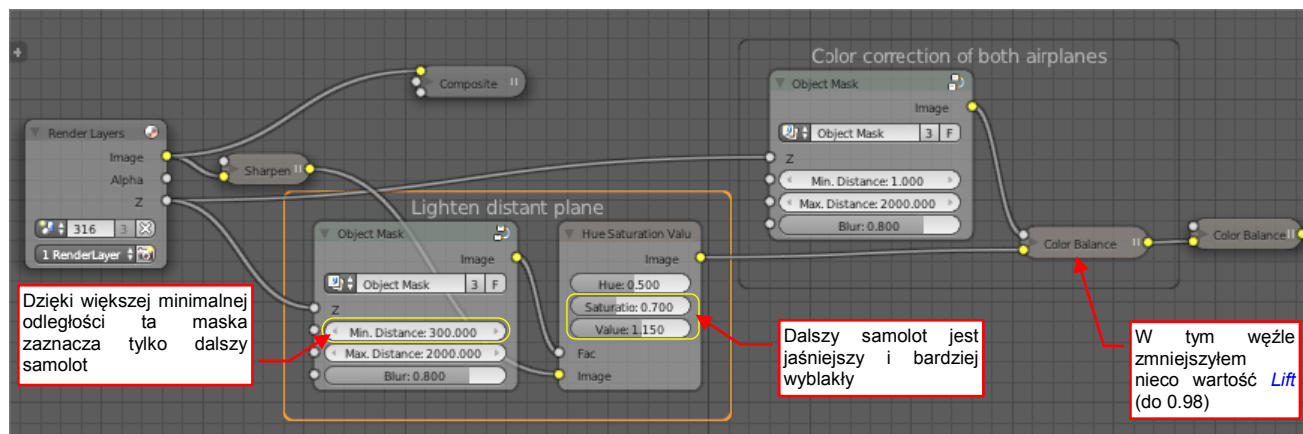


**Rysunek 3.1.21 Scena z dwoma samolotami**



Co prawda Rysunek 3.1.21 tego dobrze nie pokazuje, ale każdy z samolotów ma naprawdę inny numer boczny. Aby ten efekt uzyskać, stworzyłem drugą wersję obrazu oznaczeń taktycznych (*decals-316.png*). Następnie przypisałem kilku kluczowym obiektom maszyny „316” *Pass Index* = 1 i uzależniłem od niego wybór tekstury w materiale **B.Skin.Camouflage**. (Szczegóły — patrz str. 215).

Dalszy samolot „316” wyszedł na „surowym” renderze zbyt ciemno (obiekty bardziej odległe wydają się zawsze jaśniejsze). Dodałem więc do schematu kompozytora dodatkową sekcję, która rozjaśnia tylko ten myśliwiec (Rysunek 3.1.22):



Rysunek 3.1.22 Zmodyfikowany schemat kompozytora

Wykorzystałem w tym celu drugi raz grupę *Object Mask*. Dzięki ustawieniu minimalnego dystansu na 300 jednostek Blendera (por. str. 443), ten węzeł zaznacza tylko dalszy samolot. Kolory obszaru wybranego za pomocą takiej maski rozjaśniłem i zmniejszyłem nasycenie za pomocą węzła *Hue Saturation Value*. Plik z tą sceną towarzyszy tej książce pod nazwą *history\p40\P40B-9.01w4.blend*.

Rysunek 3.1.23 przedstawia inne ujęcie (z innym obrazem tła) sceny z dwoma samolotami. W tym widoku z boku dobrze widać ich różne numery taktyczne, uzyskane dzięki zróżnicowaniu wartości *Pass Index*:



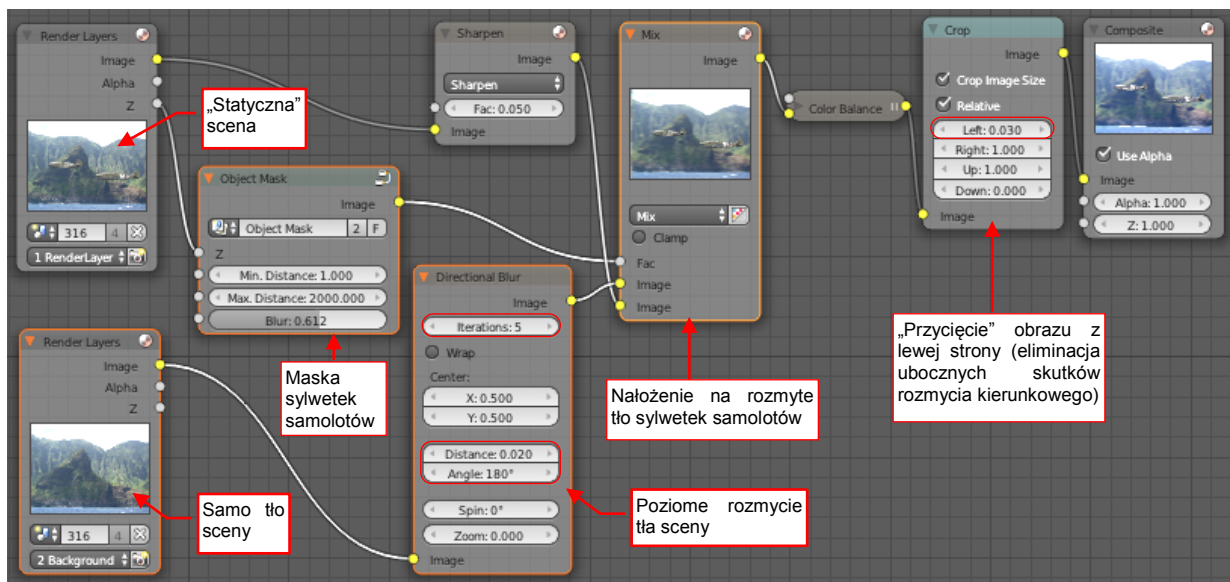
Rysunek 3.1.23 Inna scena z dwoma samolotami

Obraz który przedstawia Rysunek 3.1.23 wydaje się zbyt statyczny: te dwa samoloty są zawieszone nad wodą jak balony. Aby dodać do takiej sceny dynamikę, możesz użyć węzłów *Directional Blur* i *Object Mask* do kierunkowego rozmycia tła, imitującego rozmycie ruchu (Rysunek 3.1.24):



Rysunek 3.1.24 Scena z kierunkowym rozmyciem tła

Aby uzyskać taki efekt, użyłem w schemacie kompozycji drugiej warstwy renderowania (czyli obrazu sceny), na której znajduje samo tło (Rysunek 3.1.25):

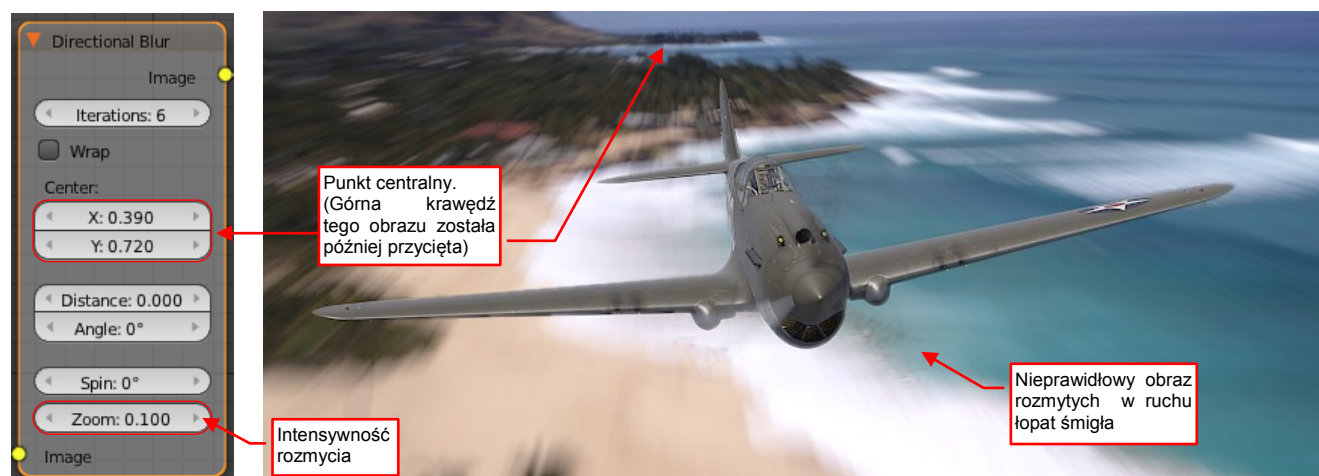


Rysunek 3.1.25 Schemat kompozycji sceny z liniowym rozmyciem tła

Obraz z tego drugiego węzła *RenderLayer* podłączyłem do efektu *Directional Blur*, poddając go rozmyciu w kierunku poziomym. Następnie w węźle *Mix* nałożyłem na to rozmyte tło oryginalny (tzn. ostry) obraz samolotów. (Aby „wyciąć” ich sylwetki ze statycznej sceny, wykorzystałem grupę *Object Mask*, która steruje udziałem obrazów łączonych w węźle *Mix*). Potem dokonuję korekty barw (w węźle *Color Balance*). Na koniec wzdłuż lewej krawędzi obrazu pozostał czarny pasek, który eliminuję „przycinając” obraz węzłem *Crop*. Więcej informacji na temat kierunkowego rozmycia ruchu i tego schematu znajdziesz na str. 446.

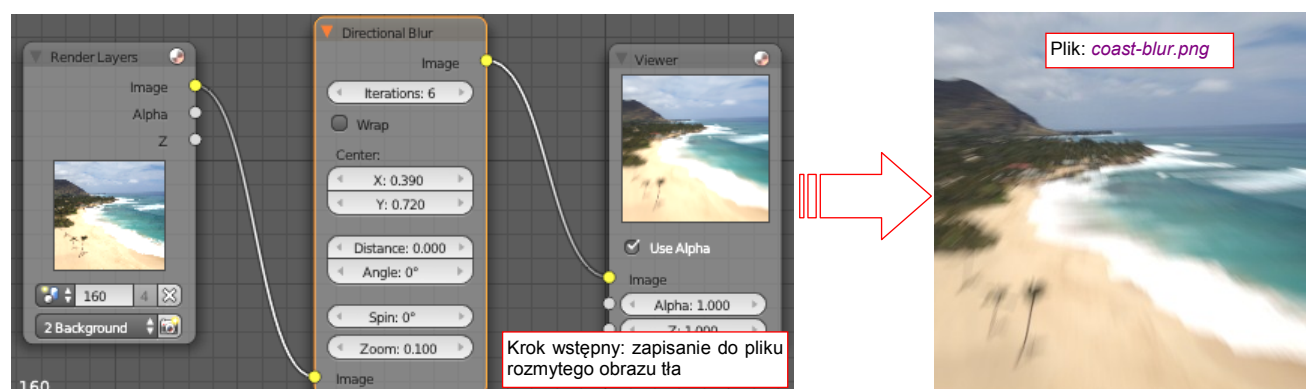


W ujęciach z przodu należy używać koncentrycznego rozmycia ruchu. Rysunek 3.1.26 przedstawia scenę, w której wykorzystuję taki sam schemat kompozycji jak ten pokazywany przez Rysunek 3.1.25. Zmieniłem tylko obraz tła, ustawienie modelu, i parametry węzła *Directional Blur*:



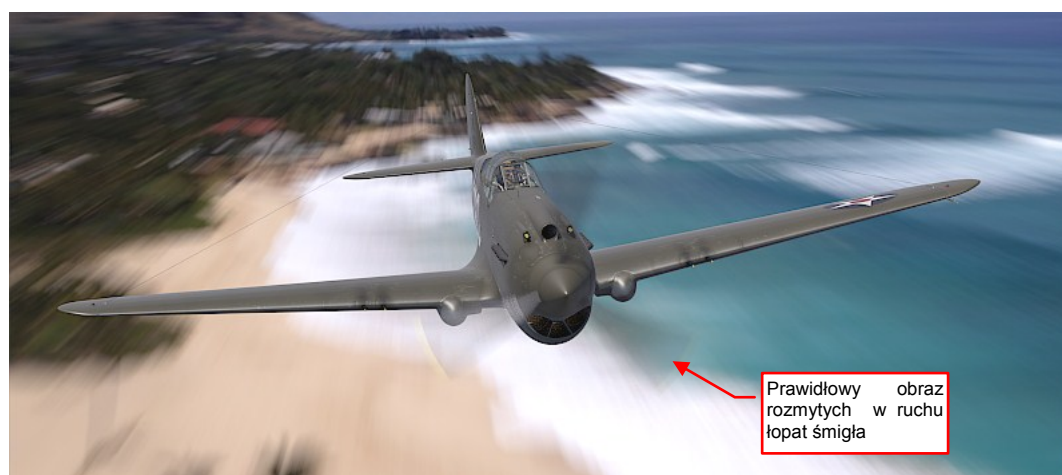
Rysunek 3.1.26 Ujęcie z przodu: zastosowanie rozmycia koncentrycznego

W takich scenach należy zwrócić uwagę, by oś samolotu biegła równolegle do kierunku rozmycia. Rysunek 3.1.26 ujawnia mankament zastosowanego schematu kompozycji: nieprawidłowy obraz wirujących łopat śmigła. Można temu zaradzić wykonując taką scenę w dwóch, a nie jednym kroku. Najpierw wykonaj rozmycie samego tła i zapisz rezultat (jest dostępny jako obraz *Viewer Node*, w *UV/Image Editor*) do pliku (Rysunek 3.1.27):



Rysunek 3.1.27 Zapisanie rozmytego obrazu tła do pliku

Potem powróć do prostego schematu bez rozmycia tła, i użyj pliku z rozmytym obrazem jako zwykłej tekstury otoczenia (por. str. 356). Rezultat przedstawia Rysunek 3.1.28:



Rysunek 3.1.28 Obraz wyrenderowany na uprzednio rozmytym tle (wykorzystuje plik *coast-blur.png*)

Rysunek 3.1.29 przedstawia obraz startu pary myśliwców z lotniska Haleiwa, stworzony w ten sam sposób:



**Rysunek 3.1.29** Inna scena, wykorzystująca to samo tło

W tej scenie nie musiałem sztucznie rozjaśniać dalszej maszyny (być może w wyniku innego położenia słońca). Zwróć uwagę że osie samolotów w kierunku pionowym są tu odchylone od kierunku rozmycia. Jednak każde odchylenie w kierunku poziomym wyglądałoby na błąd! Niesymetryczne składanie kół podwozia (najpierw jedno, potem drugie) było charakterystyczne dla wszystkich P-40.

Plik z tą sceną znajdziesz w materiałach towarzyszących książce ([history\p40\P40B-9.01w6.blend](#)).

## Podsumownie

- W tej sekcji pokazałem, jak stworzyć scenę w locie poprzez wkomponowanie modelu w fotografię. Zdjęcia, których do tego używam, znajduję w Internecie. Korzystam z wyszukiwarek, by znaleźć dostępne fotografie o odpowiedniej rozdzielczości. Na przykład w Google możesz ustawić min. rozmiar obrazu w kryteriach wyszukiwania. Oczywiście, ta metoda jest dopuszczalna, dopóki tworzysz swoje prace hobbystycznie, tylko dla własnej satysfakcji;
- „Pozowanie” samolotu odbywa się poprzez przesunięcia kamery, albo przesunięcie lub obrót jej „celu” (obiektu **Target**). Dużą pomocą jest ustawienie fotografii tła w oknie widoku z kamery (jako **Background Image** — str. 129);
- Po ustaleniu pozycji i orientacji kamery, trzeba ustalić pozycję i energię światła. Chodzi o odtworzenie warunków oświetlenia widocznych na fotografii (str. 129). Oprócz podstawowego źródła światła — **Sun** — możesz wykorzystać jakieś słabe oświetlenie pomocnicze (str. 135). Za pomocą takiego światła można uzyskać na powierzchni modelu dodatkowe odbłaski i rozjaśnienia. Często przydaje się także zmienić trochę barwę otoczenia (str. 131) (Można ją także określić jako „kolor cieni”);
- Efekt „rozmycia ruchu” łopata śmigła jest kanonem, który powstał i utrwalił się dzięki fotografii. Nie sposób się od niego uwolnić. Aby go uzyskać, musisz najpierw animować obrót zespołu śmigła (str. 130). Typowa „prędkość” tego ruchu (odpowiednia dla przesłony **Motion Blur:Shutter** = 100) to 1000-1200 klatek na pełen obrót (str. 130).
- Dla lepszego dopasowania modelu do zdjęcia tła przydatny jest zaprojektowanie dodatkowych modyfikacji obrazu po wyrenderowaniu. Ten ostatni etap określa się w CG jako **postprocessing**. W Blenderze możesz go zrealizować za pomocą schematu kompozycji, który tworzysz w **Node Editor**, używając **Composite Nodes** (str. 134). To bardzo elastyczne narzędzie o dużych możliwościach. Pozwala na łączenie ze sobą wielu różnych obrazów i poddawaniu ich różnorodnym transformacjom;
- Schemat kompozycji opisuje transformację, której zostanie poddany każdy piksel wyrenderowanego obrazu. Oczywiście, każdy węzeł tego schematu może zmienić tylko niektóre fragmenty z tej całości;
- Typowym węzłem, dostarczającym schematom kompozycji Blendera danych wejściowych jest **RenderLayer**. Reprezentuje na schemacie pojedynczą „warstwę renderowania” (**render layer**). Możesz zdefiniować wiele alternatywnych warstw renderowania dla pojedynczej sceny. Każda z nich to indywidualny zestaw parametrów dla renderera. Opisuje, co ma być przetworzone, oraz określa rodzaje danych udostępnianych na wyjściu węzła (str. 134, 136, 139). Domyślnie każda scena Blendera ma jedną taką warstwę, o nazwie **1 Render Layer**;
- Typową metodą na lepszą integrację modelu ze zdjęciem tła jest zmiana ich odcieni po wyrenderowaniu. Możesz to zrobić za pomocą węzła **Color Balance** (str. 134);
- Podobnie jak w schematach materiałów, często używane kombinacje węzłów kompozycji można łączyć w grupy. Upraszcza to budowę schematów i poprawia ich czytelność. W tej sekcji wykorzystuję grupę o nazwie **Object Mask**. Na podstawie informacji z kanału **Z** tworzy obraz, w którym białe kontury modelu są umieszczone na tle czarnego tła (str. 136). To przydatna maska do różnych efektów (np. dostosowania odcienia samolotu do tła — str. 136, albo rozmycia tła za samolotem — str. 139);
- Istnieje wiele metod stworzenia sceny z kilkoma samolotami. Na str. 137 pokazuję, jak uzyskać obraz pary P-40. Używam do tego scen Blendera (po jednej na każdy samolot);
- Aby zróżnicować oznaczenia (np. numery taktyczne) na każdym z modeli użytych w scenie grupowej, trzeba przygotować dwie wersje tekstury oznaczeń (**decals.png**). Potem zmodyfikuj materiał poszycia (**B.Skin.Camouflage**), uzależniając wybór piksela z tych tekstur od wartości **Pass Index** renderowanego obiektu (por. str. 215);
- Efekt kierunkowego rozmycia ruchu nadaje dynamiki scenie — szczególnie ujęciom robionym na bardzo małej wysokości. Uzyskuje się go za pomocą węzła **Directional Blur** (str. 139). Czasami można stosować ten efekt w dwóch krokach: stworzenia rozmytego tła i wyrenderowania na nim samolotu (str. 140);



### 3.2 Realistyczna scena na ziemi

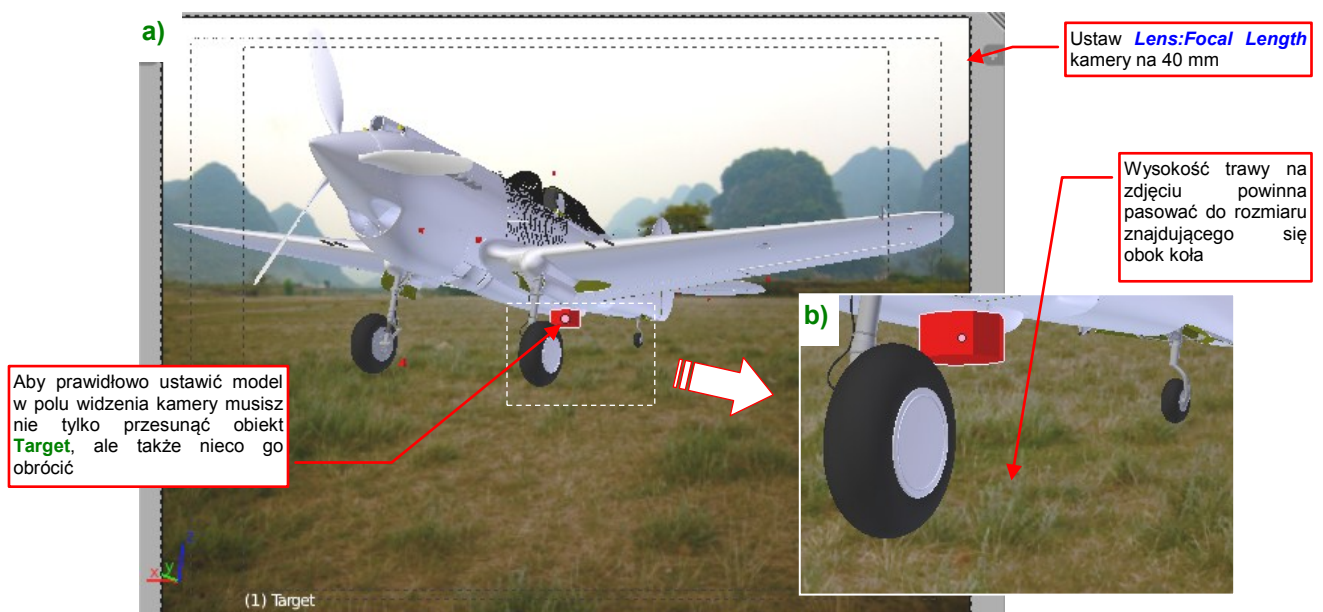
Choć może wydawać się to dziwne, trudno jest znaleźć duże, dobre zdjęcie lotniska polowego, które nadaje się na tło sceny. Szczególnie, jeżeli szukamy fotografii z określonej okolicy, z której operował nasz model. W tej sekcji „postawimy” na lotnisku polowym P-40 („białą 75”) z AVG. Szukałem przez parę dni dużych obrazów z prowincji Yunnan (Chiny), ale niczego odpowiedniego nie znalazłem. No cóż, gdy czegoś brakuje, zawsze możesz spróbować to stworzyć sam. Przeglądając posiadane zdjęcia, zdecydowałem się połączyć kawał równego jak stół stepu w Mongolii z brzegiem rzeki Jangcy (Rysunek 3.2.1):



Rysunek 3.2.1 Fotomontaż obrazu fragmentu lotniska w Kweilin

Step ze zdjęcia **a**) przypomina powierzchnię lotniska Kweilin widoczną na archiwalnych zdjęciach: jest płaski, porośnięty kępkami rzadkiej trawy. Wzgórza i brzeg rzeki ze zdjęcia **b**) wyglądają podobnie do otoczenia tego lotniska (z lat 40-tych XX w.). Obydwa zdjęcia wydają się mieć zbliżony odcień. Użyłem GIMP aby je ze sobą złożyć. Zasłoniłem obrazem stepu powierzchnię wody ze zdjęcia **b**). Potem, używając narzędzia *Color Balance* zmieniłem nieco odcień stepu, by go lepiej dopasować do odcienia chińskiego lessu widocznego na brzegu rzeki. Rezultat (Rysunek 3.2.1c) znajdziesz w pliku *model/p40/background/kweilin.png*. To, że uzyskany obraz jest dość ciemny, nie stanowi dużej przeszkody. Te barwy są gdzieś w połowie skali pomiędzy czernią i bielą, więc można je zawsze stosunkowo łatwo rozjaśnić.

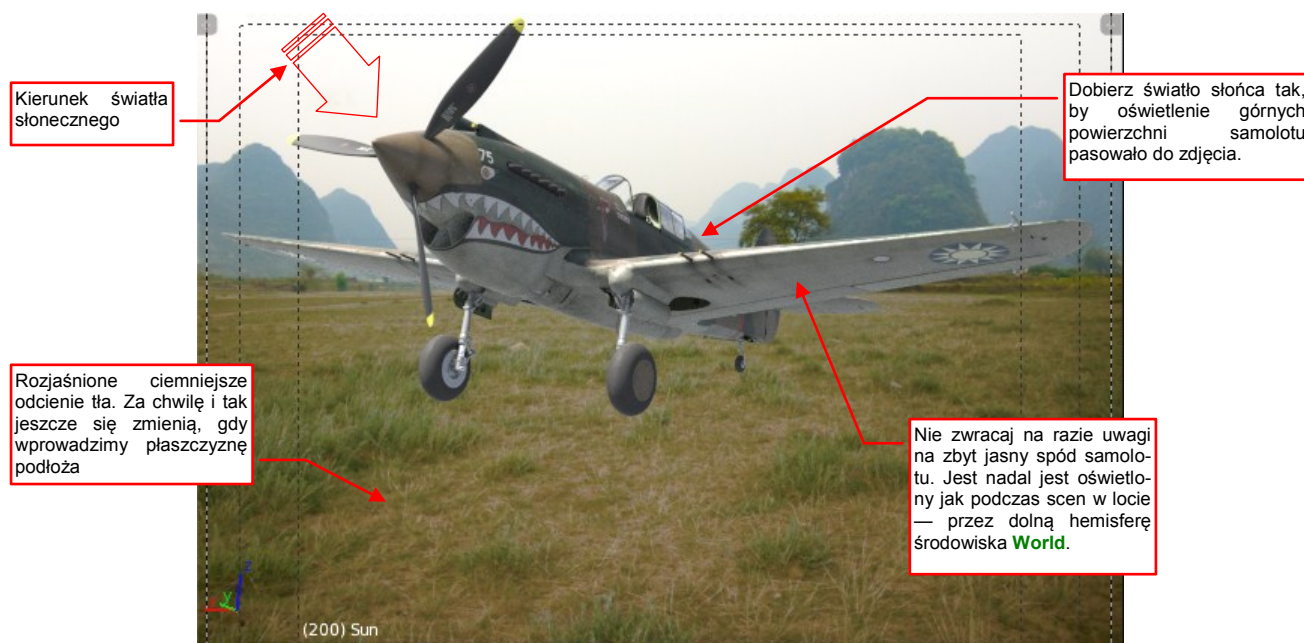
Podstaw to zdjęcie jako tło sceny (Rysunek 3.2.2a). Kamerę ustaw tak, by koła modelu „stanęły” na gruncie tam, gdzie pasują do rozmiaru trawy widocznej na tle (Rysunek 3.2.2b). To wymaga pochylenia „celu” kamery — obiektu **Target**. (Upewnij się, że w jej ograniczeniu *Track To* jest włączony przełącznik *Target Z*).



Rysunek 3.2.2 Ustawienie kamery i modelu

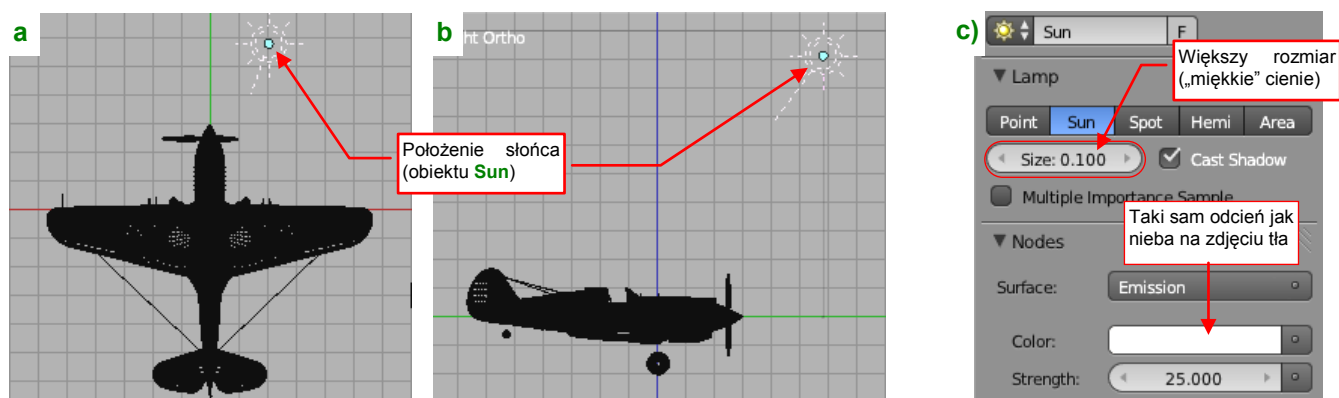


Do tej pory ustawialiśmy oświetlenie dla sceny w locie (por. Wyciąg III). W scenie na ziemi trzeba je zaaranżować nieco inaczej. Rysunek 3.2.3 przedstawia pierwsze przybliżenie:



Rysunek 3.2.3 Pierwsze przybliżenie oświetlenia sceny

Na zdjęciu tła trudno jednoznacznie ocenić kierunek promieni słonecznych, gdyż cienie na wzgórzach są odległe i zamglone. Wydaje się padać od strony obserwatora (Rysunek 3.2.4b). Ostatecznie zdecydowałem że znajduje się po lewej (Rysunek 3.2.3, Rysunek 3.2.4a):



Rysunek 3.2.4 Szczegóły ustawień słońca

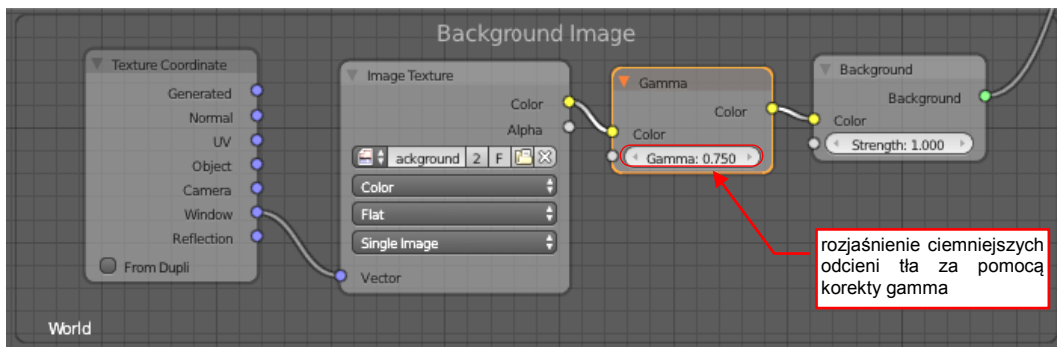
Otoczenie na zdjęciu sprawia wrażenie lekko zamglonego — stąd zdecydowałem się zmienić rozmiar słońca (**Sun**) z 0.01 do 0.1 (Rysunek 3.2.4c), by uzyskać „miękkie” krawędzie cieni. Z tego samego powodu nadałem słońcu taki kolor, jaki ma niebo (na rozjaśnionej wersji zdjęcia).

Zdecydowałem się także rozjaśnić odcień powierzchni samolotu. Zrobiłem to, zmieniając w materiale **B.Skin.Camouflage** bazową wartość parametru **X.Textured Skin:Diffuse Intensity** do 0.3 (Rysunek 3.2.5):



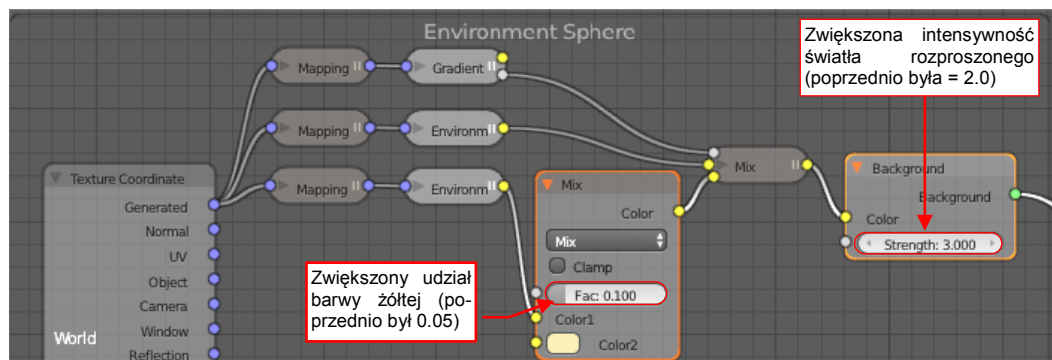
Rysunek 3.2.5 Rozjaśnienie powierzchni samolotu (B.Skin.Camouflage)

Scena przedstawiana przez Rysunek 3.2.3 wykorzystuje to samo otoczenie **World**, jakie stworzyliśmy w poprzednich rozdziałach (takie, jakie jest opisane na str. 350 i następnych). Podstawiłem tylko jako zawartość obrazu **background** nasz fotomontaż. Dodatkowo rozjaśniłem go za pomocą korekty **Color→Gamma** (aby nie „przepalić” tego jasnego nieba w zupełną biel — Rysunek 3.2.6):



Rysunek 3.2.6 Rozjaśnienie obrazu tła (środowisko **World**)

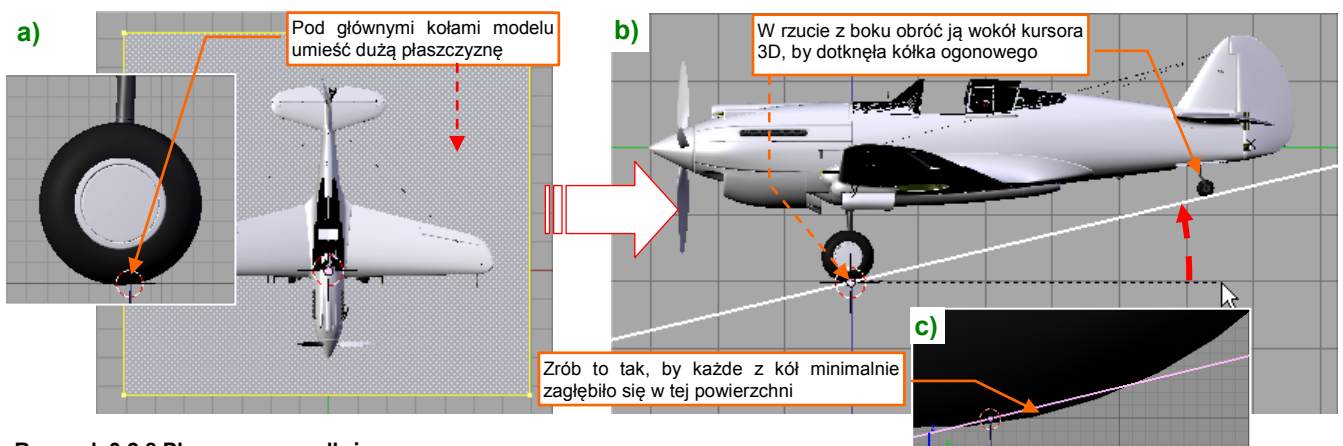
Oprócz tego zdecydowałem się rozjaśnić i zmienić zabarwienie światła rozproszonego. (Tym bardziej, że większość powierzchni samolotu w tym ujęciu widzimy od strony nie oświetlonej przez słońce). Stąd zwiększyłem jego intensywność z 2.0 do 3.0, i zwiększyłem w nim udział koloru żółtego (Rysunek 3.2.7):



Rysunek 3.2.7 Rozjaśnienie światła rozproszonego (środowisko **World**)

Bez tych zmiany render przedstawiony przez Rysunek 3.2.3 byłby zbyt ciemny.

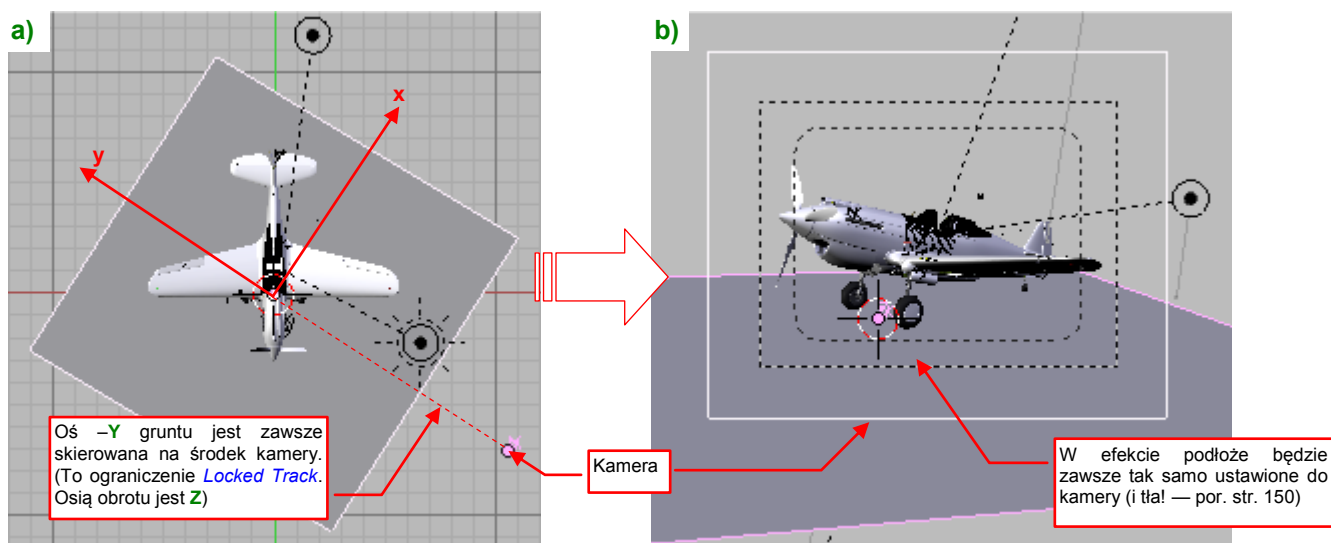
„Wierzch” samolotu już całkiem dobrze pasuje do tła. Teraz musimy zająć się jego spodem, a dokładniej: stworzyć na zdjęciu cień naszego modelu. Umieść kursor 3D w środku samolotu, ale na poziomie kół (Rysunek 3.2.8a). Wstaw w tym miejscu duży kwadrat (**Plane**) — to będzie nasz „grunt”:



Rysunek 3.2.8 Płaszczyzna podłoża

Nadaj temu obiektowi nazwę **Ground**, i umieść na warstwie 19. Podstaw go, w rzucie z góry, pod koła główne. Potem przełącz się na rzut z boku, i obróć tę płaszczyznę tak, by dotknęła kółka ogonowego (Rysunek 3.2.8b). Upewnij się, że wszystkie koła minimalnie przenikają przez jej powierzchnię (Rysunek 3.2.8c).

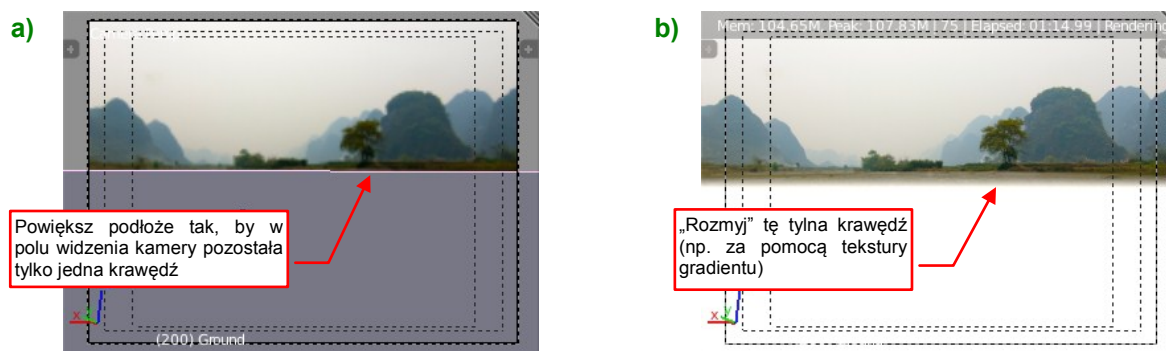
Przy okazji możesz naszemu „gruntowi” przypisać ograniczenie **Locked Track**, kierujące na kamerę oś **-Y** tej płaszczyzny (Rysunek 3.2.9a):



Rysunek 3.2.9 Skierowanie podłoża na kamerę (za pomocą ograniczenia **Locked Track**)

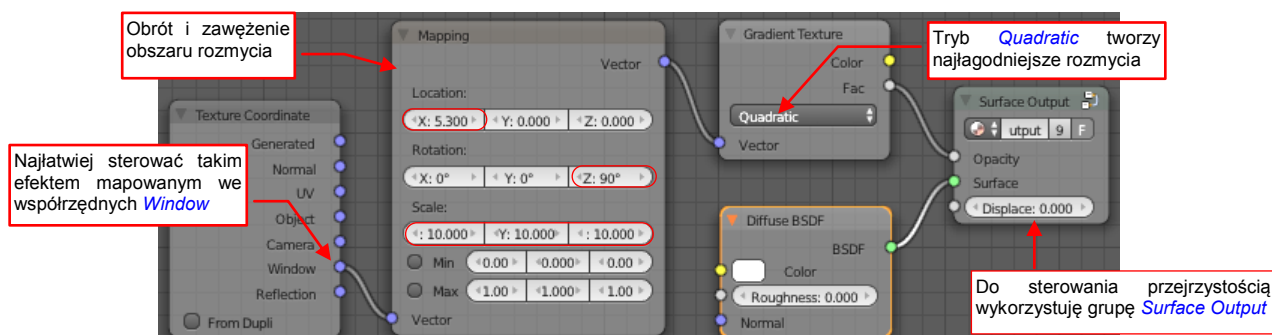
Oś obrotu tego ograniczenia jest lokalna oś **Z** podłoża. Takie ustawienie pozwala na obracanie modelu w kamerze (a właściwie — kamery wokół modelu) bez zmiany tła (por. str. 150, Rysunek 3.2.21). To bardzo ułatwia późniejsze poprawki sceny.

Powierzchnię **Ground** można jeszcze bardziej powiększyć, by w polu widzenia kamery było widać tylko jej jedną, przeciwną krawędź (Rysunek 3.2.10a). Przypiszmy do tej siatki materiał — także o nazwie **Ground**. Używając tekstury gradientu „rozmyj” tylną krawędź powierzchni podłoża (Rysunek 3.2.10b) :



Rysunek 3.2.10 Dopasowanie tylnej krawędzi podłoża do tła

Rysunek 3.2.11 przedstawia początkowy schemat materiału **Ground**, na którym widać szczegóły mapowania tekstury gradientu. Zwróć uwagę, że jest mapowania we współrzędnych okna (**Window**):

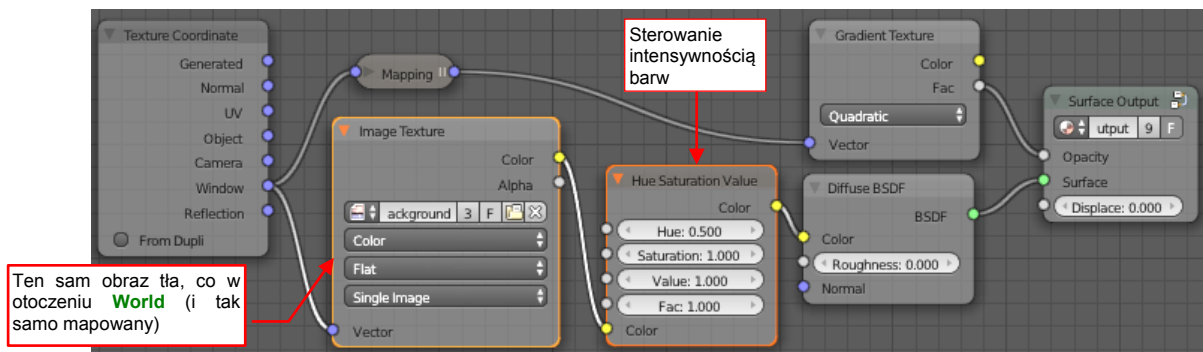


Rysunek 3.2.11 Materiał podłoża (**Ground**) — schemat początkowy

Aby lepiej ocenić efekt rozmycia, początkowo pozostawiłem powierzchnię podłoża w kolorze białym.

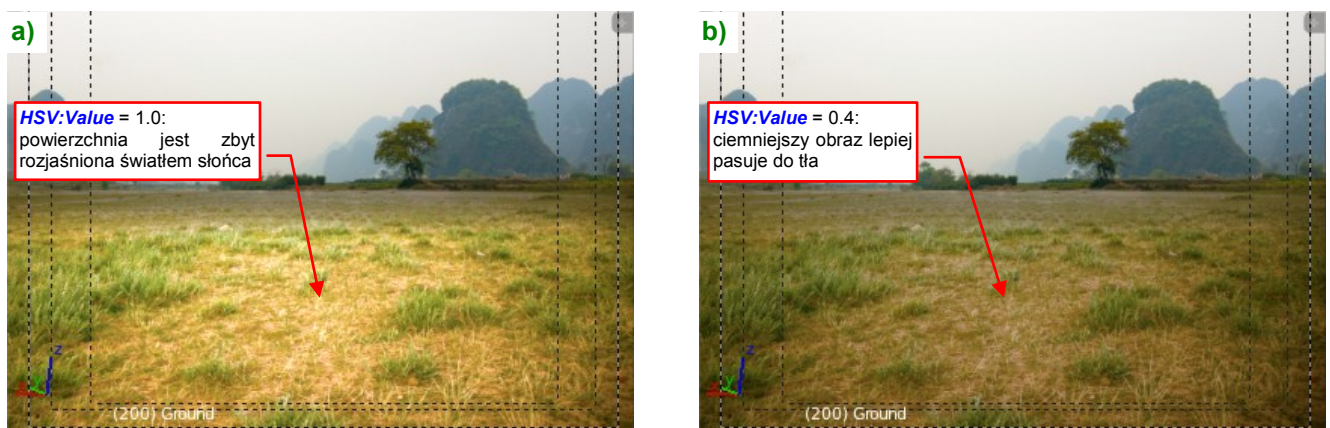


Teraz zmodyfikujmy kolor płaszczyzny: przypisz do shadera *Diffuse BSDF* obraz tła, w takim samym mapowaniu jak w otoczeniu *World: Window* (Rysunek 3.2.12):



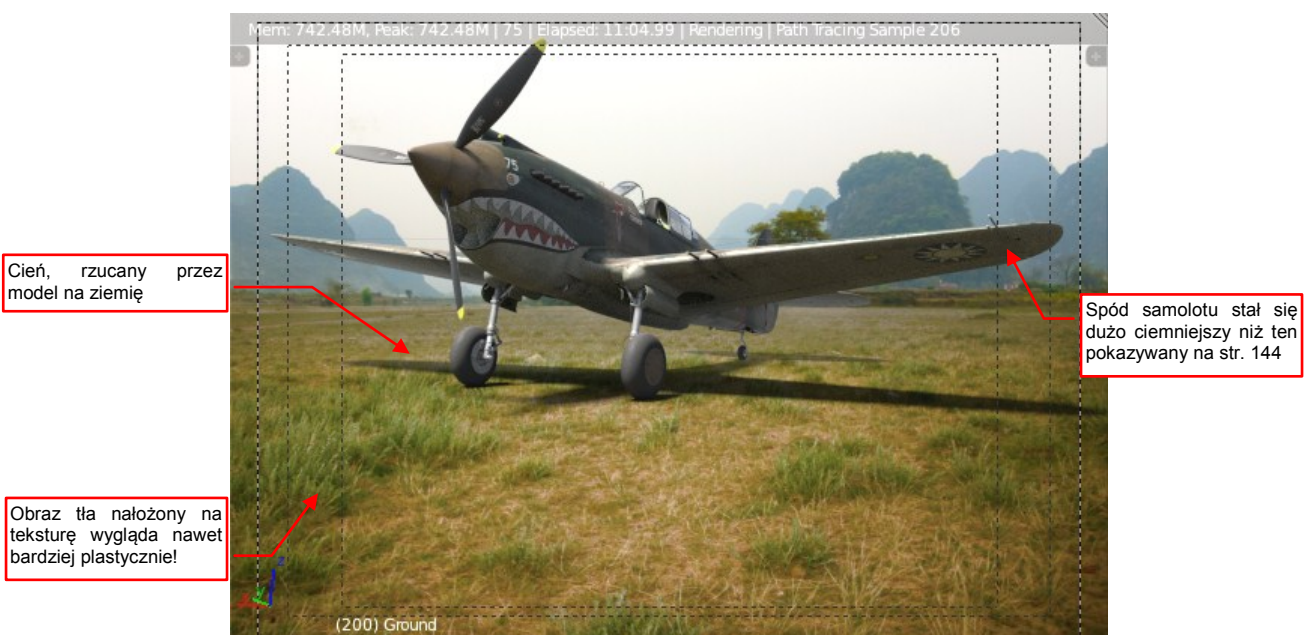
Rysunek 3.2.12 Materiał *Ground*: dodanie obrazu podłoża

Na wszelki wypadek za węzłem *Image Texture* z obrazem *background* dodałem węzeł *HSV* do sterowania intensywnością barw. Będziemy go potrzebować (Rysunek 3.2.13):



Rysunek 3.2.13 Mapowanie tekstury tła na podłoże

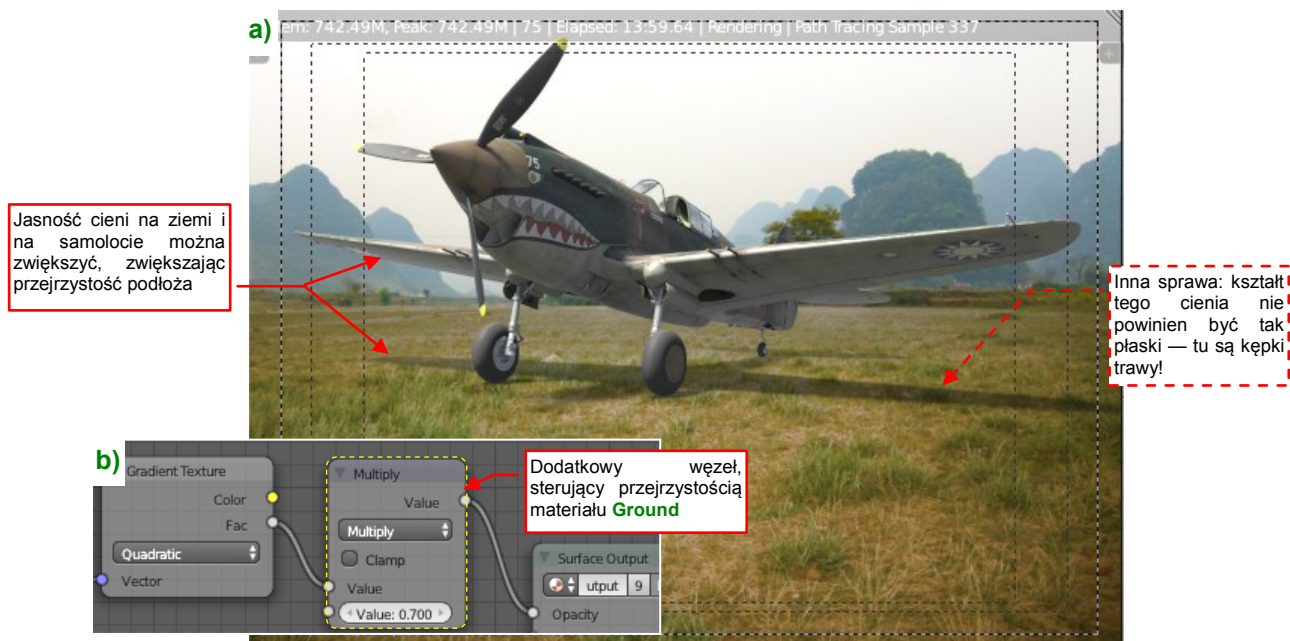
Gdy zastosujesz materiał *Ground* z ustawieniami pokazanymi przez Rysunek 3.2.12, uzyskasz zbyt jasną powierzchnię (Rysunek 3.2.13a). Dopiero zmniejszenie intensywności barw tekstury do 0.4 sprawia, że powierzchnia podłoża poprawnie łączy się z tłem (Rysunek 3.2.13b). W ten sposób uzyskaliśmy rzecz kluczową do integracji ze zdjęciem: cień (Rysunek 3.2.14):



Rysunek 3.2.14 Cień, rzucany przez model na zdjęcie tła

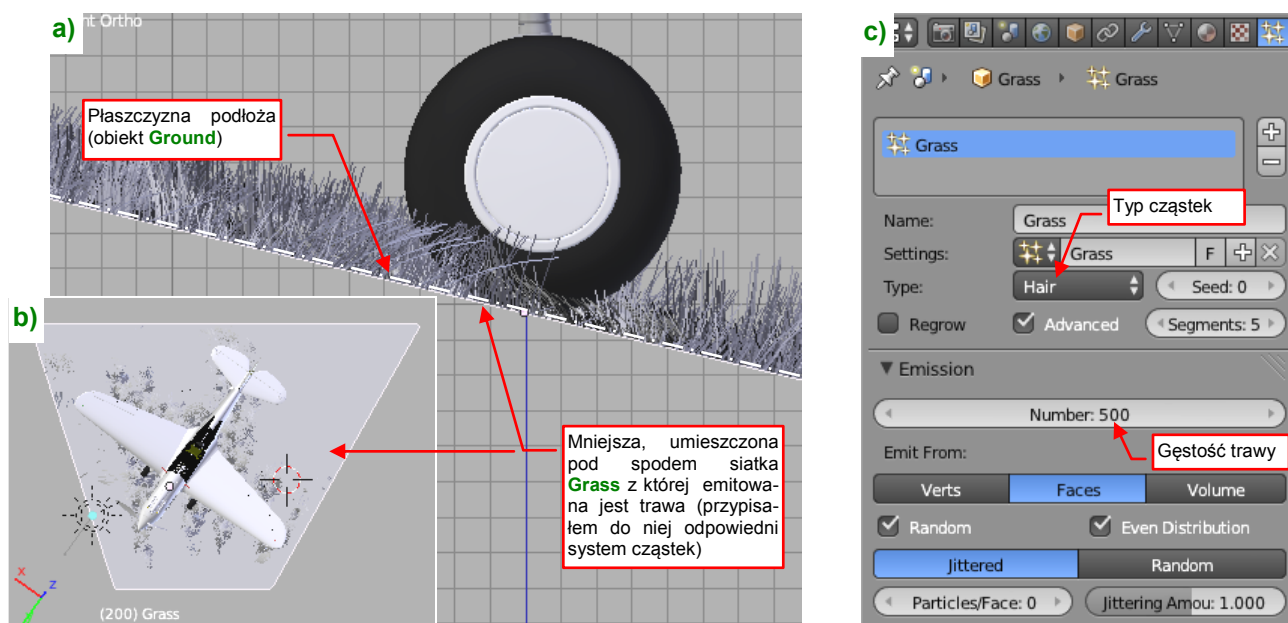


Co prawda cień na poprzedniej ilustracji wydaje się zbyt głęboki. Na szczęście to można bardzo łatwo zmienić. Wystarczy dodać do schematu materiału **Ground** za teksturą gradientu dodatkowy węzeł **Multiply** (Rysunek 3.2.15b). Ten węzeł zmniejsza jasność gradientu, a w efekcie — zwiększa przejrzystość powierzchni podłoża. Jak pamiętasz pod podłożem znajduje się nadal półsfera która emituje światło rozproszone (por. Rysunek 3.2.3). Stąd cień samolotu i jego dolne powierzchnie stają się jaśniejsze (Rysunek 3.2.15a):



Rysunek 3.2.15 Sterowanie intensywnością cieni

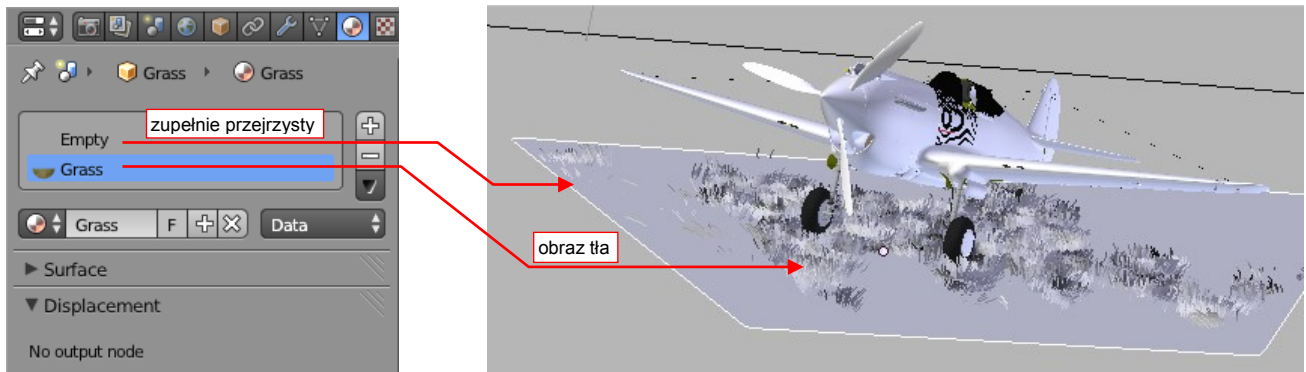
Uzyskaliśmy więc teraz jasny (może nawet nieco zbyt jasny?) cień. Jednak widzę inny problem: jest zbyt płaski (Rysunek 3.2.15a)! Na zdjęciu tła widać kępki trawy — ich obecność powinna deformować granice cienia. Aby uzyskać ten efekt ustawiłem tuż pod siatką podłoża drugą, z której wyemitowałem trawę (Rysunek 3.2.16a,b):



Rysunek 3.2.16 Podłoże z dodaną w okolicach cienia trawą

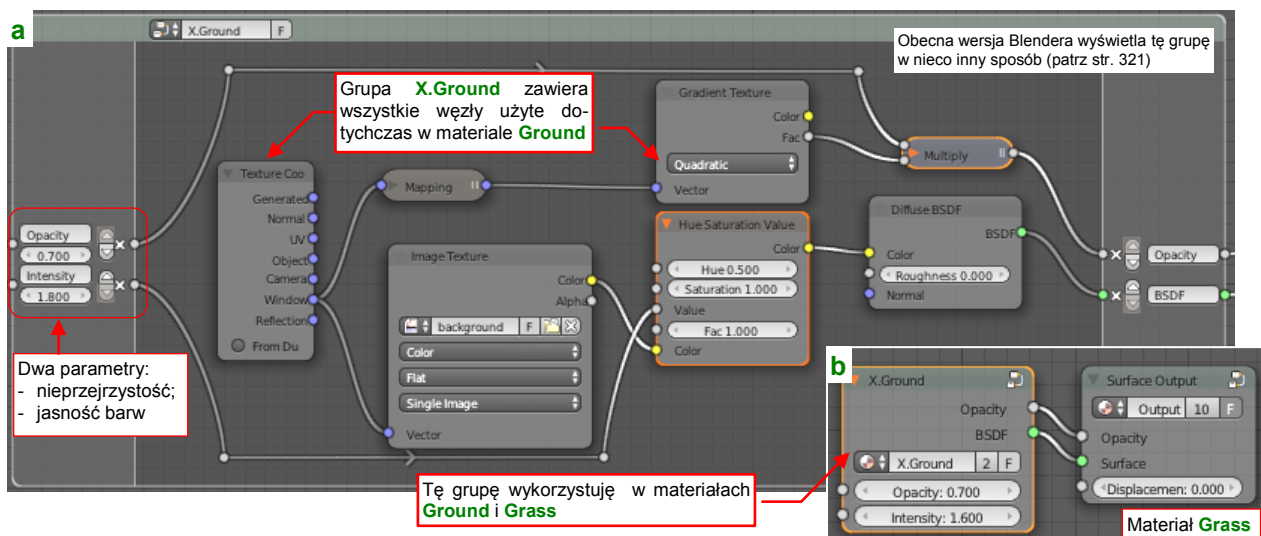
Taki efekt uzyskuję za pomocą systemu cząstek (**Particles**) typu **Hair** (Rysunek 3.2.16c). Wykorzystuję tu możliwość generowania włosów/trawy przez Cycles. Ta funkcjonalność w momencie pisania tej książki była zakwalifikowana jako „eksperymentalna”. (W Blenderze 2.67 musiałem przełączyć **Render:Feature Set** na **Experimental**). W wersji 2.69 odpowiednie panele (**Cycles Hair Rendering**, **Cycles Hair Settings**) z zestawu **Particles** są w standardzie. Szczegółowy opis tworzenia takiej trawy za pomocą systemu cząstek znajdziesz na str. 452.

Rysunek 3.2.17 przedstawia materiały przypisane do emitera (obiektu **Grass**):



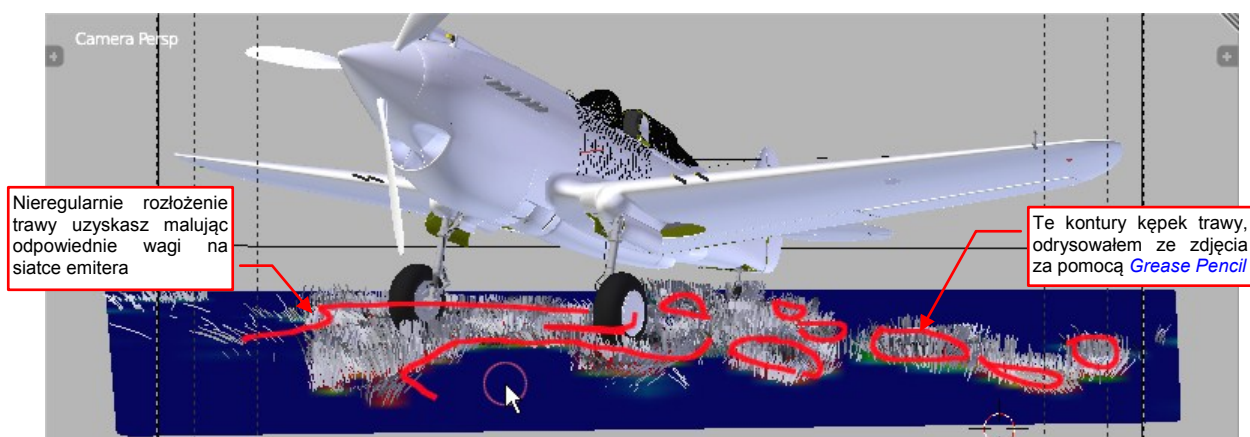
Rysunek 3.2.17 Materiały przypisane do obiektu **Grass**

Do powierzchni tej siatki przypisałem zupełnie przejrzysty materiał **Empty** (aby nie miał wpływu na jasność cieni — por. Rysunek 3.2.15). Żdźbła trawy pokryte takimi samymi teksturami co podłoże **Ground**, ale z nałożonymi z inną intensywnością. Nazwałem ten materiał **Grass**. Aby obraz tła był nałożony na trawę i podłoże dokładnie w ten sam sposób, połączyłem wszystkie węzły oryginalnego materiału **Ground** w pomocniczą grupę **X.Green** (Rysunek 3.2.18a). Wykorzystałem ją następnie w obydwu materiałach (Rysunek 3.2.18b):



Rysunek 3.2.18 Pomocnicza grupa **X.Green** i jej użycie w materiale **Grass**

Gęstością trawy można sterować za pomocą wag. Namaluj je na wierzchołkach siatki emitera w trybie **Weight Paint** (Rysunek 3.2.19, por. także str. 291):



Rysunek 3.2.19 „Malowanie” trawy

Kontury kępek trawy odrysowałem wcześniej ze zdjęcia tła wykorzystując **Grease Pencil** (por. str. 421).

Rysunek 3.2.20 przedstawia nasz model stojący na powierzchni wzbogaconej o efekt trawy. Zmodyfikowałem tu także materiał opon (por. str. 407), aby wyglądały na bardziej zużyte i pobrudzone:

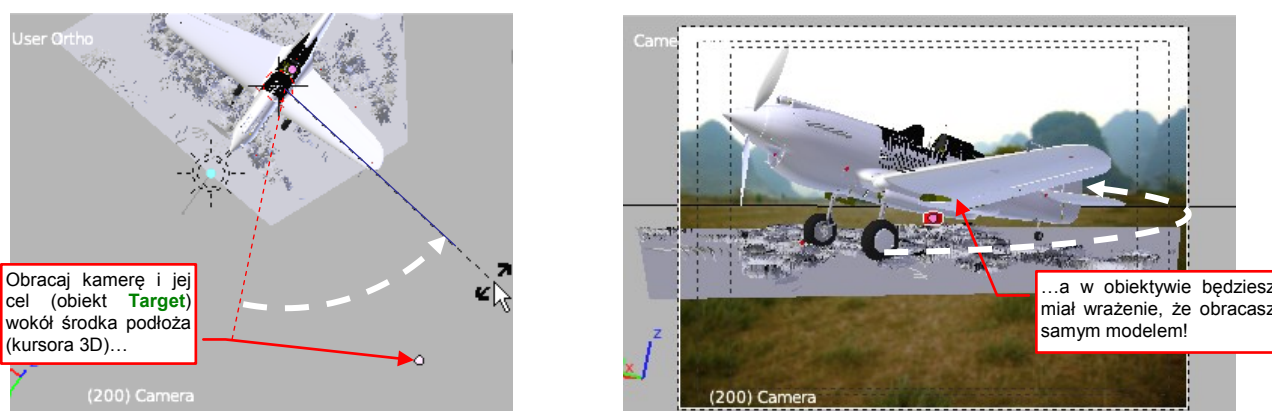


**Rysunek 3.2.20** Cień samolotu dopasowany do trawiastej powierzchni lotniska ze zdjęcia

Na pewno niejednen z grafików komputerowych wszystkie takie poprawki cienia woli po prostu domalować na ostatecznym renderze. Można to zrobić w dowolnym programie do grafiki rastrowej (Photoshop, GIMP...). Ja jednak uważam, że lepiej przygotować podłoże w przestrzeni 3D, w oryginalnej scenie Blendera. Później możesz w prosty sposób poprawić pozycję kamery, czy też wykonać inne zmiany w układzie sceny.

Zobacz chociażby, jak łatwo jest teraz zmienić kompozycję obrazu. Załóżmy, że chcesz ustawić samolot pod innym kątem, a tło i podłoże mają pozostać nieruchome. Pamiętasz, jak przypisaliśmy „gruntowi” ograniczenie *Locked Track*, śledzące położenie kamery (por. str. 146)? To właśnie dla szybkiego wykonania takich zmian!

Przypisz jeszcze tylko do „gruntu” wszystkie światła (relacją *Parent*). W ten sposób zapewnimy sobie ich niezmienną pozycję<sup>1</sup>. Teraz ustaw płaszczyznę widoku równoległą do podłoża (*View→Align View→Align View to Selected*). Umieść w jej środku kursor 3D (*Object→Snap→Cursor to Selection*). Ustaw oś obrotu (*Pivot*) na *3D Cursor*. Następnie zaznacz kamerę i jej cel (obiekt *Target*). Zacznij je obracać (Rysunek 3.2.21a):



**Rysunek 3.2.21** Obracanie kamery wraz z podłożem

<sup>1</sup> Tylko nie zapomnij przed przypisaniem światła *Sun* zmniejszyć wpływu (*Influence*) ich ograniczeń *TrackTo* do zera. (Obiekty z aktywnymi ograniczeniami „źle znoszą zmianę rodzica” — ulegają wtedy dziwnym przesunięciom i obrotom). Po przypisaniu do podłoża z powrotem włącz to ograniczenie, kierujące Słońce na obiekt *Target*.



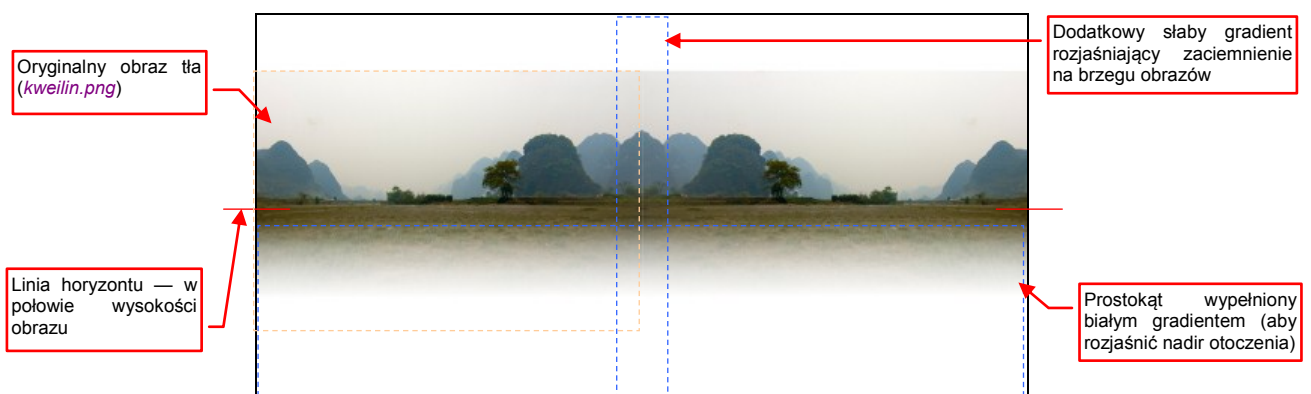
Jednocześnie obserwuj w drugim oknie, co się dzieje w obiektywie kamery. Choć obracasz tylko kamerę, to tam wygląda to wszystko tak, jak gdybyś obracał sam model! Reszta „sceny” — podłoże, oświetlenie i tło — jest w kamerze nieruchoma. I właśnie o taki efekt chodziło! W ten sposób możesz zmienić ujęcie modelu, nie tracąc czasu na dopasowania trawy i podłoża do nieruchomej fotografii tła. Zobacz sam: jednym ruchem ustawiłeś nasz samolot bokiem do kamery (Rysunek 3.2.22):



Rysunek 3.2.22 Ujęcie z boku

Grafik, który poprawia render w Photoshopie, musiałby w takim przypadku usiąść, i ponownie poprawiać „surowy” cień samolotu. A my mamy to gotowe! Teraz można byłoby po lewej wstawić drugi samolot, a wokół niego — figurki mechaników...

W tym nowym ujęciu (Rysunek 3.2.22) zmieniłem nieco odcień tekstury podłoża i trawy (w grupie **X.Ground**) — tak, aby lepiej pasował do odcienia samolotu. W GIMP na obrazu kamuflażu (*color.png* — por. Wyciąg III) nałożyłem w trybie *Multiply* dwie dodatkowe warstwy *Ref-shadows.png* (to kopia *ref\_details.png*) i *Ref-Smoke*. Zwiększyłem w ten sposób wrażenie kurzu i zużycia. Choć na tak „szorstkim” modelu to niewiele zmienia, wstawiłem w środowisku **World** panoramę otoczenia odpowiadającą zdjęciu tła. Rysunek 3.2.23 pokazuje zawartość nowego pliku (*kweilin.jpg*), który podstawiłem w Blenderze jako obraz dolnej półsfery otoczenia (**Earth**):

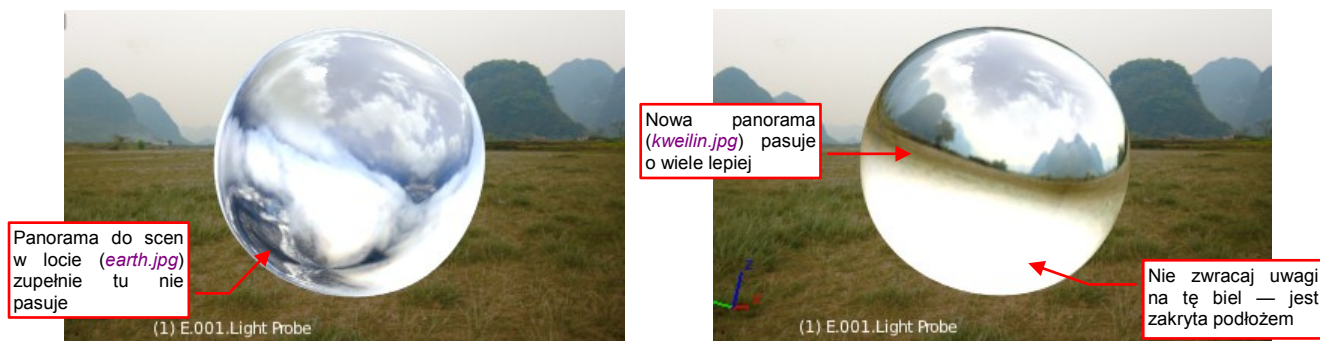


Rysunek 3.2.23 Panorama otoczenia dla środowiska **World** (*kweilin.svg*)

Panoramę przedstawioną przez Rysunek 3.2.23 stworzyłem ustawiając obok siebie oryginalny obraz tła (*model/p40/background/kweilin.png*) i jego lustrzane odbicie — tak jak to opisałem na str. 351. Tym razem jednak zrobiłem to w Inkscape, a nie w GIMP (w Inkscape łatwiej się operuje obszarami wypełnionymi gradientem). Prawa część panoramy to klon lewego obrazu, przerzucony w poziomie. Linie horyzontu umieściłem tu w połowie wysokości (bo obserwator stoi na ziemi). Głębsze obszary dolnej półsfery wypełniłem białym gradientem, aby zapewnić otoczeniu odpowiednią energię do rozjaśniania cieni (por. 148, Rysunek 3.2.15).



Zmiana obrazu panoramy otoczenia ma minimalny wpływ na matowe powierzchnie modelu w tej scenie<sup>1</sup>. Co innego, gdyby były gładkie! W aktualnej projekcji oszkleniu kabiny także nie widać odbić otoczenia. Zdecydowaną różnicę widać za to na lustrzanej powierzchni sondy (Rysunek 3.2.24):



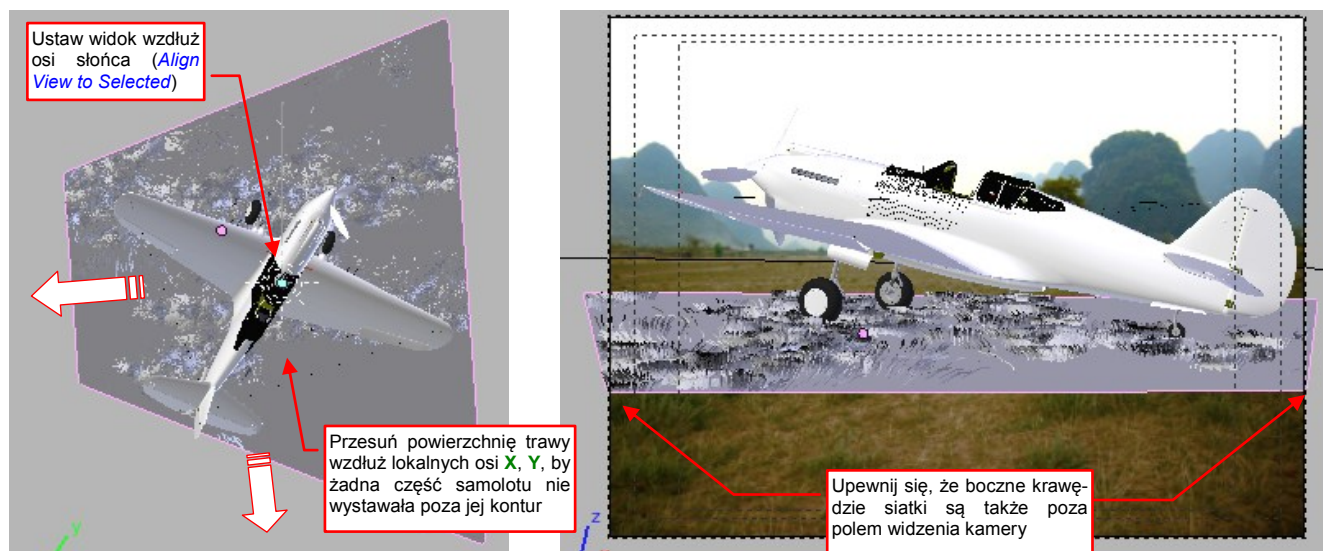
Rysunek 3.2.24 Odbicia starej i nowej panoramy otoczenia na powierzchni sondy

Zwróć uwagę, że linia horyzontu panoramy jest pochylona — to dlatego, że mamy obróconą kamerę (por. str. 143, Rysunek 3.2.2). Należy to skompensować za pomocą dodatkowego węzła transformacji (Rysunek 3.2.25):



Rysunek 3.2.25 Dostosowanie linii horyzontu panoramy

Czas dokończyć naszą scenę. Ustaw oś widoku 3D wzdłuż osi słońca (*Align View to Selected → Top*) i upewnij się czy cały cień samolotu w nowym położeniu mieści się na powierzchni trawy. Jeżeli nie — przesunij obiekt **Grass** wzdłuż lokalnych osi **X, Y** (Rysunek 3.2.26):



Rysunek 3.2.26 Dopasowanie obszaru trawy do zmienionego cienia samolotu

<sup>1</sup> Zdecydowałem się tu umieścić wzmiankę, jak przygotować ten obraz dla scen na ziemi, bo w sekcji 9.12 koncentrowałem się wyłącznie na stworzeniu panoramy do sceny w locie.

Na ostatecznym obrazie zdecydowałem się dodać drugą, lądującą maszynę, by podkreślić, że samolot stoi na lotnisku polowym (Rysunek 3.2.27):



Rysunek 3.2.27 Nowa wersja ujęcia z boku

Lądujący samolot to kopia modelu umieszczona w drugiej scenie, stworzonej metodą opisaną w poprzedniej sekcji (por. także str. 213). Aby uzyskać jej efekt rozmycia śmigła, włączyłem z powrotem w zestawie *Render* opcję *Motion Blur*. Umieściłem także pod tym lądującym samolotem podłoże — głównie po to, aby jego dolne powierzchnie nie były zbyt jasne. Plik z tą sceną znajdziesz w danych towarzyszących książce (to *model/p40/history/P40B-9.02w1.blend*).

Uzyskany obraz warto dodać drobnym korektom (postprocessing). Stworzyłem w tym celu prosty schemat, wykonujący dwie operacje: niewielkiego zwiększenia nasycenia barw (w węźle *Hue Saturation Value*) oraz drobnego wyostrenienia obrazu (w węźle *Filter:Sharpen* — Rysunek 3.2.28):



Rysunek 3.2.28 Dodatkowy postprocessing obrazu



Rysunek 3.2.29 przedstawia rezultat uzyskany w wyniku tych dwóch operacji:



Rysunek 3.2.29 Poprawiona wersja sceny na ziemi

Oprócz zwiększenia nasycenia barw zastosowałem tu także niewielkie rozjaśnienie (zwiększając wartość *HSV:Value* o 3%). Myślę, że po tych końcowych poprawkach obraz wygląda lepiej. Aby poprawić kompozycję całości, możesz jeszcze odciąć jego dolną część. (Tak aby samolot znalazł się dokładnie w połowie wysokości obrazu).

**Podsumownie**

- Gdy brakuje nam odpowiedniej fotografii tła miejsca, w którym chcielibyśmy umieścić nasz model, można spróbować skomponować odpowiedni obraz. W tej sekcji stworzyłem obraz lotniska z dwóch innych zdjęć (str. 143);
- Aby poprawnie ustawić model na tle fotografii, musisz uwzględnić położenie horyzontu i właściwe proporcje obiektów na zdjęciu i modelu (np. wysokość trawy na powierzchni lotniska polowego). Zazwyczaj nie wystarczy przesuwac kamery, trzeba także ją trochę obrócić (str. 143). W naszym przypadku kamera jest skierowana na obiekt **Target** za pomocą ograniczenia **Track To**, i jej obrotem steruje obrót tego obiektu.
- Oświetlenie sceny na ziemi składa się z tych samych elementów co sceny w powietrzu. Źródłem światła kierunkowego jest słońce (**Sun** — str. 144) o podobnej energii co w scenach w locie. Gdy na tle widać zamglenie, można zwiększyć rozmiar źródła **Sun**, by uzyskać „miękkie” granice cieni. Źródłem światła rozproszonego jest panorama, zbudowana z części górnej i dolnej. Górną częścią jest taki sam obraz nieba jak w scenach w locie, dolną — obraz skomponowany z fotografii tła (str. 151-152);
- Cień samolotu uzyskujemy na powierzchni, podstawionej pod koła modelu (str. 145). To nasz model „gruntu”. Najprostszą metodą integracji cienia z tłem jest pokrycie powierzchni gruntu tym samym obrazem co tło sceny, w takim samym mapowaniu (**Texture Coordinate:Window**). Ze względu na oświetlenie słońcem taką teksturę trzeba w tym przypadku przyciemnić (str. 147). Jasnością cienia sterujemy zmieniając przejrzystość materiału podłoża (str. 148);
- Cień samolotu można dodatkowo „urealnić”, umieszczając na podłożu widoczne na obrazie tła nierówności. (Chodzi o te, które zniekształcają cień). Zazwyczaj jest to trawa, którą można stworzyć za pomocą tzw. systemu cząsteczek (**Particles**), typu **Hair** (str. 148);
- Nieregularne kępki trawy tworzy się za pomocą „malowania” wag (tryb **Weight Paint** — por. str. 149). Do przeniesienia ich zarysu ze zdjęcia może się przydać narzędzie do nanoszenia „linii koncepcyjnych” — **Grease Pencil** (por. str. 421);
- Światła sceny warto przypisać (relacją **Parent**) do płaszczyzny podłoża. Podłoże można skierować (ograniczeniem **Locked Track**) na kamerę (str. 146). Takie ustawienia umożliwiają wygodne obracanie kamery wokół modelu (str. 150, 151);



## Szczegóły obsługi programów

Zawartość tej części bardzo przypomina zawartość plików podpowiedzi do trzech programów: GIMP, Inkscape i Blendera. Są tu zestawione opisy tych poleceń, które zostały wykorzystane w poprzednich częściach książki. Z założenia będziesz z nich korzystał wyrywkowo (por. „Jak czytać tę książkę?”, str. 12). Nie doszukuj się więc w kolejności rozdziałów i sekcji tej części jakiejś przemyślanej metody wprowadzania w obsługę programu. Do tego służy część pierwsza — „Budowa modelu”. To tam specjalnie dobrałem kolejność prezentowanego materiału w ten sposób, byś mógł stopniowo poznawać wszystkie potrzebne zagadnienia.

Gdy zdecydujesz się wydrukować tę książkę dla własnych potrzeb, proponuję pozostawić tą część w postaci elektronicznej. W przeglądarce PDF szybciej znajdziesz podaną w części pierwszej stronę, niż kartkując gruby wydruk. (No i zużyjesz mniej drzew z lasów...).

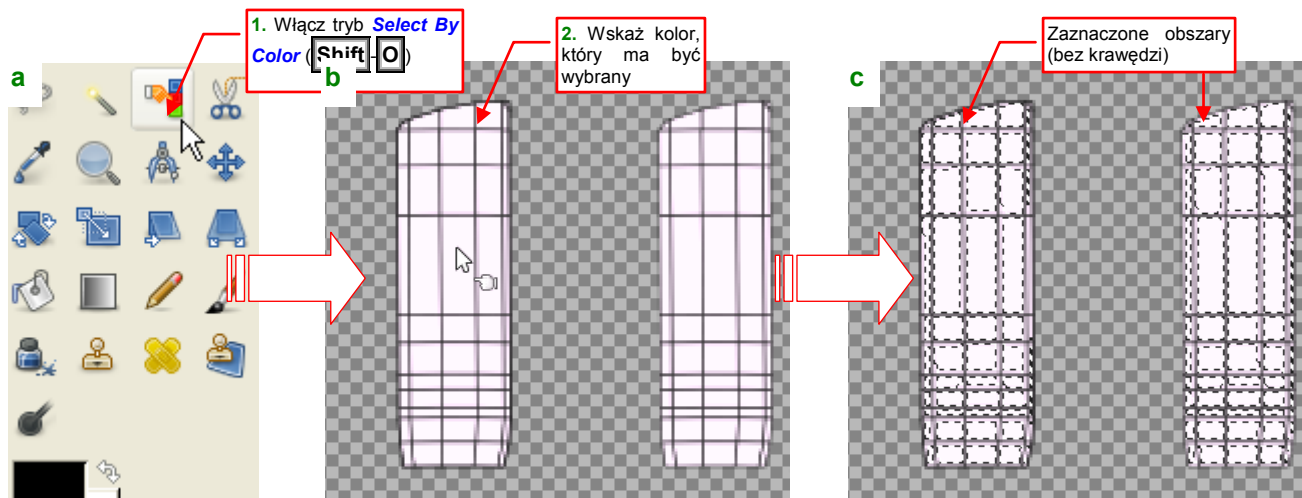
---

## **Rozdział 4. GIMP — szczegóły obsługi**

Nazwa GIMP pochodzi, jak sądzę, od pierwszych liter angielskiego **Graphic Image Processor**. Jest to jeden z pierwszych, dużych i ukończonych projektów [Open Source](#).

#### 4.1 Zaznaczanie — wg wskazanego koloru

Wywołaj polecenie z menu: **Select→By Color** (**Shift-O**), albo naciśnij przycisk z przybornika (Rysunek 4.1.1a):



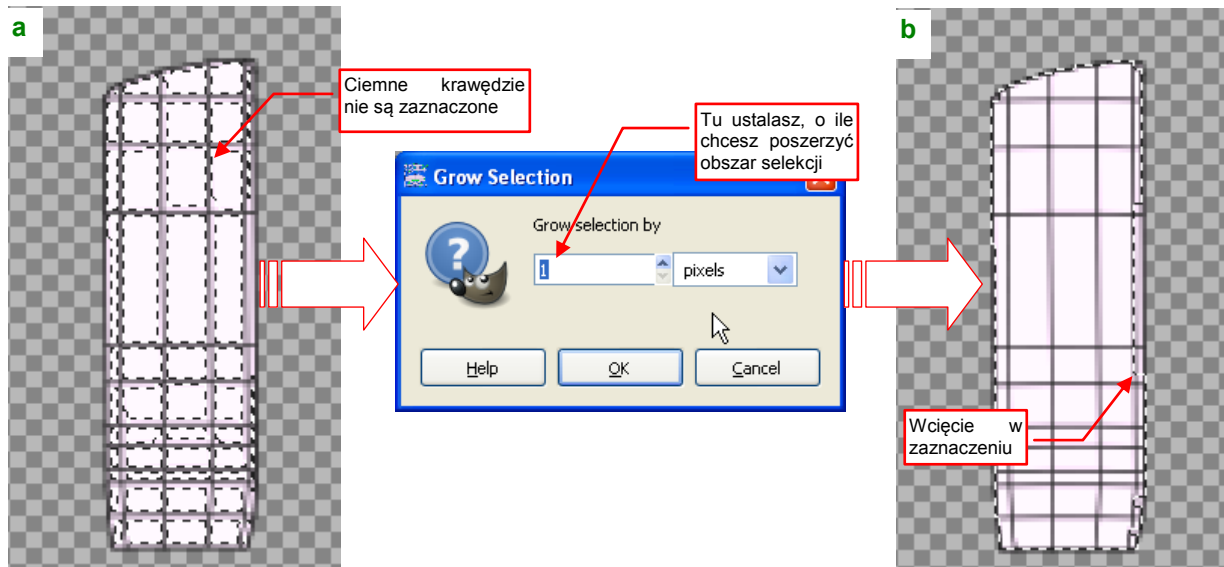
Rysunek 4.1.1 Zaznaczenie poprzez wskazanie koloru

Następnie kliknij (**LPM**) w obszar o odpowiednim kolorze (Rysunek 4.1.1b). W efekcie na rysunku zostanie zaznaczony każdy piksel o wskazanej barwie. W przykładzie na ilustracji wskazałem kolor biały, więc czarne krawędzie siatek nie zostały zaznaczone (Rysunek 4.1.1c).



## 4.2 Zmniejszanie/zwiększanie obszaru zaznaczenia

Aby poszerzyć obszar selekcji, wywołaj polecenie **Select→Grow...** (Rysunek 4.2.1):



Rysunek 4.2.1 Poszerzanie zaznaczonego obszaru (polecenie **Select→Grow...**)

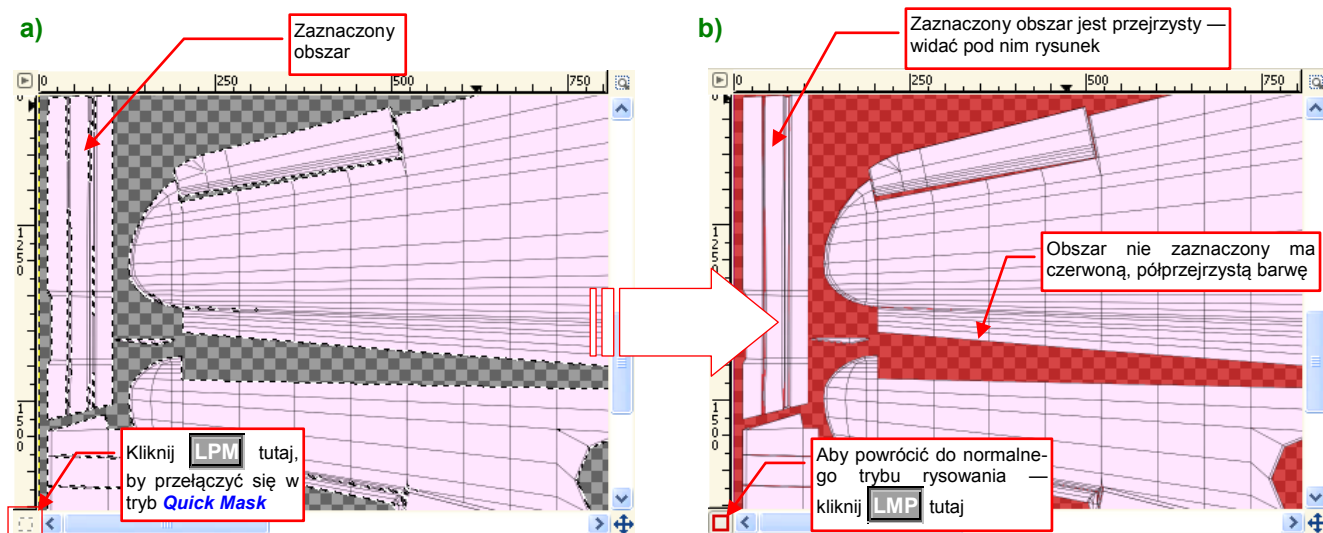
W oknie dialogowym **Grow Selection** (Rysunek 4.2.1a) ustal rozmiar powiększenia (w pikselach) i naciśnij **OK**. W ten sposób możesz usuwać mniejsze przerwy w obszarze selekcji. Większe wcięcia — takie, jakie pokazuje Rysunek 4.2.1b) — tylko się zmniejszą. Takie miejsca pozostaje zaznaczać ręcznie (np. maską — por. str. 160).

- Odwrotną operacją do **Select→Grow** jest **Select→Shrink**: zmniejsza obszar zaznaczenia

### 4.3 Zaznaczenie — poprawianie za pomocą *Quick Mask*

W sekcji 4.1 (str. 158) przedstawiłem tworzenie zaznaczenia z kanału Alfa (nieprzeźrystości) aktualnej warstwy. Podobny efekt można także osiągnąć w inny sposób — wykorzystując tzw. „chwilową maskę” (*Quick Mask*). Jest to metoda bardzo przydatna do obsługi obszarów selekcji o bardzo złożonych kształtach.

Rysunek 4.3.1a) przedstawia przykład skomplikowanego obszaru selekcji. Przełączymy się w tryb *Quick Mask*, aby go zmienić. Możesz to zrobić na kilka sposobów: wywołując polecenie z menu (*Select→Toggle Quick Mask*), skrótem **Shift-Q**, lub klikając **LPM** w lewy dolny narożnik rysunku (Rysunek 4.3.1a):

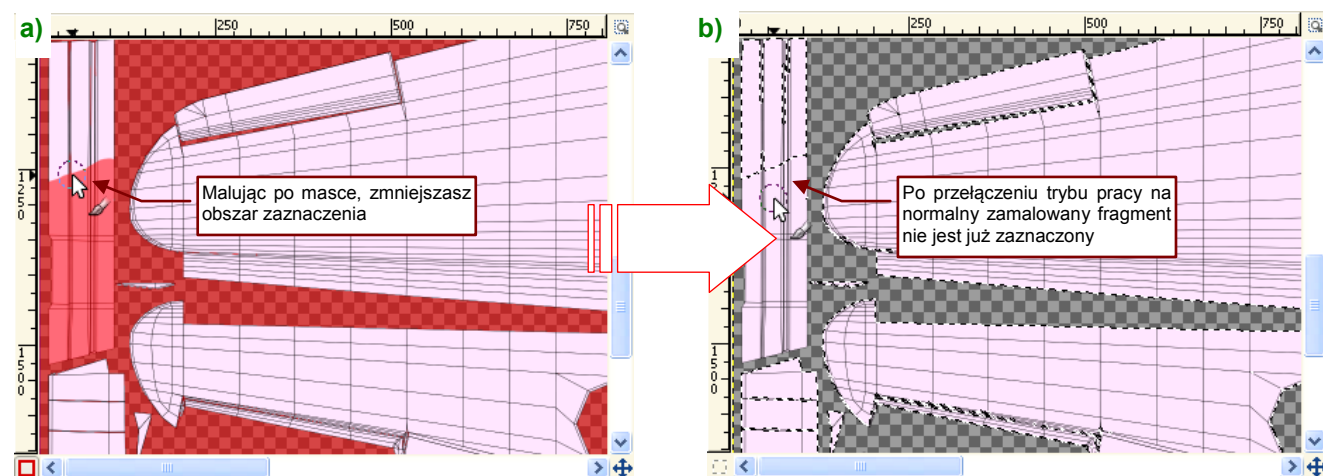


Rysunek 4.3.1 Przełączenie w tryb rysowania zaznaczenia (*Quick Mask*)

Możesz traktować *Quick Mask* jak specjalną, chwilową warstwę, nałożoną na rysunek. Zaznaczony obszar jest na niej zupełnie przezroczysty. Reszta jest pokryta półprzeźrystym kolorem czerwonym (Rysunek 4.3.1b).

- Do przełączenia się z powrotem w tryb normalny służą te same polecenia (*Select→Toggle Quick Mask*, skrót **Shift-Q**).

Po powierzchni chwilowej maski możesz rysować, używając dowolnego narzędzia z przybornika. Rysunek 4.3.2a przedstawia przykład użycia pędzelka (*Paintbrush*). Rysując, zmniejszasz obszar zaznaczenia:



Rysunek 4.3.2 Zmiana obszaru zaznaczenia — poprzez malowanie

- Aby zwiększyć obszar zaznaczenia, użyj w trybie *Quick Mask* „gumki” (*Eraser*). Wystarczy usunąć z maski kolor czerwony.

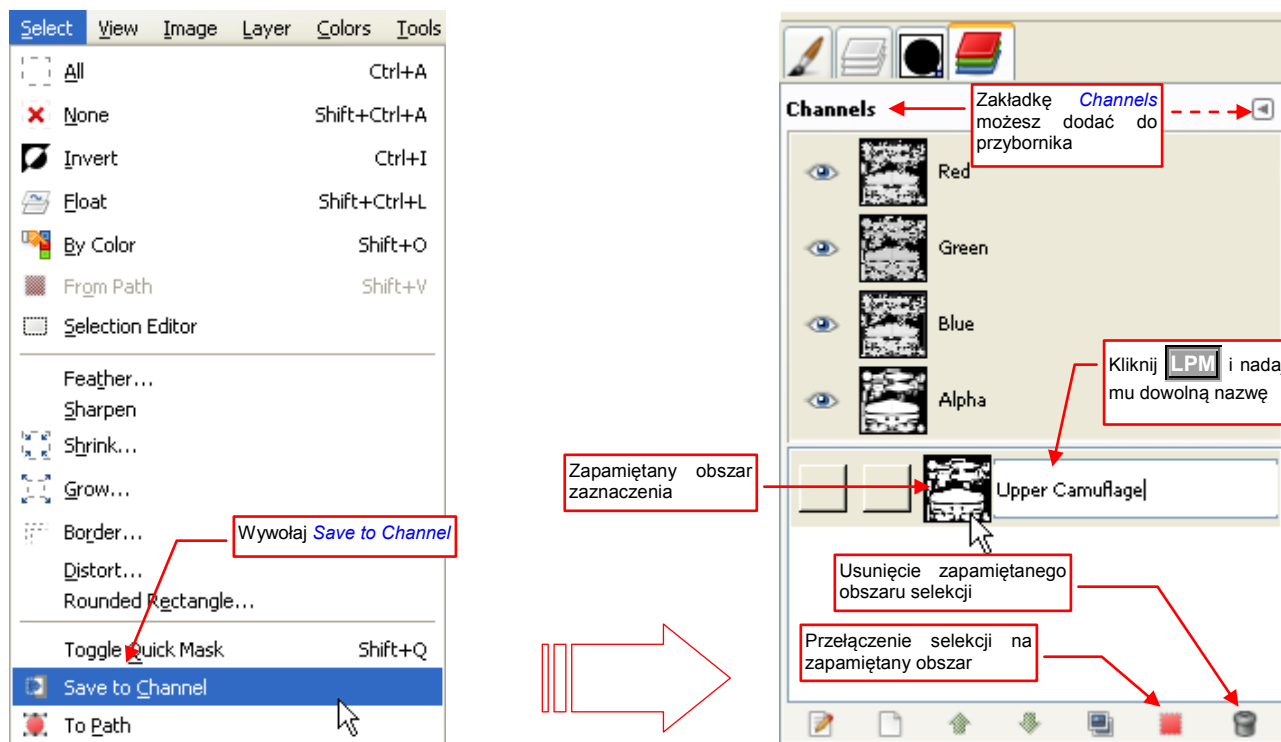
Podczas pracy z maską należy uważać na wybieranie narzędzi. Jeżeli wybrałeś w trybie normalnym wybrane narzędzie *Paintbrush*, i przełączyłeś się na *Quick Mask*, to dalej będziesz malować po powierzchni normalnej warstwy, a nie po masce.

- Wszystkie narzędzia, których chcesz użyć do zmiany *Quick Mask*, wybierz po przełączeniu się w ten tryb. Nawet jeżeli jest to ten sam *Paintbrush*, aktualnie używany w trybie normalnym — wybierz go raz jeszcze.

Może to efekt jakiegoś błędu w programie i zniknie w kolejnych wersjach — ale przezorność nie zawadzi.

#### 4.4 Zaznaczenie — zachowywanie obszaru w zakładce **Channels**

Czasami najbardziej pracochłonną częścią malowania w Gimpie jest przygotowanie odpowiedniego obszaru selekcji (na przykład w trybie **Quick Mask**). Czy można taki obszar zaznaczenia zachować do późniejszego użycia? Na szczęście — tak. Służy do tego polecenie **Select→Save to Channel** (Rysunek 4.4.1):

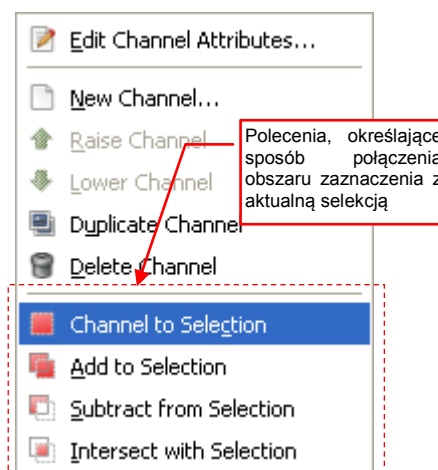


Rysunek 4.4.1 Zapisywanie obszaru zaznaczenia — do powtórznego użycia

Obszary zaznaczenia, zapamiętane w zakładce **Channels**, są przechowywane w pliku rysunku i mogą być użyte także po ponownym uruchomieniu programu. Użyj przycisku, umieszczonego u dołu zakładki **Channels** (Rysunek 4.4.1), aby przekształcić je z powrotem w aktualną selekcję.

Gdy w zakładce **Channels** klikniesz **PPM** na jednym z zapamiętanych zaznaczeń, otworzysz menu kontekstowe (Rysunek 4.4.2). Znajdziesz tam kilka przydatnych poleceń, służących do łączenia zapamiętanego obszaru zaznaczenia z aktualnym obszarem selekcji:

- **Channel to Selection** oznacza zwykłe zastąpienie aktualnego obszaru przez obszar zaznaczenia;
- **Add to Selection** pozwala dodać do aktualnie zaznaczonego na ekranie obszaru obszar zaznaczenia;
- **Subtract from Selection** zmniejsza aktualną selekcję o obszar zaznaczenia;
- **Intersect with Selection** zmniejsza obszar aktualnej selekcji do części wspólnej z obszarem zaznaczenia



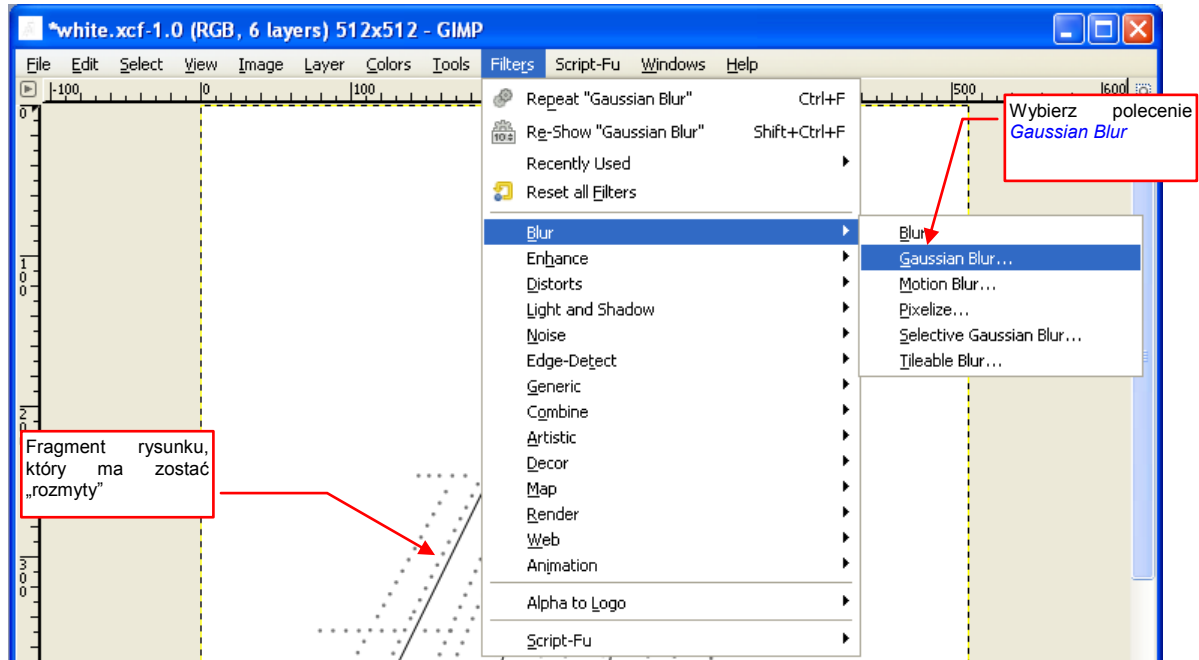
Rysunek 4.4.2 **Channels** — menu kontekstowe



#### 4.5 Rozmycie metodą Gaussa

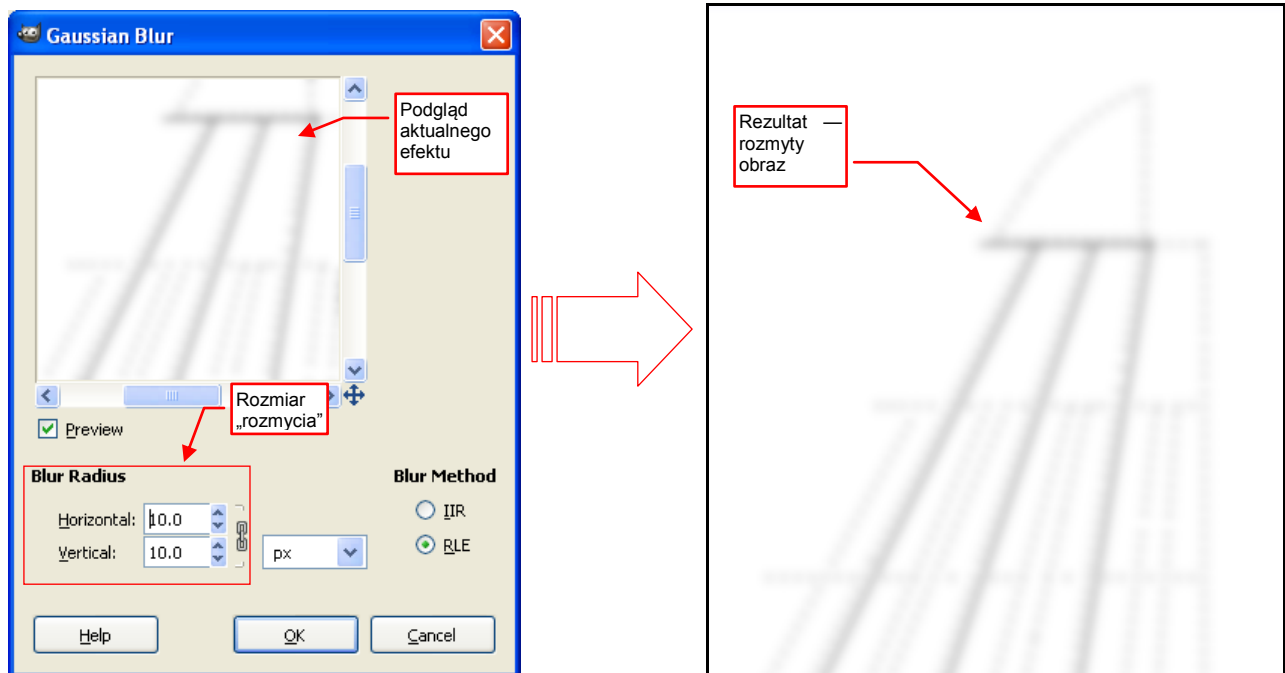
Jeżeli w GIMP jakiś obszar jest wybrany (selekcją) — transformacja będzie dotyczyć tylko tego obszaru. Jeżeli jednak nic nie jest wybrane, zmianie ulega cała zawartość aktualnej warstwy.

Rozmycie zademonstruję na przykładzie ostrego, biało-czarnego rysunku. Wywołaj polecenie **Filters→Blur→Gaussian Blur...** (Rysunek 4.5.1):



Rysunek 4.5.1 Menu efektów rozmycia (*Blur*)

W oknie dialogowym, które się pojawi, ustaw odpowiedni **Blur Radius** (Rysunek 4.5.2). Im większy promień ustawisz, tym intensywniejsze będzie rozmycie. Ma to także wpływ na czas przeliczania całego obrazu (większe promienie zabierają procesorowi więcej czasu).

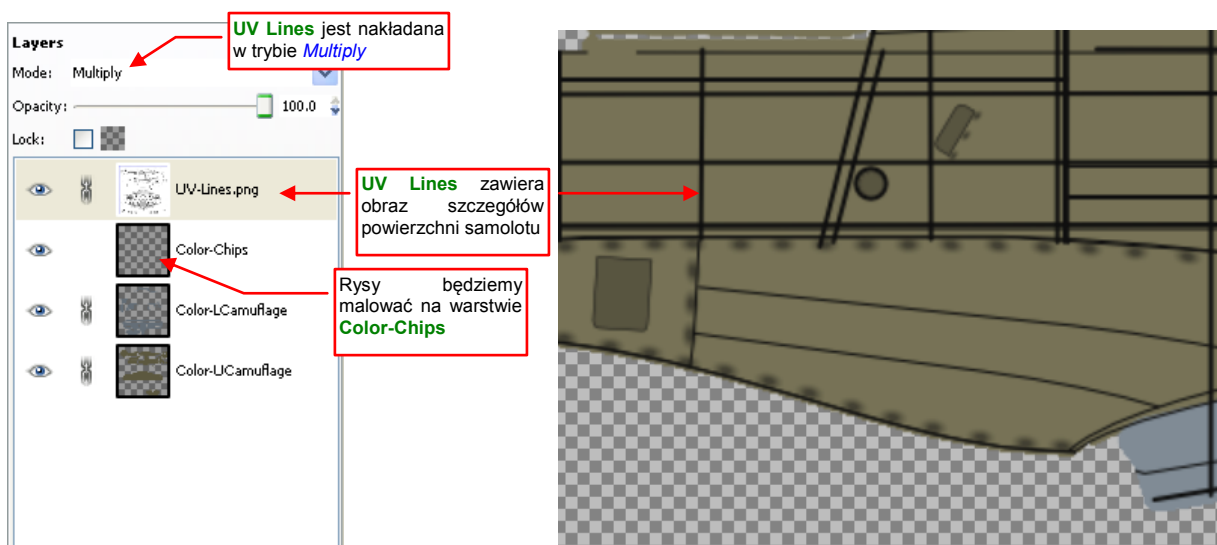


Rysunek 4.5.2 Okno polecenia **Gaussian Blur** i jego efekt

## 4.6 Nanoszenie przetarć i odprysków farby

Każda maszyna od chwili rozpoczęcia eksploatacji, zaczyna „obracać” w ślady zużycia. W przypadku samolotu są to, między innymi, odpryski i przetarcia farby. Właściwie naniesione, znacznie poprawiają realizm modelu. Każdy typ samolotu ma charakterystyczny układ takich śladów. Nim zaczniesz te skazy malować, warto najpierw przejrzeć zdjęcia, aby określić ich położenie, rozmiar i rodzaj. W czasie malowania uważaj, by nie ulec pokusie i nie pokrywać rysami całej powierzchni maszyny. W normalnej sytuacji obsługa naziemna nie próżnowała, i na bieżąco zamalowywała spostrzeżone uszkodzenia. Obszarem, z którym nie była w stanie sobie na dłuższą metę poradzić, jest w dolnopłatach górne poszycie skrzydła, przy kabinie. Mimo odpowiednich „dywaników”, widocznych na wielu zdjęciach, to miejsce zawsze było porysowane butami pilotów i mechaników. Każde zamalowanie w tym obszarze szybko znikało. W tej sekcji pokażę, jak można uzyskać efekt zużycia powłok samolotu właśnie na przykładzie tego fragmentu.

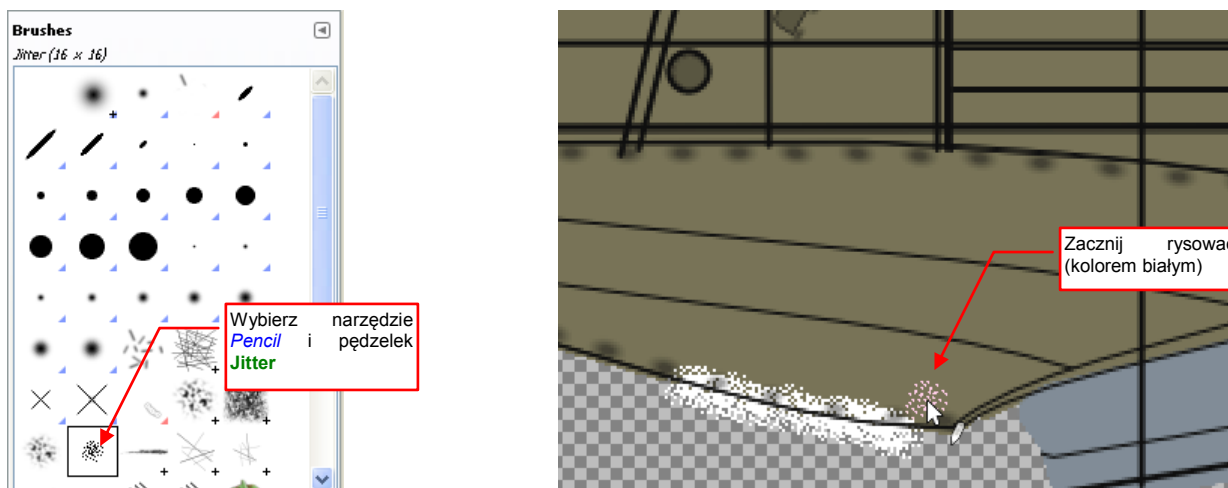
Do pracy bardzo się nam przyda obraz linii nitów i krawędzi paneli poszycia (np. wyeksportowany z wektorowego pliku Inkscape). Wczytaj go na odpowiednią warstwę referencyjną: **UV-Lines.png** (Rysunek 4.6.1):



Rysunek 4.6.1 Wstawienie warstw: podstawowej i referencyjnej

Wstaw do rysunku także nową warstwę **Color-Chips**, na którą naniesiemy obraz rys na powierzchni.

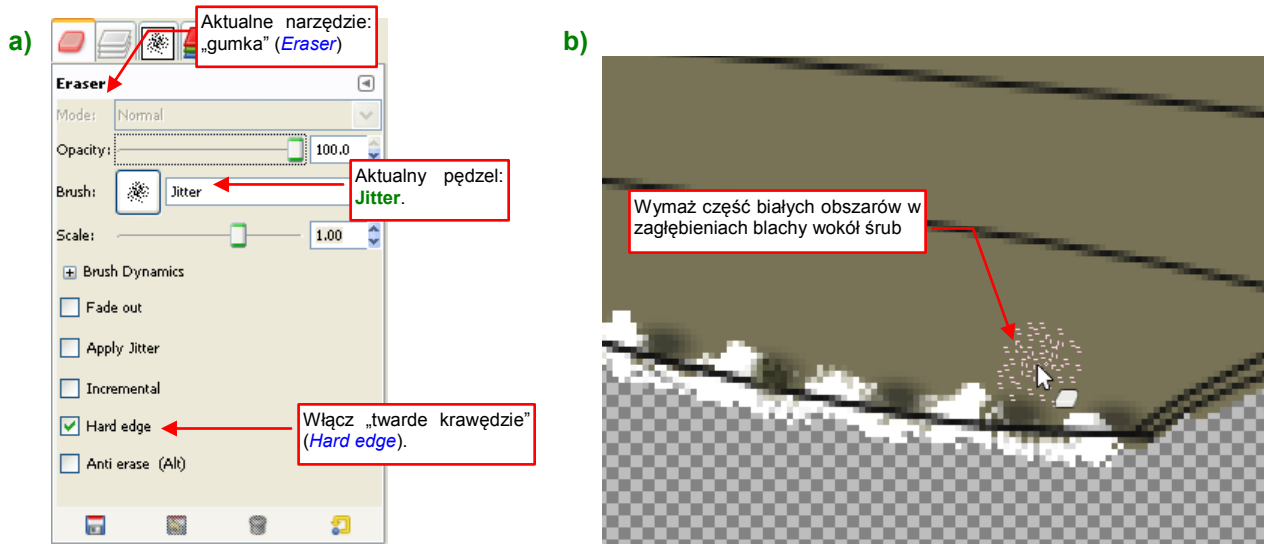
Zmień aktualny kolor na biały, i za pomocą „ołówka” (**[N]**, **Tools→Paint Tools→Pencil**) oraz specjalnego pędzelka o nazwie **Jitter**, zacznij rysować na warstwie **Color-Chips** obszar złuszczonej farby (Rysunek 4.6.2):



Rysunek 4.6.2 Wybór narzędzia i rysowanie

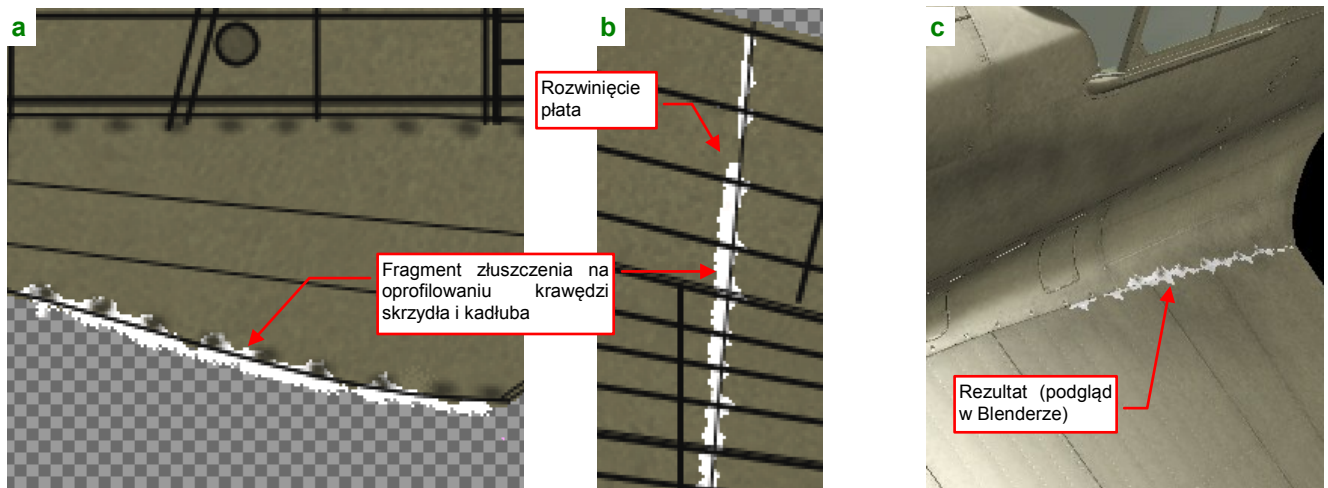
**Jitter** to dodatkowy typ pędzla, który stworzyłem specjalnie do rysowania tego rodzaju efektów. Plik z jego definicją znajdziesz w folderze `source\gimp\brushes` (por. str. 16) pod nazwą `jitter.gbr`. Aby stał się widoczny w zakładce **Brushes**, wgraj go do jednego z folderów ustalonych w konfiguracji Gimpa (por. Wyciąg III)<sup>1</sup>.

Nadal używając pędzla **Jitter**, przełącz się na usuwanie barwy za pomocą „gumki” (**Eraser**). Włącz w zakładce właściwości tego narzędzia opcję **Hard edge** (Rysunek 4.6.3a), i pousuwaj część obszarów odprysków, redukując je do właściwego kształtu. Postaraj się, aby „plamy” białego koloru miały jak najbardziej realistyczny (a jednocześnie — chaotyczny) kształt. To dobra okazja, by uwolnić w sobie uśpionego impresjonistę ☺. Postaraj się uwzględnić mniej przetartą powierzchnię w drobnych zagłębieniach — na przykład wokół śrub (Rysunek 4.6.3b):



Rysunek 4.6.3 Ustawienie „gumki” i usuwanie naddatków

Rysunek 4.6.4 pokazuje narysowaną na warstwie **Color-Chips** białą plamę, która na modelu wygląda jak złuszczenie farby:



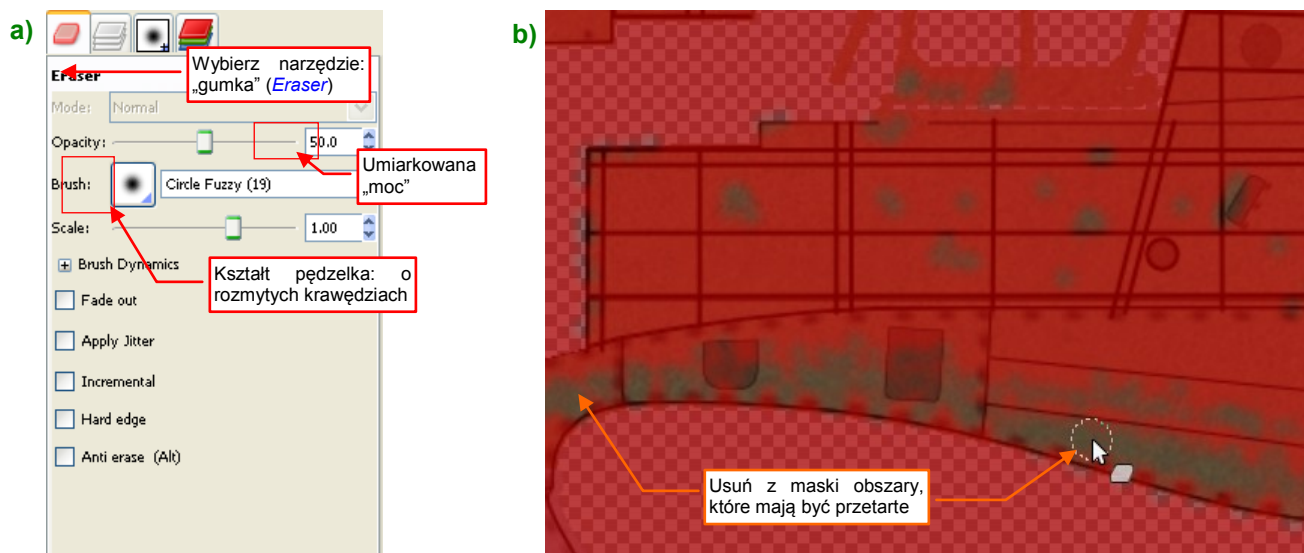
Rysunek 4.6.4 Przykład efektu łuszczącej się farby

Zwróć uwagę, że każda ze stron tego fragmentu na obrazie tekstury jest malowana osobno: część na rozwinięciu kadłuba (Rysunek 4.6.4a), a część — na rozwinięciu płata (Rysunek 4.6.4b). Mimo pozorów, niespecjalnie starałem się te dwie plamy „zsynchronizować”. Złożenie wyszło „samo” (Rysunek 4.6.4c) — wydaje mi się, że całkiem dobrze.

<sup>1</sup> W istocie pędzel **Jitter** to raptem kilkanaście ostrych, czarnych kropek

Odchodząca całym płatami farba świadczy o kiepskich składnikach, użytych do jej produkcji, i/lub problemach z obsługą naziemną<sup>1</sup>. Powłoki nanoszone bezpośrednio w zakładach Curtissa były dobrej jakości. Przejrzałem wiele zdjęć, ale nawet na szkolnych P-40, używanych intensywnie przez kilka ładnych lat, nie znalazłem śladu łuszczącej się farby. Co najwyżej — przetarte „do gołej blachy” pokrycie tam, gdzie wszyscy chodzili: na krawędzi kadłuba i skrzydeł.

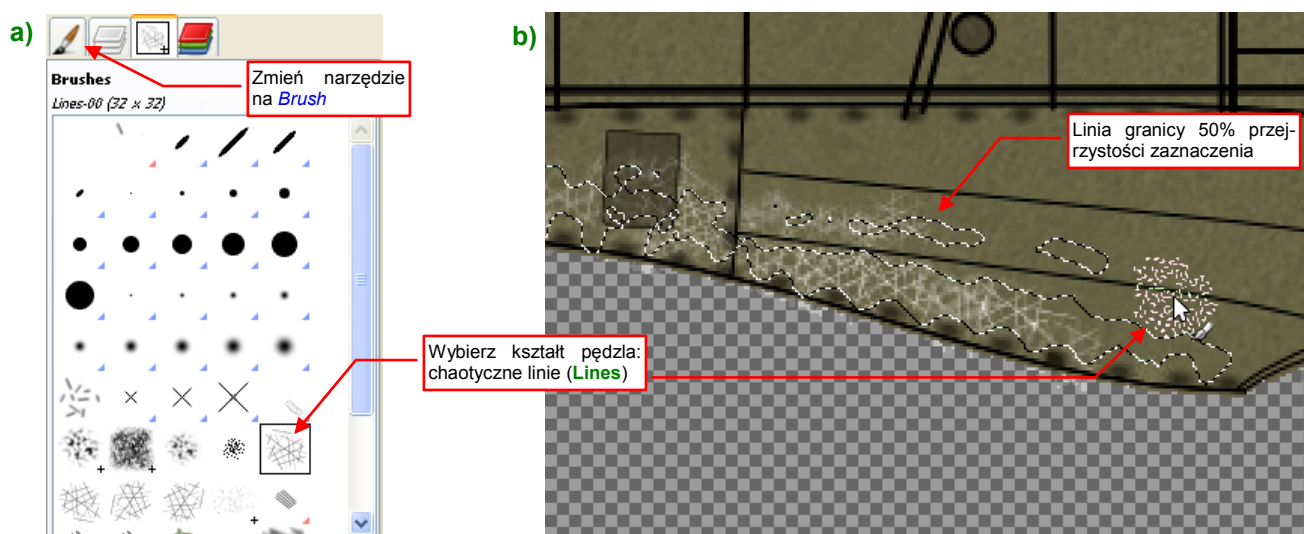
Malowanie przetarcia zacznij od przygotowania odpowiedniej maski, poprzez którą będziemy nanosić barwę. Najłatwiej jest ją narysować w trybie **Quick Mask** (**Shift**—**Q**, szczegóły — str. 160) (Rysunek 4.6.5b):



Rysunek 4.6.5 Efekt przetarcia — przygotowanie maski (tryb **Quick Mask**)

Z maski usuń za pomocą „gumki” (Eraser) kolor z tych obszarów, które mają być poprzecierane. Stosuj tu w dużym stopniu obszary półprzezroczyste — aby przetarcia płynnie zanikały w otoczeniu. Aby to osiągnąć, zmniejsz nieprzezroczystość (Opacity) „gumki” do 50%. Do usuwania stosuj pędzle o rozmytych brzegach (Circle Fuzzy — Rysunek 4.6.5a).

Teraz wyłącz **Quick Mask** i zacznij malować białe linie zarysowań powłoki. Użyj do tego zwykłego narzędzia **Paintbrush** i jednego z dołączonych do tej książki pędzli (<source\gimp\brushes\lines.gbr>) (Rysunek 4.6.6):



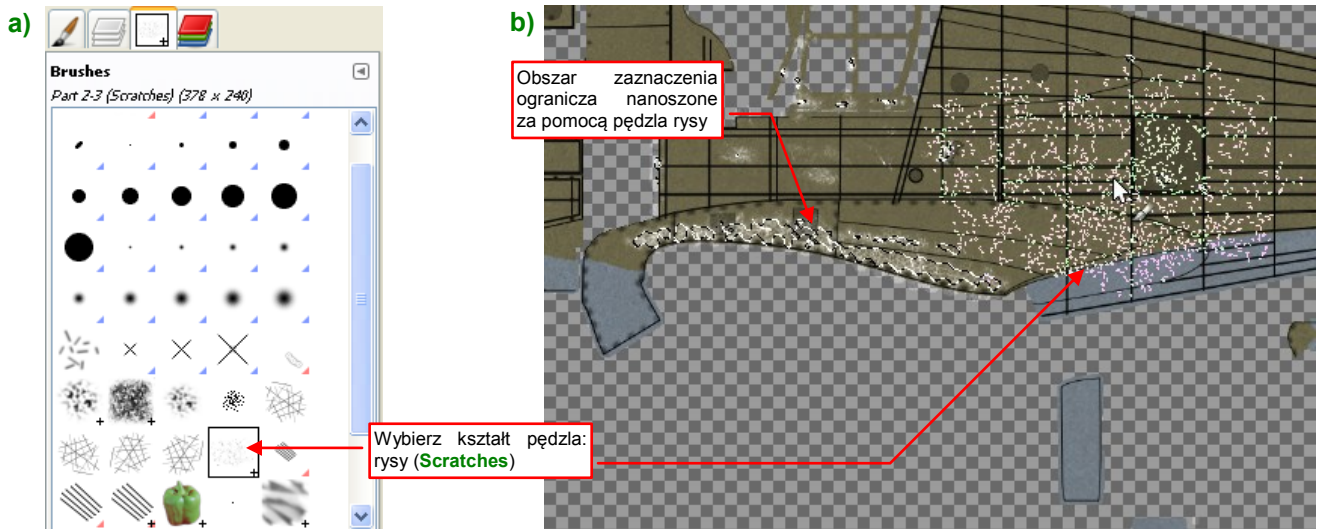
Rysunek 4.6.6 Efekt przetarcia — nanoszenie linii „zadrapań”

<sup>1</sup> Dużo „złuszczeń” widać na zdjęciach samolotów japońskich, operujących w trudnym, gorącym klimacie południowej Azji i wysp Pacyfiku. Innym przypadkiem takiego „obłazenia” były francuskie Hawk 75 pod koniec maja i w czerwcu 1940r. (Curtiss dostarczał je w barwie naturalnego duralu. Kamuflaż był nanoszony już po odbiorze, w warsztatach francuskich)



Aby znaleźć pokazywane przez Rysunek 4.6.6a) pędzle **Lines** w przyborniku **Brushes**, musisz przenieść pliki `source\gimp\brushes\*.gbr` do jednego z folderów, figurujących w konfiguracji GIMP (por. Wyciąg III). Alternatywnie — możesz dopisać w ustawieniach GIMP ten folder jako jeden z katalogów z definicjami pędzli. Podczas malowania za pomocą końcówki **Lines** unikaj nakładania równoległych linii blisko siebie. Umieść kolejne „przyłożenia” narzędzia w takiej odległości, by widać było pojedyncze kreski (Rysunek 4.6.6b).

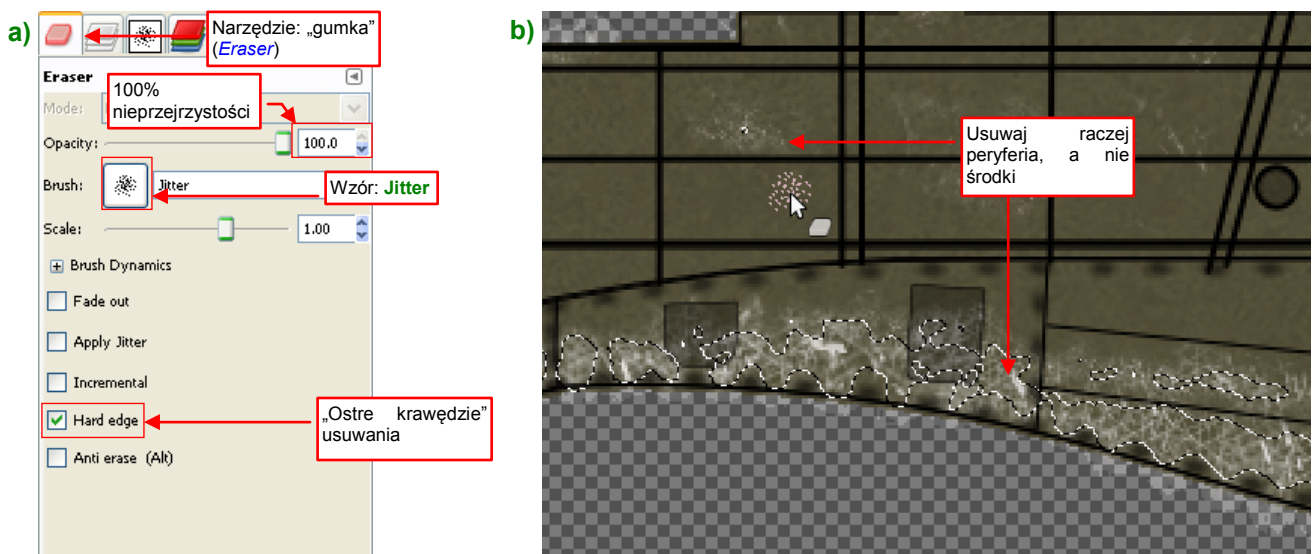
Aby uczynić układ rys na powierzchni bardziej chaotycznym, zmień teraz kształt końcówki na pędzel **Scratches** (Rysunek 4.6.7a):



Rysunek 4.6.7 Efekt przetarcia — nanoszenie innych szkaz powierzchni

Wzór tego pędzla — plik `scratches.gbr` — jest w tym samym folderze, co pozostałe wzory dołączone do tej książki (`source\gimp\brushes`). To relatywnie duża końcówka (Rysunek 4.6.7b). Nie przejmuj się jednak tak bardzo jej rozmiarem. Zaznaczony obszar rysowania skutecznie ograniczy jej zasięg. Tak samo jak linie, nanieś jej obrazy kolejnymi „kliknięciami” tak, by nie nakładały się na siebie.

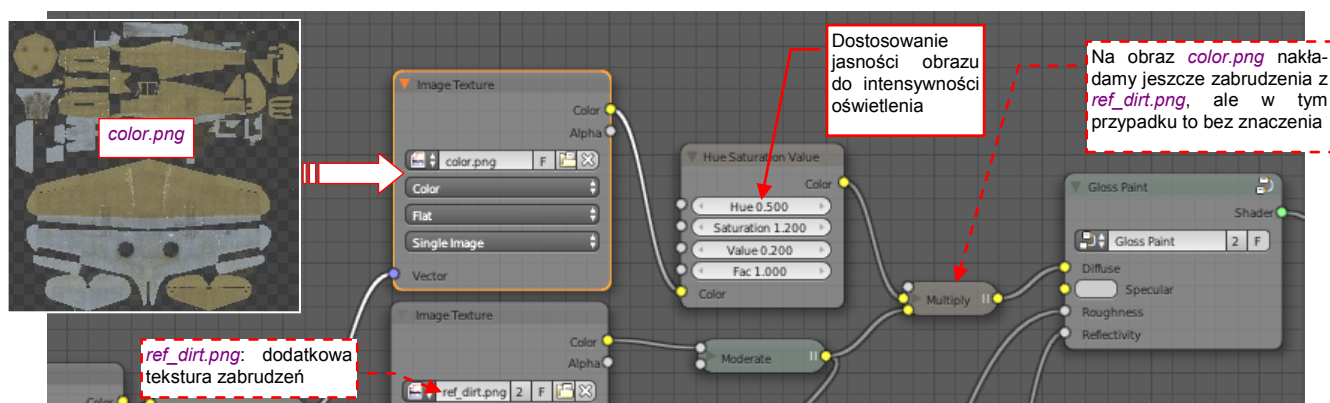
Na koniec można trochę ograniczyć różne naddatki plam, które w nieunikniony sposób pojawiły się podczas malowania. To ten sam finalny krok, co w przypadku odprysków farby. Zmień z powrotem narzędzie na „gumkę” (**Eraser**). Tym razem wymazywanie ma działać z pełną ostrością, więc zwiększ **Opacity** narzędzia **Eraser** do 100%, i włącz opcję **Hard edge** (Rysunek 4.6.8a):



Rysunek 4.6.8 Efekt przetarcia — usuwanie naddatków

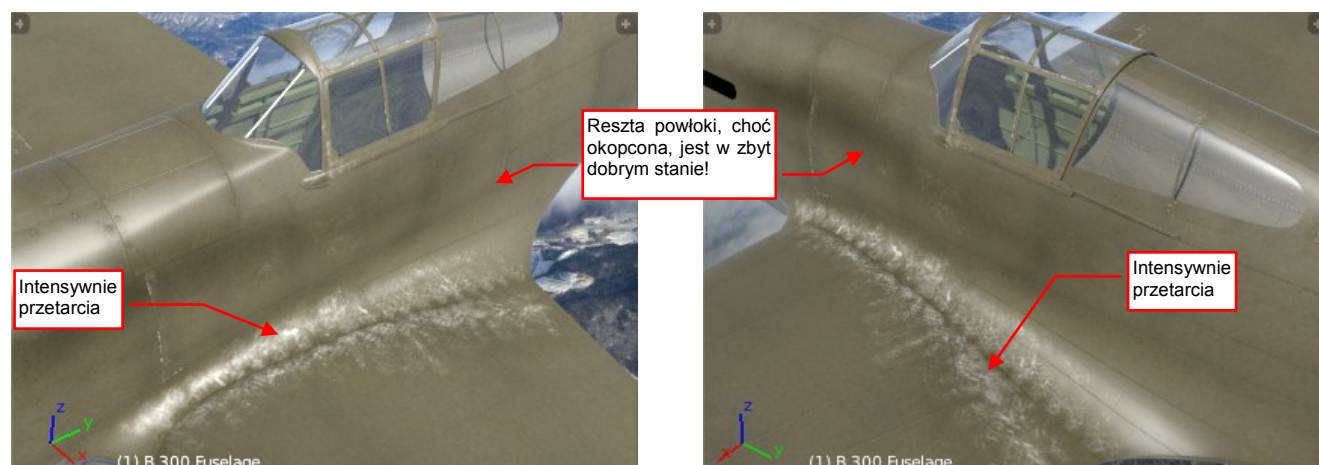
Wyłącz ograniczenie do zaznaczonego obszaru (**Select→None**). Wybierz pędzel **Jitter**. Używając „gumki” tego kształtu, postaraj się zmienić granice wytartych obszarów na mniej regularne (Rysunek 4.6.8b).

Wyeksportuj tak zmodyfikowany obraz barw do pliku *color.png*. Ten plik wykorzystujemy w Blenderze, w materiale poszycia samolotu (**B.Skin.Camouflage** — Rysunek 4.6.9, por. także Wyciąg III):



Rysunek 4.6.9 Użycie obrazu *color.png* w Blenderze (fragment schematu materiału **B.Skin.Camouflage**)

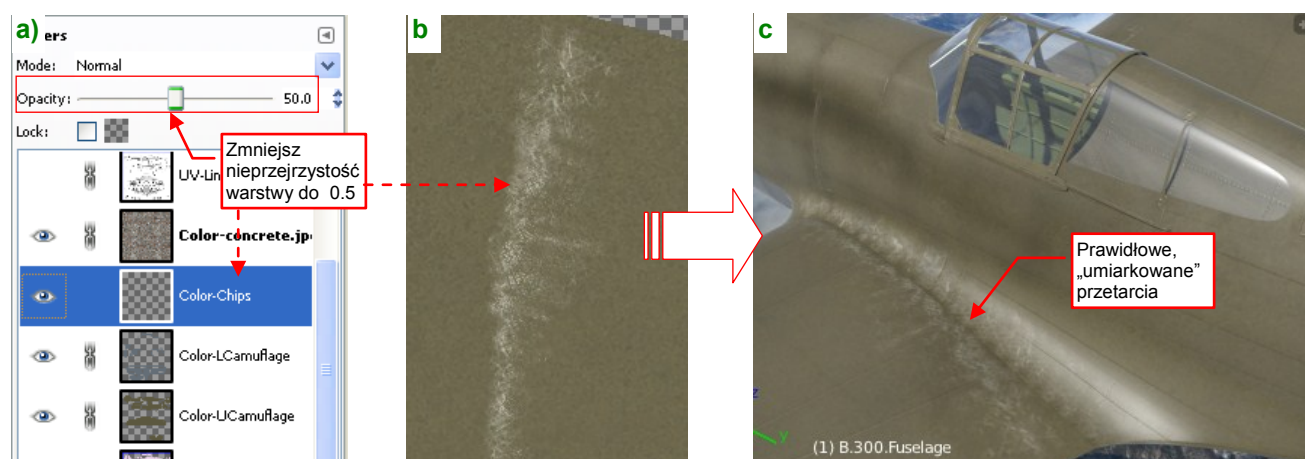
Rysunek 4.6.10 pokazuje, jak nasze przetarcia prezentują się w różnych ujęciach modelu:



Rysunek 4.6.10 Przetarcia na modelu (pełna intensywność)

Mówiąc szczerze, efekt jest na tyle intensywny, że reszta powierzchni wydaje się wręcz wołać o podobne „przetarcie”! To oznaczałoby dużo pracy. Na szczęście, obsługa naziemna zazwyczaj stawała na wysokości zadania a samoloty podczas wojny rzadko kiedy mają się okazję aż tak zestarzeć.

Zmniejsz w GIMP nieprzeźrystość warstwy **Color-Chips** do ok. 50% (Rysunek 4.6.11a). Rysunek 4.6.11b) pokazuje, jak „zbladły” w wyniku tej zmiany rysy na górnej powierzchni płata. Gdy wyeksportujesz ten obraz do pliku dla Blendera, uzyskamy na modelu właściwe, „umiarkowane” przetarcia (Rysunek 4.6.11c):

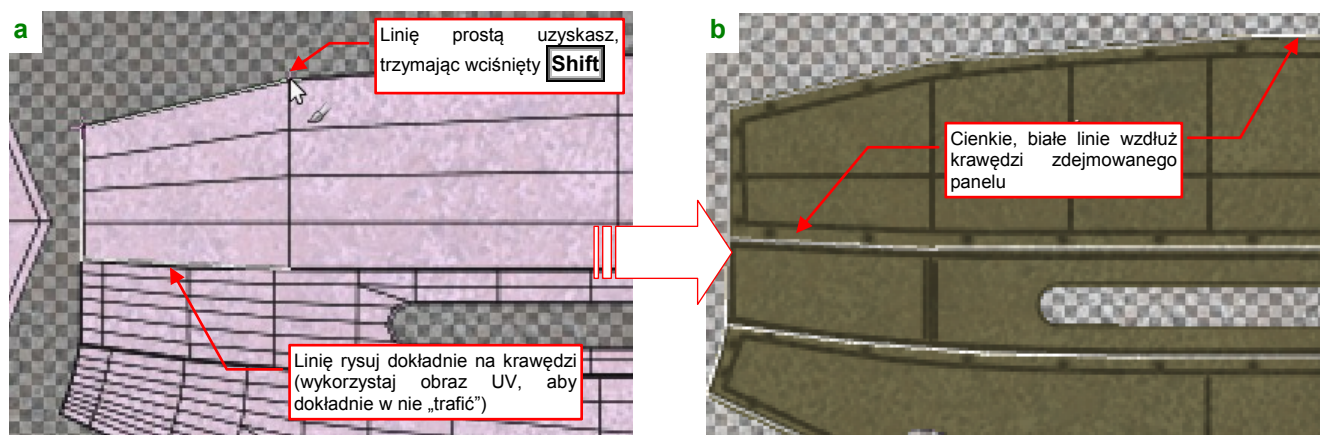


Rysunek 4.6.11 Efekt przetarcia — zmniejszenie intensywności



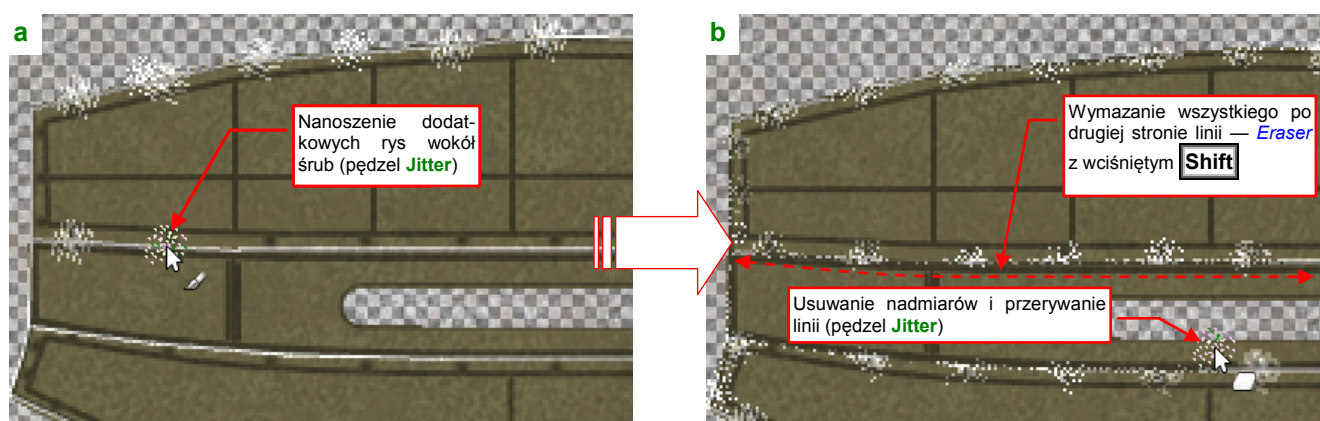
Choć powierzchnie P-40 były traktowane „z należyтым szacunkiem”, pokażę jeszcze, jak uzyskać drobne rysy wzdłuż krawędzi paneli silnika. Rysuje się je prościej niż „szerokie” przetarcia, takie jak u nasady skrzydła.

Zmień aktualny kolor na biały, narzędzie — na *Paintbrush*, i wybierz bardzo cienki pędzel (**Circle 03**). Narysuj wzdłuż krawędzi, które mają być porysowane, cienką białą linię (Rysunek 4.6.12):



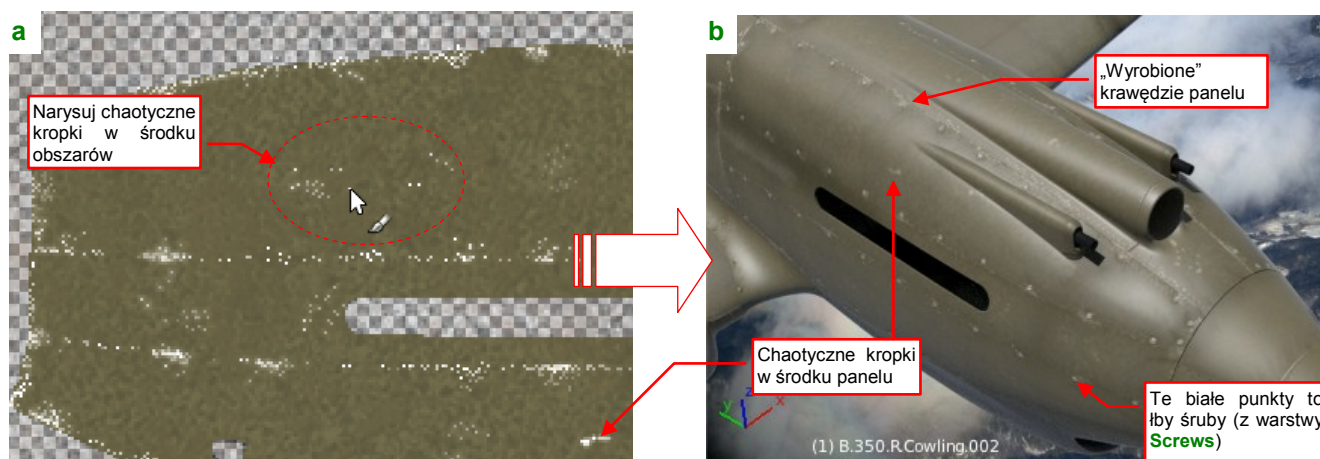
Rysunek 4.6.12 Zużyte krawędzie — naniesienie linii bazowej

Następnie zmień pędzel na **Jitter**, i dodaj po „pacnięciu” w okolicach każdej śruby (Rysunek 4.6.13a). Potem ogranicz te „pacnięcia” i poprzerywaj białą linię za pomocą „gumki” (pędzel bez zmian: **Jitter**), z włączoną opcją *Hard Edge* (Rysunek 4.6.13b):



Rysunek 4.6.13 Zużyte krawędzie — malowanie i wycieranie

Na koniec jeszcze namaluj najdrobniejszym pędzlem, gdzieś na środku każdego panelu, parę chaotycznych kropek — aby nie było tam „za czysto” (Rysunek 4.6.14a):

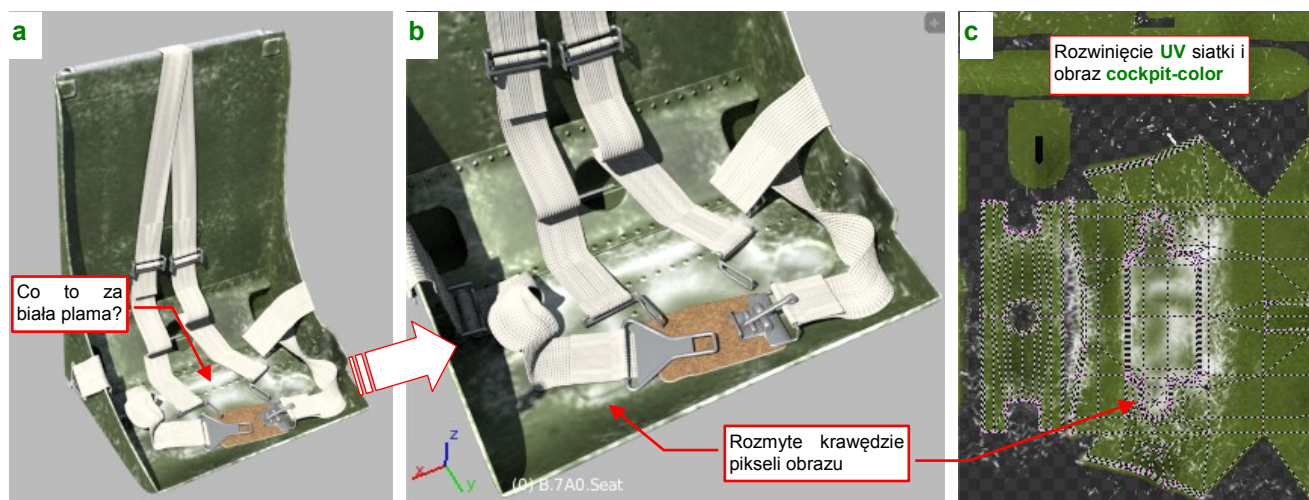


Rysunek 4.6.14 Zużyte krawędzie — ostatnie dodatki i efekt finalny

Rysunek 4.6.14b) przedstawia ostateczny efekt na modelu (po zmniejszeniu nieprzejrzystości warstwy **Color-Chips** do 50%). Uzyskaliśmy dyskretne rysy, podkreślające nieco krawędzie okapotowania silnika (na modelu wyszły nam zbyt dokładnie dopasowane, więc staram się je w ten sposób dodatkowo wyróżnić).

- Rysy na poszyciu warto malować jako bardzo wyraźne — tak, jak na jakimś „starym gracie”. Potem zawsze możesz je stonować, adekwatnie do stopnia zużycia odwzorowywanego egzemplarza samolotu. Wystarczy zmniejszyć nieprzejrzystość (**Opacity**) warstwy **Color-Chips**.

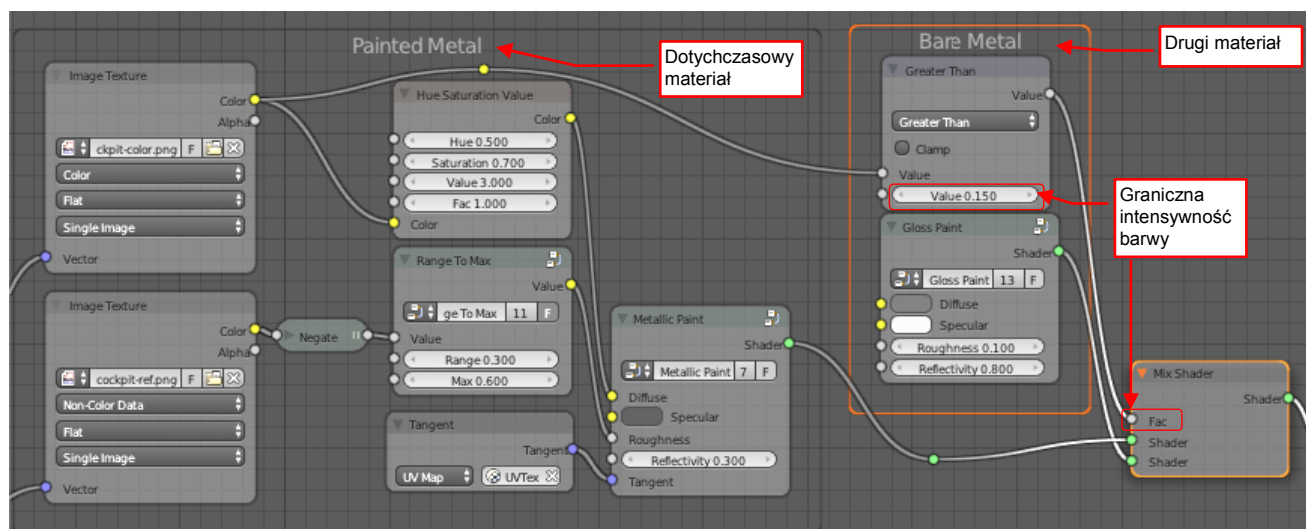
Wnętrze kabiny P-40 było pomalowane wyłącznie farbą podkładową (**Curtiss Green**). Zakładam, że łatwiej się przecierała niż powłoka powierzchni zewnętrznych. Dlatego na teksturze barwy materiału wnętrza kabiny (**B.Skin.Cockpit** — por. str. 85) namalowałem duże, białe przetarcia. Detale kabiny oglądamy z bliska, więc podczas wypełniania szczegółami tego obszaru szybko zacząłem dostrzegać że takie białe plamy w zbliżeniach nie wyglądają dobrze. Weźmy za przykład taki fotel pilota (Rysunek 4.6.15):



Rysunek 4.6.15 Duże przetarcia w zbliżeniach

Rysunek 4.6.15a) przedstawia widok ogólny. Siedzenie fotela powinno być przetarte „do gołego metalu”, stąd umieściłem w tym miejscu biały obszar (Rysunek 4.6.15c). Niestety, to nie wygląda jak przetarcie, a jak zwykła, biała plama. Szczególnie to przeszkadza w zbliżeniach (Rysunek 4.6.15b).

Trzeba było coś z tym zrobić. Przetarcia są najjaśniejszym elementem tekstury barwy, stąd spróbowałem użyć jej jak maski sterującej połączeniem z prostym shaderem metalu (odpowiednio ustawiony **Gloss Paint**). Wykorzystałem tu funkcję **Greater Than**, aby uzyskać ostre granice materiałów (Rysunek 4.6.16):



Rysunek 4.6.16 Zmodyfikowany materiał fotela (B.Skin.Cockpit)

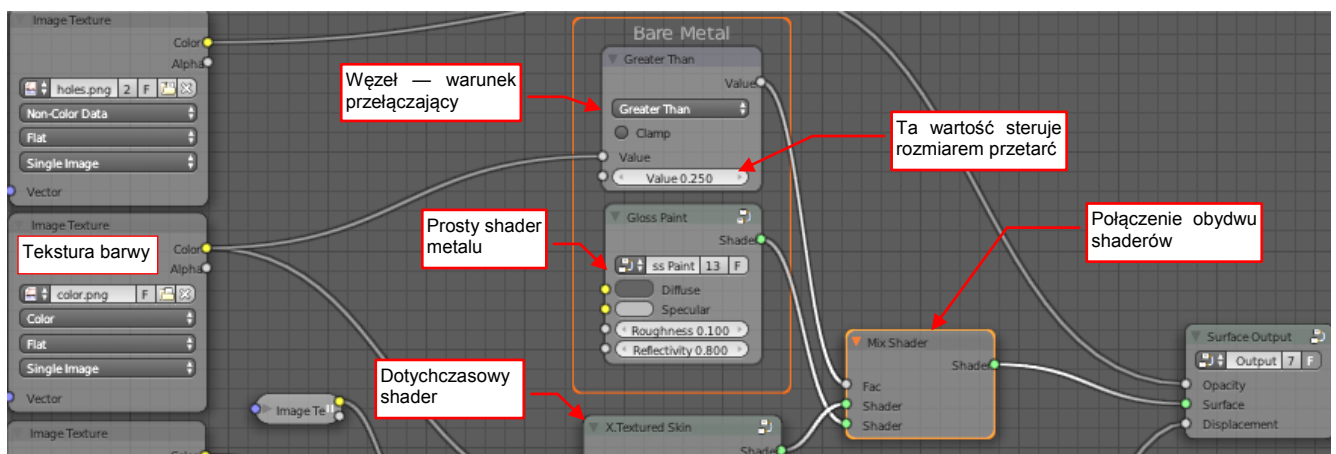


Rezultat takiej modyfikacji wygląda bardzo ciekawie. Biała plama na siedzeniu zmieniła się w metaliczną powierzchnię (Rysunek 4.6.17a). Gdy obejrzysz z bliska granice tego obszaru, zauważysz że dotychczasowe białe plamy teraz mają ostre krawędzie i wyglądają jak odpryski farby (Rysunek 4.6.17b). Widać to nawet w zbliżeniach (Rysunek 4.6.17c).



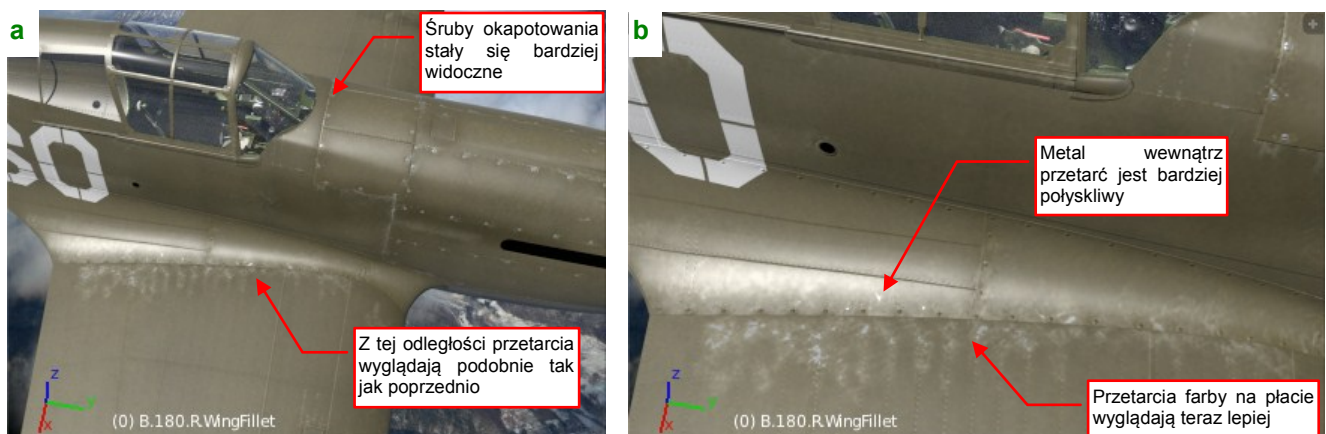
Rysunek 4.6.17 Fotel po modyfikacji materiału

Ten rezultat był na tyle zachęcający, że spróbowałem wprowadzić taką samą zmianę do materiału poszycia samolotu (**B.Skin.Camouflage** — Rysunek 4.6.18):



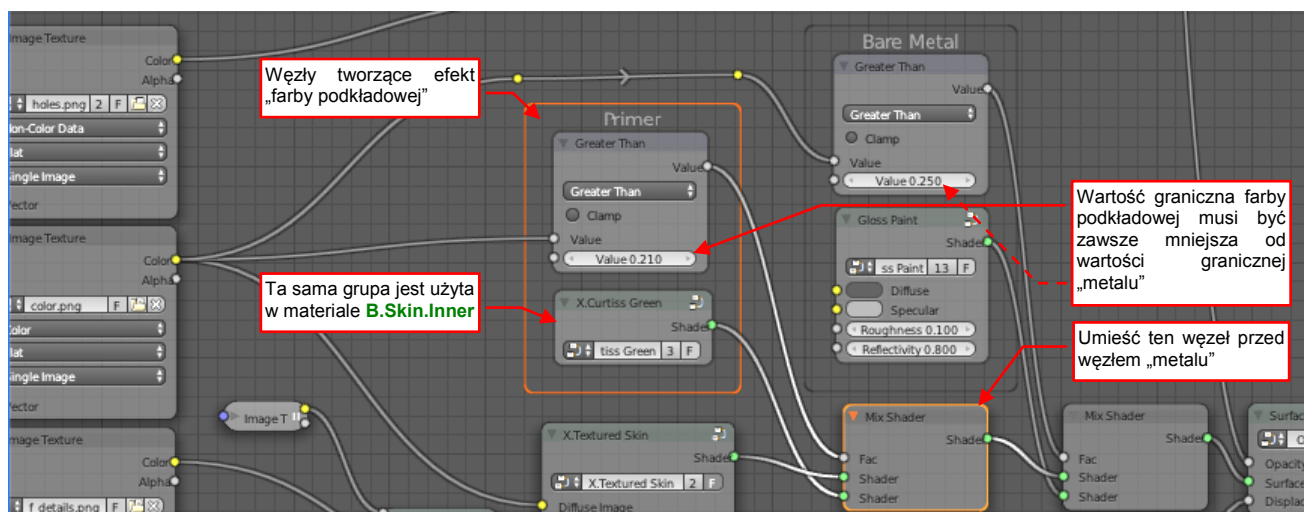
Rysunek 4.6.18 Dodanie do materiału poszycia (**B.Skin.Camouflage**) efektu „przetarcie metalu”

Po tej modyfikacji wyraźniejsze stały się śruby na okapotowaniu silnika (Rysunek 4.6.19a). W zbliżeniach ciekawe efekty daje większa połyskliwość obszarów przetarć (Rysunek 4.6.19b):



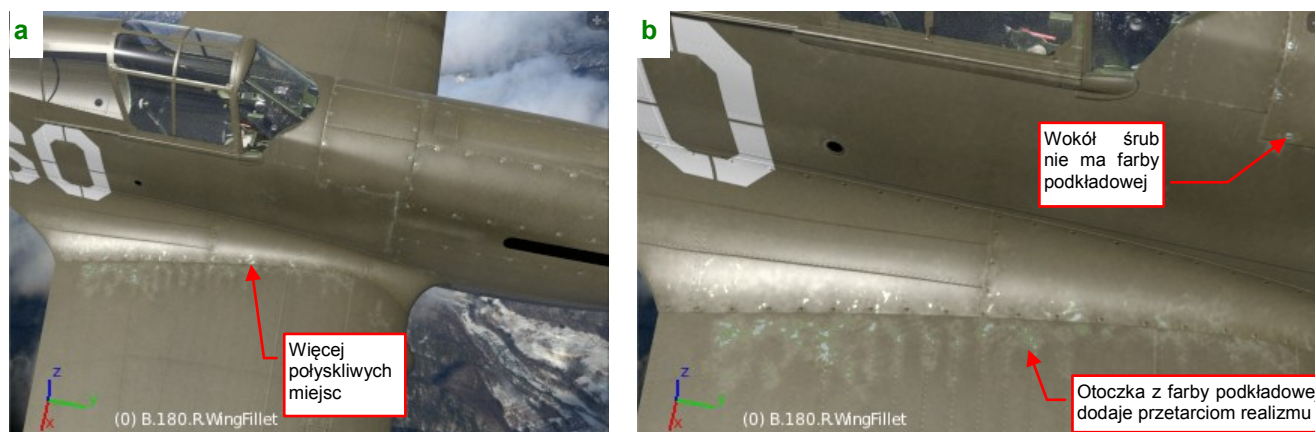
Rysunek 4.6.19 Materiał **B.Skin.Camouflage** po modyfikacji

Jeżeli producent samolotu używał jakiejś kontrastowej farby podkładowej (w wielu samolotach z USA była to jasna, żółtozielona **Zinc Chromate**), możesz spróbować ją odwzorować, tworząc wokół przetarc „otoczkę” w barwie podkładu. To może podnieść realizm rys na powierzchni. W przypadku modelu P-40 kolorem farby podkładowej był **Curtiss Green**, ten sam który wykorzystywaliśmy w materiale powierzchni wewnętrznych (**B.Skin.Inner**). Stworzyliśmy nawet specjalną grupę — shader **X.Curtiss Green**, ułatwiający powtórzenie tej barwy w wielu schematach (por. Wyciąg III). Teraz zastosujemy ją do ulepszenia efektu przetarcia. Wstaw do materiału **B.Skin.Camouflage** kolejny zespół z tym shaderem i węzłami **Greater Than** i **Mix Shader** (Rysunek 4.6.20). Węzeł **Mix Shader** efektu farby podkładowej podłącz przed węzłem **Mix Shader** efektu „przetarc do metalu”:



Rysunek 4.6.20 Dodanie do materiału poszycia efektu „farb podkładowej”

W efekcie wokół przetarc „do gołego metalu” pojawiły się zielonkawe obwódki. Widać je już z większych odległości (Rysunek 4.6.21a). Poprawiły także efekt przetarc w zbliżeniach (Rysunek 4.6.19b):



Rysunek 4.6.21 Materiał **B.Skin.Camouflage** po kolejnej modyfikacji

- Opisane powyżej rozwiązanie wykorzystujące intensywność tekstury barw do łączenia shaderów opracowałem dość późno, podczas modelowania kabiny. Dlatego materiały, które je wykorzystują, znajdziesz dopiero w pliku <model/p40/history/P40B-8.06.blend> (por. str. 16).

Rysowanie przetarc, zadrapań i zabrudzeń to zajęcie, w pewnym sensie, artystyczne. Często dwa zupełnie różne (i wykonane różnymi technikami) obrazy tekstur znajdują jednakowo wysokie uznanie wśród widzów. Pod żadnym pozorem nie należy traktować metod, które przedstawiłem w tej sekcji, jako ostatecznych. Pokazałem tu kilka rzemieślniczych „sposobów”, jak osiągnąć efekt na — bądźmy szczerzy — średnim poziomie. (To znaczy na poziomie, który przynajmniej nie razi widza). Na pewno można to zrobić lepiej lub inaczej. Jestem pewien, że z czasem dopracujesz się swojego własnego stylu.

## **Rozdział 5. Inkscape — szczegóły obsługi**

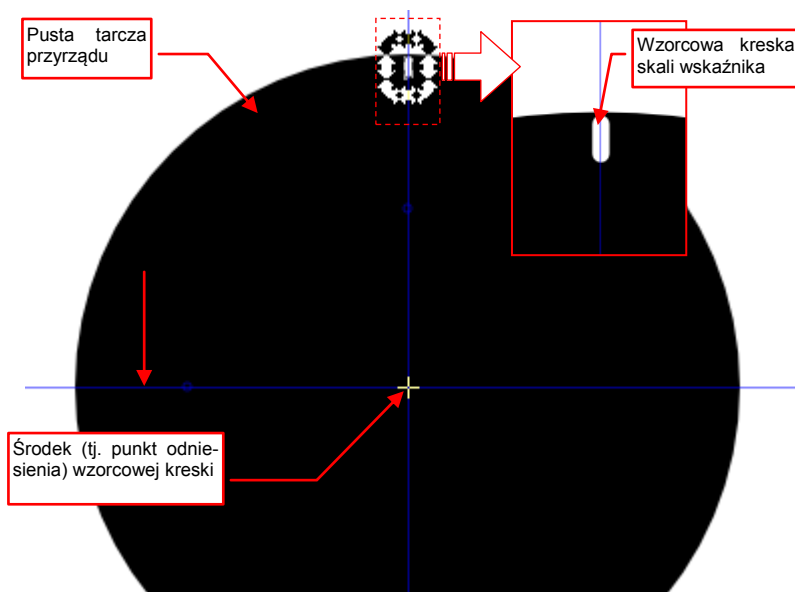
Na Inkscape natknąłem się przypadkiem. GIMP miał tylko marginalne możliwości rysowania linii wektorowych. A ja szukałem więc czegoś, w czym można by było wygodnie rysować (i poprawiać) linie nitów i łączenia blach. Gdy jednak wertowałem jego plik odpowiedzi, szukając szczegółów tej funkcji, natknąłem się na zdanie: "Rysowanie linii w GIMP nie jest tak wygodne, jak w wyspecjalizowanych programach, np. Inkscape".

Co to jest to "Inkscape"? Wystarczyło wpisać to hasło w Google, by znaleźć stronę tego projektu. Jest to najmłodszy z programów, używanych w tej książce. Nie osiągnął jeszcze "pełnoletności" (aktualne wersje mają nadal numer poniżej 1.00). W związku z tym potrafi czasami się "zawiesić", lub zakończyć się nagle z jakimś krytycznym błędem. Na szczęście nie dzieje się to zbyt często. W każdym razie nie zapominaj o częstym zapisywaniu swojej pracy do pliku!

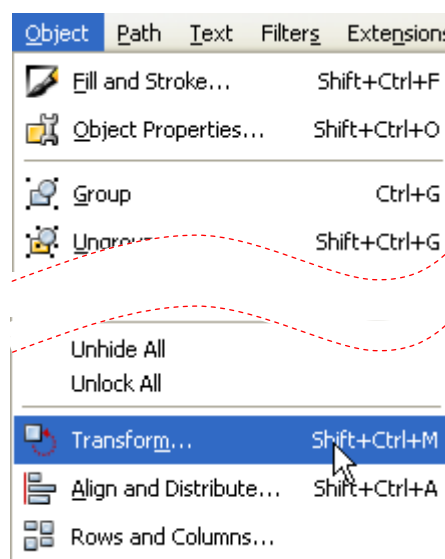
## 5.1 Dokładna transformacja obiektu (**Transform**)

Choć o wiele łatwiej jest dokonywać wszelkich przesunięć, obrotów i zmian skali „łapiąc” myszką za uchwyty obiektu, czasami trzeba wykonać transformację o dokładnie określone, ułamkowe wartości liczbowe. Taka sytuacja ma na przykład miejsce, gdy rysujesz tarcze przyrządów pokładowych. Na wielu z nich skala jest podzielona np. na 32 „kreski”. Oznacza to, że kąt pomiędzy kreskami skali musi wynosić dokładnie  $11.25^\circ$ . Takiego obrotu nie uzyskamy za pomocą skoku **Snap** (klawisz **Ctrl**). Najprościej to zrobić, używając panelu **Transform**.

Rysunek 5.1.1 pokazuje przygotowanie do takiej operacji. Narysowałem pustą tarczę przyrządu, a w jej środku umieściłem przecięcie linii pomocniczych — by jego środek sam przyciągał przesuwane obiekty. Następnie narysowałem „wzorcową”, pojedynczą kreskę skali. Przesunąłem jej środek do środka tarczy (bo wokół tego punktu będzie obracana). Następnie wywołałem polecenie **Object→Transform** (Rysunek 5.1.2):

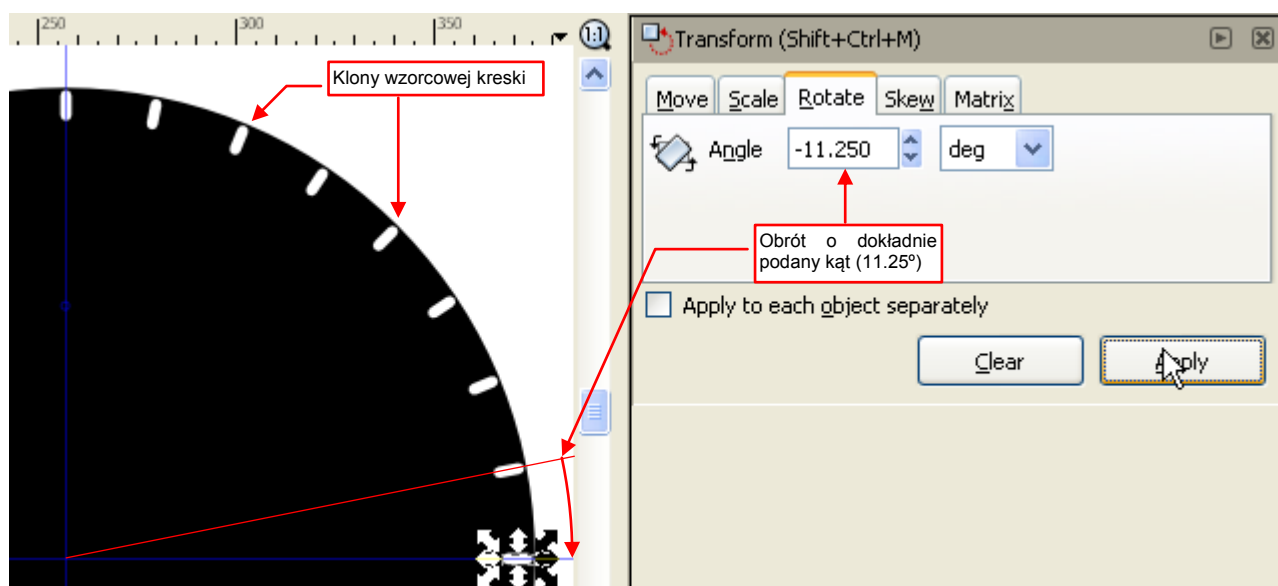


Rysunek 5.1.1 Przygotowanie obiektu do transformacji



Rysunek 5.1.2 Wywołanie polecenia

W panelu **Transform**, który w ten sposób otworzyłem, wpisałem w zakładce **Rotate**, polu **Angle**, dokładny kąt pomiędzy kreskami skali (Rysunek 5.1.3):



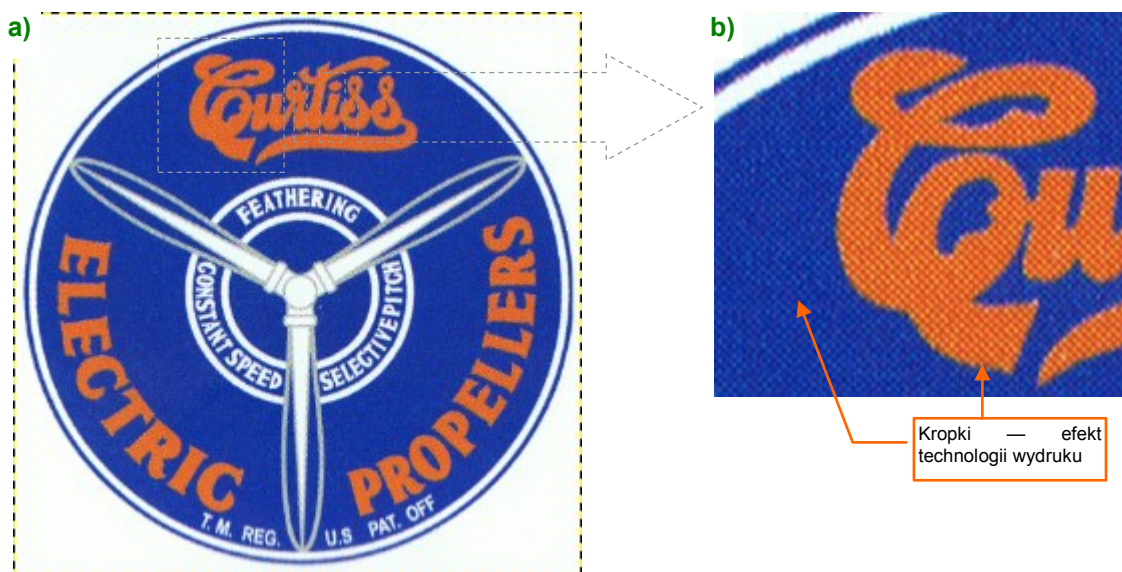
Rysunek 5.1.3 Wykonanie dokładnego obrotu

Teraz wystarczy ostatnią kreskę zaznaczyć, skopiować, i nacisnąć przycisk **Transform:Apply**. I tak po kolei, dopóki nie wypełnisz całego zakresu skali klonami „wzorca”.



## 5.2 Wektoryzacja bitmap

Czasami trzeba odwzorować na modelu jakieś skomplikowane godła lub inne oznaczenia. Na przykład — na każdej z łopat śmigła P-40 był naniesiony znak firmowy „Curtiss Electric”. Rysunek 5.2.1a) przedstawia „surowy” obraz, wycięty z zeskanowanej strony monografii:

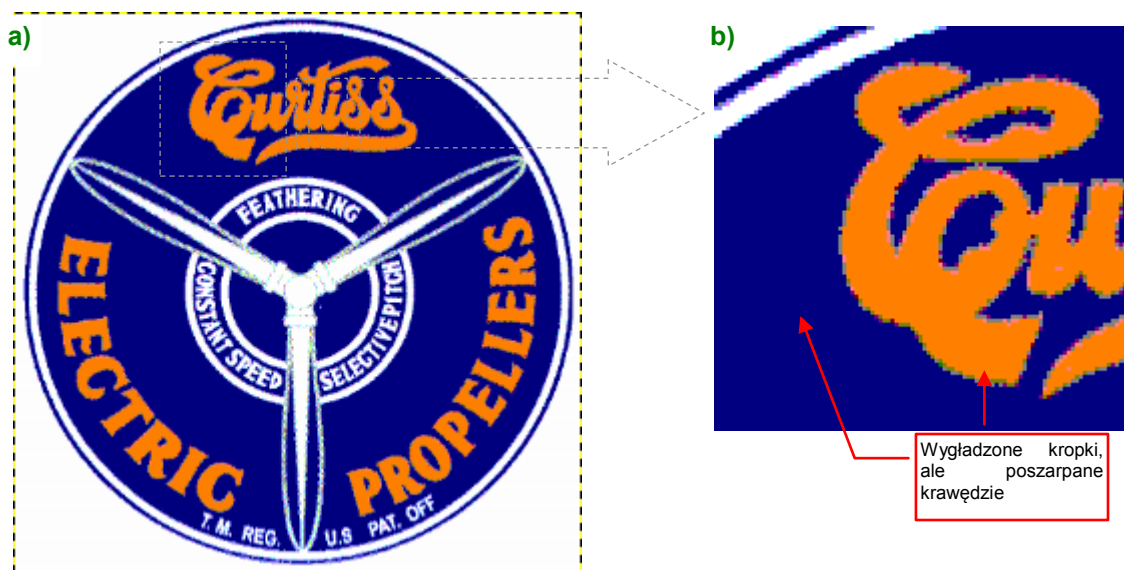


Rysunek 5.2.1 „Surowy” obraz naklejki z łopaty śmigła *Curtiss Electric*, zeskanowany z planszy barwnej

W takiej postaci obraz pokryty jest rastrem o nieco innym odcieniu (Rysunek 5.2.1b). Może to jest efekt druku w technologii offsetu? Aby te kropki wyeliminować, poddaj go „wstępnej obróbce” w GIMP:

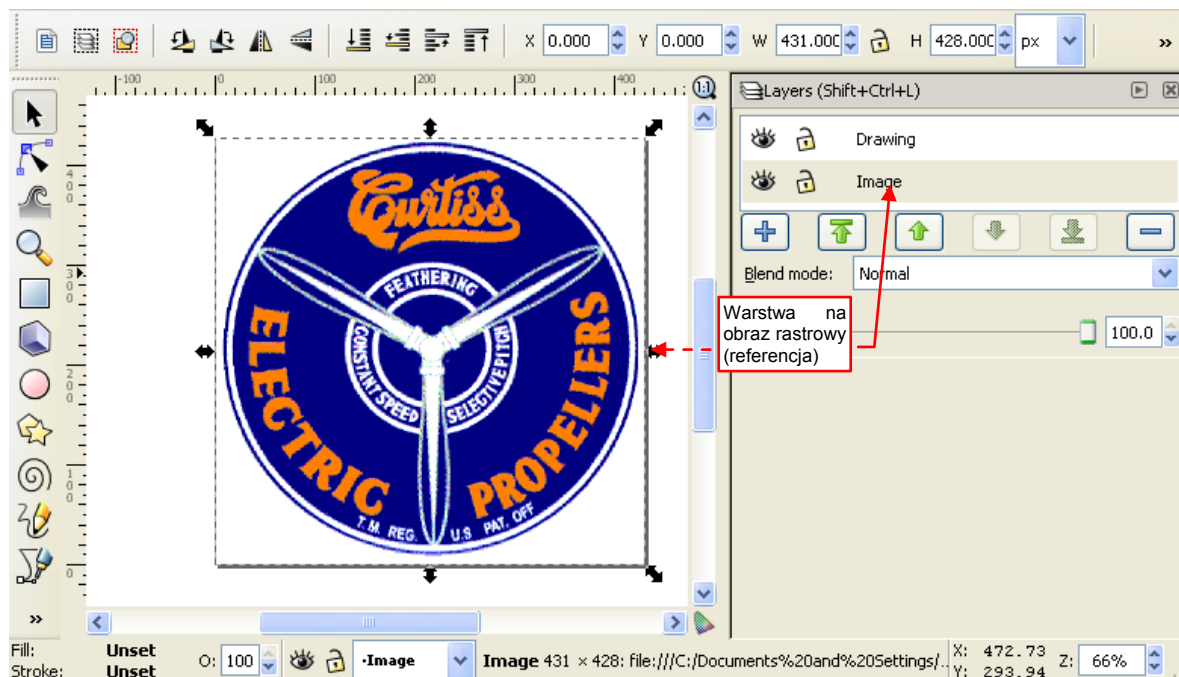
- **Image→Scale**, (zmniejsz obraz do 50%, z filtrem **Linear**) — to eliminuje część „kropek”;
- **Color→Posterize** (zmniejsz liczbę barw do 6);
- Wybrać (**Select By Color**) na napisie *Curtiss Electric Propellers* „kropki” innych kolorów niż pomarańczowy, i zamalować je na pomarańczowo (barwą pobraną z napisu);
- Powtórnie wywołaj **Color→Posterize** (zmniejszyć liczbę barw do 3);
- Wybrać (**Select By Color**) na napisie *Curtiss Electric Propellers* „kropki” innych kolorów niż pomarańczowy. Zamalować je barwą pomarańczową pobraną z napisu;

Powinieneś uzyskać w ten sposób uproszczony obraz, o liczbie barw zredukowanej do minimum (Rysunek 5.2.2):



Rysunek 5.2.2 Ten sam obraz, po ujednoliceniu barw (do trzech)

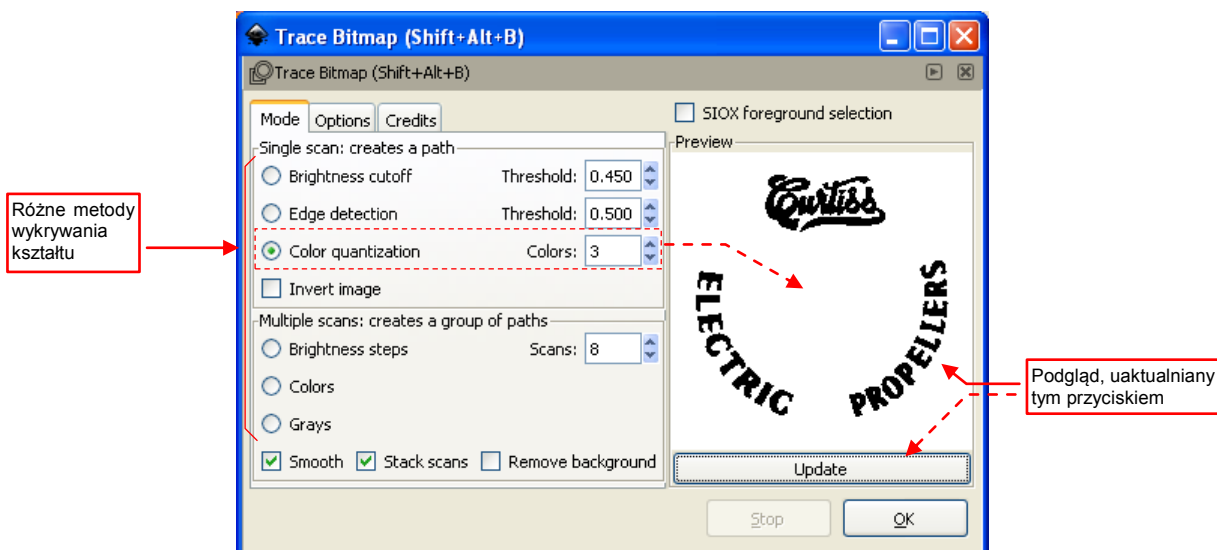
Tak przygotowany obraz załaduj do Inkscape (Rysunek 5.2.3):



Rysunek 5.2.3 Rastrowy wzorek, załadowany do Inkscape

Przyjrzyj się uważnie temu wzorcowi i zastanów, które szczegóły szybciej i dokładniej można narysować od podstaw w Inkscape, a które lepiej będzie uzyskać poprzez automatyczne odwzorowanie kształtu. (To polecenie nie działa idealnie — czasami pozostawia jakieś niewielkie przekłamania). Osobiście, patrząc na Rysunek 5.2.3, zdecydowałem się automatycznie przenieść tylko główny napis: „Curtiss Electric Propellers”. Szczególnie sam „Curtiss” jest bardzo fantazyjny i trudno byłoby go narysować od podstaw.

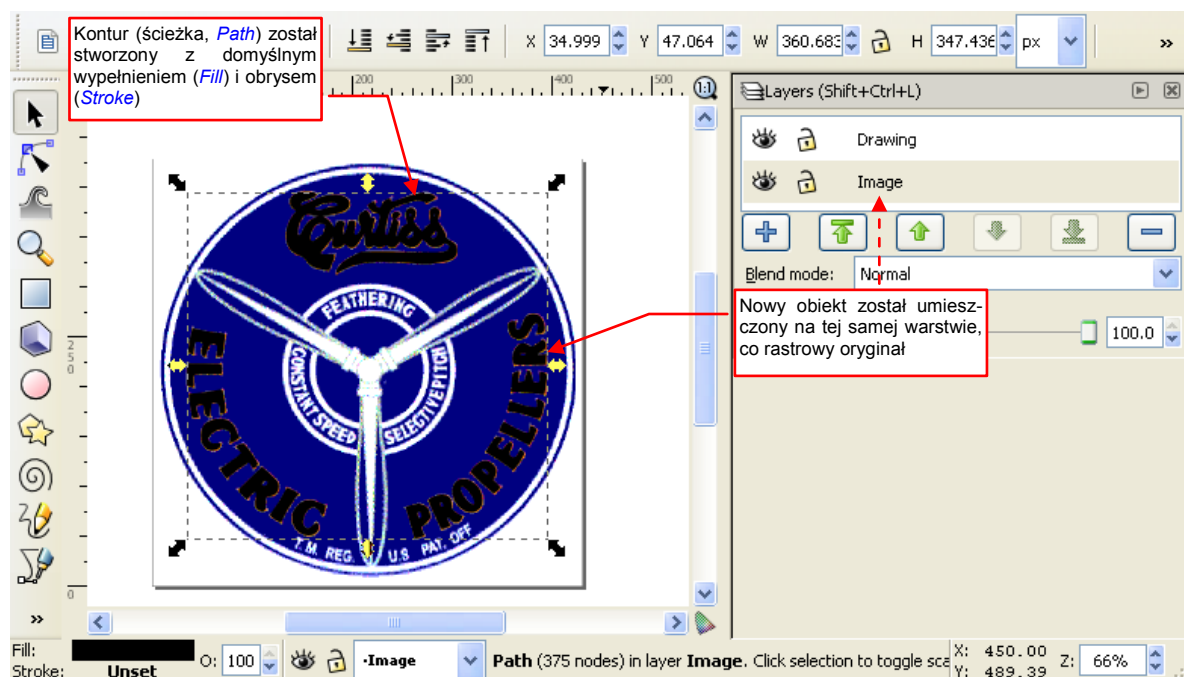
Aby stworzyć wektorową kopię obrazu rastrowego, zaznacz go na rysunku i wywołaj polecenie **Path→Trace Bitmap**. Spowoduje to pojawienie się okna dialogowego z opcjami wykrywania kształtu (Rysunek 5.2.4):



Rysunek 5.2.4 Okno dialogowe polecenia **Trace Bitmap**.

Możesz poeksperymentować chwilę z różnymi opcjami wykrywania kształtu i rezultatami ich działania, pokazwanymi obok, w obszarze podglądu (*Preview*). Przekonasz się, że każda z nich tworzy efekt znacznie różniący się od pozostałych. W tym konkretnym przypadku zdecydowałem się użyć metody **Color quantization**, z **Colors** = 3, gdyż z takim ustawieniem „wydobywa” z obrazu rastrowego sam napis „Curtiss Electric Propellers”. Niczego więcej nie potrzebuję, więc oszczędzę sobie w ten sposób usuwania zbędnych rezultatów wektoryzacji.

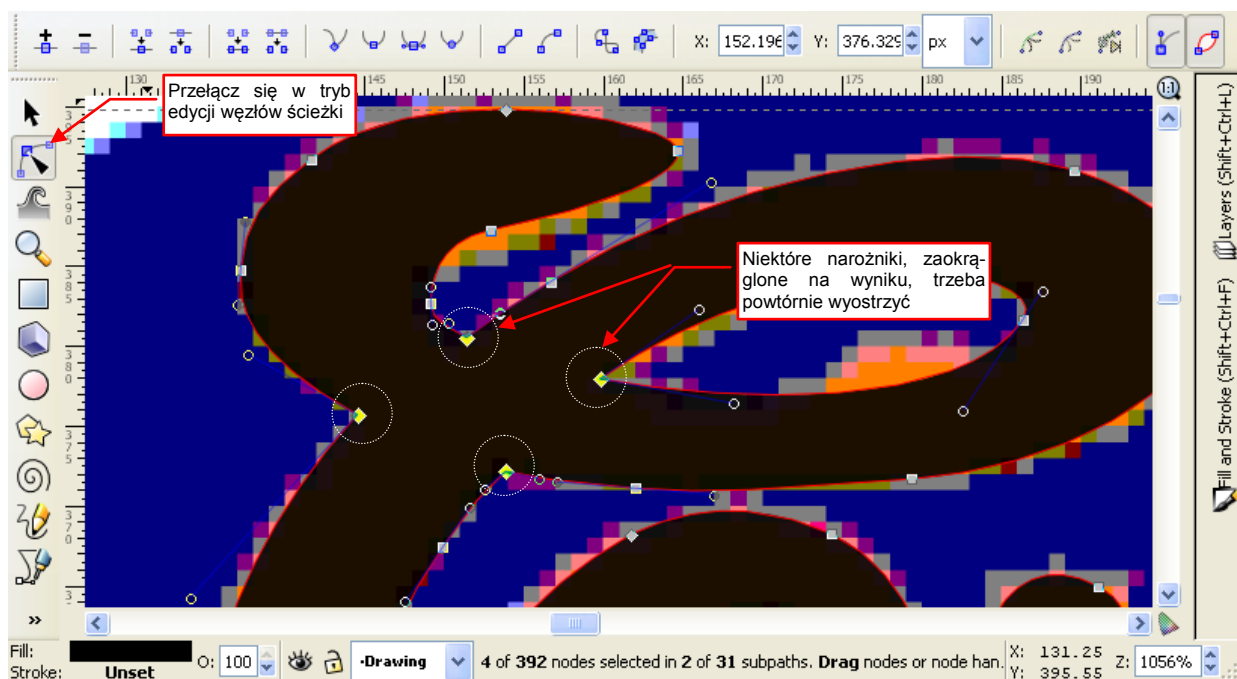
Po naciśnięciu w oknie *Trace Bitmap* przycisku *OK*, Inkscape stworzy nową ścieżkę, o konturach odpowiadających kształtom, które wykrył w obrazie rastrowym (Rysunek 5.2.5):



Rysunek 5.2.5 Rezultat polecenia *Trace Bitmap*.

Domyślnie nowy kontur powstał na tej samej warstwie, na której umieszczony jest rastrowy pierwowzór. Aby ułatwić sobie dalszą pracę, przenieś go na jakąś warstwę powyżej. Będziesz mógł wtedy łatwo manipulować przejrzystością wzorca.

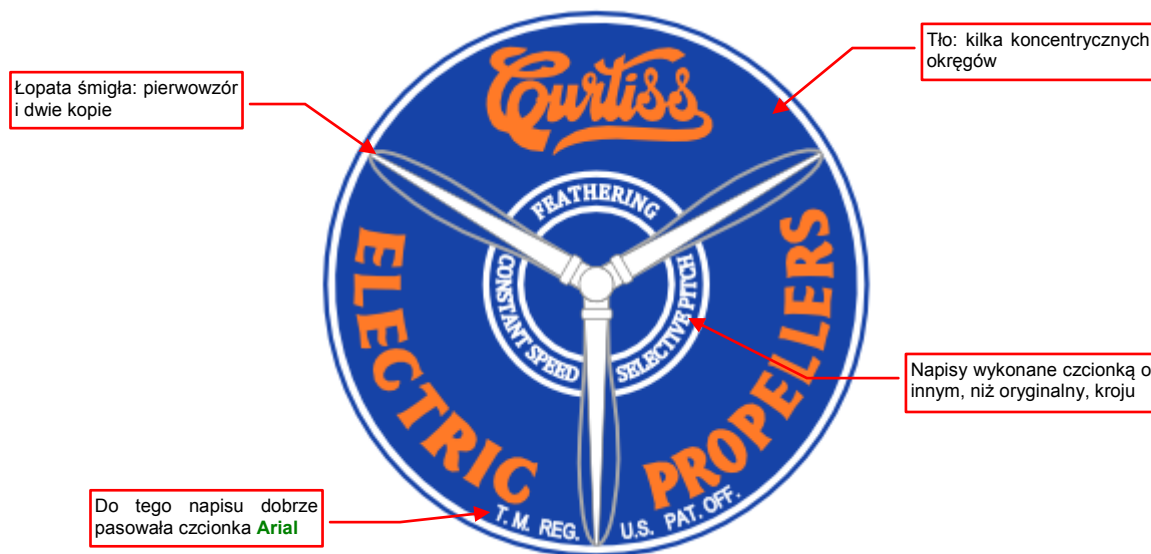
Ogólny kształt, uzyskany w wyniku polecenia *Trace Bitmap*, wygląda poprawnie. Gdy jednak wyłączysz na chwilę widoczność leżącego pod spodem wzorca, zaczniesz dostrzegać pewne różnice. Podczas śledzenia kształtu Inkscape ma tendencję do zaokrąglania wszelkich ostrych narożników i uskoków. Takie drobne zaokrąglenia składają się jednak na ogólny efekt. Warto więc skorygować nieco uzyskany kształt, nadając odpowiednim narożnikom z powrotem kąty ostre (Rysunek 5.2.6):



Rysunek 5.2.6 „Wyostrenie kątów” na ścieżce stworzonej poleceniem *Trace Bitmap*.

Tak jak wspomniałem wcześniej, pozostałe elementy obrazu prościej jest narysować od podstaw. Używając obrazu rastrowego jako „podkładki”, narysuj jedną łopatę śmigła, a następnie skopiuj ją i obróć o  $120^\circ$ , by uzyskać dwie pozostałe. Tło, wraz z wewnętrznymi kręgami, narysuj jako zespół kół, z których niektóre mają białe obwódki. Napis „T.M. REG. U.S. PAT. OFF.” bardzo ładnie można odwzorować za pomocą jakiejś prostej czcionki bezszeryfowej — nawet standardowej **Arial**.

Gorzej z białymi napisami: „CONSTANT SPEED”, „SELECTIVE PITCH”, „FEATHERING”. Użyto do nich bardzo podobnej czcionki, co w napisie „Electric Propellers”. Nie udało mi się znaleźć podobnej, a wektoryzacja tego kształtu nie wyszłaby zbyt dobrze — za dużo byłoby „babrania” się z poprawianiem kątów ostrych. Ostatecznie zdecydowałem się użyć innego kroju pisma. Choć nie jest takie samo, to na pierwszy (a może i drugi) rzut oka nie razi, szczególnie gdy nigdzie obok nie widać oryginału<sup>1</sup> (Rysunek 5.2.7):



Rysunek 5.2.7 Efekt finalny — całkowicie wektorowy obraz rastrowego pierwowzoru

<sup>1</sup> Swoją drogą — wygląda na to, że datujący się z początku XX w. krój markowego napisu „Coca-Cola” jest wykonany w bardzo popularnym w tym czasie stylu. Tyle, że inne firmy, używające w swoich logo podobnych czcionek, już zdążyły poznać.



---

## **Rozdział 6. Blender — ogólne**

W tym rozdziale przedstawione są zagadnienia, które dotyczą więcej niż jednego okna (edytora) Blendera lub więcej niż jednego trybu pracy.

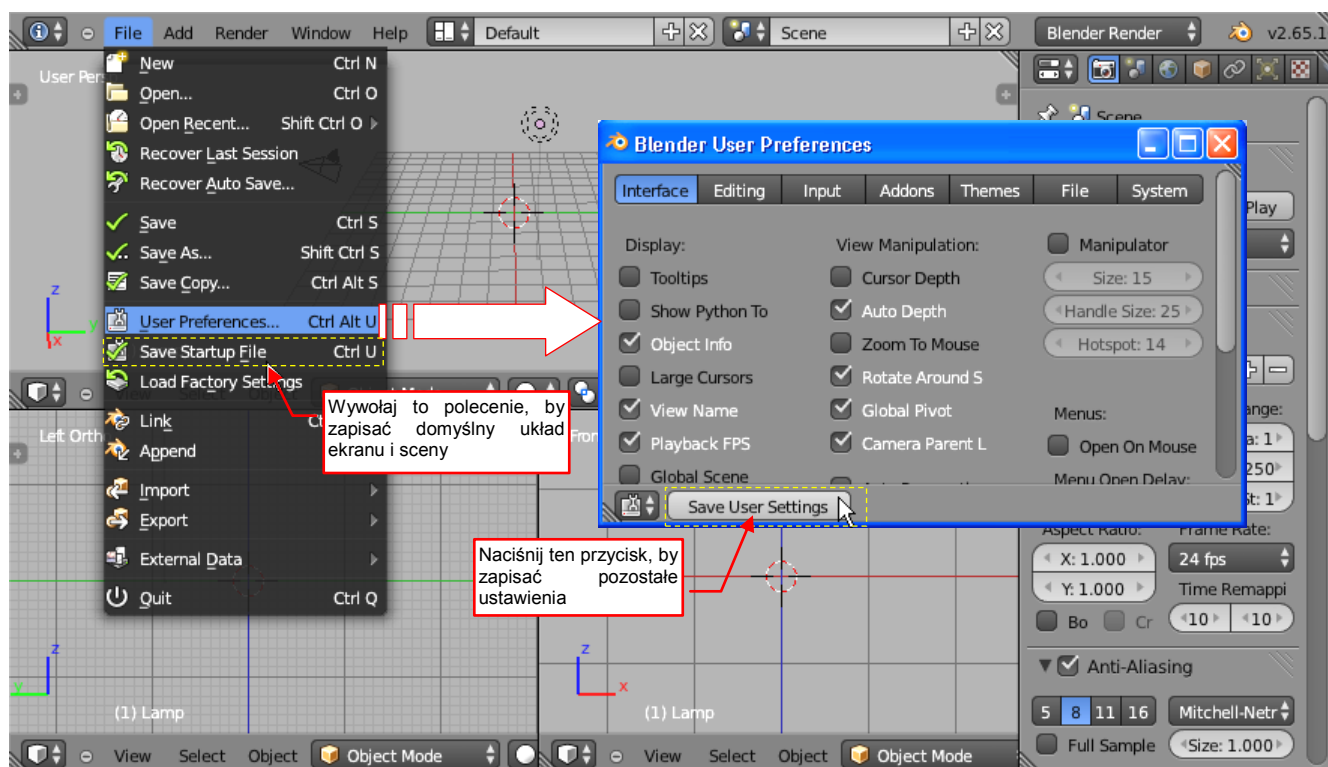
## 6.1 Ustawienie środowiska pracy

Szczegóły wyglądu i działania środowiska Blendera są wynikiem złożenia dwóch grup ustawień. Są to:

- ustawienia „ogólne” z okna *User Preferences*. Są one zapisywane do pliku o nazwie *userpref.blend* gdy naciśniesz w tym oknie przycisk *Save User Settings*. Oznacza to, że te ustawienia nie są związane z jakimkolwiek aktualnym rysunkiem. Traktuj je jako konfigurację Twojej instalacji Blendera;
- ustawienia „lokalne”, przechowywane w każdym pliku Blendera. Dotyczą one przede wszystkim interfejsu użytkownika: układ okien, alternatywnych układów ekranu (*screen layouts*). Gdy nie załadowałeś jeszcze żadnego pliku lub gdy tworzysz nowy plik poleceniem *File → New*, Blender wykorzystuje tzw. plik startowy, o nazwie *startup.blend*. (Stąd zawartość tego pliku decyduje, jak wygląda ekranu programu zaraz po uruchomieniu. Możesz ją nadpisać poleceniem *File → Save Startup File*);

W pliku *source.zip* (por. str. 16) umieściłem foldery z konfiguracją Blendera, której używałem pisząc tę książkę. Możesz je wgrać tak, jak jest to pokazane na str. 188.

Jeżeli jednak nie czujesz się w takich operacjach na folderach zbyt pewnie, istnieje inna metoda. Otwórz w Blenderze (*File → Open*) wzorcowy plik *source\config\startup.blend*. A potem w oknie *User Preferences* naciśnij przycisk *Save User Settings* i wywołaj polecenie *File → Save Startup File*, aby zachować także układ ekranu (Rysunek 6.1.1):



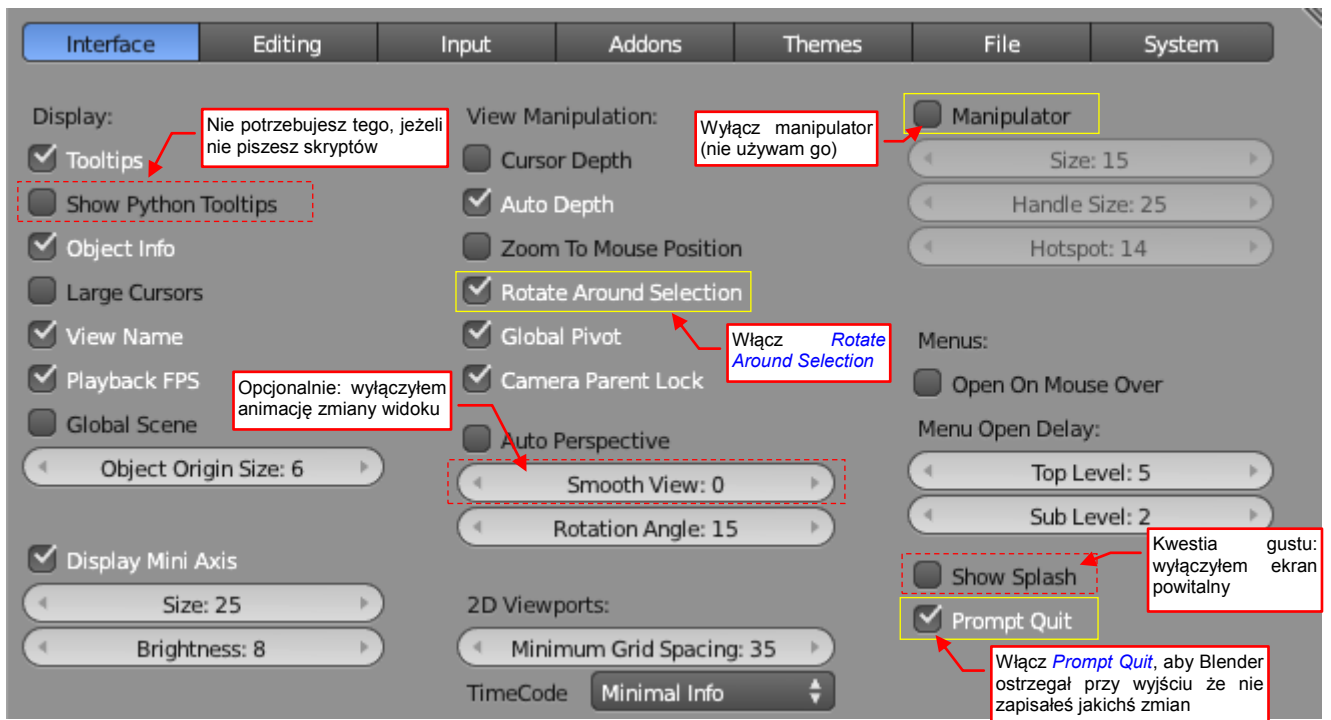
Rysunek 6.1.1 Przeniesienie ustawień z pliku *startup.blend*, dostarczonego z tą książką

I to wszystko! Kopia tego pliku, wraz z plikiem *userpref.blend* zostanie zapisana w Twoim profilu użytkownika jako aktualna konfiguracja Blendera. Dodatkowo możesz jeszcze zainstalować dodatki (*add-ons*), z których będziemy korzystać. (por. str. 201). Pliki tych dodatków umieściłem w katalogu *source\scripts\addons*.

Blender jest już skonfigurowany do dalszej pracy. Pierwszą różnicą, którą zapewne zauważyłeś, są jaśniejsze okna *3D View* (Rysunek 6.1.1). Domyślne, ciemne tło z Blendera 2.5/2.6 źle wyglądało na ilustracjach dla tej książki. Dlatego zdecydowałem się używać schematu barw, do którego przywykłem, używając Blendera 2.4.

Abyś jednak wiedział, co i dlaczego jeszcze zmieniłem, przejdę przez kolejne sekcje okna *User Preferences*. Pokażę na nich, jakie ustawienia standardowe zostały zmienione.

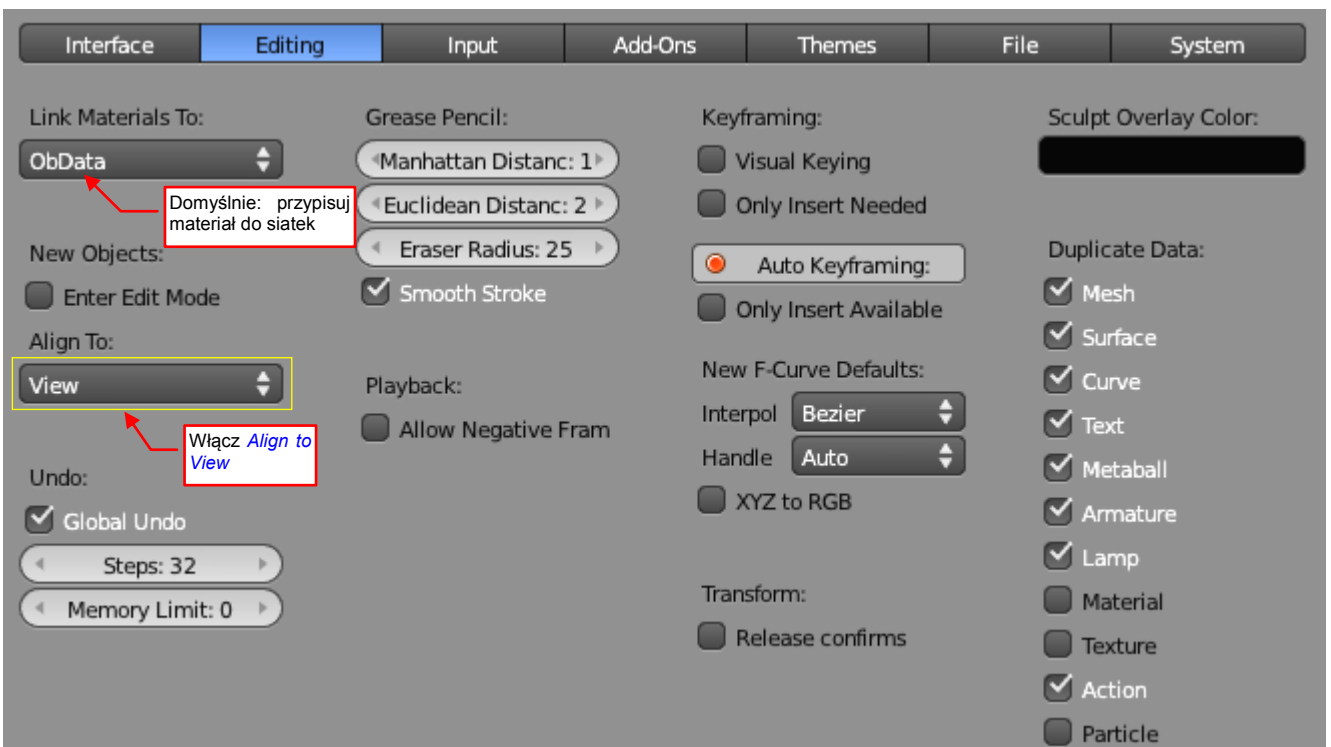
W sekcji **Interface** — włączyłem **View Manipulation: Rotate Around Selection** (Rysunek 6.1.2). To ułatwi obracanie widoku wokół detali. Oprócz tego wyłączyłem **Manipulator** (to taki pomocniczy „gadżet” w oknie **3D View**). Niektórzy uważają, że ułatwia manipulację obiektami w przestrzeni trójwymiarowej. Ja jednak należę do „starej szkoły” Blendera, która takiego udogodnienia nie знаła:



Rysunek 6.1.2 Ustawienia Blendera — sekcja **Interface**

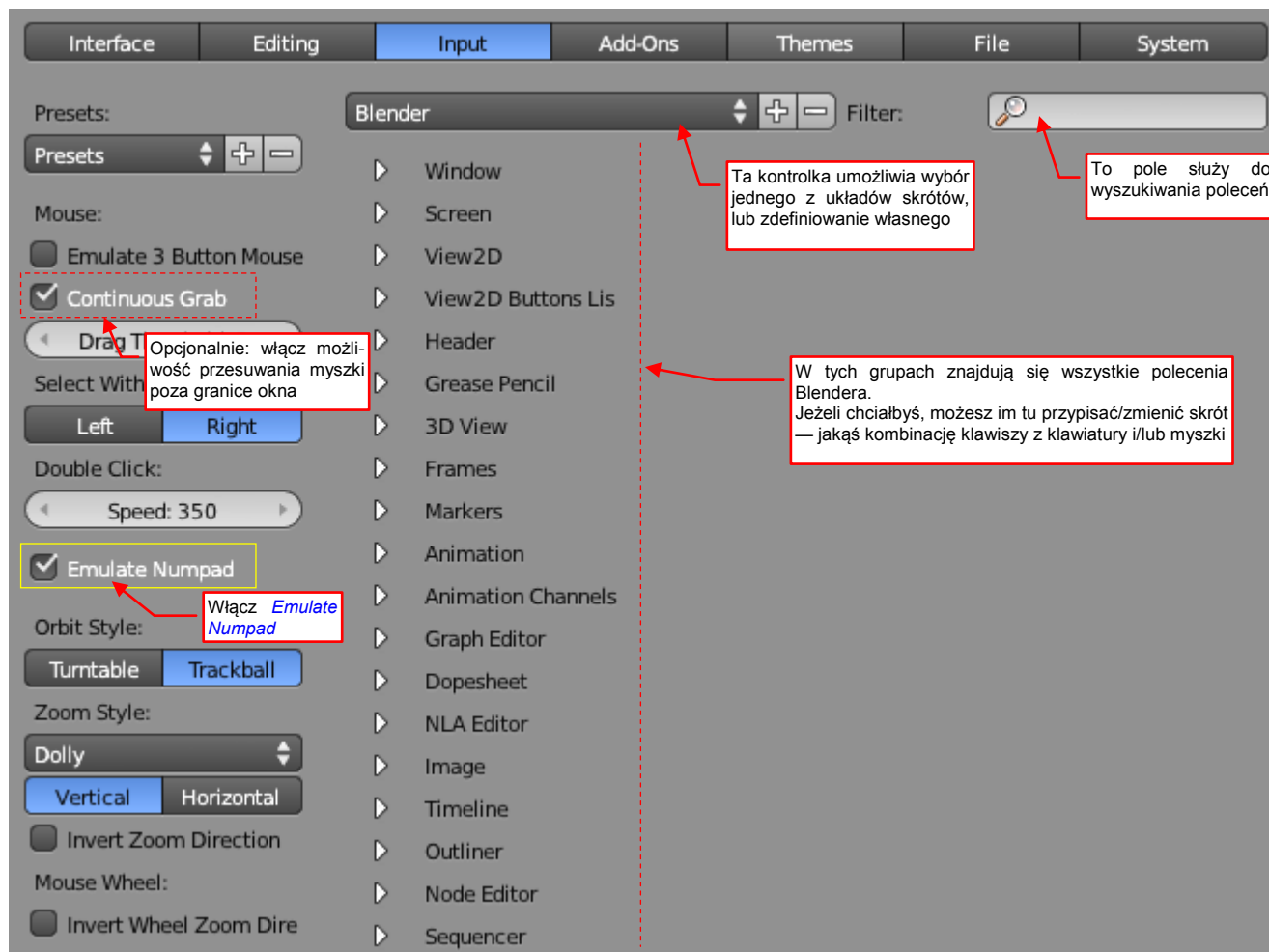
Jak pokazuje to Rysunek 6.1.2, w sekcji **Interface** możesz opcjonalnie przełączyć jeszcze parę innych opcji.

W sekcji **Editing** — włączyłem **New Objects: Align to: View** (Rysunek 6.1.3). Dzięki temu każdy obiekt, który stworzysz, powstanie w płaszczyźnie równoległej do aktualnego widoku. (Inaczej, aby odpowiednio zorientować nowo powstały element, musiałbyś go dodatkowo obracać).



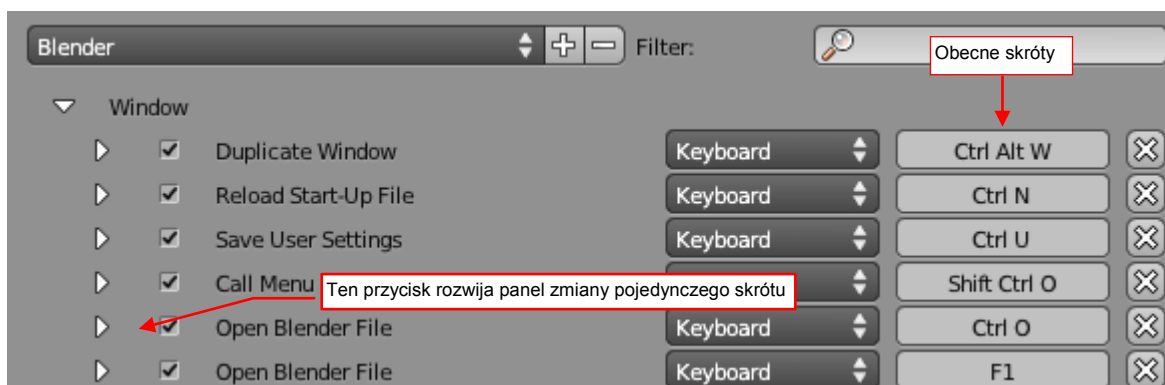
Rysunek 6.1.3 Ustawienia Blendera — sekcja **Editing**

W sekcji **Input** — włączyłem opcję **Emulate Numpad** (Rysunek 6.1.4). Choć jest to praktyczne przede wszystkim dla notebooków, może okazać się przydatne także na tradycyjnych komputerach. (Pozwoli utrzymać lewą rękę ponad centrum klawiatury, a nie przesuwając ponad całą jej szerokością).



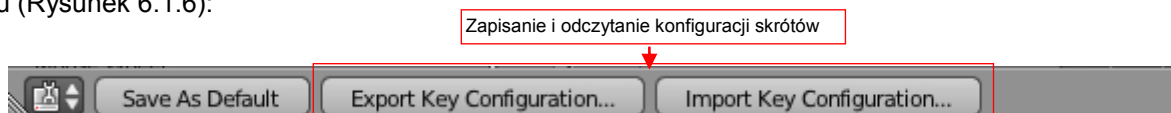
Rysunek 6.1.4 Ustawienia Blendera — sekcja **Input**

Zwróć uwagę, że w tej sekcji możesz zmienić konfigurację skrótów poleceń Blendera (Rysunek 6.1.5):



Rysunek 6.1.5 sekcja **Input** — rozwinięta grupa poleceń **Window**

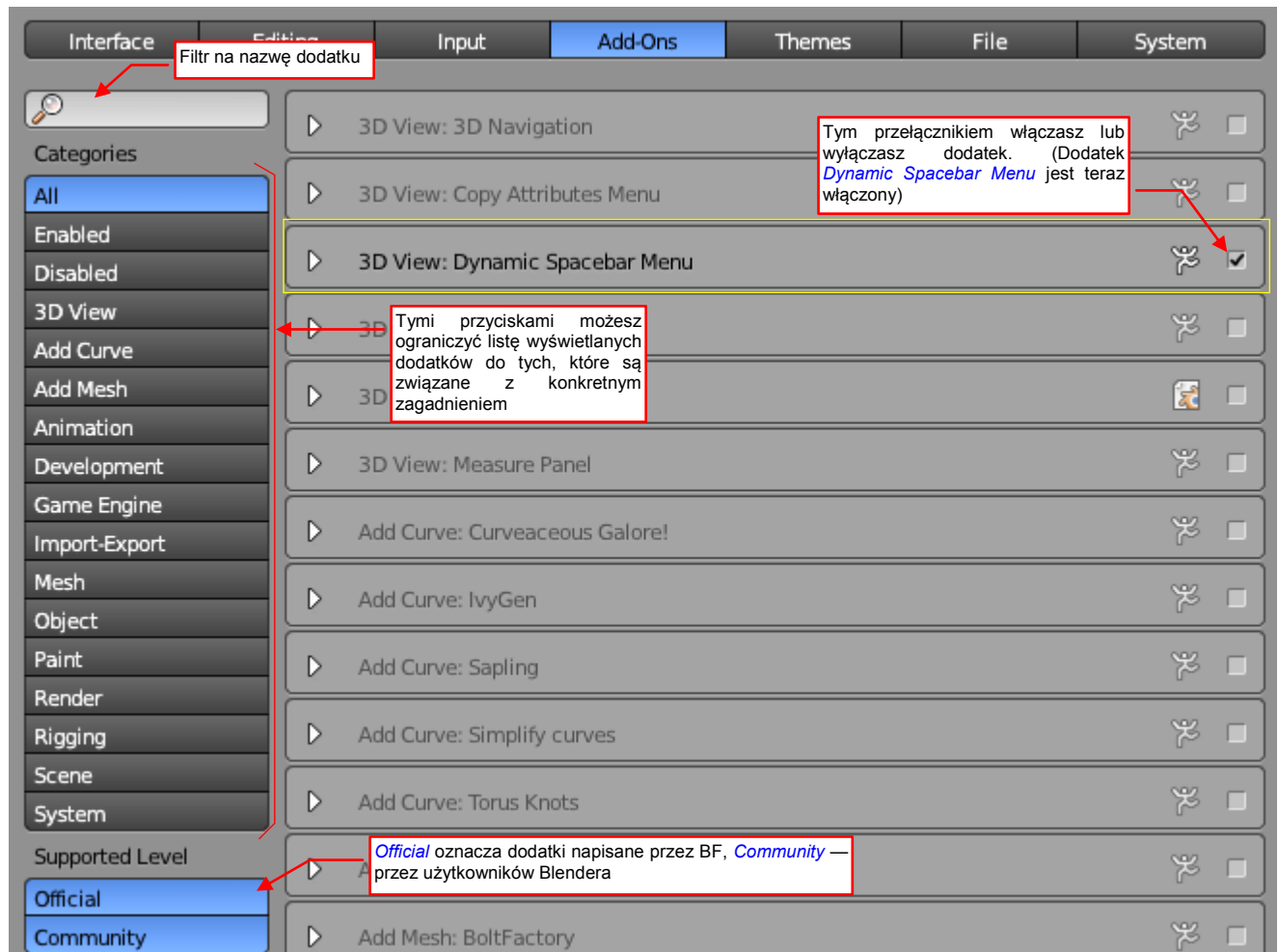
Dwa dodatkowe klawisze w nagłówku okna tej sekcji pozwalają także zapisywać i odczytywać skróty do poleceń z pliku (Rysunek 6.1.6):



Rysunek 6.1.6 Przyciski w nagłówku okna w sekcji **Input**

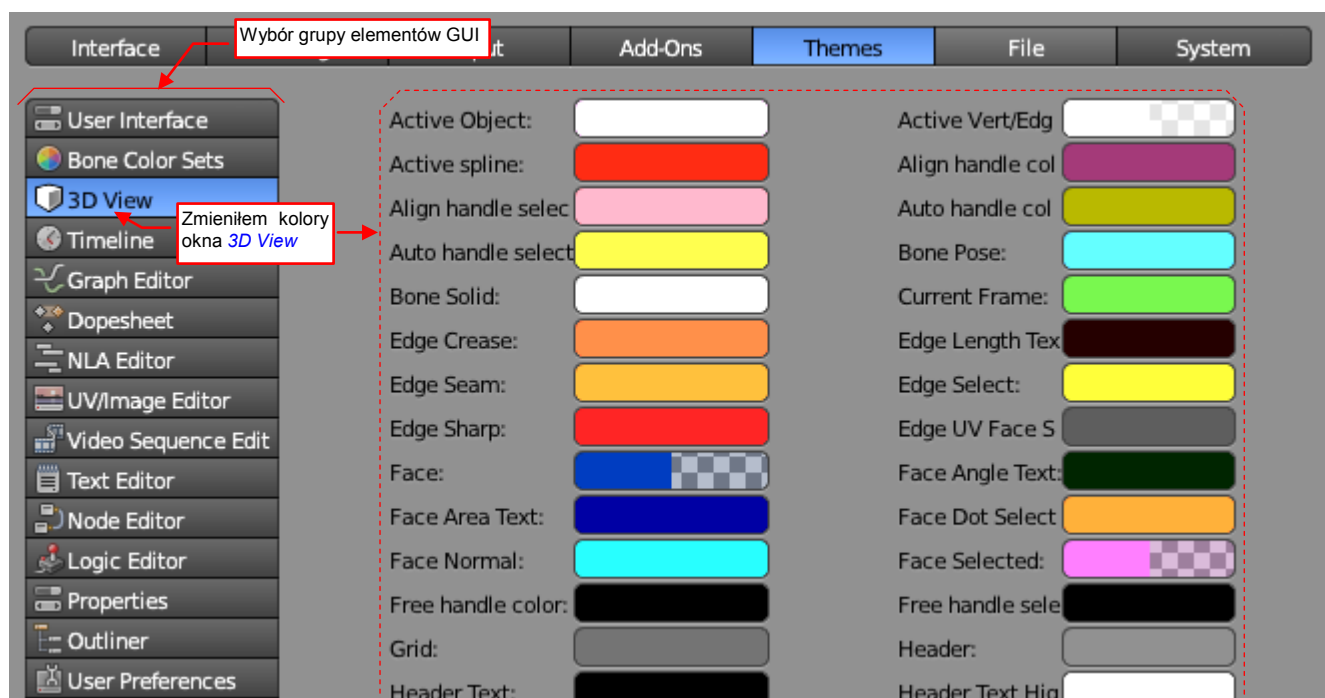


W sekcji **Add-Ons** (programowe rozszerzenia Blendera) włączyłem **Dynamic Spacebar Menu** (Rysunek 6.1.7):



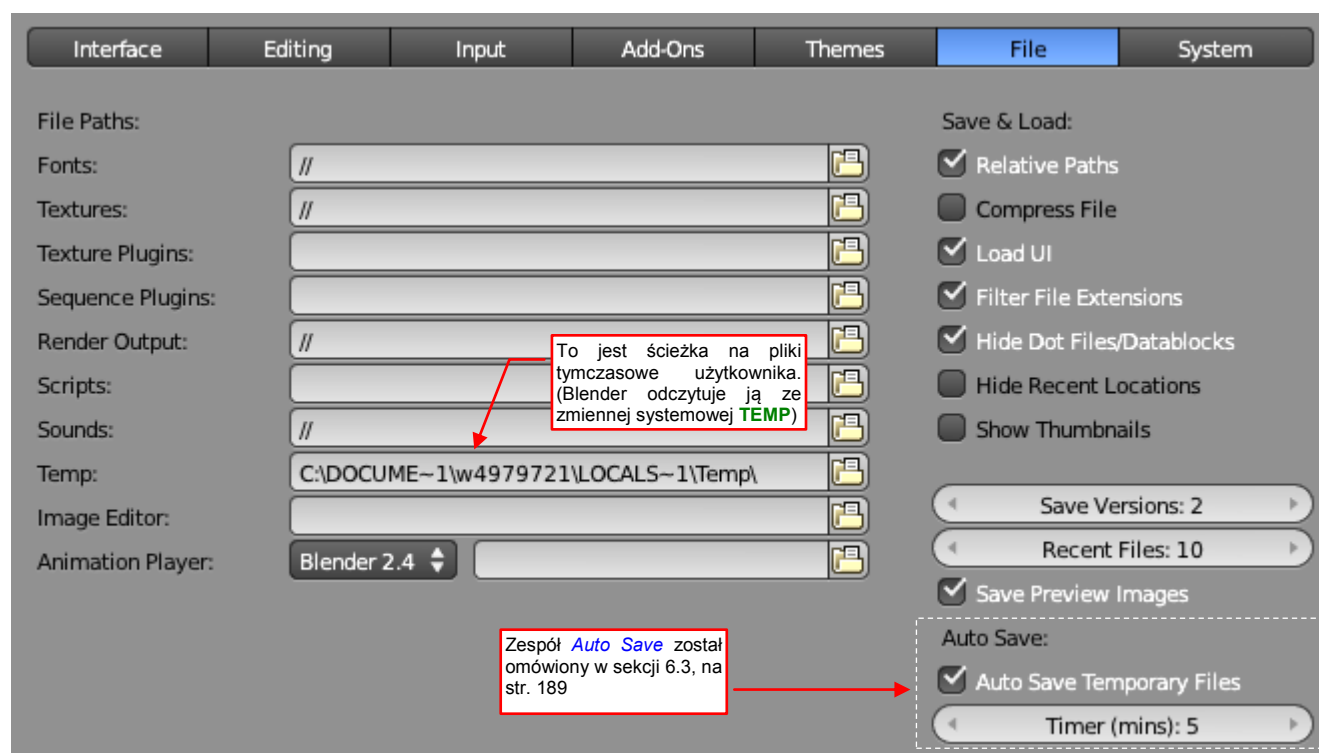
Rysunek 6.1.7 Ustawienia Blendera — sekcja **Add-Ons**

W sekcji **Themes**, jak już o tym wspominałem, pozmieniałem kolory widoku **3D View** (Rysunek 6.1.8):



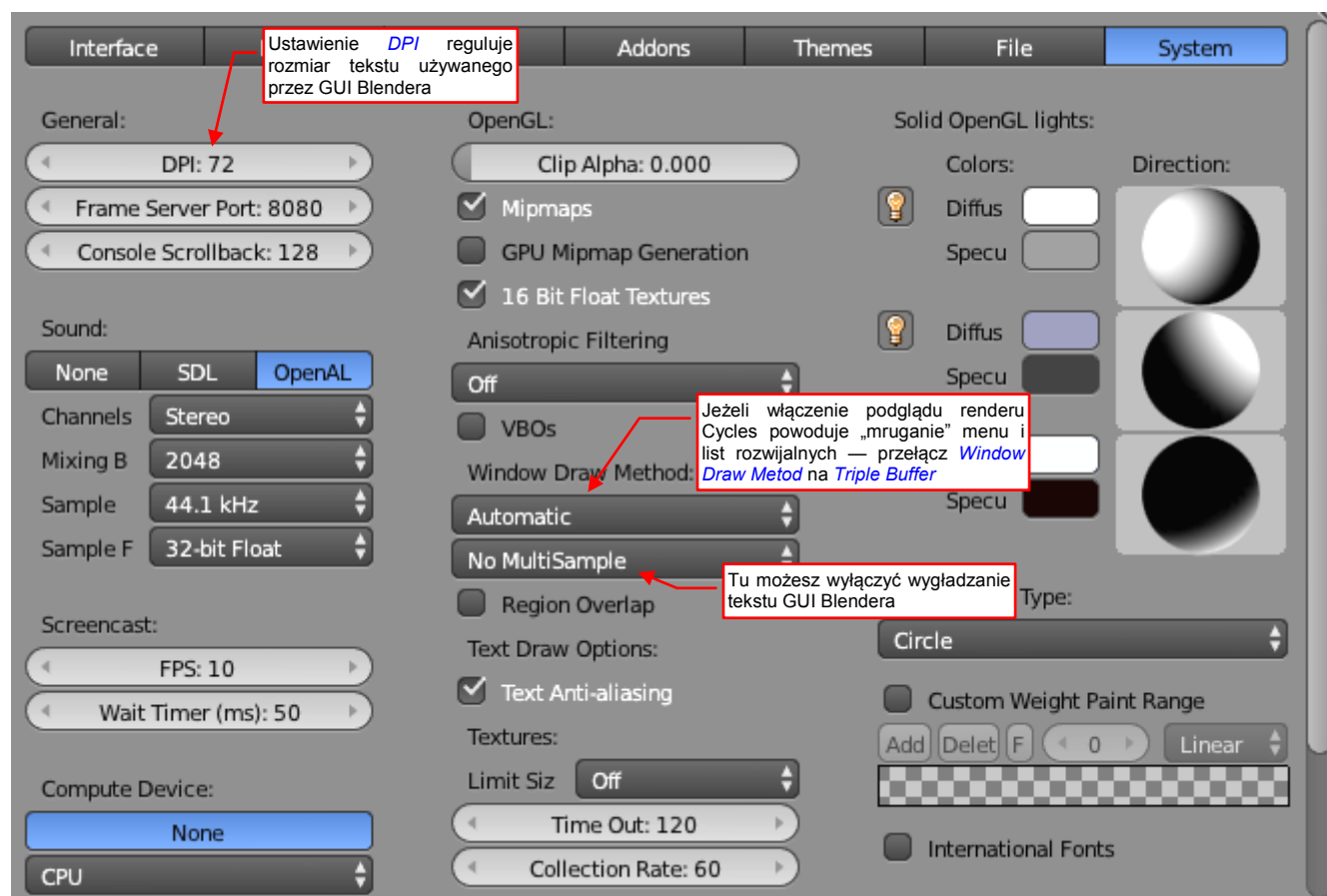
Rysunek 6.1.8 Ustawienia Blendera — sekcja **Themes**

Sekcja **File** zawiera domyślne ścieżki, na których Blender ma poszukiwać różnych rodzajów plików. Niczego w niej nie musiałem zmieniać. (Rysunek 6.1.9):



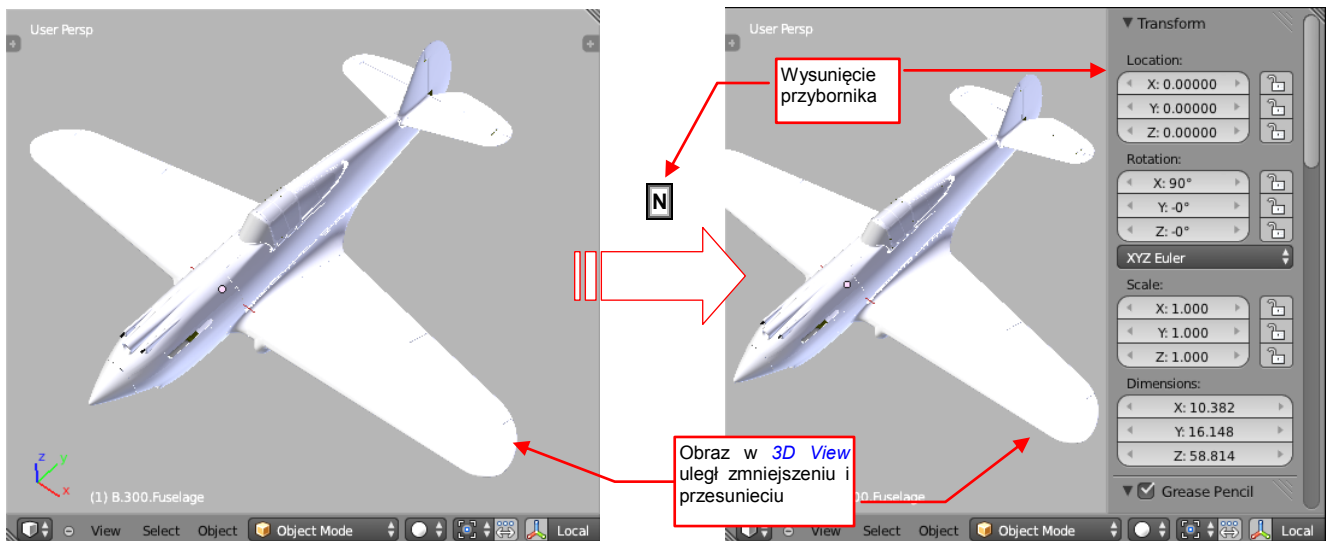
Rysunek 6.1.9 Ustawienia Blendera — sekcja **File**

Wreszcie sekcja **System** (Rysunek 6.1.10). Tu także niczego nie zmieniałem:



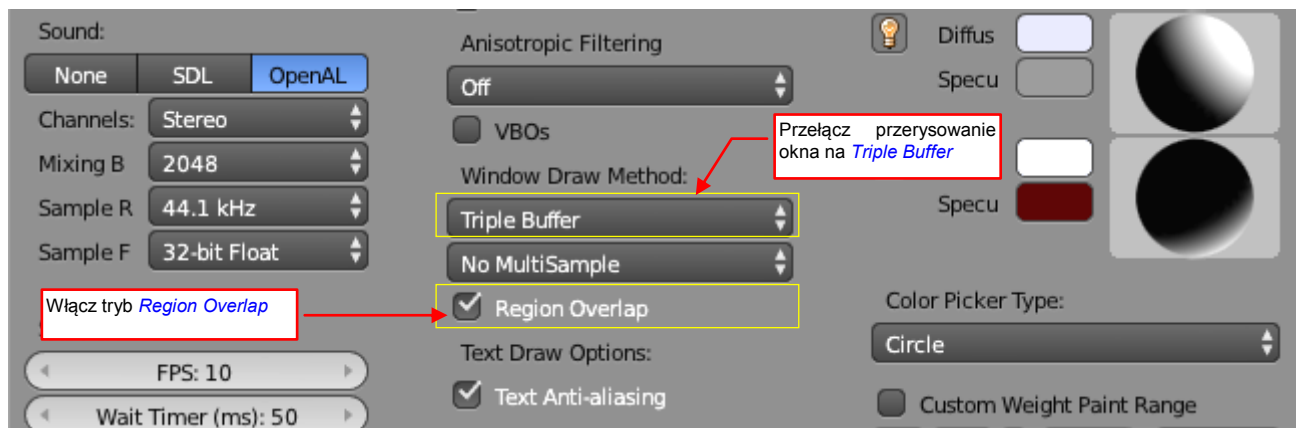
Rysunek 6.1.10 Ustawienia Blendera — sekcja **System**

Przy okazji omawiania sekcji **System** wspomnę o pewnym udogodnieniu. W domyślnej konfiguracji Blendera wysunięcie przybornika z boku okna **3D View** powoduje zmniejszenie wyświetlanego obrazu (Rysunek 6.1.11):



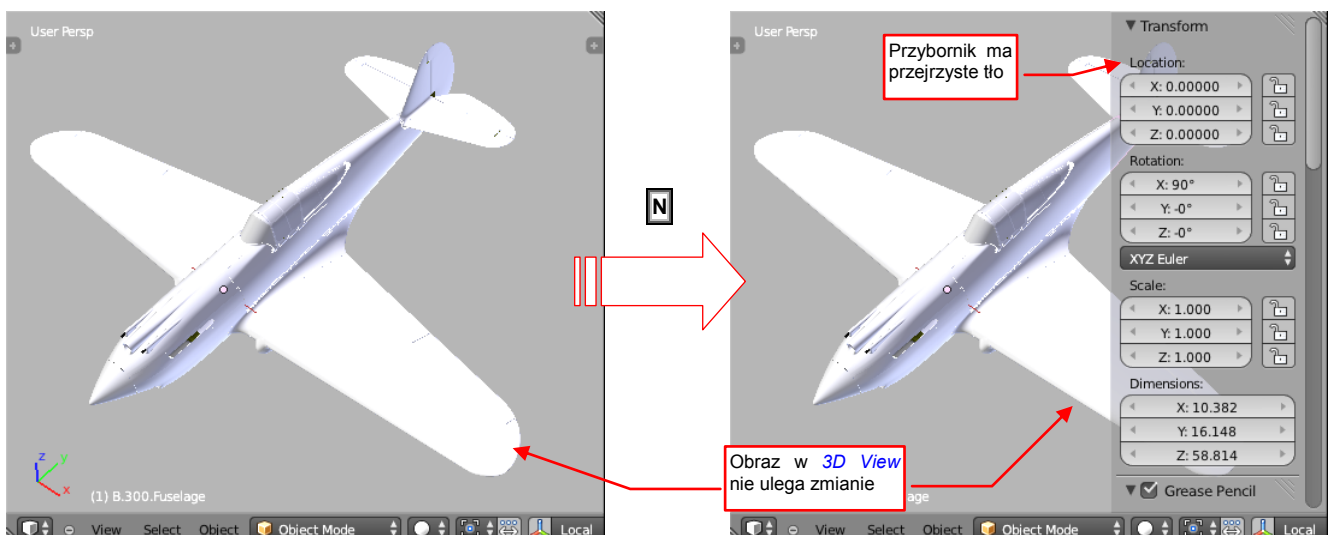
Rysunek 6.1.11 Pomniejszanie obrazu przy otwarciu przybornika

Mówiąc szczerze, niezbyt lubię ten efekt. Na szczęście w Blenderze 2.66 pojawiła się alternatywa: zaznacz w zakładce **System** opcję **Region Overlap** i przełącz **Window Draw Method** na **Triple Buffer** (Rysunek 6.1.12):



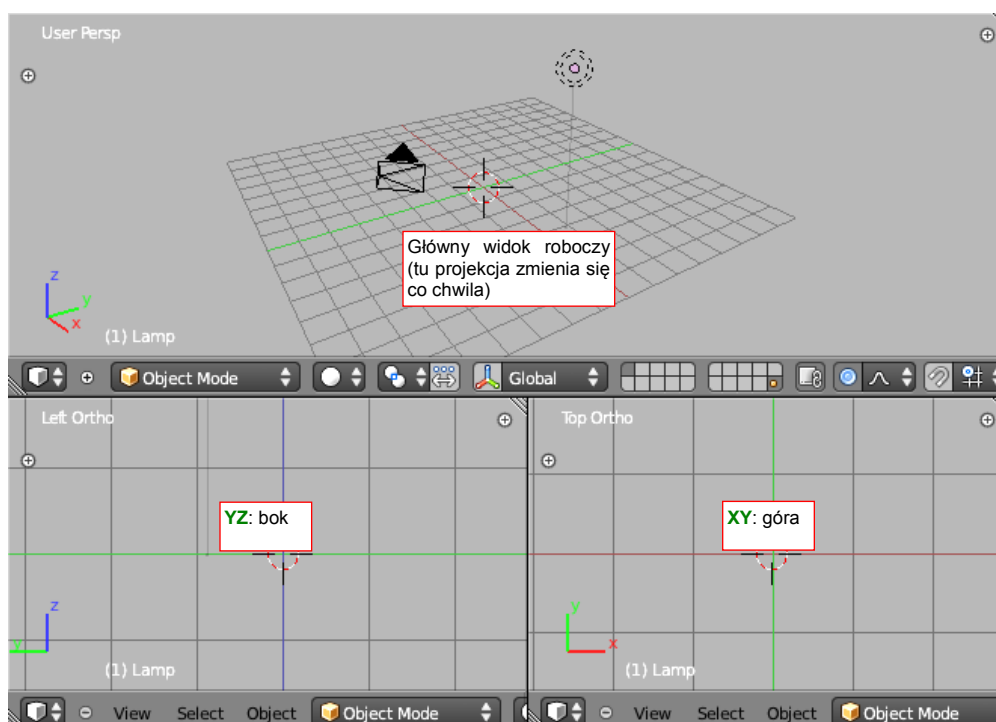
Rysunek 6.1.12 Zmiana parametrów w sekcji **User Preferences: System**

W rezultacie tło przyborników stało się półprzezroczyste, i teraz „wysuwają się” ponad obrazem (Rysunek 6.1.13):



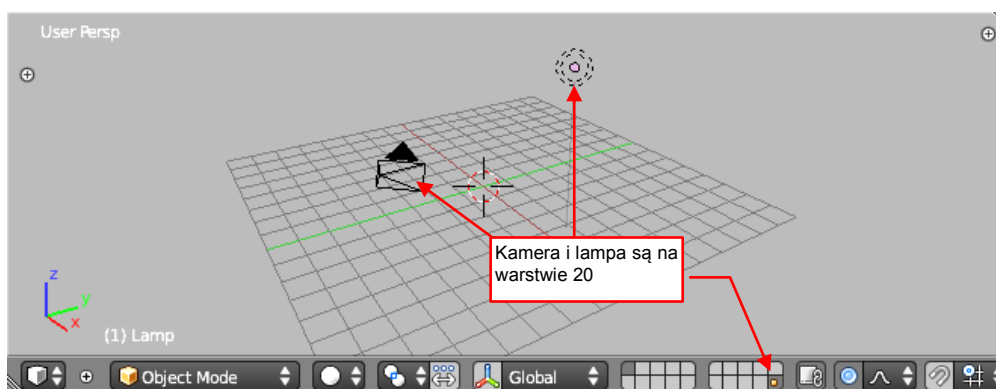
Rysunek 6.1.13 Półprzezroczyste tło przybornika, zachowujące oryginalny rozmiar obrazu

W stosunku do oryginalnej wersji Blendera, wprowadziłem także parę zmian w domyślnym układzie ekranu. Nie było to niezbędne, ale często łapałem na tym, że ciągle w ten sam sposób poprawiam zawartość nowo utworzonego rysunku. Po co? Zapisalem raz poprawiony plik jako domyślny, i w ten sposób oszczędziłem sobie ciągłej roboty (Rysunek 6.1.14):



Rysunek 6.1.14 Trzy widoki robocze

Zwróć uwagę, że usunąłem ze sceny "pudełko", które Blender domyślnie umieszcza pośrodku sceny. (Zawsze pierwszą rzeczą, którą robiłem, było pozbycie się tego obiektu). Przesunąłem także obiekty związane wyłącznie z renderingiem — kamerę i lampę — na warstwę 20 (Rysunek 6.1.15):



Rysunek 6.1.15 Warstwa obiektów związanych z renderingiem: kamery i lampy

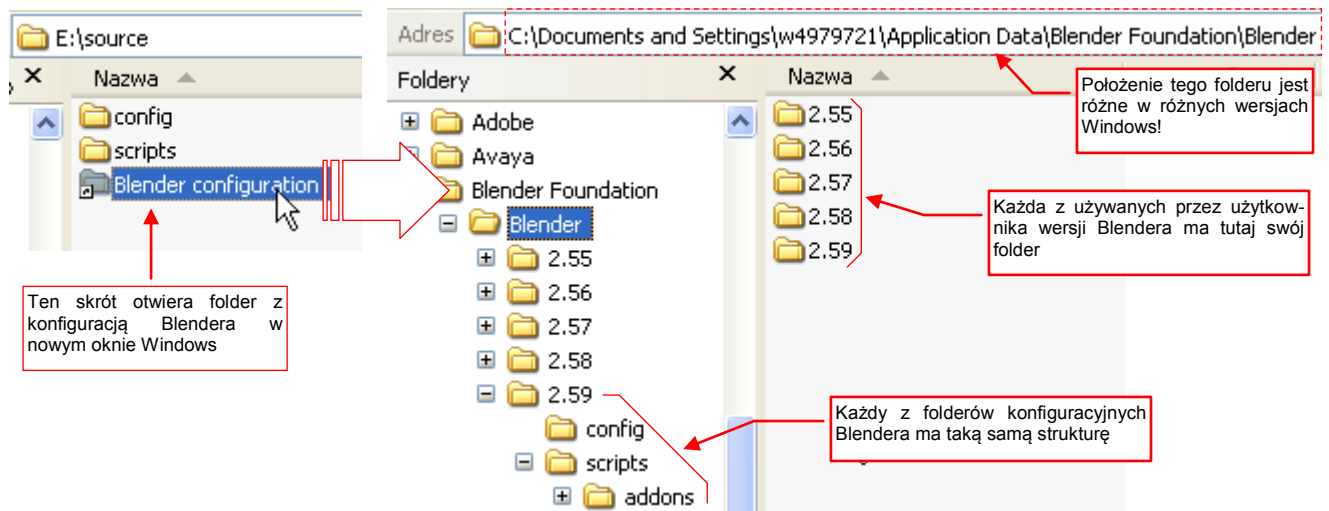
To taka moja konwencja — oczywiście, możesz to sobie zorganizować inaczej.



## 6.2 Pliki konfiguracji Blendera

Gdy choć raz wywołasz w Blenderze polecenie *File → Save Startup File*, lub naciśniesz w oknie *User Preferences* przycisk *Save User Settings*, Blender zapisze w Twoim profilu użytkownika zmienioną konfigurację. Stosunkowo trudno znaleźć folder, zawierający te dane. Jest tworzony (przynajmniej w Windows) w ukrytym katalogu *Application Data*. W różnych wersjach Windows *Application Data* znajduje się w różnych miejscach profilu użytkownika. W dodatku Eksplorator Windows w każdej wersji językowej systemu wyświetla nazwę tego folderu w lokalnym języku (podobnie jak *My Documents*) co pogłębia zamieszanie. (Nawet gdybym podał ścieżkę do tego folderu dla każdej wersji Windows, część użytkowników i tak by go nie znalazła).

Zamiast tego przygotowałem w pliku *source.zip* (por. str. 16) skrót *Blender configuration*. Wykorzystuje zmienną środowiskową Windows (**APPDATA**) by otworzyć w Eksploratorze Windows Twój folder z konfiguracją Blendera (Rysunek 6.2.1):

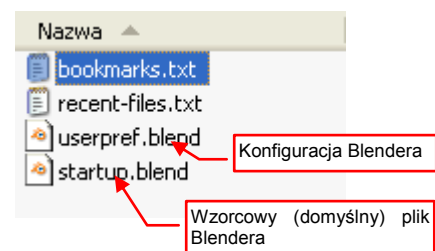


Rysunek 6.2.1 Otwarcie folderu na konfigurację Blendera

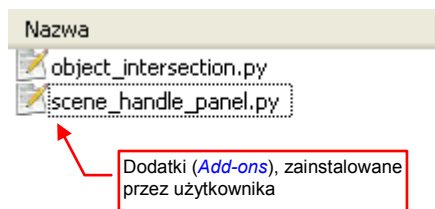
Każda z wersji Blendera przechowuje tu swoje pliki w oddzielnym katalogu (Rysunek 6.2.1). Dzięki temu możesz z nich korzystać jednocześnie (np. aby sprawdzić, jak coś działało poprzednio). Gdy instalujesz nową wersję programu, przy pierwszym uruchomieniu Blender proponuje skopiowanie konfiguracji z wersji poprzedniej (Rysunek 1.1.2).

Wewnątrz katalogu każdej wersji znajdują się dwa podfoldery:

- **config**: zawiera dwa ważne pliki (Rysunek 6.2.2). Pierwszy z nich nazywa się *userpref.blend*. Są w nim przechowywane wszelkie ustawienia z okna *User Preferences*, zapisane przyciskiem *Save User Settings*. Drugim plikiem jest *startup.blend*. Zawartość tego pliku Blender wyświetla bezpośrednio po uruchomieniu, lub gdy wybierzesz polecenie *File → New*. Plik *startup.blend* determinuje m.in. domyślny układ ekranu Blendera. Możesz go zastąpić jakimś bieżącym rysunkiem, wywołując polecenie *File → Save Startup File*.
- **scripts\addons**: w podfolderze *addons* Blender przechowuje dodatki (*add-ons*) zainstalowane przez użytkownika (Rysunek 6.2.3, por. także str. 201). Dzięki temu są przenoszone z całą konfiguracją przy aktualizacji do nowej wersji programu.



Rysunek 6.2.2 Zawartość subfolderu *config*



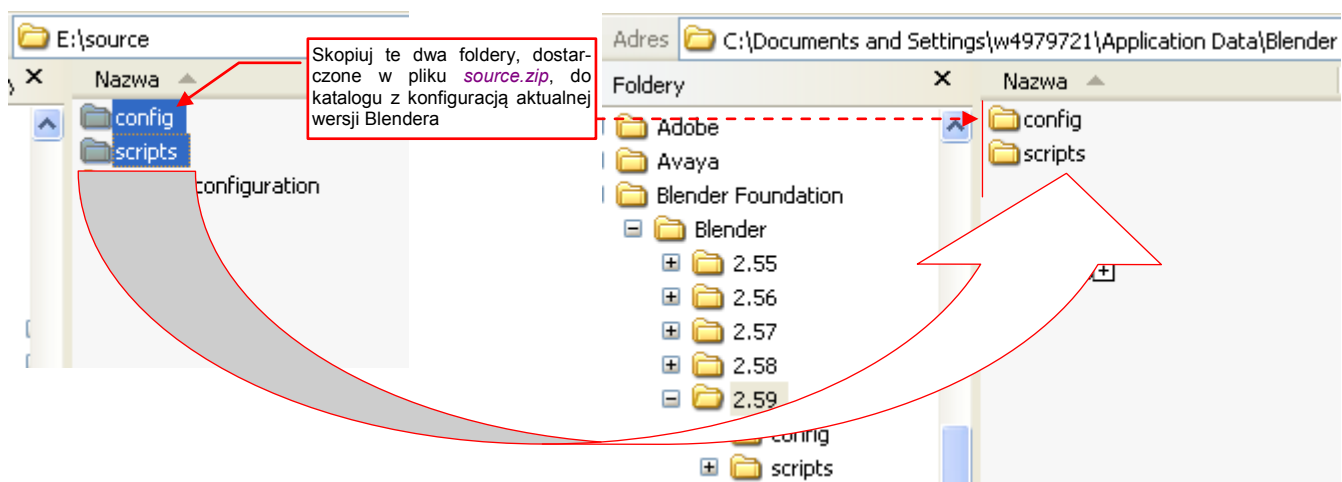
Rysunek 6.2.3 Zawartość *scripts\addons*

W operowaniu konfiguracją Blendera nie ma żadnej „ukrytej magii”. Wystarczy zmieniać odpowiednie pliki. Na przykład, instalację grupy dodatków (*add-on*), możesz wykonać na dwa sposoby:

- łatwy: używając GUI Blendera (tak jak to jest opisane na str. 201) — ale wtedy musisz wgrywać po kolei każdą wtyczkę;
- zaawansowany: jednym ruchem skopiować wszystkie pliki wtyczek do podkatalogu *scripts\addons\* folderu z konfiguracją Blendera. Wymaga to tylko wiedzy, gdzie ten folder się znajduje, i za pierwszym razem należy po prostu tam założyć takie podkatalogi;

Podobnie możesz się przełączać pomiędzy różnymi konfiguracjami Blendera — wystarczy przechowywać pod inną nazwą lub w innym miejscu alternatywne pliki *userpref.blend* i *startup.blend*. Gdy skopiujesz do folderu *config\* ich nowe wersje — Blender posłuży się nimi przy najbliższym uruchomieniu.

Abyś mógł podążać za przykładami, umieszczonymi w tej książce, w pliku *source.zip* dostarczyłem podkatalogi *config* i *scripts* z odpowiednią konfiguracją Blendera. Na wszelki wypadek najpierw skopiuj wszystkie pliki z Twoją oryginalną konfiguracją w jakieś inne miejsce. Potem po prostu przenieś odpowiednie te dwa podkatalogi z *source.zip* do folderu z konfiguracją Twojej aktualnej wersji Blendera (Rysunek 6.2.4):

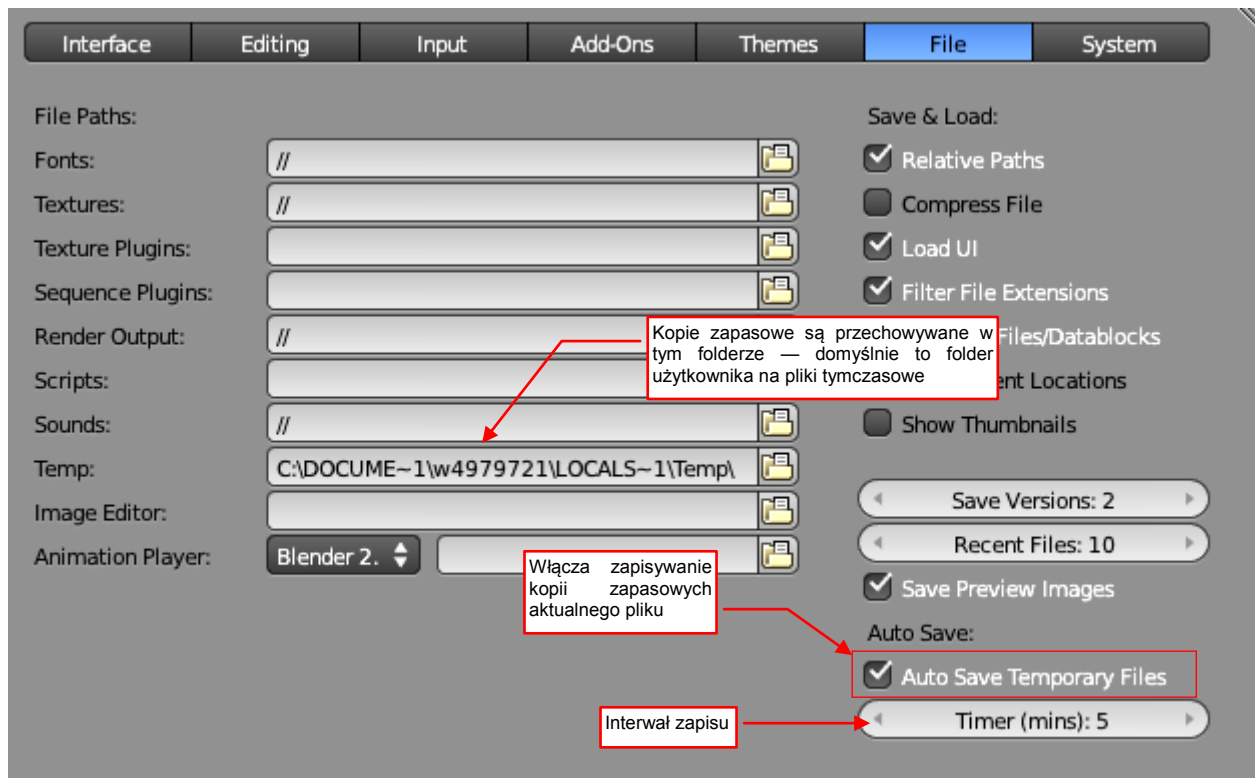


Rysunek 6.2.4 Przeniesienie konfiguracji Blendera, wymaganej dla tej książki

### 6.3 Kopia bezpieczeństwa i odtwarzanie

W tej sekcji opiszę sposób „awaryjnego” odzyskiwaniu pliku. Uczciwie mogę powiedzieć, że przez długi czas (pierwszych 7-8 miesięcy pracy) Blender zachowywał mi się bardzo stabilnie. Na palcach jednej ręki mógłbym policzyć sytuacje, gdy straciłem choć trochę pracy z powodu niespodziewanego zakończenia przez błąd w programie. Potem, gdy model przekroczył ok. 500 tys. ścian, a ja zacząłem używać różnych zaawansowanych funkcji, bywało, że czasami Blender mi się nagle „wysypał”. W każdym razie – istnieje pewien sposób, pozwalający odzyskać choć część pracy od ostatniego zapisu.

W oknie **User Preferences** przejdź do sekcji **File** (Rysunek 6.3.1):

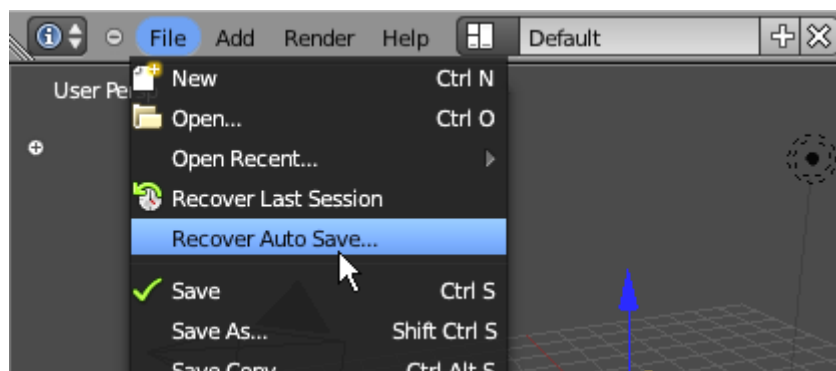


Rysunek 6.3.1 Włączenie automatycznego zapisu

Ważne są tu dwie kontrolki:

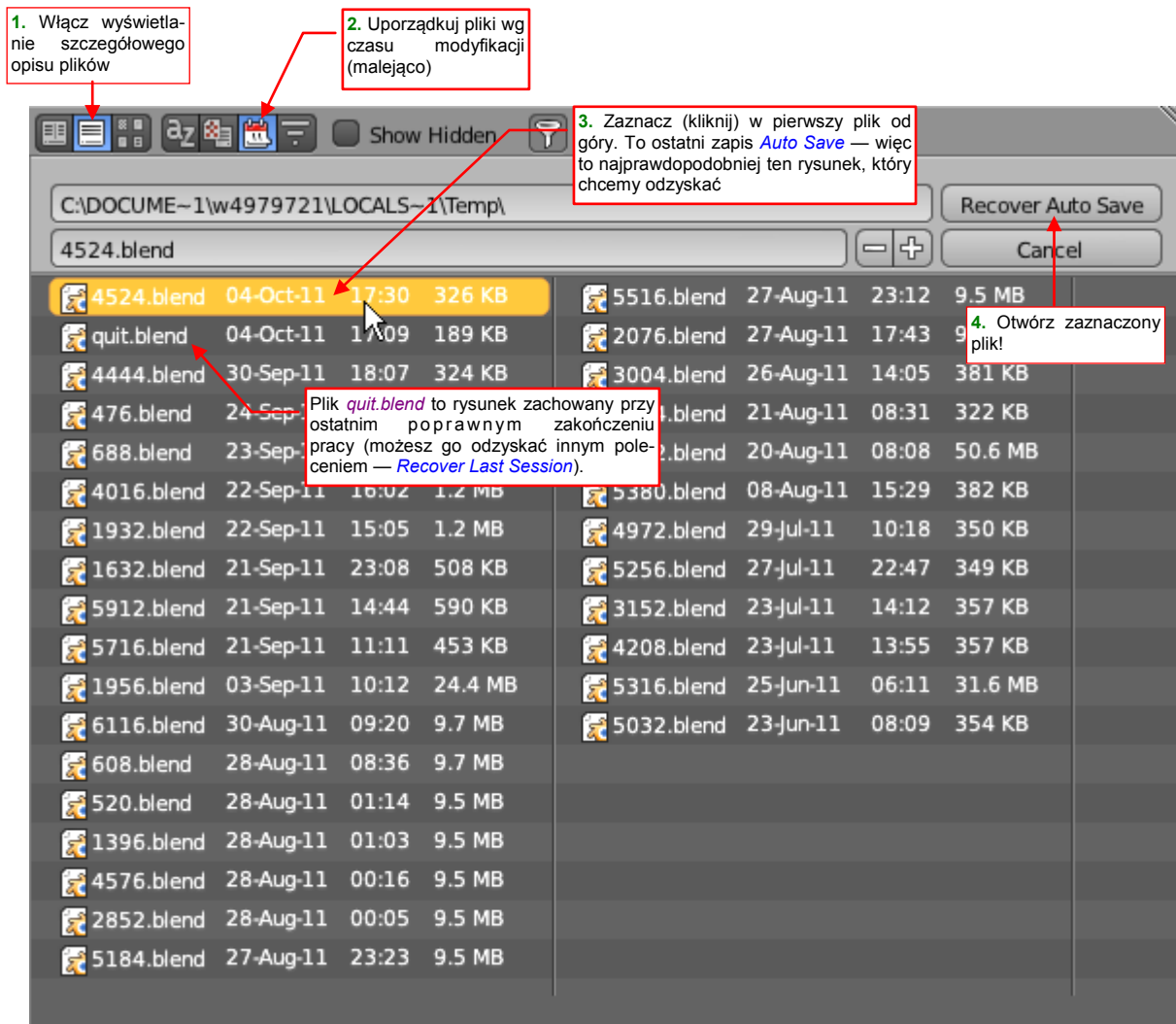
- Opcja **Auto Save Temporary Files**: moja rada: pozostaw ją zawsze włączoną, jak na ilustracji;
- Pole numeryczne **Timer**: tu wpisz, co ile minut ma być zachowywany stan pliku;

Gdy Blender nagle zgłosił „fatalny” błąd i został zamknięty przez system operacyjny, nie popadaj w rozpacz. Jeżeli miałeś włączony **Auto Save**, możesz jeszcze odzyskać swoją pracę. Uruchom ponownie Blender, a następnie wywołaj polecenie **File → Recover Auto Save** (Rysunek 6.3.2):



Rysunek 6.3.2 Przejdzie do odzyskiwania pliku

Blender otworzy wówczas w oknie wyboru plików Twój folder tymczasowy (Rysunek 6.3.3):



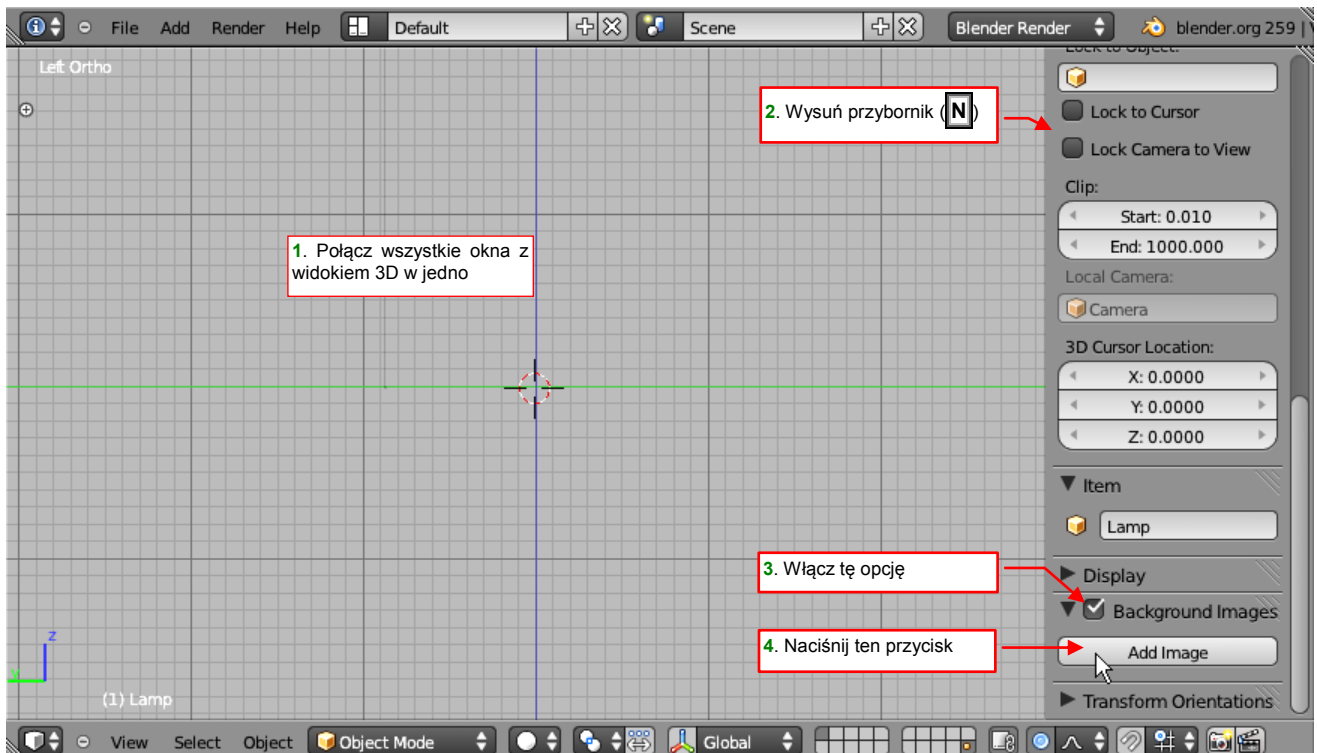
Rysunek 6.3.3 Wybór pliku do odzyskania

Dużo tutaj plików Blendera, prawda? I w dodatku nazwy wszystkich to jakieś niezrozumiałe, numeryczne nazwy! To właśnie efekty automatycznego zachowywania. Blender zapisuje plik pod nazwą, która jest tzw. „identyfikatorem procesu” (ang. *process ID* — skrót: *PID*). Co prawda — nie wiesz, jaki numer procesu miała sesja Blendera, która zakończyła się tym fatalnym błędem. Nie przejmuj się jednak tym: znajdziemy potrzebny plik, posługując się czasem jego modyfikacji. Włącz w nagłówku okna wyświetlanie szczegółów opisu plików (Rysunek 6.3.3). Następnie uporządkuj je malejąco według daty. Przypuszczalnie pierwszy od góry plik pochodzi z ostatnich kilkunastu minut — i to jest ten, którego szukamy! Zaznacz go (kliknięciem) i otwórz w Blenderze. Gdy się przekonasz, że to jest właściwy rysunek, zapisz go pod inną nazwą w tym samym folderze, w którym trzymasz poprzednie wersje swoich prac. Potem po prostu pracuj dalej. Robiłem tak niejedną raz. Pliki, odzyskane w ten sposób, są zupełnie normalne — bez żadnych luk w danych, czy skaz w zapisie.



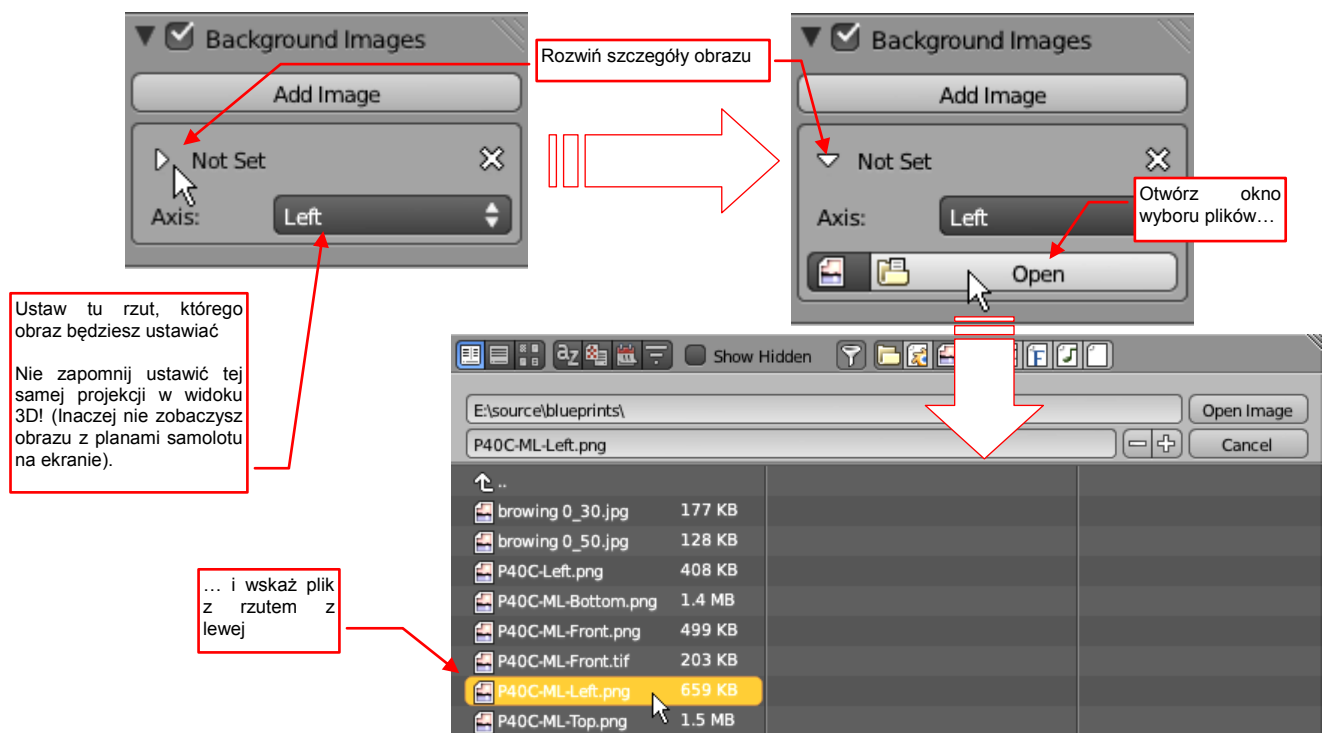
## 6.4 Ustalenie tła widoku (planów samolotu)

Aby podstawić plany samolotu jako tło, skal najpierw wszystkie okna **3D View** w jedno (Rysunek 6.4.1). Następnie otwórz przyborek **Properties** (**N**). Włącz w niej panel **Background Images**:



Rysunek 6.4.1 Przygotowanie do ustawiania tła okna **3D View**

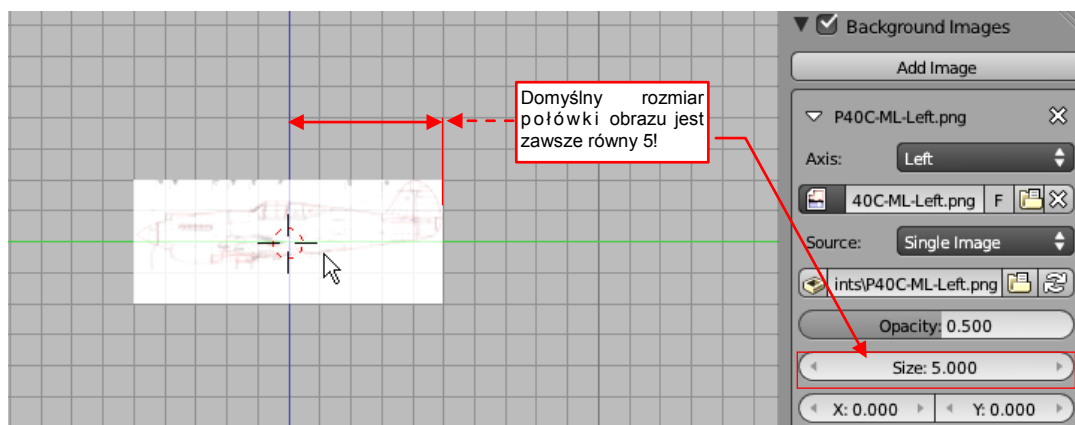
Następnie naciśnij przycisk **Add Image** (Rysunek 6.4.1). W panelu pojawi się nowa sekcja, początkowo bez obrazu (**Not Set** — Rysunek 6.4.2):



Rysunek 6.4.2 Wczytanie obrazu z rzutem samolotu

Przypisz tej sekcji obraz (przyciskiem **Open**). Ustaw też oś widoku (**Axis**), odpowiednio do wczytanego obrazu.

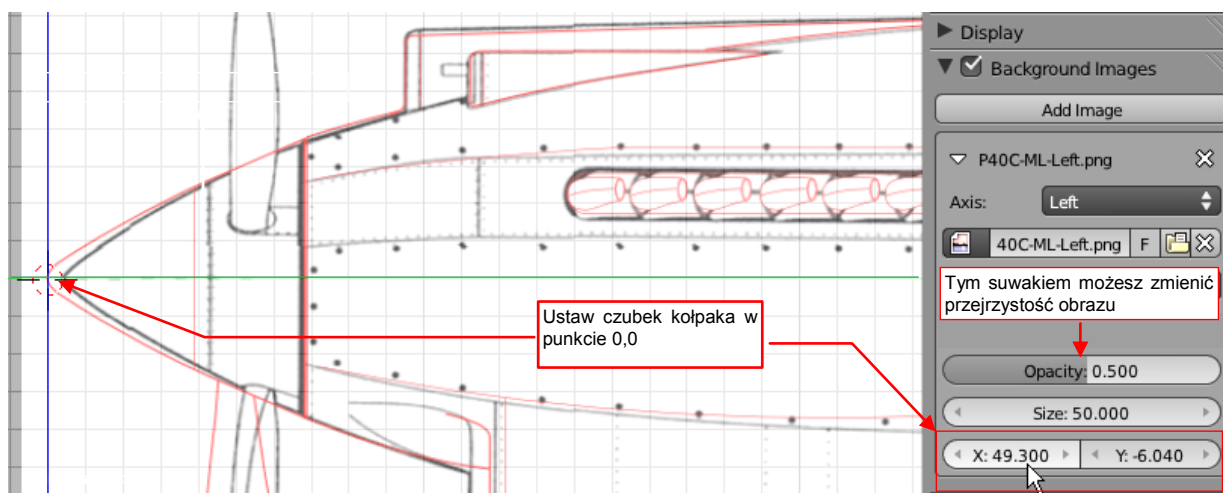
W środku układu współrzędnych widoku 3D powinieneś zobaczyć załadowany obraz (Rysunek 6.4.3). (Jeżeli miałeś włączone duże powiększenie, obraz może być mały):



Rysunek 6.4.3 Wczytany obraz

- Jeżeli na swoim ekranie nie widzisz obrazu w tle okna **3D View**, sprawdź:
  - czy masz włączony widok z tej samej strony, jaki wybrałeś z listy **Axis** (por. Rysunek 6.4.2)? Jeżeli nie — ustaw go;
  - czy widzisz na ekranie środek układu współrzędnych (przecięcie dwóch kolorowych osi)? Jeżeli nie — naciśnij **Home**, aby go zobaczyć;
  - czy masz ustawione odpowiednie powiększenie? (Obraz może być obecnie zbyt mały, abyś go zobaczył). Spróbuj zmienić powiększenie;

Blender domyślnie ustawia rozmiar wczytanego rysunku (**Size**) na **5.0** jednostek, a jego środek umieszcza w centrum układu współrzędnych (Rysunek 6.4.3). Wartość, wpisywana w pole **Size**, określa rozmiar połowy szerokości obrazu. Zakładając, że 1 jednostka Blendera to 10 cm rzeczywistego samolotu, powinniśmy zwiększyć **Size** do **50.0**. (Długość kadłuba P-40C wynosiła 967cm, a nasz cały obraz będzie się w tej skali rozciągał na niewiele więcej — 100 jednostek, czyli 1000cm). Jednocześnie przesuń obraz (wartości **X**, **Y**) tak, by oś kadłuba pokryła się z osią **Y**, a czubek kołpaka śmigła dotykał osi **Z** (Rysunek 6.4.4).

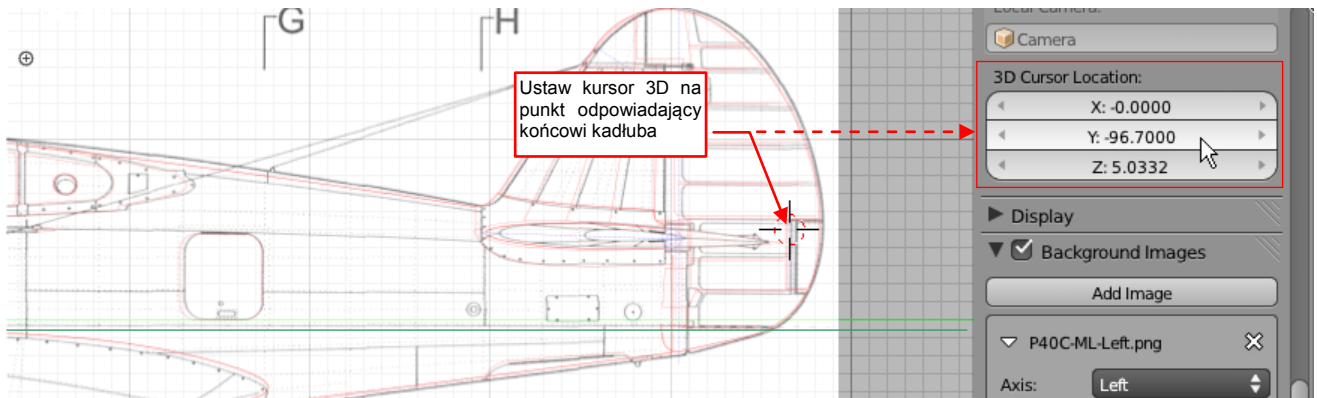


Rysunek 6.4.4 Obraz tła — wyrównanie do dokładnego ustalenia rozmiaru

- Najłatwiej jest posługiwać się tu polami **X** i **Y** jak suwakami — przesuwając myszkę (z **LPM**) ponad taką kontrolką. Możesz wtedy obserwować, jak dynamicznie przesuwa się obraz tła.

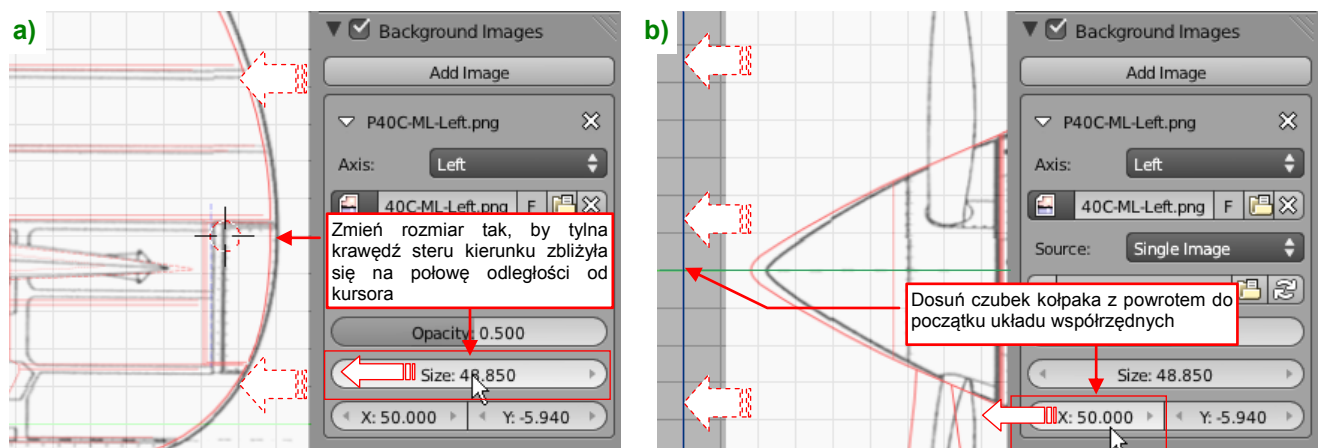
Zwróć uwagę, że kontrolki **X** i **Y** z panelu **Background Images** to lokalne współrzędne obrazu, a przestrzeni widoku. Warto także uczynić obraz tła częściowo przejrzystym, by było widać umieszczoną pod spodem siatkę. Służy do tego suwak **Opacity** (Rysunek 6.4.4).

Aby wyznaczyć punkt, do którego ma sięgać koniec kadłuba samolotu, w panelu **View** ustaw **3D Cursor Location** na współrzędne: **X = 0.0**, **Y = 96.7** (odpowiada 967 cm rzeczywistej długości kadłuba), **Z = 0.0**. Spowoduje to pojawienie się kursora 3D w odpowiednim miejscu na osi **Y** (Rysunek 6.4.5). (Potem możesz zmienić współrzędną **Z** na nieco większą, by kursor znalazł się dokładnie tam, gdzie leży koniec kadłuba — trochę ponad osią **Y**).



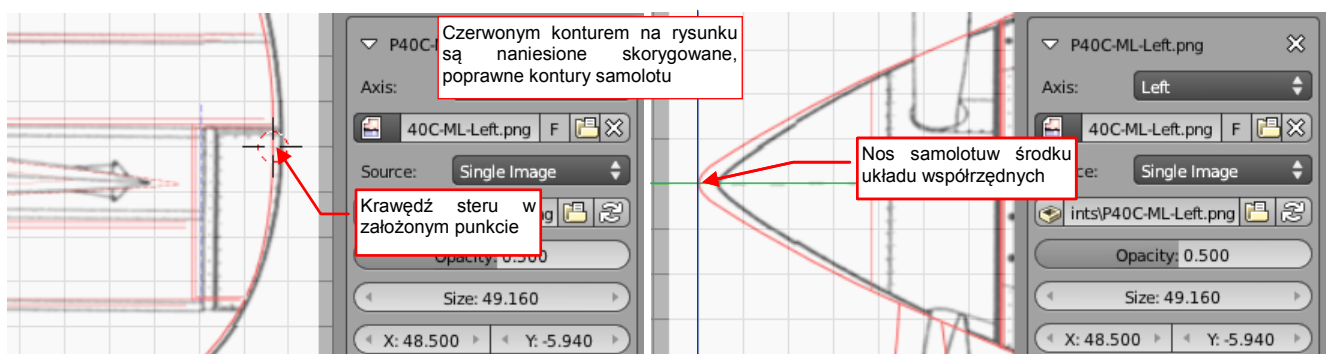
Rysunek 6.4.5 Przymiarki do ustalenia końca kadłuba

Teraz ustaw w panelu **Background Images** trochę mniejszą wartość **Size** — aby krawędź steru kierunku przybliżyć na mniej więcej połowę początkowego dystansu, wyznaczonego przez kursor 3D (Rysunek 6.4.6a):



Rysunek 6.4.6 Dopasowywanie rysunku tła do zadanych wymiarów

Pamiętaj, że zmiana rozmiaru zachodzi względem środka obrazu — w obydwu kierunkach! Po każdej zmianie skali przesunij się w pobliże środka układu współrzędnych. Zobaczysz tam, że nos samolotu z obrazu uległ odśunięciu w prawo (Rysunek 6.4.6b). Musisz to skorygować przesunięciem obrazu w lewo (zmianą współrzędnej **X**). Następnie znów zjrzyj na koniec kadłuba, i powtórnie zmień skalę tak, by znów zmniejszyć dystans pomiędzy krawędzią steru i kursorem 3D o połowę. Potem znów to skoryguj przesunięciem (kontrolką **X**) nosa w lewo. Po kilku takich przybliżeniach powinieneś osiągnąć założone położenie rzutu z lewej (Rysunek 6.4.7):



Rysunek 6.4.7 Rysunek dopasowany

Po wyskalowaniu obrazu pozostaje przesunąć go w docelowe położenie. W praktyce końcówka nosa samolotu nie jest najlepszą powierzchnią odniesienia. Proponuję ustawiać obrazy tak, by przez środek układu współrzędnych przechodziła jakaś inna, "ważniejsza" dla konstrukcji powierzchnia. Użyjmy do tego celu podstawowej wręgi kadłuba, do której były zamocowane: silnik i przedni dźwigar płata<sup>1</sup>. Przesuń obraz (zmieniając wartość **X** obrazu) tak, by środek układu współrzędnych znalazł się w tym miejscu (Rysunek 6.4.8):

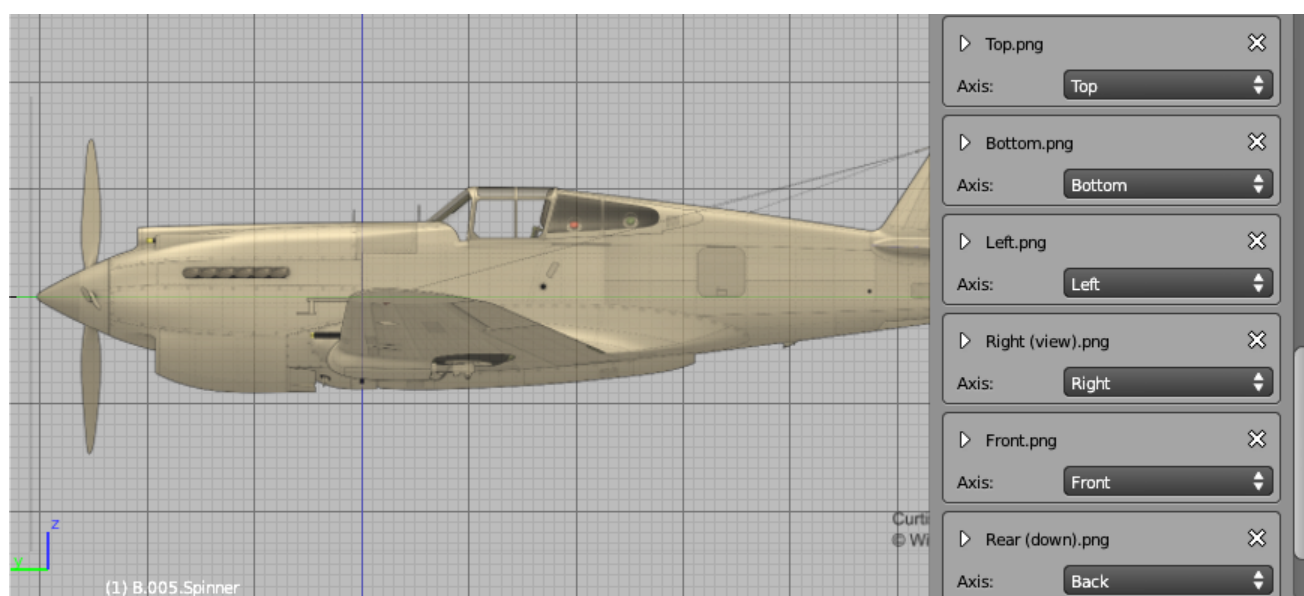


Rysunek 6.4.8 Ostateczne położenie obrazu tła — ściana ogniowa będzie powierzchnią bazową

W podobny sposób dodaj do okna kolejne obrazy, każdemu z nich przypisując odpowiednią oś (**Axis**) widoku.

- Uwaga: wartość **Size** dotyczy dłuższego boku każdego obrazu, więc będzie inna dla rzutów z boku, i inna dla rzutów z góry/dołu!

W tym przykładzie posługiwałem się rysunkami P-40, których przygotowanie opisałem w pierwszej części tej książki (Wyciąg I). Jeżeli jednak chcesz już skorzystać ze zweryfikowanych rysunków, użyj planów z pliku [source.zip](#) (por. str. 16). W jego podkatalogu [source\plans](#) znajdziesz dwa pełne zestawy rzutów P-40B. Rysunek 6.4.9 przedstawia okno z przypisanymi wszystkimi widokami z zestawu [p40b-lowres](#):



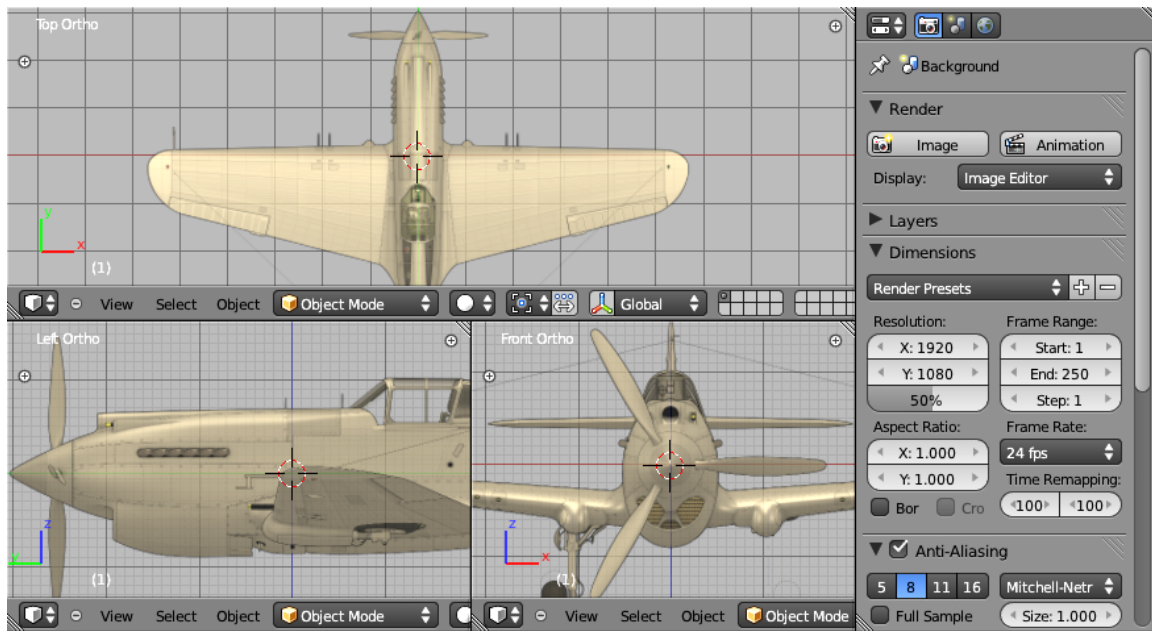
Rysunek 6.4.9 Okno z zestawem obrazów przypisanym do każdego widoku (zestawu planów do tej książki)

Nadałem tym planom odcień pomarańczowy, by odróżniały się od domyślnej szarości modelowanych siatek.

<sup>1</sup> Jeżeli nie planowałem wykonania innych wersji P-40, wybrałbym jako bazę płaszczyznę podstawy kołpaka śmigła. Płaszczyzna ta uległa jednak przesunięciu od P-40D, i taki wybór utrudniłby użycie elementów z P-40B do stworzenia modelu tej nowej wersji.



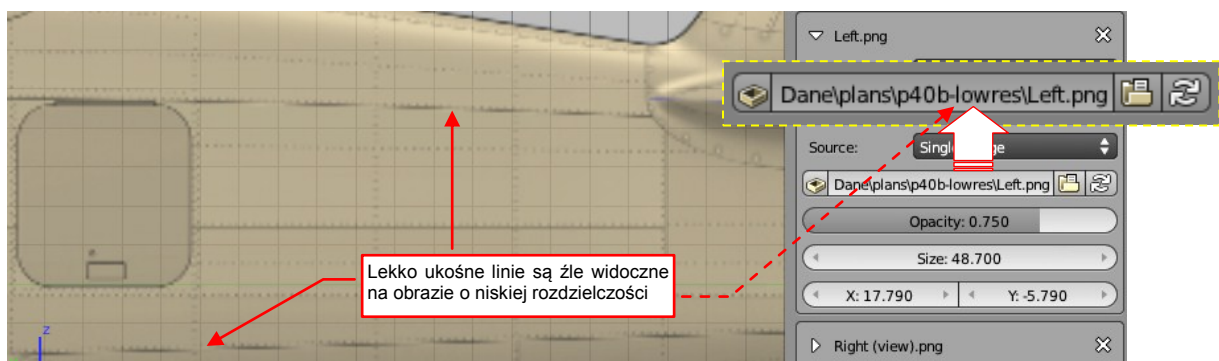
Gdy wszystkie widoki samolotu są już przypisane do okna, podziel je na kilka rzutów, których będziemy używać podczas modelowania (Rysunek 6.4.10):



Rysunek 6.4.10 Układ ekranu po ostatecznym podziale

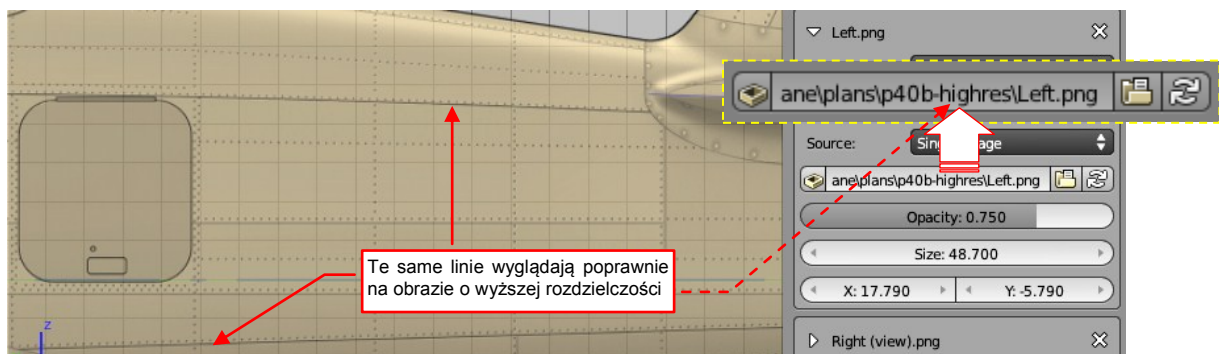
Zwróć uwagę, że Blender będzie podstawiał pod ekran obraz tła tylko wtedy, gdy wybierzesz jeden z widoków ortogonalnych: z lewej, z prawej, z góry, z dołu... (Decyduje o tym przypisana obrazowi oś *Axis*).

Na słabszych komputerach lepiej jest stosować wersje obrazów o mniejszej rozdzielczości, bo wtedy wszystkie polecenia *Pan* i *Zoom* działają szybciej. Zestaw takich obrazów znajdziesz w folderze *plans\p40b-lowres\*. Mała rozdzielczość wiąże się jednak z pewnym mankamentem: zanikaniem lekko ukośnych linii (Rysunek 6.4.11):



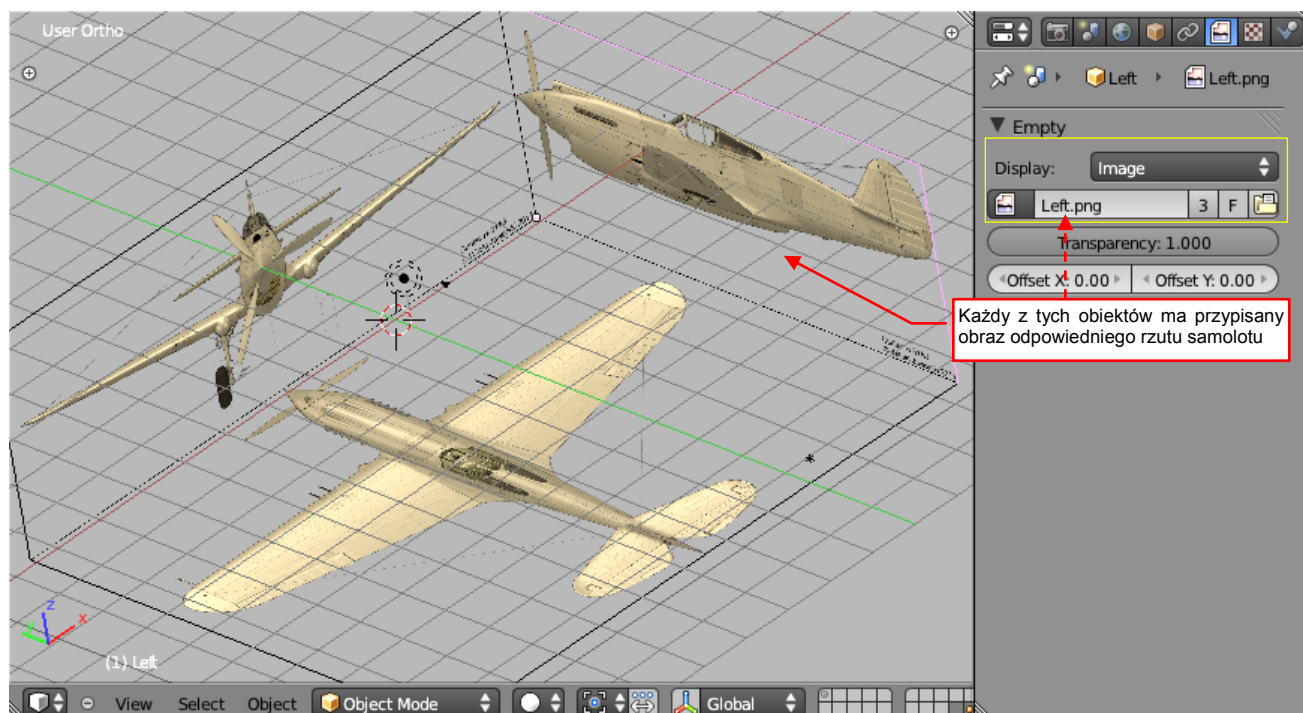
Rysunek 6.4.11 Problem z lekko ukośnymi liniami na obrazach o małej rozdzielczości

Możesz to jednak na chwilę skorygować — wystarczy że w panelu obrazu wskażesz plik z wersją o wyższej rozdzielczości (Rysunek 6.4.12). Potem w ten sam sposób z powrotem przestaw go na wersję *lowres\*.



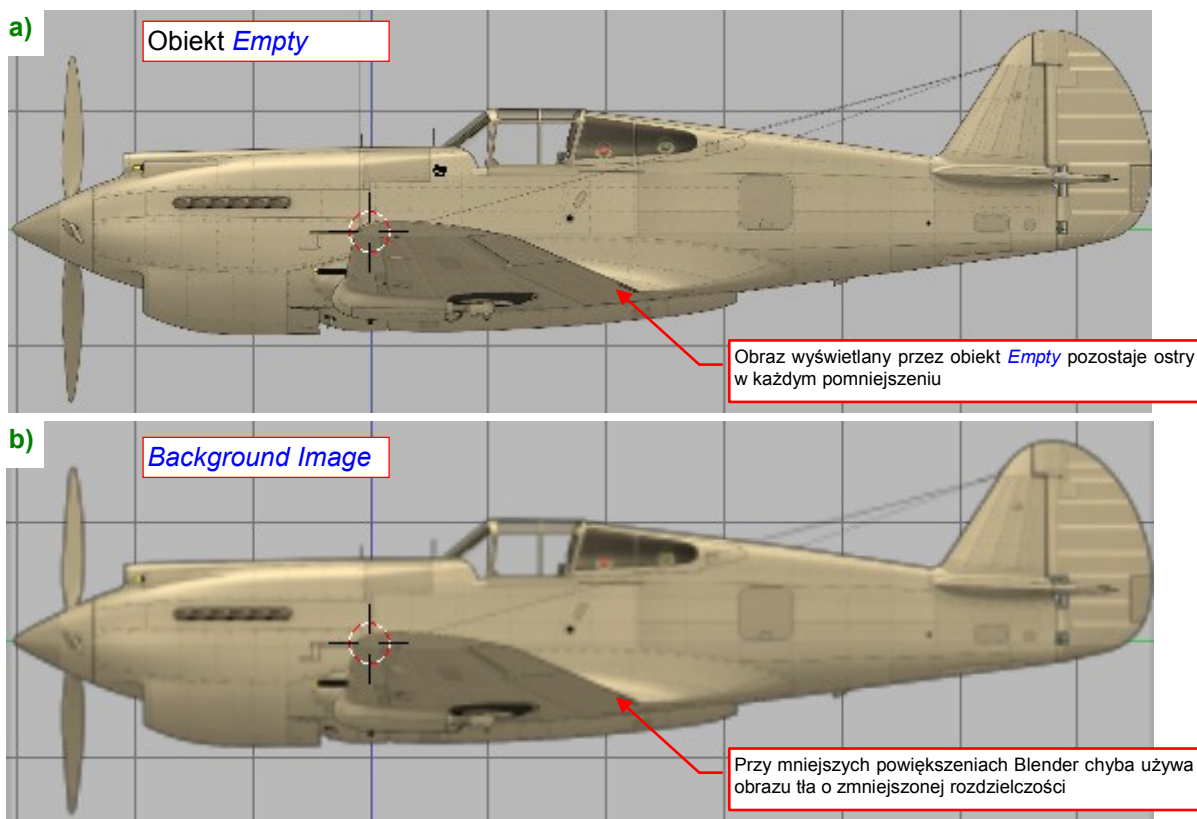
Rysunek 6.4.12 Przełączenie na obraz o większej rozdzielczości

Na zakończenie chciałbym wspomnieć o innej metodzie „podstawienia” rysunków samolotu. Zamiast obrazów tła można w tym celu wykorzystać obiekty typu *Empty* (por. str. 218), którym przypiszesz odpowiednie obrazy (Rysunek 6.4.13):



Rysunek 6.4.13 Alternatywne rozwiązanie — podstawienie obrazów samolotu za pomocą trzech obiektów typu *Empty*

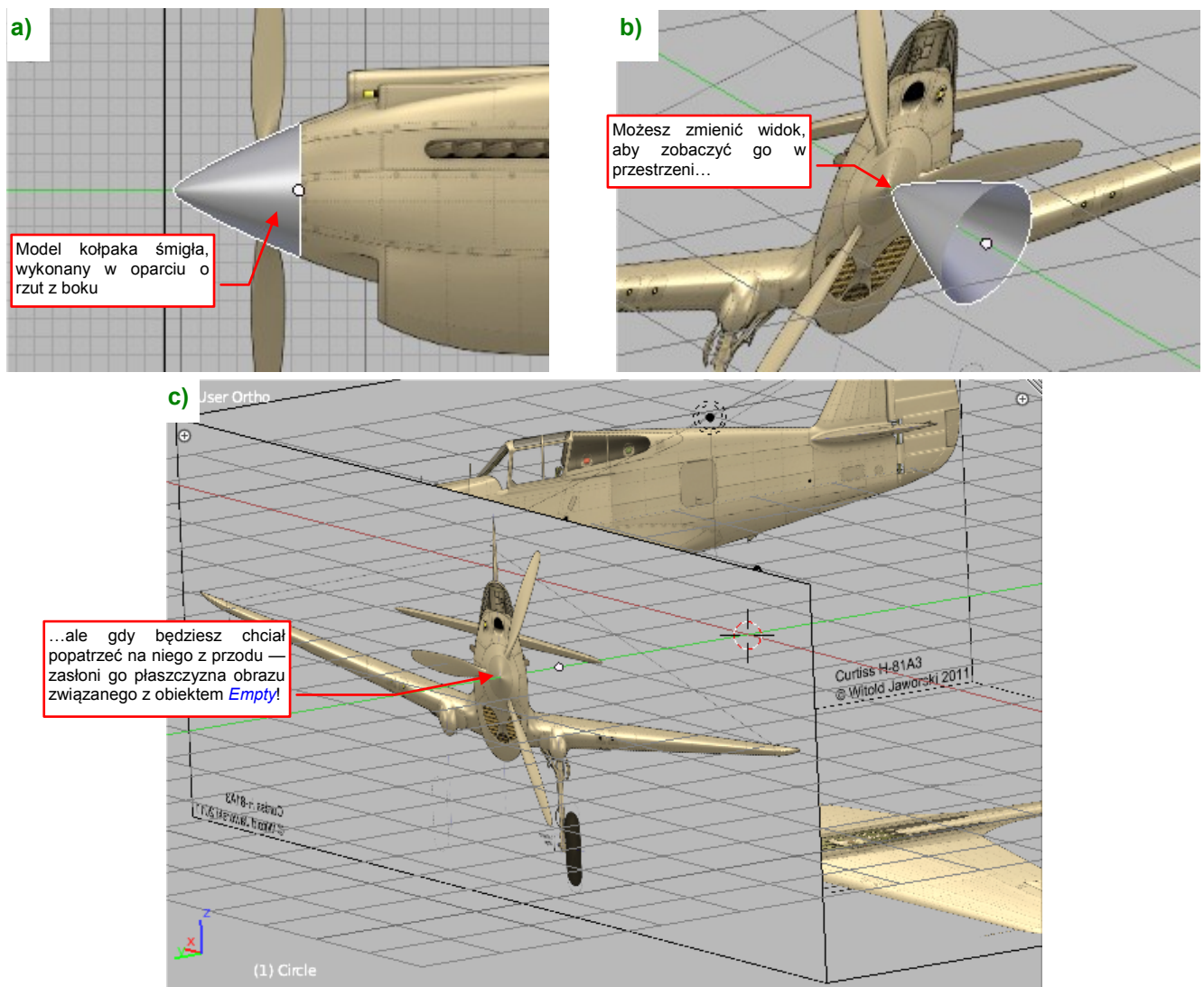
Tak wyświetlane obrazy lepiej wyglądają w mniejszych powiększeniach (Rysunek 6.4.14):



Rysunek 6.4.14 Obrazy wyświetlane w obiektach *Empty* lepiej wyglądają w mniejszych powiększeniach

Jednak gdy powiększysz obraz tła (Rysunek 6.4.14b) do projekcji w której zazwyczaj formujesz elementy modelu, stanie się tak samo ostry i wyraźny jak obraz wyświetlany przez obiekt *Empty*.

Obrazy wyświetlane przez obiekty **Empty** mają jednak inną wadę: nie są przejrzyste! Popatrz na model kołpaka śmigła, przedstawiony przez Rysunek 6.4.15a):



**Rysunek 6.4.15** Obrazy **Empty** zasłaniają modelowany obiekt

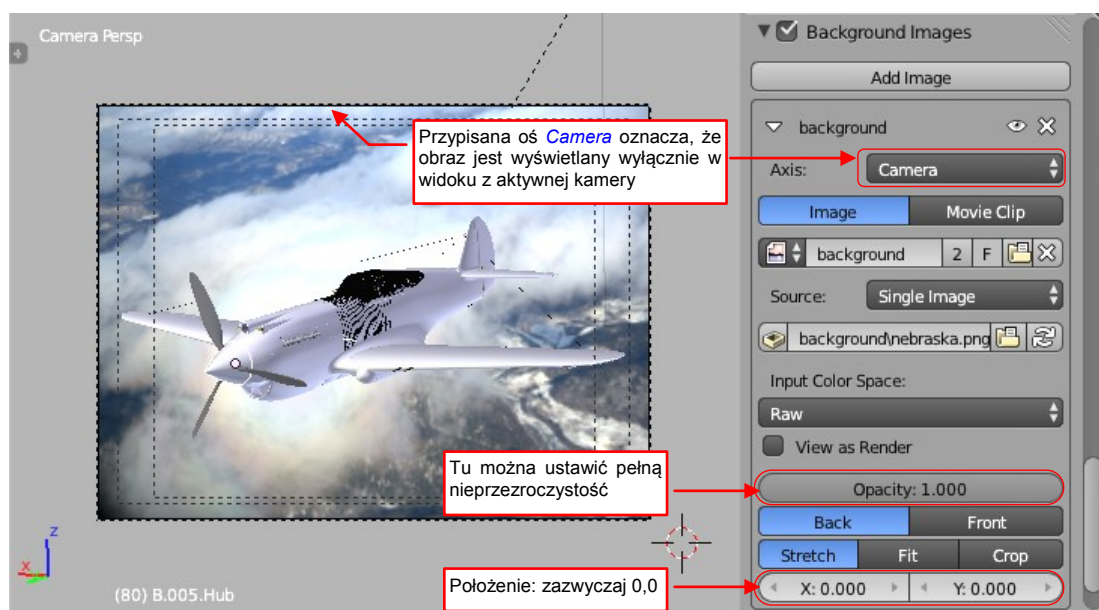
Stworzyłem go w oparciu o rzut z boku (Rysunek 6.4.15a). Potem obróciłem trochę widok, aby sprawdzić, jak wygląda w przestrzeni (Rysunek 6.4.15b). Jednak gdy chciałem go zobaczyć z przodu — płaszczyzna z obrazem skutecznie go zasłoniła (Rysunek 6.4.15c). Można to jednak częściowo skompensować, zmieniając wartość parametru **Transparency** we właściwościach obiektu **Empty** (por. Rysunek 6.4.13).

Innym problemem może także być wydajność — szczególnie gdy zdecydujesz się przez cały czas używać obrazów o wyższej rozdzielczości. O ile obrazy wyświetlane jako **Background Image** pojawiają się tylko wtedy, gdy włączysz „ich” rzut, to obrazy związane z obiektami **Empty** są rysowane przez cały czas — także w widoku przestrzennym, kiedy są raczej bezużyteczne.

Oczywiście, można ten problem rozwiązać umieszczając obiekty **Empty** na oddzielnej warstwie, którą będziesz w razie potrzeby wyłączał. Po co jednak to robić, skoro **Background Image** oferuje tę funkcjonalność „w standardzie”?

Uważam, że w praktyce modelowania lepiej się sprawdza podstawianie rysunków referencyjnych za pomocą funkcji **Background Image**. Opisałem je jednak tutaj, gdyż możesz mieć inny styl pracy, lub po prostu preferować taką formę prezentacji, jaką oferują obrazy wyświetlane za pomocą obiektów **Empty**.

**Background Image** pomaga także w kompozycji sceny: zdjęcie tła można zamapować w trybie **Camera** (Rysunek 6.4.16):



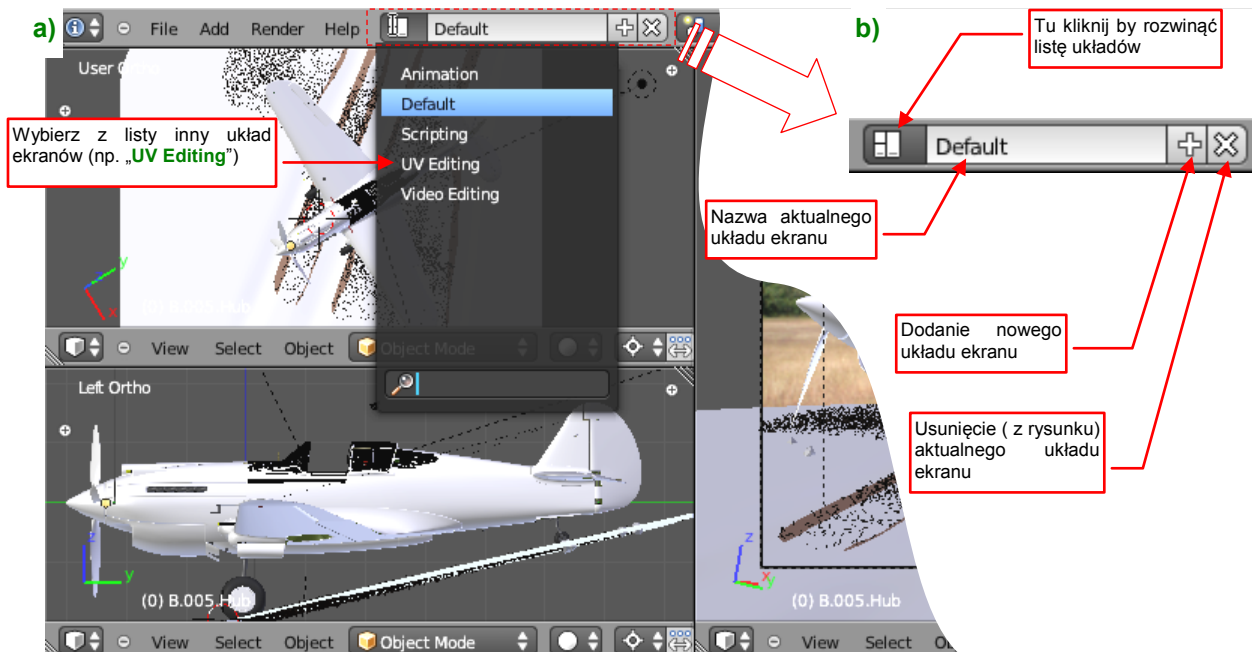
Rysunek 6.4.16 Nałożenie zdjęcia tła w trybie **Background Image:Camera**

W ten sposób łatwiej jest właściwie ustawić model w kadrze.



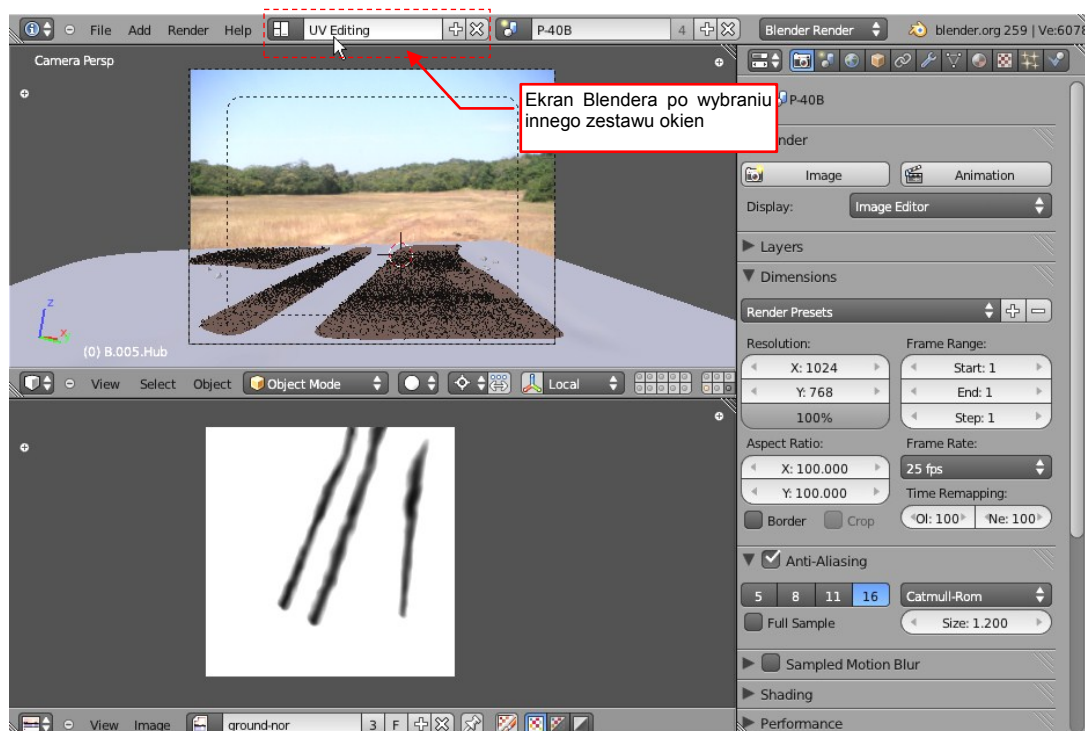
## 6.5 Układ ekranu (*Screen layout*) — zarządzanie

Blender domyślnie proponuje kilka różnych układów ekranu, przygotowanych do różnych faz pracy nad projektem. Możesz się pomiędzy nimi przełączać za pomocą listy *Screen lay-out* (Rysunek 6.5.1a):




Rysunek 6.5.1 Lista *Screen lay-out*: Zespół kontrolki do zarządzania układem ekranu

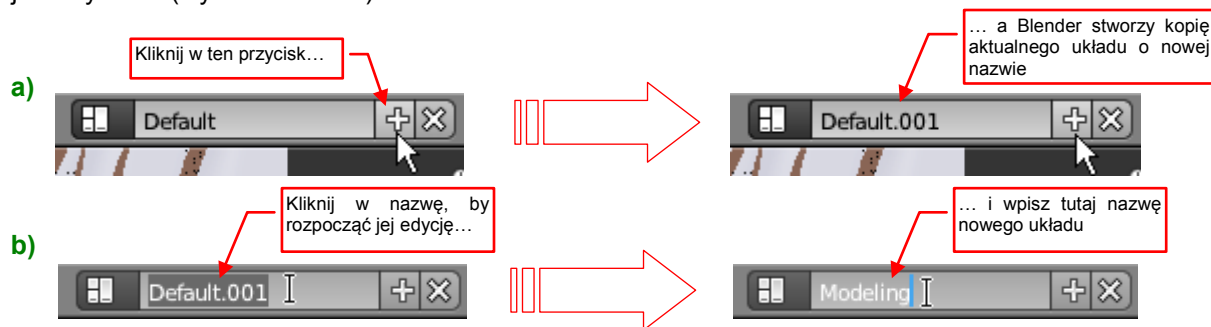
Lista układów ekranu (Rysunek 6.5.1b) to jednocześnie dobry przykład typowej kontrolki Blendera do zarządzania referencjami tzw. bloków danych (np. materiałami, teksturami, itp.)<sup>1</sup>. Po wybraniu z listy *Screen lay-out* np. układu do pracy nad teksturami (*UV Editing*), zobaczysz zupełnie inne okna (Rysunek 6.5.2):



Rysunek 6.5.2 Inny układ ekranu — (*UV Editing*)

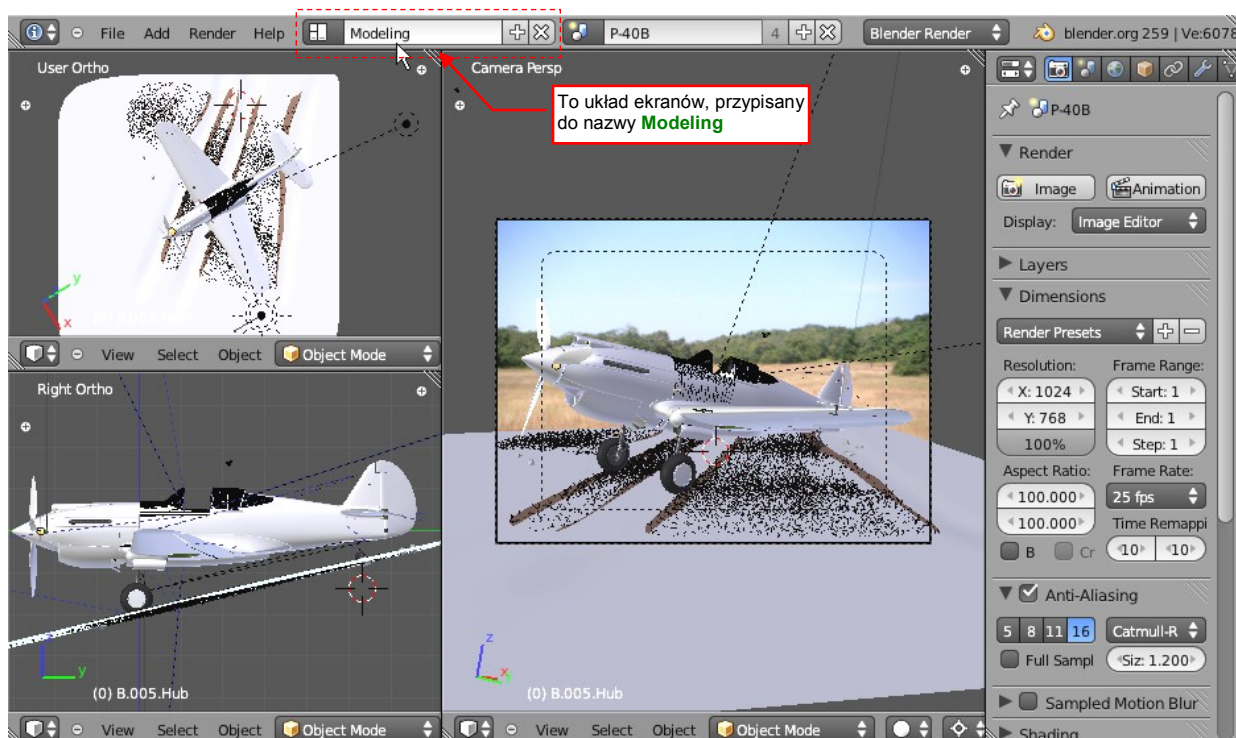
<sup>1</sup> Wyrażenie „blok danych” (*datablock*) przewija się przez całe słownictwo Blendera. W istocie to jedno ze kluczowych pojęć w tym programie. Dokładne wyjaśnienie tego określenia znajdziesz na str. 1187.

Jeżeli nie jesteś zadowolony z domyślnych ustawień Blendera — możesz tu wszystko pozmieniać. Oprócz zmiany istniejącego układu, istnieje możliwość dodania nowego. Wystarczy nacisnąć przycisk , umieszczony z lewej strony okna (Rysunek 6.5.3a):




Rysunek 6.5.3 Dodanie własnego układu ekranu

Potem możesz już go pozmieniać, jak chcesz (Rysunek 6.5.4):

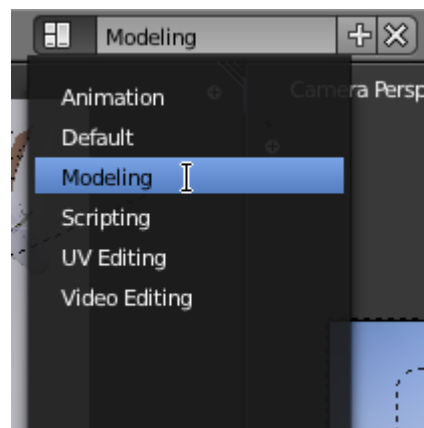


Rysunek 6.5.4 Nowy układ ekranu: **Modeling**

Nowy układ już figuruje na liście (Rysunek 6.5.5). Oczywiście, jeżeli nie będziesz zadowolony ze swojego układu — zawsze możesz go usunąć, naciskając przycisk , umieszczony po prawej stronie listy [Screen layout](#). Gdy to zrobisz, znajdziesz się w poprzednim układzie (w tym przykładzie byłby to „Default”).

- UWAGA: Blender przed usunięciem układu ekranów nie zadaje żadnego pytania typu „jesteś pewien?”. Usuwa i już!

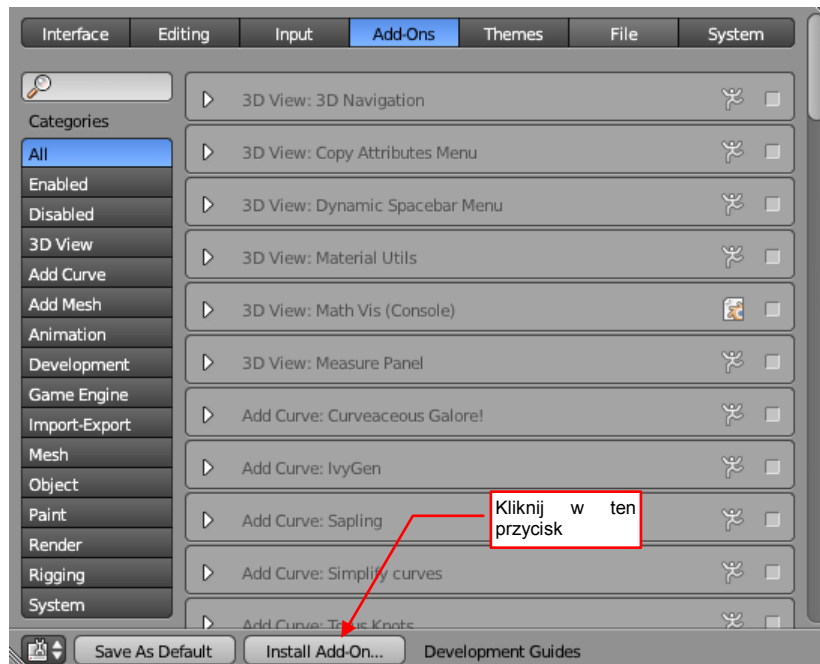
Układy ekranów są zapamiętywane oddzielnie w każdym pliku Blendera. Dlatego np. w pliku pochodzącym z Blendera 2.4 będą miały inne nazwy, niż pokazane na ilustracjach w tym przykładzie. Aby stworzyć układ, którego będziesz używać we wszystkich swoich rysunkach, musisz zapisać go do pliku startowego, poleceniem [File → Save User Settings](#) (por. str. 180).



Rysunek 6.5.5 Nowy układ na liście

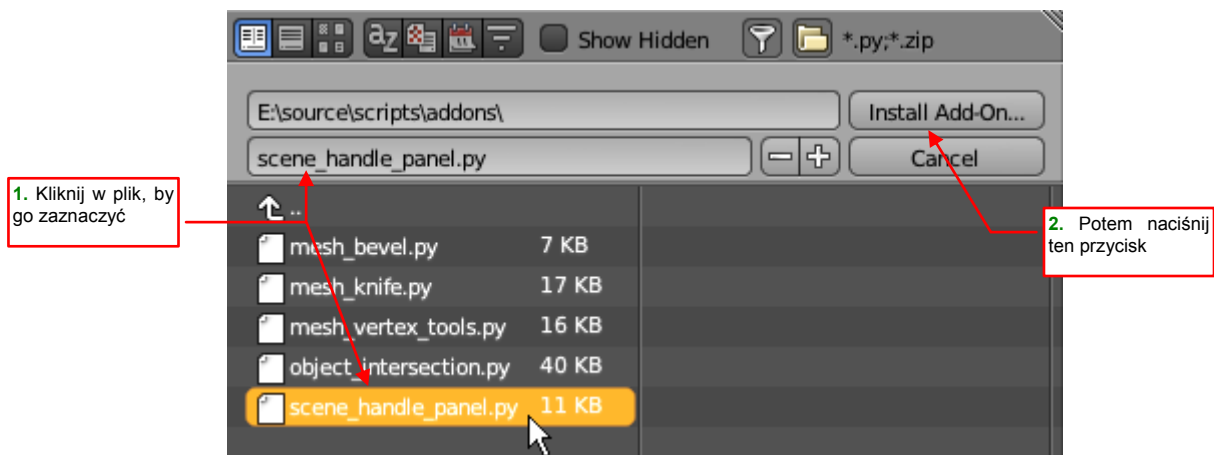
## 6.6 Instalacja i usuwanie dodatków (add-ons)

Blenderowi towarzyszy dużo dodatków<sup>1</sup>, rozszerzających jego funkcjonalność. Są to skrypty napisane w języku programowania Python, zazwyczaj przez samych użytkowników. Dodatki to zazwyczaj niewielkie pliki z rozszerzeniem `*.py`. Oprócz tych, które są dostarczane wraz z Blenderem, możesz znaleźć inne np. na forach Blendera w Internecie. Także do tej książki dodałem kilka narzędzi, które wykorzystuję od lat i uważam za bardzo przydatne i godne polecenia (znajdziesz je w pliku `source.zip`, w `scripts\addons` — por. str. 16). Instalacja dodatku Blendera może polegać po prostu na wgraniu pliku wtyczki do odp. podkatalogu w folderze z konfiguracją programu (por. str. 187). Jeżeli jednak nie czujesz się w tej materii zbyt pewnie, pokażę tutaj inną, mniej „hakerską” metodę. W oknie **User Preferences**, sekcji **Add-Ons**, naciśnij przycisk **Install Add-On...** (Rysunek 6.6.1):



Rysunek 6.6.1 Wywołanie instalacji dodatku (okno **User Preferences**)

Blender wyświetli wówczas okno **File Browser**. Przejdź w nim do katalogu, w którym umieściłeś dodatki do zainstalowania, wskaż jeden z nich, i naciśnij przycisk **Install Add-On** (Rysunek 6.6.2):

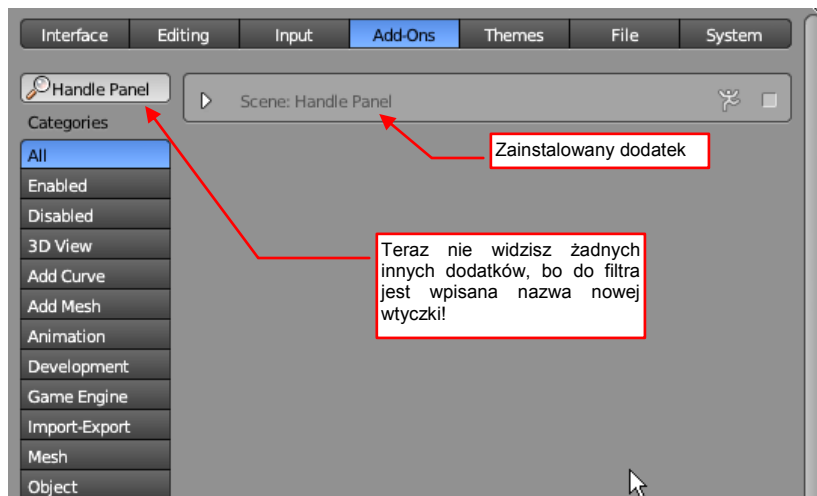


Rysunek 6.6.2 Wybór pliku skryptu do zainstalowania

<sup>1</sup> Dodatki czasami są także nazywane „wtyczkami” (ang. *plugin*), ale ta nazwa powinna raczej dotyczyć binarnych, a nie skryptowych rozszerzeń Blendera. W poprzedniej wersji Blendera — 2.4 — istniała możliwość użycia binarnych wtyczek implementujących jakieś specjalne tekstury. W wersji 2.5 i następnych niczego takiego (jeszcze) nie zaimplementowano.

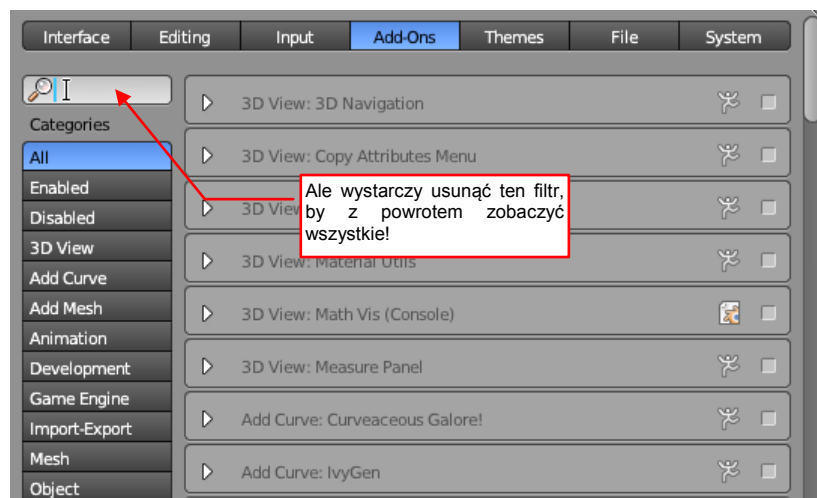
- W oknie **File Browser** wskazuj zawsze tylko jeden skrypt! Choć Blender w tym oknie pozwala zaznaczyć wiele plików dodatków, to zainstaluje tylko jeden – ten, który zaznaczyłeś jako ostatni.

Dodatek jest już zainstalowany. Abyś ją dostrzegł wśród innych, Blender przełączył wyświetlane kategorie na **All** i w polu filtra wpisał jego nazwę (Rysunek 6.6.3):



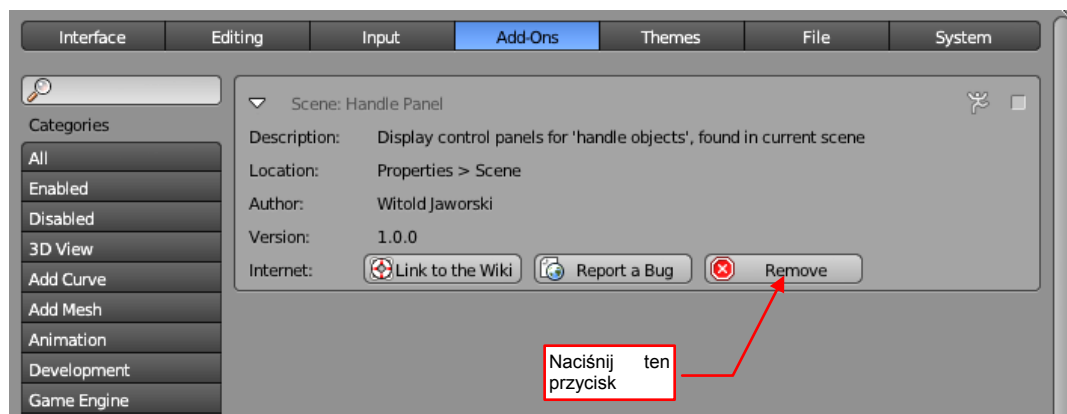
Rysunek 6.6.3 Wygląd okna **User Preferences** po zainstalowaniu dodatku (**Handle Panel**)

W razie czego nie bądź przerażony, że z Blendera znikły wszystkie pozostałe dodatki. Wystarczy usunąć filtr (Rysunek 6.6.4):



Rysunek 6.6.4 Wygląd okna **User Preferences** po usunięciu filtra

Aby usunąć dodatek z Blendera, po prostu naciśnij jej przycisk **Remove** (Rysunek 6.6.5):



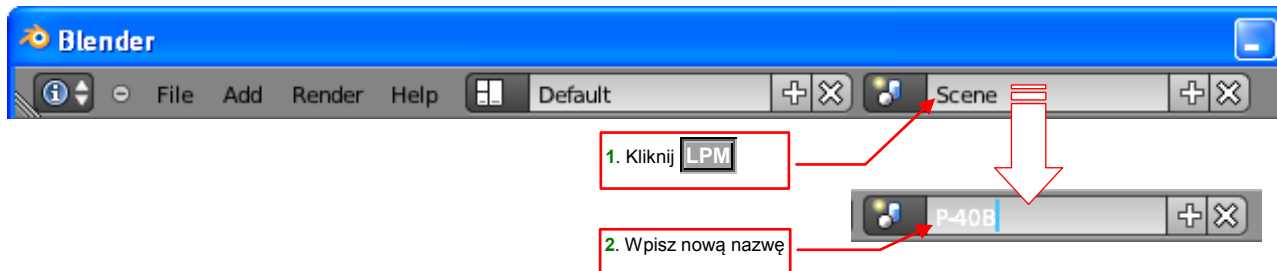
Rysunek 6.6.5 Usuwanie dodatku



## 6.7 Sceny — zarządzanie

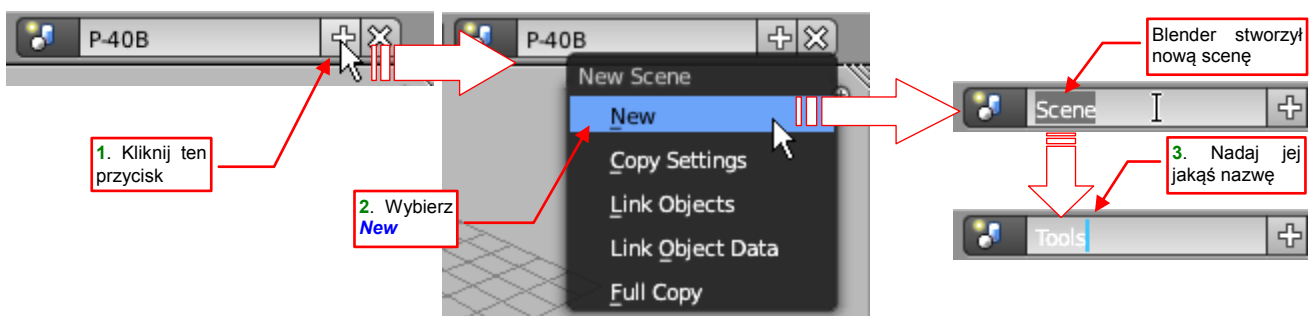
Plik Blendera może zawierać wiele scen. Każda z nich to odrębna "przestrzeń", w której można umieszczać obiekty (więcej na ten temat — str. 466). Domyślnie w nowym rysunku istnieje tylko jedna, o nazwie **Scene**. Do zarządzania scenami służy zespół kontrolki umieszczony w nagłówku okna **Info** (Rysunek 6.7.1). Składa się z pola, pokazującego nazwę aktualnej sceny, przycisku rozwijającego listę scen, oraz przycisku usuwającego aktualną scenę.

Aby zmienić nazwę aktualnej sceny, kliknij **LPM** w jej nazwę, i wpisz nową (Rysunek 6.7.1):



Rysunek 6.7.1 Zmiana nazwy aktualnej sceny

Aby stworzyć nową scenę, naciśnij przycisk **+**, umieszczony z prawej strony nazwy (Rysunek 6.7.2):



Rysunek 6.7.2 Dodanie nowej sceny

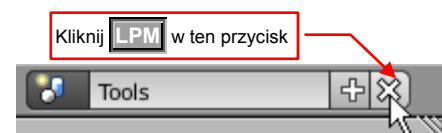
Spowoduje to pojawienie się menu **New scene**, z którego możesz wybrać jedną z opcji:

- **New**: tworzy nową, pustą scenę (z globalnymi ustawieniami jak w pliku startowym);
- **Copy Settings**: tworzy nową, pustą scenę, kopiując globalne ustawienia ze sceny aktualnej;
- **Link Objects**: wypełnia nową scenę referencjami do obiektów z aktualnej sceny. (Ten sam obiekt będzie "użyty" przez obydwie sceny. Więcej na ten temat — str. 466);
- **Link Object Data**: tworzy w nowej scenie kopie obiektów z aktualnej sceny, które współdzielą z obiektami aktualnej siatki, materiały, itp. (To tak, jak gdybyś wszystkie powielił poleceniem **Alt-D** — **Duplicate Linked**);
- **Full Copy**: cała zawartość nowej sceny będzie niezależną kopią aktualnej;

W praktyce stosuję zazwyczaj opcję **Copy Settings** (pojedyncze obiekty można później dodać do innej sceny poleceniem **Object → Make Links → To Scene**). Nowa scena otrzyma taką nazwę, jak scena oryginalna, z dodanym na końcu ".001". Zmień ją na właściwą.

Aby usunąć aktualną scenę, naciśnij przycisk **[x]** z prawej strony nazwy (Rysunek 6.7.3):

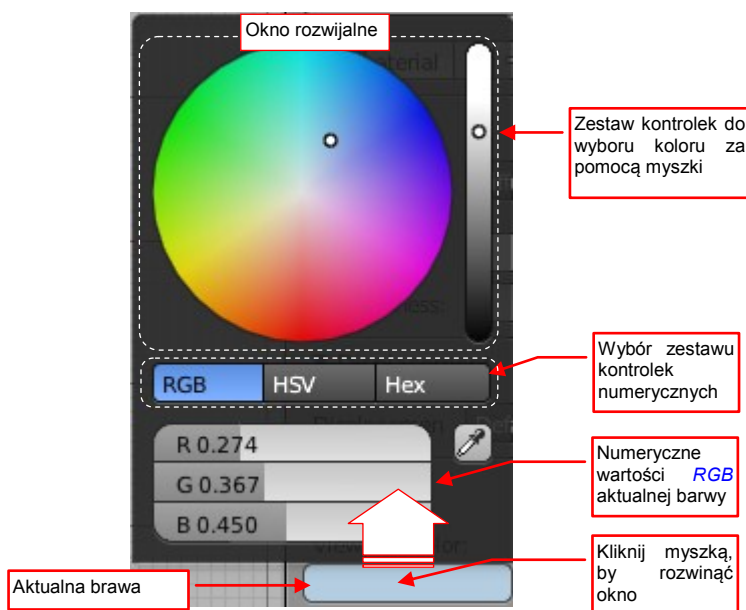
- Blender nie zadaje już żadnego pytania „czy jesteś pewien”: po naciśnięciu przycisku **[x]** scena znika. Jeżeli usunąłeś ją przez pomyłkę — możesz się z tej operacji wycofać, wywołując polecenie **Undo** (**Ctrl-Z**)



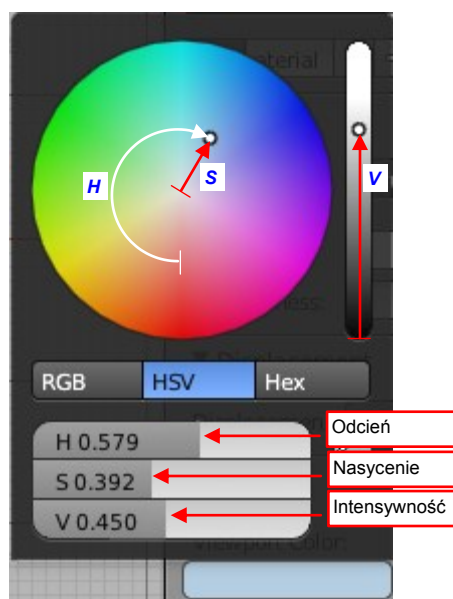
Rysunek 6.7.3 Usunięcie aktualnej sceny

## 6.8 Kontrolka wyboru barwy

Kontrolka służąca do ustalania koloru jest w Blenderze niewielkim prostokątnym polem. Aby zmienić kolor, który wyświetla, kliknij w nią **LPM** (Rysunek 6.8.1):



Rysunek 6.8.1 Kontrolka w trybie **RGB**



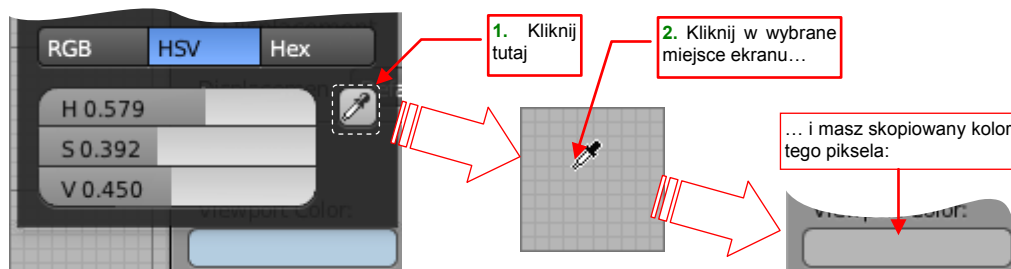
Rysunek 6.8.2 Kontrolka w trybie **HSV**

Spowoduje to rozwinięcie chwilowego okna (menu?) z zestawem kontrolki służących do zmiany barwy. W górnej części okna znajduje się koło barw i pasek intensywności. Wykorzystaj je do ustalenia koloru za pomocą myszki. Jeżeli jednak chcesz wpisać dokładną wartość numeryczną — możesz skorzystać z zestawu pól umieszczonych poniżej. Blender oferuje trzy tryby wpisywania wartości koloru: **RGB**, **HSV**, i **Hex**. W trybie **RGB** (Rysunek 6.8.1) możesz zmieniać wprost intensywności składników **R**, **G**, **B** barwy. Są one tu wyrażone jako liczby z zakresu od 0.0 do 1.0. Niezależnie od tego, którego zestawu kontrolki użyjesz, Blender na bieżąco aktualizuje pole z którego rozwinęliśmy to okno, wyświetlając tam bieżący kolor.

Parametry z zestawu **HSV** są alternatywną metodą opisu koloru (Rysunek 6.8.2):

- **H** (Odcień — ang. *Hue*): determinuje barwę. Ta wartość odpowiada poruszaniu się „wzdłuż spektrum barw”, czyli w naszym przypadku po obwodzie koła pokazywanego przez Rysunek 6.8.2;
- **S** (Nasycenie — ang. *Saturation*): Wartość 0.0 to zupełny brak barwy (wszystko jest w odcieniach szarości), a 1.0 — czyste barwy. Ta wartość odpowiada przesuwaniu punktu wzdłuż promienia koła barw (Rysunek 6.8.2);
- **V** (Intensywność — ang. *Value*): determinuje „jasność” barwy (widać ją na suwaku po prawej);

Czasami wygodniej jest po prostu wskazać jakąś istniejącą barwę. Służy do tego przycisk z „pipetką”. Gdy klikniesz w niego myszką, okno zniknie, a kursor zmieni postać (Rysunek 6.8.3):



Rysunek 6.8.3 Podstawienie barwy wskazanego piksela ekranu

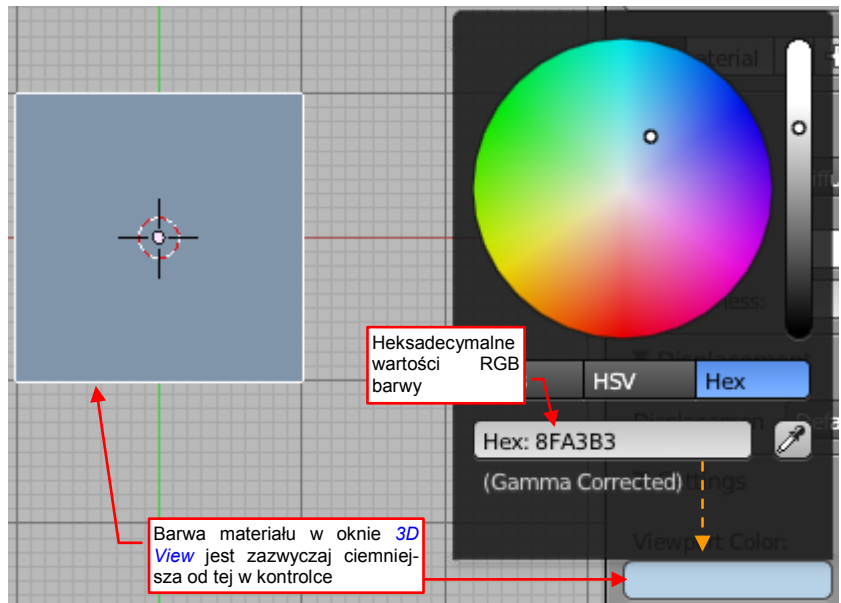
Końcówką tej „pipetki” kliknij w dowolne miejsce ekranu, a w polu kontrolki pojawi się jego kolor.

Wreszcie ostatni tryb określania barwy: **Hex** (heksadecymalny). Przydaje się, gdy chcesz np. przenieść kolor z innego programu, np. z GIMP (tam to się nazywa *HTML notation*). Kopiujesz w nim do schowka 6-znakowy kod barwy, a następnie wklejasz w tym trybie do okna Blendera (Rysunek 6.8.4):

Zwróć uwagę, że np. kolor materiału, przypisany do obiektu, jest w oknie **View 3D** ciemniejszy niż na kontrolce. To przede wszystkim wpływ domyślnego oświetlenia OpenGL, używanego przez Blender w trybie **Viewport Shading: Solid** (por. str. 184, Rysunek 6.1.10).

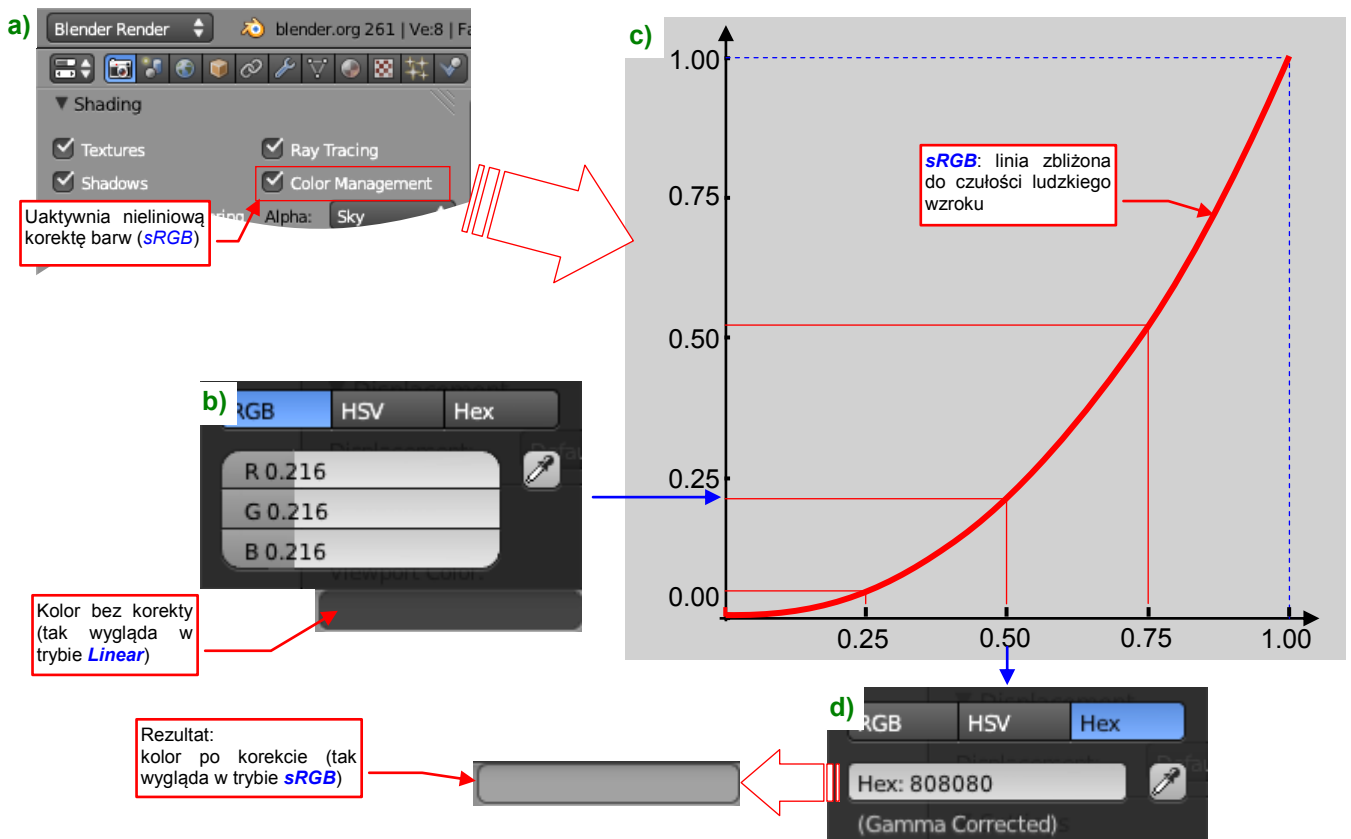
Inną przyczyną może być także brak tzw. „korekty gamma”. Pod tą tajemniczą nazwą kryje się proces lepszego dopasowania zakresu wyświetlanych barw do czułości ludzkiego oka. Domyślnie Blender wyświetla barwy bez żadnej korekty (ten tryb określany jest jako **Linear**).

Jeżeli jednak w panelu **Shading**



Rysunek 6.8.4 Kontrolka w trybie Hex

zestawu **Render** włączysz opcję **Color Management** (Rysunek 6.8.5a), wówczas wyświetlane kolory ulegną rozjaśnieniu. Przyczyną jest nieliniowa konwersja intensywności wejściowej barwy (Rysunek 6.8.5b) poprzez krzywą gamma (nazywaną także **sRGB** — Rysunek 6.8.5c) na barwę wyświetlaną (Rysunek 6.8.5d):



Rysunek 6.8.5 Efekt korekty gamma (sRGB): nieliniowe rozjaśnianie barw

Należy pamiętać, że ta konwersja działa także w drugą stronę. Gdy wkleisz do okna koloru w trybie **Hex** tekst „808080”, odpowiadający mieszance 50% czerni i bieli, to w trybie **Color Management** Blender wcale nie przyjmie dla niego wartości  $R=G=B=0.5$ , jak można by się było spodziewać (por. str. 468). Zamiast tego zobaczysz

**R=G=B=0.216** (por. Rysunek 6.8.5b). I mimo to wyświetlany w kontrolce kolor dokładnie odpowiada temu, który uzyskasz dla kodu hex. „808080” w GIMP! Dlaczego? Bo korekta gamma jest wbudowana w system operacyjny. I Windows, i Linux używają takiej samej krzywej **sRGB**. Dla większej części jej zakresu (gdzieś tak od 0.25 do 1.0) jej współczynnik gamma = 2.2. Mac OS używa krzywej o nieco innym współczynniku: 1.8, więc dla tych samych wartości heksadecymalnego kodu odcień koloru na ekranie komputera od Apple może wyglądać nieco inaczej. (Przypuszczam jednak, że **Color Management** Blendera uwzględnia i takie przypadki, dokonując odpowiedniej konwersji).

- W pracy nad modelem opisanym w tej książce **Color Management** był przez cały czas włączony. (Dokładniej — od rozdziału o materiałach: por. Wyciąg III).

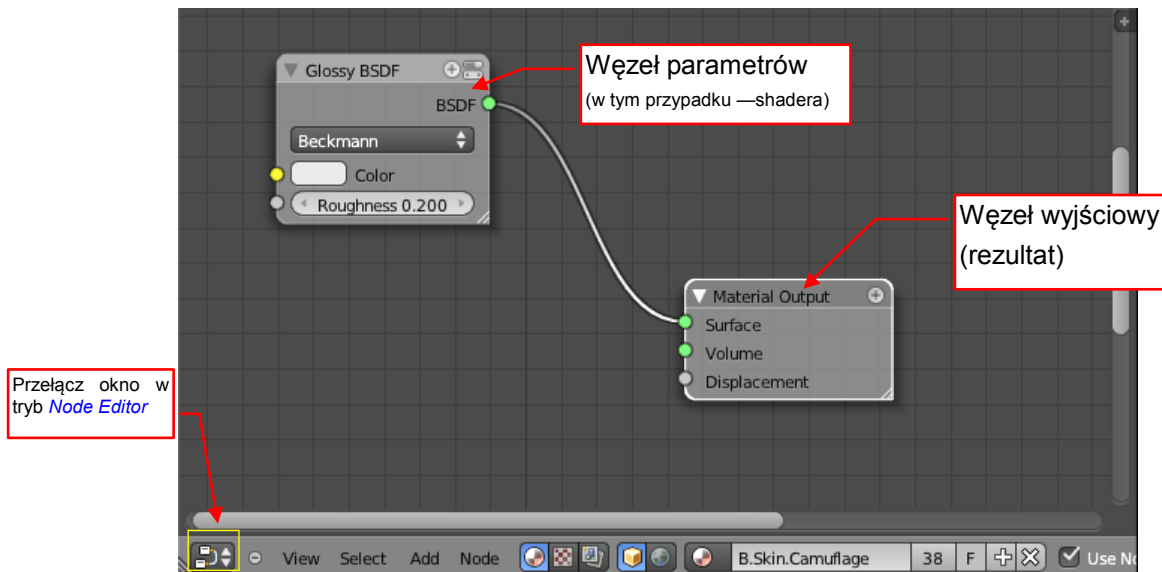
- W węzłach Cycles (np. **Image Texture**) włączasz korektę gamma (**sRGB**) wybierając tryb **Color**. Dla tekstur nie odwzorowujących barwy (np. nierówności) używa się trybu bez żadnej korekty: **Non-Color Data**.



## 6.9 Edytor węzłów (*Node Editor*)

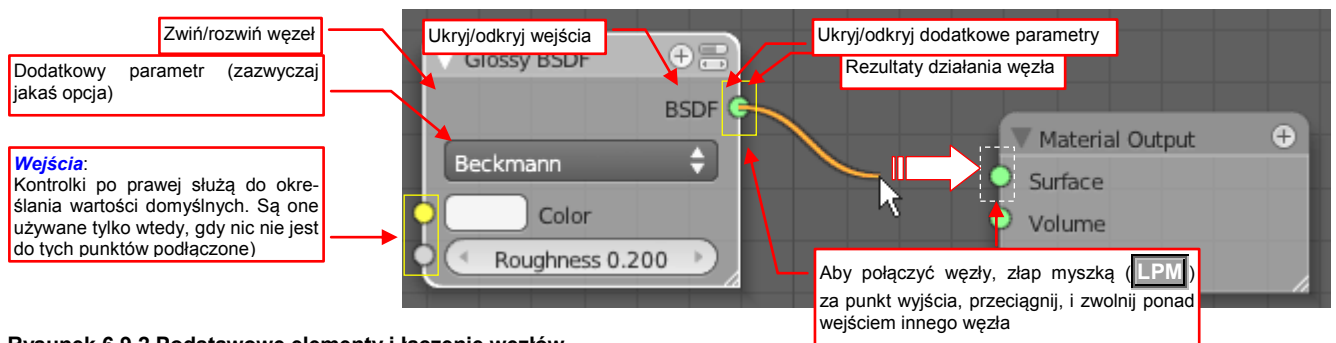
Węzły, podobnie jak znane Ci do tej pory kontrolki w panelach czy oknach dialogowych, służą do zmiany jakichś parametrów. To po prostu inna forma prezentacji. Tradycyjny interfejs użytkownika udostępnia w oknie/oknach wszystkie możliwe opcje naraz. Zazwyczaj podczas pracy zmieniasz tylko niektóre z nich. Można nazwać tę formę "statyczną". Węzły stosuje się tam, gdzie wielka liczba dostępnych opcji czyni „statyczną” prezentację w oknie lub oknach niemożliwą. Każdy węzeł to jakiś pojedynczy zespół parametrów. Budując z węzłów schemat, "dynamicznie" dobierasz tylko te kontrolki, których wartości chcesz zmienić. W związku z tym musisz się trochę orientować, jakie komponenty (węzły) masz do dyspozycji i co każdy z nich robi. To pewna wada. Z drugiej strony, węzły można łączyć na najróżniejsze sposoby, których uzyskanie nie byłoby praktycznie możliwe w tradycyjnym, "statycznym" GUI.

Gdy przełączysz jedno z okien Blendera w tryb *Node Editor*, zobaczysz widok przypominający Rysunek 6.9.1:



Rysunek 6.9.1 Okno *Node Editor* (aktualny kontekst: materiał Cycles)

Edytor węzłów może służyć do pracy z definicjami materiałów, tekstur lub kompozycji ostatecznego obrazu. Rysunek 6.9.1 przedstawia definicję materiału o nazwie **B.Skin.Camouflage**. To prosty schemat, składający się z dwóch węzłów: shadera (*Glossy BSDF*) połączonego z rezultatem (*Material Output*). Z lewej strony każdego węzła znajdują się punkty wejścia, z prawej — punkty wyjścia. W *Node Editor* budujesz definicję, łącząc wyjścia jednych węzłów z wejściami innych. Rysunek 6.9.2 przedstawia typowe kontrolki węzłów i sposób ich łączenia:



Rysunek 6.9.2 Podstawowe elementy i łączenie węzłów

Aby połączyć węzły ze sobą, „łapiesz” odpowiedni punkt wyjścia i przeciągasz ponad wejście innego węzła. Blender łączy je eleganckim „kablem” (por. Rysunek 6.9.2). Usunięcie takiego połączenia polega na złapaniu myszką końcówki takiego „kabla” i „upuszczeniu” jej gdziekolwiek poza węzłem. Możesz także „przeciąć” te linie (**Ctrl**-**LPM**).

Węzły możesz zaznaczać tak samo jak obiekty w oknie **3D View**: Kliknięcie **PPM** (a także **LPM**) zaznacza pojedynczy węzeł. Skrót **A** (**View→Select or Deselect All**) zaznacza/wyłącza zaznaczenie wszystkich węzłów schematu. Dostępne jest także zaznaczenia obszarem prostokątnym (**B** — **View→Border Select**).

Aby przesunąć pojedynczy węzeł, wystarczy go „złapać” **LPM** i przeciągnąć w nowe miejsce. Tak jak w oknie **3D View** dostępne są także skróty **G** (przesunięcie), **S** (skalowanie) i **R** (obróć). Szczególnie te dwie ostatnie operacje mają sens jedynie dla dwóch lub więcej węzłów. (Skalowanie zbliża lub oddala zaznaczone węzły, a obrót obraca ich środki).

Aby wstawić do schematu nowe węzły, wybierasz je z menu **Add** (z nagłówka okna **Node Editor**). W każdym trybie pracy zawartość tego menu jest inna. (W każdym z tych trybów używane są inne węzły). Tak jak obiekty, węzły mogą być kopiowane (**Shift-D** — **Node→Duplicate**).

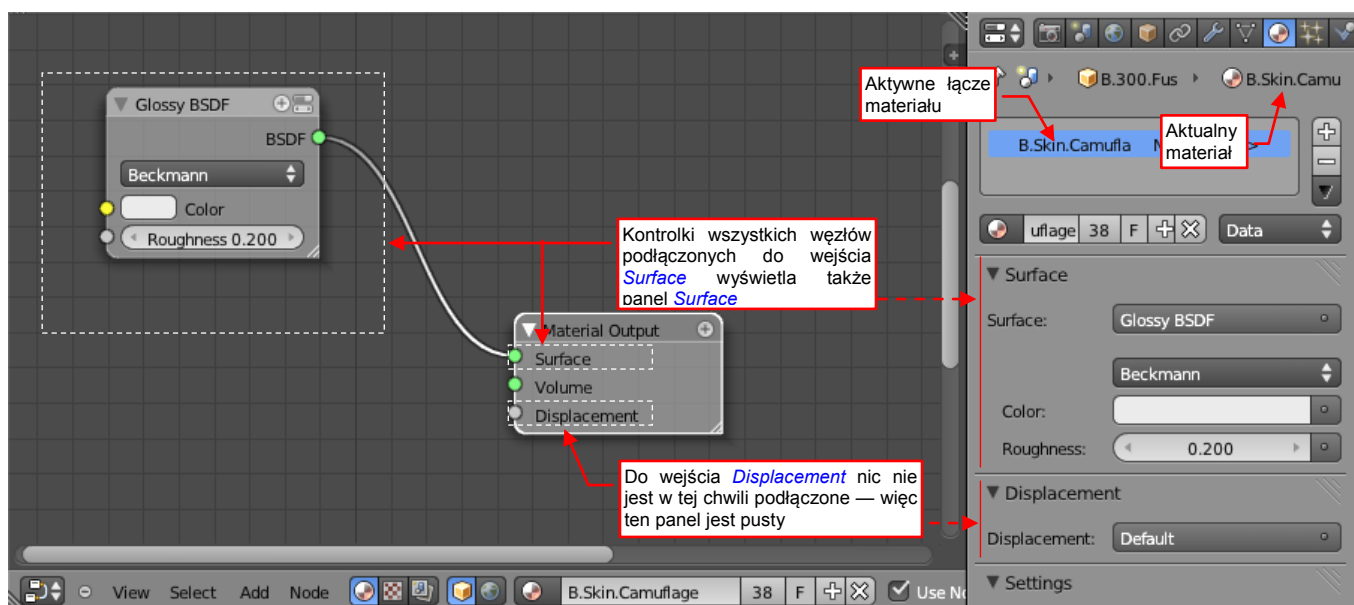
Aby usunąć jakiś węzeł, zaznacz go i naciśnij **Delete** lub **X** (**Node→Delete**).

Aby przesunąć ekran (**Pan**) na inny fragment schematu, przesuń myszkę z wciśniętym **SPM**.

Powiększenie można zmienić tak samo jak w oknie **3D View**: obracając **KM** lub przesuając myszkę z wciśniętym **Ctrl** i **SPM**. Działa tu także skrót **Home** (**View→View All**).

- Schemat materiału lub kompozycji, który budujesz w **Node Editor**, będzie wykorzystany do przetworzenia każdego piksela renderowanego obiektu lub obrazu.

Specyfiką Cycles jest dodatkowe wyświetlanie kontrolek węzłów w panelach okna **Properties** (zestawu **Material** — Rysunek 6.9.1). Każda z tych paneli wyświetla kontrolki wszystkich węzłów podłączonych do odpowiednich wejść (**Surface**, **Displacement**) węzła **Material Output** (Rysunek 6.9.3):

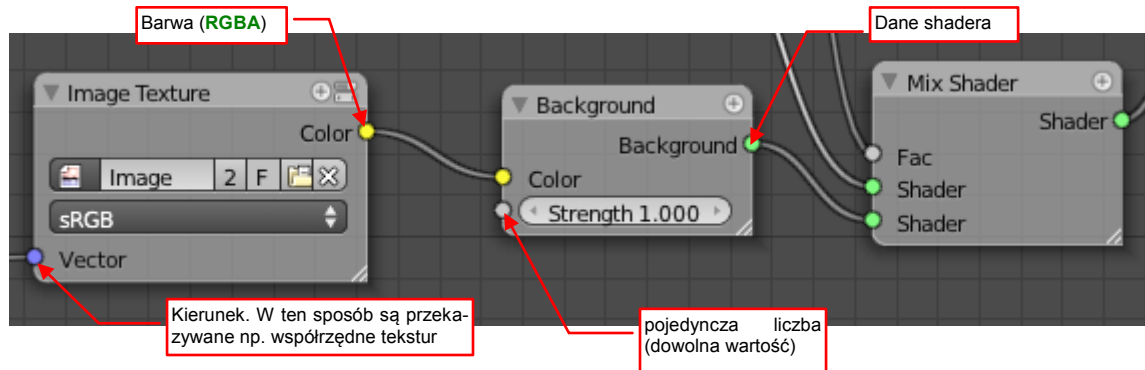


Rysunek 6.9.3 Powtórzone kontrolki węzłów materiału w panelach **Surface** i **Displacement** (Cycles)

To powiązanie powoduje, że dla nieco bardziej złożonych schematów panele gwałtownie się rozrastają — i liczba wyświetlanych w niej kontrolki czyni je praktycznie bezużytecznymi. Możesz je zwinąć, aby nie przeszkadzały. Sądzę, że ta właściwość przydaje się w dwóch przypadkach:

- dla bardzo prostych materiałów — niedoświadczeni użytkownicy mogą się czuć pewniej, używając panelu, a nie edytora węzłów;
- do obsługi parametrów materiału, który jest oparty na grupie węzłów;

Kolor punktów wyjścia i wejścia węzła zależy od jego typu danych (Rysunek 6.9.4):



Rysunek 6.9.4 Oznaczenia barwne typów wejść i wyjść węzłów

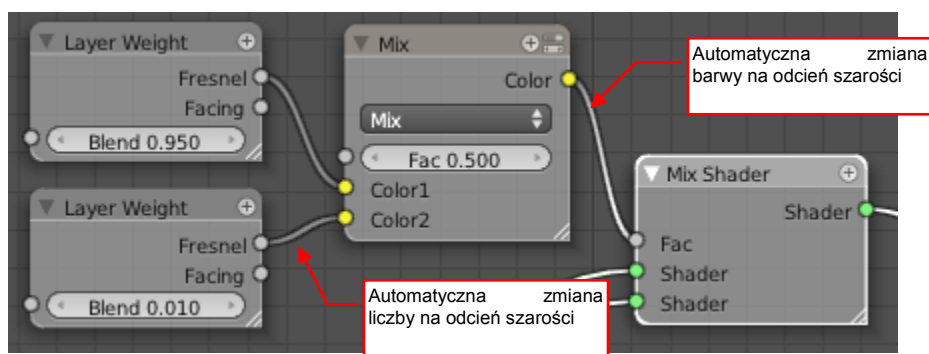
Żółty punkt oznacza barwę (*R, G, B, A*), szary — pojedynczą liczbę (rzeczywistą), zielony — informacje dla shadera, a niebieski — wektor (*X, Y, Z*), determinujący położenie lub kierunek. Wektory są zazwyczaj potrzebne na wejściu węzłów tekstur: określają, jak dwuwymiarowy obraz tekstury ma być rozłożony na trójwymiarowej powierzchni.

Istnieją specjalne typy węzłów, które służą do konwersji z jednego rodzaju danych na drugi. Przykładem takiego węzła jest chociażby *RGB to BW*, (*RGB to Black/White*). Można go użyć do przekształcenia koloru na pojedynczą liczbę (od 0.0 do 1.0) (Rysunek 6.9.5):



Rysunek 6.9.5 Przykład jawnej konwersji typu połączenia (tutaj: z barwy na liczbę)

*Node Editor* pozwala także po prostu łączyć ze sobą węzły różnych typów (Rysunek 6.9.6):



Rysunek 6.9.6 Przykłady automatycznej konwersji typów

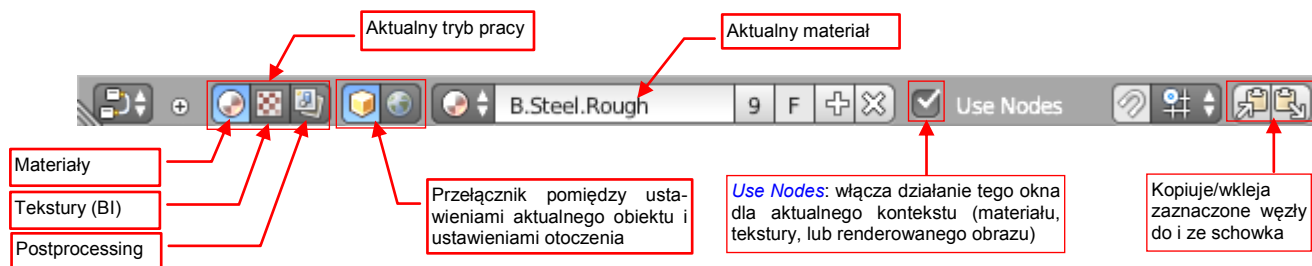
Wzdłuż takiego połączenia Blender dokonuje automatycznej konwersji z jednego typu wartości na drugi. Reguły tej automatycznej konwersji przedstawia Tabela 6.9.1 (pola na przekątnej nie są konwertowane):

Na końcu \ Na początku	<i>Value</i>	<i>Color</i>	<i>Vector</i>	<i>Shader</i>
<i>Value</i>	v	v→R, v→G, v→B	v→X, v→Y, v→Z	brak
<i>Color</i>	Odcień szarości	RGBA	R→X, G→Y, B→Z	brak
<i>Vector</i>	$(X+Y+Z)/3$	X→R, Y→G, Z→B	XYZ	brak
<i>Shader</i>	brak	brak	brak	—

Tabela 6.9.1 Automatyczna konwersja danych na połączeniu pomiędzy różnymi rodzajami wejść i wyjść węzłów

Zwróć uwagę, że nie istnieje żadna automatyczna konwersja na / z danych shadera.

Edytor węzłów można uruchamiać w trzech różnych kontekstach (trybach pracy): materiału, tekstury, i końcowej kompozycji obrazu (*postprocessing*). Przełącznik tych trybów znajduje się w nagłówku okna (Rysunek 6.9.7):



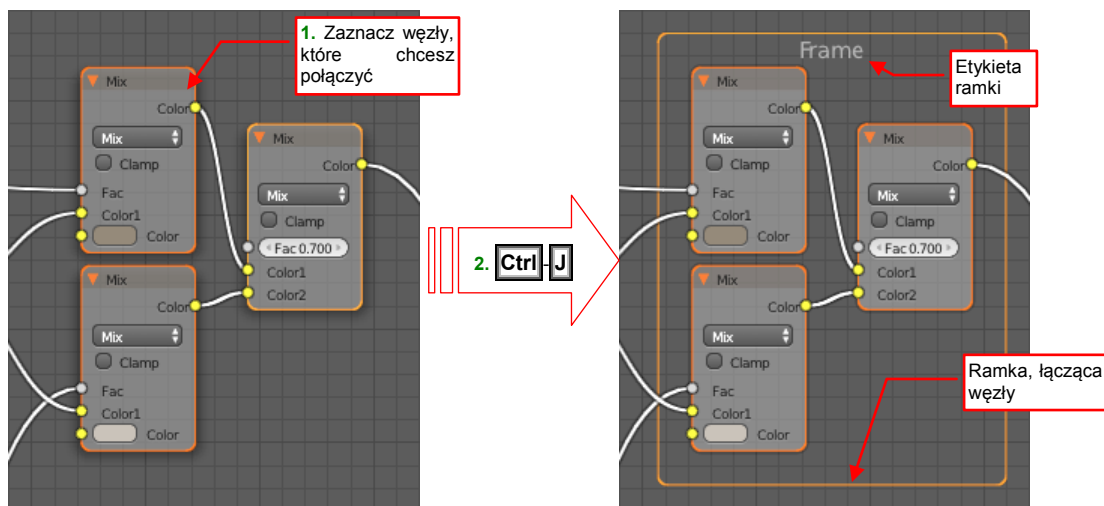
Rysunek 6.9.7 Nagłówek okna **Node Editor** (aktualny kontekst: materiał Cycles)

Każdy z tych trybów oferuje inny zestaw węzłów i służy do czego innego. Rysunek 6.9.7 przedstawia nagłówek w trybie edycji materiału. W tej książce będziemy także wykorzystywać tryb kompozycji (*postprocessing*). Trybu tekstur nie będziemy używać, gdyż jest przeznaczony wyłącznie dla renderera BI (W Cycles tekstury można komponować w trybie materiału).

Zwróć także uwagę, że w trybie materiału w nagłówku okna **Node Editor** pojawia się z prawej strony opcji wyboru trybu jeszcze dodatkowy przełącznik (Rysunek 6.9.7). Umożliwia przełączanie pomiędzy definicją aktualnego materiału (materiału przypisanego do aktywnego łącza siatki<sup>1</sup> — por. Rysunek 6.9.3) i definicją otoczenia sceny.

Sporą pomocą przy pracy nad złożonymi schematami są przyciski **Copy/Paste**, umieszczone z prawej strony nagłówka (Rysunek 6.9.7). Pozwalają na kopiowanie i wklejanie całych zespołów węzłów.

Inną pomocą w pracy nad węzłami jest możliwość łączenia węzłów w ramki (Rysunek 6.9.8)



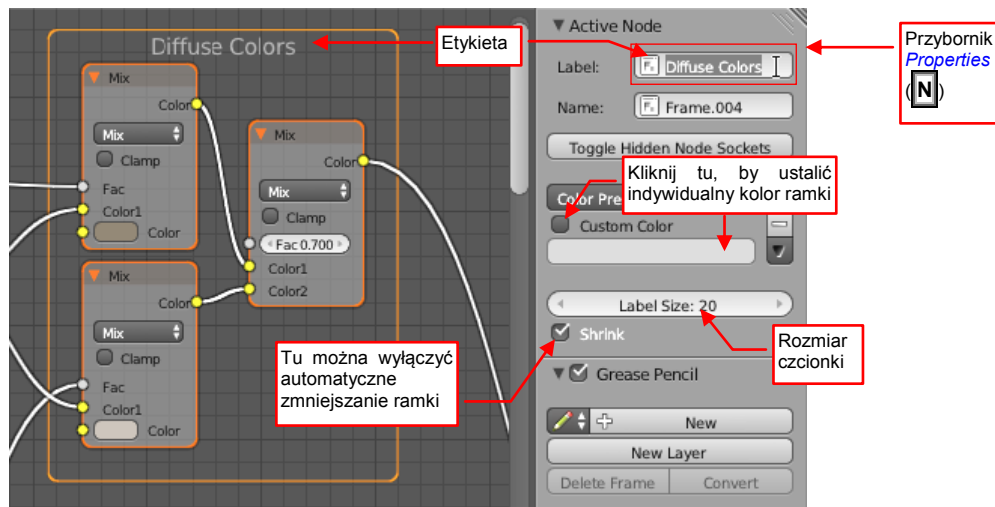
Rysunek 6.9.8 Łączenie węzłów w ramki

Zaznacz węzły, które chcesz połączyć w ramkę i naciśnij **Ctrl-J** (**Node → Join in new Frame**). Blender otoczy wówczas zaznaczone elementy ramką. To prostokąt z zaokrąglonymi narożnikami i umieszczoną etykietą pod górną krawędzią (domyślnie jest tam wpisane „Frame”). Ramki traktuj jak pomocnicze „skrzynki”. Możesz je „złapać”, naciskając **LPM** w jej wnętrzu, i trzymając wciśnięty ten klawisz myszy przeciągnąć w nowe położenie. Ramka zawsze przesuwa się wraz z całą zawartością. To często wygodniejsze niż zaznaczanie wielu węzłów.

<sup>1</sup> Lub definicji materiału przypisanego do źródła światła — w Cycles lampy mogą mieć swoje materiały, a także dowolny obiekt (siatka) może być źródłem światła. (W Cycles światło emituje wszystko, co ma przypisany shader *Emission*).



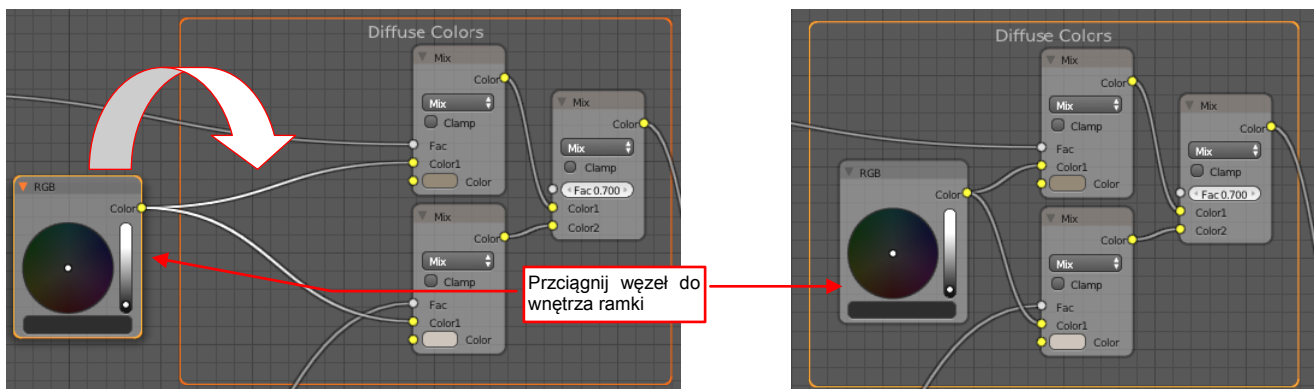
Gdy zaznaczysz ramkę i naciśniesz przycisk **N** (*View→Properties*), *Node Editor* wyświetli po prawej stronie ekranu odpowiedni przybornik. W panelu **Active Node** możesz zmienić właściwości ramki (Rysunek 6.9.9):



Rysunek 6.9.9 Właściwości ramki

Uważam że jedną z najważniejszych elementów ramki jest jej etykieta. Zawsze ją zmieniam, nadając ramce jakąś nazwę odpowiednią do roli węzłów, które zawiera. To bardzo pomaga w późniejszej analizie schematów!

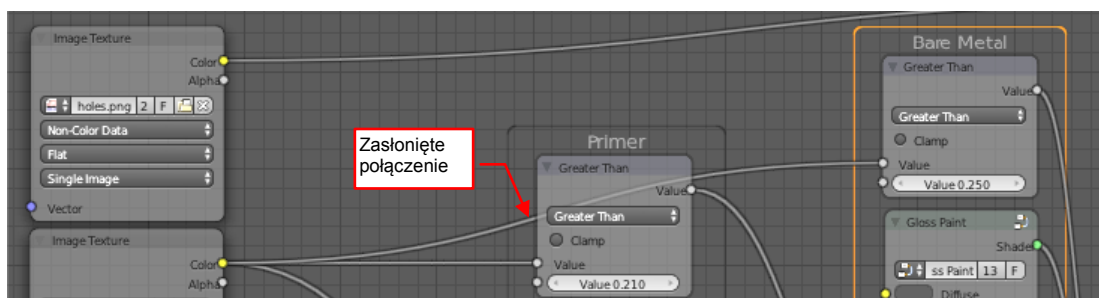
Aby włączyć jakiś węzeł do istniejącej ramki, po prostu przeciągnij go do jej wnętrza (Rysunek 6.9.10):



Rysunek 6.9.10 Dodawanie węzłów do ramki

Operacja odwrotna nie jest tak intuicyjna: aby „wyjąć” węzły z ramki musisz je zaznaczyć i wywołać polecenie *Node→Remove from Frame*. Czasami prościej jest po prostu usunąć całą ramkę (zaznacz ją i naciśnij **X** lub **Del**). To usuwa tylko obramowanie — zawartość ramki (węzły, połączenia) pozostaje na schemacie.

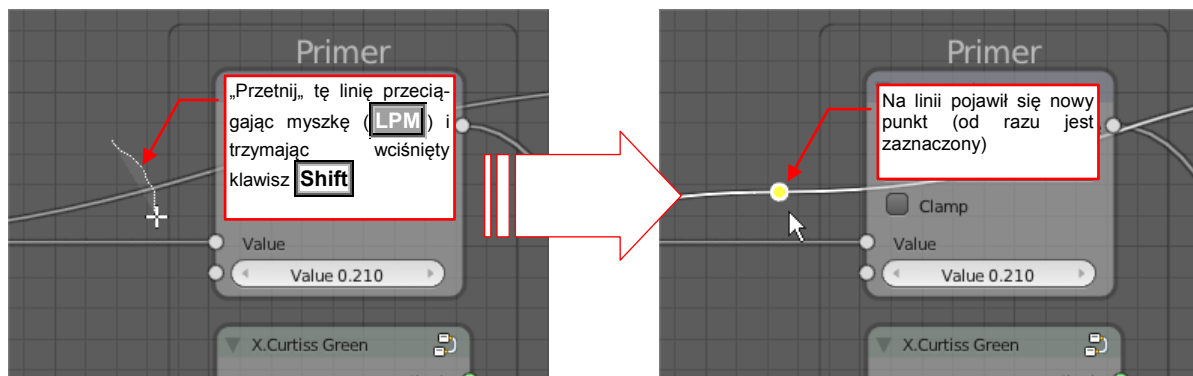
Czasami krzywa linia łącząca dwa punkty bywa przesłonięta przez inne węzły (Rysunek 6.9.10):



Rysunek 6.9.11 Połączenie, zasłonięte przez inny węzeł

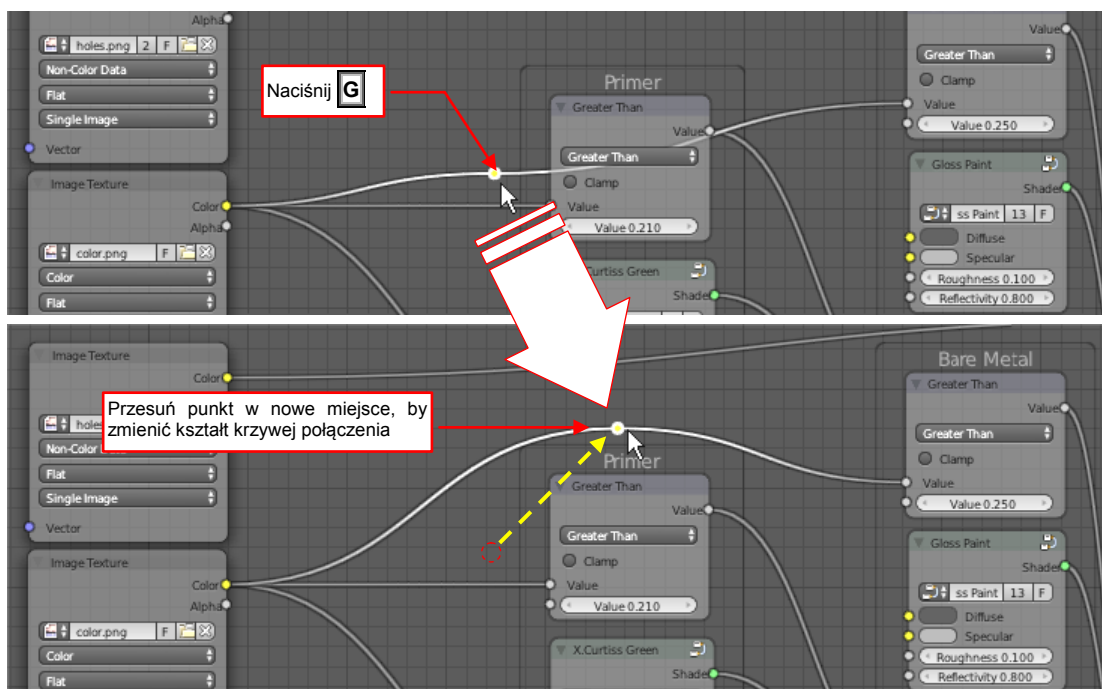
To zmniejsza czytelność schematu. Czy można wymusić inny przebieg tej linii połączenia?

Do sterowania kształtem linii połączeń używamy w Blenderze punktów pomocniczych. Aby je wstawić, musisz „przeciąć” linię myszką, trzymając na klawiaturze wciśnięty klawisz **Shift** (Rysunek 6.9.12):



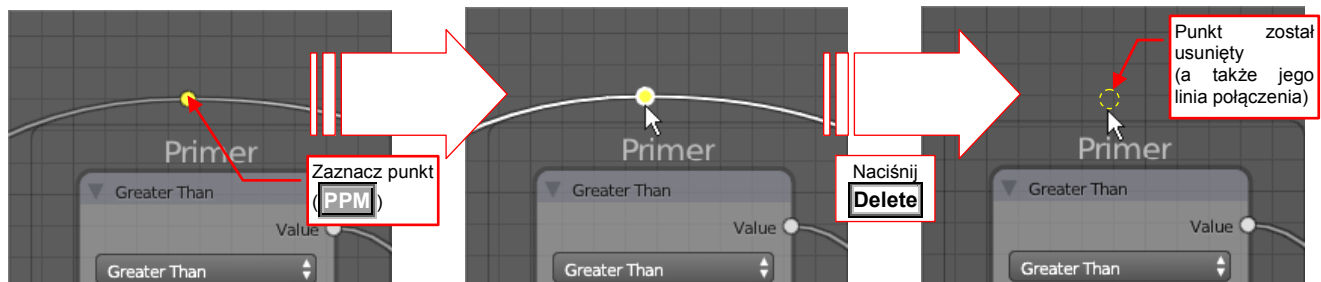
Rysunek 6.9.12 Wstawianie pomocniczego punktu w linię połączenia

Blender wstawia w miejscu przecięcia nowy punkt pomocniczy. Ten punkt od razu jest zaznaczony (ma białą obwódkę). Wystarczy nacisnąć **G** i przesunąć go w nowe miejsce (Rysunek 6.9.13):



Rysunek 6.9.13 Przesuwanie pomocniczego punktu

Aby skończyć przesuwanie punktu, kliknij **LPM** (to działa tak samo jak przesuwanie zwykłego węzła). Aby usunąć jakiś punkt pomocniczy musisz go zaznaczyć (**PPM**) i nacisnąć **X** lub **Delete** (Rysunek 6.9.14):



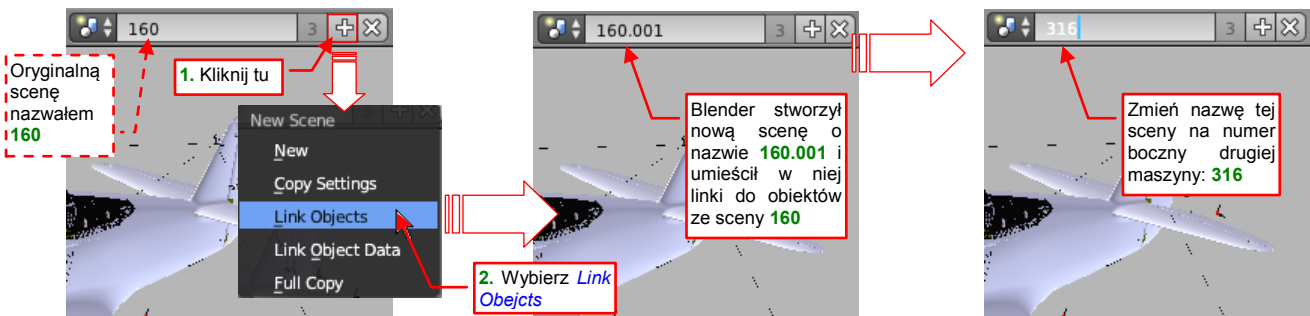
Rysunek 6.9.14 Usuwanie pomocniczego punktu

Usunięcie punktu pomocniczego usuwa także jego linię połączenia.

## 6.10 Przykład wykorzystania drugiej sceny

W tej sekcji pokażę, jak wykorzystać drugą scenę w charakterze „tła”. (To coś innego niż podstawianie zdjęcia za modelem). Zrobię to na przykładzie tworzenia wizualizacji z udziałem dwóch samolotów. W tym przypadku będą to dwie maszyny tego samego typu (samoloty myśliwskie często latały parami) .

Na razie w pliku jest jeden odpowiednio „upozowany” samolot, który ma teksturę z numerem bocznym „160”, więc zmieniłem nazwę jego sceny na **160**. Skopiowałem tę scenę, wykorzystując opcję **Link Objects**, a potem zmieniłem jej nazwę na numer drugiej maszyny: **316** (Rysunek 6.10.1):

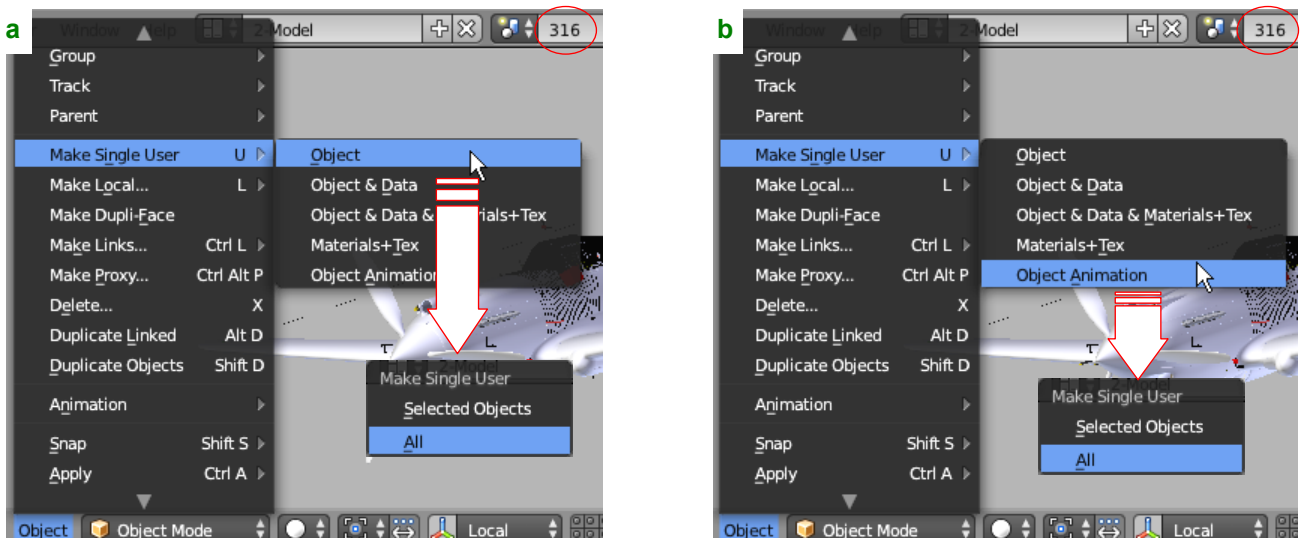


Rysunek 6.10.1 Dodanie nowej sceny (316)

W tej chwili scena **316** zawiera referencje (**Link**) do wszystkich obiektów z oryginalnej warstwy **160**. Na pierwszy rzut oka niczym się nie różnią poza nazwą. Ustalmy, że sceną główną (tą, na której będziemy przesuwając model samolotu) będzie **316**. Dla porządku proponuję usunąć ze **160** wszystkie obiekty z warstwy **20** (kamera, źródła światła) oraz obiekt **Target** z warstwy **10**. (Potrzebujemy ich tylko na scenie **316**).

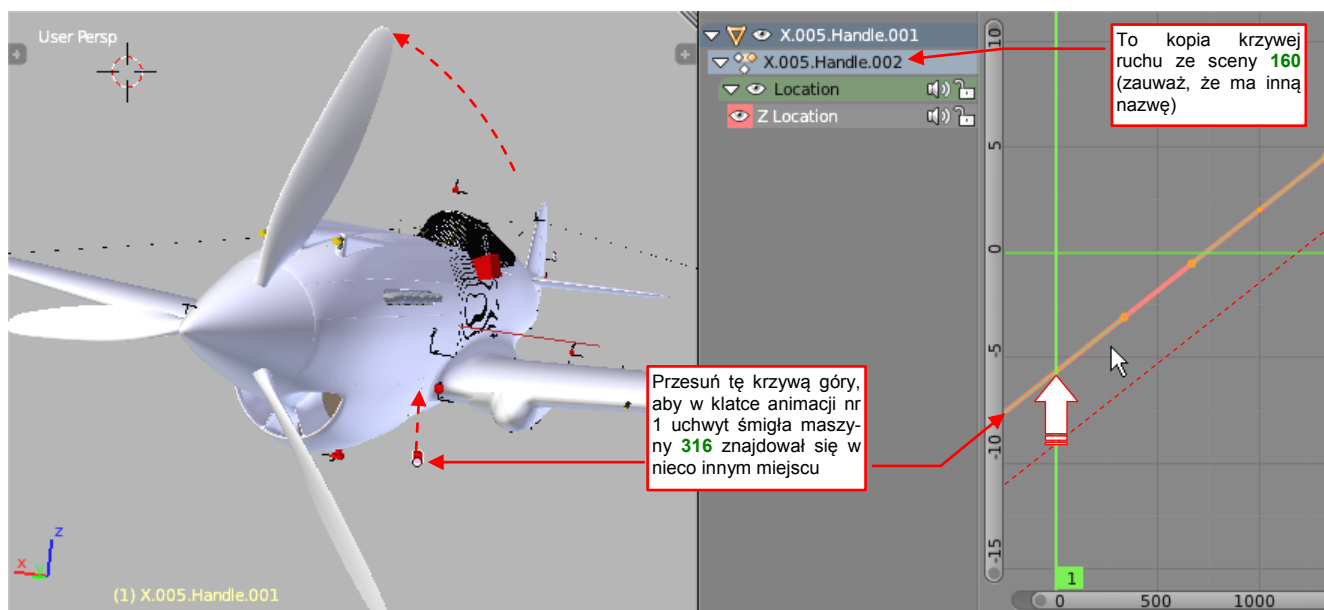
Potem możemy stworzyć w **316** lokalne kopie wszystkich obiektów i krzywych animacji (ale nie siatek czy innych skojarzonych z nimi danych!). W tym celu wywołaj w **316** z menu **Object→Make Single User** polecenia:

- **Object→All** (Rysunek 6.10.2a) — to utworzy lokalne kopie wszystkich obiektów sceny **316**. Blender zmieni odwołania we wszystkich ograniczeniach i innych referencjach do innych obiektów. Oznacza to że uchwyty poruszające poszczególnymi częściami samolotu na każdej ze scen można ustawiać niezależnie. Obiekty z obydwu scen współdzielą jednak wszystkie bloki danych, takie jak siatki, materiały czy krzywe ruchu z obiektami;
- **Object Animation→All** (Rysunek 6.10.2b) — to stworzy lokalne kopie linii ruchu śmigła, przypisane do każdej ze scen;



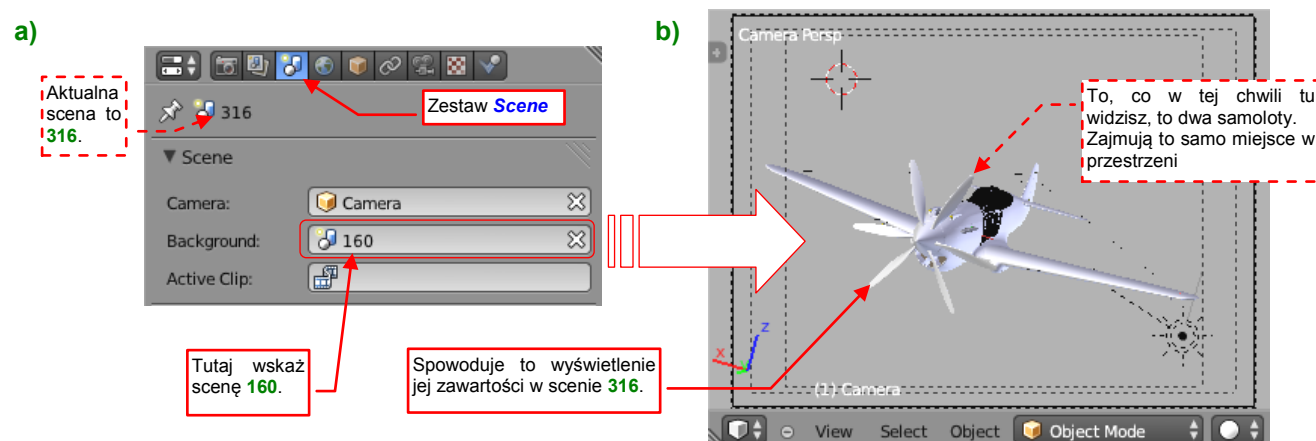
Rysunek 6.10.2 Stworzenie lokalnych kopii obiektów i ich linii ruchu (**F-Curve**)

Musieliśmy powielić tory ruchu obiektów (*F-Curves*), aby śmigła renderowanych samolotów nie były ustawione w tej samej pozycji. Teraz przesun tę drugą krzywą ruchu w górę, aby to osiągnąć (Rysunek 6.10.3):



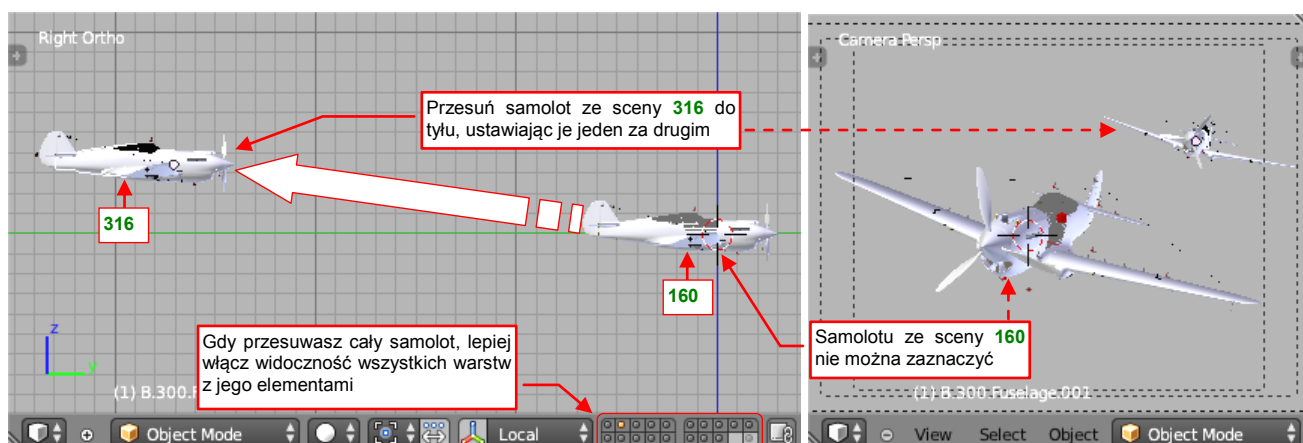
Rysunek 6.10.3 Zmiana położenia łopat śmigła w scenie 316

Teraz wpisz scenę 160 jako tło sceny 316 (w polu *Scene:Background* — Rysunek 6.10.4a):



Rysunek 6.10.4 Przypisanie scenie 316 sceny 160

W efekcie w scenie 316 pojawił się samolot ze sceny 160. Jest tylko wyświetlany (nie można go zaznaczyć), więc spokojnie przesun do tyłu, formując szyk (Rysunek 6.10.5):



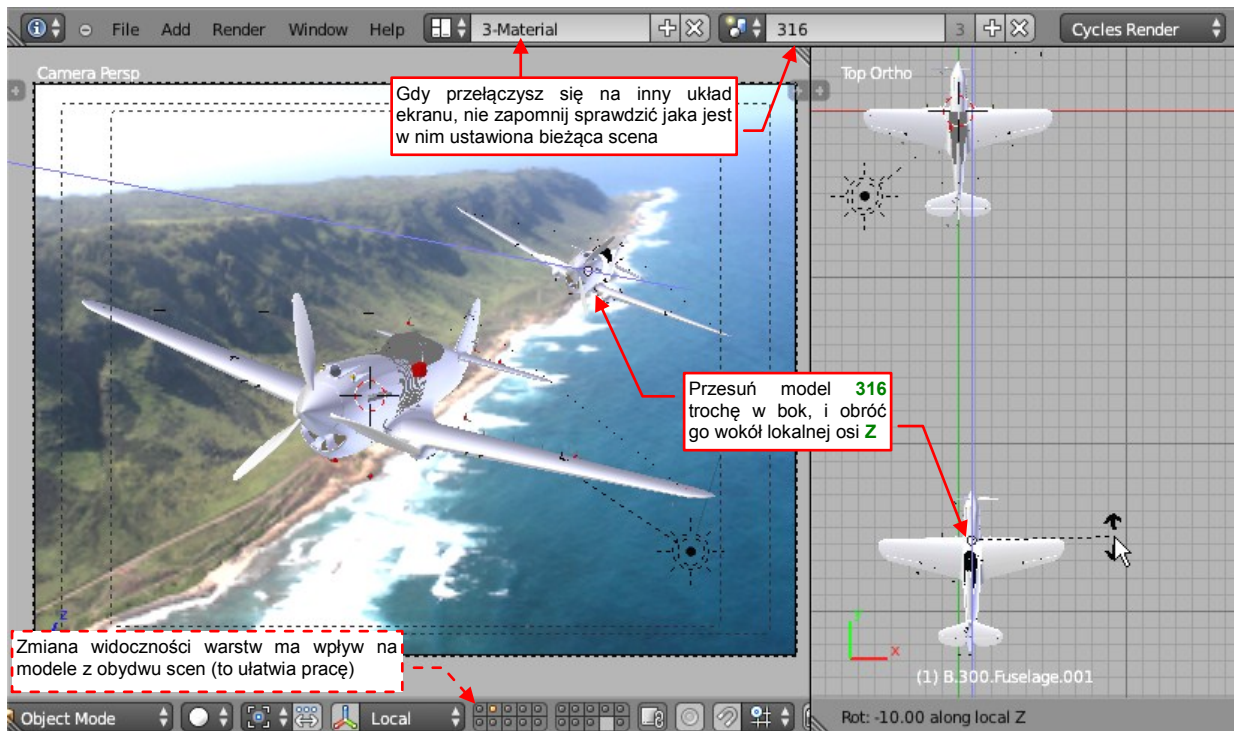
Rysunek 6.10.5 Pozowanie samolotu ze sceny 316



W takim momencie przydało się, że początkiem hierarchii wszystkich elementów naszego modelu jest jeden obiekt — kadłub (**B.300.Fuselage**). Jego odpowiednik w scenie **316** nosi nazwę **B.300.Fuselage.001** (Blender nadał wszystkim kopiom końcówki „.001”). Dzięki temu nie musisz zaznaczać wszystkich elementów, wystarczy że odpowiednio ustawisz ten jeden obiekt.

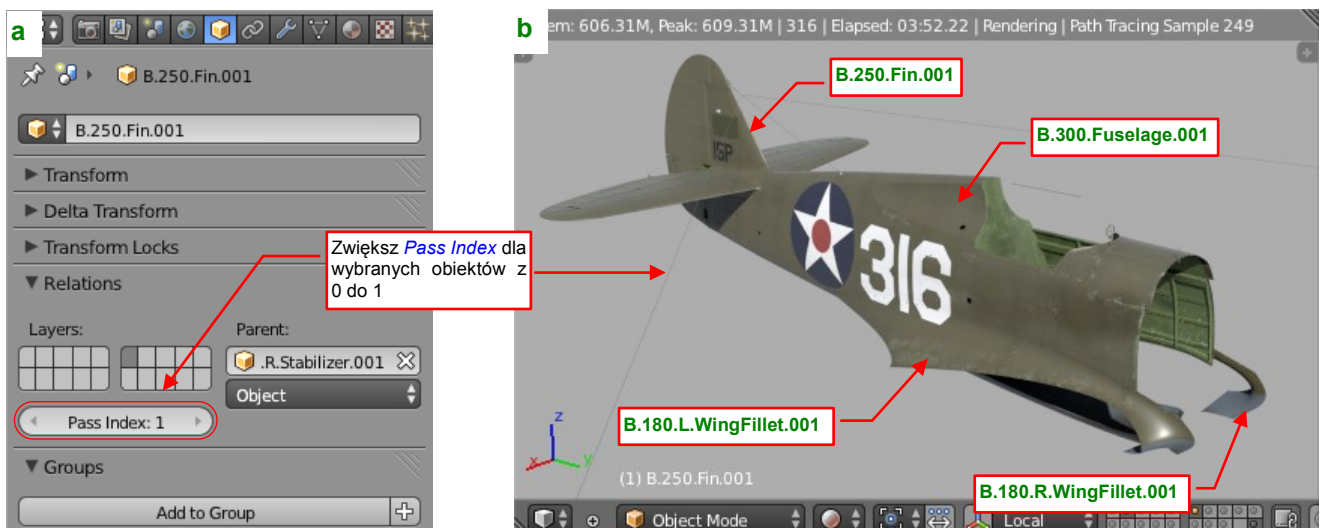
- Jeżeli przesuwany obiekt miał jakieś obiekty potomne na warstwach ukrytych podczas tej transformacji, to po ich odkryciu możesz zobaczyć ich zawartość w starym miejscu. Wystarczy jednak wykonać wtedy jakiś drobny ruch obiektem nadrzędnym, by wszystkie „wskoczyły” na swoje miejsce. To jakiś błąd w Blenderze.

Odsuń samolot **316** trochę w bok od toru lotu **160**, i pochyl nieco jego skrzydła (Rysunek 6.10.6). (To dodaje do sceny jakiś drobny element dramatyzmu — samoloty lecące w równym szyku kojarzą się z siłą i spokojem):



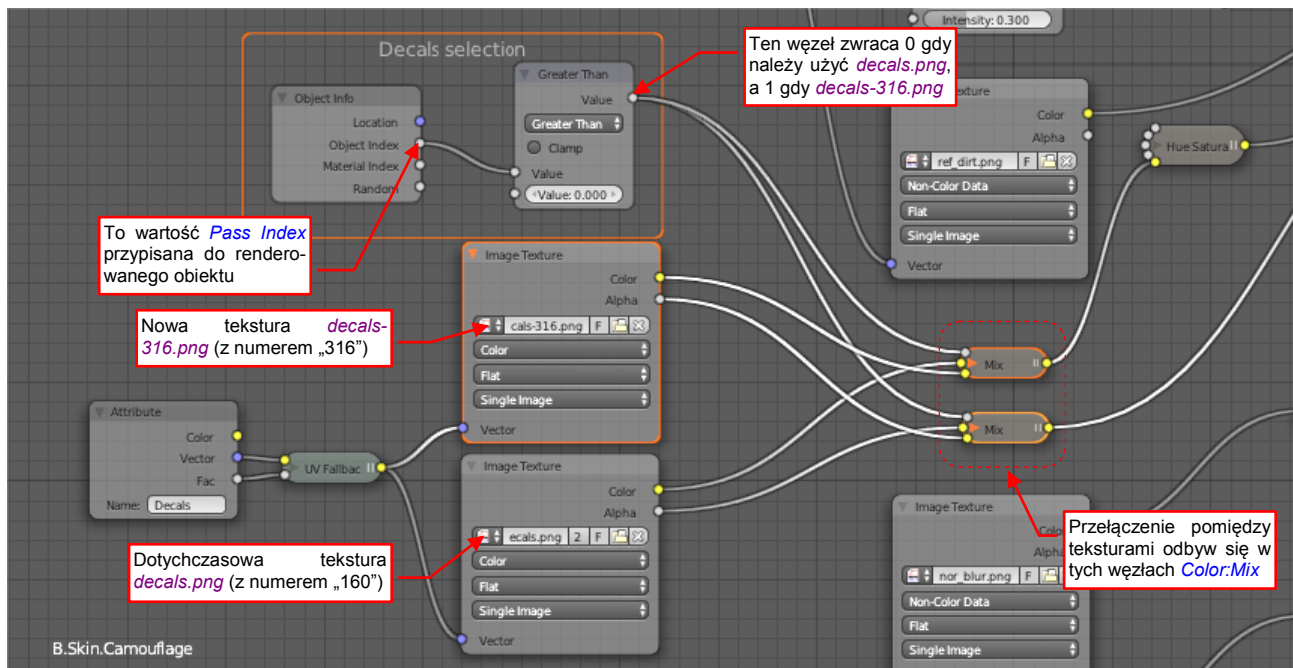
Rysunek 6.10.6 Pozowanie samolotu ze sceny **316**, c.d.

Jak umieścić na drugiej maszynie inne numery taktyczne? Przygotuj alternatywną teksturę (*decals-316.png*) i sprawdź jak wygląda na modelu (Rysunek 6.10.7b). Następnie zmień obiektom na których namalowano numery taktyczne wartość *Relations:Pass Index* z domyślnego 0 na 1 (Rysunek 6.10.7a):



Rysunek 6.10.7 Nowe numery taktyczne na drugim samolocie

**Pass Index** pozwala wyróżnić obiekty podczas renderowania. Teraz w materiale **B.Skin.Camouflage** (por. str. 126) wprowadźmy zmiany, które dla obiektów z **Pass Index** = 0 podstawia piksele z tekstury **decals.png**, a dla tych z **Pass Index** = 1 — alternatywną **decals-316.png**. W miejsce dotychczasowej pojedynczej tekstury umieść dwie, których wejścia są połączone przełącznikami **Color:Mix** (Rysunek 6.10.8):



Rysunek 6.10.8 Modyfikacja materiału poszycia, różnicująca oznaczenia taktyczne na samolotach

Współczynnik „mieszania” w węzłach **Mix** jest podłączony do wyjścia **Object Index** z węzła **Object Info**. (**Object Index** to samo co **Pass Index** — drobna niezgodność terminologii). Dla porządku „przepuszczam” tę wartość przez węzeł **Greater Than**, aby współczynnik w węzłach **Mix** na pewno miał zawsze wartość 0 lub 1. Dzięki temu ten układ działa jak przełącznik — dla obiektów z **Pass Index** > 0 podstawia teksturę z numerem bocznym „316” (**decals-316.png**).

Rysunek 6.10.9 przedstawia rezultat. (Ustawiłem tu samoloty bokiem, by było widać wyraźnie ich numery taktyczne):



Rysunek 6.10.9 Różne numery taktyczne na dwóch modelach pokrytych tym samym materiałem

---

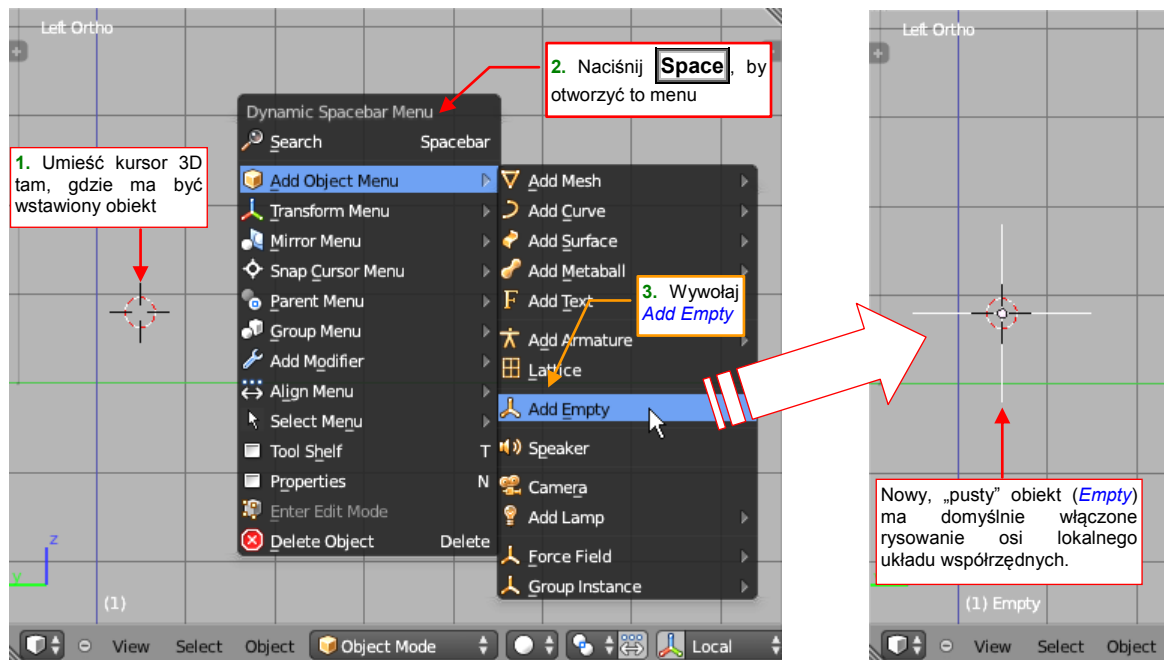
## **Rozdział 7. Blender — edytor obiektów (Object Mode)**

Z edytorem obiektów miałeś już okazję się zetknąć (Wyciągu II, Rozdział 3, "Blender – pierwsze kroki"). W tym rozdziale omówimy podstawowe pojęcia, które się z nim wiążą.

### 7.1 Wstawienie pustego obiektu (*Empty*)

Czasami, podczas pracy z ogranicznikami (*constraints*) przydają się pomocnicze elementy, służące wyłącznie jako „punkty odniesienia” w przestrzeni. Takie obiekty określane są w Blenderze jako „puste” (*Empty*), gdyż nie zawierają żadnej siatki. To samo „opakowanie”: środek (położenie), obrót, a także skala. (Skala nie ma wpływu na sam obiekt *Empty*, ale może mieć na inne, zwykłe elementy „siatkowe”, dla których jest „rodzicem”).

Aby dodać do rysunku nowy obiekt *Empty*, umieść kursor 3D w miejscu, gdzie ma być jego środek. Następnie naciśnij **Space** (lub rozwiń menu *Add* — jest zaraz obok menu *File*). Wywołaj polecenie *Add Object Menu* → *Add Empty* (Rysunek 7.1.1):



Rysunek 7.1.1 Stworzenie pustego obiektu

Spowoduje to stworzenie nowego obiektu. Aby w ogóle był widoczny, każdy obiekt typu *Empty* ma włączoną opcję rysowania osi lokalnego układu współrzędnych.

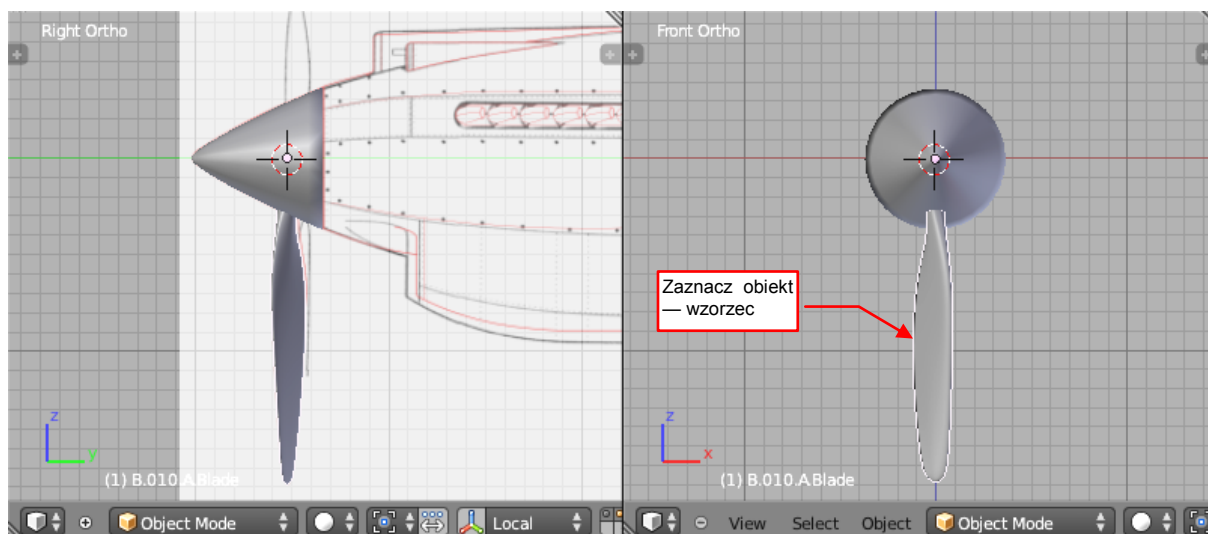


## 7.2 Powielanie obiektu (*Duplicate*)

W Blenderze istnieją dwa polecenia do tworzenia duplikatów obiektu:

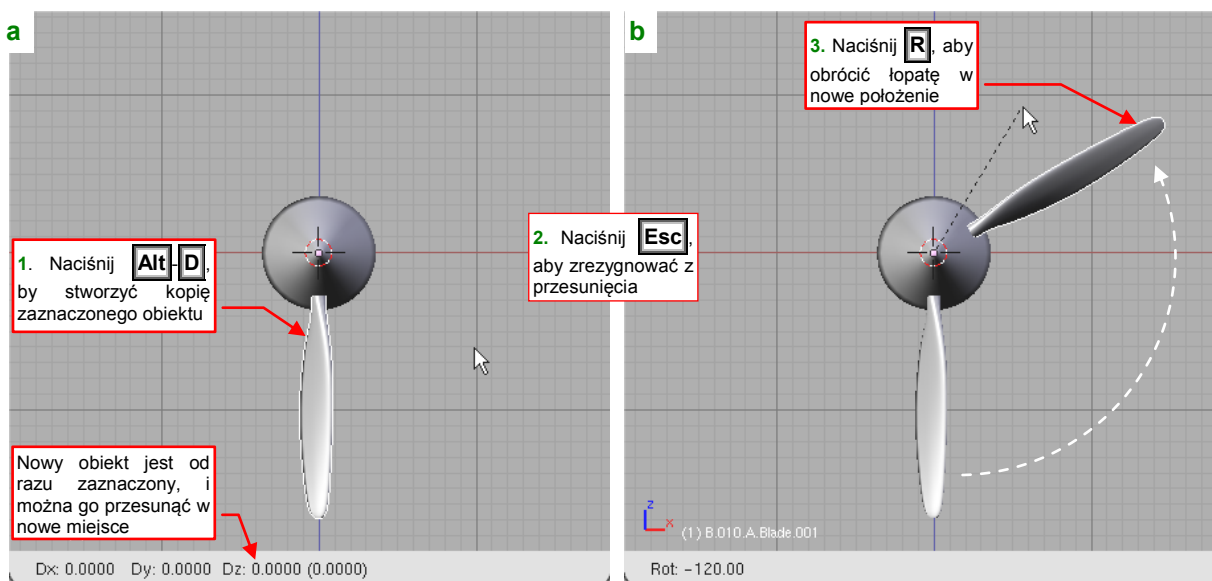
1. Duplikat jest stworzony jako kopia "powiązana" (**Object→Duplicate Linked**, **Alt-D**). Kopia może mieć oddzielną skalę, położenie, obrót, ale korzysta nadal z tej samej siatki, co wzorec. (W efekcie ma zawsze ten sam kształt, co oryginał);
2. Duplikat jest stworzony jako zwykła kopię (**Object→Duplicate**, **Shift-D**). Kopia jest obiektem, który w żaden sposób nie zależy od oryginału.

Wywołanie i obsługa obydwu poleceń jest identyczna, więc opiszę je na przykładzie tworzenia kopii "powiązanej". Rysunek 7.2.1 pokazuje sytuację początkową. Zaznaczyliśmy obiekt — wzorec: łopatkę śmigła. Utworzymy jej dwie powiązane kopie, tworząc trójłopatowe śmigło.



Rysunek 7.2.1 Łopata śmigła, do której dodamy dwie dalsze — jako powiązane kopie

Naciśnij teraz **Alt-D** (**Object→Duplicate Linked**). To stworzyło nowy obiekt. Blender nadaje mu unikalną nazwę poprzez dołączenie do nazwy oryginału końcówki w rodzaju ".001". Od razu także proponuje dokonać jego przesunięcia (Rysunek 7.2.2a):

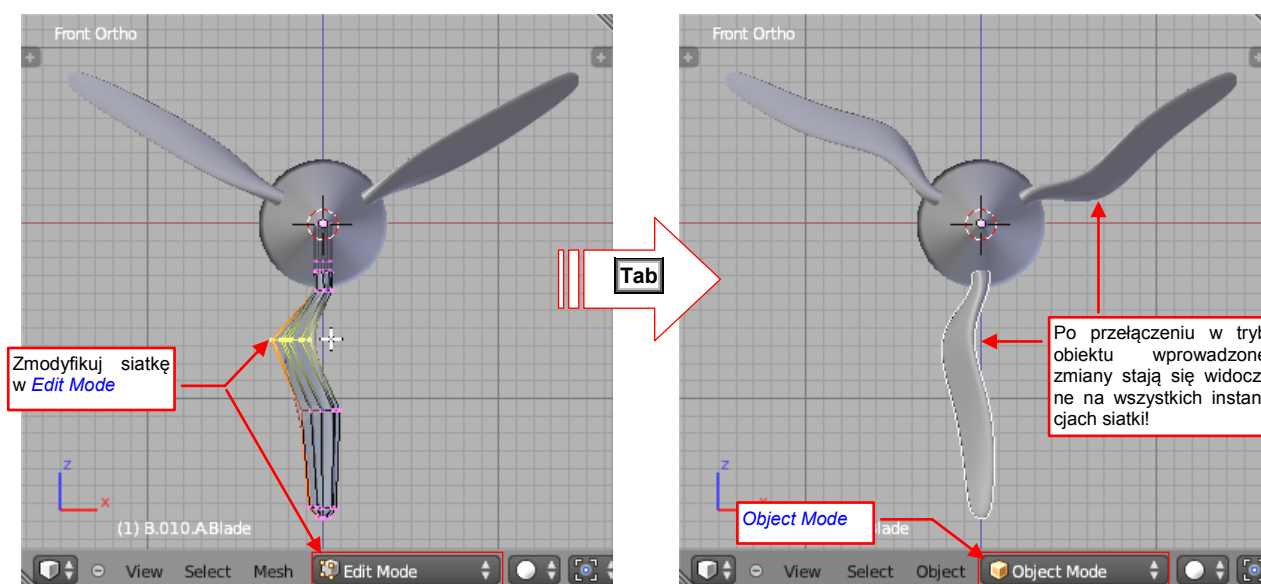


Rysunek 7.2.2 Powielenie (poprzez obrót)

Jeżeli nie chcesz przesunąć nowego obiektu, a np. obrócić, wystarczy nacisnąć **Esc**. To przerwie przesuwanie kopii, ale pozostawi ją jako zaznaczoną. Teraz wystarczy wybrać polecenie dowolnej innej transformacji, którą chcesz zastosować — na przykład **R**, by nowy obiekt obrócić<sup>1</sup> (Rysunek 7.2.2b).

- Zwykłą kopię obiektu tworzysz identycznie, tyle tylko że zaczynasz poleceniem **Shift-D** (*Object → Duplicate*)

Jaka jest w takim razie praktyczna różnica pomiędzy kopią zwykłą, a kopią powiązaną? Pokażę to na przykładzie śmigła (w którym dodałem już także trzecią łopatę). Wszystkie trzy łopaty wykorzystują tę samą siatkę. Gdy zmodyfikujesz jedną z nich, i powrócisz do trybu obiektu (*Object Mode*) — zmiana stanie się widoczna we wszystkich trzech instancjach (Rysunek 7.2.3):



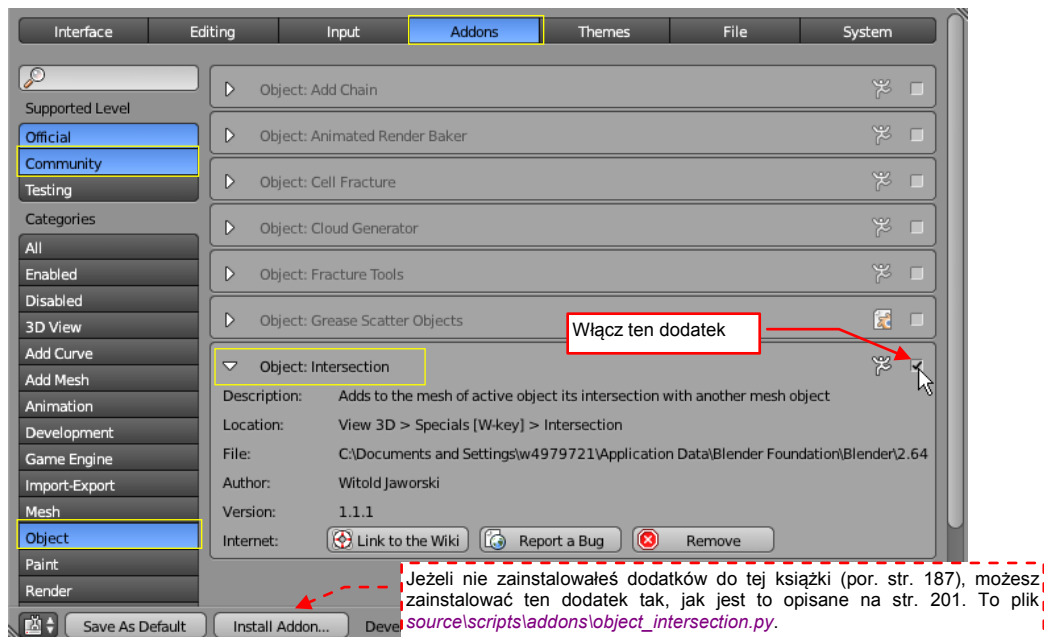
Rysunek 7.2.3 Obiekty powiązane — siatka jest zmieniana we wszystkich równocześnie

Więcej na ten temat — patrz "Struktura danych modelu i sceny w Blenderze", str. 462.

<sup>1</sup> Aby przełączyć się na obrót lub skalowanie można nie naciskać **Esc**, tylko od razu **R** lub **S**. Chciałem tu jednak pokazać ogólną zasadę, aby lepiej uświadomić Czytelnikowi jak to działa.

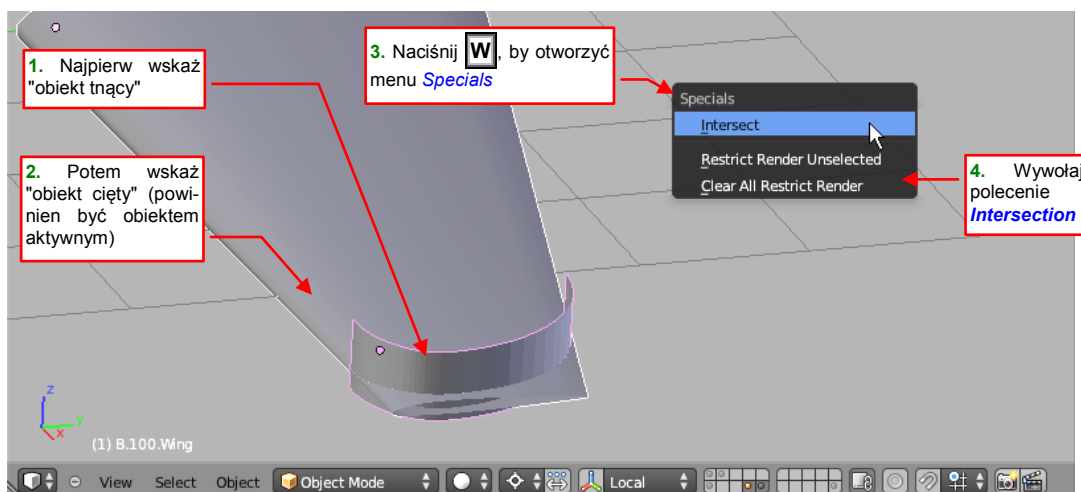
### 7.3 Wyznaczenie krawędzi przecięcia dwóch powłok

Do wyznaczenia przecięcia dwóch powłok proponuję użyć dołączonego do tej książki dodatku o nazwie *Intersection*<sup>1</sup>. Powinieneś go znaleźć w oknie *User Preferences*, sekcji *Addons: Object* (Rysunek 7.3.1):



Rysunek 7.3.1 Aktywacja dodatku do wyznaczenia krawędzi przecięcia siatek dwóch obiektów

Zaznacz najpierw dwa obiekty, dla których chcesz znaleźć krawędź przecięcia (Rysunek 7.3.2) :

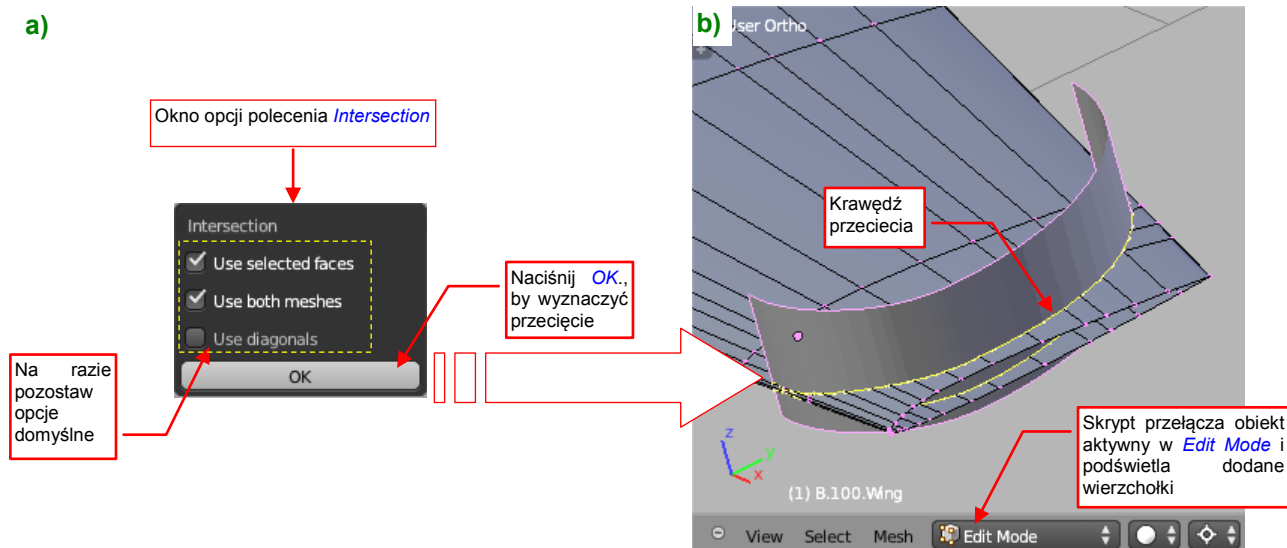


Rysunek 7.3.2 Zaznaczenie obiektów i wywołanie polecenia

Kolejność zaznaczania jest ważna: najpierw wskaż obiekt który pełni rolę „narzędzia tnącego”, a potem obiekt do którego siatki skrypt doda krawędź przecięcia. Następnie naciśnij klawisz **[W]**, by otworzyć menu *Specials*, i wybierz z niego polecenie *Intersection* (Rysunek 7.3.2).

<sup>1</sup> Blender ma zespół funkcji, pozwalających wykonać na siatkach operacje "Boole'a": różnicę, sumę, część wspólną. Mankamentem jednak wszelkich takich rozwiązań jest siatka, uzyskana jako rezultat. Zawiera wiele wierzchołków, których nie potrzeba do dalszej pracy, a które tylko niepotrzebnie zwiększają liczbę ścian i komplikują model. Osobiście wolę mieć pełną kontrolę nad siatką, którą formuję. Stąd przygotowałem skrypt, który wyznacza tylko krawędź przecięcia. Zazwyczaj, przy dobrze zaplanowanych siatkach, wkomponowanie nowej krawędzi w dotychczasową powierzchnię nie wymaga dużo pracy. Usuniesz parę ścian, stworzysz — korzystając z wierzchołków wyznaczonej krawędzi — kilkanaście nowych. W ten sposób uzyskujesz zaplanowany kształt za pomocą ładnej — bo prostej i regularnej — siatki. (Jest takie stare inżynierskie przysłowie: "co oko lubi, to i konstrukcja lubi")

Po uruchomieniu na ekranie pojawi się okno dialogowe z opcjami polecenia (Rysunek 7.3.3a):

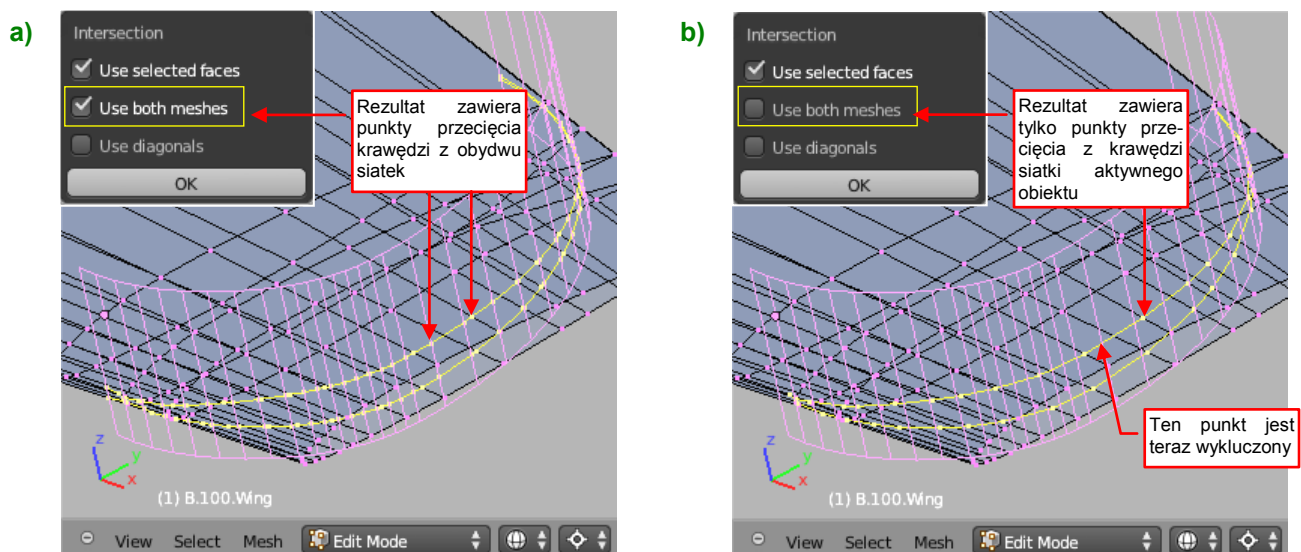


**Rysunek 7.3.3 Wyznaczenie krawędzi przecięcia**

Na razie niczego w nim nie zmieniaj, tylko naciśnij przycisk OK. W zależności od liczby ścian przecinanych siatek operacja może potrwać sekundę lub dwie. **Intersection** dodaje wyznaczoną krawędź przecięcia do siatki aktywnego obiektu, zaznacza jej wierzchołki, i przełącza Blender w tryb **Edit Mode**, abyś mógł je od razu zobaczyć (Rysunek 7.3.3b). Jeżeli nie jesteś zadowolony z rezultatów — wystarczy nacisnąć **X**, aby je usunąć, po czym powrócić (**Tab**) do **Object Mode** i spróbować wywołać polecenie jeszcze raz, z innymi ustawieniami.

Pierwsza z opcji operacji — **Use selected faces** — ma znaczenie tylko wtedy, gdy wcześniej w **Edit Mode** zaznaczyłeś jakieś ściany przecinanych siatek. Dodatek **Intersection** ograniczy wówczas zakres operacji tylko do tego zaznaczonego obszaru. Jeżeli nic nie jest na siatkach zaznaczone — sprawdzane są ich wszystkie ściany.

Gdy druga opcja — **Use both meshes** — jest włączona, skrypt buduje rezultat z punktów przecięcia krawędzi obydwu siatek (Rysunek 7.3.4a):



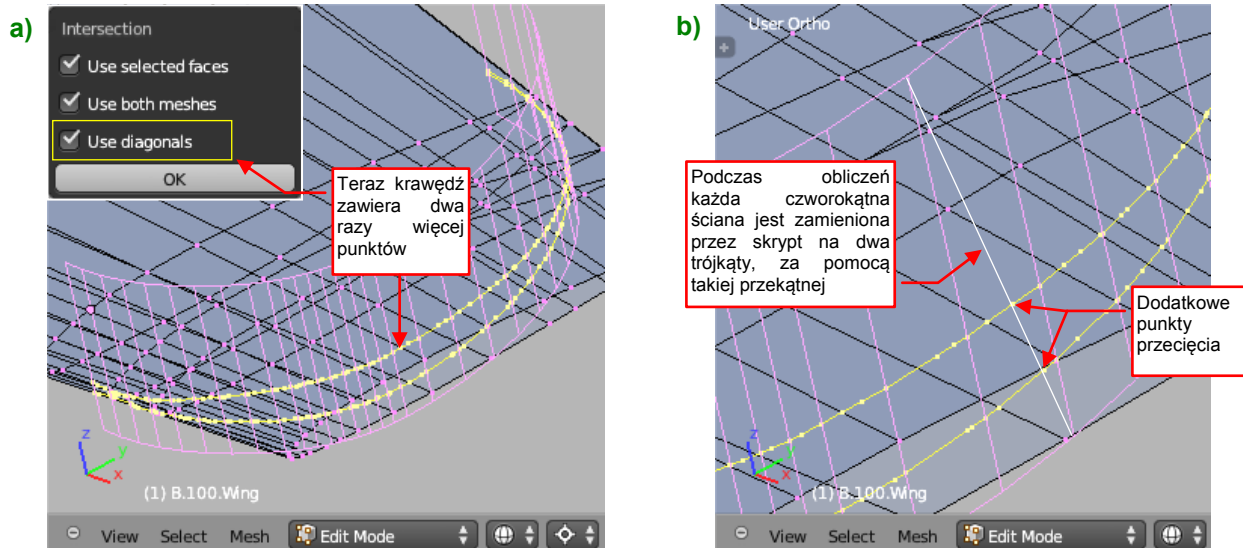
**Rysunek 7.3.4 Wpływ opcji **Use both meshes** na krawędź przecięcia**

Gdy ją wyłączysz, rezultat będzie zawierać wyłącznie punkty przecięcia aktywnego obiektu ze ścianami obiektu „tnącego” (Rysunek 7.3.4b). Taki efekt może być przydatny przy odpowiednio gęstych siatkach.



Skrypt *Intersection* wyszukuje punkty przecięcia krawędzi jednej z siatek ze ścianami drugiej i łączy je we wspólną linię. Podczas obliczeń każda czworokątna ściana jest dzielona przekątną (*diagonal*), aby zamienić ją na dwie ściany trójkątne. Zazwyczaj punkty przecięcia tych tymczasowych przekątnych ze ścianami drugiej siatki są pomijane.

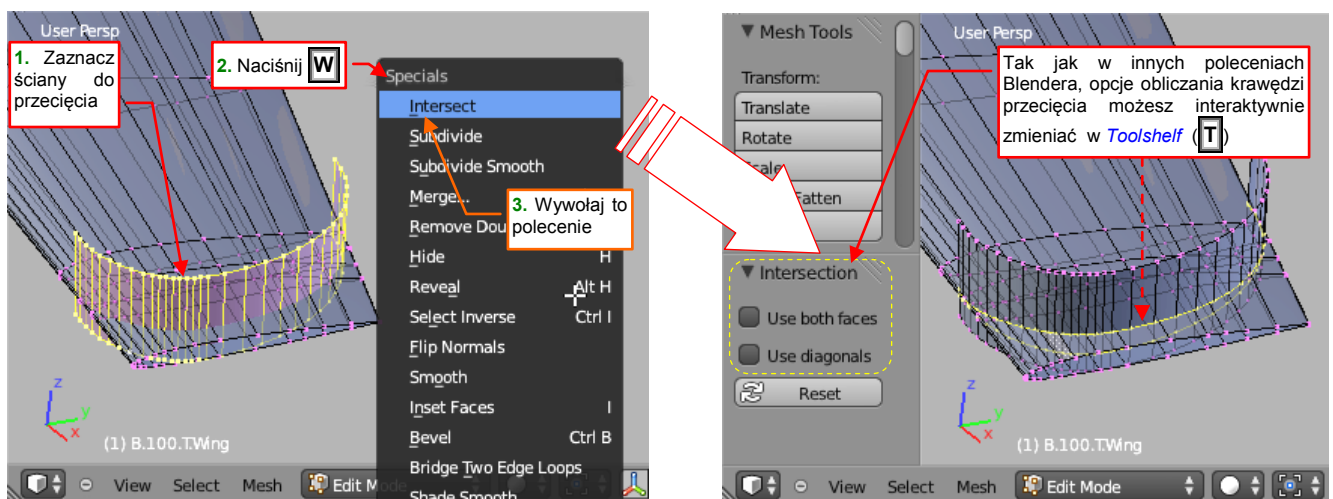
Jeżeli jednak włączysz opcję *Use diagonals*, zobaczysz je wszystkie (Rysunek 7.3.5a):



Rysunek 7.3.5 Wpływ opcji *Use diagonals* na krawędź przecięcia

Rysunek 7.3.5b) przedstawia przekątną wybranej ściany i jej punkty przecięcia ze ścianami drugiej siatki. Opcję *Use diagonals* pozostawiłem tylko na wszelki wypadek — zazwyczaj jej nie używam.

Ta sama operacja dostępna jest także w *Edit Mode*, pod nazwą *Intersect*. W tym trybie służy do przecinania zaznaczonych ścian z pozostałymi ścianami siatki (Rysunek 7.3.6):



Rysunek 7.3.6 Wywołanie polecenia *Intersect* w trybie edycji siatki

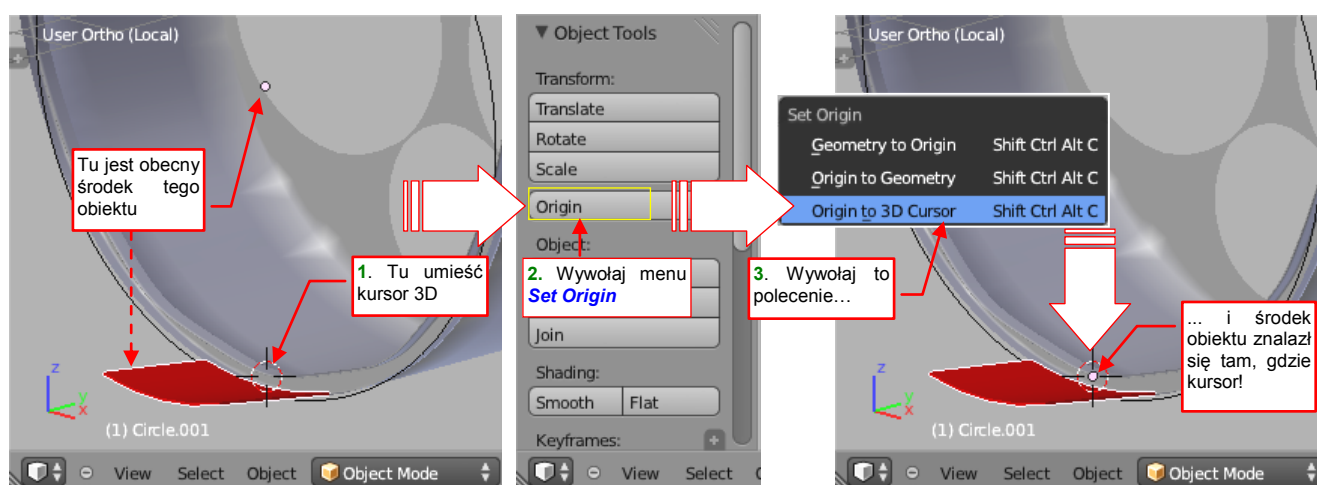
W tej wersji polecenia jego opcje są dostępne w panelu przybornika (*Toolshelf*, *T*). Możesz je tu przestawiać, interaktywnie zmieniając rezultat. Jeżeli chcesz wykluczyć jakieś ściany z przecinania — ukryj je (*H*) przed tą operacją, a po jej zakończeniu „odkryj” powtórnie (*Alt-H*).

## 7.4 Zmiana położenia środka obiektu (*Set Origin*)

Punkt odniesienia obiektu, używany dla określenia położenia, obrotu, i skali, nazywany jest w Blenderze **środkiem** obiektu. To polskie tłumaczenie angielskiego terminu *origin* wywołuje czasem niewłaściwe skojarzenia, bo środek obiektu często wcale nie leży w jego środku geometrycznym.

Stosunkowo często pojawia się konieczność zmiany położenia środka obiektu względem jego siatki. Najszybciej możesz to zmienić w następujący sposób (Rysunek 7.4.1):

- zaznacz obiekt;
- umieść kursor 3D w miejscu, w którym chcesz umieścić nowy środek obiektu;
- Wywołaj polecenie *Origin to 3D Cursor*. Możesz to zrobić z menu *Set Origin*, wywoływanego przyciskiem *Origin* z panelu *Object Tools* przybornika *Toolshelf* (T) — jak pokazuje to Rysunek 7.4.1), albo z menu *Object→Transform*;



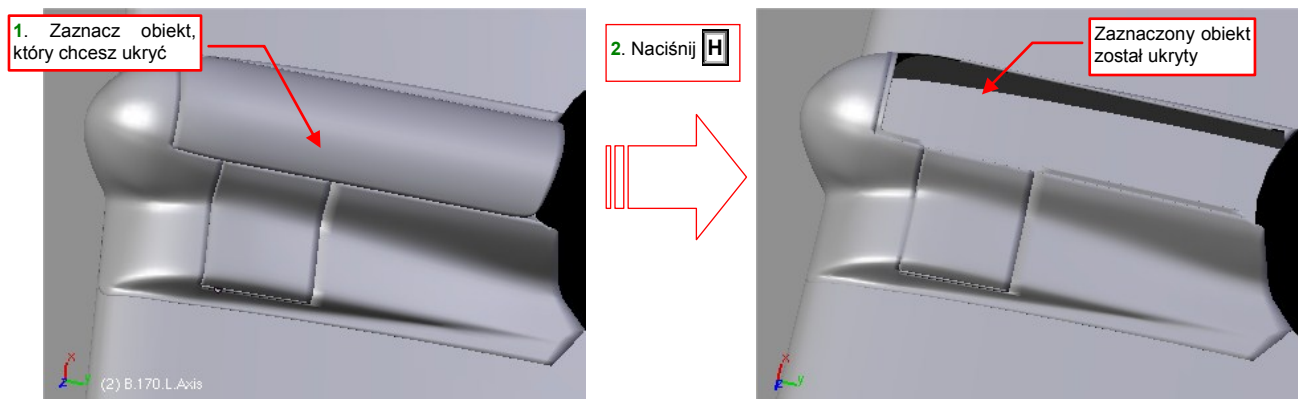
Rysunek 7.4.1 Zmiana położenia środka obiektu

W efekcie środek obiektu znajdzie się w miejscu wskazanym przez kursor 3D. Zmiana położenia środka obiektu jest "jego wewnętrzną sprawą". Nie powinna wpłynąć na jego rozmiar ani położenie w przestrzeni (chyba, że obiekt jest obrócony w jakiś specyficzny sposób!).

### 7.5 Chwilowe ukrycie obiektu (*Hide Selected*)

Czasami, podczas pracy, przydatna jest możliwość ukrycia na jakiś czas niektórych obiektów. Jest to szczególnie ważne wtedy, gdy jedna część zasłania drugą.

Aby ukryć zaznaczony obiekt (lub obiekty), po prostu naciśnij **H** (*Object → Show/Hide → Hide Selected*). To sprawi, że zaznaczenie zniknie (Rysunek 7.5.1):

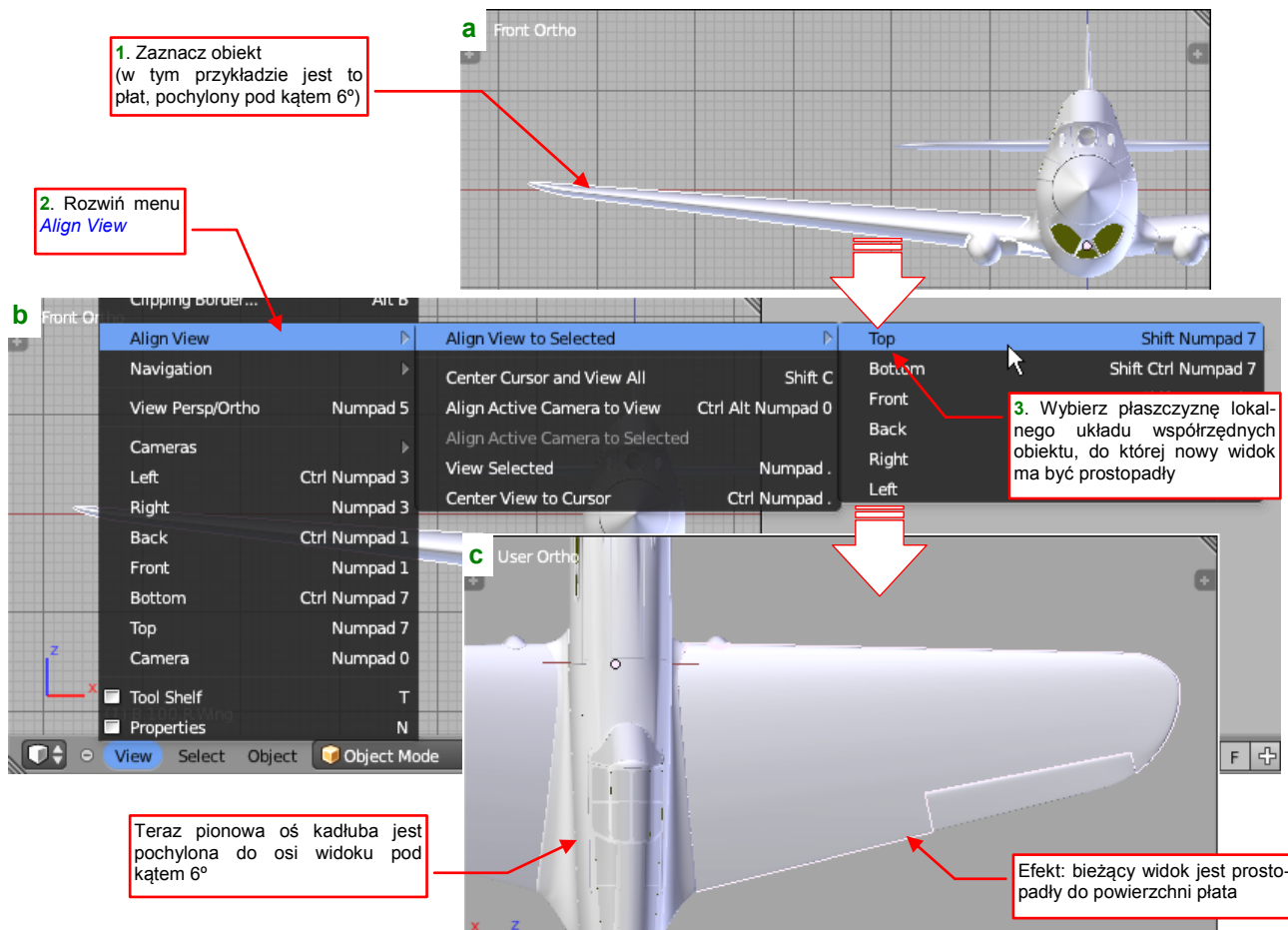


Rysunek 7.5.1 Ukrycie wybranego obiektu

Aby ukryte obiekty stały się z powrotem widoczne — naciśnij **Alt-H** (*Object → Show/Hide → Show Hidden*).

## 7.6 Wyrównanie widoku do orientacji obiektu (*Align View to Selected*)

Podczas pracy nad modelem potrzeba czasami ustawić płaszczyznę widoku (czyli płaszczyznę, po której możemy przesuwać obiekty) prostopadłą do wybranego obiektu. Aby to zrobić, zaznacz obiekt, do którego chcesz się "wyrównać" (Rysunek 7.6.1a). Następnie rozwiń menu **View→Align View→Align View to Selected** (Rysunek 7.6.1b):



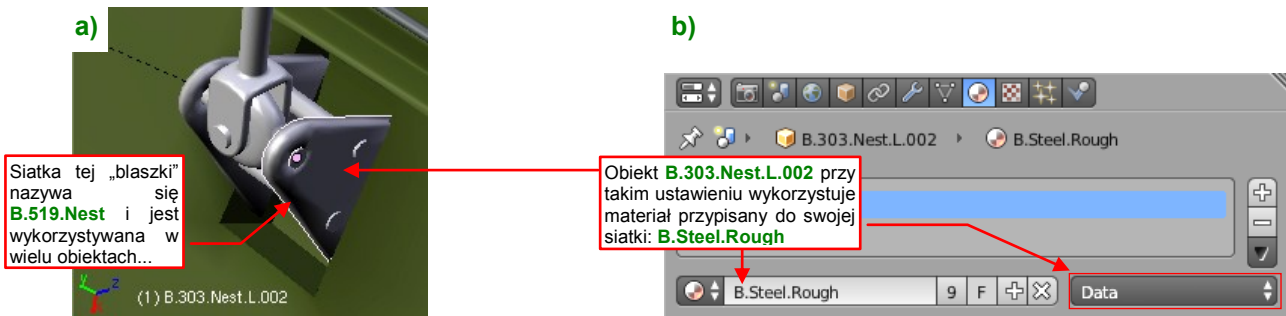
Rysunek 7.6.1 Wyrównanie aktualnego widoku do wybranego obiektu

Z menu **Align View to Selected** możesz wybrać jedną z sześciu "kardynalnych" płaszczyzn obiektu (Rysunek 7.6.1b). Efekt zobaczysz natychmiast (Rysunek 7.6.1c). W razie czego, jeżeli za pierwszym razem wybrałeś złą płaszczyznę kardynalną - wywołaj to polecenie jeszcze raz, i wybierz właściwą. Zwróć uwagę, że poszczególnym poleceniom z menu **Align View to Selected** przypisano skróty klawiaturowe. Są łatwe do zapamiętania: wystarczy poprzedzić skrót, którym zmieniasz projekcję względem globalnego układu współrzędnych klawiszem **Shift**. I tak, jeżeli normalnie wybierasz widok z góry klawiszem **7**, to na widok z góry w lokalnej płaszczyźnie aktywnego obiektu przełączy Cię **Shift-7**. Podobnie na lokalny rzut z boku przełączysz się skrótem **Shift-3**, a z powrotem na „globalny” rzut z boku — klawiszem **3**.



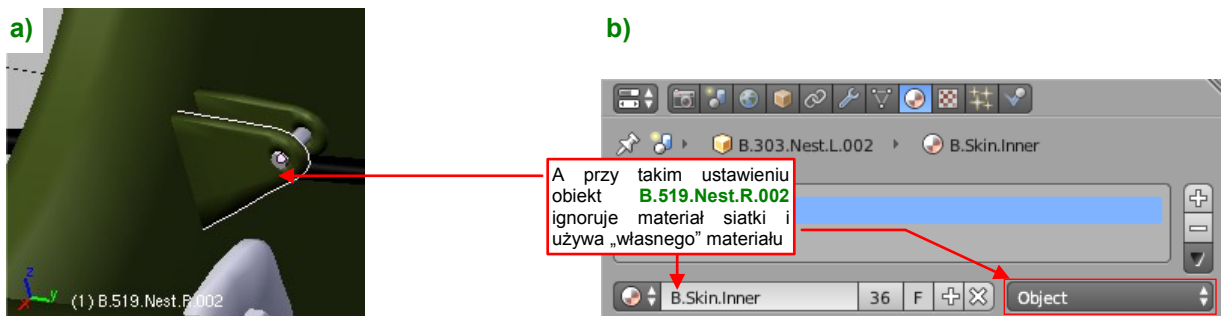
## 7.7 Przypisanie materiału do obiektu

Domyślnie Blender przypisuje materiał do siatki (por. str. 181, Rysunek 6.1.3, i str. 273). Czasami jednak potrzebujemy czegoś odwrotnego. Dotyczy to, na przykład, różnych drobnych części o identycznym kształcie — śrub, bolców, jakichś standardowych okuć. Zobacz ten trójkątny element okucia z otworem na oś, który pokazują: Rysunek 7.7.1a) i Rysunek 7.7.2a). Każde okucie składa się z dwóch takich blaszek. Wszystkie blaszki używają tej samej siatki, o nazwie **B.519.Nest**. (Są to jej „klony”). Siatce jest przypisany materiał wyglądający jak „gładka stal” — **B.Steel.Smooth**. To barwa, w której te płaskowniki występowały najczęściej. Dopóki obiekt ma w kontekście **Material** włączoną opcję **Data** (Rysunek 7.7.1b), dopóty „dziedziczy” materiał swojej siatki:



Rysunek 7.7.1 Materiał przypisany do siatki

W innym miejscu — prowadnicy cięgna, przymocowanej do wręgi kadłuba (Rysunek 7.7.2a) — te same blaszki powinny być „pomalowane” na kolor reszty szkieletu: **B.Skin.Inner**. Czy można ten efekt osiągnąć bez powielania siatki? Tak, wystarczy przełączyć tryb kontekstu **Material** z **Data** na **Object**, i wybrać odpowiedni materiał (Rysunek 7.7.2b):

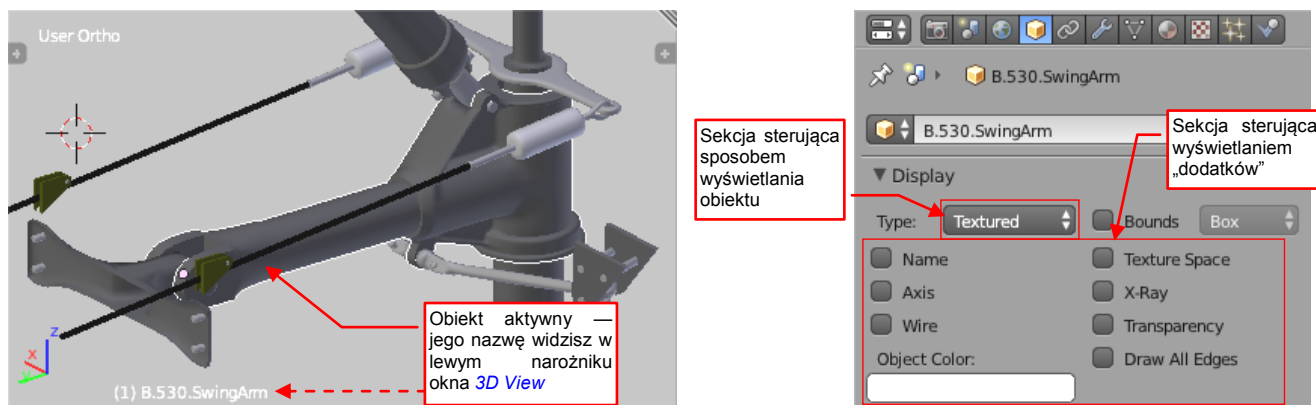


Rysunek 7.7.2 Materiał przypisany do obiektu

Co właściwie ten przełącznik **Object** robi? Informuje Blender, że dla tej konkretnej części należy użyć materiału przypisanego do obiektu, a nie do siatki (**Data**).

## 7.8 Opcje wyświetlania obiektu

W panelu **Display** zestawu **Object** znajdziesz opcje wyświetlania obiektu. Pamiętaj, że dotyczą zawsze obiektu aktywnego (Rysunek 7.8.1):



Rysunek 7.8.1 Panel **Display** — opcje rysowania obiektu w oknie widoku.

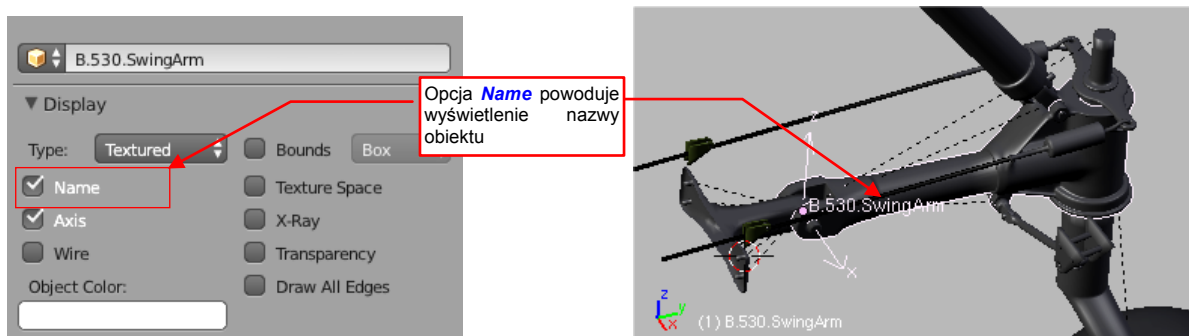
Pola **Type** zazwyczaj nic nie musisz zmieniać — pozostaw sposób wyświetlania na **Textured**. No, może czasami, można przełączyć wyświetlanie powierzchni szyb kabiny na **Wire**. (Dzięki temu także podczas modelowania będziesz „przez nie widzieć” wnętrze kabiny).

Z kolei najczęściej wyświetlanym przeze mnie elementem dodatkowym są osie lokalnego układu współrzędnych. Ich widocznością steruje opcja **Axis** (Rysunek 7.8.2):



Rysunek 7.8.2 Włączenie rysowania osi lokalnego układu współrzędnych (**Axis**).

Czasami, podczas pracy z ograniczeniami, przydaje się włączyć także wyświetlanie nazw obiektów. Służy do tego opcja **Name** (Rysunek 7.8.3):

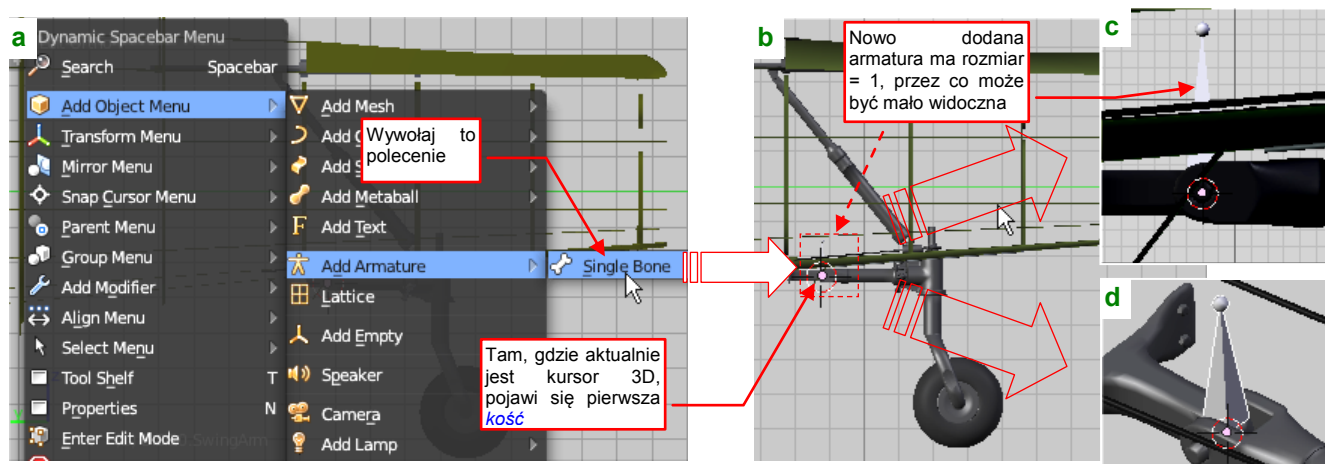


Rysunek 7.8.3 Włączenie rysowania nazwy obiektu (**Name**).

## 7.9 Dodanie armatury (*Armature*)

Aby „puścić w ruch” wykonany model, w grafice komputerowej używa się specjalnych szkieletów, nazywanych także armaturami (*Armature*). Skoro szkielet, to muszą być i kości. I tak się już utarło, że szkielety używane do animacji są złożone z „kości” (*Bones*). W tej sekcji pokażę ich zastosowanie na przykładzie animacji mechanizmu chowania kółka ogonowego (zasada działania — zob. str. 38, Rysunek 2.2.1)

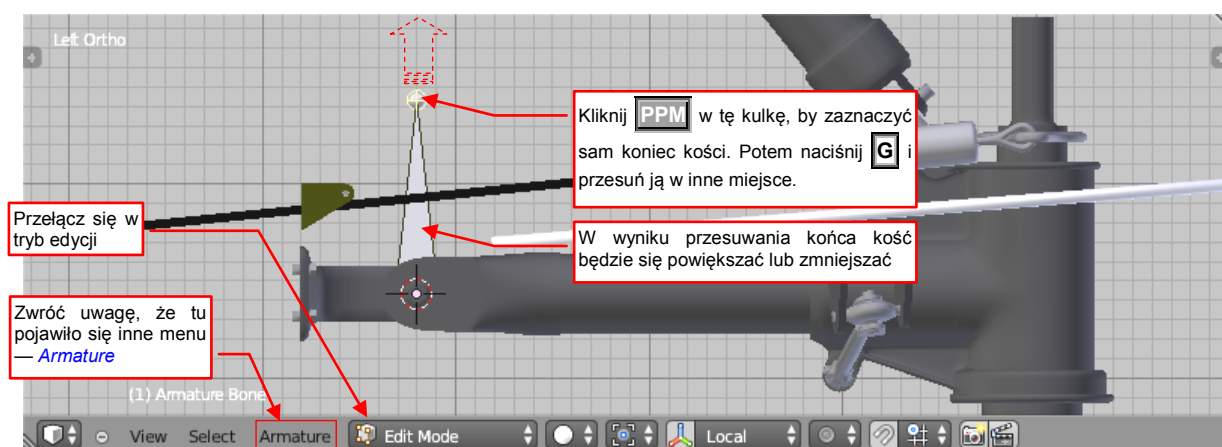
W Blenderze armaturę można dodać jak każdy inny obiekt: najpierw ustaw kursor 3D w miejscu, gdzie ma się znaleźć początek pierwszej kości. Potem wywołaj z menu **Add Object Menu** polecenie **Add Armature → Single Bone** (Rysunek 7.9.1a):



Rysunek 7.9.1 Dodanie armatury (pierwszej „kości”)

W pierwszym momencie po stworzeniu armatury, jeżeli pracujesz w mniejszym powiększeniu, możesz jej nie dostrzec (Rysunek 7.9.1b). To dlatego, że początkowo to jest jedna kość (skierowana do góry), o rozmiarze 1 jednostki (Rysunek 7.9.1c,d). Zdecydowanie odradzam zmianę skali armatury jako obiektu<sup>1</sup>.

Zamiast tego, przełącz się w tryb edycji, i zacznij dopasowywać położenie i rozmiar kości do obiektu (Rysunek 7.9.2):

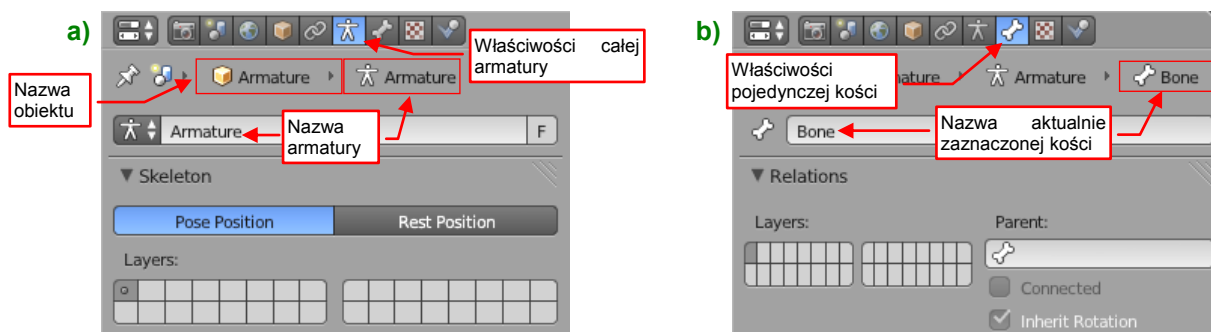


Rysunek 7.9.2 Powiększenie kości do odpowiednich rozmiarów

W trybie edycji armatury kość możesz zaznaczyć na dwa sposoby. Gdy klikniesz **PPM** w jej trzon — podświetli się cała, i możesz ją całą przesunąć w inne miejsce. Gdy klikniesz **PPM** w jej końcówkę — podświetli się tylko umieszczona na niej kula. W tym trybie przesuwasz sam koniec kości (początek pozostaje tam, gdzie jest).

<sup>1</sup> Gdy zmienisz skalę całej armatury, a potem przypiszesz do jej kości jakieś obiekty, to w trybie edycji zobaczysz je wszystkie odpowiednio pomniejszone. To może być trochę deprymujące (w *Object Mode* będą wszystkie wyglądały normalnie);

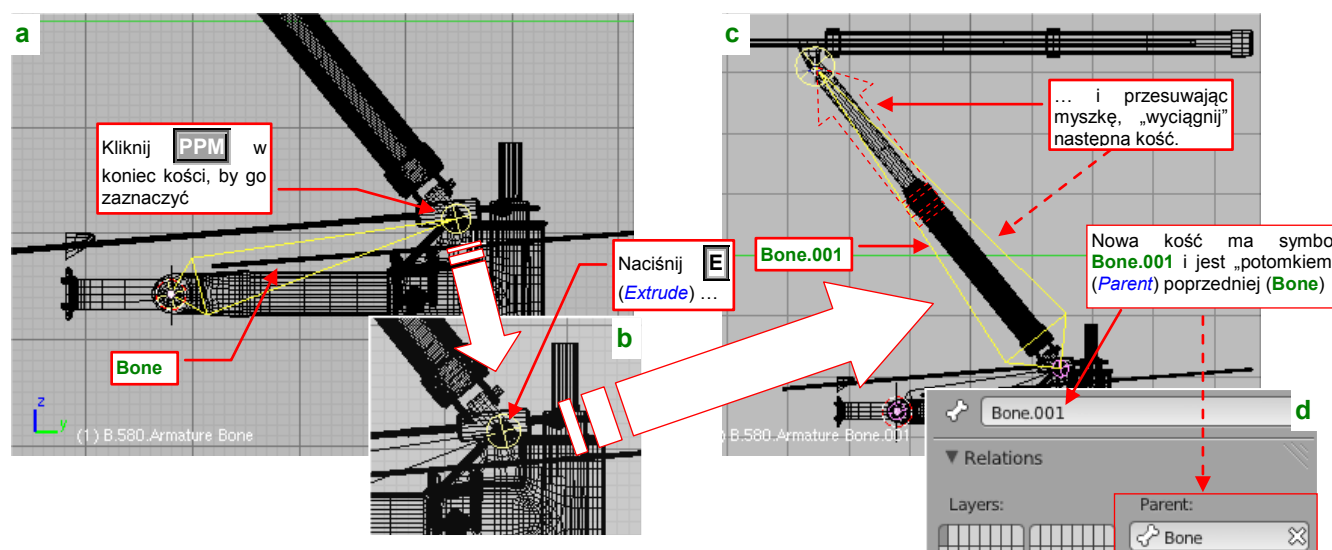
Zwróć uwagę że w oknie *Properties* zestaw *Object Data* ma teraz inną ikonę (Rysunek 7.9.3a). Teraz znajdziesz w nim ogólne ustawienia aktualnej armatury. Pojawił się także drugi zestaw — *Bone* — z ustawieniami aktualnie wybranej kości (Rysunek 7.9.3b):



Rysunek 7.9.3 Zawartość (*Object Data*) obiektu armatury i jego kości (*Bone*)

Domyślnie pierwsza kość nosi nazwę *Bone*, a następne — *Bone.001*, *Bone.002*, itd. Oczywiście, można je nazwać inaczej, adekwatnie do konkretnej roli, którą pełnią (np. *ramię*, *przedramię* czy coś w tym stylu). Nazwy kości muszą być unikalne tylko w obrębie pojedynczej armatury. (W innej armaturze możesz użyć takich samych nazw). W zestawach *Object Data* (Rysunek 7.9.3) możesz nadać nową nazwę całej armaturze, a w zestawie *Object* — obiektowi, do którego jest przypisana. (Najlepiej żeby były takie same).

Aby dodać do armatury nową kość, która łączy się z poprzednią, należy użyć lokalnej odmiany operacji wytłaczania (*Extrude*). Kliknij najpierw *PPM* w koniec kości — punkt, w którym ma się zaczynać następna kość (Rysunek 7.9.4a):



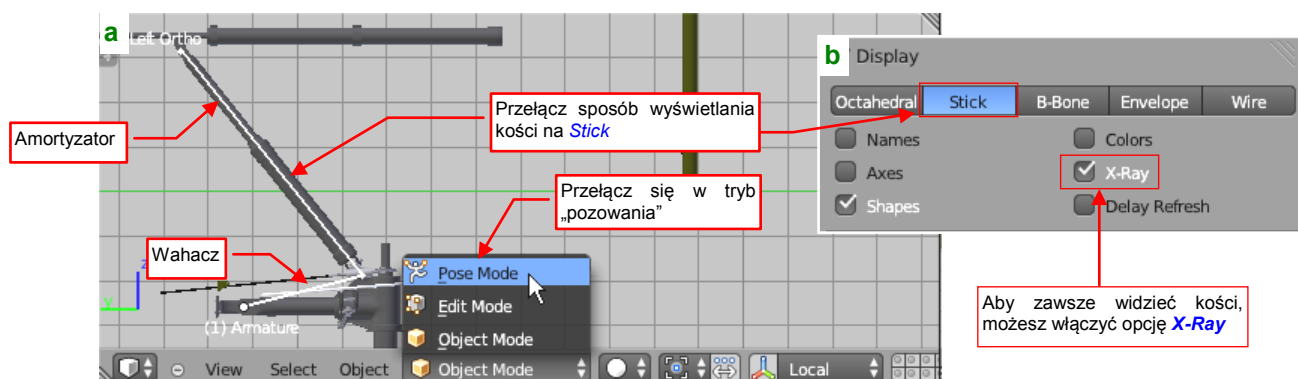
Rysunek 7.9.4 Wytyłczenie (*Extrude*) kolejnej kości

W odpowiedzi Blender powinien podświetlić samą końcówkę kości (Rysunek 7.9.4b). (Jeżeli zaznaczył całą — kliknij w *PPM* w to samo miejsce jeszcze raz. Cała kość i końcówka są wybierane naprzemiennie). Następnie naciśnij przycisk *E* (lub wybierz z menu *Armature* → *Extrude*). W tym momencie do armatury zostanie dodana nowa kość, której koniec możesz przesunąć myszką w odpowiednie miejsce (Rysunek 7.9.4c). Gdy klikniesz *LPM* — kość zostanie ostatecznie dodana. Gdybyś chciał z niej jednak zrezygnować — w trakcie przesuwania wystarczy nacisnąć *Esc*.

Nowa kość otrzymuje automatycznie taką samą nazwę, jak kość z której powstała, z dodaną numeryczną końcówką (np. z *Bone* wytłoczysz *Bone.001*, a z *Bone.001* — *Bone.002*). Blender przypisuje ją także do poprzedniej kości jako „rodzica” (zobacz pole *Parent* na panelu *Relations* zestawu *Bone* — Rysunek 7.9.4d).



„Rozpięliśmy” już dwie kości, odpowiadające wahaczowi i amortyzatorowi kółka ogonowego. Inna sprawa, że są rysowane przez Blender jako dość duże ośmioboki (*Display:Octahedral*). Na szczęście w panelu *Armature:Display* masz do wyboru kilka innych sposobów reprezentacji kości. Przełącz się na przykład na „pręty” (*Stick* — Rysunek 7.9.5):

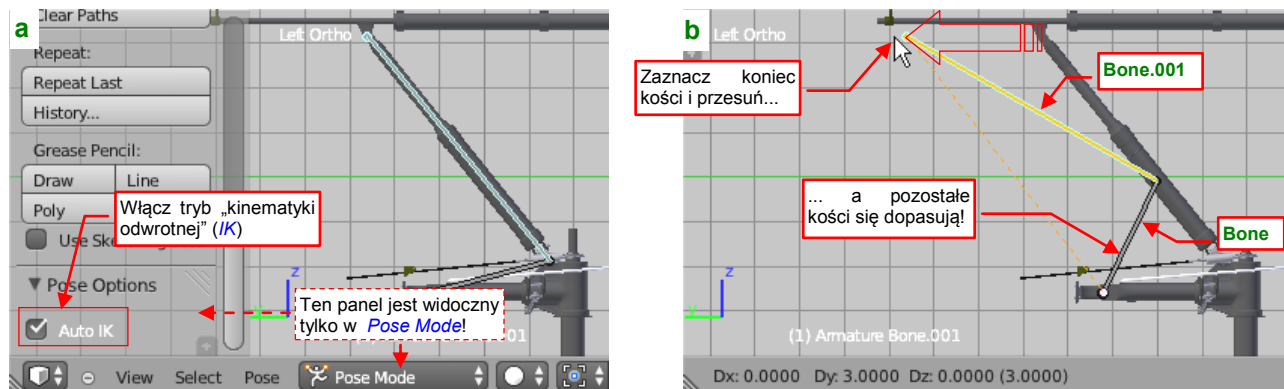


Rysunek 7.9.5 Włączenie *Pose Mode* i dostosowanie wyświetlania kości

Zobaczmy, jak to wygląda w działaniu. Przełącz się w specjalny tryb „pozowania” — *Pose Mode*. Wystarczy wybrać *Pose Mode* z listy trybów pracy w nagłówku okna widoku 3D

- Zawartość listy trybów pracy zależy od rodzaju aktywnego obiektu. Tryb *Pose Mode* pojawia się na niej tylko wtedy, gdy aktywnym obiektem jest jakaś armatura.

W *Pose Mode* włącz w przyborniku opcję *Auto IK*<sup>1</sup> (Rysunek 7.9.6a). To spowoduje, że cały układ kości zacznie się zachowywać jak prawdziwy zespół powiązanych przegubami prętów (Rysunek 7.9.6b):



Rysunek 7.9.6 Poruszanie armaturą w trybie *Auto IK*

Spróbujmy teraz poruszać naszą armaturą. Kliknij **PPM** w kość *Bone.001*, aby ją zaznaczyć (powinna się podświetlić na niebiesko). Następnie naciśnij **G** (*Pose*→*Transform*→*Grab*) i przesunij koniec zaznaczonej kości w nowe miejsce (Rysunek 7.9.6b). Czy widzisz? Obydwie kości poruszają się całkiem realistycznie, jak dwa połączone przegubowo segmenty.

W trybie pozowania można eksperymentować, ustawiając armaturę na najróżniejsze sposoby. Gdy przełączysz się z powrotem w tryb edycji, wszystko powróci do pozycji początkowej (*Rest position*).

<sup>1</sup> *IK* to skrót od angielskiego *inverse kinematics*. W języku polskim nosi nazwę „kinematyki odwrotnej” i pochodzi z automatyki/robotyki. Zazwyczaj chodziło o ramię robota, złożone z kilku ruchomych segmentów. „Zwykła” kinematyka pozwalała rozwiązać prostsze zagadnienie: „jeżeli pierwszy segment obrócimy o 30°, a drugi względem pierwszego o 20°, to gdzie się znajdzie koniec ramienia?”. Kinematyka odwrotna znajduje odpowiedź na pytanie postawione bardziej praktycznie: „o jaki kąt należy odwrócić każdy z segmentów, by koniec ramienia znalazł się tam, gdzie chcemy?”

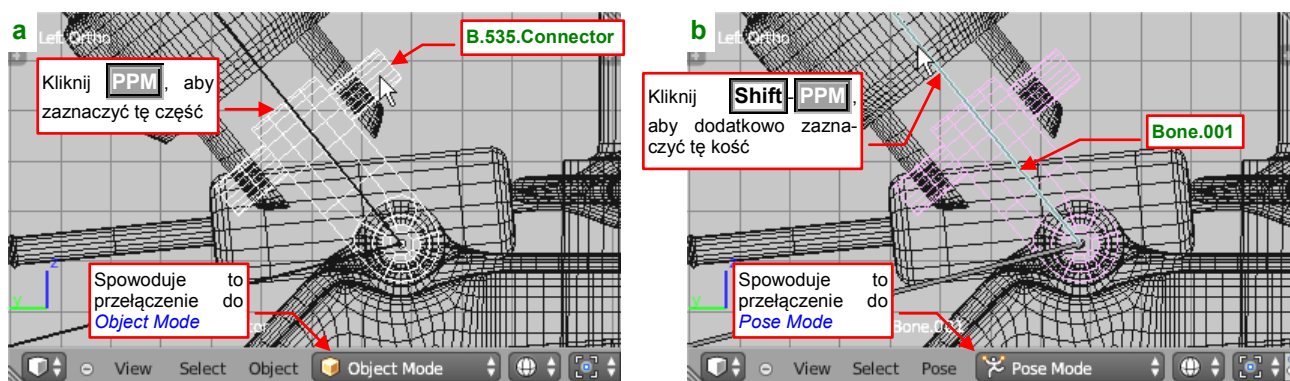
Teraz wystarczy przyłączyć do tych kości odpowiednie obiekty. Upewnij się, że jesteś w *Pose Mode*.

- Aby zaznaczyć obiekt i kość tak, jak to jest pokazane poniżej, musisz zacząć zaznaczanie w trybie *Pose Mode*. Inaczej nie uda Ci się wskazać konkretnej kości.

Jeżeli chodzi o amortyzator, to w hierarchii obiektów jego „rodzicem” powinien być niewielki **B.355.Connector**. Zaznacz go (**PPM**).

- Jeżeli w *Pose Mode*, mimo klikania **PPM**, Blender nie chce obiektu podświetlić, to oznacza to, że masz zaznaczoną jakąś kość. Wyłącz ją (klawisz **A** — *Select/Deselct All*) i spróbuj jeszcze raz.

Zwróć uwagę, że to zaznaczenie przełączyło tryb pracy w *Object Mode* (Rysunek 7.9.7a):

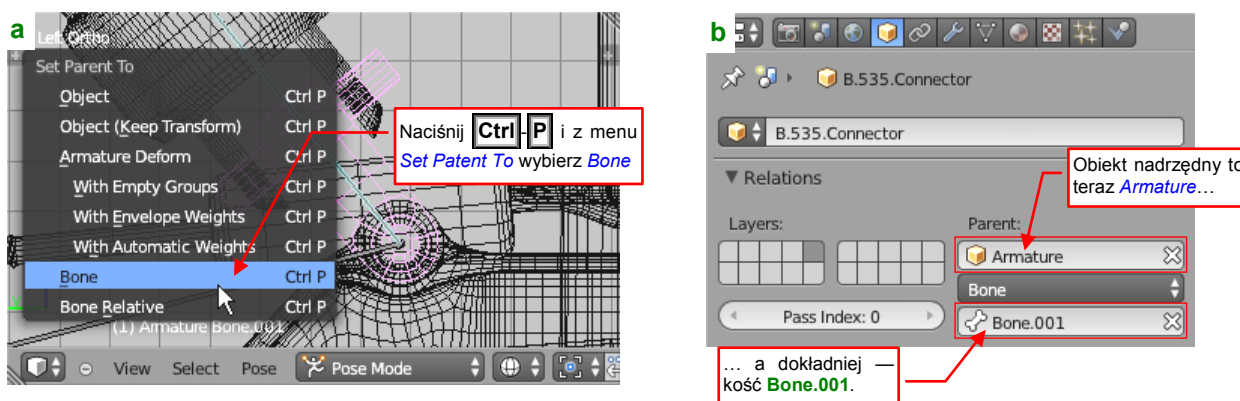


Rysunek 7.9.7 Zaznaczenie początku amortyzatora i kości (Bone.001)

Do złączki **B.535.Connector** powinny być hierarchicznie przypisane (relacją „parent — child”): cylinder (**B.540.Damper**) i tłok (**B.550.Piston**). W przypadku obiektów przyłączonych do kości, bardzo ważny jest środek. Najlepiej, aby się znajdował w tym samym miejscu, co początek kości. Zacznij od odłączenia obiektu **B.535.Connector** od obiektu nadrzędnego (**Alt-P**, *Clear Parent: Clear and Keep Transform*).

Teraz dodaj do wybranych elementów odpowiednią kość — w tym przypadku to **Bone.001**. Kliknij w nią **PPM**, trzymając wciśnięty klawisz **Shift**. Spowoduje to przełączenie Blendera w *Pose Mode*. Powinieneś w tym momencie widzieć: obiekt, który chcesz przypisać do kości w kolorze bladnoróżowym („pozostałe zaznaczenie”) i kość w kolorze niebieskim (Rysunek 7.9.7b).

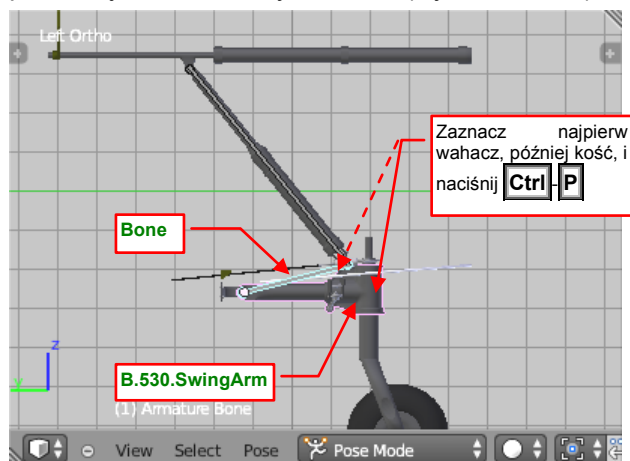
Gdy dla tak wskazanych obiektów naciśniesz **Ctrl-P**, pojawi się menu *Set Parent To*, z którego należy wybrać opcję **Bone** (Rysunek 7.9.8a):



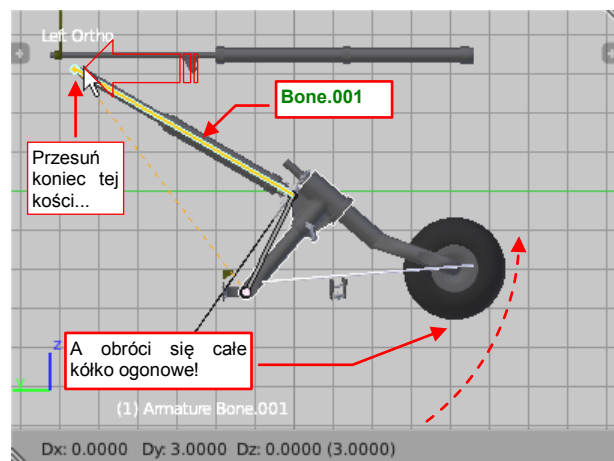
Rysunek 7.9.8 Przypisanie amortyzatora do kości (Bone.001)

To wszystko — złączka amortyzatora jest przypisana do kości (a wraz z nią — wszystkie jej obiekty „potomne”: cylinder i tłok, czyli cały amortyzator). To przypisanie widać właściwościach obiektu **B.535.Connector** (Rysunek 7.9.8b) — jego rodzicem jest kość **Bone.001** z armatury **Armature**.

W ten sam sposób przypisz do **Bone.001** tłok amortyzatora, a rodzicem wahacza (**B.530.SwingArm**) zrób pierwszą kość armatury — **Bone** (Rysunek 7.9.9):



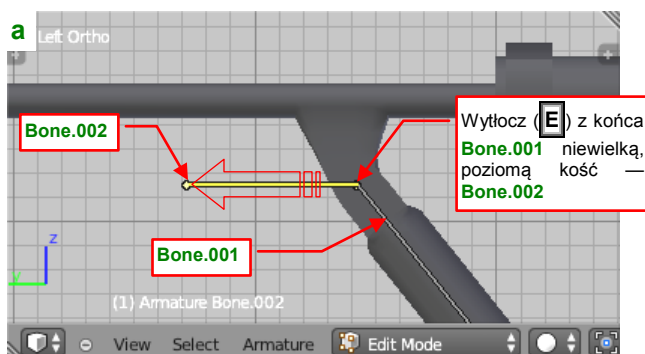
Rysunek 7.9.9 Przypisanie wahacza do kości **Bone**



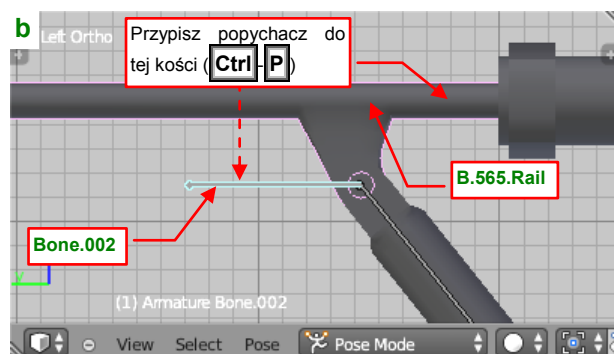
Rysunek 7.9.10 Wynik — składające się koło ogonowe

Zwróć uwagę, że środek wahacza znajduje się tam, gdzie początek kości **Bone** — w osi, do której wahacz jest przyczepiony. Teraz wystarczy w **Pose Mode** powtórnie przesunąć poziomo (**G**, **Y**) koniec kości amortyzatora (**Bone.001**) — i cały zespół koła ogonowego składa się i rozkłada jak pierwotny (Rysunek 7.9.10)!

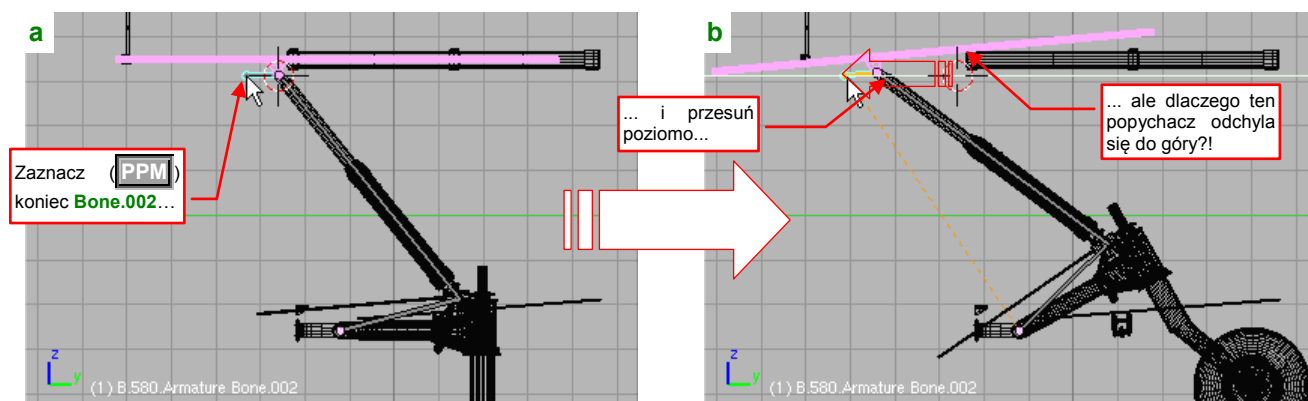
Skoro nam tak dobrze idzie, włączmy w ten mechanizm popychacz (**B.565.Rail**). W prawdziwym samolocie ten element przesuwał koniec amortyzatora do tyłu i do przodu, wymuszając otwieranie i zamykanie podwozia. Wyłóż z końca **Bone.001** kolejną, poziomą kość (**Bone.002**) (Rysunek 7.9.11a):



Rysunek 7.9.11 Wyłożenie trzeciej kości (**Bone.002**) i przypisanie jej do popychacza



Przypisz do kości **Bone.002** popychacz **B.565.Rail** (Rysunek 7.9.11b). Teraz wystarczy zaznaczyć tę ostatnią kość (Rysunek 7.9.12a) i przesunąć:

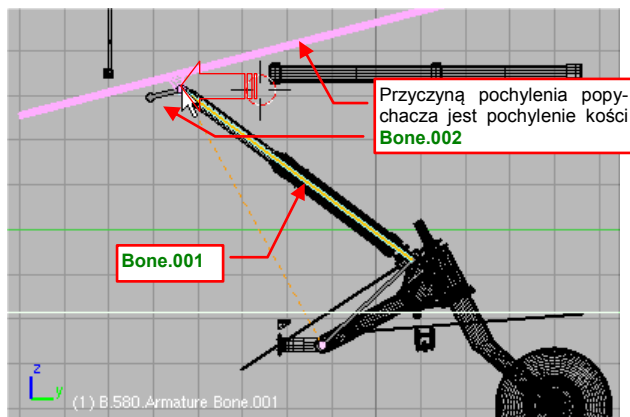


Rysunek 7.9.12 Niespodziewany rezultat poziomego przesunięcia kości **Bone.002**

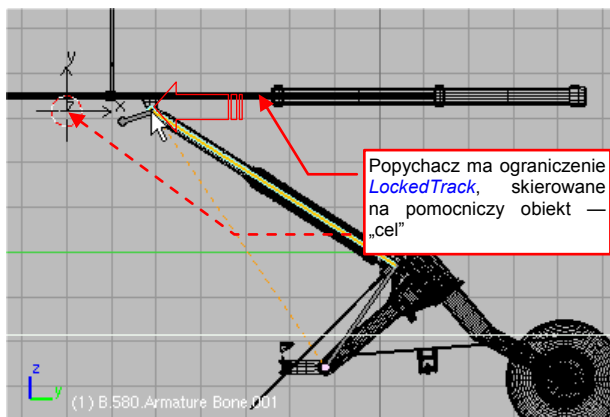
Ale zaraz, dlaczego w miarę, jak przesuwasz koniec **Bone.002** w lewo, cały popychacz coraz bardziej odchyła się do góry (Rysunek 7.9.12b)!?

Jeżeli podczas przesuwania przyjrzyj się dokładniej zachowaniu kości **Bone.002**, zauważysz że w miarę oddalania się od „pozycji spoczynku” coraz bardziej odchyła się do dołu. A skoro kość się odchyła, to robi to także jej obiekt potomny — popychacz.

W takim razie, może jeżeli przesuwając będziemy koniec poprzedniej kości — amortyzatora (**Bone.001**) — to następna (**Bone.002**) podczas ruchu zachowa poziome położenie? Spróbujmy (Rysunek 7.9.13):



Rysunek 7.9.13 Podobny efekt dla kości **Bone.001**



Rysunek 7.9.14 Rozwiązanie problemu — dodanie do popychacza ograniczenia **Locked Track**.

Niestety, nic z tego. Pozostawiona „samej sobie” swobodna kość **Bone.003** odchyła się jeszcze bardziej niż poprzednio. Takie są, niestety, efekty działania kinematyki odwrotnej (**Auto IK**).

Ten problem można rozwiązać za pomocą ograniczenia:

- wstaw na linii ruchu środka popychacza (osi obrotu tłoka amortyzatora) pomocniczy, pusty obiekt (**Add→Empty** — szczegóły na str. 218). Ten obiekt będzie pełnić rolę „celu”;
- dodaj do popychacza ograniczenie **Locked Track** (szczegóły — zob. str. 237). W parametrach ograniczenia lokalna oś **Z** obiektu na śledzić umieszczony z przodu „cel”, a obrót może następować wyłącznie wokół lokalnej osi **X**;

Rysunek 7.9.14 przedstawia rezultat tego rozwiązania. Popychacz ignoruje, dzięki dodanemu ograniczeniu, poprzeczne obroty kości **Bone.002**. Gdy przesuwasz poziomo koniec kości **Bone.001**, cały zespół prawidłowo składa się i rozkłada.



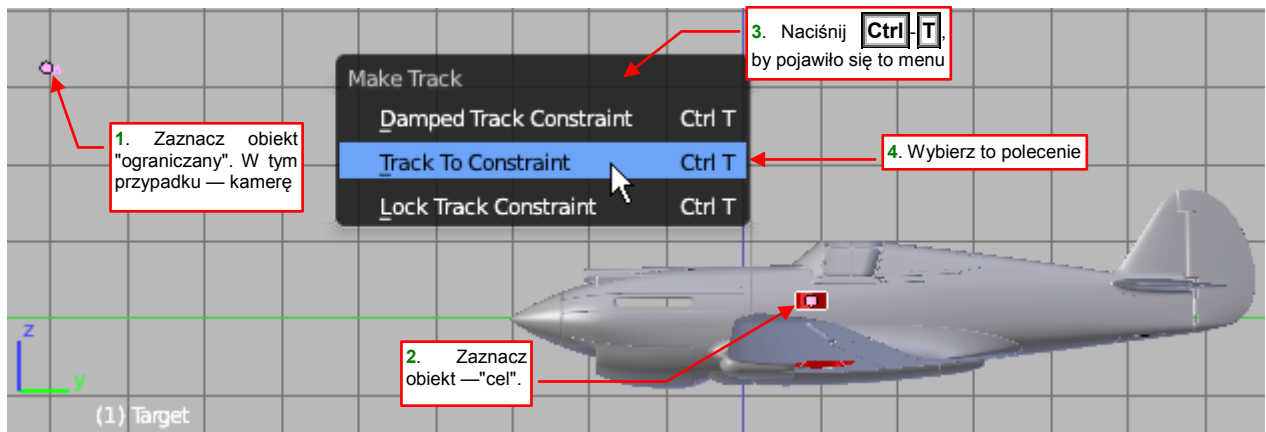
### 7.10 Przypisanie ograniczenia **Track To**

Obiekt z ograniczeniem **Track To** jest zawsze zorientowany w przestrzeni tak, by jego oś była skierowana na środek wskazanego celu (innego obiektu). Domyślnie w cel kierowana jest lokalna oś **Z**, ale oczywiście we właściwościach ograniczenia można wskazać inną (**X** lub **Y**).

Najszybsza metoda utworzenia ograniczenia **Track To** to:


- zaznaczenie obiektu/obiektów podlegających ograniczeniu;
- zaznaczenie obiektu - celu (to musi być ostatni zaznaczony — tzn. to musi być obiekt aktywny!);
- naciśnięcie skrótu **Ctrl-T** i wybranie z podręcznego menu polecenia **Track To Constraint**:

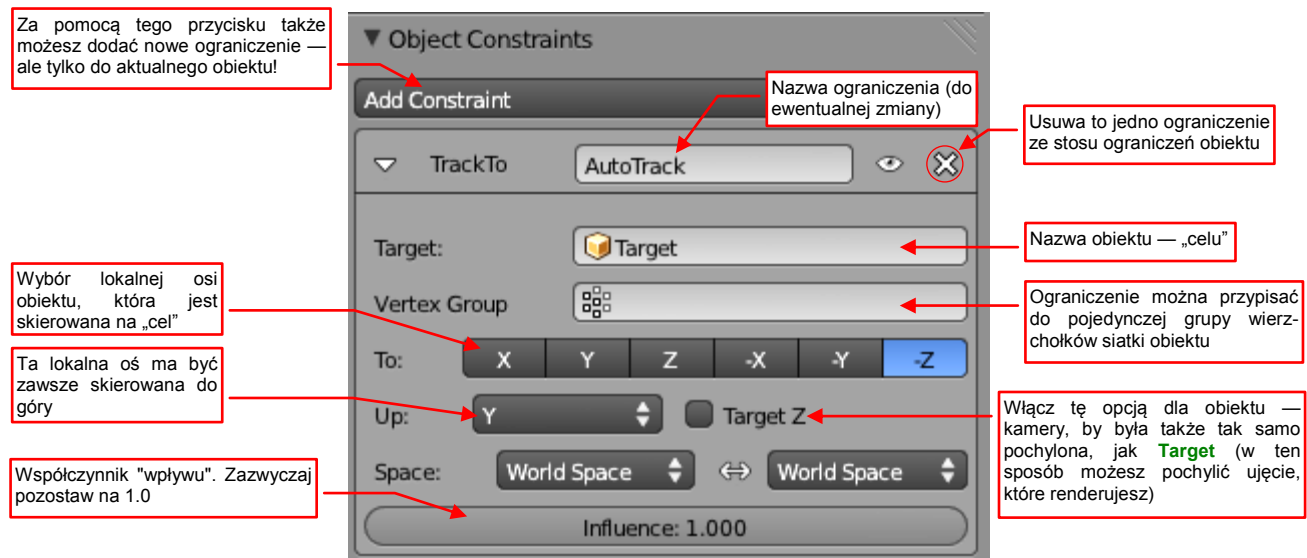
Rysunek 7.10.1 pokazuje, jak to zazwyczaj wygląda:



Rysunek 7.10.1 Dodanie ograniczenia **Track To** za pomocą skrótu **Make Track**

Bardziej ogólna metoda, pozwalająca dodać każde ograniczenie, polega na zaznaczeniu najpierw „celu” a potem obiektu „ograniczającego”, i wywołaniu polecenia **Object→Constraints→Add Constraint (with Targets)**. Lista rozwijalna, która się wówczas pojawi, zawiera wszystkie typy ograniczeń.

Właściwości ograniczenia **Track To** wybranego obiektu można zmienić w zestawie **Object Constraints** () i panelu o tej samej nazwie (Rysunek 7.10.2):



Rysunek 7.10.2 Właściwości ograniczenia **Track To**

W przypadku tworzenia powiązania kamery z celem, nie zapomnij włączyć przełącznika **Target Z**.

- Najprostszą drogą do usunięcia ograniczenia **Track To** jest użycie skrótu **Alt-T**. W menu, które wówczas się pojawi, warto wybrać opcję **Clear Track and Keep Transform**. Po jej wybraniu obiekt/obiekty zostaną "uwolnione", ale nie zmienią swojego położenia ani orientacji w przestrzeni.

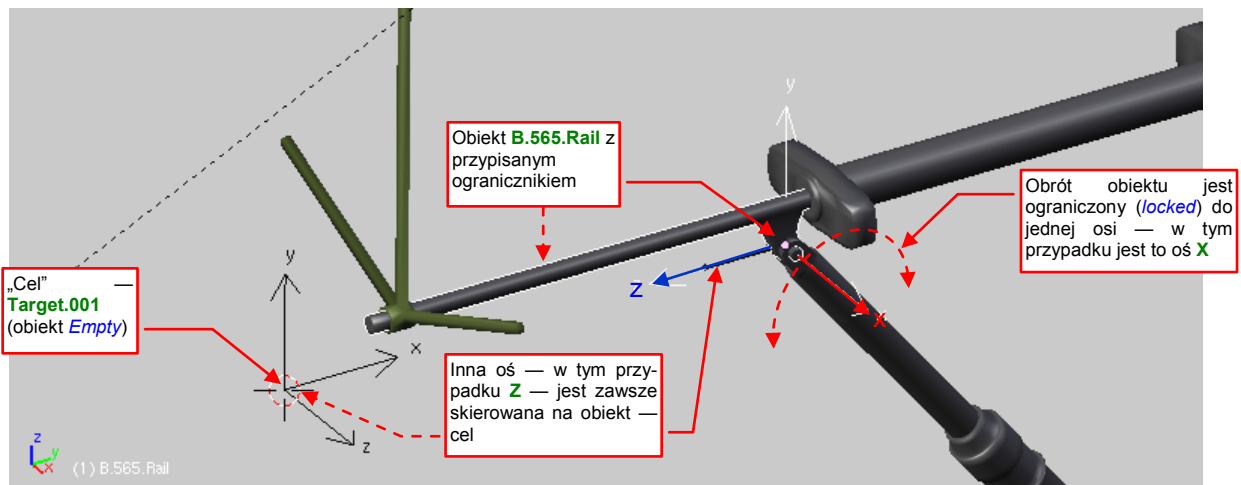
Innymi drogami do usunięcia ograniczenia są:

- wywołanie polecenia **Object→Constraints→Clear Constraints**;
- usunięcie pojedynczego ograniczenia ze stosu ograniczeń aktualnego obiektu (Rysunek 7.10.2).

Te metody nie gwarantują jednak zachowania aktualnej transformacji obiektu, jak to ma miejsce przy użyciu skrótu **Alt-T**.

### 7.11 Przypisanie ograniczenia **Locked Track**

Ograniczenie **Locked Track** działa podobnie do **Track To** (por. str. 228). Także tutaj obiekt obraca się zawsze tak, by wskazana w parametrach ograniczenia oś lokalnego układu współrzędnych była skierowana na środek wskazanego celu. Jednak w tym przypadku obiekt może się obracać tylko wokół jednej osi (Rysunek 7.11.1):

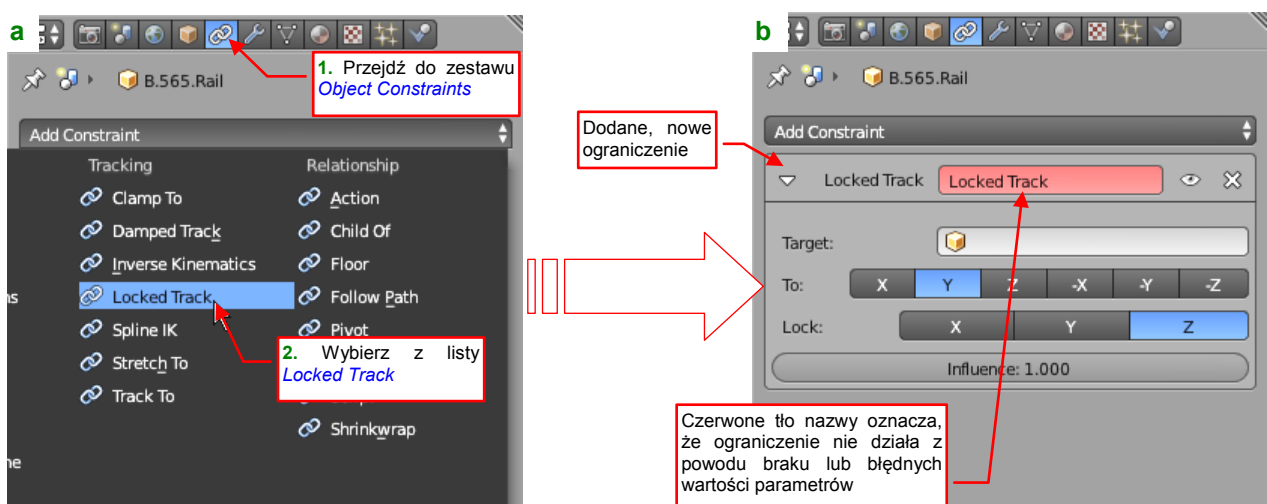


Rysunek 7.11.1 Zasada działania ograniczenia **Locked Track**.

Oznacza to, że wybrana oś może nie być dokładnie skierowana na cel, (bo np. cel jest przesunięty w bok, poza płaszczyznę obrotu). Tym niemniej obiekt z ograniczeniem **Locked Track** zawsze będzie się „starał wycelować” ją jak najlepiej — na tyle, na ile pozwala mu podana w parametrach oś obrotu.

- Ograniczenia typu **Locked Track** doskonale nadają się do modelowania mechanizmów. Prawie każdy element takich zespołów ma „jeden stopień swobody” — może się obracać wokół jednej osi.

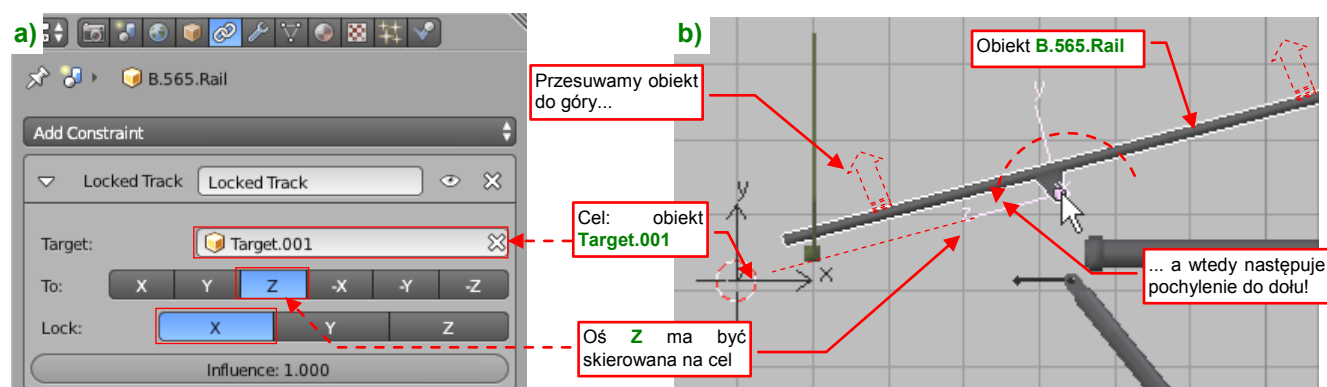
Aby dodać ograniczenie **Locked Track**, zaznacz odpowiedni obiekt, i w zestawie **Constraints**, naciśnij przycisk **Add Constraint** (Rysunek 7.11.2a):



Rysunek 7.11.2 Dodanie do obiektu ograniczenia **Locked Track**.

Z listy rozwijalnej, która się pojawi ponad przyciskiem, wybierz opcję **Locked Track**. Blender wstawi to ograniczenie na koniec listy („stosu”) ograniczeń przypisanych do obiektu. W pierwszej chwili pole **Locked Track** w panelu ograniczenia, (to z nazwą), jest czerwone (Rysunek 7.11.2b). To oznacza, że coś z parametrami tego ograniczenia jest nie tak. W tym przypadku w polu **Target** nie wpisaliśmy jeszcze nazwy obiektu — „celu”. Gdy to uzupełnimy, czerwony kolor zniknie z panelu.

Zróbmy to: wpisz w pole **Target** nazwę obiektu **Target.001** (którego używam wyłącznie w celach szkoleniowych). Oprócz tego na liście **To** zaznacz oś **Z**, a na **Lock** — **X** (Rysunek 7.11.3a):

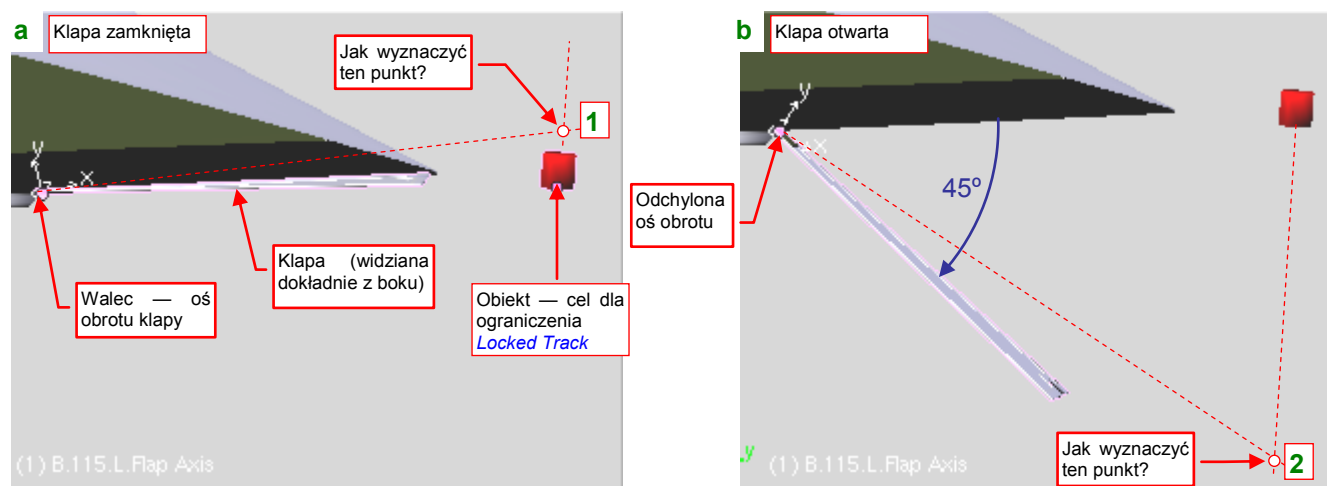


Rysunek 7.11.3 Przykład przypisania parametrów **Locked Track**.

Gdy teraz spróbujesz przesunąć ograniczony obiekt do góry — zobaczysz że zacznie się obracać tak, by jego oś **Z** nadal „celowała” w środek obiektu **Target.001** (Rysunek 7.11.3b). Gdybyś zaczął przesuwając ten obiekt **B.565.Rail** wzdłuż lokalnej osi **X**, nie spowodowałoby to żadnego dodatkowego obrotu (tym właśnie się różni to ograniczenie od ograniczenia **Track To**).

Ograniczenie **Locked Track** doskonale się nadaje do sterowania ruchem sterów, lotek lub kłap samolotu. Zazwyczaj przypisuje się je do ich osi obrotu (zamodelowanych jako oddzielny obiekt). Często jednak te osie nie są ustawione równoległe do globalnego układu współrzędnych. (Tak bywa, gdyż w czasie formowania samolotu nie zawsze byłeś w stanie o to odpowiednio zadbać). Pojawia się wówczas problem: jak umieścić obiekt — „cel” ograniczenia **Locked Track** we właściwym położeniu?

Przykładem takiego problemu jest kłapa skrzydła, którą pokazuje Rysunek 7.11.4:



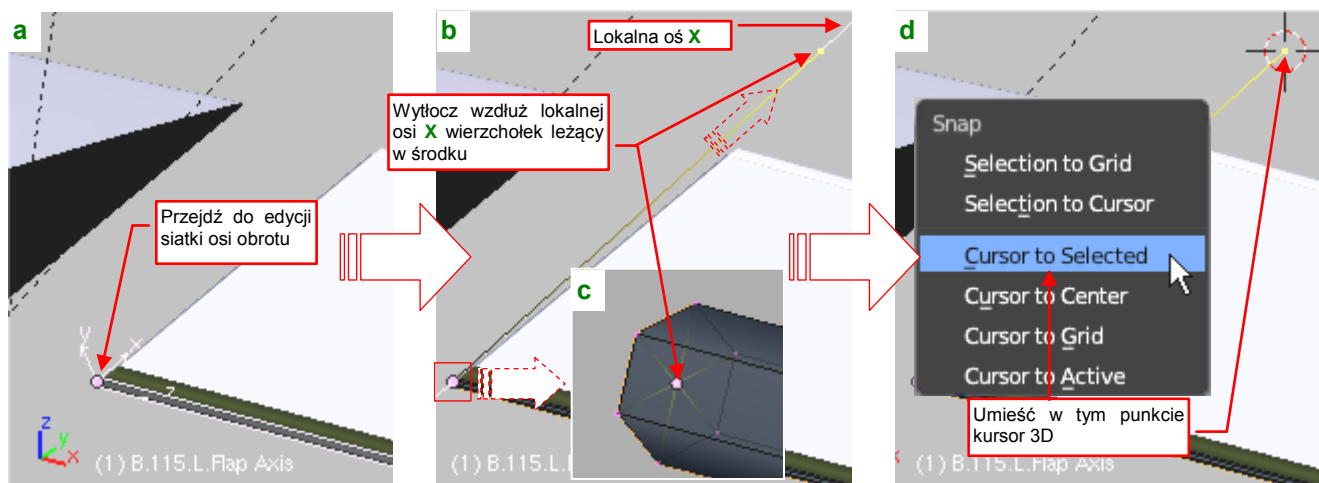
Rysunek 7.11.4 Problem: określenie dokładnego położenia „celu” ograniczenia **Locked Track**

Kłapa to z grubsza prostokątny fragment powierzchni. Jest przymocowana (relacją **Parent**) do długiego walca, imitującego jej zawiasy. Kłapę należy „zmechanizować”: przypiszemy osi **X** zawiasów ograniczenie **Locked Track**, śledząc specjalnie w tym celu przygotowany obiekt — „cel” (Rysunek 7.11.4a). Obrót będzie się odbywał wokół lokalnej osi **Z**. Dzięki temu użytkownik będzie mógł sterować kłapą za pomocą łatwo dostępnego uchwytu („celu”), a nie ukrytej głęboko osi, którą trudno jest wskazać myszką.

Niestety, lokalna oś **X** zawiasów kłapy jest pochylona w stosunku do globalnego układu współrzędnych, w którym umieściliśmy obiekt — „cel”. Jak ustawić ten „uchwyt” dokładnie w punkcie **1** (Rysunek 7.11.4a), aby po włączeniu ograniczenia zamknięta kłapa się nie „poruszyła”? Jak wyznaczyć drugie położenie „uchwytu”, odpowiadające całkowicie wychylonej (w tym przypadku — o kąt 45°) kłapie (punkt **2**, Rysunek 7.11.4b)?



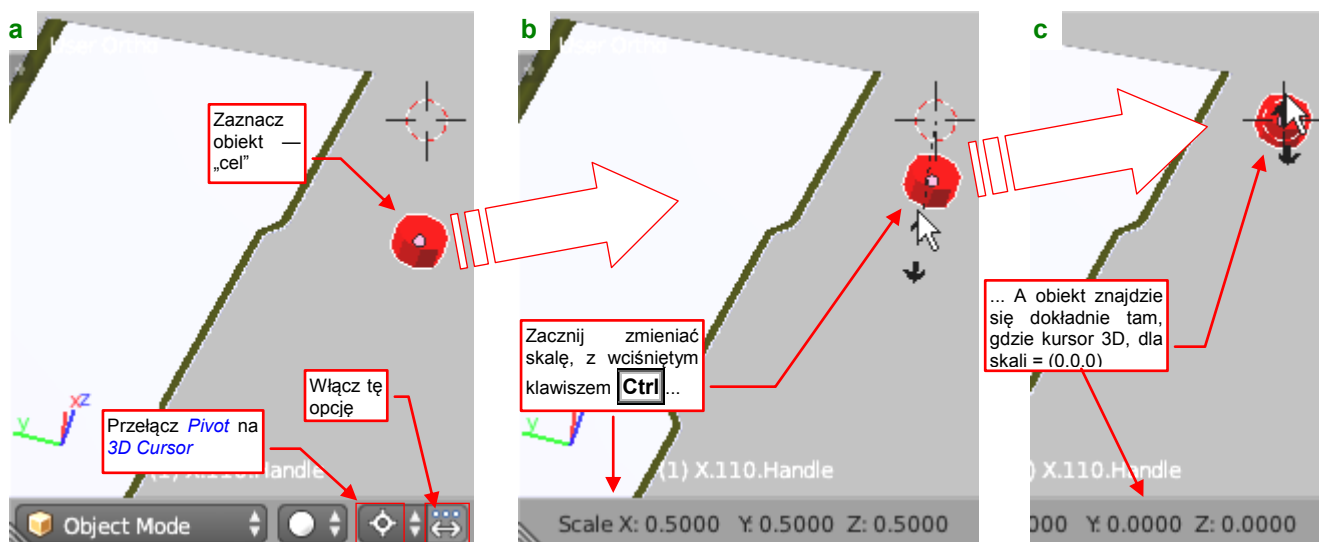
Aby wyznaczyć górne położenie „celu”, przejdź do edycji siatki zawiasów (Rysunek 7.11.5a):



Rysunek 7.11.5 Umieszczenie kursora 3D w górnym położeniu „celu”

W tym przypadku tak się szczęśliwie składa, że siatka ma wierzchołek w tym samym punkcie, co środek (punkt odniesienia) tego obiektu (Rysunek 7.11.5c). Zaznacz go, i wytłocz z niego nową krawędź wzdłuż lokalnej osi X (**E**, **X**, **X**). Odsuń jej koniec na jakąś większą odległość (Rysunek 7.11.5b). Powinna być na tyle duża, by końcowy wierzchołek znalazł się znacznie poza powierzchnią płata. Potem wystarczy wywołać polecenie, które umieści w tym punkcie kursor 3D: **Shift-S** (**Snap**), **Cursor to Selected** (Rysunek 7.11.5d).

Umieściliśmy już w odpowiednim miejscu „punkt zaczepienia” — kursor 3D. Mógłbyś teraz odczytać jego współrzędne i wpisać je jako nowe położenie uchwytu. Znam jednak prostszą metodę, nie wymagającą przepisywania, które zajmuje czas i podczas którego można się pomylić. To przesuwanie obiektów za pomocą zmiany skali (Rysunek 7.11.6):

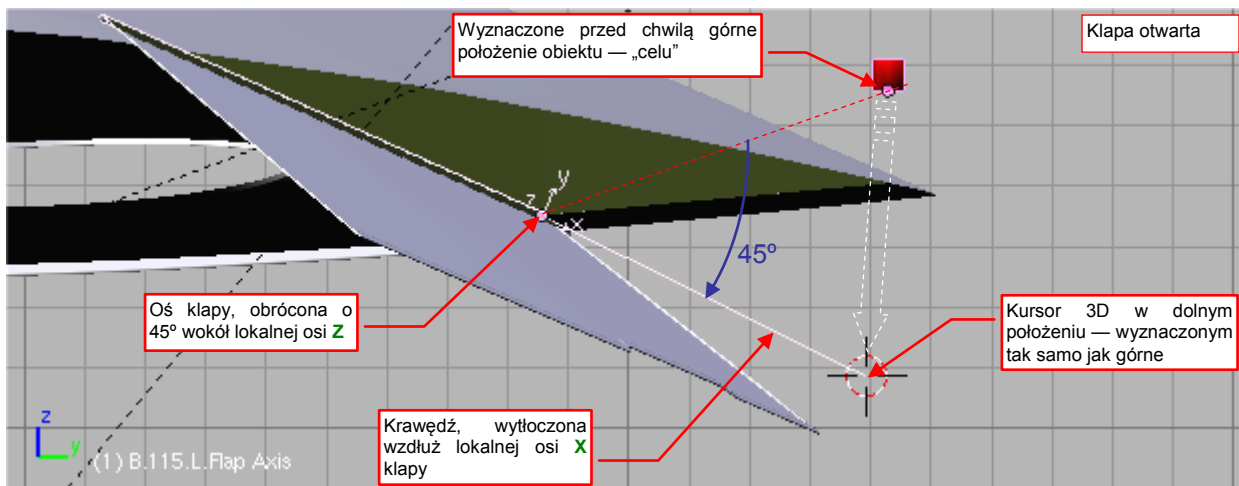


Rysunek 7.11.6 Umieszczenie środka obiektu — „celu” w punkcie wyznaczonym przez kursor 3D

Zaznacz obiekt, który ma być umieszczony tam, gdzie znajduje się kursor 3D. Następnie przełącz **Pivot** w tryb **3D Cursor** i włącz przełącznik **Manipulate center points**, umieszczony obok (Rysunek 7.11.6a). (Jego włączenie powoduje, że zmiana skali i obrót stosuje się nie do całego obiektu, a tylko do jego środka). Potem naciśnij **S** (**Scale**), i trzymając wciśnięty klawisz **Ctrl** przesunij kursor myszki tak, by znalazł się ponad kursorem 3D. Nasz obiekt („uchwyt”) podąży w tym samym kierunku, a gdy skala transformacji osiągnie (0,0,0) — jego środek znajdzie się w docelowym miejscu (Rysunek 7.11.6a,b). Zatwierdź tę transformację.

- Po skalowaniu z użyciem przełącznika **Manipulate center points** nie zapomnij go wyłączyć, bo inaczej nie będziesz mógł obrócić żadnego obiektu!

W analogiczny sposób możesz także określić drugie położenie „uchwyty”, odpowiadające klapie całkowicie otwartej (Rysunek 7.11.7):



Rysunek 7.11.7 Wyznaczenie dolnego położenia obiektu — „celu”

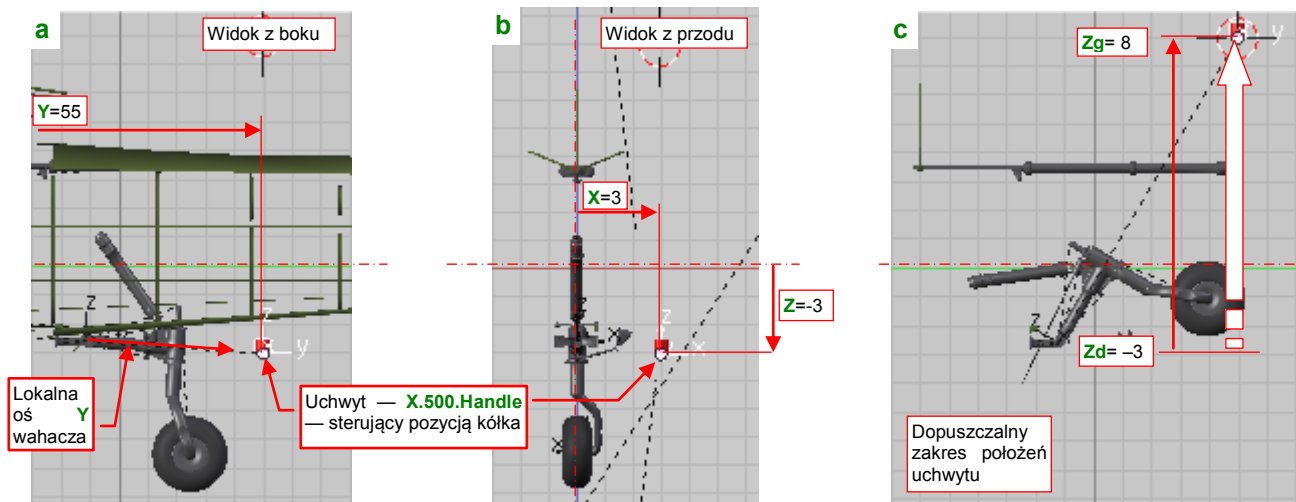
Wystarczy:

- obrócić oś (zawiasy) o odpowiedni kąt (tutaj: 45°);
- przejść do edycji siatki zawiasów;
- ponownie zaznaczyć nasz „roboczy” wierzchołek i umieścić na nim kursor 3D;
- zakończyć edycję siatki zawiasów;
- przesunąć „uchwyt” do kursora 3D;

Na koniec usuń, oczywiście, z siatki zawiasów wytłoczony podczas tej operacji wierzchołek!

### 7.12 Przypisanie ograniczenia *Limit Location*

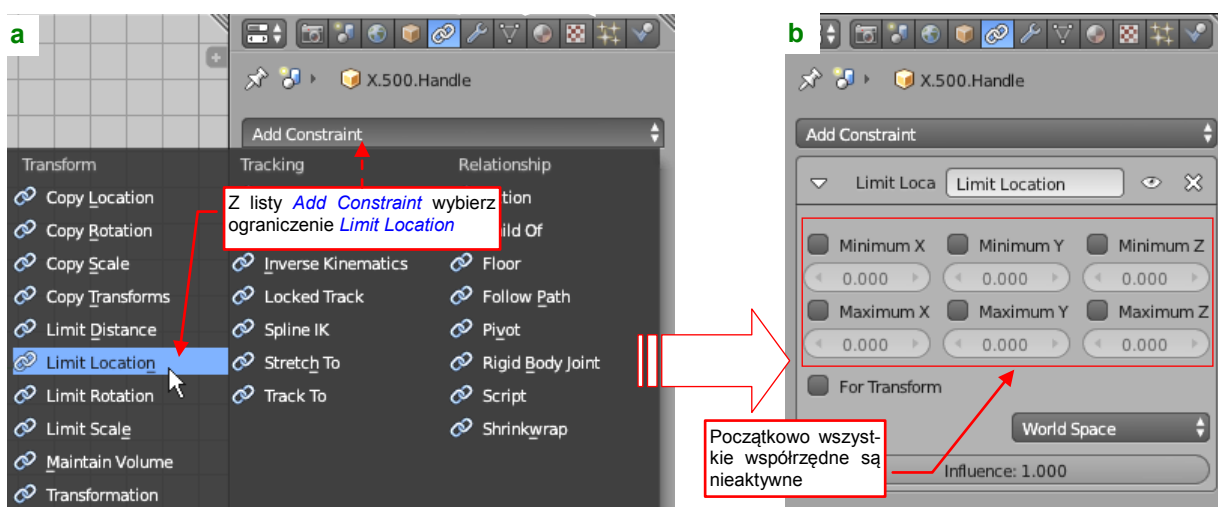
Aby części naszego modelu nie mogły się poruszać po całej scenie, tylko same „pilnowały” swoich właściwych położeń, można zastosować ograniczenie *Limit Location*. Oczywiście, opłaca się go użyć tylko dla kluczowych elementów. Na przykład — dla uchwytu odpowiedzialnego za chowanie i wysuwanie podwozia (Rysunek 7.12.1):



Rysunek 7.12.1 Położenie i dopuszczalny zakres ruchu „uchwytu”, sterującego pozycją kółka ogonowego.

O pozycji podwozia decyduje uchwyt **X.500.Handle**, którego środek jest śledzony przez lokalną oś Y wahacza (Rysunek 7.12.1a). (Ten efekt jest uzyskany pomocą ograniczenia *Locked Track* — por. str. 237). Gdy podwozie jest całkowicie otwarte, uchwyt znajduje się w punkcie  $X=3, Y=-55, Z=-3$  (Rysunek 7.12.1a,b). Na razie uchwyt można omyłkowo przesunąć w każdą stronę, bez żadnych ograniczeń. Użytkownik może się więc pomylić, i ustawić podwozie w nienaturalnej pozycji. Ograniczymy to, pozwalając uchwytowi **X.500.Handle** przesuwać się wzdłuż osi Z, w zakresie od -3 (podwozie wysunięte) do +8 (podwozie schowane) (Rysunek 7.12.1c).

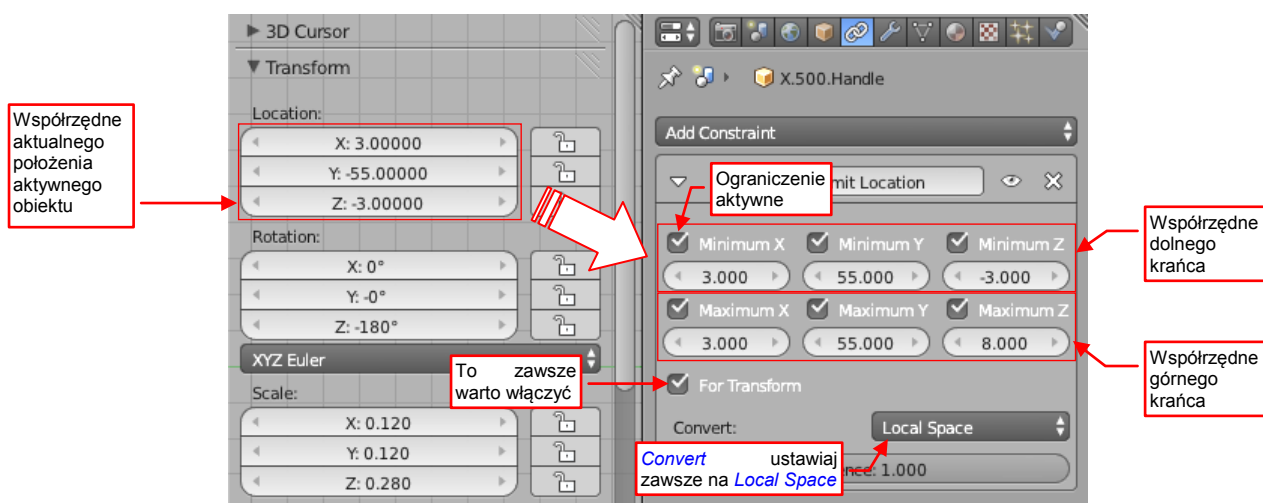
Aby dodać ograniczenie *Limit Location*, zaznacz obiekt (uchwyt), i w zestawie *Constraints* z listy *Add Constraint* wybierz opcję *Limit Location* (Rysunek 7.12.2a):



Rysunek 7.12.2 Dodanie do obiektu ograniczenia *Limit Location*.

Blender wstawi to ograniczenie na koniec listy („stosu”) ograniczeń przypisanych do obiektu. W pierwszej chwili wszystkie pola *Minimum\** i *Maximum\** są wyłączone (Rysunek 7.12.2b). Oznacza to, że nie obowiązuje jeszcze żadne z nich (tzn. ograniczenie nie jest aktywne).

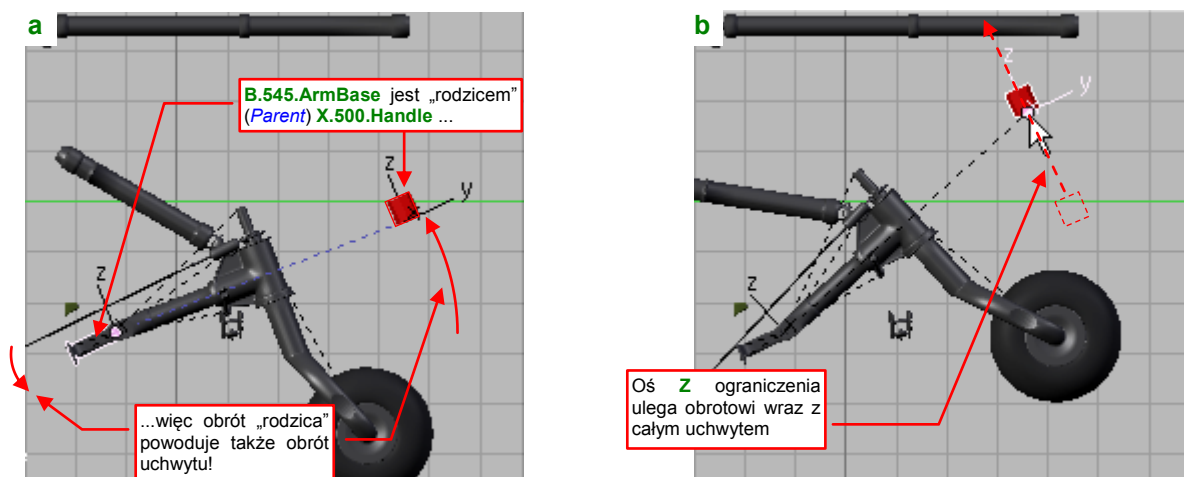
Zacznij od przepisania w pola **Minimum\*** i **Maximum\*** współrzędnych krańcowych położenia uchwytu. Możesz je np. odczytać z panelu **Transform** z przybornika **Properties** (N) (Rysunek 7.12.3):



Rysunek 7.12.3 Przykład ustawienia parametrów ograniczenia **Limit Location**.

Gdy je przepisziesz, włącz poszczególne ograniczenia (Rysunek 7.12.3b). W pole **Maximum Z** wpisz wartość dopuszczalnego, górnego położenia uchwytu. Warto jeszcze włączyć opcję **For Transform**. (Jeżeli tego nie zrobisz, po ewentualnym zdjęciu ograniczeń pozycja obiektu może Cię zaskoczyć). Przełącz także układ współrzędnych (**Convert**), w którym obliczane są ograniczenia na **Local Space**. Dlaczego właśnie ten? Także po to, by uniknąć niespodzianek. W przyszłości, podczas kompozycji sceny, możesz chcieć przechylić lub przesunąć model. Uchwyty powinny być wkomponowane w hierarchię jego zespołów, i podążać za ruchem całego samolotu. Przełączenie ich ograniczeń na lokalny układ współrzędnych to właśnie zapewnia.

Na przykład **X.500.Handle** jest przypisany do nieruchomego okucia wahacza: **B.545.ArmBase** (Rysunek 7.12.4a):



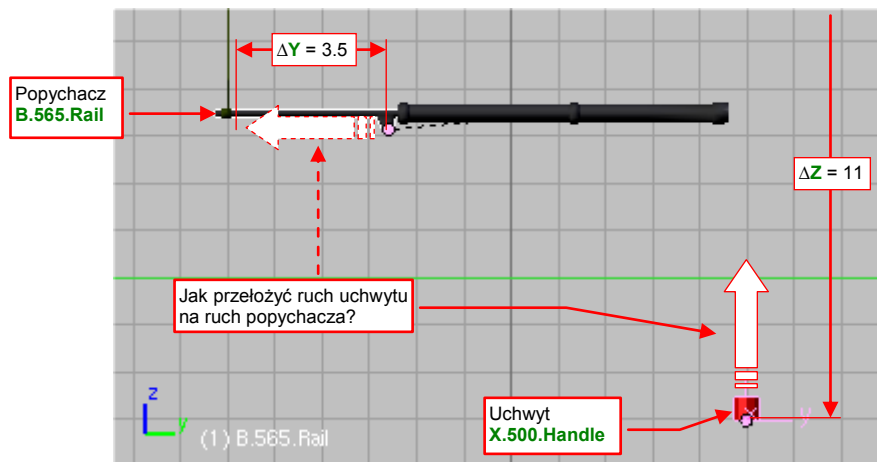
Rysunek 7.12.4 Rezultat wyboru dla **Limit Location** lokalnego układu współrzędnych (**Local**).

Gdy pochylisz to okucie (tak jak mógłbyś pochylić cały samolot, np. aby wyglądał jak podczas nurkowania), pochyli się odpowiednio także „sztywno” powiązany z nim uchwyt (Rysunek 7.12.4a). Gdybyś w ograniczeniu **Limit Location** pozostawił jako **Convert** domyślny układ współrzędnych **World Space**, to ograniczenie działałoby dalej pionowo, i podwozie „samo” zaczęłoby Ci się podczas obrotu okucia przesuwać. (Taki problem miałem we wcześniejszych wersjach Blendera — do 2.44 — w których **Limit Location** było zawsze obliczane w **World Space**). Na szczęście wybór **Local Space** powoduje, że osie ograniczenia są „przymocowane” do lokalnego układu współrzędnych uchwytu. Dzięki temu ich położenie względem reszty modelu nie ulega zmianie (Rysunek 7.12.4b) i w czasie animacji samolotu nie jesteś narażony na takie zaskoczenia, jak samoczynne otwieranie się podwozia.



### 7.13 Przypisanie ograniczenia *Transformation*

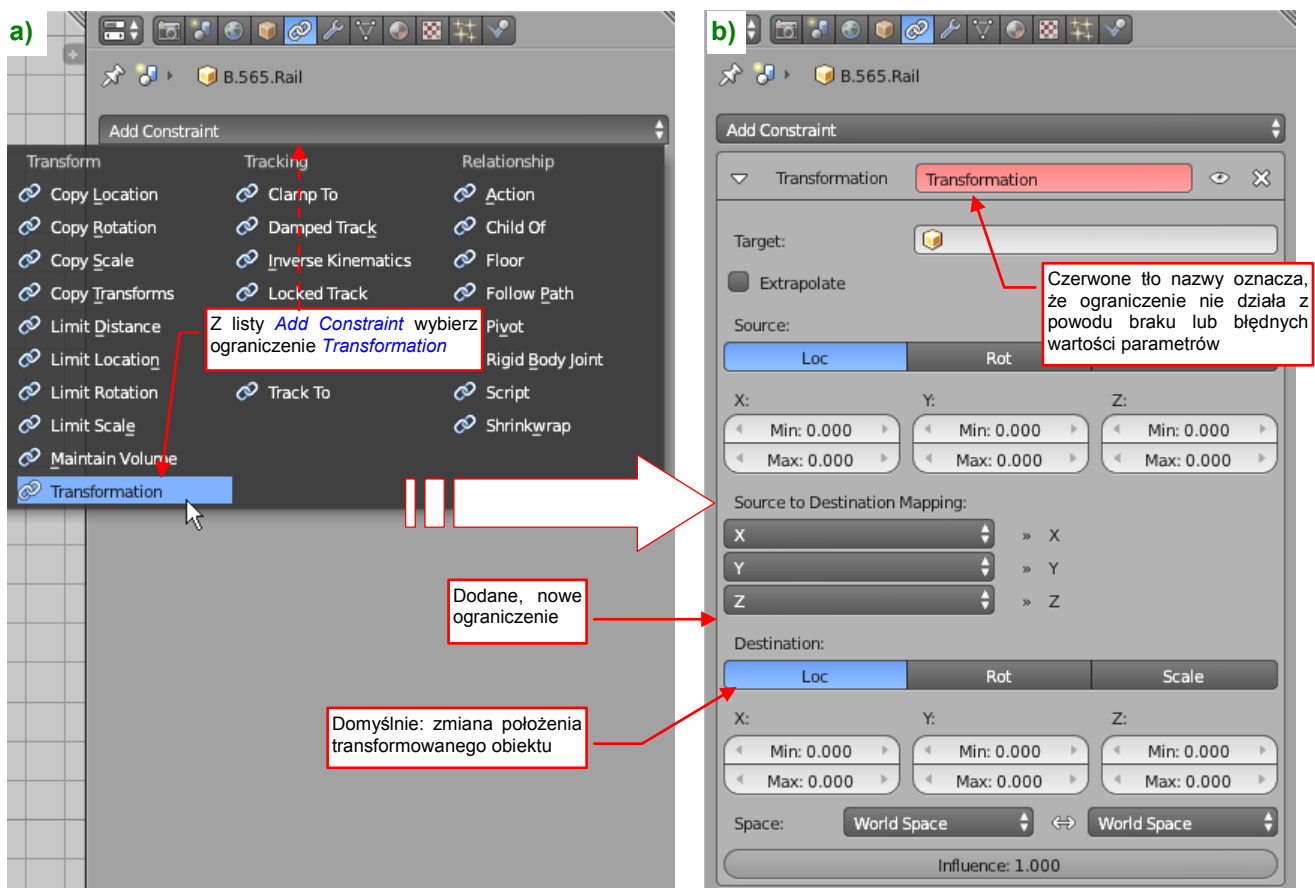
Ograniczenie *Transformation* można traktować jako rodzaj „uniwersalnej przekładni”. Pokażę je na przykładzie przełożenia ruchu pionowego uchwyty (X.500.Handle) na poziome przesunięcie popychacza (B.565.Rail) (Rysunek 7.13.1):



Rysunek 7.13.1 Przykład typowego zagadnienia dla ograniczenia *Transformation*.

Gdy użytkownik przesuwa uchwyt do góry, popychacz powinien przesuwać się w lewo, przy czym każdy z nich ma inną prędkość ruchu. W czasie gdy uchwyt X.500.Handle przesunie się o 11 jednostek, popychacz B.565.Rail ma przebyć tylko 3.5 (Rysunek 7.13.1). Aby uzyskać ten efekt, przypiszemy do popychacza ograniczenie *Transformation*.

Aby dodać to ograniczenie zaznacz obiekt (popychacz), i w zestawie *Constraints* z listy *Add Constraint* wybierz opcję *Transformation* (Rysunek 7.13.2a):

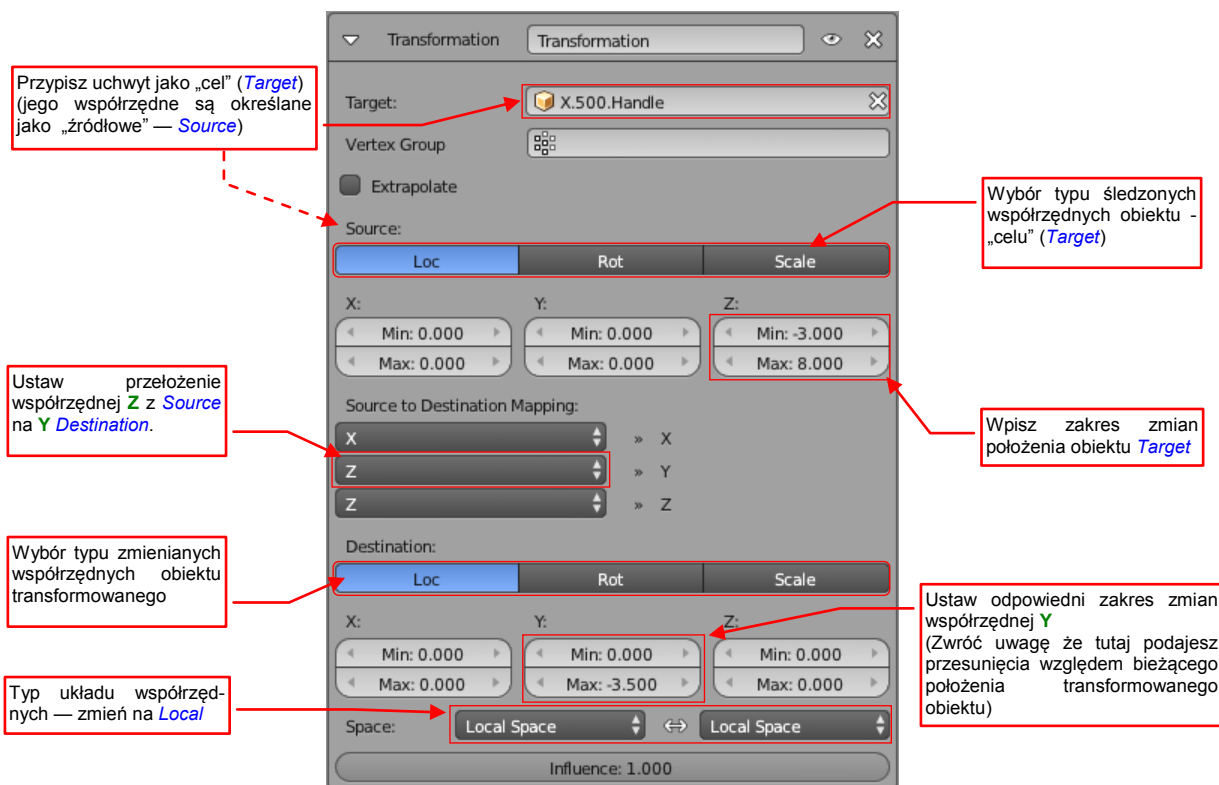


Rysunek 7.13.2 Dodanie do obiektu ograniczenia *Transformation*.

Blender wstawi to ograniczenie na koniec listy („stosu”) ograniczeń przypisanych do obiektu. W pierwszej chwili pole **Transformation** w panelu ograniczenia, (to z nazwą), jest czerwone (Rysunek 7.11.2b). To oznacza, że coś z parametrami tego ograniczenia jest nie tak. W tym przypadku w polu **Target** nie wpisaliśmy jeszcze nazwy obiektu — „celu”. Gdy to uzupełnimy, czerwony kolor zniknie z panelu.

Zwróć uwagę, że domyślnie włączone jest przełożenie położenia (**Loc**) obiektu — „celu” (**Source**) na położenie (**Loc**) obiektu **Destination** (obektu, do którego jest przypisane ograniczenie). Możliwe są jednak i inne przełożenia — np. obrotu (**Rot**) na położenie (**Loc**), skali (**Scale**) na obrót (**Rot**), itp. Wszystko zależy wyłącznie od tego, jaką kombinację tych opcji zaznaczysz.

Ustawienia transformacji dla popychacza **B.565.Rail** przedstawia Rysunek 7.13.3:



Rysunek 7.13.3 Przykład ustawienia parametrów ograniczenia **Transformation**.

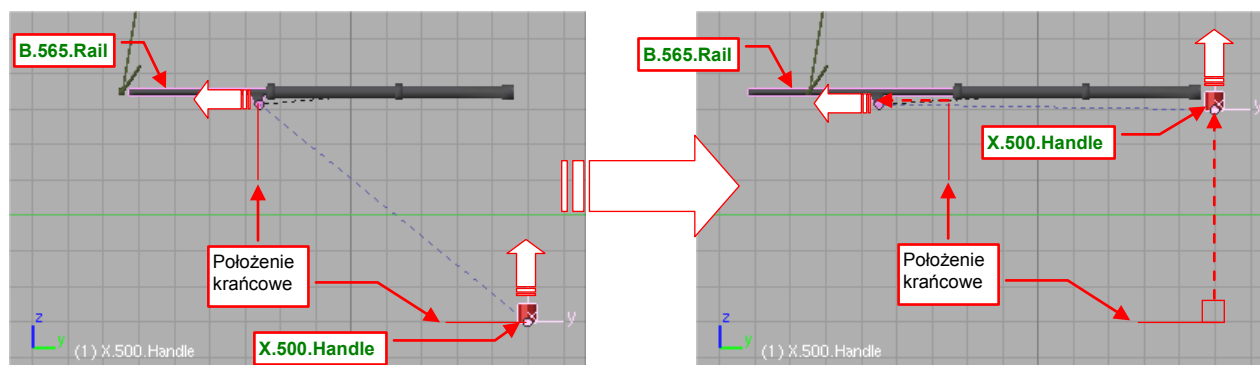
W pole **Target** wpisałem nazwę uchwytu — **X.500.Handle**. W zespole **Source** pozostawiłem śledzenie położenia (**Loc**), i wpisałem pełen zakres dopuszczalnych ruchów uchwytu (**Z: Min** = -3.00, **Max** = 8.00). Ten ruch przekładam na zmianę współrzędnej **Y** popychacza. Dlatego w zespole **Source to Destination Mapping** wybrałem dla współrzędnej **Y** z listy rozwijalnej kombinację **Z→Y** (Rysunek 7.13.3). W zespole **Destination** podajemy wartości zmiany odpowiednich współrzędnych. Dlatego w polu **Y:Min** pozostawiłem wartość 0.00. W polu **Y:Max** wpisałem wartość mniejszą: - 3.50 (w ten sposób zapewniłem popychaczowi ruch z prawej na lewo). Oczywiście, zmieniłem także typ (**Space**) układów współrzędnych celu i obiektu transformowanego na **Local Space**, by to ograniczenie było niezależne od orientacji modelu w przestrzeni (por. str. 242).

Ustawienia, które pokazuje Rysunek 7.13.3, przekazują Blenderowi prostą zależność:

- dla położenia uchwytu pomiędzy **Z** = -3.00 ... +8.00 współrzędna **Y** punktu odniesienia (**origin**) popychacza ma się proporcjonalnie zmniejszać, o wartości od 0.00 do -3.50;

Wartości przesunięcia dla pośrednich położenia obiektu transformowanego są wyliczane w sposób proporcjonalny. Taka zależność nazywana jest także liniową: np. gdy uchwyt jest „w połowie drogi” — **Z** = +2.50 — to popychacz także jest już przesunięty o połowę dystansu: **Y** = - 1.75.

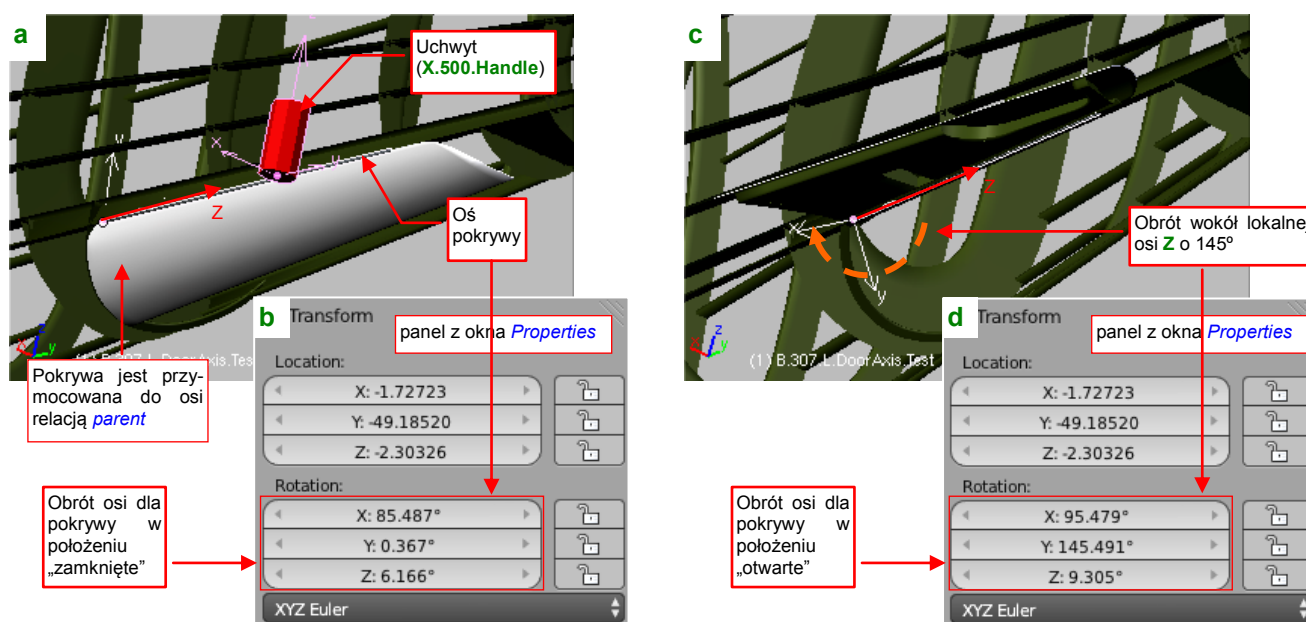
Rysunek 7.13.4 przedstawia rezultat działania ograniczenia **Transformation** — ruch popychacza **B.565.Rail** wymuszony przez przesunięcie uchwytu **X.500.Handle**:



Rysunek 7.13.4 Działanie ograniczenia **Transformation**

Liniowy sposób działania ograniczenia **Transformation** powoduje, że zazwyczaj nie można go użyć do „przełożenia” bardziej złożonych ruchów, np. równoczesnego obrotu względem wszystkich trzech osi. (W przypadku tak złożonego obrotu pośrednie położenia obiektu mogą być czasami zaskakujące). Należy sobie zdawać sprawę że nawet zwykłe pokrywy (np. podwozia) zamykane za pomocą popychaczy nie obracają się w sposób liniowy. Obrót uzyskany za pomocą **Transformation** będzie tylko przybliżeniem ich prawdziwego ruchu.

Dla przykładu, spróbujmy przełożyć za pomocą ograniczenia **Transformation** przesunięcie uchwytu na obrót pokrywy podwozia. Zaczniemy od sprawdzenia jak zmieniają się jego kąty obrotu (Rysunek 7.13.5):



Rysunek 7.13.5 Kąty obrotu, wyświetlane dla dwóch krańcowych położení pokrywy podwozia

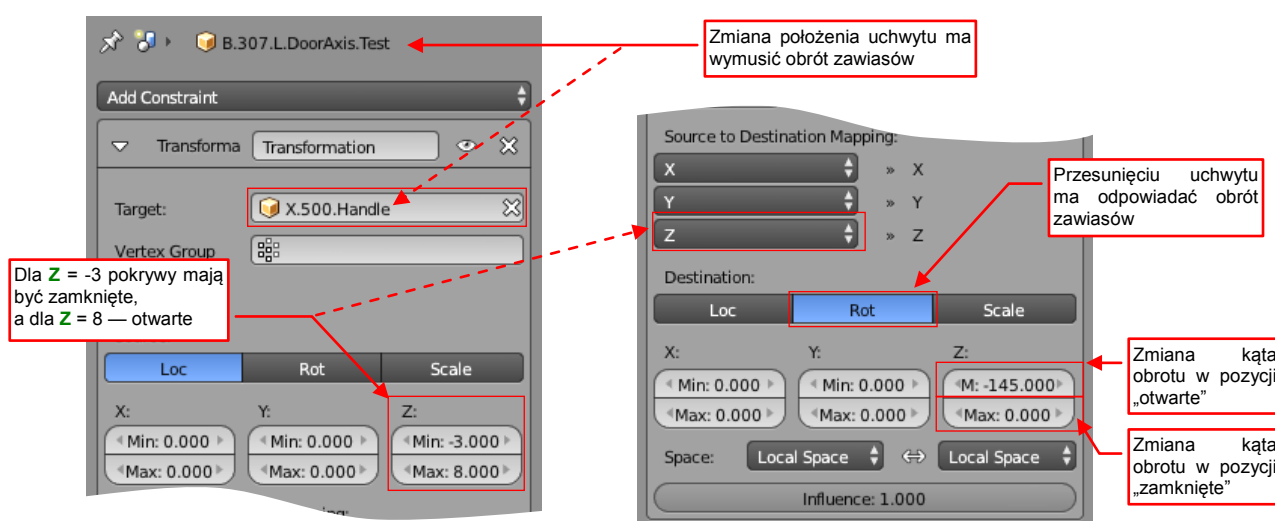
Zawiasy pokrywy, które należy obrócić, to zwykle cylindry, wyciągnięte do odpowiedniego rozmiaru wzdłuż lokalnych osi **Z** (Rysunek 7.13.5a). Ich „rodzicem” jest kadłub. Są względem niego obrócone — wokół wszystkich trzech osi (por. panel **Transform** z przybornika **Properties** — Rysunek 7.13.5b). Do zawiasów jest „przymocowana” (relacją **Parent**) sama pokrywa. Aby ją otworzyć, należy obrócić zawias wokół lokalnej osi **Z** o 145° (Rysunek 7.13.5c). Porównaj jednak w panelu **Transform** (przybornik **Properties**) obroty wokół osi **X**, **Y**, **Z** przed i po tej transformacji (por. Rysunek 7.13.5b z Rysunek 7.13.5d). Zauważ, że każdy z nich uległ zmianie: wokół osi **X** (z 85.487° na 95.479°), wokół osi **Y** (z 0.367° na 145.491°), wokół osi **Z** (z 6.166° na 9.305°).

- W panelu **Transform** przybornika **Properties** Blender wyświetla obroty względem osi obiektu „rodzica”. Jednak ich stan początkowy jest taki, jaki istniał w chwili przypisania poleceniem **Set Parent**.

Wygląda na to, że Blender „zapamiętuje” kąty obrotu obiektu w chwili gdy przypisujesz go do obiektu nadrzędnego poleceniem **Set Parent**. (Dokładniej rzecz biorąc — zapamiętuje w tym momencie tzw. „przekształcenie odwrotne”). Stąd kąty obrotu większości obiektów, które widzisz w oknie **Properties**, nie są ani kątami wyrażonymi w globalnym układzie współrzędnych, ani kątami mierzonymi względem osi „rodzica”. Stwarza to problem przy zastosowaniu ograniczenia **Transformation**, bo często obrót wokół jednej osi jest wyrażony za pomocą złożenia trzech obrotów względem globalnego układu współrzędnych!

- Zmianę kąta obrotu dla ograniczenia **Transformation** na pewno poprawnie odczytasz sprawdzając najpierw poleceniem **Rotate** (**R**), o jaki kąt musisz obrócić obiekt wokół jego lokalnej osi (np. w przypadku który pokazuje Rysunek 7.13.5b to 145°). Kąty wyświetlane w **Properties:Transform** mogą być mylące!

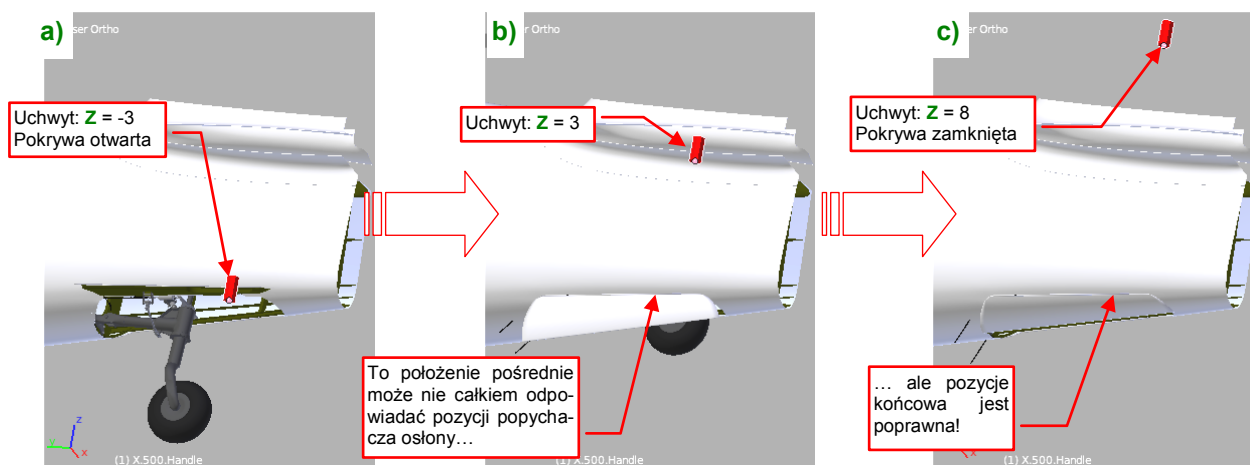
No dobrze, skoro znamy już kąt obrotu, to przypiszmy do zawiasów ograniczenie **Transformation**. Powiążmy je ze zmianą położenia (**Loc**) uchwyty (**Target: X.500.Handle**) wzdłuż osi **Z** (od **Z** = -3 do **Z** = 8) (Rysunek 7.13.6):



Rysunek 7.13.6 Ograniczenie, przypisane do zawiasów pokryw podwozia

Ustalmy w obszarze **Destination**, że zmieniać się ma obrót (**Rot**). W sekcji **Source to Destination Mapping** nie musimy niczego zmieniać, bo przełożenie **Z→Z** jest tam ustawione domyślnie. W pole **Z:Min** wpiszmy zmianę kąt obrotu wokół osi **Z** dla położenia „otwarte”. W pola **Z:Max** pozostawiamy (w tym przypadku) wartość 0.

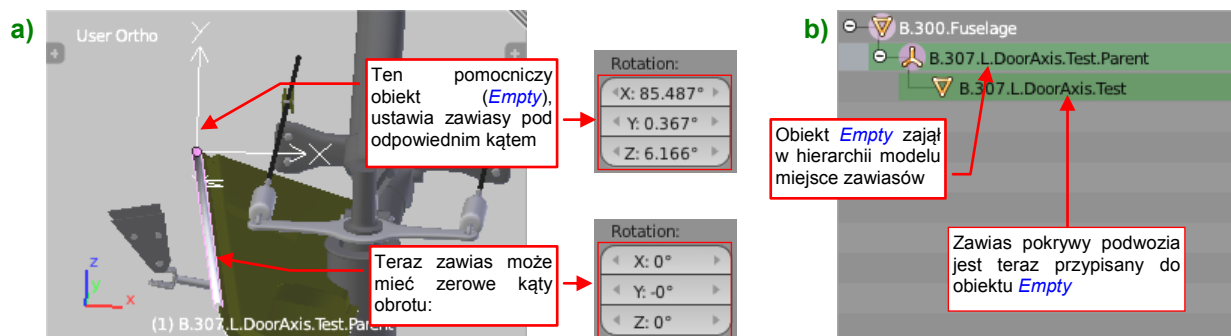
Po przypisaniu ograniczenia spróbuj przesunąć uchwyt (Rysunek 7.13.7). Zawiasy obracają się automatycznie w ten sam sposób, jak to sprawdziliśmy obracając je „ręcznie” wokół osi **Z**:



Rysunek 7.13.7 „Liniowa” droga od jednego do drugiego położenia krańcowego



W Blenderze wersji 2.67 i wcześniejszych ograniczenie *Transformation* z takimi ustawieniami działało tylko wtedy, gdy kąty obrotu zawiasu (wyświetlane w panelu *Properties: Transform*) były zerowe. Aby to uzyskać musiałem wstawić jako rodzica tego elementu (Rysunek 7.13.8b) pomocniczy obiekt typu *Empty* (por. str. 218). Ten nowy „rodzic” jest ustawiony równoległe do zawiasu (w jego *Properties: Transform* zobaczysz takie same kąty — por. Rysunek 7.13.8a), dzięki czemu zawias może mieć wszystkie kąty zerowe:

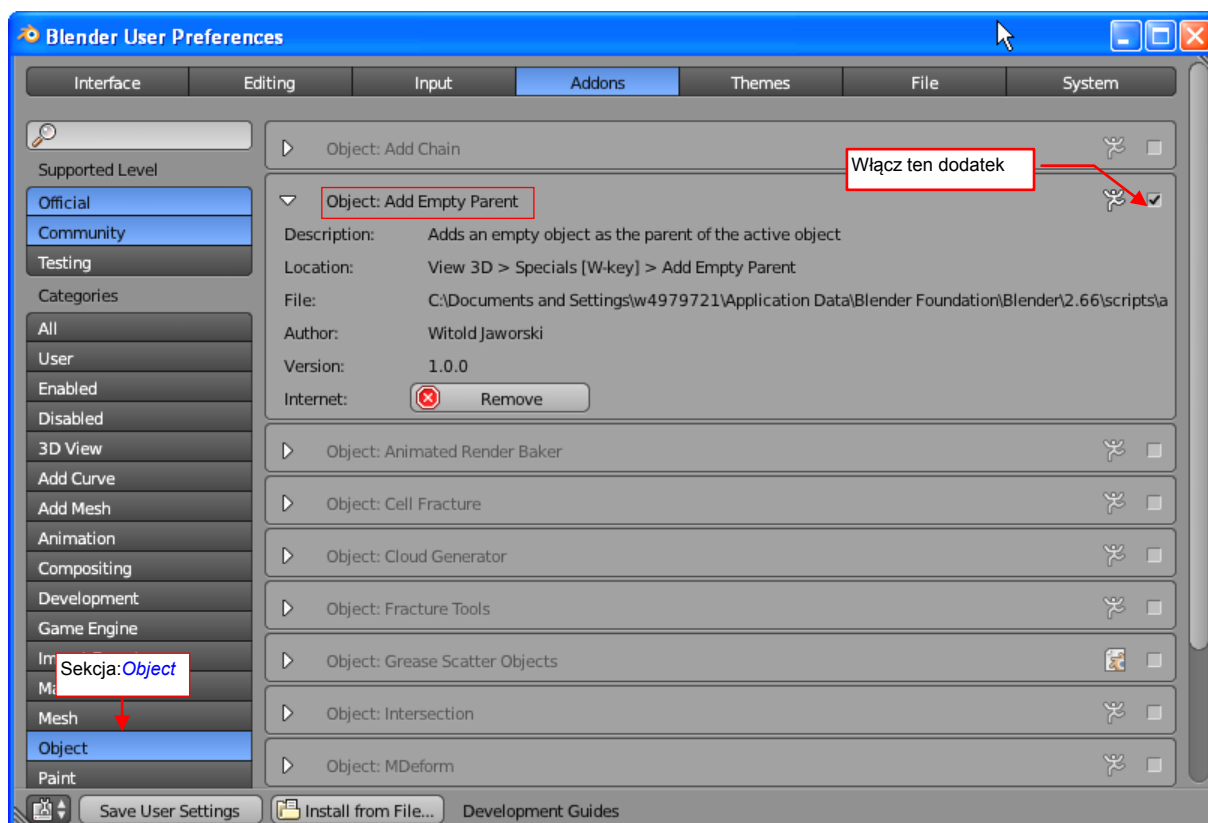


Rysunek 7.13.8 Pomocniczy obiekt — nowy „rodzic” zawiasu

Wstawienie takiego obiektu pomocniczego, jaki pokazuje Rysunek 7.13.8, wymagało nieco pracy. Trzeba było najpierw odłączyć obiekt zawiasów od dotychczasowego rodzica (*Clear Parent*), a potem wstawić w to samo miejsce pomocniczy obiekt *Empty*. Następnie zapamiętać kąty obrotu zawiasu, wyzerować je, i przypisać zawias do obiektu *Empty* (*Set Parent*). Wreszcie przypisać obiektowi *Empty* zapamiętane kąty obrotu i przypisać go do oryginalnego rodzica zawiasów.

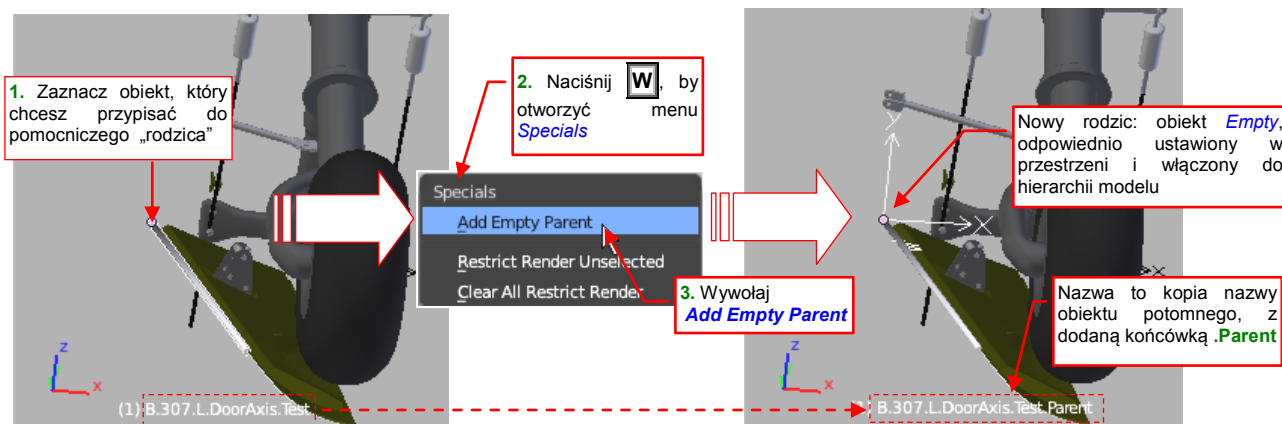
Aby ułatwić sobie taką operację, przygotowałem mały dodatek o nazwie *Add Empty Parent*. Myślę, że to polecenie może się nadal przydać do wielu innych zastosowań, dlatego krótko je tutaj opiszę.

Plik dodatku znajdziesz w pakiecie *source.zip*, nazywa się tam *source\scripts\addons\object\_add\_parent.py* (por. str. 16). Po zainstalowaniu powinieneś go znaleźć w sekcji *Addons: Object* (por. str. 183) i aktywować (Rysunek 7.13.9):



Rysunek 7.13.9 Aktywacja dodatku *Add Empty Parent*

Użycie dodatku jest proste: zaznacz obiekt, do którego chcesz uzyskać zerowe kąty obrotu. Następnie naciśnij **W**, by otworzyć menu *Specials*. Wywołaj stamtąd polecenie *Add Empty Parent*, a w tym samym miejscu co zaznaczony obiekt pojawi się odpowiedni obiekt *Empty* (Rysunek 7.13.10):



Rysunek 7.13.10 Użycie dodatku *Add Empty Parent*

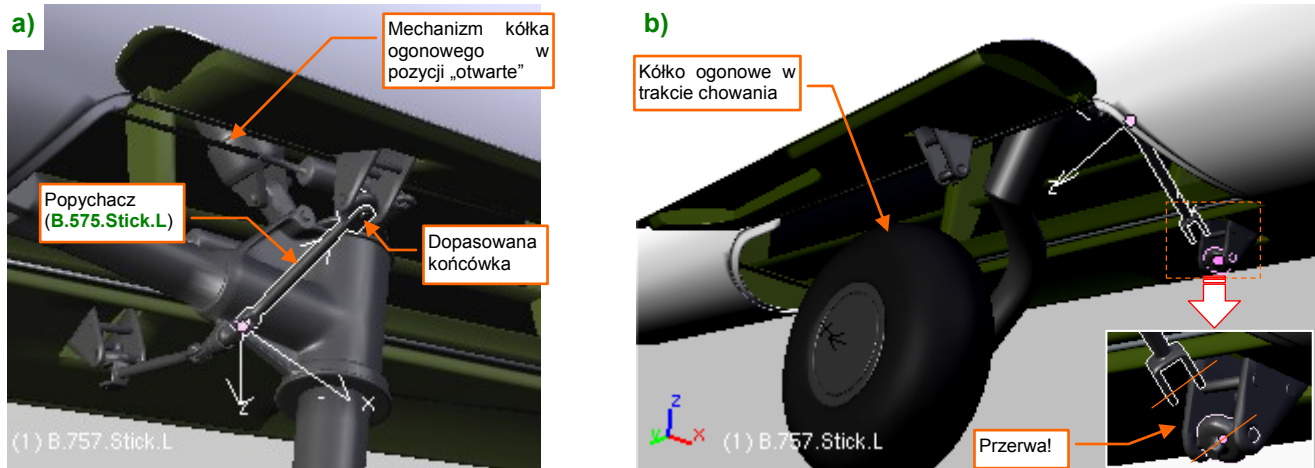
Skrypt umieszcza obiekt *Empty* na tej samej warstwie (warstwach) co zaznaczony obiekt potomny. Nadaje mu taką samą nazwę, z końcówką *.Parent*.

- Pamiętaj, aby przed przypisaniem, zmianą lub usunięciem rodzica zmniejszyć współczynniki *Influence* wszystkich ograniczeń obiektu potomnego do zera! Gdy o tym zapomnisz, Blender przypisze temu obiektowi jakieś dziwne współrzędne. Gdy nowy rodzic jest już przypisany, możesz z powrotem przestawić współczynniki *Influence* ograniczeń na ich oryginalne wartości (zazwyczaj to 1.0);

Takie szybkie wstawienie dodatkowego obiektu nadrzędnego przydaje się także do innych zastosowań (np. do obejścia problemu z ujemną skalą transformowanego obiektu)

### 7.14 Przypisanie ograniczenia *Stretch To*

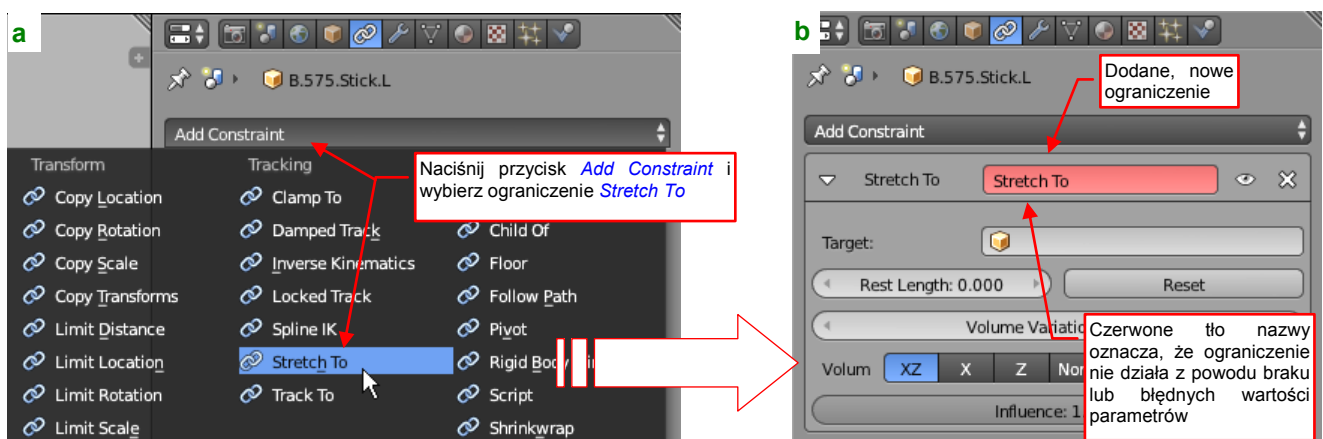
Ograniczenie *Stretch To* łączy w sobie dwie operacje: skierowania na “cel” określonej osi obiektu (przypomina w tym działanie *Locked Track*), i zmianę skali wzdłuż tej osi. Nadaje się doskonale do drobnych “naciągnięć”, maskujących pewne niedokładności w torach ruchu poszczególnych części mechanizmów. Popatrz, na przykład, na popychacz pokryw kółka ogonowego (Rysunek 7.14.1):



Rysunek 7.14.1 Przykład typowego zagadnienia dla ograniczenia *Stretch To*.

Rysunek 7.14.1a) pokazuje mechanizm podwozia w pozycji “otwarte”. Popychacz (B.575.Stick.L) jest tu skierowany na okucie pokryw ograniczeniem *Locked Track*. W tym położeniu wszystko wygląda poprawnie — w koniec popychacza wpasowana jest oś beczulki okucia. Gdy jednak zaczniesz składać podwozie, zauważysz że w szerokim zakresie położen pośrednich popychacz “odrywa” się od okucia (Rysunek 7.14.1b). Na zmianę dystansu pomiędzy obiektem i jego celem ograniczenie *Track To* nic poradzić nie może<sup>1</sup>. Pomoże za to inne — *Stretch To*.

Aby dodać ograniczenie *Stretch To*, zaznacz obiekt (popychacz), a potem w zestawie *Constraints* z listy *Add Constraint* wybierz *Stretch To* (Rysunek 7.14.2a):

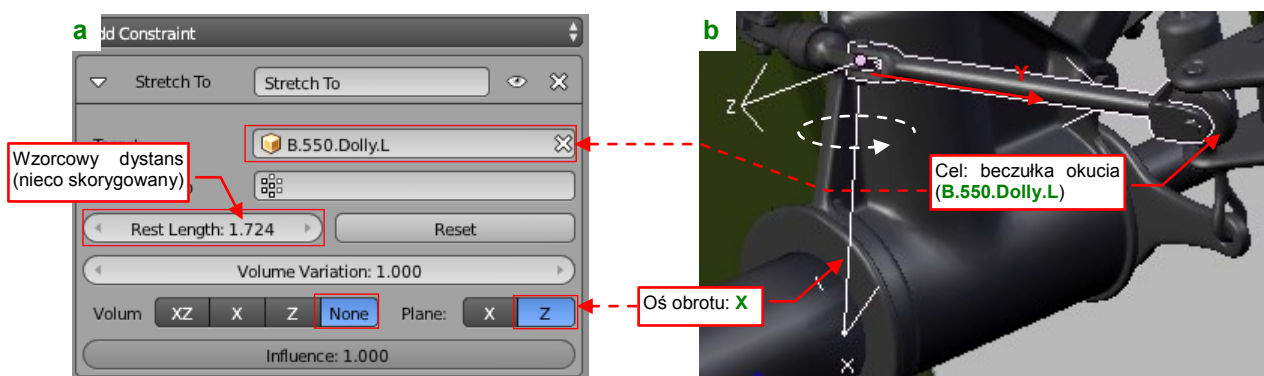


Rysunek 7.14.2 Dodanie do obiektu ograniczenia *Stretch To*.

W pierwszej chwili pole *Stretch To* w panelu ograniczenia, (to z nazwą), jest czerwone (Rysunek 7.11.2b). To oznacza, że coś z parametrami tego ograniczenia jest nie tak. W tym przypadku w polu *Target* nie wpisaliśmy jeszcze nazwy obiektu — „celu”. Gdy to uzupełnimy, czerwony kolor zniknie z panelu.

<sup>1</sup> Oczywiście, cały problem bierze się z niezbyt dokładnego odwzorowania obrotu pokryw. Na etapie, który pokazuje Rysunek 7.14.1b), powinna być już bardziej obrócona. Niestety, zrezygnowałem z dokładnego odwzorowanie jej ruchu (jest bardzo nieliniowy, więc byłoby to bardzo pracochłonne). W efekcie muszę “tuszować” takie niedociągnięcia, jak przerwy pojawiające się między elementami mechanizmu.

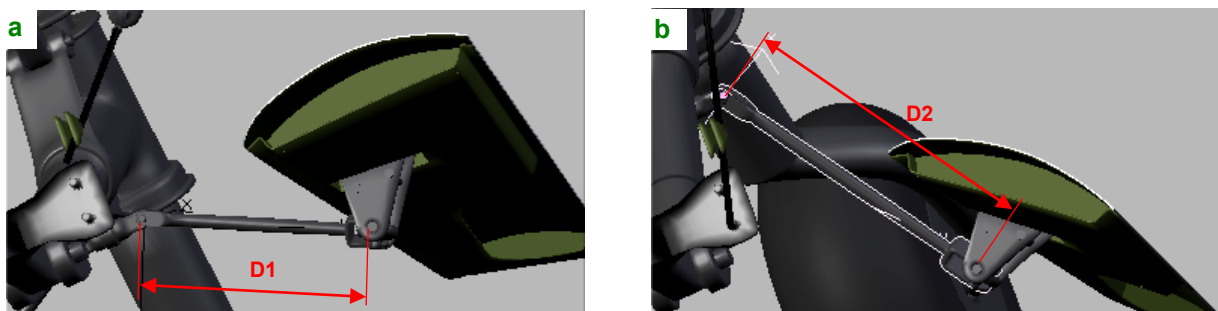
Wpisz w pole **Target** nazwę boczki okucia (**B.550.Dolly.L**). Oprócz tego na liście **Plane** zaznacz oś **X**, a na **Volume** — **None** (Rysunek 7.14.3a):



Rysunek 7.14.3 Przykład przypisania parametrów **Stretch To**.

Ograniczenie **Stretch To** zawsze „celuje” we wskazany w polu **Target** obiekt lokalną osią **Y** (Rysunek 7.14.3b). (Nie można tego zmienić, tak jak w **Locked Track**). Jeżeli masz inaczej zorientowane osie — musisz wcześniej obrócić siatkę względem środka obiektu. Możesz nadal, podobnie jak w **Locked Track**, określić oś, która ma pozostać nieruchoma — **X** lub **Z**. (Traktuj ją jako „oś obrotu” — por. Rysunek 7.14.3b). Po przypisaniu obiektu Blender odnotowuje w polu **Rest Length** odległość do środka „celu”. Od tej chwili zaczyna zmieniać skalę obiektu wzdłuż osi **Y** o tyle, o ile różni się aktualny dystans od dystansu wpisanego w **Rest Length**.

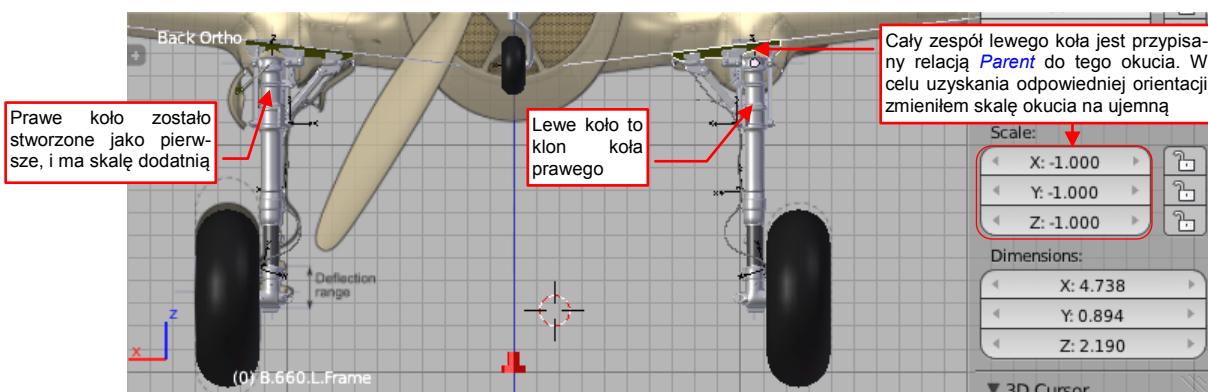
Rezultat działania ograniczenia pokazują ujęcia popychacza w dwóch kolejnych położeniach mechanizmu (Rysunek 7.14.4a i b są wykonane dokładnie w tej samej projekcji):



Rysunek 7.14.4 Rezultat ograniczenia **Stretch To**.

Koniec popychacza trzyma się „jak przyklejony” boczki okucia. Przy okazji rozciąga się jak guma — zwróć uwagę, że dystanse **D1** i **D2** na ilustracjach bardzo się różnią!

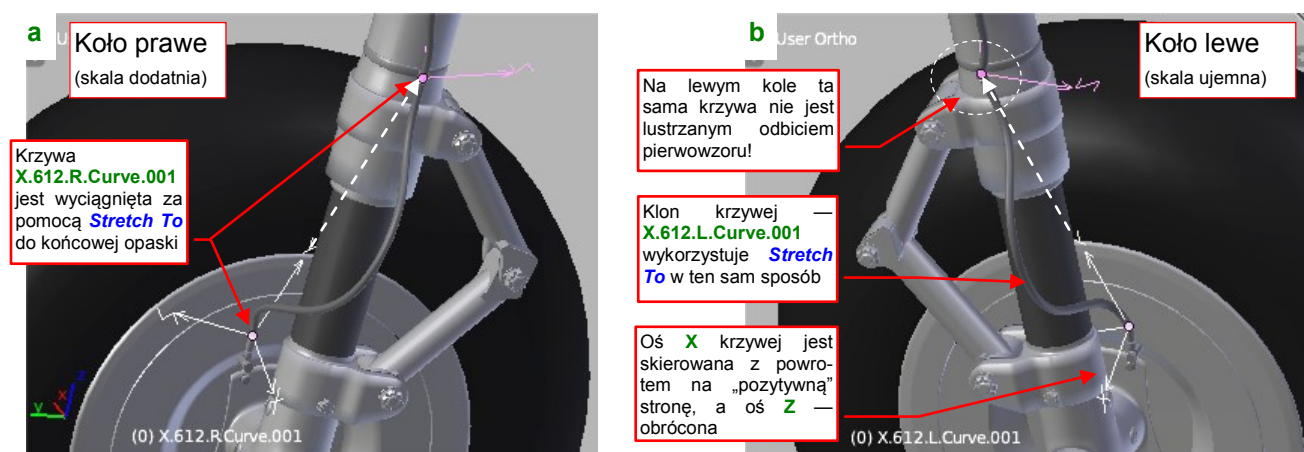
Ograniczenia **Stretch To** wymagają uwagi, gdy stosujesz je dla obiektów o ujemnej skali (także takiej „odziedziczonej” po obiekcie nadrzędnym). Pokażę to na przykładzie podwozia głównego (Rysunek 7.14.5):



Rysunek 7.14.5 Podwozie główne



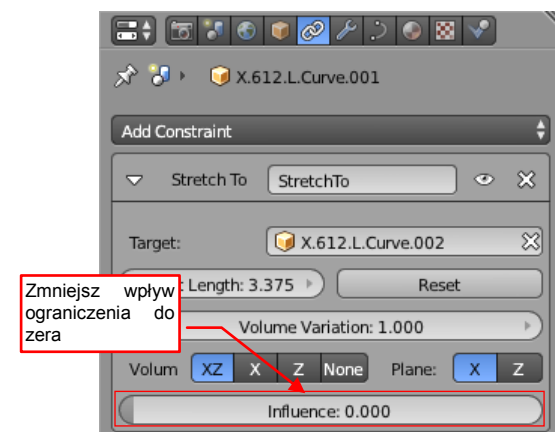
W zespołach kół głównych ograniczenia **Stretch To** są użyte do „wyciągnięcia” obiektów o niesymetrycznym kształcie: krzywych, wzdłuż których są wygięte przewody hydrauliczne (Rysunek 7.14.6a):



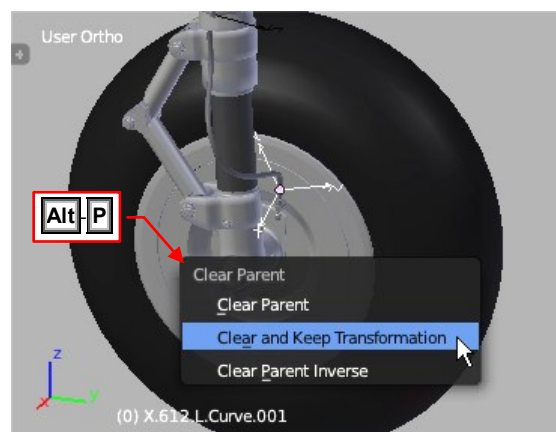
Rysunek 7.14.6 Wpływ ograniczenia **Stretch To** na „negatywne” kopie obiektów

Aby poprawnie wyglądać, wszystkie elementy lewego koła powinny mieć oś **X** przestawioną w przeciwną stronę niż te same elementy koła prawego. Wygląda jednak na to, że ograniczenie **Stretch To** przestawia z powrotem skalę obiektu na pozytywne (Rysunek 7.14.6b). Staje się to problemem dla obiektów których oś **Y** nie jest osią symetrii. W przypadku lewego koła takim przypadkiem są właśnie krzywe kabli z płynem hamulcowym. W wyniku tej zmiany przestały do siebie pasować w miejscach połączeń.

Cóż, w takim przypadku pozostaje się po prostu do tego stanu rzeczy dopasować, poprzez odpowiednią reorientację obiektu i jego krzywej. Pokażę to na przykładzie linii wygięcia dolnej części przewodu — korekty kształtu **X.612.L.Curve.001**. Zaczniemy od wyłączenia jej ograniczenia **Stretch To** (zmień jego wartość **Influence** od 0 — Rysunek 7.14.7):

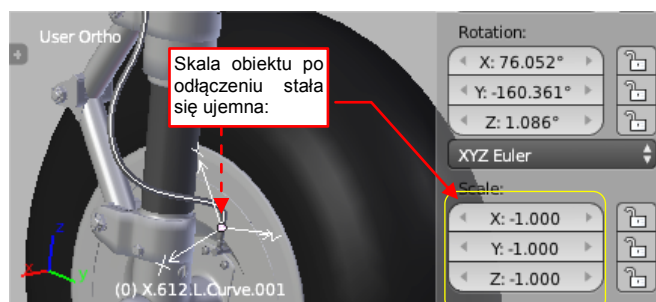


Rysunek 7.14.7 Wyłączenie ograniczenia **Stretch To**



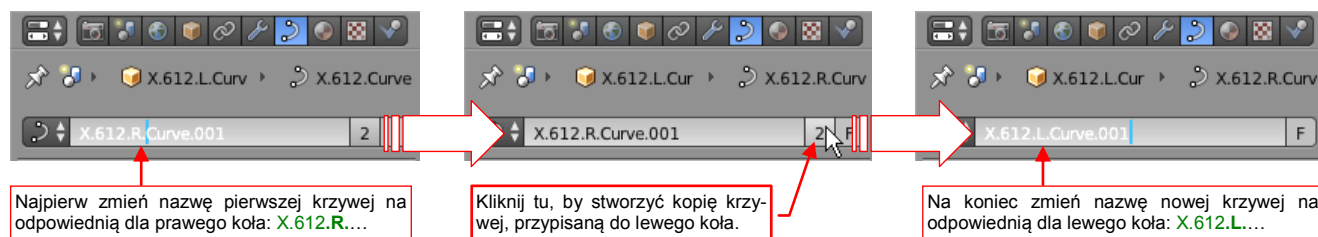
Rysunek 7.14.8 Odlączenie obiektu od zespołu lewego koła

Potem odłącz krzywą od zespołu koła (**Alt+P**, **Clear and Keep Transformation** — Rysunek 7.14.8). W rezultacie skala obiektu „urealni się”, i stanie się ujemna (Rysunek 7.14.9). Teraz powinniśmy ją „utrwalić” poleceniem **Apply→Scale**. To polecenie nie działa jednak na klony. Chodzi o to, że wszystkie elementy koła lewego używają tych samych siatek i krzywych, co koło prawe.



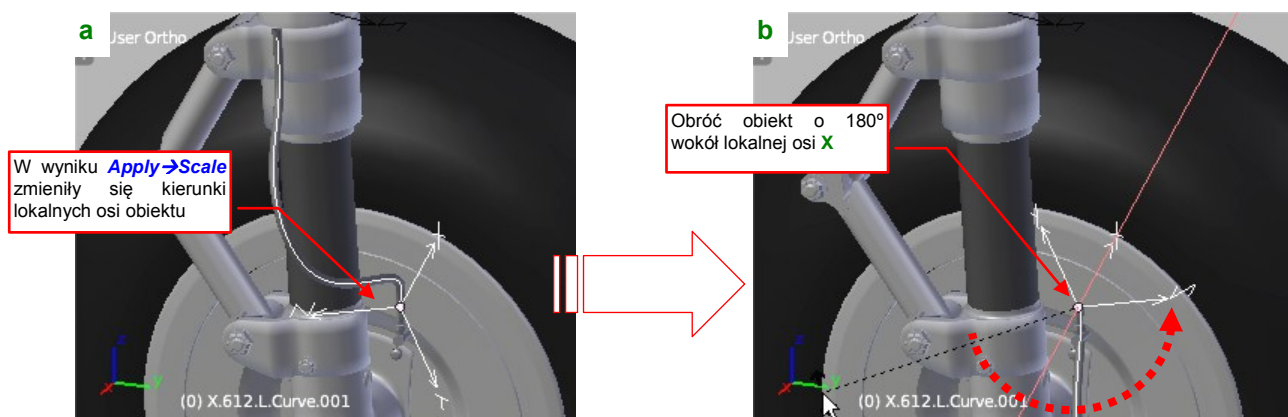
Rysunek 7.14.9 Transformacja krzywej po odlączeniu od koła

W naszym przypadku obiekt **X.612.L.Curve.001** używa krzywej **X.612.Curve.001**. Ta sama krzywa jest używana w prawym kole przez obiekt **X.612.R.Curve.001**. Musimy więc je powielić, tworząc lokalną wersję dla każdego obiektu (Rysunek 7.14.10):



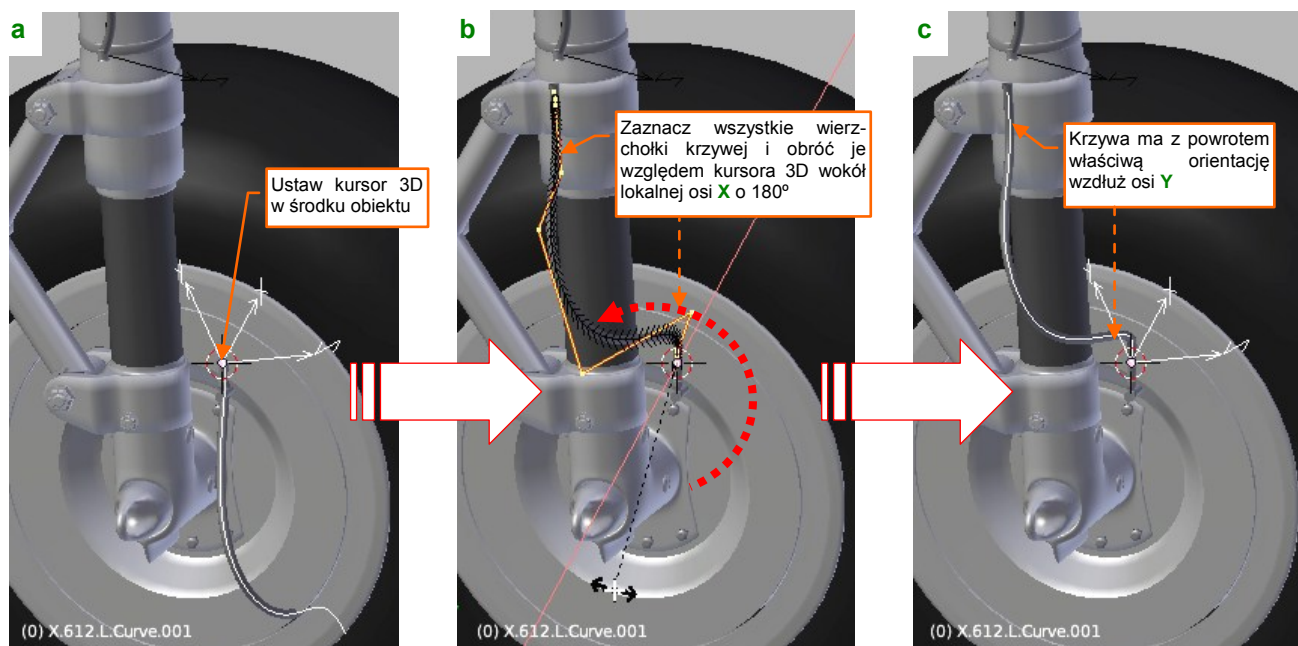
Rysunek 7.14.10 Powielenie krzywych przypisanych do dwóch obiektów

Teraz możesz już wywołać polecenie **Object→Apply→Scale**. Spowoduje to zmianę kierunku lokalnych osi obiektu (Rysunek 7.14.11a):



Rysunek 7.14.11 Obrót obiektu krzywej

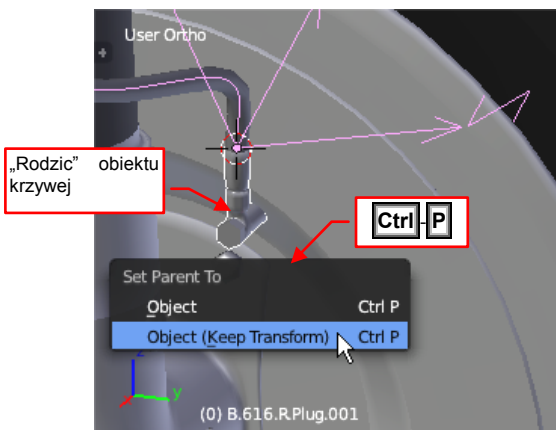
Ograniczenie **Stretch To** działa zawsze wzdłuż osi **Y**, stąd aby z powrotem skierować ją do góry, musimy obrócić obiekt wokół lokalnej osi **X** o 180° (Rysunek 7.14.11b). Aby taki obrót skompensować, przejdź do **Edit Mode** i obróć z kolei krzywą przypisaną do obiektu o kolejne 180° (Rysunek 7.14.12):



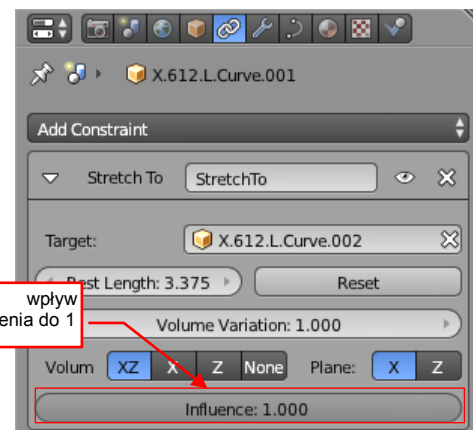
Rysunek 7.14.12 Obrót punktów krzywej

Pamiętaj tylko, aby wcześniej ustawić we właściwym miejscu kursor 3D (por. Rysunek 7.14.12a). (Najszybciej to zrobić otwierając menu **Snap** **Shift-S** i wybierając z niego polecenie **Cursor to Selected**).

Po tej modyfikacji możesz z powrotem przypisać obiekt krzywej w odpowiednie miejsce hierarchii lewego koła (Rysunek 7.14.13):

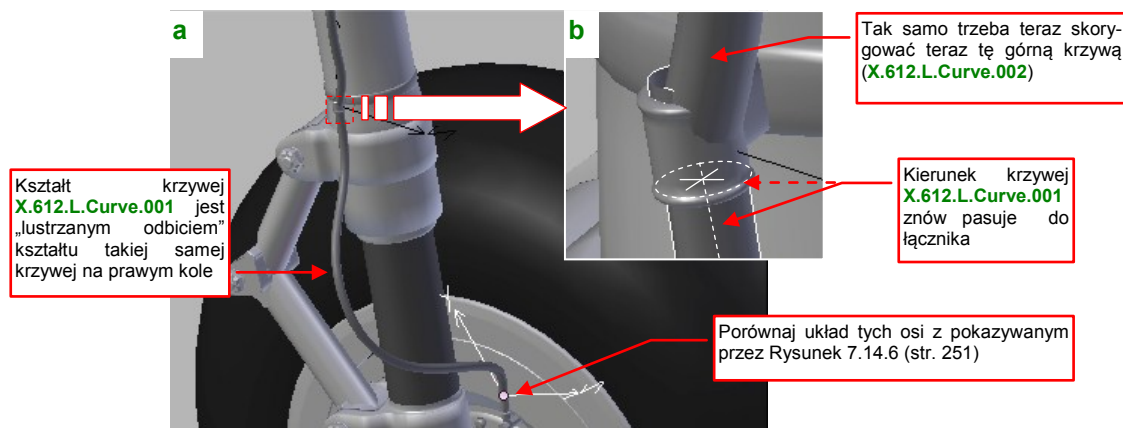


Rysunek 7.14.13 Przyłączenie obiektu od zespołu lewego koła



Rysunek 7.14.14 Włączenie ograniczenia *Stretch To*

Na koniec włącz z powrotem działanie modyfikatora *Stretch To* (Rysunek 7.14.14). Uzyskaliśmy w ten sposób to, o co chodziło: przewód płynu hamulcowego ma taki sam kształt (i tak samo się wygina) na obydwu kołach podwozia samolotu (Rysunek 7.14.15a):



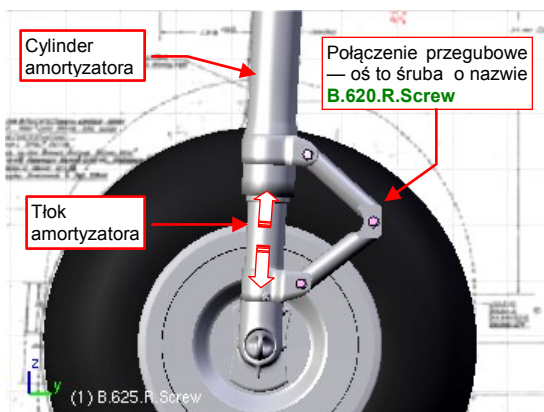
Rysunek 7.14.15 Skorygowana krzywa X.612.L.Curve.001

Pozostaje teraz powtórzyć te operacje dla kolejnego segmentu tej krzywej (por. Rysunek 7.14.15b)

### 7.15 Przypisanie ograniczenia *Limit Distance*

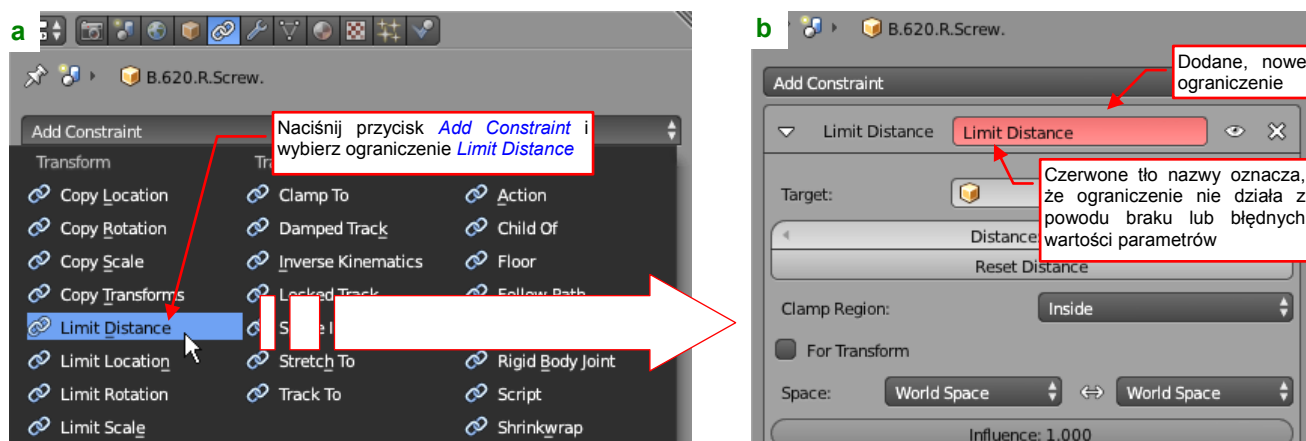
Ograniczenie *Limit Distance* wymusza na obiekcie zachowanie odpowiedniej odległości od środka obiektu — celu. Ta właściwość przyda się do zamodelowania działania połączenia przegubowego, takiego jak u tych dwóch łączników na amortyzatorze goleni podwozia (Rysunek 7.15.1). To typowe rozwiązanie, stosowane w konstrukcji samolotów od początku lat czterdziestych: dwa składające się „jak nożyce” pręty, zapobiegające przypadkowemu obrotowi koła wokół goleni.

Wydaje się, że najłatwiej odwzorować ten mechanizm za pomocą wewnętrznego szkieletu (*armatury* — por. str. 229). Można to także zrobić w inny sposób. Wystarczy użyć ograniczeń *Limit Distance* i *Stretch To*.



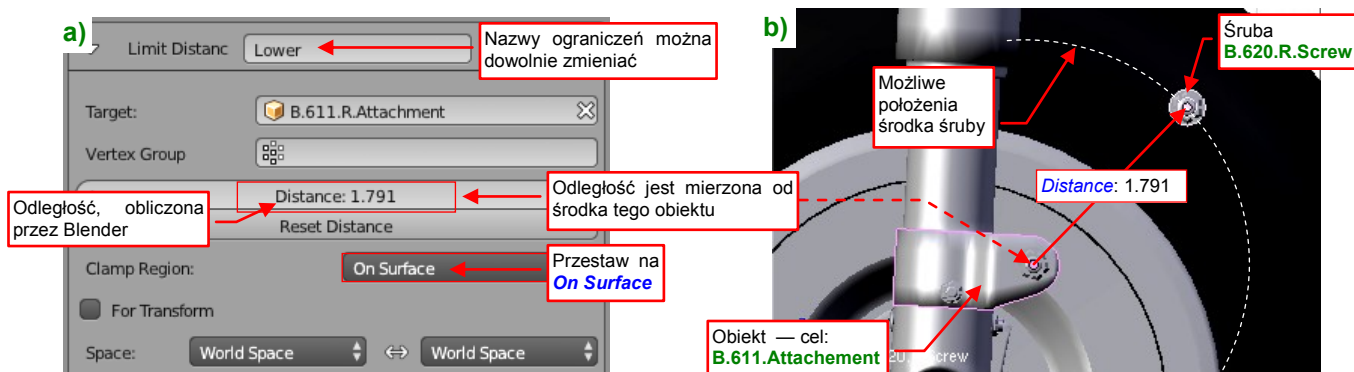
Rysunek 7.15.1 Połączenie przegubowe

Zacznijmy od dodania ograniczenia *Limit Distance*, do osi przegubu. W naszym przykładzie to śruba **B.620.R.Screw** (por. Rysunek 7.15.1). Zaznacz ją, a potem w zestawie *Constraints* z listy *Add Constraint* wybierz *Limit Distance* (Rysunek 7.15.2a):



Rysunek 7.15.2 Dodanie do obiektu ograniczenia *Limit Distance*.

W pierwszej chwili pole *Limit Distance* w panelu ograniczenia, (to z nazwą), jest czerwone (Rysunek 7.15.2b). To oznacza, że coś z parametrami tego ograniczenia jest nie tak. Po prostu w *Target* nie wskazaliśmy jeszcze obiektu — „celu”. Wpisz tam nazwę okucia (**B.611.Attachment**), a czerwony kolor zniknie z panelu (Rysunek 7.15.3a):

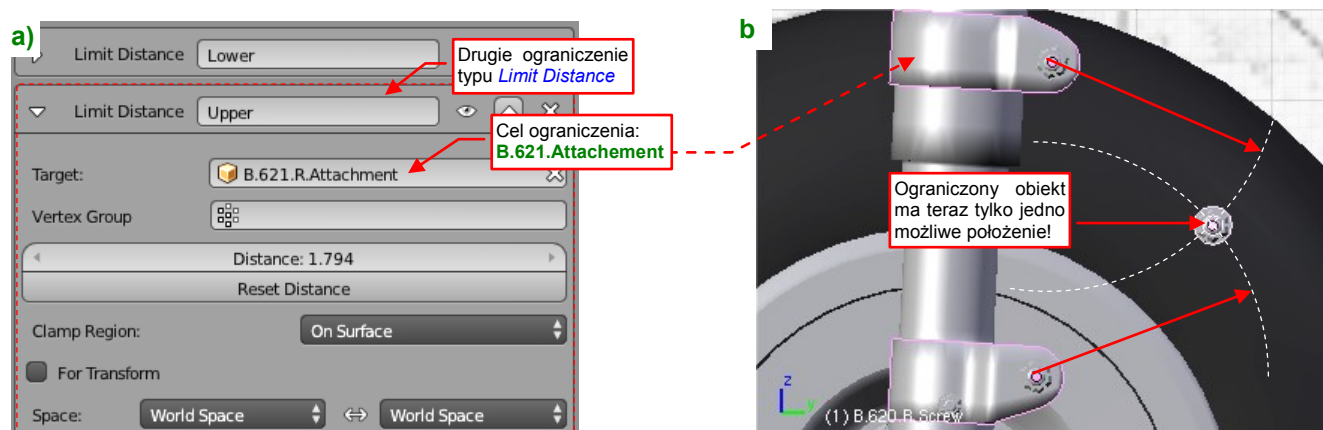


Rysunek 7.15.3 Przykład przypisania parametrów *Limit Distance*.

Jednocześnie przestaw *Clamp Region* na *On Surface*. To wymusza na śrubie zachowanie stałej odległości od środka okucia (Rysunek 7.15.3b). (Zauważ, że środek okucia nie musi być jego środkiem geometrycznym).



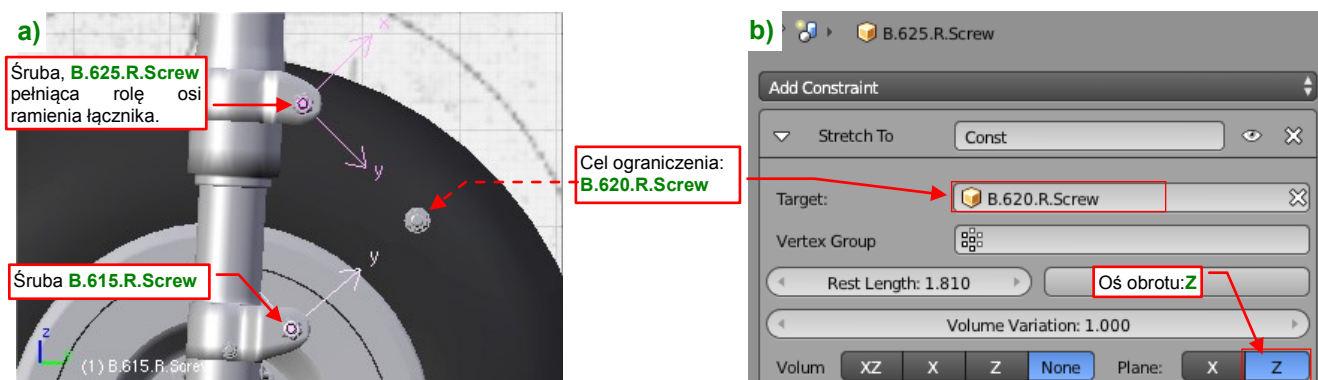
Z jednego ograniczenia *Limit Distance* nie ma jeszcze efektu, o który chodzi. Dodaj jednak do tej samej śruby (**B.620.R.Screw**) drugie ograniczenie tego samego typu — tyle że tym razem względem górnego okucia (**B.621.Attachment**) (Rysunek 7.15.4a):



Rysunek 7.15.4 Przykład zastosowania dwóch ograniczeń *Limit Distance*.

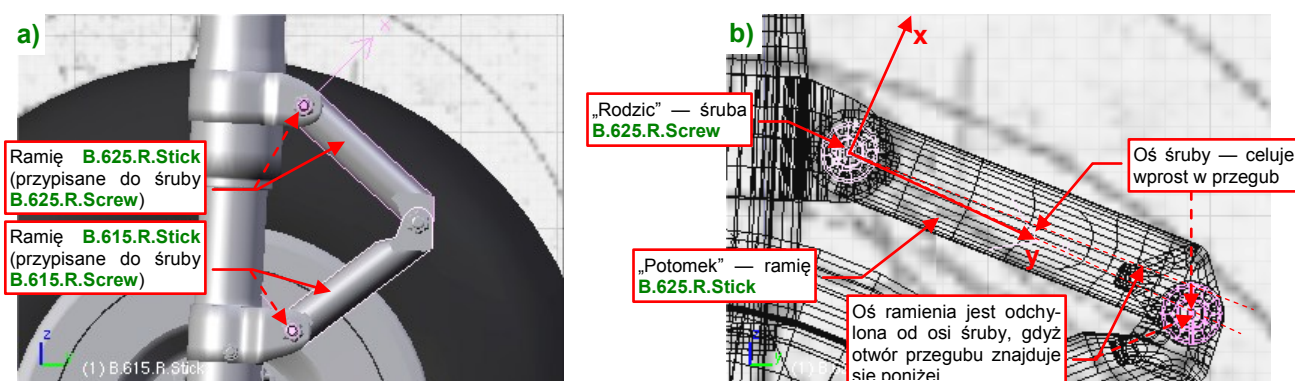
Rysunek 7.15.4b) pokazuje, że środek ograniczonej w ten sposób śruby może leżeć tylko w jednym, jedynym miejscu: tam, gdzie odległości od obydwu okuc są takie same. Ten punkt zmienia się, gdy okucia zbliżą się do siebie lub oddalą, w wyniku ruchu amortyzatora. W ten sposób ograniczenia *Limit Distance* zastępują ramiona łączników.

Teraz pozostaje tylko przypisać śrubom w osiach obrotu ramion ograniczenia *Stretch To* (por. str. 249) — tak, by ich lokalne osie **Y** „celowały” w oś przegubu (Rysunek 7.15.5a):



Rysunek 7.15.5 Przypisanie osiom łączników ograniczeń *Stretch To*.

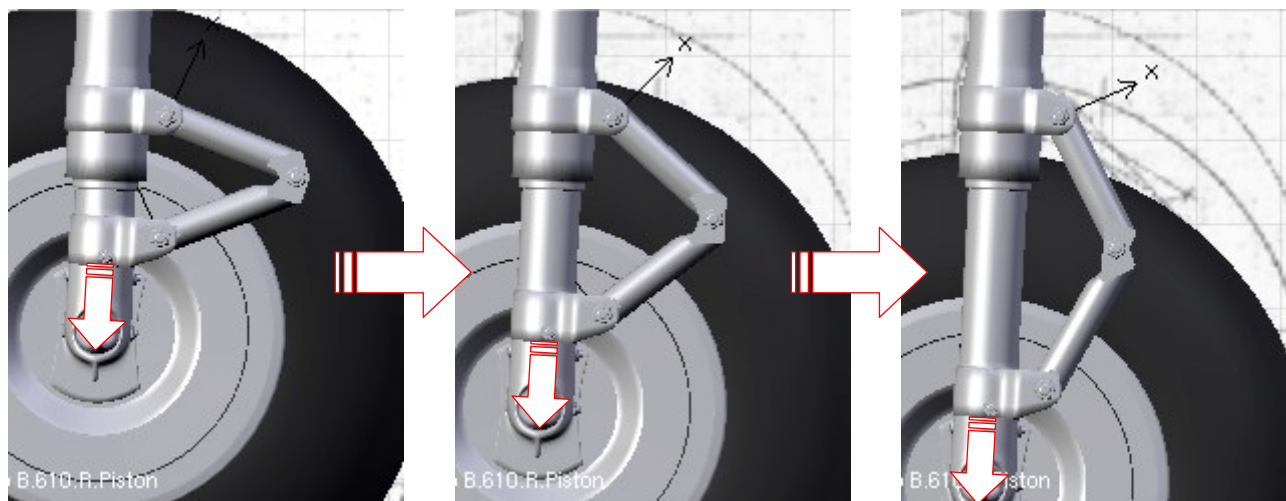
(Rysunek 7.15.5b) pokazuje parametry jednego z tych ograniczeń — jako *Target* jest wpisana śruba **B.620.R.Screw**, a osią obrotu jest lokalna oś **Z**. Potem pozostaje tylko przypisać do osi same ramiona (relacją *Parent*), by same się kierowały na oś przegubu (Rysunek 7.15.6a):



Rysunek 7.15.6 Przypisanie łączników do osi obrotu.

Zwróć uwagę, że ramiona są lekko odchylone od osi **Y** ich „rodziców” — śrub (Rysunek 7.15.6b).

Rysunek 7.15.7 przedstawia mechanizm, który stworzyliśmy, w działaniu:



Rysunek 7.15.7 Działanie mechanizmu połączenia przegubowego.

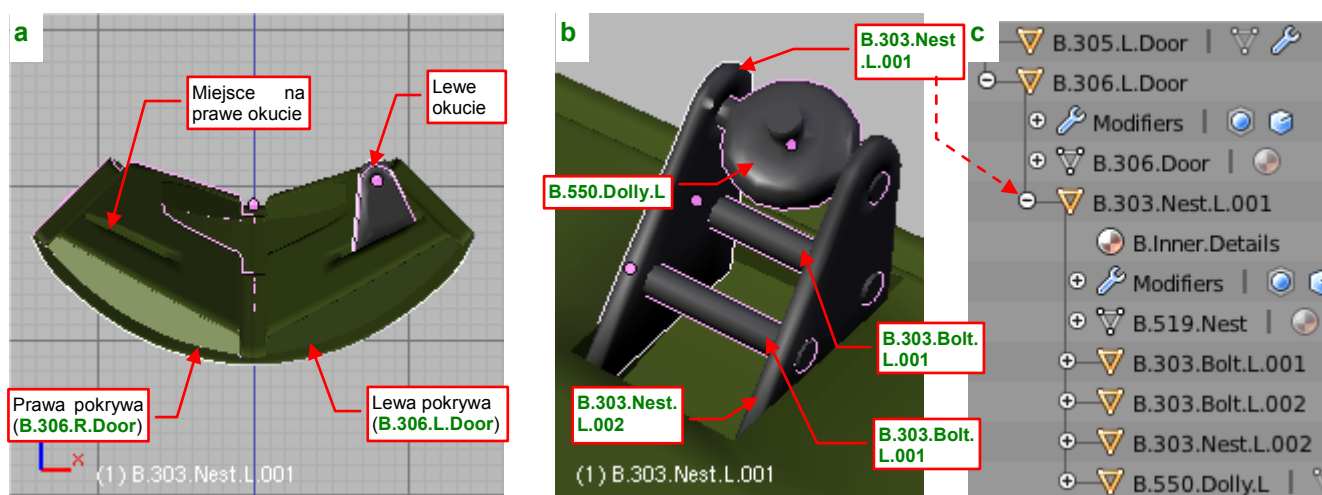
Wygłada to na poprawnie zginające się połączenie przegubowe. Na pierwszy rzut oka efektu nie można odróżnić od klasycznego zastosowania armatury (por. str. 229 i następne). Na koniec tej sekcji warto odnotować, że:

- Jako cel ograniczeń *Limit Distance* można wskazać wprost śruby w osiach obrotu ramion łącznika (chodzi o obiekty **B.625.R.Screw**, **B.615.R.Screw** — por. str. 255, Rysunek 7.15.5). W ten sposób mechanizm byłby złożony z mniejszej liczby obiektów, czyli — prostszy. Nie zrobiłem tego jednak ze względów „porządkowych”. (Blender wtedy wyświetla od czasu do czasu w konsoli ostrzeżenia o wzajemnej referencji — *circular reference* — tych dwóch śrub). Zamiast tego użyłem ich „rodziców” — okuć, którym specjalnie ustawiłem środki w osiach obrotu ramion.
- Gdyby otwory na oś przegubu (obiekt **B.620.R.Screw**) leżały na osi ramiona łączników (**B.625.R.Stick**, **B.615.R.Stick**) — ograniczenia *Stretch To* zostałyby przypisane bezpośrednio do nich. Wtedy ramiona mogłyby być bezpośrednimi „potomkami” okuć, a śruby, leżące w ich osiach obrotu, nie byłyby w mechanizmie potrzebne. Zastosowałem je wyłącznie dlatego, że oś przegubu nie leży na osi podłużnej ramienia łącznika, więc nie można jej wycelować prosto w jego środek (por. str. 255, Rysunek 7.15.6b).
- Teoretycznie, do skierowania ramion na oś przegubu, zamiast ograniczeń *Stretch To* można by było zastosować ograniczenia *Locked Track* (por. str. 237). Jednak wtedy końcówki ramion mogłyby w pewnych położeniach „odstawać” od przegubu. Wygłada na to, że Blender wyznacza położenie wynikające z ograniczeń *Limit Distance* w sposób przybliżony (iteracja?), co prowadzi do powstania takich odstępów. Jedynym lekarstwem na to są ograniczenia *Stretch To*, które same dbają o to, by końcówki ramion były „dociągnięte” do zadanego punktu. (Niewielkie zmiany skali wzdłuż lokalnych osi **Y** śrub są przenoszone na obiekty potomne — ramiona. I to wystarcza).

### 7.16 Odbicie lustrzane (*Mirror*)

Transformacja *Mirror* jest właściwie skrótem — specyficzną formą zmiany skali obiektu poprzez pomnożenie jej wartości względem jednej z lokalnych osi przez  $-1$ . Można ją wywołać skrótem **Ctrl-M** lub poleceniem *Object → Mirror*. (Ten sam rezultat można uzyskać poprzez odpowiednią zmianę skali w jednym kierunku). W tej sekcji pokażę praktyczny przypadek zastosowania lustrzanego odbicia (*Mirror*).

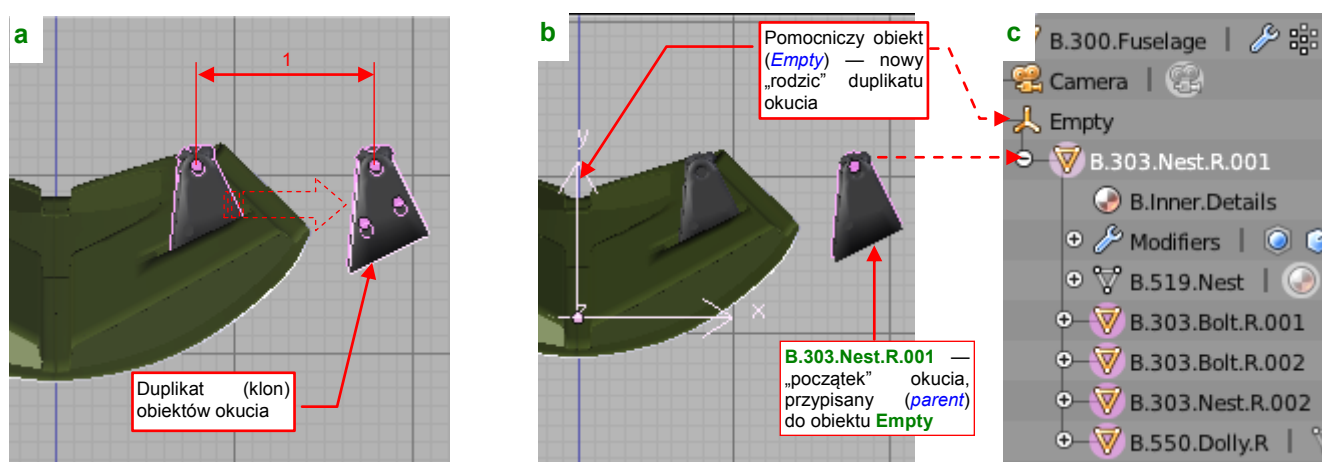
Naszym zadaniem jest skopiowanie i przypisanie do prawej pokrywy luku podwozia okucia, do którego ma być potem przyczepiony popychacz (Rysunek 7.16.1a):



Rysunek 7.16.1 Okucie pokrywy podwozia i jego struktura

Okucie składa się z dwóch trójkątnych blach, połączonych bolcami (nazwy poszczególnych części podaje Rysunek 7.16.1b). W hierarchii modelu (Rysunek 7.16.1c) jedna z tych blach (**B.303.Nest.L.001**) jest „rodzicem” (*parent*) pozostałych elementów okucia. Jednocześnie jej „rodzicem” jest lewa pokrywa podwozia (**B.306.L.Door**).

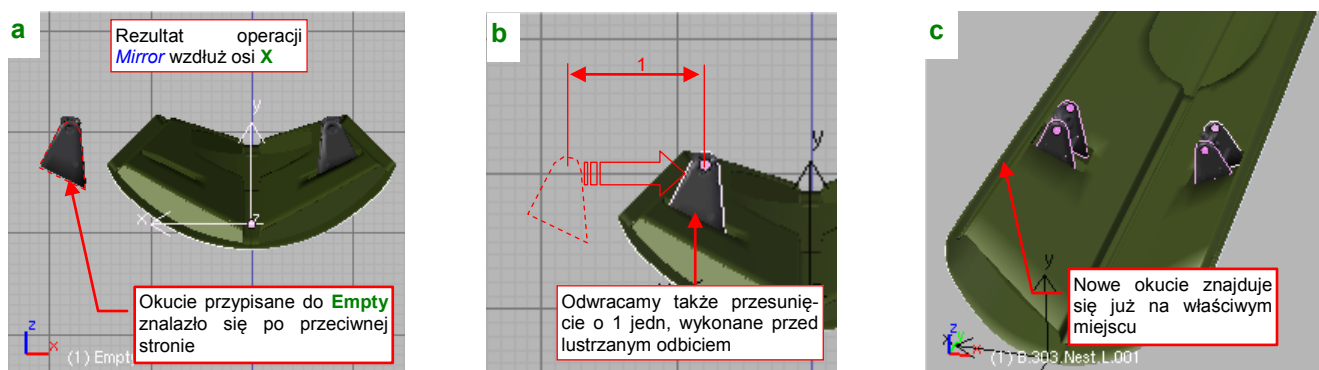
Zaznacz wszystkie elementy okucia i zduplikuj je (**Alt-D**, str. 219). Zaraz po stworzeniu przesun te duplikaty w bok, na jakiś „równy” i łatwy do zapamiętania dystans — np. o 1 jednostkę Blendera (Rysunek 7.16.2a). (Robimy to wyłącznie po to, by łatwo było je zaznaczać). Przy okazji zmień w nazwach nowo utworzonych obiektów końcówki **L** na **R** (choć jeszcze na to nie wygląda, będzie to okucie prawej pokrywy):



Rysunek 7.16.2 Powiązanie duplikatu okucia z pomocniczym obiektem *Empty*

Aby w wyniku operacji *Mirror* okucie nie odwróciło się „do góry nogami”, musimy zastosować pewne obejście. Wstaw na chwilę w osi samolotu pomocniczy obiekt *Empty* (jak — str. 218) i uczyni go rodzicem blaszki **B.303.Nest.R.001** (Rysunek 7.16.2c).

Teraz można wywołać transformację **Mirror** (**Ctrl-M**) dla obiektu **Empty**. To w istocie zmiana skali wzdłuż osi **X** z 1.0 na -1.0. „Rodzic” okucia — blaszka **B.303.Nest.R.001** — jest jednocześnie „dzieckiem” **Empty** (Rysunek 7.16.2c). Zmiana skali „rodzica” przekłada się na wszystkie obiekty do niego przypisane, stąd okucie znalazło się już po przeciwległej stronie osi samolotu (Rysunek 7.16.3a):

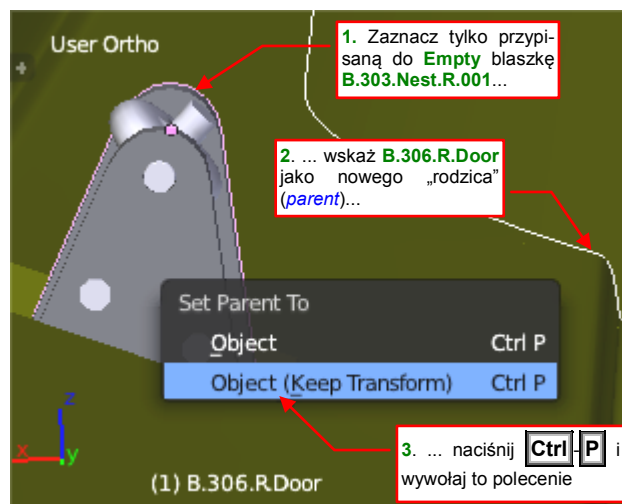


Rysunek 7.16.3 Przeniesienie duplikatu okucia ponad drugą pokrywę (**Mirror**)

Przesuń jeszcze okucie wzdłuż osi **X** o 1 jedn. (por. str. 257, Rysunek 7.16.2a) — tak, jak to pokazuje Rysunek 7.16.3b). W rezultacie mamy je już w miejscu docelowym (Rysunek 7.16.3c).

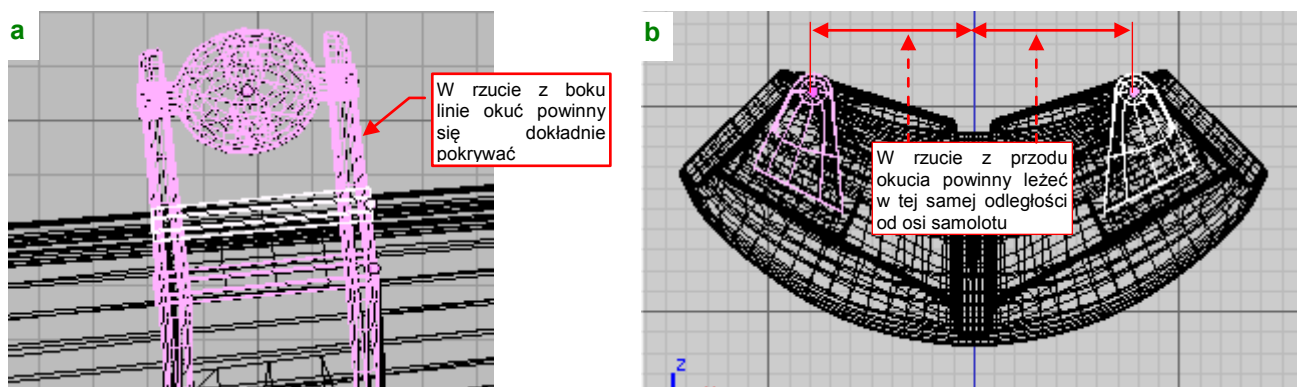
Na koniec zamieniamy „rodzica” okucia (a konkretnie: jego blaszki **B.303.Nest.R.001**) z tymczasowego obiektu **Empty** na pokrywę podwozia (**B.306.R.Door**) (Rysunek 7.16.4). Pamiętaj: zawsze w takich sytuacjach używaj polecenia **Set Parent To → Object (Keep Transform)**, aby pozostał na swoim miejscu!

Potem możesz już usunąć niepotrzebny obiekt **Empty**.



Rysunek 7.16.4 Zmiana przypisania rodzica

Gdy wykonasz wszystkie te operacji warto jeszcze przełączyć się na tryb wyświetlania **Wireframe** (**Z**), i sprawdzić, czy w rzucie z boku linie obydwu okuc dokładnie się pokrywają (Rysunek 7.16.5a):



Rysunek 7.16.5 Weryfikacja poprawności — w rzucie z boku i z przodu

Nie zaszkodzi zerknąć także na rzut z przodu (Rysunek 7.16.5b), czy są w równych odległościach od osi samolotu.



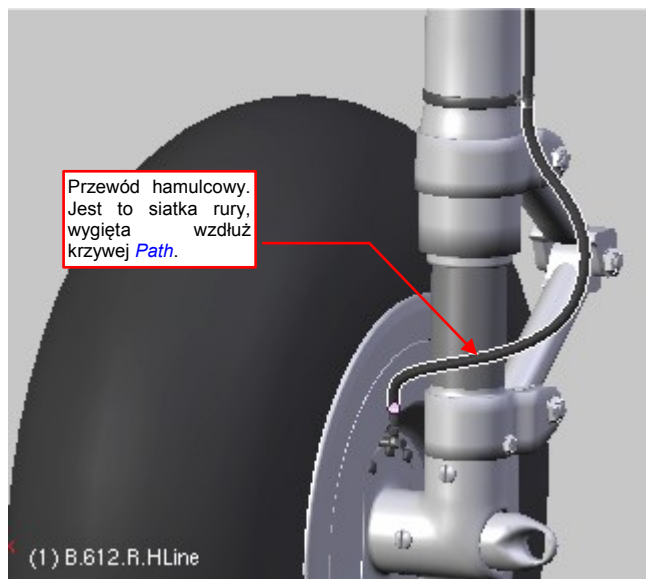
Rysunek 7.16.6 przedstawia ostateczny rezultat — prawidłowo ustawione okucia, przypisane do odpowiednich pokryw luku podwozia:



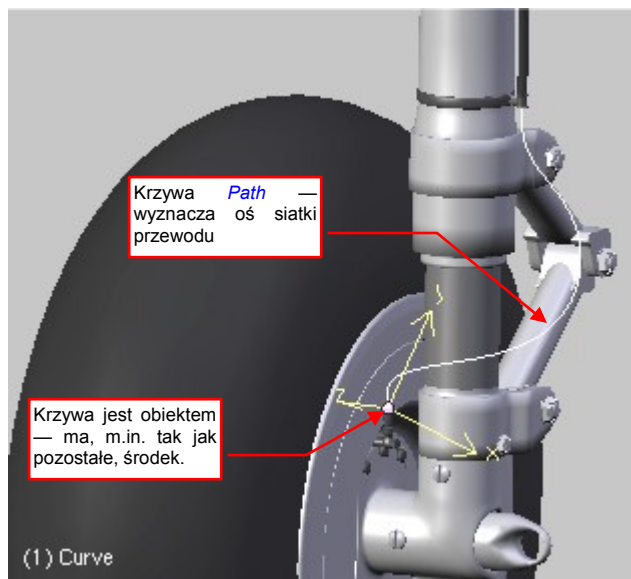
Rysunek 7.16.6 Ostateczny rezultat

### 7.17 Wstawienie krzywej (*Curve*)

Zastosowanie dla krzywych znajduję w Blenderze stosunkowo rzadko: gdy trzeba zamodelować jakiś wygięty kabel czy rurkę. Przykładem takiego elementu jest przewód z płynem hamulcowym, doprowadzanym do piasty koła głównego (Rysunek 7.17.1):



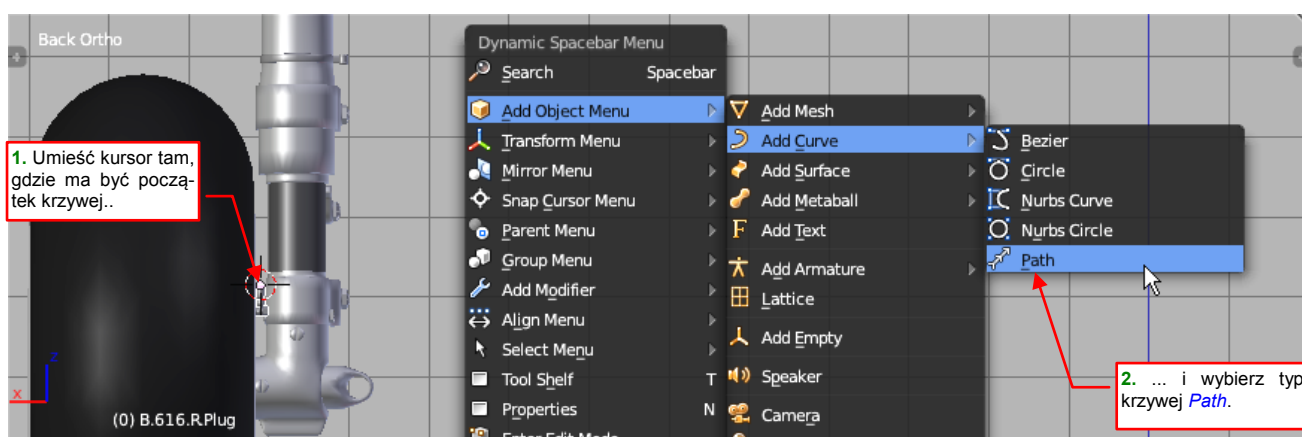
Rysunek 7.17.1 Przykład siatki, wygiętej wzdłuż krzywej



Rysunek 7.17.2 Krzywa, determinująca kształt siatki

Przewód, który pokazuje Rysunek 7.17.1, został uformowany jako prosta „rurka”. Następnie wygiąłem go, za pomocą odpowiedniego modyfikatora *Curve* (por. str. 288) wzdłuż odpowiedniej krzywej przestrzennej (Rysunek 7.17.2). To jeden z typów linii dostępnych w Blenderze. Można byłoby tę siatkę „powyginać” bez pomocy krzywej, ale tak jest wygodniej modyfikować jej kształt. W dodatku mamy gwarancję, że przekrój poprzeczny tak wygiętej „rury” nie będzie w żadnym miejscu zdeformowany.

Aby dodać do rysunku krzywą (ścieżkę), umieść kursor 3D w miejscu, gdzie ma być jej środek (tzn. punkt odniesienia obiektu, który będzie ją zawierał). Następnie naciśnij **Space**<sup>1</sup> (lub rozwiń menu *Add* — jest zaraz obok menu *File*). Wywołaj polecenie *Add* → *Add Curve* → *Path* (Rysunek 7.17.3):

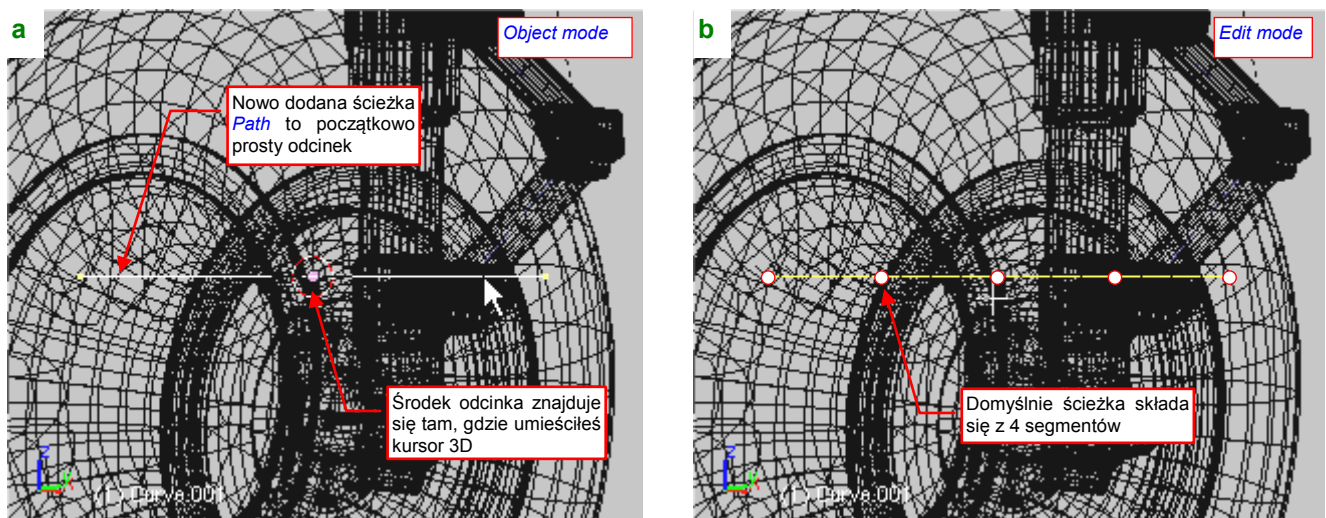


Rysunek 7.17.3 Tworzenie krzywej typu *Path*.

Zwróć uwagę, że masz do wyboru w Blenderze aż pięć rodzajów krzywych. Ja zwyczaj stosuję krzywą typu *Path*. To przestrzenna krzywa NURBS która zachowuje ciągłość promienia krzywizny.

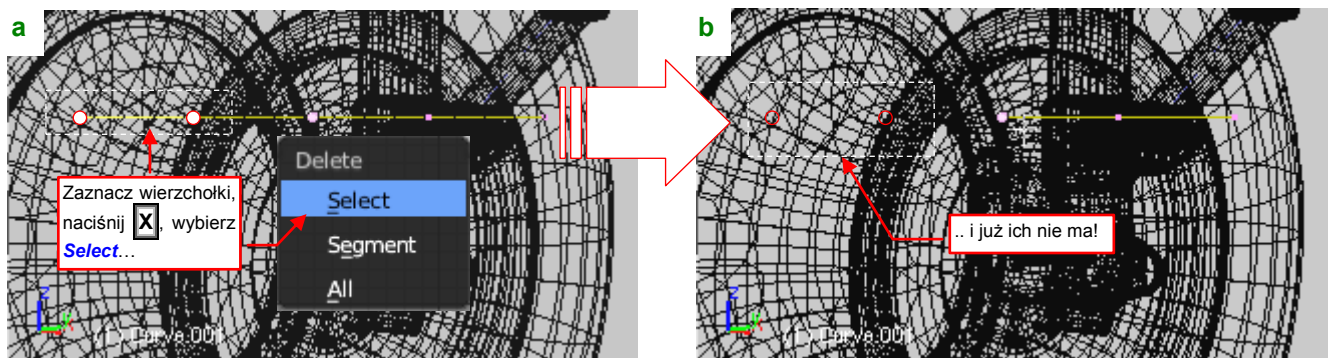
<sup>1</sup> To menu będzie się pojawiać, jeżeli włączysz w konfiguracji Blendera dodatek *Dynamic Spacebar Menu*. Ten dodatek jest dostarczany wraz z Blenderem, wystarczy go uaktywnić — tak, jak to pokazuje Rysunek 6.1.7 na str. 749

Blender zawsze wstawia obiekt typu *Path* jako zupełnie prosty, poziomy odcinek. Jego geometryczny środek znajduje się tam, gdzie kursor 3D (Rysunek 7.17.4a):



Rysunek 7.17.4 Nowo utworzona ścieżka (krzywa) *Path* — jako obiekt i w trybie edycji

Rysunek 7.17.4 pokazuje obiekty w trybie wyświetlania *Wireframe*, gdyż część odcinka przenika do wnętrza innych powłok. Krzywą można zmieniać tak samo, jak zmienia się siatki. Wystarczy przełączyć się w tryb edycji (**Tab**). Zaraz po utworzeniu ścieżka składa się z czterech równych segmentów, łączących pięć wierzchołków (Rysunek 7.17.4b). Aby usunąć niepotrzebne wierzchołki, wystarczy je zaznaczyć i nacisnąć **X** (*Curve → Delete*) (Rysunek 7.17.5):



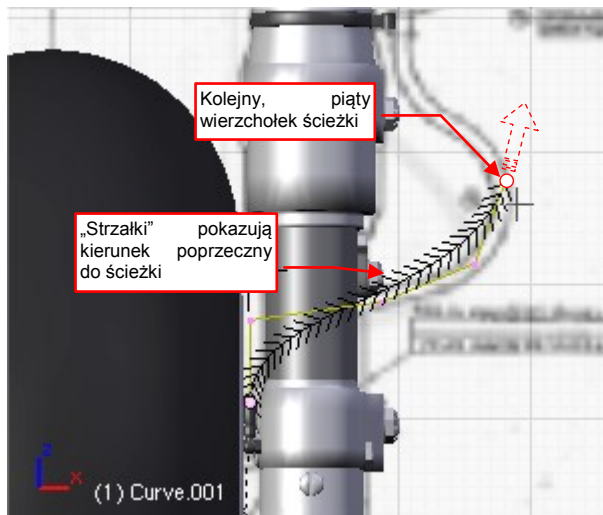
Rysunek 7.17.5 Usuwanie wierzchołków ze ścieżki

W menu, które się pojawi (Rysunek 7.17.5a), wystarczy wybrać opcję *Select*. Wierzchołki ścieżki możesz obracać (**R**), przesuwać (**G**), czy wyłaczać (**E**) tak samo, jak wierzchołki siatki (Rysunek 7.17.6):

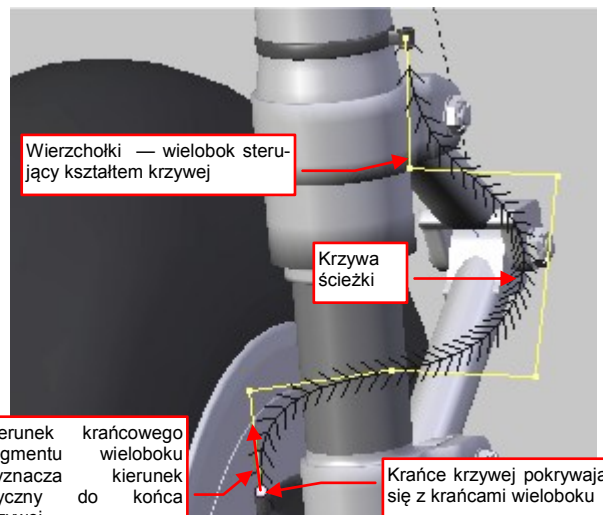


Rysunek 7.17.6 Dalsza edycja ścieżki

Po wytłoczeniu piątego wierzchołka ścieżka staje się linią krzywą (Rysunek 7.17.7):



Rysunek 7.17.7 Wytłoczenie kolejnego wierzchołka

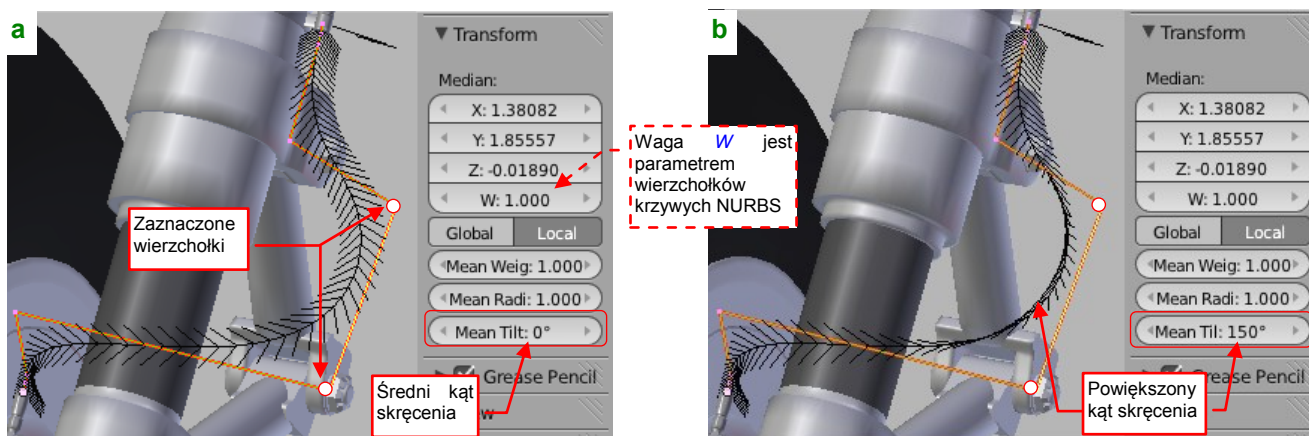


Rysunek 7.17.8 Uformowana krzywa przestrzenna

Na krzywej ścieżki Blender rysuje pomocnicze „strzałki”. Pokazuje w ten sposób kierunek poprzeczny w każdym punkcie tej linii. (Ta informacja jest używana np. do orientacji kamery, przesuwanej wzdłuż ścieżki, czy podczas deformacji siatki wzdłuż krzywej). Wierzchołki, którymi manipulujemy, to punkty sterujące kształtem krzywej. Tworzą tzw. wielobok sterujący. Odpowiednio dobierając jego kształt, można uzyskać krzywą przestrzenną, wyglądającą jak wygięty kawałek elastycznego przewodu (Rysunek 7.17.8).

- Końce krzywej **Path** znajdują się zawsze w krańcowych wierzchołkach wieloboku sterującego. Kierunki krańcowych segmentów wieloboku wyznaczają kierunki styczne na końcach krzywej (Rysunek 7.17.8).
- Minimalna liczba wierzchołków, przy której **Path** zaczyna być krzywą, wskazuje na to, że jest to tzw. krzywa czwartego stopnia (lub — używając innej terminologii — piątego rzędu). Jest to najmniejszy stopień krzywej przestrzennej, zapewniający ciągłość i „gładkie” zmiany promienia krzywizny wzdłuż jej całej długości.

Współrzędne zaznaczonego wierzchołka krzywej możesz znaleźć w panelu **Transform** z przybornika **Properties**. Gdy zaznaczysz więcej niż jeden wierzchołek Blender wyświetla w tym panelu wartości średnie. Zwróć uwagę że oprócz współrzędnych **X**, **Y**, **Z** jest tam podana także waga wierzchołka **W** (Rysunek 7.17.9):

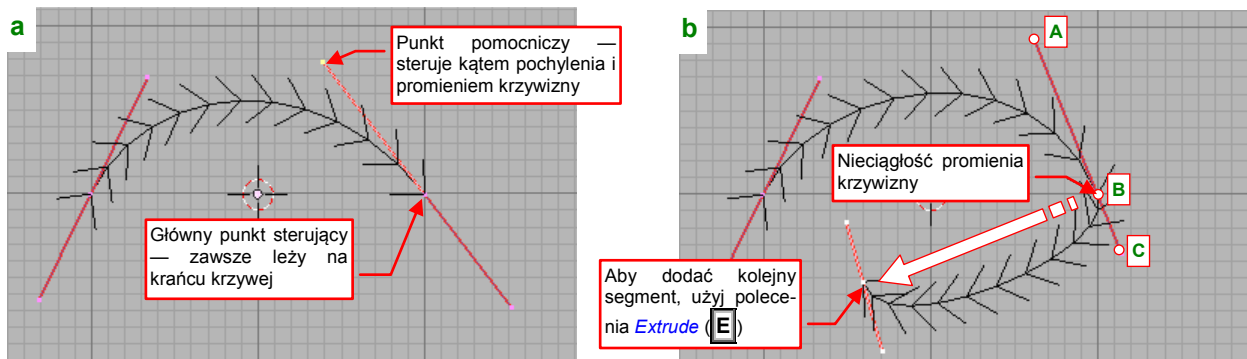


Rysunek 7.17.9 Sterowanie skreśnieniem ścieżki (**Tilt**)

Innym parametrem każdego wierzchołka sterującego jest kąt skręcenia **Tilt**. „Strzałki”, biegnące wzdłuż krzywej pokazują aktualną orientację w każdym jej punkcie. Porównaj Rysunek 7.17.9a) i Rysunek 7.17.9b). Na ilustracji b) środkowy segment tej krzywej uległ skręceniu o 150°.



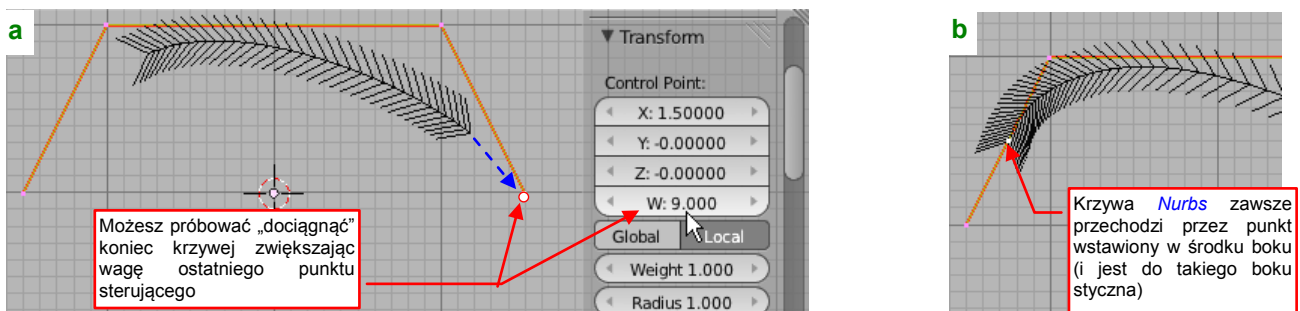
Gdy użyjesz polecenia **Add→Add Curve→Bezier**, stworzysz pojedynczy segment krzywej Beziera (4 rzędu). Mamy tu nieco inne punkty „uchwyty” punktów sterujących (Rysunek 7.17.10a):



Rysunek 7.17.10 Krzywa Beziera (4 rzędu)

Główne punkty sterujące leżą zawsze na końcach segmentu krzywej Beziera. Tylko w tych punktach można zmieniać skręcenie (*Tilt*) krzywej. Każdemu punktowi głównemu towarzyszą dwa punkty pomocnicze. Ich kierunek determinuje kąt pochylenia krzywej na końcach segmentu, a długość ich odcinków — lokalny promień krzywizny. Nowy segment krzywej Beziera możesz dodać tak samo jak w krzywej *Path* — poleceniem **Extrude** (E — Rysunek 7.17.10b). Pamiętaj że aby uniknąć skokowej zmiany promienia krzywizny na połączeniu segmentów krzywej Beziera, musisz zapewnić równą długość odcinków  $|AB|$  i  $|BC|$  (Rysunek 7.17.10b).

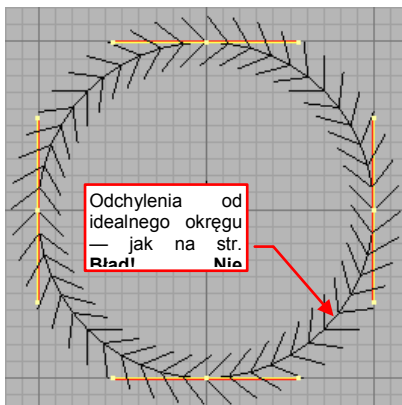
Innym rodzajem linii jest NURBS (**Add→Add Curve→Nurbs**). W odróżnieniu od *Path* jest to „klasyczna” krzywa, ze „swobodnymi” końcami (Rysunek 7.17.11a):



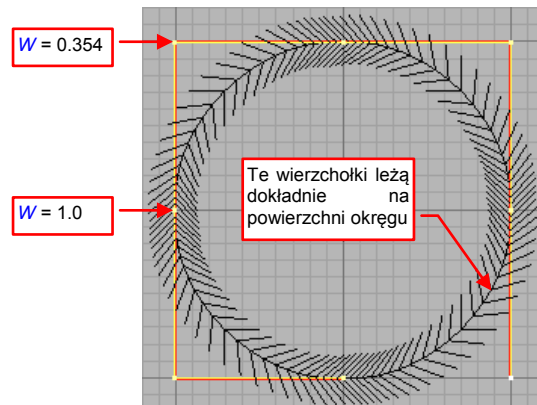
Rysunek 7.17.11 Krzywa NURBS (4 rzędu)

Możesz próbować „dociągnąć” koniec krzywej do ostatniego wierzchołka zwiększając jego wagę *W* (Rysunek 7.17.11a), ale lepszą metodą jest wstawienie w środek ostatniego segmentu dodatkowego punktu sterującego (Rysunek 7.17.11b).

Dla każdego typu krzywej istnieje polecenie tworzące krzywą zamkniętą (*Circle*):



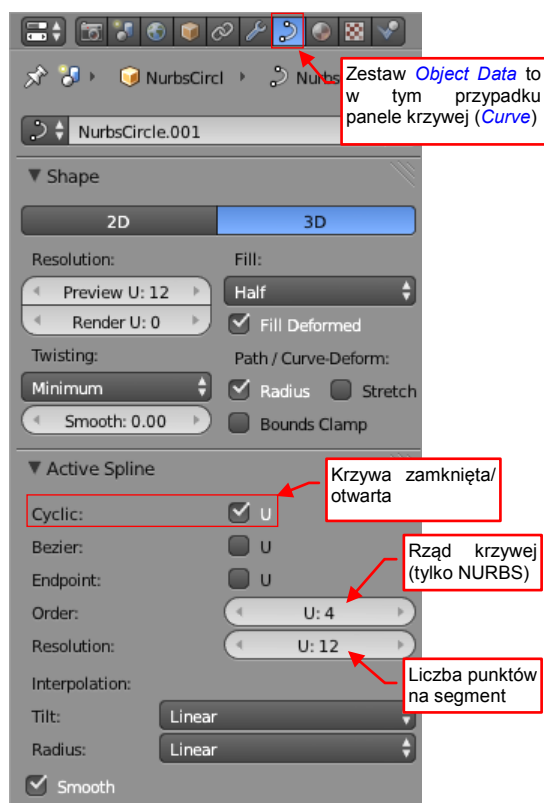
Rysunek 7.17.12 Okrąg Beziera (*Bezier Circle*)



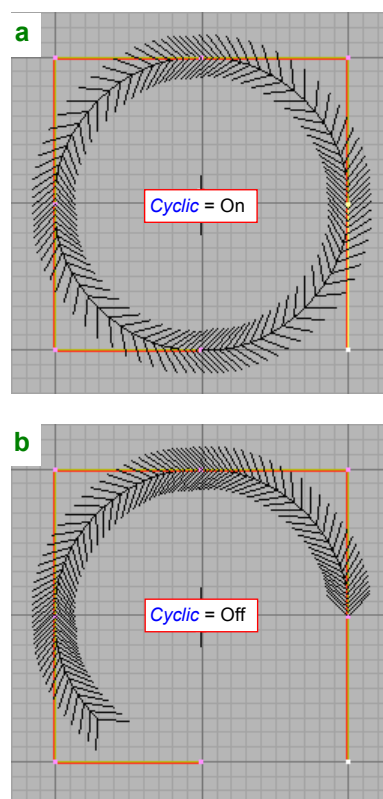
Rysunek 7.17.13 Okrąg NURBS (*Nurbs Circle*)

**Bezier Circle** (Rysunek 7.17.12) jest pewnym przybliżeniem okręgu, o dokładności zbliżonej do krzywej podziałowej uzyskanej z ośmioboku (por. Wyciąg II). Gdy potrzebujesz bardzo wysokiej dokładności — użyj **Nurbs Circle**, którego wierzchołki leżą dokładnie wzdłuż matematycznego okręgu (dzięki zmniejszonym wagom *W* narożników wieloboku sterującego — por. Rysunek 7.17.13). Inna sprawa, że wierzchołki **Nurbs Circle** nie są rozłożone równomiernie. (Zwróć uwagę na większą liczbę „strzałek” w pobliżu środków boków wielokąta sterującego).

Szczegółowe właściwości całej krzywej znajdziesz w zestawie **Object Data** okna **Properties** (Rysunek 7.17.15):

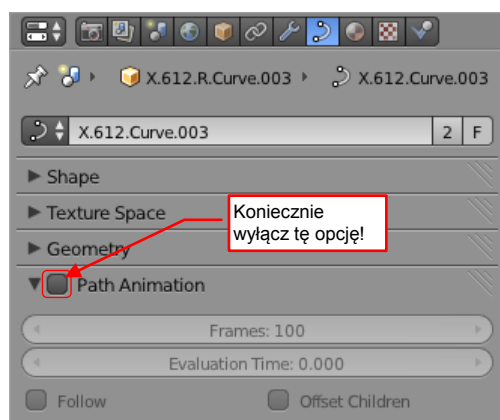


Rysunek 7.17.14 Panele właściwości krzywej



Rysunek 7.17.15 Krzywa zamknięta i otwarta (**Cyclic**)

W przypadku krzywej NURBS możesz tu zmienić m.in. rząd krzywej (**Active Spline: Order**) i liczbę wierzchołków na segment (**Resolution**). Innym ważnym przełącznikiem jest **Cyclic**. Ta flaga pozwala przełączać się pomiędzy krzywą zamkniętą i otwartą (Rysunek 7.17.15).



Rysunek 7.17.16 Wyłączenie animacji krzywej

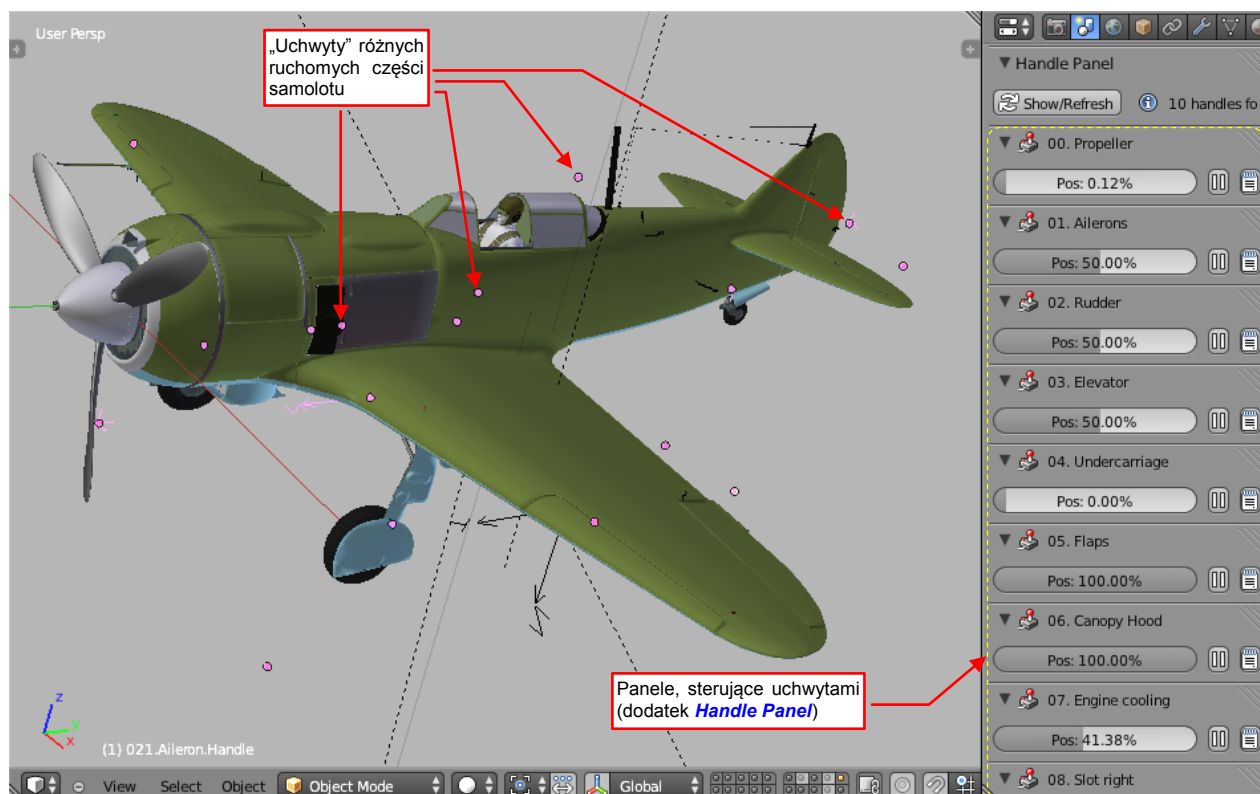
Pamiętaj, aby po stworzeniu krzywej zawsze wyłączać jej opcję **Path Animation** (Rysunek 7.17.16). Ta panel jest domyślnie włączona. Jeżeli o tym zapomnisz, w kolejnych klatkach animacji krzywa może zacząć się poruszać (a wraz z nią i siatki, których kształt modyfikuje). Lepiej to skorygować od razu.

Opis, jak zastosować krzywą do deformacji siatek (np. uzyskania wygiętego przewodu) znajdziesz w sekcji o modyfikatorze **Curve** (na str. 288 i następnych).


### 7.18 Panel sterowania ruchem obiektów (*HandlePanel*)

W ukończonym modelu znajdzie się około dziesięciu uchwytów (pomocniczych obiektów, służących do poruszania w określony sposób jakiejś części). Jeden służy do wysuwania i chowania podwozia, inny steruje obrotem steru kierunku, jeszcze inny — lotkami, kolejny — klapami, itd. Za jakiś czas, gdy zajrzysz do pliku z ukończonym modelem, możesz mieć problemy z szybkim zidentyfikowaniem, do czego służy konkretny element jego „chmurki” uchwytów. Odkrywanie na nowo ich funkcji „metodą prób i błędów” nie jest specjalnie zachęcającą perspektywą.

Dlatego opracowałem specjalny dodatek do Blendera. Jest to dodatek *Handle Panel*. Po uruchomieniu przeszukuje cały model i znajduje w nim wszystkie uchwyty. Dla każdego znalezionej uchwyty wyświetla panel z opisową nazwą i „suwakiem”, umożliwiającym wygodne określenie aktualnego położenia każdego z podzespołów samolotu (Rysunek 7.18.1):



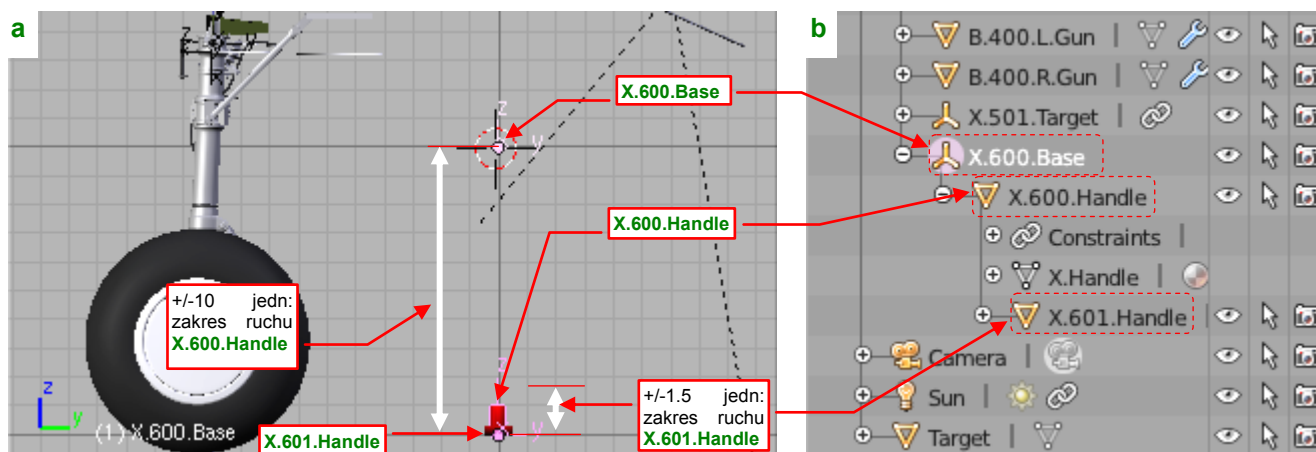
Rysunek 7.18.1 Panel sterowania zespołami samolotu

Nazwy, które widzisz na ilustracji (np. „06. Canopy Hood”), zostały przypisane poszczególnym uchwytom za pomocą przycisku , umieszczonego na każdej z paneli. Te teksty są przechowywane i zapisywane razem z całym modelem (jako dodatkowe dane poszczególnych obiektów).

- Skąd dodatek *Handle Panel* „wie”, co jest uchwytom, a co nim nie jest? To proste: ten program wyświetla panel dla każdego obiektu spełniającego następujące kryteria:
  - nazwa obiektu ma przyrostek *\*.Handle*, np. *X.600.Handle* (uważaj więc na takie nazwy);
  - obiekt ma ograniczenie *Limit Location*, w lokalnym układzie współrzędnych (*Convert:Local*);
  - obiekt jest przyporządkowany (relacją *Parent*) do jakiegoś innego obiektu („rodzica”);

Program zakłada, że środek (tzn. punkt odniesienia — *origin*) „rodzica” uchwyty wyznacza jego położenie neutralne. Odpowiada ono jakiejś szczególnej pozycji mechanizmu. (Na przykład: steru kierunku jest w nim ustawiony dokładnie wzdłuż osi kadłuba, podwozie jest zupełnie wciągnięte, a kłapy — schowane).

Przygotowałem dwa uchwyty, sterujące ruchem podwozia — **X.600.Handle** i **X.601.Handle** — tak, by skrypt mógł je odnaleźć (Rysunek 7.18.2):

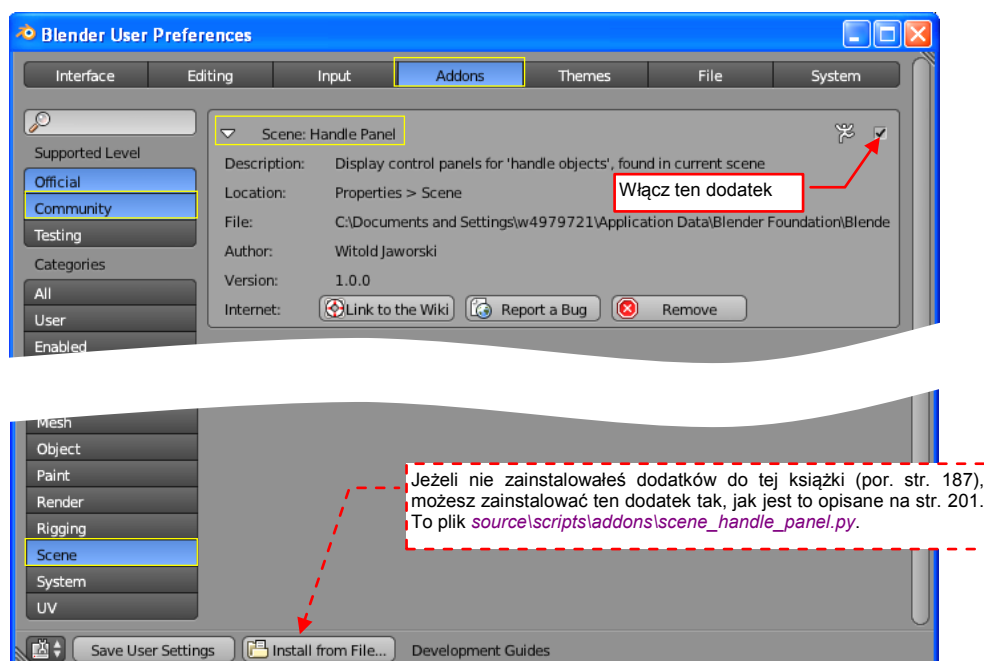


Rysunek 7.18.2 Przypisanie uchwyty „rodzica”

Uchwyt **X.600.Handle** steruje wysuwaniem i chowaniem podwozia. Ma przypisane ograniczenie *Limit Location*, wymuszające jego położenie do pionowego odcinka o długości 10 jednostki Blendera (Rysunek 7.18.2a). Gdy jest w położeniu górnym, podwozie jest schowane. W tym właśnie, górnym punkcie znajduje się jego „rodzic” (Rysunek 7.18.2b) — pusty (*Empty*) obiekt **X.600.Base**. Wyznacza w ten sposób domyślne położenie podwozia — w pozycji „złożone”.

Drugi uchwyt — **X.601.Handle** — służy do „ściskania” i „rozciągania” amortyzatorów kół głównych i pochylania wahacza kółka ogonowego. Ma przypisane ograniczenie *Limit Location*, wymuszające ruchu tego uchwyty wzdłuż pionowego odcinka o długości 1.5 jednostki Blendera. (Ta długość odpowiada skokowi amortyzatorów podwozia głównego). Rodzicem **X.601.Handle** jest uchwyt **X.600.Handle**. W dolnym położeniu, środek **X.601.Handle** pokrywa się z **X.600.Handle**. Położenie neutralne amortyzatora odpowiada więc całkowitemu rozciągnięciu (brakowi obciążenia).

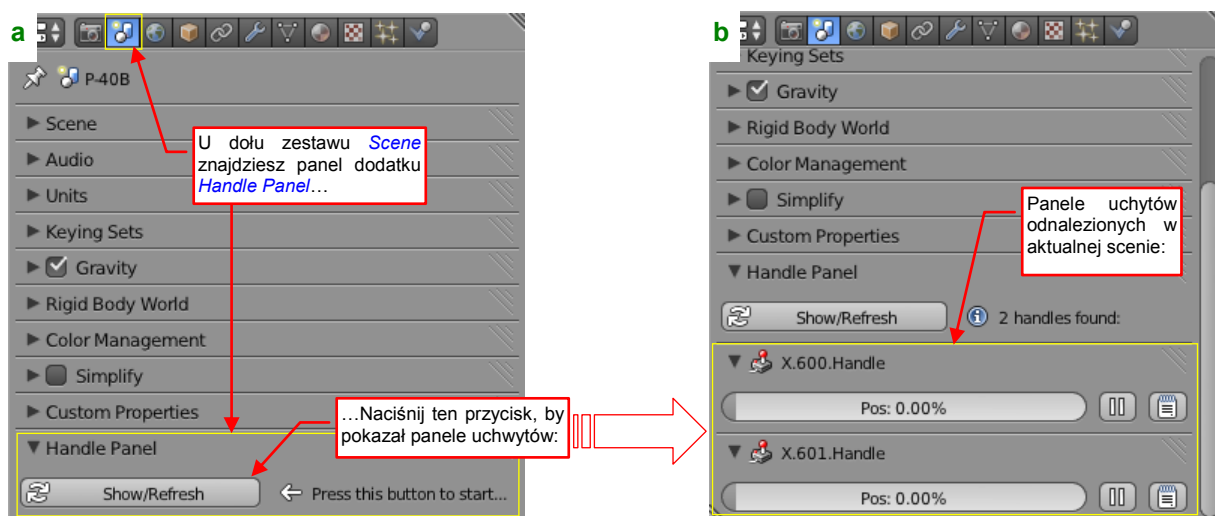
Po takich przygotowaniach można już włączyć dodatek *Handle Panel*. Powinieneś go znaleźć w oknie *User Preferences*, sekcji *Addons:Scene* (Rysunek 7.18.3):



Rysunek 7.18.3 Włączenie dodatku *Handle Panel*

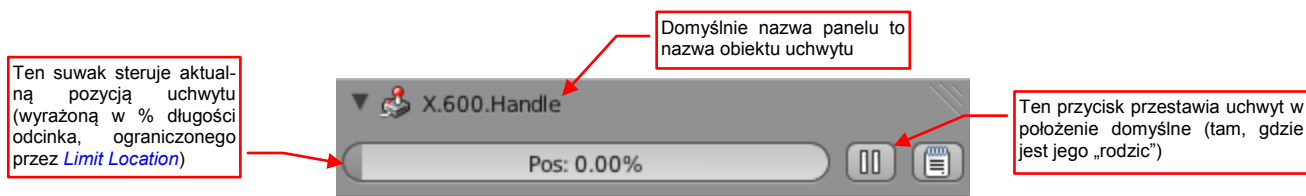


U dołu zestawu *Scene* znajdziesz panel tego dodatku (*Handle Panel* — Rysunek 7.18.4a). Naciśnij na niej przycisk *Show/Refresh*, by dodatek wyszukał i wyświetlił panel dla każdego znalezionej w scenie uchwytu (Rysunek 7.18.4b):



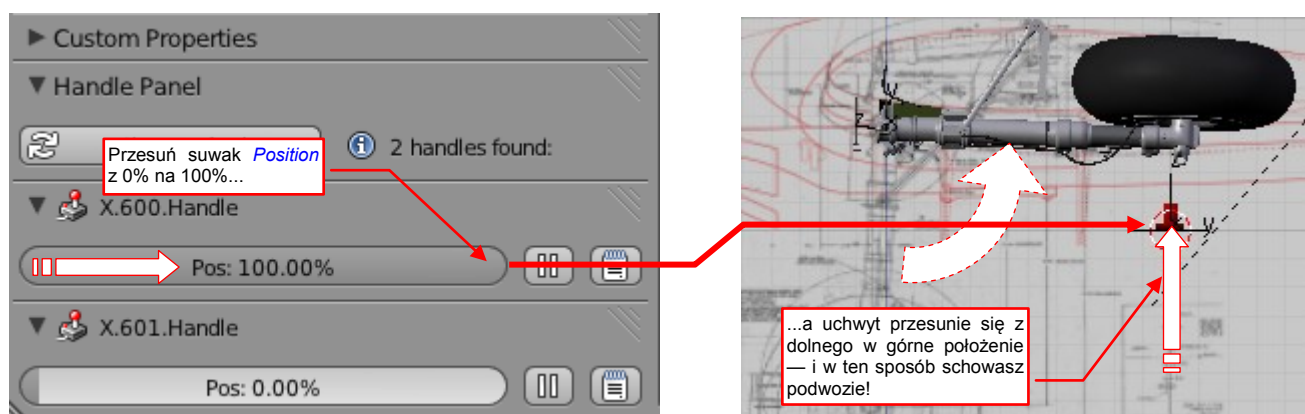
Rysunek 7.18.4 Aktywacja paneli sterujących

W naszym przypadku pojawiły się dwie panele, dla dwóch obiektów sterujących.





Rysunek 7.18.5 Panele sterowania modelem

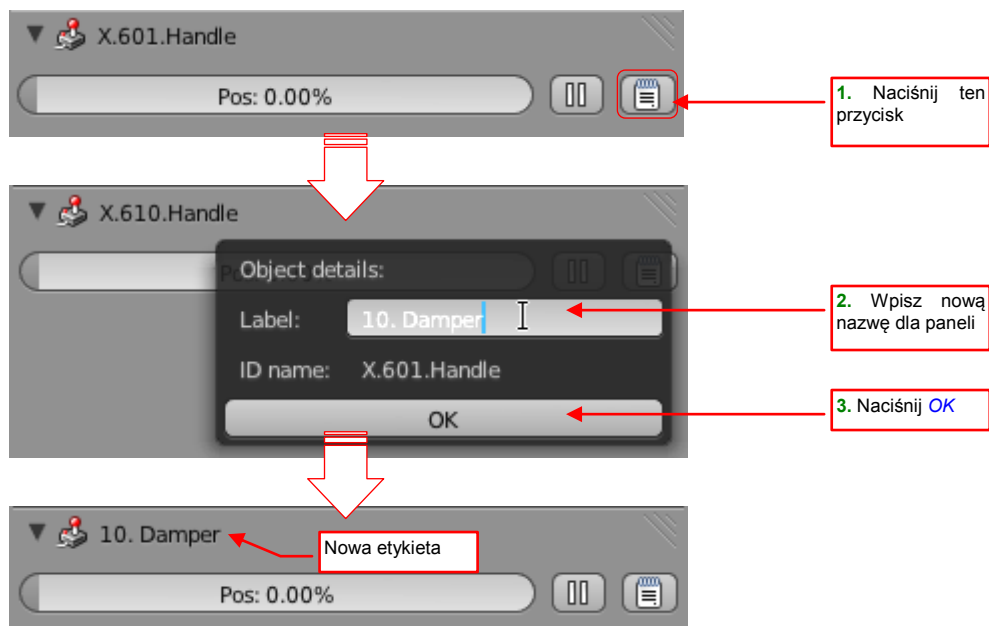
Przypuszczam, że wystarczył Ci rzut oka, by wiedzieć jak posługiwać się tym programem: należy przesunąć suwak *Pos*, umieszczony na każdym panelu (Rysunek 7.18.5). (W tę kontrolkę możesz także wpisać dokładną wartość numeryczną). Gdy podwozie jest wysunięte (por. Rysunek 7.18.2), uchwyt **X.600.Handle** jest w najniższym położeniu (0%). Także uchwyt amortyzatora **X.601.Handle** jest w położeniu 0% (najniższym) — gdy amortyzator jest całkowicie rozciągnięty. Wystarczy którykolwiek z tych suwaków przesunąć, by wywołać ruch odpowiedniego mechanizmu (Rysunek 7.18.6):



Rysunek 7.18.6 Chowanie podwozia za pomocą panelu sterowania

Dodatkowo, aby ułatwić ustawienie zespołu w jakiejś wyróżnionej pozycji neutralnej, dodałem do każdego panelu przycisk . Gdy go naciśniesz, środek uchwytu zostanie przesunięty w miejsce, w którym znajduje się środek jego obiektu — „rodzica” (*Parent*). W przypadku uchwytu z tego przykładu, „rodzic” to pomocniczy obiekt **X.600.Base**.

Domyślna nazwa paneli uchwytu jest zapożyczona z nazwy obiektu. Warto ją zmienić na coś bardziej „reprezentatywnego”. Jak wspomniałem wcześniej (por. str. 265), służy do tego przycisk  (Rysunek 7.18.7):



Rysunek 7.18.7 Zmiana nazwy panelu

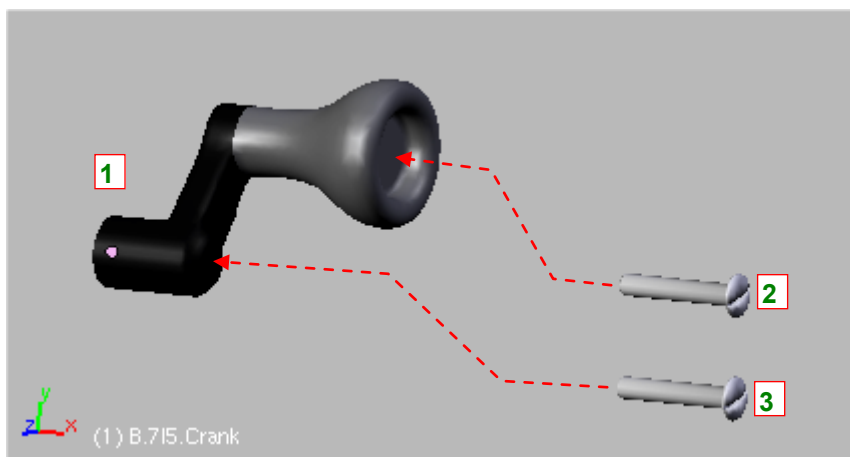
Wpisz w oknie dialogowym nową nazwę. Po naciśnięciu przycisku **OK** etykieta, wyświetlana u góry panelu, ulegnie zmianie.

- Skrypt ustawia zawsze panele wg ich nazw, w kolejności alfabetycznej. Warto więc poprzedzić je jakimś numerycznym przedrostkiem, by amortyzator został umieszczony przy podwoziu, a ster wysokości (**Elevator**) przy sterze kierunku (**Rudder**).

### 7.19 Dokładne umieszczanie obiektów w zadanym miejscu

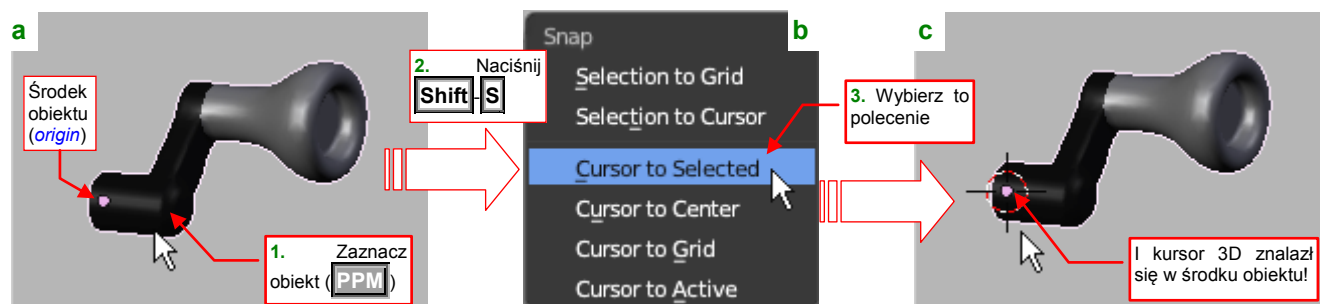
Podczas pracy nad modelem często powstaje sytuacja, gdy jakiś obiekt należy umieścić w dokładnie określonym miejscu. Jego aktualne położenie może być przypadkowe, a w każdym razie nieistotne. Robimy to dość często gdy modelujemy zespół złożony z wielu części — np. podwozie lub wnętrze kabiny. Ta sekcja pokazuje szybką i dokładną metodę wykonania takiej operacji.

Rysunek 7.19.1 przedstawia typowy przypadek opisanego powyżej problemu. Właśnie sklonowaliśmy dwie śruby (2 i 3). Śruba 3 ma się znaleźć w osi korbki 1, a śruba 2 — w centrum jej rączki. Jak to zrobić najszybciej i najłatwiej?



Rysunek 7.19.1 Typowy problem: jak najszybciej/najłatwiej osadzić śruby w osiach korbki?

Środek (punkt odniesienia — *origin*) korbki leży w jej osi obrotu. W tym samym miejscu ma się znaleźć śruba 3. Skorzystajmy z tego faktu i umieśćmy w tym punkcie kursor 3D. Aby to zrobić: zaznacz korbkę (Rysunek 7.19.2a) i naciśnij **[Shift]-[S]**, by rozwinąć podręczne menu **Snap** (Rysunek 7.19.2b). Wybierz z niego polecenie **Cursor to Selected**:



Rysunek 7.19.2 Przeniesienia kursora 3D do punktu odniesienia obiektu (*origin*)

W rezultacie, kursor 3D znajdzie się w środku (punkcie odniesienia) korbki (Rysunek 7.19.2c).

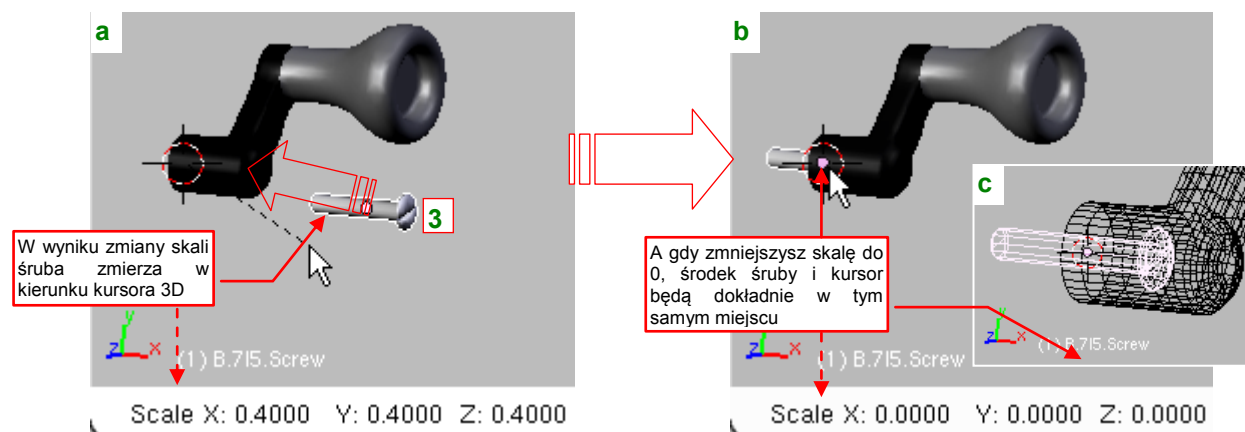
Teraz trzeba jeszcze odpowiednio ustawić tryb transformacji:

- przełącz tryb **Pivot** na **3D Cursor** (Rysunek 7.19.3a),
- włącz **Manipulate center points** (Rysunek 7.19.3b):



Rysunek 7.19.3 Ustawienie odpowiednich trybów dla punktu odniesienia i transformacji

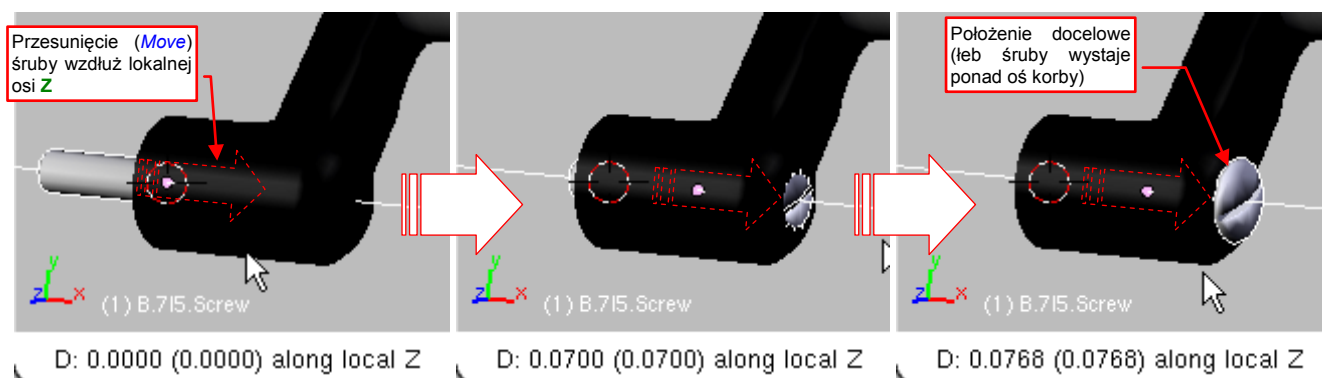
Zaznacz śrubę **3**, i naciśnij **[S]**, rozpoczynając operację zmiany skali. Gdy jest włączony tryb **Manipulate center points**, rozmiar śruby nie ulegnie zmianie. Gdy podczas zmiany skali tylko jej środek zacznie zmierzać w kierunku środka transformacji — kursora 3D. I o to chodzi! (Rysunek 7.19.4a):



**Rysunek 7.19.4 Umieszczenie śruby w zadanym punkcie (poprzez zmianę skali)**

Gdy przesuwając myszkę wciśniesz jednocześnie klawisz **[Ctrl]**, szybko i łatwo zmniejszysz skalę do zera. Dla tej wartości środek śruby **3** znajdzie się dokładnie w tym samym miejscu, co kursor 3D — czyli w osi korbki (Rysunek 7.19.4b). Tyle tylko, że został „zagłębiony” w materiale (Rysunek 7.19.4c).

To na szczęście można szybko skorygować: wystarczy przesunąć śrubę wzdłuż jej lokalnej osi **Z** (**[G][Z][Z]**) dopóty, dopóki jej łeb nie „wynurzy” się z korbki (Rysunek 7.19.5):

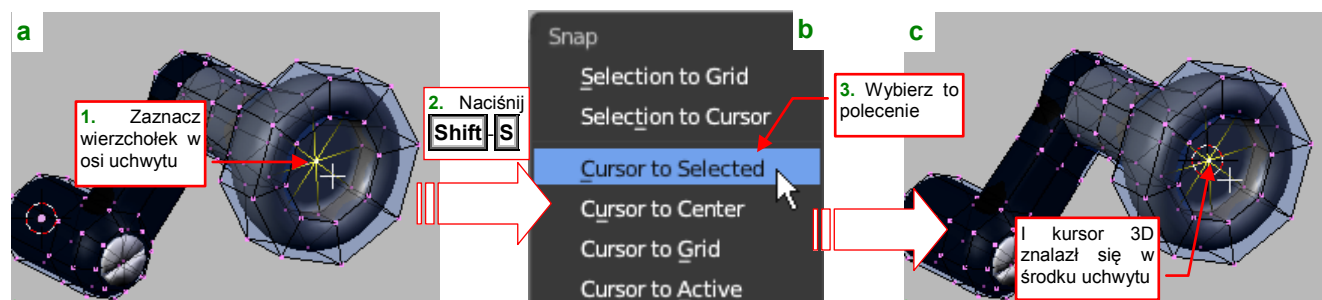


**Rysunek 7.19.5 Finalne dopasowanie (przesunięcie wzdłuż osi śruby)**

Śrubę **2** umieścimy w korbce w ten sam sposób. Musimy tylko umieścić w innym miejscu kursor 3D.

W tym celu przełączymy się na chwilę w tryb edycji siatki (**[Tab]**), i zaznaczymy wierzchołek leżący w centrum uchwytu (Rysunek 7.19.6a):

- Nie zapomnij wyłączyć opcji **Manipulate center points** po zakończeniu tej operacji. W przeciwnym razie nie będziesz mógł obrócić żadnego obiektu w tej scenie!



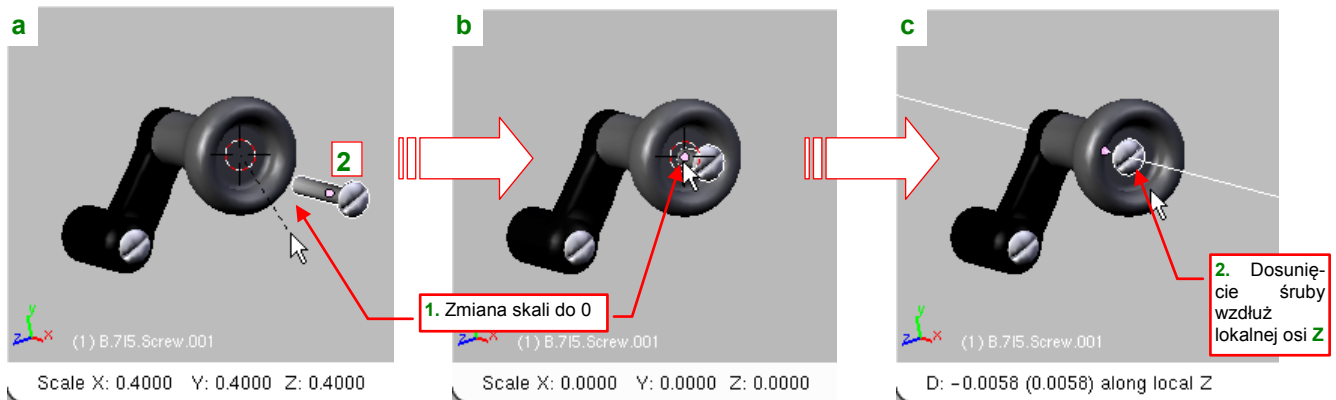


**Rysunek 7.19.6 Przeniesienia kursora 3D do środka uchwytu**

Potem postępujemy tak samo, jak poprzednio: **[Shift-S]**, by przywołać menu *Snap*, i wybrać z niego polecenie *Cursor to Selected*. W efekcie kursor 3D zostanie umieszczony w podświetlonym wierzchołku.

- Wierzchołki siatki bardzo się przydają do ustawienia kursora 3D w różnych kluczowych miejscach. Na przykład, jeżeli nawet nie masz wierzchołka w środku otworu, wystarczy zaznaczyć dwa przeciwległe wierzchołki na jego obwodzie. Polecenie *Cursor to Selected* umieści kursor 3D dokładnie pomiędzy nimi.

Gdy mamy już kursor 3D w odpowiednim miejscu, można się znowu przełączyć w *Object Mode* (**[Tab]**). Potem ze śrubą **2** postępujemy tak samo, jak ze śrubą **3**: zmiana skali do zera (Rysunek 7.19.7a, b):

**Rysunek 7.19.7 Umieszczenie śruby w centrum uchwytu**

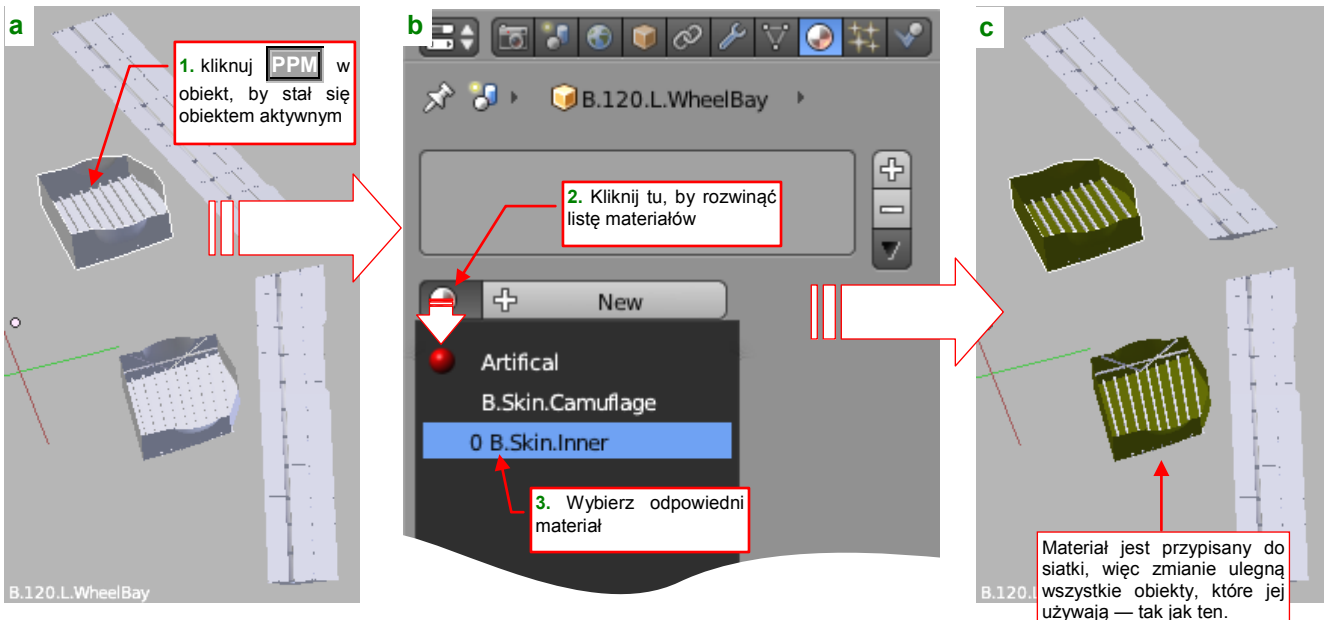
Na koniec dosuwamy łeb śruby do powierzchni korbki, przesuwając ją wzdłuż lokalnej osi **Z**. Rysunek 7.19.7c) przedstawia gotową korbkę, z dwoma śrubami osadzonymi we właściwych miejscach.

## **Rozdział 8. Blender — edytor siatki (Edit Mode)**

W Blenderze każdy kształt odwzorowany jest za pomocą ścian, połączonych w tzw. „siatkę” (*mesh*). Tryb *Edit Mode* służy do zmiany siatki pojedynczego obiektu.

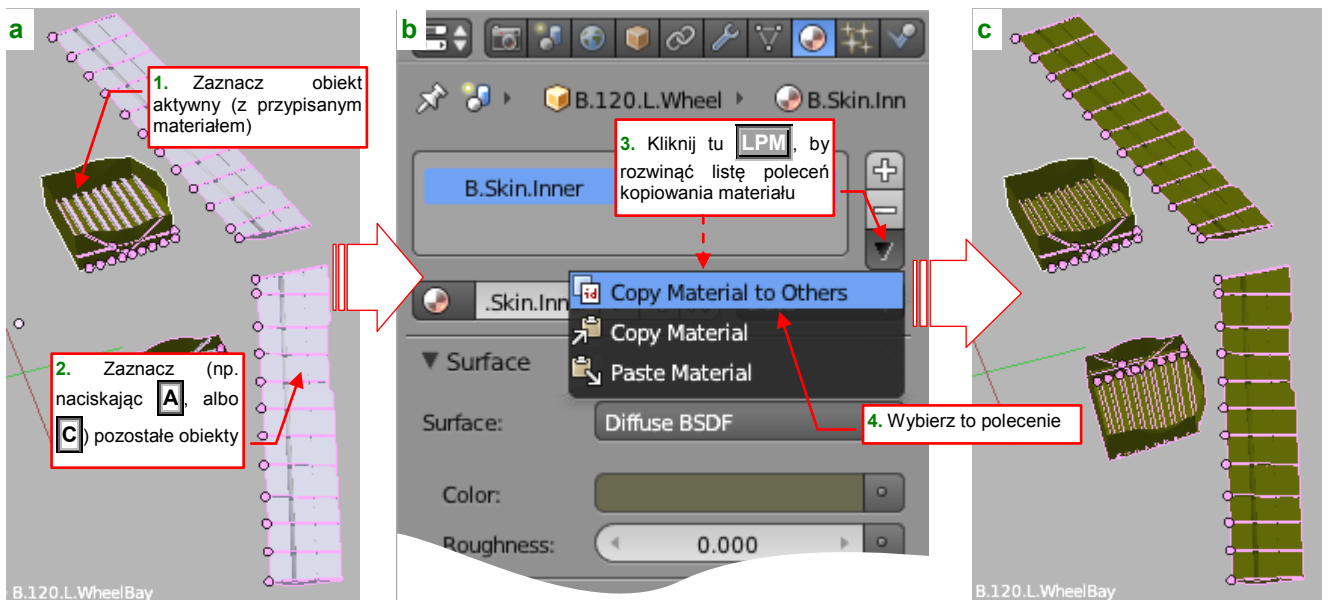
### 8.1 Przypisanie materiału do siatki

Aby przypisać pojedynczą siatkę do jakiegoś istniejącego materiału, należy zaznaczyć obiekt, który ją zawiera (tak, aby się stał obiektem aktywnym — Rysunek 8.1.1a). Następnie w oknie *Properties*, zestawie *Material* rozwiń listę z lewej strony przycisku *New*, i wybierz z niej jeden z widocznych tam materiałów (Rysunek 8.1.1b). Blender przypisuje domyślnie materiał do siatki<sup>1</sup>, a nie obiektu, stąd zmianie ulegnie nie tylko aktywny obiekt, ale także wszystkie inne obiekty, które używają tą samą siatkę (Rysunek 8.1.1c):



Rysunek 8.1.1 Przypisanie materiału do pojedynczej siatki

Aby szybko przypisać materiał do wielu obiektów (siatek) naraz, zaznacz je i wykorzystaj polecenie *Copy Material to Others* (Rysunek 8.1.2):



Rysunek 8.1.2 Skopiowanie przypisania materiału do wielu siatek naraz

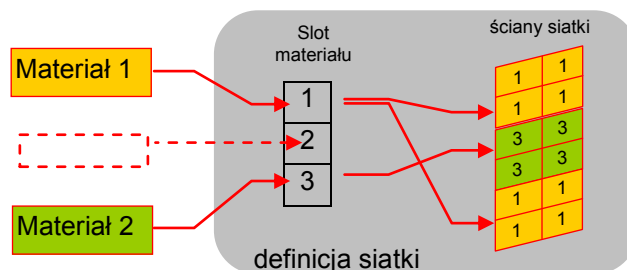
<sup>1</sup> W konfiguracji, jaką wybraliśmy dla Blendera (por. Rysunek 6.1.3, str. 747 — okno *User Preferences*, przełącznik *Link Materials To:*) materiał jest domyślnie przypisany do siatki, a nie do obiektu. Stąd operacje przedstawione na tej stronie można wykonać zarówno w *Edit Mode*, jak i w *Object Mode*. Zdecydowałem się je tutaj pokazywać w *Object Mode*, bo grupowe przypisanie materiału wymaga zaznaczenia wielu obiektów. W tym trybie wyraźnie widać, co jest obiektem aktywnym, a co zaznaczonym (Rysunek 8.1.2a).

Zwróć uwagę że w menu, które przedstawia Rysunek 8.1.2b), są także polecenia **Copy Material** i **Paste Material**. Możesz ich także użyć do przypisania pojedynczego materiału, jeżeli nie chcesz wybierać go co chwila z listy. Polecenie **Copy Material to Others** znajduje się także pod inną nazwą — **Material** — w menu **Object→Make Links...** (dostępnym pod skrótem **Ctrl-L**).

A jak przypisać do różnych fragmentów pojedynczej siatki różne materiały?

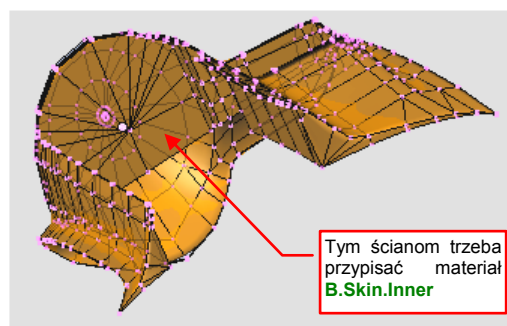
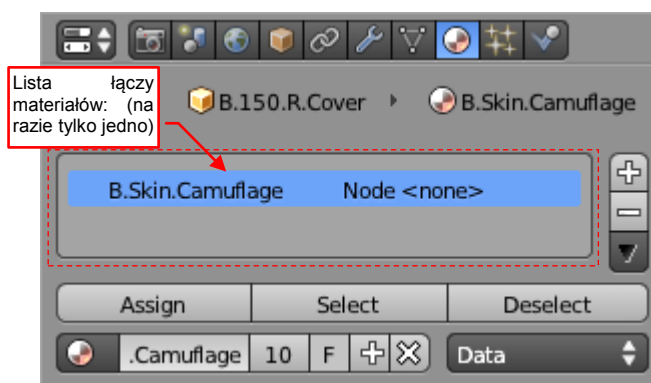
W Blenderze materiały są przypisane do ścian siatki za pośrednictwem tzw. łączy materiałów (**material slots** — Rysunek 8.1.3).

W każdej siatce możesz stworzyć wiele łączy materiałów. Każda ściana jest przypisana tylko do jednego z nich. Niektóre z takich łączy mogą nie mieć przypisanych ścian. Mogą także nie mieć przypisanych materiałów. (Takie "puste" łączy nie mają wpływu na rendering).



Rysunek 8.1.3 Powiązanie ścian siatki z materiałem poprzez tzw. łączy (**material slots**)

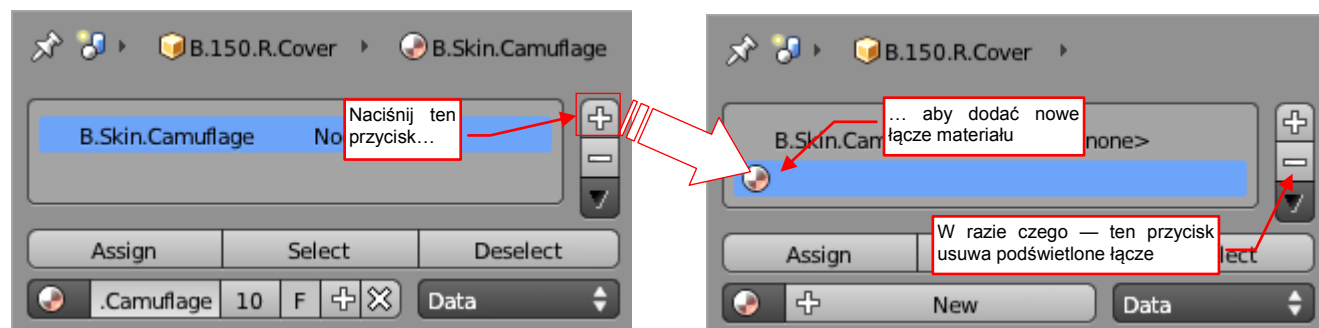
Przypisanie fragmentu siatki do innego materiału pokażę na przykładzie gondoli podwozia głównego (obiektu **B.150.R.Cover**). Rysunek 8.1.4 przedstawia „sytuację wyjściową”:



Rysunek 8.1.4 Siatka, której fragmentowi chcemy przypisać inny materiał

Wszystkie ściany siatki owiewki ma już przypisany materiał odpowiedni dla powierzchni zewnętrznych: **B.Skin.Camouflage**. Teraz chcemy ścianom wewnętrznym przypisać materiał **B.Skin.Inner**.

Zaczynamy od utworzenia nowego, pustego łączy materiału. To „miejsce” na nowy materiał (Rysunek 8.1.5):



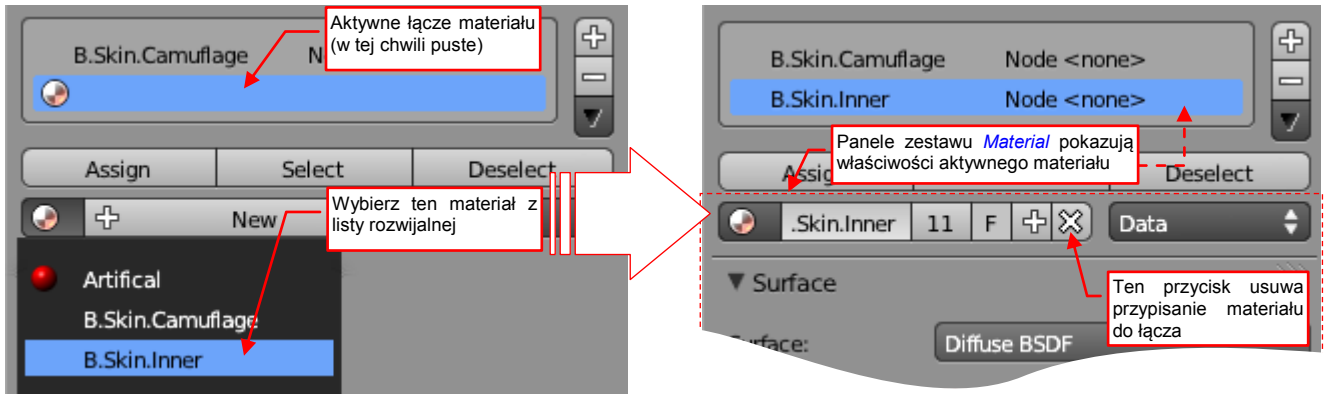
Rysunek 8.1.5 Dodanie do siatki nowego, pustego łączy materiału.

Naciśnij przycisk **+** z prawej strony listy łączy materiałów. Blender doda na koniec listy nowe, puste łączy. Przesunie także na nie podświetlenie, czyniąc je **łączem aktywnym**.

- Tak jak można się spodziewać, przycisk **-** usuwa aktywne (tzn. podświetlone na liście) łączy materiału. Można jednak to robić tylko w **Object Mode**!

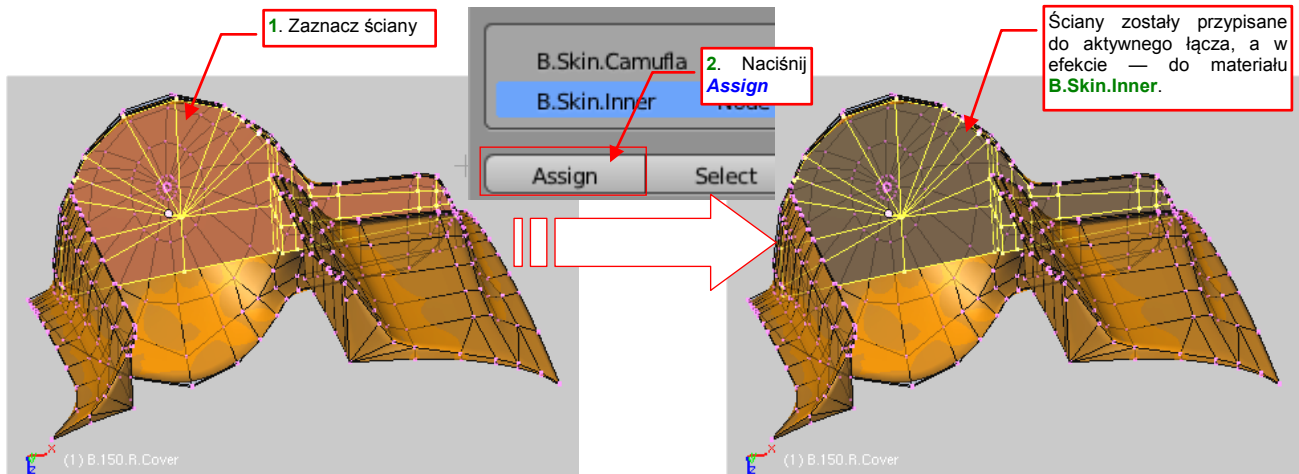


Teraz przypisujemy do naszego łącza materiał — w tym przypadku **B.Skin.Inner** (Rysunek 8.1.6):



Rysunek 8.1.6 Przypisanie materiału do łącza.

Na koniec przypisz do łącza ściany. Zaznacz je w oknie *View 3D*, a następnie naciśnij w panelu *Material* przycisk **Assign** (Rysunek 8.1.7):



Rysunek 8.1.7 Przypisanie ścian do nowego indeksu materiału.

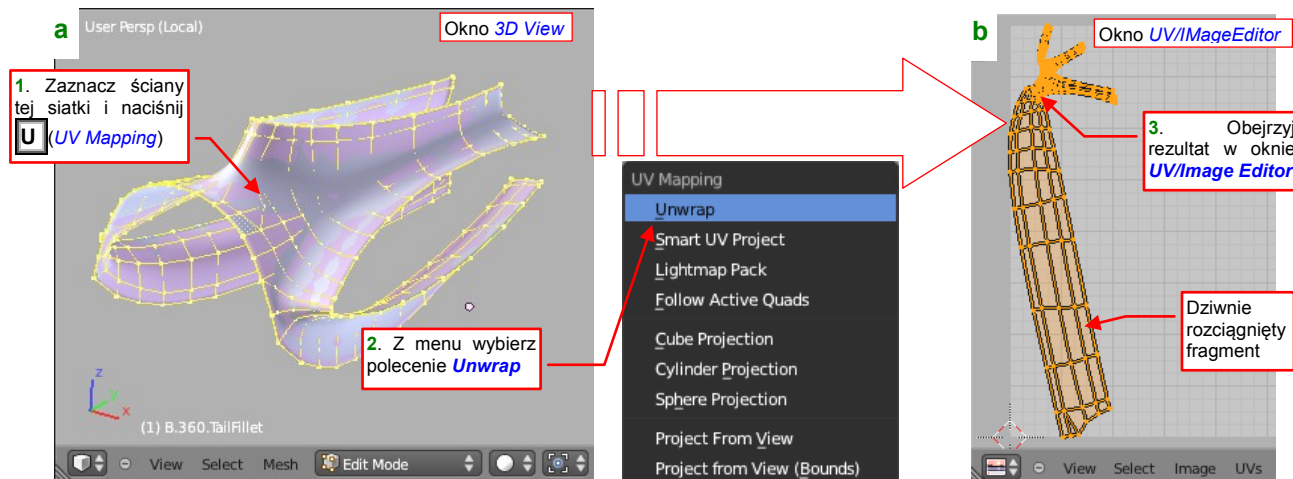
Nie musisz zaznaczać od razu wszystkich ścian. Możesz tę operację przypisania powtórzyć kilka razy (w przypadku bardzo złożonych kształtów tak jest prościej).

- Zwróć uwagę, by obszar siatki przypisany do łącza materiału nie zawierał żadnej nie powiązanej ze ścianą krawędzi lub wierzchołka. Takie miejsca mogą powodować różne artefakty na ostatecznym renderze.

Możesz używać ścian przypisanych do łącza do selekcji wierzchołków siatki w trakcie jej edycji. (Służą do tego umieszczone poniżej listy łącza przyciski **Select** i **Deselect**)

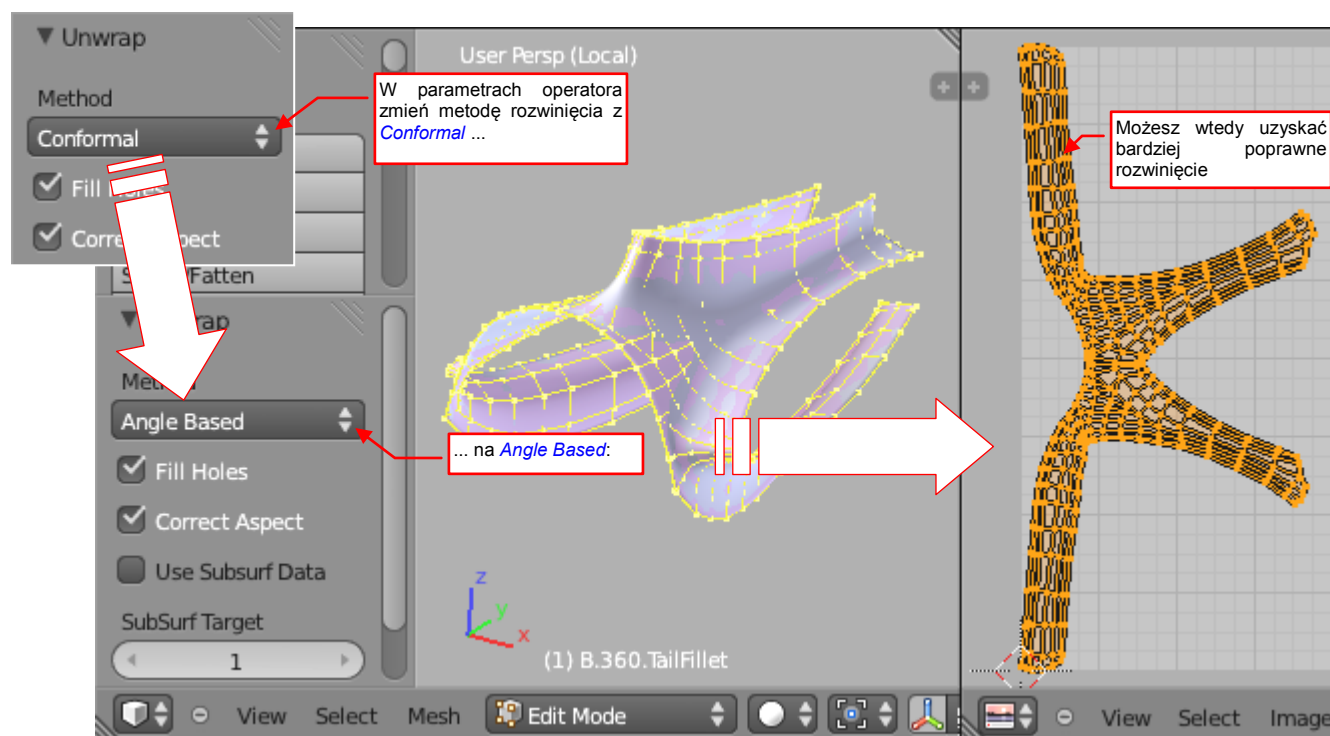
## 8.2 “Zwykłe” rozwijanie siatki (*Unwrap*)

Zaznacz ściany siatki, które chcesz rozwinąć (to może być nawet mały fragment, niekoniecznie cała siatka). Następnie naciśnij **U** (*Mesh → UV Unwrap*). Na ekranie pojawi się menu *UV Mapping*, z alternatywnymi metodami rozwijania siatek. Wybierz z niego polecenie *Unwrap* (Rysunek 8.2.1):



Rysunek 8.2.1 Zwykłe rozwinięcie (*Unwrap*) siatki w przestrzeni UV

Rezultat rozwinięcia zobaczysz w oknie *UV/Image Editor*. Jeżeli wyszło zdeformowane, np. tak jak pokazuje Rysunek 8.2.1b, możesz spróbować użyć alternatywną metodę rozwijania. W parametrach operatora (**T** lub **F6**) *Unwrap* wybierz z listy rozwijalnej inny tryb (Rysunek 8.2.2):

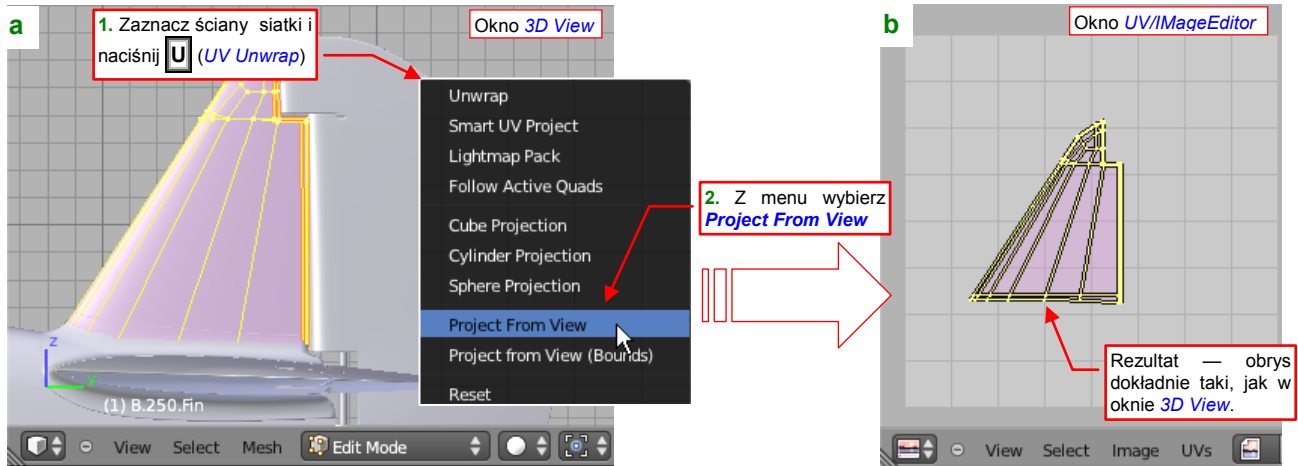


Rysunek 8.2.2 Sterowanie metodą rozwinięcia

Jak do tej pory, metoda *Unwrap* używa jednej z dwóch metod rozwijania: *Conformal* i *Angle Based*. Jeżeli nie jesteś zadowolony z aktualnej, przestaw ją na alternatywną (Rysunek 8.2.2). Rezultat operacji może wówczas ulec poprawie (porównaj Rysunek 8.2.2 i Rysunek 8.2.1b). Jeżeli nadal nie jesteś z niego zadowolony — pozostaje poprawić rozwinięcie „ręcznie”, tzn. poprzez zmianę położenia ścian w edytorze UV.

### 8.3 Rozwijanie poprzez rzutowanie siatki (*Project from View*)

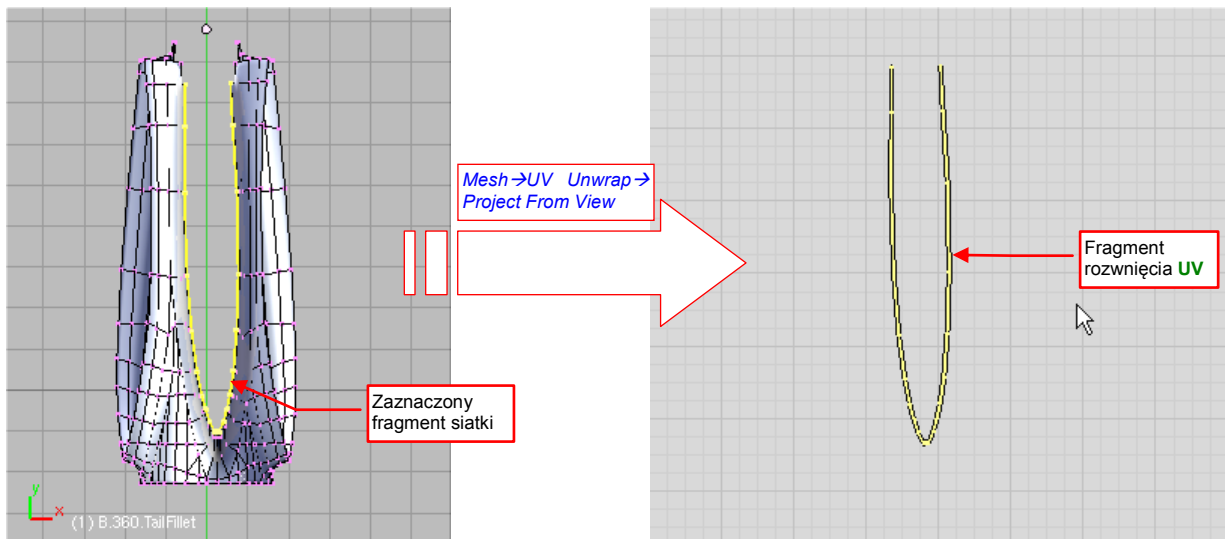
Zaznacz ściany siatki, które chcesz rozwinąć w przestrzeni **UV**. Następnie naciśnij **U** (**Mesh → UV Unwrap**). Na ekranie pojawi się menu **UV Mapping**, z alternatywnymi metodami rozwijania siatek. Wybierz z niego opcję **Project From View** (Rysunek 8.3.1):



Rysunek 8.3.1 Rozwijanie UV poprzez rzutowanie na płaszczyznę widoku (*Project From View*)

Rezultat rozwinięcia zobaczysz w oknie **UV/Image Editor**. Siatka wygląda dokładnie tak samo, jak w oknie **3D View**, ale jest płaskim obrazem, umieszczonym na płaszczyźnie **UV**.

Za pomocą polecenia **Project From View** można rozwijać nie tylko całe siatki, ale także ich pojedyncze fragmenty. Przykład takiej operacji pokazuje Rysunek 8.3.2:



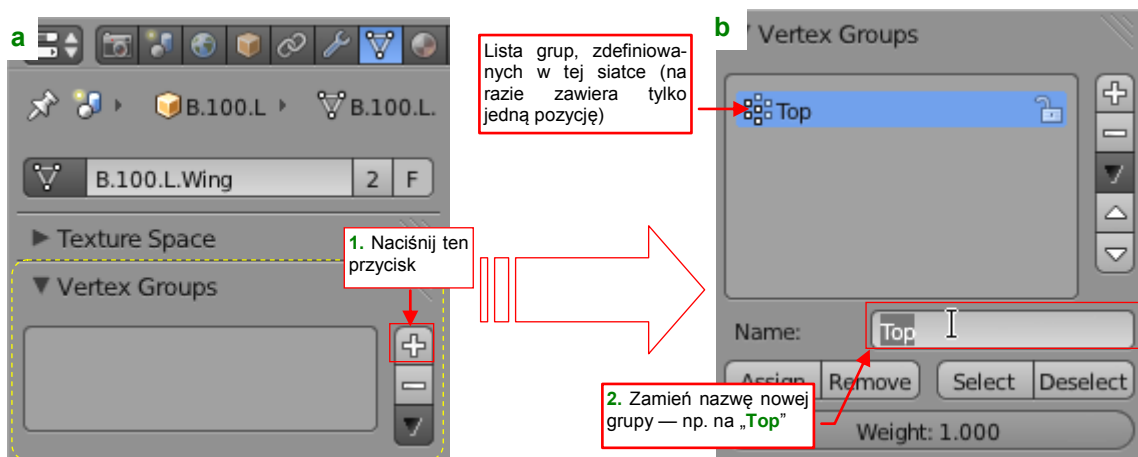
Rysunek 8.3.2 Rzutowanie na płaszczyznę UV fragmentu siatki

Gdy zaznaczysz na siatce w oknie **3D View** tylko jej fragment (jak np. krawędź owiewki, którą pokazuje Rysunek 8.3.2), i wywołasz polecenie **UV Unwrap → Project From View**, w oknie **UV/Image Editor** zobaczysz rzut tego fragmentu. Możesz go np. zaraz po rzutowaniu „przyspilić” (**UVs → Pin**).

## 8.4 Definiowanie grupy wierzchołków (*Vertex Group*)

W Blenderze istnieje możliwość przypisania wierzchołków siatki do grupy. Została stworzona z myślą o powiązaniu powierzchni modelu z animowanym szkieletem. Może się jednak przydać do „zapamiętania” jakiegoś skomplikowanego obszaru siatki, który będziesz często zaznaczał.

Nową grupę wierzchołków tworzysz w zestawie *Object Data*, panelu *Vertex Groups* (Rysunek 8.4.1):

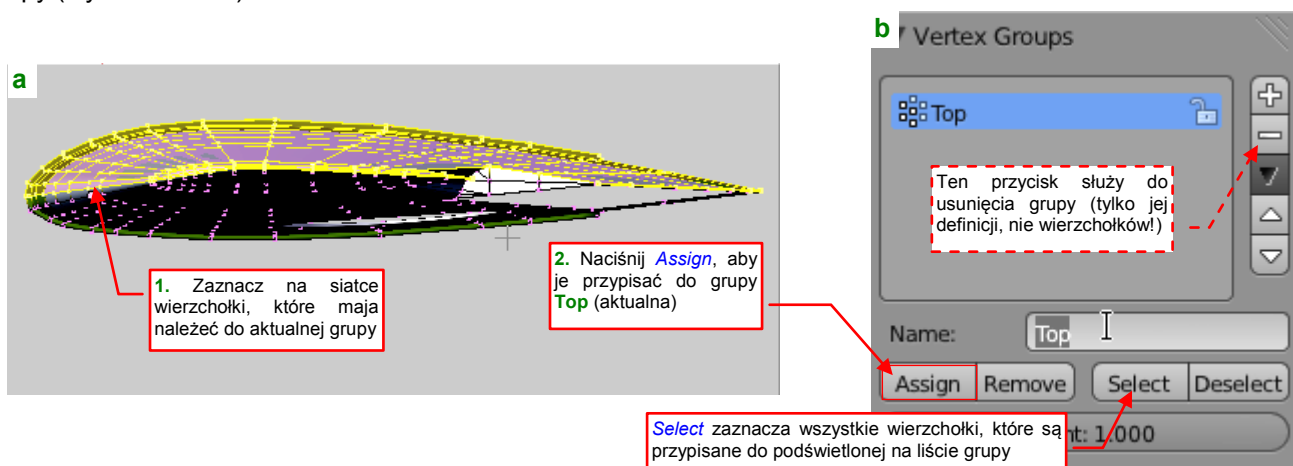


Rysunek 8.4.1 Stworzenie i zmiana nazwy nowej grupy wierzchołków

Po naciśnięciu przycisku **+** (Rysunek 8.4.1a) Blender utworzy nową grupę. Na liście *Vertex Groups* pojawi pierwsza pozycja (Rysunek 8.4.1b). Zobaczysz w niej napis **Group** (taka nazwa jest domyślnie nadawana każdej nowo utworzonej grupie). Zmień ją na jakąś inną — na przykład **Top**.

- Do usunięcia aktualnej grupy, której nie potrzebujesz, służy przycisk **-**. Usunięciu ulegają tylko przypisania wierzchołków do grupy i pozycja z listy. Same wierzchołki siatki nie są, rzecz jasna, usuwane.

Na razie z grupą nie jest związany żaden wierzchołek. Zaznacz odpowiedni obszar siatki w oknie *3D View*, a następnie naciśnij w panelu *Vertex Groups* przycisk **Assign**. To przypisze wybrane wierzchołki do aktualnej grupy (Rysunek 8.4.2):



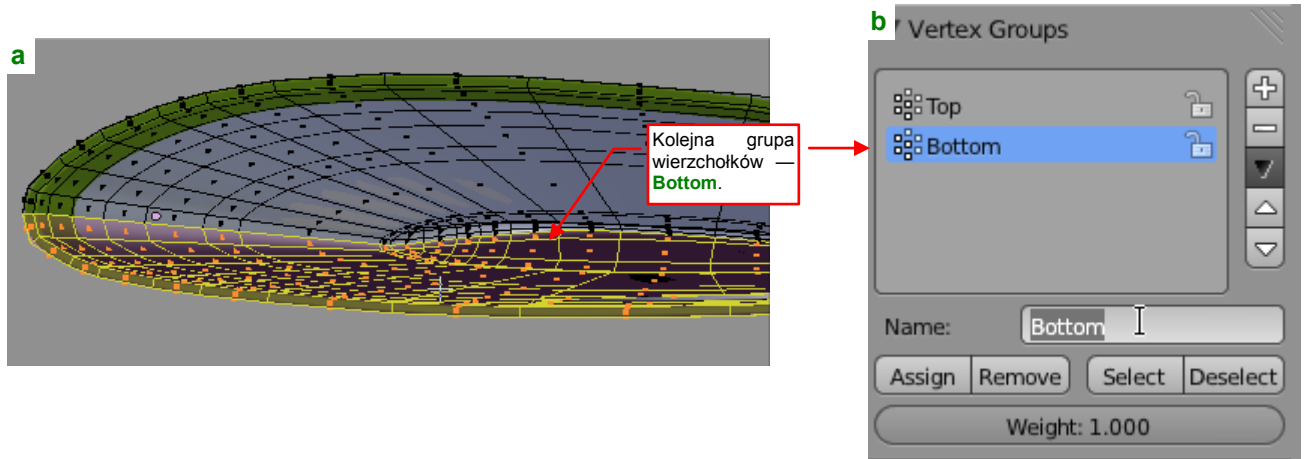
Rysunek 8.4.2 Przypisanie do grupy wierzchołków

Samo przypisanie wierzchołków do grupy nie powoduje żadnej widocznej zmiany w oknie *3D View*. Aby się przekonać, czy są naprawdę w aktualnej grupie, wyłącz aktualne zaznaczenie (**A**). Następnie naciśnij przycisk **Select** (Rysunek 8.4.2b). Sprawdź, czy wszystkie wierzchołki, które powinny należeć do grupy, są zaznaczone. Jeżeli nie — zaznacz brakujące, i jeszcze raz naciśnij przycisk **Assign**.



- Przycisk **Select** dodaje do aktualnego zaznaczenia wszystkie wierzchołki aktywnej (tzn. podświetlonej na liście *Vertex Groups*) grupy, a **Deselect** — wyłącza ich zaznaczenie.

Całą powierzchnię możesz podzielić na kilka grup. Skoro zdefiniowaliśmy już górną powierzchnię płata jako grupę, dodajmy w ten sam sposób drugą, zawierającą „całą resztę”. Stwórz ją w ten sam sposób, co poprzednią. Nadaj jej nazwę **Bottom** (Rysunek 8.4.3b):



Rysunek 8.4.3 Przypisanie reszty siatki do drugiej grupy wierzchołków (**Bottom**)

Najprościej wybrać „całą resztę” siatki zaznaczając (przycisk **Select**) grupę **Top**. Potem należy przełączyć tryb wyświetlania z wierzchołków na ściany, i z menu okna *3D View* wywołać polecenie **Selection → Inverse** (**Ctrl** - **I**). Powinieneś uzyskać takie zaznaczenie, jakie pokazuje Rysunek 8.4.3a). Na koniec pozostaje nacisnąć przycisk **Assign**, aby przypisać ten fragment siatki do grupy **Bottom**. Uważaj, aby nie zaznaczyć za dużo, gdyż Blender nie stawia tu przeszkód, i ten sam wierzchołek może należeć do wielu grup.

Na koniec kilka dodatkowych wskazówek:

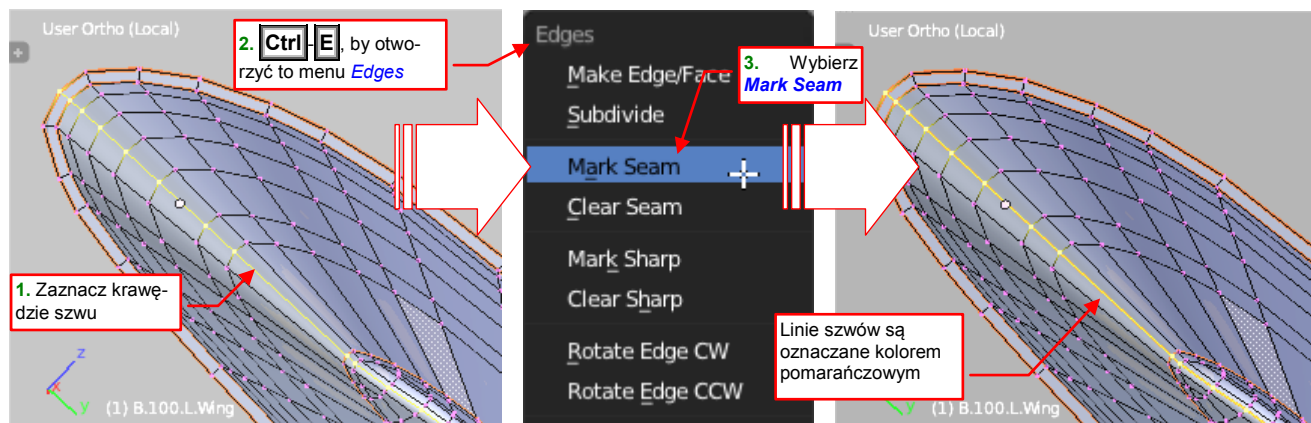
- Pojedynczy wierzchołek siatki może należeć równocześnie do wielu różnych grup wierzchołków.
- Nazwy grup muszą być unikalne tylko w obrębie tej samej siatki. Oznacza to, że wiele różnych siatek w Twoim modelu może mieć grupy o takich samych nazwach — **Top**, **Bottom**, czy podobnych.

(To pewnie ułatwienie, gdyż nie trzeba tworzyć jakiegoś kolejnego systemu nazw).

## 8.5 Zaznaczenie krawędzi jako szwu (*Seam*)

Podczas rozwinięcia UV, Blender automatycznie dzieli siatkę wzdłuż krawędzi, które oznaczysz jako „szwy” (*seams*).

Nanoszenie „szwów” jest bardzo proste: zaznacz krawędzie, wzdłuż których ma być podzielone rozwinięcie UV. Następnie naciśnij **Ctrl-E**, i z menu *Edges* wybierz polecenie **Mark Seam** (Rysunek 8.5.1):

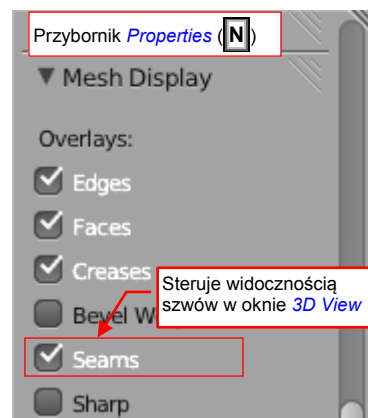


Rysunek 8.5.1 Oznaczanie krawędzi siatki jako linii szwów (*Mark Seam*)

(To samo polecenie znajdziesz także w menu: *Mesh → Edges → Mark Seam*).

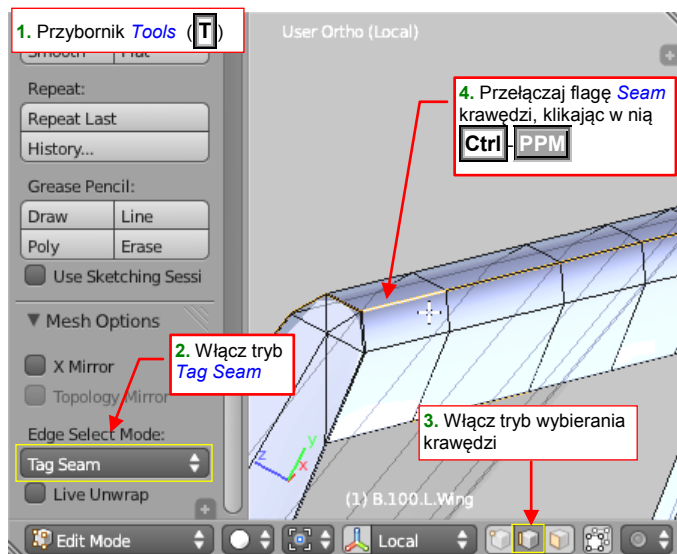
- Aby usunąć szwy z zaznaczonych krawędzi, użyj polecenia **Clear Seam**.

Szwy są na siatce oznaczane pomarańczowym pogrubieniem. Jeżeli nie widzisz żadnego dodatkowego oznaczenia linii szwów, które stworzyłeś — zajrzyj do przybornika *Properties* (**N**), panelu *Mesh Display* (Rysunek 8.4.2). Sprawdź, czy masz tam włączoną opcję **Seams**. Jeżeli nie — możesz ją tu włączyć.



Rysunek 8.5.2 Sterowanie wyświetlaniem szwów siatki

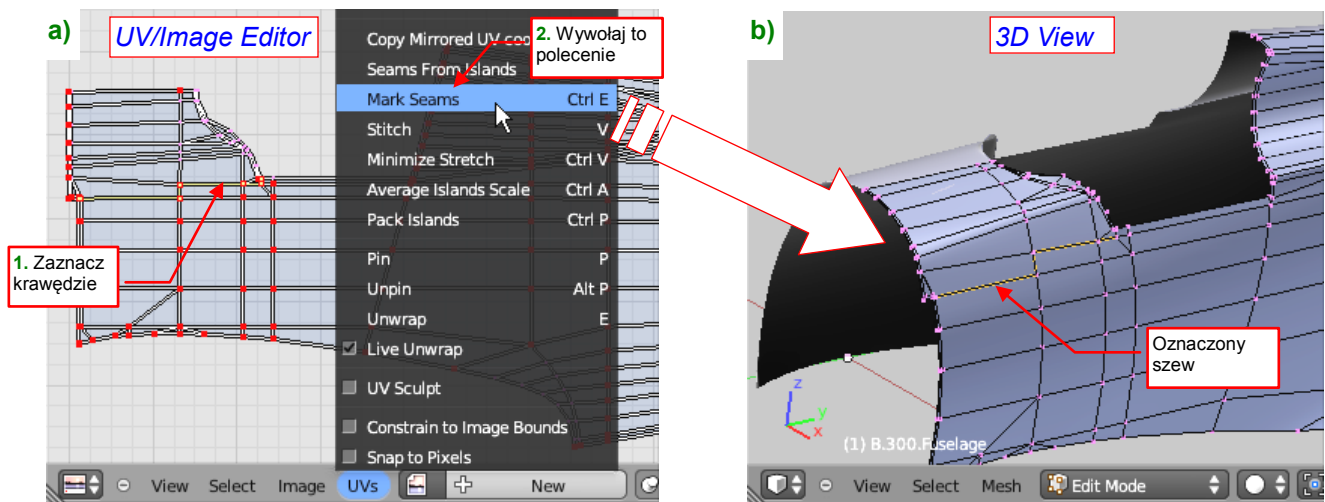
Do zaznaczania szwów możesz także wykorzystać odpowiedni tryb z panelu *Mesh Options* (z przybornika narzędzi — Rysunek 8.5.3):



Rysunek 8.5.3 Alternatywa metoda oznaczania szwów

Po włączeniu *Mesh Options:Edge Select Mode* na tryb **Tag Seam**, przełącz jeszcze w nagłówku okna *3D View* tryb selekcji siatki na krawędzie (*Edges*). Od tej chwili kliknięcie **Ctrl-PPM** w krawędź siatki oznacza ją jako szew. Powtórne takie kliknięcie wyłącza to oznaczenie.

Czasami łatwiej jest znaleźć i zaznaczać linie szwów bezpośrednio na rozwinięciach w oknie *UV/Image Editor*. Dlatego tam także jest dostępne polecenie *UVs → Mark Seams* (Rysunek 8.5.4a):



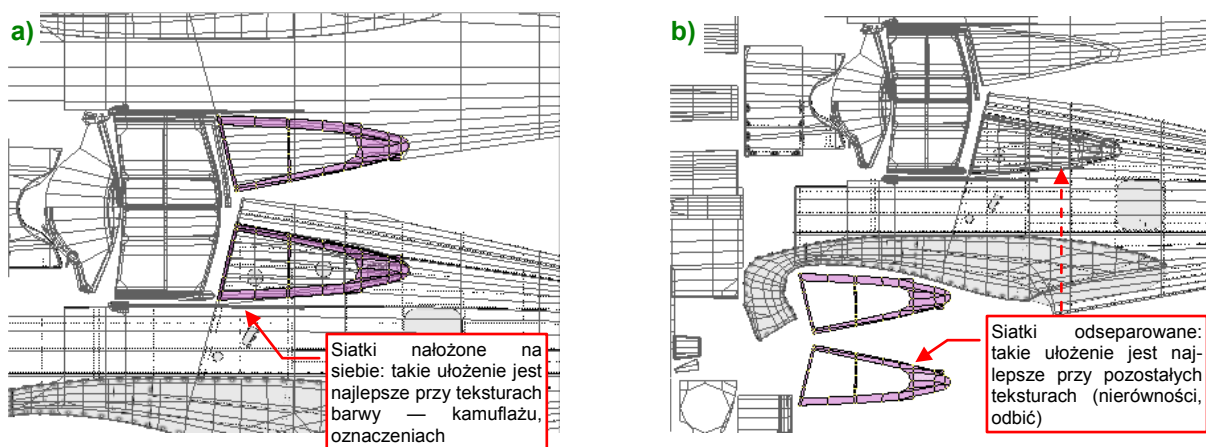
Rysunek 8.5.4 Oznaczanie linii szwów w *UV/Image Editor*

Po zaznaczeniu krawędzi i wywołaniu polecenia w samym oknie *UV/Image Editor* nic nie ulegnie zmianie<sup>1</sup>. Efekt polecenia zobaczysz za to w oknie *3D View* (Rysunek 8.5.4b).

<sup>1</sup> Przynajmniej w Blenderze 2.62, który był aktualną wersją w chwili pisania tego fragmentu książki

## 8.6 Dodanie alternatywnego rozwinięcia UV (*UV Map*)

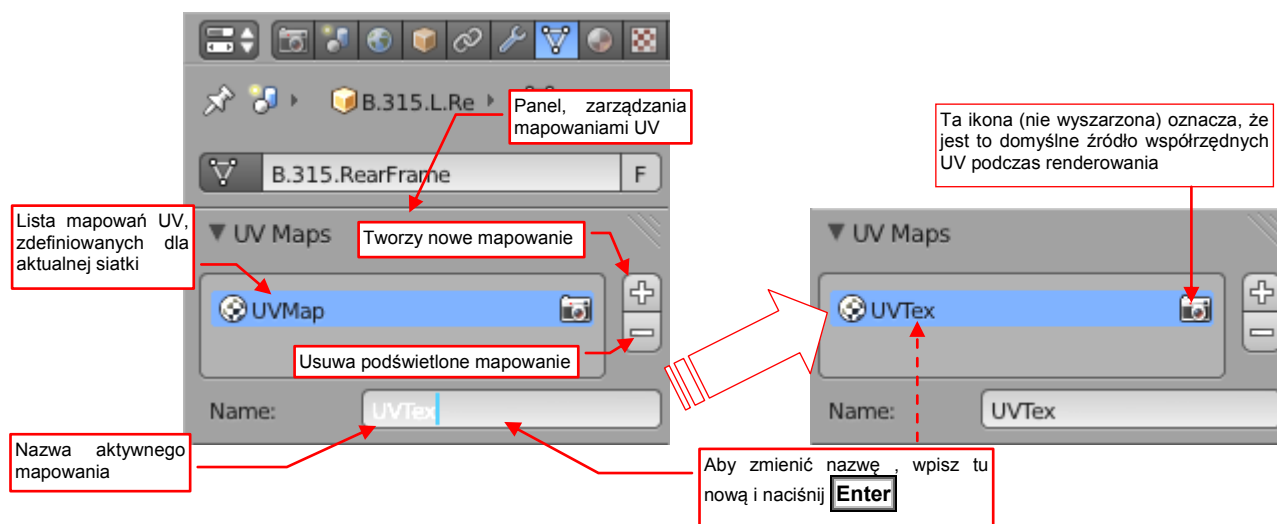
Czasami możesz natknąć się na sytuację, gdy dla dwóch różnych tekstur przydałyby się dwa różne rozwinięcia siatki w UV. Pokażę to na przykładzie ramki oszkleń za kabiną pilota P-40 (Rysunek 8.6.1):



Rysunek 8.6.1 Alternatywne rozwinięcia w UV siatki ramek oszkleń za głową pilota

Przy nakładaniu barw — plam kamuflażu, znaków rozpoznawczych, itp. — najlepiej, by siatka ramki znajdowała się tam, gdzie była przymocowana: na powierzchni kadłuba (Rysunek 8.6.1a). Przy takim ustawieniu jednym pociągnięciem w Gimpie namalujesz idealnie dopasowany kamuflaż na obydwu elementach. Z kolei dla tekstury nierówności (czy też odbić) lepiej byłoby, gdyby rozwinięcia kadłuba i ramki były odseparowane (Rysunek 8.6.1b). Każdy z tych elementów ma swój własny układ linii nitów i innych detali konstrukcyjnych. Gdy jedna siatka leży na drugiej — nie jesteś w stanie ich prawidłowo odwzorować (bo na każdej z powierzchni zobaczysz naraz wszystkie linie).

Na szczęście w Blenderze każda siatka może mieć kilka alternatywnych rozwinięć UV (określanych także jako „mapowania UV” — *UV Maps*). Do zarządzania tymi rozwinięciami służy panel *UV Maps* z zestawu *Object Data* (Rysunek 8.6.2):

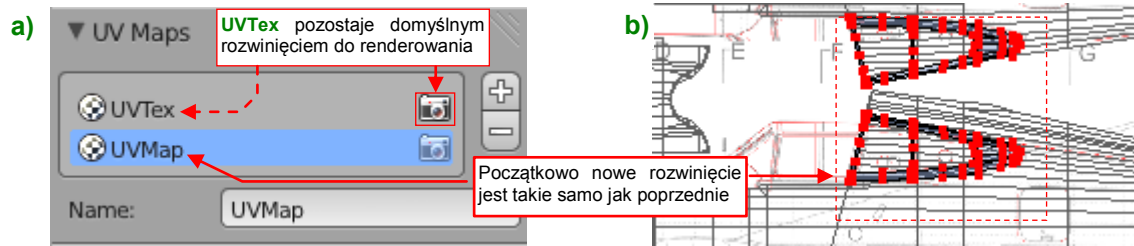


Rysunek 8.6.2 Panel zarządzania mapowaniami UV (*UV Maps*)




Gdy dokonałeś w edytorze UV rozwinięcia siatki, Blender automatycznie utworzył pierwszą mapę, o nazwie **UVMap** (Rysunek 8.6.2). Możesz zmienić jej nazwę na inną, korzystając z umieszczonego pod listą pola *Name*. Nazwy map UV muszą być unikalne tylko w obrębie pojedynczej siatki. Mogą (a w przypadku naszego modelu — muszą) się powtarzać w różnych siatkach. Gdy naciśniesz przycisk **+**, Blender stworzy duplikat podświetlonego na liście mapowania. Podobnie przycisk **-** umożliwia usunięcie rozwinięcia UV.



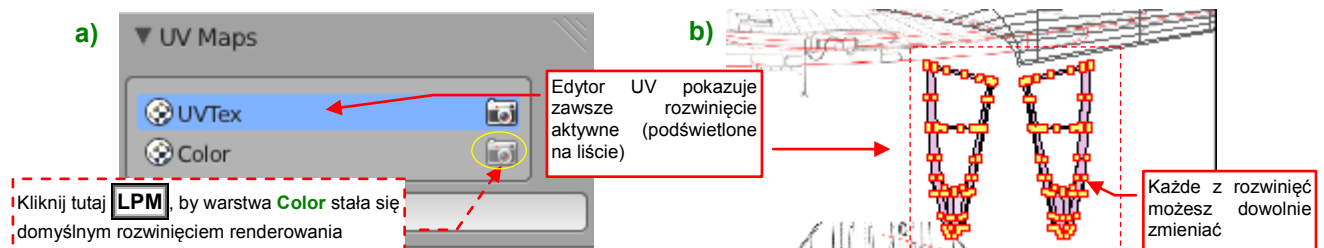
Gdy naciśniesz przycisk , poniżej mapy **UVTex** pojawi się jej duplikat, o nazwie **UVMap** (Rysunek 8.6.3):



Rysunek 8.6.3 Nowa mapa UV

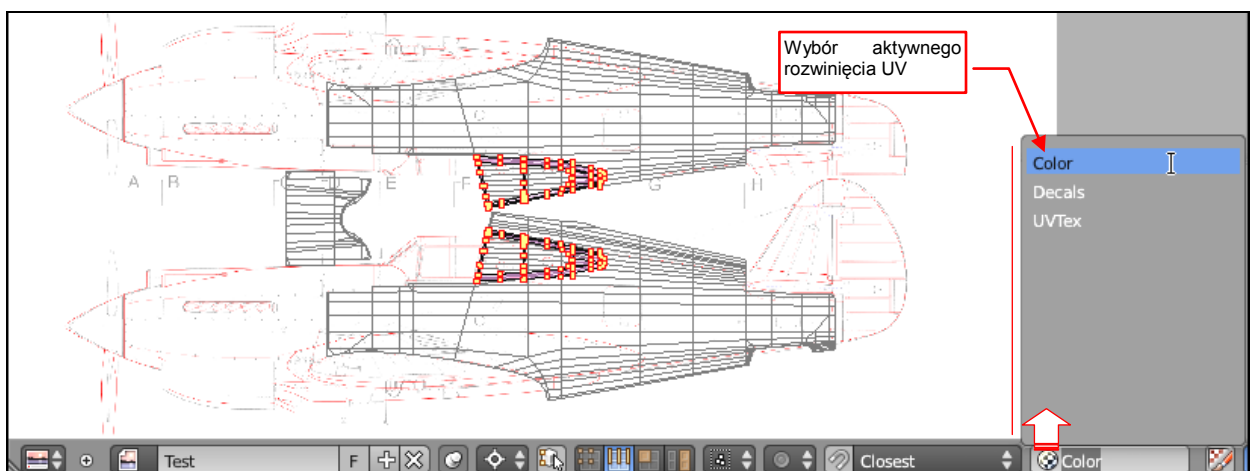
Aby o tym potem nie zapomnieć, zmień od razu nazwę nowego mapowania na docelową — np. na **Color** (Rysunek 8.6.4a). Zwróć także uwagę na ikonę aparatu fotograficznego () umieszczoną z prawej strony nazwy każdego z rozwinięć. Tylko jedna spośród wszystkich pozycji na liście może mieć tę ikonę w stanie nie wyszarzonym (por. Rysunek 8.6.3a i Rysunek 8.6.4a). Taka „żywa” ikona oznacza mapowanie używane domyślnie podczas renderowania — czyli to, które Cycles zwróci na wyjściu **Texture Coordinate:UV**. To nie jest tylko informacja, ale także kontrolka! Gdy klikniesz w jedną z tych wyszarzonych () ikon **LPM** — zmieni swój stan na „żywy” (). W ten sposób mapowanie UV, które ją zawiera, stanie się mapowaniem domyślnym. W oknie **UV/Image Editor**, a także w oknie **3D View**, w trybie **Textured**, Blender używa zawsze mapowania aktywnego — czyli tego, które jest podświetlone na liście **UV Maps**. Zwróć uwagę (Rysunek 8.6.3a) że to wcale nie musi być domyślne mapowanie renderu.

Po skopiowaniu rozwinięcia UV możesz je zaznaczyć i dowolnie zmieniać w **UV/Image Editor** (Rysunek 8.6.4):



Rysunek 8.6.4 Zmiana jednej z map UV

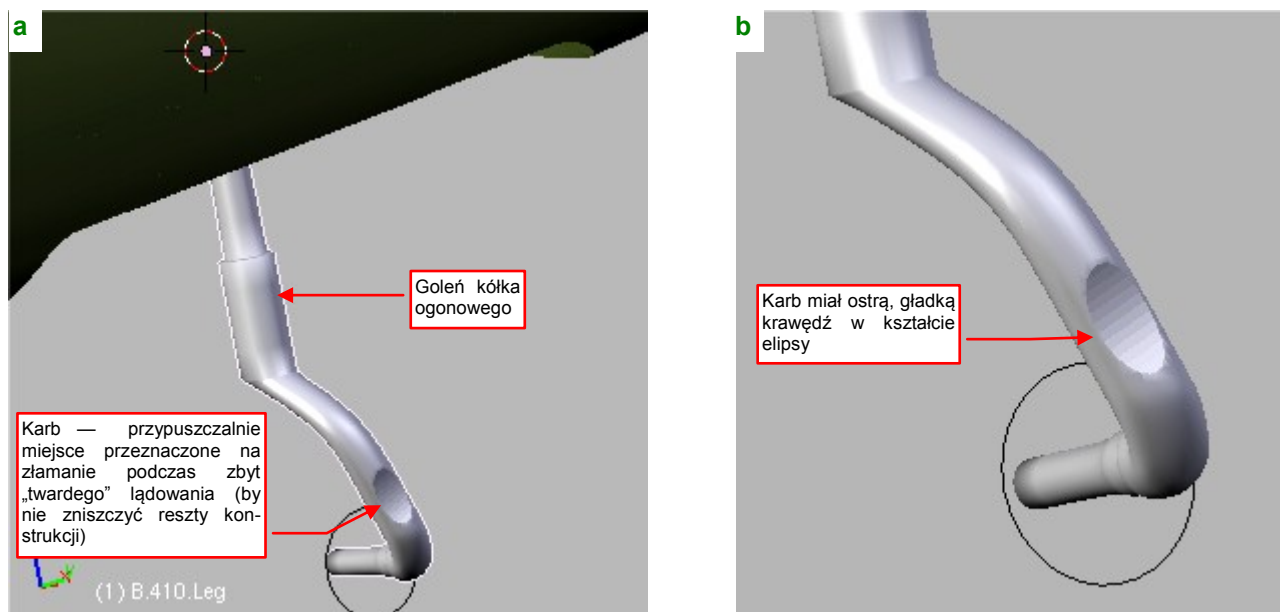
W nagłówku okna **UV/Image Editor** znajdziesz pomocniczą listę rozwijalną do wyboru aktywnej mapy UV (Rysunek 8.6.5):



Rysunek 8.6.5 Lista wyboru aktywnej mapy UV w oknie **UV/Image Editor**

## 8.7 Operacje na bryłach (**Boolean**)

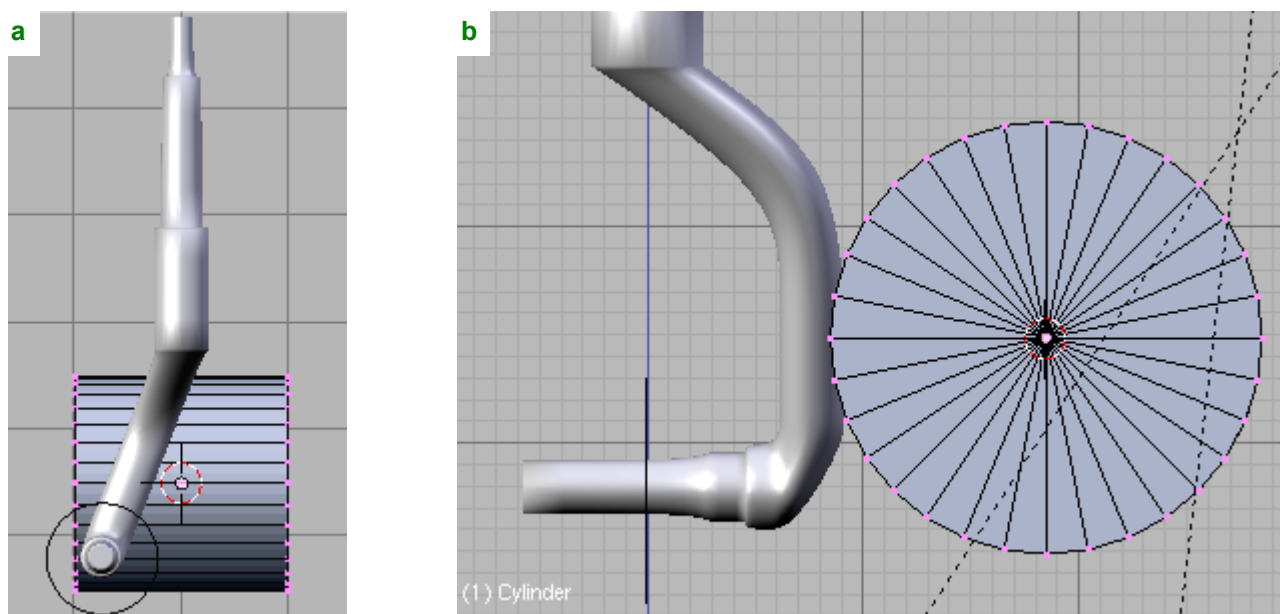
Podczas wykonywania detali, możesz się natknąć na takie drobiazgi, jak nacięcie (karb) na goleni kółka ogonowego (Rysunek 8.7.1a):



Rysunek 8.7.1 „Karb” na goleni kółka ogonowego

Jak szybko uzyskać taki szczegół? Gdybyśmy chcieli wykonać taki karb w ten sam sposób, co otwory w kadłubie — byłoby to bardzo pracochłonne. Trzeba by było wcześniej odpowiednio przygotować poprzeczny kontur goleni, z „ostrymi” (*crease* =1) krawędziami w pobliżu granicy zagłębienia (por. Wyciąg II). W efekcie otrzymalibyśmy coś, co i tak nie miałoby takiej ładnej, ostrej krawędzi jak zagłębienie, które pokazuje Rysunek 8.7.1b). W takim przypadku bardziej opłaca się zastosować modyfikator **Boolean**.

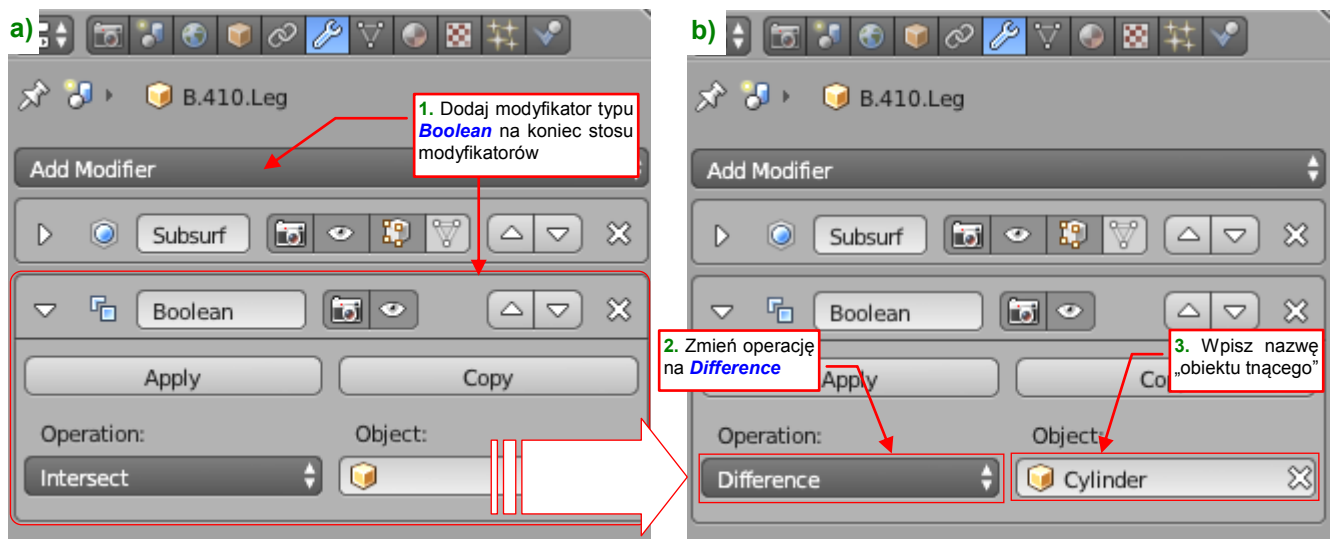
Karb wygląda jak wycięty fragmentem walca, więc wstawmy do rysunku cylinder o dość dużym promieniu (Rysunek 8.7.2a):



Rysunek 8.7.2 „Narzędzie tnące” — walec (**Cylinder**)

Rysunek 8.7.2b) pokazuje, że boczna ściana walca jest nieznacznie zagłębiona w goleni — na tyle, ile ma wynosić głębokość karbu. Na razie pozostawiłem temu obiektowi nazwę **Cylinder**. To nasze „narzędzie tnące”.

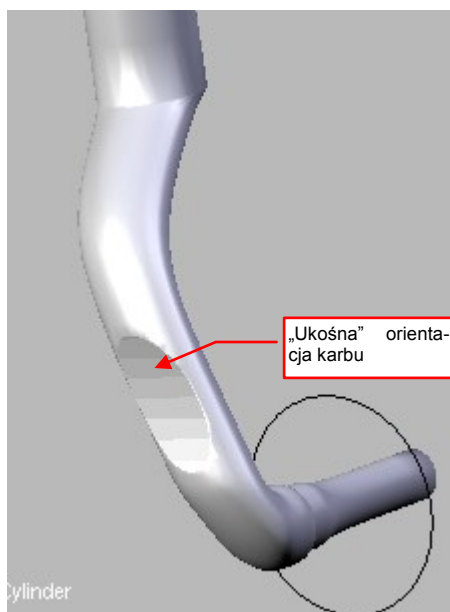
Teraz przełącz się w tryb edycji gołeni (**B.410.Leg**), a potem w zestawie **Modifiers** naciśnij przycisk **Add Modifier** i wybierz z listy **Boolean**. Rysunek 8.7.3a) pokazuje dodany modyfikator:



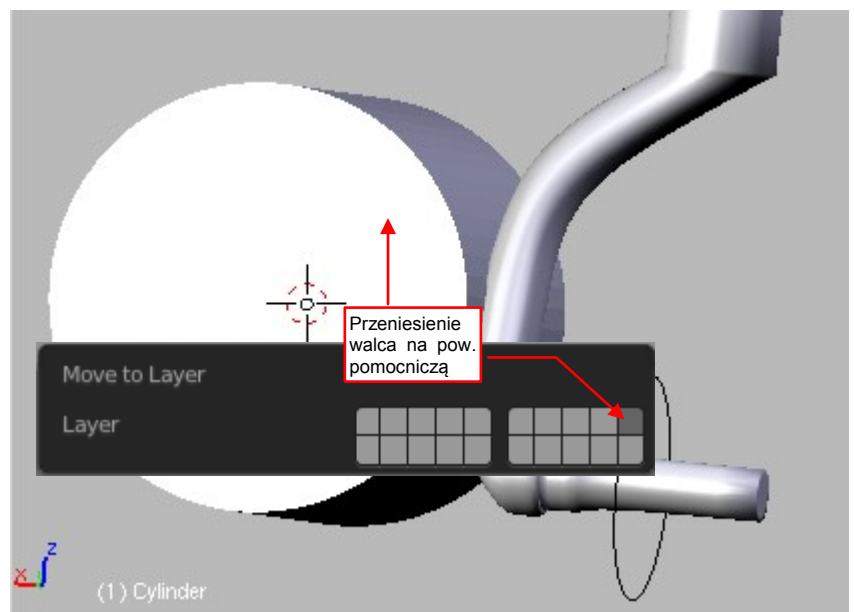
Rysunek 8.7.3 Dodanie do siatki gołeni modyfikatora typu **Boolean**

W panelu modyfikatora zmień rodzaj operacji na **Difference**, i wpisz nazwę "narzędzia tnącego" — cylindra, który przed chwilą przygotowaliśmy (Rysunek 8.7.3b).

Rysunek 8.7.4 przedstawia rezultat operacji po ukryciu (**H**) walca:



Rysunek 8.7.4 Rezultat operacji **Boolean**



Rysunek 8.7.5 Przesunięcie walca na niewidoczną podczas renderowania warstwę pomocniczą

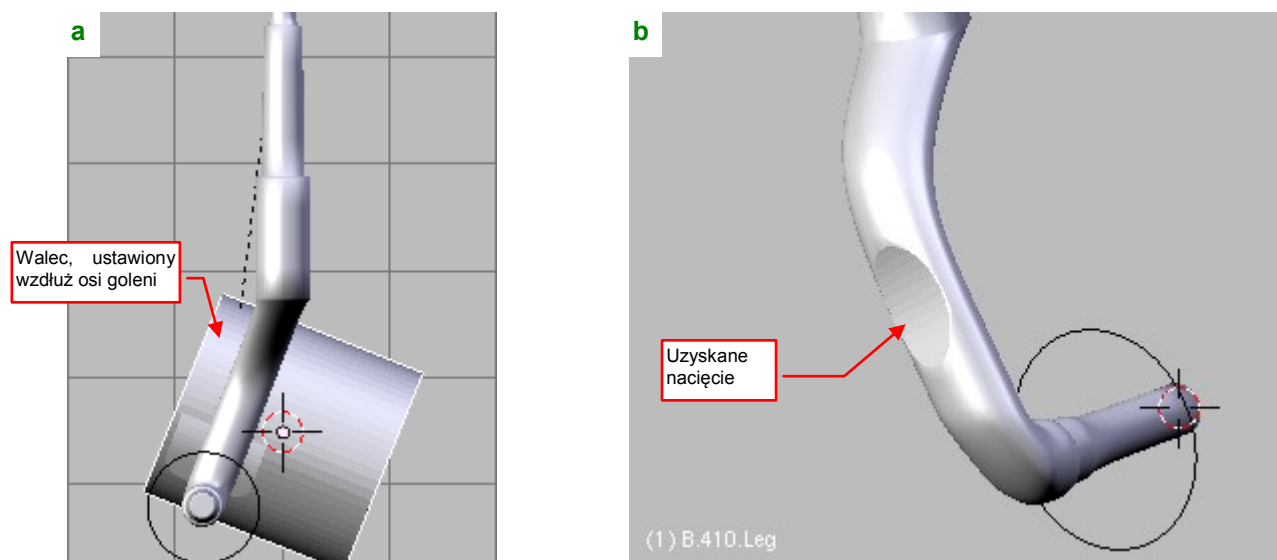
Uzyskany karb ma ładną, ostrą krawędź. Na ostatecznym renderze nie będzie widać lekkich załamań ścianek wewnątrz zagłębienia.

- Operacje typu **Boolean** w Blenderze nie są jeszcze dopracowane. Czasami w wyniku użycia tego modyfikatora możesz uzyskać dodatkowe ściany, których nie powinno być. Zmień wtedy układ siatki, poprzez dodanie jednej — dwóch nowych krawędzi, i spróbuj ponownie.

Oczywiście, nie chcemy widzieć walca na ostatecznym renderingu. Przesuń go na warstwę **10** (**M**), Rysunek 8.7.5). (Warstwę **10** przeznaczaliśmy na różne pomocnicze obiekty — por. Wyciąg II).

Zaletą operacji **Boolean**, implementowanych jako modyfikator, jest dynamiczne połączenie obydwu brył — „narzędzia” i „obiektu”. Dzięki temu nie komplikujesz oryginalnych siatek i nadal możesz łatwo zmieniać ich kształt podczas edycji. W przypadku wygładzonych powierzchni, krawędź przecięcia jest wyznaczana na ścianach powierzchni podziałowych. Dzięki temu nie zaburzają kształtu reszty powłoki.

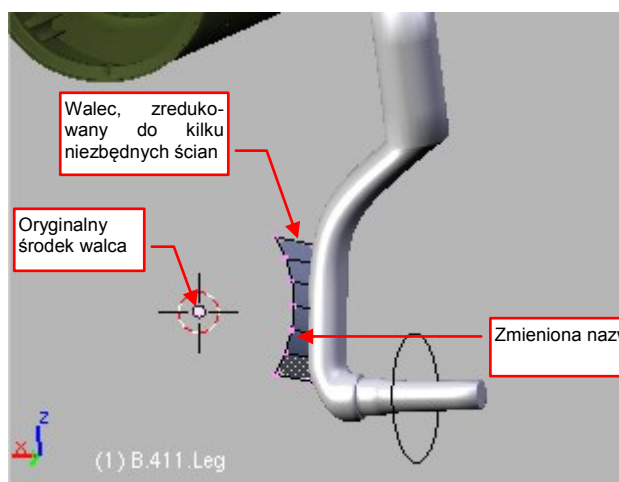
Na razie poprawmy trochę krawędzie karbu: powinny tworzyć elipsę równoległą do osi goleni, więc obróćmy odpowiednio nasz cylinder (Rysunek 8.7.6a):



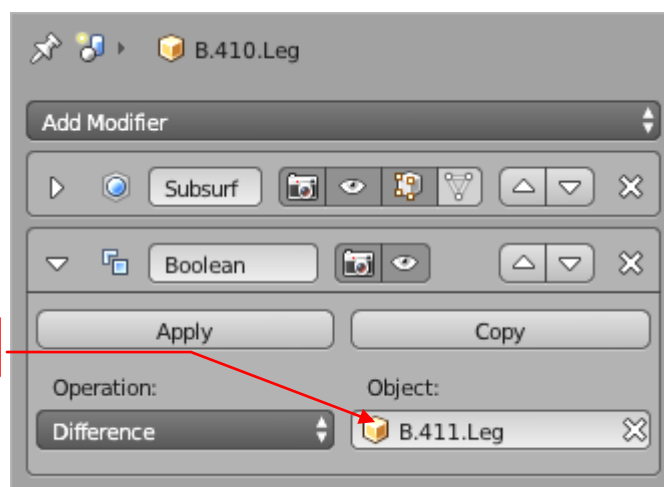
Rysunek 8.7.6 Modyfikacja kształtu nacięcia

W rezultacie uzyskaliśmy pożądany kształt (Rysunek 8.7.6b). Zwróć jednak uwagę, że teraz walec obracał się dość „opornie”. Wadą zastosowania modyfikatora **Boolean** jest wolniejsze odświeżanie ekranu podczas przesuwania, obrotu lub innych zmian każdego „składnika” tej operacji. Przy każdym Twoim ruchu wykonywane są ponownie obliczenia przenikania obydwu brył. Na szczęście na finalny czas renderingu to nie ma wpływu. Być może zdecydujesz się wyłączać podczas co intensywniejszych edycji działanie tego modyfikatora. (Wystarczy usunąć z jego parametrów nazwę obiektu „tnącego”, a po zakończeniu zmian — wpisać powtórnie).

Do uzyskania poprawnego „wycięcia” nie jest właściwie potrzebny cały walec — wystarczy fragment jego ściany (Rysunek 8.7.7):



Rysunek 8.7.7 Walec „tnący”, zredukowany do niezbędnych ścian

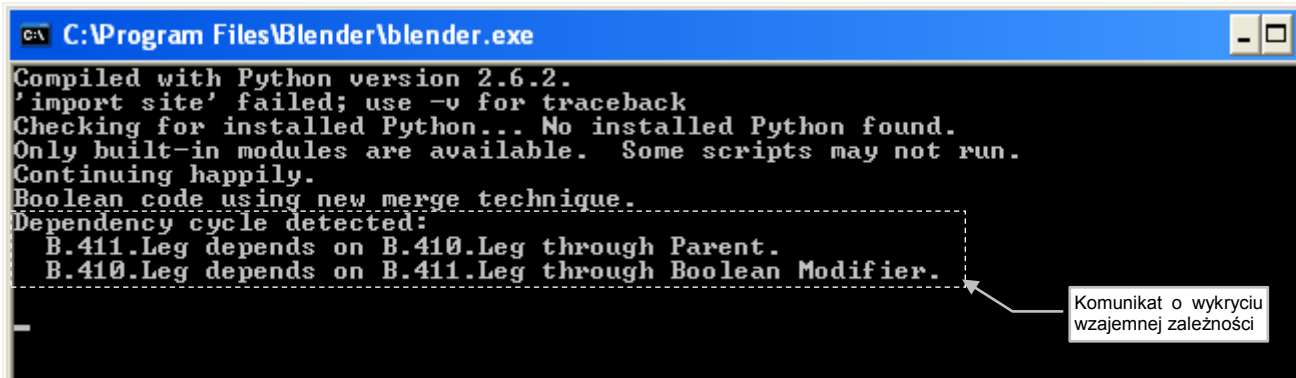


Rysunek 8.7.8 Uaktualnione parametry modyfikatora

Nadaj temu, co zostało z oryginalnego cylindra jakąś nazwę porządkową (w tym przykładzie — **B.411.Leg**). Zwróć uwagę, że odpowiedniej zmianie uległ także wpis w panelu modyfikatora (Rysunek 8.7.8). Aby karb nie „uciekał” podczas ruchu goleni (będziemy ją animować), przypisz ten obiekt do goleni (**Make Parent** — **Ctrl-P**).



W wyniku dołączenia obiektu **B.411.Leg** do goleni (**B.410.Leg**), w konsoli Blendera zacznie się pojawiać komunikat ostrzegawczy (Rysunek 8.7.9):



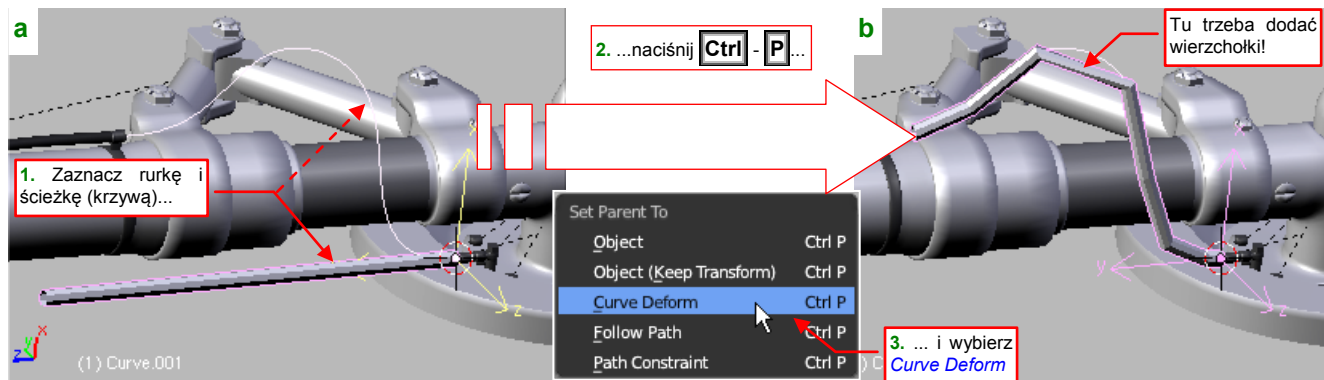
Rysunek 8.7.9 Komunikaty o wzajemnym powiązaniu goleni (**B.410.Leg**) i ściany „tnącej” (**B.411.Leg**)

To trochę „alergiczna” reakcja: **B.411.Leg** wpływa tylko na kształt siatki goleni, w żaden sposób nie zmienia położenia i orientacji obiektu **B.410.Leg**, który jest jego „rodzicem”. Nie ma się czym przejmować!

Oczywiście, gdy będziesz pewien że tego elementu nie będziesz już zmieniał, możesz „utrwalić” obydwie modyfikatory (*Subsurf* i *Boolean*), przypisane do siatki goleni (**B.410.Leg**), i wyrzucić z rysunku niepotrzebny już pomocniczy **B.411.Leg**. Zrób to jednak na sam koniec pracy, bo to jest w sumie krok nieodwracalny. Po „utrwaleniu” tych modyfikatorów, w przypadku konieczności poważniejszej zmiany kształtu, najprościej będzie stworzyć taką goleń od nowa.

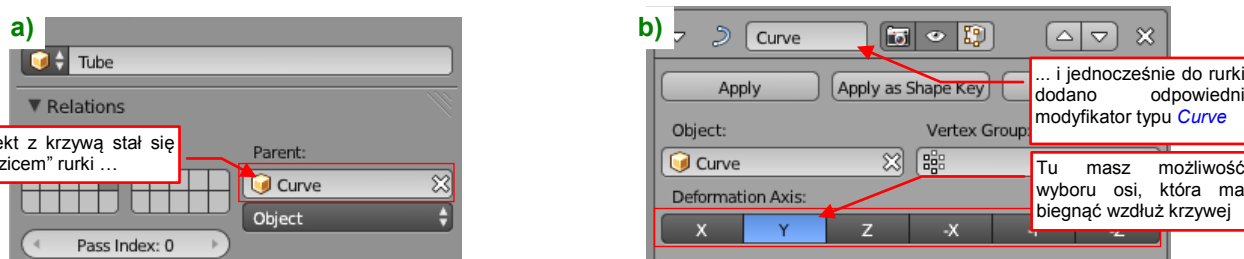
## 8.8 Wyginanie wzdłuż krzywej (*Curve*)

Zazwyczaj wzdłuż krzywej są trzeba wygiąć jakieś cylindry — aby w ten sposób stworzyć imitację jakiegoś przewodu, kabla, lub rury. W najprostszej wersji polega to na wskazaniu cylindra, krzywej (*Path* — por. str. 260), i naciśnięciu **Ctrl-P** (Rysunek 8.8.1a):



Rysunek 8.8.1 Wygięcie rury (*Cylinder*) wzdłuż krzywej (*Path*)

W odpowiedzi Blender otworzy menu *Set Parent To*. Wybierz z niego opcję *Curve Deform* (Rysunek 8.8.1b). To polecenie przypisało obiekt zawierający krzywą jako „rodzica” (*Parent*) rury (Rysunek 8.8.2a):

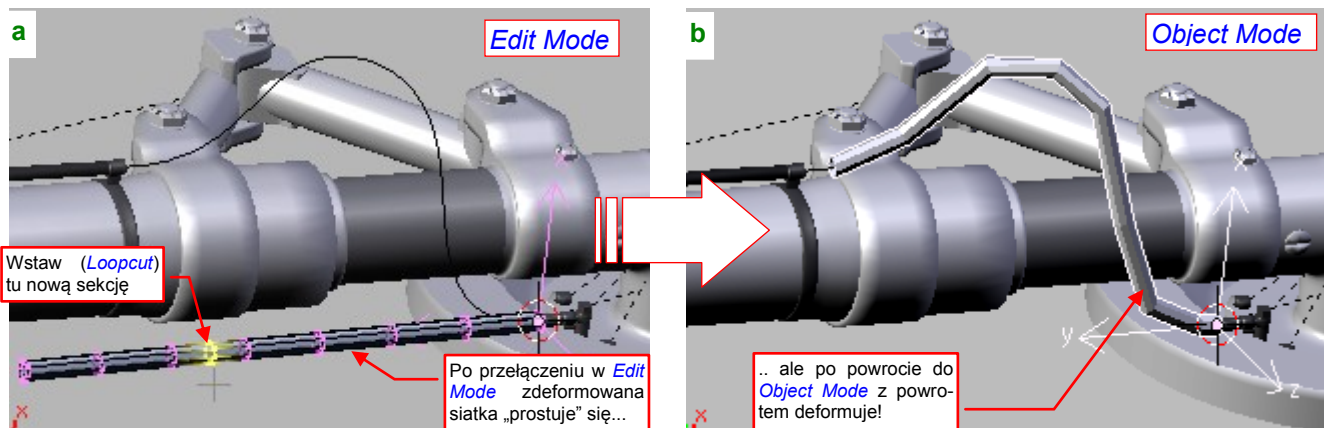


Rysunek 8.8.2 Efekt działania polecenia *Set Parent To* → *Curve Deform*

Jednocześnie rura otrzymała modyfikator *Curve*, przypisany do osi **X** (trzeba to zmienić — Rysunek 8.8.2b).

- Przed operacją warto dopilnować, by środki obydwu obiektów — cylindra i krzywej — leżały w tym samym miejscu. Najlepiej aby osie **X** ich lokalnych układów współrzędnych pokrywały się ze sobą.

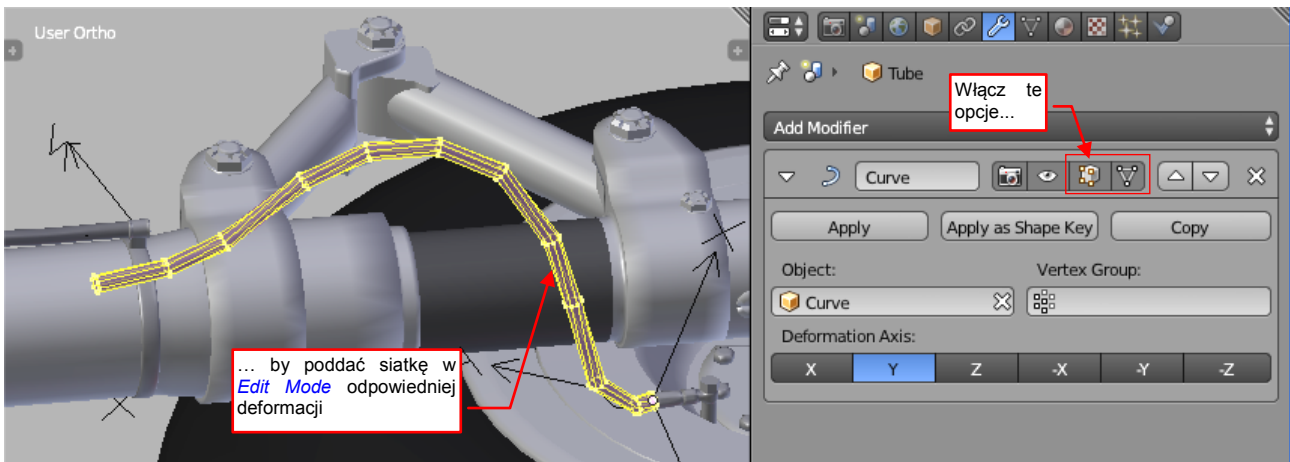
Powłóce, którą pokazuje Rysunek 8.8.1b), wyraźnie przydałby się dodatkowy segment. Gdy przejdiesz w tryb edycji wygiętej siatki — ta natychmiast się wyprostuje (Rysunek 8.8.3a):



Rysunek 8.8.3 Edycja siatki wygiętej rury

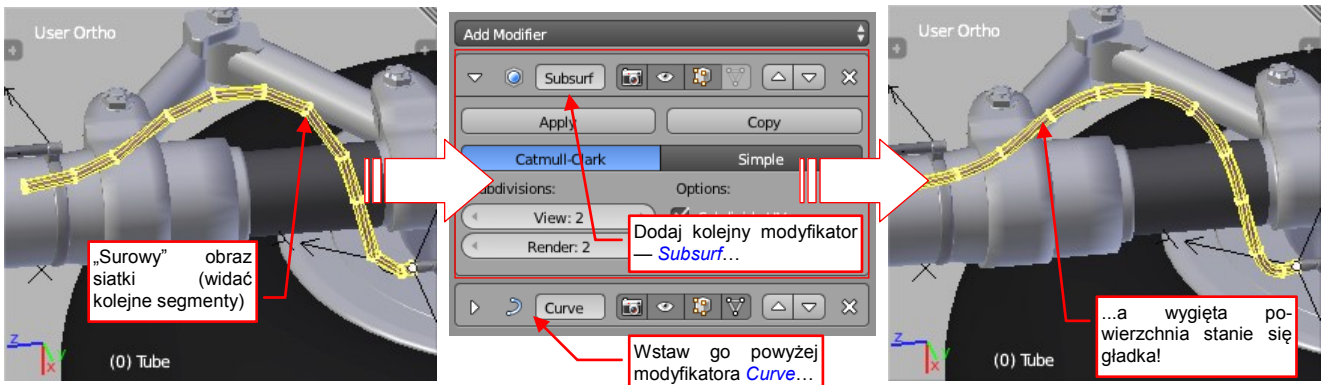
To może utrudnić dostosowanie gęstości siatki do promienia krzywizny. Tym niemniej gdy przełączysz się z powrotem w *Object Mode* — rura znów stanie się wygięta (Rysunek 8.8.3b).

W panelu **Modifiers** możesz skorzystać z przełącznika **Apply during Edit**, umieszczonego w „nagłówku” każdego modyfikatora (Rysunek 8.8.4). Gdy go włączysz, obraz wygiętej siatki pojawi się także podczas edycji:



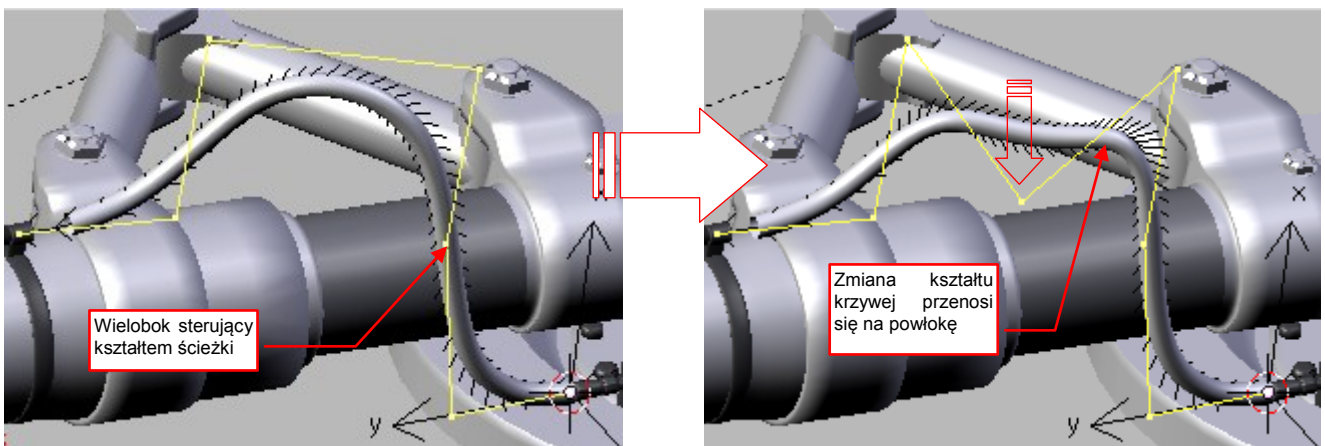
Rysunek 8.8.4 Włączenie deformacji podczas edycji siatki

Po odpowiednim zagęszczeniu siatki „na zakrętach”, można dodać kolejny modyfikator — **Subdivision Surface** — by ją dodatkowo wygładzić (Rysunek 8.8.5):



Rysunek 8.8.5 Dodanie kolejnego modyfikatora — wygładzania

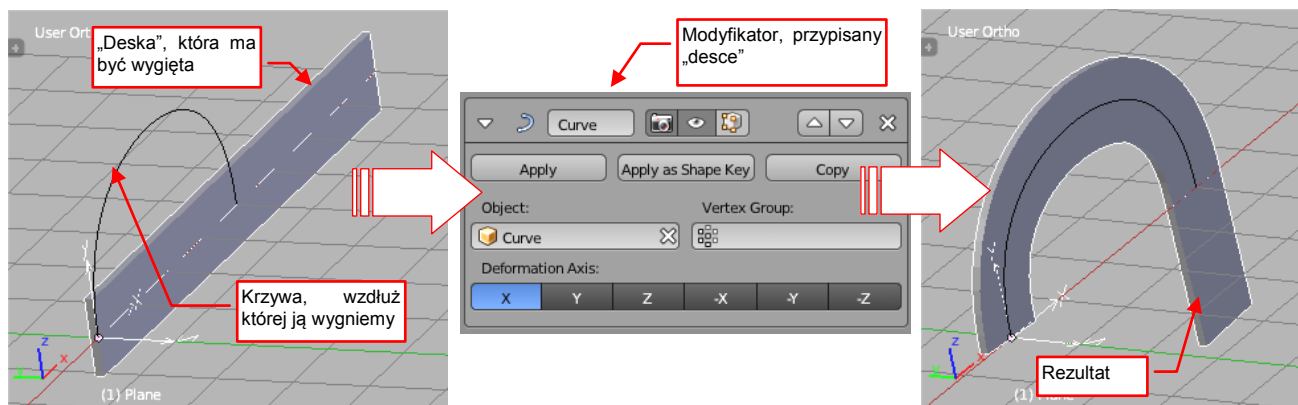
Modyfikator **Curve** można traktować jako specyficzną formę relacji hierarchicznej (**Parent** — **Child**). Jakkolwiek nie obrócimy lub przesuniemy krzywą, wygiętą wzdłuż jej linii rura podąży za nią. Co więcej, zmiana kształtu krzywej zmienia odpowiednio kształt wygiętej powłoki (Rysunek 8.8.6):



Rysunek 8.8.6 Kształt krzywej steruje kształtem rury

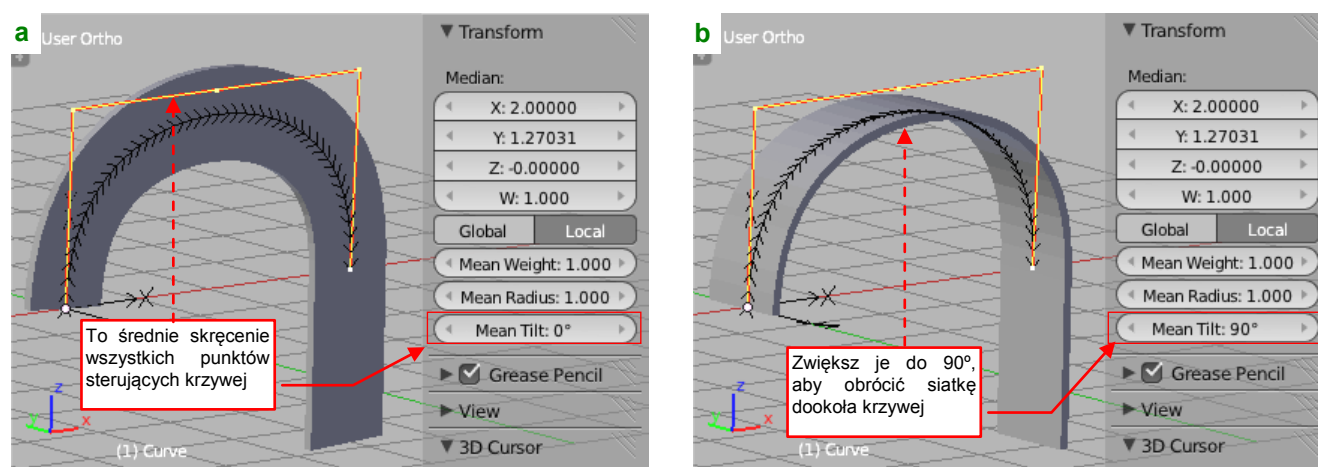
Kształt krzywej zmieniamy, przesuwając (w trybie edycji) jej punkty sterujące (por. str. 261). Gdy zaczniesz to robić, zauważysz że wygięty cylinder „podąża” za deformowaną linią. Takie dynamiczne zmiany, zachodzące w trakcie edycji, bardzo ułatwiają dopasowania zespołu do reszty modelu.

Gdy wyginamy taką „deskę” jaką pokazuje Rysunek 8.8.7, ważny staje się kierunek skręcenia poprzecznego:



Rysunek 8.8.7 Inny przypadek: płaska „deska” i jej krzywa

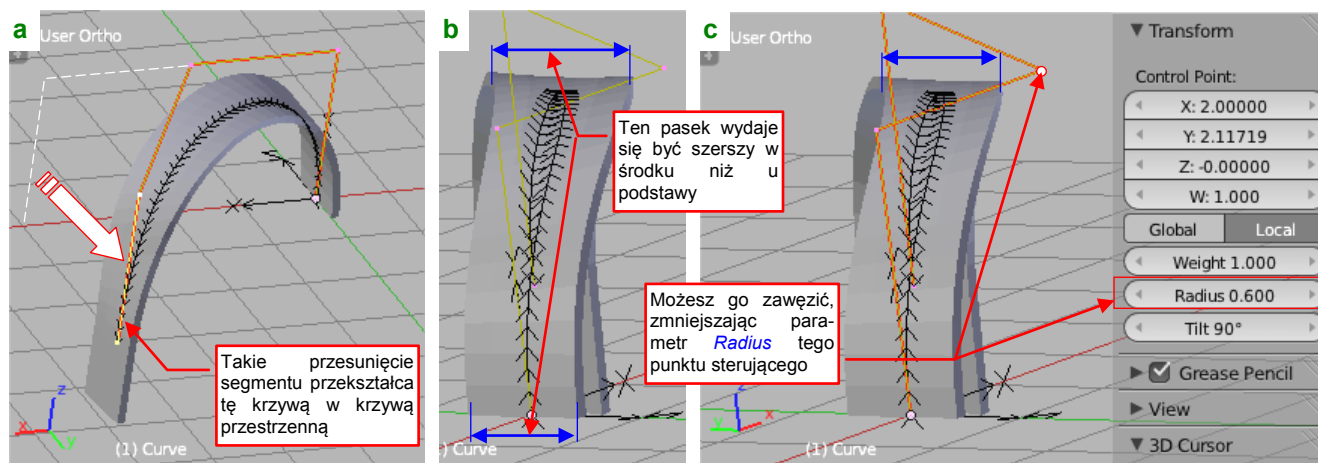
Aby wygiąć tę deskę jak „wstążkę”, musimy zmienić orientację krzywej (parametr **Tilt** — Rysunek 8.8.8a):



Rysunek 8.8.8 Zmiana skręcenia krzywej (**Tilt**)

Zaznacz krzywą i przełącz się w tryb **Edit Mode**. Zaznacz wszystkie punkty sterujące krzywej (Rysunek 8.8.8a), a następnie zwiększ ich wartość **Tilt** o 90° (Rysunek 8.8.8b).

Inny parametr punktów sterujących — **Radius** — można użyć do zmiany szerokości siatki. Na przykład — w wyniku wygięcia krzywej w przestrzeni (Rysunek 8.8.9a) można odnieść wrażenie, że jej środkowa część jest szersza niż brzegi (Rysunek 8.8.9b):



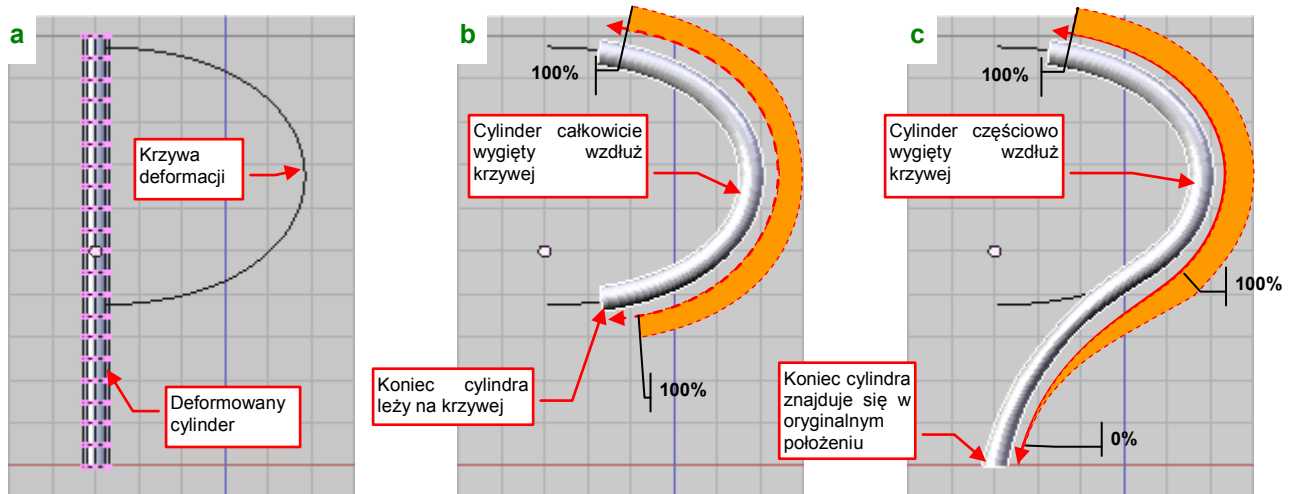
Rysunek 8.8.9 Zawężanie lokalne paska za pomocą parametru **Radius**

Aby zrównoważyć ten efekt, wystarczy zmniejszyć **Radius** środkowego punktu sterującego (Rysunek 8.8.9c).



## 8.9 Malowanie wag (*Weight paint*)

W poprzedniej sekcji (str. 288) pokazałem, jak wygiąć siatkę cylindra wzdłuż linii krzywej. Przekształcaliśmy prostą „rurę” (Rysunek 8.9.1a) w zakrzywioną powłokę, jaką pokazuje Rysunek 8.9.1b):

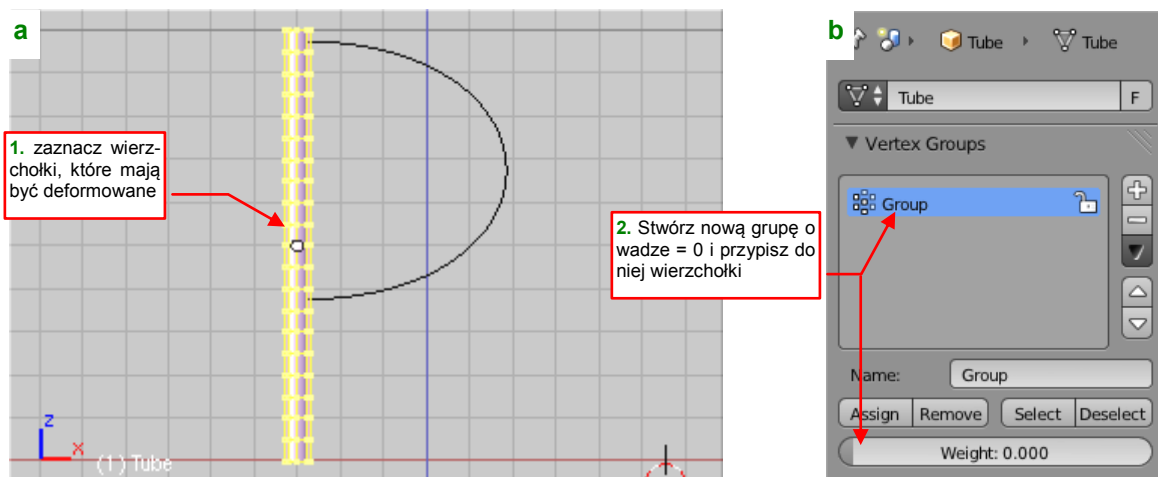


Rysunek 8.9.1 Cylinder wygięty wzdłuż krzywej: całkowicie (b) i częściowo (c)

Czasami potrzebna jest jednak taka deformacja, gdzie tylko część siatki ma się „ułożyć” wzdłuż krzywej, a reszta ma płynnie przechodzić we fragment bez zniekształceń (Rysunek 8.9.1c). Jak taki efekt uzyskać?

W Blenderze służą do tego wagi (*weight*). Waga to przypisana do każdego wierzchołka siatki liczba z zakresu od 1.0 do 0.0. Określa wpływ modyfikatorów (takich, jak *Curve*) na jego położenie. Na przykład wszystkie wierzchołki powłoki, którą pokazuje Rysunek 8.9.1b), mają *Weight* = 1.0 (100% — pełna deformacja). Za to wierzchołki końcówki cylindra, który pokazuje Rysunek 8.9.1c), mają *Weight* = 0.0 (0% — brak deformacji). Punkty leżące na „odchodzącym” od krzywej odcinku cylindra mają odpowiednie wartości pośrednie.

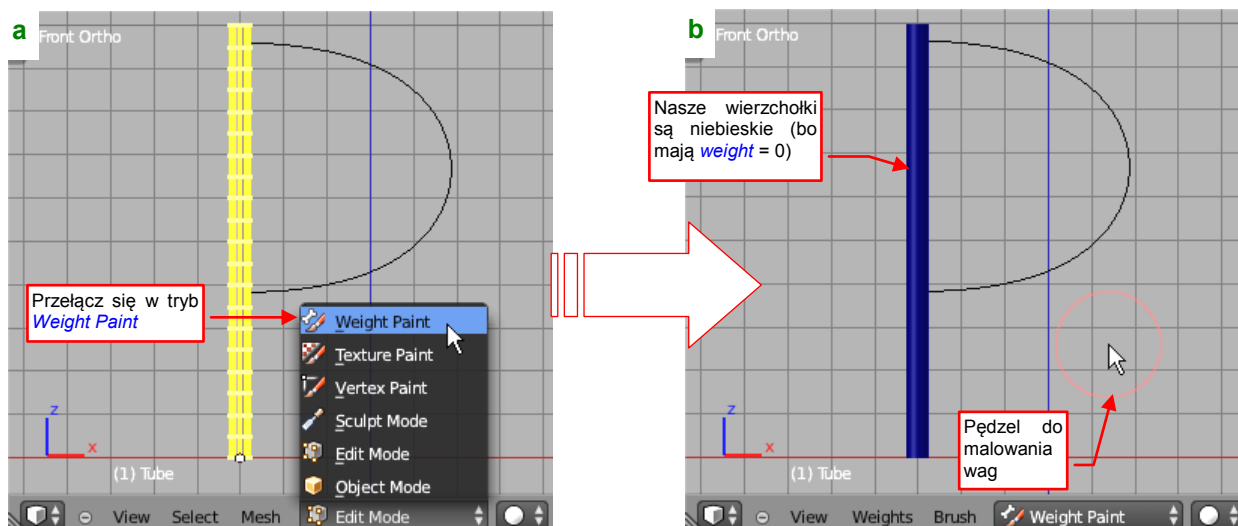
Jak nanosi się wagi na siatkę? Najpierw musisz zaznaczyć obszar, którego w ogóle ma dotyczyć deformacja (Rysunek 8.9.2a):



Rysunek 8.9.2 Zdefiniowanie grupy wierzchołków o początkowej wadze = 0

Następnie stwórz nową grupę wierzchołków (w panelu *Vertex Groups* zestawu *Object Data* — por. str. 278). W polu *Weight* wpisz początkową wartość wagi (Rysunek 8.9.2b). Na koniec naciśnij przycisk *Assign*, by przypisać zaznaczone na siatce wierzchołki do tej grupy. Na potrzeby tego przykładu nie będę jej zmieniał nazwy — będzie się nazywać *Group*. (W modelu zmieniaj je raczej na coś znaczącego. Nazwy grup wierzchołków muszą być unikalne tylko w obrębie tej samej siatki. Mogą się powtarzać w różnych powłokach, więc zazwyczaj nie ma problemu z ich doбором).

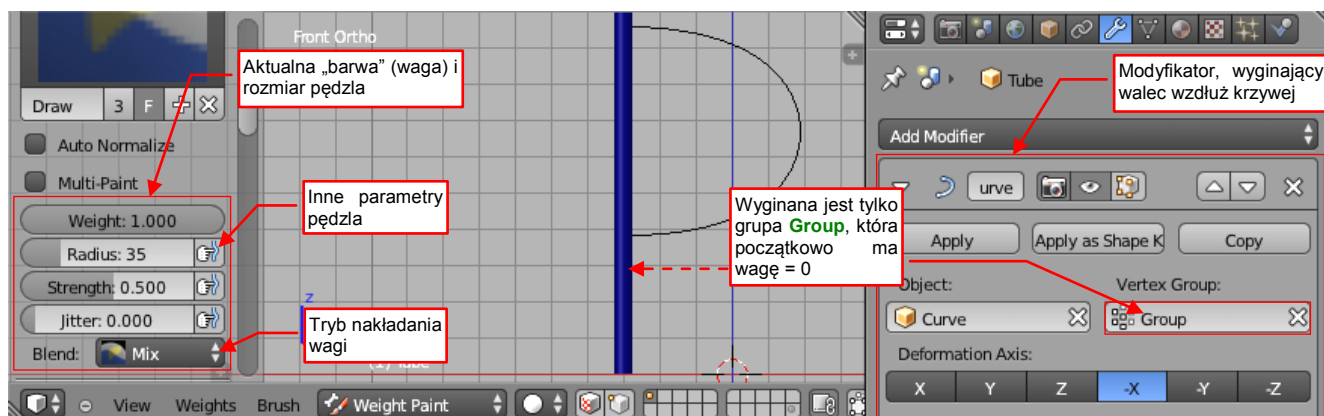
W Blenderze wagi wierzchołków się „maluje”. Zmień tryb pracy z *Edit Mode* na *Weight Paint* (Rysunek 8.9.3):



Rysunek 8.9.3 Przelączenie w tryb *Weight Paint*

Spowoduje to, rzecz jasna, zmianę menu w nagłówku (a właściwie stopce) okna — ze starych pozostanie na nim tylko menu *View*, i pojawi się nowe: *Paint*. Jednocześnie wokół kursora myszy pojawi się okrąg „pędzla”, a wierzchołki siatki staną się niebieskie (jest to kolor odpowiadający *Weight* = 0.0, który przypisaliśmy grupie).

Nim zaczniemy malować wagi, należy odpowiednio przygotować „środowisko pracy” (Rysunek 8.9.4):



Rysunek 8.9.4 Przygotowanie „środowiska pracy” przed malowaniem wag na siatce cylindra

W zestawie *Modifiers* dodaj do siatki (o ile nie zrobiłeś tego wcześniej) modyfikator *Curve*. W polu *Vertex Group* wpisz nazwę odpowiedniej grupy wierzchołków (w naszym przypadku to *Group*).

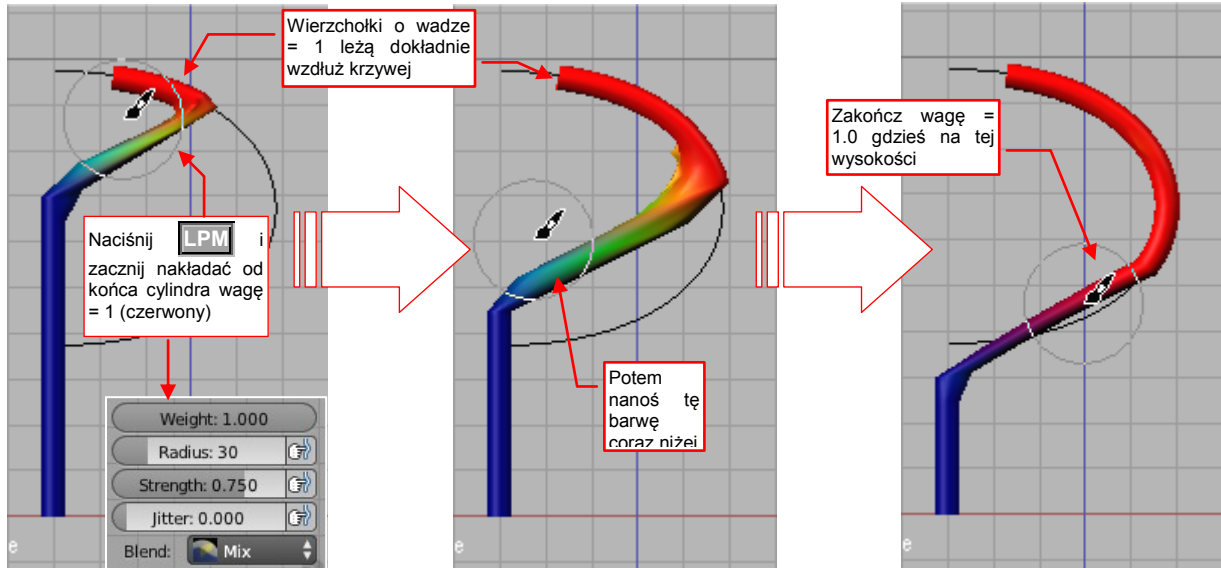
W zestawie *Object Data*: upewnij się (w panelu *Vertex Groups*), że aktualnie wybrana grupa jest grupą, której wagi chcesz zmieniać.

- Każda siatka może mieć wiele alternatywnych grup wierzchołków (a każdy wierzchołek może należeć jednocześnie do wielu różnych grup).
- Każda grupa wierzchołków ma swoje własne wagi (więc ten sam wierzchołek siatki może mieć różne wagi — w zależności od grupy, na którą się powołasz).

W trybie *Weight Paint* w przyborniku (*Tool Shelf* — **T**) pojawiają się panele związane z malowaniem (m.in. *Brush* — Rysunek 8.9.4). To „paleta” do malowania wag. *Brush* zawiera zestaw kontrolki, za pomocą których ustalisz aktualną wartość, którą będziesz nanosił na wierzchołki (*Weight*), rozmiar „pędzla” (*Radius*), oraz jego siłę (*Strength*). Ten ostatni parametr jest uwzględniany podczas „mieszania” wartości, którą malujesz, z wartością wagi, którą już posiada „malowany” wierzchołek. U dołu tego panelu możesz wybrać jeden z trybów tego mieszania (najczęściej używany jest domyślny *Mix*).

Waga 1.0 (100%) jest oznaczana kolorem czerwonym, a waga 0.0 — kolorem niebieskim. Wartościom pośrednim odpowiada spektrum pomiędzy tymi barwami. Powierzchnie z naniesionymi wagami mienia się więc w Blenderze wszystkimi kolorami tęczy.

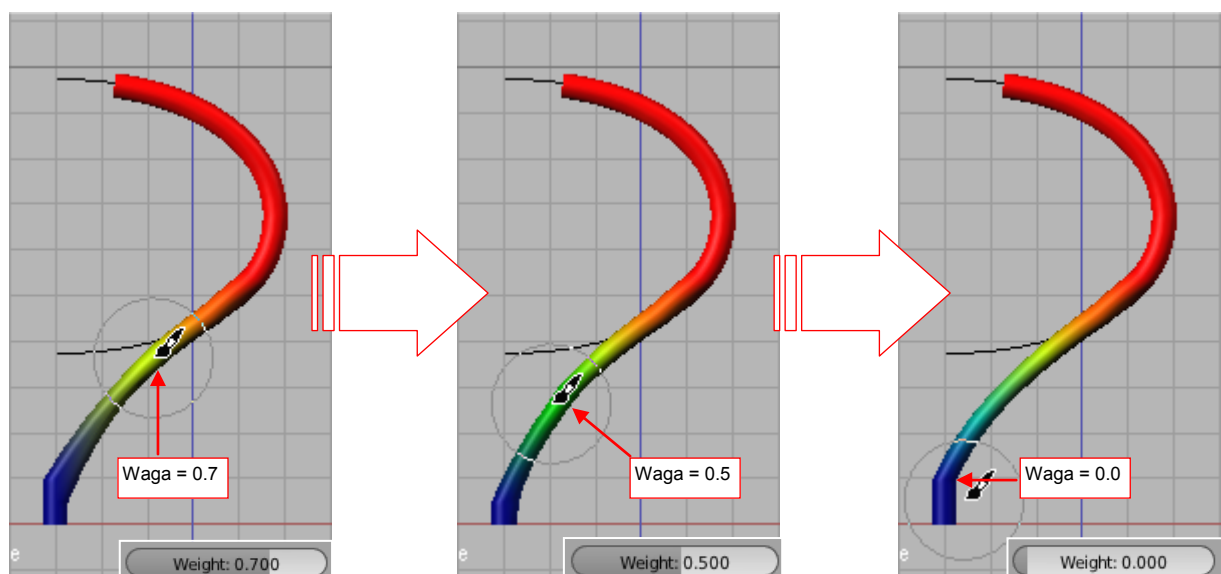
Ustaw najpierw w panelu **Brush** aktualną wagę (**Weight**) na 1.0 i umiarkowany rozmiar pędzla (**Radius**) — 30. Malujesz, naciskając **LPM** ponad siatką. Zaczynij nanosić wagę 100% od końca cylindra (Rysunek 8.9.5):



Rysunek 8.9.5 „Namalowanie” obszaru całkowicie wygiętego

Pomalowane wagą „100%” fragmenty siatki natychmiast „dokleją” się do krzywej, przypisanej przez modyfikator. Podczas malowania staraj się nie robić zbyt gwałtownych ruchów, by „trafić” z odpowiednią wagą na wszystkie ściany.

Po pomalowaniu „na czerwono” mniej więcej 60% długości cylindra, zacznij zmniejszać aktualną wagę co 0.1 (10%). Zbliżając się do drugiego końca powłoki, nanoś coraz mniejsze wartości (Rysunek 8.9.6):

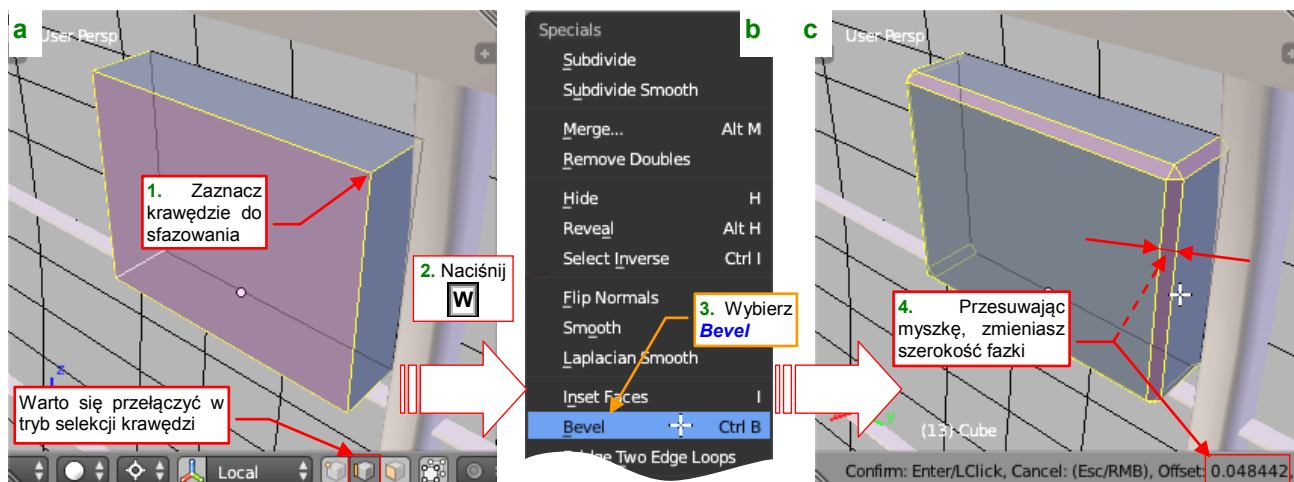


Rysunek 8.9.6 Namalowanie obszaru przejściowego

Spowoduje to pojawienie się na powłoce odcieni żółci, zieleni (waga = 50%), a potem błękitu. Pośrednie wagi nanoś w ten sposób, by na końcu siatki pozostawić wagę = 0 (kolor niebieski). W ten sposób uzyskasz efekt, który pokazywał Rysunek 8.9.1c) (str. 291).

## 8.10 Fazowanie i zaokrąglanie krawędzi (*Bevel*)

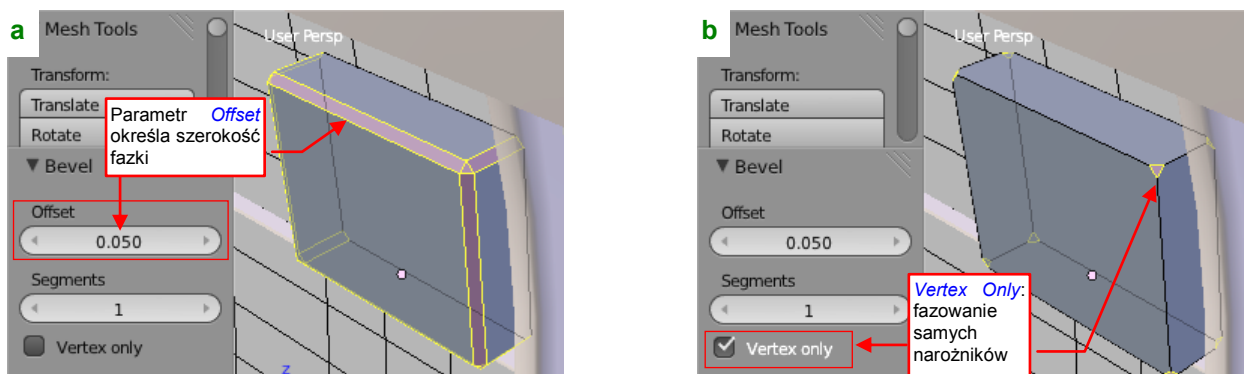
Najpierw zaznacz na siatce krawędzie, które chcesz sfazować. Najwygodniej jest na ten moment przełączyć się z domyślnego trybu selekcji wierzchołków w tryb selekcji krawędzi (Rysunek 8.10.1a):



Rysunek 8.10.1 Fazowanie wybranych krawędzi (*Bevel*)

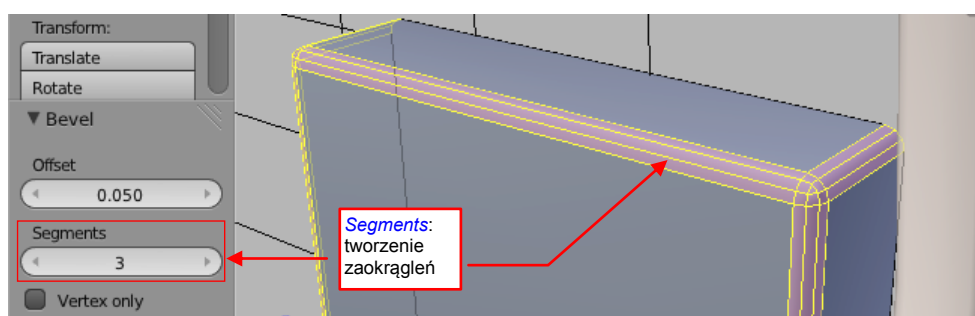
Następnie naciśnij klawisz **W**, i z menu *Specials* wybierz polecenie *Bevel* (Rysunek 8.10.1b). (To samo polecenie znajdziesz w menu: *Mesh* → *Edges* → *Bevel*). Wzdłuż zaznaczonych krawędzi pojawią się fazy, których szerokość możesz zmieniać dynamicznie, przesuwając myszkę (Rysunek 8.10.1c). Gdy ustawisz właściwą szerokość, potwierdzasz ją naciskając **LPM**. (Możesz też w każdej chwili z niej zrezygnować, naciskając **Esc**).

Klawisz **Ctrl** tu nie działa. Aby uzyskać jakąś zaokrągloną szerokość fazy, musisz zajrzeć do parametrów tego polecenia w przyborniku (**T**) i wpisać tam odpowiednią wartość (Rysunek 8.10.2a):



Rysunek 8.10.2 Parametry polecenia *Bevel*

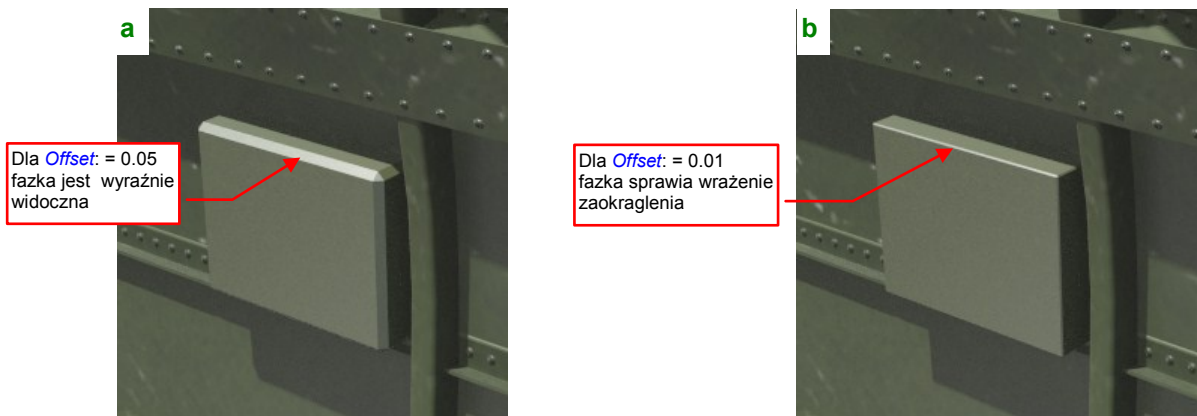
Gdy włączysz tryb *Vertex only*, uzyskasz zaokrąglenie samych narożników (Rysunek 8.10.2b). Gdy zwiększysz parametr *Segments*, uzyskasz zaokrąglenie zamiast fazy (Rysunek 8.10.3):



Rysunek 8.10.3 Użycie fazowania do zaokrąglania krawędzi siatki



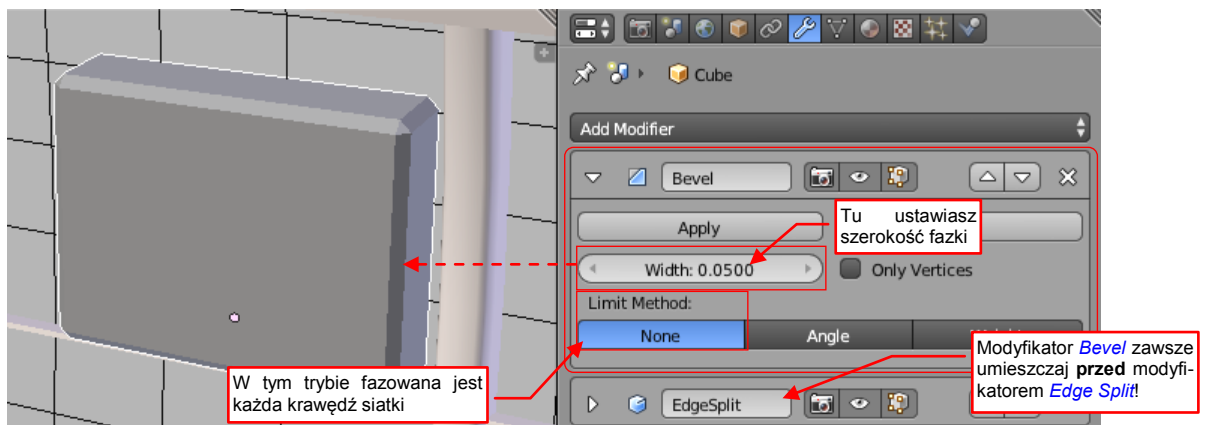
Zaokrąglanie za pomocą parametru *Segments* opłaca się stosować tylko w przypadku jakichś specjalnie dużych promieni. W Cycles wązka fazka także wygląda jak zaokrąglenie. Przekonaj się zresztą sam: Rysunek 8.10.4a) przedstawia „pudełko”, którego krawędzie sfazowałem na szerokość 0.05 jednostki. Widać na nim wyraźnie każde załamanie krawędzi:



Rysunek 8.10.4 Efekt różnej szerokości fazek na renderze

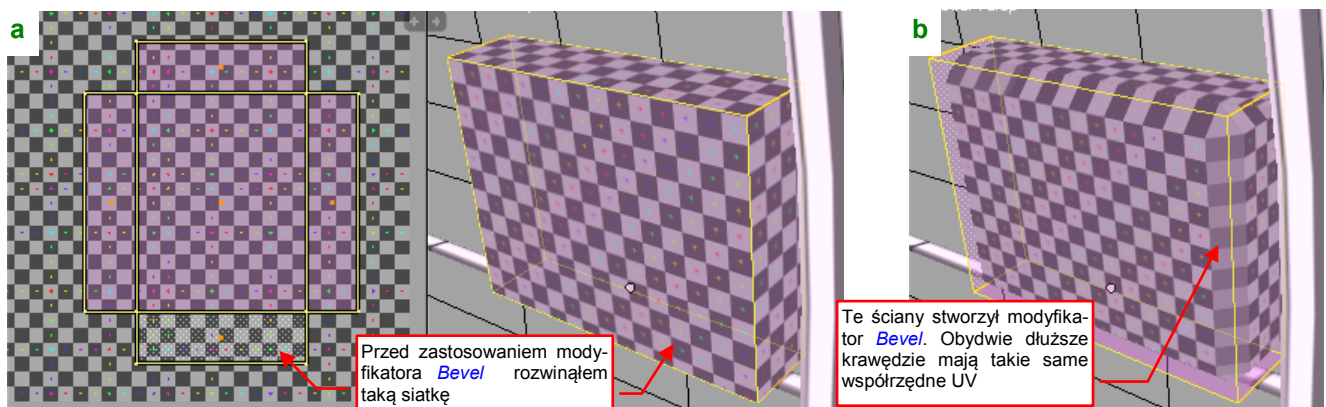
„Pudełko” przedstawione przez Rysunek 8.10.4b) także ma pojedynczą fazę, ale o szerokości 0.01 jednostki. Prawda że wygląda tak, jak gdyby miało zaokrąglone krawędzie?

Polecenie *Bevel* zmienia siatkę „na trwałe”, dodając nowe wierzchołki, ściany i krawędzie. Aby uzyskać efekt fazownia bez komplikacji siatki możesz użyć modyfikatora *Bevel* (Rysunek 8.10.5):



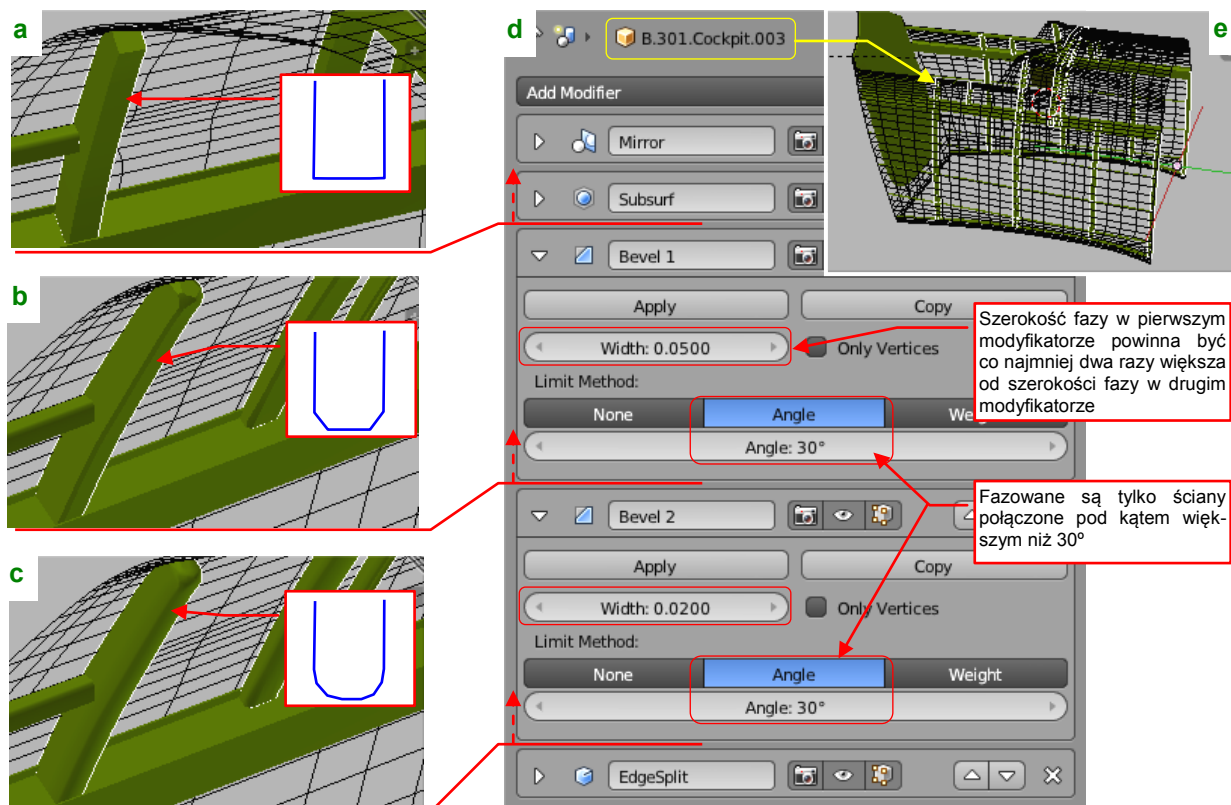
Rysunek 8.10.5 Modyfikator *Bevel*

W trybie *Limit Method:None* modyfikator fazuje wszystkie krawędzie obiektu. Jego zastosowanie pozwala zachować prostą topologię siatki, co np. ułatwia późniejsze rozwinięcia UV (Rysunek 8.10.6):



Rysunek 8.10.6 Wpływ modyfikatora *Bevel* na mapowanie UV

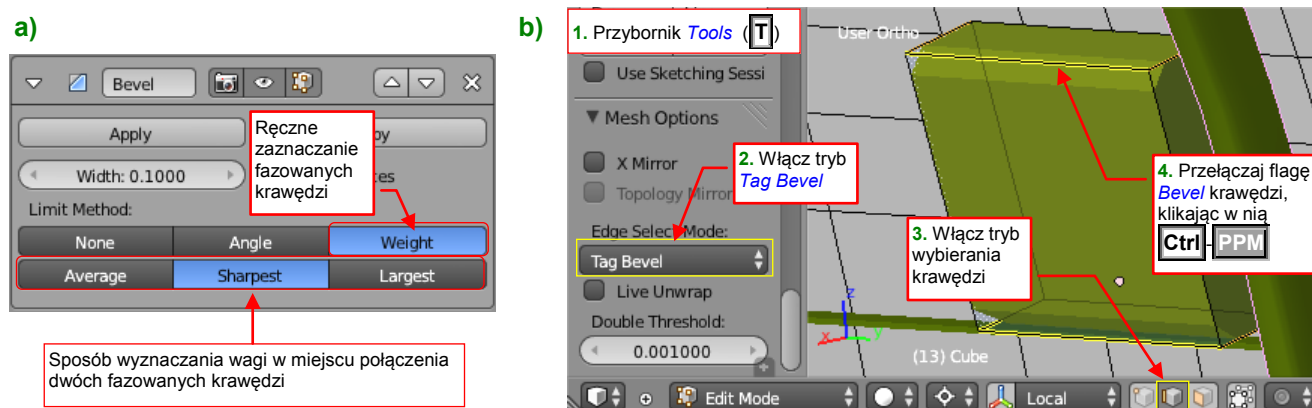
Modyfikator **Bevel** nie posiada co prawda parametru **Segments**, takiego jak polecenie **Bevel** (por. str. 294). Przybliżony efekt zaokrąglenia krawędzi można jednak osiągnąć poprzez połączenie dwóch lub więcej takich modyfikatorów. Zastosowałem je np. do uzyskania na wręgach kadłuba przekroju w kształcie „U” (Rysunek 8.10.7):



Rysunek 8.10.7 Przykład zastosowania modyfikatora **Bevel** do zaokrąglenia krawędzi wręg

Zwróć uwagę że obydwa modyfikatory pokazywane przez Rysunek 8.10.7d) mają ustawione **Limit Method: Angle** (30°). W ten sposób do zaokrąglenia wybrane są tylko „ostre” krawędzie siatki wygładzonej wcześniej modyfikatorem **Subsurf**.

Modyfikatorowi **Bevel** można także wskazać ręcznie krawędzie do fazowania. Służy do tego ostatni z trybów **Limit Method: Weight** (Rysunek 8.10.8a):



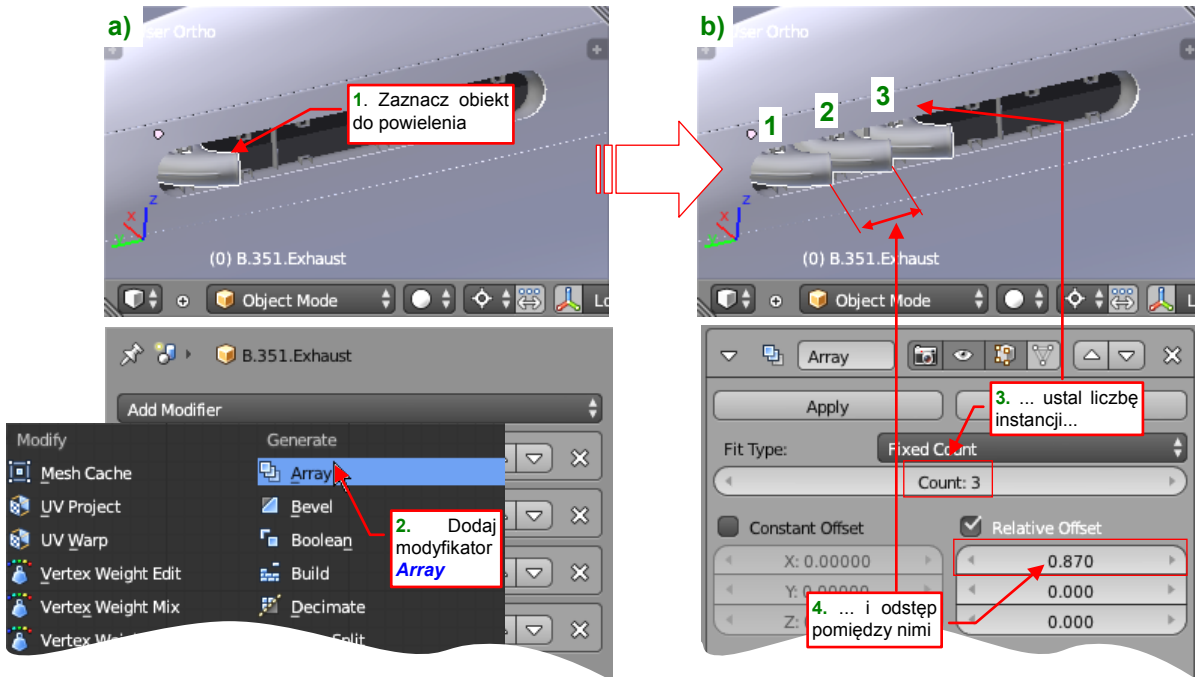
Rysunek 8.10.8 Ręczny wybór krawędzi do sfazowania przez modyfikator **Bevel**

Ustaw w panelu modyfikatora **Bevel** jeden ze trybów wyznaczania wag (np. **Average**, albo **Sharpest** — Rysunek 8.10.8a). Koniecznie przełącz w nagłówku okna **3D View** tryb selekcji siatki na krawędzie (**Edges** — Rysunek 8.10.8b). Potem otwórz przybornik (**T**) i w panelu **Mesh Options** wybierz z listy **Edge Select Mode** tryb **Tag Bevel**. Od tej chwili kliknięcie **Ctrl-PPM** w krawędź siatki oznacza ją jako fazkę. Powtórne takie kliknięcie wyłącza to oznaczenie.

### 8.11 Powielenie siatki (**Array**)

Siatkę można powielić poprzez sklonowanie obiektu, który ją zawiera (**Alt-D**, por. str. 219). Jeżeli jednak zależy Ci na szybkim rezultacie i równych odstępach pomiędzy poszczególnymi kopiami, użyj zamiast tego modyfikatora **Array**. Dla przykładu przekształcę za pomocą modyfikatorów pojedynczą rurę wydechową w pełen kolektor spalin. W oryginalnym samolocie cały ten zespół składał się z 12 takich rur, po sześć z każdej strony.

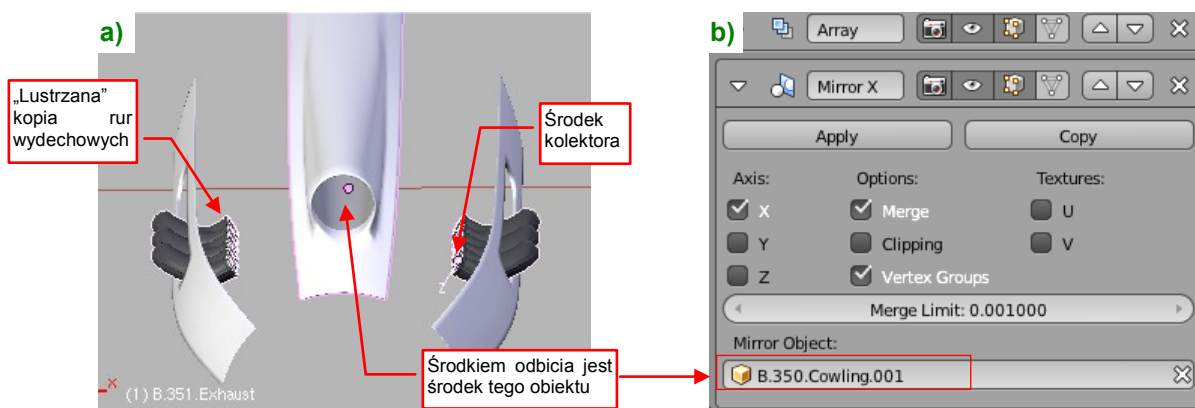
Na początek wygenerujemy za pomocą modyfikatora pierwsze trzy rury wydechowe (Rysunek 8.11.1):



Rysunek 8.11.1 Użycie modyfikatora **Array**

Zaznacz odpowiedni obiekt, i z listy **Add Modifier** wybierz **Array** (Rysunek 8.11.1a). W panelu modyfikatora **Array**, który się pojawi, ustaw licznik instancji (**Count**) na 3, i dystans pomiędzy instancjami obiektu (**Relative Offset**) na 0.87 (Rysunek 8.11.1b). Jak sam widzisz na ilustracji, modyfikator **Array** ma jeszcze wiele innych przełączników (a więc i możliwości). Nie były mi jednak potrzebne do stworzenia modelu kolektora spalin, więc nie będę ich tutaj omawiał. (Mówiąc szczerze: jeszcze ich nie sprawdziłem).

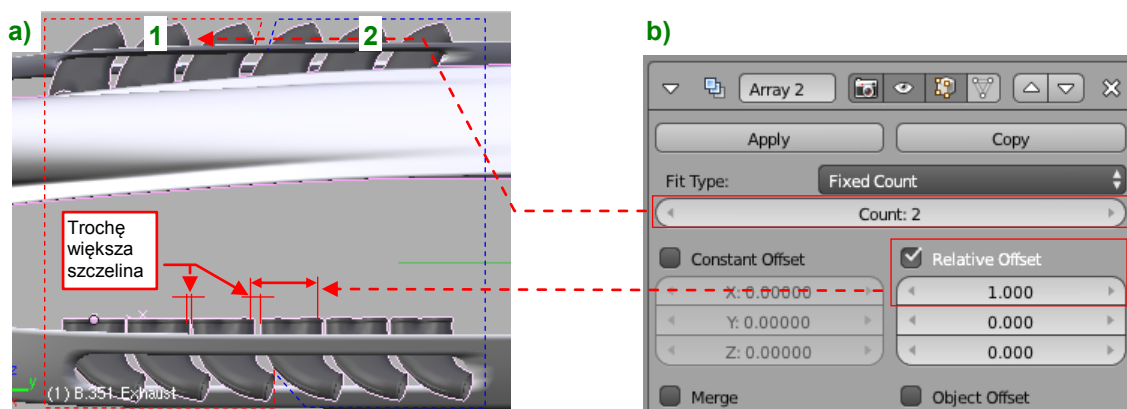
Dlaczego stworzyłem 3 rury, a nie od razu 6? Wyjaśnię za chwilę. Na razie stwórzmy lustrzane odbicie kolektora z drugiej strony kadłuba (Rysunek 8.11.2):



Rysunek 8.11.2 Powielenie kolektora spalin za pomocą modyfikatora **Mirror**

Wykorzystałem w tym celu modyfikator **Mirror**, wskazując mu jako punkt odniesienia środek pokrywy silnika. (Pokrywa ma środek w osi kadłuba, więc druga część kolektora pojawi się we właściwym miejscu).

Wróćmy do pytania: dlaczego nie stworzyłem od razu rzędu sześciu rur wydechowych? Ponieważ w odwzorowywanym silniku były podzielone na dwie sekcje, po 3 rury każda. Odstęp pomiędzy rurą trzecią i czwartą (na granicy „sekcji”) był nieco większy niż pomiędzy pozostałymi. Aby ten efekt uzyskać, wystarczy jeszcze raz użyć modyfikatora **Array** (Rysunek 8.11.3):



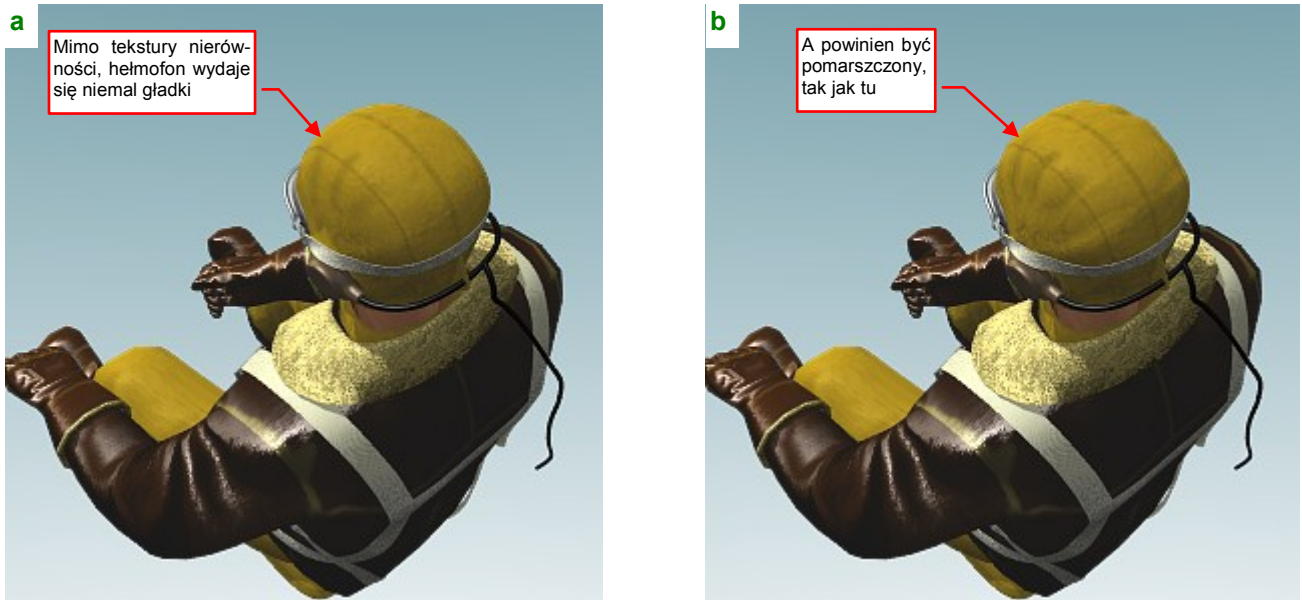
Rysunek 8.11.3 Dodanie kolejnych rur wydechowych — powtórne użycie modyfikatora **Array**

Każdy następny modyfikator **Array** powiela cały rezultat działania poprzednich modyfikatorów (Rysunek 8.11.3a), więc tym razem należy w licznik (**Count**) wpisać 2 (Rysunek 8.11.3b). Także odstęp (**Relative Offset**) pomiędzy segmentami należy zrobić nieznacznie większy — 1, a nie 0.87. (Jest o całe 13 mm szerszy — to coś dla koneserów silników Allison ©).



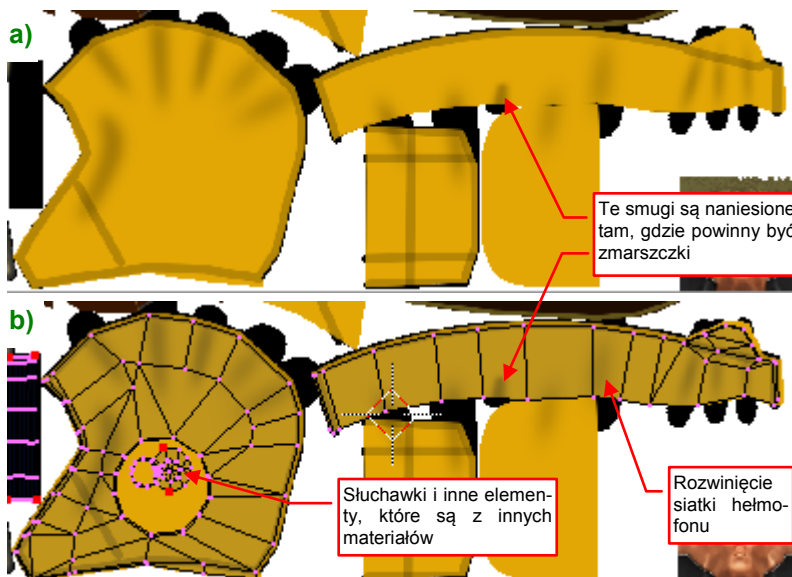
### 8.12 Deformacja według tekstury (*Displace*)

Czasami potrzebna jest deformacja siatki, która wprowadzi w jej kształt jakiś „nieład”. Na przykład — hełmofon pilota, który pokazuje Rysunek 8.12.1a), wydaje się gładki. Właściwie taki kształt wyglądałby w miarę poprawnie, gdyby to nakrycie z głowy było ze skóry. Niestety, w tym przypadku jest to płócienna „pilotka” USAAC, która powinna się marszczyć na szwach (Rysunek 8.12.1b):

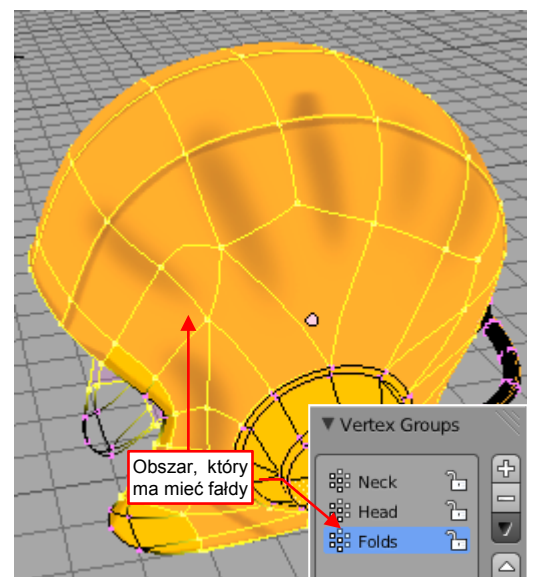


Rysunek 8.12.1 Hełmofon pilota: gładki i z fałdami

Kolory hełmofonu i linie szwów, które pokazuje Rysunek 8.12.1, uzyskałem za pomocą tekstury (Rysunek 8.12.2a). To fragment większego obrazu, naniesiony na rozwinięcie UV siatki hełmofonu (Rysunek 8.12.2b):



Rysunek 8.12.2 Obraz 000.Color, nałożony na rozwinięciu UV hełmofonu

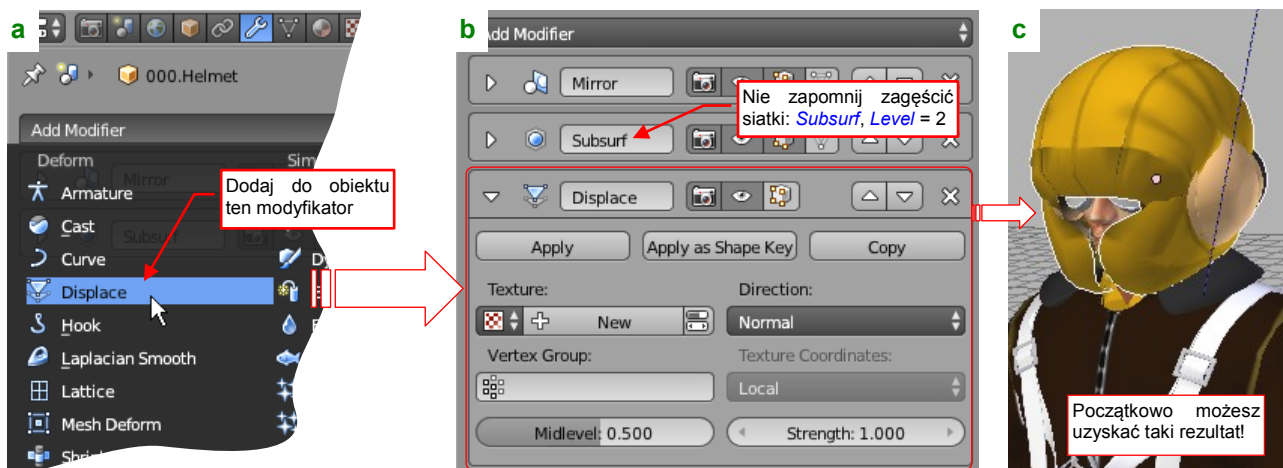


Rysunek 8.12.3 Definicja grupy Folds

Obraz 000.Color, przedstawiony przez Rysunek 8.12.2 jest użyty w dwóch rolach: tekstury koloru i nierówności. Jak pokazuje render (Rysunek 8.12.1a), wychodzi to zbyt słabo.

Skoro zwykła tekstura nierówności jest zbyt „słaba”, musimy nanieść na siatkę hełmofonu prawdziwe fałdy. Aby to było możliwe, najpierw należy ją odpowiednio zagęścić (modyfikatorem *Subsurf*, z *Level* = 2). Warto także wydzielić nową grupę wierzchołków (por. str. 278), które będą podlegały tej deformacji. Zaznacz odpowiednie wierzchołki i nadaj tej *Vertex Group* nazwę *Folds* (Rysunek 8.12.3).

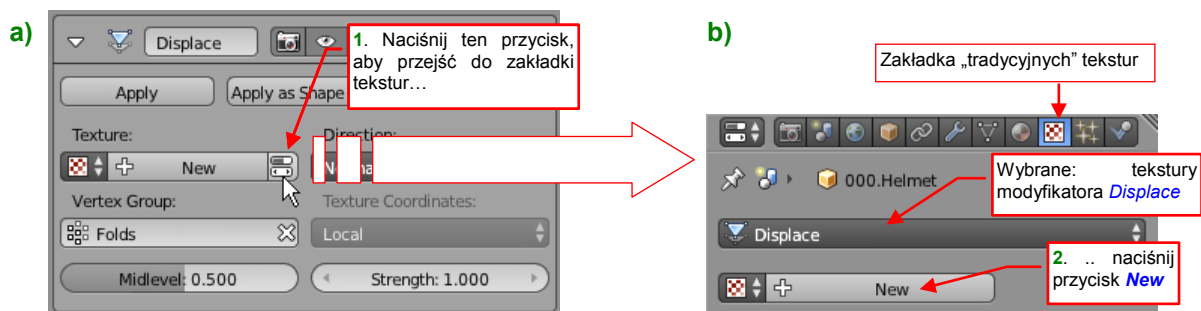
Teraz dodaj do stosu modyfikatorów siatki kolejny, o nazwie **Displace** (Rysunek 8.12.4a):



Rysunek 8.12.4 Dodanie modyfikatora **Displace**

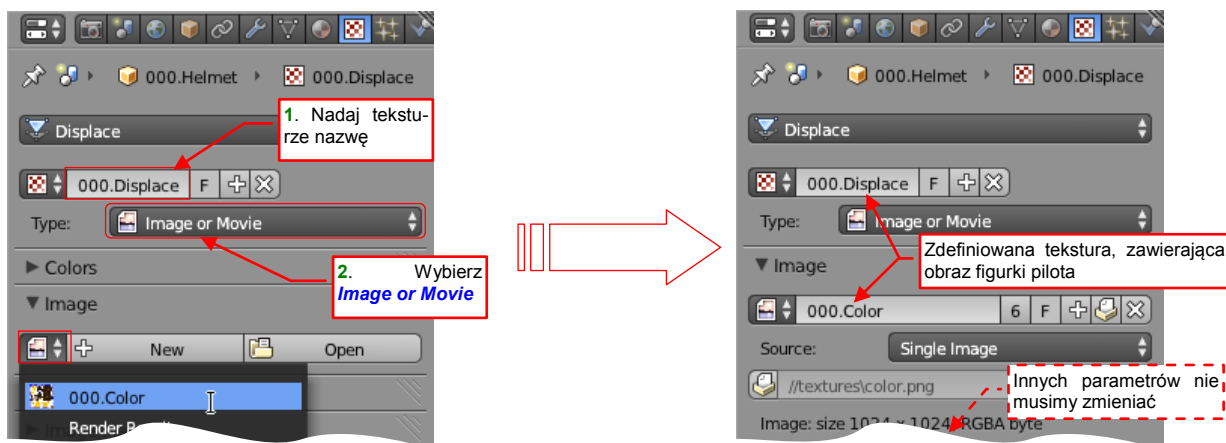
Trzeba przyznać że zaraz po dodaniu do hełmofonu tego modyfikatora, jego domyślne ustawienia (Rysunek 8.12.4b) „rozdmuchały” nasz hełmofon w coś, co przypomina balon (Rysunek 8.12.4c). Zaraz je zmienimy, i wszystko wróci na swoje miejsce!

Zacznijmy od zdefiniowania mapy deformacji. Modyfikator **Displace** wykorzystuje tekstury używane przez **Blender Renderer**, a nie **Cycles Renderer**. Stąd trzeba teraz stworzyć nową teksturę (Rysunek 8.12.5):



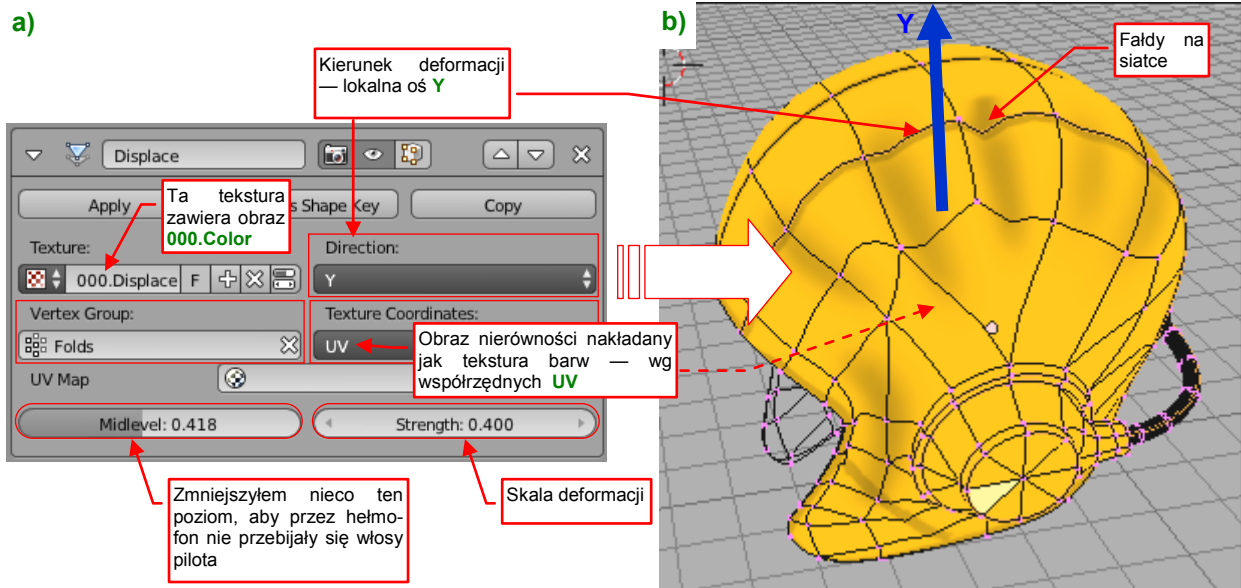
Rysunek 8.12.5 Przejście do tworzenia mapy deformacji

W panelu modyfikatora naciśnij przycisk pokazywany przez Rysunek 8.12.5a). (Uwaga: występuje tylko wtedy, gdy masz ustawiony jako aktywny renderer **Cycles**). Możesz także po prostu przejść do kontekstu **Textures** i wybrać tam zestaw związany z modyfikatorem **Displace** Rysunek 8.12.5b). Naciśnij tam przycisk **New**. Następnie nadaj nowej teksturze nazwę (w tym przypadku **000.Displace**), zmień jej typ na **Image or Movie** i przypisz do obrazu **000.Color** (Rysunek 8.12.6):



Rysunek 8.12.6 Zdefiniowanie tekstury deformacji (**000.Displace**)

Teraz możemy wrócić do ustawień naszego modyfikatora. Jest dopasowany do rozwinięcia **UV** siatki, więc uzyskanie pofałdowania jest bardzo proste (Rysunek 8.12.7a):



Rysunek 8.12.7 Dodanie modyfikatora **Displace** i jego wpływ na siatkę

W poszczególnych polach modyfikatora wpisujemy:

- **Direction:** kierunek deformacji. Ograniczyłem do lokalnej osi pionowej obiektu (w przypadku helmofonu jest to jego oś **Y**);
- **Texture Coords:** mapowanie tekstury. Ustawiłem na **UV**, bo nasz obraz pasuje do rozwinięcia obiektu;
- **Vertex Group:** ogranicza deformację do grupy wierzchołków (w tym przypadku — **Folds**);
- **Midlevel:** poziom odniesienia (0.0..1.0). Barwy ciemniejsze będą tworzyć wgłębienia, jaśniejsze — wypukłości. Ustawiłem na 0.4, aby nie przebijały się włosy pilota;
- **Strength:** skala (wysokość) deformacji — dobrałem tak, by „faldki” nie były zbyt wysokie (0.4).

Rysunek 8.12.7b) przedstawia działanie tak ustawionego modyfikatora **Displace**. Drobne ściany, stworzone przez **Subsurf**, zostały przesunięte wzdłuż osi **Y**.

Rysunek 8.12.8 przedstawia render tak zmodyfikowanego helmofonu:

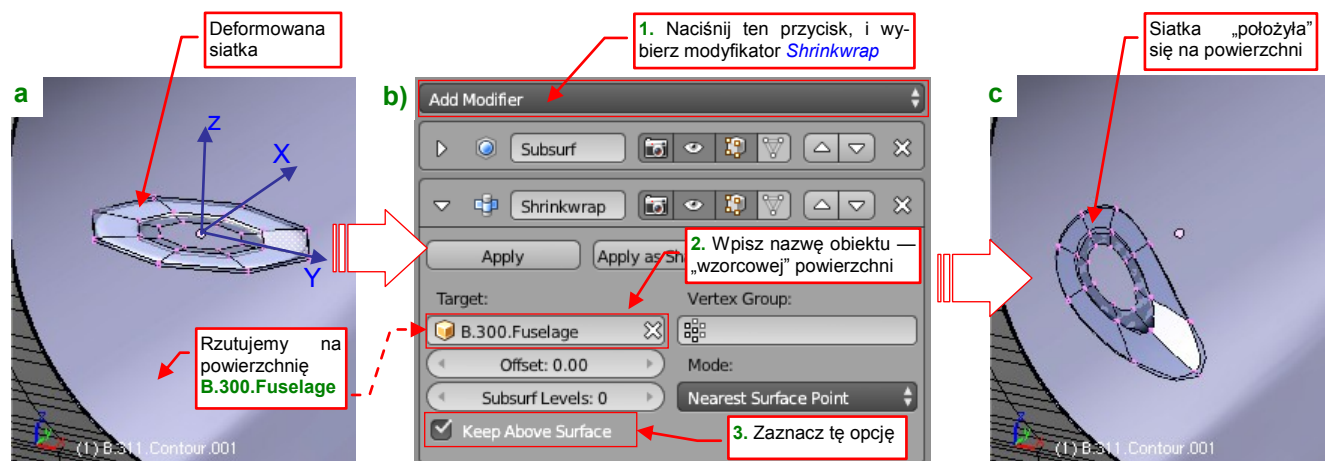


Rysunek 8.12.8 Rezultat (render helmofonu)

Zwróć uwagę, że dzięki **Displace** mogliśmy zachować do modyfikacji oryginalną, niezdeformowaną siatkę.

### 8.13 Rzutowanie na powierzchnię (*Shrinkwrap*)

Modyfikator *Shrinkwrap* pozwala położyć siatkę na wskazanej powierzchni („jak naleśnik”) (Rysunek 8.13.1):

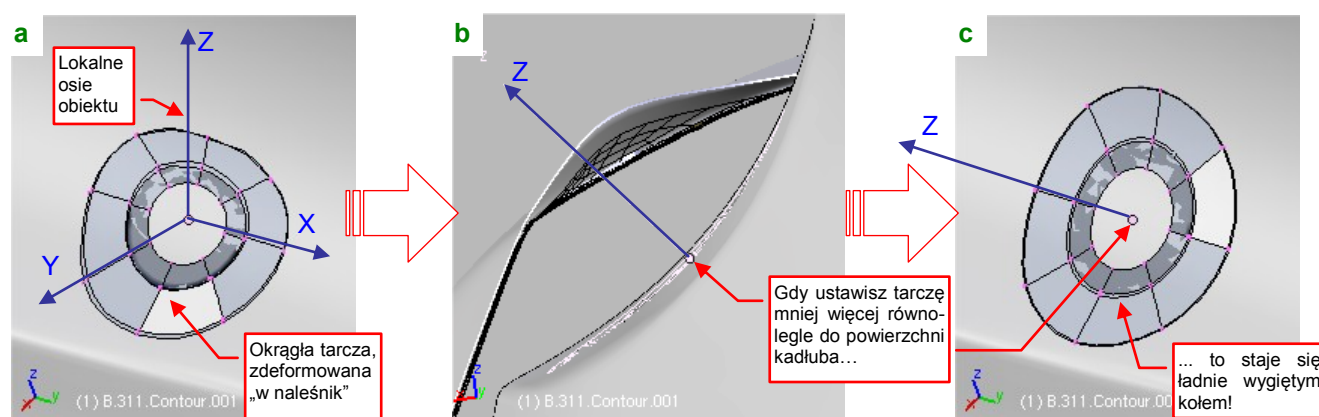


Rysunek 8.13.1 Rzutowanie siatki na inną powierzchnię

Dla przykładu „położymy” na powierzchni kadłuba (obiekt **B.300.Fuselage**) siatkę w kształcie pierścienia (Rysunek 8.13.1a). W sekcji *Modifiers* naciśnij przycisk *Add Modifier*. Spowoduje to rozwinięcie listy dostępnych modyfikatorów siatki. Wybierz spośród nich *Shrinkwrap*. W panelu modyfikatora (Rysunek 8.13.1b) wpisz w pole *Target* nazwę obiektu — powierzchni, na którą rzutujemy siatkę. Dodatkowo możesz zaznaczyć jeszcze opcję *Keep Above Surface*. W rezultacie nasz pierścień „opadł” na kadłub, zupełnie tak jak naleśnik (Rysunek 8.13.1c)

Na razie ten rezultat jest daleki od satysfakcjonującego: chcielibyśmy, aby nasz pierścień po deformacji wyglądał jak wygięte koło, a nie jak jakiś kawałek gumy!

Na szczęście nie jest to takie trudne do osiągnięcia (Rysunek 8.13.2):



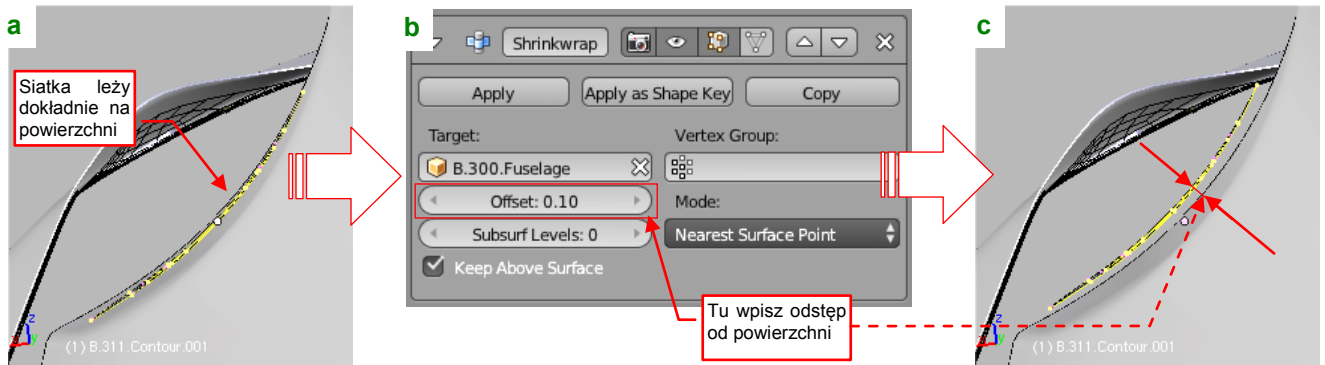
Rysunek 8.13.2 Korygowanie kształtu rzutu

Obróć pierścień wokół lokalnej osi **X**, by jego powierzchnia była w przybliżeniu równoległa do powierzchni kadłuba. Nie zaszkodzi umieścić jego środka na rzutowanej powierzchni (Rysunek 8.13.2b). Widzisz, co się stało (Rysunek 8.13.2c)? Uzyskaliśmy to, o co chodziło — okrąg, położony na kadłubie.

- Stosujemy tu tryb *Nearest Surface Point*, w którym modyfikator *Shrinkwrap* przesuwają wierzchołki deformowanej siatki do najbliższego punktu na powierzchni obiektu odniesienia (*Target*). Możesz także stosować inny tryb — *Project*. W tym trybie wszystkie wierzchołki są rzutowane w jednym kierunku, wzdłuż jednej z trzech osi lokalnego układu współrzędnych modyfikowanego obiektu.



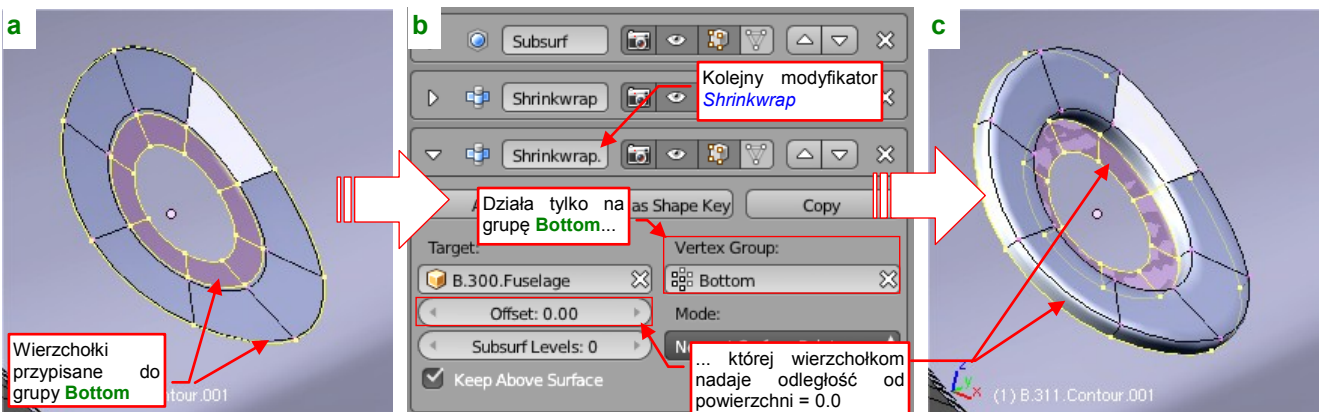
Na razie nasza siatka leży dokładnie na powierzchni kadłuba (Rysunek 8.13.3a):



Rysunek 8.13.3 Odsunięcie od powierzchni

Przy modelowaniu konstrukcji zazwyczaj jednak chcemy, by miała pewną grubość. Do ustalania tego odstępu służy pole **Offset**, umieszczone na panelu modyfikatora (Rysunek 8.13.3b). Gdy wpiszesz tam jakąś niezerową wartość, wygięta siatka ulegnie odsunięciu (Rysunek 8.13.3c).

To jednak dopiero połowa sukcesu. W tej chwili nasz wygięty pierścień „lewituje” nad kadłubem. Aby uzyskać solidne boczne ścianki, zaznacz zewnętrzne krawędzie siatki, i przypisz je do jakiejś nowej grupy wierzchołków (por. str. 278). Nadajmy jej nazwę, np. **Bottom** (Rysunek 8.13.4a):

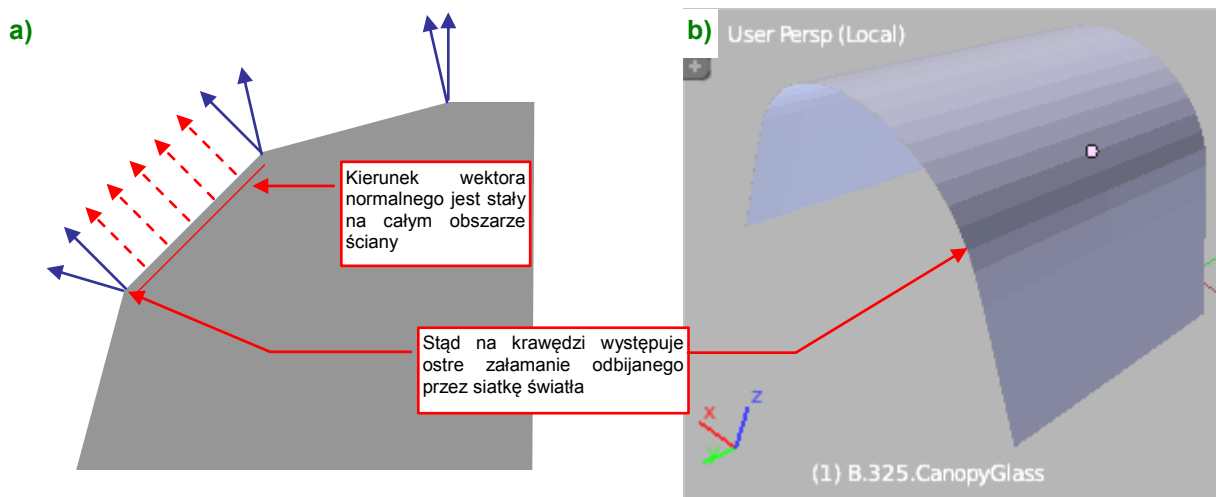


Rysunek 8.13.4 Nadanie rzutowanej siatce „grubości”

Następnie dodaj kolejny modyfikator **Shrinkwrap** (Rysunek 8.13.4b). Tym razem jednak wpisz w pole **Vertex Group** nazwę **Bottom**, by działał tylko na krawędzie należące do tej grupy. Potem zmniejsz w tym modyfikatorze **Offset** do zera. W efekcie mamy oczekiwany rezultat: wygięty pierścień z bocznymi ściankami, nadające siatce „grubość”.

### 8.14 Sterowanie kierunkiem normalnych wzdłuż krawędzi siatki

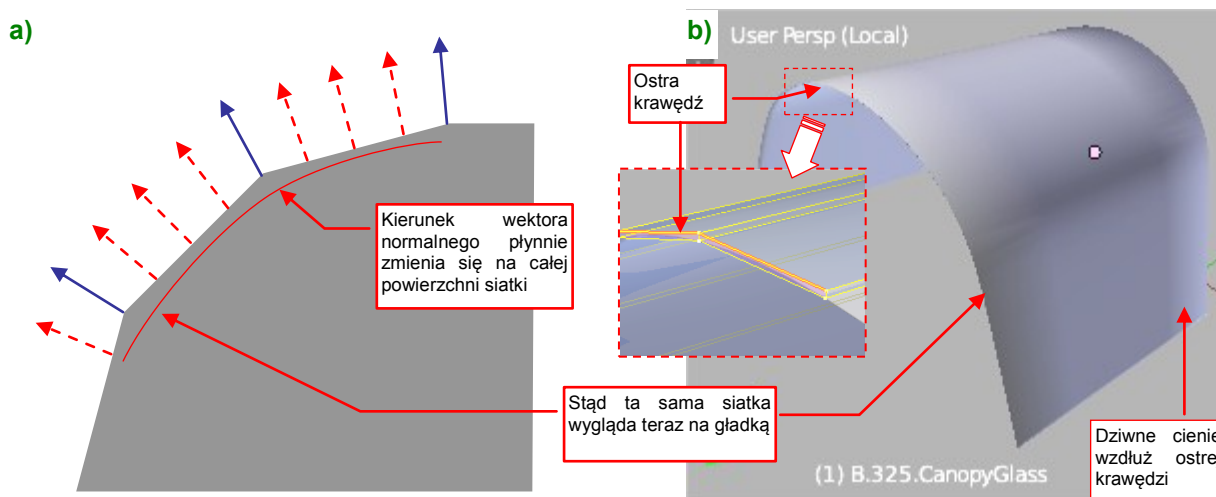
Kierunek normalnych do powierzchni wpływa na to, w jaki sposób wyświetlane są na ekranie krawędzie siatki. Program może wyznaczyć wektor normalny dla każdej płaskiej, elementarnej ściany. Każdy komputerowy model oświetlenia wymaga określenia kierunku normalnego dla dowolnego miejsca powierzchni. Jeżeli przyjmujemy najprostsze założenie, że na kierunek normalny na całym obszarze ściany jest stały (Rysunek 8.14.1a), uzyskamy efekt jak dla trybu **Shade Flat** (Rysunek 8.14.1b):



Rysunek 8.14.1 Wektory normalne w trybie **Shade Flat**

Cieniowanie **Shade Flat** jest nieodpowiednie dla obłych powierzchni, takich jak opływowe kształty współczesnych samolotów. Musiałbyś bardzo zagęścić ich siatki, by wyglądały na gładkie!

Na szczęście nie musimy tak komplikować modelu, bo mamy do dyspozycji drugi tryb cieniowania: **Shade Smooth** (Rysunek 8.14.2):



Rysunek 8.14.2 Wektory normalne w trybie **Shade Smooth**

W tym trybie kierunek normalny w dowolnym punkcie powierzchni jest interpolacją (uśrednieniem) kierunków normalnych najbliższych ścian (Rysunek 8.14.2a). Do interpolacji wykorzystuje się tzw. metodę Phong, bardzo popularną w grafice komputerowej. Używając jej, za pomocą niewielu ścian można uzyskać wrażenie gładkiej powierzchni (Rysunek 8.14.2b).

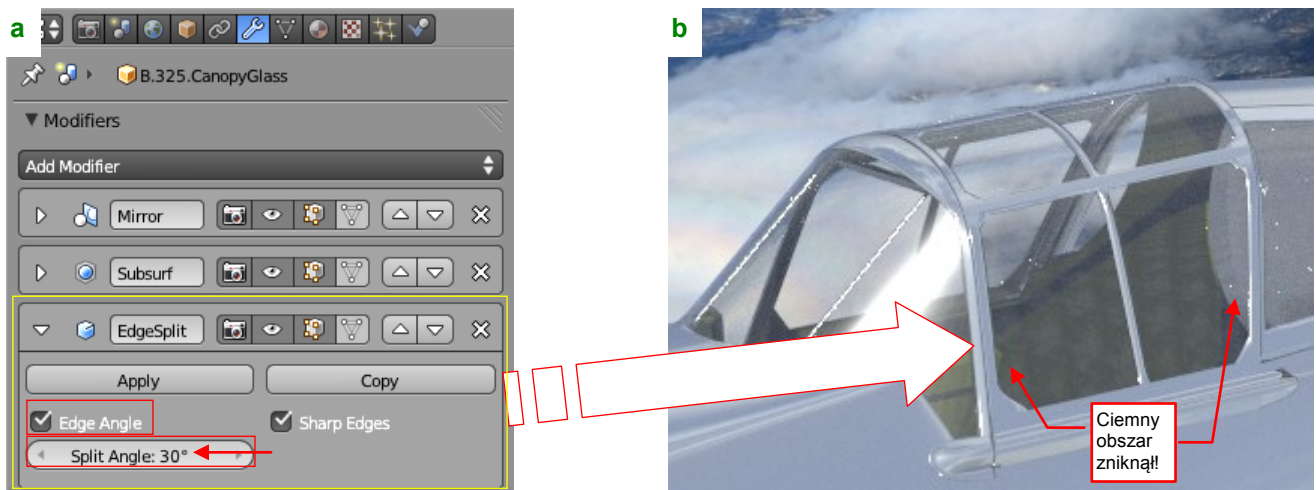
Element, który przedstawia Rysunek 8.14.2b), to oszkleenie owiewki kabiny pilota. Na powiększeniu ilustracji pokazuję, że ma ona niewielką grubość. Jej siatka oryginalnie wygładzona powierzchnią podziałową. Krawędź, biegnącą wzdłuż jej brzegu oznaczyłem jako „ostrą” (**Crease**), by owiewka wyglądała jak wygięty arkusz szkła organicznego. Spowodowało to pojawienie się na ścianach sąsiadujących z tą ostrą krawędzią dziwnych cieni (Rysunek 8.14.2b).

Jeżeli zignorujesz te cienie, staną się przyczyną dziwnych efektów na renderze (Rysunek 8.14.3):



Rysunek 8.14.3 Efekt ostrej krawędzi oszklenia powierzchni i cieniowania **Shade Smooth** na renderze

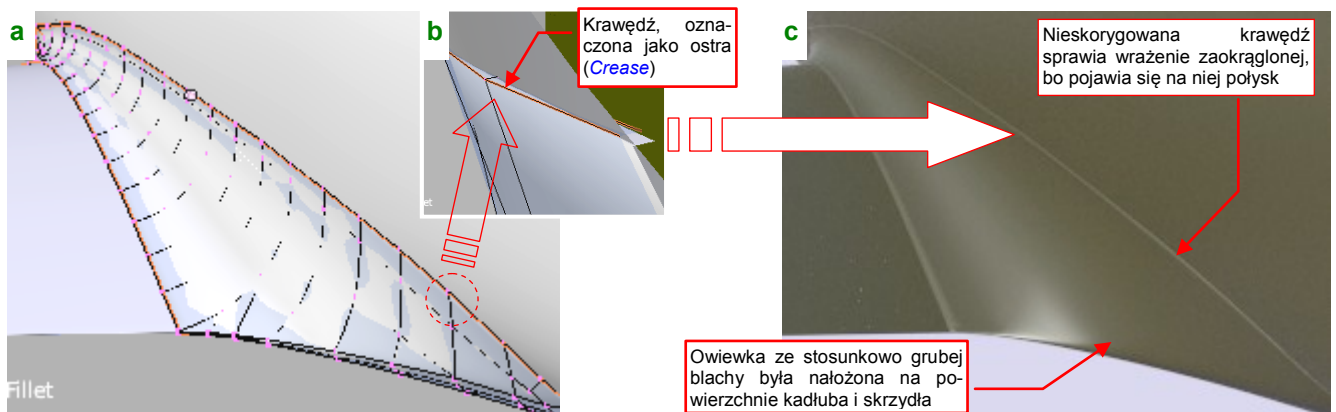
To rezultat zastosowania interpolacji normalnych metodą Phonga dla ostrej krawędzi. Na szczęście remedium jest bardzo proste: dołącz na koniec listy modyfikatorów obiektu efekt **Edge Split** (Rysunek 8.14.4):



Rysunek 8.14.4 Modyfikator **Edge Split** — automatyczne korygowanie ostrych krawędzi

Domyślnie **Edge Split** ma włączoną opcję **Edge Angle**. W tym trybie modyfikator poprawia kierunki normalne wzdłuż wszystkich krawędzi siatki, które łączą ściany pod kątem większym niż określony przez parametr **Split Angle**. (Kierunki normalnych wzdłuż takiej krawędzi są rozdzielone, jak pokazywał to Rysunek 8.14.1a).

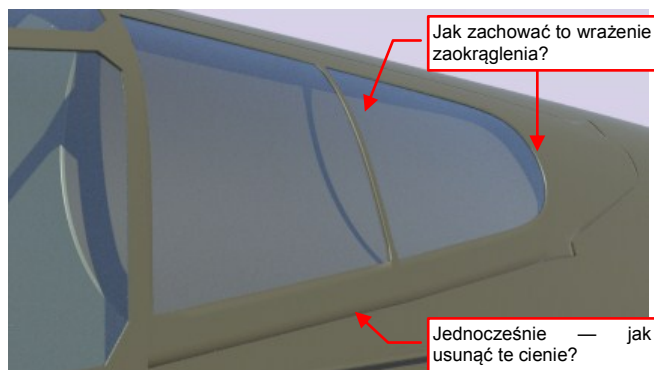
Dla powierzchni nieprzezroczystych interpolacja normalnych wzdłuż krawędzi nie powoduje tak drastycznych skutków, jak dla szkła. Przyjrzyj się na przykład tej owiewce skrzydła (Rysunek 8.14.5):



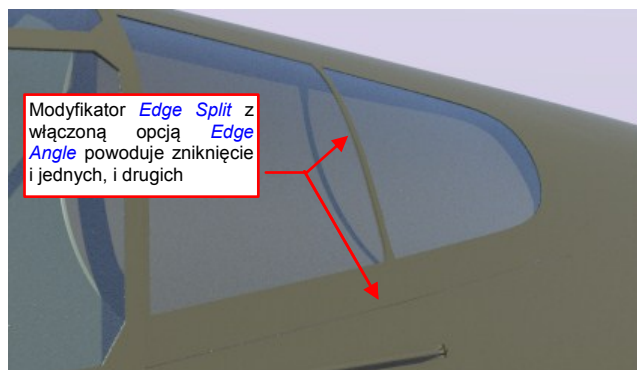
Rysunek 8.14.5 Odbłask wzdłuż nieskorygowanej krawędzi

Tutaj taka krawędź wygląda jak zaokrąglona i nadaje elementowi wrażenie grubości (Rysunek 8.14.5b).

Co jednak robić w takim przypadku, jaki pokazuje Rysunek 8.14.6? Powiedzmy, że chcielibyśmy zachować wrażenie zaokrąglonych krawędzi przy szkłe kabiny. Jednocześnie warto usunąć brzydkie cienie na zewnętrznych krawędziach tej ramki:



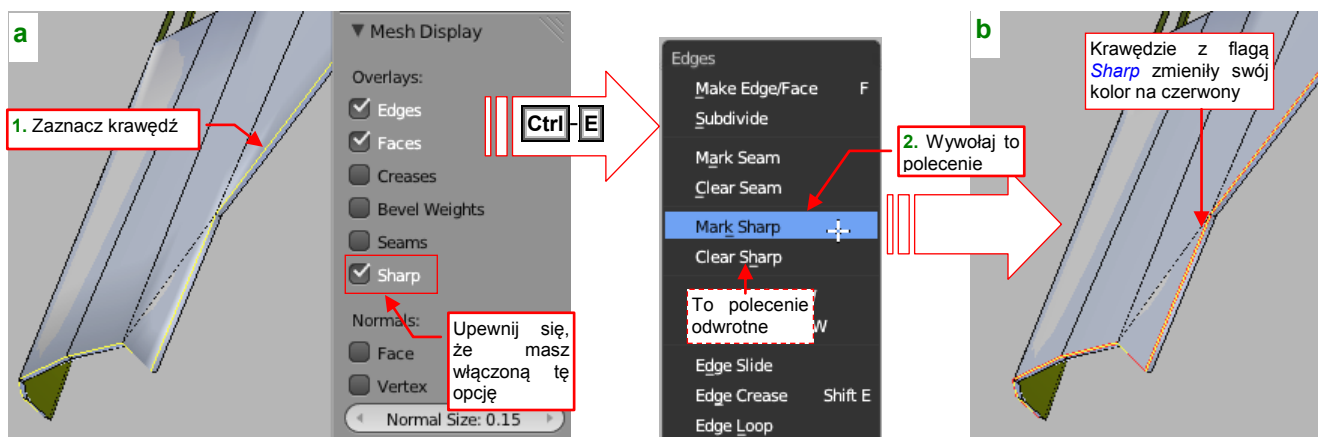
Rysunek 8.14.6 Nieskorygowane krawędzie ramki



Rysunek 8.14.7 Ramka po automatycznej korekcie

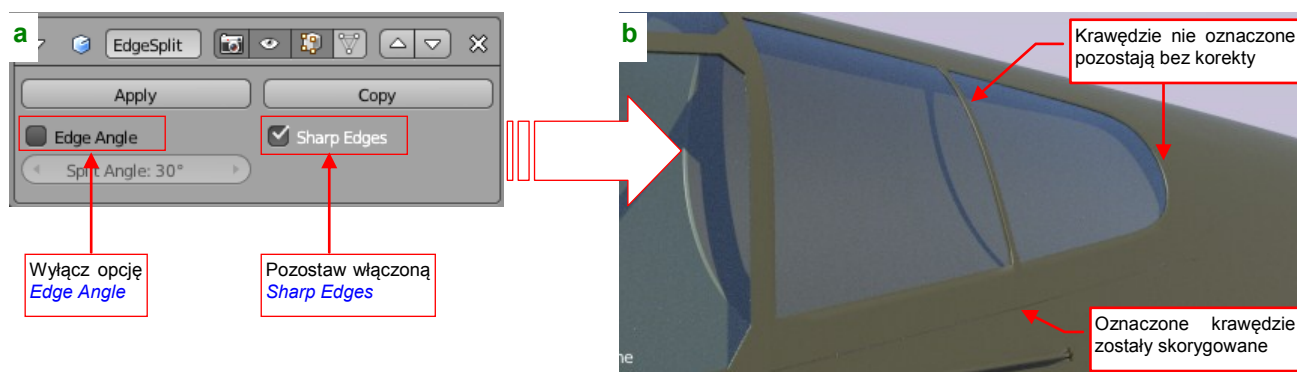
Zastosowanie „automatycznego” **Edge Split** (z włączoną opcją **Edge Angle**) „spłaszcza” wszystkie ostre krawędzie. A nam chodzi, aby tak poprawić tylko zewnętrzną krawędź ramki — tak, jak pokazuje Rysunek 8.14.7.

Taki wybiórczy efekt jest do osiągnięcia. Trzeba tylko najpierw oznaczyć krawędzie, które mają być skorygowane, specjalną flagą **Sharp**. Upewnij się, że przyborniku właściwości tej siatki masz włączoną opcję **Mesh Display:Overlays:Sharp** — Rysunek 8.14.8a). Potem zaznacz korygowane krawędzie:



Rysunek 8.14.8 Oznaczanie wybranych krawędzi jako **Sharp**

Z menu **Mesh→Edges** (**Ctrl-E**) wywołaj polecenie **Mark Sharp**. W efekcie Blender podświetli zaznaczone krawędzie na czerwono (Rysunek 8.14.8b). (Taki kolor jest przypisany fladze **Sharp**). Potem wystarczy wyłączyć w modyfikatorze **Edge Split** opcję **Edge Angle** (Rysunek 8.14.9a):

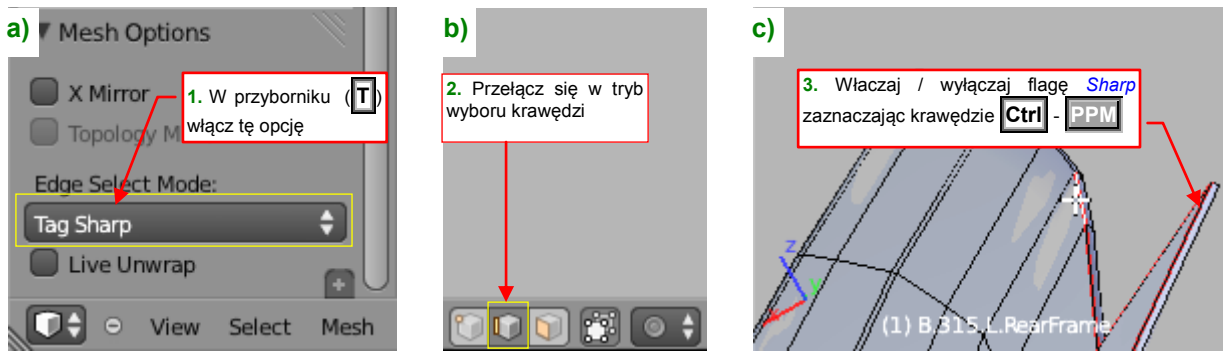


Rysunek 8.14.9 Rezultat — selektywna korekta krawędzi

Pozostawienie opcji **Sharp Edges** skoryguje tylko krawędzie oznaczone flagą **Sharp** (Rysunek 8.14.9b).

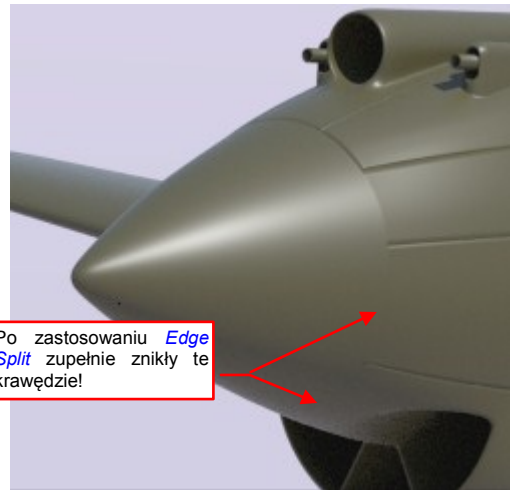
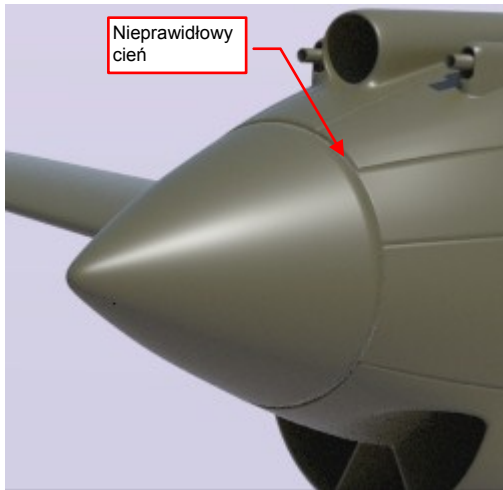


Oznaczanie krawędzi flagą **Sharp** można także wykonać w inny sposób (Rysunek 8.14.10):



Rysunek 8.14.10 Alternatywna metoda oznaczanie krawędzi flagą **Sharp**

Niestety, flaga **Sharp** nie może mieć wartości pośrednich, tak jakie ma dla powierzchni podziałowych flaga **Crease**. Czasami więc możesz natknąć się na taki przypadek, jak dla kołpaka śmigła, który przedstawia (Rysunek 8.14.11). Bez korekty kierunku normalnych krawędzi kołpaka wygląda na zbyt zaokrągloną:

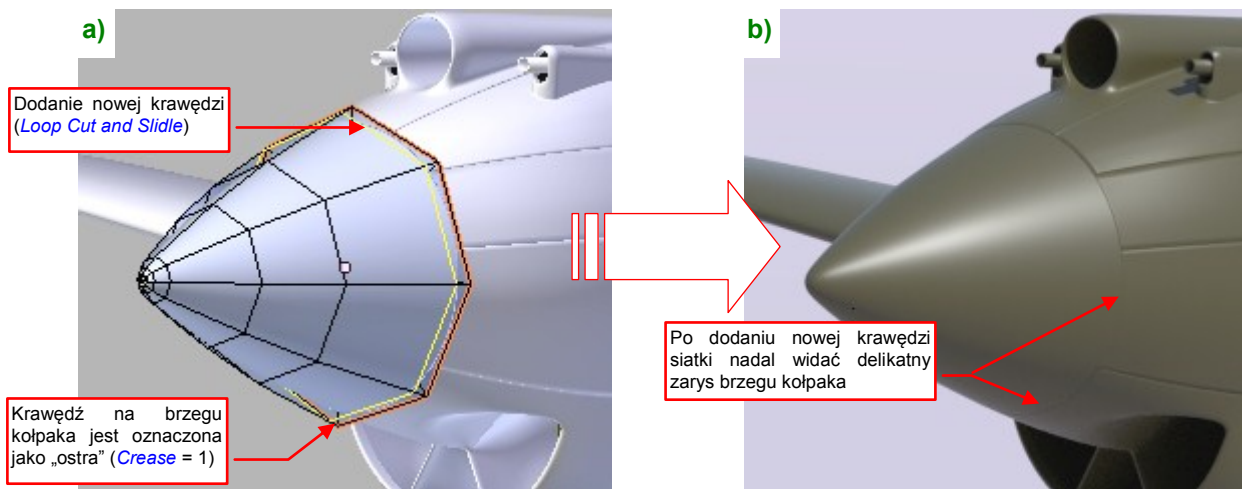


Rysunek 8.14.11 Krawędź nieskorygowana

Rysunek 8.14.12 Krawędź skorygowana

Działanie modyfikatora **Edge Split** jest w tym przypadku nieco za mocne: część krawędzi kołpaka zlewa się zupełnie z osłoną silnika (Rysunek 8.14.12).

W takim przypadku pozostaje tylko jedno: wstawienie w pobliżu „ostrej” krawędzi dodatkowego rzędu wierzchołków (np. poleceniem **Loop Cut and Slide** — Rysunek 8.14.13a):



Rysunek 8.14.13 Zmniejszenie cienia na powłoce poprzez wstawienie dodatkowej krawędzi

Nowa krawędź przy brzegu kołpaka zdecydowanie zmniejsza efekt zaokrąglenia (Rysunek 8.14.13b).

Jak widzisz, Blender nie pozwala na bezpośrednią manipulację kierunkami normalnych siatki. Możesz to robić pośrednio, aby uzyskać na renderze poprawny efekt ostrej krawędzi. Służy do tego modyfikator [Edge Split](#).

- Dopóki można, staraj się eliminować zaburzenia kierunku normalnych wzdłuż ostrych krawędzi modyfikatorem [Edge Split](#) i — ewentualnie — [Bevel](#). Dodanie nowej krawędzi do siatki nie eliminuje, a tylko zmniejsza te zaburzenia.

Aby uniknąć poważnych kłopotów na renderze (por. str. 305, Rysunek 8.14.3b), do elementów „szklanych” zawsze stosuj modyfikator [Edge Split](#) z włączoną opcją [Edge Angle](#). Można go także używać do większości innych siatek z ostrymi krawędziami (np. siatki płata). Tylko w szczególnych przypadkach warto używać zaznaczania pojedynczych krawędzi flagą [Sharp](#), tak jak to pokazałem na str. 306.

Czasami decydowałem się — ale jeszcze na etapie modelowania — wstawiać dodatkowe krawędzie aby usunąć duże zaburzenia kierunku normalnych. Zrobiłem to „na wszelki wypadek”. Sądzę, że w ten sposób te siatki sprawią mniejszy kłopot przy imporcie do jakiegoś programu, za pomocą formatu danych który nie uwzględnia informacji o kierunkach normalnych. Dodanie nowej krawędzi na późniejszych etapach prac nad modelem może być dość kłopotliwe. Gdy siatki modelu są już rozwinięte w przestrzeni UV (por. Wyciąg III) każda nowa krawędź wymaga dodatkowej korekty tego rozwinięcia.

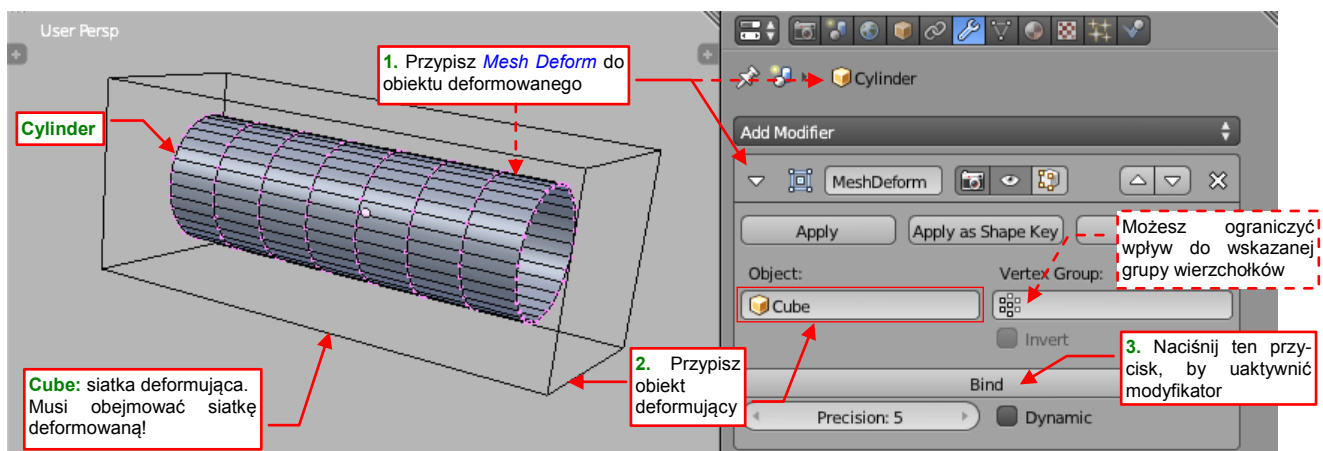
### 8.15 Deformacja za pomocą pomocniczej siatki (*Mesh Deform*)

Modyfikator *Mesh Deform* pozwala sterować kształtem skomplikowanej siatki o dużej liczbie ścian za pomocą otaczającej ją siatki o małej liczbie ścian.

Przygotuj najpierw siatkę deformującą. Może to być np. taki prostopadłościan, jaki pokazuje Rysunek 8.15.1. Powinien otaczać deformowaną siatkę. Aby ściany obiektu deformującego nie zasłaniały tego, co w środku, zmieniłem jego tryb wyświetlania (*Properties: Object: Display: Type*) na *Wire* (por. str. 228).

- Siatka deformująca musi otaczać (zawierać w środku) wszystkie deformowane obiekty.

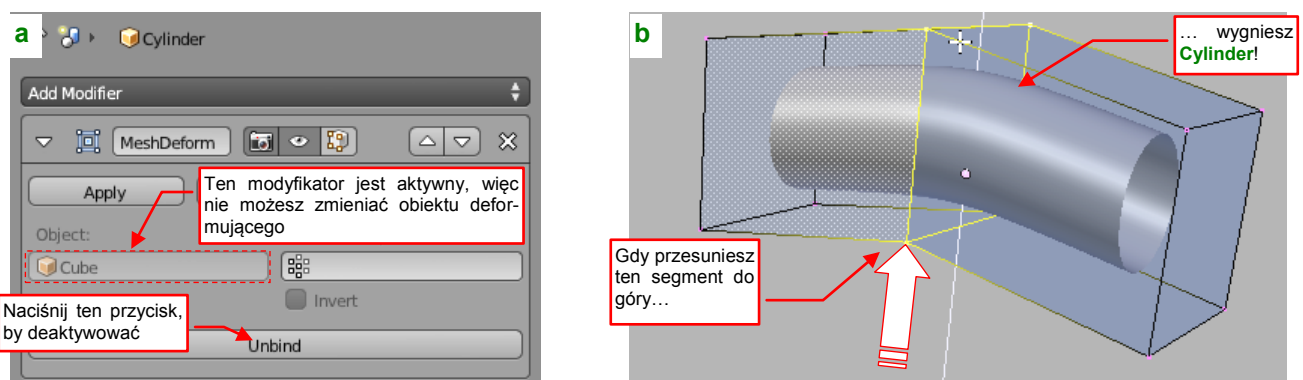
Następnie zaznacz obiekt deformowany (w naszym przykładzie to *Cylinder*), i w zestawie *Modifiers*, panelu *Modifiers*, naciśnij przycisk *Add Modifier*. Spowoduje to rozwinięcie listy dostępnych "modyfikatorów" tej siatki. Wybierz spośród nich *Mesh Deform* (Rysunek 8.15.1):



Rysunek 8.15.1 Przygotowanie modyfikatora *Mesh Deform*

We właściwościach tego modyfikatora wpisz w pole *Object* odpowiedni obiekt deformujący (w naszym przykładzie to *Cube*). Początkowo modyfikator jest nieaktywny. Aby zaczął działać, naciśnij przycisk *Bind* (Rysunek 8.15.1). Powoduje to obliczenie wpływów wierzchołków obiektu deformującego na deformowaną siatkę. W zależności od liczby wierzchołków może to potrwać dłuższą chwilę.

Aktywny modyfikator możesz poznać po tym, że zamiast przycisku *Bind* zawiera przycisk *Unbind* i ma wyszarzone pole *Object* (Rysunek 8.15.2a):

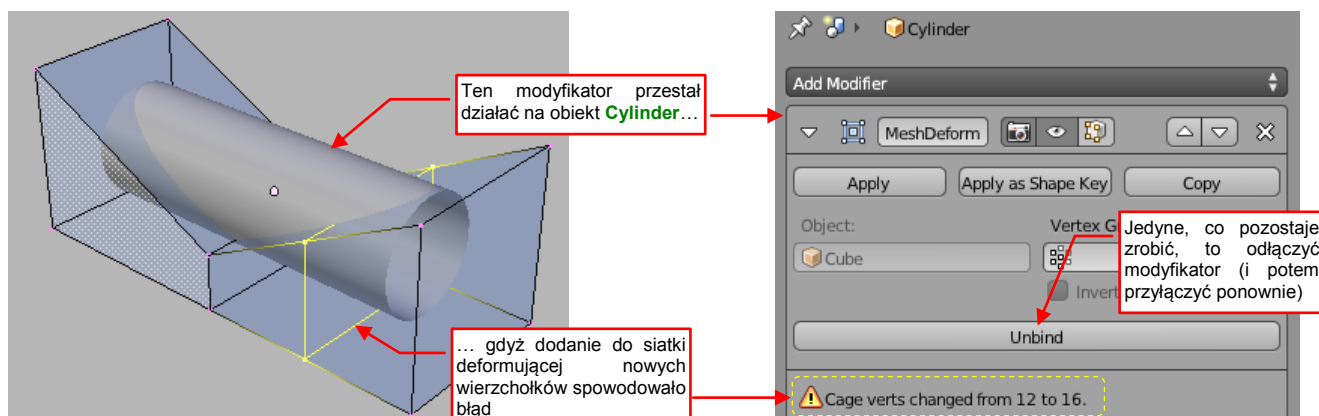


Rysunek 8.15.2 Działanie modyfikatora *Mesh Deform*

Aby zdeformować siatkę, w *Object Mode* wybierz obiekt deformujący i przełącz go w tryb edycji (Rysunek 8.15.2b). Każda zmiana jego kształtu powoduje płynną deformację modyfikowanego obiektu.

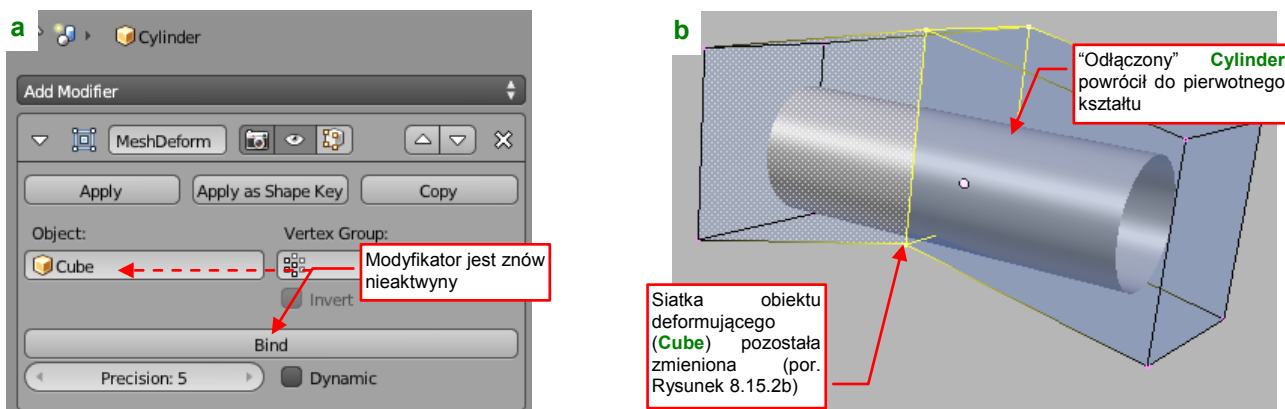
Jeżeli jesteś zadowolony z uzyskanego efektu, możesz go tak pozostawić, lub nacisnąć w panelu modyfikatora przycisk *Apply*, by go „utrwalić” w deformowanej siatce.

Pamiętaj, aby podczas takich operacji nie zmieniać w żaden sposób topologii siatki deformatora, bo to powoduje natychmiastowe wyłączenie przypisanych do niego modyfikatorów **Mesh Deform** (Rysunek 8.15.3):



Rysunek 8.15.3 Wpływ zmiany liczby wierzchołków siatki na modyfikator **Mesh Deform**

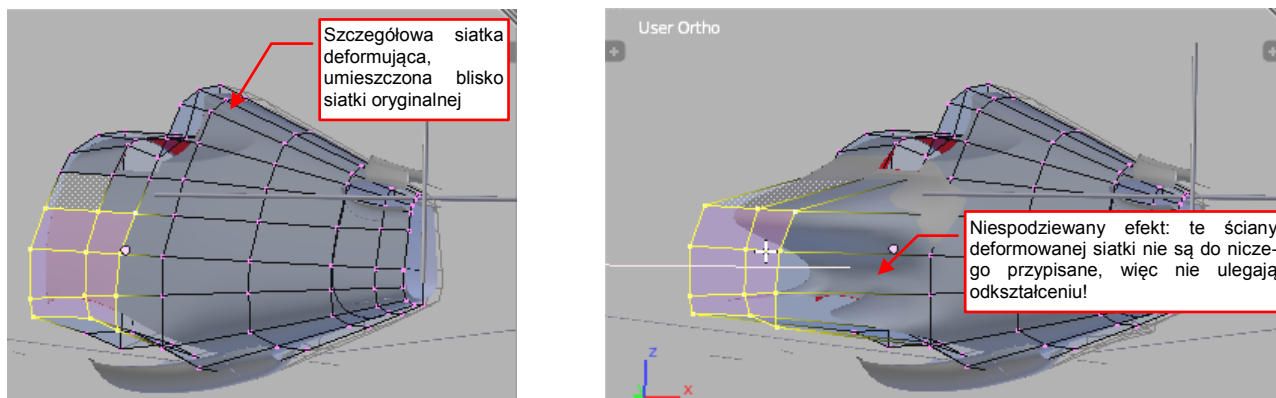
Jeżeli chcesz wprowadzić jakieś zmiany do siatki deformatora — lepiej najpierw odłączyć modyfikator (naciskając klawisz **Unbind** — por. Rysunek 8.15.2a i Rysunek 8.15.3). Wówczas obiekt deformowany powróci do pierwotnego kształtu, natomiast obiekt deformujący pozostanie w zmienionej postaci (Rysunek 8.15.4b):



Rysunek 8.15.4 Rezultat wyłączenia modyfikatora **Mesh Deform**

Teraz możesz wprowadzić dowolne zmiany do siatki deformującej. Osobiście, bardzo mi przeszkadza że po odłączeniu od **Mesh Deform** deformator nie wraca do oryginalnego kształtu. Teraz, gdy naciśniesz ponownie przycisk **Bind**, dla uzyskania podobnego do poprzedniego wygięcia obiektu **Cylinder** należy jeszcze bardziej zniekształcić siatkę **Cube**.

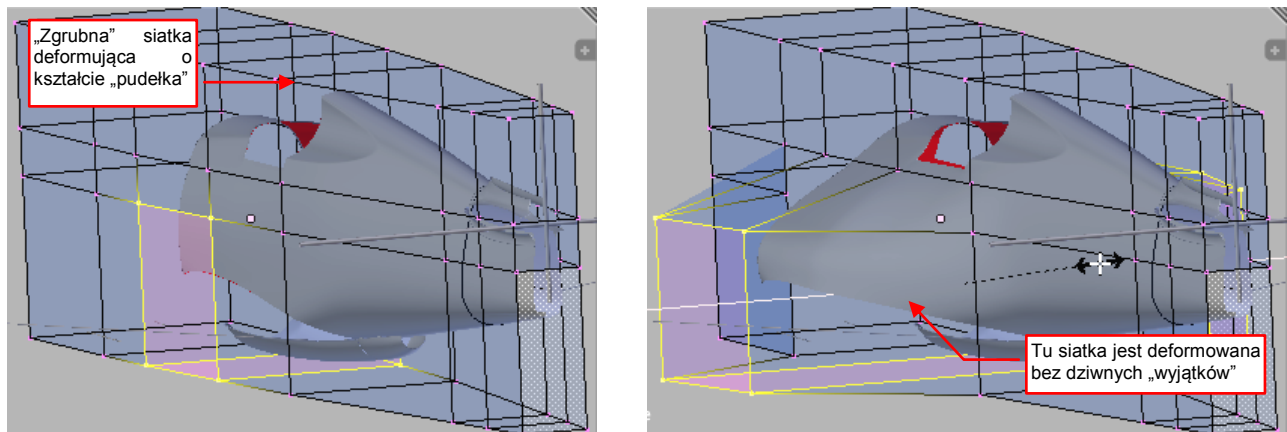
Na koniec mała rada: nie używaj zbyt szczegółowej siatki deformującej (Rysunek 8.15.5):



Rysunek 8.15.5 Problemy ze zbyt dokładną siatką deformującą



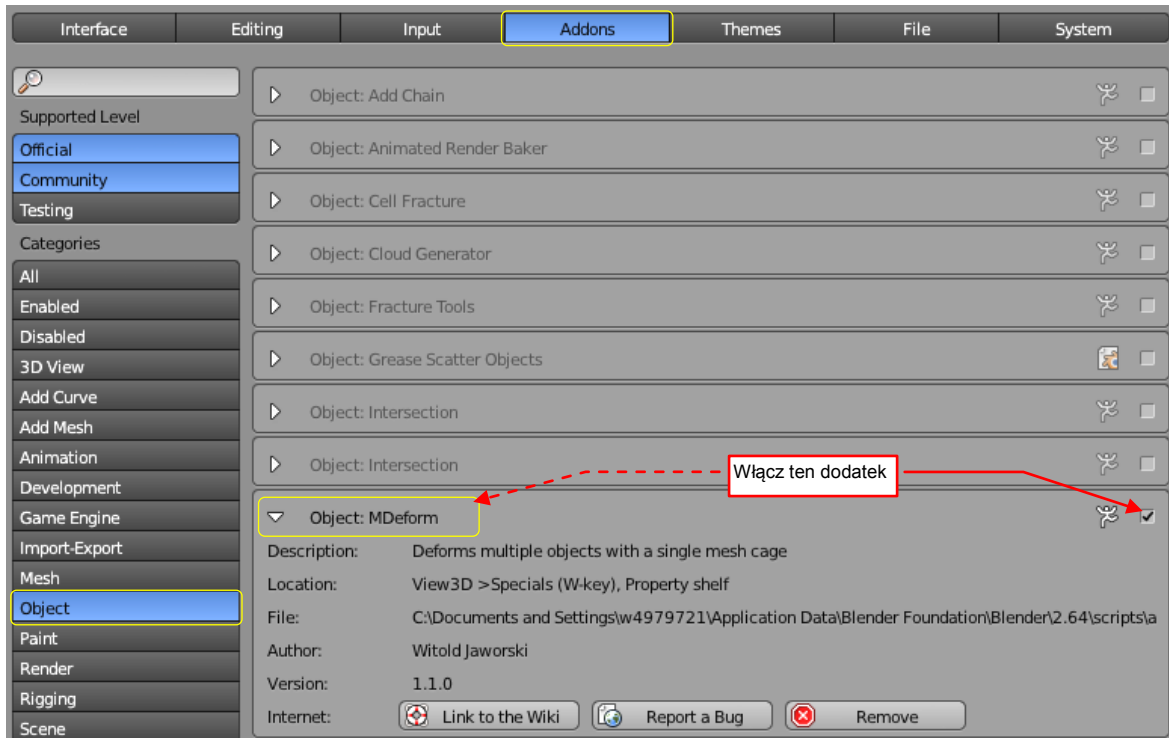
O wiele lepiej sprawdza się zwykłe „pudełko” lub kształt do niego zbliżony, o niewielkiej liczbie odpowiednio odległych ścian (Rysunek 8.15.6):



Rysunek 8.15.6 Działanie prostszej siatki deformującej

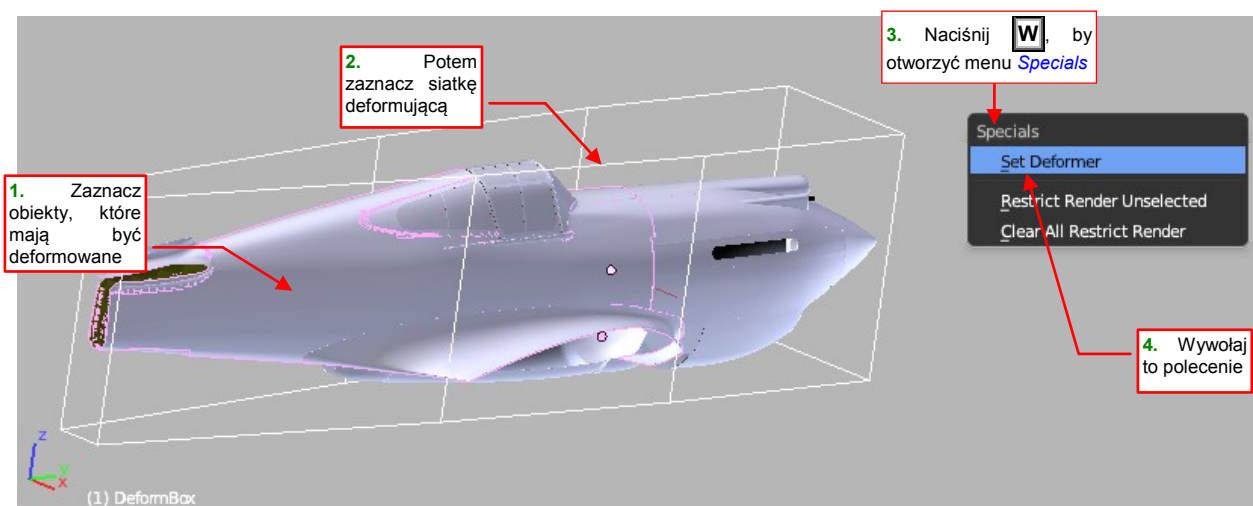
### 8.16 Deformacja wielu obiektów za pomocą siatki (dodatek *MDeform*)

Dodatek *MDeform* służy do obsługi grupy obiektów, przypisanych za pomocą modyfikatorów *Mesh Deform* do pojedynczej siatki („deformatora”). Towarzyszy tej książce jako plik `source\scripts\addons\object_mdeform.py` (por. str. 16). Znajdziesz go w sekcji *Addons* okna *User Preferences*<sup>1</sup>, wśród innych dodatków (Rysunek 8.16.1):



Rysunek 8.16.1 Uaktywnienie dodatku *MDeform* (okno *User Preferences*)

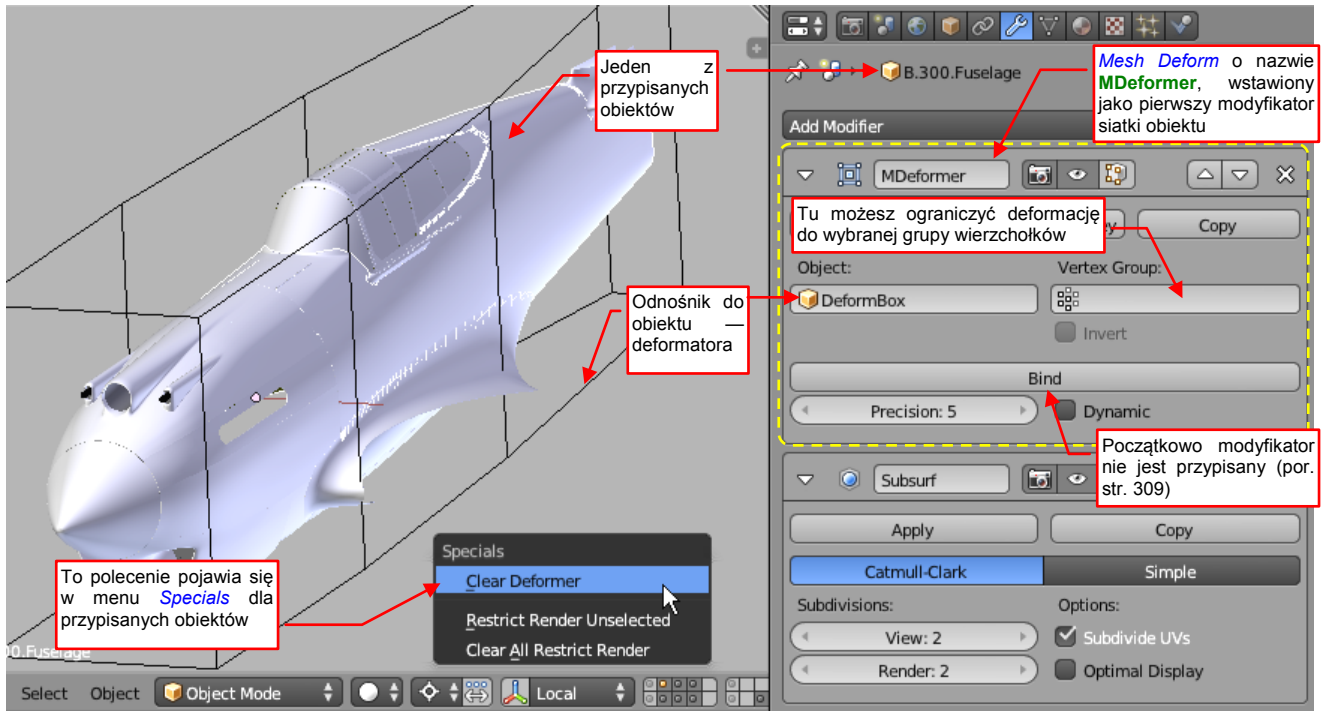
Pracę z dodatkiem rozpoczynasz w *Object Mode*, od przypisania do deformatora. Zaznacz najpierw obiekty deformowane, potem obiekt deformujący (aby stał się obiektem aktywnym) i z menu *Specials* (**W**) wywołaj polecenie *Set Deformer* (Rysunek 8.16.2):



Rysunek 8.16.2 Przypisanie obiektów do siatki deformującej

Polecenie *Set Deformer* dodaje na początek stosu modyfikatorów każdego z wybranych obiektów odpowiedni modyfikator *Mesh Deform* (Rysunek 8.16.3):

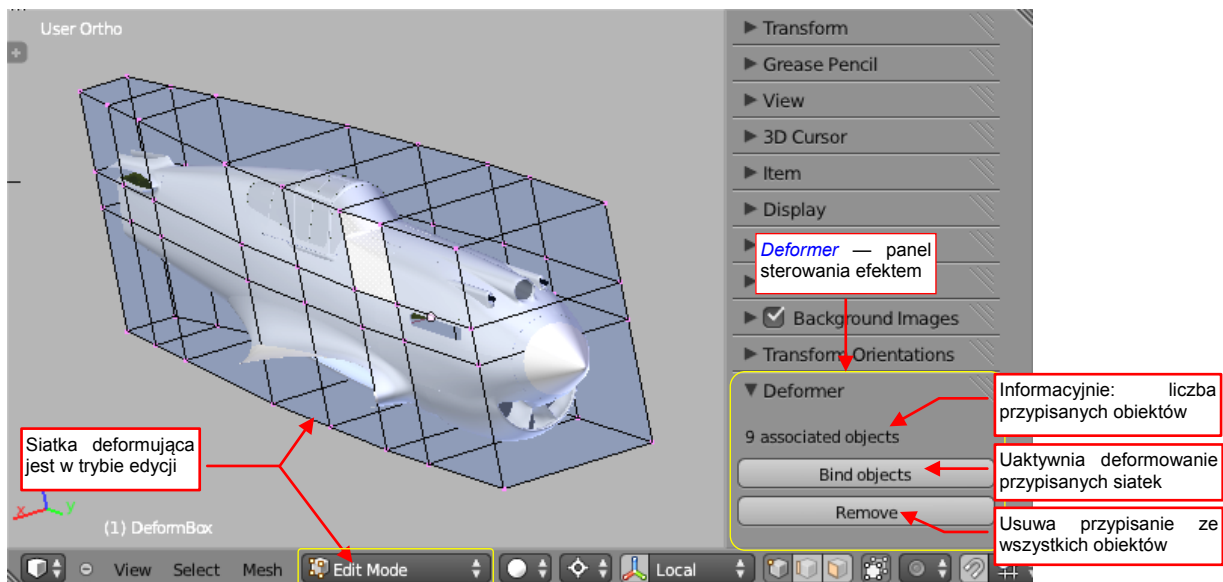
<sup>1</sup> Znajdziesz go tam, o ile dołączyłeś dostarczone wraz z tą książką dodatki do folderów z konfiguracją Blendera — por. str. 772 lub str. 753.



Rysunek 8.16.3 Rezultat polecenia **Set Deformer**

Poleceniem odwrotnym do **Set Deformer** jest **Clear Deformer**. Znajdziesz je w menu **Specials** dla obiektów, które są już przypisane do jakiegoś deformatora (Rysunek 8.16.3). Wywołanie **Clear Deformer** powoduje usunięcie ich modyfikatorów **MDeformer**.

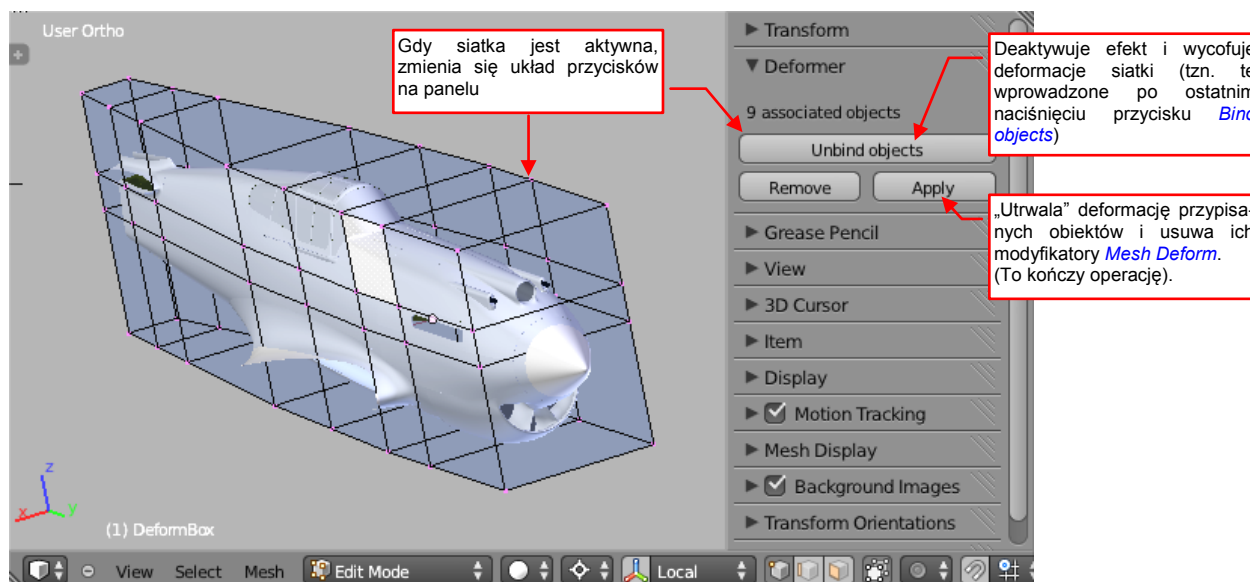
Gdy obiekty są już przypisane, przełącz siatkę deformującą w tryb **Edit Mode** (**Tab**). W tym trybie otwórz przycisk **Properties** (**N**) i znajdź w nim panel **Deformer** (Rysunek 8.16.4):



Rysunek 8.16.4 Panel sterowania efektem deformacji (siatka nieaktywna)

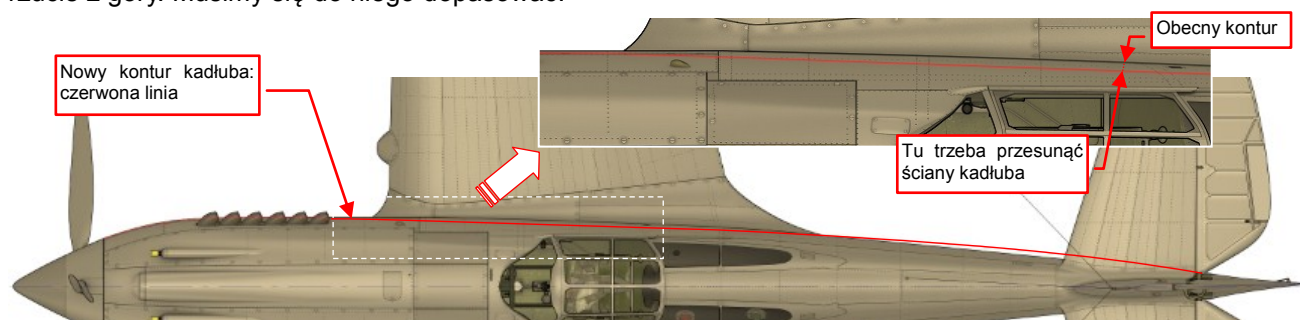
To panel wyświetlana przez dodatek **MDeform**. Gdy naciśniesz na niej przycisk **Bind objects**, uaktywnisz efekt deformacji. (To może zająć dłuższą chwilę, gdyż podczas tej aktywacji Blender przelicza wpływ wierzchołków siatki deformatora na poszczególne obiekty).

Po aktywacji panel **Deformer** ulega zmianie (Rysunek 8.16.5):



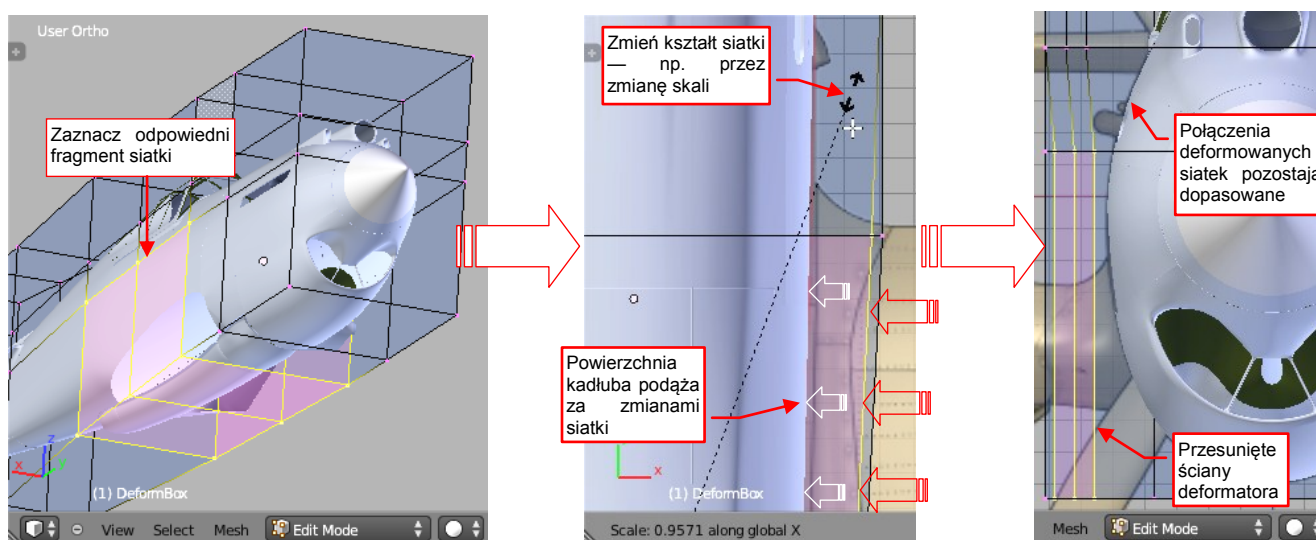
Rysunek 8.16.5 Panel sterowania efektem deformacji (siatka aktywna)

Dodatek [MDeform](#) używam zazwyczaj do poprawek kształtu takich złożonych obiektów jak kadłub. W tej sekcji pokażę jedną z takich sytuacji. Rysunek 8.16.6 przedstawia (naniesiony czerwoną linią) nowy obrys kadłuba w rzucie z góry. Musimy się do niego dopasować:



Rysunek 8.16.6 Rysunek referencyjny, do zmiany konturu kadłuba

Zmianę kształtu wprowadzamy, zmieniając położenie wierzchołków siatki sterującej (Rysunek 8.16.7):



Rysunek 8.16.7 Zmiana kształtu kadłuba

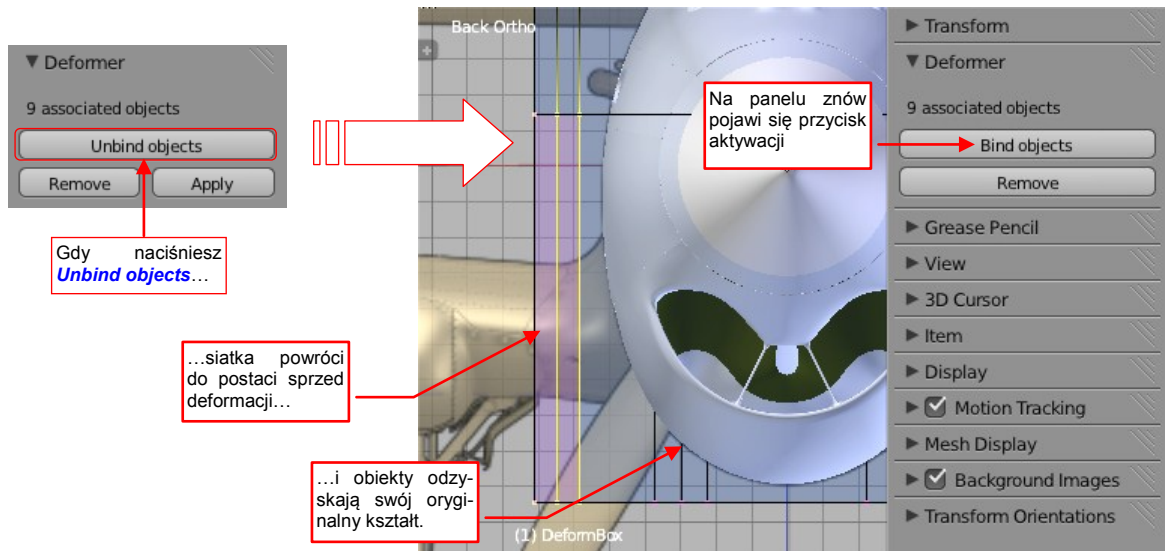
Aby przysunąć ściany kadłuba do nowego konturu, stopniowo zbliżam wybrane wierzchołki siatki do osi kadłuba (używając np. polecenia [Scale](#)). Mogłbym je także przesunąć lub obracać, gdyby to było potrzebne.



- Pamiętaj, aby na tym etapie nie wprowadzać żadnych zmian w topologii siatki deformatora (tzn. nie zmieniać liczby wierzchołków)! Gdy spróbujesz to zrobić, modyfikatory w przypisanych obiektach natychmiast przestaną działać.

Jeżeli po skorygowaniu kształtu kadłuba naciśniesz przycisk **Deformer:Apply** — zmiany zostaną „utrwalone” we wszystkich obiektach przypisanych do deformatora, a sama panel **Deformer** zniknie.

Jeżeli jednak chcesz jeszcze coś zmienić — na przykład przypisać kolejne obiekty, lub zmienić topologię siatki deformującej poprzez dodanie nowego segmentu — naciśnij przycisk **Unbind objects** (Rysunek 8.16.8):



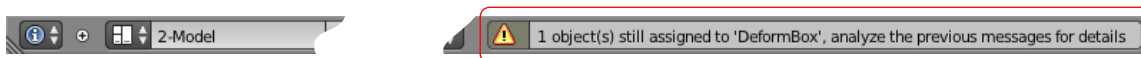
Rysunek 8.16.8 Deaktywacja siatki deformującej

Spowoduje to odtworzenie stanu sprzed deformacji (tzn. sprzed ostatniego wywołania polecenia **Bind objects**). Kształty przypisanych obiektów powrócą do swoich oryginalnych postaci, podobnie jak deformator.

Zazwyczaj trzeba przeprowadzić kilka prób, aby dobrze dopasować układu siatki deformującej do tak złożonej modyfikacji całego kadłuba. Do tego właśnie służą przełączniki **Bind/Unbind objects**.

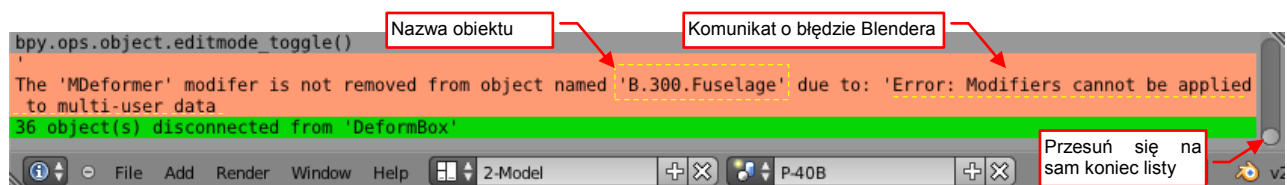
Gdy już będziesz zadowolony z rezultatu — „utrwal” go przyciskiem **Apply**. Możesz także zrezygnować z całej operacji naciskając przycisk **Remove**.

Czasami po naciśnięciu przycisku **Apply** możesz zobaczyć taki komunikat (Rysunek 8.16.9):



Rysunek 8.16.9 Informacja o problemach w „utrwalaniu” efektu deformacji

Sprawdź wtedy na końcu listy w oknie **Info**, z jakimi obiektami mamy kłopoty (Rysunek 8.16.10):



Rysunek 8.16.10 Identyfikacja „problemowego” obiektu na liście komunikatów programu

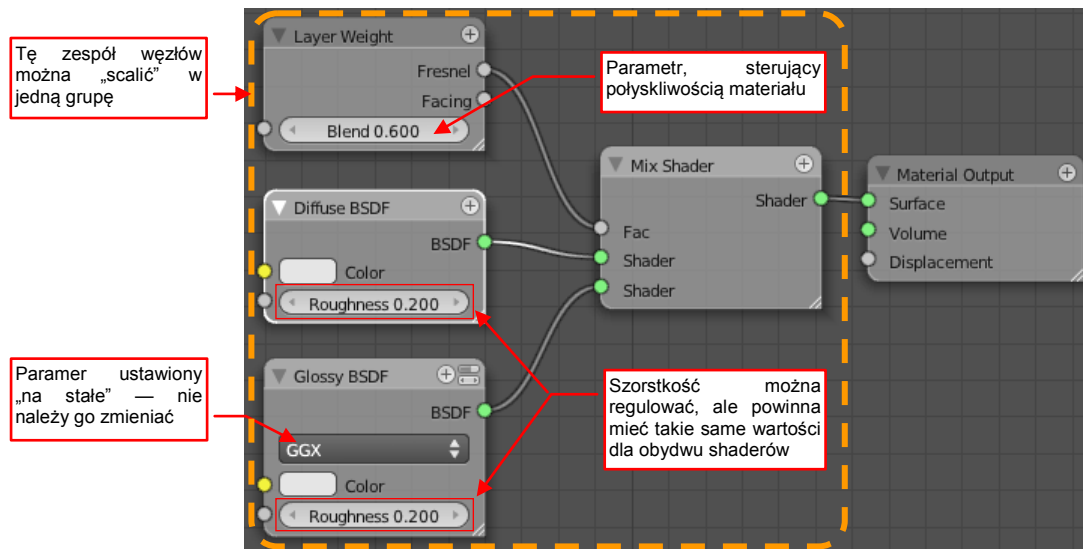
Zazwyczaj chodzi o próbę „utrwalenia” modyfikatora dla siatki współdzielonej przez dwa lub więcej obiektów. Należy stworzyć wówczas w odpowiednim obiekcie jej duplikat i powtórnie wywołać polecenie **Deformer:Apply**.

## **Rozdział 9. Blender — edytor materiałów (Cycles)**

Ten rozdział opisuje szczegóły definiowania materiałów i otoczenia dla silnika renderującego [Cycles](#). (Jest to renderer który wybrałem do użycia w tej książce — por. Wyciąg III). Jego materiały komponuje się głównie za pomocą węzłów ([nodes](#)), więc w tym rozdziale będziemy prawie przez cały czas korzystać z okna [Node Editor](#). (Wprowadzenie do [Node Editor](#) znajdziesz na str. 207).

## 9.1 Grupowanie węzłów

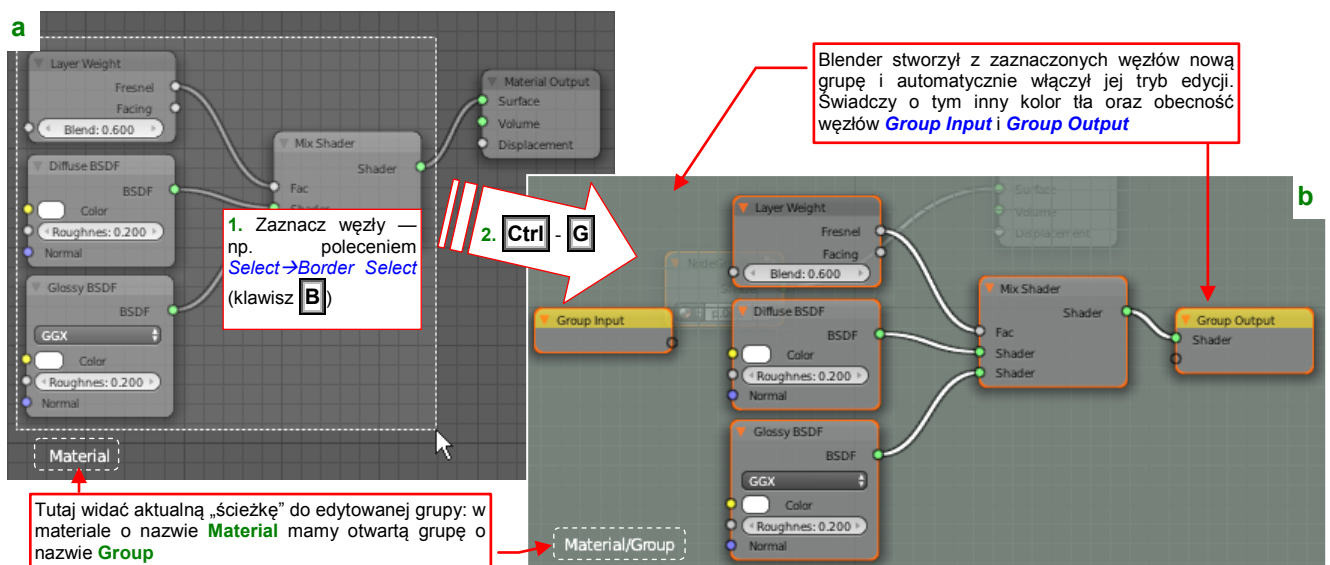
W Wyciągu III skomponowaliśmy materiał **B.Skin.Camouflage**, który ma odwzorować zewnętrzne powierzchnie samolotu (Rysunek 9.1.1):



Rysunek 9.1.1 Zespół węzłów, który zgrupujemy

To samo rozwiązanie powinniśmy wykorzystać także w drugim materiale, którym pokryte są powierzchnie wewnętrzne — **B.Skin.Inner**. Tyle, że tam materiał ma mieć inną barwę. Zamiast tworzyć od nowa ponownie ten sam schemat, możemy przekształcić zaznaczony na ilustracji zespół węzłów w tzw. grupę (*group*). Stworzymy w ten sposób nasz własny, wyspecjalizowany węzeł, który będzie można użyć w obydwu materiałach (Użycie grupy węzłów w materiale **B.Skin.Inner** — p. str. 322). Łączenie węzłów ma jeszcze tą zaletę, że gdy zdecydujemy się coś zmienić w tym zespole (np. zmienić typ odbłasków w shaderze *Glossy BSDF* z *CGX* na *Beckmann*) to ta zmiana będzie dotyczyć każdego materiału, w którym jest użyta ta grupa.

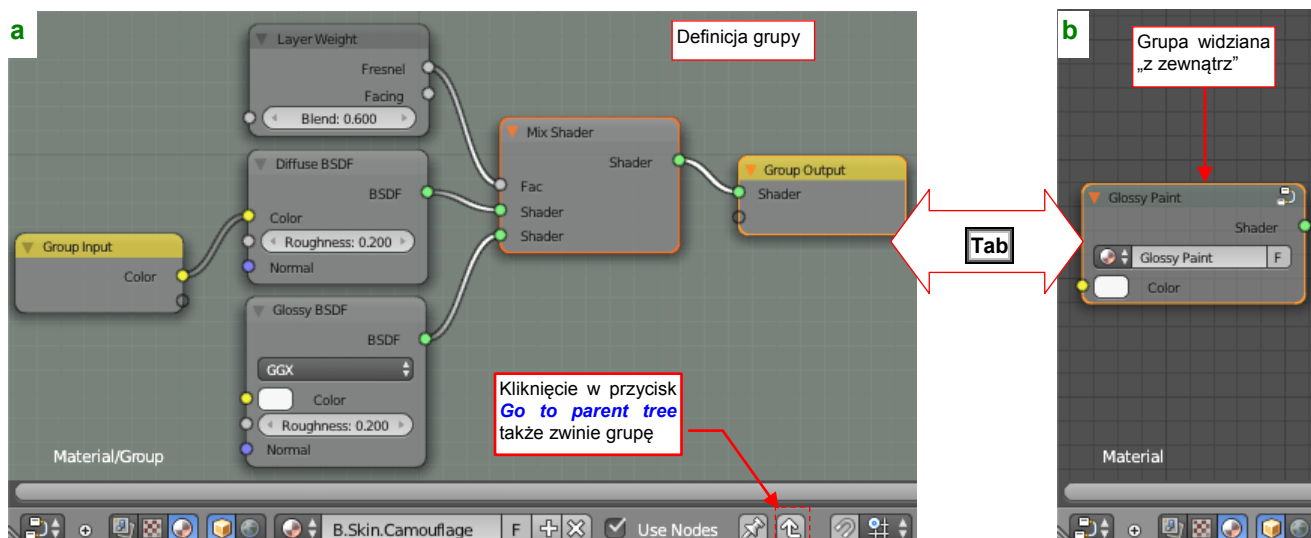
Aby stworzyć nową grupę, zaznacz obiekty które wchodzi w jej skład, np. obszarem prostokątnym (**B** — Rysunek 9.1.2a). Potem wywołaj polecenie **Node → Group → New Group** (**Ctrl-G**). Zaznaczone węzły zostaną połączone w nową grupę (Rysunek 9.1.2b):



Rysunek 9.1.2 Stworzenie nowej grupy węzłów

Blender od razu otwiera nową grupę w trybie edycji. Świadczy o tym inny kolor tła oraz obecność dwóch nowych węzłów: **Group Input** i **Group Output** (Rysunek 9.1.2b).

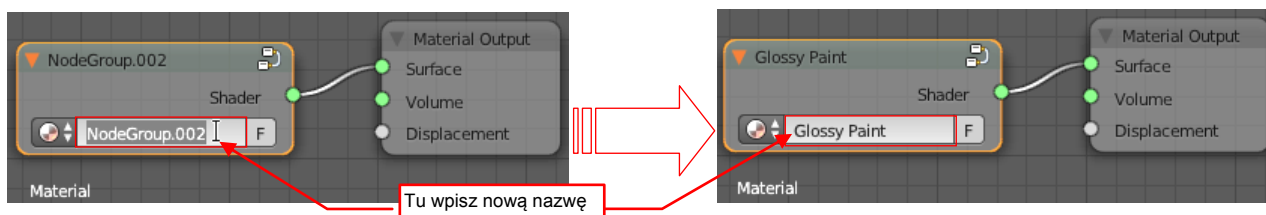
W każdej chwili możesz powrócić z definicji grupy do definicji materiału: wystarczy nacisnąć **Tab** lub przycisk **Go to parent tree** (Rysunek 9.1.3a):



Rysunek 9.1.3 Przełączanie się pomiędzy definicją grupy i definicją wykorzystującego ją materiału

Aby z powrotem przejść do definicji grupy należy zaznaczyć jej węzeł (Rysunek 9.1.3b) i nacisnąć **Tab** (lub wywołać polecenie **Node → Edit Group**).

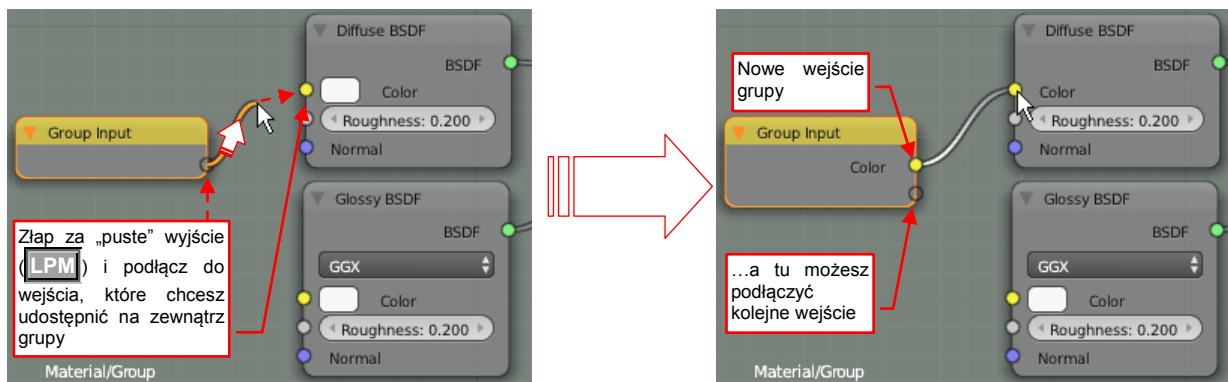
Od razu zmień w węźle grupy jej nazwę na jakąś bardziej odpowiednią (Rysunek 9.1.4):



Rysunek 9.1.4 Zmiana nazwy grupy

Grupa z tego przykładu będzie nam służyć jako shader powłok pomalowanych półmatową lub błyszczącą farbą. Dlatego proponuję ją nazwać **Glossy Paint**.

Powróćmy do definicji grupy. Przyjrzyj się dwóm węzłom o żółtych nagłówkach (por. Rysunek 9.1.3a): **Group Input** zawiera wejścia grupy, a **Group Output** — jej punkty wyjściowe. Na razie nasza grupa ma jedno wyjście (o nazwie **Shader**) i żadnych wejść. Umieścimy więc w **Group Input** kolor obydwu shaderów. Aby to uzyskać, podłącz wejście jednego z shaderów do „pustego” gniazda węzła **Group Input** (Rysunek 9.1.5):



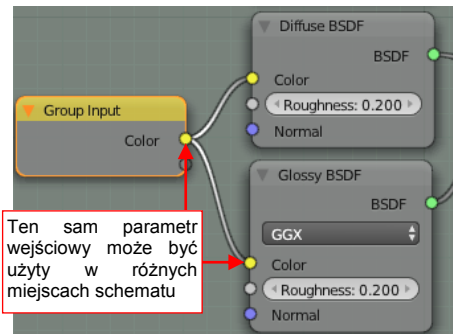
Rysunek 9.1.5 Definiowanie wejścia węzła grupy **Gloss Paint**

Zauważ, że Blender przesunął teraz puste wyjście węzła **Group Input** do dołu — ale nadal jest dostępne!

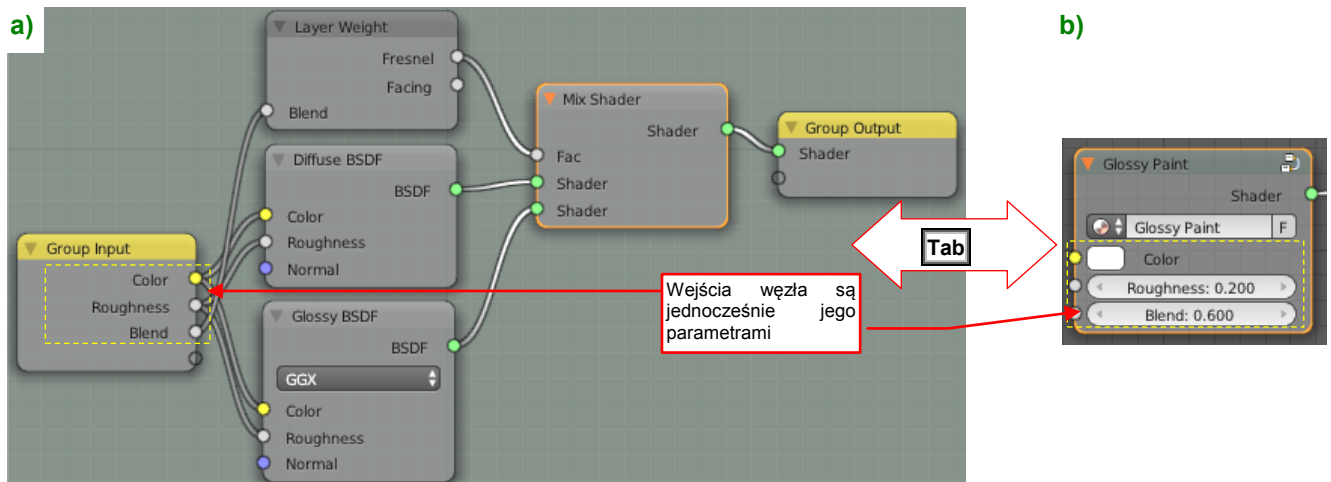


Założmy (tylko dla przykładu!) że obydwa shadery (*Glossy BxDF* i *Diffuse BxDF*) powinny mieć ten sam kolor. Możemy to teraz wymusić podłączając obydwa węzły do tego samego gniazda wejściowego (Rysunek 9.1.6).

W podobny sposób „wyciągnij” z obydwu shaderów parametr *Roughness* (jego wartości powinny być jednakowe). Na koniec przenieś także do parametrów wejściowych współczynnik *Layer Weight:Blend* (Rysunek 9.1.7a):



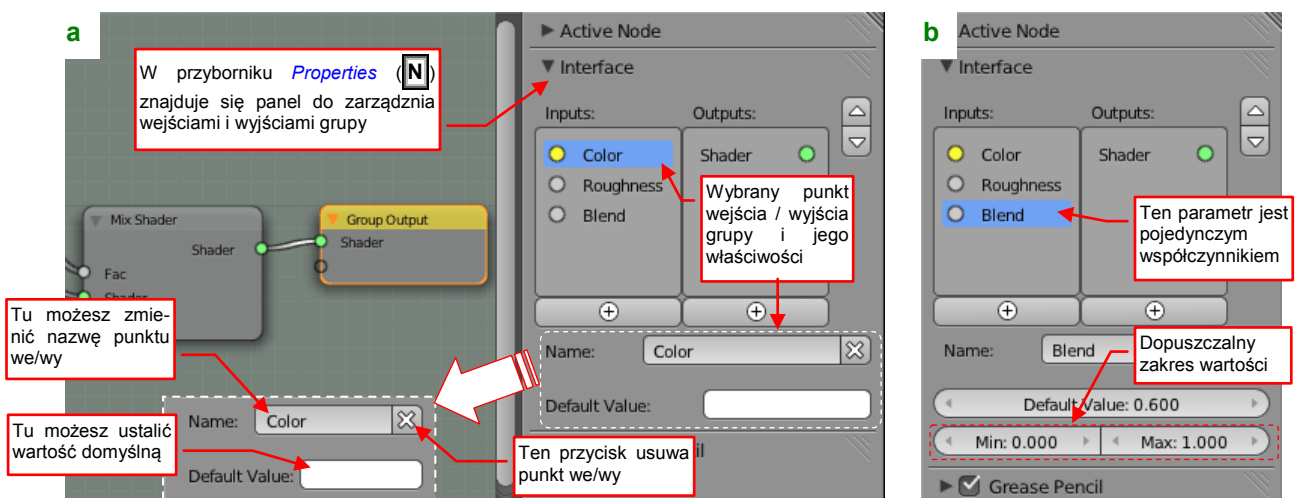
Rysunek 9.1.6 Synchronizacja ustawień shaderów



Rysunek 9.1.7 Ulepszony węzeł *Gloss Paint*

Gdy spróbujesz „zwinąć” definicję grupy przekonasz się, że „wyciągnięte” w ten sposób wejścia stały się parametrami jej węzła (Rysunek 9.1.7b).

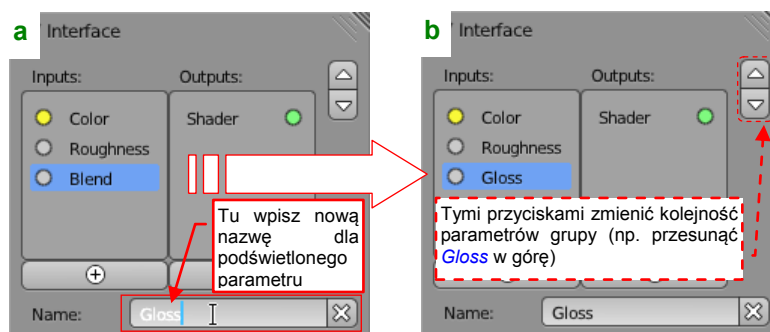
Podczas edycji grupy z prawej strony ekranu możesz otworzyć przybornik *Properties* (N). Tam, w panelu *Interface*, znajdziesz się kontrolki służące do zarządzania jej wejściami i wyjściami (Rysunek 9.1.8a). Gdy na liście *Inputs* lub *Outputs* wybierzesz (podświetlisz) jakiś punkt we/wy, u dołu panelu zobaczysz jego właściwości. Możesz tam zmienić jego nazwę lub go usunąć. W polu *Default Value* można ustawić domyślną wartość tego parametru (w przypadku pokazanym przez Rysunek 9.1.8a jest to barwa):



Rysunek 9.1.8 Panel *Interface* — zarządzanie wejściami i wyjściami grupy

Dla parametrów które są pojedynczą liczbą, Blender wyświetla w panelu *Interface* inny zestaw kontroltek (Rysunek 9.1.8b). Możesz tu ustalić dopuszczalny zakres wartości takiego współczynnika (pola *Min* i *Max*).

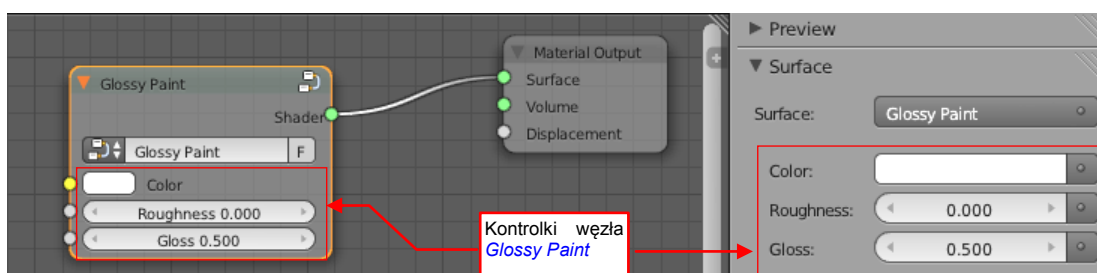
Patrząc na grupę przedstawioną przez Rysunek 9.1.7, wydaje mi się że jej użytkownik (czyli — ja sam, za parę miesięcy) może mieć problem ze zrozumieniem co właściwie zmienia parametr *Blend*. Na szczęście bez problemu możemy zmienić tę nazwę na inną. Wystarczy wpisać nową w polu *Interface:Name* (Rysunek 9.1.9a). W ten sam sposób możesz zmieniać także nazwy punktów wyjścia, z listy *Outputs:*. (W naszym przykładzie jest to jeden punkt, o domyślnej nazwie *Shader*. Myślę że nie pozostawia wątpliwości co do sposobu użycia, więc pozostawiamy ją bez zmian).



Rysunek 9.1.9 Zmiana nazwy parametru grupy

- Aby zmienić kolejność wyświetlania parametrów grupy, użyj strzałek umieszczonych z prawej strony listy *Inputs:Outputs* (Rysunek 9.1.9b).

Dzięki stworzeniu grupy schemat naszego materiału został zredukowany do dwóch węzłów i trzech parametrów, które są w nim istotne (Rysunek 9.1.10):

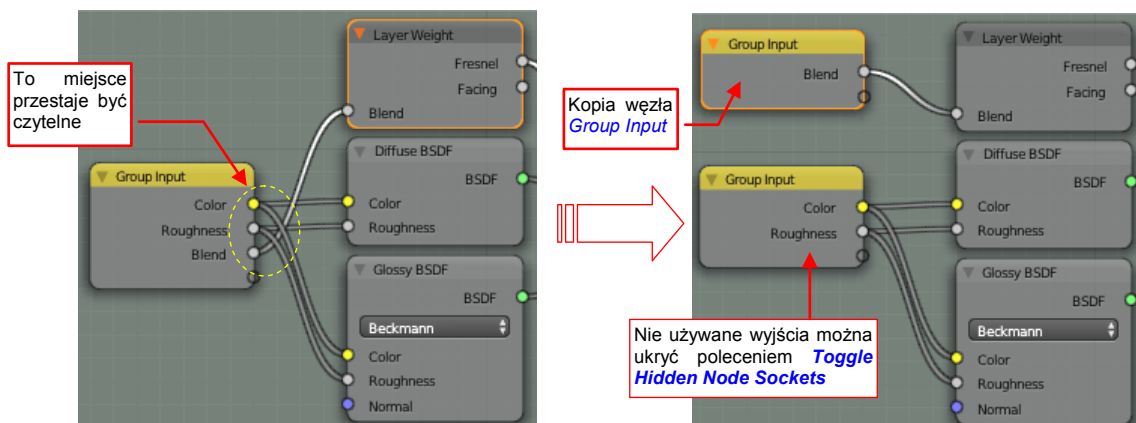


Rysunek 9.1.10 Węzeł *Glossy Paint* w oknie *Properties* (zestaw *Material*)

Zwróć uwagę, że w oknie *Properties* nawet zazwyczaj „zaśmiecona kontrolkami” panel *Material:Surface* wyświetla dla tego schematu wyłącznie parametry grupy *Glossy Paint*. Dzięki temu stała się prosta i czytelna.

- W każdym schemacie materiału są jakieś parametry, które można zmieniać, oraz inne, których nie należy ruszać bo się zepsuje cały efekt. Łączenie węzłów w grupy pozwala także wyróżnić i nadać bardziej odpowiednie nazwy tym wybranym parametrom. W ten sposób schemat staje się bardziej czytelny — bo użytkownik może traktować każdą grupę jak „czarną skrzynkę”, nie zagłębiając się w to, jak działa.

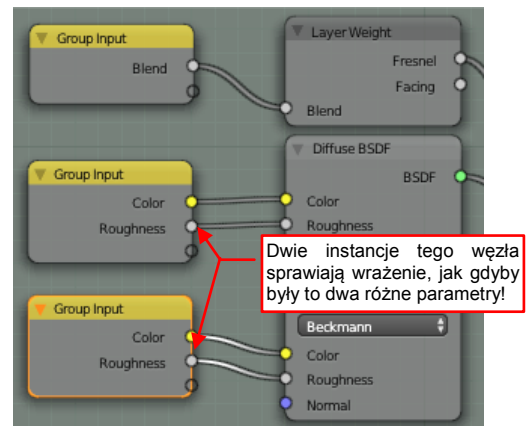
W schemacie grupy możesz stosować nie jeden, a kilka instancji *Group Input*. Przydaje się to gdy chcesz uniknąć efektu „spaghetti”: wielu krzyżujących się linii połączeń, które są trudne do śledzenia (Rysunek 9.1.11):



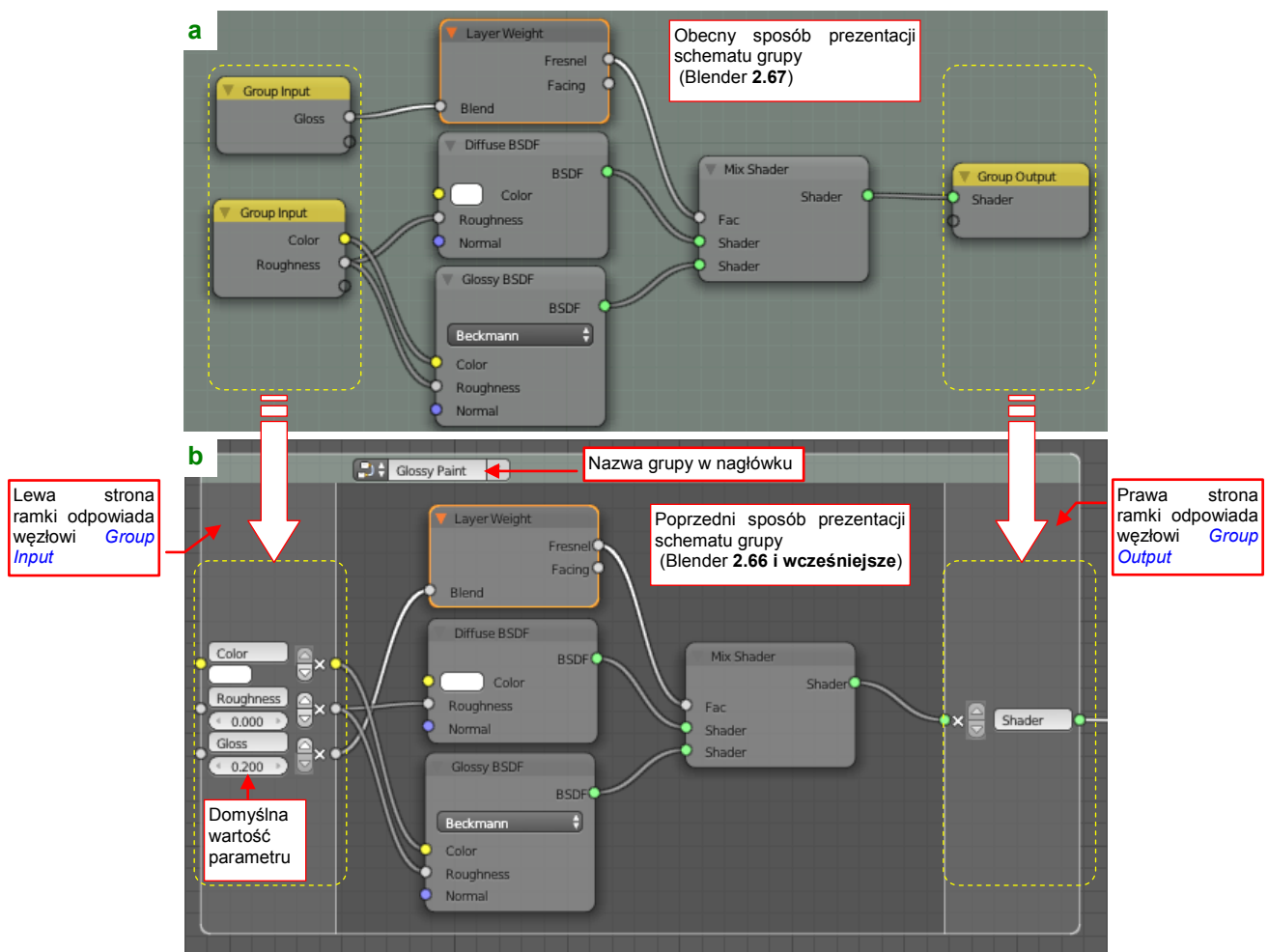
Rysunek 9.1.11 Uproszczenie układu połączeń wewnątrz grupy za pomocą drugiej instancji *Group Input*

Uważaj jednak by w takim upraszczaniu nie posunąć się zbyt daleko! Na przykład: unikaj wykorzystywania tego samego wyjścia w dwóch różnych instancjach *Group Input* (Rysunek 9.1.12). Choć z technicznego punktu widzenia jest to poprawne, wydaje mi się, że schemat jest mniej czytelny.

Przedstawiony w tej sekcji sposób prezentacji schematu grupy został wprowadzony dopiero w Blenderze 2.67. W wersji 2.66 i wcześniejszych zawartość edytowanej grupy była otoczana ramką (Rysunek 9.1.13b). Na lewym boku ramki były umieszczone gniazda wejściowe, a na prawym — wyjściowe. Umieszczone wewnątrz węzły były podłączane do tych gniazd. Rysunek 9.1.13 pokazuje porównanie obydwu sposobów wyświetlania zawartości grupy:



Rysunek 9.1.12 Użycie tego samego parametru w dwóch instancjach *Group Input*



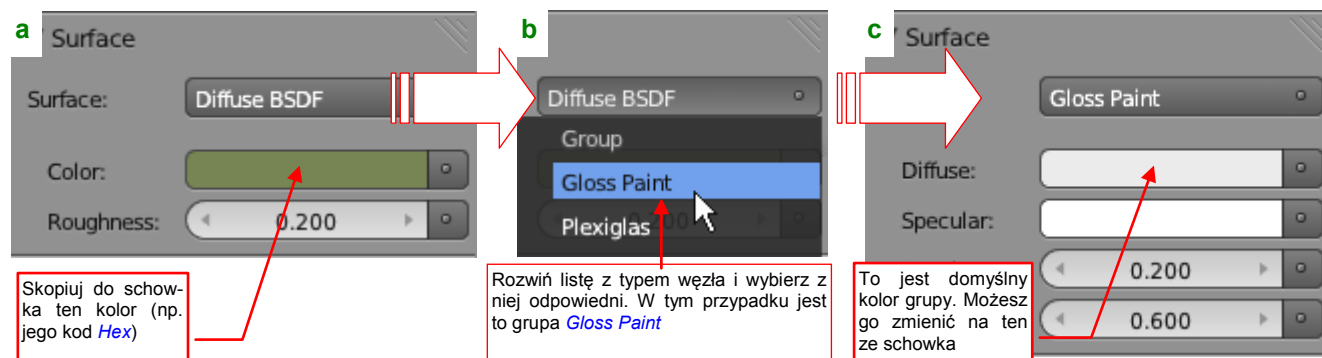
Rysunek 9.1.13 Porównanie „starej” i „nowej” metody prezentacji definicji grupy węzłów

Uważam że dla mniej złożonych grup węzłów „stary” sposób prezentacji (Rysunek 9.1.13b) był bardziej czytelny (przynajmniej w takim zakresie, jaki jest potrzebny dla ilustracji przedstawionych w tej książce). Widać było na nim nazwę grupy (w nagłówku ramki) i wartość domyślną każdego parametru. Otoczenie węzłów charakterystyczną ramką pozwala się natychmiast zorientować, że to jest definicja grupy. W dodatku większość ilustracji do tej publikacji tworzyłem w Blenderze 2.61 — 2.66. Dlatego wszystkie schematy grup węzłów pokazuję tu w „starym” układzie.

## 9.2 Wykorzystanie grupy węzłów

W sekcji 9.1 (str. 317) przekształciliśmy grupę węzłów w nowy element — shader o nazwie *Gloss Paint*. Początkowo jest używany tylko w materiale powierzchni zewnętrznych samolotu (*B.Skin.Camouflage*). W tej sekcji pokażemy, jak użyć tego shadera w materiale powierzchni wewnętrznych (*B.Skin.Inner*).

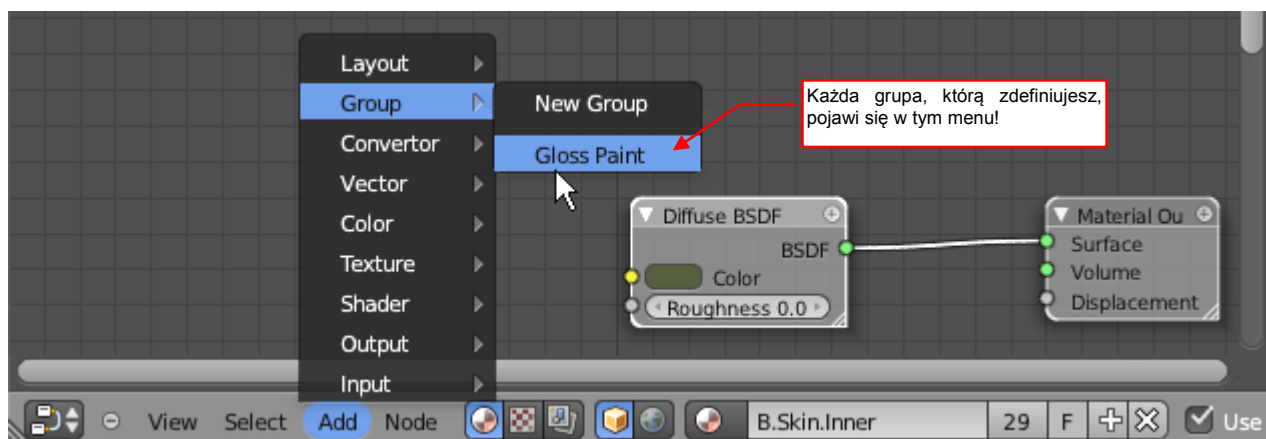
Zaznacz jakiś obiekt, który wykorzystuje materiał *B.Skin.Inner*. Gdy jest to jakiś prosty materiał, złożony z jednego shadera, najwygodniej jest go zmienić w zestawie *Material* okna *Properties* (Rysunek 9.2.1):



Rysunek 9.2.1 Zmiana typu węzła shadera na *Gloss Paint*

Zacznij od skopiowania koloru ze starego węzła<sup>1</sup> do schowka (Rysunek 9.2.1a) — chyba, że nie chcesz go zachować. Następnie rozwiń listę rozwijalną z typami shaderów (Rysunek 9.2.1b) i wybierz z niej grupę *Gloss Paint*. I to wszystko. No, może nie do końca: zwróć uwagę, że podstawił się nam domyślny kolor i inne ustawienia grupy (Rysunek 9.2.1c). Możesz teraz zmienić kolor *Glossy Paint:Diffuse*<sup>2</sup> na ten skopiowany do schowka.

Jeżeli schemat materiału jest bardziej złożony, i w oknie *Properties* nie możesz odnaleźć węzła, który chcesz zmienić, otwórz okno *Node Editor*. Poleceniem *Add → Group → Gloss Paint* dodaj do schematu tego materiału nowy węzeł (Rysunek 9.2.2):



Rysunek 9.2.2 Dodanie do schematu materiału węzła grupy *Gloss Paint*

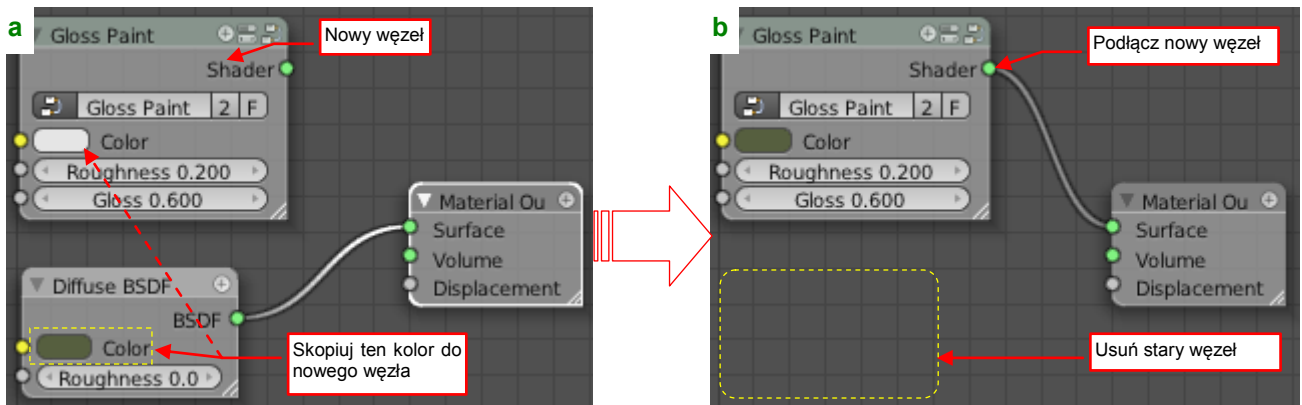
(Tak! Nie dziw się — Blender umieści w menu *Group* polecenie tworzące każdą grupę węzłów, którą masz zdefiniowaną w tym pliku *\*.blend*).

<sup>1</sup> Można to zrobić kopiując do schowka odczytany z kontrolki *Diffuse BSDF:Color* kod koloru w trybie *Hex* (por. str. 776); Można także otworzyć kontrolkę *Gloss Paint:Color* i wskazać jej ten kolor z ekranu (por. str. 775).

<sup>2</sup> Wklejając skopiowany przed chwilą kod *Hex* w odpowiednie pole kontrolki *Gloss Paint:Diffuse*.



Spowoduje to dodanie do schematu nowego, początkowo nie podłączonego węzła (Rysunek 9.2.3a):



Rysunek 9.2.3 Włączenie shadera *Gloss Paint* do schematu (na miejsce *Diffuse BSDF*)

Następnie zmień barwę *Diffuse* węzła *Gloss Paint* na kolor dotychczasowego węzła<sup>1</sup> (Rysunek 9.2.3a). Potem przenieś połączenia ze starego węzła na nowy, i ostatecznie usuń niepotrzebny element (Rysunek 9.2.3b).

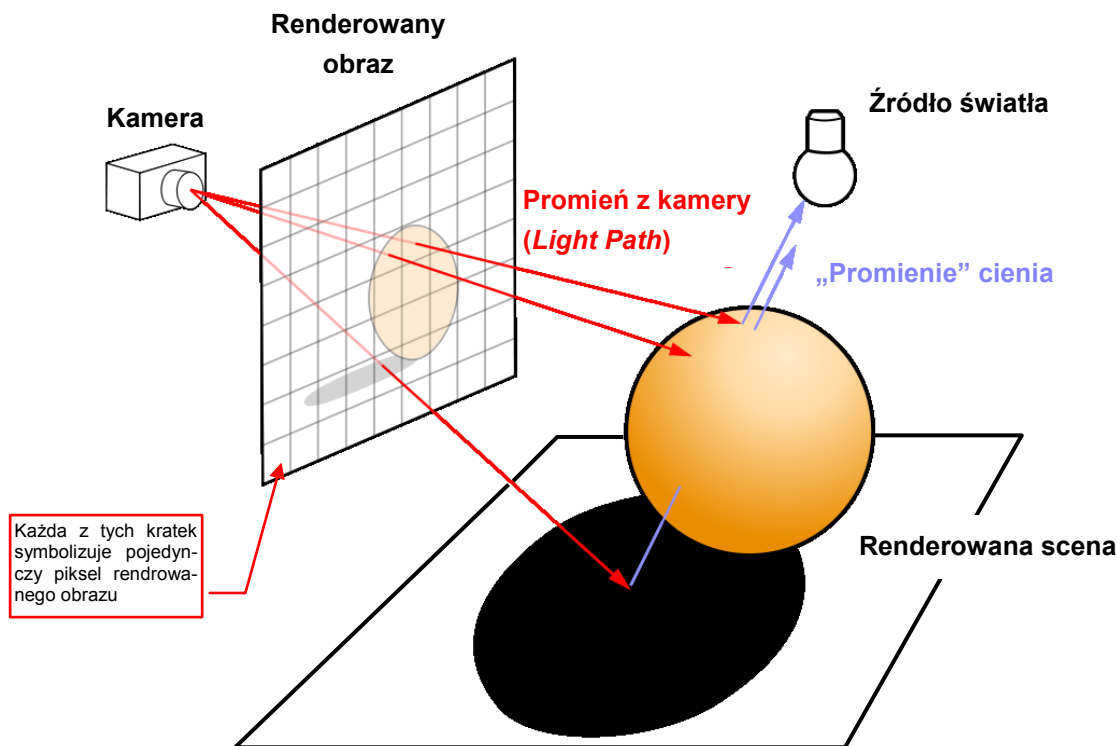
Pamiętaj, że wszystkie materiały w których używasz grupy *Glossy Paint*, odwołują się do tego samego bloku danych z jej definicją (zwróć uwagę na licznik użycia tej grupy — Rysunek 9.2.3b).

- Grupy są współdzielone pomiędzy materiałami. Oznacza to, że jakkolwiek zmiana w definicji grupy ma wpływ na wszystkie materiały, które jej używają.

<sup>1</sup> Gdy masz stary i nowy węzeł obok siebie, możesz rozwinąć kontrolkę barwy *Gloss Paint:Diffuse* i wskazać jej poprzedni kolor wprost z ekranu — klikając w odpowiednie pole starego węzła (por. str. 775).

### 9.3 Wykorzystywanie informacji o typie śledzonego promienia (*Light Path*)

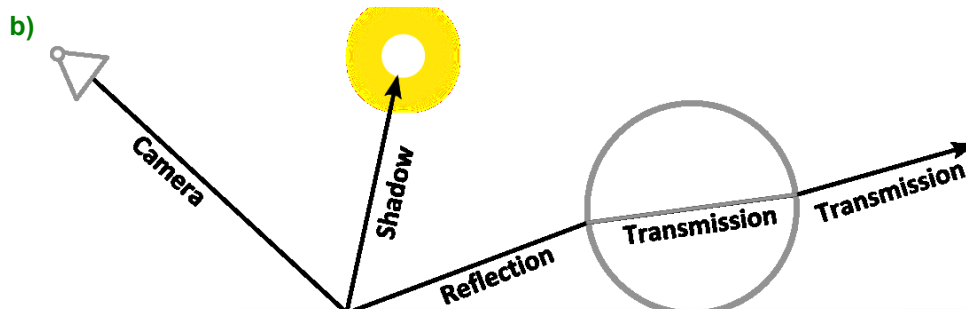
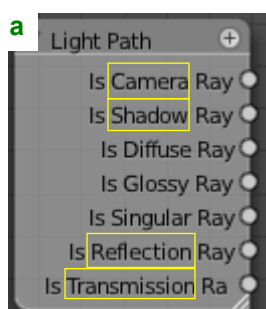
Cycles, podobnie jak większość współczesnych programów do renderowania, wykorzystuje algorytm „śledzenia promieni” (ang. *raytracing*). Teoretycznie program powinien śledzić promienie wychodzące ze źródła światła. Jednak większość z nich nigdy nie dotrze do kamery. Aby nie marnować czasu na ich obliczenia, program śledzi je w kierunku przeciwnym. Z każdego piksela renderowanego obrazu wyprowadza promień (określany także jako *light path*) w odwrotnym kierunku niż płynie światło z renderowanej sceny (Rysunek 9.3.1):



Rysunek 9.3.1 Podstawowa zasada działania renderera (wg [wikipedia.org](http://wikipedia.org))

Czasami promień nie napotka niczego po drodze, innym razem zostanie kilka razy odbity (ang. *bounced*) od obiektów wchodzących w skład sceny. Gdy natknie się na jakąś powierzchnię, renderer prowadzi pomocniczy promień wprost do źródła światła. Jeżeli napotka po drodze jakiś obiekt — piksel jest identyfikowany jako cień (Rysunek 9.3.1). W przypadku powierzchni połyskliwych lub przezroczystych śledzenie na tym się nie kończy, bo „podstawowy” promień ulega odbiciu i/lub załamaniu i trzeba go śledzić dalej. (W Cycles określasz max. liczbę śledzonych odbić w zestawie *Render*, panelu *Integrator*).

Węzeł *Light Path* umożliwia uzyskiwanie w Cycles różnych „sztuczek”. Każdy z jego punktów wyjściowych (Rysunek 9.3.2a) zwraca wartość = 1.0 dla piksela obrazu, którego promień spełnia odpowiedni warunek. Dla pozostałych pikseli węzeł zwraca wartość = 0.0. Te wartości można wykorzystać jako współczynniki dla jakichś innych operacji. Rysunek 9.3.2b) przedstawia używane w Cycles nazwy poszczególnych typów promieni:



Rysunek 9.3.2 Typy promieni, rozróżniane w Cycles (wg [wiki.blender.org](http://wiki.blender.org))

Punkty wyjściowe węzła **Light Path** zwracają wartość 1.0 dla następujących pikseli obrazu:

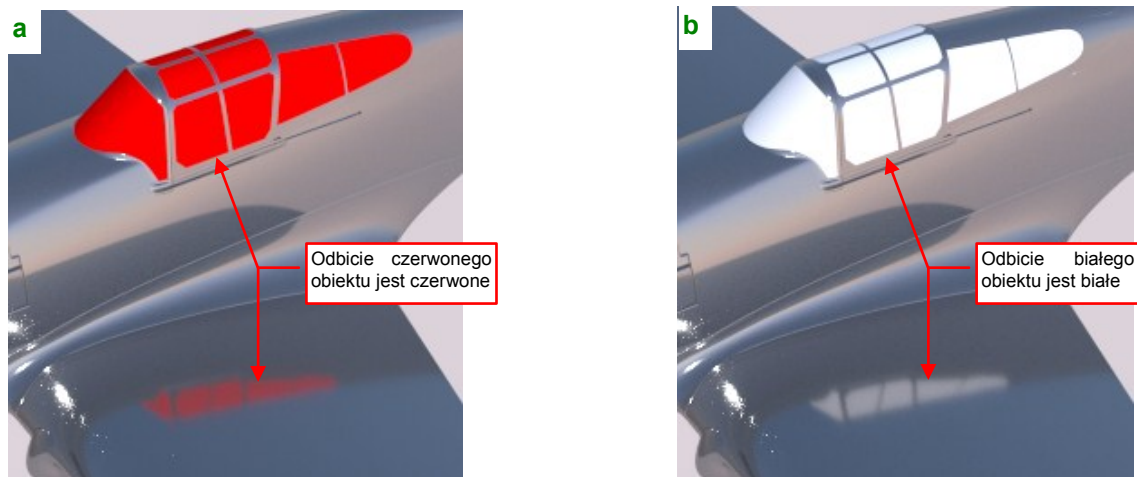
- **Is Camera Ray**: piksel tła. Jego promień nie napotkał „po drodze” żadnego elementu sceny;
- **Is Shadow Ray**: piksel wchodzący w skład jakiegoś cienia;
- **Is Reflection Ray**: piksel odbity od jakiejś powierzchni;
- **Is Transmission Ray**: piksel z jakiejś powierzchni przezroczystej;

Pozostałe trzy wyjścia węzła (por. Rysunek 9.3.2a) pozwalają wyróżnić spośród pikseli spełniających warunek **Is Reflection Ray** lub **Is Transmission Ray** następując przypadki:

- **Is Diffuse Ray**: piksel pochodzący z odbicia światła rozproszonego;
- **Is Glossy Ray**: piksel pochodzący z odbicia „lustrzanego”;
- **Is Singular Ray**: piksel należy do bezpośredniego odbicia promienia światła;

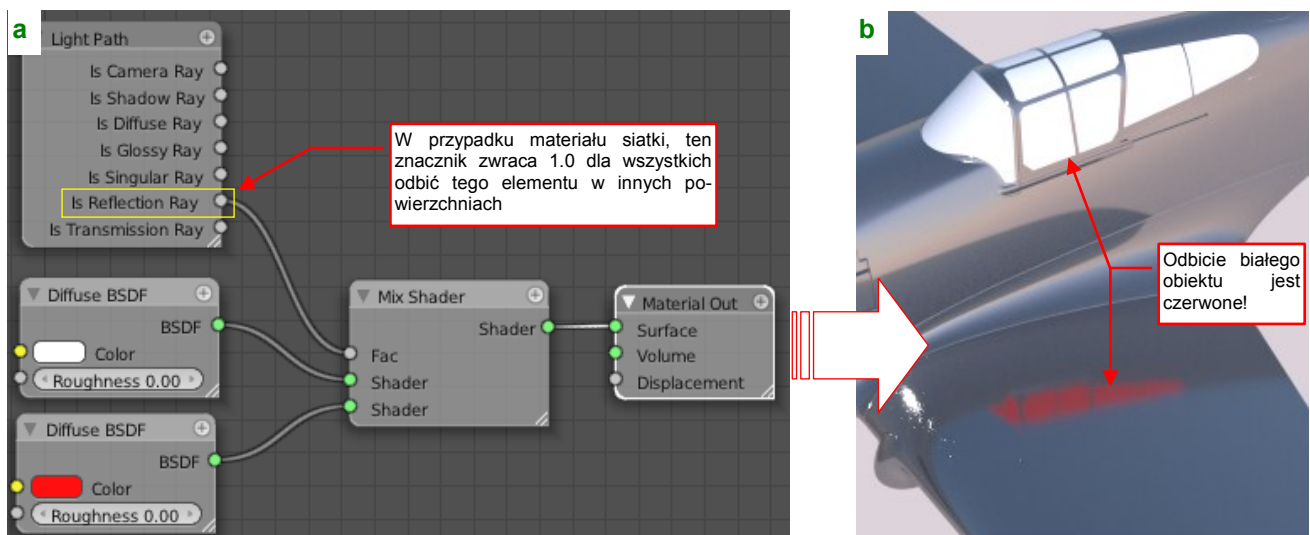
Z opisu na [wiki.blender.org](http://wiki.blender.org) można się domyślać, że wyjście **Is Singular Ray** powinno zwracać wartość 1.0 dla ostrych, punktowych odbić promieni słonecznych. Nie jestem jednak tego pewien, bo nie udało mi się uzyskać takiego efektu. (Może nie była w chwili pisania tego fragmentu książki implementowana? W każdym razie sprawdzałem w Blenderze 2.61).

Najprostszym przykładem „sztuczki” z udziałem węzła **Light Path** może być efekt „podstawienia odbicia”. W połyskliwej powłoce płata czerwona osłona kabiny odbija się na czerwono, a biała — na białą (Rysunek 9.3.3):



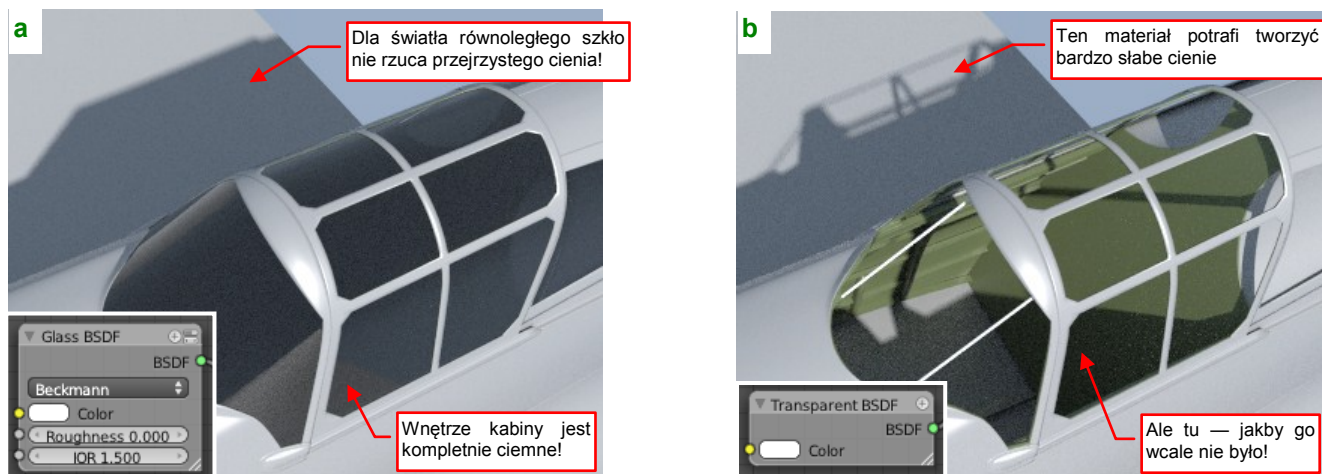
Rysunek 9.3.3 Kolor odbicia owiewki kabiny na powierzchni skrzydła

Gdy jednak zmieszasz tę czerwień i biel ze sobą wykorzystując wagę **Light Path: Is Reflection Ray** (Rysunek 9.3.4a), uzyskasz czerwony cień białego obiektu (Rysunek 9.3.4b):



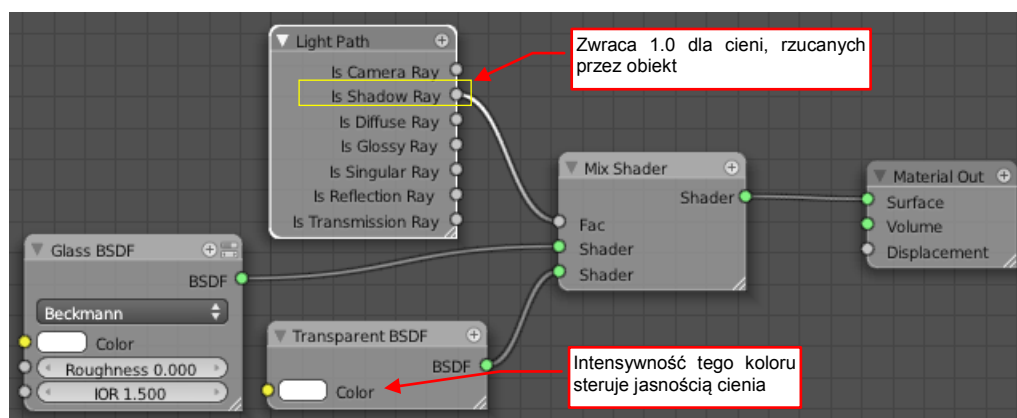
Rysunek 9.3.4 Złożenie białego materiału z czerwonym odbiciem

W podobny sposób jak odbicie można zamienić cień. W kolejnym przypadku mamy dwa shadery: szkło ([Glass BSDF](#)), które przy wyłączonej kaustyce rzuca lity cień (Rysunek 9.3.5a), oraz „pełną przejrzystość” ([Transparent](#)), która rzuca przejrzyste cienie (Rysunek 9.3.5b):



Rysunek 9.3.5 Materiały do złożenia

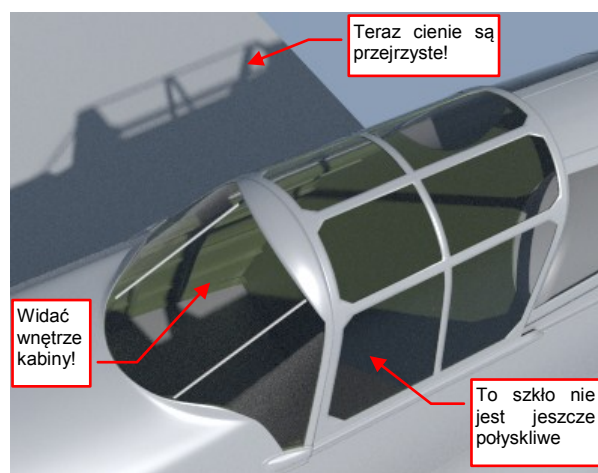
Złącz te dwie powierzchnie ze sobą w [Mix Shader](#), wykorzystując jako współczynnik udziału wyjście [Light Path: Is Shadow Ray](#) (Rysunek 9.3.6):



Rysunek 9.3.6 Schemat materiału

W efekcie owiewka kabiny, mimo wyłączonej kaustyki (por. zestaw [Render, Integrator:No Caustics](#)) rzuca przejrzyste cienie (Rysunek 9.3.7). Pozwala to także zobaczyć zasłonięte przez nią wnętrze kabiny, które wcześniej było zupełnie czarne. (Domyślna barwa szkła, które pokazuje Rysunek 9.3.7, jest szara. Wnętrze kabiny będzie jeszcze jaśniejsze, gdy zmienisz ją na białą).

Uważam, że nawet gdy przyszłe wersje Cycles będą już poprawnie obsługiwać kaustykę dla światła równoległego, warto pamiętać o tej „sztuczce”. Wyłączenie kaustyki eliminuje większość „białego szumu” z nieprzejrzystych powierzchni i przyspiesza renderowanie połyskliwych powłok — takich jak wypolerowane duraluminium. W scenach lotniczych nie potrzebujemy dokładnych i skomplikowanych cieni za szklanymi powierzchniami, jaki daje wyłączenie opcji [No Caustics](#). Efektu „zamiany cienia” można użyć dla optymalizacji czasu renderowania.

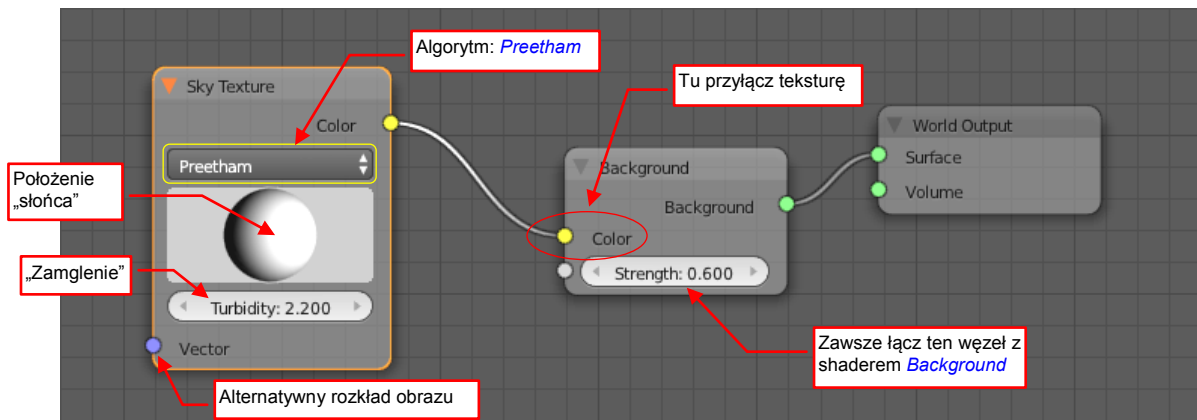


Rysunek 9.3.7 Rezultat



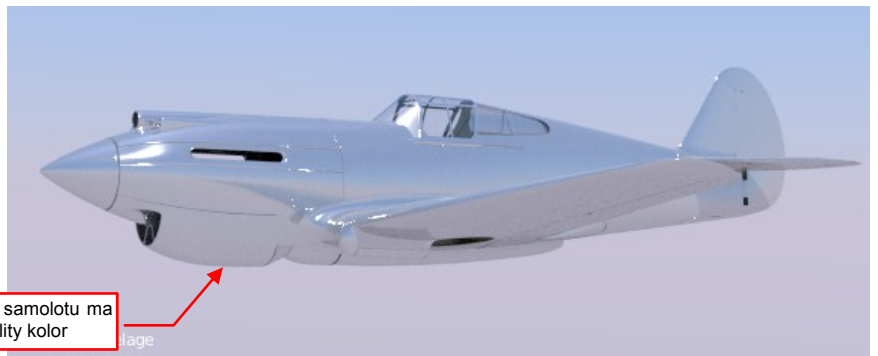
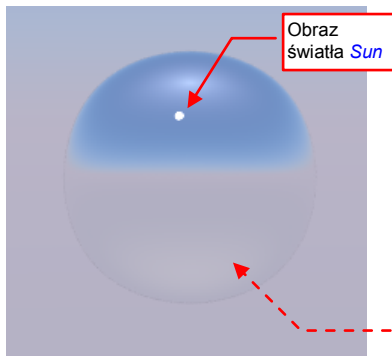
### 9.4 Użycie prostego obrazu nieba (*Sky Texture*)

Efekt *Sky Texture* to w Cycles programowo generowana panorama bezchmurnego nieba. Wstawia się ją do schematu w *Node Editor*, poleceniem **Add→Texture→Sky Texture**. Należy ją dołączyć do wejścia **Color** shadera **Background**. Rysunek 9.4.1 przedstawia ją w trybie *Preetham* (to nazwa algorytmu):



Rysunek 9.4.1 Użycie węzła *Sky Texture* w schemacie otoczenia (*World*)

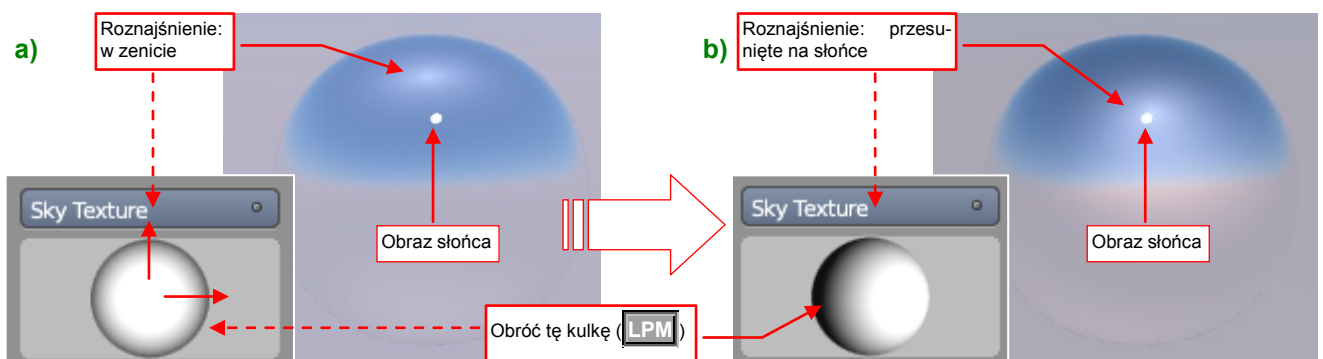
Rysunek 9.4.2 przedstawia obraz tekstury, uzyskany na powierzchni kulistej sondy (por. str. 330). Górna część otoczenia ma błękitny kolor nieba. Nieco powyżej „linii horyzontu” ta barwa płynnie przechodzi w jakiś pastelowy, jednolity kolor. Większość scen rozgrywa się na ziemi, i w takich zastosowaniach większej części tej dolnej półsfery po prostu nie widać. Dla ujęć w locie horyzont tej panoramy znajduje się zbyt wysoko (Rysunek 9.4.3):



Rysunek 9.4.2 Podgląd z użyciem sondy

Rysunek 9.4.3 Model w otoczeniu wykreowanym przez *Sky Texture*

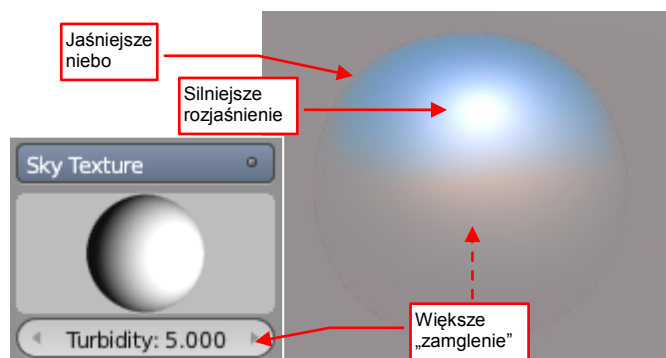
Na błękitnym „niebie” *Sky Texture* znajduje się (domyślnie w zenicie) rozjaśnienie, odpowiadające słońcu (Rysunek 9.4.4a). Położeniem tego rozjaśnienia możesz sterować za pomocą „kulki”, umieszczonej w węźle. Wystarczy ją „złapać” **LPM** i przeciągnąć, aby przesunąć to rozjaśnienie ponad odbity przez powierzchnię sondy obraz słońca (obiektu typu **Sun** — Rysunek 9.4.4b):



Rysunek 9.4.4 Dopasowanie najjaśniejszego miejsca nieba do aktualnej pozycji słońca

Lepiej zawsze mieć te dwa źródła światła odpowiednio zsynchronizowane.

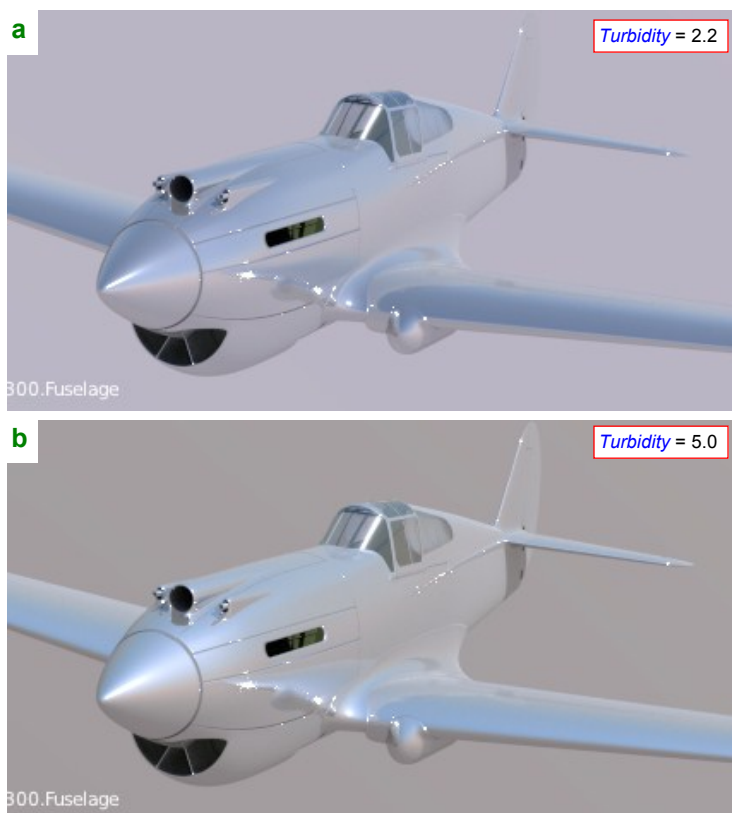
Drugi parametr tekstury — **Turbidity** — służy do sterowania efektem, określanym jako „zamglenie”. Domyślna wartość wynosi 2.2 (por. Rysunek 9.4.1). Gdy ją zwiększysz, górna półsfera ulegnie rozjaśnieniu a dolna — pociemnieje. Obszar odpowiadający położeniu słońca stanie się większy i bielszy (Rysunek 9.4.5).



Rysunek 9.4.5 Działanie parametru **Turbidity**

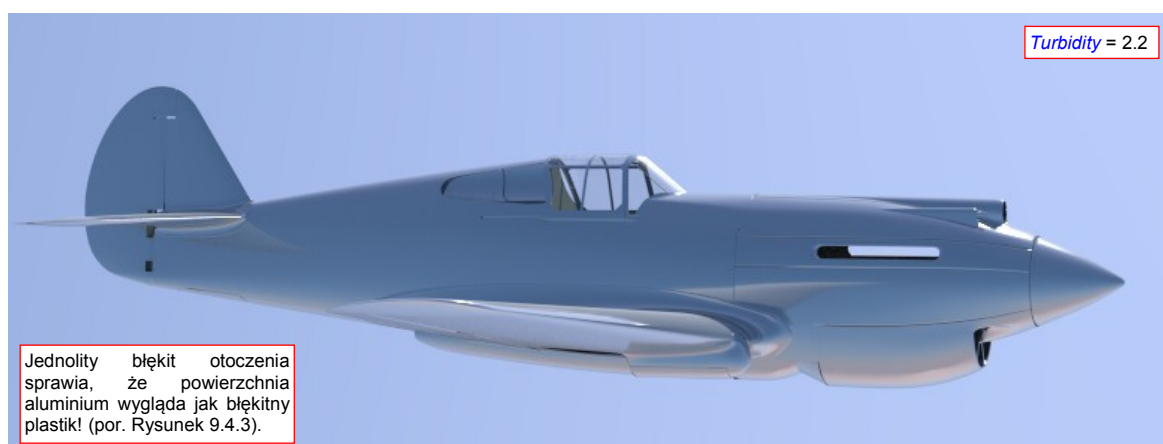
Wpływ zmiany tego parametru na połyskliwą („alumiową”) powierzchnię modelu pokazuje Rysunek 9.4.6. Domyślne **Turbidity** = 2.2 odpowiada warunkom „czystego nieba” (Rysunek 9.4.6a). Jednocześnie zbyt wysokie położenie horyzontu podstawia za modelem i w odbiciach na powierzchniach bocznych za dużo pastelowej barwy dolnej półsfery. Takie zabarwienie może mieć samolot oświetlony promieniami wschodzącego/zachodzącego słońca. (Oczywiście, trzeba by tu jeszcze zmienić barwę źródła światła **Sun** z białej na pomarańczową).

Przy zwiększonym zamgleniu (**Turbidity** = 5.0) powierzchnia modelu staje się jaśniejsza, a barwy — mniej zróżnicowane. Dolna półsfera stała się nieco ciemniejsza, ale nie zmieniło to specjalnie odcienia odbijanego przez dół i boki naszego testowego samolotu (Rysunek 9.4.6b).



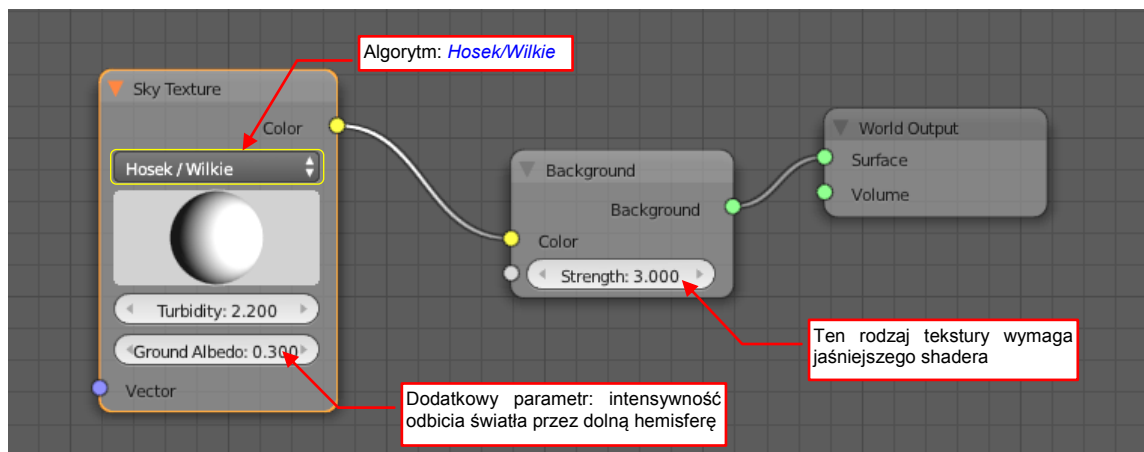
Rysunek 9.4.6 Wpływ zmiany parametru **Turbidity** na wygląd modelu (**Background:Strength** = 0.6)

Aby obniżyć linię horyzontu **Sky Texture**, wykorzystaj węzeł **Vector→Mapping** (por. str. 332). Tylko wtedy można uzyskać ujęcie z boku na tle błękitnego nieba (Rysunek 9.4.7):



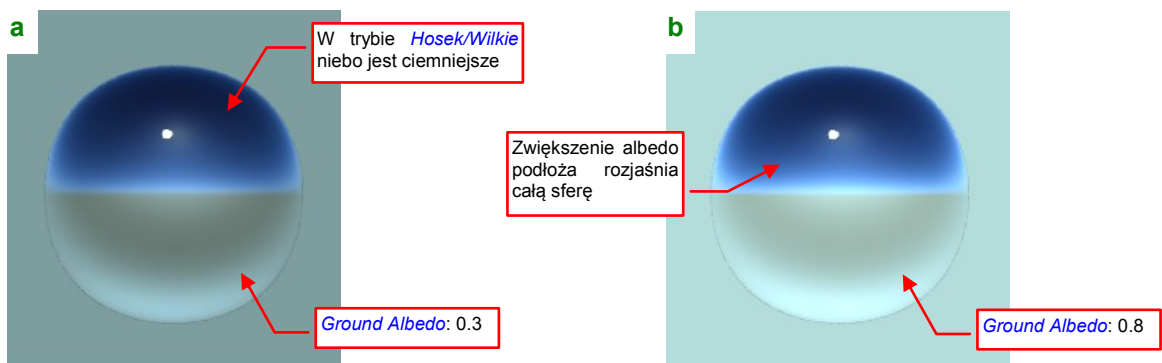
Rysunek 9.4.7 Przykładowe ujęcie modelu z boku, po obniżeniu linii horyzontu (tryb **Preetham**)

Gdy przełączysz się na alternatywny obraz nieba (**Hosek/Wilkie**) — w węźle **Sky Texture** pojawi się dodatkowy parametr (**Ground Albedo** — Rysunek 9.4.8):



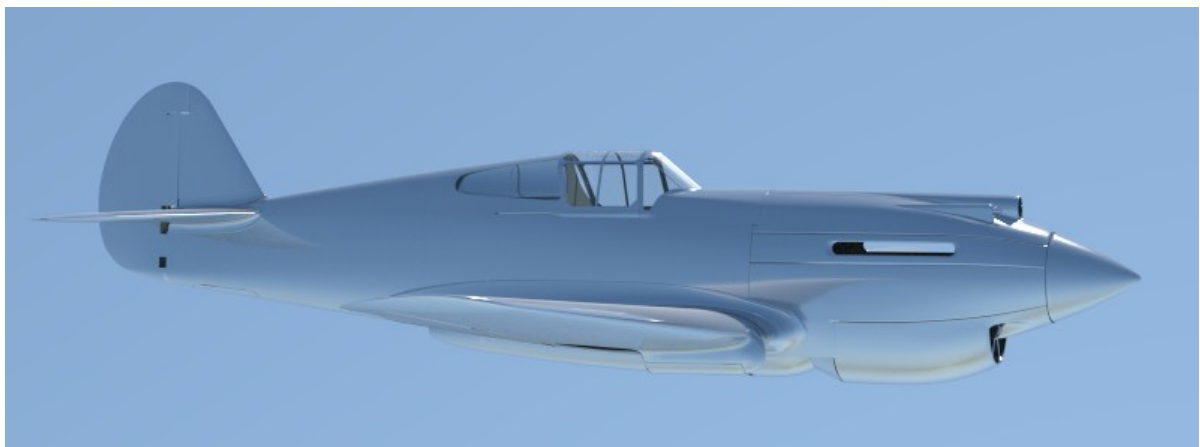
Rysunek 9.4.8 Użycie węzła **Sky Texture** w trybie **Hosek/Wilkie**

To współczynnik odbicia światła przez „ziemię” — czyli dolną półsferę panoramy. Zwiększenie tej wartości rozjaśnia nieco panoramę (por Rysunek 9.4.9), choć i tak algorytm **Hosek/Wilkie** tworzy ciemniejszy obraz od algorytmu **Preetham** (por Rysunek 9.4.9a) i Rysunek 9.4.2):



Rysunek 9.4.9 Wpływ parametru **Ground Albedo** na panoramę **Sky Texture**

Rysunek 9.4.10 przedstawia test naszego modelu w tej drugiej odmianie panoramy **Sky Texture**:



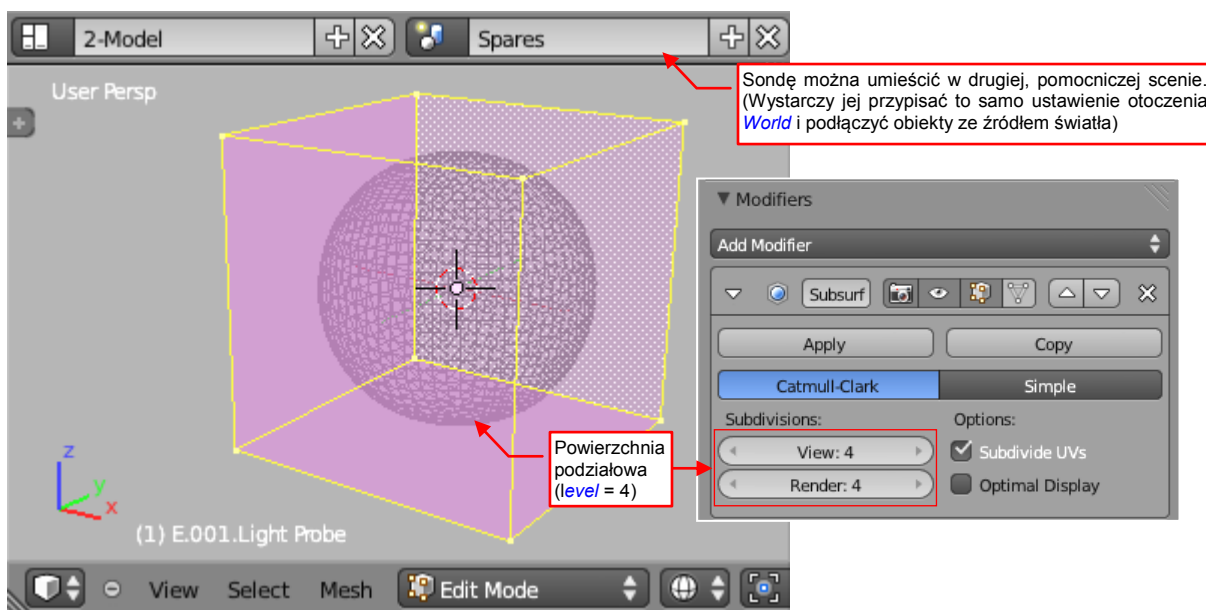
Rysunek 9.4.10 Przykładowe ujęcie modelu z boku, po obniżeniu linii horyzontu (tryb **Hosek/Wilkie**)

Wydaje się, że niebo stworzone za pomocą algorytmu **Hosek/Wilkie** jest bardziej niebieskie od nieba wygenerowanego za pomocą metody **Preethama**. W dokumentacji Blendera podano także, że **Hosek/Wilkie** lepiej się nadaje do scen zachodu lub wschodu słońca.

## 9.5 Sonda obrazu otoczenia

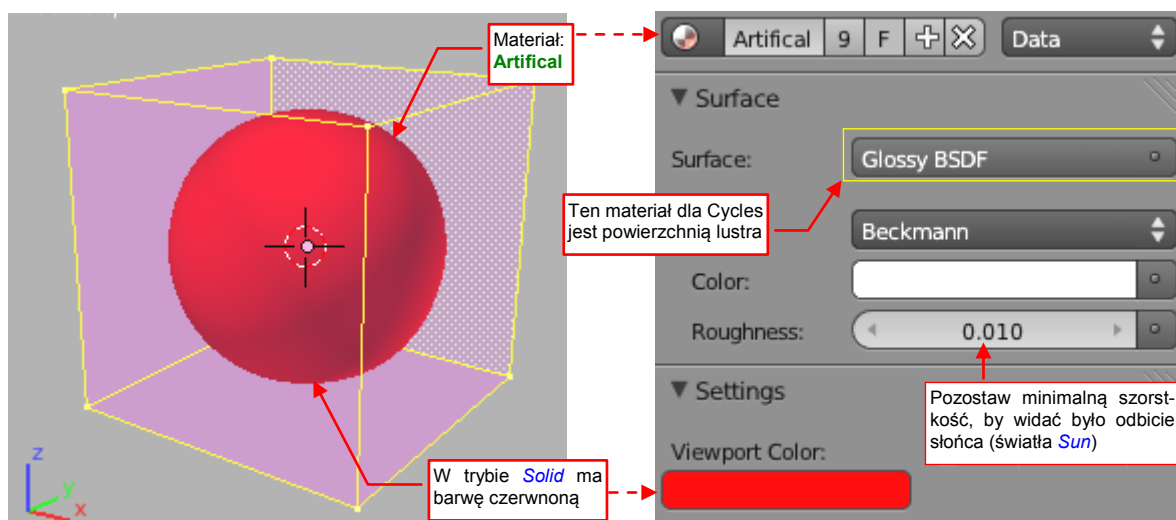
Renderowaną scenę obserwujemy przez „okienko” kamery. Dla typowych ustawień (ogniskowa rzędu 70-90 mm) oznacza to że widać przez nie około 0.3% ogólnej powierzchni obrazu, który umieściliśmy w charakterze panoramy otoczenia (np. *Environment Texture* — por. str. 336). Obserwacja tak ograniczonego obszaru bardzo utrudnia modyfikację jego ustawień, np. geometrii (zmianę skali i przesunięcie w celu obniżenia linii horyzontu, lub podobne). Czasami po prostu trudno się zorientować, co właściwie aktualnie widać w tle okna.

W takich sytuacjach warto użyć w podglądzie renderu bardzo prostego narzędzia: sondy w postaci lustrzanej kuli. Nie musi mieć idealnie kulistego kształtu, więc łatwo ją stworzyć dosłownie w parę sekund. Wystarczy umieścić w środku sceny (zazwyczaj jest to środek globalnego układu współrzędnych — punkt 0,0,0) sześcian. Następnie wygładzić go porządnie modyfikatorem *Subdivision Surface* — ustawiając parametr *Subdivisions* na 4 lub 5 — aby uzyskać kształt zbliżony do sfery (Rysunek 9.5.1):



Rysunek 9.5.1 Siatka sondy

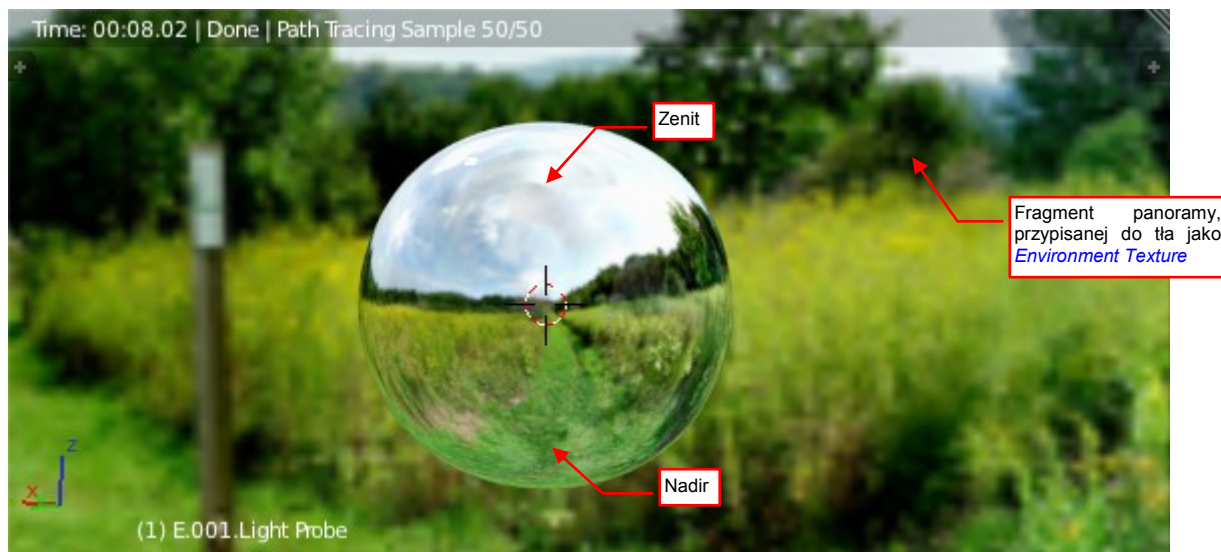
Nadałem temu pomocniczemu obiektowi nazwę **E.001.Light Probe**. Aby nie przeszkadzał w głównej scenie, dodałem go w pomocniczej scenie **Spares** (por. str. 203, 462). (Przypisz jej to samo otoczenie: **World**). Powierzchnia sondy jest pokryta materiałem **Artificial**. W Cycles nadałem mu właściwości lustra, bo nie chciałem mnożyć niepotrzebnie materiałów pomocniczych (Rysunek 9.5.2):



Rysunek 9.5.2 Materiał sondy — powierzchnia lustrzana (**Artificial**)



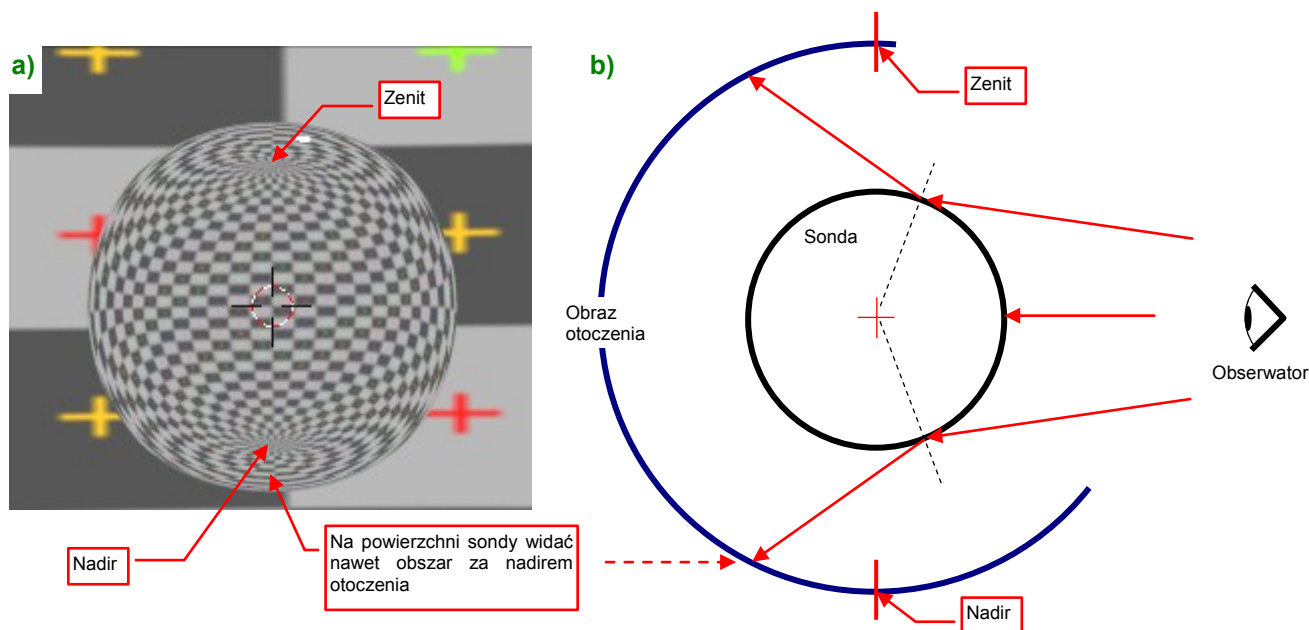
Włącz teraz w oknie **3D View** podgląd renderu, a przekonasz się, jak działa nasza sonda (Rysunek 9.5.3):



**Rysunek 9.5.3 Użycie sondy do orientacji w obrazie otoczenia**

Odbija się w niej otoczenie. Obracając się wokół tego obiektu, możesz w niej obejrzeć cały obraz panoramy — tak, jak jest rozłożony wokół sceny.

Zwróć uwagę, że na powierzchni kuli widać zarówno zenit jak i nadir sfery otoczenia. Zobaczysz to wyraźniej, gdy jako panoramę otoczenia podstawisz obraz testowej „szachownicy” (Rysunek 9.5.4a):

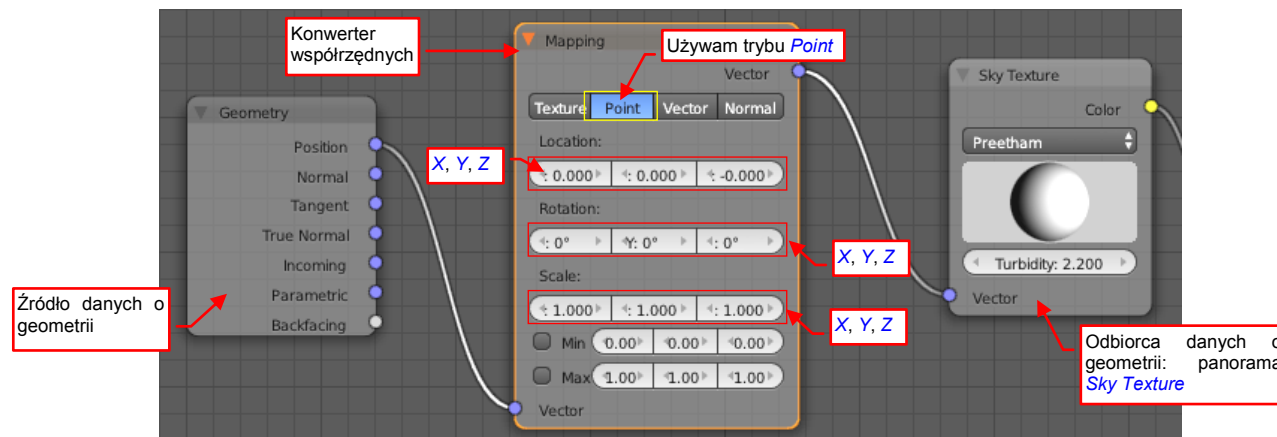


**Rysunek 9.5.4 Szczegóły geometrii odbijanego w sondzie obrazu**

Ten efekt to naturalne zniekształcenie obrazu w kuli. Obraz otoczenia jest w Cycles rozpostarty na powierzchni olbrzymiej sfery, w środku której znajduje się scena. W jej biegunach — zenicie i nadirze — wyświetlana panorama ulega największej deformacji. Rysunek 9.5.4b pokazuje drogę promieni światła, w wyniku której na powierzchni kuli widać obszary położone po przeciwnej stronie biegunów otoczenia.

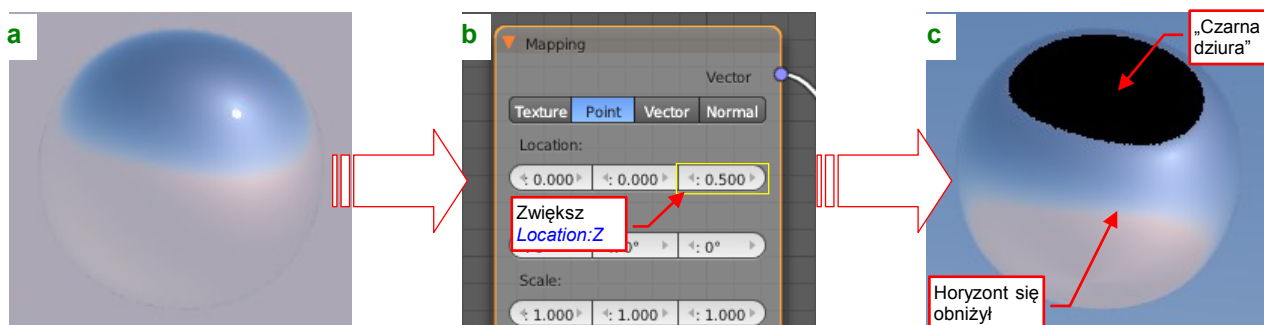
## 9.6 Transformacja współrzędnych (**Mapping**)

Węzeł **Mapping** dodaje się w **Node Editor** poleceniem **Add→Vector→Mapping**. Służy do modyfikacji danych typu **Vector**. Dane tego typu są zazwyczaj używane do ustalenia sposobu wyświetlania rozkładanych na powierzchniach obiektów obrazów (tekstur), oraz panoram otoczenia. Ten węzeł to typowy „konwerter”, wstawiany pomiędzy źródło i odbiorcę danych. Rysunek 9.6.1 przedstawia przykład użycia:



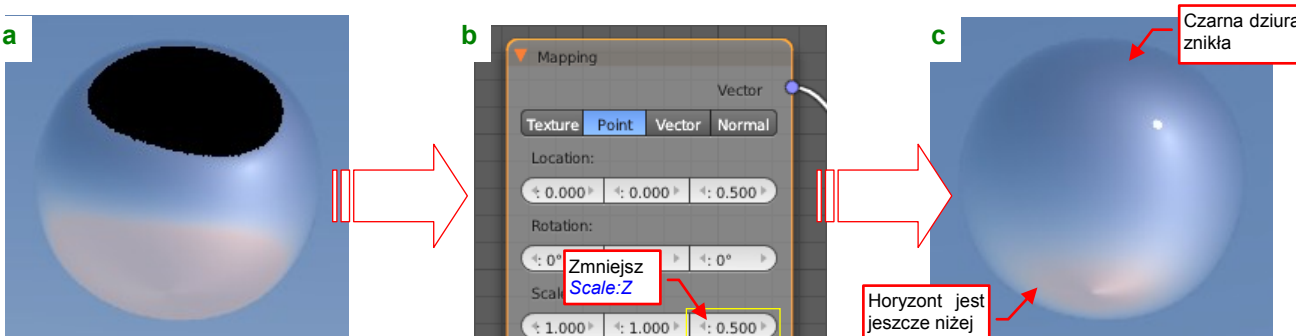
Rysunek 9.6.1 Typowe połączenie węzła **Mapping**

Źródłem danych geometrycznych jest tu węzeł **Geometry**, odbiorcą — panorama **Sky Texture**. Kontrolki węzła **Mapping** służą do transformacji współrzędnych dostarczanych na wejściu. Ustawienia węzła przedstawionego przez Rysunek 9.6.1 nie zmieniają jeszcze niczego — przesunięcie (**Location**) i obrót (**Rotation**) są zerowe. W tym stanie linia horyzontu **Sky Texture** jest w okolicy równika — tak jak pokazuje sonda (Rysunek 9.6.2a):



Rysunek 9.6.2 Zastosowanie węzła **Mapping** do obniżenia linii horyzontu panoramy otoczenia

Aby przesunąć horyzont w dół, zwiększ **Location:Z** do 0.5 (Rysunek 9.6.2b). Horyzont się przesunął, ale pojawił się niespodziewany efekt uboczny: czarna dziura w zenicie (Rysunek 9.6.2c)! (Obraz panoramy jest „za krótki” i po przesunięciu już nie sięga do końca sfery). Aby ją zlikwidować, zmniejsz **Scale:Z** z 1.0 do 0.5 (Rysunek 9.6.3b). Panorama „rozciągnie” się w obydwu kierunkach, z jednej strony docierając z powrotem do zenitu, a z drugiej — jeszcze bardziej obniżając linię horyzontu (Rysunek 9.6.3c):

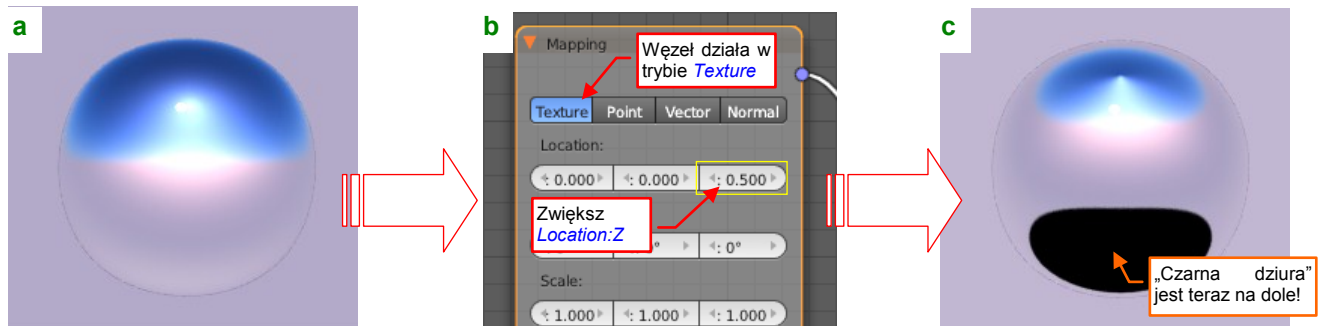


Rysunek 9.6.3 Korygowanie efektu „czarnej dziury”

Węzeł **Mapping** ma wiele trybów działania: **Texture**, **Point**, **Vector**, **Normal**. To rozróżnienie wprowadzono stosunkowo niedawno, w Blenderze 2.69. Wcześniej węzeł działał tak, jakby był zawsze ustawiony w trybie **Point** (czyli traktował podane współrzędne jakby opisywały położenie punktu w przestrzeni).

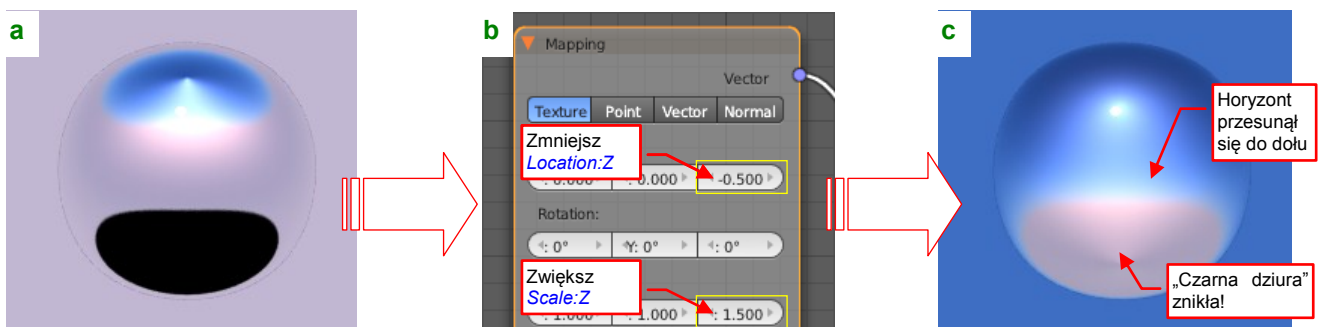
- Wszystkie przykłady w tej książce powstawały we wcześniejszych wersjach Blendera 2.6. Dlatego węzły **Mapping**, które wykorzystuję w schematach materiałów, działają w trybie **Point**.

Tryb **Texture** jest odwrotnością trybu **Point** (w zamyśle twórców ma służyć do transformacji obrazu tekstury, a nie jej współrzędnych). Jeżeli np. zastosujesz go do transformacji którą pokazuje Rysunek 9.6.2a) (por. Rysunek 9.6.4b) uzyskasz „czarną dziurę” u dołu, a nie u góry panoramy (Rysunek 9.6.4c):



Rysunek 9.6.4 Efektu „czarnej dziury” na panoramie (tryb **Texture**)

Aby uzyskać ją u góry, musisz przesunąć teksturę do dołu (wprowadzając ujemną **Location:Z** = -0.5), oraz zwiększyć, a nie zmniejszyć jej skalę (**Scale:Z** = 1.5). Ilustruje to Rysunek 9.6.5:



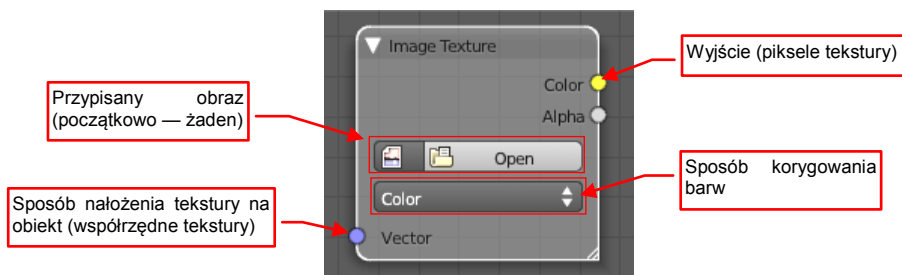
Rysunek 9.6.5 Korygowanie efektu „czarnej dziury” (tryb **Texture**)

Tryb **Vector** ma służyć do transformacji wektorów — stąd w tym trybie ignorowane są wszelkie zmiany położenia (**Location**). (Wektory można tylko obracać i skalować).

Tryb **Normal** ma służyć do transformacji wektorów o jednostkowej długości, stąd w tym trybie ignorowane są wszelkie zmiany położenia (**Location**) i skali (**Scale**). (Wektory jednostkowe można tylko obracać)

## 9.7 Dodanie obrazu tekstury (*Image Texture*)

Aby dodać obraz rastrowy jako nową teksturę<sup>1</sup>, wywołaj w *Node Editor* polecenie **Add→Texture→Image Texture**. Spowoduje to dodanie do schematu nowego węzła (Rysunek 9.7.1):



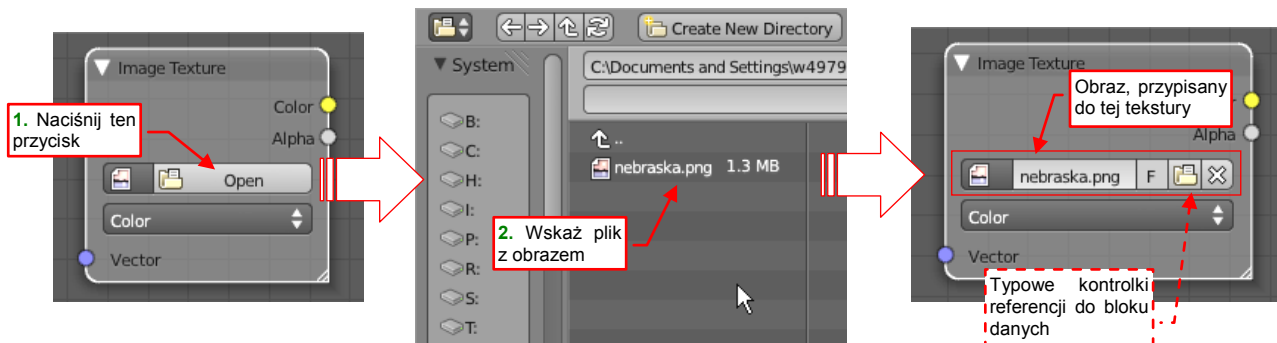
Rysunek 9.7.1 Węzeł *Image Texture*

Na wejściu należy zawsze podłączyć jakieś współrzędne tekstury (*Vector*). (Zazwyczaj wykorzystuje się w tym celu jedno z wyjść węzła *Texture Coordinate*). Jeżeli nic tu nie podłączysz, Cycles wykorzystuje niejawnie współrzędne UV siatki. Na wyjściu węzeł zwraca piksele tekstury (*Color*). Można je podłączyć np. do wejścia *Color* jakiegoś shadera, nakładając w ten sposób obraz tekstury na jego powierzchnię.



Poniżej przypisania do obrazu węzeł *Image Texture* zawiera pomocniczą kontrolkę, za pomocą której wybiera się typ korekty barw pikseli. Na razie dostępne są dwie możliwości:

- **Non-Color Data**: bez korekty (używane są odczytane z pliku obrazu wartości **RGBA**);
- **Color**: odczytane z pliku wartości **RGBA** pikseli podlegają tzw. korekcie gamma (por. str. 205), a dopiero potem są użyte w renderze;

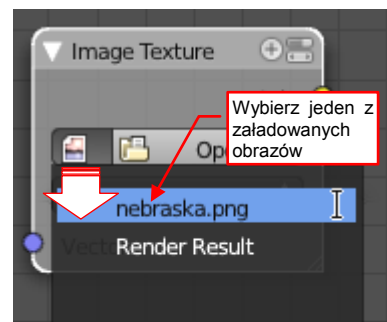
Początkowo nowy węzeł nie jest przypisany do żadnego obrazu. Nowy plik obrazu można odczytać z dysku używając przycisku **Open** (Rysunek 9.7.2):



Rysunek 9.7.2 Przypisanie teksturze nowego obrazu (z pliku)

Ładowany obraz to oddzielny blok danych (wyjaśnienie tego pojęcia — por. str. 462). Tekstura zawiera tylko jego przypisanie (referencję). Dlatego po załadowaniu obrazu na miejscu przycisku **Open** pojawił się typowy zestaw kontrolki do obsługi referencji. Przyciskiem  można załadować z dysku inny obraz na miejsce aktualnego, a przycisk  służy do usunięcia powiązania węzła z obrazem.

Po załadowaniu ten sam obraz można przypisać do wielu różnych węzłów *Image Texture*. Służy do tego lista rozwijalna, umieszczona z lewej strony przycisku **Open** (Rysunek 9.7.3).



Rysunek 9.7.3 Przypisanie teksturze istniejącego obrazu

<sup>1</sup> Słowo "tekstura" w grafice komputerowej nie całkiem odpowiada popularnemu znaczeniu — patrz słownik na str. 1303



Być może kiedyś do węzła *Image Texture* zostanie dodane jakieś okno podglądu obrazu. Póki co, każdy załadowany (w ten czy inny sposób) do Blendera obraz można obejrzeć w oknie *Image Editor* (Rysunek 9.7.4):



Rysunek 9.7.4 Podgląd pliku obrazu (*Image Editor*).

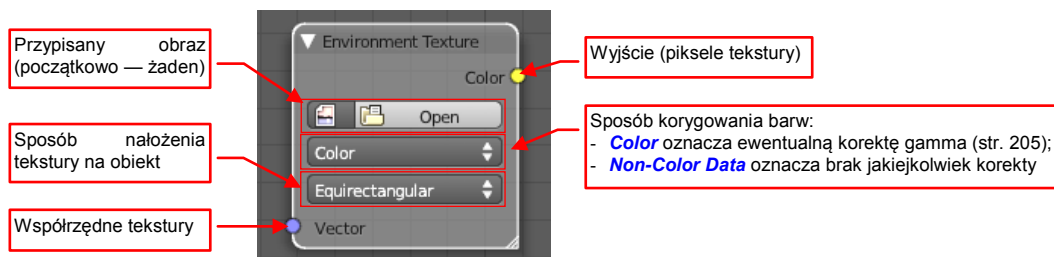
Domyślnie Blender nie kopiuje do pliku ze sceną zawartości obrazu<sup>1</sup>. Zapamiętuje tylko ścieżkę, skąd ma pobrać plik. Wynika stąd praktyczna wskazówka:

- Możesz zmieniać zawartość obrazu tekstury za pomocą zewnętrznego narzędzia (np. GIMP), bez konieczności zamykania Blendera. Po prostu otwórz, zmodyfikuj i zapisz odpowiedni plik obrazu na dysku. Potem wystarczy w *Image Editor* wywołać polecenie *Image→Reload Image* (**Alt-R**)— i zobaczysz w Blenderze jego aktualną wersję.

<sup>1</sup> Możesz go do tego zmusić: aktualny obraz w *Image Editor* można dołączyć do pliku Blendera poleceniem *Image→Pack Image*. Wszystkie obrazy naraz można dodać poleceniem *File→External data→Pack into .blend file*.

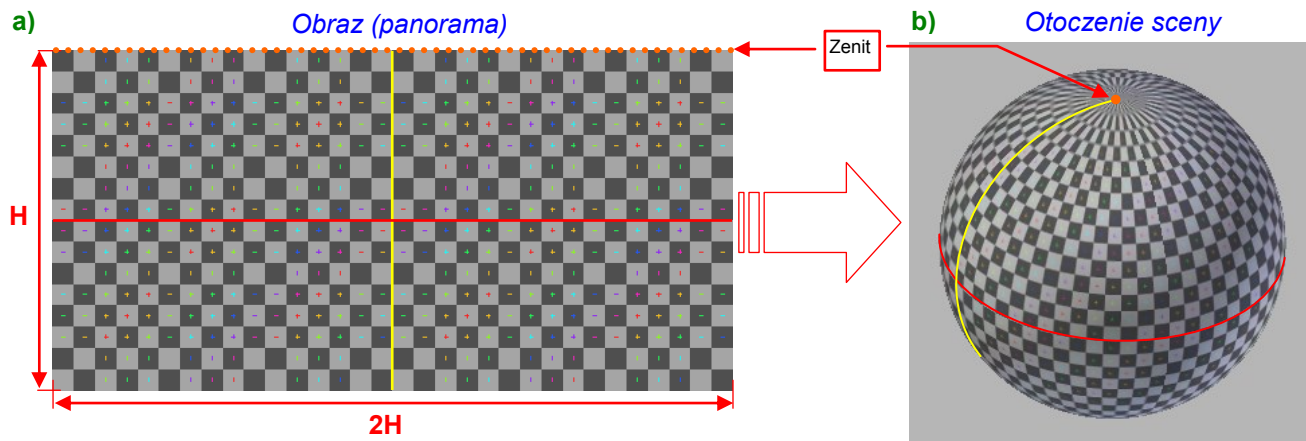
## 9.8 Dodanie panoramy otoczenia (*Environment Texture*)

Ten węzeł służy do otoczenia sceny jakimś panoramicznym obrazem, Dodaje się go w *Node Editor* poleceniem **Add→Texture→Environment Texture** (Rysunek 9.8.1):



Rysunek 9.8.1 Węzeł *Environment Texture*

Węzeł *Environment Texture* ma podobne kontrolki oraz takie same punkty wejścia i wyjścia jak węzeł *Image Texture* (por. str. 334, Rysunek 9.7.1). To dlatego, że używa się ich w podobny sposób. Czym się różnią? Kształtem obrazu! Węzeł *Image Texture* reprezentuje płaski kwadrat. A *Environment Texture* to panorama — a dokładniej, sfera. W domyślnym trybie mapowania (*Equirectangular*) Cycles oczekuje przypisania do tego węzła obrazu o proporcjach wysokości i szerokości jak 1:2 (Rysunek 9.8.2a):



Rysunek 9.8.2 Odzworowanie obrazu na otoczeniu sceny w mapowaniu *Equirectangular*

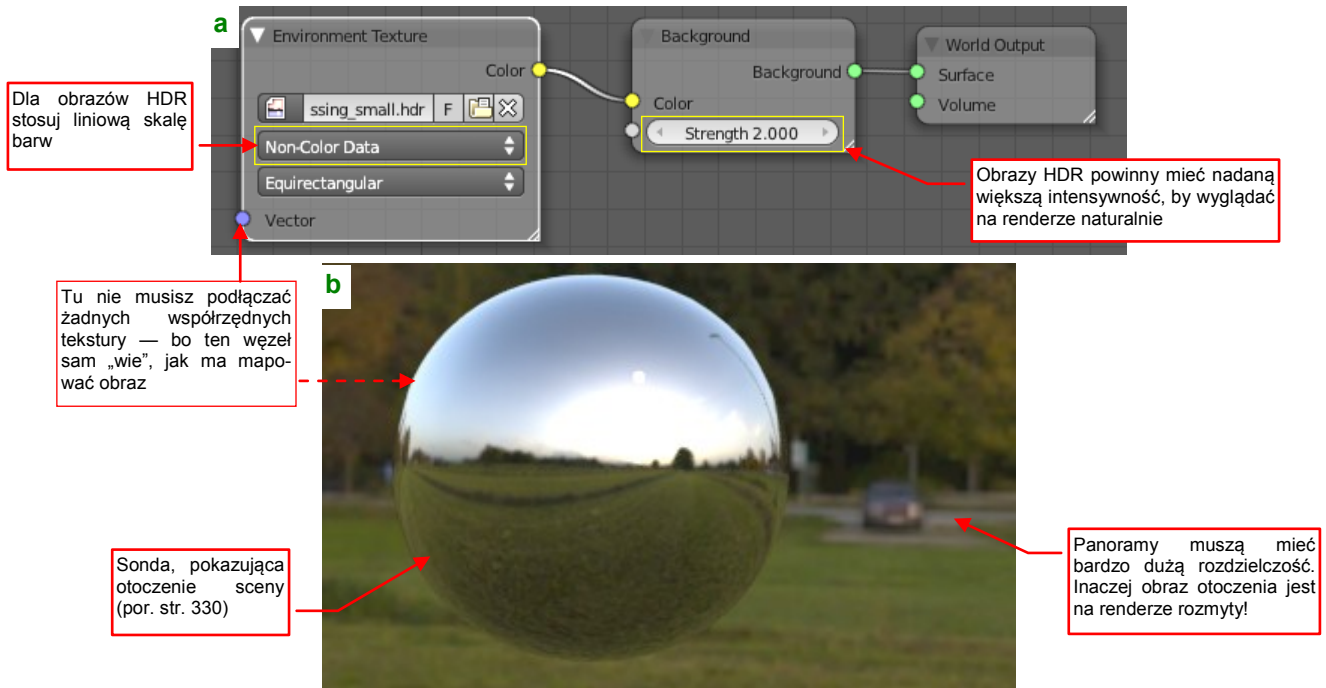
Efekt uzyskany z *Environment Texture* jest taki sam, jak gdybyś umieścił scenę w ogromnej sferze, na którą jest nałożony obraz (w znanej z geografii projekcji Merkatora: Rysunek 9.8.2b). Skąd wziąć takie panoramy? Spróbuj poszukać ich w Internecie. Wiele z nich jest zapisana w specjalnym formacie HDR (Rysunek 9.8.3):



Rysunek 9.8.3 Przykładowa panorama (obraz HDR z [www.openfootage.net](http://www.openfootage.net))

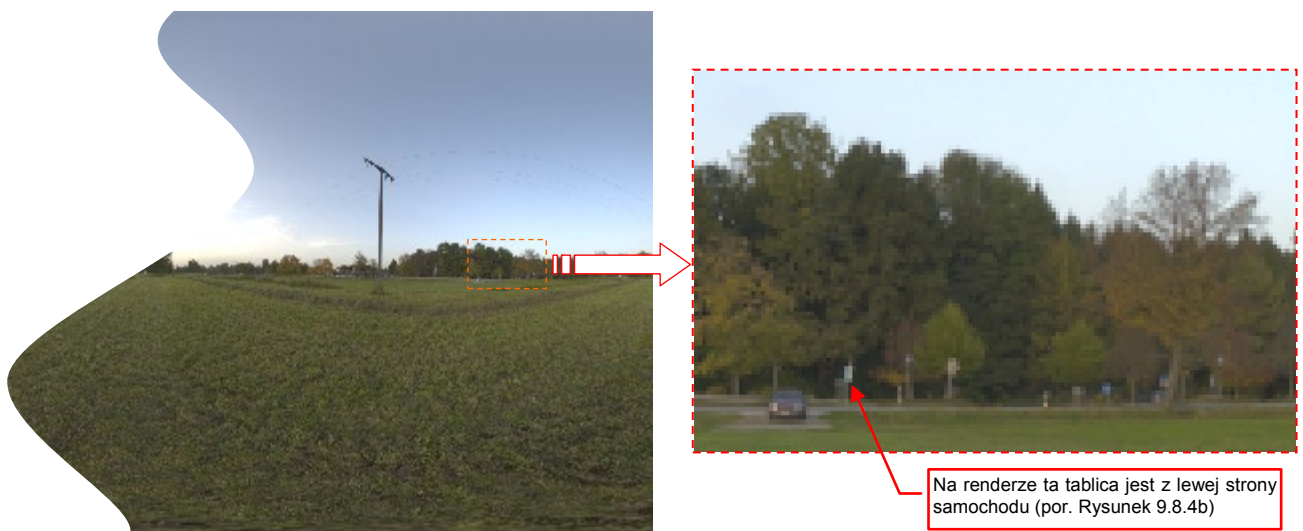
Pliki *\*.hdr* możesz podejrzeć tylko w Blenderze, bo kolory są w nich zapisane jako liczby zmiennoprzecinkowe.

(GIMP nie obsługuje jeszcze tego formatu). Gdy wyświetlisz taki obraz w *UV/Image Editor* Blendera, zapewne zauważysz że jest przyciemniony (por. Rysunek 9.8.3). To efekt uboczny innej skali intensywności, stosowanej w takich plikach. Najprostszy przykład użycia węzła *Environment Texture* przedstawia Rysunek 9.8.4a):



Rysunek 9.8.4 Przykład zastosowania węzła *Environment Texture*

Podłączasz jego wyjście (*Environment Texture:Color*) do wejścia *Color* shadera *Background*. Nie musisz niczego podłączać do wejścia *Environment Texture:Vector* (choć możesz — o tym za chwilę). Jeżeli obraz panoramy jest w jednym z typowych formatów obrazów rastrowych — *\*.jpg*, *\*.png*, *\*.tiff* — pozostaw domyślny tryb *Color* (z konwersją gamma, por. str. 334). Dla obrazów HDR użyj trybu *Non-Color Data* (to tryb bez żadnej korekty — por. Rysunek 9.8.4a). Obrazy HDR mogą także wymagać ustawienia większej intensywności (*Background:Strength*). Rezultat przedstawia Rysunek 9.8.4b). To sonda, wstawiona do tak ustawionego otoczenia. Zwróć uwagę, jak bardzo rozmyty jest obraz tła na tym próbnym renderze. To efekt zbyt małej rozdzielczości obrazu (choć ta panorama ma rozmiar 1000x2000px). Po prostu kamera „widzi” tylko jej mały wycinek (Rysunek 9.8.5):



Rysunek 9.8.5 Położenie fragmentu obrazu pokazywanego w kamerze (obraz jest rozjaśniony)



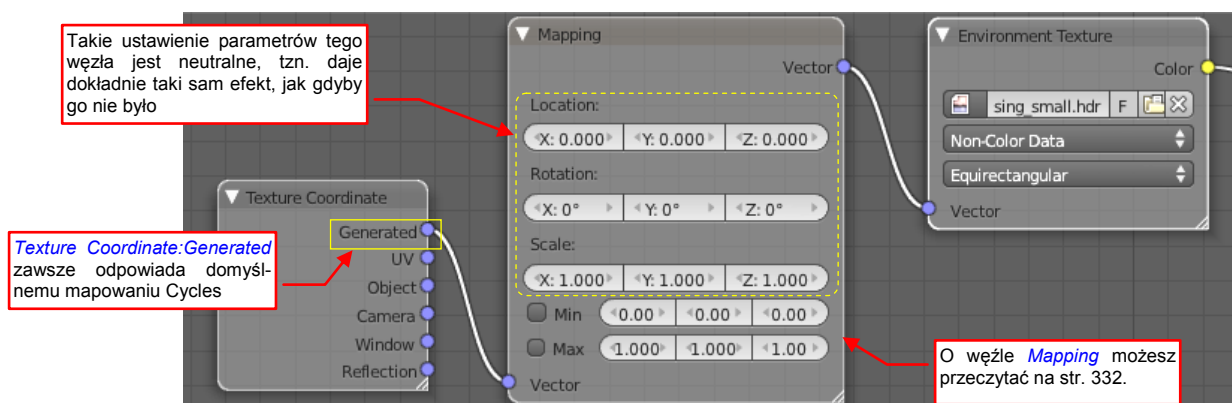
Ten rozmyty obraz wyklucza użycie panoram o rozdzielczości rzędu 2000 – 3000 px jako podstawowego tła sceny. Widać to na naszym testowym modelu — brak ostrości psuje efekt<sup>1</sup> (Rysunek 9.8.6):



Rysunek 9.8.6 Model w otoczeniu panoramy

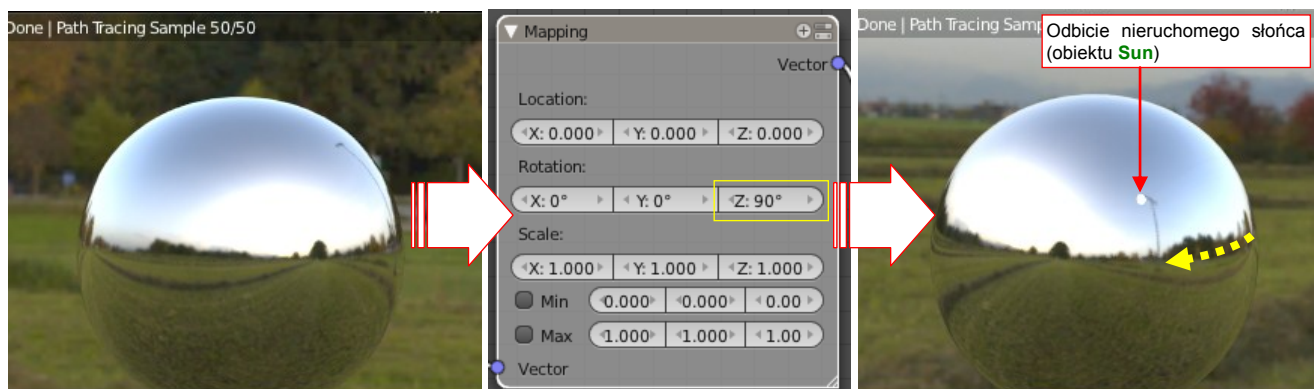
Zwróć jednak uwagę, że odbicia otoczenia na powierzchni modelu są już przy tym rozmiarze panoramy całkiem zadowalające. Można to wykorzystać przy budowaniu złożonych schematów środowiska wokół sceny (str. 350).

Sferę otoczenia można poruszać. Wystarczy dołączyć do wejścia *Environment Texture:Vector* dwa dodatkowe węzły: *Texture Coordinate* i *Mapping* (Rysunek 9.8.7):



Rysunek 9.8.7 Dodatkowe węzły, umożliwiające sterowanie sferą otoczenia

Gdy zaczniesz zmieniać wartość *Mapping:Rotation:Z*, zobaczysz że otoczenie się obraca (Rysunek 9.8.8):

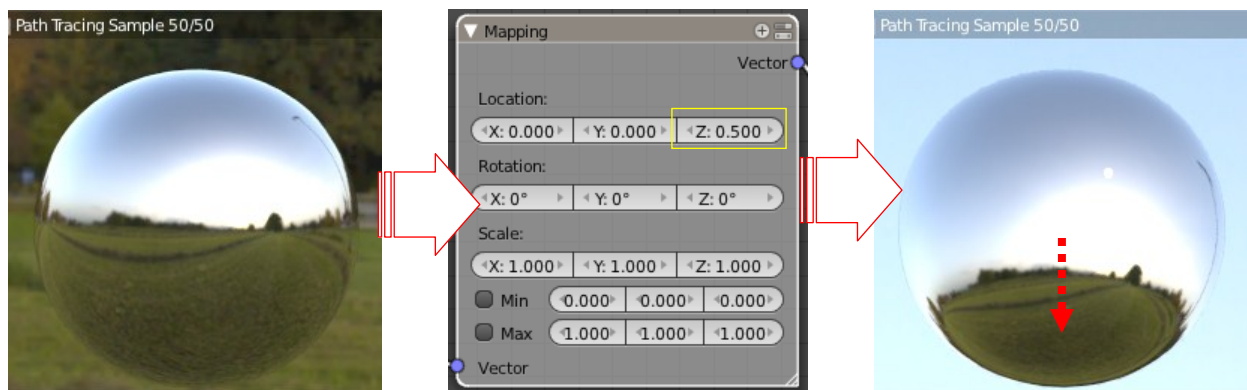


Rysunek 9.8.8 Obracanie otoczenia wokół sceny

<sup>1</sup> Chyba, że wykorzystasz ten obraz jako bazę do jakiegoś kierunkowego rozmycia — w ten sposób można stworzyć wrażenie szybkiego lotu tuż przy ziemi...



Podobnie możesz obniżyć linię horyzontu, przesuwając otoczenie w dół. Wystarczy zwiększyć wartość parametru **Mapping:Location:Z** (Rysunek 9.8.9):



Rysunek 9.8.9 Przesuwanie linii horyzontu w dół sceny

Zwróć uwagę, że mimo znacznego przesunięcia do dołu, w zenicie nie pojawiły się tu żadne artefakty, jak to miało miejsce dla **Sky Texture** (por. str. 332, Rysunek 9.6.2).

Na koniec warto wspomnieć, że w wielu filmach instruktażowych (*video tutorials*) o Blenderze autorzy wykorzystują siatkę do zbudowania sfery wokół sceny. Następnie nakładają na nią obraz otoczenia. Podczas pracy nad tą sekcją porównałem uzyskiwany w ten sposób efekt z rezultatem uzyskiwanym za pomocą węzła **Environment Texture**. I wiecie co? Nie było żadnej specjalnej różnicy!

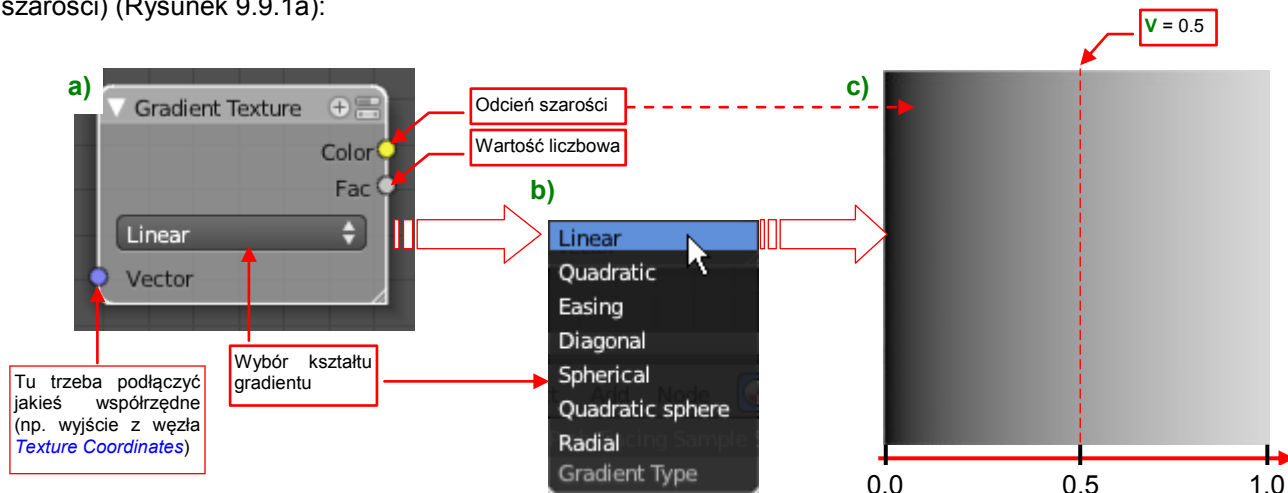
- Wykorzystując otoczenie generowane przez węzeł **Environment Texture** można uzyskać ten sam efekt co za pomocą zamodelowanej siatki sfery, otaczającej scenę.

Wykorzystanie tego węzła jest rozwiązaniem prostszym, bardziej czytelnym i eleganckim niż dodawania do sceny ogromnej sfery. Pamiętaj o tym, gdy zobaczysz taką siatkę w kolejnym tutorialu o tworzeniu sceny w Blenderze!

## 9.9 Użycie tekstury gradientu (*Gradient Texture*)

Tekstury proceduralne są oparte na jakimś wyrażeniu matematycznym, lub algorytmie, pozwalającym uzyskać dwuwymiarowy obraz. Są to zazwyczaj jakieś abstrakcyjne (co nie znaczy, że bezużyteczne!) wzory, nieregularne „chmury”. Zdarzają się jednak także wzory regularne — np. procedura generująca rzędy cegieł w ścianie. Tekstury proceduralne stosuje się zazwyczaj razem, a czasami nawet zamiast tekstur rastrowych.

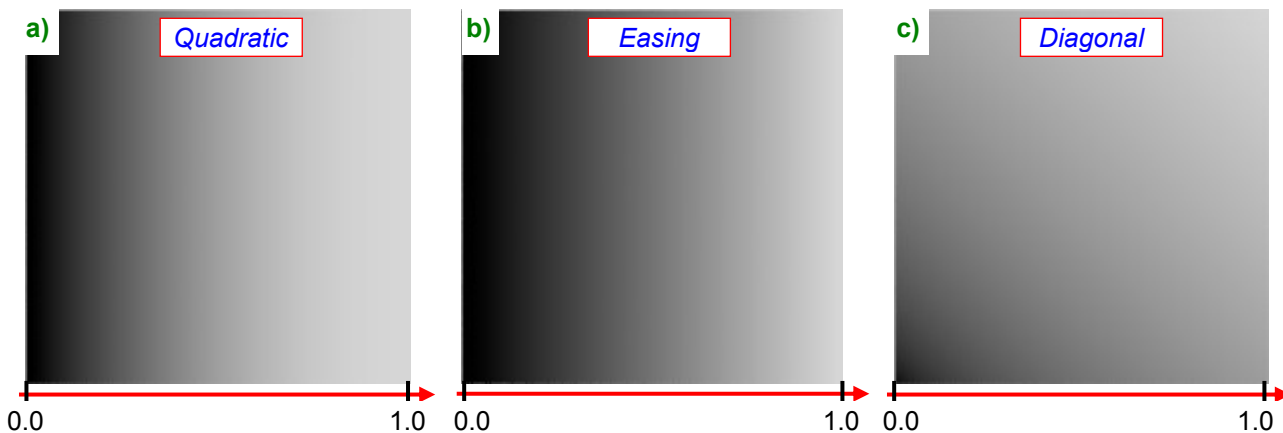
Jedną z tekstur proceduralnych dostępnych w Cycles jest prosty, czarno-biały gradient. Jego węzeł dodaje się do schematu poleceniem **Add → Texture → Gradient Texture**. Węzeł *Gradient Texture* przydaje się często jako „źródło współczynników”, dlatego ma dwa wyjścia: **Color** (barwa) i **Fac** (odpowiadająca jej wartość odcienia szarości) (Rysunek 9.9.1a):



Rysunek 9.9.1 Węzeł *Gradient Texture* i obraz, który generuje

Podobnie jak inne węzły tekstur (por. str. 327, 334, 336) gradient ma jedno wejście typu **Vector**. Służy do transformacji wyświetlanego obrazu (pokażę to na następnych stronach tej sekcji). *Gradient Texture* ma jeden parametr — listę rozwijalną, z której można wybrać jeden z kilku dostępnych kształtów gradientu (Rysunek 9.9.1b). Rysunek 9.9.1c pokazuje, jak wygląda gradient liniowy (**Linear**). To „obraz jednowymiarowy”. Gdy na wejściu **Vector** współrzędna **X** jest równa 0.0, *Gradient Texture:Linear* zwraca wartość 0 (kolor czarny). Dla współrzędnej **X** = 1.0 zwraca wartość 1 (biały). W istocie to prosta zależność liniowa:  $V = X$ .

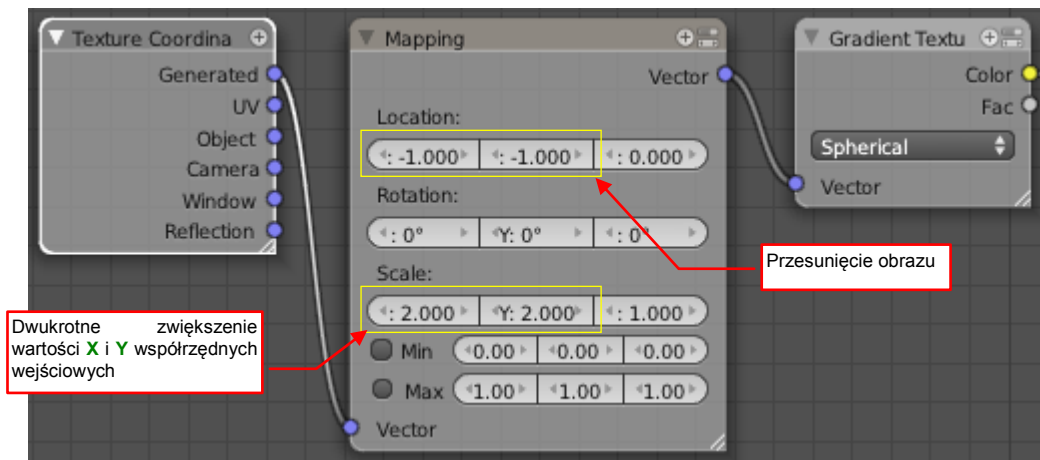
Rysunek 9.9.2a,b) przedstawia pozostałe kształty gradientów liniowych, które mamy do dyspozycji w tym węźle (**Quadratic**, **Easing**). Obydwa dla **X** = 0 zwracają 0, a dla **X** = 1 — 1. Różnią się tylko „tym, co pośrodku” — sposobem przejścia z czerni do bieli.



Rysunek 9.9.2 Pozostałe rodzaje gradientów jednowymiarowych

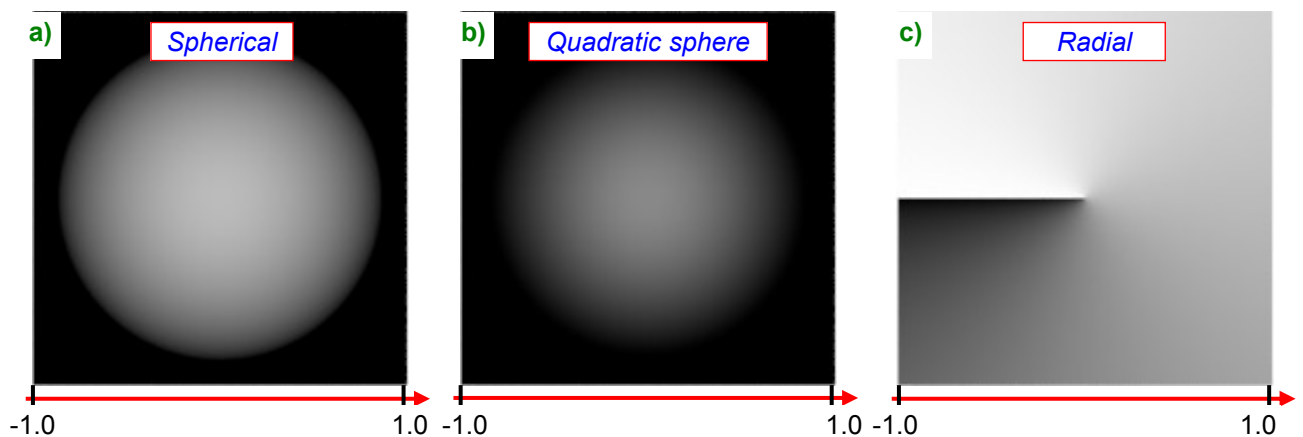
Trochę nadmiarowo jest tu także dostępny typ **Diagonal**. To właściwie gradient dwuwymiarowy. Jednak dokładnie taki sam efekt można uzyskać, pochylając pod kątem 45° (za pomocą węzła *Mapping*) gradient liniowy.

Pozostałe gradienty są oparte na okręgu o promieniu = 1.0, więc aby zobaczyć ich pełen kształt, trzeba podać teksturę transformacji (por. str. 332). Wykorzystuję tu typowe rozwiązanie, oparte o węzły *Texture Coordinate* i *Mapping*. Transformacja współrzędnych polega na dwukrotnym powiększeniu obrazu wyświetlanego dla oryginalnego zakresu współrzędnych  $X, Y = [0.0 \dots 1.0]$  i przesunięciu środka o połowę rozmiaru (-1, -1) (Rysunek 9.9.3):



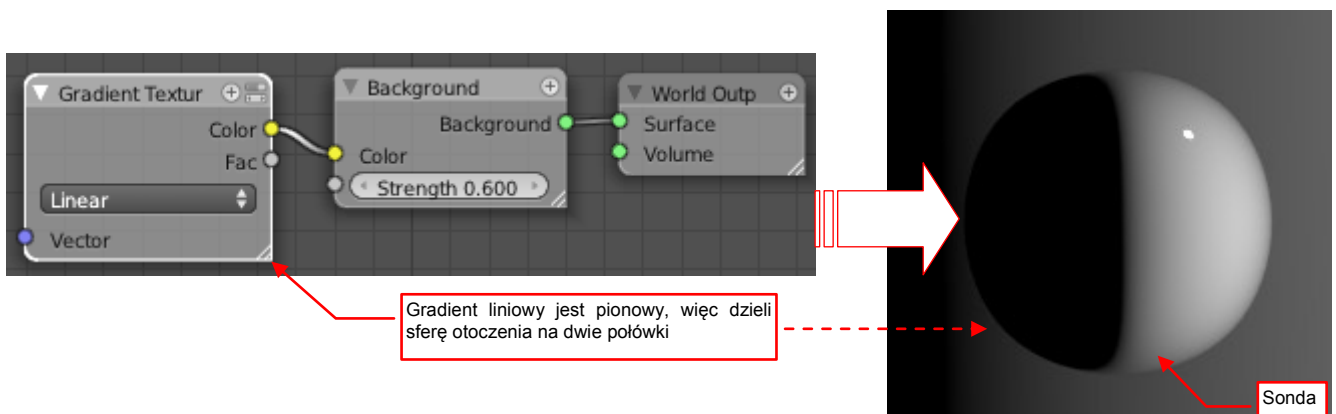
Rysunek 9.9.3 Modyfikacja położenia i rozmiaru obrazu tekstury

*Gradient Texture* dla współrzędnych wejściowych ( $X, Y$ ) z zakresu -1.0 do 1.0 wyświetla następujące kształty (Rysunek 9.9.4):



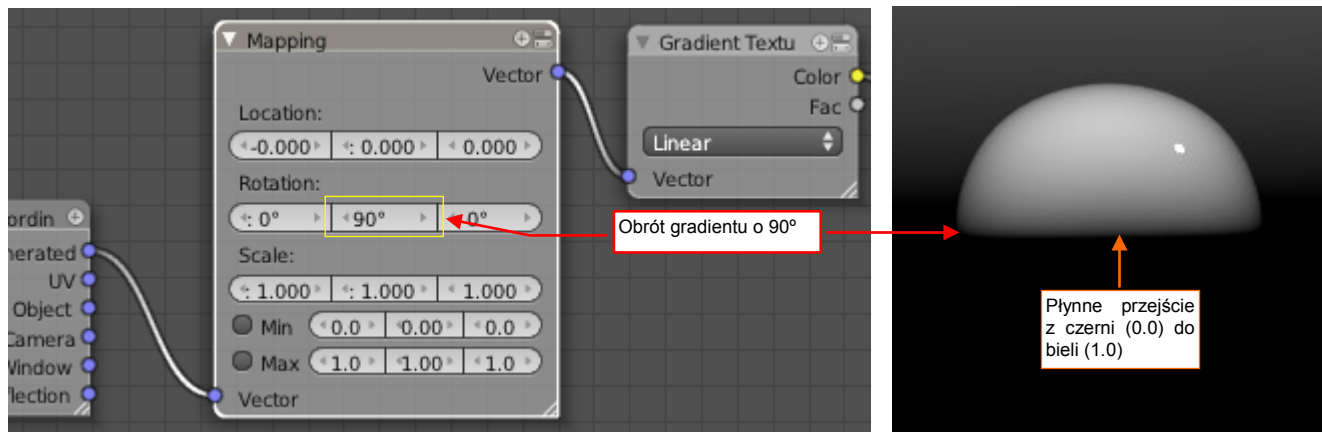
Rysunek 9.9.4 Dostępne rodzaje gradientów sferycznych

Do tej pory przedstawiałem obraz generowany przez *Gradient Texture* na płaskiej, kwadratowej płytce. Rezultat działania tego węzła można jednak także stosować dla obiektów sferycznych, takich jak otoczenie sceny (Rysunek 9.9.5). Takie wartości przydają się np. do łączenia dwóch różnych panoram:



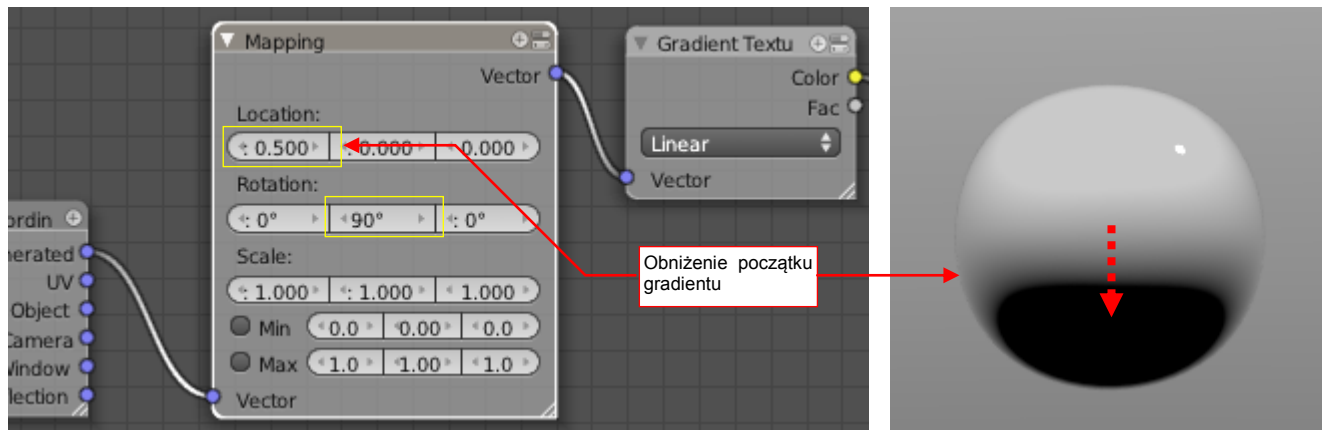
Rysunek 9.9.5 Użycie *Gradient Texture* w otoczeniu sceny

Za pomocą **Gradient Texture** można uzyskać płynne przejście pomiędzy górną półsfery z obrazem nieba i chmur, i dolną, z obrazem ziemi. Trzeba tylko obrócić kierunek gradientu (**Mapping:Rotation:Y**) o  $90^\circ$  (Rysunek 9.9.6):



Rysunek 9.9.6 Zmiana kierunku gradientu na pionowy

Potem można także sterować także wysokością tak uzyskanej „linii horyzontu” (Rysunek 9.9.7):



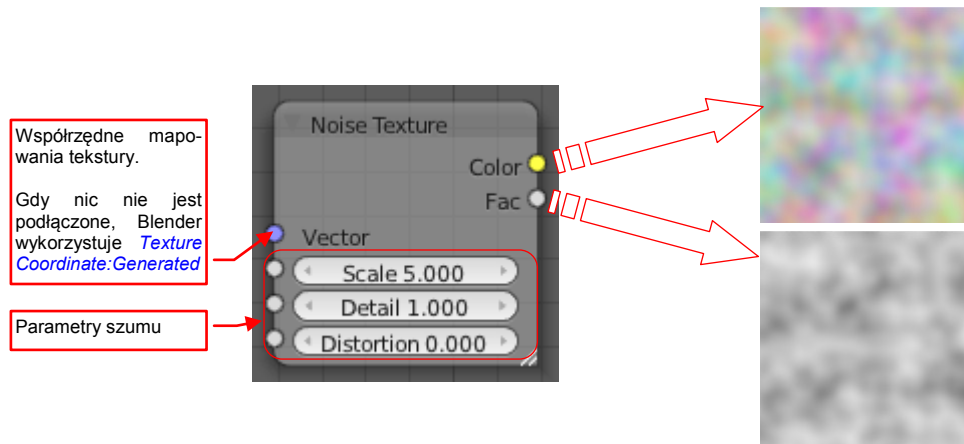
Rysunek 9.9.7 Przesunięcie granicy gradientu do dołu sfery otoczenia



### 9.10 Proceduralne tekstury „szumu” (*Noise Texture*, *Voronoi Texture*, *Musgrave Texture*)

Cycles, oprócz tekstury gradientu (omówionej w poprzedniej sekcji — por. str. 340) udostępnia węzły z kilkoma innymi teksturami proceduralnymi. Znajduje się tam m.in. kilka rodzajów nieregularnych „szumów”.

Zacznijmy od „zwykłej” tekstury szumu, czy też „chmur”. Jej węzeł dodaje się do schematu poleceniem **Add → Texture → Noise Texture**. Węzeł *Noise Texture* przydaje się wszędzie tam, gdzie chcemy wprowadzić na powierzchni jakąś nieregularność. Często jest wykorzystywany jako „źródło współczynników”, dlatego ma dwa wyjścia: **Color** (barwa) i **Fac** (intensywność, czyli odcień szarości) (Rysunek 9.10.1):

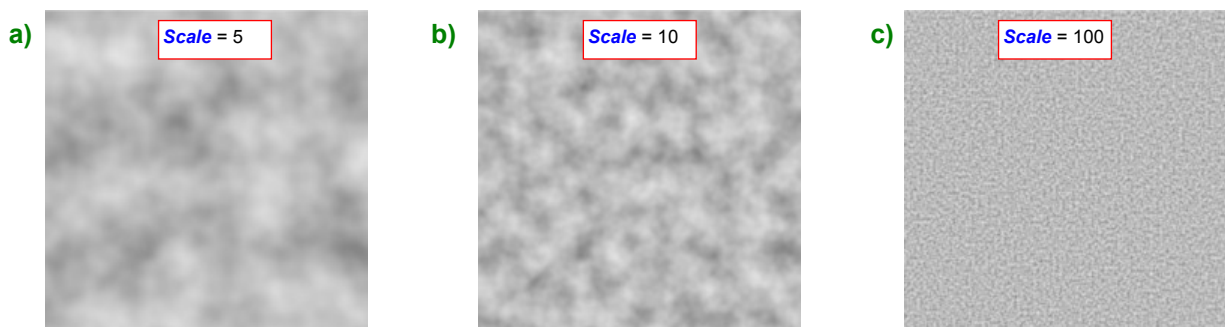


Rysunek 9.10.1 Węzeł *Noise Texture* i obraz, który generuje

*Noise Texture* ma jedno wejście (**Vector**) na współrzędne mapowania, oraz po jednym z wejść dla każdej z wartości jego parametrów. (Możesz tworzyć różne ciekawe efekty, podłączając do nich inne tekstury proceduralne). Jeżeli do wejścia **Vector** nie podłączysz żadnych współrzędnych, Cycles użyje w sposób niejawny współrzędnych *Texture Coordinate: Generated*<sup>1</sup>.

Aby dać Ci pojęcie, jak wygląda obraz generowany przez *Noise Texture*, przygotowałem po kilka przykładów ilustrujących wpływ każdego z trzech parametrów węzła (*Scale*, *Detail*, *Distortion*). Poza podaną na ilustracji wartością, pozostałe dwa parametry tekstury mają zawsze ustawione takie wartości, jakie pokazuje Rysunek 9.10.1.

Zacznijmy od parametru *Scale*. Zwiększenie skali powoduje zmniejszenie rozmiaru szumu, generowanego przez teksturę (Rysunek 9.10.2):

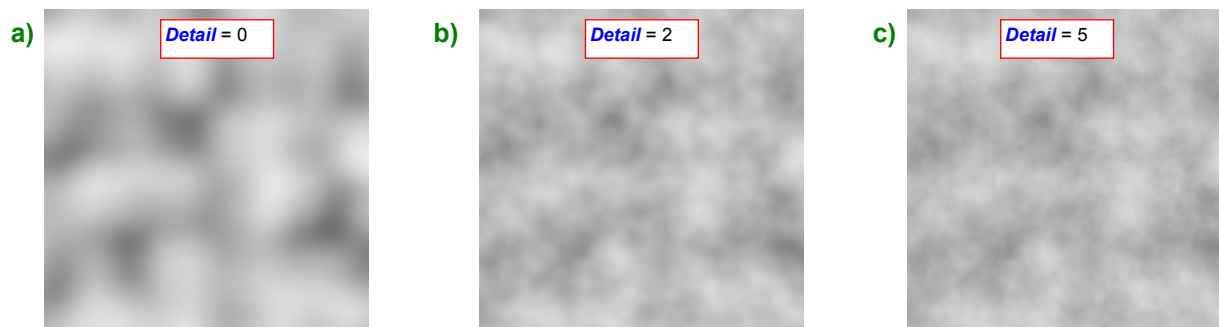


Rysunek 9.10.2 Wpływ parametru *Scale*

Zauważ, że obraz uzyskany dla *Scale* = 100 doskonale nadaje się dla jakichś chropowatych powierzchni!

<sup>1</sup> To różnica w stosunku do węzła *Image Texture*, który domyślnie wykorzystuje współrzędne UV siatki obiektu (por. str. 995). Niestety, współrzędne należy za każdym razem dla siatki zdefiniować (por. Wyciąg III). Jeżeli tego nie zrobisz — zamiast obrazu z węzła *Image Texture* zobaczysz na próbnym renderze jednolicie czarną powierzchnię. W odróżnieniu od domyślnych współrzędnych tekstur proceduralnych — *Generated* — są dostępne zawsze.

Kolejnym parametrem jest **Detail**. To poziom “szczegółowości” szumu (Rysunek 9.10.3):



Rysunek 9.10.3 Wpływ parametru **Detail**

Gdy ustawisz **Detail** = 0, uzyskasz bardzo rozmyty obraz. Wraz ze zwiększaniem wartości **Detail**, “chmury” generowane przez **Noise Image**, mają coraz więcej drobnych szczegółów. Jednocześnie wydłuża się czas renderowania powierzchni, wykorzystującej tę teksturę. (Dla porównania 100 próbek renderowało mi się: gdy **Detail** = 0 → 13s, gdy **Detail** = 1 → 14s, **Detail** = 2 → 15s, **Detail** = 5 → 18s a gdy **Detail** = 20 → 30s). Nie widziałem już specjalnej różnicy pomiędzy obrazem uzyskanym dla **Detail** = 5 i **Detail** = 20. Stąd wniosek:

- Zazwyczaj nie opłaca się używać wartości **Detail** > 5. Gdy używasz dużej skali, zmniejszaj wartość **Detail**.

Na przykład gdy **Scale** = 100 (Rysunek 9.10.2c), warto zastosować **Detail** = 0, bo i tak nie ma sensu bardziej rozdrabniać na tych mikroskopijnych „ziarenkach”.

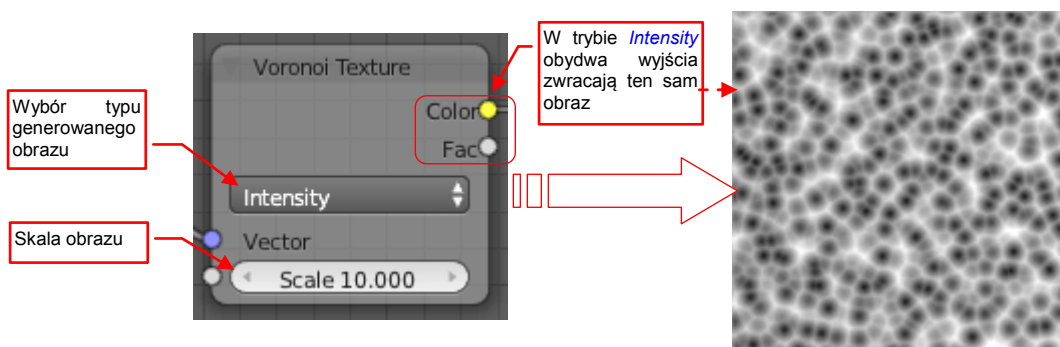
Ostatnim parametrem tekstury szumu jest **Distortion**. Ta wartość opisuje intensywność „rozmywania” obrazu szumu w różnych losowych kierunkach (Rysunek 9.10.4):



Rysunek 9.10.4 Wpływ parametru **Distortion**

W odróżnieniu od **Detail**, zmiana wartości **Distortion** nie ma znaczącego wpływu na czas liczenia każdej próbki.

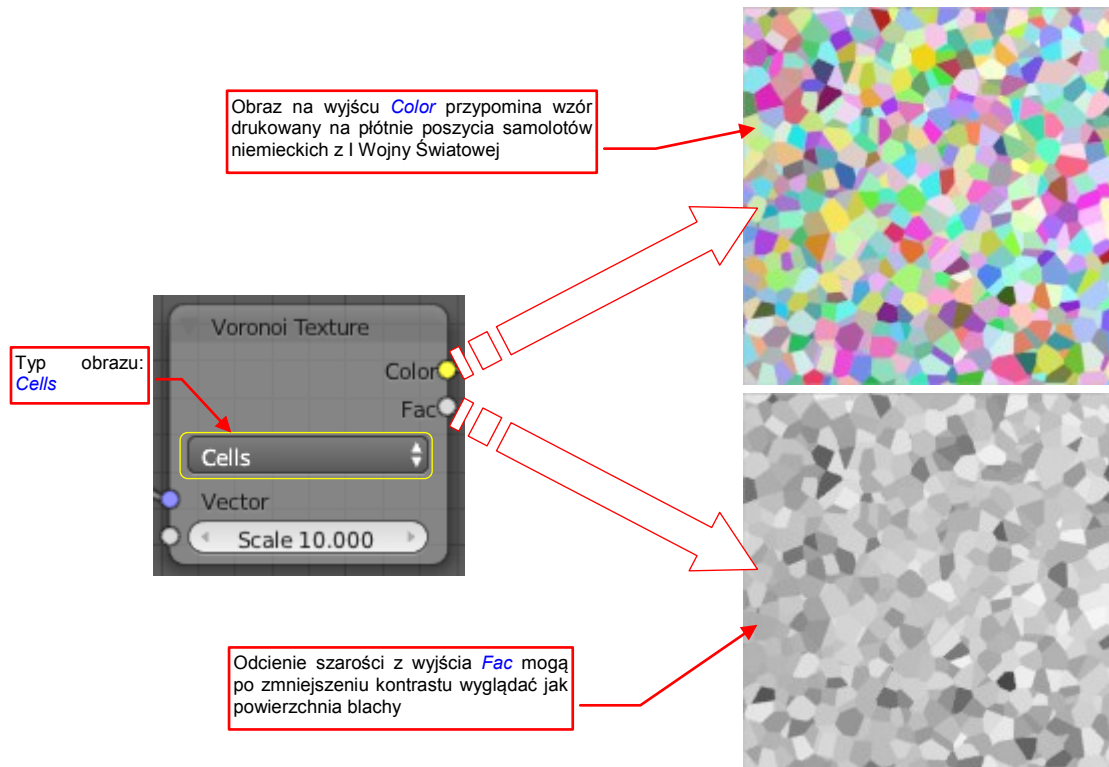
Kolejnym rodzajem węzła, który mamy do dyspozycji, jest **Voronoi Texture** (Rysunek 9.10.5):



Rysunek 9.10.5 Tekstura **Voronoi** (w trybie **Intensity**)

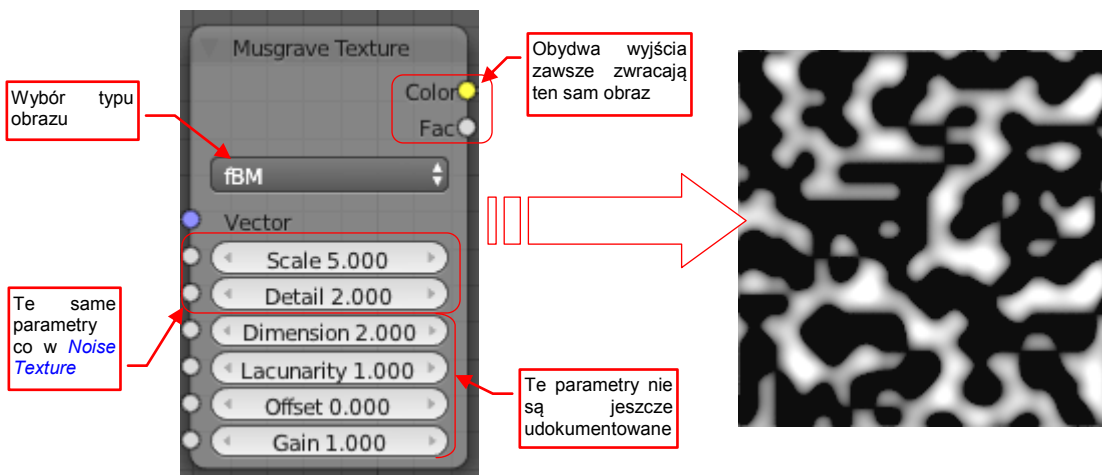
Przy odpowiednio małej skali (**Scale**) możesz użyć tej tekstury do uzyskania chropowatej powierzchni.

Po przełączenie w drugi tryb — **Cells** — **Voronoi Texture** zwraca wzór przypominający nieregularne kryształy jakiegoś minerału. Czasami taki rozkład odcieni szarości można zaobserwować na powierzchni blachy (Rysunek 9.10.6):



Rysunek 9.10.6 Tekstura **Voronoi** (w trybie **Cells**)

Trzecim rodzajem węzła, który można wykorzystać do tworzenia różnego rodzaju nieregularnych wzorów, jest **Musgrave Texture**. Ten węzeł ma najwięcej parametrów (Rysunek 9.10.7):

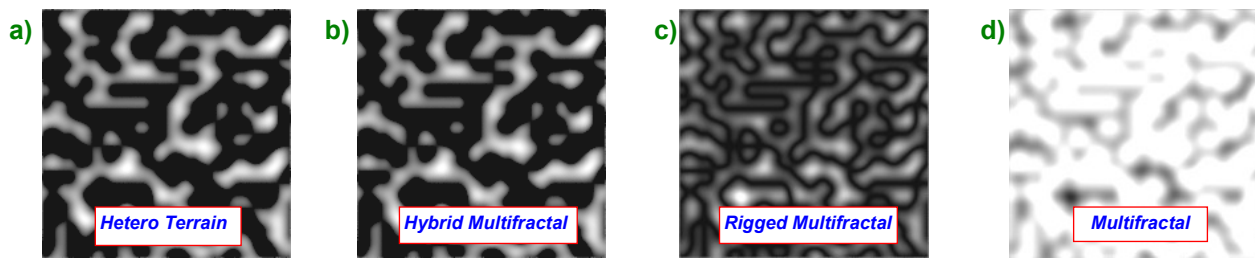


Rysunek 9.10.7 Tekstura **Musgrave**

Opis parametrów węzła **Musgrave Texture** w podstawowej dokumentacji Cycles jest bardzo lakoniczny<sup>1</sup>. Pierwsze dwa parametry — **Scale** i **Detail** — działają podobnie jak te z **Noise Texture** (por. str. 343 i 344). Nieco inna wersja tekstury **Musgrave** jest używana od kilku lat w **Blender Internal**. W porównaniu z węzłem używanym w Cycles ma jeszcze większą liczbę parametrów — m.in. wybór funkcji bazowej (**Noise Basis**). Tym niemniej gdy nie ma nic lepszego, możesz posłużyć się jej opisem jako wskazówką co do działania parametrów tego węzła (por. <http://wiki.blender.org/index.php/Doc:2.6/Manual/Textures/Types/Procedural/Musgrave>).

<sup>1</sup> Mam na myśli dokumentację Cycles z <http://wiki.blender.org>, a dokładniej opis umieszczony na tej stronie (stan na marzec 2013): <http://wiki.blender.org/index.php/Doc:2.6/Manual/Render/Cycles/Nodes/Textures>.

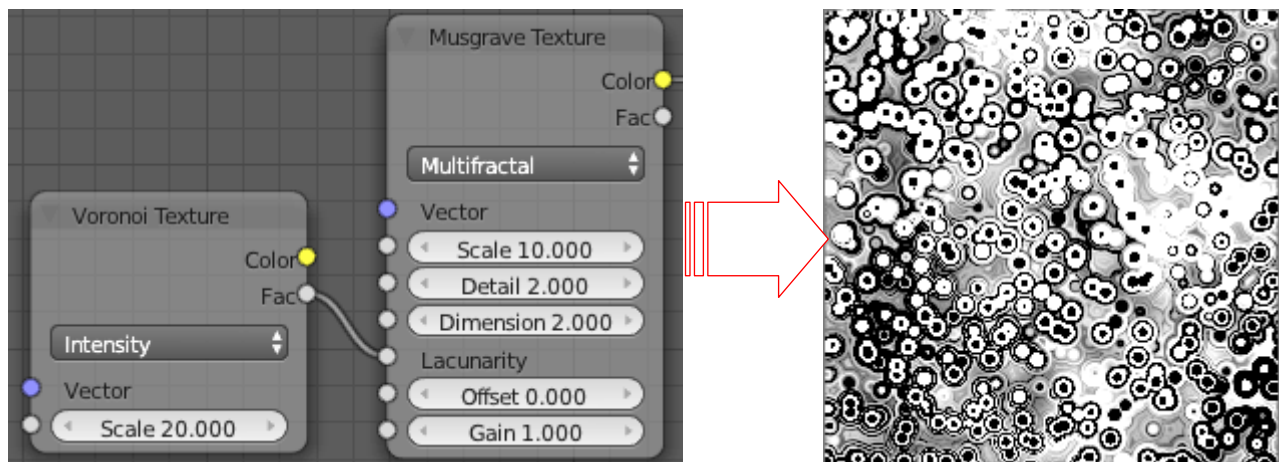
W każdym razie *Musgrave Texture* oferuje pięć różnych typów obrazów. Domyślny (*fBM*) przedstawia Rysunek 9.10.7, a pozostałe — Rysunek 9.10.8:



Rysunek 9.10.8 Inne typy tekstury *Musgrave*

W zasadzie wydaje się, że obraz typu *Hetero Terrain* (Rysunek 9.10.8a) różni się od *Hybrid Multifractal* (Rysunek 9.10.8b) tylko jasnością. Przypuszczam jednak, że dla innych wartości parametrów tej tekstury te różnice mogą być większe.

Pamiętaj, że najciekawsze efekty można osiągnąć łącząc ze sobą różne tekstury proceduralne. Na przykład — Rysunek 9.10.9 przedstawia jeden z możliwych rezultatów połączenia węzłów *Voronoi* i *Musgrave*:



Rysunek 9.10.9 Przykład efektu uzyskanego w wyniku złożenia dwóch tekstur proceduralnych

Taki wzór przypomina np. ogniska rdzy na płaskiej powierzchni blachy.



### 9.11 Podstawienie obrazu na tło renderingu

Tło, wypełnione prostymi barwami lub panoramą o niedostatecznej rozdzielczości (por. str. 337, Rysunek 9.8.4) pozostawia wiele do życzenia ze względu na brak realizmu. Rysunek 9.11.1 pokazuje jeden z takich przypadków. Czy można "podstawić" za model realistyczne zdjęcie, które pokazuje Rysunek 9.11.2?

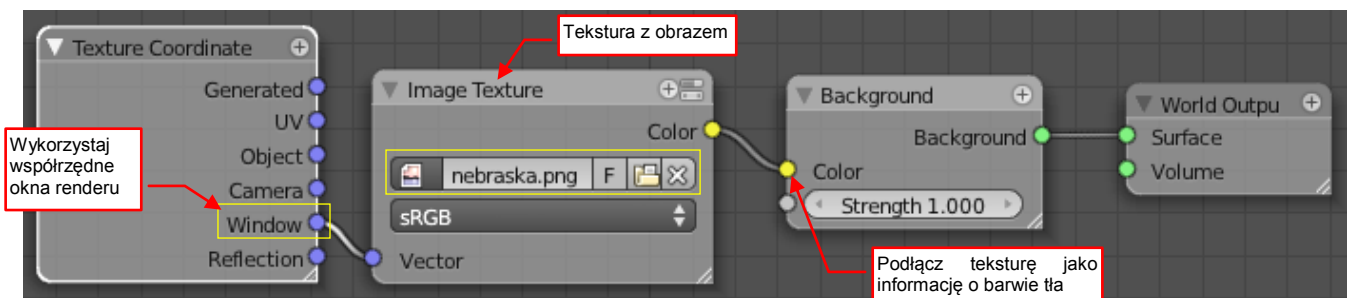


Rysunek 9.11.1 Model z kiepskim tłem



Rysunek 9.11.2 Obraz — lepsze tło dla modelu

Jest to całkiem proste. Do schematu otoczenia dodaj węzeł **Image Texture** z obrazem tła (por. str. 334). Dodaj także węzeł **Texture Coordinate** i podłącz współrzędne tekstury do jego wyjścia **Window**. Wyjście z węzła tekstury (**Image Texture:Color**) podłącz jako barwę shadera **Background** (Rysunek 9.11.3):



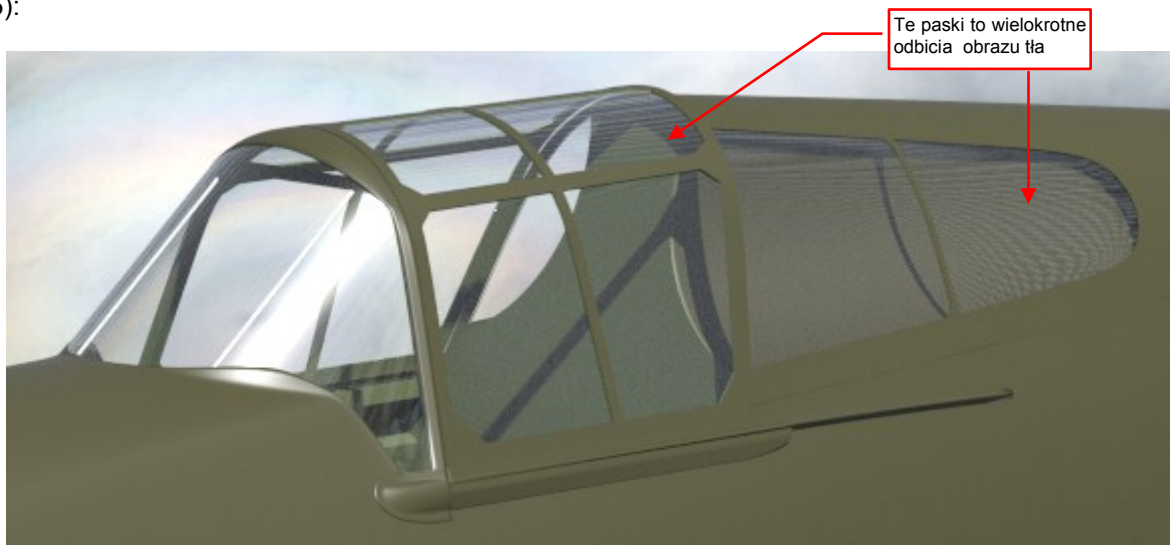
Rysunek 9.11.3 Schemat otoczenia sceny

Rysunek 9.11.4 przedstawia rezultat renderu — nasz model na tle zdjęcia:



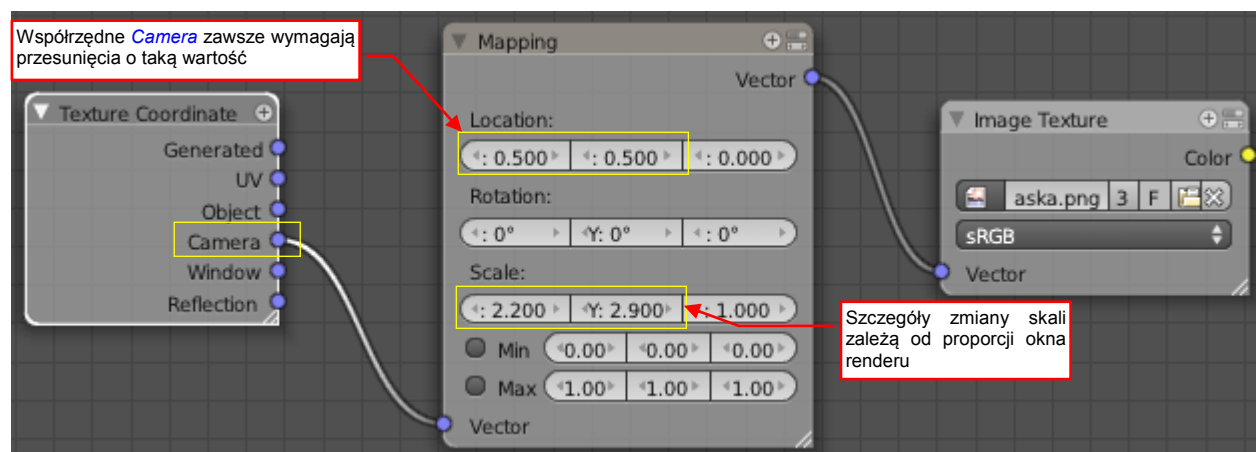
Rysunek 9.11.4 Rezultat: model z lepszym tłem

Efekt wygląda zadowalająco poza pewnym szczegółem: odbitym obrazem otoczenia. Jeżeli nawet samolot ma półmatową powierzchnię, na których tych odbić nie widać, to można je zobaczyć na oszkleniu kabiny (Rysunek 9.11.5):



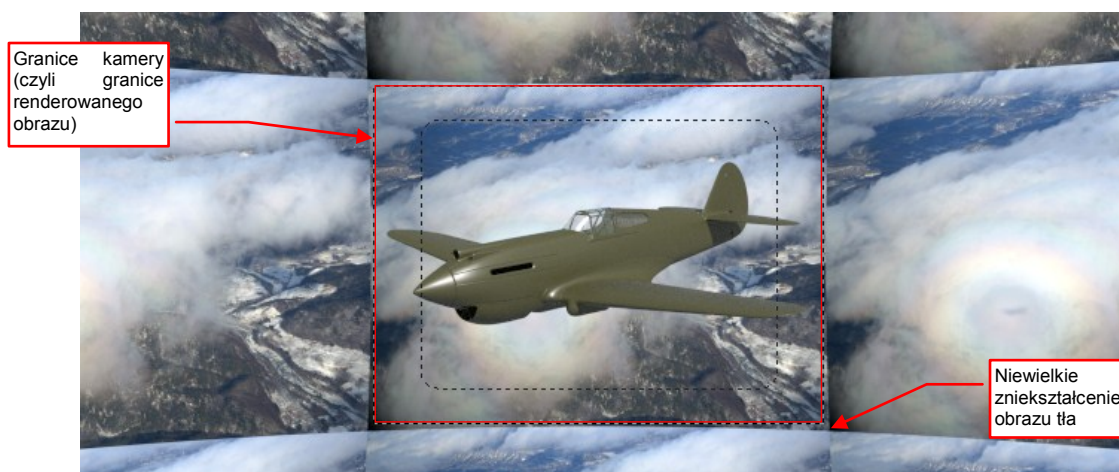
Rysunek 9.11.5 Problem z kształtem odbić otoczenia

Aby uzyskać inny układ odbić, trzeba inaczej rozłożyć obraz za modelem. W takim razie proponuję użyć **Texture Coordinate: Camera**, wraz z konieczną korektą węzłem **Mapping** (Rysunek 9.11.6):



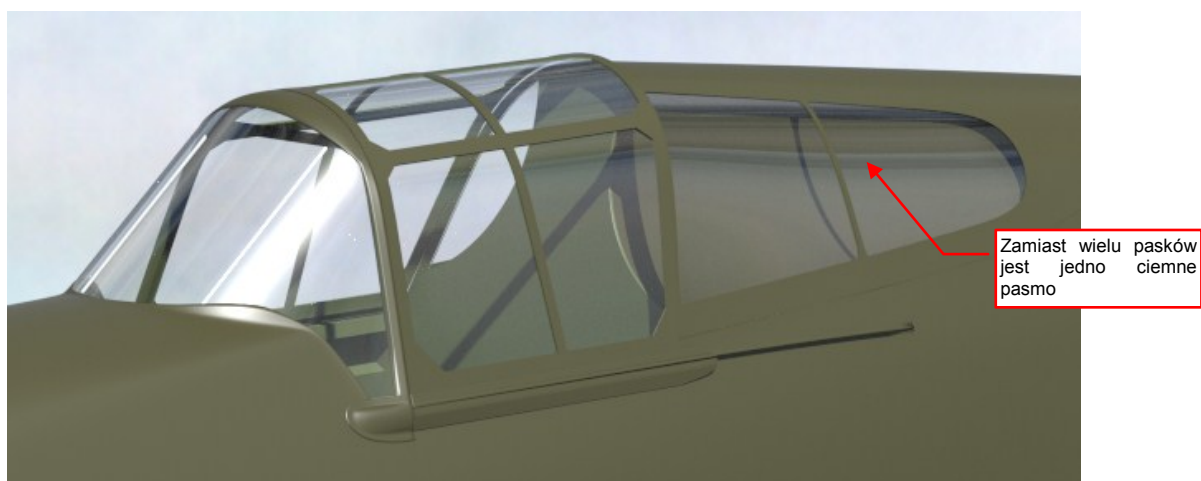
Rysunek 9.11.6 Zastosowanie innego mapowania obrazu

(Więcej na temat węzła **Mapping** znajdziesz na str. 332). Tak zastosowane współrzędne **Camera** pozwalają uzyskać za modelem niemal nie zdeformowany obraz (Rysunek 9.11.7):



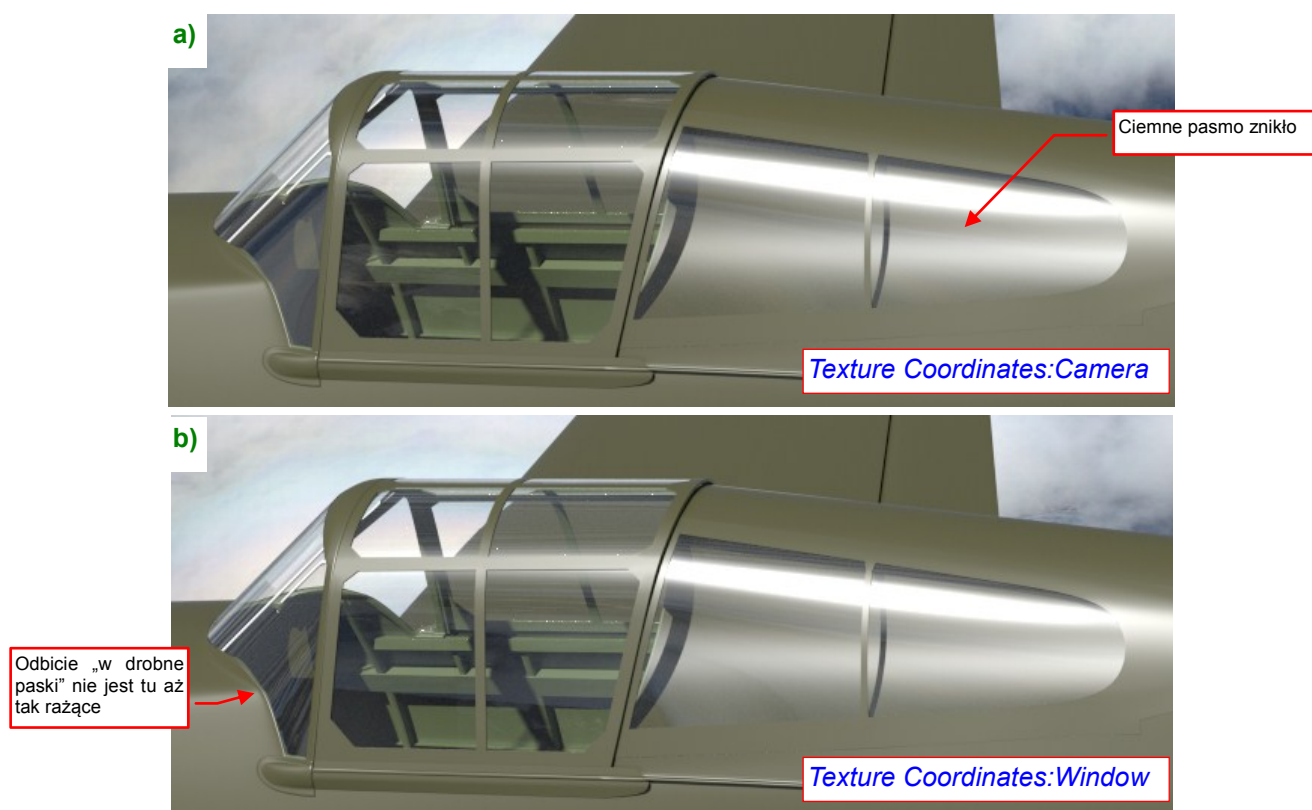
Rysunek 9.11.7 Nowe ułożenie obrazu tła na renderze

W zbliżeniu odbicie otoczenia w oszkleniu kabiny wygląda inaczej. Zamiast wielu drobnych pasków jest jeden, ale większy (Rysunek 9.11.8):



Rysunek 9.11.8 Układ odbłasków uzyskany dla mapowania *Texture Coordinate:Camera*

Spróbuj zmienić trochę punkt widzenia, a przekonasz się, że dla wielu ujęć uzyskany układ odbłasków wygląda zupełnie nieźle (Rysunek 9.11.9a):



Rysunek 9.11.9 Porównanie odbłasków dla różnych mapowań obrazu

Możesz także sprawdzić, że odbłyски uzyskane z użyciem *Texture Coordinate:Window* także wyglądają „znośnie” dla innych położen kamery (Rysunek 9.11.9b).

Na koniec warto zauważyć, że cały ten problem „szklanych odbić” dotyczy tylko zbliżeń. W scenach, w których widać cały samolot, kabina jest zazwyczaj niewielka. Widoczne na niej półprzezroczyste odbicia otoczenia są w takim przypadku zbyt małe, by mogły popsuć efekt. Co innego, gdy połyskliwa jest cała powierzchnia samolotu (a tak jest w przypadku maszyn utrzymanych w barwie naturalnego duralu). Wówczas trzeba się uciec do bardziej złożonych kompozycji shaderów otoczenia (por. str. 350).



## 9.12 Komponowanie otoczenia sceny

W scenach "na wolnym powietrzu", jakimi zazwyczaj są sceny lotnicze, trzeba umieścić wokół modelu imitację otoczenia. Najprościej jest "wstawić" za model jakąś fotografię — tak jak jest to opisane na str. 347. To nie wystarczy, gdy samolot na naszej scenie ma mieć powierzchnię w naturalnym kolorze duralu. Aby na takiej powierzchni pojawiły się przekonujące odbicia otoczenia, potrzebujemy obrazu, który zostanie rozpostarty wokół sceny. Chodzi o kulistą "panoramę", obejmującą 360° równika, i 180° południków (od zenitu do nadiru — por. str. 336). Obrazy tego rodzaju można znaleźć w Internecie, ale nie są tak liczne, jak zwykle zdjęcia. Zazwyczaj, panoramy dostępne za darmo mają za małą rozdzielczość, by można je było zastosować do pełnego odwzorowania tła (do tego celu potrzebne są obrazy rzędu 10000x5000 pikseli lub większe — por. str. 337). Dlatego musimy tu użyć złożenia "płaskiego zdjęcia za modelem", stanowiącego podstawowe tło obrazu, z panoramą otoczenia, zapewniającą odpowiednie odbłaski (Rysunek 9.12.1):



Rysunek 9.12.1 Przykład otoczenia złożonego z płaskiego obrazu i panoramy (Tło: © Tomo Yun, [www.yunphoto.net](http://www.yunphoto.net))

Na szczęście takie pomocnicze panoramy nie muszą mieć bardzo dużych rozdzielczości, i darmowe pliki, udostępniane w Internecie, zupełnie do tego celu wystarczą. W tej sekcji pokażę szczegóły tego rozwiązania.

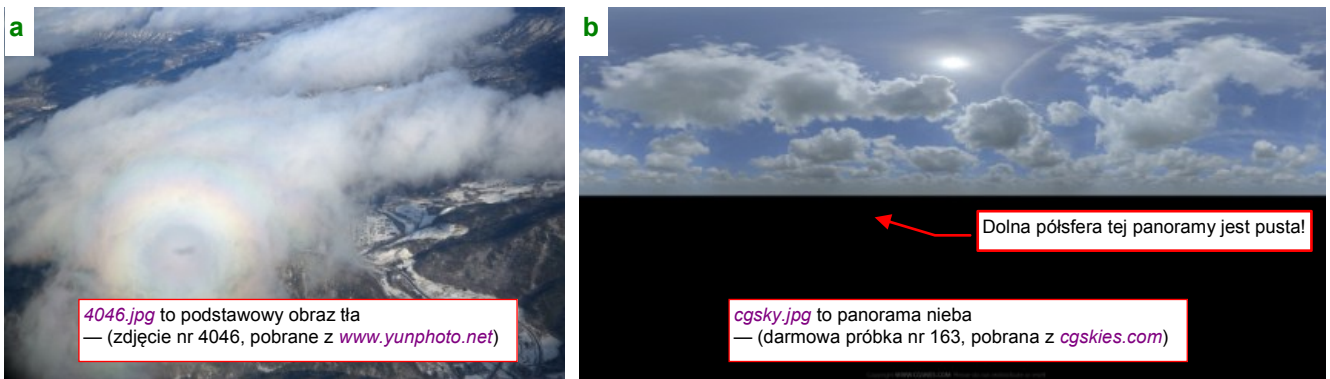
- Podczas poszukiwań w Internecie możesz się natknąć na panoramy zapisywane w specjalnych formatach: **HDR** lub **EXR**<sup>1</sup>. Taka forma zapisu wymaga nieco innych ustawień parametrów (por. str. 336)

Podstawowym obrazem, od którego zaczynamy kompozycję sceny, jest zwykłe, płaskie zdjęcie które podstawimy jako tło renderowanego obrazu. Czasami bywa tak, że masz pomysł na scenę i szukasz w Internecie odpowiedniego otoczenia. Czasami — na odwrót, znajdziesz zdjęcie, w którym dostrzeżesz jakąś nową, ciekawą możliwość. W każdym razie w podczas takich poszukiwań zawsze warto zajrzeć na [www.yunphoto.net](http://www.yunphoto.net). Autor oferuje tam za darmo różne zdjęcia w dużej rozdzielczości, m.in. nieba widzianego z samolotu.

<sup>1</sup> To dość nietypowe obrazy. Zwykły plik graficzny przechowuje informację o barwach w postaci tzw. "8-bitów na składnik" („per channel”). (Oznacza to przypisanie każdemu ze składników: **R,G,B** 8-bitowej intensywności — więcej na ten temat na str. 1218). Taki zapis umożliwia odwzorowanie do 24 milionów barw. Zazwyczaj to wystarczy (oko ludzkie nie rozróżnia ich więcej). Ale nie w przypadku panoram! Ich barwy są zapisane z większą dokładnością: 10, 12 i więcej bitów na składnik. Oznacza to, że potrafią odwzorować nie miliony, a miliardy odcieni. Ma to znaczenie przy zaawansowanych technikach imitacji oświetlenia (tzw. *global illumination* — *GI*). Stąd panoramy są zapisywane w specjalnych formatach — **\*.HDR**, **\*.EXR**. To obrazy określane jako *High Dynamic Range Image* — **HDRI**. Nie wszystkie programy graficzne potrafią obsługiwać pliki o takich "rozszerzonych" kolorach. Możesz je wczytać i oglądać w *UV/Image Editor* Blendera. Niestety, zestaw narzędzi do edycji obrazu, dostępnych w tym edytorze, jest dość ubogi. Nie można ich wczytać do GIMP, bo ten program jeszcze nie obsługuje takich plików.



W tej sekcji jako podstawowy obraz tła wykorzystamy plik, który nazwałem **4046.jpg** (Rysunek 9.12.2a). (To właśnie jedna jedno ze zdjęć z samolotu wykonane przez Tomo Yuna, udostępnione pod nr 4046):

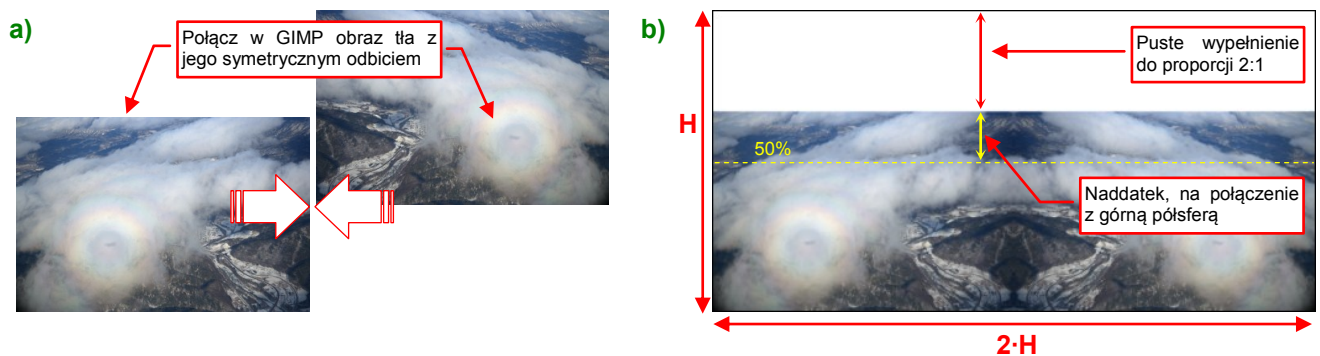


Rysunek 9.12.2 Źródła obrazów, wykorzystywanych w tej sekcji

Teraz trzeba dobrać do tego zdjęcia panoramę nieba. Dla Cycles szukaj zawsze panoram w tzw. rozwinięciu cylindrycznym (por. str. 336). To takie dobrze znane rozwinięcie Merkatora, stosowane od wieków w geografii. Rysunek 9.12.2b) przedstawia panoramę, która wydaje mi się dość dobrze pasować do **4046.jpg**. To darmowa próbka panoramy nr 163, udostępniona wyspecjalizowanym serwisie **cgskies.com**. Nazwałem ją **cgsky.jpg**.

- Wykorzystywany w tej sekcji obraz: **4046.jpg** nie jest niezbędny do wykonania modelu. Dlatego umieściłem go w dodatkowym pliku **optional.zip** (por. str. 16). Pozostałe pliki, które tutaj używam, pochodzą z pliku **p40.zip**, i znajdziesz w folderze **model/p40/background**.

Zwróć uwagę, że w panoramie **cgsky.jpg** brakuje dolnej połowy. Autorzy po prostu założyli, że dolna półsfera będzie się zawsze poza polem widzenia kamery. Na potrzeby naszej sceny w locie sami musimy wypełnić ten obszar obrazem „ziemi”. Najlepiej, aby odbicie tej półsfery na powierzchni samolotu pasowało do podstawowego tła (obrazu **4046.jpg**). Proponuję więc uzyskać ją z przekształcenia tego właśnie obrazu (Rysunek 9.12.3):



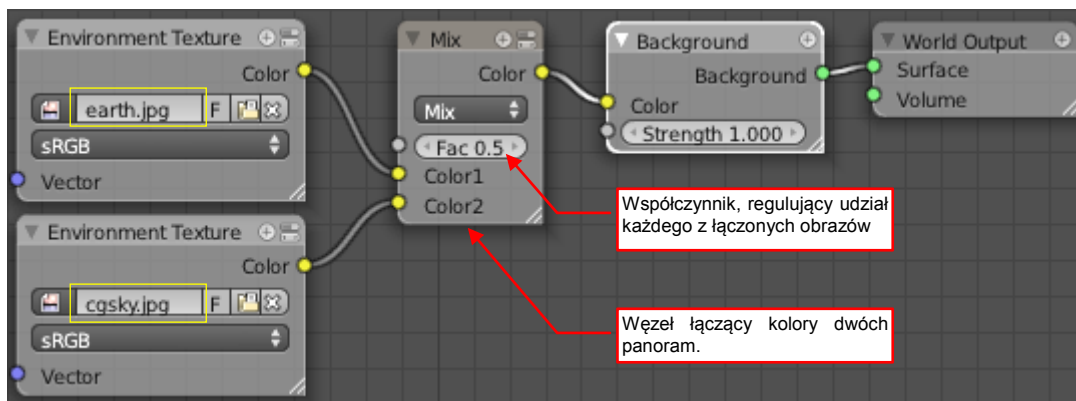
Rysunek 9.12.3 Tworzenie panoramy dolnej półsfery (**earth.jpg**)

Załaduj obraz **4046.jpg** do GIMP. Na początek możesz nieco zmniejszyć jego rozmiar (**Image → Scale Image**), na przykład dwukrotnie. (Oryginalne zdjęcie ma szerokość niemal 5000px, a w tej pomocniczej panoramie nie potrzeba aż takich rozdzielczości).

Następnie zduplikuj domyślną warstwę GIMP z obrazem (**Background**) i przerzuć tę kopię w poziomie, aby stworzyć „lustrzane odbicie” oryginalnego zdjęcia (Rysunek 9.12.3a). Zwiększ szerokość obrazu (**Image → Canvas Size**) dwukrotnie, a następnie starannie przysuń do siebie te dwie połówki. W ten sposób uzyskaliśmy podstawowy obraz, na którym nie będzie widać linii łączenia. To nic, że składa się z dwóch identycznych części. W odbiciach na powierzchni samolotu nie będzie tego widać.

Na koniec zwiększ wysokość obrazu (**Image → Canvas Size**) tak, by stała się równa połowie szerokości (Rysunek 9.12.3b). (To typowe proporcje obrazu panoramy). Ta część obrazu i tak będzie „zasłonięta” przez górną półsferę (**cgsky.jpg**). Oryginalne zdjęcie **4046.jpg** miało proporcje 2:3, stąd zajmuje 2/3 wysokości panoramy. Taka „zakładka” może się jednak przydać, gdy będziesz chciał opuścić trochę ten obraz ziemi do dołu. Uzyskanemu w ten sposób plikowi z panoramą dolnej półsfery nadałem nazwę **earth.jpg**.

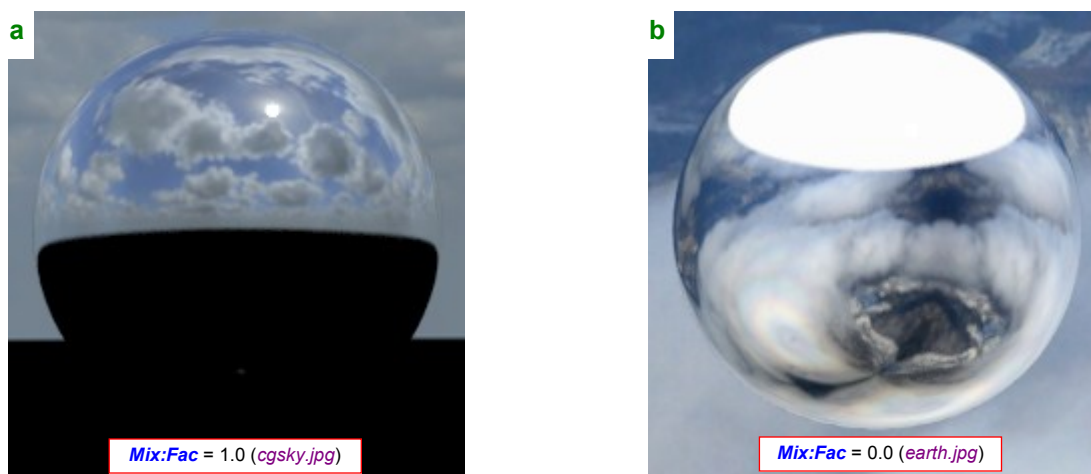
Rysunek 9.12.4 przedstawia złożenie obrazów górnej i dolnej półsfery (*Environment Texture* — por. str. 336):



Rysunek 9.12.4 Złożenie panoramy z dwóch obrazów (*earth.jpg* i *cgsky.jpg*)

Mogłem dla każdy z użytych obrazów podłączyć do odrębnego shadera *Background*, połączonych później za pomocą węzła *Mix Shader*. Zamiast łączenia shaderów wolałem połączyć w jeden obraz przygotowane panoramy. Stąd wykorzystuję do tego węzeł *Color→Mix*, którego wyjście jest połączone z pojedynczym shaderem otoczenia (Rysunek 9.12.4).

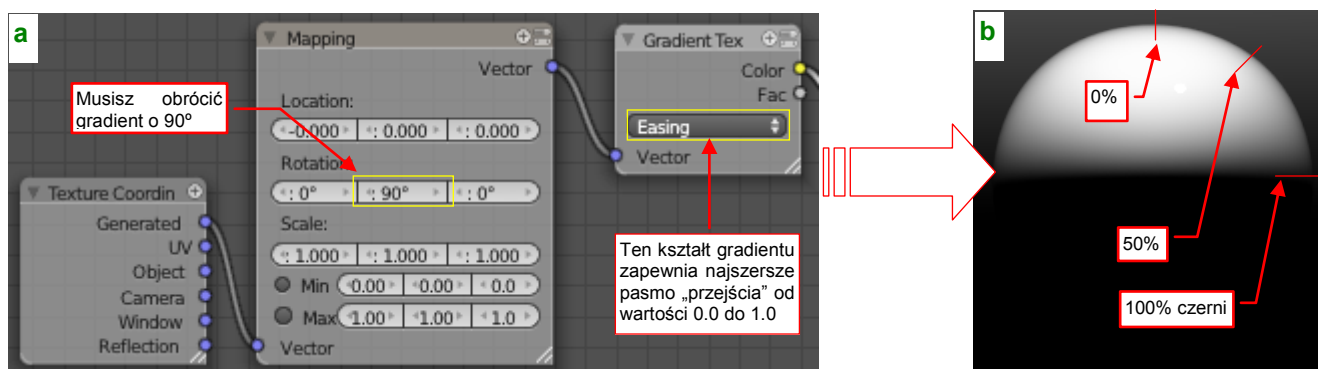
Na początek warto sprawdzić za pomocą sondy (por. str. 330), czy każda z panoram „układa” się tak jak tego byśmy chcieli (Rysunek 9.12.5):



Rysunek 9.12.5 Sprawdzenie obydwu panoram za pomocą sondy

Dla *Mix:Fac* = 0 scena jest otoczona wyłącznie przez panoramę *earth.jpg* (Rysunek 9.12.5b), a dla *Mix:Fac* = 1.0 — przez *cgsky.jpg* (Rysunek 9.12.5a).

Do właściwego złożenia tych dwóch obrazów potrzebujemy zmiennego współczynnika podziału. Użyjemy w tym charakterze obróconego o 90° gradientu typu *Easing* (*Gradient Texture* — str. 340) (Rysunek 9.12.6):



Rysunek 9.12.6 Współczynnik podziału: gradient liniowy

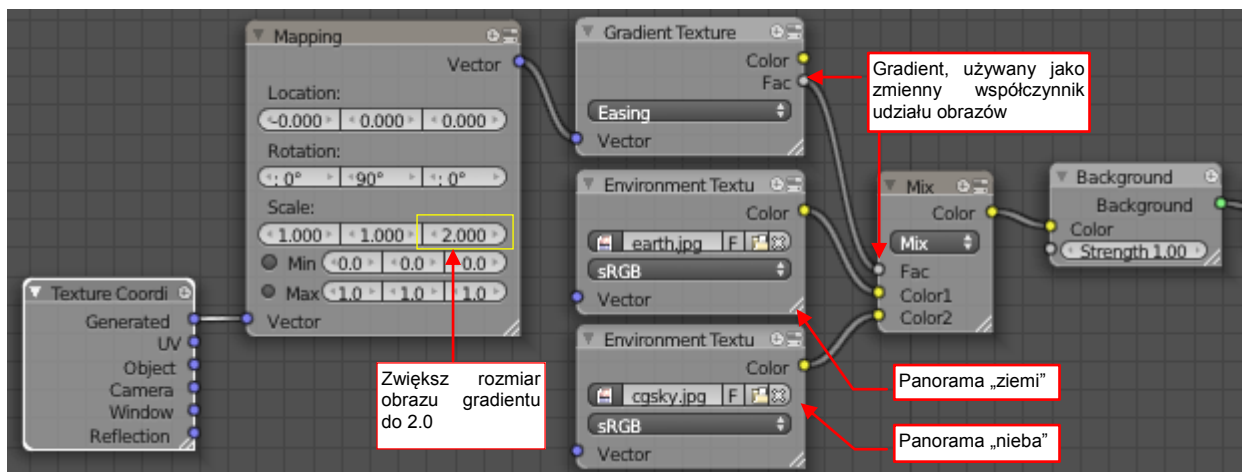
Rysunek 9.12.7 przedstawia rezultat podłączenia wyjścia *Gradient Texture:Fac* (Rysunek 9.12.6) do wejścia *Mix:Fac* zespołu pokazywanego przez Rysunek 9.12.4. Po pierwsze — coś jest nie tak z tym naszym gradientem, bo w górnej części sfery widać nadal górną granicę obrazu tła (por. Rysunek 9.12.5).

Jednocześnie warto przesunąć trochę źródło światła kierunkowego (**Sun**), aby jego odbicie pokrywało się z obrazem słońca na panoramie otoczenia (Rysunek 9.12.7). (W tym przykładzie te dwa punkty były bardzo blisko siebie, więc takie drobne poprawki położenia obiektu **Sun** nie stanowiły problemu. W przypadku większych różnic konieczny może się okazać obrót panoramy nieba — por. str. 355, Rysunek 9.12.12).



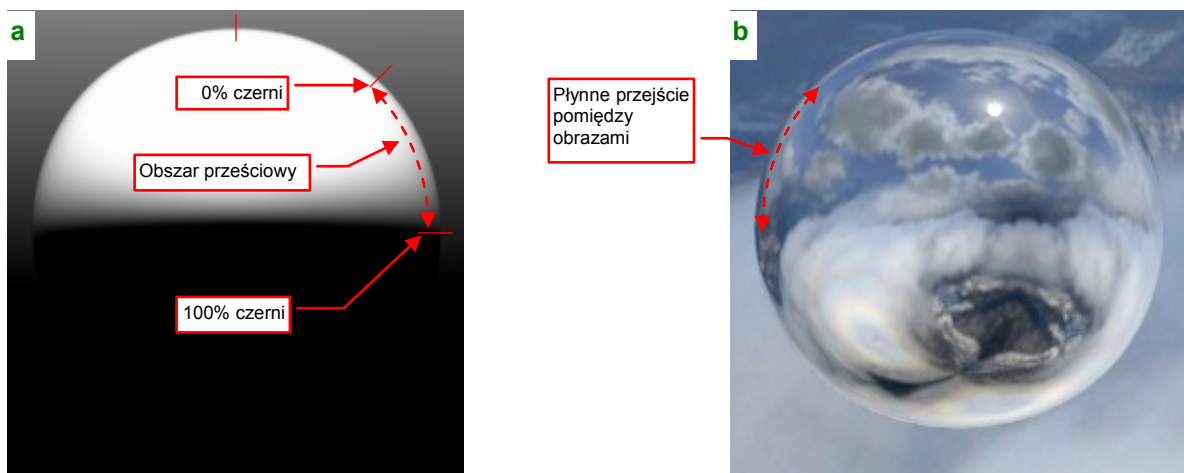
Rysunek 9.12.7 Pierwsza próba połączenia sfer

Dlaczego na górnej półsfery poprzez panoramę nieba (*cgsky.jpg*) prześwituje ziemia (*earth.jpg* — Rysunek 9.12.7)? Popatrz jeszcze raz na Rysunek 9.12.6b). Zwróć uwagę, że dopiero w połowie górnej półsfery gradient uzyskuje wartość 50%. A taka wartość oznacza, że w tym miejscu zobaczysz złożenie obydwu łączonych obrazów! 100% nieba występuje na razie w jednym punkcie: w zenicie. Aby zwiększyć ten obszar, powiększ obraz gradientu wzdłuż osi **Z** (*Mapping:Scale:Z*) — na przykład 2 razy (Rysunek 9.12.8):



Rysunek 9.12.8 Zastosowanie gradientu jako współczynnika łączenia dwóch sfer

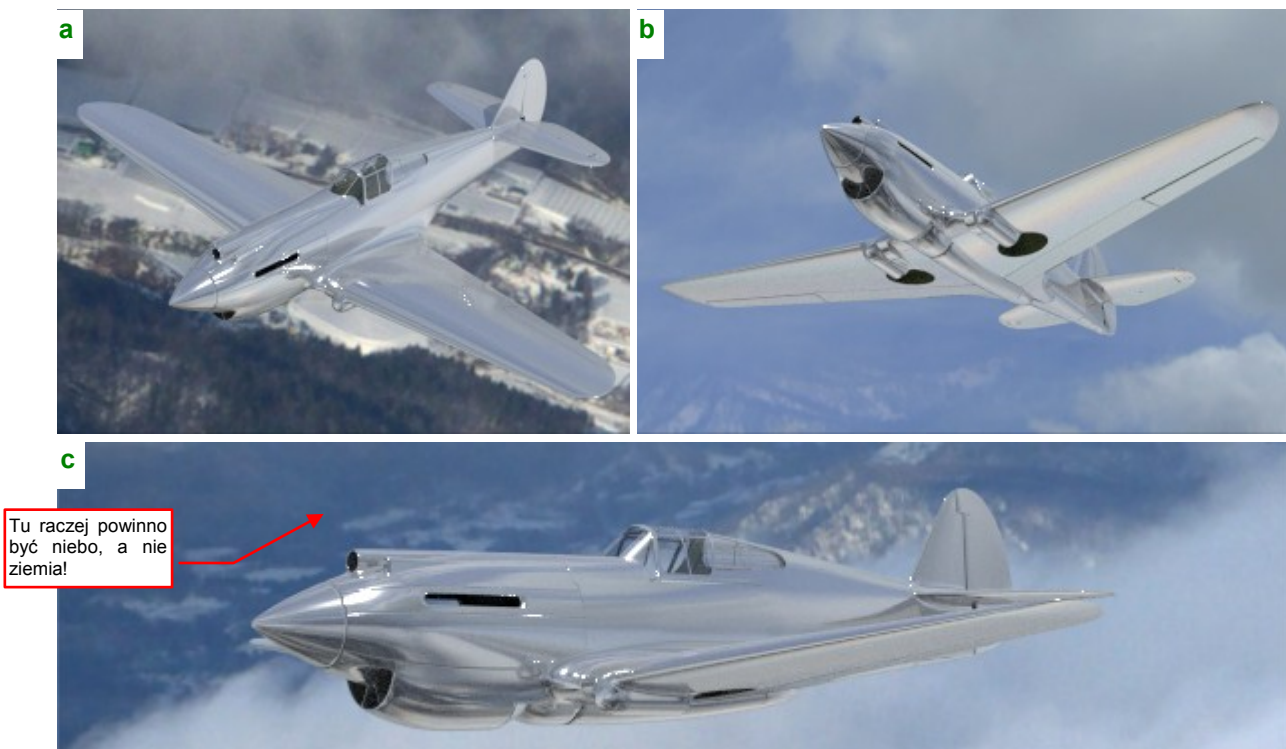
Rysunek 9.12.9 przedstawia nowy rozkład gradientu (a) oraz uzyskaną panoramę (b):



Rysunek 9.12.9 Poprawiony gradient do łączenia dwóch sfer

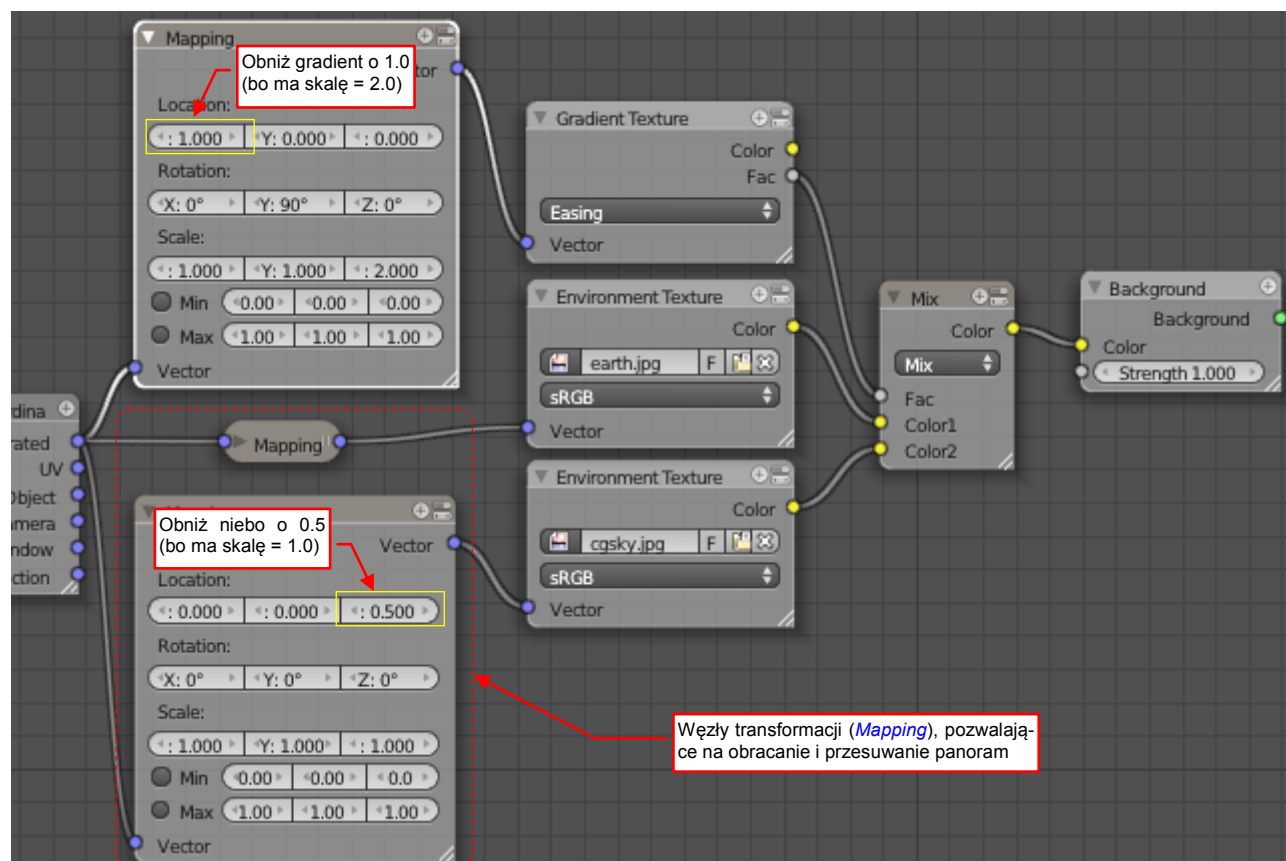


Po tej poprawce panorama wydaje się całkiem poprawna — sprawdźmy, jak wygląda w tym otoczeniu nasz model (Rysunek 9.12.10):



Rysunek 9.12.10 Model w otoczeniu panoramy (*earth.jpg* + *cgsky.jpg*)

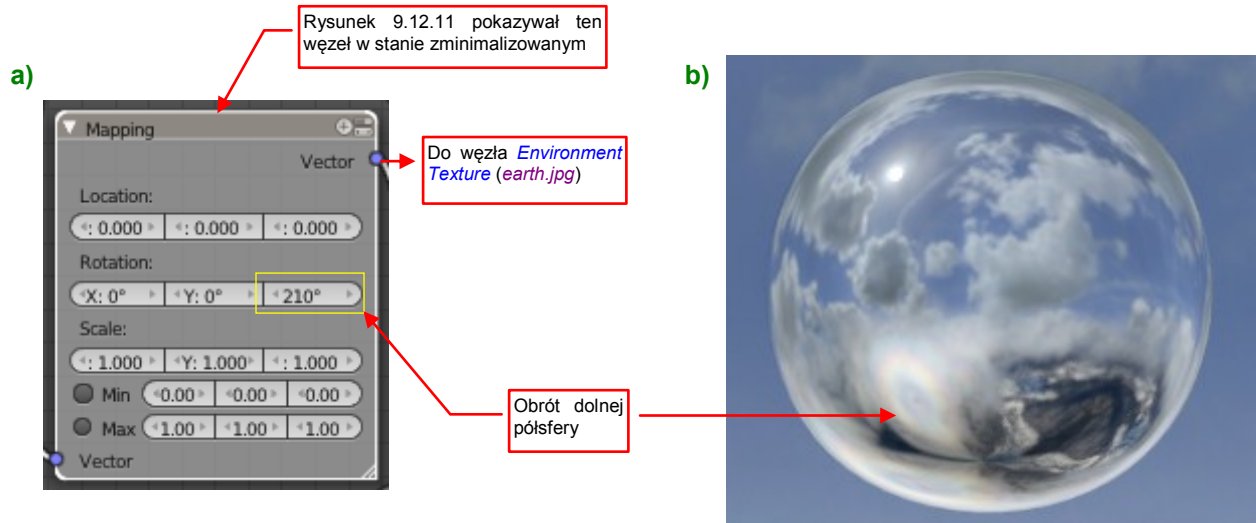
O ile ujęcia z góry i z dołu (Rysunek 9.12.10a,b) wyglądają poprawnie, to w widoku poziomym (Rysunek 9.12.10c) horyzont położony jest zbyt wysoko. Trzeba go obniżyć (Rysunek 9.12.11):



Rysunek 9.12.11 Przesunięcie linii horyzontu do dołu



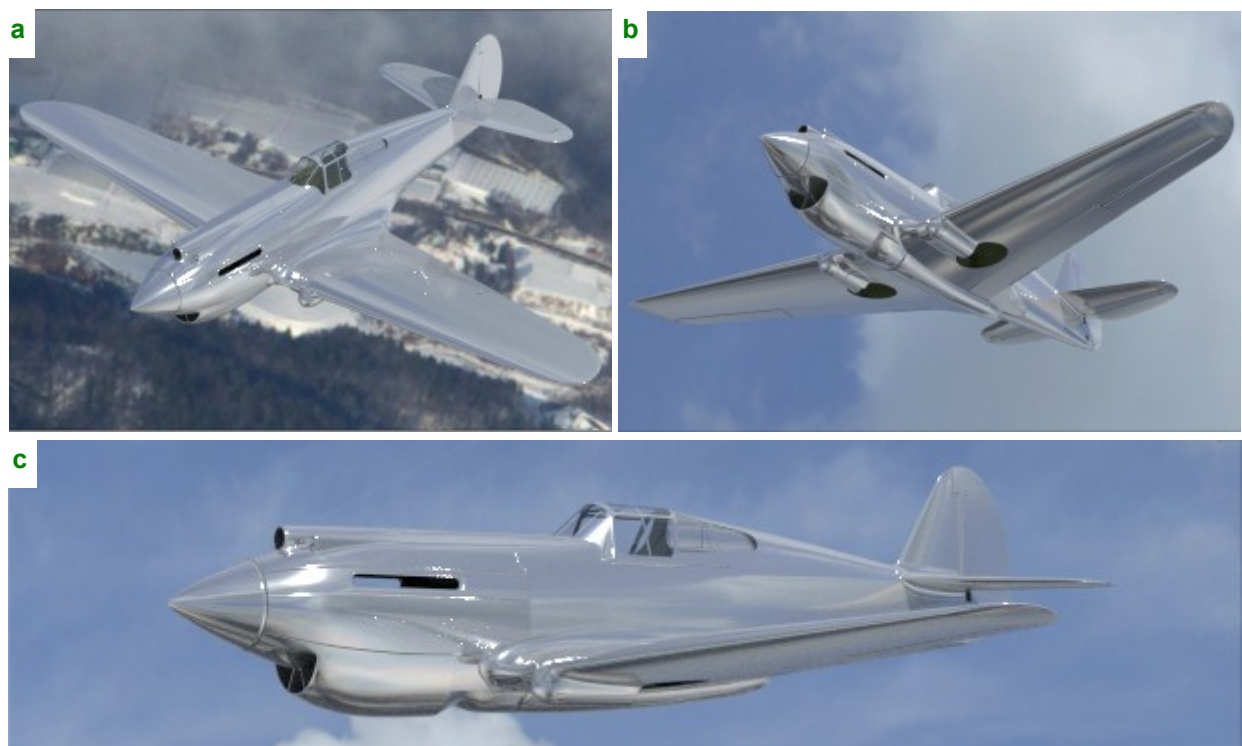
Aby zmieścić na poprzedniej stronie Rysunek 9.12.11, zmniejszyłem na nim węzeł transformacji dolnej półsfery (*earth.jpg*). Zrobiłem tak gdyż ta panorama nie ulega żadnej zmianie przy obniżaniu horyzontu. Warto jednak podłączyć do jej wejścia taki sam węzeł transformacji (*Mapping*) jak do półsfery nieba, aby móc swobodnie nią obracać (Rysunek 9.12.12a):



**Rysunek 9.12.12 Panorama otoczenia (po dodatkowym obrocie panoramy *earth.jpg*)**

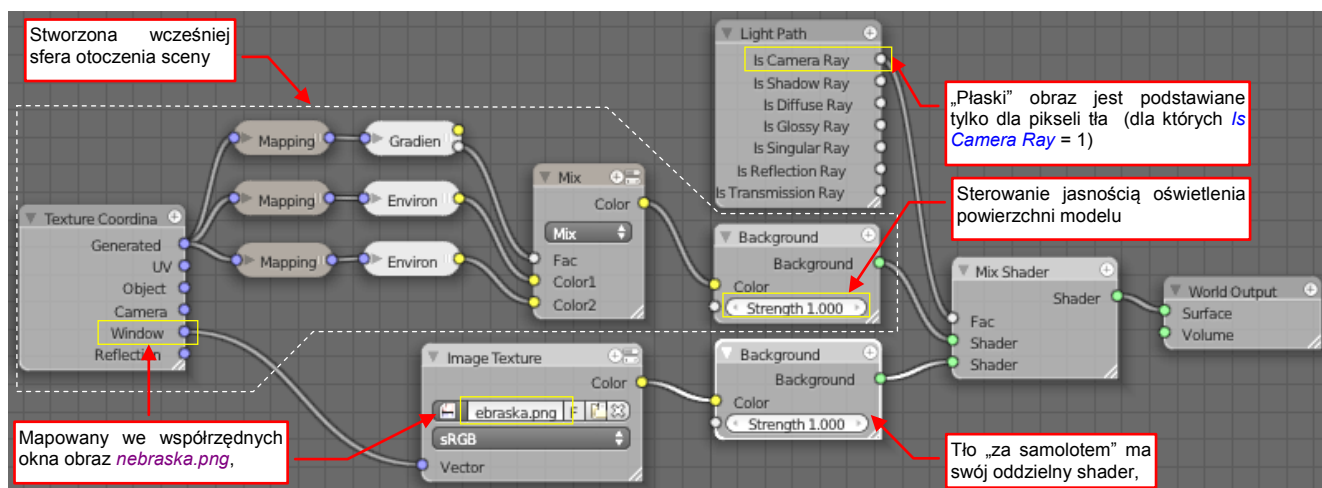
(Za pomocą obrotu dolnej półsfery możesz zmieniać układ odbić otoczenia, widocznych na dolnych powierzchniach modelu). Uzyskane w wyniku tej i poprzedniej zmiany otoczenie sceny przedstawia Rysunek 9.12.12b). Jak widać, horyzont uległ tu znacznemu obniżeniu.

Sprawdźmy, jak to wpłynęło na ujęcia naszego modelu. Dla widoku z góry (Rysunek 9.12.13a) właściwie nic się nie zmieniło (por. Rysunek 9.12.10a). Na tle ujęcia z dołu (Rysunek 9.12.13b) majaczące w oddali odległe, błękitne góry (por. Rysunek 9.12.10b) zostały zastąpione przez czyste niebo. Tę zmianę widać jeszcze bardziej dobitnie w widoku z boku (Rysunek 9.12.13c). Poprzednio nasz samolot wyglądał jakby leciał między górami (Rysunek 9.12.10c), a teraz występuje na tle błękitnego nieba:



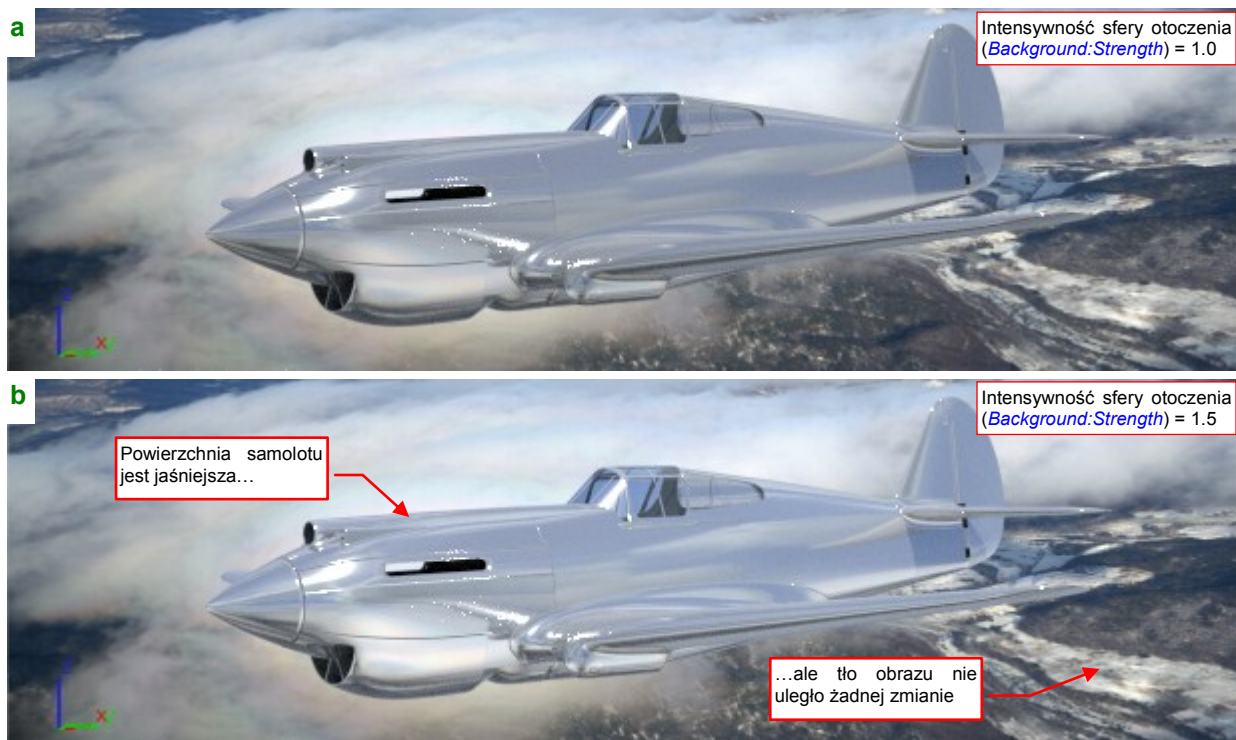
**Rysunek 9.12.13 Model w otoczeniu poprawionej panoramy (*earth.jpg* + *cgsky.jpg*)**

Dodajmy teraz do naszego schematu drugi element: konwencjonalny obraz (*Texture Image*) tła, podstawiony za modelem (Rysunek 9.12.14):



Rysunek 9.12.14 Złożenie panoramy z „płaskim” obrazem tła

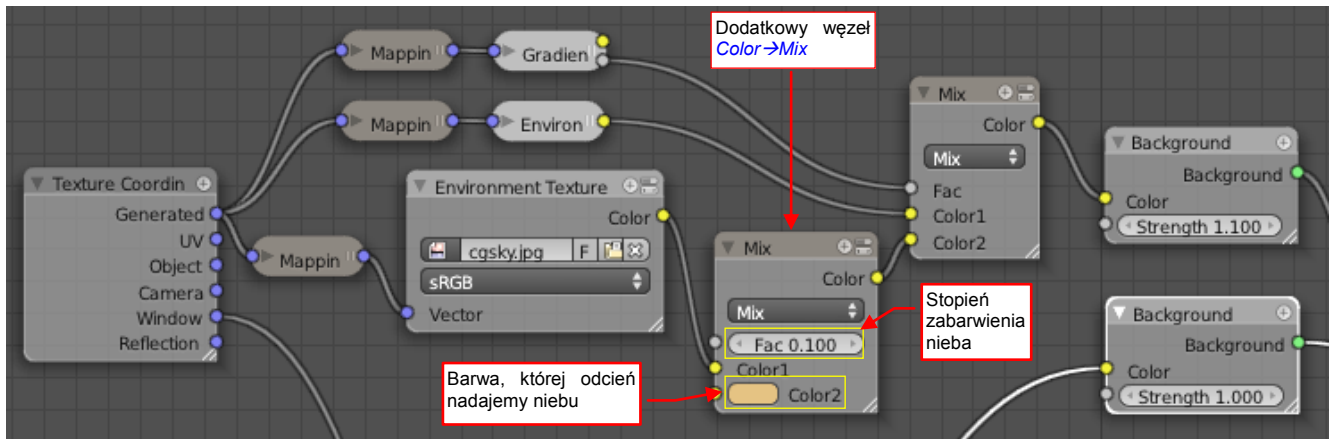
Zastosowałem tu rozwiązanie opisane na str. 347: konwencjonalny obraz (*Texture Image*) podstawiony we współrzędnych okna (*Texture Coordinate:Window*). Do rozróżnienia promieni (pikseli renderowanego obrazu) odbitych od modelu i od tła użyłem flagi *Light Path:Is Camera Ray* (str. 324). Dla pierwszych (*Is Camera Ray* = 0) wykorzystywana jest przygotowana wcześniej panorama otoczenia. Dla drugich (*Is Camera Ray* = 1) podstawiany jest obraz *nebraska.png*<sup>1</sup>. Mogłem połączyć te obraz kolejnym węzłem *Color→Mix* z flagą *Is Camera Ray* w charakterze współczynnika udziału (*Fac*). Zdecydowałem się jednak podłączyć każdy z tych obrazów do oddzielnego shadera typu *Background*, które są łączone węzłem *Mix Shader* (Rysunek 9.12.14). Taki układ pozwala na niezależną regulację intensywności oświetlenia powierzchni modelu i tła (Rysunek 9.12.15):



Rysunek 9.12.15 Regulacja jasności panoramy otoczenia

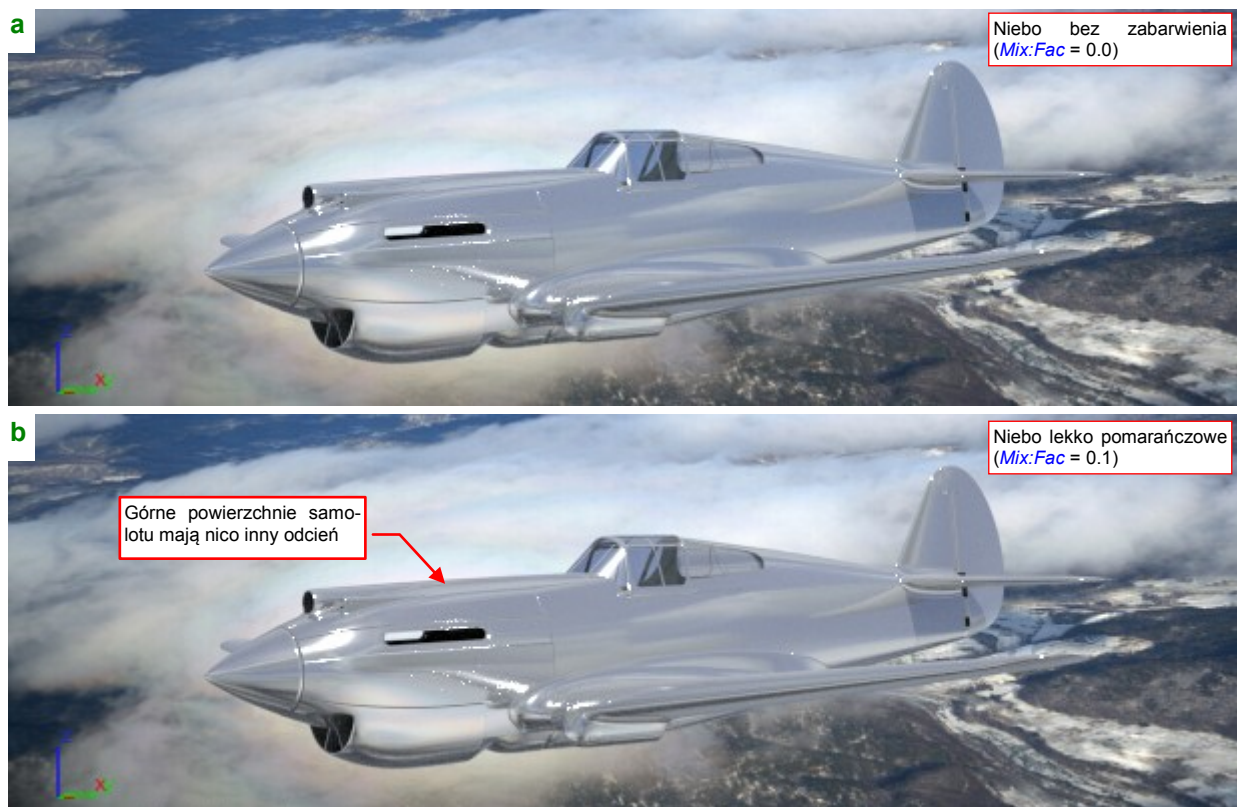
<sup>1</sup> Ten obraz znajdziesz w folderze *model/p40/background* (por. str. 20). Jest wykorzystywany w innych rozdziałach tej książki. To zmniejszona wersja zdjęcia *4046.jpg*, które wykonał Tomo Yun ([www.yunphoto.net](http://www.yunphoto.net)). Obraz ma nieco zmienione proporcje. Usunąłem z niego co wyraźniejsze ślady ludzkich zabudowań. Nazwa jest wynikiem mojej pomyłki (w rzeczywistości to fragment Japonii), ale taka już została.

Czasami przydaje się dodać niebu jakiś odrobinę inny odcień. Aby to zrobić, wstaw nowy element typu *Color→Mix* w środek linii łączącej wyjście z *Environment Texture* (z *cgsky.jpg*) z węzłem *Color→Mix* skalającym górną i dolną panoramę (Rysunek 9.12.16):



Rysunek 9.12.16 Dodatkowy efekt — odcień nieba

Ustaw wejście *Mix:Color2* nowego węzła na barwę, która „pasuje” do obrazu tła sceny. Dla takiego otoczenia jak *nebraska.png* zastosowałem kolor pomarańczowy. Trybem łączenia jest *Mix*, a do wejścia *Mix:Color1* podłączyłem obraz z *Environment Texture*. Intensywnością zabarwienia nieba steruje współczynnik *Mix:Fac*. Ustaw go na jakąś niewielką wartość — np. 0.1 (10%). Rysunek 9.12.17 przedstawia efekt takiej niewielkiej zmiany:

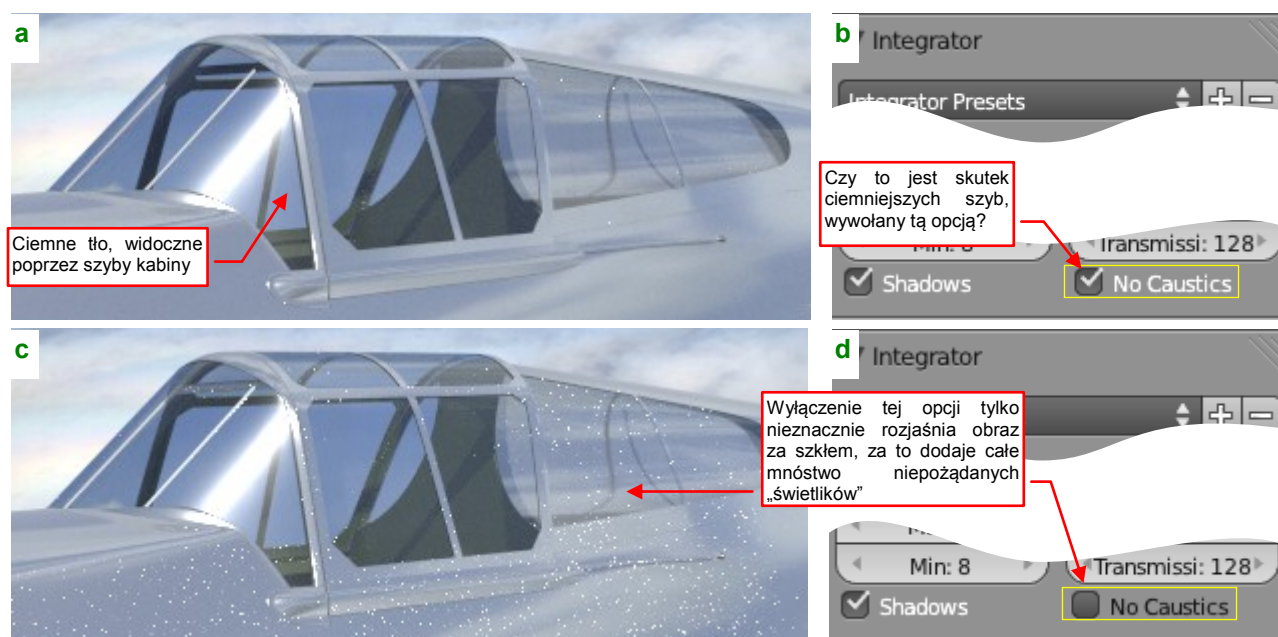


Rysunek 9.12.17 Regulacja „odcienia nieba”

Rysunek 9.12.17a) to wersja obrazu stworzona z użyciem obrazu nieba bez żadnej korekty. Rysunek 9.12.17b) przedstawia obraz, w którym niebo stało się lekko pomarańczowe (gdyż zmieniłem *Mix:Fac* na 0.1). Trzeba przyznać, że efekt jest „kosmetyczny”. Pomaga lepiej „dopasować” model do obrazu tła. Coś podobnego można także uzyskać w inny sposób — np. operując odpowiednimi obrazami w kompozytorze Blendera (por. str. 438). Ja jednak takie proste poprawki wolę wstawiać bezpośrednio do schematu otoczenia sceny. Uważam, że w ten sposób łatwiej z nich skorzystać, niż z *Render Layers* kompozytora.



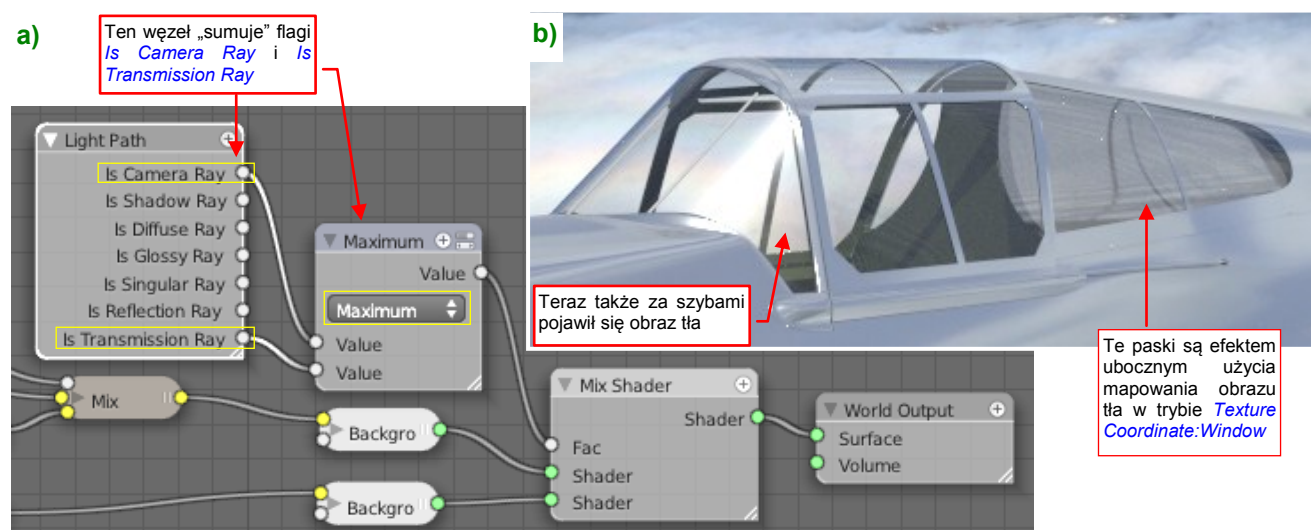
Pewien szczegół naszego modelu nadal nie pasuje do tła: za szybami kabiny wciąż widać błękitny fragment panoramy otoczenia, a nie obraz tła (Rysunek 9.12.18a):



Rysunek 9.12.18 Wpływ opcji **No Caustics** na render

W pierwszej chwili sądziłem, że to efekt zaciemnienia szyb, wywołany włączoną w panelu **Integrator** zestawu **Render** opcją **No Caustics** (Rysunek 9.12.18b). Po wyłączeniu tej opcji (Rysunek 9.12.18d) błękit widoczny za kabiną pozostał, wewnątrz kokpitu tylko nieznacznie się rozjaśniło, za to na kadłubie pojawiło się mnóstwo białych punktów („światlików” — Rysunek 9.12.18c), które zaczęły znikać dopiero po kilku tysiącach próbek. Czym prędzej z powrotem włączyłem więc opcję **No Caustics**. Nie ona jest tutaj przyczyną.

Węzeł **Light Path** zalicza promienie przechodzące przez szybę do grupy **Is Transmission Ray**. A promień (piksel renderu) może należeć tylko do jednej kategorii — więc za szybą kabiny widać fragment błękitu naszej panoramy otoczenia, a nie obraz tła. W takiej sytuacji można tak zmienić schemat, by podstawiał obraz tła i dla **Is Camera Ray**, i dla **Is Transmission Ray** (Rysunek 9.12.19a):



Rysunek 9.12.19 Uzyskanie obrazu tła za oszkleniem kabiny

Chodzi tu o sumę „logiczną”, która zwróci 1 gdy którakolwiek z tych flag będzie równa 1. Ten efekt można uzyskać za pomocą węzła **Math**, ustawionego na funkcję **Maximum** (Rysunek 9.12.19a). W efekcie na renderze za kabiną pojawi się odpowiednie tło obrazu (Rysunek 9.12.19b). Jediną wadą tego rozwiązania są nietypowe odbłaski otoczenia na szybach (szersze omówienie tego problemu znajdziesz na str. 348).



Należy zauważyć, że te odbłaski są w wyniku „zmieszania” z panoramą otoczenia słabsze. W razie czego pamiętaj, że rozwiązanie oparte na sumowaniu *Is Camera Ray* z *Is Transmission Ray* warto stosować tylko wtedy, gdy przez oszklenie widać jakiś fragment tła sceny. Czasami możesz tak ustawić kamerę, by „za szkłem” były widoczne wyłącznie powierzchnie modelu. Wtedy możesz zastosować tylko flagę *Light Path: Is Camera Ray* (jak na str. 356, Rysunek 9.12.14) dla której nie ma problemu ze środowiskiem odbitym w szybach. Innym sposobem na uniknięcie stosowania flagi *Is Transmission Ray* jest podstawienie „za szybą” modelu takiego fragmentu panoramy otoczenia, który kolorystycznie pasuje do obrazu tła. (W przypadku Rysunek 9.12.18a i c mógłby to być jakiś fragment białej chmury).

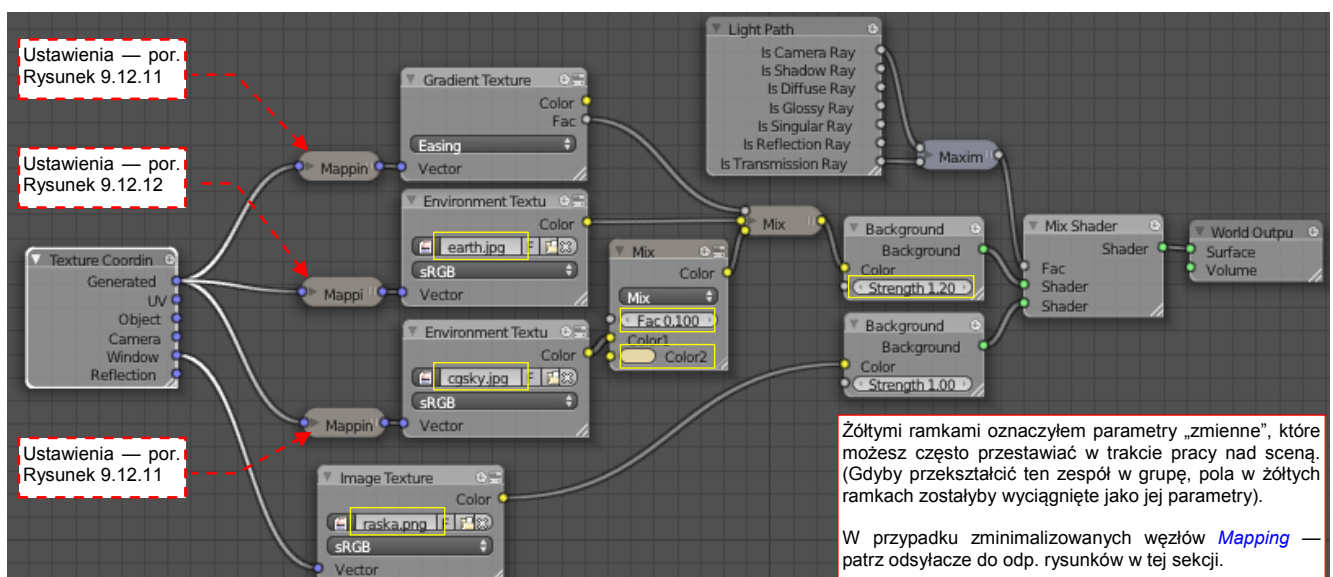
Zresztą — najczęściej te nieprawidłowe odbłaski, jakie uzyskamy stosując złożenie *Is Camera Ray* z *Is Transmission Ray* (Rysunek 9.12.19), nie są czymś specjalnie rażącym. Popatrz chociażby na model poniżej (Rysunek 9.12.20):



Rysunek 9.12.20 Rezultat — srebrzysty samolot na tle zimowego krajobrazu

Wydaje mi się, że nie zwróciłeś na nie uwagi także na renderze z początku tej sekcji (str. 350, Rysunek 9.12.1).

Rysunek 9.12.21 przedstawia uzyskany schemat otoczenia sceny. Aby zmieścić go na jednej ilustracji, pozostawiłem węzły transformacji (*Mapping*) gradientu i panoram w stanie „zminimalizowanym”:

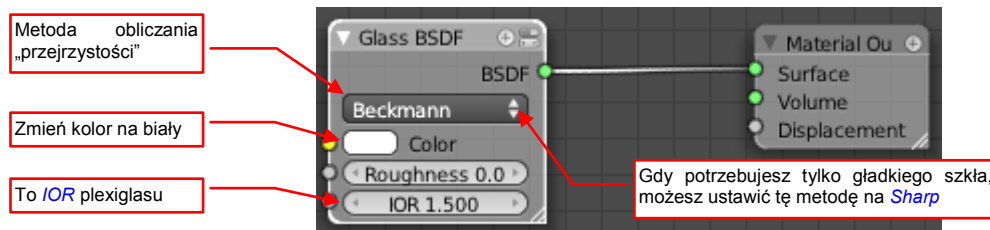


Rysunek 9.12.21 Pełen schemat otoczenia sceny (World)

Zastanawiałem się, jak można przekształcić te węzły w jakąś elegancką grupę. Jednak zrezygnowałem, gdyż obecnie w Cycles nie ma możliwości wygodnego „wyciągnięcia” wybranych parametrów z węzłów *Mapping*.

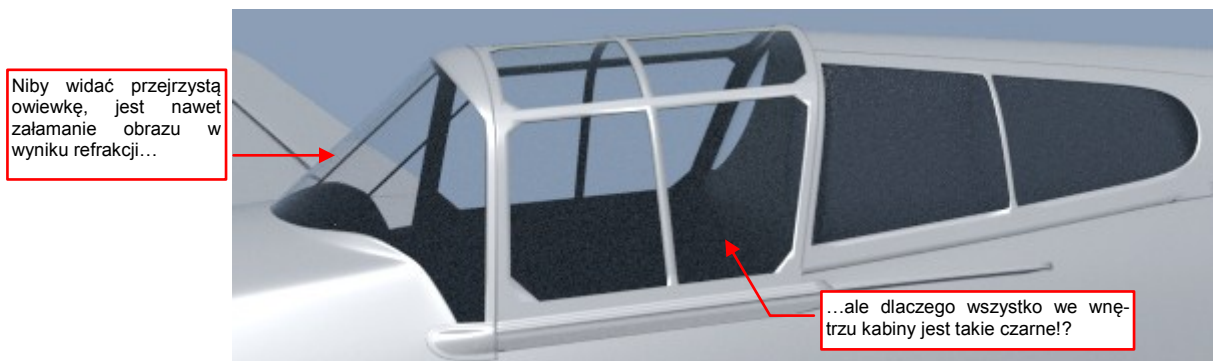
### 9.13 Skonfigurowanie materiału: szkło (*Plexiglas*)

Stwórz nowy materiał i przypisz go do elementów oszklenia kabiny. Nazwijmy go **B.Glass.Canopy**. Uzyskanie materiału odpowiadającego szkłu (a dokładniej: szkłu organicznemu) wydaje się w Cycles trywialne. Przecież mamy tu do dyspozycji gotowy shader **Glass BSDF**! Wykorzystajmy więc go w naszym materiale. Rysunek 9.13.1 pokazuje pierwszą wersję jego definicji, przygotowaną w *Node Editor*:



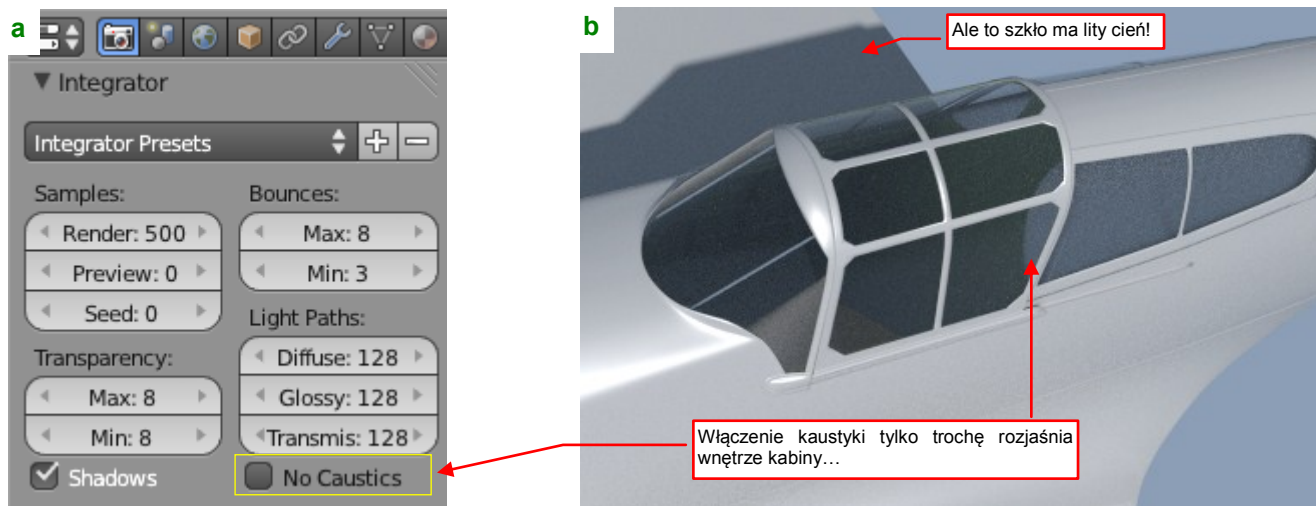
Rysunek 9.13.1 Domyślne ustawienia nowego materiału

Ustawiłem dla tej powierzchni kolor biały i idealną gładkość (*Roughness* = 0). Współczynnik refrakcji *IOR* jest odpowiedni dla plexiglasu. Jednak rezultat tych ustawień na podglądzie renderu jest daleki od oczekiwanego (Rysunek 9.13.2):



Rysunek 9.13.2 Rezultat pierwszego zastosowania shadera *Glass BSDF*

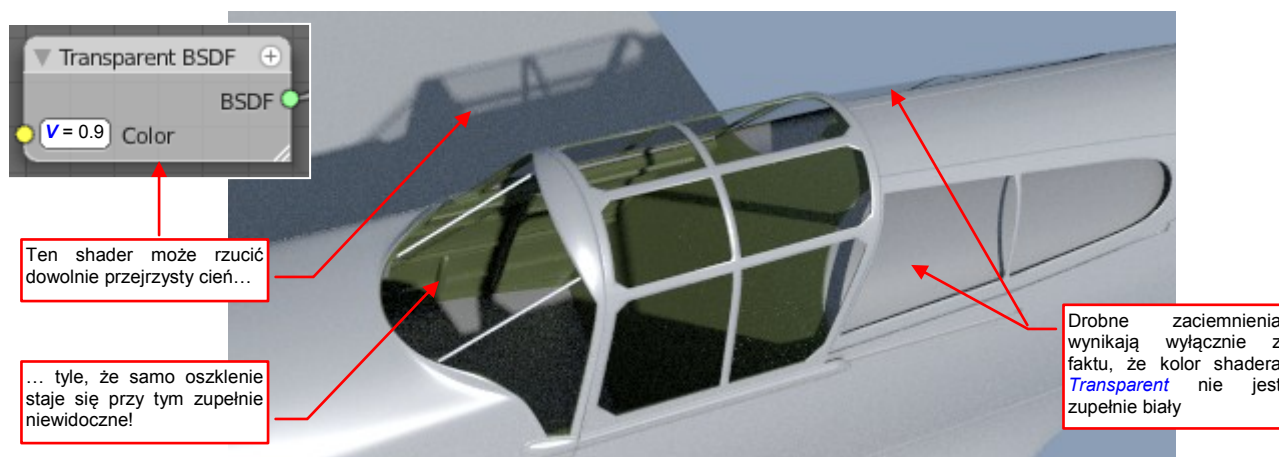
Dlaczego w środku kabiny wszystko jest niemal zupełnie czarne!? Składa się na to kilka przyczyn. Po pierwsze, miałem wyłączoną obsługę tzw. kaustykę (ang. *caustic*) czyli przenikania światła przez szkło. (Wyłączenie kaustyki daje mniej przypadkowych białych punktów na renderze nieprzezrzystych, połyskliwych powierzchni. W przypadku naszego modelu to znacznie poprawia wynik i skraca czas renderowania). Włączenie kaustyki (Rysunek 9.13.3a) tylko trochę rozjaśnia wnętrze kabiny (Rysunek 9.13.3b):



Rysunek 9.13.3 Shader *Glass BSDF*: problemy z cieniem!

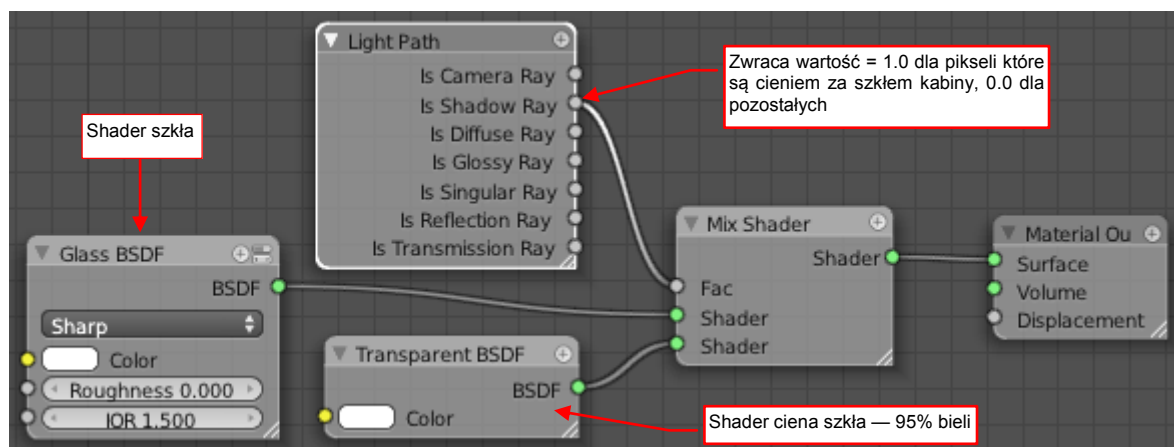
Właściwą przyczyną naszych problemów jest cień: ta szyba rzuca cień jak lity przedmiot! (Rysunek 9.13.3b).

W wersji Blendera, w której to wykonuję (2.61) Cycles nie obsługuje prawidłowo kaustyki dla źródła światła typu „słońce”, które stosujemy w tej scenie. Musimy to obejść pewną sztuczką. Otóż mamy do dyspozycji shader o nazwie **Transparent**. Kolor tego węzła pozwala płynnie regulować cień (Rysunek 9.13.4):



Rysunek 9.13.4 Shader **Transparent**: problemy z brakiem oszklenia

Powierzchnie **Transparent** nie zastąpią nam jednak szkła — jest to taki efekt „techniczny”, przeznaczony do łączenia z innymi shaderami. My podstawimy jego cień zamiast cienia szkła — używając znacznika **Light Path: Is Shadow Ray** (Rysunek 9.13.5). (Więcej na temat **Light Path** znajdziesz na str. 324).



Rysunek 9.13.5 Uzyskanie uproszczonych cieni za oszkleniem

Rysunek 9.13.6a) przedstawia rezultat, jaki otrzymaliśmy dla tak zmodyfikowanego materiału na renderze. Zwróć uwagę, że działa mimo włączonej opcji **No Caustics** (Rysunek 9.13.6b):



Rysunek 9.13.6 Efekt: przejrzysty cień, mimo wyłączonej kaustyki







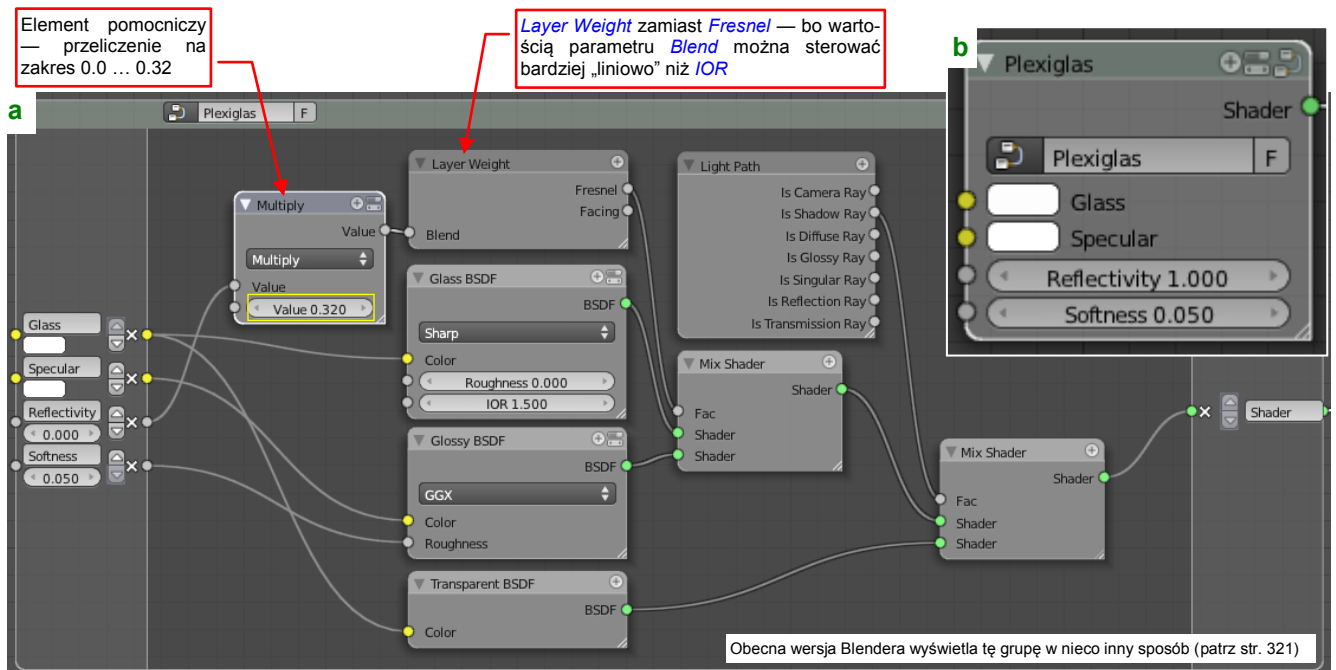
- Im gładzsza powierzchnia kadłuba (*Roughness* → 0.0), tym większa szansa wystąpienia na renderze owiewki kabiny niepożądanych białych punktów („świećlików”).

Najwięcej takich punktów tworzy się na szybach z tyłu kabiny, gdzie powierzchnia kadłuba znajduje się zaraz za powierzchnią szkła. Tak jak pisałem, to wyniki dla pierwszej produktywnej wersji Cycles (w Blenderze 2.61). Zobacz na koniec tej sekcji (str. 365), bo tam umieściłem informacje dotyczące nowszych wersji!

Na szczęście P-40B i C były zawsze pomalowane jakąś półmatową farbą, co zmniejsza ryzyko pojawienia się na renderze „świećlików”. Zdecydowałem umożliwić użytkownikowi materiału sterowanie przejrzystością szkła, czyli parametrem *IOR* z węzła *Fresnel*. (Zwróć uwagę, że chodzi tu wyłącznie o możliwość zmiany udziału shadera *Glossy BSDF*. Nie zmienia to w żadne sposób współczynnika *Glass BSDF:IOR*, który decyduje o załamaniu promieni słonecznych w szkło). Przekształciłem opracowany do tej pory zespół węzłów w grupę *Plexiglas* (Rysunek 9.13.10a), wprowadzając jednocześnie kilka drobnych udogodnień:

- zastąpiłem węzeł *Fresnel* węzłem *Layer Weight*, którego parametr *Blend* w zakresie 0.0 .. 0.4 przekłada się bardziej liniowo na zmianę współczynnika Fresnela niż *Fresnel:IOR*. Z obserwacji kilku testowych renderów wywnioskowałem, że *Fresnel:IOR* = 1.5 odpowiada *Layer Weight:Blend* = 0.32;
- Dodałem na wejściu do *Layer Weight* dodatkowy węzeł *Multiply* (*Convertor* → *Math*), który przelicza wprowadzaną przez użytkownika wartości parametru *Reflectivity* (intuicyjne 0.0 .. 1.0) na wartość odpowiednią dla *Layer Weight:Blend*;

Parametry nowego shadera przedstawia Rysunek 9.13.10b):



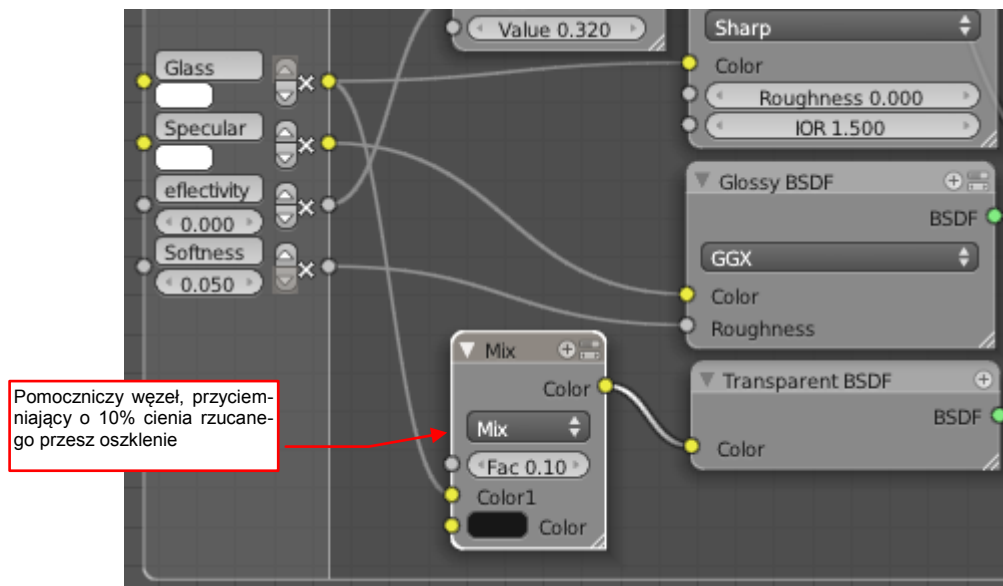
Rysunek 9.13.10 Definicja shadera *Plexiglas*

Zdecydowałem się „wyciągnąć” z tych węzłów następujące wejścia (por. Rysunek 9.13.10b):

- *Glass*: barwa szkła (i — jednocześnie — barwa przejrzystego cienia);
- *Specular*: barwa połysków (zazwyczaj biała).;
- *Reflectivity*: domyślna wartość 1.0<sup>1</sup> oznacza „zmatowienie” szkła (wynikające z udziału shadera *Glossy*) odpowiednie dla *IOR* = 1.5, a mniejsze wartości — szkło bardziej przejrzyste. Zmniejszenie tego parametru może się przydać w zbliżeniach i przy wzmocnionym świetle otoczenia;
- *Softness*: „miętkość” odbłyśków. To parametr *Glossy BSDF:Roughness*, pod zmienioną nazwą;

<sup>1</sup> Ograniczenia zakresu parametru do 0..1 uzyskałem „wyciągając” kontrolkę *Glass:Roughness*, a potem przełączając ją na inne wejście.

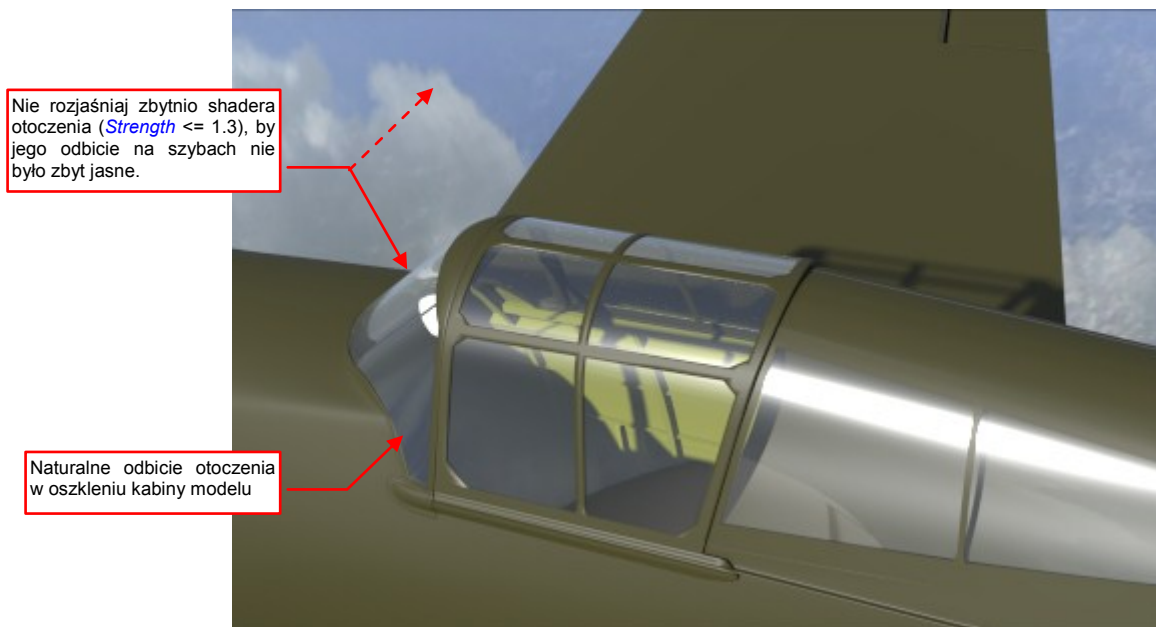
Aby dodatkowo nie komplikować schematu pokazywanego przez Rysunek 9.13.10, pokazuję oddzielnie wewnętrzne ulepszenie tej grupy. Przed wejściem do shadera *Transparent* dodałem węzeł typu *Color→Mix*. To operacja mieszania dwóch barw zgodnie z proporcją ustaloną przez współczynnik *Fac* (Rysunek 9.13.11):



Rysunek 9.13.11 Dodatkowy wewnętrzny element, przyciemniający cień rzucany przez oszklenie

W tym przypadku węzeł *Mix* służy do nieznacznego przyciemnienia barwy przezroczystych cieni. Miesza podaną na wejściu grupy barwę szkła (*Plexiglas:Glass*) z 10% czerni, i tak zmieniony kolor dostarcza do wejścia shadera *Transparent BSDF*. Oczywiście, równie dobrze możesz ustawić tu jaśniejsze cienie — np. 5%.

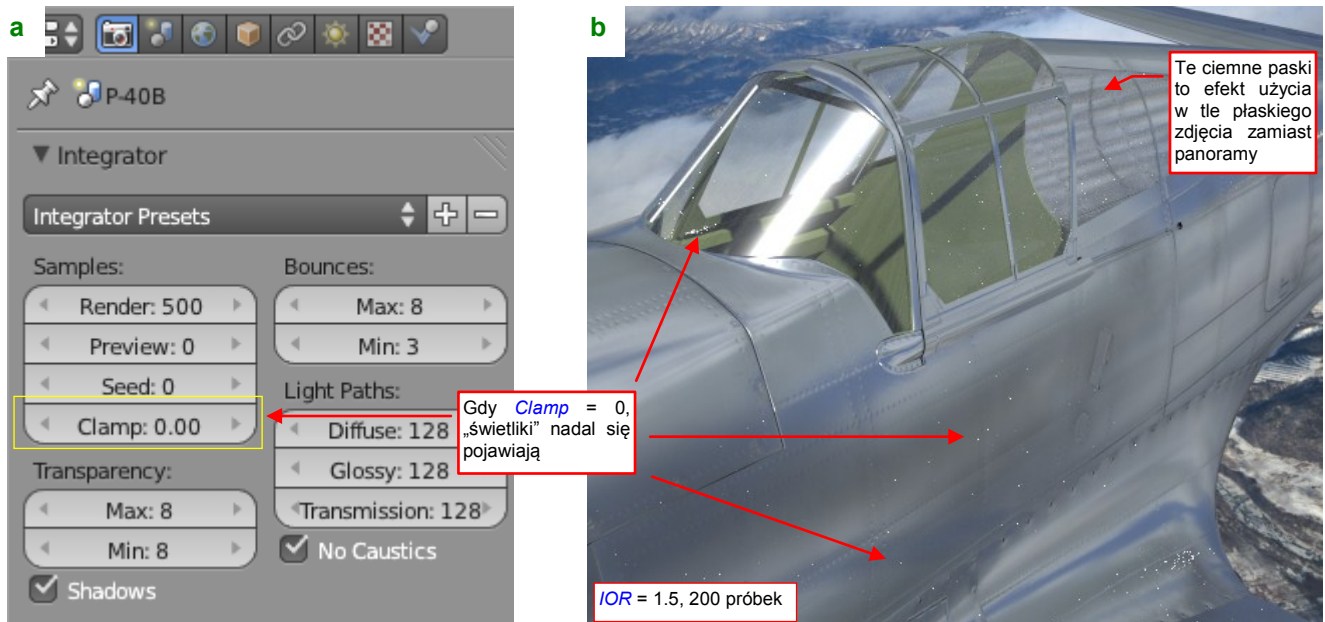
Rysunek 9.13.12 przedstawia rezultat uzyskany za pomocą shadera *Plexiglas*. Tym razem zamiast jednolitego tła użyłem panoramicznej tekstury otoczenia, z chmurami i niebem, które w naturalny sposób odbija się w owiewce:



Rysunek 9.13.12 Rezultat użycia shadera *Plexiglas* (*Reflectivity* = 100%)

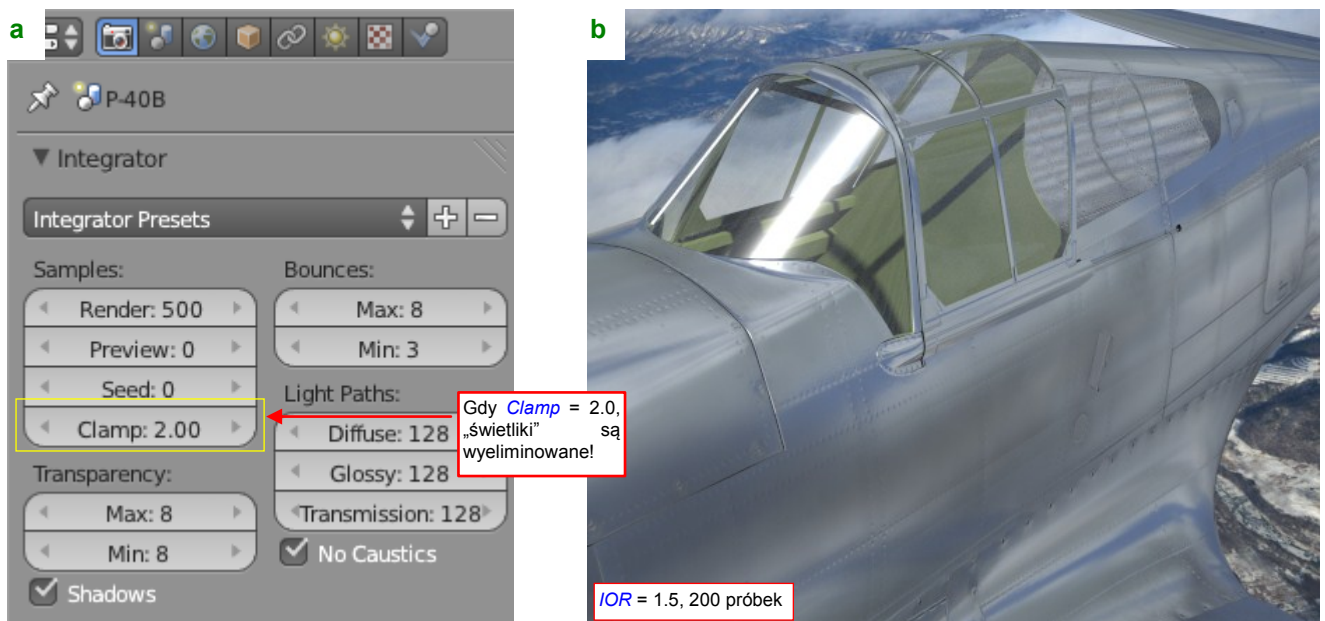
Zwróć uwagę, jak ważne jest uzyskanie jakiegoś zróżnicowanego odbicia otoczenia w szybach owiewki. Uważam, że dodają jej realizmu. Jednak bardzo trudno jest znaleźć odpowiednią panoramę, a jeszcze trudniej — panoramę o odpowiedniej rozdzielczości. Jeżeli w tle sceny podstawisz zamiast niej jakieś płaskie zdjęcie, zektniesz się z pewnymi zniekształceniami odbić w oszkleniu owiewki (por. str. 348).

Na koniec chciałbym wrócić do problemu „światełków”. W wersji 2.63 Blendera wprowadzono w Cycles pewne poprawki. Po pierwsze — zoptymalizowano nieco algorytm działania, co spowodowało, że białe punkty są jakby mniejsze. Po drugie — w panelu **Render: Integrator** pojawił się dodatkowy parametr, o nazwie **Clamp** (Rysunek 9.13.13a). To max. dopuszczalna jasność pikseli podczas obliczania kolejnych próbek renderu (wartość 1.0 oznacza biały). Domyślnie **Integrator: Clamp** jest ustawiony na wartość = 0, co oznacza że jest wyłączony. W takim przypadku białe punkty pojawiają się na renderze tak jak we wcześniejszych wersjach (Rysunek 9.13.13b):



Rysunek 9.13.13 Render bez ograniczeń na zakres jasności pikseli

Ustaw parametr **Clamp** na taką wartość, by nie „obcinał” białych pikseli, ale eliminował tylko takie, które są ewidentnie zbyt jasne (tzn. nieprawidłowe). Wypraktkowałem, że dla naszych scen wystarczy przyjąć **Clamp** = **2.0** (Rysunek 9.13.14a). Dla tej wartości światełki są wyeliminowane już po około 50 próbkach. Po 200 próbkach mamy już zupełnie porządną efekt (Rysunek 9.13.14b):



Rysunek 9.13.14 Render z ograniczeniem na zakres jasności pikseli (**Clamp**)

Jednocześnie scena nie ulega żadnemu widocznemu zaciemnieniu (tak się dzieje dla wartości **Clamp** < 1.0). Nie widać także innych negatywnych efektów ubocznych. Czyli — ustawienie godne polecenia!

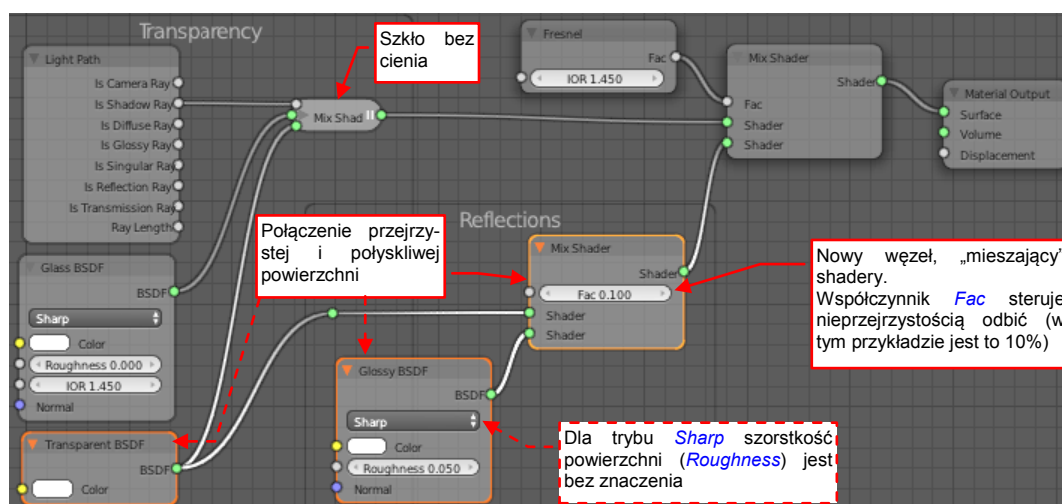


Podczas modelowania wnętrza kabiny natknąłem się na problem odbicia otoczenia przez szkło. Każdy z przyrządów pokładowych zamontowanych w tablicy przyrządów był zabezpieczony płaską szklaną płytką. Początkowo przypisałem im materiał użyty w osłonie kabiny (oparty o shader *Plexiglass*). Okazało się, że w ujęciach z boku tarcze wskaźników zmieniały się w małe lusterka (Rysunek 9.13.15):



Rysunek 9.13.15 „Lustrzane” odbicia otoczenia w osłonach przyrządów pokładowych

Taki efekt nie wygląda realistycznie. Przyczyną jest shader *Glossy BSDF*, którego udział rośnie wraz z kątem pochylenia szklanej powierzchni do obserwatora. Aby to skorygować, trzeba uczynić ten shader częściowo przezroczystym. „Zmieszałem” go więc z shaderem *Transparent BSDF* (Rysunek 9.13.16):



Rysunek 9.13.16 Schemat zmodyfikowanego materiału dla osłon przyrządów (B.Glass.Dials)

Rysunek 9.13.17 przedstawia to samo ujęcie, uzyskane z wykorzystaniem materiału o schemacie przedstawionym powyżej (nadałem mu nazwę **B.Glass.Dials**):



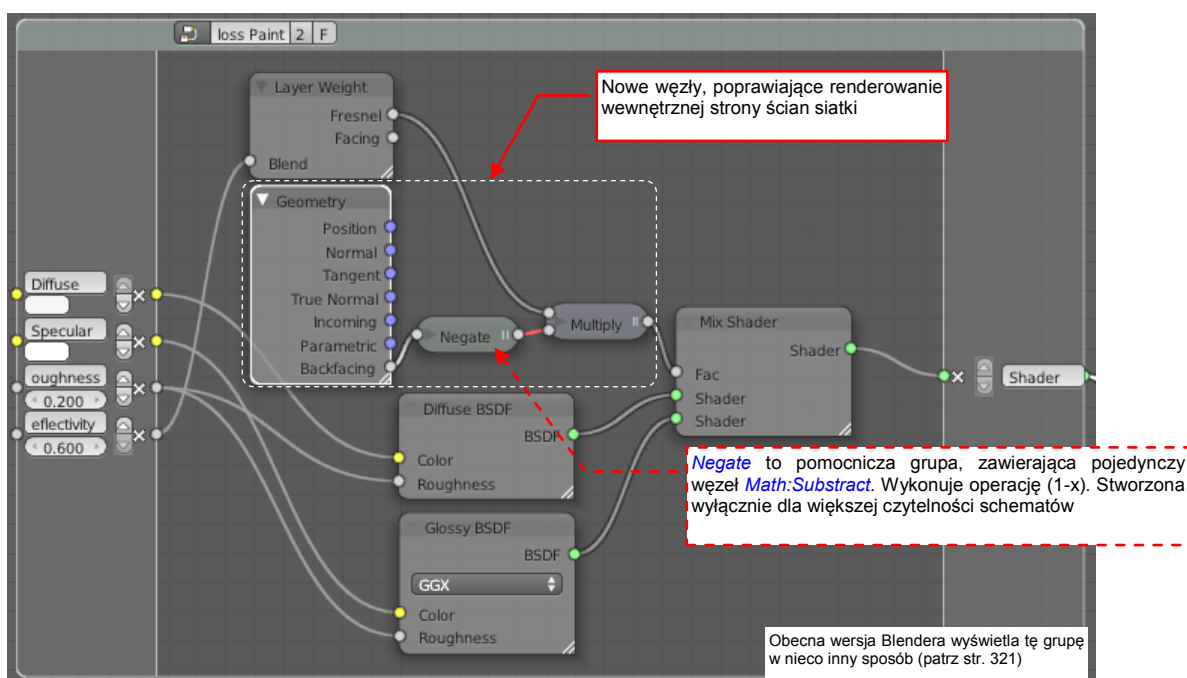
Rysunek 9.13.17 Odbicia otoczenia w materiale B.Glass.Dials





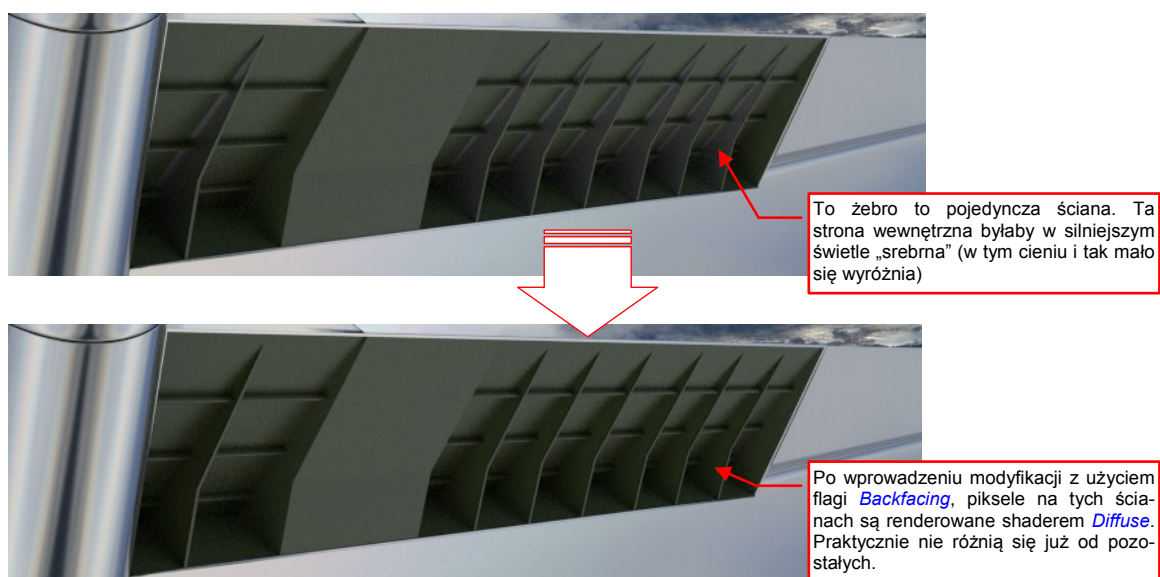
### 9.14 Skonfigurowanie materiału: dural i inne powierzchnie (*Gloss Paint*)

Jako podstawowy shader powierzchni samolotu stosuję grupę, której nadałem nazwę *Gloss Paint*. To klasyczne połączenie shaderów *Diffuse BSDF* i *Glossy BSDF* za pomocą współczynnika Fresnela. Stworzyliśmy je w Wyciągu III. Na początku tej sekcji wspomnę tylko o pewnym drobnym usprawnieniu, o które warto wzbogacić tę grupę. Rzecz polega na tym, że gdy tylko udział *Glossy BSDF* jest niezerowy (czyli gdy parametr *Reflectivity* jest  $> 0$ ), shader tworzył na wewnętrznych ścianach siatki odbicie otoczenia. Dla powierzchni matowych lub półmatowych ten efekt można zminimalizować, podstawiając na ścianach wewnętrznych sam rezultat samego shadera *Diffuse BSDF*. W Wyciągu III pokazałem to na schemacie, „mieszając” wynik z *Glossy Paint* z zewnętrznym shaderem, przy użyciu flagi *Geometry:Backfacing*. Ten sam rezultat można uzyskać wewnątrz samej grupy *Gloss Paint*, mnożąc współczynnik wagi (*Layer Weight:Fresnel*) przez 1 dla ścian zewnętrznych i 0 dla wewnętrznych (Rysunek 9.14.1):



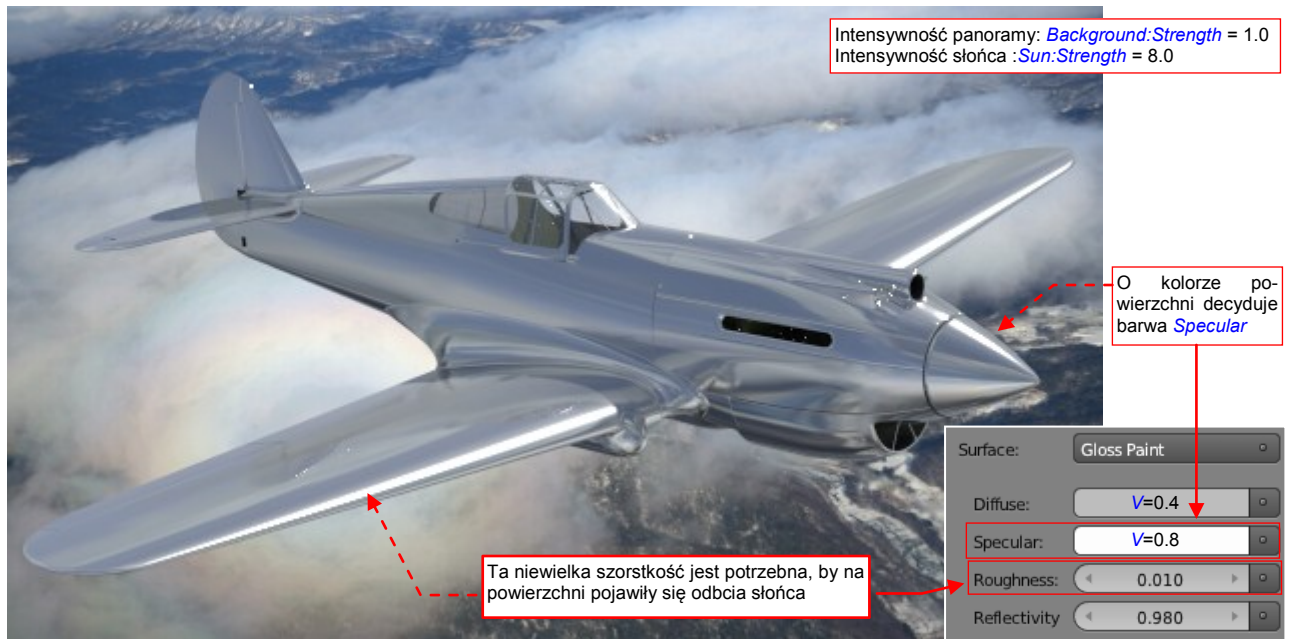
Rysunek 9.14.1 Modyfikacja shadera *Gloss Paint* — wyłączenie połyskliwości dla wewnętrznej strony ścian siatki

Ściany wewnętrzne są renderowane rzadko — w tym modelu tylko w żebrach skrzydła (Rysunek 9.14.2):



Rysunek 9.14.2 Shader *Gloss Paint*: działanie przed i po modyfikacji

Ten sam shader pozwala na odwzorowanie najróżniejszych rodzajów poszycia samolotu. Mimo nazwy, za pomocą *Glossy Paint* możesz uzyskać także efekt powierzchni utrzymanej w naturalnym kolorze metalu (Rysunek 9.14.3):



Rysunek 9.14.3 Powierzchnia jak dla wypolerowanej blachy Alcaid

(Dla takich połyskliwych powierzchni istotny jest także obraz otoczenia. Na tej i następnych ilustracjach wykorzystuję panoramę otoczenie *World* jak na schemacie ze str. 359, Rysunek 9.12.21). Tak lustrzaną powierzchnię mogły mieć samoloty z produkowanej w USA blachy Alcaid, odpowiednio wypolerowane przez obsługę naziemną. (Arkusze Alcaid były pokryte cienką warstwą czystego aluminium). Dla mniej zadbanego samolotu użyj nieco większej wartości parametru *Roughness* — od 0.02 do 0.05 (por. str. 373).

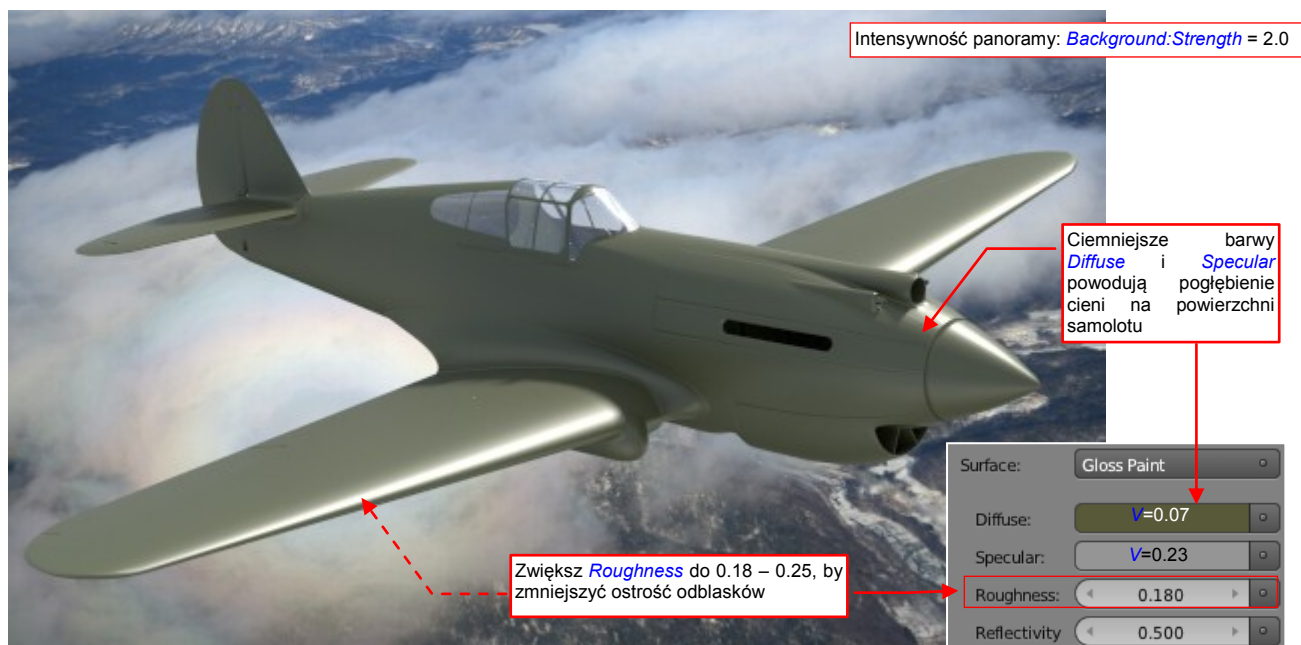
O ile dla bardzo połyskliwych (*Reflectivity* = 0.98) powierzchni o barwie decyduje niemal wyłącznie kolor *Specular*, to po zmniejszeniu połyskliwości do 0.5 — 0.6 możesz uzyskać efekt wyglądający jak błyszczący lakier lub jak połyskliwy plastik. Teraz ważny staje się także kolor *Diffuse* (Rysunek 9.14.4):



Rysunek 9.14.4 Powierzchnia jak dla połyskliwego lakieru

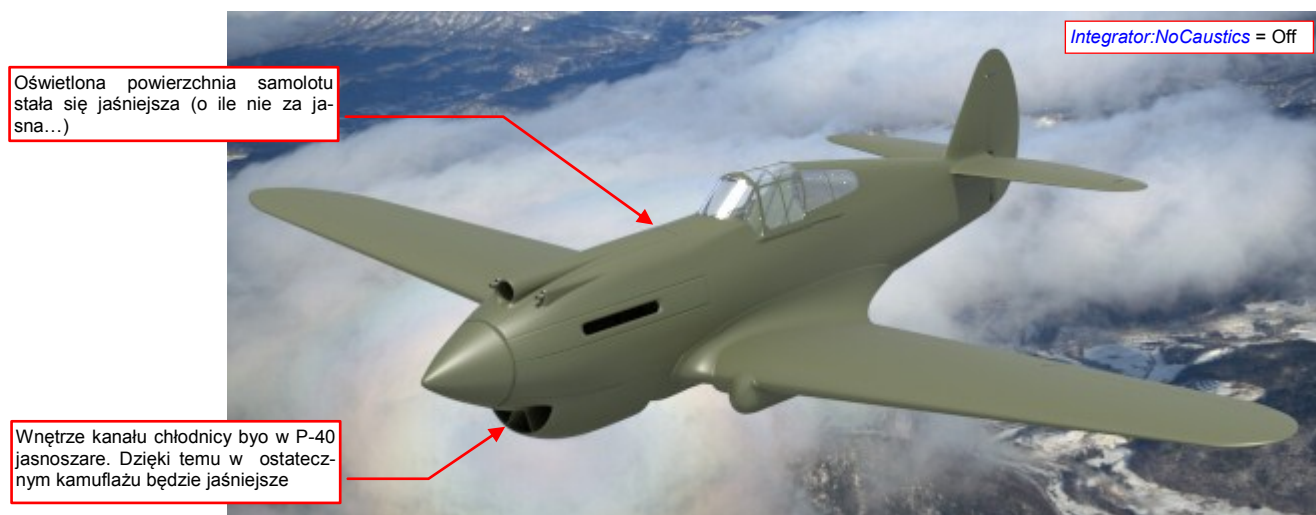


Samoloty bojowe były zazwyczaj malowane farbami półmatowymi lub matowymi, by zmniejszyć szansę na dostrzeżenie przez przeciwnika. Aby uzyskać taki efekt, zwiększ wartość **Roughness** do wartości rzędu 0.15 — 0.25. Jednocześnie zastosuj ciemniejszą barwę **Diffuse** (utrzymaj jej intensywność **V** pomiędzy 0.05 — 0.01). Zmniejsz także intensywność barwy **Specular** do poziomu 0.15 — 0.25 (Rysunek 9.14.5):



Rysunek 9.14.5 Półmatowy kamuflaż samolotu bojowego

Wszystkie te zmiany spowodowały pogłębienie cieni na powierzchni samolotu. Stały się już na tyle ciemne, że ujęcie, które pokazuje Rysunek 9.14.5, nie wygląda najlepiej. W tej scenie światło słońca pada na lewą stronę kadłuba. Dlatego po prawej stronie, w cieniu, szczegóły jego powierzchni są mniej wyraźne. (Już nie rozjaśnia ich tak bardzo odbite światło otoczenia — bo zmniejszyliśmy intensywność **Specular**). Oczywiście, ten sam materiał wygląda inaczej po stronie oświetlonej (Rysunek 9.14.6):

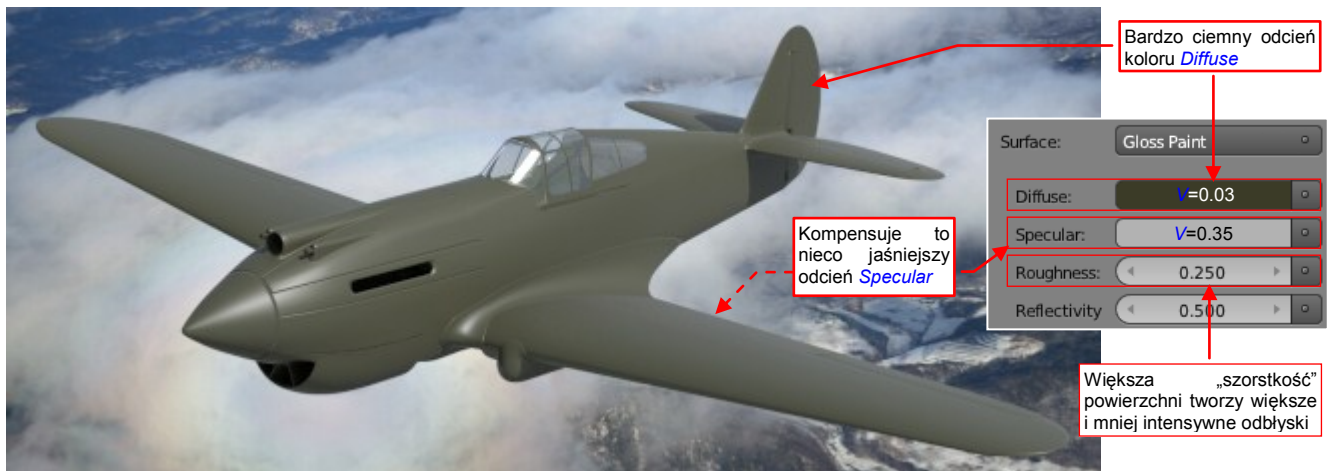


Rysunek 9.14.6 Ten sam materiał od strony oświetlonej światłem słońca

Dla takich półmatowych powierzchni modelu można przywrócić obliczenia kaustyki (wyłączyć opcję **No Caustics** w panelu **Integrator** zestawu **Render**). To rozjaśni nieco wnętrze kabiny pilota, a dla tak „chropowatych” powierzchni Cycles nie będzie już tworzył masy „światlików”, jak robił na str. 358, dla połyskliwej powierzchni (Rysunek 9.12.18). W zasadzie kamuflaż po oświetlonej stronie modelu wydaje się nawet trochę zbyt jasny. Tylko wloty do kanałów chłodnicy cieczy są bardzo ciemne. Później przypiszemy im właściwy kolor (były jasnoszare), w wyniku czego przestaną być tak czarne.



Często kamuflaż samolotów był znacznie ciemniejszy, niż ten na poprzednich ilustracjach — chociażby taki „fabrycznie świeży” kolor **Olive Drab** (Rysunek 9.14.7):



Rysunek 9.14.7 Samolot w kolorze zbliżonym do „fabrycznie nowego” Olive Drab

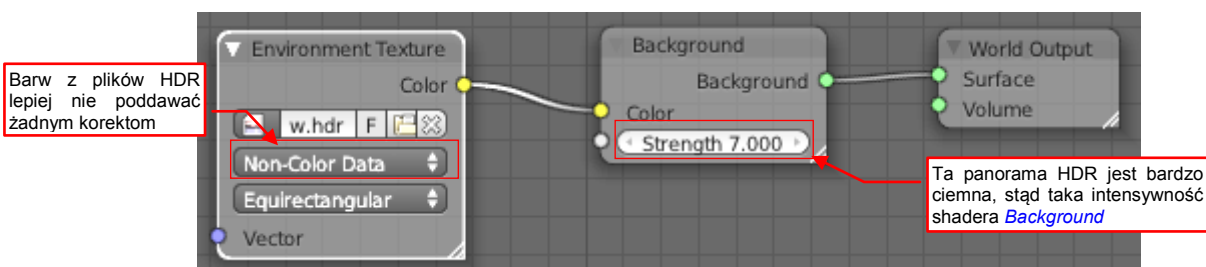
- Barwa **Diffuse** określa podstawowy kolor powierzchni, a **Specular** — udział barwy odbitego otoczenia (w scenach w locie — głównie nieba). Dlatego zazwyczaj **Specular** to jakiś odcień neutralnej szarości.

Sądzę, że po przedstawieniu tych kilku typowych sytuacji będziesz już umiał dobierać właściwe barwy **Diffuse** i **Specular**. Wydaje mi się za to, że warto jeszcze pokazać na nieco innym przykładzie, jak „działają” dwa pozostałe parametry: **Reflectivity** i **Roughness**. Myślę, że chyba nikt z nas nie oglądał zbyt często odbić otoczenia na powierzchni samolotu w locie. Abyś mógł lepiej ocenić efekt testu, zdecydowałem się umieścić nasz model „na ziemi”, w środowisku, którego odbicia znane są nam na co dzień (Rysunek 9.14.8):



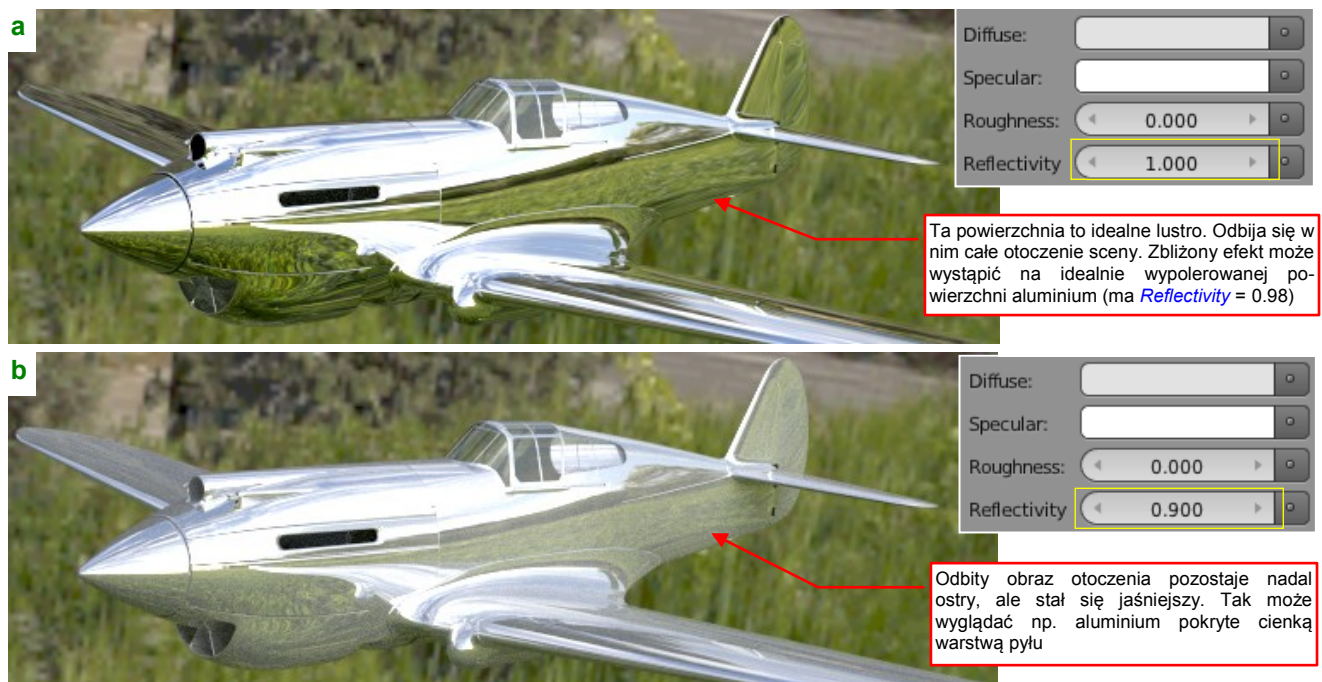
Rysunek 9.14.8 Obraz otoczenia, który wykorzystamy dalej jako środowisko testowe ([www.openfootage.net](http://www.openfootage.net))

Do dalszych testów, aby wykluczyć wpływ cieni i odbłasków słońca, zastosujemy uproszczony model oświetlenia: wyłącznie panorama otoczenia (Rysunek 9.14.9). Zmieniłem także kolor **Diffuse** powierzchni naszego modelu na jasnoszary ( $V=0.8$ ), a **Specular** — na biały ( $V = 1.0$ ).

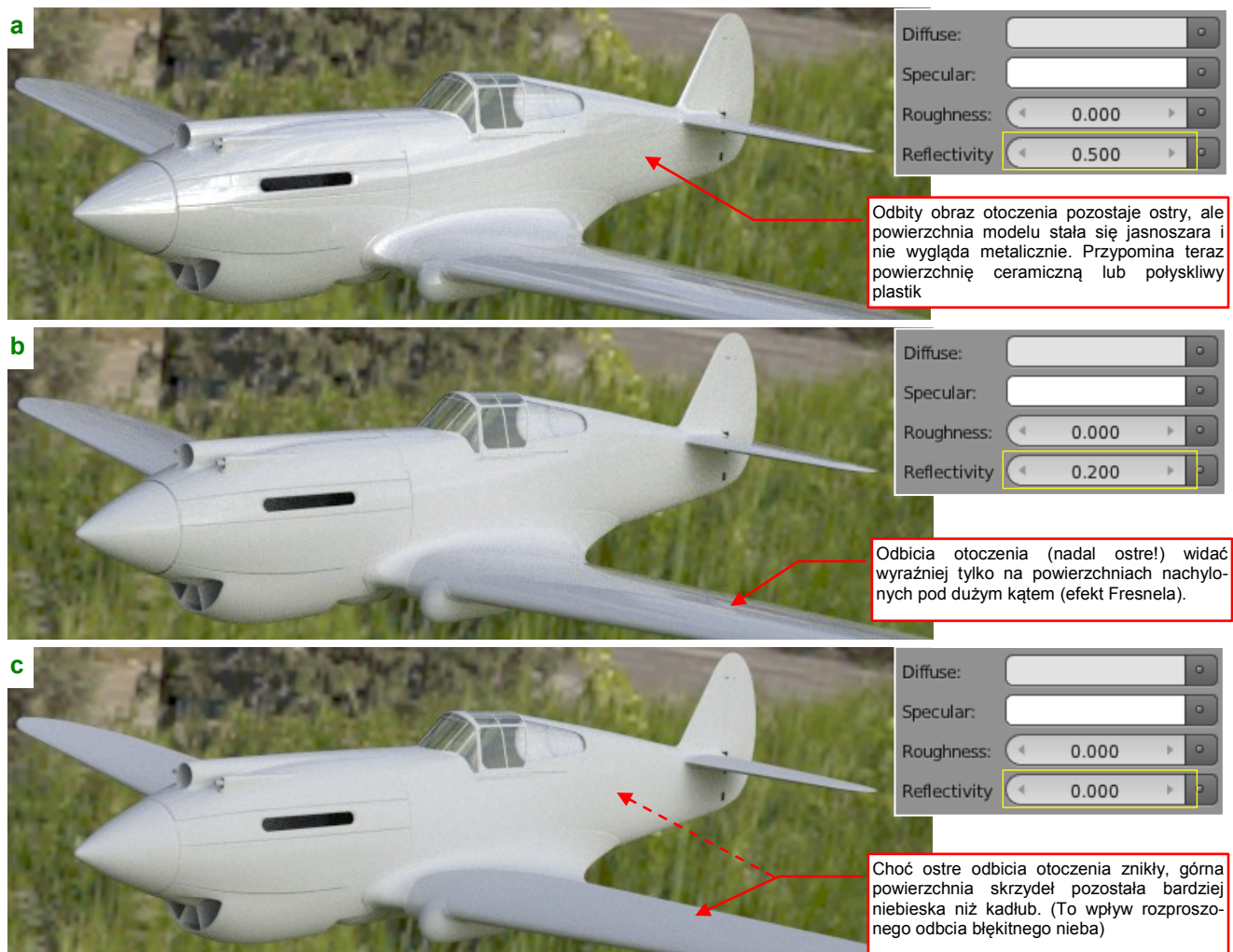


Rysunek 9.14.9 Schemat otoczenia, które wykorzystamy do testów

Wpływ zmian współczynnika **Reflectivity** przedstawiają: Rysunek 9.14.10 (dla powierzchni o wysokim połysku — „srebrnych”) i Rysunek 9.14.11 (dla pozostałych rodzajów powierzchni):



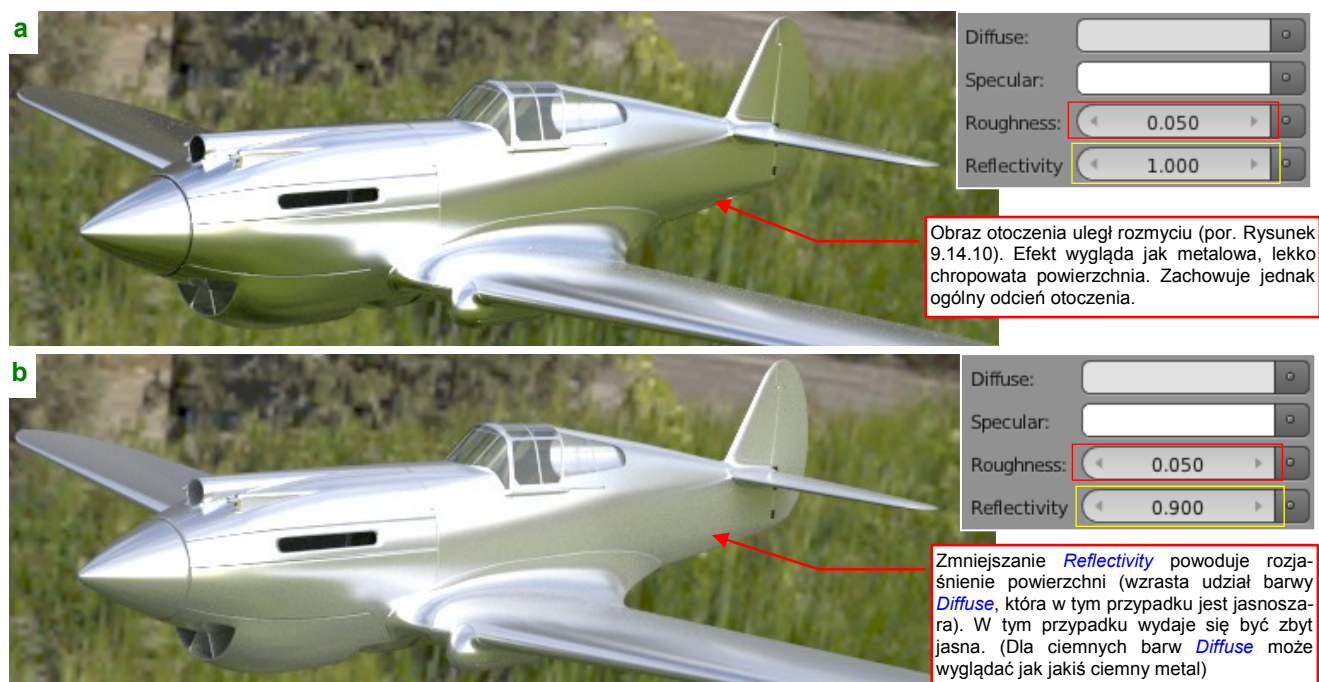
Rysunek 9.14.10 Efekt dla wysokich wartości **Reflectivity**



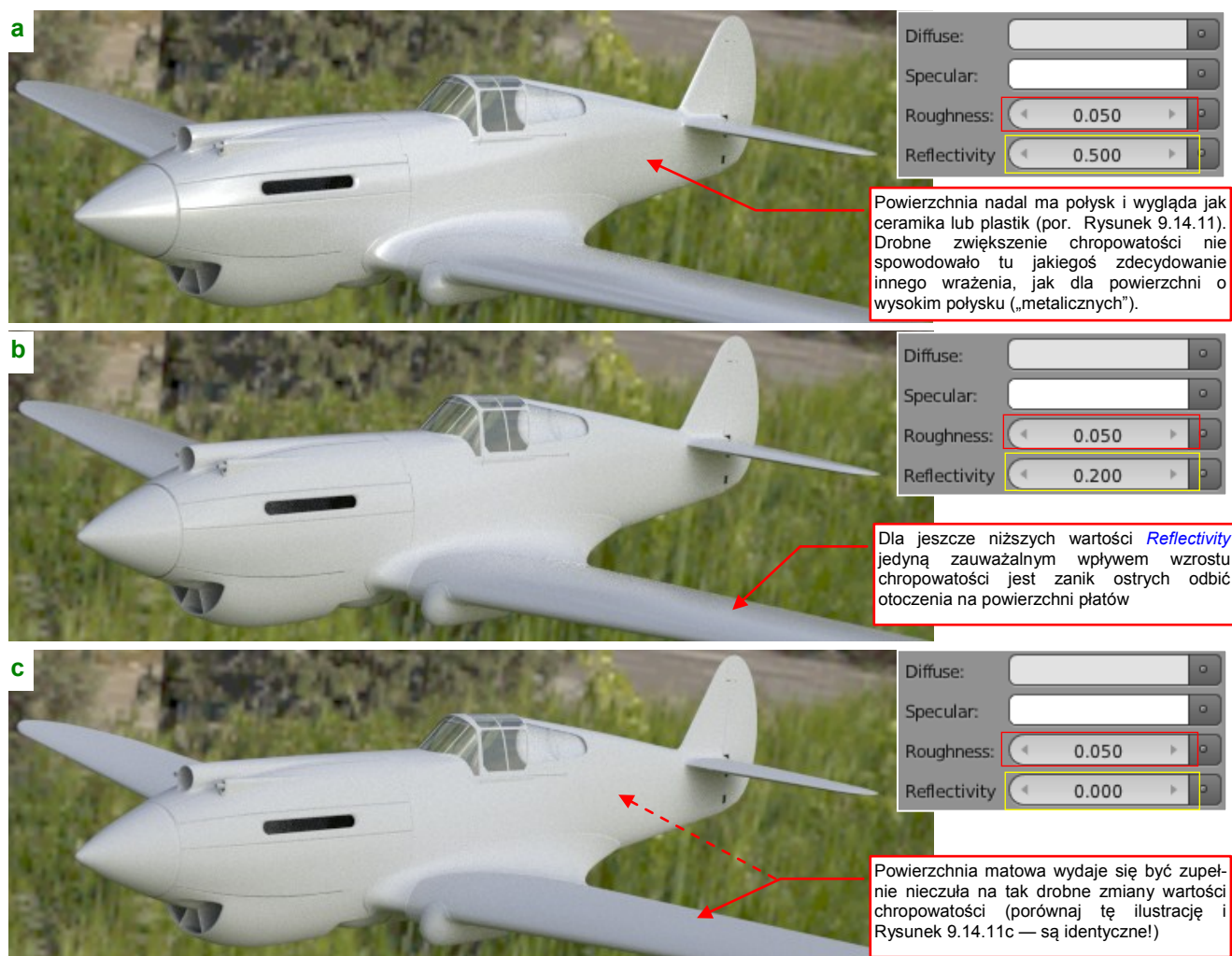
Rysunek 9.14.11 Efekt dla niższych wartości **Reflectivity**



Nawet niewielkie modyfikacje wartości **Roughness** (w tym przykładzie — o 0.05) powodują znaczące zmiany dla powierzchni o wysokim połysku (Rysunek 9.14.12). Dla pozostałych są nieznaczne (Rysunek 9.14.13):

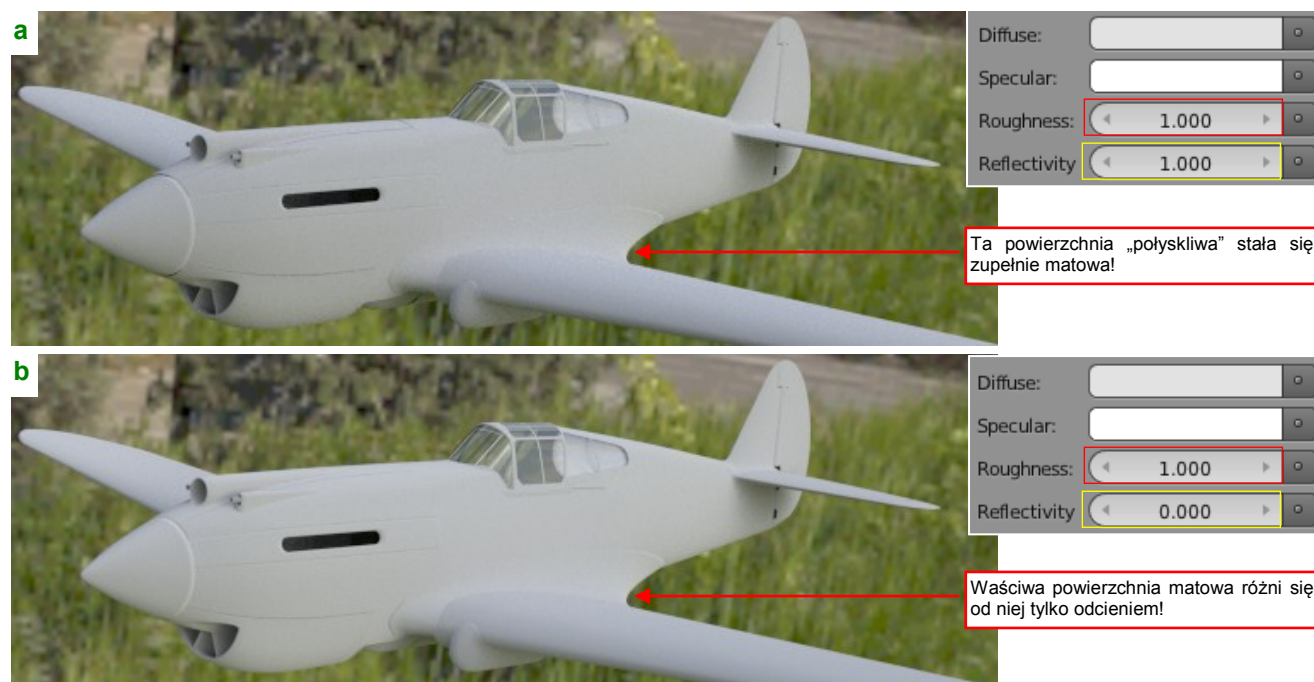


Rysunek 9.14.12 Efekt dla wysokich wartości **Reflectivity**



Rysunek 9.14.13 Efekt dla niższych wartości **Reflectivity**

W miarę zwiększania wartości **Roughness** powierzchnie „połyskliwe” (**Reflectivity** = 1) i matowe (**Reflectivity** = 0) stają się do siebie coraz bardziej podobne. Rysunek 9.14.14 przedstawia sytuację, gdy dla **Roughness** = 1.0 każda powierzchnia stała się zupełnie matowa:



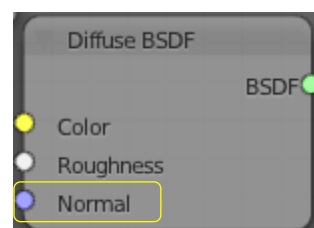
Rysunek 9.14.14 Wpływ wysokiej wartości parametru **Roughness** na powierzchnię połyskliwą (a) i matową (b)

Gdyby na takiej powierzchni umieścić jakiś czarny obszar, wyglądałby jak plama sadzy. Ten silny wpływ chropowatości na materiał „z połyskiem” wykorzystamy w następnym rozdziale do uzyskania efektów brudu i zakużenia.

- Wartości **Reflectivity** i **Roughness** powierzchni modelu można lokalnie modulować za pomocą tekstur, np. w celu uzyskania efektu zużycia lub zabrudzeń

Nie staraj się uzyskać zbyt połyskliwego kamuflażu. W Internecie możesz znaleźć dużo zdjęć różnych rekonstrukcji P-40. Choć przy ich malowaniu zazwyczaj starano się dobrać właściwy odcień farby, to użyty lakier jest z reguły połyskliwy. Ciekawe, czy to efekt zamierzony? (Taka maszyna wygląda ładniej na pokazach...). W każdym razie przy doborze kolorystyki lepiej korzystać ze zdjęć archiwalnych. W sieci można znaleźć trochę kolorowych zdjęć z tego okresu. Pamiętaj tylko, że wraz z upływem lat te fotografie zrobiły się bardziej niebieskie. Trzeba na to wziąć poprawkę!

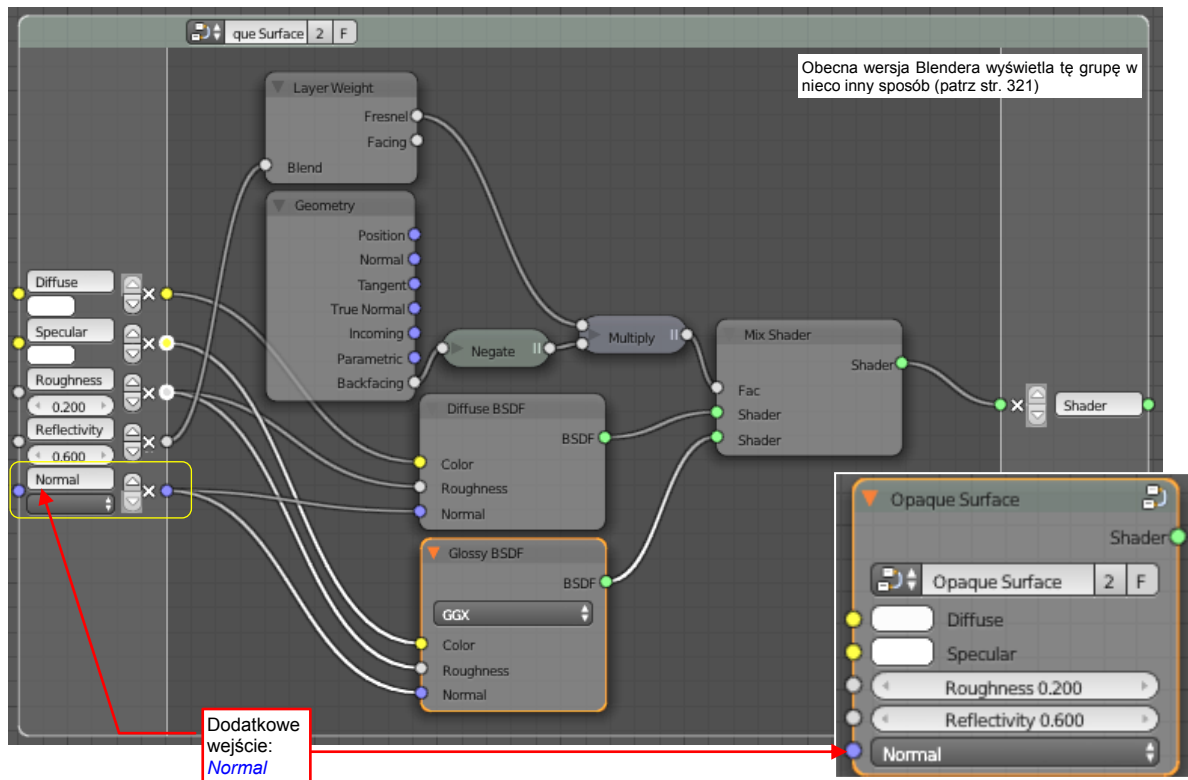
W Blenderze 2.65 w węzłach shaderów BSDF pojawiło się nowe wejście: **Normal**. Jest ono przeznaczone do podłączenia alternatywnej mapy nierówności. Do tej pory nierówności powierzchni samolotu można było odwzorować podłączając do „globalnego” wejścia **Displacement** węzła **Material Output** mapę odcieni szarości. Tworzy to złudzenie drobnych zmian „wysokości” powierzchni. Wejście **Normal** pozwala uzyskać jeszcze silniejszy efekt, gdyż możesz podać na wejściu współrzędne wektorów prostopadłych do powierzchni. (Zazwyczaj jest to specjalnie przygotowana tekstura, której wartość **R**, **G**, **B** barwy pikseli są interpretowane jako współrzędne **X**, **Y**, **Z** wektora normalnego. Często tworzy się ją operacją **Bake** z odpowiednio przygotowanego, bardzo szczegółowego modelu). W ten sposób można dobrze odwzorować, nawet w zbliżeniach, np. nierówności odlewanych pancerzy czołgów czy spawy na krawędziach ich płyt.



Rysunek 9.14.15 Wejście **Normal** shadera

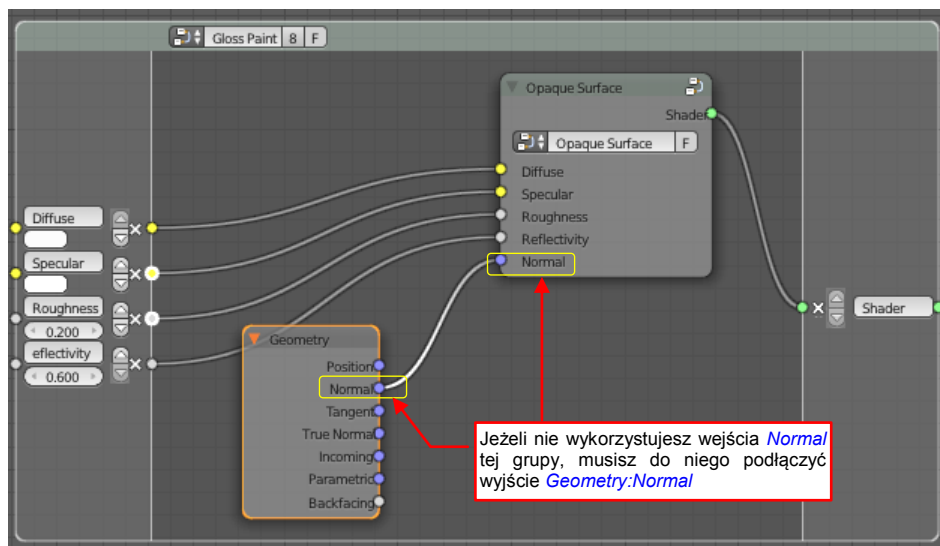


Abyś mógł użyć wejścia **Normal**, stworzyłem kopię **Gloss Paint** wzbogaconą o ten element (Rysunek 9.14.16):



Rysunek 9.14.16 Zmodyfikowana wersja shadera (**Opaque Surface**)

Nazwałem tę grupę **Opaque Surface**. Stosuj ją wszędzie tam, gdzie masz jakąś mapę normalnych do podłączenia. Pamiętaj tylko, aby zawsze do wejścia **Normal** było coś podłączone — odpowiednikiem informacji domyślnej jest **Geometry:Normal** (Rysunek 9.14.17):



Rysunek 9.14.17 Zmodyfikowany shader **Gloss Paint**

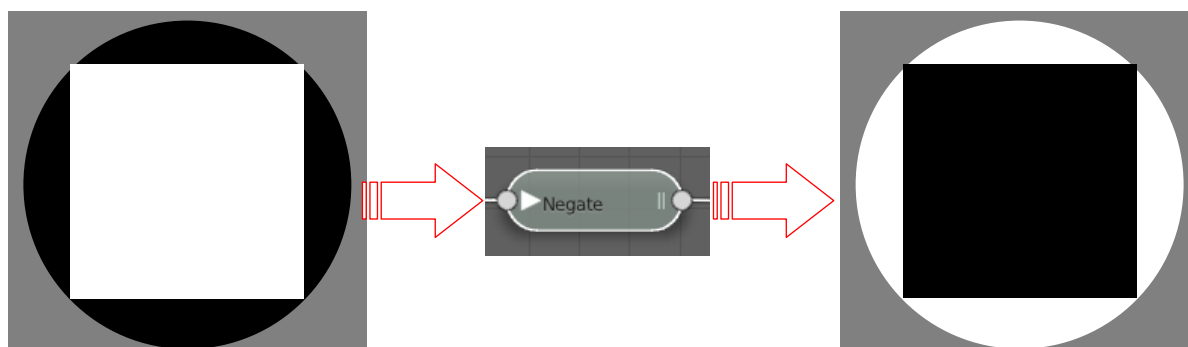
Taka konieczność podłączania kolejnego węzła to zawsze dodatkowy element komplikujący schemat. Dlatego shader **Gloss Paint** pozostał, abyś mógł go używać wszędzie tam, gdzie nie potrzebujemy map normalnych (a tak jest w większości materiałów). We wnętrzu tej grupy umieściłem teraz shader **Opaque Surface** z podłączonymi domyślnymi kierunkami normalnych (Rysunek 9.14.17).

### 9.15 Węzły pomocnicze

Wartości, którymi operujemy na schematach Cycles, są zazwyczaj w przedziale od 0.0 do 1.0. Na przykład: liczbami z tego przedziału są intensywności poszczególnych komponentów barwy (**RGBA** — por. str. 468). Także współrzędne tekstur (dane typu **Vector**) to wartości od 0 do 1. W zasadzie autorzy Cycles starali się zbudować podstawowe węzły tak, by każdy z ich parametrów miał zakres od 0.0 do 1.0. Nawet funkcja Fresnela, normalnie zależąca od **IOR** materiału, została w ten sposób sparametryzowana w węźle **Layer Weight**.

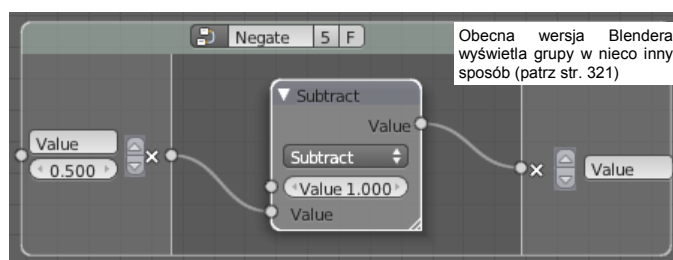
W tej sekcji przedstawię trochę pomocniczych grup węzłów, obsługujących różne powtarzające się na schematach operacje arytmetyczne na liczbach rzeczywistych od 0.0 do 1.0. Ich działanie zademonstruję, przekształcając monochromatyczne obrazy testowe<sup>1</sup>.

Zacznijmy od najprostszej operacji: **Negate**. Zwraca negatyw podłączonego obrazu (Rysunek 9.15.1):



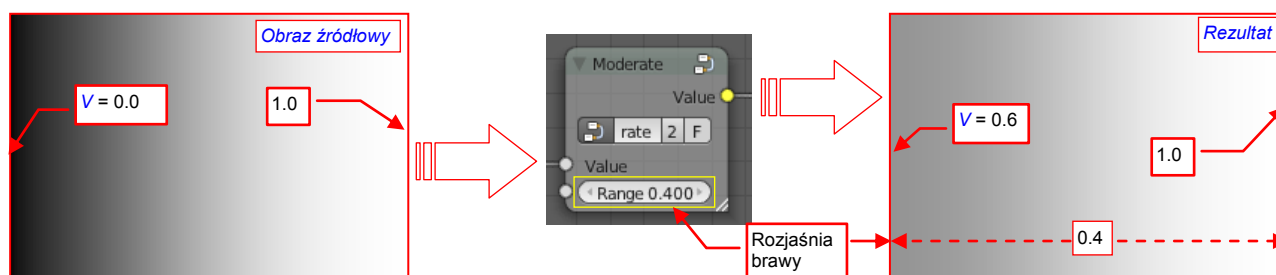
Rysunek 9.15.1 Działanie węzła **Negate**

Wewnętrznie grupa **Negate** zawiera pojedynczy węzeł **Math**, realizujący operację  $(1.0 - \text{Value})$  (Rysunek 9.15.2). Ta grupa istnieje wyłącznie po to, by uprościć schematy materiałów. Uważam, że poprawia ich czytelność. Po prostu nazwa **Negate** od razu pomaga zrozumieć, o co autorowi chodziło.



Rysunek 9.15.2 Budowa węzła **Negate**

Węzła **Moderate** używam do „łagodzenia” działania tekstur nierówności (Rysunek 9.15.3):

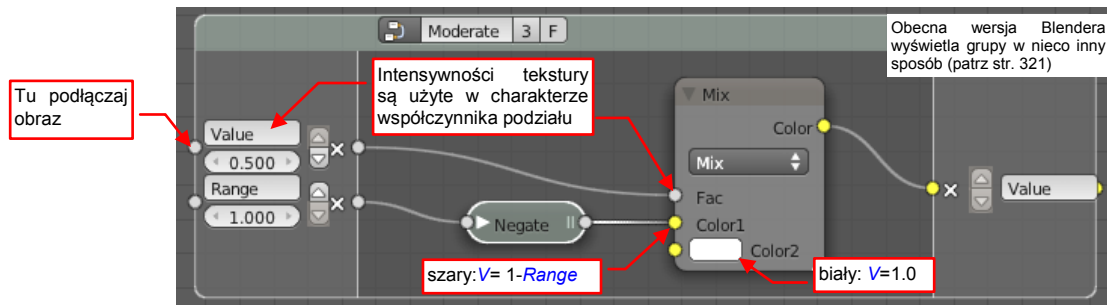


Rysunek 9.15.3 Działanie węzła **Moderate**

Gdy **Moderate:Range** = 1.0, węzeł zwraca bez zmian obraz wejściowy. Zmniejszenie wartości **Range** powoduje rozjaśnienie ciemniejszych miejsc obrazu.

<sup>1</sup> Większość obrazów tekstur w naszym modelu jest monochromatyczna (wykorzystuje odcienie szarości). W takim obrazie każdy piksel także może być traktowany jako liczba od 0.0 (czarny) do 1.0 (biały). Dlatego najczęściej opisanych w tej sekcji węzłów używam do adaptacji lub sterowania intensywnością tekstur.

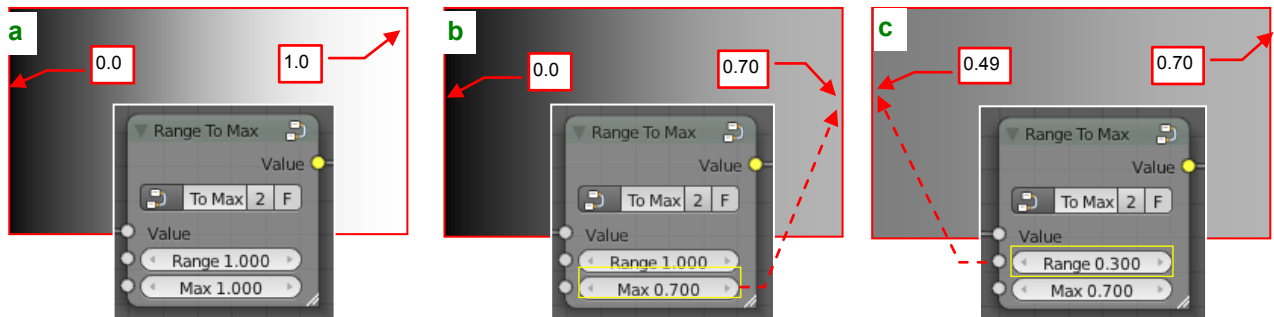
Wewnętrznie grupa **Moderate** wykorzystuje węzeł **Color:Mix** (Rysunek 9.15.4). Ten węzeł „miesza” otrzymane wartości tak, jak gdyby były kolorami (por. str. 468).



Rysunek 9.15.4 Budowa węzła **Moderate**

Pole **Value**, do którego podłączamy obraz tekstury, jest traktowane jako współczynnik udziału (**Mix:Fac**) barw **Color2** i **Color1**<sup>1</sup>. Dla czarnych pikseli tekstury (**Value** = 0) węzeł zwraca piksel o odcieniu szarości podstawowym jako **Color1**, a dla białych (**Value** = 1.0) — biały **Color2**. Jeżeli podstawimy pod **Mix:Color1** czerń — otrzymamy na wyjściu obraz wejściowy, bez żadnych zmian. Jeżeli jednak rozjaśnimy tę barwę — otrzymamy na wyjściu obraz w której oryginalnej czerni będzie odpowiadał szary z **Mix:Color1**. W rezultacie zakres intensywności występujących w obrazie ulegnie „spłaszczeniu”, zachowując bez zmian białe piksele. Aby lepiej uświadomić użytkownikowi, czym operuje, obliczam odcień szarości dla **Mix:Color1** jako „1 – **Range**” (za pomocą węzła **Negate**). Dzięki temu wartość parametru **Range** określa zakres intensywności w obrazie wyjściowym (**Range** = 1.0 podstawia pod **Mix:Color1** czerń, a **Range** = 0.4 — szary, o  $V = 0.6$ ).

O ile **Moderate** pozwalał tylko na rozjaśnienie obrazu, to jego rozbudowana wersja — węzeł **Range To Max** — pozwala także na zmniejszenie jasności obrazu. Rysunek 9.15.5 przedstawia trzy różne ustawienia tego węzła na tle uzyskiwanych rezultatów:



Rysunek 9.15.5 Działanie węzła **Range To Max**

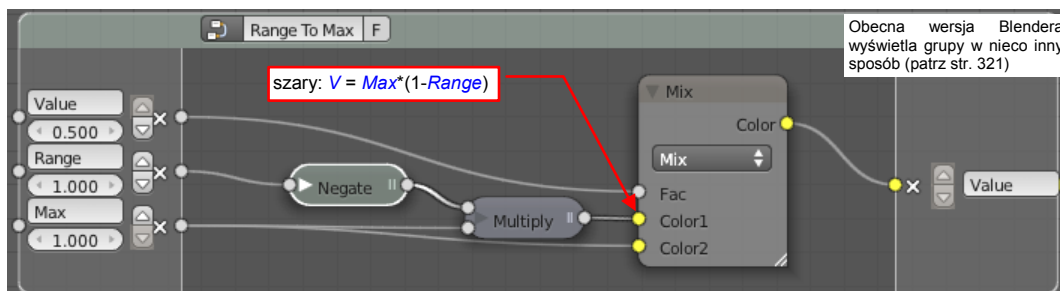
Rysunek 9.15.5a) to ustawienie neutralne — dla wartości **Range** = 1 i **Max** = 1 węzeł **Range To Max** zwraca bez żadnych zmian obraz wejściowy. Parametr **Max** określa jasność odpowiadającą białym pikselom obrazu wejściowego. Jeżeli ulegnie zmniejszeniu — rezultat będzie odpowiednio przyciemniony (Rysunek 9.15.5b).

Do rozjaśniania najciemniejszych pikseli służy, tak jak w węźle **Moderate**, parametr **Range**. Jednak tutaj jego wartość określa względny % spektrum od 0 do **Max**. Odcień szarości **Mix:Color1** jest tu określany ze wzoru:  $\text{Max} * (1 - \text{Range})$ . Dlatego najciemniejszy odcień na obrazie przedstawionym przez Rysunek 9.15.5c) ma intensywność  $V = 0.7 * (1 - 0.3) = 0.49$ . Może to nie jest zupełnie intuicyjne, ale bezpieczne<sup>2</sup>.

<sup>1</sup> Operacja **Color:Mix** to tzw. „średnia ważona”:  $\text{Fac} * \text{Color2} + (1 - \text{Fac}) * \text{Color1}$ .

<sup>2</sup> W ten sposób węzeł jest „odporny” na ewentualne błędy użytkownika. Nie wystarczy, że same pola wejściowe akceptują tylko wartości z zakresu od 0 do 1. Nawet przy tym ograniczeniu dla np. **Max** = 0.2 wpisanie „absolutnej” wartości **Range** = 0.4 dałoby min. odcień szarości  $0.2 - 0.4 = -0.2$ , czyli ujemny! Dlatego wolałem przejść na wartości względne, dla których najmniejszy odcień szarości to  $0.2 * (1 - 0.4) = 0.2 * 0.6 = 0.12$ . Pisząc o użytkowniku, mam także na myśli siebie — za miesiąc na pewno nie będę pamiętał o jakichś ograniczeniach w użyciu tego węzła!

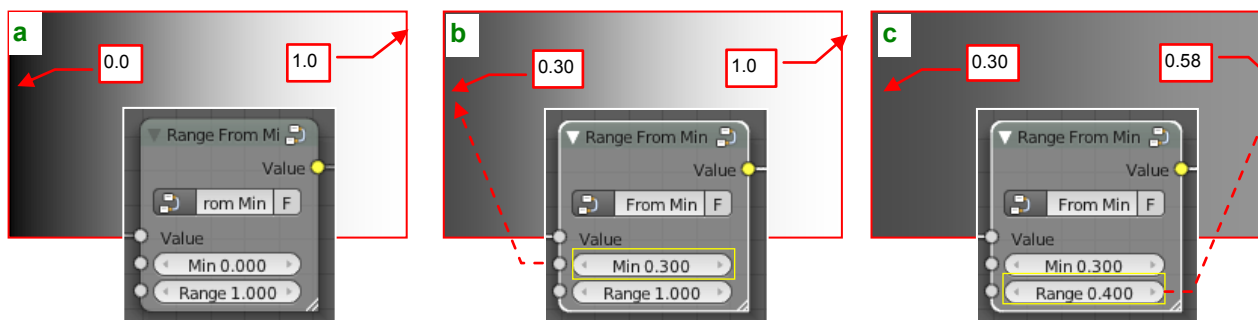
Rysunek 9.15.6 przedstawia wewnętrzną strukturę węzła *Range To Max*:



Rysunek 9.15.6 Budowa węzła *Range To Max*

Węzeł *Range To Max* najwygodniej używa się z obrazami o białym tle, które służą do lokalnego zmniejszania jakiejś wartości bazowej. (Ustawiasz wówczas tę wartość bazową jako *Max*, a parametrem *Range* regulujesz intensywność lokalnych zmniejszeń).

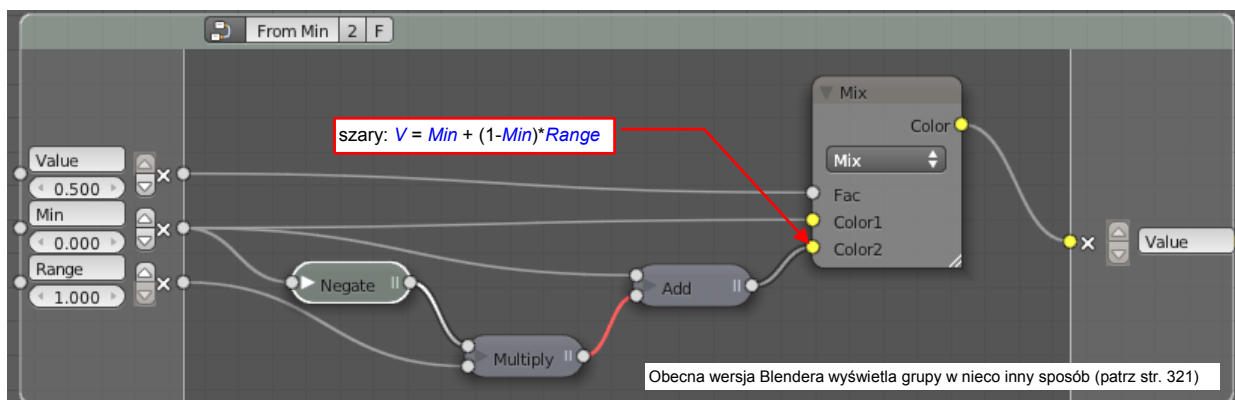
Aby wygodnie obsłużyć sytuację odwrotną: obrazu o czarnym tle, który służy do lokalnego zwiększenia jakiejś wartości, przygotowałem węzeł *Range From Min* (Rysunek 9.15.7):



Rysunek 9.15.7 Działanie węzła *Range From Min*

Rysunek 9.15.7a) przedstawia sytuację neutralną — dla wartości *Min* = 0.0 i *Range* = 1.0 węzeł *Range From Min* zwraca bez żadnych zmian obraz wejściowy. Parametr *Min* określa jasność odpowiadającą czarnym pikselom obrazu wejściowego. Jeżeli ulegnie zwiększeniu — rezultat będzie odpowiednio rozjaśniony (Rysunek 9.15.7b). Parametr *Range* tu także określa % względnego spektrum od *Min* do 1.0 ( $Mix:Color2 = Min + (1-Min)*Range$ ). Dlatego dla parametrów jakie pokazuje Rysunek 9.15.7c) białym polom obrazu wejściowego odpowiada kolor szary o  $V = 0.3 + (1-0.3)*0.4 = 0.58$ .

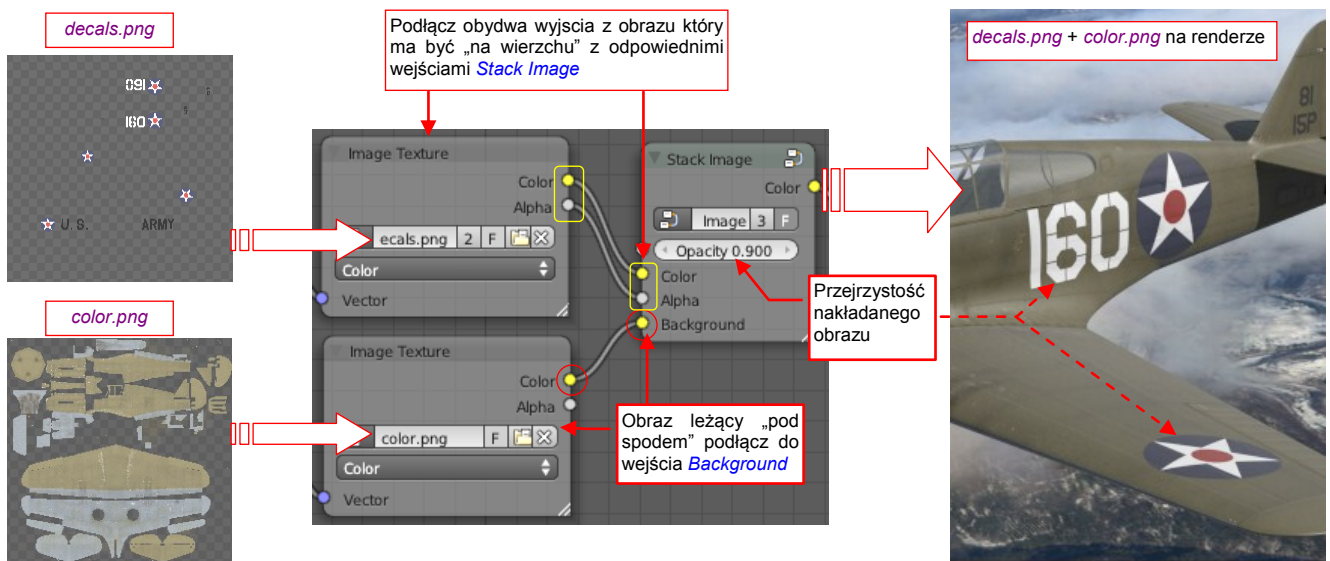
Rysunek 9.15.8 przedstawia wewnętrzną strukturę grupy *Range From Min*:



Rysunek 9.15.8 Budowa węzła *Range From Min*



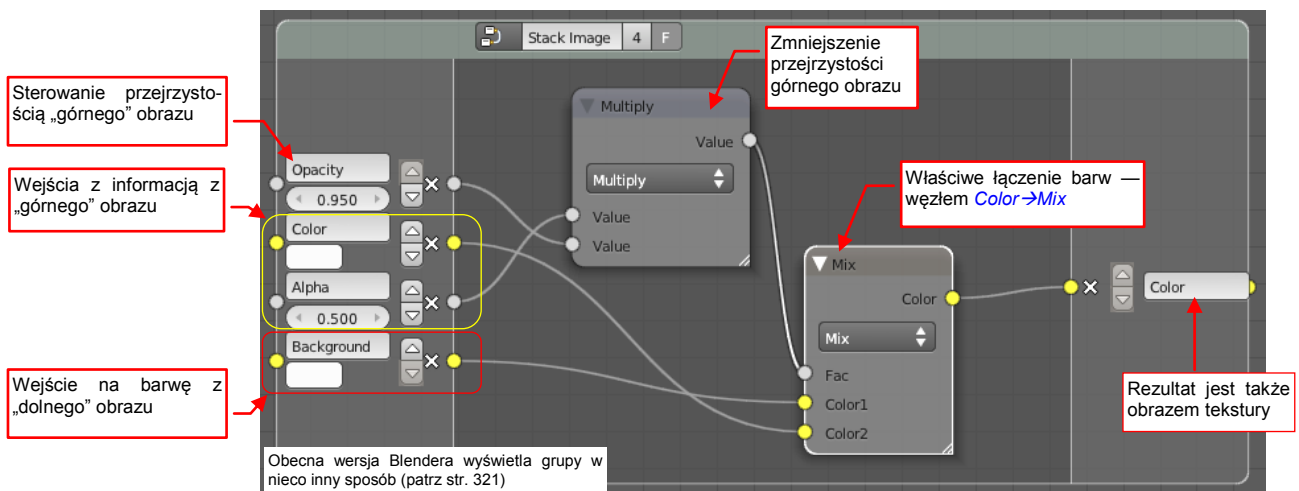
Węzeł **Stack Image** służy do nałożenia jednej tekstury na drugą, tak jak się nakładają na siebie warstwy w Inkscape lub GIMP (Rysunek 9.15.9):



Rysunek 9.15.9 Działanie węzła **Stack Image**

Podobnie jak w tych programach rysunkowych, do łączenia obydwu barw wykorzystywana jest nieprzejrzystość (komponent **Alpha**) pikseli nakładanego obrazu (por. str. 468). Stąd zazwyczaj nakładana tekstura ma przejrzyste tło — tak jak **decals.png** na ilustracji powyżej. Należy ją podłączyć do wejść **Color** i **Alpha** węzła **Stack Image**. Obraz tła powinien być nieprzejrzysty — przynajmniej w obszarach, które będą nałożone na powierzchnię modelu. Podłączasz go do wejścia **Stack Image:Background** (w naszym przykładzie jest to tekstura **color.png**). **Stack Image** ma dodatkową kontrolkę **Opacity**. Używaj jej do sterowania nieprzejrzystością nakładanej warstwy (tak samo jak sterujesz nieprzejrzystością warstw w GIMP lub Inkscape). Wyjściem z węzła **Stack Image** jest obraz tekstury, który możesz normalnie podłączyć do innego wejścia — na przykład barwy shadera. Wykorzystując węzły **Stack Image** możesz także zbudować „stos” obrazów tekstur — wystarczy podłączać rezultat z węzła leżącego „poniżej” do wejścia **Background** następnego („wyższego”) węzła **Stack Image**. W porównaniu ze „stosem” warstw w GIMP czy Inkscape scalonych w jedną teksturę, to rozwiązanie pozwala łączyć w Blenderze obrazy tekstur nakładanych z różnymi alternatywnymi rozwinięciami UV.

Rysunek 9.15.10 przedstawia wewnętrzną strukturę grupy **Stack Image**:

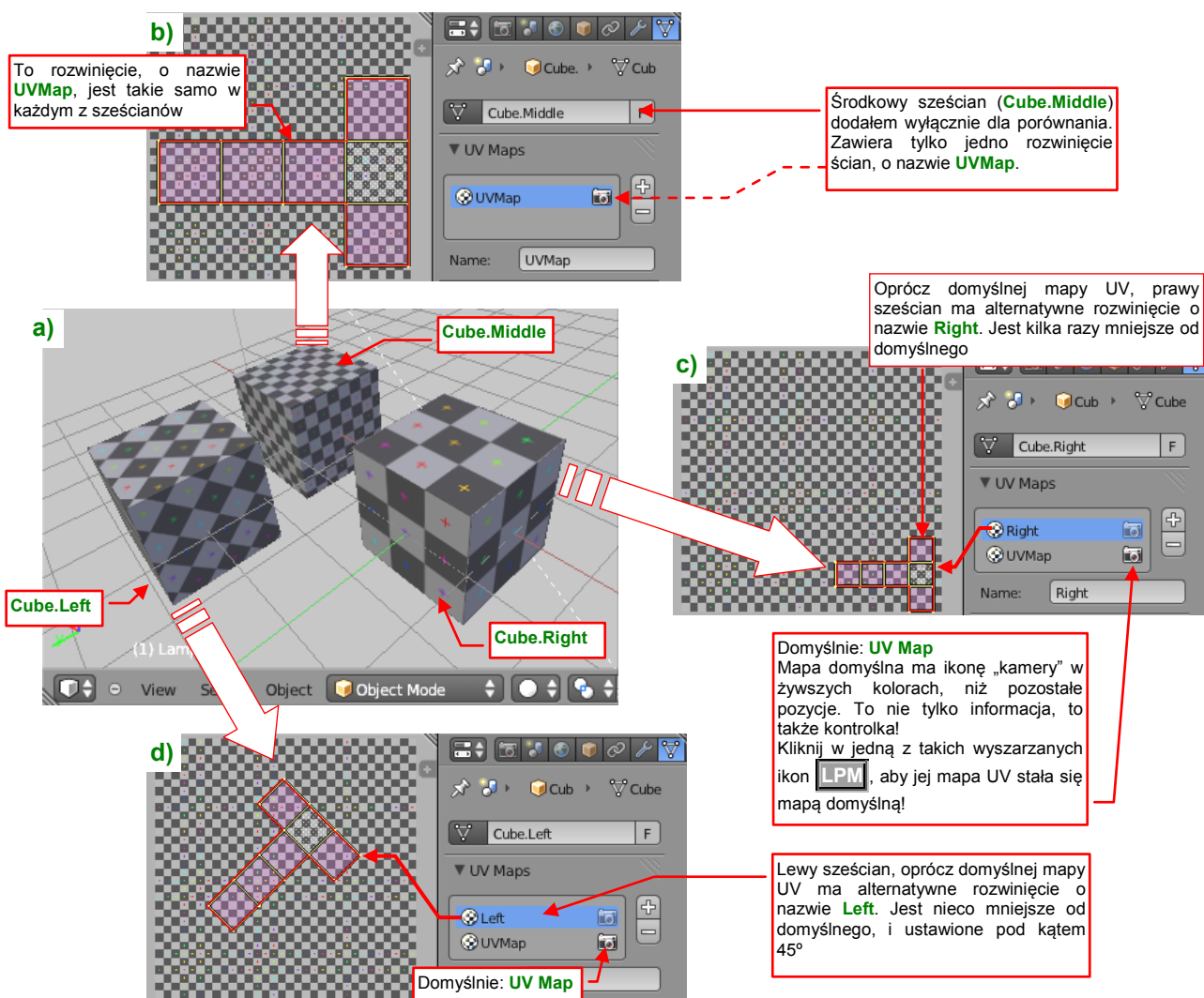


Rysunek 9.15.10 Budowa węzła **Stack Image**

Właściwe „mieszanie” barw łączonych tekstur odbywa się tu w węźle **Color→Mix** (z ustawioną domyślną operacją **Mix**). Wagą tego łączenia jest przejrzystość (**Alpha**) „górnego obrazu”. Dodałem tu jeszcze pomocniczy węzeł **Multiply**, który służy do ewentualnego osłabienia przejrzystości nakładanej tekstury.

## 9.16 Wykorzystanie alternatywnych rozwinięć UV (*Attribute*)

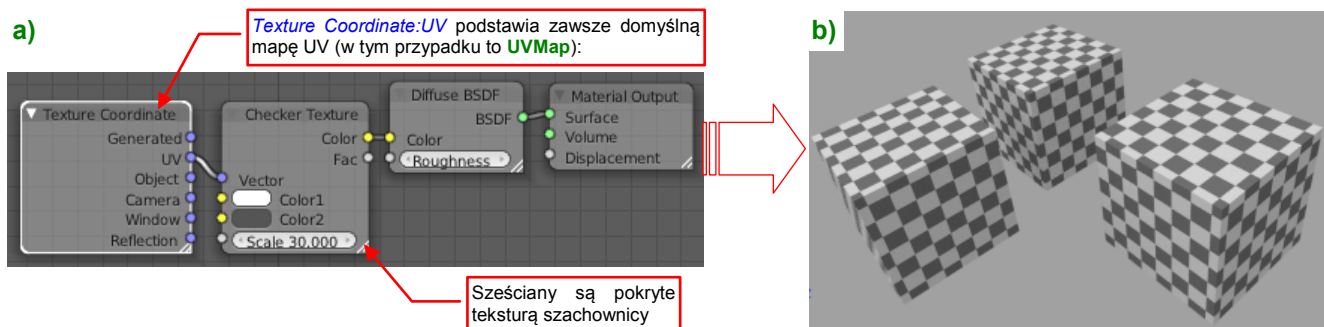
Jak właściwie używać alternatywnych map UV? W tym celu należy w Cycles użyć węzła *Attribute*. Abyś dobrze zrozumiał, jak to działa, przygotowałem przykładową scenę złożoną z trzech sześciątów (Rysunek 9.16.1a):



Rysunek 9.16.1 Rozwinięcia sześciątów na przykładowej scenie

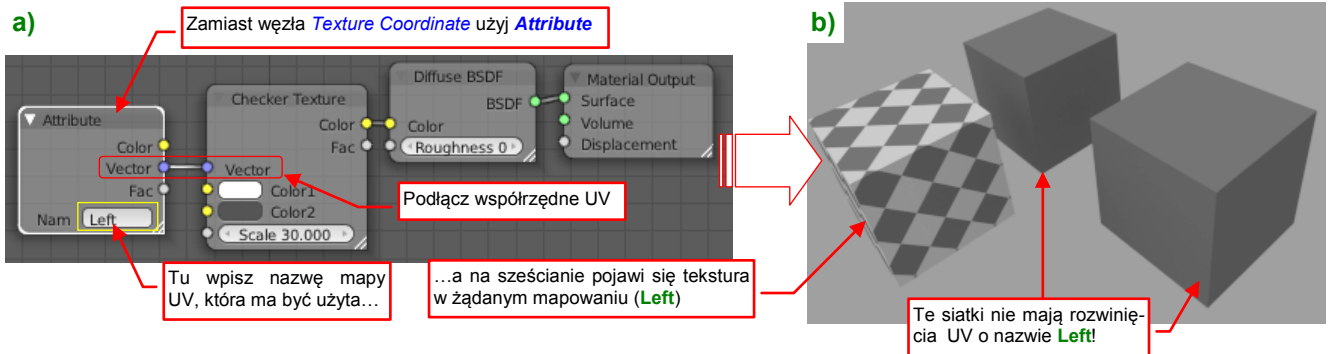
Każdy z sześciątów posiada identyczne rozwinięcie ścian o nazwie **UVMap** (Rysunek 9.16.1b). Jest to domyślna mapa UV ich siatek. (O tym, co to znaczy „domyślna” i w jaki sposób włączać to oznaczenie — por. str. 283). Oprócz tego sześciąt lewy i prawy mają alternatywne rozwinięcia UV, o nazwach **Left** (Rysunek 9.16.1d) i **Right** (Rysunek 9.16.1c).

Rysunek 9.16.2a) przedstawia materiał sześciątów, a Rysunek 9.16.2b) — ich testowy render:



Rysunek 9.16.2 Render sceny z użyciem domyślnego mapowania UV

Aby odwołać się do alternatywnego rozwinięcia UV, należy w Cycles użyć węzła **Attribute**. Wstaw go na miejsce węzła **Texture Coordinate**. W pole **Attribute:Name** wpisz nazwę mapy UV, którą chcesz wykorzystać (Rysunek 9.16.3a):

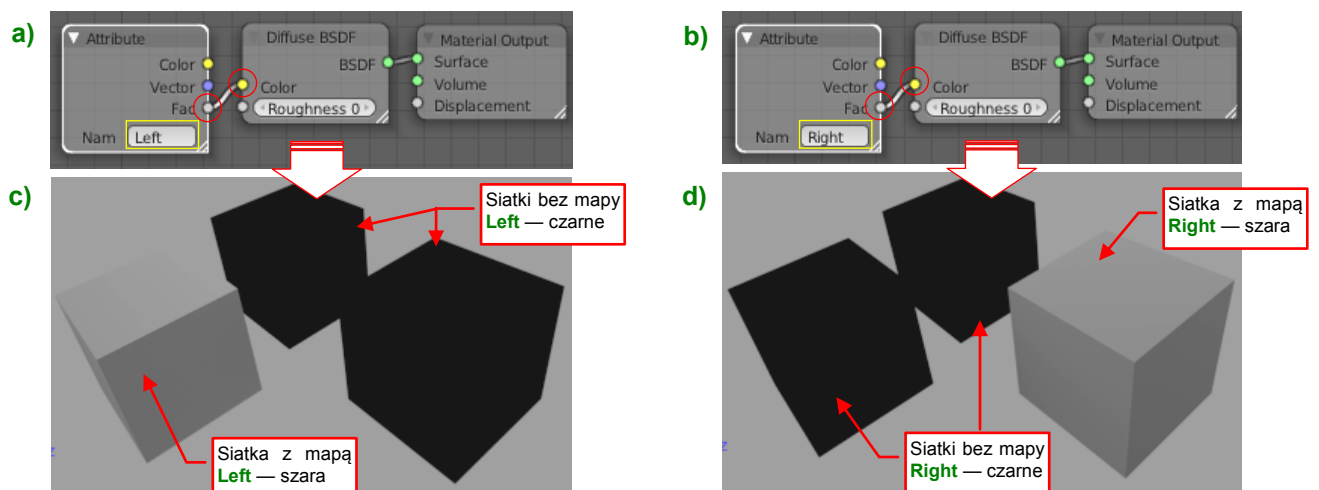


Rysunek 9.16.3 Wykorzystanie węzła **Attribute** w celu użycia innego rozwinięcia UV

Rezultat przedstawia Rysunek 9.16.3b). Tekstura pojawiła się tylko na siatce, która posiada rozwinięcie UV o podanej nazwie (w tym przypadku — **Left**). Siatki bez tego rozwinięcia są jednolicie szare.

Starszy renderer Blendera (**BI**) wykorzystywałby do pokrycia sześcianów bez mapy **Left** domyślne współrzędne UV (w tym przypadku — **UVMap**). **BI** pokryłby te siatki teksturą w taki sam sposób, w jaki pokazuje je Rysunek 9.16.2b). Ten efekt (po angielsku określany jako **fallback**) pozwolił mi w poprzedniej edycji książki przygotować alternatywne rozwinięcia UV tylko w tych siatkach, w których było to potrzebne. (Używałem wtedy **BI**, bo Cycles jeszcze nie istniał). Taki efekt wykorzystania domyślnych mapowań UV oszczędził mi sporo pracy. Cycles jednak tak nie działa. Czy można na to coś poradzić?

W takiej sytuacji zawsze próbuję szukać jakiegoś rozwiązania, sprawdzając rezultaty innych wyjść z badanego węzła. W tym przypadku sprawdziłem, co zwraca współczynnik **Attribute:Fac** (Rysunek 9.16.4):

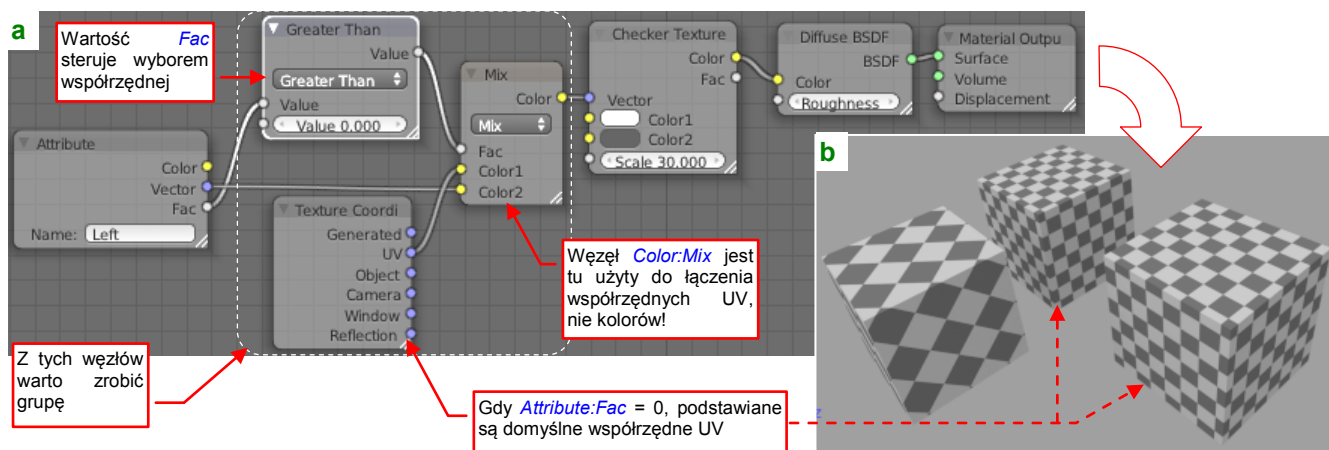


Rysunek 9.16.4 Sprawdzanie wartości **Attribute:Fac**

Zwracane przez węzły wartości skalarne można szybko sprawdzić, podłączając je bezpośrednio do koloru shadera materiału. W tym przypadku podłączyłem wyjście **Attribute:Fac** do wejścia **Color** shadera **Diffuse** (Rysunek 9.16.4a,b). Gdy w węźle **Attribute** jest wpisana nazwa (**Name**) **Left**, cała powierzchnia lewego sześcianu (tj. siatki zawierającej rozwinięcie **Left**) jest szara, a pozostałe sześciany — czarne (Rysunek 9.16.4c). Gdy zmienisz w węźle **Attribute** nazwę na **Right**, szary stanie się prawy sześcian (bo posiada rozwinięcie UV o tej samej nazwie). Pozostałe sześciany znów będą czarne (Rysunek 9.16.4d). Na tej podstawie można wnioskować, że:

- **Attribute:Fac** dla siatki zawierającej żądane rozwinięcie UV zwraca wartość 0.5, a dla pozostałych — 0.0

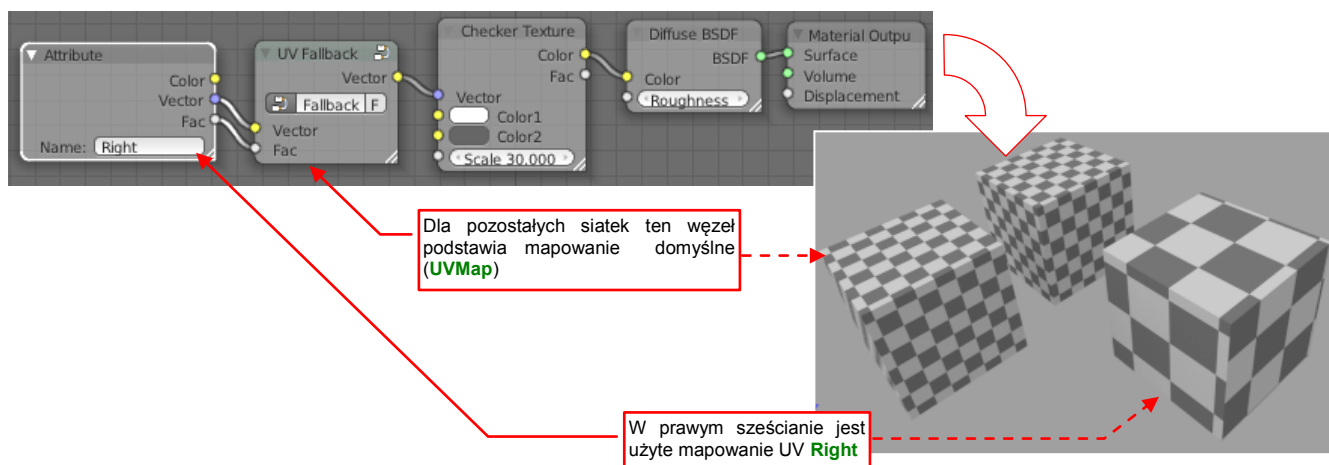
Taka reguła pozwala nam zbudować odpowiedni zespół węzłów, który podstawí domyślne współrzędne UV dla każdej siatki bez mapy UV o żądanej nazwie<sup>1</sup>. Aby przekształcić wartości uzyskiwane z *Attribute:Fac* dla siatki posiadającej mapę UV z 0.5 na 1.0, używam węzła *Math:Greater Than* (Rysunek 9.16.5a):



Rysunek 9.16.5 Dodatkowe węzły, uzupełniające brakujące współrzędne tekstury wartościami domyślnymi

(Nie mnożę ich np. przez 2, bo nie jestem pewien, czy zwracana wartość to dokładnie 0.5. Używając porównania „> 0” mniej ryzykuję). Jako „przełącznika” współrzędnych tekstur użyłem węzła *Color:Mix*<sup>2</sup> (Cycles nie zawiera wyspecjalizowanego węzła do łączenia danych typu *Vector*). Współrzędne z wejścia *Color1* będą użyte, gdy *Attribute:Fac* zwróci wartość = 0 (brak żądanej mapy UV). Powinny być wówczas użyte wartości domyślne, więc podłączyłem w to miejsce węzeł *Texture Coordinate*, który je dostarcza. Współrzędne z wejścia *Color2* będą użyte, gdy siatka zawiera żądaną mapę (*Mix:Fac* = 1.0), stąd podłączyłem w to miejsce wyjście w węzła *Attribute*. Udało mi się w ten sposób uzyskać pożądany rezultat — sześciany, które poprzednio były czarne, teraz są pokryte teksturą (Rysunek 9.16.5b).

Tak pożyteczny zespół węzłów warto jest połączyć w grupę, by zwiększyć przejrzystość schematów materiałów. Przygotowałem taki węzeł, i nadałem mu nazwę *UV Fallback* (Rysunek 9.16.6):



Rysunek 9.16.6 Wykorzystanie grupy *UV Fallback* do uzupełniania brakujących współrzędnych tekstur

Użycie jest proste: obydwa wejścia należy podłączyć do węzła atrybutu, a wyjście — do współrzędnych tekstur. Grupy *UV Fallback* będę używał wszędzie tam, gdzie wykorzystuję alternatywne mapowania UV.

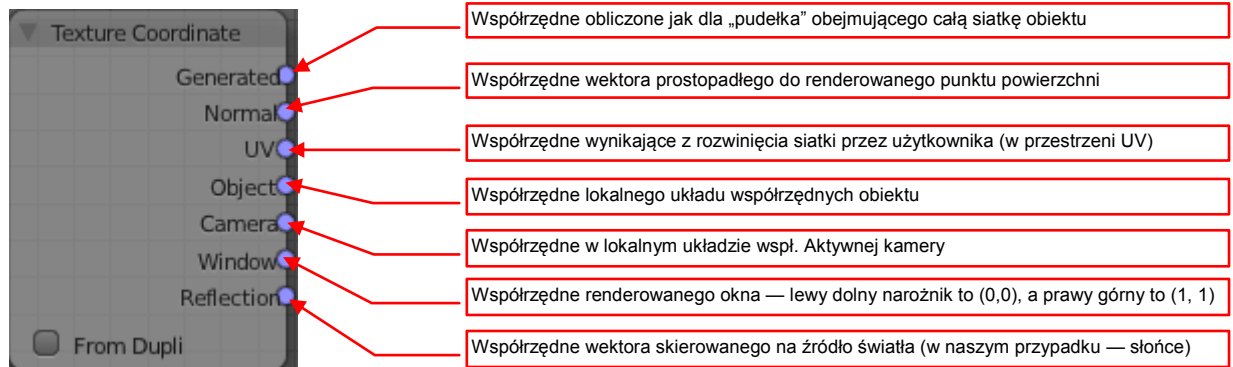
<sup>1</sup> Zwróć uwagę, że reguła, na której tu się oparłem, powstała na podstawie „obserwacji empirycznych”. Wartości, zwracane przez *Attribute:Fac* nie są na razie (kwiecień 2012) nigdzie udokumentowane. Stąd zawsze istnieje ryzyko, że to rozwiązanie przestanie działać w którejś z przyszłych wersji Blendera!

<sup>2</sup> To taki trick, który działa gdyż współrzędne UV mają taki sam zakres wartości co komponenty RGB barwy — od 0.0 do 1.0. W dodatku Cycles dokonuje „w locie” konwersji z  $U \rightarrow R$ ,  $V \rightarrow G$ ,  $R \rightarrow U$ ,  $G \rightarrow V$  — por. str. 780



### 9.17 Wykorzystanie współrzędnych tekstury (*Texture Coordinate*)

Dla każdego piksela renderowanej powierzchni można podać wiele alternatywnych współrzędnych tekstur. Każda z nich jest liczona inną metodą. W Cycles ich źródłem jest węzeł *Texture Coordinate*, dodawany do schematu poleceniem **Add → Input → Texture Coordinate** (Rysunek 9.17.1):



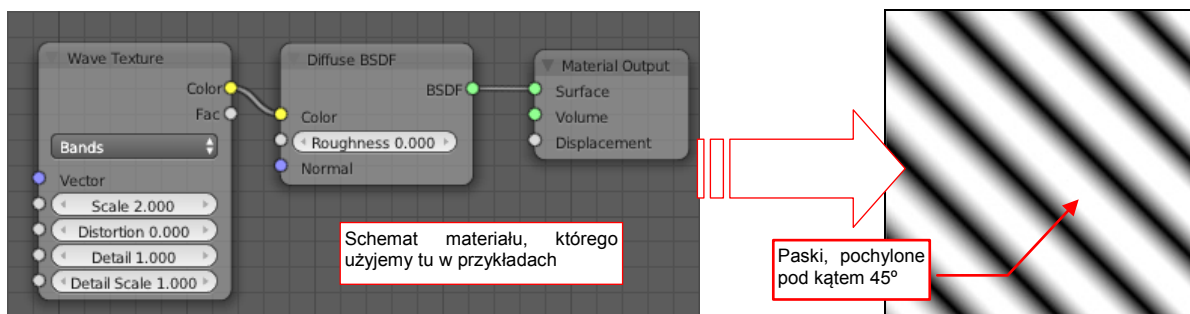
Rysunek 9.17.1 Węzeł *Texture Coordinate*

*Texture Coordinate* jest węzłem wejściowym, podobnie jak węzły *Geometry* czy *Light Path*. Każde z jego wyjść udostępnia inny rodzaj współrzędnych tekstury. Niektóre z nich są używane dość rzadko: na przykład współrzędne aktualnego okna — *Window* — możesz wykorzystać do podstawienia za modelem jakiegoś zdjęcia (por. str. 347). Dopilnuj tylko, by obszar tego obrazu tła i okna renderu miał takie same proporcje wysokości i szerokości. Inaczej będzie na renderze zdeformowany! W podobnym celu możesz wykorzystać współrzędne aktualnego piksela wyrażone w lokalnym układzie współrzędnych kamery (*Camera* — por. str. 348). W praktyce nigdy nie potrzebowałem wykorzystywać mapowania *Reflection* — musisz z nim sam poeksperymentować. Te wszystkie trzy rodzaje współrzędnych wiążą obraz z globalnym układem współrzędnych sceny. Tekstura zamapowana za pomocą wyjść *Window*, *Camera* lub *Reflection* zmienia swoje położenie na obiekcie „podążając” za obserwatorem.

Pozostałe rodzaje mapowań wiążą obraz tekstury z układem współrzędnych obiektu.

Wyjście *UV* zwraca współrzędne domyślnej mapy UV, przypisanej do siatki (czyli tej, która jest oznaczona jako „dla renderowania” — por. str. 380). Pamiętaj jednak, że najpierw sam je musisz stworzyć. W Blenderze siatki nie posiadają żadnego „domyślnego” rozwinięcia.

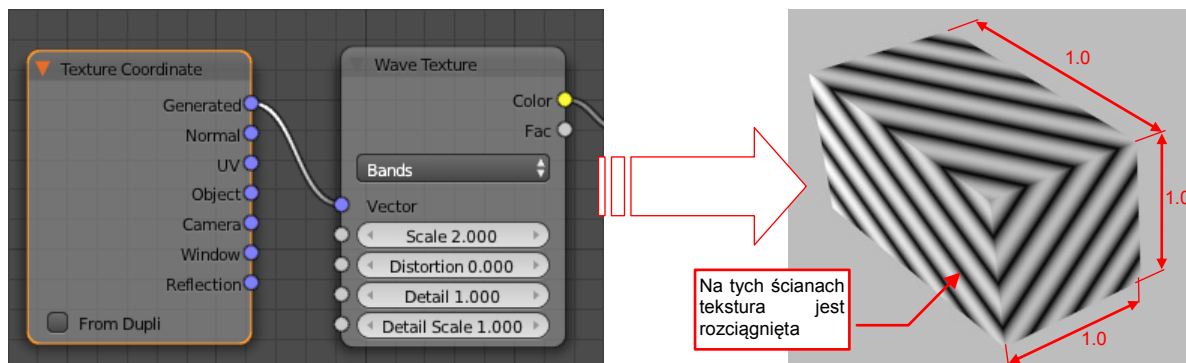
Aby pokazać działanie pozostałych mapowań: *Generated*, *Object* i *Normal*, posłużę się prostą teksturą pasków pochylonych pod kątem 45°. (Jest to domyślny efekt działania tekstury proceduralnej *Wave*). Rysunek 9.17.2 przedstawia schemat testowego materiału i obraz, który tworzy na płaskiej powierzchni:



Rysunek 9.17.2 Tekstura, której użyjemy w przykładach: *Wave*

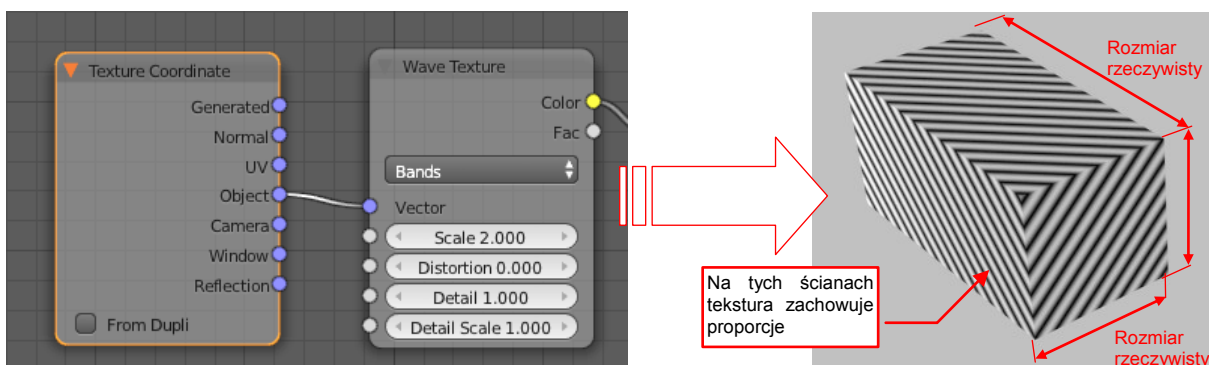
Ja widzieć, tekstura *Wave* jest wstawiona z domyślnymi parametrami: wzorem *Bands* i skalą (*Scale*) = 2.

Aby wyznaczyć współrzędne **Generated**, Blender tworzy robocze „pudełko” obejmujące całą siatkę i zorientowane tak jak lokalny układ współrzędnych obiektu. Niezależnie od proporcji uzyskanego w ten sposób prostopadłościanu, współrzędne w każdym z kierunków zmieniają się w zakresie od 0.0 do 1.0 (Rysunek 9.17.3):



Rysunek 9.17.3 Mapowanie **Generated**

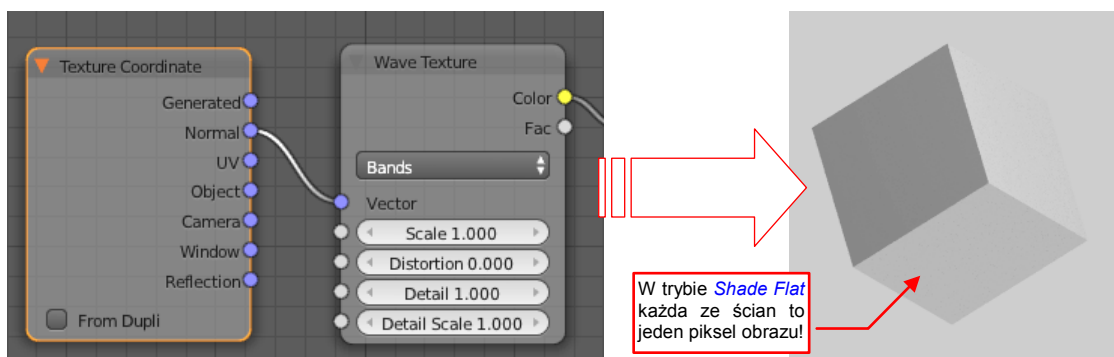
W efekcie gdy obiekt ma podłużny kształt, obraz tekstury nałożony w trybie **Generated** ulega rozciągnięciu. Aby zachować proporcje obrazu, użyj lokalnych współrzędnych obiektu — czyli danych z wyjścia **Object** (Rysunek 9.17.4):



Rysunek 9.17.4 Mapowanie **Object**

W tym trybie cały obraz tekstury ma domyślny rozmiar 1x1 jednostki Blendera. Oczywiście, możesz zmienić jego wielkość za pomocą dodatkowego węzła **Mapping** (por. str. 332). Jeżeli powierzchnia obiektu jest większa od obrazu — Blender powieli go (tak jak pola na szachownicy).

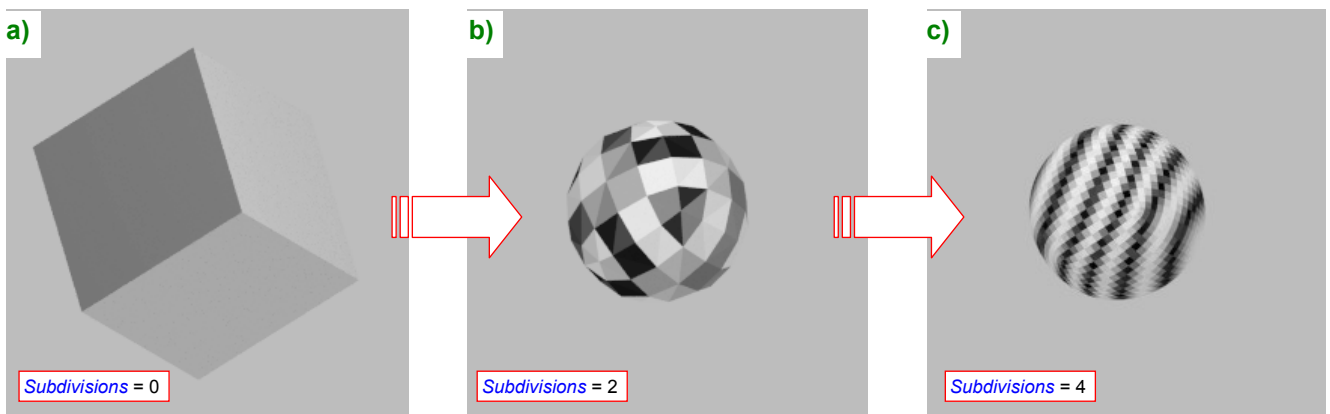
Wreszcie mapowanie wykorzystujące aktualny kierunek prostopadły do powierzchni: **Normal** (Rysunek 9.17.5):



Rysunek 9.17.5 Mapowanie **Normal**

Wykorzystanie tych współrzędnych do nałożenia tekstury pasków na domyślny sześcian nie tworzy żadnego szczególnego efektu — każda ze ścian ma jednolitą barwę. Dlaczego? Bo każdy piksel tych ścian ma taki sam kierunek wektora normalnego (to tryb **Shade Flat** — por. str. 304). Stąd na każdą z nich jest nałożony pojedynczy piksel obrazu tekstury!

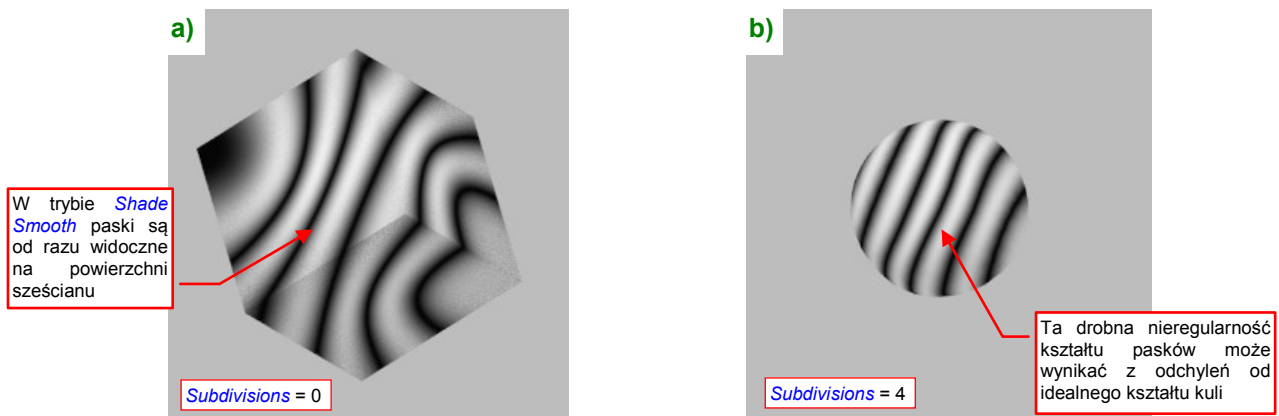
Rysunek 9.17.6 pokazuje, co się stanie gdy siatkę cieniowaną w trybie *Shade Flat* zaczniesz zwiększać liczbę ścian tak, by każda była pochylona pod innym kątem (np. za pomocą modyfikatora *Subdivision Surface*):



Rysunek 9.17.6 Mapowanie pasków w trybie *Normal* (dla *Shade Flat*)

Po lewej widzimy zwykły sześcian przed podziałem — każda z jego ścian ma jednolity kolor (Rysunek 9.17.6a). Gdy zwiększymy liczbę podziałów do 2 (Rysunek 9.17.6b), każdy z podstawowych trójkątów na jaki Blender dzieli tę siatkę ma nadal stały kolor. Barwy sąsiednich ścian różnią się między sobą. Po kolejnych podziałach zaczniesz dostrzegać na wygładzonej powierzchni siatki wzór pasków (Rysunek 9.17.6c).

Ten wzór jest od razu widoczny, nawet dla sześcianu, gdy kierunki normalne do siatki są interpolowane pomiędzy wierzchołkami w trybie *Shade Smooth* (Rysunek 9.17.7a):

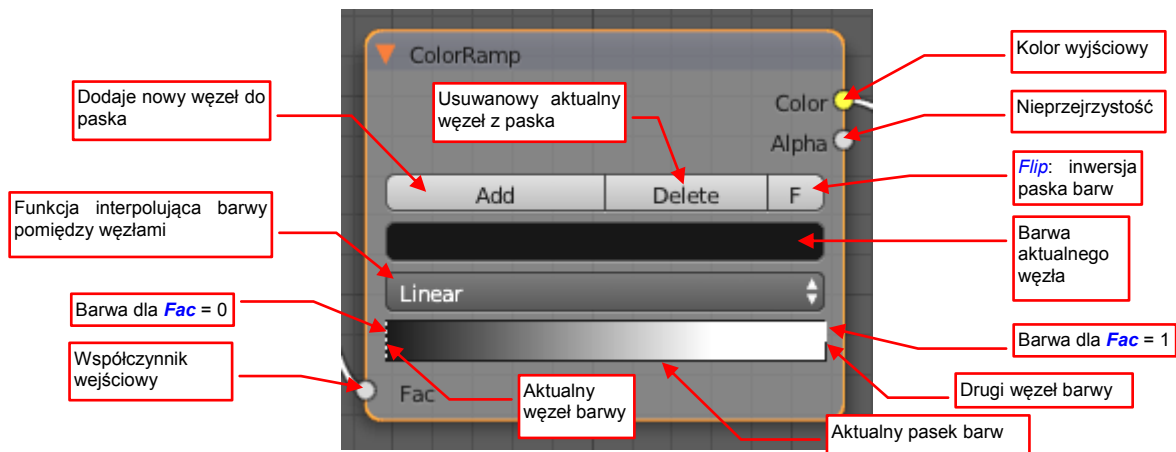


Rysunek 9.17.7 Mapowanie obrazu pasków w trybie *Normal* (dla *Shade Smooth*)

Dla większej liczby ścian dla obydwu trybów wygładzania — *Shade Flat* i *Shade Smooth* uzyskujemy bardzo podobny obraz (por. Rysunek 9.17.7b i Rysunek 9.17.6c).

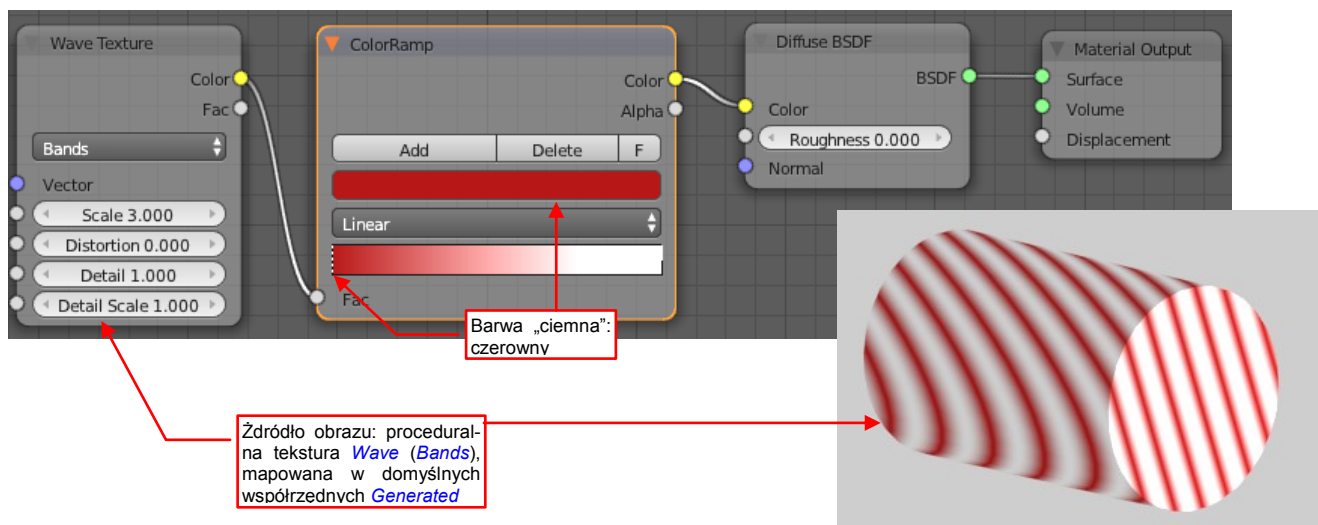
### 9.18 Wykorzystanie spektrum barw (*Color Ramp*)

Węzeł *Color Ramp* zwraca dla podanych wartości współczynnika *Fac* (0.0...1.0) odpowiedni kolor ze swojego paska spektrum barw. Możesz go dodać do schematu poleceniem *Add → Convertor → Color Ramp* (Rysunek 9.18.1):



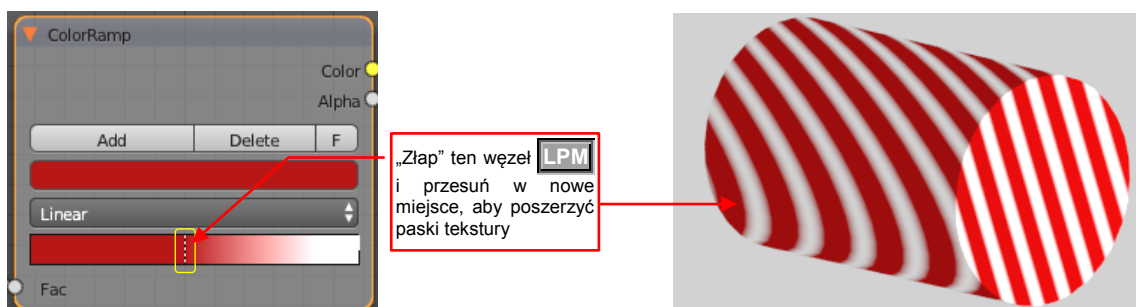
Rysunek 9.18.1 Węzeł *Color Ramp*

Ten efekt możesz wykorzystać do modyfikacji tekstur proceduralnych — na przykład zmiany domyślnej barwy tekstury pasków (*Wave*: Rysunek 9.18.2):



Rysunek 9.18.2 Zmiana barwy tekstury pasków za pomocą węzła *Color Ramp*

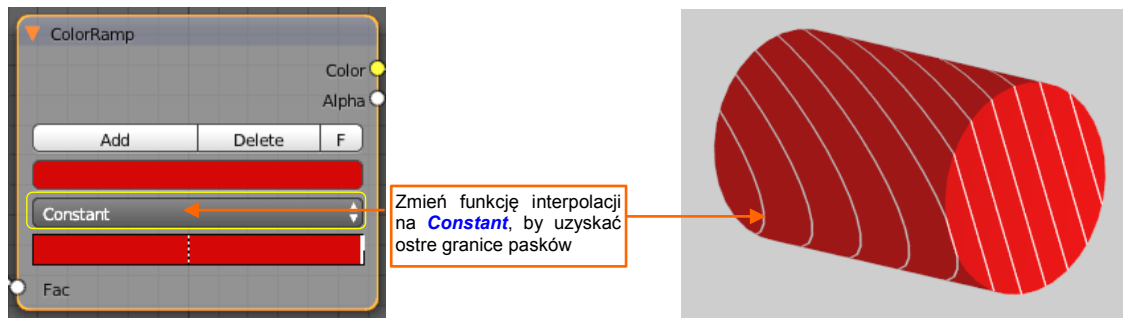
Aby zwiększyć szerokość pasków na obiekcie, złap (*LPM*) za czerwony węzeł paska barw i przesun go w prawo (Rysunek 9.18.3):



Rysunek 9.18.3 Zmiana szerokości pasków poprzez zmianę położenia węzła barwy

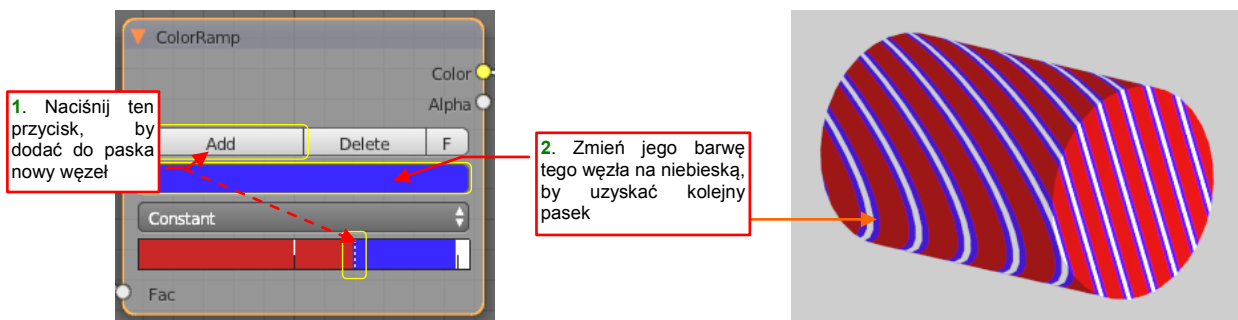


Aby uzyskać ostrą granicę pasków, możesz zmienić funkcję interpolującą na **Constant** (Rysunek 9.18.4):



Rysunek 9.18.4 Uzyskanie ostrej granicy pasków poprzez zmianę interpolacji brzo (na **Constant**)

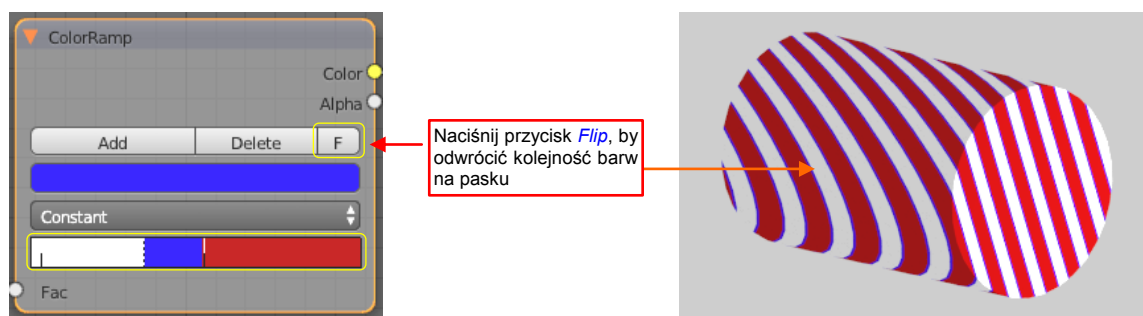
Przyciskiem **Add** możesz dodać do paska kolejny punkt węzłowy. Automatycznie po dodaniu węzeł staje się aktywny. Możesz np. zmienić jego barwę na niebieską, aby dodać do wzoru kolejny pasek (Rysunek 9.18.5):



Rysunek 9.18.5 Dodanie do paska kolejnego węzła

Węzeł aktywny, dla którego barwę wyświetla kontrolka powyżej, możesz wybrać kliknięciem **LPM**. Aby usunąć węzeł aktywny, użyj przycisku **Delete**.

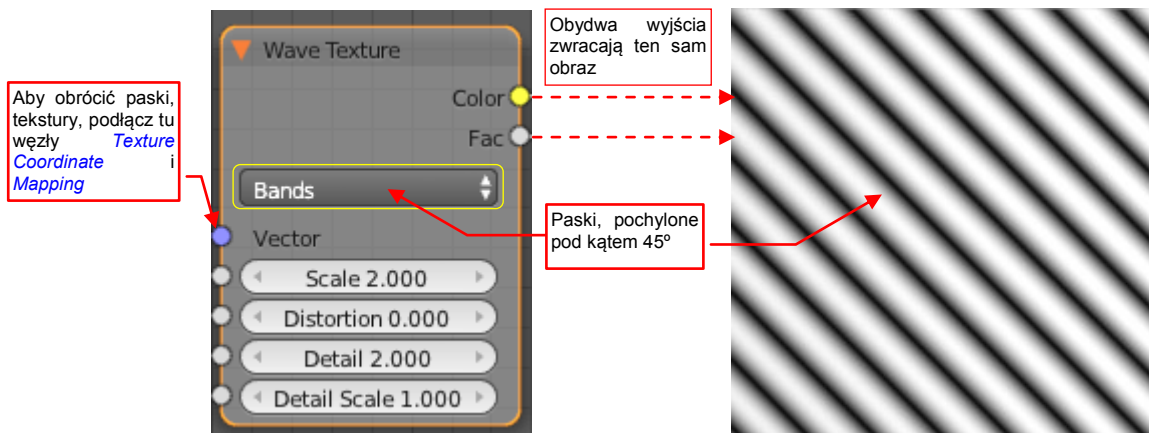
**Color Ramp** zawiera także drobne udogodnienie: przycisk **F** (**Flip**), odwracający kolejność barw na pasku (Rysunek 9.18.6):



Rysunek 9.18.6 Inwersja paska

### 9.19 Proceduralna tekstura pasków (*Wave Texture*)

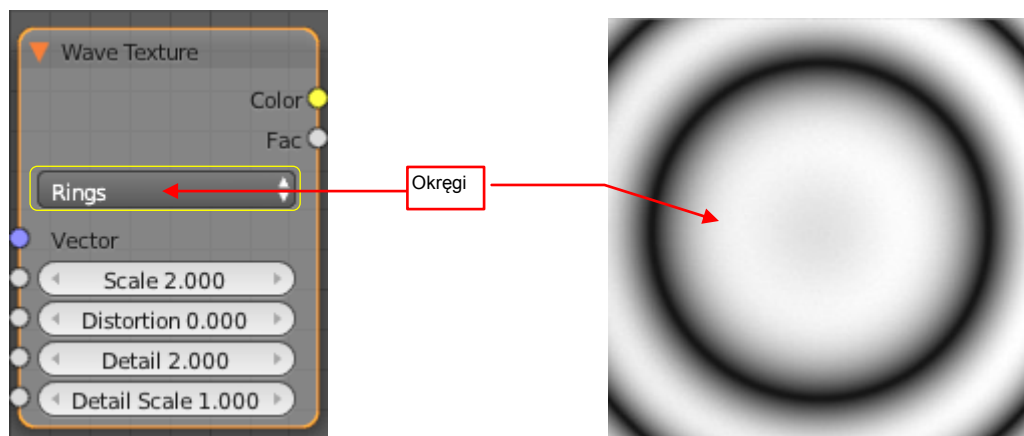
Teksturę *Wave* możesz wykorzystać gdy potrzebujesz jakiegoś wzoru pasków lub pierścieni. Podobnie jak inne węzły tekstur dodajesz ją do schematu poleceniem **Add→Texture→Wave Texture**. Bezpośrednio po wstawieniu węzeł jest ustawiony na generowanie pasków (*Bands*) pochylonych pod kątem 45° (Rysunek 9.18.1):



Rysunek 9.19.1 Węzeł *Wave Texture* i jego domyślny rezultat (paski — *Bands*)

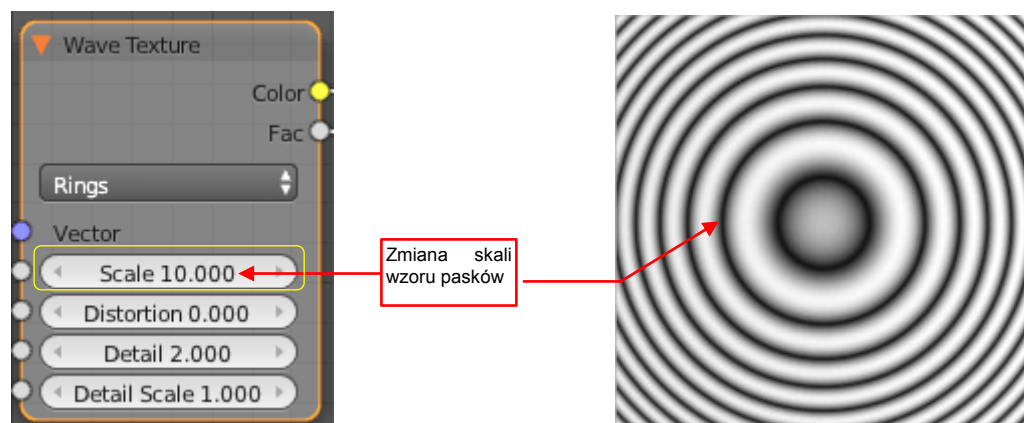
Aby zmienić kierunek pasków, podłącz do wejścia *Vector* połączone węzły *Texture Coordinate* i *Mapping*. Następnie w węźle *Mapping* ustaw odpowiedni obrót (jak na str. 342).

Gdy przełączysz węzeł w tryb *Rings*, uzyskasz wzór koncentrycznych okręgów (Rysunek 9.19.2):



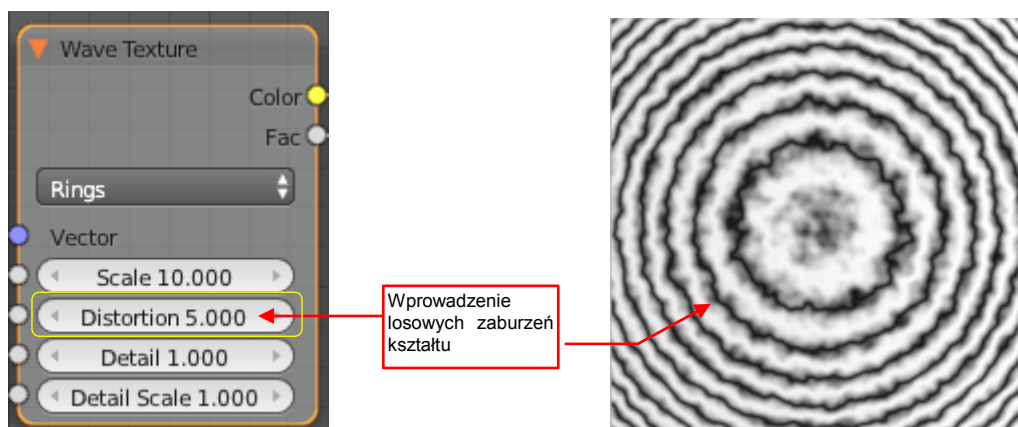
Rysunek 9.19.2 Okręgi, wygenerowane w trybie *Rings*

Aby zobaczyć ich więcej, możesz zwiększyć skalę (*Scale*) tekstury — to zagęszcza paski (Rysunek 9.19.3):



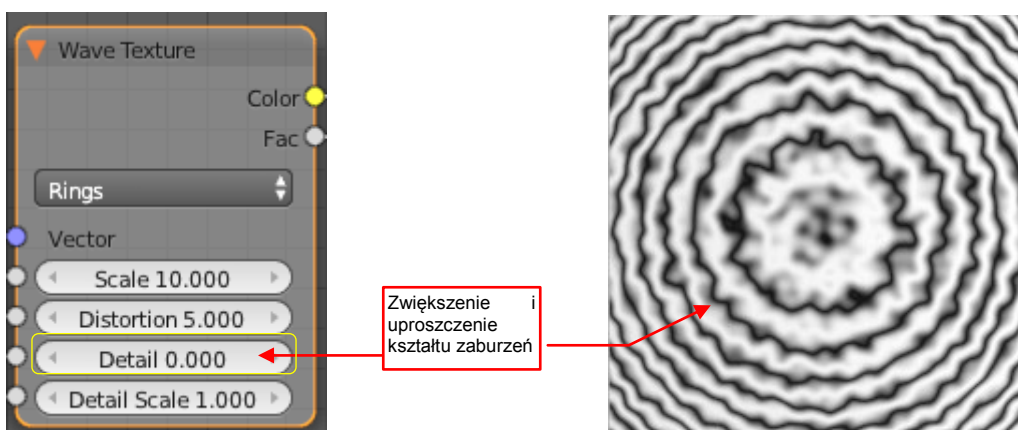
Rysunek 9.19.3 Zagęszczenie wzoru poprzez zwiększenie skali obrazu

Aby wprowadzić do uzyskanych pasków losowe zaburzenia, zwiększ parametr **Distortion** (Rysunek 9.19.4):



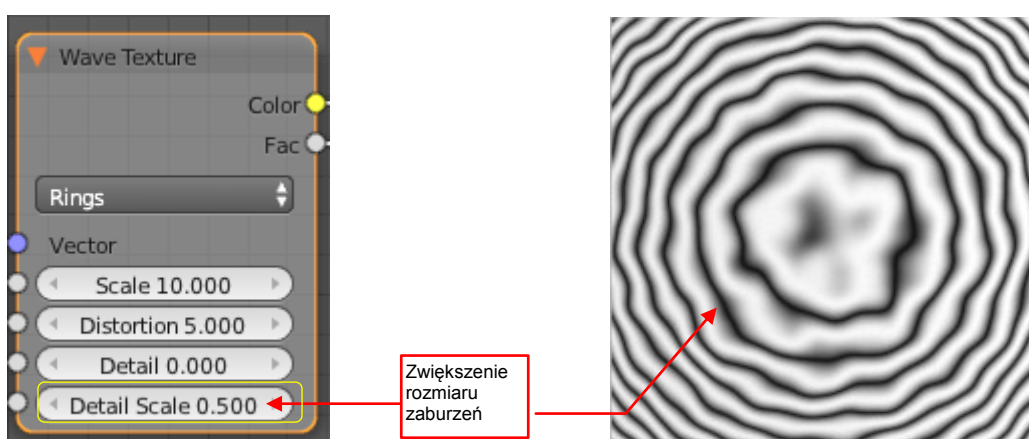
Rysunek 9.19.4 Wprowadzanie do wzoru pasków losowych zaburzeń

Postać tych zaburzeń można regulować zmieniając parametr **Detail** (Rysunek 9.19.5):



Rysunek 9.19.5 Uproszczenie kształtu zaburzeń pasków

Znacznie większy wpływ na kształt zaburzeń ma parametr **Detail Scale**. Gdy go zmniejszysz, deformacje ulegną wydłużeniu (Rysunek 9.19.6):

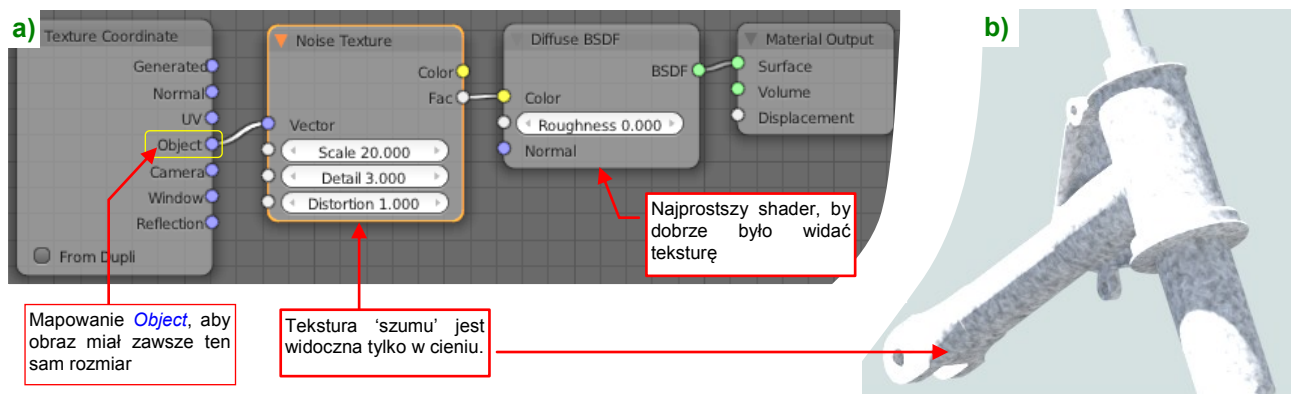


Rysunek 9.19.6 Wydłużenie deformacji pasków tekstury

## 9.20 Stworzenie tekstury „ogólnych zabrudzeń”

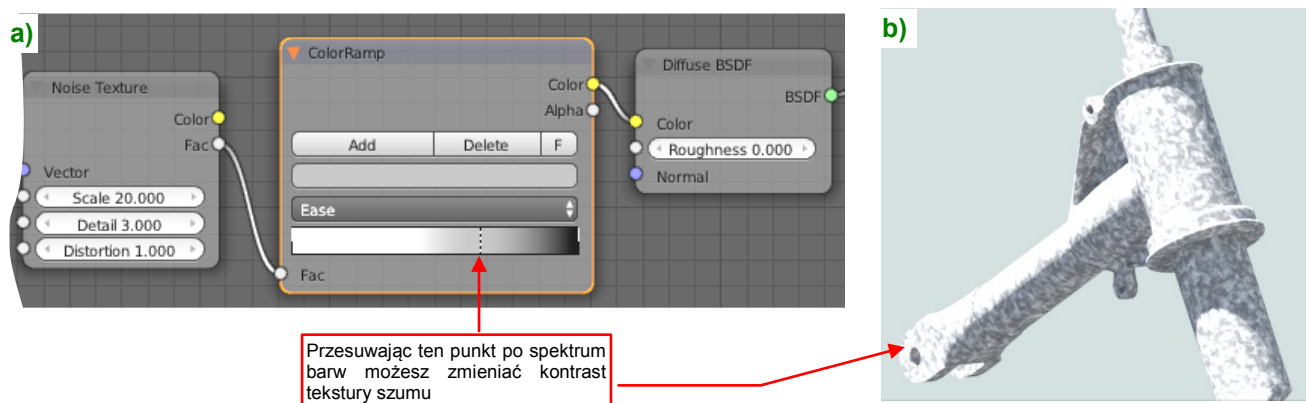
Zazwyczaj każdy element ma zróżnicowaną powierzchnię na której są losowo rozłożone ciemniejsze i jaśniejsze obszary. Na niektórych materiałach te różnice odcieni, połyskliwości i szorstkości są ledwo widoczne. Na innych — widać je wyraźnie. Zazwyczaj jest to efekt pokrycia samolotu cienką warstwą kurzu i innych zanieczyszczeń. Odtworzymy go w tej sekcji za pomocą paru tekstur proceduralnych.

Teksturę najwygodniej jest sprawdzać na najprostszym, białym materiale testowym. Na początek nałożmy tu zwykły obraz typu **Noise Texture**, zamapowany w trybie **Texture Coordinate: Object** (Rysunek 9.20.1a):



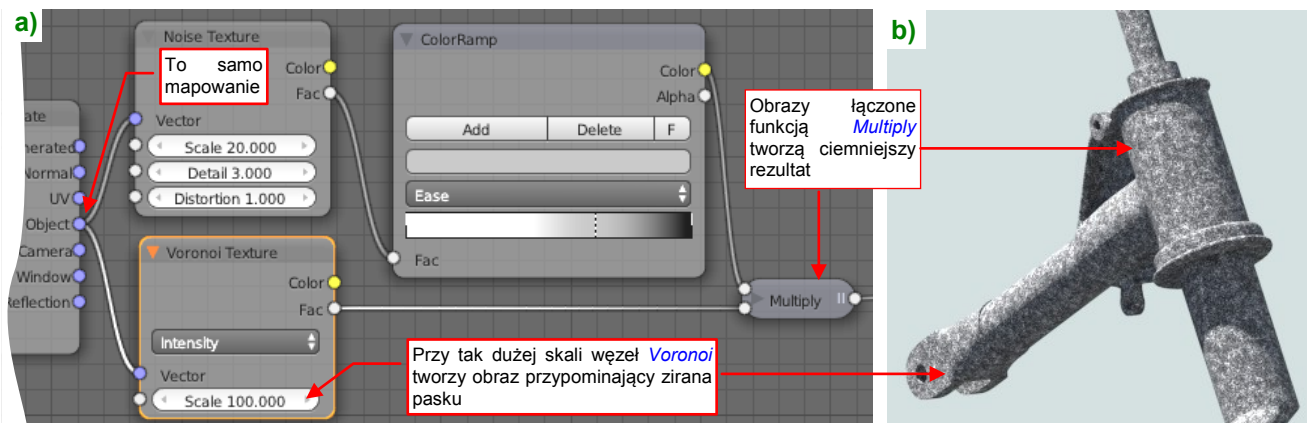
Rysunek 9.20.1 Pierwsza wersja materiału testowego

Rysunek 9.20.1b) przedstawia rezultat. Jak widać, cętki tekstury znikają na mocno naświetlonych powierzchniach. Aby je podkreślić, dodaj do schematu węzeł **Color Ramp** (Rysunek 9.20.2):



Rysunek 9.20.2 Wzmocnienie obrazu za pomocą węzła **Color Ramp**

Teraz dodaj drugi obraz — „ziarnistej” tekstury **Voronoi**, i połącz ją z pierwszym (**Multiply** — Rysunek 9.20.3):

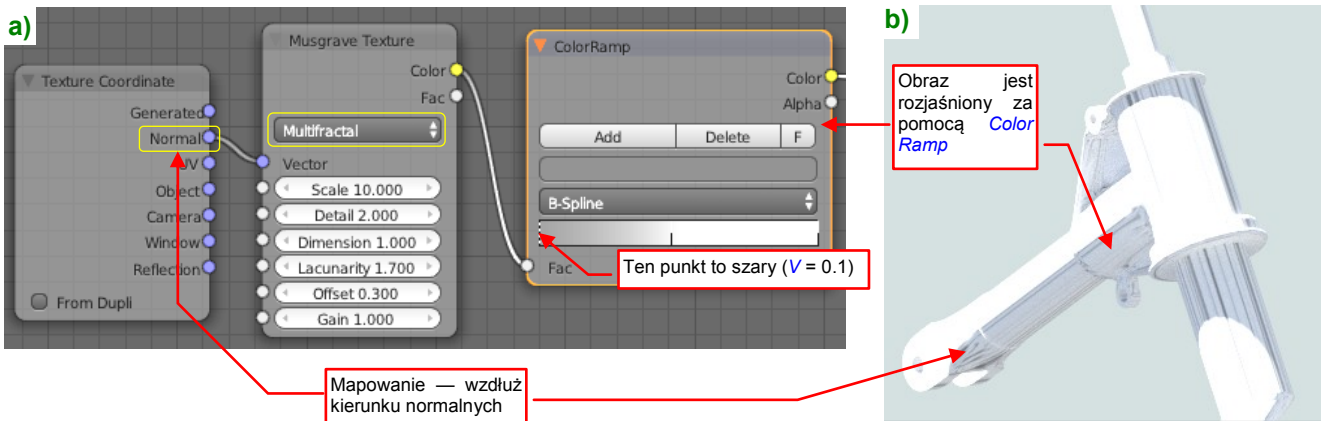


Rysunek 9.20.3 Dodanie tekstury „ziarnistości”



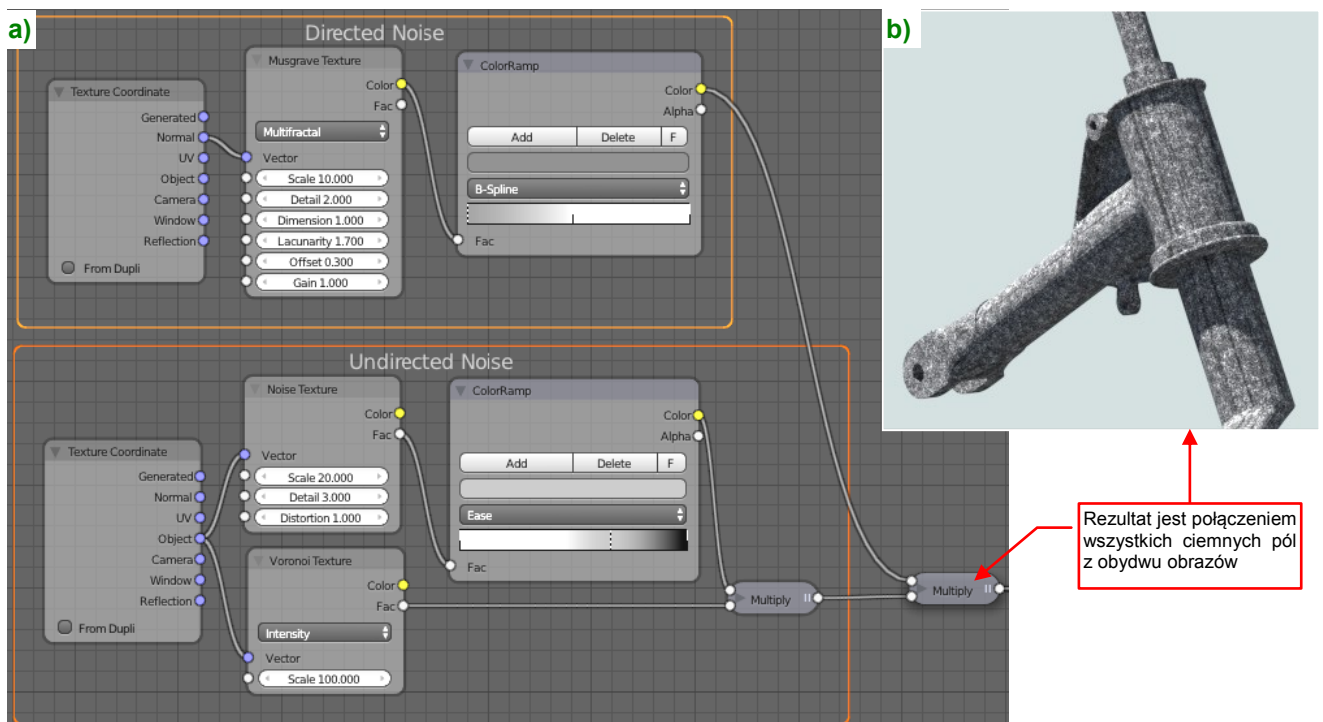
Oprócz bezkierunkowego wzoru, który pokazuje Rysunek 9.20.3b), czasami przydaje się dodać do szumu jakieś elementy o jednolitym kierunku. W tym celu używam kolejnej tekstury — **Musgrave**, zamapowanej we współrzędnych **Normal** (tzn. w kierunku prostopadłym do powierzchni). Ten rodzaj współrzędnych zamienia każdą teksturę proceduralną w paski. (Odstępny pomiędzy paskami i ich barwa zależą wyłącznie od rozmieszczenia i proporcji ciemnych i jasnych plam na oryginalnym obrazie. Dlatego wybrałem teksturę **Musgrave:Multifractal**, bo tworzy duże, jasne odstępy).

Rysunek 9.20.4a) przedstawia schemat testowego obrazu. Zwróć uwagę na użycie w tym miejscu węzła **Color Ramp**. Służy tu do rozjaśnienia obrazu na obiekcie (aby te paski dobrze się komponowały z szumem bezkierunkowym w węźle **Multiple**). Rysunek 9.20.4b) pokazuje uzyskany obraz, zamapowany na obiekcie:



Rysunek 9.20.4 Drugi obraz — szum skierowany

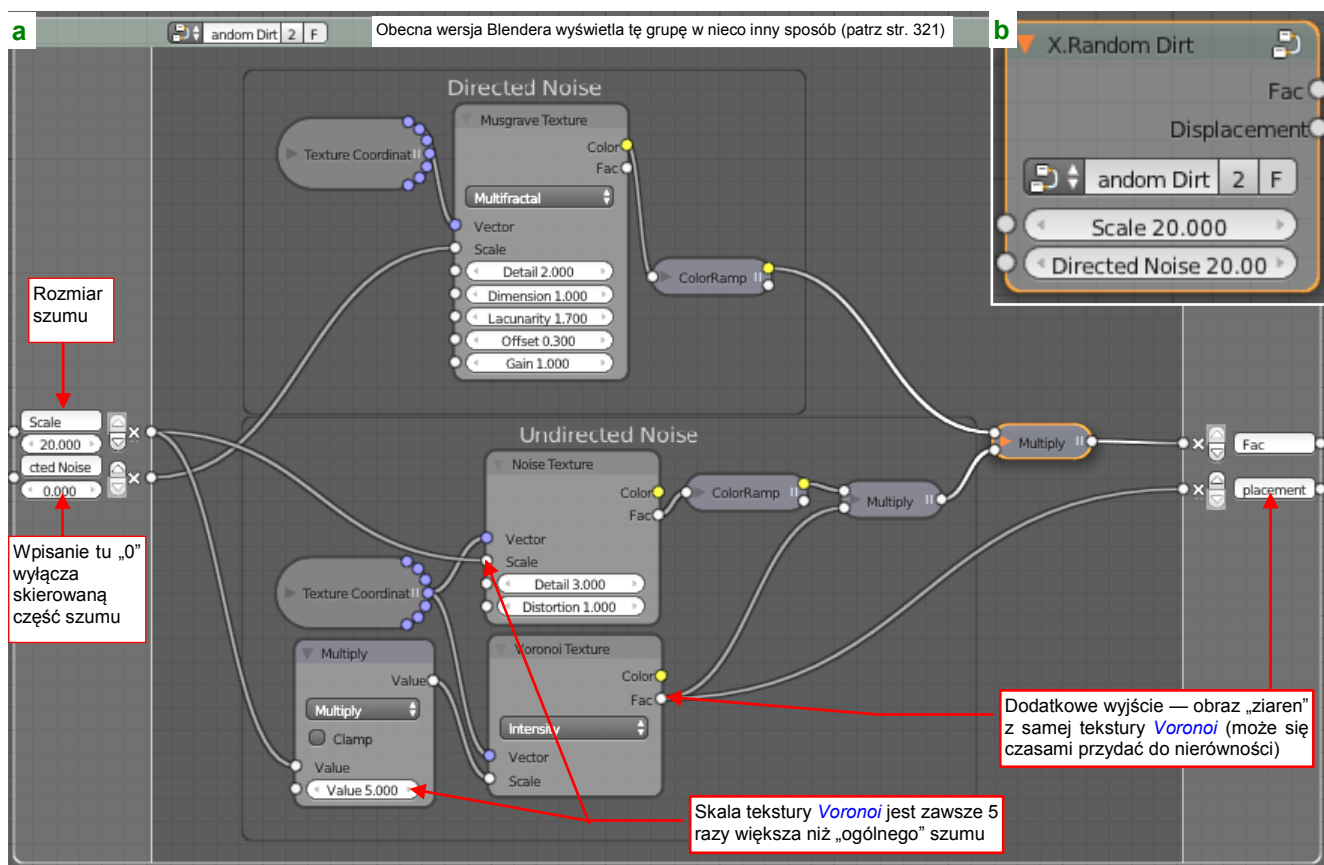
Teraz połącz ze sobą (w kolejnym węźle **Multiply**) przygotowany wcześniej szum bezkierunkowy (Rysunek 9.20.3b) z przygotowanym w ten sposób szumem skierowanym (Rysunek 9.20.5b):



Rysunek 9.20.5 Złożenie dwóch szumów

(Specjalnie wybrałem najjaśniejszą wersję tekstury **Musgrave** na obraz kierunkowy, aby nie powodować dalszego zaciemnienia otrzymanego obrazu).

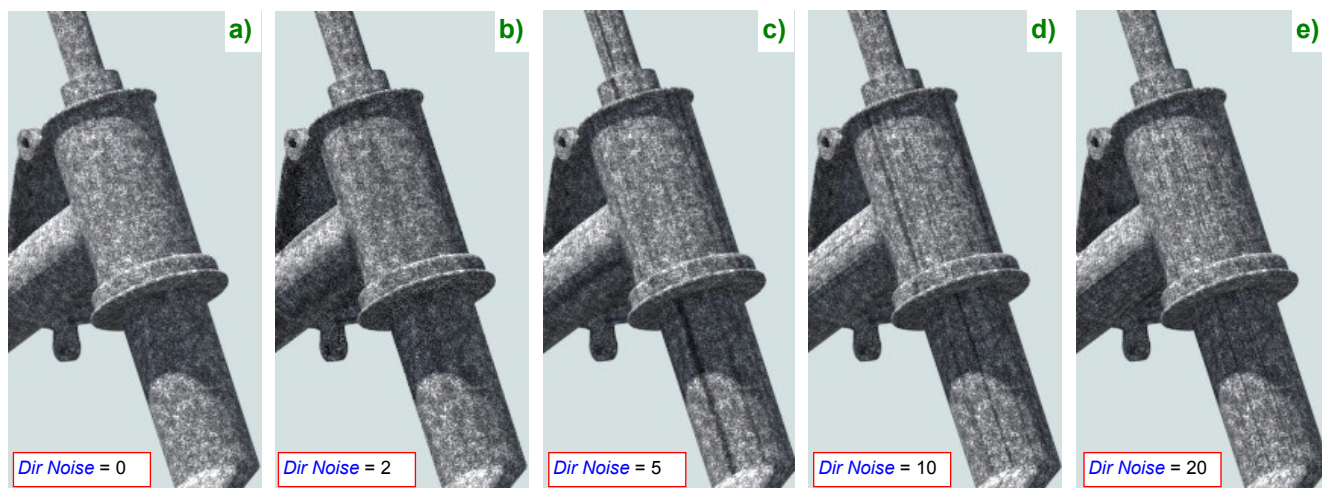
Taki zespół węzłów będziemy używać w charakterze „szumu bazowego” w wielu materiałach. Stąd proponuję „spakować” je w bardziej poręczną grupę (Rysunek 9.20.6):



Rysunek 9.20.6 Gotowy komponent — zawartość (a) i wygląd zewnętrzny (b)

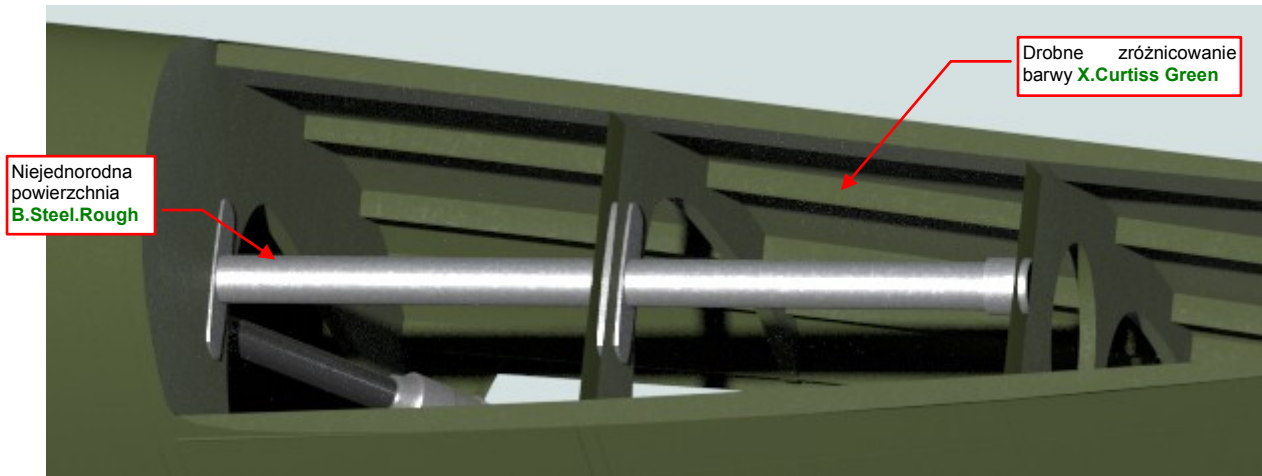
Rysunek 9.20.6a) przedstawia schemat tej tekstury, a Rysunek 9.20.6b) — jej postać zewnętrzną. Nazwałem ją **X.Random.Dirt** i jest wykorzystywana w wielu materiałach. Oprócz właściwego obrazu, udostępnionego na wyjściu **Fac**, dodałem do tego węzła drugi punkt wyjściowy: **Displacement**. Ten współczynnik to sama tekstura **Voronoi** — jeżeli chcesz, możesz ten punkt podłączyć do wejścia **Material Output:Displacement**. Uzyskasz w ten sposób efekt drobnych chropowatości powierzchni.

Pierwszy parametr grupy **X.Random Dirt** — **Scale** — pozwala regulować rozmiar uzyskanego „szumu”. Drugi — **Directed Noise** — zmienia obraz użytego szumu kierunkowego. Wartość = 0 oznacza wyłączenie tego komponentu. Wyższe wartości tego parametru zwiększają liczbę „pasków” (Rysunek 9.20.7):



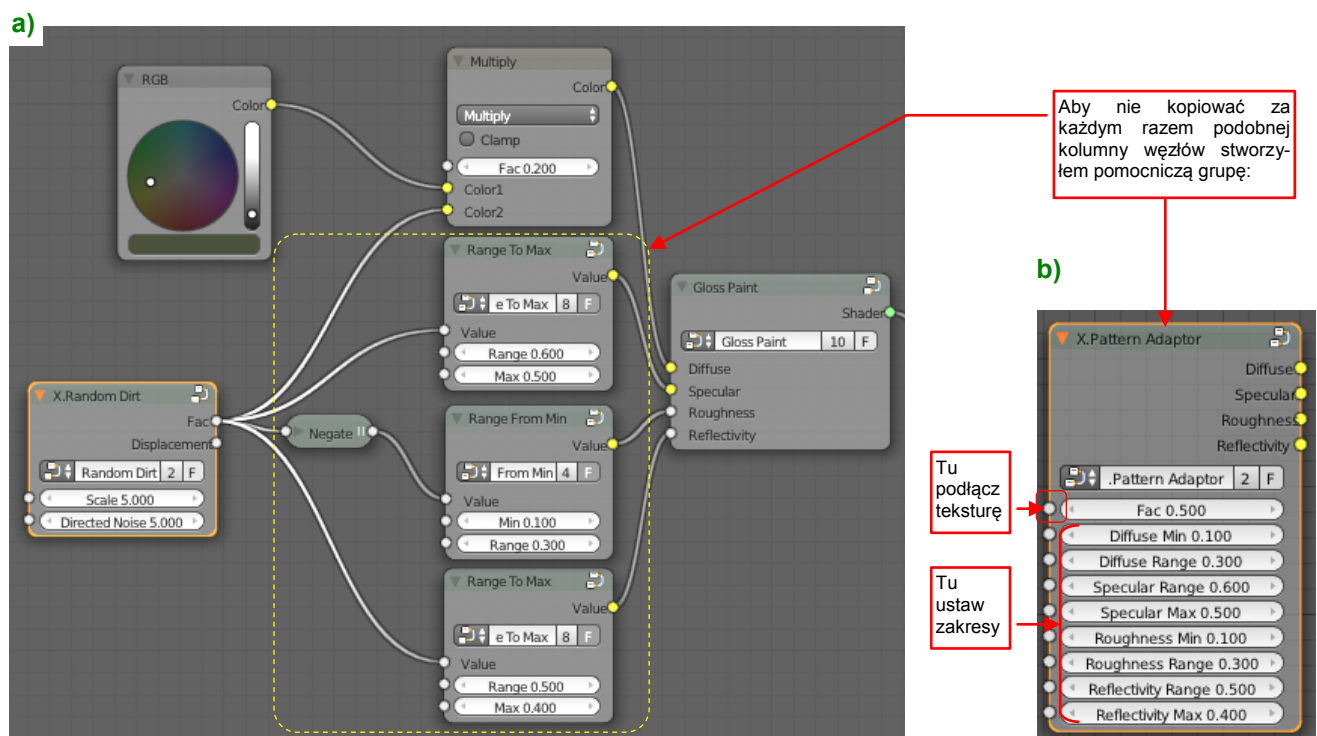
Rysunek 9.20.7 Wpływ parametru **Directed Noise**

Komponent **X.Random.Dirt** wykorzystuję w różnych materiałach powierzchni wewnętrznych (**B.Skin.Inner**) oraz detalach modelu (np. **B.Steel.Rough**). Rysunek 9.20.8 przedstawia fragment ogona, w którym wykorzystujemy obydwie materiały:



Rysunek 9.20.8 Wykorzystanie tekstury **X.Random.Dirt** w materiałach

Użycie komponentu **X.Random.Dirt** w schemacie materiału **B.Skin.Inner** przedstawia Rysunek 9.20.9a):



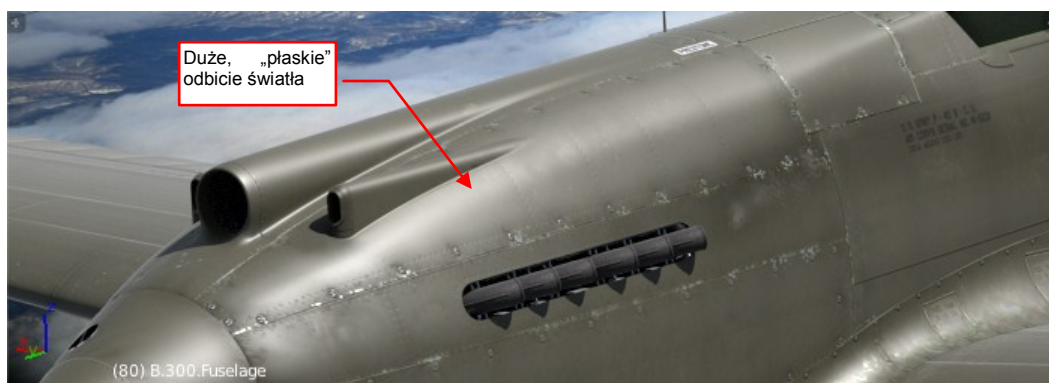
Rysunek 9.20.9 Przykład wykorzystania tekstury — schemat **B.Skin.Inner**

Jak widać, pojedyncze wyjście **Fac** z tekstury **X.Random.Dirt** jest rozdzielone na różne węzły regulujące, których rezultaty zmieniają połyskliwość i chropowatość shadera **Glossy Paint**. Aby nie tworzyć za każdym razem kolumny takich kontrolki, scalałem te wszystkie parametry w jedną grupę. Nadałem jej nazwę **X.Pattern.Adaptor**. Grupuje pary wartości (**Min** lub **Max** i **Range**) dla poszczególnych wyjść: **Diffuse**, **Specular**, **Roughness**, **Reflectivity**. Te wyjścia podłącz do odpowiednich wejść **Gloss Paint**.

Grupy **X.Pattern.Adaptor** opłaca się używać do rozdzielania sygnał z pojedynczej tekstury na trzy lub więcej kanały o różnej intensywności.

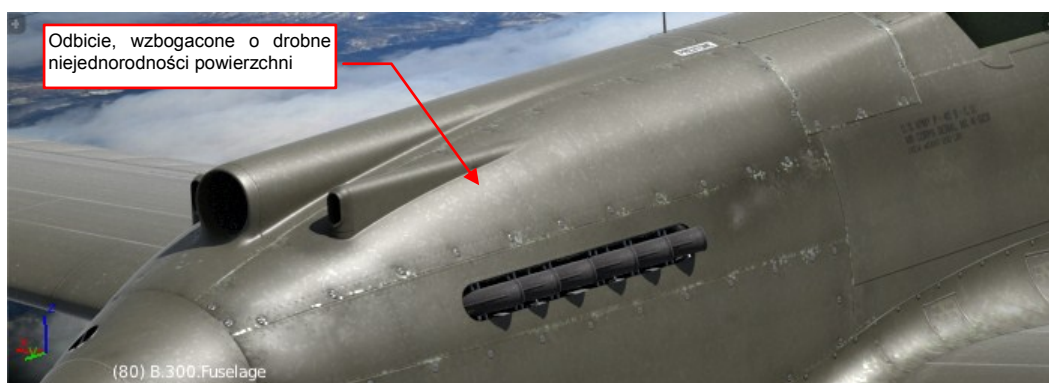


Materiał powierzchni zewnętrznych (**B.Skin.Camouflage**) przygotowany tak jak to zrobiliśmy w Wyciągu III, tworzy duże, jednolite odbicia światła (Rysunek 9.20.10):



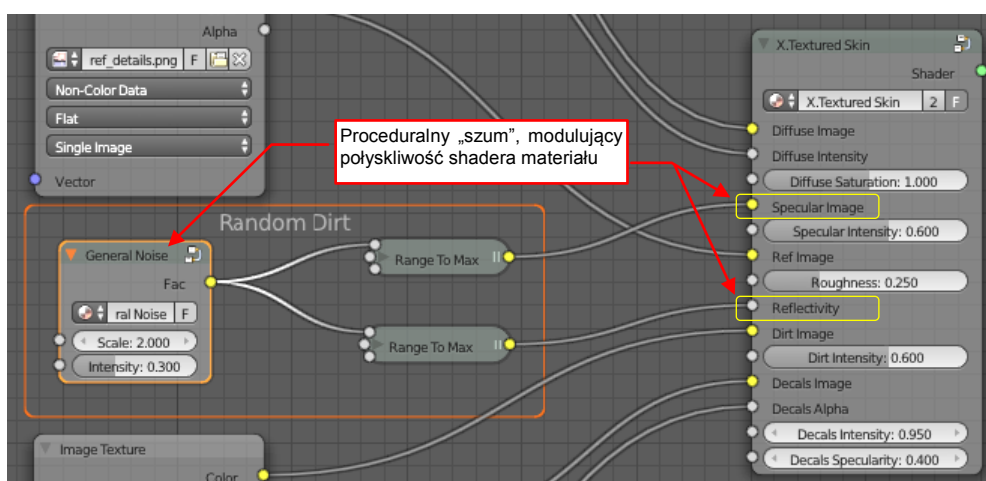
Rysunek 9.20.10 Odbicie światła na okapotowaniu silnika (materiał **B.Skin.Camouflage** jak do tej pory)

W zbliżeniu to wygląda trochę sztucznie: trzeba ten efekt „złamać” za pomocą drobnych niejednorodności w połyskliwości powierzchni (Rysunek 9.20.11):



Rysunek 9.20.11 Odbicie światła „złamane” za pomocą tekstury

Taki efekt można uzyskać podstawiając do schematu materiału **B.Skin.Camouflage** dodatkową teksturę losowych zabrudzeń. To tekstura proceduralna, modulująca obraz połyskliwości (*Specular Image*) i intensywność odbijania światła (*Reflectivity*) — por. Rysunek 9.20.12):

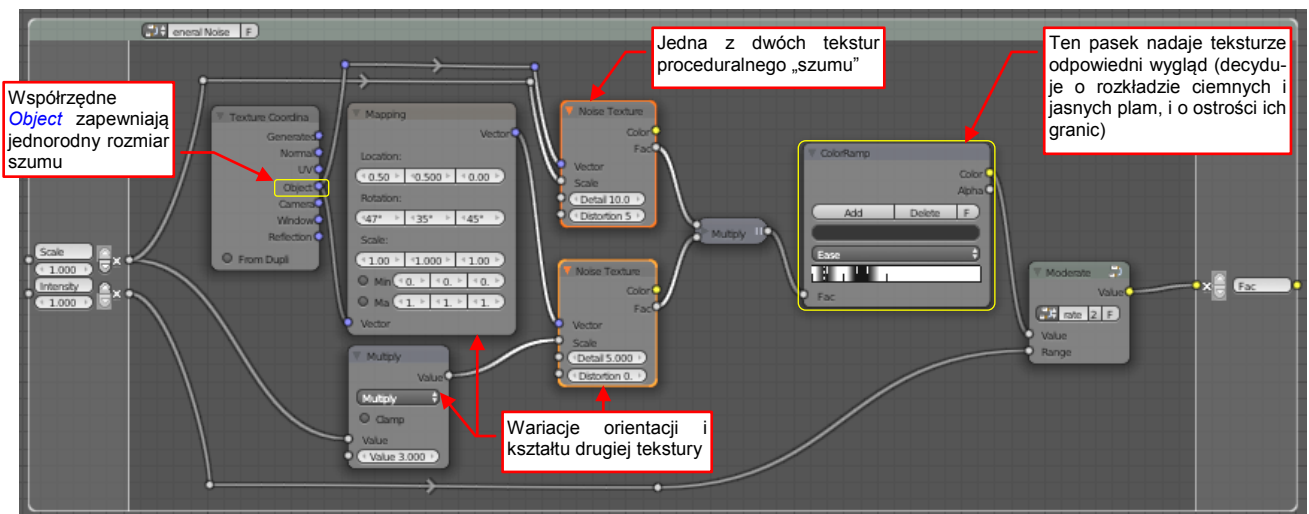


Rysunek 9.20.12 Modyfikacja schematu materiału **B.Skin.Camouflage**

Choć nazwałem ramkę widoczną na schemacie **Random Dirt**, to kształt plam generowanych przez komponent **X.Random.Dirt** niezbyt w tym miejscu pasuje. W dodatku jest bardzo kosztowny podczas renderowania. Dlatego przygotowałem do takich zastosowań inną grupę, o nazwie *General Noise* (Rysunek 9.20.12).



Grupa **General Noise** to złożenie rezultatów generowanych przez dwa węzły **Noise Texture**. Każdy z nich ma inne ustawienia (skalę, poziom detalu — Rysunek 9.20.13):



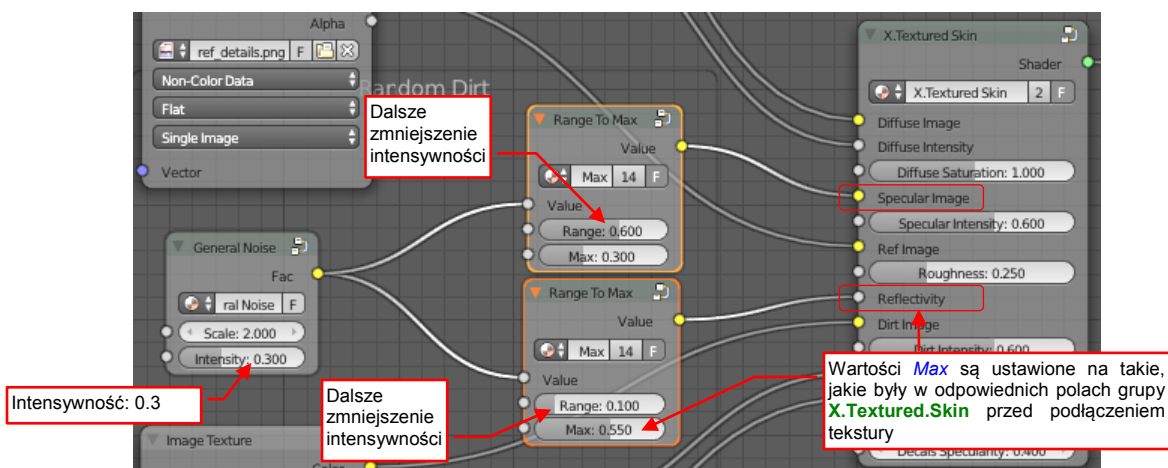
Rysunek 9.20.13 Alternatywny „szum” — grupa **General Noise**

Rysunek 9.20.14 przedstawia rezultat generowany przez tę grupę (włączoną na pełną intensywność):



Rysunek 9.20.14 Rezultat generowany przez **General Noise**

Efekt wygląda na bardzo silny, ale to dobrze — dużą moc zawsze można zmniejszyć, gorzej jest gdy jej brakuje. Aby stworzyć wrażenie drobnych nieregularności, **General Noise** użyte w materiale **B.Skin.Camouflage** jest ustawione na 30% swojej intensywności. Dodatkowo ta wartość ulega dalszemu osłabieniu w grupach sterujących obrazem odbić (**Specular Image**) i skalą połyskliwości (**Reflectivity**). Pokazuje to Rysunek 9.20.15:

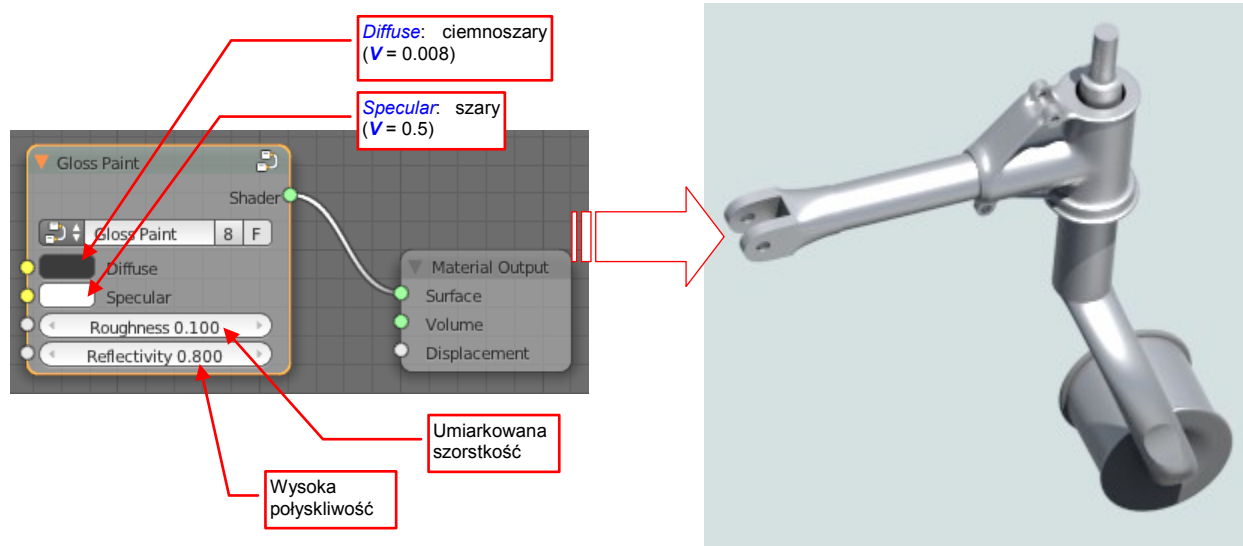


Rysunek 9.20.15 Szczegóły użycia tekstury **General Noise** w materiale **B.Skin.Camouflage**

Efekt tych ustawień na modelu pokazuje Rysunek 9.20.11. Czas renderowania wydłużył się o 20%.

## 9.21 Skonfigurowanie materiału: stal

Materiału, który wygląda jak stal, będziemy potrzebować do wykonania różnych drobnych detali samolotu. W zasadzie może to być prosty materiał oparty o dobrze nam znany shader *Gloss Paint* (Rysunek 9.21.1):

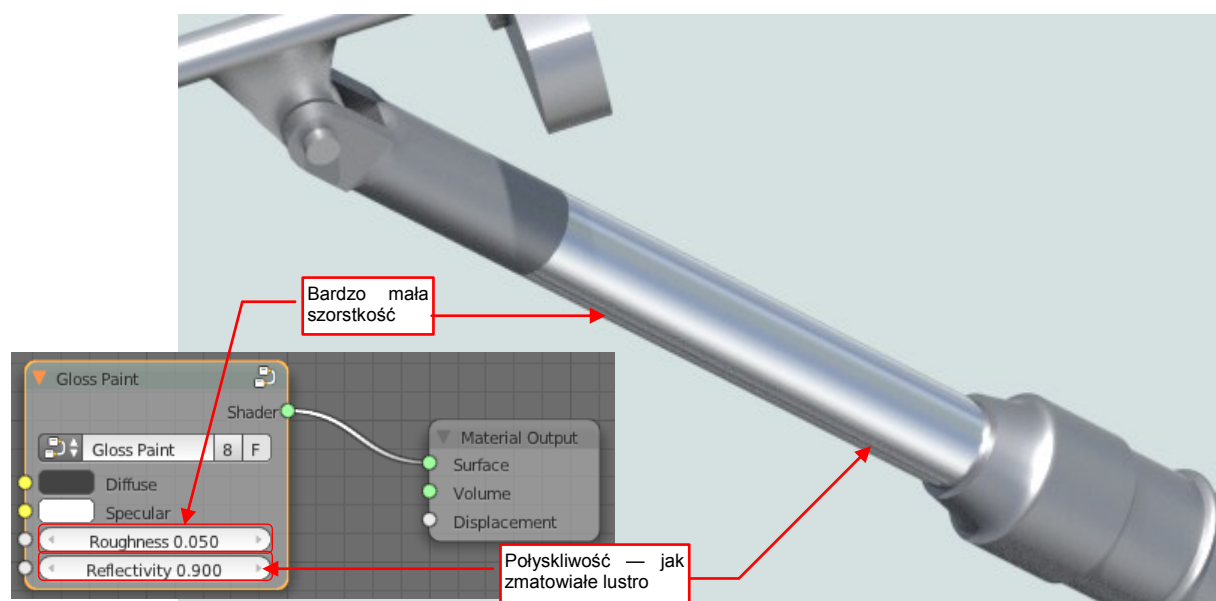


Rysunek 9.21.1 Najprostsza wersja: szary materiał o umiarkowanej połyskliwości

Nazwijmy ten materiał **B.Steel.Smooth**.

Co prawda istnieje fragment modelu, który wymaga specjalnego traktowania: jest to powierzchnia robocza tłoka amortyzatora. Podobne element możesz napotkać w różnego rodzaju siłownikach. Można je zobaczyć na co dzień w wielu miejscach — na przykład różnych dźwigach czy podnośnikach. Zauważ, że część tłoka która wsuwa się w cylinder jest w nich bardzo gładka. Jej powierzchnia przypomina lekko zamglone lustro.

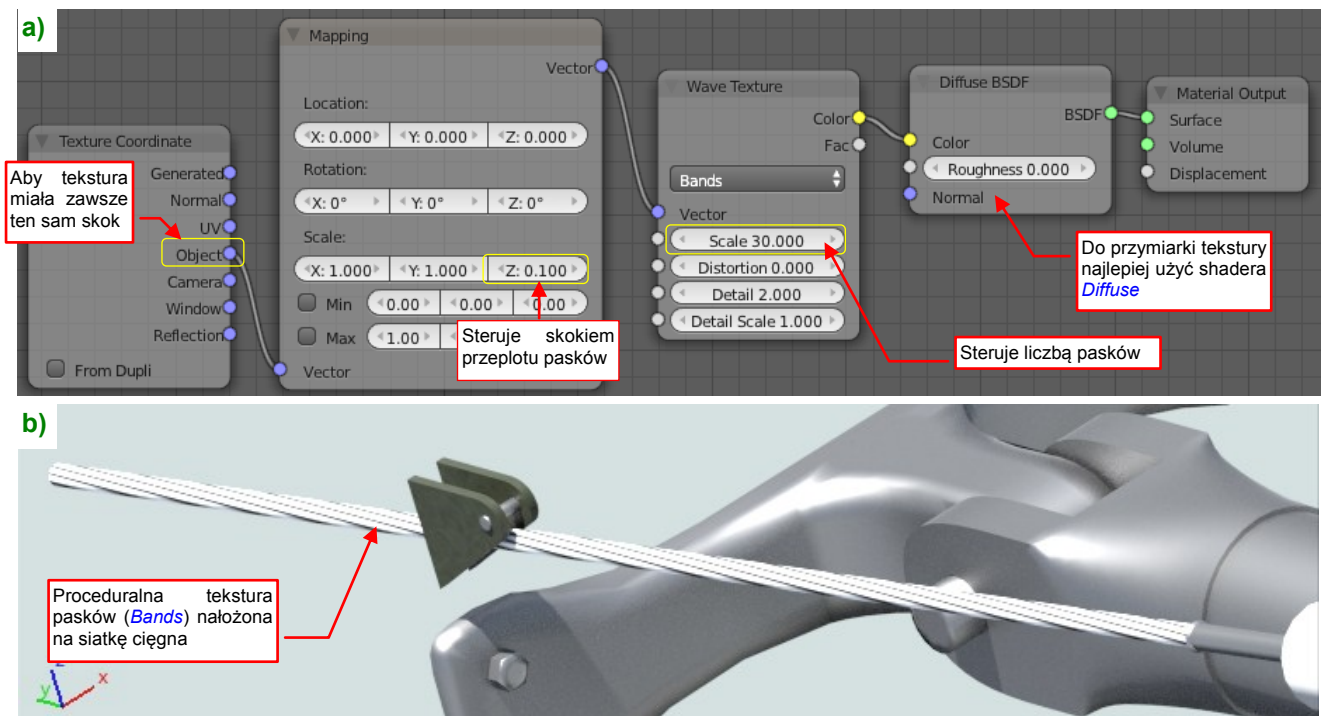
Aby odtworzyć ten efekt stworzymy oddzielny materiał — **B.Steel.Mirror** (Rysunek 9.21.2):



Rysunek 9.21.2 Specjalny materiał dla powierzchni pracującej tłoka amortyzatora

W tym materiale zmniejszyłem szorstkość powierzchni (*Roughness*) do 0.005 i zwiększyłem jej połyskliwość (*Reflectivity*) do 0.9.

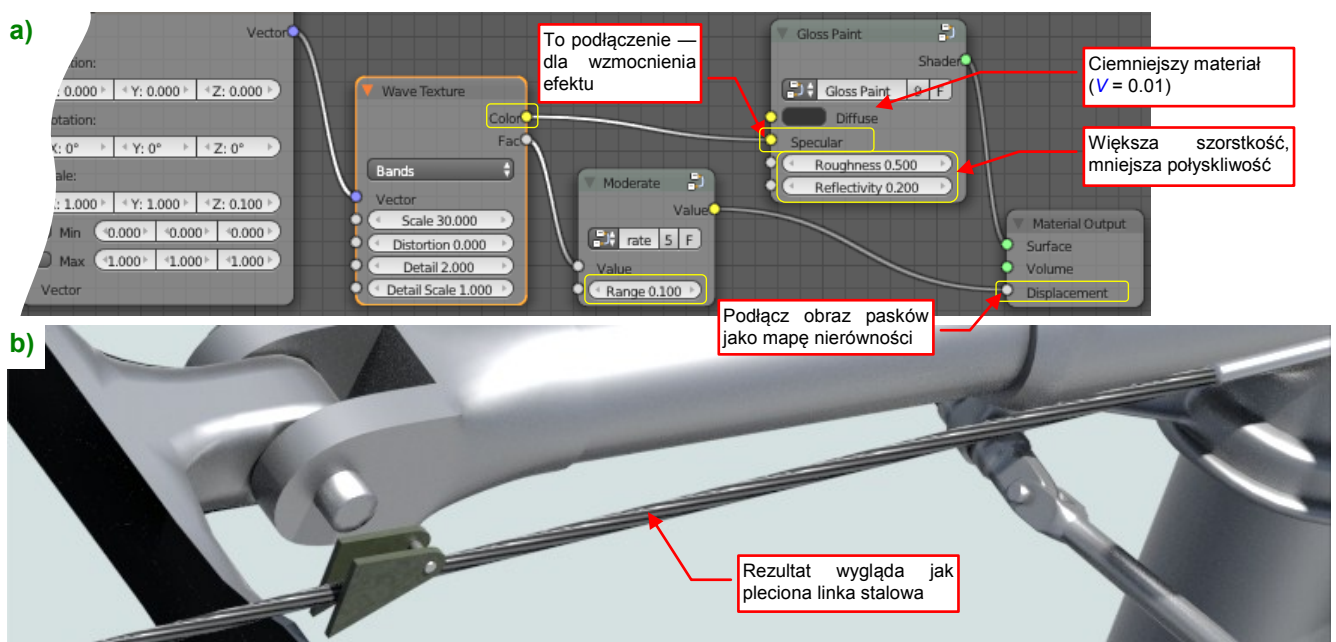
Innym specyficznym fragmentem wykonanym ze stali są plecione linki, wykorzystywane m.in. w ciągnach kółka ogonowego. Odwzorujemy je za pomocą tekstury nierówności. Wykorzystamy w tym celu teksturę *Wave:Bands* (por. str. 388). Wystarczy tylko ją trochę rozciągnąć i zwiększyć liczbę pasków (Rysunek 9.21.3):



Rysunek 9.21.3 Dopasowanie tekstury pasków do ciągną

Zaczynamy więc od materiału testowego: składa się ze zwykłego shadera *Diffuse*, do którego podłączamy naszą teksturę (Rysunek 9.21.3a). Aby zwiększyć liczbę pasków zwiększ parametr *Wave Texture:Scale*. Długością przeplotu możesz sterować zmieniając skalę tekstury wzdłuż osi ciągną (*Z*). Wreszcie mapowanie we współrzędnych *Texture Coordinate:Object* zapewnia, że ten sam materiał użyty np. dla bardzo długich linek anteny będzie miał taką samą długość przeplotu jaką pokazuje Rysunek 9.21.3b).

Teraz wystarczy tylko podstawić do tego schematu shader *Glossy Paint*, ustawiony jak dla ciemniejszego i bardziej szorstkiego metalu, i podłączyć tę teksturę jako mapę nierówności (Rysunek 9.21.4a):

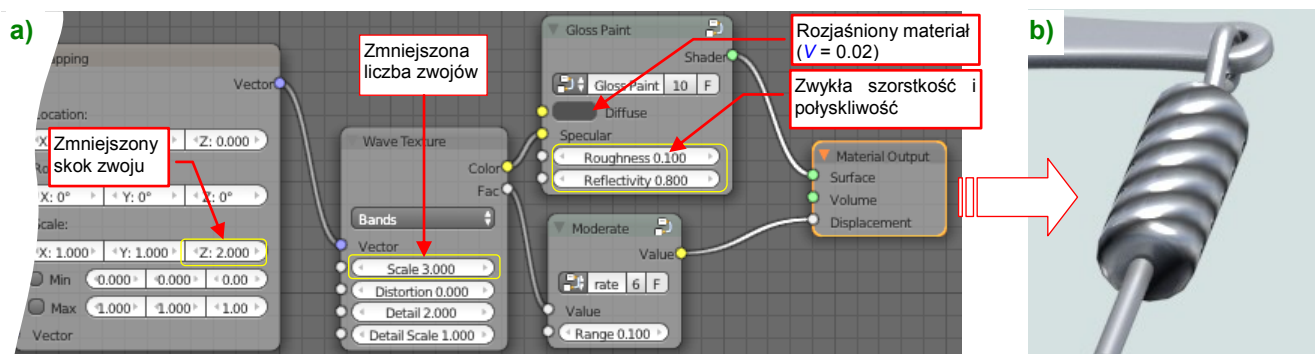


Rysunek 9.21.4 Podłączenie obrazu pasków jako tekstury nierówności



Nazwijmy ten materiał **B.Steel.Wire**. Zastosujemy go do innych metalowych linek stosowanych w samolocie, m.in. anteny radiowej. Każdy z nich jest modelowany jako podłużny cylinder. Należy tylko zwrócić uwagę, by w każdym takim elemencie oś siatki cylindra pokrywała się z lokalną osią **Z** obiektu!

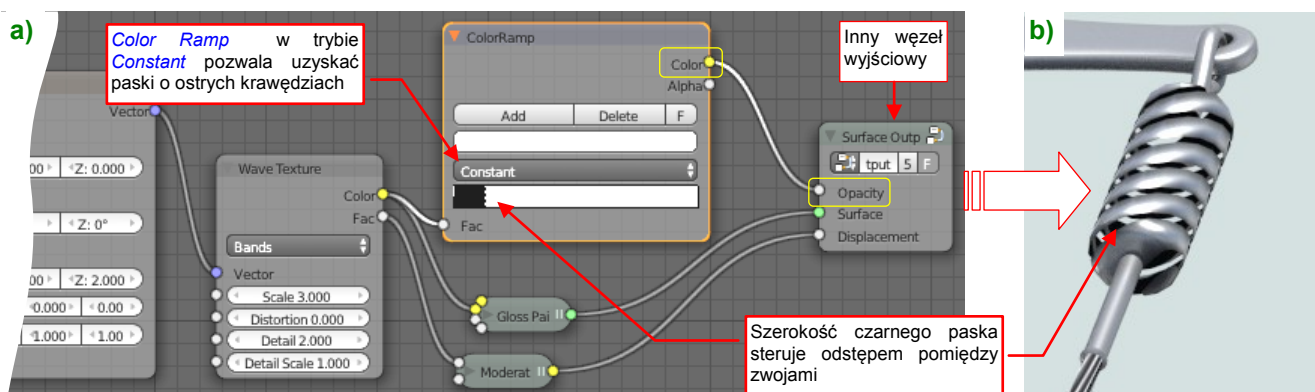
Podobne rozwiązanie wykorzystamy do stworzenia sprężyny na zakończeniu cięgna (Rysunek 9.21.5):



Rysunek 9.21.5 Wykorzystanie zmodyfikowanej tekstury pasków do odwzorowania sprężyny

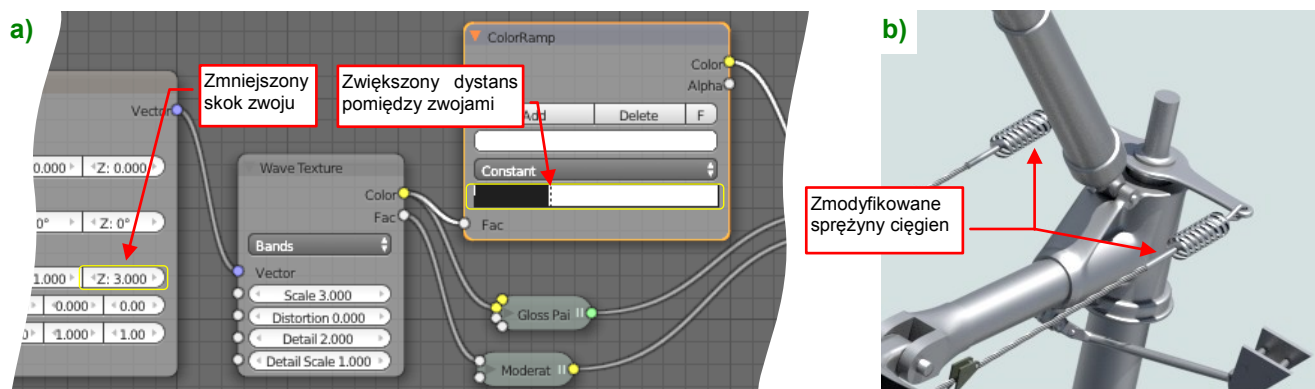
Nazwiemy ten materiał **B.Steel.Spring**. Początkowo skopiowałem tu węzły z **B.Steel.Wire**, po czym zmniejszyłem liczbę i skok zwoju oraz zwiększyłem jasność i połyskliwość (Rysunek 9.21.5a). Rysunek 9.21.5b) pokazuje uzyskany rezultat.

Ten efekt można jeszcze poprawić: wyprowadź z tekstury **Wave** drugą wersję obrazu, przetworzoną przez węzeł **Color Ramp** (uzyskując tą drogą ostre krawędzie pasków — por. str. 386). Aby użyć jej jako mapy przejrzystości, zastąp domyślny węzeł wyjściowy węzłem grupy **Surface Output** (por. Wyciąg III) (Rysunek 9.21.5):



Rysunek 9.21.6 Wykorzystanie obrazu pasków jako tekstury nieprzejrzystości

Potem pozostaje tylko obejrzeć nasz element z większej odległości, by lepiej dobrać: skok zwojów sprężyny i szerokość odstępów pomiędzy zwojami (Rysunek 9.21.7a):



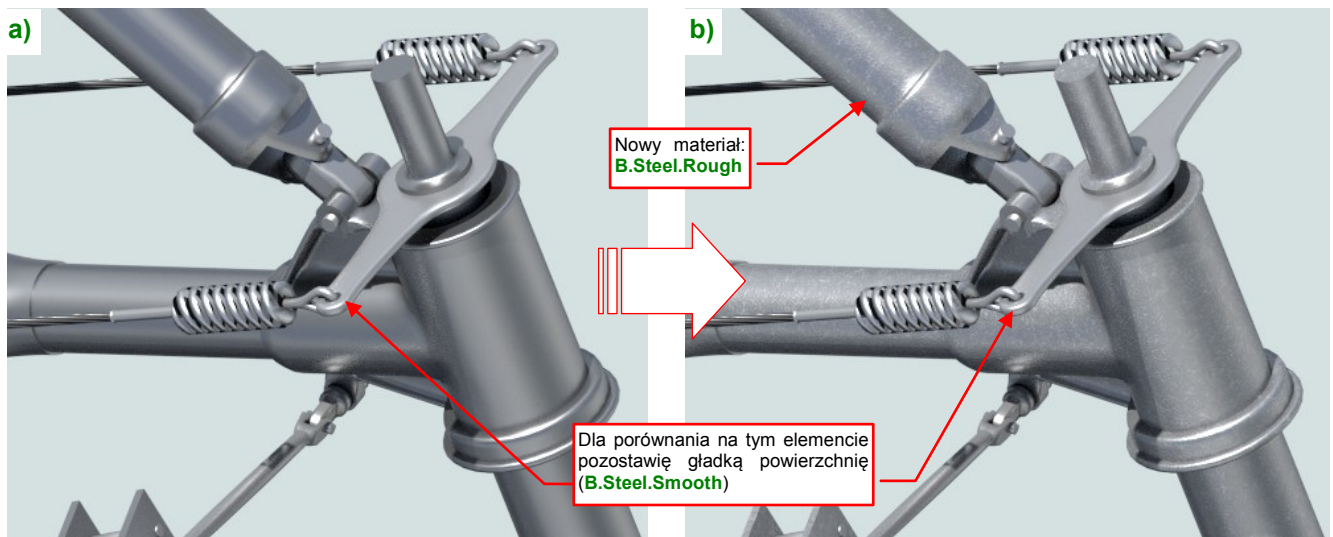
Rysunek 9.21.7 Dostosowanie efektu do widoku z większej odległości

Rysunek 9.21.7b) przedstawia uzyskany efekt. Sądzę że te sprężyny wyglądają na nim wystarczająco dobrze.



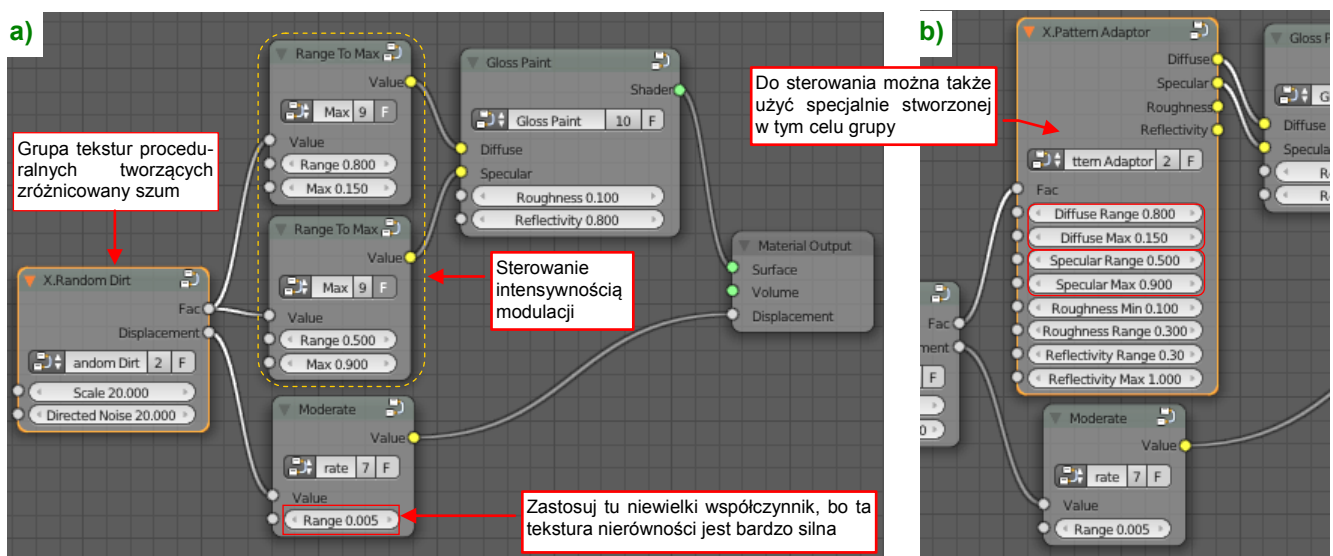
Przedstawiona w materiale **B.Steel.Spring** metoda jest dobrym rozwiązaniem przy tworzeniu różnych drobnych detali. Możesz w ten sposób szybko stworzyć elementy, które nie mają być specjalnie widoczne lub będą zazwyczaj bardzo małe.

Do tej pory stworzyliśmy efekt powierzchni metalu za pomocą shadera **Glossy Paint**, w prostym schemacie jaki pokazuje Rysunek 9.21.1 (str. 396). Uzyskana powierzchnia sprawia wrażenie bardzo gładkiej (Rysunek 9.21.8a) — jak gdyby była pokryta niklem lub innym drogim wykończeniem. Większe elementy podwozia były odlewane lub kute, i powinny mieć bardziej zróżnicowaną powierzchnię — na przykład taką, jaką przedstawia Rysunek 9.21.8b):



Rysunek 9.21.8 Zróżnicowanie powierzchni metalu

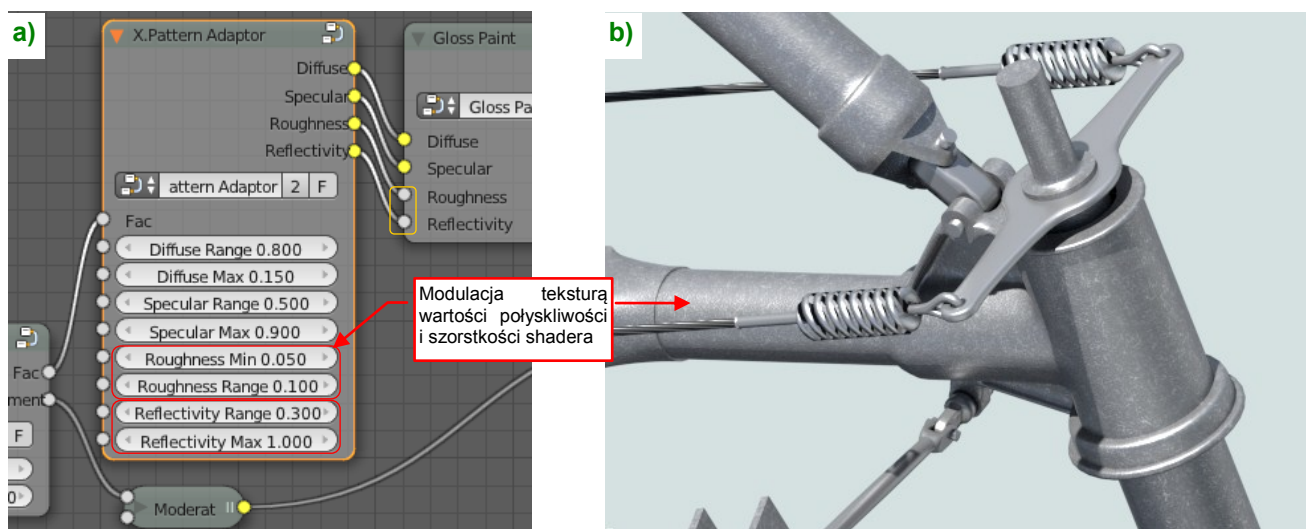
Do odwzorowania takich powierzchni stworzymy nowy materiał: **B.Steel.Rough**. Jego schemat przedstawia Rysunek 9.21.9a). Do modulacji barw **Diffuse** i **Specular** używam w tej złożonej teksturze szumu — komponentu **X.Random Dirt** (opis budowy tej tekstury znajdziesz na str. 390). Drugiego wyjścia z tej grupy — obrazu **Displacement** — używam jako tekstury tworzącej wrażenie drobnych chropowatości powierzchni.



Rysunek 9.21.9 Materiał **B.Steel.Rough**

Zastosowanie tekstur proceduralnych pozwala uzyskać ciekawe efekty bez konieczności wykonywania pracochłonnych rozwinięć **UV**. Ceną za to jest dłuższy czas renderowania: obraz pokazywany przez Rysunek 9.21.8a) został stworzony w 3min 12s, podczas gdy Rysunek 9.21.8b) — w 6min 40s.

Silniejszy efekt można uzyskać podłączając do tej samej tekstury „szumu” pozostałe współczynniki shadera *Gloss Paint*: **Roughness** i **Reflectivity** (Rysunek 9.21.10a):

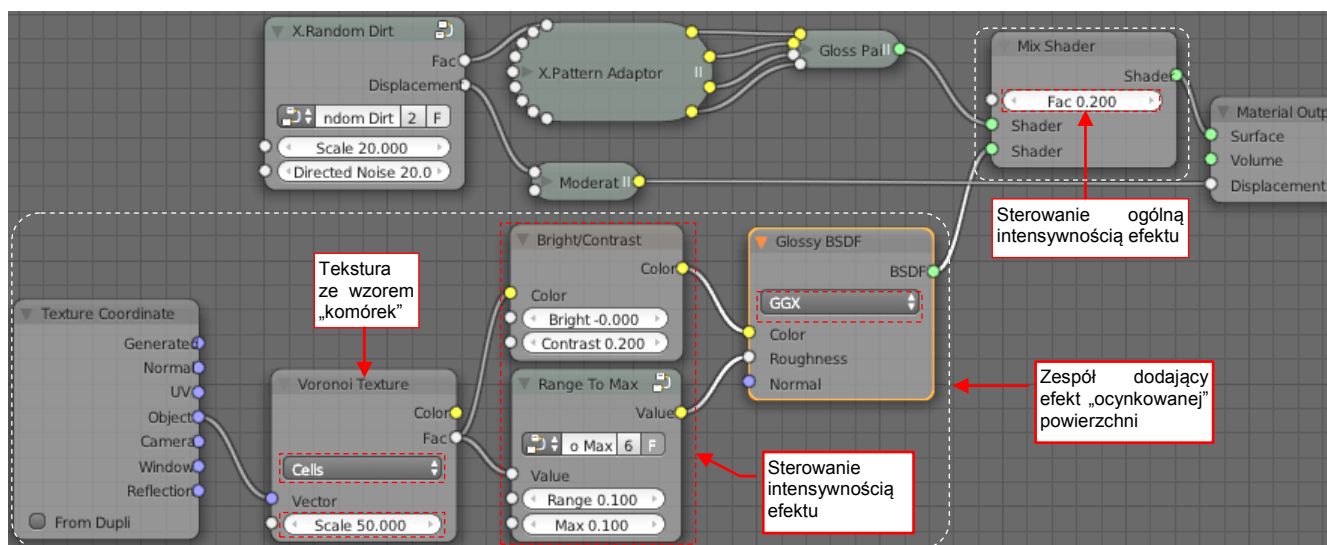


Rysunek 9.21.10 Bardziej szorstki materiał

W tym przypadku zastosowanie grupy **X.Pattern Adaptor** pozwala przedstawić wszystkie parametry w bardziej zwartej formie, upraszczając w sposób znaczący cały schemat. Aby uzyskać efekt pokazywany przez Rysunek 9.21.10b) ustaw wartości wejściowe tej grupy tak, jak pokazuje ilustracja. Potem możesz trochę poeksperymentować z ustawieniami. Zwróć uwagę, że nawet drobne zmiany parametrów sterujących wyjściem **Roughness** (**Roughness Min**, **Roughness Range**) wywołują znaczne różnice w wyglądzie uzyskanej powierzchni.

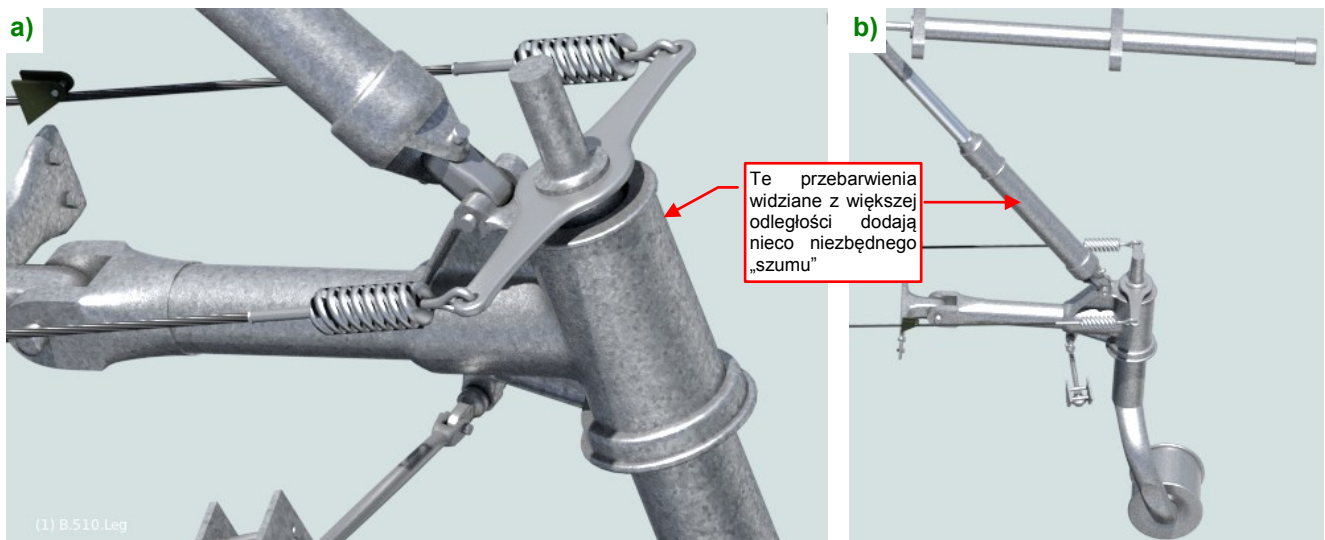
Oczywiście, ten materiał można poddać kolejnym zmianom. Na przykład — w celu zabezpieczenia przed korozją powierzchnie stalowe są często pokrywane cienką warstwą cynku. Pojawia się na nich charakterystyczny wzór obszarów o różnym odcieniu, przypominający teksturę *Voronoi:Cells* (por. str. 345). Jednocześnie powierzchnię którą pokazuje Rysunek 9.21.10b) można wzbogacić o trochę połysku.

Stąd proponuję dodać do schematu materiału **B.Steel.Rough** drugą gałąź, w której monochromatyczny obraz „komórek” uzyskany z wyjścia *Voronoi:Fac* jest podłączony do barwy i szorstkości shadera typu *Glossy*. Każdy z tych obrazów jest modulowany w inny sposób — np. barwa jest wzmacniana za pomocą węzła *Bright/Contrast*. Całość połączona z dotychczasowym wyjściem z *Gloss Paint* za pomocą nowego węzła *Mix Shader* (Rysunek 9.21.11):



Rysunek 9.21.11 Dodatkowa tekstura — „ocynk”

Rysunek 9.21.12a) przedstawia rezultat. Dodanie kolejnej tekstury proceduralnej wydłużyło czas renderu do 7min 57s. Z niewielkiej odległości efekt wydaje się być nieco przesadzony (możesz to skorygować, zmniejszając wartość `MixShader:Fac` do 0.1). Po zwiększeniu odległości kamery od obiektu te komórki tekstury `Voronoi` zlewają się w drobne kropki (Rysunek 9.21.12b):



**Rysunek 9.21.12** Efekt na podglądzie renderu

Z większej odległości może to wyglądać na kurz lub brud — w każdym razie efekt jest ciekawy. Zdecydowałem pozostawić go w tym materiale. Gdy obiekt pokryty teksturami proceduralnymi zajmuje niewielką część obrazu, ich wpływ na ogólny czas renderowania jest niewielki. Takie tekstury wpływają zazwyczaj na zbliżenia: Rysunek 9.21.12b) generował się trzy razy szybciej (2min 36s) niż Rysunek 9.21.12a).

- Złożonych tekstur proceduralnych warto używać dla detali, które nie zajmują dużej powierzchni na ostatecznym renderze.

## 9.22 Skonfigurowanie materiału: guma

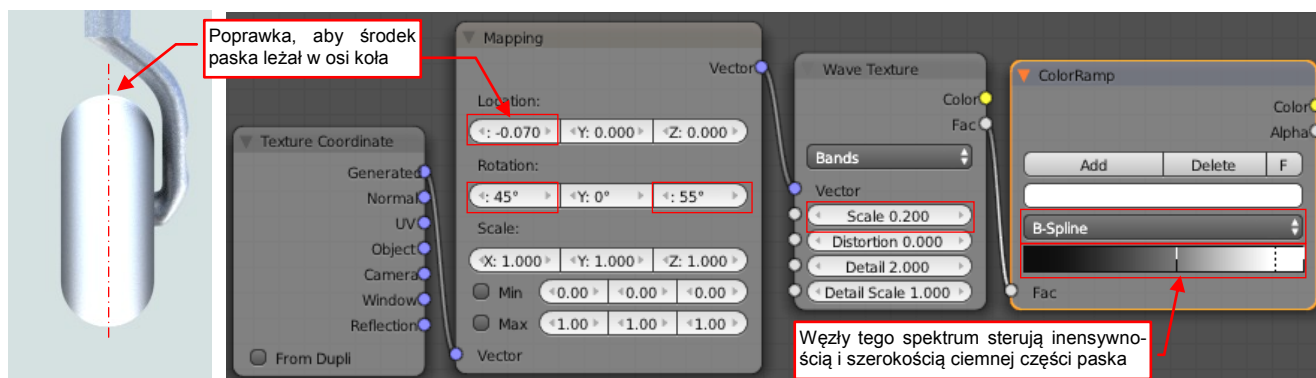
W tej sekcji pokażę krok po kroku, jak uzyskać materiał sprawiający wrażenie „zakurzonego kawałka gumy” — taki, jak opona tego kółka ogonowego (Rysunek 9.22.1):

Opony we wczesnych wersjach P-40 albo nie miały w ogóle bieżnika, albo bieżnik miał prosty kształt podłużnych rowków. Jeżeli zastosujemy dla wszystkich tekstur materiału mapowanie *Texture Coordinate:Generated* (por. str. 384), będzie można ten sam materiał wykorzystać i dla opony kółka ogonowego, i dla opon kół głównych. Dlatego proponuję nazwać ten materiał **B.Rubber.Tyre**. W tej sekcji pokażę krok po kroku jak go stworzyć za pomocą tekstur proceduralnych. Sprowadza się to do skomponowania z nich kilku obrazów, które posłużą nam do dodania takich efektów jak bieżnik, przetarcia, kurz.



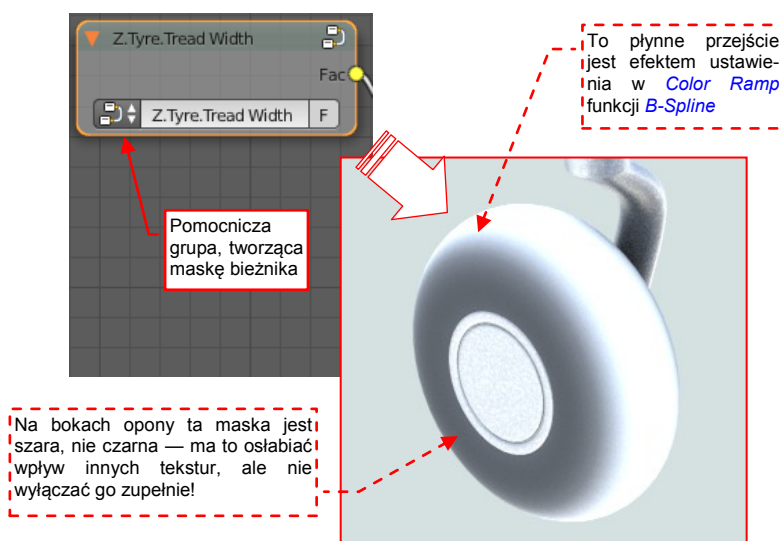
Rysunek 9.22.1 Zakurzona opona

Najbardziej zużyta częścią opony jest bieżnik. Zaczniemy od stworzenia białego pasa, który wyznacza jego granice. Ten obraz posłuży nam do tonowania pozostałych efektów zużycia po bokach opony (Rysunek 9.22.2):



Rysunek 9.22.2 Pojedynczy, szeroki pasek — maska bieżnika

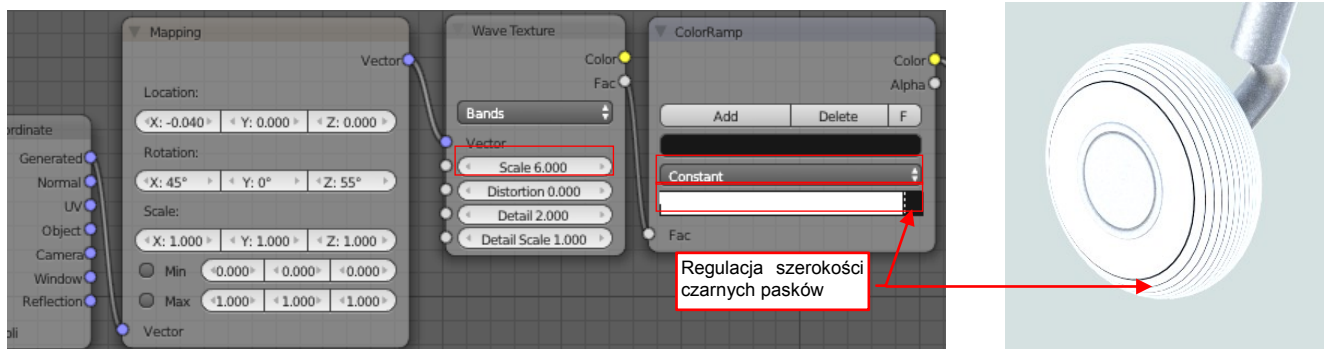
Posłużyłem się tu odpowiednio zmapowaną teksturą *Wave* (por. str. 388). Pozostaawiłem wzdłuż osi opony biały pas, podczas gdy obszar poza bieżnikiem jest ciemniejszy (ale nie czarny!). Do sterowania intensywnością tych odcieni szarości użyłem dodatkowego węzła *Color Ramp* (por. str. 386). Aby lepiej pokazać użycie na schemacie materiału tych i następnych obrazów, przekształcę je w tymczasowe grupy. Zespół węzłów tej maski bieżnika połączyłem w grupę **Z.Tyre.Tread Width** (Rysunek 9.22.3):



Rysunek 9.22.3 Pomocnicza tekstura (grupa) — **Z.Tyre.Tread Width**

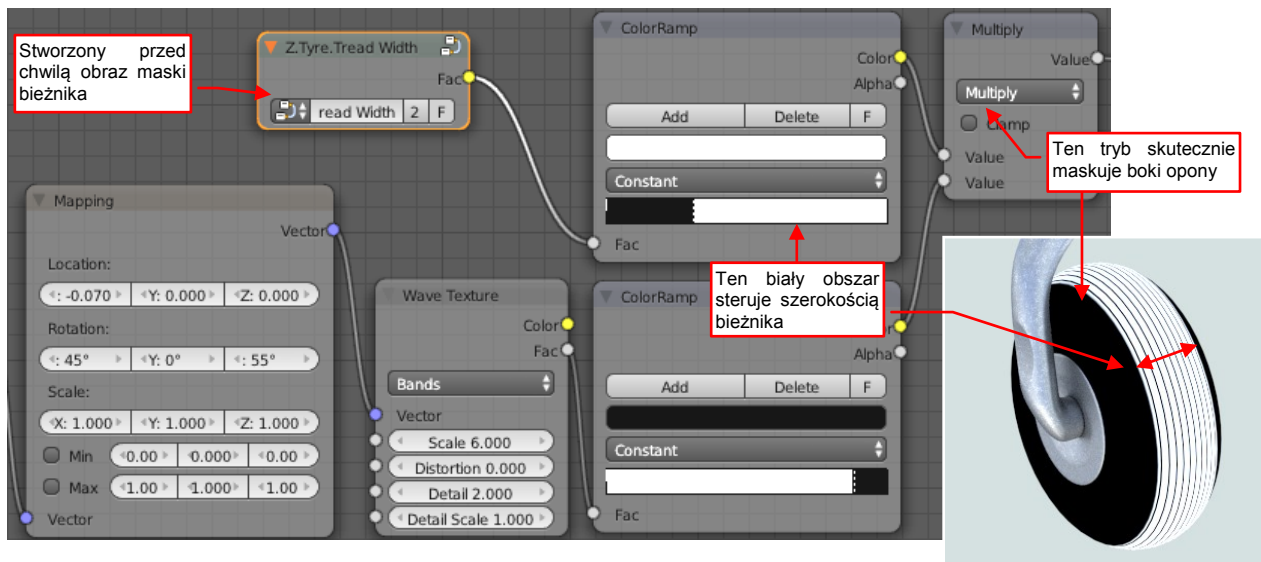


Kolejną teksturą będzie wzór bieżnika opony. Zakładam, że we wczesnych wersjach P-40 był to prosty wzór podłużnych pasków. Uzyskamy go w bardzo podobny sposób, co maskę bieżnika: za pomocą tekstury **Wave**. Wystarczy nadać jej nieco inną skalę i zmodyfikować spektrum węzła **Color Ramp** (Rysunek 9.22.4):



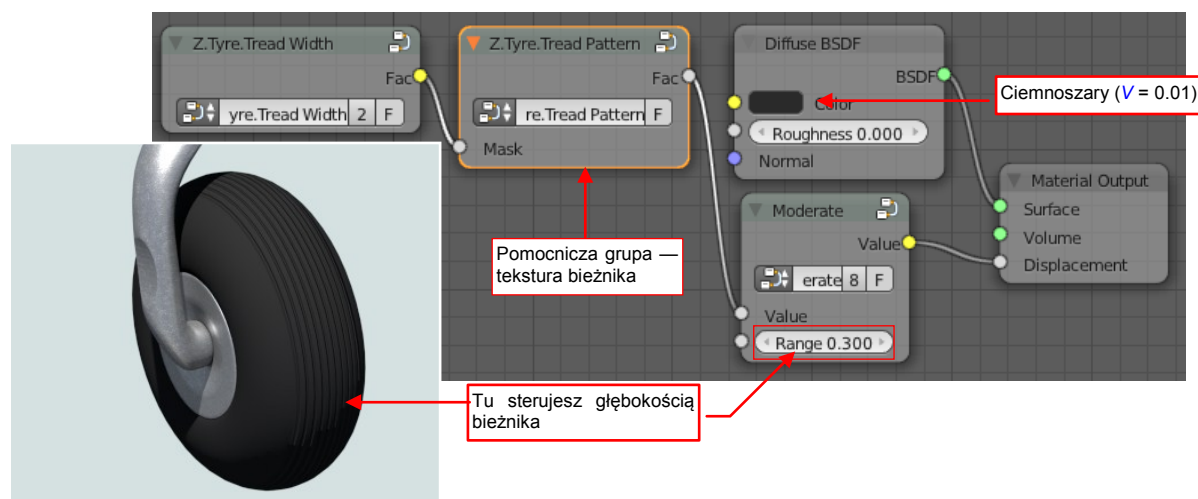
Rysunek 9.22.4 Tekstura bieżnika

Zwróć uwagę, że w węźle **Color Ramp** zastosowałem tu inną funkcję: **Constant**. W ten sposób uzyskujemy ostre granice barw pasków. Podobny węzeł użyjemy, aby za pomocą maski bieżnika (**Z.Tyre.Tread Width**) zamaskować boki opony (Rysunek 9.22.5):



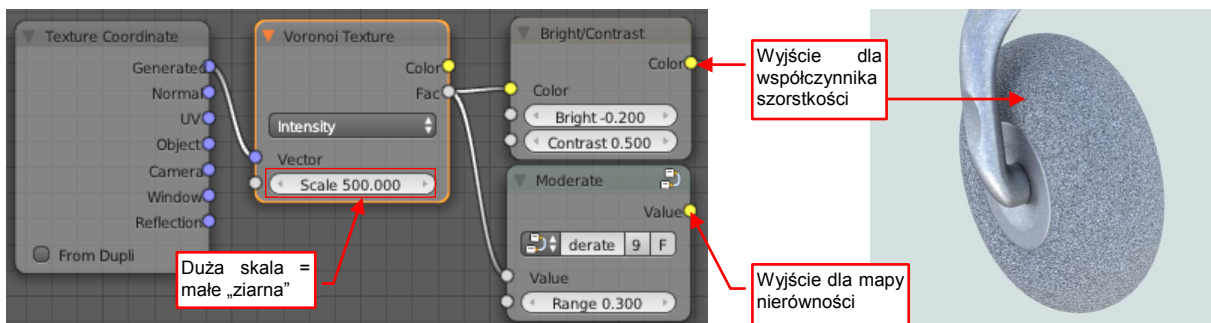
Rysunek 9.22.5 Tekstura bieżnika (po zamaskowaniu boków)

Węzły wzoru bieżnika łączymy w nową grupę o nazwie **Z.Tyre.Tread Pattern**. Rysunek 9.22.6 przedstawia użycie obydwu grup w pierwszej wersji materiału (pełnią rolę mapy nierówności):



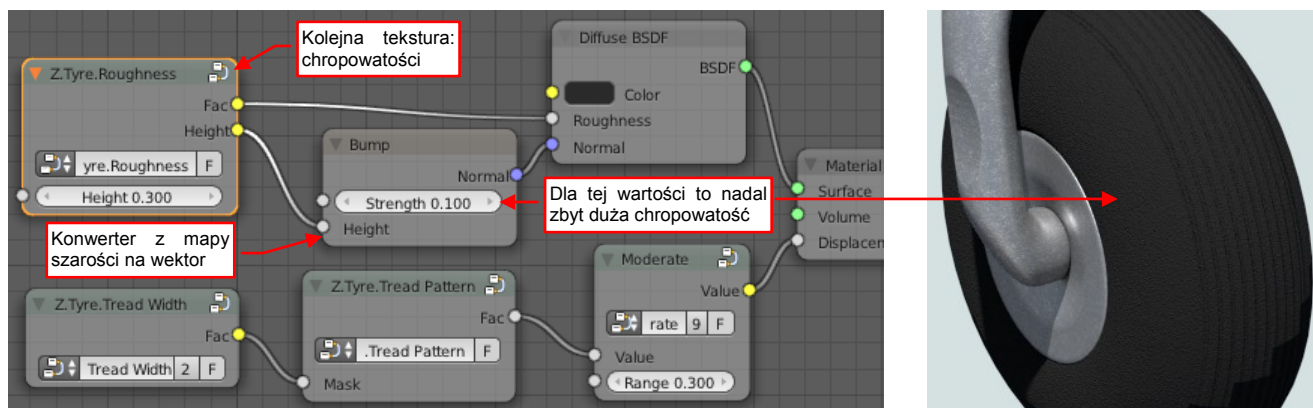
Rysunek 9.22.6 Tekstura bieżnika (mapa nierówności) w pierwszej wersji materiału opony

Teraz należy nadać naszej oponie jakąś bardziej widoczną szorstkość. Wykorzystam w tym celu teksturę **Voronoi** w trybie **Intensity** (por. str. 344). Powiększ wartość jej skali (**Scale**) do 400 — 500, aby uzyskać coś co przypomina ziarna piasku (Rysunek 9.22.7):



Rysunek 9.22.7 Tekstura chropowatości

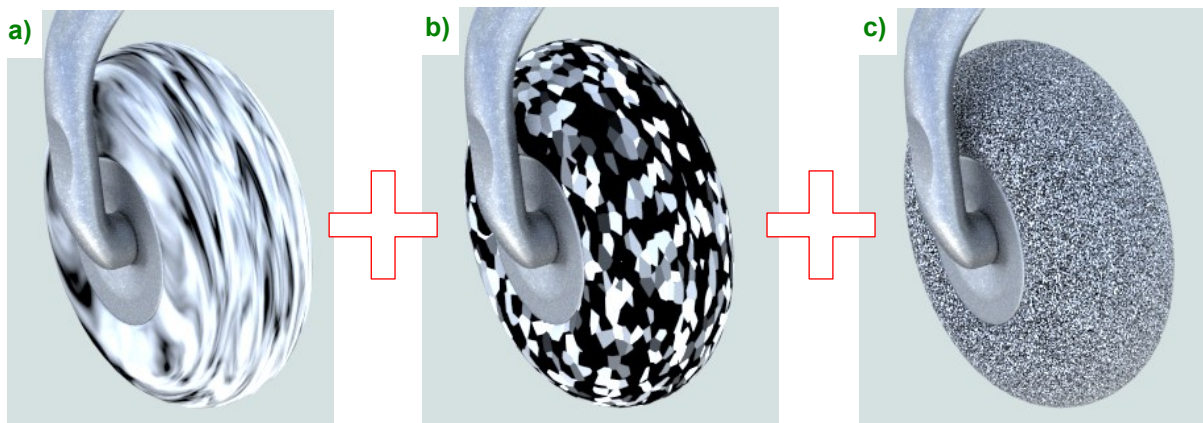
Ten wzór wykorzystamy jako teksturę nierówności oraz do modulacji szorstkości materiału opony. W każdym z tych miejsc musi mieć inną intensywność. Do podkreślenia ziaren na obrazie szorstkości wykorzystuję węzeł **Bright/Contrast** (znajdziesz go w submenu **Add → Color**). Do złagodzenia intensywności mapy nierówności użyłem grupy **Moderate** (Rysunek 9.22.7). Węzły połączyłem w grupę **Z.Tyre.Roughness**, która ma dwa wyjścia: **Fac** (współczynnik szorstkości) i **Height** (dla mapy nierówności). Użycie tej grupy w materiale opony przedstawia Rysunek 9.22.8:



Rysunek 9.22.8 Wykorzystanie tekstury chropowatości

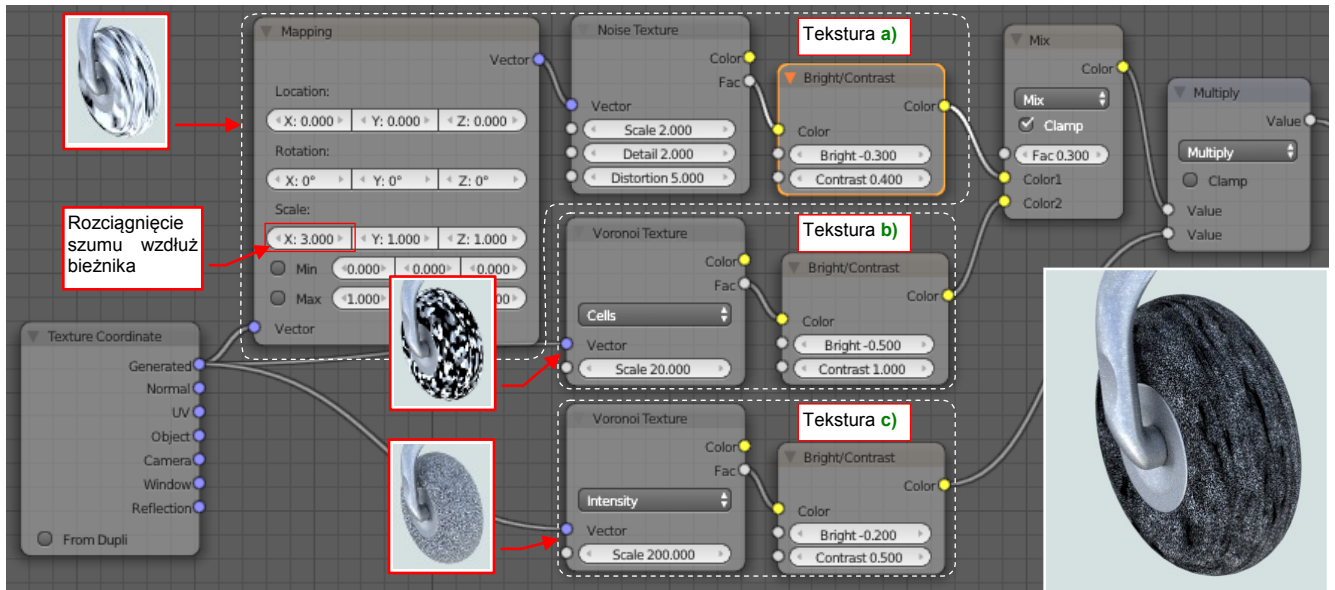
Zwróć uwagę, że wyjście **Height** podłączyłem do wejścia **Normal** shadera **Diffuse**. Zrobiłem to na zasadzie eksperymentu (Wejście **Normal** służy do alternatywnego odwzorowania nierówności za pomocą tzw. mapy normalnych). Węzeł **Bump** dokonuje tu automatycznej konwersji mapy szarości na wektory normalne.

Aby odwzorować rozkład kurzu i brudu na oponie, połączymy trzy różne tekstury (Rysunek 9.22.9):



Rysunek 9.22.9 Składniki tekstury zabrudzeń

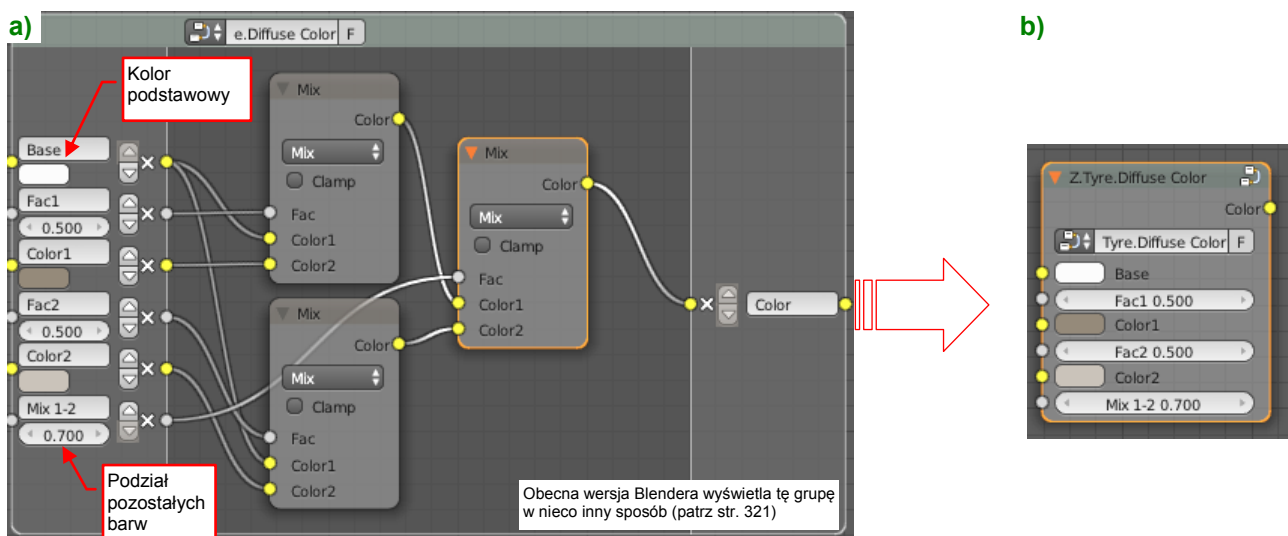
Rysunek 9.22.10 przedstawia szczegóły tych trzech tekstur oraz sposób i rezultat ich złożenia:



Rysunek 9.22.10 Złożenie tekstury zabrudzeń

Tekstura **a**) to obraz **Noise**, „spłaszczony” wzdłuż osi opony (**Mapping:Scale** = 3) w celu uzyskania efekt rozciągniętych plam (wzdłuż kierunku obrotu koła). Tekstura **b**) to klasyczny monochromatyczny wzór **Voronoi:Cells** (por. str. 345), z odpowiednio zwiększonym kontrastem. Tekstury **a**) i **b**) są połączone węzłem **Mix**, którego współczynnik **Fac** steruje udziałem „komórek” tekstury **b**) w ostatecznym obrazie. Uzyskujemy w ten sposób pewien losowy wzór, który będzie w miarę dobrze pasować na zabrudzenia opony. Aby całość nie wyglądała zbyt regularnie, łączę ją jeszcze z teksturą **c**). To typowa „drobnoziarnista” **Voronoi:Intensity** (por. str. 344). Jest łączona z sumą tekstur **a**) i **b**) operacją **Multiply**, przez co ostateczny obraz złożenia jest dość ciemny (Rysunek 9.22.10). Ten zespół węzłów połączyłem w pomocniczą grupę o nazwie **Z.Tyre.Color Pattern**.

Aby przekształcić odcienie szarości zwracane przez grupę **Z.Tyre.Color Pattern** w coś, co będzie przypominać kurz i pył, potrzebujemy jeszcze jednej grupy. Nazwałem ją **Z.Tyre.Diffuse Color**. Po połączeniu do przygotowanych wcześniej grup obrazów stworzy wrażenie zabrudzeń o odpowiednich kolorach (Rysunek 9.22.11):

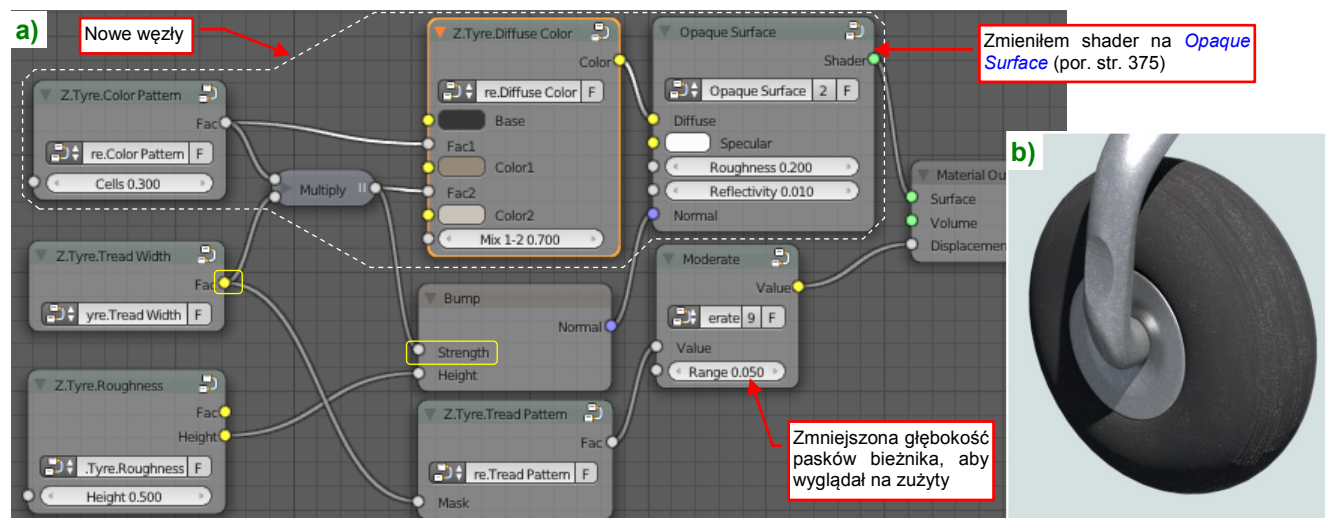


Rysunek 9.22.11 Grupa dostarczająca barwy **Diffuse**

Grupa **Z.Tyre.Diffuse Color** to złożenie trzech węzłów „mieszących” barwę podstawową **Base** (ustaw ją na ciemnoszary, tzn. **V** = 0.01) z jaśniejszymi kolorami pozostałych zabrudzeń (Rysunek 9.22.11a). Kolory tych zabrudzeń nazwałem **Color1** i **Color2** (Rysunek 9.22.11b). Dobierz je odpowiednio do tworzonej sceny!



Rysunek 9.22.12a) przedstawia zmodyfikowany materiał opony. Najpierw wymieniałem używany do tej pory shader na **Opaque Surface** (to to samo co **Gloss Paint**, ale z wejściem dla mapy normalnych — por. str. 375). Następnie włączyłem do tego schematu przygotowane grupy odpowiedzialne za efekt kurzu — **Z.Tyre.Color Pattern** i **Z.Tyre.Diffuse Pattern**. Ich rezultat jest podłączany do barwy **Diffuse** shadera:

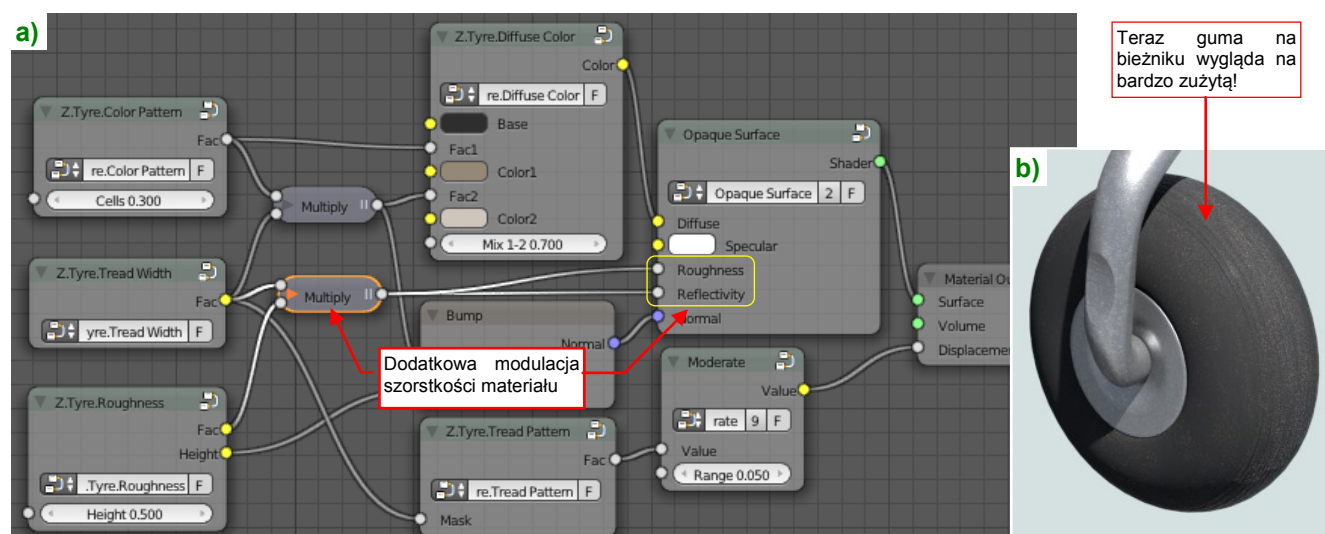


Rysunek 9.22.12 Połączenie wszystkich komponentów materiału

Zwróć uwagę na sposób połączenia tekstury zabrudzeń (**Z.Tyre.Color Pattern**) z grupą barw **Z.Tyre.Diffuse Color**. Tekstura steruje udziałem barw kurzu (**Color1** i **Color2**) w poszczególnych miejscach opony. Jedno z jej wyjść jest podłączone bezpośrednio do udziału ciemniejszej barwy (**Color1**). Drugie — najpierw jest ograniczane (węzeł **Multiply**) do obszaru bieżnika, a dopiero potem uaktywnia jaśniejszą barwę zabrudzeń (**Color2**). (W ten sposób zabrudzenia na bieżniku są jaśniejsze).

Dla dodatkowego wzmocnienia efektu tak zmodyfikowany obraz jest podłączony do wejścia **Bump:Strenght**, różnicując w ten sposób drobne nierówności powierzchni. Zmniejszyłem także głębokość bieżnika (do 0.05 — por. ze str. 403, Rysunek 9.22.6), aby wyglądał na bardzo zużyty. Rysunek 9.22.12b) przedstawia uzyskany rezultat.

Aby uzyskać jeszcze ciekawszy efekt zużycia, możesz połączyć obrazy **Z.Tyre.Roughness** i **Z.Tyre.Tread Width** w dodatkowym węźle **Multiply** (czyli skupić efekt na obszarze bieżnika). Rezultat podłącz do wejść **Roughness** i **Reflectivity** shadera (Rysunek 9.22.13a):

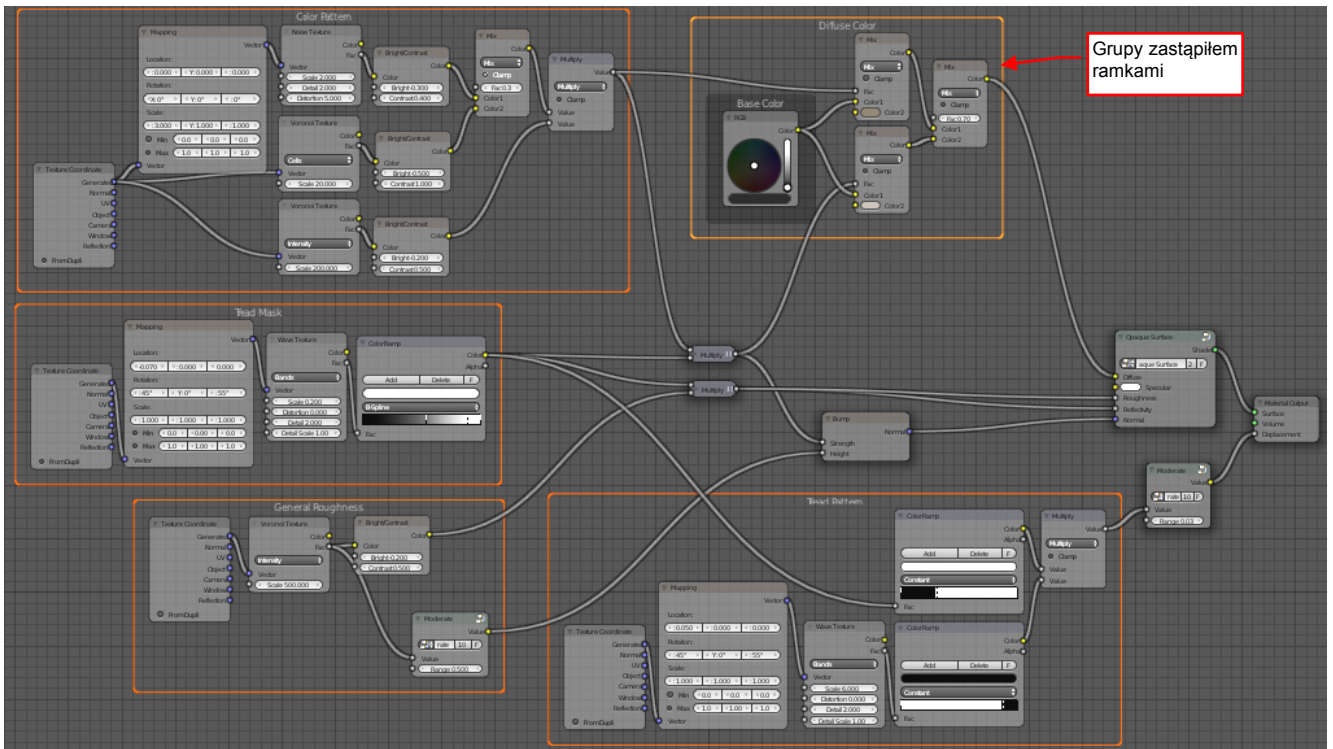


Rysunek 9.22.13 Dodatkowa modulacja szorstkości bieżnika

W rezultacie guma bieżnika stała się tak zużyta, że jest miejscami lekko połyskliwa (Rysunek 9.22.13b).

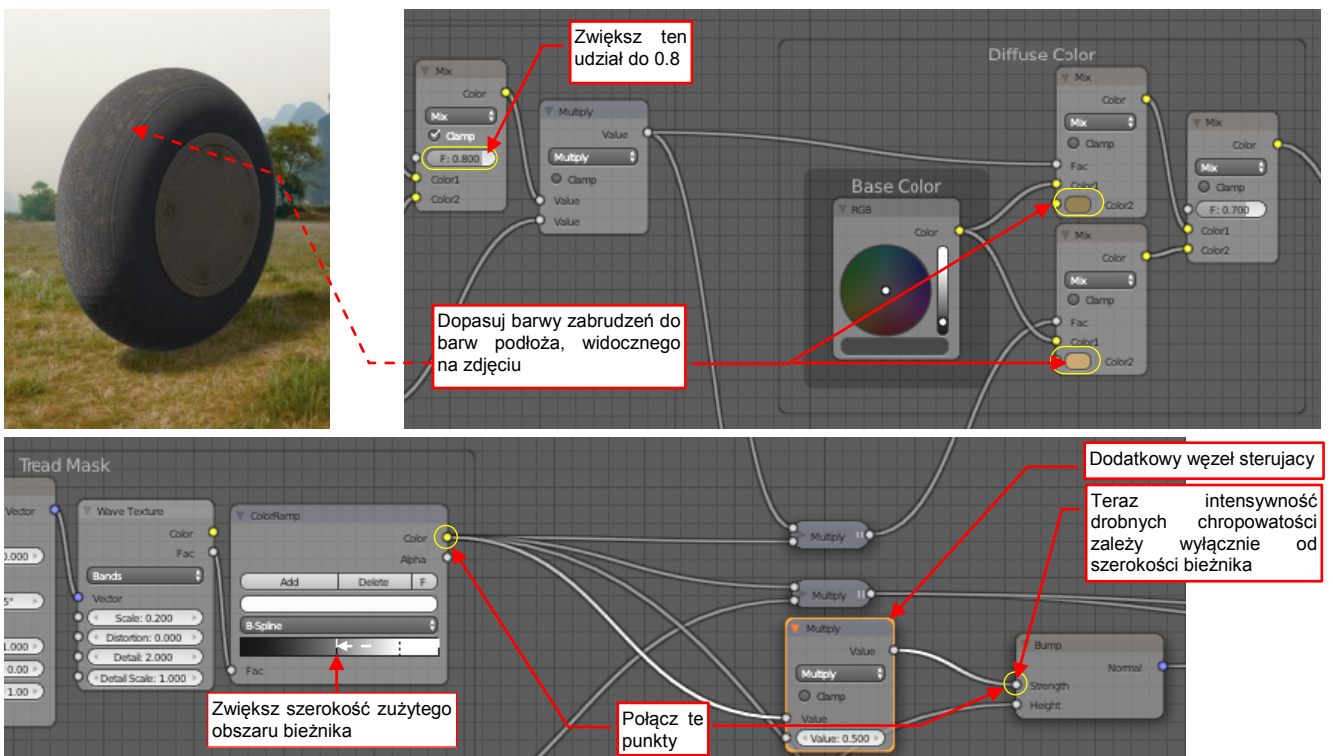


Do wyjaśnienia działania komponentów materiału **B.Rubber.Tyre** używałem grup węzłów, gdyż inaczej całość nie zmieściłaby się na ilustracjach w tej książce. (Przynajmniej nie na tyle, by etykiety węzłów pozostały czytelne). Z drugiej jednak strony — tak duża liczba grup wydłuża menu **Add → Group**, a przecież każda z nich zostanie użyta tylko raz. Dlatego na koniec zastąpiłem zawartość tych grup ramkami — uważam je za coś w rodzaju komentarzy, ułatwiających zrozumienie schematu (Rysunek 9.22.14):



Rysunek 9.22.14 Ostateczna postać materiału **B.Rubber.Tyre** (grupy zastąpione ramkami)

Rysunek 9.22.15 przedstawia co trzeba w tym schemacie zmienić, aby stworzyć wrażenie większego zużycia i zabrudzenia opony. Pokazują to na przykładzie koła na lotnisku polowym:



Rysunek 9.22.15 Modyfikacje w celu zwiększenia wrażenia zużycia opony

### 9.23 Skonfigurowanie materiału: skóra

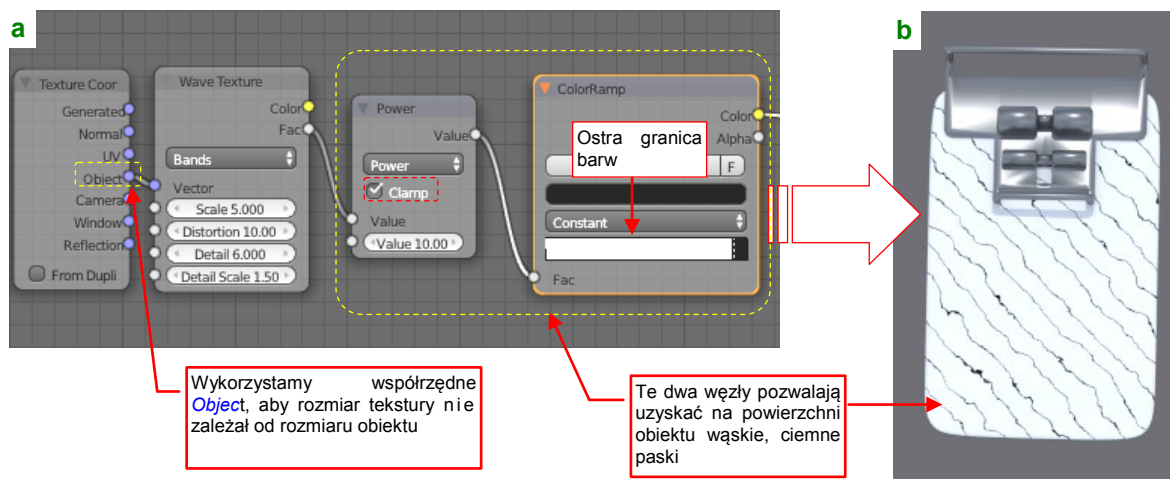
W kabinie występują dwa elementy pokryte skórą: zagłówek i podkładka pod „spinką” pasów (Rysunek 9.23.1):



Rysunek 9.23.1 Skórzane elementy modelu

Skórzane materiały mają charakterystyczny wzór powierzchni. W „starym” renderze Blendera (*Blender Render*) można go szybko uzyskać za pomocą proceduralnej tekstury *Clouds:Voronoi Crackle*. Niestety, w Cycles nie ma takiego węzła, więc musimy sami zbudować coś podobnego z innych tekstur. Rozwiązanie prezentowany w tej sekcji jest oparte na materiale opublikowany w serwisie Blend Swap<sup>1</sup>.

Tworzenie materiału pokażę to na przykładzie podkładki pod sprzączkę pasa biodrowego (Rysunek 9.23.1). (Na czarnej skórze zagłówka nie widać pewnych drobiazgów). Podstawową teksturę zbudujemy poprzez złożenie wielu tekstur *Wave Texture:Bands*. Rysunek 9.23.2 przedstawia pierwszą przymiarkę, służącą do doboru podstawowych parametrów, takich jak rozmiar:

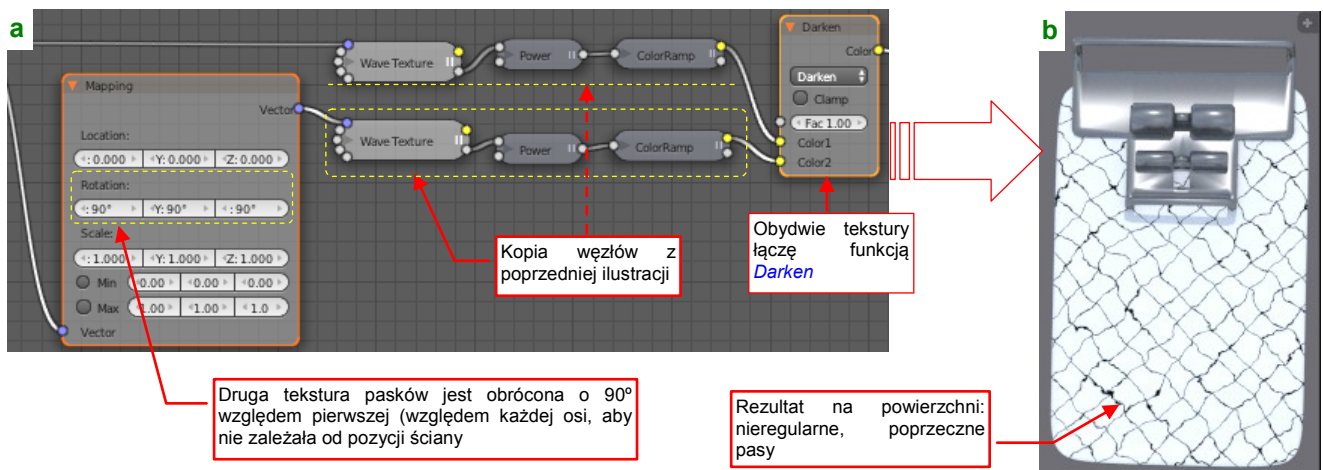


Rysunek 9.23.2 Wstępna wersja tekstury

Jako współrzędne tekstury zdecydowałem się zastosować lokalne współrzędne obiektu (*Texture Coordinate:Object*). Pozwoli to zachować taką samą rozmiar wzoru dla wszystkich obiektów o skali = 1. Za pomocą dwóch dodatkowych węzłów: *Math (Power)* i *Color Ramp* uzyskałem wzór cienkich pasków.

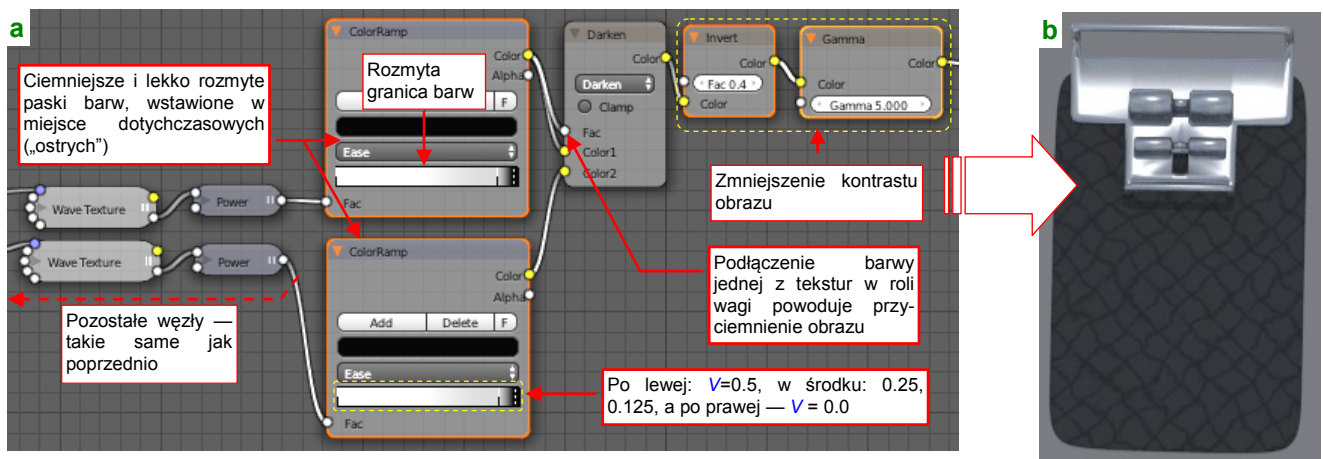
<sup>1</sup> Jego autor posługuje się tam nickiem *elbrujodelatribu*, adres strony: <http://www.blendswap.com/blends/view/66659>.

Oczywiście, takim paskom jeszcze daleko do złożonego wzoru skóry. Na początek połączmy ze sobą dwa takie same obrazy, obrócone o 90° (Rysunek 9.23.3):



Rysunek 9.23.3 Złożenie dwóch obróconych obrazów pasków

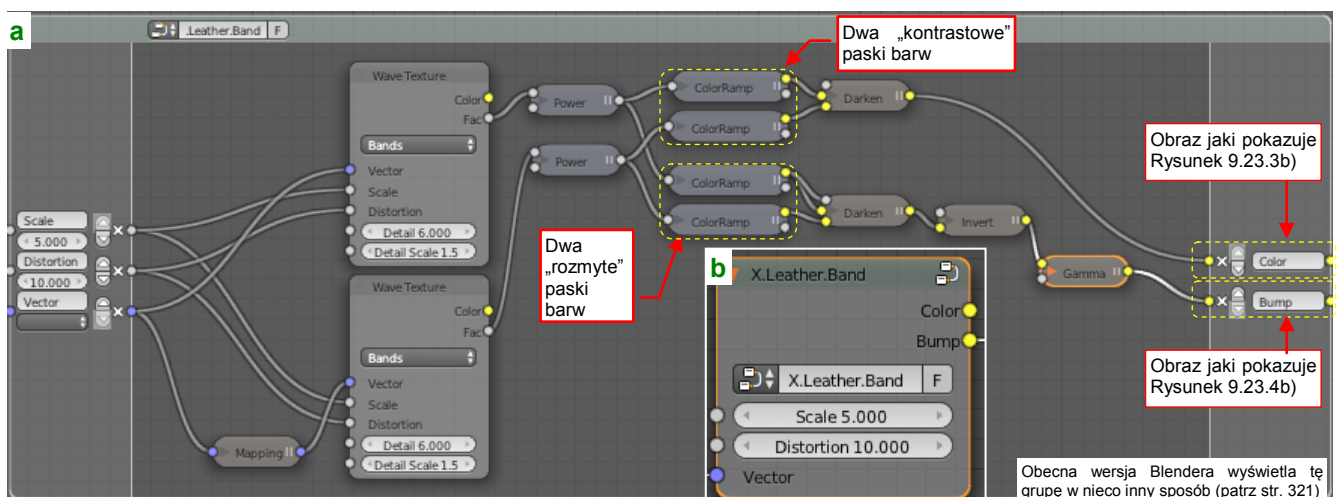
W ten sam sposób można uzyskać obraz dla mapy nierówności. Wystarczy w miejsce pasków *Color Ramp* o ostrej granicy barw (por. Rysunek 9.23.2) wstawić dwa ciemniejsze i nieco rozmyte (Rysunek 9.23.4):



Rysunek 9.23.4 Druga tekstura: nierówności

Dodatkowe węzły (*Invert* i *Gamma*) służą w tym schemacie do dalszego zmniejszenia kontrastu obrazu.

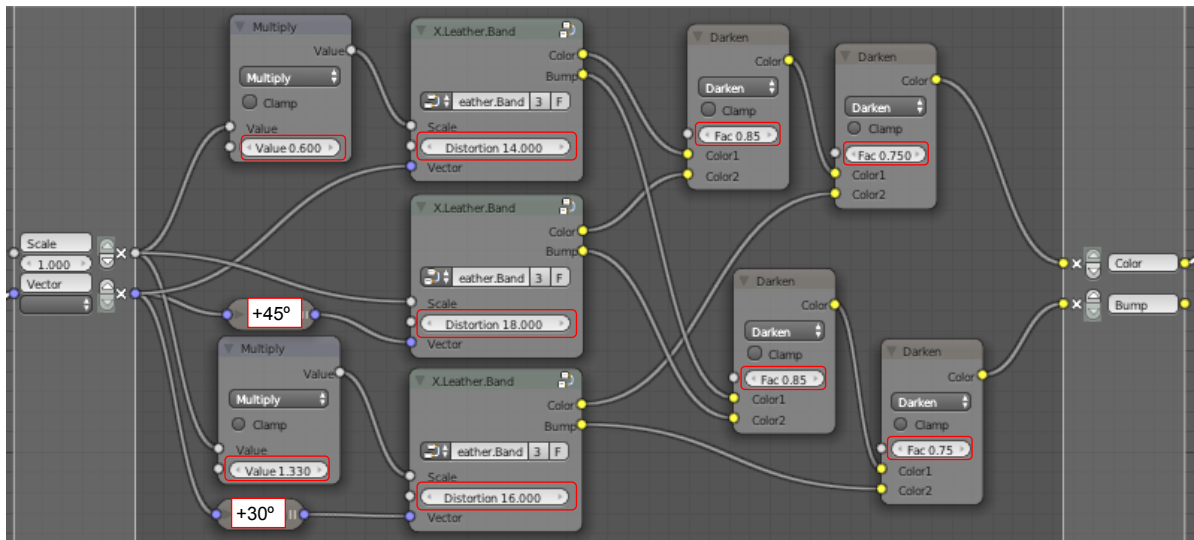
Obydwe tekstury mają wiele podobnych węzłów, stąd zdecydowałem się je połączyć w jeden pomocniczy komponent, o nazwie **X.Leather.Band** (Rysunek 9.23.5):



Rysunek 9.23.5 Pomocniczy komponent materiału **X.Leather.Band**

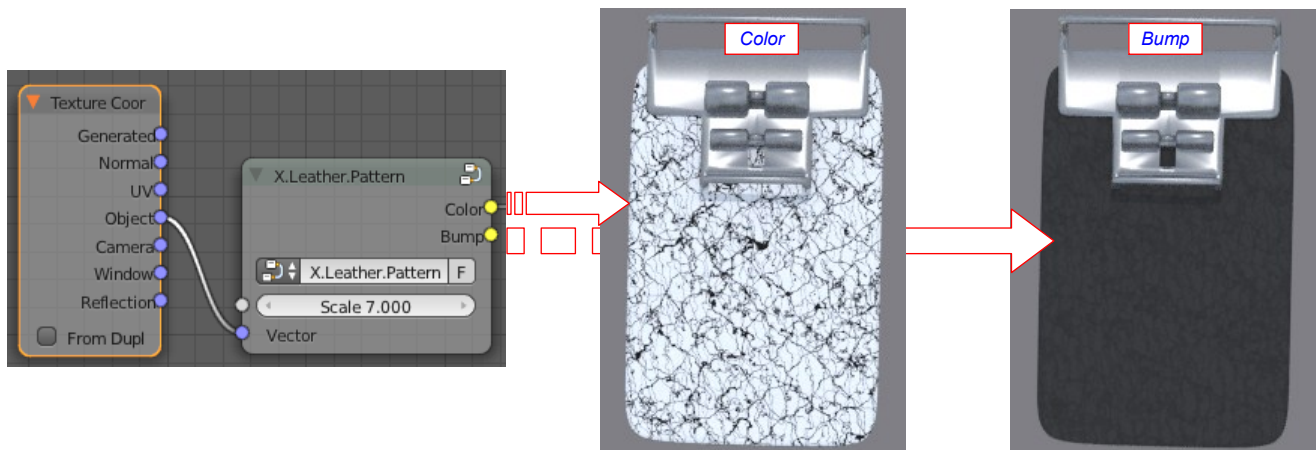


Teraz złożymy trzy takie obrazy, różnicując ich kąty obrotu (współczynniki w węzłach *Mapping*), zaburzenia kształtu (w *X.Leather.Band:Distortion*), skale (w węzłach *Multiply*) i odcienie szarości (w węzłach *Darken*). Taką pomocniczą teksturę nazwijmy *X.Leather.Pattern*. Jej schemat przedstawia Rysunek 9.23.6:



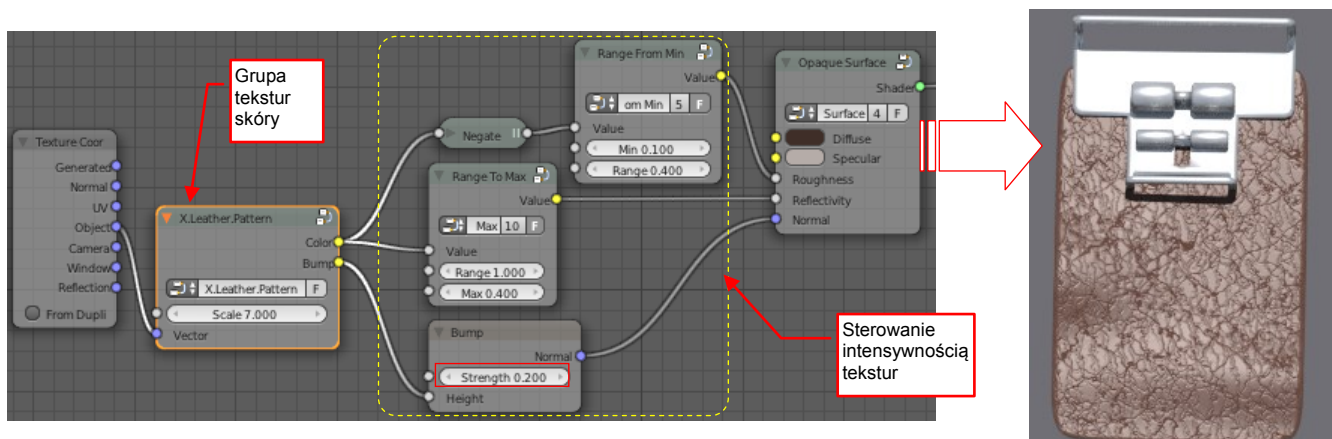
Rysunek 9.23.6 Schemat tekstury *X.Leather.Pattern*: złożenie trzech różnych obrazów *X.Leather.Band*

Rysunek 9.23.7 przedstawia obrazy generowane przez teksturę *X.Leather.Pattern*:



Rysunek 9.23.7 Przykładowy obraz, uzyskany za pomocą tekstury *X.Leather.Pattern*

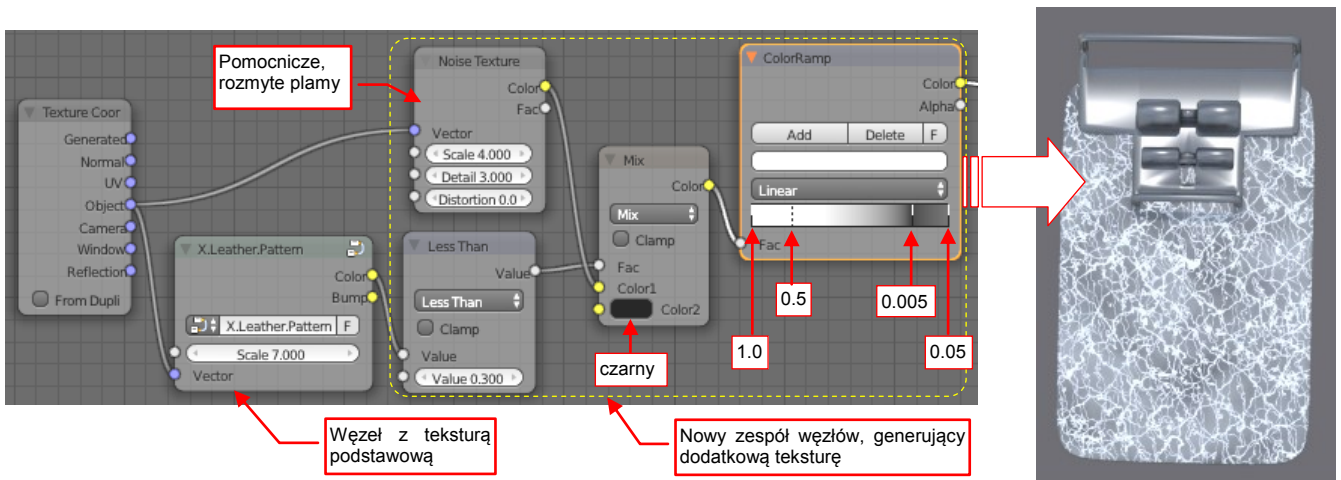
Gdy wykorzystamy wyjście *X.Leather.Pattern:Color* do modulacji szorstkości i połyskliwości standardowego materiału *Opaque Surface* (por. str. 375), a wyjście *Bump* do stworzenia mapy nierówności — uzyskamy coś przypominającego powierzchnię skóry (Rysunek 9.23.8):



Rysunek 9.23.8 Pierwsza wersja materiału skóry

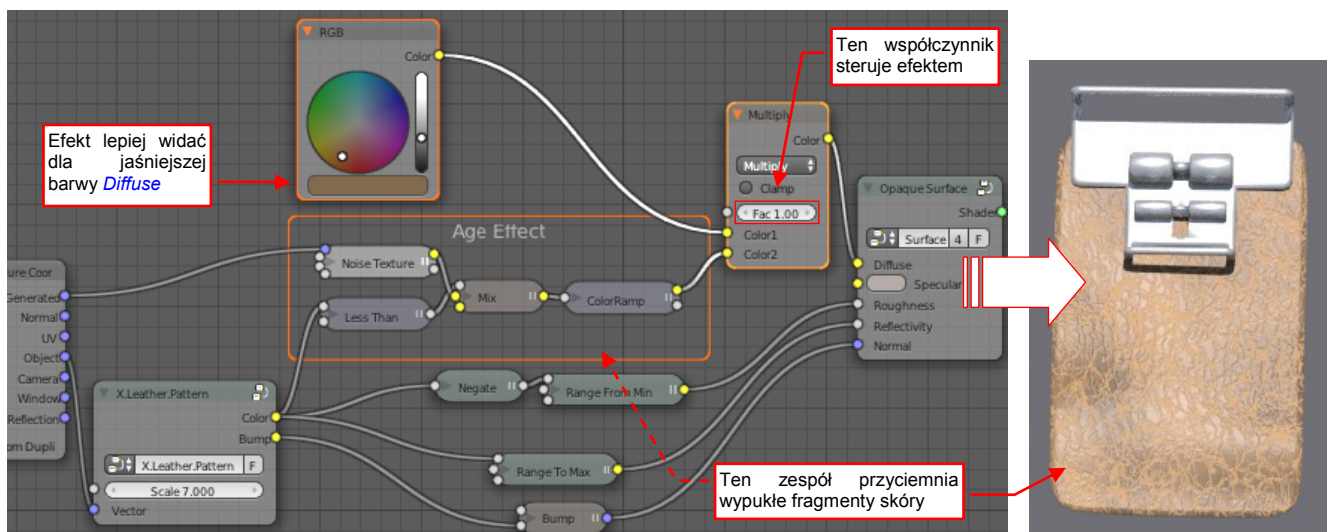


Aby osiągnąć jeszcze lepszy efekt, trzeba nanieść na tę powierzchnię jakieś ślady zużycia. W tym celu połączyłem „miękkie” plamy z tekstury **Noise** z **X.Leather.Pattern**, tworząc kolejny obraz (Rysunek 9.23.9):



Rysunek 9.23.9 Dodatkowy obraz, który wykorzystamy do efektu „zużycia”

Ten obraz zmieszamy za pomocą węzła **Color:Mix (Multiply)** z podstawową barwą **Opaque Surface:Diffuse** (Rysunek 9.23.10):



Rysunek 9.23.10 Nałożenie obrazu „zużycia” na materiał skóry (**B.Brown.Leather**)

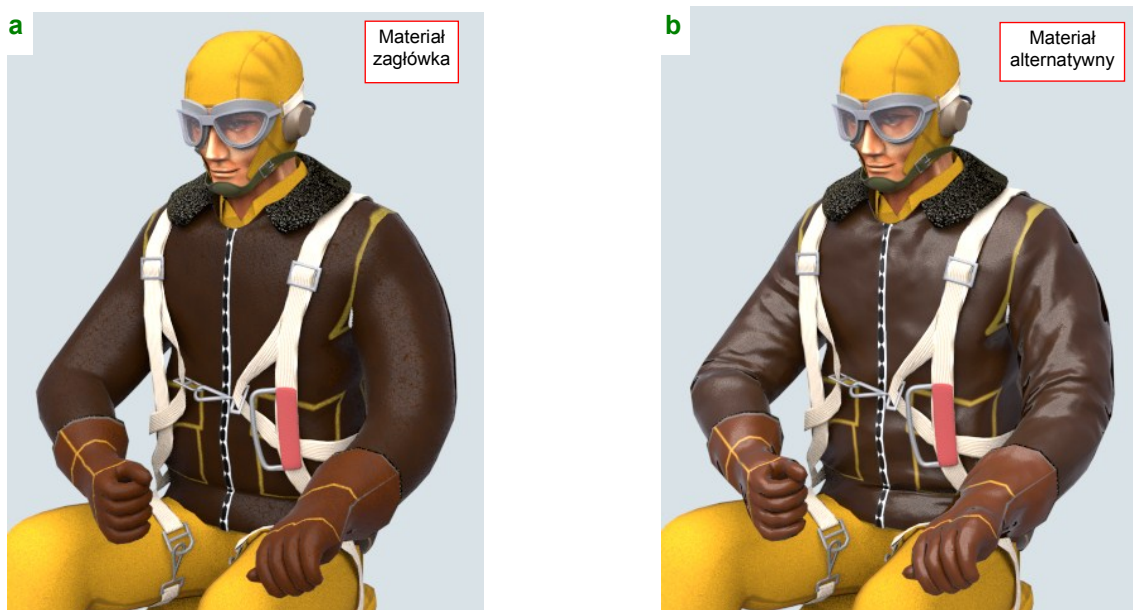
Jak widać na schemacie materiału **B.Brown.Leather**, aby wykonać takie połączenie, musiałem „wyciągnąć” barwę **Diffuse** do osobnego węzła. Efekt „wiekowania” polega na przyciemnieniu wypukłych miejsc na powierzchni skóry. Najlepiej wychodzi na jasnych powierzchniach, dlatego użyłem tu jaśniejszej barwy podstawowej (por. Rysunek 9.23.10 i Rysunek 9.23.8).

Zagłówek za siedzeniem pilota był pokryty czarną skórą (Rysunek 9.23.11). Nadałem temu materiałowi nazwę **B.Black.Leather**. Ma taki sam schemat jak ten pokazywany przez Rysunek 9.23.10, różni się tylko barwą. Zwróć uwagę że na tak ciemnej powierzchni efekt „zużycia” jest mało widoczny.



Rysunek 9.23.11 Materiał zagłówka (**B.Black.Leather**)

Jak na razie zajmowaliśmy się skórą bez fałd i „zmarszczek”. Tak wyglądała na zagłówku fotelu pilota. Jednak czasami taka gładka skóra nie jest odpowiednia. Porównaj, proszę, dwa materiały kurtki pilota, które przedstawia Rysunek 9.23.12:



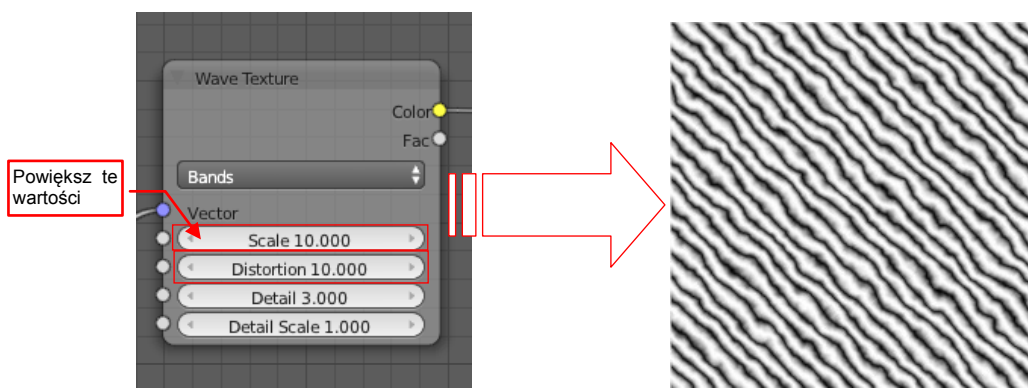
Rysunek 9.23.12 Dwa różne materiały skórzanej kurtki

Rysunek 9.23.12a) to materiał opisany na wcześniejszych stronach. Niestety, nie wygląda tu realistycznie. W żaden sposób tu nie pomaga opracowana z takim mozołem drobna tekstura powierzchni! Za to wersja, którą pokazuje Rysunek 9.23.12b) wygląda zdecydowanie lepiej. Wszystko to dzięki temu, że jest w różnych miejscach pomarszczona. Takie zmarszczki można uzyskać na kilka sposobów, m.in.:

- poprzez wymodelowanie w siatce (najbardziej pracochłonne, wymaga gęstej siatki);
- za pomocą tekstury użytej w modyfikatorze **Displace** (ale to wymaga gęstej siatki);
- za pomocą zwykłej tekstury nierówności (efekt trochę mniej od realistyczny, ale nie dodaje ścian);

Rysunek 9.23.12b) pokazuje rezultat wykorzystania tej trzeciej metody — zastosowania zwykłej tekstury nierówności. Nadałem temu materiałowi nazwę **000.Leather.Jacket** (otrzymał prefiks „000.” a nie „B”, bo należy do uniwersalnego modelu pilota, i nie jest specyficzny dla samolotu P-40B). Poniżej pokażę jak taki efekt uzyskać:

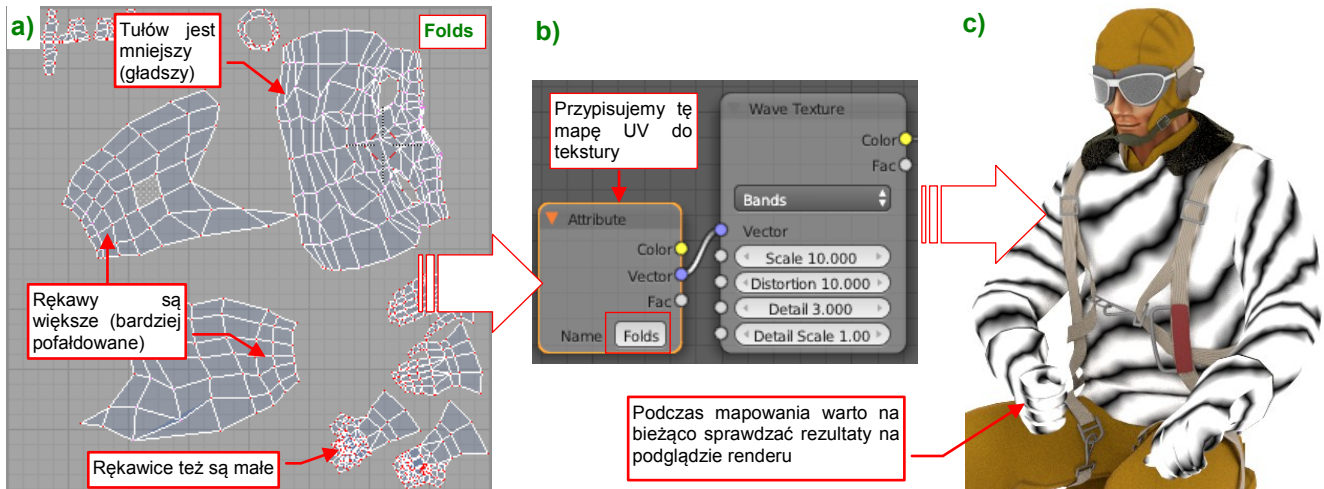
W zasadzie zarys fałd można byłoby namalować w GIMP lub Inkscape (wykorzystując filtry — por. Wyciąg III). My pójdziemy tu jednak na zupełne uproszczenie, i wykorzystamy odpowiednią teksturę proceduralną (Rysunek 9.23.13):



Rysunek 9.23.13 Tekstura ze wzorem „fałd”

To tekstura **Wave:Bands**. Po dodaniu tego węzła do schematu materiału zwiększyłem jego wartości **Scale** i **Distortion** do 10.

Aby odpowiednio nałożyć tę teksturę na kurtkę, najlepiej jest przygotować dodatkowe rozwinięcie siatki. Nadałem mu nazwę **Folds**. Powiększ na nim wszystkie obszary które mają być gęsto pofałdowane, a zmniejsz obszary które mają być bardziej płaskie. Rysunek 9.23.14a) zastosowane rozwinięcie siatki kurtki:

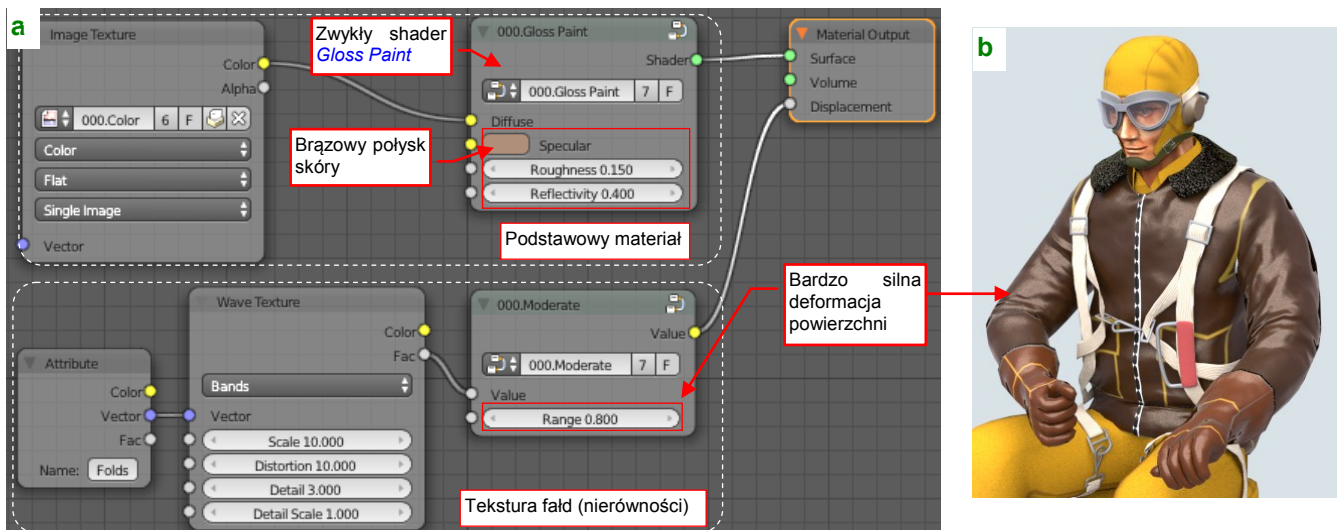


Rysunek 9.23.14 Mapowanie fałd na kurtkę pilota

Przypisz mapowanie **Folds** za pomocą węzła **Attribute** do tekstury. W tym rozwiązaniu sterujesz kształtem fałd zmieniając rozmiar i położenie poszczególnych ścian siatki w przestrzeni UV. Pamiętaj, aby wzajemnie dopasować wzór fałd na poszczególnych elementach — rękawach, torsie. Najlepiej na śledzić na bieżąco wynik na podglądzie renderu.

- Aby Cycles uwzględnił ostatnie zmiany siatki (w tym także zmiany w rozwinięciu **UV**) musisz się na chwilę przełączyć z **Edit Mode** na **Object Mode**. Potem z powrotem możesz wrócić do **Edit Mode**.

Tak przygotowany obraz wykorzystaj jako teksturę nierówności. Rysunek 9.23.15a) przedstawia schemat materiału **000.Leather.Jacket**, a Rysunek 9.23.15b) — podgląd efektu na renderze:



Rysunek 9.23.15 Materiał 000.Leather.Jacket

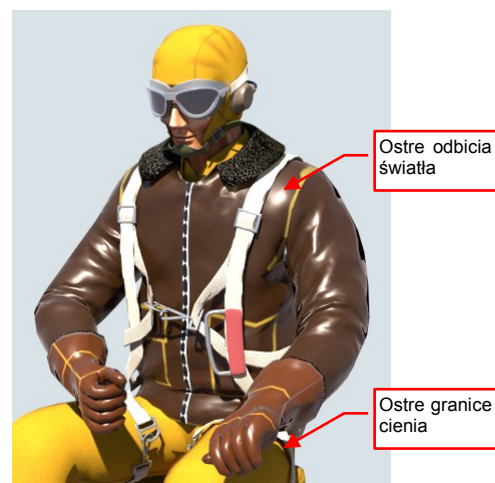
Zwróć uwagę na węzeł **Moderate**, sterujący głębokością nierówności. Jest ustawiony na stosunkowo dużą wartość (0.8), co tworzy na renderze efekt fałd (Rysunek 9.23.15b).

Właściwie to poza teksturą fałd jest to bardzo prosty materiał, wykorzystujący nasz standardowy shader **Gloss Paint** i teksturę barwy (Rysunek 9.23.15a). Barwa połysków (**Specular**) jest jasnobrązowa, a powierzchnia dość połyskliwa (**Reflectivity** = 0.4). Ten model pilota będzie używany do ujęć z większych odległości, i dla takich zastosowań nie ma potrzeby stosowania bardziej skomplikowanych rozwiązań.

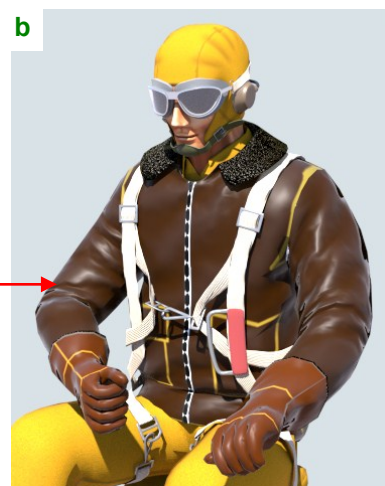
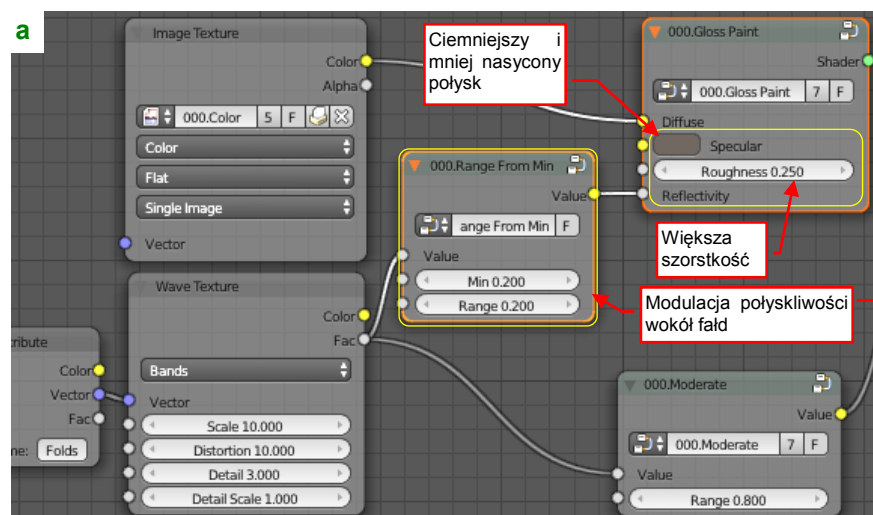


Okazuje się, że takie odbłaski na fałdach zależą nie tylko od połyskliwości materiału, ale i od oświetlenia! Render, który pokazuje Rysunek 9.23.15b) był wykonany z lampą typu *Sun*. To było źródło o dużej mocy (*Strength* = 25), ale także o dużym rozmiarze (*Size* = 1). Można to poznać po bardzo miękkich i delikatnych cieniach, jakie powstały na modelu. Gdy zmniejszymy rozmiar słońca do takiego, jaki używamy w scenie z modelem samolotu (*Size* = 0.01), efekt będzie o wiele mniej ciekawy (Rysunek 9.23.16)

Na renderze pojawiły się ciemne cienie o ostrych granicach. Odbłaski na kurtce stały się zbyt ostre, wręcz metaliczne. Ogólny efekt jest zdecydowanie gorszy. Aby to skorygować, trzeba nieco zmodulować połyskliwość (*Reflectivity*) shadera materiału (Rysunek 9.23.17a):



Rysunek 9.23.16 Ten sam materiał w ostrym świetle słonecznym (*Sun:Size* = 0.01)



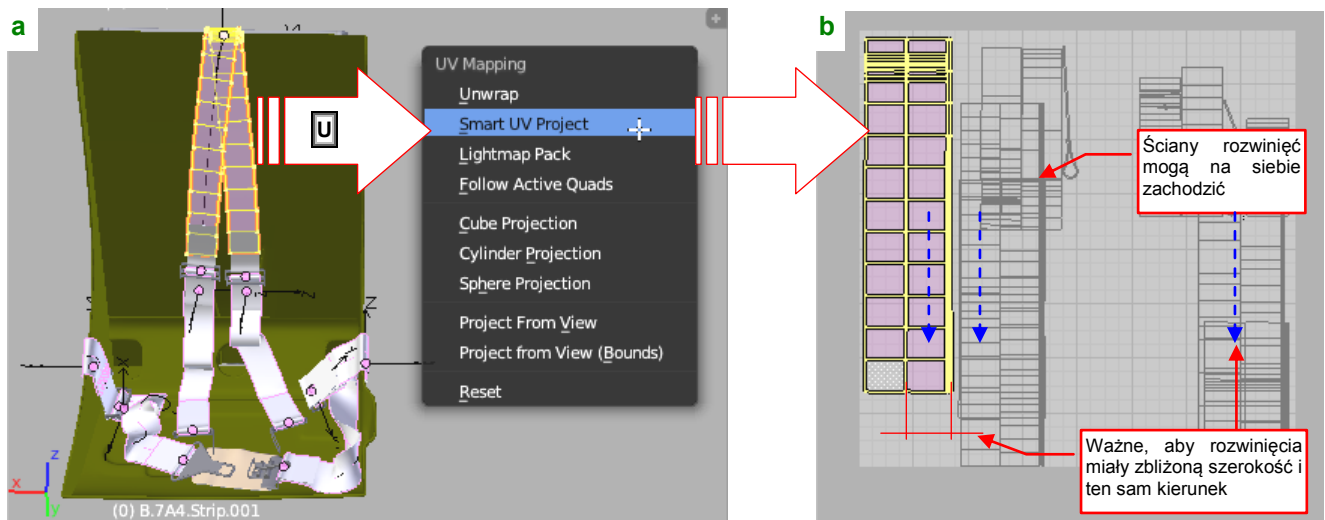
Rysunek 9.23.17 Dostosowanie materiału kurtki do ostrego oświetlenia

Po tych modyfikacjach kurtka wygląda całkiem znośnie (Rysunek 9.23.17b). Przypuszczam, że można tu wprowadzić jakieś inne modyfikacje, które nadały by jej powierzchni bardziej „skórzany” wygląd. Na razie jednak nie znalazłem niczego lepszego i pozostawiam ten efekt w postaci jak na ilustracji.



### 9.24 Skonfigurowanie materiału: tkanina (pasów)

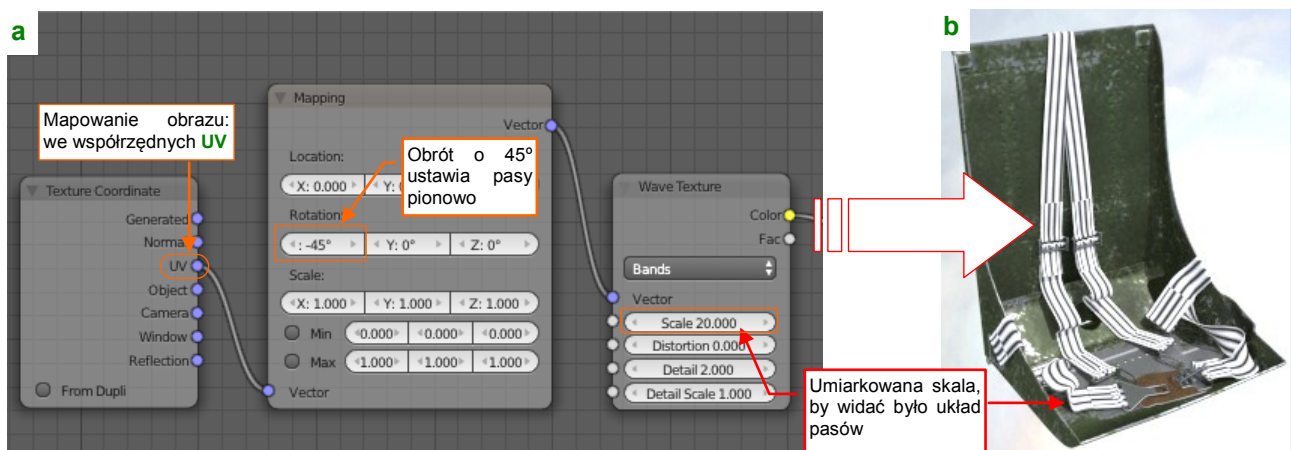
Pasy fotela pilota były wykonane z grubej tkaniny. Aby uzyskać jednolity kierunek ich wątków, rozwiń je w przestrzeni **UV** (Rysunek 9.24.1):



Rysunek 9.24.1 Rozwinięcie siatek pasów na powierzchni UV

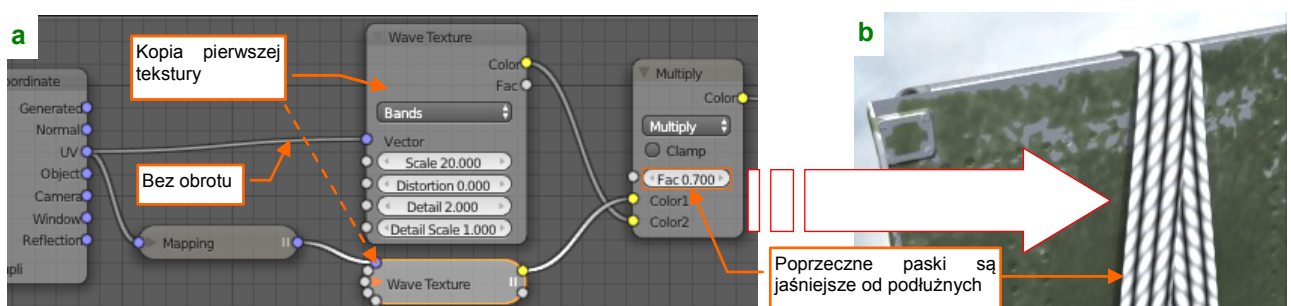
Najszybciej jest to zrobić używając polecenia **Smart UV Project** z menu **Mesh → UV Unwrap** (U) — Rysunek 9.24.1a). To rozwinięcie nie musi być staranne ani dokładne: poszczególne siatki mogą się na siebie nakładać. Ważne jest jedynie, aby miały takie same szerokości pasa, oraz kierunek (pionowy — Rysunek 9.24.1b).

Teraz nałożmy na te siatki teksturę pasków. Należy je obrócić o 45°, aby biegly wzdłuż siatek, tak jak włókna tkaniny (Rysunek 9.24.2):



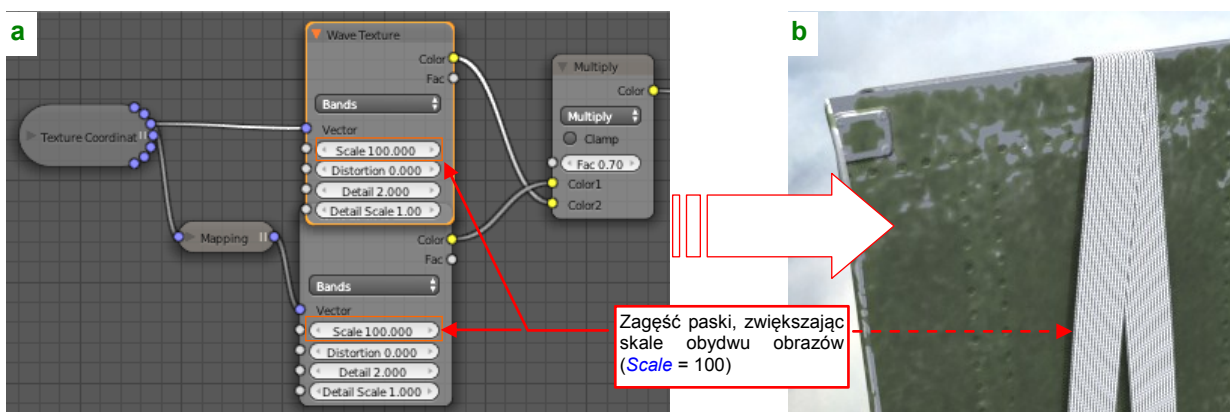
Rysunek 9.24.2 Pierwsza „przymiarka” tekstury pasów

Teraz połącz je z drugą teksturą pasków — tym razem pochyłych (o 45° — Rysunek 9.24.3)



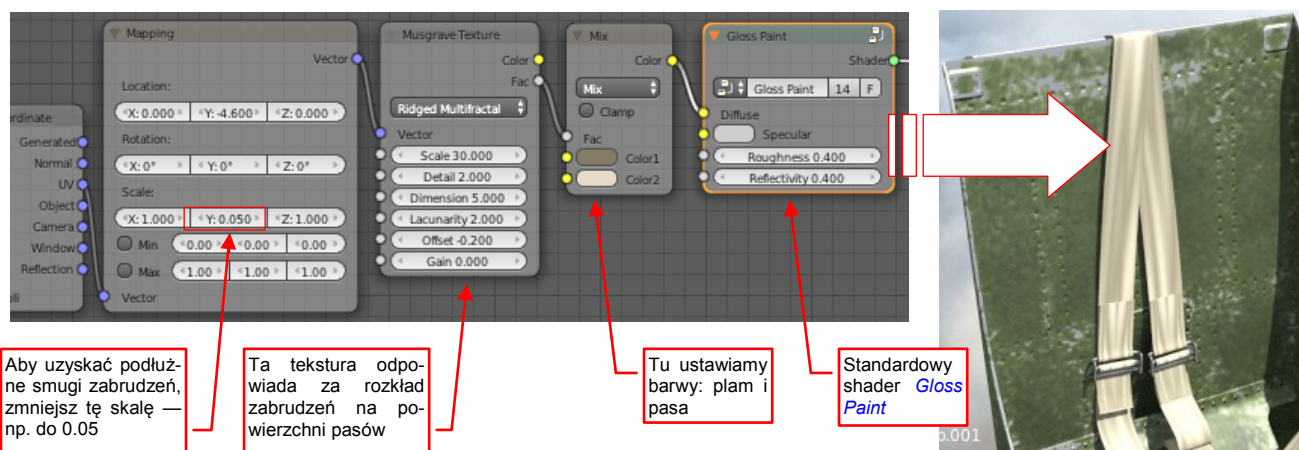
Rysunek 9.24.3 Dodanie pasów poprzecznych

Gdy pasy mają właściwy odcień i orientację, zwiększ skalę obydwu tekstur, by rozmiar pasków stał się podobny do rozmiaru włókien (Rysunek 9.24.4):



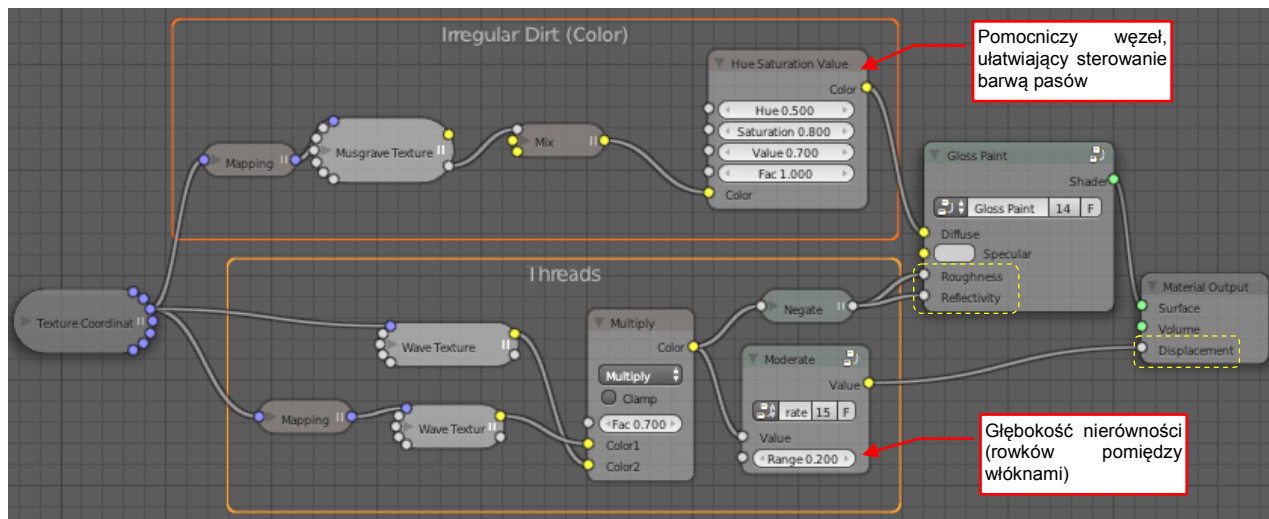
Rysunek 9.24.4 Zmniejszenie skali pasów

Przygotujmy jeszcze obraz barwy pasów. Oryginalnie te pasy były białe, ale w warunkach polowych taki kolor nie miał szans się długo utrzymać. Dlatego nadałem im kolor „kawy z mlekiem” (Rysunek 9.24.5b):



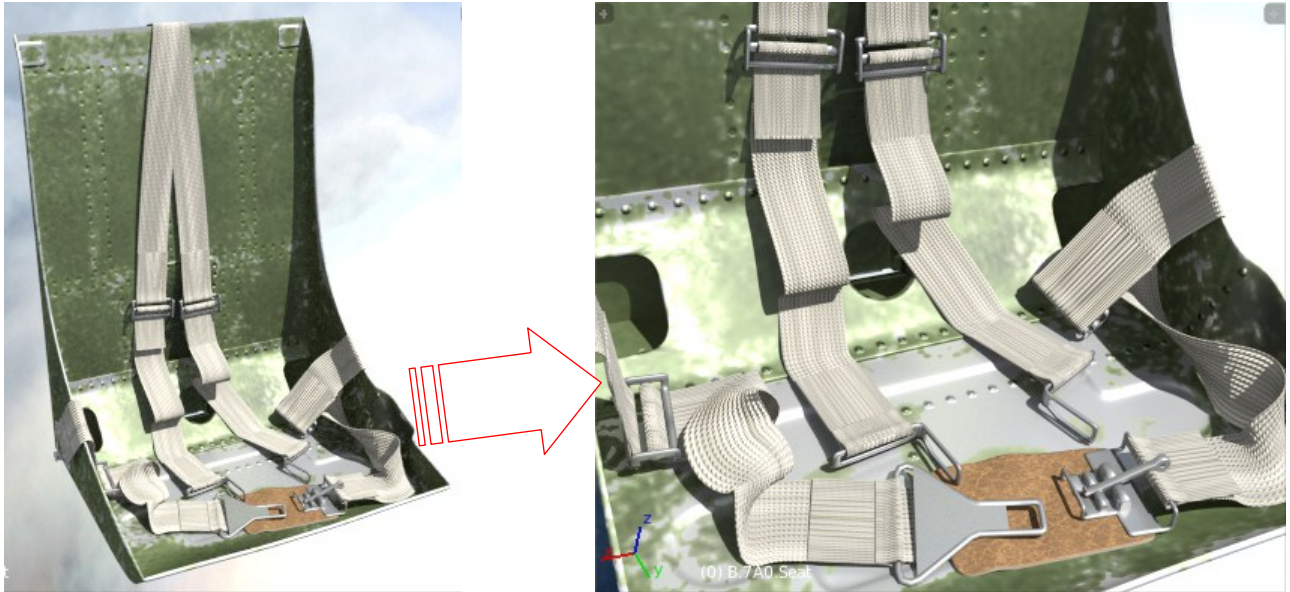
Rysunek 9.24.5 Dobór ogólnej barwy materiału

Jako podstawowego shadera użyłem grupy **Gloss Paint**. Na podstawowy kolor nałożyłem ciemniejsze plamy, którym kształt nadała znacznie rozciągnięta w pionie (**Mapping:Scale:Y** = 0.05) tekstura **Musgrave** (Rysunek 9.24.5a). Teraz podłącz do shadera odwrotność tekstury włókien — aby modulowała wartości jego szorstkości (**Roughness**) i odbić (**Reflectivity**). Tego samego obrazu użyj jako tekstury nierówności (Rysunek 9.24.6):



Rysunek 9.24.6 Materiał pasów

Rysunek 9.24.7 przedstawia uzyskany rezultat:



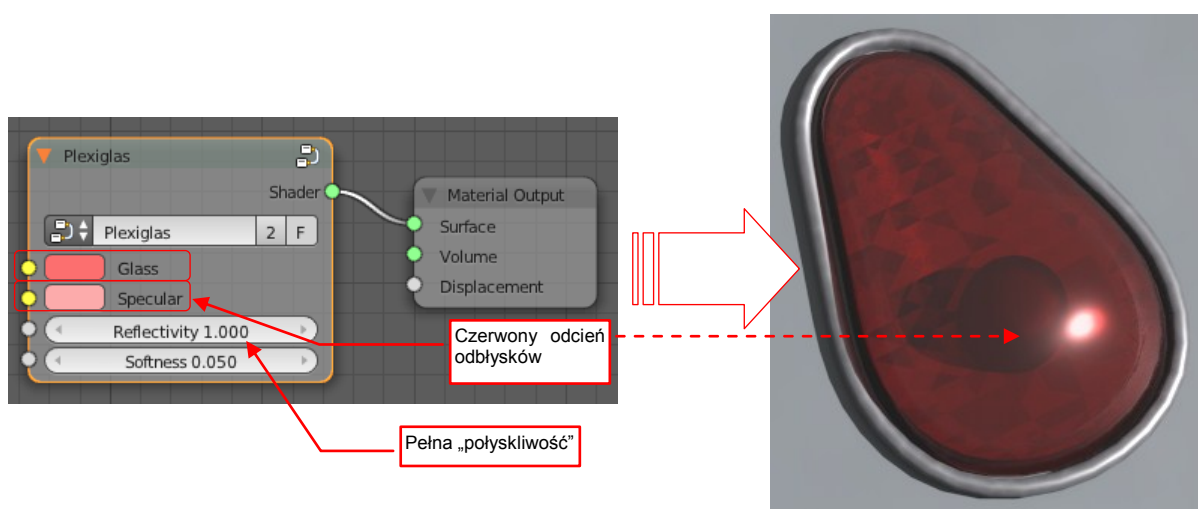
Rysunek 9.24.7 Materiał pasów — rezultat

Te pasy wyglądają na bardzo brudne. W razie czego możesz je nieco „wyczyścić”, zmniejszając kontrast ich barw. (Wystarczy dodać na schemacie przedstawionym przez Rysunek 9.24.6 węzeł *Bright/Contrast* za węzłem *Hue Saturation Value*, a potem zmniejszyć wartość *Contrast* do -0.1).

## 9.25 Skonfigurowanie materiału: plastik

W konstrukcjach z lat 40-tych występuje kilka rodzajów tworzyw sztucznych (bakelit, kauczuk syntetyczny, szkło organiczne). Połyskliwy materiał, który dziś nazywamy potocznie „plastikiem”, był wówczas zabarwionym szkłem organicznym. W P-40 wykorzystano go m.in. w osłonach świateł pozycyjnych. (Czerwonych na lewym skrzydle, zielonych na prawym, i białych na ogonie).

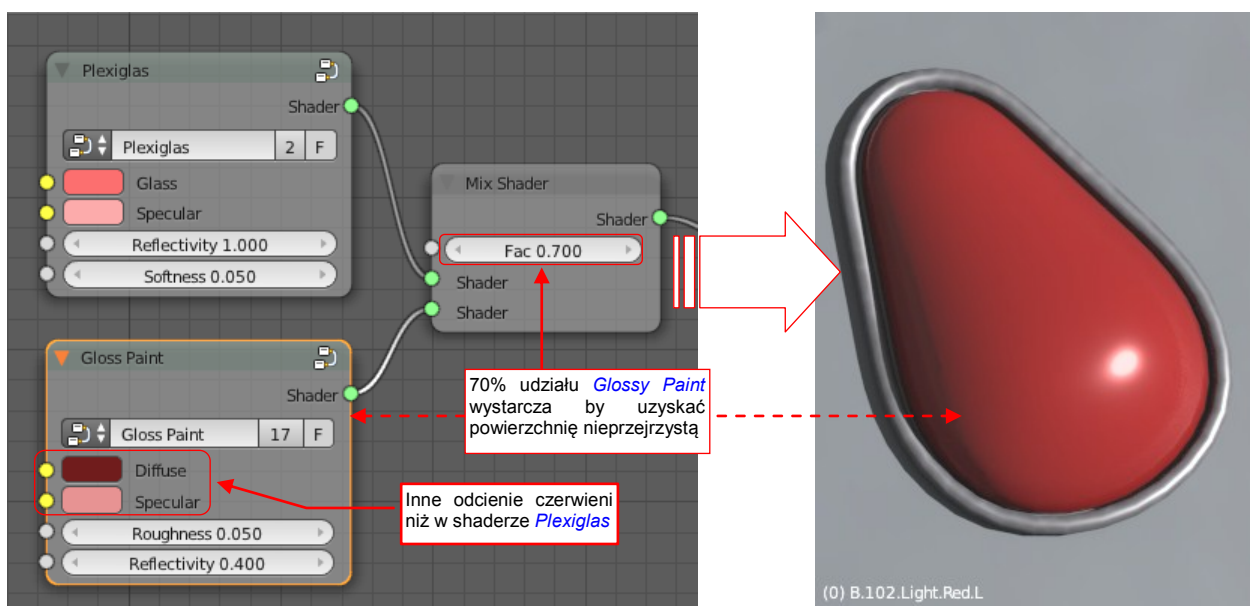
Shader szkła organicznego — grupę **Plexiglas** — stworzyliśmy w jednej z poprzednich sekcji (por. str. 360). Wykorzystajmy więc go w tym materiale — wystarczy tylko zmienić kolor na czerwony (Rysunek 9.25.1):



Rysunek 9.25.1 Przezroczysta osłona lewego światła pozycyjnego

Zwróć uwagę, że barwie odbłyśków (**Specular**) także nadałem odcień czerwony. (Choć jest to kolor jaśniejszy i o mniejszym nasyceniu niż podstawowa barwa **Glass**).

A jak uzyskać nieprzezroczystą wersję takiego materiału? Wystarczy mieszać shader **Plexiglas** z shaderem **Glossy Paint** (Rysunek 9.25.2):

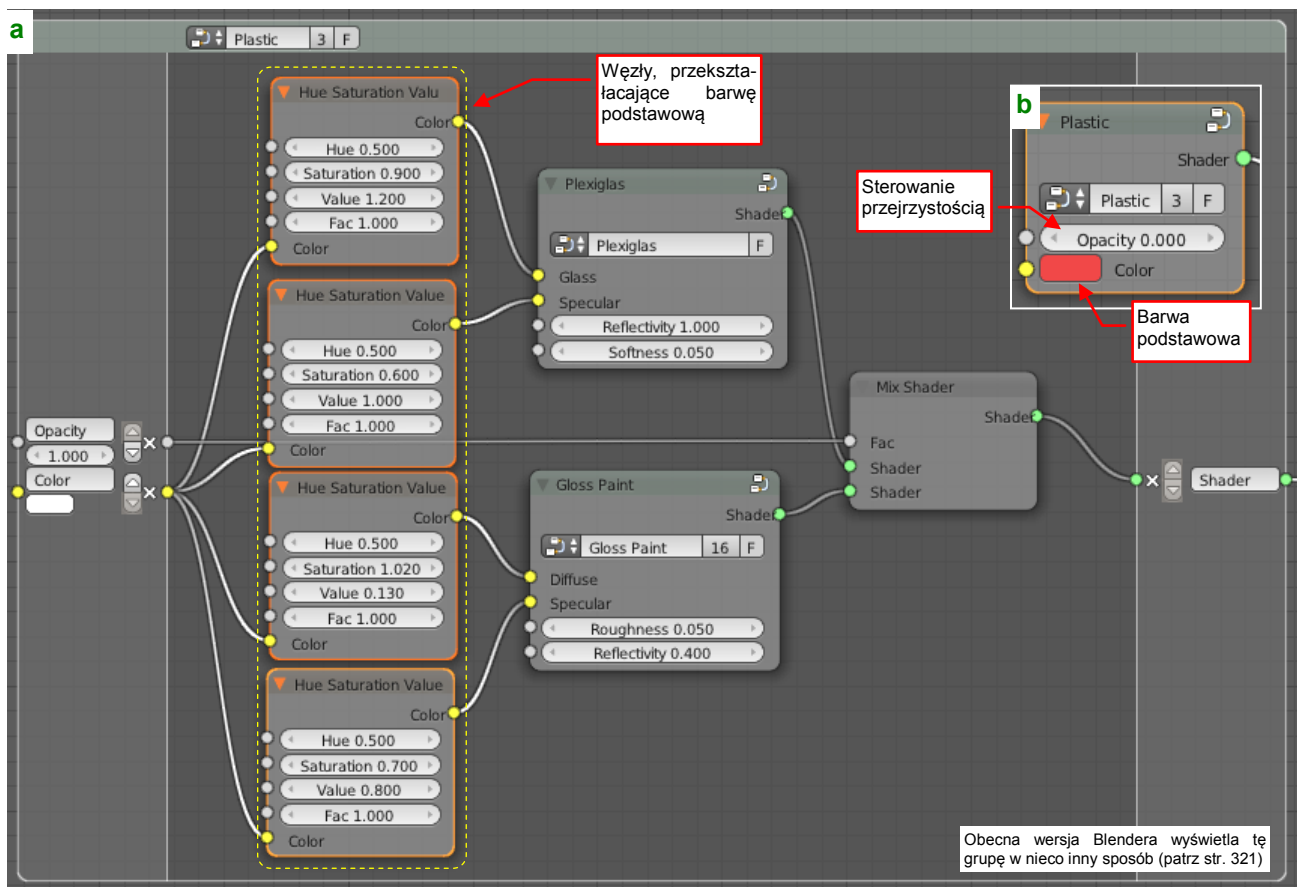


Rysunek 9.25.2 Nieprzezroczysta wersja „plastiku”

Już dla 70% udziału shadera **Glossy Paint** powoduje, że materiał wygląda zupełnie nieprzezroczystie. Jednocześnie zachowuje połysk szkła organicznego. Zwróć uwagę na dobór kolorów **Diffuse** i **Specular** w shaderze **Glossy Paint**: mają jeszcze inny odcień niż shaderze **Plexiglas**. Dobór tych odcieni wymagał trochę pracy!



Aby za każdym razem nie ustawiać tych czterech barw, stworzyłem pomocniczy shader **Plastic** (Rysunek 9.25.3a):



Rysunek 9.25.3 Shader **Plastic**

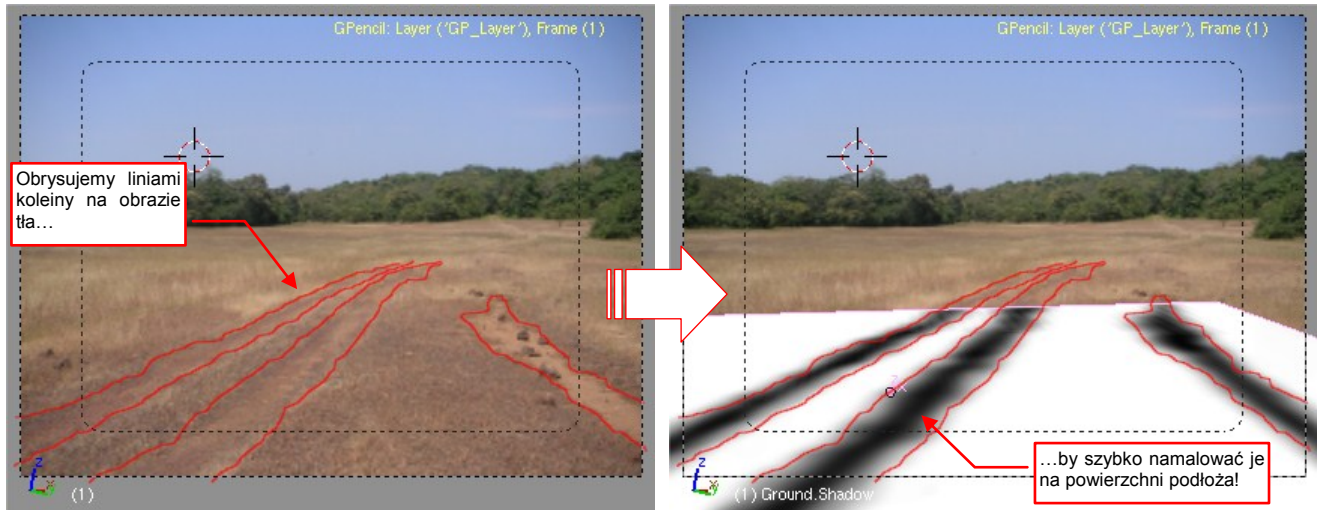
Używając tej grupy ustalasz tylko jedną barwę. W środku zmieniam jej odcienie za pomocą odpowiednich węzłów **Hue Saturation Value**, dostosowując dla poszczególnych wejść shaderów **Plexiglas** i **Glossy Paint**. Parametr **Opacity** (Rysunek 9.25.3b) pozwala uzyskać także plastik półprzezroczysty.

## **Rozdział 10. Blender — pozostałe**

Tu umieściłem wszystkie polecenia Blendera, które nie pasują do pozostałych rozdziałów. Co prawda połowa jego sekcji jest poświęcona schematom kompozycji i ich węzłom, ale było ich zbyt mało by poświęcić im odrębny rozdział.

### 10.1 Linie pomocnicze (*Grease Pencil*)

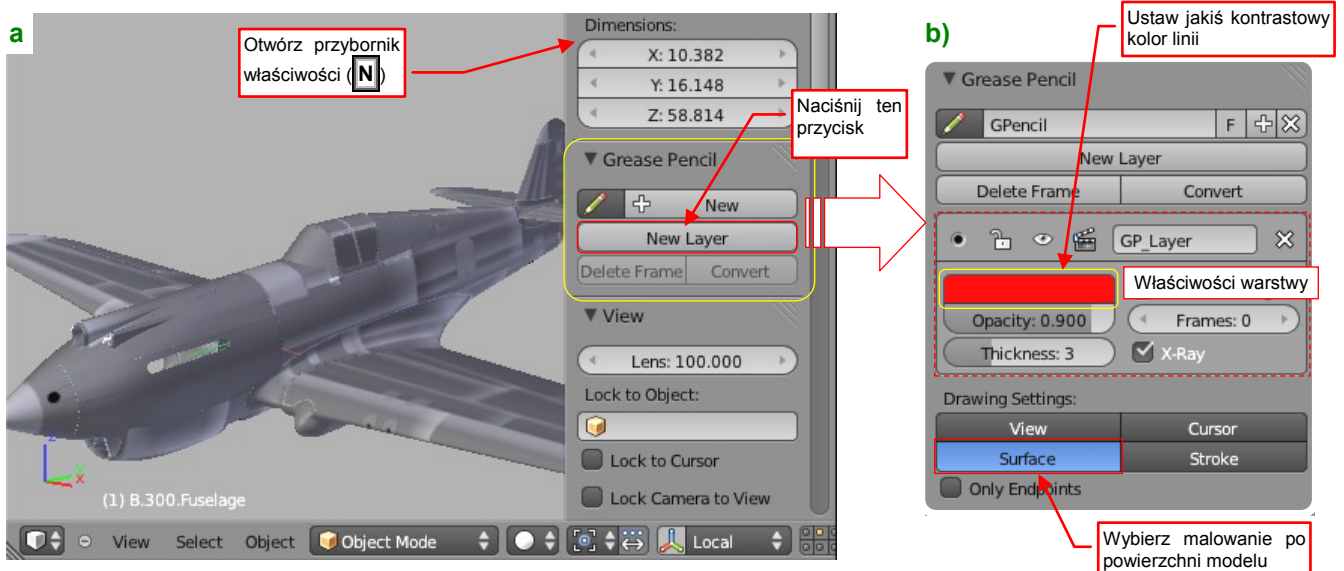
Czasami pojawia się potrzeba szybkiego naszkicowania na ekranie jakiejś linii pomocniczej. W Blenderze służy do tego funkcja *Grease Pencil*. Jej nazwa pochodzi od miękkich ołówków lub kredek świecowych, którymi w latach 90-tych animatorzy 3D rysowali na szklanych ekranach monitorów różne pomocnicze linie. *Grease Pencil* przydaje się do zaznaczania wielu różnych rzeczy — na przykład do przeniesienia zarysu kolein z fotografii tła na podstawioną pod nim płaszczyznę „gruntu” (Rysunek 10.1.1):



Rysunek 10.1.1 Wykorzystanie linii pomocniczych do stworzenia tekstury kolein

W tym przypadku nie zależy nam specjalnie na dużej dokładności linii — uzyskany obraz kolein wykorzystamy jako teksturę w modyfikatorze *Displace* (por. str. 299). Kształt kolein i tak ma być rozmyty. Do takich i podobnych zadań *Grease Pencil* nadaje się najlepiej.

Pokażę to na przykładzie malowania obrazu zabrudzeń. Użyję tu pomocniczych linii do zaznaczenia położenia jednego z jego źródeł: karabinów skrzydłowych. Zaczniemy od ustawiania parametrów narzędzia. Otwórz przybornik (*Property Shelf* — **N**) i w panelu *Grease Pencil* naciśnij przycisk *New Layer* (Rysunek 10.1.2a):

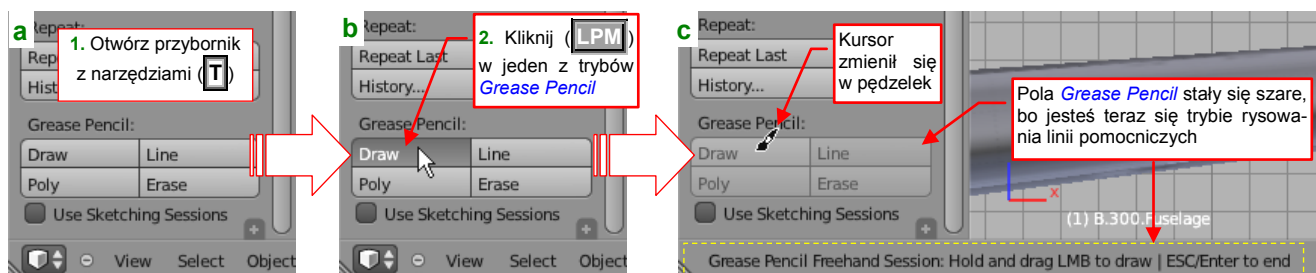


Rysunek 10.1.2 *Grease Pencil* — ustawienie parametrów aktualnej warstwy

W efekcie w panelu *Grease Pencil* pojawi się wiele nowych kontrolerek (Rysunek 10.1.2b). Co właściwie się dzieje po naciśnięciu przycisku *New Layer*? Tak jak w Inkscape czy GIMP, *Grease Pencil* rysuje swoje linie na warstwach. Ten przycisk stworzył pierwszą z nich, o domyślnej nazwie *GP\_Layer*. W nagłówku panelu możesz także zmienić nazwę warstwy, wyłączyć jej widoczność (ikona „oka”, po lewej), czy, w ostateczności, usunąć (ikona „X” po prawej). *Grease Pencil* przechowuje w takich warstwach to, co naszkicujesz.

Z każdą warstwą *Grease Pencil* (GP) jest związany jeden rodzaj linii. W panelu warstwy określasz barwę kreski, jej nieprzeźrystość (*Opacity*), oraz szerokość (*Thickness*). Dla *GP\_Layer* wybierz jakiś kontrastowy kolor linii — na przykład czerwony. Oprócz tego ustaw tryb *Drawing Settings:Surface* (Rysunek 10.1.2b). W tym trybie linie będą rysowane na powierzchni modelu.

Gdy ustawiłeś już parametry pracy, zamknij okno właściwości *3D View* (N), a otwórz jego przybornik (T). U dołu panelu *Object Tools* znajdziesz sekcję *Grease Pencil*, w której są dostępne cztery tryby rysowania. Wybierz z nich tryb *Draw* (Rysunek 10.1.3a, b):



Rysunek 10.1.3 *Grease Pencil* — przejście w tryb rysowania

Naciśnięcie przycisku *Draw* przełącza Blendera w tryb rysowania linii pomocniczych. W pasku stanu pojawia się wówczas wskazówka, jak malować i jak wyjść z tego trybu (Rysunek 10.1.3c).

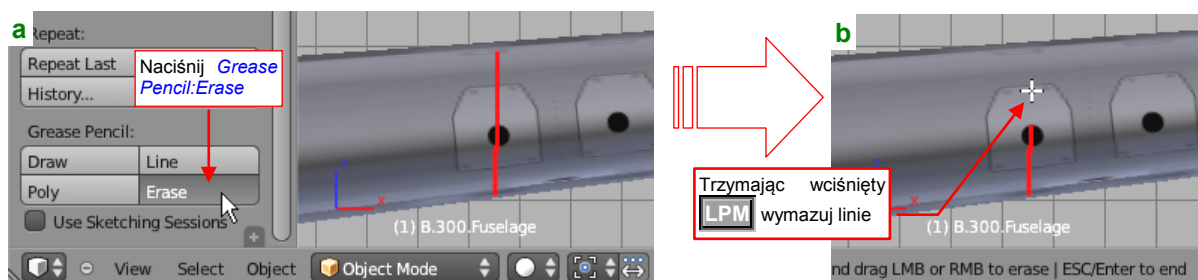
Zgodnie z tą podpowiedzią, przeciągnij po rysunku myszką z wciśniętym *LPM*, rysując linię pomocniczą (Rysunek 10.1.4a):



Rysunek 10.1.4 *Grease Pencil* — rysowanie

*Grease Pencil* traktuje to jako szkicowanie, i odpowiednio „często” zapamiętuje kolejne pozycje kursora jako nowe wierzchołki linii. Gdy tylko zwolnisz przycisk myszki, Blender uzna że rysowanie w trybie *Grease Pencil* jest zakończone, i wyjdzie z tego trybu pracy. Obejrzyj rezultaty tego rysowania, obracając model. Przy okazji przekonasz się, jak działa opcja *Surface*: linia „leży” na powierzchni płata (Rysunek 10.1.4b).

Do usuwania narysowanych kresek służy drugi przycisk z przybornika: *Grease Pencil:Erase* (Rysunek 10.1.5a). Gdy go wciśniesz, kursor zmienia się w „krzyżyk” (Rysunek 10.1.5b). Przesuwając myszkę z wciśniętym *LPM* wymazujesz narysowane wcześniej linie.

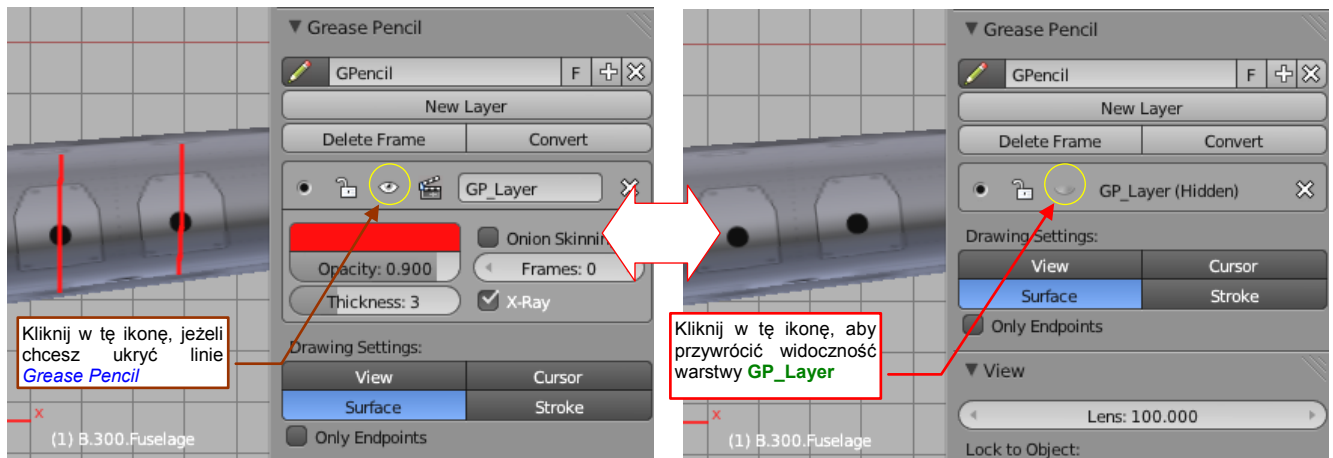


Rysunek 10.1.5 *Grease Pencil* — usuwanie linii



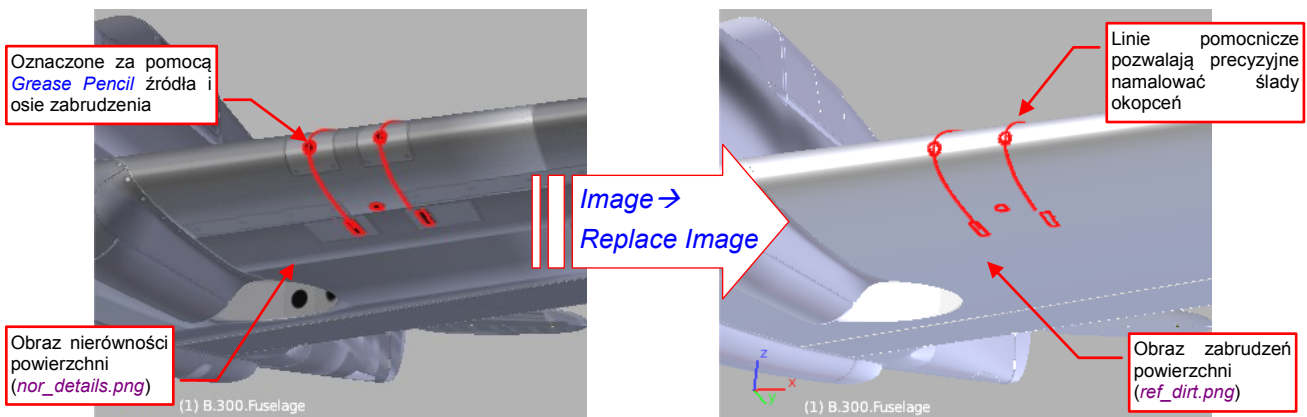
Jeżeli chciałbyś narysować pojedynczy odcinek linii prostej, użyj polecenia **Grease Pencil:Line**. Z kolei polecenie **Poly** pozwala rysować wielolinię, złożoną z linii prostych.

Podobnie jak w przypadku warstw GIMP czy Inkscape, warstwy **Grease Pencil** można chować i wyświetlać (Rysunek 10.1.6):



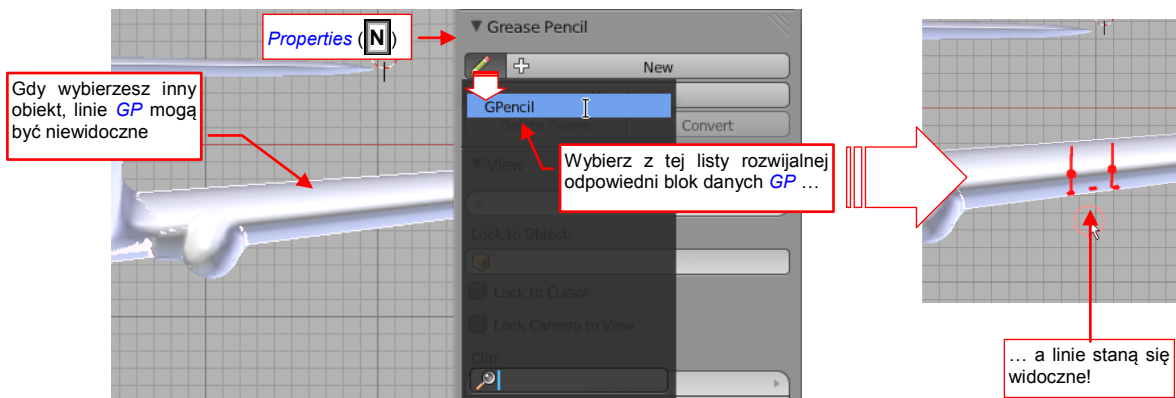
Rysunek 10.1.6 **Grease Pencil** — wyłączanie widoczności warstwy

Do tej pory wykorzystywaliśmy obraz tekstury nierówności, by zaznaczyć na modelu za pomocą **Grease Pencil** położenie luf karabinów maszynowych. Teraz, gdy już te linie mamy, podstawmy w to miejsce obraz zabrudzeń, na który należy nanieść odpowiednie okopcenia za lufami (Rysunek 10.1.7):



Rysunek 10.1.7 Podmiana obrazu tekstury

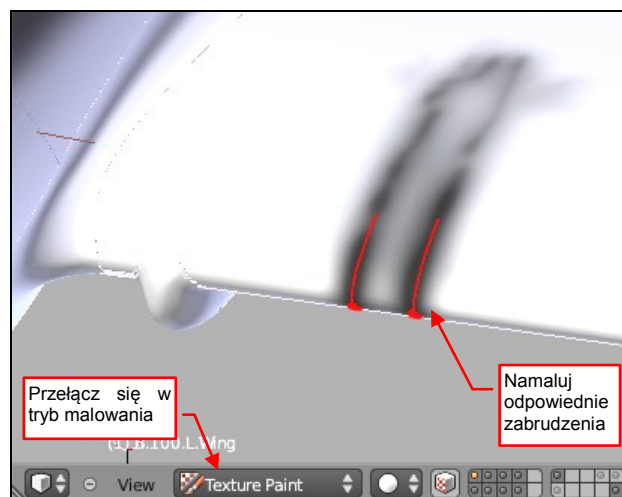
Czasami, gdy wybierzesz inny obiekt, linie pomocnicze mogą być początkowo niewidoczne (Rysunek 10.1.8):



Rysunek 10.1.8 Włączenie warstw **Grease Pencil** dla drugiego skrzydła

W takim przypadku włącz je, wybierając z listy w panelu **Grease Pencil** odpowiedni blok danych.

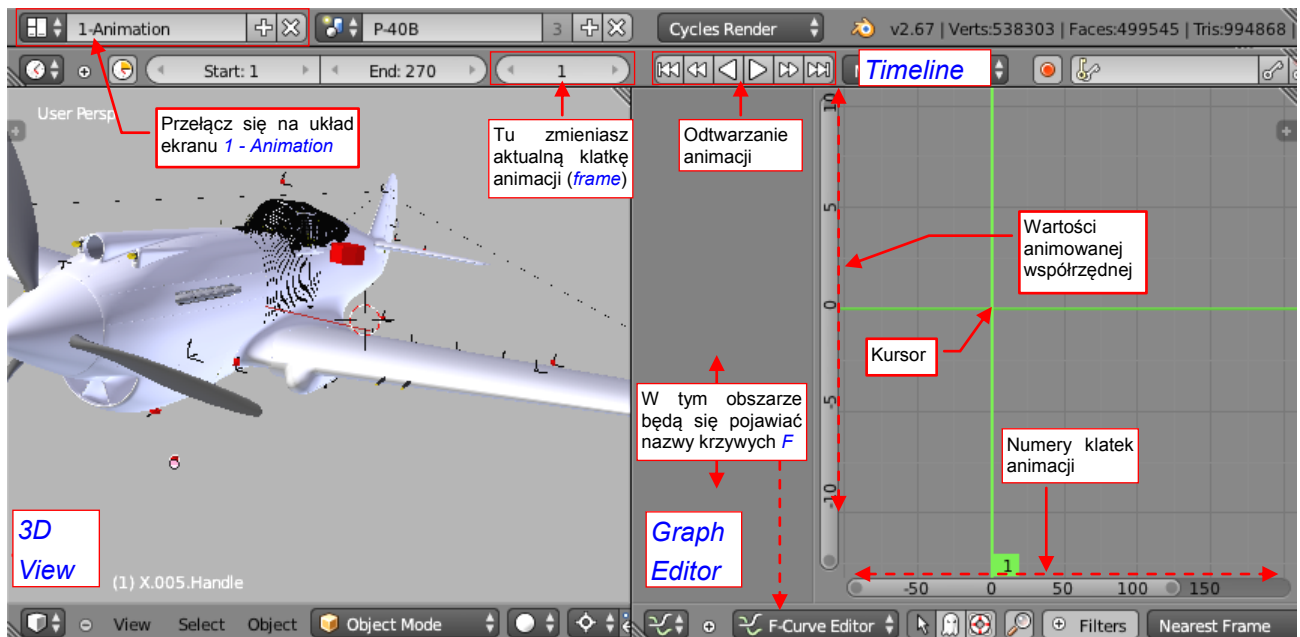
Potem możesz się już przełączyć w tryb *Texture Paint* i namalować odpowiednie plamy (Rysunek 10.1.9). Dzięki odpowiednim liniom pomocniczym możesz widzieć dokładnie, gdzie umieścić źródło śladów sadzy, oraz w jakim kierunku należy rozpraszać te zabrudzenia.



Rysunek 10.1.9 Malowanie okopceń za lufami karabinów

## 10.2 Animacja ruchu śmigła

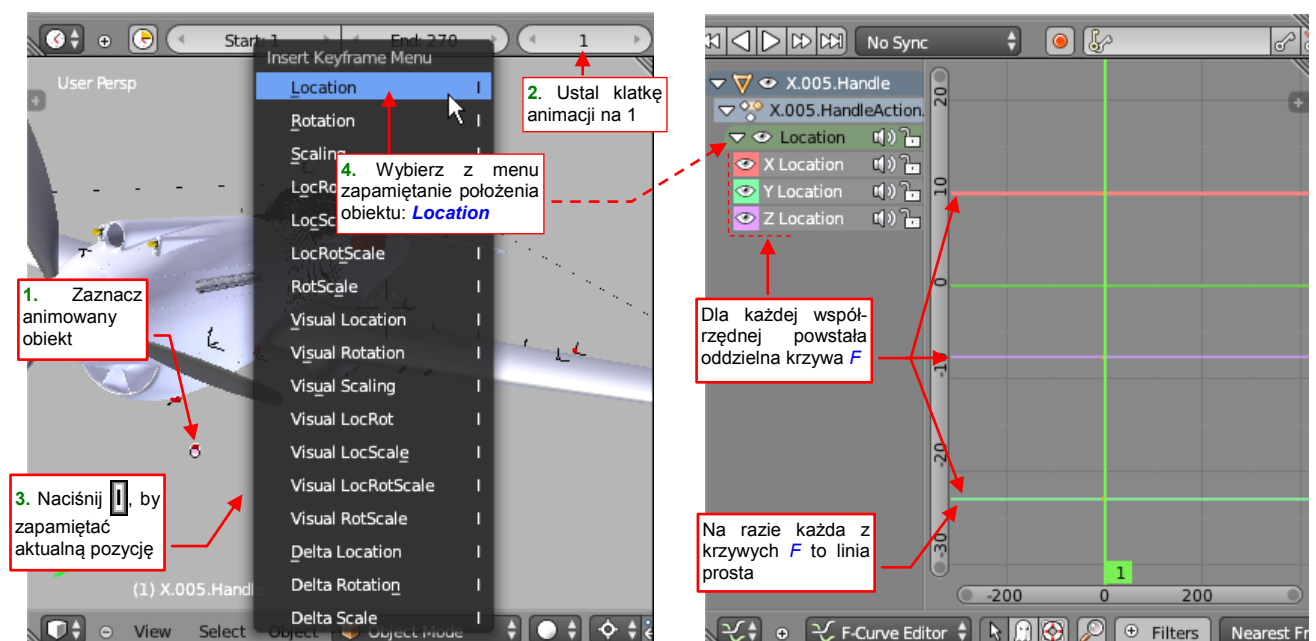
W tej sekcji pokażę, jak przygotować najprostszą animację. Będzie to jednostajny obrót śmigła. Aby zacząć, przełącz się na układ ekranu do animacji (**1-Animation** — Rysunek 10.2.1):



Rysunek 10.2.1 Układ okien Animation

Ruch na scenie Blendera jest podzielony na pojedyncze klatki (*frames*). W nagłówku okna *Timeline* znajduje się kontrolka, w której określasz aktualną klatkę (Rysunek 10.2.1). Animowany obiekt może zajmować w kolejnych klatkach różne położenia. Linia, łącząca współrzędne takich punktów, nosi nazwę *F-Curve*. W układzie *Animation* po prawej znajduje się okno *Graph Editor*, służące do pracy z krzywymi *F*. Liniją ruchu — *F-Curve* — najprościej jest stworzyć pokazując programowi kolejne położenia obiektu w kilku ważniejszych klatkach. Takie klatki nazywamy klatkami kluczowymi (*keyframes*).

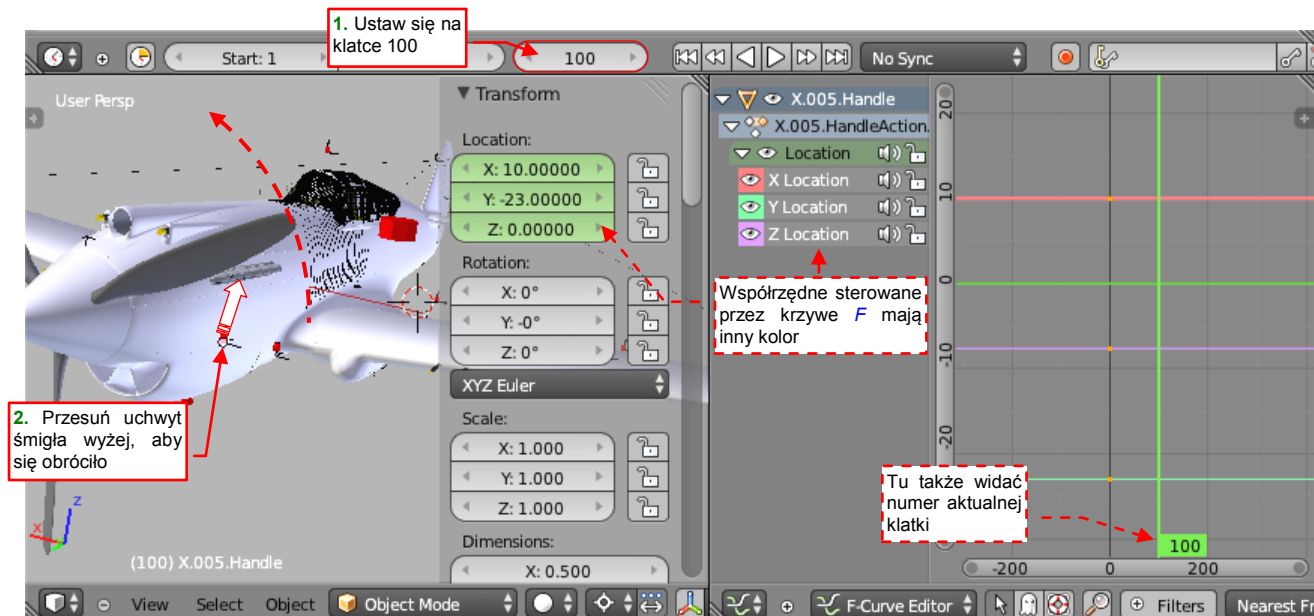
Zaznacz w widoku 3D obiekt, który ma być animowany. Ustal, wpisując w kontrolkę, numer aktualnej klatki (najlepiej zacznij od 1). Następnie naciśnij (koniecznie w widoku *3D View*!) klawisz **I** (Rysunek 10.2.2):



Rysunek 10.2.2 Wstawienie pierwszej „pozycji kluczowej” (*keyframe*)

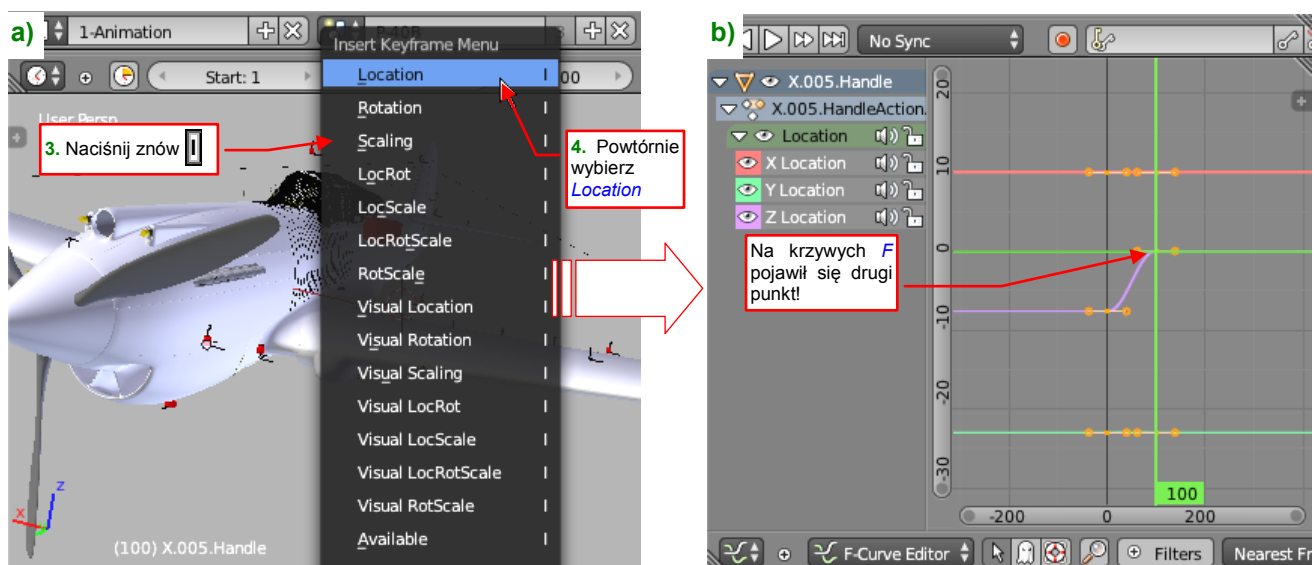
Z menu, które się pojawi, wybierz rodzaj współrzędnej, która ma się zmieniać (w tym przypadku to *Location*).

Następnie przejdź do kolejnej klatki kluczowej — powiedzmy nr 100 (Rysunek 10.2.3):



Rysunek 10.2.3 Przygotowanie drugiej pozycji kluczowej — sto klatek dalej

(Zwróć uwagę, że spowodowało to przesunięcie w oknie *Graph Editor* zielonej, pionowej linii — to nasz wskaźnik aktualnej klatki). Przesuń w tej klatce uchwyt sterujący obrotem śmigła (por. str. 113) o 7-8 jednostek Blendera. Następnie znów naciśnij w oknie *3D View* klawisz **I**. Pojawi się menu *Insert Keyframe*, z którego powtórnie wybierz *Location*<sup>1</sup> (Rysunek 10.2.4a):



Rysunek 10.2.4 Wstawienie kolejnej pozycji kluczowej (*keyframe*) — w klatce 100.

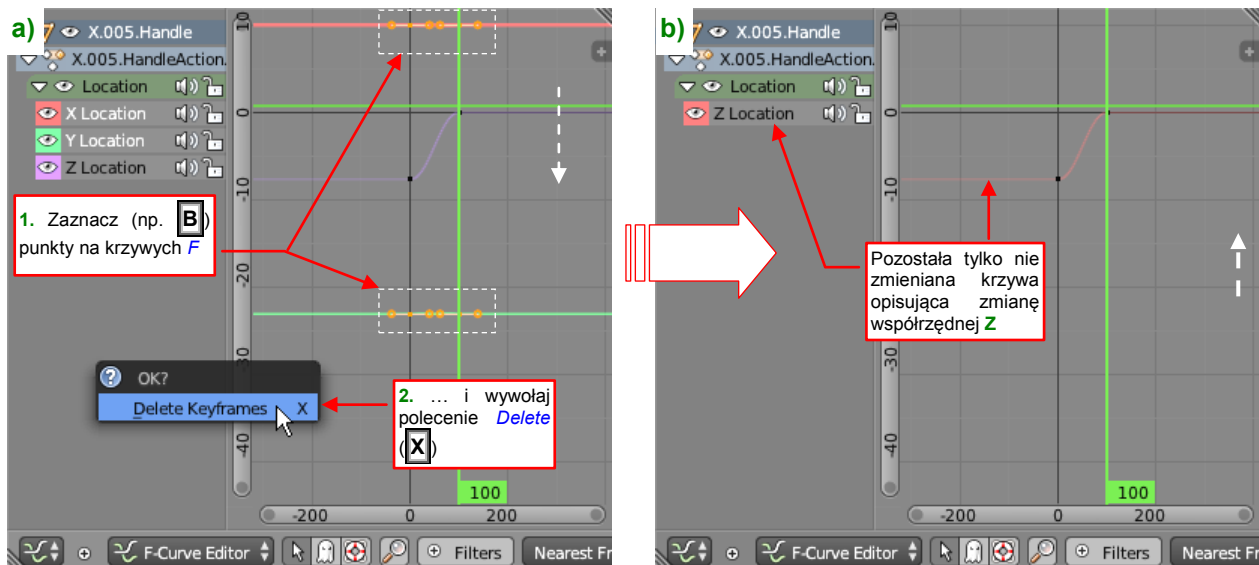
Tym razem spowodowało to dodanie do krzywych *F* drugiego punktu (Rysunek 10.2.4b). Linia z prostej stała się krzywą — dla parametru *Z Location*. Wartość tego położenia dla klatki 100 wynosi 0 (w to miejsce przesunąłeś uchwyt przed naciśnięciem **I**). Wygląda na to, że Blender zapamiętuje współrzędne w lokalnym układzie obiektu. Lokalna oś *Z* uchwytu śmigła biegnie wzdłuż globalnej osi *Z*, stąd taki efekt.

<sup>1</sup> Zwróć uwagę, że na początku menu *Insert Key* znajdują się podstawowe rodzaje współrzędnych: *Location* — położenie, *Rotation* — obrót, czy *Scale*. Wybór każdego z nich oznacza zapamiętanie odpowiednich 3 współrzędnych przestrzennych. Dalsze pozycje w menu dają możliwość jednoczesnego zachowania kombinacji tych rodzajów. *LocRot* oznacza *Location* + *Rotation*, *LocScale* to *Location* + *Scale*, a *LocRotScale* to oczywiście wszystkie współrzędne “za jednym zamachem”.



Dwa punkty wystarczą do poprowadzenia linii prostej, a tylko taka jest nam potrzebna do odwzorowania jednostajnego obrotu śmigła. Teraz proponuję zrobić małe porządki i usunąć z animacji linie **F** związane z przesunięciem wzdłuż osi **X** i **Y** — bo te parametry nie mają ulegać żadnym zmianom.

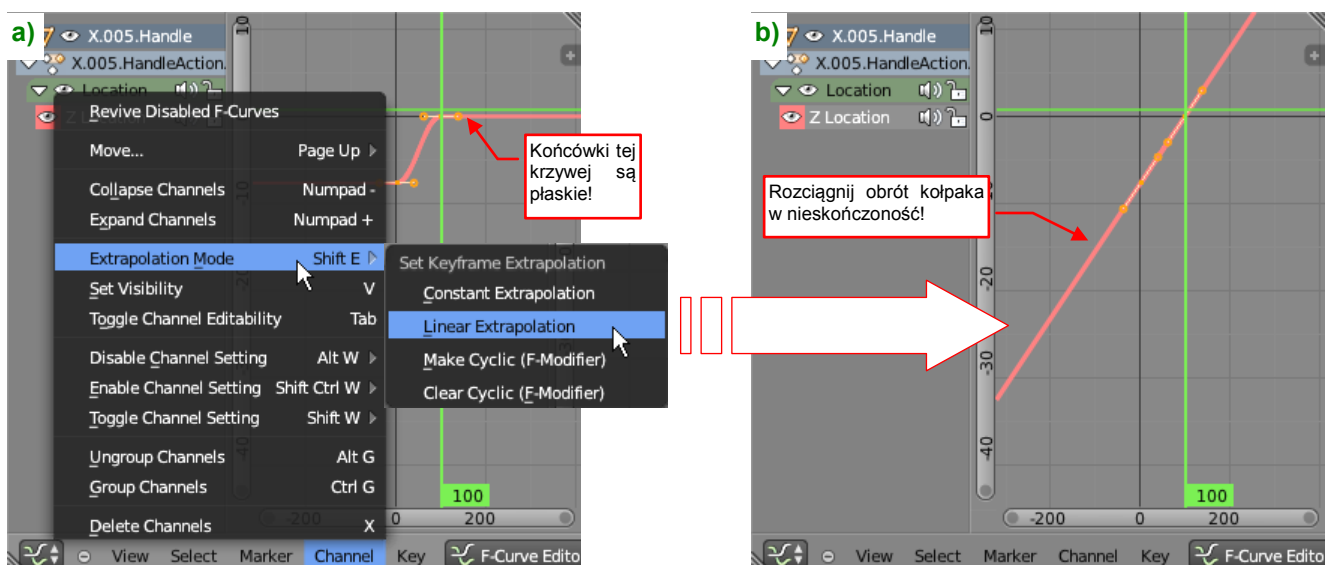
W edytorze krzywych **F** obowiązują podobne reguły do innych okien Blendera. Możesz zaznaczyć poszczególne linie zaznaczasz klikając w nie **PPM**. Na liście po prawej spowoduje to podświetlenie parametru, którego zmianę opisuje krzywa (Rysunek 10.2.5a):



Rysunek 10.2.5 Usuwanie niepotrzebnych współrzędnych krzywych **F**

Możesz także przejść do edycji punktów zaznaczonej krzywej (klawiszem **Tab**) — to odpowiednik trybu edycji siatki. Podobnie klawisz **X** (**Channel** → **Delete Channels**) powoduje usunięcie tego, co jest zaznaczone. Wskaż więc linię związaną z parametrem **X Location** i naciśnij **X**, by ją usunąć. Zrób to samo dla linii **Y Location**. Na ekranie krzywych **F** powinna pozostać tylko linia związana z „naszym” obrotem: **Z Location** (Rysunek 10.2.5b).

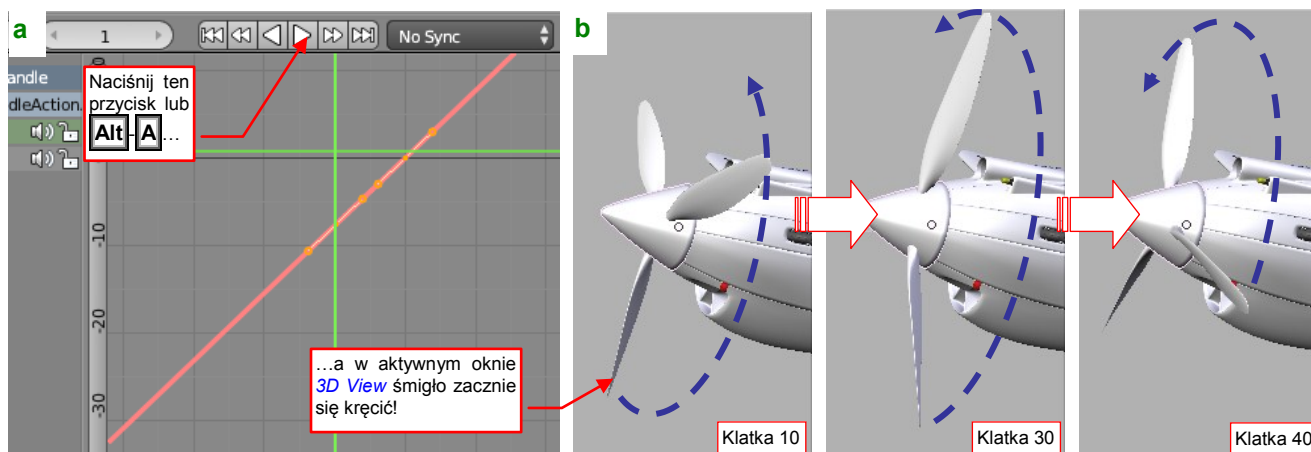
Dla takiego „wygiętego” kształtu krzywej śmigło obracałoby się od klatki 1 do 100, po czym cały ruch by ustał. Aby wymusić jednostajny obrót przez cały czas animacji, zaznacz tę linię, po czym przełącz jej **Extrapolation Mode** na **Linear Extrapolation** (Rysunek 10.2.6):



Rysunek 10.2.6 Końcowe dostosowania linii animacji

Teraz linia **Z Location** stała się pochyloną linią prostą — i o to chodziło. Gdybyś chciał zmienić nazwę zespołu krzywych **F** (np. z domyślnego **X.005.HandleAction** na **Propeller**) — musisz to zrobić w oknie **Outliner**.

Na koniec sprawdzamy „jak to się kręci”. Naciśnij w oknie **3D View** klawisze **Alt-A**. Możesz także kliknąć odpowiedni przycisk w oknie **Timeline** (Rysunek 10.2.7a). Nasze śmigło zacznie się obracać (Rysunek 10.2.7b):



Rysunek 10.2.7 Test animacji

- Animację możesz przerwać w każdym momencie, naciskając **Esc**.

Oczywiście, jeżeli obiekt nie porusza się tak, jak powinien, trzeba trochę pozmienić jego krzywe **F**. Potem znów sprawdzić, uruchamiając animację. I tak dopóty, dopóki nie uzyskasz zadowalającego efektu.

- Ograniczenia przypisane do obiektu obowiązują także dla ruchu wywołanego przez krzywe **F**.

Dlatego jeżeli chcesz, aby śmigło obracało się w nieskończoność, ustaw wartość **Influence** ograniczenia **Limit Location** animowanego w tej sekcji uchwytu **X.005.Handle** do 0. Jak się przekonasz, do wykonania pojedynczego ujęcia nie jest to potrzebne (wystarczy, aby śmigło obróciło się o 30-60°).

- Po naciśnięciu **Alt-A** Blender zacznie odtwarzanie animacji tylko w aktywnym oknie **3D View**. Jeżeli nawet na ekranie masz więcej okien tego typu, nie patrz na inne — tam nic się nie będzie poruszać!

### 10.3 Rozmycie ruchu śmigła

Obrotem śmigła w naszym modelu steruje przesunięcie uchwytu **X.005.Handle** (por. str. 113), któremu przypisa-  
liśmy odpowiedni tor ruchu wzdłuż osi **Z** (**F-Curve** — por. str. 425). W tej sekcji omówimy jak ustawić ścieżkę  
ruchu uchwytu. Celem jest uzyskanie na renderze takiego obrazu rozmytych łopata śmigła, jakie widać na zdję-  
ciach samolotu w locie.

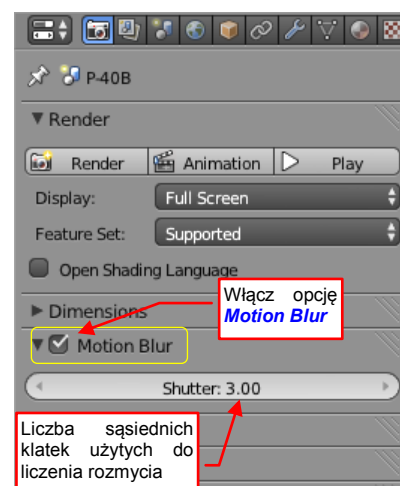
Gdy zaczniemy od odwzorowania wartości fizycznych, to będzie wyglądać tak:

- obroty silnika w trakcie lotu to około 1800 – 2400 obr/min;
- przełożenie reduktora śmigła to około 1:2 (zależne od konstrukcji), stąd prędkość obrotowa śmigła to  
około 900 – 1200 obr/min, czyli 15-20 obr/s;
- prędkość typowej animacji to 20 klatek/s;
- pełen obrót śmigła to przesunięcie uchwytu sterującego **X.005.Handle** o 10 jednostek Blendera (tak to  
ustawiliśmy w ograniczeniu **Transformation** kołpaka śmigła);

Oznacza to, że aby odwzorować prędkość 15-20 obr/s, pomiędzy każdą klatką śmigło musi wykonać od 0.75 do  
1 obrotu. Taki efekt uzyskamy przesuwając uchwyt **X.005.Handle** o 7.5 – 10 jednostek Blendera w każdej klatce  
To oznacza znaczne pochylenie krzywej ruchu (Rysunek 10.3.1):



Rysunek 10.3.1 Ustawienia krzywej ruchu (**F-Curve**) uchwytu sterującego obrotem koł-  
paka śmigła



Rysunek 10.3.2 Cycles — włączenie rozmycia ruchu

Włącz w zestawie **Render**, panelu **Motion Blur** efekt rozmycia ruchu dla Cycles (Rysunek 10.3.2). Ta panel zawiera tylko jeden parametr — **Shutter**. To liczba sąsiednich klatek animacji, na z których obrazów Cycles obliczy średnią (tak się właśnie liczy rozmycie ruchu). Na razie pozostaw tu wartość domyślną (**Motion Blur:Shutter** = 3). Cycles nie poka-  
zuje rozmycia ruchu na podglądzie renderu. Dlatego tym razem musisz nacisnąć **F12** (albo nacisnąć przycisk **Render**) aby wykonać pełen render naszego modelu.

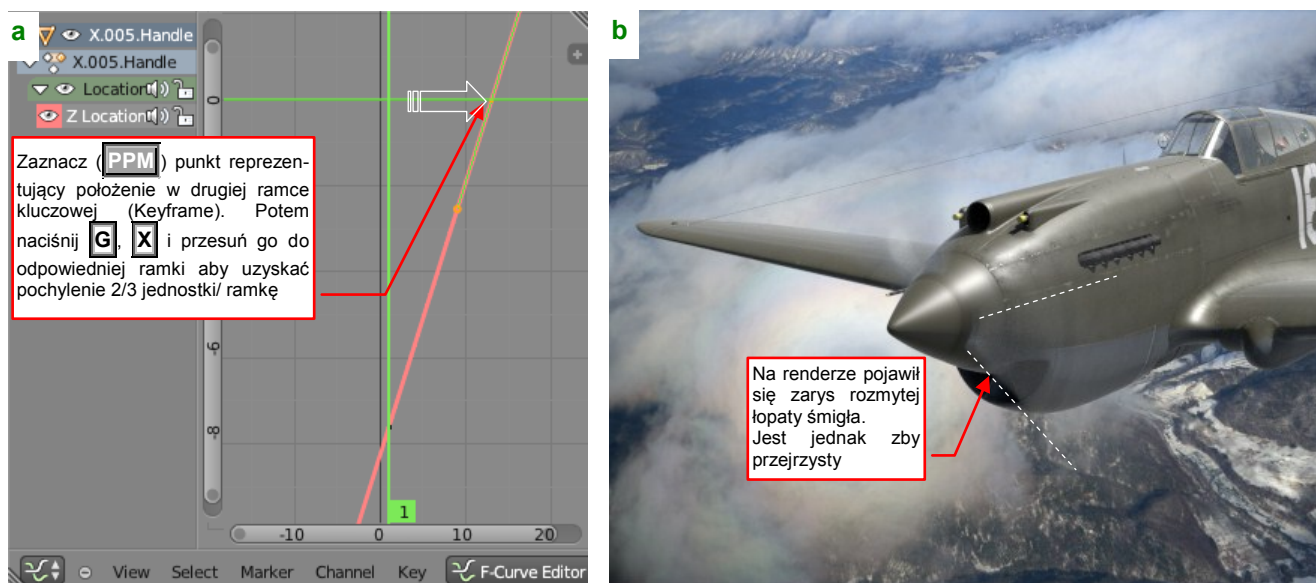
Rysunek 10.3.3 przedstawia rezultat — śmigło wirowało tu tak szybko, że stworzyło jed-  
nolity, prawie zupełnie przejrzysty krąg. W istocie – tak widzimy wirujące śmigło naszymi  
ludzkimi oczami.



Rysunek 10.3.3 Pierwszy rezultat — całkowicie rozmyty krąg śmigła

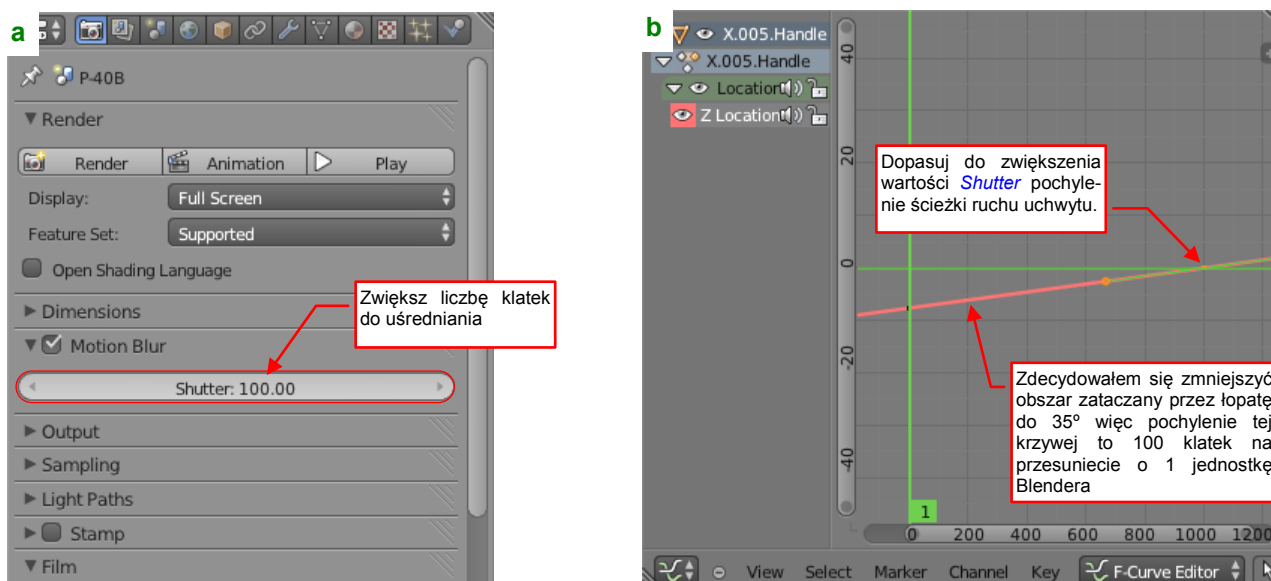
- Zawsze powinieneś się zastanowić, czy chcesz zrobić wizualizację tak, jakby była fotografią, czy tak jak widziałyby Twoją scenę ludzkie oczy. Czasami widzimy rzeczy w inny sposób niż nasze aparaty!

Nie wiem czy zastanawiałeś się, jak bardzo zdjęcia przyzwyczyły nas do pewnych zniekształceń. Na renderze który pokazuje Rysunek 10.3.3 odruchowo szukamy charakterystycznych, rozmytych łopat śmigła. Jesteśmy po prostu do nich od przyzwyczajeni! Aby uzyskać taki efekt, musimy spowolnić ruch śmigła. Im krótszy czas przesłony wykonywanego zdjęcia, tym mniejszy łuk zdążą w tym czasie zatoczyć łopaty. Na zdjęciach z czasów I Wojny Światowej i z lat 20-tych XX w. widać jeszcze całe kręgi śmigieł. To dlatego, że stosowano wtedy stosunkowo długie przesłony. Później pojawił się lepszy sprzęt i wydaje mi się, że samoloty z czasów II Wojny Światowej były fotografowane z przesłoną 1/50 lub nawet 1/100 sekundy. 1/100 z 20 obr/s to 1/5 obrotu (72°). Taki ruch uzyskamy przesuwając uchwyt o 2 jednostki Blendera. Jeżeli nadal chcemy uzyskać rozmycie z uśrednienia 3 sąsiednich klatek, to oznacza że pomiędzy każdą kolejną klatką uchwyt **X.005.Handle** powinien się przesunąć o 2/3 jednostki Blendera. Musimy więc odpowiednio spłaszczyć naszą linię ruchu (Rysunek 10.3.4a):



Rysunek 10.3.4 Spowolnienie ruchu śmigła

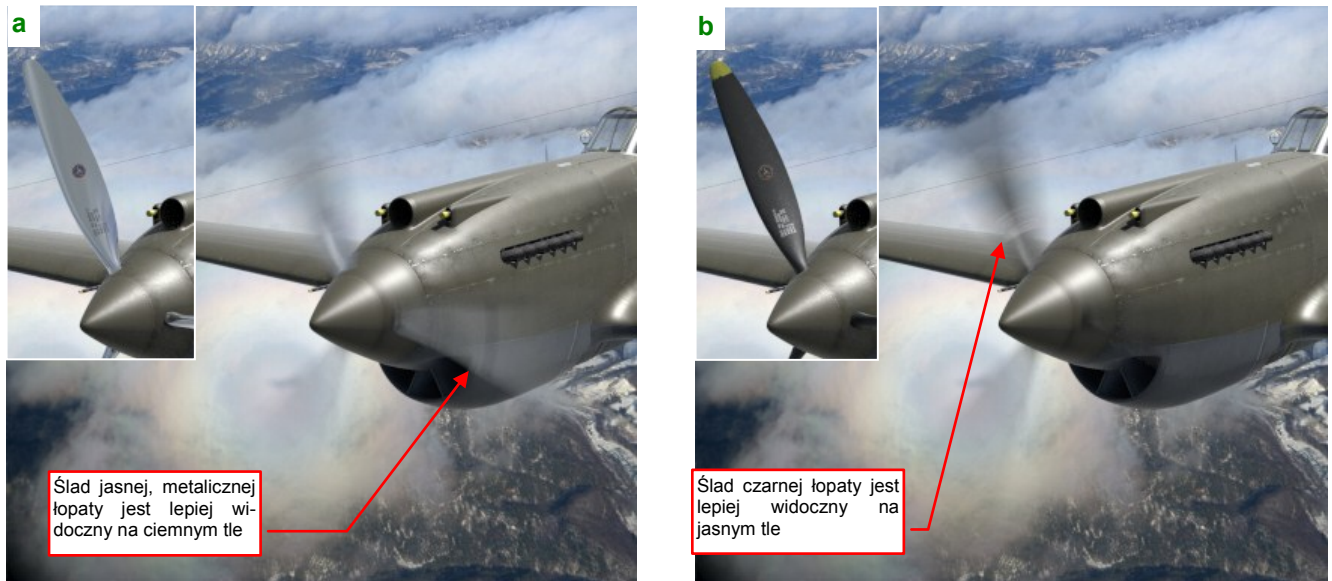
Aby „zagęścić” to rozmycie, trzeba umieścić w tym obszarze więcej uśrednianych obrazów. Dlatego zwiększymy parametr **Shutter** od 100 (Rysunek 10.3.5a). Odpowiednio spowolnimy ruch uchwytu, by przesunięcie o 2 jednostki Blendera zajmowało mu 100 lub więcej kolejnych ramek (Rysunek 10.3.5b):



Rysunek 10.3.5 Zwiększenie parametru **Shutter** i dalsze spowolnienie obrotu śmigła



Wydaje mi się że dla tych ustawień uzyskujemy odpowiedni rezultat. Dla lepszego porównania pokazuję to samo śmigło z dwoma typowymi materiałami. Rysunek 10.3.6a) przedstawia obraz wirujących łopatek z wypolerowanego duraluminium, a Rysunek 10.3.6b) — czarne („antyodblaskowe”) łopaty z żółtą końcówką:



**Rysunek 10.3.6** Rezultat – rozmycie ruchu dla dwóch różnych materiałów śmigła

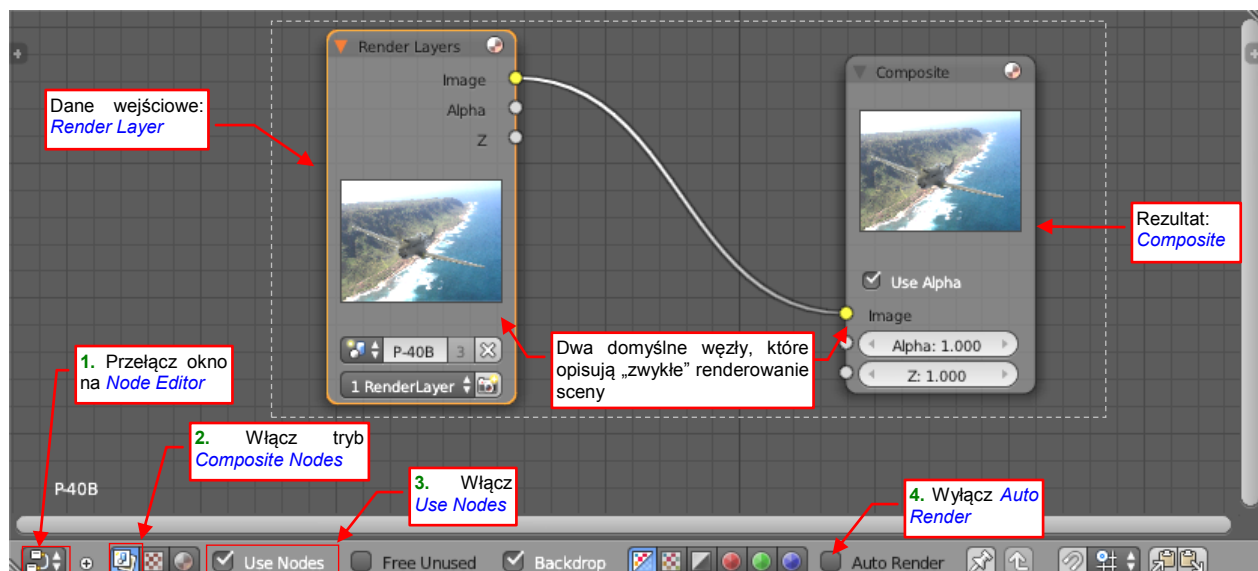
- Tworząc filmy możesz stosować zamiast łopatek wymodelowany w siatce dysk, pokryty materiałem o bardzo dużej przejrzystości. Wystarczy mu nadać powolny obrót (i nie włączać opcji [Motion Blur](#)).

(W odróżnieniu od „statycznych” wizualizacji po których widz spodziewa się fotorealizmu, w filmie oczekujemy raczej efektów zbliżonych do postrzegania świata przez nasze własne oczy).

## 10.4 Kompozycja rezultatu (*Composite Nodes*)

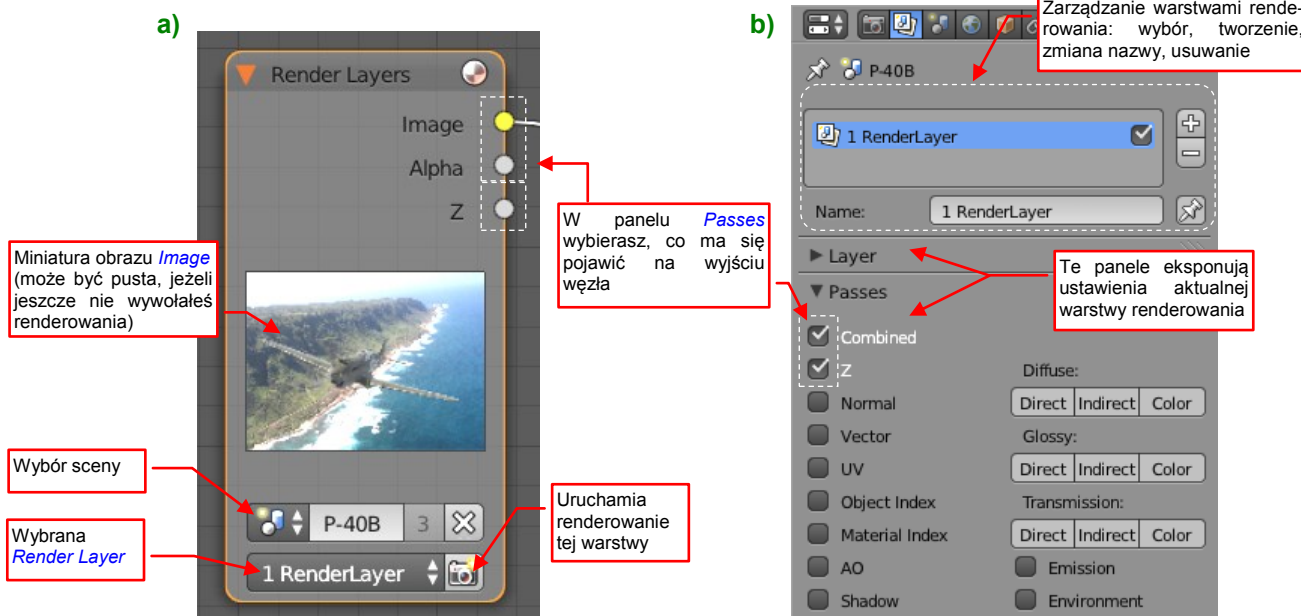
W tej sekcji zademonstruję zastosowanie edytora węzłów do kompozycji ostatecznego obrazu. Węzły, które to umożliwiają, nazywa się w Blenderze „węzłami kompozycji” (*Composite Nodes*). Jako przykładu użyjemy tu modelu omawianego na str. 133 i dalszych ([p40history\p-40.9.01w3.blend](#)).

Przełącz jedno z okien Blendera na *Node Editor* (Rysunek 10.4.1):



Rysunek 10.4.1 Edytor kompozycji i jego dwa domyślne węzły

Aby pojawiły się w nim dwa domyślne węzły, opisujące „zwykły” rendering, musisz przełączyć się w tryb *Composite Nodes* i włączyć *Use Nodes* (Rysunek 10.4.1). Po lewej widzisz węzeł typu *Render Layer*. To dane wejściowe. Po prawej znalazł się węzeł typu *Composite*. To ostateczny obraz. Te węzły przekazują sobie (linia łącząca) piksele obrazu (*Image*). Dane wejściowe może dostarczać na naszym schemacie węzeł *Render Layer* (Rysunek 10.4.2a):



Rysunek 10.4.2 Węzeł wejściowy *Render Layers* i zestaw *Render Layers* w oknie *Properties*

Co to właściwie jest, ta „warstwa renderowania”? To zespół parametrów, opisujących szczegółowo, co ma robić (i czego nie robić) wewnętrzny renderer (w tym przypadku Cycles). Domyślnie w każdej scenie istnieje jedna, domyślna warstwa renderowania, o nazwie **1 Render Layer**. Szczegóły jej ustawień możesz obejrzeć w zestawie *Render Layers* (Rysunek 10.4.2b).

Wydaje mi się, że nazwa **Render Layer** niezbyt oddaje istotę rzeczy. W istocie jest to opis oddzielnego „przebiegu” renderowania sceny. Wykorzystując kontrolki z zestawu **Render Layers** możesz przygotować wiele alternatywnych obrazów, które później wykorzystasz jako dane wejściowe kompozytora.

U góry zestawu **Render Layers** znajduje się lista, z której możesz wybrać jedną z istniejących warstw renderowania, lub zdefiniować nową (Rysunek 10.4.2b). To typowa lista, jaką spotkałeś przy okazji pracy z innymi rodzajami danych. Gdy ją rozwiniesz, zobaczysz spis istniejących elementów. Przyciski z boku pozwalają stworzyć nowy obiekt lub usunąć bieżący, a pole poniżej — zmienić nazwę. Każdą ze zdefiniowanych w scenie warstw renderowania można użyć w oddzielnym węźle **Render Layer** kompozytora, tworząc z ich połączeń różne zaawansowane efekty.

Panel **Render Layers:Passes** (Rysunek 10.4.2b) zawiera zestaw przełączników, pozwalających wybrać wyjścia udostępniane przez węzeł **Render Layer** (Rysunek 10.4.2a). Przełącznik **Combined** wyświetla powoduje udostępnienie kompletnego obrazu (wyjście **Image**) i ewentualnie przejrzystego tła (**Alpha**). Pozostałe opcje pozwalają użyć informacji cząstkowych, wykorzystywanych przez renderer do stworzenia obrazu. Na przykład **Z** dostarcza informacji o odległości od obserwatora każdego piksela obrazu (to liczba zmiennoprzecinkowa).

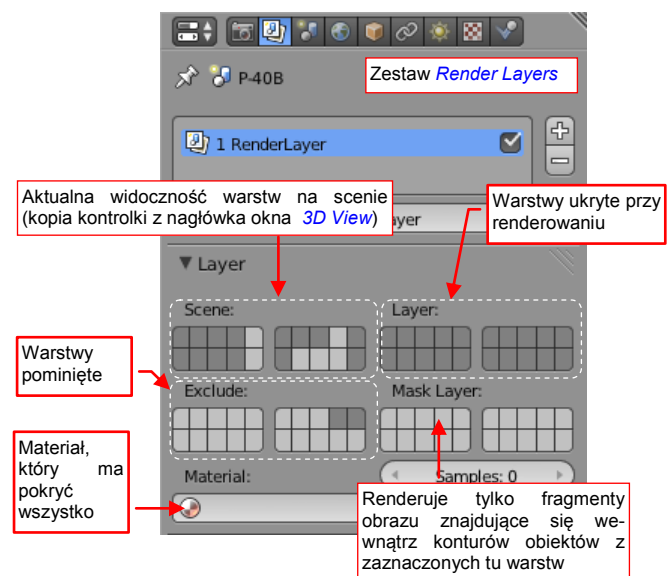
Z kolei panel **Render Layers:Layers** (Rysunek 10.4.3) pozwala wybierać warstwy sceny, które mają być użyte podczas renderowania obrazu.

Sekcja **Scene** to widoczność warstw sceny (tę samą kontrolkę masz dostępną w nagłówku okna **3D View**). W odróżnieniu od pozostałych sekcji tej paneli, **Scene** nie jest częścią definicji warstwy renderowania. Warstw (tych z obiektami), które na niej wyłączysz, nie będziesz widział w widoku 3D.

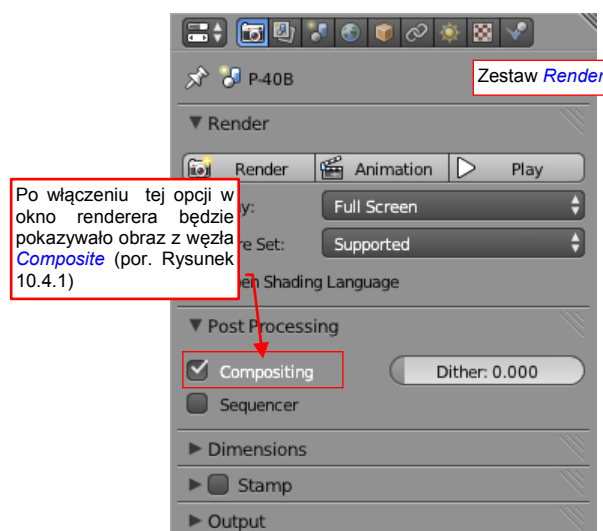
Sekcja **Layer** to maska, pozwalająca wyłączyć na renderze widoczność wybranych warstw. Blender renderuje tylko warstwy, które są jednocześnie włączone i w sekcji **Scene**, i w sekcji **Layer**. Pamiętaj jednak, że jeżeli obiekt na takiej ukrytej warstwie rzuca cień, to ten cień będzie widoczny! Aby tak nie było — czyli aby zupełnie wykluczyć warstwę z renderowania — zaznacz ją w sekcji **Exclude** (Rysunek 10.4.4).

Zwróć uwagę że w przykładzie przedstawionym na ilustracji wykluczyłem z renderowania warstwy z uchwytami i obiektami pomocniczymi (9,10). W ten sposób mogę wygodnie pracować nad krzywą ruchu uchwytu śmigła, bez konieczności ciągłego wyłączania warstwy 10 przed renderowanie. (Aby zobaczyć krzywą ruchu uchwytu, muszę zaznaczyć w widoku **3D View** ten obiekt).

Domyślnie Blender przedstawia w oknie renderera „surowy” rezultat bieżącej warstwy renderowania. Aby przycisk **Render** wywoływał cały przebieg ustalony w kompozytorze, musisz w



Rysunek 10.4.3 Panel **Layers**: wybór warstw do renderowania

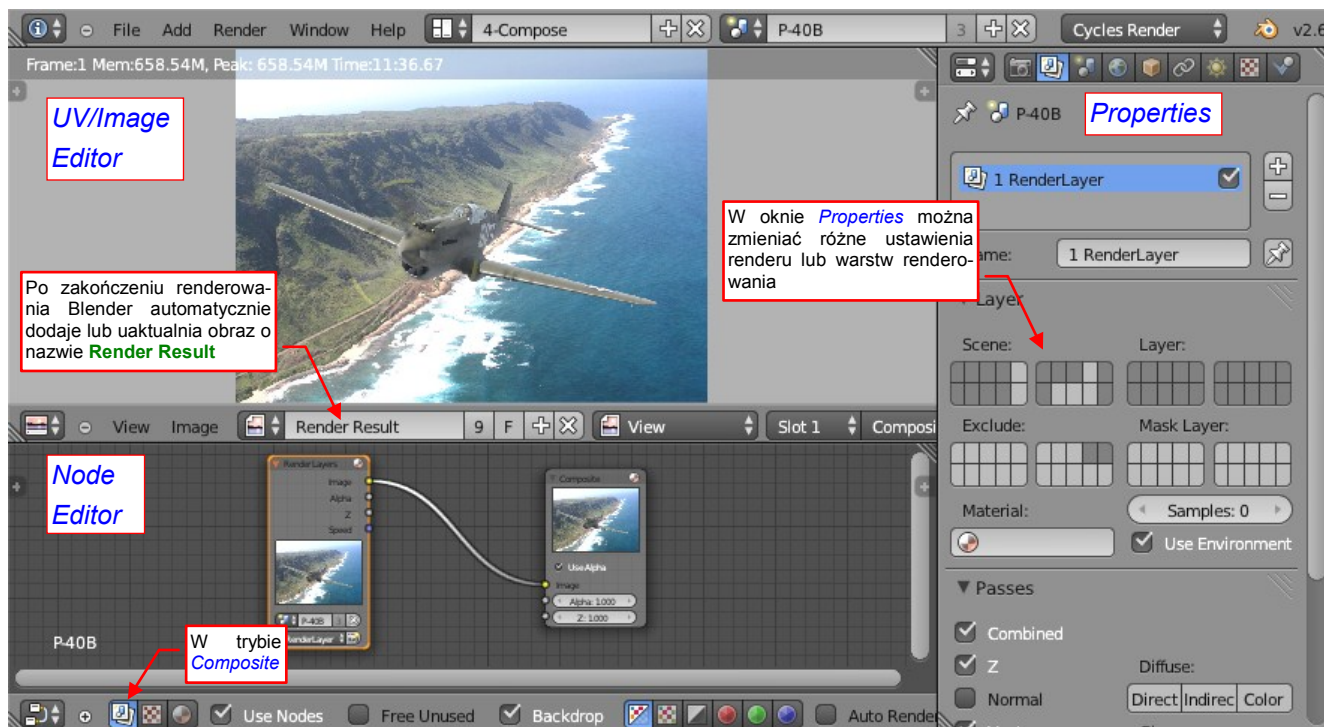


Rysunek 10.4.4 Włączenie kompozytora do przebiegu renderowania (automatycznie przetworza rezultat renderu)



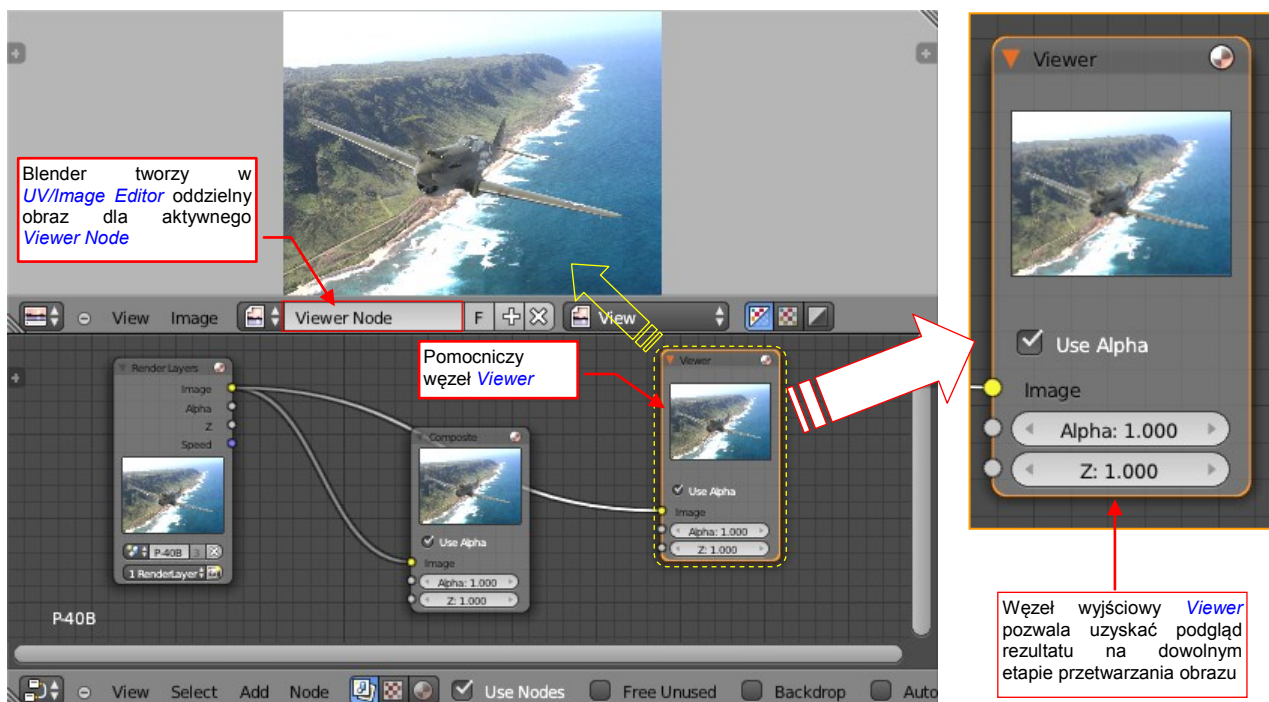
panelu **Post Processing** zestawu **Render** zaznaczyć opcję **Compositing** (Rysunek 10.4.4).

Wróćmy teraz do naszej kompozycji. Rysunek 10.4.5 przedstawia układ trzech okien, który stosujemy do pracy z jej węzłami:



Rysunek 10.4.5 Układ ekranu do tworzenia schematu kompozycji

Okno **Properties** wykorzystujemy do zmiany ustawień (zazwyczaj renderowania). W oknie **UV/Image Editor** mamy podgląd obrazów. Zazwyczaj są to specjalne tymczasowe obrazy, które Blender uaktualnia na bieżąco ale nie zachowuje do pliku **\*.blend**. Pierwszym z nich jest **Render Result**, zawierający rezultat ostatniego renderowania. Aby uzyskać drugi, musisz dodać do schematu co najmniej jeden pomocniczy węzeł wyjściowy typu **Viewer** (Rysunek 10.4.6):



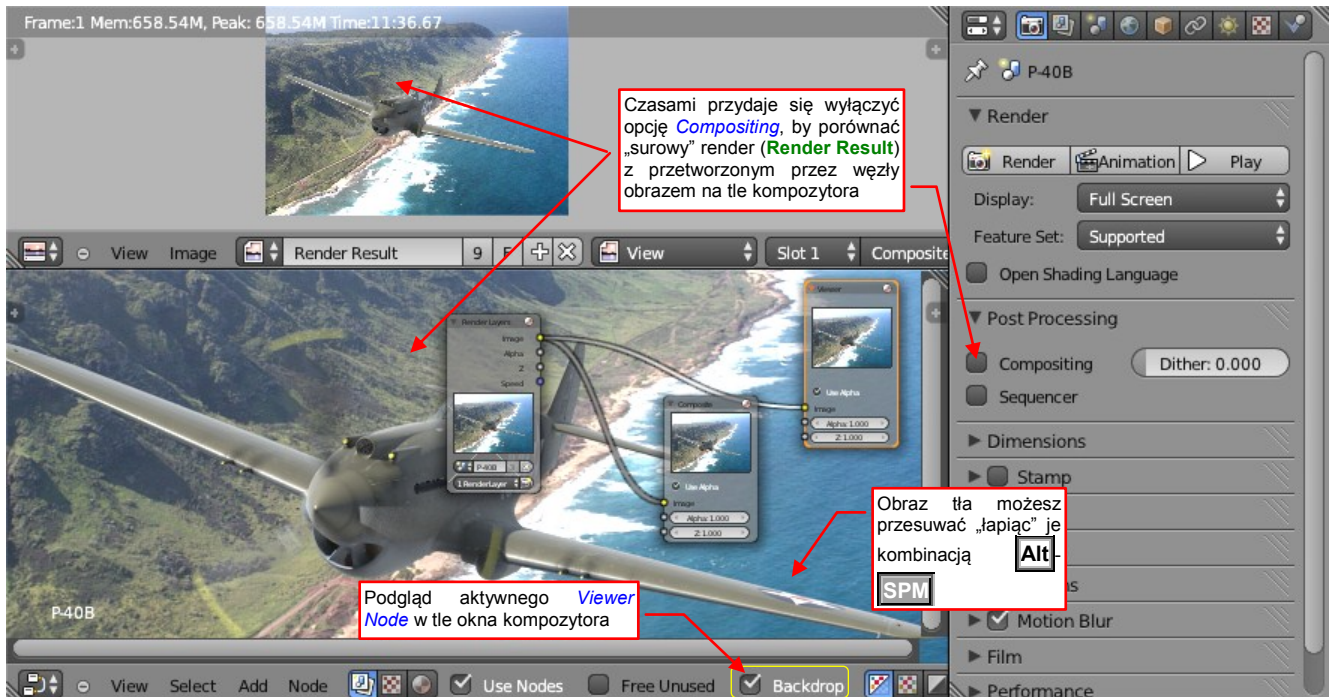
Rysunek 10.4.6 Pomocniczy węzeł **Viewer**



Blender dodaje wówczas do listy obrazów dodatkową pozycję: **Viewer Node**. Możesz ją otworzyć w *UV/Image Editor* i dokładnie sprawdzić rezultat jakiegoś etapu przetwarzania obrazu.

- Czasami węzeł **Viewer** nie wyświetla od razu podłączonego obrazu. Zazwyczaj dzieje się tak gdy wstawiłeś pierwszy taki element do schematu. Zazwyczaj najłatwiej jest wymusić jego odświeżenie odłączając i przyłączając ponownie wejście jakiegoś węzła który go poprzedza.

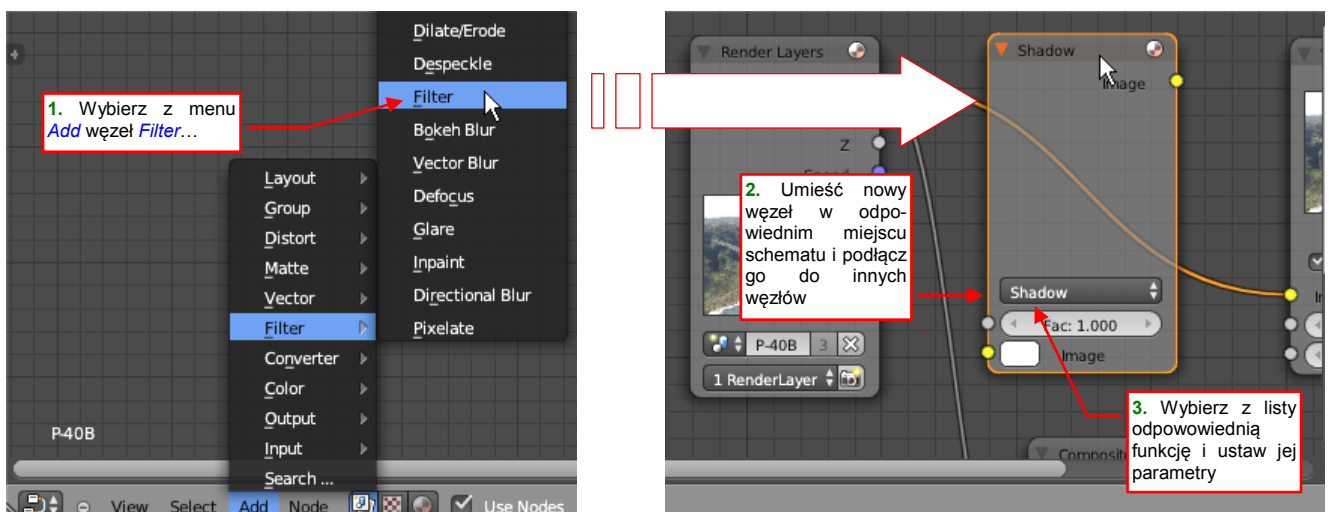
Z węzłem **Viewer** wiąże się dodatkowa ciekawa opcja **Backdrop**. Włącz ją w nagłówku *Node Editor*, a Blender wyświetli zawartość aktualnego węzła **Viewer** w tle okna kompozytora (Rysunek 10.4.7):



Rysunek 10.4.7 Włączenie obrazu **Backdrop**

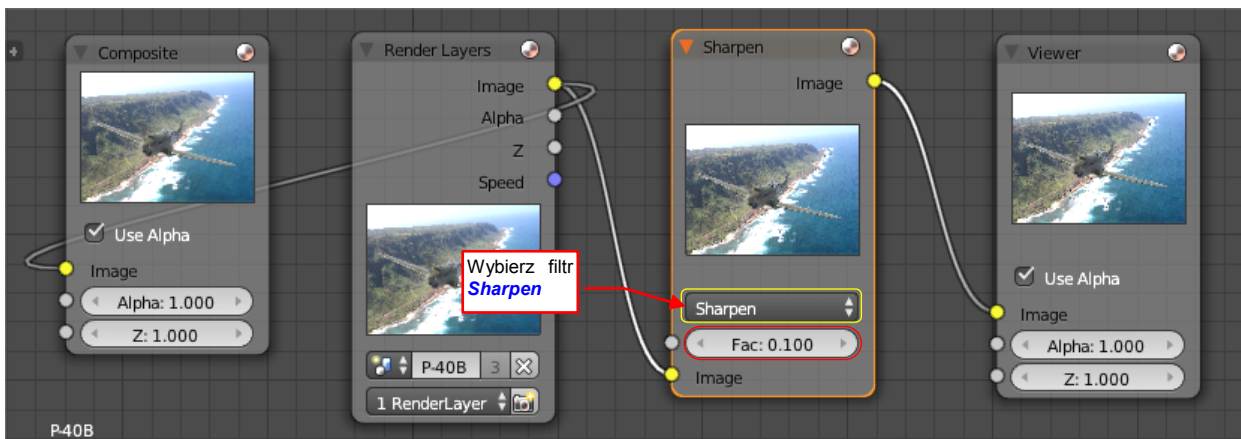
Co prawda takiego obrazu nie można zmniejszać ani zwiększać. Możesz jedynie przesunąć to tło (trzymając na klawiaturze wciśnięty klawisz **Alt**, „złap” tło **SPM** i przeciągnij w nowe miejsce). Trzeba przyznać, że edytor węzłów z włączoną funkcją **Backdrop** robi wrażenie (możesz to przetestować na znajomych). W praktyce przydawał mi się do porównań przetworzonego obrazu z „surowym” renderem. (Ten drugi wyświetlałem w oknie *UV/Image Editor* — por. Rysunek 10.4.7).

Aby jednak było co porównywać, dodajmy do naszego schematu jakiś efekt. Obraz stworzony przez render wydaje się być trochę rozmyty, więc wyostriamo go nieco za pomocą węzła **Filter:Sharpen** (Rysunek 10.4.8):



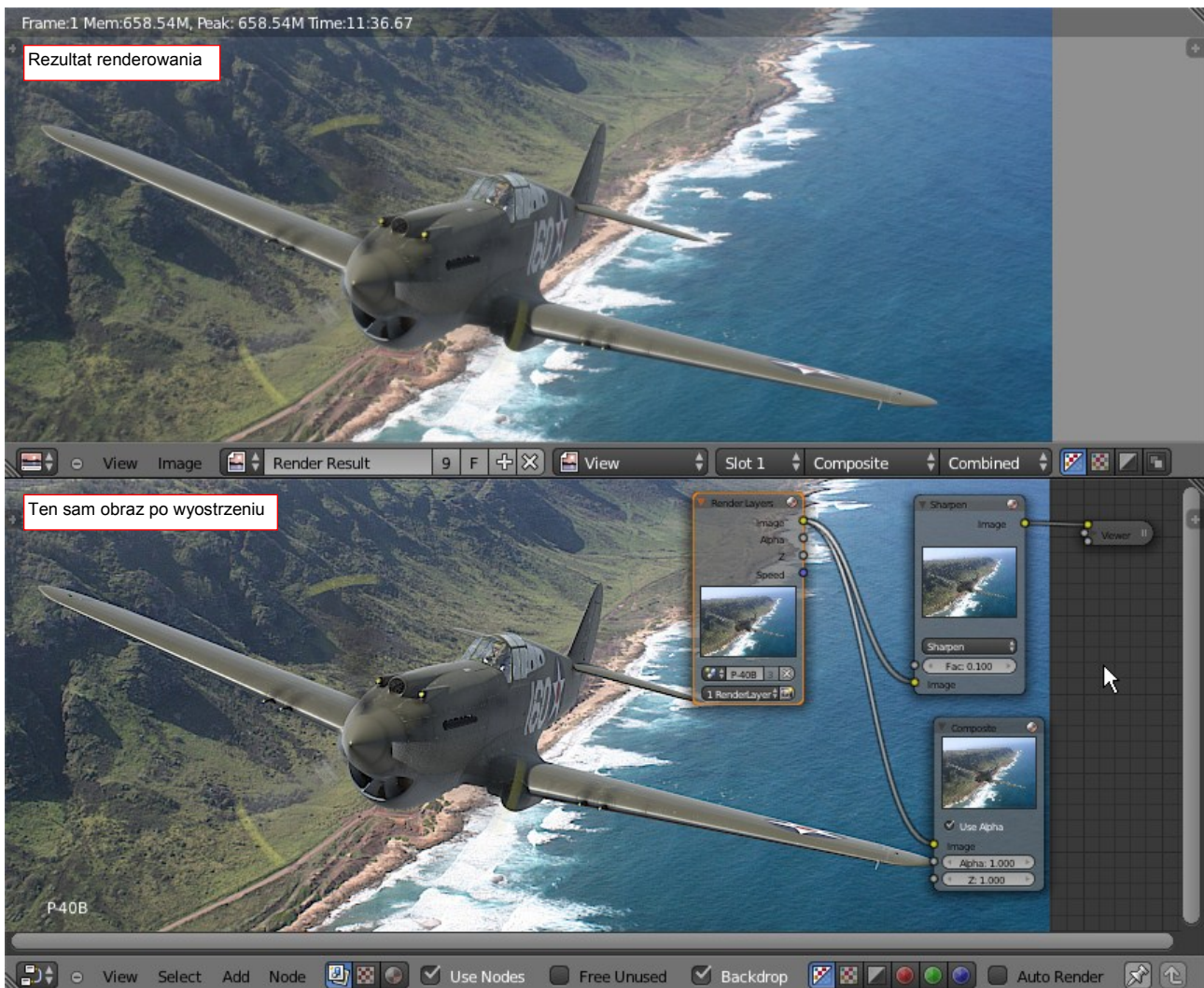
#### Rysunek 10.4.8 Wstawienie węzła **Filter**

Wybierz w węźle **Filter** funkcję **Sharpen**, i zmniejsz jej współczynnik **Fac** do 0.1 (Rysunek 10.4.9):



#### Rysunek 10.4.9 Wyostrzenie renderowanego obrazu

Porównaj rezultat „przed” i „po” (Rysunek 10.4.10):



#### Rysunek 10.4.10 Porównanie rezultatu z oryginalnym renderem

Filtr **Sharpen** zastosowany „z pełną mocą” (**Fac** = 1.0) zamienia nasz obraz w dzieło ekspresjonisty. Jednak wydaje mi się, że takie niewielkie wyostrenie zwiększa realizm sceny. Praca z kompozytorem wygląda właśnie w ten sposób: dodajesz jakiś efekt, podłączasz go do węzła **Viewer** i sprawdzasz, czy obraz wygląda teraz le-



piej. W tekście głównym pokażę kilka typowych narzędzi do poprawiania finalnego efektu. (Ten etap pracy graficy określają jako *postprocessing*. Osobiście uważam go za okazję do doskonałej zabawy barwami!).

- Obrazy **Render Result** i **Viewer Node** możesz zapisać na dysk jako zwykłe pliki rastrowe. Wystarczy w *UV/Image Editor* wybrać polecenie *Image → Save As....*

Kompozytor to ciekawe narzędzie, które możesz także wykorzystywać do modyfikacji dowolnych obrazów rastrowych, a nie tylko rezultatów renderowania. Wystarczy użyć zamiast *Render Layer* węzła wejściowego typu *Image* (Rysunek 10.4.11):



Rysunek 10.4.11 Przykład modyfikacji plików rastrowych w Blenderze

Podłącz do takiego schematu co najmniej jeden węzeł *Viewer*, bo w tym przypadku Blender tworzy w *UV/Image Editor* tylko obraz **Viewer Node**. (Stąd podczas przetwarzania samych obrazów rastrowych nie ma potrzeby stosowania węzła *Composite*).

- Aby nie wykonywać renderowania przed każdym użyciem kompozytora, możesz zapisać „surowe” obrazy renderu (i jakieś inne pomocnicze maski) na dysku jako pliki rastrowe. Potem wystarczy dodać ich węzły wejściowe do kompozytora i poddać dalszej obróbce.

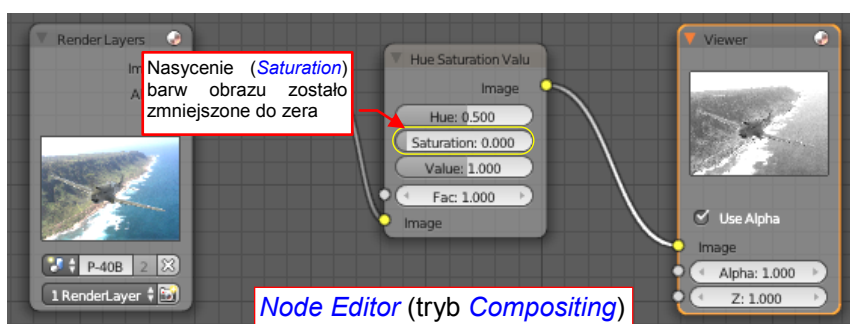
## 10.5 Zmiana odcienia obrazu (*Color Balance*)

Zmiana odcieni obrazu może poprawić efekt uzyskany z renderu. Pokażę to na przykładzie następującego obrazka (Rysunek 10.5.1):



Rysunek 10.5.1 Obraz początkowy

Aby lepiej pokazać, w czym rzecz, zastąpmy na razie barwy odcieniami szarości (Rysunek 10.5.2):



Rysunek 10.5.2 Wstępne przekształcenie obrazu (w kompozytorze Blendera)

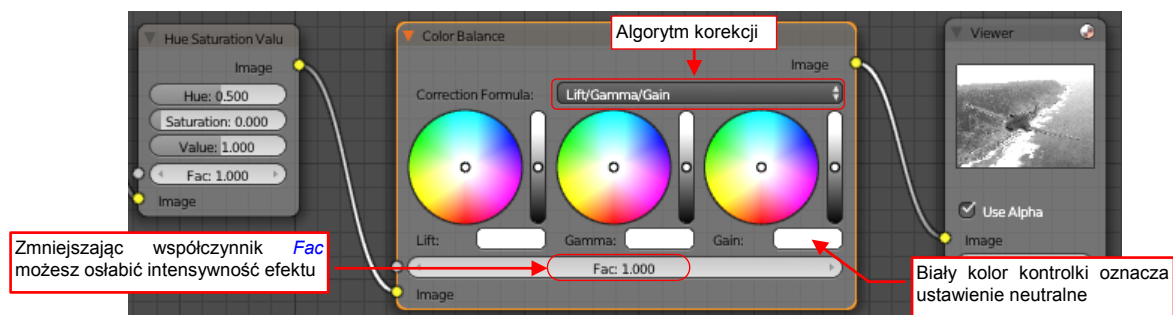
Oto ten sam obraz — teraz wygląda jak czarno-biała fotografia (Rysunek 10.5.3):



Rysunek 10.5.3 Ten sam obraz w wersji monochromatycznej



Wstawmy teraz w schemat naszej kompozycji węzeł **Color** → **Color Balance** (Rysunek 10.5.4):



Rysunek 10.5.4 Węzeł **Color Balance**

Sama nazwa węzła sugeruje jego zastosowanie do osiągnięcia „równowagi barw”. W obróbce zdjęć oznacza to korektę, mającą na celu odtworzenie na fotografii właściwych barw neutralnych — bieli i odcieni szarości. W naszym przypadku wykorzystamy ten węzeł do uzyskania odwrotnego efektu: niewielkich zmian odcieni, poprawiających wygląd obrazu. Technika wizualizacji, którą przedstawiam w tej książce, opiera się na podstawianiu za modelem jakiegoś zdjęcia. Wykonane za pomocą węzła **Color Balance** subtelne „sprowadzenie do wspólnego odcienia” modelu i tła może zwiększyć wrażenie że stanowią jedną całość. (W grafice komputerowej używa się często określenia „poprawienie integracji z tłem”).

**Color Balance** zawiera trzy kontrolki barw, o nazwach **Lift**, **Gamma** i **Gain**. Te nazwy są związane ze stosowanym przez węzeł algorytmem korekcji<sup>1</sup>. Traktuj wartość każdej z tych barw jako coś, co jest odejmowane lub dodawane do barw obrazu. O „znaku” w tej arytmetyce barw decyduje intensywność koloru kontrolki, którą można zmieniać w zakresie od 0.0 do 2.0. Gdy kontrolka węzła **Color Balance** ma kolor biały (i intensywność  $V = 1.0$ ) to nie zmienia obrazu wejściowego. W opisach algorytmu **Lift/Gamma/Gain** możesz znaleźć że:

- zmiana barwy **Lift** zmienia ciemne barwy obrazu — czyli głównie cienie;
- zmiana barwy **Gamma** zmienia pośrednie barwy obrazu;
- zmiana barwy **Gain** zmienia jasne barwy obrazu (czyli głównie odbłyски);

Z takiego opisu można by wnioskować, że obszary działania tych kontrolki są ściśle podzielone, według intensywności koloru. Sprawdźmy więc to na naszym testowym obrazie. Obecnie zawiera on samo odcienie szarości. Umożliwia to nam przetestowanie działania każdej z kontrolki poprzez samą zmianę jej intensywności (zmniejszenie z 1.0 do 0.5). Zaczniemy od **Lift** (Rysunek 10.5.5):



Rysunek 10.5.5 Wpływ zmniejszenia intensywności **Lift** (o połowę)

<sup>1</sup> Jest to algorytm o nazwie „Lift/Gamma/Gain” — domyślnie wybrany na z listy rozwijalnej u góry węzła. Ta metoda jest szeroko stosowana w różnych programach. Doczekała się nawet własnego forum dyskusyjnego — <http://liftgammagain.com>.

- Zmniejszenie intensywności **Lift** powoduje przyciemnienie i pogłębia różnicę pomiędzy najjaśniejszym a najciemniejszym fragmentem obrazu.

Oczywiście zwiększenie tej wartości spowoduje efekt odwrotny: rozjaśnienie obrazu i zmniejszenie kontrastów — aż do osiągnięcia jednolitej bieli dla wartości **Lift:V** = 2.0.

A jaki efekt spowoduje podobne zmniejszenie o połowę odcieni pośrednich? (**Gamma** — Rysunek 10.5.6):



Rysunek 10.5.6 Wpływ zmniejszenia intensywności **Gamma** (o połowę)

- Zmniejszenie intensywności **Gamma** przyciemnia cały obraz bardziej równomiernie niż **Lift**. Największej zmianie ulegają barwy o pośredniej intensywności, ale dotyczy także to w mniejszym stopniu tak cieni, jak i odbłyśków.

Zwiększenie intensywności **Gamma** powoduje efekt odwrotny — to takie specjalne rozjaśnienie obrazu, które zachowuje w dużym stopniu kontrast pomiędzy barwami.

I wreszcie zmiana wartości **Gain** z 1.0 do 0.5 (Rysunek 10.5.7):



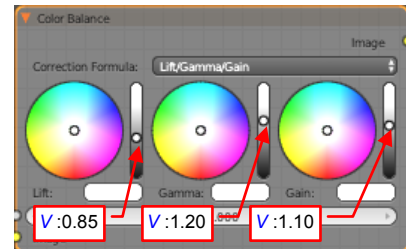
Rysunek 10.5.7 Wpływ zmniejszenia intensywności **Gain** (o połowę)

- Zmniejszenie intensywności **Gain** przyciemnia wszystkie jasne barwy, zmniejszając kontrast obrazu. (Im jaśniejsza barwa, tym większa zmiana).

Zwiększenie intensywności **Gain** powoduje rozjaśnienie obrazu połączone ze zwiększeniem kontrastów (to taki efekt „słonecznego dnia”).

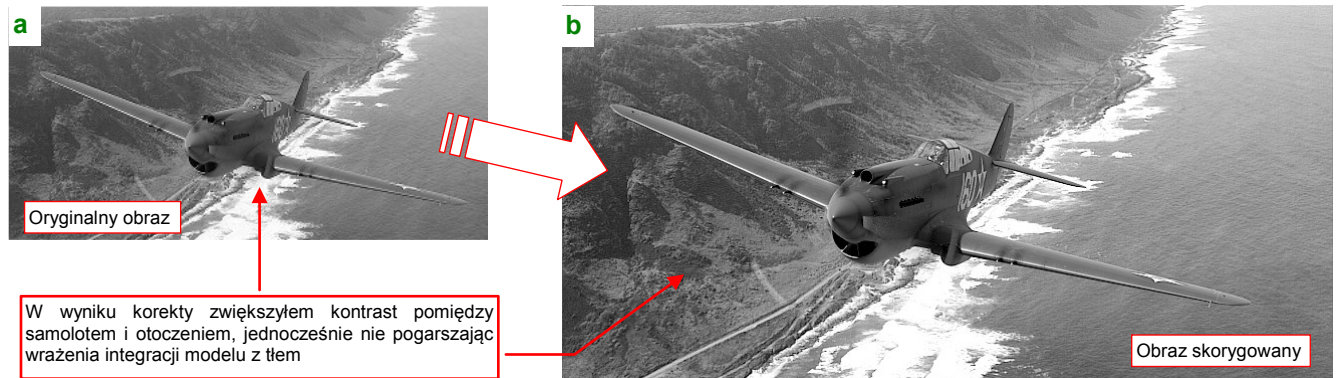


Opierając się na tych doświadczeniach, można nieco poprawić naszą „biało-czarną” fotografię. Zauważyłem, że lepiej wygląda gdy pogłębiłem nieco cienie (*Lift:V* = 0.85). Potem skompensowałem zmniejszenie jasności zwiększając intensywność tonów pośrednich (*Gamma:V* = 1.2). Na koniec zrobiłem ten dzień „bardziej słonecznym”, zwiększając intensywność bieli i odbłyśków (*Gain:V* = 1.1).



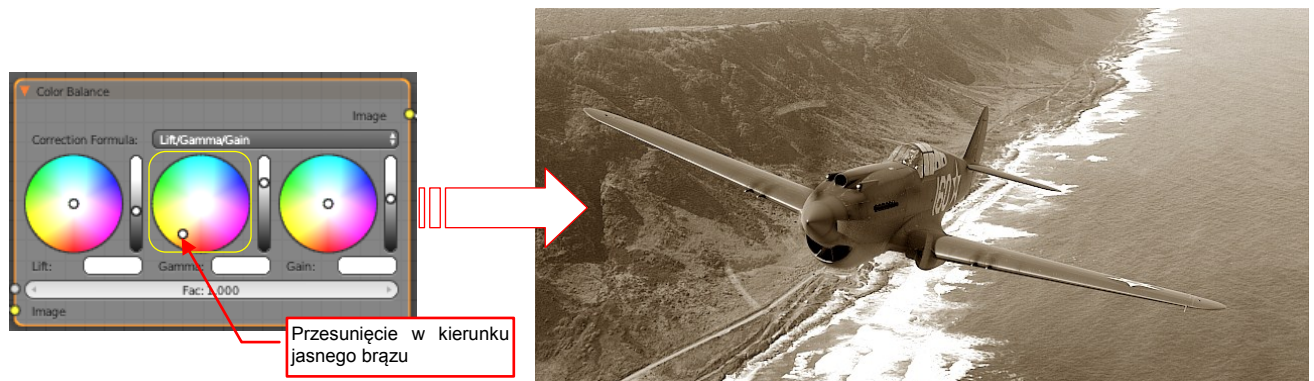
Ustawienia *Color Balance* przedstawia Rysunek 10.5.8, miniaturę oryginalnego obrazu — Rysunek 10.5.9a), a rezultat — Rysunek 10.5.9b):

Rysunek 10.5.8 Przykładowa korekta intensywności czarno-białego obrazu



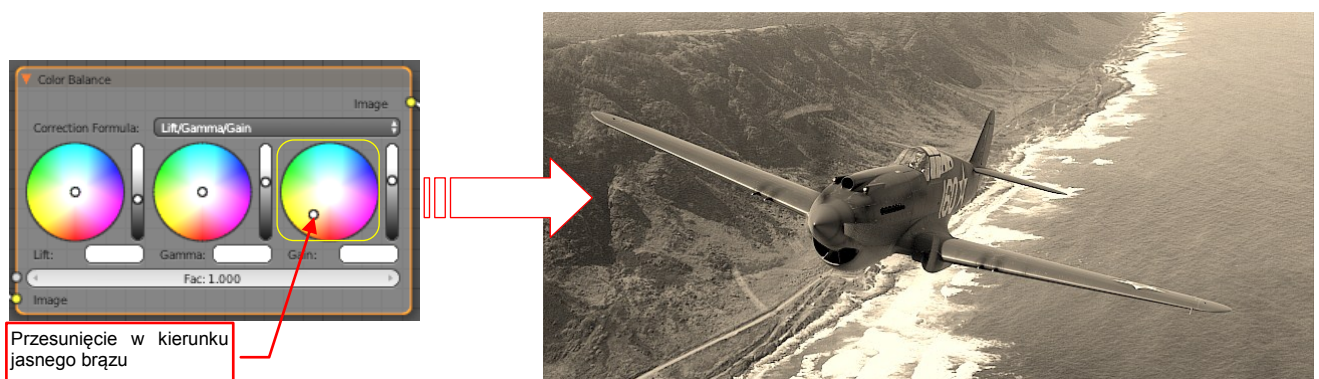
Rysunek 10.5.9 Korekta intensywności czarno-białego obrazu

Wielu twórców zaczynało swoje wizualizacje od obrazów w odcieniach szarości, bo takie łatwiej jest zintegrować z tłem. Zazwyczaj na koniec nadawali swoim pracom odcień sepii, aby sprawić wrażenie „historycznego zdjęcia”. Rysunek 10.5.10 przedstawia taki efekt uzyskany poprzez zmianę barwy *Gamma*:



Rysunek 10.5.10 Efekt „sepia”, uzyskany za pomocą zmiany odcienia *Gamma*

A Rysunek 10.5.11 przedstawia wykorzystanie w tym celu odcieni *Gain*:

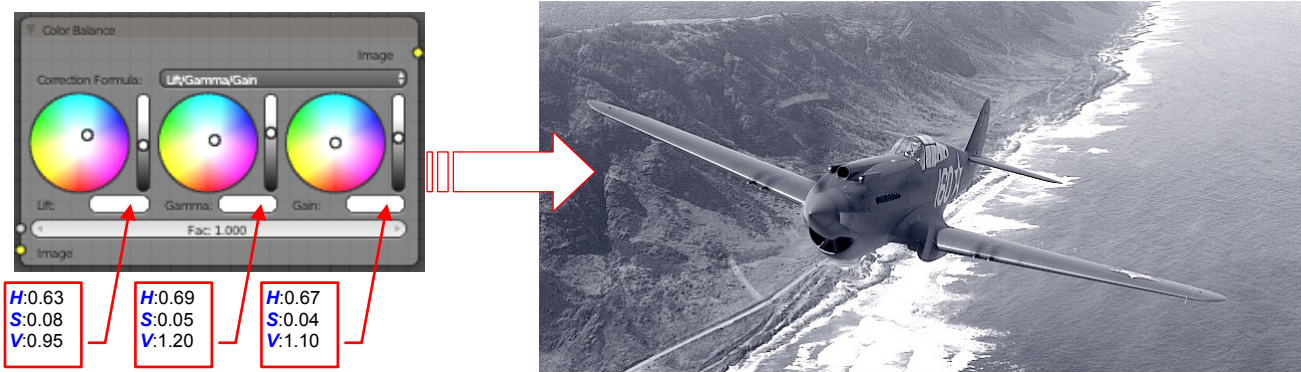


Rysunek 10.5.11 Efekt „sepia”, uzyskany za pomocą zmiany odcienia *Gain*



Porównując rezultaty przedstawione przez Rysunek 10.5.10 i Rysunek 10.5.11 widać że Blender w każdym z nich podkreślił za pomocą tej samej barwy inne fragmenty obrazu. Wyraźnie rządzą tym efektem te same reguły, które poznaliśmy badając wpływ zmiany intensywności współczynników *Lift/Gamma/Gain*.

Spróbujmy teraz nadać obrazowi lekko błękitny odcień, nie zmieniając jednak intensywności składników barw *Color Balance* (bo te dopracowaliśmy już przed chwilą). Rezultat pokazuje Rysunek 10.5.12:



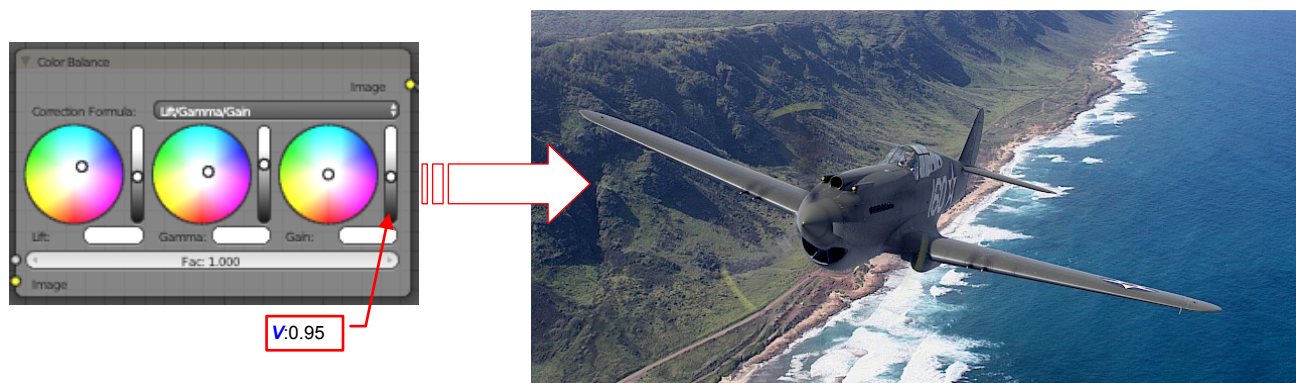
Rysunek 10.5.12 Nieznaczące zabarwienie obrazu na niebiesko

Uzyskałem to zmieniając barwę każdej z kontrolki na niebieską ( $H = 0.63..0.69$ ), różniącą się tylko nasyceniem ( $S = 0.04 .. 0.08$ ) (por. str. 204). A co się stanie jak teraz przywrócimy na tym obrazie barwy (Rysunek 10.5.13)?



Rysunek 10.5.13 Ten sam obraz po przywróceniu barw

Wydaje mi się że dodanie odrobiny błękitu poprawiło ostateczny efekt. Obraz wydaje się teraz chłodniejszy, i może przez to bardziej „poważny”. Porównując ten rezultat z oryginałem, znalazłem jednak pewien błąd: pióropusze morskiej piany stały się tak białe, że zniknęły z nich drobne szczegóły. Coś takiego w slangu grafików nazywa się „przepaleniem” bieli. Oznacza bezpowrotną stratę informacji o detalach, co zepsuje ewentualne dalsze efekty przetwarzania. No cóż, pozostaje zmniejszyć intensywność kontrolki *Gain* (Rysunek 10.5.14):



Rysunek 10.5.14 Eliminacja „przepalenia” bieli

Wydaje mi się że ogólnie obraz na tym stracił, ale można spróbować rozjaśnić sam samolot, używając maski.

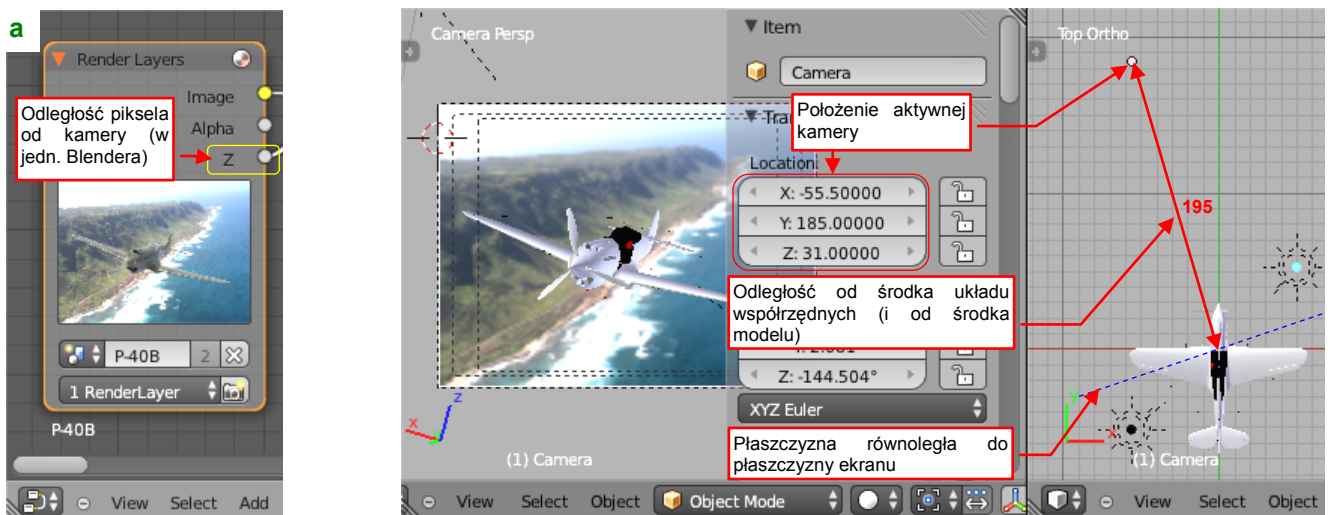


## 10.6 Tworzenie masek ze współrzędnych Z pikseli

Jak „wydobyć” z wyrenderowanego obrazu sam kontur samolotu? Najprostszym rozwiązaniem byłoby renderować go z przezroczystym tłem, a potem nałożyć na obraz (używając informacji z wyjścia *Alpha*). Taka metoda ma jednak w Blenderze pewne wady: gubi się efekt rozmycia ruchu łopát, a niemal przejrzyste szkło kabiny zupełnie znika. Dlatego renderuję model od razu z tłem, co powoduje że wartość *Alpha* całego obrazu jest = 1.0 (i przez to bezużyteczna).

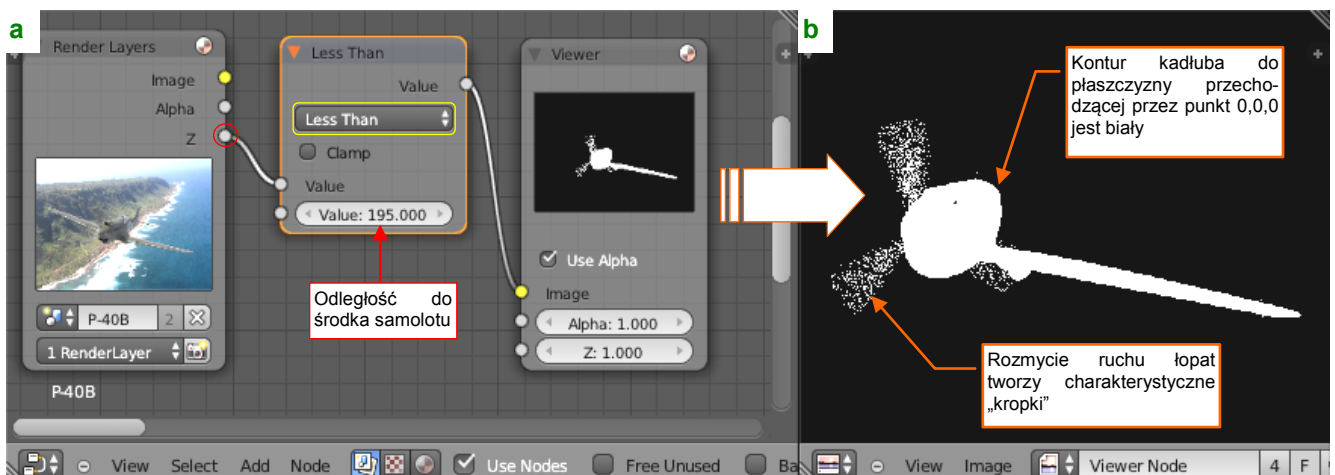
Zamiast tego możemy użyć informacji o głębokości pikseli (wyjścia *Z* z węzła *Render Layers* — Rysunek 10.6.1a). Renderer używa go wewnętrznie aby określić który piksel jest bliżej kamery, a który dalej. Aby był dostępny w węzłach kompozytora, włącz w zestawie *Render Layers*, paneli *Passes*, opcję *Z* (domyślnie zresztą jest włączona). Jest to liczba zmiennoprzecinkowa, podająca odległość (wyrażoną w jednostkach Blendera) od powierzchni kamery do renderowanego piksela.

W scenie, którą wykorzystam jako przykład w tej sekcji, współrzędne położenia kamery to -55,185,31 (Rysunek 10.6.1b). Oznacza to, że odległość od środka modelu (ściany ogniowej kadłuba, leżącej na płaszczyźnie *ZX* = 0) wynosi 195 jednostek Blendera.



Rysunek 10.6.1 Informacja o odległości od kamery (Z)

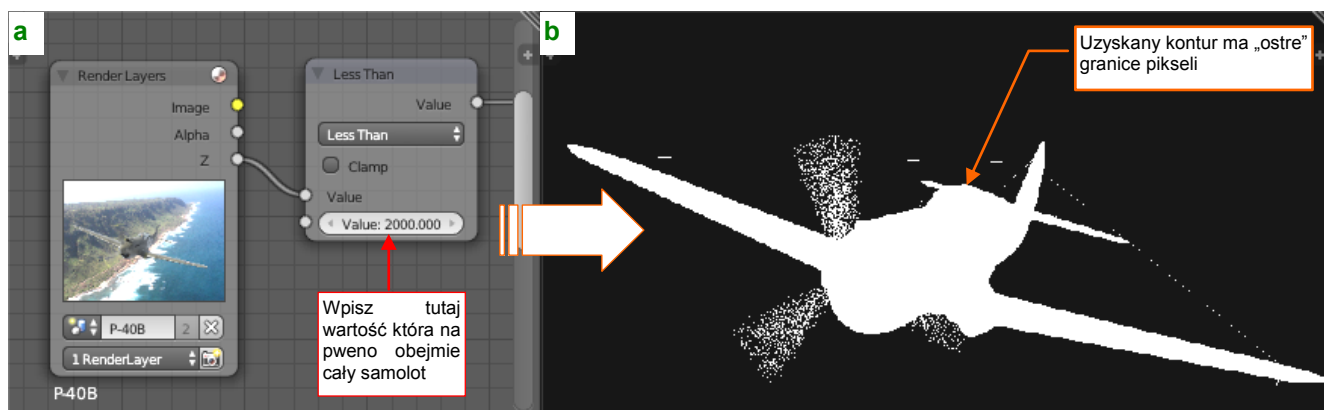
Sprawdźmy w kompozytorze, czy się nie pomylił. Dodaję do schematu węzeł *Converter* → *Math*. Wybrałem w nim funkcję *Less Than*, która dla wartości < 195 zwraca biel, a dla pozostałych — czerni (Rysunek 10.6.2a). Rezultat jest zgodny z oczekiwaniami (Rysunek 10.6.2b):



Rysunek 10.6.2 Próba wykorzystania informacji o głębokości (Z)

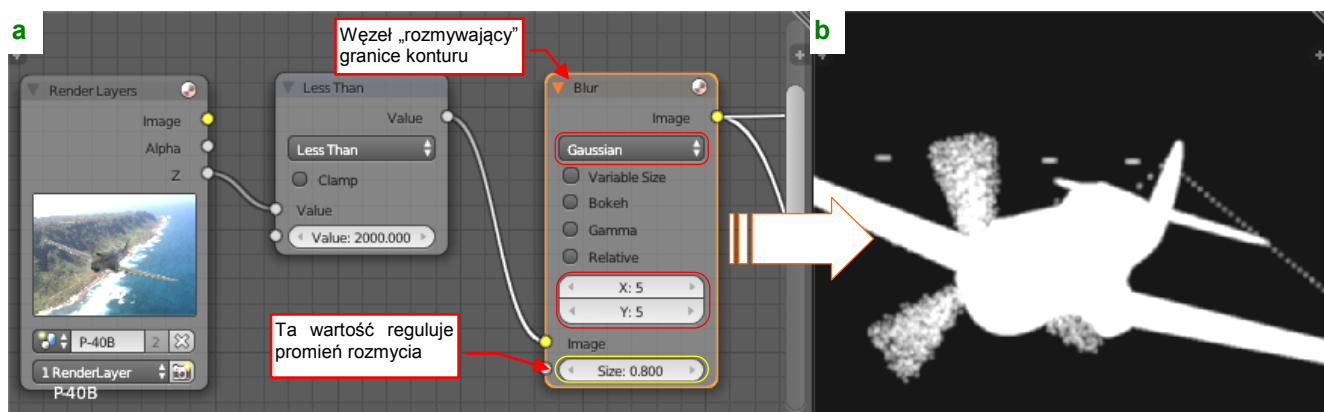
Widzimy biały kontur przodu samolotu — aż do płaszczyzny odległej od kamery o 195 jedn. Blendera.

Wygląda na to, że tło ma jakąś nieskończenie dużą wartość **Z**. W węźle **Less Than** możesz więc spokojnie wpisać odpowiednio dużą liczbę, która obejmie nie jeden, a całą scenę z kilkoma maszynami (Rysunek 10.6.3a):



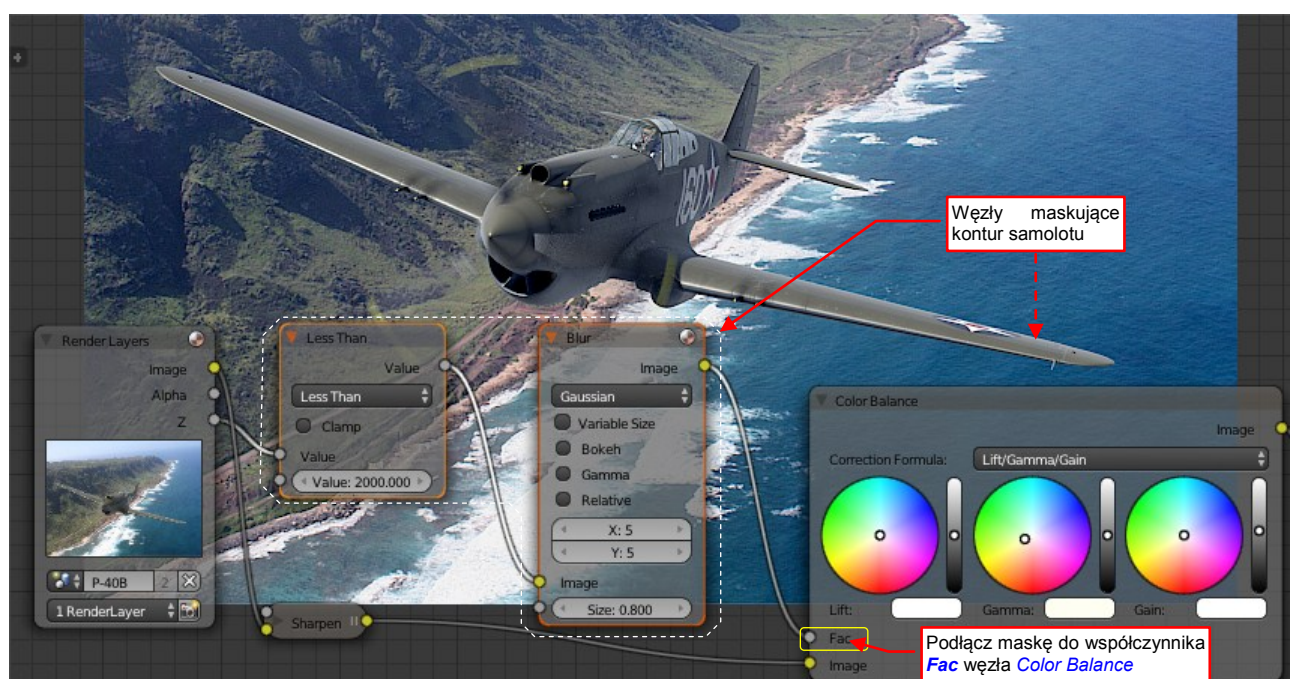
Rysunek 10.6.3 Pierwsza wersja maski

Zwróć uwagę, że tak uzyskana maska ma wyraźne granice pikseli (Rysunek 10.6.3b). Dodałem więc węzeł **Filter→Blur**, aby rozmyć jej kontury (Rysunek 10.6.4). Poprawi to także rozmyte łopaty śmigła!



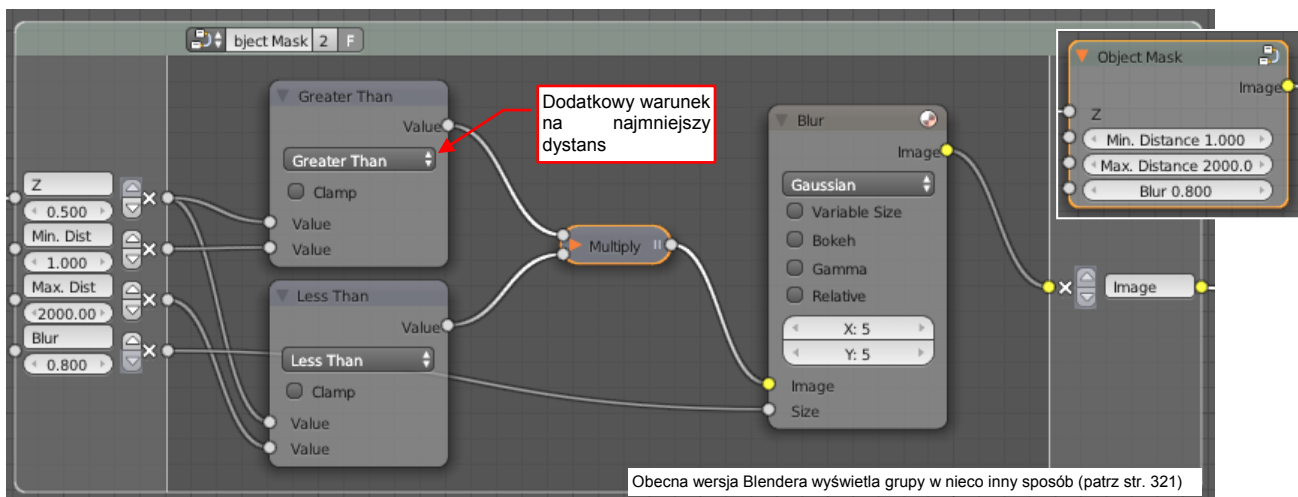
Rysunek 10.6.4 Druga wersja maski — z rozmyciem

Taką maskę można użyć w kompozycji — np. do dopasowania barw samolotu do otoczenia (Rysunek 10.6.5):



Rysunek 10.6.5 Zastosowanie maski do zmiany odcienia modelu

Sądzę, że taka maska może się często przydawać, więc przekształciłem te węzły w grupę. Nadałem jej nazwę **Object Mask** (Rysunek 10.6.6):



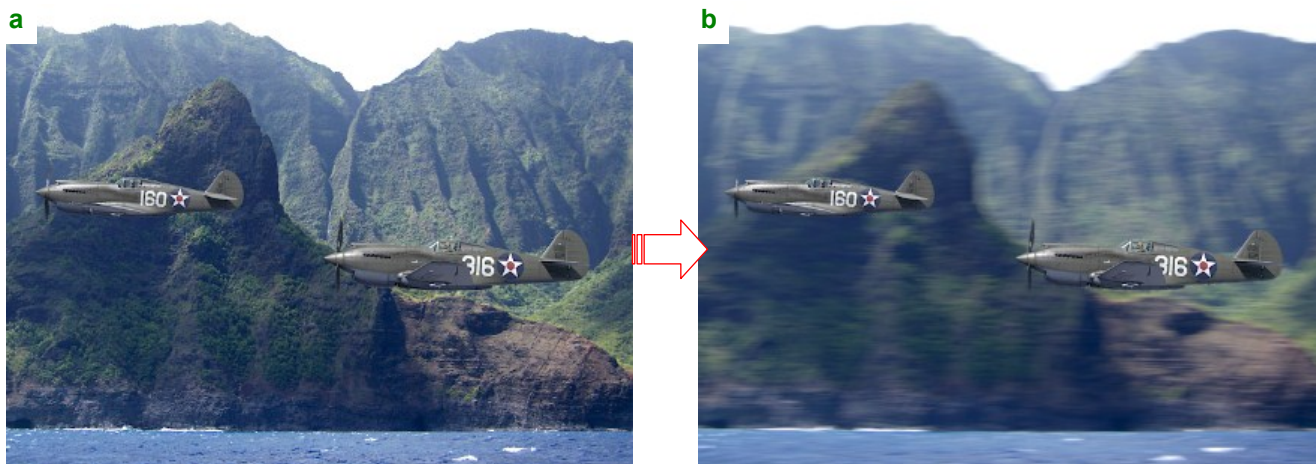
Rysunek 10.6.6 Grupa **Object Mask**

Będę ją importował z pliku do pliku — zawsze to łatwiejsze niż tworzenie tego efektu za każdym razem od nowa. Mogę dodać ją także do szablonu nowego pliku Blendera. Do schematu grupy na wszelki wypadek dodałem, oprócz testu na największą odległość, także test na najmniejszą odległość. Może się kiedyś przydać do precyzyjnego wyboru obiektu (np. tylko drugiego samolotu lecącego w szyku).



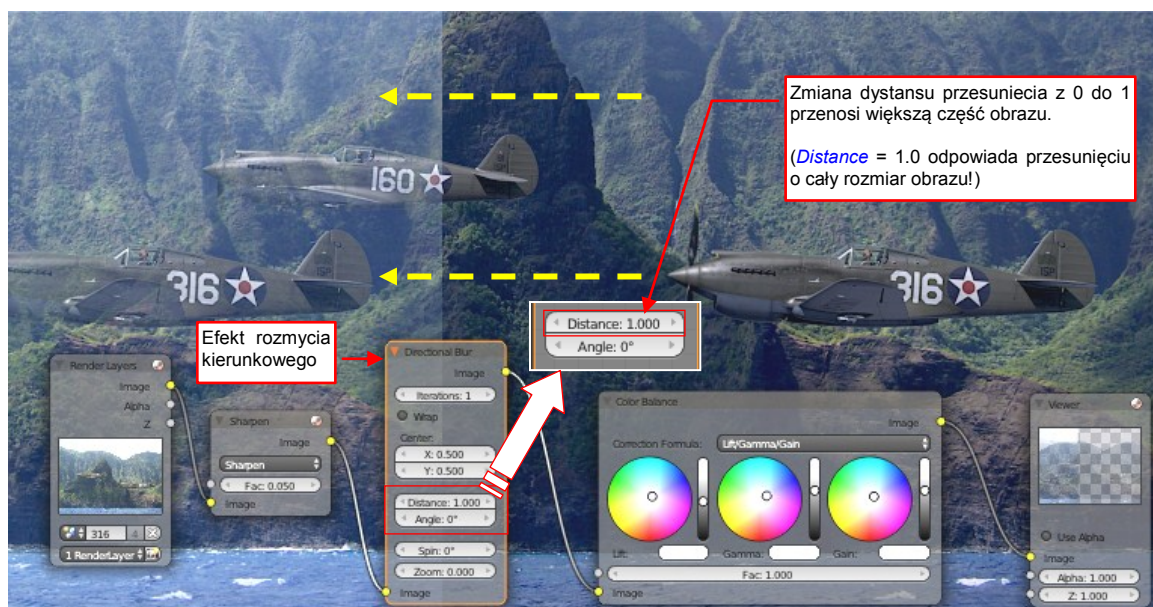
## 10.7 Rozmycie w jednym kierunku (*Directional Blur*)

Efekt rozmycia obrazu w jednym kierunku pozwala stworzyć bardzo silne wrażenie prędkości, dynamiki (Rysunek 10.7.1):



Rysunek 10.7.1 Wykorzystanie poziomego rozmycia tła

Do uzyskania rozmycia kierunkowego w kompozytorze służy węzeł *Filter* → *Directional Blur* (Rysunek 10.7.2):



Rysunek 10.7.2 Schemat kompozycji po dodaniu węzła *Directional Blur*

Włączyłem węzeł *Directional Blur* do schematu kompozycji. Umieściłem go przed węzłem *Color Balance*, bo te dwa efekty zupełnie sobie nie przeszkadzają. Domyślnie *Directional Blur* ma wszystkie parametry ustawione na wartościach neutralnych, i niczego nie zmienia na obrazie wynikowym. Kontrolki węzła pozwalają uzyskać trzy kształty rozmycia:

- Liniowe (tzn. cały obraz przesunięty w jednym kierunku: kontrolki *Distance*, *Angle*);
- Koncentryczne (tzn. „w gwiazdę” — kontrolki *Zoom* i *Center*);
- Spiralne (kontrolki *Spin* i *Center*);

Można stosować także złożenia tych efektów. Zaczniemy rozmycia liniowego. Spróbowałem wywołać rozmycie w kierunku poziomym (*Directional Blur:Angle* = 0) zwiększając *Directional Blur:Distance* z 0.0 o 1.0. Spowodowało to poważne zmiany w obrazie (Rysunek 10.7.2 pokazuje je w podglądzie pod węzłami). Jednak to w ogóle nie wygląda jak jakiegokolwiek rozmycie! Dlaczego?



Po pierwsze, wygląda na to że wartość *Distance* jest wyrażona w proporcji do rozmiaru obrazu. Spróbuj ją zmniejszyć do 0.5, a rezultat zacznie się „zagęszczać” (Rysunek 10.7.3):



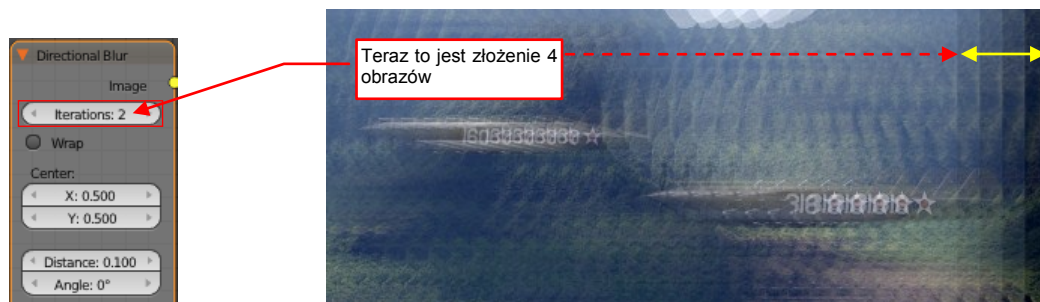
Rysunek 10.7.3 Wpływ zmniejszenia wartości *Distance* (do 0.5)

Gdy zmniejszysz *Distance* do wartości rzędu 10% rozmiaru obrazu lub mniej, poszczególne elementy zaczną się na siebie nakładać (Rysunek 10.7.4):



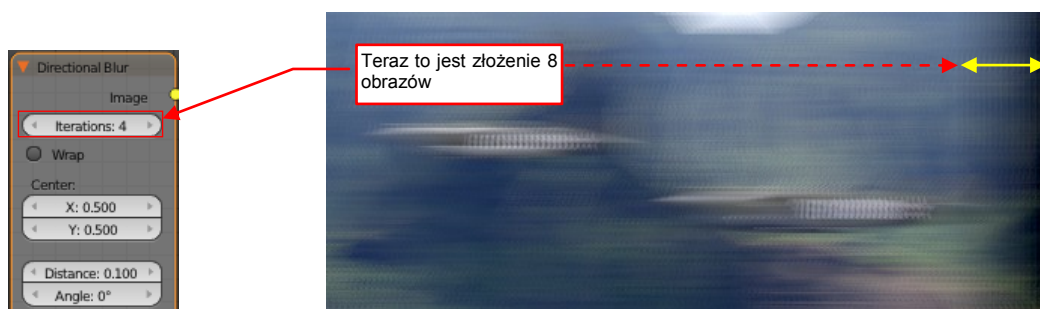
Rysunek 10.7.4 Wpływ zmniejszenia wartości *Distance* (do 0.1)

Drugim bardzo ważnym parametrem jest liczba iteracji (*Iterations*). Aby się o tym przekonać, zwiększ teraz jej wartość z 1 (brak iteracji) do 2 (Rysunek 10.7.5):



Rysunek 10.7.5 Wpływ zwiększenia wartości *Iterations* (do 2)

Gdy zwiększysz liczbę iteracji jeszcze bardziej, zobaczysz wreszcie na obrazie efekt przypominający rozmycie ruchu (Rysunek 10.7.6):

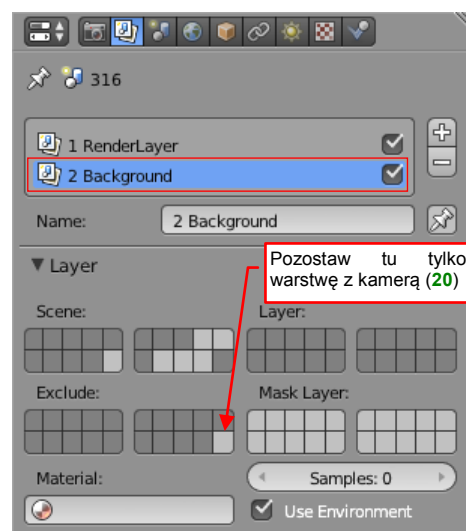


Rysunek 10.7.6 Wpływ zwiększenia wartości *Iterations* (do 4)

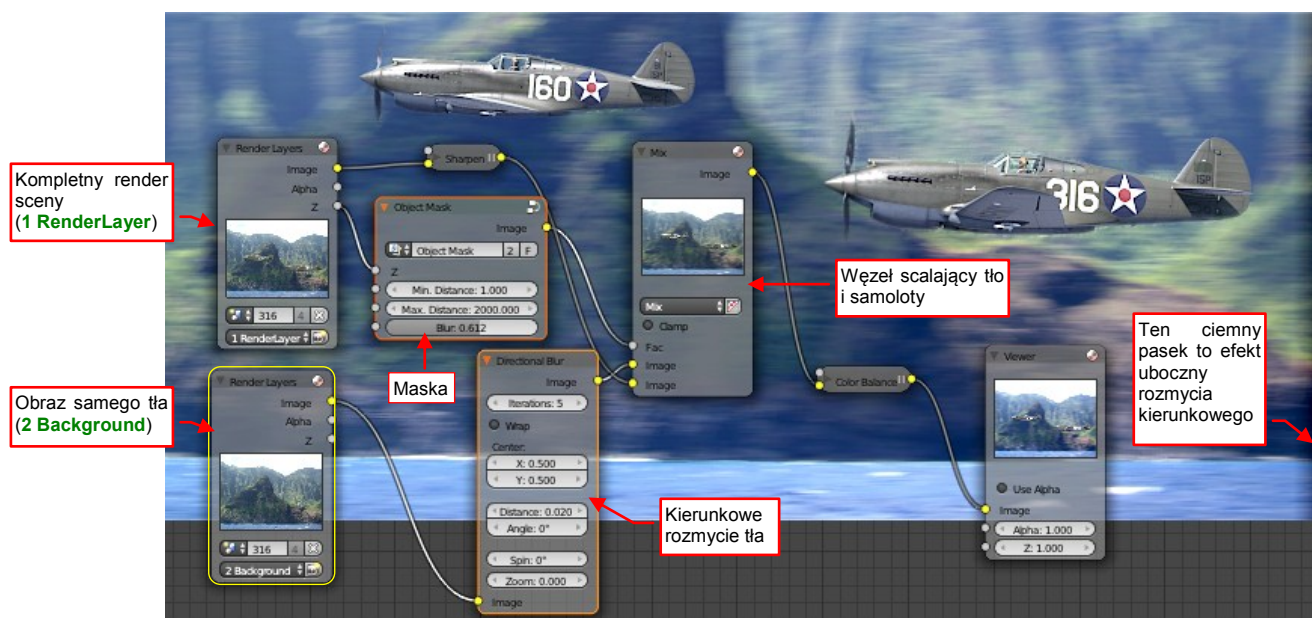
Co prawda w ten sposób rozmyciu uległo wszystko — i tło, i samoloty. Jak wykluczyć z tego pierwszy plan?

Aby rozmyć samo tło, musimy nałożyć na siebie dwa obrazy. Pierwszy z nich zawiera rozmytą fotografię bez samolotów, a drugi — same samoloty. Trzeba to jednak zrobić tak, aby barwy otoczenia widocznego za przejrzystym oszkleniem kabiny pasowały do obrazu tła. Aby uzyskać obraz samego otoczenia bez samolotów, przygotowałem drugą warstwę renderowania, o nazwie **2 Background** (Ustawioną tak, jak pokazuje Rysunek 10.7.7).

Dodałem ją do schematu jako dodatkowy węzeł **Render Layer** (Rysunek 10.7.8). Obraz z tego węzła poddałem umiarkowanemu rozmyciu w poziomie (*Distance* = 0.02, *Iterations* = 5). Następnie użyłem maski (**Object Mask** — por. str. 443) aby rezultat rozmycia „podłożyć” jako tło pod ostre sylwetki samolotów z warstwy **1 RenderLayer**. (Maska sylwetek samolotów jest użyta w charakterze współczynnika w węźle **Color→Mix** — por. Rysunek 10.7.8):

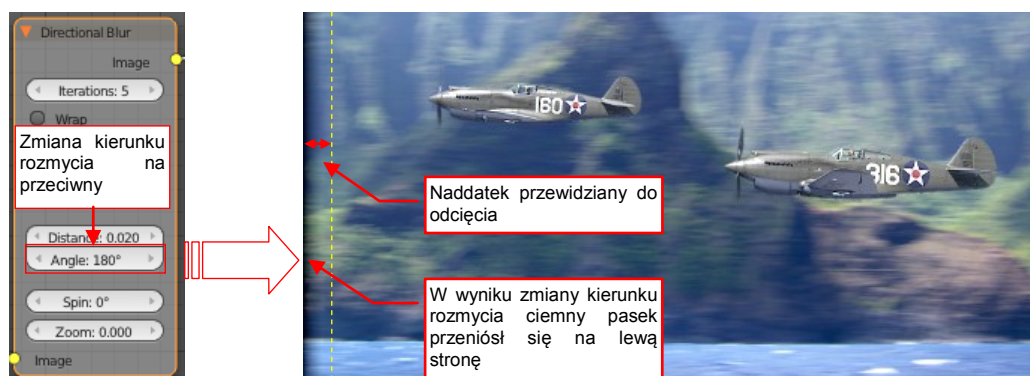


Rysunek 10.7.7 Druga warstwa renderowania



Rysunek 10.7.8 Użycie dwóch warstw renderowania do wykluczenia z rozmycia sylwetek samolotów

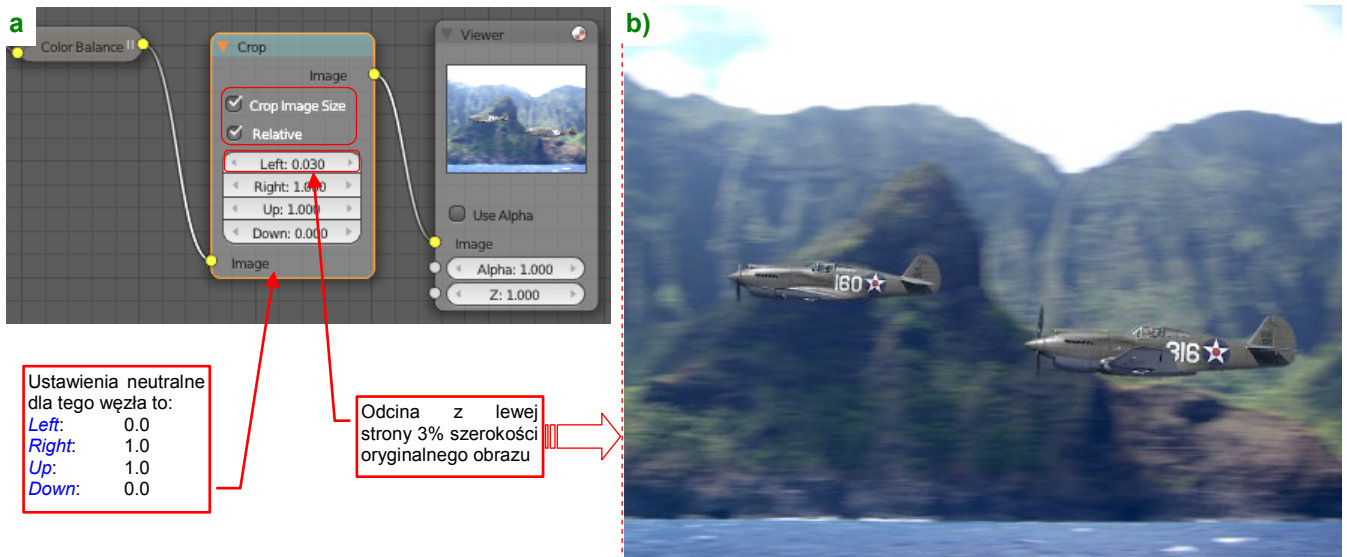
Na uzyskanym w ten sposób obrazie samoloty faktycznie wydają się mknąć ponad powierzchnią morza. Razi tylko nieco rozmyty ciemny pasek z prawej strony. To efekt uboczny działania **Directional Blur**. Podczas komponowania położenia modeli na scenie warto zawczasu przewidzieć pewien naddatek na ten artefakt. W tej scenie pozostawiłem trochę więcej przestrzeni z lewej, więc przenieś go tam zmieniając kierunek rozmycia na przeciwny. Ustawiłem wartość kąta *Angle* z 0° na 180° (Rysunek 10.7.9):



Rysunek 10.7.9 Przeniesienie „czarnego paska” na przeciwną krawędź obrazu



Na koniec pozostaje odciąć naddatek z lewej strony obrazu za pomocą węzła **Distort** → **Crop** (Rysunek 10.7.10):



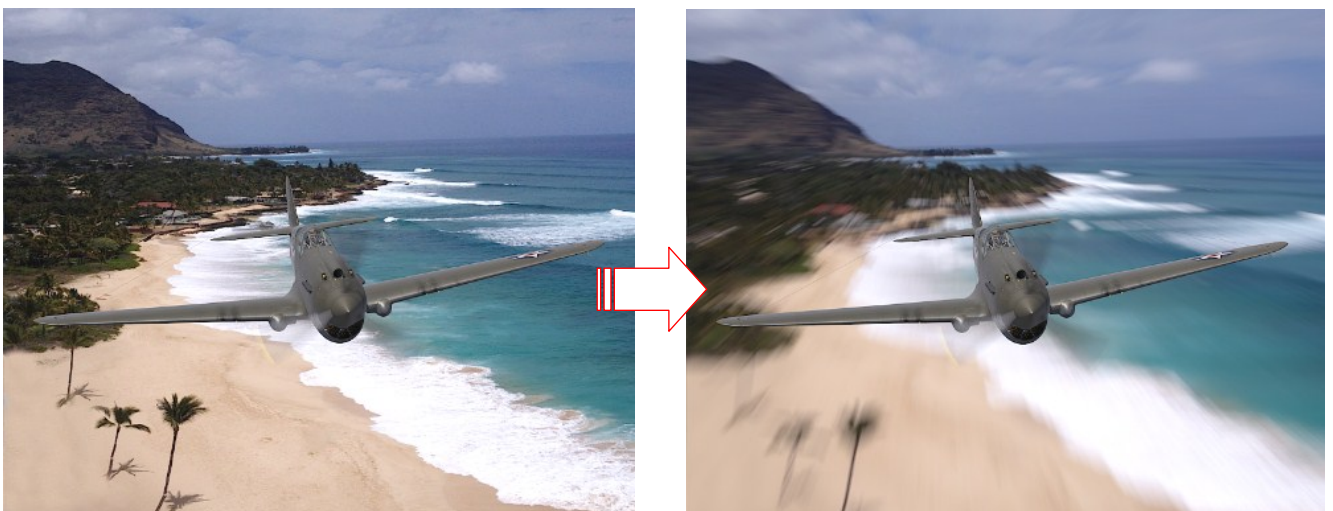
**Rysunek 10.7.10 „Przycięcie” obrazu**

Uważaj na węzeł **Crop**: w odróżnieniu od większości pozostałych nie jest wstawiany do schematu w stanie „neutralnym” (tzn. takim, który nic nie zmienia). Rysunek 10.7.10a podaje w objaśnieniu, jakie należy wpisać wartości parametrów **Left**, **Right**, **Up**, **Down** aby obraz pozostał nie naruszony. Te wartości są wyrażone jako proporcje odpowiedniego boku (1.0 = 100%). Stąd, aby odciąć 3% szerokości z lewej, wpisałem **Crop:Left** = 0.03.

Do stworzenia samego tła potrzeba tylko kilku próbek renderera. Gdy uzyskujemy ten obraz z oddzielnej warstwy renderowania (**2 Background**), Blender renderuje ten fragment taką samą liczbą próbek jak dla kompletnego modelu. (Liczba próbek jest wspólna dla wszystkich warstw renderowania na scenie).

- Jeżeli chcesz uniknąć niepotrzebnego renderowania obrazu tła, stwórz nową scenę (wywołując tę operację z opcją **Copy Settings** — por. str. 203). Z listy w zestawie **World** wybierz to samo środowisko (**World**) co w scenie podstawowej. Na koniec dodaj (**Object** → **Make Links** → **Object to Scene**) do nowej sceny link do obiektu kamery i ustaw w panelu **Sampling** zestawu **Render** odpowiednio małą liczbę próbek (np. 10).

Do dynamicznych ujęć z przodu lepiej pasuje<sup>1</sup> rozmycie koncentryczne (Rysunek 10.7.11):



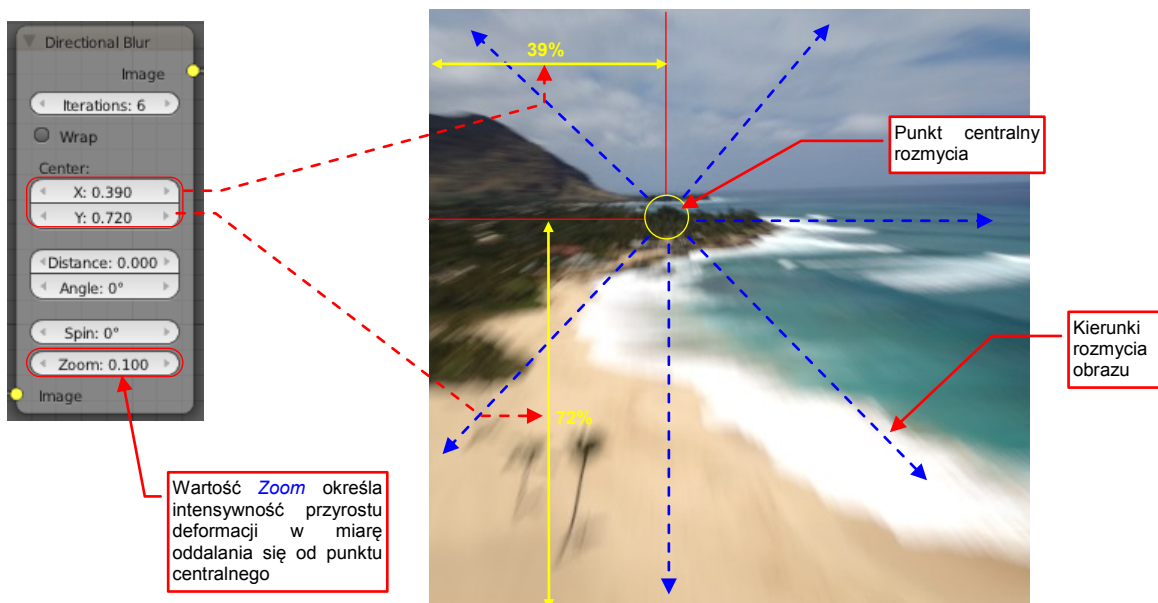
**Rysunek 10.7.11 Koncentryczne rozmycia obrazu tła**

<sup>1</sup> Zwróć uwagę że efekt rozmycia w jakiś sposób „uwiarygadnia” model. Przyjrzyj się chociażby obrazowi przedstawionemu przez Rysunek 10.7.11. Samolot po lewej wygląda trochę sztucznie na tle zdjęcia bez rozmycia. Na rozmytym tle prezentuje się o wiele lepiej.

Rozmyciem koncentrycznym obrazu sterują w węźle *Directional Blur* dwa parametry:

- Współrzędne *Center* określają środek deformacji (w tym miejscu nic nie jest rozmyte);
- Wartość *Zoom* steruje intensywnością deformacji;

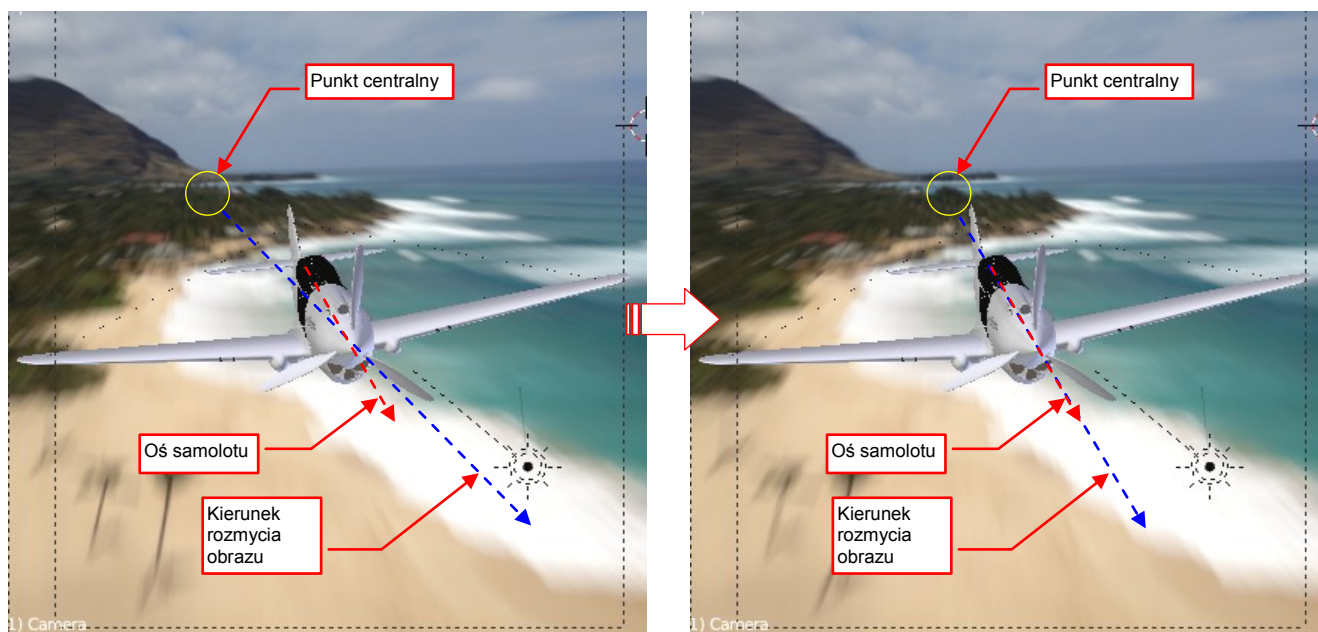
Rysunek 10.7.12 przedstawia geometrię tej deformacji obrazu:



Rysunek 10.7.12 Parametry sterujące koncentrycznym rozmyciem obrazu

Współrzędne *X*, *Y* punktu centralnego są wyrażone w sposób względny, jako ułamki odpowiednio: szerokości i wysokości renderowanego obszaru. Stąd wartość 0,0 oznacza lewy dolny narożnik, a 1,1 — prawy górny. Wydaje mi się że skala wartości *Zoom* jest zbliżona do skali parametru *Distance* — *Zoom* = 0.1 oznacza już znaczącą deformację na zewnętrznych granicach obrazu.

Przy stosowaniu koncentrycznego rozmycia tła należy ustawić osie użytych w scenie samolotu/samolotów wzdłuż promieni wychodzących z punktu centralnego (Rysunek 10.7.13):

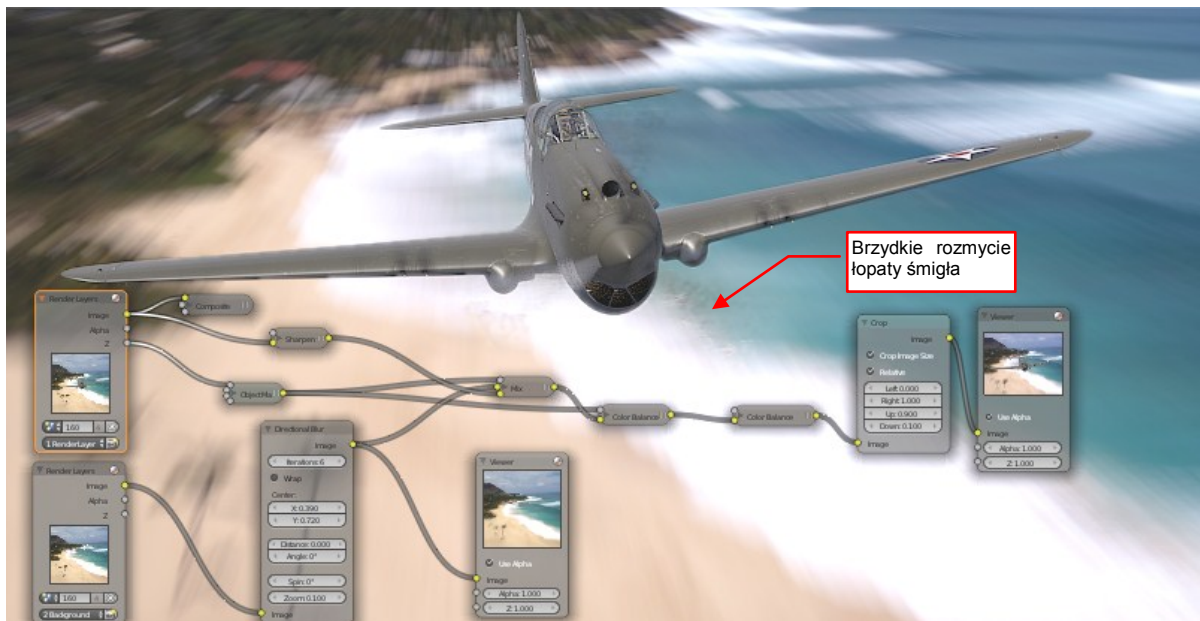


Rysunek 10.7.13 Uzgadnianie kierunku rozmycia tła i osi modelu

Jeżeli tego nie zrobisz, samolot na obrazku będzie wyglądał jak gdyby lekko trawersował. Uzgodnienie osi możesz uzyskać dopasowując położenie modelu w kamerze do tła, albo zmieniając współrzędne *Center* w węźle *Directional Blur*.

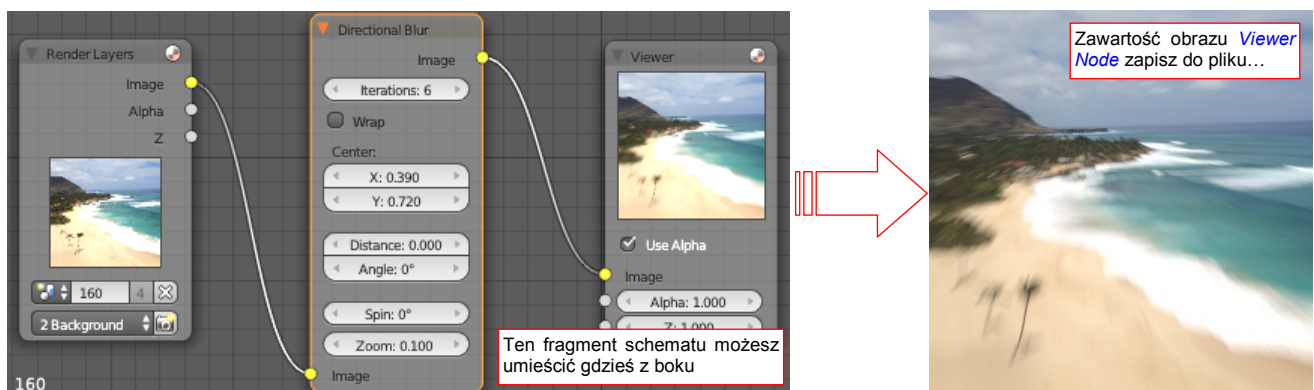


Rysunek 10.7.14 przedstawia podstawienie takiego modelu i tła do wypracowanego w tej sekcji schematu:



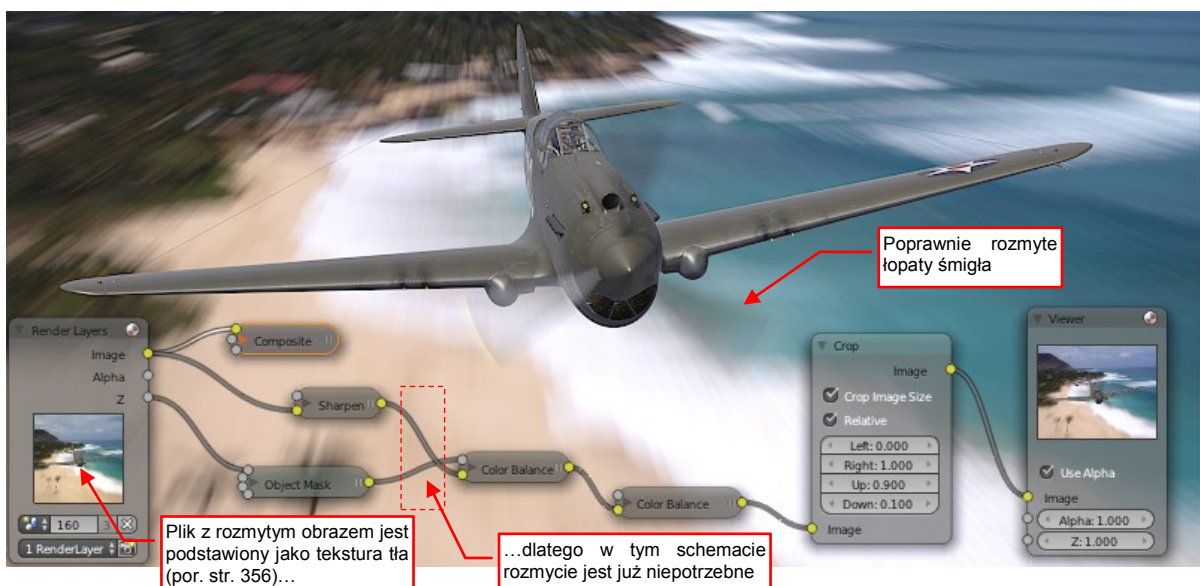
Rysunek 10.7.14 Rezultat (wersja z użyciem maski)

Aby uniknąć brzydkiego rozmycia łopat śmigła można najpierw wygenerować sam plik tła (Rysunek 10.7.15):



Rysunek 10.7.15 Zapisanie rozmytego obrazu tła do pliku

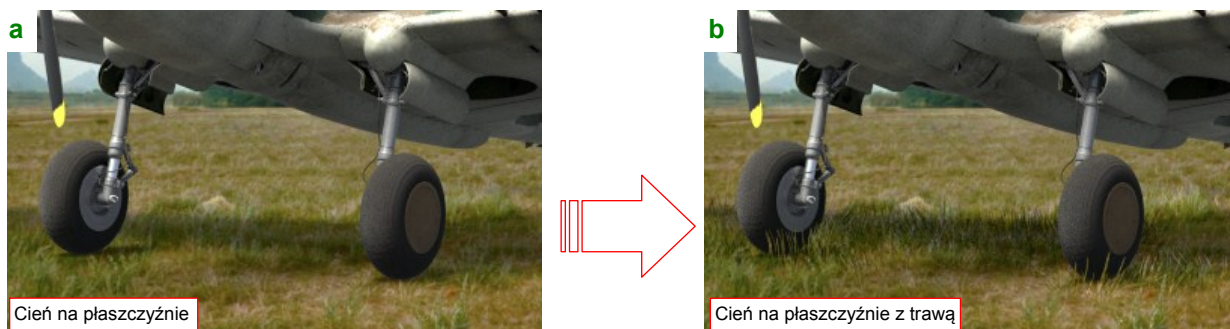
Potem wystarczy podstawić ten rozmyty obraz jako nową teksturę tła (Rysunek 10.7.16):



Rysunek 10.7.16 Zwykła kompozycja modelu z wcześniej rozmytym tłem

## 10.8 Trawa (*Particles*)

Do modelowania różnych „masowych i stochastycznych” obiektów, jak włosy czy trawa (a także innych, jak dym czy płomień) stosuje się w Blenderze tzw. „cząsteczki” (*Particles*). Można ich użyć do stworzenia całych pól i trawników. Pokażę tu jedynie dość specyficzne zastosowanie takiego systemu cząstek. W tej książce korzystamy z modelu z podstawionym, w charakterze tła, zdjęciem (por. str. 147, Rysunek 3.2.14). Obraz trawy już na nim jest. Ten sam obraz jest nałożony (w trybie mapowania *Window*) na płaszczyznę podłoża. Jedyne, co należy zrobić, to „uwiarygodnić” połączenie modelu i tła. Służy do tego m.in. umieszczenie na podłożu stosunkowo rzadkich źdźbeł trawy. Ich podstawowym zadaniem jest „złamanie” gładkich krawędzi i wnętrza cienia, rzucanego przez samolot (Rysunek 10.8.1):

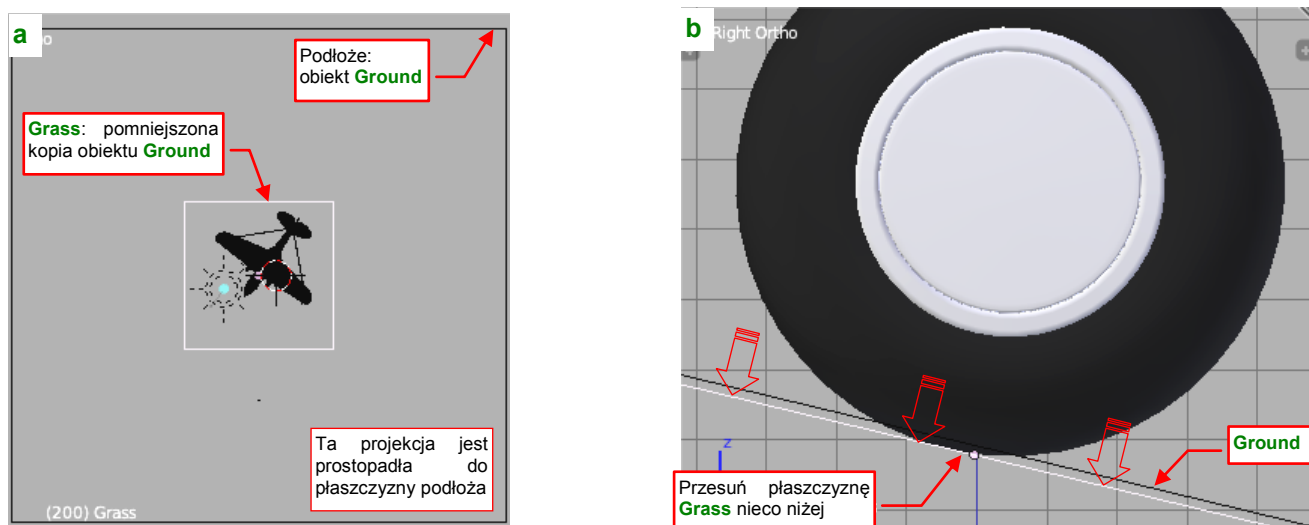


Rysunek 10.8.1 Efekt dodania źdźbeł trawy wokół i wewnątrz cienia samolotu

Rysunek 10.8.1a) przedstawia cień samolotu, rzucany na powierzchnię imitującą grunt. Intensywnością cienia steruje się, zmieniając przejrzystość powierzchni — por. str. 147. Krawędź cienia jest tu idealnie gładka, a przecież biegnie po ziemi, porośniętej drobną trawą. Także wnętrze cienia jest tu jakieś takie ...płaskie.

Rysunek 10.8.1b) przedstawia tę samą powierzchnię, ale z odpowiednio przygotowanym „efektem trawy”. Zwróć uwagę, jak pojedyncze, oświetlone słońcem źdźbła zaburzają tu przednią krawędź cienia. Tylna krawędź także jest poprzecinana zacienionymi źdźbłami, a we wnętrzu „coś jest”, dzięki czemu ten obszar nie wygląda zupełnie płasko. W tej sekcji pokażę, jak osiągnąć taki właśnie efekt.

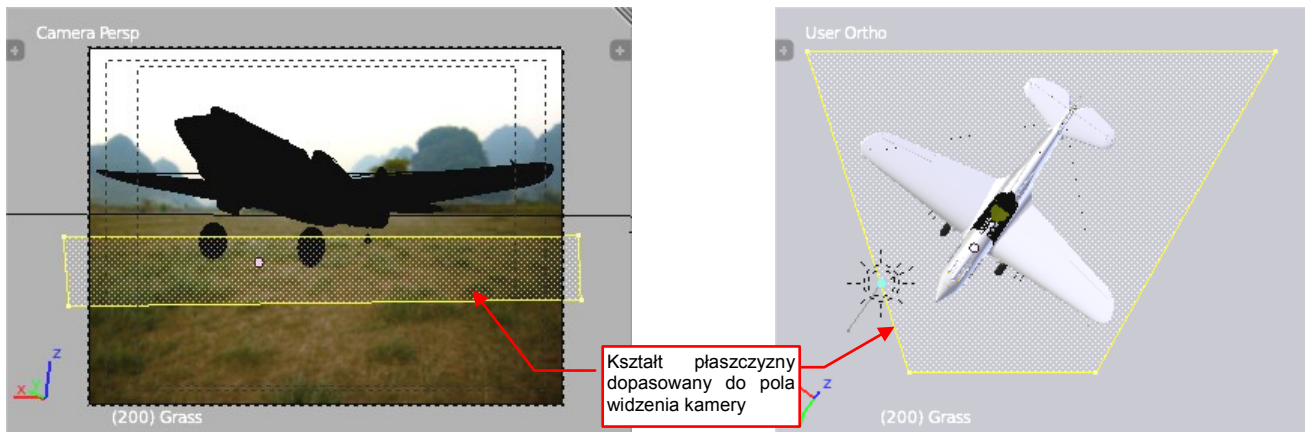
Trawa w Blenderze wyrasta („jest emitowana”) z siatki, więc zaczynamy od przygotowania powierzchni, która ma być pokryta trawą. Stworzyłem mniejszą kopię podłoża (*Ground*) i nazwałem ją *Grass* (Rysunek 10.8.2a):



Rysunek 10.8.2 Przygotowanie siatki (*emiter*)

Ten obiekt przesunąłem na niewielką odległość pod podłoże (Rysunek 10.8.2b). Zmieniłem także nazwę siatki, przypisaną do *Grass* (na taki sam jak obiektu).

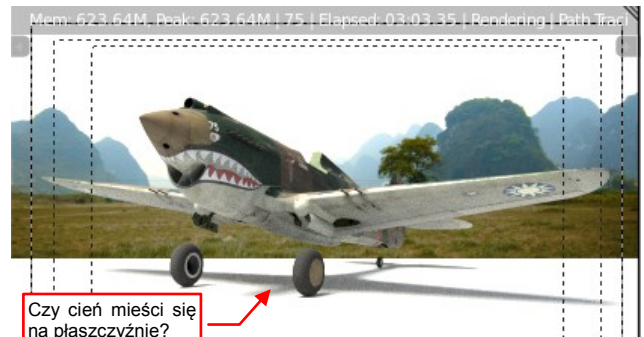
Następnie zmieniłem kształt tej pomocniczej płaszczyzny, aby nie wystawała zbyt za obszar objęty przez kamerę (Rysunek 10.8.3):



Rysunek 10.8.3 Siatka w kamerze — przybliżenie do cienia samolotu

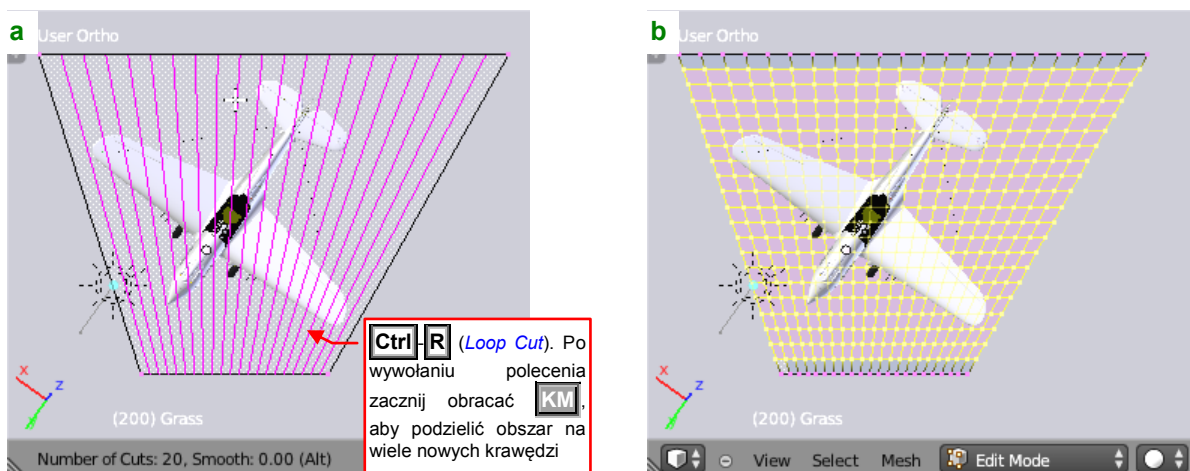
Tę modyfikację wprowadzałem posługując się dwoma oknami **3D View**. Przesuwałem wierzchołki w widoku prostokątnym do płaszczyzny, jednocześnie patrząc gdzie się znajdują w widoku z kamery.

Potem sprawdź jeszcze w podglądzie renderu, czy obszar „trawy” obejmuje cały cień samolotu (Rysunek 10.8.4). Nie zawsze trzeba pokrywać wszystko — pewne fragmenty cienia mogą pozostawać poza siatką. To zależy od rozłożenia roślinności na zdjęciu tła.



Rysunek 10.8.4 Przybliżenie do cienia samolotu

Na następnych stronach przekonasz się, że każde źdźbło trawy będzie „wyrastać” z pojedynczej ściany siatki. Dlatego musimy ją teraz zagęścić (Rysunek 10.8.5):

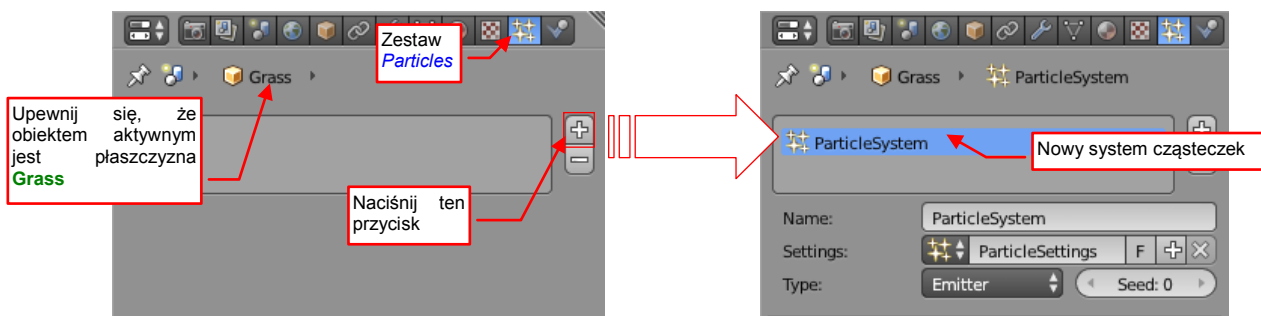


Rysunek 10.8.5 Zagęszczenie siatki podłoga trawy

Najpierw „przecinam” powierzchnię krawędziami w jednym kierunku. Robię to za pomocą polecenia przybornika **Loop Cut and Slide** (skrót: **Ctrl+R**). Gdy po jego wywołaniu zaczniesz obracać **KM**, Blender zaproponuje stworzenie nie jednej, a wielu krawędzi. Ustawiłem tę operację na wykonanie 20 nowych sekcji (Rysunek 10.8.5a). W podobny sposób dodałem krawędzie w kierunku poprzecznym (Rysunek 10.8.5b). Dodatkowo można zagęścić tę siatkę w sposób dynamiczny — modyfikatorem **Subdivision Surface**.

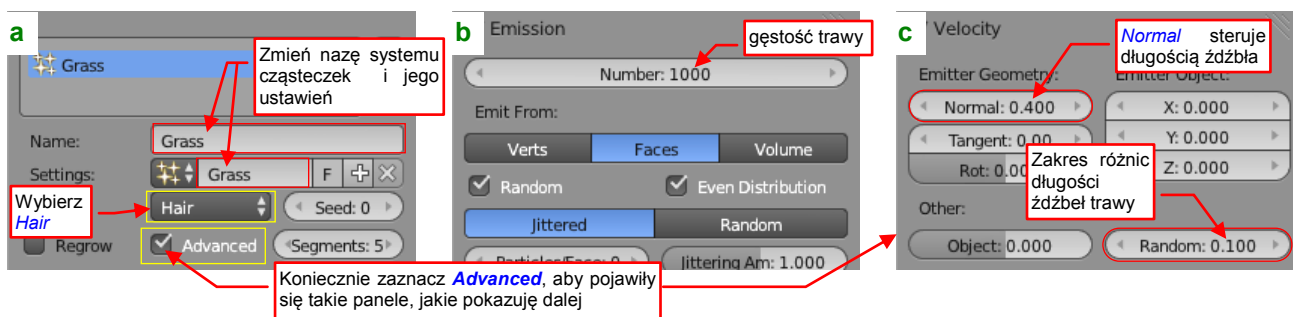


Gdy kształt „trawnika” masz już przygotowany, przejdź do zestawu **Particles** i stwórz tam nowy system cząstek (Rysunek 10.8.6):



Rysunek 10.8.6 Stworzenie nowego systemu cząstek (**Particle System**)

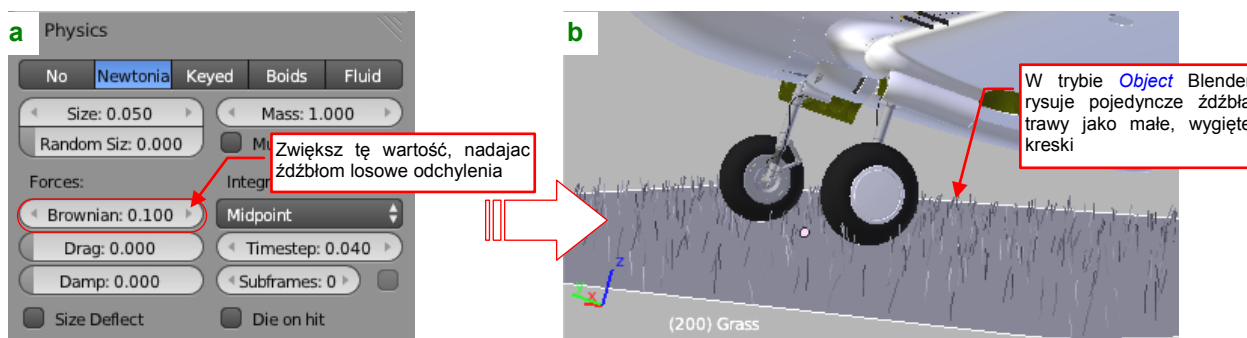
U góry zestawu znajduje się lista zdefiniowanych systemów cząstek. Na razie jest pusta. Naciśnij przycisk , umieszczony z prawej strony. Stworzysz w ten sposób nowy system cząstek (o domyślnej nazwie **ParticleSystem**). Zmień od razu nazwę na inną (np. **Grass**) i typ systemu na **Hair** (Rysunek 10.8.7a):



Rysunek 10.8.7 System cząstek **Grass** — ustawienia

Koniecznie zaznacz także opcję **Advanced**, by wstawić panele **Velocity** i **Physics**. Pole **Number** z panelu **Emission** (Rysunek 10.8.7b) steruje gęstością trawy. Po włączeniu opcji **Advanced**, długością i orientacją źdźbeł trawy steruje panel **Velocity** (każde źdźbło to jakby „ślad” poruszającej się cząsteczki). W sekcji **Emitter Geometry** parametr **Normal** odpowiada za długość trawy, a w sekcji **Other** parametr **Random** — za losowe odchylenia tej długości. Długość trawy ustawiam tak, by pasowała do zdjęcia.

Aby nadać źdźbłom losowe pochylenia, w panelu **Physics** zwiększ wartość parametru **Brownian** — np. do 0.1 (Rysunek 10.8.8a). (Chodzi tu o chaotyczne odchylenia — w fizyce nazywane „ruchami Browna”):



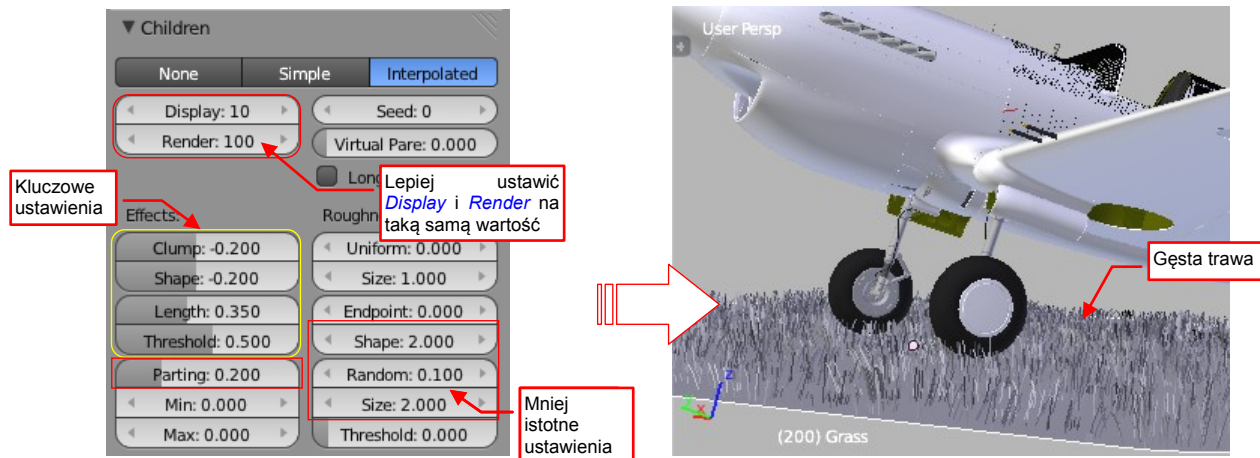
Rysunek 10.8.8 Efekt na siatce

W trakcie tych ustawień zapewne zauważyłeś zmiany, widoczne w widoku **3D View**. Blender rysuje źdźbła trawy **Grass** jako małe, wygięte kreski (Rysunek 10.8.8b). Nie są specjalnie wyraźne, bo sama trawa jest krótka, taka jak na lotniskach polowych. W dodatku w **3D View** mają różny odcień (zapewne w wyniku różnic w orientacji normalnych). Ich kształt, kierunek itp. odpowiadają wartościom wpisanym w panelach **Emission**, **Velocity** i **Physics**. Możesz tu na bieżąco obserwować, jaki wpływ na trawę mają zmiany poszczególnych parametrów (**Normal**, **Random**, **Brownian**, ...).



- Im więcej źdźbeł trawy — tzn. im większa wartość **Emission: Number** (por. Rysunek 10.8.7b) — tym większy rozmiar pliku Blendera! Gdy ustawisz tę liczbę na 10 000, to rozmiar pliku zwiększy się o 2 MB.

Istnieje oszczędniejsza metoda zagęszczenia trawy: stworzenie wokół każdego źdźbła tzw. emisji potomnych (**Children**). Parametry tego efektu ustawia się w panelu **Children** (Rysunek 10.8.9):



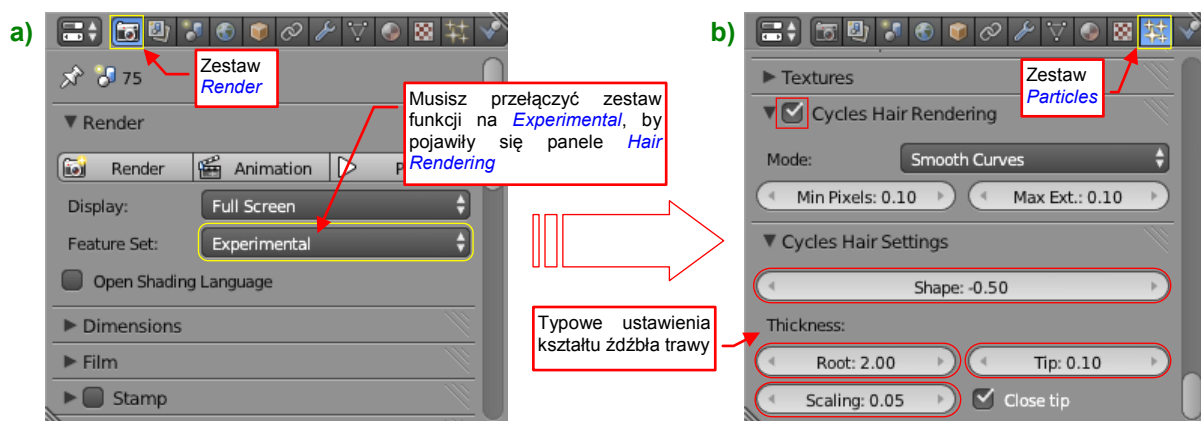
Rysunek 10.8.9 Zagęszczenie trawy za pomocą emisji potomnych (**Children**)

Najważniejsze są tu parametry z grupy **Children: Effects**. Na ilustracji oznaczyłem je kolorem żółtym. Zmieniłem także inne parametry, które zaznaczyłem na czerwono. To raczej zmiany opcjonalne, bo ich wpływ nie jest duży.

Na razie nasza trawa to niematerialne „druciki”. Są dwie metody aby nadać źdźbłom kształt na renderowanym obrazie:

- Użycie tych „drucików” wygenerowanych przez system cząsteczek jak krzywych, wzdłuż których można wygiąć pewną wzorcową siatkę. W tym celu należy w panelu **Render** wybrać tryb **Object** lub **Group**. Pojawią się wówczas odpowiednie pola, w których można wpisać nazwę wzorcowej grupy lub obiektu. To dokładna metoda, która daje użytkownikowi pełną kontrolę nad kształtem „wyhodowanej” w ten sposób trawy. Jest jednak bardziej pracochłonna (trzeba stworzyć to wzorcowe źdźbło);
- Utworzenie przez renderer automatycznego obrazu źdźbła/włosa wzdłuż każdej cząsteczki. Służy do tego opcja **Render: Path**. To szybsza metoda, choć nie tak dokładna jak pierwsza;

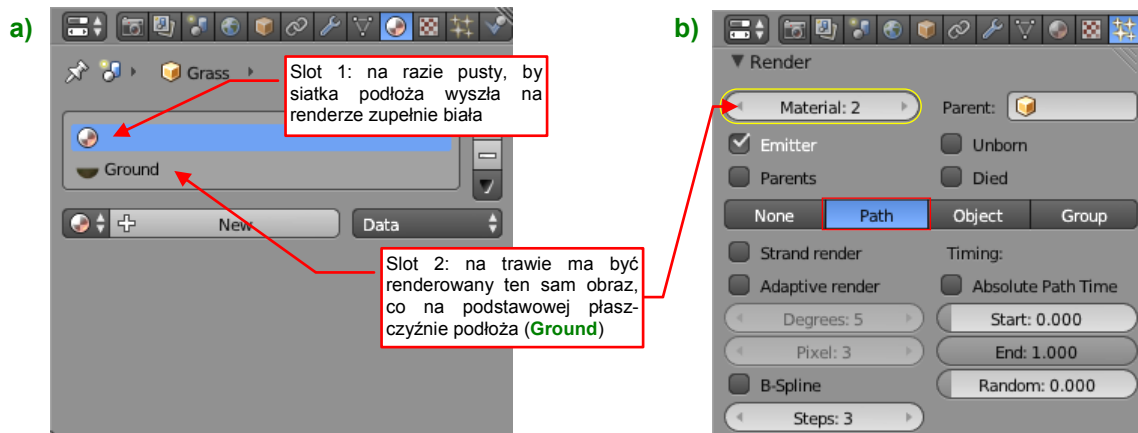
W tej ilustracji potrzebujemy tylko prostej trawy aby „złamać” krawędzie cienia pod samolotem, użyjemy więc tej prostszej metody (**Render: Path** — jest to zresztą metoda domyślna). W chwili pisania tej książki obsługująca tę opcję w Cycles funkcja **Hair Rendering** była zaliczona do zestawu „eksperymentalnego” — trzeba go włączyć w zestawie **Render** (Rysunek 10.8.10a):



Rysunek 10.8.10 Ustawienia renderowania trawy w Cycles

To spowoduje pojawienie się na końcu zestawu **Particles** paneli **Cycles Hair Rendering** i **Cycles Hair Settings**. Ustaw ich parametry tak, jak pokazuje to Rysunek 10.8.10b).

Siatka trawy ma zazwyczaj przypisane dwa materiały. Pierwszy z nich to materiał podłoża. W naszym przypadku najlepiej, aby był zupełnie przezroczysty: nie będzie przeszkadzał w sterowaniu przezroczystością podstawowej płaszczyzny **Ground** (por. str. 452). Drugi materiał to materiał nałożony na źdźbła trawy. Na naszym obrazie powinien mieć takie same barwy jak obraz podłoża. Użyjemy więc na początek materiału **Ground**. W siatce **Grass** przypisałem go do slotu 2 (Rysunek 10.8.11a) i to przypisanie zaznaczyłem w polu **Material** (slot) panelu **Particles:Render** (Rysunek 10.8.11b):



Rysunek 10.8.11 Wstępne ustawienia materiałów siatki **Grass**

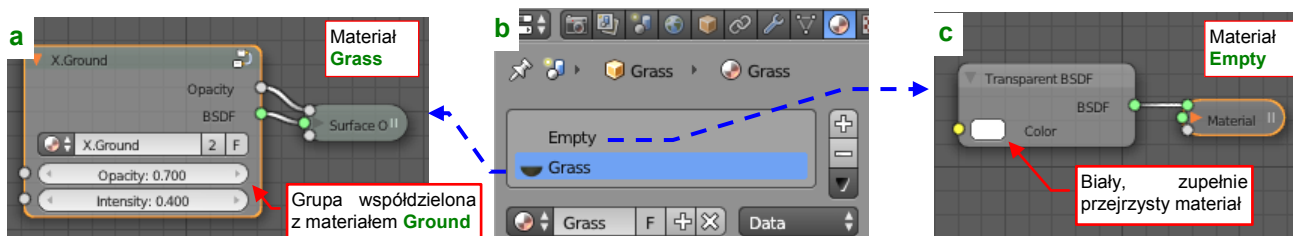
Slot 1 na razie pozostawiam pusty, aby materiał podłoża był na renderze zupełnie biał. Chcę po prostu za pierwszym razem zobaczyć źdźbła trawy na jakimś kontrastowym tle i przekonać się że w ogóle istnieją i jaką mają gęstość. Uzyskany rezultat przedstawia Rysunek 10.8.12.

Wygląda na to że gęstość trawy na renderze jest o wiele mniejsza niż na podglądzie, który mamy w **3D View**. W dodatku źdźbła sprawiają wrażenie ciemnych — ale może to wpływ białego podłoża? Z drugiej strony wygląda na to że światło słońca odbite od tej testowej białej powierzchni rozjaśnia wszystko wokół (zwróć uwagę że spód samolotu jest tu jaśniejszy niż na renderze z str. 452)...



Rysunek 10.8.12 Testowy render trawy

Pozostaje sprawdzić te wątpliwości dla takiego podłoża, jakie chcemy zastosować na ostatecznym renderze. Przypiszmy więc teraz do siatki **Grass** docelowe materiały (Rysunek 10.8.14):



Rysunek 10.8.13 Ostateczne przypisanie materiałów do siatki **Grass**

Ścianom z których jest „emitowana” trawa przypisany jest materiał ze slotu 1, o nazwie **Empty** (Rysunek 10.8.14b). To zupełnie przezroczysta, biała powierzchnia (Rysunek 10.8.14c). Trawie przypisałem kopię materiału **Ground**, którą nazwałem **Grass** (zaraz zobaczysz, dlaczego taki podział jest potrzebny). Aby jednak obydwa materiały — **Grass** i **Ground** — korzystały z tych samych tekstur, ustawionych w ten sam sposób, w obydwu wykorzystuję pomocniczą grupę **X.Ground**. Jej parametrami są nieprzezroczystość (**Opacity**) i intensywność barw tekstury (**Intensity**).

Jak więc wygląda nasza trawa na renderze? Pokazuje to Rysunek 10.8.14:



Rysunek 10.8.14 Próbny render trawy (Grass = Ground)

Teraz można powiedzieć na pewno, że jest zdecydowanie za ciemna! Dodatkowo wyraźnie widać że trzeba zmienić jej gęstość, aby rosła tam, gdzie na zdjęciu są kępki trawy.

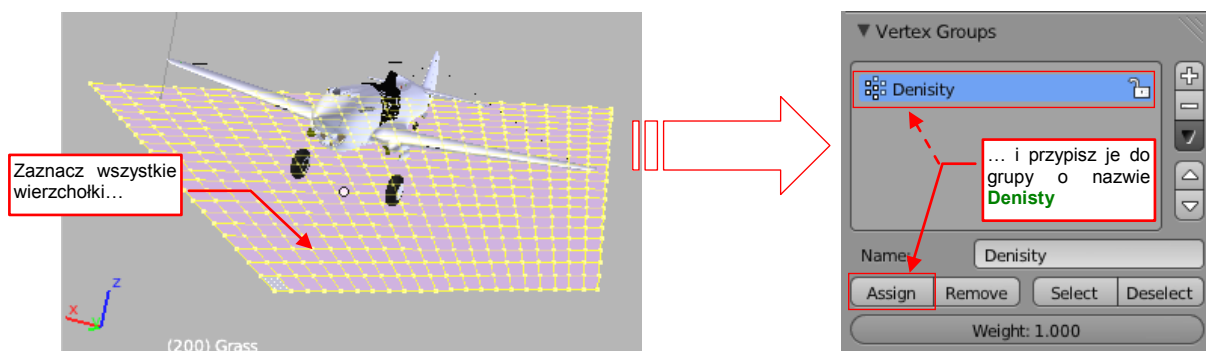
Aby coś poradzić na zbyt ciemne żdźbła, spróbujemy zwiększyć intensywność kolorów w materiale **Grass**, zwiększając parametr **X.Green:Intensity** z 0.4 do 2.0. Otrzymamy o wiele lepszy wynik (Rysunek 10.8.15):



Rysunek 10.8.15 Próbny render trawy (zwiększona jasność barw w materiale Grass)

Całość wygląda o wiele lepiej! Oświetlone obszary trawy stały się jasne (może nawet odrobinę za bardzo), a te w cieniu pasują teraz do odcienia zdjęcia. Myślę, że odpowiednia intensywność **X.Green** dla trawy na tym obrazie jest w zakresie od 1.8 do 2.0.

Do sterowania lokalną gęstością trawy użyjemy wag, malowanych na powierzchni siatki (por. str. 291). Najpierw stwórz nową grupę wierzchołków o nazwie **Density**, a potem przypisz do niej wszystkie wierzchołki siatki emitera (**Grass**) (Rysunek 10.8.16):

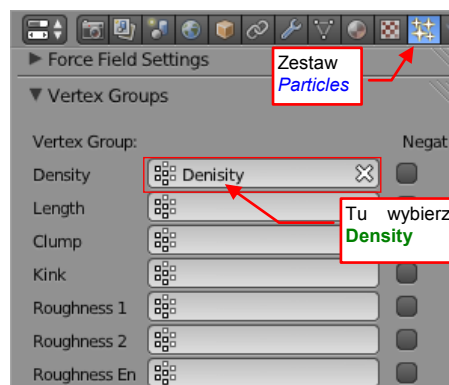


Rysunek 10.8.16 Stworzenie w siatce emitera trawy grupy **Density**



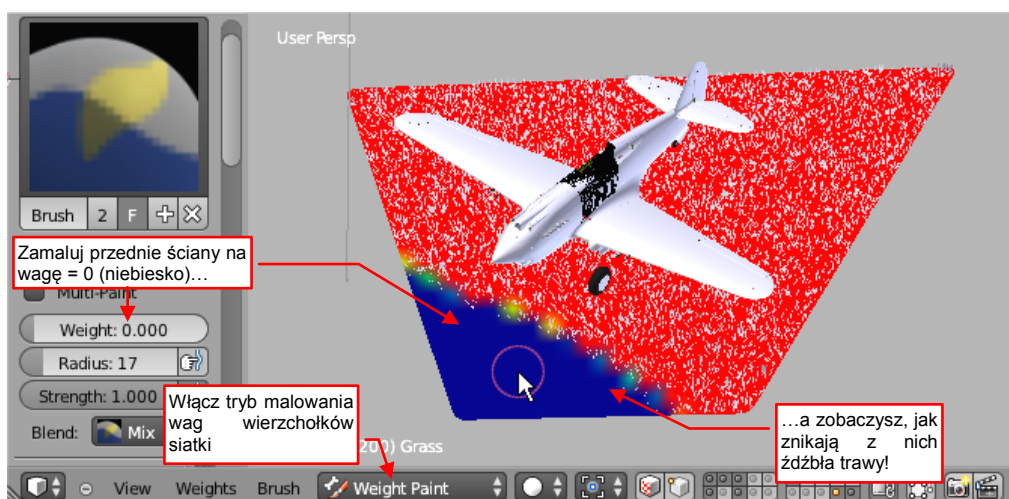
(Więcej o grupach wierzchołków znajdziesz na str. 278). Potem przejdź do zestawu *Particles*. W panelu *Vertex Groups* przypisz parametrowi *Density* przygotowaną przed chwilą grupę wierzchołków (*Density*) (Rysunek 10.8.17):

Przy okazji — zwróć uwagę na pozostałe parametry, którymi możesz sterować za pomocą wag. Na przykład — długość trawy (*Length*). Wystarczy zdefiniować kolejną grupę wierzchołków i przypisać ją tutaj. (Pamiętaj, że ten sam wierzchołek siatki może należeć do wielu grup, więc nie ma tu żadnego konfliktu — możesz stworzyć wiele grup wierzchołków obejmujących całą siatkę, każdą z inną wagą).



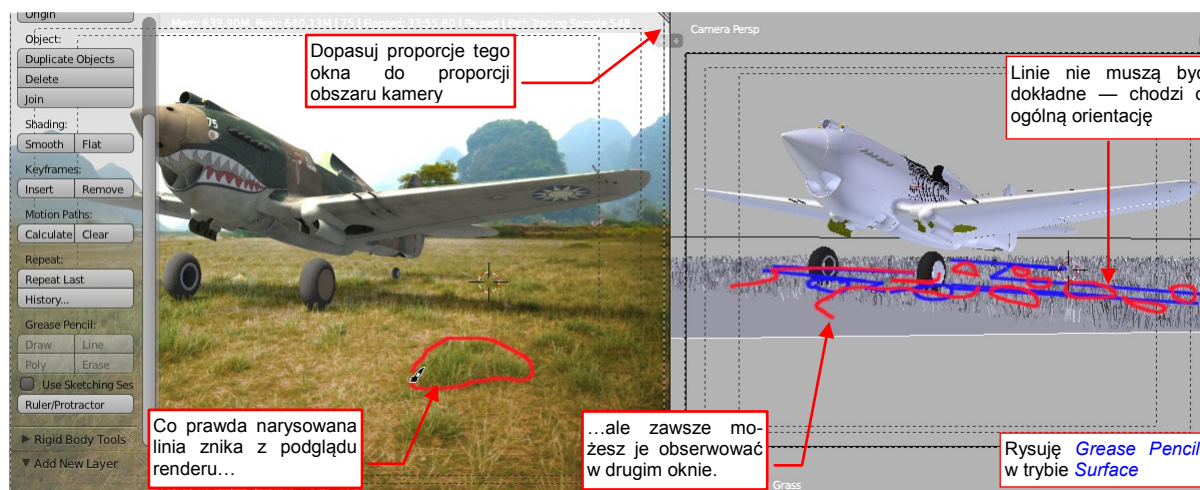
Rysunek 10.8.17 Przypisanie wag grupy wierzchołków (*Density*) do gęstości trawy

Teraz przełącz się w tryb *Weight Paint* (szczegóły — por. str. 291) i zamaluj wagą = 0 ściany siatki na które na pewno nie będzie padał cień samolotu (Rysunek 10.8.18):



Rysunek 10.8.18 Malowanie gęstości trawy

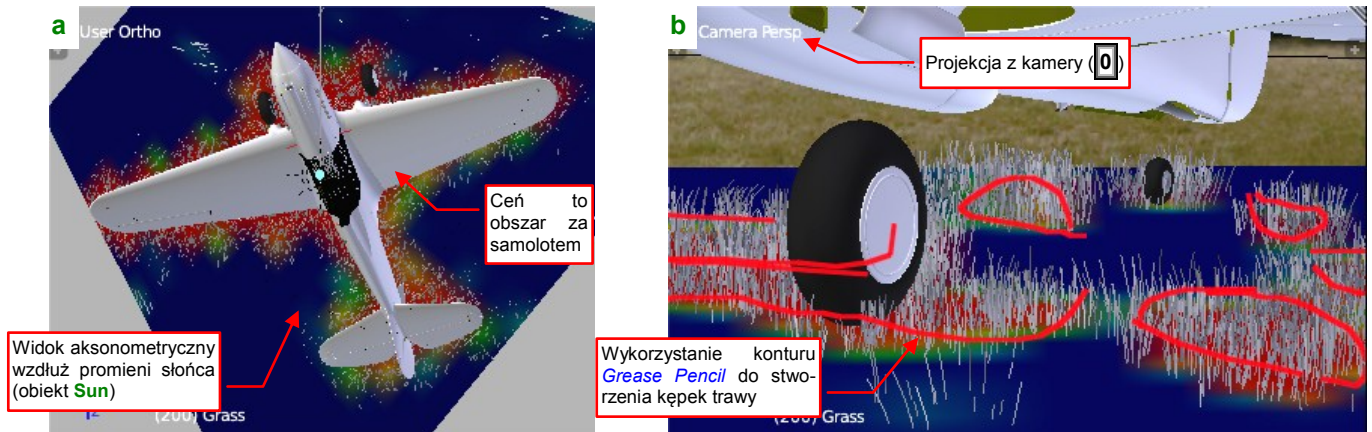
W trakcie malowania zobaczysz, jak znikają z nich żółbła. To bardzo wygodna informacja zwrotna: możesz na bieżąco decydować, gdzie zwiększyć wagę (kolor czerwony), a gdzie zmniejszyć (kolor niebieski). Trawę powinniśmy zostawić tam, gdzie pada cień samolotu. Ale gdzie to dokładnie jest? Spróbowałem obrysować te obszary za pomocą narzędzia *Grease Pencil* (por. str. 421) — cień na niebiesko, kępki trawy na czerwono (Rysunek 10.8.19):



Rysunek 10.8.19 Zaznaczanie obszaru cienia i trawy (*Grease Pencil*)



Inną metodą na znalezienie cienia samolotu na podłożu jest ustawienie w **3D View** aksonometrycznej projekcji (**Ortho**) wzdłuż promieni słońca (**View→Align View→Align View to Selected→Top**). Wtedy wystarczy zamalować na niebiesko (tzn. wagą = 0) obszaru nie zasłoniętego przez samolot (Rysunek 10.8.20a):



Rysunek 10.8.20 Zaznaczanie obszarów z trawą (c.d.)

Gdy trawa pozostała w cieniu samolotu, w widoku z kamery (0) „wytnij” prześwitły pomiędzy kępkami odrysowanymi ze zdjęcia (Rysunek 10.8.20b).

- Na czas modyfikacji siatki w trybie **Weight Paint**, trawa znika z podglądu renderu. Pojawi się ponownie, gdy przełączysz aktualny tryb na **Object Mode**.

Rysunek 10.8.21 przedstawia kolejny próbny render. Widać na nim bardziej naturalną, nieregularną gęstość trawy, z lokalnymi prześwitami:



Rysunek 10.8.21 Rezultat — cień samolotu zaburzony przez kępki trawy

## Dodatki

Czytając wcześniejsze rozdziały, zapewne przekonałeś już się, że uwielbiam umieszczać u dołu strony przypisy z dodatkowymi informacjami. Ta część książki to właściwie wybór takich "uwag na marginesie", które w trakcie pisania urosły do rozmiaru całej sekcji ☺.

Jak w każdym dodatku, są to informacje, z których nie musisz korzystać, aby stworzyć e-model samolotu. Czasami jednak mogą się przydać, by coś wyjaśnić lub ułatwić. Mam jednak nadzieję, że znajdziesz tu rozwinięcie wielu zagadnień, które świadomie skracałem w głównym tekście książki.

---

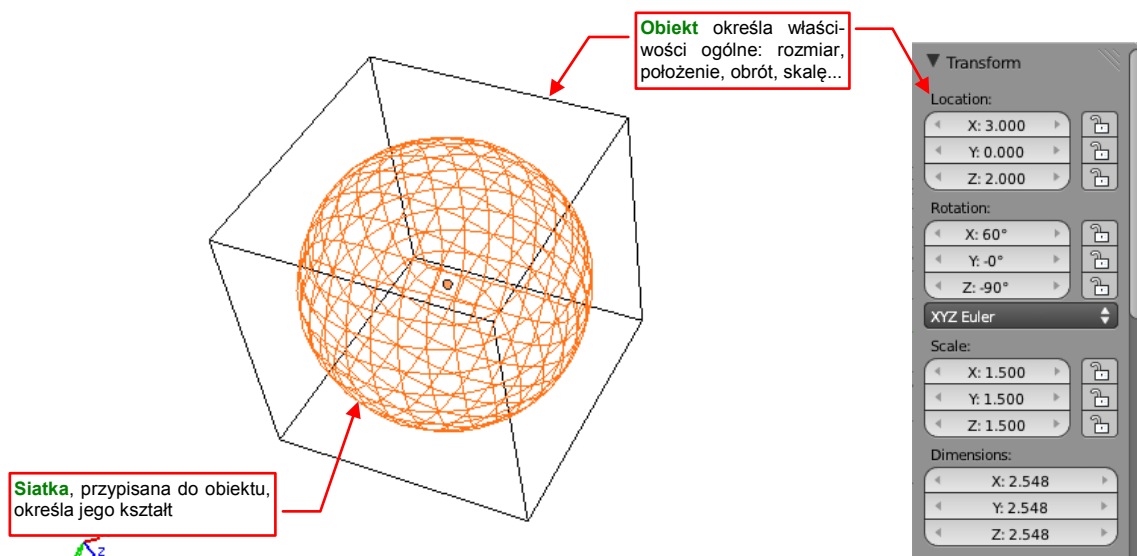
## **Rozdział 11. Dodatkowe wyjaśnienia**

W tym rozdziale umieściłem informacje "dla dociekliwych". Znajdują się tu szczegółowe wyjaśnienia wybranych zagadnień, nad którymi nie chciałem się rozwodzić w części głównej. (Aby nie zbaczać z tematu).

### 11.1 Struktura danych modelu i sceny w Blenderze

Zapewne wiele razy, podczas pracy z Blenderem, mignęło Ci przed oczami tajemnicze słowo "[datablock](#)" ("blok danych"). Pojawia się w różnych miejscach i znaczeniach. Gdy zrozumiesz, czym w istocie jest dla Blendera "blok danych", zrozumiesz także zasady, rządzące elementami rysunku (pliku). Pozwoli Ci to poprawnie przewidywać zachowanie programu, i bardzo ułatwi pracę<sup>1</sup>.



Każdy model, który stworzyłeś w Blenderze, złożony jest z obiektów ([Object](#)). Obiekt możesz traktować jak doskonale przejrzyste "pudełko": ma określony rozmiar, środek (punkt odniesienia), położenie, obrót, skalę. Obiekt "sam z siebie" nie ma żadnego kształtu. Kształt określa siatka ([Mesh](#)). Siatka właśnie jest pewnym rodzajem "bloku danych" ([datablock](#)). Zawiera złożone informacje o wierzchołkach, krawędziach, ścianach powłoki, która ma być rysowana na ekranie. Zazwyczaj każdy obiekt ma przypisaną "własną" siatkę<sup>2</sup> (Rysunek 11.1.1):




Rysunek 11.1.1 Obiekt i jego siatka

Czasami ten sam blok danych — siatka — może być użyta przez kilka różnych obiektów. Przykładem takiej sytuacji jest śmigło modelu P-40 (Rysunek 11.1.2).

To śmigło składa się z trzech identycznych łopat. Rysunek 11.1.2 przy każdym z tych obiektów pokazuje kontekst, wyświetlany w oknie [Properties](#). Dwie nazwy, które są na nim widoczne, to odpowiednio:

- poprzedzona ikoną : nazwa obiektu;
- poprzedzona ikoną : nazwa siatki;

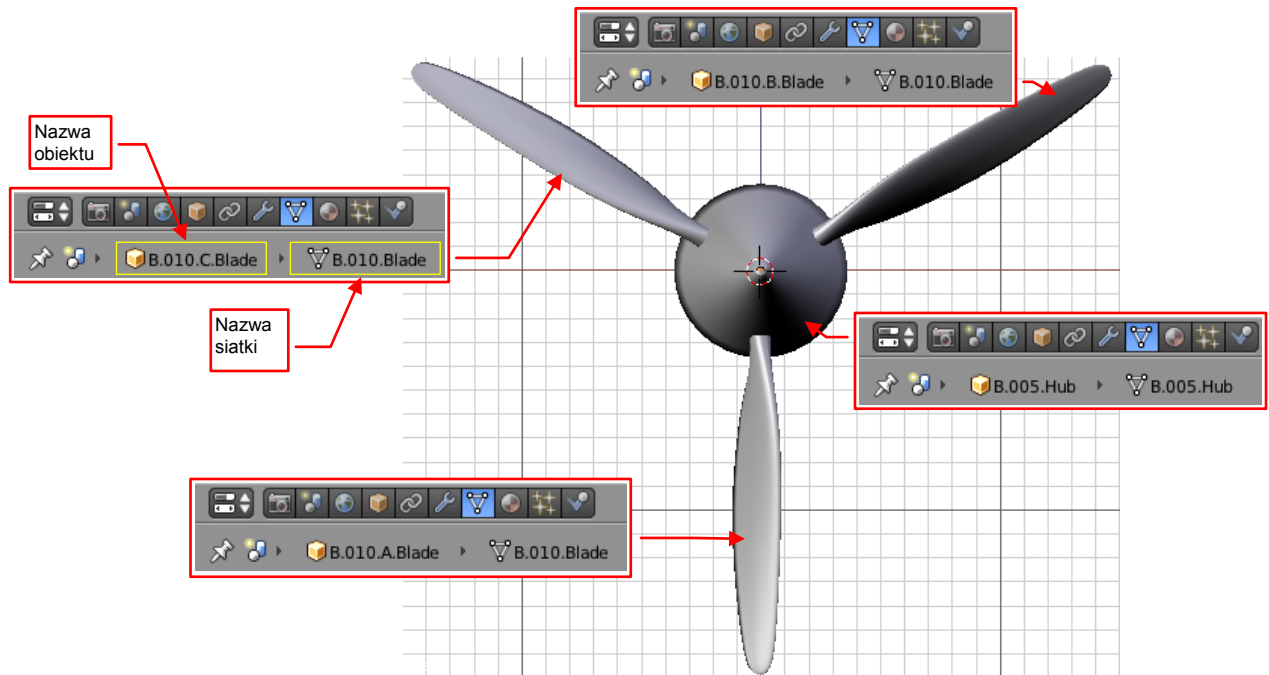
Poprzedzająca je ikona  symbolizuje aktywną scenę, w której znajduje się obiekt. Nazwa aktywnej sceny nie jest tu wyświetlana, bo zazwyczaj widać ją przez cały czas u góry ekranu, w nagłówku okna [Info](#).

<sup>1</sup> "[Datablock](#)" należy do słów, którym udaje się czasami wymknąć z żargonu programistów. W istocie każdy większy program posiada swoje własne słownictwo. Składa się na nie określenia żargonowe, które twórcy używali tak często, że bezwiednie zaczęli umieszczać w komunikatach i objaśnieniach. Ci ludzie żyją z tymi słowami na co dzień, i zupełnie nie zauważają, że "zwykli" użytkownicy mogą ich nie rozumieć. Na przykład w AutoCAD-zie podobnym słowem było "[entity](#)", przeniesione żywcem z fachowej terminologii baz danych.

<sup>2</sup> Choć nie jest to regułą. W Blenderze istnieje specjalny rodzaj obiektu "pustego", który nie zawiera żadnej siatki. Można go utworzyć za pomocą polecenia [Add → Empty](#). "Puste" obiekty używane są do wielu pomocniczych ról — oznaczania ważnego miejsca modelu, określania orientacji tekstur, itp.

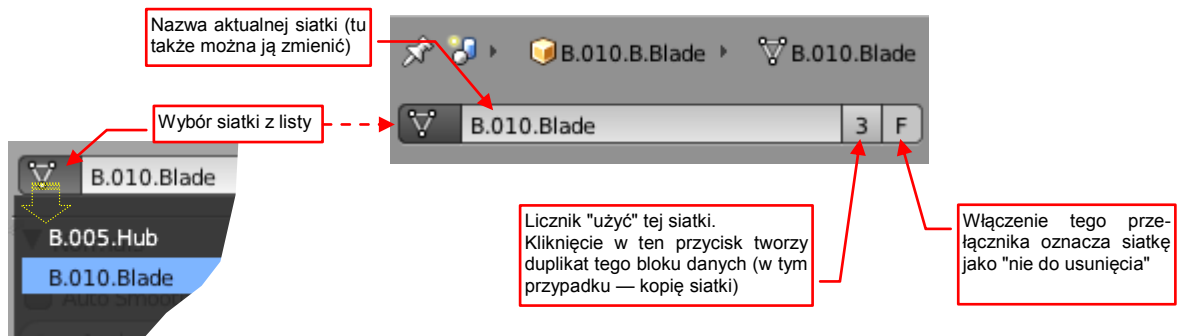


Przyjrzyj się dokładnie opisom, wyświetlanym przez konteksty, które pokazuje Rysunek 11.1.2, a stwierdzisz, że każda z łopat jest obiektem o nazwie wyróżnionej środkową literą. (**B.010.A.Blade**, **B.010.B.Blade**, **B.010.C.Blade**). Zwróć także uwagę, że każdy z tych obiektów używa tej samej siatki — **B.010.Blade**. Gdy zmodyfikujesz kształt siatki w którymkolwiek z nich — zmienisz kształt wszystkich trzech łopat.



Rysunek 11.1.2 Łopaty śmigła — przykład jednej siatki współdzielonej przez trzy obiekty

Przypisanie bloku danych — w tym przypadku siatki do obiektu — jest na panelach Blendera obsługiwane przez typowy zestaw kontrolki (Rysunek 11.1.3) :



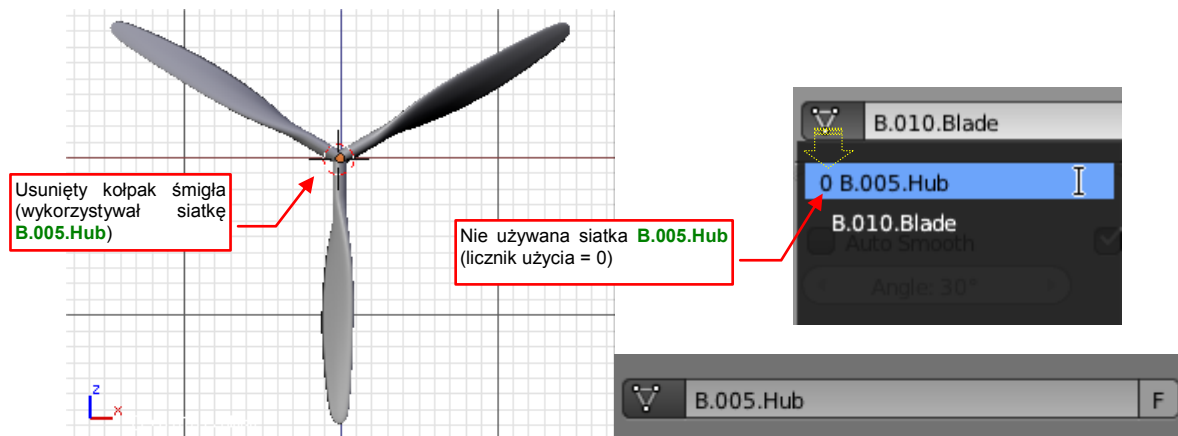
Rysunek 11.1.3 Kontrolki, zarządzające przypisaniem bloku danych

Poniżej opisu kontekstu zestawu **Object Data** wyświetlana jest nazwa siatki, aktualnie przypisanej do obiektu. Aby ją zmienić, należy wybrać inną z listy rozwijalnej, otwieranej przyciskiem po lewej. Jeżeli wybierzesz w tym miejscu siatkę o innym kształcie — zmieni się kształt obiektu.

Po prawej stronie nazwy bloku danych umieszczony jest licznik "użycia" (referencji). Siatka **B.010.Blade** jest przypisana do trzech obiektów, stąd na ilustracji widoczna jest liczba 3. Kiedy klikniesz w ten przycisk — Blender utworzy duplikat aktualnej siatki. Duplikat nosi taką samą nazwę jak pierwowzór, z dodatkową końcówką ".001". Duplikat zastępuje oryginalną siatkę, przypisaną do obiektu.

Licznik referencji ("użycia") bloku danych jest w Blenderze niezwykle ważny. Gdy siatka nie jest "użyta" przez żaden obiekt — jej licznik spada do zera. Wszelkie bloki danych o liczniku równym zero są pomijane przy zapisie pliku na dysk. Kołpak śmigła w naszym przykładzie — **B.005.Hub** — miał własną siatkę, o identycznej nazwie: **B.005.Hub**. Gdy usuniemy ten obiekt (**Delete**), siatka **B.005.Hub** nadal będzie istnieć, ale jej licznik "użycia" spadnie do zera.

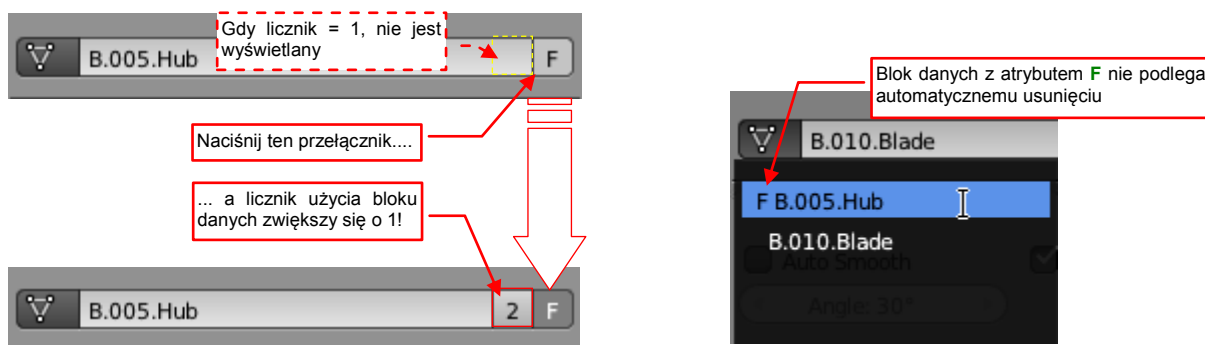
Taki nie używany blok danych Blender oznacza na liście literą "O" (Rysunek 11.1.4) :



Rysunek 11.1.4 Siatka usuniętego kołpaka śmigła

- Wszystkie bloki danych, które nie są wykorzystywane ("osierocone"), Blender pomija przy zapisie na dysk. Są w ten sposób, z opóźnieniem, usuwane z pliku.

Mimo wszystko, jeżeli tego chcesz, możesz uchronić siatkę B.005.Hub, przed "czyszką" podczas zapisu. Trick polega na włączeniu przełącznik **F** (*Fake user*) przed usunięciem obiektu, który zawiera ten blok danych (Rysunek 11.1.5):



Rysunek 11.1.5 Zabezpieczenie bloku danych przed usunięciem — "Falszywe" (*Fake*) użycie

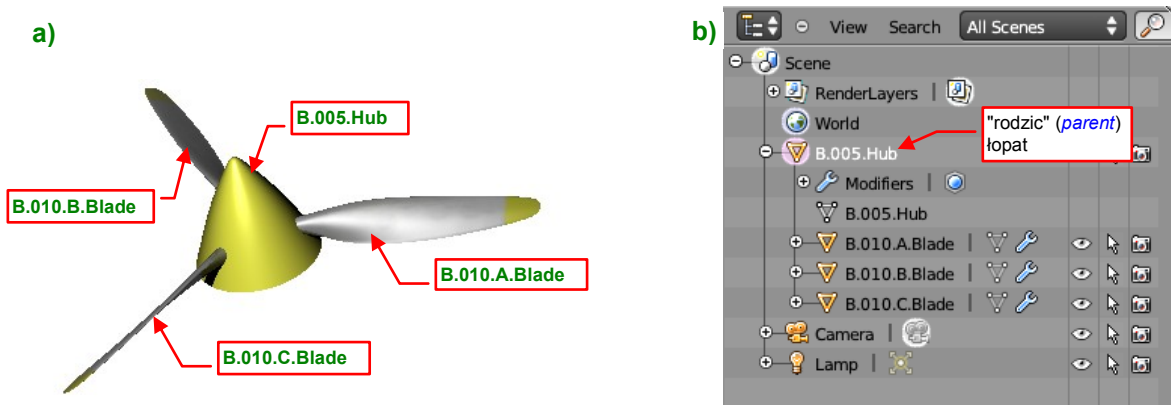
Włączenie **F** powoduje zwiększenie licznika "użycia" o 1. (Oczywiście, zawsze możesz go z powrotem wyłączyć). Dzięki przełącznikowi **F**, licznik użycia bloku danych zawsze jest > 0.

- Blok danych z atrybutem **F** nigdy nie będzie usunięty z pliku Blendera. (Nawet wtedy, gdy nie jest użyty przez jakiegokolwiek obiekt, jego licznik użycia jest = 1).

Blokiem danych w Blenderze jest praktycznie wszystko, z czym pracujesz tworząc scenę. W szczególności są to takie typowe elementy, jak:

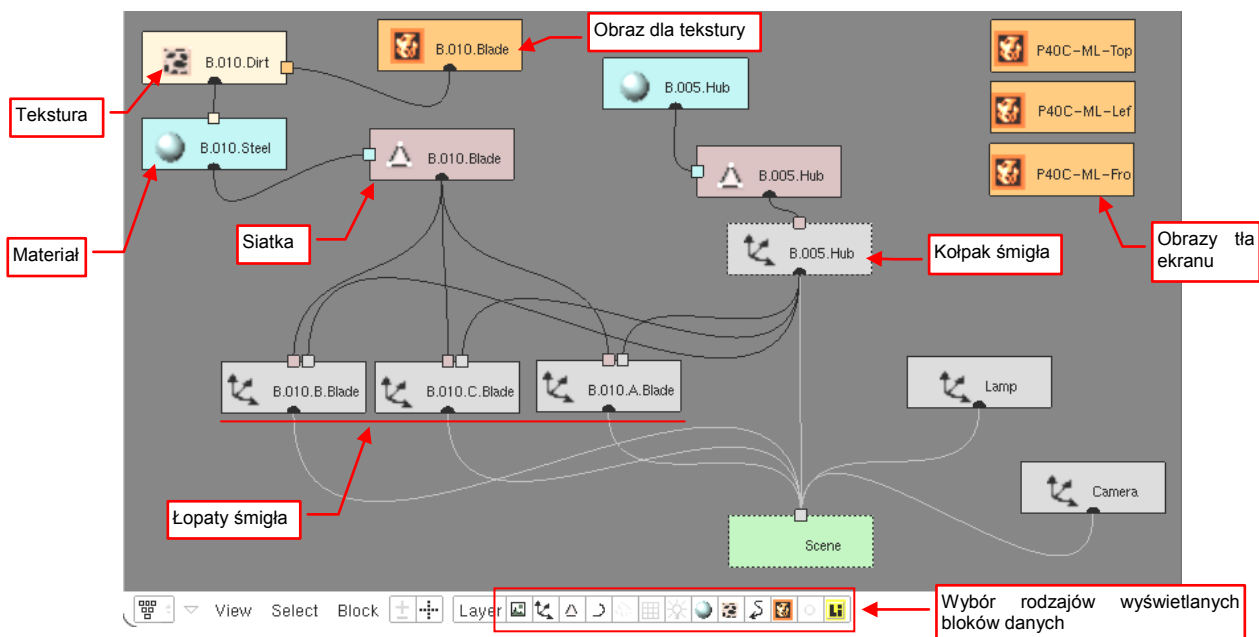
- siatki** (*meshes* — o nich już mówiliśmy): nadają obiektom kształt. Zawierają także inne szczegóły, m.in. współrzędne mapowania tekstury dla każdego wierzchołka;
- kamery** (*cameras*): pomocniczy rodzaj obiektu, definiujący projekcję. Używany podczas renderowania;
- lampy** (*lamps*): źródła światła, używane podczas renderowania
- obiekty** (*objects*): to „obudowa”, której zawartość (określana jako *Object Data*) jest referencją do jednego z bloków danych wyliczonych powyżej;
- materiały** (*materials*), przypisane do siatek: nadają powierzchniom podstawową barwę, oraz określają inne ich parametry, takie jak przejrzystość, połyskliwość, itp.;
- tekstury** (*textures*), przypisane do materiałów: wzbogacają jednolitą barwę materiału elementy w innych kolorach, zabrudzenia. Pozwalają także uzyskać efekt drobnych nierówności;
- obrazy** (*images*), przypisane do tekstur: zawierają obraz, wykorzystywany przez teksturę;

Prezentowany wcześniej model kołpaka śmigła składa się z czterech obiektów: kołpaka śmigła (**B.005.Hub**) i trzech łopat (**B.010.x.Blade**) (Rysunek 11.1.6a). Łopaty są przypisane (relacją *parent*) do kołpaka (Rysunek 11.1.6b):



Rysunek 11.1.6 Śmigło i jego struktura (w oknie *Outliner*)

Kołpak ma przypisany materiał **B.005.Hub** (nadający mu żółty kolor). Łopaty wykorzystują materiał **B.010.Steel**, który ma przypisaną teksturę **B.010.Dirt**. Tekstura **B.010.Dirt** korzysta z obrazu **B.010.Blade**, nakładając na łopatę żółte końcówki i zabrudzenia. Rysunek 11.1.7 przedstawia to na tzw. schemacie *Oops*:



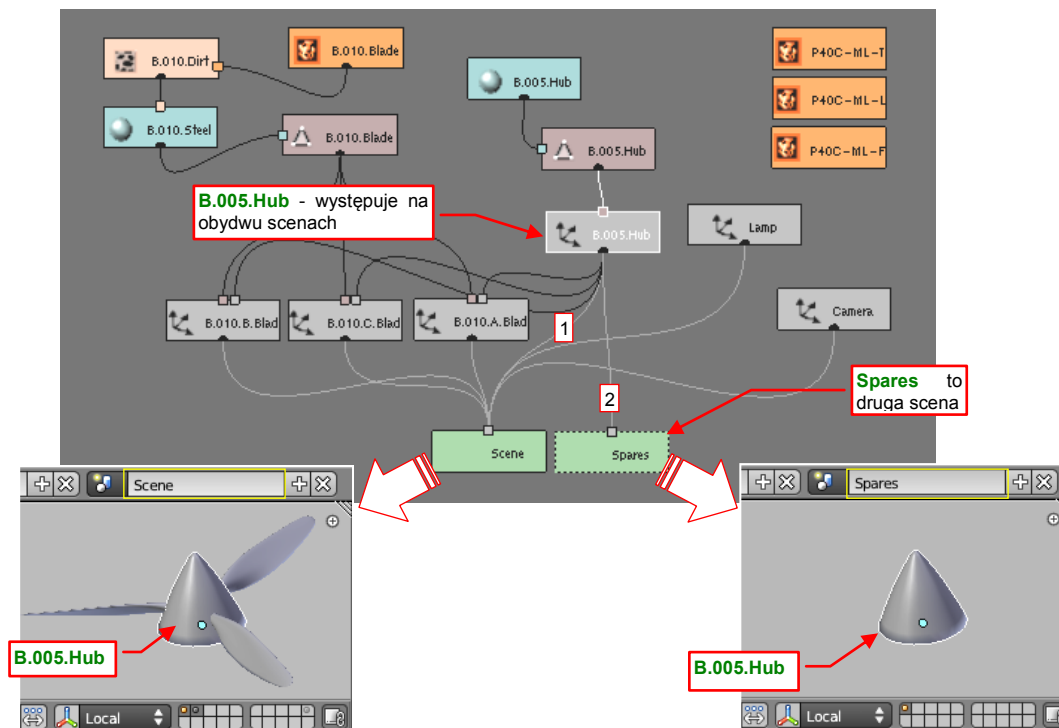
Rysunek 11.1.7 Inny sposób przedstawienia struktury modelu śmigła (schemat *Oops* z Blendera 2.49)

Schemat *Oops* był dostępny w Blenderze 2.4, w wersji 2.5 z niego zrezygnowano. Pozostawiłem go tutaj, gdyż dobrze obrazuje strukturę pliku Blendera. Każdy prostokąt, który na nim widzisz, to jakiś blok danych. Każda z linii oznacza pojedyncze zwiększenie licznika bloku danych, z którego wychodzi. Na przykład: tekstura **B.010.Dirt** jest używana tylko raz. Jeżeli usuniesz ją z materiału **B.010.Steel**, zostanie pominięta przy najbliższym zapisaniu pliku. "Osierocony" zostanie wówczas używany przez tę teksturę obraz **B.010.Blade**. Ten obraz zostanie pominięty przy kolejnym zapisie. (Stąd uwaga praktyczna: czasami trzeba dwa — trzy razy zapisać, zamknąć i otworzyć plik, by wszelkie niepotrzebne "śmieci" zostały z niego usunięte).

W prawym górnym narożniku schematu (Rysunek 11.1.7) widać trzy obrazy, które nie są "powiązane" z niczym. Dlaczego nie zostały usunięte? Ponieważ są to tła poszczególnych widoków 3D: rysunki samolotu w rzucie z góry, lewej i przodu. Każdy z nich ma w istocie licznik użycia = 1, bo jest przypisany do konkretnego okna *3D View* (jako *Background Image*). Schemat *Oops* nie pokazywał wszelkich połączeń w pliku Blendera, obrazował tylko strukturę sceny. Np. innym rodzajem bloku danych są teksty, które można przeglądać w oknie *Text Editor*. To także bloki danych "używane" przez plik, a nie konkretną scenę.

Rysunek 11.1.7 pokazuje, że obiekty — **B.005.Hub**, **B.010.x.Blade**, a także **Lamp** i **Camera** — są także blokami danych. Ich "użytkownikiem", dzięki któremu mają przypisane "użycia", jest scena.

Rysunek 11.1.8 demonstruje strukturę pliku Blendera, w którym utworzyłem drugą scenę (por. str. 199) — nazywałem ją **Spares**.



Rysunek 11.1.8 Dwie oddzielne sceny w jednym pliku

Scena — to brzmi dumnie. W tym przypadku jednak to tylko nowa, pusta przestrzeń, w której umieściłem drugi raz kołpak śmigła<sup>1</sup>. Popatrz na schemat *Oops* (Rysunek 11.1.8) — obiektowi **B.005.Hub** przybyła jeszcze jedna linia, czyli jeszcze jedno "użycie". Jeżeli teraz nawet usunę ten obiekt ze sceny **Scene**, to pozostanie w drugiej scenie — **Spares**. Zawsze mogę go stamtąd przywołać. W ten sposób można używać drugiej sceny jak "podręcznego składziku". W przestrzeni **Spares** można przechowywać różne obiekty, które się jeszcze przydadzą, a których nie chcę widzieć w podstawowej scenie.

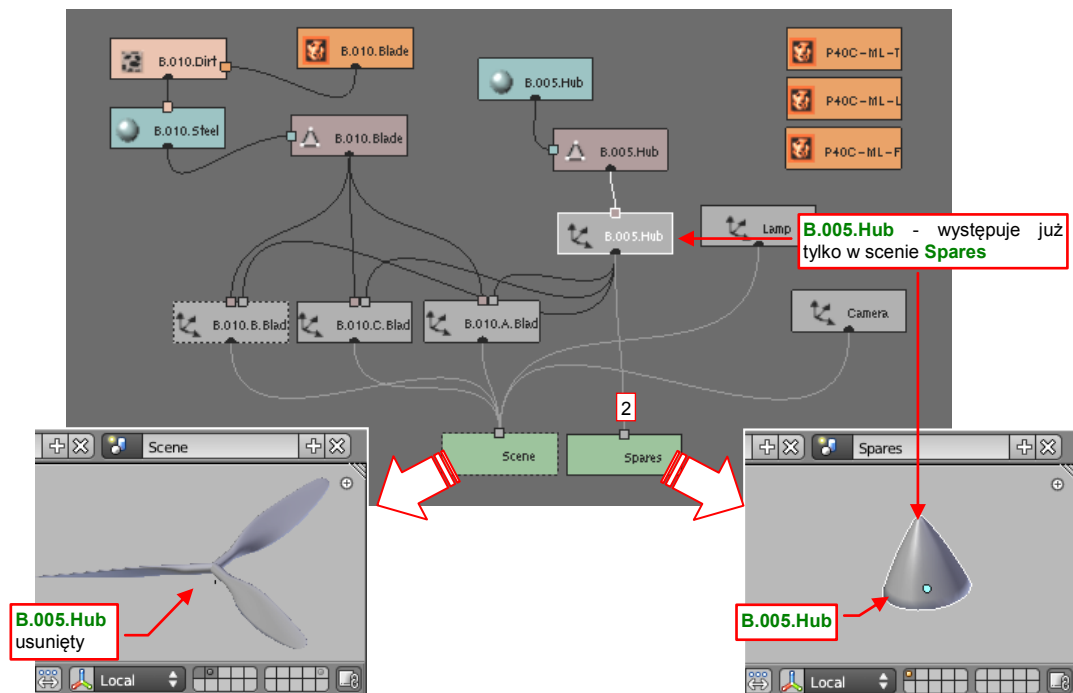
Obiekt, który jest użyty na wielu scenach, w każdej z nich ma to samo położenie, obrót, skalę. W końcu nic dziwnego — każda ze scen zawiera tylko odnośnik do jednego bloku danych, opisującego obiekt.

Rysunek 11.1.9 przedstawia strukturę pliku po usunięciu ze sceny **Scene** kołpaka śmigła (obiektu **B.005.Hub**). Usunięcie spowodowało zmniejszenie licznika użycia obiektu **B.005.Hub** o 1. Gdyby występował tylko na jednej scenie — zostałby pominięty przy najbliższym zapisie na dysk<sup>2</sup>. Tak się jednak nie stanie, gdyż **B.005.Hub** nadal występuje w drugiej scenie, i jego licznik jest nadal większy od 0.

<sup>1</sup> poleceniem *Object → Make Links → To Scene*

<sup>2</sup> Zauważ, że mimo usunięcia ze przestrzeni **Scene**, obiekt **B.005.Hub** pozostał obiektem nadrzędnym (*parent*) dla łopatek śmigła (Rysunek 11.1.9). Dzięki temu ma trzy dodatkowe "użycia". To wygląda trochę na jakieś niedopatrzenie w Blenderze, gdyż w sytuacji jak na rysunku te powiązania nie działają poprawnie. (Gdy w przestrzeni **Spares** przesuń lub obróć **B.005.Hub**, nie spowoduje to żadnych zmian w położeniu łopat w przestrzeni **Scene**). W dodatku, gdyby **B.005.Hub** istniał tylko na scenie **Scene**, jego usunięcie usunęłoby także powiązania typu "*parent*" z łopatkami. W efekcie zostałby "wyrugowany" z pliku, tak jak to opisałem.





Rysunek 11.1.9 Struktura pliku po usunięciu obiektu B.005.Hub ze sceny Scene

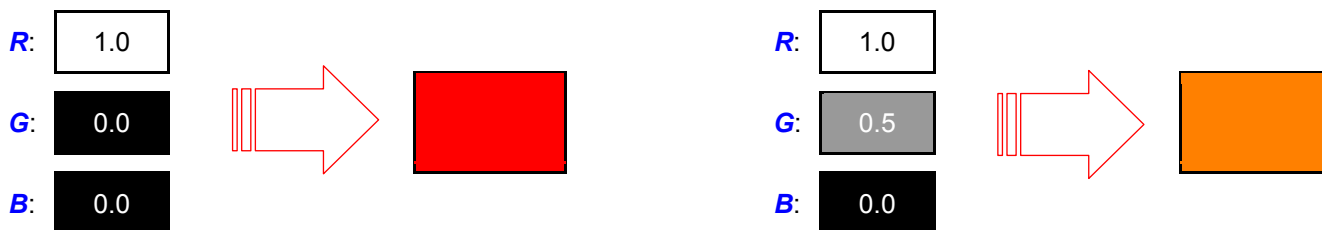
- Plik Blendera może zawierać wiele scen. Te same elementy modelu może występować w każdej z nich. Można ich także użyć tylko w jednej ze scen.

Za pomocą scen można przechowywać w jednym pliku różne wersje samolotu. Części wspólne dla wszystkich wersji są wówczas współdzielone pomiędzy scenami. Części specyficzne — występują tylko w jednej scenie.

## 11.2 Arytmetyka barw

Każdy z programów, wykorzystywanych w tej książce, używa barw wyrażonych poprzez intensywność trzech niezależnych składników (kanałów): czerwonego (**R**), zielonego (**G**) i niebieskiego (**B**). Taki model często określa się jako „RGB”, i jest niezwykle popularny w grafice komputerowej.

Intensywność każdego ze składników barwy jest wyrażona liczbą od 0.0 (brak) do 1.0 („najjaśniejszy”). Rysunek 11.2.1 pokazuje, jak z wartości **R**, **G**, **B** jest tworzona barwa:



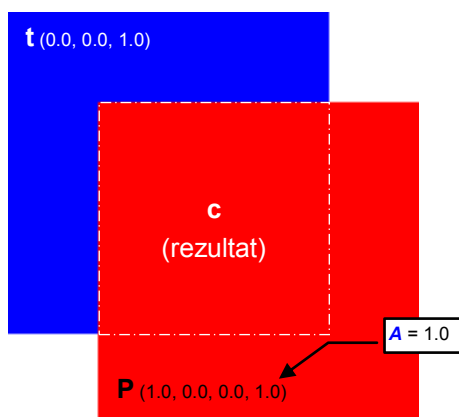
Rysunek 11.2.1 Przykład zakodowania barwy za pomocą składników RGB — różne odcienie czerwieni

Czasami wraz z trójką **R**, **G**, **B** występuje czwarty składnik — nieprzejrzystość. Oznaczany jest jako „alfa” (**A**) i używany jako dodatkowy czynnik przy łączeniu barw. Steruje udziałem każdego ze składników. **A** = 1.0 oznacza kompletną nieprzejrzystość, **A** = 0.0 — zupełną przejrzystość.

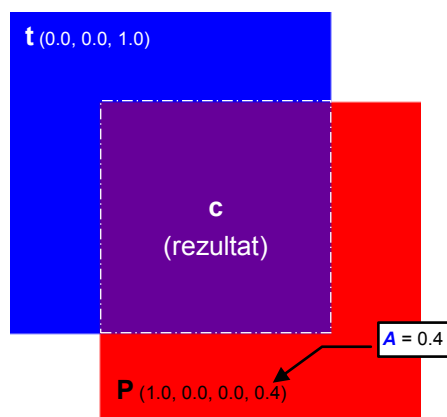
Aby przyspieszyć obliczenia i zmniejszyć ilość pamięci, potrzebnej do przechowywania obrazu, składniki **R**, **G**, **B**, **A** zapisuje się w obrazie jako 8-bitowe liczby całkowite (o wartościach od 0 do 255). W takim przypadku 255 odpowiada oryginalnej wartości 1.0. Z zapisu barw na liczbach całkowitych wziął się, występujący w GIMP i Inkscape, zwyczaj podawania intensywności składnika barwy w skali od 0 do 255.

We wszystkich programach, opisywanych w tej książce, pojawiają się nazwy operacji, przeprowadzanych na barwie (lub intensywności) pikseli: **Add**, **Multiply**, **Subtract**, .... W przypadku Blendera jest to tryb nakładania tekstury na podłoże — czyli sumę ogólnej barwy materiału i nałożonych wcześniej tekstur. W przypadku Gimpa to tryb **Mode**, a w Inkscape — **Blend Mode**. Wszystkie służą do nakładania obrazu z aktualnej warstwy na warstwy poprzednie (czyli sumę warstw leżących „poniżej”). Postaram się w tej sekcji wyjaśnić efekty każdego z tych trybów, gdyż czasami nie są intuicyjne.

Zacznijmy od zwykłego złożenia dwóch warstw — w Blenderze to nazywa się **Mix**, a w Gimpie i Inkscape — **Normal**. Rysunek 11.2.2 pokazuje przykład prostego nałożenia czerwonej, kompletnie nieprzejrzystej warstwy **P**, na niebieskie tło **t**. Rysunek 11.2.3 przedstawia tę samą sytuację, ale z częściową przejrzystością warstwy **P**. Jak programy obliczają kolor złożenia tych dwóch warstw (obszaru rezultatu — **c**)?



Rysunek 11.2.2 Złożenie podłoża **t** i warstwy **P** w trybie **Mix/Normal**. **P** zupełnie nieprzejrzysta



Rysunek 11.2.3 Złożenie podłoża **t** i warstwy **P** w trybie **Mix/Normal**. **P** o nieprzejrzystości = 0.4

Przyjrzyjmy się dokładniej przypadkowi, który pokazuje Rysunek 11.2.3. W metodzie **Mix/Normal** składniki **R, G, B** barwy wynikowej **c** są wyznaczone jako tzw. „średnia ważona” składników barw **t** i **P**. „Wagą” tej średniej jest nieprzejrzystość **A** warstwy leżącej „na wierzchu” — **P**<sup>1</sup>:

$$c = (1 - A)t + AP$$

gdzie: **c** — rezultat nałożenia barw (por. Rysunek 11.2.3);

**t** — barwa tła (por. Rysunek 11.2.3);

**P** — barwa warstwy nakładanej (por. Rysunek 11.2.3);

**A** — nieprzejrzystość (składnik **Alfa**) warstwy **P**;

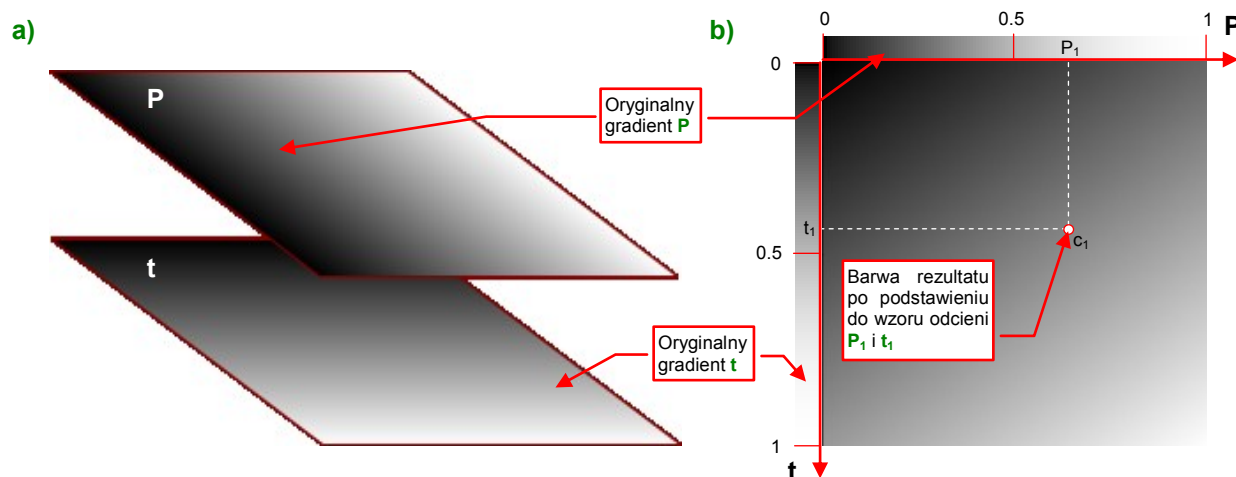
**Wyr. 11-1 Wyznaczanie rezultatu „zwykłego” mieszania barw (operacja nazywana **Mix** lub **Normal**)**

Jak „działa” Wyr. 11-1? Podstawmy do niego takie wartości, jakie podaje Rysunek 11.2.3. Na niebieskie ( $R = 0$ ,  $G = 0$ ,  $B = 1$ ) tło **t** została nałożona półprzezroczysta ( $A = 0.4$ ) warstwa **P**, w kolorze czerwonym ( $R = 1$ ,  $G = 0$ ,  $B = 0$ ). Komponenty  $R$ ,  $G$ ,  $B$  barwy **c** wyznaczamy, podstawiając do Wyr. 11-1 odpowiednie komponenty barw **t** i **P** (Rysunek 11.2.4):

<b>R:</b>	$(1.0 - 0.4)$	<b>0.0</b>	+	$0.4$	<b>1.0</b>	=	<b>0.4</b>
<b>G:</b>	$(1.0 - 0.4)$	<b>0.0</b>	+	$0.4$	<b>0.0</b>	=	<b>0.0</b>
<b>B:</b>	$(1.0 - 0.4)$	<b>1.0</b>	+	$0.4$	<b>0.0</b>	=	<b>0.6</b>
sumarycznie:	$(1.0 - 0.4)$	<b>t</b>	+	$0.4$	<b>P</b>	=	<b>c</b>

**Rysunek 11.2.4 Wyznaczenie nałożenia barw warstwy **t** i **P** ( $A = 0.4$  — por. Rysunek 11.2.3) w trybie **Mix/Normal**.**

Rysunek 11.2.4 pokazuje, że każdy z komponentów barwy można traktować jako intensywność — czyli odcień szarości. Komponenty rezultatu **c** są wyznaczone niezależnie, poprzez podstawienie do odpowiedniego wzoru. Dlatego w dalszej części tej sekcji każdą z operacji na barwach omówię, przedstawiając jej ogólny wzór (taki, jak ten z Wyr. 11-1), oraz rezultat — w postaci złożenia dwóch prostopadłych gradientów szarości (Rysunek 11.2.5a):



**Rysunek 11.2.5 Sposób prezentacji wyników nałożenia dwóch warstw, z różnymi kierunkami gradientu szarości.**

Rezultat nałożenia — z odpowiednim operatorem — warstwy **P** na **t** (Rysunek 11.2.5a) będą pokazywać na schematycznym kwadracie (Rysunek 11.2.5b). Możesz na nim zobaczyć wynik połączenia każdego odcienia

<sup>1</sup> Uwaga: wydaje się, że w Blenderze wartość **A** jest wcześniej mnożona przez parametr: **Col** z panelu **Map To** (dla mapowania barw) lub **Var** dla pozostałych wartości (**Ref**, **Spec**).

szarości **P** z każdym odcieniem szarości tła **t**. Z góry i z lewej strony kwadratu są pokazane oryginalne kolory pikseli z warstw **P** i **t**. To jakby osie układu współrzędnych, z których możesz wybrać interesujący Cię odcień pikseli wejściowych. Na przecięciu prostopadłych do osi **P** i **t** widzisz barwę rezultatu operacji (por. Rysunek 11.2.5b).

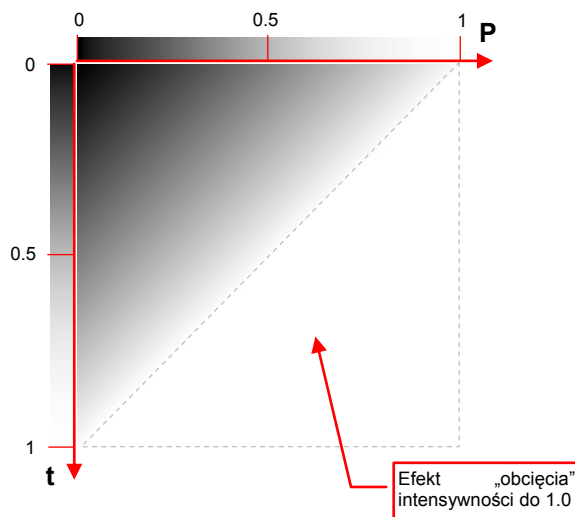
Domyślną operację — **Mix / Normal** — już omówiliśmy. Rysunek 11.2.6 przedstawia operację dodawania:

**Dodawanie:**

$$c = \text{Clip}(t + AP)$$

gdzie: **c** — rezultat nałożenia barw;  
**t** — barwa tła;  
**P** — barwa warstwy nakładanej;  
**A** — nieprzejrzystość (składnik **Alfa**) warstwy **P**;

**Clip**: funkcja „obcinająca” do 1.0 wartości  $> 1.0$ .



Wyr. 11-2 Dodawanie wartości pikseli (**Add** lub **Addition**)

Rysunek 11.2.6 Przykład dodawania pikseli

Dodawanie wartości pikseli rozjaśnia obraz. Charakterystyczne dla dodawania jest „obcinanie” rezultatu do 1.0. (Nie ma nic „bielszego od bieli”). Tę operację zapisaliśmy w Wyr. 11-2 jako funkcję **Clip**. Rysunek 11.2.6 pokazuje efekt jej „załączenia” — pół testowego kwadratu jest biała. To obszar, gdzie suma składników barwy **t** i **P** była  $> 1.0$ .

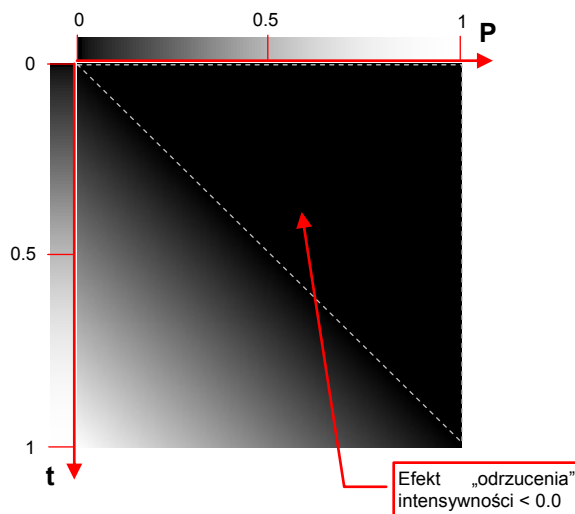
A teraz odwrotność dodawania: — **Subtract**. Rysunek 11.2.7 przedstawia rezultaty tej operacji:

**Odejmowanie:**

$$c = \text{Clip}(t - AP)$$

gdzie: **c** — rezultat nałożenia barw;  
**t** — barwa tła;  
**P** — barwa warstwy nakładanej;  
**A** — nieprzejrzystość (składnik **Alfa**) warstwy **P**;

**Clip**: funkcja „obcinająca” do 0.0 wartości  $< 0.0$ .



Wyr. 11-3 Odejmowanie wartości pikseli (**Subtract** lub **Subtraction**)

Rysunek 11.2.7 Przykład odejmowania pikseli

Odejmowanie wartości pikseli ściemnia obraz. Charakterystyczne dla odejmowania jest „odrzućcie” rezultatów  $< 0$ . (Nie ma nic „czarniejszego od czerni”). Tę operację zapisaliśmy w Wyr. 11-3 jako funkcję **Clip**. Rysunek 11.2.7 pokazuje efekt jej „załączenia” — pół testowego kwadratu jest całkowicie czarne. To obszar, gdzie różnica składników barwy **t** i **P** była  $< 0$ .

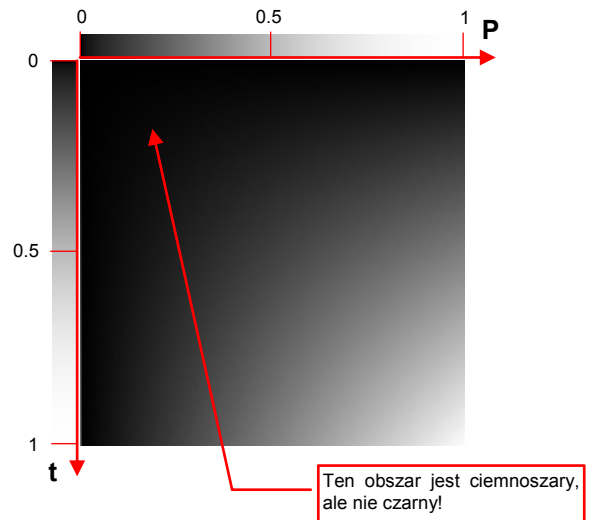


Kolejną typową operacją jest mnożenie — **Multiply**. Rysunek 11.2.8 przedstawia rezultaty tej operacji:

Mnożenie:

$$c = tAP$$

gdzie: **c** — rezultat nałożenia barw;  
**t** — barwa tła;  
**P** — barwa warstwy nakładanej;  
**A** — nieprzejrzystość (składnik **Alfa**) warstwy **P**;



Wyr. 11-4 Mnożenie wartości pikseli (**Multiply**)

Rysunek 11.2.8 Przykład mnożenia pikseli

Mnożenie wartości pikseli ściemnia obraz (gdyż mnożysz przez wartości poniżej 1.0). W odróżnieniu od odejmowania, żadna część testowego kwadratu, który przedstawia Rysunek 11.2.7, nie jest czarna. To są obszary zabarwione na bardzo ciemny kolor szary.

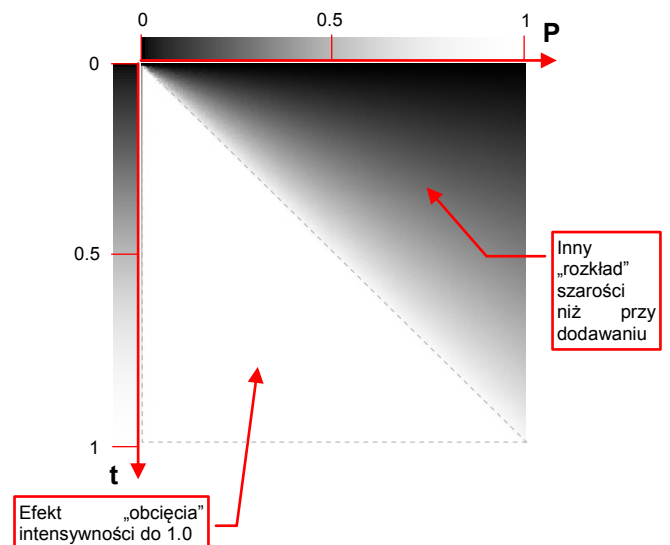
Kolejną typową operacją jest dzielenie — **Divide**. Rysunek 11.2.9 przedstawia rezultaty tej operacji:

Dzielenie:

$$c = \text{Clip}\left(\frac{t}{AP}\right)$$

gdzie: **c** — rezultat nałożenia barw;  
**t** — barwa tła;  
**P** — barwa warstwy nakładanej;  
**A** — nieprzejrzystość (składnik **Alfa**) warstwy **P**;

**Clip**: funkcja „obcinająca” do 1.0 wartości  $> 1.0$ , w tym także nieskończoność (gdy  $AP = 0$ )



Wyr. 11-5 Dzielenie wartości pikseli (**Divide**)

Rysunek 11.2.9 Przykład dzielenia pikseli

Dzielenie wartości pikseli rozjaśnia obraz (gdyż dzielisz przez wartości poniżej 1.0). Charakterystyczne dla dzielenia jest „obcinanie” rezultatu do 1.0. (W tym także nieskończoności, występującej dla  $A = 0$  lub  $P = 0$ ). Tę operację zapisaliśmy w Wyr. 11-5 jako funkcję **Clip**. Rysunek 11.2.9 pokazuje efekt jej „załączenia” — pół testowego kwadratu jest biała. To obszar, gdzie suma składników barwy **t** i **P** była  $> 1.0$ . Zwróć uwagę, że rozkład szarości na przeciwległej połowie jest inny, niż w przypadku dodawania (por. Rysunek 11.2.6)

Kolejną typową operacją jest przyciemnienie — **Darken**. Rysunek 11.2.10 przedstawia rezultaty tej operacji:

Zaciemnienie:

$$c = \text{Min}(t, AP)$$

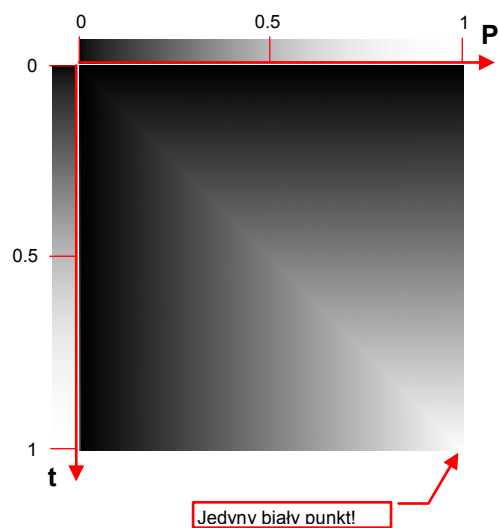
gdzie: **c** — rezultat nałożenia barw;

**t** — barwa tła;

**P** — barwa warstwy nakładanej;

**A** — nieprzejrzystość (składnik **Alfa**) warstwy **P**;

**Min**: funkcja, zwracająca wartość mniejszego z dwóch argumentów



Wyr. 11-6 Ściemnianie wartości pikseli (**Darken**)

Rysunek 11.2.10 Przykład ściemniania pikseli

Ściemnianie obrazu jest operacją dość intuicyjną. Z każdej pary pikseli (jeden należy do tła, drugi — do nakładanej warstwy) wybiera ten ciemniejszy.

Operacją odwrotną do ściemniania jest rozjaśnienie — **Lighten**. Rysunek 11.2.11 przedstawia rezultaty tej operacji:

Rozjaśnienie:

$$c = \text{Max}(t, AP)$$

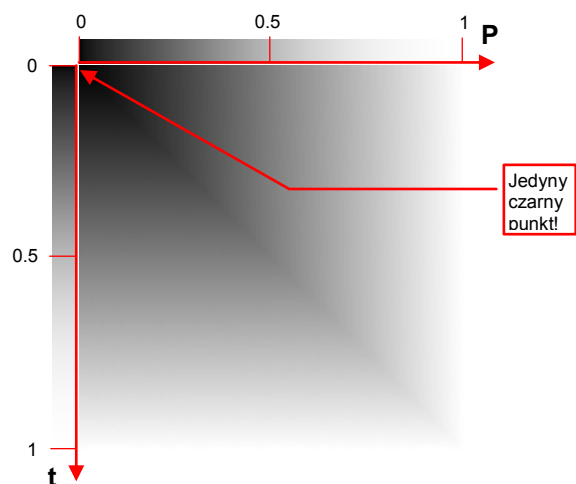
gdzie: **c** — rezultat nałożenia barw;

**t** — barwa tła;

**P** — barwa warstwy nakładanej;

**A** — nieprzejrzystość (składnik **Alfa**) warstwy **P**;

**Max**: funkcja, zwracająca wartość większego z dwóch argumentów



Wyr. 11-7 Rozjaśnianie wartości pikseli (**Lighten**)

Rysunek 11.2.11 Przykład rozjaśniania pikseli

Rozjaśnianie obrazu jest operacją dość intuicyjną. Z każdej pary pikseli (jeden należy do tła, drugi — do nakładanej warstwy) wybiera ten jaśniejszy.

Na listach rozwijalnych Blendera i Gimp'a znajdziesz więcej operacji, niż omówiłem w tym dodatku. Mówiąc szczerze, nie miałem okazji ich do tej pory używać. Jeżeli jednak chciałbyś sprawdzić, jak każda z nich działa — możesz skorzystać z pliku [optional/pixel\\_operations.xcf](#) (por. str. 16). To obraz Gimp'a, zawierający dwie warstwy, odpowiadające gradientom **t** i **P**. Wykorzystywałem go do wygenerowania ilustracji dla tej sekcji. Możesz go otworzyć i wybrać w Gimpie, z zakładki **Layers** jakiś inny tryb mieszania barw. Sam się wtedy przekonasz, jak działa.

## Skorowidz

- Nazwy poleceń, skróty klawiatURY, podane w skorowidzu, dotyczą Blendera, o ile w haśle nie jest zaznaczone inaczej ("GIMP)", "(Inkscape)").

### 3

#### 3D Cursor

- jako punkt odniesienia, 239, 269
- wpisywanie współrzędnych, 193
- wybór z listy *Pivot*, 239, 269

#### 3D Cursor Location

- sekcja z panelu Properties
- View, 193

#### 3D View

- dopasowanie rozmiaru obrazu tła, 193
- porównanie metod wyświetlania planów, 196
- problem z małą rozdzielczością obrazu tła, 195
- przejrzystość obrazu tła, 192, 198
- ustawianie obrazu tła, 191
- ustawienie tła w widoku z kamery, 129, 198
- ustawienie tła w wielu oknach, 195
- zmiana obrazu po wysunięciu przybornika, 185

### A

#### A

- składnik barwy (nieprzejrzystość), 468

#### Active Node

- panel z przybornika Properties, 211

#### Active Spline

- Cyclic*
- krzywa zamknięta/otwarta, 264

#### Order

- rząd krzywej, 264

#### Resolution

- liczba wierzchołków na segment, 264

#### Add

- sposób nakładania barw, 470

#### Add Constraint

- polecenie z zestawu Constraints, 237, 241, 243, 249, 254

#### Add Constraint (with Target)

- polecenie z menu Object, 235

#### Add Empty Parent

- add-on, 247

#### Add Image

- przycisk z panelu Background Images, 191

#### Add Modifier

- polecenie z panelu Modifiers, 285, 302, 309

#### Add Object Menu

- submenu ze Spacebar Menu, 218, 260

#### Add to Selection

- polecenie z menu kontekstowego Channels (GIMP), 162

#### Addition

- sposób nakładania barw, 470

#### Add-Ons

- sekcja okna User Preferences, 183, 201

#### Align View to Selected

- polecenie z menu View, 226

#### Alpha

- problemy z przejrzystym tłem renderu, 443

#### Alt-A, Patrz Play Animation

#### Alt-D, Patrz Duplicate Linked

#### Alt-H, Patrz Show Hidden

#### Alt-T, Patrz Clear Track

#### amortyzator

- cylinder, 64
- łączniki, 65
- kinematyka, 67
- okucia, 65
- tłok, 63
- ugięcie, 67

#### Angle

- pole z zakładki Rotate (Inkscape), 174

#### Angle Based

- opcja polecenia Unwrap, 276

#### animacja

- edycja linii ruchu (F-Curve), 427
- niespodziewane przesunięcia krzywych, 264
- przykład (śmigło), 425
- skopiowanie linii ruchu (F-Curve), 213
- śmigła, 130
- tryby ekstrapolacji linii ruchu, 427
- tworzenie nowej linii ruchu (F-Curve), 425
- układ ekranu dla, 425
- usunięcie linii ruchu (F-Curve), 427
- wybór linii ruchu (F-Curve), 427
- wybór współrzędnych do zmiany, 425
- wybranych obiektów sceny, 425

#### Animation

układ ekranu, 425

*Append*

przykład użycia, 103

*Apply*

przycisk z panelu Modifiers, 309

przycisk z panelu Transform (Inkscape), 174

armatura

dodanie do modelu, 229

dodanie nowej kości, 230

edycja kości, 229

kinematyka odwrotna, 231

pozycja spoczynkowa, 231

problemy z kierunkiem kości (przykład), 51

problemy z kinematyką odwrotną, 234

przykład zastosowania, 51

tryb wyświetlania, 231

wprowadzenie, 229

zaznaczanie kości, 229

zaznaczanie pojedynczej kości, 232

*Armature*

wprowadzenie, 229

*Array*

modyfikator siatki, 297

przykład użycia, 110

*Assign*

przycisk z zestawu Material, 275

*Attribute*

*Fac*

współczynnik, 381

*Name*

nazwa mapy UV, 381

węzeł wejściowy (Cycles), 381

*Auto IK*

nieoczekiwane rezultaty, 234

opcja przybornika (Pose Mode), 231

*Auto Save Temporary Files*

przełącznik z sekcji File, 189

autozapis

ustawienia, 184, 189

*Axis*

lista z panelu Background Images, 191

## B

*B*

składnik barwy, 468

*Backdrop*

opcja Node Editor, 435

podgląd rezultatu kompozycji, 435

przesuwanie obrazu, 435

przykład zastosowania, 435

*Background*

kontrolka z panelu Scene, 214

shader, 327, 337, 347

*Strength*

intensywność shadera, 337

*Background Images*

panel przybornika Properties, 129, 191

barwa

dokładność odwzorowania, 350

przykład łączenia (mieszania), 469

składniki, 350

średnia ważona, 469

Bates, Hume

inne materiały, 25

*Bevel*

*Limit Method*

metoda wyboru krawędzi, 295, 296

modyfikator a rozwinięcie siatki w UV, 295

modyfikator siatki, 295

*Offset*

parametr polecenia, 294

polecenie z menu Mesh, 294

*Segments*

parametr polecenia, 294

*Vertex only*

parametr polecenia, 294

*Width*

szerokość fazki, 295

*Bezier*

polecenie z menu Add (Curve), 263

*Bezier Circle*

polecenie z menu Add (Curve), 263, 264

Blender

rozszerzenia funkcjonalności, 183, 201

*Blending Mode*

sposób nakładania barw, 468

blok danych, *Patrz* datablock

linii pomocniczych, 423

wymuszenie usunięcia z pliku \*.blend, 125

*Blur*

odmiana węzła kompozycji, 444

*Blur Radius*

parametr z menu Gaussian Blur (GIMP), 163

boleć

przykład odwzorowania, 47

*Bone*

opcja z menu Set Parent To, 232

wyjaśnienie pojęcia, 229



zestaw z okna Properties, 230

### *Boolean*

modyfikator siatki, 285

modyfikator, problemy z, 39, 46

modyfikator, przykład zastosowania, 39

### *bounce*

odbicie śledzonego promienia, 324

### *Bounces*

kontrolka w panelu Integrator zestawu Render, 324

max. liczba śledzonych odbić (Cycles), 324

### *Bright/Contrast*

węzeł modyfikacji barwy, 404

### *Brush*

#### *Blend*

operacja przypisana do pędzla w trybie Weight Paint, 292

panel z przybornika Weight Paint, 292

#### *Radius*

rozmiar pędzla w trybie Weight Paint, 292

#### *Strength*

intensywność pędzla w trybie Weight Paint, 292

#### *Weight*

ustalenie wagi pędzla w trybie Weight Paint, 292

### *Bump*

konwersja mapy szarości na mapę normalnych, 404

przykład użycia, 404

węzeł (konwersji), 404

### *By Color*

polecenie z menu Select (GIMP), 158

## C

### celownik

odbłaskowy, elementy, 92

projektor, 92

przed kabiną pilota, 121

reflektor, 92

*channel*, Patrz barwa, składniki

### *Channel to Selection*

polecenie z menu kontekstowego Channels (GIMP), 162

### *Channels*

zakładka przybornika (GIMP), 162

### *Children*

#### *Effects*

sekcja panelu, 455

panel z zestawu Particles, 455

polecenie z menu Select, 76

### *chips.png*

zastosowanie w materiale B.Skin.Camouflage, 126

### chłodnica

cieczy, formowanie, 111

### cieniowanie

wybór trybu, 92

### cień

#### podłoże

przygotowanie, 145

sterowanie jasnością, 148

ustawienia materiału, 147

#### powierzchnia podłoża

przygotowanie, 146

przykład sterowania przejrzystością, 326, 361

sterowanie rozproszeniem granicy, 135, 144

uproszczony efekt dla szkła, 361

### Circle Fuzzy

rodzaj pędzla (GIMP), 166

### *Clamp*

dobór max. jasności promieni, 365

opcja z panelu Integrator zestawu Render, 365

### *Clear Constraints*

polecenie z menu Object, 236

### *Clear Deformer*

polecenie z menu Specials, 313

### *Clear Parent*

problemy z ograniczeniami, 248

### *Clear Seam*

polecenie z menu Edge Specials, 280

polecenie z menu Mesh, 280

### *Clear Sharp*

polecenie z menu Mesh, 306

### *Clear Track*

usunięcie ograniczeń Track To, Locked Track, 236

### *Clear Track and Keep Transform*

opcja przy usunięciu ograniczenia, 236

### *Color Balance*

odmiana węzła kompozycji, 134, 439

### *color button*

wybór barwy, 204

### *Color Management*

opcja z panelu Render

Shading, 205, 206

### *Color quantization*

metoda wektoryzacji bitmap (Inkscape), 176

### *Color Ramp*

#### *Add*

dodanie węzła, 387

#### *B-Spline*

przykład użycia, 402

#### *Constant*

- przykład użycia, 398, 403
  - Delete*
    - usuwanie węzła, 387
  - F*
    - inwersja paska barw, 387
  - interpolacja barw
    - Constant, 387
  - pasek barw
    - dodanie węzła, 387
    - inwersja, 387
    - usuwanie węzła, 387
    - węzły, 386
    - wybór aktywnego węzła, 387
  - polecenie z menu Convertor, 386
  - przykład użycia, 390
  - węzeł spektrum barw, 386
  - Composite*
    - Nodes*
      - odmiana węzłów do kompozycji renderu, 432
      - odmiana węzła kompozycji, 432
  - Composite Nodes*
    - przełącznik w oknie Node Editor, 432
  - Compositing*
    - kontrolka z panelu Post Processing, 434
  - config
    - folder konfiguracji Blendera, 187
  - Conformal*
    - opcja polecenia Unwrap, 276
  - constraint, *Patrz* ograniczenie
  - Constraints*
    - zestaw paneli z okna Properties, 237, 241, 243, 249, 254
  - Copy Location*
    - ograniczenie, 58
    - przykład użycia, 58
  - Copy Material*
    - polecenie z zestawu Material, 274
  - Copy Material to Others*
    - polecenie z zestawu Material, 273
  - Copy Settings*
    - opcja z menu New Scene, 203
  - Corp*
    - Down*
      - odległość od dolnej krawędzi obrazu, 449
    - Left*
      - odległość od lewej krawędzi obrazu, 449
      - odcięcie części obrazu, 139
      - odmiana węzła kompozycji, 139
    - Right*
      - odległość od prawej krawędzi obrazu, 449
    - Up*
      - odległość od górnej krawędzi obrazu, 449
  - Count*
    - pole z panelu Modifiers, 297
  - Crop*
    - odmiana węzła kompozycji, 449
  - Crp*
    - odcięcie części obrazu, 449
  - Ctrl-E, *Patrz* Edges**
  - Ctrl-G, *Patrz* New Group**
  - Ctrl-I, *Patrz* Inverse**
  - Ctrl-J, *Patrz* Join in new Frame**
  - Ctrl-L, *Patrz* Make Links**
  - Ctrl-M, *Patrz* Mirror**
  - Ctrl-T, *Patrz* Track To, Locked Track**
  - Cursor to Selected*
    - polecenie z menu Snap, 239, 269, 271
  - Curve*
    - modyfikator siatki, 288, 292
    - polecenie z menu Add, 260
    - przykład użycia, 97, 109
    - wpływ skrócenia krzywej, 290
  - Curve Deform*
    - polecenie z menu Set Parent To, 288
  - Cycles Hair Rendering*
    - panel zestawu Particles, 455
  - Cycles Hair Setings*
    - panel zestawu Particles, 455
  - czcionki ekranu
    - ustawienia, 184
- D**
- Darken*
    - sposób nakładania barw, 472
  - Data*
    - opcja panelu z zestawu Material, 227
  - datablock*
    - licznik użycia, 463
    - nie używane, 464
    - przykłady, 465
    - w rysunku Blendera, 462
    - zabezpieczone, 464
    - zestaw kontrolerek, 463
  - Default*
    - układ ekranu, 199
  - Default Value*
    - pole z panelu Interface, 319
    - wartość domyślna parametru grupy, 319

## deformacja

- krawędzi wokół otworu powierzchni podziałowej (eliminacja), 43
- siatek za pomocą tekstury, 300
- siatką
  - wielu obiektów naraz, 27, 312
  - za pomocą uproszczonej siatki, 27, 309

*Deformer**Apply*

- przycisk ‘utrwalający’ deformację, 315

*Bind objects*

- przycisk aktywujący, 313
- panel z przybornika Properties, 313

*Remove*

- przycisk usuwający deformację, 315

*Unbind objects*

- przycisk deaktywujący, 315

*Delete*

- polecenie z menu Curve, 261
- polecenie z menu Curve (F-Curve Editor), 427

## detale

- bolec, 47
- sprężyna, 46
- śruba, 47

*Difference*

- pole z panelu Modifiers, 285
- rodzaj operacji Boolean z panelu Modifiers, 285

*Direction*

- pole z panelu Modifiers (Displace), 301

*Directional Blur**Angle*

- kierunek rozmycia, 448

*Center*

- środek deformacji, 450

*Distance*

- intensywność deformacji, 446
- omówienie parametru, 447
- dostępne kształty rozmycia, 446

*Iterations*

- omówienie parametru, 447
- kierunkowe rozmycie obrazu, 139, 446
- odmiana węzła kompozycji, 139, 446
- problemy przy rozmyciu koncentrycznym, 451
- rozmycie koncentryczne, 449
- rozmycie liniowe, 446
- wykorzystanie w sposób pośredni, 140, 451

*Zoom*

- intensywność deformacji, 450

*Displace*

## modyfikator siatki, 300

- neutralna waga deformacji, 301
- przypisanie wierzchołków siatki, 301
- skala deformacji, 301
- tworzenie nowej tekstury (mapy deformacji), 300
- ustalanie kierunku deformacji, 301
- ustalanie sposobu mapowania tekstury, 301

*Display**Axis*

- wyświetlanie osi obiektu, 228

*Name*

- wyświetlanie nazwy obiektu, 228

## panel zestawu Object, 228

*Textured*

- opcja wyświetlania obiektu, 228

*Type*

- opcje rysowania obiektu, 228

*Wire*

- opcja wyświetlania obiektu, 228

*Divide*

- sposób nakładania barw, 471

*Division*

- sposób nakładania barw, 471

## dodanie

- modyfikatora (siatki), 285, 297, 300, 302, 309

## dodatki (add-ons)

- instalacja, 188, 201
- ustawienia aktywnych, 183
- usuwanie, 202

## dodawanie

- nowego obiektu, 260
- nowego układu ekranu, 200
- nowej kości, 230
- nowej sceny, 137, 203, 213
- nowej tekstury rastrowej (Image Texture), 334

## drażek

- sterowy, w kabinie, 95

*Duplicate*

- polecenie z menu Object, 219, 220

*Duplicate Linked*

- polecenie z menu Object, 219

## dural

- wykorzystanie shadera (grupy) Gloss Paint, 368

*Dynamic Spacebar Menu*

- programowe rozszerzenie Blendera (add-on), 183

**E***Edge Angle*

- opcja modyfikatora Edge Split, 305

*Edge Sharp*

opcja modyfikatora Edge Split, 306

*Edge Split*

dobór kąta granicznego, 92  
modyfikator obiektu, 305

*Edges*

podręczne menu edycji siatki, 280  
submenu menu Mesh, 306

*Edit Group*

polecenie z menu Node, 318

*Edit Mode*

odmiana trybu dla armatury, 229

*Editing*

sekcja okna User Preferences, 181

## edycja

linii ruchu (F-Curve), 427  
tryb pracy, włączanie, 261

## efekty

odpryski (złuszczenia) farby (GIMP), 164  
odpryski i przetarcia farby (GIMP), 164  
przetarcia — obwódki z barwy podkładowej, 172  
przetarcia — problem w zbliżeniach, 170  
przetarcia — ulepszone rozwiązanie, 126, 170, 171,  
172  
rozmycie obrazu (GIMP), 163  
rysy wzdłuż zdejmowanych paneli (GIMP), 169  
wykorzystanie tekstury do podziału materiałów, 170,  
171

## ekran

przełączenie na inny układ, 199, 425  
stworzenie nowego układu, 200  
układ  
Animation, 425  
Default, 199  
UV Editing, 199  
układ do animacji, 425  
układ okien, zarządzanie, 199  
usunięcie jednego z układów, 200  
zmiana nazwy układu, 200

*Emission**Number*

liczba części, 454  
pole panelu, 454, 455

*Empty*

obiekt, zastosowanie, 196  
polecenie z menu Add, 218  
pomocniczy obiekt w transformacji obrotu, 247  
przykład użycia polecenia z menu Add, 234  
utworzenie pustego obiektu, 218

*Emulate Numpad*

przełącznik z sekcji User Preferences  
Input, 182

*Environment Texture*

a modelowanie siatki sfery otoczenia, 339  
brak ostrości obrazu, 338

*Color*

automatyczna korekta gamma barw obrazu, 337  
piksele tekstury, 337  
mapowanie obrazu, 336

*Non-Color Data*

oryginalne intensywności obrazu, 337  
panorama otoczenia sceny, 336  
podgląd na scenie, 331  
polecenie z menu Add, 336  
porównanie z Image Texture, 336  
przykładowe wykorzystanie, 337

*Vector*

współrzędne tekstury, 337  
węzeł (obraz rastrowy), 336  
złożenie dwóch obrazów, 352

*Erase*

menu opcji usuwania, 261

*Eraser*

polecenie z menu Tools (GIMP), 160

*Exclude*

kontrolka z panelu Layers, 433

*Extrapolation Mode*

submenu Curve (F-Curve Editor), 427

*Extrude*

polecenie z menu Armature, 230  
polecenie z menu Curve, 261

**F***Fac*

współczynnik węzła Filter, 436

*Fake user*

bloku danych, 464

## fazowanie

ustalanie szerokości, 294  
wybranych krawędzi siatki, 294, 295  
wybranych wierzchołków siatki, 294

*F-Curve*

przykład użycia, 130  
skopiowanie, 213  
stworzenie nowej, 425  
tryby ekstrapolacji, 427  
wyjaśnienie pojęcia, 425  
zależność od czasu przesłony (Shutter), 430



*F-Curve Editor*

- edycja lini ruchu, 427
- edytor linii ruchu (animacji), 425
- elementy okna, 425
- wybór linii ruchu, 427

*File*

- sekcja okna User Preferences, 184, 189

*File Browser*

- okno, 189, 201

*Filter*

- odmiana węzła kompozycji, 134, 435

## filtr

- w sekcji Add-Ons okna User Preferences, 202

## formowanie

- chłodnic cieczy, 111
- detali steru kierunku, 121, 122
- kadłub — korekta kształtu, 27
- kadłub — korekta kształtu za pomocą MDeform, 28
- kadłub — korygowanie powierzchni wewnętrznych, 29
- kadłub — owiewka kabiny, 36
- kadłub — pierścień klap chłodnicy, 112
- kadłub — podłużnice, 83
- kadłub — porównanie z rysunkami fabrycznymi, 25
- kadłub — sprawdzanie rezultatów MDeform, 29
- kadłub — wiatrochron kabiny, 36
- kadłub — wylot z chłodnicy, 113
- kolektora spalin, 109, 110
- kopiowanie kształtu za pomocą modyfikatora Shrinkwrap, 34, 35
- luf karabinów, 114
- podwozie — kółko ogonowe, 38
- podwozie główne, 62
- reflektor do lądowania, 115
- urki Pitota, 115
- światło pozycyjne, 119

## fotel

- pilota, 84, 86
- pilota, różnice w typie, 84

## fotografia

- otoczenia
  - fotomontaż, 143
  - gdzie znaleźć, 129, 143
  - użycie, 129

*frame*

- klatka animacji, 425
- kontrolka do zmiany klatki, 425

*Frame*

- sposób łączenia węzłów, 210

*Fresnel*

- przykład wykorzystania węzła, 362, 367

*Full Copy*

- opcja z menu New Scene, 203

**G***G*

- składnik barwy, 468

*Gain*

- współczynnik węzła Color Balance, 440

## gamma

- korekcja wyświetlanych kolorów, 205

*Gamma*

- współczynnik węzła Color Balance, 440

## Gauss

- metoda rozmycia obrazu (GIMP), 163

*Gaussian Blur*

- polecenie z menu Filters (GIMP), 163

*General Noise*

- pomocnicza grupa węzłów, 395
- tekstura ‘ogólnego szumu’, 395
- zastosowanie w materiale, 126
- zastosowanie w materiale powierzchni zewnętrznych, 395

*Geometry*

- węzeł wejściowy, 332

*Glass BSDF**IOR*

- parametr, 360

*Roughness*

- parametr, 360

## shader, 360

*Gloss Paint**Diffuse*

- wpływ na powierzchnie połyskliwe, 369
- wpływ na powierzchnie półmatowe, 370
- zależność od barwy Specular, 371

- polecenie z menu Add, 322

- przykłady zastosowania, 368

*Reflectivity*

- analiza wpływu na powierzchnię, 372
- dla powierzchni połyskliwych, 369

*Roughness*

- analiza wpływu na powierzchnię, 373, 374
- dla powierzchni matowych, 370
- związek z odbłaskiem słońca, 369

## shader, 322

*Specular*

- wpływ na powierzchnie połyskliwe, 369

- wpływ na powierzchnie półmatowe, 370
- zależność od barwy Diffuse, 371
- zmodyfikowana zawartość, 375
- Glossy BSDF*
  - shader, 362
- godła
  - w postaci wektorowej (Inkscape), 175
- Grab*
  - polecenie z menu Pose, 231
- Grab/Move*
  - polecenie z menu Curve, 261
  - polecenie z menu *Object*, 270
- Gradient Texture*
  - Color*
    - piksele tekstury, 340
  - Diagonal*
    - kształt gradientu, 340
  - Easing*
    - kształt gradientu, 340
  - Fac*
    - wartości tekstury, 340
  - Linear*
    - kształt gradientu, 340
  - polecenie z menu Texture, 340
  - Quadratic*
    - kształt gradientu, 340
  - Quadratic sphere*
    - kształt gradientu, 341
  - Radial*
    - kształt gradientu, 341
  - Spherical*
    - kształt gradientu, 341
  - stosowanie w charakterze współczynnika łączenia
    - obrazów, 352
  - tekstura proceduralna, 340
  - użycie do podziału otoczenia sceny, 341
  - Vector*
    - współrzędne tekstury, 340
    - węzeł tekstury, 340
- Graph Editor*
  - okno, 425
- Grease Pencil*
  - Drawing Settings*
    - wybór powierzchni, 422
  - linie pomocnicze, 421, 458
  - New Layer*
    - przycisk, 421
  - Opacity*
    - przełącznik, 422
- panel przybornika Properties, 421
- parametry linii, 422
- przypisywanie do obiektu, 423
- rozpoczęcie rysowania, 421
- sterowanie widocznością, 423
- Thickness*
  - przełącznik, 422
- Group Input*
  - węzeł (wejścia do grupy węzłów), 317, 318
  - wiele instancji węzła (przykład), 320, 321
- Group Output*
  - węzeł (wyjścia z grupy węzłów), 317, 318
- Grow*
  - polecenie z menu Select (GIMP), 159
- Grow Selection*
  - okno dialogowe (GIMP), 159
- grupa
  - dodawanie wierzchołków siatki do, 278
  - węzłów, 317
    - definiowanie wejść, 318
    - domyślna wartość parametru, 319
    - edycja, 318
    - otwieranie i zamykanie definicji, 318
    - podział wejść, 320, 321
    - sposób wyświetlania schematu, 321
    - usuwanie wejść, 319
    - wielokrotne użycie pojedynczego wejścia, 319
    - zakres wartości parametru, 319
    - zmiana nazwy, 318
    - zmiana nazwy wejścia/wyjścia, 320
  - węzły wejściowe, 317
  - węzły wyjściowe, 317
  - wierzchołków siatki, 278
  - wierzchołków siatki, nazwa, 278
  - właściwości, 319
- guma
  - bieżnik opony, 403
  - kolor podstawowy (Diffuse), 406
  - kurz i pył, 404, 405
  - maska bieznika opony, 402
  - schemat materiału (opony), 407
  - szorstkość (tekstura), 404
  - szorstkość i połyskliwość, 406
- gumka
  - narzędzie w GIMP, 160

## H

- H, *Patrz* Hide Selected
- odcień koloru, 204

- hamulec
    - tarcza koła głównego, 63
  - Handle Panel*
    - add-on, 265
    - Label*
      - przycisk panelu sterowania uchwytem, 268
    - odświeżenie paneli, 267
    - okno programu, 265
    - Pos*
      - kontrolka panelu sterowania uchwytem, 267
    - przykład przygotowania uchwytów, 81
    - przykład zastosowania, 81
    - Reset*
      - przycisk panelu sterowania uchwytem, 267
    - Show/Refresh*
      - przycisk, 267
    - włączenie, 266
  - Hard edge*
    - zaznaczanie krawędzi pędzla (GIMP), 165, 167
  - HDR*
    - format zapisu obrazów panoramicznych, 350
  - HDR(I)*
    - pliki obrazu, 336
  - HDRI*
    - obrazy otoczenia wysokiej jakości, 350
  - Hex*
    - tryb wyboru koloru, 205
  - Hide Selected*
    - polecenie z menu Object, 225
  - hierarchia
    - ustalanie dla obiektów, 288
    - współrzędne obiektu potomnego, 245, 246
  - HSV*
    - tryb wyboru koloru, 204
  - HTML notation*
    - heksadecymalna reprezentacja barwy, 205
  - Hue Saturatiuon Value*
    - odmiana węzła kompozycji, 138, 438
- I**
- I, Patrz Insert Keyframe**
  - Image*
    - odmiana węzła kompozycji, 437
    - panel z zakładki Textures (Displace), 300
    - tryb wyświetlania obiektów Empty, 196
  - Image Texture*
    - Color*
      - piksele tekstury, 334, 347
    - Linear*
      - korekta barw obrazu, 334
  - Open*
    - przycisk, 334
  - polecenie z menu Add, 334
  - porównanie z Environment Texture, 336
  - sRGB*
    - korekta barw obrazu, 334
  - tekstura (z obrazu rastrowego), 334
  - Vector*
    - współrzędne tekstury, 334
  - węzeł tekstury, 334
  - wykorzystanie w tle renderu, 347, 356
  - Influence*
    - sterowanie wpływem ograniczenia, 77
  - Input*
    - sekcja okna User Preferences, 182
  - Inputs*
    - lista z panelu Interface, 319
  - Insert Keyframe*
    - menu rozwijalne, 426
  - instalacja
    - dodatków, 188, 201
  - Install Add-On*
    - przycisk okna File Browser, 201
    - przycisk z sekcji Add-Ons okna User Preferences, 201
  - intensywność
    - składnika barwy, 468
    - tekstury, sterowanie, 404
  - Interface*
    - panel z przybornika Properties, 319
    - sekcja okna User Preferences, 181
  - Intersect with Selection*
    - polecenie z menu kontekstowego Channels (GIMP), 162
  - Intersection, Patrz przecięcie, powłok*
    - add-on, 221
    - okno dialogowe, 222
    - Use both meshes*
      - opcja polecenia, 222
    - Use diagonals*
      - opcja polecenia, 223
    - Use selected faces*
      - opcja polecenia, 222
    - zasada działania, 223
  - Inverse*
    - polecenie z menu Select, 279
- J**
- Jitter

rodzaj pędzla (GIMP), 164

*Join in new Frame*

polecenie z menu Node, 210

## K

kabina

celownik odblaskowy, elementy, 92

detale tablicy

przrzędów pokładowych, 91

dodatkowe materiały, 91

dopasowanie do figury pilota, 103

dopasowanie figury pilota, 104

drążek sterowy, 95

drugi celownik, 121

element główny figury pilota, 103

elementy podstawowe, 83

figura pilota, 102

figura pilota, inne poprawki, 104

formowanie podłużnic, 83

formowanie wiatrochronu, 36

fotel pilota, 84, 86

import figury pilota, 103

lewa burta, 94

mechanizm osłony kabiny, 96

modelowanie wskaźników, 89

napisy eksploatacyjne, 91

orczyk, 93

otwory w tablicy przrzędów, 88

pasy — końcówki, 98

plakietki z napisami, 91

plik z figurą pilota, 102

początek modelowania wnętrza, 83

podłoga, 86, 95

podstawowe materiały wnętrza, 83

podział na strefy, 93, 100

podział na warstwy, 106

pozowanie pilota, 132

prawa burta, 96

przewody elastyczne, 96

przygotowanie powłoki, 83

rozwiniecie wnętrza w UV, 84

różnice w typie foteli pilota, 84

sprzączki pasów, 97

sterowanie hydrauliką klap i podwozia, 95

szczegóły wiatrochronu, 92

tablica instalacji elektrycznej, 94

tablica przrzędów, 87

tablica przrzędów i otoczenie, 93

taktyka wykonania szczegółów, 93, 100

tekstura barwy powierzchni wewnętrznych, 85

tekstura nierówności powierzchni wewnętrznych, 85

tekstura odbić powierzchni wewnętrznych, 85

tekstura tarcz wskaźników, 90

tylna ściana, 97

uprząż pilota, 97, 105

weryfikacja szkieletu kadłuba, 83

kadłub

formowanie owiewki kabiny pilota, 36

formowanie pierścienia klap chłodnicy, 112

formowanie podłużnic, 83

formowanie wiatrochronu kabiny pilota, 36

formowanie wylotu z chłodnicy, 113

korekta kształtu (szerokości), 27

korekta kształtu za pomocą dodatku MDeform, 28

korygowanie kształtu powierzchni wewnętrznych, 29

porównanie z rysunkami fabrycznymi, 25

sprawdzanie rezultatów MDeform, 29

szczegóły połączenia ze skrzydłem, 26

wiatrochron kabiny, 92

wykorzystanie modyfikatora Shrinkwrap do

kopiowania kształtu, 34, 35

kamera

odległość pikseli obrazu od, 443

powiązanie z powierzchnią podłoża, 146, 150

pozowanie modelu, 129, 143

warstwa dla, 186

kamuflaż

wykorzystanie shadera (grupy) Gloss Paint, 370

kanal

barwy, *Patrz* składnik, RGB

chłodnicy, materiały, 113

karabin

formowanie lufy, 114

materiał lufy, 114

katalog

z konfiguracją Blendera, 187

kaustyka

dla światła kierunkowego Sun, 360

włączanie efektu, 370

wyłączanie efektu, 326, 358, 360

*Keep Above Surface*

pole z panelu Modifiers (Shrinkwrap), 302

*keyframe*

ustalenie położenia obiektu, 425

wyjaśnienie pojęcia, 425

kinematyka

elastycznego przewodu, 72

kinematyka odwrotna



- problemy, 234
- klapy
  - chłodnicy, mechanizm, 112
  - dźwignia sterowania, 95
  - skrzydła, mechanizm, 116
- klapy chłodnicy
  - sterowanie obrotem, 113
- klapy skrzydła
  - sterowanie obrotem, 118
- klatka
  - aktualna (w animacji), 425
  - animacji, 425
  - kluczowa, 425
    - ustalenie położenia obiektu, 425
    - wyjaśnienie pojęcia, 425
  - zmiana aktualnej, 425
- klawiatura
  - laptopa, 182
  - numeryczna, 182
- klonowanie
  - podwozia głównego, 76
  - poprawianie nazw po, 78
- kolektor
  - spalin, formowanie, 109, 110
  - spalin, materiał, 111
- kolor
  - heksadecymalne wartości R,G,B, 205
  - intensywność (V), 204
  - intensywność, składnika barwy, 468
  - kontrolka w Blenderze, 204
  - korekcja gamma, 205
  - Linear, 205
  - nasycenie (S), 204
  - nieprzejrzystość, składnik barwy, 468
  - odcień (H), 204
  - opcja Color Management, 205, 206
  - składniki barwy, 468
  - sposób zapisu, 468
  - sRGB, 205
  - ustalanie wartości HSV, 204
  - ustalanie wartości RGB, 204
  - wybór 'poprzez przykład', 204
- koło główne
  - barwy wnęki, 75
  - cylinder amortyzatora, 64
  - formowanie, 62
  - goleń, 63, 64
    - okucia osi obrotu, 70
  - hak holowniczy, 64
  - luk dla, 69
  - łączniki
    - kinematyka, 67
    - okucia, 65
  - łączniki amortyzatora, 65
  - nakrętki, 66
  - nierówności wnęki, 75
  - opona, 62
  - pokrywa boczna, 73, 80
    - popychacz, 73
  - pokrywa goleni, 73, 79
  - poprawianie nazw po powieleniu, 78
  - popychacze, 70
  - powielenie, 76
  - przewód hamulcowy, 71
    - kinematyka, 72
  - szkice wykonawcze, 62
  - śruby, 66
  - tarcza hamulca, 63
  - tłok amortyzatora, 63
  - ugięcie amortyzatora, 67
  - weryfikacja ruchu, 80
  - zębaki, 70
- kompozycja
  - najprostszy schemat, 432
  - obrazów rastrowych, 437
  - podgląd rezultatu, 435
  - rezultatu, 432
  - układ ekranu, 434
- kompozytor
  - włączenie efektów na renderze, 434
  - wykorzystanie masek, 444
  - wyodrębnienie obiektu, 136, 139, 445
  - zapisanie wyniku do pliku rastrowego, 437, 451
- kopiowanie
  - obiektu, 219
- korekta
  - szerokości kadłuba, 25
- kości
  - (armatury) — wyjaśnienie pojęcia, 229
  - edycja, 229
  - kinematyka odwrotna, 231
  - nazwy własne, 230
  - parametry sterujące, 230
  - pozycja spoczynkowa, 231
  - problemy z kinematyką odwrotną, 234
  - przypisanie jako 'rodzica' obiektu, 232
  - tryb wyświetlania, 231
  - utworzenie, 230

wytłoczenie nowej, 230  
 zaznaczanie, 229  
 zaznaczanie razem z obiektem, 232

#### kółko ogonowe

amortyzator, 43, 44, 53  
 amortyzator — okucie, 43  
 ciągną (kierujące), 46, 56  
 formowanie, 38  
 goleń, 38  
 materiał goleni, 39  
 materiał opony, 40  
 mechanizacja, 51  
 mechanizm chowania, 45, 53  
 okucie pokrywy, 48, 59  
 okucie wahacza, 42  
 opona, 40  
 piasta, 40  
 pokrywa — kopia robocza, 47, 57  
 popychacz, 53, 54  
 popychacze pokryw, 48, 59  
 szkice wykonawcze, 38  
 uchwyt (sterujący), 52, 80  
 wahacz, 41, 46, 52  
 wręga (z okuciami), 47  
 wypusty wahacza, 41

#### krawędzie

ostre, automatyczna korekta normalnych, 305  
 ostre, korygowanie kierunku normalnych, 305, 306  
 ostre, max. kąt automatycznej korekty, 305  
 ostre, minimalizowanie zaburzeń kierunku normalnych, 307  
 ostre, wybiórcza korekta normalnych, 306  
 ostre, zaburzenia interpolacji normalnych, 305  
 zastosowanie flagi Sharp, 306

#### krawędź

fazowanie, 294, 295  
 oznaczanie jako szwu UV, 280  
 przecięcia dwóch siatek, 221  
 szerokość fazki, 294  
 szwy UV, widoczność, 280  
 zaokrąglanie, 294

#### krzywa

deformacja poprzeczna, 290  
 edycja, 261  
 kierunek poprzeczny do, 262  
 krańce — położenie, 262  
 krańce — styczne, 262  
 liczba wierzchołków, 264  
 przesunięcia podczas animacji, 264

rząd krzywej, 262, 264  
 skręcenie, 98, 262, 290  
 utworzenie, 260, 263  
 waga punktu sterującego, 262  
 wielobok sterujący, 262  
 właściwości, 264  
 współrzędne punktu sterującego, 262  
 wyginanie wzdłuż, 97, 109, 288, 291  
 wyłączenie animacji, 264  
 wytłoczenie (rozbudowa), 261

## L

#### lakier

wykorzystanie shadera (grupy) Gloss Paint, 369

#### lampa

warstwa dla, 186

#### laptop

klawiatura, 182

#### Layer

kontrolka z panelu Layers, 433

#### Layers

panel z zestawu Render Layers, 433

#### lewa burta

kabiny, 94

#### licznik użycia

bloku danych Blendera, 463

#### Lift

współczynnik węzła Color Balance, 439

#### Light Path

##### Is Camera Ray

identyfikacja pikseli tła, 325  
 przykład zastosowania, 356

##### Is Diffuse Ray

identyfikacja pikseli odbicia rozproszonego, 325

##### Is Glossy Ray

identyfikacja pikseli odbicia kierunkowego, 325

##### Is Reflection Ray

identyfikacja pikseli pochodzących z odbicia promieni, 325  
 przykład zastosowania, 325

##### Is Shadow Ray

identyfikacja pikseli cienia, 325  
 przykład zastosowania, 326, 361

##### Is Singular Ray

identyfikacja pikseli bezpośredniego odbicia promieni ze źródła światła, 325

##### Is Transmission Ray

identyfikacja pikseli powierzchni przejrystej, 325  
 przykład zastosowania, 358

- węzeł wejściowy (Cycles), 324
  - Lighten*
    - sposób nakładania barw, 472
  - Limit Distance*
    - Clamp Region*
      - opcja On Surface, 254
      - rodzaj obszaru ‘dozwolonego’, 254
    - lista modyfikatorów, 254
    - przykład zastosowania, 67
    - Target*
      - nazwa ‘celu’ dla osi, 254
    - utworzenie, 254
  - Limit Location*
    - Convert*
      - wybór typu układu współrzędnych, 242
    - For Transform*
      - przełącznik, ograniczający transformacje, 242
    - lista modyfikatorów, 241
    - Local Space*
      - lokalny układ współrzędnych obiektu, 242
    - Maximum*
      - współrzędne końca zakresu, 242
    - Minimum*
      - współrzędne początku zakresu, 242
    - opis ograniczenia, 241
    - przykład użycia, 53, 55
    - utworzenie, 241
    - World Space*
      - globalny układ współrzędnych, 242
  - Linear*
    - tryb konwersji koloru, 205
  - Lines*
    - rodzaj pędzla (GIMP), 167
  - linia
    - pomocnicza, 421, 458
      - barwa, 422
      - nieprzejrzyistość, 422
      - parametry, 422
      - ponowne włączanie, 423
      - rozpoczęcie rysowania, 421
      - rysowanie, 422
      - szerokość, 422
      - ukrywanie, 423
      - usuwanie, 422
      - wybór powierzchni, 422
      - wybór trybu rysowania, 422
  - Link Material*
    - opcja panelu z zestawu Material, 227
  - Link Object Data*
    - opcja z menu New Scene, 203
  - Link Objects*
    - opcja z menu New Scene, 137, 203, 213
  - Location*
    - opcja z menu Insert Keyframe, 425, 426
  - Locked Track*
    - lista modyfikatorów, 237
  - Lock*
    - oś ‘zablokowana’ (obrotu), 238
    - określanie neutralnej pozycji ‘celu’, 238
    - opis ograniczenia, 237
    - przykład użycia, 52, 53, 234
    - Target*
      - nazwa ‘celu’ dla osi, 237
    - To*
      - oś śledząca cel, 238
      - przykład — kłapa skrzydła, 238
    - utworzenie, 237
  - lotki
    - mechanizm, 117
    - sterowanie obrotem, 118
  - lustrzane odbicie
    - obiektów, przykład, 257
    - siatki, 297
- L**
- łącze
    - materiału
      - aktywne, 274
      - dodanie nowego do siatki, 274
      - usunięcie z siatki, 274
      - ze ścianami siatki, 274
  - łączenie
    - barw, przykład, 469
    - węzłów, 207
- M**
- Make Links*
    - polecenie z menu Object, 274
  - Make Parent*
    - polecenie z menu Object, 288
  - Make Single User*
    - polecenie menu Object, 137, 213
  - Make Track*
    - menu podręczne, 235
  - malowanie
    - wag wierzchołków, 291, 458, 459
  - Manipulate center points*
    - przełącznik w oknie 3D View, 239, 269

- rezultat działania, 239, 270
- mapa deformacji
  - modyfikatora Displace, 300
- mapa normalnych
  - rodzaj tekstury, 374
- mapowanie
  - cylindryczne panoramy, 351
  - UV
    - aktywne, 283
    - domyślne dla renderowania, 283
    - ikona domyślnego, 283
    - różne rozwinięcia tej samej siatki, 282
    - stworzenie, 282
    - usunięcie, 282
    - zmiana nazwy, 282
- Mapping*
  - Location*
    - przesunięcie o zadany wektor, 332, 339, 342
  - Normal*
    - tryb działania węzła, 333
  - Point*
    - tryb działania węzła, 333
  - polecenie z menu Add, 332
  - przykład zastosowania węzła, 348
  - Rotation*
    - obróć o zadany wektor, 338
    - obróć panoramy otoczenia, 338
    - obróć wokół wskazanej osi, 342
  - Scale*
    - sterowanie rozmiarem, 353
    - zmiana skali w zadanym kierunku, 332
  - Texture*
    - tryb działania węzła, 333
  - transformacja współrzędnych, 332, 338, 341
  - Vector*
    - tryb działania węzła, 333
  - węzeł pomocniczy, 332, 338, 341
- Mark Seam*
  - polecenie z menu Edges, 280
  - polecenie z menu Mesh, 280
  - polecenie z menu UVs, 280
- Mark Sharp*
  - polecenie z menu Mesh, 306
- Material*
  - polecenie z menu Object, 274
  - problemy z wyświetlaniem schematu w panelu, 208
  - wyświetlanie w panelu kontrolek węzłów Cycles, 208
  - zestaw paneli okna Properties, 273
- material slot*
  - łącze materiału, 274
- materiał
  - barwa podłoża pod modelem, 147
  - barwy kurzu i pyłu, 405
  - ciemne, półmatowe powierzchnie, 371
  - cienie na powierzchniach matowych, 370
  - dodanie nowego łącza do siatki, 274
  - drobne zabrudzenia, 126
  - duraluminium, 368
  - farba półmatowa, 370
  - gładź tłoka, 396
  - grupowa zmiana jasności, 106
  - guma (opona), 407
  - guma, bieżnik opony, 403
  - guma, kolor Diffuse, 406
  - guma, kurz i pył, 404, 405
  - guma, mapa normalnych, 404
  - guma, maska bieżnika opony, 402
  - guma, szorstkość i połyskliwość, 406
  - kanal chłodnicy, 113
  - kolektora spalin, 111
  - komponent X.Leather.Band, 409
  - kuta stal, 399
  - lufy karabinu, 114
  - łącze aktywne na liście Material, 274
  - łącze ze ścianami siatki, 274
  - modulacja połyskliwości za pomocą tekstury, 394
  - ocynkowana stal, 401
  - odbicie otoczenia, 350
  - Opaque Surface, 406
  - pasów fotela pilota, 416
  - plastik (nieprzejrzysty), 418
  - plastik (przejrzysty), 418
  - podłoża pod modelem, 146
  - podstawowe barwy tkaniny, 416
  - podstawowe powłoki samolotu, 126
  - połyskliwy lakier, 369
  - powierzchni zewnętrznych, 394
  - przypisanie do fragmentu siatki, 274
  - przypisanie do obiektu, 227
  - przypisanie do siatki, 227, 273
  - przypisanie do wielu siatek, 273, 274
  - przypisanie różnych do indywidualnych ścian, 275
  - różnicowanie ze względu na obiekt, 138, 216
  - skopiowanie przypisania, 274
  - skóra — wpływ oświetlenia, 414
  - skóra (brązowa), fałdy i odbłaski, 412
  - skóra, dodanie śladów zużycia, 411
  - skóra, geometria fałd, 413



skóra, kompozycja tekstury, 410  
skóra, podstawowa tekstura, 408  
skóra, pofałdowana, 413  
skóra, tekstura nierówności, 409  
skóry, B.Black.Leather, 411  
skóry, B.Brown.Leather, 411  
stal, 396  
stalowej linki, 397  
stalowej sprężyny, 398  
sterowanie przejrzystością (otwory), 398  
szkło, 360  
światła pozycyjnego, 119  
tekstura włókien tkaniny, 415, 416  
tekstura X.Leather.Pattern, 410  
tkanina pasów (przygotowanie siatki), 415  
usunięcie łącza z siatki, 274  
użycie tekstury proceduralnej, 402  
wklejenie przypisania, 274  
wnetrza kabiny, detali, 91  
wnetrza kabiny, 83  
wykorzystanie tekstury X.Leather.Pattern, 410

#### *Math*

##### *Less Than*

użycie funkcji do tworzenia masek, 443

##### *Maximum*

przykład zastosowania, 358  
polecenie z menu Add, 358, 443  
reprezentacja operacji matematycznej, 358, 443

#### *Max*

pole z panelu Interface, 319  
zakres wartości parametru grupy, 319

#### *MDeform, Patrz deformacja siatką (wielu obiektów)*

a siatki współdzielone pomiędzy obiektami, 30, 315  
add-on, 27, 312  
ograniczanie wpływu do grupy wierzchołków, 30, 32  
panel sterująca (Deformer), 313  
przycisk ‘utrwalający’, 315  
przycisk aktywujący, 313  
przycisk deaktywujący, 315  
przycisk usuwający efekt, 315  
przypisywanie obiektu do deformatora, 28, 312  
usuwanie przypisywanie obiektu do deformatora, 313  
weryfikacja rezultatu, 29

#### *mechanizm*

klap chłodnicy, 112  
klap skrzydła, 116  
lotek, 117  
podwozie — kółko ogonowe, 51  
uchwyt, 52, 68, 80

#### *Mesh*

w Blenderze, 462

#### *Mesh Deform*

##### *Bind*

aktywizacja obiektu deformującego, 309, 310  
mankamenty modyfikatora, 27  
modyfikator siatki, 27, 309

##### *Object*

przypisanie obiektu deformującego, 309  
przygotowanie siatki pomocniczej, 27

##### *Unbind*

deaktywizacja obiektu deformującego, 309, 310

#### *Mesh Display*

panel z przybornika Properties, 280

##### *Seams*

opcja wyświetlania, 280

#### *Mesh Options*

##### *Edge Select Mode*

lista rozwijalna, 280, 296  
panel z przybornika Tool Shelf, 280, 296

##### *Tag Bevel*

tryb oznaczania szwów siatki, 296

##### *Tag Seam*

tryb oznaczania szwów siatki, 280

#### *Midlevel*

pole z panelu Modifiers (Displace), 301

#### *mieszanie*

barw, przykład, 469

#### *Min*

pole z panelu Interface, 319  
zakres wartości parametru grupy, 319

#### *Mirror*

lustrzane odbicie obiektów, 257  
modyfikator siatki, 297  
modyfikator, wybór innego punktu odniesienia, 297  
polecenie z menu Object, 257

#### *Mix*

aktualny tryb ‘mieszania’ w malowaniu wag, 292

##### *Color1*

barwa wejściowa, 357

##### *Color2*

barwa wejściowa, 357

##### *Fac*

parametr, 352, 357  
sterowanie udziałem łączonych barw, 352, 357  
odmiana węzła kompozycji, 448  
polecenie z menu Add, 352  
sposób nakładania barw, 468, 469  
tryb malowania wag siatki, 292

- węzeł do łączenia kolorów, 357
  - węzeł do łączenia kolorów (obrazów), 352
  - złożenie dwóch obrazów, 448
  - Mode*
    - sposób nakładania barw, 468
  - model
    - pozowanie, 129, 143
  - Moderate*
    - pomocniczy węzeł (grupy), 376
  - Modifiers*
    - sekcja okna Properties, 285, 297, 300, 302, 309
  - modyfikator
    - 'utrwalenie', 309
    - aktywizacja (Mesh Deform), 309, 310
    - Boolean, przykład zastosowania, 39
    - deaktywizacja (Mesh Deform), 309, 310
    - dodadnie, 285, 302, 309
    - obiekt deformujący, 309
    - ograniczenie obszaru (siatki) dla, 303
    - ograniczenie wpływu na siatkę, 292
    - zastosowanie w trakcie edycji, 289
  - modyfikatory
    - utworzenie, 285, 297, 300, 302, 309
  - Motion Blur*
    - panel z zestawu Render, 130, 429
  - Muegrave Texture*
    - węzeł tekstury, 345
  - Multiplication*
    - sposób nakładania barw, 471
  - Multiply*
    - sposób nakładania barw, 471
  - Musgrave Texture*
    - fBM*
      - typ tekstury, 345
    - Hetero Terrain*
      - typ tekstury, 346
    - Hibrid Multifractal*
      - typ tekstury, 346
    - Multifractal*
      - typ tekstury, 346
    - polecenie z menu Texture, 345
    - przykład użycia, 391
    - Rigged Multifractal*
      - typ tekstury, 346
    - tekstura proceduralna, 345
- N**
- nadir
    - wyjaśnienie pojęcia, 331
  - nakrętki
    - śrub, 66
  - napęd
    - sterowanie, 94
  - napisy eksploatacyjne
    - zwiększanie dokładności odwzorowania, 108
  - nazwa
    - grupy wierzchołków siatki, 278
    - uchwyty, 268
    - zmiana dla sceny, 203
    - zmiana dla układu ekranu, 200
  - Negate*
    - pomocniczy węzeł (grupy), 376
  - neutralne
    - położenie, uchwytu, 265
  - New*
    - opcja z menu New Scene, 203
    - polecenie z menu File, 180, 187
    - przycisk z zestawu Material, 273
    - przycisk zakładki Textures (Displace), 300
  - New Group*
    - polecenie z menu Node, 317
  - New Scene*
    - opcje tworzenia nowej sceny, 203
  - nie używane
    - bloki danych, 464
  - No Caustics*
    - opcja z panelu Integrator zestawu Render, 326, 358, 360, 370
  - node
    - color
      - węzeł Mix, 448
    - distort
      - węzeł Corp, 139
      - węzeł Crop, 449
    - filter
      - węzeł Blur, 444
      - węzeł Directional Blur, 139, 446
      - węzeł Sharpen, 134, 435
    - input
      - węzeł Image, 437
      - węzeł Render Layer, 432
    - kolory
      - węzeł Color Balance, 134, 439
      - węzeł Hue Saturation Value, 138, 438
    - output
      - węzeł Composite, 432
      - węzeł Viewer, 434
  - Node Editor*

- kopiowanie i wklejanie węzłów, 210
- okno, 207
- powiększanie zawartości, 208
- przesuwanie zawartości, 208
- tryb pracy, 210
- węzły, 207, 209
- Noise Texture*
  - Color*
    - piksele tekstury, 343
  - Detail*
    - szczegółowość szumu, 344
  - Distortion*
    - zaburzenia szumu, 344
  - Fac*
    - wartości tekstury, 343
  - polecenie z menu Texture, 343
  - przykład użycia, 390, 405
  - Scale*
    - rozmiar szumu, 343
  - tekstura proceduralna, 343
  - Vector*
    - współrzędne tekstury, 343
  - węzeł tekstury, 343
- Normal*
  - sposób nakładania barw, 468, 469
  - węzeł wejściowy shaderów, 374
- normalne
  - automatyczna korekta dla ostrych krawędzi, 305
  - korekta dla wybranych krawędzi, 306
  - korygowanie kierunku dla ostrych krawędzi, 305, 306
  - max. kąt automatycznej korekty, 305
  - minimalizowanie zaburzeń kierunku na ostrych krawędziach, 307
  - minimalizowanie zaburzeń za pomocą dodatkowych ścian siatki, 307
  - wpływ na siatkę, 304
  - zaburzenia interpolacji przy ostrych krawędziach, 305
  - zastosowanie flagi Sharp, 306
- nowy
  - shader, 317
- numeryczna
  - klawiatura (emulacja), 182
- Nurbs*
  - polecenie z menu Add (Curve), 263
- Nurbs Circle*
  - polecenie z menu Add (Curve), 263, 264
- 'utrwalenie' deformatora, 315
- aktywacja deformatora, 313
- deaktywacja deformatora, 315
- dodanie pomocniczego obiektu dla transformacji, 247
- dokładna transformacja (Inkscape), 174
- dopasowanie widoku do lokalnego układu współrzędnych, 226
- informacje dla renderera, 216
- jako deformator innych obiektów, 313
- kopiowanie, 219
- ograniczenia (constraints), 235, 237, 241, 243
- przypisanie do deformatora, 28, 312
- przypisanie do kości, 232
- pusty, przykład użycia, 234
- sterujący ruchomymi częściami, 265
- tryb wyświetlania, 228
- ujemna skala i modyfikator Stretch To, 251
- ukrywanie, 225
- usunięcie przypisań deformatora, 315
- usuwanie przypisania do deformatora, 313
- utworzenie, 260
- utworzenie (krzywej), 260, 263
- utworzenie pustego (Empty), 218
- utworzenie w aktualnym widoku, 181
- właściwości (krzywej), 264
- wyróżnienie dla renderera, 138, 215, 216
- zmiana hierarchii, 288
- zmiana położenia środka, 224
- Object*
  - opcja panelu z zestawu Material, 227
  - pole z panelu Modifiers, 297
  - w Blenderze, 462
- Object Data*
  - Curve*
    - zestaw z okna Properties, 264
  - UV Maps, usunięcie mapowania, 282
  - UV Maps, utworzenie nowego rozwinięcia, 282
- Object Index*
  - wyróżnienie obiektu dla renderera, 138, 215, 216
- Object Info*
  - węzeł wejściowy (Cycles), 216
- Object Mask*
  - pomocniczy węzeł (grupy), 136, 139, 445
  - przykład zastosowania, 136, 448
- Object Tools*
  - Draw*
    - tryb rysowania linii pomocniczych, 422
  - Erase*
    - tryb usuwania linii pomocniczych, 422

## O

obiekt, *Patrz* Object

- Grease Pencil*
  - sekcja, 422
- Line*
  - tryb rysowania linii pomocniczych, 423
- panel przybornika Tools, 422
- Poly*
  - tryb rysowania linii pomocniczych, 423
- obraz
  - brak ostrości panoramy, 338
  - dokładność odwzorowania barw, 350
  - efekt 'przepalenia bieli', 442
  - format zapisu panoram wysokiej jakości, 350
  - kompozycja panoramy dolnej półsfery, 151, 351
  - łączenie materiałów, 126
  - mapowanie panoramy, 336, 339
  - maska renderowanego obiektu, 136, 139, 445
  - monochromatyczny, zmiana odcienia, 441
  - nakładanie jednej tekstury na drugą, 379
  - obrót panoramy, 338, 355
  - odcięcie części, 139, 449
  - odległości od pikseli, 443
  - otoczenia wysokiej jakości (HDRI), 350
  - panorama, rozwinięcie cylindryczne, 351
  - plik na tło renderingu, 347, 356
  - pliki HDR(I), 336
  - podstawianie jako tła sceny, 347, 350, 356
  - problemy z maskowaniem Alpha, 443
  - przejrzystość, 192, 198
  - przesunięcie panoramy, 339
  - przypisanie do mapy deformacji (Displace), 300
- rastrowy
  - kompozycja w Blenderze, 437
  - rastrowy, aktualizacja w Blenderze, 335
  - rastrowy, podgląd tekstury, 335
  - rastrowy, powiązanie z teksturą, 335
  - rozjaśnienie, 376
  - rozmiar na tle widoku, 192
  - rozmycie kierunkowe, 139, 446
  - rozmycie koncentryczne, 140
  - rozmycie liniowe, 139
  - sterowanie zakresem intensywności, 377, 378
  - tła za oszkleniem kabiny, 358
  - tła, przygotowanie w kompozytorze, 140, 451
  - tworzenie masek z głębokości pikseli, 444
  - tworzenie stosu tekstur, 379
  - usuwanie niepotrzebnych z pliku \*.blend, 125
  - uzyskanie negatywu, 376
  - wektorowy z bitmapy (Inkscape), 175, 177
  - wykorzystanie masek do krety, 444
  - wykorzystanie panoramy, 337
  - wykorzystanie w teksturze, 334
  - wyświetlanie przez obiekt Empty, 196
  - zapisanie wyniku renderu, 437
  - zastosowanie dodatkowej mapy odbić, 126
  - złożenie, 352
  - złożenie dwóch obrazów, 448
  - zmiana odcienia, 131, 357, 439, 440, 442
  - zwiększenie ostrości, 436
- obrót
  - wierzchołków (krzywej), 261
  - wokół selekcji, 181
- Octahedron*
  - przełącznik z panelu Display, 231
- odbicia
  - światła w zbliżeniach, 394
- odbicie
  - przykład sterowania kolorem, 325
- odejmowanie
  - siatek, 285
- odzyskiwanie
  - danych (bo błędzie programu), 189
- Offset*
  - pole z panelu Modifiers (Shrinkwrap), 303
- ograniczenia
  - problemy z poleceniami Set Parent, Clear Parent, 248
- ograniczenie
  - Limit Location, 241
  - Locked Track, 237
  - Locked Track w modelowaniu mechanizmów, 237
  - obiektu, 237, 241, 243, 249, 254
  - obiektu, właściwości, 237, 241, 243, 249, 254
  - przełożenie przesunięcia na obrót (Transformation), 57
  - przykład użycia (Copy Location), 58
  - przykład użycia (Limit Location), 53, 55
  - przykład użycia (Locked Track), 52, 53, 234
  - przykład użycia (Stretch To), 59
  - przykład użycia (Track To), 56
  - przykład użycia (Transformation), 54
  - sterowanie wpływem, 77
  - stos ograniczeń, 255
  - Track To, 235
  - Transformation, 243
  - usunięcie, 236
  - wykorzystanie pomocniczego obiektu Empty, 247
  - złożenie, 255
- okno
  - domyślny układ ekranu, 186
  - edytora węzłów, 207



przybornik właściwości (Properties), 185, 211, 319  
 przybornik z narzędziami (Tool Shelf), 185  
 ustawienia barw, 183  
 ustawień Blendera, 180, 201  
 właściwości transformacji, 262  
 zmiana układu, 199, 425

*Opacity*

suwak z panelu Background Images, 192, 198

*Opaque Surface**Normal*

parametr wejściowy, 375  
 shader (grupa), 375  
 shader, przykład użycia, 406

*Open*

polecenie z menu File, 180

## orczyk

w kabinie samolotu, 93

*Origin*

przycisk z panelu Object Tools przybornika Toolshelf, 224

*Origin to 3D Cursor*

polecenie z menu Object, 224

## oś

obrotu, 239, 269

## otwory

w tablicy przyrządów, 88

## otwór

uzyskanie gładkiej krawędzi, 43

*Outputs*

lista z panelu Interface, 319

## owiewka

kabiny, sterowanie przesunięciem, 121, 132

**P***Paintbrush*

polecenie z menu Tools (GIMP), 160

*Pan*

polecenie w Node Editor, 208

## panel

sterowania modelem, 265

## panorama

brak ostrości, 338  
 dolnej półsfery, 151, 351  
 mapowanie obrazu w Cycles, 336  
 obniżenie horyzontu, 354  
 obrazy otoczenia wysokiej jakości, 350  
 obrót, 338, 355  
 otoczenia sceny, 336, 350  
 podgląd, 331

przesunięcie, 339

przykład wykorzystania, 337

rozwiniecie cylindryczne, 351

siatka wokół sceny, 339

Sky Texture, 327

za oszkleniem kabiny, 358

zastosowanie plików HDR(I), 336

złożenie dwóch obrazów, 352

## parametr

grupy węzłów, 319

zakres wartości, 319

*Parent*

pole z panelu Relations (armatura), 230

*Particles**Advanced*

opcja zestawu, 454

analiza efektów na renderze, 456, 457

*Hair*

typ systemu cząsteczek, 454

losowe odchylenie toru cząstek, 454

początkowa prędkość cząsteczek, 454

podgląd wpływu parametrów, 454

przygotowanie nowego systemu, 454

przygotowanie siatki, 148, 452, 453, 457

przypisanie parametru do wag wierzchołków siatki, 458

różnice w prędkościach cząsteczek, 454

sprawdzanie siatki, 453

sterowanie gęstością za pomocą wag, 458

system cząstek (trawa), 452

wpływ na rozmiar pliku Blendera, 455

zestaw okna Properties, 454

## pasek barw

interpolacja barw, 387

opis węzła Color Ramp, 386

węzły, 386, 387

*Pass Index*

pole z paneli Relations, 138, 215, 216

*Passes**Combined*

wybór danych o obrazie (Image), 433

określanie danych wyjściowych, 433

panel z zestawu Render Layers, 433

*Z*

wybór danych głębokości pikseli, 433, 443

*Paste Material*

polecenie z zestawu Material, 274

*Path*

polecenie z menu Curve, 260

*Path Animation*

animacja krzywej, 264

*Pencil*

polecenie z menu Tools (GIMP), 164

*pędzel*

Circle Fuzzy (GIMP), 166

intensywność w trybie Weight Paint, 292

Jitter (GIMP), 164

kontrast na krawędzi efektu (GIMP), 165, 167

Lines (GIMP), 167

narzędzie (GIMP), 160

operacja realizowana w trybie Weight Paint, 292

rozmiar w trybie Weight Paint, 292

Scratches (GIMP), 167

ustalanie wagi w trybie Weight Paint, 292

właściwości, 292

*Physics**Brownian*

pole sekcji Forces, 454

panel z zestawu Particles, 454

*pilot*

dopasowanie do kabiny, 104

dopasowanie kabiny, 103

dopasowanie uprząży fotela, 105

dostosowanie materiałów do oświetlenia, 106

element główny (*parent*), 103

figura do wnętrza kabiny, 102

import figury do modelu, 103

inne poprawki, 104

plik z figurą, 102

pozowanie, 132

*Pivot*

lista w oknie *3D View*, 239, 269

*plakietki*

z napisami, w kabinie, 91

*plany*

wyświetlanie w widoku 3D, porównanie dwóch metod,  
196

*Plastic*

shader (grupa), 419

*plastik*

materiał (nieprzezroczysty), 418

materiał (przezroczysty), 418

*Plexiglas**Glass*

kolor podstawowy, 363

inne zastosowania, 418

*Reflectivity*

sterowanie przezroczystością, 363, 367

shader (grupa), 363

*Specular*

kolor połysku, 363

*plik Blendera*

struktura, 464

*plik startowy*

Blendera, 180, 187

*plik ustawień*

Blendera, 187

*pliki tymczasowe*

zapisywanie roboczych, 189

*podgląd*

rezultatu kompozycji, 435

*podłoga*

kabiny pilota, 86

*podmiana*

węzła, 322

*podwozie*

amortyzator kółka ogonowego, 43, 44, 53

ciągną kółka ogonowego, 46, 56

cylinder amortyzatora, 64

dźwignia sterowania, 95

goleń koła głównego, 63, 64

goleń kółka ogonowego, 38

hak holowniczy, 64

koło główne, 62

kółko ogonowe, 38, 51

luk koła głównego

dopasowanie, 69

luk kółka ogonowego — kopia robocza, 47, 57

*łączniki*

kinematyka, 67

okucia, 65

łączniki amortyzatora, 65

malowanie wnętrza, 75

materiał części stalowych, 39

materiał opony, 40

mechanizm chowania kółka ogonowego, 45, 53

nakrętki, 66

nierówności wnętrza, 75

okucia osi obrotu, 70

okucie amortyzatora, 43

okucie pokrywy kółka ogonowego, 48, 59

okucie wahacza kółka ogonowego, 42

opona, 62

opona kółka ogonowego, 40

piasta kółka ogonowego, 40

pokrywa boczna, 73, 80

popychacz, 73

- pokrywa goleni, 73, 79
- poprawianie nazw po powieleniu, 78
- popychacz kółka ogonowego, 53, 54
- popychacze koła głównego, 70
- popychacze pokryw kółka ogonowego, 48, 59
- powielenie koła głównego, 76
- przewód hamulcowy, 71
  - kinematyka, 72
- rysunki fabryczne, 62
- sterowanie za pomocą panelu, 81
  - przygotowanie, 81
- szkice wykonawcze, 38, 62
- śruby, 66
- tłok amortyzatora, 63
- tracza hamulca, 63
- wahacz kółka ogonowego, 41, 46, 52
- weryfikacja ruchu, 80
- wręga kółka ogonowego, 47
- wypusty wahacza kółka ogonowego, 41
- zębátky koła głównego, 70
- podwozie główne
  - uchwyt (sterujący), 68
- pokrywa boczna
  - podwozia głównego, 73, 80
  - popychacz, 73
- pokrywa goleni
  - podwozia głównego, 73, 79
- polecenia
  - ustawienia skrótów do, 182
- połączenie
  - węzłów, problem z widocznością, 211
  - węzłów, sterowanie kształtem, 212
- poprawki
  - rezultatu wektoryzacji (Inkscape), 177
- Pose Mode*
  - tryb pracy, 231
- Post Processing*
  - panel z zestawu Render, 434
- Posterize*
  - polecenie z menu Color (GIMP), 175
- postprocessing*
  - renderowanego obrazu, 437
  - układ ekranu, 434
  - w kompozytorze, 437
- powielanie
  - siatek, 297
- powierzchnie
  - deformacja za pomocą siatki, 312
  - krawędź przecięcia — wyznaczanie, 221
- powierzchnie podziałowe
  - łączenie gładkiej i ostrej krawędzi, 43
- powiększanie
  - zawartości okna, 208
- pozowanie
  - pozycja spoczynkowa, 231
  - tryb pracy, włączanie, 231
  - zaznaczanie kości, 232
- pozycja spoczynkowa
  - wyjaśnienie terminu, 231
- Project From View*
  - opcja z menu UV Mapping, 277
- Properties*
  - polecenie z menu View, 262
  - przybornik, 191
  - przybornik właściwości, 185, 211, 319
  - wyświetlanie kontroltek węzłów w panelu okna, 208
- przecięcie
  - powłok
    - wyznaczanie krawędzi, 221
- przegubowe
  - połączenie, modelowanie, 254, 255, 256
- przełączanie
  - konfiguracji Blendera, 188
- przesunięcie
  - obiektu, 270
  - wierzchołków (krzywej), 261
- przesuwanie
  - węzła, 208
  - zawartości okna, 208
- przewód hamulcowy
  - podwozia głównego, 71
- przybornik
  - nakładany na edytor, 185
  - zmiana rozmiaru obrazu w edytorze po wysunięciu, 185
- przycisk
  - wyboru koloru, 204
- przypisanie
  - istniejącego materiału materiału do obiektu, 227
  - nowego materiału materiału do siatki, 227
- punkt pomocniczy
  - przesunięcie (Node Editor), 212
  - usunięcie (Node Editor), 212
  - wstawienie (Node Editor), 212

## Q

### *Quick Mask*

- edycja obszaru zaznaczenia (GIMP), 160

tryb pracy (GIMP), 160  
zachowanie obszaru zaznaczenia (GIMP), 162

## R

### R

składnik barwy, 468  
ramka  
  dodawanie węzła, 211  
  etykieta, 211  
  łącząca węzły, 210  
  przesuwanie, 210  
  tworzenie, 210  
  usuwanie, 211  
  właściwości, 211  
  wyjęcie węzła, 211  
*Range From Min*  
  pomocniczy węzeł (grupy), 378  
*Range To Max*  
  pomocniczy węzeł (grupy), 377  
*raytracing*  
  ogólna zasada działania, 324  
*Recover Auto Save*  
  polecenie z menu File, 189  
reflektor  
  do lądowania, formowanie, 115  
*Region Overlap*  
  przełącznik z sekcji User Preferences  
    System, 185  
*Relations*  
  panel z zestawu Object, 215  
*Relative Offset*  
  pole z panelu Modifiers, 297  
*Reload Image*  
  polecenie z menu Image, 335  
*Remove*  
  przycisk z panelu dodatku (Add-On), 202  
*Remove from Frame*  
  polecenie z menu Node, 211  
render  
  artefakt  
    czarne obszary na szkłe, 305  
  czas dla tekstur proceduralnych, 399, 401  
  dobór intensywności światła kierunkowych i otaczających, 131, 133  
  dostosowanie rozmycia ruchu, 430  
  efekt 'przepalenia bieli', 442  
  intensywność światła kierunkowego ('słońca'), 129  
  intensywność światła otaczającego, 131  
  korygowanie kolorystyki obrazu, 134

max. dopuszczalna jasność promienia, 365  
odcień dalszego obiektu, 138  
odczytanie wiadomości o obiekcie, 216  
odrzućcie zbyt jasnych promieni, 365  
problemy z maskowaniem Alpha, 443  
problemy ze szkłem, 362, 365, 366  
przezroczyste odbicia, 366  
sceny na ziemi, 143, 145  
sceny w locie, 129  
sceny z dwoma samolotami (tego samego typu), 137, 213  
sterowanie rozmyciem cieni, 135, 144  
trawa w Cycles, 455  
tworzenie masek z przebiegu Z, 444  
włączenie efektów kompozycji, 434  
wykorzystanie światła pomocniczego, 135  
zapisanie wyniku do pliku rastrowego, 437  
zasada tworzenia obrazu, 324  
zmiana odcienia obrazu, 439, 440, 442  
zmiana odcienia obrazu monochromatycznego, 441  
zwiększenie ostrości obrazu, 436

### Render

#### *Experimental*

tryb Cycles, 455

#### *Group*

tryb systemu cząsteczek, 455

#### *Material*

przypisanie materiału do emitowanych cząstek, 456

#### *Object*

tryb systemu cząsteczek, 455

#### *Path*

tryb systemu cząsteczek, 455

### *Render Layer*

odmiana węzła kompozycji, 432

### *Render Layers*

#### *Z*

głębokości pikseli, 443

zestaw okna Properties, 432

### *Render Pass, Patrz Render Layer*

### Render Result

dynamiczny obraz, 434

### renderer

wyróżnienie obiektu, 138, 215, 216

### rendering

warstwa kamer i lamp, 186

### renderowanie

pojedyncze przebiegi, 432

#### warstwa

opcje danych wyjściowych, 433



tworzona domyślnie, 432  
 wyjaśnienie pojęcia, 432  
 wykluczanie warstw sceny z, 433

#### *Rest position*

wyjaśnienie terminu, 231

#### *RGB*

tryb wyboru koloru, 204

#### *RGB to BW*

węzeł konwertujący, 209

#### *Rotate*

polecenie z menu Curve, 261

zakładka panelu Transform (Inkscape), 174

#### *Rotate Around Selection*

przełącznik z sekcji User Preferences

Interface, 181

#### *rozmycie*

metodą Gaussa (GIMP), 163

#### *rozmycie ruchu*

centralny punkt deformacji tła, 450

intensywność rozmycia tła, 446, 450

kierunek rozmycia tła, 448

śmigła, 429, 430, 431

tła, 139, 446

tła, koncentryczne, 140, 449, 451

tła, liniowe, 139, 446

zakres (w klatkach animacji), 130, 429, 430

zgodność kierunków obrazu i modelu, 450

#### *rozwiniecie*

alternatywne, wartości domyślne, 382

siatki, metodą Smart UV Project, 415

UV, alternatywne, 282, 380, 381

UV, metody dla siatki, 276, 277

UV, zaznaczanie szwu, 280

UV, zaznaczenie w UV/Image Editor, 280

#### *rysowanie*

linii pomocniczych, 422

obszaru wyboru (GIMP), 160

pomocniczych linii na ekranie, 421, 458

wybór trybu dla linii pomocniczych, 422

#### *rzutowanie*, 34, 302

## **S**

### **S**

nasycenie koloru, 204

#### *Samples*

zespół kontrolki z panelu Render

Integrator, 365

#### *Save Startup File*

polecenie z menu File, 180, 187

#### *Save to Channel*

polecenie z menu Select (GIMP), 162

#### *Save User Settings*

przycisk z okna User Preferences, 180, 187

#### *Scale*

polecenie z menu Apply (Object Mode), 251

polecenie z menu Image (GIMP), 175

#### *scena, Patrz scene*

animacja, 130

dopasowanie położenia 'słońca', 327

dostosowanie oświetlenia do fotografii tła, 144

na ziemi, barwa materiału podłoża, 147

na ziemi, kompozycja, 143, 145, 146

na ziemi, materiał podłoża, 146

na ziemi, panorama dolnej półsfery, 151

na ziemi, sterowanie jasnością cienia, 148

na ziemi, uzyskiwanie cienia, 147

na ziemi, zmiany w kompozycji, 150

obniżenie horyzontu panoramy otoczenia, 354

obraz panoramy na siatce, 339

obraz tła za oszkleniem kabiny, 358

opcje tworzenia, 203

otoczenie, 327, 337, 347

panorama dolnej półsfery, 351

panorama otoczenia, 336, 350

panorama Sky Texture, 327

podgląd otoczenia, 331

shader Background, 327, 337, 347

tło obrazu, 347, 350, 356

usuwanie, 203

utworzenie nowej, 137, 203, 213

użycie Gradient Texture do podziału otoczenia sceny,  
341, 353

w locie, kompozycja, 129

w locie, korekta kolorystyki, 134

w locie, odcień dalszego obiektu, 138

wizualizacja dwóch samolotów, 137, 213, 214

wykorzystanie drugiej sceny jako tła, 214

wyróżnienie powierzchni przejrzystych, 358

wyświetlanie zawartości w drugiej scenie, 214

zarządzanie, 203

zastosowanie węzła Geometry, 332

zastosowanie węzła Mapping, 332, 338, 341, 348

zastosowanie węzła Texture Coordinate, 338, 341

zmiana nazwy, 203

zmiana odcienia panoramy otoczenia, 131, 357

zróżnicowanie oświetlenia modelu i obrazu tła, 356

#### *scena w locie*

rozmycie ruchu śmigła, 429, 430, 431

*Scene, Patrz scena*

- kontrolka z panelu Layers, 433
- panel z zestawu Scene, 214
- pojęcie w Blenderze, 466

*schemat*

- węzłów kompozycji, 432

*Scratches*

- rodzaj pędzla (GIMP), 167

*Screen layout*

- kontrolka z nagłówka okna Info, 199

*screws.png*

- zastosowanie w materiale B.Skin.Camouflage, 126

*scripts*

- folder konfiguracji Blendera, 187

*Seam*

- siatka, szwy rozwinięcia UV, 280

*sekcja*

- Modifiers, 285, 297, 300, 302, 309

*Select*

- opcja z menu Erase, 261

*Set Deformer*

- polecenie z menu Specials, 28, 312

*Set Parent*

- problemy z ograniczeniami, 248
- współrzędne obiektu potomnego, 245, 246

*Set Parent To*

- menu rozwijalne przypisania 'rodzica', 232

*Shade Flat*

- tryb cieniowania siatki, 304

*Shade Smooth*

- tryb cieniowania siatki, 304

*shader*

- definiowanie nowego, 317
- przejrzyste odbicia, 366

*Sharp*

- oznaczanie ostrych krawędzi, 306

*Sharpen*

- funkcja węzła Filter, 436

**Shift-D, Patrz Duplicate****Shift-O (GIMP), Patrz By Color (jedna z form zaznaczania)****Shift-Q (GIMP), Patrz Toggle Quick Mask***Show Hidden*

- polecenie z menu Object, 225

*Shrink*

- polecenie z menu Select (GIMP), 159

*Shrinkwrap*

- modyfikator siatki, 34, 302

*Vertex Group*

- parametr modyfikatora, 303

*Shutter*

- dobór wartości, 430
- kontrolka z panelu Motion Blur, 130, 429

*siatka, Patrz Mesh*

- 'utrwalenie' modyfikatora, 309
- aktywne łącze materiału, 274
- alternatywne rozwinięcia UV, 282
- cienie przy ostrych krawędziach, 305
- częściowe wyginanie wzdłuż krzywej, 291
- deformacja siatką uproszczoną, 27, 309
- deformacja za pomocą, 312
- deformacja za pomocą tekstury, 300
- deformująca
  - 'utrwalenie', 315
  - aktywacja, 313
  - deaktywacja, 315
  - odłączenie od przypisanych obiektów, 315
  - panel sterowania efektem, 313
  - przypisanie do, 28, 312
  - usuwanie przypisania, 313
- dodadnie modyfikatora, 285, 302, 309
- dodanie nowego łącza materiału, 274
- domyślna waga wierzchołków, 291
- fazowanie krawędzi, 294, 295
- fazowanie wierzchołków, 294
- grupy wierzchołków, 278
- interpolacja kierunków normalnych, 304
- lustrzane odbicie, 297
- łącze materiału, 274
- metody rozwinięcia w UV, 276, 277
- nanoszenie wag wierzchołków, 291, 458, 459
- odejmowanie, 285
- ograniczanie wpływu MDeform, 30, 32
- określanie współczynników wpływu modyfikatora, 292
- opcje rozwinięcia Unwrap, 276
- oznaczanie ostrych krawędzi, 306
- poprawianie siatki, 302
- powielanie, 110, 297
- przypisanie deformującej (Mesh Deform), 309
- przypisanie do materiału, 273
- przypisanie fragmentu do modyfikatora, 292
- przypisanie indywidualnych ścian do różnych materiałów, 275
- przypisanie materiału trawy, 149, 456
- przypisanie wielu materiałów, 274
- przypisanie wielu siatek do jednego materiału, 273, 274

- rozwiniecie w UV, 276, 277, 415
- rzutowanie na powloke, 34, 302
- skopiowanie przypisania do materialu, 274
- skręcanie, 290
- suma, różnica, część wspólna, 285
- szwy UV, widoczność, 280
- tryb wyświetlania, 83
- ustalanie szerokości fazy, 294
- usunięcie łącza materialu, 274
- utrwalanie skali, 251
- UV
  - mapowanie aktywne, 283
  - mapowanie domyślne dla renderowania, 283
  - wklejenie przypisania do materialu, 274
  - współdzielona — problemy z poleceniem Apply (modyfikatora), 30, 315
  - wyginanie, 288, 292
  - wyginanie wzdłuż krzywej, 97, 109, 288, 291
  - wygładzenie, 299
  - zaokrąglanie krawędzi, 294
  - zastosowanie modyfikatora podczas edycji, 289
  - zaznaczanie nazwanej grupy wierzchołków, 278
  - zaznaczanie szwów rozwinięcia UV, 280
- siatki
  - krawędź przecięcia, 221
- Single Bone*
  - polecenie z menu *Armature*, 229
- Size*
  - kontrolka z panelu *Object Data*
    - Lamp, 135, 144
  - pole panelu *Background Images*, 192
- skala
  - utrwalanie w siatce, 251
- skalowanie
  - środków obiektów, 239, 270
- składnik
  - barwy, 350
- skóra
  - brązowa
    - fałdy i odbłaski, 412
  - dodanie śladów zużycia, 411
  - komponent *X.Leather.Band*, 409
  - kompozycja podstawowej tekstury, 410
  - kompozycja tekstur proceduralnych, 409
  - materiał *B.Black.Leather*, 411
  - materiał *B.Brown.Leather*, 411
  - tekstura nierówności, 409
  - tekstura podstawowa, 408
  - tekstura *X.Leather.Pattern*, 410
  - wykorzystanie tekstury *X.Leather.Pattern*, 410
- skręcanie
  - siatki wokół krzywej, 290
- skrypty
  - Blendera, 183, 201
- skrzydło
  - szczegóły połączenia z kadłubem, 26
- Sky Texture*
  - Ground Albedo*
    - intensywność odbić z ‘ziemi’, 329
  - Hosek/Wilkie*
    - tryb pracy, 329
  - obraz otoczenia, 327
  - polecenie z menu *Add*, 327
  - Preetham*
    - tryb pracy, 327
  - sterowanie położeniem ‘słońca’, 327
  - sterowanie zamgleniem, 328
  - Turbidity*
    - parametr, 328
  - węzeł tekstury, 327
- Smart UV Project*
  - polecenie z menu *Mesh*, 415
- Smooth Shade*
  - efekt uboczny na ostrych krawędziach, 304, 305
- Snap*
  - polecenie z menu *Mesh*, 271
  - polecenie z menu *Object*, 239, 269
- Softness*
  - sterowanie rozmyciem połysku, 363
- sonda
  - deformacja obrazu odbitego na powierzchni kuli, 331
  - do podglądu otoczenia, 330, 331
- Space**, Patrz *Add Object Menu*
- Specials*
  - menu podręczne edycji siatki, 294
- Split Angle*
  - parametr modyfikatora *Edge Split*, 305
- sprężyna
  - przykład odwzorowania, 46
- sRGB*
  - tryb konwersji koloru, 205
- Stack Image*
  - Alpha*
    - wejście (na nieprzejrzyistość ‘górnego’ obrazu), 379
  - Background*
    - wejście (na ‘dolny’ obraz), 379
  - budowanie stosu tekstur, 379
  - Color*

- wejście (na ‘górny’ obraz), 379
- Opacity*
  - intensywnością efektu, 379
  - pomocniczy węzeł (grupy), 379
- stal
  - gładź tłoka, 396
  - kuta, materiał, 399
  - materiał, 396
  - materiał linki, 397
  - materiał sprężyny, 398
  - ocynkowana, materiał, 401
- startup.blend
  - ustawienia Blendera, 180, 187
- ster
  - kierunku, detale, 122
- ster kierunku
  - detale, 121
  - sterowanie obrotem, 122
- ster wysokości
  - sterowanie obrotem, 122
- sterowanie
  - chowaniem podwozia, 81
  - klapami chłodnicy, 113
  - klapami skrzydła, 118
  - lotkami, 118
  - obrotem śmigła, 113
  - owiewką kabiny, 121, 132
  - sterem kierunku, 122
  - sterem wysokości, 122
  - zespołami modelu, 81
  - przygotowanie, 81
- Stick*
  - przełącznik z panelu Display, 231
- strefy
  - kabiny, 93, 100
- Strength*
  - kontrolka z panelu Object Data
    - Nodes, 129
  - kontrolka z panelu World
    - Surface, 131
  - pole z panelu Modifiers (Displace), 301
- Stretch To*
  - korygowanie ujemnej skali obiektu, 251
  - lista modyfikatorów, 249
  - oś śledząca cel, 250
- Plane*
  - oś obrotu, 250
  - przykład użycia, 59
  - przykład zastosowania, 67
- Rest Length*
  - dystans odniesienia, 250
- Target*
  - nazwa źródła transformacji, 249
- utworzenie, 249
- Vol*
  - dodatkowe kierunki zmiany skali, 250
  - wpływ ujemnej skali obiektu, 78, 250
- struktura
  - pliku Blendera, 464
- Subtract*
  - sposób nakładania barw, 470
- Subtract from Selection*
  - polecenie z menu kontekstowego Channels (GIMP), 162
- Subtraction*
  - sposób nakładania barw, 470
- Subsurf*
  - modyfikator siatki, 299
- Surface Output*
  - węzeł wyjściowy (grupa), 398
- System*
  - sekcja okna User Preferences, 184, 185
- szkielet
  - kadłuba, weryfikacja, 83
- szkło
  - efekt ‘światlików’, 362, 365
  - kuloodporne, kabiny, 92
  - odbicia otoczenia, 366
  - połączenie shaderów Glass i Glossy, 362
  - połączenie shaderów Transparent i Glossy, 367
  - połysk na powierzchni, 362, 367
  - problemy z odbiciem otoczenia, 366
  - problemy z renderem, 362, 365
  - przezroczyste odbicia otoczenia, 366
  - wykorzystanie Glass BSDF, 360
  - wykorzystanie shadera Glossy BSDF, 362
  - wykorzystanie shadera Transparent BSDF, 361
  - wykorzystanie współczynnika Fresnela, 362, 367
- szum
  - drobne nieregularności powierzchni, 390
  - pomocnicza tekstura, 392

## Ś

- ścieżka
  - deformacja poprzeczna, 290
  - edycja, 261
  - kierunek poprzeczny do, 262
  - krańce krzywej — położenie, 262



krańce krzywej — styczne, 262  
 liczba wierzchołków, 264  
 rząd krzywej, 262  
 skręcenie, 98, 262, 290  
 utworzenie, 260, 263  
 waga punktu sterującego, 262  
 wielobok sterujący krzywą, 262  
 właściwości, 264  
 współrzędne punktu sterującego, 262  
 wyginanie wzdłuż, 97, 109, 288, 291  
 śmigło  
   animacja obrotu, 130, 425  
   rozmycie ruchu, 429, 430, 431  
   sterowanie obrotem, 113  
 średnia ważona  
   sposób nakładania barw, 469  
 środek  
   zmiana dla obiektu, 224  
 śruba  
   przykład odwzorowania, 47  
 śruby  
   podkreślenie na teksturze barwy, 126  
   podwozia, 66  
 światło  
   dobór intensywności, 131, 133  
   dobór parametrów dla sceny na ziemi, 144  
   dostosowanie do fotografii w tle, 144  
   kompensowanie oświetlenia tekstury, 144  
   otaczające  
     sterowanie intensywnością, 131  
   pozycyjne, formowanie, 119  
   pozycyjne, materiały, 119  
 Sun  
   sterowanie intensywnością, 129  
   sterowanie rozproszeniem cieni, 135, 144  
 źródło pomocnicze, 135

## T

tablica  
   instalacji elektrycznej, 94  
   przyszytych pokładowych, 87  
   przyszytych pokładowych, detale, 91  
   przyszytych pokładowych, otoczenie, 93  
   przyszytych pokładowych, otwory, 88  
   przyszytych pokładowych, plakietki, 91  
 tarcza  
   do przesuwania osłony kabiny, 96  
*Target*  
   pole z panelu Modifiers (Shrinkwrap), 302

tekstura  
   aktualizacja obrazu, 335  
   barwy  
     podkreślenie śrub, 126  
   barwy wnętrza kabiny, 85  
   brak ostrości obrazu, 338  
   dołączanie do schematu, 334  
   drobnych niejednorodności odbić światła, 394  
   drobnych zabrudzeń, 126  
   jak źródło współczynnika łączenia obrazów, 352  
   kompensowanie silnego oświetlenia, 144  
   kompozycja panoramy dolnej półsfery, 151, 351  
   kompozycja proceduralnych ‘szumów’ (przykład), 346  
   kompozycja tekstur proceduralnych, 409  
   korekta barw obrazu, 334, 337  
   korekta gamma, 334  
   kształt gradientu, 340  
   mapa normalnych, 374  
   mapowanie, 347  
   mapowanie obrazu panoramy, 336, 339  
   mapowanie panoramy otoczenia, 351  
   modyfikacja tekstury Waves, 386  
   nakładanie jednej na drugą, 379  
   nierówności komory podwozia, 75  
   nierówności skóry, 409  
   nierówności wnętrza kabiny, 85  
   nierówności wokół otworów na tablicy przyrządów, 88  
   odbicie wnętrza kabiny, 85  
   odwzorowanie drobnych napisów, 108  
   otworów na przyrządy pokładowe, 88  
   pasów fotela pilota, 416  
   piksele tekstury, 334, 340, 343  
   pliki HDR(I), 336  
   podgląd obrazu, 335  
   powiązanie z plikiem obrazu, 335  
   proceduralna  
     Sky Texture, 327  
   proceduralna - przykład użycia, 402  
   proceduralna, czas renderu, 399, 401  
   przygotowany obraz ‘szumu’, 392  
   przykład wykorzystania panoramy, 337  
   rozjaśnianie ciemnych miejsc, 376  
   rozwiniecie siatki w UV, 276, 277  
   różnicowanie ze względu na obiekt, 138, 216  
   skóry, 408  
   skóry, podstawowa, 410  
   skóry, przygotowanie komponentu, 409  
   sterowanie intensywnością, 404  
   sterowanie zakresem intensywności, 377, 378

- tarcz z tablicy przyrządów, 90
- tkaniny, 415
- tła, przygotowanie w kompozytorze, 140, 451
- tworzenie stosu tekstur, 379
- utworzenie nowej (rastrowej), 334
- użycie gradientu do podziału otoczenia sceny, 341, 353
- wartości ‘szumu’, 343
- wartości gradientu, 340
- Waves – ostre krawędzie pasków, 387
- Waves – użycie w materiale skóry, 408
- Waves – użycie w materiale tkaniny, 415
- Waves – zmiana wzoru pasków, 387
- węzeł (gradientu), 340
- węzeł (losowe ‘komórki’), 344
- węzeł (losowy szum), 343
- węzeł (panoramy otoczenia), 336
- węzeł (pliku rastrowego), 334
- węzeł (wzory pasków), 388
- węzeł (złożony szum), 345
- wielokrotne używanie tego samego obrazu, 334
- włókien tkaniny, 415, 416
- wnętrza kabiny, 84
- wnętrza komory podwozia, 75
- wpływ trybu cieniowania na współrzędne Normal, 385
- wpływ trybu Shade Flat na współrzędne Normal, 385
- wpływ trybu Shade Smooth na współrzędne Normal, 385
- współrzędne, 347
- współrzędne a kamera, 348
- współrzędne Generated, 384, 402
- współrzędne Normal, 384, 385
- współrzędne Object, 384, 397, 408
- współrzędne okna renderu, 347, 356
- współrzędne tekstury, 334, 340, 343
- współrzędne UV, 383
- wykorzystanie węzła Bump, 404
- wzór pasków, 388
- wzór pierścieni, 388
- zabrudzeń pasów (tkanina), 416
- zastosowanie do łączenia materiałów, 126
- zastosowanie dodatkowej mapy odbić, 126
- złożenie dwóch obrazów, 352
- złożony obraz ‘szumu’, 390
- zmiana odcienia obrazu, 131, 357
- zmiana znaku, 376
- problem z odbiciem obrazu otoczenia, 349
- współrzędne aktywnej kamery, 348
- Generated*
  - współrzędne tekstury, 384, 402
- Normal*
  - współrzędne tekstury, 384, 385
- Object*
  - przykład użycia, 390, 391
  - współrzędne tekstury, 384, 397, 408
- polecenie z menu Input, 383
- UV*
  - współrzędne tekstury, 383
- węzeł wejściowy, 338, 341, 347, 383
- Window*
  - problem z odbiciem obrazu otoczenia, 348, 349, 358
  - współrzędne okna renderu, 347, 356
- Texture Coordinates*
  - pole z panelu Modifiers (Displace), 301
- Textured*
  - tryb wyświetlania siatki, 83
- Themes*
  - sekcja okna User Preferences, 183
- Timeline*
  - okno osi czasu, 425
- Timer*
  - interwał autozapisywania, z sekcji File, 189
- tkanina
  - pasów fotela pilota, 416
  - podstawowe barwy, 416
  - przygotowanie siatek, 415
  - tekstura włókien, 415, 416
- tło
  - dopasowanie rozmiaru do widoku 3D, 193
  - dostosowanie oświetlenia sceny, 144
  - podstawienie obrazu, 129, 143
  - pozowanie modelu, 129, 143
  - problem z małą rozdzielczością obrazu, 195
  - przejrzystość obrazu, 192, 198
  - rozmiar obrazu, 192
  - sterowanie jasnością cienia modelu, 148
  - ustawienie w kamerze, 129, 198
  - ustawienie w wielu oknach, 195
  - uzyskanie cienia modelu na zdjęciu, 147
  - użycie obiektów Empty, 196
  - widoku 3D, 191
- To Scene*
  - polecenie z menu Object, 203
- Toggle Quick Mask*
  - polecenie z menu Select (GIMP), 160, 166
- Texture*
  - pole z panelu Modifiers (Displace), 300
- Texture Coordinate*
  - Camera*

*Tool Shelf*

przybornik z narzędziami, 185

*Trace Bitmap*

polecenie z menu Path (Inkscape), 176

*Track To*

opis ograniczenia, 235

przykład użycia, 56

*Target Z*

podążanie za orientacją 'celu', 235

usunięcie ograniczenia, 236

*Track To Constraint*

utworzenie ograniczenia, 235

*Transform*

panel (dla krzywej), 262

panel (Inkscape), 174

polecenie z menu Object (Inkscape), 174

*Radius*

deformacja poprzeczna, 290

*Tilt*

skręcenie krzywej, 98, 262, 290

*W*

waga punktu sterującego, 262

## transformacja

obiektu (Inkscape), 174

*Transformation**Destination*

transformacje obiektu z ograniczeniem, 244

lista modyfikatorów, 243

*Loc*

opcja translacji, 244

*Local Space*

lokalny układ współrzędnych obiektu, 244

opis ograniczenia, 243

przełożenie przesunięcia na obrót, 245

przykład obrotu, 57

przykład użycia, 54

*Rot*

opcja obrotu, 244

opcja translacji, 245

*Scale*

opcja zmiany rozmiaru, 244

*Source*

transformacje obiektu wzorcowego, 244

*Space*

wybór typu układu współrzędnych, 244

*Target*

nazwa źródła transformacji, 244

utworzenie, 243

wykorzystanie pomocniczego obiektu Empty, 247

*Z Max*

koniec zakresu, 244

*Z Min*

początek zakresu, 244

*Transparent BSDF*

shader, 361

## trawa

analiza efektów na renderze, 456, 457

długość żdźbła, 454

liczba żdźbeł, 454

losowe odchylenia żdźbeł, 454

metody odwzorowania kształtu żdźbła, 455

podgląd wpływu parametrów, 454

poprawianie granicy cienia, 148, 452

przygotowanie siatki, 148, 452, 453, 457

przypisanie materiałów, 149, 456

przypisanie materiału, 456

przypisanie parametru do wag wierzchołków siatki, 458

różnice w długości żdźbeł, 454

sprawdzanie siatki, 453

sterowanie gęstością za pomocą wag, 458

sterowanie lokalną gęstością, 149

stworzenie nowego systemu cząsteczek, 454

szczegóły materiału, 456

wpływ na rozmiar pliku Blendera, 455

*Triple Buffer*

opcja Window Draw Method (User Preferences System), 185

## tryb

edycji, włączanie, 261

pozowania, włączanie, 231

## tworzenie

węzła, 208

## typ

automatyczna konwersja dla węzłów, 209

węzła, 209

*Type*

lista rozwijalna z zakładki Textures (Displace), 300

## U

## U, Patrz UV Unwrap

## uchwyt

kryteria identyfikujące, 265

nazwa, 268

panel sterowania, 81

przygotowanie dla, 81

położenie neutralne, 265, 267

sterowanie położeniem, 267

- sterowanie ruchomymi częściami, 265
- sterownie kółkiem ogonowym, 52, 80
- sterownie podwoziem głównym, 68
- ukrywanie
  - obiektów, 225
- Unwrap*
  - opcja z menu UV Mapping, 276
  - parametry polecenia, 276
  - polecenie z menu Mesh, 276
- uprząż
  - fotelu pilota, 97
  - końcówki pasów, 98
  - modelowanie pasów, 97, 105
  - poprawianie pasów spadochronu, 104
  - sprzączki, 97
- Use Nodes*
  - przełącznik w oknie Node Editor, 432
- User Preferences*
  - okno, 180, 201
  - sekcja Add-Ons, 183
  - sekcja Editing, 181
  - sekcja File, 184, 189
  - sekcja Input, 182
  - sekcja Interface, 181
  - sekcja System, 184, 185
  - sekcja Themes, 183
- userpref.blend
  - ustawienia Blendera, 187
- ustawienia
  - Add-Ons, 183
  - aktywnych dodatków (add-ons), 183
  - autozapis, 184, 189
  - barw okna, 183
  - Blendera dla tej książki, 188
  - czcionek ekranu, 184
  - domyślnego układu ekranu, 186
  - Editing, 181
  - File, 184
  - folder config, 187
  - folder scripts, 187
  - Input, 182
  - Interface, 181
  - nadpisanie istniejących, 188
  - plik startowy, 180, 187
  - plik startup.blend, 180, 187
  - plik userpref.blend, 187
  - plik z konfiguracją programu, 187
  - położenie katalogu, 187
  - przełączanie, 188
  - skrótów klawiatury, 182
  - System, 184, 185
  - Themes, 183
  - zapisywanie, 180, 187, 188
- usuwanie
  - aktualnej sceny, 203
  - dodatków, 202
  - linii pomocniczych, 422
  - linii ruchu (F-Curve), 427
  - menu Erase, 261
  - niepotrzebnych obrazów z pliku \*.blend, 125
  - ograniczenia, 236
  - okno opcji, 261
  - ramki, 211
  - układu ekranu, 200
  - węzłów, 208
  - wszystkich wystąpień bloku danych, 125
- utworzenie
  - nowej sceny, 137, 203, 213
  - nowej tekstury rastrowej (Image Texture), 334
  - powiązania obiektu i materiału, 227
  - powiązania siatki i materiału, 227
- UV
  - alternatywne, rozwinięcia tej samej siatki, 282, 380, 381
  - mapowanie, aktywne, 283
  - mapowanie, domyślne dla renderowania, 283
  - mapowanie, kontrolka przełącznika, 283
  - mapowanie, usunięcie, 282
  - mapowanie, utworzenie nowego, 282
  - mapowanie, zmiana nazwy, 282
  - rozwinięcie wnętrza kabiny, 84
  - wartości domyślne dla alternatywnego rozwinięcia, 382
- UV Editing*
  - układ ekranu, 199
- UV Fallback*
  - pomocniczy węzeł (grupy), 382
- UV Maps*
  - alternatywne rozwinięcia UV, 282
  - dodanie nowego rozwinięcia siatki, 282
  - panel z zestawu Object Data, 282
- UV Unwrap*
  - polecenie z menu Mesh, 277
- UV/Image Editor*
  - wykorzystanie okna, 276, 277
  - zaznaczanie szwów rozwinięcia siatki, 280
- UVMapping*
  - opcje polecenia Unwrap, 276, 277



## V

## V

intensywność koloru, 204

*Vector*

mapowanie, 332, 338, 341

źródło danych, 332, 338, 341

*Velocity**Normal*

pole z sekcji emitter Geometry, 454

panel z zestawu Particles, 454

*Random*

pole panelu sekcji Other, 454

*Vertex Group*

pole z panelu Modifiers (Displace), 301

przykład użycia (w modyfikatorze), 303

*Vertex Groups**[+]*

przycisk na panelu, 278

*Assign*

przycisk na panelu, 278

*Density*

pole panelu, 458

*Deselect.*

przycisk na panelu, 279

panel z zestawu Object Data (siatki), 278, 291

panel zestawu Particles, 458

*Select*

przycisk na panelu, 278

*Weight*

pole na panelu, 291

*VGroup*

pole modyfikatora, 292

przypisanie wierzchołków do modyfikatora, 292

*View*

(New Objects Align to) opcja z sekcji User Preferences

Editing, 181

panel z przybornika Properties, 193

*Viewer*

odmiana węzła kompozycji, 434

*Viewer Node*

dynamiczny obraz, 435

*Voronoi**Cells*

przykład użycia, 400, 405

*Intensity*

przykład użycia, 390, 404, 405

*Voronoi Texture**Cells*

typ tekstury, 345

*Intensity*

typ tekstury, 344

polecenie z menu Texture, 344

*Scale*

rozmiar 'komórek', 344

tekstura proceduralna, 344

węzeł tekstury, 344

## W

*W, Patrz Specials (menu)**wagi*

sterowanie gęstością trawy, 149

wierzchołków siatki

malowanie, 291, 458, 459

wartość domyślna, 291

*warstwa*

elementów kabiny, 106

kamer i lamp, 186

linii pomocniczych, 423

*renderowania*

aktywna, 433

dodawanie, 433

domyślna, 432

opcje danych wyjściowych, 433

usuwanie, 433

wybór warstw sceny, 433

wyjaśnienie pojęcia, 432

wykluczanie warstw sceny, 433

zmiana nazwy, 433

*Wave**Bands*

przykład użycia, 402

*Wave Texture**Bands*

przykład użycia, 397

wzór pasków, 388

*Detail Scale*

rozmiar zaburzeń kształtu, 389

*Details*

szczegółowość zaburzeń kształtu, 389

*Distortion*

zaburzenia kształtu pasków, 389

modyfikacja tekstury, 386

obracanie tekstury, 388

ostre krawędzie pasków, 387

polecenie z menu Texture, 388

*Rings*

wzór pierścieni, 388

*Scale*

- rozmiar obrazu, 388
- tekstura proceduralna, 388
- węzeł tekstury, 388
- zmiana wzoru pasków, 387
- Waves*
  - przykład użycia, 408, 415
- Weight*
  - aktualna waga (współczynnik wpływu), 292
- Weight Paint*
  - malowanie wag siatki, 292
  - tryb pracy Blendera, 292
- wektory
  - sterowanie kierunkiem, 332, 338, 341
- wektoryzacja
  - nanoszenie poprawek (Inkscape), 177
  - obrazów rastrowych (Inkscape), 175
  - omówienie rezultatów (Inkscape), 177
- węzły
  - Attribute (wejściowy), 381
  - automatyczna konwersja typów, 209
  - Color Ramp, 386
  - definiowanie nowego, 317
  - definiowanie wejść grupy, 318
  - dodawanie do ramki, 211
  - dodawanie nowych, 208
  - dołączanie do Image Texture, 334
  - edycja grupy, 318
  - edytor, 207
  - Environment Texture, 336
  - Geometry, 332
  - Gradient Texture, 340
  - grupowanie, 317
  - Image Texture, 334
  - kolor wejść i wyjść, 209
  - komentarze do schematu, 211
  - kompozycji, 432
    - barwy, 134, 138, 438, 439
    - filtry, 134, 435
    - najprostszy schemat, 432
    - odcięcie części obrazu, 139, 449
    - przełączenie w oknie Node Editor, 432
    - rozmycie, 139, 444, 446
    - węzły wejściowe, 432, 437
    - węzły wyjściowe, 432, 434
    - wybór danych wyjściowych, 433
    - złożenie dwóch obrazów, 448
  - kopiowanie i wklejanie, 210
  - Light Path (Cycles), 324
  - łączenie, 207
    - łączenie za pomocą ramki, 210
  - Mapping, 332, 338, 341
  - Math, 358, 443
  - Mix, 352, 357
  - Moderate (grupa), 376
  - Musgrave Texture, 345
  - nawigacja po schematach grup ('drzew węzłów'), 318
  - nazwa grupy węzłów, 318
  - Negate (grupa), 376
  - Noise Texture, 343
  - Object Info (wejściowy), 216
  - Object Mask (grupa), 136, 139, 445
  - podmiana, 322
  - podstawy obsługi, 207
  - podział wejść grupy, 320, 321
  - problem z zasłoniętym połączeniem, 211
  - problemy z wyświetlaniem w panelu, 208
  - przesuwanie, 208
  - Range From Min (grupa), 378
  - Range To Max (grupa), 377
  - shader Background, 327, 337, 347
  - shader Glass BSDF, 360
  - shader Glossy BSDF, 362
  - shader Transparent BSDF, 361
  - Sky Texture, 327
  - Stack Image (grupa), 379
  - sterowanie kształtem połączeń, 212
  - stworzenie nowej ramki, 210
  - Surface Output (grupa), 398
  - Texture Coordinate, 338, 341, 347, 383
  - tryby pracy w oknie Node Editor, 432
  - usuwanie, 208
  - usuwanie wejść grupy, 319
  - UV Fallback (grupa), 382
  - Voronoi Texture, 344
  - Wave Texture, 388
  - wejścia, 207, 209
    - wejściowe grupy, 317
    - wielokrotne użycie wejścia grupy, 319
    - właściwości, 211, 319
  - wprowadzenie, 207
  - wybór, 208
  - wyjęcie z ramki, 211
  - wyjścia, 207, 209
    - wyjściowe grupy, 317
  - wykorzystanie ramek, 210
  - wykorzystanie shadera Gloss Paint, 368
  - wyświetlanie schematu grupy, 321
  - wyświetlanie w panelu zestawu Material, 208

zmiana nazwy wejścia grupy, 320  
 zmiana tryb pracy, 210  
 wiatrochron  
   kabiny, szczegóły, 92  
 widoczność  
   linii pomocniczych, 423  
 widok  
   wyrównanie do obiektu, 226  
 wielobok sterujący  
   krzywą, 262  
 wierchołek  
   fazowanie, 294  
   obróć, 261  
   przesunięcie, 261  
   zaznaczanie wg nazwanej grupy, 278  
 wierchołki  
   łączenie w grupy, 278  
*Window Draw Method*  
   opcje z sekcji User Preferences  
     System, 185  
*Wire*  
   tryb wyświetlania siatki, 83  
 właściwości  
   ograniczenia obiektu, 237, 241, 243, 249, 254  
 wpisywanie  
   współrzędnych kursora, 193  
 wskazanie  
   koloru na ekranie, 204  
 wskaźnik  
   z tablicy przyrządów, modelowanie, 89  
   z tablicy przyrządów, montowanie, 89  
   z tablicy przyrządów, skala, 89  
   z tablicy przyrządów, tarcze, 90  
 współrzędne  
   obektu potomnego (w hierarchii), 245, 246  
 wybór  
   kości, 229  
   linii ruchu (F-Curve), 427  
   metody łączenia obszarów (GIMP), 162  
   pomniejszenie obszaru (GIMP), 159  
   powiększenie obszaru (GIMP), 159  
   rysowanie obszaru (GIMP), 160  
   skok do, 239, 269, 271  
   węzłów, 208  
   wg barwy (GIMP), 158  
   wszystkich obiektów podrzędnych, 76  
   zachowanie aktualnego obszaru (GIMP), 162  
   zaznaczanie obiektu i kości, 232  
 wyginanie

siatki, 288, 292  
 wygładzenie  
   siatki, 299  
 wytłoczenie  
   metoda rozbudowy krzywej, 261  
   nowej kości, 230

## X

### X

pole panelu Background Images, 192, 198  
*X.Ground*  
*Intensity*  
   intensywność barw podłoża, 456, 457  
*Opacity*  
   przejrzystość cieni, 456  
   pomocniczy obiekt (powierzchni lotniska), 149, 456  
*X.Pattern Adaptor*  
   pomocnicza grupa węzłów, 393  
   przykład użycia, 400  
   tekstura ‘ogólnego szumu’, 393  
*X.Random Dirt*  
*Displacement*  
   punkt wyjścia, 392  
*Fac*  
   punkt wyjścia, 392  
   pomocnicza grupa węzłów, 392  
   przykład użycia, 393, 399  
*Scale*  
   parametr tekstury, 392  
   tekstura ‘ogólnego szumu’, 392  
*X.Random Dit*  
*Directed Noise*  
   parametr tekstury, 392  
*X.Textured Skin*  
   modyfikacja połyskliwości, 394

## Y

### Y

pole panelu Background Images, 192, 198

## Z

zaokrąglanie  
   wybranych krawędzi siatki, 294  
 zapisanie  
   plików tymczasowych, 189  
   ustawień Blendera, 180  
 zapisywanie  
   konfiguracji, 180, 187, 188  
 zarządzanie

scenami, 203  
układem okien na ekranie, 199  
zastąpienie  
konfiguracji Blendera, 188  
zaznaczanie  
nazwanej grupy wierzchołków, 278  
zenit  
wyjaśnienie pojęcia, 331

zmiana  
współrzędnych obiektu (Inkscape), 174  
zmienna środowiskowa  
utworzenie, 22  
zmiana wartości, 22  
*Zoom*  
polecenie w Node Editor, 208



## Słownik

**GPL**, licencja — [General Public Licence](#), udostępniająca produkt za darmo wszystkim odbiorcom. Licencja ta jest sformułowana w ten sposób, aby uniemożliwić komercyjne rozpowszechnianie produktu.

**heksadecymalna**, notacja liczb — sposób notacji liczb, oparty na kolejnych potęgach liczby 16, a nie 10 (zapis decymalny). Liczby od 1 do 16 są w nim zapisywane jako: 0,1,...9,A,B,C,D,E,F. W zapisie heksadecymalnym (szesnastkowym) „12” =  $1 \cdot 16 + 2$ , czyli decymalnie 18. Podobnie liczba 4F to  $4 \cdot 16 + 15 = 79$  decymalnie. Zapis szesnastkowy jest szeroko stosowany przy tworzeniu oprogramowania.

**materiał** — to w Blenderze zestaw cech, używanych przy nanoszeniu obiektu na ostateczny obraz (renderowaniu). Cechy materiału to: barwa, połyskliwość, tekstura, i dziesiątki innych parametrów. Z materiałem może być związanych wiele tekstur (nierówności, barwy), zmieniających "ogólne" właściwości materiału dla pojedynczych pikseli renderowanego obrazu.

**rastrowy**, obraz — popularna metoda reprezentacji, polegająca na zapisie obrazu jako zbioru kolorowych lub czarno-białych punktów (pikseli). Tak jest np. wyświetlany obraz na ekran telewizyjnym, monitorze komputera. Wadą reprezentacji rastrowej jest pogorszenie jakości obrazu w dużych powiększeniach — gdy zaczynają być wyraźnie widoczne pojedyncze piksele. Zaletą jest stosunkowo prosty algorytm wyświetlania zawartości. Istnieje wiele różnorodnych sposobów zapisu (formatów) przechowywania obrazów rastrowych w plikach. Najpopularniejsze to [\\*.jpg](#), [\\*.png](#), [\\*.bmp](#), [\\*.tif](#).

**renderowanie** (ang. [rendering](#)) — w tej książce oznacza tworzenie ostatecznego obrazu (lub sekwencji obrazów — animacji) przygotowanej wcześniej trójwymiarowej sceny. W pierwszych wersjach *Autodesk 3D Studio* (początek lat 90-tych XX wieku) było przetłumaczone na polski jako [powlekanie](#), ale ta nazwa się nie przyjęła.

**shader** (ang.) — model matematyczny, stosowany do wyznaczania sposobu odbicia światła przez powierzchnię. W Cycles shadery są reprezentowane przez pojedyncze węzły lub grupy węzłów.

**tekstura** — obraz (zazwyczaj rastrowy), nakładany na trójwymiarową powierzchnię. Stosowany w grafice komputerowej do "urealniania" obiektów (nieregularności barw, napisy, itp.). Zastosowanie tekstur wykracza poza odwzorowanie barw — są stosowane także do zamodelowania drobnych nierówności powierzchni obiektów (tzw. [bump maps](#)).

**True Type** — sposób zapisu wzorów czcionek, wykorzystywanych w grafice komputerowej. Metoda opracowana w latach osiemdziesiątych przez Adobe. Czcionki True Type opisują kształt znaków w sposób wektorowy — za pomocą wypełnionych obszarów, ograniczonych za pomocą linii łamanych i krzywych Beziera. Dzięki temu wyglądają poprawnie na wydruku i na ekranie — nawet w dużym powiększeniu.

**wektorowy**, obraz — metoda reprezentacji, polegająca na zapisie obrazu jako zbioru kolorowych linii i obszarów. Każdy element obrazu ma określone współrzędne i kształt (prosta, łuk, koło, krzywa Beziera, ...). W ten sposób są zapisywane stworzone na komputerze rysunki techniczne. Zaletą reprezentacji wektorowej jest zachowanie dokładności przy dowolnym powiększeniu. Wadą jest złożony algorytm wyświetlania (gdyż oznacza przekształcenie na postać rastrową). Jednym ze sposobów zapisu danych wektorowych jest format [\\*.svg](#), stosowany m.in. przez Inkscape.

## Bibliografia

### Publikacje

- [1] Ton Roosendaal, Stefano Selleri, *Blender 2.3 — Oficjalny podręcznik*, Helion, 2005.
- [2] Kamil Kukło, Jarosław Kolmaga, *Blender — Kompendium*, Helion, 2007.
- [3] Marek Ryś, *Curtiss P-40 cz.1*, AJ Press, 2000 ("Monografie Lotnicze", nr 64).
- [4] Zbigniew Kolacha, Marek Ryś, *Curtiss P-40 cz.2*, AJ Press, 2000 ("Monografie Lotnicze", nr 65).
- [5] Krzysztof Janowicz, *Curtiss P-40 cz.3*, AJ Press, 2000 ("Monografie Lotnicze", nr 66).
- [6] Krzysztof Janowicz, Leszek A. Wieliczko, *Curtiss P-40 vol.1*, Kagero, 2007.
- [7] Krzysztof Janowicz, *Curtiss P-40 vol.2*, Kagero, 2009.
- [8] Leszek A. Wieliczko, Tom Żmuda, *Curtiss P-40D/E*, Kagero, 2008.
- [9] Francis H. Dean, *America's Hundred Thousand — The US Production Fighter Aircraft of World War II*, Shiffer Military History, 1997.
- [10] Francis H. Dean, Dan Hagedorn *Curtiss Fighter Aircraft — A Photographic History 1917-1948*, Shiffer Military History, 2007.
- [11] Ernest R. McDowell, *Curtiss P-40 in action*, Squadron/Signal Publications, 1976.
- [12] Paweł Sembart, *Kittyhawk I/IA*, ROSSAGRAP, 2006 (Model Detail Photo Monograph nr 14).
- [13] Anis Elbeid, Daniel Laureult, *P-40 Curtiss From 1939 to 1945*, Histoire & Collections, 2002.
- [14] Brett Green, *Modelling the P-40*, Osprey Publishing, 2005 (Osprey Modelling nr 15).
- [15] Władysław Niestoj, *Profil modeli latających*, Wydawnictwo Komunikacji i Łączności, 1980.
- [16] Estman N. Jacobs, Kenneth E. Ward, Robert M. Pinkerton, *The Characteristics of 78 related airfoil sections...*, NACA report nr 460, 1937.
- [17] Jerzy Bukowski, Wiesław Łucjanek, *Napęd śmigłowy — teoria i konstrukcja*, Wydawnictwo MON, 1986
- [18] Denis Zorin et al, *Subdivision for Modeling and Animation*, SIGGRAPH 2000 Course Notes, 2000
- [19] Gerald Farin, *Curves and Surfaces for CAGD*, Academic Press, 1997
- [20] Justin Peatross, Michael Ware, *Physics of Light and Optics*, Brigham Young University, 2011

### Internet

- [1] <http://www.p40warhawk.com>
- [2] <http://www.blender.org>
- [3] <http://www.gimp.org>
- [4] <http://www.inkscape.org>
- [5] <http://www.python.org>
- [6] <http://www.simmerpaintshop.com>
- [7] <http://www.freewebs.com/p40-tomahawk/> — Hume Bates, *Long Nose Hawks*, (artykuł), dostępny w wersji PDF pod adresem [http://downloads.hyperscale.com/longnosehawks\\_download.html](http://downloads.hyperscale.com/longnosehawks_download.html) (opublikowany w lipcu 2008)
- [8] <http://www.ipmsstockholm.org>, m.in. Martin Waligorski: *Interior Colours of US Aircraft, 1941-45* (opublikowany w lutym 2004).
- [9] <http://www.ratomodeling.com> — artykuł o malowaniu P-40 z AVG.
- [10] <http://wiki.blender.org> — dokumentacja funkcji Blendera, i nie tylko!
- [11] <http://www.graphics.cornell.edu/~westin/misc/fresnel.html> — opis efektu Fresnela

## Szczegółowe zasady udostępnienia tej publikacji

### Licencja



#### I. Postanowienia ogólne.

Utwór (zdefiniowany w punkcie II. e. Licencji) jest udostępniony na warunkach niniejszej publicznej licencji (Creative Commons Public License, „CCPL” lub „Licencja”). Utwór udostępniany na warunkach niniejszej licencji podlega ochronie na podstawie [Ustawy z 4 lutego 1994 r. o prawie autorskim i prawach pokrewnych \(Dz.U.00.80.904.\)](#). Zabronione jest jakiegokolwiek korzystanie z Utworu, na warunkach innych niż określone w niniejszej licencji lub prawie autorskim.

Każdy kto w jakikolwiek sposób skorzysta z uprawnień objętych niniejszą licencją (Licencjobiorca), w momencie wykorzystania oświadcza w sposób dorozumiany, że akceptuje wszystkie postanowienia licencji.

#### II. Definicje.

**„Utwór Zbiorowy”** – w rozumieniu niniejszej licencji oznacza utwór, taki jak wydanie periodyczne, antologia lub encyklopedia, stanowiący całość składającą się z odrębnych i niezależnych utworów. Dla celów niniejszej licencji utwór spełniający cechy Utworu Zbiorowego nie jest uważany za Utwór Zależny.

**„Utwór Zależny”** – w rozumieniu niniejszej licencji oznacza utwór powstały na bazie Utworu, lub też Utworu i innych chronionych prawem autorskim utworów, taki jak tłumaczenie, reprodukcja artystyczna, streszczenie, skrócona wersja, lub jakakolwiek inna forma w której następuje przerobienie, przekształcenie lub adaptacja Utworu, za wyjątkiem utworu który spełnia cechy Utworu Zbiorowego i nie jest, dla celów licencji, uważany za Utwór Zależny. Dla uniknięcia wątpliwości w przypadku utworu muzycznego synchronizacja Utworu z obrazami wizualnymi prowadzi do powstania Utworu Zależnego w rozumieniu niniejszej licencji.

**„Licencjodawca”** – oznacza osobę fizyczną lub prawną, która udostępnia Utwór Pierwotny na podstawie niniejszej licencji.

**„Twórca”** – oznacza osobę fizyczną, która stworzyła Utwór lub osobę prawną, na której rzecz powstały autorskie prawa majątkowe w sposób pierwotny.

**„Utwór”** – oznacza chroniony prawem autorskim utwór udostępniany na podstawie niniejszej licencji.

**„Licencjobiorca”** – oznacza osobę fizyczną lub prawną korzystającą z uprawnień określonych niniejszą licencją, która nie naruszyła uprzednio warunków niniejszej licencji w odniesieniu do Utworu, lub która mimo uprzedniego naruszenia uzyskała wyraźną zgodę autora na wykonywanie uprawnień przyznanych licencją.

**„Uniform Resource Identifier” (URI)** – oznacza internetowy adres zbioru informacji na temat zarządzania prawami w rozumieniu art. 6 pkt 12 [Ustawy o prawie autorskim i prawach pokrewnych](#), określającego warunki korzystania z Utworu zawarte w niniejszej licencji.

#### III. Dozwolony użytek.

Żadne postanowienie niniejszej licencji nie zmierza do ograniczenia, wyłączenia lub zawężenia uprawnień wynikających z dozwolonego użytku, wyczerpania prawa lub innych ograniczeń praw wyłącznych podmiotu tych praw, przysługujących mu w myśl prawa autorskiego lub innych znajdujących zastosowanie ustaw.

#### IV. Uprawnienia.

Licencjodawca udziela niniejszej licencji, która jest licencją nieodpłatną, niewyłączną, na czas nieoznaczony, której stosowanie nie jest ograniczone terytorialnie, do korzystania z Utworu w następujący sposób:

Zwielokrotnianie Utworu, włączanie Utworu do jednego lub więcej Utworów Zbiorowych, zwielokrotnianie Utworu jako części Utworu Zbiorowego,

Rozpowszechnianie zwielokrotnionych egzemplarzy lub nośników, na których Utwór jest zwielokrotniony, publiczne odtwarzanie, publiczne odtwarzanie drogą cyfrową Utworu oraz Utworu jako części Utworu Zbiorowego.

Powyższe uprawnienia Licencjodawca może wykonywać na wszystkich rodzajach nośników, we wszystkich rodzajach środków przekazu, oraz we wszystkich aktualnie znanych formatach. Powyższe uprawnienia obejmują także uprawnienie do dokonywania drobnych, technicznych modyfikacji Utworu koniecznych do dostosowania wykonywania uprawnień do różnych środków przekazu, nośników lub formatów. Wszystkie prawa wprost nie objęte licencją uważa się za nieudzielone, w szczególności te prawa, które określone są w punkcie V. 4. i V. 5. Licencji.

## V. Ograniczenia.

Uprawnienia przyznane w punkcie IV. licencji podlegają następującym ograniczeniom:

Licencjodawca może rozpowszechniać, publicznie odtwarzać Utwór, pod warunkiem dołączenia kopii niniejszej licencji, lub też [Uniform Resource Identifier](#) niniejszej licencji, do każdego egzemplarza Utworu, rozpowszechnianego przez Licencjodawcę, także w razie publicznego odtwarzania i publicznego odtwarzania drogą cyfrową. Licencjodawca nie może oferować żadnych uprawnień do Utworu ani też narzucać żadnych ograniczeń, które zmieniają lub ograniczają warunki zawarte w niniejszej licencji.

Licencjodawca nie może udzielać sublicencji.

Licencjodawca nie może zmieniać lub usuwać oznaczeń Utworu odnoszących się do niniejszej licencji oraz zawartej w niej klauzuli uchylenia się od odpowiedzialności.

Licencjodawca nie może rozpowszechniać, publicznie odtwarzać, publicznie odtwarzać drogą cyfrową Utworu przy użyciu środków technicznych pozwalających kontrolować dostęp lub użycie Utworu, w sposób sprzeczny z postanowieniami niniejszej licencji.

Powyższe zastrzeżenia mają także zastosowanie do Utworu jako części Utworu Zbiorowego, przy czym nie jest wymagane aby Utwór Zbiorowy niezależnie od Utworu był przedmiotem niniejszej licencji.

W przypadku tworzenia Utworu Zbiorowego Licencjodawca, na żądanie jakiegokolwiek podmiotu uprawnionego, zobowiązany jest usunąć wszystkie możliwe do usunięcia oznaczenia, o których mowa w pkt. V. 3., zgodnie z takim żądaniem.

Licencjodawca nie może wykonywać żadnych udzielonych w punkcie IV. Licencji uprawnień w jakikolwiek sposób, który skierowany jest na uzyskanie korzyści majątkowej lub pieniężnego wynagrodzenia.

Wymiana Utworu na inne chronione prawem autorskim utwory przez wymianę cyfrową plików, lub też w inny sposób, nie jest w rozumieniu niniejszej licencji sposobem, który skierowany jest na uzyskanie korzyści majątkowej lub pieniężnego wynagrodzenia, tylko i wyłącznie wtedy, gdy w związku z wymianą utworów nie pobiera się żadnego wynagrodzenia.

W przypadku rozpowszechniania, publicznego odtwarzania, publicznego odtwarzania drogą cyfrową Utworu, lub Utworu Zależnego, lub Utworu Zbiorowego, Licencjodawca zobowiązany jest zachować w stanie nienaruszonym wszystkie oznaczenia związane z prawno-autorską ochroną Utworu oraz zapewnić, stosownie do możliwości używanego nośnika lub środka przekazu, oznaczenie:

- nazwiska Twórcy Utworu;
- nazw innych podmiotów (np. instytucji sponsorującej, wydawcy), jeśli Twórca Utworu lub Licencjodawca wskazali w oznaczeniach związanych z prawno-autorską ochroną Utworu lub w warunkach korzystania z Utworu lub w inny przyjęty sposób, te podmioty jako uprawnione;
- tytułu Utworu;
- w niezbędnym możliwym zakresie, [Uniform Resource Identifier](#) odsyłającego do oznaczeń związanych z prawno-autorską ochroną Utworu;



Wszystkie oznaczenia powinny być uwidocznione w jakikolwiek możliwy sposób, przy czym co najmniej w tych wszystkich miejscach, gdzie uwidocznione są inne porównywalne prawno-autorskie oznaczenia i w sposób co najmniej tak wyraźny jak inne porównywalne prawno-autorskie oznaczenia.

#### **VI. Klauzula uchylenia się od odpowiedzialności.**

JEŻELI STRONY NIE POSTANOWIĄ W ODREBNYM PISEMNYM POROZUMIENIU INACZEJ, LICENCJODAWCA UDOSTĘPNIĄ UTWÓR W TAKIEJ FORMIE W JAKIEJ ZAPOZNAŁ SIĘ Z NIM LICENCJOBIORCA, I NIE UDZIELA ŻADNYCH DODATKOWYCH RĘKOJMI ANI GWARANCJI, WYRAŻNYCH LUB DOROZUMIANYCH, W TYM, W SZCZEGÓLNOŚCI DOTYCZĄCYCH TYTUŁU, WYKORZYSTANIA HANDLOWEGO, PRZEZNACZENIA UTWORU DO KONKRETNEGO CELU, CO DO TEGO ŻE NIE NARUSZA ON PRAW INNYCH OSÓB, BRAKU JAWNYCH LUB UKRYTYCH WAD, DOKŁADNOŚCI, WYSTĘPOWANIA LUB NIEWYSTĘPOWANIA WAD ZAUWAŻALNYCH, JAK I NIEZAUWAŻALNYCH.

#### **VII. Ograniczenie odpowiedzialności.**

W ŻADNYM WYPADKU LICENCJODAWCA NIE ODPOWIADA KONTRAKTOWO ANI DELIKTOWO WOBEC LICENCJOBIORCY, Z WYŁĄCZENIEM ODPOWIEDZIALNOŚCI WYNIKAJĄCEJ Z PRAWA MAJĄCEGO ZASTOSOWANIE, ZA ŻADNE SPECJALNE, PRZYPADKOWE LUB NASTĘPCZE SZKODY WYNIKŁE Z ZASTOSOWANIA NINIEJSZEJ LICENCJI LUB WYKORZYSTANIA UTWORU, NAWET JEŻELI LICENCJODAWCA ZOSTAŁ POWIADOMIONY O MOŻLIWOŚCI WYSTĄPIENIA TAKICH SZKÓD.

#### **VIII. Wygaśnięcie stosunku umownego.**

Licencja automatycznie wygasa w przypadku jakiegokolwiek naruszenia jej warunków przez Licencjobiorcę. Licencje osób fizycznych lub prawnych, które otrzymały od Licencjobiorcy Utwór Zbiorowy nie wygasają w takim przypadku, o ile osoby te nie naruszają postanowień tych licencji. Punkty II, III, VI, VII, VIII i IX obowiązują niezależnie od wygaśnięcia niniejszej licencji.

Zgodnie z powyższymi postanowieniami licencja udzielana jest na czas nieoznaczony. Niezależnie od tego Licencjodawca zachowuje prawo do udostępnienia Utworu na innych warunkach, lub do zaprzestania udostępniania Utworu, z tym jednak zastrzeżeniem, że taka decyzja Licencjodawcy nie może służyć obchodzeniu niniejszej licencji (lub też innej licencji udzielonej na podstawie niniejszej licencji), która raz udzielona Licencjobiorcy w pełnym wymiarze obowiązuje dopóki nie nastąpi jej wygaśnięcie w okolicznościach opisanych w punkcie VIII. 1.

#### **IX. Postanowienia końcowe.**

Za każdym razem gdy Licencjobiorca rozpowszechnia lub publicznie odtwarza drogą cyfrową Utwór lub Utwór Zbiorowy, Licencjodawca jednocześnie oferuje odbiorcom licencję dla Utworu na takich samych warunkach jakie uzyskał Licencjobiorca niniejszą licencją.

Jeśli jakiegokolwiek postanowienie niniejszej licencji jest nieważne lub niewykonalne w świetle obowiązującego prawa, jego nieważność lub niewykonalność nie wpływa na pozostałą część licencji. W przypadku braku dodatkowego porozumienia między stronami dotyczącego postanowień nieważnych lub niewykonalnych, powinny być one rozumiane, przy jak najmniejszym odejściu od ich literalnego brzmienia, zgodnie z odpowiadającymi im przepisami prawa.

Żadnego z postanowień niniejszej licencji nie uznaje się za uchylone, a żadnego naruszenia nie uznaje się za zaakceptowane, dopóki druga strona nie uzna pisemnie takiego uchylenia lub też nie wyrazi na piśmie następczego zezwolenia na naruszenie.

Niniejsza licencja zawiera całość postanowień pomiędzy stronami dotyczących udostępnianego na jej podstawie Utworu. Wszystkie nie ujęte w tej licencji postanowienia i porozumienia dotyczące uprawnień wskazanych w licencji uznaje się za nieistniejące. Licencjodawcy nie wiążą żadne dodatkowe postanowienia podane mu do wiadomości przez Licencjobiorcę. Licencji nie wolno modyfikować bez obopólnej pisemnej zgody stron.

Creative Commons nie jest stroną niniejszej Licencji i nie udziela żadnych gwarancji co do Utworu. Creative Commons nie ponosi odpowiedzialności kontraktowej ani deliktowej wobec Licencjobiorcy lub innej strony umowy, za żadne szkody, w szczególności bez ograniczeń za szkody specjalne, przypadkowe lub następce związane z niniejszą Licencją. Powyższe dwa zdania nie znajdują zastosowania, jeżeli Creative Commons wskazało się samo wyraźnie jako Licencjodawca w niniejszej Licencji, gdyż Creative Commons przysługują wówczas wszystkie uprawnienia i obowiązki Licencjodawcy.

Z wyjątkiem ograniczonego celu jakim jest publiczne wskazanie, że Utwór został licencjonowany na podstawie niniejszej Licencji, żadna ze stron nie będzie wykorzystywać znaku towarowego „Creative Commons”, ani żadnego innego związanego znaku towarowego, ani logo Creative Commons bez uprzedniej pisemnej zgody Creative Commons. Każde wykorzystanie, za zezwoleniem, musi być zgodne z aktualnymi wskazówkami korzystania ze znaku towarowego Creative Commons, jakie są publikowane na jego stronie internetowej lub w inny sposób regularnie udostępniane na wniosek.

Z Creative Commons można skontaktować się pod adresem <http://creativecommons.org/>.

W XX wieku można było robić miniatury samolotów z kartonu i z plastiku. Na początku wieku XXI przyszedł czas na jeszcze jedną odmianę tego hobby: modele komputerowe. To nowa gałąź modelarstwa redukcyjnego rodziła się „po cichutku”. Mało kto dziesięć lat temu mógł wydać (prywatnie!) kilka tysięcy dolarów na odpowiedni program. A tymi „nieodpowiednimi” nie było się co chwalić... Nieliczni zapaleńcy siedzieli więc przy monitorach i tworzyli pierwsze modele, ucząc się na własnych błędach.

Teraz to się może zmienić, bo cały potrzebny software jest dostępny za darmo (na licencji GPL). Jeżeli więc kupisz swój komputer po 2005r., to masz już wszystko, czego potrzeba, by zacząć! Ta książka dostarcza niezbędnego *know how*. Pokaże Ci, krok po kroku, jak zrobić takie modele, jak na ilustracjach obok. I, podobnie jak oprogramowanie, także jest bezpłatna!

„Wirtualne modelarstwo” jest kierowane nie tylko do modelarzy. Może zainteresować wszystkich tych, którzy chcą się poznać takie ogólnodostępne programy, jak Blender, GIMP, czy Inkscape.



ISBN: 978-83-931754-9-9

Bezpłatna publikacja elektroniczna, udostępniana na licencji *Creative Commons*



Samolot z 3-go dywizjonu AVG („Hell's Angels”), Kunming 1942, wg opracowania Janusza Świątłonia. (Tło: © Tomo Yun, [www.yunphoto.net](http://www.yunphoto.net))