Rozdział 16. Blender — edytor UV (UV/Image Editor)

Edytor UV jest specjalnym narzędziem, przeznaczonym do przypisywania wierzchołkom siatki położenia na powierzchni tekstury. Nazwa pochodzi od nazw współrzędnych teksturowania. Edytor UV przydaje się wyłącznie dla tych siatek, na które tekstury są nakładane z wykorzystaniem tzw. "mapowania UV" (*Shading:Material buttons*, panel *Map Input*).

Drugim trybem pracy edytora jest modyfikacja obrazu tekstury (*Texture Painting*). Zestaw narzędzi, dostępnych do tego celu w Blenderze, nie dorównuje wyspecjalizowanym edytorom obrazów, jak GIMP. Tym niemniej jest wystarczający do wykonania jakichś mniejszych czy większych poprawek (por. str. 933, 951).

16.1 Pojęcia podstawowe

Obrazy rastrowe w grafice komputerowej są zazwyczaj prostokątne (preferowane są kwadratowe). Gdy taki kwadrat (Rysunek 16.1.1b) chcesz nałożyć na, powiedzmy, kulę (Rysunek 16.1.1a), deformacja obrazu jest nieunikniona (Rysunek 16.1.1c):



Rysunek 16.1.1 Nakładanie prostokątnej tekstury na kulę

A czy można zobaczyć na obrazie tekstury obszary, które zajmują ściany kuli (zamapowane tak, jak pokazuje to Rysunek 16.1.1c)? Oczywiście! Przedstawia je Rysunek 16.1.2:



Rysunek 16.1.2 Przypisanie ścian do powierzchni tekstury - rozwinięcie w przestrzeni UV.

Do powierzchni tekstury zostały przypisany układ współrzędnych. Jego osie, dla odróżnienia od współrzędnych wierzchołków w przestrzeni, otrzymały nazwy U (pozioma) i V (pionowa) . Zazwyczaj przyjmuje się, że każda z tych współrzędnych ma wartość 0 w lewym, dolnym narożniku obrazu. Na krańcu tekstury U lub V przyjmują wartość = 1.0 (por. Rysunek 16.1.2). (Czasami podaje się je także w innych jednostkach — pikselach obrazu.) Dwuwymiarowa przestrzeń, w której jest rozciągnięta tekstura, nosi nazwę przestrzeni UV. Rozwinięcie ścian powierzchni siatki w tej przestrzeni nazywa się rozwinięciem UV. W naszym przykładzie topologia rozwiniętych ścian siatki nie ulega zmianie, za wyjątkiem krawędzi "szwów" (por. Rysunek 16.1.2).

Typowym zagadnieniem, z którym się będziesz musiał zmierzyć, jest takie ułożenie ścian siatki w przestrzeni UV, by nałożony na model obraz tekstury nie był zdeformowany. W Blenderze służy do tego specjalne okno *edytora UV (UV Editor)* . Służy do rozwinięcia, a następnie zmiany położenia ścian siatki w przestrzeni tekstury. Rysunek 16.1.3 przedstawia rezultat użycia: poprawione rozwinięcie siatki, którą pokazywał Rysunek 16.1.2. Większość jej obszaru została przekształcona w prostokąt. Pozwoliło to zachować regularny układ prostokątów tekstury i ich kątów na większości powierzchni kuli, oraz ciągłość wzoru na "szwie":



Rysunek 16.1.3 Przykład innego rozłożenia ścian na teksturze

Rozwinięcia UV w Blenderze mogą nie zachowywać topologii oryginalnej siatki. Świadczy o tym już sam fakt istnienia "szwu", wzdłuż którego siatka z przykładu została "rozcięta" i rozwinięta. W istocie w Blenderze każda ściana przechowuje "swoje" współrzędne wierzchołków UV, które mogą być zupełnie różne od ścian z którymi sąsiaduje w przestrzeni okna *3D View* (Rysunek 16.1.4, por. także str. 935):



Rysunek 16.1.4 Rozwinięcie UV w Blenderze traktuje każdą ścianę jako osobną całość

Blender domyślnie utrzymuje topologiczną ciągłość siatki, automatycznie włączając do selekcji wszystkie wierzchołki UV o tych samych współrzędnych. Rysunek 16.1.4 pokazuje jednak, że ten tryb można wyłączyć. Czasami takie efekty się przydają.

Siatka kuli w układzie "południki — równoleżniki" (jak w naszym przykładzie), jest właściwie powierzchnią, na którą nie można nałożyć kwadratowej tekstury bez jakiejś deformacji. Można się tylko starać, by odkształcenia obrazu były niewielkie (w końcu nad podobnym zagadnieniem biedziły się pokolenia kartografów). Większość powierzchni, z którymi będziesz miał do czynienia, nie jest aż tak wymagająca. Edytor UV i trochę dobrych pomysłów na rozwinięcie pozwala Ci uniknąć wyraźnych deformacji obrazu.

16.2 Włączenie okna edytora UV

Edytor to odrębny rodzaj okna — *UV/Image Editor* — które można wybrać z listy z lewej strony nagłówka (Rysunek 16.2.1):



Rysunek 16.2.1 Przejście do edytora UV/obrazu (UV/Image Editor) - edycji siatki w przestrzeni UV

Aby efektywnie pracować z rozwinięciem UV, Blender musi być w trybie *Edit Mode* (por. str. 854). Okno Edytora UV jest ściśle powiązane z edycją siatki w oknie widoku 3D. (Obydwa modyfikują różne właściwości wierzchołków i ścian tej samej siatki.) Dlatego warto zawczasu ustawić obydwa okna obok siebie (jak to pokazuje Rysunek 16.2.1).

Konwencje i klawisze skrótu, obowiązujące w edytorze UV, są bardzo zbliżone do tych, którymi posługujesz się w oknie *3D View*. Tak jak w tamtej przestrzeni istnieje kursor 3D (*3D Cursor* — por. 770), tak tu znajdziesz jego dwuwymiarowy odpowiednik. Te same kombinacje klawiszy powodują selekcję, zmianę widoku, czy podstawowe operacje edycji (przesunięcie, obrót, zmianę skali). Nagłówek okna *UV/Image Editor* liczbą kontrolek dorównuje niemal oknu *3D View* (Rysunek 16.2.2) :



Rysunek 16.2.2 Okno edytora UV: dostępne kontrolki

Zwróć szczególną uwagę na kontrolkę [¥], pokazaną na obrazie nagłówka. Za jej pomocą możesz przełączyć się na tryb pracy z pojedynczymi ścianami rozwinięcia. (Masz do dyspozycji także dwa inne tryby.) Kontrolka nie

jest widoczna, gdy włączysz synchronizację selekcji z oknem 3D View:



Rysunek 16.2.3 UV/Image Editor - tryb "malowania" (Texture Painting włączony)

W trybie "malowania" z nagłówka okna zniknęła większość kontrolek. Dostęp do zmiany aktualnego pędzla (*brush*), barwy i funkcji malowania umożliwia okno *Image Paint*. Możesz je otworzyć skrótem klawiatury: **C**, lub poleceniem *View*→*Paint Tool...*.

16.3 Zaznaczanie elementów siatki

W edytorze UV możesz używać do selekcji tych samych metod i skrótów klawiatury, co w oknie *3D View*, w trybie edycji (zob. str. 855-857) . Dokładnie takie samo działanie uzyskasz, gdy włączysz w nagłówku okna *UV/Image Editor* tryb "synchronicznej selekcji" (Rysunek 16.3.1):



Rysunek 16.3.1 Selekcja w trybie "synchronizacji" (z widokiem 3D View)

W tym trybie obowiązuje zasada dokładnej równoważności — gdy coś zaznaczysz w edytorze UV, staje się także zaznaczone w oknie *3D View*, i odwrotnie. Nagłówek *UV/Image Editor* zawiera tylko opcję wyboru wierzchołków lub ścian (por. Rysunek 16.3.1). W odróżnieniu od edytora siatki nie ma tu oddzielnego trybu wyświetlania/wyboru krawędzi. Wszystkie skróty — np. Att - PPM — działają tak samo, jak w oknie *3D View*.

Gdy wyłączysz tryb synchronizacji — w edytorze UV widzisz (i możesz manipulować) wyłącznie ścianami, które są zaznaczone na siatce w oknie *3D View* (Rysunek 16.3.2):



Rysunek 16.3.2 Selekcja w trybie bez "synchronizacji"

Jeżeli chcesz przez cały czas widzieć rozwinięcie wszystkich ścian siatki, niezależnie od aktualnej selekcji

 włącz tryb synchronizacji z 3D View.

Wewnętrznie Blender przechowuje oddzielnie współrzędne UV każdej ściany. Oznacza to, że jednemu wierzchołkowi siatki XYZ może odpowiadać tyle punktów w przestrzeniu UV, ile ścian się w nim łączy (najczęściej dwie, trzy, lub cztery — Rysunek 16.3.3):



Rysunek 16.3.3 Koncepcyjny schemat ścian siatki w rozwinięciu UV - każda ma własne współrzędne

Zazwyczaj, gdy wskażesz jeden z wierzchołków UV, metoda wybrana na liście *Sticky UV Selection* (Rysunek 16.3.2) powoduje automatyczne zaznaczenie ("zlepienie") pozostałych wierzchołków, leżących w tym samym miejscu. Możesz także wyłączyć ten tryb (*Sticky UV Selection:disable*). Bledner umożliwia wtedy dowolną zmianę topologii (wygodniej się wówczas przełączyć na zaznaczenie ścian) (Rysunek 16.3.4):



Rysunek 16.3.4 "Zlepianie" rozwinięcia UV wyłączone (*disabled*) — można manipulować oddzielnymi ścianami

Rysunek 16.3.5 i Rysunek 16.3.6 pokazują dwa dostępne tryb "zlepiania" wierzchołków UV — *Shared Location* i *Shared Vertex*. Zazwyczaj w czasie pracy będziesz używał trybu *Shared Location*:



Zaznaczenie tego wierzchołka spowoduje zaznaczenie pozostałych. Sticky UV Selection = Shared Vertex

Rysunek 16.3.5 "Zlepianie" wskazanych wierzchołków UV w trybie *Shared Location* — efekt taki, jak gdybyś zawsze wybierał małym obszarem prostokątnym

Rysunek 16.3.6 "Zlepianie" wskazanych wierzchołków UV w trybie *Shared Vertex* — automatycznie zaznacza wszystkie punkty UV, związane z tym samym punktem na siatce (XYZ).

Gdy synchronizacja selekcji jest wyłączona, obok ikon zaznaczania ścian lub wierzchołków, w nagłówku okna pojawia się trzecia możliwość: *UV Island* (Rysunek 16.3.7):



Rysunek 16.3.7 Szybkie zaznaczanie całych fragmentów siatki w trybie wyboru UV Island.

Przy włączonym *UV Island* jedno kliknięcie **PPM** zaznacza wszystkie ściany, które w sposób bezpośredni lub pośredni są połączone z miejscem wskazanym przez kursor. Ten tryb jest przydatny np. do szybkiego zaznaczania i przesuwania drobnych fragmentów siatki.

16.4 Kursor 2D

Kursor 2D (2D Cursor) pełni w edytorze UV analogiczną rolę do kursora 3D w oknie 3D View (por. str. 770). I tam, i tu, wyznacza "aktualną pozycję", która jest używana jako punkt odniesienia podczas transformacji siatki (obrotu, zmiany skali).

Położenie kursora 2D jest oznaczane niewielkim krzyżem, otoczonym czerwono-białym rombem (Rysunek 16.4.1):



Rysunek 16.4.1 Kursor 2D w oknie UV

Kliknięcie **LPM** można traktować jako pierwszą, przybliżoną metodę wskazania położenia kursora. Z oczywistych przyczyn nie jest specjalnie dokładnie. W edytorze UV nie ma możliwości "dosunięcia" kursora do najbliższego węzła siatki pomocniczej (chociaż sama siatka jest widoczna). Gdy naciśniesz **Shift-S** (ten skrót nie ma w oknie *UV/Image Editor* odpowiednika w menu), przekonasz się że lista możliwości jest tu uboższa (Rysunek 16.4.2a):



Rysunek 16.4.2 Dosunięcie kursora do zaznaczonego wierzchołka

W edytorze UV kursor 2D można dosuwać (*Snap*) tylko do wybranych wierzchołków (*Cursor→Selection*) lub najbliższego piksela tekstury (*Cursor→Pixel*). Rysunek 16.4.2 przedstawia przykład użycia polecenia *Snap:Cursor→Selection*. Kursor został dosunięty do zaznaczonego wierzchołka. Mówiąc szczerze, tej funkcji używam częściej od pozostałych.

W *UV/Image Editor* istnieje także możliwość dokładnego wpisania współrzędnych kursora. Służy do tego polecenie *View → View Properties*. Gdy je wywołasz, zobaczysz na ekranie okno dialogowe *View Properties* (Rysunek 16.4.3):



Rysunek 16.4.3 Ustalanie położenia kursora 2D w oknie View Properties

Trochę niekonsekwentnie, autorzy tego okna podają nazwę współrzędnej poziomej jako X (to U), a pionowej — Y (to V). Obydwie wartości, jakie pokazuje Rysunek 16.4.3, są wyrażone w pikselach. Aby się przełączyć na klasyczny zakres współrzędnych UV — 0.0...1.0 (por. str. 930) — włącz opcję *Normalized Coords*.

16.5 Przesunięcie (Grab)

Zaznacz wierzchołki, które chcesz przesunąć (Rysunek 16.5.1):



Rysunek 16.5.1 Zaznaczenie wierzchołków

Następnie naciśnij **G** (*UVs→Transfrom→Grab/Move*). Przełączysz się w ten sposób w tryb przesuwania. Każde przesunięcie myszy będzie powodować przemieszczenie zaznaczonych wierzchołków. Dokonywane zmiany w kształcie siatki są pokazywane na bieżąco we wszystkich oknach Blendera (Rysunek 16.5.2):



Rysunek 16.5.2 Przesuwanie grupy wierzchołków

W nagłówku aktywnego okna jest wyświetlane aktualne przesunięcie. Aby zatwierdzić nowe położenie (zakończyć przesunięcie), należy kliknąć IPM. Aby zrezygnować z operacji, możesz w każdej chwili nacisnąć Esc. Podczas przesuwania możesz dodatkowo trzymać wciśnięty klawisz Ctrl. Spowoduje to skokową zmianę odległości (np. co 0.1 jednostki). Jeżeli będziesz trzymał wciśnięty klawisz Shift — przesunięcia na ekranie "wyhamują", pozwalając na dokładniejsze ustalenie pozycji. Wreszcie kombinacja obydwu — Shift-Ctrl — umożliwia skokowe przesuwanie położenia o niewielkie odległości (np. co 0.01 jednostki).

Jeżeli w trakcie przesuwania naciśniesz 🛛 lub 🍸 — ograniczysz swój ruch tylko do translacji wzdłuż jednej z tych osi.

16.6 Obrót

Zaznacz grupę wierzchołków, które chcesz obrócić, oraz ustal środek transformacji (*Pivot*) (Rysunek 16.6.1). Praktycznie stosuję jeden z dwóch rodzajów środka: *Bounding Box Center* albo *Cursor 2D*. W tym przypadku, środkiem obrotu będzie *Cursor 2D*. Możesz go ustawić z klawiatury — naciśnij . (kropka):



Rysunek 16.6.1 Zaznaczenie wierzchołków

Naciśnij ℝ (*UVs→Transfrom→Rotate*). Przełączysz się w ten sposób w tryb obrotu. Na rysunku pojawi się kreskowana linia od kursora myszy do środka transformacji. Każde przesunięcie myszy będzie powodować obrót zaznaczonych wierzchołków — proporcjonalnie do zmiany kąta kreskowanej linii. Dokonywane zmiany w kształcie siatki są pokazywane na bieżąco (Rysunek 16.6.2):





W nagłówku okna *UV/Image Editor* jest wyświetlany aktualny kąt obrotu. Aby zatwierdzić nowe położenie wierzchołków (zakończyć transformację), należy kliknąć LPM. Aby zrezygnować z operacji, możesz w każdej chwili nacisnąć Esc .

Podczas obrotu możesz trzymać wciśnięty klawisz **Ctrl**. Spowoduje to skokową zmianę proporcji (co 5^o). Jeżeli będziesz trzymał wciśnięty klawisz **Shift** — zmiana obrót na ekranie "wyhamuje", pozwalając na dokładniejsze ustalenie. Wreszcie kombinacja obydwu — **Shift-Ctrl** — umożliwia skokową zmianę obrotu o niewielkie kąty (np. co 0.5^o).

16.7 Skalowanie

Zaznacz grupę wierzchołków, które chcesz powiększyć/pomniejszyć, oraz ustal środek transformacji (z listy *Pivot* — Rysunek 16.7.1). Lista pozwala wskazać trzy rodzaje "środków": *Bounding Box Center, Median Point* oraz *2D Cursor. Median Point* to wypadkowy środek wszystkich zaznaczonych wierzchołków. (Leży w ich "środ-ku ciężkości"). Zacznijmy od przypadku, gdy środkiem skalowania jest właśnie *Median Point*. (Możesz go ustawić z klawiatury — **Shift**-,):



Rysunek 16.7.1 Zaznaczenie wierzchołków

Naciśnij S (*UVs→Transfrom→Scale*). Przełączysz się w ten sposób w tryb skalowania. Na rysunku pojawi się kreskowana linia od kursora myszy do środka transformacji. Każde przesunięcie myszy będzie powodować zmianę skali zaznaczonych wierzchołków — proporcjonalnie do zmiany długości kreskowanej linii. Dokonywane zmiany w kształcie siatki są pokazywane na bieżąco (Rysunek 16.7.2):



Rysunek 16.7.2 Powiększanie grupy wierzchołków

W nagłówku okna *UV/Image Editor* jest wyświetlana aktualna zmiana skali wzdłuż każdej osi. Aby zatwierdzić nowe położenie wierzchołków (zakończyć transformację), należy kliknąć **LPM**. Aby zrezygnować z operacji, możesz w każdej chwili nacisnąć **Esc**.

Podczas zmiany skali możesz dodatkowo trzymać wciśnięty klawisz **Ctrl**. Spowoduje to skokową zmianę proporcji (np. co 0.1). Jeżeli będziesz trzymał wciśnięty klawisz **Shift** — zmiana skali na ekranie "wyhamuje", pozwalając na dokładniejsze ustalenie. Wreszcie kombinacja obydwu — **Shift**-**Ctrl** — umożliwia skokową zmianę skali o niewielkie wartości (np. co 0.01).

Jeżeli w trakcie przesuwania naciśniesz \mathbf{X} lub \mathbf{Y} — ograniczysz zmianę skali do jednej z tych osi. W takim przypadku żaden rozmiar w kierunku poprzecznym nie ulegnie zmianie. Rysunek 16.7.3 pokazuje przykład skalowania wzdłuż jednej osi. Wybrałem tu także inny sposób określenia środka — przełączyłem typ punktu odniesienia (*Pivot*) na *2D Cursor* (na klawiaturze: .):



Rysunek 16.7.3 Zmiana skali względem kursora 2D - przygotowanie

Nacisnąłem **S**, **Y**. Trzymając wciśnięty klawisz **Ctrl** (aby uzyskać dokładną zmianę skali) możesz zmniejszyć odległość wierzchołków od środka skalowania nawet do zera (Rysunek 16.7.4):



Rysunek 16.7.4 Zmiana skali względem kursora 2D (i wzdłuż jednej osi - Y)

Pokazaną powyżej zmianę skali wzdłuż jednej osi można stosować jako sposób na dokładne i szybkie "spłaszczenie" grupy wierzchołków.

16.8 Przypinanie (*Pin*) i rozwijanie (*Unwrap*)

Do efektywnego przekształcania siatek UV będziesz używał dwóch poleceń, które wydają mi się nierozłączne jak... chyba jak młotek i gwoździe. Dlatego zdecydowałem się opisać je w tej samej sekcji. Pierwszym z nich jest przypinanie (*Pin*) wierzchołków siatki do określonych punktów UV. Sprawa jest bardzo prosta — zaznaczasz wybrane wierzchołki i naciskasz P (*UVs→Pin*, Rysunek 16.8.1):



Rysunek 16.8.1 Przypinanie punktów (Pin)

Po co przypinamy wierzchołki? Aby sterować rozwinięciem siatki. Punkty "przypięte" nie będą nigdy zmieniane przez jakikolwiek "automat". W przypadku punktów nie przypiętych — może być różnie. Sprawdź to sam: obróć górną część zaznaczonych wierzchołków tak, jak to pokazuje Rysunek 16.8.2a:



Rysunek 16.8.2 Wywoływanie powtórnego rozwinięcia siatki (Unwrap)

Następnie zaznacz całą siatkę i naciśnij **E** (*UVs →Unwrap*). Potwierdź jeszcze operację w okienku, które się pojawi (*OK?*). Rezultat pokazuje Rysunek 16.8.2b. Widzisz? Przemieszczenie "przyszpilonych" wierzchołków pociągnęło za sobą resztę siatki! Na drugim, dolnym obszarze, gdzie pozostawiliśmy je "w spokoju", nie nastąpiły żadne zmiany. Przed wywołaniem polecenia *Unwrap* trzeba było zaznaczyć na siatce (w oknie *3D View*!) obszar, który ma ulec zmianie. W przykładzie, który pokazuje Rysunek 16.8.2, zaznaczyliśmy po prostu całą siatkę. Oczywiście, może to być mniejszy fragment¹.

¹ Nie będąc pewnym, czy masz włączony tryb synchronizacji selekcji z oknem *3D View*, czy nie, zaproponowałem zaznaczenie obszaru, który spowoduje ten sam rezultat w obydwu przypadkach. Zwróć uwagę, że w trybie bez synchronizacji dla polecenia *Unwrap* obowiązuje prosta zasada: przekształcane są tylko te wierzchołki, które widzisz w oknie edytora UV. (Bo widzisz w nim tylko to, co jest zaznaczone w oknie *3D View* — por. str. 934, Rysunek 16.3.2)

Pomyśl, ile pracy wymagałoby takie uzyskanie takiego płynnego wygięcia siatki, jakie pokazuje Rysunek 16.8.2b, gdybyś spróbował tradycyjnie przesuwać lub obracać grupy wierzchołków! Najwięcej czasu poświęciłbyś właśnie na uzyskanie płynnego przejścia, pomiędzy tymi punktami, które przemieściłeś bardziej, a tymi, które przemieściłeś mniej. A tutaj — wystarczy wybrać jakiś rząd kluczowych wierzchołków, a reszta dopasowuje się sama!

Aby było jeszcze łatwiej, twórcy Blendera umieścili w menu *UVs* przełącznik *Live Unwrap Transform* (Rysunek 16.8.3). Gdy jest włączony, program dokonuje dynamicznego rozwinięcia siatki, gdy tylko przemieścisz jakikolwiek "przyszpilony" wierzchołek.

Aby zobaczyć, jak to działa: włącz ten tryb w menu. Na potrzeby tego przykładu zaznacz w oknie *3D View* całą siatkę, i wyłącz w oknie edytora UV tryb synchronizacji selekcji (por. Rysunek 16.8.3).



Rysunek 16.8.3 Włączenie dynamicznego rozwijania

Zaznacz teraz dolną część "przyszpilonej" krawędzi, i zacznij ją obracać (Rysunek 16.8.4):



Rysunek 16.8.4 Dynamiczne rozwijanie siatki w trakcie przemieszczania punktów (Live Unwrap Transform)

Dzięki *Live Unwrap Transform* możesz lepiej "wyczuć", jak zachowuje się rozwinięcie siatki. Oczywiście, zetkniesz się na pewno z bardzo złożonymi transformacjami, w których tryb "dynamiczny" tylko by przeszkadzał. Wtedy zawsze możesz go wyłączyć, i korzystać z prostego polecenia *Unwrap* (E).



16.9 Podstawienie obrazu

Oprócz rozwinięcia siatki, *UV/Image Editor* umożliwia edycję obrazu rastrowego. Zresztą — nie trzeba go od razu edytować. Częstym zastosowaniem jest po prostu podstawienie zawartości jakiegoś pliku pod rozwinięcie siatki w **UV**, wyłącznie w celach referencyjnych. (*UV/Image Editor* wyświetla jednocześnie i rozwinięcie siatki, i obraz, więc można jedno z drugim porównać).

Najpierw zaznacz całą siatkę w oknie *3D View*. (Obraz jest przypisywany do zaznaczonych ścian siatki — gdy ich nie zaznaczysz, nic się nie stanie). Najprostszą drogą podstawienia obrazu jest wybranie jednego z tych, które są już używane w tym pliku Blendera. Służy do tego lista w nagłówku okna (Rysunek 16.9.1):



Rysunek 16.9.1 Wybór obrazu rastrowego z listy

Jeżeli wybierzesz jakiś obraz, którego boki nie są równe (nie jest kwadratem), przygotuj się na małe zaskoczenie (Rysunek 16.9.2):



Rysunek 16.9.2 Rezultat podstawienia obrazu, który nie jest kwadratowy

Rozwinięcie **UV** siatki ulega zawsze rozciągnięciu, odpowiedniemu do proporcji boków obrazu. Dzieje się tak dlatego, że współrzędne UV wierzchołków są zawsze wartościami względnymi (0..1), jak to pokazuje Rysunek 16.9.1 i Rysunek 16.9.2. Ich pozycja na ekranie jest wyznaczana na podstawie rozmiarów obrazu tła!

Gdy usuniesz prostokątny obraz z tła (X), por. Rysunek 16.9.2), rozwinięcie siatki natychmiast wraca do właściwych proporcji (jak to pokazuje Rysunek 16.9.3a). Czy teraz już rozumiesz, dlaczego obrazy wszystkich tekstury są kwadratami?

Oprócz podstawiania jednego z załadowanych obrazów, możesz także wskazać jakikolwiek plik na Twoim dysku. Służy do tego polecenie *Image → Open…* (Rysunek 16.9.3a). Można także utworzyć nowy, pusty obraz na przykład po to, by za chwilę coś na nim namalować używając *Paint Tools* (por. str. 933, Rysunek 16.2.3). Służy do tego polecenie *Image → New…* (Alt-N — Rysunek 16.9.3a):



Rysunek 16.9.3 Utworzenie nowego obrazu (w tym przypadku jest to specjalny UV Test Grid)

W oknie dialogowym *New Image* ustal, jaką nazwę będzie nosił obraz (Rysunek 16.9.3b). Możesz tu także zmienić jego rozmiar (w pikselach) — w polach *Width* i *Height*. Nowy obraz będzie wypełniony jednolitą barwą, którą także możesz ustawić w tym oknie (por. Rysunek 16.9.3b). Jeżeli jednak włączysz opcję *UV Test Grid*, ta informacja jest ignorowana. Wybór *UV Test Grid* powoduje wygenerowanie obrazu zawierającego wzór kwadratów i krzyżyków, przypominający szachownicę (Rysunek 16.9.3c). To specjalny obraz, przeznaczony do wyszukiwania deformacji obrazu nałożonego na siatkę. Gdy otworzysz obok okno *3D View*, i przełączysz je w tryb wyświetlania *Textured*, zobaczysz o co chodzi (Rysunek 16.9.4):



Rysunek 16.9.4 Podgląd obrazu, nałożonego na siatkę w edytorze UV, w oknie 3D View.

Podczas edycji siatki w oknie 3D View, w trybie wyświetlania Textured zawsze zobaczysz obraz, aktualnie załadowany w UV/Image Editor.

946

16.10 Wyrównywanie (Align)

Wyrównanie oznacza w edytorze UV ustawienie wskazanych wierzchołków pionowo (*Align Y*) lub poziomo (*Align X*). Ten sam efekt możesz osiągnąć zmieniając skalę zaznaczonych wierzchołków wzdłuż jednej z osi do 0. *Align X/Y* jest właściwie "skrótem" takiej operacji.

Zaznacz jakąś linię wierzchołków, które chcesz wyrównać (Rysunek 16.10.1a). Następnie naciśnij **W** (lub *UVs→Weld/Align*), aby otworzyć menu *Weld/Align*. Z menu wybierz *Align Y* (bo wierzchołki mają leżeć na linii poziomej). Rezultat przedstawia Rysunek 16.10.1b:



Rysunek 16.10.1 Wyrównywanie linii wierzchołków.

Każde z poleceń *Align X/Y* zmienia tylko jedną ze współrzędnych wierzchołków — tę, którą ma w nazwie. Współrzędne X (U) punktów, które pokazuje Rysunek 16.10.1, nie uległy żadnej zmianie.

16.11 Zapisanie rozwinięcia UV do pliku

Obraz rozwinięcia UV można zapisać do pliku (rastrowego: *. *tga*, lub wektorowego: *.*svg*). Na początek proponuję wykorzystywać domyślną opcję — zapis do pliku rastrowego (w formacie TARGA — *.*tga*).

Aby zapisać rozwinięcie, wywołaj skrypt *Save UV Face Layout* (z menu *UVs →Scripts*) — jak pokazuje to Rysunek 16.11.1a:



Rysunek 16.11.1 Zapisanie rozwinięcia UV do pliku (rastrowego).

Pojawi się okno dialogowe *UV Image Export* (Rysunek 16.11.1b). Możesz w nim ustalić (w pikselach): grubość linii (*Wire*), oraz rozmiar boku obrazu (*Size*). (Obraz rozwinięcia jest zawsze kwadratowy.) Po naciśnięciu *OK*. jedno z okien Blendera zmieni się na chwilę w typowe okno *Save As*. Wskażesz w nim folder i nazwę pliku dla zapisywanego obrazu (Rysunek 16.11.1c).

Rezultat przedstawia Rysunek 16.11.2. Jest to obraz rastrowy, zapisanym w formacie **.tga*. Ma białe tło i czarne linie.

Jeżeli w oknie UV Image Export zaznaczyłeś opcję Object (Rysunek 16.11.1b), to do nazwy pliku, którą podałeś zostanie jeszcze dołączona nazwa obiektu, zawierającego rozwiniętą siatkę. Powiedzmy, że aktualny obiekt nazywa się **B.360.TailFillet**, a podczas zapisywania do pliku podałeś jako nazwę pliku **uv.tga** (Rysunek 16.11.1c). Plik z obrazem otrzyma w takim przypadku nazwę **uv_B.360.TailFillet.tga**. Gdybyś jednak nie włączał opcji *Object*, zostałby zapisany tak, jak podałeś — jako **uv.tga**.



Rysunek 16.11.2 Rastrowy obraz rozwinięcia UV

Wyeksportowane obrazy rozwinięcia UV są niezbędne do przygotowania tekstur modelu. Podstawą do tworzenia tekstur jest rysunek "technicznych" nierówności — krawędzi paneli, nitów, śrub — przygotowany w Inkscape. Jako "podkładkę" pod ten rysunek użyjemy rozwinięć UV, tym razem wyeksportowanych w postaci wektorowej (Rysunek 16.11.3) :



Rysunek 16.11.3 Zapisanie rozwinięcia UV do pliku (wektorowego).

Utworzony plik wczytaj do Inkscape (File →Import — str. 682) (Rysunek 16.11.4):





Wczytany plik jest w Inkscape grupą, składającą się z kwadratowej ramki i właściwego rozwinięcia (także grupy). Długość boku ramki jest taka, jak zadeklarowany rozmiar obrazu (2048). Dodatkowo powiększa ją o 1 (do 2049) grubość linii obrysu (1 jedn.). Stąd, aby umieścić rozwinięcie dokładnie we właściwym miejscu rysunku, wpisz we współrzędne X i Y środka wartości (-0.5, -0.5) (to połowa grubości ramki), a nie (0,0) (por. Rysunek 16.11.4). Po wczytaniu możesz także zmienić nazwę tej grupy — z "automatycznej" na jakąś opisową. Rysunek 16.11.4 pokazuje, w oknie *Object Properties*, o które pole chodzi: *Label*. Po wpisaniu nowej nazwy naciśnij przycisk *Set*. "Na oko" niczego to nie zmieni na rysunku (nawet nie zamknie okna *Object Properties*!). Ale za chwilę zobaczysz, że etykieta naprawdę uległa zmianie. Aha, pod żadnym pozorem nie zmieniaj pola *Id* — to unikalny numer tej grupy, zapamiętany w kilku miejscach rysunku. Po wczytaniu do Inkscape można wykonać jeszcze jedną, "kosmetyczną" czynność: zmniejszyć grubość linii rozwinięcia (Rysunek 16.11.5):



Rysunek 16.11.5 Zmiana grubości linii obrazu rozwinięcia UV w Inkscape.

Gdy klikniesz dwukrotnie w kontur rozwinięcia, Inkscape przeniesie Cię do wnętrza grupy. (Poznasz to po tym, że u dołu ekranu zamiast nazwy warstwy pojawi się nazwa grupy — Rysunek 16.11.5). Ten sam efekt można osiągnąć wybierając z menu kontekstowego polecenie *Enter group <nazwa>* (por. str. 719).

Wewnątrz grupy kliknij w obszar rozwinięcia (to także grupa!) i przejdź do jego ustawień *Fill and Stroke*. Zmień w nich grubość linii z 1 na np. 0.25. Dzięki temu rysunek stanie się dokładniejszy

W tej sekcji pokażę w skrócie drugą funkcję okna *UV/Image Editor*: możliwość edycji obrazów rastrowych. Są sytuacje, gdy ta funkcja bardzo się przydaje.

Działanie trybu *Texture Painting* (bo tak się ta funkcja nazywa) przedstawię na przykładzie realnego problemu, na jaki natkniesz zapewne się podczas odtwarzania jakiegoś "plamistego" kamuflażu. W moim przypadku są to zielone plamy, pokrywające górne i boczne powierzchnie samolotu z AVG (Rysunek 16.12.1):



Rysunek 16.12.1 Przygotowanie w GIMP kopii warstwy z kamuflażem dla Blendera (pomocniczy plik camuflage.png)

Przygotowałem w GIMP obraz kamuflażu samolotu. Poszczególne elementy barwne są umieszczone na różnych warstwach (Rysunek 16.12.1a). Na jednej z nich, o nazwie **Color-Green.Camuflage**, namalowałem zielone plamy, jakie pokrywały górne i boczne powierzchnie maszyny (Rysunek 16.12.1b). Na pewno granice tych plam będą przesunięte wzdłuż któregoś ze szwów modelu. Aby wykryć i poprawić takie miejsca, wykopiowuję zawartość tej warstwy do odrębnego pliku (*camuflage.png*).

Plik *camuflage.png* podstawiam (poleceniem *Image* →*Replace*) jako obraz **Test** (Rysunek 16.12.2). (Dlatego, że ten obraz został przypisany wcześniej do wszystkich siatek modelu — por. str. 346).



Rysunek 16.12.2 Przygotowanie obrazu plam kamuflażu w Blenderze

Zamiast przełączać widok *3D View* w tryb *Textured*, można włączyć widoczność obrazu tekstury w trybie *Solid* (wydaje mi się, że tak wygodniej pracować). Wystarczy nacisnąć **Shift-T** (wyłączenie tymi samymi klawiszami). Obraz *camuflage.png* ma białe tło. Dodałem je specjalnie w GIMP (por. Rysunek 16.12.1) dlatego, że Blender w *3D View* ignoruje obszary przejrzyste. Zabarwia je na czarno, a to utrudniłoby nam malowanie!

b) 🕙 a) Okno sterowania 2. Draw Soften Smear Clone wygląda Tak kursor pędzlami możesz myszki w trybie Texture + BR:White X Mix przywołać klawiszem C Painting. Rozmiar okręgu Niezgodność Airbrus 4.000 zależy od wybranego plam Opacity 1.0 i Wrap rozmiaru pędzla Size 20 F 18 Falloff 0.50 Ρ Spacing 10.1 P Add New \$ 1. Przełącz się Możesz także tryb malowania włączyć tryb Texture Paint 🗢 🌗 🕯 🗼 🔯 🗢 🗸 View Image 📝 🗢 IM:Test ÷, Paint Texture Paint X

Aby rozpocząć edycję obrazu, włącz w oknie UV/Image Editor przycisk Texture Painting (Rysunek 16.12.3):

Rysunek 16.12.3 Przełączenie w tryb Texture Paint

W tym trybie możesz otworzyć (*View →Paint Tool..*, albo po prostu **C**) okno przybornika (*Image Paint*). Możesz w nim zmieniać tryb pracy, barwę, oraz inne parametry aktualnego pędzla. Powiedzmy, że zamierzasz na razie usunąć nadmiar jakiejś plamy, więc wybierz kolor tła — biały¹ (Rysunek 16.12.3b). Zwróć uwagę na nieciągłość kamuflażu na wewnętrznej krawędzi lotki (Rysunek 16.12.3a). Skoryguj ją w oknie *3D View*, malując bezpośrednio po powierzchni modelu. Zaznacz więc (jeszcze w trybie *Object Mode*) lotkę, i przełącz tryb pracy na *Texture Paint* (Rysunek 16.12.3a).

W tym trybie "nanosisz farbę" naciskając LPM. Trochę to przypomina pracę aerografem — zamalowaniu ulegnie fragment modelu znajdujący się pod kursorem myszki (Rysunek 16.12.4):





• W trybie *Texture Paint* malujesz tylko po powierzchni obiektu aktywnego (ostatnio wybranego). Aby pomalować inną część modelu, musisz się przełączyć na chwilę w tryb *Object Mode* i ją zaznaczyć (str. 953).

¹ Gdy chcesz zmieniać teksturę wyłącznie w oknie *UV/Image Editor*, możesz pracować z obrazami o przeźroczystym tle. Usuwanie tła polega wówczas na malowaniu pędzlem przestawionym z domyślnego trybu *Mix* na *Erase Aalpha* (w oknie *Image Paint*). Można wtedy także wykorzystywać tryby *Soften*, *Smear*. Niestety, tak jak pisałem, zmiany nieprzejrzystości nie są widoczne na teksturze nałożonej na model w oknie *3D View*. Stąd, jeżeli chcesz malować bezpośrednio po powierzchni modelu, musisz użyć trybu *Mix* i "nieprzejrzystej" barwy. To właśnie z tego powodu tło obrazu *camuflage.png* jest białe.

Do dalszych korekt przyda się nam drugi, zielony pędzel. Zdefiniujemy go w oknie *Image Paint*. W oknie przybornika masz typową listę, za pomocą której (poleceniem *ADD NEW*) możesz stworzyć nowy pędzel (Rysunek 16.12.5a):



Rysunek 16.12.5 Dodawanie kolejnego pędzla (UV/Image Editor)

Nowy pędzel jest w pierwszym momencie dokładną kopią poprzedniego, którego używałeś przed chwilą. Zacznij od nadania mu nowej nazwy (Rysunek 16.12.5b). Potem kliknij w pole z barwą, by je zmienić. Spowoduje to otwarcie standardowego okna wyboru koloru (Rysunek 16.12.6a):



Rysunek 16.12.6 Zmiana barwy pędzla — poprzez wskazanie piksela na obrazie

Nie musisz pamiętać, jaki właściwie odcień mają plamy kamuflażu. Wystarczy, że naciśniesz przycisk *Sample*, a kursor zmieni się w "pipetkę". Możesz wskazać jej końcem miejsce obrazu o odpowiednim kolorze (Rysunek 16.12.5b). Wybrany kolor stanie się barwą aktualnego pędzla (Rysunek 16.12.5c).

Tej zielonej barwy potrzebujemy, by zamalować "ząbki", widoczne na skrzydle przy krawędzi lotki (Rysunek 16.12.7a). Ale to jest już inny obiekt: płat. Przełącz się na chwilę w tryb *Object Mode*:



Rysunek 16.12.7 Przełączenie trybów, by zaznaczyć do malowania inną siatkę

Wskaż płat jako nowy obiekt aktywny. Potem z powrotem włącz tryb Texture Paint i kontynuuj malowanie.

Przejrzyj w ten sposób wszelkie zakamarki modelu. W zależności od sytuacji, pewne plamy ogranicz barwą białą, inne powiększ kolorem zielonym. Aby nie wchodzić co chwila w okno zmiany barw, przygotuj sobie dwa pędzle: w kolorze tła (White) i w kolorze plam (Green).

Wszystkie zmiany, które naniosłeś na obraz *camuflage.png* w Blenderze, nie są jeszcze zapisane. Warto je co jakiś czas "utrwalić". Służy do tego polecenie *Image→Save*, z menu okna *UV/Image Editor* (Rysunek 16.12.8).

Gdy już uzgodniłeś cały kamuflaż i zapisałeś uaktualniony obraz na dysku, pora wczytać rezultaty z powrotem do GIMP. Załaduj plik *camuflage.png* jako nową warstwę¹.



Rysunek 16.12.8 Zapisanie aktualnego obrazu

Najpierw usuńmy z **Color-GreenCamuflage** nadmiary plam. Wczytaj do rysunku GIMP plik z dysku (poleceniem *File→Open As Layers*). Zostanie wczytany jako nowa warstwa, o nazwie **camuflage.png**. Umieść ją powyżej **Color-GreenCamuflage**, i przełącz jej tryb działania (*Mode*) z *Normal* na *Divide* (Rysunek 16.12.9a):



Rysunek 16.12.9 Usuwanie nadmiarów plam kamuflażu

Ten tryb spowoduje, że zielone pozostaną tylko te fragmenty warstwy **Color-GreenCamuflage**, których nie ma na **camuflage.png**. Część zgodna jest biała (Rysunek 16.12.9b). W ten sposób uzyskaliśmy dość kontrastowe zestawienie kolorów, ułatwiające dostrzeżenie różnic.

Wybierz na warstwie **camuflage.png** obszar plam (poleceniem **Select→By Color**). Następnie przełącz zaznaczenie na resztę obszaru (**Select→Invert**). Teraz przejdź na warstwę **Color-GreenCamuflage** i wymaż wszystko, co na niej widzisz zielonego (Rysunek 16.12.9c). Obszar zaznaczenia skutecznie ochroni to, co powinno pozostać nienaruszone.

¹ Teoretycznie możliwe byłoby zastąpienie plikiem **camuflage.png** oryginalnej warstwy **Color-GreenCamuflage**. Potem należałoby zaznaczyć i wymazać na niej całe białe tło. Próbowałem tej metody, ale okazuje się, że Blender nieładnie wyostrzył granice plam, oryginalnie wygładzonych przez pędzle GIMP-a. W efekcie trzeba było je z powrotem "zmiękczać" operacją *Smooth*. To sporo pracy. Przedstawiona tu metoda jest o wiele szybsza, bo poprawiasz za jej pomocą wyłącznie obszary różnic, a nie całą zawartość warstwy.

W podobny sposób domalujemy do plam brakujące fragmenty. Ponownie zaznacz obszary skorygowanych w Blenderze plam na warstwie **camuflage.png**. Następnie przełącz się na **Color-GreenCamuflage** (Rysunek 16.12.10a, b):



Rysunek 16.12.10 Domalowywanie brakujących fragmentów plam kamuflażu

(Warstwę **camuflage.png** możesz już wyłączyć, a nawet usunąć). Wybierz jako narzędzie pędzel, i ustal jego barwę na taką samą, jak zielony kolor plam. Rozmiar pędzla może być nawet bardzo duży. (W ten sposób możesz, nawet mimochodem, pokryć kolorem zielonym jakiś obszar, który powinien być zamalowany). Potem zamaluj wszystkie przejrzyste fragmenty zaznaczonego obszaru (Rysunek 16.12.10c).

Przeniosłeś w ten sposób wszystkie zmiany obrazu z Blendera do Gimpa. Zwróć uwagę, że dużym ułatwieniem było zastosowanie na warstwie **Colors-GreenCamuflage** tylko dwóch, jednolitych barw: tła i plam. Gdyby nie ta unifikacja, szybka selekcja obszarów "wg koloru" w GIMP nie byłaby możliwa. To bardzo utrudniłoby adaptację zmian, dokonanych w Blenderze.

Rozdział 17. Blender — pozostałe

Tu umieściłem wszystkie polecenia Blendera, które nie pasują do pozostałych rozdziałów

17.1 Zdefiniowanie nowego materiału

W tym przykładzie poznasz stworzenie najprostszego z możliwych materiałów¹. Nazwiemy go **Artifical** i będzie miał kolor czerwony. Materiału **Artifical** będziemy używać do wszystkich obiektów pomocniczych, które nie mają się pojawić na ostatecznym obrazie. (Intensywny kolor czerwony pozwoli nam je łatwiej dostrzec w *Object Mode*).

Zacznijmy od sytuacji wyjściowej: mamy obiekt (a właściwie — jego siatkę: Rysunek 17.1.1). Jest to obiekt pomocniczy, który nie ma się pojawić na jakimkolwiek renderingu. Chcielibyśmy teraz przypisać mu jakiś kontrastowy kolor — na przykład czerwony:



Rysunek 17.1.1 Obiekt (a właściwie — siatka) bez przypisanego materiału

Aby utworzyć nowy materiał, zaznacz obiekt i przejdź do zestawu *Shading* (F5), podzestawu *Material buttons*. Tam, w panelu *Links and Pipeline*, wybierz przycisk *Add New* (Rysunek 17.1.2):





W konfiguracji, jaką wybraliśmy (por. Rysunek 13.2.2, str. 758 — okno *User Preferences*, przełącznik *Material linked to:*) materiał jest przypisany do siatki, a nie do obiektu. Stąd w panelu *Links and Pipeline* widzisz pole z nazwą aktualnej siatki oraz licznikiem materiałów, które zostały przez nią użyte. Blender umożliwia przypisanie fragmentom pojedynczej siatki do 16 różnych materiałów (por. str. 797). Więcej informacji na temat różnic pomiędzy przypisaniem materiału do obiektu i siatki znajdziesz na końcu tej sekcji (str. 959). (Czasami przypisanie do obiektu także się przydaje!)

¹ Słowo "materiał" w grafice komputerowej nie całkiem odpowiada popularnemu znaczeniu — patrz słownik na str. 1105.

Po naciśnięciu przycisku *Add New* powstały nowy materiał, a w zestawie pojawiły się nowe panele (Rysunek 17.1.3):

📑 🔻 Panels @ 🗟 🔍 📜 📓	€ 2 2 3 3 3 3 4 1 >	Domyślna nazwa
Preview Podgląd na aktualny wygląd materiału	VCol Light VCol Paint Te Kontrolki, ustala- No Mist Env jace barwe	Links and Pipelin Link to Object MA:Materia K F Nodes E:B.310.T.Cowling OB ME < 1 Mat 1
By otworzyć okno zmiany koloru, kliknij tu (LPM)	Col R 0.800 Image: Col R 0.800 Image: Col Image: Col	Render Pipeline Halo ZTransp Zoffs: 0.00 > Full Osa Wire Strands ZInvert Radio OnlyCast Traceable Shadburf

Rysunek 17.1.3 Panele podzestawu Material Buttons, bezpośrednio po utworzeniu nowego materiału

Nowo utworzony materiał domyślnie ma barwę jasnoszarą (wszystkie komponenty barwy — R, G, B — mają wartość = 0.8). Nosi także nic nie mówiącą nazwę **Material**. Teraz to zmienimy.

Kliknij **Lem** w panelu *Material* w prostokąt z barwą (Rysunek 17.1.3), by otworzyć okno wyboru koloru (Rysunek 17.1.4):



Rysunek 17.1.4 Zmiana barwy i nazwy materiału

W oknie wyboru koloru możesz wybrać barwę z palety, lub ustawić dokładną wartość numeryczną. W tym przypadku chcemy uzyskać kolor czerwony, więc ustaw *R*ed na 1.0, *G*reen i *B*lue na 0.0 (Rysunek 17.1.4). Zmień także nazwę materiału na **Artifical**. Rysunek 17.1.5 przedstawia siatkę przypisaną do materiału, który stworzyliśmy:



Rysunek 17.1.5 Efekt zastosowania materiału Artifical — zmiana barwy obiektu na czerwoną

Raz utworzony materiał można przypisać wielu obiektom/siatkom. Służy do tego lista rozwijalna z lewej strony nazwy materiału (w panelu *Links and Pipeline* — Rysunek 17.1.5).

W Blenderze materiał może być przypisany do obiektu, lub do jego siatki. Zazwyczaj stosuje się przypisania do siatki, gdyż umożliwiają m.in. jednoczesne użycie wielu różnych materiałów (zob. str. 899). Gdy zmienisz coś w współdzielonej siatce, przypisanej do wielu obiektów, zmiana pojawi się we wszystkich jej "klonach" (por. str. 797). Popatrz na przykład, na Rysunek 6.11.13 ze str. 406. Wszystkie żebra skrzydła używają tam tej samej siatki, więc na wszystkich, w wyniku dodania do materiału nowej tekstury, pojawiły się otwory.

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 17.1.6a) i Rysunek 17.1.7a). Każde okucie składa się z dwóch takich blaszek. Wszystkie blaszki używają tej same 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 najczęściej występowały. Dopóki obiekt ma w panelu *Links and Pipeline* włączoną opcję *ME* (Rysunek 17.1.6b), dopóty "dziedziczy" materiał swojej siatki:





W innym miejscu — prowadnicy cięgna, przymocowanej do wręgi kadłuba (Rysunek 17.1.7a) — te same blaszki powinny być "pomalowane" na kolor reszty szkieletu: **B.Inner.***. Czy można ten efekt osiągnąć bez powielania siatki? Tak, wystarczy w panelu *Links and Pipeline* przełączyć się z *ME* na *OB*, i wybrać odpowiedni materiał (Rysunek 17.1.7b):



Rysunek 17.1.7 Materiał przypisany do obiektu

Co właściwie ten przełącznik *OB* robi? Informuje Blender, że dla tej konkretnej części, należy użyć materiału przypisanego do obiektu (*OB*), a nie do siatki (*ME*). Zwróć uwagę, że po lewej stronie tego przełącznika widać teraz nazwę obiektu, do którego jest przypisany materiał **B.Inner.Details** (Rysunek 17.1.7b).

17.2 Materiały wielobarwne (*Ramps*)

Materiał nie musi mieć stałej barwy. Jego kolor może się zmieniać w zależności od kąta oświetlenia poszczególnych ścian siatki. Dobrym przykładem tej właściwości są powierzchnie metalowe. Aby powłoka wyglądała bardziej "metalowo", przejdź do panelu *Ramps* i naciśnij przycisk *Colorband* (Rysunek 17.2.1):



Rysunek 17.2.1 Panel Ramps - zróżnicowanie barwy materiału w zależności od kąta padania światła

Tytułowy "pasek kolorów" (*colorband*) to spektrum barwy, która zmienia się w zależności od kierunku oświetlenia obiektu (Rysunek 17.2.1). Z lewej strony paska (*Pos* = 0.000) znajduje się barwa materiału na ścianach ustawionej prostopadle do źródła światła. W środku (*Pos* = 0.500) znajduje się barwa ścian równoległych do kierunku światła, a z prawej (*Pos* = 1.000) — ustawionych tyłem (nieoświetlonych).

Rysunek 17.2.2 przedstawia funkcje poszczególnych kontrolek panelu Ramps:



Rysunek 17.2.2 Kontrolki panelu Ramps

Barwy, ustalone na pasku spekturm, są potem "mieszane" (wg jednej z wybranych *Method*) z podstawową barwą materiału. Zwróć uwagę, że możesz za pomocą wartości *Factor* płynnie regulować intensywność tego efektu (Rysunek 17.2.2, u dołu panelu).

Za pomocą przycisku *Add* dodałem do paska barw kolejną "pozycję", pośrodku. Następnie zmieniłem każdą z nich, ustawiając ich barwy tak, jak pokazuje to Rysunek 17.15.3:

V Preview	Material Ramps	
Pos=0.0 Szary 42%	% Show Col Ramp Pos=0.4 ec Ramp Pos=1.0 Szary 70% Szary 1	0 0%
	Colorband Add Cur: 2 Del Ease 👄	
· · · · · · · · · · · · · · · · · · ·		
	✓ Pos 0.750 ► R 0.100 ■/	
Kompletna	A 1.000 B 0.100	
nieprzejrzystość	Input Method Factor	
0	Shader = Mix = 1.000	

Rysunek 17.2.3 Ustawienie spektrum kolorów materiału

W efekcie materiał na podglądzie uzyskał "metaliczny połysk".

17.3 Zdefiniowanie nowej tekstury (z obrazu rastrowego)

W tym przykładzie poznasz stworzenie typowej tekstury¹ — na podstawie wskazanego pliku z obrazem rastrowym. Przejdź do zestawu *Shading* (F5), podzestawu *Texture buttons*. Zwróć uwagę na panel *Preview*. Znajduje się tam przełącznik, którym wybierasz, do czego zostanie przypisana nowa tekstura (Rysunek 17.3.1) . Zazwyczaj będziesz wybierał przycisk *Mat*, oznaczający aktualny materiał. Czasami możesz się przełączyć na *World*, oznaczający użycie tekstury jako panoramy otoczenia.

Aby dodać nową teksturę, w panelu Texture, naciśnij przycisk Add New (Rysunek 17.3.1):



Rysunek 17.3.1 Naciśnięcie Add New - utworzenie nowej tekstury, związanej z aktualnym materiałem/otoczeniem

Spowoduje to pojawienie się w panelach nowych kontrolek. W miejscu przycisku *Add New* pojawiło się pole z nazwą tekstury. Domyślnie to **Tex**, ale od razu zmień ją na coś bardziej odpowiedniego.

Kolejnym krokiem jest określenie typu tekstury: z listy rozwijalnej wybierz Image (Rysunek 17.3.2):



Rysunek 17.3.2 Określenie rodzaju tekstury — w tym przypadku Image.

Spowoduje to pojawienie się w zestawie kolejnych paneli: *Map Image* (na razie nic na nim nie zmieniamy) i *Image*, w którym wskażemy plik obrazu.

¹ Słowo "tekstura" w grafice komputerowej nie całkiem odpowiada popularnemu znaczeniu — patrz słownik na str. 1105

W panelu *Image* naciśnij przycisk *Load*. Spowoduje to otwarcie (zapewne w jednym z obszarów *3D View*) okna wyboru plików. Należy wskazać w nim odpowiedni obraz i nacisnąć *SELECT IMAGE* (Rysunek 17.3.3):



Rysunek 17.3.3 Załadowanie pliku z obrazem tekstury.

W efekcie, w panelu *Image* pojawią się nowe kontrolki, a na podglądzie (panel *Preview*) — obraz załadowanej tekstury.

Domyślnie Blender nie kopiuje do pliku ze sceną zawartości obrazu¹. Zapamiętuje tylko ścieżkę, skąd ma pobrać plik. Wynika stąd kilka praktycznych wskazówek:

- Do wskazania położenia takich zewnętrznych danych zawsze używaj ścieżek względnych (por. Rysunek 17.3.3). Dzięki temu, gdy np. przesuniesz folder z plikiem .blend i plikami tekstur w nowe miejsce wszystko będzie dalej się normalnie renderować.
- Możesz zmieniać zawartość obrazu tekstury za pomocą zewnętrznego narzędzia (np. GIMP), bez konieczności zamykania Blendera. Potem wystarczy w panelu Image nacisnąć przycisk *Reload* (por. Rysunek 17.3.3) — i uaktualnisz obraz używany w Blenderze.

¹ Możesz go do tego zmusić, poleceniem *File->External data ->Pack into .blend file*.

17.4 Zdefiniowanie nowej tekstury proceduralnej

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.

Przejdź do zestawu *Shading* (F5), podzestawu *Texture buttons*. Aby dodać nową teksturę, w panelu *Texture*, naciśnij przycisk *Add New* (Rysunek 17.4.1):



Rysunek 17.4.1 Naciśnięcie Add New — utworzenie nowej tekstury, związanej z aktualnym materiałem

Nadaj nowo utworzonej teksturze nazwę (Rysunek 17.4.1), po czym z listy **Texture Type** wybierz któryś z dostępnych rodzajów tekstur (wszystkie są teksturami proceduralnymi, poza *Plugin* i *Image*). Powiedzmy, że wybieramy teksturę *Clouds* (Rysunek 17.4.2):



Rysunek 17.4.2 Określenie rodzaju tekstury — w tym przypadku Clouds, i ustalenie parametrów

W panelu *Preview* zobaczysz podgląd wygenerowanego obrazu, a w dodatkowym panelu, który się pojawi z prawej — parametry wybranej tekstury (Rysunek 17.4.2). W przypadku tekstur "abstrakcyjnych", takich jak *Clouds*, możesz zmienić m.in. rodzaj (funkcję matematyczną) generowanego wzoru (pole *Noise Basis*). Możesz także sterować rozmiarem generowanych zaburzeń (pole *Noise Size*). W ten sposób można uzyskać setki różnorodnych obrazów!
17.5 Materiał — powielenie przypisania tekstury

Często przypisanie tekstury do materiału zawiera wiele szczegółowych parametrów mapowania. Gdy chcesz dodać do materiału kolejną teksturę, wpisywanie ich wszystkich w panele *Map Input* i *Map To* zajęłoby dłuższą chwilę. Na szczęście istnieje w Blenderze szybsza metoda (Rysunek 17.5.1):



Rysunek 17.5.1 Skopiowanie mapowania tekstury

Zaznacz najpierw w panelu *Texture* (zestaw *Shading:Material buttons*) przypisanie tekstury, które chcesz powielić. Następnie naciśnij przycisk (kopiowanie ustawień "do bufora"). Potem zaznacz inny, pusty "slot tekstury" (Rysunek 17.5.1) i naciśnij przycisk (kopiowanie ustawień "z bufora"). W zaznaczonym slocie pojawi się kopia przypisania tekstury. (Masz teraz tę samą teksturę przypisaną dwa razy — w zależności od sytuacji może to np. dwukrotnie wzmocnić jakiś efekt.)

Zazwyczaj jednak nie chodzi o skopiowanie istniejącej tekstury, tylko, np. podstawienie nowej, bardzo podobnej. Aby to osiągnąć, przejdź do zestawu *Shading:Texture buttons*, i stwórz lokalną kopię tekstury (Rysunek 17.5.2):



Rysunek 17.5.2 Stworzenie nowej tekstury z duplikatu istniejącej

Powielenie tekstury przebiega tak samo, jak każdego innego bloku danych — poprzez kliknięcie w "licznik użycia", umieszczony przy nazwie (Rysunek 17.5.2a, więcej na ten temat — str. 565). Potem można jej zmienić nazwę i — na przykład — przypisanie obrazu (Rysunek 17.5.2b).

17.6 Nazwy materiałów i tekstur

Zazwyczaj w skład modelu wchodzi o wiele mniej materiałów, tekstur niż powłok. Stąd przy nazewnictwie tych elementów nie ma potrzeby stosowania członu z unikalnym numerem. Wystarczą same nazwy.

<u>W przypadku materiałów</u> proponuję stosować tekst składający się z następujących segmentów (rozdzielonych kropkami):

- 1 znak: segment, który będzie pełnił rolę wyróżnika wersji samolotu. Wszystkie części w pojedynczym pliku Blendera będą używać takiej samej litery. Na przykład w modelu P-40B nazwy będą zaczynać się od "B";
- segment z podstawową, "ludzką" nazwą materiału. Dwóm podstawowym materiałom, używanym do pokrycia samolotu, nadałem nazwę Skin. (To dlatego, że imitują różnorodne pokrycia, i nie można im przypisać jakiejś jednej "fizycznej materii".) W pozostałych przypadkach stosujemy nazwy "fizyczne" — "Glass" (np. osłony kabiny), "Rubber" (np. na oponach), "Leather" (np. zagłówka pilota), "Chrome" (dla gładzi cylindrów amortyzatorów). W przypadku imitacji pokrycia farbą w jednolitym kolorze, można stosować nazwę koloru tej farby — np. "Black".
- opcjonalny segment z członem "uszczegóławiającym". W przypadku podstawowych materiałów poszycia (nazwa w pierwszym segmencie — "Skin") ten segment zawiera niezbędne określenia: Inner lub Camuflage. W pozostałych przypadkach to potoczna nazwa zespołu, do którego należy stosować materiał (np. Canopy dla oszklenia kabiny).

Zgodnie z powyższymi regułami, materiał pokrywający zwenętrzne powierzchnie samolotu będzie nosił nazwę **B.Skin.Camuflage**, a materiał osłony kabiny — **B.Glass.Canopy**.

<u>W przypadku tekstur</u> proponuję także stosować tekst składający się z następujących segmentów (rozdzielonych kropkami):

- **1 znak**: segment, który będzie pełnił rolę wyróżnika wersji samolotu. (Analogicznie jak w nazwie materiału — w tym przypadku to **B**);
- segment powtarzający nazwę "rodziny" materiałów, w których tekstura jest używana. Na przykład u tekstur poszycia samolotu ten człon nazywa się **Skin**, tak samo jak u materiałów powierzchni;
- segment określający rolę tekstury. Może to być symbol modulowanej wartości materiału Color, Nor, Ref, Spec,.... Można także użyć bardziej potocznych określeń, np. Holes.

Zgodnie z powyższymi regułami, nadałem podstawowej teksturze nierówności poszycia nazwę **B.Skin.Nor-Detalis**, a teksturze otworów — **B.Skin.Holes**.

• W przypadku materiałów i tekstur związanych z konkretną sceną, a nie z modelem (np. trawa lotniska) nie stosuję żadnych 1-znakowych przedrostków.

17.7 Ustalenie koloru tła sceny

Aby ustalić barwę tła sceny, przejdź do zestawu *Shading* (F5), podzestawu *World buttons*. W panelu *Preview*, włącz przełączniki *Blend* i *Real*. W podglądzie, umieszczonym obok, zobaczysz uzyskany dwubarwny obraz (Rysunek 17.7.1):



Rysunek 17.7.1 Panele sterujące barwą tła sceny

Przełącznik *Blend* tworzy płynne przejście od koloru *Ho*ryzontu do koloru *Ze*nitu. Przełącznik *Real* dodaje "symetryczne odbicie" koloru *Ze*nitu u dołu (w nadirze). Dodatkowo opcja *Real* powoduje, że horyzont będzie się "ustawiał" odpowiednio do kamery.



Rysunek 17.7.2 Efekt zastosowania tła Blend & Real

17.8 Ustawienie światła otoczenia

Gdy rozejrzysz się wokół, zauważysz, że zazwyczaj cienie nie są zupełnie czarne. To dzięki światłu, odbitemu od całego otoczenia. Póki co, niewiele programów *raytrace* jest w stanie wyznaczyć takie piąte czy szóste kolejne odbicia (trwało by to zbyt długo). Zamiast tego, światło otoczenia jest przybliżane za pomocą dodatkowego rozjaśnienia sceny. W Blenderze to rozjaśnienie jest określane jako kolor *Ambient* (otoczenia).

Panel sterowania efektem znajduje się w zestawie przycisków *Shading* (F5), podzestawie *World buttons*. Zacznijmy od włączenia "głównego efektu". W panelu *Amb Occ* włącz przełącznik *Ambient Occlusion*, aby go uaktywnić. (Rysunek 17.8.1):

	zest	aw Shading			* 🕘 🖬 🕹	0	podzestaw World buttons
		Mist / Stars	s / Physic	Amb (Doc		
P							
	Samples: 5 🕨 Raytrace 🗢				działanie efektu		
	Max Dist: 10.00 🕨 Constant QMC 🗢						
	Use Falloff						
		Add	S	ub	Both		
Intensywność efektu		Plain	Sky	Color	Sky Texture		
	1.	Energy:1.0					

Rysunek 17.8.1 Zawartość panelu Amb Occ

Nie wdając się w szczegóły działania każdego z licznych przycisków panelu *Amb Occ*: użyj takich ustawień, jakie pokazuje Rysunek 17.8.1. Zwróć tylko uwagę na odpowiednie dobranie intensywności efektu (pole *Energy*). *Ambient Occlusion* można traktować jako specyficzny rodzaj światła otaczającego. Jeżeli użyłbyś je z pełną energią — efekt byłby zbyt silny (Rysunek 17.8.2):



Rysunek 17.8.2 Efekt Ambient Occlusion o dużej intensywności (Energy = 1.0)

Bardziej poprawny rezultat uzyskasz dla niezbyt dużych wartości parametru *Energy* — 0.2..0.4 (Rysunek 17.8.3):



Rysunek 17.8.3 Efekt Ambient Occlusion o mniejszej intensywności (Energy = 0.3)

Cienie w takich zagłębieniach, jak wloty powietrza czy połączenie usterzenia z kadłubem, choć bardziej poprawne, nadal wydają się zbyt ciemne (Rysunek 17.8.3). Aby sobie z tym poradzić, w panelu *World* zmień barwę *Amb* z czarnego na ciemnoszary (14%). (To barwa zaciemnień, wywoływanych przez *Ambient Occlusion*). Jednocześnie zmień współczynnik *Exp* na 0.4, a *Range* na 2.0 (Rysunek 17.8.4):

	🗢 Panels 🕝 🖾 🔾 🕻) 🕺 🤣 🏈	< 1 →				
DI	▼ World			Mist / Stars	/ Physics	Amb C	Dec	
Р		XF			Ambient O	cclusion	1	
				Samples: S	5 🕨	Raytrac	e	\$
	HoR 0.732	ZeR 0.022	kolor Amb na	🔹 Max Dist: 10	.00 🕨	Constar	nt QMC	+
	HoG 0.802	ZeG 0.109 14% s	zarości	Use Fallot	Ť			
	HoB 0.986	ZeB 0.400 📃						
		· · · · · · · · · · · · · · · · · · ·	Zmień <i>Exp</i> z domyślnego 0.2	Add	Sul	n	Both	
	AmbR 0.140		2 domyoinogo o.2		04	•	Boun	=
	AmbG 0.14	Ехр 0.40 🛛 💻		Plain	Sky C	olor	Sky Textur	e
	AmbB 0.140	Range 2.00		Energy:0.30				
			Zmień R z domyśl	ange nego 1.0 na 2.0				

Rysunek 17.8.4 Rozjaśnienie koloru cieni Ambient Occlusion (kolor Amb)

(Samo rozjaśnienie barwy *Amb* powoduje rozjaśnienie całego samolotu. Do kompensacji tego efektu służą właśnie parametry *Range* i *Exp*. Zwiększanie pierwszego wybiórczo zaciemnia fragmenty modelu, a drugiego — rozjaśnia inne.)



Rysunek 17.8.5 pokazuje efekt rozjaśnienia barwy Amb oraz kompensacji rozłożenia zaciemnień:

Rysunek 17.8.5 Model po skorygowaniu zbyt głębokich cieni

Krawędzie klap i sterów prawie całkiem zanikły — trudno, coś za coś. Poprawimy ich widoczność później, za pomocą tekstur. Za to cienie w zagłębieniach i we wnętrzu wlotów można jeszcze trochę poprawić. Włącz w panelu *Amb Occ* opcję *Use Falloff* (Rysunek 17.8.6):





971

17.9 Zmiana właściwości źródła światła (Lamp)

Po zaznaczeniu źródła światła, przejdź do zestawu *Shading* (F5). Pierwsze dwa panele — *Preview* i *Lamp* — mają identyczną zawartość dla wszystkich rodzajów źródeł (Rysunek 17.9.1):



Rysunek 17.9.1 Panele, wspólne dla wszystkich źródeł światła

W panelu *Preview* większość miejsca zajmuje obraz z podglądem światła. Pokazano tu efekt, jaki wywoła to oświetlenie na białej "ścianie". Możesz tu, oprócz podglądu, zmienić typ światła. Wystarczy kliknąć w jeden z dostępnych przełączników (*Lamp, Area, Spot, Sun, Hemi*).

W panelu Lamp umieszczono właściwości wspólne dla wszystkich źródeł światła (Rysunek 17.9.1). Możesz tu:

- zmienić nazwę tego źródła (nie myl tego z nazwą obiektu: tę możesz zmienić w zestawie Object F7);
- ustalić intensywność (Energy);
- określić dystans (zasięg), dla którego energia światła zmniejszy się o połowę (Dist);
- ustalić barwę światła;

W tym podręczniku użyte zostały dwa rodzaje źródeł światła:

- Sun to oświetlenie podstawowe, imitujące równoległe promienie słońca. W wewnętrznym rendererze Blendera przedmioty oświetlone tym światłem mogą rzucać cień (Rysunek 17.9.2);
- Hemi to oświetlenie uzupełniające: emitująca światło półsfera (Hemi od Hemisfera). Odpowiada za rozjaśnienie tych fragmentów obrazu, do których nie dochodzi światło słońca. To oświetlenie nie rzuca cieni (Rysunek 17.9.3).





Rysunek 17.9.2 Symbol źródła światła równoległego (Sun)

Rysunek 17.9.3 Symbol półkulistego źródła światła (Hemi)

Symbol źródła światła jest niezależny od aktualnego powiększenia. Towarzysza mu dwie linie. Linia kreskowana ilustruje "zasięg" światła — a dokładniej dystans "połowicznego zaniku". Linia ciągła to odległość od płaszczyzny XY. Pod względem parametrów do ustawienia *Hemi* jest prostym źródłem. Nie ma żadnych dodatkowych parametrów, tylko te wspólne, z panelu *Lamp*.

O wiele bardziej złożone jest źródło *Sun*. Rysunek 17.9.4a pokazuje dodatkowe dwa panele: do sterowania cieniami (*Shadow and Spot*) oraz efektami atmosferycznymi (*Sky/Atmosphere*).



Rysunek 17.9.4 Dodatkowe parametry cieni światła Sun

Źródło, którego światło wywołuje cienie (przynajmniej w wewnętrznym rendererze Blendera) jest oznaczane dodatkową otoczką z linii kreskowanej (Rysunek 17.9.4b)

17.10 Podstawienie obrazu na tło renderingu

Często tło renderingu, wypełnione prostymi barwami (por. 967), lub panoramą (por. str. 596, 987) pozostawia wiele do życzenia, ze względu na brak realizmu. Rysunek 17.10.1 pokazuje taki właśnie przypadek — chmury tła pochodzą z tekstury (panoramy) otoczenia. Są nierealistycznie "rozmyte", z powodu niedostatecznej rozdzielczości panoramy. Czy można w tło renderingu "podstawić" realistyczne zdjęcie (Rysunek 17.10.2)?





Rysunek 17.10.1 Model z kiepskim tłem

Rysunek 17.10.2 Obraz — lepsze tło dla modelu

Jest to całkiem proste. W zestawie *Scene* (F10): *Render buttons*, w panelu *Output*, wskaż Blenderowi plik z obrazem, który ma być użyty jako tło (Rysunek 17.10.3):

	zestaw Scene (F10)	podzestaw Render buttons
 Wskaż plik z obrazem, który ma być podstawiony 		
jako tło	Output Render Layers //mp\ //mp\ ///background/nebraska.png Extensions Touch No Overwrite	2. Uaktywnij podstawianie obrazu jako tła Shado SS Pano EnvMa Ray Radi
	No Set Scene Other: 0.000 Edge Edge Settings Threads: 1 Disable Tex Free Tex Images Save Buffers Render Window	OSA MBLUR 100% 5 8 11 16 Bf: 0.50 75% 50% 25% • Xparts: 4 • • Yparts: 4 • Fields Odd X Sky Premul Key 128 Border

Rysunek 17.10.3 Ustalenie obrazu tła w parametrach renderingu

Rysunek 17.10.4 przedstawia rezultat renderingu po ustawieniu obrazu jako tła:



Rysunek 17.10.4 Rezultat: model z lepszym tłem

17.11 Wygładzanie linii obrazu

Podczas renderowania rezultatu, Blender musi przeliczyć uzyskane barwne obszary na konkretne piksele ostatecznego obrazu. W najprostszym przypadku wszystkie linie ukośne na obrazie będą narysowane jako "schodki" poszczególnych pikseli (Rysunek 17.11.1a). To nie wygląda dobrze, nawet przy większej rozdzielczości. Algorytmy tzw. nadpróbkowania (*oversampling*, określany także jako *antyaliasing*) pozwalają na ich wizualne wygładzenie. W Blenderze *oversampling* występuje pod skrótem **OSA**. Po włączeniu **OSA** całość wygląda o wiele lepiej (Rysunek 17.11.1b):



Rysunek 17.11.1 Wpływ wygładzania OSA na renderowany obraz

Panel sterowania efektem znajduje się w zestawie przycisków *Scene* (F10), podzestawie *Render buttons*. Wygładzenie włącza się przełącznikiem *OSA* (Rysunek 17.11.2):



Rysunek 17.11.2 Wybór filtra wygładzania w panelu Render

Poniżej przełącznika OSA (p. Rysunek 17.11.2) możesz zmienić rozmiar obszaru wygładzania (przyda się, gdy włączysz rozmycie ruchu — *MBLUR*). Oprócz tego panel zawiera tajemniczą listę bez etykiety (p. Rysunek 17.11.2). To *Filter* — pozwalający wybrać jedną z kilku możliwych metod (tzw. "filtrów") wygładzania. Domyślnie ustawiony jest na wygładzanie metodą Gaussa (jedną z najbardziej popularnych). Możesz jednak sprawdzić, jak Twój model będzie wyglądał z użyciem innej metody.

Osobiście uważam, że dla modeli samolotów domyślny filtr (*Gauss*) jest "zbyt gładki". Nie przepadam za drobnym rozmyciem konturów, które są jego "efektem ubocznym"(Rysunek 17.11.3):



Rysunek 17.11.3 Efekt wygładzania metodą Gauss (powiększenie x4)

Wolę przełączyć się na filtr CatRom, zachowujący ostrość krawędzi (Rysunek 17.11.4):



Rysunek 17.11.4 Efekt wygładzania metodą CatRom (powiększenie x4)

Myślę jednak, że dobór filtra wygładzania jest decyzją bardzo osobistą¹ — niemal tak, jak wybór tapety na pulpit Twojego komputera ☺.

975

¹ Leonardo da Vinci wyróżniał się, w oczach współczesnych artystów Renesansu, zastosowaniem techniki malowania "sufmato". (Był jej twórcą.) Efekt "sufmato", przynajmniej na reprodukcjach, wyglądają jak gaussowskie wygładzenie, tyle że malowane ręcznie!

17.12 Skonfigurowanie materiału: szkło

Stwórz nowy materiał i nadaj mu białą barwę (por. str. 957). Rysunek 17.12.1 pokazuje, jak będą wyglądać ustawienia tego materiału w zestawie *Shading:Material buttons*:

	🕈 🔻 Panels 🕝 🗟 🍑 🗶 🔲 🕅 🔆	۱ کې 🛃 🌑	∢∵1 ⇒₽		
PI	▼ Links and Pipeline	Material	Ramps	Shaders St	SS
P	Nadaj nowemu materiałowi biała barwe		**	Lambert ¢ Ref 0.800	Tangent V
	E:B.300.Fuselage OB ME 1 Mat 1	VCol Light VCol	Paint TexFace A Shadeless		Shadow
				CookTorr = Spec 0.500	Tra.Shado Only.Shad
	Render Pipeline	Col	R 1.000		Cubic
	Halo ZTransp Zoffs: 0.00	Spe	G 1.000	GR: Ex	clusive Bias
	Full Osa Wire Strands Zinvert	Mir	B 1.000	Tralu 0.00 SBia	s 0.00
	Radio OnlyCast Traceable Shadbuf	RGB HSV DVN	A 1.000	Amb 0.500 Emit	0.000
				LBias 0.00	

Rysunek 17.12.1 Domyślne ustawienia nowego materiału

Zaczniemy od zmian w panelu **Shaders**. Zmień sposób liczenia rozproszenia światła przez materiał (*diffuse*) z **Lambert** na **Oren-Nayar** (Rysunek 17.12.2). Dodatkowo w jego parametrach ustaw:

- Ref = 1.0 (gdy dać mu szansę – np. w lustrze – szkło odbije prawie całe światło);

Zmień także sposób wyznaczania odbłysków (*specular*) światła przez materiał z *Cook-Tor* na *Blinn* (Rysunek 17.12.2). Dodatkowo ustaw:

- **Spec** = 2.0 (materiał jest bardzo "połyskliwy");
- Hard = 100 (ze zdjęć wynika, że szkło organiczne jest "miększe" od zwykłego szkła tzn. powstające na nim odbłyski są większe, niż na szkle);



Rysunek 17.12.2 Zmiana typów i parametrów shaderów

W panelu MirrorTransp włącz Ray Transp (Rysunek 17.12.3). Ustaw także:

- w panelu *MirrorTransp*, w sekcji poniżej *Ray Transp*, wartość *Depth* = 5;
- w panelu *Material* zmień wartość A (lpha) z 1.0 (całkowicie nieprzejrzyste) na 0.2;



Rysunek 17.12.3 Włączenie przejrzystości materiału

Zaczyna coś już prześwitywać, prawda? Tym niemniej wygląda to nadal jak przyciemniona szyba samochodowa. Póki co poprawimy kilka parametrów, sterujących przenikaniem światła przez materiał szyby. (Szkło kabiny ma co prawda tylko kilka milimetrów grubości. Przy ustawieniu kamery pod bardzo ostrym kątem nawet taka grubość staje się istotna. Poza tym, w tym myśliwcu wystąpi także gruba — 52 mm — szyba pancerna. Co prawda jeszcze jej nie zamodelowaliśmy, ale zrobimy to wraz z innymi szczegółami kabiny. Jej także zostanie przypisany podobny materiał).

Ustaw w panelu *Mirror Transp* (Rysunek 17.12.4):

- IOR = 1.4 (wspc. załamania światła w plastiku);
- Fresnel = 2.0 (wizualne "zagęszczenie" materiału oglądanego pod bardzo dużym kątem);
- Fac = 1.0 (granica wizualnego "zagęszczenia" na wygiętej powierzchni będzie dość ostra);
- Spec Tra = 0.0 (minimalizacja "drugiego odbicia" światła od wewnętrznych ścian szyby);



Rysunek 17.12.4 Parametry związane z grubością i zakrzywieniem szyby

Na razie nie widać specjalnych zmian (por. Rysunek 17.12.4 i Rysunek 17.12.3). Kolejnym krokiem w kierunku realizmu jest dodanie do materiału możliwości odbicia obrazu otoczenia. Podobnie jak woda w kałuży, przejrzyste szkło zmienia się w lustro, gdy jest oglądane pod odpowiednim kątem. Aby to odwzorować, włącz w panelu *Mirror Transp* funkcję odbicia: *Ray Mirror*. Dodatkowo ustaw (Rysunek 17.12.5):

- Ray Mir. = 0.3 (to nie będzie zbyt intensywne "lustro");
- Fresnel = 1.5 (szyby zmienią się z przejrzystych w lustrzane pod dość dużym kątem widzenia);
- Fac = 1.7 (ta wartość zapewni płynne przejście pomiędzy obszarem szkła przejrzystego i "lustra");
- **Depth** = 4 (na wszelki wypadek analogicznie jak w *Ray Transp*);



Rysunek 17.12.5 Parametry związane z lustrzanym odbiciem otoczenia

Nasza osłona stała się nieco bardziej przejrzysta, ale nadal jest za ciemna. Gdy jest ustawiona pod tak ostrym kątem, jak na ilustracji, na bocznych ścianach pojawiają się odbicie otoczenia. Na przykład — obraz statecznika poziomego na tylnej szybie kabiny (por. Rysunek 17.12.5). Gdybyś chciał "rozmyć" te odbicia — zmień wartość *RayMirror:Gloss* z 1.0 na 0.95 lub 0.9.

977

Zajmijmy się jeszcze cieniem, jaki rzuca owiewka na powierzchnię samolotu (Rysunek 17.12.6). Do tej pory cień rzucany przez zespół osłony wygląda tak, jak gdyby całość była wykonana z litego materiału (Rysunek 17.12.6a). Jak to zmienić?



Rysunek 17.12.6 Skorygowanie cieni, rzucanych przez oszklenie kabiny.

Cała sztuka polega na włączeniu jednej opcji w panelu Shaders: TraShadow (Transparent Shadow). Jest tylko jeden "haczyk": TraShadow włączamy dla materiału, na który pada cień! Praktycznie warto go włączyć dla wszystkich materiałów, których używasz w modelu, w szczególności dla powierzchni zewnętrznych (B.Skin.Camuflage) i wewnętrznych (B.Skin.Inner). Dodatkowo, należy go także włączyć dla samego materiału szkła. W takiej "zawiniętej" powłoce jako owiewka, jedna strona oszklenia rzuca cień na drugą. Powoduje to niepotrzebne "przyciemnienie" szyb. Porównaj Rysunek 17.12.6a (TraShadow dla szkła wyłączone) i Rysunek 17.12.6b (TraShadow dla szkła włączone). Czy zauważyłeś, że owiewka pojaśniała?

Do tej pory pracowaliśmy na podglądzie owiewki, z kamerą położoną blisko osi samolotu. Ustawmy się teraz trochę z boku, i oceńmy osiągnięty do tej pory efekt (Rysunek 17.12.7):



Rysunek 17.12.7 Efekt, osiągnięty do tej pory (AO włączone)

Odbicia na oszkleniu wyglądają całkiem realistycznie. Krawędź płata, widziana przez zakrzywiony grzbiet owiewki, jest lekko zdeformowana - tak jak byłaby w rzeczywistości. (Spokojnie, z wnętrza kabiny widok nie jest zdeformowany). Szyby wydają się lekko "przyciemnione", ale może tak musi być... Ale dlaczego ściana kadłuba pod tylną szybą jest tak ciemna?!

Muszę przyznać, że znalezienie odpowiedzi zajęło mi kilka godzin. Przyczyna okazał się jeden z efektów związanych z światłem otoczenia: Ambient Occlusion (por. str. 968). Ta trochę długa nazwa będzie dalej w tej sekcji określana skrótem: AO.

Wystarczy, że wyłączysz AO, a barwa ścian kadłuba staje się jednolita. Co więcej, szyby wreszcie przestają sprawiać wrażenie "przyciemnionych" (Rysunek 17.12.8):



Rysunek 17.12.8 Ten sam materiał, ale bez AO.

Dla przypomnienia: AO został włączony, by "zagęścić cienie" wokół krawędzi i wewnątrz otworów. Był szczególnie przydatny w przypadku słabiej oświetlonych powierzchni, takich jak spód samolotu (por. str. 968 i dalsze). Włączyliśmy go w trakcie eksperymentów z ujęciem modelu "od dołu". Wydawało się, że to samo ustawienie można pozostawić także lepiej oświetlonych projekcjach (por. str. 276). I tak było, dopóki w modelu nie pojawiły się przejrzyste powierzchnie.

Wygląda na to, że renderer Blendera, którego używamy (tzw. *internal renderer*) przy obliczaniu efektu AO nie zwraca uwagi na przejrzystość materiału. W związku z tym "przyciemnił" wewnętrzne powierzchnie osłony kabiny (gdyż były, według niego, nie oświetlone). Przyciemniony został — i to znacznie — także fragment kadłuba "zasłonięty" przez szybę ciągnącą się za fotelem pilota¹.

AO można było traktować jak specyficzny rodzaj światła: rozjaśniało większość powierzchni sceny. Po jego wyłączeniu "rozdziel" tę niezbędną energię na zwiększenie intensywności pozostałych świateł (**Sun** i **Hemi**) (Rysunek 17.12.9):



¹ Najbardziej oczywistym rozwiązaniem tego problemu byłoby wykluczenie szklanych elementów owiewki z obliczeń *AO*. Niestety, nie ma (jeszcze?) we właściwościach materiału lub obiektu takiego przełącznika. Taki efekt można uzyskać inną, bardziej złożoną drogą: wykorzystując tzw. *Composite Nodes*.

Efekt, jaki przedstawia Rysunek 17.12.9, można już zaakceptować jako wystarczająco realistyczny. W niektórych ujęciach szyby nadal mogą się jednak wydawać odrobinę za ciemne (Rysunek 17.12.10a):





Wciąż jeszcze szkło sprawia w widoku, jaki pokazuje Rysunek 17.12.10a, wrażenie przyciemnionego. Jak je jeszcze rozjaśnić, skoro już i tak nadaliśmy temu materiałowi biały kolor? Rozwiązanie jest trochę dziwne, gdybyśmy szukali odpowiednika w realnym świecie: nasz materiał ma zacząć "emitować" pewną ilość światła! Ustawia się to w panelu *Shaders*. Służy do tego parametr *Emit* (Rysunek 17.12.10b).

Wypracowany do tej pory materiał nie jest całkiem przejrzysty. Wygląda jak szkło, które już się trochę "zestarzało", bo nie było dobrej jakości. Taki efekt można zobaczyć na kabinach samolotów ZSSR, wyprodukowanych podczas wojny. Pod tym względem materiały brytyjskie, niemieckie i amerykańskie były o wiele lepsze, i "trzymały" przejrzystość mimo upływu czasu. Jak uzyskać taką "lepszą" szybę? (Rysunek 17.12.11):



Rysunek 17.12.11 Alternatywna (bardziej przejrzysta) postać szkła organicznego

Po pierwsze, trzeba użyć innego zestawu shaderów: *Fresnel* (rozproszenia) i *Wardlso* (odbłyski). Należy także wyłączyć "przyjmowanie" cieni (*Shaders:Shadow*) (Rysunek 17.12.12):



Rysunek 17.12.12 Ustawienia alternatywnej postaci szkła

Wyłączenie opcji *Shadow* w panelu *Shaders* spowodowało rozjaśnienie wewnętrznych powierzchni szyb. Uznałem, że bez tego nasze szkło wygląda korzystniej.

Po zmianie shadera światła rozproszonego na *Fresnel*, ustawiłem dwa specyficzne parametry: *Fresnel* — na 2.0 i *Fac* — także na 2.0 (wspc. *Ref* pozostał bez zmian — 1.0). Po zmianie shadera odbić na *Wardlso* zmniejszyłem nieco ich intensywność (parametr *Spec*) — z 2.0 do 1.7. W sumie nie powoduje to jakiejś specjalnej różnicy, ale *Wardlso* sam w sobie jest bardziej "połyskliwy" od innych shaderów, i wolałem to uwzględnić. "Twardością" (czyli rozmiarem powierzchni) odbłysków można teraz płynnie sterować zmieniając parametr *rms*. Ustawiłem go na umiarkowaną wartość (*rms* = 0.2).

Z innych zmian — w parametrach sterujących przejrzystością (panel *Mirror Transp*, grupa *Ray Transp*) zwiększyłem wartość *Fac* do 2.0 (Rysunek 17.12.12). W razie czego możesz tu także zmniejszyć parametr *IOR*, jeżeli uznasz aktualne załamanie światła przechodzącego przez szyby za nierealistyczne. Wartość *IOR* = 1.40 jest "podręcznikową" wartością dla plastików (szkło ma zresztą jeszcze wyższą). Mimo to np. przesunięcie obrazu płata widzianego przez szybę (por. Rysunek 17.12.11b) wydaje mi się przesadzone. W każdym razie wiesz, gdzie to skorygować.

Istotnym zmianom uległy także wartości sterujące lustrzanym odbiciem (panel *Mirror Transp*, grupa *Ray Mirror*). Zdecydowanie wzmocniłem ten efekt, zwiększając wartość *Ray Mir* do 1.0. W razie czego możesz go zmniejszyć o połowę, ale to trochę sprawa gustu. W każdym razie Rysunek 17.12.13 pokazuje różnicę:



Rysunek 17.12.13 Wpływ poziomu lustrzanego odbicia

Rysunek 17.12.13a) pokazuje efekt dla *Ray Mir* = 0.5, a Rysunek 17.12.13b) — dla *Ray Mir* = 1.0. Osobiście uważam, że włączenie pełnego odbicia (Rysunek 17.12.13b) wygląda lepiej. Inne parametry z grupy *Ray Mirror* sprawiają, że nasze szkło nie zamienia się w "lusterko": *Fresnel* = 1.9, *Fac* = 1.45 (Rysunek 17.12.12).

Wreszcie warto też zwrócić uwagę na jasność tła, widzianego i odbijanego przez szyby. To nie jest zdjęcie "podstawione" za modelem! (Por. Rysunek 17.12.11b, Rysunek 17.12.13). Obraz "dla szyb" (a także wszelkich innych odbić — np. wypolerowanego duralu) ustawiasz w zestawie *Shading:World buttons* (Rysunek 17.12.14):

😑 🔻 Panels 🥥 🗐 🕥 1	ζ 🔲	🔆 🥚 📓 👶 🛞	∃ 1 ⇒		
V Preview		Vorld		Texture and In	put Map To
1.1870-147.J.C.	Blend	⇒ WO:World	XF	Blend Ho	ri ZenUp ZenDo
	Paper				
	Real	HoR 1.000	ZeR 0.701	Stencil Neg No RC	aB Mix 🗢
		HoG 1.000	ZeG 0.850		Col 0.200
	_	HUB 1.000		R 1.000	II Nor 0.500
		Rozjaśnij kolor		G 0.000	— Var 1.000 — I
		horyzontu oraz	Gdy stosujes	sz mapę HDRI:	
State 1	Ŀ	AMBB U.14	Range 2. zmniejszając v	vspółczynnik Col	

Rysunek 17.12.14 Rozjaśnienie otoczenia sceny (wpływa na barwę odbicia, widocznego na szybach)

Copyright Witold Jaworski, 2009 - 2011.

Szczegółowe informacje o ustawieniach tekstury otoczenia (obrazu HDRI) znajdziesz na str. 596. Opis zestawu *Shading:World buttons* znajduje się na str. 986.

Kluczowe właściwości naszego materiału są związane z efektami *Ray Mirror* i *Ray Transp*, włączanymi w panelu *Mirror Transp* (por. chociażby Rysunek 17.12.12). Nie byłoby z tym żadnego problemu, gdybyś do odwzorowania otoczenia stosował jakieś bardzo szczegółowe tekstury HDRI. Na razie jednak trudno taki obraz uzyskać i potem renderować (realistyczny rednering wymaga pliku o ogromnym rozmiarze i bardzo wydajnego komputera). Sądzę, że tło dla modelu będziesz uzyskać prostszą metodą: poprzez "podstawienie" zdjęcia za finalny rendering (tak, jak jest to opisane na str. 998). W takiej sytuacji pojawia się problem: otoczenie, widziane przez szyby, nie jest tym, które widać na podstawionym zdjęciu (por. Rysunek 17.12.11b, Rysunek 17.12.13). Wynika z tego kilka praktycznych wniosków:

- Komponuj sceny tak, by szyba znajdowała się w jakimś fragmencie obrazu o możliwie jednolitej barwie (chmury, niebo). Do takiego tła można w miarę prosto dopasować jasność i barwę obrazu otoczenia (w zestawie *Materials:World*)
- W scenach, gdzie przez szybę ma być widoczny obraz otoczenia (np. widok z wnętrza kabiny) używaj alternatywnego (prostszego) modelu przejrzystości. Polega to na włączeniu opcji *ZTransp* i sterowaniu przejrzystością za pomocą parametru *A* (*lpha*).

Rysunek 17.12.15 przedstawia ustawienia takiego "uproszczonego szkła", poprzez które widać zdjęcie, podstawione jako tło:

😑 🗢 Panels @ 🗟 🖉 🗶 🔲 🕅	€ 💽 🐼 🚱 🔣 🗶 🔹	
▼ Preview	Links and Pipeline SSS	Material Ramps
	Link to Object ■ MA:Glass 3 ★ ♣ F Node 25.CanopyGlass OB ME Włącz opcję ZTransp Render Pipeline Halo ZTransp Zoffs: 0.00 Full Osa Wire Strands Zinvert Radio OnlyCast Traceable Shadbuf	VCol Light VCol Paint TexFace A Shadeles No Mist Env ObColor Shad A 1.00 Nieprzejrzystość określa parametr Alpha B 1.000 RGB HSV DYN A 0.100

Rysunek 17.12.15 Ustawienia uproszczonego "szkła", uzyskanego za pomocą parametrów ZTransp i Alpha.

Rysunek 17.12.16 przedstawia efekt, jaki daje ten uproszczony materiał (por. z Rysunek 17.12.11b):



Rysunek 17.12.16 Alternatywny, uproszczony "efekt szkła" (materiał z opcją ZTransp)

17.13 Skonfigurowanie materiału: plastik

Gładkie powłoki wykonane z tworzyw sztucznych wydają się najprostsze do odwzorowania za pomocą materiałów "komputerowych". Choć nie zawsze — w poprzedniej sekcji musieliśmy się jednak sporo napocić, by uzyskać coś, co przypomina szkło organiczne. To jest jednak materiał na bardzo wyeksponowanym elemencie modelu — kabinie pilota. Inne detale, takie jak np. osłony świateł pozycyjnych, nie są aż tak kłopotliwe.

W tej sekcji pokażę, jak uzyskać materiał na osłonę lewego światła pozycyjnego (jest to światło czerwone). Na prawym skrzydle jest umieszczone światło zielone, pokryte takim samym materiałem, tylko o zmienionej barwie.

Stwórz nowy materiał i nadaj mu barwę czerwoną (por. str. 957). Rysunek 17.13.1 pokazuje, jak należy zmienić ustawienia w zestawie *Shading:Material buttons*:



Rysunek 17.13.1 Domyślne ustawienia nowego materiału

W panelu *Links and Pipeline* włącz opcję *ZTransp*. W panelu *Material* ustaw odpowiednią barwę podstawową (*Col*), oraz odbłysków (*Spe*). Zmniejszyłem także nieprzezroczystość *A* (*lpha*). W panelu *Shaders* zmień sposób liczenia rozproszenia światła przez materiał (*diffuse*) z *Lambert* na *Oren-Nayar* (Rysunek 17.12.2). Dodatkowo w jego parametrach ustaw:

- **Ref** = 0.4 (to stosunkowo grube i niezbyt przejrzyste szkło, "wchłonie" sporo światła);
- Rough = 1.0, by materiał wydawał się nieco ciemniejszy;

Zmień także shader odbłysków z Cook-Tor na Blinn, i w jego parametrach ustaw:

- Spec = 2.0 (materiał jest bardzo "połyskliwy");
- Hard = od 150 do 300 (ja się zdecydowałem na 150: większa powierzchnia odbłysków lepiej wygląda na tak małym detalu, jak światło sygnalizacyjne);
- Refr = 10 (wartość maksymalna);

Nadałem temu materiałowi nazwę B.Red.Plastic. Rysunek 17.13.2 przedstawia rezultat jego zastosowania:



Rysunek 17.13.2 Materiał — półprzeźroczysty, czerwony plastik (próbny rendering światła sygnalizacyjnego)

983

Żadna powierzchnia w realnym świecie nie jest tak gładka (przyjrzyj się, jak wygląda ramka tego światła – Rysunek 17.13.2). Przygotujmy więc teksturę, która odwzoruje drobne nierówności (Rysunek 17.13.3):



Rysunek 17.13.3 Proceduralna tekstura "szumu" — B.Noise.General.

To tekstura proceduralna typu *Stucci*. Nadałem jej nazwę **B.Noise.General**. Po określeniu typu tekstury zmień funkcję "szumu" (*Noise Basis*) na *Vornoi Crackle*, oraz zmniejsz jego rozmiar (*Noise Size*) do 0.02.

Rysunek 17.16.5 przedstawia nałożenie tekstury **B.Noise.General** na siatkę lampy:



Rysunek 17.13.4 Mapowanie obrazu "szumu" (B.Noise.General) na siatkę lampy pozycyjnej

Żaden z parametrów, podstawionych przez Blender domyślnie w panelu *Map Input*, nie musiał być w tym przypadku zmieniany. Domyślny, purpurowy kolor tekstury "zalał" całą powierzchnię, pozostawiając tylko "kropki" w oryginalnym kolorze. Chodzi nam o efekt odwrotny (cała powierzchnia ma być w oryginalnej barwie, a "kropki" — w innej). Dlatego użyjemy negatywu tej tekstury. Szczegóły jej mapowania na parametry materiału przedstawia Rysunek 17.13.5a):



Rysunek 17.13.5 Modulacja parametrów materiału przez teksturę "szumu" (B.Noise.General)

Negatyw tekstury włączysz przełącznikiem *Map To:Neg.* Na kolor tych skaz/zabrudzeń wybrałem ciemnoszary (Rysunek 17.13.5a), ale niezbyt intensywny (*Col* = 0.300). Dodatkowo ten sam wzór chaotycznych "kropek" mapuję jako bardzo drobne nierówności (*Nor* = 0.01). Rysunek 17.13.5b) przedstawia ostateczny rezultat

17.14 Skonfigurowanie materiału: dural

Większość samolotów — od drugiej połowy lat trzydziestych do czasów współczesnych — była wykonywana z blachy duraluminiowej. Czasami - w przypadku prototypów, samolotów cywilnych, oraz części maszyn wojskowych — pozostawiano je w naturalnym kolorze poszycia. Prędzej czy później zabierzesz się za wykonanie modelu w takich właśnie barwach. Pojawi się wówczas pytanie: jak stworzyć w Blenderze materiał, który imitowałby blachę (duralową)?

Zacznijmy od podstaw: przejdź do zestawu *Shading* (F5), podzestawu *Material buttons*. Aluminium to jasny metal, więc zmień kolor poszycia na jasnoszary (85% szarości - Rysunek 17.14.1):



Rysunek 17.14.1 Wstępne ustawienia materiału - barwa podstawowa, shadery.

Oprócz tego ustaw w panelu Shaders typowe metody obliczania odbicia światła: Oren-Nayar dla rozproszonego (diffuse shader), Blinn dla odbłysków (specular shader). Zwiększ współczynnik odbicia światła rozproszonego Ref do 1.0 ("lustro" odbija prawie całe światło, które na nie pada). Zwiększ także połyskliwość Spec do 1.2. Pozostaw "twardość" (Hard) odbłysków na stosunkowo niskim poziomie - 50. (To specyfika bezbarwnego lakieru, jakim pokryte jest poszycie).

Dlaczego właściwie tak przygotowany materiał wygląda - no, ... "niemetalowo"? Ponieważ prawdziwy metal po części zachowuje się jak lustro¹. W panelu *Mirror Transp* włącz więc opcję *Ray Mirror* (Rysunek 17.14.2):

😑 🕈 🔻 Panels 🚱 🔂 🗶 🛄 📓	
Preview	V Shaders Włącz Ray Mirror Mirror Transp Dren-Na = Bet 1 000 Ustaw Ray Mir = 0.5 Ray Mirror Ray Mirror Ray Mirror Ustaw Ray Mir = 0.5 Nadow Ray Mirror Ray Mirror Ray Transp Blinn Ustaw Ray Mir = 0.5 Nadow Fresnel: 1. Fresnel: 0. Blinn Ustaw Fresnel = 1.0 - 1.3 Shadow Fresnel: 1. Fresnel: 0. Ustaw Fac = 1.25 - 1.4 Cubic Gloss: 1.00 Gloss: 1.00 Gloss: 1.00 I Tralu 0.00 SBias 0.00 I Emit 0.000 I Amb 0.500 I Emit 0.000 I Bias 0.00 I Emit 0.000 I Emit 0.000 I Imit 0.000 Imit



Poniżej przełącznika Ray Mirror znajduje się zestaw jego parametrów. Ustaw intensywność odbicia (Ray Mir) na 0.5 (50%). (Przy 100% widziałbyś wyłącznie to, co się odbiło, a nie barwę samego metalu). Zwiększ także "wytłumienie" odbicia w centrum obrazu — efekt *Fresnel* — do wartości 1.0.

985

¹ A właściwie: to lustro "udaje" metal. Szkło jest w nim potrzebne wyłącznie do ochrony powierzchni i zapewnienia niezbędnej "gładkości" cienkiej warstwie metalu, naniesionej od wewnątrz.



Wykonajmy teraz rendering, aby zobaczyć, jak materiał prezentuje się na modelu (Rysunek 17.14.3):

Rysunek 17.14.3 Odbicie lustrzane — pierwsze przybliżenie

Efekt zbliżony do oczekiwanego, ale pozostawia jeszcze dużo do życzenia. Po pierwsze, jakoś za mało urozmaicona jest powierzchnia samolotu. Na płatach widać tylko odbicie kadłuba, a na kadłubie — płata. A gdzie inne szczegóły?

To, czego nam brakuje, to otoczenie. W realnym świecie zawsze coś się znajduje dookoła — chociażby sam obserwator. Rysunek 17.14.3 pokazuje odbicia, jakie pojawiłyby się na samolocie w bezkresnej, błękitnej otchłani. Musimy ustawić dookoła samolotu jakiś obraz otoczenia — nieba, chmur, a u dołu — ziemi lub chmur... Załaduj (jak — p. str. 962) teksturę otoczenia, składającą się głównie z błękitu i chmur (Rysunek 17.14.4):





Rysunek 17.14.4 Ten sam obraz otoczenia w dwóch alternatywnych rozwinięciach — cylindrycznym i kątowym (angular map).

Plik z obrazem dla tej tekstury — w wersji z mapowaniem kątowym, jaką pokazuje Rysunek 17.14.4b — znajdziesz wśród plików dostarczonych wraz z tą książką: *p40\background\sky5-clouds.png*¹.

¹ Jest to obraz pobrany z kolekcji "nieb", udostępnianej za darmo na <u>http://lfgrafix.com/</u>. Są tam udostępniane w mapowaniu cylindrycznym, jakie pokazuje Rysunek 17.14.4a. Mapowanie kątowe uzyskałem za pomocą programu **Qtpfsgui** (udostępnionego na zasadach GPL). Przy okazji obróciłem niebo o 180°, by słońce było z tyłu obserwatora. **Qtpfsgui** można pobrać z <u>http://qtpfsgui.sourceforge.net/</u>.

987

Załadowaną teksturę otoczenia zamapuj jako *AngMap*, oraz przypisz do obszarów ekranu: *Hori*, *ZenUp*, *ZenDo* (Rysunek 17.14.5):

Panels @ Dt. [◙ ▓●₫•₽	Włącz przypisania do ł ZenUp, ZenDo	lori,
Vorid	🕞 🔻 Texture and	d Input	🔻 Мар То 🎽
P WO:World F	A B.HDRI Mapowanie obrazu:	TE:B.HDRI	Blend Hori ZenUp ZenDo
HoR 0.732 ZeR 0.022	AngMap	View Global	Stencil Neg No RGB Mix 🗢
HoG 0.802		AngMap Sphere Tube	Col 1.000
HoB 0.986		Ubject UB:	R 1.000 Nor 0.500
AmbR 0.14		dX 0.00 → ≤ sizeX 1.000 → dX 0.00 → ≤ sizeX 1.000 →	G 0.000
AmbG 0.14		d7 0.00 + size 7 1.000 +	
AmbB 0.14		a az 0.00 F a sizez 1.000 F	DVar 1.00

Rysunek 17.14.5 Mapowanie tekstury otoczenia

Rezultat wygląda już lepiej (Rysunek 17.14.6):



Rysunek 17.14.6 Aktywne lustrzane odbicia + obraz otoczenia

Kadłub i lewe skrzydło wyglądają już całkiem ciekawie (Rysunek 17.14.6). Prawy płat zachował jednolitą barwę, gdyż akurat odbija obraz bezchmurnego fragmentu nieba. (Jedynie na krawędzi natarcia widać zdeformowane odbicie horyzontu).

Pokrycie prawdziwego samolotu nie jest jednak doskonałym lustrem. Nawet maszyny pracowicie "wypucowane" przez obsługę naziemną nie dają tak idealnego odbicia, jakie pokazuje Rysunek 17.14.6. Aby uzyskać taki bardziej realistyczny efekt, zmniejsz w panelu *Mirror Transp* wartość *Gloss* — do poziomu 0.85 (Rysunek 17.14.7). To wydłuży czas renderowania. Możesz jednak tym czasem sterować za pomocą wartości *Samples*. Dla próbnego renderingu używaj *Samples* = 3. Dla ostatecznego — ustaw *Samples* na poziomie 15 - 18.



Rysunek 17.14.7 "Rozmycie" lustrzanego odbicia

Rysunek 17.14.8 przedstawia efekt ustawienia *Gloss* = 0.85:



Rysunek 17.14.8 Aktywne lustrzane odbicia + obraz otoczenia + rozmycie odbicia (Samples = 18)

Samolot wygląda na pokryty realistyczną, dość matową blachą. I o taki efekt chodziło.

 Pamiętaj, że możesz płynnie sterować poziomem "wypolerowania" poszycia, zmieniając w panelu *Mirror Transp* wartość *Gloss*.

17.15 Skonfigurowanie materiału: stal

Materiału, który wygląda jak stal, będziemy potrzebować do wykonania różnych drobnych detali samolotu. Ten materiał jest matowy, więc nie będziemy w nim włączać żadnych opcji *Ray Mirror*, jak to robiliśmy dla duralu (por str. 985). Przejdź do panelu *Material buttons* w zestawie *Shading*.

Rysunek 17.15.1 przedstawia podstawowe ustawienia kolorów i shaderów dla stali:





Podstawnym kolorem (*Material:Col*) materiału jest szary (60%). Przyciemniłem także nieco światła odbłysków (*Material:Spe*) — do 80%. Jeżeli chodzi o shadery, to dla rozpraszania światła zastosowałem typowy *Oren-Nayar*, o dużej chropowatości (*Rough* = 2.0) i w pełnym rozproszeniu światła (*Ref* = 1.0). Jako shader odblasków uczyłem tym razem nie występującego w parze z *Oren-Nayar* shadera *Blinn*, tylko bardziej odpowiedniego dla metali *WardIso*. "Podkręciłem" w nim obszar rozpraszania światła (*rms*).

Rysunek 17.14.2 przedstawia, jak takie ustawienia wyglądają na przedmiocie pokrytym tym materiałem (na przykładzie goleni kółka ogonowego — obiektu **B.410.Leg**).

Aby całość wyglądała bardziej "metalowo", nadamy materiałowi kolor zależny od kąta "patrzenia". Użyłem do tego funkcji z panelu *Ramps* (spektrum barw — więcej na ten temat znajdziesz na str. 960)



Rysunek 17.15.2 Test materiału na renderingu

Za pomocą przycisku *Add* dodałem do paska barw kolejną "pozycję", pośrodku. Następnie zmieniłem każdą z nich, ustawiając ich barwy tak, jak pokazuje to Rysunek 17.15.3:



Rysunek 17.15.3 Ustawienie spektrum kolorów materiału

We wszystkich pozycjach paska ustawiłem kompletną nieprzejrzystość (*A* = 1.0). Tryb "aplikacji" barwy pochodzącej "z paska" z podstawową barwą materiału (*Method*) jest ustawiony na *Mix*, a współczynnik udziału *Factor* = 1.0. Oznacza to zupełne wyeliminowanie barwy podstawowej, (ustawionej na str. 989). Wystarczy jednak tylko zmniejszyć wartość *Factor*, by zaczęła "prześwitywać".

Rysunek 17.15.4 przedstawia rednering naszej testowej goleni z nowymi ustawieniami. Być może w tym ujęciu nie widać tego najlepiej, ale w rezultacie zastosowania paska barw pojawił się na przedmiocie pożądany "metaliczny połysk". Na razie uzyskana powierzchnia wygląda jak gładki metal (Rysunek 17.15.4). Gdyby jej zmniejszyć nieco parametr *Ref* (z panelu *Shaders*), to całkiem dobrze udaje zwykłą, gładką powierzchnię stali.



Rysunek 17.15.4 Rezultat na renderingu

Potrzebujemy także czegoś innego: powierzchni pomalowanej "srebrną" farbą. Przez takie pokrycia sprawia wrażenie lekko chropowatej (a przynajmniej tak odbija światło). Jak wzbogacić materiał o taki efekt, jednocześnie nie tracić czasu na rysowanie obrazu abstrakcyjnego "szumu"? Wystarczy wykorzystać odpowiednią teksturę proceduralną. Jej przykład przedstawia Rysunek 17.15.5:



Rysunek 17.15.5 Tekstura proceduralna Clouds, w układzie przypominającym powierzchnię metalu

Teksturze, którą przedstawia Rysunek 17.15.5, nadałem nazwę **B.Rough.Metal**. Powstała na bazie typu *Clouds*. Wystarczyło zmienić funkcję bazowego szumu (*Noise Basis*) z *Blender Original* na *Vornoi Crackle*, oraz zmniejszyć jego rozmiar (*Noise Size*) do 0.10. (Szczegółowy opis, jak stworzyć taką teksturę, znajdziesz na str. 964).

Rysunek 17.15.6 przedstawia szczegóły mapowania **B.Rough.Metal** w materiale:





Zwróć uwagę na panel *Map Input*. Zastosowałem tu na dość rzadko używaną metodę mapowania tekstury *Refl*. Po kilku próbach doszedłem do wniosku, że w tym trybie nierówności układają się najlepiej. W panelu *Map To* nie ma żadnych rewelacji: negatyw obrazu moduluje nierówności (*Nor* jest żółte, czyli ujemne), z umiarkowaną intensywnością (=0.3).

991

Powierzchnia stała się chropowata!

Rezultat zastosowania tekstury **B.Rough.Metal** przedstawia Rysunek 17.15.7. Powierzchnia goleni przestała już wyglądać na gładką! Takie drobne nierówności są odpowiednie dla odkuwek.

Rysunek 17.15.7 Efekt nałożenia tekstury nierówności (B.Rough.Metal)

Na takiej chropowatej powierzchni łatwiej osiada kurz, który później zbiera się w jej zagłębieniach. Przygotujmy kolejną teksturę, która odwzoruje takie zagłębienia (Rysunek 17.15.8):

Wall Out
Włącz
That's Hoise
ń funkcję na
oi Crackle
i c

Rysunek 17.15.8 Proceduralna tekstura "szumu" — B.Noise.General.

To tekstura proceduralna typu *Stucci*. Nadałem jej nazwę **B.Noise.General**. Po określeniu typu tekstury zmień funkcję "szumu" (*Noise Basis*) na *Vornoi Crackle*, oraz zmniejsz jego rozmiar (*Noise Size*) do 0.02.

Rysunek 17.15.9 przedstawia szczegóły mapowania B.Noise.General w materiale:





W panelu *Map Input* pozostawiłem wartości domyślne. W panelu *Map To* włącz negatyw obrazu (*Neg*), by pola białe stały się przejrzyste. Włącz mapowanie koloru (*Col*), oraz odwrotności współczynnika rozpraszania (*Ref*) i odbicia (*Spec*) światła. Intensywność zabrudzeń można regulować wartością parametru *Col*.

Rysunek 17.15.10 przedstawia goleń z zabrudzeniami, stworzonymi przez teksturę B.Noise.General:



Rysunek 17.15.10 Efekt ostateczny — "zakurzony kawałek pomalowanego na srebrno metalu" (materiał B.Steel.Rough)

A co z powierzchniami odlewów? Zazwyczaj są bardziej chropowate. Rysunek 17.15.11 przedstawia zmodyfikowane mapowanie tekstury **B.Rough.Metal**, pozwalające uzyskać bardziej "chropowatą" powierzchnię:



Rysunek 17.15.11 Zmiana parametrów mapowania tekstury – dla "grubszych" nierówności

Przełączyłem w panelu *Map Input* tryb nakładania na *Nor*. Spowodowało to widoczne powiększenie wzoru nierówności. Zwiększyłem w *Map To* także ich skalę — do 0.5.



Rysunek 17.15.12 Efekt większej "chropowatości"

Rysunek 17.15.12 przedstawia test materiału z "pogrubionymi" nierównościami (bez zabrudzeń). Pamiętaj, że goleń podwozia P-40B wygląda na kutą, nie odlewaną, i na pewno nie miała tak chropowatej powierzchni. Potraktuj to jako test takiego efektu.

W modelu używam dwóch podstawowych "odmian stali":

- Materiału z teksturami B.Rough.Metal i B.Noise.General (por. Rysunek 17.15.10). To "chropowata stal": B.Steel.Rough;
- Tak samo ustawionego materiału, ale bez tekstury B.Rough.Metal: imitacji zakurzonej, gładkiej stali. Nosi nazwę B.Steel.Smooth;

993



Inną pochodną stali jest stalowa linka — skręcony splot wielu drobnych drutów (Rysunek 17.15.13):

Rysunek 17.15.13 Inna odmiana stali — pleciona linka (materiał B.Steel.Wire)

Nadałem temu materiałowi nazwę **B.Steel.Wire**. W stosunku do poprzednich materiałów, ten jest o wiele ciemniejszy (Rysunek 17.15.14):

Panels 3 2 2 2	₩●■*	
	Material Ramps	Shaders Texture
	P Ciemniejszy materiał	Oren-Na CRef 0.300 Tangent V Rough 2.00
	No Mist Env Ob Color Shad A 1.00	Shadow Mordian - Shado
		rms:0.400 I Only Shad
	Col R 0.000 / Mnie	jsze Cubic
	Spe G 0.000 rozpi	raszanie Exclusive Bias
	Mir B 0.000	Tralu 0.00 SBias 0.00
0	RGB HSV DVN A1.000	Amb 0.500 Emit 0.000
		LBias 0.00



Zmieniłem podstawową barwę materiału na czerń (*Col* w panelu *Materials*) i zmniejszyłem współczynnik rozpraszania światła do 0.3 (*Ref* w panelu *Shaders*). Materiał nie jest jednak zupełnie czarny, gdyż nadal używa spektrum barw z paska *Ramps* (por. Rysunek 17.15.3) — tyle, że ze zmniejszonym udziałem (*Factor* = 0.4).

Efekt "plecionki" uzyskamy za pomocą tekstury ukośnych pasków (Rysunek 17.15.15):





Jest to regularna (tzn. *NoiseSize* = 0) tekstura proceduralna typu *Wood*, z włączoną opcjami *Bands* i *Tri*. Nadałem jej nazwę **B.Stripes**. Jak za chwilę zobaczysz, przyda się także w kompozycji innych materiałów. Efekt "plecionej liny" w materiale **B.Steel.Wire** (por. Rysunek 17.15.13) uzyskałem za pomocą odpowiedniego mapowania tekstury **B.Stripes** (Rysunek 17.15.16):

Panels @ 🗟 🕽 🗶 💷	₩●■₺⊕ < 1 >	Nierówności i zmiana
V Preview	🔍 Map Input	Map To
	Glob Object Ob; UV Orco Stick Skalowanie Stress Tangent efekt lekko Flat Cube skręconych linii ofsV 0.000 Tube Sphe ofsV 0.000 ofsZ 0000 X Y Z sizeX 5.00 X Y Z sizeY 5.00 X Y Z sizeZ 10.00	Col Nor Csp Cmir Ref Spec Amb Hard RayMir Alpha Emit TransLu Disp Stencil Neg No RGB Mix # Col 0.400 Nor 10.00 0 R 0.850 Var 1.000 0 B Bardzo duże nierówności i umiarkowany kolor Var 0.000

Rysunek 17.15.16 Mapowanie tekstury B.Stripes, tworzące efekt "splecionych drutów" (materiał B.Steel.Wire)

Wzór tekstury na obiekt jest nałożony w 5-krotnym pomniejszeniu (*SizeX*, *SizeY* w panelu *Map Input*), i dodatkowo dwukrotnie rozciągnięty wzdłuż osi "linki" (*SizeZ* = 10), by plecionka wydawała się odpowiednio "skręcona". Paski **B.Stripes** modulują kolor materiału: w panelu *Map To* włączyłem opcję *Col* i nadałem teksturze jasnoszarą barwę. Zmniejszyłem także ich intensywność (*Col* = 0.400). Jednocześnie używam **B.Stripes** jako mapy nierówności (włączona opcja *Nor*). Ten efekt jest bardzo mocny (*Nor* = 10.00).

Innym przykładem wykorzystania tekstury **B.Stripes** jest specjalny materiał **B.Steel.Spring**, zmieniający zwykłą siatkę walca w coś, co z większej odległości wygląda jak sprężyna (Rysunek 17.15.17a):



Rysunek 17.15.17 Materiał, imitujący stalową sprężynę (B.Steel.Spring)

W tym przypadku użyłem **B.Stripes** jako tekstury przeźroczystości (*Alpha*). To rozwiązanie dla odwzorowania małych detali, bo w zbliżeniach (Rysunek 17.15.17b) nie wygląda już najlepiej.

Materiał **B.Steel.String** ma takie same ustawienia, jak "zwykła" stal (**B.Steel.Smooth**), ale dodatkowo włączoną przejrzystość (*ZTransp*) i wartość *A*(lpha) = 0 (Rysunek 17.16.18):



Rysunek 17.15.18 Podstawowe ustawienia materiału B.Steel.Spring

Cała nieprzejrzystość materiału B.Steel.Spring pochodzi od tekstury B.Stripes (Rysunek 17.15.19):



Rysunek 17.15.19 Modulowanie nieprzejrzystości za pomocą tekstury B.Stripes (materiał B.Steel.Spring)

 Uwaga: osią cylindra, na który nakładamy teksturę B.Steel.Spring, musi być oś Z. To zresztą domyślne ustawienie Blendera podczas dodawania nowej siatki do rysunku.

Paski **B.Stripes** zostały 5-krotnie powiększone na obwodzie walca (panel *Map Input*, *SizeX*, *SizeY* = 0.20). Jednocześnie uległy 20-krotnemu "ściśnięciu" wzdłuż osi **Z** (*SizeZ* = 20.00), by sprężyna wyglądała na odpowiednio gęsto "skręconą".

Najważniejszym parametrem, modulowanym przez teksturę, jest *Alpha* (panel *Map To*). Dzięki ustawieniu barwy tekstury na białą, środki pasków są zupełnie nieprzejrzyste. Dla wzmocnienia efektu włączyłem także silny efekt nierówności (*Nor* = 10.00), i zmieniłem funkcję "aplikacji" tekstury z *Mix* na *Difference* (Rysunek 17.15.19).

Na koniec zdecydowałem się wspomnieć o jeszcze jednym efekcie. Można go użyć do poprawienia wyglądu z większej odległości powierzchni, które mają wyglądać jak "posrebrzane" (czyli chodzi tu o alternatywne ustawienie materiału **B.Steel.Rough**). Fragmenty renderów, ilustrujące do tej pory tę sekcję, były wykonywane w mocnym, bezpośrednim oświetleniu. Pozwoliło mi to "wydobyć" z materiału metaliczne odbłyski. Niestety, zazwyczaj na scenie światło nie jest tak ostre. Aby elementy, pokryte materiałem **B.Steel.Rough**, nie wyglądały pod skrzydłem jak pomalowane szarą farbą, można spróbować użyć efektu "przesadzonego". Spróbuj wyłączyć przejścia między barwami w panelu *Ramps* (Rysunek 17.15.20):



Rysunek 17.15.20 Alternatywne, kontrastowe ustawienia paska barw (materiał B.Steel.Rough)

Oprócz tego warto także rozjaśnić materiał (Rysunek 17.15.21):

▼ Preview	Material	Ramps	▼ Shaders	
	[**	Oren-Nay = Ref 0.900 Rough:2.0	Tangent V
	No Mist	co jaśniejszy A 1.00	Wardiso = Spec 1.000	Shadow TraShado
	KOIC		rms:0.400	I Only Shad
	Col H	0.000		
	ope 5	0.000	GR:	Exclusive
		0.700	Tralu 0.00	SBias 0.00
0	RGB HSV DVN A	1.000	Amb 0.500	Emit 0.000
			LBias 0.00	

Rysunek 17.15.21 Alternatywne, kontrastowe ustawienia materiału B.Steel.Rough (c. d.)

Choć na podglądzie rezultat wygląda jak pomalowany w "łaty", to na podwoziu wygląda bardziej przekonująco (Rysunek 17.15.22):



Rysunek 17.15.22 Koło podwozia — częściowo w cieniu skrzydła

Taki "przejaskrawiony" materiał metalu dobrze zdaje egzamin na drobnych elementach, oglądanych z dużej odległości — właśnie takich, jak mechanizm podwozia.

Rozkład odcieni "przejaskrawionego" materiału, w stosowanym przez nas mapowaniu tekstury *Refl*, zależy od kształtu pokrytego nim obiektu. Na przykład, ten materiał daje złe efekty dla dużych, płaskich, prostokątnych powierzchni. Tak naprawdę o tym, czy coś będzie miało "metaliczny poblask", czy nie, decyduje nie tekstura, a zastosowane oświetlenie.

Rysunek 17.15.23 przedstawia układ oświetlenia, stosowany w scenie, którą pokazuje Rysunek 17.15.22 i wcześniej w tym tekście. Są to dwa podstawowe światła (1 i 2). Jedno to półsfera (*Hemi*), dająca odblaski (1), drugie — światło kierunkowe (*Sun*), odpowiedzialne za cienie. Na rysunku widać także kamerę i jej kierunek:



Rysunek 17.15.23 Położenie kamery i pomocniczego światła półsferycznego (Hemi)

Trick polega na dodaniu dodatkowego, słabego światła półsferycznego (3). Powinno być umieszczone nieco z boku, "za plecami" kamery (Rysunek 17.15.23a, b).

997

W ustawieniach materiału powróciłem do płynnych przejść pomiędzy odcieniami. (Czyli w zakładce *Ramps* przełączyłem z powrotem przejścia pomiędzy barwami z *Constant* na *Linear* — wycofałem to, co pokazuje Rysunek 17.15.20, na str. 995).

Rysunek 17.15.24a) przedstawia goleń podwozia, zrenderowaną metodą tradycyjną (tylko podstawowe światła 1 i 2). Rysunek 17.15.24b) przedstawia ten sam element, ale oświetlony dodatkowym światłem 3, o niewielkiej energii (0.2).



Rysunek 17.15.24 "Wydobycie" z materiału metalicznego połysku za pomocą dodatkowego źródła światła

Prawda, że goleń oświetlona światłem **3** wygląda lepiej? Światło **3** jest wielokrotnie słabsze od pozostałych, użytych w scenie, więc nie powoduje znacznych zmian w ogólnym oświetleniu samolotu. Ot, na poszyciu skrzydła pojawił się dodatkowy odblask. Jeżeli jednak ten efekt uboczny bardzo Ci przeszkadza, zawsze możesz ograniczyć działanie źródła światła do pojedynczej warstwy — tak, aby oświetlało wyłącznie metalowe elementy podwozia.

 Do uzyskania odblasków potrzebny jest odpowiedni kąt pomiędzy kamerą, powierzchnią modelu i źródłem światła półsferycznego (*Hemi*).

Zwróć uwagę, że światło **3** nie wytworzyło połysku na tarczy koła (Rysunek 17.15.24b). To dlatego, że ta powierzchnia jest ustawiona pod niewłaściwym kątem względem kamery/źródła światła. Jeżeli bardzo Ci na tym zależy, możesz oświetlić je kolejnym słabym światłem półsferycznym. Musiałbyś je umieścić niemal naprzeciw kamery (gdzieś "na godzinie 14") — aby jego promienie, odbite od tarczy koła, trafiły w obiektyw.

17.16 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 17.16.1):



Rysunek 17.16.1 Gumowa, zakurzona opona

Jak się zaraz przekonasz, taki efekt uzyskuje się za pomocą kilku tekstur proceduralnych. Na razie jednak zajmijmy się ustawieniem podstawowych parametrów materiału. Przejdź do panelu *Material buttons* w zestawie *Shading.* Rysunek 17.16.2 przedstawia podstawowe ustawienia kolorów i shaderów:



Rysunek 17.16.2 "Guma" - podstawowe ustawienia barw i shaderów

Podstawnym kolorem (*Material:Col*) materiału jest ciemnoszary (20%) — bo guma nie jest idealnie czarna. Ciemne są także światła odbłysków (*Material:Spe*) — to tylko 40% bieli.

Jeżeli chodzi o shadery, to dla rozpraszania światła zastosowałem typowy *Oren-Nayar*, o bardzo dużej chropowatości (*Rough* = 2.5) i zwykłym rozproszeniu światła (*Ref* = 0.8). Shaderem odblasków jest *Blinn*, z niewielkim natężeniem połyskliwości (*Spec* = 0.3), przeciętnej "twardości" odbłysków (*Hard* = 50) i zmniejszonym współczynniku *Refr* — do 3.0.

Rysunek 17.16.3 przedstawia, jak taki "surowy" materiał prezentuje się na renderze.



Rysunek 17.16.3 Podgląd na renderze

Resztę tej sekcji zajmują opisy poszczególnych tekstur proceduralnych, jakie nakładam na materiał. Zdecydowałem się je przedstawić w konwencji "krok po kroku". Dla każdej z nich pokażę trzy ilustracje:

- typ i parametry użytej tekstury proceduralnej (panele z zestawu Shading: Texture buttons);
- geometrię nałożenia na siatkę obiektu (Aby łatwo odróżnić obraz tekstury, na tej ilustracji pokazuję ją w kontrastowym, purpurowym kolorze);
- parametry materiału "modulowane" przez obraz tekstury, oraz podgląd renderu z ich rezultatem

Guma wydaje się zawsze "chropowata". Aby odtworzyć ten efekt, nałożę na materiał teksturę złożoną z drobnych punktów. Rysunek 17.16.4 przedstawia przykład takiego obrazu:



Rysunek 17.16.4 Losowy "szum" punktów - tekstura proceduralna B.Noise.General

Jest to tekstura proceduralna typu *Stucci*, z funkcją bazową (*Noise Basis*) ustawioną na *Vornoi Crackle*, zmniejszonym rozmiarem (*Noise Size*) do 0.020 i włączoną opcją *Hard Noise*. Nadałem jej nazwę **B.Noise.General**. (Ta sama tekstura została użyta do odwzorowania kurzu na stali — por. str. 991).

Rysunek 17.16.5 przedstawia nałożenie tekstury **B.Noise.General** na siatkę opony:



Rysunek 17.16.5 Mapowanie obrazu "szumu" (B.Noise.General) na siatkę opony

Rysunek 17.16.6a) pokazuje parametry, które moduluje tekstura "chropowatości" :



Rysunek 17.16.6 Przypisanie parametrów drobnych chropowatości do tekstury B.Noise.General

B.Noise.General wywołuje niewielkie nierówności (*Nor* = 0.30), oraz zmniejszenia połyskliwości (neg. *Spec*, *DVar* - 0.60). Rezultat działania tekstury przedstawia Rysunek 17.16.6b).

Kolejnym elementem, o który wzbogaci się nasz materiał, są smugi (a raczej "plamy") pyłu. Gdy na lotnisku jest sucho, na różnych elementach gumowych, w szczególności kołach podwozia, osiada sporo różnego rodzaju zabrudzeń.

Rysunek 17.16.7 przedstawia przykład obrazu, który można użyć w charakterze nieregularnych plam:



Rysunek 17.16.7 Losowe zabrudzenia — tekstura proceduralna B.Noise.Musgrave

Jest to tekstura proceduralna typu *Musgrave*, w odmianie *fBm*, z funkcją bazową przestawioną na *Vornoi F3*. Przestawieniu z wartości domyślnych uległy właściwie wszystkie jej parametry. Rozmiar "szumu" (*Noise Size*) uległ zmniejszeniu do 0.100, parametr *Nabla* funkcji bazowej zwiększyłem do 0.1, wartość *H* została wyzerowana. Tekstura otrzymała nazwę **B.Noise.Musgrave**.

Rysunek 17.16.8 przedstawia nałożenie tekstury B.Noise.Musgrave na siatkę opony:



Rysunek 17.16.8 Mapowanie obrazu "plam" (B.Noise.Musgrave) na siatkę opony

Zwróć uwagę na bardzo zmniejszoną (w panelu *Map Input* — Rysunek 17.16.8b) skalę tekstury — *SizeX*, *Y*,*Z* wynosi zaledwie 0.03. To bardzo powiększa obraz na obiekcie.

Rysunek 17.16.9 pokazuje użycie tekstury **B.Noise.Musgrave** do modulacji barwy opony:



Rysunek 17.16.9 Przypisanie parametrów "plam kurzu" do tekstury B.Noise.Musgrave

Zwróć uwagę w panelu *Map To* na włączony parametr *Neg*. Ten przełącznik powoduje, że czarne obszary oryginalnego obrazu tekstury są wypełniane kolorem "zabrudzeń", a białe są przejrzyste.
Nasz kawałek gumy jest już chropowaty i pobrudzony, ale nadal czegoś mu brakuje. Gdy przyjrzysz się oponie, zauważysz na pewno większe zużycie materiału wzdłuż bieżnika. Na kołach wczesnych wersji P-40 było to dodatkowo widoczne, bo w odmianach B i C stosowano gładkie opony (bez żadnego zagłębienia).

Aby nałożyć dodatkowe przetarcia i zabrudzenia nie na całą powierzchnię, a tylko na jej fragment, posłużymy się teksturą maski (*Stencil*). Potrzebujemy maski w kształcie bieżnika — rozmytego na brzegach pasa, biegnącego po obwodzie opony. Taki efekt można uzyskać za pomocą obrazu, który przedstawia Rysunek 17.16.10:



Rysunek 17.16.10 Kręgi bieżnika — tekstura proceduralna B.Tyre.Threads

Jest to tekstura proceduralna typu *Wood*, o regularnym kształcie (*Noise Size* = 0), przełączona w tryb koncentrycznych kół (*Rings*, albo *RingNoise*). Aby uzyskać płynne rozmycia na granicy każdego "słoja", użyłem przełącznika *Tri*. Nadałem jej nazwę **B.Tyre.Treads**.

Rysunek 17.16.11 przedstawia nałożenie tekstury **B.Tyre.Threads** na oponę:



Rysunek 17.16.11 Mapowanie obrazu bieżnika (B.Tyre.Threads) na siatkę opony

Zwróć na "spłaszczenie" obrazu tekstury do jednego wymiaru w panelu *Map Input* (Rysunek 17.16.11b). Tak łatwiej, przy odpowiednim doborze skali — *SizeX*, Y,Z — uzyskać "rozmyty" pasek o rozmiarze bieżnika. Co więcej, dzięki mapowaniu typu *Orco* ten układ będzie powtórzony na każdej oponie (np. koła głównego), niezależnie od jej rozmiaru!

Rysunek 17.16.12a) przedstawia ustawienia, przypisane teksturze B.Tyre.Threads w panelu Map To:

a) Map Input	Мар То	
Col Nor Cs Hard Ray Mir	braz mapowany jako <mark>spec Amb</mark> naska (<i>Stencil</i>) sLu Disp	
Stencil Neg No	RGB Mix =	
	Col 1.000	IUL
R 0.000	Nor 0.50	
G 0.000	Var 1.000	
B 0.000	— Disp 0.200 ■	
DVar 1.00	Warp fac 0.00 I	



Rysunek 17.16.12 Przypisanie teksturze B.Tyre.Threads roli "maski" (Stencil)

Działanie wszystkich kolejnych tekstur, które jeszcze dodamy do materiału, będzie ograniczone do obszaru wyznaczonego przez maskę **B.Tyre.Threads**. I o to chodzi — od tego momentu zaczynamy pracować tylko nad obszarem bieżnika. Nałożymy powtórnie te same tekstury chropowatości i plam, które zastosowaliśmy do całej powierzchni, by bieżnik wyglądał na bardziej zużyty. Przy okazji dodamy jeszcze efekt bruzd/rys, które także mogły się pojawić na oponie podczas eksploatacji.

Zacznijmy od dodatkowej chropowatości. Dodaj do materiału jeszcze raz teksturę **B.Noise.General** (Rysunek 17.16.13a — por. ze str. 999, Rysunek 17.16.4). Użyjmy takiego samego mapowania geometrii obrazu na powierzchnię opony, jak poprzednio (Rysunek 17.16.13b, por. ze str. 999). Aby nie zapomnieć o żadnym ustawieniu, najlepiej jest po prostu skopiować mapowanie tekstury (jak — zob. str. 965). Rezultat wygląda jednak inaczej, niż za pierwszym razem (Rysunek 17.16.13c):



Rysunek 17.16.13 Mapowanie dodatkowego "szumu" (B.Noise.General) na bieżnik opony

Tym razem, w wyniku działania maski, obraz "szumu" jest ograniczony do obszaru bieżnika (por. ze str. 999, Rysunek 17.16.5).

Teraz proponuję zastosować nieco odmienne przypisania parametrów, "modulowanych" przez teksturę **B.Noise.General** (Rysunek 17.16.14a) :





Obraz tekstury nadal zmienia lokalne wartości dwóch parametrów: nierówności (*Nor*) i połyskliwości (*Spec*). Zwróć jednak uwagę na różnice, w stosunku do poprzedniego wystąpienia (por. 999, Rysunek 17.16.6) :

- Obydwie zmieniane wartości (*Nor* i *Spec*) są odejmowane od wartości bazowych (użyłem operatora *Substract*). Wartości bazowe połyskliwości były już dość wysokie, przez co użycie operatora *Mix* dawało mizerny efekt. Dzięki odejmowaniu (*Substract*) efekt działania tekstury stanie się bardziej widoczny;
- W związku z użyciem operatora Substract, zmieniłem "znak" wartości połyskliwości pobieranych z tekstury — poprzednio były negatywne (Spec był w stanie "żółtym"), teraz są "pozytywne" (Spec jest w stanie "białym");
- Współczynnik wpływu zmiany połyskliwości, DVar, został zwiększony do 100%;

Rezultat nie jest specjalnie wyraźny (Rysunek 17.16.14b), ale to tylko efekt cząstkowy — zobacz dalej!

Na bieżniku mogą pojawić się, w wyniku jakichś "twardych" lądowań, podłużne, głębokie bruzdy. Do odwzorowania tego efektu posłużymy się teksturą **B.Rough.Metal** (Rysunek 17.16.15):

😑 🔻 Panels @ 🗟 🕻 🕻 📓	× • • • • • •	1 ->
V Preview	Texture	Colors Clouds
Mat	= TE:B.Rough.Metal	3 🗙 🙀 F Default Color Soft noise Hard noise
World	B.Noise.Genera	Texture Type
Lamp	B.Noise.Musgra B.Tyre.Threads	Clouds NoiseSize : 0.100 NoiseDepth: 3
Brush	B.Noise.Genera	
· · · · · · · · · · · · · · · · · · ·	D. hough . Meta	Tekstura
Alpha Alpha		
		Noise Basis
CONCERNMENT Default Vars		UVoronoi Crackle 📫 Nabla: 0.025 🕨

Rysunek 17.16.15 Losowy, "twardy szum" - tekstura proceduralna B.Rough.Metal

Ta tekstura została już użyta w materiale imitującym stal (por. str. 990, Rysunek 17.15.5). Powstała na bazie typu *Clouds*. Jako funkcji funkcję bazowego szumu (*Noise Basis*) używa *Vornoi Crackle*, o rozmiarze (*Noise Size*) równym 0.10.

Jak każda tekstura proceduralna, **B.Rough.Metal** jest teksturą trójwymiarową. Aby zmienić jej wzór w jednowymiarowe "bruzdy", przestawiłem mapowania wszystkich współrzędnych w panelu *Map Input* na **X** (Rysunek 17.16.16b,a) :



Rysunek 17.16.16 Mapowanie obrazu "bruzd" (B.Rough.Metal) na bieżnik opony

(Podobne "spłaszczenie" wykonaliśmy już dla tekstury **B.Tyre.Threads**, na str. 1001). W efekcie uzyskujemy w obszarze bieżnika drobne, nieregularne bruzdy (Rysunek 17.16.16c).

Rysunek 17.16.17a) przedstawia przypisania tekstury B.Rough.Metal w panelu Map To:



Rysunek 17.16.17 Przypisanie parametrów drobnych bruzd bieżnika do tekstury B.Rough.Metal

Najbardziej widoczny efekt powstaje w wyniku zwiększenia współczynnika rozpraszania światła *Ref*. W trybie *Add* powoduje wyraźne rozjaśnienie obszarów pomiędzy bruzdami. Dwa pozostałe parametry — nierówności (*Nor*) i wielkości odbić (*Hard*) — pełnią rolę uzupełniającą. Zwróć uwagę, że operują na negatywie tekstury (napisy na przełącznikach są żółte). Rezultat "wytartego bieżnika" przedstawia Rysunek 17.16.17b).

Bieżnik wygląda już całkiem poprawnie, na koniec pozostaje nałożyć na niego dodatkowe smugi kurzu. Tak jak poprzednio, użyjemy do tego tekstury **B.Noise.Musgrave** (por. str. 1000). Tym razem jednak zdecydowanie zmniejszymy skalę obrazu, by plamy stały się bardzo drobne (Rysunek 17.16.18b) :



Rysunek 17.16.18 Mapowanie obrazu dodatkowych "plam" (B.Noise.Musgrave) na siatkę bieżnika

Skala mapowania tekstury *Size X,Y,Z* została zwiększona do 10.00, przez co bieżnik pokrył się "chmurami" drobnych kropek (Rysunek 17.16.18c)¹.

W zasadzie parametry, "modulowane" przez drugie wystąpienie tekstury **B.Noise.Musgrave**, nie różnią się od pierwszego (Rysunek 17.16.19a) :



Rysunek 17.16.19 Przypisanie parametrów "plam kurzu" na bieżniku do tekstury B.Noise.Musgrave

Gdy porównasz te ustawienia z ustawieniami pokazanymi na str. 1000 (Rysunek 17.16.9a), przekonasz się, że są właściwie identycznie. Jedyną różnicą jest nieco mniejsza intensywność "mieszania" z kolorem bazowym (*Col* = 0.400, podczas gdy na str. 1000 jest = 0.500). To jednak już kwestia indywidualnego gustu — gdy chcesz mieć koło bardziej pobrudzone, zwiększ ten współczynnik.

Opona, którą pokazuje Rysunek 17.16.19b), wygląda już identycznie jak ta, pokazana na początku tej sekcji (str. 998, Rysunek 17.16.1). Te ilustracje różnią się tylko rozmiarem.

¹ Uwaga: Rysunek 17.16.18c) pokazuje negatyw oryginalnego obrazu tekstury — tzn. z włączonym przełącznikiem Neg w panelu Map To.

17.17 Skonfigurowanie materiału: skóra

W kabinie występują dwa elementy pokryte skórą: zagłówek i podkładka pod "spinką" pasów (Rysunek 17.17.1):



Rysunek 17.17.1 Elementy skórzane modelu

Zacznijmy może od konfiguracji czarnej, błyszczącej skóry nagłówka (Rysunek 17.17.2):



Rysunek 17.17.2 Czarna skóra (B.Black.Leather) - podstawowe ustawienia barw i shaderów

Nadałem temu materiałowi nazwę B.Black.Leather. Kolorem podstawowym (Material:Col) jest, oczywiście, czarny. Dodatkowo dodałem mu w niektórych miejscach brązowawy odcień za pomocą odpowiedniego spektrum barw (Rysunek 17.17.3). (Zagłówek na współczesnych zdjęciach wygląda, jakby był wypolerowany brązową pastą do butów). Materiał jest połyskliwy, więc kolor odblasków (Material:Spe) jest biały.

Rozpraszaniem światła steruje domyślny shader Labert, a połyskliwością -CookTorr. Materiał iest dość połyskliwy (Spec = 1.0), o stosunkowo "miękkich" odbłyskach (Hard = 30).

Rysunek 17.17.4 pokazuje, jak te ustawienia wyglądają na testowym renderze.







Rysunek 17.17.3 Spektrum barw

Rysunek 17.17.4 Próbny render

Na razie materiał błyszczy się jak buty oficera, wypolerowane na wielką paradę. Trzeba by go nieco zmatowić, ale tak — nieregularnie. Wykorzystamy do tego teksturę **B.Noise.Musgrave**, której używaliśmy już w innych materiałach (na przykład — oponach, por. str. 1000). Rysunek 17.17.12b) przedstawia parametry, sterujące geometrią nałożenia tego obrazu na zagłówek:



Rysunek 17.17.5 Mapowanie obrazu "plam" (B.Noise.Musgrave) na zagłówek

Jak wynika z opisu na ilustracji, to nic szczególnego — wykorzystałem po prostu domyślne ustawienia, przypisane przez Blender gdy dodałem teksturę **B.Noise.Musgrave** do materiału (Rysunek 17.17.12b). Dają mniej więcej taki "rozkład plam", o jaki chodziło (Rysunek 17.17.12c). Rysunek 17.17.13a) pokazuje, jakie parametry materiału są modulowane przez ten obraz:



Rysunek 17.17.6 Przypisanie modulacji odblasku i połysków do tekstury B.Noise.Musgrave

W panelu *Map To* włączyłem ("na żółto"!) przełączniki *Ref* i *Spec*. Efekt, który wywołują, nie jest specjalnie silny — raptem 20% (*DVar* = 0.2) "globalnych" wartości materiału. Na pierwszy rzut oka wydaje się, że obraz zagłówka na testowym rednerze nie uległ zmianie (Rysunek 17.17.13b). Przyjrzyj się jednak poprzedniej wersji (str. 1005, Rysunek 17.17.4), a dostrzeżesz różnicę!

Teraz nałóż na zagłówek kolejną teksturę "z odzysku": **B.Steel.Rough**. Wykorzystamy ją do stworzenia drobnych nierówności na powierzchni skóry (Rysunek 17.17.7):



Rysunek 17.17.7 Mapowanie obrazu B.Steel.Rough na zagłówek

Podobnie jak w przypadku poprzedniej tekstury, pozostawiłem domyślne parametry mapowania geometrii Rysunek 17.17.7b), bo tworzą na powierzchni zagłówka odpowiedni wzór "drobnych pęknięć" (Rysunek 17.17.7c). W panelu *Map To* przypiszemy do tego obrazu rolę tekstury nierówności (*Nor*) i — jeszcze raz — odbicia (*Ref*) (Rysunek 17.17.8a):



Rysunek 17.17.8 Przypisanie modulacji rozpraszania światła i nierówności do tekstury B.Steel.Rough

Nierówności, wywołane przez **B.Steel.Rough** nie są zbyt duże (*Nor* = 0.2), za to ta tekstura ma o wiele większy wpływ na rozpraszanie światła *Ref* (*DVar* = 0.6). Rysunek 17.17.8b) przedstawia ostateczny rezultat — skóra ma na nim ciekawą, lekko "chropowatą" powierzchnię. Taki właśnie efekt zaobserwowałem na zdjęciach z odrestaurowanego egzemplarza z Duxford. Egzemplarz był już trochę zużyty, więc przypuszczam, że na maszynach seryjnych zagłówek musiał wyglądać podobnie.

Zajmijmy się teraz drugą odmianą skóry, wykorzystaną w grubej podkładce pod "spinkę" pasów pilota (por. str. 1005, Rysunek 17.17.1). Nazwijmy go **B.Brown.Leather**, a ogólne parametry przedstawia Rysunek 17.17.9:



Rysunek 17.17.9 Ustawienia barwy materiału B.Brown.Leather

Mówiąc szczerze, materiał **B.Brown.Leather** powstał poprzez skopiowanie **B.Black.Leather**. Usunąłem tylko używane w **B.Black.Leather** tekstury. Od czarnej skóry materiał różni się, oczywiście, barwą — zmieniłem **Col**

na jasny brąz. (W pierwszym momencie może się wydawać, że nawet za jasnym). Zachowałem za to spektrum brązowych odcieni, ustalonych w panelu *Ramps* (por. str. 1005, Rysunek 17.17.3).

Typy shaderów nie uległy zmianie: są to nadal *Lambert* i *CookTorr*. Zmniejszyłem tylko intensywność odbłysków *Spec* (do 0.5) i zwiększyłem nieco "twardość" odbić *Hard* (do 50).

Rysunek 17.17.10 przedstawia pierwszy testowy render nowego materiału — jeszcze bez żadnej tekstury. Na razie wygląda płasko i nieciekawie.



Rysunek 17.17.10 Testowy render

Przypisanie pierwszej tekstury — **B.Noise.Musgrave** — po prostu przenieś, bez żadnych zmian, z materiału **B.Black.Leather** (Rysunek 17.17.11a). (Jej ustawienia są takie same, jak te, pokazane na str. 1006):

a) Texture B.Noise.Mu	TE:B.Noise.Musgrave	b) Na jasnym materiale wrbw. tekstury. jast	
	Clear 11 Fekstura B.Noise.Musgrave i jej mapowania są takie same, jak w B.Black.Leather	bardzo wyrażny	R

Rysunek 17.17.11 Rezultat zastosowania tekstury B.Noise.Musgrave

Dzięki drobnym różnicom w rozpraszaniu światła i połyskliwości, zdecydowanie poprawił się wygląd naszego kawałka skóry (Rysunek 17.17.11b). Trzeba przyznać, że na jasnym materiale ich działanie jest o wiele wyraźniejsze, niż na czarnym **B.Black.Leather**.

W charakterze drugiej tekstury zastosujemy, podobnie jak w przypadku czarnej skóry, obraz **B.Steel.Rough** (Rysunek 17.17.12a):



Rysunek 17.17.12 Mapowanie obrazu "pęknięć" B.Steel.Rough na podkładkę

Tym razem zmień w domyślnych ustawieniach, jakie przypisał Blender, skalę (**SizeX, Y, Z**) z 1.0 na 0.2 (Rysunek 17.17.12b). To zwiększy obraz "pęknięć", naniesiony na siatkę (Rysunek 17.17.12c).



Tekstura **B.Steel.Rough** moduluje wiele parametrów materiału **B.Brown.Leather** (Rysunek 17.17.13a):

Rysunek 17.17.13 Przypisanie modulacji barwy, odblasków, nierówności i połyskliwości do tekstury B.Steel.Rough

Oprócz tego, co zmieniała w poprzednim materiale (**B.Black.Leather**): nierówności (*Nor*) i rozpraszania (*Ref*), tutaj przyciemnia lokalnie (*Col*, kolor czarny), a także zmniejsza połyskliwość (*Spec*). Te trzy ostatnie wartości są modulowane w zakresie 40% (*Col* = 0.4, *DVar* = 0.4) wartości "globalnych".



Rysunek 17.17.14 pokazuje ten sam element, tylko z innej strony:

Rysunek 17.17.14 Rezultat - podkładka pod sprzączkę, pokryta materiałem B.Brown.Leather

Gdyby to była ważniejsza część, można by było tę skórę dalej "uszlachetniać". W tej chwili jednak ten drobny szczegół wygląda już całkiem dobrze. Nie przewiduję specjalnych zbliżeń w kabinie pilota, więc uważam że osiągnięty efekt jest wystarczający.

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, kurtki pilota, które przedstawia Rysunek 17.17.15:





Rysunek 17.17.15 Różnica pomiędzy skórą gładką i zmarszczoną

Rysunek 17.17.15a) to skóra, do którą przygotowaliśmy tak, jak opisałem na wcześniejszych stronach. Niestety, wygląda jak z podrzędnego serialu dla dzieci (typu "Noody"). Jest zupełnie nierealistyczna! Za to wersja, którą pokazuje Rysunek 17.17.15b) 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, mapowanej jako *Disp*(lace) (w panelu *Map To*), lub modyfikatora *Displace* (ale to wymaga gęstej siatki);
- za pomocą zwykłej tekstury nierówności (efekt trochę mniej od realistyczny, ale nie dodaje ścian);

Rysunek 17.17.15b) pokazuje rezultat wykorzystania tej trzeciej metody — zastosowania zwykłej tekstury nierówności. Zobacz dalej, jak go uzyskać. W zasadzie zarys fałd można byłoby namalować w GIMP lub Inkscape (wykorzystując filtry — por. str. 732). My pójdziemy tu jednak na zupełne uproszczenie, i wykorzystamy odpowiednio zamapowaną teksturę proceduralną (Rysunek 17.17.16):



Rysunek 17.17.16 Odpowiednio przygotowany wzór "marmuru" – tekstura proceduralna 000. Waves

Nadałem tej teksturze nazwę **000.Waves** (otrzymała prefiks "**000.**" a nie "**B**", bo należy do uniwersalnego modelu pilota, i nie jest specyficzna dla samolotu P-40B). Powstała na bazie typu *Marble* (marmur). Właściwie niewiele tu zmieniłem ustawienia domyślne. *NoiseSize* został zmniejszony do 0.1, a zwiększyłem *Turbulence*.

Rysunek 17.17.17 przedstawia nałożenie tekstury **000.Waves** na siatkę kurtki pilota:



Rysunek 17.17.17 Mapowanie "marmurkowego" wzoru (000.Waves) na kurtkę pilota

Trzeba przyznać, że testowa purpura robi dziwne wrażenie na tej figurce. Ale nie przejmujmy się tym szczegółem! Najważniejszym ustawieniem w panelu *Map Input* jest włączenie mapowania wg naprężenia siatki (*Stress* — Rysunek 17.17.17b). Siatka w "stanie spoczynku" nie ma żadnych naprężeń. Wewnątrz tej figury znajduje się jednak szkielet (*Armature* — por. str. 815). Zgięcie ręki do drążka czy do manetki gazu wywołało w siatce lokalne naprężenia, w wyniku czego pojawiają się w odpowiednich miejscach cienie (Rysunek 17.17.17c)!

Rysunek 17.17.18a) pokazuje, jakie parametry przypisałem do tak nałożonej tekstury:

a) Map Input	Мар То		b	(Alashi
Col Nor Csp C Hard Ray rir Tekstu	ra nierówności	Amb Disp		
Stencil Neg No RGB	Mix	÷		
	Col 1.000			A K
R 1.000	Nor 4.00			
G 1.000	Var 1.000			
B 1.000	Disp 0. dużej "sile"			
DVar 1.00	Warp fac 0.000			ALL

Rysunek 17.17.18 Przypisanie parametrów do tekstury 000.Waves

Obraz **000.Waves** pełni tu przede wszystkim rolę tekstury nierówności, o bardzo dużej "sile" (*Nor* = 4.0), pozwalające uzyskać efekt pokazywany przez Rysunek 17.17.18b). Mapowania *Ref* i *Spec* nie są już tak ważne. Rezultat nie zależy specjalnie od rozmiaru obrazu, który można ustawić w panelu *Map Input*. Jest za to bardzo "czuły" na jego przesunięcia *ofsX, Y, Z*. Rysunek 17.17.19 przedstawia rozkład fałd i odblasków, uzyskany po przesunięciu tekstury o 20% w kierunku Y (*ofs* Y):



Rysunek 17.17.19 Efekt przesunięcia tekstury 000.Waves o 20%

Taka, stosunkowo niewielka zmiana (Rysunek 17.17.19a), potrafi zupełnie przestawić rezultat (por. Rysunek 17.17.18b) i Rysunek 17.17.19b). W dodatku, "po drodze" wystąpiły jeszcze inne, zupełnie inne układy fałd i odblasków. Zobacz, na przykład, jak wyglądają dla ofsY = 10% (Rysunek 17.17.20):



Rysunek 17.17.20 Efekt przesunięcia tekstury 000.Waves o 10%

Musisz sam dobrać sobie układ fałd, który najbardziej Ci odpowiada. Aby uzyskać jakieś drobne zmiany, a nie tak poważne, jak te powyżej, zmieniaj wartość przesunięć o dziesiąte części % (czyli o wartości rzędu 0.001).

17.18 Skonfigurowanie materiału: tkanina (pasów)

Pasy fotela pilota były wykonane z grubej tkaniny. Miały kolor białokremowy (Rysunek 17.18.1):



Rysunek 17.18.1 Pasy, przymocowane do fotela pilota — przykład materiału, imitującego tkaninę

Podstawowe ustawienia materiału B.White.Belt wyglądają tak, jak jest to pokazuje Rysunek 17.18.2:





Pasy były w zasadzie białe — dlatego taką barwę ustawiłem w sekcji *Material:Col*. Dodatkowo "złamałem" podstawowy kolor, dodając w spektrum barw odcienie kremowe (Rysunek 17.18.3a), a nawet (z przodu) półprzeźroczystą żółć (Rysunek 17.18.3b). W rezultacie pasy wydają się kremowobiałe. Kolor odblasków (*Material:Spe*) jest za to ciemnoszary, by tkanina nie wydawała się zbyt połyskliwą.

Dla światła rozproszonego użyłem shader Oren-Nayar, o niewielkiej chropowatości (*Rough* = 0.2) i maksymalnym współczynniku odbicia światła (Ref = 1.0). Shaderem odblasków jest Blinn, z małą połyskliwością (Spec = 0.2) i bardzo małą "twardością" odbłysków (*Hard* = 16). Za to *Refr* jest ustawione na max. (10.0). Rysunek 17.18.4 przedstawia efekt na próbnym renderze:

Spec Ramp
el Linear :
= - -
]
200 A
) — ——
)

Rysunek 17.18.3 Spektrum barw



Rysunek 17.18.4 Efekt na renderze

Pierwszą teksturą, którą dodamy do naszego materiału, są smugi zabrudzeń. (Choć pasy były fabrycznie białe, to na pewno w trakcie eksploatacji szybko "łapały" różne zanieczyszczenia).

Panels @ 🗟 隆 其 🔤	※◎₫ ∛⊕ < 1 →	
V Preview	Texture Colors	▼ Musgrave
Mat	TE:B.Linen.Rough 3 X 🚱 F	Multifractal
World	B.Linen.Rough	H: 1.000 Lacu: 2.000
Lamp	Nodes	Octs: 2.000 Zwiększ NoiseSize
Brush		do 0.5
Aluba	Wybierz typ	iScale: 1.000 Noise Size: 0.500
Alpha	Musgrave	Noise Basis
Default Vars		Blender Original 🗢 🔹 Nabla: 0.025 🔹

Rysunek 17.18.5 przedstawia obraz, który wykorzystamy do stworzenia wrażenia smug:

Rysunek 17.18.5 Nieregularne plamy na tkaninie - tekstura proceduralna B.Linen.Rough

Jest to tekstura proceduralna typu *Musgrave*, w odmianie *Multifractal*. Większość ustawień widocznych w panelu *Musgrave*, to wartości domyślne. Zwiększyłem tylko dwukrotnie wartość *NoiseSize* (do 0.5), by generowane plamy stały się większe. Nadałem tej teksturze nazwę **B.Linen.Rough** (bo zastosujemy ją za chwilę także do odwzorowania chropowatości).

Rysunek 17.18.6 przedstawia mapowanie tekstury **B.Linen.Rough**. Przygotowałem je w tym przypadku tak, by wyglądało jak przypadkowe zabrudzenia pasów:



Rysunek 17.18.6 Mapowanie obrazu "smug" (B.Linen.Rough) na siatkę pasów

Jak widzisz, nie stosuję jakiegoś mapowania *UV*, tylko domyślne *Orco*. (Dzięki temu nie muszę przygotowywać rozwinięć pasów). Zmieniłem tylko tryb nakładania — z *Flat* na *Tube*, oraz zmniejszyłem dwukrotnie skalę (*Size** – Rysunek 17.18.6b), by powiększyć rozmiary ciemnych i jasnych plam (Rysunek 17.18.6c).

Rysunek 17.18.7a) pokazuje, jaką barwę przypisaliśmy smugom, a Rysunek 17.18.7b) — rezultat:

Włącz modula- cję barwy Col	a) Map Input Col Nor Csp (Hard RayMir Alph	Map To Cmir Ref Spec Amb a Emit TransLu Disp	b Smugi zabrudzeń
Kolor plam: ciemnobrązowy	Stencil Neg No RGB H 0.148 Image: Stence Image: Stence S 0.316 Image: Stence Image: Stence V 0.224 Image: Stence Image: Stence DVar 1.00 Image: Stence Image: Stence	Mix = Col 0.600 Nor 0.32 Var 1.000 Disp 0.200 Warp fac 0.00	

Rysunek 17.18.7 Przypisanie teksturze B.Linen.Rough roli zabrudzeń

Nasz pas jest już pobrudzony, ale trzeba go pokryć regularnymi nierównościami, imitującymi włókna materiału. Przygotowałem odpowiednią teksturę (Rysunek 17.18.8):

😑 🔻 Panels 🚱 🗟 🔍 🔍 📓	₩@200 (1)	
▼ Preview	Texture Colors	Vood
Mat World Lamp Brush	TE:B.Linen.Stripe S.Linen.Rough B.Linen.Stripe Wood Nodes Wybierz typ Musgrave	Bands Rings BandNoise RingNoise Sin Saw Tri Soft noise Hard noise Image: NoiseSize : 0.150 Image: Turbulence: 5.00 Image: Turbulence: 5.00 Image: Turbulence: 5.00 Image: Turbulence: 5.00 Image: Turbulence: 5.00 Image: Turbulencence: 5.
Default Vars		Noise Basis Blender Original 🗢 < Nabla: 0.025 🔸

Rysunek 17.18.8 Regularne prążki włókien - tekstura proceduralna B.Linen.Stripes

Jest to obraz, wygenerowany przez teksturę proceduralną typu *Wood*, o niemal regularnym kształcie (*Wood:NoiseSize* = 0.150). (Pozostałe przełączniki pozostawiłem w ich ustawieniach domyślnych). Nadałem jej nazwę **B.Linen.Stripes**.

Rysunek 17.18.9 przedstawia parametry nałożenia tej tekstury na siatkę pasów, oraz wzór, który w ten sposób uzyskaliśmy:



Rysunek 17.18.9 Mapowanie obrazu włókiem (B.Linen.Stripes) na siatkę pasów

Podobnie jak w przypadku tekstury smug, stosujemy tu domyślne mapowanie *Orco*, przełączając tylko tryb na *Tube*. Zwróć za to uwagę na różnorodne wartości *SizeX*, *Y*, *Z* (Rysunek 17.18.9b). Te parametry pozwoliły pokryć powierzchnie siatek serią drobnych, równoległych pasków. W zasadzie prawie wszędzie ich wzór jest taki, jak trzeba — regularny. Tylko miejscami przekształca się w coś innego (Rysunek 17.18.9c).

Rysunek 17.18.10a) pokazuje parametry, które zamieniają obraz B.Linen.Stripes w teksturę nierówności:





W panelu *Map To* włączyłem ("na żółto"!) przełącznik *Nor*, oraz ustawiłem odpowiednio duży rozmiar nierówności (*Nor* = 1.28). Rysunek 17.18.10b) pokazuje render, na którym sprawdzałem te ustawienia. Zwróć uwagę, że na pasie po prawej nierówności ułożyły się w dziwny wzór. Tym niemniej — jakoś mnie to nie razi.

a) Map Input b Map To B.Linen.Ro Glob Object Ob B.Linen.St Win Nor UV. Orco Stick Refl B.Linen.St Pojawiła się From Dupli Stress Tangent odpowiednia Tylko to "kratka" Cube Flat 00 mapowanie hO Ponownie użyta Tube Sphe uległo tekstura zmianie າກ **B.Linen.Stripes** ٧ sizeX -1.50 Х Ζ Х sizeY 50.00 Y Ζ sizeZ 30.00

Przeplot włókien tkaniny uzyskamy, nakładając jeszcze raz tę samą teksturę nierówności (Rysunek 17.18.11):

Rysunek 17.18.11 Przeplot włókien: ta sama tekstura B.Linen.Stripes, tylko że skierowana w przeciwną stronę

Wstaw zaraz za pierwszym wystąpieniem tekstury **B.Linen.Stripes** następne Rysunek 17.18.11a). Jedyna różnica w stosunku do poprzedniego polega na zmianie przypisania współrzędnych w panelu *Map To* — porównaj Rysunek 17.18.11a) i Rysunek 17.18.9a). Na testowym renderze pojawił się charakterystyczny wzór "kratki" przeplotu włókien (Rysunek 17.18.11b).

Ostatnią teksturą, jaką zastosujemy w tym materiale, będą drobniutkie nierówności. Użyjemy w tym celu jeszcze raz tekstury **B.Linen.Rough** (por. str. 1013, Rysunek 17.18.5. Po prawdzie właśnie od tego zastosowania pochodzi jej nazwa). Wstaw ją jeszcze raz, na końcu stosu tekstur (Rysunek 17.18.12a):



Rysunek 17.18.12 Tekstura B.Linen.Rough, nałożona tym razem bardzo "drobnoziarniście"

Tym razem w panelu *Map To* wpisz bardzo duże wartości *Size** (= 20). Spowoduje to drastyczne zmniejszenie plamek obrazu, nanoszonego na pasy (Rysunek 17.18.12b).

Ten wzór przekształć w drobne nierówności, o niewielkim rozmiarze (Rysunek 17.18.13a):



Rysunek 17.18.13 Mapowanie B.Linen.Rough, tym razem w charakterze drobnych chropowatości

Tym razem ustawiłem *Map To:Nor* na wartość na o wiele mniejszą, niż dla włókien: 0.32. Rysunek 17.18.13b) pokazuje testowy render: układ włókien na tkaninie uległ niewielkiemu rozmyciu.

Rysunek 17.18.14 przedstawia efekt, który uzyskujemy stosując gotowy materiał B.White.Belt:



Rysunek 17.18.14 Efekt zastosowania gotowego materiału B.White.Belt

Jeżeli uważasz, że pasy nie powinny być aż tak brudne, zmniejsz intensywność (wartości *Map To:Col*) tekstury smug, na przykład do 40% (Rysunek 17.18.15a):



Rysunek 17.18.15 Ten sam materiał, ale w ostrzejszym świetle i ze zredukowanym wpływem "brudu" (B.Linen.Rough)

Rysunek 17.18.15b) przedstawia rezultat zastosowania materiału **B.White.Belt** z tym ustawieniem, oraz nieco jaśniejszym światłem kierunkowym (energia **Sun** zwiększona z 0.7 do 1.0). Na tym renderze materiał pasów zasługuje na swoją nazwę (**B.White.Belt** — bo faktycznie wyglądają na realistycznie białe).

17.19 Załadowanie wtyczki (plugin) tekstury

Obraz, wykorzystywany przez teksturę, nie musi zawsze być plikiem rastrowym (*Image*). To tylko jeden z wielu dostępnych w Blenderze typów tekstur. W szczególności twórcy programu umożliwili użytkownikom tworzenie własnych "wtyczek" (*plugins*), generujących tekstury. Wtyczka Blendera to odpowiednio przygotowany, binarna biblioteka funkcji (*Dynamic Linking Library* — plik *.*dll* — pod Windows, *Shared Objects* — plik *.*so* — pod Linuxem). Aby z niego skorzystać, zmień najpierw typ tekstury na *Plugin* (Rysunek 17.19.1):



Rysunek 17.19.1 Zmiana typu tekstury na "wtyczkę" (Plugin)

Spowoduje to zmianę układu paneli: w zestawie pojawi się nowa panel *Plugin*. Początkowo zawiera tylko przycisk *Load Plugin* (Rysunek 17.19.2). Naciśnij go:

!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!</th <th>🛃 👶 🌍 🛛 🔍 1</th> <th>Þ</th> <th></th>	🛃 👶 🌍 🛛 🔍 1	Þ	
▼ Preview Mat World Lamp Brush Alpha Default Var	Texture TE:B.Skin.Decals Skin.Nor-Det B.Skin.Nor-Det B.Skin.Ref B.Skin.Ref B.Skin.Col B.Skin.Decals B.Skin.Holes	Colors 3 X P F Texture Type Plugin + Nodes	Plugin Load Plugin Naciśnij ten przycisk

Rysunek 17.19.2 Panel Plugin - stan początkowy

W oknie wyboru plików wskaż plik wtyczki (w tym przykładzie to wtyczka svgtex.dll) (Rysunek 17.19.3):



Rysunek 17.19.3 Wskazywanie pliku wtyczki

Wskazując wtyczkę musisz używać pełnych ścieżek dostępu (przełącznik Relative Paths musi być wyłączony). W przeciwnym razie Blender nie odnajdzie pliku *.dll!

Jeżeli wtyczka załadowała się poprawnie, powinna wyświetlić w panelu *Plugin* jakieś dodatkowe kontrolki (Rysunek 17.19.4):

(≘: ▶ @⊠@\(⊥⊠) ※●	🛃 👶 🎯 🦳 🔍 1 🔸		
▼ Preview Mat World Lamp Brush Alpha Default Var	Texture Colors	Plugin Load Plugin tnetWM\CD\Source\Textures\Plugins\svgtex.dll Extend Repeat Clip Interpol: Fid: 0.00 Of.LoD: Force Df.LoD Mem Max 20 & Used: 0 svgtex 1.0 File name Enable	

Rysunek 17.19.4 Przykładowa zawartość panelu Plugin po pomyślnym uruchomieniu wtyczki

Konieczność wpisywania pełnych nazw ścieżek jest dość uciążliwa, gdy musisz przenieść plik na inny komputer. Tam odpowiednia wtyczka może być w innym folderze, i trzeba parametry tekstur typu *Plugin* ustawiać od nowa.

Najwygodniej byłoby wpisywać w panelu *Plugin* samą nazwę pliku *.*dll* — i niech komputer go sam znajdzie! To, na szczęście, jest możliwe. Wystarczy zebrać wszystkie wtyczki, z których korzystasz, w jedno folder, a następnie dopisać jego nazwę do zmiennej środowiskowej **Path** (Rysunek 17.19.5):

a)	Zmienne środowiskowe
	Zmienne użytkownika dla W4979721
	Zmienna Wartość
	INCLUDE C:\Program Files\Microsoft Visual Studio
	LANG C LIB C:\Program Files\V b) MVXAPPPATH C:\Program Files\ <mark>V Edytowanie zmiennej systemowej</mark>
	MVXCONFIGUR v12 Nowa Nazwa zmiennej: Path
Podświetl na liście	Zmienne systemowe (C:\Program Files\Blender\plugins\texture)
zmienną Path	Zmienna Wartość OK Anuluj
	NUMBER_OF_P 1 OS Windows_NT Path C:\WINDOWS\system32;C:\WINDO PATHEXT .COM;.EXE;.BAT;.CMD;.VB5;.VBE;.35 PROCESSOR_A x86 Nowa Edytui
	OK Anuluj



W Windows, aby otworzyć okno Zmienne środowiskowe, musisz przejść do Właściwości ikony Mój Komputer¹. Tam, w oknie dialogowym Właściwości systemu przejdź do zakładki Zaawansowane i naciśnij umieszczony u dołu przycisk Zmienne środowiskowe. W oknie dialogowym podświetl na liście Zmienne systemowe zmienną **Path** i naciśnij przycisk Edytuj (Rysunek 17.19.5a). W oknie dialogowym Edytowanie zmiennej systemowej możesz dopisać ścieżkę do folderu z wtyczkami Blendera (Rysunek 17.19.5b). Zwróć uwagę, że separatorem poszczególnych ścieżek jest średnik. Nie zapomnij sprawdzić, czy jest przed i za dopisaną przez Ciebie nazwą!

¹ Alternatywną metodę dotarcia do okna *Właściwości systemu* pokazuje, na str. 24, Rysunek 1.4.1.

Zazwyczaj zawartość zmiennej **Path** jest dość długa, więc, aby łatwiej było ją edytować, skopiuj ją do schowka a ze schowka do jakiegoś edytora tekstów — chociażby Notatnika (Rysunek 17.19.6):

Bez tytułu - Notatnik Koniecznie włącz tę opcję! Plik Edycja Format Widok Pomoc Koniecznie włącz tę opcję!	
<pre>\$SystemRc Zawijania wierszy { /stemRoot\$;\$SystemRoot\$\System32\Wbem;C:\Program Files\Mic Czcionka Files\Mic Czcionka Files\GTK\2.0\bin;C:\Program Files\Python;C:\Program Files\Microsoft Visual Studio\VC98\Bin;C:\Program Files\Blender\plugins\texture;</pre>	8

Rysunek 17.19.6 Wygodniejsze narzędzie do zmiany wartości Path - Notatnik

W Notatniku zobaczysz cały tekst, a nie tylko jego fragment (nie zapomnij tylko o włączeniu opcji *Zawijanie wierszy*). Zmienioną wartość zmiennej **Path** skopiuj z powrotem do schowka i wklej do pola *Wartość zmiennej* w oknie *Edytowanie zmiennej systemowej*.

Po tej zmianie nie zaszkodzi zamknąć Blender i powtórnie go otworzyć (by mógł się uruchomić ze zmienionymi zmiennymi środowiskowymi). Od tej chwili będziesz mógł wskazywać wtyczkę, posługując się samą nazwą, bez ścieżki (Rysunek 17.19.7):

	Blender sam wyszuka ten plik – w folderach wyliczonych w Path		
Preview Mat World Lamp Brush Alpha. Default Var	Texture TE:B.Skin.Nor-Det B.Skin.Nor-Det B.Skin.Nor-Blu B.Skin.Ref B.Skin.Col B.Skin.Col B.Skin.Holes	Colors	Plugin Ioad Figin svgtex.dll Fid: 0.00 Alpha: 1. Fid: 0.00 Mem Max 20 Svgtex.l.0 File name Enable

Rysunek 17.19.7 Wtyczka, zapamiętana w pliku Blendera w sposób niezależny od konfiguracji komputera

Taki sposób zapisu pozwala na wygodne przenoszenie pliku Blendera pomiędzy różnymi konfiguracjami. Jeżeli korzystasz jednocześnie z Linuxa i z Windows, możesz nawet "pójść na całość" i zmienić pod Windows rozszerzenie pliku wtyczki z *.*dll* na *.so. Blenderowi jest w istocie wszystko jedno, jakie rozszerzenie ma plik wtyczki, a w ten sposób jej nazwa jest taka sama w obydwu systemach. Nie będę tu opisywał, jak zmienić ścieżkę **Path** pod Linuxem, ale zasada jest taka sama.

17.20 Wtyczka svgtex – zasada działania i obsługa

Svgtex to napisana przeze mnie wtyczka Belndera, podstawiającą jako teksturę zawartość pliku *.*svg*. Plik wtyczki — *svgtex.dll* — znajdziesz wśród materiałach dodatkowych, towarzyszących tej książce (w folderze *source\textures\plugins*). Możesz go wgrać w katalog Blendera, gdzie jest miejsce dla wtyczek tekstur: w podka-talog *plugins\texture*. Uruchomienie wtyczki — patrz str. 1017. Przykłady użycia — str. 613.

Zasada działania **svgtex** nie jest jakąś szczególną tajemnicą. Podczas renderowania Blender żąda od wtyczki podania barwy konkretnego teksela (piksela tekstury). **Svgtex**, aby "udzielić odpowiedzi", tworzy w pamięci rastrowe fragmenty obrazu. Użyje ich także przy okazji następnych "pytań" Blendera. Każdy z tych fragmentów ma rozmiar 256x256 pikseli — tak, by zajmował w RAM nie więcej niż 1 MB. W ten sposób oryginalny obraz jest dzielony jak szachownica, na kwadratowe, jednolite obszary. **Svgtex** przygotowuje i przechowuje tylko takie "pola" tej szachownicy, których zażądał od niego Blender. Dodatkowo, jeżeli całe takie "pole" jest zupełnie jednolite, zamiast 1 MB danych przechowywany jest pojedynczy, reprezentujący je piksel¹.

Svgtex tworzy fragmenty wielu wersji obrazu, różniących się rozdzielczością. Każda kolejna wersja ma czterokrotnie większą liczbę pikseli od poprzedniej. Pierwszy obraz mieści się w pojedynczym fragmencie — czyli jest "wpasowany" w obszar 256x256 pikseli. Następny ma wymiary 512x512 pikseli, i składa się z 4 fragmentów. Kolejna wersja to obraz 1024x1024 pikseli (złożony z 16 fragmentów o rozmiarach 256x256). W grafice komputerowej o takim szeregu obrazów mówi się, że są uporządkowane według "poziomu szczegółowości" (*Level of Details* — LoD) . Poziom ten określa się jako liczbę całkowitą. Na potrzeby **Svgtex** przypisałem LoD = 0 dla obrazu 256x256, LoD = 1 dla 512x512, LoD = 2 dla 1024x1024, itd. W odpowiedzi na żądane Blendera **svgtex** określa wymagany LoD obrazu, i zwraca piksel z jego odpowiedniej wersji. Jeżeli żądany fragment jeszcze nie istnieje — jest w tym momencie generowany.

Aby svgtex działał poprawnie, musisz w ustawieniach renderingu (panel *Renderer*) włączyć opcję OSA. W przeciwnym razie wyświetlany będzie obraz o niewielkiej rozdzielczości!

Ten "technologiczny" wstęp był konieczny, abyś mógł zrozumieć działanie niektórych kontrolek, udostępnianych przez wtyczkę (Rysunek 17.20.1):



Rysunek 17.20.1 Kontrolki wtyczki svgtex.

¹ Ten pomysł rzucił w lutym 2008, na Blender Artists Forum, **Marcelo de Gomensoro Malheiros**. Opublikował źródła "studyjnego" komponentu, który wykorzystywał do rysowania bibliotekę **AGG** (kilka lat wcześniej opracował ją **Maxim Shemanarev**). Do wczytywania plików XML zastosowano parser **EXPAT** (autor — **James Clark**). Komponent nazywał się **vectex** i z całego bogactwa formatu SVG potrafił rysować tylko kontury (z wypełnieniem lub bez). Mój wkład to uporządkowanie kodu i rozbudowa interpretera, umożliwiająca rysowanie większości elementów wymaganych przez standard SVG (w tym także tekstu).

Rodzaj i położenie kontrolek wtyczki **svgtex** w panelu *Plugin* jest efektem wielu bolesnych kompromisów z ograniczeniami, jakie narzuca struktura danych Blendera (zdefiniowana w *plugin.h*). Na przykład — pole na nazwę pliku *.*svg* powinno być pierwszym polem tej wtyczki. Niestety, okazuje się, że można je "wepchnąć" wyłącznie pewnym, nie opisanym w dokumentacji Blendera sposobem, i w dodatku tylko na ostatnie miejsce panelu (po prawej, u dołu).

Poniżej podaję znaczenie wszystkich kontrolek, w tej kolejności, jak występują w paneli:

- Extend/Repeat/Clip: Przełącznik określający, co ma być przekazane do Blendera, gdy współrzędne tekstury wykraczają poza granice obrazu. (Analogiczny, nieco bardziej rozbudowany przełącznik jest także w teksturach opartych o obrazy rastrowe: zobacz zestaw Shading:Texture buttons, panel Map Image). Znaczenie poszczególnych alternatyw:
 - Extend: na zewnątrz obrazu są powielone piksele z jego krawędzi;
 - Repeat: obraz jest powielany w nieskończoność;
 - o *Clip*: poza obszarem obrazu nie jest nic rysowane (obraz jest absolutnie przejrzysty);
- Interpol (interpolation) : Poziom wygładzania pikseli obrazu. Gdy Interpol = 0, wykorzystywana jest wartość najbliższego piksela. W takim przypadku granice obrazów mogą być nieco "ziarniste". Dla wartości = 1, wykorzystywana jest średnia z 4 najbliższych pikseli, a dla wartości Interpol = 2 z 16 sąsiednich. Poziom Interpol = 3 oznacza uśrednienie z 64 sąsiednich pikseli. Każde zwiększenie poziomu interpolacji w sposób widoczny "rozmywa" ostre granice kolorów na teksturze. Wydłuża także czas obliczeń (każdy kolejny poziom wymaga ich cztery razy więcej, niż poprzedni). Przypuszczam, że zazwyczaj będziesz używał dla Interpol wartości = 1 lub 0;
- *Fid* (*fidelity*) : Poziom doboru rozdzielczości (LoD) obrazu. Możliwe jest ustawienie od -5.0 do 0.0, przy czym wartość = 0.0 oznacza największą dokładność. (Przy wartości = 0.0 wykorzystywane są wyłącznie piksele z obrazu o LoD >= żądanego przez Blender. Z takim ustawieniem komponent działa najszybciej). Każdy niższy poziom *Fidelity* oznacza wyznaczenie piksela poprzez wyciagnięcie średniej z dwóch obrazów o najbliższych poziomach szczegółowości (LoD). Na przykład, załóżmy że svgtex określił dla konkretnego żądania Blendera optymalny LoD = 4 (4096x4096 pikseli). Jeżeli *Fid* = -0.5, to zwracana Blenderowi wartość teksela jest w 50% wartością z obrazu o LoD = 4, a w 50% z obrazu o LoD = 3 (2048x2048). To przypadek klasycznej techniki "mipmapowania". Gdyby *Fid* = -0.2, wówczas zwracany Blenderowi teksel byłby średnią pochodzącą w 80% z pikseli tekstury 4096x4096, i 20% tekstury 2048x2048. Gdyby *Fid* = -1.6, uśrednianie rozegrałoby się pomiędzy pikselami z obrazu o rozdzielczości 2048x2048 (LoD = Max(LoD + *Fid*) = Max(4 1.6) = 3) i 1024x1024 (LoD = Max(LoD + *Fid*) = Min(4 1.6) = 2). W każdym razie używaj tego ustawienia z umiarem dla map nierówności gdzieś pomiędzy 0.0 a -1.0. Dla zwykłych tekstur barwy pozostaw *Fid* = 0.0;
- barwa tła: Standardowy przybornik Blendera do wyboru barwy tła. Ta barwa jest widoczna tylko wtedy, gdy oryginalny obraz *.svg miał tło przejrzyste. Uwaga: dla rysunków na przejrzystym tle, takich jak oznaczenia samolotu, barwa tła delikatnie wpływa na kontury znaków rozpoznawczych i innych emblematów, nawet gdy ustawisz *Alpha* na 0.0. Stąd do mapowania *decals.svg* ustawiłem ją nie na czerń, a 40% szarość (por. Rysunek 10.8.1).
- Alpha: Dodatkowy parametr barwy tła. Określa poziom nieprzejrzystości. Gdy Alpha = 0.0 tło jest zupełnie przejrzyste;
- Df.LoD (Default Level of Details) : po włączeniu Enabled, svgtex wygeneruje wszystkie fragmenty obrazu o LoD <= DfLoD. Ustawiaj zazwyczaj na poziomie 1-2, gdyż inaczej możesz za długo czekać na pojawienie się podglądu tekstury w panelu Preview;
- Force Df.LoD: Wymusza na svgtex wyłączne korzystanie z wersji obrazu o LoD = DfLoD. Sugestie dotyczące rozdzielczości, dostarczane przez Blender, są ignorowane;
- Mem Max: Maksymalna ilość RAM (w MB), jaką mogą zająć fragmenty obrazów, zarządzanych przez svgtex. Dobieraj tę wartość uważnie, obserwując wartość umieszczonego obok pola Used. Jeżeli ustawisz MemMax na poziomie mniejszym od typowych wartości Used, zmusisz wtyczkę do "porzuca-

nia" fragmentów, które jeszcze się jej przydadzą podczas tej samej sesji renderingu. W takiej sytuacji może tylko, gdy nadejdzie żądanie Blendera, powtórnie wygenerować obraz. To bardzo wydłuża czas renderingu;

- Used: Pole wyłącznie informacyjne: aktualna ilość pamięci RAM, zajętej przez fragmenty obrazów (w MB); Czasami może nieco przekroczyć wartość Mem Max o rozmiar pamięci, potrzebnej do przechowania zawartości pliku *.svg. Used służy do wyrobienia sobie poglądu, jak powinno być ustawione Mem Max aby nie zajmować za dużo RAM, a jednocześnie nie spowalniać niepotrzebnie procesu renderingu;
- **Enable**: Uaktywnia teksturę (sprawia, że w Blenderze pojawia się jej obraz). Możesz także użyć tego przełącznika do "zresetowania" pamięci, zajętej przez fragmenty renderowanych obrazów;
- File name: Ścieżka do pliku *.svg lub (lub *.xml por. str. 1025); To pole nie jest specjalnie długie ma 63 znaki. Najlepiej wpisuj tu ścieżki względne (względem aktualnego pliku Blendera). W ten sposób, gdy kiedyś przeniesiesz folder z całą strukturą projektu w nowe miejsce, nie zaszkodzi to działaniu wtyczki.

Po tym wyczerpującym opisie pozostaje dorzucić kilka rad praktycznych, na podstawie wskazówek z tej i poprzedniej strony:

- Zaraz po załadowaniu wtyczki możesz podnieść limit Mem Max i wykonać pierwszy rendering, by potem odczytać z pola Used, ile svgtex zużył pamięci. Potem ustaw Mem Max na 110% wartości Used;
- Dla tekstur oznaczeń i innych dokładnych rysunków, stosuj *Interpol* = 0 lub 1 i *Fid* = 0.0;
- Dla tekstur nierówności (*bump map*), stosuj *Interpol* = 0 lub 1 i *Fid* z zakresu -1.0 .. 0.0 (np. -0.5). Mniejsza wartość *Fid* zaokrągli nieco ostre krawędzie zagłębień;

Wtyczkę **svgtex** pisałem przede wszystko z myślą o użytku własnym. Jednak zdawałem sobie sprawę, że kiedyś udostępnię ją innym użytkownikom. Poświęciłem więc trochę czasu na przygotowanie obsługi większości elementów graficznych, które mogą wystąpić w plikach SVG. W efekcie komponent potrafi narysować prawie wszystko, co przewiduje ten standard. Poniżej wyliczam to, co nie zostało zaimplementowane:

- znaczniki (markers) bo to sporo pracy, a nie miałem pomysłu, do czego mógłbym je wykorzystywać;
- filtry (*filters*) obecnie obsługiwany jest tylko jeden typ: *Gaussian blur*. W kodzie przygotowałem miejsce na pozostałe, ale nie były mi potrzebne;
- napisy (*text*) brak obsługi wewnętrznych czcionek SVG. Svgtex wykorzystuje obecnie wyłącznie czcionki True Type, więc w środowisku Windows wszelkie teksty napisane przy ich użyciu są wyświetlane poprawnie. W kodzie programu jest przygotowane miejsce dla innych typów czcionek np. FreeFont. Nie potrzebowałem ich jednak, gdyż tworzę swoje modele w środowisku Windows;
- obrazy rastrowe wszelkie referencje do obrazów rastrowych są ignorowane. Svgtex ma służyć do rysowania wektorowego;

Oczywiście, odwieczna mądrość programistów mówi "w każdym programie zawsze jest jeszcze co najmniej jeden nie wykryty błąd". Komponent na razie był porównywany z rezultatami, które zwracał Inkscape, oraz przeszedł test praktyczny na rysunkach, które przygotowywałem dla moich modeli. Wydaje mi się, że udało mi się wyeliminować przynajmniej większość jego błędów. W razie czego, kod źródłowy w całości jest dostępny na <u>www.samoloty3d.pl</u> (dokładny adres: *http://www.samoloty3d.pl/downloads-svgtex_p.xml*).

17.21 Program svgview — podgląd obrazów SVG

Program **svgview.exe** służy do podglądu plików *.*svg*. Jest "ubocznym produktem" prac nad wtyczką **svgtex** (por. str. 613, 1020). Znajdziesz go wśród materiałów towarzyszących tej książce (w pliku *source.zip*), w tym samym folderze, co plik *svgtex.dll: source\textures\plugin\svgview.exe*. To w istocie testowy program z biblioteki **AGG**, na którym weryfikowałem działanie parsera SVG.

Możesz używać *svgview.exe* do sprawdzenia, jak **svgtex** wyświetli konkretny plik *.*svg*. Wystarczy wywołać ten program w linii poleceń, podając jako argument odpowiednią ścieżkę (Rysunek 17.21.1):



Rysunek 17.21.1 Pomocnicza przeglądarka plików SVG - svgview.exe.

Powiększenie obrazu możesz w *svgview.exe* zmieniać za pomocą suwaka *Scale*, umieszczonego u góry ekranu. Do przesuwania obrazu służy myszka — wystarczy ją przesunąć z wciśniętym **LPM**. Przy każdej zmianie skali lub przesunięciu obraz jest rysowany na nowo. Liczby u góry ekranu (Rysunek 17.21.1) podają, ile wierzchołków napotkano, i ile trwało przerysowanie obrazu.

Okno, które pokazuje Rysunek 17.21.1, to podstawowy obraz nierówności powierzchni samolotu. Jednocześnie to mój "stress test": plik *skin.svg*. Znajdziesz go wśród danych towarzyszących tej książce, w pliku *textures.zip*, w folderze *p40/textures/6.10*.

Zwróć uwagę, jak ogromną liczbę wierzchołków pokazuje Rysunek 17.21.1 — niemal 2 miliony. Prawie wszystkie z nich (95%) należą do nitów (po 32 na pojedynczy nit). To trochę tłumaczy problemy, jakie miałem pod koniec tworzenia tego obrazu w edytorze.

Gdy *skin.svg* został ukończony, czas jego przerysowywani w Inkscape osiągnął około 45 sekund, a "świeżo załadowane" dane zajmowały prawie 350 MB RAM. (Potem, w trakcie pracy, ta ilość RAM jeszcze się zwiększała). Już lepiej to wygląda w Firefox — 12 sekund na przerysowanie *skin.svg*, przy zużyciu 27 MB RAM. A *svgview.exe* potrzebuje na to zaledwie 1.4 sekundy i 2 MB RAM! Przy czym zapewniam, że w kodzie można dokonać jeszcze wielu prostych optymalizacji, a sam program jest był skompilowany w pochodzącym z 1998r MS VC 6.0. Co prawda Inkscape jest edytorem, więc musi przechowywać i przetwarzać dane SVG w trochę inny sposób. Powiedzmy, że może to spowolnić przetwarzanie dwukrotnie. Ale taka różnica!? Ta rozrzutność w wykorzystaniu zasobów komputera przez Inkscape może być przejawem jakiegoś grubego błędu w architektu-rze tego programu.

17.22 Wtyczka svgtex — skróty XML

Svgtex to napisana przeze mnie wtyczka Belndera, podstawiającą jako teksturę zawartość pliku *.*svg*. W innych miejscach tej książki opisałem jej instalację (str. 1017), zasadę działania i obsługę (str. 1020), użycie (str. 613). W tej sekcji chciałbym wspomnieć o pewnych dodatkowych funkcjach **svgtex**, pozwalających osiągnąć ciekawe efekty, np. dynamiczne kompozycje z kilku obrazów SVG. Możesz z nich skorzystać, tworząc proste pliki *.*xml*. Jeżeli nie wiesz, co to jest **XML**, spróbuj gdzieś znaleźć wprowadzenie na ten temat. Podkreślam, że przedstawione tu rozwiązanie jest zupełnie opcjonalne — jak najbardziej możesz używać **svgtex** bez informacji, zawartych poniżej.

 Do zrozumienia reszty tej sekcji przydatna jest podstawowa znajomość składni XML. Informacje na ten temat znajdziesz choćby w Wikipedii: po polsku (*http://pl.wikipedia.org/wiki/XML*), lub trochę więcej po angielsku (*http://en.wikipedia.org/wiki/XML*).

Nawet ubogi opis XML w wersji polskiej wystarczy aż nadto. Najważniejsze, abyś w tych artykułach przyjrzał się przykładom. Zupełnie nie przejmuj się takimi rzeczami jak DTD, którego opis umieszczono w wersji polskiej — to tutaj bez znaczenia.

Zacznijmy od prostego przykładu: czy pamiętasz jak narzekałem, że w panelu *Plugin* ścieżka do pliku *.*svg* nie może przekroczyć 63 znaków? Aby użyć dłuższych ścieżek, wystarczy stworzyć gdzieś w pobliżu aktualnego pliku Blendera plik *.*xml*, pełniący rolę "skrótu". Jego zawartość wygląda tak, jak to pokazuje Rysunek 17.22.1:



Rysunek 17.22.1 Zawartość pliku nor_details.xml — skrótu do obrazu skin.svg

Wszystkie elementy tego XML-a koniecznie poprzedzaj przedrostkiem (tzw. przestrzenią nazw) "*tex.*", bo inaczej plik nie zostanie poprawnie zinterpretowany. Element główny — *tex:texture* — jest zawsze taki sam. W środku jest tylko prosty odnośnik *tex:src*. W wartość jego atrybutu — *path* — wpisz pełną ścieżkę do pliku obrazu. W tym miejscu może mieć do 1024 znaków, więc na pewno każda się zmieści!

Nadaj temu plikowi nazwę *nor_details.xml*. Gdy wpiszesz tę nazwę w pole *File name* wtyczki, w oknie podglądu powinna się pojawić miniatura pliku *skin.svg* (Rysunek 17.22.2). To dowód, że skrót działa poprawnie:



Rysunek 17.22.2 Przykład użycia pliku nor_details.xml

Skróty xml pozwalają wykorzystać pliki *.*svg* jako "biblioteki obrazów". Powiedzmy, że ten sam *skin.svg*, używany w teksturze nierówności (**B.Skin.Nor-Details**), chcemy wykorzystać w innej teksturze: otworów (**B.Skin.Holes** — por. str. 405).



Rysunek 17.22.3 pokazuje zawartość skrótu holes.xml, który to umożliwia:

Rysunek 17.22.3 Zawartość pliku holes.xml – kolejny skrótu do obrazu skin.svg

W stosunku do poprzedniego przykładu (por. Rysunek 17.22.2), element *tex:src* został tu rozbudowany o kilka elementów *tex:override*. Zadaniem *tex:src* jest wskazanie pliku *.*svg*, który ma być wyświetlony (służy do tego atrybut *path*). Do wnętrza *tex:src* można wstawić elementy *tex:override*. Każdy z nich służy do zmiany właści-wości jakiegoś obiektu w pliku SVG, wskazanym przez *tex:src*. O tym, o który konkretnie obiekt chodzi, informuje atrybut *for*. Jego wartością jest identyfikator "celu zmian". W przykładzie pokazywanym przez Rysunek 17.22.3, pierwsze dwa elementy *tex:override* wyłączają widoczność (*display="none"*) warstw **Bkg-Grey** i **Ref:Nor-Details**. Musimy to zrobić, bo te warstwy są w oryginalnym *skin.svg* włączone (używamy ich w teksturze **B.Skin.Nor-Details**). Kolejne dwa elementy *tex:override* służą do włączenia widoczności (*display="inline"*) warstw potrzebnych do tekstury otworów: **Bkg-White** i **Holes**. Propagowanie tych zmian na elementy potomne warstw jest wyłączne (*propagate="none"*) - wystarczy zmienić widoczność samej warstwy, nie ma potrzeby zmieniać tego ustawienia dla obiektów, które zawiera.

Wszystko pięknie, ale zapewne zapytasz, Uważny Czytelniku, skąd wiem, że identyfikator warstwy **Bkg-Grey** to "layer5", a **Result:Nor-Details** to "layer19"? Otóż to można wyszukać w treści pliku SVG (Rysunek 17.22.4) :



Rysunek 17.22.4 Wyszukiwanie identyfikatora warstwy w dokumencie *.svg

Otwórz plik *.*svg* w przeglądarce internetowej (np. Firefox) i wybierz polecenie *Pokaż źródło strony*. W oknie, które się pojawi, naciśnij **Ctrl-F**, i wyszukaj warstwę po nazwie (Rysunek 17.22.4b). Warstwy Inkscape to w istocie grupy (<g>) elementów. Atrybut *id* znalezionej grupy to identyfikator warstwy.



Rysunek 17.22.5 przedstawia rezultat zastosowania skrótu holes.xml.

Rysunek 17.22.5 Działanie pliku holes.xml

Na podglądzie tekstury (w panelu *Preview*) widać złożenie warstw **Holes** i **Bkg-White**, mimo tego, że w źródłowym pliku *skin.svg* są nadal wyłączone.

Poniżej podaję pełen opis formatu skrótów xml używanych przez wtyczkę svgtex:

Elementem głównym (*root element*) pliku musi być *tex:texture* (por. str. 1025, Rysunek 17.22.1). Poza definicją przedrostka (inaczej mówiąc: przestrzeni nazw — *namespace*) *tex:* nie posiada żadnych atrybutów. Może zawierać jeden lub więcej elementów *tex:src*¹. Oprócz nich może zawierać także, o ile jest to potrzebne, inne elementy przewidziane przez format SVG. Ich nazwy nie mogą mieć przedrostka *tex:*.

Element *tex:src* wskazuje na plik SVG, który należy narysować. Pełna ścieżkę do tego pliku jest wartością atrybutu *path* (max długość — 1024 znaków). Element *tex:src* może (ale nie musi!) zawierać elementy *tex:override*.

Element *tex:override* zmienia właściwości (atrybuty) wybranego elementu pliku SVG. (Tego pliku, który wskazuje jego "rodzic": element *tex:src*). Identyfikator (*id*) zmienianego elementu jest podawany w obowiązkowej wartości *for*. Oprócz tego *tex:override* powinien podawać jakieś atrybuty SVG, które ulegają zmianie (np. *display*). Wreszcie wartość *propagate* określa zakres ("głębokość") wprowadzanych zmian. Jeżeli *propagate="none"*, zmiana atrybutów dotyczy wyłącznie "celu" (obiektu wskazywanego przez *for*). Jeżeli *propagate="all"*, zmiana jest dodatkowo przenoszona na wszystkie elementy zawarte wewnątrz "celu".

W elemencie *tex:override* mogą wystąpić w zasadzie wszystkie elementy stylu rysowania, które przewiduje specyfikacja SVG (dostępna pod adresem <u>http://www.w3.org/TR/SVG11</u>). Poniżej wyliczam te atrybuty, których zmiana może Ci się najczęściej przydać:

- display: widoczność elementu (display="inline" jest widoczny, a display="none" niewidoczny);
- opacity: ogólna nieprzejrzystość elementu liczba od 0.0 do 1.0. Np. opacity="0.5" oznacza 50% przejrzystości;
- fill: barwa wypełnienia (np. fill="#080808" wypełnia szarością o 50% czerni); Można tu także użyć odsyłacza do gradientu lub wzoru (pattern) patrz specyfikacja SVG;
- *fill-opacity*: nieprzejrzystość wypełnienia liczba od 0.0 do 1.0, np. *fill-opacity="0.6"* oznacza wypełnienie w 40% przejrzyste;
- stroke: barwa konturu (np. stroke="#080000" oznacza kontur ciemnoczerwony);
- stroke-opacity: nieprzejrzystość konturu liczba od 0.0 do 1.0;
- stroke-width: szerokość konturu jakaś liczba dodatnia;
- stroke-dasharray: referencja do wzoru kreskowania linii (szczegóły patrz specyfikacja SVG);

¹ Na przykład mogą to być dwa lub elementy *tex:src*, z których każdy odnoszący się do innego pliku SVG. **Svgtex** narysuje wówczas złożenie tych dwóch obrazów. Istnieje jedno ograniczenie: każdy obiekt z użytych plików musi mieć unikalną wartość swojego atrybutu *id*. Aby pliki stworzone za pomocą Inkscape spełniły ten warunek, w jednym z nich trzeba zmienić *id* warstw (Inkscape zawsze przypisuje im "layer1", "layer2", itd.). Można to zrobić w samym Inkscape (w oknie otwieranym poleceniem *Edit* →*XML Editor*).

- font: specyfikacja czcionki (używana w elementach tekstowych); Szczegóły patrz specyfikacja SVG;
- font-size: rozmiar czcionki (używany w elementach tekstowych); Szczegóły patrz specyfikacja SVG;
- filter: referencja do definicji filtru; Szczegóły patrz specyfikacja SVG;
- mask: referencja do definicji maski; Szczegóły patrz specyfikacja SVG;
- clip-path:referencja do obszaru, "obcinającego" rysunek; Szczegóły patrz specyfikacja SVG;

Rysunek 17.22.6 przedstawia przykład zaawansowanego skrótu dla **svgtex**, w którym wykorzystałem wiele spośród wyliczonych powyżej reguł¹:



Rysunek 17.22.6 Przykład bardziej złożonego pliku skrótu

W przykładzie stworzyłem kompozycję, nakładając na rysunek *t.svg* obraz *w.svg*. (To przykład wielokrotnego zastosowania elementu *tex:src*). Dodatkowo zdefiniowane jest tu (wewnątrz grupy *defs*) efekt gaussowskiego rozmycia. (Jest to element SVG typu *filter*, o identyfikatorze "Blur3"). Ten efekt zastosowałem do "rozmycia" obrazu *t.svg*, nadpisując domyślny filtr (wyrażeniem *filter="#Blur3"*). Zwróć uwagę, że komponent *tex:override*, w którym umieściłem ten atrybut *filter*, wskazuje w *for* jako "cel" własnego "rodzica" (element *tex:src* o identyfikatorze t). Takie "sztuczki" także są dopuszczalne!

Uff, jeżeli, drogi Czytelniku, dotarłeś aż do tego miejsca, to znaczy że jesteś programistą lub przynajmniej masz zadatki na programistę! Nie chciałbyś może napisać w Pythonie jakiegoś przyjemnego, okienkowego "klikadełka", które ułatwiłoby szerokim masom tworzenie własnych skrótów xml? Biblioteki do przetwarzania xml masz w standardzie Pythona, a jakieś środowisko "okienkowe" także się znajdzie. Ja sam nigdy nie mogłem na to znaleźć czasu, bo, jak już o tym wspominałem, tworzyłem ten komponent głównie na własne potrzeby...

¹ Nazwy plików: *t.svg* i *w.svg* są przykładowe. Nie ma ich nigdzie w tej książce

17.23 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 (Rysunek 17.23.1):





Ruch na scenie Blendera jest podzielony na pojedyncze klatki (*frames*) . W nagłówku *Buttons Window*, i innych (np. *Timeline*) znajduje się kontrolka, w której określasz aktualną klatkę (Rysunek 17.23.1). Animowany obiekt może zajmować w kolejnych klatkach różne położenia. Linia, łącząca współrzędne takich punktów, nosi nazwę *IPO Curve*. W układzie **Animation** po prawej znajduje się okno *IPO Curve Editor*, służące do pracy z takimi liniami. Linię ruchu — *IPO Curve* — najprościej jest stworzyć pokazując programowi kolejne położenia obiektu w kilku ważniejszych klatkach. Takie klatki nazywamy klatkami kluczowymi (*keyframes*).

Wskaż 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 17.23.2):



Rysunek 17.23.2 Wstawienie pierwszej "pozycji kluczowej" (keyframe)

Z menu, które się pojawi, wybierz rodzaj współrzędnej, która ma się zmieniać (w tym przypadku to Rotation).



Następnie przejdź do kolejnej klatki kluczowej — powiedzmy nr 10 (Rysunek 17.23.3):

Rysunek 17.23.3 Przygotowanie drugiej pozycji kluczowej - dziesięć klatek dalej

(Zwróć uwagę, że spowodowało to przesunięcie w oknie *IPO* zielonej, pionowej linii — to nasz wskaźnik aktualnej klatki). Obróć w tej klatce kołpak śmigła wokół lokalnej osi Z o jakiś solidny kąt — np. 90°. Następnie znów naciśnij w oknie *3D View* klawisz 1. Pojawi się menu *Insert Key*, z którego powtórnie wybierz *Rotation*² (Rysunek 17.23.4a):



Rysunek 17.23.4 Wstawienie kolejnej pozycji kluczowej (keyframe) — w klatce 10.

Tym razem spowodowało to dodanie do linii IPO drugiego punktu (Rysunek 17.23.4b). Linia z prostej stała się łamaną — dla parametru *RotY*. Wartość tego obrotu dla klatki 10 wynosi 90° (tyle, o ile obróciłeś kołpak przed naciśnięciem). Wygląda na to, że Blender zapamiętuje współrzędne obrotów w globalnym układzie współrzędnych. Lokalna oś **Z** kołpaka śmigła biegnie wzdłuż globalnej osi **Y**, stąd taki efekt.

¹ To skrót do polecenia *Object-Insert Keyframe*. Wybieranie jednak co chwilę tej komendy z menu byłoby wysoce niepraktyczne!

² Zwróć uwagę, że na początku menu *Insert Key* znajdują się podstawowe rodzaje współrzędnych: *Loc*(ation) — położenie, *Rot*(ation) — 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 współrzędny "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 rysunku linie IPO związane z obrotami wokół globalnych osi X i Y — bo te parametry nie mają ulegać żadnym zmianom.

W edytorze linii IPO obowiązują podobne reguły do innych okien Blendera. Poszczególne linie IPO zaznaczasz klikając w nie **PPM**. Na liście po prawej spowoduje to podświetlenie parametru, którego zmianę opisuje krzywa (Rysunek 17.23.5a):



Rysunek 17.23.5 Usuwanie niepotrzebnych współrzędnych obrotu

Możesz także przejść do edycji punktów zaznaczonej krzywej (klawiszem **Tab**) — to odpowiednik trybu edycji siatki. Podobnie klawisz **X** (*Curve →Delete*) powoduje usunięcie tego, co jest zaznaczone. Wskaż więc linię związaną z parametrem *RotX* i naciśnij **X**, by ją usunąć. Zrób to samo dla linii *RotZ*. Na ekranie IPO powinna pozostać tylko linia łamana, związana z "naszym" obrotem: *RotY* (Rysunek 17.23.5b).

Dla takiego "łamanego" kształtu IPO kołpak obracałby się jednostajnie od klatki 1 do 10, po czym cały ruch by ustał. Aby wymusić jednostajny obrót przez cały czas animacji, zaznacz tę linię, po czym przełącz jej *Extend Mode* na *Extrapolation* (Rysunek 17.23.6):



Rysunek 17.23.6 Końcowe dostosowania linii animacji

Teraz linia *RotY* stała się pochyloną linią prostą — i o to chodziło. Na koniec możesz ustalić nazwę zestawu linii IPO, związanych z obrotem śmigłą (Rysunek 17.23.6b). Nadałem im tu nazwę **Propeller**.

 Zestaw krzywych IPO to dla Blendera "blok danych", przypisany do obiektu tak samo, jak np. siatka. Oznacza to, że możesz użyć wielokrotnie tego samego zestawu IPO do różnych obiektów. Czasami ta możliwość się przydaje, więc warto tworzyć krzywe IPO w sposób przemyślany, i nadawać im nazwy, ułatwiające identyfikację

Na koniec sprawdzamy "jak to się kręci". Naciśnij Att-A. Możesz także wybrać jedno z dwóch poleceń menu: *View→Play Animation* w oknie edytora IPO (Rysunek 17.23.7a), lub *View→Play Back Animation* w oknie *3D View*. Nasze śmigło zacznie się obracać (Rysunek 17.23.7b):



Rysunek 17.23.7 Test animacji

•	Animację możesz	przerwać w każdym	n momencie,	naciskając	Esc
---	-----------------	-------------------	-------------	------------	-----

Oczywiście, jeżeli obiekt nie porusza się tak, jak powinien, trzeba trochę pozmieniać jego linie IPO. Potem znów sprawdzić, uruchamiając animację. I tak dopóty, dopóki nie uzyskasz zadowalającego efektu.

• Po naciśnięciu Att-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!

17.24 Kompozycja rezultatu (Composite Nodes)

W tej sekcji pokażę posługiwanie się edytorem węzłów (*Node Editor*). Węzły to przygotowane przez twórców Blendera symbole bloków danych lub operacji. Możesz je ze sobą łączyć, tworząc w ten sposób różne ciekawe efekty, nieosiągalne za pomocą innych metod. W wersji 2.49 Blendera *Node Editor* umożliwia komponowanie tekstur, materiałów, oraz ostatecznych obrazów sceny (renderów). W tej sekcji zademonstruję jego zastosowanie właśnie w zakresie kompozycji ostatecznego renderu. Węzły, które to umożliwiają, nazywa się w Blenderze "węzłami kompozycji" (*Composite Nodes*). Jako przykładu użyjemy tu modelu omawianego na str. 509 i dalszych (*p40\history\p-40.8.01a.blend*).



Przełącz jedno z okien Blendera na Node Editor (Rysunek 17.24.1):

Rysunek 17.24.1 Okno edytora węzłów - na przykładzie węzłów kompozycji (renderu)

Aby pojawiły się w nim dwa domyślne węzły, opisujące "zwykły" rendering, musisz przełączyć się w tryb *Composite Nodes* i włączyć *Use Nodes* (Rysunek 17.24.1). Po lewej widzisz węzeł typu *Render Layer*. To dane wejściowe. Po prawej znalazł się węzeł typu *Composite*. To okno renderu. Te węzły przekazują sobie (linia łą-cząca) piksele obrazu (*Image*). Typowe elementy węzła wejściowego pokazuje Rysunek 17.24.2a):



Rysunek 17.24.2 Obsługa węzła — na przykładzie *Render Layers*.

Połączenia pomiędzy węzłami można tworzyć, przeciągając myszką koniec danej linii do "wtyczki" innego węzła. Jeżeli złapiesz za taki koniec i upuścisz gdzieś poza węzłem (Rysunek 17.24.2b) — usuniesz to połączenie. Dane wejściowe generuje na naszym schemacie węzeł *Render Layer*. Co to właściwie jest, ta "warstwa renderowania" ? To zespół parametrów, opisujących szczegółowo, co ma robić (i czego nie robić) wewnętrzny renderer Blendera. Domyślnie w każdej scenie istnieje jedna, domyślna warstwa renderowania, o nazwie **1 Render Layer**. Jej definicję możesz obejrzeć w panelu *Render Layers* (Rysunek 17.24.3):



Rysunek 17.24.3 Ustawienia sceny, związane z Composite Nodes.

Kontrolka umieszczona u góry panelu *Render Layers* jest nieco myląca. To nie jest część definicji aktualnej warstwy renderowania, tylko kopia przełączników z nagłówka okna *3D View*. Można tu zobaczyć (i przełączyć) widoczność sceny warstw (tych z obiektami). Poniżej znajduje się lista, z której możesz wybrać istniejącą warstwę renderowania, lub zdefiniować nową. (To typowa lista, jaką spotkałeś przy okazji pracy z innymi rodzajami danych. Gdy ją rozwiniesz, zobaczysz spis istniejących elementów oraz polecenie *ADD NEW*, pozwalające dodać nowy). Poniżej listy znajdują się kontrolki, pozwalające zmienić parametry wybranej warstwy renderowania. To w istocie parametry pracy wewnętrznego renderera Blendera. (Każdy węzeł *Render Layer* na schemacie oznacza odrębny "przebieg" renderowania). Można tu wyróżnić trzy zespoły kontrolek (numeracja wg ilustracji):

- Zestaw przełączników, pozwalający określić, które warstwy (obiektów) sceny renderer ma przetwarzać, a które nie. (Traktuj to jako coś w rodzaju "maski", nakładanej na aktualną widoczność warstw sceny. Na renderze zobaczysz wyłącznie warstwy (obiektów), które w chwili przetwarzania były włączone w oknie 3D View, i które są zaznaczone na tej masce warstwy renderowania);
- Zestaw przełączników określających, co ma być w tym przebiegu renderowane. (Można tu np. wyłączyć generowanie obrazu tła — wyłączając opcję Sky);
- 3. Zestaw przełączników, pozwalających włączyć lub wykluczyć jakiś rodzaj informacji, udostępnianej przez węzeł *Render Layer*. Przełącznik *Combined* oznacza połączenie wszystkich opcji z dolnego wiersza (*Col, Diff, Sphe*, ..). To składniki danych, widocznych na schemacie jako *Image*. Te dolne przełączniki (z "szarym kółkiem") mają trzy stany. Oprócz zwykłych "włączone" i "wyłączone" trzeci stan oznacza "wykluczone z *Combined*". Twórcy Blendera chcieli w ten sposób ułatwić wykluczanie pojedynczych składników informacji z danych przekazywanych do innych węzłów;

W tym przykładzie nasz węzeł *Render Layer* musi udostępnić dodatkową informację o przemieszczeniu pikseli (w efekcie animacji) — *Speed*. Uzyskasz to, włączając w panelu przełącznik *Vect* (Rysunek 17.24.4):



Rysunek 17.24.4 Ustawienia sceny, związane z Composite Nodes.

Po tym przygotowaniu możemy wzbogacić nasz schemat o efekt rozmycia ruchu. Wywołaj polecenie *Add→Filter→Vector Blur*. Spowoduje to pojawienie się nowego węzła (Rysunek 17.24.5):



Rysunek 17.24.5 Dodanie węzła Vector Blur.

Teraz "poprzepinaj kabelki" — linie łączące węzły — tak, by do Vector Blur dochodziły dane ze sceny (*Render Layer*), a wychodził — obraz (*Image*) do węzła rezultatu (*Composite*) (Rysunek 17.24.6):



Rysunek 17.24.6 Podłączenie węzła Vector Blur dla uzyskania efektu rozmycia ruchu wirujących łopat śmigła.

Jeżeli miałeś już wcześniej wykonany render warstwy **Render Layer** (tzn. w okienku węzła widać obrazek) to podłączenie węzła *Vector Blur* od razu stworzy odpowiedni efekt. Nie musisz naciskać klawisza *Render*. Możesz nawet otworzyć okno *Render Window*, i na bieżąco śledzić wpływ zmiany parametrów węzła *Vector Blur* na rozmycie śmigła modelu. Jest to możliwe dlatego, że Blender przez cały czas przechowuje w węźle *Render Layer* rezultat ostatniego przebiegu. To spora oszczędność czasu, prawda?

W modelu, na którym pracujemy w tym przykładzie, jedynym ruchomym (tj. animowanym) elementem jest śmigło (kołpak i łopaty). Dodaliśmy do sceny efekt *Vector Blur* aby uzyskać jak najbardziej realistyczne rozmycie ruchu tego zespołu. Dlatego ustawiłem w parametrach węzła wartości maksymalne: *Samples* na 256, a *Blur Fac* na 2.0 (Rysunek 17.24.6). Być może powinienem nadać jeszcze śmigłu większą prędkość obrotową, bo rezultat jest daleki od zadowalającego (Rysunek 17.24.7):



Rysunek 17.24.7 Ocena rezultatów działania Vector Blur

Śmigło, mimo maksymalnych wartości *Samples* i *Blur Fac*, nie wygląda na specjalnie rozmyte. W dodatku za jedną z łopat widać na kołpaku wyraźny cień, jak gdyby była nieruchoma. To, niestety, nie jest jakiś drobny błąd w programie. Twórcy tego węzła zastrzegli w instrukcji, że algorytm *Vector Blur* jest szybki, ale nie rozmywa żadnych cieni! Tym niemniej przydaje się do dodatkowego "rozmycia" efektów *MBLUR* (por. str. 518).

Innym przydatnym elementem jest węzeł rezultatu typu *File Output*. Służy do zapisania obrazu/animacji do pliku/plików. Możesz go podłączyć do dowolnego wyjścia typu *Image* (Rysunek 17.24.8):



Rysunek 17.24.8 Inny, przydatny węzeł: File Output.

W parametrach węzła określasz folder, w którym mają być umieszczone pliki z obrazami, oraz format tworzonych plików rastrowych. (Parametr *Quality* ma znaczenie jedynie wtedy, gdy wybierzesz format JPG). *File Output* nadaje plikom nazwy numery kolejnych klatek — stąd lepiej ustaw odpowiednio parametry *SFra* i *EFra* (*S*tart *Fra*me i *E*nd *Fra*me). W naszym przykładzie ustawiłem obydwie na klatkę 10, więc węzeł umieścił we wskazanym folderze jeden plik, o nazwie *0010.png*. Zwróć uwagę, że *File Output* generuje plik gdy tylko podłączysz go do "wypełnionego" obrazkiem węzła *Render Layer* (także poprzez węzły pośrednie). Lepiej więc ustaw najpierw wszystkie parametry, a dopiero na koniec "podłączaj kable".
17.25 Linie pomocnicze (Grease Pencil)

Czasami pojawia się potrzeba szybkiego naszkicowania na ekranie jakiejś linii pomocniczej. W Blenderze służy do tego wprowadzona, bodajże w wersji 2.48, 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. W tym przykładzie *Grease Pencil* wykorzystamy do przeniesienia zarysu kolein z foto-grafii tła na podstawioną pod nim płaszczyznę "gruntu" (Rysunek 17.25.1):



Rysunek 17.25.1 Wykorzystanie linii pomocniczych do stworzenia tekstury kolein

W tym przypadku nie zależy nam specjalnie nią dużej dokładności linii — uzyskany obraz kolein wykorzystamy jako teksturę w modyfikatorze *Displace* (por. str. 530, 925). Kształt kolein i tak ma być rozmyty. Do takich właśnie zdań *Grease Pencil* nadaje się najlepiej.

Okno sterujące funkcją wywołujesz z menu *View→Grease Pencil* (Rysunek 17.25.2):



Rysunek 17.25.2 Wywołanie Grease Pencil

Na razie wszystko jest wyłączone. Naciśnij przycisk Use Grease Pencil, by włączyć tę funkcję.

Tak jak w Inkscape czy GIMP, *Grease Pencil* rysuje swoje linie na warstwach. Przy pierwszym uruchomieniu jeszcze nie ma żadnej, więc musisz teraz nacisnąć przycisk *Add New Layer* (Rysunek 17.25.3a):

a) / Grease Pencil	b) 7 Grease Pencil	c) Grease Pencil
Use Grease Pencil Draw Mode Add New Layer Sketch in 3D	Use Grease Pencil Draw Mode Add New Layer Sketch in 3D	Use Grease Pencil Draw Mode Add New Layer Sketch in 3D
Naciśnij ten przycisk, by stworzyć nową warstwę!	Dpacity: 0 Conion - Skin Conion - Skin GS Podstawowe para Kreski: barwa, ni Conion - Skin Conion - Skin	Te przełączniki uak- tywniają tryb rysowania Onion – Skin GStep:0 ametry ickness:
		Del Last Stroke

Rysunek 17.25.3 Grease Pencil - przygotowanie do pierwszego rysowania

Panel nowej warstwy pojawi się w tym samym oknie, poniżej przycisków (Rysunek 17.25.3b). Jak się już chyba domyślasz, możesz stworzyć wiele różnych warstw *Grease Pencil*. Z każdą z nich jest związany jeden rodzaj linii. W panelu warstwy określasz barwę kreski, jej nieprzejrzystość (*Opacity*), oraz szerokość (*Thickness*) (Rysunek 17.25.3b). 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.

Aby rozpocząć rysowanie, włącz przełączniki *Draw Mode* i *Sketch in 3D* (Rysunek 17.25.3c). (*Sketch in 3D* powoduje, że narysowane linie będą zmniejszane i powiększane zgodnie z aktualnym widokiem sceny).

Przed rozpoczęciem rysowania ustal parametry kreski. W naszym przypadku zależy nam na kolorze, który wyraźnie będzie się odcinał od czerni i bieli tekstury nierówności. Wybierz więc czerwony (Rysunek 17.25.4a):

а		
E	🔻 Grease Pencil	······
	Use Grease Pencil Add New Layer	Draw Mode Sketch in 3D
1	a Info:GP_Layer	Ustaw barwę, przej- rzystość i grubość Oni aktualnej linii
	Opacity: 0	GStep:0
	Thickness	Convert to Del Last Stroke
1		
7		Rysuj, przesu- wając myszkę z wciśniętym LPM
1,	🗳 (1)	



Rysunek 17.25.4 Grease Pencil - rysowanie

Rysujesz, przesuwając myszkę z wciśniętym **PM**. *Grease Pencil* traktuje to jako szkicowanie, i odpowiednio "często" zapamiętuje kolejne pozycje kursora jako nowe wierzchołki linii. Do usuwania narysowanych kresek służy drugi przycisk myszki: **PPM**. Gdy go wciśniesz, wokół kursora pojawia się kółko (Rysunek 17.25.4b). To rozmiar naszej "gumki". Przesuwając myszkę z wciśniętym **PPM** wymazujesz narysowane wcześniej linie.

Jeżeli chciałbyś narysować odcinek linii prostej, rysuj z wciśniętym klawiszem **Ctrl**. Działanie *Grease Pencil* jest w tym przypadku trochę nieintuicyjne, ale można się przyzwyczaić. Z wciśniętym **Ctrl**-**LPM** dalej powstaje "szkicowana" linia. Jednak gdy zwolnisz przycisk myszki (**LPM**), usunięte zostaną z niej wszystkie środkowe wierzchołki. Powstanie w ten sposób segment linii prostej, łączący punkty wciśnięcia **LPM** i zwolnienia **LPM**.

• Warstwy (a więc i linie) *Grease Pencil* są związane z konkretnym oknem widoku. Gdy narysowałeś je w jednym oknie *3D View*, nie zobaczysz ich w drugim!

Gdy skończyłeś rysowanie, wyłącz przełącznik *Grease Pencil:Draw Mode* (Rysunek 17.25.5). Narysowane linie pozostaną na ekranie, ale zwolnione zostaną przypisania funkcji do klawiszy myszki. Pozwala to na pracę w innych trybach Blendera.

Na zakończenie pokażę, jak wykorzystuję naniesione na okno linie *Grese Pencil*. Podstawiam pod nie (włączając widoczność odpowiedniej warstwy) kwadratową powierzchnię, pełniącą rolę "gruntu" sceny (Rysunek 17.25.5). Do siatki gruntu przypisuję w *UV/Image Editor* nowy, biały obraz (Rysunek 17.25.6):





Rysunek 17.25.5 Podstawienie płaszczyzny ziemi (Ground)

Rysunek 17.25.6 Pusty obraz, przypisany do płaszczyzny Ground

Na ten obraz nanoszę, w trybie Texture Paint i projekcji "z kamery", zarys kolein (Rysunek 17.25.7):



a potem rozmywam i wygładzam krawędzie Soften Smear Clone Draw BR:White XF Mix Airbrus 1.000 Wrap Opacity 1.0 P ize 200 Falloff 0.50 P Spacing 10.1. P Add New × 🔅 🚺 🗧 🤝 View Image 📌 🗢 IM:ground-nor

Rysunek 17.25.7 Malowanie kolein na powierzchni "gruntu"

Rysunek 17.25.8 Końcowe poprawki uzyskanego obrazu

Na koniec "rozmywam" krawędzie czarnych plam za pomocą półprzeźroczystych bieli i narzędzia *Soften*, dostępnych w przyborniku *UV/Image Editor* (Rysunek 17.25.8). W rezultacie uzyskuję obraz nierówności, odpowiedni do użycia w modyfikatorze *Displace*.

17.26 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. 533, Rysunek 8.2.27). Obraz trawy już na nim jest. 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 na podłoże (Rysunek 17.26.1):



Rysunek 17.26.1 Efekt dodania źdźbeł trawy wokół i wewnątrz cienia samolotu

Rysunek 17.26.1a) przedstawia cień samolotu, rzucamy na powierzchnię, imitującą grunt. (Materiał powierzchni jest półprzejrzysty i ma włączoną opcję *Only Shad*, dlatego "widać" poprzez nią zdjęcie tła — por. str. 529). Krawędź cienia jest tu idealnie gładka, a przecież biegnie po ziemi, porośniętej drobną trawą. Także wnętrze cienia jest tu jakieś takie …płaskie.

Rysunek 17.26.1b) 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, które mają być pokryte trawą. Dopasujemy ją do kolein, biegnących przez podłoże (Rysunek 17.26.2):





Włączyłem (**Shift**-**T**) wyświetlanie tekstury, by na powierzchni "gruntu" widać było obraz kolein. W widoku ustawionym równolegle do podłoża naniosłem kwadrat (*Plane*). Potem tę siatkę "pokrzywiłem" i zagęściłem (ułatwi to sterowanie lokalną gęstością trawy. Pociąłem (*Knife*) jej powierzchnię w ten sposób, by omijała koleiny, które nie są na zdjęciu zarośnięte. Rozkład siatki "trawnika" musi w miarę dokładnie odpowiadać układowi trawy, widocznemu na tle sceny (Rysunek 17.26.3a):





Rysunek 17.26.3 Siatka w kamerze — przymiarka do cienia samolotu

Potem sprawdź jeszcze w podglądzie renderu (**Shift**-**P**), jak siatka "trawy" obejmuje cień samolotu (Rysunek 17.26.3b). Nie zawsze trzeba pokrywać wszystko — pewne fragmenty cienia mogą pozostawać poza siatką. To zawsze zależy od rozłożenia roślinności na zdjęciu tła.

Gdy kształt "trawnika" masz już przygotowany, przejdź do zestawu Object: Particles (Rysunek 17.26.5):



Rysunek 17.26.4 Stworzenie nowego systemu cząstek (Particle System)

W panelu *Particle System* naciśnij przycisk *Add New*. Stworzy to nowy system cząsteczek (o domyślnej nazwie **PSys**). Zmień od razu nazwę na inną (np. **Grass.Short**) i typ systemu na *Hair* (Rysunek 17.26.5):



Rysunek 17.26.5 System cząsteczek Grass.Short — ustawienia

W pole *Amount* wpisz liczbę źdźbeł trawy. Zaznacz także opcję *Random*, by były losowo rozrzucone po ścianach siatki. W panelu *Phisics* ustal kształt trawy. W sekcji *Initial velocity* parametr *Normal* odpowiada za długość trawy, a *Random* — za losowe odchylenia tej długości. Dodatkowo ustaw *AccZ*, by źdźbła były pochylone, i *Brown* (chodzi o chaotyczne odchylenia — "ruchy Browna"), by trawa była odpowiednio "zmierzwiona". Dodany do siatki system cząsteczek widać także w panelu *Modifiers* (z zestawu *Editing*). Aby trawa nam lepiej rosła, dodaj przed modyfikator systemu cząsteczek podział *Subsurf* (Rysunek 17.26.6a):



Rysunek 17.26.6 Efekt na siatce

W trakcie tych ustawień zapewne zauważyłeś zmiany, widoczne w widoku *3D View*. Blender rysuje źdźbła trawy jako małe, wygięte kreski. Dla tak niewielkich wartości *Phisics:Normal*, których używamy, wyglądają jak kropki (Rysunek 17.26.6b). (To dlatego, że odtwarzamy krótką trawę — zresztą na lotniskach trawa chyba nigdy nie jest długa). Gdy jednak powiększysz jakiś fragment tego obrazu, zauważysz że są to kreski (Rysunek 17.26.6c). Ich kształt, kierunek itp. odpowiadają wartościom wpisanym w panel *Phisics*. Możesz tu na bieżąco obserwować, jaki wpływ mają poszczególne parametry (*Normal, Random, Brown*, ...).

 Im więcej źdźbeł trawy — tzn. im większa wartość *Particie System:Amount* (str. 1041, Rysunek 17.26.5) tym większy rozmiar pliku Blendera! Wygląda na to, że przechowywane są w nim wszystkie wierzchołki każdego źdźbła.

Siatce trawy należy przypisać jeszcze materiał. Nazwijmy go **Grass**. Rysunek 17.26.7 przedstawia pierwsze, wstępne ustawienia shaderów i barwy:

Links and Pipeline SSS	🔽 🔻 Shaders	Material Ramps
Link to Object	Oren-N ≠ Ref 0.90 Tangent	***
MA:Grass X G F Nodes	Rough:1.0 Shadow	VCol Ligh VCol Pain TexFace A Shadeles
	Blinn = Spec 0.00 TraShad	No Mist Env ObColor Shad A 1.00
	Hard:1 Only Sha	(na razie)
Render Pipeline	LINETT I JUDI	Col R
Halo ZTransp Zoffs: 0.00	GR: Exclusive Blas	Spe G 1.000
Full Osa Wire Strands Zinvert	Tralu 0.00 SBias 0.0 Zupełny brak	Mir B 1.000
Radio OnlyCast Traceable Shadbuf	Amb 0.30 Emit 0.00 Połysku	RGB HSV DYN A 1.000
	LBias 0.00	

Rysunek 17.26.7 Podstawowe ustawienia materiału trawy (Grass)

Materiał ma kolor biały, bo za barwę odpowiada tekstura z obrazem tła. Tekstura ta jest nałożona w specjalnym trybie mapowania *Win* (Rysunek 17.26.8). (W tym trybie jest "przyklejona do kamery", a nie do siatki):



Rysunek 17.26.8 Mapowanie obrazu tła jako tekstury materiału Grass

Trawa jest byt jasna ... a z przod - rzadka!

Zróbmy teraz pierwszy, próbny render. Na razie będziemy starali się jak najbardziej "wpasować" trawę w zdjęcie tła, więc wyłączyłem wszystkie warstwy z modelem samolotu (Rysunek 17.26.9):

Rysunek 17.26.9 Próbny render trawy

W zasadzie trawa wyszła zbyt jasna (choć zdarzają się w niej ciemne punkty). Widać także różnice w gęstości, wynikające ze zniekształcenia perspektywicznego. (Ta sama ilość źdźbeł widziana z większej odległości wydaje się gęstsza, niż tuż przy krawędzi ekranu). Poprawki zacznijmy od sterowania gęstością trawy. Można przypisać wartość tego parametru do wag, malowanych na powierzchni siatki (por. str. 919). Najpierw przypisz wszystkie wierzchołki siatki emitera do jednej grupy i nadaj jej nazwę **AII** (Rysunek 17.26.10):



Rysunek 17.26.10 Stworzenie w siatce grupy wszystkich wierzchołków (All)

(Więcej o grupach wierzchołków znajdziesz na str. 903). Potem przejdź do zestawu *Object:Particles*. W panelu *Extras* wybierz z listy parametr do zmiany (*Density*) oraz przygotowaną przed chwilą grupę wierzchołków (**AII**) (Rysunek 17.26.11):



Rysunek 17.26.11 Przypisanie wag grupy wierzchołków (All) do gęstości trawy

Teraz przełącz się w tryb *Weight Paint* (szczegóły — por. str. 919) i zamaluj wagą = 0 ściany położone najdalej od kamery (Rysunek 17.26.12a):



Rysunek 17.26.12 Malowanie gęstości trawy

W trakcie malowania zobaczysz, jak znikają z nich źdźbła (Rysunek 17.26.12b). To bardzo wygodna informacja zwrotna: możesz na bieżąco decydować, gdzie zwiększyć wagę (kolor czerwony), a gdzie zmniejszyć (kolor niebieski). W zasadzie powinieneś dążyć do tego, by jak najbardziej zagęścić trawę tam, gdzie pada cień samolotu. Podczas takiego malowania możesz włączyć widoczność podwozia modelu. Położenie kół pozwala się lepiej zorientować, gdzie (mniej więcej) pada cień modelu.

Rysunek 17.26.13 przedstawia kolejny próbny render. Widać na nim bardziej naturalną, nieregularną gęstość trawy, z lokalnymi prześwitami:



Rysunek 17.26.13 Rezultat — bardziej naturalna, zmienna gęstość trawy

Zajmijmy się z kolei barwą trawy. Na następnej ilustracji (Rysunek 17.26.14) przedstawiam poprawioną wersję ustawień kolorów i shaderów materiału **Grass**. Do tychy ustawień dochodziłem stopniowo, zmieniając jeden z parametrów, wykonując próbny render, i analizując wynik. Zależą od konkretnego zdjęcia tła, które stosujesz.

• Aby poprawnie ocenić efekt trawy, musisz wykonywać próbne rendery w pełnej (tzn. docelowej) rozdzielczości. W mniejszych rezultat może się nieco różnić od ostatecznego! Zmiany, dopasowujące barwę trawy do naszego tła, sprowadzają się do dodania niewielkiego połysku, ale w barwie zbliżonej do jaśniejszej trawy (por. *Material:Spe* i *Shaders:Spec*). Przyciemnienie trawy uzyskałem zmniejszając wartość *Shaders:Ref*, oraz nadając rozproszonemu światłu ciemniejszą barwę (*Material:Col*). Zmianie uległo także parę innych parametrów (Rysunek 17.26.14):



Rysunek 17.26.14 Modyfikacja ustawień materiału Grass

Wartość **Shaders:***Amb* wpływa na zaczernienie końcówek niektórych źdźbeł trawy w obszarze cienia. Dla *Amb* = 0.4 trawa w cieniu ma w sobie coś ze szczeciny. Dla *Amb* = 0.0 wnętrze cienia wygląda (jak dla mnie) zbyt płasko. Dobierz tę wartość odpowiednio do tła, rodzaju trawy, i własnego wyczucia.

Ważny zespół parametrów kryje się w panelu *Links and Pipeline* pod niepozornym napisem *Strands* (Rysunek 17.26.14). Gdy w ten napis klikniesz, rozwija się ni to menu, ni to okno dialogowe (Rysunek 17.26.15). To wartości sterujące kształtem źdźbeł trawy!

Kształt źdźbła jest widoczny np. na tle opon modelu, lub jaśniejszych fragmentów tła. Jeżeli pozostawisz w *Strands* wartości domyślne, źdźbła będą na renderze nieco "druciane". (Mają stałą szerokość, wzdłuż całej długości). Zmieńmy to: pozostaw początkową szerokość źdźbła (*Start*) na 1.0, ale zmniejsz szerokość końca (*End*) do 0.25. Nadaje to trawie kształt trapezu. Dodatkowe zaostrzenie końcówek, oraz wklęsłe boki, uzyskasz zmieniając wartość *Shape* na ujemną (w tym przypadku do -0.5).

Use Tangent Shading				
Surface Diffuse	iurface Diffuse 4 Dist 0.000			
Use Blender Units				
Start 1.00 🗡				
End 0.250 🛛 🛶	Szerokość końca			
Shape -0.500 🖕 💷				
Width Fade 1.000 UV:	Zakrzywienie boków			
Strands Zinvert				
Traceab <mark>n Shadb</mark>	uf RGB HSV			

Rysunek 17.26.15 Kształt źdźbła trawy – parametry Strands

Rysunek 17.26.16 przedstawia kolejny próby render. Tym razem nasza oświetlona trawa, dzięki poprawionym parametrom materiału **Grass**, nie odróżnia się od tej oryginalnej, ze zdjęcia:



Rysunek 17.26.16 Próbny render trawy z poprawionym materiałem Grass

Skoro dopracowaliśmy się już trawy, która dobrze wygląda w słońcu, trzeba zobaczyć, jak sprawdza się w cieniu. Włącz widoczność modelu i wykonaj kolejny render (Rysunek 17.26.17):



Rysunek 17.26.17 Próba z cieniem samolotu

W zasadzie całość wygląda zadowalająco. Zwróć uwagę np. na nieregularny cień, rzucany na trawę przez rurkę Pitota (Rysunek 17.26.17a). Także pojedyncze źdźbła, widoczne na krawędzi cienia, mają realne, cienkie końcówki (Rysunek 17.26.17b). Jedyne, do czego można mieć zastrzeżenia, to duża liczba ciemnych "szczecinek" wewnątrz cienia. Tym efektem sterujesz w materiale **Grass** za pomocą parametru *Shaders:Amb* (por. Rysunek 17.26.14). Przyjąłem dla tego renderu *Amb* = 0.4, i to chyba jest za dużo. Docelowo zdecydowałem się zmniejszyć wartość tego parametru do 0.1, co zdecydowanie zmniejszyło liczbę czarnych źdźbeł trawy.