

Przy generowaniu grafiki 3D procesor nie ma już co robić!

# Trójwymiarowa rewolucja

Nie samymi klatkami karta graficzna żyje! Tak parafrazując znane powiedzenie, można podsumować to, co ostatnimi czasy dzieje się w dziedzinie konstrukcji akceleratorów 3D. Aby sprostać nowym grom, coraz częściej znacznie ważniejsze staje się to, jak karty tworzą trójwymiarową grafikę.

**Marcin Bieńkowski**

**W**zrost wydajności kart graficznych nie jest wcale głównym wyznacznikiem postępu technologicznego, zwłaszcza jeśli chodzi o konstruowanie akceleratorów 3D. Tutaj równie ważny, jeśli nie ważniejszy, okazuje się sposób generowania grafiki. To właśnie on w najnowszych grach przekłada się na liczbę obsługiwanych przez kartę efektów graficznych przy jednoczesnym zachowaniu szybkości wyświetlania obrazu. Ale czy ktoś z Was, Szanowni Czytelnicy, zastanawiał się nad tym, jak tworzone są superrealistyczne wirtualne światy w takich grach, jak Doom 3, FarCry czy Half-Life 2? Zapraszamy Was zatem na wycieczkę do pasjonującego świata grafiki – popatrzymy, jak najnowsze akceleratory 3D tworzą trójwymiarowe cuda na ekranach pecetowych monitorów. Wiedza ta z pewnością

przyda się nie tylko po to, żeby zadziwić znajomych. Przede wszystkim pozwoli ona na podstawie opisu jeszcze w sklepie zorientować się, jak dana gra będzie wyglądała na naszym komputerze – a różnice w wyglądzie w zależności od karty mogą być naprawdę spore!

## DirectX wyznacznikiem postępu

Większość gier komputerowych wykorzystuje programowy interfejs API (Application Programming Interface) systemu Windows, noszący nazwę DirectX. Kolejne jego wersje przystosowane są do możliwości coraz to nowszych kart graficznych. Pozwalają one programistom na znacznie bardziej realistyczne i łatwiejsze modelowanie scen 3D i znajdujących się na nich postaci. Aby dana gra uruchomiła się w pełni

swoich możliwości, musi być zatem zapewniona zgodność pomiędzy wersją DirectX, w której została napisana, i zdolnością karty do sprzętowej realizacji funkcji obsługiwanych przez tę odsłonę DirectX.

Najnowsze gry wymagają stosowania bibliotek DirectX 9.0. Jej sprzętową obsługę zapewniają dwie ostatnie generacje kart – m.in. nVidia GeForce FX 5900, ATI Radeon 9800 oraz nowsze GeForce 6800 i Radeon X800. Karty obsługujące DirectX 8.0 również radzą sobie z problemem zgodności z DX 9.0, jednak odbywa się to zawsze kosztem wydajności i obniżenia jakości generowanego obrazu – wyłączone zostają wówczas efekty nieobsługiwane przez karty. Gorzej jest w wypadku wcześniejszych akceleratorów 3D współpracujących jedynie z bibliotekami DirectX 7.0. Tutaj pojawiają się poważne problemy z kompatybilnością, wynikające ze zmiany sposobu budowy trójwymiarowej sceny, o czym za chwilę.

Podobna sytuacja ma miejsce dla nieco mniej popularnego interfejsu API – OpenGL. Starsze karty są w stanie zapewnić zgodność jedynie do OpenGL 1.2. Nowsze wersje OpenGL-a (1.3, 1.4 i 1.5) obsługują zaś tylko najnowsze akceleratory, z tym że z wersją OpenGL 1.5, podobnie jak z DX 9.0c, w pełni radzą sobie jedynie karty nVidii z serii 6xxx, takie jak wspomniany przed chwilą GeForce 6800.

## Najważniejszy jest strumień

Karta graficzna, aby wyświetlić na ekranie gotowy obraz, musi najpierw wykonać szereg następujących po sobie czynności na danych, które

otrzymała chwilę wcześniej od aplikacji 3D. Te realizowane kolejno kroki obliczeniowe nazywane są strumieniem graficznym (ang. 3D Graphics Pipeline), w którym ze względu na realizowane procesy wydzielić można trzy główne fazy obliczeń. Są to odpowiednio: operacje geometryczne, takie jak np. skalowanie obiektów wraz z kalkulacjami (modelowaniem) oświetlenia, rendering (czyli m.in. procesy nakładania tekstur i cieniowania) oraz rasteryzacja polegająca na przygotowaniu gotowego obrazu do wyświetlenia na monitorze. Oczywiście im więcej operacji przeprowadzanych jest na danych w strumieniu graficznym i im dokładniej są one realizowane, tym wierniej będzie przedstawiona na końcu nasza trójwymiarowa scena.

W najstarszych kartach graficznych, tzw. buforach ramki, sprzętowo przez te urządzenia realizowany był tylko ostatni etap wyświetlania grafiki – rasteryzacja. Całym procesem generowania obrazu, w tym grafiki 3D, zajmował się procesor komputera. Akcelerator, jak sama nazwa wskazuje, ma zaś pomóc w generowaniu obrazu. Takimi urządzeniami spotykanymi na rynku jeszcze kilka lat temu były akceleratory grafiki płaskiej (np. S3 Trio32, S3 Vision 968 czy ATI Mach 64). Układy te odciążały jednostkę centralną podczas wyświetlania wielokątów i linii prostych, przesuwania oraz skalowania okien itp.

Pierwszym domowym akceleratorem 3D (urządzenia profesjonalne produkowano już kilka lat wcześniej) była legendarna karta 3dfx Voodoo, która weszła do sprzedaży w 1996 roku. Kość ta umożliwiała tworzenie sceny 3D, lecz do swojego prawidłowego działania potrzebowała również zwykłej karty 2D. Nic dziwnego, że producenci szybko stworzyli współczesną grupę kart nazywanych obecnie akceleratorami 3D. Urządzenia te potrafią generować i wspomagać operacje związane zarówno z grafiką dwu-, jak i trójwymiarową. Pionierami w produkcji tego typu kart stały się firmy nVidia z układem Riva 128, Matrox z kartami Millennium oraz S3 ze swoim Virge'em. Wszystkie te karty włączały się w przetwarzanie potoku graficznego na etapie renderingu.

Kolejna grupa urządzeń, począwszy od GeForce'a 256 (koniec 1999 roku) z wbudowanym modulem T&L (ang. Transform and Lighting), wspomagała już sporą część operacji geometrycznych i kalkulacji oświetlenia. Dopiero wraz z pojawieniem się bibliotek DirectX 8.0 i urządzeń z jednostkami Vertex i Pixel Shader można mówić o przejściu wszystkich funkcji strumienia graficznego przez kartę. Dzięki temu układ graficzny zaczął coraz częściej nosić nazwę GPU (Graphics Processing Unit). Wiedząc już, co to jest strumień graficzny, popatrzmy zatem, jak składające się na niego przekształcenia realizowane są przez współczesne karty graficzne. Naszą wędrówkę rozpoczniemy od pięciu najważniejszych etapów składających się na operacje geometryczne.

## 1 Punkty i kropki, czyli ustalenie współrzędnych obiektów

Zarówno aplikacja graficzna, jak i gra – niezależnie od tego, czy obliczenia będą realizowane za pomocą procesora czy też układu 3D – na początku musi utworzyć kompletny plan generowanej sceny. Na owej „mapie” zostają wyznaczone dokładne położenia wszystkich bez wyjątku występujących na niej obiektów. Do opisu położenia przedmiotów w przestrzeni wykorzystuje się najprostszą możliwą metodę – zestaw trzech współrzędnych X, Y, Z.

Co więcej, posługując się współrzędnymi X, Y, Z dla kilku lub kilkunastu punktów charakterystycznych (tzw. węzłów) wziętych z każdej bryły tworzącej scenę 3D, można nie tylko wyznaczyć, w którym miejscu w przestrzeni znajduje się dany obiekt, ale również bardzo dokładnie opisać jego kształt i wielkość. W wypadku sześcianu wystarczy poznać współrzędne dla ośmiu jego wierzchołków (werteksów), aby nie mieć najmniejszych wątpliwości, jak duży jest ten obiekt i gdzie jest on umiejscowiony w przestrzeni.

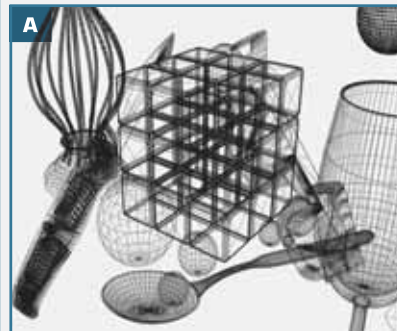
Dla bardziej skomplikowanych kształtów, takich jak np. kula, jabłko czy też czyjaś twarz, wszystkie węzły wyznaczone zostają w kilku, kilkudziesięciu lub kilkuset punktach na powierzchni bryły. Tworzą one wówczas wektorowy opis obiektu na scenie 3D – wszystkie operacje związane z umieszczeniem obiektów i przypisaniem im współrzędnych na scenie 3D wykonuje się raz w operacji wstępnego przetwarzania obrazu (ang. preprocessing). Oczywiście im więcej wyznaczaliśmy punktów na powierzchni przedmiotu, tym dokładniej taka bryła zostanie później narysowana. Niemniej należy też zachować kompromis pomiędzy dokładnością odwzorowania a szybkością obliczeń, gdyż zbyt dużo węzłów znacznie wydłuży czas wykonywania wszystkich operacji w strumieniu graficznym.

## 2 Teselacja: do walki wkracza karta graficzna!

Aplikacja graficzna 3D, taka jak np. gra, przekazuje za pośrednictwem bibliotek DirectX lub OpenGL do sterowników karty ową „kropkowaną” mapę sceny z zaznaczonymi węzłami, należącymi do poszczególnych obiektów. W tym miejscu akcelerator 3D rozpoczyna proces nazywany teselacją lub triangularyzacją (ang. triangle setup processing). Operacja ta polega na grupowaniu należących do każdej bryły węzłów (wierzchołków) w trójkąty. W ten sposób cała generowana scena, w tym powierzchnie sferyczne, podzielona zostaje na mniejsze lub większe płaskie trójkąty utworzone wyłącznie z trójkątów.

Wybranie trójkąta, a nie np. prostokąta czy sześciokąta, jako elementarnego wielokąta będącego podstawowym budulcem trójwymiarowej sceny spowodowane zostało względami praktycznymi związanymi z przyspieszeniem obliczeń. Kalkulacje przeprowadzane na trójkątach są łatwiejsze w implementacji niż rachunki prowadzone na innych wielokątach. Ponadto usprawnione zostają dalsze etapy generowania sceny 3D, takie jak obliczenia natężenia światła, cieniowanie i wypełnianie obszarów teksturami. Należy jednak pamiętać, że prędkość, z jaką są realizowane dalsze przekształcenia, zależy od liczby trójkątów tworzących scenę – im jest ich mniej, tym szybciej prowadzone są obliczenia, ale uzyskane efekty będą mniej realistyczne.

### Fazy tworzenia sceny 3D



**Początkowo każdy obiekt na scenie 3D składa się wyłącznie z wierzchołków stanowiących węzły sieci (A), które poddawane są operacjom geometrycznym. Później w procesie renderingu na pierwotny szkielet nakładają się tekstury (B) i odpowiednio oświetla obiekty. Ostatnim etapem jest wyświetlenie gotowego obrazu na ekranie (C).**

Najstarsze konstrukcje akceleratorów, począwszy od układów 3dfx Voodoo i nVidia Riva 128, oraz moduły 3D spotykane obecnie w wielu chipsetach płyt głównych (np. Intel Graphics Media Accelerator 900 z kości i915) proces teselacji wykonują zawsze samodzielnie. Następnie jednak za pośrednictwem interfejsu API zwracają do CPU podzielony na trójkątne fragmenty szkielet sceny. Ów szkielet poddany jest ponownej obróbce geometrycznej przez procesor



komputera, a w tym czasie akcelerator nie robi nic do chwili rozpoczęcia renderingu.

Z kolei kości graficzne wyposażone w moduł T&L same wykonują dalsze obliczenia, wspomagając się jedynie mocą obliczeniową procesora przy uwzględnianiu interakcji (wzajemnych oddziaływań) między obiektami znajdującymi się na scenie 3D. Akceleratory te mogą same realizować proste operacje geometryczne, takie jak skalowanie obiektów, ich rotację oraz translację, o czym za chwilę, ale nie mają możliwości wykonywania opisanych funkcją dowolnych zmian położenia wierzchołka. Nie radzą sobie one też z zachodzącymi na siebie wierzchołkami.

Powyższego ograniczenia nie mają układy z Vertex Shaderami – wszystko robią same. Niemniej układy te geometryczne operacje skalowania, rotacji i translacji wykonują na wierzchołkach jeszcze przed procesem teselacji. Zmiana kolejności wykonywania zadań wynika z lepszego dopasowania jednostek Vertex Shader do operacji na wierzchołkach niż na trójkątach.

### 3 Ustawianie geometrii i kadrowanie sceny

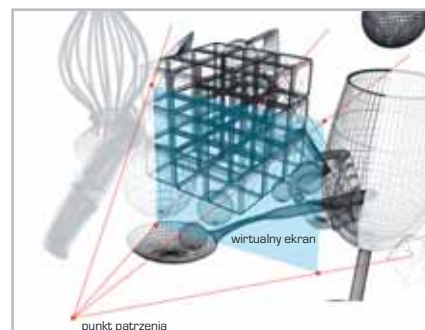
Ponieważ podczas generowania sceny 3D zawsze wykorzystuje się wzorce przedmiotów i postaci przygotowane wcześniej przez programistów i artystów, trudno uniknąć sytuacji, że wstawione na scenę 3D obiekty będą miały nieodpowiednie wymiary. Oczywiście wszelkie obiekty mogą też znajdować się nie w tym

miejsku, gdzie trzeba, oraz być obrócone tyłem w stosunku do obserwatora. Wszystkie obecne na scenie 3D bryły należy zatem (w wypadku starszych kart odbywa się to po procesie teselacji, a w nowych przed) poddać trzem wspomnianym już operacjom geometrycznym, a mianowicie skalowaniu, translacji i rotacji. Mają one za zadanie ustawić obiekty na swoim miejscu (stąd nazwa: ustawianie geometrii) oraz odpowiednio je przeskalować.

Na tym etapie odbywa się też kadrowanie sceny, czyli usunięcie z niej fragmentów niewidocznych dla obserwatora. Dzięki temu zmniejsza się liczba obiektów przetwarzanych w kolejnych fazach w strumieniu graficznym. Niemniej we wszelkich obliczeniach geometrycznych pod uwagę brane są zawsze wszystkie obiekty znajdujące się w polu widzenia, nawet jeśli któryś z nich został zasłonięty przez drugi przedmiot. Liczone są też wszystkie niewidoczne ścianki tyłne.

### 4 Jak akcelerator radzi sobie z oświetleniem sceny 3D

Po ustawieniu geometrii i skadrowaniu sceny przychodzi czas na jej oświetlenie. Kalkulacje te polegają na wprowadzeniu do obliczeń współrzędnych położenia oraz typów źródeł światła: punktowe, rozproszone czy ruchome i jego barwy – monochromatyczne lub kolorowe. W rachunkach uwzględniane zostają też znajdujące się poza kadrem źródła światła.



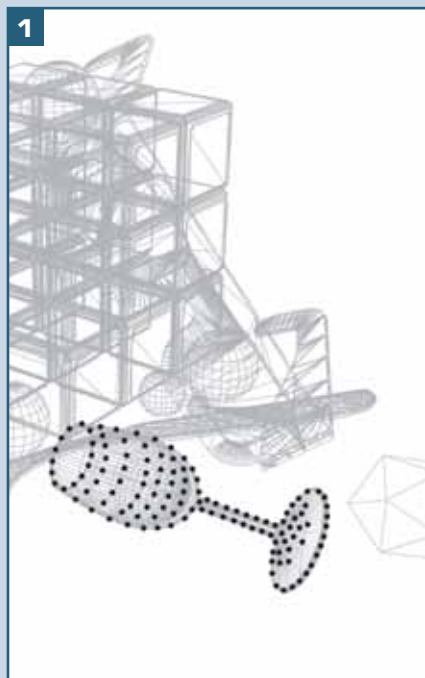
**Usunięcie z pola widzenia niewidocznych obiektów (szare elementy) zmniejsza liczbę obiektów przetwarzanych w dalszych fazach strumienia graficznego. W ten prosty sposób można przyspieszyć generowanie grafiki 3D nawet o ponad 50%.**

Każdemu węzłowi przypisane zostają wektor opisujący natężenie światła w danym miejscu oraz informacja o wypadkowym kolorze światła. W obliczeniach tych pod uwagę brane są też odbicia od powierzchni metalicznych, luster i tym podobnych obiektów, również tych, których bezpośrednio nie widać. Co ciekawe, ze względu na podobieństwo wykonywanych w tej fazie obliczeń (operacje na macierzach) modelowanie oświetlenia zalicza się również do przekształceń geometrycznych.

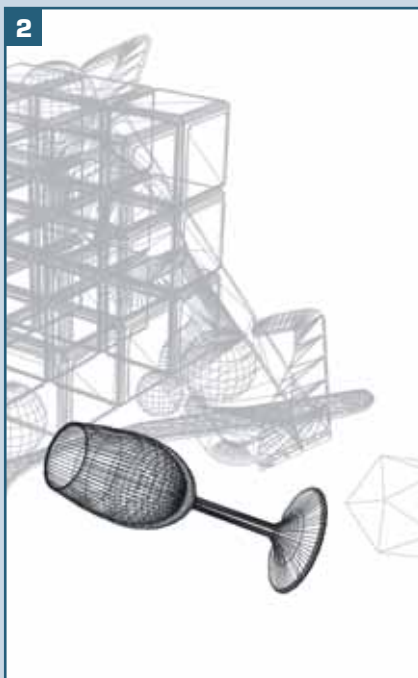
W tym miejscu kończą się możliwości wspomagania operacji geometrycznych przez karty z modułem T&L. Jeżeli w jakiś sposób życzymy

90»

## Strumień graficzny 3D – najważniejsze etapy operacji geometrycznych



Pierwszym krokiem przy tworzeniu obrazu 3D jest wyznaczenie współrzędnych X,Y,Z dla wszystkich punktów (węzłów i wierzchołków) opisujących obiekty znajdujące się na scenie. Czynności te realizowane są przez procesor.



Kolejnym krokiem jest **teselacja**, czyli pogrupowanie węzłów w trójki, na których przeprowadza się dalsze obliczenia. Modele kart graficznych zgodnych z DX 8.0 i 9.0 czynność tę wykonują dopiero po operacjach geometrycznych.



**Ustawianie geometrii** polega na dopasowaniu sceny 3D do punktu patrzenia. Na ustawianie geometrii składają się takie operacje geometryczne, jak skalowanie, rotacja, translacja oraz korekcja perspektywy i kadrowanie sceny.

sobie, aby obiekt na scenie został zdeformowany lub nietypowo oświetlony, np. po trafieniu pociskiem, lub chcemy przedstawić falowanie powierzchni wody, wówczas sterowanie przebiegiem procesu obróbki strumienia graficznego należy znów oddać jednostce centralnej komputera lub posłużyć się programem „shaderowym”.

## 5 Do czego służą shadery, czyli dynamiczne modyfikacje siatki

Pojawienie się w kartach graficznych jednostek Vertex i Pixel Shader wprowadziło małe zamieszanie do klasycznego sposobu generowania grafiki. I nie chodzi tu wcale o wspomnianą już zmianę kolejności wykonywania niektórych operacji, ale o zastąpienie statycznej metody przekształceń obiektów na generowanej scenie 3D na system dynamiczny.

Możliwość wykorzystania dynamicznego bądź statycznego sposobu generowania trójwymiarowej sceny związana jest z mocą obliczeniową procesora graficznego. Operacje statyczne nie potrzebują bardzo wydajnych akceleratorów. Tutaj wszystkie elementy znajdujące się na scenie opisane są wyłącznie za pomocą trzech współrzędnych (X, Y, Z) we względny dla danego „obrazka” układzie odniesienia. Przesunięcie obiektu, np. postaci, odbywa się poprzez wprowadzenie wektora translacji (x, y, z), który po dodaniu do współrzędnych każdego punktu z transformowanego obiektu wyznacza jego



**Dzięki obsłudze Vertex i Pixel Shaderów w wersji 3.0 karty z serii nVidia GeForce 6200, 6600 i 6800 są najbardziej zaawansowanymi technicznie domowymi akceleratorami 3D.**

nowe położenie (X+x, Y+y, Z+z). W obliczeniach statycznych każda klatka animacji jest przeliczana oddzielnie, a obsługa interakcji pomiędzy przesuwającym się obiektem a otoczeniem spoczywa na jednostce centralnej komputera.

Bardziej zaawansowany obliczeniowo model dynamiczny wykorzystuje jedną przygotowaną w procesie tzw. preprocessingu scenę 3D oraz krótkie programy zawierające ciąg instrukcji modelujących zachowanie się obiektu na kilku lub kilkunastu kolejno generowanych klatkach. Innymi słowy każda postać widoczna na scenie 3D będzie miała zapisaną własną funkcję opisującą jej zachowanie – np. kierunek ruchu bohatera gry – oraz model „obsługi” interakcji z otoczeniem. Sam proces przetwarzania programów (nazywanych shaderami, stąd nazwa

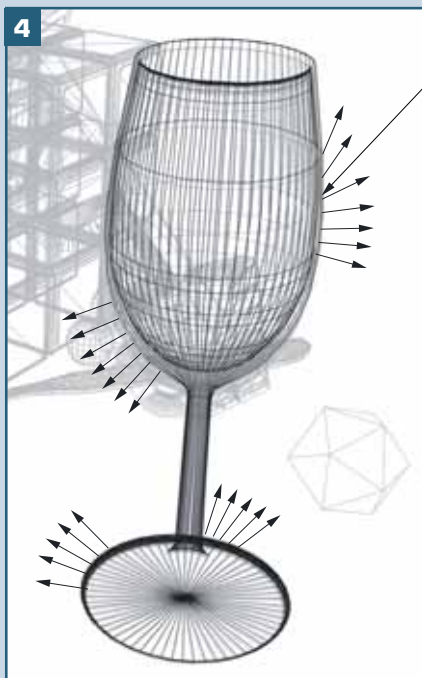
jednostek Vertex i Pixel Shader) opisujących zachowanie się obiektów realizowany jest w karcie graficznej, dzięki czemu CPU komputera może w danej chwili zająć się innymi zadaniami związanymi z obsługą gry. „Shaderowy” model opisu generowanej sceny 3D daje znacznie większe możliwości szczegółowego odwzorowania przedmiotów i postaci.

Wprowadzenie modelu wykorzystującego programy „shaderowe” z punktu widzenia zasad generowania grafiki 3D nie wniosło niczego nowego. Przeniesiono jedynie proces modyfikacji generowanej sceny 3D z jednostki centralnej na procesor znajdujący się na karcie graficznej. Same sposoby tworzenia trójwymiarowej sceny pozostały bez zmian! Co więcej, sterowniki najnowszych akceleratorów mają swobodną możliwość programowej konwersji „shaderowego” opisu sceny 3D na model statyczny i odwrotnie – w starszych kartach proces ten nie jest możliwy ze względu na brak sprzętowych jednostek Vertex i Pixel Shader. Niemniej warto wiedzieć, że Vertex Shadery dają się dość łatwo zastąpić emulacją software’ową w sterownikach, co skwapliwie wykorzystują producenci mniej zaawansowanych akceleratorów 3D.

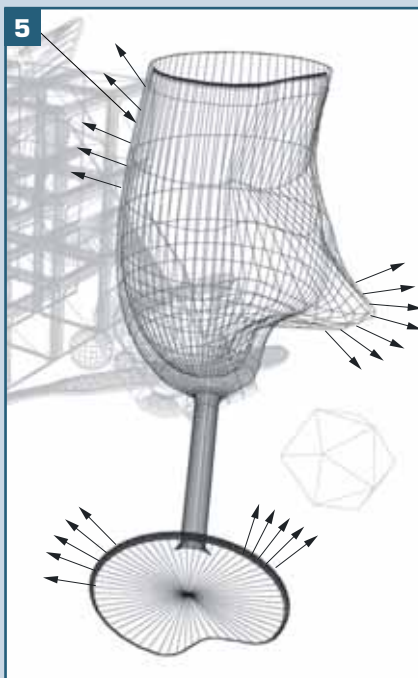
## Jeśli się da, to nie licz na dwa

I tak dotarliśmy do miejsca, w którym karta dysponuje gotowym, statycznym obrazem szkieletu sceny 3D. Obraz ten zostaje zachowany w pamięci karty. Chodzi mianowicie o skrócenie czasu obliczeń kolejnej klatki animacji, tak jak ma to miejsce we wszystkich grach komputerowych. Wówczas bardzo czasochłonne obliczenia geometryczne wykonywane są raz. Dzięki temu można przyspieszyć obliczenia nawet do 50%, ograniczając się w następnych scenach do niewielkich modyfikacji punktu widzenia obserwatora, zmian położenia niektórych tylko obiektów, ponownych kalkulacji oświetlenia (część przedmiotów może zasłonić źródła światła) oraz powtórzenia renderingu.

No dobrze, ale co dalej dzieje się z owym trójwymiarowym szkieletem sceny – wszak my go wcale nie widzimy na ekranie? Otóż teraz pozostało mu nadać ostatni szlif, nakładając na ten stelaż realistyczną powłokę. Proces ten jest nie mniej skomplikowany niż obliczenia geometryczne. Może się on też składać, w zależności od możliwości karty, z kilku lub kilkunastu etapów, ale o tym napiszemy w jednym z najbliższych numerów CHIP-a. K



**Kalkulacje oświetlenia (zwane też modelowaniem) polegają na przypisaniu każdemu węzłowi wektora o wartościach odpowiadających natężeniu, kierunkowi i barwie światła padającego na dany wierzchołek oświetlanej bryły.**



**Dynamiczne operacje modyfikacji położenia węzłów siatki i zmiany wartości wektorów oświetlenia realizowane są wewnątrz akceleratora 3D dzięki modułom Vertex Shader i krótkim programom „shaderowym”.**

## Więcej informacji

Informacje na temat generowania grafiki 3D

<http://www.beyond3d.com/>

### Literatura

„GPU Gems – Programming Techniques, Tips, and Tricks for Real-Time Graphics”, Edited by Randima Fernando, nVidia & Addison-Wesley, Boston 2004