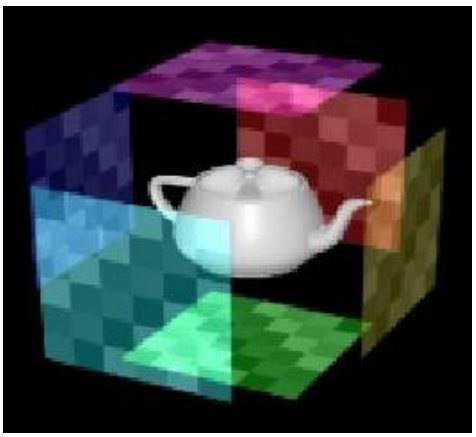
Wiemy zatem już sporo o mapowaniu środowiskowym - jak emulować rozbłyski, bardzo błyszczące powierzchnie i ogólnie materiały, które znacząco wpływają na ożywienie naszej sceny. Technika mapowania sferycznego, którą poznaliśmy umożliwia w bardzo prosty sposób stworzenie właśnie takich efektów, jednak z faktu, że jest jedynie sztuczką dlatego także jest ograniczona w pewien dla nas denerwujący sposób - nie możemy sobie zmieniać naszego środowiska, bo wszystko zepsujemy. Dzisiaj dowiemy się jak temu zaradzić, czyli powiemy coś niecoś o mapowaniu sześciennym.

Na sam początek spróbujmy sobie wyobrazić otaczający nasz świat bez rozbłysków. Niby na co dzień nie zwracamy tak szczególnie na to uwagi, bo jesteśmy do tego przyzwyczajeni, ale gdyby nam to zabrać sporo w tym naszym świecie by się zapewne zmieniło - zacząłby trochę przypominać nam te nasze sceny z plastykowymi sześcianami, które są jakieś takie bez wyrazu. Od kiedy wymyślono symulację świata rzeczywistego na komputerach za pomocą grafiki 3D omawiane przez nas odbicia stanowią dla wszystkich twórców zarówno oprogramowania i sprzętu potężne wyzwanie, któremu do pewnego stopnia udało się sprostać. Na samym początku walki o bardziej realistyczny świat do boju wytoczono ciężkie działa pod postacią mapowania sferycznego - co to takiego mam nadzieję, że już zdajecie sobie doskonale sprawę. Ponieważ w czasach kiedy było one wymyślone sprzęt jeszcze nie był może najwyższych lotów więc radzono sobie w miarę możliwości. Różne kombinacje, założenia i ograniczenia oczywiście spowodowały to, że mapowanie sferyczne spełnia swoją rolę, ale tylko pod poważnymi warunkami, których absolutnie należy przestrzegać aby osiągnąć zamierzony efekt. Jak wiemy, mapa środowiska, która ma odzwierciedlać to co się odbija w obiekcie jest tworzona jeszcze przed przystąpieniem do renderingu - wygenerowanie takiej mapy w czasie rzeczywistym jest jak na możliwości nawet dzisiejszego sprzętu trochę trudne i raczej się go nie stosuje.

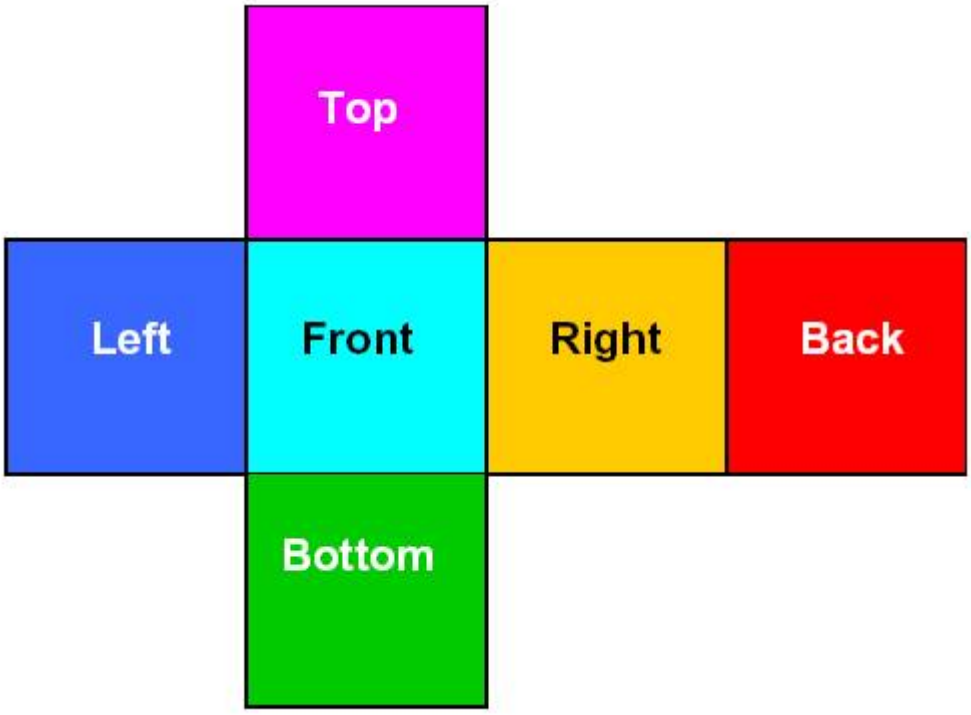
- Pierwszym więc, bardzo poważnym ograniczeniem było to, że otoczenie naszego obiektu nie może się za bardzo zmieniać, jeśli wszystko miało grać - dopóki wokół obiektu znajdowało się, to co reprezentowała mapa wszystko było w porządku - ale kiedy zaczynało się coś dziać wszystko brało w łeb i już trzeba było kombinować.
- Po drugie widok, z którego patrzyliśmy na obiekt - wiadomo nawet z doświadczenia, że jeśli zmienimy punkt naszego patrzenia to z naszego punktu widzenia otoczenie się zmienia - więc sytuacja stawała się analogiczna do tej z poprzedniego przypadku - w zasadzie mogliśmy poruszać się bez obaw wokół obiektu tylko w bardzo małym zakresie i to nie osiągając jakiś oszałamiających rezultatów.
- Kolejna rzecz - przygotowanie mapy środowiska - w zasadzie wszystkie techniki tworzenia takowej opierają się na tym, że próbuje się prostokątną mapę reprezentującą jakiś obraz (fragment widzianej przestrzeni) rozłożyć na powierzchni kuli i pokazać ją z kolei na całkiem płaskim obrazie. Niesie to ze sobą niewątpliwie całe mnóstwo możliwości zakłóceń, co okazało się w praktyce bardzo uciążliwe i uniemożliwiło właśnie oglądanie obiektów z różnych stron.

Ponieważ sprzęt poszedł szybko do przodu, myśl techniczna oczywiście wyprzedzała wszystko o lata, więc wkrótce stało się całkiem jasne, że to nie wystarczy - wymagania użytkowników ciągle rosły, sprzęt potrafił coraz więcej, więc zaczęto wymyślać coraz to nowe techniki. Jak wiadomo najciemniej jest zawsze pod latarnią, ale komuś coś nie umknęło i doczekaliśmy się naszego dzisiejszego bohatera - jak to ktoś ładnie określił - proste rozwiązanie skomplikowanego problemu.

Na czym tak naprawdę polega mapowanie środowiskowe? Najogólniejsza idea ciągle pozostaje ta sama, jak w przypadku mapowania sferycznego - otaczające nasz obiekt inne obiekty (czyli środowisko) należy zrenderować do tekstury a następnie w jakiś sposób nałożyć na nasz główny, błyszczący obiekt zainteresowania. Ale tym razem postawiono sobie za cel trochę więcej niż tylko ładne odblaski- tym razem odblaski te mają przedstawiać dokładnie to, co dzieje się na scenie. Zapewne po wielu głębokich przemyśleniach okazało się, że przy wsparciu sprzętu wszystko da się zrobić, co lepsze wyszło na jaw także to, że wbrew pozorom można to zrobić o wiele prościej niż poprzednio! Zamieniając mapę środowiska z mozolnie wygenerowanej kuli obciążonej zakłóceniami na sześć ścian otaczających obiekt, który ma zaszczytny cel reprezentowania na swojej powierzchni całego otaczającego go świata ułatwiono sobie zadanie na tyle, że dzisiaj nawet dla nas, początkujących okaże się to banalnie proste! Ale może, żeby od razu nie skakać na głęboką wodę zacznijmy o rysunków poglądowych. Najpierw wyobraźmy sobie już słynny reprezentacyjny "czajnik", któremu przyjdzie robić dzisiaj za lusterko. Spójrzmy na rysunek poniżej:



Jak się domyślacie, kolorowe płaszczyzny rozmieszczone wokół czajniczka będą w jakiś tam sposób powiązane z naszą mapą środowiska. Są to tak zwane tekstury projekcyjne - nazwa brzmi trochę groźnie, ale tak naprawdę nie jest to nic strasznego. Są to po prosty kolejne obrazy zrenderowane przy odpowiednim ustawieniu kamery dla potrzeb przykładu. Najprościej będzie sobie to wyobrazić w ten sposób, że środek czajnika znajduje się w środku układu współrzędnych, oko obserwatora (kamery) w każdym przypadku znajduje się w środku tego układu natomiast dla każdej z tekstur projekcyjnych (których jak łatwo policzyć jest sześć) cel kamery jest ustawiany w innym kierunku. Ponieważ znajdujemy się w środku układu współrzędnych więc bardzo łatwo będzie ustawiać cel - dla przykładu w kierunku każdej z osi układu. Kierunek poszczególnych wektorów, według których orientowany jest cel kamery w zasadzie nie miałby aż takiego znaczenia, gdyby nie jeden mały szczegół - wszystkie kolejne płaszczyzny, które są prostopadłe do wektora łączącego oko z celem kamery muszą być też wzajemnie prostopadłe względem siebie, co oczywiście pociąga za sobą prostopadłość tych wektorów. Ponieważ jednak osie układu współrzędnych mają tę jedną, jedyną cechę, więc nadają się do tego celu wręcz idealnie. Jak znowu sobie łatwo wyimaginować, kolejne tekstury będą po kolei oznaczać przód obiektu, tył, oba boki, górę i dół. Biorąc powyższy rysunek za przykład możemy sobie nasze tekstury projekcyjne przedstawić w postaci siatki sześcianu otaczającego nasz czajnik:



Dokładnie przypatrując się poszczególnym płaszczyznom da się zauważyć, wzdłuż której osi i w jakim kierunku patrzenia kamery powstały poszczególne - poniższa tabelka powinna wyjaśnić wam wątpliwości, jeśli macie jakiekolwiek:

Tekstura	Oko	Cel
Przód	(0, 0, 0)	(0, 0,-1)
Tył	(0, 0, 0)	(0, 0, 1)
Lewa	(0, 0, 0)	(-1, 0, 0)
Prawa	(0, 0, 0)	(1, 0, 0)
Góra	(0, 0, 0)	(0, 1, 0)
Dół	(0, 0, 0)	(0,-1, 0)

Jak widać tekstury projekcyjnie tworzą wokół czajnika coś w rodzaju sześcianu, stąd także nazwa naszego mapowania - sześcienne. A teraz najważniejsza dla nas kwestia, czyli dlaczego mapowanie sześcienne będzie dla nas lepsze niż mapowanie sferyczne.

- Po pierwsze - łatwość tworzenia mapy świata. Sześć tekstur projekcyjnych możemy spokojnie wyrenderować nawet jako początkujący, tylko odpowiednio ustawiając naszą kamerę według osi układu współrzędnych. Mając wyrenderowanych sześć różnych map dla każdego kierunku, które (co ważne!) przedstawiają aktualny stan sceny wokół środka z którego renderujemy możemy z niewielką pomocą naszego sprzętu odpowiednio umieścić na obiekcie i uzyskać bardzo pożądany przez nas od dłuższego czasu efekt.

- Po drugie - niezależność od położenia kamery. Ponieważ generowanie naszej mapy odbywa się w taki a nie inny sposób więc po wyrenderowaniu wszystkich jej składowych i odpowiednim ich nałożeniu na obiekt możemy sobie potem dowolnie skakać po scenie kamerą, która naszej mapie już nie zaszkodzi, bo ta będzie już na obiekcie. I w zależności od tego, z której strony popatrzymy na obiekt zobaczymy dokładnie to co chcemy zobaczyć - odbicie środowiska, które tam jest bo chwilę wcześniej zostało zobaczone przez inną kamerę i wyrenderowane do mapy reprezentującej otoczenie!
- Po trzecie - szybkość. Wydawać by się mogło na pierwszy rzut oka, że wygenerowanie jednej mapy środowiska dla mapowania sferycznego powinno przebiegać szybciej niż robienie sześciu map dla każdej ściany naszego sześcianu. Jednak nic bardziej mylącego - obliczenia jakie trzeba wykonać dla mapowania sferycznego są o wiele bardziej skomplikowane i czasochłonne niż prosty rendering sceny w widoku z kamery a to nasz akcelerator potrafi robić naprawdę znakomicie.
- Po czwarte - brak zakłóceń. Pamiętamy, że mapa sferyczna mogła zawierać różne zniekształcenia i artefakty, które mogły powodować różne niemiłe niespodzianki podczas nakładania i oglądania obiektu. Mapa sześcienna przy odpowiednich ustawieniach reprezentuje dokładnie wycinki świata, które po ustawieniu ich na mapie sześciennej dadzą ciągły obraz przestrzeni nic nie urywając i nie tracąc.
- Po piąte, ostatnie i najważniejsze - mapa reprezentująca środowisko zawiera dokładnie to, co dzieje się na scenie. Nie musimy się więc już martwić, że nie możemy się poruszyć wokół obiektu ani, że nic nie może się zmienić. Będzie dokładnie odwrotnie - rozkoszować się możemy od teraz dynamicznie zmieniającym się otoczeniem obiektu jak i dowolnie zmieniać punkt widzenia. To na pewien czas na pewno powinno nas zadowolić.

Jednak jak wszystkie sztuczki tak i ta ma pewne wady i ograniczenia, które jednak wynikają dopiero przy bardziej skomplikowanych modelach, zwłaszcza posiadających wklęsłości, dziury i tym podobne rzeczy, ale i z nimi jakoś sobie można poradzić stosując różne manewry i sztuczki a także kombinując z różnymi rodzajami mapowania jednocześnie. Ale z praktyki wynika, że takie przypadki zdarzają się naprawdę rzadko a na konkretne zakłócenia trzeba dobierać techniki indywidualnie, bo nie ma jednej określonej i stuprocentowo skutecznej metody. Ponieważ jak wiemy mapa środowiska nie musi być bardzo dokładna i szczegółowa więc w większości zastosowań to co mamy zupełnie nam wystarczy, zwłaszcza że wszystko mamy niejako na żywo :-). Widz oszołomiony efektownym obrazem na pewno nie zwróci uwagi na ewentualne drobne niedoskonałości formy.

Ponieważ wiemy już jak mniej więcej to wszystko działa i skąd się bierze, więc czas na wyjaśnienie co w tym wszystkim robi nasz sprzęt oraz co my, jako programiści będziemy musieli policzyć, żeby cokolwiek osiągnąć. Zacznijmy może od sprzętu, bo to dzięki niemu będziemy mogli się rozkoszować nowym efektem, który przyprawia o szybsze bicie serca. Wspominałem o teksturach projekcyjnych i o nakładaniu ich na obiekt. Ponieważ w programie przykładowym zobaczymy, że na obiekt nakładana będzie tylko jedna tekstura specjalnego typu więc pytanie teraz co się dzieje z teksturami projekcyjnymi. Otóż w programie zostanie stworzony specjalny typ tekstury - sześcienna, która niejako na sobie przechowa w odpowiedni sposób kolejne wyrenderowane składowe tekstury projekcyjne. Każdą kolejną wyrenderowaną w kolejnym kierunku tekstura sześcienna przechowa do dalszego przetworzenia. Tworzenie właśnie takich tekstur będzie nam umożliwiało urządzenia, które posiada wsparcie dla mapowania sześciennego. O drugą sprawie już wspominałem, ale przypomnę. Ponieważ akcelerator jest na tyle mocny, żeby wyrenderować scenę w zwykły sposób, więc podobnie jak w przypadku renderingu całej sceny zastosujemy tutaj jego umiejętności. Najpierw wyrenderujemy scenę bez obiektu, który ma odbijać otoczenie, wygenerujemy mapę sześcienną a następnie obliczając odpowiednie współrzędne mapowania nałożymy tę mapę sześcienną na obiekt i wyrenderujemy scenę ponownie już z udziałem naszego obiektu. Może wydawać się skomplikowane i powolne, ale jak sami się przekonacie zadziała naprawdę zaskakująco dobrze i efektywnie ;-)).

Sprawa druga - obliczanie współrzędnych mapowania, no bo skoro mamy teksturę, to dla niej trzeba je właśnie policzyć. W tym celu napiszemy ponownie nasz własny shader bez polegania na gotowych rozwiązaniach (które oczywiście istnieją), ale my z nich nie skorzystamy, bo chcemy się czegoś konkretnego nauczyć. Wydawać by się mogło, że w tym momencie sprawa nam się bardzo skomplikuje, ponieważ dostaniemy do ręki jakąś cudowną teksturę o nieznanym typie i tak naprawdę nie będziemy wiedzieć co z nią zrobić i w jaki sposób policzyć dla niej współrzędne. Sprawa jednak okazuje się jeszcze bardziej banalna niż w przypadku mapowania sferycznego - na pewno pamiętacie te wprawdzie niezbyt skomplikowane wzory , ale zawsze to było jakieś utrudnienie. Obliczaliśmy tam w pewnym momencie coś takiego jak wektor refleksu ze wzoru:

$$R = V - 2 \cdot d \cdot N$$

Okazuje się, że dla typu sześciennego tekstur wspomaganych przez nasze urządzenie ten wektor w zupełności wystarczy! Nie musimy nic więcej wiedzieć tak naprawdę - tylko tyle, że urządzenie umożliwia nam stworzenie niezbędnego formatu tekstury oraz, że współrzędne mapowania to nic innego jak obliczony wektor refleksu (oczywiście znaczenie będą miały dwie pierwsze składowe x i y, bo tylko takie znajdziemy na teksturze)!

No i na koniec nadszedł czas na pytania filozoficzno - egzystencjalne, czyli do czego może się nam to przydać? Oczywiście pierwsze i najważniejsze zadanie to generowanie refleksów na powierzchni w czasie rzeczywistym. O dzisiaj wokół naszego obiektu będzie mogło sobie krążyć mnóstwo mniejszych lub większych jego towarzyszy a jeśli my sobie zażyczymy, żeby na przykład nasz przykładowy czajnik stał się chromowany to on zgodnie z naszym życzeniem odbije wszystkie szalejące wokół niego przedmioty. Za pomocą mapowania sferycznego nie zrobilibyśmy tego na pewno :-). Możemy też za pomocą

mapowania środowiska mapować refleksy i odbicia światła. Zamiast generować obiekt o bardzo dużej liczbie trójkątów i ustawiać w programie światła, które tylko wszystko spowolnią możemy stworzyć jakieś mapy symulujące oświetlenie w środowisku i nakładać to za pomocą mapowania sześciennego na interesujący nas element sceny - ale to jest temat na oddzielny artykuł. Mam nadzieję, że ten krótki opis pomoże wam zrozumieć jak to wszystko działa tak nieco dokładniej, bo wbrew pozorom to naprawdę jest bardzo proste i powinno na zawsze zapaść wam w pamięć jako prosty sposób na skomplikowany problem ;-)).