

Witam.

Dzisiaj będzie trochę nietypowy ale bardzo potrzebny tutorial. Już zrobiliśmy bardzo dużo jeśli chodzi o samego DirectX'a, znamy sporo różnych fajnych sztuczek, ale pewnie wielu z was ma niejasne wrażenie, że czegoś mu brakuje. Owszem, potrafimy posługiwać się macierzami, możemy sobie obiektami obracać, przesuwac i robić co nam się tylko podoba. Mamy gotowe funkcje do tworzenia potrzebnych nam macierzy przekształceń bez wgłębiania się w istotę ich budowy. Wielu z was zapewne jednak czuje (i zupełnie słusznie), że jeśli nie dowiemy się w końcu prawdy jak to wszystko działa to będziemy mieć coraz trudniej. Aby stworzyć jakieś wymyślne sztuczki będziemy potrzebować wiedzy na trochę niższym poziomie niż dotychczas. Wgłębimy się więc w tym tutorialu w istotę matematyki stosowanej, na której to opiera się dzisiaj praktycznie cała grafika 3D. W serii tych tutoriali dowiemy się co to wektor, macierz i inne podobne wynalazki, jakie można wykonać na nich operacje i po co nam one będą potrzebne. I zapewniam, że jeśli znudziła wam się już trochę męka z naszymi tutorialami to odkryjemy na nowo piękno całej tej grafiki 3D ;). W dzisiejszym tutorialiku powiem więc sobie o wektorach. Tak w zasadzie powinniście wiedzieć czym jest wektor i do czego on służy, ale tak dla porządku powiemy sobie dzisiaj od początku do końca.

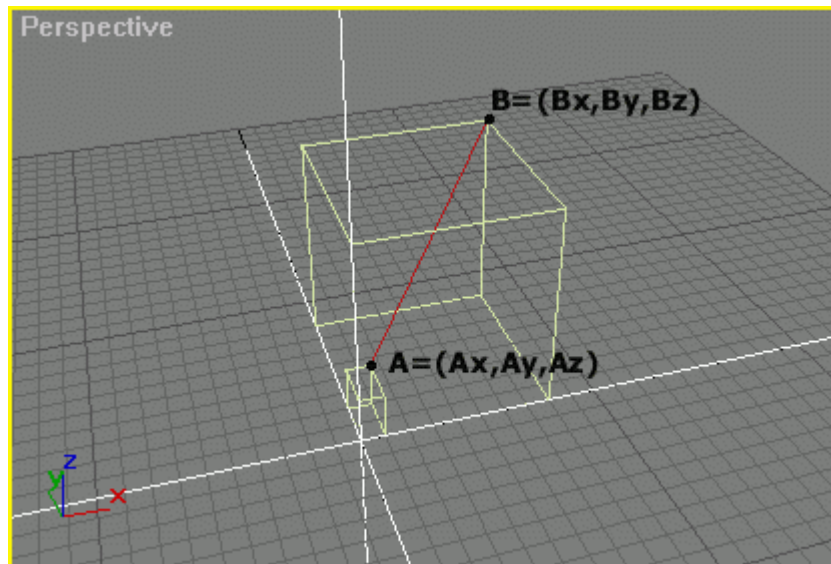
A więc przechodząc do sedna sprawy. Przypomnimy sobie po prawdzie co ważniejsze pojęcia, ale będzie to tylko po łebkach. Pytanie coż więc będziemy takiego robić. Otóż wrócimy do całkowitych podstaw. Dowiemy się o operacjach na wektorach, potem przejdziemy do macierzy a skończymy na zaawansowanych zagadnieniach i sztuczkach, które będziemy wykorzystywać w dalszej nauce DirectX. Rozpoczniemy od rzeczy najprostszych a zapewniam was, że po tej serii tutoriali grafika 3D nie będzie miała dla was tajemnic. Zacznijmy więc bez zwlekania.

Pierwszy raz o wektorach pewnie słyszeliście na fizyce a może na matematyce. Pamiętacie na pewno strzałki rysowane przez Pana/Panią na tablicy pokazujące kierunek lotu pocisku albo prędkość ciała na zmienawidzonej równi pochyłej ;). Może większość z was nie rozumiała a może nawet do dziś nie rozumie co tak naprawdę oznaczał ten wektor. Teraz właśnie spróbujemy wytłumaczyć to sobie. Wyobraźmy sobie samochód jadący po drodze ruchem jednostajnym po prostej. Jedzie on w określonym kierunku, pokazywanym przez nasz wektor. I to jest chyba jasne - kierunek tej strzałki to inaczej kierunek wektora. Samochód jak wiemy może jechać szybciej lub wolniej - powstaje zatem pytanie - jak to można by pokazać za pomocą wektora. Robiliście na pewno to w szkole wiele razy a może nawet nie zwracaliście na to uwagi. Rzecz zupełnie przecież banalna - im dłuższą strzałkę (wektor) narysujemy, będzie to znaczyło, że samochód jedzie szybciej. Mamy więc na tapecie dwie podstawowe własności wektora - jego kierunek i długość. Wszyscy pewnie zadają sobie teraz pytanie po jaką cholere o tym mówię. Co to ma wspólnego z grafiką komputerową ? Jak wielu się już przekonało a wielu przekona - bardzo, bardzo wiele. Zarówno kierunek wektora jak i jego długość to będą dwie wielkości, z których będziemy bardzo często korzystać. Aby jednak móc z nich korzystać musimy wiedzieć do czego nam się przydadzą. W skrócie mówiąc - przydadzą nam się... do wszystkiego. Do wyznaczania punktów, do obliczania odległości, identyfikowania powierzchni, animacji itd. itd. Można by rzec, że grafika 3D wektorem stoi. No dobra - piszę i piszę a czym tak naprawdę jest ten wektor. Jedną z bardziej przydatnych nam - komputerowym grafikom definicji mówi, że wektor jest to pewien zbiór liczb rzeczywistych, zawierający n - elementów. Dla nas to "n" będzie jasno i ściśle określone. Jeśli będziemy robić grafikę to nasze wektory będą zbiorami, a należałoby raczej powiedzieć, że tablicami liczb. Dla 2D taka tablica będzie zawierać 2 liczby, dla grafiki 3D $n = 3$. Więcej wymiarów na razie nie jest przewidziane w naszym świecie, ale kto wie, co przyniesie przyszłość ;). Wektory jak pamiętacie oznacza się strzałką, coś w tym stylu na przykład:



Coż dla nas mogą oznaczać te poszczególne liczby. Nie możemy zapomnieć, że patrzmy cały czas na to wszystko przez pryzmat współrzędnych, brył trójwymiarowych i przekształceń. Czym dla nas może być tablica liczb rzeczywistych ? Pierwszym i zupełnie oczywistym zastosowaniem wektora jest przechowywanie w nim współrzędnych naszych punktów. Jest to bardzo dobre miejsce. Opierając się na tym można by więc powiedzieć, że wektor określa położenie punktu w przestrzeni. I dokładnie tak będzie jeśli założymy pewną rzecz - musimy mieć zdefiniowany układ współrzędnych, w którym nasze punkty umieścimy ! Zapomnijmy na chwilę o strzałkach widzianych na fizyce czy matematyce rysowanych na tablicy. Czy nie mając układu współrzędnych mogą istnieć wektory ? Ależ oczywiście ! Przecież napisałem, że wektor to tylko tablica liczb ! Wyobraźcie sobie tablicę zawierającą na przykład 10 liczb - to przecież też będzie wektor ! Tak samo jak zawierający 1 czy 100 liczb... to cały czas jest wektor.

My jednak będziemy traktować wektory jako coś bardziej namacalnego a nawet widocznego. Dla nas wektory będzie można przedstawić też w trochę innej formie niż tylko tablicę liczb. My wektory będziemy mogli wręcz zobaczyć. Oczywiście niesie to ze sobą pewne ograniczenia co do ilości jego składowych, ale o tym już wiemy. Więc jeśli założymy, że mamy nasze odpowiednie wektory, mamy układ współrzędnych i mamy przestrzeń, w której możemy widzieć - czym będą składowe wektora ? Jak wiemy, układ współrzędnych ma taką fajną cechę jak osie współrzędnych z osiami i jednostkami na tych osiach. Każdy punkt, jaki sobie umieścimy w takim układzie możemy sobie jednoznacznie określić, zarówno w przestrzeni dwu- jak i trójwymiarowej. W porządku, to wiemy. Co natomiast z wektorami ? Załóżmy, że mamy wektor dwuwymiarowy. Za tym oczywiście idzie ograniczenie naszej przestrzeni do dwóch wymiarów. Czym będą składowe wektora ? Będą one tym, czym są dla punktów poszczególne wartości - będą współrzędnymi wektora. Ponieważ jednak wektor różni się trochę od punktu, więc te współrzędne to nie będzie dokładnie to samo. Współrzędne naszego wektora "graficznego" będą określać przez ile jednostek wzdłuż każdej osi nasza strzałka się ciągnie. Popatrzcie na rysunek poniżej, który powinien wam co nieco wyjaśnić.



Wektor wisi sobie gdzieś w powietrzu. Patrząc na niego wzdłuż którejkolwiek z osi układu współrzędnych możemy policzyć przez na ile jednostek na tej osi jest on długi. I tak patrząc kolejno na wszystkie osie odczytamy sobie kolejne "współrzędne". W tablicy liczb, która będzie naszym wektorem umieścimy sobie kolejno ilość tych jednostek na osi X, Y i Z. Skoro wiemy już co będą dla nas oznaczać kolejne składowe wektora wróćmy do naszego punktu. Napisałem, że wektor może określać położenie punktu w przestrzeni. Dlaczego? Zastanówmy się teraz jeszcze raz nad współrzędnymi wektora. Nasz wektor w przestrzeni jak łatwo zauważyć można w prosty sposób określić - ma on jakiś koniec i jakiś początek. Są to... no oczywiście - punkty! Punkty te mają swoje współrzędne. Punkt **A** to (A_x, A_y, A_z) i punkt **B** (B_x, B_y, B_z) . Czy mając te współrzędne można obliczyć współrzędne wektora? Uczyli was na pewno tego w szkole średniej, ale dla zapominalskich powiem - jasne, że tak! A jak? Nic prostszego - jeśli dobrze się przyjrzeć to wystarczy od każdej składowej punktu oznaczającego koniec wektora odjąć odpowiednią składową punktu oznaczającego początek. Więc mając dwa punkty możemy sobie określić wektor jako:

A = (A_x, A_y, A_z) - początek wektora
B = (B_x, B_y, B_z) - koniec wektora, więc

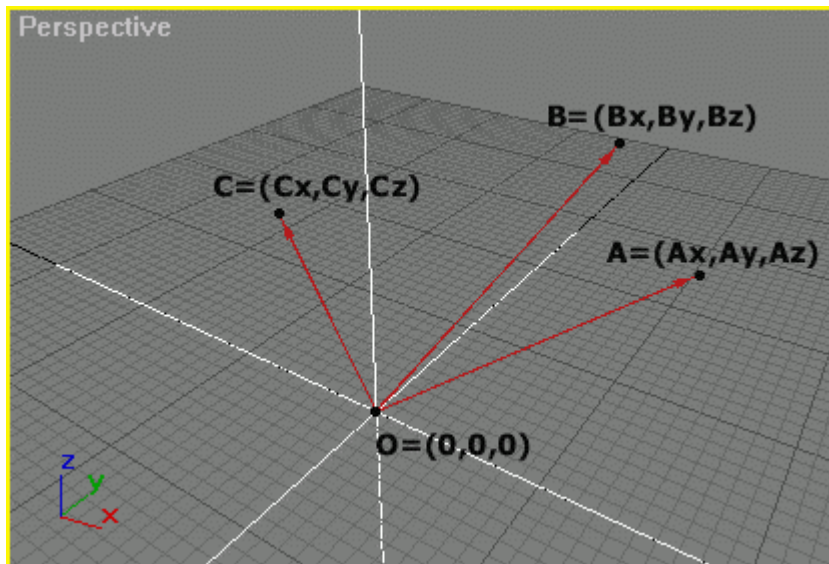
$$\vec{u} = [B_x - A_x, B_y - A_y, B_z - A_z]$$

i w ten sposób otrzymujemy dobrze nam znaną tablicę liczb. A teraz znowu wracając do punktów i ich położenia. Ponieważ już wiemy jak robić wektory więc stwórzmy sobie parę. Mamy punkty, które chcemy opisać za pomocą wektorów, wiecie zatem jak to zrobić? Stwórzmy wektory, których końce będą w punktach, które nas interesują a początki w środku układu współrzędnych, czyli w punkcie $(0, 0, 0)$. Jeśli teraz policzymy sobie współrzędne wektorów to otrzymamy co? Od razu oczywiście przykład:

A = (A_x, A_y, A_z) - koniec wektora, nasz punkt.
O = $(0, 0, 0)$ - początek wektora.

$$\vec{u} = [A_x - O_x, A_y - O_y, A_z - O_z] = [A_x, A_y, A_z]$$

Przypatrzmy się współrzędnym punktu i wektora. Wszystko jest już jasne? Jeśli nie to dodatkowy rysunek dla wątpliwych ;)



No dobra, skoro mamy takie wektory to co my w zasadzie możemy sobie z nimi zrobić. Możemy wykonać dla przykładu na nich pewne operacje. Operacje, które pozwolą nam na dokonywanie sztuczek, w wyniku których krytycznym "ogładczy" naszych dzieł opadną szczęki, ale zanim to nastąpi będziemy musieli jeszcze się pomęczyć. Pierwszą z operacji będzie dodawanie. Ale nie będzie to dodawanie zwykłe jak można się domyśleć. No bo pomyślcie - jak dodać dwie tablice liczb? Co nam z tego wyjdzie? Dodawanie wektorów odbywać się będzie na trochę innej zasadzie niż zwykle. Dodawać do siebie będziemy poszczególne składowe naszych tablic. Jest to dosyć logiczne bo jeśli dodajemy do siebie dwa wektory to chcielibyśmy żeby wyszedł nam trzeci, prawda? A jak to robić? Ano najlepiej zobaczyć sobie znowu na przykładzie. Będziemy potrzebować dwóch wektorów:

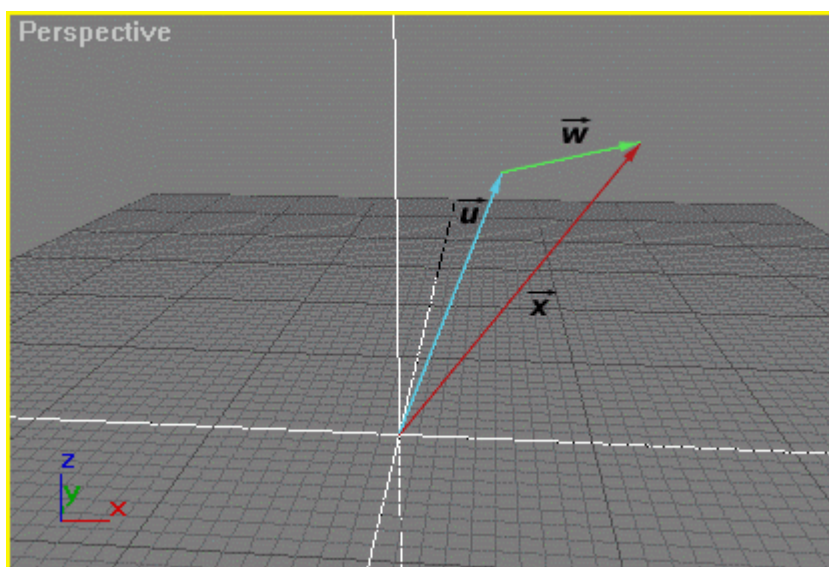
$$\mathbf{u} = [u_1, u_2, u_3]$$

$$\mathbf{w} = [w_1, w_2, w_3]$$

Jeśli chcemy je teraz dodać to robimy tak:

$$\vec{x} = [u_1 + w_1, u_2 + w_2, u_3 + w_3]$$

Wychodzi nam trzeci, którego kolejne składowe są sumą odpowiadających sobie składowych z wektorów, które dodajemy. Myślę, że jest to dla wszystkich jasne? Oczywiście to samo przekłada nam się na nasze wektory graficzne. Po dodaniu dwóch wektorów, które są narysowane w naszej przestrzeni powstanie nam trzeci.



Można oczywiście obejść się doskonale bez pomocy tablic i korzystać metod graficznych do dokonywania operacji na wektorach (jak na przykład reguła równoległoboku) ale nam one zupełnie się nie przydadzą. My zapamiętać powinniśmy raczej właśnie operacje czysto matematyczne. Oczywiście odejmowanie wektorów będzie przebiegać na tej samej zasadzie. Czyli przy założeniu, że mamy te same wektory wejściowe możemy wykonać:

$$\vec{x} = \vec{u} - \vec{w} = [u_1 - w_1, u_2 - w_2, u_3 - w_3]$$

Drugą operacją jest mnożenie wektora przez liczbę rzeczywistą. Jest to równie proste jak dodawanie, z tą różnicą że teraz mamy jeden wektor i jedną liczbę. Istota tej operacji polega na tym, że każdą składową wektora mnożymy przez naszą liczbę, o tak:

a - czyli liczba, przez którą mnożymy.

u = [**u1**, **u2**, **u3**] - oczywiście nasz wektor przez który mnożymy

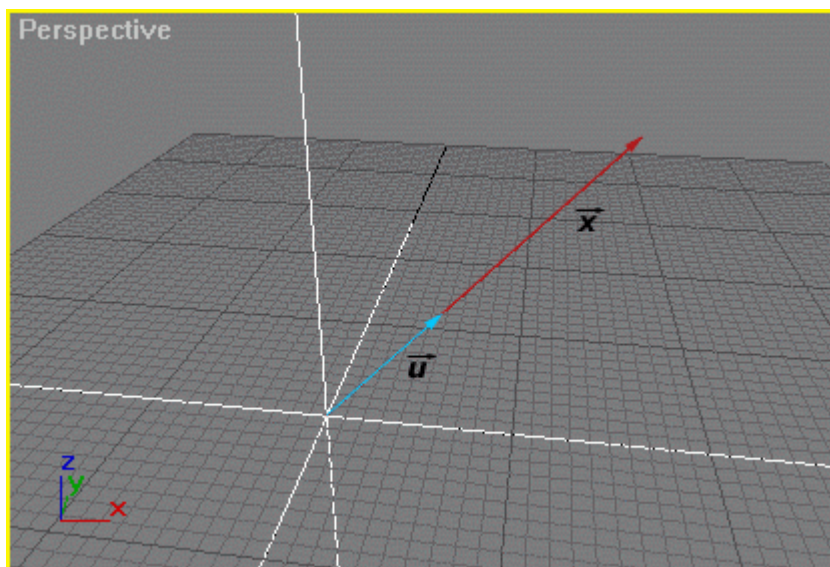
Mnożenie w zapisie matematycznym wygląda tak:

$$\vec{x} = \vec{u} \cdot a = [u_1 \cdot a, u_2 \cdot a, u_3 \cdot a]$$

Sądzę, że nie wymaga to większego komentarza ? Skoro mamy mnożenie, nie może się także obejść bez dzielenia. I tak samo jak w przypadku dodawania i odejmowania będziemy mieć analogię:

$$\vec{x} = \frac{\vec{u}}{a} = \left[\frac{u_1}{a}, \frac{u_2}{a}, \frac{u_3}{a} \right]$$

Aby uzmysłowić wam jak działa mnożenie i dzielenie oczywiście mały rysunek prezentujący wprawdzie mnożenie, ale w drugą stronę nie trudno sobie chyba wyobrazić...



Powyżej wspomniałem o długości wektora. Czym ona jest ? Jeśli wyobrazimy sobie nasz wektor graficznie to idąc niejako "po nim" w przestrzeni będziemy mogli powiedzieć na ile jednostek (które są na osiach układu współrzędnych) nasz wektor się ciągnie. Wielkość ta będzie liczbą rzeczywistą, jak każda inna długość - w tym względzie wektory nie różnią się niczym od innych wytworów matematyki. A jak policzyć jego długość ? Jak zwykle posłużymy się naszym ulubieńcem:

$$\mathbf{u} = [\mathbf{u1}, \mathbf{u2}, \mathbf{u3}].$$

W takim przypadku jego długość wyrażona zostanie wzorem:

$$|\vec{u}| = \sqrt{(u_1)^2 + (u_2)^2 + (u_3)^2}$$

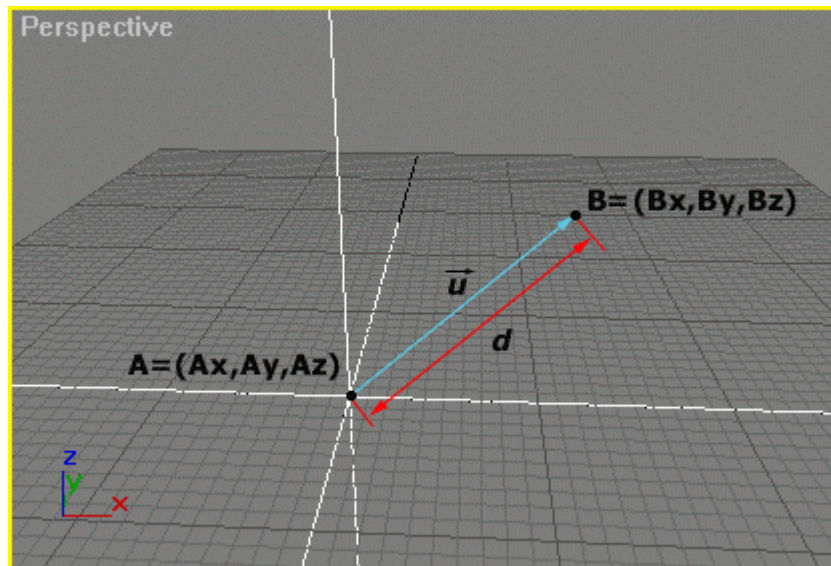
Długość wektora będziemy oznaczali jako $|\mathbf{u}|$. Bardzo łatwo zauważyć, że znając wzór na długość wektora możemy sobie obliczyć odległość pomiędzy punktami w przestrzeni, prawda ? Wystarczy tylko wrócić trochę wstecz i już będziemy wiedzieć jak:

A = (**Ax**, **Ay**, **Az**) - punkt oznaczający początek wektora,

B = (**Bx**, **By**, **Bz**) - punkt oznaczający koniec wektora, więc nasza szukana odległość:

$$d = \sqrt{(B_x - A_x)^2 + (B_y - A_y)^2 + (B_z - A_z)^2}$$

Założę się, że ten wzór też był w szkole :). No i nie możemy oczywiście zapomnieć o rysunczku:



Na pewno też natknęliście się już w dokumentacji na takie pojęcie jak wektor jednostkowy, może słyszeliście też coś takiego jak wersor (to to samo). Co to jest ? Otóż jest to wektor o długości 1. Jeśli narysujemy sobie pewien wektor w przestrzeni, zmierzmy go linijką z jednostkami, które są na naszych osiach i okaże, że tych jednostek nasz wektor ma dokładnie 1 to będzie on wektorem jednostkowym. Dotychczas mogliście się spotykać z wektorami kierunkowymi na osiach układu współrzędnych, które to wyznaczały kierunek tych osi. Każdy z takich wektorów miał oczywiście długość 1 a ponieważ każdy z nich był równoległy do którejś z osi więc dwie z jego współrzędnych musiały mieć wartość zero a jedna równała się 1. Dla poszczególnych osi współrzędne wektorów wynosiły więc:

$$\begin{aligned} \vec{x} &= [1, 0, 0] \\ \vec{y} &= [0, 1, 0] \\ \vec{z} &= [0, 0, 1] \end{aligned}$$

Jeśli wektor jednostkowy będzie natomiast skierowany gdzieś w świat (po dowolnym kącie) to jego współrzędne zaczną trochę się gmatwać, ale łatwo się także domyśleć, że składowe nie będą nigdy mogły przekroczyć wartości 1. Dlaczego piszę o wektorach jednostkowych ? Otóż dla pakietów graficznych obliczanie pewnych wartości lepiej jest dokonywać na wektorach jednostkowych. Mogą czasem wystąpić problemy jeśli na przykład będziemy chcieli obliczyć jakiś kąt pomiędzy wektorami i podamy takie wektory, których długości wynoszą na przykład 1000. Dlatego też często istnieje potrzeba tak zwanej normalizacji wektora, czyli nadania mu długości 1 bez zmiany jego kierunku. Jak tego dokonać ? Ano nic prostszego:

$$d = |\vec{v}|$$

$$\vec{v}^1 = \frac{\vec{v}}{d} = \left[\frac{v_x}{d}, \frac{v_y}{d}, \frac{v_z}{d} \right]$$

czyli dzielimy każdą składową wektora przez jego długość. Po takiej operacji będziemy mieli nowy wektor o takim samym kierunku jak nasz stary, ale będzie on miał długość 1.

Ale to jeszcze nie koniec. Przechodzi do fajniejszej zabawy. Aby osiągać niebanalne efekty w dzisiejszej grafice musimy się dowiedzieć o dwóch specjalnych operacjach jakich możemy dokonać za pomocą dwóch wektorów. Są to iloczyny wektorów (mnożenie). I pewnie wielu z was nie zaskoczę jeśli powiem o dwóch jego rodzajach. Będziemy mieć więc do czynienia z tak zwanym iloczynem kartezjańskim i skalarnym. Pewnie wielu z was czytając dokumentację do DX SDK lub literaturę obcojęzyczną natknęło się na określenia "cross product" i "dot product". "Cross Product" to na nasze iloczyn kartezjański, natomiast "Dot Product" to inaczej iloczyn skalarny. Czym jest jedno i drugie ? Ano, żeby wyjaśnić powiedzmy najpierw sobie o bardzo, bardzo istotnej rzeczy. Wyobraźmy sobie dwa dowolne wektory w naszej przestrzeni \mathbf{u} i \mathbf{w} . Leżą sobie one gdzieś, dowolnie. Mają swoje współrzędne, czyli $\mathbf{u} = [u_1, u_2, u_3]$ i $\mathbf{w} = [w_1, w_2, w_3]$. Zauważmy, że wektor ma taką

ciekawą a jednocześnie ważną właściwość - jeśli przesuniemy oba końcowe punkty wektora na taką samą odległość w takim samym kierunku to... no właśnie, to co ? Otóż - punkty po przesunięciu wyznaczać będą wyznaczać nowy wektor, co jest chyba logiczne. Ale wektor ten będzie... no jaki ? Czy już wiadomo ? No tak ! Będzie dokładnie taki sam, jak przed przesunięciem wierzchołków ! Mówiąc taki sam mam na myśli to, że będzie miał ten sam kierunek i tę samą długość ! Więc zapamiętajmy raz na zawsze. Przesuwanie punktów określających wektor, a co za tym idzie i samego wektora nie zmienia zupełnie jego właściwości ! Jeśli ktoś nie wierzy, niech narysuje choćby na płaszczyźnie 2D jakiś wektor, poprzesuwa jego końce o te same wartości i sprawdzi jego współrzędne. Właściwość ta jest bardzo, bardzo ważna. A dlaczego - zaraz się okaże.

Wracając zatem do naszych iloczynów omówimy sobie może najpierw iloczyn skalarny (dot product). Czym to jest ? Iloczyn skalarny dwóch wektorów w zapisie matematycznym będzie następujący. Jeśli założymy, że mamy dwa wektory o n - rozmiarze, czyli:

$$\mathbf{u} = [u_1, u_2, u_3, \dots, u_n]$$

$$\mathbf{w} = [w_1, w_2, w_3, \dots, w_n],$$

to ich iloczyn skalarny będzie miał postać:

$$\vec{u} \cdot \vec{w} = u_1 \cdot w_1 + u_2 \cdot w_2 + u_3 \cdot w_3 + \dots + u_n \cdot w_n$$

Taki jest sens matematyczny. Ktoś oczywiście zapyta natychmiast jaki to ma sens graficzny. Co może mieć wspólnego liczba rzeczywista (bo to jak widać będzie wynikiem takiego iloczynu) z grafiką 3D ? Jeśli nadal będziemy myśleć o tym iloczynie w ten sposób to faktycznie nic nam się nie nasunie ciekawego i moglibyśmy dać mu spokój. Ale był ktoś w przeszłości, kto postanowił się dowiedzieć więcej i w czasie swoich poszukiwań przypadkiem wymyślił inny wzór na iloczyn skalarny wektorów. Popatrzmy zatem na to:

$$\vec{u} \cdot \vec{w} = \cos(\alpha) \cdot \|\vec{u}\| \cdot \|\vec{w}\|$$

gdzie $\|\mathbf{u}\|$ i $\|\mathbf{w}\|$ to oczywiście długości wektorów, α - kąt pomiędzy zawarty pomiędzy tymi wektorami a cosinus to... no wiadomo co ;) . Ktoś znowu zawoła - zaraz, zaraz... jaki kąt ? Przecież te wektory leżą zupełnie gdzieś dowolnie w przestrzeni a kąt to wszyscy wiedzą jak wygląda, więc skąd ten kąt się bierze ? Wróćmy do przesuwania wektorów w przestrzeni. Powiedzieliśmy sobie, że przesunięcie nic nie zmienia jeśli chodzi o kierunek i wartości wektora. No to co nam teraz szkodzi przesunąć jeden z wektorów (w zasadzie dowolny z nich) tak, aby jego początek znalazł się w początku wektora drugiego ? Nic ! Absolutnie nic. Możemy go tam przesunąć no i wtedy już wszyscy chyba zobaczą ten kąt pomiędzy nimi, prawda ? Ale ponieważ przesunięcie nic nie zmienia, więc my sobie nie będziemy go przesuwac ! No bo po co to robić skoro to niczego nie zmienia ? We wzorze widzimy, że będą nam potrzebne długości wektorów i kąt pomiędzy nimi. Na pewno przesunięcie nam długości nie zmieni, kierunku wektorów też nie a więc i kąt się nie zmienia ! Więc możemy sobie spokojnie nasze wektory zostawić na miejscu i policzyć dla nich iloczyn skalarny ! Po cóż nam będzie potrzebny taki iloczyn skalarny. Wyobraźmy sobie na przykład, że piszemy nowy sposób oświetlania naszej sceny, bo nie chcemy korzystać ani z cieniowania płaskiego ani Gourauda ani żadnego innego. Chcemy na przykład stworzyć nowe, genialne cieniowanie Kowalskiego. na pewno do tego celu będzie nam potrzebna jedna rzecz. Kąt jaki tworzy promień światła z wektorem normalnym (co to już za moment). Promienie światła rozchodzą się różnych kierunkach i w zależności od tego pod jakim kątem padają na powierzchnię tak ją rozświetlają. My w naszym programie będziemy przeważnie podawać pozycję światła, jego kierunek, pozycje wierzchołków i tym podobne. Będziemy więc na ogół posługiwać się wektorami i punktami. Skąd więc wziąć ten kąt ? Ano właśnie przed chwilą dowiedzieliśmy się skąd ! Posłuży nam do tego właśnie iloczyn skalarny. Przypatrzmy się wzorowi jeszcze raz:

$$\vec{u} \cdot \vec{w} = \cos(\alpha) \cdot \|\vec{u}\| \cdot \|\vec{w}\|$$

Właściwie już stąd moglibyśmy policzyć kąt, ale nie byłoby to dobre rozwiązanie. Długości wektorów mogą nam tutaj bruździć wprowadzając pewne niedokładności (jeśli będą one bardzo duże o co nietrudno w przypadku położenia światła na przykład). Co możemy sobie zrobić ? Ano... znormalizujemy nasze wektory ! Kierunki im się nie zmieniają więc żadne kąty także nie. Po normalizacji zatem długości obydwu będą wynosić odpowiednio $\|\mathbf{u}\| = 1$ i $\|\mathbf{w}\| = 1$. Wstawmy to do naszego wzoru:

$$\vec{u} \cdot \vec{w} = \cos(\alpha) \cdot 1 \cdot 1 = \cos(\alpha)$$

;) , aż miło popatrzeć prawda ? Taki prosty wzorek. Teraz nadszedł czas aby połączyć teorię matematyczną z praktyką graficzną. My będziemy znać tylko wektory, ich kierunki i współrzędne. Chcemy kąt, cóż więc robić ? Ano przypomnijmy sobie pierwszy wzór matematyczny:

$$\vec{u} \cdot \vec{w} = u_1 \cdot w_1 + u_2 \cdot w_2 + u_3 \cdot w_3 + \dots + u_n \cdot w_n$$

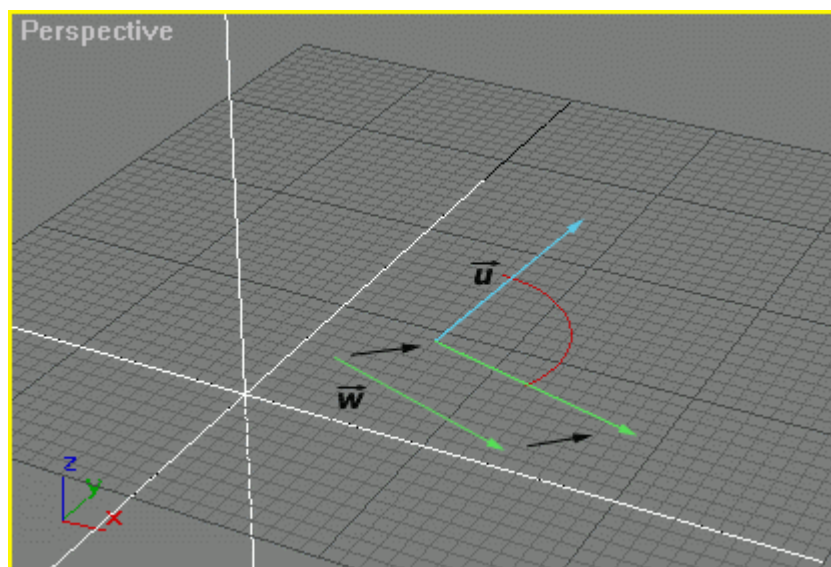
Mając dane naszych wektorów możemy sobie to bez problemu policzyć ! Mamy już więc prawie to czego szukamy. Wstawiamy dalej i możemy sobie napisać:

$$u_1 \cdot w_1 + u_2 \cdot w_2 + u_3 \cdot w_3 + \dots + u_n \cdot w_n = \cos(\alpha)$$

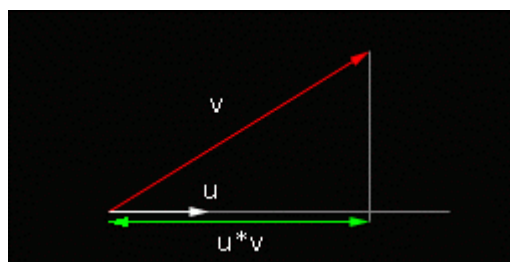
stąd szukany nasz kąt będzie wynosił:

$$\alpha = \arccos(u_1 \cdot w_1 + u_2 \cdot w_2 + u_3 \cdot w_3 + \dots + u_n \cdot w_n)$$

To taki prosty przykład na to, do czego może nam się przydać iloczyn skalarny wektorów. Jak się okaże w następnych lekcjach DirectX będziemy z niego często korzystać, zwłaszcza, jeśli będziemy pisać własne shadery mające coś wspólnego ze światłem, nakładaniem tekstur czy zaawansowanymi przekształceniami brył. Będziemy mogli go używać do rzutowania, obliczania normalnych, ale konkretne zastosowania wyjdą na pewno w praktyce. Tutaj nie będziemy sobie wszystkich omawiać bo zamieszalibyśmy za bardzo. A żeby nie było, że coś nie rozumiemy to aż się prosi o ilustrację:



Przy iloczynie skalarnym musimy omówić sobie jeszcze jedną ciekawą właściwość tego działania, która przyda nam się w grafice nie raz. Otóż wyobraźmy sobie, że mamy dwa wektory, z których jeden będzie znormalizowany a drugi nie. Co to dla nas będzie oznaczać ? Ano ktoś zauważył taką interesującą rzecz - po przeprowadzeniu operacji iloczynu skalarnego na takich dwóch wektorach otrzymamy sytuację jak na rysunku:



Wektor \mathbf{u} jest wektorem jednostkowym, \mathbf{v} nie. Interesujący będzie wynik iloczynu skalarnego. Z naszych wzorów będzie bowiem wynikać dosyć prosta zależność:

$$\vec{u} \cdot \vec{v} = \cos(\alpha) \cdot |\vec{v}|$$

Jak widać wartość $\mathbf{u} \cdot \mathbf{v}$ będzie sobą przedstawiać długość odcinka, który powstanie po zrzutowaniu wektora niejednostkowego, na prostą wyznaczoną przez wektor jednostkowy. Taka właściwość przyda nam się przy wyznaczaniu odległość w przestrzeni - na przykład przy obliczaniu odległości punktu od płaszczyzny i w kilku innych miejscach, na pewno będziemy jeszcze mieli okazję się z tym zapoznać.

Drugim iloczynem związanym z wektorami będzie iloczyn wektorowy. W przeciwieństwie do poprzedniego, wynikiem tego

działania będzie kolejny wektor. Nie będę już omawiał sztuczek z przesuwaniem wektorów i łączeniem ich początków aby mieć je razem, bo ufam, że już doskonale to rozumiecie. Zajmijmy się więc może tylko tą, równie ważną operacją. Jak zwykle na początku podam matematyczny zapis iloczynu wektorowego. Jeśli mamy dane dwa wektory $\mathbf{u} = [u_1, u_2, u_3, \dots, u_n]$ i $\mathbf{w} = [w_1, w_2, w_3, \dots, w_n]$, to ich iloczyn skalarny będzie miał postać:

$$\vec{u} \cdot \vec{w} = x$$

gdzie:

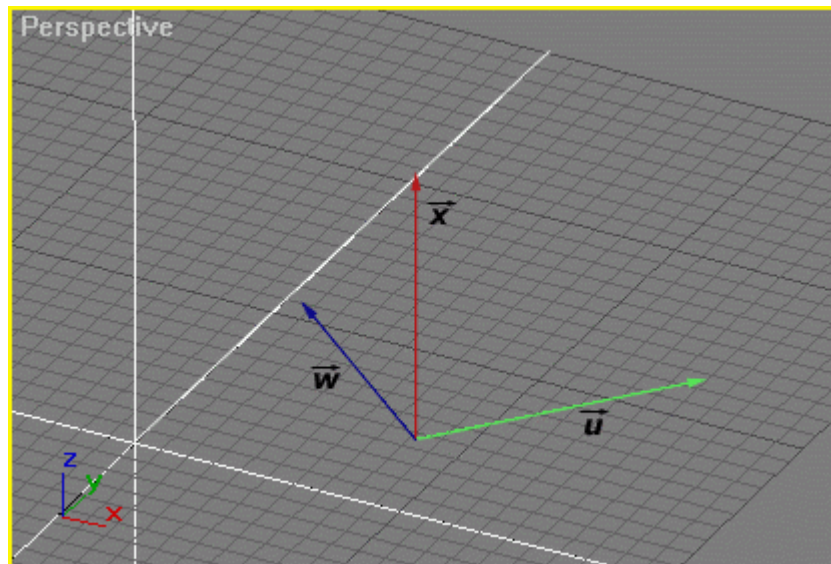
$$\begin{aligned} x_1 &= u_2 \cdot w_3 - u_3 \cdot w_2 \\ x_2 &= u_1 \cdot w_3 - u_3 \cdot w_1 \\ x_3 &= u_1 \cdot w_2 - u_2 \cdot w_1 \end{aligned}$$

gdzie \mathbf{x} jest nowym wektorem powstałym po pewnej operacji na wektorach \mathbf{u} i \mathbf{w} . Większość z nas korzystała wiele razy z tej operacji nawet o niej nie wiedząc. Taki pakiet jak DirectX oblicza iloczyny wektorowe przy pewnych operacjach, które są mu potrzebne nawet bez naszej wiedzy. A oblicza je po co? - ktoś zapyta. I bardzo słusznie. Kto pyta ten na naszym kursie dużo się dowie :). Jest wektor, więc od razu nas korci wiedzieć, co będzie sobą reprezentował taki wektor. Jeśli wsiąść dwa wektory źródłowe i połączyć razem ich początki to otrzymamy trójkę punktów. Wspólny początek i dwa końce. Co charakterystyczne dla punktów, ich trójka, nie leżąca na jednej linii wyznacza nam płaszczyznę. Jedną jedyną, na której leżą owe punkty a co za tym idzie i wektory, które to wyznaczają nasze punkty. Wektor \mathbf{x} , który utworzymy po operacji iloczynu wektorowego będzie się charakteryzował tym, że będzie on prostopadły do tej płaszczyzny, wyznaczonej przez nasze wektory. A co za tym idzie będzie prostopadły do obydwóch tych wektorów co jest oczywiste. Mam nadzieję, że jest to dla was zupełnie jasne. Pewne wątpliwości może budzić jedynie kierunek tego wektora. Jak łatwo również zauważyć, wektorów prostopadłych może (jeśli chodzi o kierunek) będzie para. Jeden skierowany w jedną stronę do płaszczyzny a drugi w przeciwną. Będą one innymi słowy leżały na jednej linii, ale będą skierowane w przeciwną stronę. Oczywiście, po wyniku operacji iloczynu wektorowego, powstanie nam jeden. To, w którą stronę płaszczyzny będzie on skierowany zależy od kolejności, w jakiej będziemy podawać wektory, które mnożymy przez siebie. Załóżmy, że policzyliśmy sobie wektor prostopadły do danych dwóch. Jeśli teraz wykonamy jeszcze raz operację iloczynu ale z zamienioną kolejnością wektorów wejściowych to otrzymamy wektor przeciwny do wyniku poprzedniego działania - czyli po prostu skierowany w drugą stronę. Jest to bardzo istotne i warto to zapamiętać. Skąd będziemy wiedzieć w którą stronę będzie skierowany nasz wektor? Wszystko zależy od tego jak uznamy sobie kierunki na naszej scenie. Jeśli będziemy stosować technikę ukrywania powierzchni za pomocą kolejności rysowanych wierzchołków to często doświadczalnie trzeba będzie sprawdzić która kolejność wektorów tworzących nasze trójkąty da nam pożądany efekt. Ogólna zasada jest jednak taka. Jeśli używamy lewoskrętnego układu współrzędnych (jak DirectX) to układamy nasze palce zgodnie z regułą lewej ręki wzdłuż wektorów biorących udział w iloczynie wektorowym. Regułę tę mam nadzieję wszyscy pamiętają. Jeśli ułożymy nasze palce odpowiednio wtedy kciuk pokaże nam kierunek wektora powstałego z iloczynu. Oczywiście przy rysowaniu w 3D nie będziemy mieli możliwości włożyć ręki do monitora i sprawdzić jak będą układały się normalne. Jeśli jednak będziemy renderować powierzchnie tylko z jednej strony (za pomocą kolejności wierzchołków) to na pewno wektory utworzone z wierzchołków w takiej kolejności jak trójkąty po dokonaniu na nich operacji iloczynu wektorowego utworzą normalne skierowane w stronę widocznej części trójkąta, który same tworzą. Większość programów do modelowania 3D potrafi wręcz generować normalne dla nas i zapisywać je w pliku. Nie musimy się więc tym zbytnio martwić. Co warto jeszcze zauważyć. Co z długością wektora prostopadłego? Ano, tutaj mamy taki ciekawy wzór, który być może też nam się kiedyś przyda, w każdym razie warto go zapamiętać.

$$\left| \vec{x} \right| = \left| \vec{u} \right| \cdot \left| \vec{w} \right| \cdot \sin(\alpha)$$

$\|\mathbf{u}\|$ i $\|\mathbf{w}\|$ to oczywiście długości wektorów składowych, natomiast z sinusa bierzemy wartość bezwzględną (chyba nie musimy nikomu tłumaczyć co to jest).

I znowu powstaje pytanie - po co nam iloczyn wektorowy? Jedno już wiemy na pewno - do obliczania wektorów prostopadłych do danych dwóch. Może się nam to przydać, kiedy na przykład "ręcznie" wprowadzimy jakąś bryłę i będziemy potrzebować jakiś wektorów prostopadłych, może bardzo się przydać przy obliczaniu kolizji na naszej scenie w jakimś nowym genialnym silniku do gry 3D. Przyda się także do wyznaczania parametrów powierzchni, równań i kilku innych rzeczy. A jak wygląda? Proszę bardzo:



No i cóż. To chyba by było na tyle. Mam nadzieję, że artykuł wam się spodoba i wektory już nie będą miały dla was od dzisiaj tajemnic. Takich podstaw nam brakowało od początku, ale tym odcinkiem zaczynamy nową serię, w której nadrobimy nasze zaległości czysto matematyczne. Nie ma oczywiście żadnego kodu źródłowego, choć pewnie można by napisać jakiś ciekawy program. Może zatem Wy wpadniecie na jakiś ciekawy pomysł? Może jakaś fajna prezentacja działania na wektorach z ich zastosowaniem w grafice 3D? Skoro już wszystko o nich wiecie ;)...