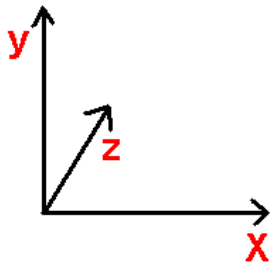
Pov-Ray część 1

ver. 2.1

Stworzono wiele programów graficznych. Na lekcjach Technologii Informacyjnej poznasz jeden z ciekawszych, czyli Pov-Ray. Służy on do generowania obrazów trójwymiarowych zwanych potocznie 3D, jednak w odróżnieniu od innych programów graficznych nie ma on możliwości rysowania od ręki - jak ma to miejsce w innych edytorach grafiki rastrowej typu Paint. Pov-Ray pozwala na opis różnych obiektów za pomocą wiersza poleceń Kiedy już opiszemy wszystkie kształty, ich położenie, kolory itd. Możemy ustawić kamerę i zlecić programowi wykonanie zdjęcia stworzonej przez was scenki.
Poleceń w Pov-Ray jest bardzo dużo, gdyż nawet na jego potrzeby stworzono specjalny język oparty na języku C++. Program dostępny jest bezpłatnie pod adresem www.pov-ray.org . Znajdziecie tam także różne wersje tego programu zarówno 3.1 (na którym będziemy pracować na lekcjach) oraz wersja 3.6 (nowsza z większą ilością komponentów, ale zasadniczo nie różniąca się od poprzedniej w składni poleceń). Instrukcje do programu można znaleźć w samym programie w języku angielskim w menu "help" (w wersji 3.6)
Skrót Pov-Ray to rozwinięcie magicznej nazwy The Persistence of Vision™ Ray-Tracer .Program działa w systemach MS-Dos, Windows 3.x, Windows 95, Windows 98/ME, Windows NT, Windows 2000, Windows XP, Apple Macintosh 68k, Macintosh Power PC, Linux, Sun-OS, UNIX i innych.

Zanim przejdziemy do opisu pierwszej sceny niezbędna będzie nam trochę teorii:
Program operuje na opisie obiektów w przestrzeni trójwymiarowej. Każdy element opisuje się trzema współrzędnymi <x,y,z> , które odpowiadają osiom jak na rysunku poniżej:



Załóżmy, że środek monitora to punkt <0,0,0> regulacja w lewo-prawo powoduje zmiany na osi x (pierwsza współrzędna), góra-dół to oś y (druga współrzędna), problemy pojawiają się zwykle przy osi z(trzecia współrzędna), gdyż wbrew intuicji wartości dodatnie z skierowane są w stronę monitora (strzałka skierowana w naszą stronę to "–z")

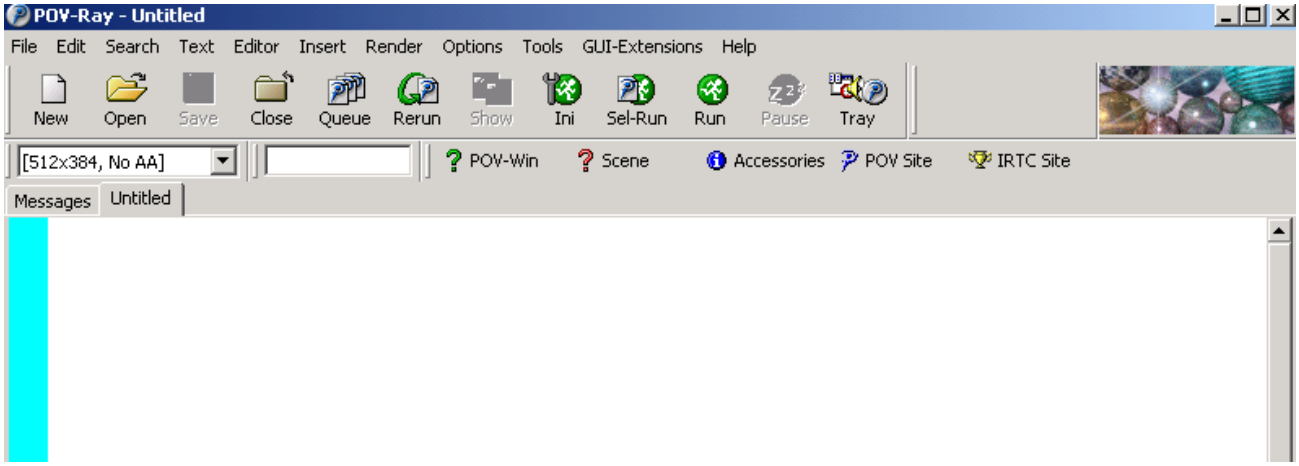
Pamiętajmy:
Zapis <1,2,8> oznacza przesunięcie o jeden w prawo, dwie jednostki w górę i 8 i przed nami (w "dal")

Zacznijmy tworzyć obiekty. Zanim stworzymy pierwszy ustalmy sobie jakie "elementy" muszą istnieć by powstał pierwszy obraz. Zatem musimy:

- Ustawić kamerę (miejsce z którego patrzymy)
- Ustawić źródła światła (aby mieć możliwość oświetlenia obiektów) .Pamiętaj bez światłą możemy nic nie zobaczyć
- Stworzyć obiekt, na który patrzymy

Teraz „opowiem” krok po kroku poprzez poszczególne części kursu, jak używać języka opisu sceny, aby stworzyć swoją własną grafikę. Szczegółowo wyjaśniono tu większość funkcji języka opisu sceny Pov-Ray'a. Nauczymy się prostych elementów, takich jak umiejscawianie kamery (obserwatora) i źródła światła. Nauczymy się również, jak stworzyć bardziej skomplikowane obiekty i jak przypisać im różne tekstury.

Pov-Raya uruchomimy plikiem "pvengine.exe" w katalogu C:\Program Files\Pov Ray for Windows\Bin\
Zaraz po uruchomieniu pojawi się poniższe okno (tu zdjęcie z wersji 3.6) :



Jak widać okno nie posiada żadnych narzędzi edycyjnych. W naszych rękach pozostanie teraz klawiatura oraz przycisk Run.

Pov-Ray posiada wiele udogodnień w postaci bibliotek, które stworzyli twórcy programu. Dzięki nim nie musimy znać opisów matematycznych kolorów i tekstur np. drewna czy metali.
Zamiast stworzyć wszystko od nowa, warto używać polecenia include (pisanego zawsze na początku programu) Powoduje ono wczytanie stworzonych uprzednio elementów z pliku, którego nazwę podajemy po tym poleceniu. Napiszemy więc:

```
#include "colors.inc"
#include "textures.inc"
```

Dzięki pierwszej linijce kodu będziemy mogli korzystać z wcześniej zdefiniowanych kolorów (bez tego Pov-Ray nie będzie rozumiał co oznacza definicja koloru: Red), dzięki drugiej linijce będziemy mogli wypełniać figury teksturą np. drewnianą.

Do właściwej części programu będą nam potrzebne miejsce z którego patrzymy, światło i obiekt.

Obiektyw

Obiektyw możesz umieszczać dowolnym miejscu przestrzeni, ale pamiętajmy aby go wycelować w odpowiednie miejsce (najlepiej tam gdzie ustawimy obiekt ☺)
Napiszmy polecenie:

```
#include "colors.inc"
#include "textures.inc"

camera{ location <0,0,-4>      //wsółrzędne położenia kamery
        look_at <0,0,0>      } //określamy na jaki punkt przestrzeni patrzymy
```

Zwróć uwagę, iż definiując polecenie jego zawartość zawsze będzie umieszczona w nawiasach klamrowych
W celach dydaktycznych umieściłem symbol "//" oznaczający „komentarz programisty”, który nie jest interpretowany przez Pov-ray
Kamerę umieściliśmy w punkcie <0,0,-4> "to punkt z którego patrzymy na monitor przed nami"

Światło

Trzeba jeszcze określić, gdzie znajduje się żarówka (światło). Umówmy się na początku, że będziemy umieszczać światło w tym samym miejscu co kamerę, dzięki temu nie będziemy się martwić o oświetlenie obiektu.
Uwaga:
Kamera musi mieć określony kierunek patrzenia, natomiast światło rozchodzi się równomiernie we wszystkich kierunkach.
Dodajmy światło:

```
#include "colors.inc"
#include "textures.inc"

camera{ location <0,0,-4>      //wsółrzędne położenia kamery
        look_at <0,0,0>      } //określamy na jaki punkt przestrzeni patrzymy

light_source{ <0,0,-4> color White} //tutaj ustalono biały kolor czyli
                                    //światło naturalne.
```

Wygenerowanie obrazu niestety stworzy nam czarny obraz, gdyż brakuje nam obiektu. Światła nie widzimy, dopiero umieszczenie obiektu w przestrzeni spowoduje oświetlenie go.

Pierwszy obiekt - KULA

Najprostszym obiektem jest kula, którą definiują dwie cechy: współrzędne środka i promień

```
#include "colors.inc"
#include "textures.inc"

camera{ location <0,0,-4>      //wsółrzędne położenia kamery
        look_at <0,0,0>      } //określamy na jaki punkt przestrzeni patrzymy

light_source{ <0,0,-4> color White} //tutaj ustalono biały kolor czyli
                                    //światło naturalne.
sphere{ <0,0,0> 1}          //kula o środku w <0,0,0> i promieniu 1
```

„Nadeszła wiekopomna chwila”: uruchomienie naszego rysunku przyciskiem "Run". Program poprosi o zapisanie pliku. Po zapisaniu zacznie generować obraz o rozdzielczości wskazanej w lewym górnym rogu ekranu (u mnie jest to 512x384) Im większy obraz tym większa jakość, im większa jakość tym dłużej generuje się obraz.

Uwaga:

Wielkość liter w Pov-Ray ma istotne znaczenie. Dlatego nazwy kolorów, tektur zaczynamy DUŻĄ LITERĄ

Pov-Ray stworzy śliczną czarną kulkę, gdyż nie nalaliśmy jej żadnej tektury ani koloru.
Dodajmy teraz kolor naszej kulce:

```
sphere{ <0,0,0> 1 texture{pigment{color Red}}}   //kula o środku w <0,0,0> i promieniu 1
```

Zadanie 0 (domowe)

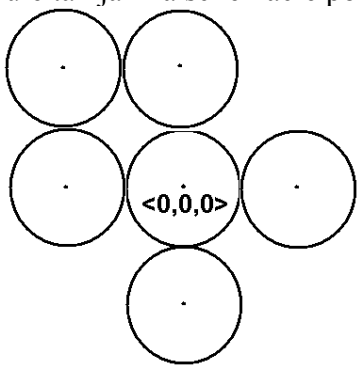
Wejdź na stronę www.povray.org i zobacz efekty pracy programistów Pov-Raya. W przypadku niektórych zdjęć trudno odróżnić je od efektów pracy zwykłego fotografa.

Zadanie 1

Na podstawie powyższej instrukcji napisz program tworzący czerwoną kulę umieszczoną w przestrzeni.

Zadanie 2

Rozszerz zadanie pierwsze o dodatkowe kule tak jak na schemacie poniżej. Przyjmij promień kuli równy 1



Zadanie 3

Sprawdź jakie inne kolory są dostępne w Pov-Ray. Zmień ustawienia kamery i nadaj każdej kulce inny kolor.

UWAGA:

Nauczony doświadczeniem wielu z was robi błąd, gdy np. zobaczy zbyt małą kulkę powiększacie okienko z wygenerowanym obrazem. To bardzo zły nawyk z systemu Windows. W takiej sytuacji wystarczy ustawić kamerę w innym miejscu.

Zadanie 4

Zbuduj piramidę z kulek