

# Grafika 3D

Celem zadania jest utworzenie programu do rysowania trójwymiarowych scen za pomocą różnych API graficznych (tzw. *rendererów*). Do dyspozycji mamy kilka standardowych rendererów oraz zbiór trójwymiarowych obiektów do narysowania. Wszystkie obiekty można podzielić na dwie kategorie: siatkowe (np. odcinek lub siatka) oraz wypełnione (trójkąt, czworobok).

Dla uproszczenia przyjęto, że różne API są w stanie narysować konkretny obiekt na podstawie jego typu i właściwości – czy obiekt jest przezroczysty i w jakim stopniu (obiekty wypełnione) oraz jaka jest grubość rysowanej linii (w przypadku obiektów siatkowych). ***To jakie cechy ma obiekt wnioskujemy po rodzaju obiektu dopiero w momencie gdy trafia ona do konkretnego renderera i ma zostać narysowany.*** Nie wolno tworzyć konstrukcji ręcznego rozpoznawania obiektów, np. za pomocą switch-a.

Gdy wybrany renderer nie potrafi narysować danego obiektu jest on po prostu ignorowany. Programista będzie testował rysowanie różnych obiektów 3D w zależności od wybranego renderera. Scena będzie składała się z listy konkretnych obiektów 3D.

W programie zaimplementowane są już częściowo klasy reprezentujące obiekty: odcinek, siatka, trójkąt i czworobok. Częściowo zaimplementowane są także klasy rendererów które można użyć do rysowania na ekranie: OpenGL, Direct3D, Software. Każdy z obiektów posiada podstawowe właściwości: nazwę i przestrzeń kolorów w której jest rysowany (domyśla dla renderera lub specyficzna, żądana przez programistę). Wszystkie specyficzne właściwości ustalane są podczas definiowania obiektu i są niezmiennie.

Celem zadania jest uzupełnienie klas o dodatkowe metody tak aby możliwa była obsługa rysowania scen zdefiniowanych przez programistę.

Rysowanie obiektu 3D polega na wypisaniu nazwy renderera który go rysuje, nazwy samego obiektu oraz wszystkich jego specyficznych właściwości (stopień przezroczystości lub grubość linii).

## Zadania do wykonania:

1. Programista testuje rysowanie sceny składającej się z listy konkretnych obiektów 3D. Do dyspozycji ma także zbiór rendererów które będzie testowany. Celem jest narysowanie wszystkich obiektów z listy, bez względu na to który renderer będzie to w stanie zrobić. Nie ma znaczenia, że więcej niż jeden renderer narysuje taki sam obiekt ponieważ będą one identyczne i się pokryją. Podczas rysowania renderer wypisuje: <nazwa\_renderera> rysuje <element> o nazwie <nazwa> i <specyficzne\_ustawienia obiektu> (według załączonego przykładu).

2. Programista chce wykonać sprawdzenie poprawności rysowania obiektów przez różne renderery. W tym celu potrzebuje dla każdego obiektu dowiedzieć się ile jest różnych rendererów które są w stanie narysować podany obiekt. Należy zaimplementować wypisywanie, dla każdego obiektu liczby rendererów potrafiących narysować dany obiekt (według załączonego przykładu).

3. Programista chce zbadać ile razy każdy z rendererów narysował jakiś obiekt. Uzupełnić implementację w taki sposób aby zliczać liczbę narysowań obiektów przez konkretny renderer. Wypisać na ekran ile razy każdy z rendererów narysował jakiś obiekt.

### Właściwości rendererów:

#### *Renderer Direct3D*

- \* Direct3D potrafi rysować obiekty siatkowe i wypełnione. Siatkowe nie mogą mieć grubości linii większej niż 1.0.
- \* Wypełnione muszą być w pełni nieprzezroczyste. Wszystkie przestrzenie kolorów są obsługiwane.

#### *Renderer OpenGL*

- \* OpenGL potrafi rysować obiekty siatkowe i wypełnione bez ograniczeń grubości linii i stopnia przezroczystości.
- \* Obsługiwana jest wyłączenie przestrzeni kolorów RGB i domyślna (którą jest właśnie RGB).

#### *Renderer Software*

- \* Software renderer potrafi rysować wyłącznie obiekty siatkowe. Siatkowe nie mogą mieć grubości linii większej niż 3.0.
- \* Wszystkie przestrzenie kolorów są obsługiwane.

### Właściwości obiektów 3D:

odcinek:

- \* siatkowy, nazwa: Segment3D

siatka:

- \* siatkowy, nazwa: WireMesh3D

trójkąt:

- \* wypełniony, nazwa: Triangle3D

czworobok:

- \* wypełniony, nazwa: Quad3D

### Oczekiwany wynik:

==> Rysowanie elementów 3D z użyciem różnych rendererów

Lista elementów do narysowania:

- [1] obiekt 3D 'Triangle3D' w przestrzeni kolorów 'RGB'
- [2] obiekt 3D 'WireMesh3D' w przestrzeni kolorów 'YUV'
- [3] obiekt 3D 'Segment3D' w przestrzeni kolorów 'CMYK'
- [4] obiekt 3D 'Quad3D' w przestrzeni kolorów 'XYZ'
- [5] obiekt 3D 'Triangle3D' w przestrzeni kolorów 'Default'

Software rysuje siatkowy obiekt o nazwie 'WireMesh3D' i grubości linii '2'

OpenGL rysuje wypełniony obiekt o nazwie 'Triangle3D' i przezroczystości '0' %

Direct3D rysuje wypełniony obiekt o nazwie 'Triangle3D' i przezroczystości '0' %

==> Wyznaczanie liczby rendererów na których można narysować konkretne obiekty  
Liczba rendererów mogących narysować Triangle3D: 1  
Liczba rendererów mogących narysować WireMesh3D: 3  
Liczba rendererów mogących narysować Segment3D: 3  
Liczba rendererów mogących narysować Quad3D: 2

==> Ile razy renderer narysował obiekt na ekranie  
Renderer 1 narysował 2 obiektów  
Renderer 2 narysował 4 obiektów  
Renderer 3 narysował 3 obiektów