

# Grafika Komputerowa

## Ćwiczenie 8

### Wizualizacja sceny z interaktywnym ustalaniem obserwatora/kamery

**Wobec występowania niepokojących przypadków korzystania w prezentowanych programach z obcego i nierozumianego kodu po raz kolejny przypominam, że programy należy implementować samodzielnie i ze zrozumieniem metody którą implementujemy. W szczególności dotyczy to wykonywania niezbędnych obliczeń.**

#### Zakres zadania:

Zaprojektuj i zrealizuj program, który wczyta opis sceny 3D i wyświetli go za pomocą OpenGL. W programie zrealizuj prostą funkcjonalność pozycjonowania kamery opierając się na następujących założeniach:

1. Opis sceny znajduje się w pliku w formacie Wavefront obj (zob. np. w [https://en.wikipedia.org/wiki/Wavefront\\_obj\\_file](https://en.wikipedia.org/wiki/Wavefront_obj_file) )
2. Początkowa pozycja i parametry kamery są wczytywane z pliku (format podany dalej).
3. Jeśli brak pliku to ustaw obserwatora tak aby patrzył w kierunku środka domeny<sup>1</sup> wzdłuż kierunku przekątnej prostopadłościanu domeny, tak aby kierunek OY był kierunkiem pionowym w obrazie a kąt widzenia taki aby objął całą domenę (ale nie więcej, tzn. aby prostokąt obrazu był najmniejszym możliwym obejmującym wszystkie geometryczne elementy sceny).
4. Za zaimplementuj następujące zmiany pozycji kamery:
  - przesuwanie równoległe do płaszczyzny obrazu lewo/prawo, góra/dół (panning).
  - spacerowanie - przesuwanie równoległe kamery wzdłuż kierunku patrzenie do przodu/do tyłu
  - orbitowanie (przemieszczanie obserwatora po powierzchni sfery o środku w punkcie na który patrzymy).
  - rozglądanie się - przemieszczanie punktu target przy ustalonej pozycji obserwatora.
5. Sterowanie kamerą zaimplementuj za pomocą klawiatury.

#### Dodatkowe uwagi i wymagania нефunkcjonalne

Operacje manipulacji kamerą które należy zaimplementować wykorzystywane są w większości programów do modelowanie 3D. Aby zapoznać się z ich efektami możesz

---

<sup>1</sup> Domena - najmniejszy prostopadłościan o ścianach prostopadłych do osi układu współrzędnych, który zawiera wszystkie elementy sceny

skorzystać z darmowej wersji jednego z nich (np. bardzo dobry, również w wersji darmowej pCon.planner). Obliczenia związane z wyznaczaniem parametrów kamery (pozycja oka, kierunek patrzenia/punkt target, kąt fov należy w swoim kodzie zaimplementować SAMODZIELNIE. Wskazówki dotyczące przekształceń geometrycznych związanych z takimi manipulacjami kamerą znajdziesz w slajdach do wykładu 12 ( slajdy 24-30).

W ćwiczeniu należy zaimplementować wczytywanie podstawowych danych w pliku w formacie obj i stowarzyszonym z nim pliku w formacie mtl. Plik obj może zawierać wiele różnych informacji, ale na użytek ćwiczenia należy wczytać i wykorzystać tylko:

- pozycje wierzchołków,
- wektory normalne w wierzchołkach (jeśli są zapisane, jeśli nie - wykorzystujemy wektory geometryczne),
- informacje o wielokątach (face) tzn, z których wierzchołków skonstruowany jest wielokąt,
- grupowanie wielokątów w zbiory przypisane tym samym własnościom materiałowym (rekordom w pliku mtl),

Z pliku mtl należy wczytać i wykorzystać wszystkie atrybuty powierzchni które można zinterpretować jako parametry materiału w OpenGL. Jeśli do powierzchni przypisany jest obraz tekstury to NIE NALEŻY GO BEZPOŚREDNIO UŻYWAĆ JAKO TEKSTURY W OpenGL. Zamiast tego należy zliczyć z obrazu tekstury jej średni kolor i zastosować jako parametr *kd*.

Możesz wykorzystać przykładowy kod w C++ który wczytuje dane z pliku obj z jednego z udostępnionych na ePortalu programów demo lub wyjątkowo z innego źródła obcego (**UWAGA: dotyczy to tylko samego wczytywania pliku**)

Format obj nie obejmuje pozycji świateł ani kamery. Światła musisz rozmieścić automatycznie przy ładowaniu sceny. Zaproponuj taką metodę automatycznego umieszczania świateł aby w większości przypadków zapewniały oświetlenie sceny pozwalające na jej oglądanie.

Założ że parametry obserwatora zapisane są w dodatkowym pliku (z rozszerzeniem cam) w następującej postaci (postać dostosowana do wizualizacji metodą RT):

- pozycja obserwatora w przestrzeni sceny (x y z )
- pozycja punktu target (x y z )
- rozdzielczość obrazu (xres yres )
- pozycja górnego lewego wierzchołka obrazu z układzie sceny
- wektor dU odpowiadający przesunięciu w przestrzeni sceny pomiędzy dwoma sąsiednimi pikselami tego samego wiersza obrazu (x y z )
- wektor dV odpowiadający przesunięciu w przestrzeni sceny pomiędzy dwoma sąsiednimi pikselami tej samej kolumny (x y z )

Pozostałe liczby z pliku należy zignorować.

Należy opracować i zaimplementować metod, która wyliczy na podstawie wczytanych parametry kamery w postaci używanej przez funkcje w OpenGL/glu.

Przykładowe sceny w obj/mtl będą udostępnione na ePortalu. Możesz również samodzielnie przygotować sobie sceny do testowania korzystając z jednego z wielu programów które umożliwiają eksport sceny do obj (np. Blender lub wspomniany już pCon.planner).