

# **Laboratorium grafiki komputerowej**

## **Ćwiczenie nr 3 - Interaktywna aplikacja z elementami grafiki komputerowej do definiowania parametrów kamery dla sceny 3D z analizą widoczności powierzchni obiektów, cieniowaniem Gouraud'a oraz Phong'a**

Zaawansowane aplikacje graficzne wymagają wprowadzenia wielu danych opisujących scenę będącą przedmiotem przetwarzania. Bardzo wygodną i naturalną techniką określania takich złożonych danych jest technika interakcyjna, gdzie użytkownik manipuluje elementami sceny bezpośrednio na jej widoku na ekranie. Może on więc np. dodawać nowe bryły do sceny, przemieszczać już istniejące elementy sceny oraz definiować ich parametry, np. parametry powierzchniowe. Implementując interfejs aplikacji w takiej konwencji należy zapewnić jego naturalność. Powszechnie stosowaną w tym zakresie techniką jest technika zwana „przesuń-i-upuść”. Oznacza to, że operacje użytkownika w wirtualnej scenie wyświetlonej na ekranie muszą odpowiadać operacjom, jakie wykonywałby on mając do czynienia z realną sceną. Jeśli np. chcemy przesunąć w scenie pewną jej część to na wyświetlonym widoku sceny przyciskamy klawisz myszy, gdy kursor znajduje się nad tą częścią (chwytamy obiekt) a następnie z przyciśniętym klawiszem myszy przesuwamy kursor po obrazie do miejsca gdzie ma znaleźć się ta część (przesuwamy część w scenie). Gdy kursor osiągnie miejsce, w którym ostatecznie ma się znaleźć ta część zwalniamy klawisz myszy (upuszczamy część). Program nadzorujący tę interakcję wylicza odpowiednio nową pozycję części w scenie i tam ją umieszcza.

W niniejszym ćwiczeniu należy zaprojektować i zaimplementować aplikację o interfejsie w takim właśnie stylu. Ze względu na duży stopień złożoności operacji interaktywnych ograniczymy się jedynie do manipulowania parametrami kamery.

### **Cele ćwiczenia:**

- opanowanie podstawowych technik programowania interakcji pomiędzy użytkownikiem a modelem sceny,
- praktyczne zapoznanie z elementami API niezbędnymi do implementacji tego typu aplikacji a w szczególności z mechanizmem obsługi zdarzeń związanych z urządzeniem wskazującym (myszą),
- zapoznanie z podstawowymi elementami potoku wizualizacji w zakresie niezbędnym do uzyskania prostego widoku sceny.

### **Specyfikacja zadania do wykonania:**

Należy zaimplementować aplikację, która dla zadanej sceny pozwoli w naturalny i wygodny sposób określić parametry kamery.

### **Wymagane funkcje programu są następujące:**

- wczytywanie modelu geometrii w reprezentacji brzegowej z pliku,
- wczytywanie początkowych ustawień kamery z tego samego pliku
- automatyczne ustalenie rozmiarów i położenia domeny i określanie zakresu widoków ortogonalnych

- wyświetlanie sceny w widokach ortogonalnych i perspektywnym,
- natychmiastowe odświeżanie widoku perspektywnego po każdorazowej zmianie jakiegokolwiek parametru kamery,
- interaktywna manipulacja parametrami kamery,
- zapis ustawionych parametrów kamery w pliku w takim samym formacie jak na wejściu.

Typowy sposób korzystania z programu jest następujący:

- użytkownik uruchamia aplikację, scena jeszcze nie jest załadowana,
- użytkownik wprowadza nazwę i lokalizację pliku opisu sceny (wykorzystując odpowiednio zaprojektowany element interfejsu aktywowany przyciśnięciem przycisku funkcyjnego),
- program ładuje wskazaną scenę i wyświetla ją w widokach ortogonalnych i widoku perspektywnym,
- użytkownik manipuluje parametrami kamery tak aby uzyskać odpowiedni widok perspektywny (np. tak aby w polu widzenia kamery znalazła się określona część sceny), w tym celu przesuwa odpowiednio punkty oznaczające środek kamery i środek obrazu na widokach ortogonalnych i obserwuje aktualny widok perspektywny,
- po ustaleniu wymaganych parametrów widoku użytkownik zapisuje dane w pliku tekstowym, operacja zapisu wywoływana jest przez przyciśnięcie odpowiedniego przycisku.
- po ponownym wczytaniu uprzednio zapisanych parametrów kamery (np. po kolejnym uruchomieniu programu) uzyskujemy taki sam widok perspektywny jak przy zapisywaniu sceny.

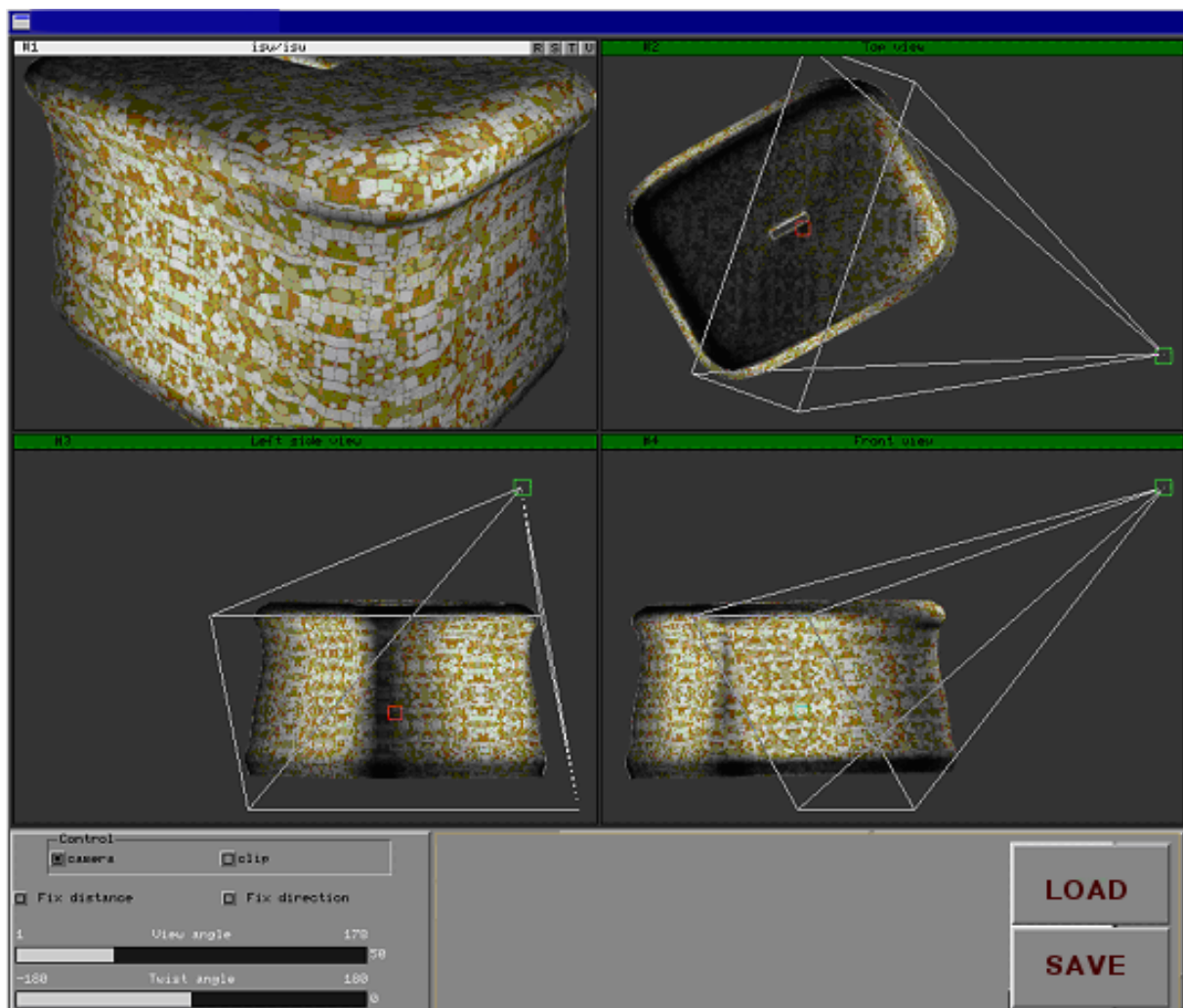
### Wymagania szczegółowe

- Stosujemy brzegową trójkątową reprezentację geometrii sceny.
- Pozycja kamery opisana jest przez podanie:
  - o punktu położenie kamery  $P_o$ ,
  - o punktu środka obrazu  $P_c$ ,
  - o kąta  $\alpha$  rozwarcia piramidy widzenia, tzn. kąta pomiędzy osią kamery a płaszczyzną ograniczająca piramidę widzenia od strony lewej (lub prawej – kąty te są identyczne)
 Dla uproszczenia zakładamy, że kierunek pionowy na obrazie perspektywnym to kierunek w układzie sceny, leżący w płaszczyźnie wyznaczonej przez oś OY i punkt  $P_o$ .
- Scenę w widokach ortogonalnych i perspektywnym należy wyświetlać wykorzystując zaimplementowany programowo algorytm bufora głębokości. Należy prawidłowo obsłużyć problem przycinania (oraz ewentualnej eliminacji) wyświetlanych trójkątów, w sytuacji kiedy częściowo (lub w całości) znajdują się poza obszarem widzenia.
- Trójkąty należy cieniować wyznaczając postrzegane barwy ich wierzchołków stosując model oświetlenia Phong'a oraz cieniowanie Gouraud'a oraz cieniowanie Phong'a.
- Zakładamy, że scena oświetlona jest jednym światłem punktowym znajdującym się w punkcie określonym w pliku opisu sceny.
- Kontur piramidy widzenia należy wyświetlić rysując na widokach perspektywnych krawędzie boczne stożka piramidy widzenia oraz prostokąt jego podstawy. Wierzchołki tego prostokąta należy wyznaczyć jako przecięcia krawędzi bocznych z płaszczyzną prostopadłą do wektora  $P_c - P_o$ .
- Manipulowanie położeniem punktu obserwatora i punktu określającego kierunek patrzenia ma odbywać się techniką *przeciągnij-i-upuść* (ang. *drag and drop*) na widokach

ortogonalnych. Dodatkowo należy stworzyć możliwość określania położenia tych punktów w sposób jawny przez wprowadzenie ich współrzędnych w polach edycyjnych. Wartości w tych polach powinny być modyfikowane automatycznie przy przeciąganiu punktów na widokach ortogonalnych.

- Kąt rozwarcia piramidy widzenia powinien być kontrolowany za pomocą sladera o zakresie 1 - 89°.
- Wczytywanie sceny i jej zapis po modyfikacji położenia kamery należy zaimplementować jako funkcje wykonywane w reakcji na przyciśnięcie odpowiednich przycisków: LOAD i SAVE.

Przykładowy widok interfejsu aplikacji realizującej podobną funkcjonalność przedstawiono poniżej:



#### UWAGA:

Jest to tylko przykład pokazujący sposób prezentacji sceny w widokach ortogonalnych i perspektywicznym. W pokazanym oknie programu występuje więcej elementów manipulacyjnych niż w programie, który należy wykonać.

#### Format opisu sceny

Specyfikacja sceny jest zapisana w pliku tekstowym składającym się z czterech sekcji:

- sekcji wierzchołków,

- sekcji trójkątów,
- sekcji części,
- specyfikacji źródła światła

Sekcja wierzchołków zawiera współrzędne  $x, y, z$  wierzchołków siatki trójkątowej geometrii sceny. Sekcja trójkątów zawiera indeksy wierzchołków kolejnych trójkątów sceny. Sekcja części zawiera indeksy części dla kolejnych trójkątów sceny oraz parametry powierzchniowe dla kolejnych części.

Układ danych w pliku opisu sceny jest następujący

```

n                                // liczba wierzchołków
x0 y0 z0                        // współrzędne pierwszego wierzchołka
x1 y1 z1                        // współrzędne drugiego wierzchołka
. . .
xn-1 yn-1 zn-1 // współrzędne n-tego wierzchołka

m                                // liczba trójkątów
i0 j0 k0                        // indeksy wierzchołków pierwszego trójkąta
i1 j1 k1                        // indeksy wierzchołków drugiego trójkąta
. . .
im-1 jm-1 km-1 // indeksy wierzchołków m-tego trójkąta

p0                                // indeks części pierwszego trójkąta
p1                                // indeks części drugiego trójkąta
. . .
pm-1                            // indeks części m-tego trójkąta

k                                // liczba części
R0 G0 B0 kd0 ks0 g0        // parametry powierzchniowe pierwszej części
R1 G1 B1 kd1 ks1 g1        // parametry powierzchniowe drugiej części
...
Rk-1 Gk-1 Bk-1 kdk-1 ksk-1 gk-1 // parametry powierzchniowe ostatniej części

x y z R G B                      // pozycja i barwa światła punkowego oświetlającego scenę

```

Parametry kamery zapisane są w osobnym pliku tekstowym zawierającym następujące dane:

```

x0 y0 z0        // położenie środka kamery
xc yc zc        // położenie punktu środka obrazu
a                // kat rozwarcia piramidy widzenia w stopniach

```

Zakładamy, że plik opisu geometrii sceny ma rozszerzenie `brp` a plik opisu kamery – rozszerzenie `cam`. Obydwa pliki mają tę samą nazwę podstawową i znajdują się w tym samym folderze.

## Elementy implementacji

Realizacja ćwiczenia wymaga zaprojektowania i implementacji komponentów realizujących następujące zadania:

- wczytanie opisu sceny i utworzenie odpowiedniej wewnętrznej jej reprezentacji,
- zapis aktualnych parametrów kamery w pliku tekstowym,
- wyznaczenie obszaru domeny sceny,

- wyznaczenie macierzy rzutowania perspektywicznego dla aktualnych parametrów kamery,
- wyliczenie kolorów w wierzchołkach siatki reprezentacji brzegowej,
- wyznaczenie macierzy rzutowania równoległego dla rzutów ortogonalnych
- wyświetlenie widoku perspektywicznego dla aktualnej macierzy rzutowania
- wyświetlenie widoków ortogonalnych sceny,
- wyznaczenie wierzchołków ostrosłupa widzenia w przestrzeni sceny,
- wyświetlenie konturów ostrosłupa widzenia na rzutach ortogonalnych.

### **Kryteria oceny zadania:**

- spełnienie wymagań określonych w specyfikacji zadania,
- ergonomia, estetyka i wygoda użytkowania,
- poprawność modularyzacji kodu,
- czytelność i estetyka kodu,
- poprawność wykorzystania elementów API w zakresie obsługi zdarzeń związanych z urządzeniem wskazującym.