



POLITECHNIKA WROCŁAWSKA
Instytut Informatyki, Automatyki i Robotyki
Zakład Systemów Komputerowych

Wprowadzenie do grafiki komputerowej

Kurs: INE4234L

Sprawozdanie z ćwiczenia nr 4

TEMAT ĆWICZENIA OpenGL - interakcja z użytkownikiem

Wykonał:	Bartłomiej Sawicki
Termin:	PN/P 7.30-10.30
Data wykonania ćwiczenia:	8.11.2021r.
Data oddania sprawozdania:	16.11.2021r.
Ocena:	

Uwagi prowadzącego:

1 Wstęp teoretyczny

Celem laboratorium było zapoznanie się ze sposobami interakcji pomiędzy użytkownikiem, a obiektami graficznymi biblioteki OpenGL. Zachowania elementów widocznych na ekranie miały być modyfikowane przez użytkownika przy pomocy myszy. Zadanie składa się z dwóch części. W pierwszej należało napisać program umożliwiający użytkownikowi zmianę położenia rysowanego obiektu. W drugiej części należało natomiast zmieniać położenie obserwatora.

2 Realizacja zadania

Szkielet programu bazuje na kodzie zawartym w instrukcji laboratoryjnej. Funkcja odpowiedzialna za rysowanie jajka jest wywoływana w procedurze `RenderScene`. Funkcje odpowiedzialne za reakcje programu gdy naciśnięte są przyciski pyszki lub gdy mysz się porusza to odpowiednio `Mouse()` oraz `Motion()`. Procedury te są wywoływane z funkcji `main()` przy pomocy `glutMouseFunc()` oraz `glutMotionFunc()`.

2.1 Obsługa przysisków myszy

Funkcja `Mouse()` sprawdza stan przycisków oraz pozycję kursora w momencie naciśnięcia któregoś z nich. Jeżeli przycisk jest wciśnięty pobierana jest aktualna pozycja kursora, która następnie jest przechowywana w zmiennych `x_pos_old`, `y_pos_old` oraz zmieniana jest wartość zmiennej `status` przechowującej informacje, o tym który przycisk jest wciśnięty.

```

/*****
// Funkcja "bada" stan myszy i ustawia wartości odpowiednich zmiennych globalnych
void Mouse(int btn, int state, int x, int y)
{
    if (btn == GLUT_LEFT_BUTTON && state == GLUT_DOWN)
    {
        x_pos_old = x;        // przypisanie aktualnie odczytanej pozycji kursora
                               // jako pozycji poprzedniej
        y_pos_old = y;

        status = 1;           // wcinięty został lewy klawisz myszy
    }
    else if (btn == GLUT_RIGHT_BUTTON && state == GLUT_DOWN)
    {
        y_pos_old = y;

        status = 2;
    }
    else
    {
        status = 0;           // nie został wcinięty żaden klawisz
    }
}
}

```

2.2 Obsługa ruchu myszy

Funkcja *Motion()* oblicza przemieszczenie kursora.

```

/*****
// Funkcja "monitoruje" położenie kursora myszy i ustawia wartości odpowiednich
// zmiennych globalnych
void Motion(GLsizei x, GLsizei y)
{
    delta_x = x - x_pos_old;    // obliczenie różnicy położenia kursora myszy
    delta_y = y - y_pos_old;

    x_pos_old = x;              // podstawienie bieżącego położenia jako poprzednie
    y_pos_old = y;

    glutPostRedisplay();        // przerysowanie obrazu sceny
}

```

2.3 Poruszanie kamerą

Poruszanie kamerą odbywa się poprzez modyfikację położenia obserwatora w zależności od aktualnego kąta.

$$\begin{aligned}
x_s(\Theta, \Phi) &= R \cos(\Theta) \cos(\Phi) \\
y_s(\Theta, \Phi) &= R \sin(\Phi) \\
z_s(\Theta, \Phi) &= R \sin(\Theta) \cos(\Phi)
\end{aligned}
\qquad
\begin{aligned}
0 \leq \Theta &\leq 2\pi \\
0 \leq \Phi &\leq 2\pi
\end{aligned}$$

Rysunek 1: Funkcje matematyczne opisujące położenie obserwatora w zależności od kąta

```

if (camera == 2)
{
    if (status == 1) // jeśli lewy klawisz myszy wciśnięty
    {
        theta += delta_x * pix2angle / 100; // modyfikacja kąta obrotu o kat proporcjonalny
        // do różnicy położenia kursora myszy

        phi += delta_y * pix2angle / 100;

        if (phi > pi / 2)
        {
            phi = pi / 2;
        }
        if (phi < -pi / 2)
        {
            phi = -pi / 2;
        }
    }
    if (status == 2)
    {
        radius += delta_y * pix2angle;
        if (radius > 30.0)
        {
            radius = 30.0;
        }
        if (radius < 1.0)
        {
            radius = 1.0;
        }
    }

    viewer[0] = radius * cos(theta) * cos(phi);
    viewer[1] = radius * sin(phi);
    viewer[2] = radius * sin(theta) * cos(phi);
}

```

2.4 Obracanie obiektu

Obracanie obiektu następuje przy pomocy funkcji *glRotatef()*. Kąt obrotu jest obliczany na podstawie zmiany pozycji kursora.

```

if (camera == 1)
{
    if (status == 1)                // jeśli lewy klawisz myszy wciśnięty
    {
        theta += delta_x * pix2angle;
        phi += delta_y * pix2angle;
    }
    glRotatef(theta, 0.0, 1.0, 0.0);
    glRotatef(phi, 1.0, 0.0, 0.0);
    if (status == 2)
    {
        viewer[2] += delta_y;
        if (viewer[2] > 30)
        {
            viewer[2] = 30;
        }
        if (viewer[2] < 1)
        {
            viewer[2] = 1;
        }
    }
}

```

2.5 Sterowanie wyświetlanymi obiektami

Sterowanie wyświetlanym obrazem i metodą poruszania kamerą odbywa się przez wciśnięcie odpowiedniego klawisza na klawiaturze

```

p - punkty
w - siatka
s - trojkaty
o - poruszanie obiektem
c - poruszanie kamera

a - osie układu
t - czajnik
e - jajko

```

3 Wnioski

Instrukcja laboratoryjna była głównym źródłem informacji o interakcji użytkownika z programem. Rysowane obiekty poruszają się poprawnie. Ruch kamery jest poprawny. Dzięki "menu" można zmienić wyświetlany obiekt lub tryb obracania.