

Programowanie Grafiki Komputerowej

Laboratorium 1

OpenGL

– rysowanie obiektów 3D, animacje, tekstury

Opracowanie:
Krzysztof Skabek
Politechnika Krakowska, 2012

Opracowano na podstawie podręcznika:
Angel: Interactive Computer Graphics, Addison-Wesley 2009

Cel ćwiczenia:

Zapoznanie się ze strukturami oraz metodami biblioteki OpenGL oraz mechanizmami biblioteki Glut.

Zagadnienia szczegółowe:

1. Rysowanie obiektów, modyfikacje macierzy transformacji, proste animacje
2. Przykład 1: Animacja sześcianu
3. Zadanie 1: Modyfikacja obiektu
4. Zadanie 2: Modyfikacja macierzy transformacji
5. Zadanie 3: Modyfikacja animacji
6. Zadanie 4: Rysowanie tekstur

Przykład 1: Sześcian

Rysowanie sześcianu i animacja wokół zadanej osi.

Kod źródłowy programu:

```
#include <GL/glut.h>

GLfloat vertices[][3] = {{{-1.0,-1.0,-1.0},{1.0,-1.0,-1.0},
{1.0,1.0,-1.0}, {-1.0,1.0,-1.0}, {-1.0,-1.0,1.0},
{1.0,-1.0,1.0}, {1.0,1.0,1.0}, {-1.0,1.0,1.0}}};

GLfloat normals[][3] = {{{-1.0,-1.0,-1.0},{1.0,-1.0,-1.0},
{1.0,1.0,-1.0}, {-1.0,1.0,-1.0}, {-1.0,-1.0,1.0},
{1.0,-1.0,1.0}, {1.0,1.0,1.0}, {-1.0,1.0,1.0}}};

GLfloat colors[][3] = {{ {0.0,0.0,0.0},{1.0,0.0,0.0},
{1.0,1.0,0.0}, {0.0,1.0,0.0}, {0.0,0.0,1.0},
{1.0,0.0,1.0}, {1.0,1.0,1.0}, {0.0,1.0,1.0}}};
```

```

void polygon(int a, int b, int c , int d)
{
/* rysowanie wielościanu na podstawie listy wierzchołków */

    glBegin(GL_POLYGON);
        glColor3fv(colors[a]);
        glNormal3fv(normals[a]);
        glVertex3fv(vertices[a]);
        glColor3fv(colors[b]);
        glNormal3fv(normals[b]);
        glVertex3fv(vertices[b]);
        glColor3fv(colors[c]);
        glNormal3fv(normals[c]);
        glVertex3fv(vertices[c]);
        glColor3fv(colors[d]);
        glNormal3fv(normals[d]);
        glVertex3fv(vertices[d]);
    glEnd();
}

void colorcube(void)
{
/* odwzorowanie wierzchołków na ściany sześcianu */

    polygon(0,3,2,1);
    polygon(2,3,7,6);
    polygon(0,4,7,3);
    polygon(1,2,6,5);
    polygon(4,5,6,7);
    polygon(0,1,5,4);
}

static GLfloat theta[] = {0.0,0.0,0.0};
static GLint axis = 2;

void display(void)
{
/* funkcja wyświetlania, czyści bufor okna i bufor głębi,
   braca sześcian, rysuje i podmienia bufory */

    glClear(GL_COLOR_BUFFER_BIT | GL_DEPTH_BUFFER_BIT);
    glLoadIdentity();
    glRotatef(theta[0], 1.0, 0.0, 0.0);
    glRotatef(theta[1], 0.0, 1.0, 0.0);
    glRotatef(theta[2], 0.0, 0.0, 1.0);

    colorcube();

    glFlush();
    glutSwapBuffers();
}

```

```

void spinCube()
{
    /* funkcja Idle, realizowana w czasie bezczynności
       obraca sześciian o 2 stopnie wokół zadanej osi */

    theta[axis] += 2.0;
    if( theta[axis] > 360.0 ) theta[axis] -= 360.0;
    /* display(); */
    glutPostRedisplay();
}

void mouse(int btn, int state, int x, int y)
{
    /* obsługa myszki, wybiera oś rotacji sześciianu */

    if(btn==GLUT_LEFT_BUTTON && state == GLUT_DOWN) axis = 0;
    if(btn==GLUT_MIDDLE_BUTTON && state == GLUT_DOWN) axis = 1;
    if(btn==GLUT_RIGHT_BUTTON && state == GLUT_DOWN) axis = 2;
}

void myReshape(int w, int h)
{
    glViewport(0, 0, w, h);
    glMatrixMode(GL_PROJECTION);
    glLoadIdentity();
    if (w <= h)
        glOrtho(-2.0, 2.0, -2.0 * (GLfloat) h / (GLfloat) w,
                2.0 * (GLfloat) h / (GLfloat) w, -10.0, 10.0);
    else
        glOrtho(-2.0 * (GLfloat) w / (GLfloat) h,
                2.0 * (GLfloat) w / (GLfloat) h, -2.0, 2.0, -10.0, 10.0);
    glMatrixMode(GL_MODELVIEW);
}

void main(int argc, char **argv)
{
    glutInit(&argc, argv);

    /* przyjmujemy podwójne buforowanie oraz bufor głębokości */

    glutInitDisplayMode(GLUT_DOUBLE | GLUT_RGB | GLUT_DEPTH);
    glutInitWindowSize(500, 500);
    glutCreateWindow("Sześciian");
    glutReshapeFunc(myReshape);
    glutDisplayFunc(display);
    glutIdleFunc(spinCube);
    glutMouseFunc(mouse);
    glEnable(GL_DEPTH_TEST);
    glutMainLoop();
}

```

Zadania do wykonania:

Zadanie 1

Na podstawie przykładu 1 dokonaj modyfikacji wyświetlanego obiektu w zakresie kształtu oraz kolorystyki:

- wyświetlanie sześcianu złożonego z linii (reprezentacja szkieletowa),
- zaprojektuj i wyświetl obiekt ośmiościenny lub o większej liczbie ścian,
- zmodyfikuj kolorystykę wyświetlanych ścian,

Zadanie 2:

Zapoznaj się z wpływem modyfikacji macierzy transformacji w trybach:

- `glMatrixMode(GL_PROJECTION)`
- `glMatrixMode(GL_MODELVIEW)`

za pomocą instrukcji:

- `glRotatef(theta, vx, vy, vz)`
- `glTranslatef(dx, dy, dz)`
- `glScalef(sx, sy, sz)`

Zadanie 3:

Na podstawie przykładu 1 napisz program do animacji obracającego się sześcianu po okręgu o zadany promieniu leżącego na płaszczyźnie podłoża.

Zadanie 4:

Na podstawie przykładu 1 napisz program rysujący animowany sześciąn o teksturze szachownicy.

Wskazówki:

1. zadeklaruj strukturę tekstury: `Glubyte my_texels[512][512];`
2. wypełnij tablicę tekstury wzorcem reprezentującym szachownicę o zadanej szerokości kraty, np. 3
3. w ustawieniach OpenGL zezwól na mapowanie tekstury: `glEnable(GL_TEXTURE_2D)`
4. zdefiniuj obraz tekstury za pomocą instrukcji: `glTexImage2D(target, level, components, w, h, border, format, type, texels);`
np. `glTexImage2D(GL_TEXTURE_2D, 0, 3, 512, 512, 0, GL_RGB, GL_UNSIGNED_BYTE, my_texels);`
5. Do każdego wierzchołka dodaj mapowanie tekstury za pomocą instrukcji:
 `glTexCoord2f(s0, t0);`