

SPRAWOZDANIE

Zajęcia: Grafika Komputerowa

Prowadzący: mgr inż. Mikołaj Grygiel

Laboratorium: 7

Data: 09.04.2025

Temat: " Światło i materiały w OpenGL "

Wariant: 14

Illia Bryka,
Informatyka I stopień,
stacjonarne,
4 semestr,
Gr.1a

Zadanie 1

1. Polecenie:

Celem jest stworzenie piramidy z użyciem różnych materiałów określonych wariantem zadania i umieszczenie jej na „podstawie”. Użytkownik może obracać podstawę wokół osi Y, przeciągając mysz w poziomie. Scena wykorzystuje globalne światło otoczenia (ambient) oraz źródło światła o kształcie kuli z możliwością animacji obrotu wokół piramidy. Podstawą piramidy jest wielokąt o 14 wierzchołkach.

2. Wprowadzane dane:

Do zadania został wykorzystany wariant otrzymany od prowadzącego (14-kąt).

3. Wykorzystane komendy:

Do wykonania zadania należało zmodyfikować kod:

```
public void display(GLAutoDrawable drawable) {

    GL2 gl2 = drawable.getGL().getGL2(); // The object that contains all the OpenGL methods.

    gl2.glClear(GL2.GL_COLOR_BUFFER_BIT | GL2.GL_DEPTH_BUFFER_BIT);

    gl2.glLoadIdentity();
    glu.gluLookAt(0, 8, 40, 0, 1, 0, 0, 1, 0); // viewing transform

    gl2.glRotated(rotateY, 0, 1, 0); // modeling transform: rotation of the scene about y-axis

    float[] gray = {0.6f, 0.6f, 0.6f, 1};
    float[] zero = {0, 0, 0, 1};
    gl2.glMaterialfv(GL2.GL_FRONT_AND_BACK, GL2.GL_AMBIENT_AND_DIFFUSE, gray, 0);
    gl2.glMaterialfv(GL2.GL_FRONT_AND_BACK, GL2.GL_SPECULAR, zero, 0);
    gl2.glPushMatrix();
    gl2.glTranslated(0, -1.5, 0); // Move top of stage down to y = 0
    gl2.glScaled(1, 0.05, 1); // Stage will be one unit thick,
    glut.glutSolidCube(20);
    gl2.glPopMatrix();

    int numOfVertices = 13;
    double angle = 2 * Math.PI / numOfVertices;
    int radius = 6;
    float x = 0;
    float z = 0;

    for (var i = 0; i < 8; i += 4)
        gl2.glMaterialfv(GL2.GL_FRONT_AND_BACK, GL2.GL_AMBIENT, materials[12], i);

    gl2.glMaterialf(GL2.GL_FRONT_AND_BACK, GL2.GL_SHININESS, materials[12][12]);
```

```
for (int i = 1; i <= numOfVertices; i++) {
    gl2.glBegin(GL2.GL_POLYGON);

    x = (float) (radius * Math.cos(angle * (i - 1)));
    z = (float) (radius * Math.sin(angle * (i - 1)));

    float x0 = (float) (radius * Math.cos(angle * i));
    float z0 = (float) (radius * Math.sin(angle * i));

    float[] U = {x - 0, -2.0f - 8, z - 0};
    float[] V = {x - x0, 0, z - z0};

    gl2.glNormal3f(
        U[1] * V[2] - U[2] * V[1],
        U[2] * V[0] - U[0] * V[2],
        U[0] * V[1] - U[1] * V[0]
    );
    gl2.glVertex3f(x, -2.0f, z);
    gl2.glVertex3f(0, 8, 0);
    gl2.glVertex3f(x0, -2.0f, z0);

    gl2.glEnd();
}
```

```

public void init(GLAutoDrawable drawable) {
    GL2 gl2 = drawable.getGL().getGL2();
    gl2.glMatrixMode(GL2.GL_PROJECTION);
    gl2.glLoadIdentity();
    glu.gluPerspective(20, (double) getWidth() / getHeight(), 1, 100);
    gl2.glMatrixMode(GL2.GL_MODELVIEW);
    gl2.glEnable(GL2.GL_DEPTH_TEST);
    gl2.glEnable(GL2.GL_NORMALIZE);
    gl2.glEnable(GL2.GL_LIGHTING);
    gl2.glEnable(GL2.GL_LIGHT0);
    // TODO configure better lighting!

    float[] blue1 = {0.4f, 0.4f, 0.4f, 1};
    float[] blue2 = {0, 0, 0.8f, 1};
    float[] blue3 = {0.3f, 0.15f, 0.15f, 1};

    gl2.glLightfv (gl2.GL_LIGHT1, gl2.GL_DIFFUSE, blue1,0);
    gl2.glLightfv (gl2.GL_LIGHT1, gl2.GL_SPECULAR, blue2,0);
    gl2.glLightfv (gl2.GL_LIGHT1, gl2.GL_AMBIENT, blue3,0);

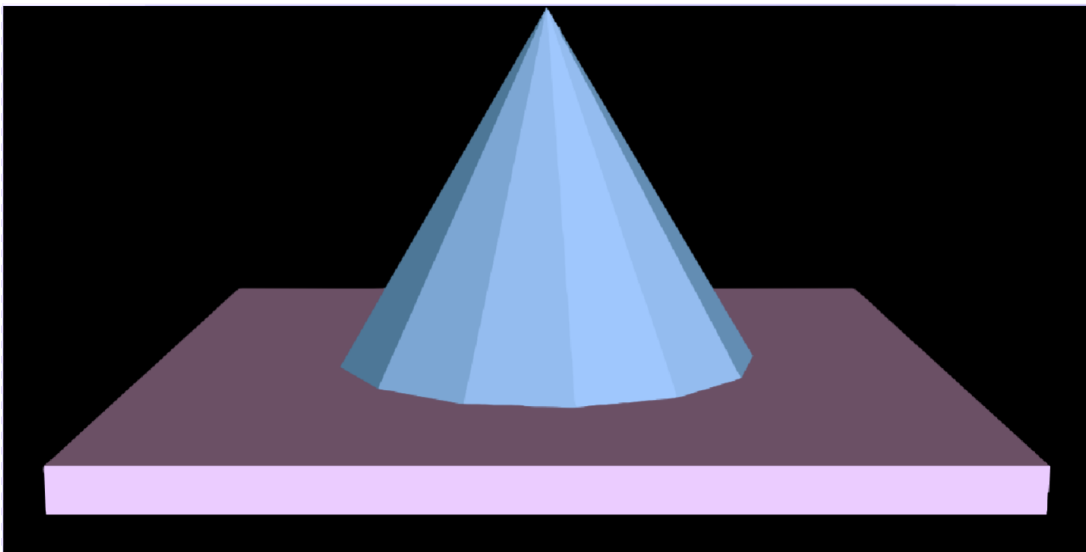
    float[] position = {1,0,0,0};
    gl2.glLightfv (gl2.GL_LIGHT1, GLLightingFunc.GL_POSITION, position, 0);
    gl2.glEnable(GL2.GL_LIGHT1);
}

```

Link do Repozytorium:

<https://github.com/bebrabimba/Grafika-Komputerowa/tree/main/Lab7>

4. Wynik działania:



Wnioski:

Biblioteka OpenGL poza tworzeniem obiektów umożliwia też zarządzanie niektórymi właściwościami jak np. kolor bryły. Dodatkowo można także zmienić światło, kąt jego padania, czy też jasność. Dzięki tym elementom można spowodować, że niektóre elementy mogą być bardziej uwidocznione, co daje możliwość sterowania tym na czym skupia się użytkownik.