

SPRAWOZDANIE 6

Zajęcia: Grafika komputerowa

Prowadzący:

prof. dr hab. Vasyl Martsenyuk

Laboratorium: Grafika Komputerowa 27.03.2020

Temat: Światło i materiały

Sebastian Pierog

Informatyka I stopień, stacjonarne, 4 semestr, Gr.1b

Polecenie:

Celem jest stworzenie piramidy z użyciem różnych materiałów i umieszczenie jej na „podstawie”.

Użytkownik może obracać podstawę wokół osi Y, przeciągając mysz w poziomie. Scena wykorzystuje oświetlenie. Początkowo włączone jest tylko podstawowe oświetlenie. W ramach laboratorium będziesz musiał poprawić oświetlenie.

Możesz wykonać laboratorium w Javie lub C. Aby wykonać laboratorium w Javie, potrzebujesz plików Lab6.java. Aby wykonać laboratorium w C, potrzebujesz plik lab6.c.

Kod źródłowy:

```

gl2.glPushMatrix();
float[] gray = { 0.6f, 0.6f, 0.6f, 1 };
gl2.glPushMatrix();
gl2.glRotatef( (float) 360/11, v1: 0, v2: 1, v3: 0);
gl2.glMaterialfv(GL2.GL_FRONT_AND_BACK, GL2.GL_AMBIENT_AND_DIFFUSE, new float[] {materials[7][0], materials[10][1], materials[10][2], materials[10][3], materials[11][4], materials[11][5], materials[11][6], materials[11][7], materials[11][8], materials[11][9] });
gl2.glMaterialfv(GL2.GL_FRONT_AND_BACK, GL2.GL_SPECULAR, new float[] {materials[10][6], materials[10][7], materials[10][8]}, (float) 0);
gl2.glMaterialf(GL2.GL_FRONT_AND_BACK, GL2.GL_SHININESS, (float) 90);
scianka(gl2, xPoints[0], yPoints[0], a2: 0, b2: 0, xPoints[1], yPoints[1], h);
gl2.glPopMatrix();
gl2.glPushMatrix();
gl2.glRotatef( (float) 360/11, v1: 0, v2: 1, v3: 0);
gl2.glMaterialfv(GL2.GL_FRONT_AND_BACK, GL2.GL_AMBIENT_AND_DIFFUSE, new float[] {materials[6][0], materials[4][1], materials[0][2], materials[4][3], materials[9][4], materials[3][5], materials[3][6], materials[3][7], materials[3][8], materials[3][9] });
gl2.glMaterialfv(GL2.GL_FRONT_AND_BACK, GL2.GL_SPECULAR, new float[] {materials[11][6], materials[4][7], materials[4][8]}, (float) 0);
gl2.glMaterialf(GL2.GL_FRONT_AND_BACK, GL2.GL_SHININESS, (float) 50);
scianka(gl2, xPoints[1], yPoints[1], a2: 0, b2: 0, xPoints[2], yPoints[2], h);
gl2.glPopMatrix();
gl2.glPushMatrix();
gl2.glRotatef( (float) 360/11, v1: 0, v2: 1, v3: 0);
gl2.glMaterialfv(GL2.GL_FRONT_AND_BACK, GL2.GL_AMBIENT_AND_DIFFUSE, new float[] {materials[13][0], materials[13][1], materials[13][2], materials[13][3], materials[13][4], materials[13][5], materials[13][6], materials[13][7], materials[13][8], materials[13][9] });
gl2.glMaterialfv(GL2.GL_FRONT_AND_BACK, GL2.GL_SPECULAR, new float[] {materials[11][6], materials[13][7], materials[13][8]}, (float) 0);
gl2.glMaterialf(GL2.GL_FRONT_AND_BACK, GL2.GL_SHININESS, (float) 80);
scianka(gl2, xPoints[2], yPoints[2], a2: 0, b2: 0, xPoints[3], yPoints[3], h);
gl2.glPopMatrix();
gl2.glPushMatrix();
gl2.glRotatef( (float) 360/11, v1: 0, v2: 1, v3: 0);
gl2.glMaterialfv(GL2.GL_FRONT_AND_BACK, GL2.GL_AMBIENT_AND_DIFFUSE, new float[] {materials[0][0], materials[0][1], materials[0][2], materials[0][3], materials[0][4], materials[0][5], materials[0][6], materials[0][7], materials[0][8], materials[0][9] });
gl2.glMaterialfv(GL2.GL_FRONT_AND_BACK, GL2.GL_SPECULAR, new float[] {materials[5][6], materials[0][7], materials[12][8]}, (float) 0);
gl2.glMaterialf(GL2.GL_FRONT_AND_BACK, GL2.GL_SHININESS, (float) 20);
scianka(gl2, xPoints[3], yPoints[3], a2: 0, b2: 0, xPoints[4], yPoints[4], h);
gl2.glPopMatrix();
gl2.glPushMatrix();
gl2.glRotatef( (float) 360/11, v1: 0, v2: 1, v3: 0);
gl2.glMaterialfv(GL2.GL_FRONT_AND_BACK, GL2.GL_AMBIENT_AND_DIFFUSE, new float[] {materials[16][0], materials[16][1], materials[16][2], materials[16][3], materials[16][4], materials[16][5], materials[16][6], materials[16][7], materials[16][8], materials[16][9] });
gl2.glMaterialfv(GL2.GL_FRONT_AND_BACK, GL2.GL_SPECULAR, new float[] {materials[16][6], materials[16][7], materials[16][8]}, (float) 0);
gl2.glMaterialf(GL2.GL_FRONT_AND_BACK, GL2.GL_SHININESS, (float) 64);
scianka(gl2, xPoints[4], yPoints[4], a2: 0, b2: 0, xPoints[5], yPoints[5], h);
gl2.glPopMatrix();
gl2.glPushMatrix();

```

```

void scianka(GL2 gl2, float a1, float b1, float a2, float b2, float a3, float b3, float h) {

    gl2.glBegin(GL_TRIANGLE_FAN);
    gl2.glVertex3f(a1, v1: 0, b1);
    gl2.glVertex3f(a2, h, b2);
    gl2.glVertex3f(a3, v1: 0, b3);
    gl2.glEnd();
}

void piramidka(GL2 gl2, float r, float h) {
    float[] xPoints = new float[11];
    float[] yPoints = new float[11];
    for(int i = 1 ; i<=11;i++) {
        xPoints[i - 1] = (float) (r * Math.sin((2 * Math.PI / 11) * i));
        yPoints[i - 1] = (float) (r * Math.cos((2 * Math.PI / 11) * i));
    }
}

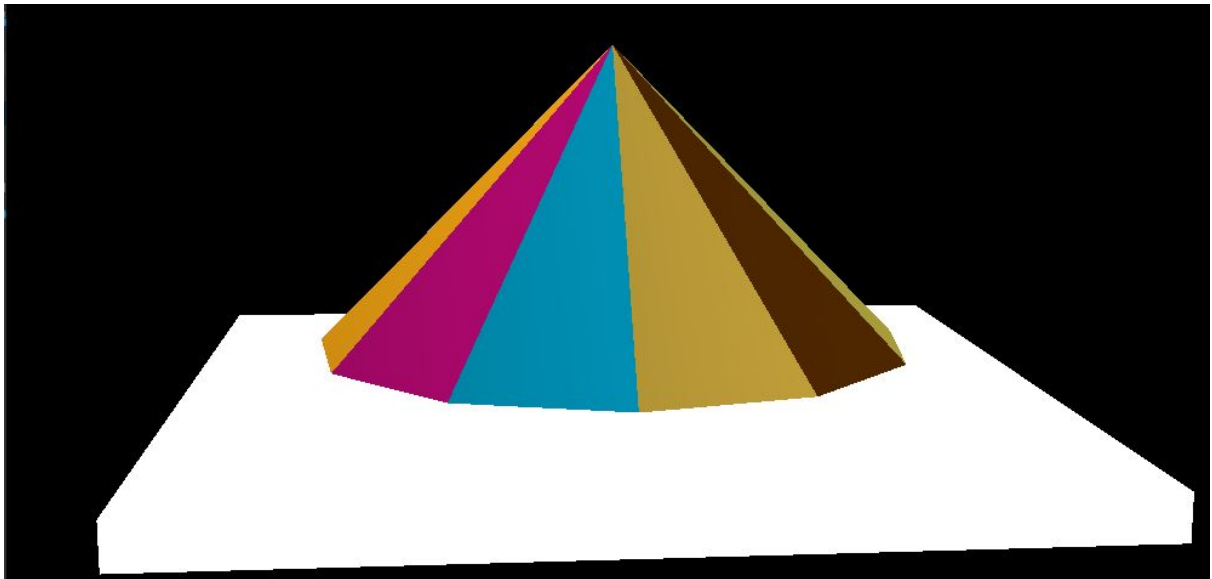
```

```

// TODO configure better lighting!
// ustawienie swiatla
float[] ambientLevel = { 0.8f, 0.8f, 0.8f, 1 };
gl2.glLightModelfv( GL2.GL_LIGHT_MODEL_AMBIENT, ambientLevel, i1: 0 );
// swiatla biale
float[] w1 = {0.6f, 0.6f, 0.6f, 1};
float[] w2 = {0.8f, 0.8f, 0.8f, 1};
float[] w3 = {0.15f, 0.15f, 0.15f, 1};
gl2.glTranslatef( v: 0, v1: 0, v2: 0);
gl2.glEnable(GL2.GL_LIGHT0);
float[] light1 = { 3,2,0,1 };
gl2.glLightfv(GL2.GL_LIGHT0, GL2.GL_POSITION, light1, i2: 0);
gl2.glLightfv( GL2.GL_LIGHT0, GL2.GL_DIFFUSE, w1, i2: 0);
gl2.glLightfv( GL2.GL_LIGHT0, GL2.GL_SPECULAR, w2, i2: 0);
gl2.glLightfv( GL2.GL_LIGHT0, GL2.GL_AMBIENT, w3, i2: 0);
gl2.glTranslatef( v: 0, v1: 0, v2: 0);
gl2.glEnable(GL2.GL_LIGHT1);
float[] light2 = { 0,2,3,1 };
gl2.glLightfv(GL2.GL_LIGHT1, GL2.GL_POSITION, light2, i2: 0);
gl2.glLightfv(GL2.GL_LIGHT1, GL2.GL_DIFFUSE, w1, i2: 0);
gl2.glLightfv(GL2.GL_LIGHT1, GL2.GL_SPECULAR, w2, i2: 0);
gl2.glLightfv(GL2.GL_LIGHT1, GL2.GL_AMBIENT, w3, i2: 0);
gl2.glTranslatef( v: 0, v1: 0, v2: 0);
gl2.glEnable(GL2.GL_LIGHT2);
float[] light3 = {-3,2,0,1 }; // ustawienia w osiach polozenie
gl2.glLightfv(GL2.GL_LIGHT2, GL2.GL_POSITION, light3, i2: 0);
gl2.glLightfv( GL2.GL_LIGHT2, GL2.GL_DIFFUSE, w1, i2: 0);
gl2.glLightfv( GL2.GL_LIGHT2, GL2.GL_SPECULAR, w2, i2: 0);
gl2.glLightfv( GL2.GL_LIGHT2, GL2.GL_AMBIENT, w3, i2: 0);
gl2.glTranslatef( v: 0, v1: 0, v2: 0);
gl2.glEnable(GL2.GL_LIGHT3);
float[] light4 = {0,2,-3,1 };
gl2.glLightfv(GL2.GL_LIGHT3, GL2.GL_POSITION, light4, i2: 0);
gl2.glLightfv( GL2.GL_LIGHT3, GL2.GL_DIFFUSE, w1, i2: 0);
gl2.glLightfv( GL2.GL_LIGHT3, GL2.GL_SPECULAR, w2, i2: 0);
gl2.glLightfv( GL2.GL_LIGHT3, GL2.GL_AMBIENT, w3, i2: 0);
}

```

Wynik:



Podsumowanie:

Na podstawie otrzymanego wyniku można stwierdzić, że Światło w OpenGL jest strasznie nie intuicyjne, aczkolwiek działa poprawnie, a materiały działają poprawnie.