

# **SPRAWOZDANIE**

Zajęcia: Grafika Komputerowa

Prowadzący: prof. dr hab. Vasyl Martsenyuk

## **Laboratorium 5**

06.04.2022

**Temat:** Geometria trójwymiarowa OpenGL

**Wariant:** 13, Black

Manuel Hływa

Informatyka I stopień,

stacjonarne,

4 semestr,

Gr. 1b

## 1.Polecenie:

Stworzyć dwa obiekty przy użyciu OpenGL (w języku C lub Java). Po uruchomieniu zakończonego programu naciśnięcie jednego z klawiszy numerycznych 1 lub 2 spowoduje wybranie wyświetlanego obiektu. Program już ustawia wartość zmiennej globalnej, `objectNumber`, aby powiedzieć, który obiekt ma zostać narysowany. Użytkownik może obracać obiekt za pomocą klawiszy strzałek, `PageUp`, `PageDown` i `Home`. Podprogram `display()` jest wywoływany, aby narysować obiekt. Podprogram ten z kolei wywołuje `draw()` i właśnie w `draw()` powinien być wykonana podstawowa praca. (Miejsce jest oznaczone `TODO`.) Dodaj również kilka nowych podprogramów do programu.

Obiekt 1. Korkociąg wokół osi  $\{x \mid y \mid z\}$  zawierający  $N$  obrotów. Punkty są stopniowo powiększane. Ustalić aktualny kolor rysujący na  $\{\text{zielony} \mid \text{niebieski} \mid \text{brązowy} \mid \dots\}$ .

Obiekt 2. Pyramida, wykorzystując dwa wachlarze trójkątów oraz modelowanie hierarchiczne (najpierw tworzymy podprogram rysowania jednego trójkąta; dalej wykorzystując przekształcenia geometryczne tworzymy piramidę). Podstawą piramidy jest wielokąt o  $N$  wierzchołkach.

## 2.Wprowadzane dane:

1)Kolor: czarny

2)Liczba kątów wielokąta: 13

## 3.Wykorzystane komendy:

a) korkociąg

```

private void korkociag(double n, GL2 gl2) {

    double pi = Math.PI;
    double t = 5;
    double r = 3;
    double kat, nastKat, x, y, z, h;

    h = t / 1550;
    z = -5;
    kat = 0;
    gl2.glLineWidth(10);
    gl2.glColor3d(0, 1, 0);

    gl2.glBegin(GL.GL_LINE_STRIP);
    gl2.glColor3f(0, 0, 0);
    int res = 36;
    float deg = 360/res;
    for(float i = 1; i <= n*res; i++) {
        x = Math.cos(Math.toRadians(deg*i));
        y = Math.sin(Math.toRadians(deg*i));
        gl2.glVertex3d(x*(0.01f*i), y*(0.01f*i), (i/res)-(n/2));
    }

    gl2.glEnd();
}

```

b) piramida

```

private void trojkat(float j, float k, float n, GL2 gl2) {

    gl2.glBegin(GL_TRIANGLE_FAN);

    gl2.glVertex3d((Math.cos(j * 2 * Math.PI / n)), (Math.sin(j * 2 * Math.PI / n)), 1.5);
    gl2.glVertex3d((Math.cos(k * 2 * Math.PI / n)), (Math.sin(k * 2 * Math.PI / n)), 1.5);
    gl2.glVertex3d(0, 0, 0);

    gl2.glEnd();
}

private void piramida(float size, float n, GL2 gl2) {

    gl2.glScalef(size, size, size);
    gl2.glRotatef(90, 1, 0, 0);
    gl2.glTranslatef(0, 0, -1);

    for (int i = 0; i < n; i++) {
        trojkat(i - 1, i, n, gl2);
    }
}

switch (objectNumber) {
    case 1:
        korkociag(13, gl2);
        break;
    case 2:
        piramida(7, 13, gl2);
        break;
}

```

Link do repozytorium: <https://github.com/DryPanther/Grafika-komputerowa>

#### 4. Wynik działania:

a)



b)

