**SPRAWOZDANIE**

Zajęcia: Grafika komputerowa

Prowadzący: prof. dr hab. Vasyl Martsenyuk

**Laboratorium 5**

**Data 22.03.2022**

**Temat: Geometria trójwymiarowa OpenGL**

**Wariant black/z**

Natalia Pierzchała

Informatyka I stopień,

stacjonarne,

4 semestr,

Gr. 2a

# 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() powinieneś wykonać podstawową pracę. (Miejsce jest oznaczone TODO.) Dodaj również kilka nowych podprogramów do programu.

Obiekt 1. Korkociąg wokół osi {x | y | z} zawierający N obrotów. Punkty są stopniowo powiększane. Ustalić aktualny kolor rysujący na {zielony | niebieski | brązowy | … }.

Obiekt 2. Pyramida, wykorzystując dwa wachlarze trójkątów oraz modelowanie hierarchiczne (najpierw tworzymy podprogramę rysowania jednego trójkonta; dalej wykorzystując przekształcenia geometryczne tworzymy pyramidę). Podstawą pyramidy jest wielokąt o N wierzchówkach.

# Wprowadzane dane i wykorzystane komendy:

**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);

}

}

**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;

y = -5;

kat = 0;

gl2.glLineWidth(3);

gl2.glColor3d(0, 0, 0);

gl2.glBegin(GL.***GL\_LINE\_STRIP***);

gl2.glColor3f(0, 0, 0);

**while** (y <= t) {

nastKat = (kat \* pi / 180);

x = Math.*cos*(nastKat) \* r;

z = Math.*sin*(nastKat) \* r;

gl2.glVertex3d(x, y, z);

y+=h;

kat += (n / 9);

}

gl2.glEnd();

}

**private** **void** draw(GL2 gl2) {

gl2.glRotatef(rotateZ, 0, 0, 1);

gl2.glRotatef(rotateY, 0, 1, 0);

gl2.glRotatef(rotateX, 1, 0, 0);

**switch** (objectNumber) {

**case** 1:

korkociag(8, gl2);

**break**;

**case** 2:

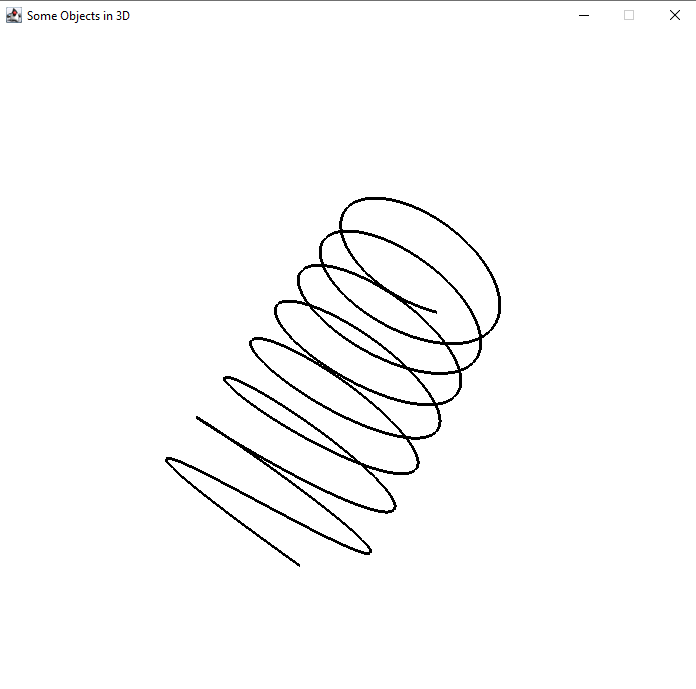
piramida(5, 15, gl2);

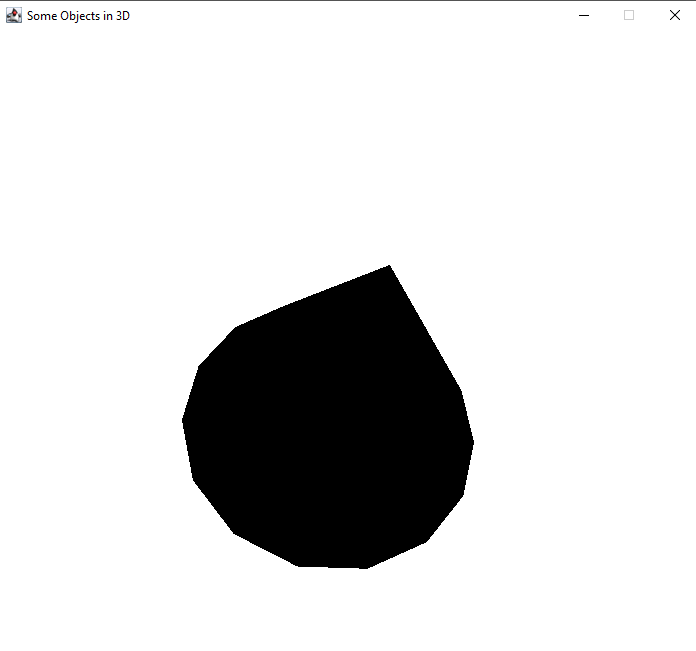
**break**;

}

}

# Wyniki działania:





# Wnioski:

Dzięki wykorzystaniu biblioteki OpenGL możemy stworzyć figury 3D i obserwować je z dowolnie wybranej perspektywy poprzez obracanie ich.