

SPRAWOZDANIE

Zajęcia: Grafika komputerowa

Prowadzący: prof. dr hab. Vasyl Martsenyuk

Laboratorium 4

Data 23.03.2022

Temat: „Geometria trójwymiarowa OpenGL”

Wariant oś x | kolor żółty

Wojciech Biegun
Informatyka I stopień,
stacjonarne,
4 semestr,
Gr. 2b

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()` powinieneś wykonać podstawową pracę. (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 podprogramę rysowania jednego trójkąta; dalej wykorzystując przekształcenia geometryczne tworzymy piramidę). Podstawą piramidy jest wielokąt o N wierzchołkach.

3. Wykorzystane komendy:

a) Użyte funkcje:

`glBegin()`
`glEnd()`
`glVertex3d()`
`glColor3f()`
`glLineWidth()`
`glScaleF()`
`glRotateF()`
`glTranslatef()`

b) Kod źródłowy:

```
private void triangle(float j, float k, float n, GL2 gl2) {  
    gl2.glBegin(gl2.GL_TRIANGLE_FAN);  
    gl2.glVertex3d((Math.cos(j * 2 * Math.PI / n)), (Math.sin(j * 2 * Math.PI / n)),  
1.5);
```

```

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

    gl2.glEnd();
}

private void piramid(float n, GL2 gl2) {

    gl2.glScalef(3, 3, 3);
    gl2.glRotatef(90, 1, 0, 0);
    gl2.glTranslatef(0, 0, -1);

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

}

private void corkscrew(double n, GL2 gl2) {

    double pi = Math.PI;
    double t = 7;
    double r = 2;
    double kat, nastKat, x, y, z, h;

    h = t / 1550;
    x = -5;
    kat = 0;
    gl2.glLineWidth(3);
    gl2.glColor3d(1, 1, 0);

    gl2.glBegin(GL.GL_LINE_STRIP);
    gl2.glColor3f(1, 1, 0);
    while (x <= t) {
        nastKat = (kat * pi / 180);
        y = Math.cos(nastKat) * r;
        z = Math.sin(nastKat) * r;

        gl2.glVertex3d(x, y, z);
        x+=h;
        kat += (n / 9);
    }
    gl2.glEnd();
}

/**
 * The method that draws the current object, with its modeling transformation.
 */
private void draw(GL2 gl2) {

    gl2.glRotatef(rotateZ,0,0,1);    // Apply rotations to complete object.
    gl2.glRotatef(rotateY,0,1,0);
    gl2.glRotatef(rotateX,1,0,0);

    // TODO: Draw the currently selected object, number 1, 2, 3, 4, 5, or 6.
    // (Objects should lie in the cube with x, y, and z coordinates in the
    // range -5 to 5.)

    switch (objectNumber) {
case 1:

```

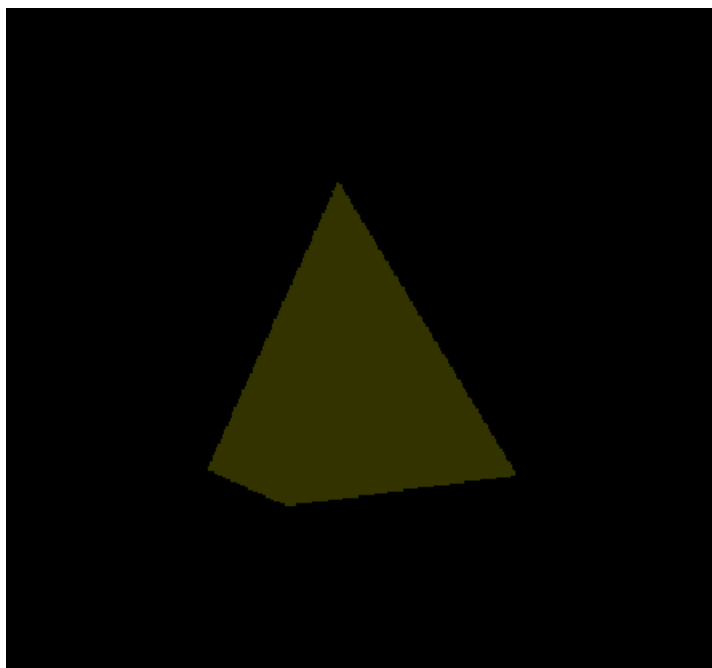
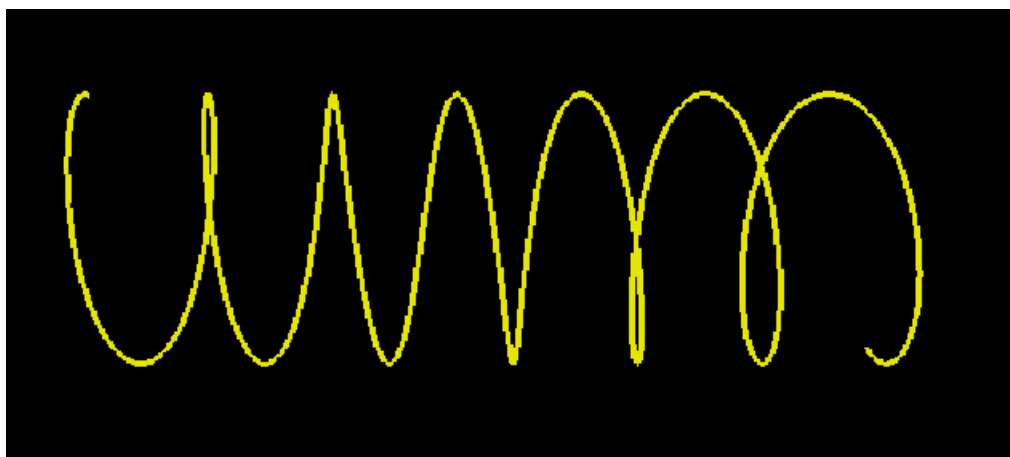
```
        corkscrew(8, gl2);  
        break;  
    case 2:  
        piramid(3, gl2);  
        break;  
    }  
}
```

Link do zdalnego repozytorium:

<https://github.com/WojciechBiegun/GK>

Zadanie zostało umieszczone w folderze OpenGL.

4. Wynik działania:



Wnioski:

Biblioteka OpenGL jest użytecznym narzędziem pozwalającym tworzyć grafikę 3D. Możemy dzięki niej tworzyć skomplikowane obiekty graficzne i dokonywać ich przekształceń.