

# **SPRAWOZDANIE**

Zajęcia: Grafika Komputerowa

Prowadzący: prof. dr hab. Vasyl Martsenyuk

**Laboratorium: 5**

**Data: 19.03.2024**

**Temat: "Geometria trójwymiarowa OpenGL"**

Michał Michalik  
Informatyka I stopień,  
stacjonarne,  
4 semestr,  
Gr.3a

## **Zadanie 1**

### **1. Polecenie:**

Stworzyć dwa obiekty przy użyciu OpenGL. Po uruchomieniu programu naciśnięcie

jednego z klawiszy numerycznych 1 lub 2 spowoduje wybranie wyświetlanego obiektu. Program 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.

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 podprogramy rysowania jednego trójkąta;

dalej wykorzystując przekształcenia geometryczne tworzymy piramidę).

Podstawą

pyramidy jest wielokąt o  $N$  wierzchołkach.

## 2. Wprowadzane dane:

Do zadania wybrałem informację na temat koloru oraz osi rotacji.

## 3. Wykorzystane komendy:

Do wykonania zadania należało zmodyfikować kod w Javascript

Obiekt 1:

```
41
42     function korkociag(size) {
43         var n = 14;
44         glColor3f(1, 0, 0); // Red
45         glScalef(size, size, size);
46         glBegin(GL_LINE_STRIP);
47         var angle = Math.PI / n;
48         for (var i = 0; i < n * n; i++) {
49             var x = Math.cos(angle * i) * (0.03 * i);
50             var y = i / (3 * n);
51             var z = Math.sin(angle * i) * (0.03 * i);
52             glVertex3d(x, y, z);
53         }
54         glEnd();
55     }
```

Obiekt 2:

```
57  ✓ function trojkat(p1, p2) {
58      glColor3f(1, 0, 0);
59      glBegin(GL_TRIANGLE_FAN);
60      glVertex3d(p1[0], p1[1], 2);
61      glVertex3d(p2[0], p2[1], 2);
62      glVertex3d(0, 0, 0);
63      glEnd();
64  }
65
66  ✓ function piramida(size) {
67      glScalef(size, size, size);
68      glRotatef(45, 0, 1, 0);
69      var points = [];
70  ✓   for (var i = 0; i < 13; i++) {
71          var x = Math.cos(i * 2 * Math.PI / 13);
72          var y = Math.sin(i * 2 * Math.PI / 13);
73          points.push([x, y]);
74      }
75  ✓   for (var i = 0; i < 13; i++) {
76          var p1 = points[(i == 0 ? points.length : i) - 1];
77          var p2 = points[i];
78          trojkat(p1, p2);
79      }
80  }
```

Wyświetlenie odpowiedniego:

```
    if (objectnumber == 1) {
        korkociag(0.1);
    } else if (objectnumber == 2) {
        piramida[0.4]
    }
}
```

4.

#### 4.1 Wynik działania po wciśnięciu 1:

##### A Simple Unlit Cube in OpenGL 1.1

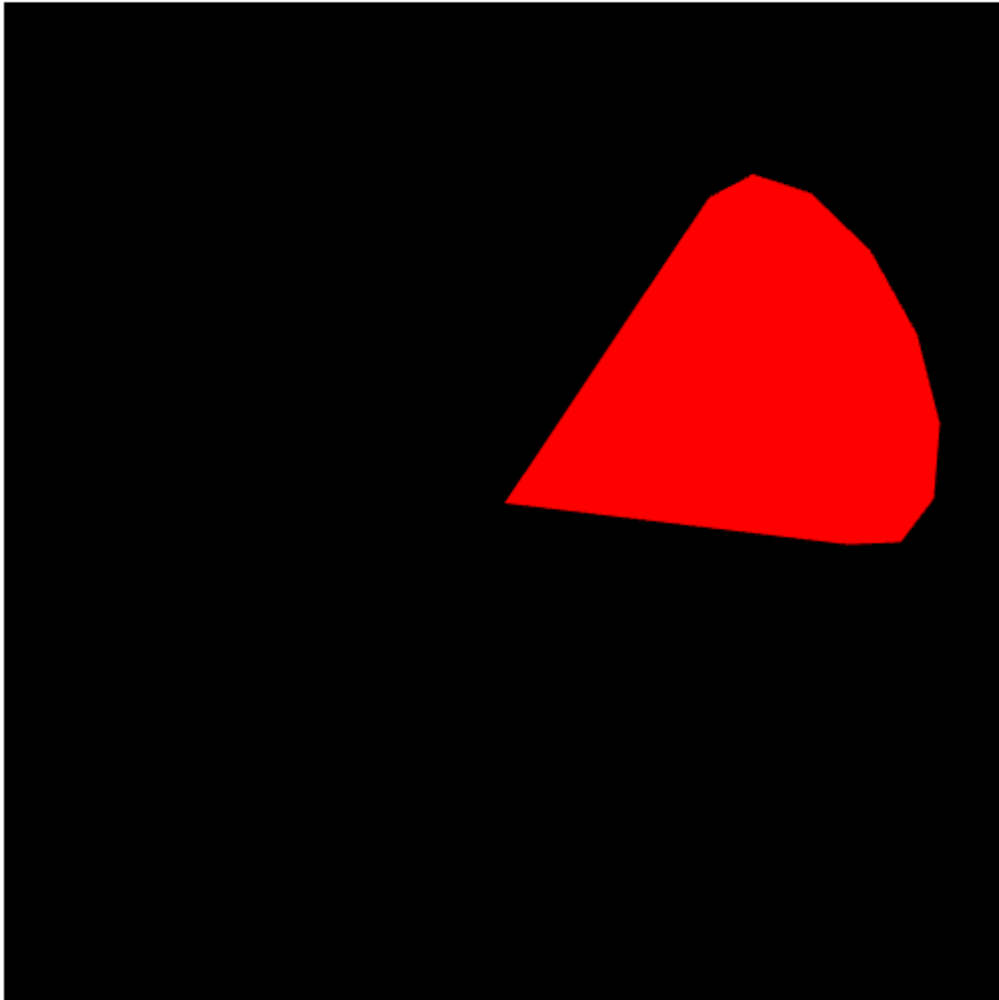
(Rotate using arrow keys, page up, page down, and home keys.)



#### 4.2 Wynik działania po wciśnięciu 2:

## A Simple Unlit Cube in OpenGL 1.1

(Rotate using arrow keys, page up, page down, and home keys.)



### 5. Wnioski:

Dzięki użyciu opengl, znając odpowiednie metody możemy stworzyć wielokąty o dowolnej ilości wierzchołków a ponadto dokonywać różnorodnych przekształceń: obrócenie, przesunięcie, pochylenie.