

SPRAWOZDANIE

Zajęcia: Grafika komputerowa

Prowadzący: prof. dr hab. Vasyl Martsenyuk

Laboratorium: Grafika Komputerowa

31.03.2020

Temat: „Geometria trójwymiarowa biblioteki OpenGL”

Wariant: 15

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

1. Polecenie:

Zadanie na laboratoriach składało się z poleceń dwóch:

- Pierwsze z nich dotyczyło korkociągu zawierającego N obrotów (w moim przypadku założyłem, że 15-taście tak jak w poprzednich zadaniach). Punkty korkociągu miały się stopniowo powiększać, a kolor aktualny rysowany miał być ustawiony niebieski | zielony | brązowy.
- Drugim zadaniem było narysowanie Piramidy, przy pomocy tworzenia trójkątów. Podstawą piramidy miał być N wielokąt. W zadaniu należało użyć modelowania hierarchicznego.

2. Wprowadzane dane:

Dane w zadaniu w większości są ustawione na stałe w programie jest to choćby promień wielokąta, czy wymiary brył. Rzeczą którą można przełączać w zadaniu i zmieniać która jest pobierana to informacja na temat rysowanego wielokąta. Klikając klawisze 1-6, można zmieniać rysowaną bryłę (choć dostępne bryły są jedynie pod klawiszami 1 i 2). Pod klawiszem 1 można znaleźć piramidę (zadanie 2), a pod klawiszem 2 jest korkociąg (zadanie 1). Inne informacje, które pobiera program dotyczą obracania bryły, można na bieżąco operować strzałkami by obracać bryłę (powinny także działać klawisze takie jak np. home/ page up/ page down).

3. Wykorzystane komendy:

Komendy wykorzystywane podczas pracy nad projektem to instrukcje: `glColor3f()`, który odpowiada za ustawienia koloru i przyjmuje 3 parametry stanowiące o barwie koloru, same jedynki ustawiają kolor biały, natomiast same zera kolor czarny. Kolejną komendą było `glbegin` i `glend`. Pomiedzy `glbegin` i `glend` odbywa się rysowanie całej bryły. W moim kodzie `glbegin` otrzymuje parametr `gl_triangle`, bądź `gl_lines` co ustawia jako obiekt rysujący trójkąt/ linie. W kodzie użyłem także `glEnable` aby zmienić stan `openGLa`, użyta jest także komenda `glLineWidth` ustawiająca grubość linii, komenda ta przyjmuje jeden parametr mówiący o tym jak gruba ma być linia. Inna komendą używaną w zadaniu jest `glVertex3f` która służy do ustawienia punktu w przestrzeni 3d. Dzięki tej komendzie możliwe było rysowanie brył 3d gdyż składają się one z wierzchołków które połączone są ze sobą tworząc całość. Cały kod do zadania tworzył poligon, w podstawie (wielokąt piętnastokątny), a następnie tworzył ściany piramidy tworząc obiekt 3d (w zadaniu 2gim). W innej części kod odpowiadał za rysowanie wielokątu, który się rozszerzał i zmieniał kolor (korkociąg).

Poniższy kod prezentuje rysowanie piramidy (zadanie 2-gie):

```
gl2.glColor3f( 1.0f, 1.0f, 1.0f );
/*
gl2.glVertex3f( 2.0f, 0.f, 0.0f );
gl2.glVertex3f( -2.0f, 0.f, 0.0f );
gl2.glVertex3f( 0.0f, 2.f, 0.0f ); */

int sides=15;
float r=2;

for(int i=0;i<sides;i++){
    double angle=i*2*Math.PI/sides;
    gl2.glColor3f( 1.0f, 0.0f, 0.0f );
    gl2.glVertex3f((float)(r*Math.cos(angle)),(float)(r*Math.sin(angle)),0.0f);
    angle=(i+1)*2*Math.PI/sides;
    gl2.glVertex3f((float)(r*Math.cos(angle)),(float)(r*Math.sin(angle)),0.0f);
    gl2.glVertex3f(0.0f,0.0f,0.0f);
}
for(int i=0;i<sides;i++){
    double angle=i*2*Math.PI/sides;
    gl2.glColor3f( 1.0f, 0.0f, 0.0f );
    gl2.glVertex3f((float)(r*Math.cos(angle)),(float)(r*Math.sin(angle)),0.0f);
    angle=(i+1)*2*Math.PI/sides;
    gl2.glVertex3f((float)(r*Math.cos(angle)),(float)(r*Math.sin(angle)),0.0f);
    gl2.glVertex3f(0.0f,0.0f,4.0f);
}
```

Natomiast ten fragment ukazuje rysowanie korkociągu:

```
gl2.glBegin( gl2.GL_LINES );//red

//korkociąg
int sides=15;
float r=0;
gl2.glEnable(gl2.GL_LINE_SMOOTH);
gl2.glLineWidth(1.0f);
//gl2 GLfloat lineWidthRange[2] = {0.0f, 0.0f};
// gl2 GLfloat lineWidthRange[2] = {0.0f, 0.0f};
float height=4;
for(int j=0; j<6;j++) {
    r+=0.5;
    for(int i=0;i<sides;i++){
        if(j<=1)
            gl2.glColor3f(0.0f, 1.0f, 0.0f );
        else if(j>1&&j<=3)
            gl2.glColor3f( 0.0f, 0.0f, 1.0f );
        else if(j>3)
            gl2.glColor3f(0.5f, 0.35f, 0.05f);
        double angle=i*2*Math.PI/sides;
        gl2.glVertex3f((float)(r*Math.cos(angle)),(float)(r*Math.sin(angle)),height);
        angle=(i+1)*2*Math.PI/sides;
        gl2.glVertex3f((float)(r*Math.cos(angle)),(float)(r*Math.sin(angle)),height);
        height-=0.1f;
    }
}
gl2.glEnd();
```

A tutaj natomiast jest pokazane wywołanie odpowiedniego rysowania zależnie od wciśniętego klawisza:

```
if(objectNumber==1) {

    draw(gl2);}
else if (objectNumber==2)
{
    draw2(gl2);

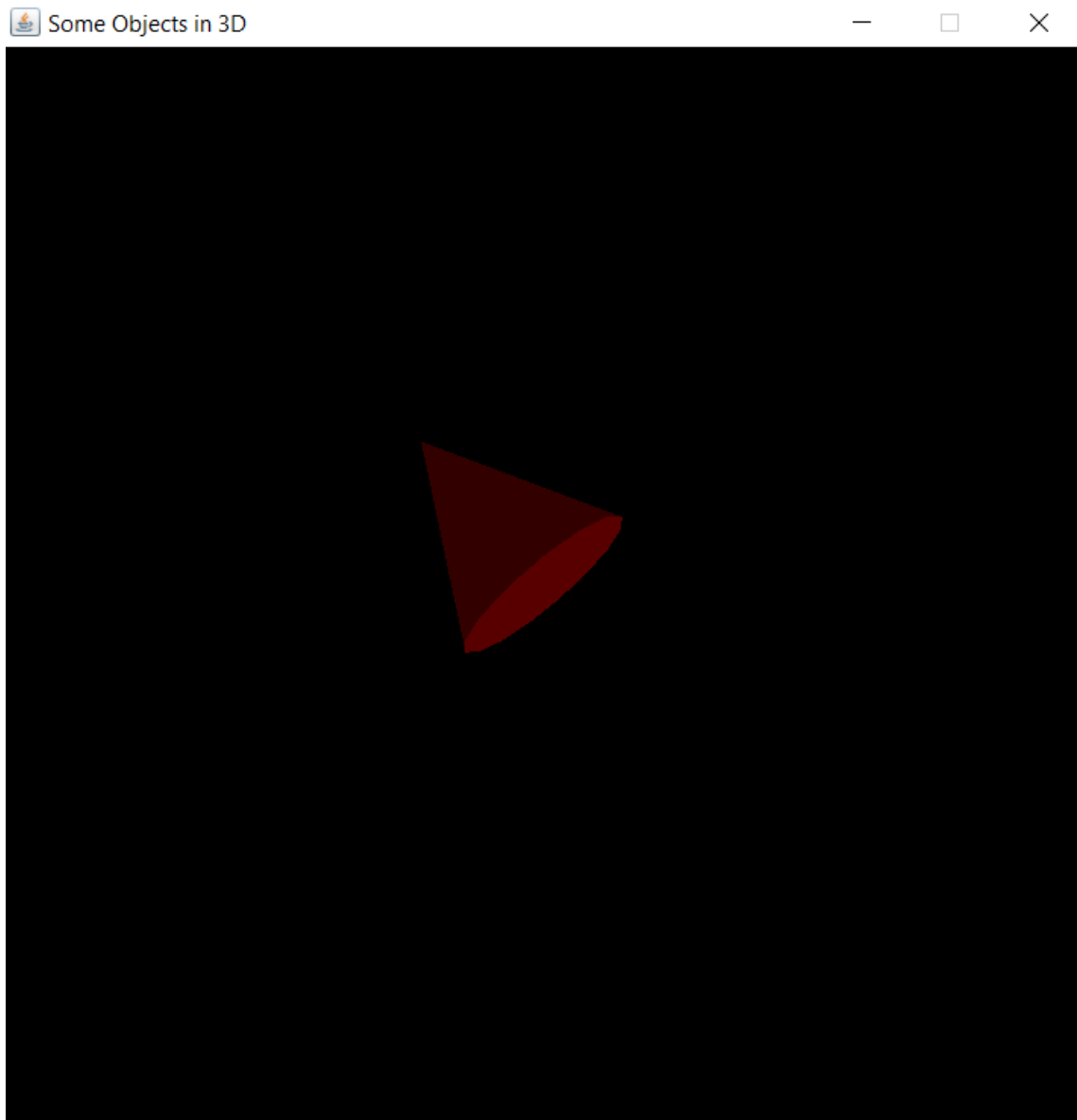
}
```

Link do zdalnego repozytorium zawierającego projekt:

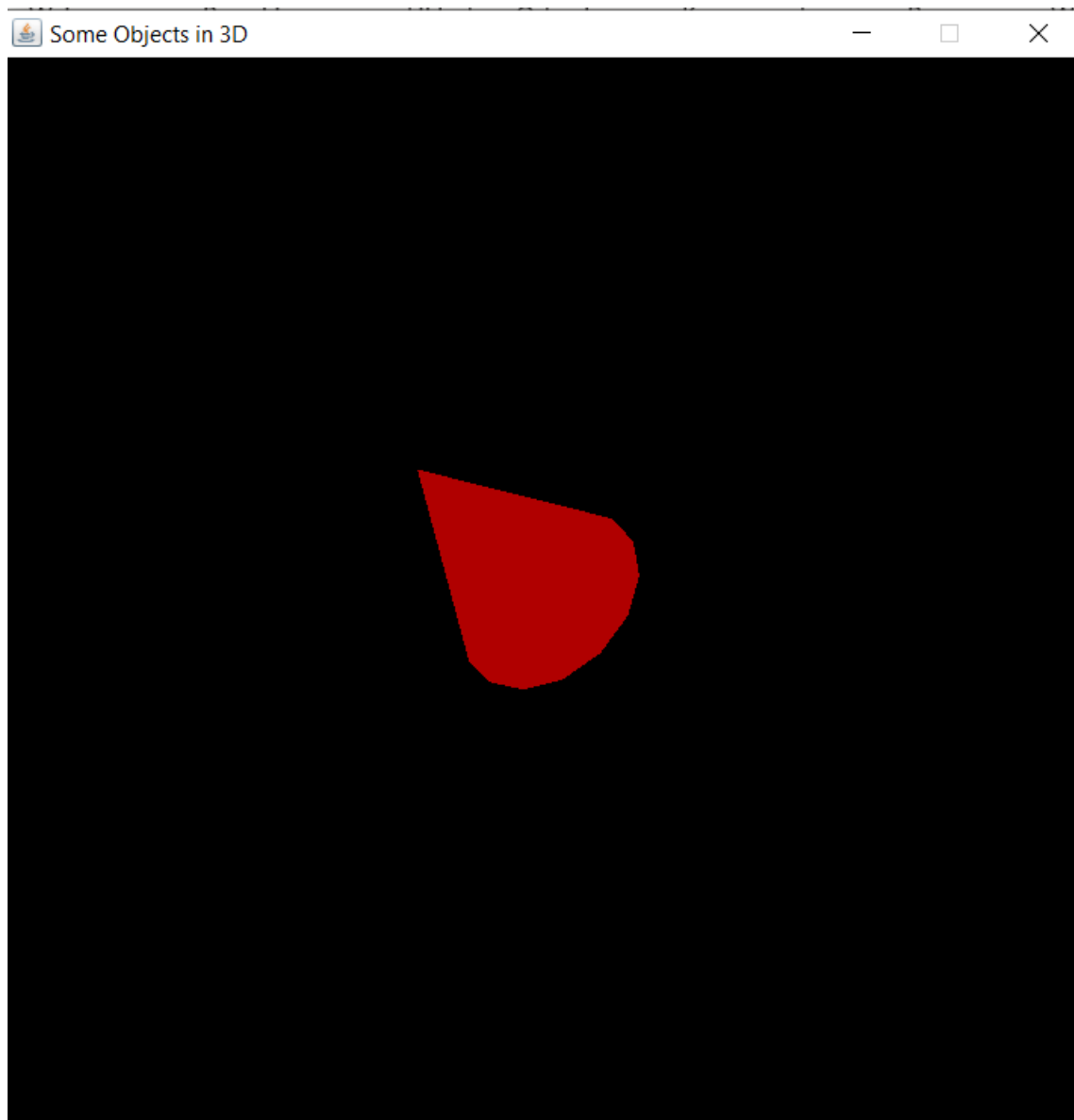
<https://github.com/Jarverr/GrafikaKomputerowaLab5>

4. Wynik działania:

Zrzuty ekranu prezentujące wyniki zadania:

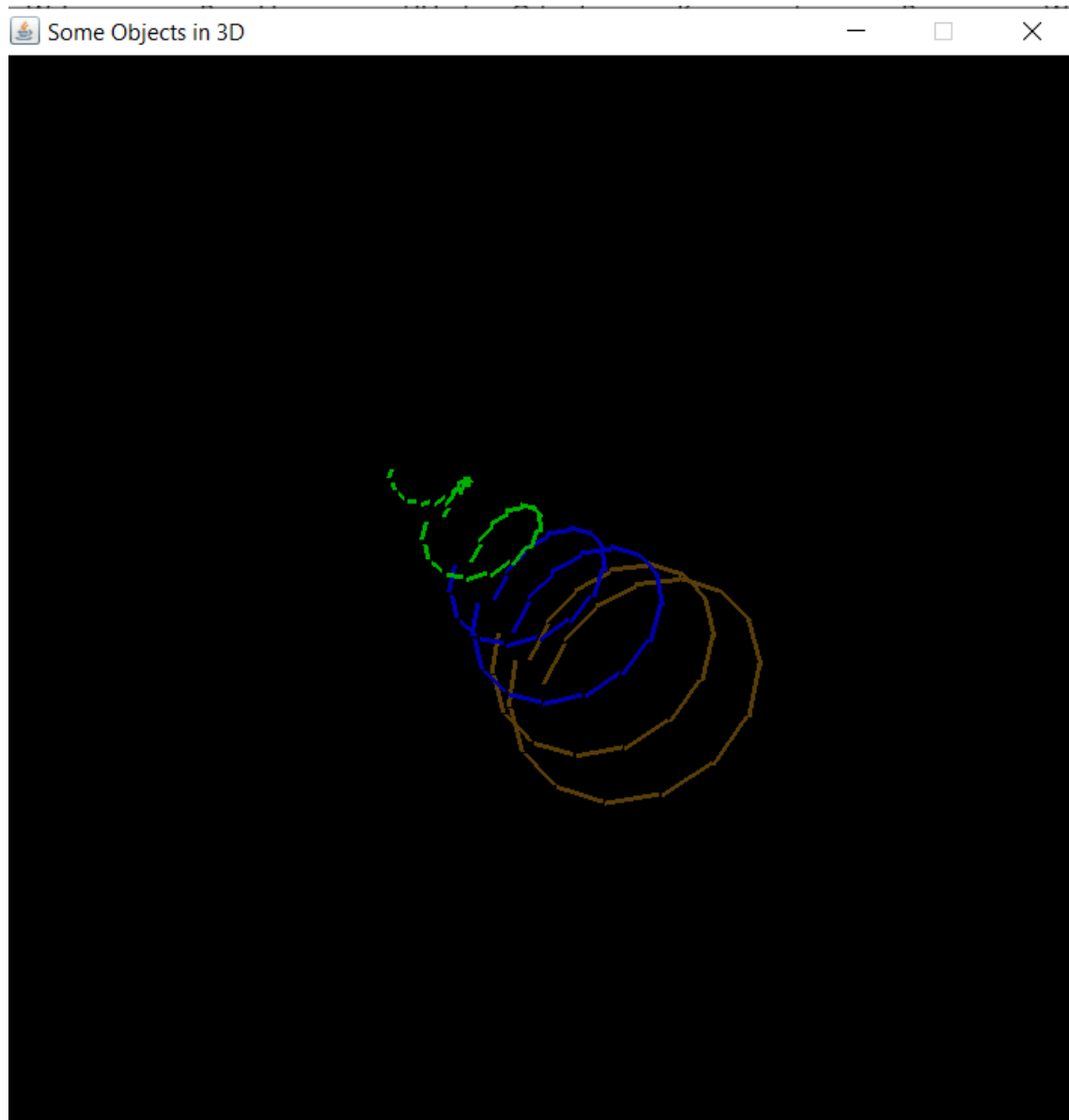


Screenshot 1 – zadanie 2

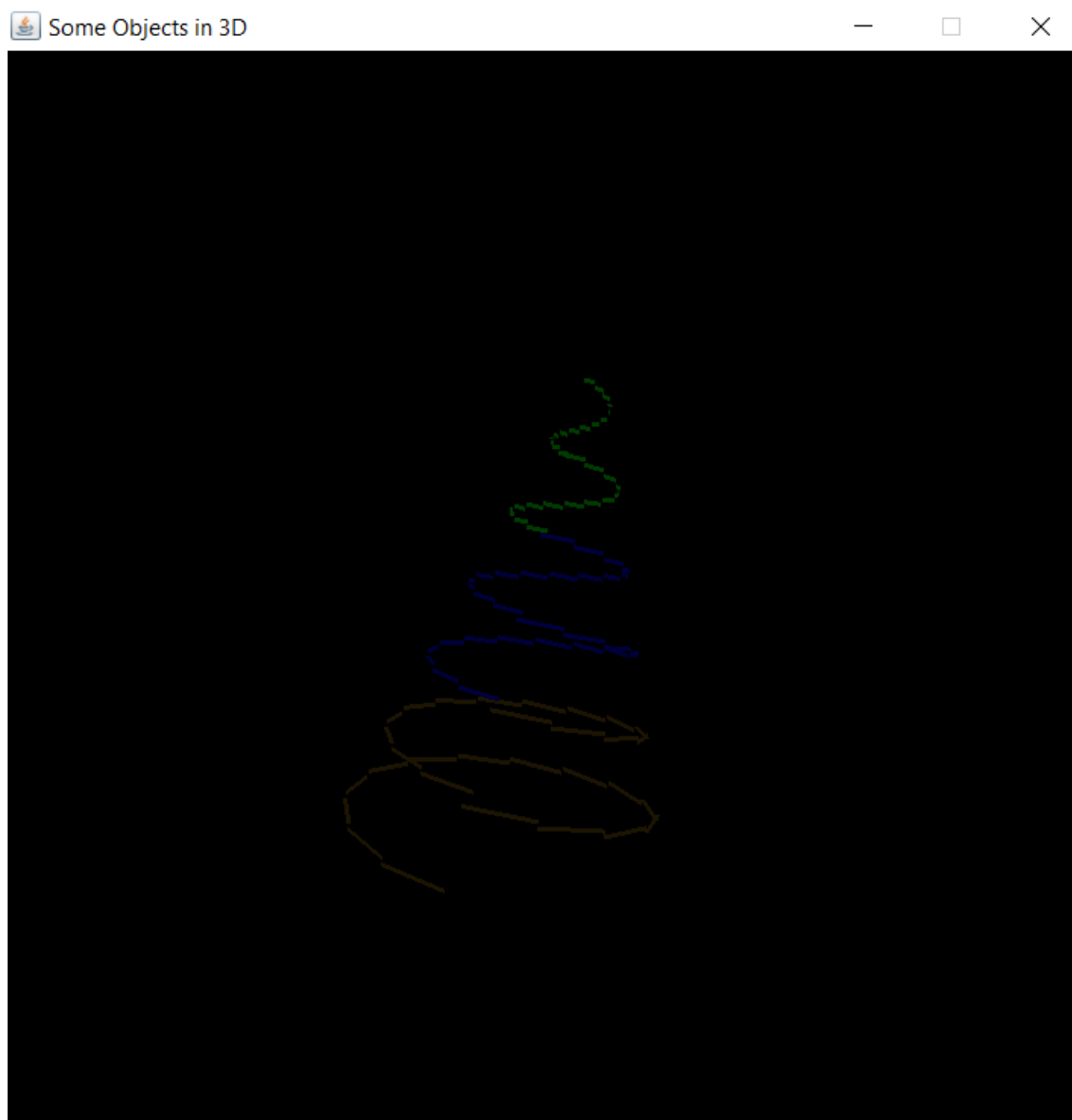


Screenshot 2 – zadanie 2

W zadaniu 1-szym ze względu na problem z interpretacją polecenia, nie jestem pewien czy chodziło o taki kształt, aczkolwiek zrozumiałem, że taki był cel korkociąg (czyli coś na kształt wiru), który posiada w sobie 3 kolory.



Screenshot 3 – zadanie 1



Screenshot 4 – zadanie 1

5. Wnioski:

Dzięki możliwościom jakie daje biblioteka OpenGL, jestem w stanie w stosunkowo prosty sposób stworzyć trójwymiarową bryłę. Utrudnione nieco dla OpenGLa wydaje się rysowanie powierzchni krzywych, ze względu na brak narzędzia (przynajmniej dla tych zadań), które służyłoby do rysowania krzywych. Całość działa jednak sprawnie umożliwia rysowanie w 2d i 3d, zmienia dowolne kolorów i modyfikację tworzonego obrazu.