

SPRAWOZDANIE 5

Zajęcia: Grafika komputerowa

Prowadzący:

prof. dr hab. Vasyl Martsenyuk

Laboratorium: Grafika Komputerowa 20.03.2020

Temat: Geometria trójwymiarowa OpenGL

Sebastian Pierog

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

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.

Kod źródłowy:

```

private void draw(GL2 gl2) {
    // tu można obracać
    gl2.glRotatef(rotateZ, v1: 0, v2: 0, v3: 1); // Apply rotations to complete object.
    gl2.glRotatef(rotateV, v1: 0, v2: 1, v3: 0);
    gl2.glRotatef(rotateX, v1: 1, v2: 0, v3: 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.)
    // wybrane figury

    switch(objectNumber){
        case 1:piramidka(gl2, r: 7, N: 11, h: 7);
            break;
        case 2:korkociąg(gl2, N: 15, r: 0, g: 255, b: 0, os: 'X');
            break;
        case 3:korkociąg(gl2, N: 15, r: 0, g: 0, b: 255, os: 'Y');
            break;
        case 4:korkociąg(gl2, N: 15, r: 150, g: 75, b: 0, os: 'Z');
            break;
    }
}

```

```

void scianka(GL2 gl2, float a1,float b1, float a2, float b2, float a3 ,float b3,float h) {

    // f s d i jak jest v na koncu to tablica i wrzucam tam tablice a nie 3
    // jak 4d to x y z i skala jeszcze czyli w.      czyli x*w y*w z*w

    gl2.glBegin(GL_TRIANGLE_FAN);
    gl2.glVertex3f(a1, v1: 0,b1);
    gl2.glVertex3f(a2, h,b2);
    gl2.glVertex3f(a3, v1: 0,b3);
    gl2.glEnd();
}

void piramidka(GL2 gl2, float r, int N,float h) {
    float[] xPoints = new float[N];
    float[] yPoints = new float[N];
    //
    // podstawa
    for(int i = 1 ; i<=N;i++) {
        xPoints[i - 1] = (float) (r * Math.sin((2 * Math.PI / N) * i));
        yPoints[i - 1] = (float) (r * Math.cos((2 * Math.PI / N) * i));
    }
    gl2.glPushMatrix();
    gl2.glColor3f(v: 2, v1: 0, v2: 0);
    //11 scian
    for(int i = 0 ; i<N-1;i++)
    {
        gl2.glPushMatrix();
        gl2.glRotatef(v: 360/N, v1: 0, v2: 1, v3: 0);
        scianka(gl2,xPoints[i], yPoints[i], a2: 0, b2: 0, xPoints[i+1], yPoints[i+1],h);
        gl2.glPopMatrix();
    }
    gl2.glPushMatrix();
    gl2.glRotatef(v: 360/N, v1: 0, v2: 1, v3: 0);
    scianka(gl2,xPoints[N-1], yPoints[N-1], a2: 0, b2: 0, xPoints[0], yPoints[0],h);
    gl2.glPopMatrix();
}

```

```

void korkociag(GL2 gl2,int N, int r, int g, int b,char os)
{
    float[] xPoints = new float[30];
    float[] yPoints = new float[30];
    for(int i = 1 ; i<=30;i++) {
        xPoints[i - 1] = (float) (1 * Math.sin((2 * Math.PI / 30) * i));
        yPoints[i - 1] = (float) (1 * Math.cos((2 * Math.PI / 30) * i));
    }
    gl2.glPushMatrix();
    gl2.glColor3f(r, g, b);
    float wielkosc = 1.0f;
    float ilosc = -4f;
    for (int i = 0;i<N;i++) {
        for (int j = 0; j < 30; j++){
            gl2.glPointSize(wielkosc);
            gl2.glBegin(GL_POINTS);
            gl2.glPushMatrix();
            gl2.glRotatef( v1: 12, v1: 0, v2: 1, v3: 0);
            switch(os) {
                case 'X': gl2.glVertex3f(ilosc,xPoints[j], yPoints[j]);
                    break;
                case 'Y': gl2.glVertex3f(xPoints[j], ilosc, yPoints[j]);
                    break;
                case 'Z': gl2.glVertex3f(xPoints[j], yPoints[j],ilosc);
                    break;
            }
            gl2.glPopMatrix();

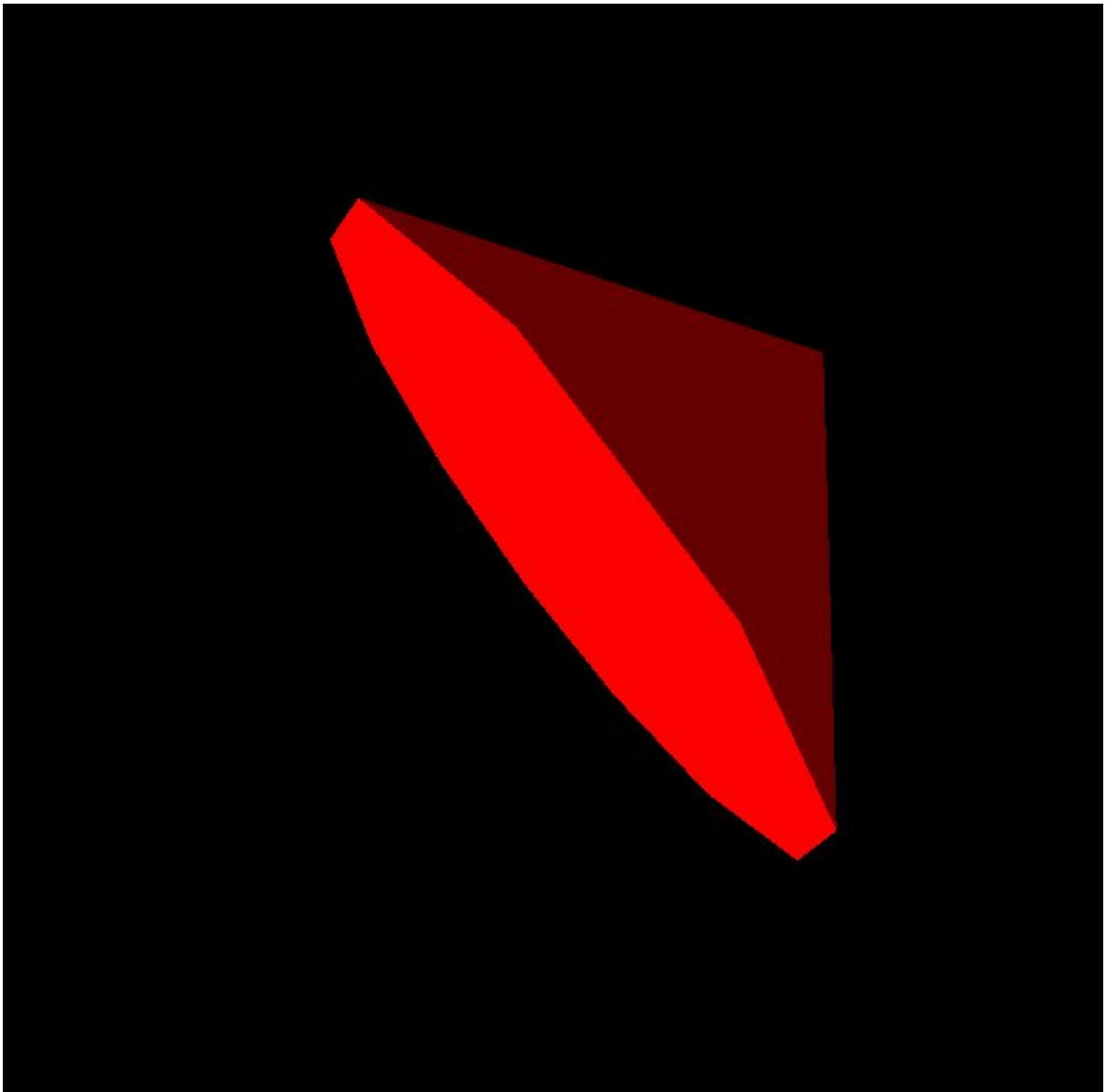
            gl2.glEnd();

            wielkosc+=(float)0.05/N;

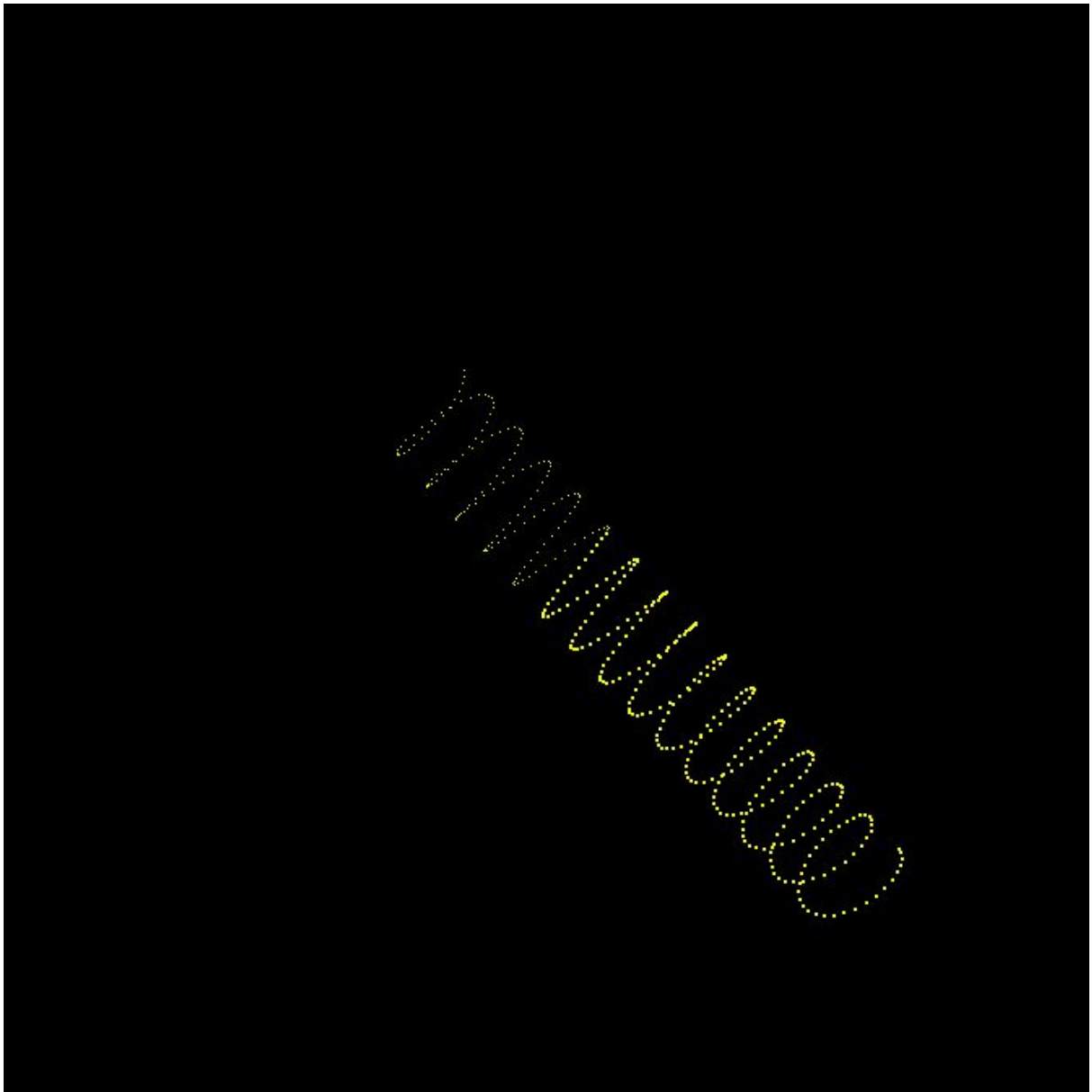
            ilosc+=(float)12/30/N;
        }
    }
}

```

Wyniki:



COMMUNITY



Podsumowanie:

Na podstawie wyników możemy stwierdzić, że tworzenie grafiki za pomocą biblioteki OpenGL jest sprawne i proste. Ćwiczenie przebiegło poprawnie oraz spełnia założenia wynikające z treści zadania.