Московский авиационный институт

(Национальный исследовательский университет)

Факультет: «Информационные технологии и прикладная математика»
Кафедра: 806 «Вычислительная математика и программирование»
Дисциплина: «Компьютерная графика»

Лабораторная работа № 7

Тема: Построение плоских полиномиальных кривых.

Студентка: Довженко А.А.
Группа: 80-307
Преподаватель: Чернышов Л.Н.
Дата:
Оценка:

Москва, 2018

1. Постановка задачи

Написать программу, строящую полиномиальную кривую по заданным точкам. Обеспечить возможность изменения позиции точек и, при необходимости, значений касательных векторов и натяжения.

Вариант 7.

Кривая Безье 5-й степени

2. Решение задачи

Кривая строится в соответствии с формулой по 6-ти точкам:

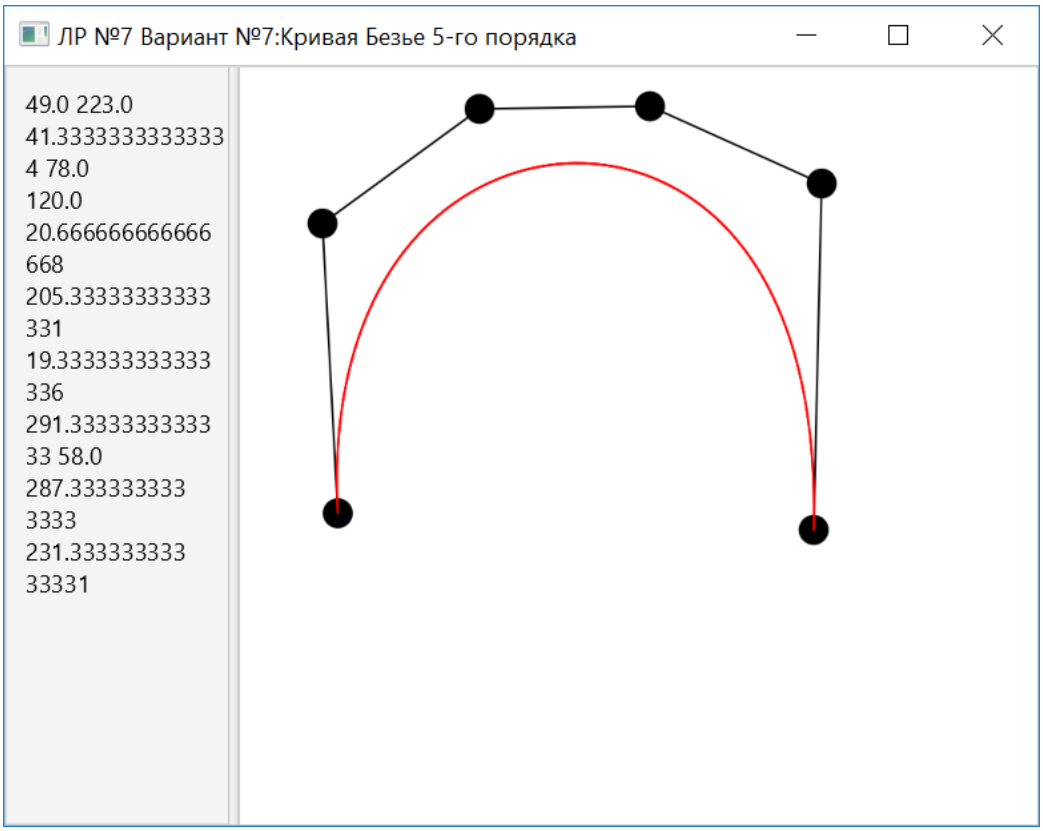
$$B(t)=(1-t)^5 P_0+5t(1-t)^4 P_1+10t^2(1-t)^3 P_2$$

$$+10t^3(1-t)^2 P_3+5t^4(1-t)P_4+t^5 P_5$$

Параметр t изменяется в промежутке [0;1]. В связи с простотой использования мною была использована библиотека JavaFX.

3. Руководство по использованию программы

Запустить IntelliJ, открыть в нём проект с программой и запустить его. Построенная фигура.



4. Листинг программы

```
// Довженко А.А. М8О-307Б
// Лабораторная работа №7
// Кривая Безье 5го порядка
// Главный класс
package main;

import javafx.application.Application;
import javafx.fxml.FXMLLoader;
import javafx.scene.Parent;
import javafx.scene.Scene;
import javafx.stage.Stage;

public class Main extends Application {

    @Override
    public void start(Stage primaryStage) throws Exception{
        Parent root = FXMLLoader.load(getClass().getResource("sample.fxml"));
                primaryStage.setTitle("ЛР №7 Вариант №7:Кривая Безье 5-го
порядка");
        primaryStage.setScene(new Scene(root, 300, 275));
        primaryStage.show();
    }


    public static void main(String[] args) {
        launch(args);
    }

}

package main;


import javafx.fxml.FXML;
import javafx.scene.input.MouseEvent;
import javafx.scene.layout.AnchorPane;
import javafx.scene.text.Text;

public class Controller {

    ResizableCanvas canvas;
```

```
    @FXML
    public AnchorPane canvasHolder;

    @FXML
    public Text text1;

    @FXML
    public Text text2;

    @FXML
    public Text text3;

    @FXML
    public Text text4;

    @FXML
    public Text text5;

    @FXML
    public Text text6;

    @FXML
    void initialize() {
                        canvas = new ResizableCanvas(canvasHolder.getWidth(),
canvasHolder.getWidth());

        AnchorPane.setTopAnchor(canvas, 0.);
        AnchorPane.setBottomAnchor(canvas, 0.);
        AnchorPane.setLeftAnchor(canvas, 0.);
        AnchorPane.setRightAnchor(canvas, 0.);
        canvasHolder.getChildren().add(canvas);
        printText();
        canvas.resize(canvasHolder.getWidth(), canvasHolder.getHeight());

        canvas.addEventHandler(MouseEvent.MOUSE_PRESSED, event -> {
            for (int i = 0; i < 6; i++) {
                //проверка на расположение мышки во время клика
                        //если клик совершается по какой-то точке, она считается
активной
                if(canvas.dots[i].inner(event.getX(),event.getY())) {
                    canvas.chosen = i;
                    break;
```

```
            }
        }
    });

    canvas.addEventHandler(MouseEvent.MOUSE_DRAGGED, event -> {
        if(canvas.chosen != -1) {
            //перемещение активной точки
             double x = Math.min(Math.max(10, event.getX()), canvas.width -
10);
             double y = Math.min(Math.max(10, event.getY()), canvas.height -
10);
            canvas.dots[canvas.chosen].x = x;
            canvas.dots[canvas.chosen].y = y;
            printText();
            canvas.resize(canvas.width, canvas.height);

        }
    });

        canvas.addEventHandler(MouseEvent.MOUSE_RELEASED, event ->
canvas.chosen = -1);
  }

  private void printText() {
    text1.setText(canvas.dots[0].x + " " + canvas.dots[0].y);
    text2.setText(canvas.dots[1].x + " " + canvas.dots[1].y);
    text3.setText(canvas.dots[2].x + " " + canvas.dots[2].y);
    text4.setText(canvas.dots[3].x + " " + canvas.dots[3].y);
    text5.setText(canvas.dots[4].x + " " + canvas.dots[4].y);
    text6.setText(canvas.dots[5].x + " " + canvas.dots[5].y);
  }

} package main;

import java.util.Vector;

public class Curve {
  public double[] x;
  public double[] y;

  public Curve(Dot[] dots) {
    x = new double[6];
    y = new double[6];
```

```
    for (int i = 0; i < 6; i++) {
        x[i] = dots[i].x;
        y[i] = dots[i].y;
    }
}

Vector<Double> count(double t) {
    //вычисление точки кривой в зависимости от заданного t
    Vector<Double> res = new Vector<>();

    double[] tp = new double[6];
    tp[0] = 1;
    tp[1] = t;

    double[] t_1p = new double[6];
    t_1p[0] = 1;
    t_1p[1] = 1 - t;
    for (int i = 2; i < 6; i++) {
        t_1p[i] = t_1p[i - 1] * t_1p[1];
        tp[i] = tp[i - 1] * tp[1];
    }

    double x = 0;
    double y = 0;
    int[] k = new int[] {1, 5, 10, 10, 5, 1};
    for (int i = 0; i < 6; i++) {
        x += tp[i] * t_1p[5 - i] * this.x[i] * k[i];
        y += tp[i] * t_1p[5 - i] * this.y[i] * k[i];
    }

    res.add(x);
    res.add(y);
    return res;
}

Vector<Vector<Double>> countCurve() {
    //рассчет точек кривой
    Vector<Vector<Double>> res = new Vector<>();
    Vector<Double> xs = new Vector<>();
    Vector<Double> ys = new Vector<>();

    double n = Math.pow(10,3);
    double step = 1 / n;
```

```
        for (int i = 0; i <= n; i++) {
            Vector<Double> tmp = count(step * i);
            xs.add(tmp.get(0));
            ys.add(tmp.get(1));
        }

        res.add(xs);
        res.add(ys);
        return res;
    }
}
```

```
package main;

import javafx.scene.paint.Color;


public class Dot {
    public double x;
    public double y;
    public double r;
    Color color;

    public Dot(double x, double y, double r, Color color) {
        this.x = x;
        this.y = y;
        this.r = r;
        this.color = color;
    }

    public boolean inner(double x, double y) {
        //проверка, является ли точка внутренней для данного круга
        double dx = this.x - x;
        double dy = this.y - y;
        return r * r >= dx * dx  +  dy * dy;
    }

}
```

```
// Довженко А.А. М8О-307Б
// Холст
```

```
package main;

import javafx.scene.canvas.Canvas;
import javafx.scene.canvas.GraphicsContext;
import javafx.scene.paint.Color;

import java.util.Random;
import java.util.Vector;


public class ResizableCanvas extends Canvas{

    public double height,width;

    public Dot[] dots;
    int chosen;

    @Override
    public boolean isResizable() {
        return true;
    }

    public ResizableCanvas(double width, double height) {

        super(width,height);
        this.width = width;
        this.height = height;

        dots = new Dot[6];
        Random rand = new Random();
        //в начале работы точки расположены по умолчанию
        dots[0] = new Dot(49, 223, 15, Color.rgb(0,0,0));
        dots[1] = new Dot(60, 34, 15,  Color.rgb(0, 0, 0));
        dots[2] = new Dot(90, 34, 15,  Color.rgb(0,0,0));
        dots[3] = new Dot(90, 125, 15, Color.rgb(0, 0,  0));
        dots[4] = new Dot(110, 220, 15,Color.rgb(0,  0, 0 ));
        dots[5] = new Dot(170, 50, 15, Color.rgb( 0, 0,0));

        chosen = -1;

    }

    @Override
```

```
    public void resize(double width,double height) {
        //во время изменения размера происходит отрисовка
        setWidth(width);
        setHeight(height);

        GraphicsContext gc = this.getGraphicsContext2D();
        gc.setFill(Color.WHITE);
        gc.fillRect(0,0,getWidth(),getHeight());


        for (int i = 0; i < 6; i++){
            if(i != 5) {
                gc.setStroke(Color.BLACK);
                gc.strokeLine(dots[i].x,dots[i].y, dots[i + 1].x, dots[i + 1].y);
            }

            gc.setFill(dots[i].color);
                 gc.fillOval(dots[i].x - dots[i].r / 2, dots[i].y - dots[i].r / 2, dots[i].r,
dots[i].r);
        }

        this.height = height;
        this.width = width;

        Vector<Vector<Double>> curve = new Curve(dots).countCurve();

        gc.setStroke(Color.rgb(255, 0, 0));

        Vector<Double> x = curve.get(0);
        Vector<Double> y = curve.get(1);
        for (int i = 1; i < x.size(); i++) {
            gc.strokeLine(x.get(i - 1), y.get(i - 1), x.get(i), y.get(i));
        }


    }
}
```

5. Используемые источники

1. Документация библиотеки JavaFX [Электронный ресурс]. URL: https://docs.oracle.com/javafx/2/ (дата обращения: 21.12.2018).
2. Статья по OpenGL [Электронный ресурс]. URL: https://habr.com/post/111175/ (дата обращения: 20.12.2018).