**Московский авиационный институт**

**(Национальный исследовательский университет)**

Факультет прикладной математики и физики

Кафедра вычислительной математики и программирования

**Лабораторная работа № 7**

по курсу «Компьютерная графика»

Тема: Построение плоских полиномиальных кривых

Студент: Сахарин Н.А.

Группа: 80-308

Преподаватель: Чернышов Л.Н.

Оценка:

Москва, 2018

1. Постановка задачи

Написать программу, строящую полиномиальную кривую по заданным точкам. Обеспечить возможность изменения позиции точек.

2. Решения задачи

Данная лабораторная работа выполнена на языке программирования C++. Кривая Безье задается параметрически, параметр t пробегает множество от 0 до 1. Кривая строится по опорным точкам.

Кривая всегда лежит в выпуклом многоугольнике, образованном особыми точками. Также она не всегда проходит через них.

3. Руководство по использованию программы

При запуске программы пользователь видит приглашение для ввода точек. Необходимо ввести 3 точки, покоординатно, то есть сначала x\_i затем y\_i.

После ввода точек строится кривая. Положение любой из 3 точек можно изменить, предварительно нажав одну из клавиш F1 F2 F3, затем необходимо ввести новые координаты точки.

4. Листинг программы

#include <GL/glut.h>

#include <iostream>

#include <GL/glu.h> /\* this includes the necessary X headers \*/

#include <GL/gl.h>

#include <cmath>

using namespace std;

#define PI 3.14159265

double rx = 0;

double ry = 0;

double rz = 0;

float \*x;

float \*y;

void dis() {

glClearColor(1.0, 1.0, 1.0, 1.0);

glClear(GL\_COLOR\_BUFFER\_BIT | GL\_DEPTH\_BUFFER\_BIT);

glLoadIdentity();

glMatrixMode(GL\_MODELVIEW);

glLineWidth(1);

glBegin(GL\_LINES);

glColor3f(0.0, 0.0, 0.0);

glVertex2f(1.0, 0.0);

glVertex2f(-1.0, 0.0);

glEnd();

glBegin(GL\_LINES);

glColor3f(0.0, 0.0, 0.0);

glVertex2f(0.0, 1.0);

glVertex2f(0.0, -1.0);

glEnd();

float x\_1, y\_1 = 0;

glBegin(GL\_LINE\_STRIP);

glColor3f(1.0, 0.0, 0.0);

for (double t = 0; t < 1; t += 0.01) {

x\_1 = pow(1. - t, 2) \* x[0] + t \* (1. - t) \* 2 \* x[1] + t \* t \* x[2];

y\_1 = pow(1. - t, 2) \* y[0] + t \* (1. - t) \* 2 \* y[1] + t \* t \* y[2];

glVertex2f(x\_1, y\_1);

}

glEnd();

glFlush();

glutSwapBuffers();

}

void reshape(int w, int h)

{

glViewport( 0, 0, w, h); // сброс текущей области вывода

glMatrixMode( GL\_PROJECTION );

glLoadIdentity();

glMatrixMode( GL\_MODELVIEW );

glLoadIdentity();

}

void key\_board(int key, int a, int b)

{

int i = 0;

if (key == GLUT\_KEY\_F1) {

cin>>x[i];

cin>>y[i];

}else if (key == GLUT\_KEY\_F2) {

i = 1;

cin>>x[i];

cin>>y[i];

}

else if (key == GLUT\_KEY\_F3) {

i = 2;

cin>>x[i];

cin>>y[i];

}

// Запрос обновления изображения

glutPostRedisplay();

}

int main(int argcp, char \*\*argv) {

cout<<"Enter point"<<endl;

x = new float [3];

y = new float [3];

for (int i = 0; i < 3; i++) {

cin>>x[i];

cin>>y[i];

}

glutInit(&argcp, argv);

glutInitDisplayMode(GLUT\_RGBA | GLUT\_DOUBLE | GLUT\_DEPTH);

glutInitWindowSize(640, 480);

glutInitWindowPosition(0, 0);

glutCreateWindow("Sakharin 80-308");

glEnable(GL\_DEPTH\_TEST);

glutReshapeFunc(reshape);

glutDisplayFunc(dis);

glutSpecialFunc(key\_board);

glutMainLoop();

return 0;

}

Screenshot

