**МИНОБРНАУКИ РОССИИ**

**Санкт-Петербургский государственный**

**электротехнический университет**

**«ЛЭТИ» им. В.И. Ульянова (Ленина)**

**Кафедра САПР**

отчет

**по лабораторной работе №2**

**по дисциплине «Компьютерная графика»**

Тема: **Формирование различных кривых с использованием ортогонального проектирования на плоскость визуализации**

|  |  |  |
| --- | --- | --- |
| Студенты гр. 8362 |  | Ларионова Е.Е.  Матвеев Н.Д. |
| Преподаватель |  | Матвеева И. В. |

Санкт-Петербург

2021

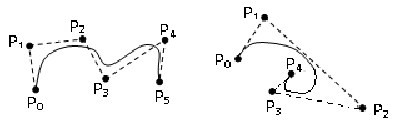
**ЗАДАНИЕ**

Сформировать на плоскости кривую Безье на основе задающей ломаной, определяемой 3 и большим количеством точек. Обеспечить редактирование координат точек задающей ломаной с перерисовкой сплайна Безье.

**ТЕОРЕТИЧЕСКАЯ ЧАСТЬ**

Кривые Безье. Их формирование и особенности.

Точки задания этих кривых только определяют ход кривой, сама строящаяся кривая в общем случае не проходит через внутренние точки задающего многоугольника:



Особенности:

1. Подходит по касательной к внешним ребрам (сторонам) задающего многоугольника, а остальные точки определяют ход кривой. Они позволяют качественно оценить ход кривой в зависимости от вида задающего многоугольника.

2. Кривая задается параметрически в функции от независимого параметра.

3. Это кривая n-ой степени, т.е. сколько ребер у задающего многоугольника – такой степени и получается кривая. Влиять на степень кривой можно только изменением количества задающих ее точек.

Математически такая кривая описывается параметрическим уравнением:

,

где Р(t) – полиномиальная функция,

Рi –координаты i-ой точки задания,

Ni, n – весовой коэффициент i-той вершины (коэффициент бинома Ньютона),

i – номер точки,

n – количество сторон задающего многоугольника

t – задающий параметр, причем 0 <= t <= 1

**РЕАЛИЗАЦИЯ ПРОГРАММЫ**

При запуске программы открываются 2 окна «MainWindow» и «Form». В «MainWindow» задаются координаты точек. В «Form» служит для отрисовки сплайна Безье (Рисунок 1 и Рисунок 2).

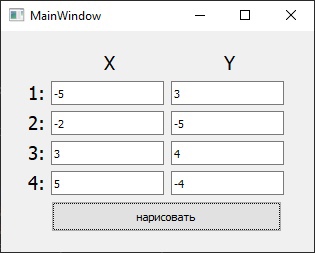


Рисунок 1 – Окно «MainWindow»

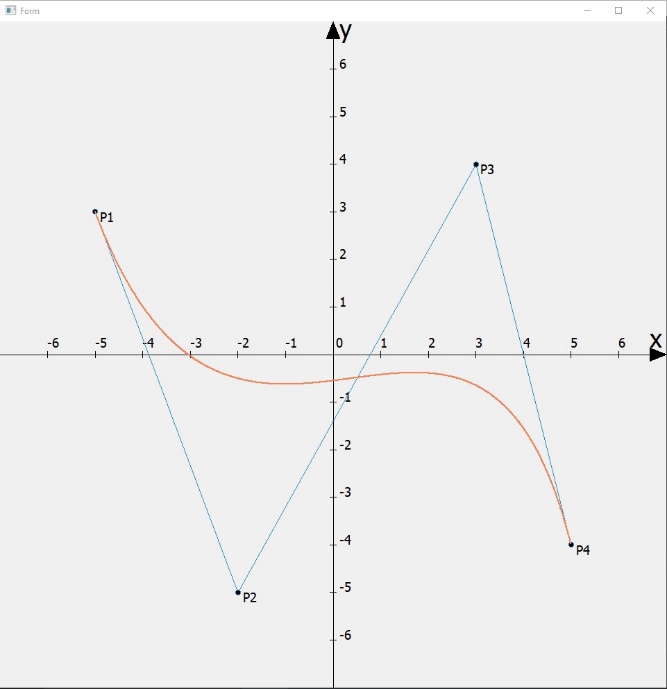


Рисунок 1 – Окно «Form»

**ПРИЛОЖЕНИЕ 1 – КОД ПРОГРАММЫ**

Файл main.cpp

#include <application.h>

int main(int argc, char \*argv[])

{

Application a(argc, *argv*);

return a.exec();

}

Файл mainwindow.cpp

#include "mainwindow.h"

#include "ui\_mainwindow.h"

MainWindow::**MainWindow**(QWidget \*parent)

: QMainWindow(*parent*)

, ui(new Ui::MainWindow)

{

ui->setupUi(this);

}

MainWindow::~***MainWindow***()

{

delete ui;

}

void MainWindow::**on\_pushButton\_clicked**()

{

QPointF \*p = new QPointF[4];

p[0].setX(ui->lineEdit\_x\_1->text().toDouble());

p[0].setY(ui->lineEdit\_y\_1->text().toDouble());

p[1].setX(ui->lineEdit\_x\_2->text().toDouble());

p[1].setY(ui->lineEdit\_y\_2->text().toDouble());

p[2].setX(ui->lineEdit\_x\_3->text().toDouble());

p[2].setY(ui->lineEdit\_y\_3->text().toDouble());

p[3].setX(ui->lineEdit\_x\_4->text().toDouble());

p[3].setY(ui->lineEdit\_y\_4->text().toDouble());

emit send\_control(*p*);

}

Файл drawwindow.cpp

#include "mainwindow.h"

#include "ui\_mainwindow.h"

MainWindow::**MainWindow**(QWidget \*parent)

: QMainWindow(*parent*)

, ui(new Ui::MainWindow)

{

ui->setupUi(this);

}

MainWindow::~***MainWindow***()

{

delete ui;

}

void MainWindow::**on\_pushButton\_clicked**()

{

QPointF \*p = new QPointF[4];

p[0].setX(ui->lineEdit\_x\_1->text().toDouble());

p[0].setY(ui->lineEdit\_y\_1->text().toDouble());

p[1].setX(ui->lineEdit\_x\_2->text().toDouble());

p[1].setY(ui->lineEdit\_y\_2->text().toDouble());

p[2].setX(ui->lineEdit\_x\_3->text().toDouble());

p[2].setY(ui->lineEdit\_y\_3->text().toDouble());

p[3].setX(ui->lineEdit\_x\_4->text().toDouble());

p[3].setY(ui->lineEdit\_y\_4->text().toDouble());

emit send\_control(*p*);

}

Файл application.cpp

#include "application.h"

#define T\_STEP 0.001;

Application::**Application**(int argc, char \*argv[])

: QApplication(*argc*,*argv*)

{

m = new MainWindow;

m->show();

d = new DrawWindow;

d->show();

connect(m,SIGNAL(send\_control(QPointF\*)),

this,SLOT(recieve\_control(QPointF\*)));

connect(this,SIGNAL(send\_state(State\*)),

d,SLOT(recive\_state(State\*)));

}

void Application::**recieve\_control**(QPointF\* p)

{

qreal step = T\_STEP;

State s;

s.d = p;

s.max = 5;

for (size\_t i = 0; i < 4; i++)

{

unsigned int tmp = static\_cast<unsigned int>(qAbs(p[i].x())>qAbs(p[i].y())?qAbs(p[i].x()):qAbs(p[i].y()));

s.max = s.max>tmp?s.max:tmp;

}

for (size\_t i = 0; i\*step < 1; i++)

{

s.p.push\_back(sub\_calc(*p*,i\*step));

}

emit send\_state(*&s*);

}

QPointF Application::**sub\_calc**(QPointF\* p, qreal t)

{

return qPow(1-t,3)\*p[0] + 3\*qPow(1-t,2)\*t\*p[1] + 3\*(1-t)\*qPow(t,2)\*p[2] + qPow(t,3)\*p[3];

}