Московский авиационный институт

(Национальный исследовательский университет)

**Институт «Информационных технологий и прикладной математики»**

**Лабораторная работа №7**

Построение плоских полиномиальных кривых

Работу выполнил:

Рябыкин Алексей Сергеевич

Группа: М8О-309Б-18

Преподаватель: Филиппов Г.С.

Оценка:

Дата:

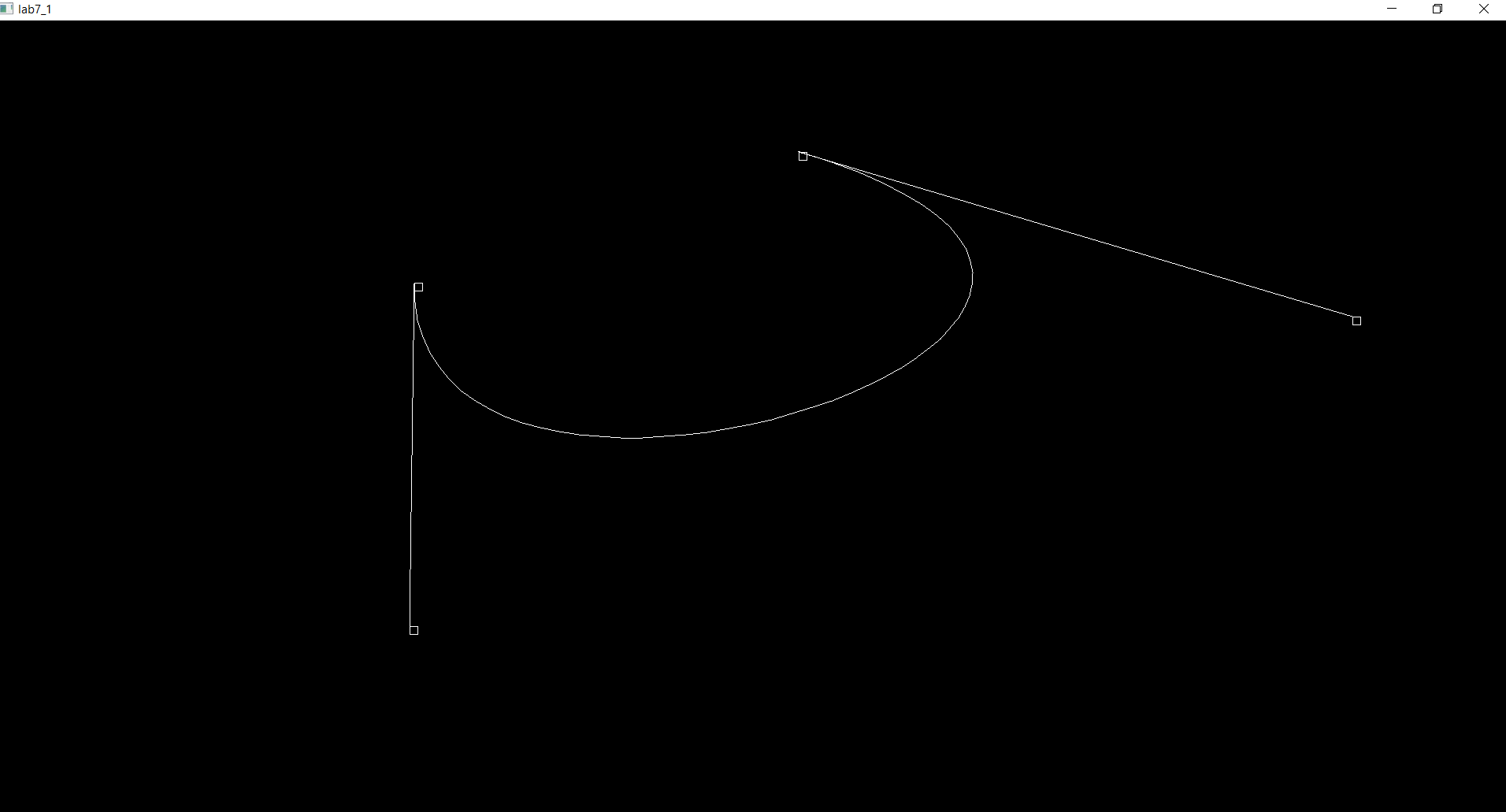
**Москва, 2020**

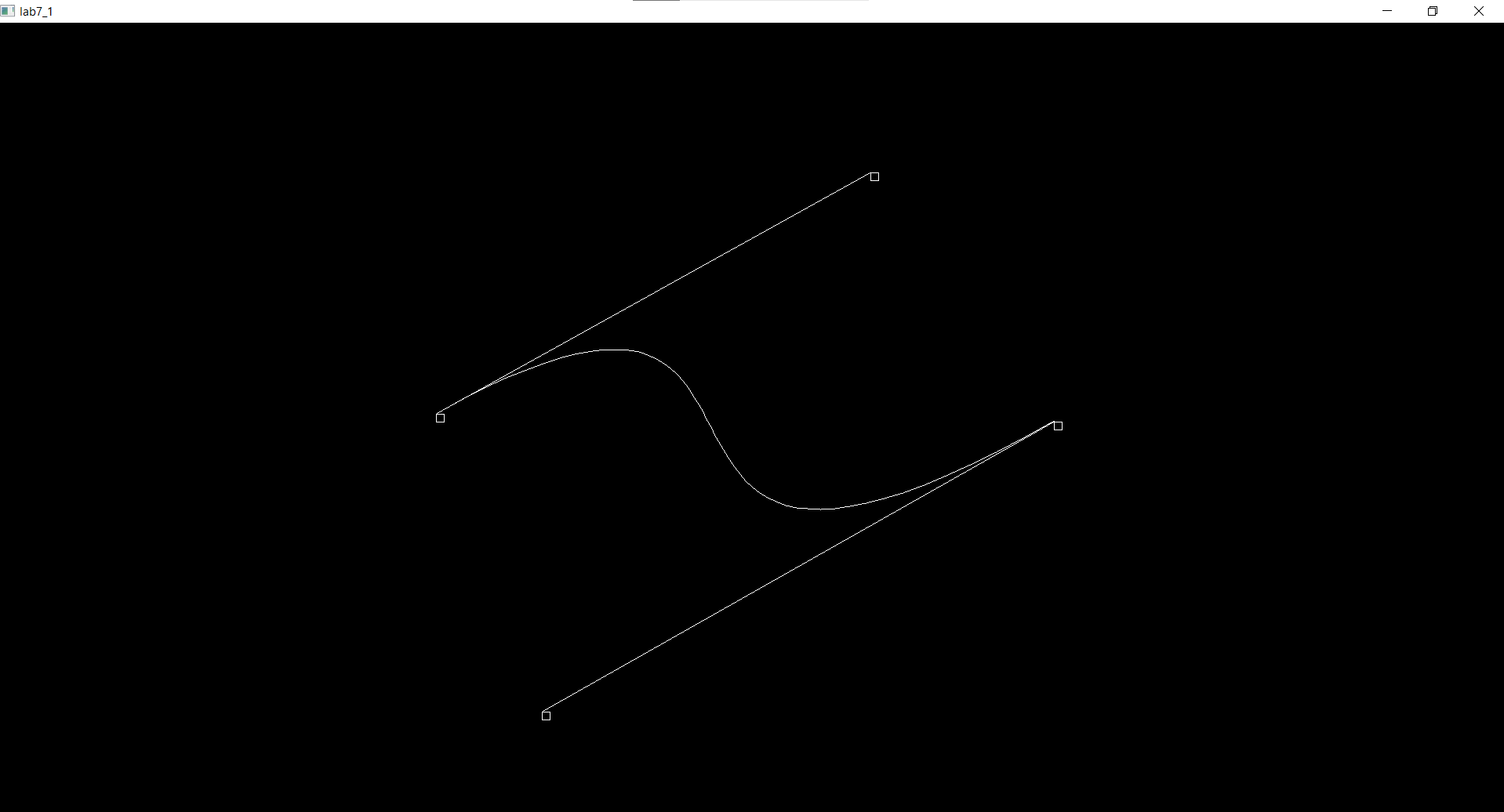
**Постановка задачи**

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

**Вариант многогранника: 4. Кривая Безье 3-го порядка**

**С**криншоты работы программы:

****

****

Фрагменты кода:

Отрисовка кривой:

void bezie::draw(QPainter \*pen){

pen->setPen(QColor(255, 255, 255, 255));

double t = 0;

double P1\_x = (1-t)\*(1-t)\*(1-t) \* Point1.x();

double P1\_y = (1-t)\*(1-t)\*(1-t) \* Point1.y();

double P2\_x = 3 \* (1-t)\*(1-t) \* t \* Point2.x();

double P2\_y = 3 \* (1-t)\*(1-t) \* t \* Point2.y();

double P3\_x = 3 \* (1-t)\* t\*t \* Point3.x();

double P3\_y = 3 \* (1-t)\* t\*t \* Point3.y();

double P4\_x = t\*t\*t \* Point4.x();

double P4\_y = t\*t\*t \* Point4.y();

double x\_prev = P1\_x + P2\_x + P3\_x + P4\_x;

double y\_prev = P1\_y + P2\_y + P3\_y + P4\_y;

double speed = 0.02;

*while* (t<1){

t += speed;

P1\_x = (1-t)\*(1-t)\*(1-t) \* Point1.x();

P1\_y = (1-t)\*(1-t)\*(1-t) \* Point1.y();

P2\_x = 3 \* (1-t)\*(1-t) \* t \* Point2.x();

P2\_y = 3 \* (1-t)\*(1-t) \* t \* Point2.y();

P3\_x = 3 \* (1-t)\* t\*t \* Point3.x();

P3\_y = 3 \* (1-t)\* t\*t \* Point3.y();

P4\_x = t\*t\*t \* Point4.x();

P4\_y = t\*t\*t \* Point4.y();

double x = P1\_x + P2\_x + P3\_x + P4\_x;

double y = P1\_y + P2\_y + P3\_y + P4\_y;

pen->drawLine(*static\_cast*<int>(x\_prev),*static\_cast*<int>(y\_prev),*static\_cast*<int>(x),*static\_cast*<int>(y));

x\_prev = x;

y\_prev = y;

*//pen->drawEllipse(static\_cast<int>(x),static\_cast<int>(y),1,1);*

}

}

Отрисовка точек(+вызов функции отрисовки кривой) и работа с точками:

void view::*paintEvent*(QPaintEvent \*)

{

QPainter ptr{*this*};

ptr.setPen(QColor(255, 255, 255, 255));

*if* (!flag){

*if* (flag1) {

ptr.drawRect(*static\_cast*<int>(mPoint1.x()),*static\_cast*<int>(mPoint1.y()), 10, 10);

}

*if* (flag2) {

ptr.drawRect(*static\_cast*<int>(mPoint2.x()), *static\_cast*<int>(mPoint2.y()), 10, 10);

}

*if* (flag3) {

ptr.drawRect(*static\_cast*<int>(mPoint3.x()), *static\_cast*<int>(mPoint3.y()), 10, 10);

}

*if* (flag4) {

ptr.drawRect(*static\_cast*<int>(mPoint4.x()), *static\_cast*<int>(mPoint4.y()), 10, 10);

}

}

*else* {

ptr.drawRect(*static\_cast*<int>(mPoint1.x()),*static\_cast*<int>(mPoint1.y()), 10, 10);

ptr.drawRect(*static\_cast*<int>(mPoint4.x()), *static\_cast*<int>(mPoint4.y()), 10, 10);

ptr.drawRect(*static\_cast*<int>(mPoint2.x()), *static\_cast*<int>(mPoint2.y()), 10, 10);

ptr.drawRect(*static\_cast*<int>(mPoint3.x()), *static\_cast*<int>(mPoint3.y()), 10, 10);

ptr.drawLine(*static\_cast*<int>(mPoint1.x()), *static\_cast*<int>(mPoint1.y()),

*static\_cast*<int>(mPoint2.x()), *static\_cast*<int>(mPoint2.y()));

*//* *ptr.drawLine(static\_cast<int>(mPoint2.x()),* *static\_cast<int>(mPoint2.y()),*

*//* *static\_cast<int>(mPoint3.x()),* *static\_cast<int>(mPoint3.y()));*

ptr.drawLine(*static\_cast*<int>(mPoint3.x()), *static\_cast*<int>(mPoint3.y()),

*static\_cast*<int>(mPoint4.x()), *static\_cast*<int>(mPoint4.y()));

update();

bezie fig1 = bezie(mPoint1, mPoint2, mPoint3, mPoint4);

fig1.draw(&ptr);

*//fig1.draw();*

}

}

void view::*mouseMoveEvent*(QMouseEvent \*e)

{

*if* (flag1 && e->pos().x() <= mPoint1.x() + 10 && e->pos().x() >= mPoint1.x() - 10 &&

e->pos().y() <= mPoint1.y() + 10 && e->pos().y() >= mPoint1.y() - 10) {

QPointF newPoint = e->pos();

double deltaX = newPoint.x() - mPoint1.x();

double deltaY = newPoint.y() - mPoint1.y();

mPoint1.rx() += deltaX;

mPoint1.ry() += deltaY;

update();

*return*;

}

*if* (flag2 && e->pos().x() <= mPoint2.x() + 10 && e->pos().x() >= mPoint2.x() - 10 &&

e->pos().y() <= mPoint2.y() + 10 && e->pos().y() >= mPoint2.y() - 10) {

QPointF newPoint = e->pos();

double deltaX = newPoint.x() - mPoint2.x();

double deltaY = newPoint.y() - mPoint2.y();

mPoint2.rx() += deltaX;

mPoint2.ry() += deltaY;

update();

*return*;

}

*if* (flag3 && e->pos().x() <= mPoint3.x() + 10 && e->pos().x() >= mPoint3.x() - 10 &&

e->pos().y() <= mPoint3.y() + 10 && e->pos().y() >= mPoint3.y() - 10) {

QPointF newPoint = e->pos();

double deltaX = newPoint.x() - mPoint3.x();

double deltaY = newPoint.y() - mPoint3.y();

mPoint3.rx() += deltaX;

mPoint3.ry() += deltaY;

update();

*return*;

}

*if* (flag2 && e->pos().x() <= mPoint4.x() + 10 && e->pos().x() >= mPoint4.x() - 10 &&

e->pos().y() <= mPoint4.y() + 10 && e->pos().y() >= mPoint4.y() - 10) {

QPointF newPoint = e->pos();

double deltaX = newPoint.x() - mPoint4.x();

double deltaY = newPoint.y() - mPoint4.y();

mPoint4.rx() += deltaX;

mPoint4.ry() += deltaY;

update();

*return*;

}

}

**Среда разработки:** Qt Creator 4.10.1

**Вывод:** В процессе выполнения лабораторной работы научился отрисовывать кривую Безье по заданным (с помощью клика мыши) на экран точкам.