Московский авиационный институт

(Национальный исследовательский университет)

Факультет: «Информационные технологии и прикладная математика»

Кафедра: «Вычислительная математика и программирование»

Дисциплина: «Компьютерная графика»

Лабораторная работа №7 по курсу «Компьютерная графика»

Тема: Построение плоских полиномиальных кривых.

Студент: Тимофеев А. В.

Преподаватель: Морозов А. В.

Группа: М80-307Б

Дата:

Оценка:

Подпись:

Москва, 2022

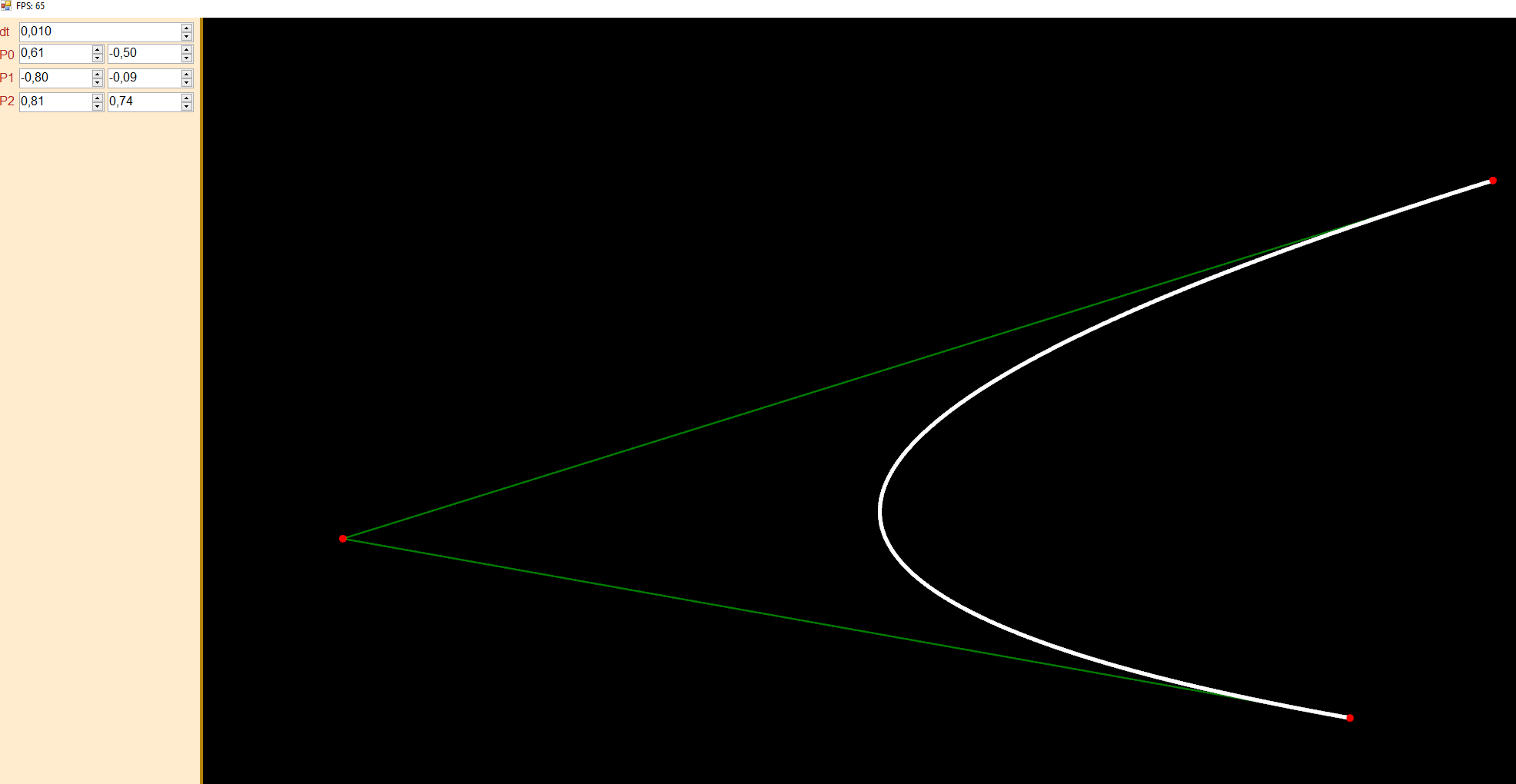
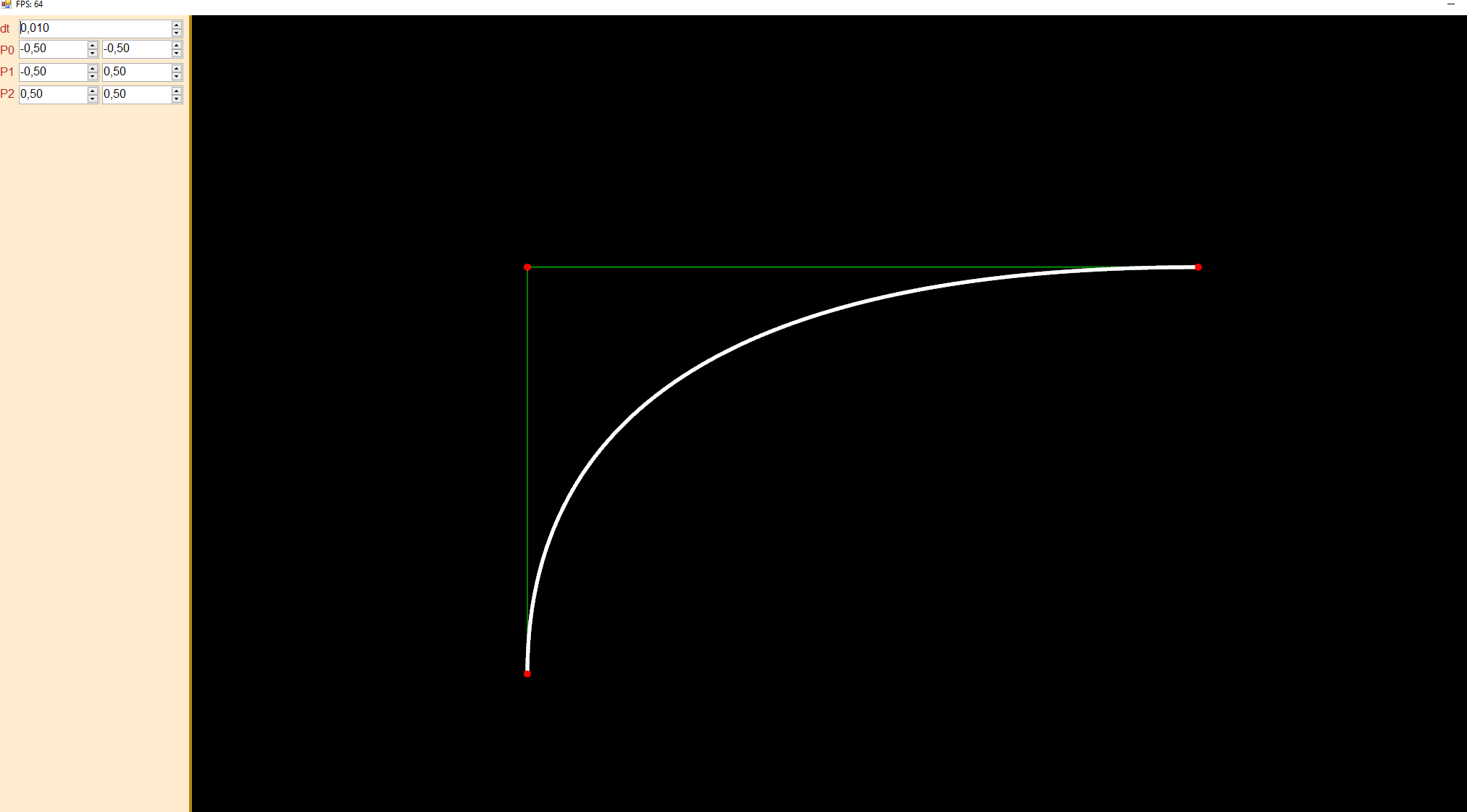
**Постановка задачи**

**Задание**: Написать программу, строящую полиномиальную кривую по заданным точкам. Обеспечить возможность изменения позиции точек и, при необходимости, значений касательных векторов и натяжения.  
Вариант: 5. Кривая Безье 2-й степени.

**Решение задачи**

Кривая Безье 2-й степени строится по 3 точкам и имеет вид: B(t) = (1 − t) 2P0 + 2t(1 − t)P1 + t 2P2, где P0, P1, P2 - точки, а t - параметр от 0 до 1. Для построения составной прямой нужно чередовать точки, то есть брать первую вторую третью, третью чевертую пятую и т.д. Также есть возможность двигать ближайшие точки, для это надо пройтись по всем точкам и посчитать расстояние от них до позиции мыши, выбрав ту, у которой это расстояние будет наименьшим.

**Пример работы**



**Листинг программы**

**Program.cs**

protected override unsafe void OnDeviceUpdate(object s, DeviceArgs e){

if (curve == null) return;

var gl = e.gl;

gl.Clear(OpenGL.GL\_COLOR\_BUFFER\_BIT |OpenGL.GL\_DEPTH\_BUFFER\_BIT |OpenGL.GL\_STENCIL\_BUFFER\_BIT);

gl.LoadIdentity();

gl.MatrixMode(OpenGL.GL\_PROJECTION);

var t = GetTranslateMat();// Получаем матрицу преобразований

curve.ApplyTransform(t);

gl.LineWidth(2f);// Касательные

gl.Begin(OpenGL.GL\_LINE\_STRIP);

gl.Color(0.0f, 0.5f, 0.0f);

foreach (var d in curve.Dots.Select(x => x.pointInWorld)) gl.Vertex(d.X, d.Y);

gl.End();

gl.LineWidth(5f);// Кривая Безье

gl.Begin(OpenGL.GL\_LINE\_STRIP);

gl.Color(1.0f, 1.0f, 1.0f);

foreach (var p in curve.Points.Select(x => x.pointInWorld)) gl.Vertex(p.X, p.Y);

gl.End();

gl.PointSize(dotRadius);// Точки P0, P1, P2

gl.Begin(OpenGL.GL\_POINTS);

gl.Color(1.0f, 0.0f, 0.0f);

foreach (var d in curve.Dots.Select(x => x.pointInWorld)) gl.Vertex(d.X, d.Y);

gl.End(); }

**Bezier2Curve.cs**

public class Bezier2Curve{

public Vertex P0, P1, P2;

public Vertex[] Points;

public Bezier2Curve(DVector2 p0, DVector2 p1, DVector2 p2, double dt){

P0 = new Vertex(p0);

P1 = new Vertex(p1);

P2 = new Vertex(p2);

Points = new Vertex[(int) (1 / dt) + 2];

var i = 0;

for (var t = 0.0; t <= 1; t += dt, i++) Points[i] = new Vertex(Bezier2(p0, p1, p2, t));

Points[i] = new Vertex(Bezier2(p0, p1, p2, 1.0));

}

public void ApplyTransform(DMatrix3 t){// Вершины

foreach (var p in Points) p.pointInWorld = t \* p.pointInLocalSpace;

P0.pointInWorld = t \* P0.pointInLocalSpace;

P1.pointInWorld = t \* P1.pointInLocalSpace;

P2.pointInWorld = t \* P2.pointInLocalSpace;

}

private DVector2 Bezier2(DVector2 p0, DVector2 p1, DVector2 p2, double t){

return (1.0 - t) \* (1.0 - t) \* p0 + 2 \* t \* (1.0 - t) \* p1 + t \* t \* p2;

}

public IEnumerable<Vertex> Dots{

get{ return new Vertex[]{P0, P1, P2}; }

}

}

public class Vertex{

public readonly DVector3 pointInLocalSpace;

public DVector3 pointInWorld;

public Vertex(DVector2 v){

pointInLocalSpace = new DVector3(v, 1.0);

}

}

}

**Вывод**

В результате выполнения данной лабораторной работы я познакомился построением плоских полиномиальных кривых с использованием технологий OpenGL.