

**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: «Формирования различных кривых с использованием**  
**ортогонального проектирования на плоскость визуализации (экране**  
**дисплея)»**

Студент гр. 8374	_____	Адаменко Е.А.
Студент гр. 8374	_____	Зелинский М.В.
Студент гр. 8374	_____	Стрелков А.Н.
Преподаватель	_____	Матвеева И.В.

Санкт-Петербург

2021

## Цель работы

Сформировать различные кривые с использованием ортогонального проектирования на плоскость визуализации (экране дисплея). Задание 5: 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.

## Теоретические положения

В качестве языка программирования был выбран JavaScript. Для формирования интерфейса был использован фреймворк jquery, а также язык гипер разметки HTML5 с использованием CSS.

Наше 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.

Кривые Безье.

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

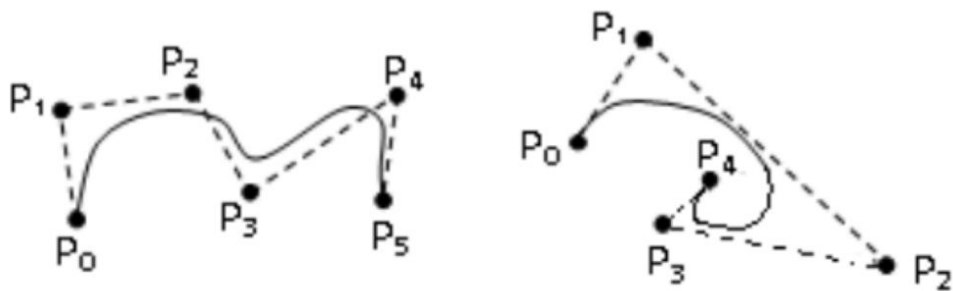


Рис.1

Их 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(рис 2)

$$P(t) = \sum_{i=0}^n P_i \times N_{i,n}(t) \text{ , где } P(t) \text{ – полиномиальная функция,}$$

$P_i$  – вес (координаты)  $i$ -ой точки задания,  
 $N_{i,n}$  – весовой коэффициент  $i$ -той вершины,  
 $i$  – номер вершины (точки),  
 $n$  – количество сторон задающего многоугольника  
 $t$  – задающий параметр, причем  $0 \leq t \leq 1$

$$N_{i,n}(t) = \frac{n!}{i!(n-i)!} \times t^i \times (1-t)^{n-i} \text{ , где } n! = 5*4*3*2*1, \quad 0! = 1$$

Рис.2

## Описание интерфейса

Для запуска web-приложения необходимо запустить файл.html

Для взаимодействия используется 2 кнопки ( Нарисовать кривую, Очистить доску) и сама доска(рабочая область)



Рис. 3 Интерфейс



Рис. 4 Постановка точек

Для постановления точек необходимо ЛКМ установить точки на рабочей области, далее нажать кнопку “Нарисовать кривую”

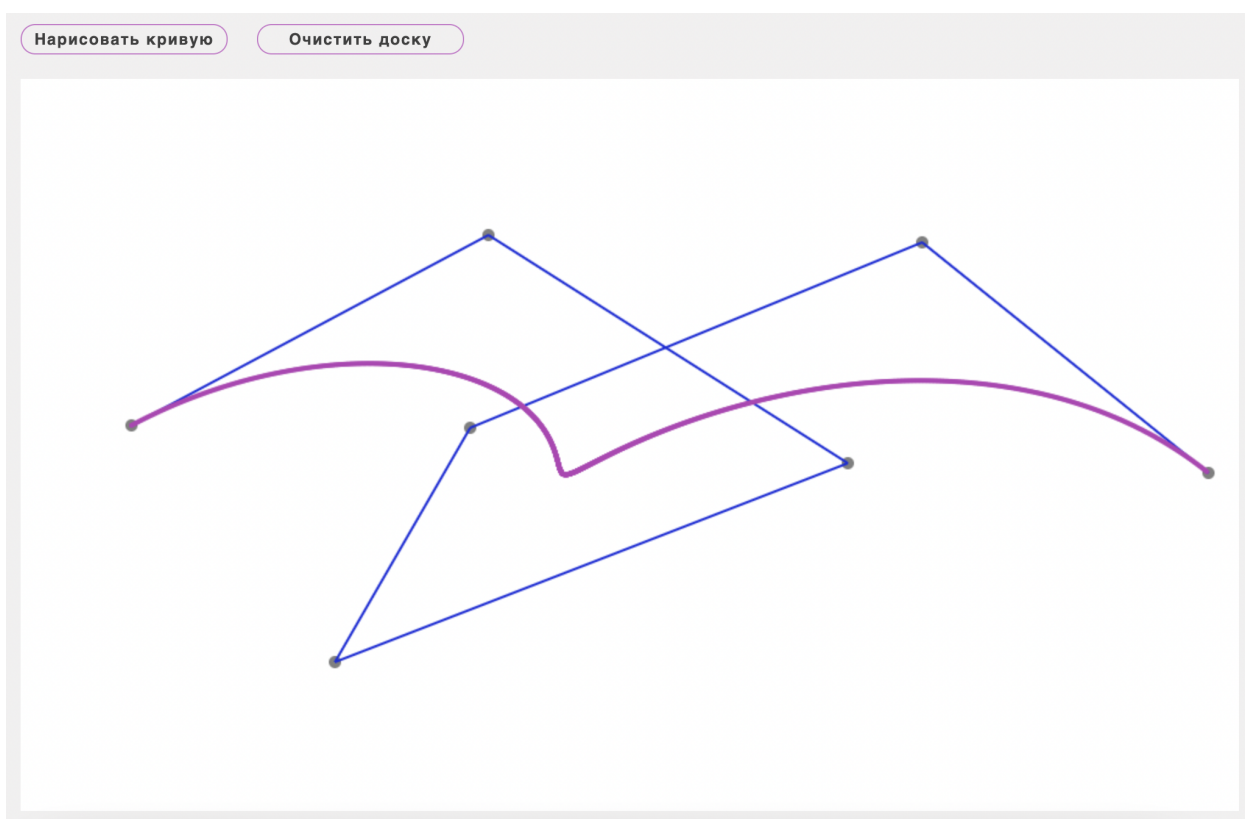


Рис. 5 Изображение кривой

## Снимки экрана

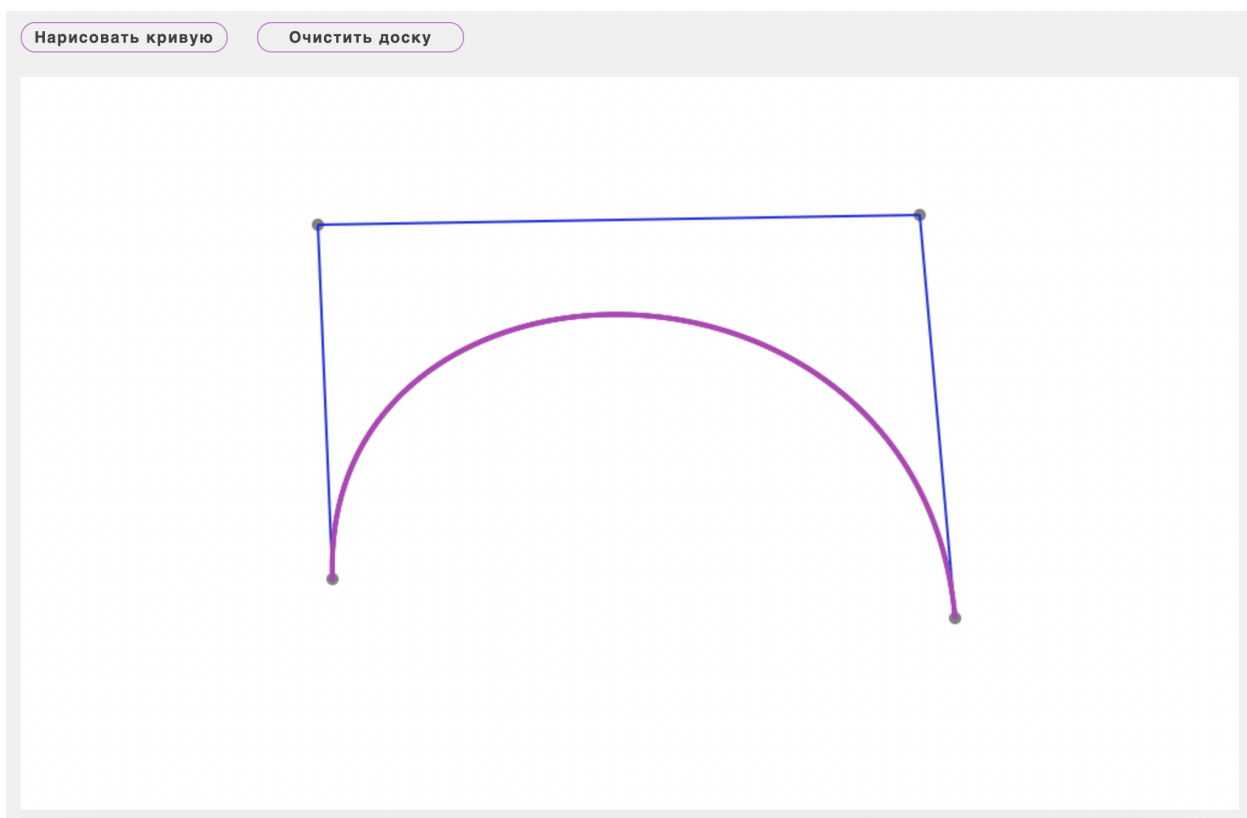


Рис. 6 Пример 1

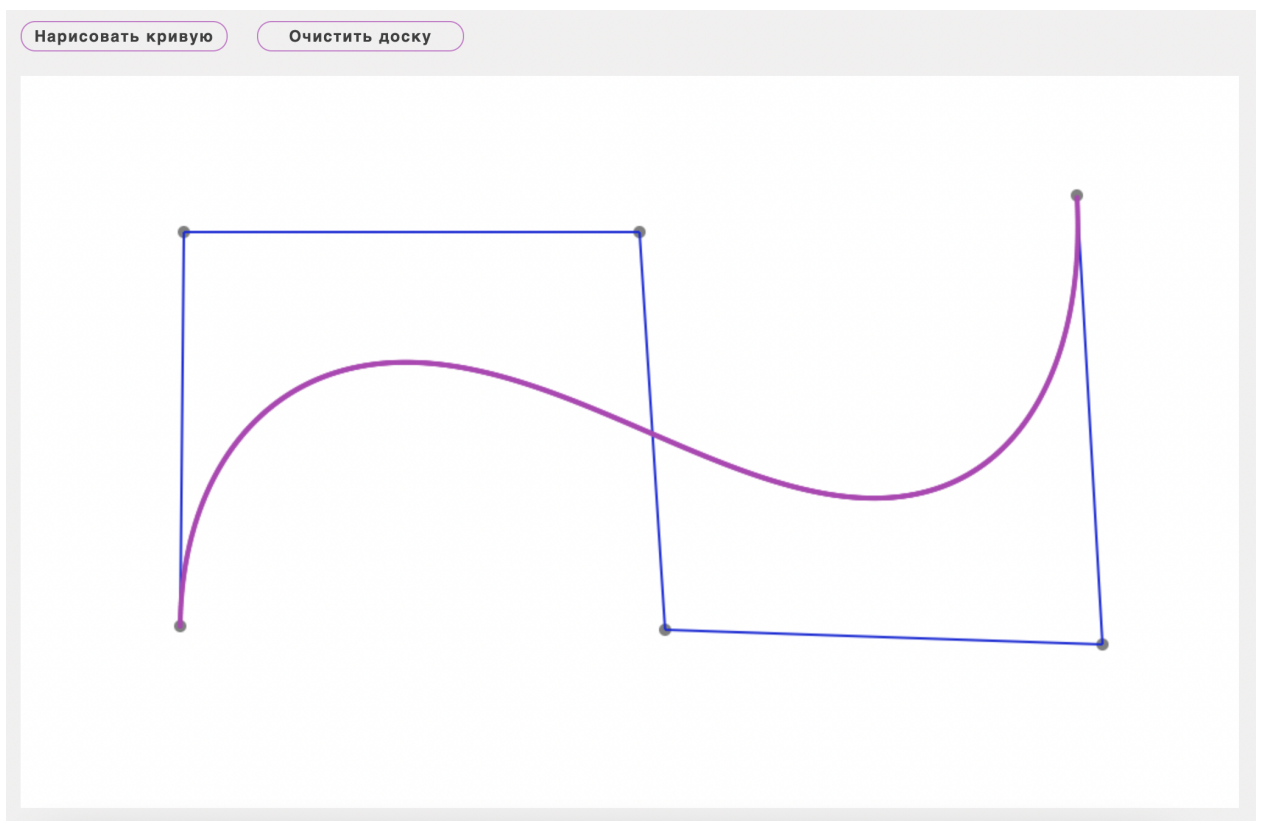


Рис. 7 Пример 2

[Ссылка на видео](#)

### Код программы

```
<script>
  var points = [];
  var knots = [];
  var degree = 0;
  var isCanvasUsingNow = false

  function clear() {
    location.reload();
  }

  function clearCanvasWindow() {
    const context = canvas.getContext('2d');
    context.clearRect(0, 0, canvas.width, canvas.height);
    points = [];
    knots = [];
    degree = 0;
    isCanvasUsingNow = false
  }

  $('#canvas').click(function (e) {
    var point = setPointFromUser(e);
    points.push(point);
```

```

    drawBasePoint(point, 'grey'); // grey
  });

function setPointFromUser(event) {
  if (isCanvasUsingNow){
    alert('Сначала нужно очистить доску!')
  }
  else{
    var rect = canvas.getBoundingClientRect();

    var x = event.clientX - rect.left;
    var y = event.clientY - rect.top;

    var point = new Point(x, y);

    Point(point);
    return point;
  }
}

function drawBezierCurve(point1, point2, point3) {

  for (var i = 0; i < points.length - 1; i++) {
    var ctx = document.getElementById('canvas').getContext('2d');
    ctx.beginPath();
    ctx.strokeStyle = '#1a2edb';
    ctx.lineWidth = 2;
    ctx.moveTo(points[i].x, points[i].y);
    ctx.lineTo(points[i+1].x, points[i+1].y);
    ctx.stroke();
  }

  if (points.length >= 3){
    var t = 0;
    var inter = setInterval(function () {

      var mainBezierPoint = findBezierPointOfPoints([...points], t);
      drawPoint(mainBezierPoint, '#b640b6'); // purple
      t = t + 0.001;
      if (t > 1) {
        clearInterval(inter);
      }
    });
    isCanvasUsingNow = true
  }
  else alert('Введено меньше 3 точек')
}

```



```

function findBezierPointOfPoints(points, t) {
    if (points.length == 1) {
        return points[0];
    }
    var tempPoint = [];
    for (var i = 0; i < points.length - 1; i++) {
        tempPoint.push(findBezierPoint(points[i], points[i + 1], t));
    }

    return findBezierPointOfPoints([...tempPoint], t);
}

function findBezierPoint(point1, point2, t) {
    var x = (1 - t) * point1.x + t * point2.x;
    var y = (1 - t) * point1.y + t * point2.y;

    var newPoint = new Point(x, y);

    return newPoint;
}

function drawBasePoint(point, color) {
    var ctx = document.getElementById('canvas').getContext('2d');

    ctx.fillStyle = color;

    ctx.beginPath();
    ctx.arc(point.x, point.y, 5, 0, Math.PI * 2, true);

    ctx.fill();
}

function drawPoint(point, color) {
    var ctx = document.getElementById('canvas').getContext('2d');
    ctx.fillStyle = color;
    ctx.beginPath();
    ctx.arc(point.x, point.y, 2, 0, Math.PI * 2, true);
    ctx.fill();
}

function Point(x, y) {
    var obj = {};
    obj.x = x;
    obj.y = y;
    return obj;
}
</script>

```

## Вывод

В ходе выполнения лабораторной работы было изучено формирование различных кривых с использованием ортогонального проектирования на плоскость визуализации (экране дисплея). Сформирована на плоскости кривая Безье на основе задающей ломаной, определяемой 3 и большим количеством точек. Обеспечено редактирование координат точек задающей ломаной с перерисовкой сплайна Безье.