

# 华东师范大学计算机科学与技术实验报告

实验课程：计算机图形学	年级：2018	实验成绩：
实验名称：Bezier曲线	姓名：董辰尧	实验日期：2021/04/25
实验编号：8	学号：10185102144	实验时间：13:00-14:40
指导教师：李洋、王长波	组号：	

## 一、实验目的

练习利用GDI绘制实现基本Bezier曲线。

## 二、实验环境

Visual studio 2017

## 三、实验内容

- 根据控制点生成Bezier曲线点
- 使用生成点绘制小线段

## 四、实验过程与分析

1. 本次实验主要是对Bezier曲线点的实现，算法的精髓在于，对于2个控制点，就只用做一次插值得到曲线点，对于多个控制点，需要做多次插值，每做一次点的个数就减去1，直到点的个数变成1为止，这个点就是曲线点。主要代码如下：

```
1 void draw_bezier(const std::vector<Vector2> &control_points,  
2                 std::vector<std::pair<int, int>> &buffer) {  
3     /*  
4         void draw_line(int src_x, int src_y, int dst_x, int dst_y,  
5             std::vector<std::pair<int, int>> &buffer)  
6     */  
7     // calculate the Bezier curve  
8     // and use draw_line to draw the curve  
9  
10    int size = control_points.size();  
11    Vector2 control_init = control_points[0];  
12    Vector2 controls[100];  
13  
14    for(double t=0; t<1; t+=0.01)  
15    {  
16        for(int j=0; j<size; j++) //遍历每个控制点  
17        {  
18            for(int k=j; k< size; k++)  
19            {
```

```

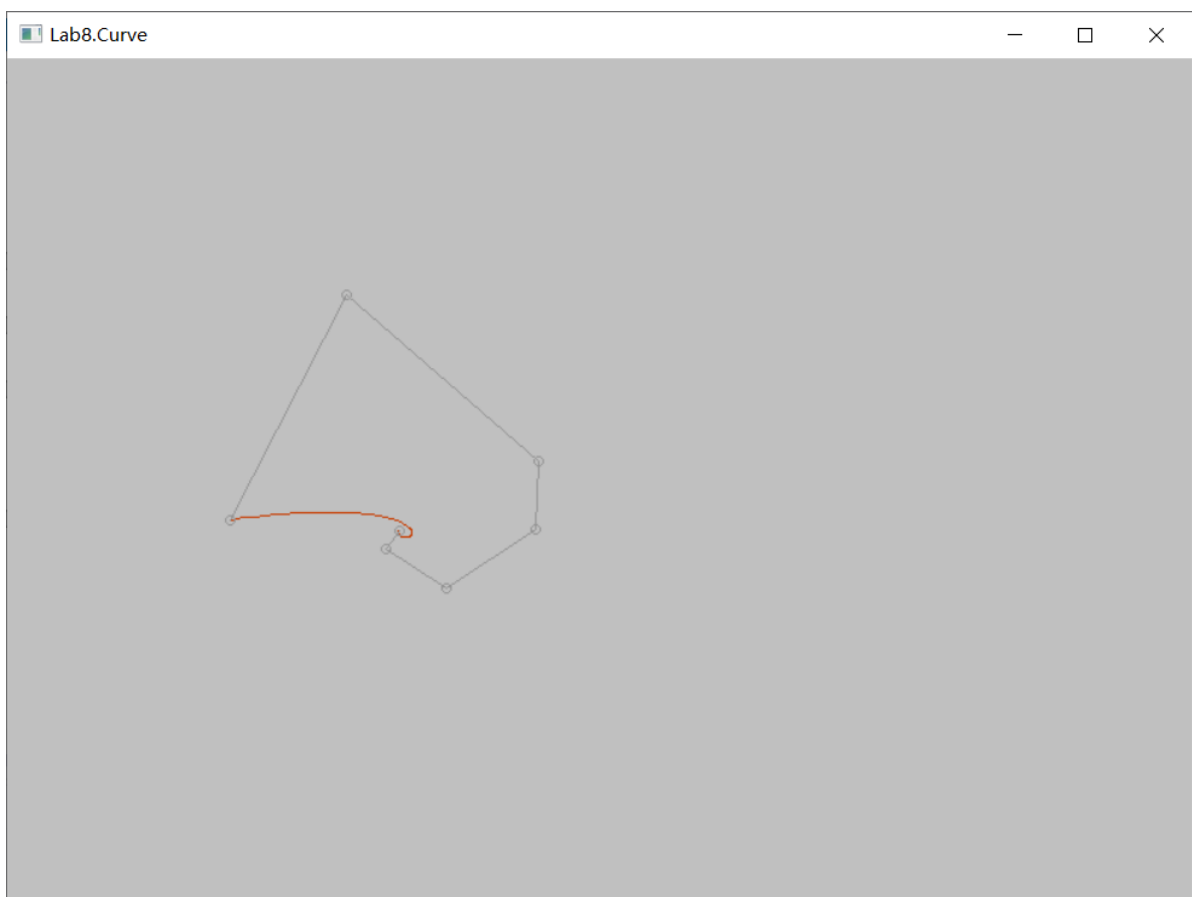
20         if(j == 0)
21             controls[k] = control_points[k];
22         else
23             {
24                 controls[k].x() = t * control_points[k].x() + (1-t) *
controls[k-1].x();
25                 controls[k].y() = t * control_points[k].y() + (1-t) *
controls[k-1].y();
26             }
27         }
28     }
29     draw_line(control_init.x(),control_init.y(),controls[size-
1].x(),controls[size-1].y(),buffer);
30     control_init = controls[size-1];
31 }
32 }
33

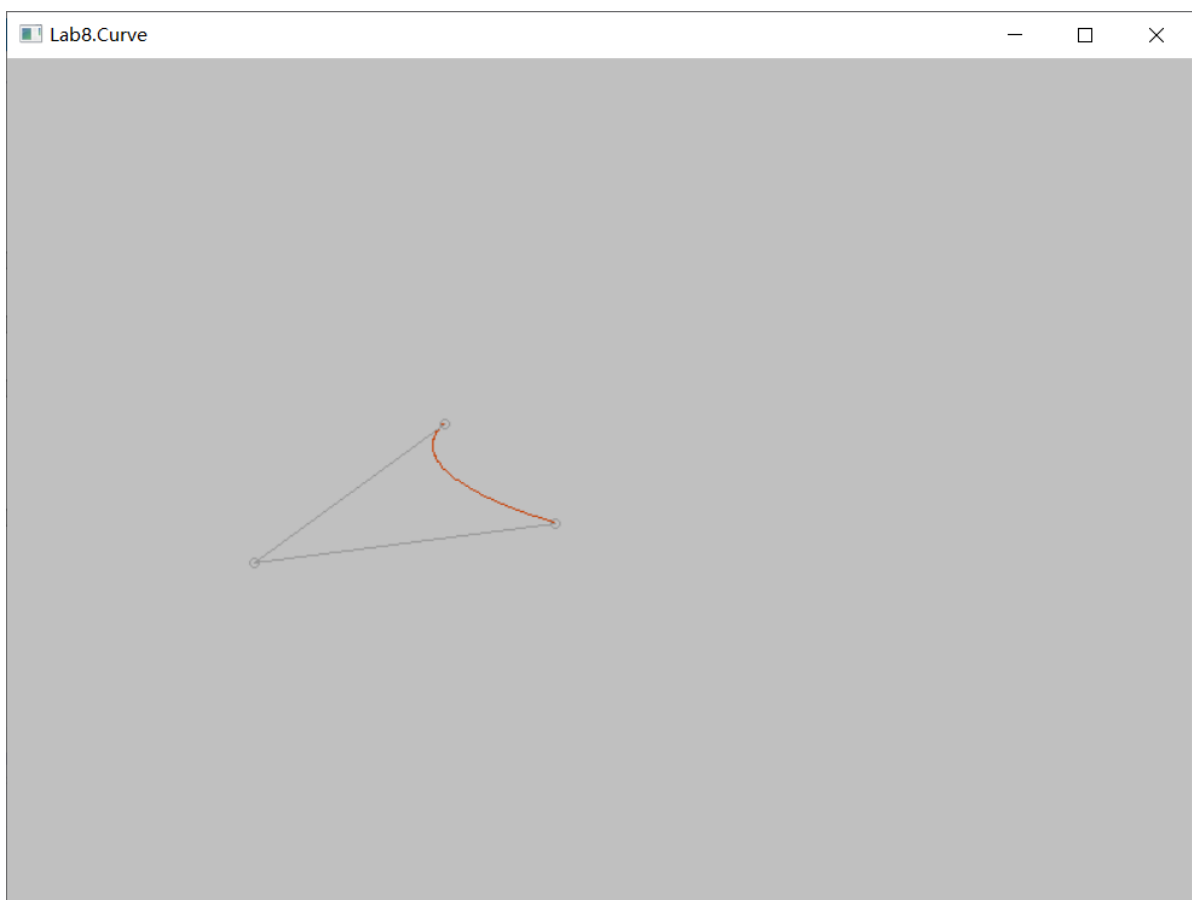
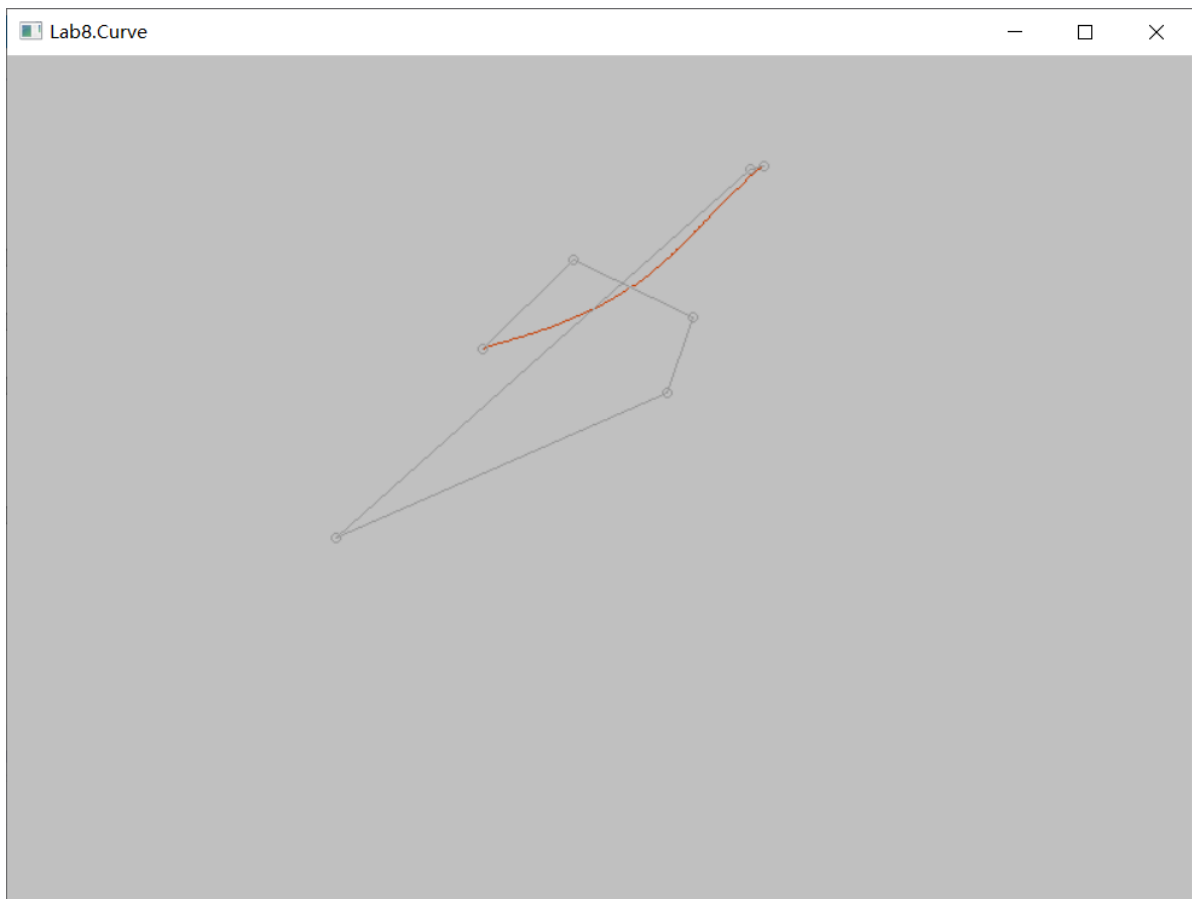
```

2.上面的代码就是直接用公式法做出来了。利用了controls存历史点，最后每次连线 + 更新。

## 五、实验过程总结

截图如下：





总结：本次实验的算法的精神很容易理解，实际操作起来比较难。一开始我用了递归的做法，写了另外一个函数，可是结果不尽如人意。多次debug无果，最后只能放弃。后面我换一种做法，直接用公式就好了。我认为主要难点在于一些c++语言的使用上，可能哪一点就会出错，还有可能是机器不会报出来的错误，所以写的过程还是比较坎坷。

