

Homework 9

此次作业需要实现一个Bezier Curve的绘制工具，它应能够接受鼠标输入。

Bezier Curve

Bezier曲线本质上是由调和函数（Harmonic functions）根据控制点插值生成，其参数方程如下：

$$Q(t) = \sum_{i=0}^n P_i B_{i,n}(t), \quad t \in [0,1]$$

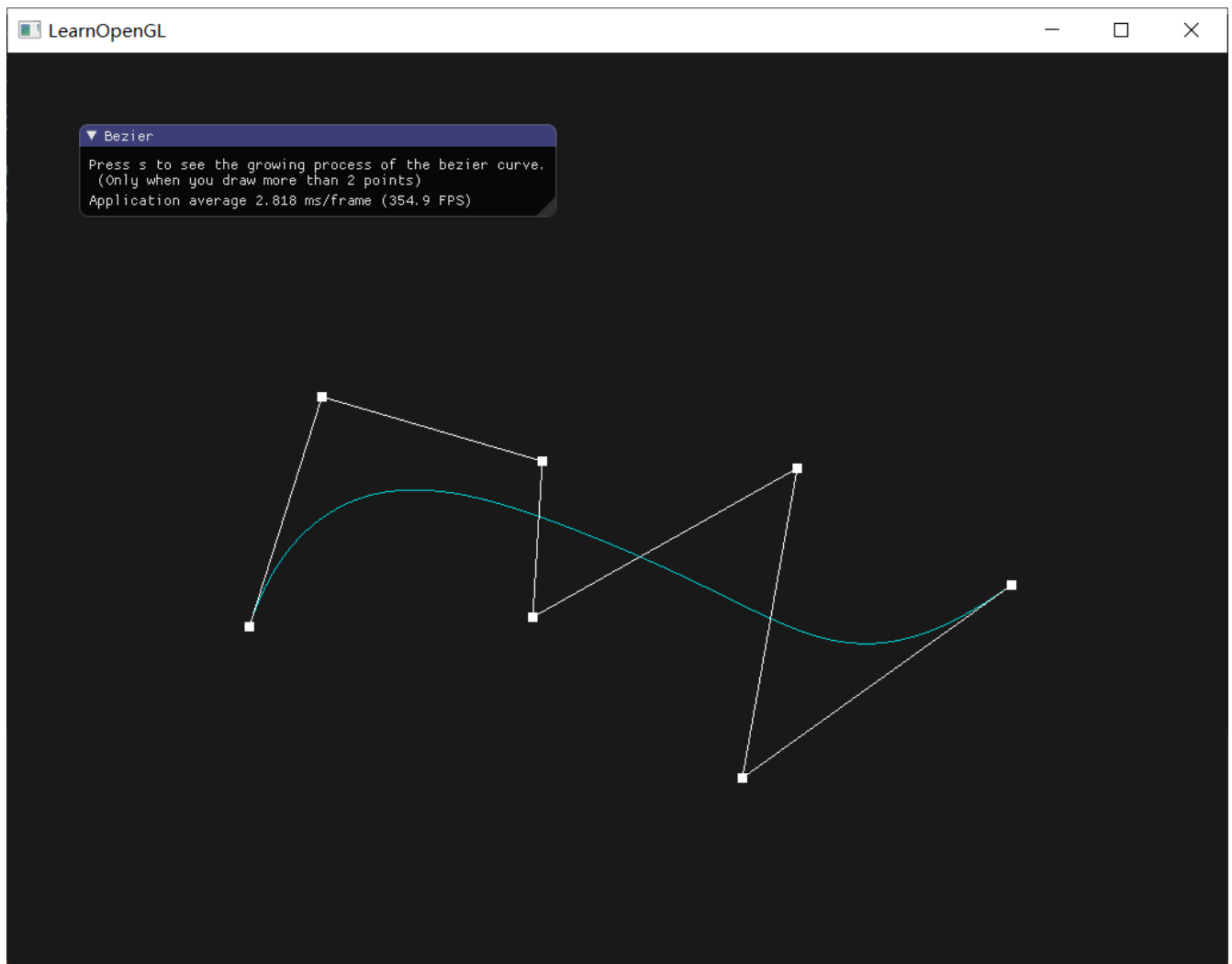
上式为n次多项式，具有n+1项。其中 P_i ($i = 0, 1 \dots n$)表示特征多边形的n+1个顶点向量，在这里就是我们用鼠标标出的n+1个坐标点。 $B_{i,n}(t)$ 为伯恩斯坦（Bernstein）基函数，其多项式表示为：

$$B_{i,n}(t) = \frac{n!}{i!(n-i)!} t^i (1-t)^{n-i}, \quad i=0, 1 \dots n$$

那么绘制方法就是存储鼠标点击的每个点的坐标，代入以上公式进行计算，其中t的值的变化幅度可以自己控制，这个值主要影响绘制的精度。还需注意的是，在opengl中使用 `glfwGetCursorPos` 函数获得的鼠标位置与绘图所需的标准化设备坐标不同，需要做一个转换：

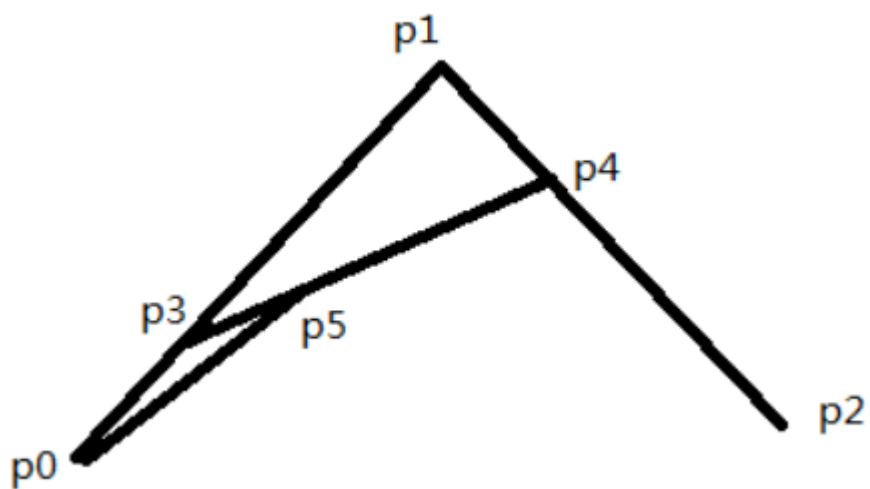
```
// xpos, ypos为glfwGetCursorPos获得的坐标
// SCR_WIDTH, SCR_HEIGHT为窗口宽高
x = -1.0 + 2 * (xpos / SCR_WIDTH)
y = 1.0 - 2 * (ypos / SCR_HEIGHT)
```

实现效果：



动态呈现Bezier曲线

动态呈现Bezier曲线，需要给出其运动路径。以二次方Bezier曲线为例：

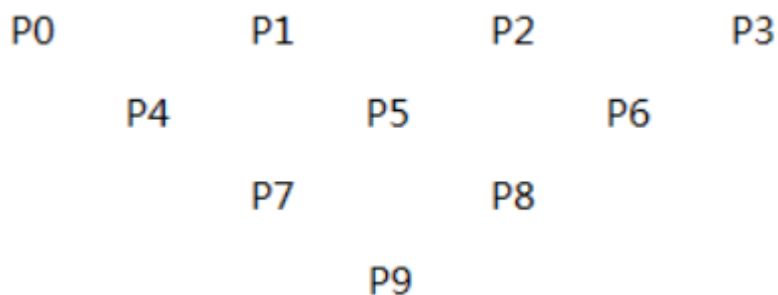


其中P0、P1、P2为鼠标标注的点，P3与P4分别在两边上移动，P5的运动轨迹即为Bezier曲线。这里就是要动态呈现P3、P4与P5的运动轨迹。

很容易得出它们的计算公式：

```
p3 = (1 - t) * p0 + t * p1
p4 = (1 - t) * p1 + t * p2
p5 = (1 - t) * p3 + t * p4
```

当标注点为三个时，首先计算出P3、P4，再计算P5（可看作三层计算）。推广到更多点上也是一样的步骤，只是计算的层数会更多，比如四个标注点时，需要计算：



最终效果：

