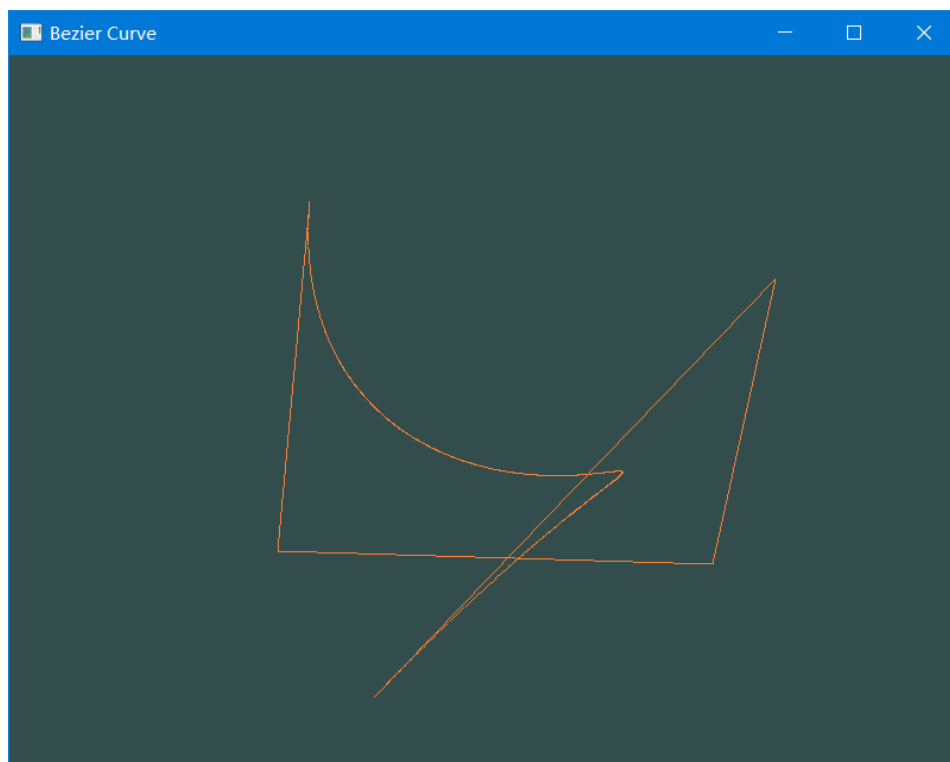
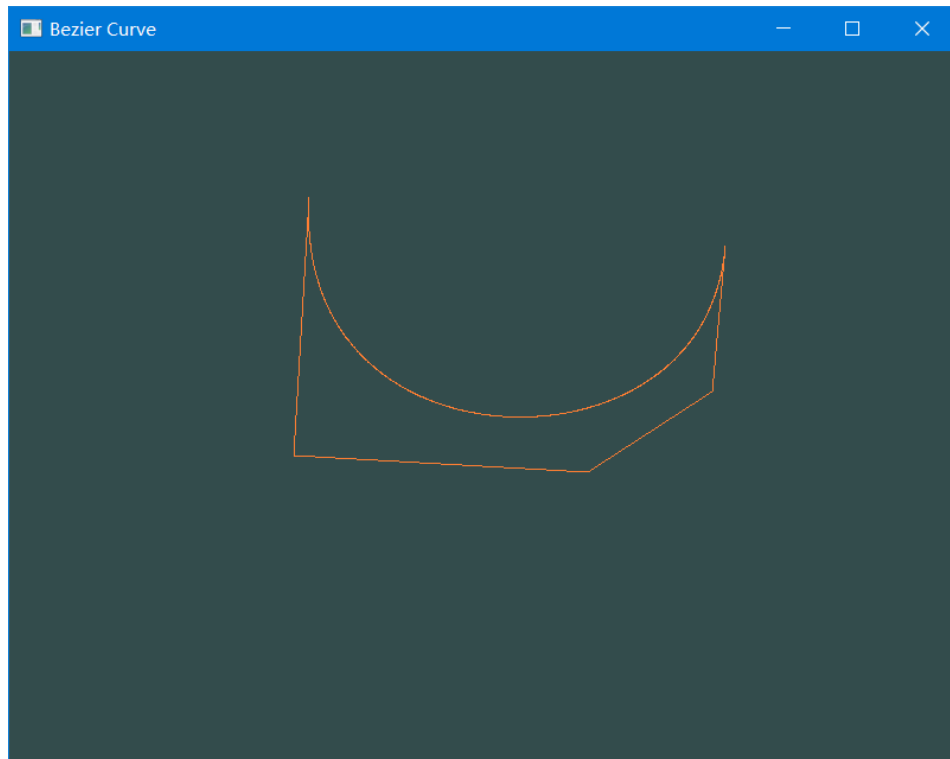


Report

运行结果：



附 gif。

实现思路：

- ① 声明公有变量 xpos、ypos，将获得的鼠标坐标转换为空间坐标后存储，并在鼠标点击的回调函数中实现将坐标增删到 vector 中，并更新数组和顶点数组。这样顶点组只会在有点击事件发生时更改以及重绑定。

```
void curse_poscallback(GLFWwindow *window, double x, double y)
{
    xpos = x / SCR_WIDTH * 2 - 1;
    ypos = y / SCR_HEIGHT * (-2) + 1;
}
```

- ② 实现一个函数，输入行数，返回杨辉三角对应行的数组，作为 Bezier 曲线公式的多项式系数。

```
int *getPara(int n) {
    int i, j, b[100], t[100];
    static int a[100];
    a[0] = 0;
    a[1] = 1;
    for (int i = 1; i <= n; i++) {
        a[i + 1] = 0;
        for (j = 1; j <= i; j++) {
            b[j] = a[j - 1] + a[j];
            t[j] = b[j];
        }
        for (j = 1; j <= i; j++) {
            a[j] = t[j];
        }
    }
    return a;
}
```

- ③ 根据公式

$$B(t) = \sum_{i=0}^n \binom{n}{i} P_i (1-t)^{n-i} t^i = \binom{n}{0} P_0 (1-t)^n t^0 + \binom{n}{1} P_1 (1-t)^{n-1} t^1 + \dots + \binom{n}{n-1} P_{n-1} (1-t)^1 t^{n-1} + \binom{n}{n} P_n (1-t)^0 t^n, t \in [0, 1]$$

实现函数，能修改公有数组变量，将得到的点存储进去。

```
void getCurve(float *curve, int *mi) {
    int iter = 0;
    for (int i = 0; i < precision / 2; i++) {
        float t = (float)i / (precision / 2);
        float temp0 = 0.0f;
        float temp1 = 0.0f;
        for (int j = 0; j < points.size(); j++) {
            temp0 += pow(1 - t, points.size() - j - 1) * points[j][0] * pow(t, j) * mi[j + 1];
            temp1 += pow(1 - t, points.size() - j - 1) * points[j][1] * pow(t, j) * mi[j + 1];
        }
        curve[iter++] = temp0;
        curve[iter++] = temp1;
    }
}
```

- ④ 在鼠标点击的回调函数中，分别用不同的顶点数组存储线与点集（Bezier 曲线的点集）坐标，并在主函数的渲染循环中分别绘制。