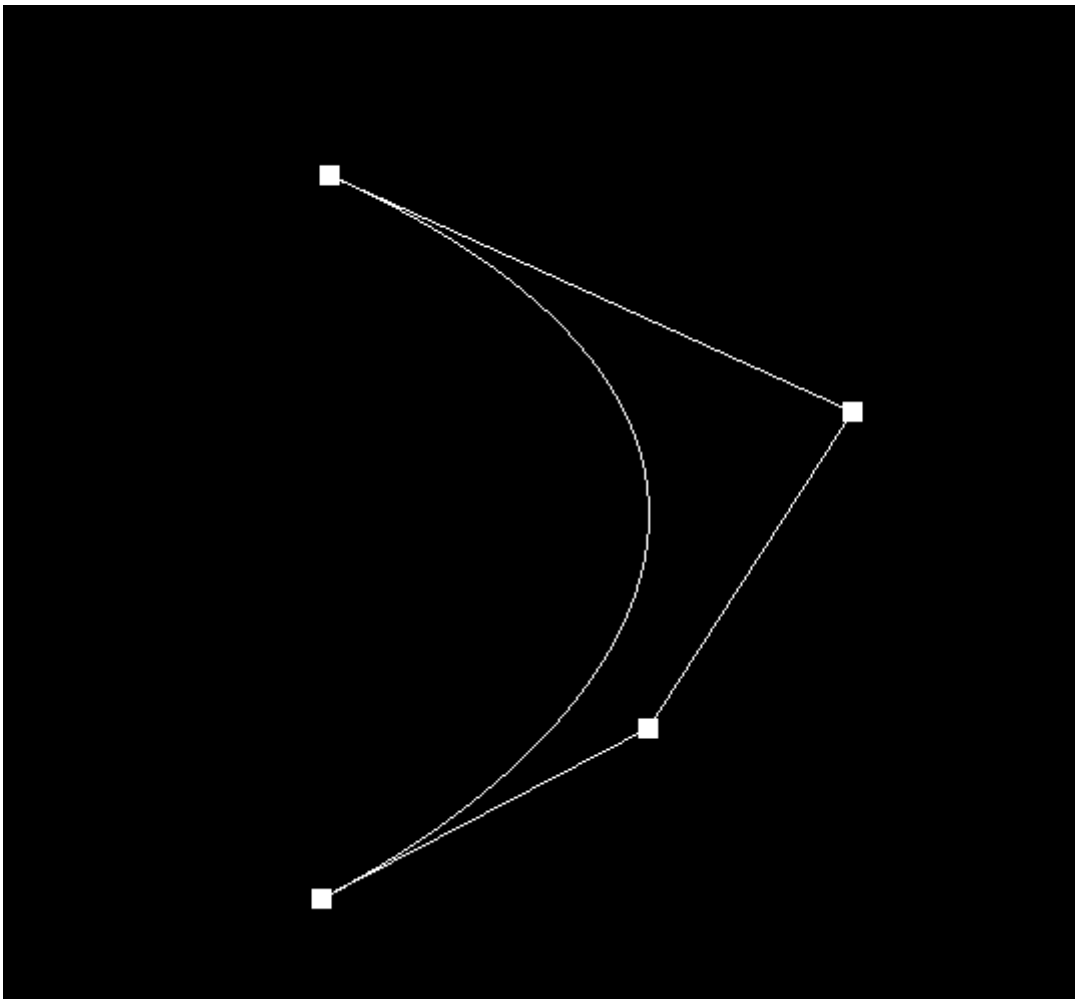
HW8

16340272 杨淼

Basic



glfwSetMouseButtonCallback回调函数监听鼠标点击事件，左键点击则添加一个坐标为当前鼠标位置的点，右键点击移除最后一个点。

```
void mouse_button_callback(GLFWwindow * window, int button, int action, int mods)
{
    if (button == GLFW_MOUSE_BUTTON_LEFT && action == GLFW_PRESS)     {
        float x = (mousePosX - WIDTH / 2) * 2 / WIDTH;
        float y = -(mousePosY - HEIGHT / 2) * 2 / HEIGHT;
        hw8.addPoint(x, y);
    }
    if (button == GLFW_MOUSE_BUTTON_RIGHT && action == GLFW_PRESS)     {
        hw8.removeLastPoint();
    }
}
//使用vector来保存节点，是因为c++标准保证了vector的数据在内存中是连续存放的
struct pos {
    float x;
    float y;
    pos(float x, float y):x(x), y(y){}
};
vector<pos> points;
```

Bezier曲线是通过对调和函数

取多个t值，并将得到的点存储下来，使用glDrawArrays(GL_LINE_STRIP,...)绘制曲线。

```
inline void HW8::calculateBezierPoint()
{
    bezierPoints.clear();
    const int num = 50;
    float detaT = float(1) / 50;
    float t = 0;
    for (int i = 0; i <= num; i++, t+= detaT)
    {
        bezierPoints.push_back(QFunc(t));
    }
}

inline pos HW8::QFunc(const float t) const
{
    pos result(0.0f, 0.0f);
    int n = points.size() - 1;
    for (int i = 0; i <= n; i++) {
        result = result + points[i] * BFunc(i, n, t);
    }
    return result;
}

inline float HW8::BFunc(const int i, const int n, const float t) const
{
    int sum1 = 1, sum2 = 1;
    for (int j = 0; j < i; j++) {
        sum1 *= (n - j);
    }
    for (int j = 1; j <= i; j++) {
        sum2 *= j;
    }
    return (float)sum1 / sum2 * pow(t, i) * pow(1 - t, n - i);
}
```