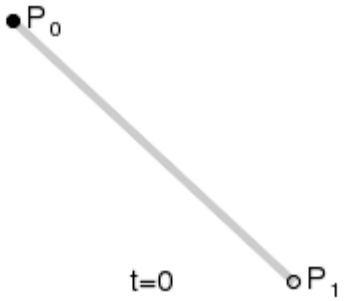
计算机图形学作业：利用 OpenGL 绘制 Bezier 贝塞尔曲线

Bezier 曲线原理

Bezier 曲线的原理我参考了这篇博客：https://www.cnblogs.com/hyb1/p/3875468.html。

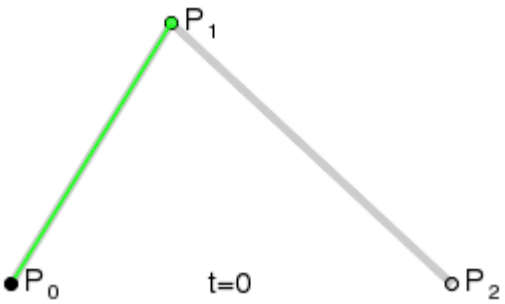
Bezier 曲线是应用于二维图形的曲线。曲线由顶点和控制点组成，通过改变控制点坐标可以改变曲线的形状。

一次 **Bezier** 曲线公式：



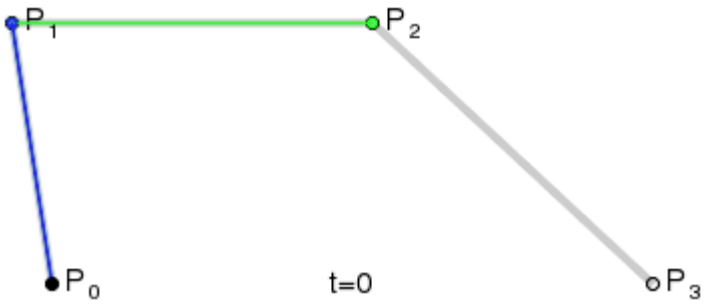
一次 Bezier 曲线是由 P0 至 P1 的连续点，描述的一条线段。

二次 **Bezier** 曲线公式：



二次 Bezier 曲线是 P0 至 P1 的连续点 Q0 和 P1 至 P2 的连续点 Q1 组成的线段上的连续点 B(t)，描述一条抛物线。

三次 **Bezier** 曲线公式：



由此可得 Bezier 曲线的一般方程为：

$$B(t)=\sum_{i=0}^{n}\binom{n}{i}Pi(1-t)^{n-i}t^{i}=\binom{n}{0}P0(1-t)^{n}t^{0}+\binom{n}{1}P1(1-t)^{n-1}t^{1}+\ldots+\binom{n}{n-1}Pn-1(1-t)^{1}t^{n-1}+\binom{n}{n}Pn(1-t)^{0}t^{n},t\in[0,1]$$

OpenGL 实现思路

在 OpenGL 窗口中，我们希望能通过左键点击窗口添加 Bezier 曲线的控制点，右键点击则对当前添加的最后一个控制点进行消除。然后根据鼠标绘制的控制点实时更新 Bezier 曲线。

捕获鼠标点击时的坐标

我们需要用一个回调函数，该函数是在鼠标移动时不断获取鼠标在窗口的坐标。

首先我们要声明全局的鼠标位置变量，代码如下：

```
float mouseXPos, mouseYPos;
```

然后在鼠标事件中不断更新全局位置变量的值。代码如下

```
void cursor_position_callback(GLFWwindow* window, double x, double y) {
        mouseXPos = float((x - WINDOW_WIDTH / 2) / WINDOW_WIDTH) * 2;
        mouseYPos = float(0 - (y - WINDOW_HEIGHT / 2) / WINDOW_HEIGHT) * 2;
        return;
}
```

根据顶点画出连续的线段

前面我们获取的鼠标的当前位置，那么当鼠标点击左键时，我们要捕获该点击事件，将顶点数据添加到 lineVertices 并通过绑定 VAO 画出线段。

先声明全局的顶点数据变量：

```
// 声明全局顶点变量, 点个数为 vertexLen / 3
int lineVertexLen = 0;
float lineVertices[MAX_VERTEX_LEN] = {
        //位置
        //-0.5f, 0.5f, 0.0f
};
```

然后再鼠标点击事件中操作：

```
void mouse_button_callback(GLFWwindow* window, int button, int action, int mods) {
        if (action == GLFW_PRESS) switch (button) {
                case GLFW_MOUSE_BUTTON_LEFT:
                        // 每隔两个点画一条直线
```

```
                        // 鼠标点击的点
                        lineVertices[lineVertexLen] = mouseXPos;
                        lineVertexLen++;
                        lineVertices[lineVertexLen] = mouseYPos;
                        lineVertexLen++;
                        lineVertices[lineVertexLen] = 0.0f;
                        lineVertexLen++;
                        // 添加索引,前一个点也新的点一起确定新线段
                        if (lineIndicesLen >= 2) {
                                lineIndices[lineIndicesLen] =
lineIndices[lineIndicesLen - 1];
                                lineIndicesLen++;
                                lineIndices[lineIndicesLen] =
lineIndices[lineIndicesLen - 1] + 1;
                                lineIndicesLen++;
                        }
                        else {
                                lineIndices[lineIndicesLen] = lineIndicesLen;
                                lineIndicesLen++;
                        }
                        break;
                default:
                        break;
        }
        return;
}
```

之后便是通过 VAO、VBO 和 GL_LINES 等画出线段。

根据顶点画出 Bezier 贝塞尔曲线

根据之前的 Bezier 曲线一般式，我们能很容易地根据顶点计算出 Bezier 曲线的点数据。只要新声明一个函数，传入顶点的数据和长度，就能计算出各个位置上的 Bezier 曲线的点数据，如下：

```
int getBezierVertex(float lineVertices[MAX_VERTEX_LEN], int lineVertexLen, float
bezierVertices[MAX_BEZIER_VERTEX_LEN]) {
        int bezierVertexLen = 0;
        if (lineVertexLen == 0) return bezierVertexLen;
        else if (lineVertexLen == 3) {
                bezierVertices[bezierVertexLen] = lineVertices[0];
                bezierVertexLen++;
                bezierVertices[bezierVertexLen] = lineVertices[1];
                bezierVertexLen++;
                bezierVertices[bezierVertexLen] = lineVertices[2];
                bezierVertexLen++;
        }
        else {
                for (float t = 0.000f; t <= 1.000f; t = t + 0.001f) {
                        double new_xPos = 0, new_yPos = 0;
                        for (int index = 0; index < lineVertexLen / 3; index++) {
                                // x坐标
```

```
                                new_xPos += lineVertices[index * 3] * pow(t,
index) * pow((1 - t), (lineVertexLen / 3 - 1 - index));
                                // y坐标
                                new_yPos += lineVertices[index * 3 + 1] * pow(t,
index) * pow((1 - t), (lineVertexLen / 3 - 1 - index));
                        }
                        bezierVertices[bezierVertexLen] = new_xPos;
                        bezierVertexLen++;
                        bezierVertices[bezierVertexLen] = new_yPos;
                        bezierVertexLen++;
                        bezierVertices[bezierVertexLen] = 0.0f;
                        bezierVertexLen++;
                }
        }
        return bezierVertexLen;
}
```

之后便可以通过 VAO、VBO 和 GL_POINTS 等画出 Bezier 曲线。

效果