

实现思路:

1. 按照 TA 给出的参考 gif, 先要画出贝塞尔曲线;
2. 画出控制点及它们之间的连线;
3. 在每帧的渲染中, 为使参数  $t$  为  $[0, 1]$  中循环的值, 采用函数  $t = (\sin((\text{float})\text{glfwGetTime()} + 1) / 2) + 1$  来生成  $t$ ;
4. 使用参数  $t$  来迭代生成中间的控制点及它们的连线;
5. 实现鼠标位置回调函数与鼠标点击回调函数并与主窗口绑定上。

主要 function:

1. `void BezierCurve::drawBezier(vector<point> cpoint, int bsize);`  
给出贝塞尔曲线的按顺序排放的控制点集 `cpoint`, 可以生成相应贝塞尔曲线的采样点集 `bpoint`(存储在类 `BezierCurve` 中), `bsize` 为指定的采样点的个数。
2. `void cursor_position_callback(GLFWwindow* window, double x, double y);`  
获取光标位置的回调函数, 会实时把鼠标位置赋值给全局的鼠标位置变量。
3. `void mouse_button_callback(GLFWwindow* window, int button, int action, int mods);`  
鼠标点击的回调函数, 左键在鼠标目前位置创建控制点, 右键取消最后一个点。

主要 algorithm:

Bezier 曲线:

此为 Bernstein 基函数:

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

Bezier 曲线的参数方程:

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

其中  $P_i$  为控制点的坐标。