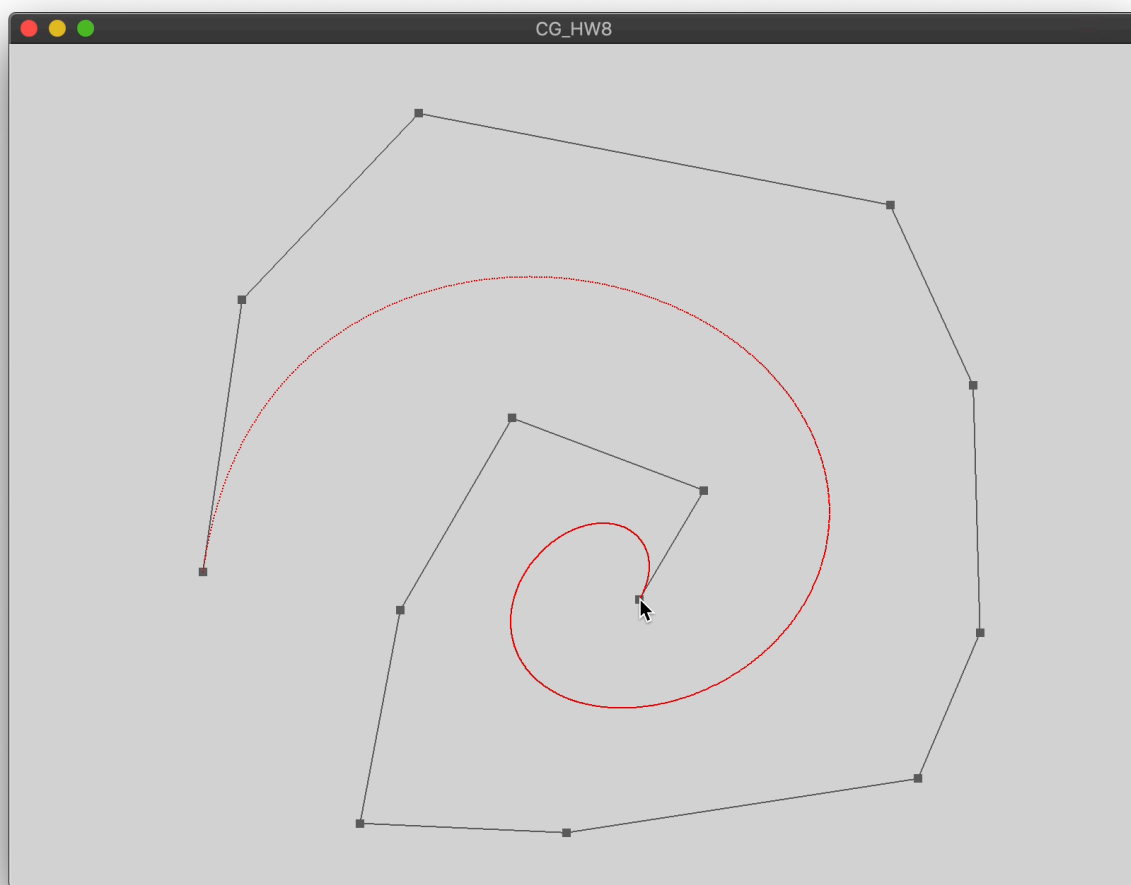


Basic:



1. 用户能通过左键点击添加**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。

使用`glfwSetCursorPosCallback`传入回调函数`cursor_position_callback`，在`cursor_position_callback`中获得当前鼠标`x`、`y`坐标赋值给全局变量`move_x`、`move_y`。

用全局变量`main_nodes`记录控制点，使用`glfwSetMouseButtonCallback`传入回调函数`mouse_button_callback`，当鼠标有点击时，如果是左击，将`move_x`、`move_y`作为新点加入`main_nodes`；如果是右击，从`main_nodes`推出最后一个元素；然后调用`make_bezier`（接下来说明）。

2. 根据鼠标绘制的控制点实时更新**Bezier**曲线。

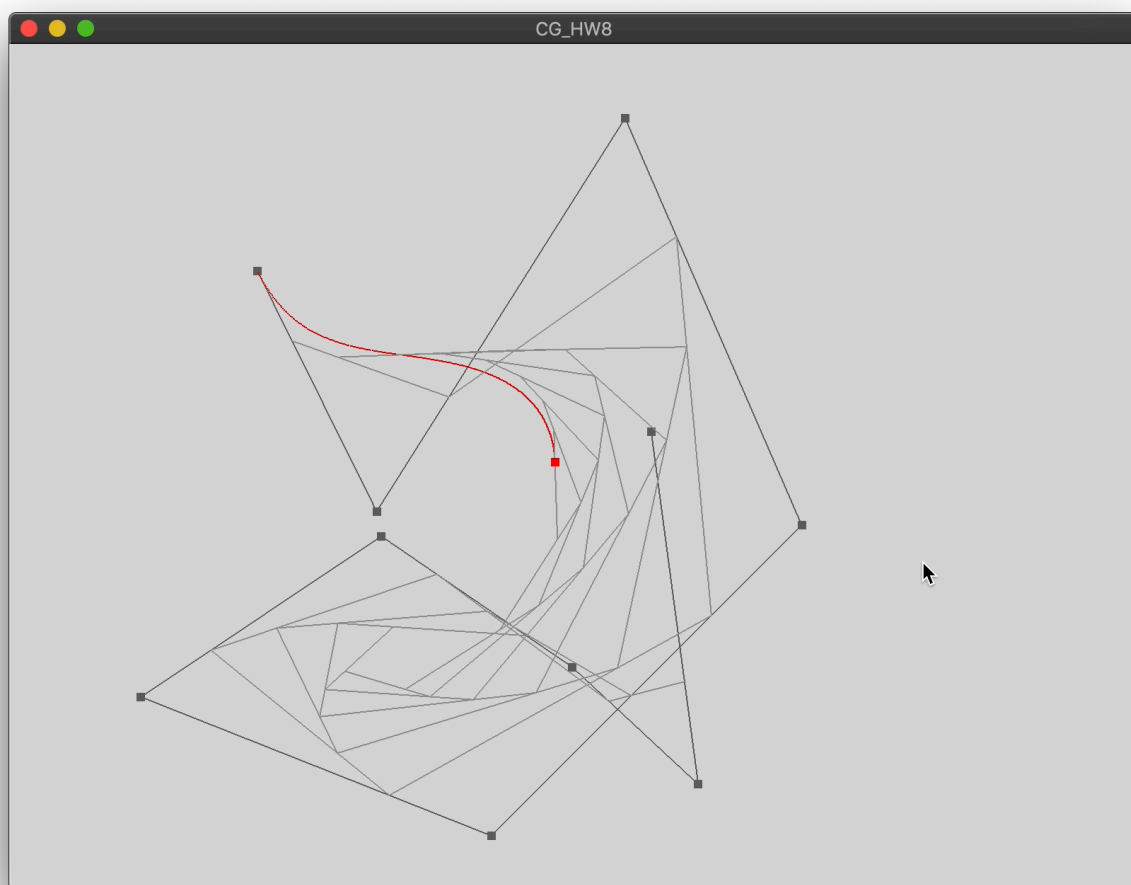
Bezier曲线的参数方程是 $Q(t) = \sum_{i=0}^n P_i B_{i,n}(t)$, $t \in [0,1]$, 这里

$$B_{i,n}(t) = \frac{n!}{i!(n-i)!} t^i (1-t)^{n-i}, i = 0, 1, \dots, n, P_i \text{ 为 } n+1 \text{ 个控制点的向量。}$$

使用全局变量**bezier_points**记录Bezier曲线上的点，在**make_bezier**函数中，首先将**bezier_points**清空，然后迭代将**t**从0赋值到1，每步递增0.001，对**main_nodes**的每个顶点的**x**坐标、**y**坐标分别乘 $B_{i,n}(t)$ 累加，分别作为**bezier_points**新点的**x**坐标、**y**坐标。

为了实现在实时更新，只要在渲染循环中将**main_nodes**、**bezier_points**（不管它们是否有元素）的点用**glBegin(GL_POINTS)**画控制点和Bezier曲线（假设**t**每递增0.001已足够形成曲线），并对**main_nodes**每对点用**glBegin(GL_LINES)**连线显示出控制多边形。

Bonus:



1. 可以动态地呈现Bezier曲线的生成过程。

为了判断用户已经确定好控制点，使用`glfwGetKey(window, GLFW_KEY_ENTER) == GLFW_PRESS`判断用户是否按下回车键，切换`bool`变量`drawing`的值。如果当前没有在动态展示，按下回车键就开始动态展示（此时用`glfwSetCursorPosCallback(window, NULL)`，`glfwSetMouseButtonCallback(window, NULL)`停止捕捉鼠标状态，并让`t`从0开始，每个渲染循环增加0.001，并调用`draw_process`（接下来说明））；如果当前在展示就结束展示（此时用`glfwSetCursorPosCallback`，`glfwSetMouseButtonCallback`重新绑定相应回调函数恢复捕捉，并将`t`归0，清空`bezier_points`）。

在函数`draw_process`中，对于给定的`t`，先在给定的 $n + 1$ 个控制点形成的 n 条线上各取`t`比例的点，即 $(1 - t)P_{i-1} + tP_i$ ，得到 n 个点，再用`glBegin(GL_LINES)`将每对点连起来画 $n - 1$ 条线，再用这 n 个点递归调用`draw_process`，`t`不变，执行相同的过程，当点变成2个时，取其形成的线上`t`比例的点，存入`bezier_points`，并结束调用`draw_process`。

`bezier_points`的点在渲染循环中渲染显示，已存入的点会一直显示，而每个渲染迭代传给`draw_process`的`t`不一样，`draw_process`中渲染的线就会不断变化，展示出动态生成过程。