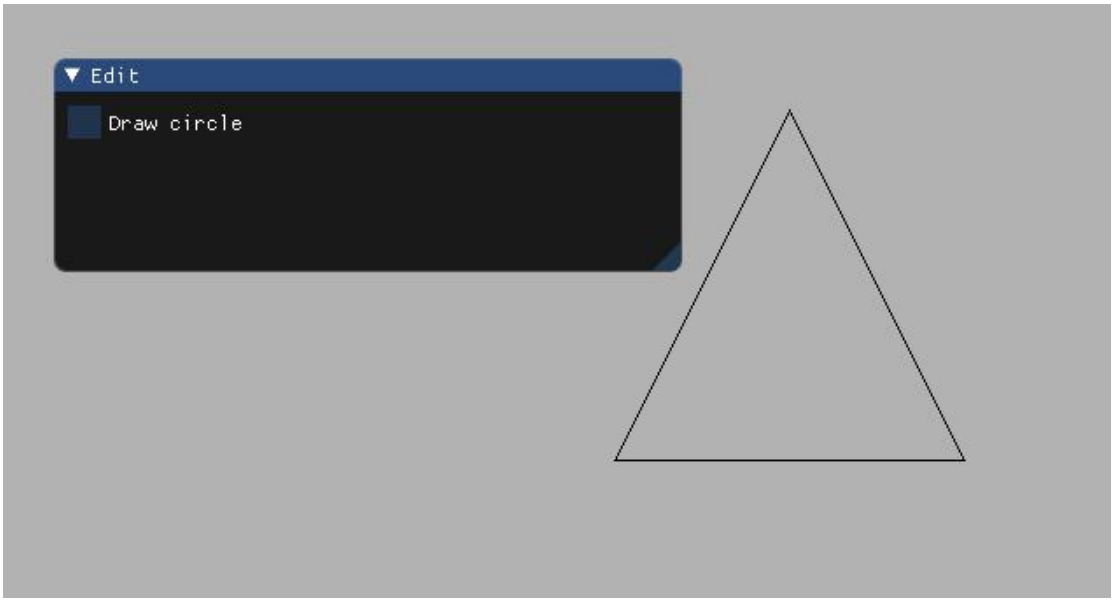
1. 使用 Bresenham 算法(只使用 integer arithmetic)画一个三角形边框：input 为三个 2D 点；output 三条直线（要 求图元只能用 GL_POINTS ，不能使用其他，比如 GL_LINES 等）。

2. 使用 Bresenham 算法(只使用 integer arithmetic)画一个圆：input 为一个 2D 点(圆心)、一个 integer 半径； output 为一个圆。

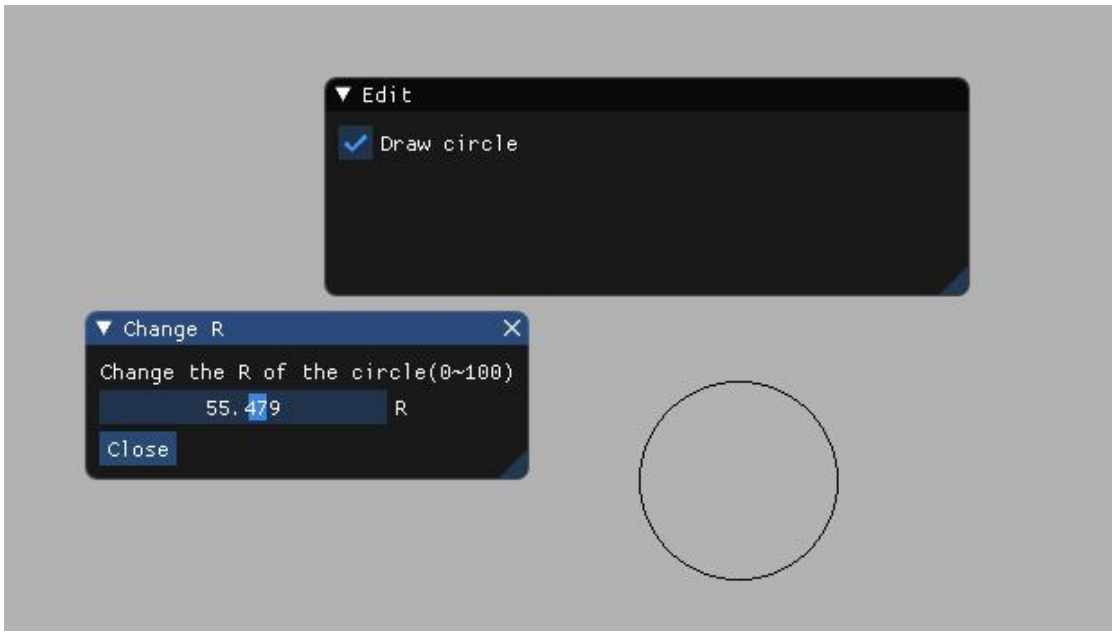
3. 在 GUI 在添加菜单栏，可以选择是三角形边框还是圆，以及能调整圆的大小(圆心固定即可)。

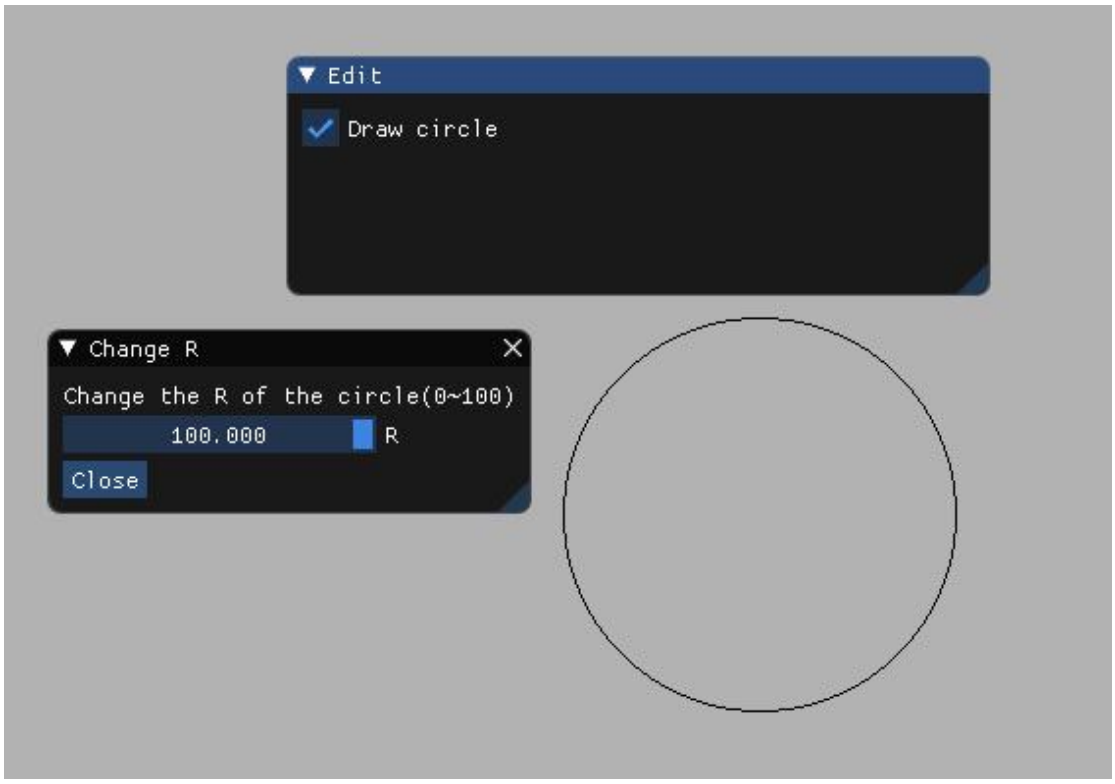
运行结果截图：

三角形



勾选 Draw circle，出现改变圆半径窗口，可滑动滑块选择半径开始画圆




实现思路：

vector<int> Bresenham(int x0, int y0, int x1, int y1)：通过 Bresenham 直线算法得到各个点坐标

vector<int> DrawTriangleBresenham(int x1, int y1, int x2, int y2, int x3, int y3)：得到（x1，y1）（x2，y2）（x3，y3）三点构成三角形的三条边上各个点坐标

void getAllPoints(int x0, int y0, int x, int y, vector<int> &points)：利用圆的八对称性，通过一个点坐标得到得到圆上对称八个点坐标

vector<int> DrawCircleBresenham(int x0, int y0, int r)：通过计算 1/8 圆周上的点来的得到整个圆上所有的点坐标