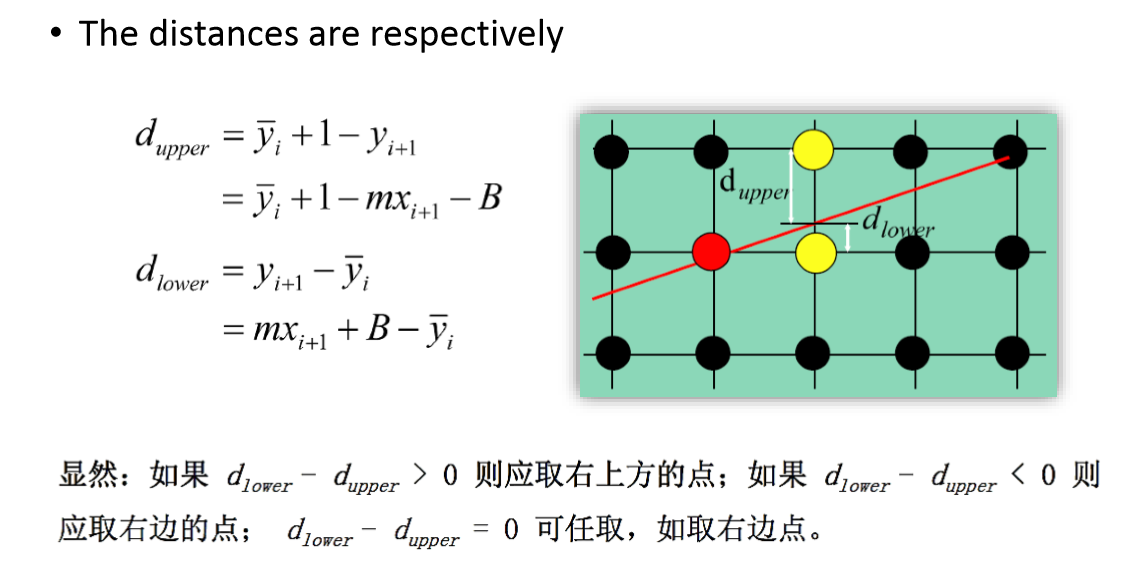
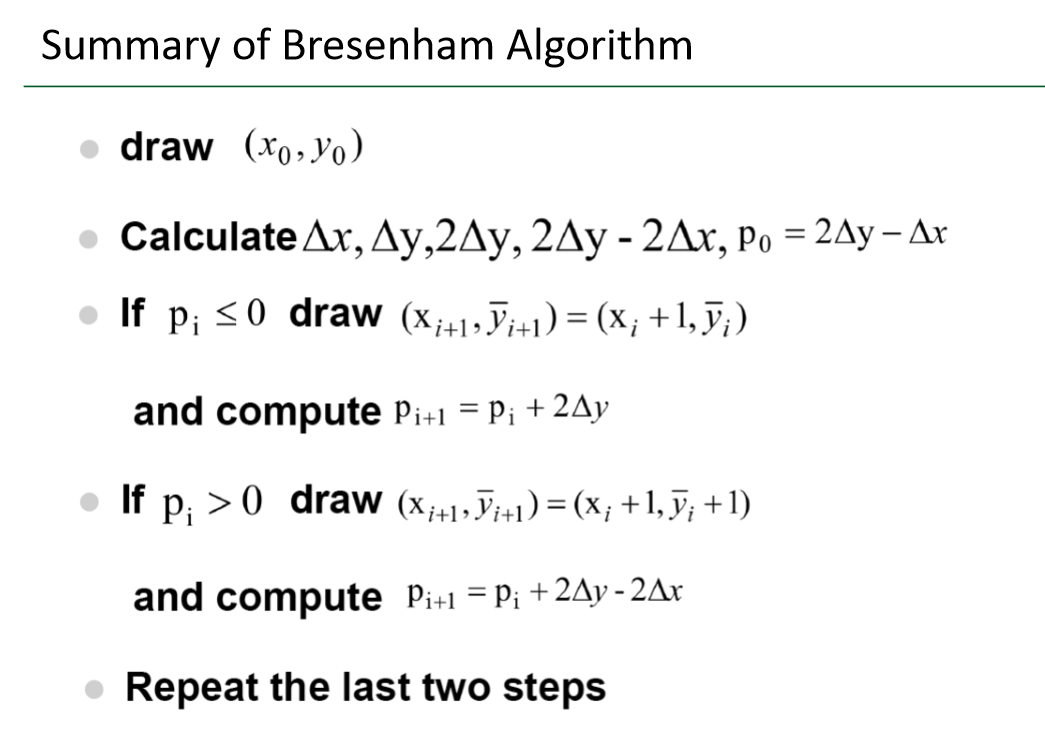


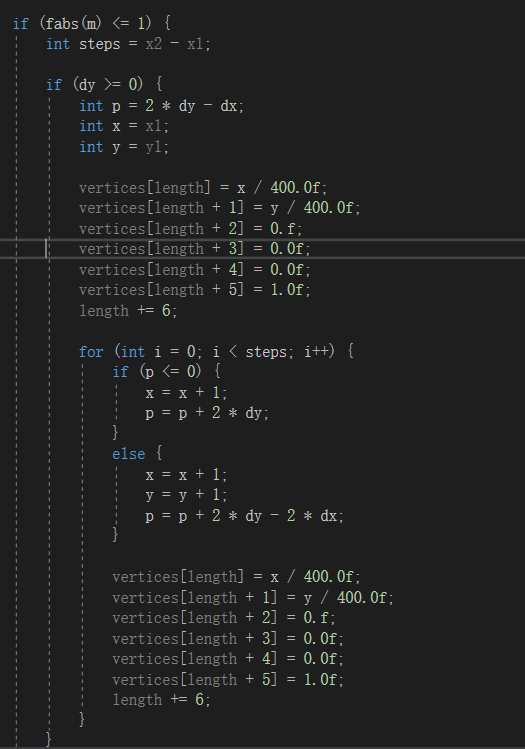
**实现思路:**

1. **Bresenham画直线的基本思想**

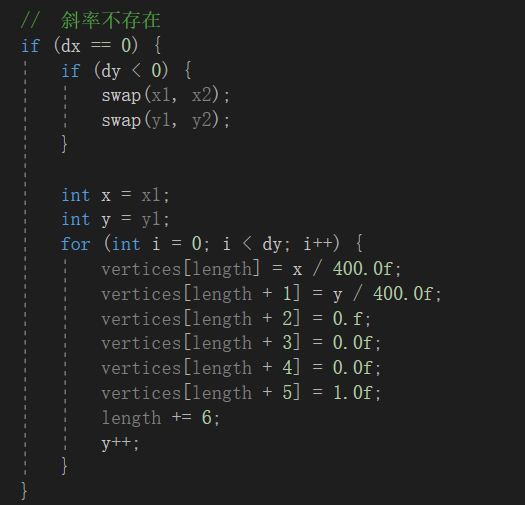


1. **Bresenham算法的伪代码实现(斜率k为0~1)**

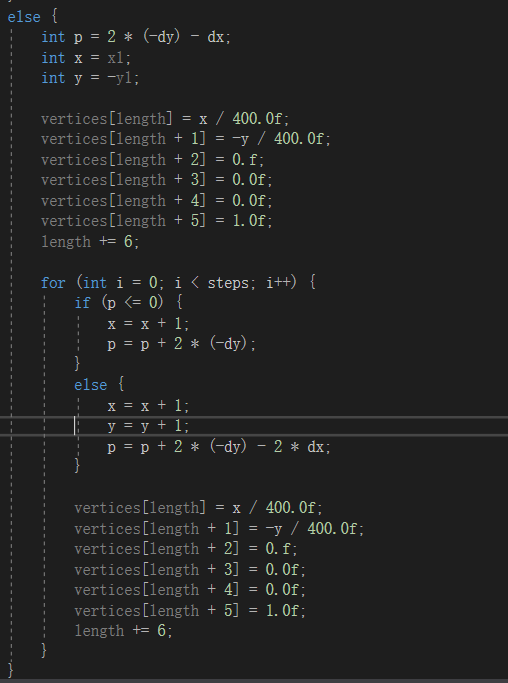




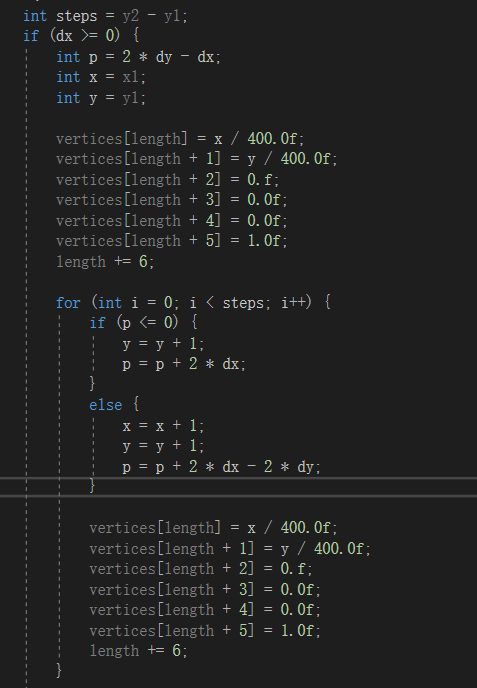
1. **将算法扩展到斜率k: 不存在, >1, -1~0, <-1**
2. **当斜率k不存在: x值不变, y值从y1递增到y2**



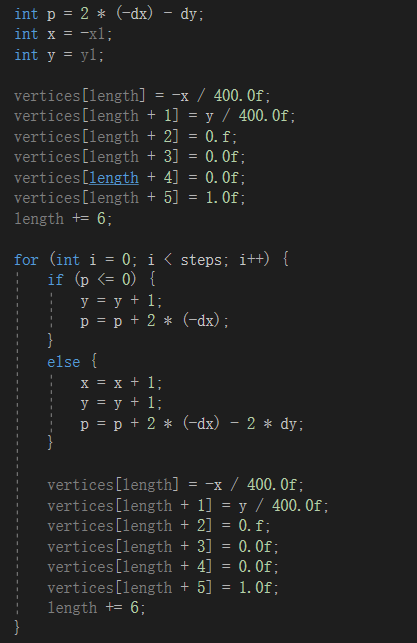
1. **当-1<k<0: 此时的图像与0<k<1关于x轴对称, 因此将算法中的y, dy作关于x轴对称**



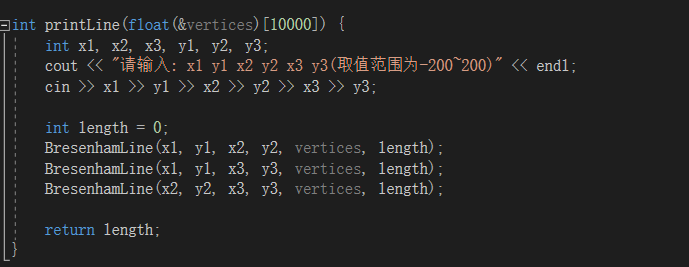
1. **当k>1时: 此时的图像与0<k<1关于y=x对称, 因此将算法中的dx, dy作y=x对称**

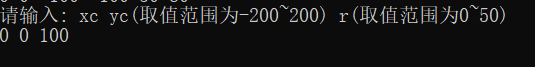


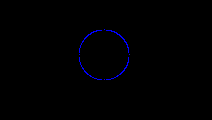
1. **当k<-1时: 此时的图像与k>1关于x轴对称, 因此将算法中的y, dy作关于x轴对称**



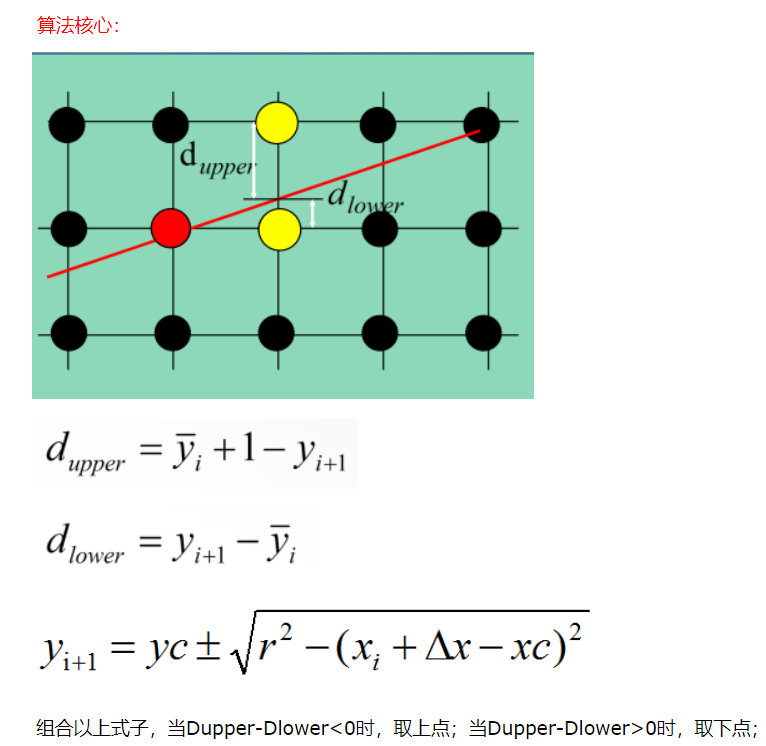
1. **根据3个2D点两两作直线**



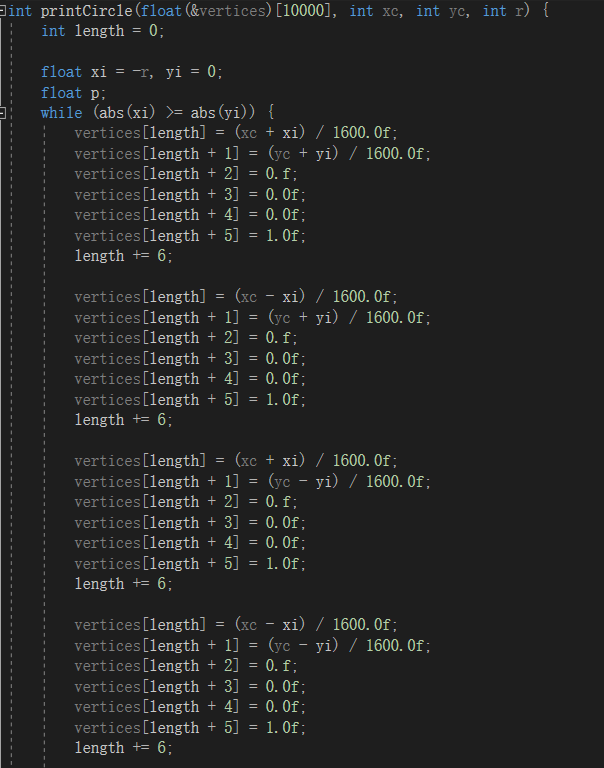


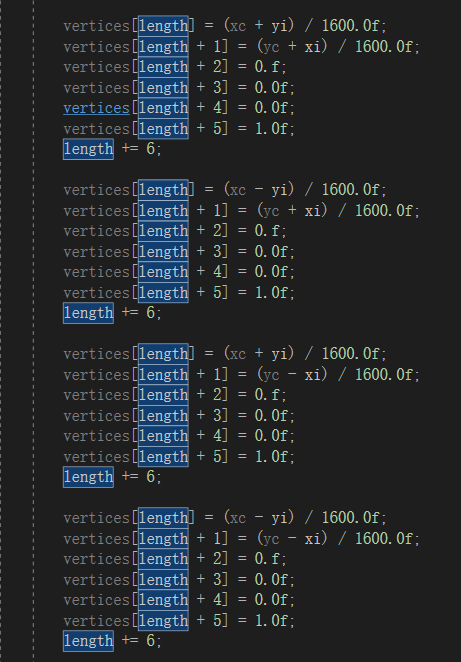


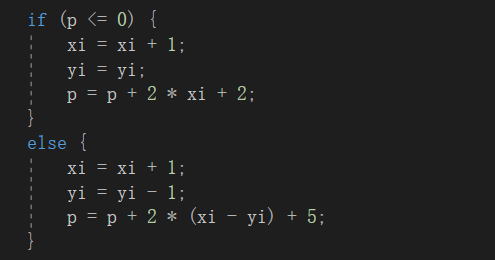
**实现思路:**



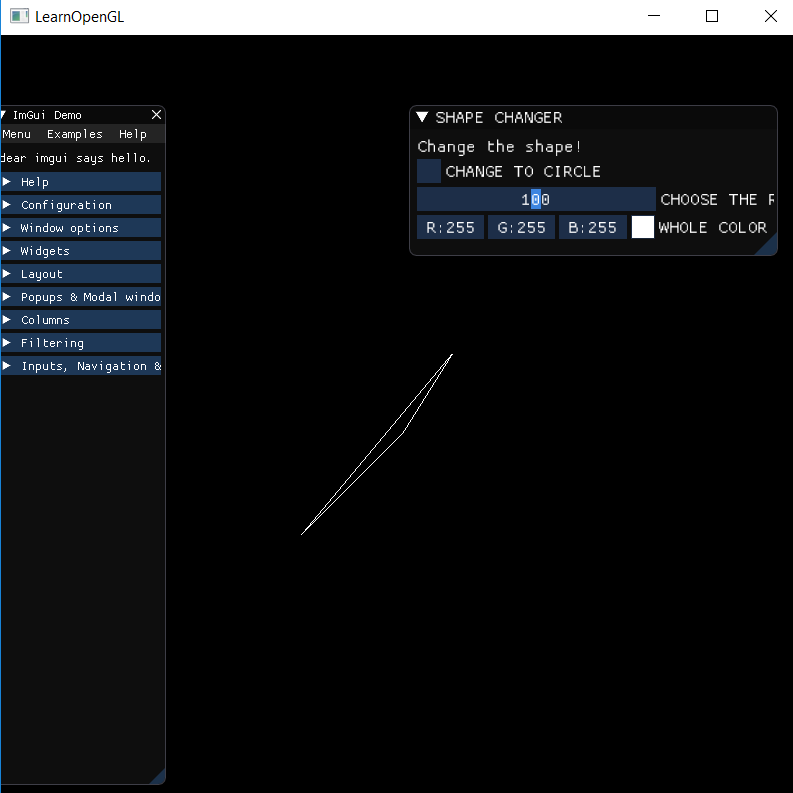
1. 利用圆的对称性质，仅仅计算1/8的圆弧的点, 然后映射到其他7个部分上

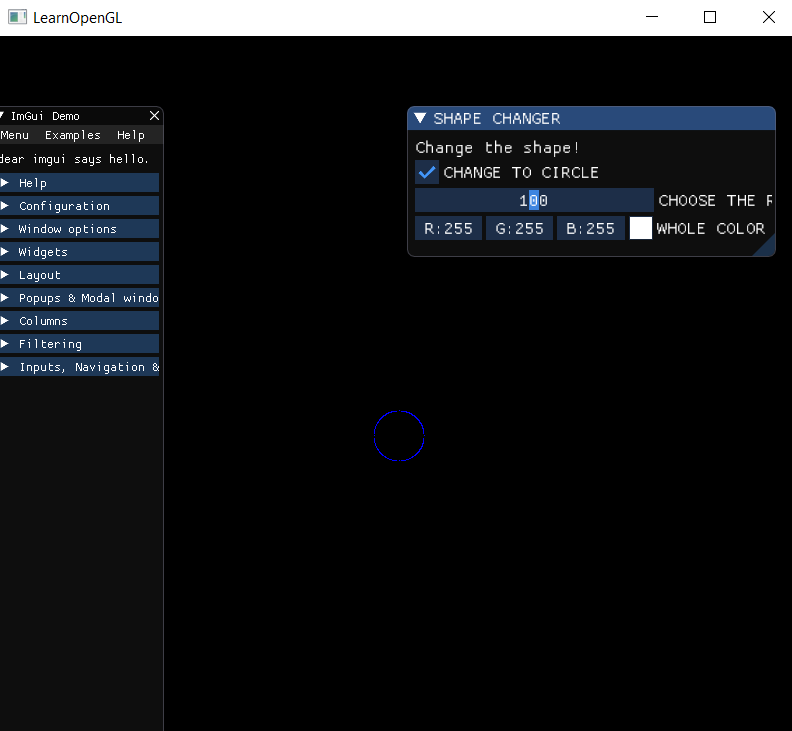


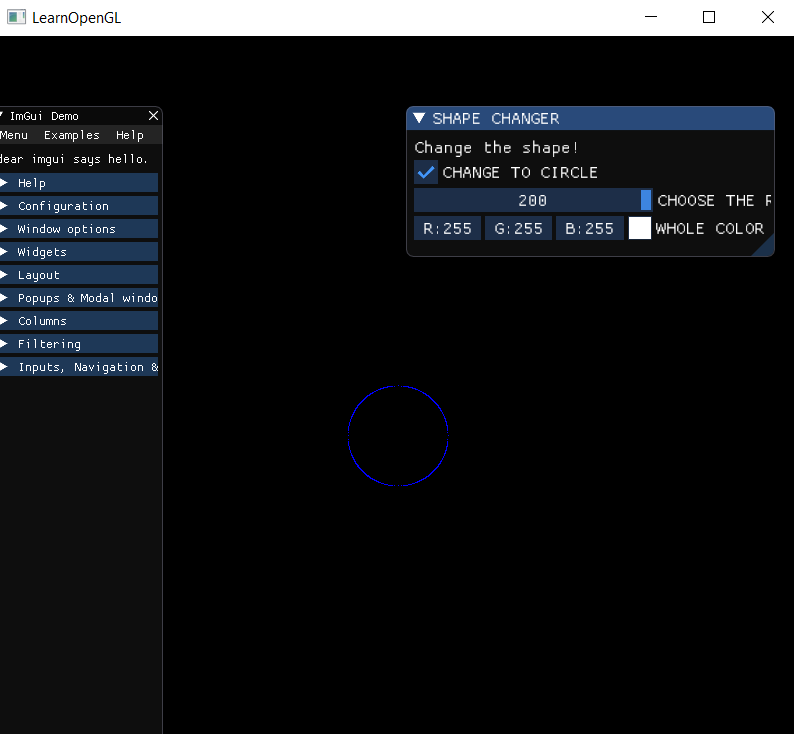






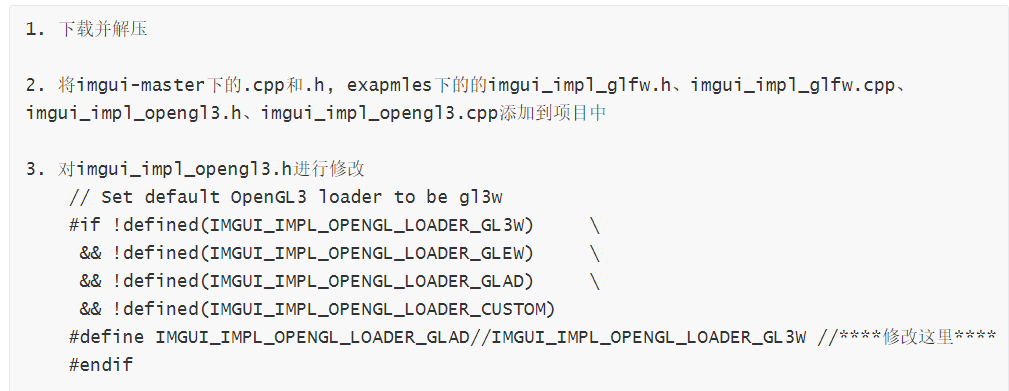




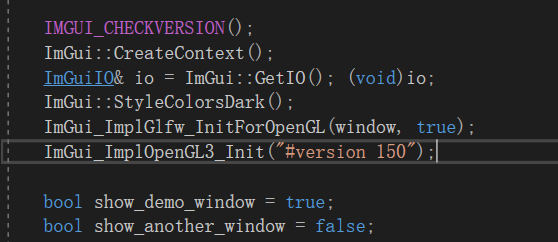


**实现思路:**

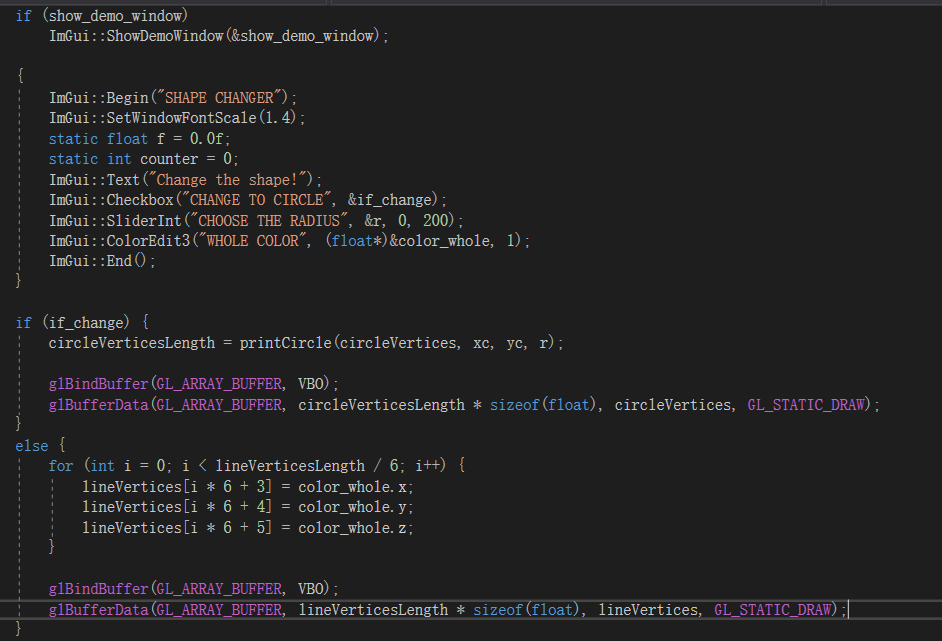
1. 配置imgUI.



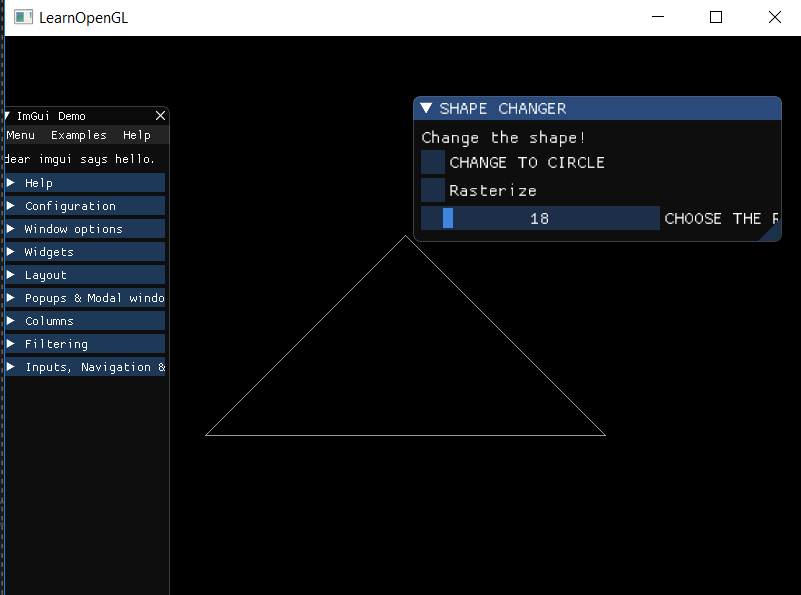
1. 初始化imgUI

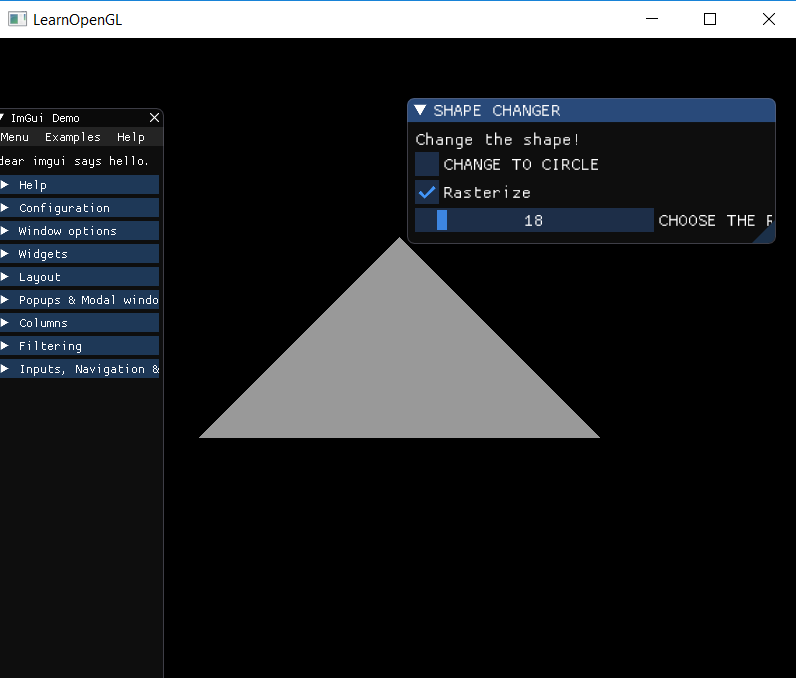


1. 设置面板, 以选择形状、调节圆的半径



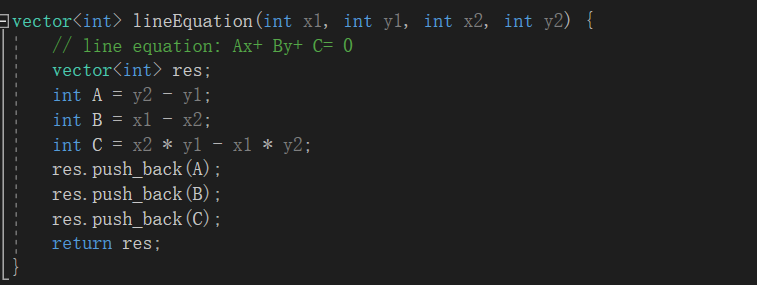
**Bonus:**





**实现思路:**

1. 计算三条边的Ax+By+C=0式的方程



(2) 计算三角形的外接矩形, 并遍历外接矩形中的每一个点，用得到的3条边的方程判断点在不在三角形中，如果在就画出该点

