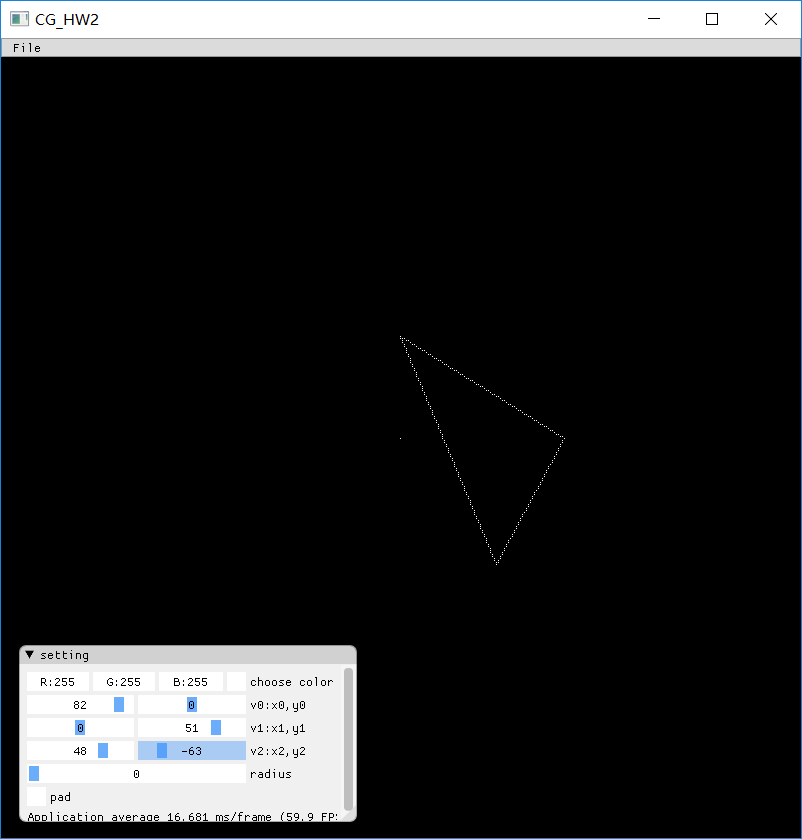
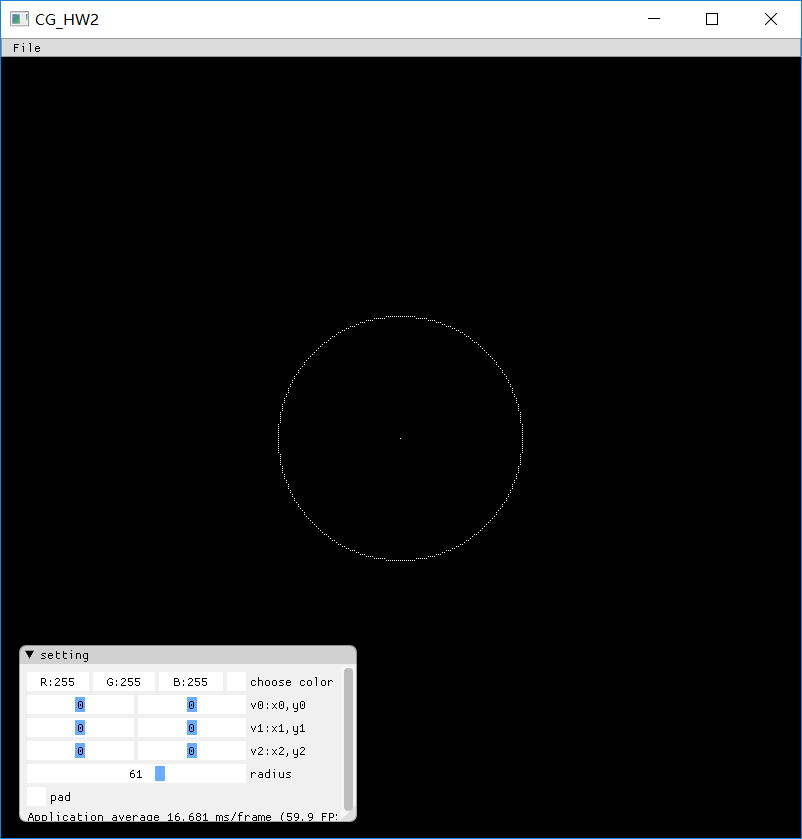
# BASIC：

**为了更好展示算法的效果用400\*400的区域均匀展示200\*200的点阵，从而能看出算法的颗粒效果，动态效果见附视频文件hw3.mp4。**

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）。



1. 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。

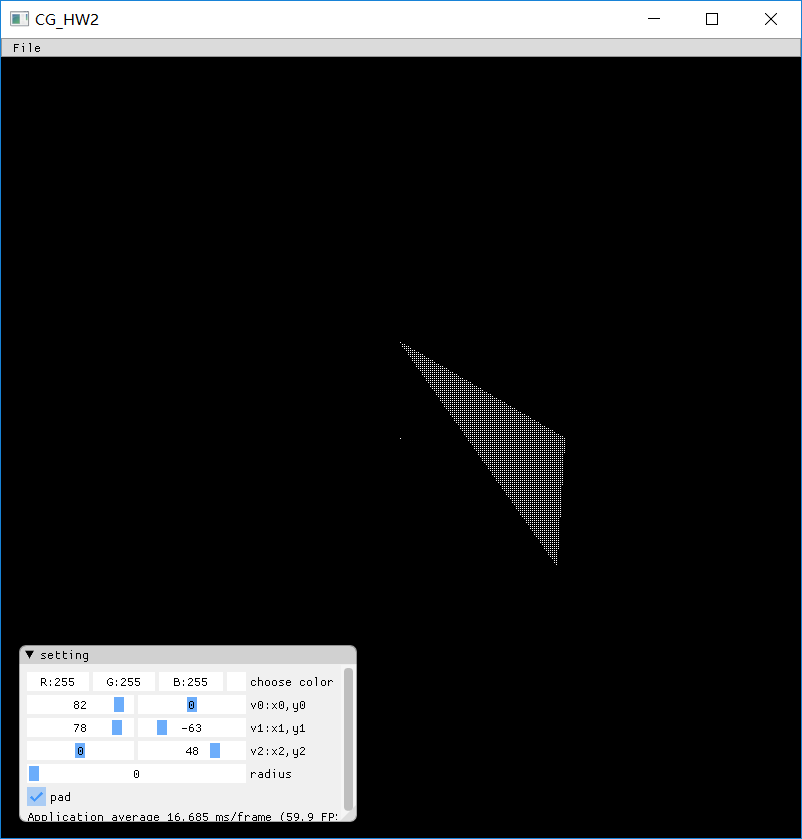


1. 在GUI在添加菜单栏，可以选择是三角形边框还是圆，以及能调整圆的大小(圆心固定即可)。

如上图可以通过菜单调节三角形三个顶点的坐标，和圆的半径，pad勾选时则填充三角形。

# Bonus

1. 使用三角形光栅转换算法，用和背景不同的颜色，填充你的三角形

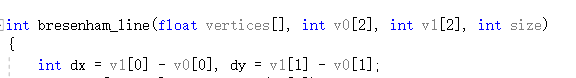


# 代码思路解析

附代码文件（本次作业的核心代码在hw3.cpp内）

1. 直线算法按照课件的标准实现

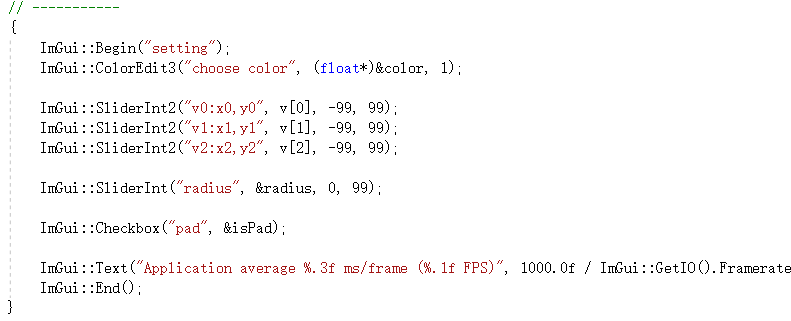
使用下面划线函数在三点之间都画一条直线



1. 画圆算法采用Bresenham算法（圆心固定0，0）



1. Imgui控件的实现部分



1. 填充三角形采用线性方程的方法

主要使用两个函数

根据两点生成直线方程的函数



使用直线三条方程设置点亮三角形内所有的点的函数

