**第二次作业 三角形的光栅化**

第二次作业包括两项任务：任务1，任务2，见后面的任务说明。

1、作业提交要求：

（1）对开发和运行环境作以说明；

（2）对算法的设计思想和实现技术作以阐述；

（3）需要有程序的运行结果，并对结果进行分析（如：性能、问题、改进等）。

2、提交截止时间：2020年4月16日 23：59：59。

**任务1：阅读理解如下算法1、算法2，说明算法2是如何改进算法1的？**

算法1：

For 每一个三角形

Compute 三角形的包围盒bbox, 用视口裁剪包围盒（即：包围盒不能超过显示区域）。

For all pixels in bbox

设当前像素为(x,y)。 把(x,y) 或者(x+0.5,y+0.5) 代入三角形每一条边的方程Ei: aix + biy + ci ，判断像素点在边的那一侧(可见侧、不可见侧)。

If all > 0 （如果全部在可见侧：像素点在三角形所有边的可见侧）

用三角形的颜色显示该像素。

算法2：

For 每一个三角形

Compute 三角形的包围盒bbox, 用视口裁剪包围盒（即：包围盒不能超过显示区域）。

For all scanlines y in bbox

从左到右扫描，设当前扫描线最左侧像素为(x0,y)。 把(x0,y) 或者(x0+0.5,y+0.5) 代入三角形每一条边的方程Ei = aix + biy + ci ，计算该扫描线对于每一条边Ei的初值。

For 该扫描线在包围盒中的所有像素

当前像素全部在可见侧，即 If all Ei>0

用三角形的颜色显示该像素。

每一条边增量法：Ei += ai （对应于x0=x0+1）

**任务2：编程实现以上两个算法，并至少绘制5个不同的填充三角形。**

**提示**：每一条直线把平面分成两部分

- 直线方程 Ax + By + C=0

- 令：L(x,y) = Ax + By + C

- 在直线上: L(x,y) = 0

- 在直线负的一侧: L(x,y) < 0

- 在直线正的一侧: L(x,y) > 0