

## Homework 3 - Draw line

deadline: 3.28 周三 23:59前

### Introduction

---

在这次作业里，要求大家掌握Bresenham Algorithm，去绘制直线和圆，从而去更好地理解光栅化(Rasterization)的原理。

**作业严禁抄袭，被发现者当次作业0分！**

### References

---

#### 1. Bresenham Algorithm

Bresenham算法在进行直线的扫描转换时，由于不涉及浮点数运算，只是整数类型的运算，所以大大提高了计算速率。由于Bresenham算法精简、高效的特性，当前已经完全由硬件实现。虽然后面都不再需要大家自己实现Bresenham算法，但是毕竟此算法对CG贡献极其之大，因此仍希望大家通过这次作业去了解算法背后的原理。

### Homework

---

#### Basic:

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，可以选择是线还是圆，以及能调整圆的大小(圆心固定即可)。

#### 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。

#### 作业要求:

1. 把运行结果截图贴到报告里，并回答作业里提出的问题。
2. 报告里简要说明实现思路，以及主要function/algorithm的解释。