

# 图形学实验 PA0：光栅图形学 实验报告

徐康平 计07 2020011016

## 代码逻辑

### Line

参考教材上Bresenham算法的实现，具体根据A, B两点的坐标情况进行绘制

- 特殊处理A, B横坐标相同的情况，此时固定x，遍历A, B之间的像素点绘制即可
- 如果斜率k存在，则根据k的绝对值与1的关系：
  - $|k| \leq 1$ 时，按照x轴坐标均匀增加，计算y的坐标点的方式进行，将A, B中位于下方的（即y小的）作为起点 $(x_1, y_1)$ ，向 $(x_2, y_2)$ 绘制，进一步分为 $k \geq 0$ 和 $k < 0$ 的两种情况：
    - $k \geq 0$ 时，即与书上的代码相同，e从-0.5开始，横坐标从 $x_0$ 增加至 $x_1$ ，e每次增加 $|k|$ ，计算y的像素位置
    - $k < 0$ 时， $x_1 < x_0$ ，循环x坐标从 $x_0$ 开始递减至 $x_1$ ，这一过程中y依然是递增的，与上面相同
  - $|k| > 1$ 时，按照y轴坐标均匀增加，计算x的坐标点的方式进行，将A, B中x较小的作为起点，e的初始值仍为-0.5，表示的是x轴的差值，每次增量是 $1/|k|$ ，其他的逻辑与上面一样分为 $k \geq 0$ 和 $k < 0$ 两种情况处理

### Circle

首先实现同时绘制8个对称点的方法circlePoints，与教材上的算法完全相同，只是增加了圆心坐标 $(cx, cy)$ ，每次同时绘制 $(cx \pm x, cy \pm y), (cx \pm y, cy \pm x)$ 共8个点

draw方法只需要计算出 $x > 0, y > 0$ 且 $x \leq y$ 的第一象限1/8个圆的点坐标即可，使用教材上的中点画圆法：

- 从 $x=0, y=r$  开始，初始判别式为 $d = 1.5 - \text{radius}$
- 循环增加x，根据当前的d符号：
  - 如果 $d < 0$ ，则说明中点在圆内，应当取圆外的像素点 $(x+1, y)$ ，更新d
  - 如果 $d \geq 0$ ，说明中点在圆外（上），取圆内的点为下一个像素点，y减一，更新d
- $x > y$  时停止

### Fill

通过深度优先方式查找到从seed开始所有连通的同样颜色的像素，改为指定颜色：

- 使用queue，初始化加入source point，记录原来的颜色用于区分是否已经染过色
- 每次从队列中获取一个坐标点，检查其是否在图像内部，是否已经染过色，是则跳过，取下一个点
- 对于未染色的点，将其染色，并且将**四个方向**上的点加入队列
- 迭代至队列为空

另外，对于Line和Circle，检查每次设置像素的坐标是否在图内

## 代码参考

没有和其他同学讨论或者借鉴网上代码

参考了教材上的算法思路和代码实现

## 问题

在实现line的绘制时，对于各种情况的分类处理比较复杂，用了一些时间