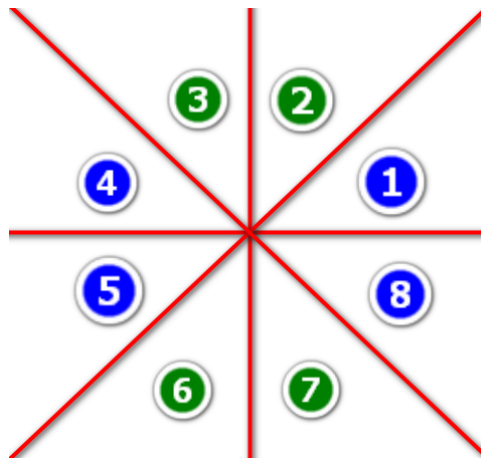


图形学 PA0 report

计83 何雨泽 2018011351

1. 你所实现的画线、画圆和区域填充逻辑是怎样的？

- 画线采用 **Bresenham 算法**：首先对斜率的绝对值进行讨论，若为 ∞ ，则直接画一条竖线；剩下按 $|k| > 1$ （图中绿色）和 $|k| < 1$ （图中蓝色）进行讨论，每个分支下又要讨论 4 种平面区域，如下所示：



以 $|k| < 1$ 、**区域 8** 为例：每次横坐标一定增加 1，纵坐标有可能增加 1，也有可能不增加。是否增加使用误差 e 来进行判断，初始值为 -0.5 ，每次画一个点 e 就自增 $|k|$ ，如果 $e > 0$ 则纵坐标增加，否则不增加。

$|k| < 1$ 的其他 3 种情况可由 x 轴对称、 y 轴对称、中心对称得到。

$|k| > 1$ 时的 4 种情况将横坐标与纵坐标的地位对换，其他算法完全相同。

- 画圆采用**中点画圆法**：将平面分成 8 份（与上图相同），先以区域 7 举例，构造以下判别式 d ：

$$d = F(M) = F(x_p + 1, y_p - 0.5) = (x_p + 1)^2 + (y_p - 0.5)^2 - R^2$$

若 $d < 0$ ，纵坐标不变画点，判别式值改为 $d + 2x_p + 3$ ；

若 $d \geq 0$ ，纵坐标减小 1 画点，判别式值改为 $d + 2(x_p - y_p) + 5$ 。

每次画一个点，都同时将其其他 7 个区域的对应点也画上。对于其他 7 个区域，分别进行 x 轴对称、 y 轴对称、 x, y 值交换即可。

- 区域填充采用**广度优先搜索**：先记录下填充点位置的原始颜色，由该位置开始广搜，每搜到一个与原始颜色相同的点就填充它，并标记为已访问，直到找不到满足条件的点。

2. 你在实现中遇到了哪些问题？

- 最开始使用 Bresenham 算法时很多情况如 $k = \infty$ 没有考虑到，以及 float 转换出了一些问题，导致缺了一些线条，某一些线条的斜率也不对，后来都修改过了。
- 第一次画圆的结果被分成了 4 段散乱的弧，后来发现是在通过对称性调换 x, y 的时候不小心把中点的坐标 cx, cy 也调换了，改正后结果正确。

3. 你在完成作业的时候和哪些同学进行了怎样的讨论？是否借鉴了网上/别的同学的代码？

没有！完成时间为 4.25，估计也没啥同学在这个时间点做完 >_<

4. 你的代码有哪些未解决的bug？如果给你更多时间来完成作业，你将会怎样进行调试？

暂时貌似没有发现 bug。

5. 你对本次作业有什么建议？文档或代码中有哪些需要我们改进的地方？

暂时感觉没有啦.....这次作业比其他简单许多，就涉及到一些对称性的转换和数学公式的运用（逃）