

图形学实验PA0 实验报告

计81 肖光烜 2018011271

实现逻辑

直线绘制

实现了光栅图形学中采用最广泛的Bresenham直线扫描转换算法，已知起点和终点，先逐一循环起点到终点的x坐标，依次算出对应的y坐标，将其最接近的整y坐标画上；再逐一循环起点到终点的y坐标，依次算出对应的x坐标，将其最接近的整x坐标画上。这样两遍画线的算法可以保证所画的直线连续且在所有斜率的范围内均不出错。

圆形绘制

按照书上的八分画圆算法实现即可，遍历八分之一圆上的每个点，然后将其对称8份画出8个点，即可画出整个圆。

区域填充

简单实现一个BFS算法即可，从中心点像四面进行扩散遍历，遇到颜色与old Color相同的点就染色并加入队列，直到遍历完联通的old Color颜色的点并染色完毕。

遇到的问题

遍历方向问题

课本上Bresenham算法并没有保证 $dx(dy) > 0$ ，导致如果直线遍历起点的x(y)坐标小于终点x(y)坐标时会导致直线画不出来，虽然这次作业的测试样例没有这种情况，但是我为了提高我的算法的鲁棒性还是统一将起点设为x(y)坐标较小的一边。（当遍历x的时候设x较小的一边，遍历y的时候设y较小的一边）。

直线斜率问题

一开始我没看到讲义上的算法是只针对斜率 $0 \leq k \leq 1$ 的直线成立的，这样会造成画出的直线两种后果：

- 画出 $1 < k < +\infty$ 的直线不连续，在区域填充的时候会出现漏色的问题。
- 无法画出 $k < 0$ 和 $k = \infty$ 的直线。

为了解决这一问题，我将逐步累积的x的算法对y如法炮制地做一遍。这样两遍画线的算法可以保证所画的直线连续且在所有斜率的范围内均不出错。

Honor Code

本次作业我完全独立完成，没有与同学讨论，也没有借鉴网上/别的同学的代码。

Bug以及调试说明

我目前在我的代码中没有找到未解决的bug，如果更多时间我会再设计几个样例做一些边界情况的检查。

建议与改进

我个人认为教参上的代码过于冗余而且并不完全正确（直线绘制部分），希望能得到改进。