计算机图形学基础 PA0
实验报告

计 97 郭昊 2019013292

2021 年 3 月 16 日

1 画线、画圆和区域填充

1.1 实现逻辑

对于画线，采用课本算法 2.4，改进的 Bresenham 画线法实现。由于该算法只能处理斜率在 $[0,1]$ 的情况，所以需要对各种情况进行判断。

为了代码的简洁，我采用了一种特殊的方式：比较两点的水平距离和竖直距离，选择二者中较长者从左向右（从下向上）进行遍历，当另一坐标轴上点需要前进一格时，需要按照两点在该坐标轴方向上的距离正负来判断行进的方向。

对于画圆，只需要采用课本算法 2.5，中点画圆算法即可。为了避免浮点数计算，我采用把所有相关 d 的值乘 4 的方法，即可使 d 可用 `int` 类型表示。

这里需要注意的是圆心并不一定是原点，需要对此进行处理。假设圆上某一点的坐标向量为圆心的坐标向量 $\vec{A}$ 加上它相对圆心的向量 $\vec{B}$，对 $\vec{B}$ 进行对称即可方便地算出 8 个对称点的坐标。

对于区域填充，我采用的是课本算法 2.10，即区域填充扫描线算法。这里需要注意的点只有两个：就是判断边界，以及对于原色填充的特判。首先需要满足着色点是否在图像中，其次，如果需要填充的颜色与原来该点的颜色相同，就会进入死循环，故需要判断一下。

1.2 遇到问题

1.2.1 如果直接使用书中改进 Bresenham 算法，当 $dx < 0$ 无法绘制

只需从左往右（从下往上）画，让被遍历的坐标轴始终增加即可，并且通过判断起点与终点在另一坐标轴方向上的距离正负来判断其行进方向。

1.2.2 画圆时圆不像圆，而是扁了下去

对于公式的推导有错误，在计算过程中，由于书中算法默认原点为中心，故需要将 $x - cx$ 置为新的 x，把 $y - cy$ 置为新的 y 进行推导。

1.2.3 区域填充扫描线算法在运行时会 Segmentation Fault

需要对是否达到图像边界作判断。边界为 $0 \leq x < r$，超出边界的着色则会被判断为 Error。

1.2.4 填充颜色与原色相同时会死循环

特判即可。

2 讨论与借鉴代码

本次作业仅仅借鉴了书中相关算法的代码，并对其进行了完善与优化。

3 未解决的 bug

暂未发现，如果继续调试，我可能会采用继续构造测例的方法。

4 建议

在 instruction.pdf 中有一处小错误：第 4 部分测试用例，笑脸表情的描述中，“只对半部分就行区域填充”，应改为 “只对半部分进行区域填充”。

提供的自动化测试脚本，可以进一步满足同学们对于自定义测例快速测试的需求：在根目录 `run_test.sh` 中，我编写了一个简单的脚本，这个脚本可以自动对 `testcases` 目录下的测例进行遍历，并自动生成各自对应的图片。（为了实现简洁，图片命名与测试不同）

在 `testcases` 目录中，我还编写了两个测例。`radar` 测例可以生成一个“雷达”，虽然美观程度不高，但是它能测试出所有方向的画线，以及画圆功能是否正确实现，测试覆盖面较广；而 `android` 测例则可以生成一个安卓机器人，比较美观。两个测例的图片如下：