

《计算机图形学》4 月报告

151110079, 姚荣春, 15895873878@163.com

2019 年 5 月 30 日

1 综述

图形学大作业期中报告, 计划使用 python3+qt 来实现。目前实现了命令行界面, 可以从文本文件中读取命令执行。还实现了重置画布, 保存画布, 设置画笔颜色, 绘制线段, 绘制多边形这几个功能, 最后对性能做了简单的测试

2 算法介绍

2.1 命令行界面程序:

利用 python 实现从命令序列文本文件中读取命令并依次调取算法核心模块来绘制图形并最终保存图形为 BMP 文件

2.2 图形化界面程序:

拟利用 pyqt 实现图形化界面, 暂时未有代码实现

2.3 核心算法模块:

1. 重置画布

实现思想: 在画布类中使用 numpy 数组来保存 R,B,G 三种颜色, 重置的时候清空 numpy 数组并重新根据大小申请空间, 并调用垃圾回收函数

2. 保存画布

实现原理: 根据博客上的 BMP 格式详解直接构造 BMP 文件头和结构头, 采用 24 位色彩按照 BGR 的顺序讲画布中的数据以二进制方式写入到文件

性能测试: 循环遍历像素数组输出到文件所以比较慢, 2.4Gcpu, 1000×1000 的画布保存需要 3 秒。以后会再提升这部分性能。

参考文献 BMP 格式详解 <https://www.cnblogs.com/wainiwann/p/7086844.html>

3. 设置画笔颜色

实现思想: 根据参数直接修改画布类中的色彩

4. 绘制线段

DDA 算法实现原理: 记斜率为 r , 若 $r < 1$ 则沿 x 轴从 x_1 点开始向 x_2 点增长, 记初始 y 值为 y_1 , 所计算点所对应的 y 值为 $y' + r$ 其中 y' 为上一个点计算的 y 值, 而最终涂色的像素的 y 坐标为 $y' + r$ 的四舍五入。若 $r > 1$ 则从 y_1 开始增长 y 的值, 利用 y 去计算所对应的 x 的值。

Bresenham 算法实现原理: 同 DDA 中一样根据斜率选取利用 x 坐标计算 y 坐标或者反之。以利用 x 计算 y 为例 (并且 x 的取值按照 y 增长的方向), 因为 y 只有可能是 $y' + 1$ 或者维持 y' 不变。因此, 记 $y = rx + c$ 计算出来的 y 值为 y_{real} 在 $y' + 1 - y_{real} < y_{real} - y'$ 的情况下坐标为 $y' + 1$ 否则维持不动。所以问题变为如何确定不等式是否成立。 $r = (y_2 - y_1) / (x_2 - x_1)$ 如果 $x_2 > x_1$ 则左右同乘 $x_2 - x_1$ 否则乘 $x_1 - x_2$, 那么确定不等式是否成立只需要进行整数乘法, 从而避免了浮点数运算。

测试性能: 只做了简单的测试。Bresenham 和 DDA 在笔记本上性能差异很难看出, 因为计算时间太短很难准确计时, 这部分测试以后再进行

5. 绘制多边形

我的理解: 继承直线的类, 调用绘制直线方法进行绘制多边形

3 系统介绍

使用 python3 实现。命令行入口为 main.py, canvas.py 继承了 bmp.py 因此是具有保存 bmp 功能的一个画布类, 所有的图形也都通过相关类来实现, 比如多边形类, 直线类。画布类使用一个队列来保存画布上的所有图形类的实例, 绝大部分的功能都是通过画布类的方法来实现, 通过向队列中添加相关图形的类的实例来达到在画布中增加图形的目的。各个图形对应的一些比如平移裁剪等功能在各自类的相应方法中实现。

在核心功能都实现之后, 利用 PyQt 来实现图形化界面。

4 总结

...