

《计算机图形学》五月报告

161240056, 谈婧, jingtan@smail.nju.edu.cn

2019 年 5 月 26 日

1 综述

本系统为一个包含了图形学用户界面程序和命令行界面程序的绘图系统。本绘图系统包含如下功能函数：重置画布、保存画布、设置画笔颜色、绘制线段绘制多边形、绘制椭圆、绘制曲线、对图元平移、对图元旋转、对图元缩放、对线段裁剪。

2 算法介绍

2.1 DDA 数值分析

由于线段斜率固定，所以可以通过在 x 或 y 方向上用单位 1 长度取样，估计另一方向增量，并对其增量和取整得到近似整数像素点。设斜率为 m ，如果 $m < 1$ ，则从左往右在 x 方向上取样，每次取单位 1（如果从右往左则是-1），并通过斜率计算 $\delta y = m$ ，因此有递推式 $y_{k+1} = y_k + m$ 。对于第 k 个点， $x_k = x_0 + k$ ， $y_k = y_0 + \text{round}(k * m)$ ，如此就可以近似直线。如果 $m > 1$ ，则在 y 方向取样， $\delta x = \frac{1}{m}$ ，递推式为 $x_{k+1} = x_k + \frac{1}{m}$ ，对于第 k 个点， $x_k = x_0 + \text{round}(k * \frac{1}{m})$ ， $y_k = y_0 + k * m$ 。

2.2 Bresenham 算法

Bresenham 算法引入了一个整数参量来衡量对于当前已选像素点的下一步的两个候选像素点与实际直线中下一个路径点的偏离关系，通过对于整型参数的衡量来选择下一个像素点。

假设在第 k 步确定了像素位置为 (x_k, y_k) ，那么有整型参数 $p_k = \delta x(d_1 - d_2) = 2\delta y x_k - 2\delta x y_k + c$ （ d_1 为 y_k 到 y 的距离， d_2 为 y_{k+1} 到 y 的距离），也称为决策参数。若 $p_k > 0$ ，即 $d_1 > d_2$ ，说明 y_{k+1} 比 y_k 更靠近线段，选择 (x_{k+1}, y_{k+1}) 。反之 $p_k < 0$ ， y_k 比 y_{k+1} 更接近线段点，选择 (x_{k+1}, y_k) 。

对于左端点为 (x_0, y_0) ，斜率小于 1 的情况：沿着 x 方向间隔单位 1 取样，在每选中像素后都计算决策参数，进行下一个像素的选择。

决策参数可以靠增量公式计算， $p_{k+1} - p_k = 2\delta y \cdot (x_{k+1} - x_k) - 2\delta x \cdot (y_{k+1} - y_k)$ 。因此如果在决策时 $p_k > 0$ 就在取高像素 (x_{k+1}, y_{k+1}) 的同时将决策参数的值增加 $2\delta y - 2\delta x$ ；如果 $p_k < 0$ ，那么 p_k 增加 $2\delta y$ 。

取样的方向和增长趋势和 DDA 的情况一致。

2.3 ...

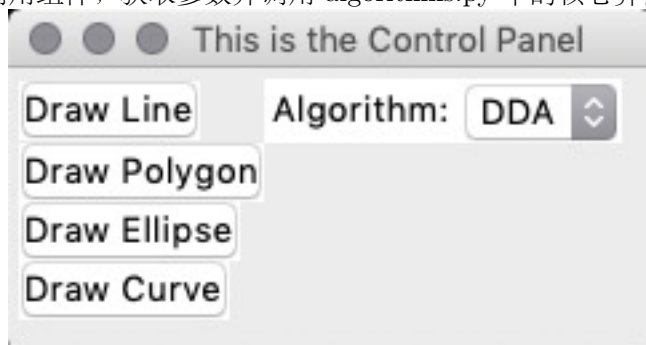
...

3 系统介绍

本系统使用 python3.6 开发，图形学用户界面程序由 python tkinter package 支持开发。本系统中的绘图核心算法在 ‘/algorithms.py’ 中实现。对于每个绘图算法，除了必要的绘图所需参数传递外，增添了画布参数，方便确认需要图形绘制的特定画布。由于图形学用户界面不会和命令行界面发生交互，因此在本系统中图形学用户界面和命令行界面的实现过程中会各自自行定义画布。

3.1 图形学用户界面设计说明

本用户界面包含了一个 App 类，类在初始化时会调用每个绘图函数方法，实现各种函数调用组件，获取参数并调用 algorithms.py 中的核心算法绘图。界面预览如下图：



本界面主要包含两个 Tk 窗口，分别为 Panel 和 master。Panel 中包含函数调用按钮，master 则为绘图展示窗口。Panel 的设计结构中包含一个 Frame 组件 (PanelFrame)，用于承载各种零碎部件。在 PanelFrame 中主要包含 Button、Label 和 OptionMenu 组件，其中 Button 组件用于完成算法执行按钮的设置，位置分布在最左边一栏；Label 组件，用于提示系统使用步骤，提高用户友好度；OptionMenu 组件用于设计绘图所需 Algorithm 的下拉菜单。master 中主要包含一个 Canvas 控件，用于展现绘图结果。图形绘制时关键顶点的确定需要依靠鼠标交互，因此在设计时利用了 tkinter 的 bind <B1-Motion> event 和 <Button-1> event 来确定鼠标位置和移动轨迹。用栈 clickList 和 MotionList 来保存位置变化和移动轨迹。

4 总结

...

5 参考文献