

JGraphics

系统技术文档

131242023

李峻凡

1 引言

1.1 编写目的

说明本项目用到的关键技术，以及对本项目进行的测试过程与结果。

1.2 背景

本学期我们学习了计算机图形学，一定程度上了解了各种图形绘制、编辑的算法，了解了计算机显示图形的原理。为了更好地掌握所学知识，我希望设计并实现一个图形数据编辑系统。

为了使本项目能够稳定运行，我请一些志愿者进行了相关测试，并修复了发现的一系列 bug。

1.3 参考资料

1. 《计算机图形学教程》
2. 计算机图形学教学课件 孙正兴
3. 技术博客：扫描线填充算法(<http://blog.csdn.net/orbit/article/details/7368996>)
4. 技术博客：递归种子填充算法(<http://blog.csdn.net/orbit/article/details/7323090>)
5. 技术博客：Bezier 曲线原理(<http://blog.csdn.net/joogle/article/details/7975118>)
6. 技术博客：基于 Win32 应用程序的 OpenGL 环境
<http://blog.csdn.net/smstong/article/details/50277427>
7. 技术博客：OpenGL 状态、视口设置
<http://blog.csdn.net/smstong/article/details/50277619>
8. 技术博客：OpenGL 入门学习
<http://www.cppblog.com/doing5552/archive/2009/01/08/71532.html>
9. 技术博客：Win32 静态链接库
(<http://blog.csdn.net/xiexiaopingroma/article/details/5402480>)

2 关键技术

2.1 技术概要

本项目没有采用更为方便的 glut 框架，原因是 glut 框架不方便提供人性化的用户界面。我使用了 win32 应用程序框架，以标准 windows 应用程序界面展现给用户，并由 Windows API 管理所有的交互事件。

同时，将主面板与 OpenGL 绑定，让 OpenGL 在主面板进行绘制。

2.2 关键设计

1. 全局变量

- `vector<PicElem*> vec`
 - `PicElem` 是所有形状的父亲类，同时也是裁剪、填充操作的父亲类。
 - `vec` 中按顺序存储了每次绘制的图形
- `Mode mode`
 - 用户在绘制不同图形时，处在不同的模式下，变量 `mode` 与之关联
- `Color** pixels`
 - 一个动态二维数组，保存了每个像素点的颜色

2. 绘制策略

多数情况下，直接将 `pixels` 存储的点阵绘制到屏幕。这样做可以保证复杂度比较低（绘制窗口的长*宽），且复杂度与绘制图形的数量无关。否则，如果仅存储图形，使用 OpenGL 直接绘制线段、多边形等，随着绘制的图形增加，复杂度会变得很高，将会产生明显延迟。

特殊情况说明：

用户在绘制线段时，拖动选择端点，此线段是一直移动的(类似‘画图’中的线段绘制)。此时刷新频率很高，如果将这些点同步到 `pixels` 再绘制点阵，每次刷新时先擦除再更新再绘制，会造成明显延迟。因此，这些时候调用 OpenGL 的绘制线段的功能进行绘制。当鼠标停下时再同步到 `pixels` 数组中。

3. 存储策略

每种形状的都重写 `output()` 函数，返回一个 `float output[]` 数组。`output` 数组的第一个成员是数组长度，第二个成员是形状类型，后面是这个形状的相关数据。

每种形状的都重写 `input()` 函数，可以解析自己生成的 `output[]` 数组。

存储时，首先写入一个 `Int` 类型的 `magic number` 来确定文件类型，然后按 `vec` 中的图形顺序，依次调用 `vec[i]->output()`，存储得到的数组。

读取时，首先读取 `magic number`，如果不是设定的 `magic number`，报告错误。如果是，则进行解析。

3 测试

3.1 测试概要

测试比较简陋，要求每位志愿者自由组合使用各个功能，并报告发现的 bug。

3.2 测试结果

1. 初步测试
缺少 MSVCP140.dll，无法运行。
解决方案：开启 /MT 选项，静态链接。
2. 灰度测试
 - 在一些模式下，点击 `clear`，再在主面板进行点击会崩溃。(fixed)
 - 裁剪模式，绘制了裁剪多边形后不进行裁剪，切换到其它模式,遗留的红色多边形可能会造成错误。(fixed)
 - 裁剪模式，裁剪窗口边框(红色)覆盖了绿色点后，再将裁剪窗口移开，绿色不会恢复。(fixed)
 - 裁剪模式，对某些顶点判断错误。(fixed)
 - 由于使用了 `int` 表示坐标，将多边形过度缩小再放大会变形。(fixed:使用了 `float` 表示坐标，对坐标变换，存储过程进行了较大改动)
3. 发布