

计算机图形学 Project1

《四边形网格扫描转化与交互式编辑》文档

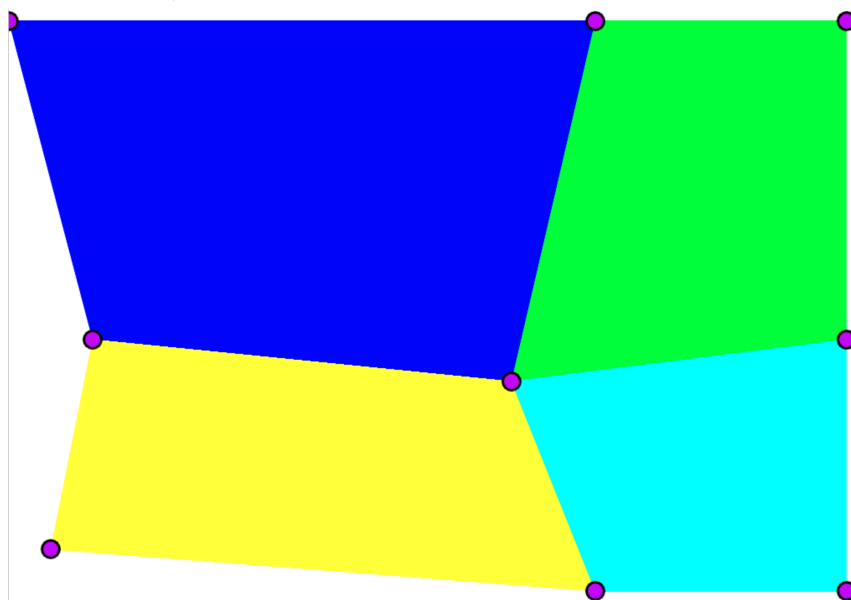
20SS 19307110374 罗俊韬

一、项目目录及文件说明

- src/config.js: JSON格式的配置文件，包含默认配置以及我的程序中用到的常量。
- src/scanConversion.html: 本次PJ的HTML文件，我将原本内嵌其中的扫描转换相关JavaScript代码提取成了单独的JS文件。
- src/implement.js: 提取出来的扫描转换相关JavaScript代码，其中函数被分为以下几部分：默认自带的画点、线函数；我自己定义的edge, rect数据结构；实现扫描转化和交互所需的计算相关utility函数；扫描转化和填色相关函数；交互相关函数以及顶层函数。

二、开发、运行环境及使用方法

- 开发环境：VScode + Chrome/Safari
- 运行环境：支持HTML5和webgl的浏览器
- 使用方法：用支持HTML5和webgl的浏览器打开src/scanConversion.html文件，鼠标点击并拖动紫色圆形顶点即可交互，可任意改变顶点位置、四边形形状来测试扫描转化算法效果。



三、项目亮点

- 定义了明确的色块覆盖优先级顺序，可以预测交互显示的结果，方便debug。同时函数功能清晰、注释明确、层次分明，代码可读性较好。

四、遇到的问题

1. 初次接触canvas带来的坐标系问题：第一次使用canvas绘制图形，惯性思维地认为坐标系原点在左下角，导致在计算和比较坐标大小时经常弄反。

解决方法：在发现canvas坐标原点在左上角，y越大越靠下之后，重新修改了相关代码。

2. 绘制顺序带来的问题：一开始绘制四边形和顶点时没有考虑到像素覆盖的问题，比如先画了vertex再填色，就会导致vertex被遮挡。同时，交互后不同四边形之间的覆盖优先级也没有定义明确，表现为自己的代码交互运行结果与sample不完全一致。

解决方法：首先调整了vertex和矩形的绘制顺序。覆盖优先级方面，在咨询了助教后，明确任意覆盖优先级均可，于是自定义了优先级（用矩形下标表示）：

默认：0 < 1 < 2 < 3

交互后，以被拖动点为顶点的矩形 > 其他矩形，同类矩形内仍然按照下标大小排序

3. 计算交点时的精确度问题：Javascript在进行浮点数运算时存在精确度问题，如果简单地使用“==”来判断，很可能导致交点个数计算出错。在debug发现这个问题后，我用 `Math.abs()` 和科学计数法等方式避免了精度影响。

五、存在的缺陷

- 数据结构复杂，性能不友好：在完成本次PJ时，我是根据自己对扫描线算法的理解以及PJ文档中的实现步骤引导，自行编写的代码。在完成后，通过上网查阅资料，以及查看课程群内同学分享的教学资料，我发现自己设计的edge和rect等数据结构太重，且遍历所有边的渲染形式性能不佳。实际大型应用中应该学习应用活性边表、新边表等数据结构。

六、对Project的意见和建议

无