**Assignment 2 – 扫描线**

519021910475 王廷宇

1. **概述**

实现带深度检测的扫描线填充算法，绘制给定文件中的三角形，以及自行设计的编织图案。

1. **程序使用说明**

（在Loading结束后）

鼠标左键——下一页

鼠标右键——上一页

Esc——关闭窗口

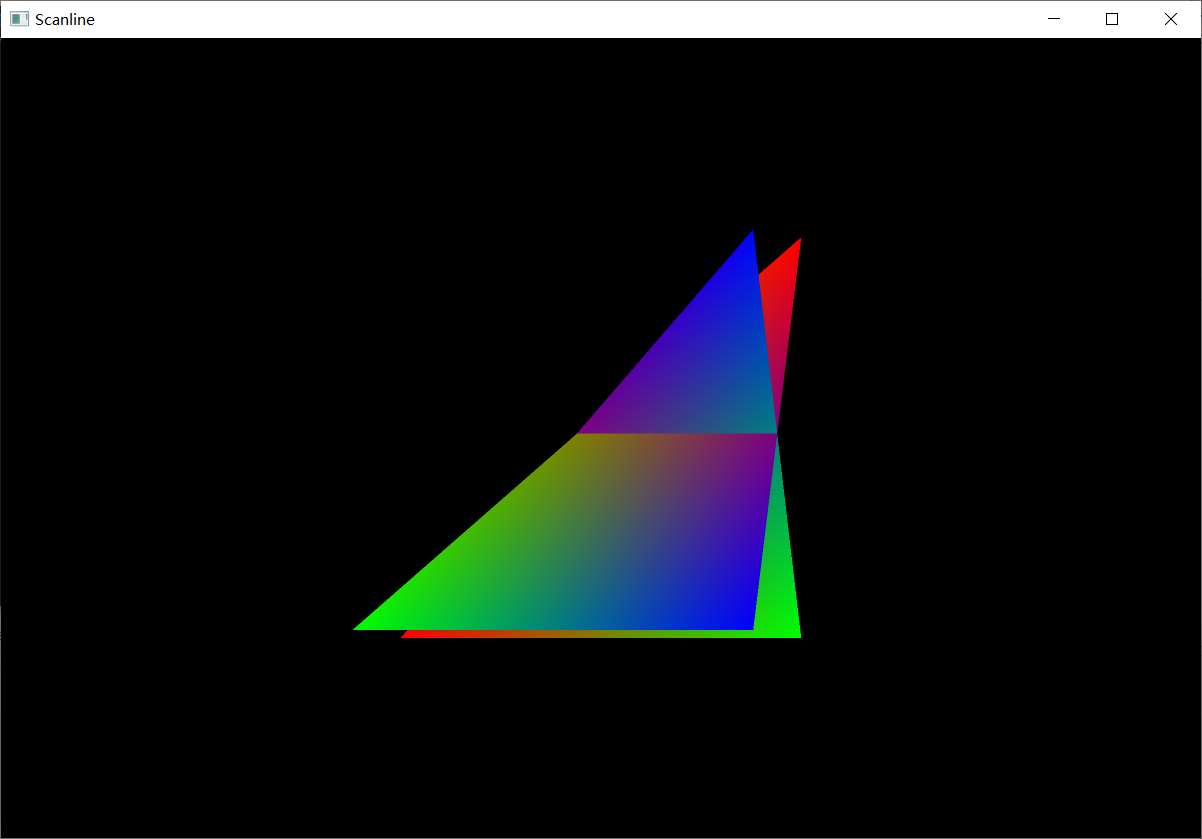
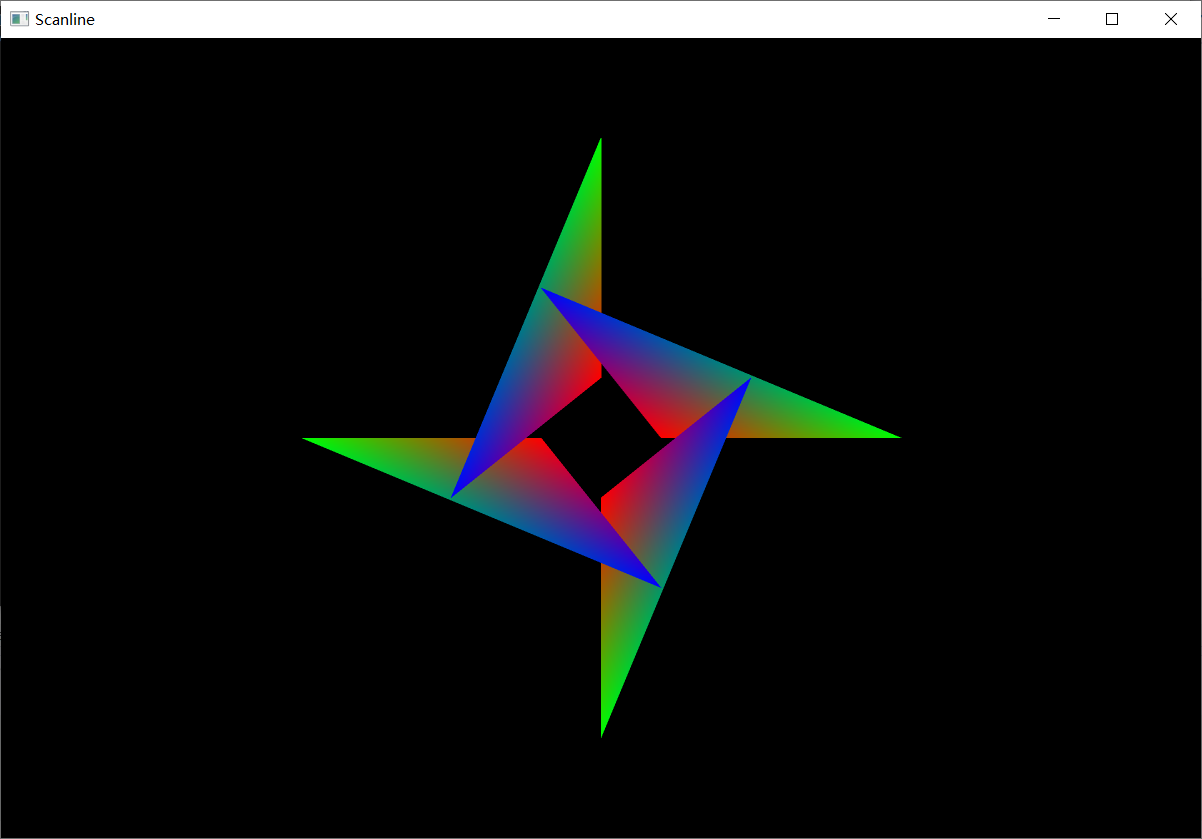
1. **具体工作与技术方案**
2. 扫描线填充算法

首先按照边起始的y值建立分类边表，然后传入scanline()函数中，利用活动边表，逐行逐点绘制图像，大致思路如下：

对于每一行：

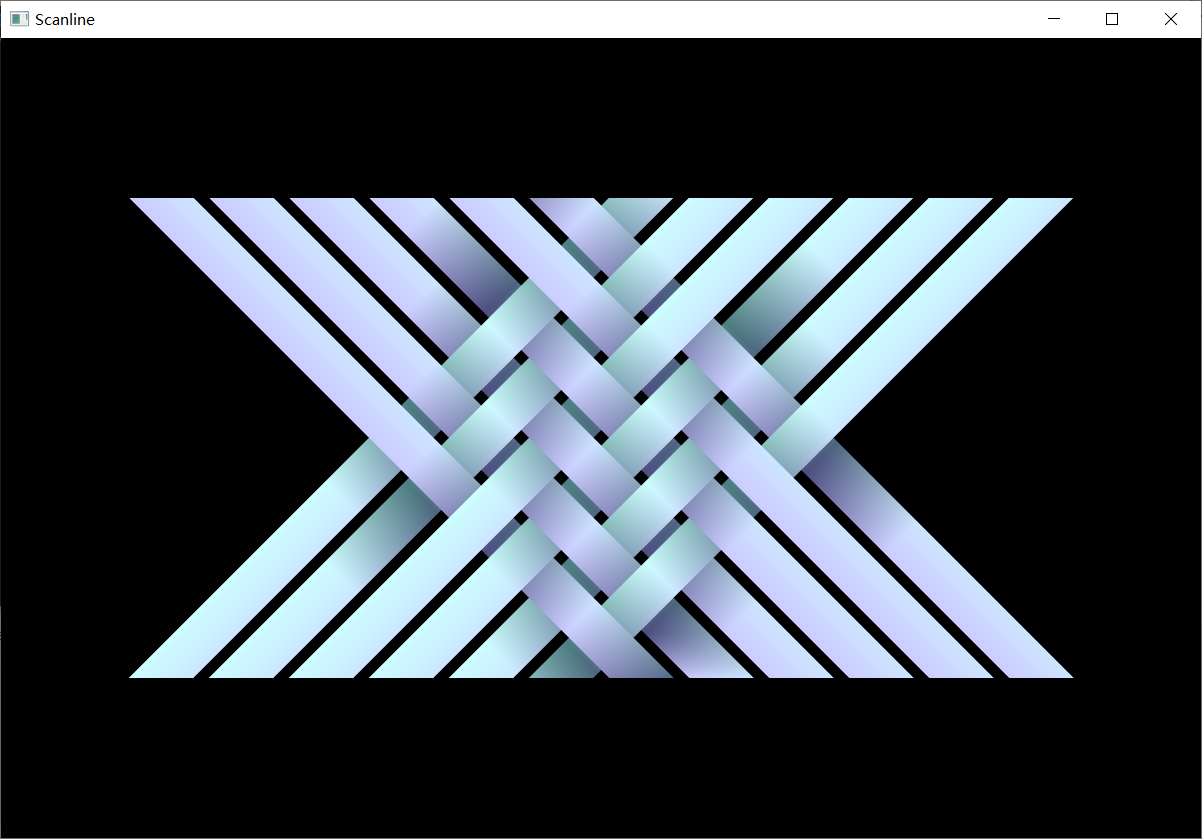
1. 移除最大y值小于当前y值的边
2. 插入排序，将分类边表中当前y值对应的边，按边的当前x值升序插入活动边表中的适当位置
3. 绘制行，创建当前行的次级分类边表（按起始x值分类储存边配对后形成的在XZ平面上截得的次级连边（SubEdge）），对于行中的每个点：
   1. 移除最大x值小于当前x值的次级边
   2. 插入排序，将次级分类边表中当前x值对应的边，按边的当前z值降序插入次级活动边表中的适当位置
   3. 绘制点，若次级活动边表非空，将次级活动边表首位的边对应的颜色写入颜色缓冲的对应位置
   4. 更新边的状态，将次级活动边表中的每一条边的z值和颜色值，增量式更新到下个点位（x = x + 1）对应的状态
   5. 冒泡排序，将更新后的边按当前z值降序调整顺序
4. 更新边的状态，将活动边表中的每一条边的x、z值和颜色值，增量式更新到下个点位（y = y + 1）对应的状态
5. 冒泡排序，将更新后的边按当前x值升序调整顺序
6. 给定三角形图案的绘制

从给定的.tri文件中读取顶点数据，由顶点数据构建点对象（Point）并调整图像位置、缩放比例和观察视角（根据z值按一定比例偏置x、y的值）；然后由点构建边对象（Edge），使得沿边方向y严格增大，y相等的两点连边将被忽略；最后将边插入分类边表并交由scanline()处理绘制。



1. 编织图案的绘制

通过构建12个深度值周期线性变化的平行四边形编织带，绘制形成编织图案；设置颜色随深度变化，以实现简单的阴影效果；设置每个编织带的左右颜色不同，以模拟简单的材质。公式及效果图如下：



1. 超采样抗锯齿（SSAA）

为了实现简单，本次作业采用SSAA进行反走样处理，基于性能考虑，只使用2x2的超采样。

1. 预渲染

由于自行实现的扫描线算法需要做大量计算并逐点编辑颜色缓冲，效率较低，故采用预渲染的方式，即事先执行扫描线算法将图案渲染到一张纹理中，在渲染循环中直接将预渲染好的纹理绘制到屏幕，避免了重复的计算。

1. Loading界面

预渲染阶段需要等待数秒，时间较长，故添加简单的Loading界面。引用了FreeType库进行文字的显示，将字体加载到自建的Font对象中，在屏幕上绘制合适的矩形并将字形纹理渲染于上。效果如下：

