

本科生实验报告

实验课程:\_\_\_\_\_\_\_\_\_\_\_\_\_\_计算机图形学\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

实验名称:\_\_\_\_\_\_\_\_\_\_\_\_\_\_assignment 0\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

专业名称:\_\_\_\_\_\_\_\_\_\_\_\_计算机科学与技术\_\_\_\_\_\_\_\_\_\_\_\_\_\_

学生姓名:\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_李骏豪\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

学生学号:\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_21307359\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

实验地点:\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

实验成绩:\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

报告时间:\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

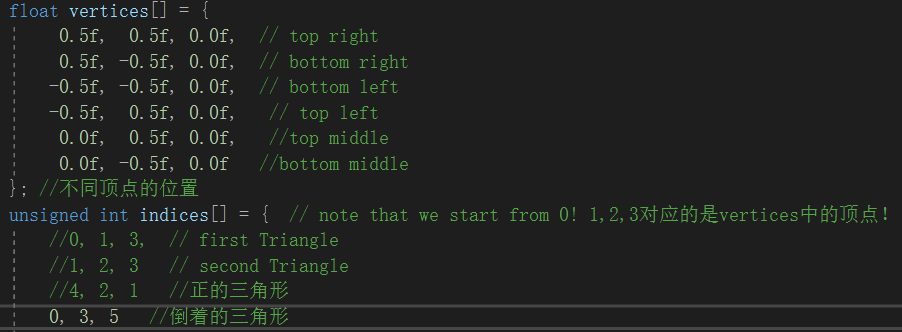
Task 1、什么是OpenGL？OpenGL与计算机图形学的关系是什么？

OpenGL（Open Graphics Library）是一个跨平台的图形库，用于开发 2D 和 3D 图形应用程序。它是一种底层的 API，通常与高级图形管道框架（例如 Unity 和 Unreal Engine）一起使用。

计算机图形学是一个广泛的领域，涉及各种技术和算法，用于从数据中创建和操作图形。OpenGL 是计算机图形学的一个重要组成部分，它提供了一个标准的方式来呈现 3D 场景，是当前最为流行的用于实现硬件加速的 3D 图形库之一。利用OpenGL，图形程序员可以使用这个API去创建各种视觉效果，例如线框模式、纹理、照明、阴影等。通过OpenGL，程序员可以直接控制图形硬件，并极其方便地实现非常高效的图形渲染。同时，OpenGL的跨平台性使得程序员能够在不同的操作系统中开发运行相同的图形应用程序。

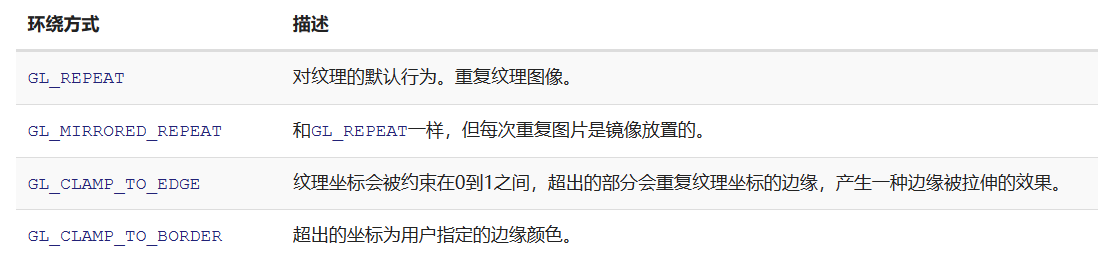
Task 2、完成了着色器章节之后，请修改顶点着色器让三角形上下颠倒。

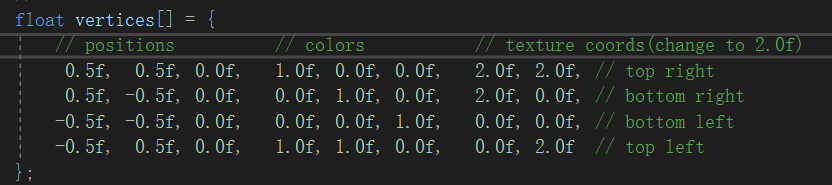
如下图修改顶点即可：



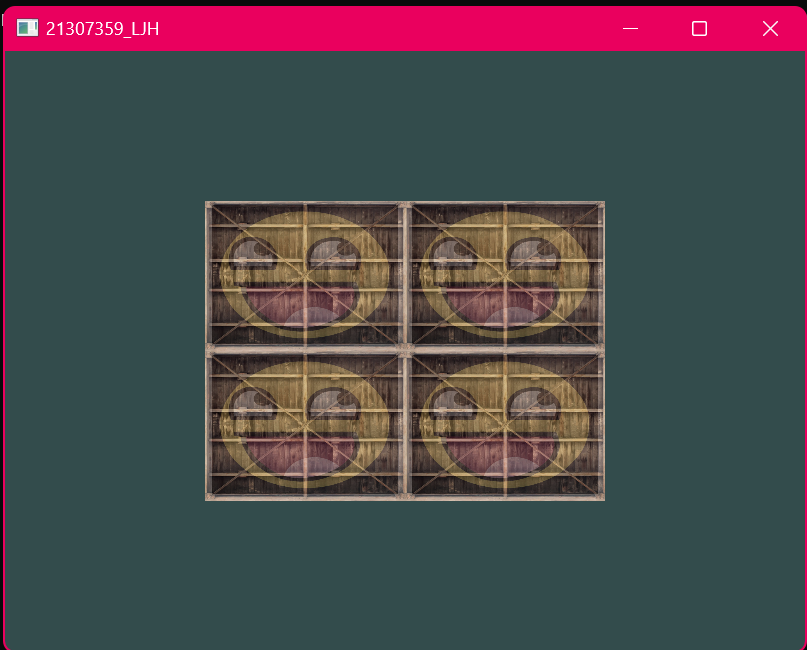


Task 3、完成了纹理章节之后，尝试用不同的纹理环绕方式，设定一个从 0.0f 到 2.0f 范围内的纹理坐标。试试看能不能在箱子的角落放置4个笑脸。简述原因并贴上结果。

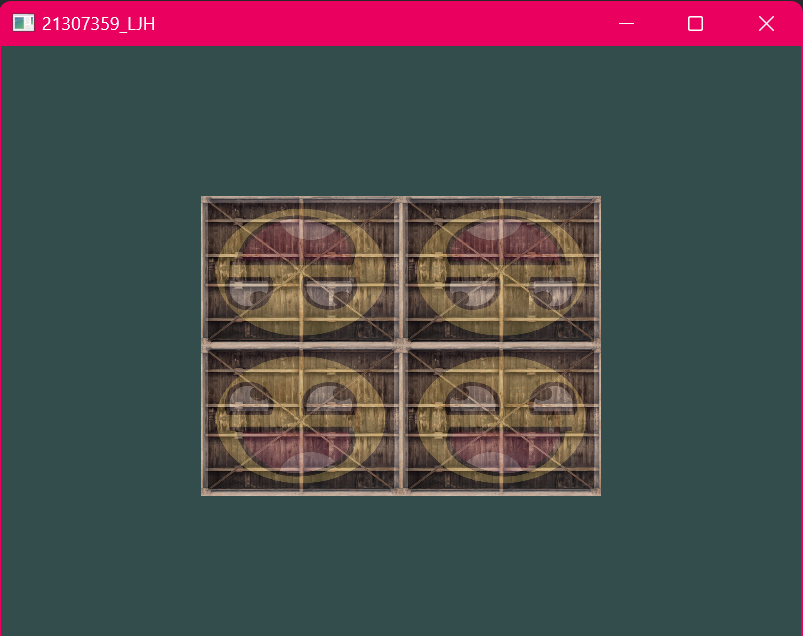




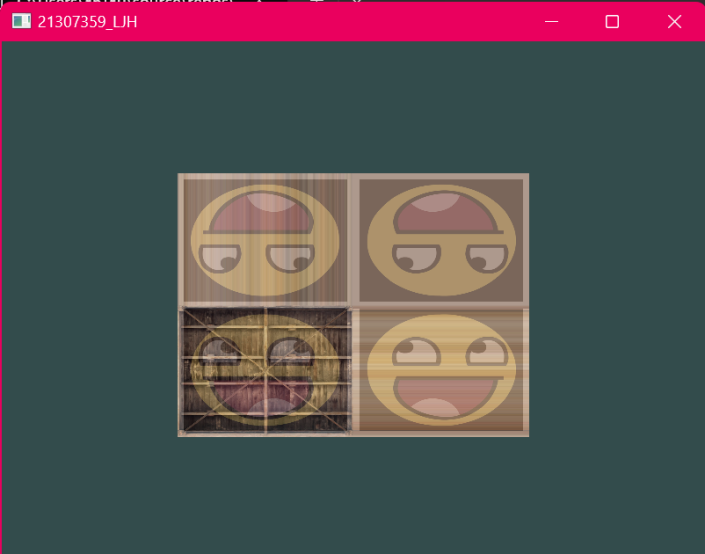
使用GL\_REPEAT:



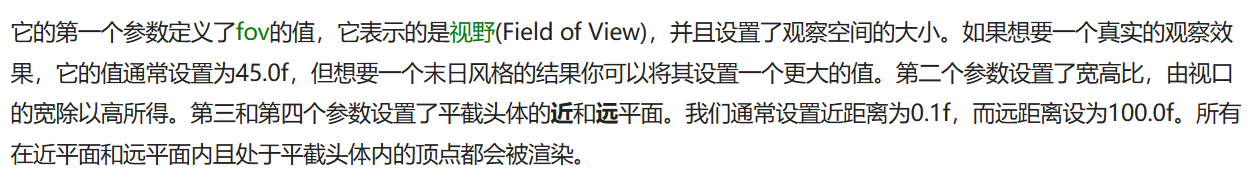
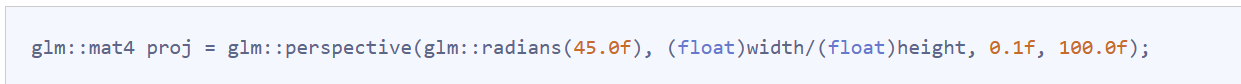
对笑脸使用GL\_MIRRORED\_REPEAT:



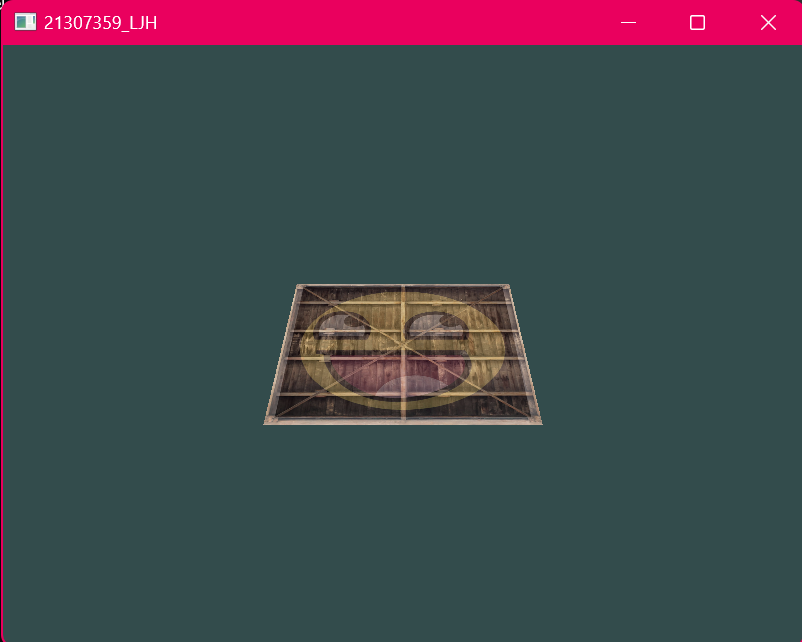
再对木箱使用GL\_CLAMP\_TO\_EDGE：

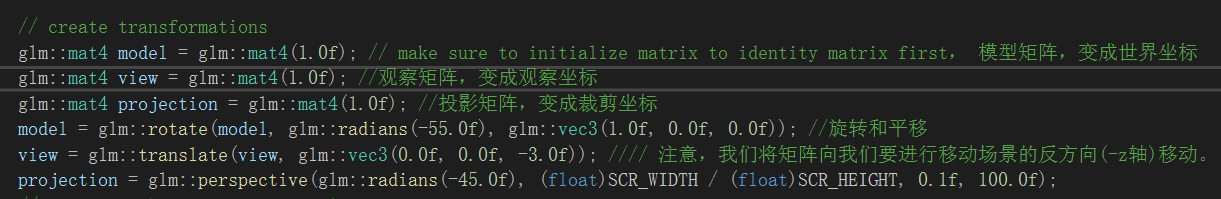


Task 4、完成了坐标系统章节之后，对GLM的projection函数中的FoV和aspect-ratio参数进行实验。看能否搞懂它们是如何影响透视平截头体的。

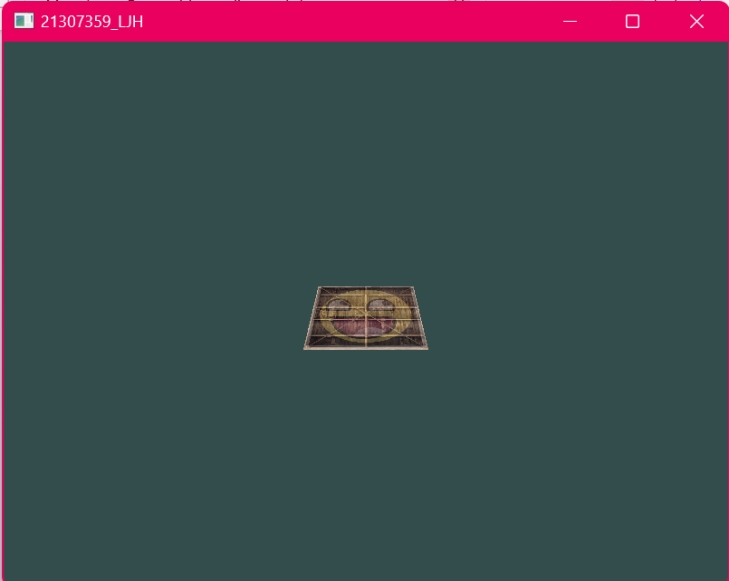


正常的变化结果如下：

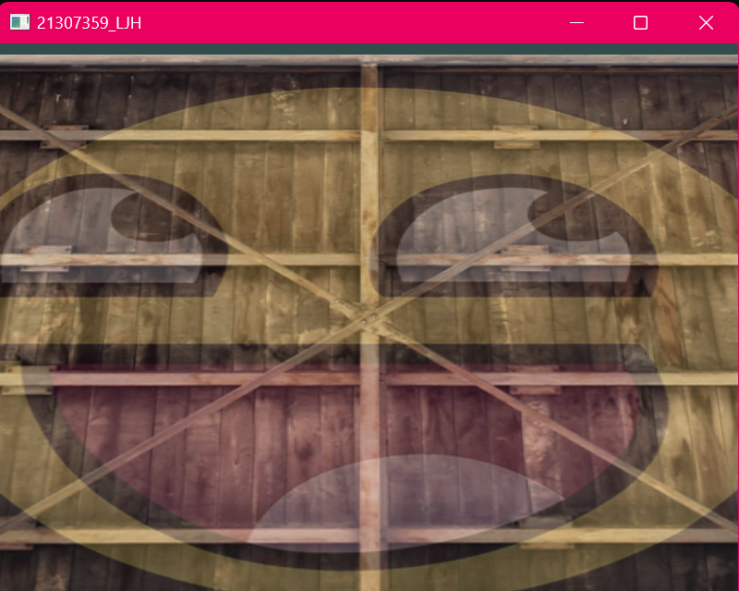




将FOV修改为80.0f后：



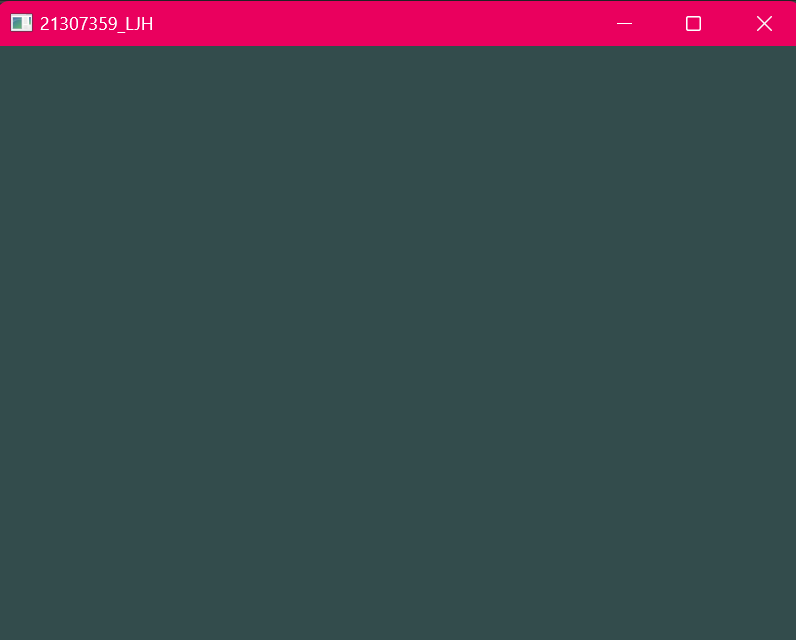
而当FOV改成10.0f后：



可见其影响着我们视野的宽阔程度。

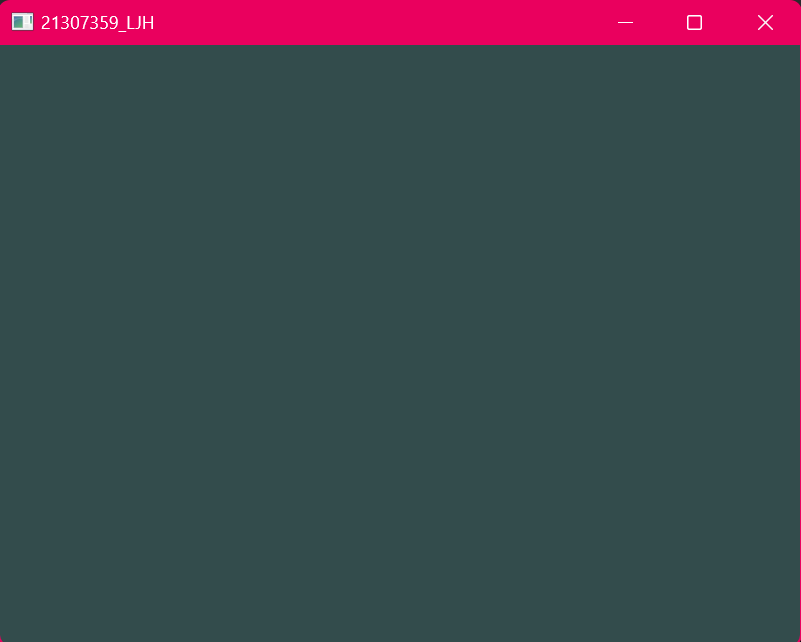
而当将把透视矩阵的 near 值设置太大时（如10.0f），OpenGL会将靠近摄像机的坐标（在0.0f和10.0f之间）都裁剪掉，这会导致（在太过靠近一个物体的时候）视线直接穿过去。

于是什么都看不见（毕竟我们只有一个箱子）：

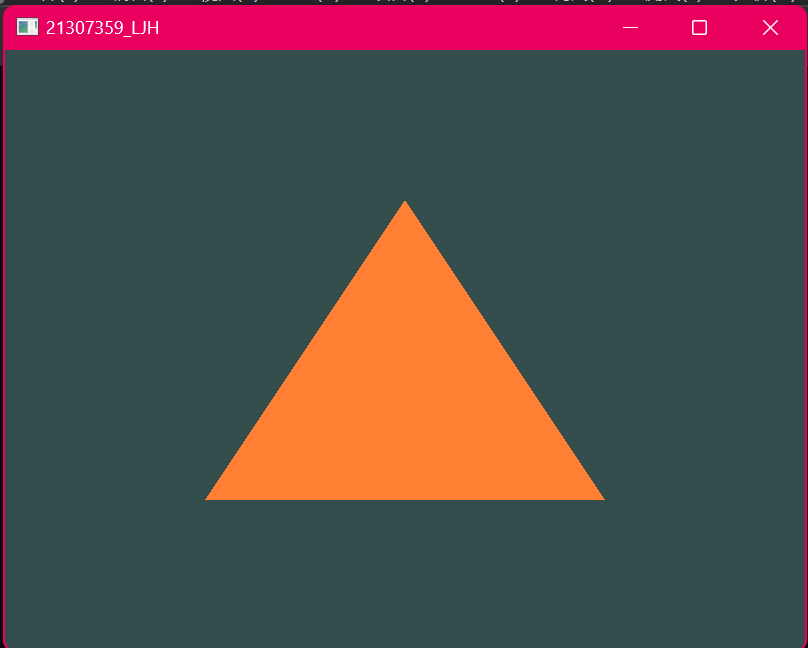


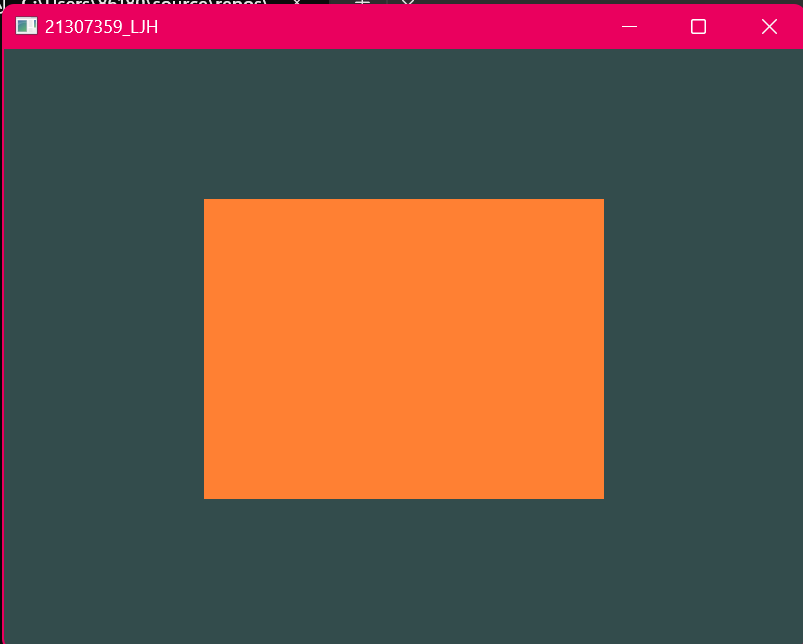
Task 5、请按照顺序将跟着教程实现的运行结果贴出来，要求将运行出来的窗口的标题改成自己的学号。(Tip: glfwCreateWindow函数）

1. 你好，窗口

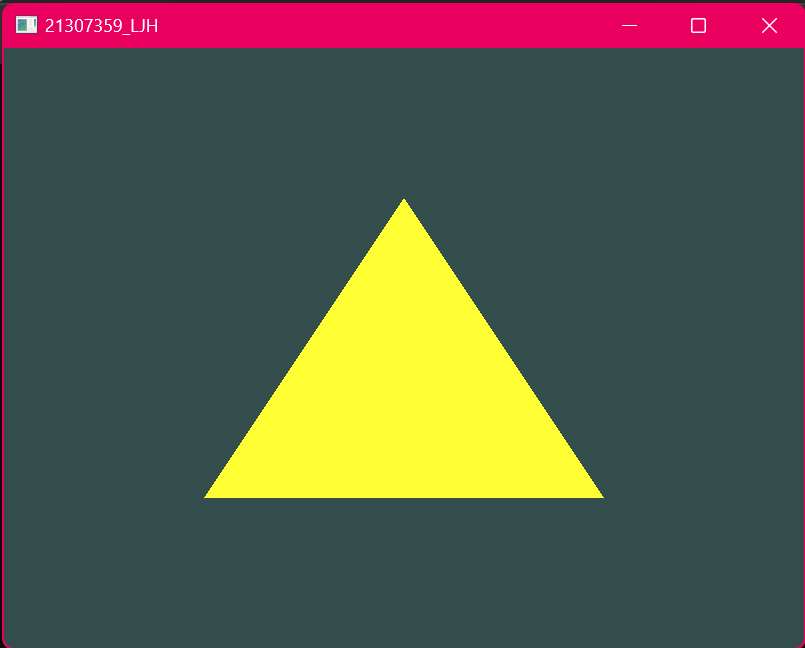


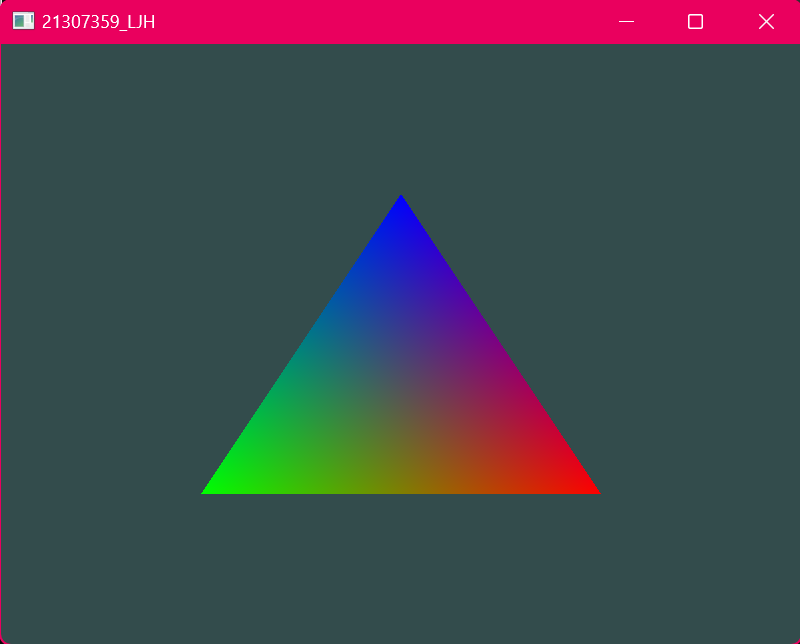
1. 你好，三角形





1. 着色器 (尝试了红橙黄颜色的变化，而不是原来的绿色)



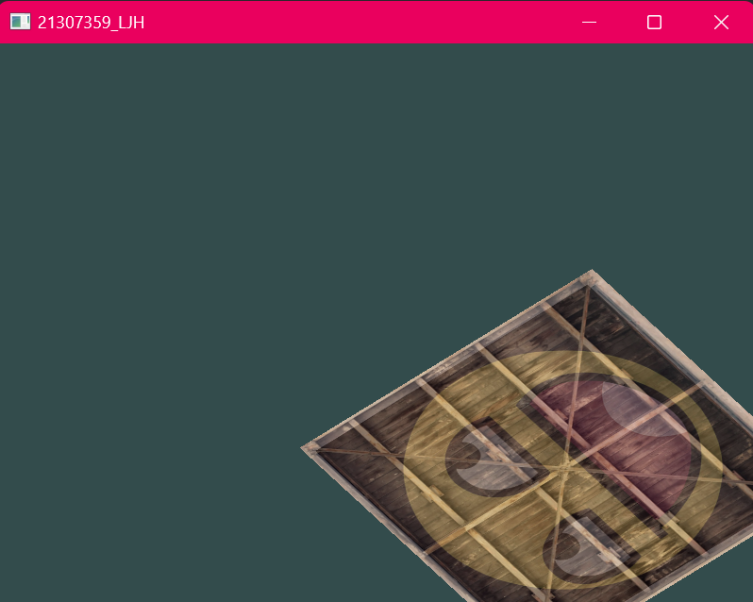


1. 纹理

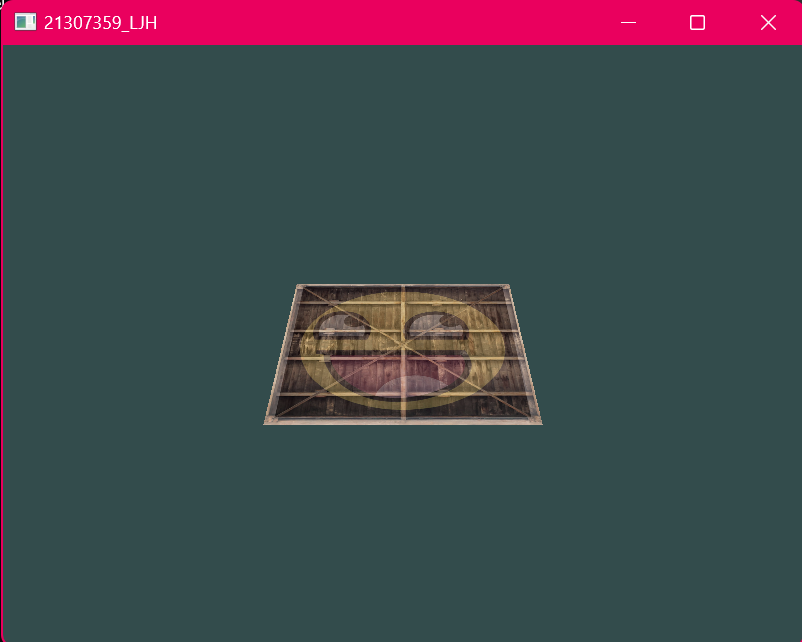




1. 变换

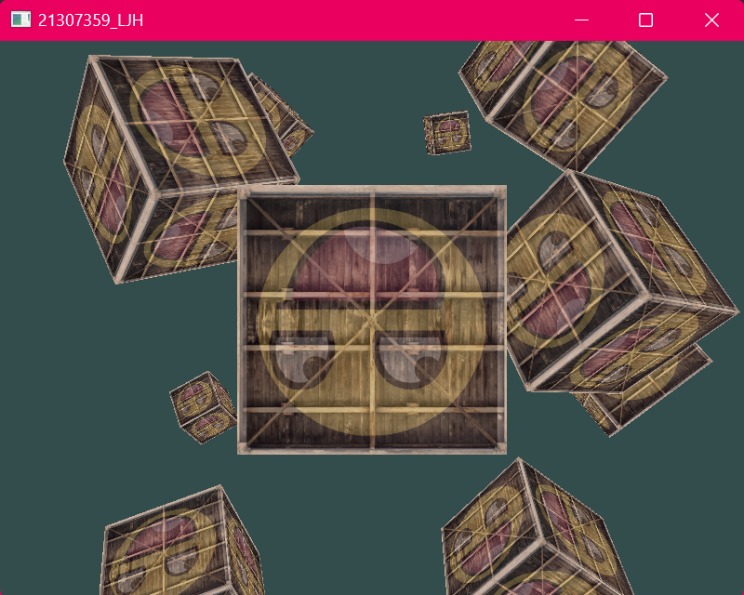


1. 坐标系统





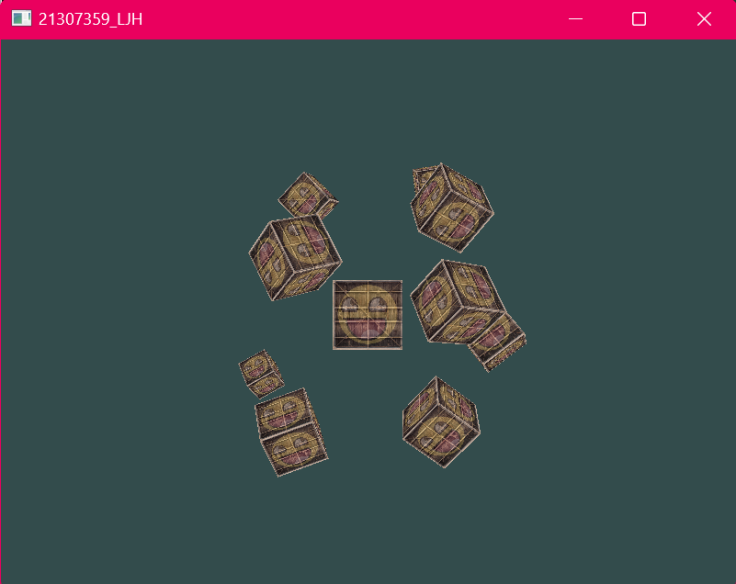
立方体的图形布局已经定义好了，所以当渲染更多物体的时候我们不需要改变我们的缓冲数组和属性数组，我们唯一需要做的只是改变每个对象的模型矩阵来将立方体变换到世界坐标系中。

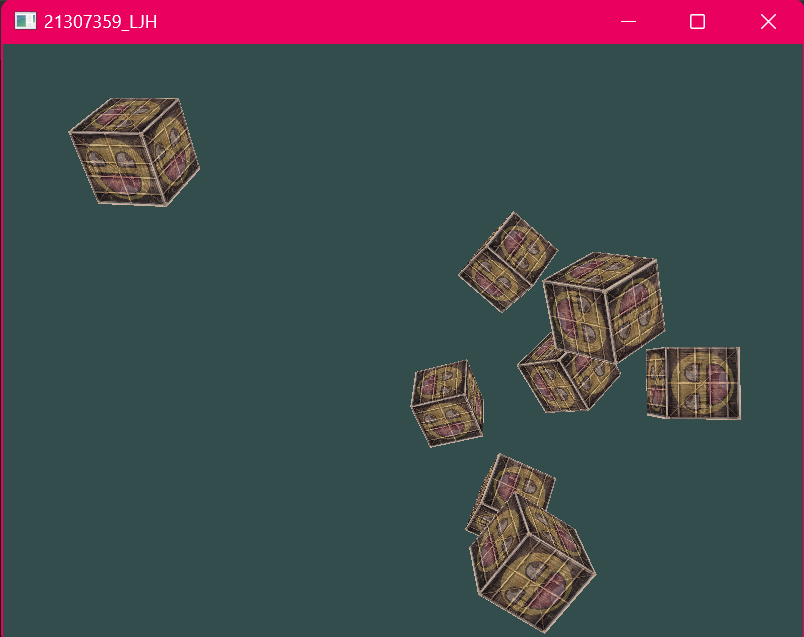


（btw，要是调整fs文件中的FragColor = mix(texture(texture1, TexCoord), texture(texture2, TexCoord), 0.2);将最后一个表示纹理混合权重的参数改为0.8，则会得到下面的丑陋结果）



1. 摄像机





**至此，入门教程已经全部完成！**