

计算机图形学实验

实验2、三角形绘制

2024年 4月18日

目录

[Task1：示例程序运行 3](#_Toc164893431)

[2.随机颜色. 3](#_Toc164893432)

[Task2: 绘制三维物体 7](#_Toc164893433)

[1. 7](#_Toc164893434)

[2.正方体旋转 8](#_Toc164893435)

[a.深度缓冲区 8](#_Toc164893436)

[b.正方体自己旋转 10](#_Toc164893437)

[c.交互输入 14](#_Toc164893438)

[Task3 18](#_Toc164893439)

# Task1：示例程序运行

## 2.随机颜色.

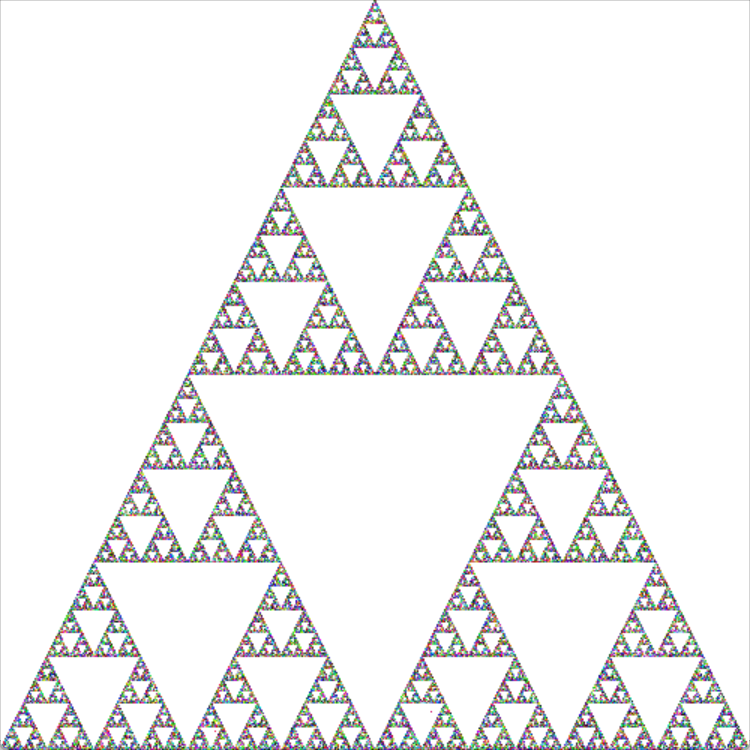
完成

操作流程：在绘制点时，随机产生颜色。

代码片段：



运行结果截图：



3.颜色随时间变化，旋转，缩放

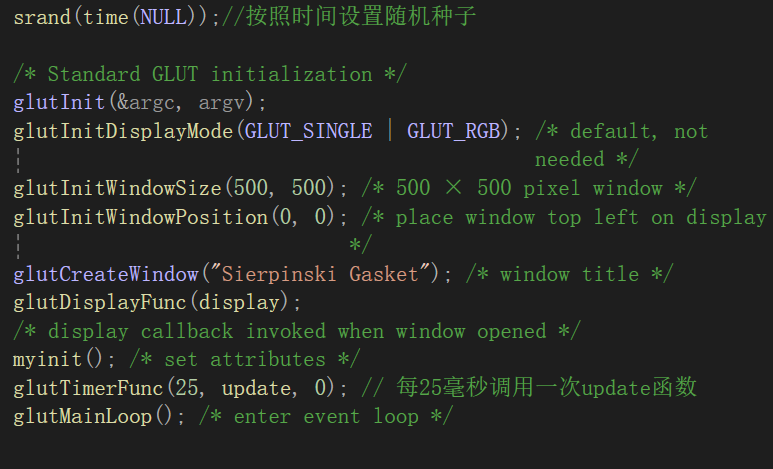
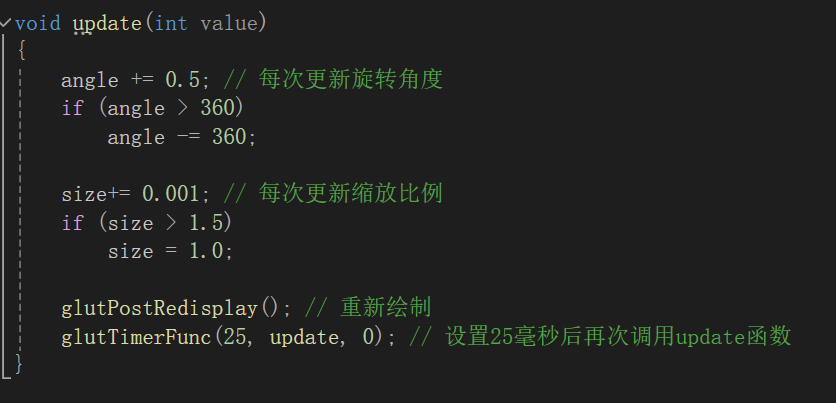
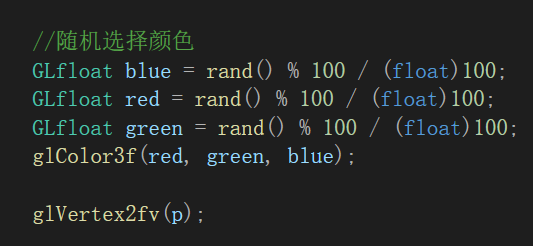
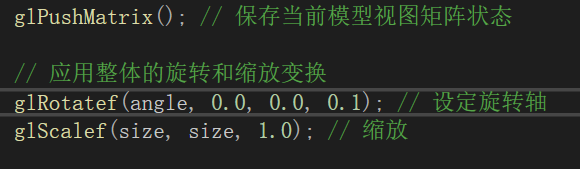
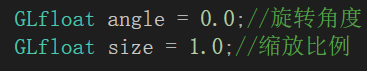
完成

操作过程：

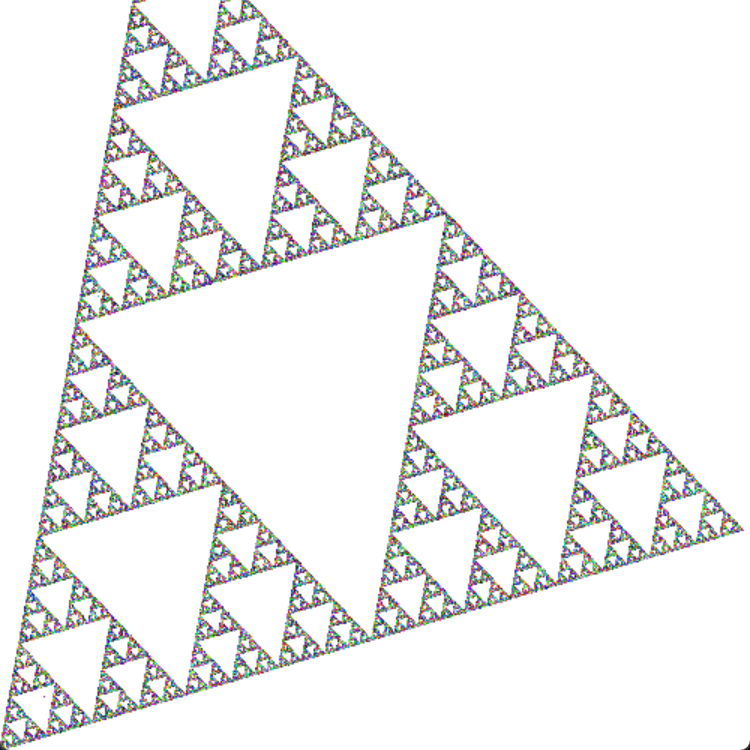
1.设定旋转角度和缩放大小的循环

2.在绘制时先压入栈保存视图矩阵状态。利用glRotatef设定旋转轴，glScalef设定旋转大小，颜色部分同上一小题。

关键代码：



运行结果：



# Task2: 绘制三维物体

## 1.

完成

思考题

* OpenGL中，三维空间的坐标系是怎么样的？

答：右边为x轴，上边为y轴，指向屏幕外为z轴。是以笛卡尔坐标系为基础的左手直角坐标系。

* OpenGL中，相机的方位是怎么样的？如何调整相机朝向呢？

答：相机方位由相机放置点和观察点组成。

OpenGL中使用gluLookAt()控制相机点和观察点和角度组成的矩阵。前三个控制相机点的位置，分别控制x,y,z轴。4-6个控制观察点的位置坐标，分别为x,y,z轴。7-9控制相机的方向，又或者为控制旋转相机角度。upX,upY,upZ组成一个指向相机顶的向量。其中upZ为了不定义出一个平行于x轴或y轴的向量。

* OpenGL中，相机可见范围是如何设置的？

答：使用gluPerspective()控制可见范围。第一个参数控制范围的高度，第二个控制范围的宽度。第三个控制近剪切面，第四个控制远剪切面。

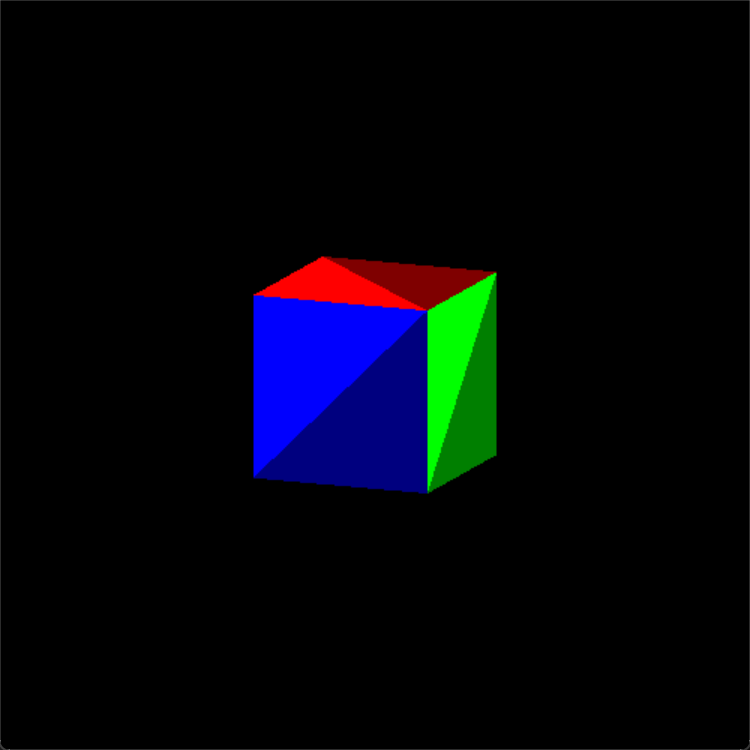
## 2.正方体旋转

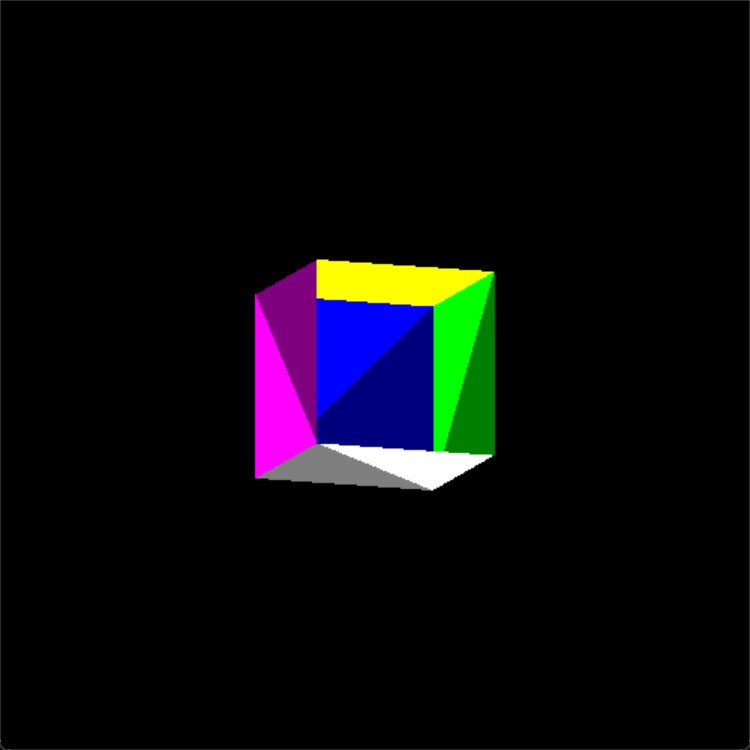
### a.深度缓冲区

完成

操作流程：开启/不开启深度缓冲区 glEnable(GL\_DEPTHTEST)

运行结果截图：打开深度缓冲区：



关闭深度缓冲区：

深度缓冲区作用：是物体按照远近（z值）正常显示，防止后面的物体后渲染遮盖住了已经渲染好的部分。

用法：开启 glEnable(GL\_DEPTHTEST)。并在后面使用glClear(GL\_COLOR\_BUFFER\_BIT | GL\_DEPTH\_BUFFER\_BIT);在正式绘制的时候需要清除颜色和深度缓冲区。

### b.正方体自己旋转

完成

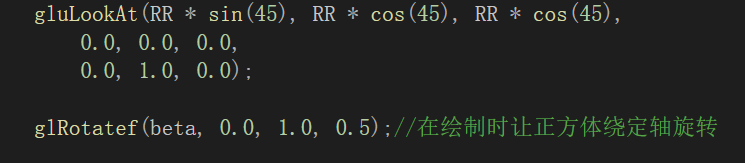
操作流程：1.将相机角度固定

2.利用glRotatef确定正方体的旋转轴，绘制正方体的6个面

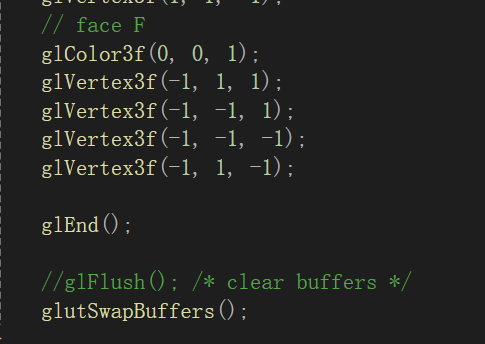
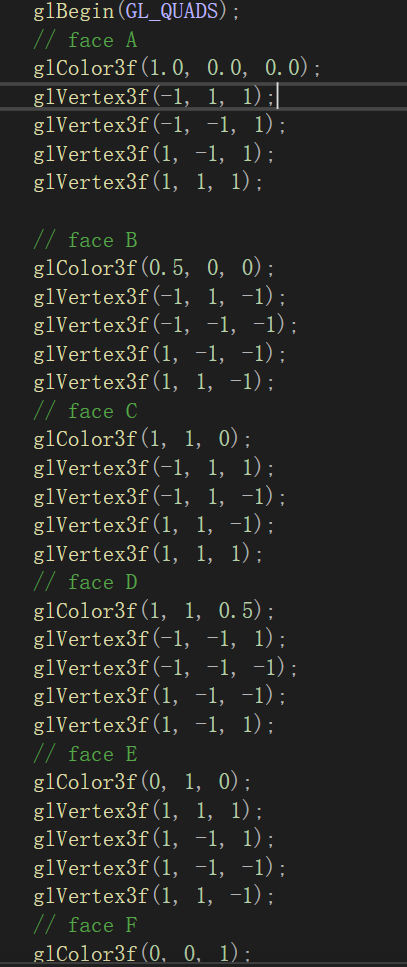
3.循环改变旋转角度的大小

关键代码截图：

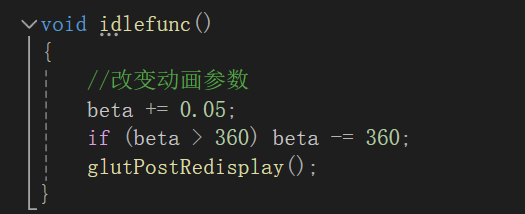
固定相机角度，设置旋转轴

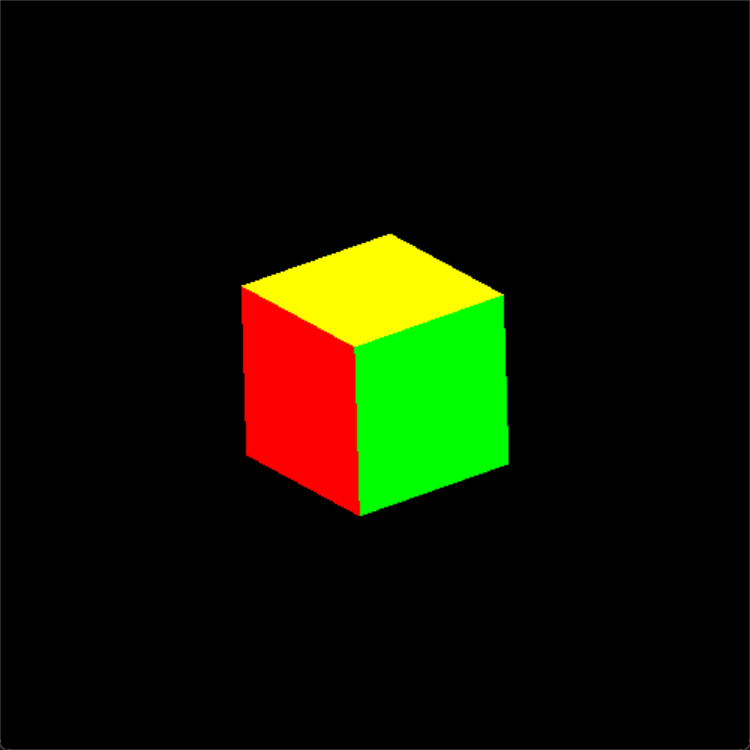


绘制6个面



循环角度：



运行结果截图： 

### c.交互输入

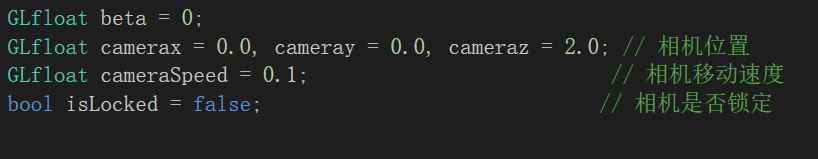
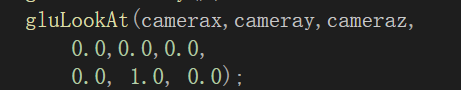
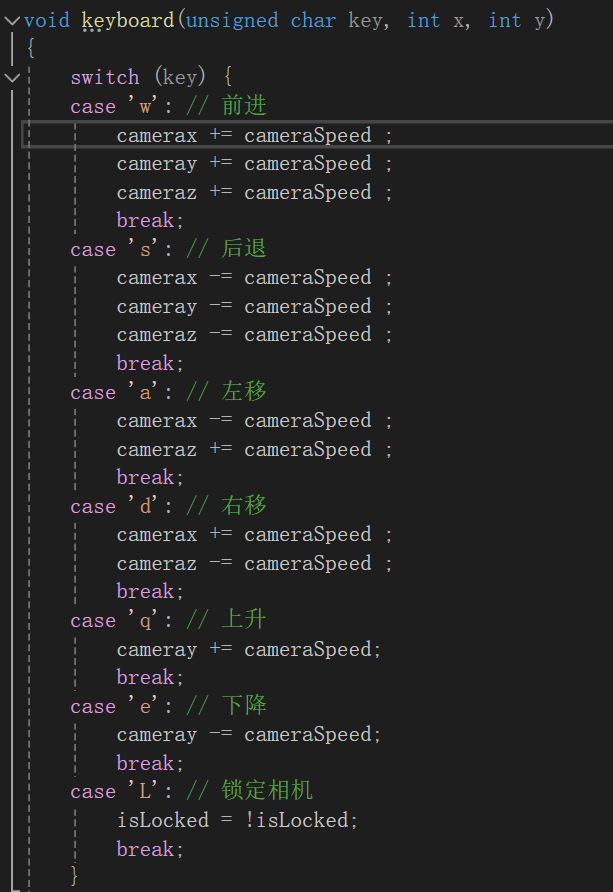
部分完成

操作流程：

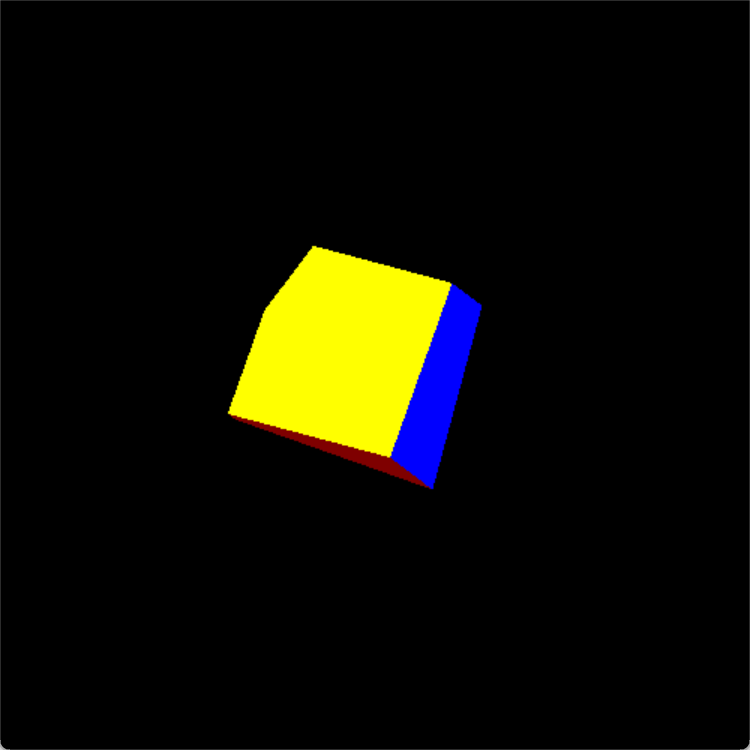
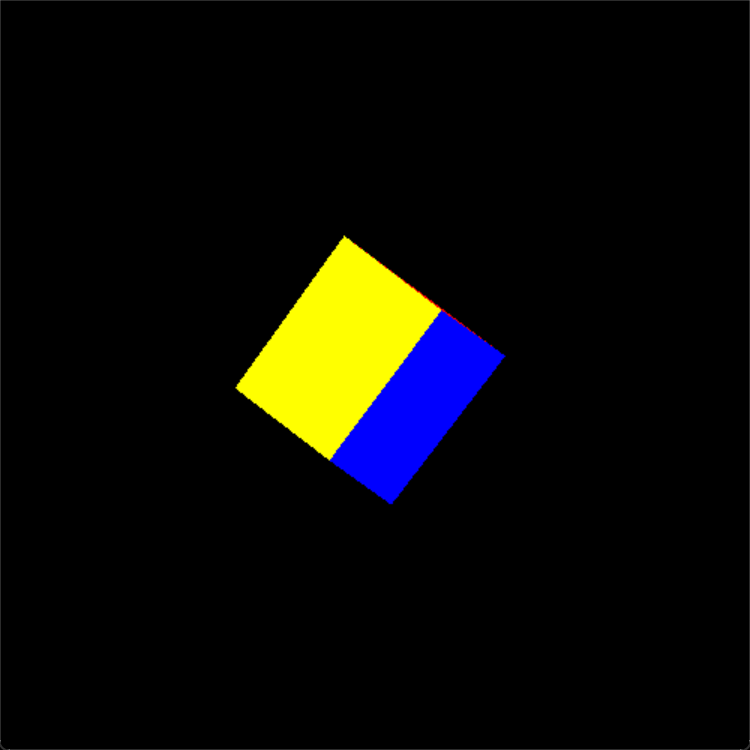
1.先定义相机的位置变量，移动速度，锁定状态；并写入gluLookAt方法中。

2.在keyboard方法中，针对不同的键盘输入分别在x,y,z轴上进行移动。

关键代码截图：

运行结果截图：

# Task3

1.

完成

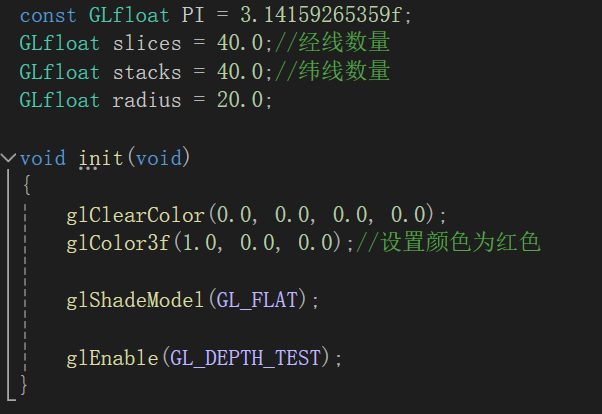
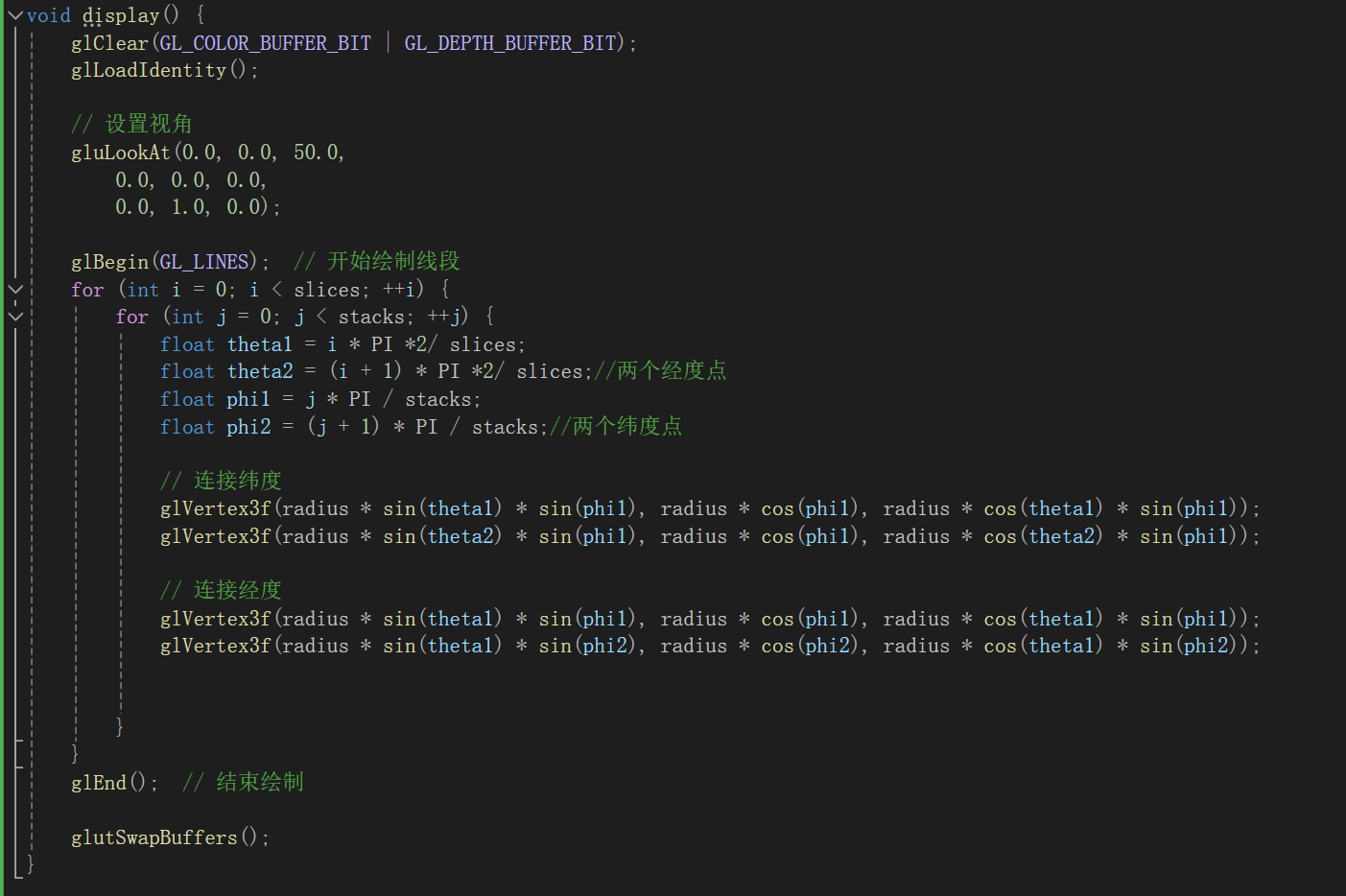
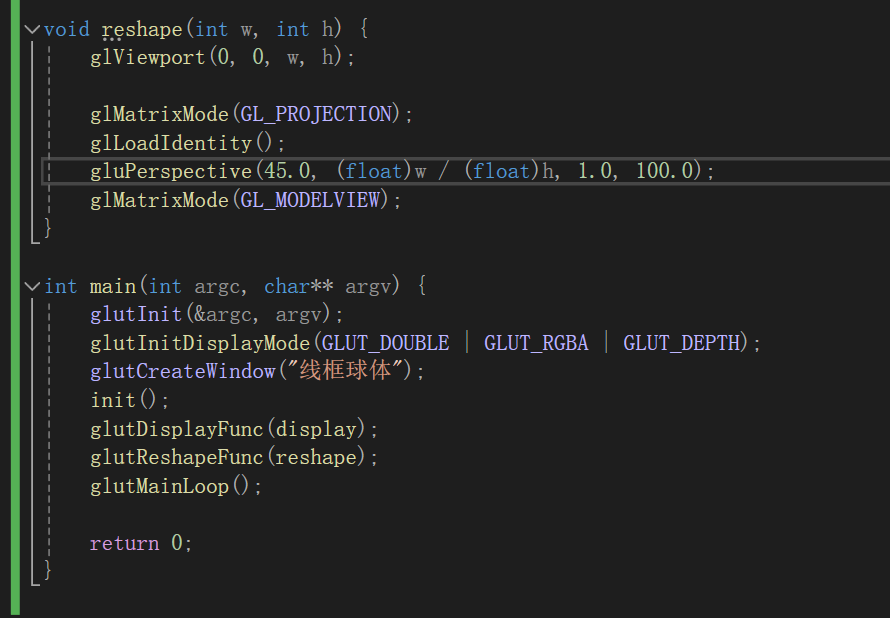
操作步骤：

1.定义经纬线数量和半径

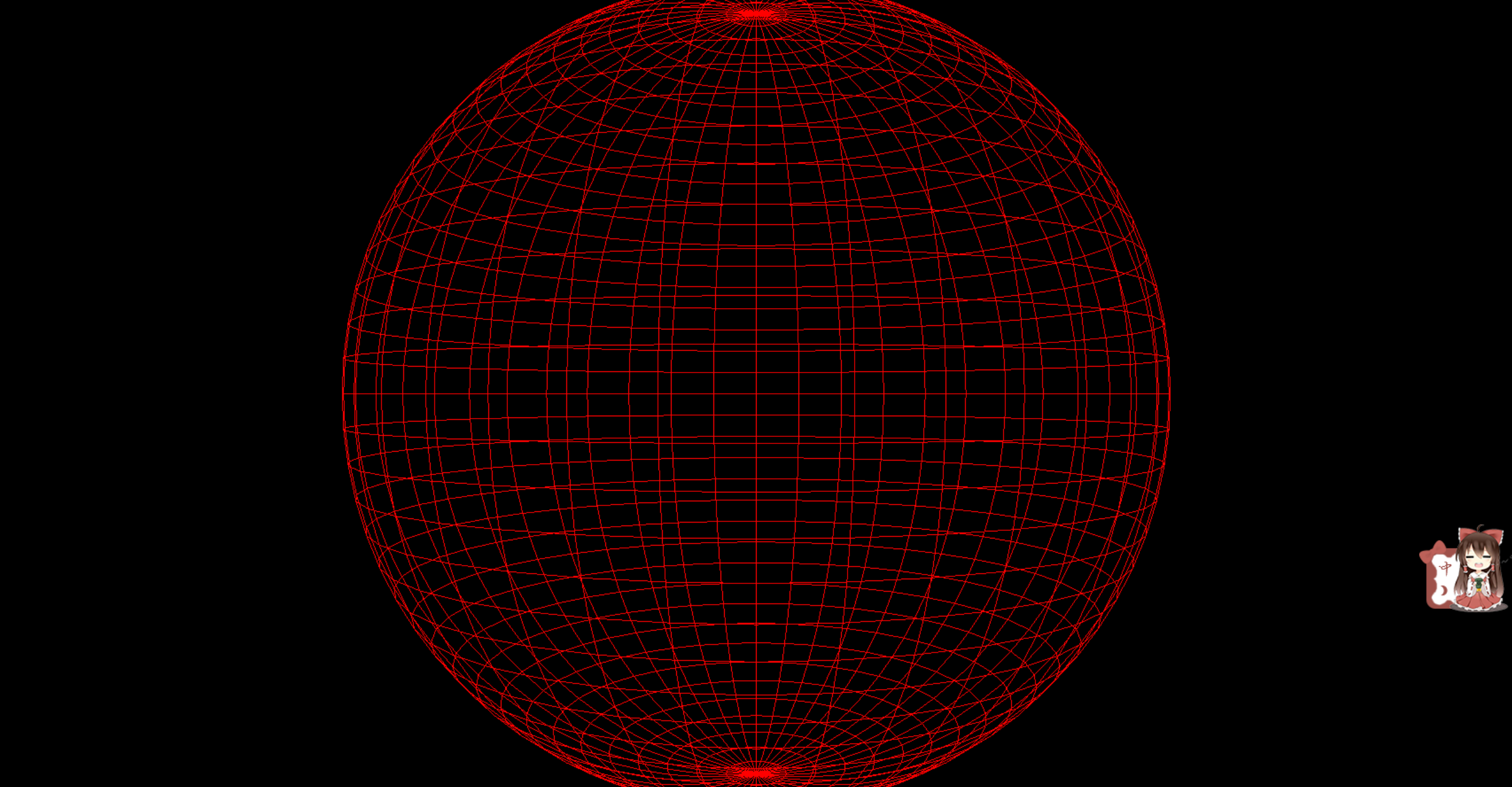
2.开启深度测试，并设置为颜色。

3.算出相应的经纬度并绘制相关点，连接线段。

关键代码：

运行结果：



2.

完成

操作步骤：

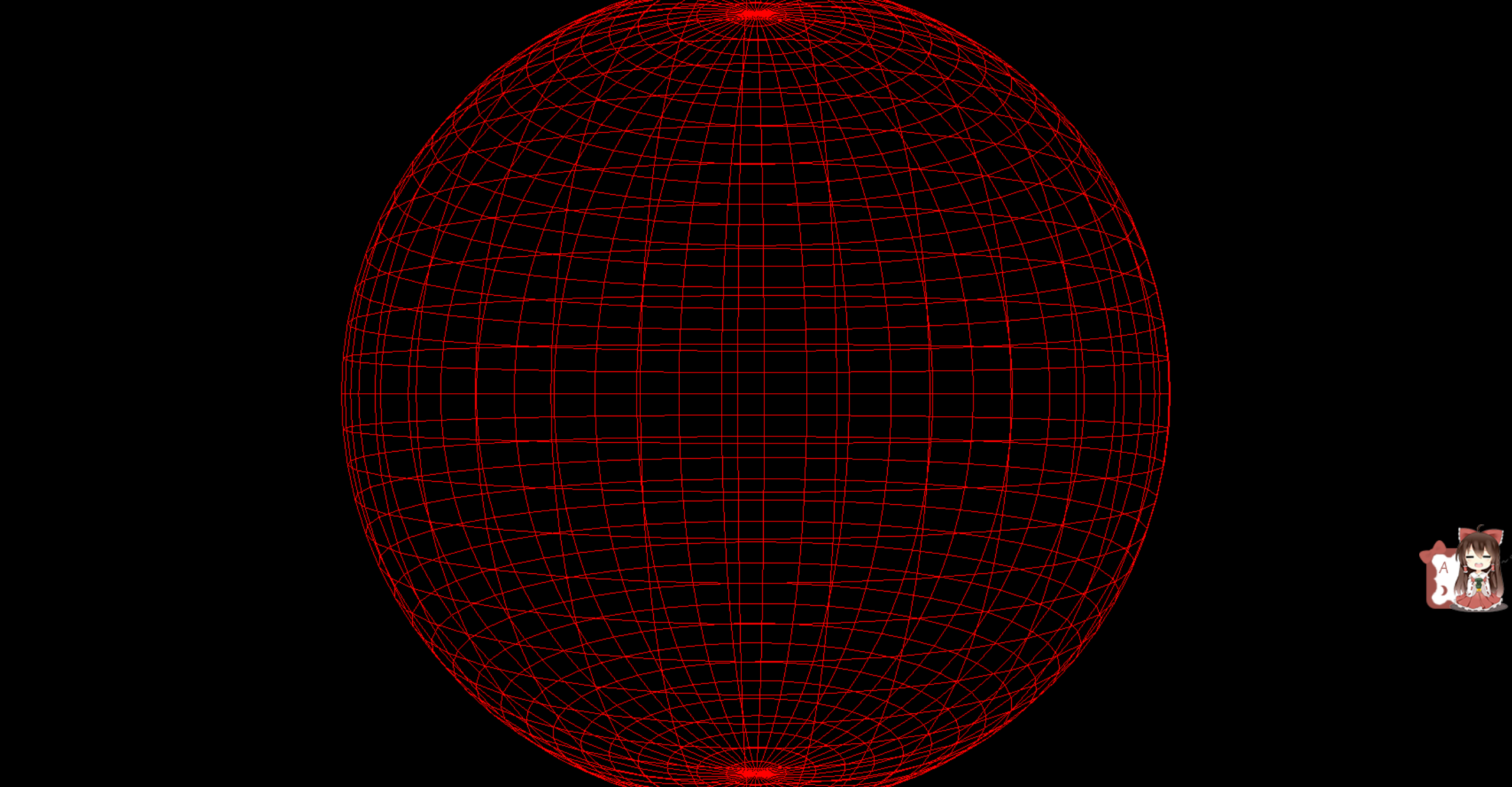
1.定义旋转角。利用glRotatef确定旋转轴。

2.更新旋转角的量

关键代码截图：

运行结果截图:



3.

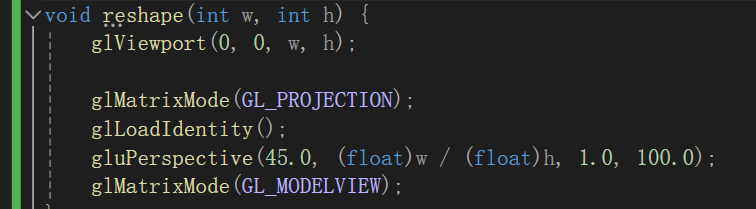
完成

操作步骤：

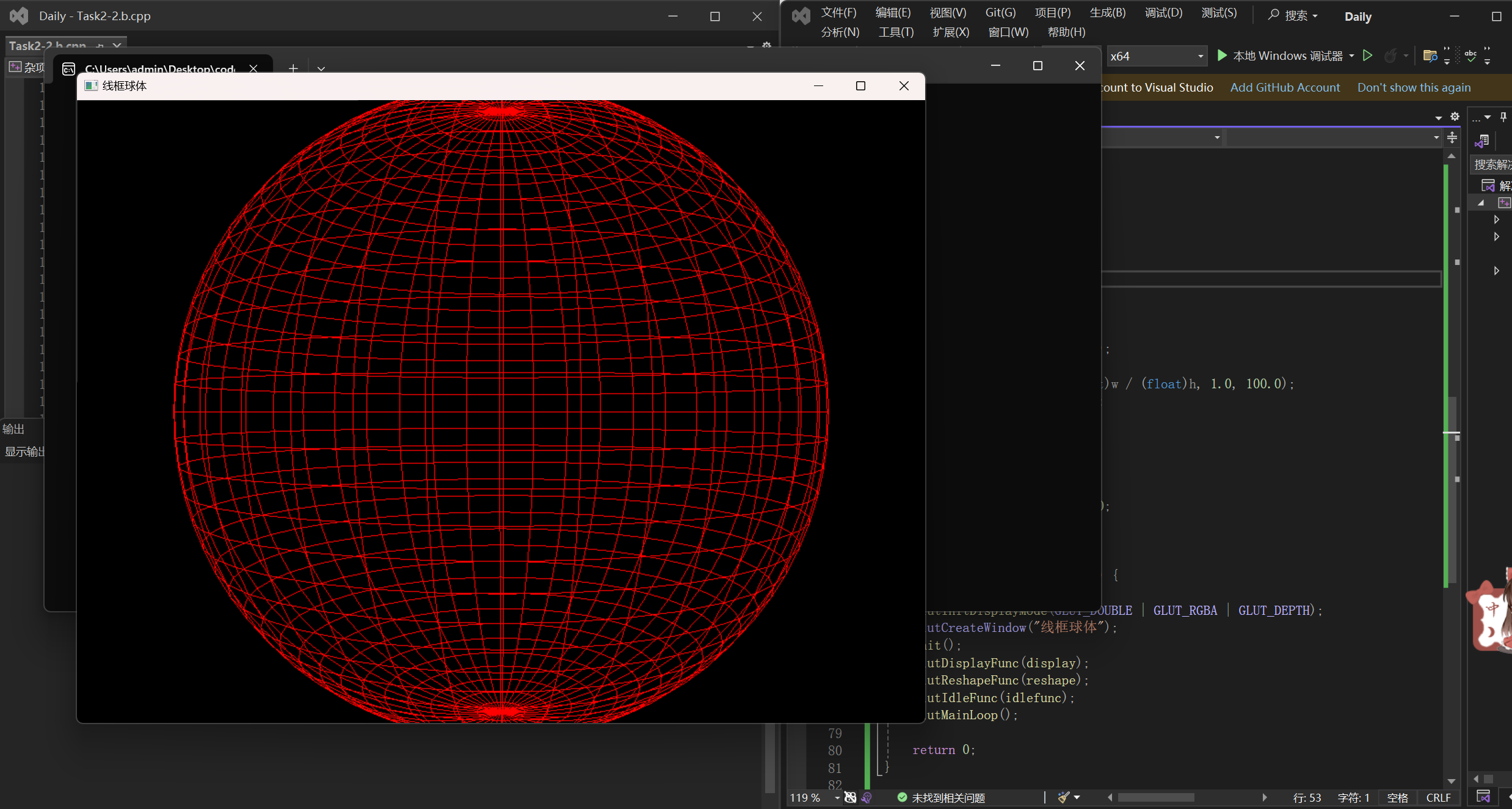
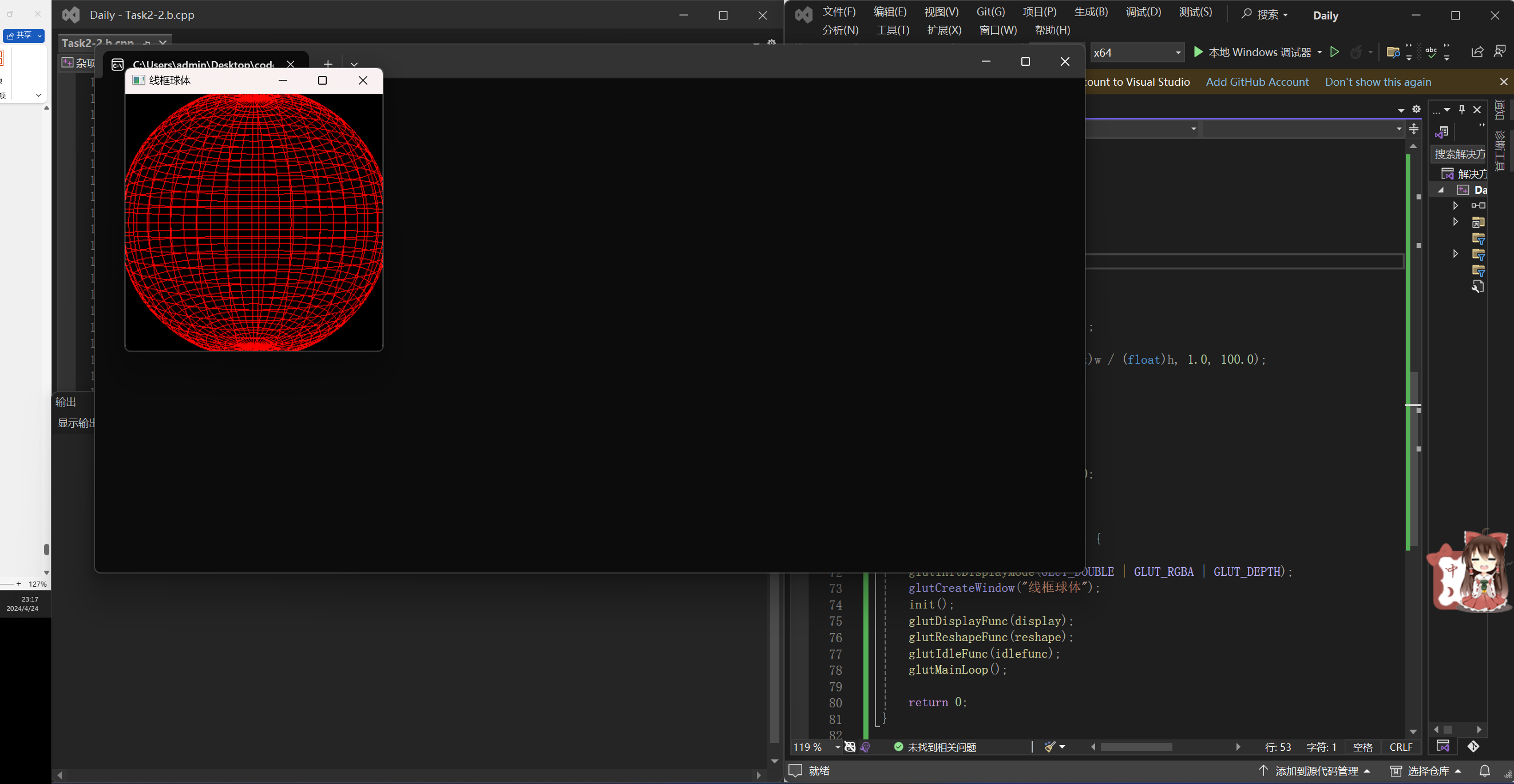
1.更新视口的大小以匹配新的窗口尺寸。

2.计算窗口的宽高比，并使用 gluPerspective 函数创建一个新的透视投影矩阵，以确保球体在新窗口大小下保持比例不变。

关键代码：



运行结果:



3.

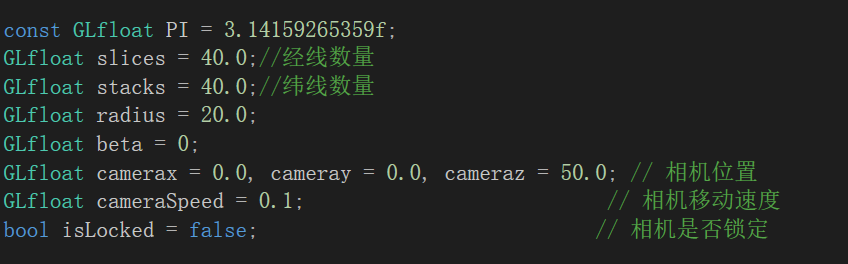
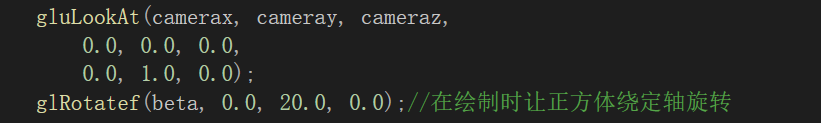
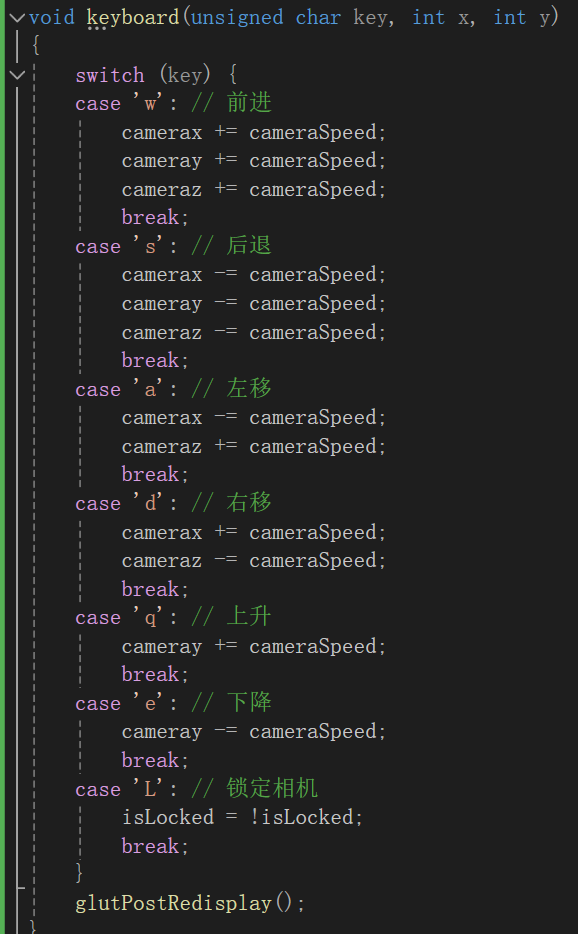
部分完成

操作流程：

1.先定义相机的位置变量，移动速度，锁定状态；并写入gluLookAt方法中。

2.在keyboard方法中，针对不同的键盘输入分别在x,y,z轴上进行移动。

关键代码截图：

运行结果截图：

