

计算机图形学实验

实验2 OpenGL绘制基础

姓 名：马鑫\_\_\_\_\_\_\_\_\_\_\_

学 号：\_37220232203780\_

学 院：\_信息学院\_\_\_\_\_\_

专 业：\_数字媒体技术\_\_

年 级：\_2023级\_\_\_\_\_\_\_

2025年 4 月 2日

目录

[Task1：绘制Sierpinski镂垫程序 3](#_Toc23980)

[1.完成。理解并实现课本程序 3](#_Toc10647)

[2.完成？尝试为不同三角形设置不同的颜色，使得看起来颜色更加美观。 4](#_Toc6883)

[3.完成。为这个镂垫生成动画： 7](#_Toc28738)

[Task2: 三维物体绘制 12](#_Toc15120)

[1.完成。设置合适的相机位置和相机投影矩阵，使得OpenGL相机能够“看到”需要绘制的物体。 12](#_Toc26072)

[2.完成。运行提供的示例程序(exp2-2-1.cpp)，对程序进行改写 14](#_Toc16411)

[Task3: 绘制一个线框球体 24](#_Toc27886)

[1.完成。在不使用glut对象的情况下，绘制一个线框球体（即仅仅使用基本图元进行绘制，不能使用glutWireSphere 绘制对象函数） 24](#_Toc4434)

[2.完成。添加动画效果，让球体绕圆心旋转 26](#_Toc19069)

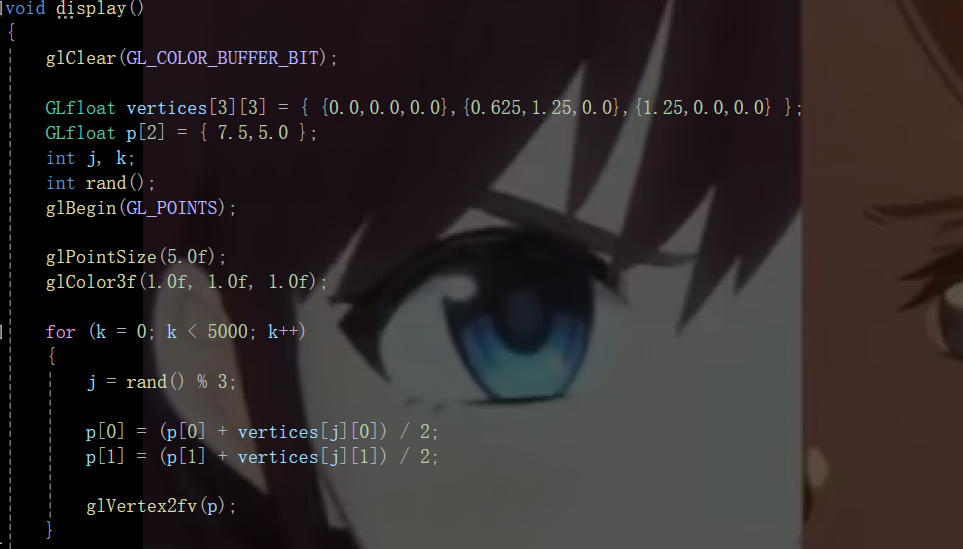
[3.完成。添加reshape回调函数，使其保持比例，不会随着窗口大小的改变而产生形变 28](#_Toc30498)

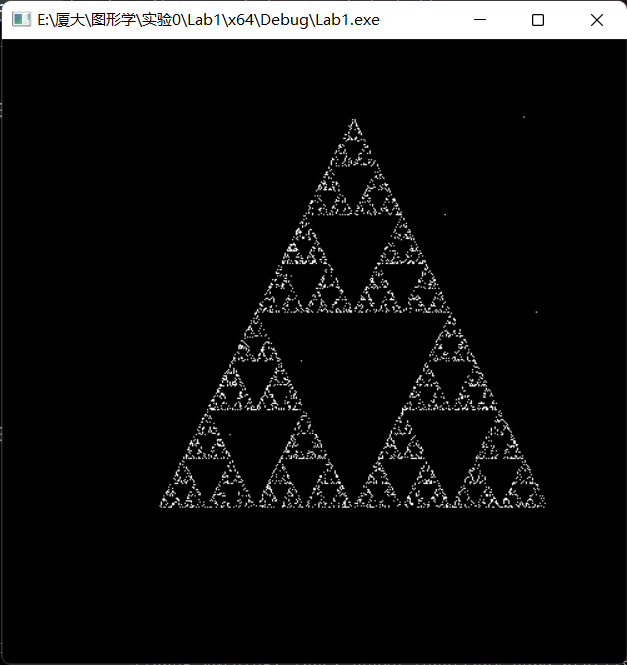
[4.完成。添加交互式的相机控制（同task2） 29](#_Toc18229)

# Task1：绘制Sierpinski镂垫程序

## 1.完成。理解并实现课本程序

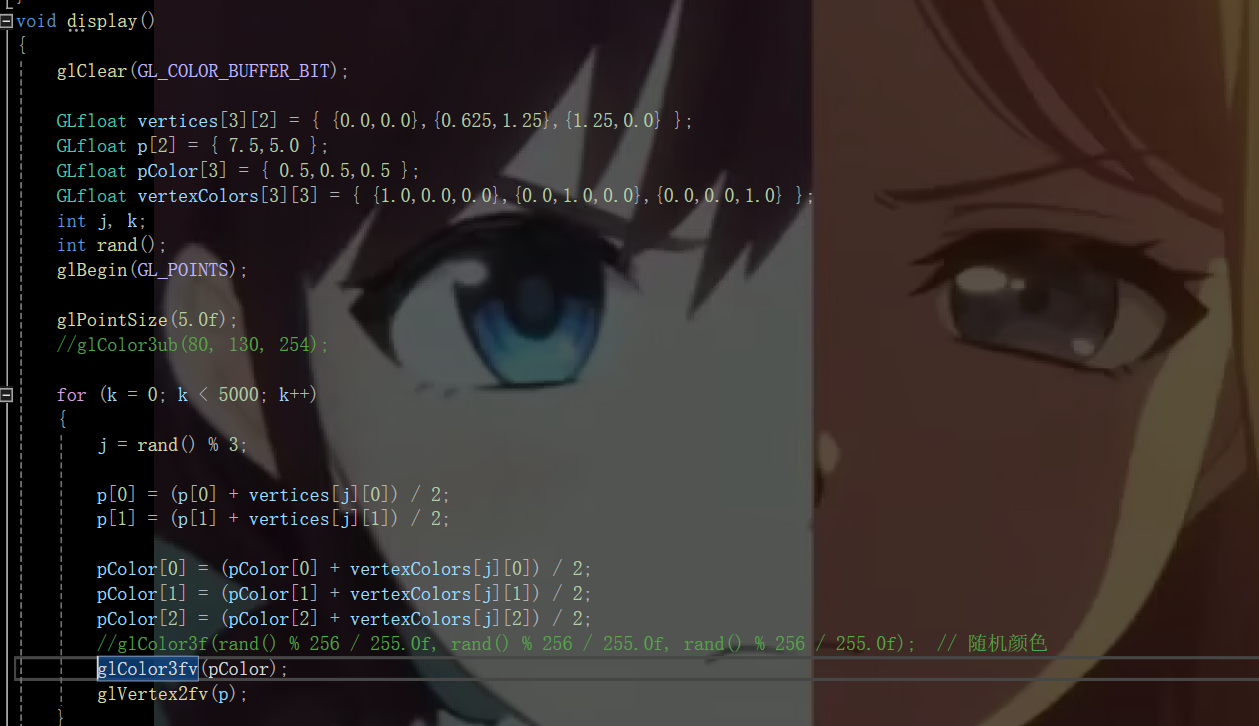
操作流程：实现课本代码，根据实际情况进行修改。由于使用的是正交矩阵，减少三角形的缩放比例，使之能够出现在摄像机上。

关键代码截图：

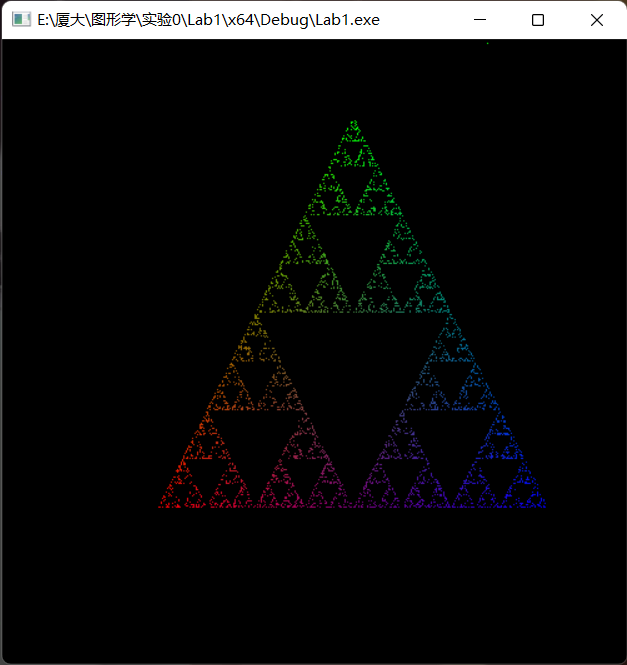
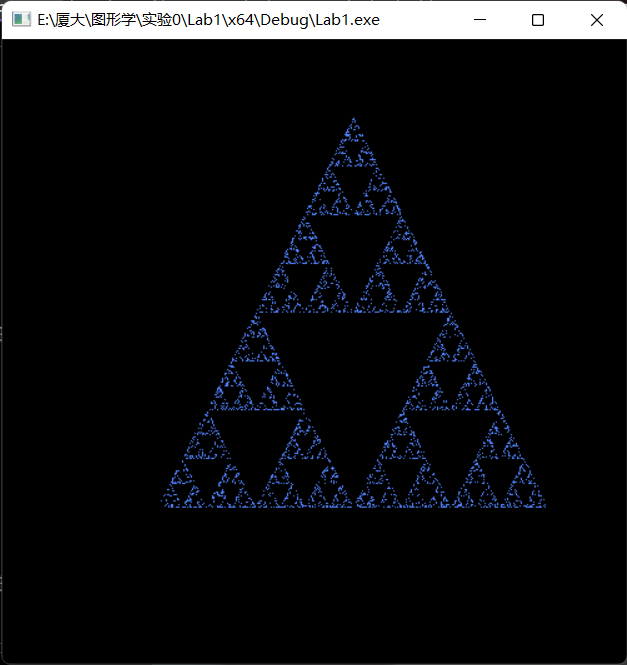
运行结果截图： 

## 2.完成？尝试为不同三角形设置不同的颜色，使得看起来颜色更加美观。

操作流程：这道题的题目真是真是模棱两可，题干要求不同的颜色，示例又是纯色，纯色只用修改参数即可。而“不同三角形”的定义又不清晰，本题的代码又是通过随机点生成的，不能确定是哪三个点组成了三角形，为了美观需求，采取与生成Sierpinski镂垫程序思想类似的渐变色生成。

关键代码截图：

运行结果截图：



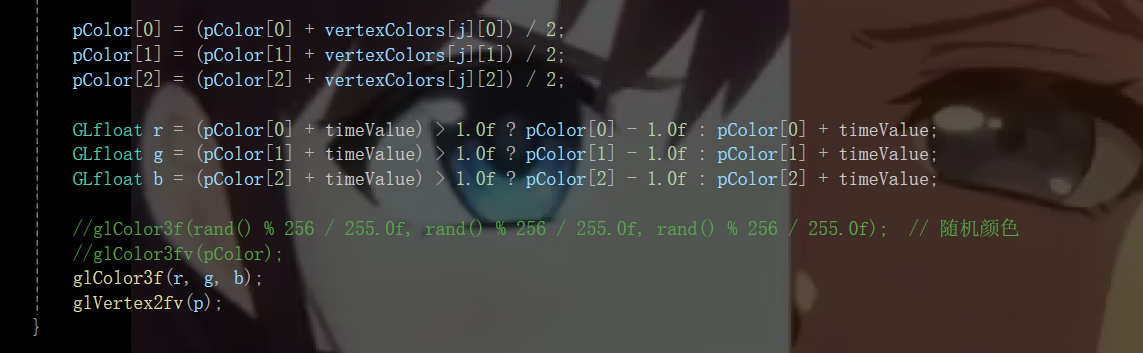
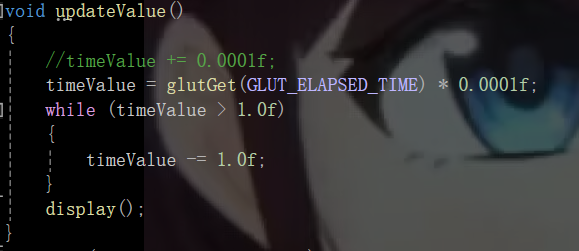
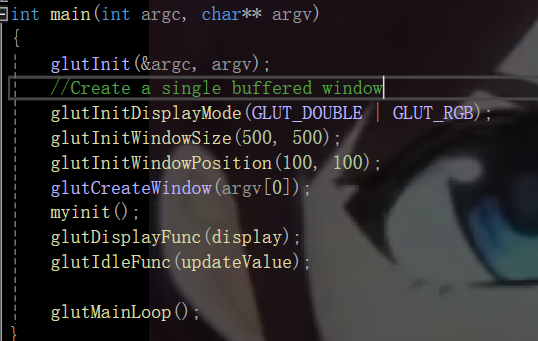
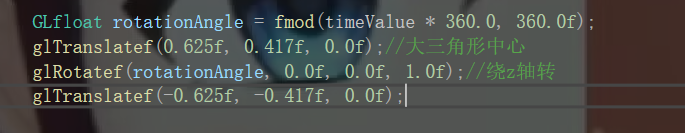
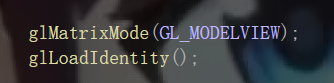
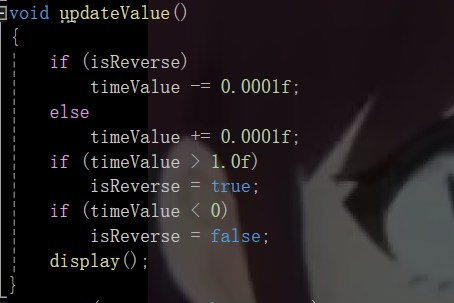
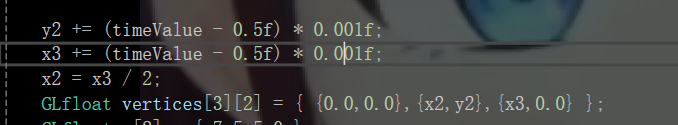
## 3.完成。为这个镂垫生成动画：

完成。①镂垫的颜色随时间不断变化。

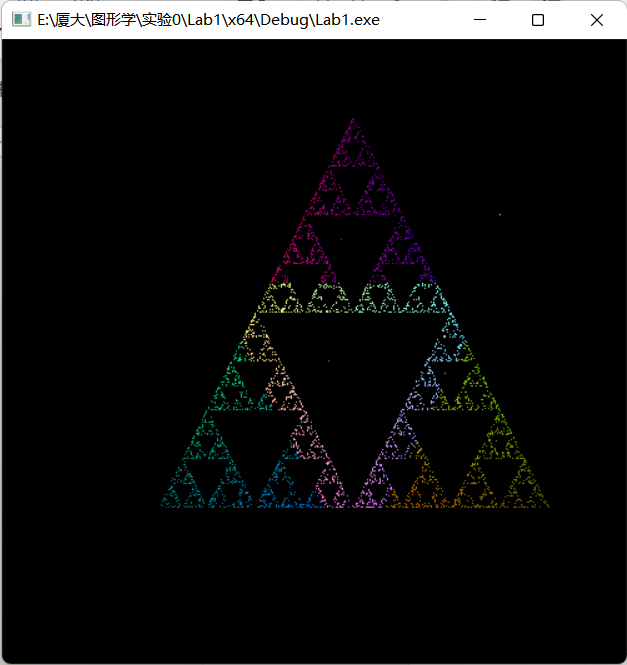
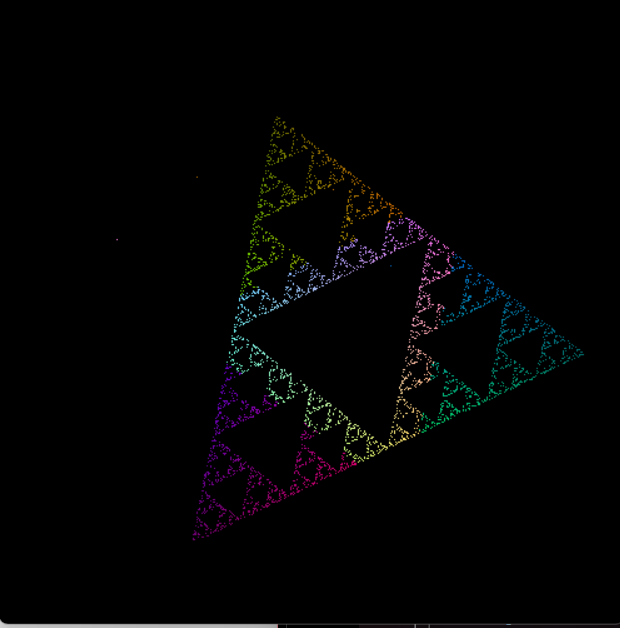
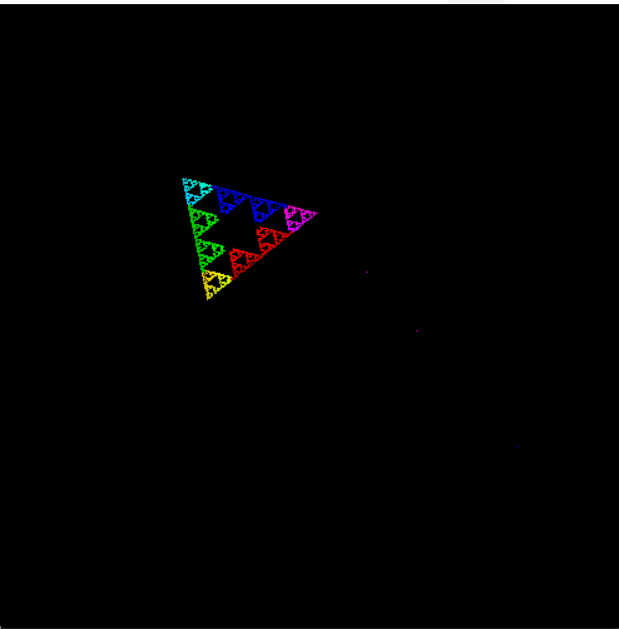
完成。②在①的基础上增加旋转效果。

完成。③在②的基础上增加缩放效果。

操作流程：①增加timeValue变量， 颜色值的变化与timeValue相关，改变现实模式为双缓冲，增加glutIdleFunc改变timeValue值。②将参考点移至三角形重心，调用glrotate绕z轴旋转，将矩阵模式改为模型视图矩阵。③通过改变三角形顶点的值，实现缩放，缩放的中心位于其中一顶点，为了使缩放更加顺滑，改变timeValue的变化逻辑，使之能够在0~1之间改变。

关键代码截图：①②③

运行结果截图： ①

②③

# Task2: 三维物体绘制

## 1.完成。设置合适的相机位置和相机投影矩阵，使得OpenGL相机能够“看到”需要绘制的物体。

操作流程：运行程序 projection/projection.exe，运行情况如图表2，调整glOrtho、 gluLookAt等函数的参数，了解各参数意义。

思考题

1. OpenGL中，三维空间的坐标系是怎么样的？

答：OpenGL中的坐标系是右手坐标系，一般用x轴表示左右，y轴表示上下，z轴表示前后。

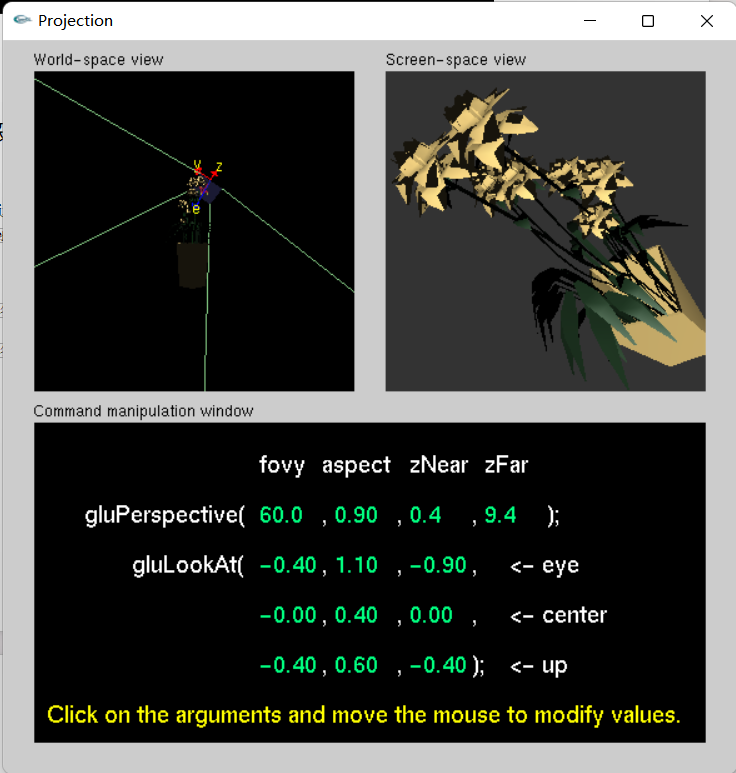
b. OpenGL中，相机的方位是怎么样的？如何调整相机朝向呢？

答：OpenGL中，摄像机方位由三个向量共同表示，即表示摄像机位置的向量（eye），看向目标的坐标向量（center），指向摄像机上方的上向量（up），通过调用gluLookAt函数调节其中的9个参数即可调整相机朝向。

c. OpenGL中，相机可见范围是如何设置的？

答：OpenGL中当使用透视摄像机时，可通过调用gluPerspective函数，调节其中fovy,aspect,zNear,zFar四个参数来设置相机可见范围。

关键代码截图：略

运行结果截图： 

## 2.完成。运行提供的示例程序(exp2-2-1.cpp)，对程序进行改写

* 1. 比较开启/不开启深度缓冲区 glEnable(GL\_DEPTH\_TEST) 的效果;理解深度缓冲区的作用、用法
  2. 让正方体自行旋转，而不是相机旋转
  3. 在正方体自行旋转的前提下，实现交互式的相机控制（wasd控制相机的前进后退左右移动，qe实现相机的升降，使用鼠标调整相机的朝向，L锁定相机的移动和旋转）

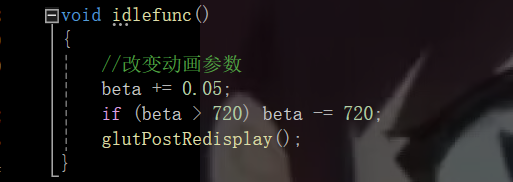
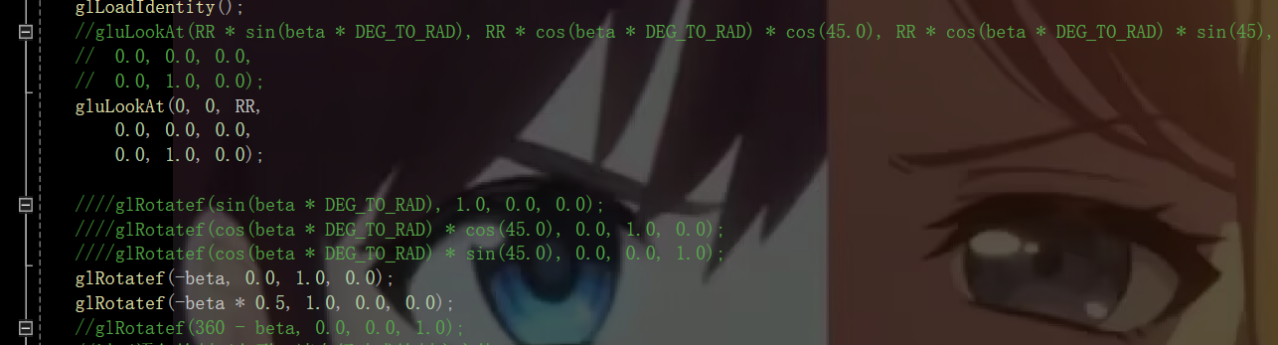
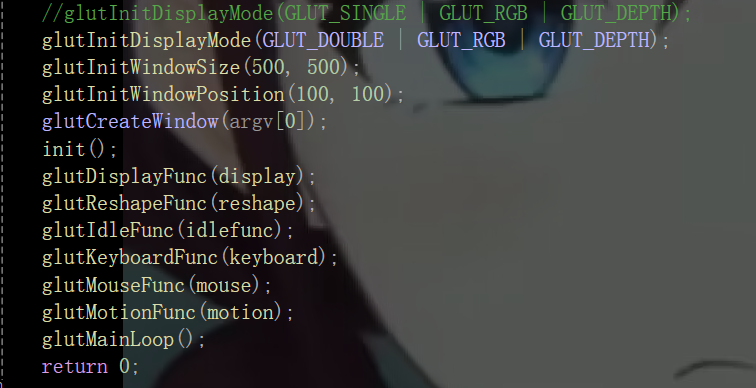
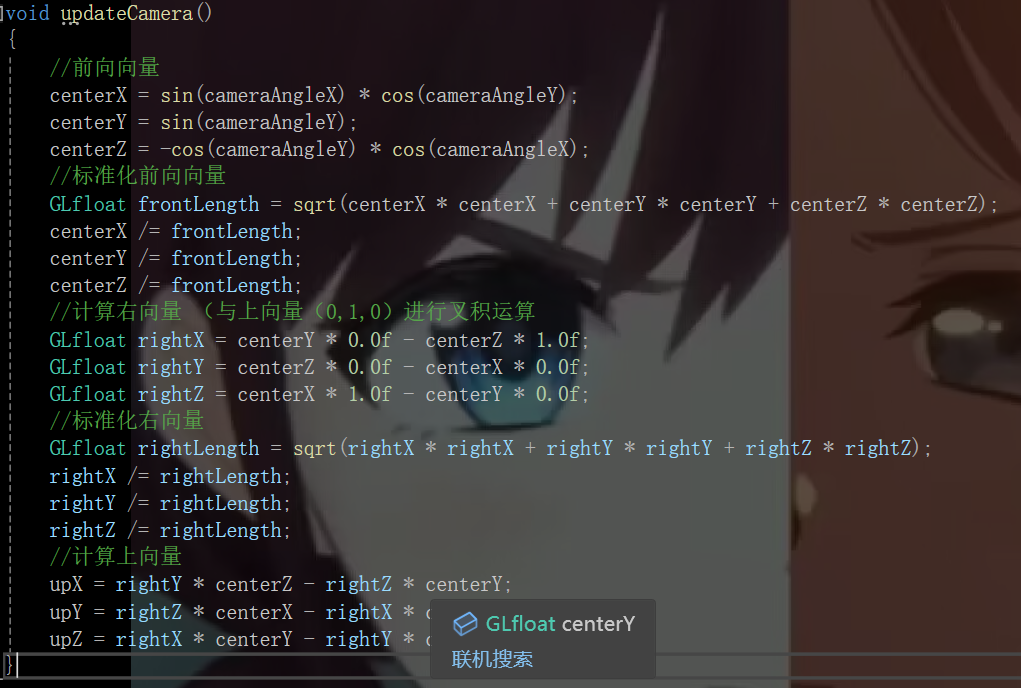
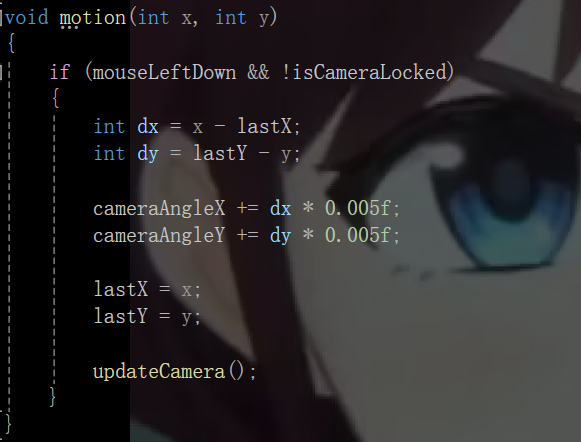
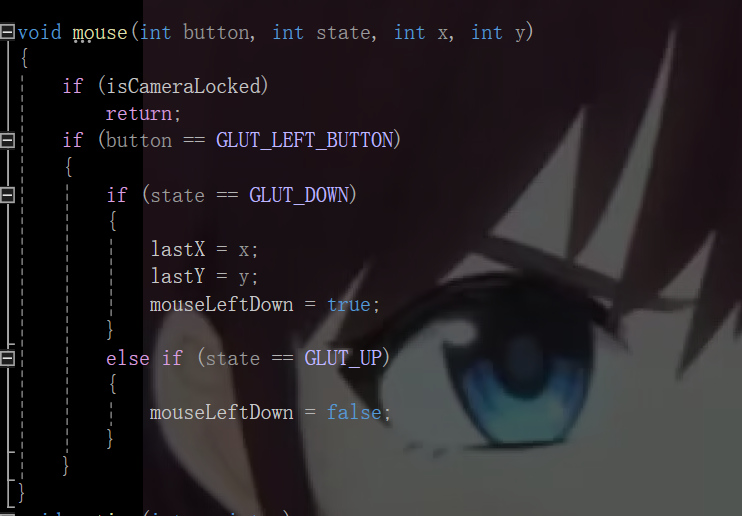
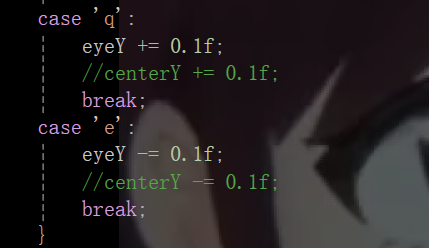
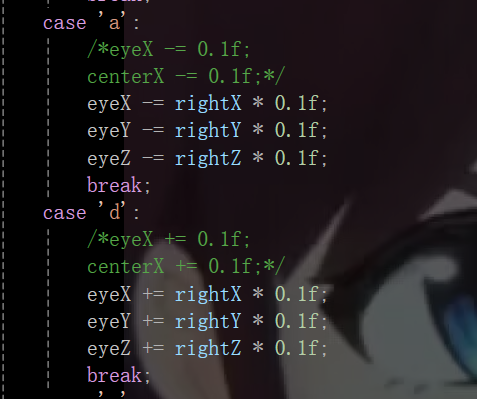
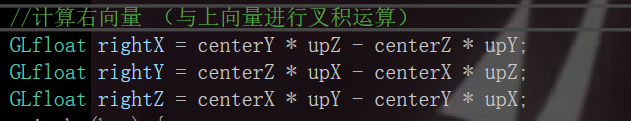
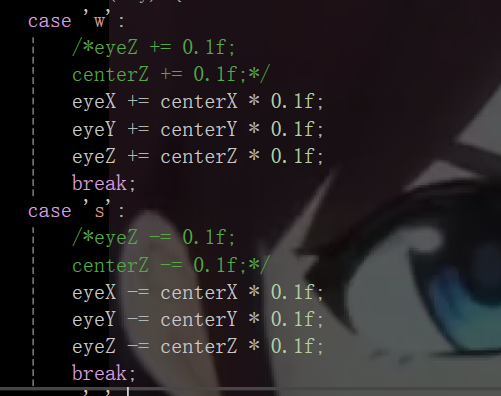
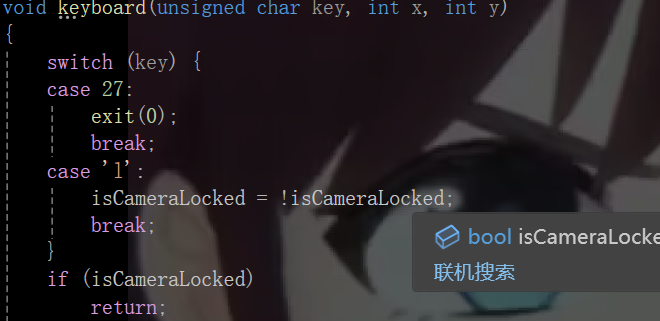
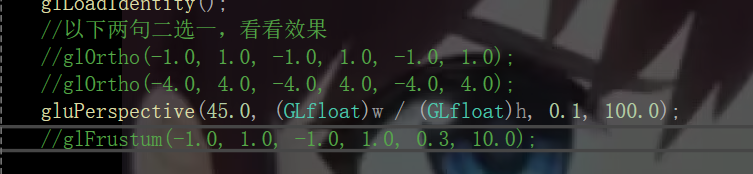
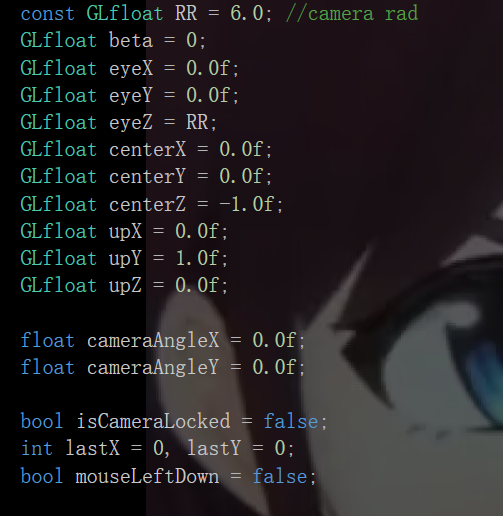
操作流程：运行示例程序。

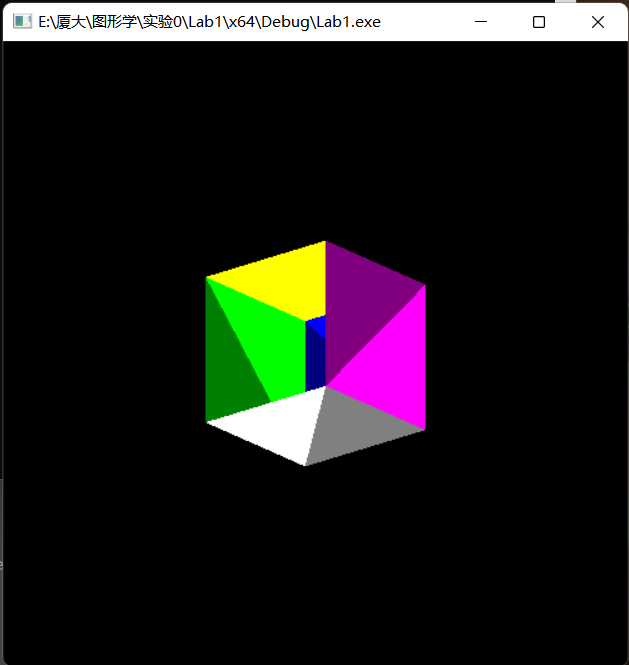
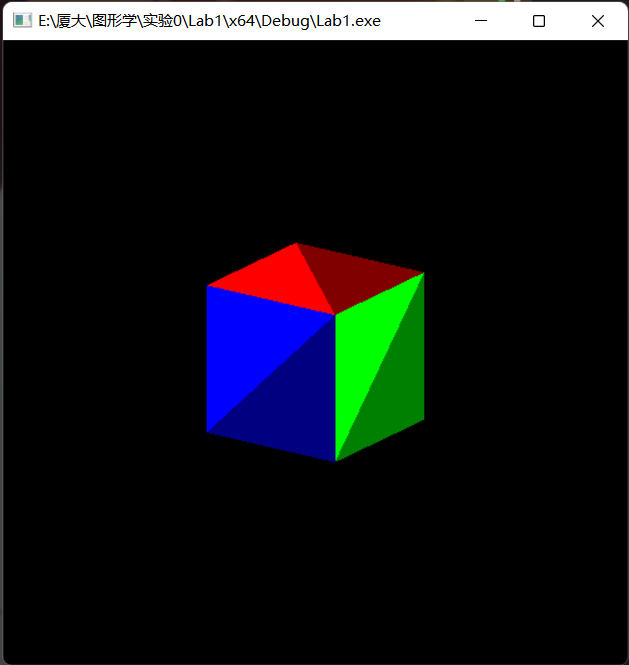
1. 完成。关闭深度缓冲，再次运行。深度缓冲区可以记录摄像机与观察物体之间的距离，明确相互之间的遮挡关系，使被遮挡的像素不再绘制。
2. 完成。不改变gluLookAt的参数，使用glRotatef使立方体绕特定轴旋转，为了尽可能贴近原来的视图效果，使之先绕y轴旋转，再绕x轴旋转一半，因此需要改变beta的范围为0-720。
3. 完成。将gluLookAt的9个参数定义出来，以便进行修改，其中center参数为eye的相对参数。为了便于观察，将正交相机改为透视相机。修改keyboard回调函数，增加对wsadqe和l的判断，先判断l，用于修改isCameraLocked变量，对于w和s，对摄像机的位置按朝向方向进行增减，对于a和d，对摄像机位置按右向量方向增减，对于q和e，调整摄像机的y轴。定义mouse回调函数，判断按下和松开左键时修改参数，定义motion回调函数，当按住鼠标左键时，获取当前鼠标位置和上一帧鼠标的位置关系，对摄像机参数进行修改。

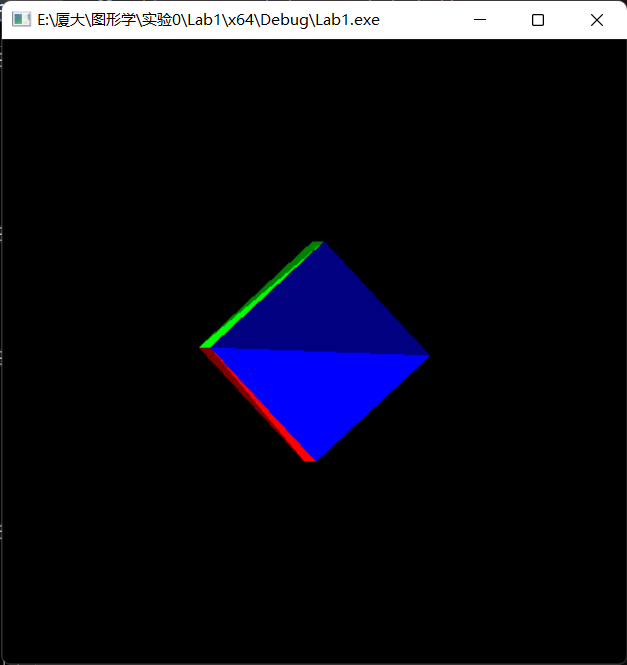
思考题：如何实现前后面的遮挡？你需要先自行了解一下深度缓冲区和深度测试的作用。搜索：GL\_DEPTH\_TEST关键字。

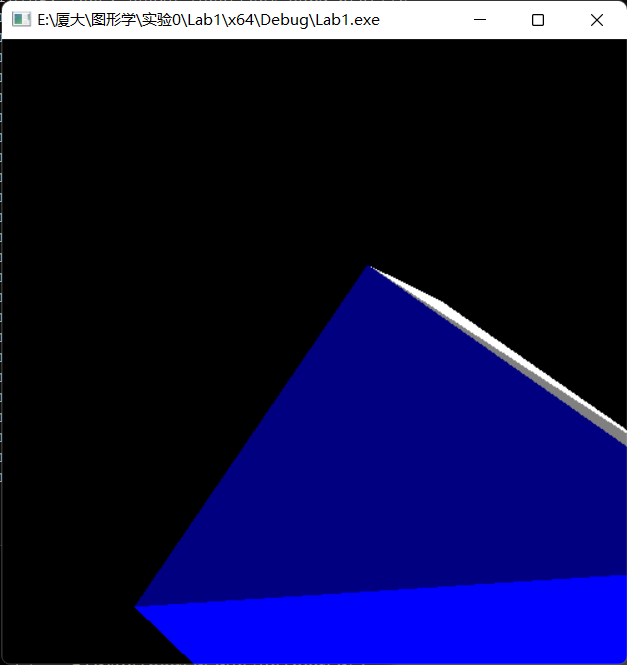
答：在初始化显示模式glutInitDisplayMode中传入GLUT\_DEPTH参数，使用glEnable函数传入参数GL\_DEPTH\_TEST开启深度测试，在显示前调用glClear函数传入GL\_DEPTH\_BUFFER\_BIT清空深度缓冲区。开启深度测试，会在片段着色器运行之后多包含一个z坐标，表示实际深度值，在绘制时会丢弃同一位置上深度值高于当前深度缓冲区的值的片段，从而实现遮挡。

关键代码截图：a. 

1. 
2. 

运行结果截图： a. 

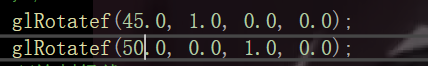
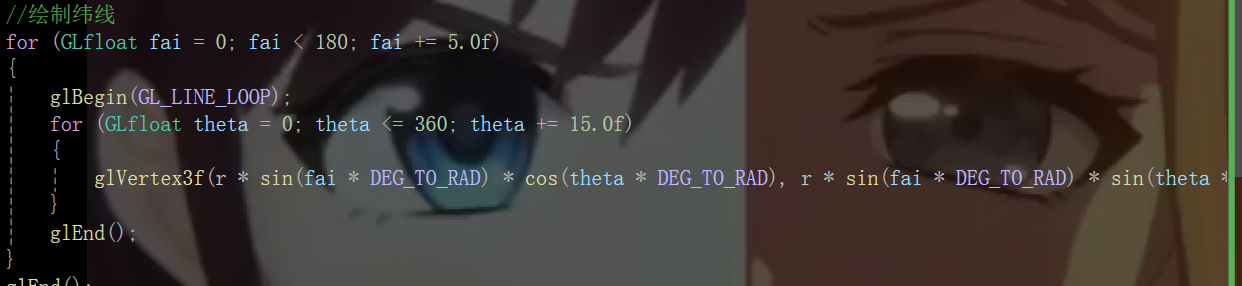
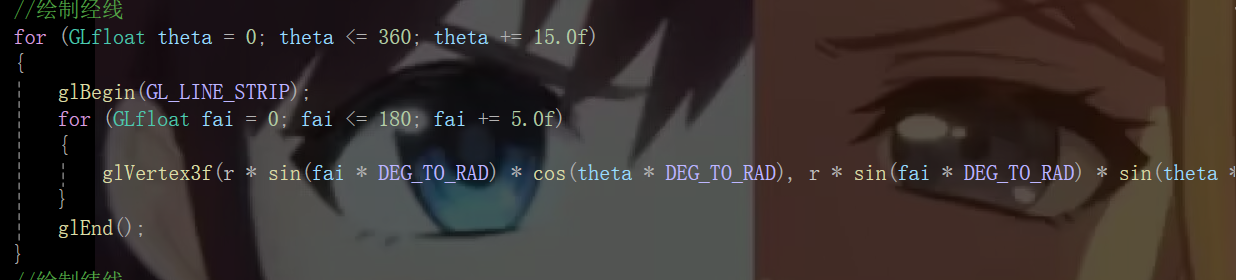
b. 

c. 

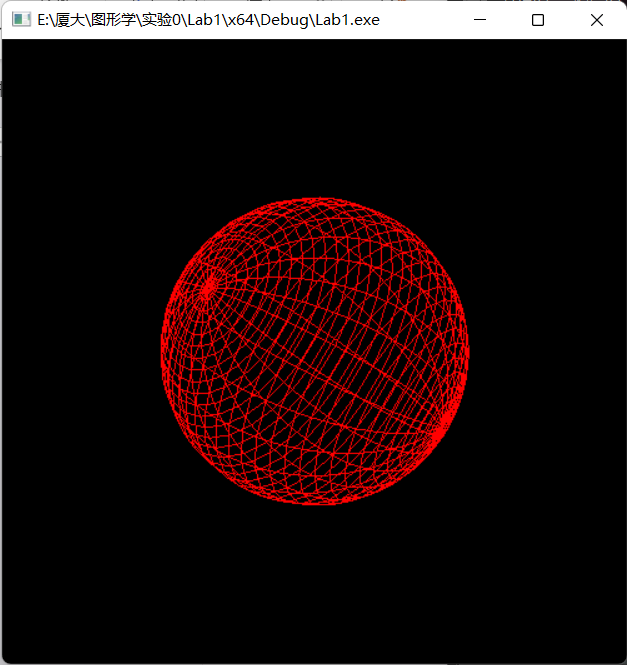
# Task3: 绘制一个线框球体

## 1.完成。在不使用glut对象的情况下，绘制一个线框球体（即仅仅使用基本图元进行绘制，不能使用glutWireSphere 绘制对象函数）

操作流程：使用极坐标公式绘制球体，先用GL\_LINE\_STRIP绘制经线，先遍历theta后遍历phi，再用GL\_LINE\_LOOP绘制纬线，先遍历phi再遍历theta，通过对旋转操作使之看起来更具具体感，更接近示例。

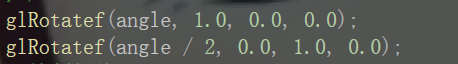
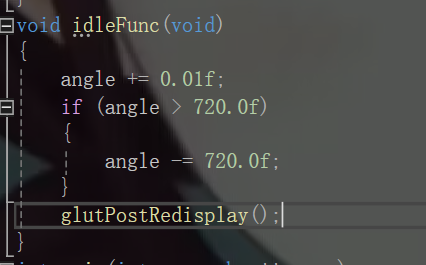
关键代码截图：

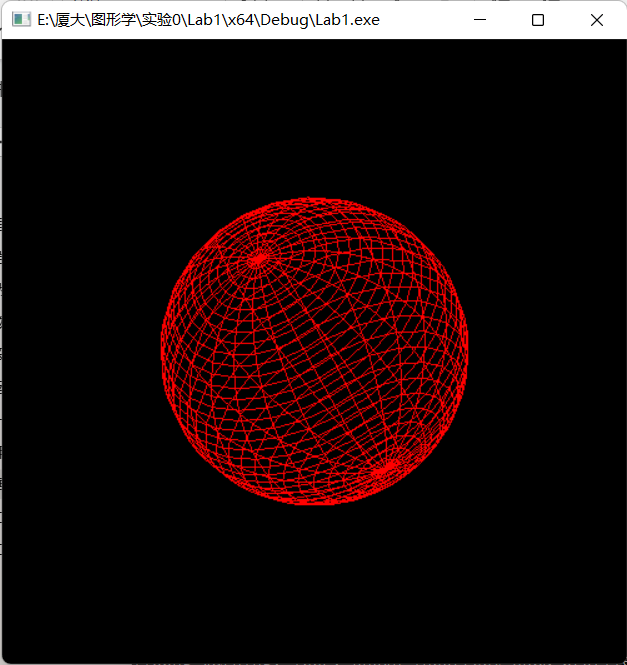
运行结果截图：



## 2.完成。添加动画效果，让球体绕圆心旋转

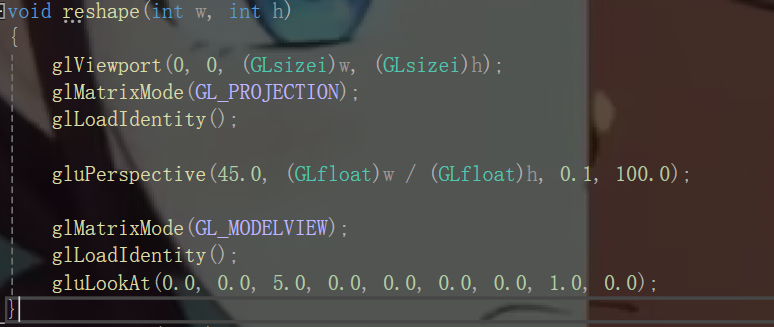
操作流程：增加angle参数和idleFunc回调函数，每帧修改angle的值并传入Rotatef中。

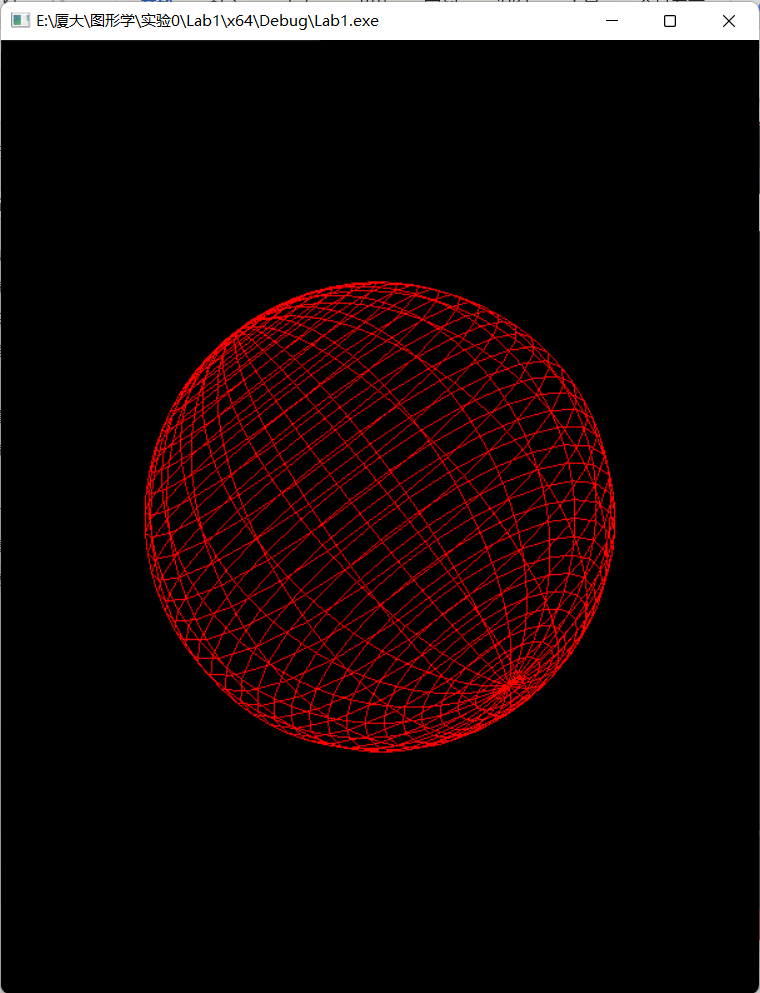
关键代码截图：

运行结果截图： 

## 3.完成。添加reshape回调函数，使其保持比例，不会随着窗口大小的改变而产生形变

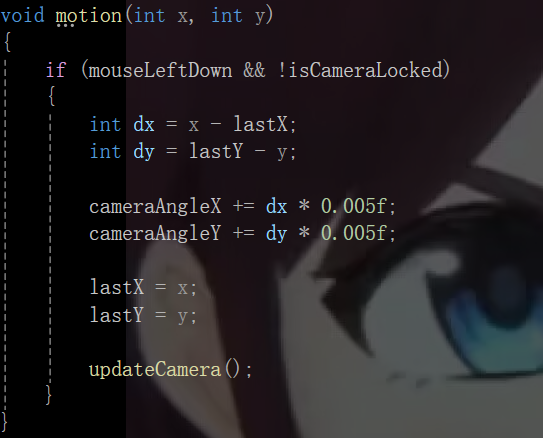
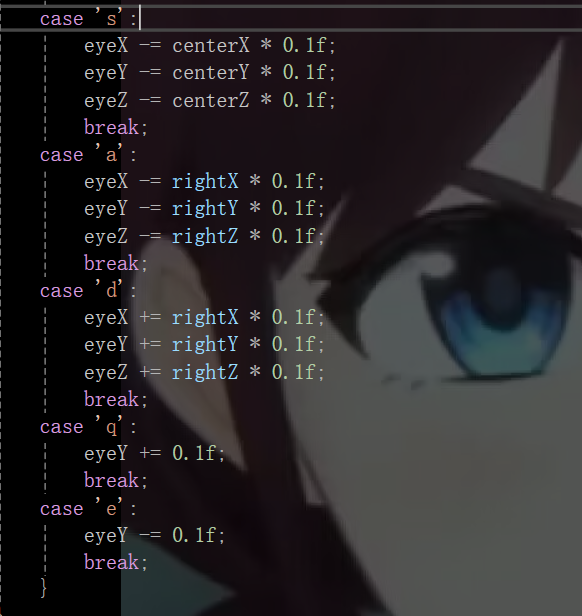
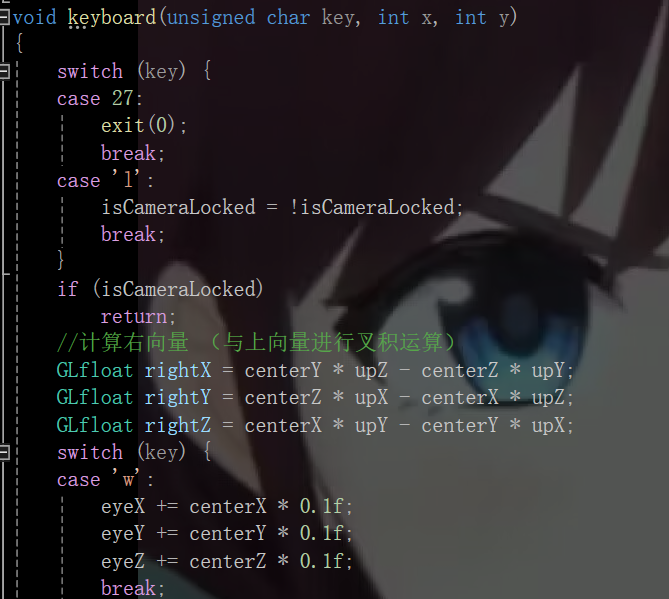
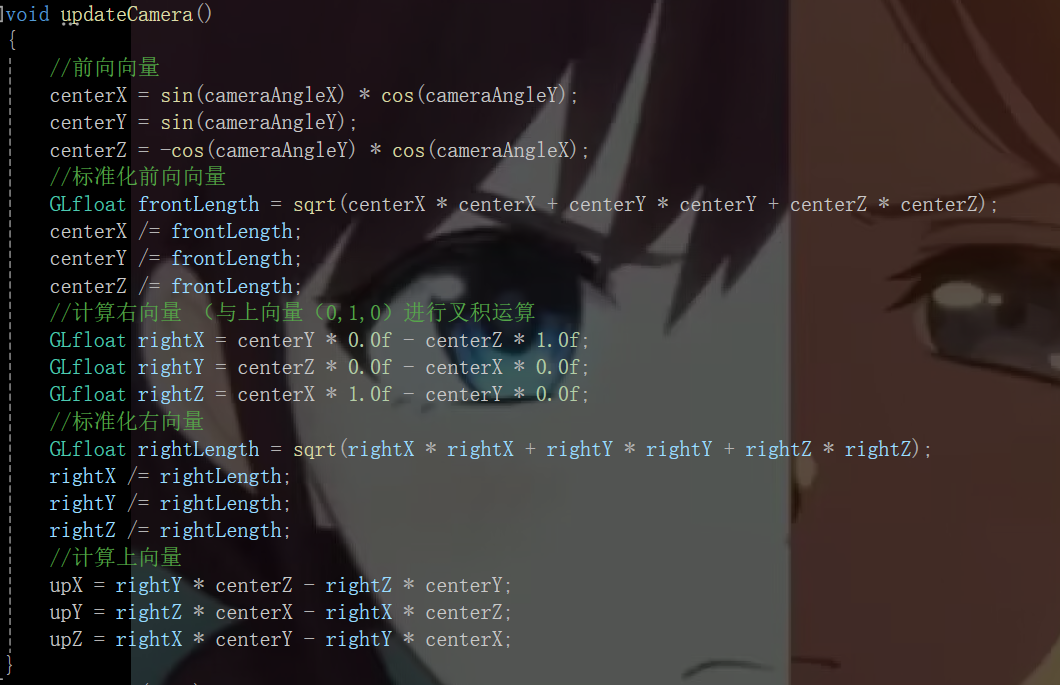
操作流程：添加reshape回调函数，使用gluPerspective透视摄像机。

关键代码截图：

运行结果截图： 

## 4.完成。添加交互式的相机控制（同task2）

操作流程：拷贝task2的代码，代码逻辑同task2。

关键代码截图：

运行结果截图： 