### 

****

**计算机图形作业（一）报告**



**学 院 软件学院**

**专 业 软件工程**

**学 号 3017218135**

**姓 名 全康连**

目录

**摘要2**

项目名称2

操作简介2

**主体介绍2**

程序简介2

算法2

实现细节9

**结果与讨论**1**4**

结果14

讨论16

收获和建议**17**

课程收获17

项目开发收获17

课程建议18

一．摘要

1. 项目名称：Assignment1---Illumination Effect
2. 操作简介
3. 从obj文件中加载3D模型，通过鼠标旋转并缩放模型进行查看
4. 为模型加上材料属性并使用Phong光照模型产生不同的光照效果

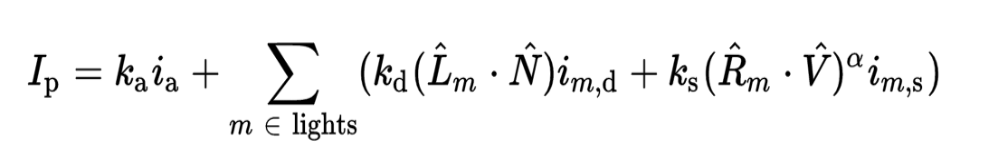
二．主体介绍

1. 程序简介

该程序完成了从obj文件中加载3D模型，并给模型加上材料属性，使用Phong光照模型产生不同的光照效果的功能；程序中可以通过鼠标，键盘在3D场景中移动来观察模型

1. 算法
2. Phong光照模型

程序使用Phong光照模型的数学原理使得物体产生不同的光照效果。在Phong模型中，有公式如下：



其中物体呈现的颜色为

Reflected=Ambient+Diffuse+Specular;

分别对应于Phong模型公式中的相应位置。

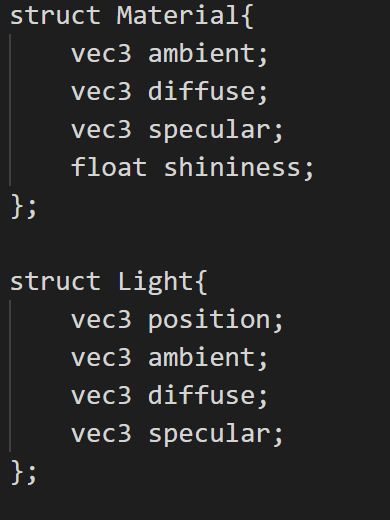
对公式中的参数解释如下：

Ambient：ka为物体材质的环境光反射系数，ia为环境光强度因子；

Diffuse：kd为物体材质的漫反射系数，im,d为光源漫反射强度因子，Lm为光源方向,N为顶点法向量；

Specular：ks为物体材质的镜面反射系数，im,s为光源镜面反射强度因子，Rm为 方向，V为视线方向，指数为高光系数；

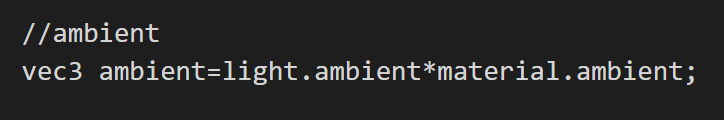
根据Phong模型公式，程序定义了lighting.material.glsl文件，文件中有两个结构体



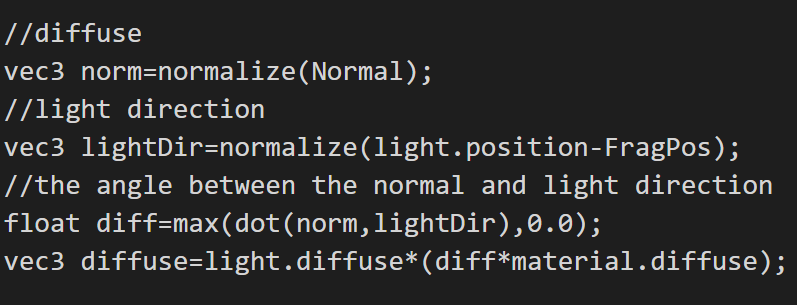
Material用于存储物体材质对环境光，漫反射和镜面反射的参数；Light用于存储光源中三种不同光的数据

然后根据Phong模型中对三种光的计算分别计算物体顶点的环境光，漫反射和镜面反射的结果，将三种光进行加和最后得到的四维向量便是顶点的颜色

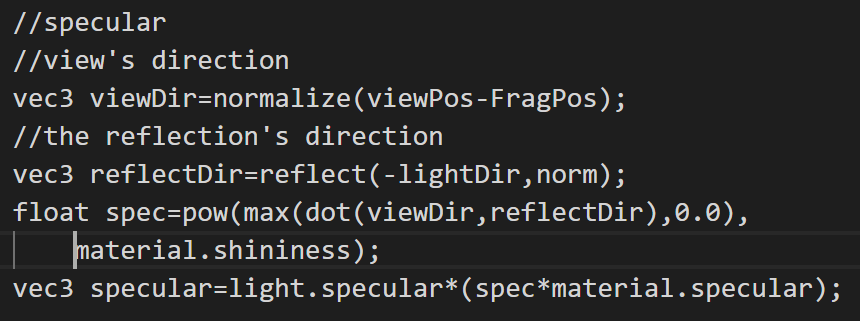
环境光计算如下：



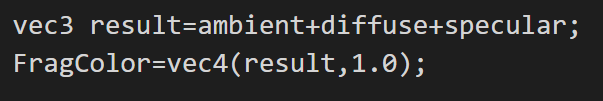
漫反射计算如下：



镜面反射计算如下：



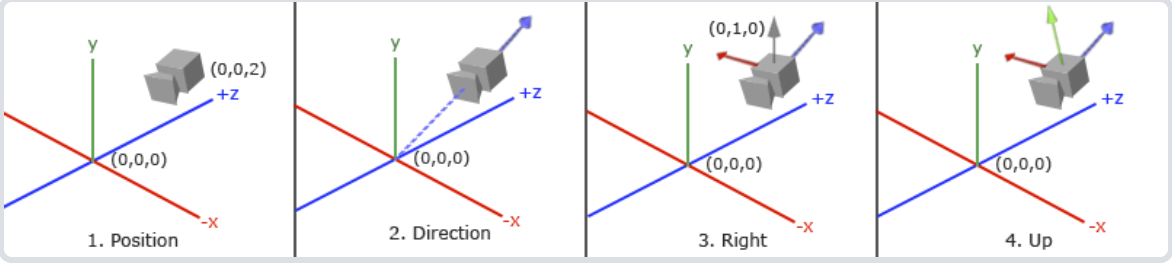
结果如下：



1. 正投影相机

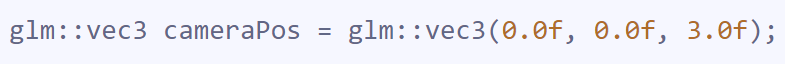
本程序在openGL中配置了摄像机，并通过键盘和鼠标的控制使得用户能够在3D场景中自由移动

摄像机的观察空间是以摄像机的视角作为场景原点时场景中所有的顶点坐标，为了完成程序功能，需要定义摄像机坐标系和得到观察矩阵—将所有的世界坐标变换为相对于摄像机位置与方向的观察坐标。

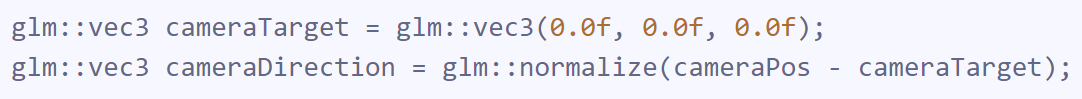


* 定义摄像机坐标系

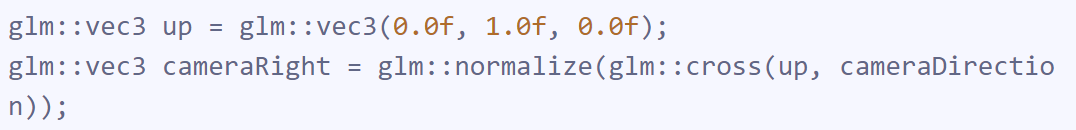
自定义一个摄像机的位置cameraPos;



得到摄像机的方向（摄像机的指向方向,指向z轴正方向），让其指向场景原点(0,0,0), cameraDirection=cameraPos-origin;



计算摄像机的右向量（代表x轴的正方向），先定义一个上向量up，将up与cameraDirection叉乘得到cameraRight;

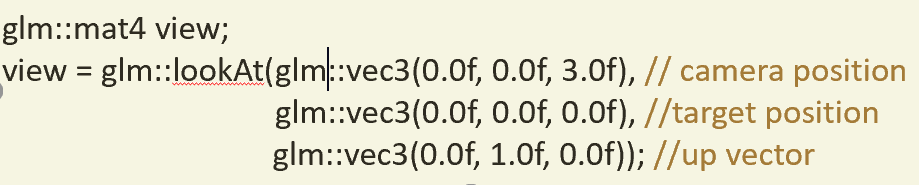


计算摄像机的正y轴方向cameraUp=cross(cameraDirection,cameraRight);



* 得到观察矩阵

openGL中提供了LookAt函数，通过其可以得到观察矩阵

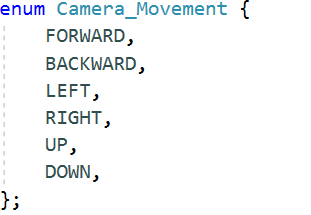


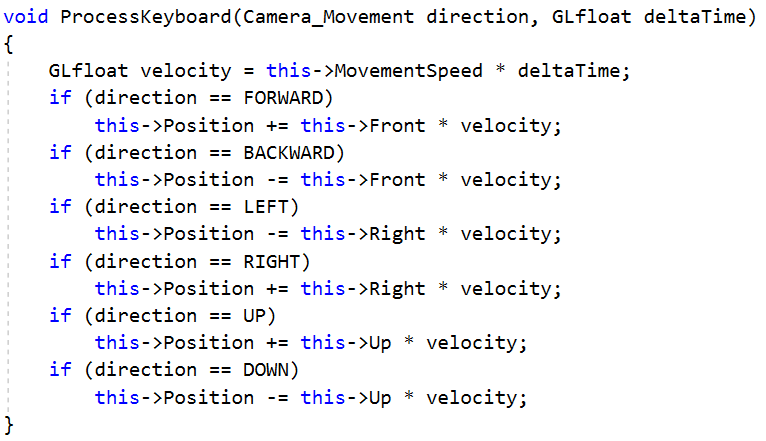
根据摄像机的这些特性，程序中定义了camera.h头文件，其中实现了摄像机的以上特性

* 自由移动

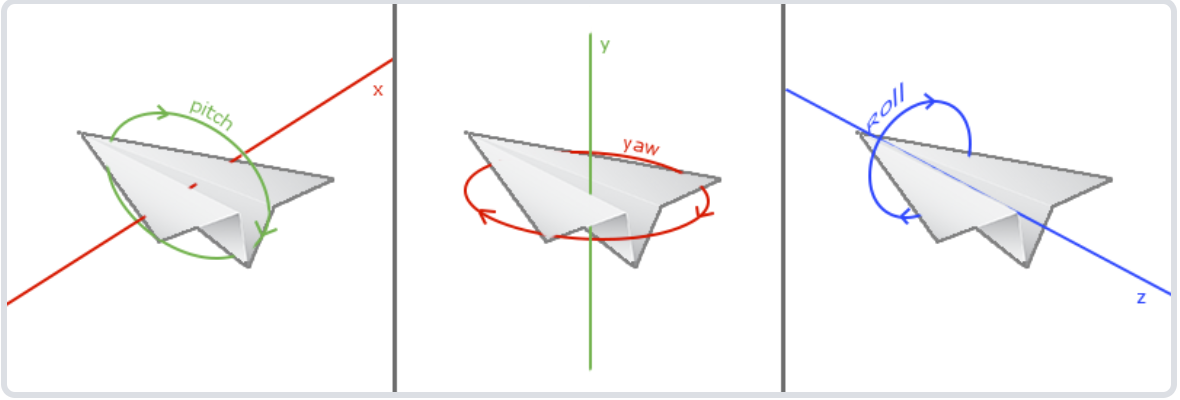
程序实现了通过键盘和鼠标同时配合使得，视点能够在场景中自由移动的功能。要实现此功能，需要对键盘按键和鼠标的移动位置进行监视

对于键盘的输入，使用WASD来进行上下左右的移动，在cemara.h文件中，定义了枚举类来检查相机前后左右的移动。通过对键盘的按键检查，更新当前的摄像机的对应向量；函数ProcessKeyboard()中，除了对按键进行检查，还会对当前帧数时间差进行调整，使得场景移动速度对于每个用户体验都一样





键盘可以控制场景左右移动，而通过鼠标的移动，可以使得摄像机进行旋转，从而使得用户视角进行移动。相机的视角改变需要用到欧拉角的知识，如图所示：



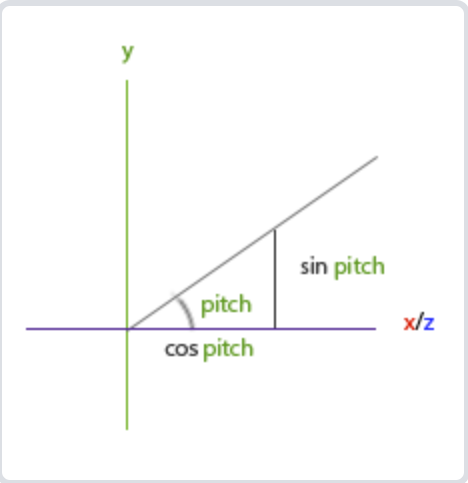
俯仰角(pitch)：往上或往下看的角

偏航角(yaw)：往左或往右看的角

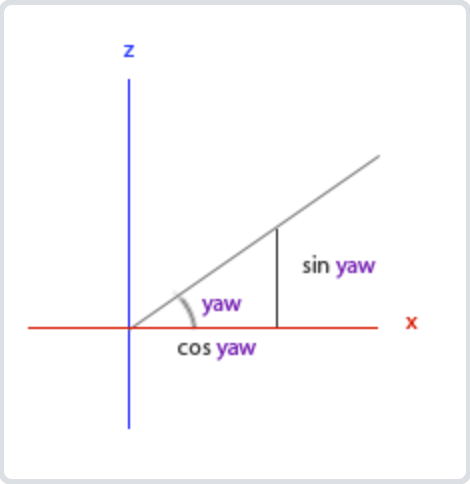
滚转角(roll)：如何翻滚照相机

程序只需要考虑俯仰角和偏航角，

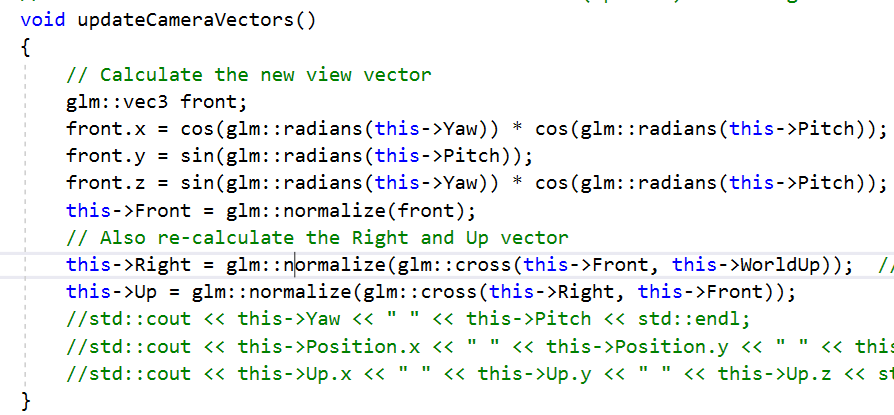
对于俯仰角有

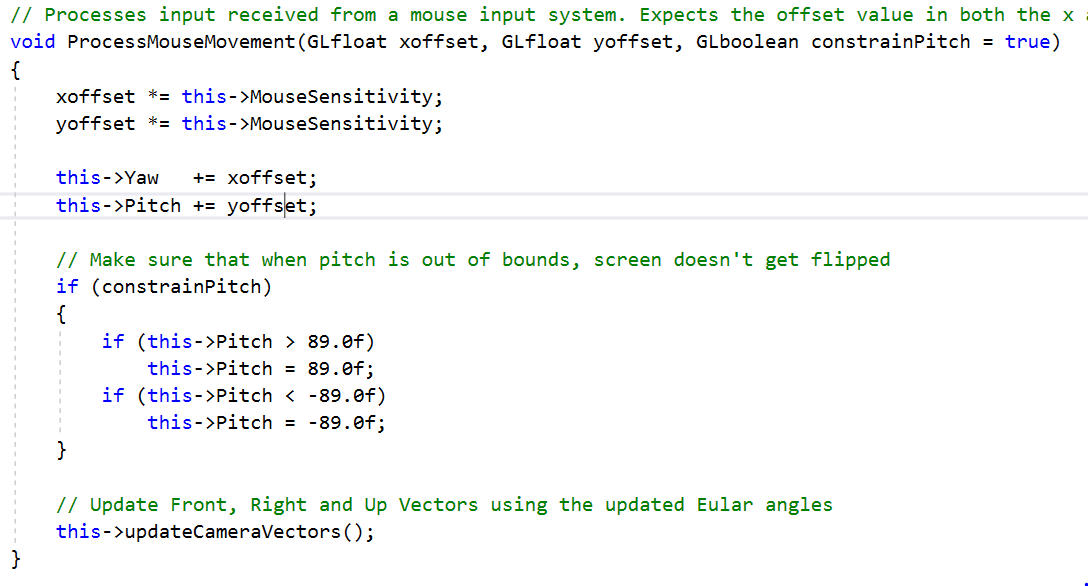


对于偏航角有

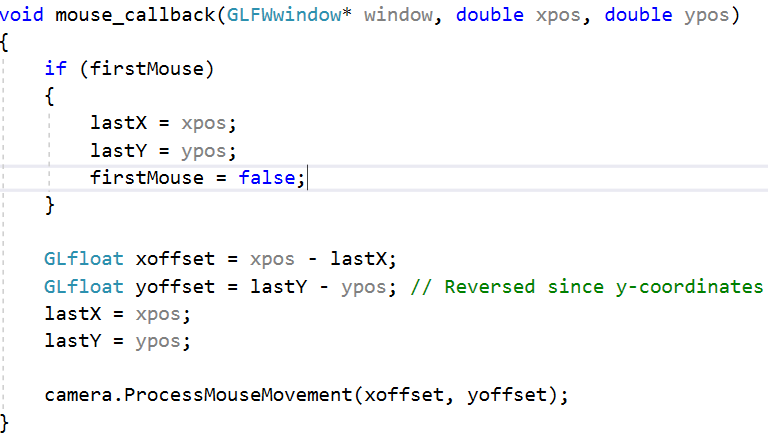


Cemara.h文件中定义了updateCameraVectors()函数，用来根据俯仰角和偏航角的角度更新相机的当前坐标系; ProcessMouseMovement()函数改变当前相机视角

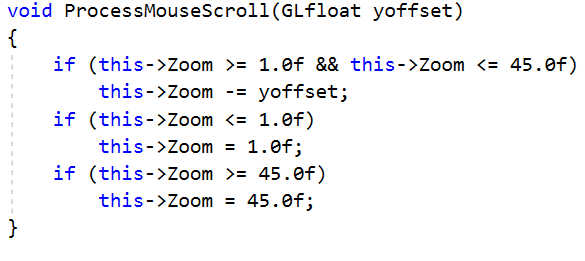




为了跟踪鼠标当前的位置，主程序中使用函数mouse\_callback()记录鼠标的位置信息，并调用camera的ProcessMouseMovement()函数改变相机的位置



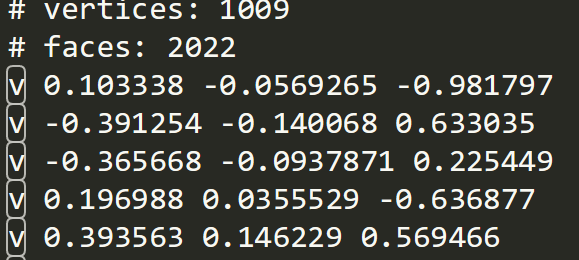
场景中的缩放功能通过鼠标滚轮的滚动来实现，实际上是改变了相机的视锥体范围，决定了我们能看到多大范围的场景。通过改变相机的fov角来实现场景的缩放功能。Cemara.h中ProcessMouseScroll()函数通过滚轮的滚动改变相机的fov角，实现了缩放功能（缩放级别限制在1.0f~45.0f）

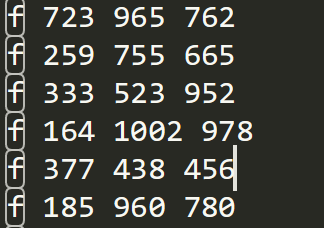


1. 实现细节
2. 从obj文件中读入3D模型

程序自定义了头文件SimpleObjLoader.h，在该头文件中实现了从obj文件中读入模型数据的功能

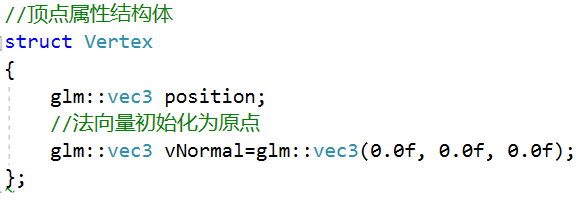
打开obj文件，可以看到数据文件形式如下，其中#为注释行，说明了顶点的个数和面的个数；“v“开头的数据为顶点数据，记录了每个顶点的坐标信息；”f“开头的数据为组成模型的三角面片的顶点数据，每个面由哪三个点组成，记录了组成顶点的下标位置（从1开始）



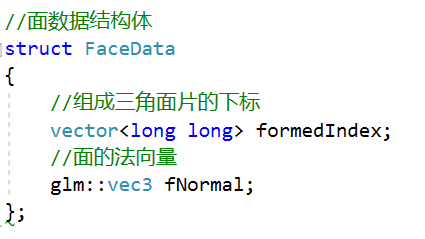


根据obj文件的格式，头文件中定义了两个结构体，

Struct Vertex记录每个顶点的位置信息和法向量信息，法向量用于之后Phong光照模型的计算

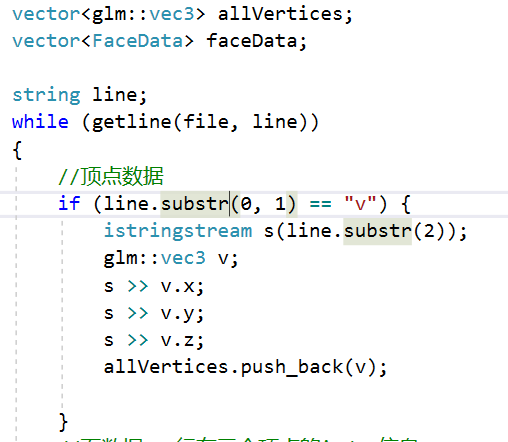


Struct FaceData记录了每个面的顶点下标数组和每个面的法向量信息，用于计算组成该面的顶点法向量

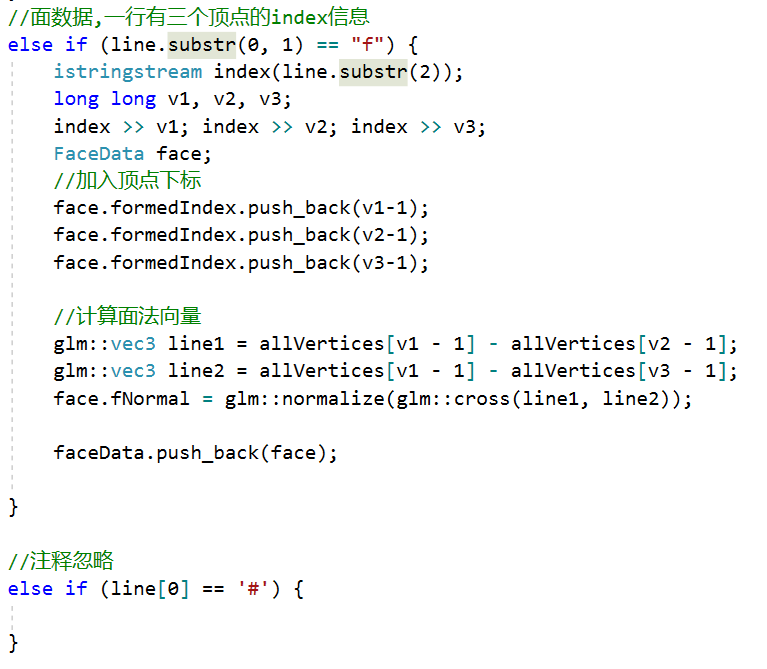


头文件中定义了类ObjLoader，其中的静态方法bool loadObjFile(const char\* path, vector<Vertex> &vertexData)用来从obj文件中将模型信息存入Vertex数组，用于模型的绘制

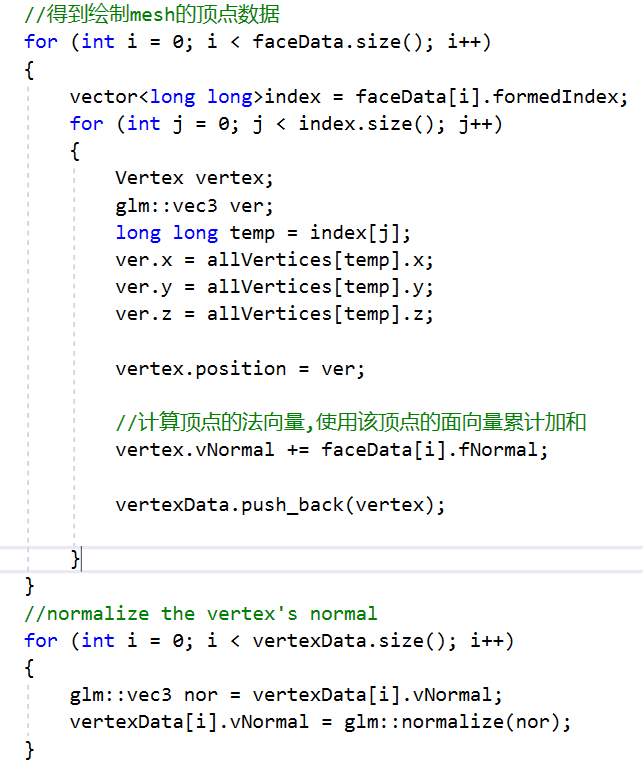
对顶点信息的读取



对面数据信息的读取以及注释的忽略



将顶点信息加入到vertex Data中，计算每个顶点的法向量，并将其标准化



1. Mesh的绘制

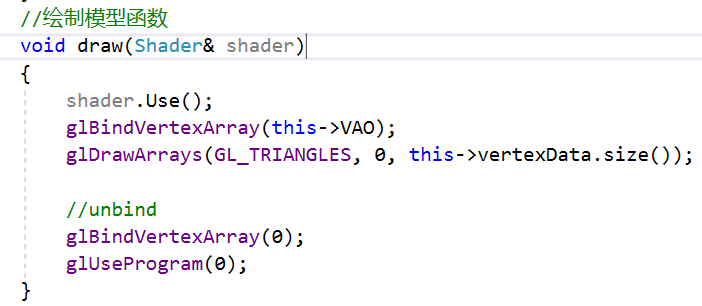
SimpleObjLoader.h中定义了绘制Mesh的类，将绘制模型过程封装



类中setupBuffers()函数根据构造函数中的vertexData顶点数据，生成VAO和VBO顶点缓存对象，将vertexData绑定至OpenGL的顶点数据对象中，读取其中的顶点位置信息和法向量信息

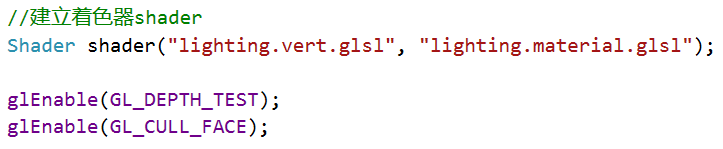


函数draw(Shader&shader)使用传入参数shader进行模型的绘制

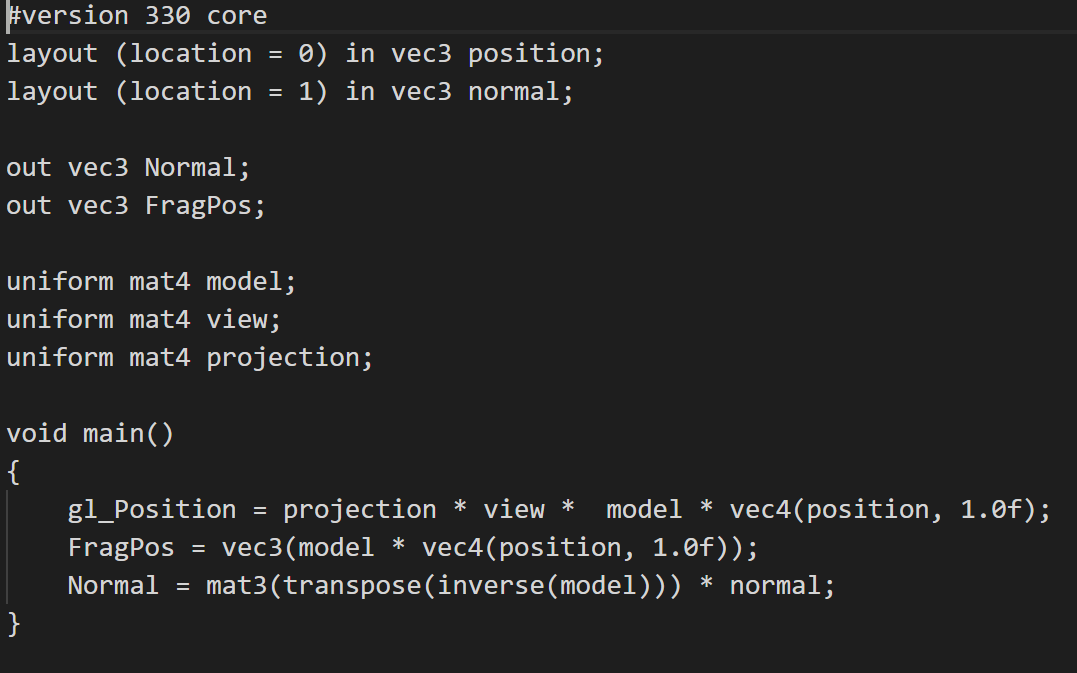


1. 定义lighting.vert.glsl, lighting.material.glsl文件

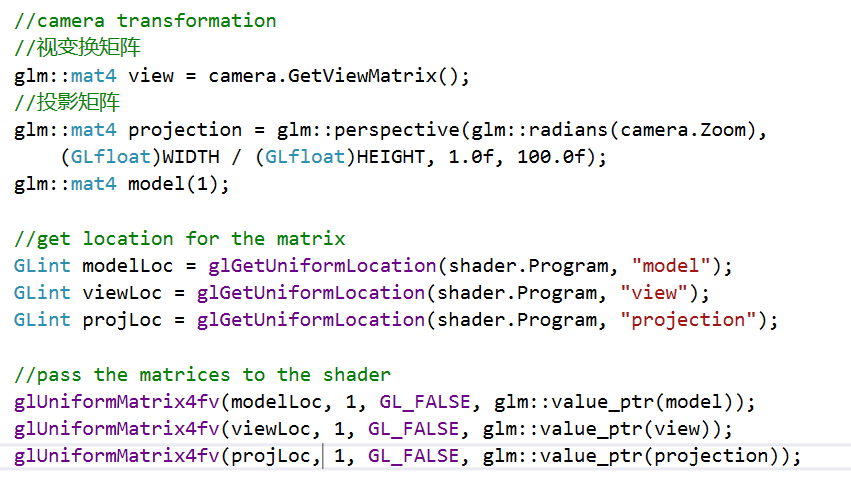
主程序中创建shader对象时，输入该两个文件进行数据的读取和设定，并开启openGL的深度测试和cull face选项



Lighting.vert.glsl文件中，定义了顶点位置信息和顶点法向量信息，输出为顶点的法向量矩阵，片段fragment的位置信息（经过矩阵变换得到）

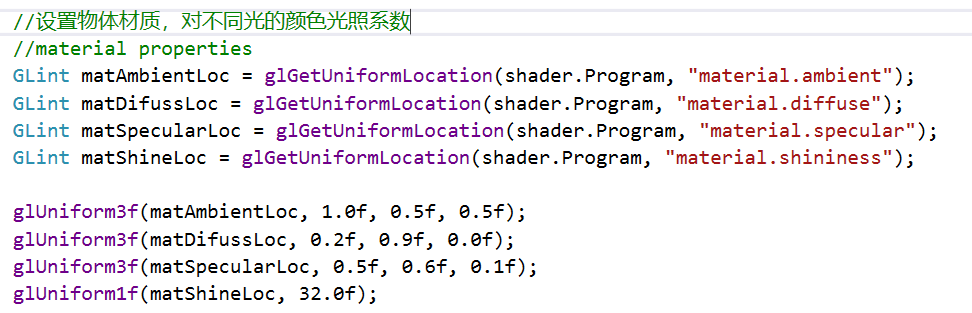


主程序中我们使用openGl提供的函数分别得到模型变换矩阵，投影矩阵，视变换矩阵，将其输入到shader中

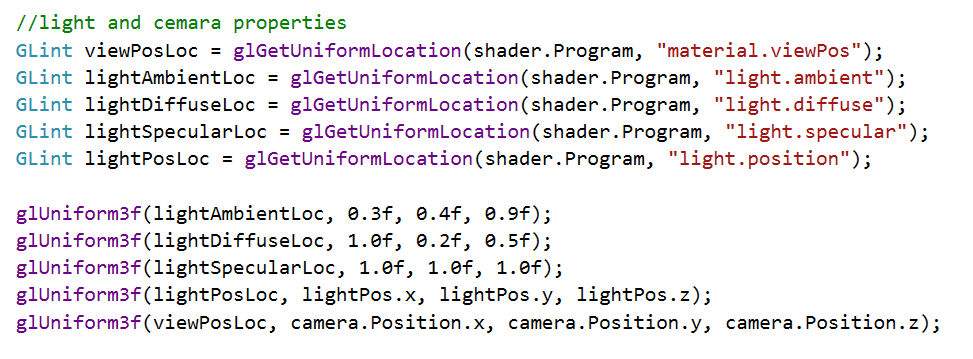


Lighting.material.glsl文件中，定义了物体表面材质结构体和光照信息结构体，并使用Phong光照模型计算物体模型的光照效果（具体讲解见Phong光照模型）；主程序中使用OpenGL提供的函数对物体的材质和光照，视点位置进行设置，

对物体材质设置



对光照和相机信息进行设置

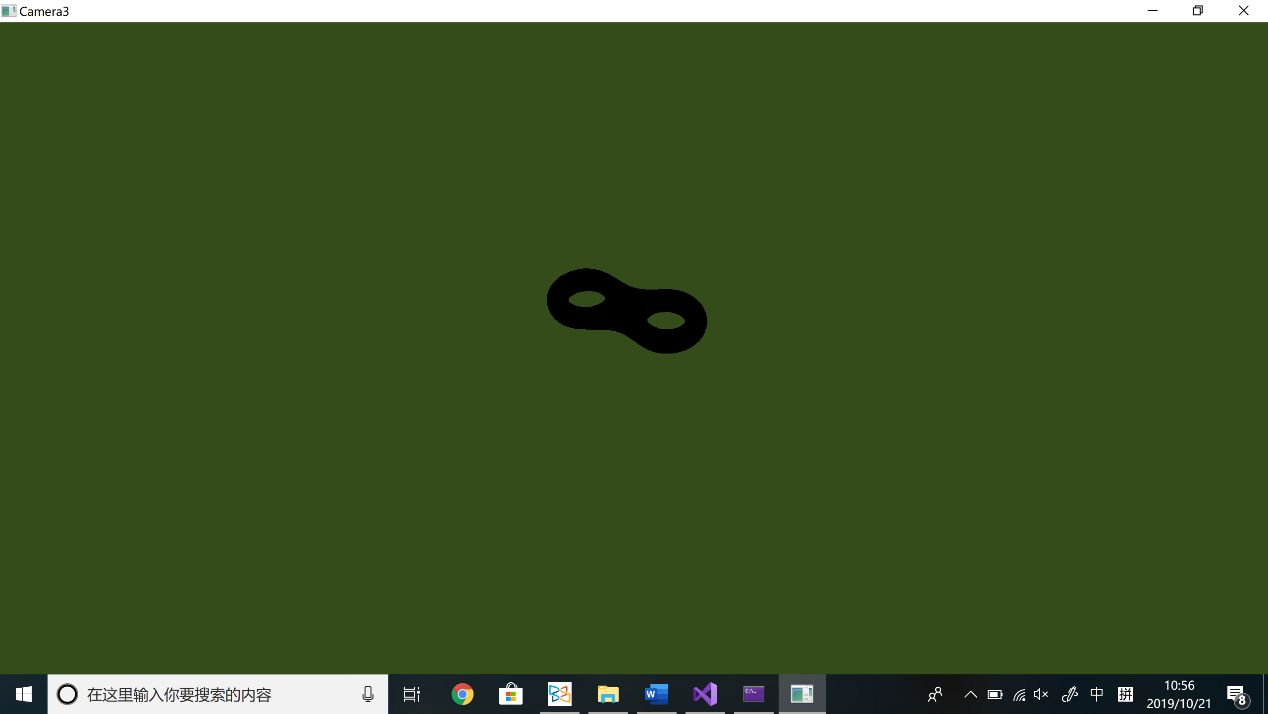


三．结果讨论

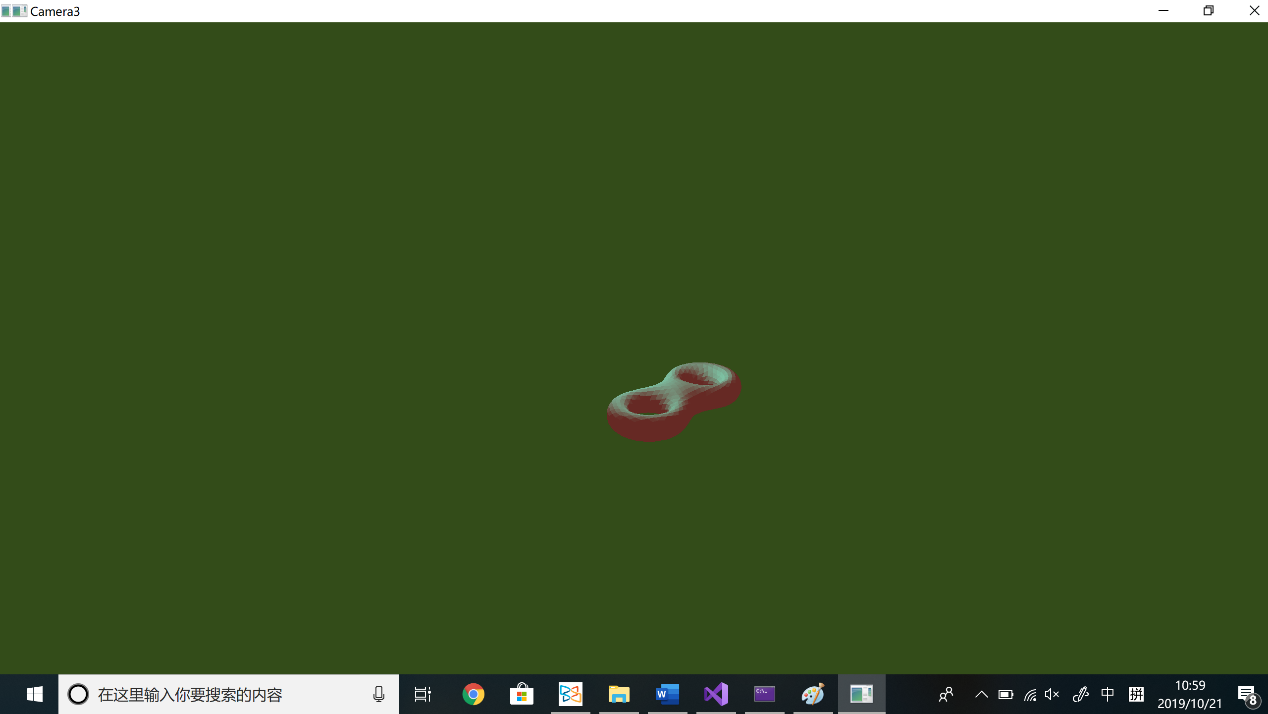
1. 结果

运行程序，通过对物体材质的设定和光源的设定，可看到物体模型展现出不同的光照效果：

环境光效果：



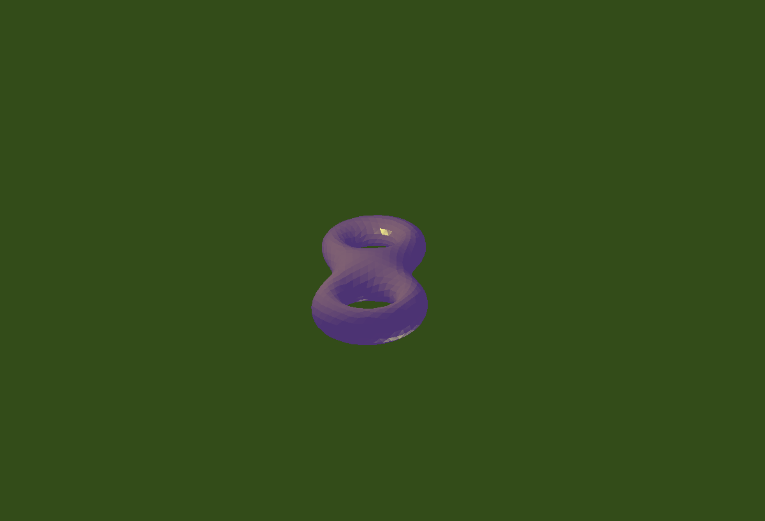
漫反射效果



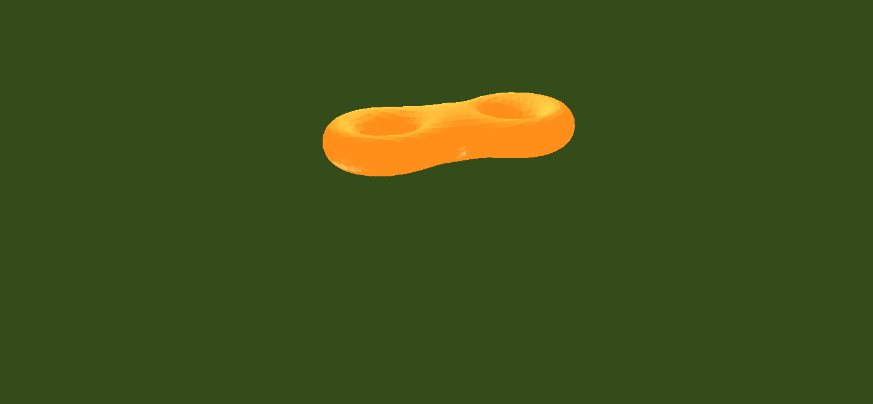
镜面发射效果



结合三种光的光照效果







通过键盘和鼠标移动的配合，可以实现场景的移动和缩放（点击图片播放视频）



1. 讨论

在完成程序的过程中遇到了一些问题，通过上网查询资料，参考他人代码，对问题有了自己的理解和解决方案。

1. 如何读入obj文件

观察obj文件可看到，每行有特定的数据表示字符，分别有v, vn, f, vt四种字符标识。本次程序中读取的模型文件较为简单，只有v，f两种，需要创建适当的数据结构对文件数据进行存储

本程序根据obj文件的数据，创建了Vertex和FaceData结构体分别用以存储顶点数据和面数据，通过参考github上的开源代码，根据程序需求进行适当的修改，成功完成了obj文件的读取和数据的存储。因为无法提前预知数据量的大小，选择可变数组vector解决了数据存储的问题。详细算法参见“主体介绍”->“实现细节”->“读入obj文件”，参考GitHub地址为<https://github.com/wangdingqiao/noteForOpenGL>，参考博客地址为<https://blog.csdn.net/wangdingqiaoit/article/details/51879737>

1. Mesh的概念

在3D图形处理中，一个模型通过由一个或多个Mesh组成，Mesh由顶点，边，面组成，包含了绘制所需的信息，例如顶点位置，纹理坐标，法向量，材料属性等。本程序将模型的顶点数组输入openGL，模型绘制过程封装进自定义的Mesh类，使得主程序中代码简洁化，绘制模型时只需将顶点数组传进Mesh构造函数，提供shader参数至Mesh中draw()函数即可绘制模型；通过将程序的相同操作步骤进行类封装，简化程序操作，使得程序结构更加清晰明了；具体实现参见“主体介绍”->“实现细节”->“Mesh绘制”，参考GitHub代码地址

<https://github.com/wangdingqiao/noteForOpenGL>

1. 法向量的计算

本程序使用的obj文件只提供了顶点的位置信息，但根据Phong光照模型计算光照效果需要每个顶点的法向量信息，因此在读入顶点位置的同时还需要计算每个顶点的法向量信息；对于法向量的计算，要先对面进行法向量计算，从而才能得到组成该面的顶点法向量

面向量的计算如下，假设面f1由顶点v1,v2,v3组成，则

而对于顶点的法向量而言，将该顶点相交的所有面向量进行加和并标准化

代码实现细节参见“主体介绍”->“实现细节”->“读入obj文件”

四．收获和建议

1. 课程收获

计算机图形学课程中，老师讲解清晰，实验课中代码原理实现细节讲解清楚。在课程中，老师会将实验中的基本实现步骤给出，对于本次作业有着很大的帮助。通过对老师的实验指导学习和本次实验的完成，我学习到了很多知识。

* Phong光照模型原理

openGL中使用Phong光照模型，结合物体表面的材质属性来对物体进行渲染和着色。

公式中最终物体的颜色由三种光加和得到，分别为环境光，漫反射和镜面反射；理解了Phong模型的数学原理，通过在openGL中设置材质对不同反射的数据得到不同的光照效果

* 正投影相机原理

通过自定义的摄像机坐标系，可以通过移动照相机达到场景移动的效果。我们需要创建以摄像机为原点的坐标系，通过欧拉角的知识移动摄像机来移动用户视角。在绘制物体模型时，模型是定义在局部坐标系中的，我们要将其转换到摄像机坐标系，需要通过模型矩阵，投影矩阵和视变换矩阵相乘转换物体坐标。

1. 项目开发收获

通过本次实验，学习了Phong光照模型原理和正投影相机原理，并将原理知识结合到实验中完成了程序要求的功能；在实验过程中收获颇丰

* Obj模型数据格式

物体模型数据有多种数据格式存储，本程序中使用的obj格式文件。Obj文件中以数据字符标识标明每一行数据的信息，其中v为顶点位置信息，vn为顶点法向量数据，vt为纹理数据，f为组成模型中面片的数据。了解了obj格式文件后，我们需要在实验中设置数据结构并学习如何读取obj模型，然后绘制模型。本次实验设置了Verex, FaceData结构体用以存储顶点信息和面片信息，在读取文件时按行读取并将数据存入对应的数组中

* 法向量的计算

本次实验中使用的obj文件并没有存储每个顶点的方向量信息，对于Phong光照模型来说，要实现物体的光照效果，需要使用每个顶点的方向向量，程序中对于每个顶点的法向量进行了计算。通过上网查询资料和阅读书籍，对顶点法向量的计算进行了设计，在遍历obj文件行数时，同时对顶点法向量进行计算并存储，为之后模型的绘制和渲染提供数据支持

* 物体材质的设置

Phong光照模型中结合了物体材质属性进行了光照效果的计算。程序中定义了lighting.material.glsl文件，在创建着色器shader时按照该文件进行物体材质的设定，光源信息的设定和相机的设定。结果按照Phong模型计算公式输出最后的光照效果颜色

1. 课程建议

* 希望每一次实验老师可以选取比较好的作品进行展示和讲解，这样可以学习他人好的代码思维，了解到更多的计算机图形学知识
* 希望老师对于下一节课的内容可以提前告知所讲范围，方便学生的预习