**练 习 题 报 告**

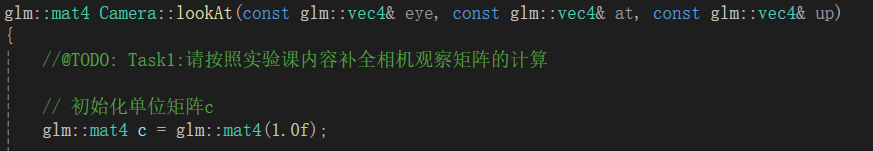
|  |
| --- |
| **课程名称 计算机图形学**  **项目名称 投影和硬阴影**  **学 院 计算机与软件**  **专 业 计算机科学与技术**  **指导教师 熊卫丹**  **报 告 人 杜良衡 学号 2022150255** |

1. **练习目的**
2. 熟悉在OpenGL中实现正交投影变换。
3. 了解使用投影变换实现场景的硬阴影效果。

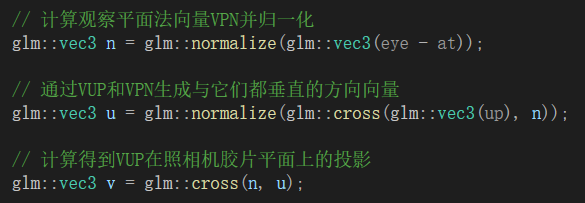
**二．练习完成过程及主要代码说明**

**Task-1：在Camera.cpp中完善lookAt函数、ortho函数和perspective函数**

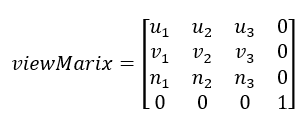
在lookAt函数中先初始化一个单位矩阵c作为相机观察矩阵

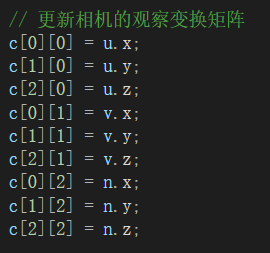
****

接着需要根据观察平面法向量VPN和观察正向向量VUP计算得到的3个向量n、u和v来更新相机的观察变换矩阵，也就是相机的局部坐标系。先计算出VPN再归一化得到向量n，再通过VUP和VPN生成与它们都垂直的方向向量u，最后计算得到VUP在照相机胶片平面上的投影v。这里的函数normalize和cross分别表示向量的归一化与叉乘。

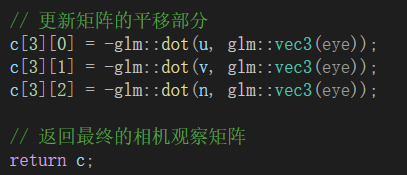
****

然后按照下图更新相机的观察矩阵c

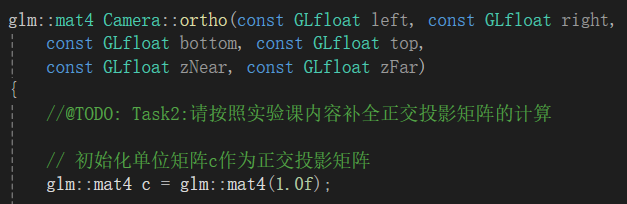
****

****

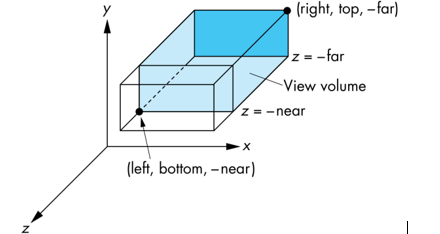
因为在最开始，需要将相机从坐标原点移动到视点，所以要更新观察矩阵的平移部分，这里通过将原始矩阵加上第四个维度，即齐次坐标的方式来实现。

****

在计算正交投影的ortho函数中先初始化一个单位矩阵c作为正交投影矩阵。

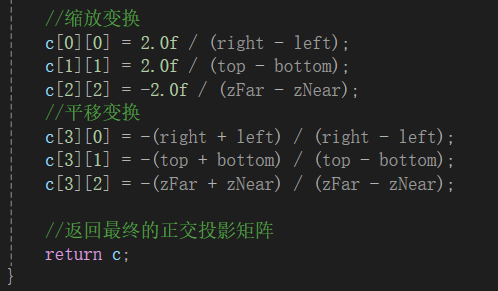


在OpenGL中使用的正交投影是定义在一个平行六面体的视景体中。如下图所示，该六面体由六个参数决定，分别为左右裁剪平面（left和right），上下裁剪平面（top和bottom），远近裁剪平面（near和far）。

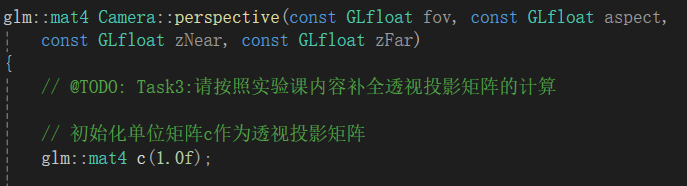


而在OpenGL渲染过程中，需要将定义的正交投影视景体通过平移和旋转变换到标准视景体中。因此，需要按照下图所示矩阵来更新正交投影矩阵c，并返回最终得到的矩阵。

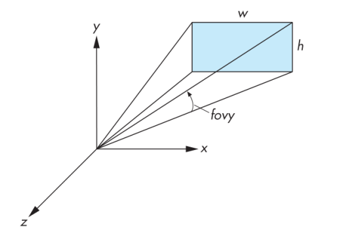
****



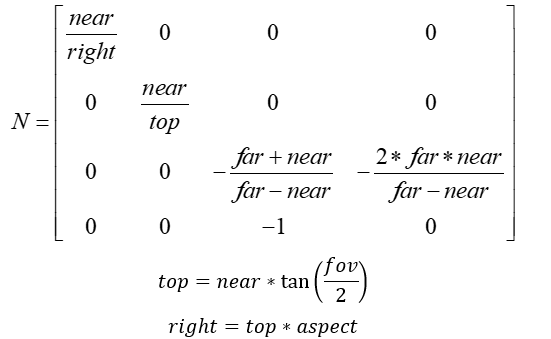
在计算透视投影的perspective函数中先初始化一个单位矩阵c作为透视投影矩阵。

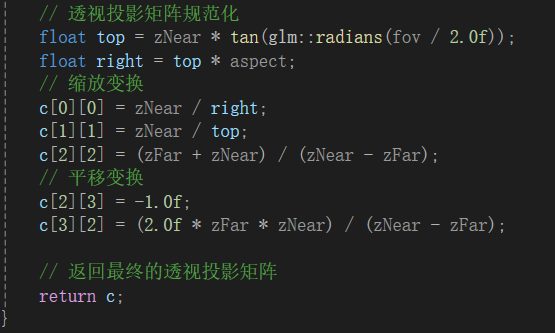


对于透视投影也需要设置视景体来裁剪三维物体。在OpenGl中可以通过视域（FoV，Field of View）来定义视景体，其由视角（Field of View），投影平面长宽比（aspect）和远近裁剪平面（near和far）决定。



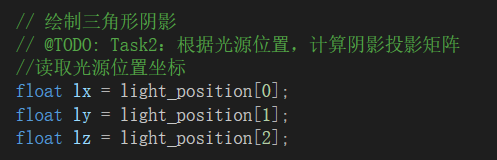
而透视投影同样也需要执行投影规范化过程。按照下图所示矩阵来更新透视投影矩阵c，并返回最终得到的矩阵。



****

**Task-2：在main.cpp中完善display()函数**

在main.cpp的display函数中，先创建3个float类型的变量来分别存储光源位置light\_position的x、y、z坐标。

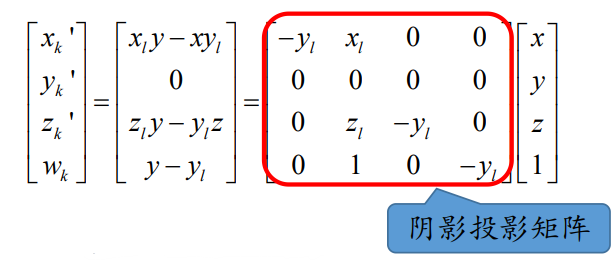
****

假定光源位置在，物体由三角形表示，投影平面上的黑色三角形区域即为阴影，称之为阴影多边形。假设三角形任意一个顶点坐标为，投影到投影平面之后的坐标为，因为该点在 平面上，所以 。根据比例关系可得如下公式：

求解可得，

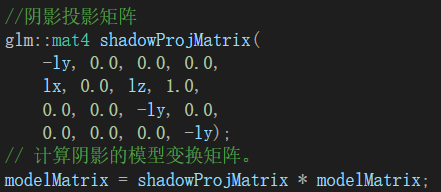
同理可得和。

为了能够方便地通过矩阵表示出投影关系，将所有坐标设置在齐次坐标系下，投影关系就能表示成如下公式。

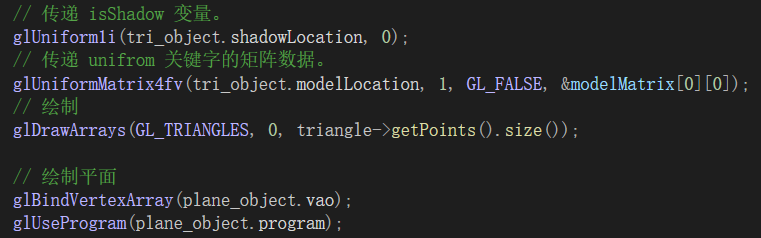


最后可得到：

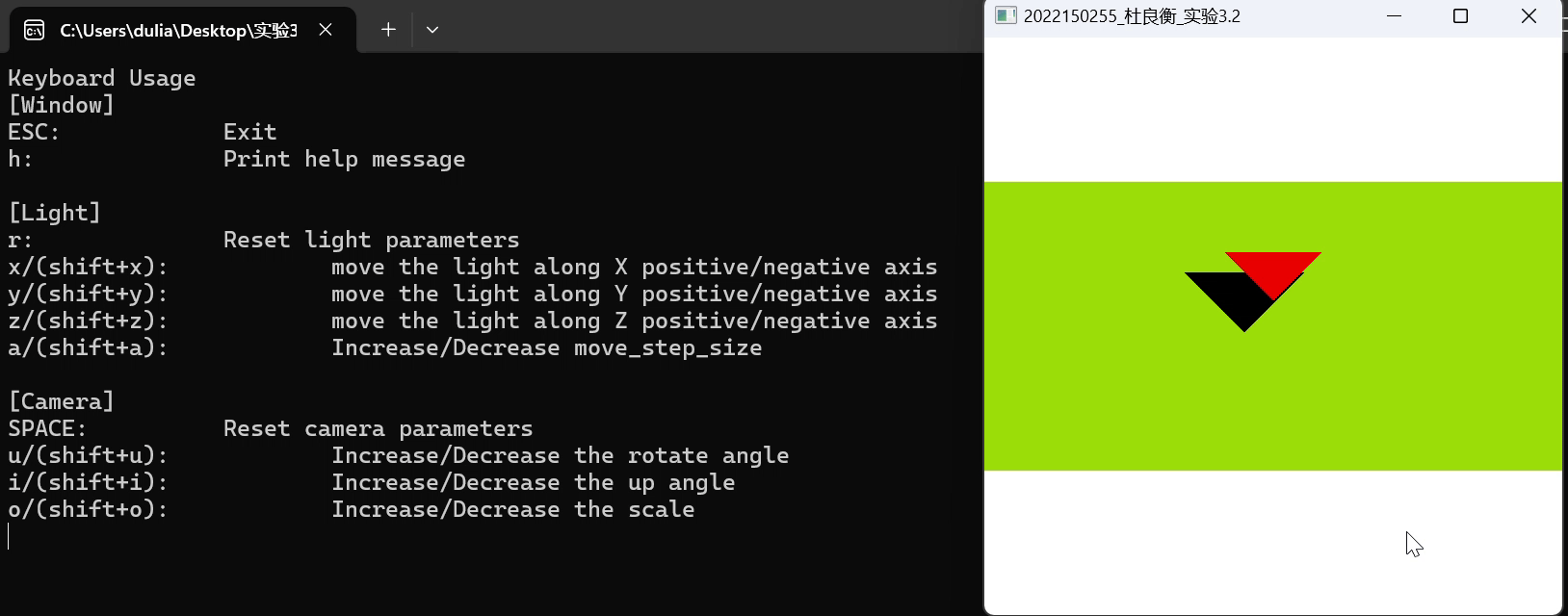
由此，仿照上述公式中的矩阵来初始化阴影投影矩阵shadowProjMatrix，并更新三角形的物体变换矩阵modelMatrix。

****

接着将isShadow变量设置为0，表示绘制阴影，再传递矩阵数据给着色器中的统一变量。最后绘制三角形的阴影以及投影平面。



**最终效果演示**

****