**深 圳 大 学 实 验 报 告**

**课程名称：­ 计算机图形学**

**实验项目名称： 实验三 光照与阴影**

**学院： 计算机与软件学院**

**专业： 计算机科学与技术**

**指导教师： 胡瑞珍**

**报告人： 梁润宇 学号：2021220003 班级： 2班**

**实验时间：2023年 10月30日 -- 2023年 11月19 日**

**实验报告提交时间： 2023年11月19日**

**教务部制**

|  |
| --- |
| 实验目的与要求：   1. 掌握OpenGL三维场景的读取与绘制方法，理解光照和物体材质对渲染结果的影响，强化场景坐标系转换过程中常见矩阵的计算方法，熟悉阴影的绘制方法。 2. 创建OpenGL绘制窗口，读入三维场景文件并绘制。 3. 设置相机并添加交互，实现从不同位置/角度、以正交或透视投影方式观察场景。 4. 实现Phong光照效果和物体材质效果。 5. 自定义投影平面（为计算方便，推荐使用y=0平面），计算阴影投影矩阵，为三维物体生成阴影。 6. 使用鼠标点击（或其他方式）控制光源位置并更新光照效果，并同时更新三维物体的阴影。 |
| 实验过程及内容：   1. 绘制场景、模型   使用实验3.3给出的4个三维场景模型，球、皮卡丘、杰尼龟、粗球体。根据用户的键盘交互来决定使用哪一个off文件。    图1-读入三维场景文件  为了与黑色的阴影区分，场景的背景色设置为灰色。在init函数里进行相应的修改。  具体代码部分为：glClearColor(0.3f, 0.3f, 0.3f, 1.0f);   1. 设置相机   相机用于生成视图矩阵和投影矩阵。接下来对各个函数进行解释。   * 1. lookAt 函数：   生成一个观察矩阵，用于将场景从世界坐标系变换到相机坐标系。    图2-lookAt函数  构建坐标系：lookAt 函数会根据提供的 eye（相机位置）、center（相机看的点）、和 up（相机的上方向）来构建一个右手坐标系。  计算观察矩阵：观察矩阵是一个变换矩阵，它将世界坐标系中的点变换到相机坐标系。这个矩阵包含了相机的位置、观察点的位置以及上方向。  基向量的构建：  forward：相机指向的方向，是 center - eye 的单位向量。  right：相机右侧的方向，是 glm::normalize(glm::cross(forward, up)) 的单位向量。  up：相机的上方向，是 glm::cross(right, forward) 的单位向量。  建立变换矩阵：将上述基向量和相机位置组合成一个变换矩阵，这个矩阵将场景中的点映射到相机坐标系。  2.2 updateCamera函数  这个函数主要执行以下操作：  1.设定相机的上方向 (up):  检查是否需要调整相机的上方向。如果 upAngle 大于90度，相机的上方向 (up) 的y分量将被设置为-1，以防止相机坐标系中的 u 向量反转。  2.计算相机的新位置 (eye):  使用极坐标系的坐标转换，根据 radius（半径）、upAngle（仰角）和 rotateAngle（旋转角度）计算相机的新位置。  3.设定相机的朝向 (at)  2.3 ortho函数  ortho函数用于生成正交投影矩阵。正交投影矩阵用于定义一个立方体的视景体，使得在此视景体内的物体在投影后保持其大小不变。    图3-ortho函数  核心部分在矩阵的计算，这是根据正交投影的数学公式计算得来的。  2.4 perspective函数  perspective 函数用于生成透视投影矩阵。透视投影矩阵用于定义一个视锥体，物体在这个视锥体内的部分将被投影到屏幕上。    图4-perspective函数  核心仍然是矩阵计算，这是通过透视投影的数学公式计算得来的。   1. 添加光照和材质效果   3.1 Phong光照效果  Phong光照模型是计算三维图形上物体表面反射光照的一种经典模型，它分为环境光照、漫反射光照和镜面反射光照三个部分。这个模型由Bui Tuong Phong在1975年提出，是一种基于物理的渲染模型，被广泛用于计算机图形学和计算机游戏中。  Phong光照模型的基本组成部分包括：  1. 环境光照（Ambient Lighting）：表示物体表面在没有直接光照的情况下仍然能够反射一定光照。它通常是一个全局常量，与物体的表面颜色相乘。  2. 漫反射光照（Diffuse Lighting）：表示光线以一定的方向撞击物体表面，并根据表面的法向量和光线的方向计算光的强度。漫反射光照的强度取决于光线方向与法向量之间的夹角。  3. 镜面反射光照（Specular Lighting）：表示光线以特定的方向撞击物体表面，产生明亮的高光区域。镜面反射光照的强度取决于观察方向、反射方向和表面的反射光度（shininess）。  Phong光照模型的计算可以使用以下公式表示：    公式1-Phong光照模型  具体代码实现中，顶点着色器（Vertex Shader）主要负责将顶点的位置和法向量传递到片段着色器，并进行变换到裁剪坐标；片段着色器（Fragment Shader）计算光照效果，包括环境光照、漫反射光照和镜面反射光照。这些计算涉及光照、材质和视图方向的向量运算。    图5-顶点着色器代码  顶点着色器执行了以下操作：  顶点坐标变换：通过模型变换矩阵 (model) 将顶点从对象坐标系变换到世界坐标系。  透视除法：为了获得正确的位置，执行透视除法，将裁剪坐标中的齐次坐标变为非齐次坐标。  相机和投影变换：考虑相机观察矩阵 (view) 和投影矩阵 (projection)，将顶点变换到裁剪坐标系。  传递给片元着色器：将变换后的顶点位置 (position) 和法向量 (normal) 传递给片元着色器，用于光照计算。  有一行代码被注释掉，即 // position = vPosition; 和 // normal = vNormal。原本传递给片元着色器的顶点位置和法向量直接使用输入的顶点位置和法向量。被注释掉的这行代码被替换为通过矩阵变换计算得到的位置和法向量，以便在经过模型变换后的坐标系中正确计算光照效果。    图6-片段着色器代码  片段着色器执行了以下操作：  阴影处理：如果 isShadow 变量为1，表示当前像素处于阴影中，将颜色设置为全黑。  光照计算：根据Phong光照模型，计算环境光分量、漫反射分量和镜面反射分量，并根据 lightype 变量选择输出的光照效果。  颜色合并：将三个光照分量合并，得到最终的颜色。设置透明度为1.0。  总的来说就是计算光照效果，并根据光照类型进行选择输出。  3.2 物体材质效果    图7-材质效果设置  每种材质都有其独特的环境光、漫反射、镜面反射和高光系数值，这些值影响光如何与表面相互作用。网上可以找到非常多材质，在本实验中我选择使用翡翠材质。   1. 添加阴影效果     图8-display函数绘制阴影  阴影部分其实就是把图形投影到平面上，然后颜色设置为黑色。之前已经把背景色设置为灰色避免冲突了。    图9-绘制阴影矩阵  和实验2.1不同，本实验的阴影是投影到y\_min，而不是y=0处。所以要对绘制阴影的函数进行修改。同时还设置了光源不能低于0.5,不然阴影效果会很奇怪。   1. 交互控制光源位置并更新阴影     图10-根据鼠标点击设置光源位置  获取鼠标交互信息，据此设置光源位置。都是调用最基本的函数决定的。 |

|  |
| --- |
| 实验结论：  本实验提供4个三维模型，按qaws进行切换。按u,i,o可以进行旋转和缩放。按123456789可以对光线效果进行调整。鼠标点击的位置为光源位置（过低会设置为默认最低位置y=0.5）。    图11-使用说明      图12-运行结果截图    动图1-运行结果 |
| 指导教师批阅意见：  成绩评定：  指导教师签字：  年 月 日 |
| 备注： |

注：1、报告内的项目或内容设置，可根据实际情况加以调整和补充。

2、教师批改学生实验报告时间应在学生提交实验报告时间后10日内。