**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： 周虹**

**报告人： 林浩晟 学号： 2022280310 班级： 01**

**实验时间：2024年 10月22日 -- 2024年 11月25 日**

**实验报告提交时间： 2024年11 月24 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. 首先更改窗口大小与名称，这里将窗口名称不能设置“实验三”，如果将数字“3“三改为中文“三“就会报错，猜测是与源文件（这里的copy的是实验3.4的源文件）的UNICODE宏定义有关，由于unicode占用的字符过多导致报错；将编码改为GBK也无法解决；        1. 复制实验3.4的代码，使用Trimesh.cpp中的readOff函数获取文件；原理是用fin打开文件，再提取所需要的vertices、faces和edges，再将获取到的数据存储起来；最后调用storeFacesPoints函数将得到的数据细化为点、颜色和向量等数据，用于传入gpu。      1. 使用键盘回调响应按键，对应读取不同模型的数据；再设置在display函数中设置窗口的背景颜色为灰色，用glClear初始化才能将颜色正常显示出来；       如图可以看见背景颜色变为灰色了     1. 参照实验3.1可以得到以下：   ①观察矩阵；    ②正交投影矩阵；    ③透视投影矩阵；    ④视锥体投影矩阵，可以用于非对称的视锥体场景；    ⑤更新相机的位置、目标点和上方向向量；    ⑥根据获取键盘响应切换相机参数，比如旋转角度、俯仰角、半径等；     1. 参考实验3.4，可以实现Phong光照效果（fshader）；首先定义N、V、L和R向量；再分别求出环境光照分量、漫反射分量和镜面反射分量，最后合并全部的光照分量。      1. 然后在init函数中设置光源位置、旋转位移与材质，将初始化的数据与定义的vbo和vao绑定传给着色器；      1. 随后根据实验3.2来产生投影；   ①先设置光源位置(这里的light\_position与inith函数中的光源位置一致)与光源移动参数；    ②根据光源位置与投影平面来求投影矩阵，这里的投影平面为y=0；    ③将投影矩阵传给着色器，isshadow设为1，表示画阴影；     1. 设置键盘响应部分；   ①首先先将printhelp部分补充完整；    ②ESC与Print help；    ③更换Model部分；    ④调整环境光变量；    ⑤调整漫反射系数；    ⑥调整镜面反射系数；    ⑦调整高光系数；    ⑧调整光源的位置；    ⑨调整光源更改大小；     1. 实验结果截图   投影截图    调整投影位置 |

深圳大学学生实验报告用纸

|  |
| --- |
| 实验结论：  实验成功完成。  实验成功实现了建造球形，加入光源并生成投影；同时可以从不同的方向观察球体并且更改光源生成不同的投影。  **实验感想**  通过这次实验，我深入理解了三维场景的渲染流程，包括场景读取、绘制、光照模型、材质效果以及阴影生成等关键技术。实验中，我不仅掌握了相机设置和用户交互的实现方法，还通过实践加深了对坐标系转换和矩阵计算的认识。此外，通过控制光源位置和更新阴影效果，我体会到了光影对于增强三维场景真实感的重要性。这个实验让我对三维模型的建造与投影的渲染更加熟练。 |
| 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。