**深 圳 大 学 实 验 报 告**

**课程名称：­ 计算机图形学**

**实验项目名称： 实验二 三维模型读取与控制**

**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： 吴嘉楷 学号： 2022150168 班级： 国际班**

**实验时间： 2024年 09月24日 -- 2024年 11月04日**

**实验报告提交时间： 2024年10月21日**

**教务部制**

|  |
| --- |
| 实验目的与要求：   1. 熟悉OpenGL 三维模型的读取与处理；理解三维模型的基本变换操作；掌握鼠标键盘交互控制逻辑；掌握着色器中uniform关键字的使用以及数据传输方法。 2. OFF格式三维模型文件的读取：完成对OFF格式三维模型文件的读取与显示，可改变物体的显示颜色。 3. 三维模型的旋转动画：结合模型进行旋转变换的过程，为模型添加自动的旋转动画。 4. 键盘鼠标的交互：通过键盘设定选择绕x、y、z轴进行旋转，鼠标左右键控制动画的开始与暂停。 |
| 实验过程及内容：   1. Copy实验2.3的项目代码作为基础代码，在此基础上进行改动     图1 基础代码的目录结构   1. 使用实验二的Model文件夹提供的off文件，以绘制牛的图像 2. 将Models文件拷贝到TriMess.cpp的同级目录下     图2.1 拷贝Models文件夹   1. 更改init函数以读入cow.off文件     图2.2 读入cow.off模型文件  代码说明：  TriMesh 类中封装了两个关键方法：generateCube 和 readOff。其中，generateCube 方法专门用于生成立方体的几何表示，而 readOff 方法则负责读取 OFF 文件格式的模型数据。鉴于本次实验聚焦于绘制牛的图形，因此，在初始化函数 init 中，我需要使用 readOff 方法来加载 cow.off 模型文件，以便准确获取并呈现牛的三维模型数据。   1. 修改牛的颜色 2. 在TriMesh.cpp中定义一个牛的基准颜色cow\_colors     图3.1 定义cow\_colors   1. 修改vertex\_colors数组的值   在readOff函数中，将读取到的颜色 + 基准颜色所得到的新颜色push进vertex\_colors中，从而让颜色在cow\_colors颜色的基础上产生渐变。    图3.2 生成渐变颜色   1. 修改完的新颜色     图4 新颜色效果   1. 修改旋转参数，确定可以合适的初始旋转速度     图5 设置初始旋转速度   1. 定义currentAxios变量用于记录当前的旋转轴     图6 记录当前的旋转轴   1. 修改key\_callback键盘交互回调函数     图7 key\_callback回调函数  函数说明：  参数中，key 表示用户按下的键，action 指示按键的行为（按下、释放或重复按下）。  首先，函数通过 switch 语句检查按键。当用户按下 ESC 键时，action 被检测为 GLFW\_PRESS，此时调用 glfwSetWindowShouldClose 将窗口的关闭标志设置为 GL\_TRUE，让程序退出。对于 X、Y 和 Z 键，如果 action 是 GLFW\_PRESS 或 GLFW\_REPEAT，程序会将 currentAxios 设置为相应的 X\_AXIS、Y\_AXIS 或 Z\_AXIS，用于控制模型围绕这些轴旋转。  此外，R 键用于增加变化量，通过调用 updateDelta(1) 实现，而 F 键用于减少变化量，调用 updateDelta(-1) 来实现调节。最后，按下 T 键时会调用 resetTheta()，将所有旋转参数重置为初始状态。   1. 修改resetTheta函数，需要reset旋转轴为x轴（默认轴）     图8 修改resetTheta函数   1. 定义鼠标点击回调事件     图9 mouse\_button\_callback函数  函数说明：  mouse\_button\_callback函数通过修改一个布尔变量 isRotate 来控制是否开启旋转动画。IsRotate将在main函数中使用，用于判断当前是否为旋转状态。  首先，检查按下的鼠标按钮 button 是否是左键（GLFW\_MOUSE\_BUTTON\_LEFT），并且检测到的动作 action 是否是按下（GLFW\_PRESS）。如果条件满足，程序将 isRotate 设置为 true，表示开启旋转动画。  接下来，函数又检查是否按下的是右键（GLFW\_MOUSE\_BUTTON\_RIGHT），如果满足条件，同样检测到的动作为按下，此时将 isRotate 设置为 false，表示关闭旋转动画。   1. 在main函数中绑定鼠标回调事件     图10 绑定鼠标回调函数   1. 修改提示语输出函数     图11 printHelp函数   1. 在main函数中添加旋转动画     图12 编写旋转动画逻辑  代码说明：  在循环体内，首先检查 isRotate 变量。如果 isRotate 为 true，表示启用了旋转动画，则调用 updateTheta(currentAxios, 1) 函数更新旋转角度。currentAxios 表示当前旋转的轴，第二个参数 1 表示旋转的增量值。  接下来调用 display() 函数，该函数负责渲染当前的图形或场景。此处的 display() 函数通常会包含 OpenGL 的绘制命令，用于将更新后的场景绘制到屏幕上。   1. 修改运行窗口的标题、尺寸等属性参数     图13 修改窗口属性   1. 运行效果：   图片包含 地图  描述已自动生成  图14.1 绕X轴旋转（默认）  图片包含 图标  描述已自动生成  图14.2 绕Y轴旋转  卡通人物  低可信度描述已自动生成  图14.3 绕Z轴旋转 |

深圳大学学生实验报告用纸

|  |
| --- |
| 实验结论：  通过本次计算机图形学实验，我对OpenGL相关知识有了更为深刻的理解与应用能力的提升。  我不仅熟练掌握了OpenGL在三维模型读取与处理方面的技巧，还深入理解了三维模型的基本变换操作原理。实验过程中，我成功实现了对OFF格式三维模型文件的读取与显示，并能够灵活调整物体的显示颜色，从而增强了模型的视觉效果。  在三维模型的旋转动画方面，我结合模型的旋转变换过程，巧妙地在main函数中添加了自动旋转动画，使模型呈现出生动、动态的效果。同时，我通过精心设计的键盘与鼠标交互逻辑，实现了对旋转轴选择的精确控制，以及动画开始与暂停的便捷操作。  在着色器编程方面，我深入掌握了uniform关键字的使用以及数据传输方法。通过巧妙地运用这些知识，我成功地将OFF格式的三维模型文件整合到项目中，并实现了对模型颜色的渐变调整，使模型在视觉上更加丰富多彩。  综上所述，通过本次实验，我不仅巩固了OpenGL三维模型读取与处理、基本变换操作以及鼠标键盘交互控制等基础知识，还进一步提升了自己的实践能力和问题解决能力。这些宝贵的经验和知识，将对我未来的学习和工作产生深远的影响。  实验难点：   1. **OFF格式文件解析**：OFF格式的三维模型文件包含顶点和面的信息，需要正确解析这些数据并将其转换为OpenGL可以理解的格式，这是实验中的一个挑战。 2. **颜色渐变实现**：为了使模型颜色更加丰富，我需要在读取模型颜色的基础上添加渐变效果。这涉及到对颜色数据的理解和操作，以及如何将这些颜色应用到模型的顶点上。   **解决方法**：在从.off文件中读取的颜色数据的基础上，加上一个rgb偏移值（基准颜色），使得新的颜色数据在基准颜色附近范围内变化）。   1. **旋转动画的控制逻辑：**实现模型的旋转动画需要对OpenGL的变换操作有深入的理解，同时还需要编写键盘交互逻辑来控制旋转轴，这在编程上较为复杂。   **解决方法**：在main函数的循环渲染逻辑中，添加绕轴旋转逻辑。   1. **键盘鼠标交互的实现**：为了实现用户通过键盘和鼠标与模型的交互，我需要编写相应的回调函数，并在这些函数中处理用户的输入，这要求我对OpenGL的事件处理机制有较好的掌握。   **解决方法**：使用glfwSetKeyCallback、glfwSetMouseButtonCallback的OpenGL内置方法绑定键盘、鼠标回调函数，在回调函数中设置不同的反馈逻辑。 |
| 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。