

# 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： 林宪亮 学号： 2022150130 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 29 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. 模型的读取以及颜色改变

### 1.1 模型读取

```
//cube->generateCube();

cube->readOff("./Models/cow.off");
bindObjectAndData(cube, cube_object, vshader, fshader);
```

图 1 模型读取

在实验 2.3 的基础上，调用 readoff 函数，传入 off 格式文件的地址即可。

### 1.2 颜色改变

```
// 根据顶点数，循环读取每个顶点坐标
for (int i = 0; i < nVertices; i++)
{
    glm::vec3 tmp_node;
    fin >> tmp_node.x >> tmp_node.y >> tmp_node.z;
    vertex_positions.push_back(tmp_node);
    vertex_colors.push_back(basic_colors[i%3]);
}
```

图 2 改变颜色

在 readoff 函数中，改变传入 vertex\_colors 的颜色即可改变牛牛的颜色。

## 2. 三维模型旋转动画

```
// 更新旋转角度
if (isRotating) {
    // 根据当前旋转轴进行旋转
    if (currentAxis == Y_AXIS) {
        rotateTheta.y += rotateDelta; // 沿Y轴旋转
    }

    else if (currentAxis == Z_AXIS) {
        rotateTheta.z += rotateDelta; // 沿Z轴旋转
    }

    else if (currentAxis == X_AXIS) {
        rotateTheta.x += rotateDelta; // 沿X轴旋转
    }
}
```

图 3 旋转动画

如图 3，我使用了四个条件判断语句，第一个条件判断语句用于判断当前是否需要自动旋转，后面三个判断语句用于判断是围绕哪个轴进行转动。

### 3. 鼠标交互

#### 3.1 开始与暂停

```
void mouse_button_callback(GLFWwindow* window, int button, int action, int mods) {
    if (button == GLFW_MOUSE_BUTTON_LEFT && action == GLFW_PRESS) {
        // 左键按下, 开始旋转
        isRotating = true;
    }
    if (button == GLFW_MOUSE_BUTTON_RIGHT && action == GLFW_PRESS) {
        // 右键按下, 停止旋转
        isRotating = false;
    }
}
```

图 4 开始与停止

如图 4，我使用两个鼠标事件，使用点击左键控制牛牛开始旋转，点击右键控制牛牛停止旋转。

#### 3.2 绕不同轴旋转

```
// Y: 切换到X轴旋转
case GLFW_KEY_X:
    if (key == GLFW_KEY_X && action == GLFW_PRESS) {
        currentAxis = X_AXIS;
    }
    break;

case GLFW_KEY_Y:
    if (key == GLFW_KEY_Y && action == GLFW_PRESS) {
        currentAxis = Y_AXIS;
    }
    break;

case GLFW_KEY_Z:
    // Z: 切换到Z轴旋转
    if (key == GLFW_KEY_Z && action == GLFW_PRESS) {
        currentAxis = Z_AXIS;
    }
    break;
```

图 5 绕不同轴转动

如图 5，我新加了三个键盘事件 X，Y，Z 分别控制牛牛绕不同的轴进行旋转。

### 4. 实验结果

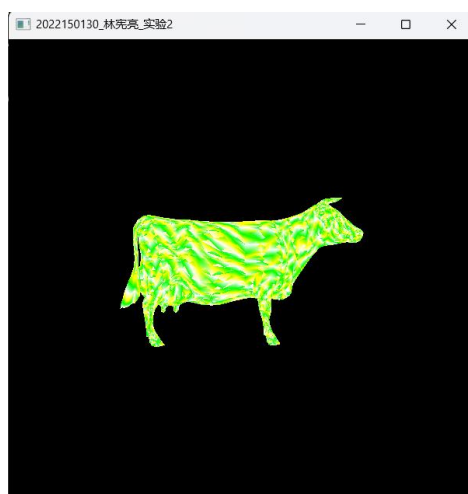


图 6 牛图像

点击运行，如图 6 为牛的图像。

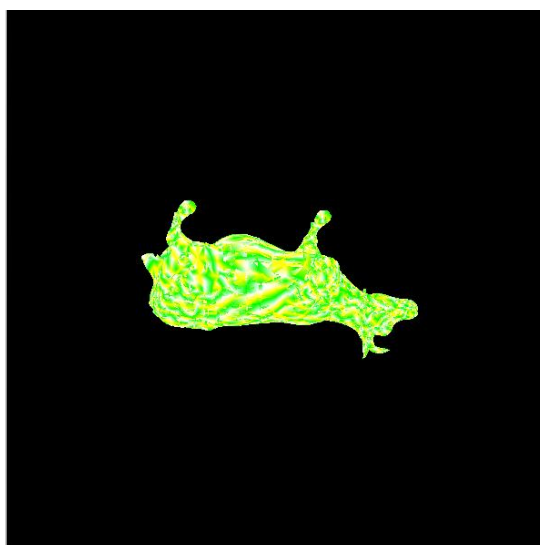


图 7 绕 X 转

如图 7，点击 X，牛牛开始自动绕 X 轴旋转，具体旋转视频可以看下面的 mp4 文件。



绕X轴.mp4

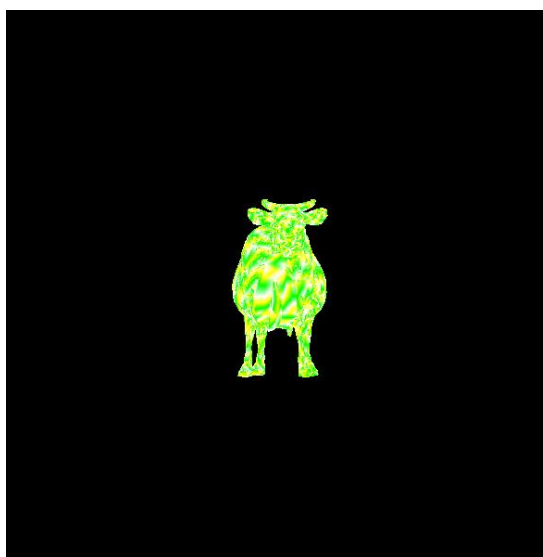


图 8 绕 Y 转

如图 8，点击 Y，牛牛开始自动绕 Y 轴旋转，具体旋转视频可以看下面的 mp4 文件。



绕Y轴.mp4

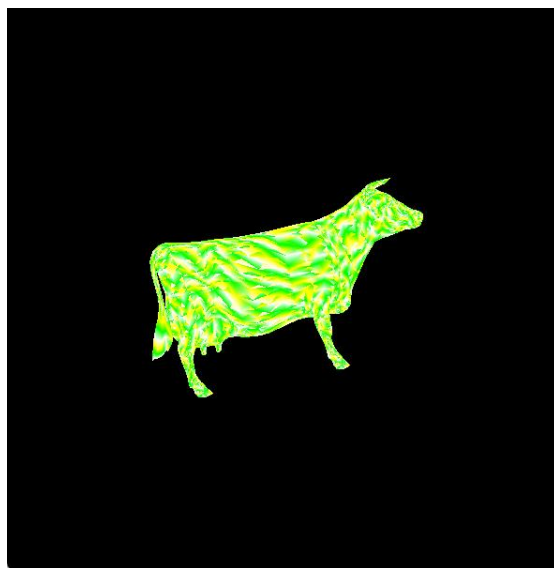


图 9 绕 Z 转

如图 9，点击 Z，牛牛开始自动绕 Z 轴旋转，具体旋转视频可以看下面的 mp4 文件。点击鼠标右键，牛牛就会停止旋转。



绕Z轴.mp4

实验成功!!!

### 实验结论：

在本次实验中，我深入学习并实践了 OpenGL 中的三维模型读取与处理技术，成功实现了对 OFF 格式三维模型文件的读取与显示，并能动态改变模型的显示颜色。这一过程让我加深了对几何变换的理解，尤其是平移、旋转和缩放等基本变换操作。我通过构造变换矩阵，掌握了如何在 OpenGL 中实现物体的平移与旋转。

此外，我实现了丰富的键盘和鼠标交互控制逻辑，允许用户通过按键选择绕 X、Y、Z 轴进行旋转，并通过鼠标左右键控制动画的开始与暂停。这一设计不仅提升了用户体验，也让我理解了事件驱动编程的重要性和实现方法。在此过程中，我还学习并应用了着色器中的 uniform 关键字，成功将变换矩阵和物体颜色等数据传输到着色器中，为运行时的动态效果提供了强有力的支持。

最后，我为三维模型添加了自动旋转动画，观察到不同旋转轴和角度对视觉表现的显著影响。这些实践让我更加熟悉计算机图形学的基本概念，同时为我在未来更高级的图形学应用开发中奠定了坚实的基础。通过本次实验，我不仅提升了编程技能，也加深了对三维图形渲染和动画制作流程的理解。

成绩评定：

指导教师签字：

年 月 日

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。

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。