专业：计算机科学与技术
姓名：方彦祺
学号：3220102829
日期：2024.11.5
地点：无

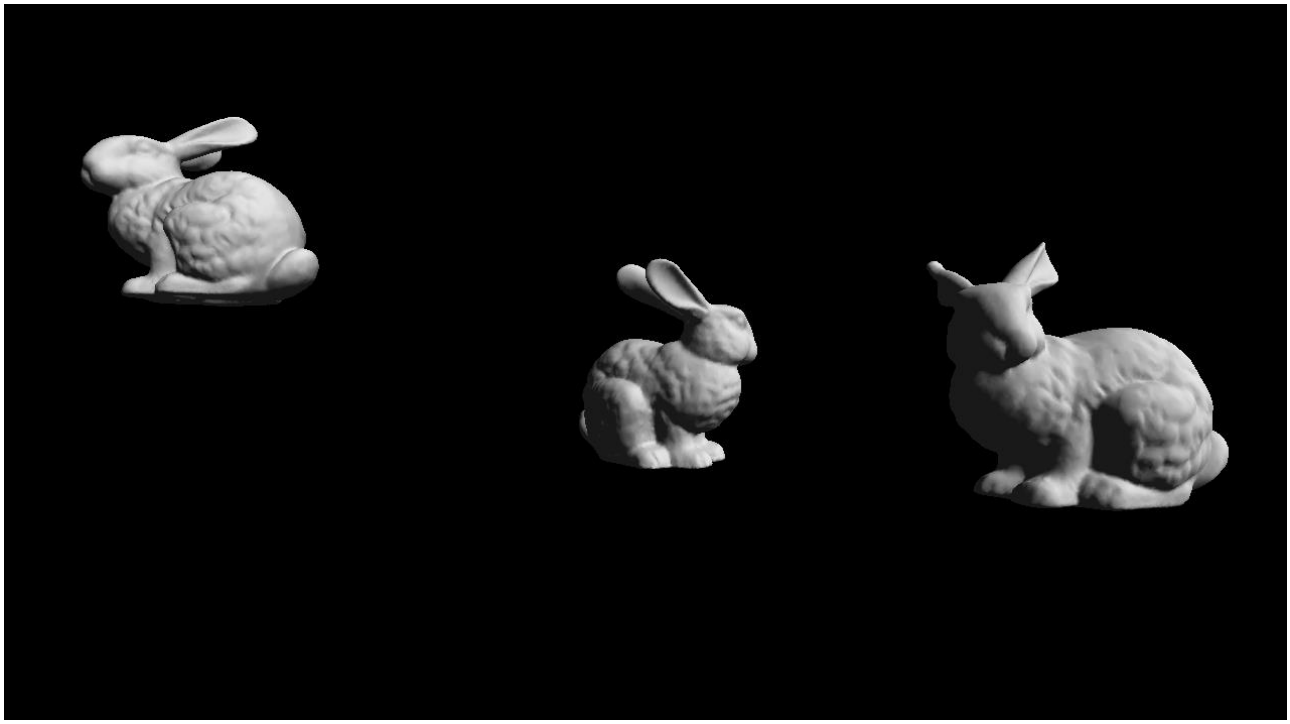
浙江大学实验报告

课程名称：计算机图形学　指导老师：童若锋　成绩：
实验名称：OpenGL 三维变换　实验类型：基础实验　同组学生姓名：无

装 订 线

一、实验目的和要求

在理解 OpenGL 绘制物体的基础上，利用齐次坐标与矩阵变换，绘制出如下周期性变化的三只兔子，其中，其中最左边的兔子循环平移，中间的兔子不断旋转，最右边的兔子循环缩放。



二、实验内容和原理

本次实验内容主要包括绘制三只不同形式运动的兔子，其中需要我们完善的代码部分包括三只兔子的移动、旋转、缩放矩阵的计算。

首先阅读原始代码，在 transformation.h 中发现以下代码：

```
private:
    std::vector<Bunny> _bunnies;

    glm::vec3 _positions[3] = {
        glm::vec3(-10.0f, 0.0f, 0.0f), glm::vec3(0.0f, 0.0f, 0.0f), glm::vec3(10.0f, 0.0f, 0.0f)};

    glm::vec3 _rotateAxis[3] = {
        glm::vec3(0.0f, 1.0f, 0.0f), glm::vec3(0.0f, 1.0f, 0.0f), glm::vec3(0.0f, 1.0f, 0.0f)};

    float _rotateAngles[3] = {0.0f, 0.0f, 0.0f};

    glm::vec3 _scales[3] = {
        glm::vec3(1.0f, 1.0f, 1.0f), glm::vec3(1.0f, 1.0f, 1.0f), glm::vec3(1.0f, 1.0f, 1.0f)};
```

这其中的_position 为兔子的位置向量，_rotateAxis 代表选择绕哪个坐标轴旋转，_rorateAngles 代表旋转的角度，_scales 代表缩放的比例，而他们分别是一个有三个量的数组，每个代表一只兔子。

接下来介绍本实验中我们用到的三个重要的函数的原理：

glm::translate() 创建一个平移矩阵，第一个参数是原始矩阵，第二个参数是一个 glm::vec3 向量，平移向量。

glm::rotate() 用于创建一个旋转矩阵，第一个参数是原始矩阵，第二个参数是旋转的角度，第三个参数是一个 glm::vec3 向量，代表旋转轴。

glm::scale() 用于创建一个缩放矩阵，第一个参数是原始矩阵，第二个参数是一个 glm::vec3 向量，代表 x、y、z 轴的缩放比例。

而后我们注意到源代码将这三个矩阵相乘，这样根据矩阵的性质使这几个变换结合起来，得到的矩阵可以实现平移、旋转、缩放的功能。

```
glm::mat4 model = translation * rotation * scale;
shader->setUniformMat4("model", model);
```

三、主要仪器设备

Windows 系统、Visual Studio 2022.

四、操作方法和实验步骤

transformation.cpp 代码中相应的部分：handleInput 函数

```
// 获取当前时间来创建周期性效果
float time = glfwGetTime();

// 处理最左边兔子的平移：沿Y轴周期性移动
_positions[0] = glm::vec3(-10.0f, 0.0f, 0.0f) + sin(time) * velocity;
// 处理中间兔子的旋转：围绕Y轴周期性旋转
_rotateAngles[1] = angulerVelocity * time;

// 处理最右边兔子的缩放：周期性缩放
float scaleFactor = sin(time) * scaleRate;
_scales[2] = glm::vec3(1.0f, 1.0f, 1.0f) + glm::vec3(scaleFactor, scaleFactor, scaleFactor);
```

我们使用 glfwGetTime()函数获取时间来实现周期性的变化。对于兔子[0]，只有位置需要改变，所以我们更改其位置向量_position[0]而其他向量不变，兔子[1]需要绕 y 轴旋转，我们为其设置旋转角度，而[0]和[2]的旋转角度不变仍未 0（即不发生旋转），对于兔子[2]，需要对其进行缩放处理，所以我们设置其缩放参数，而[0]和[1]的缩放参数保持不变。

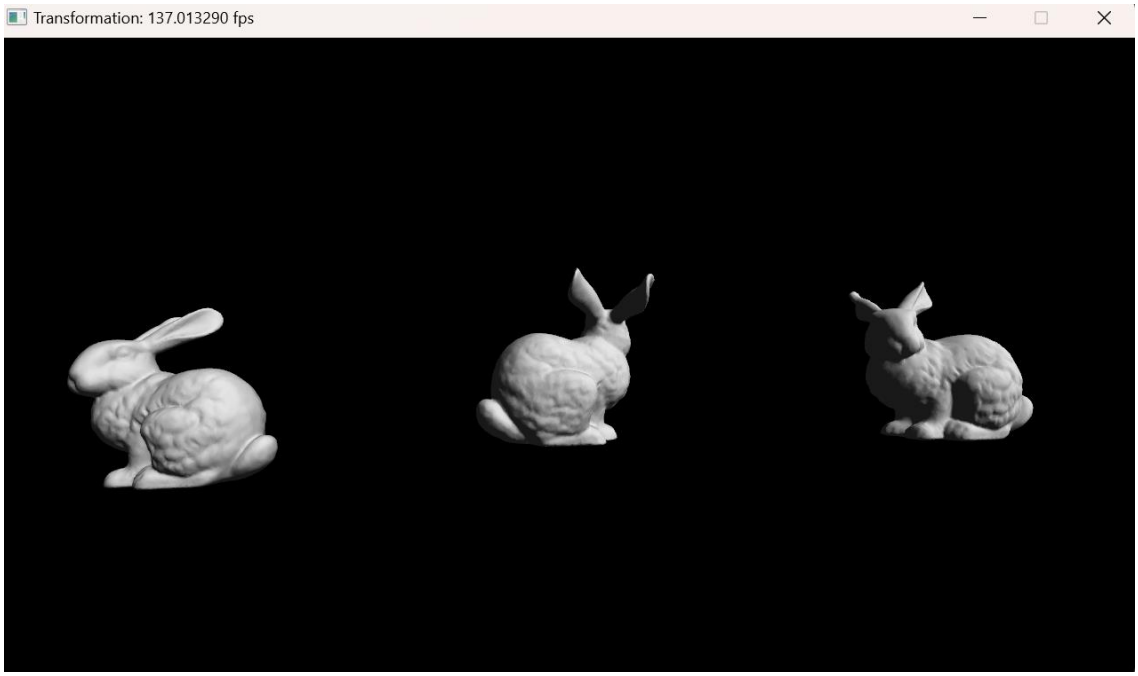
然后是 renderFrame()函数中需要我们完成的部分：

```
translation = glm::translate(glm::mat4(1.0f), _positions[i]);

rotation = glm::rotate(glm::mat4(1.0f), _rotateAngles[i], _rotateAxis[i]);

scale = glm::scale(glm::mat4(1.0f), _scales[i]);
```

使用了实验原理部分所介绍的三个重要的创建矩阵的函数。

五、实验结果与分析

运行，得到结果符合预期：



六、讨论、心得

本次实验需要补充的部分不多，主要是理解这三个创建矩阵的函数以及用矩阵来实现变换的原理，便能很快完成实验。

七、参考链接

[1] 变换 - LearnOpenGL CN (learnopengl-cn.github.io)
[2] 坐标系统 - LearnOpenGL CN (learnopengl-cn.github.io)