

# 图形学上机实验一

---

## 实验目的

利用opengl显示一个三维模型，三维模型需具有一定复杂度，实现对模型进行放缩、旋转和平移。

## 实验环境

- Windows 10 系统
- Visual Studio 2017
- glfw,glad,glm,Assimp,stb\_image

## 代码组成

本实验代码参考了[LearnOpenGL](#)网站教程，如camera类，model类和shader类的实现等。

### 头文件

- camera.h

定义camera类，实现相机，定义相机位置、朝向等参数。

- mesh.h

定义mesh类。mesh是渲染的基本单位，包括顶点数组、索引数组、纹理和OpenGL所需各缓冲对象等成员变量；成员函数setupMesh()初始化缓冲对象，Draw()调用着色器渲染图形。

- model.h

定义model类。加载obj文件，定义模型；利用Assimp库加载obj文件，std\_image类加载纹理；遍历加载的模型数据，将其处理为顶点和网格(mesh)供渲染使用。

- shader\_m.h

定义shader类。用于渲染模型的着色器，接受顶点着色器和片元着色器代码，对常用的着色器操作进行了包装。

- std\_image.h

处理纹理的库的头文件。

### 源文件

- glad.c

GLAD库，用于代替glew。

- model\_loading.cpp

主函数。实现渲染循环及各种事件处理。

- stb\_image.h

处理纹理的库的源代码。

## 资源文件

- basic\_lighting.vs

实现了加载纹理和光照的顶点着色器

- basic\_lighting.fs

实现了加载纹理和光照的片元着色器

## 算法思路

利用全局变量记录模型属性，如模型位置，旋转状态，缩放比例；在渲染循环中读取全局变量构造model矩阵，渲染模型；在事件处理函数中读取键盘和鼠标操作，修改对应的模型属性全局变量，从而实现对模型的平移、旋转和放缩的实时操作。

旋转和放缩操作需要指定不动点，默认不动点为原点，为了实现合理的操作，我们需要将模型的中心指定为不动点，为此我们在进行其它操作前，先执行一个平移操作，使得模型中心与原点重合。

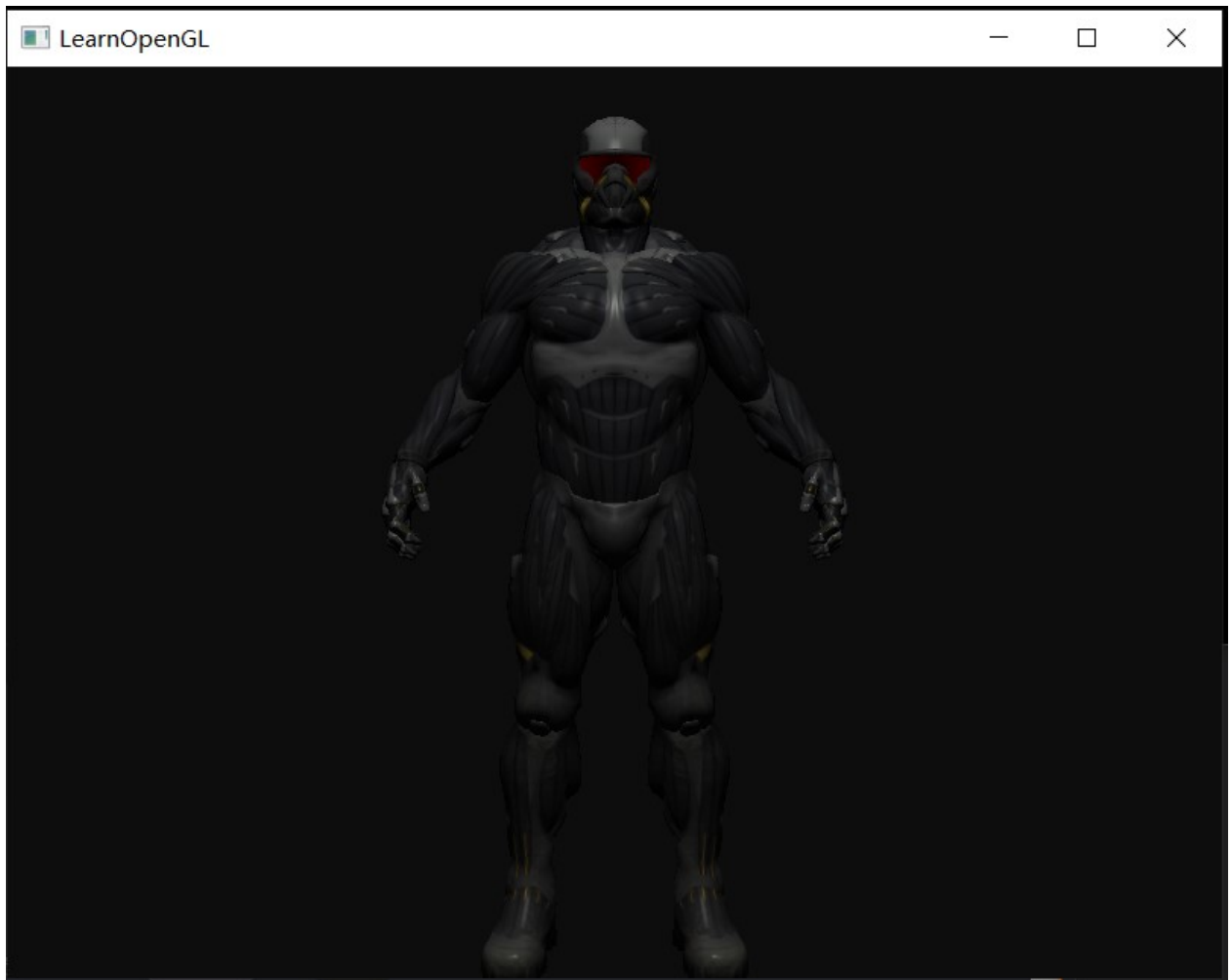
## 算法描述

- 首先平移模型中心：`glm::translate(model, glm::vec3(- x, - y, - z))`，其中x,y,z为模型中心原坐标；
- 执行缩放操作，由鼠标滚轮控制，`scroll_callback()`将滚轮的前后偏移映射到缩放比例的增减；`glm::scale(model,glm::vec3(scale,scale,scale))`构造缩放函数；
- 执行旋转操作，由鼠标左键和右键控制；其中左键摁下控制围绕x轴和y轴的旋转，右键摁下控制围绕z轴的旋转；`mouse_callback()`和`mouse_button_callback()`将鼠标的偏移映射到围绕三个轴旋转的角度:首先维护鼠标按键状态，根据不同键被摁下及偏移修改全局变量旋转角度；利用三个角度构造四元数  
`q=glm::tquat(glm::radians(glm::vec3(angle_x,angle_y,angle_z)))`，利用四元数构造旋转矩阵`glm::mat4_cast(q)`；

- 最后执行平移操作，由键盘上下左右及加号减号控制在三维空间的平移操作；  
processInput()将按键映射到全局位置坐标的变化，构造平移矩阵  
`glm::translate(glm::mat4(1.0f), glm::vec3(pos_x,pos_y,pos_z))`。

## 实验结果

- 模型中心坐标为(0.0,0.0,8.0),初始缩放因子为0.15，光源位置为(0.0,2.0,2.0),光源颜色为(1.0,1.0,1.0),环境光系数为0.1，漫反射和镜面光系数为1.0。



- 通过平移、操作和放缩操作修改参数放缩系数为0.1348,位置为(-0.7128,0.1039,0),旋转角度为(-11.4,61.1,7.3)

