

# 图形学上机实验二

---

## 实验目的

基于实验一的功能，利用slerp和四元数实现模型在两个位置和姿态之间平滑的转换效果。

## 实验环境

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

## 代码组成

本实验代码和实验一代码整合在一起，相当于实验一添加的功能。因此源代码文件组成和实验一相同。

### 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

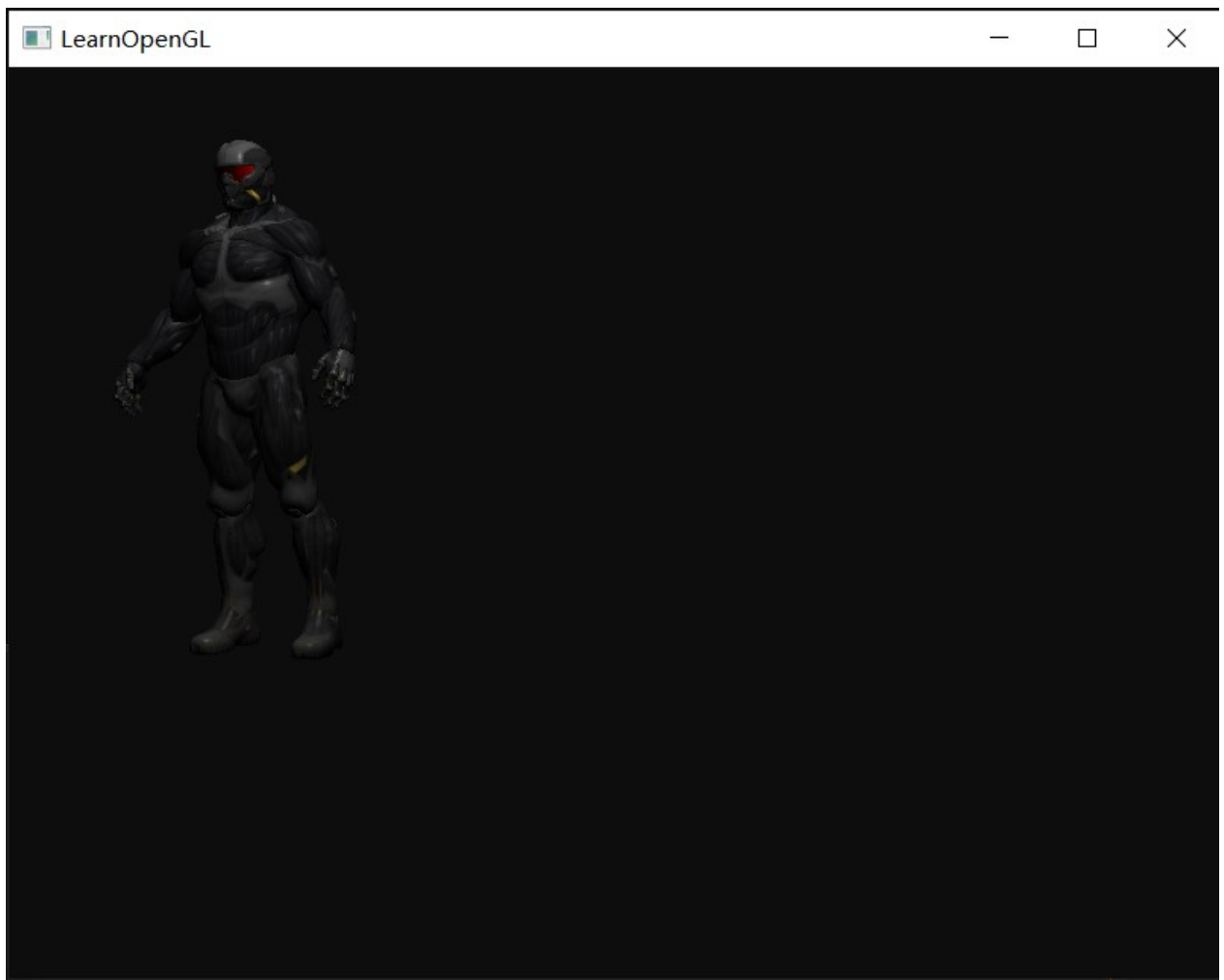
利用全局变量记录移动的初始位置和结束位置，构造四元数，利用`glm::slerp`函数实现旋转角度的插值,从而实现平滑旋转。

## 算法描述

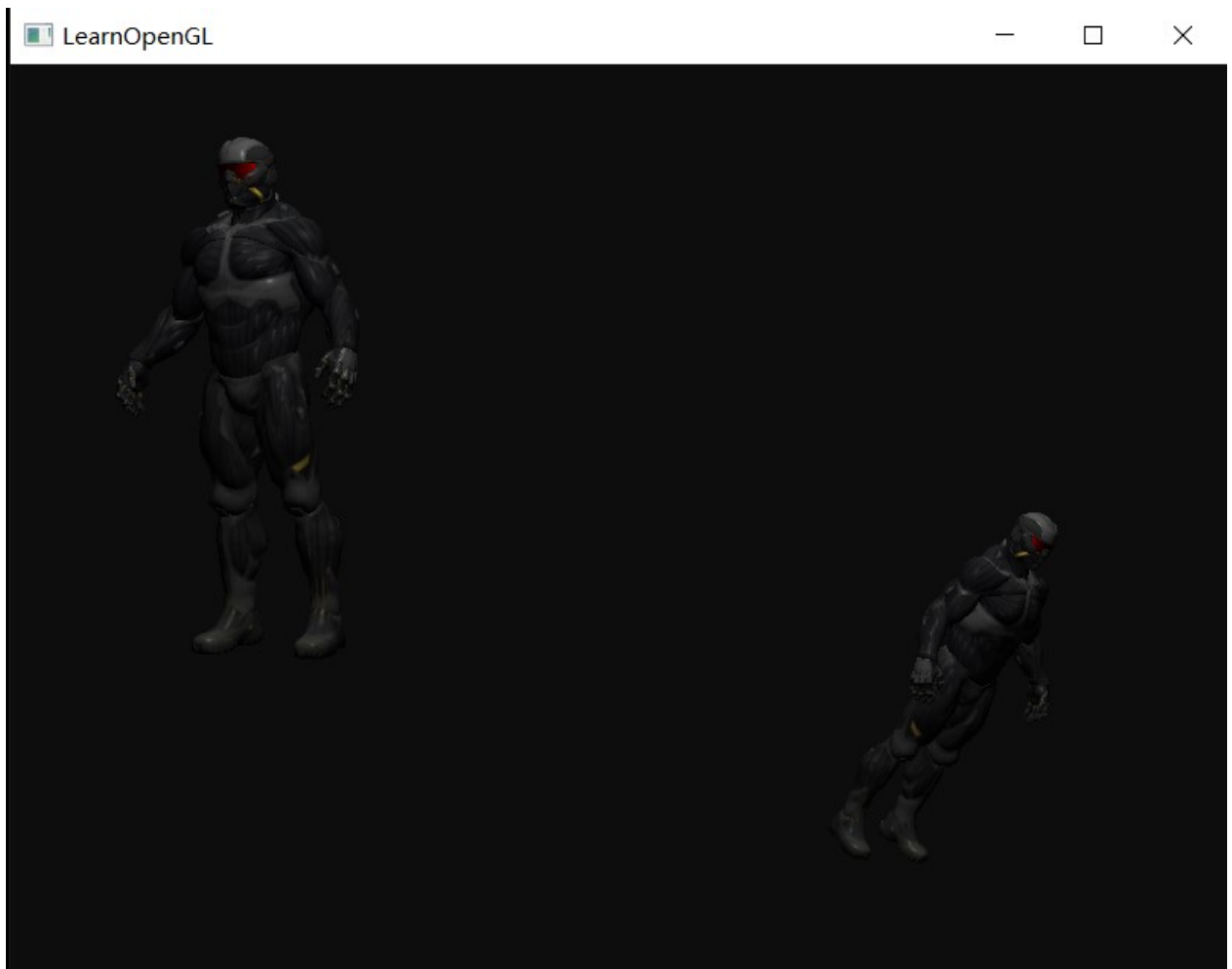
- 利用左ctrl键记录当前位置及姿态，空格键控制开始平滑移动，右ctrl键退出平滑移动模式
- `processInput()`函数记录按键状态，如果左ctrl键按下就进入移动模式并记录当前位置和姿态；如果右ctrl按下就退出移动模式并更新当前位置和姿态；如果空格键按下且位于移动模式，计算插值，渲染模型。
- 四元数插值函数：`glm::slerp(glm::tquat(glm::radians(glm::vec3(angle_x_old, angle_y_old, angle_z_old))),glm::tquat(glm::radians(glm::vec3(angle_x, angle_y, angle_z))),p)`; 参数一是初始状态四元数，参数二是结束状态四元数，参数三是当前帧数。

## 实验结果

- 初始放缩系数为0.09,位置为(-0.9867,0.3581,0),旋转角度为(6.3,-18.8,0)



- 按下左ctrl键进入移动模式记录初始位置，通过平移、操作和放缩操作修改参数放缩系数为0.0675,位置为(-0.93, -0.4351, 0),旋转角度为(15.9,34.1,-14.5)



- 按下空格键开始插值

