实验报告二

姓名：胡力杭

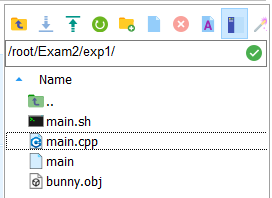
学号：2019K8009926002

**代码明细**：

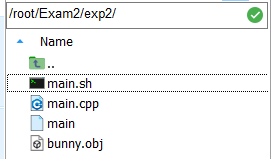
基本实验的主体代码使用了老师上课讲过的示例代码，我们要补充的是头文件、main函数、确定正交投影的位置。进阶实验中我们添加了在鼠标键盘交互功能的实现以及表面法向的计算。

代码和执行脚本放置于/root/Exam2/目录，exp1为立体显示三维模型实验，exp2为鼠标交互及法向计算。

exp1目录包含main.cpp、可执行文件main、执行脚本main.sh、bunny.obj



exp2目录包含main.cpp、可执行文件main、执行脚本main.sh、bunny.obj



**环境配置：**

无

**程序编译及运行命令：**

exp1：

使用sh main.sh命令进行编译运行

在main.sh中包含编译命令和执行命令，

编译：g++ main.cpp -lglut -lGLU -lGL -o main

运行：./main

exp2：编译和运行命令完全相同

**实验效果（效果展示+操作说明）：**

exp1：

立体显示三维模型实验：



读取名为bunny的obj文件，使用脚本后即可显示图像。可以看出实验效果和预想的效果相同。

exp2：

鼠标交互及法向计算：

（1）鼠标交互：

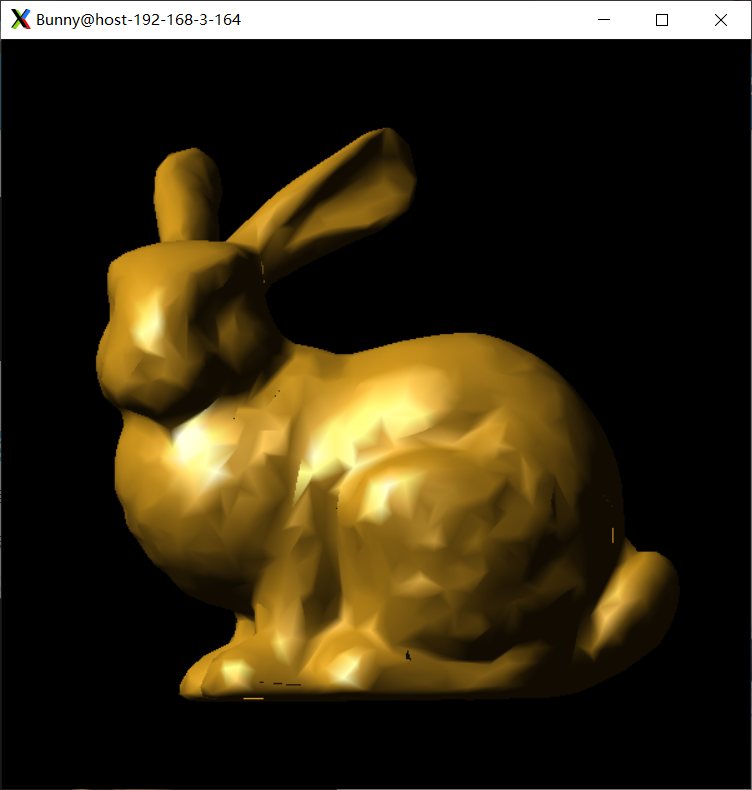
实现方法：

用一个times的浮点数记录放大或者缩小倍数。写一个mouseclick函数，当左键弹起时times增大，右键弹起时times减小。最后在显示图像之前使用

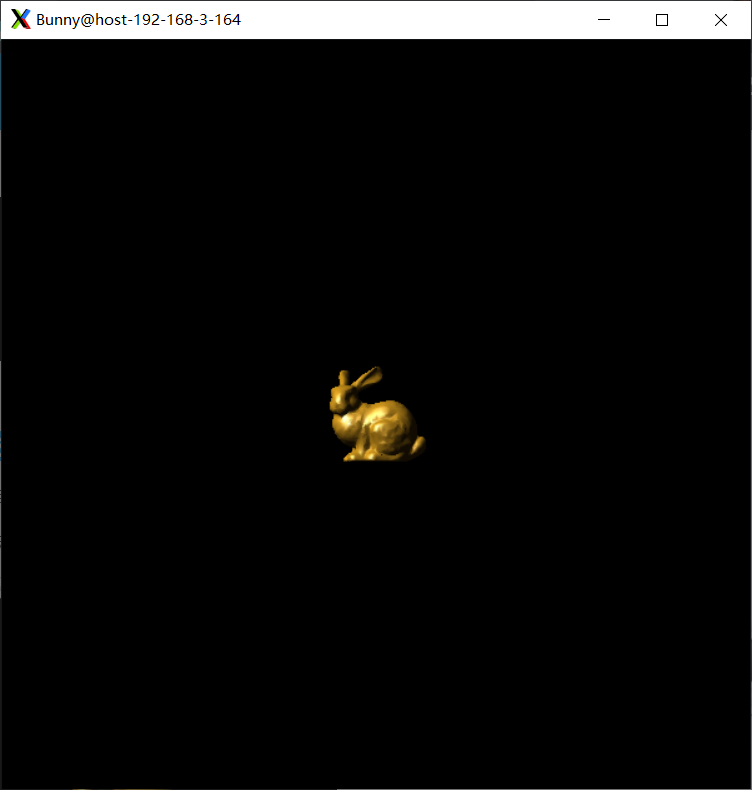
glScalef(times, times, times);

使得显示的图像放大或者缩小。

单击鼠标左键，会放大图像，效果如下图：



单击鼠标右键，可以缩小图像，效果如下图：



（2）键盘交互：

实现方法：

键盘按下x，y，z值时修改x\_rotate，y\_rotate和z\_rotate。最后在显示图像前使用

    glRotated(x\_rotate, 1.0f, 0.0f, 0.0f);

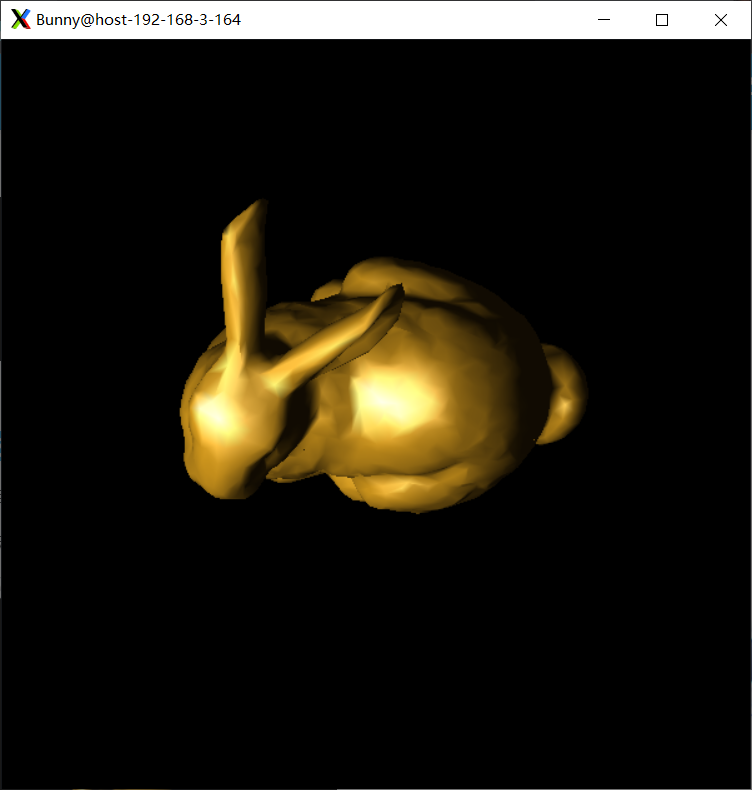
    glRotated(y\_rotate, 0.0f, 1.0f, 0.0f);

    glRotated(z\_rotate, 0.0f, 0.0f, 1.0f);

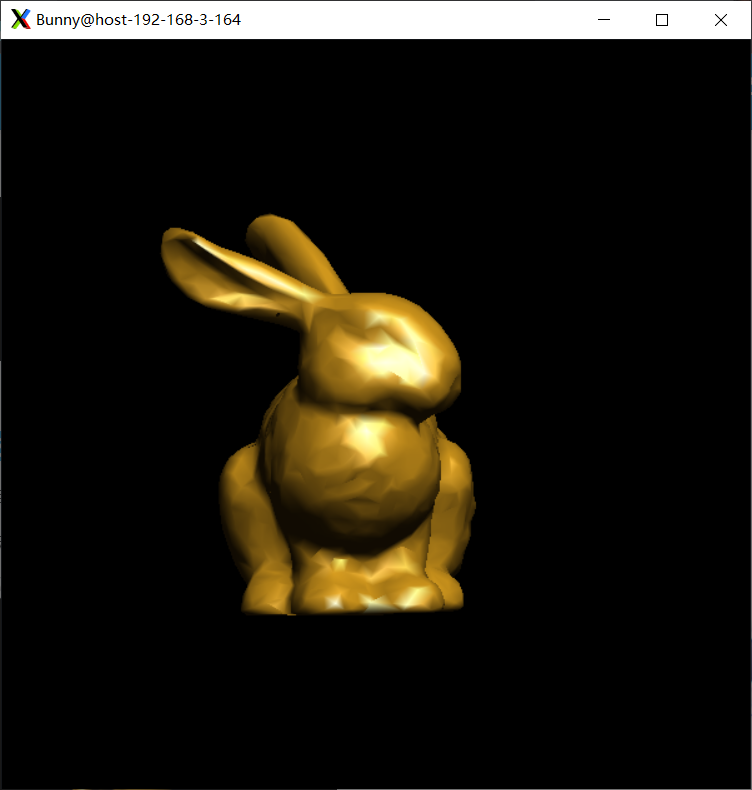
即可。

我们只允许bunny沿着x、y、z轴的正方向移动。

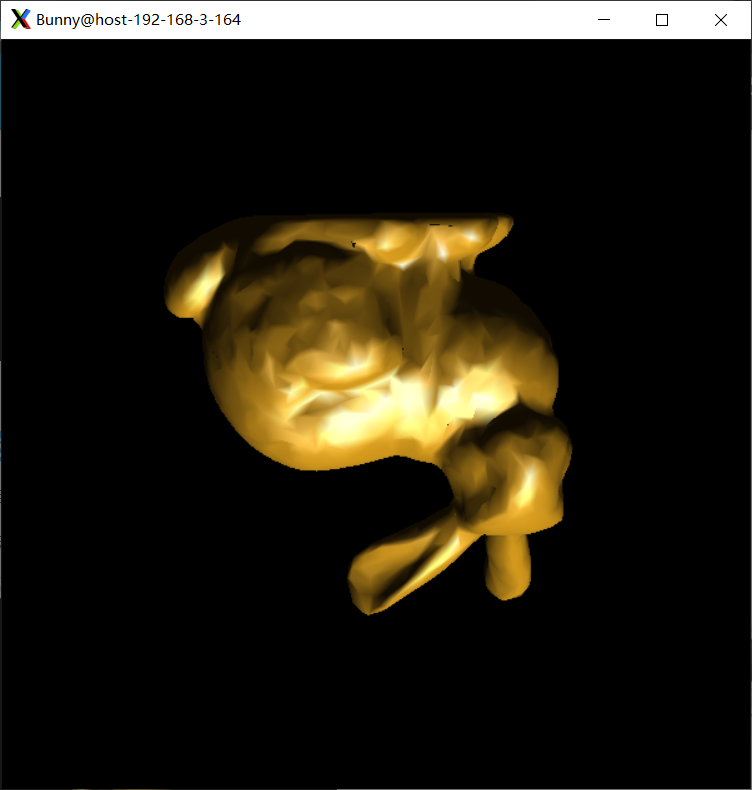
按住x键，bunny沿着x轴移动：



按住y键，bunny沿着y轴移动：



按住z键，bunny沿着z轴移动：



（3）自行计算法向：

实现方法：

通过顶点算出每个面的法向量，然后通过取平均算出顶点的法向量。

Exp2的实现中顶点的法向都是自行计算的。从上面的图像可以看出与直接读取法向效果差不多。