

**东北师范大学本科生课程作业**

**（**2020春季学期）

**课程名称：**  **基于图形引擎的软件开发**

**论文题目： 期末作业说明文档**

**任课老师： 封子军**

**学生姓名： 孔一言**

**学　　号： 2017012842**

**专 业： 软件工程**

**年 级： 2017级**

**学 院： 信息科学与技术学院**

**年 月 日： 2020年 7 月 14日**

目录

[一、 执行程序界面截图 3](#_Toc45707856)

[二、功能 3](#_Toc45707857)

[三、代码说明 4](#_Toc45707858)

[1、Draw3D()函数 4](#_Toc45707859)

[2、Draw3D1()函数 4](#_Toc45707860)

[3、Draw3D2()函数 6](#_Toc45707861)

[4、Draw3D2()函数 7](#_Toc45707862)

[5、power\_of\_two()和load\_texture()函数 7](#_Toc45707863)

[6、SetupRC()函数 7](#_Toc45707864)

[7、timerProc()函数 7](#_Toc45707865)

[8、commandBtn()函数 8](#_Toc45707866)

[9、快捷键 8](#_Toc45707867)

[四、bug解决 9](#_Toc45707868)

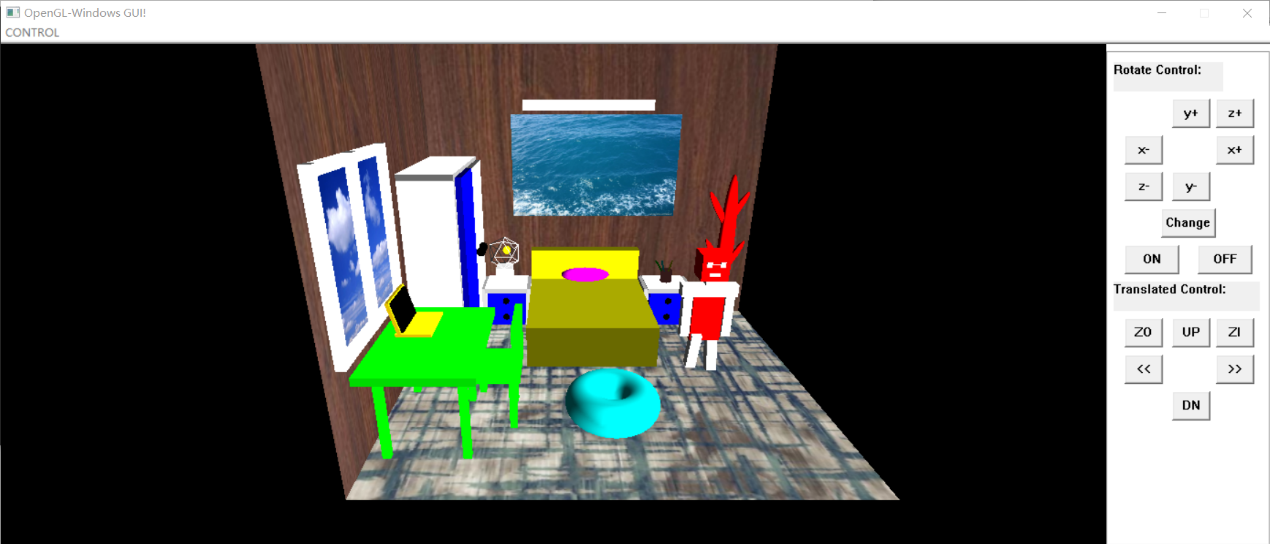
[1、黑屏问题 9](#_Toc45707869)

[2、视点变换 9](#_Toc45707870)

[3、物体颜色被纹理干扰 10](#_Toc45707871)

## 执行程序界面截图

1、卧室



2、书房



## 二、功能

1、可以切换房间（change按钮）

2、两个房间可分别绕x轴，y轴，z轴旋转

3、两个房间可分别绕x轴，y轴，z轴平移

4、两个房间的视点可分别向x轴，y轴，z轴平移

5、可控制两个房间整个场景旋转或是平移（卧室中的机器人除外）

6、可使卧室中的机器人行走或是停止

7、可控制卧室中的投影仪放下（ON按钮）或是收起（OFF按钮）

8、可切换背景颜色为：黑色、灰色、天蓝色、白色

## 三、代码说明

（由于代码太长，因此只在文档中作主要说明）

本项目是在老师之前项目的基础上进行修改的，我主要修改了Draw3D()函数，添加了Draw3D1()，Draw3D2()，Draw3D3()，power\_of\_two()，load\_texture()和SetupRC()三个函数。

### 1、Draw3D()函数

控制房间切换的函数。

### 2、Draw3D1()函数

（1）床

床由两个部分组成，一个床体和床靠背，均由立方体加glScalef()函数变形得到。

（2）床头柜

一共有两个床头柜。

每一个床头柜由两个抽屉，两个把手以及外包部分组成。每个抽屉又由四个变形立方体组成；外包部分是三个变形立方体包在两个抽屉的外部；两个把手是两个球体。

（3）衣柜

衣柜是由变形立方体拼接而成。和床头柜把手一样，衣柜把手也是两个黑色的球体。

（4）台灯

与大作业2的台灯类似，台灯由底座，灯罩和中间的灯球组成。

底座是一个变形的立方体，灯罩是一个线框二十面体。

黄色灯球部分我借鉴了老师给的移动光源的代码，给这个球设置的是聚光灯，角度设置的是30度。

这个灯光我测试过是有效果的，但是由于我设置是整个场景一起运动，因此这个灯球的照亮效果并不明显。

（5）香薰

香薰罐子及三根香薰棒均是由GL\_QUAD\_STRIP绘制的圆柱面。三根香薰棒经过各个方向的旋转，得到插在香薰罐子中的效果。

（6）投影仪

投影仪较为粗糙，是一个立方体经过变形得到的。幕布放下与收起的效果是通过一个状态判断来完成的。

（7）懒人沙发

我直接调用glutSolidTorus()函数绘制而成。

（8）梳妆台

桌子由五个部分组成，一个桌子面和四条桌子腿，均由立方体加glScalef()函数变形得到。

椅子与桌子类似。

（9）衣架

衣架由衣架体和衣架底座组成。

衣架体由五个变形球体组成；衣架底座是一个圆柱体，所用方法与香薰类似，在形成的圆柱面的顶部与底部加了两个圆形得到封闭的圆柱体。

（10）笔记本电脑

笔记本电脑是由两个金色的变形立方体和一个黑色的变形立方体组合而成。

（11）窗户

窗户是由一幅图画，再加上图画周围的窗户架组合而成。窗户架也均是变形立方体。

（12）地板，墙壁以及壁画

三者均由四边形贴图完成。

（13）灯光

在卧室中，我用到了三个灯光，两个定向灯光以及一个聚光灯。

（14）机器人

由于我不会画比较复杂的人物等，于是我便用多个立方体组合成了一个机器人。

### 3、Draw3D2()函数

（1）桌子和椅子

桌子由五个部分组成，一个桌子面和四条桌子腿，均由立方体加glScalef()函数变形得到。

椅子与桌子类似。

（2）茶壶与托盘

茶壶直接用glutSolidTeapot()函数生成。

托盘是由两个部分组成，底面和侧面组成。

我本来想用一个黑色的矮圆锥挡住另一个不是黑色的高圆锥，这样高圆锥未被遮挡的部分就是圆台形状，但是由于场景是3维的，这样子会对其他物体造成影响，我便放弃了这个想法，改用GL\_QUAD\_STRIP画出圆台的侧面，用GL\_POLYGON画出圆台的底面。

（3）台灯

桌子的最左边放着的就是台灯，台灯由底座，灯罩和中间的灯球组成。

底座就是一个变形的立方体，灯罩是一个线框二十面体，两者其实都是白色，但由于灯光原因，灯罩一部分显示出来了原本颜色，剩下部分和底座都为灰色。

黄色灯球部分我借鉴了老师给的移动光源的代码，给这个球设置的是聚光灯，角度设置的是30度。

这个灯光我测试过是有效果的，但是由于我设置是整个场景一起运动，因此这个灯球的照亮效果并不明显。

（4）衣架

衣架由衣架体和衣架底座组成。

衣架体由五个变形球体组成；衣架底座是一个圆柱体，所用方法与茶壶托盘类似。

（5）墙面，地板，门及墙上图画

以上均由四边形贴图完成。

（6）灯光

在书房中我一共设置了两个灯光，一个主灯光，另一个是台灯的灯光。主灯光是定向光源，台灯灯光是聚光灯。

### 4、Draw3D2()函数

这个函数绘制了卧室中投影仪的幕布，在点击按钮“ON”时被调用。

### 5、power\_of\_two()和load\_texture()函数

这两个函数都是从老师的代码中复制过来的，为了实现纹理贴图效果。

### 6、SetupRC()函数

这个函数实现加载纹理的功能。

### 7、timerProc()函数

这个函数用来实现机器人行走的功能。

（1）腿部

初始时，我便设置机器人的两条腿往不同方向各旋转15度，当这个程序每隔固定时间执行时，我便使机器人的两条腿旋转，每次旋转角度为30度。

（2）往前往后行走

当这个程序每隔固定时间执行时，我便使机器人沿z轴方向向前或向后平移，与机器人腿部旋转同时进行。

（3）机器人来回走动

首先，我给机器人设置了一个走动范围，当这个函数每次执行时，我先判断这个机器人所在的位置（z坐标），如果到了他能到的最大范围，则令他旋转180度，然后朝着之前反方向进行平移。在此过程中，我利用两个布尔值flag2和flag3来控制机器人往前或是往后行走。

### 8、commandBtn()函数

在这个函数中，我修改了几个按钮的功能，以及添加了三个按钮。

1、修改的按钮

我修改了“x+”，“x-”，“y+”，“y-”，“z+”，“z-”这六个按钮的功能。这六个按钮原本控制的是整个场景绕x，y，z轴旋转，但是我将他们的功能修改为控制场景的视点的坐标值的增加与减少。

2、添加的按钮

（1）“Change”按钮

这个按钮通过改变flag4的值来控制房间的切换。

（2）“ON”和“OFF”按钮

这两个按钮通过改变变量projector的值来控制投影仪幕布的放下与升起。

### 9、快捷键

1、Up：绕x轴旋转-30度

2、Down：绕x轴旋转30度

3、Left：绕y轴旋转-30度

4、Right：绕y轴旋转30度

5、Home：绕z轴旋转30度

6、End：绕z轴旋转-30度

7、A：向x轴负方向平移0.5

8、D：向x轴正方形平移0.5

9、W：向y轴正方向平移0.5

10、S：向y轴负方向平移0.5

11、Q：向z轴负方向平移0.5

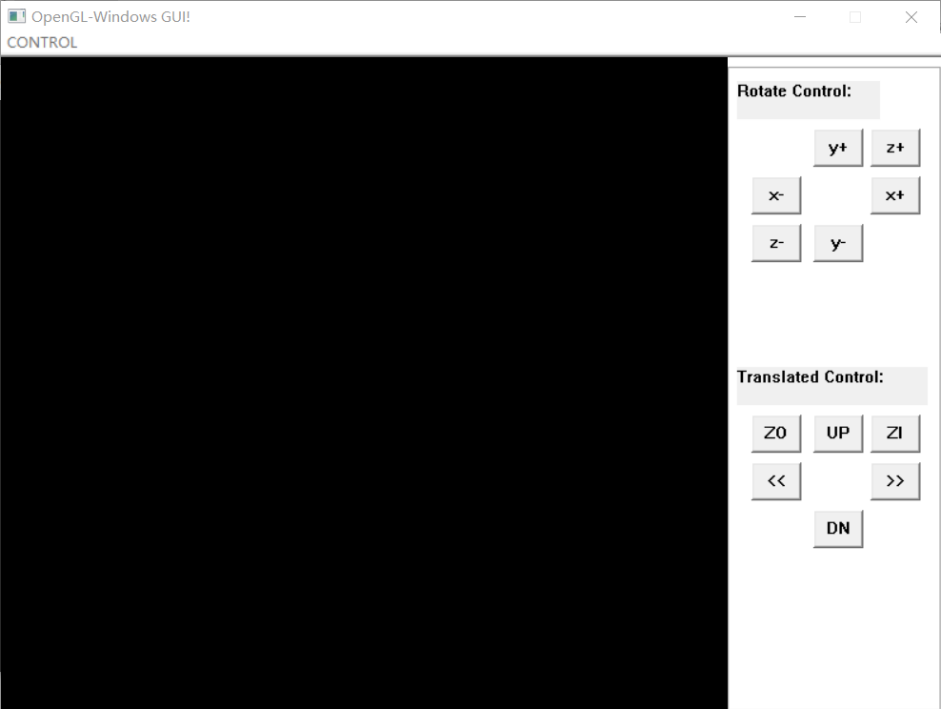
12、E：向z轴正方向平移0.5

13、Escape：关闭窗口

## 四、bug解决

### 1、黑屏问题

当我在处理灯光时，出现了一个bug。当我点击运行，出来的显示场景的页面是全黑的，如图所示：



当我对场景进行操作，比如让它旋转一下或是切换一下页面，场景又重新正常显示。

刚开始的解决办法是直接在WM\_PAINT中再给WM\_PAINT发送消息，即添加：SendMessage(hWnd, WM\_PAINT, 0, 0L)这行代码。

黑屏的问题解决了，但是又有了一个新的问题——卡顿。在控制台运行的时候对场景进行操作时会出现卡顿，但还是可以正常运行。但是当我运行Debug目录下的exe文件时，便一直显示未响应，即出现死机情况。

于是我又对WM\_PAINT进行修改，设置一个布尔值，只让程序在第一次运行的时候给WM\_PAINT发送两次消息，这样既解决了黑屏的问题，运行的时候也不会出现卡顿现象。

### 2、视点变换

由于以前写作业时并未用到gluLookAt()函数，因此我对这个函数十分陌生，因此在用这个函数的时候出现了很多问题。

在刚开始我在用这个函数的时候并未注意到这个函数要和glMatrixMode()函数和glLoadIdentity()函数共同使用。我刚开始以为只有gluLookAt()这个函数就可以了，当我运行的时候，发现我点击任一按钮，整个场景就会缩小并逐渐变远，后来我去看了PPT，才知道必须要加上glLoadIdentity()函数才可以，这个函数的作用就是将矩阵初始化为单位矩阵。在做第一次大作业时，我并不理解这个函数的意义，只是照葫芦画瓢，将老师的代码上有的都复制到我的代码中，经过这次，我才真正理解了。

但是在加上了必需的两个函数后，我的场景直接显示不出来了，我苦恼了好长时间，不断的去尝试视点的位置，但一直是黑屏。终于在有一次尝试中，我看到了我的场景的一个角，这下我明白之前一直黑屏只是由于我的参数设置不当，导致在我的场景不能显示在窗口中。

这下我的问题暂时解决了，但是又出现了一个新的问题，当我将y值设置成正数时，整个场景都是反的，我只能将y的初值设置为-0.2。但是当y值不断增加时，到了一定程度，整个场景就倒过来（即y值变为正数）了，我也一直没有办法解决这个问题。

之后我尝试着将照相机的方向修改，即将upz的值由1修改为-1这下当y值为正数的时候，整个场景不会再试倒着的了。但是y值变化范围很小的问题，依旧存在。

后来，我看到老师贴图小船的那个代码中也用到这个函数，而且y的初值设置的是7，跟我的-0.2相比，是较大的数了。我觉得奇怪，我便尝试了一下7，本以为我的场景会消失不见，没想到我的场景不仅没有消失，而且效果和我将y设置为小于1的数的效果差不多！我终于知道问题所在了，于是我便修改了参数，终于成功了！y值可取的范围变大， 再也不用担心场景会倒过来了。（我在录完视频后又修改了视点的参数，因此最终运行出来的效果与视频中所展示的效果不同）。

### 3、物体颜色被纹理干扰

当我将纹理运用到墙壁地板时，一些离得近的物体，比如桌子凳子之类的物体的颜色会受到纹理的干扰。

后来我在网上看到有关解法，只需将受干扰物体的代码前加上禁用纹理的代码即可，即加上glDisable(GL\_TEXTURE\_2D)，便不会再受到纹理的影响了。