

**计算机图形学作业报告**

|  |  |
| --- | --- |
| 作业名称： | 实现键盘功能 |
| 姓 名： | 郭炅 |
| 学 号： | 3170105370 |
| 电子邮箱： | [793881807@qq.com](mailto:793881807@qq.com) |
| 联系电话： | 18888923171 |
| 指导老师： | 张宏鑫 |

2019年 11 月 1 日

实验名字

1. 作业已实现的功能简述及运行简要说明

代码主要实现了PPT要求的键盘功能

* WASDZC：控制相机上下左右前后移动；
* P：切换投影方式（正投影与透视投影）；
* O：切换渲染方式（填充模式与线框模式）；
* 空格键 ：启动与暂停旋转（桌子与茶壶一起绕桌子中心轴旋转）；
* IKJL：控制茶壶前后左右移动
* E：控制茶壶开始/暂停旋转
* Q：退出

运行Visual Studio2017工程，按照相关键盘进行操作即可

1. 作业的开发与运行环境

* Visual Studio2017 Enterprise
* Glut

1. 系统或算法的基本思路、原理、及流程或步骤等

* 将绘制函数的参量设定为全局变量，相关的函数涉及缩放、平移等
* 定义键盘回调函数，根据相应按键修改相关全局变量，重绘
* 不断更新全局变量保存的旋转角度，OpenGL不断重绘显示含有新的角度的图形
  + 实现图形的旋转

1. 具体如何实现，包括关键（伪）代码、主要用到函数与算法等

* 定义键盘回调函数：
  + glutKeyboardFunc(key); 注册键盘回调函数
  + void key(unsigned char k, int x, int y); 定义键盘回调函数的具体操作
  + { case 's': {//todo

eye[0] = eye[0] - 0.5;

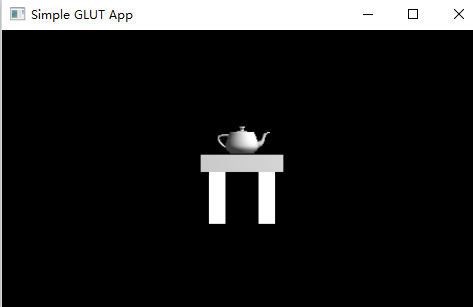
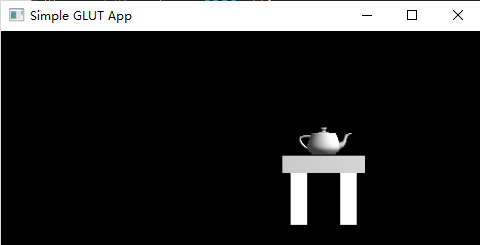
center[0] = center[0] -0.5;

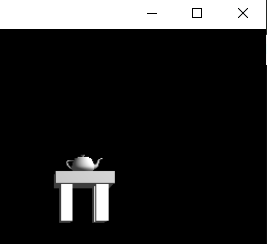
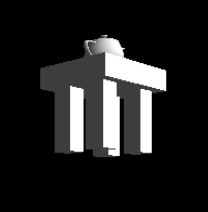
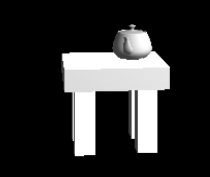
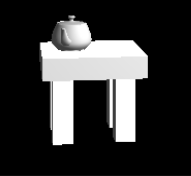
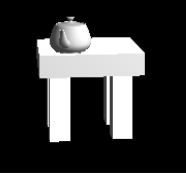
break;} 回调函数内部实现细节举例，此为按键 S 的操作

* 定义全局参量
  + 控制茶壶方位的全局变量：
    - float tTransX=0;
    - float tTransZ=0;
  + 控制视图形式、渲染模式的全局变量
    - bool bPersp = false;
    - bool bAnim = false;
    - bool bWire = false;
  + 控制摄像机和视点坐标的全局变量
    - float eye[] = {0, 0, 8};
    - float center[] = {0, 0, 0};
  + 控制茶壶是否旋转以及旋转度的参量
    - float teaRotate=0;
    - bool teaAnim = false;
* 根据全局变量参数更新视图投影函数
  + void updateView(int width, int height)
    - 内部设置是正投影或透视投影
* 根据全局参量刷新全局图形的填充方式、旋转角度
  + void redraw()
    - glPolygonMode(GL\_FRONT\_AND\_BACK, GL\_LINE);设置线框模式或填充模式
    - glRotatef(fRotate, 0, 1.0f, 0); 设置全局图形的旋转角度
    - if (bAnim) fRotate += 0.5f; 修改旋转角度
    - if (teaAnim) teaRotate += 0.5f;
* 图形绘制函数
  + void Draw\_Scene()
    - 绘制桌子
    - 根据全局参数绘制旋转后的茶壶
* 摄像机移动对应的参数变化
  + 相机的六维平行移动需要同时对应地修改视点位置，模拟实际情况下相机的移动
    - 同步修改全局变量 eye 和 center即可

1. 实验结果与分析

* 正投影下的移动：

* + 平行投影下，图形丢失立体属性，只维持二维图形
* 透视投影下的移动
  +  
    - 上下平移，图形维持立体属性，摄像机过于向上时，产生俯视效果，过于向下时，产生仰视效果
  +  
    - 左右平移，过左过右均可以拍摄得到相关的侧面特征
  +  
    - 前后移动，图形保持立体特征的同时变大缩小
* 线框模式
  + 
* 全局图形旋转
  + 
* 茶壶旋转
  + 
* 茶壶平移
  + 
  + 
  + 
  + 

1. 结论与心得体会

通过设置全局变量，调整相关绘制函数的参量，调用OpenGL中的特殊函数，实现了使用键盘控制图形变化并且进行实时刷新。

1. 参考文献

无