**光照模型的实现**

日期 2020/6/1

**一、实验目的**

1、掌握光照模型的三种光照强度表示；

2、掌握Phong光照模型；

3、熟练OpenGL中光照模型代码。

**二、实验要求**

1、利用光照模型完成太阳、地球和月亮的绘制；

2、注释给定代码中的光照模型。

**三、实验内容（代码及注释）**

#include "stdafx.h"

#include <windows.h>

#include <gl/glut.h>

void Initial()

{

    GLfloat light0\_diffuse[]={1.0f,0.5f,0.0f,1.0f};//light0中漫反射光分量

    GLfloat light0\_position[]={10.0f,20.0f,-150.0f,1.0f};//light0的坐标位置

    GLfloat light0\_direction[]={0.0f,0.0f,-1.0f};//light0的聚光灯方向角

    glLightfv(GL\_LIGHT0,GL\_DIFFUSE,light0\_diffuse);//light0在太阳中心

    glLightfv(GL\_LIGHT0,GL\_POSITION,light0\_position);

    glLightfv(GL\_LIGHT0,GL\_SPOT\_DIRECTION,light0\_direction);

    glEnable(GL\_DEPTH\_TEST);// 启用深度测试

    glEnable(GL\_LIGHTING);// 启用光源

    glEnable(GL\_LIGHT0);

    glEnable(GL\_COLOR\_MATERIAL);//启用颜色材质模式

    glFrontFace(GL\_CCW);// 指定逆时针绕法表示多边形正面

    glClearColor(0.75f, 0.75f,0.75f, 1.0f );//背景

}

void ChangeSize(int w, int h)

{

    if(h==0)h=1;

    glViewport(0,0,w,h);// 设置视区尺寸

    glMatrixMode(GL\_PROJECTION);//指定当前操作投影矩阵堆栈

    glLoadIdentity();//重置投影矩阵

    gluPerspective(35.0,(float)w/(float)h,1.0,500.0);//指定透视投影的观察空间

    glMatrixMode(GL\_MODELVIEW);

    glLoadIdentity();

}

void Display(void)

{

    static float angle1=0.0f,angle2=0.0f;

    //angle1地球绕太阳旋转的角度，angle2月亮绕地球旋转的角度

    GLfloat sun\_emission[]={0.6f,0.0f,0.0f,1.0f};//太阳颜色

    GLfloat earth\_emission[]={0.0f,0.0f,0.8f,1.0f};//地球颜色

    GLfloat moon\_emission[]={0.98f,0.625f,0.12f,1.0f};//月亮颜色

    GLfloat light1\_diffuse[]={0.5f,0.8f,0.0f,1.0f};//光源light1中漫反射光分量

    GLfloat light1\_position[]={50.0f,100.0f,100.0f,1.0f};//光源light1的坐标位置

    GLfloat light1\_direction[]={0.0f,0.0f,-1.0f};//光源light1的聚光灯方向角

    glClear(GL\_COLOR\_BUFFER\_BIT|GL\_DEPTH\_BUFFER\_BIT);//清除颜色和深度缓冲区

    glMatrixMode(GL\_MODELVIEW);//指定当前操作模型视图矩阵堆栈

    glLoadIdentity();//重置模型视图矩阵

    glTranslatef(0.0f,0.0f,-150.0f);//将图形沿z轴负向移动150.0f

    //绘制太阳

    glEnable(GL\_LIGHT1);

    glLightfv(GL\_LIGHT1,GL\_DIFFUSE,light1\_diffuse);

    glLightfv(GL\_LIGHT1,GL\_POSITION,light1\_position);

    glLightfv(GL\_LIGHT1,GL\_SPOT\_DIRECTION,light1\_direction);

    glMaterialfv(GL\_FRONT\_AND\_BACK,GL\_EMISSION,sun\_emission);

    glutSolidSphere(12.0f,30,30);//绘制太阳

    glDisable(GL\_LIGHT1);

    //绘制地球

    glPushMatrix();//保存当前的矩阵视图模型

    glRotatef(angle1,0.0f,10.0f,1.0f);//旋转一定角度

    glTranslatef(40.0f,0.0f,0.0f);//绕x轴正向移动40.0f

    glMaterialfv(GL\_FRONT\_AND\_BACK,GL\_EMISSION,earth\_emission);

    glutSolidSphere(6.0f,20,20);//绘制地球

    //绘制月亮

    glRotatef(angle2,0.0f,1.0f,0.0f);

    glTranslatef(15.0f,0.0f,0.0f);//绕x轴方向移动15.0f

    glMaterialfv(GL\_FRONT\_AND\_BACK,GL\_EMISSION,moon\_emission);

    glutSolidSphere(3.0f,20,20);//绘制月亮

    glPopMatrix();//恢复矩阵视图模型

    angle1+=2.0f;//增加旋转步长，产生动画效果

    if(angle1==360.0f)

        angle1=0.0f;

    angle2+=20.0f;

    if(angle2==360.0f)

        angle2=0.0f;

    glutSwapBuffers();

}

void TimerFunc(int x)

{

    glutPostRedisplay();

    glutTimerFunc(100,TimerFunc,1);

}

int main(int argc, char\* argv[])

{

    glutInit(&argc, argv);

    glutInitDisplayMode(GLUT\_DOUBLE | GLUT\_RGB | GLUT\_DEPTH);

    glutInitWindowSize(640,480);

    glutInitWindowPosition(400,200);

    glutCreateWindow("太阳地球月亮");

    glutReshapeFunc(ChangeSize);

    glutDisplayFunc(Display);

    glutTimerFunc(500,TimerFunc,1);

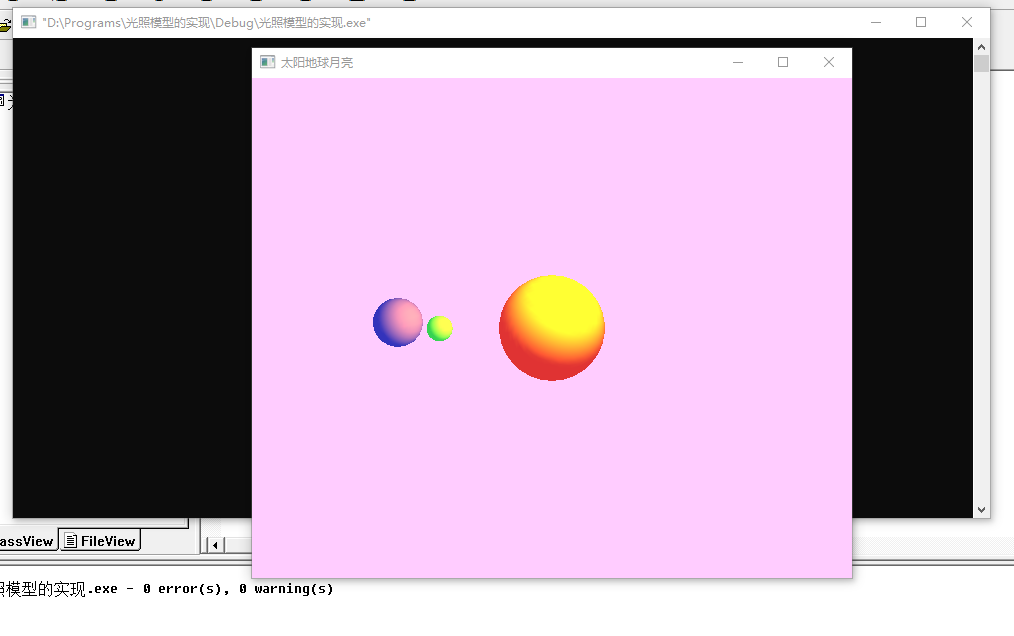
    Initial();

    glutMainLoop();

    return 0;

}

程序测试



1. **实验总结**
2. 进一步理解图形生成的原理。
3. 通过不断修改旋转变换的角度实现动画，其中angle1为地球绕太阳旋转的角度，angle2为月亮绕地球旋转的角度，月亮绕地球旋转的速度是地球绕太阳旋转速度的12倍。
4. 深度测试决定了是否绘制较远的像素点，通常选择较近的，而较远优先能实现透视的效果。调用函数glEnable(GL\_DEPTH\_TEST)。
5. 此次实验使用了2个光源，一个在太阳中心light0，照亮地球和月亮；另一个光源light1只照亮太阳。