

代码：

#include <GL/glut.h>

#define WIDTH 640

#define HEIGHT 480

static GLfloat angle = 30.0f;

void myDisplay(void)

{

glClear(GL\_COLOR\_BUFFER\_BIT | GL\_DEPTH\_BUFFER\_BIT);

// 创建透视效果视图

glMatrixMode(GL\_PROJECTION);//就是要对投影相关进行操作，也就是把物体投影到一个平面上，就像我们照相一样，把3维物体投到2维的平面上。这样，接下来的语句可以是跟透视相关的函数，比如glFrustum()或gluPerspective()；

glLoadIdentity();//重置当前指定的矩阵为单位矩阵。

gluPerspective(90.0f, 1.0f, 1.0f, 20.0f); // oid gluPerspective(GLdouble fovy, GLdouble aspect, GLdouble zNear, GLdouble zFar)near 和 far 决定了投影参考点与裁剪窗口的距离，fovy决定了裁剪窗口的高度，还有个参数 aspect 确定了裁剪窗口的长宽比例。

glMatrixMode(GL\_MODELVIEW);//这个是对模型视景的操作，接下来的语句描绘一个以模型为基础的适应，这样来设置参数，接下来用到的就是像gluLookAt()这样的函数；

glLoadIdentity();

gluLookAt(0.0, 5.0, -10.0, 0.0, 0.0, 0.0, 0.0, 1.0, 0.0);

{

GLfloat sun\_light\_position[] = { 1.0f, 1.0f, 0.0f, 0.0f };

GLfloat sun\_light\_ambient[] = { 0.0f, 0.0f, 0.0f, 1.0f }; //RGBA模式的环境光，为0

GLfloat sun\_light\_diffuse[] = { 1.0f, 1.0f, 1.0f, 1.0f }; //RGBA模式的漫反射光，全白光

GLfloat sun\_light\_specular[] = { 1.0f, 1.0f, 1.0f, 1.0f };

glLightfv(GL\_LIGHT0, GL\_POSITION, sun\_light\_position);//使用GL\_LIGHT0表示第0号光源，GL\_LIGHT1表示第1号光源，依次类推，OpenGL至少会支持8个光源，即GL\_LIGHT0到GL\_LIGHT7。使用glEnable函数可以开启它们。

glLightfv(GL\_LIGHT0, GL\_AMBIENT, sun\_light\_ambient);

glLightfv(GL\_LIGHT0, GL\_DIFFUSE, sun\_light\_diffuse);

glLightfv(GL\_LIGHT0, GL\_SPECULAR, sun\_light\_specular);

glEnable(GL\_LIGHT0);//开启第0号光源,使用glDisable函数则可以关闭光源

glEnable(GL\_LIGHTING);//要打开光照处理功能，使用这条的语句

glEnable(GL\_DEPTH\_TEST);

}

{

GLfloat sun\_mat\_ambient[] = { 0.0f, 0.0f, 0.0f, 1.0f };

GLfloat sun\_mat\_diffuse[] = { 0.0f, 0.0f, 0.0f, 1.0f };

GLfloat sun\_mat\_specular[] = { 0.0f, 0.0f, 0.0f, 0.0f };

GLfloat sun\_mat\_emission[] = { 0.5f, 0.0f, 0.0f, 1.0f };

GLfloat sun\_mat\_shininess = 0.0f;

glMaterialfv(GL\_FRONT, GL\_AMBIENT, sun\_mat\_ambient);

glMaterialfv(GL\_FRONT, GL\_DIFFUSE, sun\_mat\_diffuse);

glMaterialfv(GL\_FRONT, GL\_SPECULAR, sun\_mat\_specular);

glMaterialfv(GL\_FRONT, GL\_EMISSION, sun\_mat\_emission);//GL\_EMISSION属性。该属性由四个值组成，表示一种颜色。OpenGL认为该材质本身就微微的向外发射光线，以至于眼睛感觉到它有这样的颜色，但这光线又比较微弱，以至于不会影响到其它物体的颜色。

glMaterialf(GL\_FRONT, GL\_SHININESS, sun\_mat\_shininess);

}

// 定义地球的材质并绘制地球

{

GLfloat earth\_mat\_ambient[] = { 0.0f, 0.0f, 0.5f, 1.0f };

GLfloat earth\_mat\_diffuse[] = { 0.0f, 0.0f, 0.5f, 1.0f };

GLfloat earth\_mat\_specular[] = { 0.0f, 0.0f, 1.0f, 1.0f };

GLfloat earth\_mat\_emission[] = { 0.0f, 0.0f, 0.0f, 1.0f };

GLfloat earth\_mat\_shininess = 50.0f;

glMaterialfv(GL\_FRONT, GL\_AMBIENT, earth\_mat\_ambient);

glMaterialfv(GL\_FRONT, GL\_DIFFUSE, earth\_mat\_diffuse);

glMaterialfv(GL\_FRONT, GL\_SPECULAR, earth\_mat\_specular);

glMaterialfv(GL\_FRONT, GL\_EMISSION, earth\_mat\_emission);

glMaterialf(GL\_FRONT, GL\_SHININESS, earth\_mat\_shininess);

glTranslatef(3.0f, 0.0f, 0.0f);

glutSolidSphere(4.0, 50, 32);

}

glutSwapBuffers();

}

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

{

glutInit(&argc, argv);

glutInitDisplayMode(GLUT\_RGBA | GLUT\_DOUBLE);

glutInitWindowPosition(100, 100);

glutInitWindowSize(WIDTH, HEIGHT);

glutCreateWindow("gouraud光照模型");

glutDisplayFunc(&myDisplay);

glutMainLoop();

return 0;

}