

```

/*
  Name: Luz Movil
  Copyright: Ninguno
  Author: Ma Carmen Villar
  Date: 09/10/04 22:56
  Description: Este hace una luz, y la mueve ademas ejemplifica el uso de la
  funcion de callback Idle
*/

#include<GL/glut.h>

#define ROT_INC 0.5

/*Variables globales para los materiales, las opciones y la luz*/
float light_ambient [] = {0.0, 0.2, 0.5, 1.0};
float light_diffuse_specular [] = {0.8, 0.8, 0.8, 1.0};
float light_pos [] = {0.0, 0.0, 2.0, 1.0};
float spot_dir [] = {0.0, 0.0, -1.0};
float spot_cutoff = 15.0;
float spot_exponent = 1.0;

float mat_ambient_diffuse [] = {0.0, 0.0, 1.0, 1.0};
float mat_specular [] = {0.7, 0.1, 0.2, 1.0};
float mat_shininess = 1.0;

float focus_ambient_diffuse [] = {0.6, 0.1, 0.5, 1.0};
float focus_emission [] = {0.8, 0.8, 0.8, 1.0};
float rot_angle_x = 0.0;
float rot_angle_y = 0.0;

/*Declaracion de prototipos de funcion*/
void initgl (void);
void idle (void);
void display (void);

int main (int argc, char **argv)
{
    glutInit(&argc, argv);
    glutInitDisplayMode(GLUT_DOUBLE | GLUT_RGBA | GLUT_DEPTH);
    glutInitWindowPosition(20, 20);
    glutInitWindowSize(350, 350);
    glutCreateWindow("Luz Movil");
    initgl();
    /*registro de funciones de callback*/
    glutDisplayFunc(display);
    glutIdleFunc(idle);
    glutMainLoop();

    return 0;
}

//===== initgl =====

void initgl (void)
{
    glEnable(GL_DEPTH_TEST);
    glClearColor(0.0, 0.0, 0.0, 0.0);

    glEnable(GL_LIGHTING);
    glLightfv(GL_LIGHT0, GL_AMBIENT, light_ambient);
    glLightfv(GL_LIGHT0, GL_DIFFUSE, light_diffuse_specular);
    glLightfv(GL_LIGHT0, GL_SPECULAR, light_diffuse_specular);
    glLightf(GL_LIGHT0, GL_SPOT_CUTOFF, spot_cutoff);
    glLightf(GL_LIGHT0, GL_SPOT_EXPONENT, spot_exponent);
    glEnable(GL_LIGHT0);

    glMatrixMode(GL_PROJECTION);
    glLoadIdentity();
    gluPerspective(60.0, 1.0, 1.0, 100.0);
    glMatrixMode(GL_MODELVIEW);
    glTranslatef(0.0, 0.0, -5.0);
}

```

```

//===== idle =====
void idle (void)
{
    rot_angle_y = (rot_angle_y > 360 ? 0 : rot_angle_y + ROT_INC);
    rot_angle_x = (rot_angle_x > 360 ? 0 : rot_angle_x + ROT_INC / (2 * 3.141516));
    glutPostRedisplay();
}

//===== display =====
void display (void)
{
    glClear(GL_COLOR_BUFFER_BIT | GL_DEPTH_BUFFER_BIT);

    glPushMatrix();
        glRotatef(rot_angle_y, 0.0, 1.0, 0.0);
        glRotatef(rot_angle_x, 1.0, 0.0, 0.0);
        glTranslatef(light_pos[0], light_pos[1], light_pos[2]);

        glLightfv(GL_LIGHT0, GL_POSITION, light_pos);
        glLightfv(GL_LIGHT0, GL_SPOT_DIRECTION, spot_dir);

        glMaterialfv(GL_FRONT, GL_AMBIENT_AND_DIFFUSE, focus_ambient_diffuse);
        glMaterialfv(GL_FRONT, GL_EMISSION, focus_emission);
        glutSolidCone(0.2, 0.5, 10, 10);
    glPopMatrix();

    glMaterialfv(GL_FRONT, GL_AMBIENT_AND_DIFFUSE, mat_ambient_diffuse);
    glMaterialfv(GL_FRONT, GL_SPECULAR, mat_specular);
    glMaterialf(GL_FRONT, GL_SHININESS, mat_shininess);
    glutSolidSphere(1.25, 35, 35);

    glutSwapBuffers();
}

```