```
void initScene() {
…
   initLight();
}


void initLight()
{
       // enable color tracking

   //glLightfv(GL_LIGHT0, GL_POSITION, light_position);

  glLightModelfv(GL_LIGHT_MODEL_AMBIENT, ambientLight);
  glLightModeli(GL_LIGHT_MODEL_COLOR_CONTROL, GL_SEPARATE_SPECULAR_COLOR);
  glLightfv(GL_LIGHT0, GL_DIFFUSE, lightColor);
  glLightfv(GL_LIGHT0, GL_AMBIENT, ambientLight);
  glLightfv(GL_LIGHT0, GL_DIFFUSE, diffuseLight);
  glLightfv(GL_LIGHT0, GL_SPECULAR, specularLight);
  glLightf(GL_LIGHT0, GL_CONSTANT_ATTENUATION, 0.1);
  glLightf(GL_LIGHT0, GL_LINEAR_ATTENUATION, 0.05);
  glEnable(GL_LIGHT0);

  glLightfv(GL_LIGHT1, GL_DIFFUSE, lightColor);
  glLightfv(GL_LIGHT1, GL_AMBIENT, ambientLight);
  glLightfv(GL_LIGHT1, GL_DIFFUSE, diffuseLight);
  glLightfv(GL_LIGHT1, GL_SPECULAR, specularLight);
  glLightf(GL_LIGHT1, GL_CONSTANT_ATTENUATION, 0.1);
  glLightf(GL_LIGHT1, GL_LINEAR_ATTENUATION, 0.05);
  glEnable(GL_LIGHT1);
  glEnable(GL_LIGHTING);


}

void motion(int x, int y)
{
       ..
       if (lightMoving) {
              lightAngle += (x - lightStartX)/40.0;
              lightHeight += (lightStartY - y)/20.0;
              lightStartX = x;
              lightStartY = y;
              glutPostRedisplay();
  }
}

void mouse(int btn, int state, int x, int y)
{
       if (state == GLUT_DOWN)
       {
              if (btn == GLUT_LEFT_BUTTON) {
                     mx = x; my = y;
                     drag = 1;
                     }
              if (btn == GLUT_RIGHT_BUTTON) {
             //spin = (spin + 30) % 360;
                     lightMoving = 1;
      lightStartX = x;
```

```
      lightStartY = y;


        }

       }
       if (state == GLUT_UP)
       {
              drag = 0;
              if (btn == GLUT_RIGHT_BUTTON) {
                lightMoving = 0;
              }


       }
}


void keyboard(unsigned char key, int x, int y)
{

              case 'l':
                     glEnable(GL_LIGHTING);
                     glEnable(GL_LIGHT0);
                     lightSwitch = !lightSwitch;
                     break;
              case 'L':
                     glDisable(GL_LIGHTING);
                     glDisable(GL_LIGHT0);
                     lightSwitch = !lightSwitch;
                     break;

       }
}


void renderScene(void) {

       glClear(GL_COLOR_BUFFER_BIT | GL_DEPTH_BUFFER_BIT);

       glLoadIdentity();
       cx = radius * cos(angle1)*cos(angle2);
       cy = radius * sin(angle1)*cos(angle2);
       cz = radius * sin(angle2);
       gluLookAt(cx, cy, cz, lx,ly,lz, 0.0f,0.0f,1.0f);

       /* Reposition the light source. */
  lpos[0] = 12*cos(lightAngle);
  lpos[1] = lightHeight;
  lpos[2] = 12*sin(lightAngle);
  if (directionalLight) {
    lpos[3] = 0.0;
  } else {
    lpos[3] = 1.0;
  }


       glLightfv(GL_LIGHT0, GL_POSITION, lpos);
```

```
        glLightfv(GL_LIGHT1, GL_POSITION, lpos2);

        // Coordinate system
        glBegin(GL_LINES);
```

```
GLfloat ambientLight[] = { 0.2f, 0.2f, 0.2f, 1.0f };
GLfloat diffuseLight[] = { 0.8f, 0.8f, 0.8, 1.0f };
GLfloat specularLight[] = { 0.5f, 0.5f, 0.5f, 1.0f };

static GLfloat lightColor[] = {0.8, 1.0, 0.8, 1.0}; /* green-tinted */

float lpos[4] = {1,0.5,1,0};
float lpos2[4] = {-1,-0.5,-1,0};

static int directionalLight = 1;

static float lightAngle = 0.0, lightHeight = 20;

int lightMoving = 0, lightStartX, lightStartY;

static GLboolean lightSwitch = GL_TRUE;
```