```
#include<windows.h>
#include<GL/glut.h>
#include <time.h>
double w=1280,h=720;
double view[3]={2,2,12.9};
double look[3]={2,2,2};
int flag=-1;
void steps(void);
void window(void);
void sgate(void);
void gate(void);
double angle=0,speed=5,maino=0,romo=0,tro=0,mgo=0,sgo=0;
//declarating quadric objects
GLUquadricObj *Cylinder;
GLUquadricObj *Disk;
struct tm *newtime;
time_t ltime;
GLfloat angle1;
//initialisation
void myinit(void)
{
glMatrixMode(GL_PROJECTION);
glLoadIdentity();
glFrustum(-1.0,1.0,-1*w/h,1*w/h,1,200.0);
glMatrixMode(GL_MODELVIEW);
glLoadIdentity();
//defining new quadric object
 Cylinder = gluNewQuadric();
//to set drawing style
 gluQuadricDrawStyle( Cylinder, GLU_FILL);
//to set automatic normals
 gluQuadricNormals( Cylinder,GLU_SMOOTH);
 Disk = gluNewQuadric();
 gluQuadricDrawStyle( Disk, GLU_FILL);
 gluQuadricNormals( Disk, GLU_SMOOTH);
GLfloat gam[]={0.2,.2,.2,1};
glLightModelfv(GL_LIGHT_MODEL_AMBIENT,gam);
}
//set material property
void matprop(GLfloat amb[],GLfloat dif[],GLfloat spec[],GLfloat shi[])
{
glMaterialfv(GL_FRONT_AND_BACK,GL_AMBIENT,amb);
glMaterialfv(GL_FRONT_AND_BACK,GL_DIFFUSE,dif);
glMaterialfv(GL_FRONT_AND_BACK,GL_SPECULAR,spec);
glMaterialfv(GL_FRONT_AND_BACK,GL_SHININESS,shi);
}
//to create wall
void wall(double thickness)
{
glPushMatrix();
glTranslated(2,.5*thickness,2);
glScaled(4.0,thickness,4.0);
glutSolidCube(1.0);
glPopMatrix();
}
//to create compound wall
void wall2(double thickness)
{
glPushMatrix();
glTranslated(.8,.5*thickness*4,3.5);
glScaled(1.6,thickness*4,7.0);
glutSolidCube(1.0);
glPopMatrix();
}
//to create earth
void earth(void)
```

```
{
GLfloat ambient[]={1,0,0,1};
GLfloat specular[]={0,1,1,1};
GLfloat diffuse[]={.5,.5,.5,1};
GLfloat shininess[]={50};
matprop(ambient,diffuse,specular,shininess);
GLfloat lightIntensity[]={.7,.7,.7,1};
GLfloat light_position[]={2,5,-3,0};
glLightfv(GL_LIGHT0,GL_POSITION,light_position);
glLightfv(GL_LIGHT0,GL_DIFFUSE,lightIntensity);
glPushMatrix();
glTranslated(0,-.25,0);
glScaled(10000,.5,1000000);
glutSolidCube(1.0);
glPopMatrix();
glFlush();
}
void compound(void)
{
GLfloat ambient[]={1,0,0,1};
GLfloat specular[]={0,1,1,1};
GLfloat diffuse[]={.7,1,.7,1};
GLfloat shininess[]={50};
matprop(ambient,diffuse,specular,shininess);
GLfloat lightIntensity[]={.7,.7,.7,1};
GLfloat light_position[]={2,6,1.5,0};
glLightfv(GL_LIGHT0,GL_POSITION,light_position);
glLightfv(GL_LIGHT0,GL_DIFFUSE,lightIntensity);
//left wall of compound
glPushMatrix();
glPushMatrix();
glTranslated(-4,0,-1-.04);
glRotated(90.0,0,0,1);
wall2(0.08);
glPopMatrix();
//right wall of compound
glPushMatrix();
glTranslated(8,0,-1-.02);
glRotated(90.0,0,0,1);
wall2(0.08);
glPopMatrix();
//back wall of compound
glPushMatrix();
glTranslated(2,.8,-1);
glRotated(-90,1,0,0);
glScaled(12,.02*4,1.6);
glutSolidCube(1.0);
glPopMatrix();
//front left wall of compound
glPushMatrix();
glTranslated(-3,.8,6-.08);
glRotated(-90,1,0,0);
glScaled(2,.02*4,1.6);
glutSolidCube(1.0);
glPopMatrix();
//front middle wall of compound
glPushMatrix();
glTranslated(2.5,.8,6-.08);
glRotated(-90,1,0,0);
glScaled(6,.02*4,1.6);
glutSolidCube(1.0);
glPopMatrix();
//front right wall of compound
glPushMatrix();
glTranslated(7,.8,6-.08);
glRotated(-90,1,0,0);
```

```
glScaled(2,.02*4,1.6);
glutSolidCube(1.0);
glPopMatrix();
glPopMatrix();
GLfloat ambient2[]={0,1,0,1};
GLfloat specular2[]={1,1,1,1};
GLfloat diffuse2[]={.2,.6,0.1,1};
GLfloat shininess2[]={50};
matprop(ambient2,diffuse2,specular2,shininess2);
//floor
glPushMatrix();
glTranslated(-4,-0.05,-1);
glScaled(3,3,1.7);
wall(0.08);
glPopMatrix();
gate();
 sgate();
 glFlush();
}
void room()
{
GLfloat ambient1[]={1,0,1,1};
GLfloat specular1[]={1,1,1,1};
GLfloat diffuse1[]={0.5,0.5,0.5,1};
GLfloat mat_shininess[]={50};
 matprop(ambient1,diffuse1,specular1,mat_shininess);
glPushMatrix();
glTranslated(.5,4,.5);
//roof
glPushMatrix();
glTranslated(-.02*4,.7*3.9,-.02*4);
glScaled(.6+.02,1.5,.5+.02+.1);
wall(0.08);
glPopMatrix();
GLfloat ambient2[]={1,0,0,1};
GLfloat specular2[]={1,1,1,1};
GLfloat diffuse2[]={1,1,.7,1};
GLfloat shininess1[]={50};
matprop(ambient2,diffuse2,specular2,shininess1);
//left wall
glPushMatrix();
glTranslated(0,0,-.02);
glScaled(1,.7,.5);
glRotated(90.0,0,0,1);
wall(0.08);
glPopMatrix();
//right wall
glPushMatrix();
glTranslated(2.4,0,-.02);
glScaled(1,.7,.5);
glRotated(90.0,0,0,1);
wall(0.08);
glPopMatrix();
//back wall
glPushMatrix();
glTranslated(-.08,0,0);
glScaled(.62,.7,1);
glRotated(-90.0,1,0,0);
wall(0.08);
glPopMatrix();
//front wall
glPushMatrix();
glTranslated(-0.08,0,2);
glScaled(.5,.7,1);
glRotated(-90.0,1,0,0);
wall(0.08);
```

```
glPopMatrix();
//wall above the room door
glPushMatrix();
glTranslated(1.9,.7*3,2);
glScaled(.11,.7*.25,1);
glRotated(-90.0,1,0,0);
wall(0.08);
glPopMatrix();
GLfloat ambient[]={1,0.5,.5,1};
GLfloat specular[]={1,1,1,1};
GLfloat diffuse[]={1,0.5,0.5,1};
 matprop(ambient,diffuse,specular,mat_shininess);
//door
glPushMatrix();
glTranslated(2.3,0,(2-.05));
glRotated(-tro,0,1,0);
glTranslated(-2.3,0,-(2-.05));
glPushMatrix();
glTranslated(1.927,0,2);
glScaled(.09,.525,1);
glRotated(-90.0,1,0,0);
wall(0.02);
glPopMatrix();
glPushMatrix();
glTranslated(2.3,0,2-.05);
glScaled(.6,.7,.8);
glRotated(-90,1,0,0);
gluCylinder(Cylinder, 0.05, 0.05, 3, 16, 16);
glPopMatrix();
glPopMatrix();
glPopMatrix();
}
void tankwall(float thk)
{
glTranslated(.5,.5*thk,.5);
glScaled(1,thk,1);
glutSolidCube(1);
}
void watertank(void)
{
float thk=.04,hght=1,wdth=1,bdth=1;
GLfloat ambient1[]={.5,0,1,1};
GLfloat specular1[]={1,1,1,1};
GLfloat diffuse1[]={.5,.8,1,1};
GLfloat mat_shininess[]={50};
 matprop(ambient1,diffuse1,specular1,mat_shininess);
glPushMatrix();
glTranslated(1.5,4+4*.7,1.5);
glScaled(.8,.8,.8);
//tank floor
glPushMatrix();
glScaled(wdth,1,bdth);
tankwall(thk);
glPopMatrix();
//tank left wall
glPushMatrix();
glScaled(1,hght,bdth);
glRotated(90.0,0,0,1);
tankwall(thk);
glPopMatrix();
//tank right wall
glPushMatrix();
glTranslated(wdth,0,0);
glScaled(1,hght,bdth);
glRotated(90.0,0,0,1);
tankwall(thk);
```

```
glPopMatrix();
//tank back wall
glPushMatrix();
glScaled(wdth,hght,1);
glRotated(-90.0,1,0,0);
tankwall(0.04);
glPopMatrix();
//tank front wall
glPushMatrix();
glTranslated(0,0,bdth);
glScaled(wdth,hght,1);
glRotated(-90.0,1,0,0);
tankwall(0.04);
glPopMatrix();
//tank roof
glPushMatrix();
glTranslated(-thk,hght,0);
glScaled(wdth*.8,1,bdth);
tankwall(0.04);
glPopMatrix();
glPushMatrix();
glTranslated(wdth*.8-thk,hght,0);
glScaled(wdth*.2+thk,1,bdth*.6);
tankwall(0.04);
glPopMatrix();
glPopMatrix();
}
void terece(void)
{
GLfloat ambient1[]={1,0,1,1};
GLfloat specular1[]={1,1,1,1};
GLfloat diffuse1[]={0.5,0.5,0.5,1};
GLfloat mat_shininess[]={50};
 matprop(ambient1,diffuse1,specular1,mat_shininess);
glPushMatrix();
glTranslated(0,4,0);
glScaled(1,.1,1);
//left wall
glPushMatrix();
glTranslated(0,0,-.02-.25);
glScaled(1,1,.95);
glRotated(90.0,0,0,1);
wall(0.08);
glPopMatrix();
//right wall
glPushMatrix();
glTranslated(6+.12,0,-.02-.27);
glScaled(1,1,1.1);
glRotated(90.0,0,0,1);
wall(0.08);
glPopMatrix();
//back wall
glPushMatrix();
glTranslated(-.08,0,-.21);
glScaled(1.5+.05,1,1);
glRotated(-90.0,1,0,0);
wall(0.08);
glPopMatrix();
//front wall
glPushMatrix();
glTranslated(-.08,0,4+.11);
glScaled(1.5+.05,1,1);
glRotated(-90.0,1,0,0);
wall(0.08);
glPopMatrix();
glPushMatrix();
```

```
glTranslated(-.04,2,4);
glScaled(.08,4,.1);
glutSolidCube(1);
glPopMatrix();
 glPopMatrix();
}
void fanwing(float winglen)
{
 glPushMatrix();
glRotated(90,1,0,0);
glRotated(90,0,1,0);
glScaled(1,15,1);
gluCylinder(Cylinder,.01,.01,1,4,1);
glPopMatrix();
}
void fanbottom()
{
glPushMatrix();
glTranslated(1,3.2,1);
glPushMatrix();
glRotated(90,1,0,0);
gluCylinder(Cylinder, .2, .2,.05, 128, 16);
glPopMatrix();
glPushMatrix();
glTranslated(0,0.00025,0);
glRotated(90,1,0,0);
gluDisk(Disk,0,.2,32,16);
glPopMatrix();
glPushMatrix();
glTranslated(0,-.05,0);
glRotated(90,1,0,0);
gluCylinder(Cylinder,.2,.15,.1,128,16);
glPopMatrix();
glPushMatrix();
glTranslated(0,-.1,0);
glRotated(90,1,0,0);
gluDisk(Disk,0,.15,32,16);
glPopMatrix();
glPushMatrix();
glTranslated(0.1,0.0,0);
fanwing(.6);
glPopMatrix();
glPushMatrix();
glRotated(120,0,1,0);
glTranslated(.1,0,0);
fanwing(.6);
glPopMatrix();
glPushMatrix();
glRotated(240,0,1,0);
glTranslated(0.1,0.0,0);
fanwing(.6);
glPopMatrix();
glPopMatrix();
}
void fan(void)
{
    glPushMatrix();
glTranslated(2.5,1.9,0);
glScaled(.5,.5,.5);
GLfloat mat_ambient[]={.5,0,0,1};
GLfloat mat_specular[]={0,1,1,0};
GLfloat mat_diffuse[]={.8,1,.8,1};
GLfloat mat_shininess[]={50};
glMaterialfv(GL_FRONT,GL_AMBIENT,mat_ambient);
glMaterialfv(GL_FRONT,GL_DIFFUSE,mat_diffuse);
glMaterialfv(GL_FRONT,GL_SPECULAR,mat_specular);
glMaterialfv(GL_FRONT,GL_SHININESS,mat_shininess);
if(flag==-1)
{
glPushMatrix();
fanbottom();
glPopMatrix();
}
else
{
angle+=speed;
glPushMatrix();
glTranslated(1,0,1);
glRotated(angle,0,1,0);
glTranslated(-1,0,-1);
fanbottom();
glPopMatrix();
}
glPushMatrix();
glTranslatef(1,3.3,1);
glRotated(-90,1,0,0);
gluCylinder(Cylinder, .1, 0.005, .25, 16, 16);
glPopMatrix();
glPushMatrix();
glTranslatef(1,4,1);
glRotated(90,1,0,0);
gluCylinder(Cylinder, .006, 0.006, .6, 16, 16);
glPopMatrix();
glPushMatrix();
glTranslatef(1,3.96,1);
glRotated(90,1,0,0);
gluCylinder(Cylinder, .1, 0.005, .25, 16, 16);
glPopMatrix();
glPopMatrix();
if(flag==1)
glutPostRedisplay();
}
void tableg(float llen,float lthk)
{
glPushMatrix();
glRotated(-90,1,0,0);
gluCylinder(Cylinder,lthk,lthk,llen,32,32);
glPopMatrix();
}
void table(float tabwid,float tablen,float tabthk,float llen,float lthk)
{
glPushMatrix();
glPushMatrix();
glTranslated(0,llen,0);
glScaled(tabwid,tabthk,tablen);
glutSolidCube(1);
glPopMatrix();
float dist1=.95*tablen/2-lthk/2;
float dist2=.95*tabwid/2-lthk/2;
// front right leg
glPushMatrix();
glTranslated(dist2,0,dist1);
tableg(llen,lthk);
glPopMatrix();
//back right leg
glPushMatrix();
glTranslated(dist2,0,-dist1);
tableg(llen,lthk);
glPopMatrix();
//back left leg
glPushMatrix();
glTranslated(-dist2,0,-dist1);
```

```
tableg(llen,lthk);
glPopMatrix();
//front left leg
glPushMatrix();
glTranslated(-dist2,0,dist1);
tableg(llen,lthk);
glPopMatrix();
glPopMatrix();
}
void cot(float cwid,float clen,float cthk,float llen,float lthk)
{
GLfloat ambient1[]={1,0,.4,1};
GLfloat specular1[]={1,1,1,1};
GLfloat diffuse1[]={0.5,0.5,0.5,1};
GLfloat mat_shininess[]={50};
 matprop(ambient1,diffuse1,specular1,mat_shininess);
glPushMatrix();
glTranslated(5.6,0,.5);
table(cwid,clen,cthk,llen,lthk);
glPushMatrix();
glTranslated(0,llen,clen/2);
GLdouble eqn[3] = {0.0,1.0, 0.0};
glPushMatrix();
glClipPlane(GL_CLIP_PLANE0, eqn);
//void glClipPlane(GLenum plane,const GLdouble *equation);
glEnable (GL_CLIP_PLANE0);//enable clip plane
gluDisk(Disk,0,cwid/2,32,32);
glPopMatrix();
glDisable(GL_CLIP_PLANE0);
glPopMatrix();
glPushMatrix();
glTranslated(0,llen,-clen/2);
glPushMatrix();
glClipPlane (GL_CLIP_PLANE0, eqn);
glEnable (GL_CLIP_PLANE0);
glScaled(1,1.5,1);
gluDisk(Disk,0,cwid/2,32,32);
glPopMatrix();
glDisable(GL_CLIP_PLANE0);
glPopMatrix();
glPopMatrix();
}
void cleg(float cllen,float clwid)
{
glRotated(90,1,0,0);
gluCylinder(Cylinder,clwid,clwid,cllen,32,32);
}
void chair(float cblen,float cbwid,float cbthk,float cllen,float clwid)
{
GLfloat ambient1[]={.5,1,.5,1};
GLfloat specular1[]={1,1,1,1};
GLfloat diffuse1[]={0.5,0.5,0.5,1};
GLfloat mat_shininess[]={50};
 matprop(ambient1,diffuse1,specular1,mat_shininess);
glPushMatrix();
glTranslated(0,cllen,0);
//chair base
glPushMatrix();
glScaled(cblen,cbthk,cbwid);
glutSolidCube(1);
glPopMatrix();
float dist=cblen/2-clwid/2;
//chair legs
glPushMatrix();
glTranslated(dist,0,dist);
cleg(cllen,clwid);
```

```
glPopMatrix();
glPushMatrix();
glTranslated(-dist,0,dist);
cleg(cllen,clwid);
glPopMatrix();
glPushMatrix();
glTranslated(-dist,0,-dist);
cleg(cllen,clwid);
glPopMatrix();
glPushMatrix();
glTranslated(dist,0,-dist);
cleg(cllen,clwid);
glPopMatrix();
//base pipes
glPushMatrix();
glTranslated(-.085,-clwid/2,cbwid/3);
 glRotated(90,0,1,0);
gluCylinder(Cylinder,-clwid,clwid,cblen,32,32);
 glPopMatrix();
glPushMatrix();
glTranslated(-.085,clwid/2,-cbwid/3);
 glRotated(90,0,1,0);
gluCylinder(Cylinder,clwid,clwid,cblen,32,32);
 glPopMatrix();
//back support pipes
glPushMatrix();
glTranslated(-.085,-clwid/2,cbwid/3);
 glRotated(-90,1,0,0);
gluCylinder(Cylinder,clwid,clwid,cllen,32,32);
 glPopMatrix();
glPushMatrix();
glTranslated(-.085,-clwid/2,-cbwid/3);
 glRotated(-90,1,0,0);
gluCylinder(Cylinder,clwid,clwid,cllen,32,32);
 glPopMatrix();
//back support
glPushMatrix();
glTranslated(-cblen/2,cllen/2+cblen/2,0);
glRotated(90,0,0,1);
glScaled(cblen,.01,cbwid);
glutSolidCube(1);
glPopMatrix();
glPopMatrix();
}
void diningtable()
{
glPushMatrix();
glTranslated(3,0,1);
glScaled(1.5,1.5,1.5);
table(.3,.5,.025,.4,.005);
//front left chair
glPushMatrix();
glTranslated(-.1,0,.1);
chair(.15,.15,.02,.3,.005);
glPopMatrix();
//back left chair
glPushMatrix();
glTranslated(-.1,0,-.1);
chair(.15,.15,.02,.3,.005);
glPopMatrix();
//front right chair
glPushMatrix();
glTranslated(.1,0,.1);
glRotated(180,0,1,0);
chair(.15,.15,.02,.3,.005);
glPopMatrix();
```

```
//back right chair
glPushMatrix();
glTranslated(.1,0,-.1);
glRotated(180,0,1,0);
chair(.15,.15,.02,.3,.005);
glPopMatrix();
//back chair
glPushMatrix();
glTranslated(0,0,-.27);
glRotated(-90,0,1,0);
chair(.15,.15,.02,.3,.005);
glPopMatrix();
//front chair
glPushMatrix();
glTranslated(0,0,.27);
glRotated(90,0,1,0);
chair(.15,.15,.02,.3,.005);
glPopMatrix();
glPopMatrix();
}
void steps(void)
{
 int i;
GLfloat ambient1[]={1,0,1,1};
GLfloat specular1[]={1,1,1,1};
GLfloat diffuse1[]={0.5,0.5,0.5,1};
GLfloat mat_shininess[]={50};
 matprop(ambient1,diffuse1,specular1,mat_shininess);
glPushMatrix();
glTranslated(-.25,.1,.2);
for(i=0;i<19;i++)
{
glPushMatrix();
glTranslated(0,i*.2,i*.2);
glScaled(.4,.2,.3);
glutSolidCube(1);
glPopMatrix();
}
glPopMatrix();
glPushMatrix();
glRotated(-45,1,0,0);
glTranslated(-.45,.3,2.7);
glScaled(.04,1,5.4);
glutSolidCube(1);
glPopMatrix();
glPushMatrix();
glTranslated(-.45,4,3.6);
glScaled(.04,.8,.75);
glutSolidCube(1);
glPopMatrix();
glPushMatrix();
glTranslated(-.25,4,3.96);
glScaled(.4,.8,.04);
glutSolidCube(1);
glPopMatrix();
}
void sleg(float len,float thk)
{
glScaled(thk,len,thk);
glutSolidCube(1);
}
void solar(void)
{
GLfloat ambient1[]={.1,.1,.1,1};
GLfloat specular1[]={1,1,1,1};
GLfloat diffuse1[]={1,1,1,1};
GLfloat mat_shininess[]={50};
 matprop(ambient1,diffuse1,specular1,mat_shininess);
GLfloat lightIntensity[]={.7,.7,.7,1};
GLfloat light_position[]={-20,4,60,0};
glLightfv(GL_LIGHT2,GL_POSITION,light_position);
glLightfv(GL_LIGHT2,GL_DIFFUSE,lightIntensity);
glEnable(GL_LIGHT2);
//base
glPushMatrix();
glTranslated(4,4,3);
glPushMatrix();
glTranslated(0.4,.4,0);
glScaled(1,.8,1);
glutSolidCube(1);
glPopMatrix();
GLfloat     ambient2[]={.7,.7,.7,1};
GLfloat specular2[]={1,1,1,1};
GLfloat diffuse2[]={1,1,1,1};
 matprop(ambient2,diffuse2,specular2,mat_shininess);
glPushMatrix();
glTranslated(0,.8,0);
glPushMatrix();
glTranslated(.6,.6,0);
gluCylinder(Cylinder,.1,.1,.4,32,32);
glPopMatrix();
GLfloat ambient3[]={1,0,.2,1};
GLfloat specular3[]={1,1,1,1};
GLfloat diffuse3[]={1,0,.5,1};
GLfloat mat_shininess3[]={50};
 matprop(ambient3,diffuse3,specular3,mat_shininess3);
glPushMatrix();
glTranslated(.6,.6,0);
gluDisk(Disk,0,.1,32,32);
glPopMatrix();
glPushMatrix();
glTranslated(.6,.6,0.4);
gluDisk(Disk,0,.1,32,32);
glPopMatrix();
GLfloat ambient4[]={0,0,0,1};
GLfloat specular4[]={1,1,1,1};
GLfloat diffuse4[]={0,0,0,1};
GLfloat mat_shininess4[]={50};
 matprop(ambient4,diffuse4,specular4,mat_shininess4);
glPushMatrix();
glTranslated(.5,.3,.05);
sleg(.6,.01);
glPopMatrix();
glPushMatrix();
glTranslated(.7,.3,.05);
sleg(.6,.01);
glPopMatrix();
glPushMatrix();
glTranslated(.5,.3,.35);
sleg(.6,.01);
glPopMatrix();
glPushMatrix();
glTranslated(.7,.3,.35);
sleg(.6,.01);
glPopMatrix();
glPushMatrix();
glRotated(45,0,0,1);
glTranslated(.3,.015,.2);
glScaled(.6,.03,.4);
glutSolidCube(1);
glPopMatrix();
glPushMatrix();
```

```
glTranslated(.4,.21,0);
sleg(.425,.01);
glPopMatrix();
glPushMatrix();
glTranslated(.4,.21,.4);
sleg(.425,.01);
glPopMatrix();
glPushMatrix();
glTranslated(.4,.4,0);
glRotated(30,0,0,1);
glRotated(90,0,1,0);
gluCylinder(Cylinder,.01,.01,.2,32,32);
glPopMatrix();
glPopMatrix();
glPopMatrix();
}

void myclock()
{
GLfloat mat_ambient[]={.4,.8,.4,1};
GLfloat mat_specular[]={1,1,1,1};
GLfloat mat_diffuse[]={0.4,.8,.4,1};
GLfloat mat_shininess[]={50};
matprop(mat_ambient,mat_diffuse,mat_specular,mat_shininess);
int hour_ticks , sec_ticks;
glPushMatrix();
glTranslated(2,3.2,-.02);
glScaled(.03,.06,.03);
glPushMatrix(); // Draw clock face

glTranslatef( 0, 0, 1.0);
gluDisk(Disk, 0, 7, 32, 16);
glPopMatrix();
GLfloat mat_ambien[]={1,0,0,1};
matprop(mat_ambien,mat_diffuse,mat_specular,mat_shininess);

glPushMatrix();
glTranslatef( 0, 0, 1.95);
gluDisk(Disk, 0, .8, 32, 16);
glPopMatrix();
GLfloat ambient[]={0,0,0,1};
GLfloat specular[]={1,1,1,1};
GLfloat diffuse[]={0,0,0,1};
 matprop(ambient,diffuse,specular,mat_shininess);
// Draw hour hand
glPushMatrix();
glColor3f(1.0, 0.5, 0.5);
glTranslatef( 0, 0, 1.5);
glRotatef( -(360/12) * (newtime->tm_hour+newtime->tm_min/60.0), 0.0,0.0, 1.0);
glRotatef( -90, 1.0, 0.0, 0.0);
gluCylinder(Cylinder, 0.45, 0, 4, 16, 16);
glPopMatrix();
GLfloat     ambient1[]={0,0,1,1};
GLfloat specular1[]={1,1,1,1};
GLfloat diffuse1[]={0,0,1,1};
 matprop(ambient1,diffuse1,specular1,mat_shininess);
// Draw minute hand
glPushMatrix();
glColor3f(1.0, 0.5, 1.0);
glTranslatef( 0, 0, 1.25);
glRotatef( -(360/60) * newtime->tm_min, 0.0, 0.0, 1.0);
glRotatef(-90, 1.0, 0.0, 0.0);
gluCylinder(Cylinder, 0.4, 0, 6, 16, 16);
glPopMatrix();
GLfloat     ambient2[]={1,0,0,1};
GLfloat specular2[]={1,1,1,1};
GLfloat diffuse2[]={1,0,0,1};
 matprop(ambient2,diffuse2,specular2,mat_shininess);
// Draw second hand
glPushMatrix();
glTranslatef( 0, 0, 1);
glRotatef(-(360/60) * newtime->tm_sec, 0.0, 0.0, 1.0);
glRotatef( -90, 1.0, 0.0, 0.0);
gluCylinder(Cylinder, 0.3, 0, 6, 16, 16);
glPopMatrix();
GLfloat     ambient3[]={1,1,1,1};
GLfloat specular3[]={1,1,1,1};
GLfloat diffuse3[]={1,0,1,1};

 matprop(ambient3,diffuse3,specular3,mat_shininess);
for(hour_ticks = 0; hour_ticks < 12; hour_ticks++)
 {
glPushMatrix();// Draw next arm axis.
 glTranslatef(0.0, 0.0, 1);
glRotatef( (360/12) * hour_ticks, 0.0, 0.0, 1.0);
glTranslatef( 6.0, 0.0, 0.0);
glutSolidCube(.8);
glPopMatrix();
}
for(sec_ticks = 0; sec_ticks < 60; sec_ticks++)
{
 glPushMatrix();
glTranslatef(0.0, 0.0, 1.1);
glRotatef( (360/60) * sec_ticks, 0.0, 0.0, 1.0);
glTranslatef(6.0, 0.0, 0.0);
glutSolidCube(0.25);
glPopMatrix();
}
 glPopMatrix();
}
void window(void)
{
int i;
GLfloat lightIntensity[]={.7,.7,.7,1};
GLfloat light_position[]={-20,4,-60,0};
glLightfv(GL_LIGHT1,GL_POSITION,light_position);
glLightfv(GL_LIGHT1,GL_DIFFUSE,lightIntensity);
glEnable(GL_LIGHT1);
glPushMatrix();
glTranslated(3.185,1,3.95);
//left edge of window
glPushMatrix();
glTranslated(.02,1,.02);
glScaled(.04,2.2,.04);
glutSolidCube(1);
glPopMatrix();
//right edge
glPushMatrix();
glTranslated(1.6+.02,1,0.02);
glScaled(.04,2.2,.04);
glutSolidCube(1);
glPopMatrix();
//top edge
glPushMatrix();
glTranslated(.9,2+.02,0.02);
glScaled(1.8,.04,.04);
glutSolidCube(1);
glPopMatrix();
//bottom edge
glPushMatrix();
glTranslated(.8,.02,0.02);
glScaled(1.8,.04,.04);
```

```
glutSolidCube(1);
glPopMatrix();
for(i=1;i<=3;i++)
{
glPushMatrix();
glTranslated(.4*i,0,0);
glRotated(-90,1,0,0);
gluCylinder(Cylinder,.01,.01,2,32,32);
glPopMatrix();
}
for(i=1;i<=3;i++)
{
glPushMatrix();
glTranslated(.1+.4*i,0,0);
glRotated(-90,1,0,0);
gluCylinder(Cylinder,.01,.01,2,32,32);
glPopMatrix();
}
for(i=1;i<=4;i++)
{
glPushMatrix();
glTranslated(0,.4*i,0);
glRotated(90,0,1,0);
gluCylinder(Cylinder,.03,.03,1.6,32,32);
glPopMatrix();
}
glPopMatrix();
}
void gate(void)
{
int i;
GLfloat ambient1[]={1,.5,1,1};
GLfloat specular1[]={1,1,1,1};
GLfloat diffuse1[]={.5,.5,.5,1};
GLfloat mat_shininess[]={120};
 matprop(ambient1,diffuse1,specular1,mat_shininess);
glPushMatrix();
//if flag mgo=1 the open the main gate
if(mgo==1)
glTranslated(1.5,0,0);
glTranslated(-1.3,0,6);
//top frame of the main gate
glPushMatrix();
glTranslated(0,1.5,0);
glScaled(1.7,.04,.04);
glutSolidCube(1);
glPopMatrix();
//bottom frame of main gate
glPushMatrix();
glTranslated(0,.05,0);
glScaled(1.7,.04,.04);
glutSolidCube(1);
glPopMatrix();
//left frame of the main gate
glPushMatrix();
glTranslated(-.8,.75,0);
glScaled(.04,1.5,.04);
glutSolidCube(1);
glPopMatrix();
//right frame of the main gate
glPushMatrix();
glTranslated(.8,.75,0);
glScaled(.04,1.5,.04);
glutSolidCube(1);
glPopMatrix();
//horizantal pipes of the main gate
```

```
for(i=1;i<=3;i++)
{
glPushMatrix();
glTranslated(-.85,.4*i,0);
glRotated(90,0,1,0);
gluCylinder(Cylinder,.02,.02,1.7,32,32);
glPopMatrix();
}
//vertical strips of the main gate
for(i=1;i<=5;i++)
{
glPushMatrix();
glTranslated(-.9+.3*i,.75,0);
glScaled(.2,1.5,.02);
glutSolidCube(1);
glPopMatrix();
}
glPopMatrix();
}
void sgate(void )
{
int i;
GLfloat ambient1[]={1,.5,1,1};
GLfloat specular1[]={1,1,1,1};
GLfloat diffuse1[]={.5,.5,.5,1};
GLfloat mat_shininess[]={120};
 matprop(ambient1,diffuse1,specular1,mat_shininess);
glPushMatrix();
//to open the sub gate
glTranslated(5.75-.25,.05,6);
glRotated(sgo,0,1,0);
glTranslated(-5.75+.25,-.05,-6);
//to translate sub gate to required position
glTranslated(5.75,.05,6);
//top edge of the sub gate
glPushMatrix();
glTranslated(0,1.5,0);
glScaled(.5,.08,.08);
glutSolidCube(1);
glPopMatrix();
//bottom edge of the sub gate
glPushMatrix();
glTranslated(0,.05,0);
glScaled(.5,.08,.08);
glutSolidCube(1);
glPopMatrix();
//left edge of the sub gate
glPushMatrix();
glTranslated(-.25,.85,0);
glScaled(.04,1.7,.04);
glutSolidCube(1);
glPopMatrix();
//right edge of the sub gate
glPushMatrix();
glTranslated(.25,.8,0);
glScaled(.04,1.6,.04);
glutSolidCube(1);
glPopMatrix();
//vertical pipes of the sub gate
for(i=1;i<=4;i++)
{
glPushMatrix();
glTranslated(-.25+.1*i,0,0);
glRotated(-90,1,0,0);
gluCylinder(Cylinder,.01,.01,1.5,32,32);
glPopMatrix();
}
//horizantal pipes of the sub gate
for( i=1;i<=4;i++)
{
glPushMatrix();
glTranslated(-.25,.05+.3*i,0);
glRotated(90,0,1,0);
gluCylinder(Cylinder,.02,.02,.5,32,32);
glPopMatrix();
}
glPopMatrix();
}

void house(void)
{
GLfloat mat_ambient[]={1,0,0,1};
GLfloat mat_specular[]={1,1,1,1};
GLfloat mat_diffuse[]={1,1,.7,1};
GLfloat mat_shininess[]={50};
matprop(mat_ambient,mat_diffuse,mat_specular,mat_shininess);
GLfloat lightIntensity4[]={.7,.7,.7,.7};
GLfloat light_position4[]={3,1,.5,1};
glLightfv(GL_LIGHT6,GL_POSITION,light_position4);
glLightfv(GL_LIGHT6,GL_DIFFUSE,lightIntensity4);
glEnable(GL_LIGHT6);
glPushMatrix();
glTranslated(0,.15,0);

//roof
 glPushMatrix();
glTranslated(-.02*4,3.9,-.01*4-.25);
glScaled(1.5+.05,1.5,1.1);
wall(0.08);
glPopMatrix();
GLfloat ambient2[]={1,0,0,1};
GLfloat specular2[]={1,1,1,1};
GLfloat diffuse2[]={.7,1,0.8,1};
GLfloat shininess[]={50};
matprop(ambient2,diffuse2,specular2,shininess);

//floor
glPushMatrix();
glTranslated(-.02*3,-0.05,-.01*4);
glScaled(1.5+.01,1.5,1);
wall(0.08);
glPopMatrix();

GLfloat ambient1[]={1,0,0,1};
GLfloat specular1[]={1,1,1,1};
GLfloat diffuse1[]={1,1,.7,1};
GLfloat shininess1[]={50};
matprop(ambient1,diffuse1,specular1,shininess1);
//left wall
glPushMatrix();
glRotated(90.0,0,0,1);
wall(0.08);
glPopMatrix();
//right wall
glPushMatrix();
glTranslated(6,0,0);
glRotated(90.0,0,0,1);
wall(0.08);
glPopMatrix();
//back wall
glPushMatrix();
glTranslated(-.08,0,0);
```

```
glScaled(1.5+.02,1,1);
glRotated(-90.0,1,0,0);
wall(0.08);
glPopMatrix();
//room vertical wall
glPushMatrix();
glTranslated(4,0,0);
glScaled(1,1,.5);
glRotated(90.0,0,0,1);
wall(0.08);
glPopMatrix();
//room horizantal wall
glPushMatrix();
glTranslated(4.4,0,2);
glScaled(.4,1,1);
glRotated(-90.0,1,0,0);
wall(0.08);
glPopMatrix();
//wall above the room door
glPushMatrix();
glTranslated(4,3,2);
glScaled(.11,.25,1);
glRotated(-90.0,1,0,0);
wall(0.08);
glPopMatrix();
//left room horizantal wall
glPushMatrix();
glTranslated(0,0,2);
glScaled(.4,1,1);
glRotated(-90.0,1,0,0);
wall(0.08);
glPopMatrix();
//lroom vertical wall
glPushMatrix();
glTranslated(1.6,0,0);
glScaled(1,1,.35);
glRotated(90.0,0,0,1);
wall(0.08);
glPopMatrix();
//entrance room right wall
glPushMatrix();
glTranslated(1.6,0,2.59);
glScaled(1,1,.35);
glRotated(90.0,0,0,1);
wall(0.08);
glPopMatrix();
//wall above main door
glPushMatrix();
glTranslated(-0.02,3,4);
glScaled(.13,.27,1);
glRotated(-90.0,1,0,0);
wall(0.08);
glPopMatrix();
//wall right to the main door
glPushMatrix();
glTranslated(.48,0,4);
glScaled(.68,1,1);
glRotated(-90.0,1,0,0);
wall(0.08);
glPopMatrix();
//wall right to the window
glPushMatrix();
glTranslated(4.8,0,4);
glScaled(.3,1,1);
glRotated(-90.0,1,0,0);
wall(0.08);
```

```
glPopMatrix();
//wall below the window
glPushMatrix();
glTranslated(3.2,0,4);
glScaled(.4,.25,1);
glRotated(-90.0,1,0,0);
wall(0.08);
glPopMatrix();
//wall above the window
glPushMatrix();
glTranslated(3.2,3.03,4);
glScaled(.4,.25,1);
glRotated(-90.0,1,0,0);
wall(0.08);
glPopMatrix();
room();
watertank();
terece();
steps();
window();
fan();
 cot(.6,.9,.06,.35,.009);
 diningtable();
myclock();
solar();
GLfloat ambient[]={1,0.5,.5,1};
GLfloat specular[]={1,1,1,1};
GLfloat diffuse[]={1,.5,.5,1};
 matprop(ambient,diffuse,specular,mat_shininess);
//main door
glPushMatrix();
glTranslated(0,0,4);
glRotated(maino,0,1,0);
glTranslated(0,0,-4);
glPushMatrix();
glTranslated(0,0,4);
glScaled(.12,.75,1);
glRotated(-90.0,1,0,0);
wall(0.04);
glPopMatrix();
glPushMatrix();
glTranslated(0,0,4);
glScaled(.5,1,.2);
glRotated(-90,1,0,0);
gluCylinder(Cylinder, 0.05, 0.05, 3, 16, 16);
glPopMatrix();
glPopMatrix();
//bolow room door
glPushMatrix();
glTranslated(4,0,(2-.025));
glRotated(romo,0,1,0);
glTranslated(-4,0,-(2-.025));
glPushMatrix();
glTranslated(4,0,2);
glScaled(.099,.75,1);
glRotated(-90.0,1,0,0);
wall(0.01);
glPopMatrix();
glPushMatrix();

glTranslated(4.01,0,2-.025);
glScaled(.5,1,.6);
glRotated(-90,1,0,0);
gluCylinder(Cylinder, 0.05, 0.05, 3, 16, 16);
glPopMatrix();
glPopMatrix();
```

```
 glPopMatrix();
glFlush();
}
void display(void)
{
time(&ltime); // Get time
newtime = localtime(&ltime); // Convert to local time
glMatrixMode(GL_MODELVIEW);
glLoadIdentity();
glClear(GL_COLOR_BUFFER_BIT|GL_DEPTH_BUFFER_BIT);
gluLookAt(view[0],view[1],view[2],look[0],look[1],look[2],0.0,1.0,0.0);
earth();
compound();
house();
glFlush();
glutSwapBuffers();
glutPostRedisplay();
}
void Keyboard(unsigned char key,int x,int y)
{
switch(key)
{
//to move the camera along -ve x axis
case '4':
view[0]-=.1;
glutPostRedisplay();
break;
//to move the camera along +ve x axis
case '6':
view[0]+=.1;
glutPostRedisplay();
break;
//to move the camera along +ve y axis
case '7':
view[1]+=.1;
glutPostRedisplay();
break;
//to move the camera along -ve y axis
case '1':
if(view[1]>1.9)
view[1]-=.1;
glutPostRedisplay();
break;
//to move the camera along -ve z axis
case '8':
    view[2]-=.1;
    glutPostRedisplay();
break;
//to move the camera along +ve z axis
case '2':
    view[2]+=.1;
    glutPostRedisplay();
break;
//to run and stop the fan
case 'S':
case 's':
flag*=-1;
glutPostRedisplay();
break;
//to move the look position along +ve x axis
case 'r':
case 'R':
look[0]+=.1;
break;
//to move the look position along -ve x axis
case 'l':
```

```
case 'L':
look[0]-=.1;
break;
//to move the look position along +ve y axis
case 'U':
case 'u':
look[1]+=.1;
break;
//to move the look position along -ve y axis
case 'D':
case 'd':
look[1]-=.1;
break;
//to move the look position along +ve z axis
case 'f':
case 'F':
look[2]+=.1;
break;
//to move the look position along -ve z axis
case 'B':
case 'b':
look[2]-=.1;
break;
//to open and close the main door
case 'q':
case 'Q':
if(maino==0)
 maino=85;
else
maino=0;
break;
//to open and close the below room door
case 'O':
case 'o':
if(romo==0)
 romo=75;
else
romo=0;
break;
//to open and close the above room door
case 'p':
case 'P':
if(tro==0)
 tro=70;
else
tro=0;
break;
//to open and close the main gate
case 'g':
case 'G':
if(mgo==0)
 mgo=1;
else
mgo=0;
break;
//to open and close the sub gate
case 'h':
case 'H':
if(sgo==0)
 sgo=50;
else
sgo=0;
break;
//inside view
case 'i':
case 'I':
```

```
view[0]=2.8;
view[1]=2;
view[2]=4.8;
look[0]=2.8;
look[1]=2;
look[2]=1;
break;
//top view
case 'T':
case 't':
view[0]=6;
view[1]=12;
view[2]=10;
look[0]=2;
look[1]=4;
look[2]=2;
break;
//front view
case 'j':
case 'J':
view[0]=2;
view[1]=2;
view[2]=12.9;
look[0]=3;
look[1]=2;
look[2]=3;
break;
//back view
case 'k':
case 'K':
    view[0]=1;
    view[1]=6;
    view[2]=-7;
    look[0]=2;
    look[1]=4;
    look[2]=2;
break;
}
}
void mySpecialKeyFunc( int key, int x, int y )
{
switch ( key ) {
case GLUT_KEY_UP:
if ( speed < 25.0) {
speed+=5;
}
break;
case GLUT_KEY_DOWN:
if (speed>0) {
speed-=5;
}
break;
}
}
void main_menu(int m)
{
switch(m)
{
case 1:
exit(0);
}
}
void fan_menu(int m)
{
switch(m)
{
    case 1:
    flag*=-1;
    glutPostRedisplay();
break;
case 2:
    if ( speed < 30.0)
        {
            speed+=5;
        }
break;
case 3:
  if (speed>0)
    {
     speed-=5;
    }
break;
}
}
void door_menu(int m)
{
switch(m)
{
case 1:
    if(maino==0)
    maino=85;
    else
    maino=0;
break;
case 2:
    if(romo==0)
    romo=75;
    else
    romo=0;
break;
case 3:
    if(tro==0)
    tro=90;
    else
    tro=0;
break;
}
}
void gate_menu(int m)
{
    switch(m)
    {
    case 1:
        if(mgo==0)
            mgo=1;
        else
            mgo=0;
        break;
    case 2:
        if(sgo==0)
            sgo=50;
        else
            sgo=0;
    break;
    }
}
void house_view(int m)
{
    switch(m)
    {
    case 1:
```

```
    view[0]=2.8;
    view[1]=2;
    view[2]=4.8;
    look[0]=2.8;
    look[1]=2;
    look[2]=1;
break;
case 2:
    view[0]=6;
    view[1]=12;
    view[2]=10;
    look[0]=2;
    look[1]=8;
    look[2]=2;
break;
case 3:
    view[0]=2;
    view[1]=2;
    view[2]=12.9;
    look[0]=3;
    look[1]=2;
    look[2]=3;
break;
case 4:
view[0]=1;
view[1]=6;
view[2]=-7;
look[0]=2;
look[1]=4;
look[2]=2;
break;
}
}
void menu()
{
int sub_menu1=glutCreateMenu(fan_menu);
glutAddMenuEntry("on/off fan(s)",1);
glutAddMenuEntry("speed up(up arrow)",2);
glutAddMenuEntry("speed down(down arrow)",3);

int sub_menu2=glutCreateMenu(door_menu);
glutAddMenuEntry("main door(q)",1);
glutAddMenuEntry("ground floor room door(o)",2);
glutAddMenuEntry("1st floor room door(p)",3);


int sub_menu3=glutCreateMenu(gate_menu);
glutAddMenuEntry("main gate(g)",1);
glutAddMenuEntry("sub gate(h)",2);

int sub_menu4=glutCreateMenu(house_view);
glutAddMenuEntry("front view(j)",3);
glutAddMenuEntry("top view(t)",2);
glutAddMenuEntry("inside view(i)",1);
glutAddMenuEntry("back view(k)",4);

glutCreateMenu(main_menu);
glutAddMenuEntry("Quit",1);
glutAddSubMenu("fan menu",sub_menu1);
glutAddSubMenu("open/close door",sub_menu2);
glutAddSubMenu("open/close gate",sub_menu3);
glutAddSubMenu("house view",sub_menu4);
glutAttachMenu(GLUT_LEFT_BUTTON);

}
int main(int argc,char**argv)
```

```
{
glutInit(&argc,argv);//to initialize the glut library
glutInitDisplayMode(GLUT_DOUBLE | GLUT_RGB | GLUT_DEPTH);
glutInitWindowSize(w,h);
glutInitWindowPosition(0,0);
glutCreateWindow("3D-HOUSE");
myinit();
glutDisplayFunc(display);
glutKeyboardFunc(Keyboard);
glutSpecialFunc(mySpecialKeyFunc);
menu();
//glutFullScreen();//to see o/p in full screen on monitor
glEnable(GL_LIGHTING);
glEnable(GL_LIGHT0);
glShadeModel(GL_SMOOTH);//smooth shaded
glEnable(GL_DEPTH_TEST);//to remove hidden surface
glEnable(GL_NORMALIZE);//to make normal vector to unit vector
glClearColor(0,.3,.8,0);
glViewport(0,0,w,h);
glutMainLoop();
//return 0;
}
```