```
#include <GL/gl.h>
#include <GL/glut.h>
#include <math.h>
const double PI = 3.141592654;
int frameNumber = 0;
void drawDisk(double radius) {
int d;
glBegin(GL_POLYGON);
for (d = 0; d < 32; d++) {
double angle = 2*PI/32 * d;
glVertex2d( radius*cos(angle), radius*sin(angle));
}
glEnd();
}
void drawship() {
glColor3f(1.5,1,0);
glBegin(GL_POLYGON);
glVertex2f(1.0f,5);//x1
glVertex2f(-2.0f,4.5);//x2
glVertex2f(1.0f,4);//y2
glEnd();
glColor3f(1.5,0,1);
glBegin(GL_LINES);
glVertex2f(1.0f,2);//x1
```

```
glVertex2f(1.0f,4);//y2
glEnd();
glColor3f(1,0,0);
glBegin(GL_POLYGON);
glVertex2f(-2.0f,0);//x1
glVertex2f(2.0f,0);//x2
glVertex2f(3.0f,2);//y2
glVertex2f(-3.0f,2);//y1
glEnd();
}
void drawSun() {
int i;
glColor3f(1.0,0.8,0.0);
for (i = 0; i < 20; i++) {
glRotatef( 360 / 20, 0, 0, 1 );
glBegin(GL_LINES);
glVertex2f(0, 0);
glVertex2f(0.75f, 0);
glEnd();
}
drawDisk(0.5);
glColor3f(0,0,0);
}
void display() {
```

```
glClear(GL_COLOR_BUFFER_BIT);
glLoadIdentity();
glColor3f(0, 0.6f, 0.2f);
glBegin(GL_POLYGON);
glVertex2f(-3,-1);
glVertex2f(1.5f,1.65f);
glVertex2f(5,-1);
glEnd();
glBegin(GL_POLYGON);
glVertex2f(-3,-1);
glVertex2f(3,2.1f);
glVertex2f(7,-1);
glEnd();
glBegin(GL_POLYGON);
glVertex2f(0,-1);
glVertex2f(6,1.2f);
glVertex2f(20,-1);
glEnd();
glColor3f(0.2f, 0.2f, 1.0f);
glBegin(GL_POLYGON);
glVertex2f(0,-0.4f);
glVertex2f(7,-0.4f);
glVertex2f(7,0.4f);
glVertex2f(0,0.4f);
```

```
glEnd();
glPushMatrix();
glTranslated(1.8,3,0);
glRotated(-frameNumber*2.7,0,0,1);
drawSun();
glPopMatrix();
glPushMatrix();
glTranslated(-3 + 15*(frameNumber % 300) / 300.0, 0, 0);
glScaled(0.3,0.3,1);
drawship();
glPopMatrix();
glutSwapBuffers();
}
void doFrame(int v) {
frameNumber++;
glutPostRedisplay();
glutTimerFunc(30,doFrame,0);
}
void init() {
glClearColor(0.6f, 0.6f, 1.0f, 0);
glMatrixMode(GL_PROJECTION);
glLoadIdentity();
glOrtho(0, 7, -1, 4, -1, 1);
glMatrixMode(GL_MODELVIEW);
```

```
}
int main(int argc, char** argv) {
glutInit(&argc, argv);
glutInitDisplayMode(GLUT_DOUBLE);
glutInitWindowSize(700,500);
glutInitWindowPosition(100,100);
glutCreateWindow("Animation");
init();
glutDisplayFunc(display);
glutTimerFunc(200,doFrame,0);
glutMainLoop();
return 0;
}
```