#include<GL/glut.h>  
#include<GL/glu.h>  
#include<GL/gl.h>  
#include<windows.h>  
#include<math.h>  
#include<iostream>  
#include<time.h>  
#define PI 3.1416  
  
//forw is for initial movement of the plane  
  
GLint forw=1,bflag=0 ,takeoff1=0,takeoff2=0,takeoff3=0 , plane1=0,plane2=0,plane3=0;  
GLfloat i, j, k , p=0 ,q=0,r=0;  
GLfloat sun\_spin=0, sun\_x=0, sun\_y=0;  
GLfloat ax=0,bx=0,cx=0,dx=0,str=500.0,mn=500.0;  
GLfloat sr=0.0,sg=0.749,sb=1.0;  
GLfloat spin = 0.0;  
void keyboard(unsigned char key , int x , int y);  
  
void update(int value)  
{  
//Plane position takeoff on x axis  
glutTimerFunc(1,update,0);//delay  
if(plane1==1)  
{  
  
if(forw==1)  
{  
        p+=1; //move plane to right  
}  
if(p>=65)  
{  
    forw=0;  
}  
if(forw==0)  //move plane downwards to left  
{  
    p-=1;  
    q-=0.3;  
    if(p<=4)  
    {  
        bflag=1;  
        p=4;  
        q=-20;  
        glTranslatef(p,q,0);  
  
}  
  
    if(takeoff1==1)  
    {  
    p+=5;  
    q+=2;  
    glTranslatef(p,q,0);  
    if(p>120)  
    {  
        forw=1;  
        bflag=0;  
        p=0;  
        q=0;  
        plane1=0;  
        takeoff1=0;  
    }  
    }  
}  
  
}  
  
if(plane2==1)  
{  
if(forw==1)  
{  
        p+=1; //move plane to right  
}  
if(p>=65)  
{  
    forw=0;  
}  
if(forw==0)  //move plane downwards to left  
{  
    p-=1;  
    q-=0.5;  
    if(p<=4)  
    {  
        bflag=1;  
        p=4;  
        q=-35;  
        glTranslatef(p,q,0);  
  
}  
  
    if(takeoff2==1)  
    {  
    p+=5;  
    q+=2;  
    glTranslatef(p,q,0);  
        if(p>120)  
    {  
        forw=1;  
        bflag=0;  
        p=0;  
        q=0;  
        plane2=0;  
        takeoff2=0;  
    }  
  
    }  
    }  
}  
if(plane3==1)  
{  
if(forw==1)  
{  
        p+=1; //move plane to right  
}  
if(p>=65)  
{  
    forw=0;  
}  
if(forw==0)  //move plane downwards to left  
{  
    p-=1;  
    q-=0.7;  
    if(p<=4)  
    {  
        bflag=1;  
        p=4;  
        q=-47;  
        glTranslatef(p,q,0);  
}  
  
    if(takeoff3==1)  
    {  
    p+=5;  
    q+=2;  
    glTranslatef(p,q,0);  
        if(p>120)  
    {  
        forw=1;  
        bflag=0;  
            p=0;  
        q=0;  
        plane3=0;  
        takeoff3=0;  
    }  
  
    }  
    }  
  
}  
  
glutPostRedisplay();  
}  
  
  
void drawJet()  
{  
    glScalef(4,4,0);  
    glTranslatef(0,20,0);  
    if(forw==0 && bflag==0)  
    {  
        glTranslatef(p+25,q+50,0);  
        glRotatef(180,0,1,0);  
        glTranslatef(-p-25,-q-50,0);  
    }  
//    if(bflag==1)  
//    {  
//        glTranslatef();  
//    }  
glTranslatef(p,q,0);  
Sleep(50);  
glColor3f(0.6,0.6,0.6);  
glBegin(GL\_POLYGON);  
glVertex2f(5.5,47.0);  
glVertex2f(8.5,47.0);  
glVertex2f(5.5,48.0);  
glVertex2f(4.5,48.0);  
glEnd();  
  
  
//left front wing  
  
glColor3f(0.6,0.6,0.6);  
glBegin(GL\_POLYGON);  
glVertex2f(13.0,47.0);  
glVertex2f(20.0,47.0);  
glVertex2f(13.0,50.0);  
glVertex2f(11.0,50.0);  
glEnd();  
  
//tail  
  
glColor3f(0.5,0.5,0.5);  
glBegin(GL\_POLYGON);  
glVertex2f(4.7,45.0);  
glVertex2f(5.5,51.0);  
glVertex2f(7.0,51.0);  
glVertex2f(9.0,45.0);  
glEnd();  
  
  
//body  
  
glColor3f(0.5,0.5,0.5);  
glBegin(GL\_POLYGON);  
glVertex2f(5.0,48.0);  
glVertex2f(11.0,48.0);  
glVertex2f(22.0,46.5);  
glVertex2f(22.0,45.0);  
glVertex2f(5.0,45.0);  
glEnd();  
  
  
//right front wing  
  
glColor3f(0.6,0.6,0.6);  
glBegin(GL\_POLYGON);  
glVertex2f(13.0,46.0);  
glVertex2f(18.0,46.0);  
glVertex2f(13.0,41.0);  
glVertex2f(11.0,41.0);  
glEnd();  
  
  
//dome  
  
glColor3f(0.0,0.0,0.0);  
glBegin(GL\_POLYGON);  
glVertex2f(13.0,47.0);  
glVertex2f(15.0,48.5);  
glVertex2f(17.0,49.0);  
glVertex2f(19.0,48.0);  
glVertex2f(21.0,46.0);  
glVertex2f(17.0,46.0);  
glVertex2f(15.0,47.5);  
glVertex2f(13.0,47.0);  
glEnd();  
  
  
//right tail wing  
  
glColor3f(0.6,0.6,0.6);  
glBegin(GL\_POLYGON);  
glVertex2f(5.5,47.0);  
glVertex2f(8.5,47.0);  
glVertex2f(5.5,43.0);  
glVertex2f(4.5,43.0);  
glEnd();  
  
  
// front tip  
  
glColor3f(0.4,0.4,0.4);  
glBegin(GL\_POLYGON);  
glVertex2f(22.0,45.0);  
glVertex2f(22.3,45.375);  
glVertex2f(22.6,45.75);  
glVertex2f(22.3,46.125);  
glVertex2f(22.0,46.5);  
glEnd();  
  
glFlush();  
//glutSwapBuffers();  
  
}  
  
  
void strip(int m , int l)  
{  
glPushMatrix();  
glTranslated(m,l,0.0);  
glColor3f(1.0,1.0,1.0);  
glBegin(GL\_POLYGON);//white strips on road  
glVertex2f(0.0,40.0);  
glVertex2f(4.0,30.0);  
glVertex2f(30.0,30.0);  
glVertex2f(25.0,40.0);  
glEnd();  
glPopMatrix();  
}  
  
void left\_road()  
{  
    glColor3f(0,0,0);  
    glBegin(GL\_POLYGON);  
    glVertex2f(40,200);  
    glVertex2f(80,200);  
    glVertex2f(150,10);  
    glVertex2f(80,10);  
    glEnd();  
    glFlush();  
}  
  
void track1()  
{  glColor3f(0,0,0);  
    glBegin(GL\_POLYGON);  
    glVertex2f(80,190);  
    glVertex2f(600,190);  
    glVertex2f(700,170);  
    glVertex2f(80,170);  
    glEnd();  
    glFlush();  
}  
  
void track2()  
{  glColor3f(0,0,0);  
    glBegin(GL\_POLYGON);  
    glVertex2f(80,140);  
    glVertex2f(600,140);  
    glVertex2f(700,120);  
    glVertex2f(80,120);  
    glEnd();  
    glFlush();  
}  
  
void track3()  
{  glColor3f(0,0,0);  
    glBegin(GL\_POLYGON);  
    glVertex2f(80,90);  
    glVertex2f(600,90);  
    glVertex2f(700,70);  
    glVertex2f(80,70);  
    glEnd();  
    glFlush();  
}  
  
  
  
void ground()  
{  
    glColor3f(0,0.2,0.13);  
    glBegin(GL\_POLYGON);  
    glVertex2f(0,0);  
    glVertex2f(500,0);  
    glVertex2f(500,200);  
    glVertex2f(0,200);  
    glEnd();  
    glFlush();  
}  
  
void circle(int r,int x,int y)  
{  
    float theetha;  
    //glColor3f(1.0,1.0,1.0);  
    glBegin(GL\_POLYGON);  
    for(int i=0;i<360;i++)  
    {  
        theetha=i\*3.142/180;  
        glVertex2f(x+r\*cos(theetha),y+r\*sin(theetha));  
    }  
    glEnd();  
    glFlush();  
  
}  
void cloud\_model\_one(){  
  
    glColor3f(1.25, 0.924, 0.930);  
  
    ///Top\_Left  
  
    glPushMatrix();  
    glTranslatef(320,210,0);  
    circle(15,50,100);  
    glPopMatrix();  
  
    ///Top  
  
    glPushMatrix();  
    glTranslatef(340, 225, 0);  
    circle(16,50,100);  
    glPopMatrix();  
  
    ///Right  
  
    glPushMatrix();  
    glTranslatef(360,210,0);  
    circle(16,50,100);  
    glPopMatrix();  
  
  
    ///middle\_Fill  
    glPushMatrix();  
    glTranslatef(355,210,0);  
    circle(16,50,100);  
    glPopMatrix();  
  
    glPushMatrix();  
    glTranslatef(350,210,0);  
    circle(16,50,100);  
    glPopMatrix();  
  
    glPushMatrix();  
    glTranslatef(345,204,0);  
    circle(10,50,100);  
    glPopMatrix();  
  
    glPushMatrix();  
    glTranslatef(340,204,0);  
    circle(10,50,100);  
    glPopMatrix();  
  
    glPushMatrix();  
    glTranslatef(335,204,0);  
    circle(10,50,100);  
    glPopMatrix();  
  
    glPushMatrix();  
    glTranslatef(330,204,0);  
    circle(10,50,100);  
    glPopMatrix();  
  
    glPushMatrix();  
    glTranslatef(325,204,0);  
    circle(10,50,100);  
    glPopMatrix();  
  
    glPushMatrix();  
    glTranslatef(320,204,0);  
    circle(10,50,100);  
    glPopMatrix();  
  
    glPushMatrix();  
    glTranslatef(315,204,0);  
    circle(10,50,100);  
    glPopMatrix();  
  
    glPushMatrix();  
    glTranslatef(310,204,0);  
    circle(10,50,100);  
    glPopMatrix();  
  
    glPushMatrix();  
    glTranslatef(305,204,0);  
    circle(10,50,100);  
    glPopMatrix();  
  
}  
  
void cloud\_model\_Two(){  
    glColor3f(1.25, 0.924, 0.930);  
  
    ///Left\_Part  
    glPushMatrix();  
    glTranslatef(305,205,0);  
    circle(10,50,100);  
    glPopMatrix();  
  
    ///Top  
  
    glPushMatrix();  
    glTranslatef(320,210,0);  
    circle(15,50,100);  
    glPopMatrix();  
  
    ///Right\_Part  
    glPushMatrix();  
    glTranslatef(334,207,0);  
    circle(10,50,100);  
    glPopMatrix();  
  
    ///Bottom\_Part  
    glPushMatrix();  
    glTranslatef(320,207,0);  
    circle(10,50,100);  
    glPopMatrix();  
  
  
  
}  
  
void cloud\_model\_Three(){  
    glColor3f(1.25, 0.924, 0.930);  
  
    ///Left\_Part  
    glPushMatrix();  
    glTranslatef(300,200,0);  
    circle(15,50,100);  
    glPopMatrix();  
  
    ///Top\_Left  
  
    glPushMatrix();  
    glTranslatef(320,210,0);  
    circle(15,50,100);  
    glPopMatrix();  
  
    ///Top  
    glPushMatrix();  
    glTranslatef(340,220,0);  
    circle(16,50,100);  
    glPopMatrix();  
  
    ///Top\_Right  
    glPushMatrix();  
    glTranslatef(360,210,0);  
    circle(15,50,100);  
    glPopMatrix();  
  
    ///Right\_Part  
    glPushMatrix();  
    glTranslatef(380,200,0);  
    circle(15,50,100);  
    glPopMatrix();  
  
    ///Bottom\_Right  
    glPushMatrix();  
    glTranslatef(360,190,0);  
    circle(20,50,100);  
    glPopMatrix();  
  
    ///Bottom\_Left  
    glPushMatrix();  
    glTranslatef(320,190,0);  
    circle(20,50,100);  
    glPopMatrix();  
  
    ///Bottom  
    glPushMatrix();  
    glTranslatef(340,190,0);  
    circle(20,50,100);  
    glPopMatrix();  
  
  
  
  
    ///\*\*\*\*Fill End\*\*\*\*  
  
}  
  
  
///=================///  
///\*\*\*   Object  \*\*\*///  
///=================///  
///\*\*\* Sun \*\*\*///  
void Sun(){  
    glColor3f(1, 1, 0);  
    glPushMatrix();  
    //Moving\_Sun\_Model();  
    circle(50,180,300);  
    glPopMatrix();  
}  
///\*\*\* Cloud\_One\_Model\_One \*\*\*///  
void cloud\_one(){  
    glPushMatrix();  
    glTranslatef(cx,-40,0);  
    cloud\_model\_one();  
    glPopMatrix();  
  
}  
  
///\*\*\* Cloud\_Two\_Model\_one \*\*\*///  
  
void cloud\_two(){  
    glPushMatrix();  
    glTranslatef(bx+100,150,0);  
    cloud\_model\_one();  
    glPopMatrix();  
  
}  
  
///\*\*\* Cloud\_Three\_Model\_Two \*\*\*///  
  
void cloud\_three(){  
    glPushMatrix();  
    glTranslatef(ax-80,80,0);  
    cloud\_model\_Two();  
    glPopMatrix();  
  
}  
///\*\*\* Cloud\_Four\_Model\_Two \*\*\*///  
  
void cloud\_four(){  
    glPushMatrix();  
    glTranslatef(dx+300,125,0);  
    cloud\_model\_Two();  
    glPopMatrix();  
  
}  
///\*\*\* Cloud\_Five\_Model\_Three \*\*\*///  
void cloud\_five(){  
  
    glPushMatrix();  
    glTranslatef(ax+-300,170,0);  
    cloud\_model\_Three();  
    glPopMatrix();  
}  
///\*\*\* Cloud\_Six\_Model\_Three \*\*\*///  
void cloud\_six(){  
  
    glPushMatrix();  
    glTranslatef(cx+-500,20,0);  
    cloud\_model\_Three();  
    glPopMatrix();  
}  
  
  
  
void display()  
{  
    glClear(GL\_COLOR\_BUFFER\_BIT);  
    glLoadIdentity();  
    //circle(50,180,300);  
    Sun();  
    ground();  
    cloud\_three();  
    cloud\_four();  
  
    cloud\_one();  
  
    cloud\_two();  
    cloud\_five();  
    cloud\_six();  
    left\_road();  
    track1();  
    track2();  
    track3();  
    strip(150,45);  
    strip(250,45);  
    strip(350,45);  
    strip(450,45);  
  
    strip(150-20,95);  
    strip(250-20,95);  
    strip(350-20,95);  
    strip(450-20,95);  
  
    strip(150-40,95+50);  
    strip(250-40,95+50);  
    strip(350-40,95+50);  
    strip(450-40,95+50);  
  
    drawJet();  
    glutSwapBuffers();  
    glFlush();  
  
}  
  
void doInit()  
{  
  
/\* Background and foreground color \*/  
    glClearColor(0.49,0.73,0.91,0.0);  
    glMatrixMode(GL\_PROJECTION);  
    glLoadIdentity();  
    gluOrtho2D(0.0,500.0,0.0,500.0);  
    glMatrixMode(GL\_MODELVIEW);  
  
  
}  
  
void keyboard(unsigned char key , int x , int y)  
{  
    if(key=='a'||key=='A') //Add plane 1  
    {  
        plane1=1;  
  
    }  
    if(key=='1') // Take off plane 1  
    {  
        takeoff1=1;  
    }  
  
    if(key=='s'||key=='S') //Add plane 2  
    {  
        plane2=1;  
  
    }  
    if(key=='2') // Take off plane 2  
        takeoff2=1;  
  
  
    if(key=='d'||key=='D') //Add plane 3  
    {  
        plane3=1;  
  
    }  
    if(key=='3') // Take off plane 3  
    {  
        takeoff3=1;  
    }  
  
    if(key=='e'||key=='E') // Exit the program  
       {  
           exit(0);  
       }  
}  
  
int main(int argc, char \*argv[])  
{  
    glutInit(&argc, argv);  
  
glutInitDisplayMode(GLUT\_DOUBLE|GLUT\_RGB);  
  
glutInitWindowSize(800,700);  
    glutInitWindowPosition(100,100);  
    glutCreateWindow("Glut Plane");  
  
glutDisplayFunc(display);  
glutKeyboardFunc(keyboard);  
  
  
    doInit();  
glutTimerFunc(1,update,0);  
  
    glutMainLoop();  
return 0;  
}