```
#include<GL/glut.h>
#include <GL/glu.h>
#include<math.h>
#include <stdlib.h>
#include<stdio.h>
#define PI 3.1416



GLint i, j, k;
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 init(void)
{
        glClearColor(.40, .110, 1.0, 0.0);
        glMatrixMode(GL_PROJECTION);
        gluOrtho2D(0.0, 1000.0, 0.0, 500.0);
}




///*** Circle_Model***///
void circle(GLdouble rad)
{
   GLint points = 50;
   GLdouble delTheta = (2.0 * PI) / (GLdouble)points;
   GLdouble theta = 0.0;

   glBegin(GL_POLYGON);
   {
      for( i = 0; i <=50; i++, theta += delTheta )
      {
         glVertex2f(rad * cos(theta),rad * sin(theta));
      }
   }
   glEnd();
}
/// *** Sun_Model **///
void Sun_Model(){
   glPushMatrix();
   glTranslatef(500,0,0);
   circle(30);
   glPopMatrix();

}
void Moving_Sun_Model(){
   glPushMatrix();
```

```
    glRotatef(-sun_spin, 0,0,-.009);
    Sun_Model();
    glPopMatrix();

}
///*** Cloud_Model***///
void cloud_model_one(){

    glColor3f(1.25, 0.924, 0.930);

    ///Top_Left

    glPushMatrix();
    glTranslatef(320,210,0);
    circle(15);
    glPopMatrix();

    ///Top

    glPushMatrix();
    glTranslatef(340, 225, 0);
    circle(16);
    glPopMatrix();

    ///Right

    glPushMatrix();
    glTranslatef(360,210,0);
    circle(16);
    glPopMatrix();


    ///middle_Fill
    glPushMatrix();
    glTranslatef(355,210,0);
    circle(16);
    glPopMatrix();

    glPushMatrix();
    glTranslatef(350,210,0);
    circle(16);
    glPopMatrix();

    glPushMatrix();
    glTranslatef(345,204,0);
    circle(10);
    glPopMatrix();

    glPushMatrix();
    glTranslatef(340,204,0);
    circle(10);
    glPopMatrix();
```

```
    glPushMatrix();
    glTranslatef(335,204,0);
    circle(10);
    glPopMatrix();

    glPushMatrix();
    glTranslatef(330,204,0);
    circle(10);
    glPopMatrix();

    glPushMatrix();
    glTranslatef(325,204,0);
    circle(10);
    glPopMatrix();

    glPushMatrix();
    glTranslatef(320,204,0);
    circle(10);
    glPopMatrix();

    glPushMatrix();
    glTranslatef(315,204,0);
    circle(10);
    glPopMatrix();

    glPushMatrix();
    glTranslatef(310,204,0);
    circle(10);
    glPopMatrix();

    glPushMatrix();
    glTranslatef(305,204,0);
    circle(10);
    glPopMatrix();

    ///****Fill End****

}

void cloud_model_Two(){
    glColor3f(1.25, 0.924, 0.930);

    ///Left_Part
    glPushMatrix();
    glTranslatef(305,205,0);
    circle(10);
    glPopMatrix();

    ///Top

    glPushMatrix();
```

```
    glTranslatef(320,210,0);
    circle(15);
    glPopMatrix();

    ///Right_Part
    glPushMatrix();
    glTranslatef(334,207,0);
    circle(10);
    glPopMatrix();

    ///Bottom_Part
    glPushMatrix();
    glTranslatef(320,207,0);
    circle(10);
    glPopMatrix();



}

void cloud_model_Three(){
    glColor3f(1.25, 0.924, 0.930);

    ///Left_Part
    glPushMatrix();
    glTranslatef(300,200,0);
    circle(15);
    glPopMatrix();

    ///Top_Left

    glPushMatrix();
    glTranslatef(320,210,0);
    circle(15);
    glPopMatrix();

    ///Top
    glPushMatrix();
    glTranslatef(340,220,0);
    circle(16);
    glPopMatrix();

    ///Top_Right
    glPushMatrix();
    glTranslatef(360,210,0);
    circle(15);
    glPopMatrix();

    ///Right_Part
    glPushMatrix();
    glTranslatef(380,200,0);
    circle(15);
```

```
    glPopMatrix();

    ///Bottom_Right
    glPushMatrix();
    glTranslatef(360,190,0);
    circle(20);
    glPopMatrix();

    ///Bottom_Left
    glPushMatrix();
    glTranslatef(320,190,0);
    circle(20);
    glPopMatrix();

    ///Bottom
    glPushMatrix();
    glTranslatef(340,190,0);
    circle(20);
    glPopMatrix();




    ///****Fill End****

}
///*** Hill_Model***///
void hill_big(){

    ///Hill
        glBegin(GL_POLYGON);
    glColor3f(0.38, 0.41, 0.36);
        glVertex2i(70, 70);
        glVertex2i(200, 225);
        glVertex2i(330, 70);

        glEnd();

        ///Hill_Snow
    glBegin(GL_POLYGON);
    glColor3f(1.25, 0.924, 0.930);

        glVertex2i(200, 225);
        glVertex2i(230, 190);
        glVertex2i(220, 180);
        glVertex2i(200, 190);
        glVertex2i(190, 180);
        glVertex2i(170, 190);

        glEnd();

}
```

```
void hill_small(){
   ///Hill_Small
         glBegin(GL_POLYGON);
   glColor3f(0.11, 0.23, 0.36);
         glVertex2i(250, 100);
         glVertex2i(310, 175);
         glVertex2i(370, 100);

         glEnd();

   ///Hill_Small_Snow
         glBegin(GL_POLYGON);
   glColor3f(1.25, 0.924, 0.930);
   glVertex2i(290, 150);
         glVertex2i(310, 175);
         glVertex2i(330, 150);
         glVertex2i(325, 140);
         glVertex2i(310, 150);
         glVertex2i(300, 140);


         glEnd();
}
///*** Tilla_Model ***///
void Tilla_One_Model(){
   ///Tilla
         glBegin(GL_POLYGON);
         glColor3f(0.1, 1.293, 0.0);
         glVertex2i(125, 70);
         glVertex2i(150, 80);
         glVertex2i(160, 90);
         glVertex2i(170, 90);
         glVertex2i(180, 100);
         glVertex2i(190, 105);
         glVertex2i(200, 108);
         glVertex2i(300, 110);
         glVertex2i(300, 70);

         glEnd();

}

void Tilla_Two_Model(){

         glColor3f(0.1, 1.293, 0.0);
   /// Left_Part
   glPushMatrix();
   glTranslatef(430,90,0);
   circle(30);
   glPopMatrix();

   glPushMatrix();
```

```
glTranslatef(420,87,0);
circle(30);
glPopMatrix();

glPushMatrix();
glTranslatef(410,80,0);
circle(30);
glPopMatrix();

glPushMatrix();
glTranslatef(400,80,0);
circle(30);
glPopMatrix();

glPushMatrix();
glTranslatef(390,70,0);
circle(30);
glPopMatrix();

///Right_Part
glPushMatrix();
glTranslatef(445,80,0);
circle(30);
glPopMatrix();

glPushMatrix();
glTranslatef(455,75,0);
circle(30);
glPopMatrix();

glPushMatrix();
glTranslatef(465,70,0);
circle(30);
glPopMatrix();

glPushMatrix();
glTranslatef(470,65,0);
circle(30);
glPopMatrix();

glPushMatrix();
glTranslatef(480,60,0);
circle(30);
glPopMatrix();

glPushMatrix();
glTranslatef(485,55,0);
circle(20);
glPopMatrix();
```

```
}
///*** House_Model ***///
void house(){
   ///House_Roof
        glBegin(GL_POLYGON);
   glColor3f(.990, 0.0, 0.0);
        glVertex2i(285, 105);
        glVertex2i(285, 130);
        glVertex2i(380, 115);
        glVertex2i(380, 105);

        glEnd();

   ///House_Roof_Shadow
        glBegin(GL_POLYGON);
   glColor3f(.890, 0.0, 0.0);
        glVertex2i(285, 105);
        glVertex2i(285, 120);
        glVertex2i(380, 105);
        glVertex2i(380, 105);

        glEnd();

   ///House_Fence
        glBegin(GL_POLYGON);
   glColor3f(.555, 1.0, 1.0);
        glVertex2i(290, 70);
        glVertex2i(290, 104);
        glVertex2i(375, 104);
        glVertex2i(375, 70);

        glEnd();

   ///House_Fence_Shadow
        glBegin(GL_POLYGON);
   glColor3f(.555, 0.924, 0.930);
        glVertex2i(310, 70);
        glVertex2i(350, 104);
        glVertex2i(375, 104);
        glVertex2i(375, 70);

        glEnd();

   ///House_Door
        glBegin(GL_POLYGON);
   glColor3f(0.38, 0.41, 0.36);
        glVertex2i(330, 70);
        glVertex2i(330, 100);
        glVertex2i(350, 100);
        glVertex2i(350, 70);

        glEnd();
```

```
///House_Window1
	glBegin(GL_POLYGON);
glColor3f(0.38, 0.21, 0.26);
	glVertex2i(295, 75);
	glVertex2i(295, 90);
	glVertex2i(310, 90);
	glVertex2i(310, 75);

	glEnd();

///House_Window2
	glBegin(GL_POLYGON);
glColor3f(0.38, 0.21, 0.26);
	glVertex2i(312, 75);
	glVertex2i(312, 90);
	glVertex2i(327, 90);
	glVertex2i(327, 75);

	glEnd();

///House_Window3
	glBegin(GL_POLYGON);
glColor3f(0.38, 0.21, 0.26);
	glVertex2i(355, 75);
	glVertex2i(355, 90);
	glVertex2i(370, 90);
	glVertex2i(370, 75);

	glEnd();

///House_Small_Roof
	glBegin(GL_POLYGON);
glColor3f(1.0, 0.0, 0.0);
	glVertex2i(250, 90);
	glVertex2i(257, 104);
	glVertex2i(290, 104);
	glVertex2i(290, 90);

	glEnd();

///House_Small_Fence
	glBegin(GL_POLYGON);
glColor3f(.555, .724, .930);
	glVertex2i(255, 70);
	glVertex2i(255, 90);
	glVertex2i(290, 90);
	glVertex2i(290, 70);

	glEnd();

///House_Small_Door
```

```
        glBegin(GL_POLYGON);
    glColor3f(0.11, 0.23, 0.36);
        glVertex2i(260, 70);
        glVertex2i(260, 80);
        glVertex2i(285, 80);
        glVertex2i(285, 70);

        glEnd();


}
///*** Field_Model ***///
void field(){
    ///Field
        glBegin(GL_POLYGON);
    glColor3f(0.533, 1.293, 0.0);
        glVertex2i(0, 50);
        glVertex2i(0, 70);
        glVertex2i(1000, 70);
        glVertex2i(1000, 50);

        glEnd();


    ///Field_Shadow
        glBegin(GL_POLYGON);
        glColor3f(0.1, 1.293, 0.0);
        glVertex2i(0, 0);
        glVertex2i(0, 50);
        glVertex2i(1000, 50);
        glVertex2i(1000, 0);

        glEnd();


}
///*** Tree_Model ***///
void Tree_Model_One(){


    glPushMatrix();
    glTranslatef(110,110,0);
    circle(15);
    glPopMatrix();

    glPushMatrix();
    glTranslatef(110,100,0);
    circle(15);
    glPopMatrix();

    glBegin(GL_POLYGON);
    glColor3f(0.38, 0.21, 0.26);
```

```
        glVertex2f(109, 70);
        glVertex2f(109, 90);
        glVertex2f(111, 90);
        glVertex2f(111, 70);

    glEnd();

}
void Tree_Model_Two(){

    glPushMatrix();
    glTranslatef(130,130,0);
    circle(5);
    glPopMatrix();

    glPushMatrix();
    glTranslatef(125,126,0);
    circle(5);
    glPopMatrix();

    glPushMatrix();
    glTranslatef(135,126,0);
    circle(5);
    glPopMatrix();

    glPushMatrix();
    glTranslatef(130,125,0);
    circle(5);
    glPopMatrix();

    glBegin(GL_POLYGON);
    glColor3f(0.38, 0.21, 0.26);
        glVertex2f(129, 110);
        glVertex2f(129, 124);
        glVertex2f(131, 124);
        glVertex2f(131, 110);

    glEnd();
}

void Tree_Model_Three(){


    glBegin(GL_POLYGON);

        glVertex2f(125, 123);
        glVertex2f(133, 145);
        glVertex2f(141, 123);

        glEnd();

    glBegin(GL_POLYGON);
```

```
    glColor3f(0.38, 0.21, 0.26);
        glVertex2f(132, 110);
        glVertex2f(132, 124);
        glVertex2f(134, 124);
        glVertex2f(134, 110);

    glEnd();
}

/// *** Windmill_Stand_Model ***///
void Windmill_Stand_Model(){

    glColor3f(0.38, 0.41, 0.36);
    glBegin(GL_POLYGON);
    glVertex2i(375, 100);
    glVertex2i(380, 240);
    glVertex2i(384, 240);
    glVertex2i(390, 100);
    glEnd();
}

///*** Windmill_Blades_Model ***///

void Windmill_Blade(){

    ///Blade_One
    glPushMatrix();
    glRotatef(spin,0,0,90);
    glBegin(GL_POLYGON);
    glVertex2i(-5, 0);
    glVertex2i(-85, -36);
    glVertex2i(-83, -37);
    glVertex2i(-3, -8);
    glEnd();
    glPopMatrix();

    ///Blade_Two
    glPushMatrix();
    glRotatef(spin,0,0,90);
    glBegin(GL_POLYGON);
    glVertex2i(0, 5);
    glVertex2i(45, 70);
    glVertex2i(50, 73);
    glVertex2i(5, 0);
    glEnd();
    glPopMatrix();

    ///Blade_Three
    glPushMatrix();
    glRotatef(spin,0,0,90);
    glBegin(GL_POLYGON);
    glVertex2i(68, -78);
```

```
    glVertex2i(0,0);
    glVertex2i(5, 5);
    glVertex2i(70, -77);
    glEnd();
    glPopMatrix();

}
///*** Windmill_Final_Model ***///
void Windmill(){



        ///Windmill_Stand
    glColor3f(0.38, 0.41, 0.36);
    glPushMatrix();
    Windmill_Stand_Model();
    glPopMatrix();

    ///Windmill_Motor
    glColor3f(0.11, 0.23, 0.36);
    glPushMatrix();
    glTranslatef(380,250,0);
    circle(10);
    glPopMatrix();

    ///Windmill_Rotary_Blades
    glColor3f(0.11, 0.23, 0.36);
    glPushMatrix();
    glTranslatef(380,251,0);
    Windmill_Blade();
    glPopMatrix();




}


///Model_End
///
================================================================================
================================///


///=================///
///***   Object  ***///
///=================///

///*** Sun ***///
void Sun(){
    glColor3f(1, 1, 0);
    glPushMatrix();
```

```
    Moving_Sun_Model();
    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();
}

///*** House_One ***///
void house_one(){
    glPushMatrix();
    glTranslatef(0,0,0);
    house();
    glPopMatrix();
}
///*** House_Two ***///
void house_two(){
    glPushMatrix();
    glTranslatef(450,0,0);
    house();
    glPopMatrix();
}
///*** House_Two ***///
void house_three(){
    glPushMatrix();
    glTranslatef(320, 37,0);
    house();
    glPopMatrix();
}
///*** Hill_big_One ***///
void Hill_Big_One(){
    glPushMatrix();
    glTranslatef(0,0,0);
    hill_big();
    glPopMatrix();
}
///*** Hill_big_Two ***///
void Hill_Big_Two(){
    glPushMatrix();
    glTranslatef(550,-20,0);
    hill_big();
    glPopMatrix();
}
///*** Hill_Small_One ***///
void Hill_Small_One(){
    glPushMatrix();
    glTranslatef(0,0,0);
    hill_small();
    glPopMatrix();

}
/// *** Tilla_One_Model_One ***///

void Tilla_One(){
```

```
    glPushMatrix();
    glTranslatef(0,0,0);
    Tilla_One_Model();
    glPopMatrix();

}
/// *** Tilla_Two_Model_Two ***///
void Tilla_Two(){

    glPushMatrix();
    glTranslatef(0,0,0);
    Tilla_Two_Model();
    glPopMatrix();


}
/// *** Tilla_Three_Model_Two ***///
void Tilla_Three(){

    glPushMatrix();
    glTranslatef(400,0,0);
    Tilla_Two_Model();
    glPopMatrix();


}
/// *** Tilla_Four_Model_One ***///
void Tilla_Four(){

    glColor3f(0.833, 1., 0.0);
    glPushMatrix();
    glTranslatef(380,0,0);
    Tilla_One_Model();
    glPopMatrix();


}
///*** Tree_1 ***///
void Tree_One(){
    glColor3f(0.533, 1.293, 0.0);
    glPushMatrix();
    glTranslatef(0,0,0);
    Tree_Model_One();
    glPopMatrix();
}

///*** Tree_2 ***///
void Tree_Two(){
    glColor3f(0.533, 1.293, 0.0);
    glPushMatrix();
    glTranslatef(540,0,0);
    Tree_Model_One();
```

```
    glPopMatrix();
}

///*** Tree_3 ***///
void Tree_Three(){
    glColor3f(0.533, 1.293, 0.0);
    glPushMatrix();
    glTranslatef(750,0,0);
    Tree_Model_One();
    glPopMatrix();
}
///*** Tree_4 ***///
void Tree_Four(){
    glColor3f(0.533, 1.293, 0.0);
    glPushMatrix();
    glTranslatef(292,40,0);
    Tree_Model_One();
    glPopMatrix();
}

///*** Tree_5 ***///
void Tree_Five(){
    glColor3f(0.533, 1.293, 0.0);
    glPushMatrix();
    glTranslatef(30,-20,0);
    Tree_Model_Two();
    glPopMatrix();
}

///*** Tree_6 ***///
void Tree_Six(){
    glColor3f(0.533, 1.293, 0.0);
    glPushMatrix();
    glTranslatef(50,-10,0);
    Tree_Model_Two();
    glPopMatrix();
}
///*** Tree_7 ***///
void Tree_Seven(){
    glColor3f(0.533, 1.293, 0.0);
    glPushMatrix();
    glTranslatef(322,0,0);
    Tree_Model_Two();
    glPopMatrix();
}

///*** Tree_8 ***///
void Tree_Eight(){
    glColor3f(0.533, 1.293, 0.0);
    glPushMatrix();
    glTranslatef(350,-15,0);
    Tree_Model_Two();
```

```
    glPopMatrix();
}

///*** Tree_9 ***///
void Tree_Nine(){
    glColor3f(0.533, 1.293, 0.0);
    glPushMatrix();
    glTranslatef(760,-25,0);
    Tree_Model_Two();
    glPopMatrix();
}

///*** Tree_10 ***///
void Tree_Ten(){
    glColor3f(0.533, 1.293, 0.0);
    glPushMatrix();
    glTranslatef(90,-2,0);
    Tree_Model_Three();
    glPopMatrix();
}

///*** Tree_11 ***///
void Tree_Eleven(){
    glColor3f(0.533, 1.293, 0.0);
    glPushMatrix();
    glTranslatef(125,0,0);
    Tree_Model_Three();
    glPopMatrix();
}

///*** Tree_12 ***///
void Tree_Twelve(){
    glColor3f(0.533, 1.293, 0.0);
    glPushMatrix();
    glTranslatef(408,-22,0);
    Tree_Model_Three();
    glPopMatrix();
}

/// *** Windmill ***///
void Windmill_One(){
    glColor3f(0.11, 0.23, 0.36);
    glPushMatrix();
    glTranslatef(0,-10,0);
    Windmill();
    glPopMatrix();

}

void Windmill_Two(){
    glColor3f(0.11, 0.23, 0.36);
    glPushMatrix();
```

```
    glTranslatef(508,-70,0);
    Windmill();
    glPopMatrix();

}
void Windmill_Three(){
    glColor3f(0.11, 0.23, 0.36);
    glPushMatrix();
    glTranslatef(108,-90,0);
    Windmill();
    glPopMatrix();

}


void display(void)
{
        glClear(GL_COLOR_BUFFER_BIT);
        glColor3f(0.0, 0.0, 1.0);

    ///*** Object_Layer ***///
    Sun();

    Windmill_Three();

    Hill_Big_One();
    Tilla_Four();

    house_three();

    Hill_Big_Two();
    Hill_Small_One();

    cloud_three();
    cloud_four();

    Windmill_One();
    Windmill_Two();


    Tilla_One();
    Tilla_Two();
    Tilla_Three();


    house_one();
    cloud_one();
    house_two();


    Tree_One();
    Tree_Two();
```

```
    Tree_Three();
    Tree_Four();
    Tree_Five();
    Tree_Six();
    Tree_Seven();
    Tree_Eight();
    Tree_Nine();
    Tree_Ten();
    Tree_Eleven();
    Tree_Twelve();



    cloud_two();
    cloud_five();
    cloud_six();
    field();

        glFlush();
}
///=======================///
///*** Speed & Movement ***///
///=======================///
///*** Sun_Move ***///
void sun_move(){

    sun_spin = sun_spin + 0.0008;

}
void move_right(){

    sun_move();

    spin = spin +.1;
    ax = ax + .05;
    bx = bx + .08;
    cx = cx + .10;
    dx = dx + .15;

    if(cx>1000){
       cx = -300;
    }
    if(bx>1000){
        bx= -400;

    }
    if(cx>1000){
        cx= -400;

    }
    if(dx>1000){
        dx= -500;
```

```
    }


    glutPostRedisplay();
}



void mouse(int key, int state, int x, int y){
    switch (key)
    {
    case GLUT_LEFT_BUTTON:
        if (state == GLUT_DOWN)
        {
            glutIdleFunc(move_right);
        }
        break;
    case GLUT_MIDDLE_BUTTON:
    case GLUT_RIGHT_BUTTON:
        if (state == GLUT_DOWN)
        {
            glutIdleFunc(NULL);
        }
        break;
    default:
        break;
    }
}

int main(int argc, char** argv)
{
            glutInit(&argc, argv);
            glutInitDisplayMode(GLUT_SINGLE | GLUT_RGB);
            glutInitWindowPosition(50, 50);
            glutInitWindowSize(1900, 1900);
            glutCreateWindow("Smart Village");
            init();
            glutDisplayFunc(display);
    glutMouseFunc(mouse);
            glutMainLoop();
}
```