```
#include<cstdio>
#include<math.h>
#include <vector>
#include <cstdlib>
# define PI 3.14159265358979323846
#include <GL/gl.h>
#include <GL/glut.h>
void PointLight(const float x, const float y, const float z,  const float amb, const float diff, const
float spec);
void PointLight(const float x, const float y, const float z, const float amb, const float diff, const float
spec)
{
  glEnable(GL_LIGHTING);
  GLfloat light_ambient[] = { amb,amb,amb, 1.0 };
  GLfloat light_position[] = {-0.9,.9,0, 0.0 };
  glLightfv(GL_LIGHT0, GL_AMBIENT, light_ambient);
  glLightfv(GL_LIGHT0, GL_POSITION, light_position);
  glEnable(GL_LIGHT0); //enable the light after setting the properties
}
GLfloat position22 = 0.0f;
GLfloat speed22 = 0.007f;
void birdd(int value) {
    if(position22 > 1.0)
        position22 =-1.0f;
    position22 += speed22;
            glutPostRedisplay();
            glutTimerFunc(100, birdd, 0);
}
GLfloat position4 = 0.0f;
GLfloat speed4 =-0.01f;
void sunn(int value)
{
   if(position4 > 1.0)
        position4 = 0.0f;
    position4 += speed4;
            glutPostRedisplay();
glutTimerFunc(100, sunn, 0);
}
GLfloat position3 = 0.0f;
GLfloat speed3 =- 0.5f;

GLfloat position2 = 0.0f;
GLfloat speed2 = 0.004f;
void cloud(int value) {
    if(position2 > 1.0)
        position2 =-1.0f;
    position2 += speed2;
            glutPostRedisplay();
            glutTimerFunc(100, cloud, 0);
}
GLfloat position1 = 1.0f;
GLfloat speed1 =-0.005f;
```

```
void boat(int value)
{
  if(position1 <- 1.0)
       position1 = 1.0f;
   position1 += speed1;
glutPostRedisplay();
glutTimerFunc(100, boat, 0);
}
void cloud1()
{
   int i;

          GLfloat x=.5f; GLfloat y=.86f; GLfloat radius =.05f;
          int triangleAmount = 20;
          GLfloat twicePi = 2.0f * PI;

          glBegin(GL_TRIANGLE_FAN);
             glColor3ub(255, 217, 255);
                   glVertex2f(x, y); // center of circle
                   for(i = 0; i <= triangleAmount;i++) {
                             glVertex2f(
                             x + (radius * cos(i *  twicePi / triangleAmount)),
                                y + (radius * sin(i * twicePi / triangleAmount))
                             );
                   }
          glEnd();

GLfloat a=.55f; GLfloat b=.83f;

          glBegin(GL_TRIANGLE_FAN);
             glColor3ub(255, 217, 255);
                   glVertex2f(a, b); // center of circle
                   for(i = 0; i <= triangleAmount;i++) {
                             glVertex2f(
                             a + (radius * cos(i *  twicePi / triangleAmount)),
                                b + (radius * sin(i * twicePi / triangleAmount))
                             );
                   }
          glEnd();

          GLfloat c=.45f; GLfloat d=.83f;

          glBegin(GL_TRIANGLE_FAN);
             glColor3ub(255, 217, 255);
                   glVertex2f(c, d); // center of circle
                   for(i = 0; i <= triangleAmount;i++) {
                             glVertex2f(
                             c + (radius * cos(i *  twicePi / triangleAmount)),
                                d + (radius * sin(i * twicePi / triangleAmount))
                             );
                   }
          glEnd();
```

```
        GLfloat e=.52f; GLfloat f=.8f;

        glBegin(GL_TRIANGLE_FAN);
           glColor3ub(255, 217, 255);
                glVertex2f(e, f); // center of circle
                for(i = 0; i <= triangleAmount;i++) {
                        glVertex2f(
                        e + (radius * cos(i *  twicePi / triangleAmount)),
                           f+ (radius * sin(i * twicePi / triangleAmount))
                        );
                }
        glEnd();

        GLfloat g=.6f; GLfloat h=.82f;

        glBegin(GL_TRIANGLE_FAN);
           glColor3ub(255, 217, 255);
                glVertex2f(g, h); // center of circle
                for(i = 0; i <= triangleAmount;i++) {
                        glVertex2f(
                        g + (radius * cos(i *  twicePi / triangleAmount)),
                           h+ (radius * sin(i * twicePi / triangleAmount))
                        );
                }
        glEnd();
}
void sky()
{
   glBegin(GL_QUADS);
   glColor3ub(50,204,255);//50,204,255
        glVertex2f(-1.0f, 0.45f);
        glVertex2f(1.0f, 0.45f);
   glVertex2f(1.0f, 1.0f);
        glVertex2f(-1.0f, 1.0f);
   glEnd();
}

void backgroundtree()
{
   //pamtrees
   glBegin(GL_POLYGON);
        glColor3ub(102, 51, 0);//102, 51, 0
        glVertex2f(-0.52f,0.45f);
        glVertex2f(-0.48f, 0.45f);
        glVertex2f(-0.48f,0.62f);
   glVertex2f(-0.5f,0.64f);
   glVertex2f(-0.52f,0.62f);
   glEnd();
   glBegin(GL_POLYGON);
        glColor3ub(34,139,34);
   glVertex2f(-0.5f,0.64f);
```

```
glVertex2f(-0.52f,0.62f);
glVertex2f(-0.58f,0.6f);
        glVertex2f(-0.55f, 0.63f);
        glVertex2f(-0.58f,0.66f);
        glVertex2f(-0.52f,0.66f);
        glVertex2f(-0.52f, 0.72f);
        glVertex2f(-0.5f,0.69f);///
        glVertex2f(-0.48f,0.72f);
        glVertex2f(-0.48f, 0.66f);
        glVertex2f(-0.42f,0.66f);
        glVertex2f(-0.45f,0.63f);
        glVertex2f(-0.42f, 0.6f);
        glVertex2f(-0.48f,0.62f);
glEnd();
glBegin(GL_POLYGON);
        glColor3ub(102, 51, 0);
        glVertex2f(0.52f,0.45f);
        glVertex2f(0.48f, 0.45f);
        glVertex2f(0.48f,0.62f);
glVertex2f(0.5f,0.64f);
glVertex2f(0.52f,0.62f);
glEnd();
 glBegin(GL_POLYGON);
        glColor3ub(34,139,34);
glVertex2f(0.5f,0.64f);
glVertex2f(0.52f,0.62f);
glVertex2f(0.58f,0.6f);
        glVertex2f(0.55f, 0.63f);
        glVertex2f(0.58f,0.66f);
        glVertex2f(0.52f,0.66f);
        glVertex2f(0.52f, 0.72f);
        glVertex2f(0.5f,0.69f);///
        glVertex2f(0.48f,0.72f);
        glVertex2f(0.48f, 0.66f);
        glVertex2f(0.42f,0.66f);
        glVertex2f(0.45f,0.63f);
        glVertex2f(0.42f, 0.6f);
        glVertex2f(0.48f,0.62f);
glEnd();
        glBegin(GL_POLYGON);
        glColor3ub(34,139,34);
        glVertex2f(-1.0f,0.45f);
        glVertex2f(-0.98f, 0.5f);
glVertex2f(-0.93f, 0.56);
glVertex2f(-0.9f,0.6f);
glVertex2f(-0.82f, 0.64);
glVertex2f(-0.75f, 0.67);///
glVertex2f(-0.68f, 0.64);
 glVertex2f(-0.6f,0.6f);
 glVertex2f(-0.57f, 0.56);
 glVertex2f(-0.52f, 0.5f);
 glVertex2f(-0.5f,0.45f);
 glVertex2f(-1.0f,0.45f);
        glEnd();
        glBegin(GL_POLYGON);
        glColor3ub(34,139,34);
        glVertex2f(1.0f,0.45f);
        glVertex2f(0.98f, 0.5f);
glVertex2f(0.93f, 0.56);
glVertex2f(0.9f,0.6f);
glVertex2f(0.82f, 0.64);
glVertex2f(0.75f, 0.67);///
glVertex2f(0.68f, 0.64);
 glVertex2f(0.6f,0.6f);
 glVertex2f(0.57f, 0.56);
 glVertex2f(0.52f, 0.5f);
 glVertex2f(0.5f,0.45f);
 glVertex2f(1.0f,0.45f);
        glEnd();

        glBegin(GL_POLYGON);
        glColor3ub(34,139,34);
        glVertex2f(-0.5f,0.45f);
        glVertex2f(-0.48f, 0.5f);
glVertex2f(-0.45f, 0.56);
glVertex2f(-0.42f,0.6f);
glVertex2f(-0.37f, 0.62);///
glVertex2f(-0.32f, 0.6);
glVertex2f(-0.29f, 0.56f);
 glVertex2f(-0.27f, 0.5f);
glVertex2f(-0.25f,0.45f);
glEnd();

glBegin(GL_POLYGON);
        glColor3ub(34,139,34);
        glVertex2f(0.5f,0.45f);
        glVertex2f(0.48f, 0.5f);
glVertex2f(0.45f, 0.56);
glVertex2f(0.42f,0.6f);
glVertex2f(0.37f, 0.62);///
glVertex2f(0.32f, 0.6);
glVertex2f(0.29f, 0.56f);
 glVertex2f(0.27f, 0.5f);
glVertex2f(0.25f,0.45f);
glEnd();
glBegin(GL_POLYGON);
        glColor3ub(34,139,34);
        glVertex2f(-.25f,0.45f);
        glVertex2f(-0.23f, 0.5f);
glVertex2f(-0.18f, 0.56);
glVertex2f(-0.15f,0.6f);
glVertex2f(-0.07f, 0.64);
glVertex2f(-0.00f, 0.67);///
glVertex2f(0.07f, 0.64);
```

```
    glVertex2f(0.15f,0.6f);
    glVertex2f(0.18f, 0.56);
    glVertex2f(0.23f, 0.5f);
    glVertex2f(.25f,0.45f);
            glEnd();
}
void bird()
{
       int i;
            GLfloat mm=0.182f; GLfloat nn=.801f; GLfloat radiusmm =.01f;
            int triangleAmount = 20;
            GLfloat twicePi = 2.0f * PI;
            glBegin(GL_TRIANGLE_FAN);
                glColor3ub(225, 225, 208);
                        glVertex2f(mm, nn); // center of circle
                        for(i = 0; i <= triangleAmount;i++) {
                                    glVertex2f(
                                    mm + (radiusmm * cos(i *  twicePi / triangleAmount)),
                                        nn + (radiusmm * sin(i * twicePi / triangleAmount))
                                    );
                        }
            glEnd();
    glBegin(GL_POLYGON);
    glColor3ub(225, 225, 208 );
    glVertex2f(0.1f,0.8f);
    glVertex2f(0.11f,0.79f);
    glVertex2f(0.12f,0.78f);
    glVertex2f(0.16f,0.77f);
    glVertex2f(0.19f,0.79f);
    glVertex2f(0.201f,0.8f);
    glEnd();
    glBegin(GL_TRIANGLES);
    glColor3ub(217, 217, 217);
    glVertex2f(0.175f,0.8f);
    glVertex2f(0.15f,0.8f);
    glVertex2f(0.14f,0.84f);
    glEnd();
glBegin(GL_TRIANGLES);
    glColor3ub(242, 242, 242 );
    glVertex2f(0.175f,0.8f);
    glVertex2f(0.144f,0.8f);
    glVertex2f(0.12f,0.83f);
    glEnd();
            /////2nd bird////
            glBegin(GL_POLYGON);
    glColor3ub(225, 225, 208 );
    glVertex2f(-0.02f,0.8f);
    glVertex2f(-0.01f,0.79f);
    glVertex2f(0.0f,0.78f);
    glVertex2f(0.04f,0.77f);
    glVertex2f(0.07f,0.79f);
    glVertex2f(0.081f,0.8f);
    glEnd();
    glBegin(GL_TRIANGLES);
    glColor3ub(217, 217, 217);
    glVertex2f(0.055f,0.8f);
    glVertex2f(0.03f,0.8f);
    glVertex2f(0.02f,0.84f);
    glEnd();
    glBegin(GL_TRIANGLES);
    glColor3ub(242, 242, 242 );
    glVertex2f(0.055f,0.8f);
    glVertex2f(0.024f,0.8f);
    glVertex2f(0.0f,0.83f);
    glEnd();
        GLfloat mmm=0.062f; GLfloat nnn=.801f; GLfloat radiusmmm =.01f;
        glBegin(GL_TRIANGLE_FAN);
            glColor3ub(225, 225, 208);
                glVertex2f(mmm, nnn); // center of circle
                for(i = 0; i <= triangleAmount;i++) {
                        glVertex2f(
                        mmm + (radiusmmm * cos(i *  twicePi / triangleAmount)),
                            nnn + (radiusmmm * sin(i * twicePi / triangleAmount))
                        );
                }
        glEnd();
        }

void sun()
{
    int i;

        GLfloat x=.0f; GLfloat y=.9f; GLfloat radius =.06f;
        int triangleAmount = 20;
        GLfloat twicePi = 2.0f * PI;

        glBegin(GL_TRIANGLE_FAN);
            glColor3ub(255, 204, 0);
                glVertex2f(x, y); // center of circle
                for(i = 0; i <= triangleAmount;i++) {
                        glVertex2f(
                        x + (radius * cos(i *  twicePi / triangleAmount)),
                            y + (radius * sin(i * twicePi / triangleAmount))
                        );
                }
        glEnd();
}
void ground()
{
    glBegin(GL_POLYGON);
        glColor3ub(102, 255, 51);
        glVertex2f(-1.0f,-1.0f);
        glVertex2f(-1.0f,0.0f);
        glVertex2f(0.0f, 0.0f);
    glVertex2f(0.1f, -0.03);
    glVertex2f(0.2f, -0.07);
    glVertex2f(0.3f, -0.1);//6
    glVertex2f(0.2f, -0.13);
    glVertex2f(0.1f, -0.17);
    glVertex2f(0.2f, -0.2);
    glVertex2f(0.35f, -0.23);
    glVertex2f(0.25f, -0.25);
    glVertex2f(0.18f, -0.28);//12
    glVertex2f(0.3f, -0.32);
    glVertex2f(0.2f, -0.35);
    glVertex2f(0.4f, -0.4);
    glVertex2f(0.4f, -0.6);
    glVertex2f(0.2f, -0.65);
    glVertex2f(0.3f, -0.7);
    glVertex2f(0.2f, -0.75);
    glVertex2f(0.4f, -0.8);
    glVertex2f(0.2f, -0.85);
    glVertex2f(0.35f, -0.9);
    glVertex2f(0.25f, -0.95);
    glVertex2f(0.4f, -1.0);
    glEnd();
glBegin(GL_QUADS);
            glColor3ub(153, 153, 102);
    glVertex2f(0.17f, -0.19);
    glVertex2f(0.19f, -0.2f);
    glVertex2f(0.3f, -0.12f);
    glVertex2f(0.3f, -0.1);
    glEnd();
glBegin(GL_QUADS);
            glColor3ub(153, 153, 102);
             glVertex2f(0.23f, -0.295);
             glVertex2f(0.25f, -0.305f);
             glVertex2f(0.35f, -0.25f);
            glVertex2f(0.35f, -0.23);
    glEnd();
glBegin(GL_QUADS);
            glColor3ub(153, 153, 102);
            glVertex2f(0.3f, -0.32);
            glVertex2f(0.3f, -0.34);
    glVertex2f(0.25f, -0.365f);
    glVertex2f(0.2f, -0.35);
    glEnd();
glBegin(GL_QUADS);
            glColor3ub(153, 153, 102);
            glVertex2f(0.205f, -0.655);
            glVertex2f(0.4f, -0.6);
            glVertex2f(0.4f, -0.625);
            glVertex2f(0.25f, -0.675);
    glEnd();
glBegin(GL_QUADS);
            glColor3ub(153, 153, 102);
```

```
        glVertex2f(0.3f, -0.7);
        glVertex2f(0.3f, -0.72);
        glVertex2f(0.24f, -0.7595);
    glVertex2f(0.2f, -0.75);
    glEnd();
glBegin(GL_QUADS);
        glColor3ub(153, 153, 102);
        glVertex2f(0.4f, -0.8);
        glVertex2f(0.4f, -0.825);
        glVertex2f(0.24f, -0.865);
    glVertex2f(0.2f, -0.85);
    glEnd();
    glBegin(GL_QUADS);
        glColor3ub(153, 153, 102);
        glVertex2f(0.35f, -0.9);
        glVertex2f(0.35f, -0.925);
        glVertex2f(0.27f, -0.958);
    glVertex2f(0.25f, -0.948);
    glEnd();
}
void river()
{
   glBegin(GL_QUADS);
   glColor3ub(38, 154, 214);
        //glColor3ub(0,122,204);
        glVertex2f(-1.0f,0.45f);
        glVertex2f(1.0f,0.45f);
        glVertex2f(1.0f,-1.0f);
    glVertex2f(-1.0f,-1.0f);
    glEnd();
}
void hut()
{
    glBegin(GL_POLYGON);
    glColor3ub(204, 153, 0);
    glVertex2f(-0.5f,0.2f);
    glVertex2f(-0.1f,0.2f);
    glVertex2f(-0.18f,0.5f);
    glVertex2f(-0.58f,0.5f);
    glEnd();
     glBegin(GL_POLYGON);
    glColor3ub(204, 153, 0);
    glVertex2f(-0.49f,0.2f);
    glVertex2f(-0.13f,0.2f);
    glVertex2f(-0.13f,-0.2f);
    glVertex2f(-0.49f,-0.2f);
    glEnd();
    glBegin(GL_POLYGON);
    glColor3ub(204, 153, 0);
    glVertex2f(-0.58f,0.5f);
    glVertex2f(-0.63f,0.2f);
    glVertex2f(-0.61f,0.2f);
```

```
glVertex2f(-0.61f,-0.15f);
glVertex2f(-0.49f,-0.2f);
glVertex2f(-0.17f,-0.2f);
glEnd();
glLineWidth(2);
 glBegin(GL_LINES);
glColor3ub(0,0,0);
glVertex2f(-0.5f,0.2f);
glVertex2f(-0.1f,0.2f);
glColor3ub(0,0,0);
glVertex2f(-0.49f,0.2f);
glVertex2f(-0.49f,-0.2f);
glColor3ub(0,0,0);
 glVertex2f(-0.13f,-0.2f);
glVertex2f(-0.49f,-0.2f);
 glColor3ub(0,0,0);
 glVertex2f(-0.13f,-0.2f);
glVertex2f(-0.13f,0.2f);
glColor3ub(0,0,0);
glVertex2f(-0.5f,0.2f);
glVertex2f(-0.58f,0.5f);
glColor3ub(0,0,0);
glVertex2f(-0.58f,0.5f);
glVertex2f(-0.63f,0.2f);
glColor3ub(0,0,0);
glVertex2f(-0.57f,0.445f);
glVertex2f(-0.61f,0.2f);

glColor3ub(0,0,0);
glVertex2f(-0.61f,0.2f);
glVertex2f(-0.61f,-0.15f);

glColor3ub(0,0,0);
glVertex2f(-0.61f,-0.15f);
glVertex2f(-0.49f,-0.2f);

glColor3ub(0,0,0);
glVertex2f(-0.1f,0.2f);
glVertex2f(-0.18f,0.5f);

glColor3ub(0,0,0);
glVertex2f(-0.18f,0.5f);
glVertex2f(-0.58f,0.5f);

glEnd();

 glBegin(GL_POLYGON);
glColor3ub(153, 115, 0);
glVertex2f(-0.35f,0.1f);
glVertex2f(-0.22f,0.1f);    //main door
glVertex2f(-0.22f,-0.2f);
glVertex2f(-0.35f,-0.2f);
```

```
    glEnd();
    glLineWidth(2);
    glBegin(GL_LINES);
    glColor3ub(0,0,0);
    glVertex2f(-0.35f,0.1f);
    glVertex2f(-0.22f,0.1f);
    glVertex2f(-0.22f,0.1f);
    glVertex2f(-0.22f,-0.2f);
    glVertex2f(-0.35f,0.1f);
    glVertex2f(-0.35f,-0.2f);
    glVertex2f(-0.285f,0.1f);
    glVertex2f(-0.285f,-0.2f);
    glEnd();

   glBegin(GL_POLYGON);
    glColor3ub(153, 115, 0);
    glVertex2f(-0.44f,0.05f);
    glVertex2f(-0.38f,0.05f); //left window
    glVertex2f(-0.38f,-0.05f);
    glVertex2f(-0.44f,-0.05f);
    glEnd();
     glLineWidth(2);
    glBegin(GL_LINES);
    glColor3ub(0,0,0);
    glVertex2f(-0.44f,0.05f);
    glVertex2f(-0.44f,-0.05f);
    glVertex2f(-0.38f,0.05f);
    glVertex2f(-0.38f,-0.05f);
    glVertex2f(-0.44f,0.05f);
    glVertex2f(-0.38f,0.05f);
    glVertex2f(-0.44f,-0.05f);
    glVertex2f(-0.38f,-0.05f);
    glVertex2f(-0.41f,0.05f);
    glVertex2f(-0.41f,-0.05f);
    glEnd();

glBegin(GL_POLYGON);
    glColor3ub(153, 115, 0);
     glVertex2f(-0.51f,0.12f); //2nd door
    glVertex2f(-0.58f,0.14f);
    glVertex2f(-0.58f,-0.17f);
    glVertex2f(-0.51f,-0.2f);
    glEnd();
    glLineWidth(2);
    glBegin(GL_LINES);
    glColor3ub(0,0,0);
    glVertex2f(-0.58f,0.14f);
    glVertex2f(-0.58f,-0.17f);
    glVertex2f(-0.51f,0.12f);
    glVertex2f(-0.51f,-0.2f);
    glVertex2f(-0.58f,0.14f);
    glVertex2f(-0.51f,0.12f);
    glVertex2f(-0.545f,0.13f);
    glVertex2f(-0.545f,-0.185f);
    glEnd();

     glBegin(GL_POLYGON);
    glColor3ub(153, 115, 0);
    glVertex2f(-0.14f,0.05f);
    glVertex2f(-0.2f,0.05f);   //rightwindow
    glVertex2f(-0.2f,-0.05f);
    glVertex2f(-0.14f,-0.05f);
    glEnd();
     glLineWidth(2);
    glBegin(GL_LINES);
    glColor3ub(0,0,0);
    glVertex2f(-0.14f,0.05f);
    glVertex2f(-0.14f,-0.05f);
    glVertex2f(-0.2f,0.05f);
    glVertex2f(-0.2f,-0.05f);
    glVertex2f(-0.14f,0.05f);
    glVertex2f(-0.2f,0.05f);
    glVertex2f(-0.14f,-0.05f);
    glVertex2f(-0.2f,-0.05f);
    glVertex2f(-0.17f,0.05f);
    glVertex2f(-0.17f,-0.05f);
    glEnd();
}

void tree()
{
    glBegin(GL_POLYGON);
    glColor3ub(102, 51, 0);
    glVertex2f(-0.72f,-0.15f);
    glVertex2f(-0.65f,-0.2f);
    glVertex2f(-0.735f,-0.17f);
    glVertex2f(-0.74f,-0.25f);
    glVertex2f(-0.775f,-0.17f);
    glVertex2f(-0.85f,-0.2f);
    glVertex2f(-0.78f,-0.15f);
    //glVertex2f(-0.7f,-0.25f);
    glEnd();

    glBegin(GL_QUADS);
    glColor3ub(102, 51, 0);
    glVertex2f(-0.78f,-0.15f);
    glVertex2f(-0.78f,0.23f);
    glVertex2f(-0.72f,0.23f);
    glVertex2f(-0.72f,-0.15f);
    glEnd();
    glBegin(GL_QUADS);
    glColor3ub(102, 51, 0);
    glVertex2f(-0.76f,0.23f);
    glVertex2f(-0.76f,0.3f);
    glVertex2f(-0.74f,0.3f);
    glVertex2f(-0.74f,0.23f);
    glEnd();
    glBegin(GL_QUADS);
    glColor3ub(102, 51, 0);
    glVertex2f(-0.74f,0.23f);
    glVertex2f(-0.71f,0.29f);
    glVertex2f(-0.7f,0.28f);
    glVertex2f(-0.72f,0.23f);
    glEnd();
    glBegin(GL_QUADS);
    glColor3ub(102, 51, 0);
    glVertex2f(-0.78f,0.23f);
    glVertex2f(-0.8f,0.28f);
    glVertex2f(-0.79f,0.29f);
    glVertex2f(-0.76f,0.23f);
    glEnd();

     int i;

        GLfloat x=-.75f; GLfloat y=.33f; GLfloat radius =.06f;
        int triangleAmount = 20;
        GLfloat twicePi = 2.0f * PI;

        glBegin(GL_TRIANGLE_FAN);
            glColor3ub(51, 204, 51);
                glVertex2f(x, y); // center of circle
                for(i = 0; i <= triangleAmount;i++) {
                        glVertex2f(
                        x + (radius * cos(i *  twicePi / triangleAmount)),
                            y + (radius * sin(i * twicePi / triangleAmount))
                        );
                }
        glEnd();

GLfloat a=-.68f; GLfloat b=.31f;

        glBegin(GL_TRIANGLE_FAN);
            glColor3ub(51, 204, 51);
                glVertex2f(a, b); // center of circle
                for(i = 0; i <= triangleAmount;i++) {
                        glVertex2f(
                        a + (radius * cos(i *  twicePi / triangleAmount)),
                            b + (radius * sin(i * twicePi / triangleAmount))
                        );
                }
        glEnd();

        GLfloat c=-.81f; GLfloat d=.31f;

        glBegin(GL_TRIANGLE_FAN);
            glColor3ub(51, 204, 51);
            glVertex2f(c, d); // center of circle
            for(i = 0; i <= triangleAmount;i++) {
                        glVertex2f(
                        c + (radius * cos(i *  twicePi / triangleAmount)),
                            d + (radius * sin(i * twicePi / triangleAmount))
                        );
            }
glEnd();

GLfloat e=-.87f; GLfloat f=.35f;

glBegin(GL_TRIANGLE_FAN);
    glColor3ub(51, 204, 51);
            glVertex2f(e, f); // center of circle
            for(i = 0; i <= triangleAmount;i++) {
                        glVertex2f(
                        e + (radius * cos(i *  twicePi / triangleAmount)),
                            f+ (radius * sin(i * twicePi / triangleAmount))
                        );
            }
glEnd();

GLfloat g=-.61f; GLfloat h=.35f;

glBegin(GL_TRIANGLE_FAN);
    glColor3ub(51, 204, 51);
            glVertex2f(g, h); // center of circle
            for(i = 0; i <= triangleAmount;i++) {
                        glVertex2f(
                        g + (radius * cos(i *  twicePi / triangleAmount)),
                            h+ (radius * sin(i * twicePi / triangleAmount))
                        );
            }
glEnd();
GLfloat a1=-.61f; GLfloat b1=.4f;

glBegin(GL_TRIANGLE_FAN);
    glColor3ub(51, 204, 51);
            glVertex2f(a1, b1); // center of circle
            for(i = 0; i <= triangleAmount;i++) {
                        glVertex2f(
                        a1 + (radius * cos(i *  twicePi / triangleAmount)),
                            b1 + (radius * sin(i * twicePi / triangleAmount))
                        );
            }
glEnd();

GLfloat c1=-.88f; GLfloat d1=.4f;

glBegin(GL_TRIANGLE_FAN);
    glColor3ub(51, 204, 51);
            glVertex2f(c, d); // center of circle
```

```
        for(i = 0; i <= triangleAmount;i++) {
                glVertex2f(
                c1 + (radius * cos(i *  twicePi / triangleAmount)),
                   d1 + (radius * sin(i * twicePi / triangleAmount))
                );
        }
glEnd();

GLfloat e1=-.87f; GLfloat f1=.44f;

glBegin(GL_TRIANGLE_FAN);
   glColor3ub(51, 204, 51);
        glVertex2f(e1, f1); // center of circle
        for(i = 0; i <= triangleAmount;i++) {
                glVertex2f(
                e1 + (radius * cos(i *  twicePi / triangleAmount)),
                   f1+ (radius * sin(i * twicePi / triangleAmount))
                );
        }
glEnd();

GLfloat g1=-.61f; GLfloat h1=.4f;

glBegin(GL_TRIANGLE_FAN);
   glColor3ub(51, 204, 51);
        glVertex2f(g, h); // center of circle
        for(i = 0; i <= triangleAmount;i++) {
                glVertex2f(
                g1 + (radius * cos(i *  twicePi / triangleAmount)),
                   h1+ (radius * sin(i * twicePi / triangleAmount))
                );
        }
glEnd();
GLfloat e11=-.64f; GLfloat f11=.44f;

glBegin(GL_TRIANGLE_FAN);
   glColor3ub(51, 204, 51);
        glVertex2f(e11, f11); // center of circle
        for(i = 0; i <= triangleAmount;i++) {
                glVertex2f(
                e11 + (radius * cos(i *  twicePi / triangleAmount)),
                   f11+ (radius * sin(i * twicePi / triangleAmount))
                );
        }
glEnd();

GLfloat e12=-.75f; GLfloat f12=.44f;

glBegin(GL_TRIANGLE_FAN);
   glColor3ub(51, 204, 51);
        glVertex2f(e12, f12); // center of circle
        for(i = 0; i <= triangleAmount;i++) {
```

```
                        glVertex2f(
                        e12 + (radius * cos(i *  twicePi / triangleAmount)),
                           f12+ (radius * sin(i * twicePi / triangleAmount))
                        );
                }
        glEnd();
         glBegin(GL_QUADS);
   glColor3ub(51, 204, 51);
   glVertex2f(-0.85f,0.33f);
   glVertex2f(-0.85f,0.44f);
   glVertex2f(-0.65f,0.44f);
   glVertex2f(-0.65f,0.33f);
   glEnd();

   GLfloat e123=-.8f; GLfloat f123=.5f;

        glBegin(GL_TRIANGLE_FAN);
           glColor3ub(51, 204, 51);
                glVertex2f(e123, f123); // center of circle
                for(i = 0; i <= triangleAmount;i++) {
                        glVertex2f(
                        e123 + (radius * cos(i *  twicePi / triangleAmount)),
                           f123+ (radius * sin(i * twicePi / triangleAmount))
                        );
                }
        glEnd();
        GLfloat e1232=-.7f; GLfloat f1232=.5f;

        glBegin(GL_TRIANGLE_FAN);
           glColor3ub(51, 204, 51);
                glVertex2f(e1232, f1232); // center of circle
                for(i = 0; i <= triangleAmount;i++) {
                        glVertex2f(
                        e1232 + (radius * cos(i *  twicePi / triangleAmount)),
                           f1232+ (radius * sin(i * twicePi / triangleAmount))
                        );
                }
        glEnd();
}

void boat()
{
   glBegin(GL_POLYGON);
        glColor3ub(0,0,0);
        glVertex2f(-0.2f, 0.4f);
        glVertex2f(-0.15f, 0.35f);
        glVertex2f(0.15f, 0.35f);
   glVertex2f(0.2f, 0.4f);
   glEnd();

        glBegin(GL_POLYGON);
        glColor3ub(255, 153, 0);
```

```
            glVertex2f(-0.13f, 0.4f);
            glVertex2f(-0.11f,0.48f);
            glVertex2f(-0.088f, 0.52f);
            glVertex2f(0.13f, 0.52f);
            glVertex2f(0.14f, 0.49f);
    glVertex2f(0.15f, 0.4f);
    glEnd();

            glBegin(GL_POLYGON);
            glColor3ub(255,25,25);
            glVertex2f(-0.038f, 0.57f);
            glVertex2f(-0.038f, 0.73f);
            glVertex2f(-0.035f, 0.75f);
            glVertex2f(0.064f, 0.73f);
            glVertex2f(0.065f, 0.71f);
    glVertex2f(0.065f, 0.55f);
    glEnd();

    glBegin(GL_POLYGON);
            glColor3ub(136,204,0);
            glVertex2f(0.0f, 0.52f);
            glVertex2f(0.0f, 0.79f);
            glVertex2f(0.01f, 0.79f);
            glVertex2f(0.01f, 0.52f);
            glEnd();
}
void way()
{
    glBegin(GL_QUADS);
    glColor3ub(153, 153, 102);
    glVertex2f(-0.35f,-0.2f);
    glVertex2f(-0.22f,-0.2f);
    glVertex2f(-0.28f,-0.5f);
    glVertex2f(-0.43f,-0.5f);
    glEnd();

    glBegin(GL_QUADS);
    glColor3ub(153, 153, 102);
    glVertex2f(-0.43f,-0.5f);
    glVertex2f(-0.75f,-1.0f);
    glVertex2f(-0.56f,-1.0f);
    glVertex2f(-0.28f,-0.5f);
    glEnd();


}
void grass1()
{
     glLineWidth(4);
    glBegin(GL_LINES);
    glColor3ub(0, 102, 0);
            glVertex2f(-.05f, -0.35f);
            glVertex2f(-0.0f, -0.4f);//
            glVertex2f(0.05f, -0.35f);
            glVertex2f(0.0f, -0.4f);//
            glVertex2f(0.027f, -0.33f);
            glVertex2f(0.0f, -0.4f);//
            glVertex2f(-0.027f, -0.33f);
            glVertex2f(0.0f, -0.4f);//
            glVertex2f(0.0f, -0.3f);
            glVertex2f(0.0f, -0.4f);//
            glVertex2f(-0.075f, -0.37f);
            glVertex2f(-0.0f, -0.4f);//
            glVertex2f(0.0745f, -0.37f);
            glVertex2f(-0.0f, -0.4f);//
    glEnd();
    int i;
    int triangleAmount = 20;
GLfloat twicePi = 2.0f * PI;


            GLfloat e=-.05f; GLfloat f=-.35f; GLfloat radius11 =.02f;

            glBegin(GL_TRIANGLE_FAN);
                glColor3ub(255, 51, 0);
                        glVertex2f(e, f); // center of circle
                        for(i = 0; i <= triangleAmount;i++) {
                                    glVertex2f(
                                    e + (radius11 * cos(i *  twicePi / triangleAmount)),
                                        f+ (radius11 * sin(i * twicePi / triangleAmount))
                                    );
                        }
            glEnd();

            GLfloat g=0.05f; GLfloat h=-0.35f;
glBegin(GL_TRIANGLE_FAN);
                glColor3ub(255, 102, 0);
                        glVertex2f(g, h); // center of circle
                        for(i = 0; i <= triangleAmount;i++) {
                                    glVertex2f(
                                    g + (radius11 * cos(i *  twicePi / triangleAmount)),
                                        h+ (radius11 * sin(i * twicePi / triangleAmount))
                                    );
                        }
            glEnd();
GLfloat a1=0.0f; GLfloat b1=-0.3f;

            glBegin(GL_TRIANGLE_FAN);
                glColor3ub(255, 255, 0);
                        glVertex2f(a1, b1); // center of circle
                        for(i = 0; i <= triangleAmount;i++) {
                                    glVertex2f(
                                    a1 + (radius11 * cos(i *  twicePi / triangleAmount)),
                                        b1 + (radius11 * sin(i * twicePi / triangleAmount))
                        );
                }
        glEnd();
}
void fence()
{
   glLineWidth(4);
   glBegin(GL_LINES);
  glColor3ub(255, 255, 102);
  glVertex2f(-1.0f,-0.1f);
  glVertex2f(-0.6f,-0.1f);

  glColor3ub(255, 255, 102);
  glVertex2f(-1.0f,-0.05f);
  glVertex2f(-0.6f,-0.05f);

  glColor3ub(255, 255, 102);
  glVertex2f(-1.0f,0.0f);
  glVertex2f(-0.6f,0.0f);

   glColor3ub(255, 255, 102);
  glVertex2f(-1.0f,0.05f);
  glVertex2f(-0.6f,0.05f);

   glColor3ub(255, 255, 102);
  glVertex2f(-1.0f,0.1f);
  glVertex2f(-0.6f,0.1f);

   glColor3ub(255, 255, 102);
  glVertex2f(-0.95f,0.13f);
  glVertex2f(-0.95f,-0.12f);

  glColor3ub(255, 255, 102);
  glVertex2f(-0.9f,0.13f);
  glVertex2f(-0.9f,-0.12f);

  glColor3ub(255, 255, 102);
  glVertex2f(-0.85f,0.13f);
  glVertex2f(-0.85f,-0.12f);

  glColor3ub(255, 255, 102);
  glVertex2f(-0.8f,0.13f);
  glVertex2f(-0.8f,-0.12f);

  glColor3ub(255, 255, 102);
  glVertex2f(-0.75f,0.13f);
  glVertex2f(-0.75f,-0.12f);

  glColor3ub(255, 255, 102);
  glVertex2f(-0.7f,0.13f);
  glVertex2f(-0.7f,-0.12f);
```

```
   glColor3ub(255, 255, 102);
   glVertex2f(-0.65f,0.13f);
   glVertex2f(-0.65f,-0.12f);

   glEnd();
}
void grass6()
{

   int i;
   int triangleAmount = 20;
GLfloat twicePi = 2.0f * PI;
         GLfloat e=-.85f; GLfloat f=-.75f; GLfloat radius11 =.02f;

         glBegin(GL_TRIANGLE_FAN);
            glColor3ub(255, 51, 0);
                  glVertex2f(e, f); // center of circle
                  for(i = 0; i <= triangleAmount;i++) {
                           glVertex2f(
                           e + (radius11 * cos(i *  twicePi / triangleAmount)),
                              f+ (radius11 * sin(i * twicePi / triangleAmount))
                           );
                  }
         glEnd();

         GLfloat g=-0.75f; GLfloat h=-0.75f;
glBegin(GL_TRIANGLE_FAN);
            glColor3ub(255, 255, 0);
                  glVertex2f(g, h); // center of circle
                  for(i = 0; i <= triangleAmount;i++) {
                           glVertex2f(
                           g + (radius11 * cos(i *  twicePi / triangleAmount)),
                              h+ (radius11 * sin(i * twicePi / triangleAmount))
                           );
                  }
         glEnd();
GLfloat a1=-0.8f; GLfloat b1=-0.7f;

         glBegin(GL_TRIANGLE_FAN);
            glColor3ub(255, 102, 0);
                  glVertex2f(a1, b1); // center of circle
                  for(i = 0; i <= triangleAmount;i++) {
                           glVertex2f(
                           a1 + (radius11 * cos(i *  twicePi / triangleAmount)),
                              b1 + (radius11 * sin(i * twicePi / triangleAmount))
                           );
                  }
         glEnd();
         glLineWidth(4);
   glBegin(GL_LINES);
   glColor3ub(0, 102, 0);
         glVertex2f(-.85f, -0.75f);
```

```
            glVertex2f(-0.8f, -0.8f);
            glVertex2f(-0.75f, -0.75f);
            glVertex2f(-0.8f, -0.8f);
            glVertex2f(-0.827f, -0.72f);
            glVertex2f(-0.8f, -0.8f);
            glVertex2f(-0.773f, -0.72f);
            glVertex2f(-0.8f, -0.8f);
            glVertex2f(-0.8f, -0.7f);
            glVertex2f(-0.8f, -0.8f);
            glVertex2f(-0.725f, -0.77f);
            glVertex2f(-0.8f, -0.8f);
            glVertex2f(-0.875f, -0.77f);
            glVertex2f(-0.8f, -0.8f);
    glEnd();
}
void well1()
{

     glLineWidth(5);
     glBegin(GL_LINES);
    glColor3ub(204, 51, 0);//204, 51, 0
     glVertex2f(-0.9f,-0.35f);
    glVertex2f(-0.85f,-0.33f);//
    glVertex2f(-0.85f,-0.33f);
    glVertex2f(-0.8f,-0.325f);//
    glVertex2f(-0.8f,-0.325f);
    glVertex2f(-0.7f,-0.325f);//
    glVertex2f(-0.7f,-0.325f);
    glVertex2f(-0.65f,-0.33f);//
     glVertex2f(-0.65f,-0.33f);
     glVertex2f(-0.6f,-0.35f);
    glEnd();

    glLineWidth(3);
     glBegin(GL_LINES);
    glColor3ub(194, 194, 163);
    glVertex2f(-0.59f,-0.43f);
    glVertex2f(-0.57f,-0.39f);//
    glVertex2f(-0.57f,-0.39f);
    glVertex2f(-0.55f,-0.39f);//
    glVertex2f(-0.55f,-0.39f);
    glVertex2f(-0.52f,-0.39f);//
    glVertex2f(-0.52f,-0.39f);
    glVertex2f(-0.5f,-0.43f);//
    glEnd();

    glLineWidth(2.5);
     glBegin(GL_LINES);
    glColor3ub(230, 172, 0);
    glVertex2f(-0.545f,-0.385f);
    glVertex2f(-0.57f,-0.45f);//
    glVertex2f(-0.57f,-0.45f);
```

```
glVertex2f(-0.575f,-0.5f);//
glVertex2f(-0.575f,-0.5f);
glVertex2f(-0.58f,-0.53f);//
glVertex2f(-0.58f,-0.53f);
glVertex2f(-0.57f,-0.55f);//
glVertex2f(-0.57f,-0.55f);
glVertex2f(-0.48f,-0.53f);//
glEnd();

glBegin(GL_POLYGON);
glColor3ub(38, 154, 214);
glVertex2f(-0.585f,-0.43f);
glVertex2f(-0.568f,-0.44f);
glVertex2f(-0.528f,-0.44f);
glVertex2f(-0.505f,-0.43f);
glVertex2f(-0.528f,-0.425f);
glVertex2f(-0.57f,-0.425f);
glEnd();

glBegin(GL_POLYGON);
glColor3ub(204, 51, 0);
glVertex2f(-0.9f,-0.35f);
glVertex2f(-0.9f,-0.55f);
glVertex2f(-0.85f,-0.575f);
glVertex2f(-0.8f,-0.59f);
glVertex2f(-0.7f,-0.59f);
glVertex2f(-0.65f,-0.575f);
glVertex2f(-0.6f,-0.55f);
glVertex2f(-0.6f,-0.35f);
glEnd();

glBegin(GL_POLYGON);
glColor3ub(255, 102, 51);
glVertex2f(-0.9f,-0.35f);
glVertex2f(-0.85f,-0.375f);
glVertex2f(-0.8f,-0.38f);
glVertex2f(-0.7f,-0.38f);
glVertex2f(-0.65f,-0.375f);
glVertex2f(-0.6f,-0.35f);
glVertex2f(-0.65f,-0.33f);
glVertex2f(-0.7f,-0.325f);
 glVertex2f(-0.8f,-0.325f);
 glVertex2f(-0.85f,-0.33f);
glEnd();
glBegin(GL_POLYGON);
glColor3ub(194, 194, 163);
glVertex2f(-0.59f,-0.43f);
glVertex2f(-0.57f,-0.5f);
glVertex2f(-0.52f,-0.5f);
glVertex2f(-0.5f,-0.43f);
glVertex2f(-0.52f,-0.42f);
glVertex2f(-0.57f,-0.42f);
```

```
    glEnd();
}
void DrawSphere()
{
    glColorMaterial ( GL_FRONT_AND_BACK, GL_AMBIENT_AND_DIFFUSE ) ;
    glEnable ( GL_COLOR_MATERIAL ) ;
    glColor4f(1.0f, 0.0f, 0.0f, 0.0f);
    glClear(GL_COLOR_BUFFER_BIT | GL_DEPTH_BUFFER_BIT);
    glEnable(GL_DEPTH_TEST);
    glLoadIdentity();
    glPushMatrix();
    glTranslatef(position2,0.0f, 0.0f);
    cloud1();
    glPopMatrix();
   // boat();
    //hut1();
    tree();
    backgroundtree();
    fence();
    grass6();
    grass6();
    grass6();
    way();
    well1();
    ground();
    river();
    glPushMatrix();
    glTranslatef(0.0f,position4, 0.0f);
    sun();
    glPopMatrix();
    glFlush();

}
struct Point
{
    float x, y;
    unsigned char r, g, b, a;
};
std::vector< Point > points;
void display() {
        glClearColor(0.0f, 0.0f, 0.0f, 1.0f);
        glClear(GL_COLOR_BUFFER_BIT);
        glMatrixMode(GL_PROJECTION);
        glLoadIdentity();
        glLineWidth(2);
    sky();
    sun();
    glPushMatrix();
    glTranslatef(position2,0.0f, 0.0f);
    cloud1();
    glPopMatrix();
    glPushMatrix();
    glTranslatef(position22,0.0f, 0.0f);
    bird();
    glPopMatrix();
    backgroundtree();
    river();
    glPushMatrix();
    glTranslatef(position1,0.0f, 0.0f);
    boat();
    glPopMatrix();
    ground();
    grass1();
    grass6();
    grass6();
    way();
   // boat();
    fence();
    tree();
    well1();
    hut();
        glFlush();
        glutSwapBuffers();
}
void reshape(int w, int h)
{
    float aspectRatio = (float)w/(float)h;
        glMatrixMode(GL_PROJECTION);
    glLoadIdentity();
        gluPerspective(145, aspectRatio, 1.0, 100.0);
    glMatrixMode(GL_MODELVIEW);

}
void Display(void)
{
    glClear(GL_COLOR_BUFFER_BIT | GL_DEPTH_BUFFER_BIT);
    glLoadIdentity();
    glTranslatef(0,0,-20);
    glFlush();
    glutSwapBuffers();
}
void init(void)
{
    glClearColor( 1.0f, 1.0f, 1.0f, 1.0f);
    glClearDepth( 1.0 );
    glEnable(GL_DEPTH_TEST);
    glEnable(GL_LIGHTING);
    glShadeModel(GL_SMOOTH);
    glEnable(GL_COLOR_MATERIAL);
    glColorMaterial(GL_FRONT, GL_AMBIENT_AND_DIFFUSE);
    glEnable(GL_LIGHT0);
}
void handleKeypress(unsigned char key, int x, int y) {
        switch (key) {
            case  'd':
                glutDestroyWindow(1);
                glutInitWindowSize(1240, 750);
        glutInitWindowPosition((glutGet(GLUT_SCREEN_WIDTH)-1240)/2,
(glutGet(GLUT_SCREEN_HEIGHT)-750)/2);
        glutCreateWindow("village scenario");
        glutKeyboardFunc(handleKeypress);
        glutDisplayFunc(display);

        break;

        case  'D':
                glutDestroyWindow(1);
                glutInitWindowSize(1240, 750);
        glutInitWindowPosition((glutGet(GLUT_SCREEN_WIDTH)-1240)/2,
(glutGet(GLUT_SCREEN_HEIGHT)-750)/2);
        glutCreateWindow("village scenario");
        glutKeyboardFunc(handleKeypress);
        glutDisplayFunc(display);

        break;
    }
}
int main(int argc, char** argv)
{
    glutInit(&argc, argv);
    glutInitWindowSize(1240, 750);
    glutInitWindowPosition((glutGet(GLUT_SCREEN_WIDTH)-1240)/2,
(glutGet(GLUT_SCREEN_HEIGHT)-750)/2);
    glutCreateWindow("village scenario");
    init();
    glutReshapeFunc(reshape);
    glutDisplayFunc(Display);
     for( size_t i = 0; i < 1000; ++i )
    {
        Point pt;
        pt.x = -50 + (rand() % 100);
        pt.y = -50 + (rand() % 100);
        pt.r = 255;
        pt.g = 255;
        pt.b = 255;
        pt.a = 255;
        points.push_back(pt);
    }
    glutTimerFunc(100, cloud, 0);
    glutTimerFunc(100, sunn, 0);
    glutTimerFunc(100, boat, 0);
    glutTimerFunc(100, birdd, 0);
    glutKeyboardFunc(handleKeypress);
    init();
    glutMainLoop();
    return 0;
```

```
}
```