Computer Graphics (UCS505)

Project Name: - 2D Village Scenario

Branch

B.E. 3rd Year CSE

Submitted By –

Kunal (102216002)

Anant Vishwakarma (102216012)

Deepankar Sharma (102396007)

Submitted To –

Dr. Archana Kumari

Computer Science and Engineering Department

Thapar Institute of Engineering and Technology

Patiala – 147001

Introduction

The *2D Village Scenario* project is a visually engaging computer graphics application developed using **C++** and the **OpenGL graphics library**. This project aims to demonstrate the fundamental principles of 2D graphics rendering through the creation of a vibrant village scene. It showcases various static and dynamic graphical elements such as houses, trees, windmill, the sun, and moving clouds, simulating a day-night environment in a rural setting. By incorporating animation and transformation functions available in OpenGL, the project not only enhances visual appeal but also strengthens understanding of essential graphics concepts like translation, rotation, and scaling.

Project Overview

The *2D Village Scenario* project is a graphical simulation created using **C++** and the **OpenGL** library, designed to depict a peaceful village environment through animated 2D visuals. The scene typically includes elements such as houses, trees, windmill, mountains, rivers, clouds, and a sun or moon, offering a natural and immersive rural landscape.

The program utilizes basic 2D primitives (like lines, circles, and polygons) and applies OpenGL functions to animate elements — for example, moving clouds, moving windmill, and transitioning between day and night. This project emphasizes the use of coordinate systems, color handling, and transformation operations (like translation and rotation) to bring a static scene to life.

Serving as an entry-level graphics project, it is both educational and creative, giving learners a hands-on experience in scene composition, graphical transformations, and animation in OpenGL.

Scope of the Project

The *2D Village Scenario* project offers a practical introduction to the world of computer graphics and serves as a foundational step toward more advanced graphical applications. The scope of this project includes:

- **Educational Purpose**: Designed to help students and beginners understand core computer graphics concepts such as object modeling, coordinate systems, and 2D transformations.
- **Graphics Programming Practice**: Provides hands-on experience with **OpenGL**, one of the most widely used graphics APIs, and **C++**, enhancing programming and problem-solving skills.
- **Scene Rendering**: Demonstrates how to construct and render a complete 2D environment, including both static (houses, trees) and dynamic (sun, clouds, windmill) elements.
- **Animation and Interactivity**: Introduces basic animation techniques, allowing elements like the sun and clouds to move, and sets the groundwork for future additions such as user interaction or sound integration.
- **Project Expansion**: The current 2D scene can be further extended into a 3D environment or en hanced with interactive elements like character movement, weather effects, or real-time changes using keyboard/mouse input.

User Defined Functions

S No.	Function Name	Function Description
1	UP Arrow Key	Switches to day scene (drawScene)
2	DOWN Arrow Key	Switches to night scene (drawScene2)
3	‘s' key	Stops boat animations (sets speed to 0)
4	'r' key	Resumes animations (sets speed to 0.02)
5	Left Mouse Button	Increases animation speed by 0.05
6	Right Mouse Button	Decreases animation speed by 0.05

Code Snippets

```
#include<windows.h>
#include<GL\glut.h>
#include <GL/glu.h>
#include<math.h>
#include <stdlib.h>
#include<stdio.h>
#define PI 3.1416
using namespace std;



float _moveA = 0.0f;
float _moveB = 0.0f;
float _moveC = 0.0f;
float _angle1 = 0.0f;
float speed = 0.02f;
void drawScene() { // morning
    glClearColor(1.0f, 1.0f, 1.0f, 1.0f); //white background when screen is cleared
    glClear(GL_COLOR_BUFFER_BIT);
    glColor3d(1, 0, 0);// default color unless changed
    glLoadIdentity(); //Clear any previous Transformation that might be saved
    gluOrtho2D(-12, 38, -19, 14);//range of the window
    glMatrixMode(GL_MODELVIEW);  //tells to prepare for transformations



    ///sky///
    glBegin(GL_POLYGON);// Draw a Red 1x1 Square centered at origin
    glColor3ub(135, 206, 250);
    glVertex2f(38.0f, 3.0f);
    glVertex2f(38.0f, 14.0f);
    glVertex2f(-12.0f, 14.0f);
    glVertex2f(-12.0f, 3.0f);
```

```
    glEnd();


    ///sun///
    glPushMatrix();
    glColor3d(255, 255, 0);
    glTranslatef(30.0, 7.0, 0);  // cordiantes are fixed so not moving but is needed for glutSolidSphere function as it is by default at orgin
    glutSolidSphere(1.0, 250, 250);
    glPopMatrix();


    ///cloud 1///
    glPushMatrix();


    glTranslatef(_moveC, 0.0f, 0.0f);  //responsible for cloud movement
    glPushMatrix();
    glColor3d(255, 255, 255);
    glTranslatef(1.0, 6.0, 0);
    glutSolidSphere(0.7, 250, 250);
    glPopMatrix();


    glPushMatrix();
    glColor3d(255, 255, 255);
    glTranslatef(1.0, 7.0, 0);
    glutSolidSphere(0.7, 250, 250);
    glPopMatrix();


    glPushMatrix();
    glColor3d(255, 255, 255);
    glTranslatef(2.0, 7.0, 0);
    glutSolidSphere(0.7, 250, 250);
    glPopMatrix();


    glPushMatrix();
    glColor3d(255, 255, 255);
    glTranslatef(2.0, 6.0, 0);
```

```
glutSolidSphere(0.7, 250, 250);
glPopMatrix();

glPushMatrix();
glColor3d(255, 255, 255);
glTranslatef(0.0, 6.5, 0);
glutSolidSphere(0.7, 250, 250);
glPopMatrix();

glPushMatrix();
glColor3d(255, 255, 255);
glTranslatef(3.0, 6.5, 0);
glutSolidSphere(0.7, 250, 250);
glPopMatrix();

glPopMatrix();
```

///cloud 2///

```
glPushMatrix();

glTranslatef(_moveC, 0.0f, 0.0f);
glPushMatrix();
glColor3d(255, 255, 255);
glTranslatef(15.0, 9.0, 0);
glutSolidSphere(0.8, 250, 250);
glPopMatrix();

glPushMatrix();
glColor3d(255, 255, 255);
```

```
glTranslatef(15.0, 8.0, 0);
glutSolidSphere(0.8, 250, 250);
glPopMatrix();

glPushMatrix();
glColor3d(255, 255, 255);
glTranslatef(16.0, 8.0, 0);
glutSolidSphere(0.8, 250, 250);
glPopMatrix();

glPushMatrix();
glColor3d(255, 255, 255);
glTranslatef(16.0, 9.0, 0);
glutSolidSphere(0.8, 250, 250);
glPopMatrix();

glPushMatrix();
glColor3d(255, 255, 255);
glTranslatef(14.0, 8.5, 0);
glutSolidSphere(0.8, 250, 250);
glPopMatrix();

glPushMatrix();
glColor3d(255, 255, 255);
glTranslatef(17.0, 8.5, 0);
glutSolidSphere(0.8, 250, 250);
glPopMatrix();

glPopMatrix();

///cloud 3///

glPushMatrix();
```

```
glTranslatef(_moveC, 0.0f, 0.0f);

glPushMatrix();
glColor3d(255, 255, 255);
glTranslatef(22.0, 8.0, 0);
glutSolidSphere(0.75, 250, 250);
glPopMatrix();

glPushMatrix();
glColor3d(255, 255, 255);
glTranslatef(22.0, 7.0, 0);
glutSolidSphere(0.75, 250, 250);
glPopMatrix();

glPushMatrix();
glColor3d(255, 255, 255);
glTranslatef(23.0, 8.0, 0);
glutSolidSphere(0.75, 250, 250);
glPopMatrix();

glPushMatrix();
glColor3d(255, 255, 255);
glTranslatef(23.0, 7.0, 0);
glutSolidSphere(0.75, 250, 250);
glPopMatrix();

glPushMatrix();
glColor3d(255, 255, 255);
glTranslatef(24.0, 7.5, 0);
glutSolidSphere(0.75, 250, 250);
glPopMatrix();

glPushMatrix();
glColor3d(255, 255, 255);
```

```
glTranslatef(21.0, 7.5, 0);
glutSolidSphere(0.75, 250, 250);
glPopMatrix();

glPopMatrix();



///sea portion///
glBegin(GL_POLYGON);
glColor3ub(65, 105, 225);
glVertex2f(38.0f, -19.0f);
glVertex2f(38.0f, 3.0f);
glVertex2f(-12.0f, 3.0f);
glVertex2f(-12.0f, -19.0f);

glEnd();

///left soil portion///
//polygon 1
glBegin(GL_POLYGON);
glColor3ub(0, 128, 0);
glVertex2f(4.0f, 0.0f);
glVertex2f(6.0f, 1.0f);
glVertex2f(3.0f, 3.0f);
glVertex2f(-12.0f, 3.0f);
glVertex2f(-12.0f, 0.0f);

glEnd();

//polygon 2
glBegin(GL_POLYGON);// Draw a Red 1x1 Square centered at origin
glColor3ub(0, 128, 0);
glVertex2f(5.0f, -3.0f);
glVertex2f(7.0f, -1.5f);
```

```
glVertex2f(4.0f, 0.0f);
glVertex2f(-12.0f, 0.0f);
glVertex2f(-12.0f, -3.0f);

glEnd();

//polygon 3
glBegin(GL_POLYGON);// Draw a Red 1x1 Square centered at origin
glColor3ub(0, 128, 0);
glVertex2f(6.0f, -4.5f);
glVertex2f(8.0f, -4.0f);
glVertex2f(5.0f, -3.0f);
glVertex2f(-12.0f, -3.0f);
glVertex2f(-12.0f, -4.5f);

glEnd();

//polygon 4
glBegin(GL_POLYGON);// Draw a Red 1x1 Square centered at origin
glColor3ub(0, 128, 0);
glVertex2f(2.0f, -9.0f);
glVertex2f(9.5f, -7.0f);
glVertex2f(6.0f, -4.5f);
glVertex2f(-12.0f, -4.5f);
glVertex2f(-12.0f, -9.5f);

glEnd();

//polygon 5
glBegin(GL_POLYGON);// Draw a Red 1x1 Square centered at origin
glColor3ub(0, 128, 0);
glVertex2f(2.0f, -11.5f);
glVertex2f(5.0f, -10.0f);
glVertex2f(2.0f, -9.0f);
```

```
glVertex2f(-12.0f, -9.0f);
glVertex2f(-12.0f, -11.5f);

glEnd();

///left sidewise border partition///
 //polygon 1
glBegin(GL_POLYGON);// Draw a Red 1x1 Square centered at origin
glColor3ub(0, 0, 0);
glVertex2f(6.0f, 0.5f);
glVertex2f(6.0f, 1.0f);
glVertex2f(4.0f, 0.0f);
glVertex2f(4.5f, -0.25f);

glEnd();

//polygon 2
glBegin(GL_POLYGON);// Draw a Red 1x1 Square centered at origin
glColor3ub(0, 0, 0);
glVertex2f(7.0f, -2.0f);
glVertex2f(7.0f, -1.5f);
glVertex2f(5.0f, -3.0f);
glVertex2f(5.5f, -3.15f);

glEnd();

//polygon 3
glBegin(GL_POLYGON);// Draw a Red 1x1 Square centered at origin
glColor3ub(0, 0, 0);
glVertex2f(8.0f, -4.4f);
glVertex2f(8.0f, -4.0f);
glVertex2f(6.0f, -4.5f);
glVertex2f(6.3f, -4.8f);
```

```
glEnd();

//polygon 4
glBegin(GL_POLYGON);// Draw a Red 1x1 Square centered at origin
glColor3ub(0, 0, 0);
glVertex2f(9.5f, -7.5f);
glVertex2f(9.5f, -7.0f);
glVertex2f(2.0f, -9.0f);
glVertex2f(3.0f, -9.3f);

glEnd();

//polygon 5
glBegin(GL_POLYGON);// Draw a Red 1x1 Square centered at origin
glColor3ub(0, 0, 0);
glVertex2f(5.0f, -10.5f);
glVertex2f(5.0f, -10.0f);
glVertex2f(2.0f, -11.5f);
glVertex2f(2.0f, -12.0f);

glEnd();

//polygon 6
glBegin(GL_POLYGON);// Draw a Red 1x1 Square centered at origin
glColor3ub(0, 0, 0);
glVertex2f(2.0f, -12.0f);
glVertex2f(2.0f, -11.5f);
glVertex2f(-12.0f, -11.5f);
glVertex2f(-12.0f, -12.0f);

glEnd();
```

```
///straw///
//polygon 1
glBegin(GL_POLYGON);// Draw a Red 1x1 Square centered at origin
glColor3ub(189, 183, 107);
glVertex2f(2.5f, -3.0f);
glVertex2f(2.0f, -1.0f);
glVertex2f(-3.0f, -1.0f);
glVertex2f(-3.5f, -3.0f);

glEnd();

//polygon 2
glBegin(GL_POLYGON);// Draw a Red 1x1 Square centered at origin
glColor3ub(189, 183, 107);
glVertex2f(2.0f, -1.0f);
glVertex2f(1.0f, 0.5f);
glVertex2f(-2.0f, 0.5f);
glVertex2f(-3.0f, -1.0f);

glEnd();
//polygon 3
glBegin(GL_POLYGON);// Draw a Red 1x1 Square centered at origin
glColor3ub(189, 183, 107);
glVertex2f(1.0f, 0.5f);
glVertex2f(0.0f, 1.0f);
glVertex2f(-1.0f, 1.0f);
glVertex2f(-2.0f, 0.5f);

glEnd();

///triangle///
```

```
glBegin(GL_POLYGON);// Draw a Red 1x1 Square centered at origin
glColor3ub(189, 183, 107);
glVertex2f(0.0f, 1.0f);
glVertex2f(-0.5f, 1.2f);
glVertex2f(-1.0f, 1.0f);

glEnd();

//polygon

glBegin(GL_POLYGON);// Draw a Red 1x1 Square centered at origin
glColor3ub(189, 183, 107);
glVertex2f(-0.4f, 1.1f);
glVertex2f(-0.4f, 1.5f);
glVertex2f(-0.6f, 1.5f);
glVertex2f(-0.6f, 1.1f);

glEnd();



///house on left side///
//polygon 1
glBegin(GL_POLYGON);// Draw a Red 1x1 Square centered at origin
glColor3ub(139, 69, 19);
glVertex2f(-2.7f, 1.2f);
glVertex2f(-6.7f, 1.2f);
glVertex2f(-5.0f, -1.0f);
glVertex2f(-1.0f, -1.0f);

glEnd();



//polygon 2
glBegin(GL_POLYGON);// Draw a Red 1x1 Square centered at origin
```

```
glColor3ub(139, 69, 19);
glVertex2f(-6.5f, 1.0f);
glVertex2f(-6.7f, 1.2f);
glVertex2f(-8.5f, -1.0f);
glVertex2f(-8.0f, -1.0f);

glEnd();

//polygon 3
glBegin(GL_POLYGON);// Draw a Red 1x1 Square centered at origin
glColor3ub(205, 133, 63);
glVertex2f(-6.5f, 1.0f);
glVertex2f(-8.0f, -1.0f);
glVertex2f(-8.0f, -2.5f);
glVertex2f(-5.0f, -3.0f);
glVertex2f(-5.0f, -1.0f);

glEnd();

//polygon 4
glBegin(GL_POLYGON);// Draw a Red 1x1 Square centered at origin
glColor3ub(128, 135, 32);
glVertex2f(-6.0f, -2.0f);
glVertex2f(-6.0f, -1.0f);
glVertex2f(-7.0f, -1.0f);
glVertex2f(-7.0f, -2.0f);

glEnd();

//polygon 5
glBegin(GL_POLYGON);// Draw a Red 1x1 Square centered at origin
glColor3ub(139, 69, 19);
glVertex2f(-5.0f, -3.0f);
glVertex2f(-8.0f, -2.5f);
```

```
glVertex2f(-8.3f, -2.9f);
glVertex2f(-5.0f, -3.5f);

glEnd();

//polygon 6
glBegin(GL_POLYGON);// Draw a Red 1x1 Square centered at origin
glColor3ub(184, 134, 11);
glVertex2f(-1.5f, -3.0f);
glVertex2f(-1.5f, -1.0f);
glVertex2f(-5.0f, -1.0f);
glVertex2f(-5.0f, -3.0f);

glEnd();

//polygon 7
glBegin(GL_POLYGON);// Draw a Red 1x1 Square centered at origin
glColor3ub(139, 69, 19);
glVertex2f(-5.0f, -3.0f);
glVertex2f(-5.0f, -3.5f);
glVertex2f(-1.0f, -3.5f);
glVertex2f(-1.5f, -3.0f);

glEnd();


///door:left
//polygon a
glBegin(GL_POLYGON);// Draw a Red 1x1 Square centered at origin
glColor3ub(0, 0, 0);
glVertex2f(-2.5f, -3.0f);
glVertex2f(-2.5f, -1.2f);
glVertex2f(-4.0f, -1.2f);
glVertex2f(-4.0f, -3.0f);
```

```
glEnd();

//polygon b
glBegin(GL_POLYGON);// Draw a Red 1x1 Square centered at origin
glColor3ub(128, 135, 32);
glVertex2f(-3.3f, -3.0f);
glVertex2f(-3.3f, -1.4f);
glVertex2f(-4.0f, -1.2f);
glVertex2f(-4.0f, -3.0f);

glEnd();

//polygon c
glBegin(GL_POLYGON);// Draw a Red 1x1 Square centered at origin
glColor3ub(128, 135, 32);
glVertex2f(-2.5f, -3.0f);
glVertex2f(-2.5f, -1.2f);
glVertex2f(-3.2f, -1.4f);
glVertex2f(-3.2f, -3.0f);

glEnd();

///tree left side
//polygon 1
glBegin(GL_POLYGON);
glColor3ub(139, 69, 19);
glVertex2f(-9.5, 1);
glVertex2f(-9, 1.5);
glVertex2f(-9.5, 2.5);
glVertex2f(-10, 2);
glEnd();

//polygon 2
```

```
glBegin(GL_POLYGON);
glColor3ub(139, 69, 19);
glVertex2f(-9, 1.5);
glVertex2f(-8.5, 1);
glVertex2f(-8, 2);
glVertex2f(-8, 2.5);
glEnd();

//polygon 3
glBegin(GL_POLYGON);
glColor3ub(139, 69, 19);
glVertex2f(-9.5, 1);
glVertex2f(-9.5, -3.5);
glVertex2f(-9, -4);
glVertex2f(-8.5, -3.5);
glVertex2f(-8.5, 1);
glEnd();

//polygon 4
glBegin(GL_POLYGON);
glColor3ub(139, 69, 19);
glVertex2f(-8.5, -3);
glVertex2f(-9.5, -3);
glVertex2f(-10.5, -3.5);
glVertex2f(-7.5, -3.5);
glEnd();



//triangle 1
glBegin(GL_TRIANGLES);
glColor3ub(139, 69, 19);
glVertex2f(-10, -0.5);
glVertex2f(-9.5, -1);
```

```
glVertex2f(-9.5, -0.6);
glEnd();

//triangle 2
glBegin(GL_TRIANGLES);
glColor3ub(139, 69, 19);
glVertex2f(-9.5, 1);
glVertex2f(-9, 1.5);
glVertex2f(-8.5, 1);
glEnd();



///right soil portion///
//polygon1//
glBegin(GL_POLYGON);

glColor3ub(0, 128, 0);
glVertex2f(13.0f, 3.0f);
glVertex2f(16.0f, 1.0f);
glVertex2f(38.0f, 1.0f);
glVertex2f(38.0f, 3.0f);

glEnd();

//polygon2//
glBegin(GL_POLYGON);

glColor3ub(0, 128, 0);
glVertex2f(16, 1);
glVertex2f(14, 0);
glVertex2f(17, -1.5);
glVertex2f(38, -1.5);
glVertex2f(38, 1);
```

```
glEnd();

//polygon3//
glBegin(GL_POLYGON);

glColor3ub(0, 128, 0);
glVertex2f(17, -1.5);
glVertex2f(15, -3);
glVertex2f(18, -4);
glVertex2f(38, -4);
glVertex2f(38, -1.5);

glEnd();

//polygon4//
glBegin(GL_POLYGON);

glColor3ub(0, 128, 0);
glVertex2f(18, -4);
glVertex2f(16, -4.5);
glVertex2f(21, -7);
glVertex2f(38, -7);
glVertex2f(38, -4);

glEnd();

//polygon5//
glBegin(GL_POLYGON);

glColor3ub(0, 128, 0);
glVertex2f(21, -7);
glVertex2f(19, -8);
glVertex2f(22, -9);
```

```
glVertex2f(38, -9);
glVertex2f(38, -7);

glEnd();

///right sidewise border partition///

//polygon1//
glBegin(GL_POLYGON);

glColor3ub(0, 0, 0);
glVertex2f(12.5, 3);
glVertex2f(15.6, 0.8);
glVertex2f(16, 1);
glVertex2f(13, 3);

glEnd();

//polygon2//
glBegin(GL_POLYGON);

glColor3ub(0, 0, 0);
glVertex2f(14, -0.5);
glVertex2f(16.4, -1.9);
glVertex2f(17, -1.5);
glVertex2f(14, 0);

glEnd();

//polygon3//
glBegin(GL_POLYGON);

glColor3ub(0, 0, 0);
glVertex2f(15, -3.5);
```

```
glVertex2f(17, -4.2);
glVertex2f(18, -4);
glVertex2f(15, -3);

glEnd();

//polygon4//
glBegin(GL_POLYGON);

glColor3ub(0, 0, 0);
glVertex2f(16, -5);
glVertex2f(20.5, -7.2);
glVertex2f(21, -7);
glVertex2f(16, -4.5);

glEnd();

//polygon5//
glBegin(GL_POLYGON);

glColor3ub(0, 0, 0);
glVertex2f(19, -8.5);
glVertex2f(22, -9.5);
glVertex2f(22, -9);
glVertex2f(19, -8);

glEnd();

//polygon6//
glBegin(GL_POLYGON);

glColor3ub(0, 0, 0);
glVertex2f(38, -9);
glVertex2f(22, -9);
```

```
glVertex2f(22, -9.5);
glVertex2f(38, -9.5);

glEnd();

///house on right side///

//polygon1//
glBegin(GL_POLYGON);

//glColor3ub(25,25,112);
glColor3ub(47, 79, 79);
glVertex2f(25.3, 1.2);
glVertex2f(21.3, 1.2);
glVertex2f(23, -1);
glVertex2f(27, -1);

glEnd();

//polygon2//
glBegin(GL_POLYGON);
glColor3ub(47, 79, 79);
//glColor3ub(139,69,19);
glVertex2f(21.5, 1);
glVertex2f(21.3, 1.2);
glVertex2f(19.5, -1);
glVertex2f(20, -1);

glEnd();

//polygon3//
glBegin(GL_POLYGON);

glColor3ub(184, 134, 11);
```

```
glVertex2f(21.5, 1);
glVertex2f(20, -1);
glVertex2f(20, -2.5);
glVertex2f(23, -3);
glVertex2f(23, -1);

glEnd();

//polygon4//
glBegin(GL_POLYGON);

glColor3ub(160, 82, 45);
glVertex2f(22, -2);
glVertex2f(22, -1);
glVertex2f(21, -1);
glVertex2f(21, -2);

glEnd();

//polygon5//
glBegin(GL_POLYGON);
glColor3ub(47, 79, 79);
//glColor3ub(139,69,19);
glVertex2f(23, -3);
glVertex2f(20, -2.5);
glVertex2f(19.7, -2.9);
glVertex2f(23, -3.5);

glEnd();

//polygon6//
glBegin(GL_POLYGON);

glColor3ub(205, 133, 63);
```

```
glVertex2f(26.5, -3);
glVertex2f(26.5, -1);
glVertex2f(23, -1);
glVertex2f(23, -3);

glEnd();

//polygon7//
glBegin(GL_POLYGON);
glColor3ub(47, 79, 79);
//glColor3ub(139,69,19);
glVertex2f(23, -3);
glVertex2f(23, -3.5);
glVertex2f(27, -3.5);
glVertex2f(26.5, -3);

glEnd();

///door:right///
//polygon a//
glBegin(GL_POLYGON);

glColor3ub(0, 0, 0);
glVertex2f(25.5, -3);
glVertex2f(25.5, -1.2);
glVertex2f(24, -1.2);
glVertex2f(24, -3);

glEnd();

//polygon b//
glBegin(GL_POLYGON);

glColor3ub(160, 82, 45);
```

```
glVertex2f(24.7, -3);
glVertex2f(24.7, -1.4);
glVertex2f(24, -1.2);
glVertex2f(24, -3);

glEnd();

//polygon c//
glBegin(GL_POLYGON);

glColor3ub(160, 82, 45);
glVertex2f(25.5, -3);
glVertex2f(25.5, -1.2);
glVertex2f(24.8, -1.4);
glVertex2f(24.8, -3);

glEnd();

///hillview///
///hill-1///
//polygon1//
glBegin(GL_POLYGON);
glColor3ub(0, 100, 0);
glVertex2f(-12, 3);
glVertex2f(-11.5, 4.5);
glVertex2f(-7.5, 4.5);
glVertex2f(-7, 3);
//glVertex2f();
glEnd();

//polygon2//
glBegin(GL_POLYGON);
glColor3ub(0, 100, 0);
glVertex2f(-11.5, 4.5);
```

```
glVertex2f(-11, 5.5);
glVertex2f(-8, 5.5);
glVertex2f(-7.5, 4.5);
//glVertex2f();
glEnd();

//polygon3//
glBegin(GL_POLYGON);
glColor3ub(0, 100, 0);
glVertex2f(-11, 5.5);
glVertex2f(-10.5, 6);
glVertex2f(-8.5, 6);
glVertex2f(-8, 5.5);
//glVertex2f();
glEnd();

//polygon4//
glBegin(GL_POLYGON);
glColor3ub(0, 100, 0);
glVertex2f(-10.5, 6);
glVertex2f(-10.3, 6.2);
glVertex2f(-8.7, 6.2);
glVertex2f(-8.5, 6);
//glVertex2f();
glEnd();

//polygon5//
glBegin(GL_POLYGON);
glColor3ub(0, 100, 0);
glVertex2f(-10.3, 6.2);
glVertex2f(-10.2, 6.3);
glVertex2f(-8.8, 6.3);
glVertex2f(-8.7, 6);
//glVertex2f();
```

```
glEnd();

///Hill 2///
//polygon1//
glBegin(GL_POLYGON);
glColor3ub(0, 100, 0);
glVertex2f(-7, 3);
glVertex2f(-6.5, 4);
glVertex2f(-4.5, 4);
glVertex2f(-4, 3);
//glVertex2f();
glEnd();

//polygon2//
glBegin(GL_POLYGON);
glColor3ub(0, 100, 0);
glVertex2f(-6.5, 4);
glVertex2f(-6, 4.5);
glVertex2f(-5, 4.5);
glVertex2f(-4.5, 4);
//glVertex2f();
glEnd();

///Hill 3///
//polygon1//
glBegin(GL_POLYGON);
glColor3ub(0, 100, 0);
glVertex2f(-4, 3);
glVertex2f(-3.5, 4.5);
glVertex2f(-1.5, 4.5);
glVertex2f(-1, 3);
//glVertex2f();
glEnd();
```

```
//polygon2//
glBegin(GL_POLYGON);
glColor3ub(0, 100, 0);
glVertex2f(-3.5, 4.5);
glVertex2f(-3, 5);
glVertex2f(-2, 5);
glVertex2f(-1.5, 4.5);
//glVertex2f();
glEnd();

///Hill 4///
//polygon1//
glBegin(GL_POLYGON);
glColor3ub(0, 100, 0);
glVertex2f(-1, 3);
glVertex2f(-0.5, 6);
glVertex2f(4.5, 6);
glVertex2f(5, 3);
//glVertex2f();
glEnd();

//polygon2//
glBegin(GL_POLYGON);
glColor3ub(0, 100, 0);
glVertex2f(-0.5, 6);
glVertex2f(0, 7);
glVertex2f(4, 7);
glVertex2f(4.5, 6);
//glVertex2f();
glEnd();

//polygon3//
glBegin(GL_POLYGON);
glColor3ub(0, 100, 0);
```

```
glVertex2f(0, 7);
glVertex2f(1, 8);
glVertex2f(3, 8);
glVertex2f(4, 7);
//glVertex2f();
glEnd();
//triangle//
glBegin(GL_TRIANGLES);
glColor3ub(0, 100, 0);
glVertex2f(1, 8);
glVertex2f(3, 8);
glVertex2f(2, 8.3);
glEnd();

///hill 5///
//polygon1//
glBegin(GL_POLYGON);
glColor3ub(0, 100, 0);
glVertex2f(5, 3);
glVertex2f(5.5, 4);
glVertex2f(6.5, 4);
glVertex2f(7, 3);
//glVertex2f();
glEnd();

//triangle//
glBegin(GL_TRIANGLES);
glColor3ub(0, 100, 0);
glVertex2f(5.5, 4);
glVertex2f(6.5, 4);
glVertex2f(6, 4.5);
glEnd();

///hill 6///
```

```
//polygon1//
glBegin(GL_POLYGON);
glColor3ub(0, 100, 0);
glVertex2f(7, 3);
glVertex2f(8, 5);
glVertex2f(13, 5);
glVertex2f(14, 3);
//glVertex2f();
glEnd();

//polygon2//
glBegin(GL_POLYGON);
glColor3ub(0, 100, 0);
glVertex2f(8, 5);
glVertex2f(9, 6);
glVertex2f(12, 6);
glVertex2f(13, 5);
//glVertex2f();
glEnd();
//triangle//
glBegin(GL_TRIANGLES);
glColor3ub(0, 100, 0);
glVertex2f(9, 6);
glVertex2f(12, 6);
glVertex2f(10.5, 6.5);
glEnd();

///hill 7///
//polygon1//
glBegin(GL_POLYGON);
glColor3ub(0, 100, 0);
glVertex2f(14, 3);
glVertex2f(15, 5);
glVertex2f(20, 5);
```

```
glVertex2f(21, 3);
//glVertex2f();
glEnd();

//polygon2//
glBegin(GL_POLYGON);
glColor3ub(0, 100, 0);
glVertex2f(15, 5);
glVertex2f(16, 6);
glVertex2f(19, 6);
glVertex2f(20, 5);
//glVertex2f();
glEnd();

//triangle//
glBegin(GL_TRIANGLES);
glColor3ub(0, 100, 0);
glVertex2f(16, 6);
glVertex2f(19, 6);
glVertex2f(17.5, 6.5);
glEnd();

///hill 8///
//polygon1//
glBegin(GL_POLYGON);
glColor3ub(0, 100, 0);
glVertex2f(21, 3);
glVertex2f(22, 5);
glVertex2f(24, 5);
glVertex2f(25, 3);
//glVertex2f();
glEnd();

//polygon2//
```

```
glBegin(GL_POLYGON);
glColor3ub(0, 100, 0);
glVertex2f(22, 5);
glVertex2f(22.5, 5.5);
glVertex2f(23.5, 5.5);
glVertex2f(24, 5);
//glVertex2f();
glEnd();

///hill 9///
//polygon1//
glBegin(GL_POLYGON);
glColor3ub(0, 100, 0);
glVertex2f(25, 3);
glVertex2f(25.5, 4.5);
glVertex2f(28.5, 4.5);
glVertex2f(29, 3);
//glVertex2f();
glEnd();

//polygon2//
glBegin(GL_POLYGON);
glColor3ub(0, 100, 0);
glVertex2f(25.5, 4.5);
glVertex2f(26, 5);
glVertex2f(28, 5);
glVertex2f(28.5, 4.5);
//glVertex2f();
glEnd();

///hill 10///
//polygon4//
glBegin(GL_POLYGON);
glColor3ub(0, 100, 0);
```

```
glVertex2f(29, 3);
glVertex2f(29.5, 4);
glVertex2f(30.5, 4);
glVertex2f(31, 3);
//glVertex2f();
glEnd();

//triangle//
glBegin(GL_TRIANGLES);
glColor3ub(0, 100, 0);
glVertex2f(29.5, 4);
glVertex2f(30, 4.5);
glVertex2f(30.5, 4);
glEnd();

///hill 11///
//polygon1//
glBegin(GL_POLYGON);
glColor3ub(0, 100, 0);
glVertex2f(31, 3);
glVertex2f(31.5, 4.5);
glVertex2f(32.5, 4.5);
glVertex2f(33, 3);
//glVertex2f();
glEnd();

//triangle//
glBegin(GL_TRIANGLES);
glColor3ub(0, 100, 0);
glVertex2f(31.5, 4.5);
glVertex2f(32.5, 4.5);
glVertex2f(32, 5);
glEnd();
```

```
///hill 12///
//polygon1//
glBegin(GL_POLYGON);
glColor3ub(0, 100, 0);
glVertex2f(33, 3);
glVertex2f(33.5, 4.5);
glVertex2f(37.5, 4.5);
glVertex2f(38, 3);
//glVertex2f();
glEnd();

//polygon2//
glBegin(GL_POLYGON);
glColor3ub(0, 100, 0);
glVertex2f(33.5, 4.5);
glVertex2f(34, 5);
glVertex2f(37, 5);
glVertex2f(37.5, 4.5);
//glVertex2f();
glEnd();

//triangle//
glBegin(GL_TRIANGLES);
glColor3ub(0, 100, 0);
glVertex2f(34, 5);
glVertex2f(37, 5);
glVertex2f(35.5, 5.5);
glEnd();

///tree upper side///

glPushMatrix();
glColor3ub(50, 205, 50);
glTranslatef(-10.5, 2.5, 0);
```

```
glutSolidSphere(1.0, 150, 150);
glPopMatrix();

glPushMatrix();
glColor3ub(50, 205, 50);
glTranslatef(-9.5, 3.0, 0);
glutSolidSphere(1.0, 150, 150);
glPopMatrix();

glPushMatrix();
glColor3ub(50, 205, 50);
glTranslatef(-7.5, 1.5, 0);
glutSolidSphere(1.0, 150, 150);
glPopMatrix();

glPushMatrix();
glColor3ub(50, 205, 50);
glTranslatef(-7.5, 2.5, 0);
glutSolidSphere(1.0, 150, 150);
glPopMatrix();

glPushMatrix();
glColor3ub(50, 205, 50);
glTranslatef(-8.5, 3.5, 0);
glutSolidSphere(1.0, 150, 150);
glPopMatrix();

glPushMatrix();
glColor3ub(50, 205, 50);
glTranslatef(-10.0, 4.0, 0);
glutSolidSphere(1.0, 150, 150);
glPopMatrix();

glPushMatrix();
```

```
glColor3ub(50, 205, 50);
glTranslatef(-8.0, 4.0, 0);
glutSolidSphere(1.0, 150, 150);
glPopMatrix();

glPushMatrix();
glColor3ub(50, 205, 50);
glTranslatef(-9.0, 5.0, 0);
glutSolidSphere(1.0, 150, 150);
glPopMatrix();
```

```
///boat-1 motion left to right
glPushMatrix();
//glColor3d(1,0,0);
glTranslatef(_moveA, 0.0f, 0.0f);
glBegin(GL_QUADS);
glColor3ub(139, 69, 19);
glVertex2f(1.0f, -14.0f);
glVertex2f(1.50f, -13.0f);
glVertex2f(-2.0f, -13.0f);
glVertex2f(-3.0f, -14.0f);
glEnd();

glBegin(GL_QUADS);
glColor3ub(0, 0, 0);
glVertex2f(2.0f, -14.5f);
glVertex2f(2.0f, -14.0f);
glVertex2f(-3.0f, -14.0f);
```

```
glVertex2f(-3.0f, -14.5f);
glEnd();

glBegin(GL_TRIANGLES);
glColor3ub(0, 0, 0);
glVertex2f(-3.0f, -14.5f);
glVertex2f(-3.0f, -14.0f);
glVertex2f(-4.5f, -13.7f);

glEnd();

glBegin(GL_TRIANGLES);
glColor3ub(0, 0, 0);
glVertex2f(2.0f, -14.5f);
glVertex2f(3.5f, -13.7f);
glVertex2f(2.0f, -14.0f);

glEnd();

glBegin(GL_TRIANGLES);
glColor3ub(0, 0, 0);
glVertex2f(2.0f, -14.0f);
glVertex2f(1.5f, -13.0f);
glVertex2f(1.0f, -14.0f);

glEnd();

glPopMatrix();
```

```
///////////////
///boat-2 motion right to left
glPushMatrix();
//glColor3d(1,0,0);
glTranslatef(_moveB, 0.0f, 0.0f);
glBegin(GL_QUADS);
glColor3ub(139, 69, 19);
glVertex2f(22.0f, -17.5f);
glVertex2f(21.0f, -16.5f);
glVertex2f(17.50f, -16.5f);
glVertex2f(18.0f, -17.5f);
glEnd();

glBegin(GL_QUADS);
glColor3ub(0, 0, 0);
glVertex2f(22.0f, -18.0f);
glVertex2f(22.0f, -17.5f);
glVertex2f(17.0f, -17.5f);
glVertex2f(17.0f, -18.0f);
glEnd();

glBegin(GL_QUADS);
glColor3ub(255, 99, 71);
glVertex2f(20.5f, -16.5f);
glVertex2f(21.0f, -14.5f);
glVertex2f(18.5f, -14.5f);
glVertex2f(18.0f, -16.5f);
glEnd();

glBegin(GL_QUADS);
glColor3ub(139, 69, 19);
glVertex2f(19.8f, -14.5f);
glVertex2f(19.8f, -14.0f);
```

```
glVertex2f(19.7f, -14.0f);
glVertex2f(19.7f, -14.5f);
glEnd();

glBegin(GL_TRIANGLES);
glColor3ub(0, 0, 0);
glVertex2f(17.0f, -18.0f);
glVertex2f(17.0f, -17.5f);
glVertex2f(15.5f, -17.2f);

glEnd();

glBegin(GL_TRIANGLES);
glColor3ub(0, 0, 0);
glVertex2f(22.0f, -17.5f);
glVertex2f(22.0f, -18.0f);
glVertex2f(23.5f, -17.2f);

glEnd();

glBegin(GL_TRIANGLES);
glColor3ub(0, 0, 0);
glVertex2f(17.0f, -17.5f);
glVertex2f(18.0f, -17.5f);
glVertex2f(17.5f, -16.5f);

glEnd();

glPopMatrix();

///windmill structure///
glBegin(GL_POLYGON);// Draw a Red 1x1 Square centered at origin
glColor3ub(192, 192, 192);
//glColor3ub(0,206,209);
```

```
glVertex2f(31.0f, 2.0f);
glVertex2f(30.0f, -6.0f);
glVertex2f(34.0f, -6.0f);
glVertex2f(33.0f, 2.0f);
glVertex2f(32.5f, 3.0f);
glVertex2f(31.5f, 3.0f);



glEnd();



//circle//
glPushMatrix();
glColor3ub(0, 0, 0);
glTranslatef(32.0f, 3.0f, 0.0f);
glutSolidSphere(0.5, 150, 150);
glRotatef(_angle1, 0.0f, 0.0f, 1.0f);  // z cordinate, so it rotates in sy plane



glBegin(GL_QUADS);// first stand to hold the blade
glColor3ub(255, 255, 0);
glVertex2f(0.0f, 0.0f);
glVertex2f(5.0f, 0.0f);
glVertex2f(5.0f, 0.25f);
glVertex2f(0.0f, 0.25f);
glEnd();

glBegin(GL_QUADS);// second stand to hold the blade
glColor3ub(255, 255, 0);
glVertex2f(0.0f, 0.0f);
glVertex2f(0.0f, 5.0f);
glVertex2f(-0.25f, 5.0f);
```

```
glVertex2f(-0.25f, 0.0f);
glEnd();

glBegin(GL_QUADS);// third stand to hold the blade
glColor3ub(255, 255, 0);
glVertex2f(0.0f, -0.25f);
glVertex2f(0.0f, 0.0f);
glVertex2f(-5.0f, 0.0f);
glVertex2f(-5.0f, -0.25f);
glEnd();

glBegin(GL_QUADS);// fourth stand to hold the blade
glColor3ub(255, 255, 0);
glVertex2f(0.25f, 0.0f);
glVertex2f(0.0f, 0.0f);
glVertex2f(0.0f, -5.0f);
glVertex2f(0.25f, -5.0f);
glEnd();

glBegin(GL_TRIANGLES);// first triangular blade to hold the blade
glColor3ub(128, 0, 0);
glVertex2f(0.25f, 0.0f);
glVertex2f(5.0f, -2.5f);
glVertex2f(5.0f, 0.0f);
glEnd();

glBegin(GL_TRIANGLES);// second triangular blade to hold the blade
glColor3ub(128, 0, 0);
glVertex2f(0.0f, 0.025f);
glVertex2f(2.5f, 5.0f);
glVertex2f(0.0f, 5.0f);
glEnd();

glBegin(GL_TRIANGLES);// third triangular blade to hold the blade
```

```
    glColor3ub(128, 0, 0);
    glVertex2f(-0.25f, 0.0f);
    glVertex2f(-5.0f, 2.5f);
    glVertex2f(-5.0f, 0.0f);
    glEnd();

    glBegin(GL_TRIANGLES);// fourth triangular blade to hold the blade
    glColor3ub(128, 0, 0);
    glVertex2f(0.0f, -0.025f);
    glVertex2f(-2.5f, -5.0f);
    glVertex2f(0.0f, -5.0f);
    glEnd();









    glPopMatrix();









    glutSwapBuffers();// for GLUT_DOUBLE, shows after everything works, not realtime
}



void init() {// initial state of the OpenGL rendering environment before rendering begin and we will get black screen after GLUT buffer clears
    glClearColor(0.0f, 0.0f, 0.0f, 1.0f);
```

```
}

void drawScene2() { //night
    glClearColor(1.0f, 1.0f, 1.0f, 1.0f);
    glClear(GL_COLOR_BUFFER_BIT);
    glColor3d(1, 0, 0);
    glLoadIdentity(); //Reset the drawing perspective
    gluOrtho2D(-12, 38, -19, 14);//range
    glMatrixMode(GL_MODELVIEW);



    ///sky///
    glBegin(GL_POLYGON);// Draw a Red 1x1 Square centered at origin
    glColor3ub(0, 0, 0);
    glVertex2f(38.0f, 3.0f);
    glVertex2f(38.0f, 14.0f);
    glVertex2f(-12.0f, 14.0f);
    glVertex2f(-12.0f, 3.0f);

    glEnd();

    ///Stars///

    glPushMatrix();



    glPointSize(2.0);
    glColor3ub(255, 255, 255);

    glBegin(GL_POINTS);

    glVertex2f(1, 4);
    glVertex2f(2, 7.5);
    glVertex2f(4, 5);
```

```
glVertex2f(-2, 8);
glVertex2f(30, 5);
glVertex2f(-1, 9);
glVertex2f(35, 7);
glVertex2f(-3, 7);
glVertex2f(-7, 5);
glVertex2f(10, 7.5);
glVertex2f(12, 8.5);
glVertex2f(-11, 6);
glVertex2f(37, 7.5);
glVertex2f(-10, 4.5);
glVertex2f(13, 7);
glVertex2f(14, 7.5);
glVertex2f(18, 10);
// glVertex2f(-10,4.5);

glEnd();



glPopMatrix();



///moon///
    //1st circle for moon

glPushMatrix();
glColor3ub(217, 217, 217);
glTranslatef(-8.0, 7.0, 0);
glutSolidSphere(1, 250, 250);
glPopMatrix();

//2nd circle for moon
```

```
glPushMatrix();
glColor3ub(0, 0, 0);
glTranslatef(-8, 7.25, 0);
glutSolidSphere(1, 250, 250);
glPopMatrix();



///cloud 1///
glPushMatrix();

glTranslatef(_moveC, 0.0f, 0.0f);
glPushMatrix();
glColor3ub(105, 105, 105);
glTranslatef(1.0, 6.0, 0);
glutSolidSphere(0.7, 250, 250);
glPopMatrix();

glPushMatrix();
glColor3ub(105, 105, 105);
glTranslatef(1.0, 7.0, 0);
glutSolidSphere(0.7, 250, 250);
glPopMatrix();

glPushMatrix();
glColor3ub(105, 105, 105);
glTranslatef(2.0, 7.0, 0);
glutSolidSphere(0.7, 250, 250);
glPopMatrix();

glPushMatrix();
glColor3ub(105, 105, 105);
glTranslatef(2.0, 6.0, 0);
glutSolidSphere(0.7, 250, 250);
```

```
glPopMatrix();

glPushMatrix();
glColor3ub(105, 105, 105);
glTranslatef(0.0, 6.5, 0);
glutSolidSphere(0.7, 250, 250);
glPopMatrix();

glPushMatrix();
glColor3ub(105, 105, 105);
glTranslatef(3.0, 6.5, 0);
glutSolidSphere(0.7, 250, 250);
glPopMatrix();

glPopMatrix();



///cloud 2///



glPushMatrix();



glTranslatef(_moveC, 0.0f, 0.0f);
glPushMatrix();
glColor3ub(105, 105, 105);
glTranslatef(15.0, 9.0, 0);
glutSolidSphere(0.8, 250, 250);
glPopMatrix();
```

```
glPushMatrix();
glColor3ub(105, 105, 105);
glTranslatef(15.0, 8.0, 0);
glutSolidSphere(0.8, 250, 250);
glPopMatrix();



glPushMatrix();
glColor3ub(105, 105, 105);
glTranslatef(16.0, 8.0, 0);
glutSolidSphere(0.8, 250, 250);
glPopMatrix();



glPushMatrix();
glColor3ub(105, 105, 105);
glTranslatef(16.0, 9.0, 0);
glutSolidSphere(0.8, 250, 250);
glPopMatrix();



glPushMatrix();
glColor3ub(105, 105, 105);
glTranslatef(14.0, 8.5, 0);
glutSolidSphere(0.8, 250, 250);
glPopMatrix();
```

```
glPushMatrix();
glColor3ub(105, 105, 105);
glTranslatef(17.0, 8.5, 0);
glutSolidSphere(0.8, 250, 250);
glPopMatrix();




glPopMatrix();



///cloud 3///


glPushMatrix();


glTranslatef(_moveC, 0.0f, 0.0f);


glPushMatrix();
glColor3ub(105, 105, 105);
glTranslatef(22.0, 8.0, 0);
glutSolidSphere(0.75, 250, 250);
glPopMatrix();


glPushMatrix();
glColor3ub(105, 105, 105);
glTranslatef(22.0, 7.0, 0);
glutSolidSphere(0.75, 250, 250);
glPopMatrix();


glPushMatrix();
glColor3ub(105, 105, 105);
glTranslatef(23.0, 8.0, 0);
glutSolidSphere(0.75, 250, 250);
glPopMatrix();
```

```
glPushMatrix();
glColor3ub(105, 105, 105);
glTranslatef(23.0, 7.0, 0);
glutSolidSphere(0.75, 250, 250);
glPopMatrix();

glPushMatrix();
glColor3ub(105, 105, 105);
glTranslatef(24.0, 7.5, 0);
glutSolidSphere(0.75, 250, 250);
glPopMatrix();

glPushMatrix();
glColor3ub(105, 105, 105);
glTranslatef(21.0, 7.5, 0);
glutSolidSphere(0.75, 250, 250);
glPopMatrix();

glPopMatrix();



///sea portion///
glBegin(GL_POLYGON);// Draw a Red 1x1 Square centered at origin
glColor3ub(65, 105, 225);
glVertex2f(38.0f, -19.0f);
glVertex2f(38.0f, 3.0f);
glVertex2f(-12.0f, 3.0f);
glVertex2f(-12.0f, -19.0f);

glEnd();

///left soil portion///
//polygon 1
```

```
glBegin(GL_POLYGON);// Draw a Red 1x1 Square centered at origin
glColor3ub(0, 128, 0);
glVertex2f(4.0f, 0.0f);
glVertex2f(6.0f, 1.0f);
glVertex2f(3.0f, 3.0f);
glVertex2f(-12.0f, 3.0f);
glVertex2f(-12.0f, 0.0f);

glEnd();

//polygon 2
glBegin(GL_POLYGON);// Draw a Red 1x1 Square centered at origin
glColor3ub(0, 128, 0);
glVertex2f(5.0f, -3.0f);
glVertex2f(7.0f, -1.5f);
glVertex2f(4.0f, 0.0f);
glVertex2f(-12.0f, 0.0f);
glVertex2f(-12.0f, -3.0f);

glEnd();

//polygon 3
glBegin(GL_POLYGON);// Draw a Red 1x1 Square centered at origin
glColor3ub(0, 128, 0);
glVertex2f(6.0f, -4.5f);
glVertex2f(8.0f, -4.0f);
glVertex2f(5.0f, -3.0f);
glVertex2f(-12.0f, -3.0f);
glVertex2f(-12.0f, -4.5f);

glEnd();

//polygon 4
glBegin(GL_POLYGON);// Draw a Red 1x1 Square centered at origin
```

```
glColor3ub(0, 128, 0);
glVertex2f(2.0f, -9.0f);
glVertex2f(9.5f, -7.0f);
glVertex2f(6.0f, -4.5f);
glVertex2f(-12.0f, -4.5f);
glVertex2f(-12.0f, -9.5f);

glEnd();

//polygon 5
glBegin(GL_POLYGON);// Draw a Red 1x1 Square centered at origin
glColor3ub(0, 128, 0);
glVertex2f(2.0f, -11.5f);
glVertex2f(5.0f, -10.0f);
glVertex2f(2.0f, -9.0f);
glVertex2f(-12.0f, -9.0f);
glVertex2f(-12.0f, -11.5f);

glEnd();

///left sidewise border partition///
 //polygon 1
glBegin(GL_POLYGON);// Draw a Red 1x1 Square centered at origin
glColor3ub(0, 0, 0);
glVertex2f(6.0f, 0.5f);
glVertex2f(6.0f, 1.0f);
glVertex2f(4.0f, 0.0f);
glVertex2f(4.5f, -0.25f);

glEnd();

//polygon 2
glBegin(GL_POLYGON);// Draw a Red 1x1 Square centered at origin
glColor3ub(0, 0, 0);
```

```
glVertex2f(7.0f, -2.0f);
glVertex2f(7.0f, -1.5f);
glVertex2f(5.0f, -3.0f);
glVertex2f(5.5f, -3.15f);

glEnd();

//polygon 3
glBegin(GL_POLYGON);// Draw a Red 1x1 Square centered at origin
glColor3ub(0, 0, 0);
glVertex2f(8.0f, -4.4f);
glVertex2f(8.0f, -4.0f);
glVertex2f(6.0f, -4.5f);
glVertex2f(6.3f, -4.8f);

glEnd();

//polygon 4
glBegin(GL_POLYGON);// Draw a Red 1x1 Square centered at origin
glColor3ub(0, 0, 0);
glVertex2f(9.5f, -7.5f);
glVertex2f(9.5f, -7.0f);
glVertex2f(2.0f, -9.0f);
glVertex2f(3.0f, -9.3f);

glEnd();

//polygon 5
glBegin(GL_POLYGON);// Draw a Red 1x1 Square centered at origin
glColor3ub(0, 0, 0);
glVertex2f(5.0f, -10.5f);
glVertex2f(5.0f, -10.0f);
glVertex2f(2.0f, -11.5f);
glVertex2f(2.0f, -12.0f);
```

```
glEnd();

//polygon 6
glBegin(GL_POLYGON);// Draw a Red 1x1 Square centered at origin
glColor3ub(0, 0, 0);
glVertex2f(2.0f, -12.0f);
glVertex2f(2.0f, -11.5f);
glVertex2f(-12.0f, -11.5f);
glVertex2f(-12.0f, -12.0f);

glEnd();




///straw///
//polygon 1
glBegin(GL_POLYGON);// Draw a Red 1x1 Square centered at origin
glColor3ub(189, 183, 107);
glVertex2f(2.5f, -3.0f);
glVertex2f(2.0f, -1.0f);
glVertex2f(-3.0f, -1.0f);
glVertex2f(-3.5f, -3.0f);

glEnd();

//polygon 2
glBegin(GL_POLYGON);// Draw a Red 1x1 Square centered at origin
glColor3ub(189, 183, 107);
glVertex2f(2.0f, -1.0f);
glVertex2f(1.0f, 0.5f);
glVertex2f(-2.0f, 0.5f);
```

```
glVertex2f(-3.0f, -1.0f);

glEnd();
//polygon 3
glBegin(GL_POLYGON);// Draw a Red 1x1 Square centered at origin
glColor3ub(189, 183, 107);
glVertex2f(1.0f, 0.5f);
glVertex2f(0.0f, 1.0f);
glVertex2f(-1.0f, 1.0f);
glVertex2f(-2.0f, 0.5f);

glEnd();

///triangle///

glBegin(GL_POLYGON);// Draw a Red 1x1 Square centered at origin
glColor3ub(189, 183, 107);
glVertex2f(0.0f, 1.0f);
glVertex2f(-0.5f, 1.2f);
glVertex2f(-1.0f, 1.0f);

glEnd();

//polygon

glBegin(GL_POLYGON);// Draw a Red 1x1 Square centered at origin
glColor3ub(189, 183, 107);
glVertex2f(-0.4f, 1.1f);
glVertex2f(-0.4f, 1.5f);
glVertex2f(-0.6f, 1.5f);
glVertex2f(-0.6f, 1.1f);

glEnd();
```

```
///house on left side///
//polygon 1
glBegin(GL_POLYGON);// Draw a Red 1x1 Square centered at origin
glColor3ub(139, 69, 19);
glVertex2f(-2.7f, 1.2f);
glVertex2f(-6.7f, 1.2f);
glVertex2f(-5.0f, -1.0f);
glVertex2f(-1.0f, -1.0f);

glEnd();


//polygon 2
glBegin(GL_POLYGON);// Draw a Red 1x1 Square centered at origin
glColor3ub(139, 69, 19);
glVertex2f(-6.5f, 1.0f);
glVertex2f(-6.7f, 1.2f);
glVertex2f(-8.5f, -1.0f);
glVertex2f(-8.0f, -1.0f);

glEnd();

//polygon 3
glBegin(GL_POLYGON);// Draw a Red 1x1 Square centered at origin
glColor3ub(205, 133, 63);
glVertex2f(-6.5f, 1.0f);
glVertex2f(-8.0f, -1.0f);
glVertex2f(-8.0f, -2.5f);
glVertex2f(-5.0f, -3.0f);
glVertex2f(-5.0f, -1.0f);

glEnd();
```

```
//polygon 4
glBegin(GL_POLYGON);// Draw a Red 1x1 Square centered at origin
glColor3ub(128, 135, 32);
glVertex2f(-6.0f, -2.0f);
glVertex2f(-6.0f, -1.0f);
glVertex2f(-7.0f, -1.0f);
glVertex2f(-7.0f, -2.0f);

glEnd();

//polygon 5
glBegin(GL_POLYGON);// Draw a Red 1x1 Square centered at origin
glColor3ub(139, 69, 19);
glVertex2f(-5.0f, -3.0f);
glVertex2f(-8.0f, -2.5f);
glVertex2f(-8.3f, -2.9f);
glVertex2f(-5.0f, -3.5f);

glEnd();

//polygon 6
glBegin(GL_POLYGON);// Draw a Red 1x1 Square centered at origin
glColor3ub(184, 134, 11);
glVertex2f(-1.5f, -3.0f);
glVertex2f(-1.5f, -1.0f);
glVertex2f(-5.0f, -1.0f);
glVertex2f(-5.0f, -3.0f);

glEnd();

//polygon 7
glBegin(GL_POLYGON);// Draw a Red 1x1 Square centered at origin
glColor3ub(139, 69, 19);
glVertex2f(-5.0f, -3.0f);
```

```
glVertex2f(-5.0f, -3.5f);
glVertex2f(-1.0f, -3.5f);
glVertex2f(-1.5f, -3.0f);

glEnd();



///door:left
//polygon a
glBegin(GL_POLYGON);// Draw a Red 1x1 Square centered at origin
glColor3ub(0, 0, 0);
glVertex2f(-2.5f, -3.0f);
glVertex2f(-2.5f, -1.2f);
glVertex2f(-4.0f, -1.2f);
glVertex2f(-4.0f, -3.0f);

glEnd();

//polygon b
glBegin(GL_POLYGON);// Draw a Red 1x1 Square centered at origin
glColor3ub(128, 135, 32);
glVertex2f(-3.3f, -3.0f);
glVertex2f(-3.3f, -1.4f);
glVertex2f(-4.0f, -1.2f);
glVertex2f(-4.0f, -3.0f);

glEnd();

//polygon c
glBegin(GL_POLYGON);// Draw a Red 1x1 Square centered at origin
glColor3ub(128, 135, 32);
glVertex2f(-2.5f, -3.0f);
glVertex2f(-2.5f, -1.2f);
glVertex2f(-3.2f, -1.4f);
```

```
glVertex2f(-3.2f, -3.0f);

glEnd();

///tree left side
//polygon 1
glBegin(GL_POLYGON);
glColor3ub(139, 69, 19);
glVertex2f(-9.5, 1);
glVertex2f(-9, 1.5);
glVertex2f(-9.5, 2.5);
glVertex2f(-10, 2);
glEnd();

//polygon 2
glBegin(GL_POLYGON);
glColor3ub(139, 69, 19);
glVertex2f(-9, 1.5);
glVertex2f(-8.5, 1);
glVertex2f(-8, 2);
glVertex2f(-8, 2.5);
glEnd();

//polygon 3
glBegin(GL_POLYGON);
glColor3ub(139, 69, 19);
glVertex2f(-9.5, 1);
glVertex2f(-9.5, -3.5);
glVertex2f(-9, -4);
glVertex2f(-8.5, -3.5);
glVertex2f(-8.5, 1);
glEnd();

//polygon 4
```

```
glBegin(GL_POLYGON);
glColor3ub(139, 69, 19);
glVertex2f(-8.5, -3);
glVertex2f(-9.5, -3);
glVertex2f(-10.5, -3.5);
glVertex2f(-7.5, -3.5);
glEnd();



//triangle 1
glBegin(GL_TRIANGLES);
glColor3ub(139, 69, 19);
glVertex2f(-10, -0.5);
glVertex2f(-9.5, -1);
glVertex2f(-9.5, -0.6);
glEnd();

//triangle 2
glBegin(GL_TRIANGLES);
glColor3ub(139, 69, 19);
glVertex2f(-9.5, 1);
glVertex2f(-9, 1.5);
glVertex2f(-8.5, 1);
glEnd();



///right soil portion///
//polygon1//
glBegin(GL_POLYGON);

glColor3ub(0, 128, 0);
glVertex2f(13.0f, 3.0f);
```

```
glVertex2f(16.0f, 1.0f);
glVertex2f(38.0f, 1.0f);
glVertex2f(38.0f, 3.0f);

glEnd();

//polygon2//
glBegin(GL_POLYGON);

glColor3ub(0, 128, 0);
glVertex2f(16, 1);
glVertex2f(14, 0);
glVertex2f(17, -1.5);
glVertex2f(38, -1.5);
glVertex2f(38, 1);

glEnd();

//polygon3//
glBegin(GL_POLYGON);

glColor3ub(0, 128, 0);
glVertex2f(17, -1.5);
glVertex2f(15, -3);
glVertex2f(18, -4);
glVertex2f(38, -4);
glVertex2f(38, -1.5);

glEnd();

//polygon4//
glBegin(GL_POLYGON);

glColor3ub(0, 128, 0);
```

```
glVertex2f(18, -4);
glVertex2f(16, -4.5);
glVertex2f(21, -7);
glVertex2f(38, -7);
glVertex2f(38, -4);

glEnd();

//polygon5//
glBegin(GL_POLYGON);

glColor3ub(0, 128, 0);
glVertex2f(21, -7);
glVertex2f(19, -8);
glVertex2f(22, -9);
glVertex2f(38, -9);
glVertex2f(38, -7);

glEnd();

///right sidewise border partition///

//polygon1//
glBegin(GL_POLYGON);

glColor3ub(0, 0, 0);
glVertex2f(12.5, 3);
glVertex2f(15.6, 0.8);
glVertex2f(16, 1);
glVertex2f(13, 3);

glEnd();

//polygon2//
```

```
glBegin(GL_POLYGON);

glColor3ub(0, 0, 0);
glVertex2f(14, -0.5);
glVertex2f(16.4, -1.9);
glVertex2f(17, -1.5);
glVertex2f(14, 0);

glEnd();

//polygon3//
glBegin(GL_POLYGON);

glColor3ub(0, 0, 0);
glVertex2f(15, -3.5);
glVertex2f(17, -4.2);
glVertex2f(18, -4);
glVertex2f(15, -3);

glEnd();

//polygon4//
glBegin(GL_POLYGON);

glColor3ub(0, 0, 0);
glVertex2f(16, -5);
glVertex2f(20.5, -7.2);
glVertex2f(21, -7);
glVertex2f(16, -4.5);

glEnd();

//polygon5//
glBegin(GL_POLYGON);
```

```
glColor3ub(0, 0, 0);
glVertex2f(19, -8.5);
glVertex2f(22, -9.5);
glVertex2f(22, -9);
glVertex2f(19, -8);

glEnd();

//polygon6//
glBegin(GL_POLYGON);

glColor3ub(0, 0, 0);
glVertex2f(38, -9);
glVertex2f(22, -9);
glVertex2f(22, -9.5);
glVertex2f(38, -9.5);

glEnd();

///house on right side///

//polygon1//
glBegin(GL_POLYGON);

//glColor3ub(25,25,112);
glColor3ub(47, 79, 79);
glVertex2f(25.3, 1.2);
glVertex2f(21.3, 1.2);
glVertex2f(23, -1);
glVertex2f(27, -1);

glEnd();
```

```
//polygon2//
glBegin(GL_POLYGON);
glColor3ub(47, 79, 79);
//glColor3ub(139,69,19);
glVertex2f(21.5, 1);
glVertex2f(21.3, 1.2);
glVertex2f(19.5, -1);
glVertex2f(20, -1);

glEnd();

//polygon3//
glBegin(GL_POLYGON);

glColor3ub(184, 134, 11);
glVertex2f(21.5, 1);
glVertex2f(20, -1);
glVertex2f(20, -2.5);
glVertex2f(23, -3);
glVertex2f(23, -1);

glEnd();

//polygon4//
glBegin(GL_POLYGON);

glColor3ub(160, 82, 45);
glVertex2f(22, -2);
glVertex2f(22, -1);
glVertex2f(21, -1);
glVertex2f(21, -2);

glEnd();
```

```
//polygon5//
glBegin(GL_POLYGON);
glColor3ub(47, 79, 79);
//glColor3ub(139,69,19);
glVertex2f(23, -3);
glVertex2f(20, -2.5);
glVertex2f(19.7, -2.9);
glVertex2f(23, -3.5);

glEnd();

//polygon6//
glBegin(GL_POLYGON);

glColor3ub(205, 133, 63);
glVertex2f(26.5, -3);
glVertex2f(26.5, -1);
glVertex2f(23, -1);
glVertex2f(23, -3);

glEnd();

//polygon7//
glBegin(GL_POLYGON);
glColor3ub(47, 79, 79);
//glColor3ub(139,69,19);
glVertex2f(23, -3);
glVertex2f(23, -3.5);
glVertex2f(27, -3.5);
glVertex2f(26.5, -3);

glEnd();

///door:right///
```

```
//polygon a//
glBegin(GL_POLYGON);

glColor3ub(0, 0, 0);
glVertex2f(25.5, -3);
glVertex2f(25.5, -1.2);
glVertex2f(24, -1.2);
glVertex2f(24, -3);

glEnd();

//polygon b//
glBegin(GL_POLYGON);

glColor3ub(160, 82, 45);
glVertex2f(24.7, -3);
glVertex2f(24.7, -1.4);
glVertex2f(24, -1.2);
glVertex2f(24, -3);

glEnd();

//polygon c//
glBegin(GL_POLYGON);

glColor3ub(160, 82, 45);
glVertex2f(25.5, -3);
glVertex2f(25.5, -1.2);
glVertex2f(24.8, -1.4);
glVertex2f(24.8, -3);

glEnd();

///hillview///
```

```
///hill-1///
//polygon1//
glBegin(GL_POLYGON);
glColor3ub(0, 100, 0);
glVertex2f(-12, 3);
glVertex2f(-11.5, 4.5);
glVertex2f(-7.5, 4.5);
glVertex2f(-7, 3);
//glVertex2f();
glEnd();

//polygon2//
glBegin(GL_POLYGON);
glColor3ub(0, 100, 0);
glVertex2f(-11.5, 4.5);
glVertex2f(-11, 5.5);
glVertex2f(-8, 5.5);
glVertex2f(-7.5, 4.5);
//glVertex2f();
glEnd();

//polygon3//
glBegin(GL_POLYGON);
glColor3ub(0, 100, 0);
glVertex2f(-11, 5.5);
glVertex2f(-10.5, 6);
glVertex2f(-8.5, 6);
glVertex2f(-8, 5.5);
//glVertex2f();
glEnd();

//polygon4//
glBegin(GL_POLYGON);
glColor3ub(0, 100, 0);
```

```
glVertex2f(-10.5, 6);
glVertex2f(-10.3, 6.2);
glVertex2f(-8.7, 6.2);
glVertex2f(-8.5, 6);
//glVertex2f();
glEnd();

//polygon5//
glBegin(GL_POLYGON);
glColor3ub(0, 100, 0);
glVertex2f(-10.3, 6.2);
glVertex2f(-10.2, 6.3);
glVertex2f(-8.8, 6.3);
glVertex2f(-8.7, 6);
//glVertex2f();
glEnd();

///Hill 2///
//polygon1//
glBegin(GL_POLYGON);
glColor3ub(0, 100, 0);
glVertex2f(-7, 3);
glVertex2f(-6.5, 4);
glVertex2f(-4.5, 4);
glVertex2f(-4, 3);
//glVertex2f();
glEnd();

//polygon2//
glBegin(GL_POLYGON);
glColor3ub(0, 100, 0);
glVertex2f(-6.5, 4);
glVertex2f(-6, 4.5);
glVertex2f(-5, 4.5);
```

```
glVertex2f(-4.5, 4);
//glVertex2f();
glEnd();

///Hill 3///
//polygon1//
glBegin(GL_POLYGON);
glColor3ub(0, 100, 0);
glVertex2f(-4, 3);
glVertex2f(-3.5, 4.5);
glVertex2f(-1.5, 4.5);
glVertex2f(-1, 3);
//glVertex2f();
glEnd();

//polygon2//
glBegin(GL_POLYGON);
glColor3ub(0, 100, 0);
glVertex2f(-3.5, 4.5);
glVertex2f(-3, 5);
glVertex2f(-2, 5);
glVertex2f(-1.5, 4.5);
//glVertex2f();
glEnd();

///Hill 4///
//polygon1//
glBegin(GL_POLYGON);
glColor3ub(0, 100, 0);
glVertex2f(-1, 3);
glVertex2f(-0.5, 6);
glVertex2f(4.5, 6);
glVertex2f(5, 3);
//glVertex2f();
```

```
glEnd();

//polygon2//
glBegin(GL_POLYGON);
glColor3ub(0, 100, 0);
glVertex2f(-0.5, 6);
glVertex2f(0, 7);
glVertex2f(4, 7);
glVertex2f(4.5, 6);
//glVertex2f();
glEnd();

//polygon3//
glBegin(GL_POLYGON);
glColor3ub(0, 100, 0);
glVertex2f(0, 7);
glVertex2f(1, 8);
glVertex2f(3, 8);
glVertex2f(4, 7);
//glVertex2f();
glEnd();
//triangle//
glBegin(GL_TRIANGLES);
glColor3ub(0, 100, 0);
glVertex2f(1, 8);
glVertex2f(3, 8);
glVertex2f(2, 8.3);
glEnd();

///hill 5///
//polygon1//
glBegin(GL_POLYGON);
glColor3ub(0, 100, 0);
glVertex2f(5, 3);
```

```
glVertex2f(5.5, 4);
glVertex2f(6.5, 4);
glVertex2f(7, 3);
//glVertex2f();
glEnd();

//triangle//
glBegin(GL_TRIANGLES);
glColor3ub(0, 100, 0);
glVertex2f(5.5, 4);
glVertex2f(6.5, 4);
glVertex2f(6, 4.5);
glEnd();

///hill 6///
//polygon1//
glBegin(GL_POLYGON);
glColor3ub(0, 100, 0);
glVertex2f(7, 3);
glVertex2f(8, 5);
glVertex2f(13, 5);
glVertex2f(14, 3);
//glVertex2f();
glEnd();

//polygon2//
glBegin(GL_POLYGON);
glColor3ub(0, 100, 0);
glVertex2f(8, 5);
glVertex2f(9, 6);
glVertex2f(12, 6);
glVertex2f(13, 5);
//glVertex2f();
glEnd();
```

```
//triangle//
glBegin(GL_TRIANGLES);
glColor3ub(0, 100, 0);
glVertex2f(9, 6);
glVertex2f(12, 6);
glVertex2f(10.5, 6.5);
glEnd();

///hill 7///
//polygon1//
glBegin(GL_POLYGON);
glColor3ub(0, 100, 0);
glVertex2f(14, 3);
glVertex2f(15, 5);
glVertex2f(20, 5);
glVertex2f(21, 3);
//glVertex2f();
glEnd();

//polygon2//
glBegin(GL_POLYGON);
glColor3ub(0, 100, 0);
glVertex2f(15, 5);
glVertex2f(16, 6);
glVertex2f(19, 6);
glVertex2f(20, 5);
//glVertex2f();
glEnd();

//triangle//
glBegin(GL_TRIANGLES);
glColor3ub(0, 100, 0);
glVertex2f(16, 6);
glVertex2f(19, 6);
```

```
glVertex2f(17.5, 6.5);
glEnd();

///hill 8///
//polygon1//
glBegin(GL_POLYGON);
glColor3ub(0, 100, 0);
glVertex2f(21, 3);
glVertex2f(22, 5);
glVertex2f(24, 5);
glVertex2f(25, 3);
//glVertex2f();
glEnd();

//polygon2//
glBegin(GL_POLYGON);
glColor3ub(0, 100, 0);
glVertex2f(22, 5);
glVertex2f(22.5, 5.5);
glVertex2f(23.5, 5.5);
glVertex2f(24, 5);
//glVertex2f();
glEnd();

///hill 9///
//polygon1//
glBegin(GL_POLYGON);
glColor3ub(0, 100, 0);
glVertex2f(25, 3);
glVertex2f(25.5, 4.5);
glVertex2f(28.5, 4.5);
glVertex2f(29, 3);
//glVertex2f();
glEnd();
```

```
//polygon2//
glBegin(GL_POLYGON);
glColor3ub(0, 100, 0);
glVertex2f(25.5, 4.5);
glVertex2f(26, 5);
glVertex2f(28, 5);
glVertex2f(28.5, 4.5);
//glVertex2f();
glEnd();

///hill 10///
//polygon4//
glBegin(GL_POLYGON);
glColor3ub(0, 100, 0);
glVertex2f(29, 3);
glVertex2f(29.5, 4);
glVertex2f(30.5, 4);
glVertex2f(31, 3);
//glVertex2f();
glEnd();

//triangle//
glBegin(GL_TRIANGLES);
glColor3ub(0, 100, 0);
glVertex2f(29.5, 4);
glVertex2f(30, 4.5);
glVertex2f(30.5, 4);
glEnd();

///hill 11///
//polygon1//
glBegin(GL_POLYGON);
glColor3ub(0, 100, 0);
```

```
glVertex2f(31, 3);
glVertex2f(31.5, 4.5);
glVertex2f(32.5, 4.5);
glVertex2f(33, 3);
//glVertex2f();
glEnd();

//triangle//
glBegin(GL_TRIANGLES);
glColor3ub(0, 100, 0);
glVertex2f(31.5, 4.5);
glVertex2f(32.5, 4.5);
glVertex2f(32, 5);
glEnd();

///hill 12///
//polygon1//
glBegin(GL_POLYGON);
glColor3ub(0, 100, 0);
glVertex2f(33, 3);
glVertex2f(33.5, 4.5);
glVertex2f(37.5, 4.5);
glVertex2f(38, 3);
//glVertex2f();
glEnd();

//polygon2//
glBegin(GL_POLYGON);
glColor3ub(0, 100, 0);
glVertex2f(33.5, 4.5);
glVertex2f(34, 5);
glVertex2f(37, 5);
glVertex2f(37.5, 4.5);
//glVertex2f();
```

```
glEnd();

//triangle//
glBegin(GL_TRIANGLES);
glColor3ub(0, 100, 0);
glVertex2f(34, 5);
glVertex2f(37, 5);
glVertex2f(35.5, 5.5);
glEnd();

///tree upper side///

glPushMatrix();
glColor3ub(50, 205, 50);
glTranslatef(-10.5, 2.5, 0);
glutSolidSphere(1.0, 150, 150);
glPopMatrix();

glPushMatrix();
glColor3ub(50, 205, 50);
glTranslatef(-9.5, 3.0, 0);
glutSolidSphere(1.0, 150, 150);
glPopMatrix();

glPushMatrix();
glColor3ub(50, 205, 50);
glTranslatef(-7.5, 1.5, 0);
glutSolidSphere(1.0, 150, 150);
glPopMatrix();

glPushMatrix();
glColor3ub(50, 205, 50);
glTranslatef(-7.5, 2.5, 0);
glutSolidSphere(1.0, 150, 150);
```

```
glPopMatrix();

glPushMatrix();
glColor3ub(50, 205, 50);
glTranslatef(-8.5, 3.5, 0);
glutSolidSphere(1.0, 150, 150);
glPopMatrix();

glPushMatrix();
glColor3ub(50, 205, 50);
glTranslatef(-10.0, 4.0, 0);
glutSolidSphere(1.0, 150, 150);
glPopMatrix();

glPushMatrix();
glColor3ub(50, 205, 50);
glTranslatef(-8.0, 4.0, 0);
glutSolidSphere(1.0, 150, 150);
glPopMatrix();

glPushMatrix();
glColor3ub(50, 205, 50);
glTranslatef(-9.0, 5.0, 0);
glutSolidSphere(1.0, 150, 150);
glPopMatrix();

///boat-1 motion left to right
glPushMatrix();
```

```
//glColor3d(1,0,0);
glTranslatef(_moveA, 0.0f, 0.0f);
glBegin(GL_QUADS);
glColor3ub(139, 69, 19);
glVertex2f(1.0f, -14.0f);
glVertex2f(1.50f, -13.0f);
glVertex2f(-2.0f, -13.0f);
glVertex2f(-3.0f, -14.0f);
glEnd();

glBegin(GL_QUADS);
glColor3ub(0, 0, 0);
glVertex2f(2.0f, -14.5f);
glVertex2f(2.0f, -14.0f);
glVertex2f(-3.0f, -14.0f);
glVertex2f(-3.0f, -14.5f);
glEnd();

glBegin(GL_TRIANGLES);
glColor3ub(0, 0, 0);
glVertex2f(-3.0f, -14.5f);
glVertex2f(-3.0f, -14.0f);
glVertex2f(-4.5f, -13.7f);

glEnd();

glBegin(GL_TRIANGLES);
glColor3ub(0, 0, 0);
glVertex2f(2.0f, -14.5f);
glVertex2f(3.5f, -13.7f);
glVertex2f(2.0f, -14.0f);

glEnd();
```

```
glBegin(GL_TRIANGLES);
glColor3ub(0, 0, 0);
glVertex2f(2.0f, -14.0f);
glVertex2f(1.5f, -13.0f);
glVertex2f(1.0f, -14.0f);

glEnd();

glPopMatrix();









///////////////
///boat-2 motion right to left
glPushMatrix();
//glColor3d(1,0,0);
glTranslatef(_moveB, 0.0f, 0.0f);
glBegin(GL_QUADS);
glColor3ub(139, 69, 19);
glVertex2f(22.0f, -17.5f);
glVertex2f(21.0f, -16.5f);
glVertex2f(17.50f, -16.5f);
glVertex2f(18.0f, -17.5f);
glEnd();

glBegin(GL_QUADS);
glColor3ub(0, 0, 0);
glVertex2f(22.0f, -18.0f);
glVertex2f(22.0f, -17.5f);
glVertex2f(17.0f, -17.5f);
```

```
glVertex2f(17.0f, -18.0f);
glEnd();

glBegin(GL_QUADS);
glColor3ub(255, 99, 71);
glVertex2f(20.5f, -16.5f);
glVertex2f(21.0f, -14.5f);
glVertex2f(18.5f, -14.5f);
glVertex2f(18.0f, -16.5f);
glEnd();

glBegin(GL_QUADS);
glColor3ub(139, 69, 19);
glVertex2f(19.8f, -14.5f);
glVertex2f(19.8f, -14.0f);
glVertex2f(19.7f, -14.0f);
glVertex2f(19.7f, -14.5f);
glEnd();

glBegin(GL_TRIANGLES);
glColor3ub(0, 0, 0);
glVertex2f(17.0f, -18.0f);
glVertex2f(17.0f, -17.5f);
glVertex2f(15.5f, -17.2f);

glEnd();

glBegin(GL_TRIANGLES);
glColor3ub(0, 0, 0);
glVertex2f(22.0f, -17.5f);
glVertex2f(22.0f, -18.0f);
glVertex2f(23.5f, -17.2f);

glEnd();
```

```
glBegin(GL_TRIANGLES);
glColor3ub(0, 0, 0);
glVertex2f(17.0f, -17.5f);
glVertex2f(18.0f, -17.5f);
glVertex2f(17.5f, -16.5f);

glEnd();

glPopMatrix();



///windmill structure///
glBegin(GL_POLYGON);// Draw a Red 1x1 Square centered at origin
glColor3ub(192, 192, 192);
//glColor3ub(0,206,209);

glVertex2f(31.0f, 2.0f);
glVertex2f(30.0f, -6.0f);
glVertex2f(34.0f, -6.0f);
glVertex2f(33.0f, 2.0f);
glVertex2f(32.5f, 3.0f);
glVertex2f(31.5f, 3.0f);



glEnd();


//circle//
glPushMatrix();
glColor3ub(0, 0, 0);
glTranslatef(32.0f, 3.0f, 0.0f);
```

```
glutSolidSphere(0.5, 150, 150);
glRotatef(_angle1, 0.0f, 0.0f, 1.0f);


glBegin(GL_QUADS);// first stand to hold the blade
glColor3ub(255, 255, 0);
glVertex2f(0.0f, 0.0f);
glVertex2f(5.0f, 0.0f);
glVertex2f(5.0f, 0.25f);
glVertex2f(0.0f, 0.25f);
glEnd();

glBegin(GL_QUADS);// second stand to hold the blade
glColor3ub(255, 255, 0);
glVertex2f(0.0f, 0.0f);
glVertex2f(0.0f, 5.0f);
glVertex2f(-0.25f, 5.0f);
glVertex2f(-0.25f, 0.0f);
glEnd();

glBegin(GL_QUADS);// third stand to hold the blade
glColor3ub(255, 255, 0);
glVertex2f(0.0f, -0.25f);
glVertex2f(0.0f, 0.0f);
glVertex2f(-5.0f, 0.0f);
glVertex2f(-5.0f, -0.25f);
glEnd();

glBegin(GL_QUADS);// fourth stand to hold the blade
glColor3ub(255, 255, 0);
glVertex2f(0.25f, 0.0f);
glVertex2f(0.0f, 0.0f);
glVertex2f(0.0f, -5.0f);
glVertex2f(0.25f, -5.0f);
```

```
glEnd();

glBegin(GL_TRIANGLES);// first triangular blade to hold the blade
glColor3ub(128, 0, 0);
glVertex2f(0.25f, 0.0f);
glVertex2f(5.0f, -2.5f);
glVertex2f(5.0f, 0.0f);
glEnd();

glBegin(GL_TRIANGLES);// second triangular blade to hold the blade
glColor3ub(128, 0, 0);
glVertex2f(0.0f, 0.025f);
glVertex2f(2.5f, 5.0f);
glVertex2f(0.0f, 5.0f);
glEnd();

glBegin(GL_TRIANGLES);// third triangular blade to hold the blade
glColor3ub(128, 0, 0);
glVertex2f(-0.25f, 0.0f);
glVertex2f(-5.0f, 2.5f);
glVertex2f(-5.0f, 0.0f);
glEnd();

glBegin(GL_TRIANGLES);// fourth triangular blade to hold the blade
glColor3ub(128, 0, 0);
glVertex2f(0.0f, -0.025f);
glVertex2f(-2.5f, -5.0f);
glVertex2f(0.0f, -5.0f);
glEnd();
```

```
    glPopMatrix();









    glutSwapBuffers();
}









void SpecialInput(int key, int x, int y) // day and night
{
    switch (key)
    {
    case GLUT_KEY_UP:

        glutDisplayFunc(drawScene);
        break;
    case GLUT_KEY_DOWN:

        glutDisplayFunc(drawScene2);
        break;

    }
    glutPostRedisplay(); // redraw or update the screen
}
```

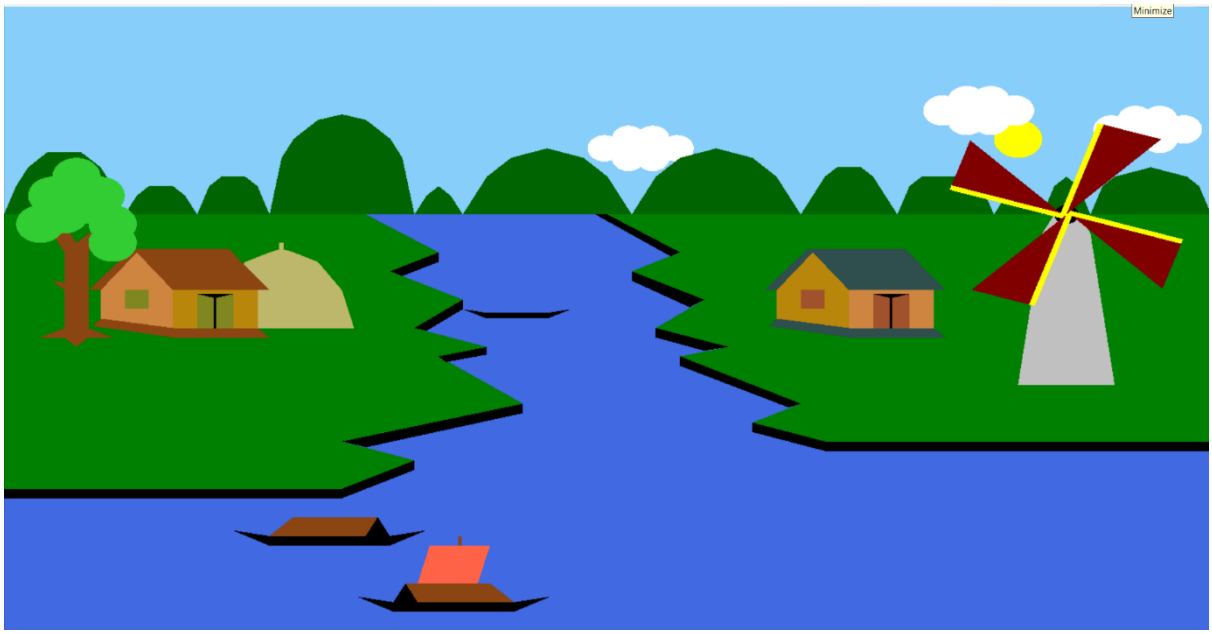
```
void update1(int value) { // boat 1 motion

    _moveA += speed;
    if (_moveA > 38)
    {
        _moveA = -38;
    }
    // If the object goes too far(off the screen), reset its position back to the start.

//_moveA += 0.11;

    glutTimerFunc(20, update1, 0); //Updates the Update 1 function
    glutPostRedisplay();
}

void handleMouse(int button, int state, int x, int y) { //using mouse to increase or decrease speed

    if (button == GLUT_LEFT_BUTTON)
    {
        speed += 0.05f;
    }



    else if (button == GLUT_RIGHT_BUTTON)
    {
        speed -= 0.05f;
    }
    glutPostRedisplay();
}
void handleKeypress(unsigned char key, int x, int y) { // using keyboard to increase or decrease speed
    switch (key) {
    case 's'://stops
```

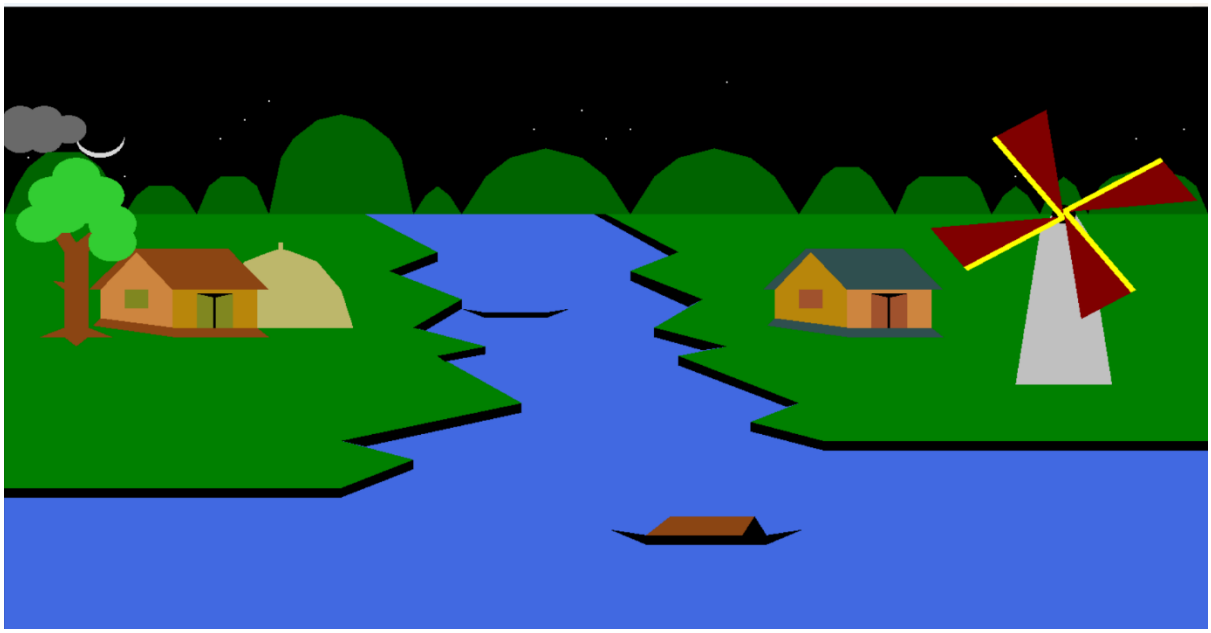
```
        speed = 0.0f;
        break;
    case 'r'://runs
        speed = 0.02f;
        break;
        glutPostRedisplay();
    }
}

void update2(int value) { //boat 2 motion


    if (_moveB < -36)
    {
        _moveB = +38;
    }
    //creates loop

    _moveB -= 0.13;

    glutTimerFunc(20, update2, 0); //updates update2 periodically. 0 is a parameter passed.
    glutPostRedisplay();
}

void update3(int value) {//function for windlmill

    _angle1 -= 2.0f;
    if (_angle1 > 360.0)
    {
        _angle1 -= 360;
    }
    glutPostRedisplay(); //update the display

    glutTimerFunc(10, update3, 0); //update
```

```
}
void update4(int value) {//cloud motion


    if (_moveC > +36)
    {
        _moveC = -38;
    }
    //loop


    _moveC += 0.13;


    glutTimerFunc(20, update4, 0); //update
    glutPostRedisplay();
}









int main(int argc, char** argv) {
    glutInit(&argc, argv); // initalise GLUT library and prepare the GLUT to handle
    glutInitDisplayMode(GLUT_DOUBLE | GLUT_RGB); // double buffering (remove flickering) and RGB
mode
    glutInitWindowSize(1500, 1000);
    glutCreateWindow("Transformation");
    init();
    glutSpecialFunc(SpecialInput); // takes special characters for day and night
    glutDisplayFunc(drawScene);// tells to apply drawscene funtion
    glutTimerFunc(20, update1, 0); //initiate moves boat 1; call periodically
    glutTimerFunc(20, update2, 0); //initiate moves boat 2; call periodically


    glutTimerFunc(10, update3, 0);// initiate moves windmill
```

```
    glutTimerFunc(20, update4, 0);// initiate moves cloud
    glutKeyboardFunc(handleKeypress); //run boat1 using keyboard
    glutMouseFunc(handleMouse); // run boat2 using mouse
    glutMainLoop();  // handles all user input and main function
    return 0;
}
```

Screenshots





References

- OpenGL Official Documentation – https://www.khronos.org/opengl/
- GeeksforGeeks: Introduction to Computer Graphics – https://www.geeksforgeeks.org/introduction-to-computer-graphics/?
- LearnOpenGL: Modern OpenGL Tutorials – https://learnopengl.com/
- Stack Overflow: Solution to object rendering issue – https://stackoverflow.com/questions/12345678/rendering-issue-in-opengl
- “Computer Graphics Principles and Practice” by John F. Hughes, 3rd Edition
- YouTube – Computer Graphics Project Tutorials using OpenGL
- GitHub – Sample OpenGL Project Code