

Mini Project

River side Scenery

Student Name: Abinash Kakoti

Reg: 11912480

Roll: 53

Section: KE076

Submitted to

Dr. Om Prakash Yadav (26121)

Asst. Professor,

School of Computer Science and Engineering



**LOVELY PROFESSIONAL UNIVERSITY PHAGWARA,
PUNJAB**

March, 2023

Code

```
#include<windows.h>
#include <GL/glut.h>
# define PI 3.14159265358979323846
#include <math.h>
#include<string>

void drawText(const char *text, int length, int x, int y){

    glMatrixMode(GL_PROJECTION);
    double *matrix = new double[16];
    glGetDoublev(GL_PROJECTION_MATRIX, matrix);

    glLoadIdentity();
    glOrtho(0, 85, 0, 82, 0, 1);
    glMatrixMode(GL_MODELVIEW);
    glLoadIdentity();
    glPushMatrix();
    glLoadIdentity();
    //glColor3f(r, g, b);
    glRasterPos2i(x, y);
    for (int i=0; i<length; i++)
        glutBitmapCharacter(GLUT_BITMAP_TIMES_ROMAN_24, (int)text[i]);
    glPopMatrix();
    glMatrixMode(GL_PROJECTION);
    glLoadMatrixd(matrix);
    glMatrixMode(GL_MODELVIEW);
}

int posx = 0;
void init(void)
{
    glClearColor(1.0, 1.0, 1.0, 0.0); // Set display window colour to white

    glMatrixMode(GL_PROJECTION); // Set projection parameters
    gluOrtho2D(0.0, 85.0, 0.0, 82.0);

}
```

```
void line_drawing(float a, float b, float c, float d)
{
    // Draw an outlined triangle
    glBegin(GL_LINES);

        glVertex2f(a, b);
        glVertex2f(c, d);

    glEnd();
}
```

```
void quads(float a, float b, float c, float d, float e, float f, float g, float h){

    glBegin(GL_QUADS); //Begin quadrilateral coordinates

        //Trapezoid
        glVertex2f(a, b);
        glVertex2f(c,d);
        glVertex2f(e, f);
        glVertex2f(g, h);

        glEnd(); //End quadrilateral coordinates

}
```

```
void triangle(float a, float b, float c, float d, float e, float f){
    glBegin(GL_TRIANGLES);
    //Triangle
        glVertex2f(a, b);
        glVertex2f(c, d);
        glVertex2f(e, f);

        glEnd();//End triangle coordinates

}
```

```

void pentagon(int a, int b, int c, int d, int e, int f, int g, int h, int i, int j, int k, int l, int
m, int n, int o, int p, int q, int r)
{
    glBegin(GL_TRIANGLES); //Begin triangle coordinates

    //Pentagon
    glVertex2i(a, b);
    glVertex2i(c, d);
    glVertex2i(e, f);

    glVertex2i(g, h);
    glVertex2i(i, j);
    glVertex2i(k, l);

    glVertex2i(m, n);
    glVertex2i(o, p);
    glVertex2i(q, r);

    glEnd();//End triangle coordinates
}

///river
void river()
{
    glColor3f(0.0352941176470588, 0.5098039215686275, 0.9568627450980392);
    glBegin(GL_POLYGON);

    glVertex3i(0, 0, 0);
    glVertex3i(0, 12, 0);
    glVertex3i(50, 12, 0);
    glVertex3i(49, 0, 0);

    glEnd();

}

```

```
void boat()
{
    quads(3.0,20.0,3.0,24.0,5.0,24.0,5.0,20.0);
    quads(1.4,16.0,1.4,24.0,3.0,24.0,3.0,16.0);
    glColor3f(0,0,0);
    quads(2.5,24.0,2.5,28.0,4.0,28.0,4.0,24.0);
    glColor3f(1,0,1);
    quads(5.5,20.0,5.5,22.0,11.0,22.0,11.0,20.0);
```

```
    glColor3f(0,0,0);
```

```
    glBegin(GL_POLYGON);
```

```
        glVertex3i(4,12,0);
        glVertex3i(1,16,0);
        glVertex3i(15,16,0);
        glVertex3i(11,12,0);
        glEnd();
        glColor3f(1,0,0);
        glBegin(GL_POLYGON);
        glVertex3i(3,16,0);
        glVertex3i(3,20,0);
        glVertex3i(13,20,0);
        glVertex3i(13,16,0);
        glEnd();
        glColor3f(1,0.7,0);
        glBegin(GL_TRIANGLES);
```

```
            glVertex3i(13,16,0);
            glVertex3i(13,20,0);
            glVertex3i(15,16,0);
```

```
        glEnd();
```

```
    }
```

```

void update(int value) {

    if(posx>=38)
    {
        posx=0;
    }
    else{
        posx++;
    }
    glutPostRedisplay();
    glutTimerFunc(250, update, 0);
}

void tree()
{
    ///tree1
    glColor3f(0.6156863,0,0);
    glBegin(GL_POLYGON);
    glVertex3i(55, 12, 0);
    glVertex3i(57, 12, 0);
    glVertex3i(57, 30, 0);
    glVertex3i(55, 30, 0);
    glEnd();
    glColor3f(0.0, 0.5, 0.0);
    glBegin(GL_POLYGON);
    glVertex3i(51, 30, 0);
    glVertex3i(61, 30, 0);
    glVertex3i(56, 40, 0);
    glEnd();
    glBegin(GL_POLYGON);
    glVertex3i(51, 35, 0);
    glVertex3i(60, 35, 0);
    glVertex3i(56, 45, 0);
    glEnd();

}

```

```

void tree2()
{
    ///tree1
    glColor3f(0.6156863,0,0);
    glBegin(GL_POLYGON);
    glVertex3i(52, 0, 0);
    glVertex3i(54, 0, 0);
    glVertex3i(54, 18, 0);
    glVertex3i(52, 18, 0);
    glEnd();
    glColor3f(0.0, 0.4, 0.0);
    glBegin(GL_POLYGON);
    glVertex3i(48, 18, 0);
    glVertex3i(58, 18, 0);
    glVertex3i(53, 28, 0);
    glEnd();
    glBegin(GL_POLYGON);
    glVertex3i(48, 23, 0);
    glVertex3i(57, 23, 0);
    glVertex3i(53, 33, 0);
    glEnd();

    glColor3f(0.6156863,0,0);
    glBegin(GL_POLYGON);
    glVertex3i(82, 0, 0);
    glVertex3i(84, 0, 0);
    glVertex3i(84, 18, 0);
    glVertex3i(82, 18, 0);
    glEnd();
    glColor3f(0.0, 0.4, 0.0);
    glBegin(GL_POLYGON);
    glVertex3i(78, 18, 0);
    glVertex3i(88, 18, 0);
    glVertex3i(83, 28, 0);
    glEnd();
    glBegin(GL_POLYGON);
    glVertex3i(78, 23, 0);
    glVertex3i(87, 23, 0);
    glVertex3i(83, 33, 0);

```

```
glEnd();
```

```
}
```

```
void tree1()
```

```
{
```

```
    ///tree1
```

```
    glColor3f(0.6156863,0,0);
```

```
    glBegin(GL_POLYGON);
```

```
    glVertex3i(65, 12, 0);
```

```
    glVertex3i(67, 12, 0);
```

```
    glVertex3i(67, 30, 0);
```

```
    glVertex3i(65, 30, 0);
```

```
    glEnd();
```

```
    glColor3f(0.0, 0.5, 0.0);
```

```
    glBegin(GL_POLYGON);
```

```
    glVertex3i(61, 30, 0);
```

```
    glVertex3i(71, 30, 0);
```

```
    glVertex3i(66, 40, 0);
```

```
    glEnd();
```

```
    glBegin(GL_POLYGON);
```

```
    glVertex3i(61, 35, 0);
```

```
    glVertex3i(70, 35, 0);
```

```
    glVertex3i(66, 45, 0);
```

```
glEnd();
```

```
glColor3f(0.6156863,0,0);
```

```
    glBegin(GL_POLYGON);
```

```
    glVertex3i(75, 12, 0);
```

```
    glVertex3i(77, 12, 0);
```

```
    glVertex3i(77, 30, 0);
```

```
    glVertex3i(75, 30, 0);
```

```
    glEnd();
```

```
    glColor3f(0.0, 0.5, 0.0);
```

```
    glBegin(GL_POLYGON);
```

```
    glVertex3i(71, 30, 0);
```

```
    glVertex3i(81, 30, 0);
```

```
    glVertex3i(76, 40, 0);
```

```
    glEnd();
```



```
glBegin(GL_POLYGON);
glVertex3i(71, 35, 0);
glVertex3i(80, 35, 0);
glVertex3i(76, 45, 0);
glEnd();
```

```
glColor3f(0.6156863,0,0);
glBegin(GL_POLYGON);
glVertex3i(85, 12, 0);
glVertex3i(87, 12, 0);
glVertex3i(87, 30, 0);
glVertex3i(85, 30, 0);
glEnd();
glColor3f(0.0, 0.5, 0.0);
glBegin(GL_POLYGON);
glVertex3i(81, 30, 0);
glVertex3i(91, 30, 0);
glVertex3i(86, 40, 0);
glEnd();
glBegin(GL_POLYGON);
glVertex3i(81, 35, 0);
glVertex3i(90, 35, 0);
glVertex3i(86, 45, 0);
glEnd();
```

```
}
```

```
void sun()
```

```
{
```

```
    int i;
```

```
    GLfloat x=48; GLfloat y=45; GLfloat radius =8;
    int triangleAmount = 20;
    GLfloat twicePi = 2 * PI;
```

```
    glBegin(GL_TRIANGLE_FAN);
        glColor3ub(255, 85, 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 cloud()
{
    int i;

    GLfloat x=20; GLfloat y=60; GLfloat radius =3;
    int triangleAmount = 20;
    GLfloat twicePi = 2 * PI;

    glBegin(GL_TRIANGLE_FAN);
        glColor3ub(255,191,179);
        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 x1=23; GLfloat y1=62; GLfloat radius1 =3;
    glBegin(GL_TRIANGLE_FAN);
        glColor3ub(255,191,179);
        glVertex2f(x1, y1); // center of circle
        for(i = 0; i <= triangleAmount;i++) {
            glVertex2f(
                x1 + (radius1 * cos(i * twicePi / triangleAmount)),
                y1 + (radius1 * sin(i * twicePi / triangleAmount))
            );
        }
    }

```

```
}
```

```
glEnd();
```

```
GLfloat x2=27; GLfloat y2=61; GLfloat radius2 =4;
```

```
glBegin(GL_TRIANGLE_FAN);
```

```
glColor3ub(255,191,179);
```

```
glVertex2f(x2, y2); // center of circle
```

```
for(i = 0; i <= triangleAmount;i++) {
```

```
glVertex2f(
```

```
    x2 + (radius2 * cos(i * twicePi / triangleAmount)),
```

```
    y2 + (radius2 * sin(i * twicePi / triangleAmount))
```

```
);
```

```
}
```

```
glEnd();
```

```
GLfloat x3=30; GLfloat y3=59; GLfloat radius3 =3;
```

```
glBegin(GL_TRIANGLE_FAN);
```

```
glColor3ub(255,191,179);
```

```
glVertex2f(x3, y3); // center of circle
```

```
for(i = 0; i <= triangleAmount;i++) {
```

```
glVertex2f(
```

```
    x3 + (radius3 * cos(i * twicePi / triangleAmount)),
```

```
    y3 + (radius3 * sin(i * twicePi / triangleAmount))
```

```
);
```

```
}
```

```
glEnd();
```

```
GLfloat x4=27; GLfloat y4=59; GLfloat radius4 =3;
```

```
glBegin(GL_TRIANGLE_FAN);
```

```
glColor3ub(255,191,179);
```

```
glVertex2f(x4, y4); // center of circle
```

```
for(i = 0; i <= triangleAmount;i++) {
```

```
glVertex2f(
```

```
    x4 + (radius4 * cos(i * twicePi / triangleAmount)),
```

```
    y4 + (radius4 * sin(i * twicePi / triangleAmount))
```

```
);
```

```
}
```

```
glEnd();
```

```
GLfloat x5=23; GLfloat y5=58; GLfloat radius5 =3;
```

```
glBegin(GL_TRIANGLE_FAN);
```

```
    glColor3ub(255,191,179);
```

```
    glVertex2f(x5, y5); // center of circle
```

```
    for(i = 0; i <= triangleAmount;i++) {
```

```
        glVertex2f(
```

```
            x5 + (radius5 * cos(i * twicePi / triangleAmount)),
```

```
            y5 + (radius5 * sin(i * twicePi / triangleAmount))
```

```
        );
```

```
    }
```

```
glEnd();
```

```
}
```

```
void cloud2()
```

```
{
```

```
    int i;
```

```
    GLfloat x=50; GLfloat y=65; GLfloat radius =3;
```

```
    int triangleAmount = 20;
```

```
    GLfloat twicePi = 2 * PI;
```

```
    glBegin(GL_TRIANGLE_FAN);
```

```
        glColor3ub(255,191,179);
```

```
        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 x1=53; GLfloat y1=67; GLfloat radius1 =3;
```

```

glBegin(GL_TRIANGLE_FAN);
    glColor3ub(255,191,179);
    glVertex2f(x1, y1); // center of circle
    for(i = 0; i <= triangleAmount;i++) {
        glVertex2f(
            x1 + (radius1 * cos(i * twicePi / triangleAmount)),
            y1 + (radius1 * sin(i * twicePi / triangleAmount))
        );
    }

```

```

glEnd();

```

```

GLfloat x2=57; GLfloat y2=66; GLfloat radius2 =4;
glBegin(GL_TRIANGLE_FAN);
    glColor3ub(255,191,179);
    glVertex2f(x2, y2); // center of circle
    for(i = 0; i <= triangleAmount;i++) {
        glVertex2f(
            x2 + (radius2 * cos(i * twicePi / triangleAmount)),
            y2 + (radius2 * sin(i * twicePi / triangleAmount))
        );
    }

```

```

glEnd();

```

```

GLfloat x3=60; GLfloat y3=64; GLfloat radius3 =3;
glBegin(GL_TRIANGLE_FAN);
    glColor3ub(255,191,179);
    glVertex2f(x3, y3); // center of circle
    for(i = 0; i <= triangleAmount;i++) {
        glVertex2f(
            x3 + (radius3 * cos(i * twicePi / triangleAmount)),
            y3 + (radius3 * sin(i * twicePi / triangleAmount))
        );
    }

```

```

glEnd();

```

```

GLfloat x4=57; GLfloat y4=64; GLfloat radius4 =3;

```

```

glBegin(GL_TRIANGLE_FAN);
    glColor3ub(255,191,179);
    glVertex2f(x4, y4); // center of circle
    for(i = 0; i <= triangleAmount;i++) {
        glVertex2f(
            x4 + (radius4 * cos(i * twicePi / triangleAmount)),
            y4 + (radius4 * sin(i * twicePi / triangleAmount))
        );
    }

```

```

glEnd();

```

```

GLfloat x5=53; GLfloat y5=63; GLfloat radius5 =3;
glBegin(GL_TRIANGLE_FAN);
    glColor3ub(255,191,179);
    glVertex2f(x5, y5); // center of circle
    for(i = 0; i <= triangleAmount;i++) {
        glVertex2f(
            x5 + (radius5 * cos(i * twicePi / triangleAmount)),
            y5 + (radius5 * sin(i * twicePi / triangleAmount))
        );
    }

```

```

glEnd();

```

```

}

```

```

void backgroundtree()

```

```

{

```

```

    glBegin(GL_POLYGON);
    glColor3ub(102,68,0);
    glVertex2f(10,12);
    glVertex2f(5,35);
    glVertex2f(2,47);
    glVertex2f(-1,52);
    glVertex2f(-4,47);
    glVertex2f(-7,35);
    glVertex2f(-8,12);
    glVertex2f(55,12);
    glEnd();

```

```
glBegin(GL_POLYGON);  
glColor3ub(102,68,0);  
glVertex2f(45,12);  
glVertex2f(38,40);  
glVertex2f(31,52);  
glVertex2f(24,57);  
glVertex2f(18,52);  
glVertex2f(11,40);  
glVertex2f(6,12);  
glVertex2f(45,12);  
glEnd();
```

```
glBegin(GL_POLYGON);  
glColor3ub(102,68,0);  
glVertex2f(75,12);  
glVertex2f(68,30);  
glVertex2f(61,42);  
glVertex2f(54,48);  
glVertex2f(48,42);  
glVertex2f(41,30);  
glVertex2f(36,12);  
glVertex2f(75,12);  
glEnd();
```

```
glBegin(GL_POLYGON);  
glColor3ub(102,68,0);  
glVertex2f(105,12);  
glVertex2f(98,40);  
glVertex2f(91,52);  
glVertex2f(84,57);  
glVertex2f(78,52);  
glVertex2f(71,40);  
glVertex2f(66,12);  
glVertex2f(105,12);  
glEnd();
```

```
glBegin(GL_POLYGON);  
glColor3ub(125,85,0);
```

```
glVertex2f(85,12);  
glVertex2f(80,35);  
glVertex2f(75,47);  
glVertex2f(70,52);  
glVertex2f(65,47);  
glVertex2f(60,35);  
glVertex2f(55,12);  
glVertex2f(85,12);  
glEnd();
```

```
glBegin(GL_POLYGON);  
glColor3ub(125,85,0);  
glVertex2f(55,12);  
glVertex2f(48,35);  
glVertex2f(41,47);  
glVertex2f(34,52);  
glVertex2f(28,47);  
glVertex2f(21,35);  
glVertex2f(16,12);  
glVertex2f(55,12);  
glEnd();
```

```
glBegin(GL_POLYGON);  
glColor3ub(125,85,0);  
glVertex2f(16,12);  
glVertex2f(13,35);  
glVertex2f(10,47);  
glVertex2f(7,52);  
glVertex2f(3,47);  
glVertex2f(1,35);  
glVertex2f(0,12);  
glVertex2f(16,12);  
glEnd();
```

```
}
```



```

void drawShapes(void)
{
    glClear(GL_COLOR_BUFFER_BIT); // Clear display window

    // glColor3f(0.0f, 0.1f, 0.0f); // Forest Green
    glColor3f(0.0f, 1.0f, 0.0f); // Green

    quads(30.0,0.0,30.0,13.0,90.0,13.0,90.0,0.0); //

    glColor3f(0.0f, 1.0f, 1.0f);
    //sky

    glColor3f(0.0,0.9,0.9);
    quads(0.0,12.0,0.0,85.0,85.0,85.0,85.0,12.0);

    sun();

    cloud2();
    backgroundtree();
    cloud();
    glColor3ub(0,0,0);
    std::string text = "Abinash, 11912480";
    drawText(text.data(),text.size(), 70, 70);

    tree();
    tree1();

    //glColor3f(0.0f, 1.0f, 0.0f); // Green
    //quads(4,11.0,4.0,32.0,6.0,32.0,6.0,11.0);

```

```
//Set colour to black
glColor3f(0.0, 0.0, 0.0);
//Adjust the point size
glPointSize(5.0);
```

```
glColor3ub(255,25,140);
quads(69.0, 6.0, 69.0, 28.0, 77.0, 28.0, 77.0, 6.0 );
```

```
//Set colour to red
glColor3f(1.0, 0.0, 0.0);
```

```
line_drawing(69.0,6.0,69.0,16.0);
line_drawing(69.0,16.0,77.0,16.0);
line_drawing(77.0,16.0,77.0,6.0);
line_drawing(69.0,6.0,77.0,6.0);
quads(70.5,10.0,70.5,12.0,71.5,12.0,71.5,10.0);
glColor3f(1.0, 0.0, 1.0);
quads(71.5,10.0,71.5,12.0,72.0,12.0,72.0,10.0);
glColor3f(0.0, 0.0, 1.0);
quads(72.0,10.0,72.0,12.0,73.0,12.0,73.0,10.0);
line_drawing(75.0,6.0,75.0,10.0);
line_drawing(75.0,10.0,76.7,10.0);
line_drawing(76.7,10.0,76.7,6.0);
quads(75.5,9.7,76.5,9.7,76.5,6.5,75.5,6.5);
```

```
line_drawing(69.0, 16.0 ,69.0,28.0);
line_drawing(69.0, 28.0 ,77.0,28.0);
line_drawing(77.0, 28.0 ,77.0,16.0);
quads(70.0,20.0,70.0,22.0,72.0,22.0,72.0,20.0);
quads(73.0,20.0,73.0,22.0,75.0,22.0,75.0,20.0);
glColor3f(1.0, 0.50, 1.0);
quads(77.0,6.0,77.0,38.0,82.0,38.0,82.0,6.0);
glColor3f(0.50, 1.0, 0.0);
triangle(77.0,38.0,79.0,48.0,82.0,38.0);
glColor3f(1.0, 0.60, 1.0);
//tr
```

```
// glColor3f(0.0, 0.60, 1.0);
```

```
glColor4f(1.0f, 0.0f, 0.0f, 0.0f); //red
```

```
//moi
```

```
line_drawing(63.0, 6.0 ,68.0,28.0);
```

```
line_drawing(64.0, 6.0 ,69.0,28.0);
```

```
line_drawing(63.3, 8.0 ,64.6,8.0);
```

```
line_drawing(64.0, 10.0 ,64.9,10.0);
```

```
line_drawing(64.3, 12.0 ,65.3,12.0);
```

```
line_drawing(64.9, 14.0 ,65.9,14.0);
```

```
line_drawing(65.2, 16.0 ,66.2,16.0);
```

```
line_drawing(65.8, 18.0 ,66.8,18.0);
```

```
line_drawing(66.1, 20.0 ,67.2,20.0);
```

```
line_drawing(66.4, 22.0 ,67.6,22.0);
```

```
line_drawing(67.0, 24.0 ,68.0,24.0);
```

```
line_drawing(67.4, 26.0 ,68.5,26.0);
```

```
line_drawing(48.0, 6.0 ,49.3,7.9);
```

```
glColor3f(1.0f, 1.89f, 0.89f);
```

```
triangle(55.0,25.0,58.0,35.0,61.0,25.0);
```

```
glColor3f(0.5411764705882353, 0.0, 0.5411764705882353);
```

```
quads(55.0,12.0,55.0,25.0,61.0,25.0,61.0,12.0);
```

```
river();
```

```
// pentagon();
```

```
glPushMatrix();
```

```
glTranslatef(posx,-1,-1);
```

```
boat();
```

```
glPopMatrix();
```

```
tree2();
```

```

    glFlush(); // Process all OpenGL routines
}

int main(int argc, char* argv[])
{
    glutInit(&argc, argv);                // Initialise GLUT
    glutInitDisplayMode(GLUT_SINGLE|GLUT_RGB); // Set display mode

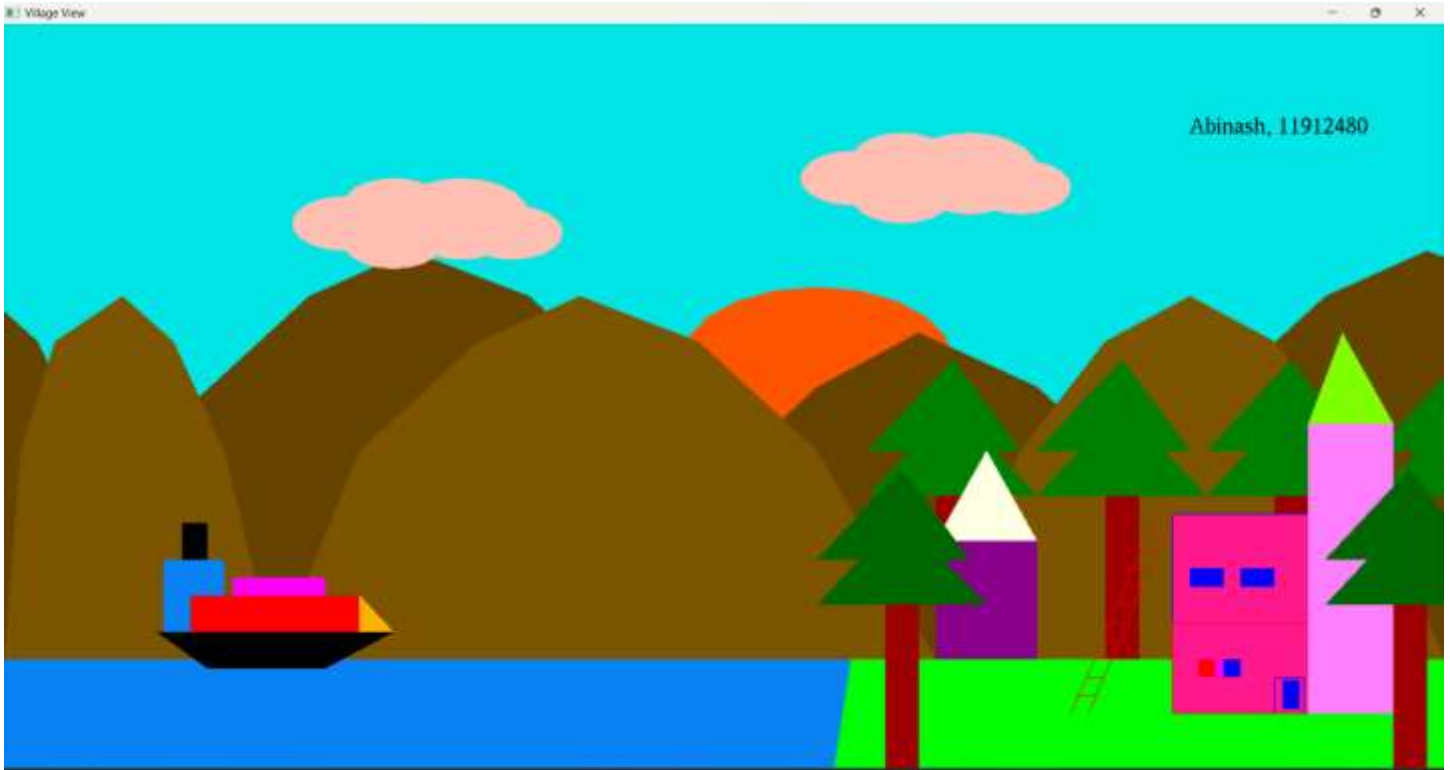
    glutInitWindowPosition(50, 100);      // Set window
position
    glutInitWindowSize(600, 600);        // Set window size
    glutCreateWindow("Village View");    // Create display window

    init();                               // Execute initialisation
procedure
    glutDisplayFunc(drawShapes);
    //glutTimerFunc(100, cloud, 0);
    glutTimerFunc(0, update, 0);        // Send graphics to display window
    glutMainLoop();                     // Display everything and wait

    return 0;
}

```

Screenshots





Future Enhancement

This project can be improved by adding many other elements like

- Birds with Animation.
- Animation to clouds.
- Color shading.
- Wind effect.
- Rain effect.

References

Computer Graphics with OpenGL by Hearn, Baker and Carithers
4th edition.

- Thank You -