

CS352 Assignment-1

Krishanu Saini

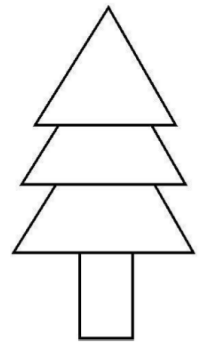
190001029

Write a program using glut library to draw the following diagrams:

Problem 1: Tree

Solution

Draw 3 overlapping isosceles triangles from bottom to top with green color.
Draw the tree trunk with brown color.
Draw the points at the corners in black.



Code

```
#include <GL/glut.h>

/*
 * Name: Krishanu Saini
 * Roll: 190001029
 * Ques: Problem 1) Tree
 * Date: 13/01/22
 */

void Draw() {
    glClear(GL_COLOR_BUFFER_BIT);
    glPointSize(10);

    /*----- corners -----*/
    glColor3f(0,0,0);
    glBegin(GL_POINTS);
    glVertex2i(0,0);
    glVertex2i(0,800);
    glVertex2i(800,800);
    glVertex2i(800,0);
    glEnd();

    /*----- Leaves -----*/
    glColor3f(0.1,0.8,0.1);
    glBegin(GL_POLYGON);
    glVertex2i(400, 450);
    glVertex2i(220, 250);
    glVertex2i(580, 250);
    glEnd();

    glBegin(GL_POLYGON);
    glVertex2i(400, 550);
    glVertex2i(240, 350);
    glVertex2i(560, 350);
    glEnd();

    glBegin(GL_POLYGON);
    glVertex2i(400, 650);
    glVertex2i(260, 450);
    glVertex2i(540, 450);
```

```

    glEnd();

    /*----- wood -----*/
    glColor3f(0.6,0.3,0);
    glBegin(GL_POLYGON);
    glVertex2i(350, 100);
    glVertex2i(350, 250);
    glVertex2i(450, 250);
    glVertex2i(450, 100);
    glEnd();

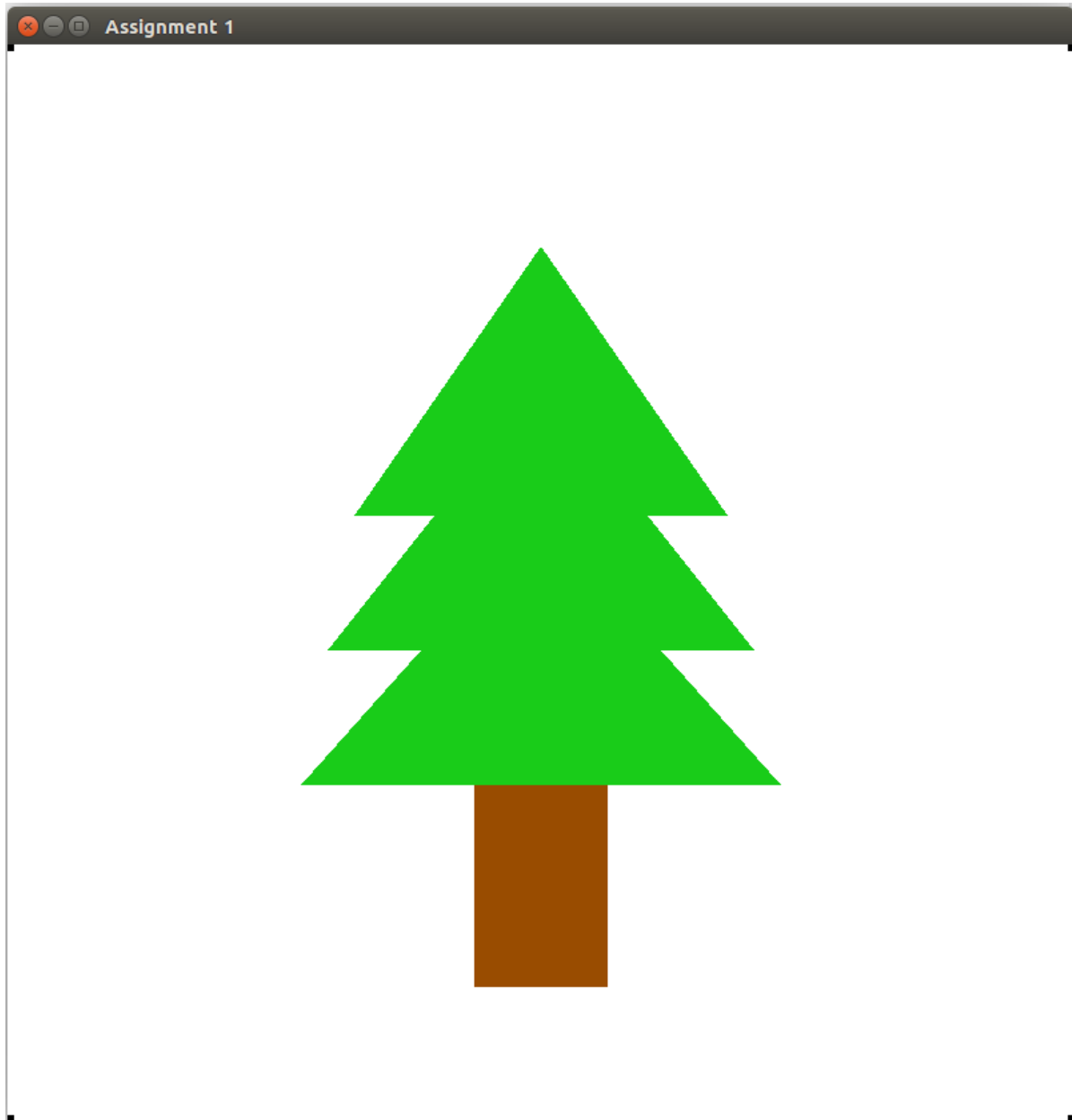
    glFlush();
}

int main(int argc, char *argv[]) {
    glutInit(&argc, argv);
    glutInitWindowPosition(100, 100);
    glutInitWindowSize(800, 800);
    glutInitDisplayMode(GLUT_RGB);

    glutCreateWindow("Assignment 1");
    gluOrtho2D(0, 800, 0, 800);
    glClearColor(1,1,1,0.0);
    glutDisplayFunc(Draw);
    glutMainLoop();
    return 0;
}

```

Output



Problem 2: House

Solution

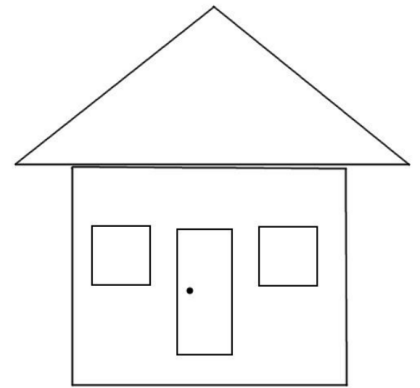
Draw the house in layers.

Draw the wall rectangle.

Draw the roof as a triangle.

Draw the windows square and door rectangle.

Draw the door knob as a rounded point.



Code

```
#include <GL/glut.h>
```

```
/*
```

```
 * Name: Krishanu Saini
```

```
 * Roll: 190001029
```

```
 * Ques: Problem 2) House
```

```
 * Date: 13/01/22
```

```
*/
```

```
void Draw() {
```

```
    glClear(GL_COLOR_BUFFER_BIT);
```

```
    glPointSize(10);
```

```
    /*----- big_rectangle -----*/
```

```
    glColor3f(0.5, 0.3, 0);
```

```
    glBegin(GL_POLYGON);
```

```
    glVertex2i(200, 100);
```

```
    glVertex2i(200, 400);
```

```
    glVertex2i(600, 400);
```

```
    glVertex2i(600, 100);
```

```
    glEnd();
```

```
    /*----- roof_triangle -----*/
```

```
    glColor3f(0.5, 0.3, 0.5);
```

```
    glBegin(GL_POLYGON);
```

```
    glVertex2i(100, 401);
```

```
    glVertex2i(400, 700);
```

```
    glVertex2i(700, 401);
```

```
    glEnd();
```

```
    /*----- door -----*/
```

```
    glColor3f(0, 0.2, 0.6);
```

```
    glBegin(GL_POLYGON);
```

```
    glVertex2i(360, 150);
```

```
    glVertex2i(360, 320);
```

```
    glVertex2i(440, 320);
```

```
    glVertex2i(440, 150);
```

```
    glEnd();
```

```
    /*----- door_knob -----*/
```

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

```
    glEnable(GL_POINT_SMOOTH);
```

```

glHint(GL_POINT_SMOOTH_HINT, GL_NICEST);
glBegin(GL_POINTS);
glPointSize(10);
glVertex2i(370, 240);
glEnd();

/*----- window_left -----*/
glColor3f(0.2, 0.8, 1);
glBegin(GL_POLYGON);
glVertex2i(240, 240);
glVertex2i(240, 320);
glVertex2i(320, 320);
glVertex2i(320, 240);
glEnd();

/*----- window_right -----*/
glColor3f(0.2, 0.8, 1);
glBegin(GL_POLYGON);
glVertex2i(480, 240);
glVertex2i(480, 320);
glVertex2i(560, 320);
glVertex2i(560, 240);
glEnd();

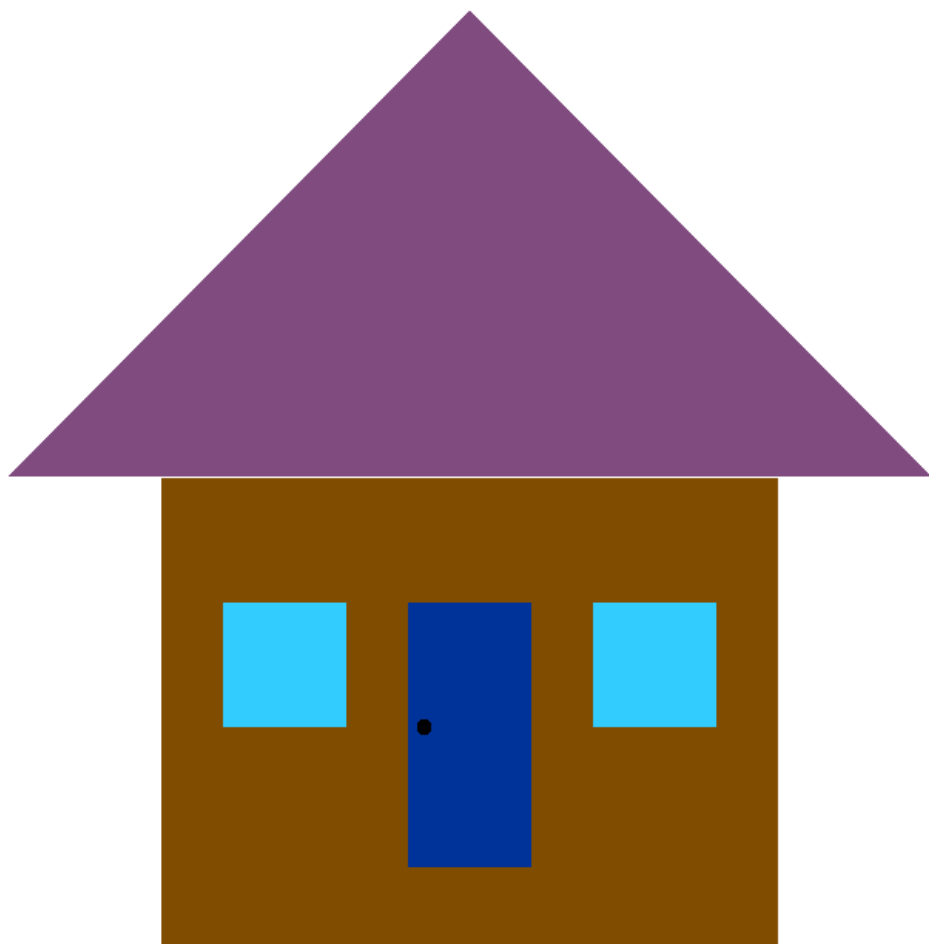
glFlush();
}

int main(int argc, char *argv[]) {
    glutInit(&argc, argv);
    glutInitWindowPosition(100, 100);
    glutInitWindowSize(800, 800);
    glutInitDisplayMode(GLUT_RGB);

    glutCreateWindow("Assignment 1");
    gluOrtho2D(0, 800, 0, 800);
    glClearColor(1,1,1,0.0);
    glutDisplayFunc(Draw);
    glutMainLoop();
    return 0;
}

```

Output



Problem 3: Bicycle

Solution

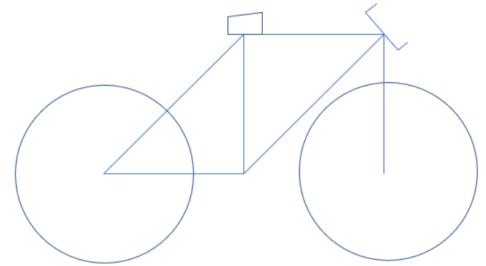
We will draw the bicycle in parts.

Draw base parallelogram and 2 lines through it.

Draw wheels as circles with the center at the bicycle frame.

Draw a seat as trapezium.

Draw handle as 45 degrees inclined.



Code

```
#include <math.h>
#include <GL/glut.h>

/*
 * Name: Krishanu Saini
 * Roll: 190001029
 * Ques: Problem 3) Bicycle
 * Date: 13/01/22
 */

void DrawCircle(float cx, float cy, float r, float num_segments) {
    glBegin(GL_LINE_LOOP);
    for (int ii = 0; ii < num_segments; ii++) {
        float theta = 2.0f * 3.1415926f * float(ii) / float(num_segments); //get the current angle
        float x = r * cosf(theta); //calculate the x component
        float y = r * sinf(theta); //calculate the y component
        glVertex2f(x + cx, y + cy); //output vertex
    }
    glEnd();
}

void Draw() {
    glClear(GL_COLOR_BUFFER_BIT);
    glPointSize(10);

    /*----- bars -----*/
    glColor3f(0.1, 0.1, 0.8);
    glLineWidth(2);
    glBegin(GL_LINE_LOOP);
    glVertex2i(200, 300);
    glVertex2i(400, 600);
    glVertex2i(600, 600);
    glVertex2i(400, 300);
    glVertex2i(200, 300);
    glEnd();

    glBegin(GL_LINES);
    glVertex2i(400, 300);
    glVertex2i(400, 600);
    glEnd();
}
```

```

glBegin(GL_LINES);
glVertex2i(600, 600);
glVertex2i(600, 300);
glEnd();

/*----- seat -----*/
glColor3f(0.5,0.2,0);
glLineWidth(4);
glBegin(GL_POLYGON);
glVertex2i(380, 600);
glVertex2i(450, 600);
glVertex2i(450, 630);
glVertex2i(380, 620);
glEnd();

/*----- handle -----*/
glBegin(GL_LINES);
glColor3f(0.4,0.4,0.4);
glLineWidth(4);
glVertex2i(600, 700);
glVertex2i(550, 650);
glVertex2i(550, 650);
glVertex2i(650, 550);
glVertex2i(650, 550);
glVertex2i(700, 600);
glEnd();

/*----- tires -----*/
glColor3f(0.2,0.2,0.2);
glLineWidth(5);
DrawCircle(200, 300, 120, 100);

DrawCircle(600, 300, 120, 100);

glFlush();
}

int main(int argc, char *argv[]) {
    glutInit(&argc, argv);
    glutInitWindowPosition(100, 100);
    glutInitWindowSize(800, 800);
    glutInitDisplayMode(GLUT_RGB);

```

```
glutCreateWindow("Assignment 1");  
gluOrtho2D(0, 800, 0, 800);  
glClearColor(1,1,1,0.0);  
glutDisplayFunc(Draw);  
glutMainLoop();  
return 0;  
}
```

Output

