NUML
National University of Modern Languages

Team Members:

Nimra Ali(2828)

Ashna Wasif(2791)

Nimra Asif(2815)

Sheeza Ijaz(2792)

Class: BSCS 7B (Morning)

Subject: Computer Graphics

Date: 31st March, 2025

Group Assignment

Task: Transformation in Park Simulation

Before Transformation:

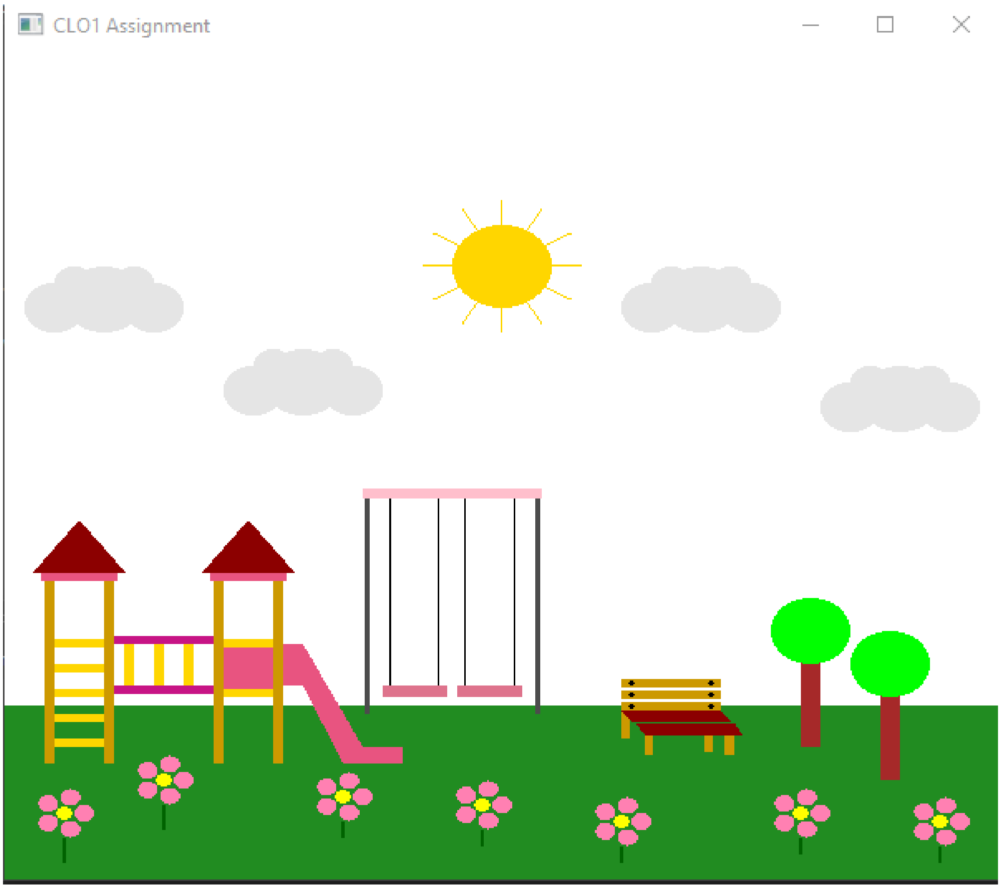
Code:

```
#include <GL/glut.h>
#include <cmath>
#include <windows.h>
#define PI 3.1415926

void drawCircle(float cx, float cy, float radius, float red, float green, float blue)
{
    glBegin(GL_POLYGON);
    glColor3f(red, green, blue);
    for (float i = 0; i < (2 * PI); i += 0.001) {
        float x = cx + radius * cos(i);
        float y = cy + radius * sin(i);
        glVertex2f(x, y);
    }
    glEnd();
}

void drawCloud(float x, float y)
{
    drawCircle(x, y, 20, 0.9f, 0.9f, 0.9f);
    drawCircle(x - 25, y - 5, 15, 0.9f, 0.9f, 0.9f);
    drawCircle(x + 25, y - 5, 15, 0.9f, 0.9f, 0.9f);
    drawCircle(x - 15, y + 10, 10, 0.9f, 0.9f, 0.9f);
    drawCircle(x + 15, y + 10, 10, 0.9f, 0.9f, 0.9f);
}

void drawFlower(float cx, float cy)
{
    // petals
    for (int i = 0; i < 5; i++) {
        float angle = i * (2 * PI / 5);
        float px = cx + 10 * cos(angle);
        float py = cy + 10 * sin(angle);
        drawCircle(px, py, 5, 1.0f, 0.5f, 0.7f);
    }

     // center
    drawCircle(cx, cy, 4, 1.0f, 1.0f, 0.0f);
}

void drawTriangle(float x1, float y1, float x2, float y2, float x3, float y3)
{
    glBegin(GL_TRIANGLES);
    glVertex2f(x1, y1);
    glVertex2f(x2, y2);
    glVertex2f(x3, y3);
    glEnd();
}

void drawRectangle(float x1, float y1, float x2, float y2) {
    glBegin(GL_POLYGON);
    glVertex2f(x1, y1);
    glVertex2f(x2, y1);
    glVertex2f(x2, y2);
    glVertex2f(x1, y2);
    glEnd();
}


void myDisplay() {

    glClear(GL_COLOR_BUFFER_BIT);
    //grass
    glColor3f(0.13f, 0.55f, 0.13f);
     drawRectangle(0, 0,500, 105);

    // tree1
    glColor3f(0.65f, 0.16f, 0.16f);
    drawRectangle(440, 60, 450, 120);
    drawCircle(445, 130, 20, 0.0f, 1.0f, 0.0f);
    // tree2
    glColor3f(0.65f, 0.16f, 0.16f);
    drawRectangle(400, 80, 410, 140);
    drawCircle(405, 150, 20, 0.0f, 1.0f, 0.0f);

    //bench
    glColor3f(0.8f, 0.6f, 0.0f);
     drawRectangle(360, 102, 310, 107);
     drawRectangle(360, 109, 310, 114);
     drawRectangle(360, 116, 310, 121);

    drawCircle(315, 104.5, 1.5, 0.0f, 0.0f, 0.0f);
    drawCircle(355, 104.5, 1.5, 0.0f, 0.0f, 0.0f);

    drawCircle(315, 111.5, 1.5, 0.0f, 0.0f, 0.0f);
    drawCircle(355, 111.5, 1.5, 0.0f, 0.0f, 0.0f);


    drawCircle(315, 118.5, 1.5, 0.0f, 0.0f, 0.0f);
    drawCircle(355, 118.5, 1.5, 0.0f, 0.0f, 0.0f);

    glColor3f(0.8f, 0.6f, 0.0f);
    drawRectangle(310, 102, 314, 85);

    glBegin(GL_POLYGON);
    glColor3f(0.55f, 0.0f, 0.0f);
        glVertex2f(310, 102);
        glVertex2f(360, 102);
        glVertex2f(365, 95);
        glVertex2f(315, 95);
        glEnd();

        glColor3f(0.8f, 0.6f, 0.0f);
        drawRectangle(322, 94, 326, 75);
        drawRectangle(367, 94, 362, 75);
        drawRectangle(356, 94, 352, 77);

        glBegin(GL_POLYGON);
        glColor3f(0.55f, 0.0f, 0.0f);
        glVertex2f(317, 94);
        glVertex2f(367, 94);
        glVertex2f(371,87);
        glVertex2f(322, 87);
        glEnd();


     // clouds
    drawCloud(350, 350);
     drawCloud(450, 290);
     drawCloud(150, 300);
     drawCloud(50, 350);

    //sun
     drawCircle(250, 370, 25, 1.0f, 0.84f, 0.0f);
    //sun rays
    glBegin(GL_LINES);
    for (int i = 0; i < 12; i++)
    {
        float angle = i * (PI / 6);
        float xStart = 250 + 25 * cos(angle);
        float yStart = 370 + 25 * sin(angle);
        float xEnd = 250 + (25 + 15) * cos(angle);
        float yEnd = 370 + (25 + 15) * sin(angle);

        glVertex2f(xStart, yStart);
        glVertex2f(xEnd, yEnd);
    }
    glEnd();

    //slide
    glColor3f(0.8f, 0.6f, 0.0f);
    drawRectangle(20, 70, 25, 180);
    drawRectangle(50, 70, 55, 180);
     drawRectangle(105, 70, 110, 180);
     drawRectangle(135, 70, 140, 180);

    glColor3f(1.0f, 0.84f, 0.0f);
    drawRectangle(25, 80, 50, 85);
     drawRectangle(25, 95, 50, 100);
     drawRectangle(25, 110, 50, 115);
     drawRectangle(25, 125, 50, 130);
     drawRectangle(25, 140, 50, 145);

     drawRectangle(110, 140, 135, 145);
     drawRectangle(110, 110, 135, 115);

     drawRectangle(60, 117, 65, 142);
     drawRectangle(75, 117, 80, 142);
     drawRectangle(90, 117, 95, 142);

    glColor3f(0.78f, 0.08f, 0.52f);
     drawRectangle(55, 142, 105, 147);
     drawRectangle(55, 112, 105, 117);

    glColor3f(0.91f, 0.33f, 0.50f);
    drawRectangle(110, 115, 135, 140);
    drawRectangle(18, 180, 57, 185);
     drawRectangle(103, 180, 142, 185);
     drawRectangle(140, 117, 150, 142);

    glBegin(GL_POLYGON);
    glVertex2f(150, 142);
    glVertex2f(150, 117);
```

```
glVertex2f(170, 70);
glVertex2f(180, 80);
glEnd();

drawRectangle(170, 70, 200, 80);

glColor3f(0.55f, 0.0f, 0.0f);
drawTriangle(14,185,37.5,217,61,185);
 drawTriangle(99, 185, 122.5, 217, 146, 185);

//swing
glColor3f(1.0f, 0.75f, 0.8f);
 drawRectangle(270, 230, 180, 236);

 glColor3f(0.87f, 0.45f, 0.55f);
 drawRectangle(260, 110, 227.5, 117);
 drawRectangle(190, 110, 222.5, 117);

glBegin(GL_LINES);
glColor3f(0.0f, 0.0f, 0.0f);
glVertex2f(256, 230);
glVertex2f(256, 117);

glVertex2f(231, 230);
glVertex2f(231, 117);

glVertex2f(194, 230);
glVertex2f(194, 117);

glVertex2f(218, 230);
glVertex2f(218, 117);

glEnd();

glLineWidth(3.0f);
glBegin(GL_LINES);
glColor3f(0.3f, 0.3f, 0.3f);
glVertex2f(268, 230);
glVertex2f(268, 100);

glVertex2f(182, 230);
glVertex2f(182, 100);
glEnd();

//flowers
drawFlower(30, 40);
drawFlower(80, 60);
 drawFlower(170, 50);
 drawFlower(240, 45);
 drawFlower(310, 35);
 drawFlower(400, 40);
 drawFlower(470, 35);

glLineWidth(2.0f);
glBegin(GL_LINES);
glColor3f(0.0f, 0.39f, 0.0f);
glVertex2f(30, 25);
glVertex2f(30, 10);

glVertex2f(80, 45);
```

```
    glVertex2f(80, 30);

    glVertex2f(170, 35);
    glVertex2f(170, 25);

     glVertex2f(240, 30);
     glVertex2f(240, 20);

     glVertex2f(310, 20);
     glVertex2f(310, 10);

     glVertex2f(400, 25);
     glVertex2f(400, 15);

     glVertex2f(470, 20);
     glVertex2f(470, 10);


    glEnd();
    glFlush();

}

void MyInit()
   {
        glClearColor(1.0, 1.0, 1.0, 1.0);
        glColor3f(0.0, 0.0, 0.0);
        glPointSize(5.0);
        glMatrixMode(GL_PROJECTION);
        glLoadIdentity();
           // coordinates should be set according to this ftn
        gluOrtho2D(0.0, 500.0, 0.0, 500.0);
    }


    int main(int argc, char* argv[])
    {
        glutInit(&argc, argv);
        glutInitDisplayMode(GLUT_SINGLE | GLUT_RGB);
        glutInitWindowSize(600, 500);
        glutInitWindowPosition(100, 100);
        glutCreateWindow("CLO1 Assignment");

        MyInit();
        glutDisplayFunc(myDisplay);

        glutMainLoop();
        return 0;
    }
```

Output:



After Transfromation:

Code:

```
#include "windows.h"
#include "gl/Gl.h"
#include "gl/glut.h"
#include "cmath"

#define PI 3.1415926


void benchSeats(float x1, float x2,float x3,float x4, float y1, float y2)
{
    glColor3f(0.55f, 0.0f, 0.0f);
    glBegin(GL_POLYGON);

    glVertex2f(x1, y1); //x1,y1
    glVertex2f(x2, y1);  //x2,y1
    glVertex2f(x3, y2);  // x3,y2
    glVertex2f(x4, y2);  // x4,y2
    glEnd();
}


void  swingPoles(float a, float b, float c)
{
    glColor3f(0.3f, 0.3f, 0.3f);
    glLineWidth(3.0f);

    glBegin(GL_LINES);

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

    glEnd();
}


void drawStem(float a, float b, float c)
{
    glColor3f(0.4f, 0.2f, 0.0f); // Dark Brown Color

    glLineWidth(2.0f);
    glBegin(GL_LINES);

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

    glEnd();
}


void  swingRopes(float a, float b, float c)
{
    glColor3f(0.0f, 0.0f, 0.0f);
    glBegin(GL_LINES);

    glVertex2f(a, b);
    glVertex2f(a, c);
    glEnd();
}

void drawCircle(float cx, float cy, float radius, float red, float green, float blue)
 {
    glBegin(GL_POLYGON);
    glColor3f(red, green, blue);
    for (float i = 0; i < (2 * PI); i += 0.001) {
        float x = cx + radius * cos(i);
        float y = cy + radius * sin(i);
        glVertex2f(x, y);
    }
    glEnd();
}

void drawCloud(float x, float y)
{
    drawCircle(x, y, 20, 0.9f, 0.9f, 0.9f);
    drawCircle(x - 25, y - 5, 15, 0.9f, 0.9f, 0.9f);
    drawCircle(x + 25, y - 5, 15, 0.9f, 0.9f, 0.9f);
    drawCircle(x - 15, y + 10, 10, 0.9f, 0.9f, 0.9f);
    drawCircle(x + 15, y + 10, 10, 0.9f, 0.9f, 0.9f);
}

void drawFlower(float cx, float cy)
{
    // petals
    for (int i = 0; i < 5; i++) {
        float angle = i * (2 * PI / 5);
        float px = cx + 10 * cos(angle);
        float py = cy + 10 * sin(angle);
        drawCircle(px, py, 5, 1.0f, 0.5f, 0.7f);
    }

    // center
    drawCircle(cx, cy, 4, 1.0f, 1.0f, 0.0f);
}


void drawTriangle(float x1, float y1, float x2, float y2, float x3, float y3)
{
    glBegin(GL_TRIANGLES);
    glVertex2f(x1, y1);
    glVertex2f(x2, y2);
    glVertex2f(x3, y3);
    glEnd();
}

void drawRectangle(float x1, float y1, float x2, float y2) {
    glBegin(GL_POLYGON);
    glVertex2f(x1, y1);
    glVertex2f(x2, y1);
    glVertex2f(x2, y2);
    glVertex2f(x1, y2);
    glEnd();
}

void myDisplay()
{

    glClear(GL_COLOR_BUFFER_BIT);
    //grass
    glColor3f(0.13f, 0.55f, 0.13f);
    drawRectangle(0, 0, 500, 105);


    // tree1
    glColor3f(0.65f, 0.16f, 0.16f);
    drawRectangle(400, 80, 410, 140);
    drawCircle(405, 150, 20, 0.0f, 1.0f, 0.0f);

    // tree2
    glPushMatrix();
    glColor3f(0.65f, 0.16f, 0.16f);
    glTranslatef(40.0, -20.0, 0.0);
    drawRectangle(400, 80, 410, 140); //drawRectangle(440, 60, 450, 120);
    drawCircle(405, 150, 20, 0.0f, 1.0f, 0.0f); //drawCircle(445,130,20,0.0f,1.0f,0.0f);
    glPopMatrix();

    //bench

    /*bench backs*/

    glColor3f(0.8f, 0.6f, 0.0f);
    drawRectangle(360, 102, 310, 107);  //back 1

    //back 2
```

```
glPushMatrix();
glTranslatef(0.0, 7.0, 0.0);
drawRectangle(360, 102, 310, 107); //drawRectangle(360, 109, 310, 114);
glPopMatrix();

//back 3
glPushMatrix();
glTranslatef(0.0, 14.0, 0.0);
drawRectangle(360, 102, 310, 107); //  drawRectangle(360, 116, 310, 121);
glPopMatrix();


/* circles on backs */

//back 1
drawCircle(315, 104.5, 1.5, 0.0f, 0.0f, 0.0f);

glPushMatrix();
glTranslatef(40.0, 0.0, 0.0);
drawCircle(315,104.5, 1.5, 0.0f, 0.0f, 0.0f); //drawCircle(355,104.5,1.5,0.0f,0.0f,0.0f);
glPopMatrix();

//back 2

glPushMatrix();
glTranslatef(0.0, 7.0, 0.0);
drawCircle(315,104.5,1.5, 0.0f,0.0f, 0.0f);   //drawCircle(315,111.5,1.5,0.0f,0.0f,0.0f);
glPopMatrix();

glPushMatrix();
glTranslatef(40.0, 7.0, 0.0);
drawCircle(315,104.5,1.5, 0.0f, 0.0f, 0.0f);  //drawCircle(355,111.5,1.5,0.0f,0.0f,0.0f);
glPopMatrix();

//back 3

glPushMatrix();
glTranslatef(0.0, 14.0, 0.0);
drawCircle(315,104.5,1.5, 0.0f, 0.0f, 0.0f); //drawCircle(315,118.5,1.5,0.0f,0.0f,0.0f);
glPopMatrix();

glPushMatrix();
glTranslatef(40.0, 14.0, 0.0);
drawCircle(315,104.5,1.5, 0.0f, 0.0f, 0.0f); // drawCircle(355,118.5,1.5,0.0f,0.0f,0.0f);
glPopMatrix();

/* bench legs */

//leg 1
glColor3f(0.8f, 0.6f, 0.0f);
drawRectangle(310, 102, 314, 85);

//leg 2
glPushMatrix();
glTranslatef(12.0, -8.0, 0.0);
drawRectangle(310, 102, 314, 85); //  drawRectangle(322, 94, 326, 75);
glPopMatrix();

//leg 3
glPushMatrix();
```

```
      glTranslatef(57.0, -8.0, 0.0);
      drawRectangle(310, 102, 314, 85); //  drawRectangle(367, 94, 362, 75);
      glPopMatrix();

      //leg 4
      glPushMatrix();
      glTranslatef(46.0, -8.0, 0.0);
      drawRectangle(310, 102, 314, 85); //  drawRectangle(356, 94, 352, 77);
      glPopMatrix();


      /* bench seats */

      //seat 1
      benchSeats(310, 360, 365, 315, 102, 95);   //x1, x2, x3, x4, y1, y2

      //seat 2
      benchSeats(317, 367, 371, 322, 94, 87);  //x1, x2, x3, x4, y1, y2


      // clouds

      drawCloud(50, 350);

      glPushMatrix();
      glTranslatef(100.0, -50.0, 0.0);
      drawCloud(50, 350);      //drawCloud(150, 300);
      glPopMatrix();

      glPushMatrix();
      glTranslatef(300.0, 0.0, 0.0);
      drawCloud(50, 350);      //  drawCloud(350, 350);
      glPopMatrix();

      glPushMatrix();
      glTranslatef(400.0, -60.0, 0.0);
      drawCloud(50, 350);      // drawCloud(450, 290);
      glPopMatrix();


      //sun
      drawCircle(250, 370, 25, 1.0f, 0.84f, 0.0f);

      //sun rays
      glBegin(GL_LINES);
      for (int i = 0; i < 12; i++)
      {
          float angle = i * (PI / 6);
          float xStart = 250 + 25 * cos(angle);
          float yStart = 370 + 25 * sin(angle);
          float xEnd = 250 + (25 + 15) * cos(angle);
          float yEnd = 370 + (25 + 15) * sin(angle);

          glVertex2f(xStart, yStart);
          glVertex2f(xEnd, yEnd);
      }
      glEnd();


    //slide
```

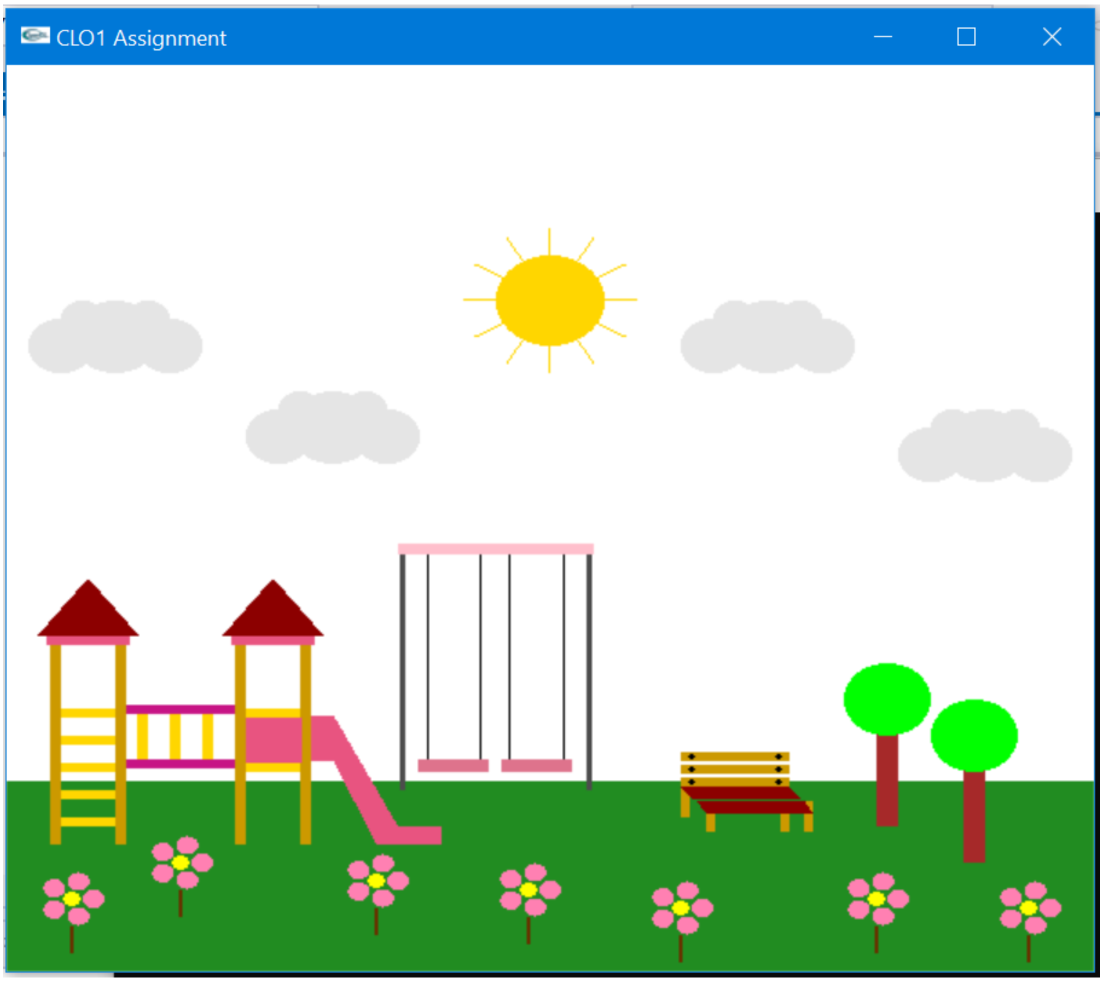
```
// slide poles

glColor3f(0.8f, 0.6f, 0.0f);
drawRectangle(20, 70, 25, 180);

glPushMatrix();
glTranslatef(30.0, 0, 0);
drawRectangle(20, 70, 25, 180); // drawRectangle(50, 70, 55, 180);
glPopMatrix();

glPushMatrix();
glTranslatef(85.0, 0, 0);
drawRectangle(20, 70, 25, 180); // drawRectangle(105, 70, 110, 180);
glPopMatrix();

glPushMatrix();
glTranslatef(115.0, 0, 0);
drawRectangle(20, 70, 25, 180); // drawRectangle(135, 70, 140, 180);
glPopMatrix();

// horizontal lines in left part

glColor3f(1.0f, 0.84f, 0.0f);
drawRectangle(25, 80, 50, 85);

glPushMatrix();
glTranslatef(0, 15, 0);
drawRectangle(25, 80, 50, 85);   //  drawRectangle(25, 95, 50, 100);
glPopMatrix();

glPushMatrix();
glTranslatef(0, 30, 0);
drawRectangle(25, 80, 50, 85);   // drawRectangle(25, 110, 50, 115);
glPopMatrix();

glPushMatrix();
glTranslatef(0, 45, 0);
drawRectangle(25, 80, 50, 85);    //  drawRectangle(25, 125, 50, 130);
glPopMatrix();

glPushMatrix();
glTranslatef(0, 60, 0);
drawRectangle(25, 80, 50, 85);     //  drawRectangle(25, 140, 50, 145);
glPopMatrix();


// horizontal lines in right part

drawRectangle(110, 140, 135, 145);

glPushMatrix();
glTranslatef(0,-30, 0);
drawRectangle(110, 140, 135, 145);    // drawRectangle(110, 110, 135, 115);
glPopMatrix();


//vertical lines in middle

drawRectangle(60, 117, 65, 142);
```

```
  glPushMatrix();
  glTranslatef(15, 0, 0);
  drawRectangle(60, 117, 65, 142);    // drawRectangle(75, 117, 80, 142);
  glPopMatrix();

  glPushMatrix();
  glTranslatef(30, 0, 0);
  drawRectangle(60, 117, 65, 142);    // drawRectangle(90, 117, 95, 142);
  glPopMatrix();


  //horizontal lines in middle

  glColor3f(0.78f, 0.08f, 0.52f);
  drawRectangle(55, 142, 105, 147);

  glPushMatrix();
  glTranslatef(0, -30, 0);
  drawRectangle(55, 142, 105, 147);   // drawRectangle(55, 112, 105, 117);
  glPopMatrix();



  //slide pink boxes

  glColor3f(0.91f, 0.33f, 0.50f); //pink color

  drawRectangle(18, 180, 57, 185);   //pink box in left part


//pink box at top in right part

  glPushMatrix();
  glTranslatef(85.0, 0, 0);
  drawRectangle(18, 180, 57, 185);  // drawRectangle(103, 180, 142, 185);
  glPopMatrix();


  //pink box at center in right part

  glPushMatrix();
  glTranslatef(92.0, -65, 0);
  drawRectangle(18, 180, 44, 205);  // drawRectangle(110, 115, 135, 140);
  glPopMatrix();

  //pink box outside the right part

  glPushMatrix();
  glTranslatef(122.0,-64, 0);
  drawRectangle(18, 180, 28, 205);  // drawRectangle(140, 117, 150, 142);
  glPopMatrix();

  glEnd();


  //slide slope

  glColor3f(0.91f, 0.33f, 0.50f); //pink color
  glBegin(GL_POLYGON);
```

```
 glVertex2f(150, 142);
 glVertex2f(150, 117);
 glVertex2f(170, 70);
 glVertex2f(180, 80);
 glEnd();

 drawRectangle(170, 70, 200, 80);

 //slide triangles

 glColor3f(0.55f, 0.0f, 0.0f);

 //left part
 drawTriangle(14, 185, 37.5, 217, 61, 185);

 //right part
 glPushMatrix();
 glTranslatef(85.0, 0 , 0);
 drawTriangle(14, 185, 37.5, 217, 61, 185);   //drawTriangle(99,185,122.5,217,146,185);
 glPopMatrix();

 //swing

 // Top pole
 glColor3f(1.0f, 0.75f, 0.8f);
 drawRectangle(270, 230, 180, 236);

 //seats
 glColor3f(0.87f, 0.45f, 0.55f);
 drawRectangle(260, 110, 227.5, 117);

 glPushMatrix();
 glTranslatef(-38.0, 0, 0);
 drawRectangle(260, 110, 227.5, 117);  // drawRectangle(190,110,222.5, 117);
 glPopMatrix();

 //swing inner ropes

 /*  glVertex2f(256, 230);
 glVertex2f(256, 117);*/

  swingRopes(256,230,117);

  /*glVertex2f(231, 230);
  glVertex2f(231, 117);*/

  glPushMatrix();
  glTranslatef(-25,0,0);
  swingRopes(256,230,117);
  glPopMatrix();

/*  glVertex2f(194, 230);
 glVertex2f(194, 117);*/

  glPushMatrix();
  glTranslatef(-62, 0, 0);
  swingRopes(256, 230, 117);
  glPopMatrix();
 /* glVertex2f(218, 230);
  glVertex2f(218, 117); */

   glPushMatrix();
   glTranslatef(-38, 0, 0);
   swingRopes(256, 230, 117);
   glPopMatrix();


  glEnd();

  //swing outer poles

  /* glVertex2f(268, 230);
      glVertex2f(268, 100);   */

  swingPoles(268,230,100);

/*  glVertex2f(182, 230);
  glVertex2f(182, 100);*/

  glPushMatrix();
  glTranslatef(-86, 0, 0);
  swingPoles(268, 230, 100);
  glPopMatrix();


    //flowers
    drawFlower(30, 40);

    glPushMatrix();
    glTranslatef(50.0, 20.0, 0.0);
    drawFlower(30, 40);            //drawFlower(80, 60);
    glPopMatrix();

    glPushMatrix();
    glTranslatef(140.0, 10.0, 0.0);
    drawFlower(30, 40);      //drawFlower(170, 50);
    glPopMatrix();

    glPushMatrix();
    glTranslatef(210.0, 5.0, 0.0);
    drawFlower(30, 40);      //  drawFlower(240, 45);
    glPopMatrix();

     glPushMatrix();
    glTranslatef(280.0, -5.0, 0.0);
    drawFlower(30, 40);      //  drawFlower(310, 35);
    glPopMatrix();

     glPushMatrix();
    glTranslatef(370.0, 0.0, 0.0);
    drawFlower(30, 40);      // drawFlower(400, 40);
    glPopMatrix();

    glPushMatrix();
    glTranslatef(440.0, -5.0, 0.0);
    drawFlower(30, 40);      // drawFlower(470, 35);
    glPopMatrix();
    /* Flowers sticks */

     //stem 1

    /*  glVertex2f(30, 25);
    glVertex2f(30, 10);  */

    drawStem(30, 25, 10);

    //stem 2

    /* glVertex2f(80, 45);
       glVertex2f(80, 30);*/

      glPushMatrix();
      glTranslatef(50, 20, 0);    // (30 + 50, 25 + 20) →(80, 45)
                                    //(30 + 50, 10 + 20) →(80, 30)
      drawStem(30, 25, 10);
      glPopMatrix();


    //stem 3

    /* glVertex2f(170, 35);
    glVertex2f(170, 25);  */

    glPushMatrix();

    glTranslatef(140, 10, 0);
    drawStem(30, 25, 10);
    glPopMatrix();

    //stem 4

    /* glVertex2f(240, 30);
    glVertex2f(240, 20); */

    glPushMatrix();

    glTranslatef(210, 5 , 0);
    drawStem(30, 25, 10);
    glPopMatrix();

    //stem 5

    /* glVertex2f(310, 20);
    glVertex2f(310, 10); */

    glPushMatrix();

    glTranslatef(280, -5 , 0);
    drawStem(30, 25, 10);
    glPopMatrix();

    //stem 6

 /*   glVertex2f(400, 25);
    glVertex2f(400, 15);*/
```

```
    glPushMatrix();

    glTranslatef(370, 0 , 0);
    drawStem(30, 25, 10);
    glPopMatrix();

    //stem 7

    /*  glVertex2f(470, 20);
    glVertex2f(470, 10);  */


    glPushMatrix();

    glTranslatef(440, -5 , 0);
    drawStem(30, 25, 10);
    glPopMatrix();


    glEnd();
    glFlush();

}



void MyInit()
{
    glClearColor(1.0, 1.0, 1.0, 1.0);
    glColor3f(0.0, 0.0, 0.0);
    glPointSize(5.0);
    glMatrixMode(GL_PROJECTION);
    glLoadIdentity();
    // coordinates should be set according to these points
    gluOrtho2D(0.0, 500.0, 0.0, 500.0);
}

int main(int argc, char* argv[]) {
    glutInit(&argc, argv);
    glutInitDisplayMode(GLUT_SINGLE | GLUT_RGB);
    glutInitWindowSize(600, 500);
    glutInitWindowPosition(100, 100);
    glutCreateWindow("CLO1 Assignment");

    MyInit();
    glutDisplayFunc(myDisplay);

    glutMainLoop();
    return 0;
}
```

Output:



CLO1 Assignment