Lab 5

Name : Abdelaziz Mohamed Abdelaziz.

I.D : 19015941.

Code Snippets :

In this code we define function setup in which we initialize two display lists one for seat and the other for legs and we used built in cylinder in case of wire frame glutwirecylinder and in case of solid glutsolidcylinder

```
1  ////////////////////////////////////////////////////////////
2  // helixList.cpp
3  //
4  // This program draws several helixes using a display list.
5  //
6  // Sumanta Guha.
7  ////////////////////////////////////////////////////////////
8
9  #define _USE_MATH_DEFINES
10 #include <cstdlib>
11 #include <cmath>
12 #include <iostream>
13 #include <GL/glew.h>
14 #include <GL/freeglut.h>
15 // Globals.
16 static unsigned int aStool; // List index.
17 static float xAngle= 0.0, yAngle = 0.0, zAngle = 0.0; // Angles for rotation.
18 static bool drawMode= true;
19 static unsigned int aStool2;
20 // Initialization routine.
21 void setup(void)
22 {
23     float angle; // Angle parameter.
24     aStool = glGenLists(1); // Return a list index.
25     aStool2 = glGenLists(1);
26
27     // Begin create a display list.
28     // Draw the top of the stool.
29     glNewList(aStool2, GL_COMPILE);
30     float radius = 5.0, height = 1.0;
31     int slices = 10, stacks = 3;
32     glPushMatrix();
33
34     if (drawMode) glutSolidCylinder(radius, height, slices, stacks);
```

```
34      if (drawMode) glutSolidCylinder(radius, height, slices, stacks);
35      else
36      {
37           glPolygonMode(GL_FRONT_AND_BACK, GL_FILL);
38           glutSolidCylinder(radius, height, slices, stacks);
39      }
40      glPopMatrix();
41      glEndList();
42
43
44      // Draw the three legs of the stool.
45      glNewList(aStool, GL_COMPILE);
46      radius = 0.5; height = 9.0;
47      glPushMatrix();
48      if (drawMode) glutWireCylinder(radius, height, slices, stacks);
49      else
50      {
51           glPolygonMode(GL_FRONT_AND_BACK, GL_FILL);
52           glutSolidCylinder(radius, height, slices, stacks);
53      }
54      glPopMatrix();
55      glEndList();
56      // End create a display list.
57      glClearColor(1.0, 1.0, 1.0, 0.0);
```

In the draw scene we setup view transformation for seat and legs using gl rotate and gltransalete

```
61  void drawScene(void)
62  {
63      glClear(GL_COLOR_BUFFER_BIT | GL_DEPTH_BUFFER_BIT);
64      glLoadIdentity();
65
66      // Set up viewing transformation.
67      glTranslatef(0.0, 3.0, -40.0);
68      glRotatef(103.0, 90.0, 1.0, 0.0);
69      glRotatef(xAngle, 1.0, 0.0, 0.0);
70      glRotatef(yAngle, 0.0, 1.0, 0.0);
71      glRotatef(zAngle, 0.0, 0.0, 1.0);
72      glPolygonMode(GL_FRONT_AND_BACK, GL_LINE);
73
74
75      // Draw the stool using the display list.
76      glColor3f(0.0f, 0.0f, 0.0f);
77      glPushMatrix();
78      glTranslatef(0.252, 0.0, 0.0);
79      glCallList(aStool2); // Execute display list.
80      glPopMatrix();
81
82      glColor3f(0.0, 0.0, 0.0);
83      glPushMatrix();
84      glTranslatef(-1.2, -1.5, 1.0);
85      glRotatef(-30.0, 0.0, 1.0, 0.0);
86      glCallList(aStool); // Execute display list.
87      glPopMatrix();
88
89      glColor3f(0.0, 0.0, 0.0);
90      glPushMatrix();
91      glTranslatef(2.0, -1.5, 1.0);
92      glRotatef(30.0, 0.0, 1.0, 0.0);
93      glCallList(aStool); // Execute display list.
94      glPopMatrix();
```

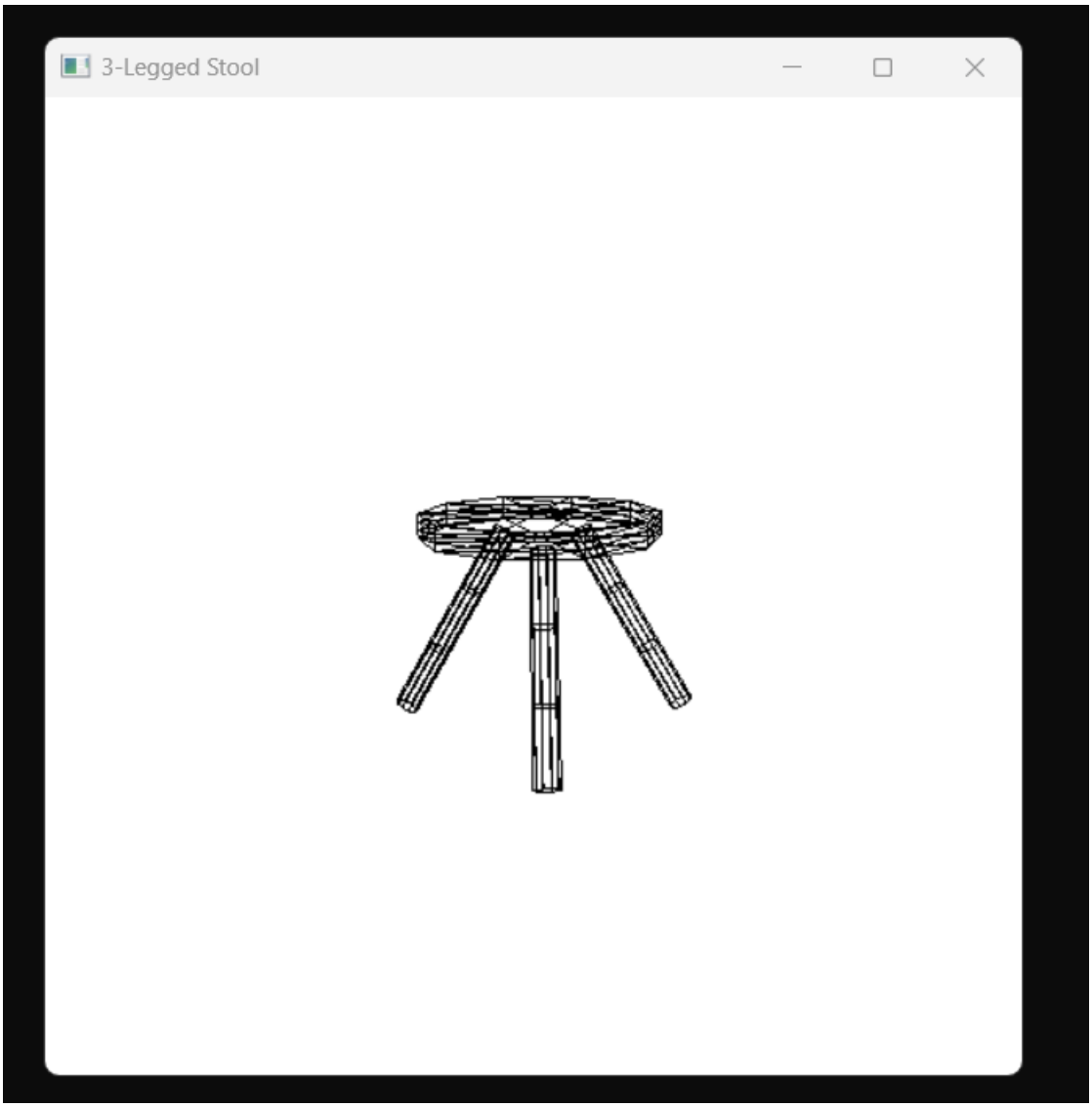
```
    glColor3f(0.0, 0.0, 0.0);
    glPushMatrix();
    glTranslatef(2.0, -1.5, 1.0);
    glRotatef(30.0, 0.0, 1.0, 0.0);
    glCallList(aStool); // Execute display list.
    glPopMatrix();

    glColor3f(0.0, 0.0, 0.0);
    glPushMatrix();
    glTranslatef(0.3, 3.0, 1.0);
    glRotatef(-30.0, 1.0, 0.0, 0.0);
    glCallList(aStool); // Execute display list.
    glPopMatrix();


    glFlush();
```

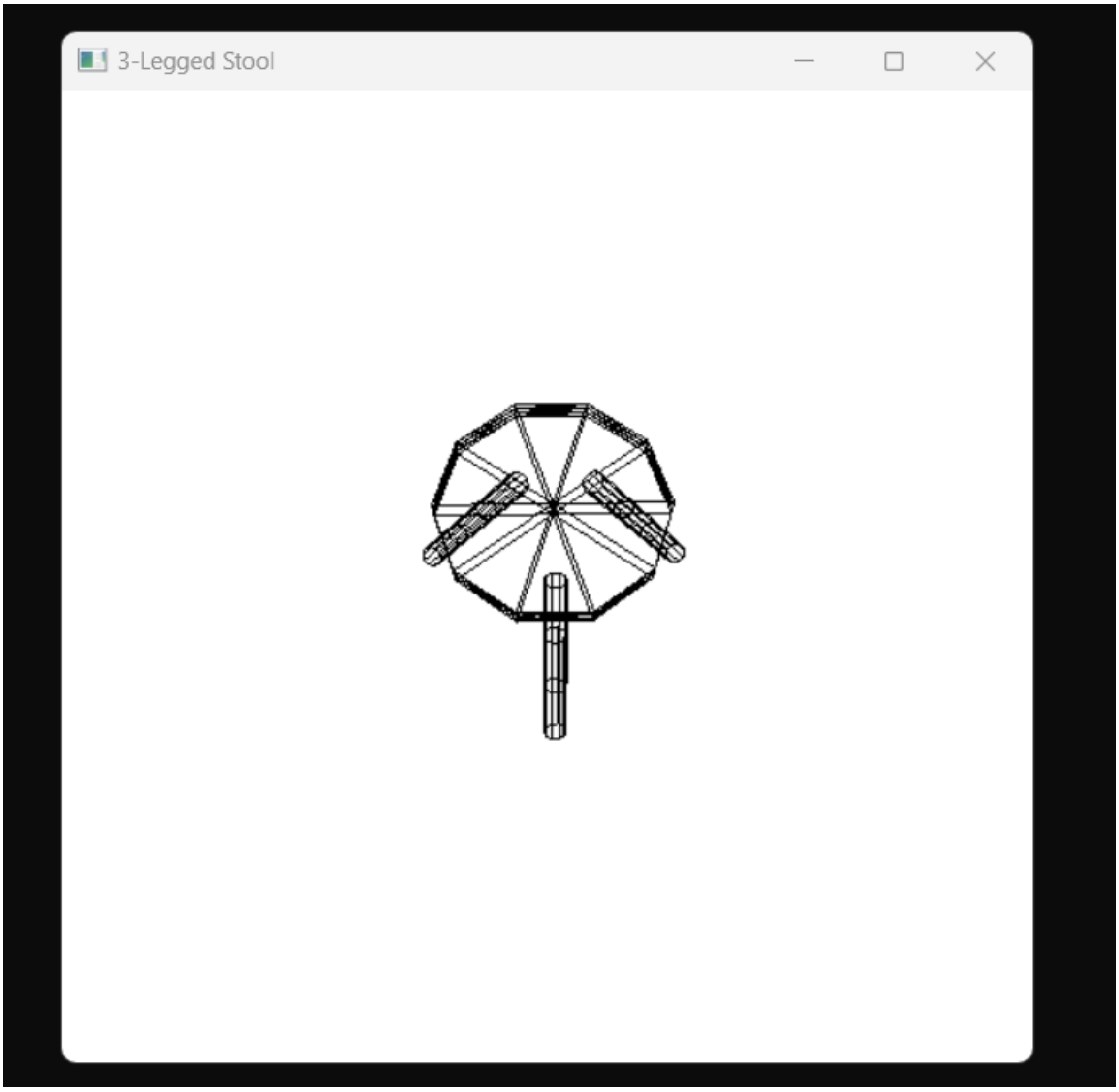
```
void resize(int w, int h)
{
    glViewport(0, 0, w, h);
    glMatrixMode(GL_PROJECTION);
    glLoadIdentity();
    glFrustum(-0.5, 0.5, -0.5, 0.5, 1.0, 100.0);
    glMatrixMode(GL_MODELVIEW);
    glLoadIdentity();
}
```

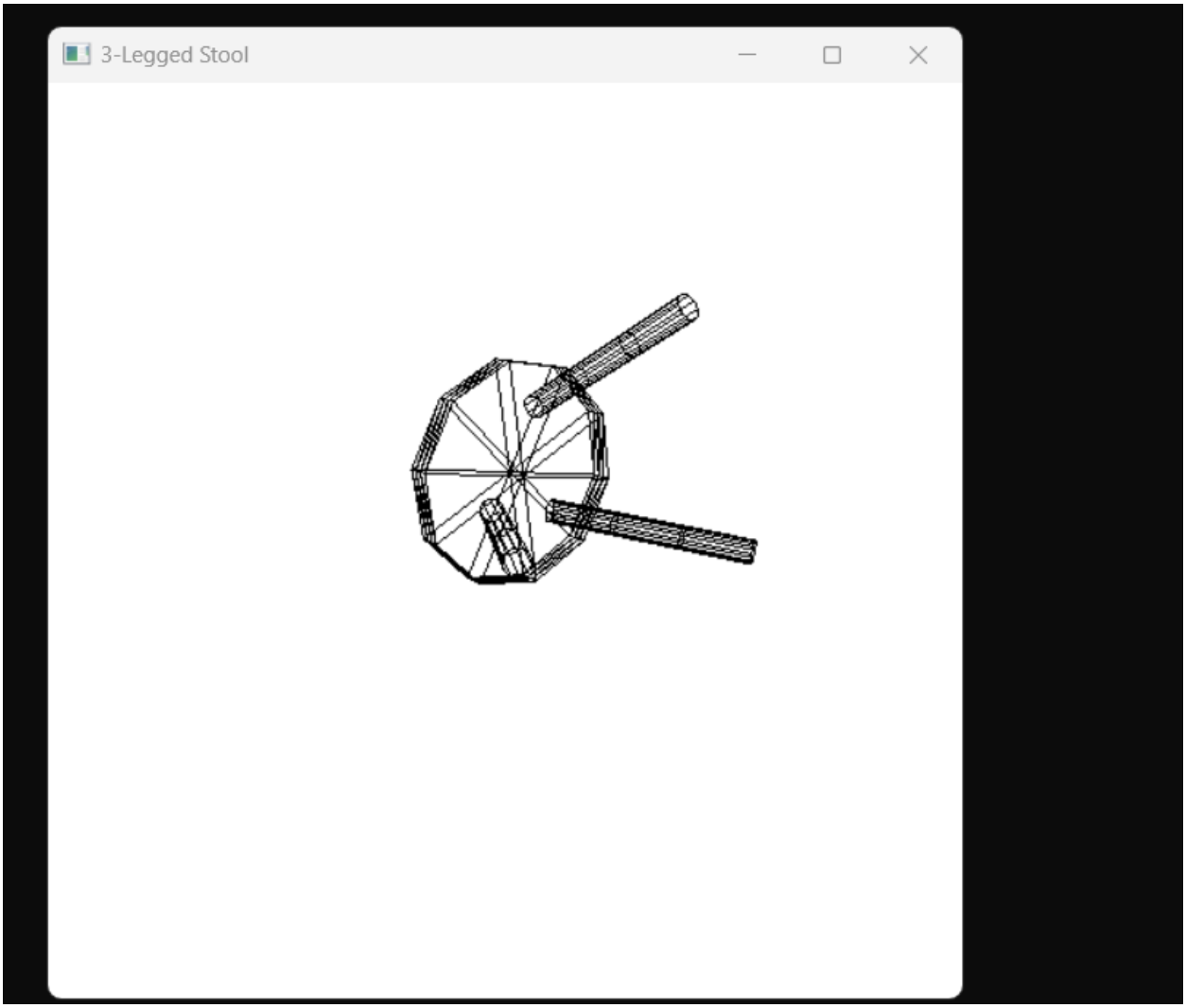
```
116  // Keyboard input processing routine.
117  void keyInput(unsigned char key, int x, int y)
118  {
119      switch (key)
120      {
121      case 'x':
122           xAngle -= 5.0;
123          if (xAngle < -360.0) xAngle += 360.0;
124          glutPostRedisplay();
125          break;
126      case 'X':
127          xAngle += 5.0;
128          if (xAngle > 360.0) xAngle -= 360.0;
129          glutPostRedisplay();
130          break;
131      case 'y':
132           yAngle -= 5.0;
133          if (yAngle < -360.0) yAngle += 360.0;
134          glutPostRedisplay();
135          break;
136      case 'Y':
137          yAngle += 5.0;
138          if (yAngle > 360.0) yAngle -= 360.0;
139          glutPostRedisplay();
140          break;
141      case 'z':
142          zAngle -= 5.0;
143          if (zAngle < -360.0) zAngle += 360.0;
144          glutPostRedisplay();
145          break;
146      case 'Z':
147          zAngle += 5.0;
148          if (zAngle > 360.0) zAngle -= 360.0;
149          glutPostRedisplay();
150          break;
151      case ' ':
152          drawMode = !drawMode;
153          setup();
```

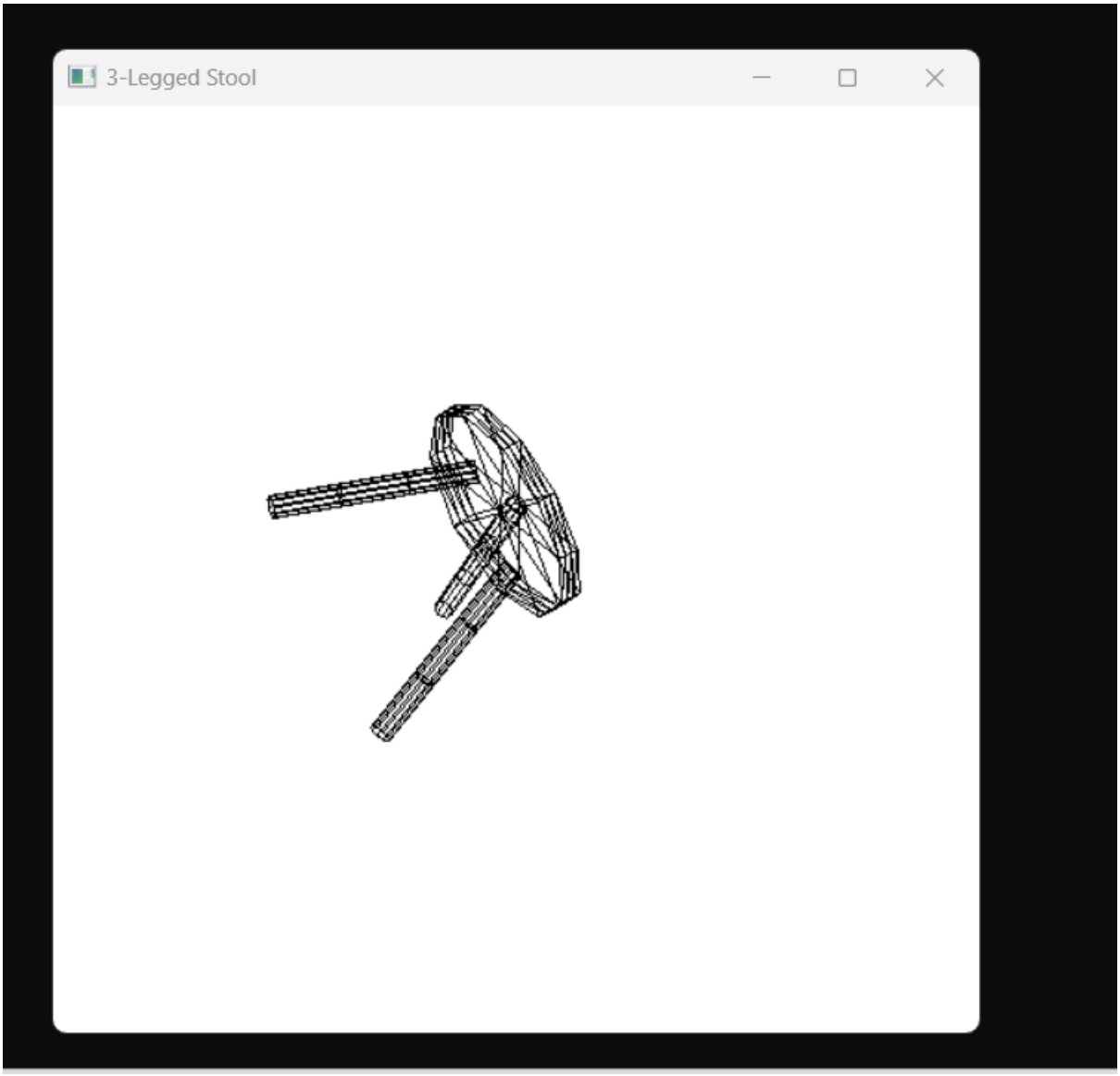
```
51
52   // Main routine.
53   int main(int argc, char **argv)
54   {
55       glutInit(&argc, argv);
56       glutInitContextVersion(4, 3);
57       glutInitContextProfile(GLUT_COMPATIBILITY_PROFILE);
58       glutInitDisplayMode(GLUT_SINGLE | GLUT_RGBA);
59       glutInitWindowSize(500, 500);
60       glutInitWindowPosition(100, 100);
61       glutCreateWindow("3-Legged Stool");
62       glutDisplayFunc(drawScene);
63       glutReshapeFunc(resize);
64       glutKeyboardFunc(keyInput);
65       glewExperimental = GL_TRUE;
66       glewInit();
67       setup();
68       glutMainLoop();
69   }
```

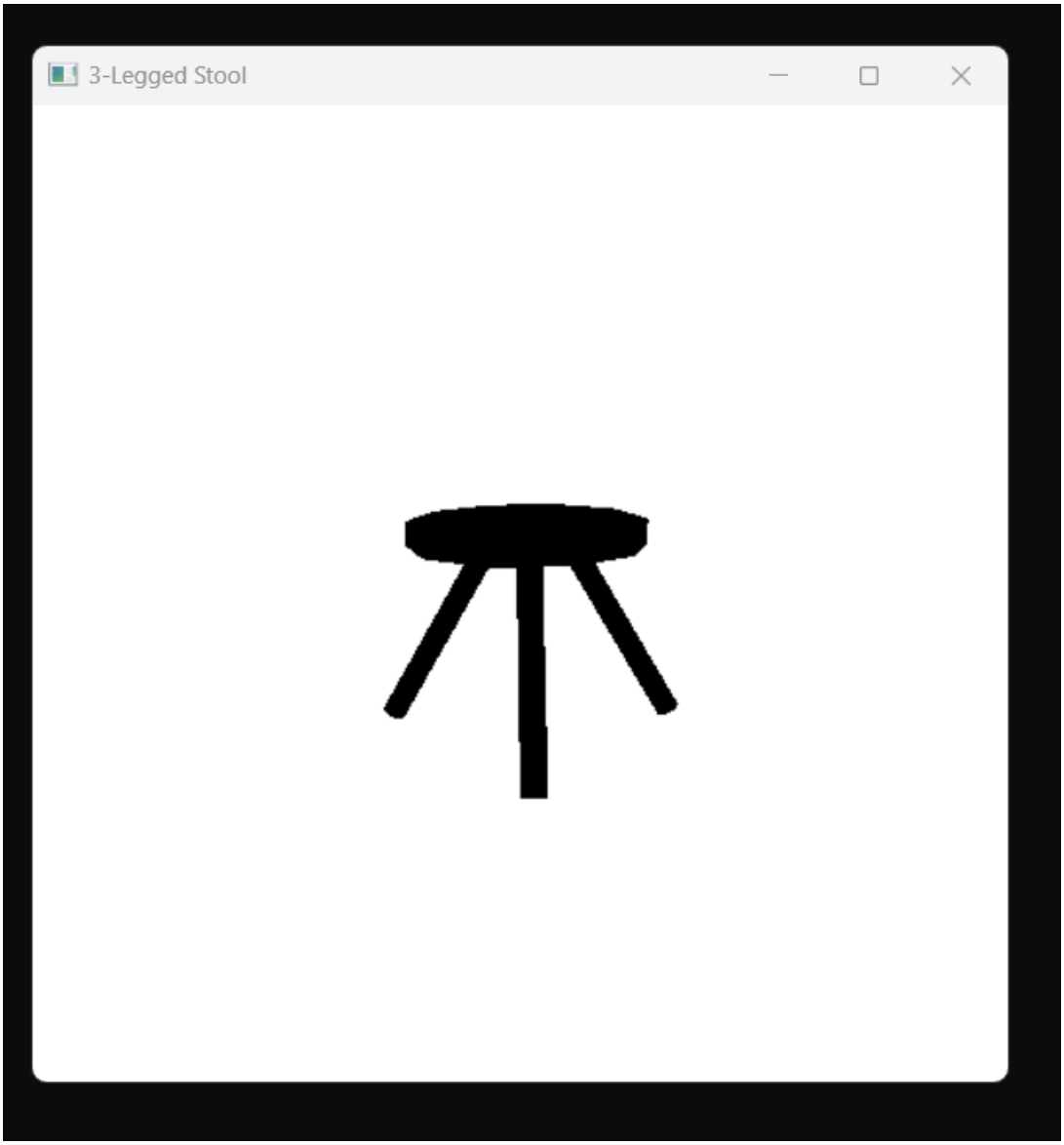
Sample Run :

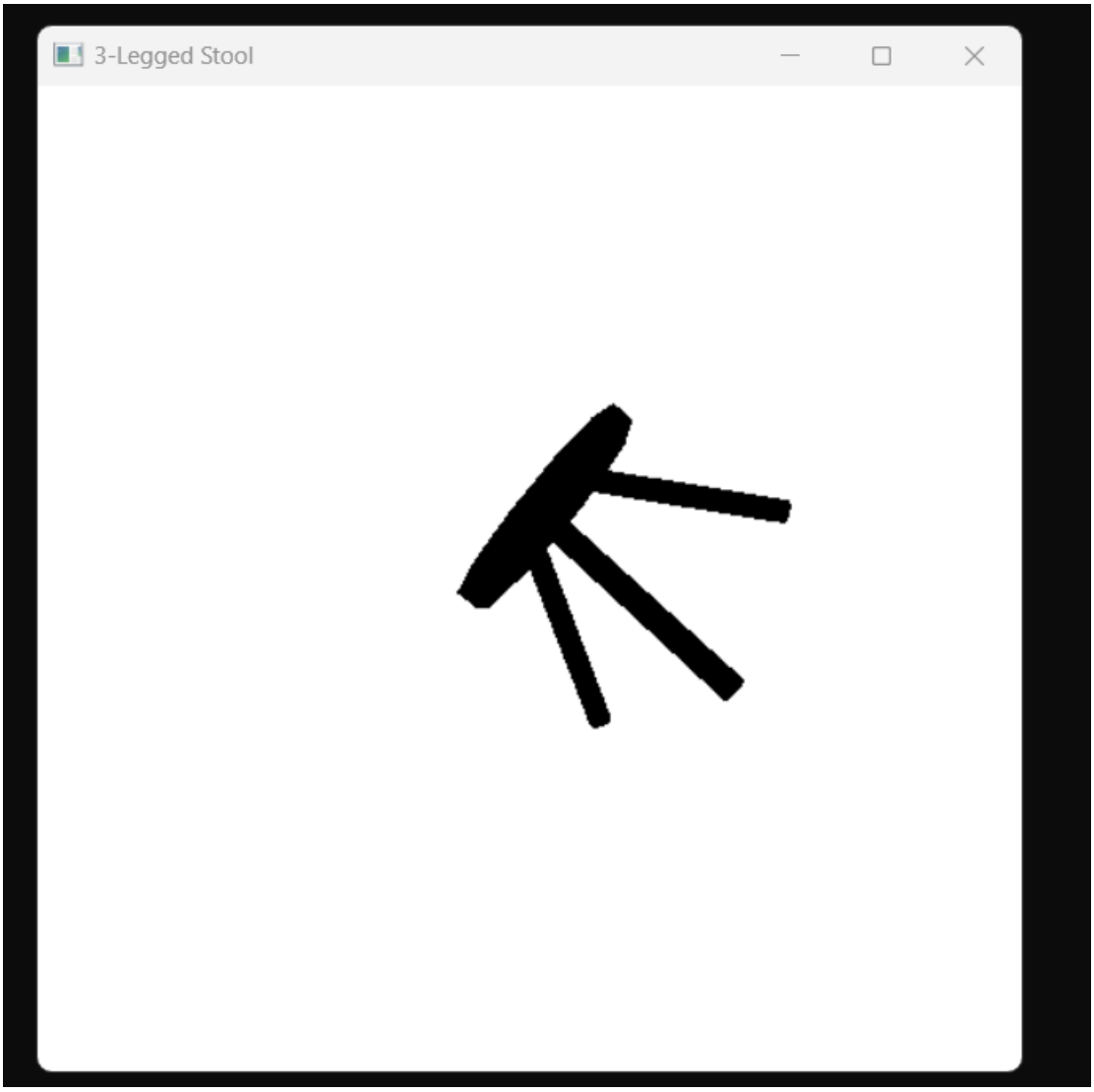


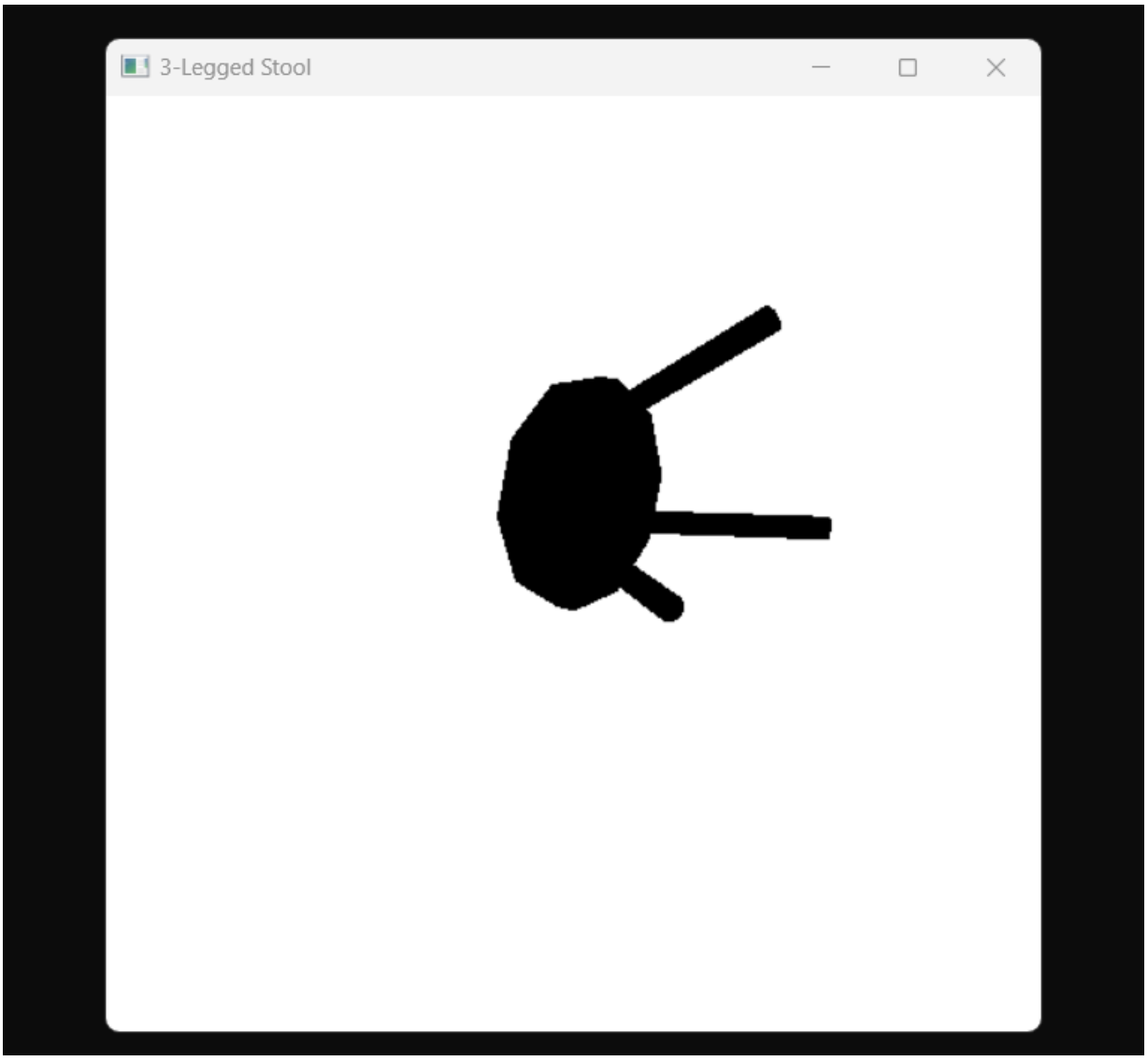

3-Legged Stool


3-Legged Stool


3-Legged Stool


3-Legged Stool


3-Legged Stool


3-Legged Stool