Lab 5

Computer Graphics

Name	ID
Ahmed Adel Abudef	19015264

Table Of Contents:

I. Snippets

```
void init(void) {
    // Create display list for leg
    legList = glGenLists( range: 1);
    glNewList( list: legList,  mode: GL_COMPILE);

// Define the number of sides for the cylinder
    int sides = 20;

// Define the radius and height of the cylinder
    float radius = 0.2f;
    float height = 0.7f;

// Draw the bottom circle
    glBegin( mode: GL_POLYGON);
    for (int i = 0; i < sides; ++i) {
        float angle = i * 2.0f * M_PI / sides;
        glVertex3f( x: radius*cosf( x: angle),  y: 0.0f,  z: radius*sinf( x: angle));
    }
    glEnd();

// Draw the top circle
    glBegin( mode: GL_POLYGON);
    for (int i = 0; i < sides; ++i) {
        float angle = i * 2.0f * M_PI / sides;
        glVertex3f( x: radius*cosf( x: angle),  y: height,  z: radius*sinf( x: angle));
    }
    glEnd();
```

```
// Draw the sides of the cylinder
    for (int i = 0; i < sides; ++i) {
        float angle1 = i * 2.0f * M_PI / sides;
        float angle2 = (i+1) * 2.0f * M_PI / sides;

        glBegin( mode: GL_POLYGON);
        glVertex3f( x: radius*cosf( x: angle1),  y: 0.0f,  z: radius*sinf( x: angle1));
        glVertex3f( x: radius*cosf( x: angle2),  y: 0.0f,  z: radius*sinf( x: angle2));
        glVertex3f( x: radius*cosf( x: angle2),  y: height,  z: radius*sinf( x: angle2));
        glVertex3f( x: radius*cosf( x: angle1),  y: height,  z: radius*sinf( x: angle1));
        glEnd();
    }

    glEndList();
    glClearColor( red: 0.0,  green: 0.0,  blue: 0.0,  alpha: 1.0);
    glEnable( cap: GL_DEPTH_TEST);

}
```

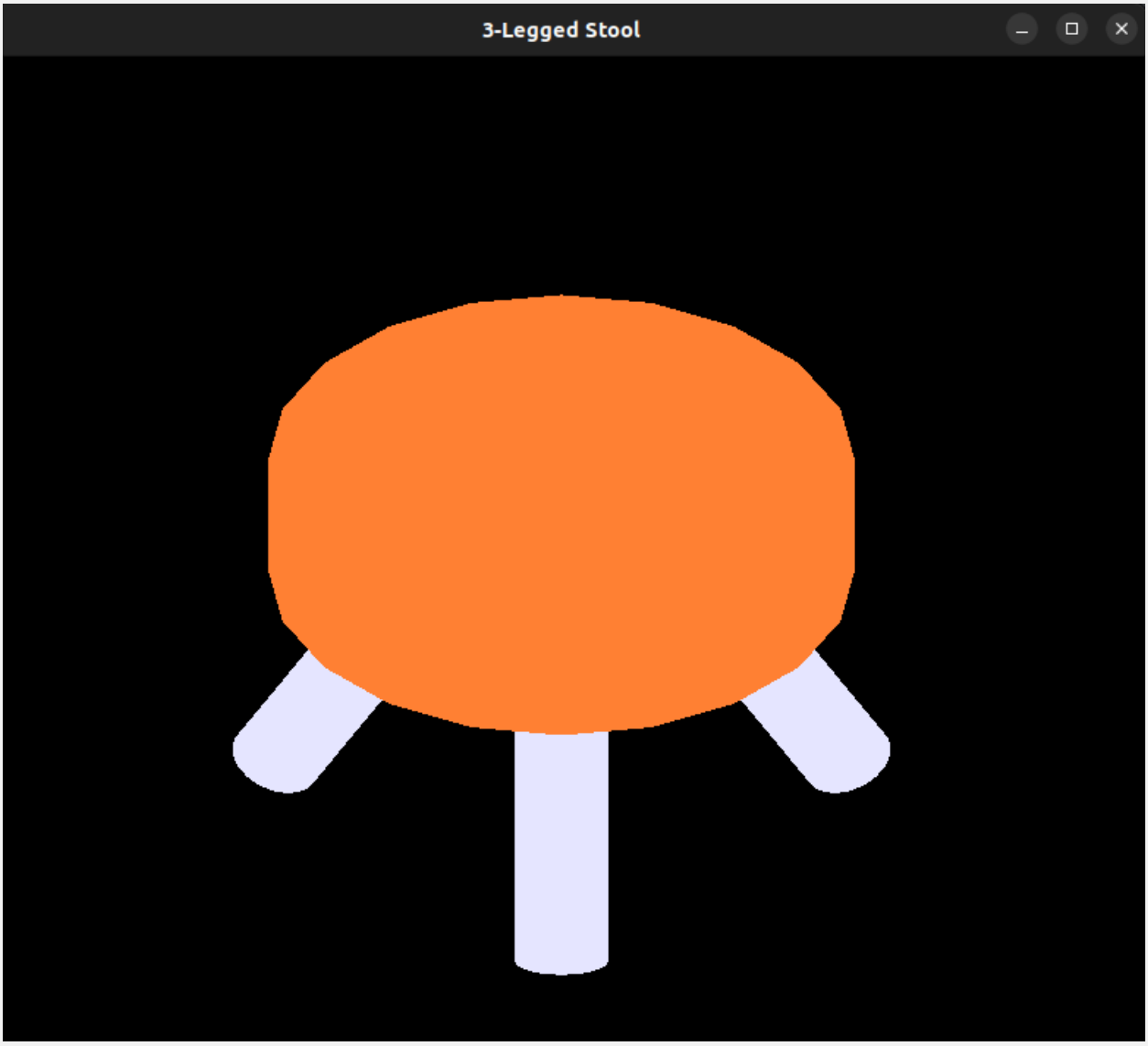
```
void display(void) {
    // Clear color and depth buffers
    glClear( mask: GL_COLOR_BUFFER_BIT | GL_DEPTH_BUFFER_BIT);

    // Apply rotation transformations based on user input
    glRotatef( angle: angleX,  x: 1.0f,  y: 0.0f,  z: 0.0f);
    glRotatef( angle: angleY,  x: 0.0f,  y: 1.0f,  z: 0.0f);
    glRotatef( angle: angleZ,  x: 0.0f,  y: 0.0f,  z: 1.0f);

    glPushMatrix();
    glScaled( x: 2.5,  y: 0.4,  z: 2.5);
    glColor3f( red: 1.0f,  green: 0.5f,  blue: 0.2f);
    glCallList( list: legList); // Execute display list.
    glPopMatrix();
```

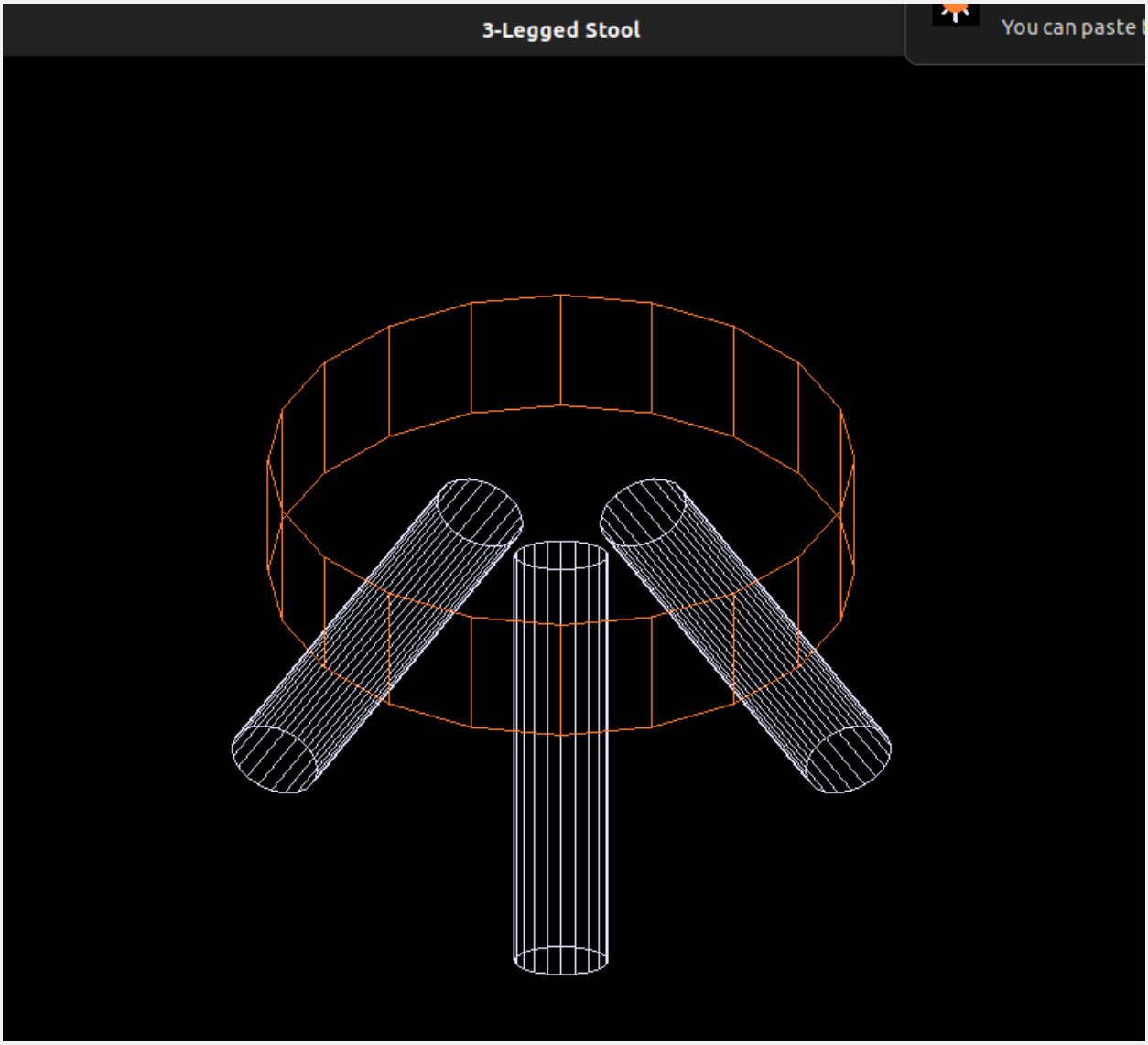
```
//     Draw three legs using display list and appropriate rotations
    for (int i = 0; i < 3; i++) {
        glPushMatrix();
        glColor3f( red: 0.9f,  green: 0.9f,  blue: 1.0f);

        switch (i) {
            case 0:
                glRotatef( angle: 20,  x: 1.0f,  y: 0.0f,  z: 0.0f);

                glScaled( x: 0.4,  y: 1.2,  z: 0.4);
                glTranslated( x: 0.0,  y: -0.65,  z: -0.2);

                glCallList( list: legList);

                break;
            case 1:
                glRotatef( angle: 25,  x: 0.0f,  y: 0.0f,  z: 1.0f);
                glRotatef( angle: -10,  x: 1.0f,  y: 0.0f,  z: 0.0f);

                glScaled( x: 0.4,  y: 1.2,  z: 0.4);
                glTranslated( x: 0.4,  y: -0.7,  z: 0.2);
                glCallList( list: legList);

                break;
            case 2:
                glRotatef( angle: -25,  x: 0.0f,  y: 0.0f,  z: 1.0f);
                glRotatef( angle: -10,  x: 1.0f,  y: 0.0f,  z: 0.0f);

                glScaled( x: 0.4,  y: 1.2,  z: 0.4);
                glTranslated( x: -0.4,  y: -0.7,  z: 0.2);
                glCallList( list: legList);
                break;
        }
        glPopMatrix();
    }
```

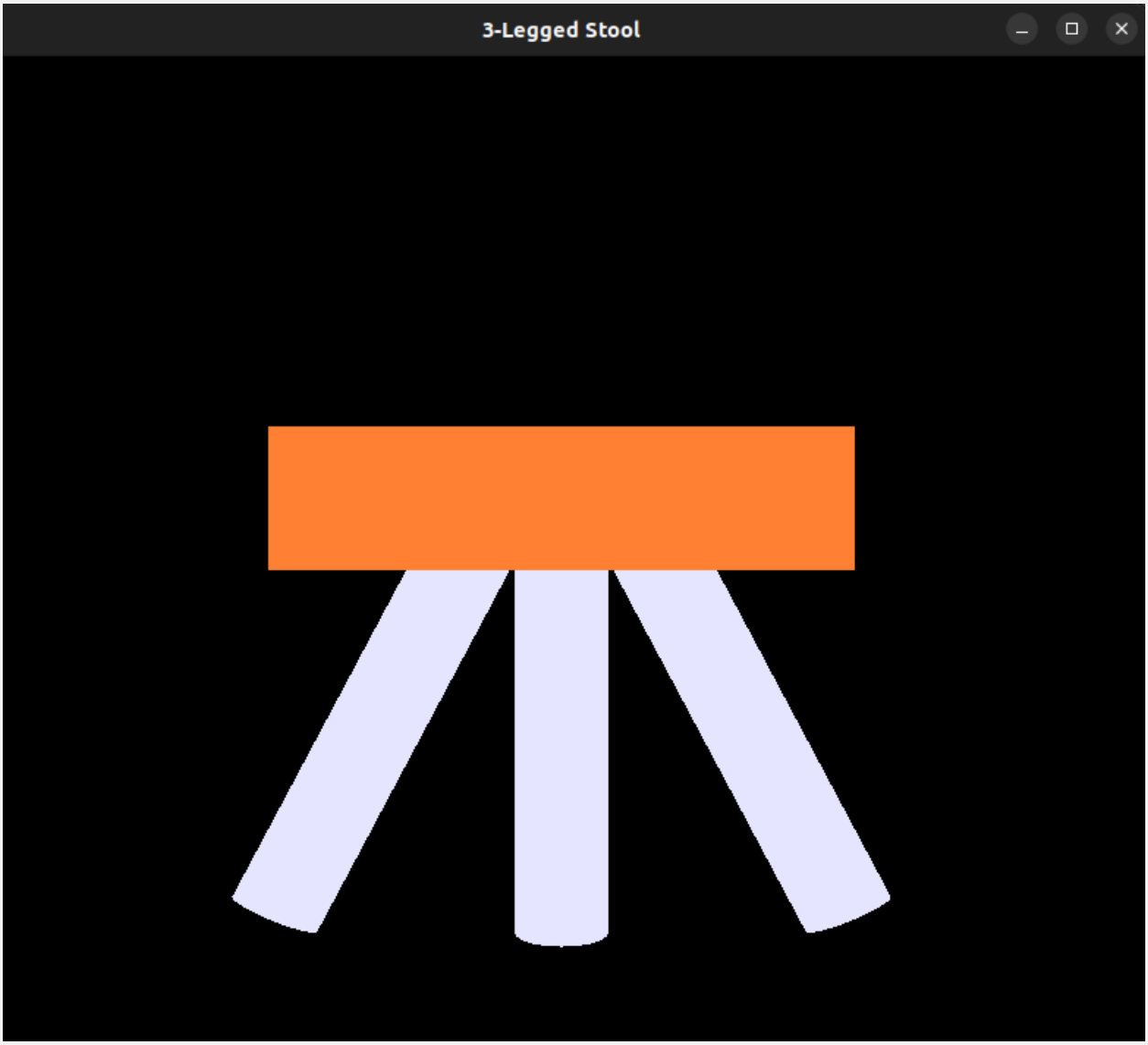
```
void keyboard(unsigned char key, int x, int y) {
    switch (key) {
        case 'x':
        case 'X':
            angleX = 5.0f;
            angleY = 0.0f;
            angleZ = 0.0f;
            break;
        case 'y':
        case 'Y':
            angleY = 5.0f;
            angleX = 0.0f;
            angleZ = 0.0f;
            break;
        case 'z':
        case 'Z':
            angleZ = 5.0f;
            angleX = 0.0f;
            angleY = 0.0f;
            break;
        case ' ':
            isWireframeMode = !isWireframeMode;
            if (isWireframeMode) {
                glPolygonMode( face: GL_FRONT_AND_BACK,  mode: GL_LINE);
            } else {
                glPolygonMode( face: GL_FRONT_AND_BACK,  mode: GL_FILL);
            }
            angleY = 0.0f;
            angleX = 0.0f;
            angleZ = 0.0f;
            break;
    }
```

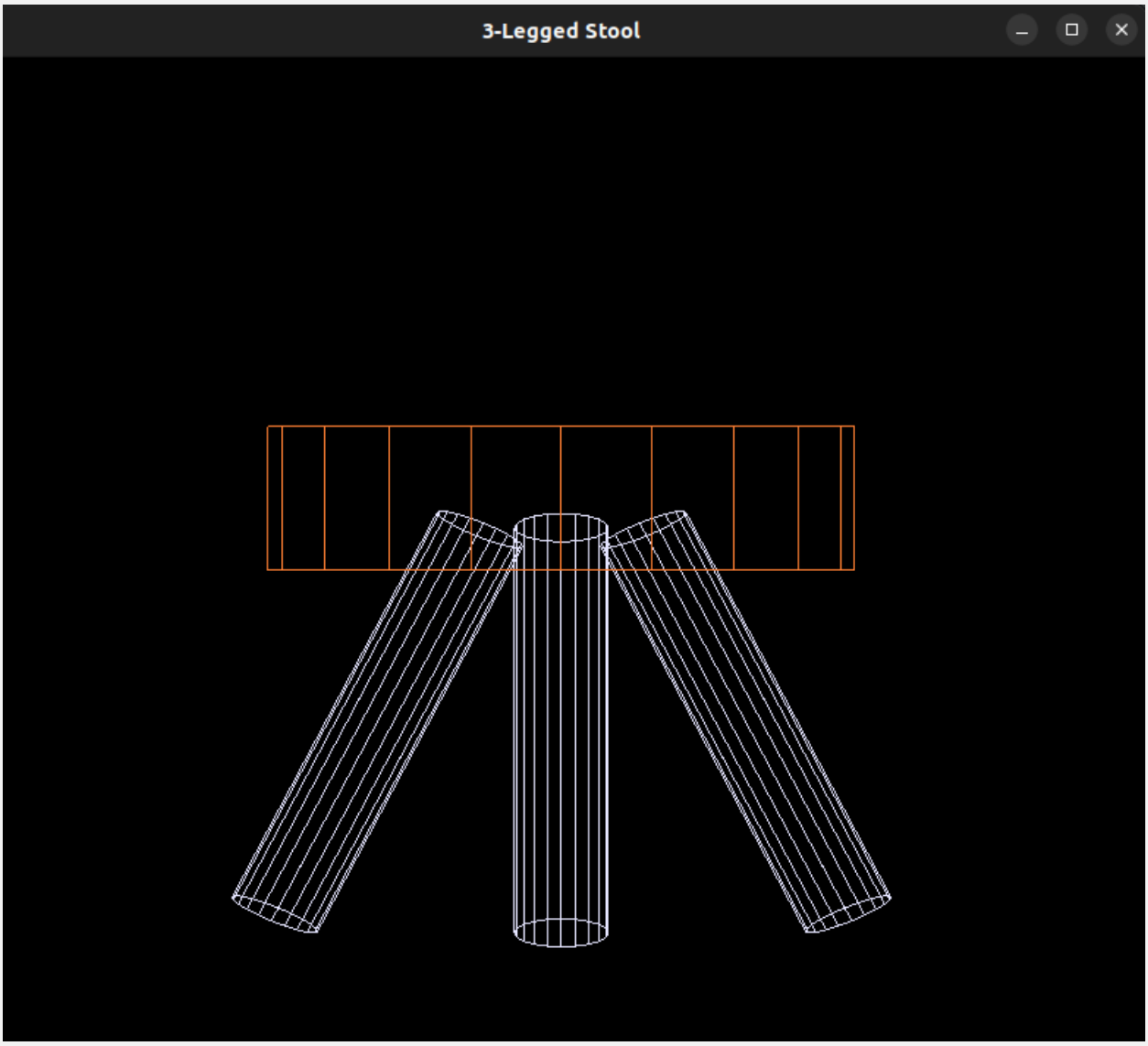
```
int main(int argc, char** argv) {
// Initialize GLUT and create window
    glutInit( pargc: &argc, argv);
    glutInitDisplayMode( displayMode: GLUT_DOUBLE | GLUT_RGB | GLUT_DEPTH);
    glutInitWindowSize( width: 800,  height: 700);
    glutCreateWindow( title: "3-Legged Stool");

// Initialize scene and set up callback functions
    init();
    glutDisplayFunc( callback: display);
    glutKeyboardFunc( callback: keyboard);

// Start main loop
    glutMainLoop();
    return 0;
}
```

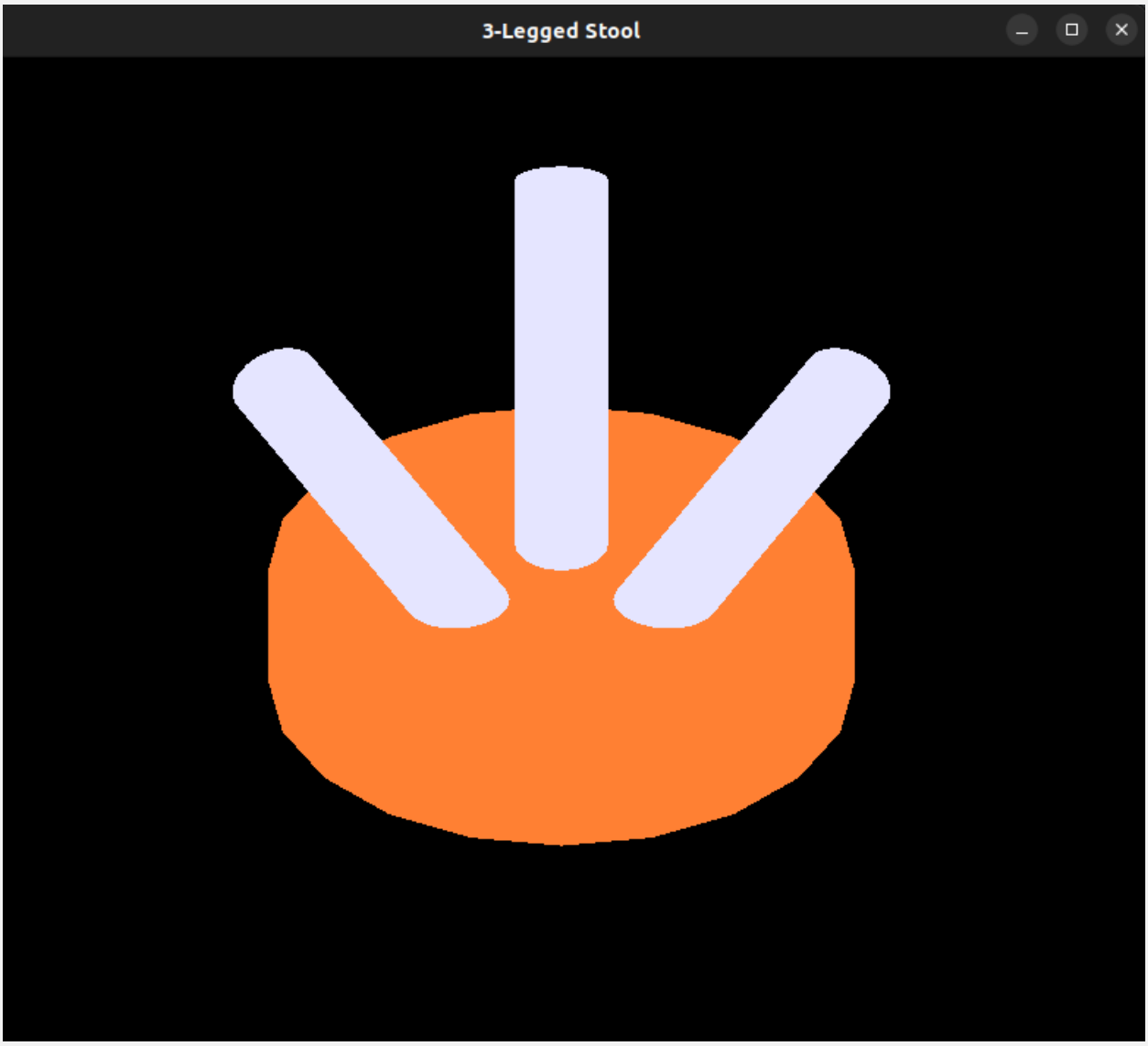
II. Sample Runs

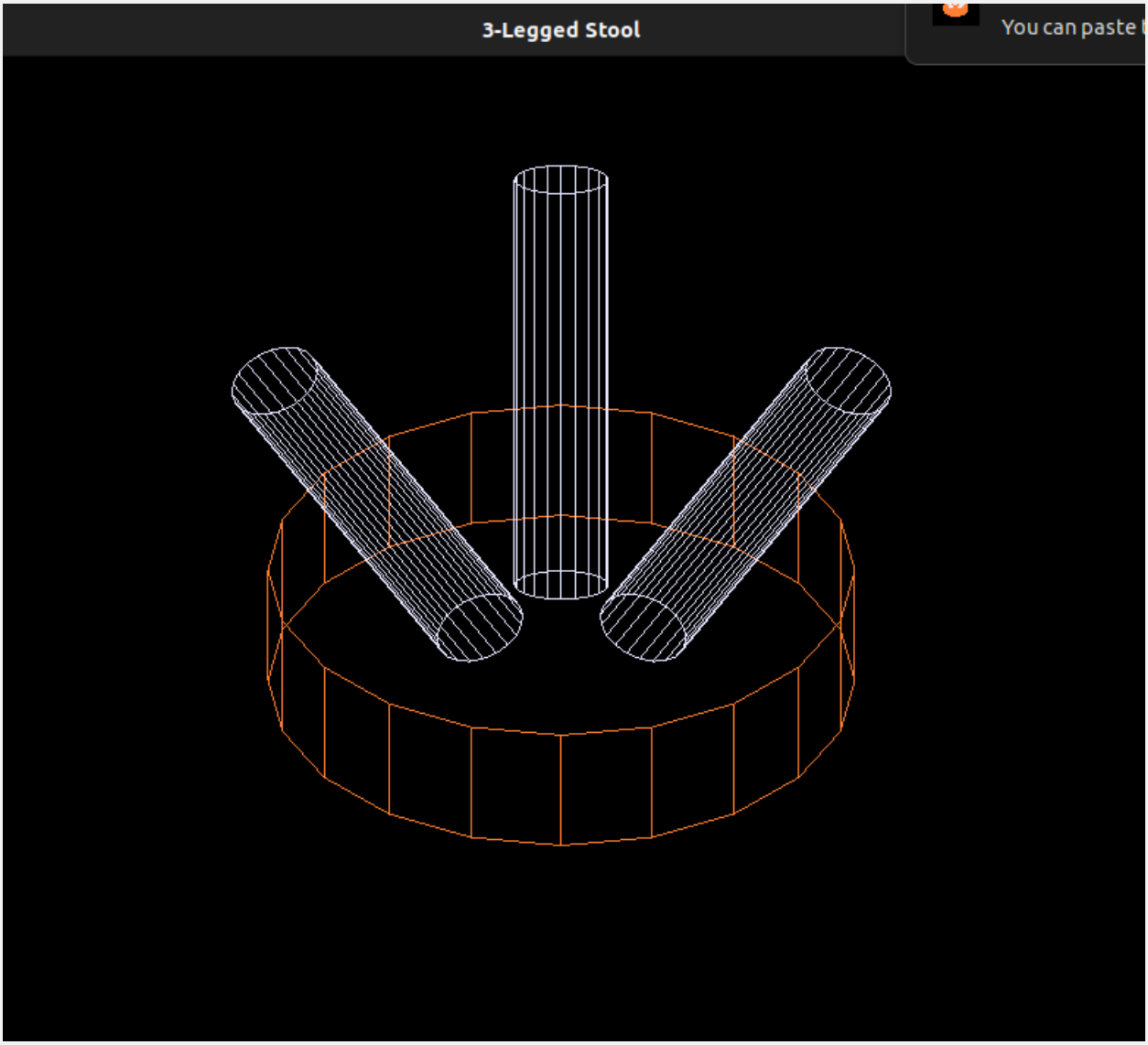



3-Legged Stool
You can paste


3-Legged Stool


3-Legged Stool


3-Legged Stool


3-Legged Stool
You can paste