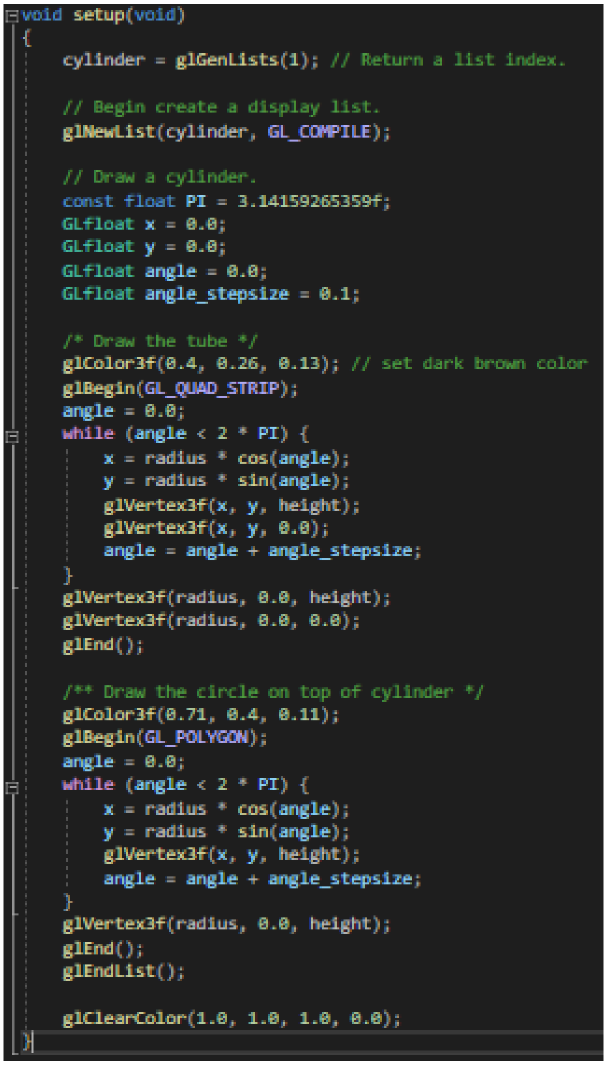
Lab 5 Graphics

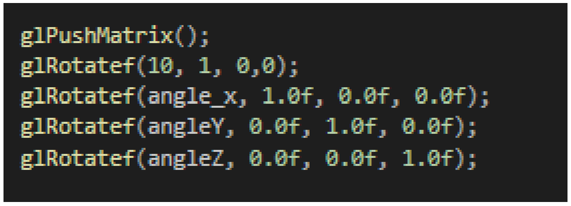
Code description:

- **In function setup():**
  First I begin creating list named **cylinder** then I begin drawing the cylinder and finally end the list



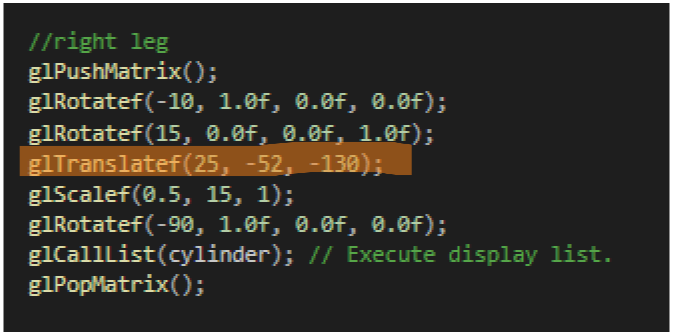
```
void setup(void)
{
    cylinder = glGenLists(1); // Return a list index.

    // Begin create a display list.
    glNewList(cylinder, GL_COMPILE);

    // Draw a cylinder.
    const float PI = 3.14159265359f;
    GLfloat x = 0.0;
    GLfloat y = 0.0;
    GLfloat angle = 0.0;
    GLfloat angle_stepsize = 0.1;

    /* Draw the tube */
    glColor3f(0.4, 0.26, 0.13); // set dark brown color
    glBegin(GL_QUAD_STRIP);
    angle = 0.0;
    while (angle < 2 * PI) {
        x = radius * cos(angle);
        y = radius * sin(angle);
        glVertex3f(x, y, height);
        glVertex3f(x, y, 0.0);
        angle = angle + angle_stepsize;
    }
    glVertex3f(radius, 0.0, height);
    glVertex3f(radius, 0.0, 0.0);
    glEnd();

    /** Draw the circle on top of cylinder */
    glColor3f(0.71, 0.4, 0.11);
    glBegin(GL_POLYGON);
    angle = 0.0;
    while (angle < 2 * PI) {
        x = radius * cos(angle);
        y = radius * sin(angle);
        glVertex3f(x, y, height);
        angle = angle + angle_stepsize;
    }
    glVertex3f(radius, 0.0, height);
    glEnd();
    glEndList();

    glClearColor(1.0, 1.0, 1.0, 0.0);
}
```

- **In function drawScene():**
  - First I start push matrix to allow the rotates if the user enter x,y,z.



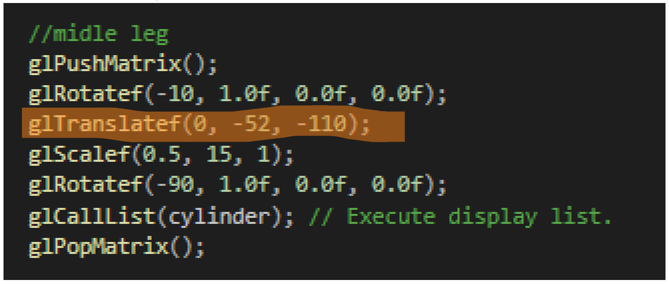
```
glPushMatrix();
glRotatef(10, 1, 0,0);
glRotatef(angle_x, 1.0f, 0.0f, 0.0f);
glRotatef(angleY, 0.0f, 1.0f, 0.0f);
glRotatef(angleZ, 0.0f, 0.0f, 1.0f);
```

  - Then I draw right leg



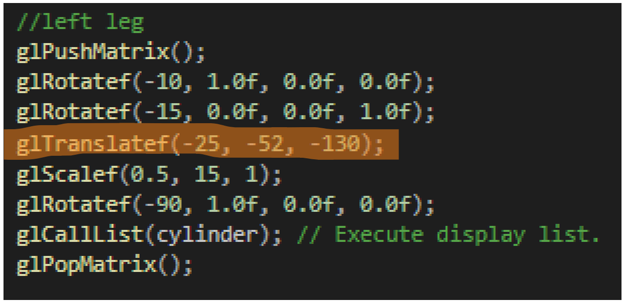
```
//right leg
glPushMatrix();
glRotatef(-10, 1.0f, 0.0f, 0.0f);
glRotatef(15, 0.0f, 0.0f, 1.0f);
glTranslatef(25, -52, -130);
glScalef(0.5, 15, 1);
glRotatef(-90, 1.0f, 0.0f, 0.0f);
glCallList(cylinder); // Execute display list.
glPopMatrix();
```

  - Middle leg



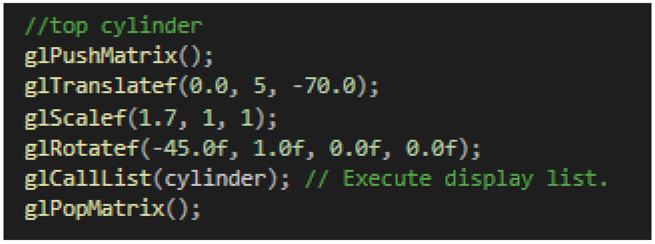
```
//midle leg
glPushMatrix();
glRotatef(-10, 1.0f, 0.0f, 0.0f);
glTranslatef(0, -52, -110);
glScalef(0.5, 15, 1);
glRotatef(-90, 1.0f, 0.0f, 0.0f);
glCallList(cylinder); // Execute display list.
glPopMatrix();
```

  - Left leg



```
//left leg
glPushMatrix();
glRotatef(-10, 1.0f, 0.0f, 0.0f);
glRotatef(-15, 0.0f, 0.0f, 1.0f);
glTranslatef(-25, -52, -130);
glScalef(0.5, 15, 1);
glRotatef(-90, 1.0f, 0.0f, 0.0f);
glCallList(cylinder); // Execute display list.
glPopMatrix();
```

  - Top cylinder

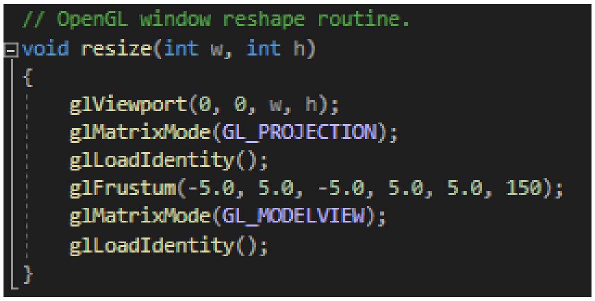


```
//top cylinder
glPushMatrix();
glTranslatef(0.0, 5, -70.0);
glScalef(1.7, 1, 1);
glRotatef(-45.0f, 1.0f, 0.0f, 0.0f);
glCallList(cylinder); // Execute display list.
glPopMatrix();
```

❖ Note in legs:

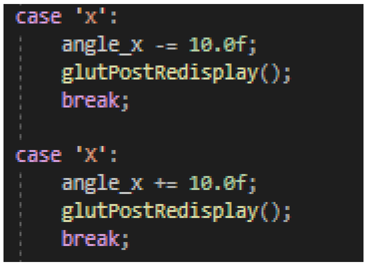
all legs are the same **scale** and middle leg is the nearest one

- **resize function:**



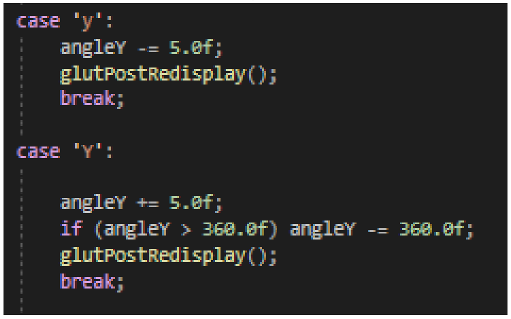
```
// OpenGL window reshape routine.
void resize(int w, int h)
{
    glViewport(0, 0, w, h);
    glMatrixMode(GL_PROJECTION);
    glLoadIdentity();
    glFrustum(-5.0, 5.0, -5.0, 5.0, 5.0, 150);
    glMatrixMode(GL_MODELVIEW);
    glLoadIdentity();
}
```

- **keyboard function:**
  - to rotate around x:



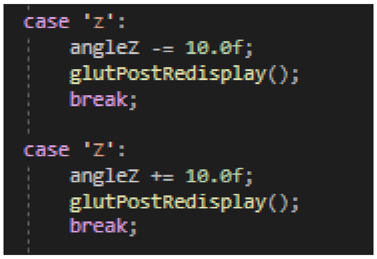
```
case 'x':
    angle_x -= 10.0f;
    glutPostRedisplay();
    break;

case 'X':
    angle_x += 10.0f;
    glutPostRedisplay();
    break;
```

  - to rotate around y:



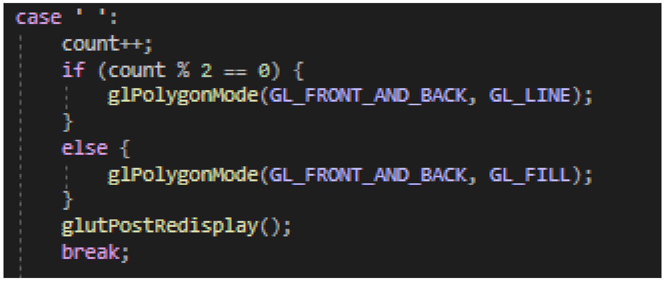
```
case 'y':
    angleY -= 5.0f;
    glutPostRedisplay();
    break;

case 'Y':

    angleY += 5.0f;
    if (angleY > 360.0f) angleY -= 360.0f;
    glutPostRedisplay();
    break;
```

  - to rotate around z:



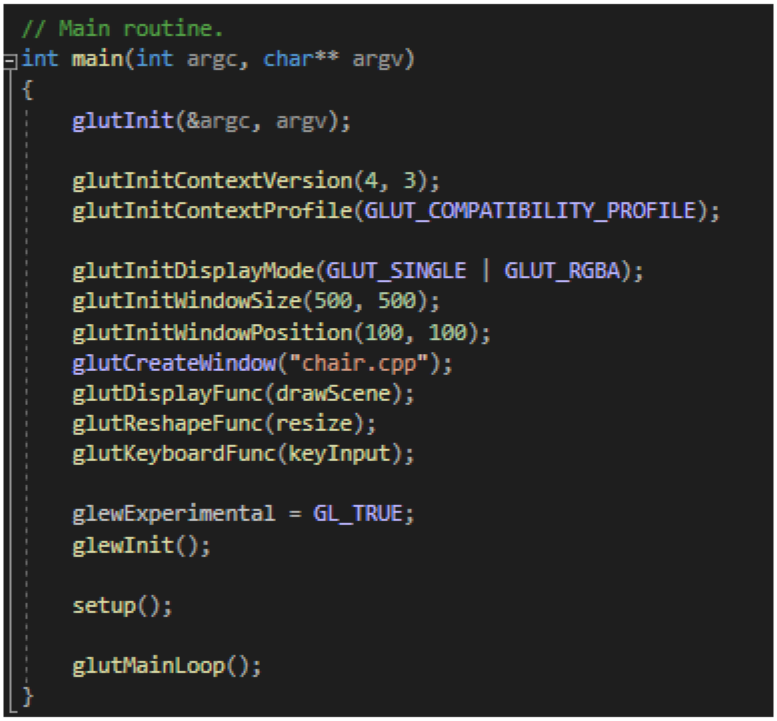
```
case 'z':
    angleZ -= 10.0f;
    glutPostRedisplay();
    break;

case 'Z':
    angleZ += 10.0f;
    glutPostRedisplay();
    break;
```

  - If the user enter space:



```
case ' ':
    count++;
    if (count % 2 == 0) {
        glPolygonMode(GL_FRONT_AND_BACK, GL_LINE);
    }
    else {
        glPolygonMode(GL_FRONT_AND_BACK, GL_FILL);
    }
    glutPostRedisplay();
    break;
```

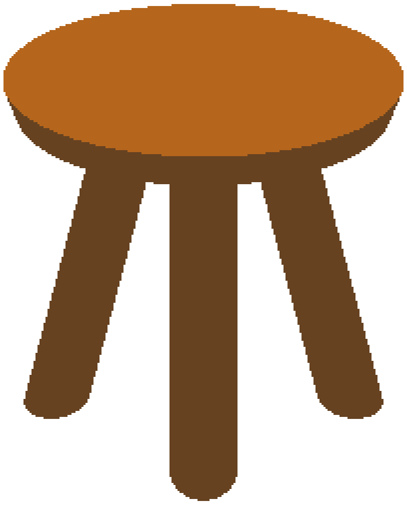
- **Main function:**



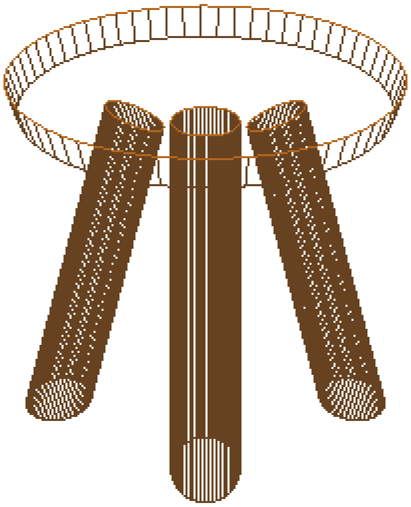
```
// Main routine.
int main(int argc, char** argv)
{
    glutInit(&argc, argv);

    glutInitContextVersion(4, 3);
    glutInitContextProfile(GLUT_COMPATIBILITY_PROFILE);

    glutInitDisplayMode(GLUT_SINGLE | GLUT_RGBA);
    glutInitWindowSize(500, 500);
    glutInitWindowPosition(100, 100);
    glutCreateWindow("chair.cpp");
    glutDisplayFunc(drawScene);
    glutReshapeFunc(resize);
    glutKeyboardFunc(keyInput);

    glewExperimental = GL_TRUE;
    glewInit();

    setup();

    glutMainLoop();
}
```

Runs:

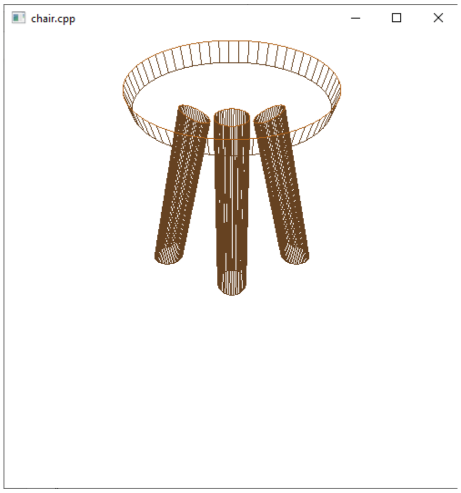
- **First screen:**

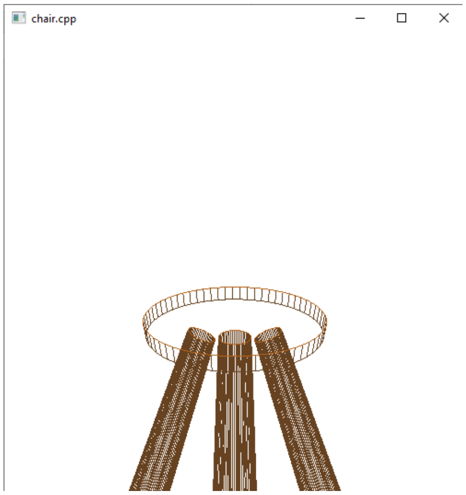


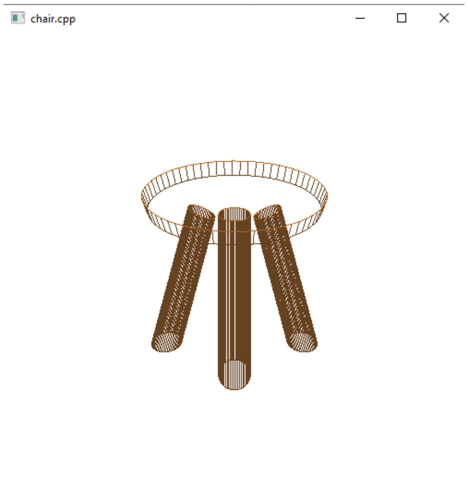
➢ Enter space:



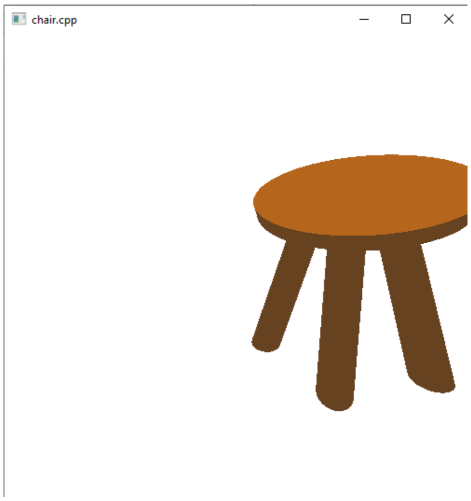
➢ Enter X, x:

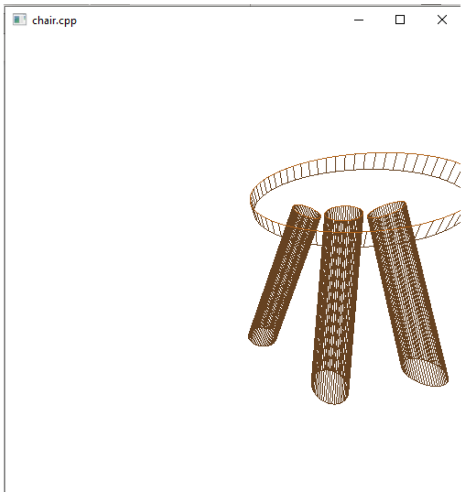


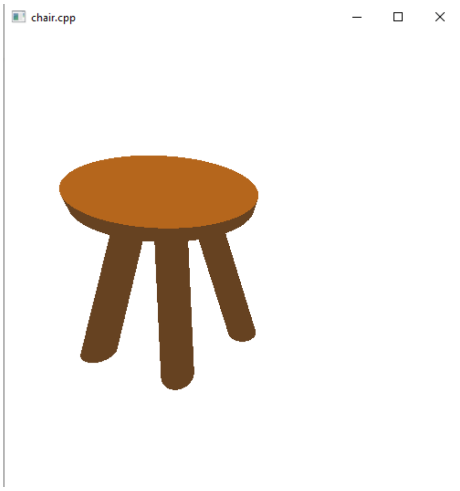




➢ Enter Y, y:







➢ Enter Z, z: