Alexandria University

Faculty of Engineering

Computer Graphics

Lab #5 OpenGL

Name: Toka Ashraf Abo Elwafa Ahmed

ID: 19015539

Problem Statement:

You are required to implement an application that draws a 3-legged stool as shown in figure 1. For the legs, first create one in a display list and then draw it three times rotated appropriately using appropriate transformation. Your application should handle user input at runtime as follows:

- When user presses ‘x’ / ‘X’. the 3-legged stool should rotate around x-axis in CW/CCW manner respectively.

- When user presses ‘y’ / ‘Y’. the 3-legged stool should rotate around y-axis in CW/CCW manner respectively.

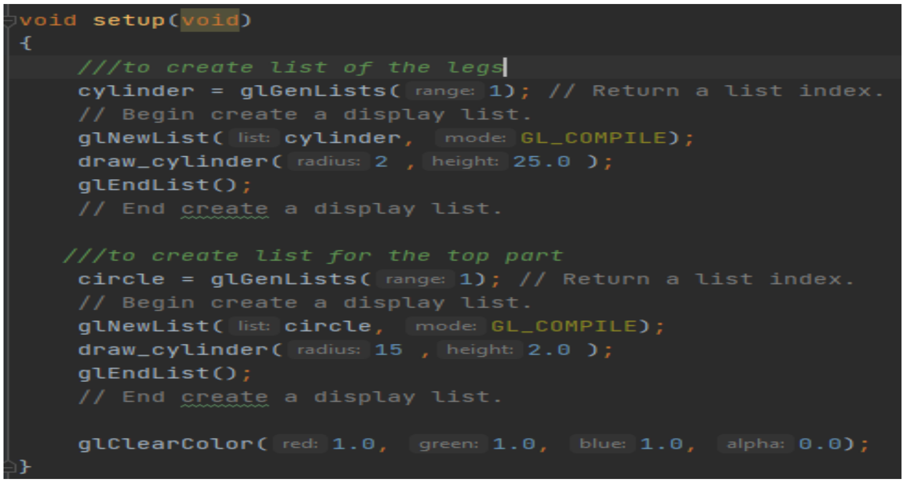
- When user presses ‘z’ / ‘Z’. the 3-legged stool should rotate around z-axis in CW/CCW manner respectively.

- When user presses ‘space’. Toggle drawing mode of the 3-legged between drawing a wireframe object or a solid one.

Code Flow:

- **Function of setup():**

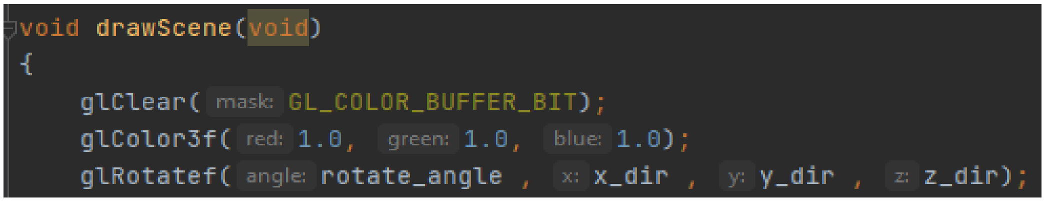
  In this function, two lists are created: one to draw the legs of the chair and the other to draw the top part of it as the radius and height of the legs differ from the top part owns.



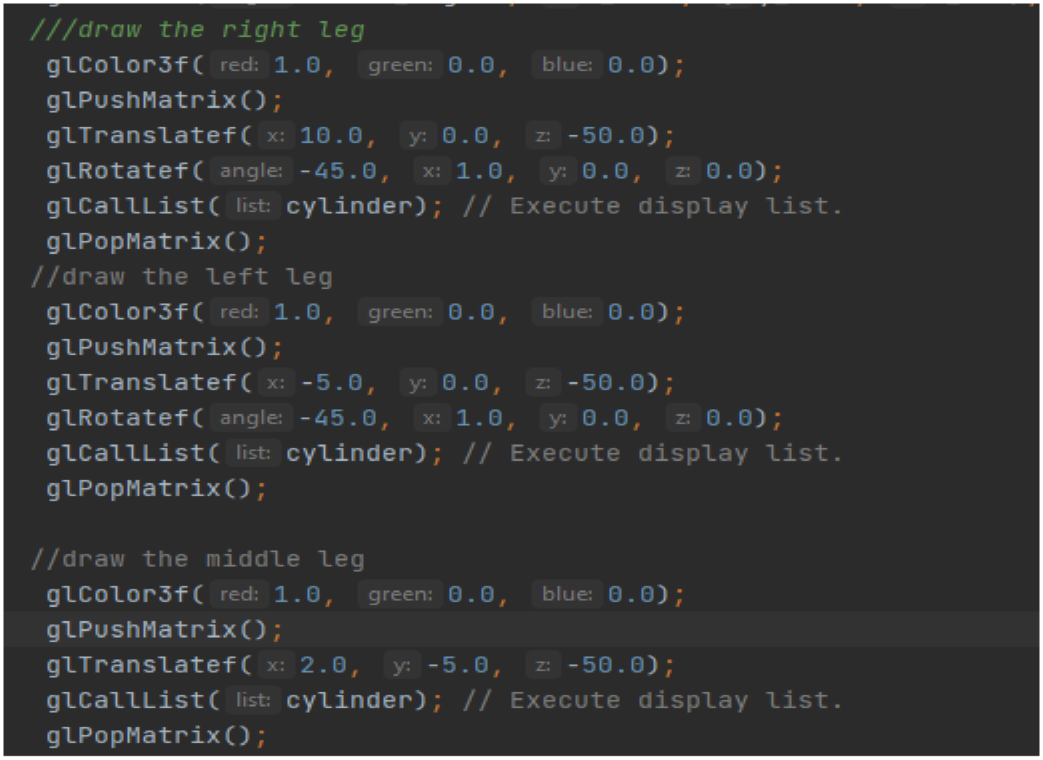
```
void setup(void)
{
    ///to create list of the legs
    cylinder = glGenLists( range: 1); // Return a list index.
    // Begin create a display list.
    glNewList( list: cylinder,  mode: GL_COMPILE);
    draw_cylinder( radius: 2 , height: 25.0 );
    glEndList();
    // End create a display list.

   ///to create list for the top part
    circle = glGenLists( range: 1); // Return a list index.
    // Begin create a display list.
    glNewList( list: circle,  mode: GL_COMPILE);
    draw_cylinder( radius: 15 , height: 2.0 );
    glEndList();
    // End create a display list.

    glClearColor( red: 1.0,  green: 1.0,  blue: 1.0,  alpha: 0.0);
}
```

- ### Function of drawScene():

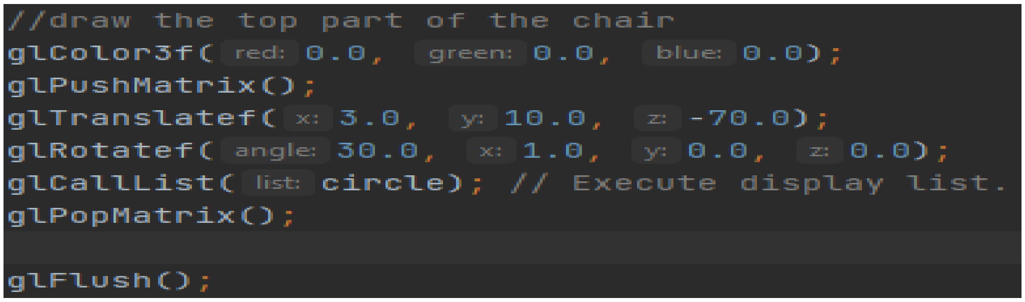
  In this function, the chair is drawn by drawing each part of it in the right position and rotation of it by pushing matrix for each part and apply the rotations and what it needs then pop it and push another matrix and so on. But there is some rotation is applied on the whole chair around certain axis that the user chooses and with certain angle all these variables are initialized to zero then changed according to the interaction of the user.



```
void drawScene(void)
{
    glClear( mask: GL_COLOR_BUFFER_BIT);
    glColor3f( red: 1.0,  green: 1.0,  blue: 1.0);
    glRotatef( angle: rotate_angle ,  x: x_dir ,  y: y_dir ,  z: z_dir);
```



```
 ///draw the right leg
  glColor3f( red: 1.0,  green: 0.0,  blue: 0.0);
  glPushMatrix();
  glTranslatef( x: 10.0,  y: 0.0,  z: -50.0);
  glRotatef( angle: -45.0,  x: 1.0,  y: 0.0,  z: 0.0);
  glCallList( list: cylinder); // Execute display list.
  glPopMatrix();
 //draw the left leg
  glColor3f( red: 1.0,  green: 0.0,  blue: 0.0);
  glPushMatrix();
  glTranslatef( x: -5.0,  y: 0.0,  z: -50.0);
  glRotatef( angle: -45.0,  x: 1.0,  y: 0.0,  z: 0.0);
  glCallList( list: cylinder); // Execute display list.
  glPopMatrix();

 //draw the middle leg
  glColor3f( red: 1.0,  green: 0.0,  blue: 0.0);
  glPushMatrix();
  glTranslatef( x: 2.0,  y: -5.0,  z: -50.0);
  glCallList( list: cylinder); // Execute display list.
  glPopMatrix();
```

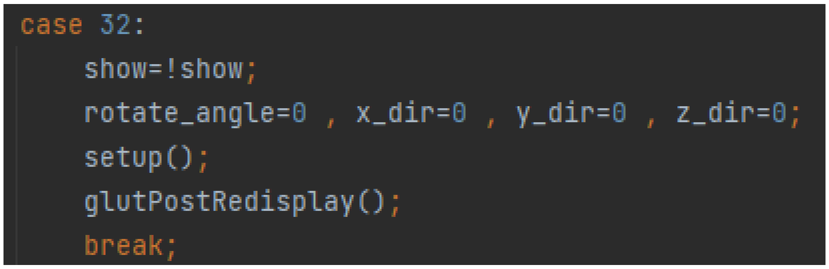


```
//draw the top part of the chair
glColor3f( red: 0.0,  green: 0.0,  blue: 0.0);
glPushMatrix();
glTranslatef( x: 3.0,  y: 10.0,  z: -70.0);
glRotatef( angle: 30.0,  x: 1.0,  y: 0.0,  z: 0.0);
glCallList( list: circle); // Execute display list.
glPopMatrix();

glFlush();
```

Function of keyInput():

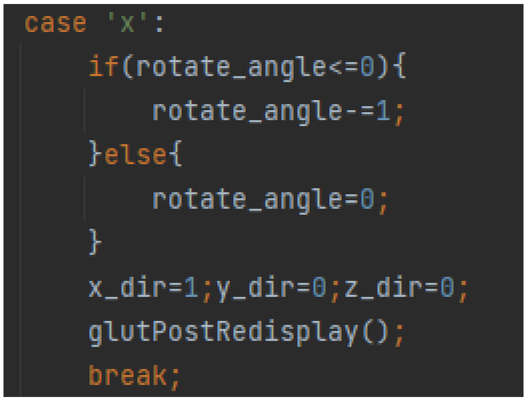
In this function, user interaction is applied:

- When pressing the space button that represents number 32 in the keyboard, it toggle between the wireframe object and solid one by making a Boolean global variable and initialize it to false and if this variable equals false draw in wireframe object else draw in solid one by making this variable equals its negation and make all variables of the rotation part be zeros and then call function of setup and glutPostRedisplay to redraw after the modification.



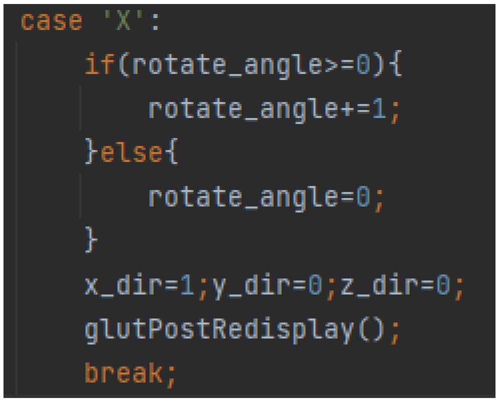
```
case 32:
    show=!show;
    rotate_angle=0 , x_dir=0 , y_dir=0 , z_dir=0;
    setup();
    glutPostRedisplay();
    break;
```

- When pressing x, it rotates around x axis in clockwise manner by decreasing the angle if it less than or equal to zero and set it to zero if bigger than zero then assign the x_dir to 1 to rotate around x axis.



```
case 'x':
    if(rotate_angle<=0){
        rotate_angle-=1;
    }else{
        rotate_angle=0;
    }
    x_dir=1;y_dir=0;z_dir=0;
    glutPostRedisplay();
    break;
```

- When pressing X, it does the same as the previous one but it increases the angle if it greater than or equals to zero and set to zero otherwise.

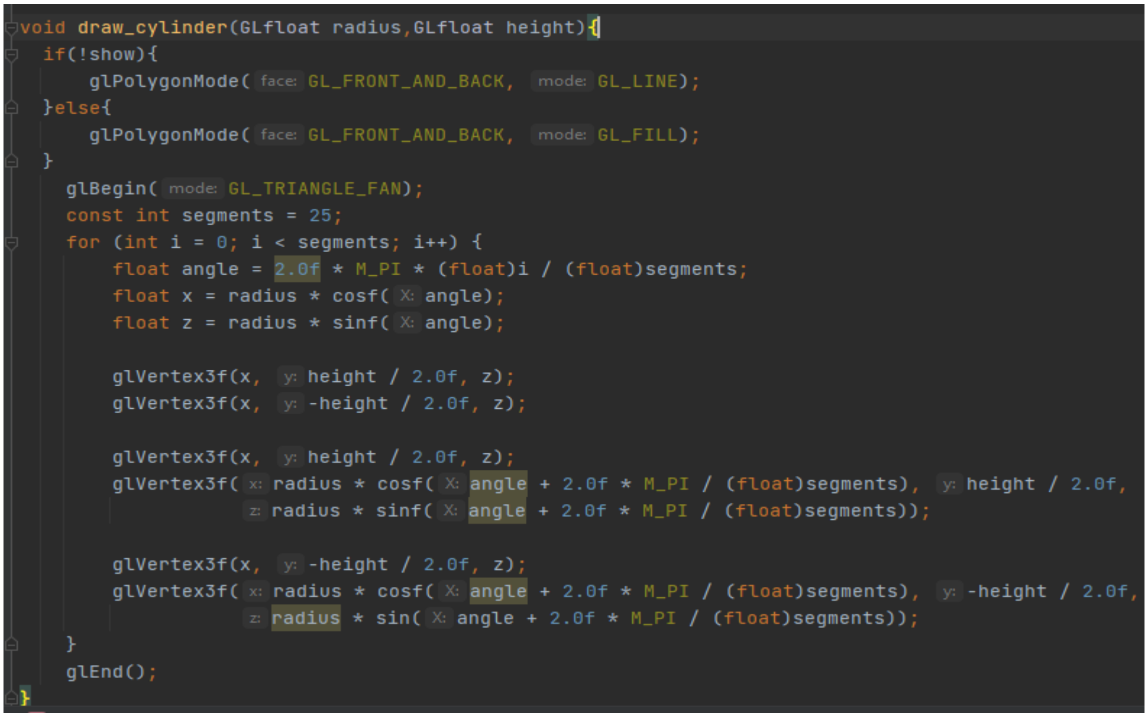


```
case 'X':
    if(rotate_angle>=0){
        rotate_angle+=1;
    }else{
        rotate_angle=0;
    }
    x_dir=1;y_dir=0;z_dir=0;
    glutPostRedisplay();
    break;
```

- When pressing y, Y, z, or Z, it acts as x manner but each of them in its direction put it variables equals 1 and the others set to zero.
- After changing the angle and the variables that determines the direction of the rotation, glutPostRedisplay is called to redraw the scene.

Function of draw_cylinder:

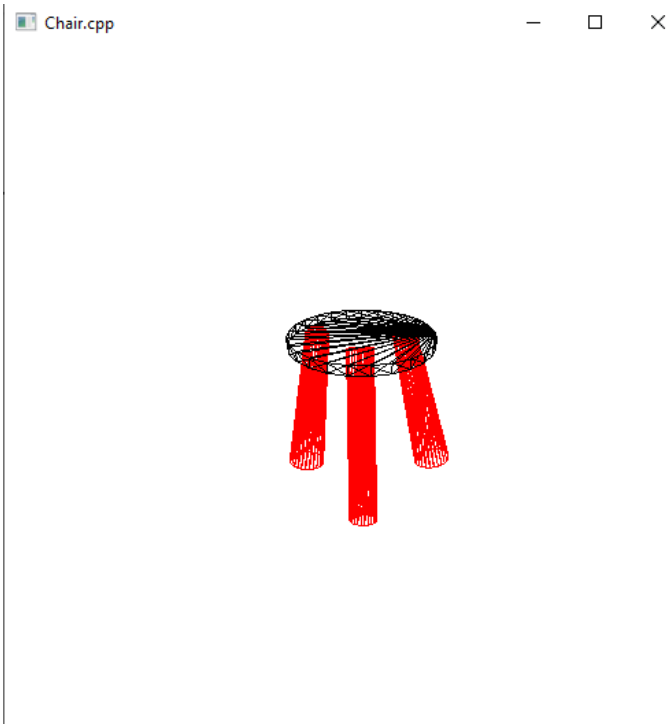
This function takes the radius and height as parameters and draw the cylinder. It is used to draw the three legs and the top part with the appropriate radius and height.



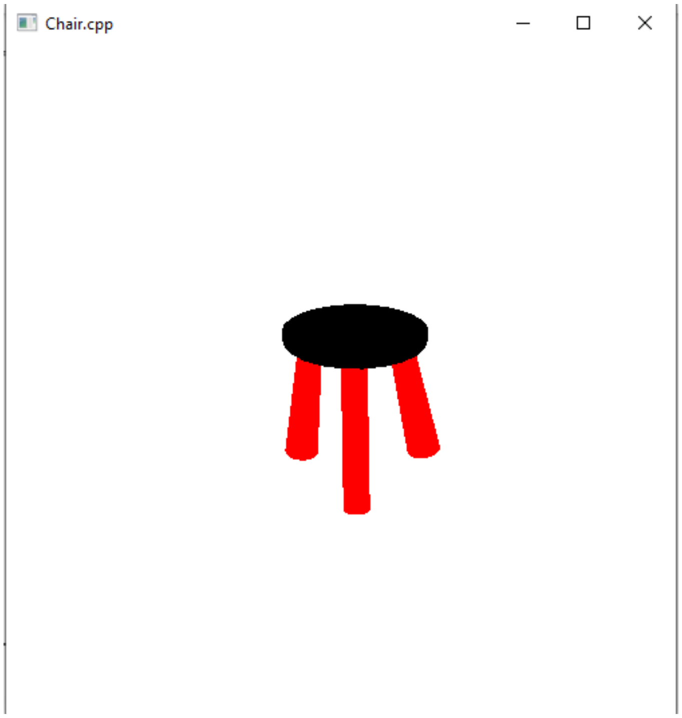
```
void draw_cylinder(GLfloat radius,GLfloat height){
  if(!show){
      glPolygonMode( face: GL_FRONT_AND_BACK,  mode: GL_LINE);
  }else{
      glPolygonMode( face: GL_FRONT_AND_BACK,  mode: GL_FILL);
  }
    glBegin( mode: GL_TRIANGLE_FAN);
    const int segments = 25;
    for (int i = 0; i < segments; i++) {
        float angle = 2.0f * M_PI * (float)i / (float)segments;
        float x = radius * cosf( X: angle);
        float z = radius * sinf( X: angle);

        glVertex3f(x,  y: height / 2.0f, z);
        glVertex3f(x,  y: -height / 2.0f, z);

        glVertex3f(x,  y: height / 2.0f, z);
        glVertex3f( x: radius * cosf( X: angle + 2.0f * M_PI / (float)segments),  y: height / 2.0f,
                    z: radius * sinf( X: angle + 2.0f * M_PI / (float)segments));

        glVertex3f(x,  y: -height / 2.0f, z);
        glVertex3f( x: radius * cosf( X: angle + 2.0f * M_PI / (float)segments),  y: -height / 2.0f,
                    z: radius * sin( X: angle + 2.0f * M_PI / (float)segments));
    }
    glEnd();
}
```

Sample Runs:

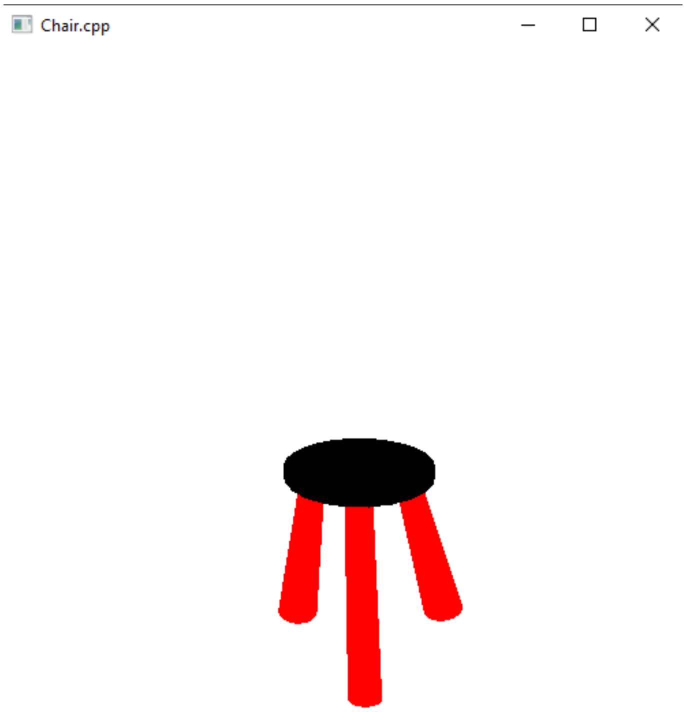
Once you run the program, this will appear:



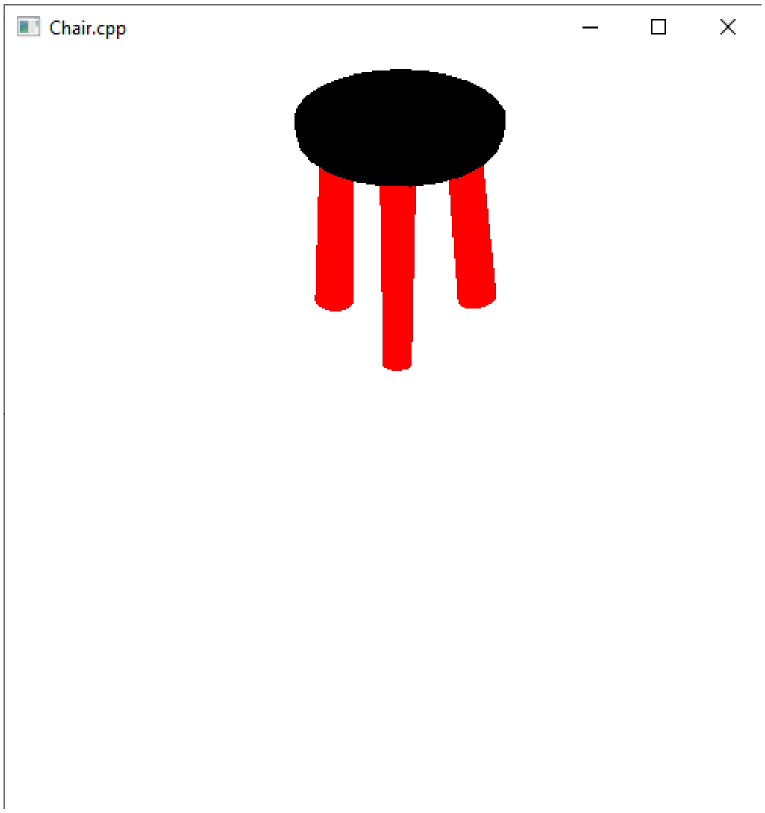
After pressing space button:



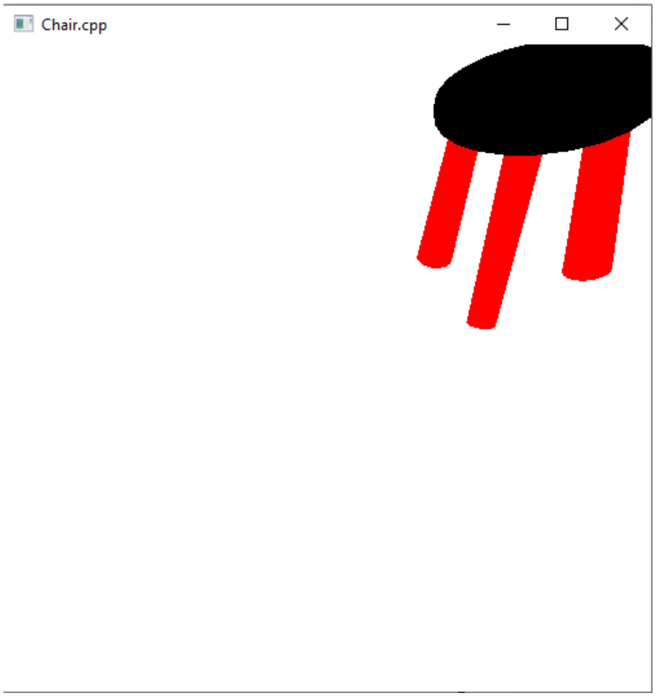
After pressing x to rotate in x direction CW:



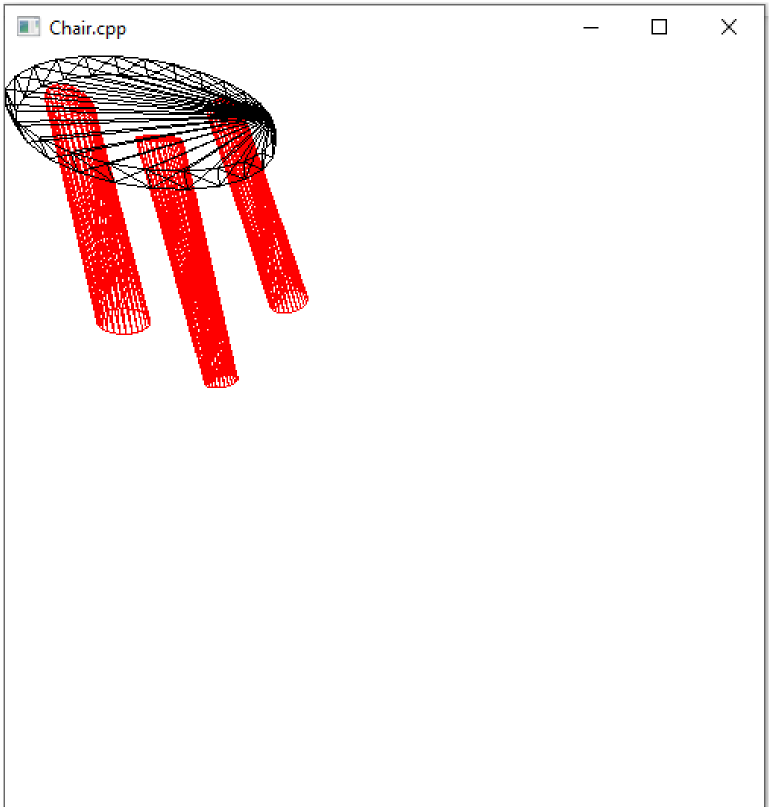
After pressing X:



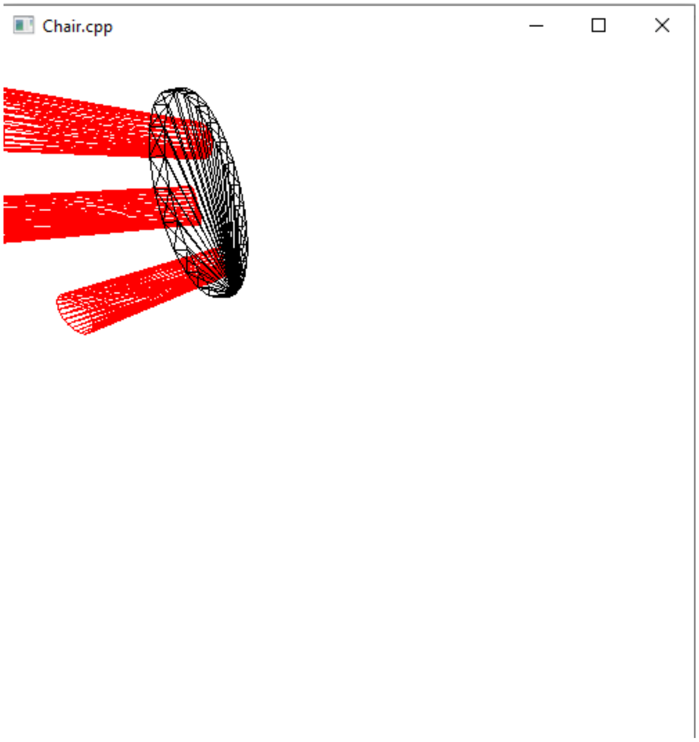
After pressing y:



After pressing space button then Y:



After pressing z:



After pressing space then Z: