3D Graphics Coursework #1

000540476

Toby Gates

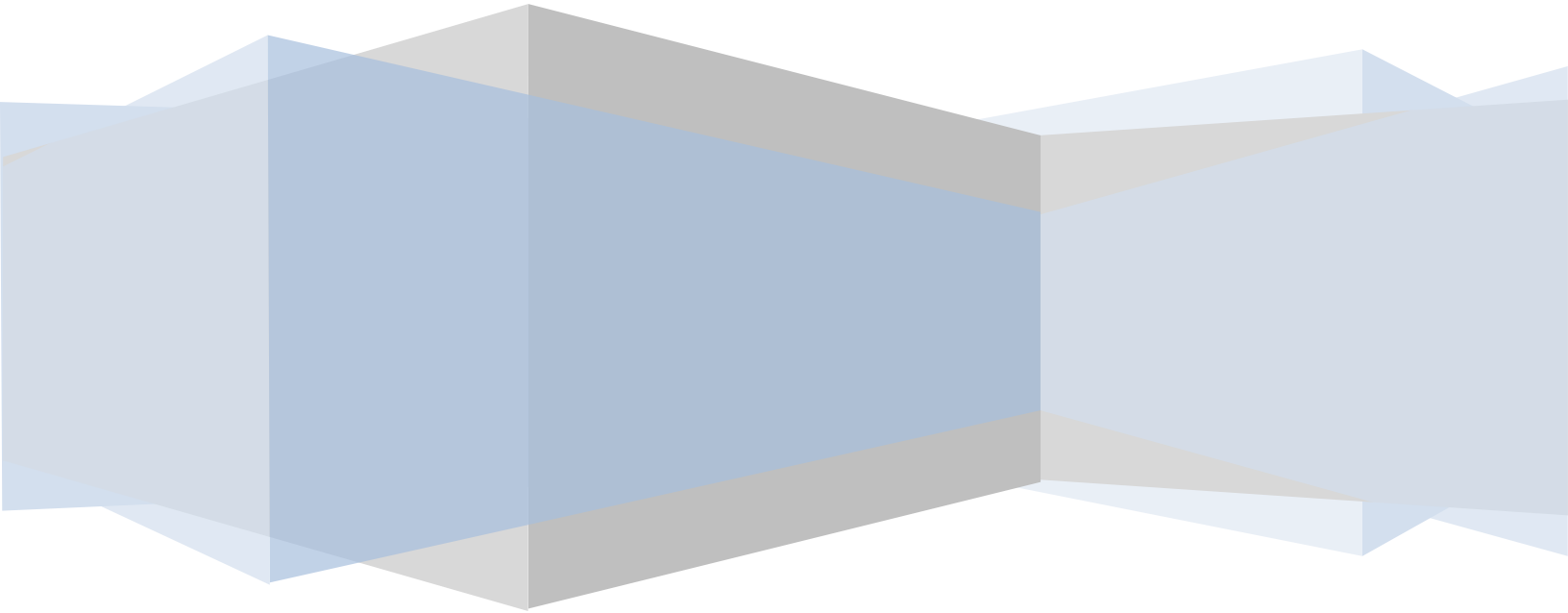


Table of Contents

How the scene was designed and came to be what it is now

The scene was mostly designed by brainstorming the different kinds of robots present in everyday life as well as the robots that were present in fiction and what they are able to do. This was a difficult decision as there were quite a few to choose from; however in the labs when working on the lab work I discussed robots with a couple of people including the lab tutor Carl and he suggested various robots. One of these robot ideas happened to be a transformer which led me to make use of a reference image of a transformer for the general design for the robot, bar a few tweaks.

At first I just coded in the default robot class with comments indicating which piece of code was for which part of the robot, this was done in order to not have to think about the creation of different classes yet as my idea for a robot still wasn't too much of a solid idea due to wanting it to be similar to the reference image and yet slightly unique compared to it too. So I made use of glut objects to create different robot shapes whilst making a few tweaks to my robot in the process. When it was completed, I then needed to go about rendering and adding lighting to the scene; lighting was added by adding certain bits of code to both the display() and init () methods, once lighting was added it allowed me to push forward and work on the rendering.

Rendering was undertaken based around the idea that I required the use of 3 colours for my robot; metallic silver, a redish/brownish colour and a deep blue colour. This was done by creating 3 float arrays (which represented the Ambient, Diffuse and Specular attributes) for each colour; after these were created each glut object that was drawn in the display method simply required 3 lines of code which made use of a particular colour that I made available. Aside from colour there were also two other single float values for shine which were added; one of these produced a high level of shine whilst the other was more of a low dull effect rather than an actual shine. The shine was mainly use for the silvery colour which made it look metallic whilst the dull effect was used for other objects such as the hands or arms for example which turned out to be rendered as a darker red or a darker blue colour.

A decent level of camera work was added to the scene in order to boost interactivity levels between the scene and the user. Features that allow the user to move the view vertically, zooming in and out and change between different camera views were added and are also documented later in this report. The background of the scene was worked on after the camera work had been completed as it was easier to tell that it was working without having a half finished scene background in the mix whilst rotating and zooming.

Having a feature to turn the lights on and off was decided somewhat at the mid-point of the project as it would add more interaction to the scene and allow the user to turn on and off the lights. It would also show how the same scene would look different when everything is hit by the lights and also when everything is hit with no lights. This was an easier part of the project due to the fact that some of the stuff we had covered in labs required us to turn lights on and off.

I had many different ideas for an animation or two such as; firing rockets off, transforming and so forth. However, it was implemented late into development so I decided to create and implement something

called Battle Orb Animation which has two golden orbs rotating the robot like a shield. One orb rotates one way whilst the other orb rotates the other way leading to more complete shield effect rather than having both orbs move the same way.

The shift button on the keyboard is used in order to activate and deactivate the orbs which rotate the robot. One thing to note about deactivation is that sometimes it deactivates and the orbs disappear and sometimes it deactivates successfully but the orbs have shown to have stopped in the scene until the user moves the screen.

User Interaction included in the Robot's Scene

Left and Right Rotation (Default Feature)

This feature was initially included in the dummy template for the robot; it allows the user to click the left and right mouse key buttons in order to rotate the scene and everything in it left and right. There is a screenshot demonstrating this feature (even though it is a default one) in Appendix A.

Up and Down Rotation

This feature works similarly to the default left and right rotation except for the fact that it rotates the screen up and down rather than left and right. It also uses the up and down arrow keys to do this instead of the left and right keys. There are screenshots in Appendix A demonstrating this feature.

The ability to change camera views

This feature involves various different views of the scene which contains the robot model and ways of the user accessing them. These camera views are added to the accessibility between the user and the robot program. There are also screenshots in Appendix A for a demonstration of this feature too.

Zoom Feature

This feature involves the click of a button to either zoom into and closer to the objects found in the scene or the click of a button to zoom outwards and further away so more things in the scene can fit on screen. Further screenshots of Zooming in action can be found in Appendix A of this report.

Toggling Lights On and Off

This feature involves turning the lights off and on in the scene. It uses two buttons one to turn the lights off and the other to turn the lights on instead of using the same button for both and relying on the R button to reset the whole scene (including the lights). Screenshots of this feature can also be found in Appendix A.

Animation

This feature involves animation implemented into the robot scene; one animation implemented is the battle orbs which involves two orbs rotating the robot functioning as a shield. The animation is started by the user and also can be stopped by the user so it counts towards user interaction. Screenshots of this animation in action can be found in Appendix A of this report.

Difficulties encounter1ed whilst creating the Robot and how a solution to these difficulties did or didn't come about

Whilst creating the robot, a problem with the cube used for the robot's shoulder occurred. This happens when the user rotates the screen in any direction; after about a second or two of rotating the left shoulder seems to rotate slightly forward and move the scene camera to a different angle instead of the angle it started at.
A solution to this problem was removing the piece of code saying "gl.glRotatef(20.0f, 1.0f, 0.0f, 0.0f);" in the display() method which rotated the first drawn object in the display method() by 20 degrees on the X-Axis which happened to be the cube used for the shoulder of my robot.

Appendices

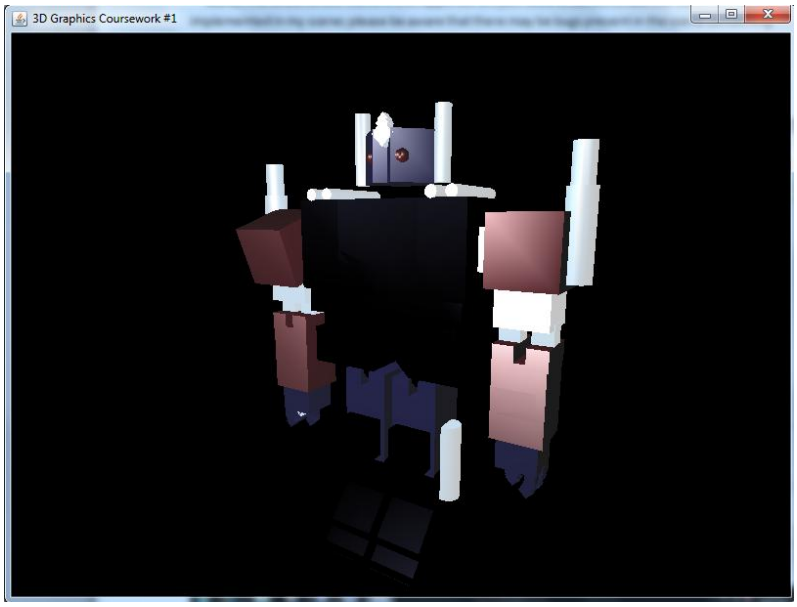
Appendix A – Screenshots of scene demonstrating the features available

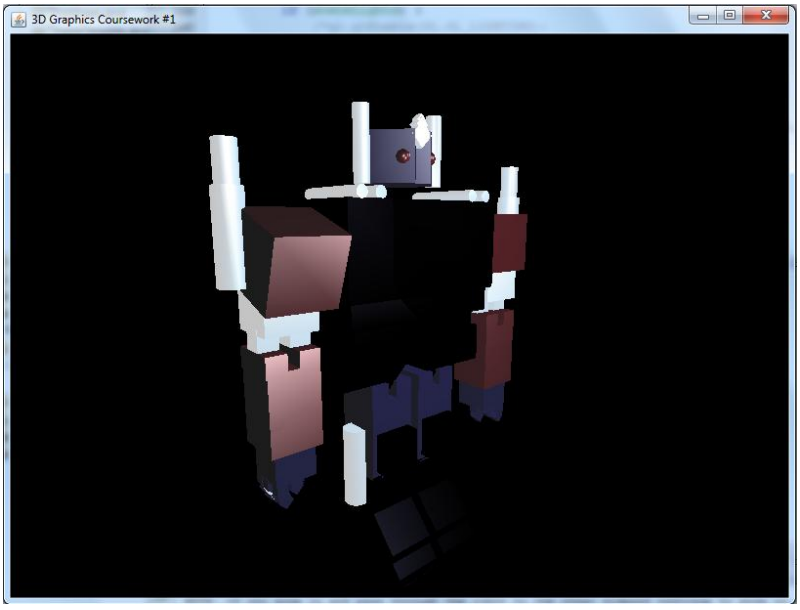
**NOTE: The screenshots found in this Appendix are just used to demonstrate the features that are implemented in my scene; please be aware that there may be bugs present in the scene concerning other things as well as parts and objects missing from the scene as opposed to the final version, this is because these screenshots were taken when it was still a work in progress. **

Left <–> Right Rotation

The screenshot to the right shows an image of the robot after the screen show the robot has been rotated to the right.

NOTE: As you can see the shoulder part is messed up in this screenshot it is also the cause of the camera changing vertical angle without input too.





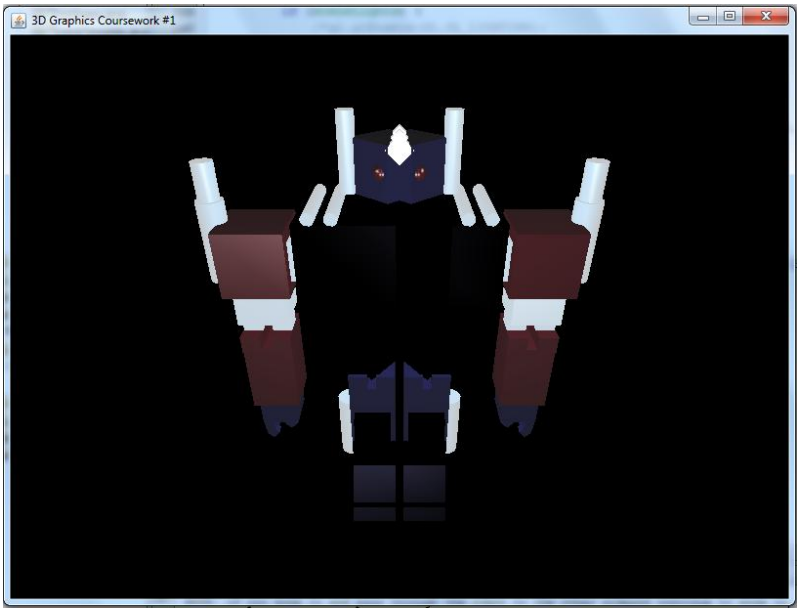
The screenshot to the left shows an image of the robot after the screen has been rotated to the left.

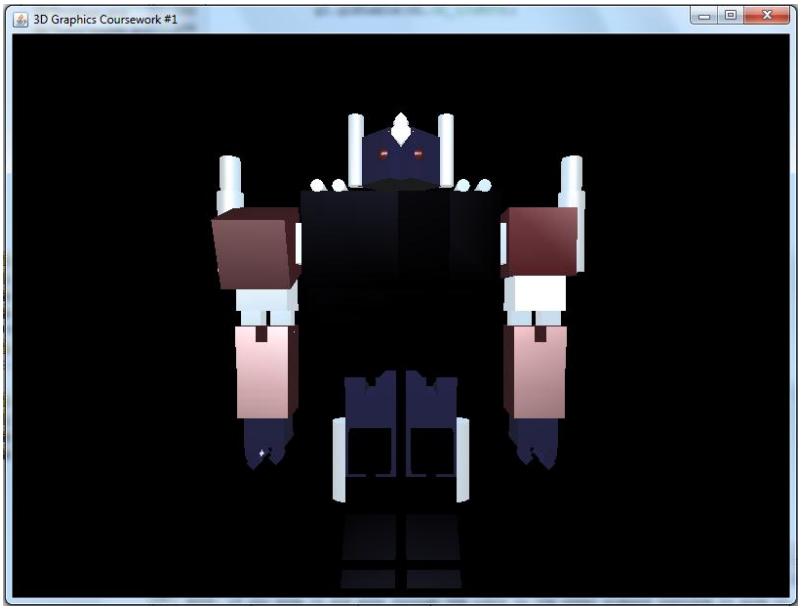
NOTE: Being similar to the picture before you can see the shoulder part is messed up in this screenshot it is also the cause of the camera changing vertical angle without input too.

Up <–> Down Rotation

The screenshot the right shows the robot scene in default view when the program starts with nothing applied to it.

It is also as high as the camera can go when rotating it vertically upwards.



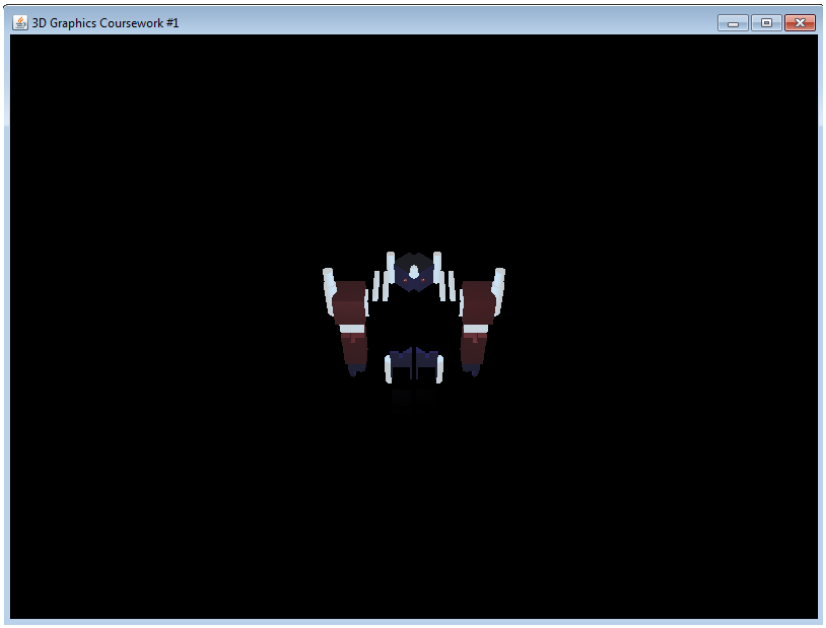


The screenshot to the left shows the robot scene after it has been rotated vertically downwards.

NOTE: As with before you can see the shoulder part is messed up in this screenshot it is also the cause of the camera changing vertical angle without input too.

The ability to change camera views

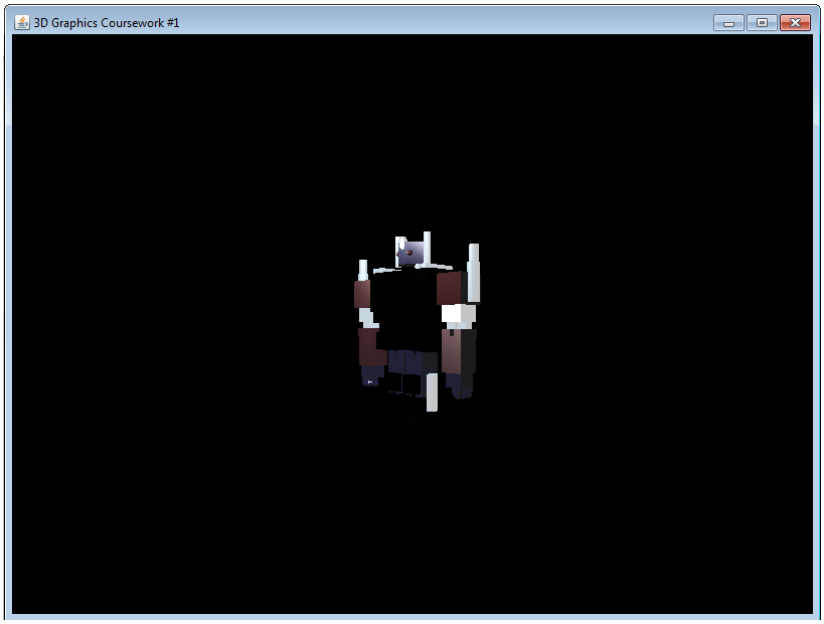
Key “1” (from Default View)



The screenshot to the left shows an alternative view if the 2 key is pressed whilst the program is in default view.

If the 2 key is pressed when the program is not in default view I believe something else happens.

Key “2” (from Default View)



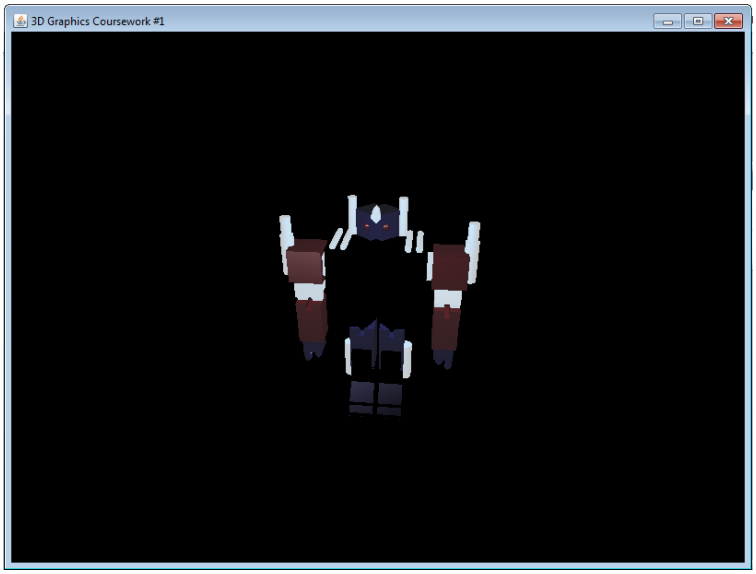
The screenshot to the left shows an alternative view if the 2 key is pressed whilst the program is in default view.

If the 2 key is pressed when the program is not in default view I believe something weirder happens.

Key "3" (from Default View)

The screenshot to the left shows an alternative view if the 3 key is pressed whilst the program is in default view.

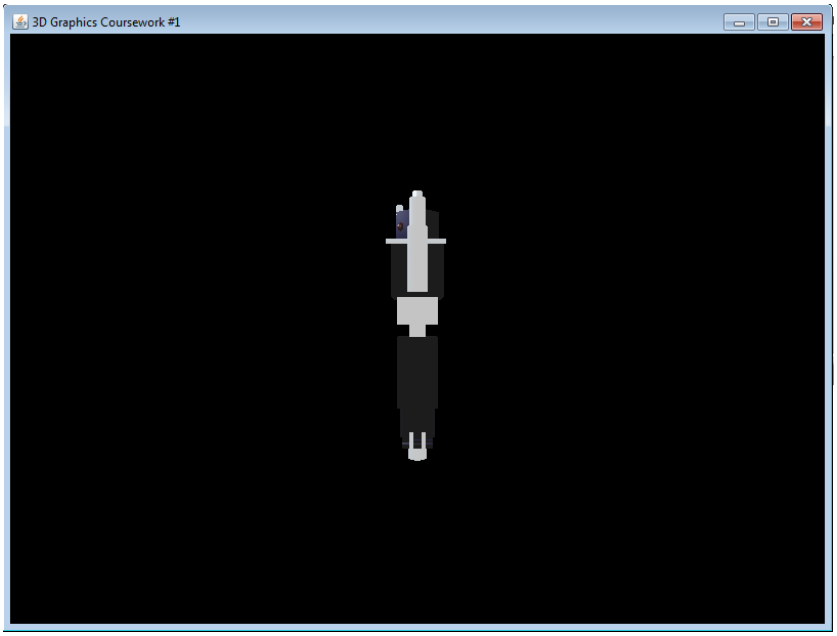
If the 2 key is pressed when the program is not in default view I believe something else tends to happen.



Key "4" (from Default View)

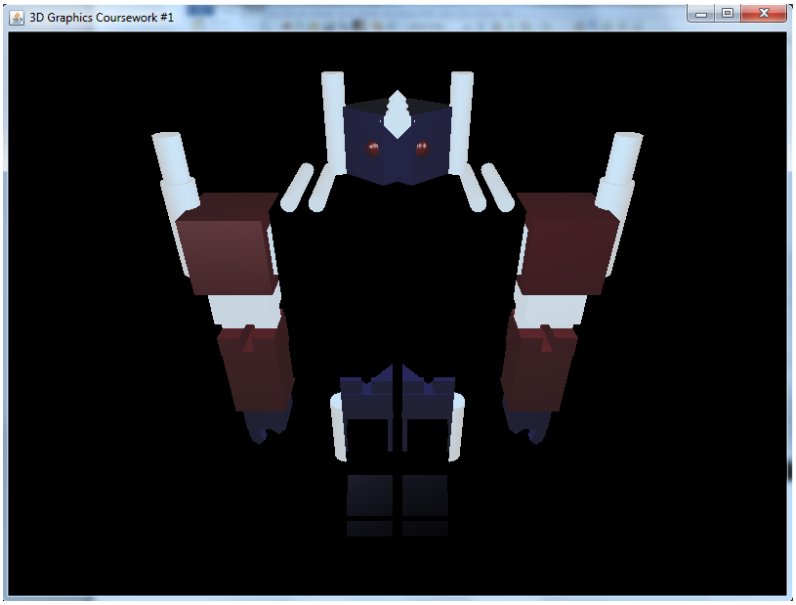
The screenshot to the left shows an alternative view if the 4 key is pressed whilst the program is in default view.

If the 2 key is pressed when the program is not in default view I believe something weirder happens.



Zoom Feature

Zoomed In



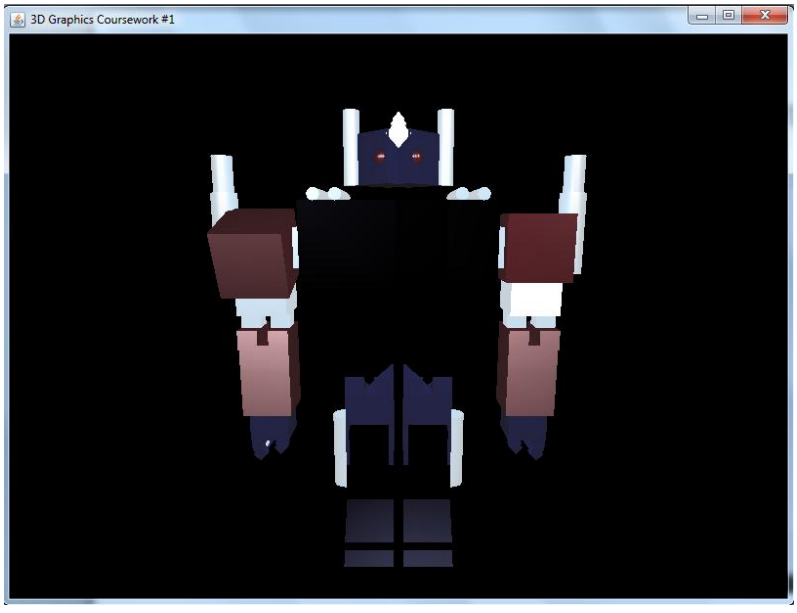
The screenshot to the left shows a more zoomed in shot to prove that the program does actually zoom in.

I believe if you keep zooming in the camera will then pass through the robot and zoom out displaying the back of the robot.

Zoomed Out

The screenshot to the right shows a more zoomed out image starting at the point that was shown in the zoomed in image.

If you keep zooming out past the furthest point of zooming out the camera will then reverse and start zooming in towards the robot.

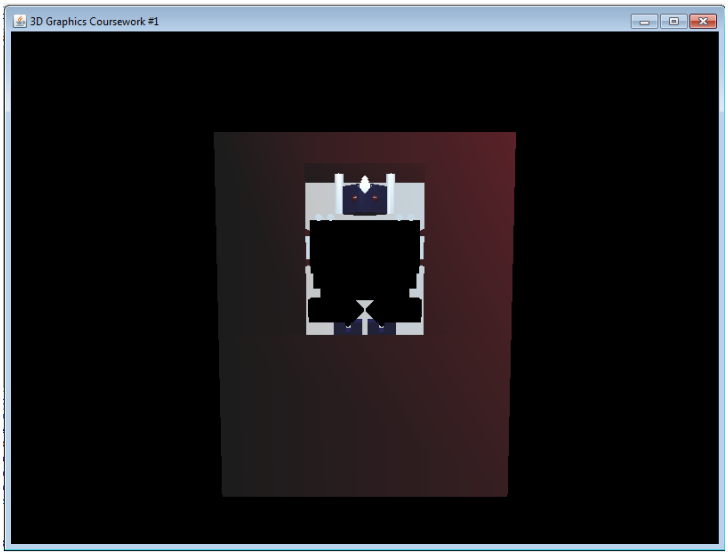


Toggling Lights

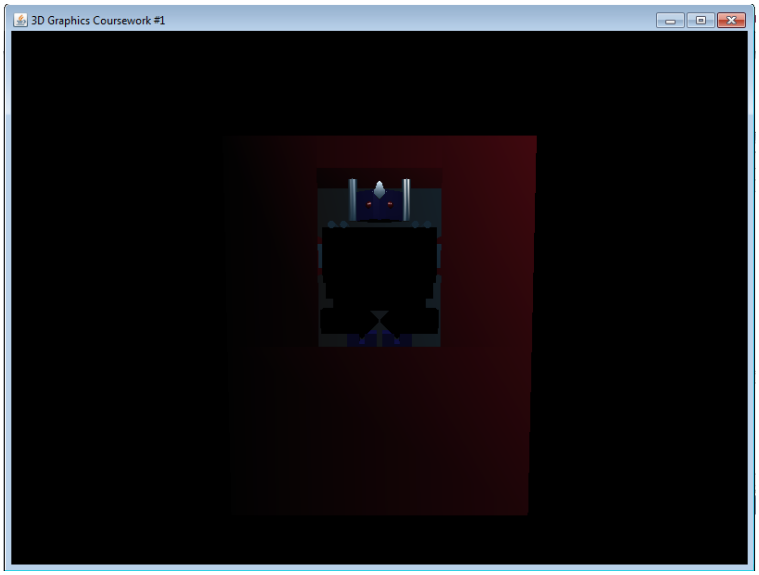
Lights On

The screenshot to the right shows the scene with the lights turned on.

With lights on it's much brighter and easier to see the various objects.



Lights Off



The screenshot to the left shows the robot scene with the lights turned off.

As you can see there is a significant difference between the lights on and the lights off and it's much harder to see with the lights off.

Animation

Battle Orb Animation

The screenshot to the left demonstrates the battle orb animation which has two golden orbs rotate around the robot like a shield.

They look like they are 2D in this image but when actually playing the animation they have a 3D appearance.

Appendix B - 3D Graphics Robot Source Code

```
package gates_toby;


import java.awt.event.*;
import javax.swing.*;
import javax.media.opengl.*;
import com.sun.opengl.util.*;
import javax.media.opengl.glu.GLU;


public class Robot extends GLJPanel implements GLEventListener {


    private GLU glu;
    private GLUT glut;
    private boolean reset = true;
    private float[] viewer = new float[3];
    private float viewerAngle;
    private boolean perspective = true;
    //variables used for animation
    private float x1, x2, x3, x4;
    //rotation variables and animator used for battle orb animation
    private float rotation = 0.0f;
    private float frameRotation = 3.0f;
    private Animator rotateAnimate;
    private final float DEG2RAD = (float) Math.PI / 180.0f;
    //boolean used to determine wether lights are on or off
    private boolean sceneLights = true;


    public Robot() {
        super();
        addGLEventListener(this);
        addKeyListener(new KeyResponder());
```

```
        addMouseListener(new MouseResponder());
        glu = new GLU();
        glut = new GLUT();
        //animator instantiation
        rotateAnimate = new Animator(this);
    }

    //resets the lights, used when R is pressed by the user
    private void resetLights(GL gl) {
        float ambientGlobal[] = {0.5f, 0.5f, 0.5f, 1.0f};
        gl.glLightModelfv(GL.GL_LIGHT_MODEL_AMBIENT, ambientGlobal, 0);
        {
            float ambient[] = {0.2f, 0.2f, 0.2f, 1.0f};
            float diffuse[] = {0.5f, 0.5f, 0.5f, 1.0f};
            float specular[] = {0.2f, 0.2f, 0.2f, 1.0f};
            float position[] = {2.0f, 2.0f, -2.0f, 0.0f};

            gl.glLightfv(GL.GL_LIGHT0, GL.GL_AMBIENT, ambient, 0);
            gl.glLightfv(GL.GL_LIGHT0, GL.GL_DIFFUSE, diffuse, 0);
            gl.glLightfv(GL.GL_LIGHT0, GL.GL_SPECULAR, specular, 0);
            gl.glLightfv(GL.GL_LIGHT0, GL.GL_POSITION, position, 0);

            gl.glEnable(GL.GL_LIGHT0);
        }
    }

    public void init(GLAutoDrawable drawable) {
        GL gl = drawable.getGL();
        //lights are initialized before the program starts running.
        float ambient[] = {0.0f, 0.0f, 0.0f, 1.0f};
        float diffuse[] = {1.0f, 1.0f, 1.0f, 1.0f};
        float specular[] = {1.0f, 1.0f, 1.0f, 1.0f};
```

```
        float position[] = {0.0f, 3.0f, 3.0f, 0.0f};

        float[] ambientGlobal = {0.2f, 0.2f, 0.2f, 1.0f};

        gl.glLightfv(GL.GL_LIGHT0, GL.GL_AMBIENT, ambient, 0);
        gl.glLightfv(GL.GL_LIGHT0, GL.GL_DIFFUSE, diffuse, 0);
        gl.glLightfv(GL.GL_LIGHT0, GL.GL_SPECULAR, specular, 0);
        gl.glLightfv(GL.GL_LIGHT0, GL.GL_POSITION, position, 0);

        gl.glLightModelfv(GL.GL_LIGHT_MODEL_AMBIENT, ambientGlobal, 0);

        gl.glEnable(GL.GL_LIGHTING);
        gl.glEnable(GL.GL_LIGHT0);

        gl.glEnable(GL.GL_NORMALIZE);
        gl.glDepthFunc(GL.GL_LESS);
        gl.glEnable(GL.GL_DEPTH_TEST);



        //when the program starts, the animator is stopped if running.
        if (rotateAnimate.isAnimating()) {
            rotateAnimate.stop();
        }

    }

    public void display(GLAutoDrawable drawable) {
        requestFocusInWindow();

        GL gl = drawable.getGL();

        gl.glClear(GL.GL_COLOR_BUFFER_BIT | GL.GL_DEPTH_BUFFER_BIT);
```

```
gl.glLoadIdentity();
if (reset) {
    //if reset is true then the scene resets to the default view and the
    //lights reset.
    resetLights(gl);
    resetViewer();
    reset = false;
}


if (!sceneLights) {
    //calls the lightsOff method if sceneLights is false
    lightsOff(gl);
}
if (sceneLights) {
    //calls the lightsOn method if sceneLights is true
    lightsOn(gl);
}


glu.gluLookAt(viewer[0], viewer[1], viewer[2], 0.0, 0.0, 0.0, 0.0, 1.0, 0.0);


{
    gl.glPushMatrix();


    /*
     **ROBOT REFERENCE IMAGE v**
    http://theenamelista.files.wordpress.com/2011/07/optimus-prime.jpg
     */


    //float arrays used for rendering the robot.
```

```
float[] ambientRed = {0.1f, 0.1f, 0.1f, 0.1f};
float[] diffuseRed = {1.0f, 0.1f, 0.2f, 0.1f};
float[] specularRed = {1.0f, 1.0f, 1.0f, 1.0f};

float[] ambientGrey = {0.7f, 0.7f, 0.7f, 0.1f};
float[] diffuseGrey = {0.1f, 0.5f, 0.8f, 0.1f};
float[] specularGrey = {1.0f, 1.0f, 1.0f, 1.0f};

float[] ambientBlue = {0.1f, 0.1f, 0.1f, 0.1f};
float[] diffuseBlue = {0.2f, 0.2f, 1.0f, 0.1f};
float[] specularBlue = {1.0f, 1.0f, 1.0f, 1.0f};

float[] ambientBlack = {0.0f, 0.0f, 0.0f, 0.0f};
float[] diffuseBlack = {0.0f, 0.0f, 0.0f, 0.1f};
float[] specularBlack = {0.2f, 0.2f, 0.3f, 0.4f};

float[] ambientGold = {0.5f, 0.4f, 0.1f, 0.1f};
float[] diffuseGold = {0.8f, 0.6f, 0.2f, 0.1f};
float[] specularGold = {0.5f, 0.5f, 0.5f, 0.5f};

//two different values for intensity of shine
float shineLow = 15.0f;
float shineHigh = 100.0f;

// light #1 for the scene
float ambient[] = {0.2f, 0.2f, 0.2f, 1.0f};
float diffuse[] = {0.1f, 0.1f, 0.1f, 1.0f};
float specular[] = {1.0f, 1.0f, 1.0f, 1.0f};
float position[] = {-1.50f, 2.0f, 2.0f, 1.0f};

gl.glLightfv(GL.GL_LIGHT0, GL.GL_AMBIENT, ambient, 0);
gl.glLightfv(GL.GL_LIGHT0, GL.GL_DIFFUSE, diffuse, 0);
```

```
gl.glLightfv(GL.GL_LIGHT0, GL.GL_SPECULAR, specular, 0);
gl.glLightfv(GL.GL_LIGHT0, GL.GL_POSITION, position, 0);

gl.glEnable(GL.GL_LIGHTING);
gl.glEnable(GL.GL_LIGHT0);

// light #2 for the scene
float ambience[] = {0.2f, 0.2f, 0.2f, 1.0f};
float diffusion[] = {0.1f, 0.1f, 0.1f, 1.0f};
float speculars[] = {1.0f, 1.0f, 1.0f, 1.0f};
float positioning[] = {1.50f, 2.0f, 1.2f, 1.0f};

gl.glLightfv(GL.GL_LIGHT1, GL.GL_AMBIENT, ambience, 0);
gl.glLightfv(GL.GL_LIGHT1, GL.GL_DIFFUSE, diffusion, 0);
gl.glLightfv(GL.GL_LIGHT1, GL.GL_SPECULAR, speculars, 0);
gl.glLightfv(GL.GL_LIGHT1, GL.GL_POSITION, positioning, 0);

gl.glEnable(GL.GL_LIGHTING);
gl.glEnable(GL.GL_LIGHT1);

// light #3 for the scene
float ambienist[] = {0.2f, 0.2f, 0.2f, 1.0f};
float diffusionist[] = {0.1f, 0.1f, 0.1f, 1.0f};
float speculist[] = {1.0f, 1.0f, 1.0f, 1.0f};
float positionist[] = {0.0f, 2.0f, 1.2f, 1.0f};

gl.glLightfv(GL.GL_LIGHT2, GL.GL_AMBIENT, ambienist, 0);
gl.glLightfv(GL.GL_LIGHT2, GL.GL_DIFFUSE, diffusionist, 0);
gl.glLightfv(GL.GL_LIGHT2, GL.GL_SPECULAR, speculist, 0);
gl.glLightfv(GL.GL_LIGHT2, GL.GL_POSITION, positionist, 0);

gl.glEnable(GL.GL_LIGHTING);
```

```
gl.glEnable(GL.GL_LIGHT2);



// upper arms
gl.glPushMatrix();
gl.glTranslatef(-1.36f, 0.72f, 0.0f);
gl.glRotatef(90.0f, 1.0f, 0.0f, 0.0f);
gl.glMaterialfv(GL.GL_FRONT, GL.GL_AMBIENT, ambientRed, 0);
gl.glMaterialfv(GL.GL_FRONT, GL.GL_DIFFUSE, diffuseRed, 0);
gl.glMaterialfv(GL.GL_FRONT, GL.GL_SPECULAR, specularRed, 0);
gl.glMaterialf(GL.GL_FRONT, GL.GL_SHININESS, shineLow);
glut.glutSolidCube(0.65f);
gl.glPopMatrix();

gl.glPushMatrix();
gl.glTranslatef(1.36f, 0.72f, 0.0f);
gl.glRotatef(90.0f, 1.0f, 0.0f, 0.0f);
gl.glMaterialfv(GL.GL_FRONT, GL.GL_AMBIENT, ambientRed, 0);
gl.glMaterialfv(GL.GL_FRONT, GL.GL_DIFFUSE, diffuseRed, 0);
gl.glMaterialfv(GL.GL_FRONT, GL.GL_SPECULAR, specularRed, 0);
gl.glMaterialf(GL.GL_FRONT, GL.GL_SHININESS, shineHigh);
glut.glutSolidCube(0.65f);
gl.glPopMatrix();

gl.glPushMatrix();
gl.glTranslatef(-1.36f, 0.30f, 0.0f);
gl.glRotatef(90.0f, 1.0f, 0.0f, 0.0f);
gl.glMaterialfv(GL.GL_FRONT, GL.GL_AMBIENT, ambientGrey, 0);
gl.glMaterialfv(GL.GL_FRONT, GL.GL_DIFFUSE, diffuseGrey, 0);
gl.glMaterialfv(GL.GL_FRONT, GL.GL_SPECULAR, specularGrey, 0);
gl.glMaterialf(GL.GL_FRONT, GL.GL_SHININESS, shineHigh);
glut.glutSolidCube(0.5f);
```

```
gl.glPopMatrix();


gl.glPushMatrix();
gl.glTranslatef(1.36f, 0.30f, 0.0f);
gl.glRotatef(90.0f, 1.0f, 0.0f, 0.0f);
gl.glMaterialfv(GL.GL_FRONT, GL.GL_AMBIENT, ambientGrey, 0);
gl.glMaterialfv(GL.GL_FRONT, GL.GL_DIFFUSE, diffuseGrey, 0);
gl.glMaterialfv(GL.GL_FRONT, GL.GL_SPECULAR, specularGrey, 0);
gl.glMaterialf(GL.GL_FRONT, GL.GL_SHININESS, shineLow);
glut.glutSolidCube(0.5f);
gl.glPopMatrix();



// upper arm attachments
gl.glPushMatrix();
gl.glTranslatef(-1.74f, 1.25f, 0.0f);
gl.glRotatef(90.0f, 1.0f, 0.0f, 0.0f);
gl.glMaterialfv(GL.GL_FRONT, GL.GL_AMBIENT, ambientGrey, 0);
gl.glMaterialfv(GL.GL_FRONT, GL.GL_DIFFUSE, diffuseGrey, 0);
gl.glMaterialfv(GL.GL_FRONT, GL.GL_SPECULAR, specularGrey, 0);
gl.glMaterialf(GL.GL_FRONT, GL.GL_SHININESS, shineHigh);
//radius, height, slices, stacks
glut.glutSolidCylinder(0.13, 0.8, 16, 1);
gl.glPopMatrix();


gl.glPushMatrix();
gl.glTranslatef(-1.74f, 1.60f, 0.0f);
gl.glRotatef(90.0f, 1.0f, 0.0f, 0.0f);
gl.glMaterialfv(GL.GL_FRONT, GL.GL_AMBIENT, ambientGrey, 0);
gl.glMaterialfv(GL.GL_FRONT, GL.GL_DIFFUSE, diffuseGrey, 0);
gl.glMaterialfv(GL.GL_FRONT, GL.GL_SPECULAR, specularGrey, 0);
gl.glMaterialf(GL.GL_FRONT, GL.GL_SHININESS, shineHigh);
```

```
//radius, height, slices, stacks
glut.glutSolidCylinder(0.1, 0.8, 16, 1);
gl.glPopMatrix();


gl.glPushMatrix();
gl.glTranslatef(1.74f, 1.25f, 0.0f);
gl.glRotatef(90.0f, 1.0f, 0.0f, 0.0f);
gl.glMaterialfv(GL.GL_FRONT, GL.GL_AMBIENT, ambientGrey, 0);
gl.glMaterialfv(GL.GL_FRONT, GL.GL_DIFFUSE, diffuseGrey, 0);
gl.glMaterialfv(GL.GL_FRONT, GL.GL_SPECULAR, specularGrey, 0);
gl.glMaterialf(GL.GL_FRONT, GL.GL_SHININESS, shineHigh);
glut.glutSolidCylinder(0.13, 0.8, 16, 1);
gl.glPopMatrix();


gl.glPushMatrix();
gl.glTranslatef(1.74f, 1.60f, 0.0f);
gl.glRotatef(90.0f, 1.0f, 0.0f, 0.0f);
gl.glMaterialfv(GL.GL_FRONT, GL.GL_AMBIENT, ambientGrey, 0);
gl.glMaterialfv(GL.GL_FRONT, GL.GL_DIFFUSE, diffuseGrey, 0);
gl.glMaterialfv(GL.GL_FRONT, GL.GL_SPECULAR, specularGrey, 0);
gl.glMaterialf(GL.GL_FRONT, GL.GL_SHININESS, shineHigh);
glut.glutSolidCylinder(0.1, 0.8, 16, 1);
gl.glPopMatrix();



//upper arm-body connections
//left
gl.glPushMatrix();
gl.glTranslatef(-1.10f, 0.72f, 0.0f);
gl.glRotatef(90.0f, 1.0f, 0.0f, 0.0f);
glut.glutSolidCube(0.4f);
gl.glPopMatrix();
```

```
//right
gl.glPushMatrix();
gl.glTranslatef(1.10f, 0.72f, 0.0f);
gl.glRotatef(90.0f, 1.0f, 0.0f, 0.0f);
glut.glutSolidCube(0.4f);
gl.glPopMatrix();



//upper-lower arm connections
//left
gl.glPushMatrix();
gl.glTranslatef(-1.515f, 0.0f, 0.0f);
gl.glRotatef(90.0f, 1.0f, 0.0f, 0.0f);
glut.glutSolidCube(0.2f);
gl.glPopMatrix();

gl.glPushMatrix();
gl.glTranslatef(-1.215f, 0.0f, 0.0f);
gl.glRotatef(90.0f, 1.0f, 0.0f, 0.0f);
glut.glutSolidCube(0.2f);
gl.glPopMatrix();

//right
gl.glPushMatrix();
gl.glTranslatef(1.515f, 0.0f, 0.0f);
gl.glRotatef(90.0f, 1.0f, 0.0f, 0.0f);
glut.glutSolidCube(0.2f);
//radius, height slices, stacks.
//glut.glutSolidCylinder(10, 20, 1, 1);
gl.glPopMatrix();
```

```
gl.glPushMatrix();
gl.glTranslatef(1.215f, 0.0f, 0.0f);
gl.glRotatef(90.0f, 1.0f, 0.0f, 0.0f);
glut.glutSolidCube(0.2f);
//radius, height slices, stacks.
//glut.glutSolidCylinder(10, 20, 1, 1);
gl.glPopMatrix();



// lower arms
gl.glPushMatrix();
gl.glTranslatef(-1.52f, -0.2f, 0.15f);
gl.glRotatef(90.0f, 1.0f, 0.0f, 0.0f);
gl.glMaterialfv(GL.GL_FRONT, GL.GL_AMBIENT, ambientRed, 0);
gl.glMaterialfv(GL.GL_FRONT, GL.GL_DIFFUSE, diffuseRed, 0);
gl.glMaterialfv(GL.GL_FRONT, GL.GL_SPECULAR, specularRed, 0);
gl.glMaterialf(GL.GL_FRONT, GL.GL_SHININESS, shineLow);
glut.glutSolidCube(0.2f);
gl.glPopMatrix();

gl.glPushMatrix();
gl.glTranslatef(-1.52f, -0.2f, 0.0f);
gl.glRotatef(90.0f, 1.0f, 0.0f, 0.0f);
gl.glMaterialfv(GL.GL_FRONT, GL.GL_AMBIENT, ambientRed, 0);
gl.glMaterialfv(GL.GL_FRONT, GL.GL_DIFFUSE, diffuseRed, 0);
gl.glMaterialfv(GL.GL_FRONT, GL.GL_SPECULAR, specularRed, 0);
gl.glMaterialf(GL.GL_FRONT, GL.GL_SHININESS, shineLow);
glut.glutSolidCube(0.2f);
gl.glPopMatrix();

gl.glPushMatrix();
gl.glTranslatef(-1.52f, -0.2f, -0.15f);
```

```
gl.glRotatef(90.0f, 1.0f, 0.0f, 0.0f);
gl.glMaterialfv(GL.GL_FRONT, GL.GL_AMBIENT, ambientRed, 0);
gl.glMaterialfv(GL.GL_FRONT, GL.GL_DIFFUSE, diffuseRed, 0);
gl.glMaterialfv(GL.GL_FRONT, GL.GL_SPECULAR, specularRed, 0);
gl.glMaterialf(GL.GL_FRONT, GL.GL_SHININESS, shineLow);
glut.glutSolidCube(0.2f);
gl.glPopMatrix();

gl.glPushMatrix();
gl.glTranslatef(-1.21f, -0.2f, 0.15f);
gl.glRotatef(90.0f, 1.0f, 0.0f, 0.0f);
gl.glMaterialfv(GL.GL_FRONT, GL.GL_AMBIENT, ambientRed, 0);
gl.glMaterialfv(GL.GL_FRONT, GL.GL_DIFFUSE, diffuseRed, 0);
gl.glMaterialfv(GL.GL_FRONT, GL.GL_SPECULAR, specularRed, 0);
gl.glMaterialf(GL.GL_FRONT, GL.GL_SHININESS, shineLow);
glut.glutSolidCube(0.2f);
gl.glPopMatrix();

gl.glPushMatrix();
gl.glTranslatef(-1.21f, -0.2f, 0.0f);
gl.glRotatef(90.0f, 1.0f, 0.0f, 0.0f);
gl.glMaterialfv(GL.GL_FRONT, GL.GL_AMBIENT, ambientRed, 0);
gl.glMaterialfv(GL.GL_FRONT, GL.GL_DIFFUSE, diffuseRed, 0);
gl.glMaterialfv(GL.GL_FRONT, GL.GL_SPECULAR, specularRed, 0);
gl.glMaterialf(GL.GL_FRONT, GL.GL_SHININESS, shineLow);
glut.glutSolidCube(0.2f);
gl.glPopMatrix();

gl.glPushMatrix();
gl.glTranslatef(-1.21f, -0.2f, -0.15f);
gl.glRotatef(90.0f, 1.0f, 0.0f, 0.0f);
gl.glMaterialfv(GL.GL_FRONT, GL.GL_AMBIENT, ambientRed, 0);
```

```
gl.glMaterialfv(GL.GL_FRONT, GL.GL_DIFFUSE, diffuseRed, 0);
gl.glMaterialfv(GL.GL_FRONT, GL.GL_SPECULAR, specularRed, 0);
gl.glMaterialf(GL.GL_FRONT, GL.GL_SHININESS, shineLow);
glut.glutSolidCube(0.2f);
gl.glPopMatrix();

gl.glPushMatrix();
gl.glTranslatef(1.52f, -0.2f, 0.15f);
gl.glRotatef(90.0f, 1.0f, 0.0f, 0.0f);
gl.glMaterialfv(GL.GL_FRONT, GL.GL_AMBIENT, ambientRed, 0);
gl.glMaterialfv(GL.GL_FRONT, GL.GL_DIFFUSE, diffuseRed, 0);
gl.glMaterialfv(GL.GL_FRONT, GL.GL_SPECULAR, specularRed, 0);
gl.glMaterialf(GL.GL_FRONT, GL.GL_SHININESS, shineLow);
glut.glutSolidCube(0.2f);
gl.glPopMatrix();

gl.glPushMatrix();
gl.glTranslatef(1.52f, -0.2f, 0.0f);
gl.glRotatef(90.0f, 1.0f, 0.0f, 0.0f);
gl.glMaterialfv(GL.GL_FRONT, GL.GL_AMBIENT, ambientRed, 0);
gl.glMaterialfv(GL.GL_FRONT, GL.GL_DIFFUSE, diffuseRed, 0);
gl.glMaterialfv(GL.GL_FRONT, GL.GL_SPECULAR, specularRed, 0);
gl.glMaterialf(GL.GL_FRONT, GL.GL_SHININESS, shineLow);
glut.glutSolidCube(0.2f);
gl.glPopMatrix();

gl.glPushMatrix();
gl.glTranslatef(1.52f, -0.2f, -0.15f);
gl.glRotatef(90.0f, 1.0f, 0.0f, 0.0f);
gl.glMaterialfv(GL.GL_FRONT, GL.GL_AMBIENT, ambientRed, 0);
gl.glMaterialfv(GL.GL_FRONT, GL.GL_DIFFUSE, diffuseRed, 0);
gl.glMaterialfv(GL.GL_FRONT, GL.GL_SPECULAR, specularRed, 0);
```

```
gl.glMaterialf(GL.GL_FRONT, GL.GL_SHININESS, shineLow);
glut.glutSolidCube(0.2f);
gl.glPopMatrix();

gl.glPushMatrix();
gl.glTranslatef(1.21f, -0.2f, 0.15f);
gl.glRotatef(90.0f, 1.0f, 0.0f, 0.0f);
gl.glMaterialfv(GL.GL_FRONT, GL.GL_AMBIENT, ambientRed, 0);
gl.glMaterialfv(GL.GL_FRONT, GL.GL_DIFFUSE, diffuseRed, 0);
gl.glMaterialfv(GL.GL_FRONT, GL.GL_SPECULAR, specularRed, 0);
gl.glMaterialf(GL.GL_FRONT, GL.GL_SHININESS, shineLow);
glut.glutSolidCube(0.2f);
gl.glPopMatrix();

gl.glPushMatrix();
gl.glTranslatef(1.21f, -0.2f, 0.0f);
gl.glRotatef(90.0f, 1.0f, 0.0f, 0.0f);
gl.glMaterialfv(GL.GL_FRONT, GL.GL_AMBIENT, ambientRed, 0);
gl.glMaterialfv(GL.GL_FRONT, GL.GL_DIFFUSE, diffuseRed, 0);
gl.glMaterialfv(GL.GL_FRONT, GL.GL_SPECULAR, specularRed, 0);
gl.glMaterialf(GL.GL_FRONT, GL.GL_SHININESS, shineLow);
glut.glutSolidCube(0.2f);
gl.glPopMatrix();

gl.glPushMatrix();
gl.glTranslatef(1.21f, -0.2f, -0.15f);
gl.glRotatef(90.0f, 1.0f, 0.0f, 0.0f);
gl.glMaterialfv(GL.GL_FRONT, GL.GL_AMBIENT, ambientRed, 0);
gl.glMaterialfv(GL.GL_FRONT, GL.GL_DIFFUSE, diffuseRed, 0);
gl.glMaterialfv(GL.GL_FRONT, GL.GL_SPECULAR, specularRed, 0);
gl.glMaterialf(GL.GL_FRONT, GL.GL_SHININESS, shineLow);
glut.glutSolidCube(0.2f);
```

```
gl.glPopMatrix();


gl.glPushMatrix();
gl.glTranslatef(-1.36f, -0.75f, 0.0f);
gl.glRotatef(90.0f, 1.0f, 0.0f, 0.0f);
gl.glMaterialfv(GL.GL_FRONT, GL.GL_AMBIENT, ambientRed, 0);
gl.glMaterialfv(GL.GL_FRONT, GL.GL_DIFFUSE, diffuseRed, 0);
gl.glMaterialfv(GL.GL_FRONT, GL.GL_SPECULAR, specularRed, 0);
gl.glMaterialf(GL.GL_FRONT, GL.GL_SHININESS, shineLow);
glut.glutSolidCube(0.5f);
gl.glPopMatrix();


gl.glPushMatrix();
gl.glTranslatef(-1.36f, -0.5f, 0.0f);
gl.glRotatef(90.0f, 1.0f, 0.0f, 0.0f);
gl.glMaterialfv(GL.GL_FRONT, GL.GL_AMBIENT, ambientRed, 0);
gl.glMaterialfv(GL.GL_FRONT, GL.GL_DIFFUSE, diffuseRed, 0);
gl.glMaterialfv(GL.GL_FRONT, GL.GL_SPECULAR, specularRed, 0);
gl.glMaterialf(GL.GL_FRONT, GL.GL_SHININESS, shineLow);
glut.glutSolidCube(0.5f);
gl.glPopMatrix();


gl.glPushMatrix();
gl.glTranslatef(1.36f, -0.75f, 0.0f);
gl.glRotatef(90.0f, 1.0f, 0.0f, 0.0f);
gl.glMaterialfv(GL.GL_FRONT, GL.GL_AMBIENT, ambientRed, 0);
gl.glMaterialfv(GL.GL_FRONT, GL.GL_DIFFUSE, diffuseRed, 0);
gl.glMaterialfv(GL.GL_FRONT, GL.GL_SPECULAR, specularRed, 0);
gl.glMaterialf(GL.GL_FRONT, GL.GL_SHININESS, shineLow);
glut.glutSolidCube(0.5f);
gl.glPopMatrix();
```

```
gl.glPushMatrix();
gl.glTranslatef(1.36f, -0.5f, 0.0f);
gl.glRotatef(90.0f, 1.0f, 0.0f, 0.0f);
gl.glMaterialfv(GL.GL_FRONT, GL.GL_AMBIENT, ambientRed, 0);
gl.glMaterialfv(GL.GL_FRONT, GL.GL_DIFFUSE, diffuseRed, 0);
gl.glMaterialfv(GL.GL_FRONT, GL.GL_SPECULAR, specularRed, 0);
gl.glMaterialf(GL.GL_FRONT, GL.GL_SHININESS, shineLow);
glut.glutSolidCube(0.5f);
gl.glPopMatrix();



// hands
gl.glPushMatrix();
gl.glTranslatef(-1.36f, -1.1f, 0.0f);
gl.glRotatef(90.0f, 1.0f, 0.0f, 0.0f);
gl.glMaterialfv(GL.GL_FRONT, GL.GL_AMBIENT, ambientBlue, 0);
gl.glMaterialfv(GL.GL_FRONT, GL.GL_DIFFUSE, diffuseBlue, 0);
gl.glMaterialfv(GL.GL_FRONT, GL.GL_SPECULAR, specularBlue, 0);
gl.glMaterialf(GL.GL_FRONT, GL.GL_SHININESS, shineHigh);
glut.glutSolidCube(0.4f);
gl.glPopMatrix();

gl.glPushMatrix();
gl.glTranslatef(1.36f, -1.1f, 0.0f);
gl.glRotatef(90.0f, 1.0f, 0.0f, 0.0f);
gl.glMaterialfv(GL.GL_FRONT, GL.GL_AMBIENT, ambientBlue, 0);
gl.glMaterialfv(GL.GL_FRONT, GL.GL_DIFFUSE, diffuseBlue, 0);
gl.glMaterialfv(GL.GL_FRONT, GL.GL_SPECULAR, specularBlue, 0);
gl.glMaterialf(GL.GL_FRONT, GL.GL_SHININESS, shineHigh);
glut.glutSolidCube(0.4f);
gl.glPopMatrix();
```

```
// left hand - fingers
// first finger
gl.glPushMatrix();
gl.glTranslatef(-1.50f, -1.34f, 0.15f);
gl.glRotatef(90.0f, 1.0f, 0.0f, 0.0f);
gl.glMaterialfv(GL.GL_FRONT, GL.GL_AMBIENT, ambientBlue, 0);
gl.glMaterialfv(GL.GL_FRONT, GL.GL_DIFFUSE, diffuseBlue, 0);
gl.glMaterialfv(GL.GL_FRONT, GL.GL_SPECULAR, specularBlue, 0);
gl.glMaterialf(GL.GL_FRONT, GL.GL_SHININESS, shineHigh);
glut.glutSolidCube(0.1f);
gl.glPopMatrix();

gl.glPushMatrix();
gl.glTranslatef(-1.49f, -1.38f, 0.15f);
//angle, x, y, z
gl.glRotatef(105.0f, 0.0f, 0.0f, 1.0f);
gl.glMaterialfv(GL.GL_FRONT, GL.GL_AMBIENT, ambientBlue, 0);
gl.glMaterialfv(GL.GL_FRONT, GL.GL_DIFFUSE, diffuseBlue, 0);
gl.glMaterialfv(GL.GL_FRONT, GL.GL_SPECULAR, specularBlue, 0);
gl.glMaterialf(GL.GL_FRONT, GL.GL_SHININESS, shineHigh);
glut.glutSolidCube(0.1f);
gl.glPopMatrix();

gl.glPushMatrix();
gl.glTranslatef(-1.48f, -1.42f, 0.15f);
gl.glRotatef(121.0f, 0.0f, 0.0f, 1.0f);
gl.glMaterialfv(GL.GL_FRONT, GL.GL_AMBIENT, ambientBlue, 0);
gl.glMaterialfv(GL.GL_FRONT, GL.GL_DIFFUSE, diffuseBlue, 0);
gl.glMaterialfv(GL.GL_FRONT, GL.GL_SPECULAR, specularBlue, 0);
gl.glMaterialf(GL.GL_FRONT, GL.GL_SHININESS, shineHigh);
glut.glutSolidCube(0.1f);
```

```
gl.glPopMatrix();


gl.glPushMatrix();
gl.glTranslatef(-1.47f, -1.46f, 0.15f);
gl.glRotatef(135.0f, 0.0f, 0.0f, 1.0f);
gl.glMaterialfv(GL.GL_FRONT, GL.GL_AMBIENT, ambientBlue, 0);
gl.glMaterialfv(GL.GL_FRONT, GL.GL_DIFFUSE, diffuseBlue, 0);
gl.glMaterialfv(GL.GL_FRONT, GL.GL_SPECULAR, specularBlue, 0);
gl.glMaterialf(GL.GL_FRONT, GL.GL_SHININESS, shineHigh);
glut.glutSolidCube(0.1f);
gl.glPopMatrix();


//second finger
gl.glPushMatrix();
gl.glTranslatef(-1.52f, -1.34f, 0.0f);
gl.glRotatef(90.0f, 1.0f, 0.0f, 0.0f);
gl.glMaterialfv(GL.GL_FRONT, GL.GL_AMBIENT, ambientBlue, 0);
gl.glMaterialfv(GL.GL_FRONT, GL.GL_DIFFUSE, diffuseBlue, 0);
gl.glMaterialfv(GL.GL_FRONT, GL.GL_SPECULAR, specularBlue, 0);
gl.glMaterialf(GL.GL_FRONT, GL.GL_SHININESS, shineHigh);
glut.glutSolidCube(0.1f);
gl.glPopMatrix();


gl.glPushMatrix();
gl.glTranslatef(-1.51f, -1.38f, 0.0f);
//angle, x, y, z
gl.glRotatef(105.0f, 0.0f, 0.0f, 1.0f);
gl.glMaterialfv(GL.GL_FRONT, GL.GL_AMBIENT, ambientBlue, 0);
gl.glMaterialfv(GL.GL_FRONT, GL.GL_DIFFUSE, diffuseBlue, 0);
gl.glMaterialfv(GL.GL_FRONT, GL.GL_SPECULAR, specularBlue, 0);
gl.glMaterialf(GL.GL_FRONT, GL.GL_SHININESS, shineHigh);
glut.glutSolidCube(0.1f);
```

```
gl.glPopMatrix();


gl.glPushMatrix();
gl.glTranslatef(-1.50f, -1.42f, 0.0f);
gl.glRotatef(121.0f, 0.0f, 0.0f, 1.0f);
gl.glMaterialfv(GL.GL_FRONT, GL.GL_AMBIENT, ambientGrey, 0);
gl.glMaterialfv(GL.GL_FRONT, GL.GL_DIFFUSE, diffuseGrey, 0);
gl.glMaterialfv(GL.GL_FRONT, GL.GL_SPECULAR, specularGrey, 0);
glut.glutSolidCube(0.1f);
gl.glPopMatrix();


gl.glPushMatrix();
gl.glTranslatef(-1.49f, -1.46f, 0.0f);
gl.glRotatef(135.0f, 0.0f, 0.0f, 1.0f);
gl.glMaterialfv(GL.GL_FRONT, GL.GL_AMBIENT, ambientBlue, 0);
gl.glMaterialfv(GL.GL_FRONT, GL.GL_DIFFUSE, diffuseBlue, 0);
gl.glMaterialfv(GL.GL_FRONT, GL.GL_SPECULAR, specularBlue, 0);
gl.glMaterialf(GL.GL_FRONT, GL.GL_SHININESS, shineHigh);
glut.glutSolidCube(0.1f);
gl.glPopMatrix();


//third finger
gl.glPushMatrix();
gl.glTranslatef(-1.51f, -1.34f, -0.15f);
gl.glRotatef(90.0f, 1.0f, 0.0f, 0.0f);
gl.glMaterialfv(GL.GL_FRONT, GL.GL_AMBIENT, ambientBlue, 0);
gl.glMaterialfv(GL.GL_FRONT, GL.GL_DIFFUSE, diffuseBlue, 0);
gl.glMaterialfv(GL.GL_FRONT, GL.GL_SPECULAR, specularBlue, 0);
gl.glMaterialf(GL.GL_FRONT, GL.GL_SHININESS, shineHigh);
glut.glutSolidCube(0.1f);
gl.glPopMatrix();
```

```
gl.glPushMatrix();
gl.glTranslatef(-1.50f, -1.38f, -0.15f);
//angle, x, y, z
gl.glRotatef(105.0f, 0.0f, 0.0f, 1.0f);
gl.glMaterialfv(GL.GL_FRONT, GL.GL_AMBIENT, ambientBlue, 0);
gl.glMaterialfv(GL.GL_FRONT, GL.GL_DIFFUSE, diffuseBlue, 0);
gl.glMaterialfv(GL.GL_FRONT, GL.GL_SPECULAR, specularBlue, 0);
gl.glMaterialf(GL.GL_FRONT, GL.GL_SHININESS, shineHigh);
glut.glutSolidCube(0.1f);
gl.glPopMatrix();

gl.glPushMatrix();
gl.glTranslatef(-1.49f, -1.42f, -0.15f);
gl.glRotatef(121.0f, 0.0f, 0.0f, 1.0f);
gl.glMaterialfv(GL.GL_FRONT, GL.GL_AMBIENT, ambientBlue, 0);
gl.glMaterialfv(GL.GL_FRONT, GL.GL_DIFFUSE, diffuseBlue, 0);
gl.glMaterialfv(GL.GL_FRONT, GL.GL_SPECULAR, specularBlue, 0);
gl.glMaterialf(GL.GL_FRONT, GL.GL_SHININESS, shineHigh);
glut.glutSolidCube(0.1f);
gl.glPopMatrix();

gl.glPushMatrix();
gl.glTranslatef(-1.48f, -1.46f, -0.15f);
gl.glRotatef(135.0f, 0.0f, 0.0f, 1.0f);
gl.glMaterialfv(GL.GL_FRONT, GL.GL_AMBIENT, ambientBlue, 0);
gl.glMaterialfv(GL.GL_FRONT, GL.GL_DIFFUSE, diffuseBlue, 0);
gl.glMaterialfv(GL.GL_FRONT, GL.GL_SPECULAR, specularBlue, 0);
gl.glMaterialf(GL.GL_FRONT, GL.GL_SHININESS, shineHigh);
glut.glutSolidCube(0.1f);
gl.glPopMatrix();

//left hand - thumb
```

```
gl.glPushMatrix();
gl.glTranslatef(-1.26f, -1.34f, 0.0f);
gl.glRotatef(90.0f, 1.0f, 0.0f, 0.0f);
gl.glMaterialfv(GL.GL_FRONT, GL.GL_AMBIENT, ambientBlue, 0);
gl.glMaterialfv(GL.GL_FRONT, GL.GL_DIFFUSE, diffuseBlue, 0);
gl.glMaterialfv(GL.GL_FRONT, GL.GL_SPECULAR, specularBlue, 0);
gl.glMaterialf(GL.GL_FRONT, GL.GL_SHININESS, shineHigh);
glut.glutSolidCube(0.1f);
gl.glPopMatrix();

gl.glPushMatrix();
gl.glTranslatef(-1.27f, -1.38f, 0.0f);
//angle, x, y, z
gl.glRotatef(105.0f, 0.0f, 0.0f, 1.0f);
gl.glMaterialfv(GL.GL_FRONT, GL.GL_AMBIENT, ambientBlue, 0);
gl.glMaterialfv(GL.GL_FRONT, GL.GL_DIFFUSE, diffuseBlue, 0);
gl.glMaterialfv(GL.GL_FRONT, GL.GL_SPECULAR, specularBlue, 0);
gl.glMaterialf(GL.GL_FRONT, GL.GL_SHININESS, shineHigh);
glut.glutSolidCube(0.1f);
gl.glPopMatrix();

gl.glPushMatrix();
gl.glTranslatef(-1.28f, -1.42f, 0.0f);
gl.glRotatef(121.0f, 0.0f, 0.0f, 1.0f);
gl.glMaterialfv(GL.GL_FRONT, GL.GL_AMBIENT, ambientBlue, 0);
gl.glMaterialfv(GL.GL_FRONT, GL.GL_DIFFUSE, diffuseBlue, 0);
gl.glMaterialfv(GL.GL_FRONT, GL.GL_SPECULAR, specularBlue, 0);
gl.glMaterialf(GL.GL_FRONT, GL.GL_SHININESS, shineHigh);
glut.glutSolidCube(0.1f);
gl.glPopMatrix();

gl.glPushMatrix();
```

```
gl.glTranslatef(-1.29f, -1.46f, 0.0f);
gl.glRotatef(135.0f, 0.0f, 0.0f, 1.0f);
gl.glMaterialfv(GL.GL_FRONT, GL.GL_AMBIENT, ambientBlue, 0);
gl.glMaterialfv(GL.GL_FRONT, GL.GL_DIFFUSE, diffuseBlue, 0);
gl.glMaterialfv(GL.GL_FRONT, GL.GL_SPECULAR, specularBlue, 0);
gl.glMaterialf(GL.GL_FRONT, GL.GL_SHININESS, shineHigh);
glut.glutSolidCube(0.1f);
gl.glPopMatrix();



// right hand - fingers
// first finger
gl.glPushMatrix();
gl.glTranslatef(1.50f, -1.34f, 0.15f);
gl.glRotatef(-90.0f, 1.0f, 0.0f, 0.0f);
gl.glMaterialfv(GL.GL_FRONT, GL.GL_AMBIENT, ambientBlue, 0);
gl.glMaterialfv(GL.GL_FRONT, GL.GL_DIFFUSE, diffuseBlue, 0);
gl.glMaterialfv(GL.GL_FRONT, GL.GL_SPECULAR, specularBlue, 0);
gl.glMaterialf(GL.GL_FRONT, GL.GL_SHININESS, shineHigh);
glut.glutSolidCube(0.1f);
gl.glPopMatrix();

gl.glPushMatrix();
gl.glTranslatef(1.49f, -1.38f, 0.15f);
//angle, x, y, z
gl.glRotatef(-105.0f, 0.0f, 0.0f, 1.0f);
gl.glMaterialfv(GL.GL_FRONT, GL.GL_AMBIENT, ambientBlue, 0);
gl.glMaterialfv(GL.GL_FRONT, GL.GL_DIFFUSE, diffuseBlue, 0);
gl.glMaterialfv(GL.GL_FRONT, GL.GL_SPECULAR, specularBlue, 0);
gl.glMaterialf(GL.GL_FRONT, GL.GL_SHININESS, shineHigh);
glut.glutSolidCube(0.1f);
gl.glPopMatrix();
```

```
gl.glPushMatrix();
gl.glTranslatef(1.48f, -1.42f, 0.15f);
gl.glRotatef(-121.0f, 0.0f, 0.0f, 1.0f);
gl.glMaterialfv(GL.GL_FRONT, GL.GL_AMBIENT, ambientBlue, 0);
gl.glMaterialfv(GL.GL_FRONT, GL.GL_DIFFUSE, diffuseBlue, 0);
gl.glMaterialfv(GL.GL_FRONT, GL.GL_SPECULAR, specularBlue, 0);
gl.glMaterialf(GL.GL_FRONT, GL.GL_SHININESS, shineHigh);
glut.glutSolidCube(0.1f);
gl.glPopMatrix();

gl.glPushMatrix();
gl.glTranslatef(1.47f, -1.46f, 0.15f);
gl.glRotatef(-135.0f, 0.0f, 0.0f, 1.0f);
gl.glMaterialfv(GL.GL_FRONT, GL.GL_AMBIENT, ambientBlue, 0);
gl.glMaterialfv(GL.GL_FRONT, GL.GL_DIFFUSE, diffuseBlue, 0);
gl.glMaterialfv(GL.GL_FRONT, GL.GL_SPECULAR, specularBlue, 0);
gl.glMaterialf(GL.GL_FRONT, GL.GL_SHININESS, shineHigh);
glut.glutSolidCube(0.1f);
gl.glPopMatrix();

//second finger
gl.glPushMatrix();
gl.glTranslatef(1.52f, -1.34f, 0.0f);
gl.glRotatef(-90.0f, 1.0f, 0.0f, 0.0f);
gl.glMaterialfv(GL.GL_FRONT, GL.GL_AMBIENT, ambientBlue, 0);
gl.glMaterialfv(GL.GL_FRONT, GL.GL_DIFFUSE, diffuseBlue, 0);
gl.glMaterialfv(GL.GL_FRONT, GL.GL_SPECULAR, specularBlue, 0);
gl.glMaterialf(GL.GL_FRONT, GL.GL_SHININESS, shineHigh);
glut.glutSolidCube(0.1f);
gl.glPopMatrix();
```

```
gl.glPushMatrix();
gl.glTranslatef(1.51f, -1.38f, 0.0f);
//angle, x, y, z
gl.glRotatef(-105.0f, 0.0f, 0.0f, 1.0f);
gl.glMaterialfv(GL.GL_FRONT, GL.GL_AMBIENT, ambientBlue, 0);
gl.glMaterialfv(GL.GL_FRONT, GL.GL_DIFFUSE, diffuseBlue, 0);
gl.glMaterialfv(GL.GL_FRONT, GL.GL_SPECULAR, specularBlue, 0);
gl.glMaterialf(GL.GL_FRONT, GL.GL_SHININESS, shineHigh);
glut.glutSolidCube(0.1f);
gl.glPopMatrix();

gl.glPushMatrix();
gl.glTranslatef(1.50f, -1.42f, 0.0f);
gl.glRotatef(-121.0f, 0.0f, 0.0f, 1.0f);
gl.glMaterialfv(GL.GL_FRONT, GL.GL_AMBIENT, ambientBlue, 0);
gl.glMaterialfv(GL.GL_FRONT, GL.GL_DIFFUSE, diffuseBlue, 0);
gl.glMaterialfv(GL.GL_FRONT, GL.GL_SPECULAR, specularBlue, 0);
gl.glMaterialf(GL.GL_FRONT, GL.GL_SHININESS, shineHigh);
glut.glutSolidCube(0.1f);
gl.glPopMatrix();

gl.glPushMatrix();
gl.glTranslatef(1.49f, -1.46f, 0.0f);
gl.glRotatef(-135.0f, 0.0f, 0.0f, 1.0f);
gl.glMaterialfv(GL.GL_FRONT, GL.GL_AMBIENT, ambientBlue, 0);
gl.glMaterialfv(GL.GL_FRONT, GL.GL_DIFFUSE, diffuseBlue, 0);
gl.glMaterialfv(GL.GL_FRONT, GL.GL_SPECULAR, specularBlue, 0);
gl.glMaterialf(GL.GL_FRONT, GL.GL_SHININESS, shineHigh);
glut.glutSolidCube(0.1f);
gl.glPopMatrix();

//third finger
```

```
gl.glPushMatrix();
gl.glTranslatef(1.51f, -1.34f, -0.15f);
gl.glRotatef(-90.0f, 1.0f, 0.0f, 0.0f);
gl.glMaterialfv(GL.GL_FRONT, GL.GL_AMBIENT, ambientBlue, 0);
gl.glMaterialfv(GL.GL_FRONT, GL.GL_DIFFUSE, diffuseBlue, 0);
gl.glMaterialfv(GL.GL_FRONT, GL.GL_SPECULAR, specularBlue, 0);
gl.glMaterialf(GL.GL_FRONT, GL.GL_SHININESS, shineHigh);
glut.glutSolidCube(0.1f);
gl.glPopMatrix();

gl.glPushMatrix();
gl.glTranslatef(1.50f, -1.38f, -0.15f);
//angle, x, y, z
gl.glRotatef(-105.0f, 0.0f, 0.0f, 1.0f);
gl.glMaterialfv(GL.GL_FRONT, GL.GL_AMBIENT, ambientBlue, 0);
gl.glMaterialfv(GL.GL_FRONT, GL.GL_DIFFUSE, diffuseBlue, 0);
gl.glMaterialfv(GL.GL_FRONT, GL.GL_SPECULAR, specularBlue, 0);
gl.glMaterialf(GL.GL_FRONT, GL.GL_SHININESS, shineHigh);
glut.glutSolidCube(0.1f);
gl.glPopMatrix();

gl.glPushMatrix();
gl.glTranslatef(1.49f, -1.42f, -0.15f);
gl.glRotatef(-121.0f, 0.0f, 0.0f, 1.0f);
gl.glMaterialfv(GL.GL_FRONT, GL.GL_AMBIENT, ambientBlue, 0);
gl.glMaterialfv(GL.GL_FRONT, GL.GL_DIFFUSE, diffuseBlue, 0);
gl.glMaterialfv(GL.GL_FRONT, GL.GL_SPECULAR, specularBlue, 0);
gl.glMaterialf(GL.GL_FRONT, GL.GL_SHININESS, shineHigh);
glut.glutSolidCube(0.1f);
gl.glPopMatrix();

gl.glPushMatrix();
```

```
gl.glTranslatef(1.48f, -1.46f, -0.15f);
gl.glRotatef(-135.0f, 0.0f, 0.0f, 1.0f);
gl.glMaterialfv(GL.GL_FRONT, GL.GL_AMBIENT, ambientBlue, 0);
gl.glMaterialfv(GL.GL_FRONT, GL.GL_DIFFUSE, diffuseBlue, 0);
gl.glMaterialfv(GL.GL_FRONT, GL.GL_SPECULAR, specularBlue, 0);
gl.glMaterialf(GL.GL_FRONT, GL.GL_SHININESS, shineHigh);
glut.glutSolidCube(0.1f);
gl.glPopMatrix();

//right hand - thumb
gl.glPushMatrix();
gl.glTranslatef(1.26f, -1.34f, 0.0f);
gl.glRotatef(-90.0f, 1.0f, 0.0f, 0.0f);
gl.glMaterialfv(GL.GL_FRONT, GL.GL_AMBIENT, ambientBlue, 0);
gl.glMaterialfv(GL.GL_FRONT, GL.GL_DIFFUSE, diffuseBlue, 0);
gl.glMaterialfv(GL.GL_FRONT, GL.GL_SPECULAR, specularBlue, 0);
gl.glMaterialf(GL.GL_FRONT, GL.GL_SHININESS, shineHigh);
glut.glutSolidCube(0.1f);
gl.glPopMatrix();

gl.glPushMatrix();
gl.glTranslatef(1.27f, -1.38f, 0.0f);
//angle, x, y, z
gl.glRotatef(-105.0f, 0.0f, 0.0f, 1.0f);
gl.glMaterialfv(GL.GL_FRONT, GL.GL_AMBIENT, ambientBlue, 0);
gl.glMaterialfv(GL.GL_FRONT, GL.GL_DIFFUSE, diffuseBlue, 0);
gl.glMaterialfv(GL.GL_FRONT, GL.GL_SPECULAR, specularBlue, 0);
gl.glMaterialf(GL.GL_FRONT, GL.GL_SHININESS, shineHigh);
glut.glutSolidCube(0.1f);
gl.glPopMatrix();

gl.glPushMatrix();
```

```
gl.glTranslatef(1.28f, -1.42f, 0.0f);
gl.glRotatef(-121.0f, 0.0f, 0.0f, 1.0f);
gl.glMaterialfv(GL.GL_FRONT, GL.GL_AMBIENT, ambientBlue, 0);
gl.glMaterialfv(GL.GL_FRONT, GL.GL_DIFFUSE, diffuseBlue, 0);
gl.glMaterialfv(GL.GL_FRONT, GL.GL_SPECULAR, specularBlue, 0);
gl.glMaterialf(GL.GL_FRONT, GL.GL_SHININESS, shineHigh);
glut.glutSolidCube(0.1f);
gl.glPopMatrix();

gl.glPushMatrix();
gl.glTranslatef(1.29f, -1.46f, 0.0f);
gl.glRotatef(-135.0f, 0.0f, 0.0f, 1.0f);
gl.glMaterialfv(GL.GL_FRONT, GL.GL_AMBIENT, ambientBlue, 0);
gl.glMaterialfv(GL.GL_FRONT, GL.GL_DIFFUSE, diffuseBlue, 0);
gl.glMaterialfv(GL.GL_FRONT, GL.GL_SPECULAR, specularBlue, 0);
gl.glMaterialf(GL.GL_FRONT, GL.GL_SHININESS, shineHigh);
glut.glutSolidCube(0.1f);
gl.glPopMatrix();



// upper legs
gl.glPushMatrix();
gl.glTranslatef(-0.45f, -0.70f, 0.15f);
gl.glRotatef(90.0f, 1.0f, 0.0f, 0.0f);
gl.glMaterialfv(GL.GL_FRONT, GL.GL_AMBIENT, ambientBlue, 0);
gl.glMaterialfv(GL.GL_FRONT, GL.GL_DIFFUSE, diffuseBlue, 0);
gl.glMaterialfv(GL.GL_FRONT, GL.GL_SPECULAR, specularBlue, 0);
gl.glMaterialf(GL.GL_FRONT, GL.GL_SHININESS, shineHigh);
glut.glutSolidCube(0.2f);
gl.glPopMatrix();

gl.glPushMatrix();
```

```
gl.glTranslatef(-0.45f, -0.70f, 0.0f);
gl.glRotatef(90.0f, 1.0f, 0.0f, 0.0f);
gl.glMaterialfv(GL.GL_FRONT, GL.GL_AMBIENT, ambientBlue, 0);
gl.glMaterialfv(GL.GL_FRONT, GL.GL_DIFFUSE, diffuseBlue, 0);
gl.glMaterialfv(GL.GL_FRONT, GL.GL_SPECULAR, specularBlue, 0);
gl.glMaterialf(GL.GL_FRONT, GL.GL_SHININESS, shineHigh);
glut.glutSolidCube(0.2f);
gl.glPopMatrix();

gl.glPushMatrix();
gl.glTranslatef(-0.45f, -0.70f, -0.15f);
gl.glRotatef(90.0f, 1.0f, 0.0f, 0.0f);
gl.glMaterialfv(GL.GL_FRONT, GL.GL_AMBIENT, ambientBlue, 0);
gl.glMaterialfv(GL.GL_FRONT, GL.GL_DIFFUSE, diffuseBlue, 0);
gl.glMaterialfv(GL.GL_FRONT, GL.GL_SPECULAR, specularBlue, 0);
gl.glMaterialf(GL.GL_FRONT, GL.GL_SHININESS, shineHigh);
glut.glutSolidCube(0.2f);
gl.glPopMatrix();

gl.glPushMatrix();
gl.glTranslatef(0.45f, -0.70f, 0.15f);
gl.glRotatef(90.0f, 1.0f, 0.0f, 0.0f);
gl.glMaterialfv(GL.GL_FRONT, GL.GL_AMBIENT, ambientBlue, 0);
gl.glMaterialfv(GL.GL_FRONT, GL.GL_DIFFUSE, diffuseBlue, 0);
gl.glMaterialfv(GL.GL_FRONT, GL.GL_SPECULAR, specularBlue, 0);
gl.glMaterialf(GL.GL_FRONT, GL.GL_SHININESS, shineHigh);
glut.glutSolidCube(0.2f);
gl.glPopMatrix();

gl.glPushMatrix();
gl.glTranslatef(0.45f, -0.70f, 0.0f);
gl.glRotatef(90.0f, 1.0f, 0.0f, 0.0f);
```

```
gl.glMaterialfv(GL.GL_FRONT, GL.GL_AMBIENT, ambientBlue, 0);
gl.glMaterialfv(GL.GL_FRONT, GL.GL_DIFFUSE, diffuseBlue, 0);
gl.glMaterialfv(GL.GL_FRONT, GL.GL_SPECULAR, specularBlue, 0);
gl.glMaterialf(GL.GL_FRONT, GL.GL_SHININESS, shineHigh);
glut.glutSolidCube(0.2f);
gl.glPopMatrix();

gl.glPushMatrix();
gl.glTranslatef(0.45f, -0.70f, -0.15f);
gl.glRotatef(90.0f, 1.0f, 0.0f, 0.0f);
gl.glMaterialfv(GL.GL_FRONT, GL.GL_AMBIENT, ambientBlue, 0);
gl.glMaterialfv(GL.GL_FRONT, GL.GL_DIFFUSE, diffuseBlue, 0);
gl.glMaterialfv(GL.GL_FRONT, GL.GL_SPECULAR, specularBlue, 0);
gl.glMaterialf(GL.GL_FRONT, GL.GL_SHININESS, shineHigh);
glut.glutSolidCube(0.2f);
gl.glPopMatrix();

gl.glPushMatrix();
gl.glTranslatef(-0.15f, -0.70f, 0.15f);
gl.glRotatef(90.0f, 1.0f, 0.0f, 0.0f);
gl.glMaterialfv(GL.GL_FRONT, GL.GL_AMBIENT, ambientBlue, 0);
gl.glMaterialfv(GL.GL_FRONT, GL.GL_DIFFUSE, diffuseBlue, 0);
gl.glMaterialfv(GL.GL_FRONT, GL.GL_SPECULAR, specularBlue, 0);
gl.glMaterialf(GL.GL_FRONT, GL.GL_SHININESS, shineHigh);
glut.glutSolidCube(0.2f);
gl.glPopMatrix();

gl.glPushMatrix();
gl.glTranslatef(-0.15f, -0.70f, 0.0f);
gl.glRotatef(90.0f, 1.0f, 0.0f, 0.0f);
gl.glMaterialfv(GL.GL_FRONT, GL.GL_AMBIENT, ambientBlue, 0);
gl.glMaterialfv(GL.GL_FRONT, GL.GL_DIFFUSE, diffuseBlue, 0);
```

```
gl.glMaterialfv(GL.GL_FRONT, GL.GL_SPECULAR, specularBlue, 0);
gl.glMaterialf(GL.GL_FRONT, GL.GL_SHININESS, shineHigh);
glut.glutSolidCube(0.2f);
gl.glPopMatrix();

gl.glPushMatrix();
gl.glTranslatef(-0.15f, -0.70f, -0.15f);
gl.glRotatef(90.0f, 1.0f, 0.0f, 0.0f);
gl.glMaterialfv(GL.GL_FRONT, GL.GL_AMBIENT, ambientBlue, 0);
gl.glMaterialfv(GL.GL_FRONT, GL.GL_DIFFUSE, diffuseBlue, 0);
gl.glMaterialfv(GL.GL_FRONT, GL.GL_SPECULAR, specularBlue, 0);
gl.glMaterialf(GL.GL_FRONT, GL.GL_SHININESS, shineHigh);
glut.glutSolidCube(0.2f);
gl.glPopMatrix();

gl.glPushMatrix();
gl.glTranslatef(0.15f, -0.70f, 0.15f);
gl.glRotatef(90.0f, 1.0f, 0.0f, 0.0f);
gl.glMaterialfv(GL.GL_FRONT, GL.GL_AMBIENT, ambientBlue, 0);
gl.glMaterialfv(GL.GL_FRONT, GL.GL_DIFFUSE, diffuseBlue, 0);
gl.glMaterialfv(GL.GL_FRONT, GL.GL_SPECULAR, specularBlue, 0);
gl.glMaterialf(GL.GL_FRONT, GL.GL_SHININESS, shineHigh);
glut.glutSolidCube(0.2f);
gl.glPopMatrix();

gl.glPushMatrix();
gl.glTranslatef(0.15f, -0.70f, 0.0f);
gl.glRotatef(90.0f, 1.0f, 0.0f, 0.0f);
gl.glMaterialfv(GL.GL_FRONT, GL.GL_AMBIENT, ambientBlue, 0);
gl.glMaterialfv(GL.GL_FRONT, GL.GL_DIFFUSE, diffuseBlue, 0);
gl.glMaterialfv(GL.GL_FRONT, GL.GL_SPECULAR, specularBlue, 0);
gl.glMaterialf(GL.GL_FRONT, GL.GL_SHININESS, shineHigh);
```

```
glut.glutSolidCube(0.2f);
gl.glPopMatrix();

gl.glPushMatrix();
gl.glTranslatef(0.15f, -0.70f, -0.15f);
gl.glRotatef(90.0f, 1.0f, 0.0f, 0.0f);
gl.glMaterialfv(GL.GL_FRONT, GL.GL_AMBIENT, ambientBlue, 0);
gl.glMaterialfv(GL.GL_FRONT, GL.GL_DIFFUSE, diffuseBlue, 0);
gl.glMaterialfv(GL.GL_FRONT, GL.GL_SPECULAR, specularBlue, 0);
gl.glMaterialf(GL.GL_FRONT, GL.GL_SHININESS, shineHigh);
glut.glutSolidCube(0.2f);
gl.glPopMatrix();

gl.glPushMatrix();
gl.glTranslatef(-0.30f, -1.02f, 0.0f);
gl.glRotatef(90.0f, 1.0f, 0.0f, 0.0f);
gl.glMaterialfv(GL.GL_FRONT, GL.GL_AMBIENT, ambientBlue, 0);
gl.glMaterialfv(GL.GL_FRONT, GL.GL_DIFFUSE, diffuseBlue, 0);
gl.glMaterialfv(GL.GL_FRONT, GL.GL_SPECULAR, specularBlue, 0);
gl.glMaterialf(GL.GL_FRONT, GL.GL_SHININESS, shineHigh);
glut.glutSolidCube(0.5f);
gl.glPopMatrix();

gl.glPushMatrix();
gl.glTranslatef(0.30f, -1.02f, 0.0f);
gl.glRotatef(90.0f, 1.0f, 0.0f, 0.0f);
gl.glMaterialfv(GL.GL_FRONT, GL.GL_AMBIENT, ambientBlue, 0);
gl.glMaterialfv(GL.GL_FRONT, GL.GL_DIFFUSE, diffuseBlue, 0);
gl.glMaterialfv(GL.GL_FRONT, GL.GL_SPECULAR, specularBlue, 0);
gl.glMaterialf(GL.GL_FRONT, GL.GL_SHININESS, shineHigh);
glut.glutSolidCube(0.5f);
gl.glPopMatrix();
```

```
gl.glPushMatrix();
gl.glTranslatef(-0.30f, -1.14f, 0.0f);
gl.glRotatef(90.0f, 1.0f, 0.0f, 0.0f);
gl.glMaterialfv(GL.GL_FRONT, GL.GL_AMBIENT, ambientBlue, 0);
gl.glMaterialfv(GL.GL_FRONT, GL.GL_DIFFUSE, diffuseBlue, 0);
gl.glMaterialfv(GL.GL_FRONT, GL.GL_SPECULAR, specularBlue, 0);
gl.glMaterialf(GL.GL_FRONT, GL.GL_SHININESS, shineHigh);
glut.glutSolidCube(0.5f);
gl.glPopMatrix();

gl.glPushMatrix();
gl.glTranslatef(0.30f, -1.14f, 0.0f);
gl.glRotatef(90.0f, 1.0f, 0.0f, 0.0f);
gl.glMaterialfv(GL.GL_FRONT, GL.GL_AMBIENT, ambientBlue, 0);
gl.glMaterialfv(GL.GL_FRONT, GL.GL_DIFFUSE, diffuseBlue, 0);
gl.glMaterialfv(GL.GL_FRONT, GL.GL_SPECULAR, specularBlue, 0);
gl.glMaterialf(GL.GL_FRONT, GL.GL_SHININESS, shineHigh);
glut.glutSolidCube(0.5f);
gl.glPopMatrix();

gl.glPushMatrix();
gl.glTranslatef(0.45f, -1.48f, 0.15f);
gl.glRotatef(90.0f, 1.0f, 0.0f, 0.0f);
gl.glMaterialfv(GL.GL_FRONT, GL.GL_AMBIENT, ambientBlue, 0);
gl.glMaterialfv(GL.GL_FRONT, GL.GL_DIFFUSE, diffuseBlue, 0);
gl.glMaterialfv(GL.GL_FRONT, GL.GL_SPECULAR, specularBlue, 0);
gl.glMaterialf(GL.GL_FRONT, GL.GL_SHININESS, shineHigh);
glut.glutSolidCube(0.2f);
gl.glPopMatrix();

gl.glPushMatrix();
```

```
gl.glTranslatef(0.45f, -1.48f, 0.0f);
gl.glRotatef(90.0f, 1.0f, 0.0f, 0.0f);
gl.glMaterialfv(GL.GL_FRONT, GL.GL_AMBIENT, ambientBlue, 0);
gl.glMaterialfv(GL.GL_FRONT, GL.GL_DIFFUSE, diffuseBlue, 0);
gl.glMaterialfv(GL.GL_FRONT, GL.GL_SPECULAR, specularBlue, 0);
gl.glMaterialf(GL.GL_FRONT, GL.GL_SHININESS, shineHigh);
glut.glutSolidCube(0.2f);
gl.glPopMatrix();

gl.glPushMatrix();
gl.glTranslatef(0.45f, -1.48f, -0.15f);
gl.glRotatef(90.0f, 1.0f, 0.0f, 0.0f);
gl.glMaterialfv(GL.GL_FRONT, GL.GL_AMBIENT, ambientBlue, 0);
gl.glMaterialfv(GL.GL_FRONT, GL.GL_DIFFUSE, diffuseBlue, 0);
gl.glMaterialfv(GL.GL_FRONT, GL.GL_SPECULAR, specularBlue, 0);
gl.glMaterialf(GL.GL_FRONT, GL.GL_SHININESS, shineHigh);
glut.glutSolidCube(0.2f);
gl.glPopMatrix();

gl.glPushMatrix();
gl.glTranslatef(-0.45f, -1.48f, 0.15f);
gl.glRotatef(90.0f, 1.0f, 0.0f, 0.0f);
gl.glMaterialfv(GL.GL_FRONT, GL.GL_AMBIENT, ambientBlue, 0);
gl.glMaterialfv(GL.GL_FRONT, GL.GL_DIFFUSE, diffuseBlue, 0);
gl.glMaterialfv(GL.GL_FRONT, GL.GL_SPECULAR, specularBlue, 0);
gl.glMaterialf(GL.GL_FRONT, GL.GL_SHININESS, shineHigh);
glut.glutSolidCube(0.2f);
gl.glPopMatrix();

gl.glPushMatrix();
gl.glTranslatef(-0.45f, -1.48f, 0.0f);
gl.glRotatef(90.0f, 1.0f, 0.0f, 0.0f);
```

```
gl.glMaterialfv(GL.GL_FRONT, GL.GL_AMBIENT, ambientBlue, 0);
gl.glMaterialfv(GL.GL_FRONT, GL.GL_DIFFUSE, diffuseBlue, 0);
gl.glMaterialfv(GL.GL_FRONT, GL.GL_SPECULAR, specularBlue, 0);
gl.glMaterialf(GL.GL_FRONT, GL.GL_SHININESS, shineHigh);
glut.glutSolidCube(0.2f);
gl.glPopMatrix();

gl.glPushMatrix();
gl.glTranslatef(-0.45f, -1.48f, -0.15f);
gl.glRotatef(90.0f, 1.0f, 0.0f, 0.0f);
gl.glMaterialfv(GL.GL_FRONT, GL.GL_AMBIENT, ambientBlue, 0);
gl.glMaterialfv(GL.GL_FRONT, GL.GL_DIFFUSE, diffuseBlue, 0);
gl.glMaterialfv(GL.GL_FRONT, GL.GL_SPECULAR, specularBlue, 0);
gl.glMaterialf(GL.GL_FRONT, GL.GL_SHININESS, shineHigh);
glut.glutSolidCube(0.2f);
gl.glPopMatrix();

gl.glPushMatrix();
gl.glTranslatef(0.15f, -1.48f, 0.15f);
gl.glRotatef(90.0f, 1.0f, 0.0f, 0.0f);
gl.glMaterialfv(GL.GL_FRONT, GL.GL_AMBIENT, ambientBlue, 0);
gl.glMaterialfv(GL.GL_FRONT, GL.GL_DIFFUSE, diffuseBlue, 0);
gl.glMaterialfv(GL.GL_FRONT, GL.GL_SPECULAR, specularBlue, 0);
gl.glMaterialf(GL.GL_FRONT, GL.GL_SHININESS, shineHigh);
glut.glutSolidCube(0.2f);
gl.glPopMatrix();

gl.glPushMatrix();
gl.glTranslatef(0.15f, -1.48f, 0.0f);
gl.glRotatef(90.0f, 1.0f, 0.0f, 0.0f);
gl.glMaterialfv(GL.GL_FRONT, GL.GL_AMBIENT, ambientBlue, 0);
gl.glMaterialfv(GL.GL_FRONT, GL.GL_DIFFUSE, diffuseBlue, 0);
```

```
gl.glMaterialfv(GL.GL_FRONT, GL.GL_SPECULAR, specularBlue, 0);
gl.glMaterialf(GL.GL_FRONT, GL.GL_SHININESS, shineHigh);
glut.glutSolidCube(0.2f);
gl.glPopMatrix();

gl.glPushMatrix();
gl.glTranslatef(0.15f, -1.48f, -0.15f);
gl.glRotatef(90.0f, 1.0f, 0.0f, 0.0f);
gl.glMaterialfv(GL.GL_FRONT, GL.GL_AMBIENT, ambientBlue, 0);
gl.glMaterialfv(GL.GL_FRONT, GL.GL_DIFFUSE, diffuseBlue, 0);
gl.glMaterialfv(GL.GL_FRONT, GL.GL_SPECULAR, specularBlue, 0);
gl.glMaterialf(GL.GL_FRONT, GL.GL_SHININESS, shineHigh);
glut.glutSolidCube(0.2f);
gl.glPopMatrix();

gl.glPushMatrix();
gl.glTranslatef(-0.15f, -1.48f, 0.15f);
gl.glRotatef(90.0f, 1.0f, 0.0f, 0.0f);
gl.glMaterialfv(GL.GL_FRONT, GL.GL_AMBIENT, ambientBlue, 0);
gl.glMaterialfv(GL.GL_FRONT, GL.GL_DIFFUSE, diffuseBlue, 0);
gl.glMaterialfv(GL.GL_FRONT, GL.GL_SPECULAR, specularBlue, 0);
gl.glMaterialf(GL.GL_FRONT, GL.GL_SHININESS, shineHigh);
glut.glutSolidCube(0.2f);
gl.glPopMatrix();

gl.glPushMatrix();
gl.glTranslatef(-0.15f, -1.48f, 0.0f);
gl.glRotatef(90.0f, 1.0f, 0.0f, 0.0f);
gl.glMaterialfv(GL.GL_FRONT, GL.GL_AMBIENT, ambientBlue, 0);
gl.glMaterialfv(GL.GL_FRONT, GL.GL_DIFFUSE, diffuseBlue, 0);
gl.glMaterialfv(GL.GL_FRONT, GL.GL_SPECULAR, specularBlue, 0);
gl.glMaterialf(GL.GL_FRONT, GL.GL_SHININESS, shineHigh);
```

```
glut.glutSolidCube(0.2f);
gl.glPopMatrix();


gl.glPushMatrix();
gl.glTranslatef(-0.15f, -1.48f, -0.15f);
gl.glRotatef(90.0f, 1.0f, 0.0f, 0.0f);
gl.glMaterialfv(GL.GL_FRONT, GL.GL_AMBIENT, ambientBlue, 0);
gl.glMaterialfv(GL.GL_FRONT, GL.GL_DIFFUSE, diffuseBlue, 0);
gl.glMaterialfv(GL.GL_FRONT, GL.GL_SPECULAR, specularBlue, 0);
gl.glMaterialf(GL.GL_FRONT, GL.GL_SHININESS, shineHigh);
glut.glutSolidCube(0.2f);
gl.glPopMatrix();



// lower legs
gl.glPushMatrix();
gl.glTranslatef(-0.30f, -1.80f, 0.0f);
gl.glRotatef(90.0f, 1.0f, 0.0f, 0.0f);
gl.glMaterialfv(GL.GL_FRONT, GL.GL_AMBIENT, ambientBlack, 0);
gl.glMaterialfv(GL.GL_FRONT, GL.GL_DIFFUSE, diffuseBlack, 0);
gl.glMaterialfv(GL.GL_FRONT, GL.GL_SPECULAR, specularBlack, 0);
gl.glMaterialf(GL.GL_FRONT, GL.GL_SHININESS, shineHigh);
glut.glutSolidCube(0.5f);
gl.glPopMatrix();


gl.glPushMatrix();
gl.glTranslatef(-0.30f, -2.22f, 0.0f);
gl.glRotatef(90.0f, 1.0f, 0.0f, 0.0f);
gl.glMaterialfv(GL.GL_FRONT, GL.GL_AMBIENT, ambientBlack, 0);
gl.glMaterialfv(GL.GL_FRONT, GL.GL_DIFFUSE, diffuseBlack, 0);
gl.glMaterialfv(GL.GL_FRONT, GL.GL_SPECULAR, specularBlack, 0);
gl.glMaterialf(GL.GL_FRONT, GL.GL_SHININESS, shineHigh);
```

```
glut.glutSolidCube(0.5f);
gl.glPopMatrix();

gl.glPushMatrix();
gl.glTranslatef(0.30f, -1.80f, 0.0f);
gl.glRotatef(90.0f, 1.0f, 0.0f, 0.0f);
gl.glMaterialfv(GL.GL_FRONT, GL.GL_AMBIENT, ambientBlack, 0);
gl.glMaterialfv(GL.GL_FRONT, GL.GL_DIFFUSE, diffuseBlack, 0);
gl.glMaterialfv(GL.GL_FRONT, GL.GL_SPECULAR, specularBlack, 0);
gl.glMaterialf(GL.GL_FRONT, GL.GL_SHININESS, shineHigh);
glut.glutSolidCube(0.5f);
gl.glPopMatrix();

gl.glPushMatrix();
gl.glTranslatef(0.30f, -2.22f, 0.0f);
gl.glRotatef(90.0f, 1.0f, 0.0f, 0.0f);
gl.glMaterialfv(GL.GL_FRONT, GL.GL_AMBIENT, ambientBlack, 0);
gl.glMaterialfv(GL.GL_FRONT, GL.GL_DIFFUSE, diffuseBlack, 0);
gl.glMaterialfv(GL.GL_FRONT, GL.GL_SPECULAR, specularBlack, 0);
gl.glMaterialf(GL.GL_FRONT, GL.GL_SHININESS, shineHigh);
glut.glutSolidCube(0.5f);
gl.glPopMatrix();



//left side leg plating
gl.glPushMatrix();
gl.glTranslatef(-0.30f, -1.30f, 0.20f);
gl.glRotatef(90.0f, 1.0f, 0.0f, 0.0f);
glut.glutSolidCube(0.4f);
gl.glPopMatrix();

//right side leg plating
```

```
gl.glPushMatrix();
gl.glTranslatef(0.30f, -1.30f, 0.20f);
gl.glRotatef(90.0f, 1.0f, 0.0f, 0.0f);
glut.glutSolidCube(0.4f);
gl.glPopMatrix();



//left side foot
gl.glPushMatrix();
gl.glTranslatef(-0.30f, -2.50f, 0.30f);
gl.glRotatef(90.0f, 1.0f, 0.0f, 0.0f);
gl.glMaterialfv(GL.GL_FRONT, GL.GL_AMBIENT, ambientBlack, 0);
gl.glMaterialfv(GL.GL_FRONT, GL.GL_DIFFUSE, diffuseBlack, 0);
gl.glMaterialfv(GL.GL_FRONT, GL.GL_SPECULAR, specularBlack, 0);
gl.glMaterialf(GL.GL_FRONT, GL.GL_SHININESS, shineHigh);
glut.glutSolidCube(0.5f);
gl.glPopMatrix();

gl.glPushMatrix();
gl.glTranslatef(-0.30f, -2.25f, 0.20f);
gl.glRotatef(45.0f, 1.0f, 0.0f, 0.0f);
gl.glMaterialfv(GL.GL_FRONT, GL.GL_AMBIENT, ambientBlack, 0);
gl.glMaterialfv(GL.GL_FRONT, GL.GL_DIFFUSE, diffuseBlack, 0);
gl.glMaterialfv(GL.GL_FRONT, GL.GL_SPECULAR, specularBlack, 0);
gl.glMaterialf(GL.GL_FRONT, GL.GL_SHININESS, shineHigh);
glut.glutSolidCube(0.5f);
gl.glPopMatrix();

gl.glPushMatrix();
gl.glTranslatef(-0.30f, -2.50f, 0.0f);
gl.glRotatef(90.0f, 1.0f, 0.0f, 0.0f);
gl.glMaterialfv(GL.GL_FRONT, GL.GL_AMBIENT, ambientBlack, 0);
```

```
gl.glMaterialfv(GL.GL_FRONT, GL.GL_DIFFUSE, diffuseBlack, 0);
gl.glMaterialfv(GL.GL_FRONT, GL.GL_SPECULAR, specularBlack, 0);
gl.glMaterialf(GL.GL_FRONT, GL.GL_SHININESS, shineHigh);
glut.glutSolidCube(0.5f);
gl.glPopMatrix();

gl.glPushMatrix();
gl.glTranslatef(-0.20f, -2.50f, 0.50f);
gl.glRotatef(45.0f, 1.0f, 0.0f, 0.0f);
gl.glMaterialfv(GL.GL_FRONT, GL.GL_AMBIENT, ambientBlack, 0);
gl.glMaterialfv(GL.GL_FRONT, GL.GL_DIFFUSE, diffuseBlack, 0);
gl.glMaterialfv(GL.GL_FRONT, GL.GL_SPECULAR, specularBlack, 0);
gl.glMaterialf(GL.GL_FRONT, GL.GL_SHININESS, shineHigh);
glut.glutSolidCube(0.3f);
gl.glPopMatrix();

gl.glPushMatrix();
gl.glTranslatef(-0.40f, -2.50f, 0.50f);
gl.glRotatef(45.0f, 1.0f, 0.0f, 0.0f);
gl.glMaterialfv(GL.GL_FRONT, GL.GL_AMBIENT, ambientBlack, 0);
gl.glMaterialfv(GL.GL_FRONT, GL.GL_DIFFUSE, diffuseBlack, 0);
gl.glMaterialfv(GL.GL_FRONT, GL.GL_SPECULAR, specularBlack, 0);
gl.glMaterialf(GL.GL_FRONT, GL.GL_SHININESS, shineHigh);
glut.glutSolidCube(0.3f);
gl.glPopMatrix();

gl.glPushMatrix();
gl.glTranslatef(-0.20f, -2.60f, 0.56f);
gl.glRotatef(90.0f, 1.0f, 0.0f, 0.0f);
gl.glMaterialfv(GL.GL_FRONT, GL.GL_AMBIENT, ambientBlack, 0);
gl.glMaterialfv(GL.GL_FRONT, GL.GL_DIFFUSE, diffuseBlack, 0);
gl.glMaterialfv(GL.GL_FRONT, GL.GL_SPECULAR, specularBlack, 0);
```

```
gl.glMaterialf(GL.GL_FRONT, GL.GL_SHININESS, shineHigh);
glut.glutSolidCube(0.3f);
gl.glPopMatrix();

gl.glPushMatrix();
gl.glTranslatef(-0.40f, -2.60f, 0.56f);
gl.glRotatef(90.0f, 1.0f, 0.0f, 0.0f);
gl.glMaterialfv(GL.GL_FRONT, GL.GL_AMBIENT, ambientBlack, 0);
gl.glMaterialfv(GL.GL_FRONT, GL.GL_DIFFUSE, diffuseBlack, 0);
gl.glMaterialfv(GL.GL_FRONT, GL.GL_SPECULAR, specularBlack, 0);
gl.glMaterialf(GL.GL_FRONT, GL.GL_SHININESS, shineHigh);
glut.glutSolidCube(0.3f);
gl.glPopMatrix();



//right side foot
gl.glPushMatrix();
gl.glTranslatef(0.30f, -2.50f, 0.30f);
gl.glRotatef(90.0f, 1.0f, 0.0f, 0.0f);
gl.glMaterialfv(GL.GL_FRONT, GL.GL_AMBIENT, ambientBlack, 0);
gl.glMaterialfv(GL.GL_FRONT, GL.GL_DIFFUSE, diffuseBlack, 0);
gl.glMaterialfv(GL.GL_FRONT, GL.GL_SPECULAR, specularBlack, 0);
gl.glMaterialf(GL.GL_FRONT, GL.GL_SHININESS, shineHigh);
glut.glutSolidCube(0.5f);
gl.glPopMatrix();

gl.glPushMatrix();
gl.glTranslatef(0.30f, -2.25f, 0.20f);
gl.glRotatef(45.0f, 1.0f, 0.0f, 0.0f);
gl.glMaterialfv(GL.GL_FRONT, GL.GL_AMBIENT, ambientBlack, 0);
gl.glMaterialfv(GL.GL_FRONT, GL.GL_DIFFUSE, diffuseBlack, 0);
gl.glMaterialfv(GL.GL_FRONT, GL.GL_SPECULAR, specularBlack, 0);
```

```
gl.glMaterialf(GL.GL_FRONT, GL.GL_SHININESS, shineHigh);
glut.glutSolidCube(0.5f);
gl.glPopMatrix();

gl.glPushMatrix();
gl.glTranslatef(0.30f, -2.50f, 0.0f);
gl.glRotatef(90.0f, 1.0f, 0.0f, 0.0f);
gl.glMaterialfv(GL.GL_FRONT, GL.GL_AMBIENT, ambientBlack, 0);
gl.glMaterialfv(GL.GL_FRONT, GL.GL_DIFFUSE, diffuseBlack, 0);
gl.glMaterialfv(GL.GL_FRONT, GL.GL_SPECULAR, specularBlack, 0);
gl.glMaterialf(GL.GL_FRONT, GL.GL_SHININESS, shineHigh);
glut.glutSolidCube(0.5f);
gl.glPopMatrix();

gl.glPushMatrix();
gl.glTranslatef(0.20f, -2.50f, 0.50f);
gl.glRotatef(45.0f, 1.0f, 0.0f, 0.0f);
gl.glMaterialfv(GL.GL_FRONT, GL.GL_AMBIENT, ambientBlack, 0);
gl.glMaterialfv(GL.GL_FRONT, GL.GL_DIFFUSE, diffuseBlack, 0);
gl.glMaterialfv(GL.GL_FRONT, GL.GL_SPECULAR, specularBlack, 0);
gl.glMaterialf(GL.GL_FRONT, GL.GL_SHININESS, shineHigh);
glut.glutSolidCube(0.3f);
gl.glPopMatrix();

gl.glPushMatrix();
gl.glTranslatef(0.40f, -2.50f, 0.50f);
gl.glRotatef(45.0f, 1.0f, 0.0f, 0.0f);
gl.glMaterialfv(GL.GL_FRONT, GL.GL_AMBIENT, ambientBlack, 0);
gl.glMaterialfv(GL.GL_FRONT, GL.GL_DIFFUSE, diffuseBlack, 0);
gl.glMaterialfv(GL.GL_FRONT, GL.GL_SPECULAR, specularBlack, 0);
gl.glMaterialf(GL.GL_FRONT, GL.GL_SHININESS, shineHigh);
glut.glutSolidCube(0.3f);
```

```
gl.glPopMatrix();


gl.glPushMatrix();
gl.glTranslatef(0.20f, -2.60f, 0.56f);
gl.glRotatef(90.0f, 1.0f, 0.0f, 0.0f);
gl.glMaterialfv(GL.GL_FRONT, GL.GL_AMBIENT, ambientBlack, 0);
gl.glMaterialfv(GL.GL_FRONT, GL.GL_DIFFUSE, diffuseBlack, 0);
gl.glMaterialfv(GL.GL_FRONT, GL.GL_SPECULAR, specularBlack, 0);
gl.glMaterialf(GL.GL_FRONT, GL.GL_SHININESS, shineHigh);
glut.glutSolidCube(0.3f);
gl.glPopMatrix();


gl.glPushMatrix();
gl.glTranslatef(0.40f, -2.60f, 0.56f);
gl.glRotatef(90.0f, 1.0f, 0.0f, 0.0f);
gl.glMaterialfv(GL.GL_FRONT, GL.GL_AMBIENT, ambientBlack, 0);
gl.glMaterialfv(GL.GL_FRONT, GL.GL_DIFFUSE, diffuseBlack, 0);
gl.glMaterialfv(GL.GL_FRONT, GL.GL_SPECULAR, specularBlack, 0);
gl.glMaterialf(GL.GL_FRONT, GL.GL_SHININESS, shineHigh);
glut.glutSolidCube(0.3f);
gl.glPopMatrix();



//leg attachments
gl.glPushMatrix();
gl.glTranslatef(-0.58f, -1.08f, 0.0f);
gl.glRotatef(90.0f, 1.0f, 0.0f, 0.0f);
gl.glMaterialfv(GL.GL_FRONT, GL.GL_AMBIENT, ambientGrey, 0);
gl.glMaterialfv(GL.GL_FRONT, GL.GL_DIFFUSE, diffuseGrey, 0);
gl.glMaterialfv(GL.GL_FRONT, GL.GL_SPECULAR, specularGrey, 0);
gl.glMaterialf(GL.GL_FRONT, GL.GL_SHININESS, shineHigh);
//radius, height, slices, stacks
```

```
glut.glutSolidCylinder(0.13, 0.8, 16, 1);
gl.glPopMatrix();

gl.glPushMatrix();
gl.glTranslatef(0.58f, -1.08f, 0.0f);
gl.glRotatef(90.0f, 1.0f, 0.0f, 0.0f);
gl.glMaterialfv(GL.GL_FRONT, GL.GL_AMBIENT, ambientGrey, 0);
gl.glMaterialfv(GL.GL_FRONT, GL.GL_DIFFUSE, diffuseGrey, 0);
gl.glMaterialfv(GL.GL_FRONT, GL.GL_SPECULAR, specularGrey, 0);
gl.glMaterialf(GL.GL_FRONT, GL.GL_SHININESS, shineHigh);
//radius, height, slices, stacks
glut.glutSolidCylinder(0.13, 0.8, 16, 1);
gl.glPopMatrix();



// body
gl.glPushMatrix();
gl.glTranslatef(-0.48f, 0.72f, 0.0f);
gl.glRotatef(90.0f, 1.0f, 0.0f, 0.0f);
gl.glMaterialfv(GL.GL_FRONT, GL.GL_AMBIENT, ambientBlack, 0);
gl.glMaterialfv(GL.GL_FRONT, GL.GL_DIFFUSE, diffuseBlack, 0);
gl.glMaterialfv(GL.GL_FRONT, GL.GL_SPECULAR, specularBlack, 0);
gl.glMaterialf(GL.GL_FRONT, GL.GL_SHININESS, shineHigh);
glut.glutSolidCube(0.9f);
gl.glPopMatrix();

gl.glPushMatrix();
gl.glTranslatef(0.00f, 0.72f, 0.0f);
gl.glRotatef(90.0f, 1.0f, 0.0f, 0.0f);
glut.glutSolidCube(0.9f);
gl.glPopMatrix();
```

```
gl.glPushMatrix();
gl.glTranslatef(0.48f, 0.72f, 0.0f);
gl.glRotatef(90.0f, 1.0f, 0.0f, 0.0f);
glut.glutSolidCube(0.9f);
gl.glPopMatrix();

gl.glPushMatrix();
gl.glTranslatef(-0.48f, 0.42f, 0.0f);
gl.glRotatef(90.0f, 1.0f, 0.0f, 0.0f);
glut.glutSolidCube(0.8f);
gl.glPopMatrix();

gl.glPushMatrix();
gl.glTranslatef(0.00f, 0.42f, 0.0f);
gl.glRotatef(90.0f, 1.0f, 0.0f, 0.0f);
glut.glutSolidCube(0.8f);
gl.glPopMatrix();

gl.glPushMatrix();
gl.glTranslatef(0.48f, 0.42f, 0.0f);
gl.glRotatef(90.0f, 1.0f, 0.0f, 0.0f);
glut.glutSolidCube(0.8f);
gl.glPopMatrix();

gl.glPushMatrix();
gl.glTranslatef(-0.48f, 0.10f, 0.0f);
gl.glRotatef(90.0f, 1.0f, 0.0f, 0.0f);
glut.glutSolidCube(0.6f);
gl.glPopMatrix();

gl.glPushMatrix();
gl.glTranslatef(0.00f, 0.10f, 0.0f);
```

```
gl.glRotatef(90.0f, 1.0f, 0.0f, 0.0f);
glut.glutSolidCube(0.6f);
gl.glPopMatrix();

gl.glPushMatrix();
gl.glTranslatef(0.48f, 0.10f, 0.0f);
gl.glRotatef(90.0f, 1.0f, 0.0f, 0.0f);
glut.glutSolidCube(0.6f);
gl.glPopMatrix();



//body-leg connection
gl.glPushMatrix();
gl.glTranslatef(0.80f, -0.40f, 0.0f);
gl.glRotatef(90.0f, 1.0f, 0.0f, 0.0f);
glut.glutSolidCube(0.4f);
gl.glPopMatrix();

gl.glPushMatrix();
gl.glTranslatef(0.55f, -0.40f, 0.0f);
gl.glRotatef(90.0f, 1.0f, 0.0f, 0.0f);
glut.glutSolidCube(0.4f);
gl.glPopMatrix();

gl.glPushMatrix();
gl.glTranslatef(0.30f, -0.40f, 0.0f);
gl.glRotatef(45.0f, 0.0f, 0.0f, 1.0f);
glut.glutSolidCube(0.4f);
gl.glPopMatrix();

gl.glPushMatrix();
gl.glTranslatef(-0.30f, -0.40f, 0.0f);
```

```
gl.glRotatef(45.0f, 0.0f, 0.0f, 1.0f);
glut.glutSolidCube(0.4f);
gl.glPopMatrix();

gl.glPushMatrix();
gl.glTranslatef(-0.55f, -0.40f, 0.0f);
gl.glRotatef(90.0f, 1.0f, 0.0f, 0.0f);
glut.glutSolidCube(0.4f);
gl.glPopMatrix();

gl.glPushMatrix();
gl.glTranslatef(-0.80f, -0.40f, 0.0f);
gl.glRotatef(90.0f, 1.0f, 0.0f, 0.0f);
glut.glutSolidCube(0.4f);
gl.glPopMatrix();



//body attachments
//left
gl.glPushMatrix();
gl.glTranslatef(-0.78f, 1.22f, -0.4f);
gl.glRotatef(0.0f, 1.0f, 0.0f, 0.0f);
gl.glMaterialfv(GL.GL_FRONT, GL.GL_AMBIENT, ambientGrey, 0);
gl.glMaterialfv(GL.GL_FRONT, GL.GL_DIFFUSE, diffuseGrey, 0);
gl.glMaterialfv(GL.GL_FRONT, GL.GL_SPECULAR, specularGrey, 0);
gl.glMaterialf(GL.GL_FRONT, GL.GL_SHININESS, shineHigh);
//radius, height, slices, stacks.
glut.glutSolidCylinder(0.06, 0.85, 16, 1);
gl.glPopMatrix();

gl.glPushMatrix();
```

```
gl.glTranslatef(-0.58f, 1.22f, -0.4f);
gl.glRotatef(0.0f, 1.0f, 0.0f, 0.0f);
gl.glMaterialfv(GL.GL_FRONT, GL.GL_AMBIENT, ambientGrey, 0);
gl.glMaterialfv(GL.GL_FRONT, GL.GL_DIFFUSE, diffuseGrey, 0);
gl.glMaterialfv(GL.GL_FRONT, GL.GL_SPECULAR, specularGrey, 0);
gl.glMaterialf(GL.GL_FRONT, GL.GL_SHININESS, shineHigh);
//radius, height, slices, stacks.
glut.glutSolidCylinder(0.06, 0.85, 16, 1);
gl.glPopMatrix();



//right
gl.glPushMatrix();
gl.glTranslatef(0.78f, 1.22f, -0.4f);
gl.glRotatef(0.0f, 1.0f, 0.0f, 0.0f);
gl.glMaterialfv(GL.GL_FRONT, GL.GL_AMBIENT, ambientGrey, 0);
gl.glMaterialfv(GL.GL_FRONT, GL.GL_DIFFUSE, diffuseGrey, 0);
gl.glMaterialfv(GL.GL_FRONT, GL.GL_SPECULAR, specularGrey, 0);
gl.glMaterialf(GL.GL_FRONT, GL.GL_SHININESS, shineHigh);
//radius, height, slices, stacks.
glut.glutSolidCylinder(0.06, 0.85, 16, 1);
gl.glPopMatrix();

gl.glPushMatrix();
gl.glTranslatef(0.58f, 1.22f, -0.4f);
//sets (angle, x, y, z) if x, y or z is set to one then it will
//rotate on that/those axis.
gl.glRotatef(0.0f, 1.0f, 0.0f, 0.0f);
gl.glMaterialfv(GL.GL_FRONT, GL.GL_AMBIENT, ambientGrey, 0);
gl.glMaterialfv(GL.GL_FRONT, GL.GL_DIFFUSE, diffuseGrey, 0);
gl.glMaterialfv(GL.GL_FRONT, GL.GL_SPECULAR, specularGrey, 0);
gl.glMaterialf(GL.GL_FRONT, GL.GL_SHININESS, shineHigh);
```

```
//radius, height, slices, stacks.
glut.glutSolidCylinder(0.06, 0.85, 16, 1);
gl.glPopMatrix();



//head
gl.glPushMatrix();
gl.glTranslatef(-0.10f, 1.56f, 0.0f);
gl.glRotatef(45.0f, 0.0f, 1.0f, 0.0f);
gl.glMaterialfv(GL.GL_FRONT, GL.GL_AMBIENT, ambientBlue, 0);
gl.glMaterialfv(GL.GL_FRONT, GL.GL_DIFFUSE, diffuseBlue, 0);
gl.glMaterialfv(GL.GL_FRONT, GL.GL_SPECULAR, specularBlue, 0);
gl.glMaterialf(GL.GL_FRONT, GL.GL_SHININESS, shineHigh);
glut.glutSolidCube(0.48f);
gl.glPopMatrix();

gl.glPushMatrix();
gl.glTranslatef(0.10f, 1.56f, 0.0f);
gl.glRotatef(45.0f, 0.0f, 1.0f, 0.0f);
gl.glMaterialfv(GL.GL_FRONT, GL.GL_AMBIENT, ambientBlue, 0);
gl.glMaterialfv(GL.GL_FRONT, GL.GL_DIFFUSE, diffuseBlue, 0);
gl.glMaterialfv(GL.GL_FRONT, GL.GL_SPECULAR, specularBlue, 0);
gl.glMaterialf(GL.GL_FRONT, GL.GL_SHININESS, shineHigh);
//size
glut.glutSolidCube(0.48f);
gl.glPopMatrix();



// head - eyes
//left eye
gl.glPushMatrix();
gl.glTranslatef(-0.16f, 1.56f, 0.24f);
```

```
gl.glRotatef(0.0f, 1.0f, 0.0f, 0.0f);
gl.glMaterialfv(GL.GL_FRONT, GL.GL_AMBIENT, ambientRed, 0);
gl.glMaterialfv(GL.GL_FRONT, GL.GL_DIFFUSE, diffuseRed, 0);
gl.glMaterialfv(GL.GL_FRONT, GL.GL_SPECULAR, specularRed, 0);
gl.glMaterialf(GL.GL_FRONT, GL.GL_SHININESS, shineHigh);
//radius, slices, stacks
glut.glutSolidSphere(0.07, 16, 16);
gl.glPopMatrix();

gl.glPushMatrix();
gl.glTranslatef(-0.14f, 1.56f, 0.24f);
gl.glRotatef(0.0f, 1.0f, 0.0f, 0.0f);
gl.glMaterialfv(GL.GL_FRONT, GL.GL_AMBIENT, ambientRed, 0);
gl.glMaterialfv(GL.GL_FRONT, GL.GL_DIFFUSE, diffuseRed, 0);
gl.glMaterialfv(GL.GL_FRONT, GL.GL_SPECULAR, specularRed, 0);
gl.glMaterialf(GL.GL_FRONT, GL.GL_SHININESS, shineHigh);
//radius, slices, stacks
glut.glutSolidSphere(0.07, 16, 16);
gl.glPopMatrix();

//right eye
gl.glPushMatrix();
gl.glTranslatef(0.14f, 1.56f, 0.24f);
gl.glRotatef(0.0f, 1.0f, 0.0f, 0.0f);
gl.glMaterialfv(GL.GL_FRONT, GL.GL_AMBIENT, ambientRed, 0);
gl.glMaterialfv(GL.GL_FRONT, GL.GL_DIFFUSE, diffuseRed, 0);
gl.glMaterialfv(GL.GL_FRONT, GL.GL_SPECULAR, specularRed, 0);
gl.glMaterialf(GL.GL_FRONT, GL.GL_SHININESS, shineHigh);
glut.glutSolidSphere(0.07, 16, 16);
gl.glPopMatrix();

gl.glPushMatrix();
```

```
gl.glTranslatef(0.16f, 1.56f, 0.24f);
gl.glRotatef(0.0f, 1.0f, 0.0f, 0.0f);
gl.glMaterialfv(GL.GL_FRONT, GL.GL_AMBIENT, ambientRed, 0);
gl.glMaterialfv(GL.GL_FRONT, GL.GL_DIFFUSE, diffuseRed, 0);
gl.glMaterialfv(GL.GL_FRONT, GL.GL_SPECULAR, specularRed, 0);
gl.glMaterialf(GL.GL_FRONT, GL.GL_SHININESS, shineHigh);
glut.glutSolidSphere(0.07, 16, 16);
gl.glPopMatrix();



// head - side attachment left
gl.glPushMatrix();
gl.glTranslatef(-0.46f, 2.02f, 0.0f);
gl.glRotatef(90.0f, 1.0f, 0.0f, 0.0f);
// radius, height, slices, stacks ..
gl.glMaterialfv(GL.GL_FRONT, GL.GL_AMBIENT, ambientGrey, 0);
gl.glMaterialfv(GL.GL_FRONT, GL.GL_DIFFUSE, diffuseGrey, 0);
gl.glMaterialfv(GL.GL_FRONT, GL.GL_SPECULAR, specularGrey, 0);
gl.glMaterialf(GL.GL_FRONT, GL.GL_SHININESS, shineHigh);
glut.glutSolidCylinder(0.08, 0.7, 16, 1);
gl.glPopMatrix();

// head - side attachment right
gl.glPushMatrix();
gl.glTranslatef(0.46f, 2.02f, 0.0f);
gl.glRotatef(90.0f, 1.0f, 0.0f, 0.0f);
// radius, height, slices, stacks ..
gl.glMaterialfv(GL.GL_FRONT, GL.GL_AMBIENT, ambientGrey, 0);
gl.glMaterialfv(GL.GL_FRONT, GL.GL_DIFFUSE, diffuseGrey, 0);
gl.glMaterialfv(GL.GL_FRONT, GL.GL_SPECULAR, specularGrey, 0);
gl.glMaterialf(GL.GL_FRONT, GL.GL_SHININESS, shineHigh);
```

```
glut.glutSolidCylinder(0.08, 0.7, 16, 1);
gl.glPopMatrix();

// head - forehead shape
gl.glPushMatrix();
gl.glTranslatef(0.0f, 1.89f, 0.28f);
gl.glRotatef(45.0f, 0.0f, 0.0f, 1.0f);
glut.glutSolidCube(0.08f);
gl.glPopMatrix();

gl.glPushMatrix();
gl.glTranslatef(0.0f, 1.85f, 0.28f);
gl.glRotatef(45.0f, 0.0f, 0.0f, 1.0f);
glut.glutSolidCube(0.1f);
gl.glPopMatrix();

gl.glPushMatrix();
gl.glTranslatef(0.0f, 1.81f, 0.28f);
gl.glRotatef(45.0f, 0.0f, 0.0f, 1.0f);
glut.glutSolidCube(0.1f);
gl.glPopMatrix();

gl.glPushMatrix();
gl.glTranslatef(0.0f, 1.76f, 0.28f);
gl.glRotatef(45.0f, 0.0f, 0.0f, 1.0f);
glut.glutSolidCube(0.1f);
gl.glPopMatrix();

gl.glPushMatrix();
gl.glTranslatef(0.05f, 1.76f, 0.28f);
gl.glRotatef(45.0f, 0.0f, 0.0f, 1.0f);
glut.glutSolidCube(0.1f);
```

```
gl.glPopMatrix();


gl.glPushMatrix();
gl.glTranslatef(-0.05f, 1.76f, 0.28f);
gl.glRotatef(45.0f, 0.0f, 0.0f, 1.0f);
glut.glutSolidCube(0.1f);
gl.glPopMatrix();


gl.glPushMatrix();
gl.glTranslatef(0.0f, 1.71f, 0.28f);
gl.glRotatef(45.0f, 0.0f, 0.0f, 1.0f);
glut.glutSolidCube(0.1f);
gl.glPopMatrix();




//The following code is for sections of the box that the robot is
//encased in.


//box - back
gl.glBegin(GL.GL_QUADS);
//top left, top right, bottom right, bottom left of quad
gl.glVertex3f(-2.00f, 2.20f, -2.00f);
gl.glVertex3f(2.00f, 2.20f, -2.00f);
gl.glVertex3f(2.00f, -2.80f, -2.00f);
gl.glVertex3f(-2.00f, -2.80f, -2.00f);
gl.glMaterialfv(GL.GL_FRONT, GL.GL_AMBIENT, ambientRed, 0);
gl.glMaterialfv(GL.GL_FRONT, GL.GL_DIFFUSE, diffuseRed, 0);
gl.glMaterialfv(GL.GL_FRONT, GL.GL_SPECULAR, specularRed, 0);
gl.glMaterialf(GL.GL_FRONT, GL.GL_SHININESS, shineHigh);


//box - floor
```

```
//top left, top right, bottom right, bottom left of quad
gl.glVertex3f(2.00f, -2.80f, -2.00f);
gl.glVertex3f(-2.00f, -2.80f, -2.00f);
gl.glVertex3f(-2.00f, -2.80f, 2.00f);
gl.glVertex3f(2.00f, -2.80f, 2.00f);
gl.glMaterialfv(GL.GL_FRONT, GL.GL_AMBIENT, ambientRed, 0);
gl.glMaterialfv(GL.GL_FRONT, GL.GL_DIFFUSE, diffuseRed, 0);
gl.glMaterialfv(GL.GL_FRONT, GL.GL_SPECULAR, specularRed, 0);
gl.glMaterialf(GL.GL_FRONT, GL.GL_SHININESS, shineHigh);

//box - roof
//top left, top right, bottom right, bottom left of quad
gl.glVertex3f(2.00f, 2.20f, -2.00f);
gl.glVertex3f(-2.00f, 2.20f, -2.00f);
gl.glVertex3f(-2.00f, 2.20f, 2.00f);
gl.glVertex3f(2.00f, 2.20f, 2.00f);
gl.glMaterialfv(GL.GL_FRONT, GL.GL_AMBIENT, ambientRed, 0);
gl.glMaterialfv(GL.GL_FRONT, GL.GL_DIFFUSE, diffuseRed, 0);
gl.glMaterialfv(GL.GL_FRONT, GL.GL_SPECULAR, specularRed, 0);
gl.glMaterialf(GL.GL_FRONT, GL.GL_SHININESS, shineHigh);

//box - left
//top left, top right, bottom right, bottom left of quad
gl.glVertex3f(-2.00f, 2.20f, -2.00f);
gl.glVertex3f(-2.00f, -2.80f, -2.00f);
gl.glVertex3f(-2.00f, -2.80f, 2.00f);
gl.glVertex3f(-2.00f, 2.20f, 2.00f);
gl.glMaterialfv(GL.GL_FRONT, GL.GL_AMBIENT, ambientRed, 0);
gl.glMaterialfv(GL.GL_FRONT, GL.GL_DIFFUSE, diffuseRed, 0);
gl.glMaterialfv(GL.GL_FRONT, GL.GL_SPECULAR, specularRed, 0);
gl.glMaterialf(GL.GL_FRONT, GL.GL_SHININESS, shineHigh);
```

```
//box - right
//top left, top right, bottom right, bottom left of quad
gl.glVertex3f(2.00f, 2.20f, -2.00f);
gl.glVertex3f(2.00f, -2.80f, -2.00f);
gl.glVertex3f(2.00f, -2.80f, 2.00f);
gl.glVertex3f(2.00f, 2.20f, 2.00f);
gl.glMaterialfv(GL.GL_FRONT, GL.GL_AMBIENT, ambientRed, 0);
gl.glMaterialfv(GL.GL_FRONT, GL.GL_DIFFUSE, diffuseRed, 0);
gl.glMaterialfv(GL.GL_FRONT, GL.GL_SPECULAR, specularRed, 0);
gl.glMaterialf(GL.GL_FRONT, GL.GL_SHININESS, shineHigh);



// front - lower
//top left, top right, bottom right, bottom left of quad
gl.glVertex3f(-2.00f, -1.60f, 2.00f);
gl.glVertex3f(2.00f, -1.60f, 2.00f);
gl.glVertex3f(2.00f, -2.80f, 2.00f);
gl.glVertex3f(-2.00f, -2.80f, 2.00f);
gl.glMaterialfv(GL.GL_FRONT, GL.GL_AMBIENT, ambientRed, 0);
gl.glMaterialfv(GL.GL_FRONT, GL.GL_DIFFUSE, diffuseRed, 0);
gl.glMaterialfv(GL.GL_FRONT, GL.GL_SPECULAR, specularRed, 0);
gl.glMaterialf(GL.GL_FRONT, GL.GL_SHININESS, shineHigh);

// front - left
//top left, top right, bottom right, bottom left of quad
gl.glVertex3f(-2.00f, 2.20f, 2.00f);
gl.glVertex3f(-1.20f, 2.20f, 2.00f);
gl.glVertex3f(-1.20f, -2.00f, 2.00f);
gl.glVertex3f(-2.00f, -2.00f, 2.00f);
gl.glMaterialfv(GL.GL_FRONT, GL.GL_AMBIENT, ambientRed, 0);
gl.glMaterialfv(GL.GL_FRONT, GL.GL_DIFFUSE, diffuseRed, 0);
```

```
gl.glMaterialfv(GL.GL_FRONT, GL.GL_SPECULAR, specularRed, 0);
gl.glMaterialf(GL.GL_FRONT, GL.GL_SHININESS, shineHigh);

//front - right
//top left, top right, bottom right, bottom left of quad
gl.glVertex3f(2.00f, 2.20f, 2.00f);
gl.glVertex3f(1.20f, 2.20f, 2.00f);
gl.glVertex3f(1.20f, -2.00f, 2.00f);
gl.glVertex3f(2.00f, -2.00f, 2.00f);
gl.glMaterialfv(GL.GL_FRONT, GL.GL_AMBIENT, ambientRed, 0);
gl.glMaterialfv(GL.GL_FRONT, GL.GL_DIFFUSE, diffuseRed, 0);
gl.glMaterialfv(GL.GL_FRONT, GL.GL_SPECULAR, specularRed, 0);
gl.glMaterialf(GL.GL_FRONT, GL.GL_SHININESS, shineHigh);

//front - upper
//top left, top right, bottom right, bottom left of quad
gl.glVertex3f(-1.20f, 2.20f, 2.00f);
gl.glVertex3f(1.20f, 2.20f, 2.00f);
gl.glVertex3f(1.20f, 1.80f, 2.00f);
gl.glVertex3f(-1.20f, 1.80f, 2.00f);
gl.glMaterialfv(GL.GL_FRONT, GL.GL_AMBIENT, ambientRed, 0);
gl.glMaterialfv(GL.GL_FRONT, GL.GL_DIFFUSE, diffuseRed, 0);
gl.glMaterialfv(GL.GL_FRONT, GL.GL_SPECULAR, specularRed, 0);
gl.glMaterialf(GL.GL_FRONT, GL.GL_SHININESS, shineHigh);

//front - see-through part
//top left, top right, bottom right, bottom left of quad
gl.glVertex3f(-0.80f, 1.80f, 2.00f);
gl.glVertex3f(0.80f, -0.50f, 2.00f);
gl.glVertex3f(0.80f, -0.50f, 2.00f);
gl.glVertex3f(-0.80f, 1.80f, 2.00f);
gl.glEnable(GL.GL_BLEND);
```

```
gl.glBlendFunc(GL.GL_SRC_ALPHA, GL.GL_ONE_MINUS_SRC_ALPHA);
gl.glEnd();
//empties buffers that are implemented by gl
gl.glFlush();




//battle orb animation
if (rotateAnimate.isAnimating()) {
    //sets the value of rotation to add frameRotation onto it
    rotation += frameRotation;
    x1 = 0.3f * (float) Math.cos(DEG2RAD * rotation);
    x2 = 0.3f * (float) Math.sin(DEG2RAD * rotation);
    x3 = 4.0f * (float) Math.cos(DEG2RAD * rotation);
    x4 = 4.0f * (float) Math.sin(DEG2RAD * rotation);
    //made use of the rotating square from the square rotating program
    //order to make it look like the robot has some sort of rotating
    //inner parts.
    gl.glBegin(GL.GL_POLYGON);
    gl.glVertex2f(x1, x2);
    gl.glVertex2f(-x2, x1);
    gl.glVertex2f(-x1, -x2);
    gl.glVertex2f(x2, -x1);
    gl.glColor3f(0.6f, 0.2f, 0.4f);
    gl.glEnd();

    //orb #1
    gl.glPushMatrix();
    gl.glTranslatef(x3, x4, 0.0f);
    gl.glRotatef(45.0f, 0.0f, 1.0f, 0.0f);
    gl.glMaterialfv(GL.GL_FRONT, GL.GL_AMBIENT, ambientGold, 0);
    gl.glMaterialfv(GL.GL_FRONT, GL.GL_DIFFUSE, diffuseGold, 0);
```

```
            gl.glMaterialfv(GL.GL_FRONT, GL.GL_SPECULAR, specularGold, 0);
            gl.glMaterialf(GL.GL_FRONT, GL.GL_SHININESS, shineHigh);
            glut.glutSolidSphere(0.34, 24, 24);
            gl.glPopMatrix();

            // orb #2
            gl.glPushMatrix();
            //gl.glTranslatef(0.10f, 1.56f, 0.0f);
            gl.glTranslatef(x4, x3, 0.0f);
            gl.glRotatef(45.0f, 0.0f, 1.0f, 0.0f);
            gl.glMaterialfv(GL.GL_FRONT, GL.GL_AMBIENT, ambientGold, 0);
            gl.glMaterialfv(GL.GL_FRONT, GL.GL_DIFFUSE, diffuseGold, 0);
            gl.glMaterialfv(GL.GL_FRONT, GL.GL_SPECULAR, specularGold, 0);
            gl.glMaterialf(GL.GL_FRONT, GL.GL_SHININESS, shineHigh);
            //size
            glut.glutSolidSphere(0.34, 16, 16);
            gl.glPopMatrix();
        }
    }
}

public void reshape(GLAutoDrawable drawable, int x, int y, int width, int height) {
    GL gl = drawable.getGL();
    //
    gl.glViewport(0, 0, width, height);
    gl.glMatrixMode(GL.GL_PROJECTION);
    gl.glLoadIdentity();
    if (perspective) {
        glu.gluPerspective(60.0, (float) width / (float) height, 0.1, 80.0);
    } else {
        if (width <= height) {
```

```
                gl.glOrtho(-2.5, 2.5, -2.5 * (float) height / (float) width, 2.5 * (float) height
/ (float) width, -10.0, 10.0);

            } else {

                gl.glOrtho(-2.5 * (float) width / (float) height, 2.5 * (float) width / (float)
height, -2.5, 2.5, -10.0, 10.0);

            }

        }

        gl.glMatrixMode(GL.GL_MODELVIEW);

        reset = true;

    }


    public void displayChanged(GLAutoDrawable drawable, boolean modeChanged,

            boolean deviceChanged) {

    }


    private void resetViewer() {

        for (int i = 0; i < 3; ++i) {

            viewer[i] = 0.0f;

        }

        viewerAngle = 0.0f;

        adjustHorizontalView(0.0f);

        adjustVerticalView(0.0f);

        adjustZoom(0.0f);

    }


    //adjustHorizontalView takes a float value called change

    private void adjustHorizontalView(float change) {

        //this float value is added onto the value of the viewer angle

        viewerAngle += change;

        //math.cos and math.sin are used in order to rotate

        viewer[2] = 8.0f * (float) Math.cos(viewerAngle * DEG2RAD);

        viewer[0] = 8.0f * (float) Math.sin(viewerAngle * DEG2RAD);

        //redraw
```

```
        repaint();
    }
    //adjustVerticalView also takes a float value called change
    private void adjustVerticalView(float change) {
        //this float value is also added onto the viewer angle as before
        viewerAngle += change;
        //math.cos is used in order to rotate
        viewer[1] = 8.0f * (float) Math.cos(viewerAngle * DEG2RAD);
        //redraw
        repaint();
    }

    //adjustZoom takes a float value called change as well
    private void adjustZoom(float change) {
        //this float value is also added onto the viewer angle as before
        viewerAngle += change;
        //math.cos is used in order to rotate
        viewer[2] = 8.0f * (float) Math.cos(viewerAngle * DEG2RAD);
        //redraw
        repaint();
    }

    public void lightsOff(GL gl) {
        //turns the lighting off but doesn't make it pitch black
        float ambientGlobal[] = {-0.5f, -0.5f, -0.5f, -0.5f};
        gl.glLightModelfv(GL.GL_LIGHT_MODEL_AMBIENT, ambientGlobal, 0);
        //disables the lights
        gl.glDisable(GL.GL_LIGHTING);
        gl.glDisable(GL.GL_LIGHT0);
        gl.glDisable(GL.GL_LIGHT1);
        gl.glDisable(GL.GL_LIGHT2);
    }
```

```
private void lightsOn(GL gl) {
    //sets the ambient light to a brightness that allows the user to see.
    float ambientGlobal[] = {0.5f, 0.5f, 0.5f, 1.0f};
    gl.glLightModelfv(GL.GL_LIGHT_MODEL_AMBIENT, ambientGlobal, 0);
    //enables the lights
    gl.glEnable(GL.GL_LIGHTING);
    gl.glEnable(GL.GL_LIGHT0);
    gl.glEnable(GL.GL_LIGHT1);
    gl.glEnable(GL.GL_LIGHT2);
}

private void switchCamera(int view) {
    //switch case statement used to switch views.
    switch (view) {
        case 1:
            viewer[0] = 8.0f * (float) Math.cos(viewerAngle);
            viewer[1] = (float) Math.sin(viewerAngle);
            return;
        case 2:
            viewer[1] = 8.0f * (float) Math.cos(viewerAngle);
            viewer[0] = (float) Math.sin(viewerAngle);
            return;
        case 3:
            viewer[2] = 8.0f * (float) Math.sin(viewerAngle);
            viewer[0] = (float) Math.cos(viewerAngle);
            return;
        case 4:
            viewer[0] = 8.0f / (float) Math.cos(-viewerAngle);
            viewer[1] = (float) Math.sin(-viewerAngle);
            return;
    }
```

```
        //redraw
        repaint();
    }

    private class KeyResponder extends KeyAdapter {
        //various keyboard key presses for interaction

        @Override
        public void keyPressed(KeyEvent key) {
            switch (key.getKeyCode()) {
                case KeyEvent.VK_ESCAPE:
                    System.exit(0);
                    break;
                case KeyEvent.VK_RIGHT:
                    adjustHorizontalView(2.0f);
                    break;
                case KeyEvent.VK_LEFT:
                    adjustHorizontalView(-2.0f);
                    break;
                case KeyEvent.VK_UP:
                    adjustVerticalView(2.0f);
                    break;
                case KeyEvent.VK_DOWN:
                    adjustVerticalView(-2.0f);
                    break;

                case KeyEvent.VK_R:
                    reset = true;
                    break;
                case KeyEvent.VK_L:
                    sceneLights = false;
                    System.out.print("off -");
```

```
            break;
        case KeyEvent.VK_K:
            sceneLights = true;
            System.out.print("on -");
            break;
        case KeyEvent.VK_SHIFT:
            if (rotateAnimate.isAnimating()) {
                rotateAnimate.stop();
                System.out.print("stop -");
            } else {
                rotateAnimate.start();
                System.out.print("start -");
            }
            break;
        case KeyEvent.VK_ENTER:
            //reverses direction when enter is pressed.
            frameRotation = -frameRotation;
            break;
        case KeyEvent.VK_1:
            switchCamera(1);
            break;
        case KeyEvent.VK_2:
            switchCamera(2);
            break;
        case KeyEvent.VK_3:
            switchCamera(3);
            break;
        case KeyEvent.VK_4:
            switchCamera(4);
            break;
    }
    repaint();
```

```
        }
    }

    private class MouseResponder extends MouseAdapter {
        //various mouse events for interaction

        @Override
        public void mousePressed(MouseEvent e) {
            switch (e.getButton()) {
                case MouseEvent.BUTTON1:
                    //left mouse button
                    adjustZoom(1.0f);
                    break;
                case MouseEvent.BUTTON3:
                    //right mouse button
                    adjustZoom(-1.0f);
                    break;
            }
        }
    }

    public static void main(String[] args) {
        GLJPanel canvas = new Robot();
        //JFrame properties
        JFrame frame = new JFrame("3D Graphics Coursework #1");
        frame.setSize(812, 614);
        frame.setLocationRelativeTo(null); // centre of screen
        frame.add(canvas);
        frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        frame.setVisible(true);
        canvas.requestFocusInWindow();
        //Instructions and Commands for the program
```

```
        System.out.println("Usage:\n LEFT and RIGHT arrows rotate the scene horizontally\n "
                + "UP and DOWN arrows rotate the scene vertically"
                + "LEFT MOUSE CLICK zooms in, RIGHT MOUSE CLICK zooms out"
                + "\n NOTE: if you zoom in and pass through the robot to the other side"
                + "and continue to zoom in, it will start zooming out."
                + "\n NUMBERS 1 to 4 switch between camera views "
                + "\n the L key turns the lighting off, the K key turns it on"
                + "\n press SHIFT to turn battle orbs on/off"
                + "\n the ENTER key reverses the way each battle orb rotates"
                + "\n\n R to reset\n ESC to quit \n");
    }
}
```