NetID: jtt3923
Name: Junda TU

EECS 351 Project A Report

Goal

Design and draw two different 3D parts which have two or more sequential, moving joints.

User-guide

Open the html file, there are one sequential rectangles, which models the posture of swimming, and a planet system, where three small balls spin around a big ball.

1. Press buttons on the web page to change the rotating speed or make the graphic stop.
2. Use mouse to drag the graphic to observe graphic with different view.
3. Stop the graphic quickly through press “Space” on the keyboard.

Result

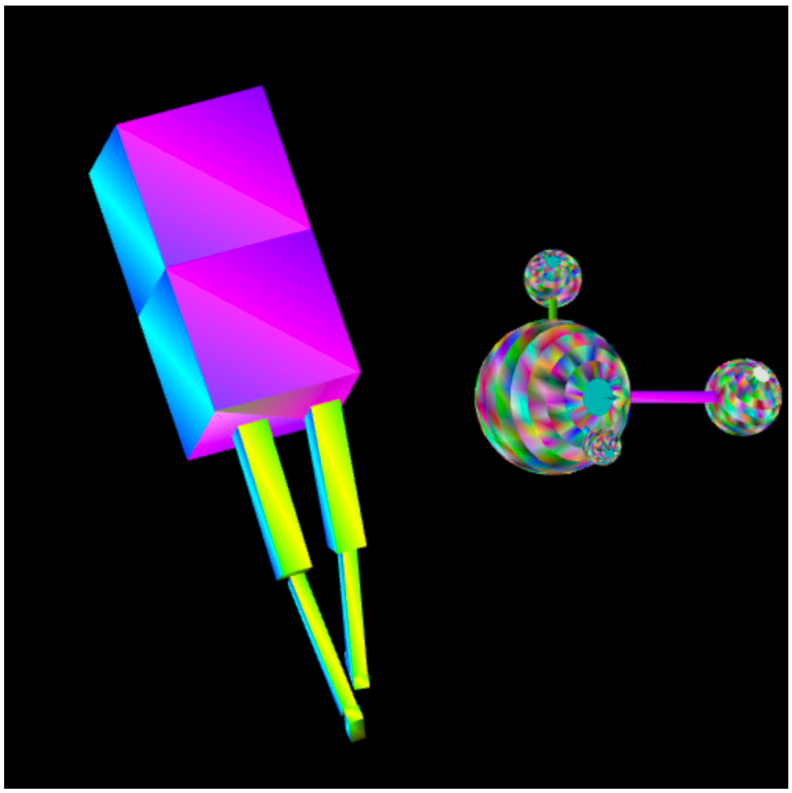

Figure 1

NetID: jtt3923
Name: Junda TU

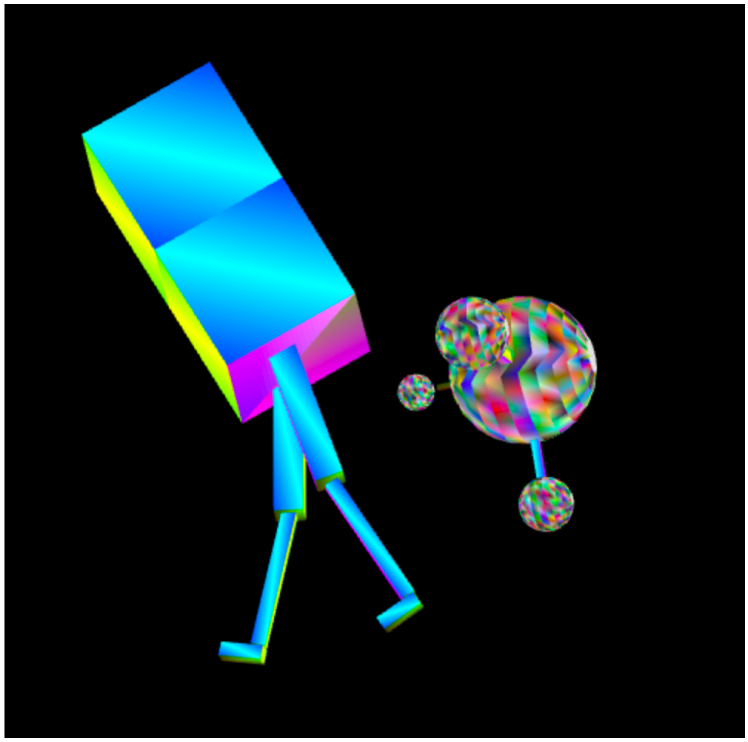

Figure 2

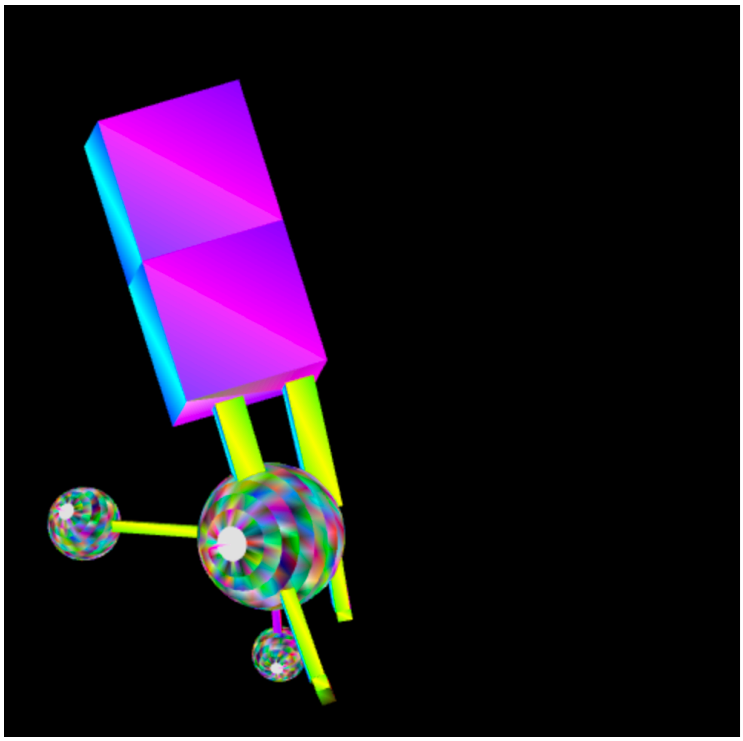

Figure 3

NetID: jtt3923
Name: Junda TU

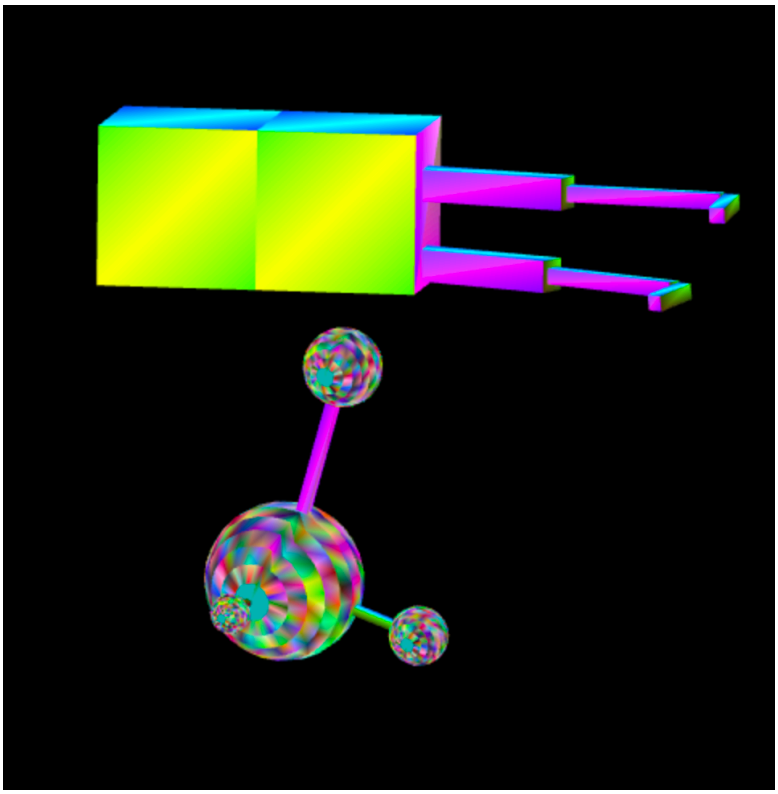

Figure 4

NetID: jtt3923
Name: Junda TU

Sketch of program's scene-graph

```
  modelMatrix.setTranslate(-0.5, 0.5, 0);  // 'set' means DISCARD old matrix,
              // (drawing axes centered in CVV), and then make new
              // drawing axes moved to the lower-left corner of CVV.
  modelMatrix.scale(1,1,-1);
  modelMatrix.rotate(15, 0, 0, 1);
  modelMatrix.rotate(15, 0, 1, 0);
  modelMatrix.rotate(-15, 1, 0, 0);
  var dist = Math.sqrt(xMdragTot*xMdragTot + yMdragTot*yMdragTot);
  //             // why add 0.001? avoids divide-by-zero in next statement
  //             // in cases where user didn't drag the mouse.)
  modelMatrix.rotate(dist*120.0, -yMdragTot+0.0001, xMdragTot+0.0001, 0.0);
  // modelMatrix.
              // convert to left-handed coord sys
                                    // to match WebGL display canvas.
  modelMatrix.scale(0.2, 0.2, 0.2);
              // Make it smaller:
  modelMatrix.rotate(0, 0, 0, 1);  // Spin on XY diagonal axis
  modelMatrix.translate(-1, -1, -1);
//-------------------------
//--------mouse interaction
//-------------------------
  // var dist = Math.sqrt(xMdragTot*xMdragTot + yMdragTot*yMdragTot);
              // why add 0.001? avoids divide-by-zero in next statement
              // in cases where user didn't drag the mouse.)
  // modelMatrix.rotate(dist*120.0, -yMdragTot+0.0001, xMdragTot+0.0001, 0.0);
  // DRAW CUBE:   Use ths matrix to transform & draw
  //            the second set of vertices stored in our VBO:
  gl.uniformMatrix4fv(u_ModelMatrix, false, modelMatrix.elements);
      // Draw just the first set of vertices: start at vertex SHAPE_0_SIZE
  gl.drawArrays(gl.TRIANGLES, recStart,36);
// body two
  modelMatrix.translate(0, -2, 0);  // 'set' means DISCARD old matrix,
  gl.uniformMatrix4fv(u_ModelMatrix, false, modelMatrix.elements);
      // Draw just the first set of vertices: start at vertex SHAPE_0_SIZE
  gl.drawArrays(gl.TRIANGLES, recStart/floatsPerVertex,36);
  //thigh
// one of the legs in order to easily go back to draw another thigh push it first
//push to stack first
  pushMatrix(modelMatrix);
  /////////
  modelMatrix.translate(0.5, 0, 1);
  modelMatrix.rotate(180, 1, 0, 0);
  modelMatrix.rotate(currentAngle*0.15+20, 1, 0, 0);
  modelMatrix.scale(0.2, 1, 0.2);
  // modelMatrix.translate(-1, 0, -1);
  gl.uniformMatrix4fv(u_ModelMatrix, false, modelMatrix.elements);
      // Draw just the first set of vertices: start at vertex SHAPE_0_SIZE
  gl.drawArrays(gl.TRIANGLES, recStart/floatsPerVertex,36);
//draw the first calf
//the
```

Figure 5