```
1  /* $Id: Tutorial_02_01.java,v 1.2 2006/04/14 22:01:53 jbelcher Exp $ */
2  /**
3   * @author John Belcher
4   * @version $Revision: 1.0 $
5   */
6
7  package tealsim.physics.tutorials;
8  import java.awt.event.ActionEvent;
9  import javax.media.j3d.*;
10 import javax.vecmath.*;
11 import teal.framework.TealAction;
12 import teal.sim.collision.SphereCollisionController;
13 import teal.sim.physical.EMModel;
14 import teal.sim.physical.Wall;
15 import teal.sim.physical.em.RingOfCurrent;
16 import teal.sim.simulation.Sim3D;
17 import teal.util.TDebug;
18
19 public class Tutorial_02_01 extends Sim3D {
20
21     private static final long serialVersionUID = 3257008735204554035L;
22
23     RingOfCurrent floatingCoil;
24     Vector3d floatingCoilPos;
25     double ringRad = 0.43;
26     double torR = 0.08;
27     double ringMass = 3.5;
28
29     public Tutorial_02_01() {
30         super();
31
32         TDebug.setGlobalLevel(0);
33
34         title = "Tutorial_02_01";
35
36         ///// Set properties on the SimModel /////
37         // Bounding area represents the size of the simulation
38         // setDeltaTime() sets the time step of the simulation.
39         // setDamping() sets the damping in the system.
40         EMModel emmodel = new EMModel();
41         setModel(emmodel);
42         BoundingSphere bs = new BoundingSphere(new Point3d(0, 1.6, 0), 03.5);
43         theModel.setBoundingArea(bs);
44         theModel.setDeltaTime(0.02); // Was 0.005
45         ((EMModel)theModel).setDamping(0.);
46         mViewer.setBoundingArea(bs);
47
48         floatingCoil = new RingOfCurrent();
49         floatingCoil.setID("Ring");
50         floatingCoil.setDirection(new Vector3d(0., 1., 0.));
51         floatingCoilPos = new Vector3d(0., 1.25, 0.);
52         floatingCoil.setPickable(true);
53         floatingCoil.setRotable(true);
54         floatingCoil.setMoveable(true);
55         floatingCoil.setInducing(false);
56         floatingCoil.setRadius(ringRad);
57         floatingCoil.setTorusRadius(torR);
```

```
58          floatingCoil.setMass(ringMass);
59          floatingCoil.setInducing(false);
60          floatingCoil.setInductance(0.1);
61
62          // Here we add a collisionController to coil so that it will be
63          // registered as a colliding object when it touches the "wall".
64          // We represent the coil as a sphere for the purposes of collision
65          // detection, since we're only concerned with stopping the coil from
66          // "falling through the floor"
67              // SphereCollisionController constructor.
68
69          SphereCollisionController sccx =
70              new SphereCollisionController(floatingCoil);
71          sccx.setRadius(torR);
72          sccx.setTolerance(0.01);
73          floatingCoil.setColliding(true);
74          floatingCoil.setCollisionController(sccx);
75          addElement(floatingCoil);
76
77          // We create a "wall" that the floating coil will interact with
78
79          // Wall constructor.
80          Wall wall = new Wall(new Vector3d(0., 0, 0.),
81                    new Vector3d(2., 0., 0.), new Vector3d(0., 0., 2.));
82          wall.setElasticity(1.);
83          addElement(wall);
84
85          // set paramters for mouseScale
86
87          Vector3d mouseScale = mViewer.getVpTranslateScale();
88          mouseScale.x *= 0.05;
89          mouseScale.y *= 0.05;
90          mouseScale.z *= 0.5;
91          mViewer.setVpTranslateScale(mouseScale);
92
93          mSMC.init();
94          resetCamera();
95          // addAction for pulldown menus on TEALsim windows
96          addActions();
97          reset();
98      }
99
100     void addActions() {
101         TealAction ta = new TealAction("Tutorial_02_01", this);
102         addAction("Help", ta);
103     }
104
105     public void actionPerformed(ActionEvent e) {
106         TDebug.println(1, " Action comamnd: " + e.getActionCommand());
107         if (e.getActionCommand().compareToIgnoreCase("Tutorial_02_01") == 0)
108         {
109             mFramework.openBrowser("resources/help/tutorial_02_01.html");
110         }  else {
111             super.actionPerformed(e);
112         }
113     }
114
```

```
115     public void reset() {
116         floatingCoil.setPosition(floatingCoilPos);
117             theModel.requestRefresh();
118     }
119
120     public void resetCamera() {
121         mViewer.setLookAt(new Point3d(0.0, 0.025, 0.4),
122                   new Point3d(0., 0.025, 0.), new Vector3d(0., 1., 0.));
123     }
124 }
```