

```
1  /* $Id: Tutorial_02_04.java,v 1.1 2006/04/12 21:39:26 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 java.beans.PropertyChangeEvent;
10 import javax.media.j3d.*;
11 import javax.vecmath.*;
12 import teal.framework.TealAction;
13 import teal.plot.PlotProperties;
14 import teal.plot.Graph;
15 import teal.render.Rendered;
16 import teal.sim.collision.SphereCollisionController;
17 import teal.sim.physical.EMModel;
18 import teal.sim.physical.Wall;
19 import teal.sim.physical.em.MagneticDipole;
20 import teal.sim.physical.em.RingOfCurrent;
21 import teal.render.j3d.*;
22 import teal.sim.simulation.Sim3D;
23 import teal.ui.control.*;
24 import teal.util.TDebug;
25
26 public class Tutorial_02_04 extends Sim3D {
27
28     private static final long serialVersionUID = 3257008735204554035L;
29
30     Rendered nativeObject = new Rendered();
31     ShapeNode ShapeNodeNative = new ShapeNode();
32     PropertyDouble currentSlider = new PropertyDouble();
33     RingOfCurrent floatingCoil;
34     Vector3d floatingCoilPos;
35     MagneticDipole magDipole;
36     Graph position_graph;
37     PlotProperties position_plot;
38     double ringRad = 0.43;
39     double torR = 0.08;
40     double ringMass = 3.5;
41     double current = -50.;
42     double magLen = 0.24;
43     double magR = 0.09;
44     double searchRad = magR;
45     double fLen = 0.033;
46     double minD = 0.03;
47     int kMax = 300; //300
48     double friction = 0.;
49
50     public Tutorial_02_04() {
51         super();
52
53         TDebug.setGlobalLevel(0);
54
55         title = "Tutorial_02_04";
56
57         ///// Set properties on the SimModel /////
```

```

58         // Bounding area represents the characteristic size of the space.
59         // setDeltaTime() sets the time step of the simulation.
60         // setDamping() sets the damping on the system.
61         EMModel emmodel = new EMModel();
62         setModel(emmodel);
63         BoundingSphere bs = new BoundingSphere(new Point3d(0, 1.6, 0), 03.5);
64         theModel.setBoundingArea(bs);
65         theModel.setDeltaTime(0.005);
66         ((EMModel)theModel).setDamping(friction);
67         mViewer.setBoundingArea(bs);
68
69         magDipole = new MagneticDipole();
70         magDipole.setMu(10.);
71         magDipole.setPosition(new Vector3d(0., 0., 0.));
72         magDipole.setDirection(new Vector3d(0, 1, 0));
73         magDipole.setPickable(false);
74         magDipole.setRotable(false);
75         magDipole.setMoveable(false);
76         magDipole.setRadius(magR);
77         magDipole.setLength(magLen);
78         addElement(magDipole);
79
80         floatingCoil = new RingOfCurrent();
81         floatingCoil.setID("Ring");
82         floatingCoil.setDirection(new Vector3d(0., 1., 0.));
83         floatingCoil.setPickable(true);
84         floatingCoil.setRotable(true);
85         floatingCoil.setMoveable(true);
86         floatingCoil.setInducing(false);
87         floatingCoil.setRadius(ringRad);
88         floatingCoil.setTorusRadius(torR);
89         floatingCoil.setMass(ringMass);
90         floatingCoil.setInducing(false);
91         floatingCoil.setInductance(0.1);
92         floatingCoil.setCurrent(current);
93         // Here we add a collisionController to the RingOfCurrent
94         //so that it will be registered as a colliding object, and
95         // react appropriately when it touches the Wall.
96         SphereCollisionController sccx =
97             new SphereCollisionController(floatingCoil);
98         sccx.setRadius(torR);
99         sccx.setTolerance(0.01);
100        floatingCoil.setColliding(true);
101        floatingCoil.setCollisionController(sccx);
102        floatingCoilPos = new Vector3d(0.,
103            1.25+ ringRad + (ringRad * 0.02), 0.);
104        addElement(floatingCoil);
105
106        // We create a wall that the floating coil sits on.
107        // Wall constructor.
108        Wall wall = new Wall(new Vector3d(0., 0, 0.),
109            new Vector3d(2., 0., 0.), new Vector3d(0., 0., 2.));
110        wall.setElasticity(1.);
111        addElement(wall);
112
113        // create the sliders to control the amount of current
114

```

```
115     currentSlider.setText("I");
116     currentSlider.setMinimum(-100);
117     currentSlider.setMaximum(100);
118     currentSlider.setPaintTicks(true);
119     currentSlider.addPropertyChangeListener("value", this);
120     currentSlider.setValue(current);
121     currentSlider.setVisible(true);
122
123     // add the slider to a control group and add
124
125     ControlGroup controls = new ControlGroup();
126     controls.setText("Parameters");
127     controls.add(currentSlider);
128     addElement(controls);
129
130     // set paramters for mouseScale
131
132     Vector3d mouseScale = mViewer.getVpTranslateScale();
133     mouseScale.x *= 0.05;
134     mouseScale.y *= 0.05;
135     mouseScale.z *= 0.5;
136     mViewer.setVpTranslateScale(mouseScale);
137
138     mSMC.init();
139     resetCamera();
140     // addAction for pulldown menus on TEALsim windows
141     addAction();
142     reset();
143
144 }
145
146
147 void addAction() {
148     TealAction ta = new TealAction("Tutorial_02_04", this);
149     addAction("Help", ta);
150 }
151
152 public void actionPerformed(ActionEvent e) {
153     TDebug.println(1, " Action comamnd: " + e.getActionCommand());
154     if (e.getActionCommand().compareToIgnoreCase("Tutorial_02_04") == 0) {
155         mFramework.openBrowser("resources/help/tutorial_02_04.html");
156     } else {
157         super.actionPerformed(e);
158     }
159 }
160
161 public void reset() {
162     floatingCoil.setPosition(floatingCoilPos);
163     floatingCoil.setVelocity(new Vector3d());
164     floatingCoil.setDirection(new Vector3d(0., 1., 0.));
165     ((EMModel)theModel).setDamping(friction);
166     currentSlider.setValue(current);
167     theModel.requestRefresh();
168 }
169
170 public void resetCamera() {
171     mViewer.setLookAt(new Point3d(0.0, 0.025, 0.4),
```

```
172         new Point3d(0., 0.025, 0.), new Vector3d(0., 1., 0.));
173     }
174
175     public void propertyChange(PropertyChangeEvent pce) {
176         Object source = pce.getSource();
177         if (source == currentSlider) {
178             current = ((Double) pce.getNewValue()).doubleValue();
179             floatingCoil.setCurrent(current);
180         } else {
181             super.propertyChange(pce);
182         }
183     }
184
185 }
```