```
/* $Id: Tutorial_02_02.java,v 1.2 2006/04/14 22:01:53 jbelcher Exp $ */
/**
 * @author John Belcher
 * @version $Revision: 1.0 $
 */

package tealsim.physics.tutorials;
import java.awt.event.ActionEvent;
import javax.media.j3d.*;
import javax.vecmath.*;

import teal.framework.TealAction;
import teal.sim.collision.SphereCollisionController;
import teal.sim.physical.EMModel;
import teal.sim.physical.Wall;
import teal.sim.physical.em.RingOfCurrent;
import teal.sim.simulation.Sim3D;
import teal.util.TDebug;
import teal.ui.control.ControlGroup;
import teal.ui.control.PropertyDouble;
import java.beans.PropertyChangeEvent;
import teal.plot.PlotProperties;
import teal.plot.Graph;

public class Tutorial_02_02 extends Sim3D {

    private static final long serialVersionUID = 3257008735204554035L;

    PropertyDouble frictionSlider = new PropertyDouble();
    Graph position_graph;
    PlotProperties position_plot;
    double friction;
    RingOfCurrent floatingCoil;
    Vector3d floatingCoilPos;
    double ringRad = 0.43;
    double torR = 0.08;
    double ringMass = 3.5;

    public Tutorial_02_02() {
        super();
        TDebug.setGlobalLevel(0);
        title = "Tutorial_02_02";
        ///// Set properties on the SimModel /////
        // Bounding area represents the size of the simulation
        // setDeltaTime() sets the time step of the simulation.
        // setDamping() sets the damping in the system.
        EMModel emmodel = new EMModel();
        setModel(emmodel);
        BoundingSphere bs = new BoundingSphere(new Point3d(0, 1.6, 0), 03.5);
        theModel.setBoundingArea(bs);
        theModel.setDeltaTime(0.02); // Was 0.005
        ((EMModel)theModel).setDamping(0.);
        mViewer.setBoundingArea(bs);

        floatingCoil = new RingOfCurrent();
        floatingCoil.setID("Ring");
        floatingCoil.setDirection(new Vector3d(0., 1., 0.));
```

```
58          floatingCoilPos = new Vector3d(0., 1.25, 0.);
59          floatingCoil.setPickable(true);
60          floatingCoil.setRotable(true);
61          floatingCoil.setMoveable(true);
62          floatingCoil.setInducing(false);
63          floatingCoil.setRadius(ringRad);
64          floatingCoil.setTorusRadius(torR);
65          floatingCoil.setMass(ringMass);
66          floatingCoil.setInducing(false);
67          floatingCoil.setInductance(0.1);
68
69          // Here we add a collisionController to coil so that it will be
70          // registered as a colliding object when it touches the "wall".
71          // We represent the coil as a sphere for the purposes of collision
72          // detection, since we're only concerned with stopping the coil from
73          // "falling through the floor"
74              // SphereCollisionController constructor.
75
76          SphereCollisionController sccx =
77              new SphereCollisionController(floatingCoil);
78          sccx.setRadius(torR);
79          sccx.setTolerance(0.01);
80          floatingCoil.setColliding(true);
81          floatingCoil.setCollisionController(sccx);
82          addElement(floatingCoil);
83
84          // We create a "wall" that the floating coil will interact with
85
86          // Wall constructor.
87          Wall wall = new Wall(new Vector3d(0., 0, 0.),
88                    new Vector3d(2., 0., 0.), new Vector3d(0., 0., 2.));
89          wall.setElasticity(1.);
90          addElement(wall);
91
92          // create the sliders to control the amount of friction in the model
93          frictionSlider.setText("Friction");
94          frictionSlider.setMinimum(0.);
95          frictionSlider.setMaximum(2.0);
96          frictionSlider.setPaintTicks(true);
97          frictionSlider.addPropertyChangeListener("value", this);
98          frictionSlider.setValue(0.0);
99          frictionSlider.setVisible(true);
100
101         // add the slider to a control group and add this to the scene
102
103         ControlGroup controls = new ControlGroup();
104         controls.setText("Parameters");
105         controls.add(frictionSlider);
106         addElement(controls);
107
108         // Create a graph of the height of the coil, and add it to the GUI.
109         // This involves creating a graph, adding a "plot" (which defines the
110         // quantities being plotted), and adding it in its own Control Group.
111
112         // Graph constructor.
113         position_graph = new Graph();
114         position_graph.setSize(150, 400);
```

```
        position_graph.setXRange(0., 6.);
        position_graph.setYRange(-0., 2.);
        position_graph.setWrap(true);
        position_graph.setClearOnWrap(true);
        position_graph.setXLabel("Time");
        position_graph.setYLabel("position");
        // Here we create the PlotItem being drawn by this graph.
        // We want to plot the y-position of the RingOfCurrent versus time,
        // so we use PlotProperties
        position_plot = new PlotProperties();
        position_plot.setObjectX(theModel);
        position_plot.setPropertyX("time");
        position_plot.setObjectY(floatingCoil);
        position_plot.setPropertyY("y");
        // adds the supplied PlotItem to the graph.
        position_graph.addPlotItem(position_plot);

        // Here we create a new Control Group for the graph
        ControlGroup graphPanel = new ControlGroup();
        graphPanel.setText("Graphs");
        graphPanel.addElement(position_graph);
        addElement(graphPanel);

        // set paramters for mouseScale

        Vector3d mouseScale = mViewer.getVpTranslateScale();
        mouseScale.x *= 0.05;
        mouseScale.y *= 0.05;
        mouseScale.z *= 0.5;
        mViewer.setVpTranslateScale(mouseScale);

        mSMC.init();
        resetCamera();
        // addAction for pulldown menus on TEALsim windows
        addActions();
        reset();
    }

    void addActions() {
        TealAction ta = new TealAction("Tutorial_02_02", this);
        addAction("Help", ta);
    }

    public void actionPerformed(ActionEvent e) {
        TDebug.println(1, " Action comamnd: " + e.getActionCommand());
        if (e.getActionCommand().compareToIgnoreCase("Tutorial_02_02") == 0)
        {
            mFramework.openBrowser("resources/help/tutorial_02_02.html");
        }  else {
            super.actionPerformed(e);
        }
    }

    public void reset() {
        floatingCoil.setPosition(floatingCoilPos);
        floatingCoil.setVelocity(new Vector3d(0.,0.,0.));
        position_graph.clear(0);
```

```
172         ((EMModel)theModel).setDamping(0.);
173         frictionSlider.setValue(0.);
174             theModel.requestRefresh();
175     }
176 
177     public void resetCamera() {
178         mViewer.setLookAt(new Point3d(0.0, 0.025, 0.4),
179                   new Point3d(0., 0.025, 0.), new Vector3d(0., 1., 0.));
180     }
181 
182     public void propertyChange(PropertyChangeEvent pce) {
183         Object source = pce.getSource();
184         if (source == frictionSlider) {
185             friction = ((Double) pce.getNewValue()).doubleValue();
186             ((EMModel)theModel).setDamping(friction);
187         } else {
188             super.propertyChange(pce);
189         }
190     }
191 }
```