

```
1  /* $Id: Tutorial_02_03.java,v 1.3 2006/04/14 00:10:23 jbelcher Exp $ */
2  /**
3   * @author John Belcher - Department of Physics / MIT
4   * @version $Revision: 1.0 $
5   */
6
7  package tealsim.physics.tutorials;
8  import java.awt.*;
9  import java.awt.event.ActionEvent;
10 import java.beans.PropertyChangeEvent;
11 import java.io.FileNotFoundException;
12 import javax.media.j3d.*;
13 import javax.vecmath.*;
14 import com.sun.j3d.loaders.IncorrectFormatException;
15 import com.sun.j3d.loaders.ParseException;
16 import com.sun.j3d.loaders.Scene;
17 import com.sun.j3d.loaders.objectfile.ObjectFile;
18 import teal.framework.TealAction;
19 import teal.render.Rendered;
20 import teal.render.geometry.Cylinder;
21 import teal.render.geometry.Sphere;
22 import teal.render.j3d.*;
23 import teal.render.j3d.loaders.Loader3DS;
24 import teal.sim.simulation.Sim3D;
25 import teal.ui.control.*;
26 import teal.util.TDebug;
27
28 public class Tutorial_02_03 extends Sim3D {
29
30     private static final long serialVersionUID = 3257008735204554035L;
31
32     Rendered nativeObject01 = new Rendered();
33     ShapeNode ShapeNodeNative01 = new ShapeNode();
34     Rendered nativeObject02 = new Rendered();
35     ShapeNode ShapeNodeNative02 = new ShapeNode();
36     PropertyDouble posSlider01 = new PropertyDouble();
37     PropertyDouble posSlider02 = new PropertyDouble();
38
39     public Tutorial_02_03() {
40         super();
41
42         TDebug.setGlobalLevel(0);
43
44         title = "Tutorial_02_03";
45
46         // create two objects using teal.render.geometry
47         // and add them to the scene
48
49         ShapeNodeNative01.setGeometry(Cylinder.makeGeometry(32, 2., 0.001));
50         nativeObject01.setNode3D(ShapeNodeNative01);
51         nativeObject01.setColor(new Color(255, 0, 0));
52         nativeObject01.setPosition(new Vector3d(0,0.,0.));
53         addElement(nativeObject01);
54
55         ShapeNodeNative02.setGeometry(Sphere.makeGeometry(16,0.5));
56         nativeObject02.setNode3D(ShapeNodeNative02);
57         nativeObject02.setColor(new Color(0, 255, 0));
```

```
58 nativeObject02.setPosition(new Vector3d(0, 2, 0));
59 addElement(nativeObject02);
60
61 // import two .3DS files objects using Loader3DS
62 // The conversion between max units and Java3D units
63 // is 1 Java3D unit = 1 Max inch
64
65 double scale3DS = 0.01; // overall scale factor for .3DS objects
66
67 Loader3DS max = new Loader3DS();
68
69 BranchGroup bg01 =
70     max.getBranchGroup("resources/models/geoSphere.3DS",
71         "resources/models/");
72 Node3D node01 = new Node3D();
73 node01.setScale(scale3DS);
74 node01.addContents(bg01);
75
76 Rendered importedObject01 = new Rendered();
77 importedObject01.setNode3D(node01);
78 importedObject01.setPosition(new Vector3d(0., 0., 0.));
79 addElement(importedObject01);
80
81 BranchGroup bg02 =
82     max.getBranchGroup("resources/models/cone.3DS", "resources/models/");
83 Node3D node02 = new Node3D();
84 node02.setScale(scale3DS);
85 node02.addContents(bg02);
86
87 Rendered importedObject02 = new Rendered();
88 importedObject02.setNode3D(node02);
89 importedObject02.setPosition(new Vector3d(0., 0., 0.));
90 addElement(importedObject02);
91
92 // import an .obj file object using the ObjectFile loader
93
94 double scaleObj = 2.0; // this is an overall scale Obj objects
95 Scene scene = null;
96     ObjectFile f = new ObjectFile ();
97     f.setFlags
98         (ObjectFile.RESIZE
99             | ObjectFile.TRIANGULATE | ObjectFile.STRIPIFY);
100 try {
101     scene = f.load("./src/java/simulations/resources/models/box.obj");
102 } catch (FileNotFoundException e) {
103     e.printStackTrace();
104 } catch (IncorrectFormatException e) {
105     e.printStackTrace();
106 } catch (ParseException e) {
107     e.printStackTrace();
108 }
109
110
111 BranchGroup bg03 = scene.getSceneGroup();
112 Node3D node03 = new Node3D();
113 node03.setScale(scaleObj);
114 node03.addContents(bg03);
```

```
115     Rendered importedObject03 = new Rendered();
116     importedObject03.setNode3D(node03);
117     importedObject03.setPosition(new Vector3d(0., -.5, 0.));
118     addElement(importedObject03);
119
120     // create the two sliders for the two native objects
121
122     posSlider01.setText("Disk Position ");
123     posSlider01.setMinimum(-3.);
124     posSlider01.setMaximum(3.0);
125     posSlider01.setPaintTicks(true);
126     posSlider01.addPropertyChangeListener("value", this);
127     posSlider01.setValue(-1.);
128     posSlider01.setVisible(true);
129
130     posSlider02.setText("Sphere Position");
131     posSlider02.setMinimum(-3.);
132     posSlider02.setMaximum(3.0);
133     posSlider02.setPaintTicks(true);
134     posSlider02.addPropertyChangeListener("value", this);
135     posSlider02.setValue(2.0);
136     posSlider02.setVisible(true);
137
138     // add the sliders to control groups and add those to the scene
139
140     ControlGroup controls01 = new ControlGroup();
141     controls01.setText("Parameters01");
142     controls01.add(posSlider01);
143     addElement(controls01);
144
145     ControlGroup controls02 = new ControlGroup();
146     controls02.setText("Parameters02");
147     controls02.add(posSlider02);
148     addElement(controls02);
149
150     // set paramters for mouseScale
151
152     Vector3d mouseScale = mViewer.getVpTranslateScale();
153     mouseScale.x *= 0.05;
154     mouseScale.y *= 0.05;
155     mouseScale.z *= 0.5;
156     mViewer.setVpTranslateScale(mouseScale);
157
158     mSMC.init();
159     resetCamera();
160     // addAction for pulldown menus on TEALsim windows
161     addActions();
162
163 }
164
165 void addActions() {
166     TealAction ta = new TealAction("Tutorial_02_03", this);
167     addAction("Help", ta);
168 }
169
170 public void actionPerformed(ActionEvent e) {
```

```
172         TDebug.println(1, " Action comamnd: " + e.getActionCommand());
173         if (e.getActionCommand().compareToIgnoreCase("Tutorial_02_03") == 0) {
174             mFramework.openBrowser("resources/help/tutorial_02_03.html");
175         } else {
176             super.actionPerformed(e);
177         }
178     }
179
180     public void reset() {
181     }
182
183     public void resetCamera() {
184         mViewer.setLookAt(new Point3d(0.0, 0.025, 0.4),
185             new Point3d(0., 0.025, 0.), new Vector3d(0., 1., 0.));
186     }
187
188     public void propertyChange(PropertyChangeEvent pce) {
189         Object source = pce.getSource();
190         if (source == posSlider01) {
191             double posV01 = ((Double) pce.getNewValue()).doubleValue();
192             nativeObject01.setNode3D(ShapeNodeNative01);
193             nativeObject01.setPosition(new Vector3d(0., posV01, 0.));
194         } else
195         if (source == posSlider02) {
196             double posV02 = ((Double) pce.getNewValue()).doubleValue();
197             nativeObject02.setNode3D(ShapeNodeNative02);
198             nativeObject02.setPosition(new Vector3d(0., posV02, 0.));
199         } else {
200             super.propertyChange(pce);
201         }
202     }
203
204 }
```