```
1 /**
 2 settingsWindow.java
 3 This file is used to create/open the window in which the user can edit the settings.
 4 **/
 6 import org.eclipse.swt.widgets.Display;
 7 import org.eclipse.swt.widgets.Shell;
 8 import org.eclipse.swt.widgets.Button;
 9 import org.eclipse.swt.widgets.ColorDialog;
10
11 import java.awt.FileDialog;
12 import java.io.*;
13 import java.io.IOException;
14 import java.nio.file.Files;
15 import java.nio.file.Paths;
16
17 import org.eclipse.core.runtime.Path;
18 import org.eclipse.swt.SWT;
19 import org.eclipse.swt.layout.GridLayout;
20 import org.eclipse.swt.widgets.Text;
21 import org.eclipse.swt.widgets.Label;
22 import org.eclipse.swt.layout.GridData;
23 import org.eclipse.swt.widgets.Scale;
24 import org.eclipse.swt.events.SelectionAdapter;
25 import org.eclipse.swt.events.SelectionEvent;
26 import org.eclipse.swt.graphics.Color;
27 import org.eclipse.swt.events.MouseAdapter;
28 import org.eclipse.swt.events.MouseEvent;
29
30 public class settingsWindow {
31
```

```
32
      public Settings currentSettings = new Settings();
33
      protected Shell shlSettings;
34
      private Text textSaveName;
35
36
      /**
37
       * Launch the application.
38
       * @param args
39
       */
40
      public static void main(String[] args) {
41
          try {
42
               settingsWindow window = new settingsWindow();
43
              Settings dummySettings = new Settings();
44
              window.open(dummySettings);
45
          } catch (Exception e) {
              e.printStackTrace();
46
47
          }
48
      }
49
50
      /**
51
       * Open the window.
52
       */
53
      public void open(Settings mainWindowSettings) {
54
          Display display = Display.getDefault();
55
          Shell shlSettings = new Shell();
56
          shlSettings.setSize(758, 272);
57
          shlSettings.setText("Settings");
58
          shlSettings.setLayout(new GridLayout(3, false));
59
60
          currentSettings = mainWindowSettings;
61
62
          Button btnShowAxes = new Button(shlSettings, SWT.CHECK);
```

```
63
          btnShowAxes.setSelection(currentSettings.showAxes);
64
          btnShowAxes.addSelectionListener(new SelectionAdapter() {
65
              @Override
66
              public void widgetSelected(SelectionEvent e) {
                  currentSettings.showAxes = btnShowAxes.getSelection();
67
68
69
          });
70
          btnShowAxes.setText("Show axes");
71
72
          Label lblInequalityTransparency = new Label(shlSettings, SWT.NONE);
73
          lblInequalityTransparency.setLayoutData(new GridData(SWT.RIGHT, SWT.CENTER, false, false,
  1, 1));
74
          lblInequalityTransparency.setText("Inequality Transparency");
75
76
          Scale scaleTransparency = new Scale(shlSettings, SWT.NONE);
77
          scaleTransparency.setEnabled(false);
78
          scaleTransparency.addMouseListener(new MouseAdapter() {
79
              @Override
80
              public void mouseUp(MouseEvent e) {
81
                  currentSettings.inequalityTransparency = scaleTransparency.getIncrement()/100.0;
82
              }
83
          }):
84
          scaleTransparency.setLayoutData(new GridData(SWT.LEFT, SWT.CENTER, true, false, 1, 1));
85
86
          Button btnShowGridlines = new Button(shlSettings, SWT.CHECK);
87
          btnShowGridlines.setEnabled(false);
88
          btnShowGridlines.setSelection(currentSettings.showGridlines);
89
          btnShowGridlines.addSelectionListener(new SelectionAdapter() {
90
              @Override
91
              public void widgetSelected(SelectionEvent e) {
                  currentSettings.showGridlines = btnShowGridlines.getSelection();
92
```

```
93
94
           });
95
           btnShowGridlines.setText("Show gridlines");
96
97
           Button btnPickBackgroundColour = new Button(shlSettings, SWT.NONE);
           btnPickBackgroundColour.setLayoutData(new GridData(SWT.FILL, SWT.CENTER, false, false, 1,
98
   1));
99
           btnPickBackgroundColour.addSelectionListener(new SelectionAdapter() {
               @Override
100
101
               public void widgetSelected(SelectionEvent e) {
102
                   ColorDialog newColourPicker = new ColorDialog(shlSettings);
103
                   newColourPicker.open();
104
                   currentSettings.backgroundColor = new Color(Display.getCurrent(),
   newColourPicker.getRGB());
105
               }
           });
106
107
           btnPickBackgroundColour.setText("Pick Background Colour");
108
109
           Button btnPickPenColour = new Button(shlSettings, SWT.NONE);
110
           btnPickPenColour.addSelectionListener(new SelectionAdapter() {
111
               @Override
112
               public void widgetSelected(SelectionEvent e) {
                   ColorDialog newColourPicker = new ColorDialog(shlSettings);
113
114
                   newColourPicker.open():
115
                   currentSettings.penColor = new Color(Display.getCurrent(),
   newColourPicker.getRGB());
116
117
           }):
118
           btnPickPenColour.setText("Pick Pen Colour");
119
           Button btnLabelAxes = new Button(shlSettings, SWT.CHECK);
120
```

```
121
           btnLabelAxes.setEnabled(false);
122
           btnLabelAxes.setGrayed(true);
123
           btnLabelAxes.addSelectionListener(new SelectionAdapter() {
124
               @Override
125
               public void widgetSelected(SelectionEvent e) {
126
                   currentSettings.labelAxes = btnLabelAxes.getSelection();
127
128
           });
129
           btnLabelAxes.setText("Label axes");
130
131
           Button btnOpenFile = new Button(shlSettings, SWT.NONE);
132
           btnOpenFile.addSelectionListener(new SelectionAdapter() {
133
               @Override
134
               public void widgetSelected(SelectionEvent e) {
135
                   try {
136
                       org.eclipse.swt.widgets.FileDialog newFilePicker = new
   org.eclipse.swt.widgets.FileDialog(shlSettings, FileDialog.LOAD);
137
                       newFilePicker.open();
138
                       newFilePicker.setFilterExtensions(new String[] { "*.qwf", "*.*" });
139
140
                       String fileName = newFilePicker.getFilterPath() + "\\" +
   newFilePicker.getFileName():
141
                       //Reads the contents of the file at the given location into a string.
142
                       //Note that this does not need to be paired with a 'close' statement.
143
                       String fileContents = Files.readString(Paths.get(fileName));
144
145
                        currentSettings.currentEquations += "\n" + fileContents;
146
                   } catch (IOException ioException) {
147
                       ioException.printStackTrace();
148
               }
149
```

```
150
           });
151
           btnOpenFile.setLayoutData(new GridData(SWT.FILL, SWT.CENTER, false, false, 1, 1));
152
           btnOpenFile.setText("Open file");
153
           Button btnSaveToFile = new Button(shlSettings, SWT.NONE);
154
155
           btnSaveToFile.addSelectionListener(new SelectionAdapter() {
156
               @Override
157
               public void widgetSelected(SelectionEvent e) {
158
                   org.eclipse.swt.widgets.MessageBox saveWarningDialog = new
   org.eclipse.swt.widgets.MessageBox(shlSettings, SWT.OK | SWT.CANCEL);
159
                   saveWarningDialog.setText("Save/Overwrite");
                   saveWarningDialog.setMessage("Warning: This will overwrite any file named " +
160
   textSaveName.getText() + " in the specified folder.");
161
                   saveWarningDialog.open();
162
                   try {
163
                       org.eclipse.swt.widgets.DirectoryDialog newFolderPicker = new
   org.eclipse.swt.widgets.DirectoryDialog(shlSettings, SWT.SAVE);
164
                       File location =
   Path.fromOSString(newFolderPicker.open()).makeAbsolute().toFile();
165
                       String fileName = location.toString() + "\\" + textSaveName.getText();
166
                       if (currentSettings.debugMode) {
167
                            System.out.println("Writing to file "+fileName);
168
                            System.out.println("Equations to be written:
   "+currentSettings currentEquations);
169
170
                       PrintWriter printWriter = new PrintWriter(fileName);
171
                       printWriter.println(currentSettings.currentEquations);
172
                       printWriter.close():
173
                   } catch (Exception exceptionInstance) {
174
                       exceptionInstance.printStackTrace();
175
```

```
176
177
           });
           btnSaveToFile.setText("Pick folder and save/overwrite");
178
179
180
           Button btnLabelIntercepts = new Button(shlSettings, SWT.CHECK);
181
           btnLabelIntercepts.setEnabled(false):
182
           btnLabelIntercepts.setGrayed(true);
           btnLabelIntercepts.addSelectionListener(new SelectionAdapter() {
183
184
               @Override
185
               public void widgetSelected(SelectionEvent e) {
186
                   currentSettings.labelIntersections = btnLabelIntercepts.getSelection();
187
               }
188
           }):
           btnLabelIntercepts.setText("Label intercepts with axes");
189
190
191
           Label lblFileName = new Label(shlSettings, SWT.NONE);
192
           lblFileName.setLayoutData(new GridData(SWT.RIGHT, SWT.CENTER, false, false, 1, 1));
193
           lblFileName.setText("File name:");
194
195
           textSaveName = new Text(shlSettings, SWT.BORDER);
196
           textSaveName.setText("Graph.gwf");
           GridData gd textSaveName = new GridData(SWT. LEFT. SWT. CENTER. true. false. 1. 1):
197
198
           gd textSaveName.widthHint = 143;
199
           textSaveName.setLayoutData(gd textSaveName);
200
201
           Button btnLabelIntersections = new Button(shlSettings, SWT.CHECK);
202
           btnLabelIntersections.setEnabled(false);
203
           btnLabelIntersections.setGraved(true):
204
           btnLabelIntersections.addSelectionListener(new SelectionAdapter() {
               @Override
205
               public void widgetSelected(SelectionEvent e) {
206
```

```
207
                   currentSettings.labelIntersections = btnLabelIntersections.getSelection();
208
           });
209
           btnLabelIntersections.setText("Label intersections on curves");
210
211
           new Label(shlSettings, SWT.NONE);
           new Label(shlSettings, SWT.NONE);
212
213
214
           shlSettings.open();
215
           shlSettings.layout();
           while (!shlSettings.isDisposed()) {
216
               if (!display.readAndDispatch()) {
217
                   display.sleep();
218
219
220
           }
221
       }
222 }
```