```java
package catan;

import java.awt.BorderLayout;

/**
 * This is the window holding all of the help stuff.
 *
 * @author Nicole Downer
 * @author Eddie Gurnee
 * @version 0.0.06 11/16/2013
 * @since 11/12/2013
 */
public class HelpFrame extends JFrame {
    private final int WIDTH = 600;
    private final int HEIGHT = 400;

    private int helpIndex = 0;

    private ArrayList<String> helpData;
    private ArrayList<String> foundHelp;
    private ArrayList<String> refinedHelp;

    private boolean refined;

    JTextArea discoveredHelp = new JTextArea("");
    JButton nextBtn = new JButton("Next Search Result");
    JButton lastBtn = new JButton("Previous Search Result");
    JPanel bottomBtns = new JPanel();

    public HelpFrame() {
        super("Settler's of Catan - Help");

        this.setResizable(false);
        setSize(WIDTH, HEIGHT);
        setLocationRelativeTo(null);
        setDefaultCloseOperation(JFrame.DISPOSE_ON_CLOSE);
        getContentPane().setBackground(Color.ORANGE);
        setLayout(new BorderLayout(5, 5));

        JPanel rightSide = new JPanel();
        rightSide.setLayout(new BoxLayout(rightSide, BoxLayout.Y_AXIS));
        add(rightSide, BorderLayout.EAST);

        JButton loadHelp = new JButton("Reload Help");
        JButton searchHelp = new JButton("Search Help");
        JButton browseHelp = new JButton("Browse Help");
        loadHelp.addActionListener(new LoadHelpListener());
        searchHelp.addActionListener(new SearchHelpListener());
        browseHelp.addActionListener(new BrowseHelpListener());

```

```
67          rightSide.add(searchHelp);
68          rightSide.add(browseHelp);
69          rightSide.add(loadHelp);
70
71          this.helpData = new ArrayList<String>();
72
73          this.load();
74
75          JPanel leftSide = new JPanel();
76          leftSide.setLayout(new BorderLayout());
77          add(leftSide, BorderLayout.WEST);
78          leftSide.add(discoveredHelp, BorderLayout.CENTER);
79          this.discoveredHelp.setTabSize(3);
80
81          nextBtn.addActionListener(new ScrollHelpListener(true));
82          lastBtn.addActionListener(new ScrollHelpListener(false));
83          // add(discoveredHelp, BorderLayout.WEST);
84
85          add(this.bottomBtns, BorderLayout.SOUTH);
86          bottomBtns.add(lastBtn);
87          bottomBtns.add(nextBtn);
88      }
89
90      /**
91       * @author Nicole Downer
92       * @author Eddie Gurnee
93       * @version 0.0.03 11/17/2013
94       * @since 11/14/2013
95       */
96      private void load() {
97          Scanner infile = null;
98          try {
99              infile = new Scanner(new FileInputStream(new File("GLOSSARY.txt")));
100             while (infile.hasNextLine()) {
101                 String line = infile.nextLine();
102                 int start = line.indexOf("<!BS@>");
103
104                 if (start > -1) {
105                     String data = "";
106
107                     while (true) {
108                         String temp = infile.nextLine() + "\n";
109                         int end = temp.indexOf("</!BS@>");
110                         if (end == -1) {
111                             data += temp;
112                         } else {
113                             break;
114                         }
115                     }
116                     helpData.add(data);
```

```
117                 }
118             }
119         } catch (FileNotFoundException ex) {
120             ex.printStackTrace();
121         } finally {
122             infile.close();
123         }
124     }
125 
126     /**
127      * @author Nicole Downer
128      * @author Eddie Gurnee
129      * @param searchTerm
130      * @version 0.0.02 11/16/2013
131      * @since 11/15/2013
132      */
133     private void search(String searchTerm) {
134         searchTerm = searchTerm.toLowerCase();
135         this.foundHelp = new ArrayList<String>();
136         this.refinedHelp = new ArrayList<String>();
137         this.refined = false;
138 
139         this.helpIndex = 0;
140         for (String s : this.helpData) {
141             String tempStr = s.toLowerCase();
142             if (tempStr.indexOf(searchTerm) > -1) {
143                 this.foundHelp.add(s);
144             }
145         }
146         if (this.foundHelp.size() == 0) {
147             this.foundHelp.add("Your search term was not found.");
148         }
149         this.discoveredHelp.setText(foundHelp.get(this.helpIndex));
150         if (this.foundHelp.size() > 5) {
151             int n = JOptionPane
152                     .showConfirmDialog(
153                             null,
154                             "There are "
155                                     + foundHelp.size()
156                                     + " results found."
157                                     + "\nWould you like to refine your search with
    another term?",
158                             "Refine Search", JOptionPane.YES_NO_OPTION,
159                             JOptionPane.QUESTION_MESSAGE, null);
160             if (n == 0) {
161                 String term = (String)JOptionPane.showInputDialog(null,
162                         "Enter search term:", "Search",
163                         JOptionPane.PLAIN_MESSAGE);
164                 this.refined = true;
165                 this.refineSearch(term);
```

```
166             }
167         }
168     }
169 
170     /**
171      * @author Eddie Gurnee
172      * @author Nicole Downer
173      * @param searchTerm
174      * @version 0.0.01 11/17/2013
175      * @since 11/17/2013
176      */
177     private void refineSearch(String searchTerm) {
178         searchTerm = searchTerm.toLowerCase();
179         this.refinedHelp = new ArrayList<String>();
180         this.helpIndex = 0;
181         for (String s : this.foundHelp) {
182             String tempStr = s.toLowerCase();
183             if (tempStr.indexOf(searchTerm) > -1) {
184                 this.refinedHelp.add(s);
185             }
186         }
187         if (this.refinedHelp.size() == 0) {
188             this.refinedHelp.add("Your search term was not found.");
189         }
190         this.discoveredHelp.setText(this.refinedHelp.get(this.helpIndex));
191     }
192 
193     /**
194      * @author Eddie Gurnee
195      * @author Nicole Downer
196      * @version 0.0.03 11/16/2013
197      * @since 11/15/2013
198      */
199     private class LoadHelpListener implements ActionListener {
200 
201         @Override
202         public void actionPerformed(ActionEvent e) {
203             HelpFrame.this.load();
204         }
205     }
206     private class ScrollHelpListener implements ActionListener {
207         private boolean forward;
208 
209         private ScrollHelpListener(boolean forward) {
210             this.forward = forward;
211         }
212 
213         @Override
214         public void actionPerformed(ActionEvent e) {
215             if (forward) {
```

```
216                 if (!HelpFrame.this.refined) {
217                     if (HelpFrame.this.helpIndex < HelpFrame.this.foundHelp
218                             .size() - 1) {
219                         HelpFrame.this.helpIndex++;
220                         HelpFrame.this.discoveredHelp.setText(foundHelp
221                                 .get(HelpFrame.this.helpIndex));
222                     }
223                 } else {
224                     if (HelpFrame.this.helpIndex < HelpFrame.this.refinedHelp
225                             .size() - 1) {
226                         HelpFrame.this.helpIndex++;
227                         HelpFrame.this.discoveredHelp.setText(refinedHelp
228                                 .get(HelpFrame.this.helpIndex));
229                     }
230                 }
231             } else {
232                 if (!HelpFrame.this.refined) {
233                     if (HelpFrame.this.helpIndex > 0) {
234                         HelpFrame.this.helpIndex--;
235                         HelpFrame.this.discoveredHelp.setText(foundHelp
236                                 .get(HelpFrame.this.helpIndex));
237                     }
238                 } else {
239                     if (HelpFrame.this.helpIndex > 0) {
240                         HelpFrame.this.helpIndex--;
241                         HelpFrame.this.discoveredHelp.setText(refinedHelp
242                                 .get(HelpFrame.this.helpIndex));
243                     }
244                 }
245             }
246         }
247     }
248     private class SearchHelpListener implements ActionListener {
249 
250         @Override
251         public void actionPerformed(ActionEvent e) {
252             String term = (String)JOptionPane.showInputDialog(null,
253                     "Enter search term:", "Search", JOptionPane.PLAIN_MESSAGE);
254 
255             HelpFrame.this.search(term);
256         }
257     }
258     private class BrowseHelpListener implements ActionListener {
259 
260         @Override
261         public void actionPerformed(ActionEvent e) {
262             HelpFrame.this.search(" ");
263         }
264     }
265 }
```