

```
1 package view;
2
3 import com.google.common.base.Joiner;
4
5 import interfaces.ViewInterface;
6
7 import java.awt.Color;
8 import java.awt.FontMetrics;
9 import java.awt.Graphics;
10 import java.awt.Graphics2D;
11 import java.awt.RenderingHints;
12 import java.awt.event.ComponentEvent;
13 import java.awt.event.ComponentListener;
14 import java.util.ArrayList;
15 import java.util.Arrays;
16 import java.util.List;
17
18 import javax.swing.BorderFactory;
19 import javax.swing.JPanel;
20
21 public class TopText extends JPanel implements ComponentListener, ViewInterface {
22
23     private boolean resized = true;
24     private String[] textWords;
25     private final List<String> tokens;
26
27     /**
28      * Constructor for the welcome panel at the top of the gui
29      * @param text the text to display
30      */
31     public TopText(String text) {
32         setBorder(BorderFactory.createRaisedBevelBorder());
33         setDoubleBuffered(true);
34         setBackground(new Color(Integer.parseInt("303B5E", 16)));
35         textWords = text.split("\\s+");
36         tokens = Arrays.asList(textWords);
37         addComponentListener(this);
38     }
39
40     private void setWordsPerLine(Graphics2D graphics) {
41
42         // Calculate the length of the full string
43         FontMetrics fmet = graphics.getFontMetrics(graphics.getFont());
44         List<String> rows = new ArrayList<String>();
45
46         int textlen = (int) Math
47             .min(fmet.stringWidth(Joiner.on(' ').join(tokens)),
48                 getWidth() / 1.5);
49
50         // Get the words that should fit on each line
51         for (int i = 0; i < tokens.size(); i) {
52             List<String> words = new ArrayList<String>();
53             for (int genWidth = 0; i < tokens.size() && genWidth < textlen; i++) {
54                 genWidth += fmet.stringWidth(tokens.get(i));
55                 words.add(tokens.get(i));
56                 if (genWidth >= textlen || (i + 1 == tokens.size())) {
57                     rows.add(Joiner.on(' ').join(words));
58                 }
59             }
60         }
61         textWords = rows.toArray(new String[rows.size()]);
62     }
```

```

63
64     @Override
65     protected void paintComponent(Graphics graphic) {
66         super.paintComponent(graphic);
67
68         Graphics2D graphics = (Graphics2D) graphic;
69         if (resized) {
70             graphics.setFont(graphics.getFont().deriveFont(22.0f));
71             setWordsPerLine(graphics);
72             resized = false;
73         }
74         drawStringCentered(graphics, true);
75     }
76
77     private void drawStringCentered(Graphics2D graphics, boolean wrapLong) {
78         FontMetrics fmet = graphics.getFontMetrics(graphics.getFont());
79         graphics.setPaint(new Color(Integer.parseInt("09BD90", 16)));
80         graphics.setRenderingHint(RenderingHints.KEY_ANTIALIASING,
81             RenderingHints.VALUE_ANTIALIAS_ON);
82
83         if (wrapLong) {
84             int textHeight = fmet.getHeight() * textWords.length;
85             int heightStart = getHeight() / 2 - textHeight / 2
86                 + fmet.getHeight() / 2;
87
88             for (String line : textWords) {
89                 int width = fmet.stringWidth(line);
90                 graphics.drawString(line, getWidth() / 2 - width / 2,
91                     heightStart);
92                 heightStart += fmet.getHeight();
93             }
94         } else {
95             String line = Joiner.on(' ').join(textWords);
96             int width = fmet.stringWidth(line);
97             graphics.drawString(line, getWidth() / 2 - width / 2, getHeight()
98                 / 2 - fmet.getHeight() / 2);
99         }
100     }
101
102     @Override
103     public void componentHidden(ComponentEvent arg0) {
104         // TODO Auto-generated method stub
105     }
106
107     @Override
108     public void componentMoved(ComponentEvent arg0) {
109         // TODO Auto-generated method stub
110     }
111
112     @Override
113     public void componentResized(ComponentEvent arg0) {
114         resized = true;
115     }
116
117     @Override
118     public void componentShown(ComponentEvent arg0) {
119         // TODO Auto-generated method stub
120     }
121
122     @Override
123     public void initComponents() {
124         // TODO Auto-generated method stub

```

```
125     }  
126  
127     @Override  
128     public void showMessage(String msg, int msgType) throws UnsupportedOperationException {  
129         throw new UnsupportedOperationException();  
130     }  
131 }
```