

```
1 // (World, Actor, GreenfootImage, Greenfoot and MouseInfo)
2 import greenfoot.*;
3 import java.awt.*;
4
5 /**
6  * This class will be used to display messages to the
7  * game player such as instructions on how to play the game
8  * and messages when the level end or the game ends
9  *
10 * @author Derek Peacock
11 * @version 1.0 29/Jan/2015
12 */
13 public class MessageBox extends Actor
14 {
15     GreenfootImage image; // current Actor image
16
17     private static final int LEFT_MARGIN = 60;
18
19     private static final int LINE_HEIGHT = 20;
20
21     private int yPos = 80; // The current y position for drawing text
22
23     private Font baseFont;
24
25     /**
26      * Create a new MessageBox and scale the image to width
27      * by height (make sure the aspect ratio is preserved)
28      */
29     public MessageBox(int width, int height)
30     {
31         image = getImage();
32         image.scale(width, height);
33         baseFont = image.getFont();
34     }
35
36     private void drawHeading(String text)
37     {
38         Font largeFont = new Font("Arial", Font.BOLD, 24);
39         image.setFont(largeFont);
40         image.setColor(Color.BLUE);
41         drawLine(text);
42
43         image.setFont(baseFont);
44         image.setColor(Color.BLACK);
45         yPos = yPos + LINE_HEIGHT;
46     }
47
48     /**
49      * Draw one line of text (message) on top of the current
50      * image starting at the left margin and current yPos
51      * Increase yPos ready for the next line of text.
52      */
53     private void drawLine(String message)
```

```
54     {
55         image.drawString(message, LEFT_MARGIN, yPos);
56         yPos = yPos + LINE_HEIGHT;
57     }
58
59     private void drawLines(String [] messages)
60     {
61         for(String message : messages)
62         {
63             drawLine(message);
64         }
65
66         yPos = yPos + LINE_HEIGHT;
67     }
68
69     /**
70      *
71      */
72     private void explainGame ()
73     {
74         drawHeading("Game Description");
75
76         String [] lines =
77         {
78             "In this game the crab moves around ",
79             "in order to eat worm. An otter will however ",
80             "chase the crab if it can see the crab."
81         };
82
83         drawLines(lines);
84     }
85
86     private void explainScoring ()
87     {
88         drawHeading("Game Scoring");
89
90         String [] lines =
91         {
92             "10 points: Each worm eaten in 10 seconds",
93             "-1 point: For every second over 10 taken to eat worm",
94             "All points lost on level if eaten by otter"
95         };
96
97         drawLines(lines);
98     }
99
100
101     private void showKeys ()
102     {
103         drawHeading("Game Keys");
104
105         String [] lines =
106         {
```

```
107         "Move Left: Left Arrow Key",
108         "Move Right: Right Arrow Key",
109         "Move Up: Up Arrow Key",
110         "Move Down: Down Arrow Key",
111         "Pause: Space Bar",
112         "Abandon Game: Escape Key"
113     };
114
115     drawLines(lines);
116 }
117
118
119 /**
120  * This method will show the user a summary of how
121  * the game works, it will display what keys to use
122  * and it will explain how the scoring system works
123  */
124 public void showInstructions()
125 {
126     explainGame();
127     showKeys();
128     explainScoring();
129 }
130 }
131
```