

```
1 import java.awt.*;
2 import javax.swing.*;
3 import java.awt.event.*;
4 import java.util.*;
5 import java.io.*;
6 import javax.imageio.*;
7 import java.awt.image.*;
8 /*****
9  * The GUI for the Game class.
10  *
11  * @author Ben Parsell
12  * @version 1.0.0 12/2/2015
13  *****/
14 public class GUI extends JFrame implements ActionListener{
15     /** instantiate new game */
16     Game g;
17
18     /** instantiate Direction Buttons */
19     JButton north;
20     JButton south;
21     JButton east;
22     JButton west;
23     JButton downstairs;
24     JButton upstairs;
25
26     /** instantiate Action Buttons */
27     JButton look;
28     JButton help;
29     JButton pickup;
30     JButton drop;
31     JButton eat;
32     JButton list;
33     JButton fire;
34     JButton continueButton;
35
36     /** instantiate Text area */
37     JTextArea results;
38
39     /** instantiate Menu and items */
40     JMenuBar menus;
41     JMenu fileMenu;
42     JMenuItem quitItem;
43     JMenuItem newGame;
44
45     /*****
46     * Main method
47     *****/
48     public static void main(String [] args) {
49         // Instantiate and Initialize new GUI
```

```
50     GUI gui = new GUI();
51
52     // Shutdown program when exxit
53     gui.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
54
55     // Final settings
56     gui.setTitle("WW2");
57     gui.pack();
58     gui.setVisible(true);
59
60 }
61
62 /*****
63  * GUI constructor which handles positioning and
64  * initializing variables
65  *****/
66 public GUI() {
67     // Initialize a new game
68     g = new Game();
69
70     // Initialize Gridbag
71     setLayout(new GridBagLayout());
72     GridBagConstraints loc = new GridBagConstraints();
73
74     // Sets up the text area as "results"
75     results = new JTextArea(25,60);
76     JScrollPane scrollPane = new JScrollPane(results);
77     loc.gridx = 0;
78     loc.gridy = 1;
79     loc.gridheight = 12;
80     loc.gridwidth = 10;
81     loc.insets.left = 20;
82     loc.insets.right = 20;
83     loc.insets.bottom = 5;
84     add(scrollPane, loc);
85     loc = new GridBagConstraints();
86
87     // create results label
88     loc.gridx = 5;
89     loc.gridy = 0;
90     loc.insets.bottom = 20;
91     add(new JLabel("Results"), loc);
92
93     // buttons
94
95     // Create directions label
96     loc.gridx = 10;
97     loc.gridy = 1;
98     loc.insets.right = 20;
```

```
99         add(new JLabel("Directions"), loc);
100
101         // North Button
102         north = new JButton("North");
103         loc.gridx = 10;
104         loc.gridy = 2;
105         loc.gridwidth = 2;
106         loc.insets.top = 10;
107         loc.insets.right = 20;
108         loc.anchor = loc.CENTER;
109         north.setBackground(Color.cyan);
110         add(north, loc);
111
112         // South Button
113         south = new JButton("South");
114         loc.gridx = 10;
115         loc.gridy = 3;
116         loc.gridwidth = 2;
117         loc.insets.top = 10;
118         loc.insets.right = 20;
119         loc.anchor = loc.CENTER;
120         south.setBackground(Color.cyan);
121         add(south, loc);
122
123         // East Button
124         east = new JButton("East");
125         loc.gridx = 10;
126         loc.gridy = 4;
127         loc.gridwidth = 2;
128         loc.insets.top = 10;
129         loc.insets.right = 20;
130         loc.anchor = loc.CENTER;
131         east.setBackground(Color.cyan);
132         add(east, loc);
133
134         // West Button
135         west = new JButton("West");
136         loc.gridx = 10;
137         loc.gridy = 5;
138         loc.gridwidth = 2;
139         loc.insets.top = 10;
140         loc.insets.right = 20;
141         loc.anchor = loc.CENTER;
142         west.setBackground(Color.cyan);
143         add(west, loc);
144
145         // Upstairs button
146         upstairs = new JButton("Upstairs");
147         loc.gridx = 10;
```

```
148         loc.gridy = 6;
149         loc.gridwidth = 2;
150         loc.insets.top = 10;
151         loc.insets.right = 20;
152         loc.anchor = loc.CENTER;
153         upstairs.setBackground(Color.cyan);
154         add(upstairs, loc);
155
156         // Down Stairs Button
157         downstairs = new JButton("Downstairs");
158         loc.gridx = 10;
159         loc.gridy = 7;
160         loc.gridwidth = 2;
161         loc.insets.top = 10;
162         loc.insets.right = 20;
163         loc.anchor = loc.CENTER;
164         downstairs.setBackground(Color.cyan);
165         add(downstairs, loc);
166         loc = new GridBagConstraints();
167
168         // Actions Label
169         loc.gridx = 0;
170         loc.gridy = 13;
171         loc.anchor = loc.WEST;
172         loc.insets.left = 10;
173         add(new JLabel("Actions"), loc);
174         loc = new GridBagConstraints();
175
176         // Help Button
177         help = new JButton("Help");
178         loc.gridx = 1;
179         loc.gridy = 13;
180         loc.gridwidth = 1;
181         loc.insets.right = 10;
182         loc.insets.left = 10;
183         help.setBackground(Color.green);
184         add(help, loc);
185         loc = new GridBagConstraints();
186
187         // Pickup Button
188         pickup = new JButton("Pickup");
189         loc.gridx = 2;
190         loc.gridy = 13;
191         loc.insets.right = 10;
192         loc.gridwidth = 1;
193         pickup.setBackground(Color.green);
194         add(pickup, loc);
195         loc = new GridBagConstraints();
196
```

```
197         // Drop Button
198         drop = new JButton("Drop");
199         loc.gridx = 3;
200         loc.gridy = 13;
201         loc.insets.right = 10;
202         loc.gridwidth = 1;
203         drop.setBackground(Color.green);
204         add(drop, loc);
205         loc = new GridBagConstraints();
206
207         // Eat Button
208         eat = new JButton("Eat");
209         loc.gridx = 4;
210         loc.gridy = 13;
211         loc.insets.right = 10;
212         loc.gridwidth = 1;
213         eat.setBackground(Color.green);
214         add(eat, loc);
215         loc = new GridBagConstraints();
216
217         // look Button
218         look = new JButton("Look");
219         loc.gridx = 5;
220         loc.gridy = 13;
221         loc.insets.right = 10;
222         loc.insets.right = 10;
223         loc.gridwidth = 1;
224         look.setBackground(Color.green);
225         add(look, loc);
226         loc = new GridBagConstraints();
227
228         // List Button
229         list = new JButton("List");
230         loc.gridx = 6;
231         loc.gridy = 13;
232         loc.insets.right = 10;
233         loc.gridwidth = 1;
234         list.setBackground(Color.green);
235         add(list, loc);
236         loc = new GridBagConstraints();
237
238         // Continue Button
239         continueButton = new JButton("Continue");
240         loc.gridx = 7;
241         loc.gridy = 13;
242         loc.gridwidth = 1;
243         loc.insets.right = 10;
244         continueButton.setBackground(Color.red);
245         add(continueButton, loc);
```

```
246         loc = new GridBagConstraints();
247
248         // Fire Button
249         fire = new JButton("Fire");
250         loc.gridx = 8;
251         loc.gridy = 13;
252         loc.gridwidth = 1;
253         loc.insets.right = 10;
254         fire.setBackground(Color.red);
255         add(fire, loc);
256         loc = new GridBagConstraints();
257
258         // add action listeners for direction buttons
259         north.addActionListener(this);
260         south.addActionListener(this);
261         east.addActionListener(this);
262         west.addActionListener(this);
263         downstairs.addActionListener(this);
264         upstairs.addActionListener(this);
265
266         // Add Action Listeners for action buttons
267         help.addActionListener(this);
268         pickup.addActionListener(this);
269         drop.addActionListener(this);
270         eat.addActionListener(this);
271         look.addActionListener(this);
272         list.addActionListener(this);
273         fire.addActionListener(this);
274         continueButton.addActionListener(this);
275
276         // set up File menu
277         fileMenu = new JMenu("File");
278         quitItem = new JMenuItem("Quit");
279         newGame = new JMenuItem("New Game");
280         fileMenu.add(newGame);
281         fileMenu.add(quitItem);
282         menus = new JMenuBar();
283         setJMenuBar(menus);
284         menus.add(fileMenu);
285
286         // Action Listeners for menu items
287         fileMenu.addActionListener(this);
288         quitItem.addActionListener(this);
289         newGame.addActionListener(this);
290
291         // Setup the opening message
292         g.setWelcomeMessage();
293         results.append(g.getMessage());
294         g.look();
```

```
295         results.append(g.getMessage());
296
297         // We don't want continue on yet
298         continueButton.setEnabled(false);
299     }
300
301     /*****
302     * Method to disable buttons if the game is over
303     *****/
304     private void gameOver() {
305         // Checks for if the gameOver() method is true
306         if(g.gameOver() == true) {
307
308             // Disable all buttons
309             north.setEnabled(false);
310             south.setEnabled(false);
311             east.setEnabled(false);
312             west.setEnabled(false);
313             help.setEnabled(false);
314             pickup.setEnabled(false);
315             drop.setEnabled(false);
316             eat.setEnabled(false);
317             look.setEnabled(false);
318             fire.setEnabled(false);
319             list.setEnabled(false);
320             downstairs.setEnabled(false);
321             upstairs.setEnabled(false);
322         }
323     }
324
325     /*****
326     * Method to start a brand new game... Resets
327     * everything in game
328     *****/
329     private void newGame() {
330         // Initialize new game
331         g = new Game();
332
333         // Setup opening message
334         g.setWelcomeMessage();
335         results.setText(g.getMessage());
336         g.look();
337         results.append(g.getMessage());
338
339         // Enable all buttons
340         north.setEnabled(true);
341         south.setEnabled(true);
342         east.setEnabled(true);
343         west.setEnabled(true);
```

```
344         help.setEnabled(true);
345         pickup.setEnabled(true);
346         drop.setEnabled(true);
347         eat.setEnabled(true);
348         look.setEnabled(true);
349         fire.setEnabled(true);
350         list.setEnabled(true);
351         downstairs.setEnabled(true);
352         upstairs.setEnabled(true);
353     }
354
355     /*****
356     * Method to track button presses and actions
357     * Updates the results screen also
358     *****/
359     public void actionPerformed(ActionEvent e) {
360         JComponent buttonPressed = (JComponent) e.getSource();
361
362         // Quit menu item
363         if(buttonPressed == quitItem) {
364             System.exit(1);
365         }
366
367         // New game menu item
368         if(buttonPressed == newGame) {
369             newGame();
370         }
371
372         // Help button
373         if (buttonPressed == help){
374             g.help();
375             results.append(g.getMessage());
376         }
377
378         // Pickup button
379         if(buttonPressed == pickup) {
380             String message = "What do you want to pickup?";
381
382             // Check for what user wants to pickup
383             String pickup = JOptionPane.showInputDialog(null, message);
384             g.pickup(pickup);
385             results.append(g.getMessage());
386         }
387
388         // Drop Button
389         if(buttonPressed == drop) {
390             String message = "What do you want to drop?";
391
392             // Check for what they want to drop
```



```
393         String drop = JOptionPane.showInputDialog(null, message);
394         g.drop(drop);
395         results.append(g.getMessage());
396     }
397
398     // List button
399     if(buttonPressed == list) {
400         g.list();
401         results.append(g.getMessage());
402     }
403
404     // Look button
405     if(buttonPressed == look) {
406         g.look();
407         results.append(g.getMessage());
408     }
409
410     // Fire button
411     if(buttonPressed == fire) {
412         g.fire();
413         results.append(g.getMessage());
414
415         // Check if it is time to search, disables button
416         if(g.isTimeToSearch()) {
417             continueButton.setEnabled(true);
418             fire.setEnabled(false);
419         }
420     }
421
422     // Continue button
423     if(buttonPressed == continueButton) {
424         g.continueGame();
425         results.append(g.getMessage());
426         continueButton.setEnabled(false);
427         fire.setEnabled(true);
428     }
429
430     // Eat Button
431     if(buttonPressed == eat) {
432         String message = "What do you want to eat?";
433
434         // Check for what user wants to eat
435         String eat = JOptionPane.showInputDialog(null, message);
436         g.eat(eat);
437         gameOver();
438         results.append(g.getMessage());
439     }
440
441     // North button
```

```
442         if(buttonPressed == north) {
443             g.move("north");
444             gameOver();
445             results.append(g.getMessage());
446         }
447     }
448
449     // South button
450     if(buttonPressed == south) {
451         g.move("south");
452         gameOver();
453         results.append(g.getMessage());
454     }
455
456     // East button
457     if(buttonPressed == east) {
458         g.move("east");
459         gameOver();
460         results.append(g.getMessage());
461     }
462
463     // West button
464     if(buttonPressed == west) {
465         g.move("west");
466         gameOver();
467         results.append(g.getMessage());
468     }
469
470     // Downstairs button
471     if(buttonPressed == downstairs) {
472         g.move("downstairs");
473         gameOver();
474         results.append(g.getMessage());
475     }
476
477     // Upstairs button
478     if(buttonPressed == upstairs) {
479         g.move("upstairs");
480         gameOver();
481         results.append(g.getMessage());
482     }
483 }
484 }
485 }
```