```
import java.util.*;
import javax.swing.*;
import java.awt.*;
import java.awt.event.*;
         public class PlayGameGui {
              //things for the function of the game private GameMap game; private final int MAX DOORS = 4; private final int MAX THEYS = 10; private final int MAX TIME X = 815; private final int MIN TIME X = 200; private final int MAX TIME Y = 340; private final int MIN TIME Y = 40; private int numOfButts; 7/trace... private Boolean won = false; private String spaces; //trace...
10
11
12
16
17
18
19
              private Key pickedKey;
private Door pickedDoor;
private Room pickedRoom;
20
21
22
23
                //containers
              private JFrame frame;
private JPanel instructs;
25
26
27
28
              //objects for the panel
private JButton startButt;
private JButton isSure;
private JButton notSure;
30
         private JButton[] roomButts;//holds the generated door buttons to enter other
rooms
34
              private JButton[] keyButts;//holds the generated key buttons
35
36
              private JLabel instructionTxt;
private JLabel qameTxt;
private JLabel doorTxt;
private JLabel chancesL;
private JLabel keyTxt;
private JLabel areYouSure;
38
39
40
41
\overline{42}
43
              private ImageIcon imageG;
private JLabel imgLabel;
45
         //frame and panel will be created but mostly just the frame
public PlayGameGui(){
   numOfButts = 0;//keeps track of how many buttons where generated for roomButts
   spaces = "               
anbsp;     "//for aesthic purposes
   game = new GameMap();
   pickedKey = null;
   pickedDoor = null;
   pickedRoom = null;
50
51
52
53
54
55
56
                    instructs = new JPanel();
instructs.setLayout(null);
57
58
59
                    frame = new JFrame();
frame.setDefaultCloseOperation(JFrame.EXIT ON CLOSE);
frame.setPreferredSize(new Dimension(870,700));
60
61
62
63
64
                    frame.setTitle("Adventure Time");
frame.setResizable(false);
frame.pack();
65
66
                    startButt = new JButton("");
67
68
                    /*Objects created in the constructer are place on the panel
* and setVisibility to false unless noted bc it makes it
* easier to remove, setVisible(true) or manipulate variables
* */
69
70
71
72
73
74
75
77
                    isSure = new JButton("YES!");
event5 eventSure = new event5();
isSure.addActionListener(eventSure);
instructs.add(isSure);
isSure.setVisible(false);
isSure.setBounds(250, 640, 50,30 );
79
80
                   notSure = new JButton("No!");
event5 eventNotSure = new event5();
notSure.addActionListener(eventNotSure);
instructs.add(notSure);
notSure.setVisible(false);
notSure.setBounds(310, 640, 50,30 );
81
82
83
84
86
                    roomButts = new JButton[MAX DOORS];
keyButts = new JButton[MAX_KEYS];
88
89
                    instructionTxt = new JLabel("");
gameTxt = new JLabel("");
          /Users/s160540/Desktop/FINALPROJECT_lluo_jaguilar_avalle/FinalProject/PlayGameGui.java 1
```

```
93
94
                         doorTxt = new JLabel("");
                         keyTxt = new JLabel("");
instructs.add(keyTxt);
keyTxt.setVisible(false);
keyTxt.setBounds(190 , 470, 680, 235);
96
97
98
99
                         chancesL = new JLabel("");
instructs.add(chancesL);
chancesL.setVisible(false);
chancesL.setBounds(650,450, 120,50);
100
100
101
102
103
104
105
106
107
                         areYouSure = new JLabel("");
instructs.add(areYouSure);
areYouSure.setVisible(false);
areYouSure.setBounds(190 , 495, 680, 235);
108
109
                         imageG = new ImageIcon("10room.jpg");
\frac{110}{111}
                         imgLabel = new JLabel(imageG);
instructs.add(imgLabel);
Color customColor2 = new Color(0,0,0);
imgLabel.setOpaque(true);
imgLabel.setBackground(customColor2);
imgLabel.setBounds(190 , 30, 680, 370);
112
113
114
115
116
117
118
118
119
120
121
122
123
124
125
                  }
                  /*the first page the user encounters with
 * will begin the entirety of the game by calling
 * action listener also sets up how the static
 * vars are going to look like for the entirety of the game
126
127
128
129
                  public void startPage(){
                         JLabel titleTxt = new JLabel("");
instructs.add(titleTxt);
titleTxt.setOpaque(true);
Color black = new Color(0,0,0);
titleTxt.setBackground(black);
titleTxt.setBounds(190 , 0 , 680, 30);
130
131
132
133
134
135
           //this will set the instructions on the side column and will not be changed
throughout the game
   instructionTxt.setText("<html><font color= 'white'><b>&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;
Instructions</h></hr>
136
          playing game in which, you the user, chooses"+
beloved Don who is kept prisoner by the Dragon. Along your journey through this maze, you will encounter many challenges."+
of which you cannot enter so you will have to find the Keys that open the doors."+
it requires to be opended and the doors already opened will be unlocked. Keep in
mind that all items in the room want"+

"to be used so there."
138
139
141
         mind that all items in the room want"+

"to be used so they may lie and try to trick
you to choose them, however, only one is a key to open the door. "+

"If you happen to choose the wrong key three
times, we're sorry but the dragon will then become aware of your presence"+

and move your Don to a different location
and you will lose the game. Your goal is to get to the final final room"+

"where you will face the final challenge
instructionTxt.setFont(new Font("Source Code Pro Light", Font.BOLD, 10));

//Serif
instructs.add(instructionTxt):
142
144
145
146
                         instructs.add(instructionTxt);
instructs.setForeground(new Color(0xffffdd));
instructionTxt.setOpaque(true);
instructionTxt.setBackground(black);
instructionTxt.setBounds(0 , 0 , 190, 678);
147
147
148
149
150
151
152
          String x = "<html>"+spaces+spaces+ "&nbsp;&nbsp;&nbsp;Welcome to ADVENTURE
TIME</html>";
    gameTxt.setText("<html><font color = 'lime' size = '5'>" + x+"");
    instructs.add(gameTxt);
    gameTxt.setOpaque(true);
    gameTxt.setBackground(black);
    gameTxt.setBounds(190 , 400, 680, 235);
155
156
157
158
159
160
161
162
163
                         startButt = new JButton("Start Game");
event3 eve = new event3();
startButt.addActionListener(eve);//calls to action listener and when pressed
165
            the game begins instructs.add(startButt); startButt.setBounds(480, 640, 100, 30);
167
168
169
```

```
171
172
173
174
175
                   JButton quit = new JButton("Exit");
event2 ev = new event2();
quit.addActionListener(ev);
instructs.add(quit);
quit.setBounds(600, 640, 100, 30);
176
177
178
179
180
                   frame.add(instructs);
frame.setVisible(true);
              }
181
182
183
184
185
                * sets the domino effect for the rest of the game ***/
              public void startGame(){
  instructs.remove(startButt);
  gameTxt.setText("");
  createRoom();
186
187
188
189
              }
190
191
192
193
                 * sets the display for the rest game
194
195
              public void createRoom(){
196
                   if(!game.endOfMap() && game.chancesLeft()){
197
198
199
200
                         String imgNom = game.getCurrentRoom().getRoomName();
ImageIcon n = new ImageIcon(imgNom);
        gameTxt.setText("<html><font color = 'lime' size = '5'>"+spaces + spaces+"
&nbsp;&nbsp;"+
201
                         "Click the room number you wish to enter. </font></html>");
LinkedList<Room> neighbors = game.getConnectingRooms();
Color gray = new Color(211,211,211);
202
203
204
205
206
207
                         //generates rooom buttons to enter other rooms
for (int i = 0; i < neighbors.size(); i++){
   String name = neighbors.get(i).getRoomName();
   roomButts[i] = new JButton(name);
   event e = new event();
   roomButts[i].addActionListener(e);
   roomButts[i].setForeground(new Color(0,128,0));
   roomButts[i].setBackground(gray);
   instructs.add(roomButts[i]);
   roomButts[i].setBounds(300 + (130 * i),410, 75,35);
   numOfButts++;</pre>
208
209
210
211
212
213
214
215
216
217
218
219
                   if(game.endOfMap()){    //hangman
    System.out.println("almost done");
220
221
222
223
224
225
                   }
              }
226
227
       public void createKeys(){
    Key[] currentKeys = game.getCurrentRoom().getRoomKeys();//gets an array of all
the keys in the room
    Random ran = new Random();
228
229
230
231
                         //makes keys show up on screen in random x & y coordinates
for (int i = 0 ; i < MAX KEYS; i++){
   String name = currentKeys[i] getName();
   String bImage = name + ".jpg";
   int x = ran.nextInt((MAX_IMG_X - MIN_IMG_X) + 1) + MIN_IMG_X;
   int y = ran.nextInt((MAX_IMG_Y - MIN_IMG_Y) + 1) + MIN_IMG_Y;</pre>
232
233
234
235
236
237
238
239
240
                             keyButts[i] = new JButton(new ImageIcon(bImage,bImage));
keyButts[i].setOpaque(true);
event4 e4 = new event4();
keyButts[i].addActionListener(e4);
keyButts[i].setVisible(true);
instructs.add(keyButts[i]);
keyButts[i].setBounds(x,y, 30,30);
241
242
243
244
245
246
247
              }
248
249
250
251
252
                * while taversing through a door actions will take place * **/
              public void goThroughDoor(Room rM) {
   Door selectedDoor = game.getDoor(rM);
   pickedDoor = selectedDoor;
253
254
255
256
257
        if(selectedDoor.isLocked()){
    gameTxt.setText("<html><font = 'Source Code Pro Light' color = 'lime' size
= '4'>" +selectedDoor.getLockedMsg()+"</html>");
258
259
         //NEEDS TO MOVED BETTER LEFT FOR LATERS chancesL.setText("<html>You have " + game.getChances()+" chances left to /Users/s160540/Desktop/FINALPROJECT_lluo_jaguilar_avalle/FinalProject/PlayGameGui.java
260
```

```
262
263
264
265
266
267
        } else{
   gameTxt.setText("<html><font = 'Source Code Pro Light' color = 'lime' size
'4'>" +selectedDoor.getUnlockedMsg()+"</html>");
   game.setCurrentRoom(rM);
269
270
271
272
273
274
      public void useKey(Key k){
   keyTxt.setText("<html><font = 'Source Code Pro Light' color = 'lime' size =
'4'>"+ "The key tells you: <br>" + k.getActiveMsg() +"</html>");
   keyTxt.setVisible(true);
275
276
277
      areYouSure.setText("<html><font = 'Source Code Pro Light' color = 'lime' size =
'4'>"+ "Is this the key you choose to use to unlock?</html>");
areYouSure.setVisible(true);
278
279
280
281
282
             isSure.setVisible(true);
283
             notSure.setVisible(true);
284
285
         //action listener for door buttons
public class event implements ActionListener{
   public void actionPerformed(ActionEvent e){
286
287
288
289
290
                 Object o = e.getSource();//gets which button it was from JButton b = null;//crerates a new button
291
292
     for (int i = 0; i < numOfButts; i++){ //sets that all other buttons can't be
pressed once one is pressed
    roomButts[i].setVisible(false);
    roomButts[i].remove(startButt);
    roomButts[i] = null;</pre>
294
295
296
297
298
299
300
                 numOfButts=0; //resets the count
                if(o instanceof JButton){
  b = (JButton)o; //the null button is given the same attributes as source
302
      button
303
304
305
                String bName = b.getText();//it gets the buttons name
306
307
308
309
                bName = bName.substring(0,bName.indexOf("r"));
Room selectedRoom = game.getRoom(bName);
310
311
312
313
                pickedRoom = selectedRoom;
goThroughDoor(pickedRoom);
                createKeys();
314
315
316
317
318
         }
           public class event2 implements ActionListener{
  public void actionPerformed(ActionEvent ev) {
    System.exit(0);
319
320
321
322
323
324
325
326
327
328
329
330
         }
           public class event3 implements ActionListener{
   public void actionPerformed(ActionEvent eve){
      startGame();
               }
331
332
333
334
           public class event4 implements ActionListener{
  public void actionPerformed(ActionEvent even){
      String selectedKeyStr = ((ImageIcon)(((JButton)even.getSource()).getIcon())).
getDescription();
    System.out.println(selectedKeyStr);
336
337
338
                                                                       i++){ //disables all other keys from clicking
                 for (int i = 0; i < MAX KEYS; i
   keyButts[i].setEnabled(false);</pre>
339
340
     Key choosenKey = game.getCurrentRoom().getKey(selectedKeyStr.substring(0,
selectedKeyStr.indexOf(".")));
343
344
345
346
                pickedKey = choosenKey;
                useKey(pickedKey);
      /Users/s160540/Desktop/FINALPROJECT_lluo_jaguilar_avalle/FinalProject/PlayGameGui.java
```

```
}
348
349
350
351
352
353
354
355
356
357
360
361
362
                public class event5 implements ActionListener{
  public void actionPerformed(ActionEvent event) {
    if(event.getSource() == isSure) {
        if(pickedDoor.rightKey(pickedRey)) {
            keyTxt.setText("<html>" + pickedDoor.getUnlockedMsg() + "</html>");
            game.unlockDoor(pickedRoom);
            for(int i =0; i < MAX KEYS; i++) {
                keyButts[i].setVisible(false);
                instructs.remove(keyButts[i]);
        }
}</pre>
                                       } areYouSure.setVisible(false);
                                 createRoom();
}else{
keyTxt.setText("<html>The key says: " + pickedKey.getInactiveMsg()+
363
364
          "</html>");
                                       areYouSure.setText("Oh no! That's not the right key!");
game.wrongAnswer();
366
367
368
370
371
377
377
377
377
377
378
379
381
382
383
                            }else{
  for (int i = 0; i < MAX KEYS; i++){
    keyButts[i].setEnabled(true);
}</pre>
                                 keyTxt.setText("");
areYouSure.setText("");
                            }
isSure.setVisible(false);
//instructs.remove(isSure);
                            notSure.setVisible(false);
//instructs.remove(notSure);
                }
384
385
386
               public static void main(String[] args){
  PlayGameGui n = new PlayGameGui();
  n.startPage();
387
388
389
390 }
```