

[illegible]

```

94 doorTxt = new JLabel("");
95 keyTxt = new JLabel("");
96 instructs.add(keyTxt);
97 keyTxt.setVisible(false);
98 keyTxt.setBounds(190 , 470, 680, 235);
99 
100 chancesL = new JLabel("");
101 instructs.add(chancesL);
102 chancesL.setVisible(false);
103 chancesL.setBounds(650,450, 120,50);
104 
105 areYouSure = new JLabel("");
106 instructs.add(areYouSure);
107 areYouSure.setVisible(false);
108 areYouSure.setBounds(190 , 495, 680, 235);
109 
110 imageG = new ImageIcon("10room.jpg");
111 
112 imgLabel = new JLabel(imageG);
113 instructs.add(imgLabel);
114 Color customColor2 = new Color(0,0,0);
115 imgLabel.setOpaque(true);
116 imgLabel.setBackground(customColor2);
117 imgLabel.setBounds(190 , 30, 680, 370);
118 }
119 
120 /*the first page the user encounters with
121 * will begin the entirety of the game by calling
122 * action listener also sets up how the static
123 * vars are going to look like for the entirety of the game
124 */
125 public void startPage(){
126     JLabel titleTxt = new JLabel("");
127     instructs.add(titleTxt);
128     titleTxt.setOpaque(true);
129     Color black = new Color(0,0,0);
130     titleTxt.setBackground(black);
131     titleTxt.setBounds(190 , 0 , 680, 30);
132 
133 //this will set the instructions on the side column and will not be changed
134 throughout the game
135 instructionTxt.setText("<html><font color= 'white'><b>&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&~\n\nInstructions</b><br>\n\nplaying game in which, you the user, chooses \na path to take in order to save your beloved Don who is kept prisoner by the Dragon. Along your journey through this maze, you will encounter many challenges.\nThere will be doors presented to you some of which you cannot enter so you will have to find the Keys that open the doors.\"+ it requires to be opened and the doors already opened will be unlocked. Keep in 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.\"+ 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\"+ and win or lose you Don (and the game).</font><html>\"));
136 instructionTxt.setFont(new Font(\"Source Code Pro Light\",Font.BOLD, 10));
137 //Serif
138 instructs.add(instructionTxt);
139 instructs.setForeground(new Color(0xffffdd));
140 instructionTxt.setOpaque(true);
141 instructionTxt.setBackground(black);
142 instructionTxt.setBounds(0 , 0 , 190, 678);
143 
144 String x = "<html>" + spaces + spaces + " &nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;& ~\\nWelcome to ADVENTURE TIME</html>";
145 gameTxt.setText("<html><font color = 'lime' size = '5'>" + x + "");
146 instructs.add(gameTxt);
147 gameTxt.setOpaque(true);
148 gameTxt.setBackground(black);
149 gameTxt.setBounds(190 , 400, 680, 235);
150 
151 startButt = new JButton("Start Game");
152 event3 eve = new event3();
153 startButt.addActionListener(eve);//calls to action listener and when pressed the game begins
154 instructs.add(startButt);
155 startButt.setBounds(480 , 640, 100, 30);
156 
157 /Users/sl60540/Desktop/FINALPROJECT lluo jaquilar avalle/FinalProject/PlayGameGui.java

```



```

262 guess wrong or the dragon carries your Don off!</html>");
263     chancesL.setVisible(true);
264     instructs.add(chancesL);
265     chancesL.setBounds(200,450, 120,35);
266 }
267 else{
268     gameTxt.setText("<html><font = 'Source Code Pro Light' color = 'lime' size
= '4'>" + selectedDoor.getUnlockedMsg()+"</html>");
269     game.setCurrentRoom(rM);
270     createRoom();
271 }
272 }
273 }
274 public void useKey(Key k){
275     keyTxt.setText("<html><font = 'Source Code Pro Light' color = 'lime' size =
'4'>" + "The key tells you: <br>" + k.getActiveMsg() + "</html>");
276     keyTxt.setVisible(true);
277 }
278     areYouSure.setText("<html><font = 'Source Code Pro Light' color = 'lime' size =
'4'>" + "Is this the key you choose to use to unlock?</html>");
279     areYouSure.setVisible(true);
280 }
281     isSure.setVisible(true);
282     notSure.setVisible(true);
283 }
284 }
285 //action listener for door buttons
286 public class event implements ActionListener{
287     public void actionPerformed(ActionEvent e){
288         Object o = e.getSource();//gets which button it was from
289         JButton b = null;//creates a new button
290         for (int i = 0; i < numOfButts ; i++){ //sets that all other buttons can't be
291             pressed once one is pressed
292             roomButts[i].setVisible(false);
293             roomButts[i].remove(startButt);
294             roomButts[i] = null;
295         }
296         numOfButts=0; //resets the count
297         if(o instanceof JButton){
298             b = (JButton)o; //the null button is given the same attributes as source
299             button
300         }
301         String bName = b.getText();//it gets the buttons name
302         bName = bName.substring(0,bName.indexOf("r"));
303         Room selectedRoom = game.getRoom(bName);
304         pickedRoom = selectedRoom;
305         goThroughDoor(pickedRoom);
306         createKeys();
307     }
308 }
309 }
310 }
311 }
312 }
313 }
314 }
315 }
316 }
317 }
318 }
319 public class event2 implements ActionListener{
320     public void actionPerformed(ActionEvent ev) {
321         System.exit(0);
322     }
323 }
324 }
325 }
326 public class event3 implements ActionListener{
327     public void actionPerformed(ActionEvent eve){
328         startGame();
329     }
330 }
331 }
332 public class event4 implements ActionListener{
333     public void actionPerformed(ActionEvent even){
334         String selectedKeyStr = ((ImageIcon)((JButton)even.getSource()).getIcon()).
335         getDescription();
336         System.out.println(selectedKeyStr);
337         for (int i = 0; i < MAX KEYS ; i++){ //disables all other keys from clicking
338             keyButts[i].setEnabled(false);
339         }
340         Key choosenKey = game.getCurrentRoom().getKey(selectedKeyStr.substring(0,
341         selectedKeyStr.indexOf(".")));
342         pickedKey = choosenKey;
343         useKey(pickedKey);
344     }
345 }
346 }
347 }

```

```

348     }
349 }
350
351 public class event5 implements ActionListener{
352     public void actionPerformed(ActionEvent event){
353         if(event.getSource() == isSure){
354             if(pickedDoor.rightKey(pickedKey)){
355                 keyTxt.setText("<html>" + pickedDoor.getUnlockedMsg() + "</html>");
356                 game.unlockDoor(pickedRoom);
357                 for(int i=0; i < MAX_KEYS ; i++){
358                     keyButts[i].setVisible(false);
359                     instructs.remove(keyButts[i]);
360                 }
361                 areYouSure.setVisible(false);
362                 createRoom();
363             }else{
364                 keyTxt.setText("<html>The key says: " + pickedKey.getInactiveMsg()+
365 "</html>");
366                 areYouSure.setText("Oh no! That's not the right key!");
367                 game.wrongAnswer();
368             }
369         }else{
370             for (int i = 0; i < MAX_KEYS ; i++){
371                 keyButts[i].setEnabled(true);
372             }
373             keyTxt.setText("");
374             areYouSure.setText("");
375         }
376         isSure.setVisible(false);
377         //instructs.remove(isSure);
378         notSure.setVisible(false);
379         //instructs.remove(notSure);
380     }
381 }
382
383 }
384
385 public static void main(String[] args){
386     PlayGameGui n = new PlayGameGui();
387     n.startPage();
388 }
389
390 }

```