

```

1  /**
2  * PlayGame.java
3  * Written By: Adrianna Valle & Jessenia Aguilar
4  * Written On: Dec 19, 2016
5  *
6  * Driver class. User can use PlayGame class to play the game through
7  * the interactions panel and user input. Instantiates a GameMap object.
8  */
9
10 import java.util.Scanner;
11 public class PlayGame {
12
13     public static void main(String[] args) {
14         Scanner scan = new Scanner(System.in);
15         String resp = "y";
16         Boolean won = false;
17
18         do {
19             //User input to start a new game or start game for first time.
20             System.out.println("Welcome to Aventure Time!!");
21             System.out.println("Do you want to play? Y/N");
22             resp = scan.nextLine().toLowerCase(); //nextLine to account for blank entries
23             while(!resp.equals("y") && !resp.equals("n")){
24                 System.out.println("Please enter a valid response to proceed.");
25                 System.out.println("Do you want to play? Y/N");
26                 resp = scan.nextLine();
27             }
28             if(resp.equals("n")) //User doesn't want to play anymore
29                 break;
30
31             //Start game properties
32             GameMap game = new GameMap();
33
34             while(!game.endOfMap() && game.chancesLeft()){
35
36                 //Showing rooms.
37                 System.out.println("\nEnter the room number you wish to enter. Enter Q to
38 quit. (-)locked & (+)unlocked");
39                 System.out.println(game.printRooms());
40
41                 //Gets the response for the room number.
42                 String choosenRoom = "";
43                 while(game.getRoom(choosenRoom)==null&& !choosenRoom.equals("q")){
44                     try{
45                         System.out.println("Please select a valid door number.");
46                         choosenRoom = scan.nextLine().toLowerCase();
47                     }
48                     catch(IllegalArgumentException ex){
49                         System.out.println("Input is not a valid number entry.");
50                     }
51                     catch(NullPointerException ex){
52                         System.out.println("Input is not a valid number entry.");
53                     }
54                 }
55                 if(choosenRoom.equals("q")){ //User doesn't want to play anymore
56                     resp = "n";
57                     break;
58                 }
59
60                 Room selectedRoom = game.getRoom(choosenRoom);
61                 //Check if door is locked or not
62                 Door selectedDoor = game.getDoor(selectedRoom);
63                 if(selectedDoor.isLocked()){
64                     System.out.println("\n"+selectedDoor.getLockedMsg());
65                     String choice = "n";
66                     //key loop: until door is resolved or you have no more chances
67                     while(!choice.equals("q") && !choice.equals("y") && game.chancesLeft()){
68
69                         //Deals with user interaction and correct response
70                         System.out.println("You have " + game.getChances()+" chances left to
71 guess wrong or the dragon carries your Don off!\n");
72                         System.out.println("Type the name of the key you want to view. Type Q
73 to quit the game");
74                         System.out.println(game.printKeys());
75
76                         String selectedKeyStr = scan.nextLine().toLowerCase();
77                         //Checks if key is in the list
78                         while(!game.getCurrentRoom().validKey(selectedKeyStr) && !
79 selectedKeyStr.equals("q")){
80                             System.out.println("Response is invalid. Type in a vaild key or enter
81 Q to quit.");
82                             selectedKeyStr = scan.nextLine().toLowerCase();
83                         }
84                         if(selectedKeyStr.equals("q")){ //User doesn't want to play anymore
85                             resp = "n";
86                             break;
87                         }
88
89                         Key choosenKey = game.getCurrentRoom().getKey(selectedKeyStr);
90                         System.out.println("\nThe key tells you: " +choosenKey.getActiveMsg());
91
92                         //User interaction to get it to get a vaild response
93                         System.out.println("Is this the key you choose to use to unlock?(Y/N).
94 /Users/s160540/Desktop/FINALPROJECT_lluo_jaguilar_avalle/FinalProject/PlayGame.java 1

```

```

90 Enter Q to Quit.");
91     choice = scan.nextLine().toLowerCase();
92     while(!choice.equals("y") && !choice.equals("n") && !choice.equals
93 ("q")){
94         System.out.println("Response is invalid. Enter y/n or Q to quit.");
95         System.out.println("Is this the key you choose to use to unlock the
96 door?(Y/N). Enter Q to Quit.");
97         choice = scan.nextLine().toLowerCase();
98         if(choice.equals("q")){
99             resp = "n";
100             break;
101         }
102         if(choice.equals("y")){
103             if(selectedDoor.rightKey(choosenKey)){
104                 System.out.println("\n"+selectedDoor.getUnlockedMsg()+"\n");
105                 game.unlockDoor(selectedRoom);
106             }
107             else{
108                 System.out.println("The key says: " + choosenKey.getInactiveMsg());
109                 System.out.println("Oh no! That's not the right key!");
110                 game.wrongAnswer();
111                 choice = "n";
112             }
113         }
114     }
115     else{ //Door is not locked
116         System.out.println("\n" + selectedDoor.getUnlockedMsg());
117         //Ensures correct user response
118         System.out.println("Do you wish to enter room " + selectedRoom + "? (Y/N)
119 Enter Q to quit.");
120         String enteringDoorResp = scan.nextLine().toLowerCase();
121         while(!enteringDoorResp.equals("y") && !enteringDoorResp.equals("n") && !
122 enteringDoorResp.equals("q")){
123             System.out.println("Response is invalid. Enter a valid response");
124             System.out.println("Is this the key you choose to use to unlock the
125 door?(Y/N). Enter Q to Quit.");
126             enteringDoorResp = scan.nextLine().toLowerCase();
127         }
128         if(enteringDoorResp.equals("q")){
129             resp = "n";
130             break;
131         }
132         if(enteringDoorResp.equals("y")){
133             game.setCurrentRoom(selectedRoom);
134         }
135     }
136     if(resp.equals("n"))
137         break;
138 }
139 if(game.endOfMap()){
140     Hangman minigame = new Hangman("hangmanText.txt");
141     won = minigame.playHangman();
142 }
143 if(won){ //winning message
144     System.out.println("You entered the room and managed to save your Don right
145 before the Dragon took him away! He tells you, 'Aahh my hero' right before he takes
146 you in for a kiss");
147     System.out.println("Congratulation");
148     System.out.println("Congratulation");
149     System.out.println("Congratulation");
150     System.out.println("Congratulation");
151     System.out.println("Congratulation");
152     System.out.println("Congratulation");
153     System.out.println("Congratulation");
154     System.out.println("Congratulation");
155     System.out.println("Would you like to play again?(Y/N)");
156     resp = scan.nextLine().toLowerCase();
157     while(!resp.equals("y") && !resp.equals("n")){
158         System.out.println("Response is invalid.");
159         System.out.println("Would you like to play again?(Y/N)");
160         resp = scan.next();
161     }
162 }
163 else{ //game was lost
164     System.out.println("Oh no! One of the dragon's henchmans approaches you and
165 tells you the dragon knew you were coming and hid him again!");
166     System.out.println("You have to go save him! Would you like to play again?
167 (Y/N)");
168     resp = scan.nextLine().toLowerCase();
169     while(!resp.equals("y") && !resp.equals("n")){
170         System.out.println("Response is invalid.");
171         System.out.println("Would you like to play again?(Y/N)");
172         resp = scan.next();
173     }
174 }

```

```
174     while(resp.equals("y"));
175     System.out.println("Sorry to see you go! Come again!");
176     scan.close();
177 }
178 }
```