

```

1  /* CS248 Assignment 4: Hunt the Wumpus
2     Author: Casey Sobechs
3     This program is a text based game that will have the user move through room
   looking for a wumpus to kill.
4     It will implement rooms as objects and fill them with obstacles accordingly.
5  */
6
7  import java.io.*;
8  import java.util.*;
9
10 class HuntTheWumpus
11 {
12     static void Delay (int a)
13     {
14         try{Thread.sleep(a*1000);} //stop program for "a" seconds
15         catch(InterruptedException ie){Thread.currentThread().interrupt();} //return
   from sleep
16     }
17
18     public static void main(String [] args) throws IOException
19     {
20         //variables
21         int moveshoot=0;
22         int arrows=3;
23         int shotroom=0;
24         int done=0;
25         int spiderroom[]= new int [2];
26         int pitroom[]= new int[2];
27         int wumpusroom=0;
28         int arrowroom=0;
29         int prevlocal=0;
30         int game=1;
31         int location=0;
32
33         //start of program
34         Scanner input=new Scanner(System.in);
35         Scanner roomread= new Scanner(new FileReader("rooms.txt")); //read the the
   text file with room information
36         int roomtot=roomread.nextInt(); //get total room count
37
38         Room [] cave; //initialize the cave full of rooms
39         cave=new Room[roomtot];
40
41         for(int i=0; i<cave.length; i++) //fill the rooms with their respective
   values
42         {
43             cave[i]=new Room(roomread);
44         }
45
46         while(done==0) //obstacle/enemy placement
47         {
48             wumpusroom=(int)(roomtot*Math.random()); //generate random location for
   wumpus, spiders, and bottomless pit
49             arrowroom=1; //(int)(roomtot*Math.random());
50             for(int i=0; i<2; i++)
51             {
52                 spiderroom[i]=(int)(roomtot*Math.random());
53                 pitroom[i]=(int)(roomtot*Math.random());
54             }
55         }

```

```

56         //check if any of the locations are forbidden i.e. overlap or starting
room
57
58         if(wumpusroom==0||spiderroom[0]==0||spiderroom[1]==0||pitroom[0]==0||pitroom[1]==0|
|wumpusroom==spiderroom[0]||wumpusroom==spiderroom[1]
59
60         ||wumpusroom==pitroom[0]||wumpusroom==pitroom[1]||spiderroom[0]==pitroom[0]||spider
room[0]==pitroom[1]||spiderroom[0]==spiderroom[1]
61
62         ||spiderroom[1]==pitroom[0]||spiderroom[1]==pitroom[1]||pitroom[0]==pitroom[1]||arr
owroom==wumpusroom||arrowroom==spiderroom[0]
63
64         ||arrowroom==spiderroom[1]||arrowroom==pitroom[0]||arrowroom==pitroom[1])
        {}
65         else//all locations are different and not first room
        {done=1;}
66     }
67     //set room locations with proper status message
68     cave[spiderroom[0]].status='s'; //s for spider
69     cave[spiderroom[1]].status='s';
70     cave[wumpusroom].status='w'; //w for wumps
71     cave[pitroom[0]].status='p'; //p for pit
72     cave[pitroom[1]].status='p';
73     cave[arrowroom].status='a'; //a for arrows
74
75     //Startup screen
76     System.out.println("\n\nWelcome to Hunt the Wumpus: Sonic Edition\n\n");
77     Delay(1);
78     System.out.println("You are Sonic the Hedgehod (TM) and you need to fight
the evil Eggman." +
79
80     "You have three dash attacks that you can use on him,
but he is protected by his robot minions and spike traps."
81
82     +"Will you be able to defeat him and save your animal
friends????\n\n");
83     Delay(3);
84     System.out.println("Level Start\n");
85     Delay(1);
86     //print current status
87     cave[location].print();
88     System.out.println("You have "+ arrows+ " dash attacks.");
89
90     //check if any adjacent rooms have a status
91     if(cave[cave[prevlocal].adjac1-1].status=='s' || cave[cave[prevlocal].adjac2-
1].status=='s' || cave[cave[prevlocal].adjac3-1].status=='s')
92     {
93         System.out.println("you hear some cranky robot noises"); //spiders
nearby
94     }
95
96     if(cave[cave[prevlocal].adjac1-1].status=='a' || cave[cave[prevlocal].adjac2-
1].status=='a' || cave[cave[prevlocal].adjac3-1].status=='a')
97     {
98         System.out.println("There are some sparkling noises nearby"); //arrows
nearby
99     }
100
101     if (cave[cave[prevlocal].adjac1-1].status=='p' || cave[cave[prevlocal].adjac2-
1].status=='p' || cave[cave[prevlocal].adjac3-1].status=='p')
102     {
103         System.out.println("Some stabby noises are coming from nearby"); //pit
nearby

```

```

100     }
101
102     if(cave[cave[prevlocal].adjac1-1].status=='w' || cave[cave[prevlocal].adjac2-
103 1].status=='w' || cave[cave[prevlocal].adjac3-1].status=='w')
104     {
105         System.out.println("There is a sound of an egg...could it be him?");
106         //wumpus nearby
107     }
108     else{} //nothing nearby
109
110     System.out.println(arrowroom);
111
112     //begin game
113     while(game==1)
114     {
115         Delay(1);
116         //check for movement or shooting
117         System.out.println("\nDo you want to move or shoot??\n1 for move 0 for
118 shoot");
119         moveshoot=input.nextInt();
120
121         if(arrows==0) //out of arrows
122         {
123             if(cave[arrowroom].status!='a') //there are no more arrows left in
124 the game
125             {
126                 System.out.println("You are out of dash attacks and have
127 failed.");
128                 Delay(1);
129                 game=0;
130                 break;
131             }
132             else //you can still get some arrows from the arrow room
133             {
134                 System.out.println("You're out of dash attacks, but maybe
135 there's something that can reinvigorate you");
136                 Delay(1);
137             }
138         }
139
140         if(moveshoot==0) //shoot the arrow
141         {
142             System.out.println("Okay!\nIt's go time!");
143             Delay(1);
144             System.out.println("which room do you want to shoot into?? Choose
145 wisely...");
146
147             shotroom=input.nextInt(); //pick shooting room
148
149             if(shotroom!=cave[location].adjac1&&shotroom!=cave[location].adjac2&&shotroom!=cave
150 [location].adjac3) //shotroom not adjacent to current room
151             {
152                 System.out.println("That doesn't work. \n"); //insult
153                 Delay(1);
154                 arrows--; //subtract arrows
155
156                 cave[location].print();
157                 System.out.println("You have "+ arrows+ " dash attacks.");
158             }
159         }
160     }
161 }

```

```

152     }
153     else if(cave[shotroom-1].status=='w') //you shot the wumpus
154     {
155         Delay(1);
156         System.out.println("It's eggman! ");
157         Delay(1);
158         System.out.println("You strike him and end his rein of
terror!");
159         Delay(1);
160         System.out.println("\nCongratulations!");
161         game=0;
162     }
163     else //you shoot in a room not with wumpus
164     {
165         Delay(1);
166         System.out.println("\nyou miss your attack\n");
167         arrows--;
168         cave[location].print();
169         System.out.println("You have "+ arrows+ " dash attacks.");
170
171     }
172
173 }
174 else if(moveshoot==1) //move
175 {
176     System.out.println("\nwhich room do you want to move to??");
177     location=input.nextInt()-1;//receive new room location
178
179     if(location!=cave[prevlocal].adjac1-
1&&location!=cave[prevlocal].adjac2-1&&location!=cave[prevlocal].adjac3-1) //the
number chosen isnt an adjacent room
180     {
181         //insult
182         System.out.println("doesn't work, try again");
183     }
184     else //move to adjacent room
185     {
186         if(cave[location].status=='w') //walked into wumpus room
187         {
188             System.out.println("\neggman throws you off the stage");
189             game=0;
190         }
191         else if(cave[location].status=='s') //walked into spider room
192         {
193             System.out.println("\neggman's minions throw you off the
stage.");
194             game=0;
195         }
196         else if(cave[location].status=='p') //walk into pit room
197         {
198             System.out.println("\neggman's spikes stab you off the
stage.");
199             game=0;
200         }
201         else //move into a safe room
202         {
203             if(cave[location].status=='a') //enter an arrow room
204             {
205                 System.out.println("You found some rings! oh boy you
feel like a whole new hedgehog.");
206                 arrows=arrows+3; //increase arrow supply

```

```

207         cave[location].status='0'; //set status of room to empty
because you picked up the arrows
208         System.out.println("You now have "+ arrows+ " dash
attacks.\n");
209         Delay(2);
210     }
211
212
213     prevlocal=location;
214     cave[location].print();
215     System.out.println("You have "+ arrows+ " dash attacks.");
216
217     //spider room near by
218     if(cave[cave[prevlocal].adjac1-
1].status=='s' || cave[cave[prevlocal].adjac2-
1].status=='s' || cave[cave[prevlocal].adjac3-1].status=='s')
219     {
220         System.out.println("you hear some cranky robot noises");
221     }
222     //pit room nearby
223     if (cave[cave[prevlocal].adjac1-
1].status=='p' || cave[cave[prevlocal].adjac2-
1].status=='p' || cave[cave[prevlocal].adjac3-1].status=='p')
224     {
225         System.out.println("Some stabby noises are coming from
nearby");
226     }
227     //wumpus room nearby
228     if(cave[cave[prevlocal].adjac1-
1].status=='w' || cave[cave[prevlocal].adjac2-
1].status=='w' || cave[cave[prevlocal].adjac3-1].status=='w')
229     {
230         System.out.println("There is a sound of an egg...could
it be him?");
231     }
232     //arrow room nearby
233     if(cave[cave[prevlocal].adjac1-
1].status=='a' || cave[cave[prevlocal].adjac2-
1].status=='a' || cave[cave[prevlocal].adjac3-1].status=='a')
234     {
235         System.out.println("There are some sparkling noises
nearby"); //arrows nearby
236     }
237     else //nothing nearby
238     {}
239     }
240 }
241
242 }
243 else //you enter a number that's not 0 or 1 to move or shoot
244 {
245     System.out.println("that's not an option, stick to the rules");
246 }
247 }
248
249 //endgame sequence
250 input.close(); //close input so java stops yelling at me
251 Delay(3);
252 System.out.println("\nYou have saved your animal friends! Congrats!");
253 Delay(2);

```

```
254         System.out.println("\ncredits:\n1.me, I made the game\n2.you, player of the  
game\n");  
255         Delay(2);  
256         System.out.println("\nthanks for playing!\n");  
257         Delay(1);  
258     }  
259 }
```