Casey Sobecks
Hunt The Wumpus Code

```
PS C:\Users\csobe\OneDrive\Documents\Butler\(13) 2020 Spring\CS248\Assignment
4> java HuntTheWumpus


Welcome to Hunt the Wumpus: Sonic Edition


You are Sonic the Hedgehod (TM) and you need to fight the evil Eggman.You
have three dash attacks that you can use on him, but he is protected by his
robot minions and spike traps.Will you be able to defeat him and save your
animal friends?????


Level Start

Your current room:1 The Adjacent rooms:2 6 10
 sign says EGG
You have 3 dash attacks.
There are some sparkling noises nearby
1

Do you want to move or shoot??
1 for move 0 for shoot
1

which room do you want to move to??
2
You found some rings! oh boy you feel like a whole new hedgehog.
You now have 6 dash attacks.

Your current room:2 The Adjacent rooms:1 3 7
 level is dark
You have 6 dash attacks.
Some stabby noises are coming from nearby
There is a sound of an egg...could it be him?

Do you want to move or shoot??
1 for move 0 for shoot
0
Okay!
It's go time!
which room do you want to shoot into?? Choose wisely...
3

you miss your attack

Your current room:2 The Adjacent rooms:1 3 7
 level is dark
You have 5 dash attacks.

Do you want to move or shoot??
1 for move 0 for shoot
0
```

Casey Sobecks
Hunt The Wumpus Code

```
Okay!
It's go time!
which room do you want to shoot into?? Choose wisely...
7
It's eggman!
You strike him and end his rein of terror!

Congratulations!

You have saved your animal friends! Congrats!

credits:
1.me, I made the game
2.you, player of the game


thanks for playing!

PS C:\Users\csobe\OneDrive\Documents\Butler\(13) 2020 Spring\CS248\Assignment
4>
```

Casey Sobecks
Hunt The Wumpus Code

```
Room.txt
10
1 2 6 10
A sign says EGG
2 1 3 7
This level is dark
3 2 4 8
The level is stinky
4 3 5 9
The level is round
5 4 6 10
The level has a little man in the corner
6 1 5 7
Yikes! you stepped in something
7 2 6 8
The eggman is not in this room
8 3 7 9
OOOOOO spooky ghost egg
9 4 8 10
There's an empty ring case here
10 1 5 9
ReeEEEEEEEEE says the robot in the corner
```

```java
import java.util.Scanner;

public class Room
{
    //data members
    int current;
    int adjac1;
    int adjac2;
    int adjac3;
    String message;
    char status;

    //constructor

    /** @param current current room
     * @param adjac1 first adjacent room
     * @param adjac2 second adjacent room
     * @param adjac3 third adjacent room
     * @param message the message of the room
     * @param status the status of the room if it has anything in it
    */

    public Room(Scanner roomread) //read in from a file and fill in the appropriate data
    {
        current=roomread.nextInt();
        adjac1=roomread.nextInt();
        adjac2=roomread.nextInt();
        adjac3=roomread.nextInt();
        roomread.next();
        message=roomread.nextLine();
        status='0';
    }

    //accessor
    public void print() //display current room info
    {
        System.out.println("Your current room:"+current+" The Adjacent rooms:"
        +adjac1+" "+adjac2+" "+adjac3+"\n"+message);
    }
}
```

```java
/* CS248 Assignment 4: Hunt the Wumpus
    Author: Casey Sobecks
    This program is a text based game that will have the user move through room looking for a wumpus to kill.
    It will implement rooms as objects and fill them with obstacles accordingly.
*/

import java.io.*;
import java.util.*;

class HuntTheWumpus
{
    static void Delay (int a)
    {
        try{Thread.sleep(a*1000);} //stop program for "a" seconds
        catch(InterruptedException ie){Thread.currentThread().interrupt();} //return from sleep

    }

    public static void main(String [] args) throws IOException
    {
        //variables
        int moveshoot=0;
        int arrows=3;
        int shotroom=0;
        int done=0;
        int spiderroom[]= new int [2];
        int pitroom[]= new int[2];
        int wumpusroom=0;
        int arrowroom=0;
        int prevlocal=0;
        int game=1;
        int location=0;

        //start of program
        Scanner input=new Scanner(System.in);
        Scanner roomread= new Scanner(new FileReader("rooms.txt"));//read the the text file with room information
        int roomtot=roomread.nextInt(); //get total room count

        Room [] cave; //initialize the cave full of rooms
        cave=new Room[roomtot];

        for(int i=0; i<cave.length; i++) //fill the rooms with their respective values
        {
            cave[i]=new Room(roomread);
        }

        while(done==0) //obstacle/enemy placement
        {
            wumpusroom=(int)(roomtot*Math.random()); //generate random location for wumpus, spiders, and bottomless pit
            arrowroom=(int)(roomtot*Math.random());
            for(int i=0; i<2; i++)
            {
                spiderroom[i]=(int)(roomtot*Math.random());
                pitroom[i]=(int)(roomtot*Math.random());
            }
```

```
56              //check if any of the locations are forbidden i.e. overlap or starting
    room
57
     if(wumpusroom==0||spiderroom[0]==0||spiderroom[1]==0||pitroom[0]==0||pitroom[1]==0|
    |wumpusroom==spiderroom[0]||wumpusroom==spiderroom[1]
58
     ||wumpusroom==pitroom[0]||wumpusroom==pitroom[1]||spiderroom[0]==pitroom[0]||spider
    room[0]==pitroom[1]||spiderroom[0]==spiderroom[1]
59
     ||spiderroom[1]==pitroom[0]||spiderroom[1]==pitroom[1]||pitroom[0]==pitroom[1]||arr
    owroom==wumpusroom||arrowroom==spiderroom[0]
60
     ||arrowroom==spiderroom[1]||arrowroom==pitroom[0]||arrowroom==pitroom[1])
61              {}
62              else//all locations are different and not first room
63              {done=1;}
64          }
65          //set room locations with proper status message
66          cave[spiderroom[0]].status='s'; //s for spider
67          cave[spiderroom[1]].status='s';
68          cave[wumpusroom].status='w'; //w for wumps
69          cave[pitroom[0]].status='p'; //p for pit
70          cave[pitroom[1]].status='p';
71          cave[arrowroom].status='a'; //a for arrows
72
73          //Startup screen
74          System.out.println("\n\nWelcome to Hunt the Wumpus: Sonic Edition\n\n");
75          Delay(1);
76          System.out.println("You are Sonic the Hedgehod (TM) and you need to fight
    the evil Eggman." +
77                             "You have three dash attacks that you can use on him,
    but he is protected by his robot minions and spike traps."
78                             +"Will you be able to defeat him and save your animal
    friends?????\n\n");
79          Delay(3);
80          System.out.println("Level Start\n");
81          Delay(1);
82          //print current status
83          cave[location].print();
84          System.out.println("You have "+ arrows+ " dash attacks.");
85
86          //check if any adjacent rooms have a status
87          if(cave[cave[prevlocal].adjac1-1].status=='s'||cave[cave[prevlocal].adjac2-
    1].status=='s'||cave[cave[prevlocal].adjac3-1].status=='s')
88          {
89              System.out.println("you hear some cranky robot noises"); //spiders
    nearby
90          }
91
92          if(cave[cave[prevlocal].adjac1-1].status=='a'||cave[cave[prevlocal].adjac2-
    1].status=='a'||cave[cave[prevlocal].adjac3-1].status=='a')
93          {
94              System.out.println("There are some sparkling noises nearby"); //arrows
    nearby
95          }
96
97          if (cave[cave[prevlocal].adjac1-1].status=='p'||cave[cave[prevlocal].adjac2-
    1].status=='p'||cave[cave[prevlocal].adjac3-1].status=='p')
98          {
99              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-
    1].status=='w'||cave[cave[prevlocal].adjac3-1].status=='w')
103         {
104             System.out.println("There is a sound of an egg...could it be him?");
    //wumpus nearby
105         }
106         else{} //nothing nearby
107
108         System.out.println(arrowroom);
109
110
111         //begin game
112         while(game==1)
113         {
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
    shoot");
118             moveshoot=input.nextInt();
119
120             if(arrows==0) //out of arrows
121             {
122                 if(cave[arrowroom].status!='a') //there are no more arrows left in
    the game
123                 {
124                     System.out.println("You are out of dash attacks and have
    failed.");
125                     Delay(1);
126                     game=0;
127                     break;
128                 }
129                 else //you can still get some arrows from the arrow room
130                 {
131                     System.out.println("You're out of dash attacks, but maybe
    there's something that can reinvigorate you");
132                     Delay(1);
133                 }
134             }
135
136             if(moveshoot==0) //shoot the arrow
137             {
138                 System.out.println("Okay!\nIt's go time!");
139                 Delay(1);
140                 System.out.println("which room do you want to shoot into?? Choose
    wisely...");
141
142                 shotroom=input.nextInt(); //pick shooting room
143
 if(shotroom!=cave[location].adjac1&&shotroom!=cave[location].adjac2&&shotroom!=cave
    [location].adjac3) //shotroom not adjacent to current room
144                 {
145                     System.out.println("That doesn't work. \n"); //insult
146                     Delay(1);
147                     arrows--; //subtract arrows
148
149                     cave[location].print();
150                     System.out.println("You have "+ arrows+ " dash attacks.");
151
```

```
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 }
```