Abdelghafour Abdou

Abderafii Abdou

Project #2: Wumpus Game

AL AKHAWAYN UNIVERSITY in Ifrane

School of Science and Engineering

CSC 4301 01: Intro. to Artificial Intelligence

DR. Tajjeeddine Rachidi

June 29, 2022

In this project, we tried to create a simple Wumpus game where the player can ask different questions to the agent in order to get some information about the state of the game, the possible moves of the player, the location of pits, the possibility to shoot the Wumpus, if there is any breeze in a room, which rooms are safe to go to ….

Key Predicates, The meaning of variables:

Room:

```
15  %lets assume that we have a 8*8 matrix as a board for our game.
16  %So the player must be in coordinates of rooms bigger or equal to 1 and smaller or equal to 8.
17  room(X,Y):-
18      X @>= 1,
19      X @=< 8,
20      Y @>= 1,
21      Y @=< 8.
```

This is a small function to make sure that if the user enters an invalid room coordinate, the code will print back false. It also gives the user and idea about the size of the matrix or the board.

Breezes:

```
22
23  %Lets create breeze so we can know the location of pits in our game.
24  breeze(room(X,Y)):-
25  NewX1 is X - 1,
26  pit(room(NewX1,Y)).
27
28  breeze(room(X,Y)):-
29  NewX2 is X + 1,
30  pit(room(NewX2,Y)).
31
32  breeze(room(X,Y)):-
33  NewY1 is Y - 1,
34  pit(room(X,NewY1)).
35
36  breeze(room(X,Y)):-
37  NewY2 is Y + 1,
38  pit(room(X,NewY2)).
```

In this part, we tried to create the breeze indicator in each room adjacent to a pit by calculating the coordinate of the adjacent rooms to the pit, all the 4 possibilities of the room's coordinates are included.

Pits:

```
3  %The location of the pit.
4  pit(room(4,2)).
```

We created a pit with already known coordinates in order to simplify the testing.

Wumpus:

```
1  %The location of the wumpus.
2  wumpus(room(3,2)).
```

We created a Wumpus with already known coordinates for the sake of making the testing easier.

Stench:

```
39
40  %Lets create an indicator for the wumpus using the Stench.
41  stench(room(X, Y)):-
42  NewX1 is X - 1,
43  wumpus(room(NewX1,Y)).
44
45  stench(room(X, Y)):-
46  NewX2 is X + 1,
47  wumpus(room(NewX2,Y)).
48
49  stench(room(X, Y)):-
50  NewY1 is Y - 1,
51  wumpus(room(X,NewY1)).
52
53  stench(room(X, Y)):-
54  NewY2 is Y + 1,
55  wumpus(room(X,NewY2)).
```

We created a stench coming from the Wumpus in the nearby cells by calculating the coordinate of the adjacent rooms to the Wumpus, all the 4 possibilities of the room's coordinates are included.

Adjacent To:

```
5   %The location of the room to be checked for adjacents.
6   room((1,1)).
```

```
56
57  %Lets see which rooms are Adjacent to the player.
58  adjacentto(room(X,Y)):-
59  NewX1 is X - 1,
60  room((NewX1,Y)).
61
62  adjacentto(room(X,Y)):-
63  NewX2 is X + 1,
64  room((NewX2,Y)).
65
66  adjacentto(room(X,Y)):-
67  NewY1 is Y - 1,
68  room((X,NewY1)).
69
70  adjacentto(room(X,Y)):-
71  NewY2 is Y + 1,
72  room((X,NewY2)).
73
```

We created a function to check if a room given by the user is adjacent to a predeclared room, in this case the predeclared room is (1,1).

Gold:

```
11  %The location of the gold in the map.
12  gold(room(5,7)).
```

We created gold in a certain room manually with already known coordinates in order to simplify the testing.

Safe:

```
74
75    %Lets check if the room is safe.
76    checkifsafe(room(X,Y)):-
77    room(X,Y),
78    not(pit(room(X,Y))), not(wumpus(room(X,Y))).
79
80    checkifsafe(room(X,Y)):-
81    room(X,Y),
82    not(pit(room(X,Y))), (wumpusisdead(0)).
83
84    %Here we check if the adjacent rooms are safe or not.
85    checkroomontheleft():-
86    startingpoint((X,Y)),
87    NewX is X-1,
88    checkifsafe(room(NewX,Y)).
89
90    checkroomontheright():-
91    startingpoint((X,Y)),
92    NewX is X+1,
93    checkifsafe(room(NewX,Y)).
94
95    checkroomonsouth():-
96    startingpoint((X,Y)),
97    NewY is Y-1,
98    checkifsafe(room(X,NewY)).
99
100   checkroomonnorth():-
101   startingpoint((X,Y)),
102   NewY is Y+1,
103   checkifsafe(room(X,NewY)).
104
```

```
9     %The starting point of the player.
10    startingpoint((3,2)).
```

In this function, we check if a room is safe, the coordinates of the room are given by the user and they are used to check if the room is safe, by checking if there is no pit in the room and if there is no Wumpus in the room or if its dead. Moreover, we can check if a room to the right, left, north or south is safe by simply giving a starting point which is already declared to make testing easier, and using the checkifsafe() function again on the visited room.

Grab gold:

```
103
104   %We grab the gold if we are in the same room as the gold.
105   grabgold(room(X1,Y1)):-
106   gold(room(X,Y)),
107   X1 == X, Y1 == Y, write("Success").
108
```

In this function, the user can pick up the gold if he is in the same room as the gold and that is achieved through a comparison between X1 and Y1 which are the coordinates of the player and X and Y which are the coordinates of the gold.

Shoot Wumpus:

```
111    %We shoot the wumpus if its in one of the adjacent rooms to the player.
112    shootwumpus(room(X1,Y1)):-
113    wumpus(room(X,Y)),
114    stench(room(X1, Y1)),
115    wumpusisdead(0).
116
117
```

The player shoots the Wumpus if he is in an adjacent room to the Wumpus, we can make sure that this information is obtained throughout the use of the function stench(room(X1,Y1)) which returns true if the room is adjacent to a Wumpus, otherwise if returns false , if the returned value is true then we change the Wampus status to dead which is implemented by wumpusisdead(0).

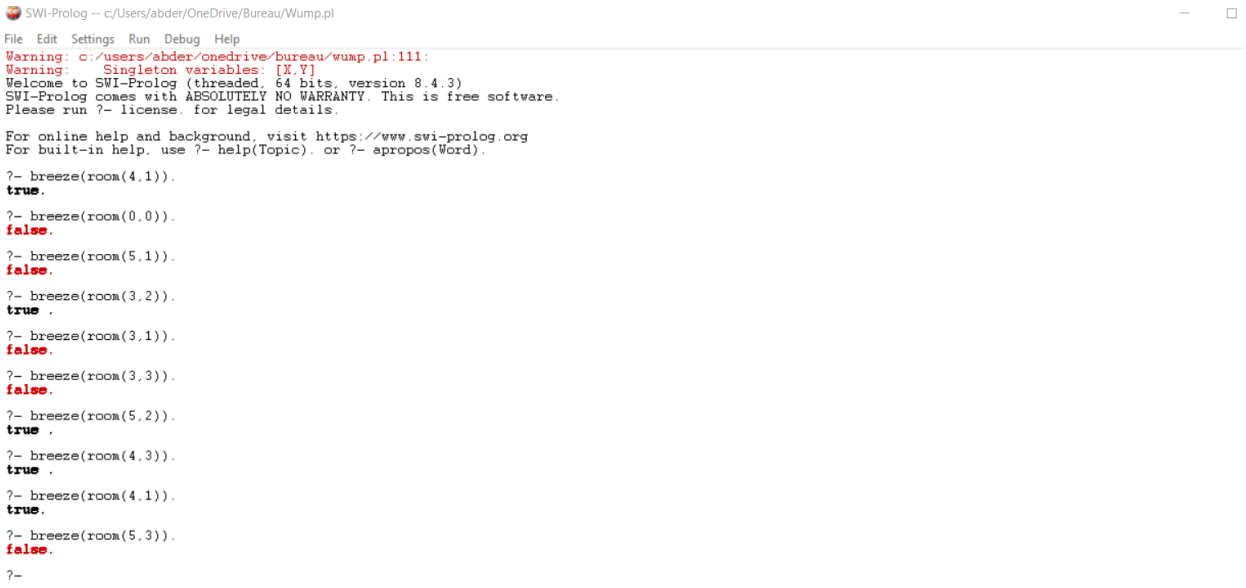
Experiments:

Experiment 1:

Testing on the room function:

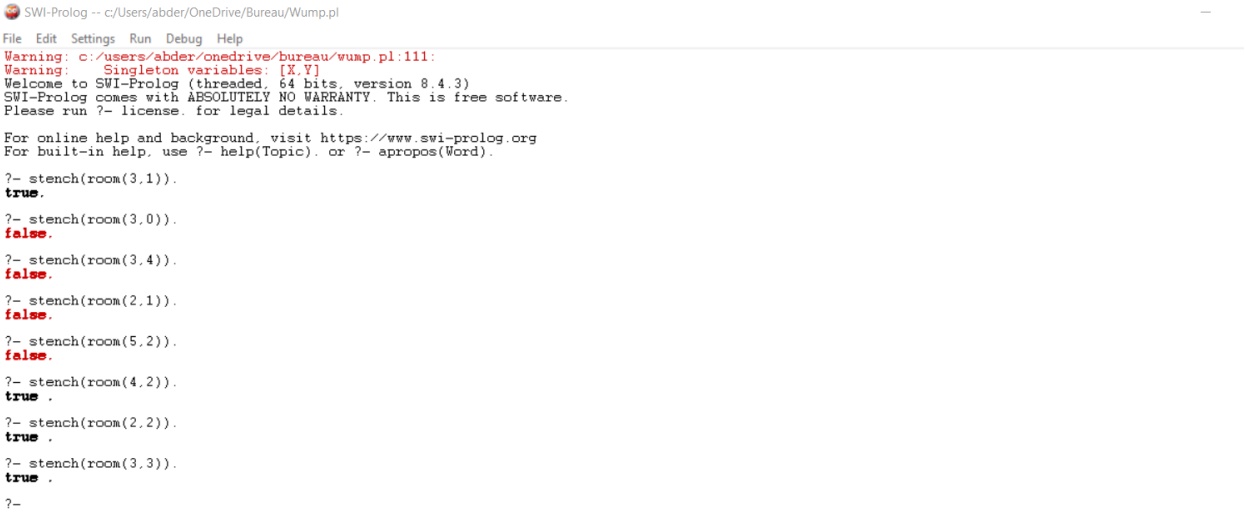
Checking for breeze:

We know that the pit is already declared in pit(room(4,2)).



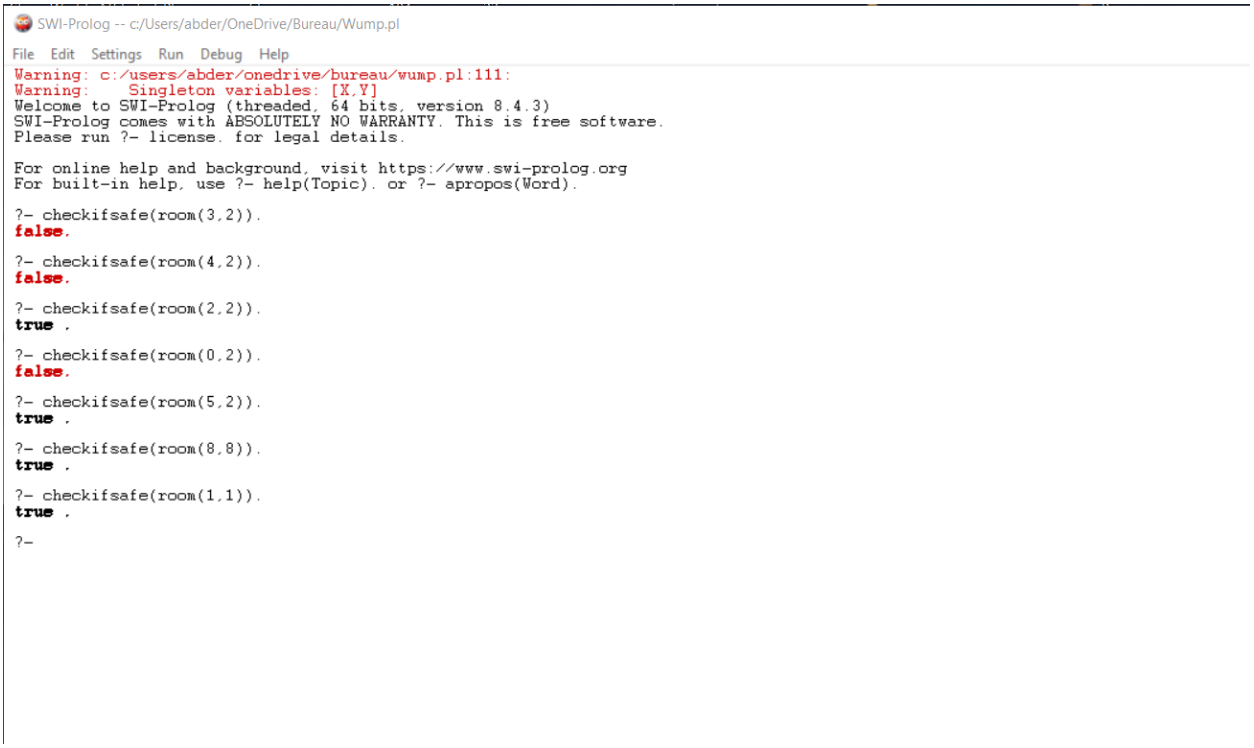
Testing the stench indicator:

Knowing that the Wumpus is in Wumpus(room(3,2)).



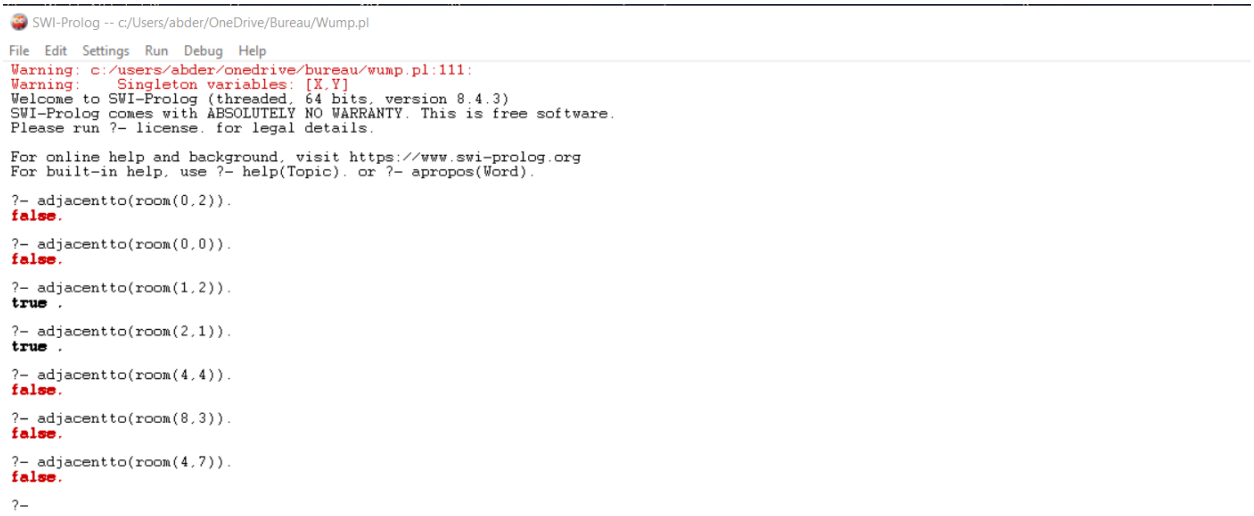
Testing to see if a certain room is safe:

The Wumpus is in room (3,2) and the pit is in room (4,2), so let's test.



Testing Adjacent to:

We have a predeclared room which is room((1,1)).



Testing for checking the four directions for safety:

We have four directions, so we will have to check for the all of them using a starting point already declared which is startingpoint((2,2)).

```
SWI-Prolog -- c:/Users/abder/OneDrive/Bureau/Wump.pl
File  Edit  Settings  Run  Debug  Help
Warning: c:/users/abder/onedrive/bureau/wump.pl:111:
Warning:    Singleton variables: [X,Y]
Welcome to SWI-Prolog (threaded, 64 bits, version 8.4.3)
SWI-Prolog comes with ABSOLUTELY NO WARRANTY. This is free software.
Please run ?- license. for legal details.

For online help and background, visit https://www.swi-prolog.org
For built-in help, use ?- help(Topic). or ?- apropos(Word).

?- checkroomontheleft().
true .

?- checkroomontheright().
false.

?- checkroomonsouth().
true .

?- checkroomonnorth().
true .

?-
```

We can see that the only room that is not safe is on the right which is true since there is a Wumpus in room (3,2).

Testing grab the gold:

The gold is in gold(room(5,7)).

```
?- grabgold(room(5,7)).
Success
% 13,398 inferences, 0.000 CPU in 0.007 seconds (0% CPU, Infinite Lips)

?- grabgold(room(5,3)).
false.

?- grabgold(room(3,3)).
false.

?- grabgold(room(3,1)).
false.

?-
```

Testing shooting the Wumpus:

```
?- shootwumpus(room(4,2)).
% 13,445 inferences, 0.000 CPU in 0.004 seconds (0% CPU, Infinite Lips)
```

```
SWI-Prolog -- c:/Users/abder/OneDrive/Bureau/Wump.pl
File  Edit  Settings  Run  Debug  Help
Warning: c:/users/abder/onedrive/bureau/wump.pl:106:
Warning:    Singleton variables: [T]
Warning: c:/users/abder/onedrive/bureau/wump.pl:112:
Warning:    Singleton variables: [X,Y]
Welcome to SWI-Prolog (threaded, 64 bits, version 8.4.3)
SWI-Prolog comes with ABSOLUTELY NO WARRANTY. This is free software.
Please run ?- license. for legal details.

For online help and background, visit https://www.swi-prolog.org
For built-in help, use ?- help(Topic). or ?- apropos(Word).

?- shootwumpus(room(4,2)).
true .

?- shootwumpus(room(5,2)).
false.

?- shootwumpus(room(3,1)).
true.

?- shootwumpus(room(2,2)).
true
Unknown action: 0 (h for help)
Action? .

?- shootwumpus(room(2,1)).
false.

?- shootwumpus(room(4,2)).
true .

?-
```

Experiment 2:

Testing on the room function:

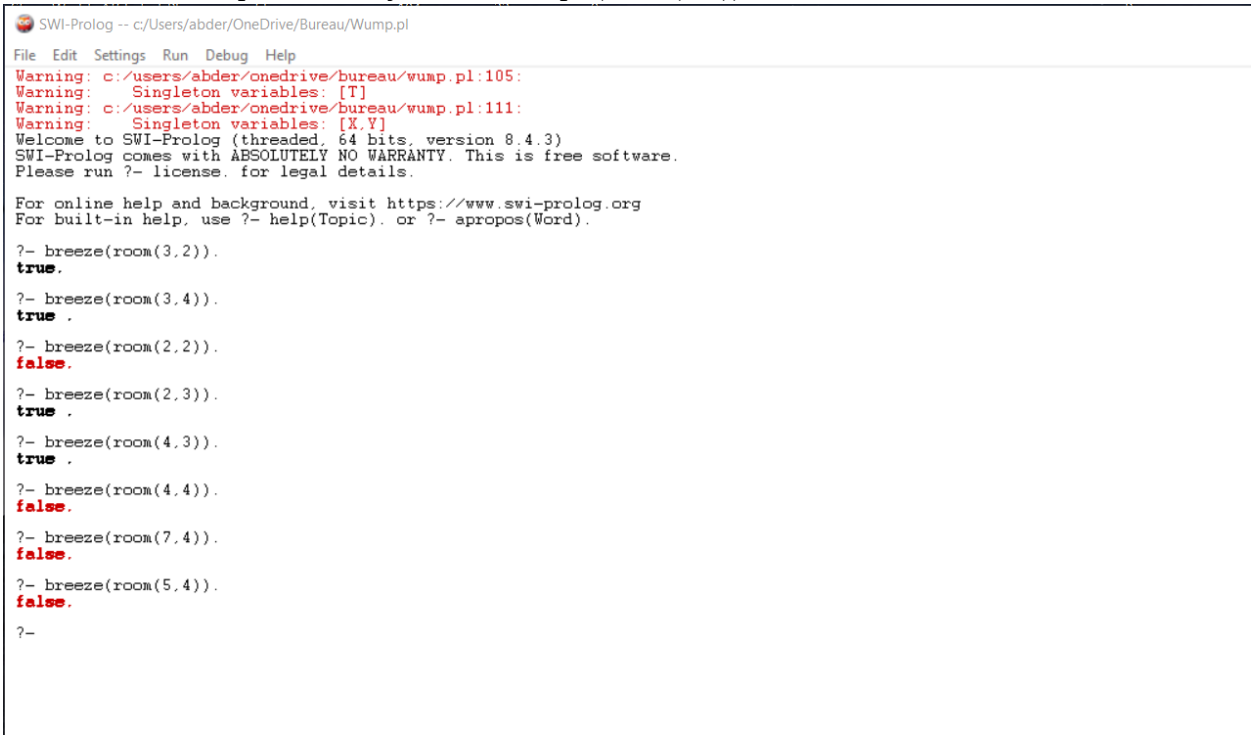
```
SWI-Prolog -- c:/Users/abder/OneDrive/Bureau/Wump.pl
File  Edit  Settings  Run  Debug  Help
Warning: c:/users/abder/onedrive/bureau/wump.pl:105:
Warning:    Singleton variables: [T]
Warning: c:/users/abder/onedrive/bureau/wump.pl:111:
Warning:    Singleton variables: [X,Y]
Welcome to SWI-Prolog (threaded, 64 bits, version 8.4.3)
SWI-Prolog comes with ABSOLUTELY NO WARRANTY. This is free software.
Please run ?- license. for legal details.

For online help and background, visit https://www.swi-prolog.org
For built-in help, use ?- help(Topic). or ?- apropos(Word).

?- room(1,1).
true.

?- room(7,7).
false.

?- room(3,3).
true.

?- room(7,1).
false.

?- room(5,1).
true.

?- room(3,4).
true.

?-
```

```
?- room(3,3).
% 56 inferences, 0.000 CPU in 0.000 seconds (0% CPU, Infinite Lips)
```

The time taken to check if a room exists in the matrix is so low.

Checking for breeze:

We know that the pit is already declared in pit(room(3,3)).




```
?- breeze(room(2,3)).
% 13,445 inferences, 0.016 CPU in 0.004 seconds (398% CPU, 860480 Lips)
```

Testing the stench indicator:

Knowing that the Wumpus is in Wumpus(room(5,5)).

```
?- stench(room(4,5)).
% 13,445 inferences, 0.016 CPU in 0.004 seconds (398% CPU, 860480 Lips)
```

```
File  Edit  Settings  Run  Debug  Help
Warning: c:/users/abder/onedrive/bureau/wump.pl:106:
Warning:    Singleton variables: [T]
Warning: c:/users/abder/onedrive/bureau/wump.pl:112:
Warning:    Singleton variables: [X,Y]
Welcome to SWI-Prolog (threaded, 64 bits, version 8.4.3)
SWI-Prolog comes with ABSOLUTELY NO WARRANTY. This is free software.
Please run ?- license. for legal details.

For online help and background, visit https://www.swi-prolog.org
For built-in help, use ?- help(Topic). or ?- apropos(Word).

?- stench(room(4,5)).
true .

?- stench(room(5,6)).
true .

?- stench(room(5,4)).
true.

?-
|    stench(room(6,5)).
true .

?- stench(room(7,5)).
false.

?- stench(room(1,5)).
false.

?- stench(room(1,1)).
false.

?- stench(room(1,6)).
false.

?-
```

Testing to see if a certain room is safe:

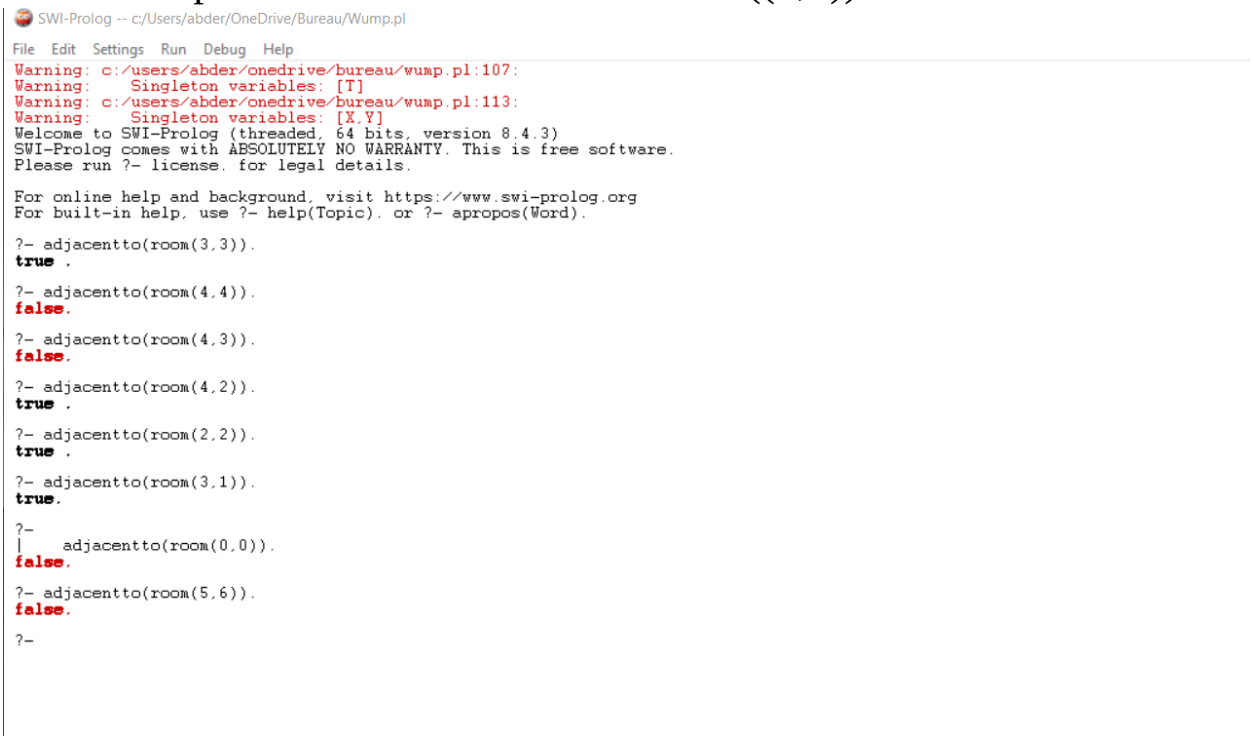
The Wumpus is in room (5,5) and the pit is in room (3,3), so let's test.

```
File  Edit  Settings  Run  Debug  Help
Warning: c:/users/abder/onedrive/bureau/wump.pl:106:
Warning:    Singleton variables: [T]
Warning: c:/users/abder/onedrive/bureau/wump.pl:112:
Warning:    Singleton variables: [X,Y]
Welcome to SWI-Prolog (threaded, 64 bits, version 8.4.3)
SWI-Prolog comes with ABSOLUTELY NO WARRANTY. This is free software.
Please run ?- license. for legal details.

For online help and background, visit https://www.swi-prolog.org
For built-in help, use ?- help(Topic). or ?- apropos(Word).

?- checkifsafe(room(5,5)).
false.

?- checkifsafe(room(3,3)).
false.

?- checkifsafe(room(3,3)).
false.

?- checkifsafe(room(4,8)).
false.

?- checkifsafe(room(5,5)).
false.

?- checkifsafe(room(1,3)).
true
Unknown action: ▯ (h for help)
Action? ,

?- checkifsafe(room(4,5)).
true .

?-
```

```
?- checkifsafe(room(1,3)).
% 13,445 inferences, 0.000 CPU in 0.004 seconds (0% CPU, Infinite Lips)
```

Testing Adjacent to:

We have a predeclared room which is room((3,2)).




```
?- adjacentto(room(3,3)).
% 13,449 inferences, 0.016 CPU in 0.004 seconds (404% CPU, 860736 Lips)
```

Testing for checking the four directions for safety:

We have four directions, so we will have to check for the all of them using a starting point already declared which is startingpoint((5,4)).

```
?- checkroomontheleft().
% 13,461 inferences, 0.000 CPU in 0.004 seconds (0% CPU, Infinite Lips)
```

We can see that the only room that is not safe is in the north which is true since there is a Wumpus in room (5,5).

Testing grab the gold:

The gold is in gold(room(6,6)).

```
?- grabgold(room(6,6)).
% 13,445 inferences, 0.000 CPU in 0.004 seconds (0% CPU, Infinite Lips)
```

SWI-Prolog -- c:/Users/abder/OneDrive/Bureau/Wump.pl

File Edit Settings Run Debug Help

```
Warning: c:/users/abder/onedrive/bureau/wump.pl:111:
Warning:    Singleton variables: [X,Y]
Welcome to SWI-Prolog (threaded, 64 bits, version 8.4.3)
SWI-Prolog comes with ABSOLUTELY NO WARRANTY. This is free software.
Please run ?- license. for legal details.

For online help and background, visit https://www.swi-prolog.org
For built-in help, use ?- help(Topic). or ?- apropos(Word).

?- grabgold(room(6,6)).
Success
true.

?- grabgold(room(6,5)).
false.

?- grabgold(room(5,5)).
false.

?- grabgold(room(5,4)).
false.

?- grabgold(room(1,4)).
false.

?- grabgold(room(1,7)).
false.

?-
```

Testing shooting the Wumpus:

```
?- shootwumpus(room(5,4)).
% 13,461 inferences, 0.000 CPU in 0.004 seconds (0% CPU, Infinite Lips)
```

File Edit Settings Run Debug Help

```
Warning: c:/users/abder/onedrive/bureau/wump.pl:112:
Warning:    Singleton variables: [X,Y]
Welcome to SWI-Prolog (threaded, 64 bits, version 8.4.3)
SWI-Prolog comes with ABSOLUTELY NO WARRANTY. This is free software.
Please run ?- license. for legal details.

For online help and background, visit https://www.swi-prolog.org
For built-in help, use ?- help(Topic). or ?- apropos(Word).

?- shootwumpus(room(5,4)).
true.

?- shootwumpus(room(1,2)).
false.

?- shootwumpus(room(4,5)).
true .

?- shootwumpus(room(6,5)).
true .

?- shootwumpus(room(5,6)).
true .

?- shootwumpus(room(7,6)).
false.

?- shootwumpus(room(1,6)).
false.

?-
```

Limitation and improvements:

There are some things that can be done to make this code much better, such as printing the list of the safe rooms, by giving the starting position or by giving a random point.

Implementing multiple pits can also be an improvement for this code, it can be implemented easily too but just need to think how to avoid bugs and keeping the code clear to understand.

Adding the direction in which the player is looking at can also be a nice improvement but that will change a lot on how we shoot the Wumpus since we will need to take the direction of the player into consideration.

The code is too simple, it needs some work to become able to solve every possible situation such as 2 adjacent pits.

The code needs always input from the user about which rooms are going to be visited, making it automatic can be a good quality of life improvement.

I tried to go over the code multiple times but there is one problem that remained which is the state of life for the Wumpus since I planed for it to chance from 0 to 1 when the player shoots the Wumpus but sadly I could not implement it correctly and it gave too many bugs, so I gave up on it at the end and stick with one value which is 0.

I couldn't detect the time efficiency for this code since it is so low, so I could not really analyze the time needed for each part efficiently.

I tried to automatically generate the gold in a random room, a Wumpus in a random room, a pit in a random room with a random starting point but I could not implement it, I though about using

gold(room(random_between(1,8,X)),random_between(1,8,Y)).