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

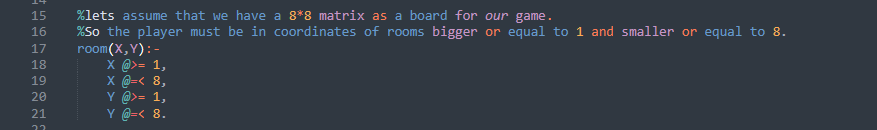
DR. Tajjeeddine Rachidi

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 ….

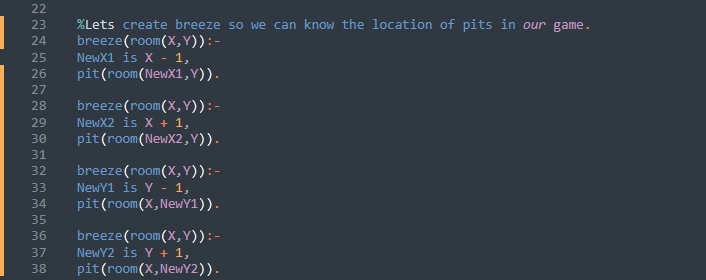
**Key Predicates, The meaning of variables:**

**Room:**



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:**



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:**

****

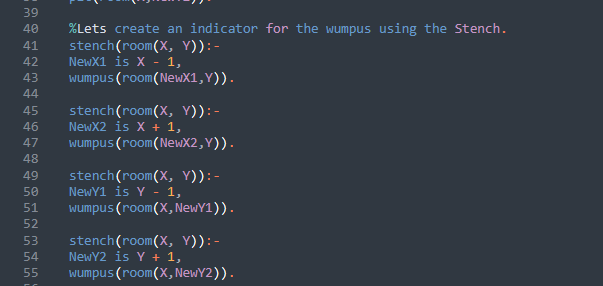
We created a pit with already known coordinates in order to simplify the testing.

**Wumpus:**

****

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

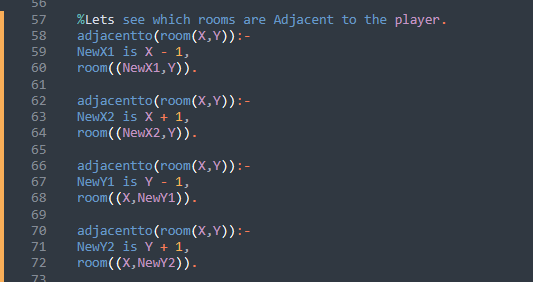
**Stench:**



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:**

****

****

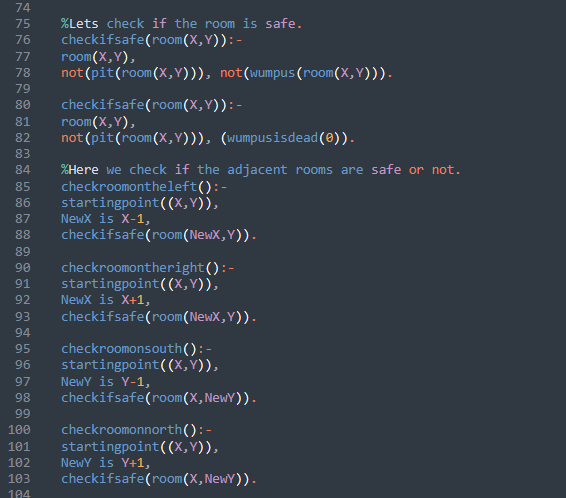
We created a function to check if a room given by the user is adjacent to a predeclared room, in this case the predeclared room is (1,1).

**Gold:**

****

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

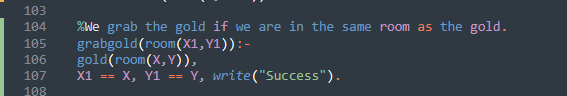
**Safe:**

****



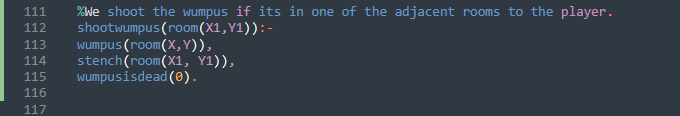
In this function, we check if a room is safe, the coordinates of the room are given by the user and they are used to check if the room is safe, by checking if there is no pit in the room and if there is no Wumpus in the room or if its dead. Moreover, we can check if a room to the right, left, north or 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:**

****

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:**

****

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)).

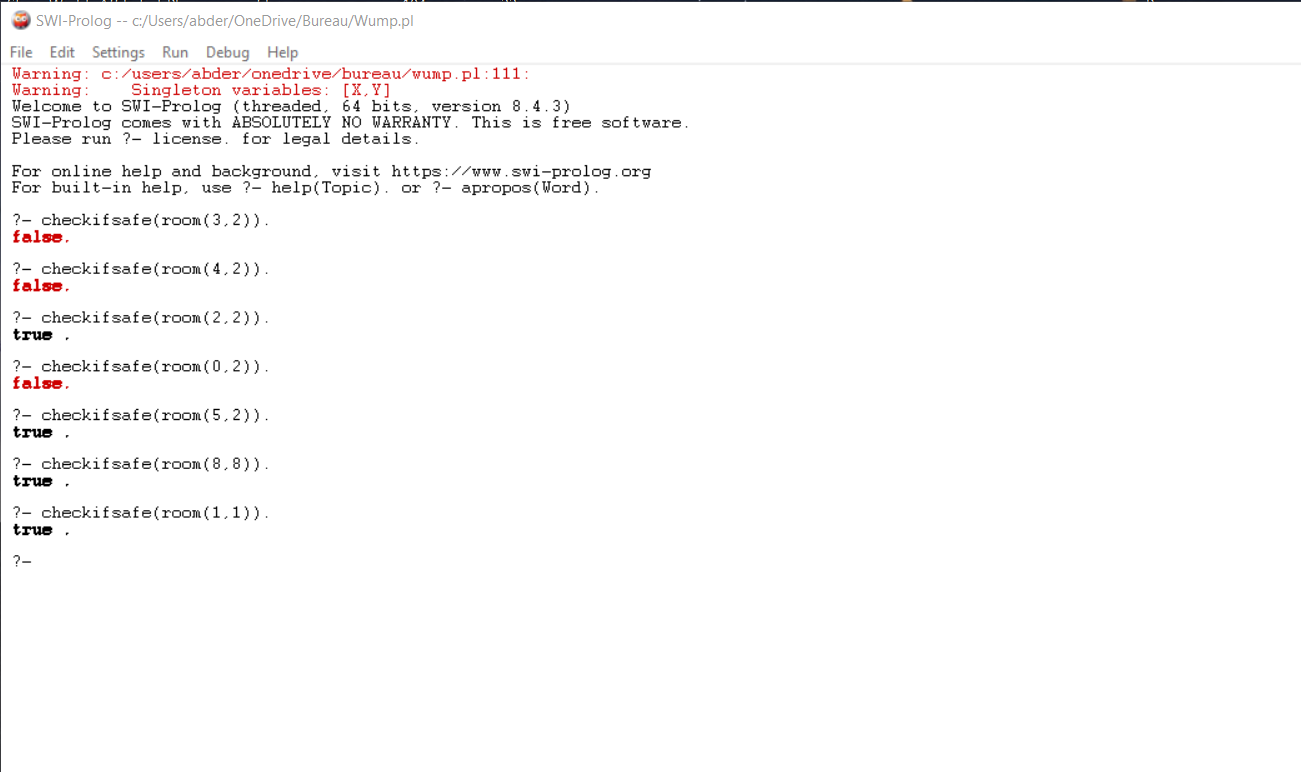


**Testing the stench indicator:**

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

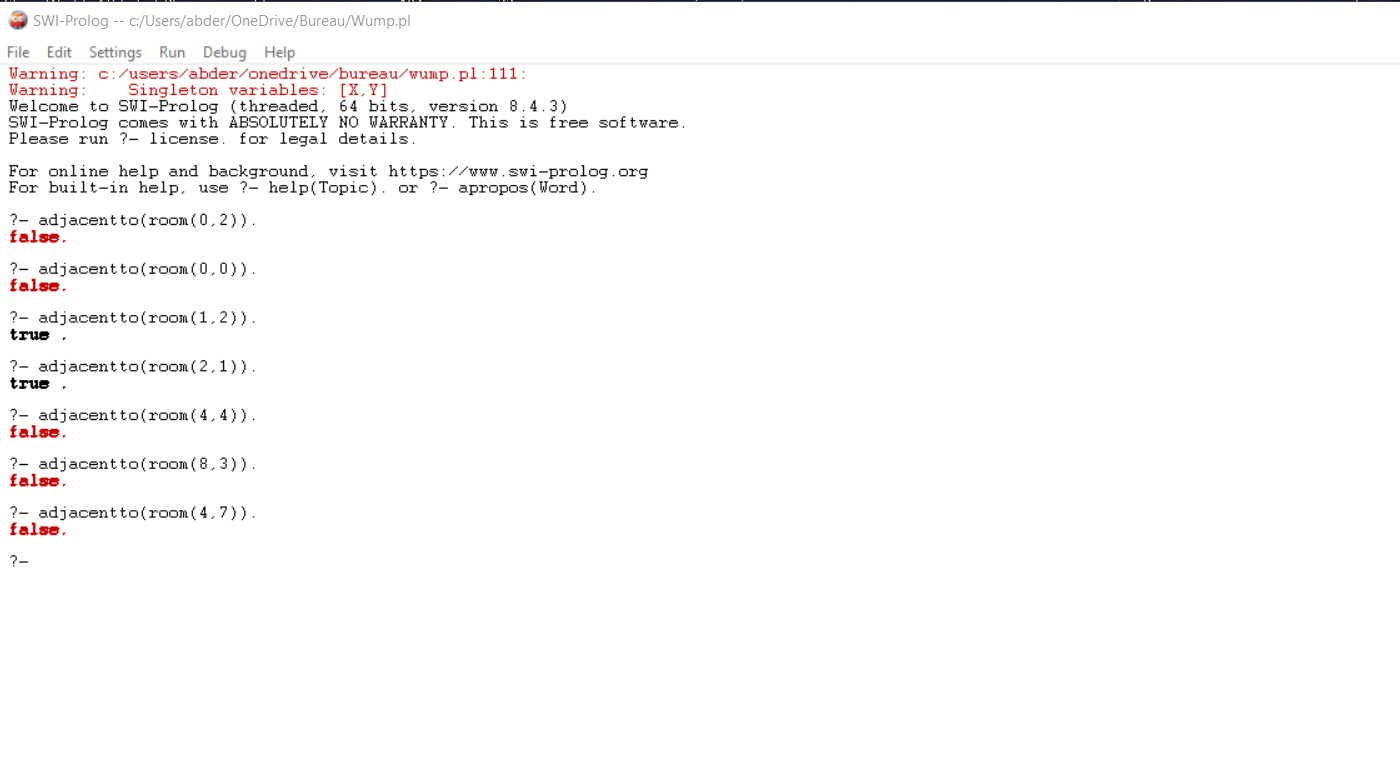


**Testing to see if a certain room is safe:**

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

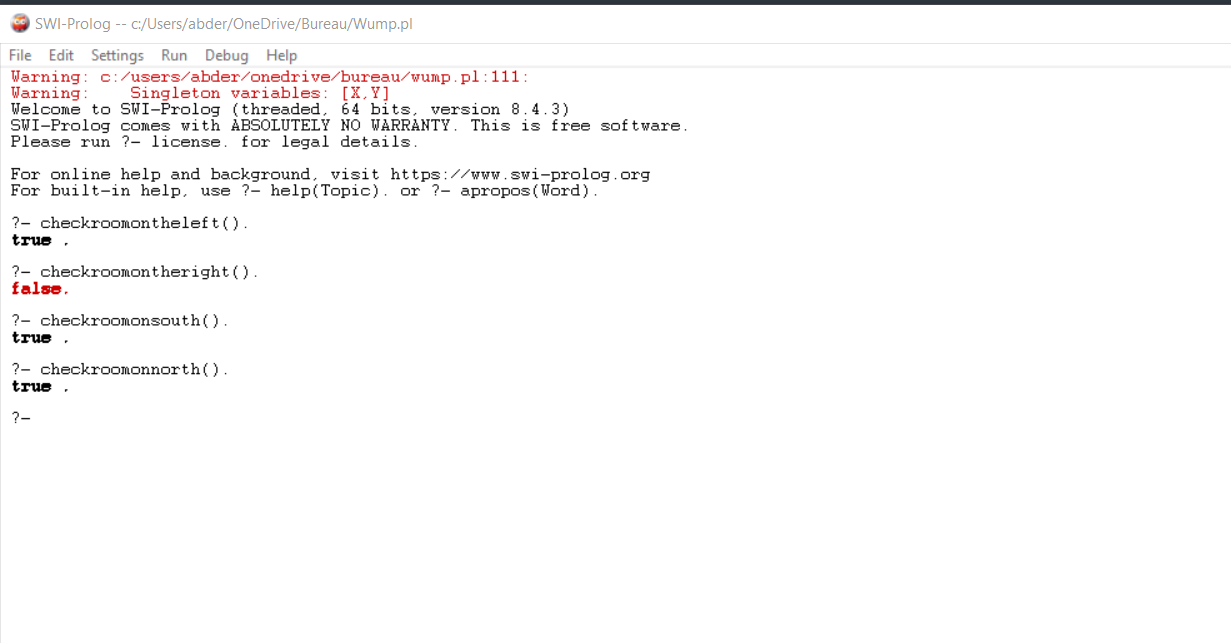
**Testing Adjacent to:**

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)).



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).

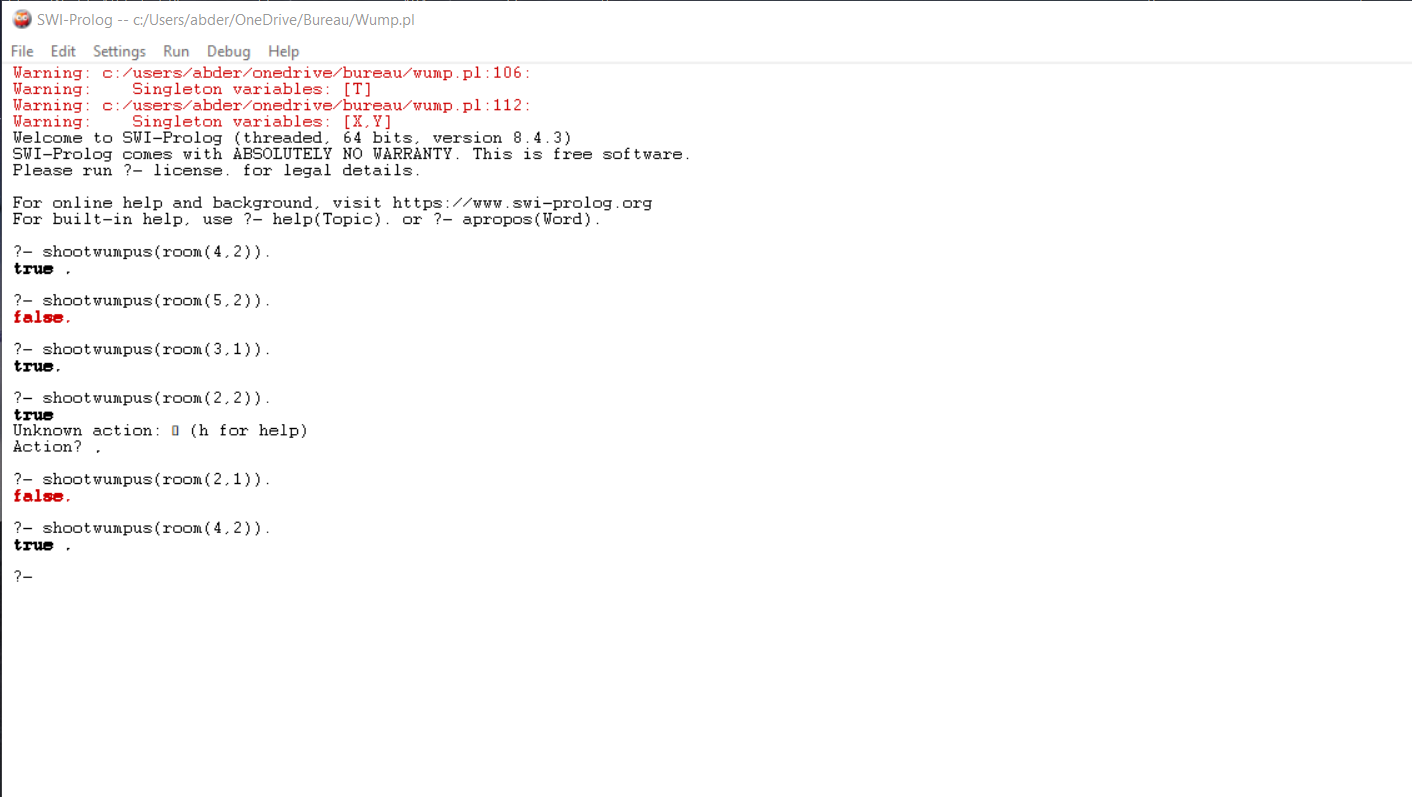
**Testing grab the gold:**

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



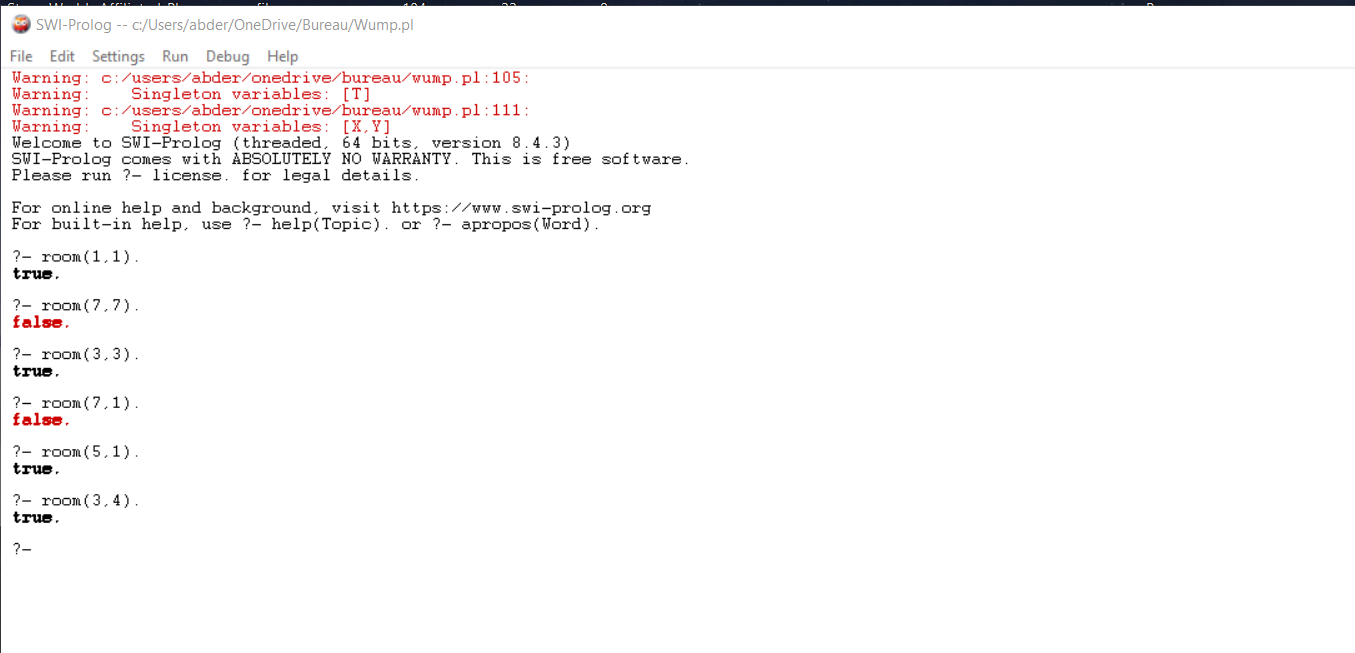
**Testing shooting the Wumpus:**





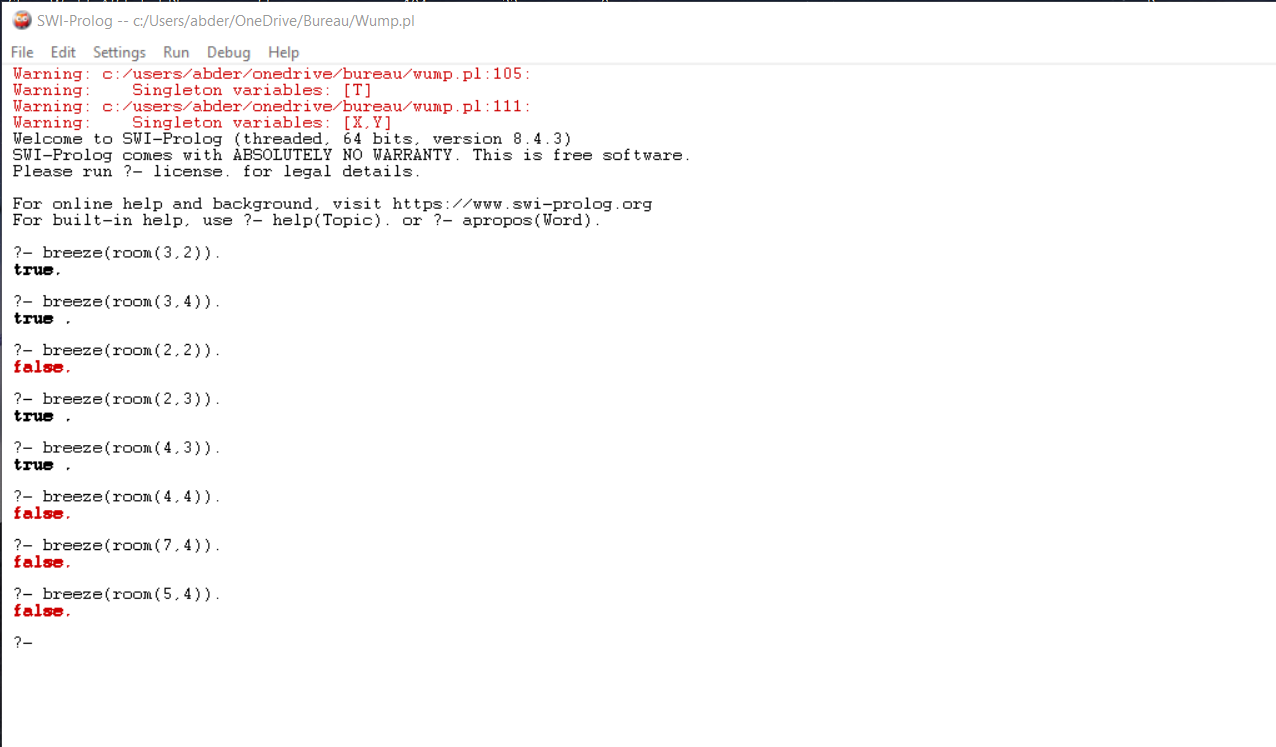
**Experiment 2:**

**Testing on the room function:**



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

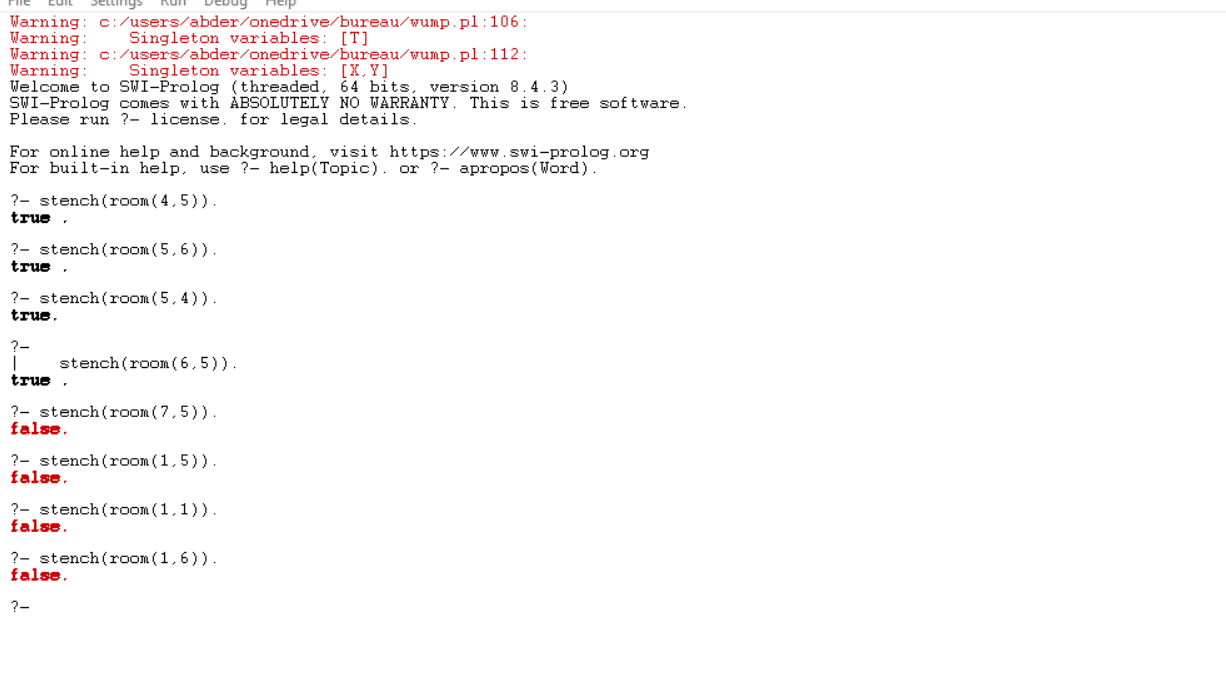
**Checking for breeze:**

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

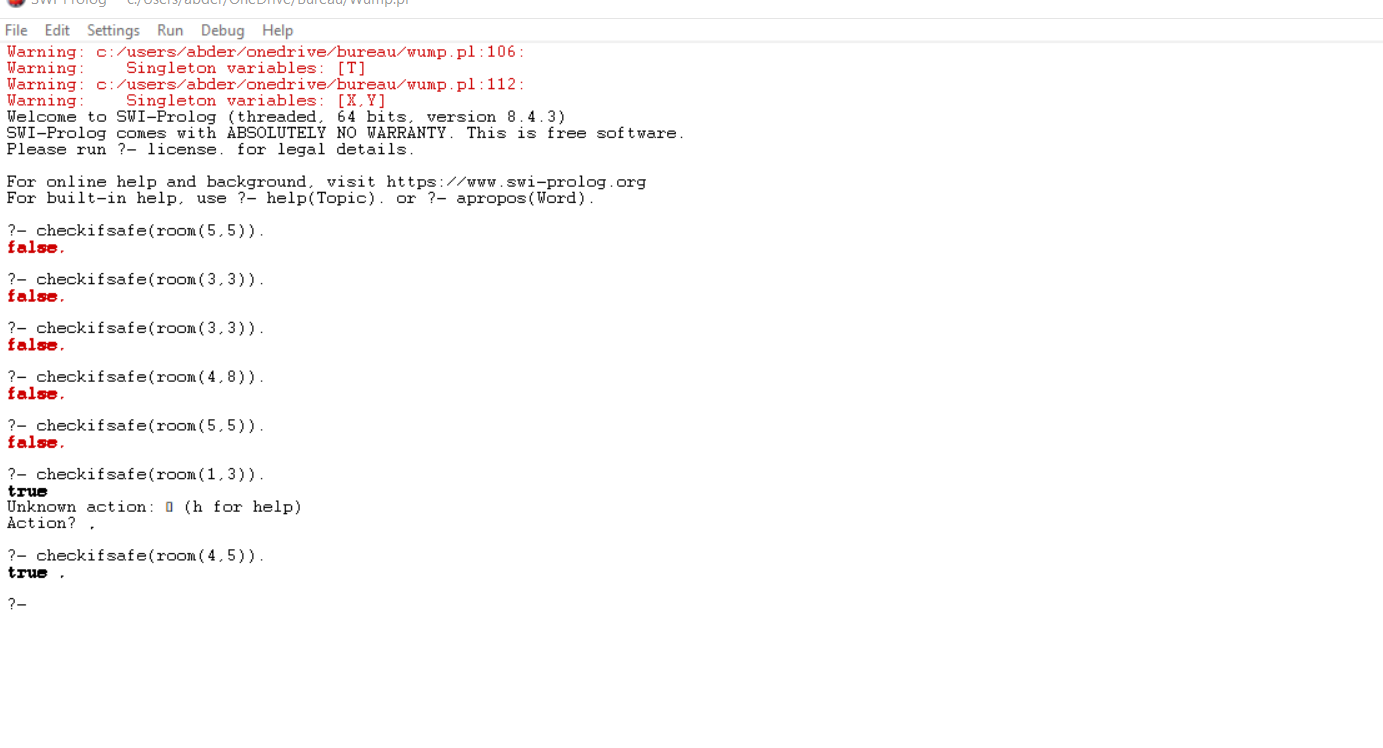


**Testing the stench indicator:**

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

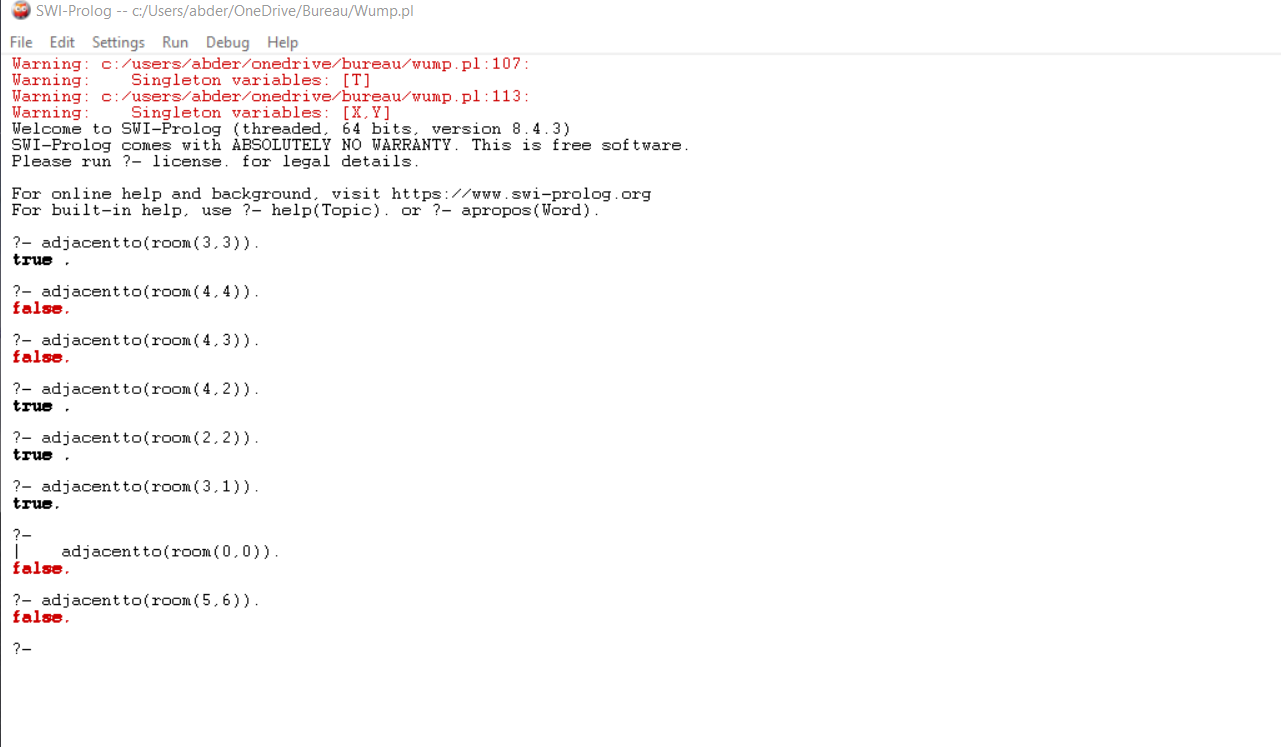


**Testing to see if a certain room is safe:**

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



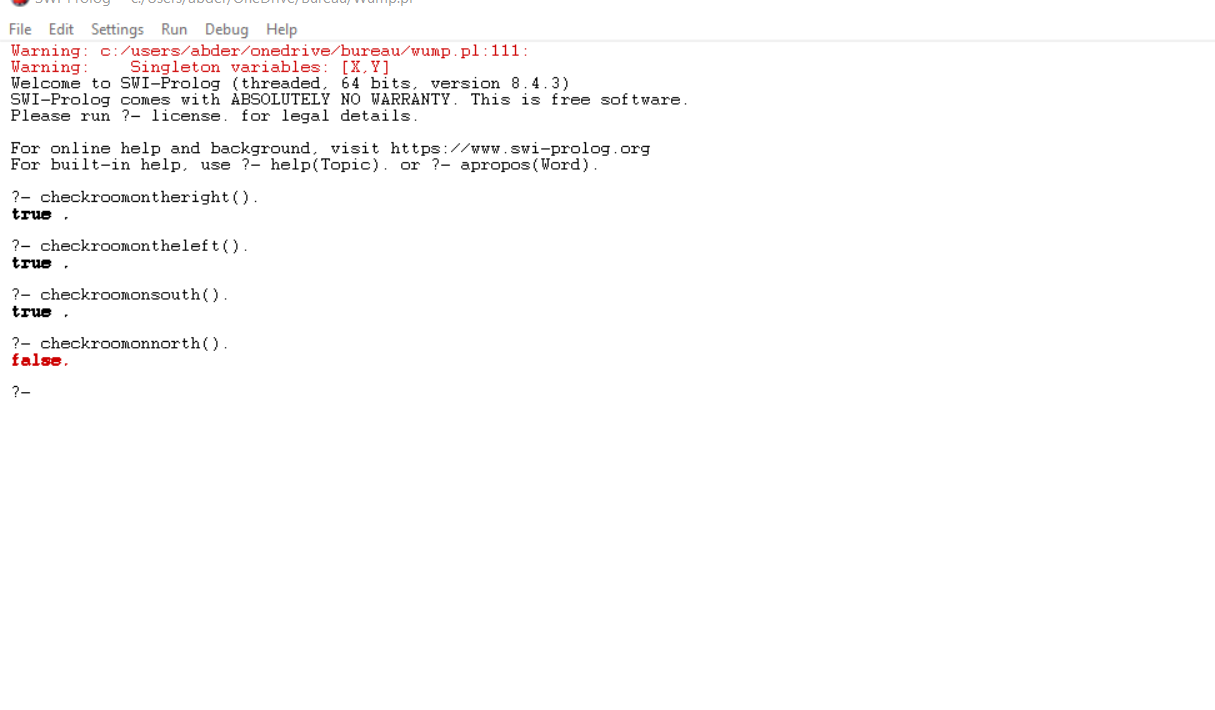
**Testing Adjacent to:**

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



**Testing for checking the four directions for safety:**

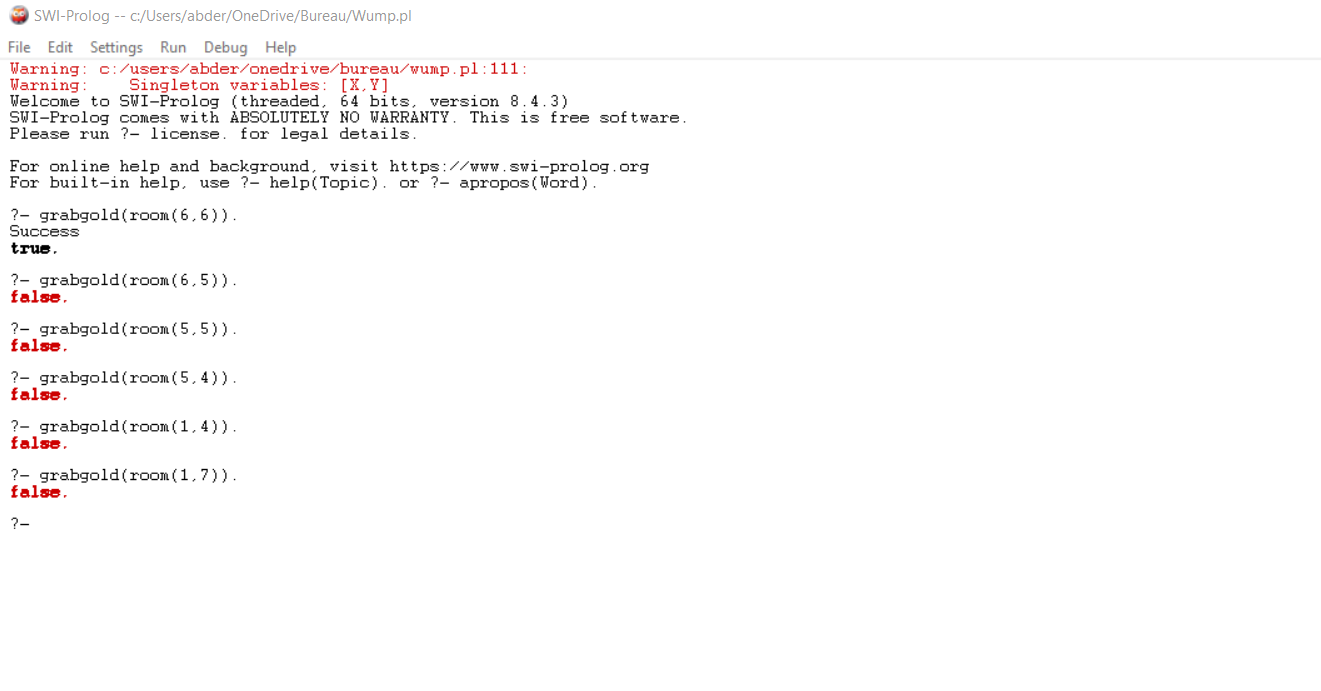
We have four directions, so we will have to check for the all of them using a starting point already declared which is startingpoint((5,4)).



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)).



**Testing shooting the Wumpus:**





**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)).