جامعة الأخوين

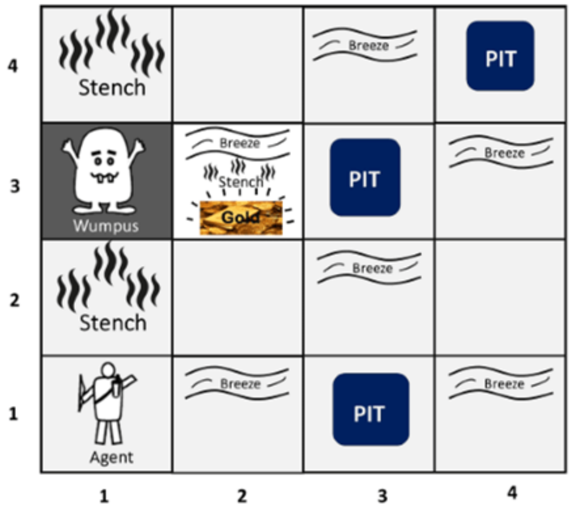
AL AKHAWAYN
UNIVERSITY

AI project 2: Logical agent

Ouwais Zlaigi & Mohamed Moumou

06/29/2000

Introduction:



The Wumpus world is a simple world example to illustrate the worth of a knowledgebased agent and to represent knowledge representation. It was inspired by a video game Hunt the Wumpus by Gregory Yob in 1973.

Implementation/Used technologies:

Inspiration:

Following the workflow of our course, we went over the logical that represents knowledge of its world, its goals and the current situation by sentences in logic and decide what to do by inferring that a certain action or course of action is appropriate to achieve its goals, and that exactly what SWI-PROLOG provides us to do.

Swi-prolog:

SWI Prolog

SWI-Prolog offers a comprehensive free Prolog environment. Since its start in 1987, SWIProlog development has been driven by the needs of real world applications. SWI-Prolog is widely used in research and education as well as commercial applications.

Project aim:

The aim is to build a single iteration of a logical agent, that given (TOLD) a starting position and ASKED about one single action (safety of rooms, shooting, or picking up gold) will return a reply. That is, your agent will not try to work out the best action, but just answer questions ASKED by a human given a knowledge base (rules of the game and the starting configuration).

Predicates:

SetHunterAt(): set the hunter in a certain position that take 3 parameters (X,Y,d(direction)). For example: hunter at (1,1) with a direction: N (north).

setWumpusAt(): set the Wumpus in a certain position.

SetPitsPos(): sit the pits position.(takes 1 parameter)

SetGoldPos(): set the gold in a certain position. (takes 1 parameter)

GetAdjacentRoom(): Getting the list of all adjacent rooms to a specified room. (takes 2 parameters)

AdjacentTo(): Checking whether a room is adjacent to another.(takes 2 parameters)

Breeze(): checking whether there is a breeze at specific position. If invoked as breeze(r(X,Y)), it will list all the rooms having a breeze. (takes 1 parameter)

Stench(): Checking whether there is a stench at specific position. If invoked as stench(r(X,Y)), it will list all the rooms having a stench. (takes 1 parameter)

SafeRoomCheck():Checking whether a room is safe or not.(Takes 1 parameter)

Safe(): Checking whether adjacent rooms to the hunter's current position are safe.

SafeWumpus(): check if the Wumpus is not adjacent to the hunter, it is safe.

WumpusAlive(): check the Wumpus if is alive.

HasArrow(): check if the hunter has the arrow or not.

WallCheck(): checking walls.

ShootWumpus(): Shoot wumpus without checking the direction of the hunter and whether it faces the wumpus.

ShootWumpusWithDirection(): Shhot wumpus while taking into account the direction of the hunter. If the hunter is facing the wumpus, they can kill the Wumpus.

GrabGold(): Grabbing the gold if the hunter is at the gold's position.

TurnLeft(): Turning the hunter to the left.

TurnRight():Turning the hunter to the right.

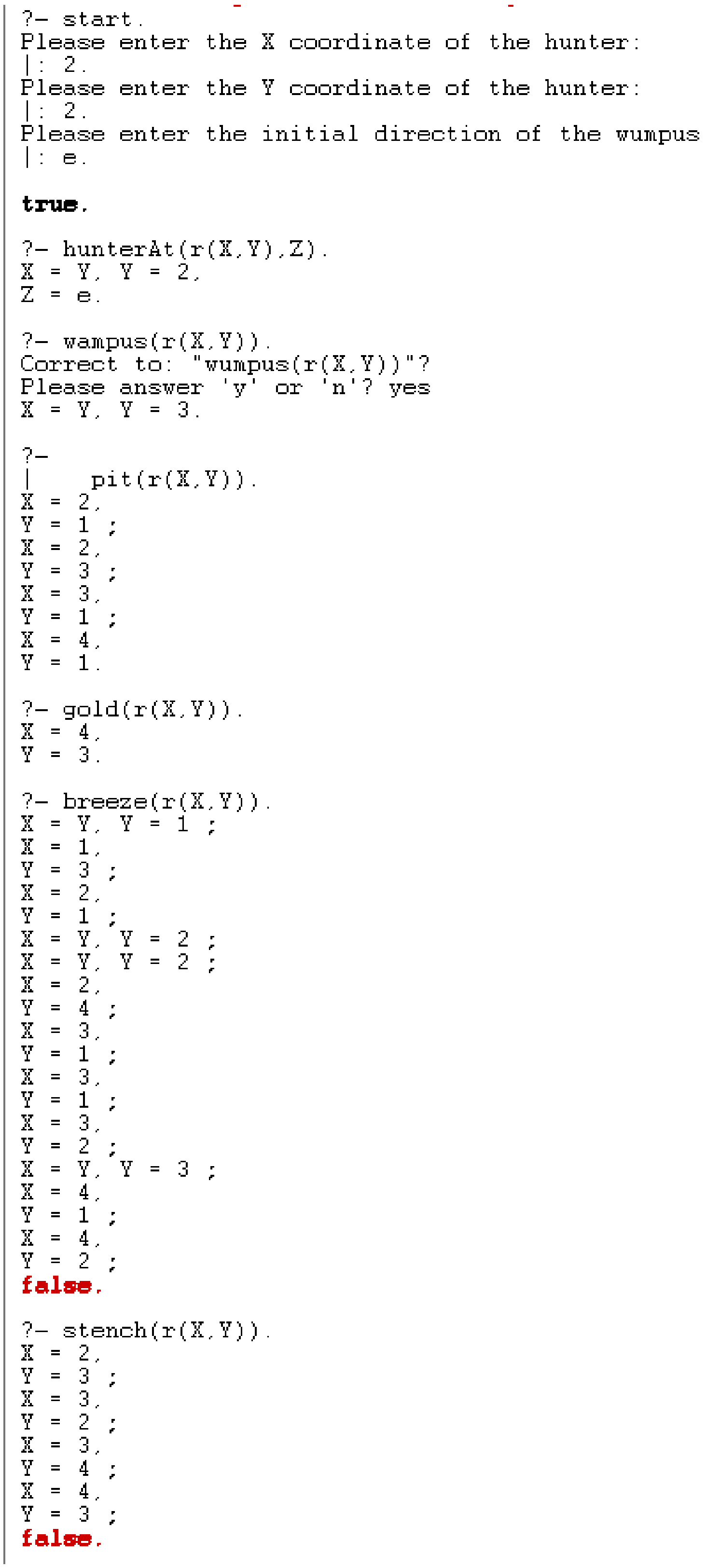
Let us assume that we are in direction N (north) and we want to kill the Wumpus who is in our left, so we will just turn left to the west and vice versa.

Start: initializing the game: Asking for the postion of the hunter, setting the pits positions, setting the wumpus position, and the gold's position.

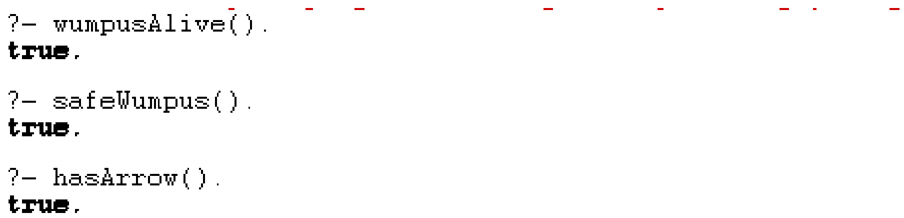
Experiments:

So far we have 5 experiments with different configurations.

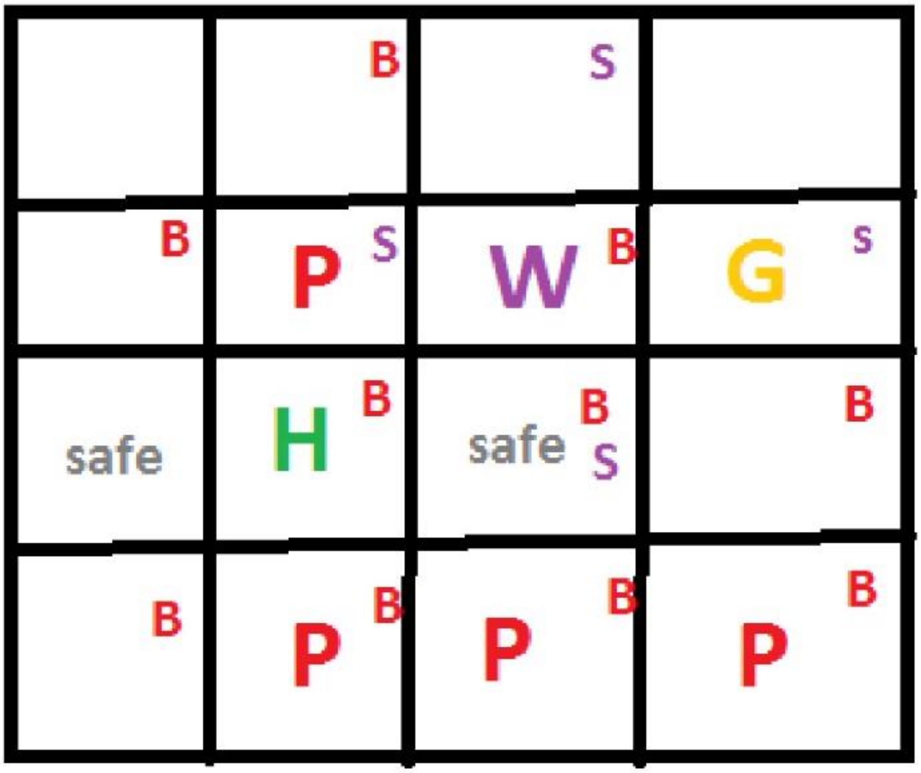
Number 1:



```
?- start.
Please enter the X coordinate of the hunter:
|: 2.
Please enter the Y coordinate of the hunter:
|: 2.
Please enter the initial direction of the wumpus
|: e.

true.

?- hunterAt(r(X,Y),Z).
X = Y, Y = 2,
Z = e.

?- wampus(r(X,Y)).
Correct to: "wumpus(r(X,Y))"?
Please answer 'y' or 'n'? yes
X = Y, Y = 3.

?-
|    pit(r(X,Y)).
X = 2,
Y = 1 ;
X = 2,
Y = 3 ;
X = 3,
Y = 1 ;
X = 4,
Y = 1.

?- gold(r(X,Y)).
X = 4,
Y = 3.

?- breeze(r(X,Y)).
X = Y, Y = 1 ;
X = 1,
Y = 3 ;
X = 2,
Y = 1 ;
X = Y, Y = 2 ;
X = Y, Y = 2 ;
X = 2,
Y = 4 ;
X = 3,
Y = 1 ;
X = 3,
Y = 1 ;
X = 3,
Y = 2 ;
X = Y, Y = 3 ;
X = 4,
Y = 1 ;
X = 4,
Y = 2 ;
false.

?- stench(r(X,Y)).
X = 2,
Y = 3 ;
X = 3,
Y = 2 ;
X = 3,
Y = 4 ;
X = 4,
Y = 3 ;
false.
```

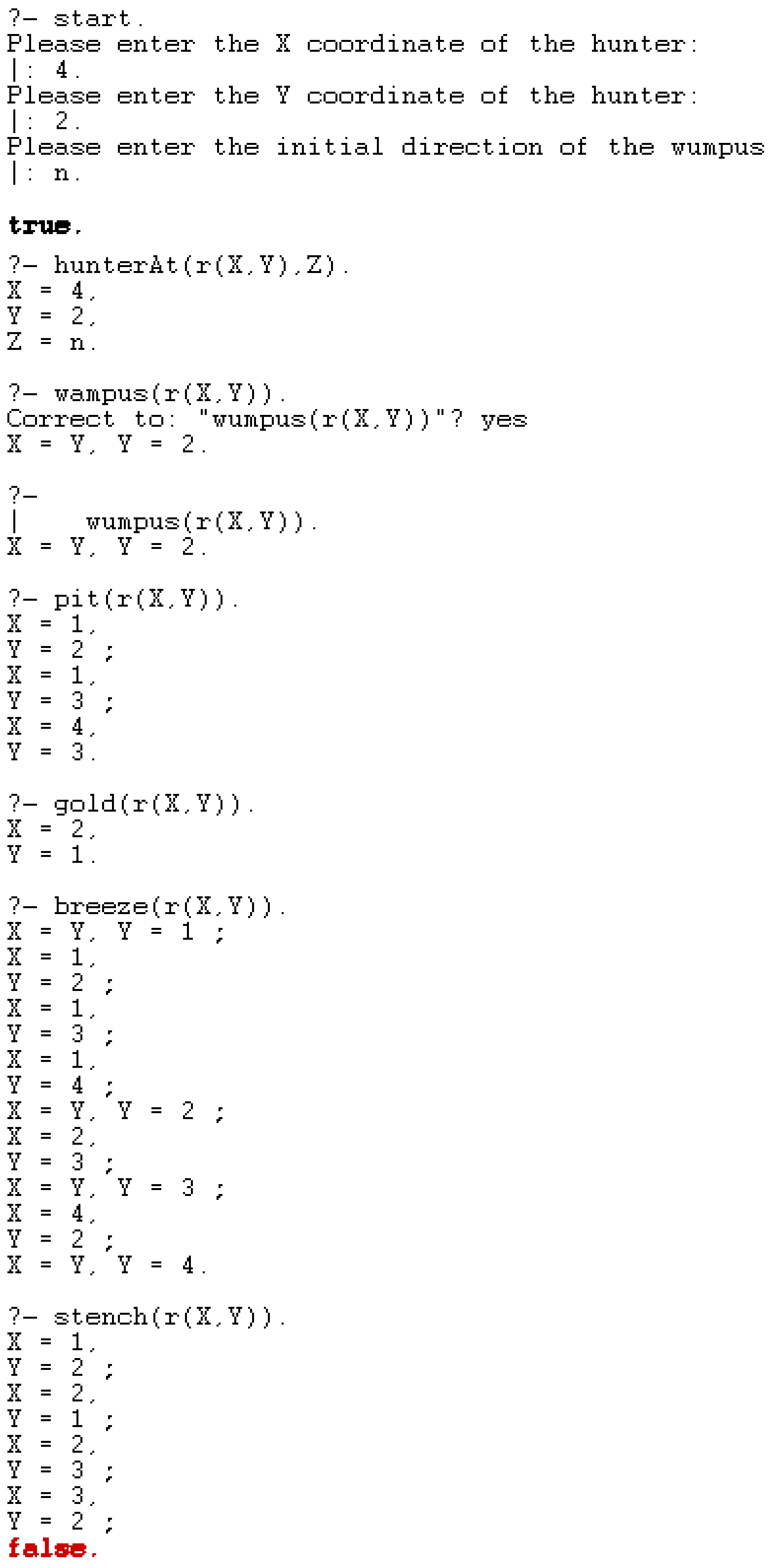


```
?- wumpusAlive().
true.

?- safeWumpus().
true.

?- hasArrow().
true.
```

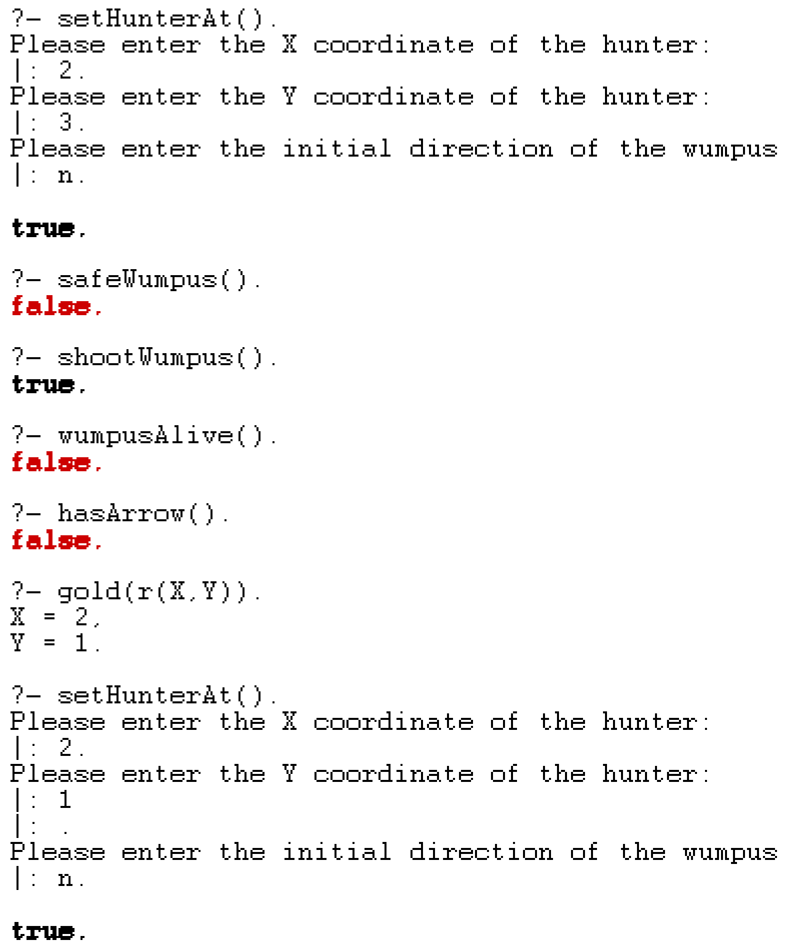


As stated in this figure, the initial position of the hunter is (2,2) and Wumpus (3,3) and for pits are all the position where we have P. the B is where we have the breeze and these breeze room are the adjacent of the pit room obviously. And Gold position is G. For sure, the Wumpus is alive because it is not adjacent to the hunter so the return will be true as stated in screenshot. Hence, the hunter still have the arrow because the Wumpus. Eventually, the function safe return all the rooms that does not consist on the pit or Wumpus.

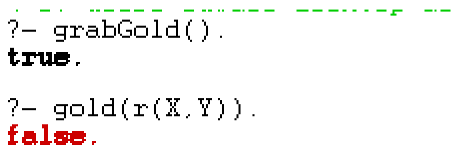
Number2:



```
?- start.
Please enter the X coordinate of the hunter:
|: 4.
Please enter the Y coordinate of the hunter:
|: 2.
Please enter the initial direction of the wumpus
|: n.

true.

?- hunterAt(r(X,Y),Z).
X = 4,
Y = 2,
Z = n.

?- wampus(r(X,Y)).
Correct to: "wumpus(r(X,Y))"? yes
X = Y, Y = 2.

?-
|    wumpus(r(X,Y)).
X = Y, Y = 2.

?- pit(r(X,Y)).
X = 1,
Y = 2 ;
X = 1,
Y = 3 ;
X = 4,
Y = 3.

?- gold(r(X,Y)).
X = 2,
Y = 1.

?- breeze(r(X,Y)).
X = Y, Y = 1 ;
X = 1,
Y = 2 ;
X = 1,
Y = 3 ;
X = 1,
Y = 4 ;
X = Y, Y = 2 ;
X = 2,
Y = 3 ;
X = Y, Y = 3 ;
X = 4,
Y = 2 ;
X = Y, Y = 4.

?- stench(r(X,Y)).
X = 1,
Y = 2 ;
X = 2,
Y = 1 ;
X = 2,
Y = 3 ;
X = 3,
Y = 2 ;
false.
```



```
?- setHunterAt().
Please enter the X coordinate of the hunter:
|: 2.
Please enter the Y coordinate of the hunter:
|: 3.
Please enter the initial direction of the wumpus
|: n.

true.

?- safeWumpus().
false.

?- shootWumpus().
true.

?- wumpusAlive().
false.

?- hasArrow().
false.

?- gold(r(X,Y)).
X = 2,
Y = 1.

?- setHunterAt().
Please enter the X coordinate of the hunter:
|: 2.
Please enter the Y coordinate of the hunter:
|: 1
|: .
Please enter the initial direction of the wumpus
|: n.

true.
```

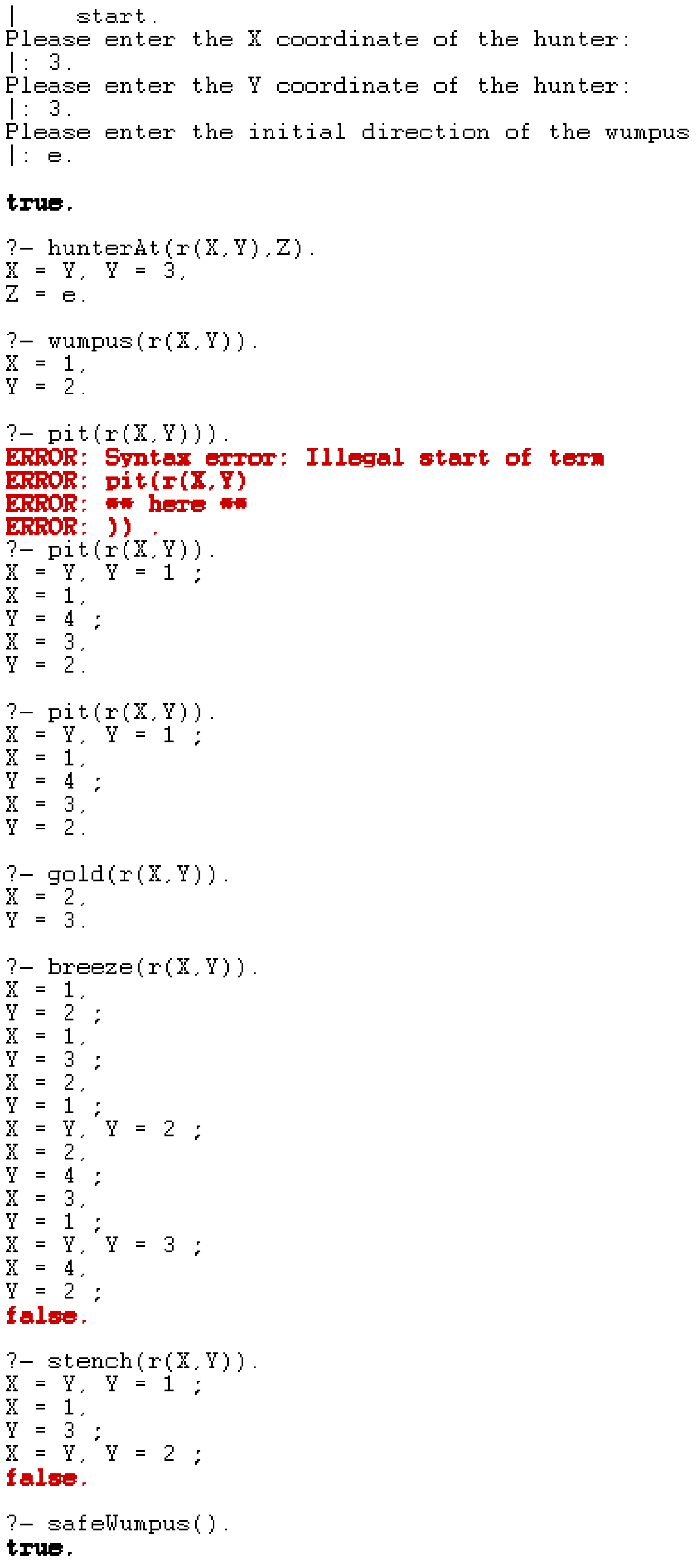


```
?- grabGold().
true.

?- gold(r(X,Y)).
false.
```

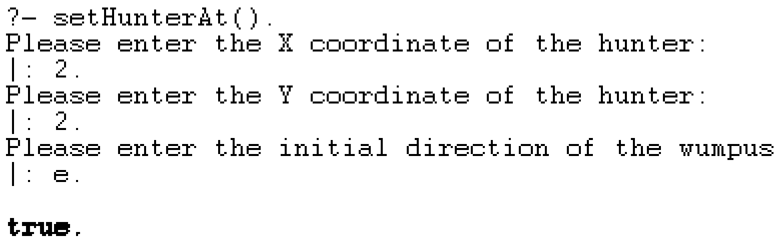
B			B
P B	B s	B	P
P B s	W B	safe s	H B
B	G s		safe

In this configuration we set the hunter, Wumpus, and pits in different positions. In addition, we went over new functions that were not in config1. After the initial position of the hunter, we set hunter to a new position to shoot the Wumpus. In order the Wumpus be killed, the hunter must be adjacent to Wumpus. After that, once the hunter and the Wumpus are adjacent, obviously the Wumpus is not safe. Therefore, we got the false return in safeWumpus(),then we shoot the Wumpus and check whether is alive or not (of course it is not live as stated , false return). Once the Wumpus is died the hunter has no arrow which is the case (false return in has Arrow() function). Finally, we set a new position to the hunter in order to grab the gold.

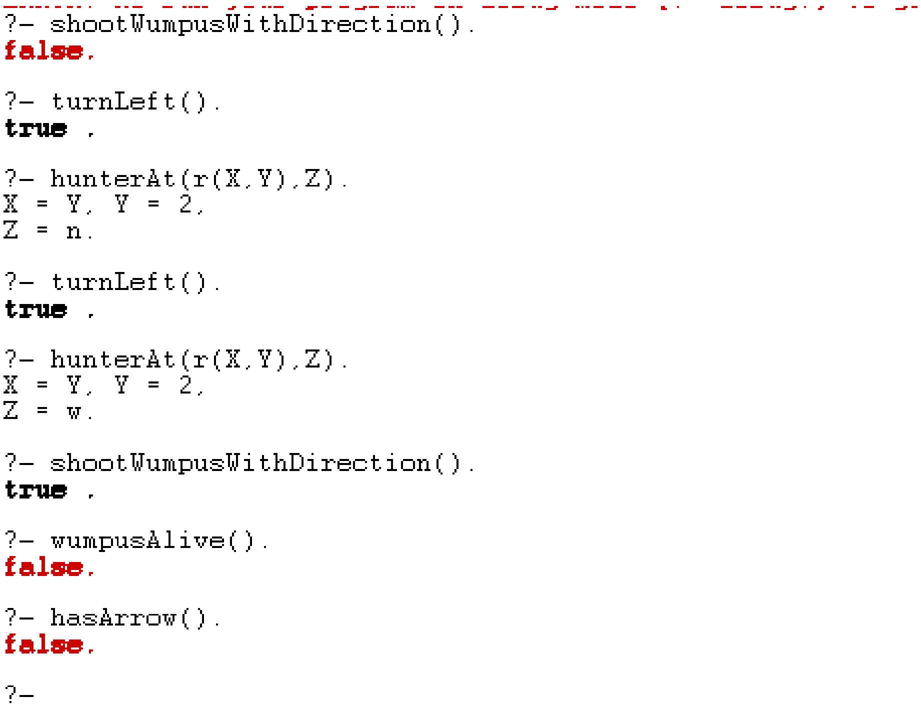
Number3:



```
|    start.
Please enter the X coordinate of the hunter:
|: 3.
Please enter the Y coordinate of the hunter:
|: 3.
Please enter the initial direction of the wumpus
|: e.

true.

?- hunterAt(r(X,Y),Z).
X = Y, Y = 3,
Z = e.

?- wumpus(r(X,Y)).
X = 1,
Y = 2.

?- pit(r(X,Y))).
ERROR: Syntax error: Illegal start of term
ERROR: pit(r(X,Y)
ERROR: ** here **
ERROR: )) .
?- pit(r(X,Y)).
X = Y, Y = 1 ;
X = 1,
Y = 4 ;
X = 3,
Y = 2.

?- pit(r(X,Y)).
X = Y, Y = 1 ;
X = 1,
Y = 4 ;
X = 3,
Y = 2.

?- gold(r(X,Y)).
X = 2,
Y = 3.

?- breeze(r(X,Y)).
X = 1,
Y = 2 ;
X = 1,
Y = 3 ;
X = 2,
Y = 1 ;
X = Y, Y = 2 ;
X = 2,
Y = 4 ;
X = 3,
Y = 1 ;
X = Y, Y = 3 ;
X = 4,
Y = 2 ;
false.

?- stench(r(X,Y)).
X = Y, Y = 1 ;
X = 1,
Y = 3 ;
X = Y, Y = 2 ;
false.

?- safeWumpus().
true.
```

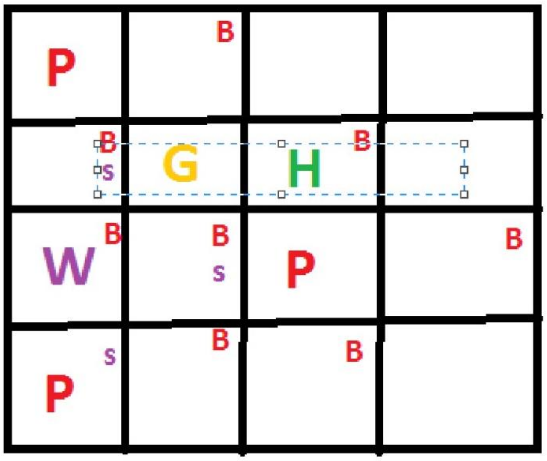


```
?- setHunterAt().
Please enter the X coordinate of the hunter:
|: 2.
Please enter the Y coordinate of the hunter:
|: 2.
Please enter the initial direction of the wumpus
|: e.

true.
```



```
?- shootWumpusWithDirection().
false.

?- turnLeft().
true .

?- hunterAt(r(X,Y),Z).
X = Y, Y = 2,
Z = n.

?- turnLeft().
true .

?- hunterAt(r(X,Y),Z).
X = Y, Y = 2,
Z = w.

?- shootWumpusWithDirection().
true .

?- wumpusAlive().
false.

?- hasArrow().
false.

?-
```

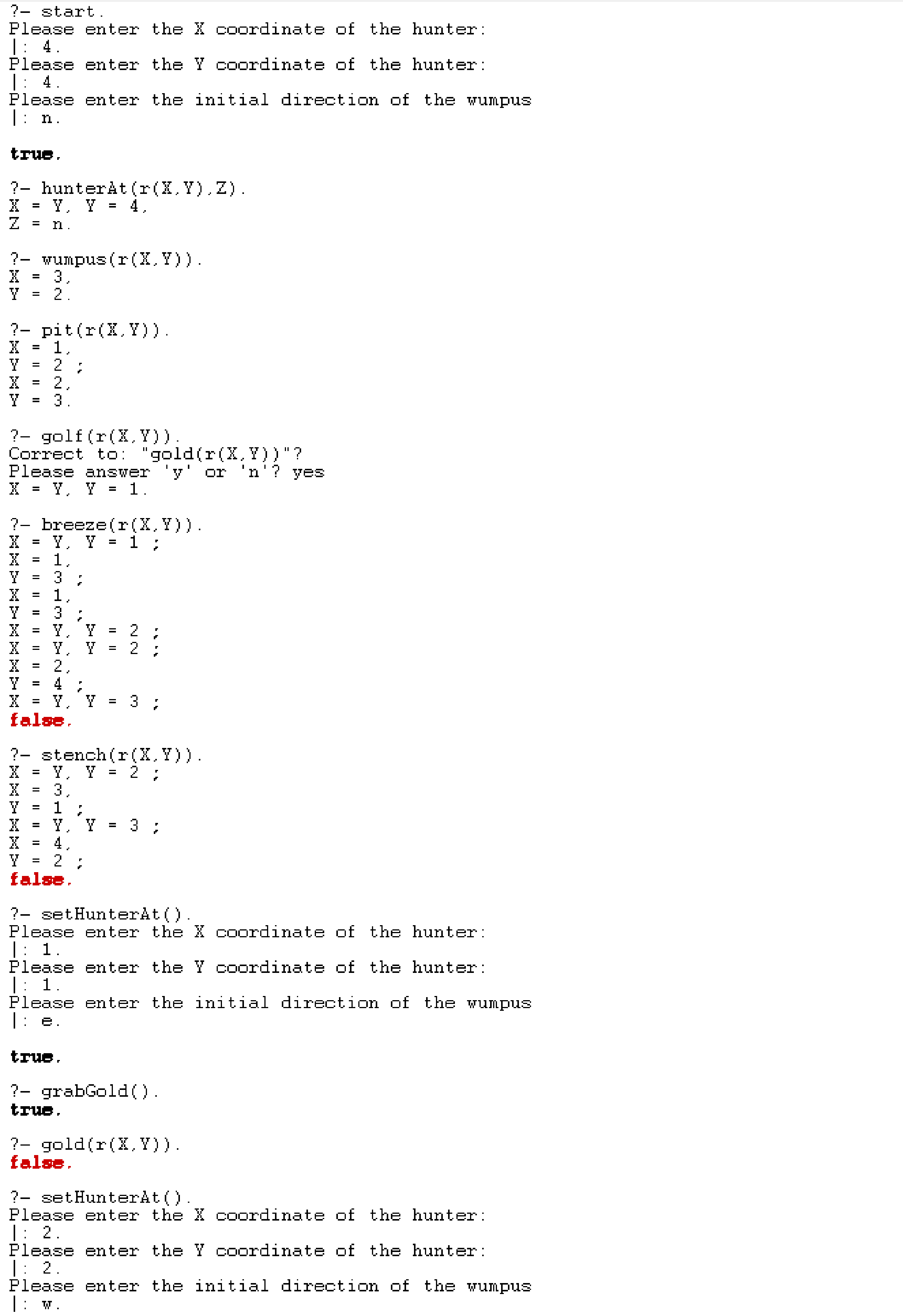
NB: the first screenshot has an error; is just a mistyping.



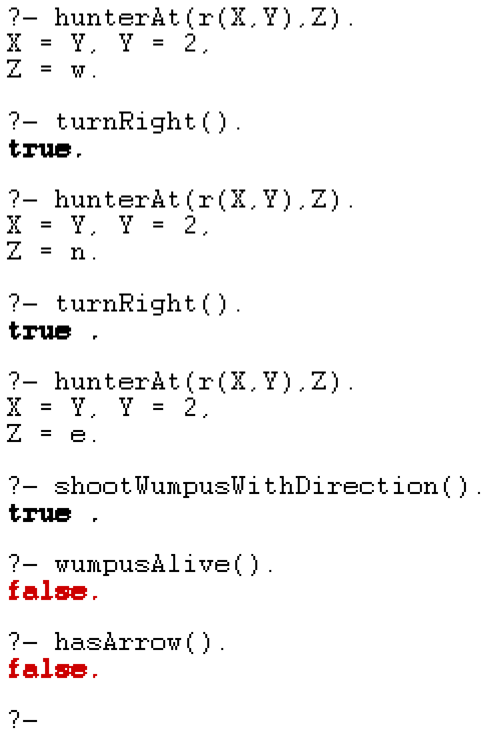
In this configuration was mainly about shootwumpuswithDirection.

Shoot Wumpus with direction required the appropriate direction in order to kill the Wumpus, so the adjacent of the Wumpus and hunter is not enough without the right direction. Initially, the hunter was in (2,2) with an east direction. Therefore, in order to kill the Wumpus we need to turn left to kill it (calling turnLeft()).

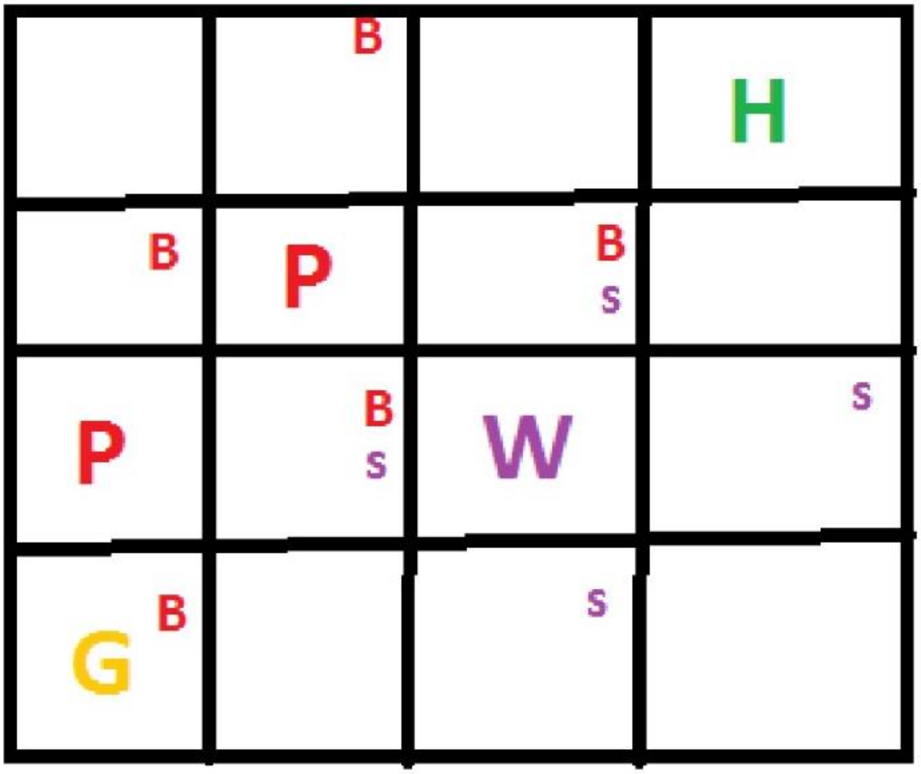
Number4:

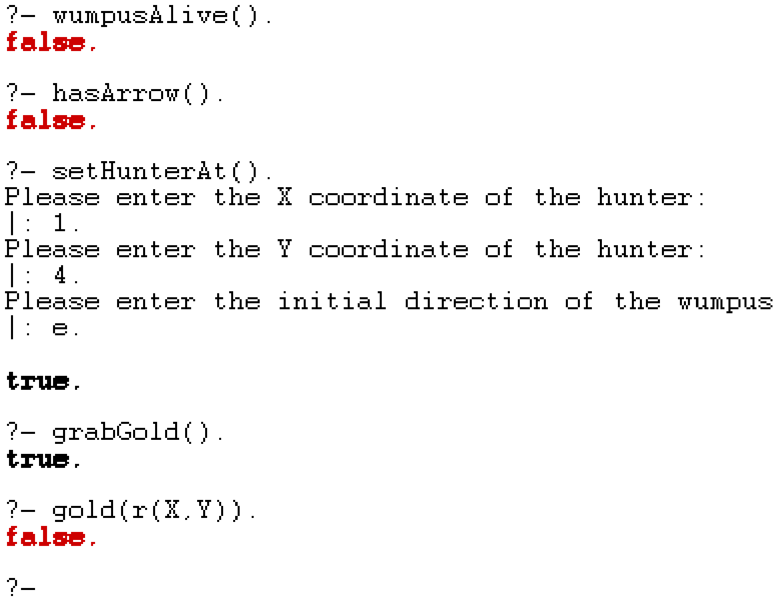


```
?- start.
Please enter the X coordinate of the hunter:
|: 4.
Please enter the Y coordinate of the hunter:
|: 4.
Please enter the initial direction of the wumpus
|: n.

true.

?- hunterAt(r(X,Y),Z).
X = Y, Y = 4,
Z = n.

?- wumpus(r(X,Y)).
X = 3,
Y = 2.

?- pit(r(X,Y)).
X = 1,
Y = 2 ;
X = 2,
Y = 3.

?- golf(r(X,Y)).
Correct to: "gold(r(X,Y))"?
Please answer 'y' or 'n'? yes
X = Y, Y = 1.

?- breeze(r(X,Y)).
X = Y, Y = 1 ;
X = 1,
Y = 3 ;
X = 1,
Y = 3 ;
X = Y, Y = 2 ;
X = Y, Y = 2 ;
X = 2,
Y = 4 ;
X = Y, Y = 3 ;
false.

?- stench(r(X,Y)).
X = Y, Y = 2 ;
X = 3,
Y = 1 ;
X = Y, Y = 3 ;
X = 4,
Y = 2 ;
false.

?- setHunterAt().
Please enter the X coordinate of the hunter:
|: 1.
Please enter the Y coordinate of the hunter:
|: 1.
Please enter the initial direction of the wumpus
|: e.

true.

?- grabGold().
true.

?- gold(r(X,Y)).
false.

?- setHunterAt().
Please enter the X coordinate of the hunter:
|: 2.
Please enter the Y coordinate of the hunter:
|: 2.
Please enter the initial direction of the wumpus
|: w.
```

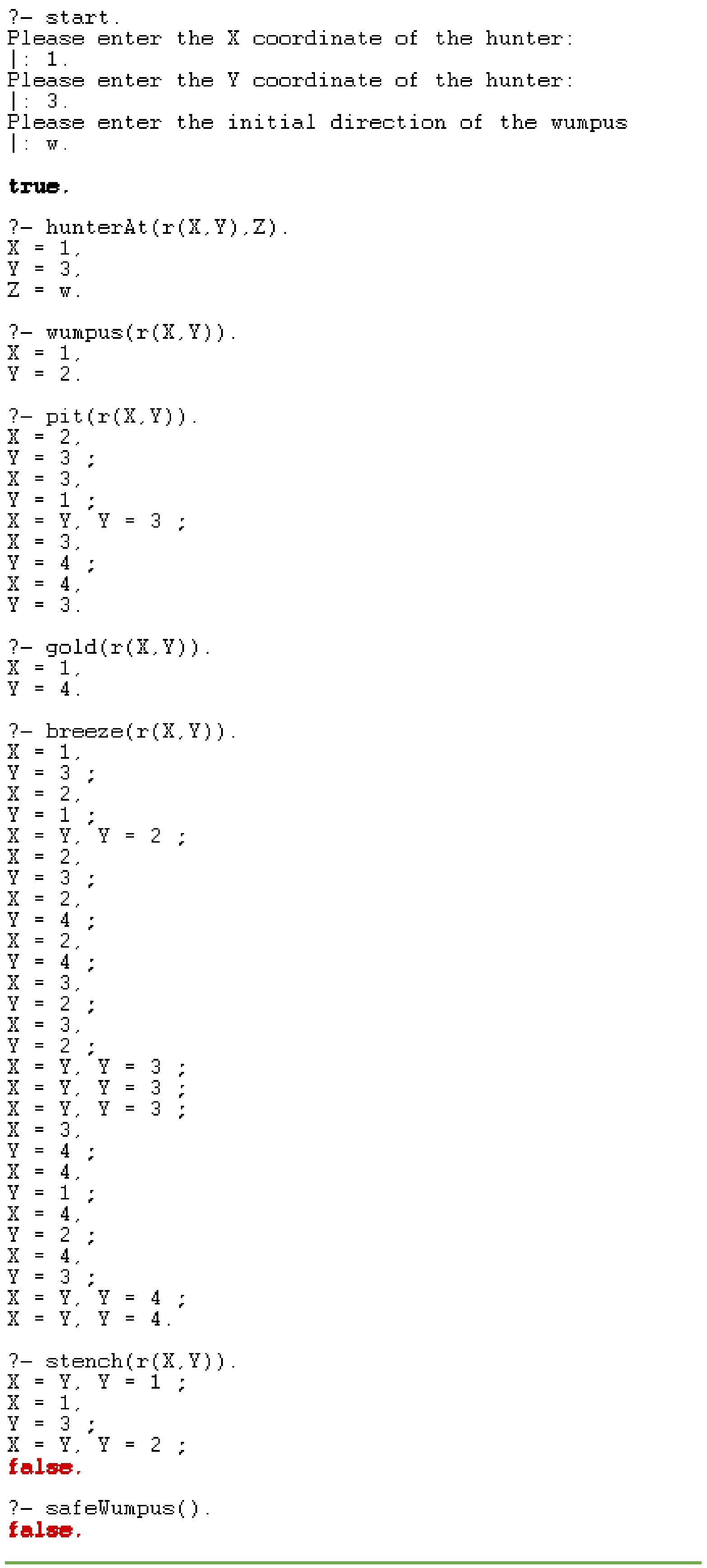


```
?- hunterAt(r(X,Y),Z).
X = Y, Y = 2,
Z = w.

?- turnRight().
true.

?- hunterAt(r(X,Y),Z).
X = Y, Y = 2,
Z = n.

?- turnRight().
true .

?- hunterAt(r(X,Y),Z).
X = Y, Y = 2,
Z = e.

?- shootWumpusWithDirection().
true .

?- wumpusAlive().
false.

?- hasArrow().
false.

?-
```





```
?- wumpusAlive().
false.

?- hasArrow().
false.

?- setHunterAt().
Please enter the X coordinate of the hunter:
|: 1.
Please enter the Y coordinate of the hunter:
|: 4.
Please enter the initial direction of the wumpus
|: e.

true.

?- grabGold().
true.

?- gold(r(X,Y)).
false.

?-
```



```
?- start.
Please enter the X coordinate of the hunter:
|: 1.
Please enter the Y coordinate of the hunter:
|: 3.
Please enter the initial direction of the wumpus
|: w.

true.

?- hunterAt(r(X,Y),Z).
X = 1,
Y = 3,
Z = w.

?- wumpus(r(X,Y)).
X = 1,
Y = 2.

?- pit(r(X,Y)).
X = 2,
Y = 3 ;
X = 3,
Y = 1 ;
X = Y, Y = 3 ;
X = 3,
Y = 4 ;
X = 4,
Y = 3.

?- gold(r(X,Y)).
X = 1,
Y = 4.

?- breeze(r(X,Y)).
X = 1,
Y = 3 ;
X = 2,
Y = 1 ;
X = Y, Y = 2 ;
X = 2,
Y = 3 ;
X = 2,
Y = 4 ;
X = 2,
Y = 4 ;
X = 3,
Y = 2 ;
X = 3,
Y = 2 ;
X = Y, Y = 3 ;
X = Y, Y = 3 ;
X = Y, Y = 3 ;
X = 3,
Y = 4 ;
X = 4,
Y = 1 ;
X = 4,
Y = 2 ;
X = 4,
Y = 3 ;
X = Y, Y = 4 ;
X = Y, Y = 4.

?- stench(r(X,Y)).
X = Y, Y = 1 ;
X = 1,
Y = 3 ;
X = Y, Y = 2 ;
false.

?- safeWumpus().
false.
```

Limitation of the code and performance rate:

Limitaion1:

We could have a stench and breeze in the same room. So avoid this problem we can add a no pit (\+ pit()) when we have a breeze.

Limitation2:

Assuming that the hunter has infinite arrows. Solution is once the hunter shoot we have to add the predicate to our knowledgebased.

Limitaion3:

Maybe the game player lost the game if he gets the wrong predicate. So to solve this problem we can use some heuristic functions that could help the user get an overall view about the right next step.