**WUMPUS World in Prolog**

Q1. **Why do you think Prolog is a good choice for logic?**

Ans: Prolog is a declarative programming language which is centred around a small set of basic mechanisms, and also including concepts like pattern matching, tree-based data structuring and automatic backtracking. Prolog is very useful for problems involving structured objects and relations between them. Therefore:

1. Prolog is easy for a user to use since the concepts used are very basic
2. The user has many inbuilt concepts like pattern matching, tree-based data structuring and automatic backtracking which gives a lot of flexibility to implement the logic and thoughts he/she is intending to express
3. Since Prolog can express data in terms of relation, we can think of the same attribute being used redundantly in the form of functions (which can represent relation). Hence the script becomes very concise and organized and easy to understand and code as well.

Overall, to conclude: Prolog is a great language to express varipus relations, goals in a succinct, concise form , easily understandable by the users.

Q2. **The default testing map for the agent is depicted in Russell and Norvig’s Figure 7.2. The code snippet for initializing this map can be found wumpus\_world.pl line 169-181. Check lines 173-179 and explain what each one is doing.**

Ans: Lines 173-179 are highlighted below:

**initialize\_world(fig72) :-**

ww\_retractall,

retractall(ww\_initial\_state(\_)),

assert(ww\_initial\_state([])),

addto\_ww\_init\_state(wumpus\_world\_extent(4)),

addto\_ww\_init\_state(wumpus\_location(1,3)),

addto\_ww\_init\_state(wumpus\_health(alive)),

addto\_ww\_init\_state(gold(2,3)),

addto\_ww\_init\_state(pit(3,1)),

addto\_ww\_init\_state(pit(3,3)),

addto\_ww\_init\_state(pit(4,4)),

ww\_initial\_state(L),

assert\_list(L).

Before understanding the above mentioned lines, I would like to state the following fact from Prolog Documentation:

Recursion is the primary control mechanism for Prolog programming, and the list structure is the primary data structure used for representing complex data. A way to represent a list is to use the head/tail notation [H|T]. Here the head of the list, H, is separated from the tail of the list, T, by a vertical bar. The tail of a list is the original list with its first element removed. The tail of a list is always a list, even if it's the empty list.

Explanation of the highlighted Lines: Basically we are trying to create a database of facts in terms of lists since list is the primary data structure Prolog uses to store complex data.

Addto\_ww\_init\_state is defined as followed. Basically it is trying to add a new Fact as the Head of the total list L:

% addto\_ww\_init\_state(Fact): Adds Fact to the list L stored in

% ww\_initial\_state(L).

addto\_ww\_init\_state(Fact) :-

retract(ww\_initial\_state(L)),

assert(ww\_initial\_state([Fact|L])).

After understanding the basic work of the function addto\_ww\_init\_state, here are the explanations for lines 173-179:

1. addto\_ww\_init\_state(wumpus\_world\_extent(4)): Adding the first Fact that the Wumpus world considered in the problem is a 4x4 wumpus world.
2. addto\_ww\_init\_state(wumpus\_location(1,3)): Defines the fact that the position of the Wumpus in the 2D grid is at (1,3).
3. addto\_ww\_init\_state(wumpus\_health(alive)): Defines the fact that the the health status of Wumpus is alive
4. addto\_ww\_init\_state(gold(2,3)): Defines the fact that the gold is placed at the location (1,3) in the 2D grid
5. addto\_ww\_init\_state(pit(3,1)): Defines the fact that the pit is placed in the location (3,1) in the 2D grid
6. addto\_ww\_init\_state(pit(3,3)): Defines the fact that the pit is placed in the location (3,1) in the 2D grid
7. addto\_ww\_init\_state(pit(4,4)): Defines the fact that the pit is placed in the location (3,1) in the 2D grid

Basically, the above statements created the entire Wumpus environment with the placements of various objects in the 2D grid

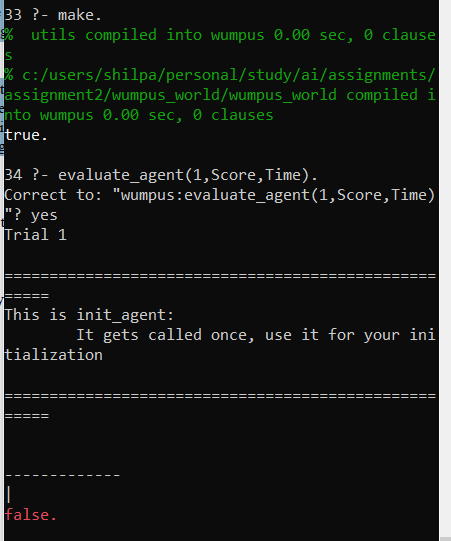
Q3. **Check test\_ww.pl. In this file, you are given several different maps. Not all the setups require the same level of sophistication from an agent to arrive at the best decision. You start from a naive agent and try to complete the set of logical rules in pursuit of better policies. Look at the 7 maps. Each one is designed to examine a certain aspect of your agent’s inference abilities. For each one, explain:**

**A. Whether the agent can grab the gold or not?**

**B. The ultimate goal is to maximize the expected score. This means that in some cases, it is better to climb out of the cave without retrieving the gold. For each map, describe the inference chain that an intelligent agent would follow to achieve this.**

Ans:

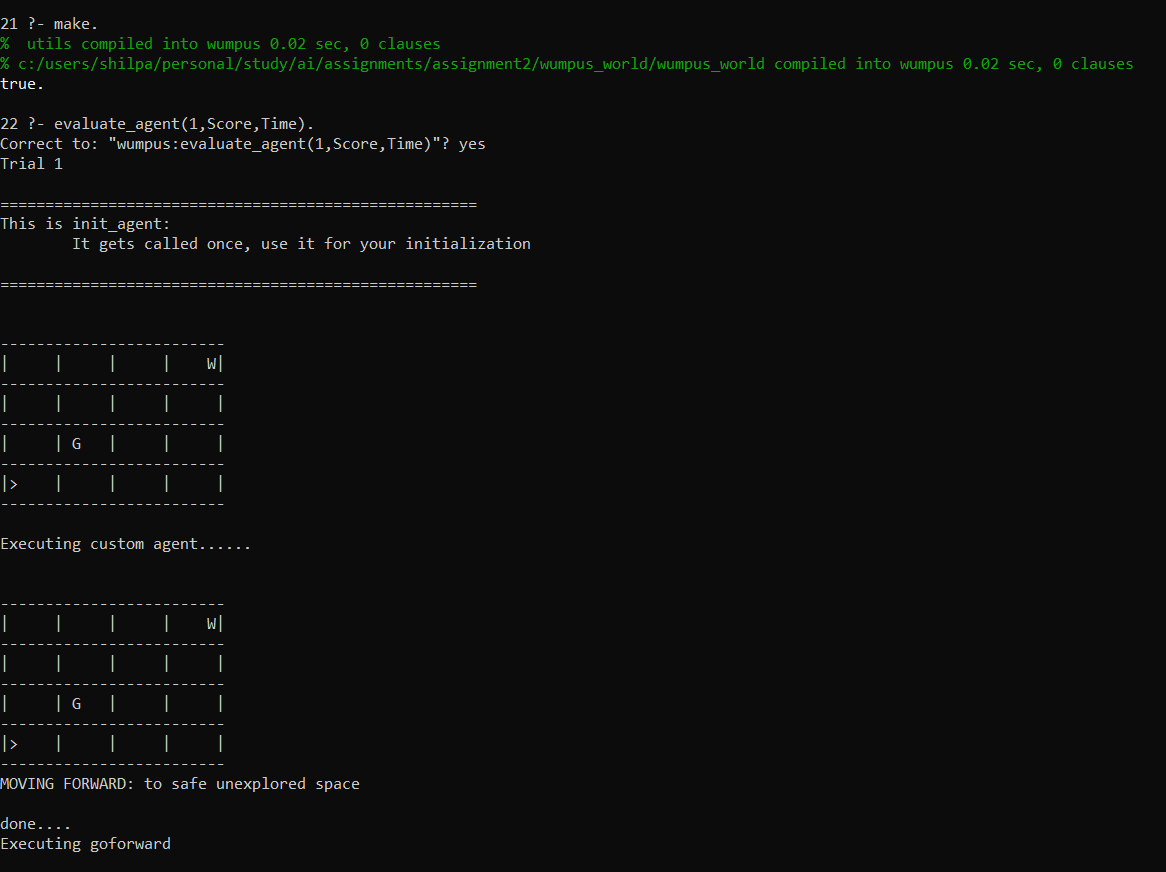
1. Map\_1

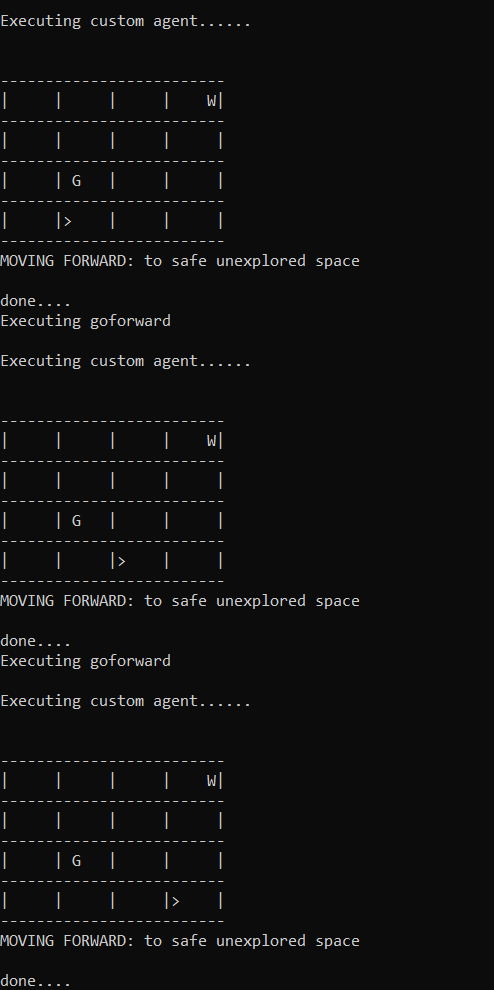
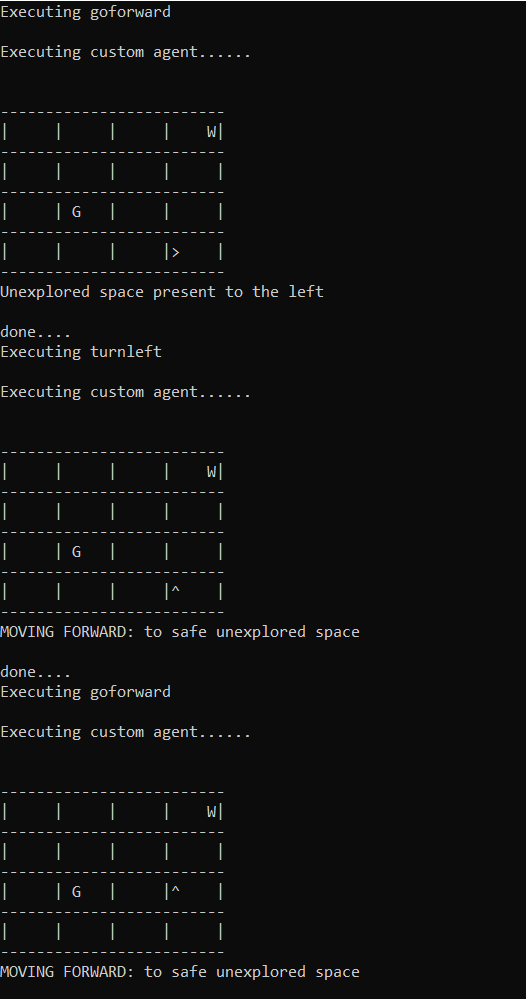


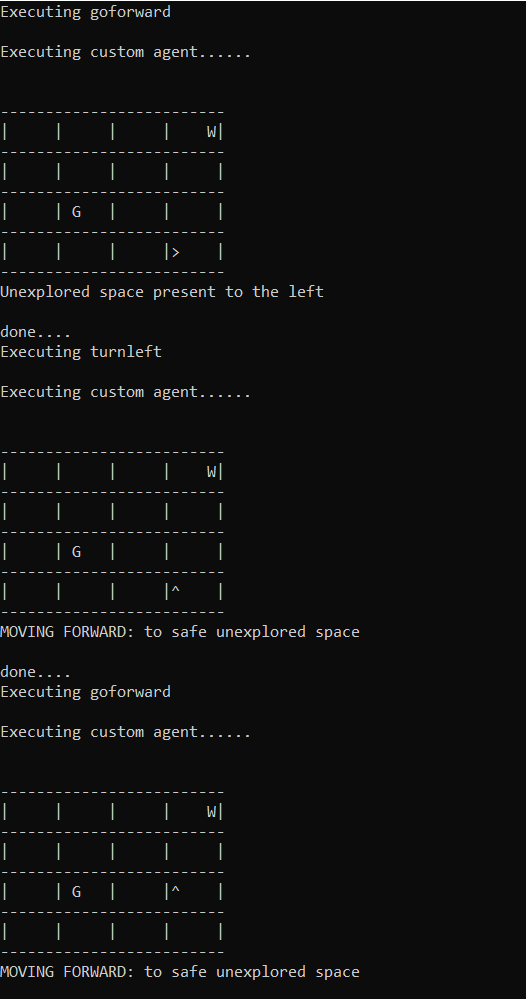
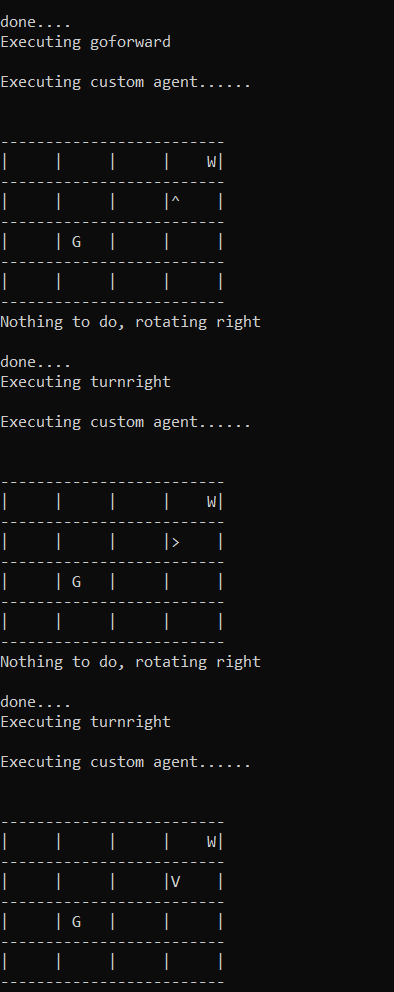
1. Map\_2

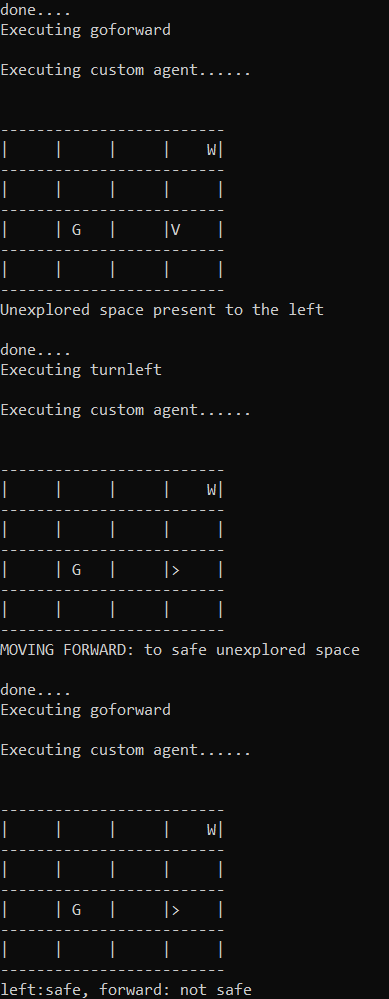
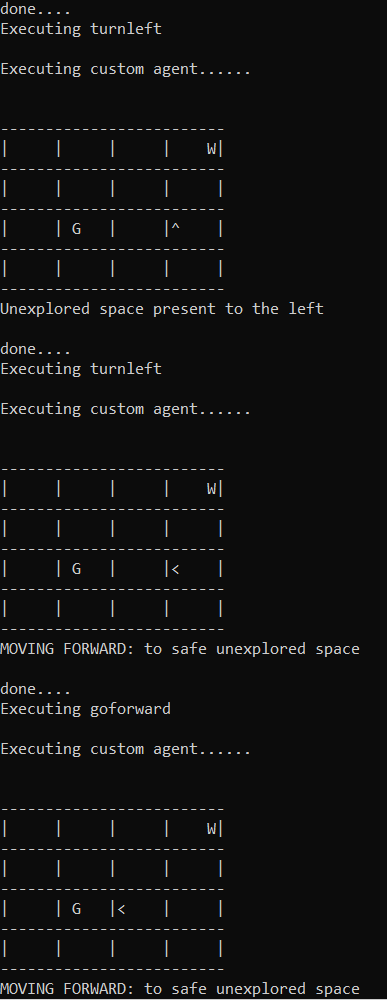
* The Gold can be achieved as shown in the 8th screenshot. The Wumpus has also been killed in the 13th screenshot.
* Inference chain:

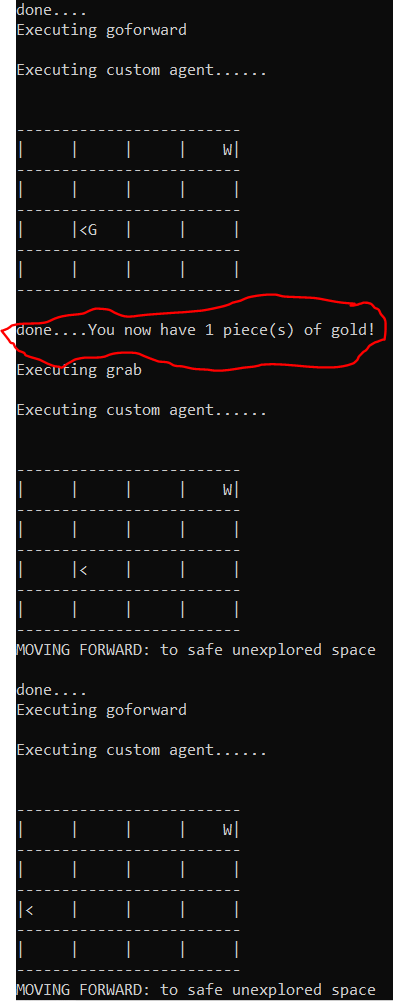
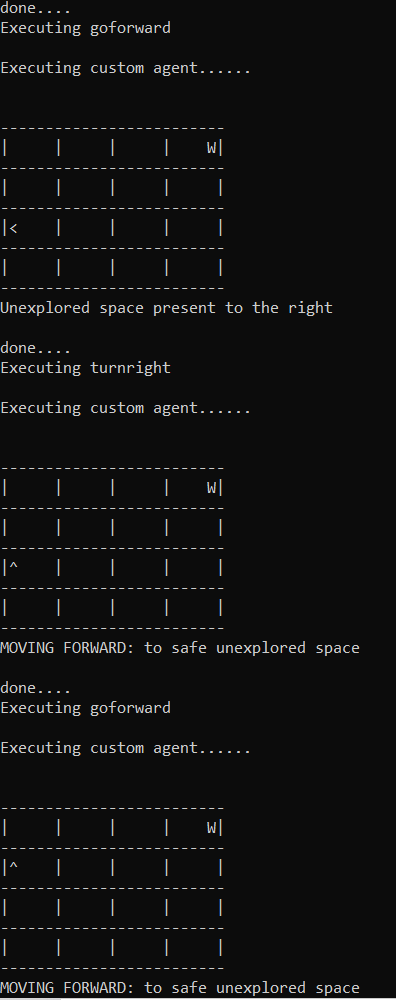
Go forward to 2,1-> go forward to 3,1 -> go forward to 4,1 -> encountered wall, left is safe, hence turned left -> go forward to 4,2 -> go forward to 4,3 -> rotate right since received stench-> rotate right -> go forward to 4,2 -> turn right-> left safe, forward not safe, hence turn left -> turn left to unexplored space-> go forward to 3,2 -> go forward to 2,2-> GRAB GOLD -> go forward to 1,2 -> turn right to safe space -> go forward to 1,2 -> go forward to 1,3 -> go forward to 1,4 -> unexplored space to left, turn left-> wall encountered, hence not safe, turn left-> unexplored space on left, turn left-> go forward to 2,4-> go forward to 3,4 -> shoot arrow(since adjacent cells have stenches), Wumpus killed -> go forward to 4,4-> un explored space to left, but not safe since wall encountered, hence turn left twice-> go forward to 3,4-> turn left to unexplored space-> go forward to 3,3-> turn left to unexplored space-> go forward to 2,3->go forward to 1,3-> forward not safe since wall encountered-> turn left-> go forward to 1,2-> go forward to 1,1-> nothing to explore more, hence Climb out.

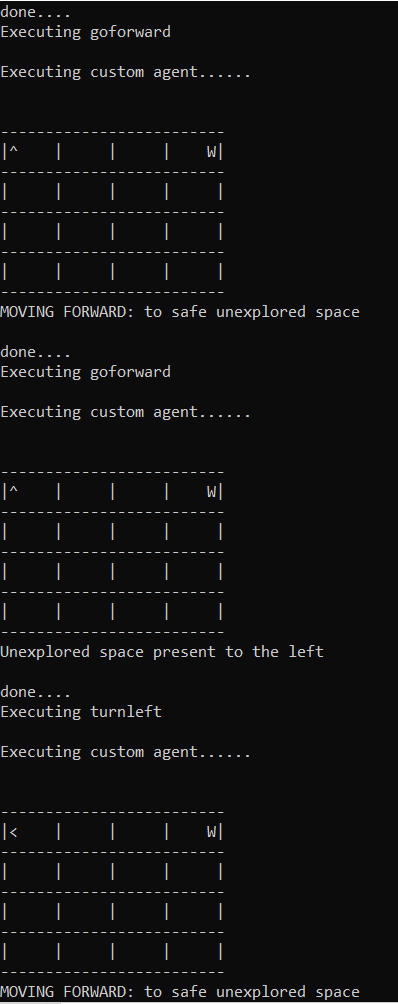
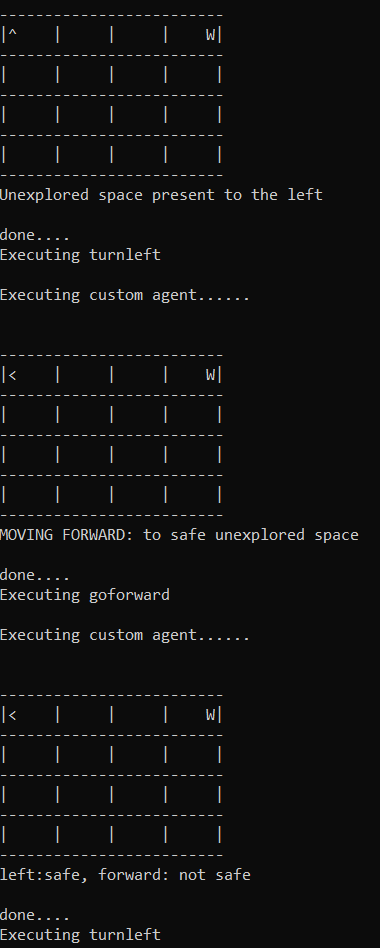


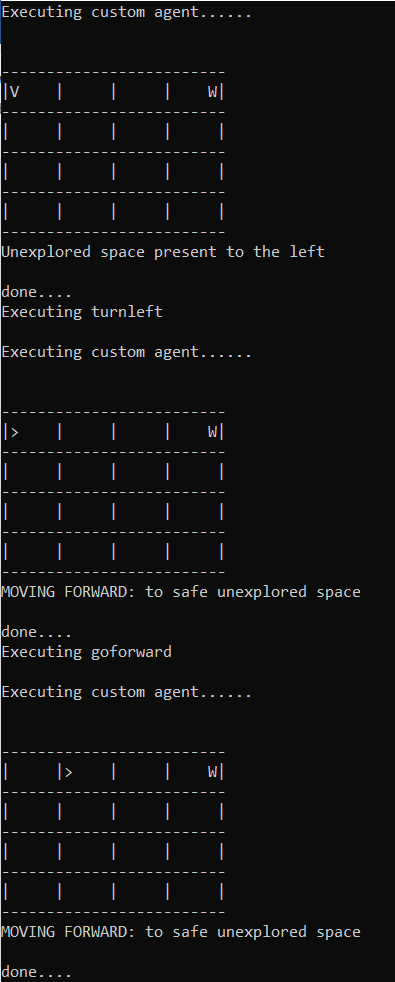
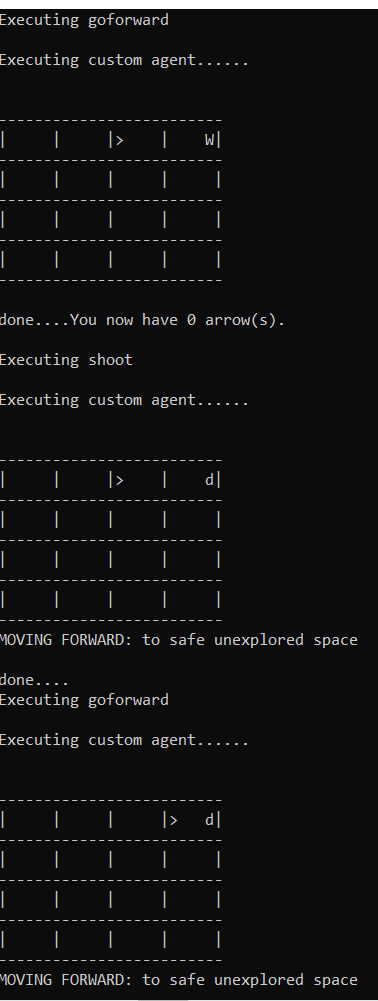
 

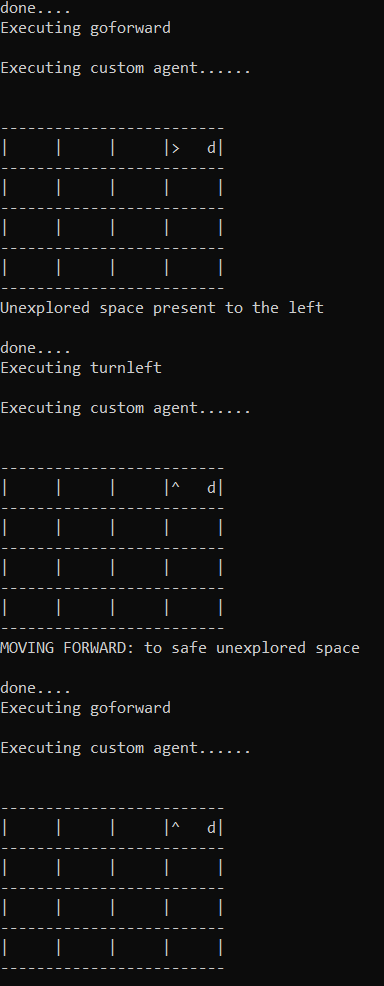
 

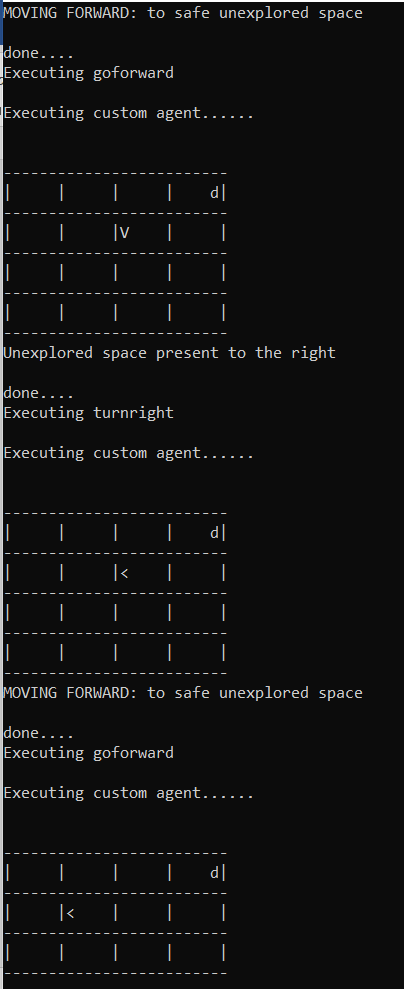
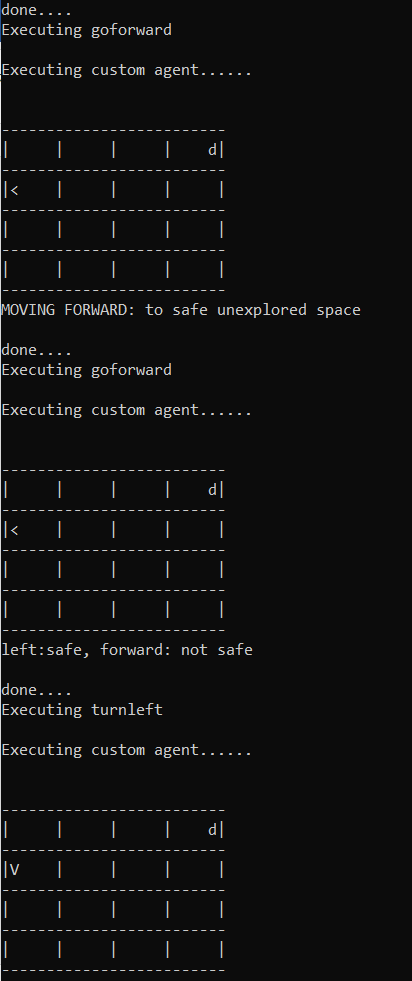
 

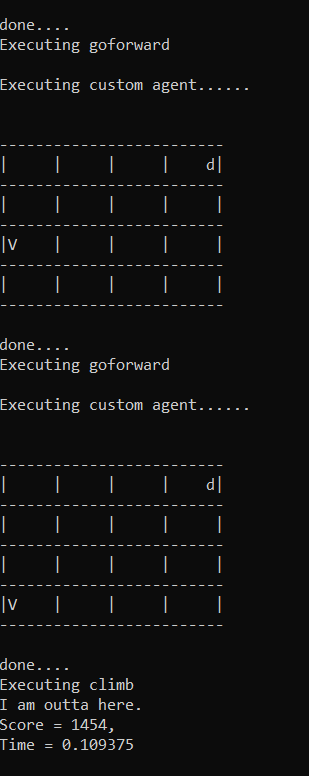
 

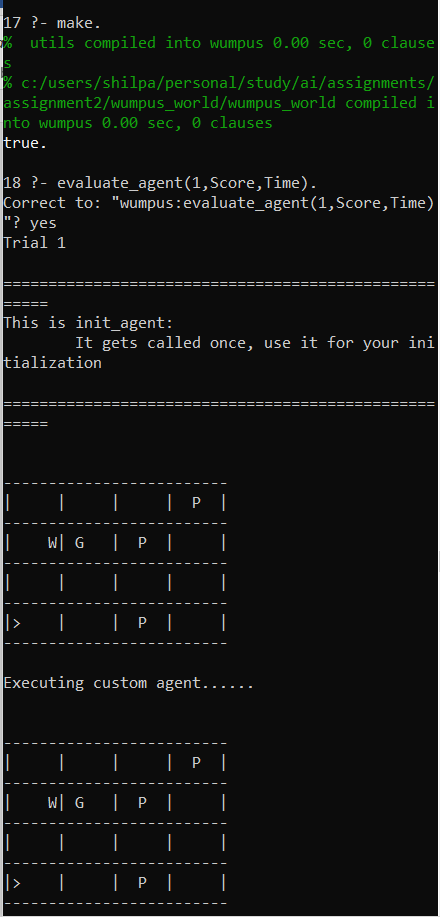
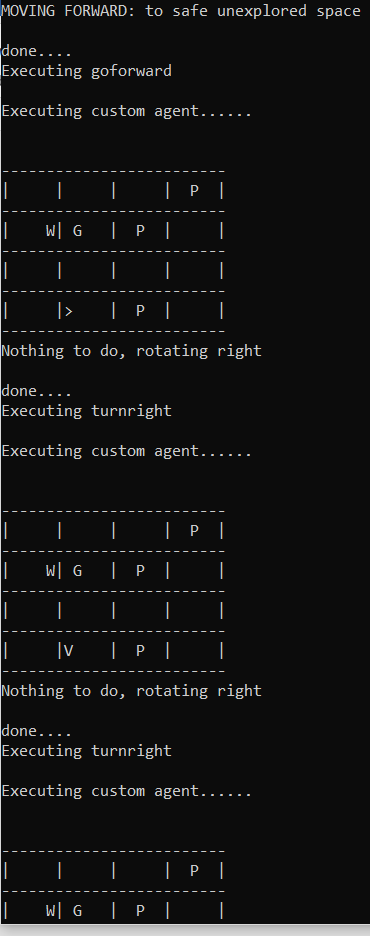


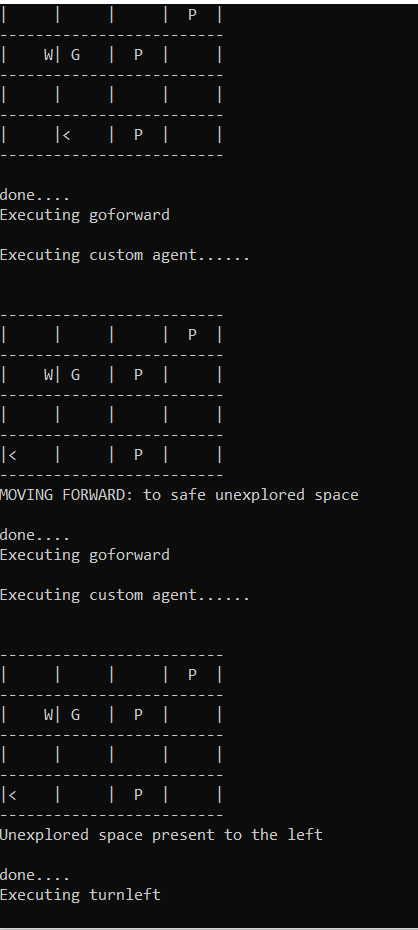
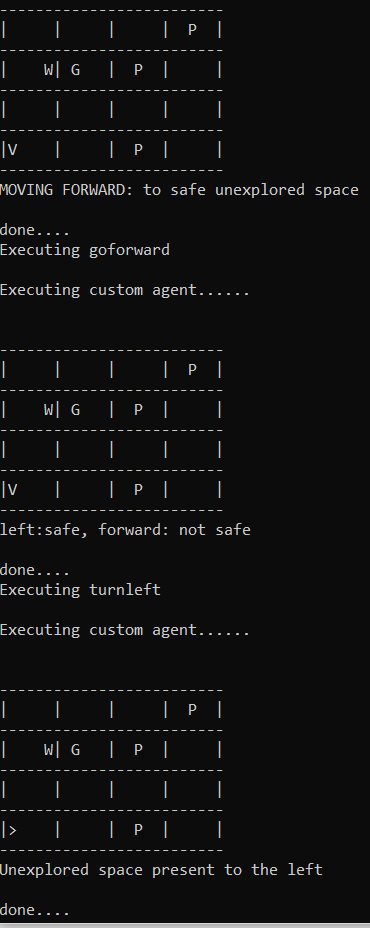
1. Map\_3

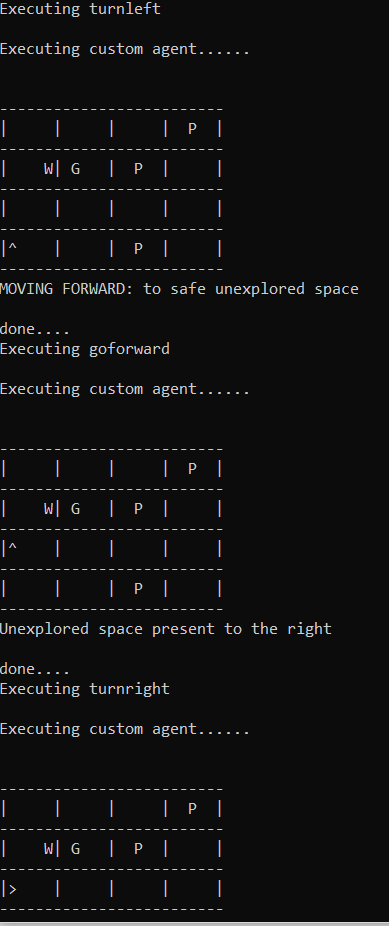
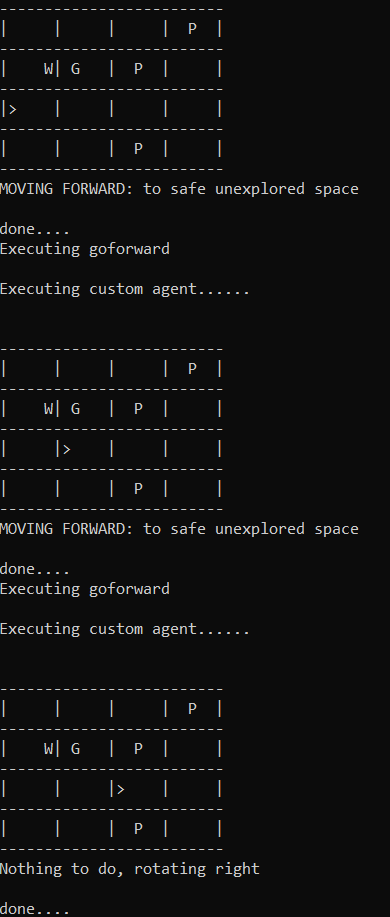
-The Gold can be achieved. The Wumpus has also been killed.

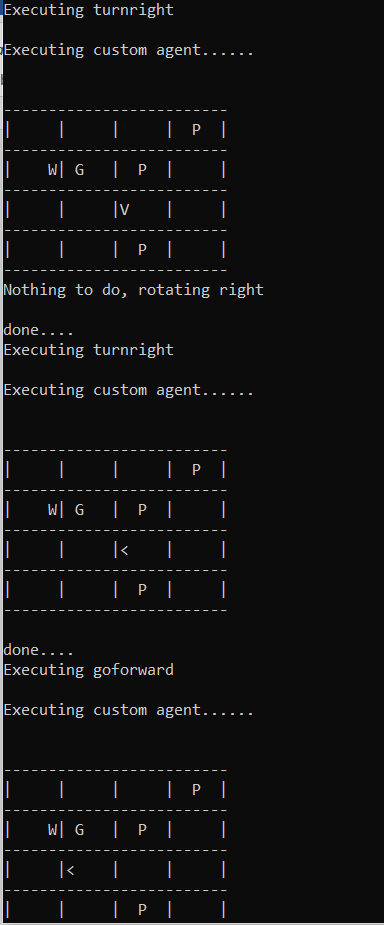
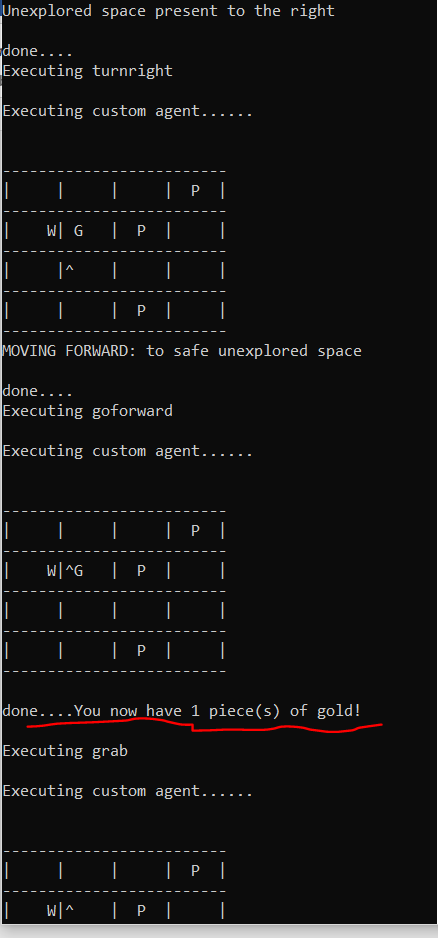
-Inference chain:

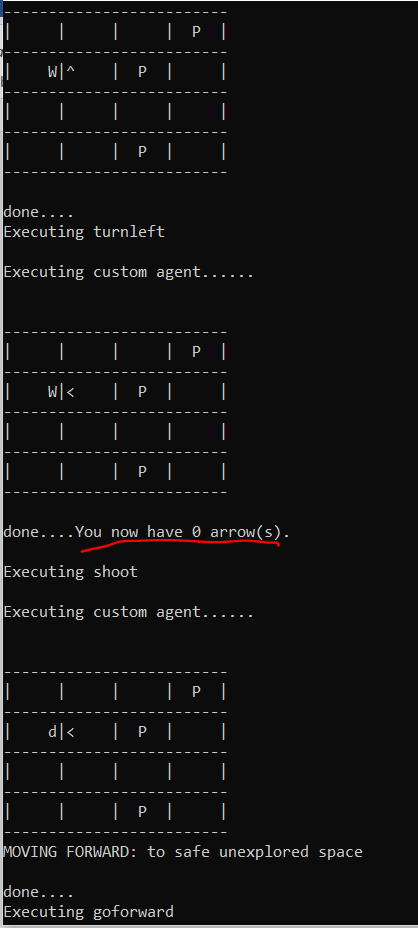
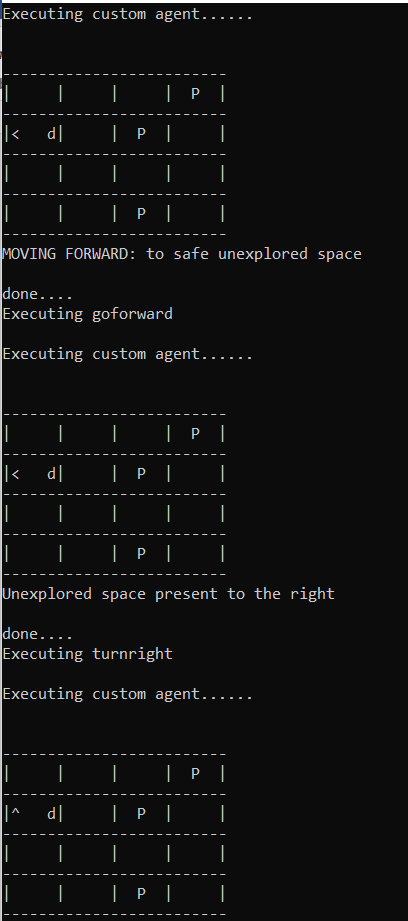
Go forward to 2,1-> experienced stench, turn right since nothing else to do-> turn right-> go forward to 1,1 again-> turn left to unexplored space -> experienced wall, turn left-> turn left to unexplored space-> turn left-> go forward to 1,2-> turn right and shoot Wumpus since (we encountered to adjacent stenches, hence Wumpus confirmed)-> go forward to 2,2-> go forward 3,2 -> go forward to 4,2-> not safe moving forward since wall encountered. Hence turn left since left is safe-> go forward to 4,3-> go forward to 4,4 -> turn left since wall encountered-> go forward to 3,4-> go forward to 2,4-> go forward to 1,4 -> encountered wall, hence turn to left -> go forward to 1,3-> turn left to unexplored space-> go forward to 2,3-> go forward to 3,3-> go forward to 4,3 -> turn left-> go forward to 4,4-> turn right since right is unexplored-> turn right-> go forward to 4,3-> go forward to 4,2-> go forward to 4,1-> wall encountered-> turn left-> forward not safe since wall encountered-> turn left-> turn left to unexplored space again -> go forward to 3,1-> GRAB GOLD -> turn left-> turn left->go forward to 4,1-> turn left-> go forward to 4,2-> go forward to 4,3-> go forward to 4,4-< turn left-> go forward 3,4-> turn right to unexplored space-> wall encountered, left is safe, hence turn left-> go forward to 2,4-> turn right to unexplored space-> wall encountered, hence turn left->go forward to 1,4-> turn right to unexplored space-> wall encountered, turn right-> go forward to 2,4-> go forward to 3,4-> go forward to 4,4-> turn right since nothing else to do-> go forward to 4,3->go forward to 4,2-> go forward to 4,1-> turn right since nothing else to do-> go forward to 3,1-> go forward to 2,1-> turn left to unexplored space-> wall encountered, not safe, turn left since left is safe-> go forward to 3,1-> go forward to 4,1-> turn left to safe space-> go forward to 4,2-> go forward to 4,3-> go forward to 4,4-> turn left to safe space-> go forward to 3,4-> go forward to 2,4-> go forward to 1,4-> turn left to safe space-> go forward to 1,3-> turn right to unexplored space-> not safe, wall encountered, turn left-> go forward to 1,2-> turn right to unexplored space-> Wall encountered, not safe, hence turn left-> go forward to 1,1-> climb out.

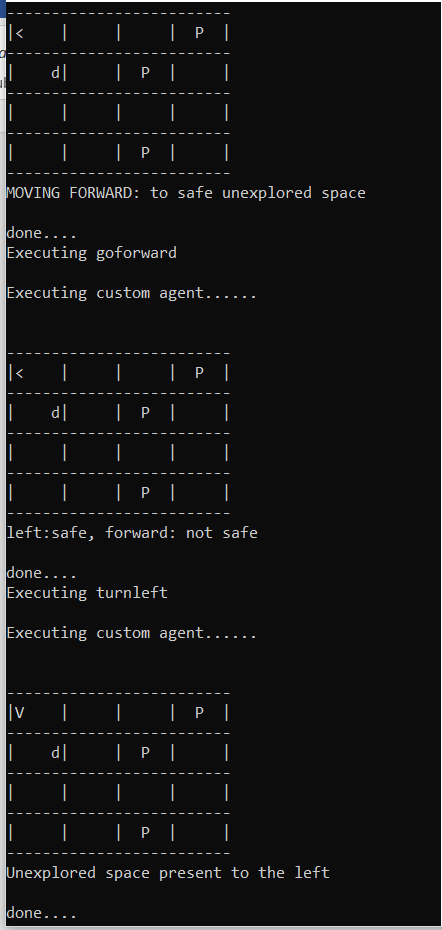
 

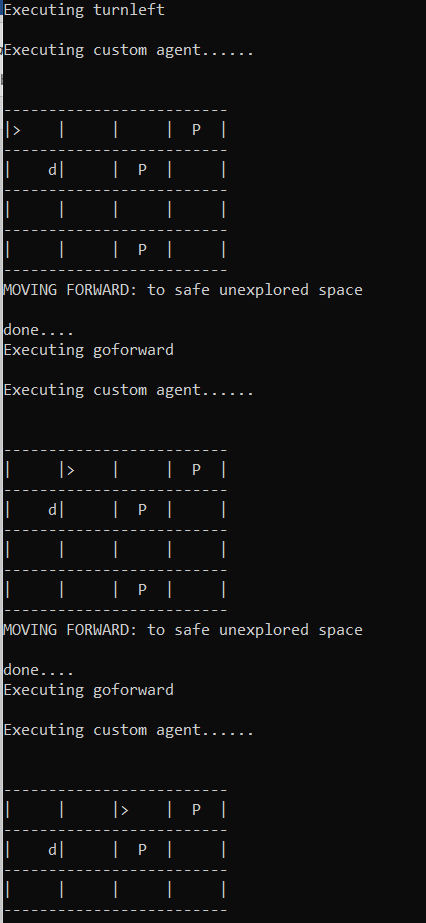
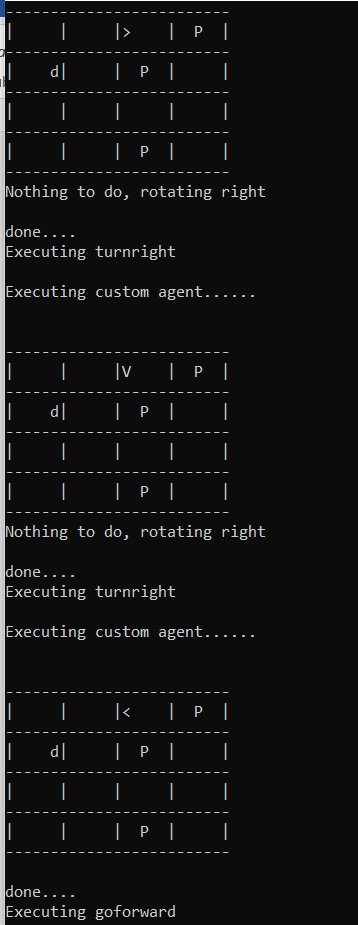
 

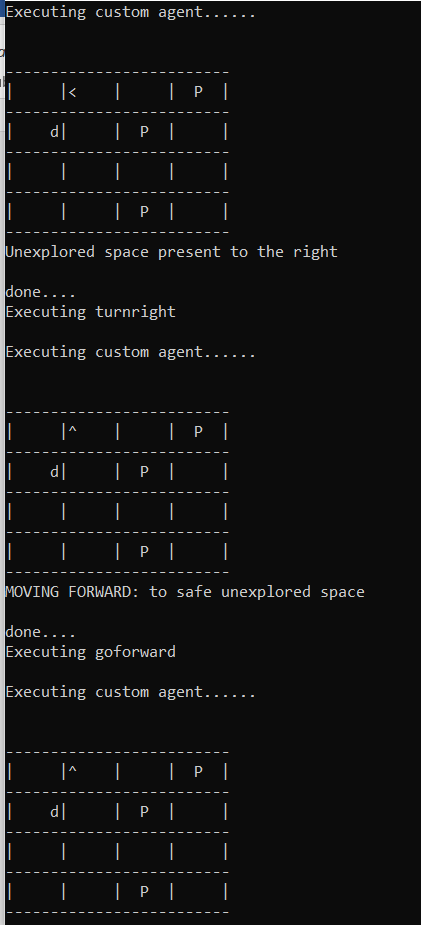
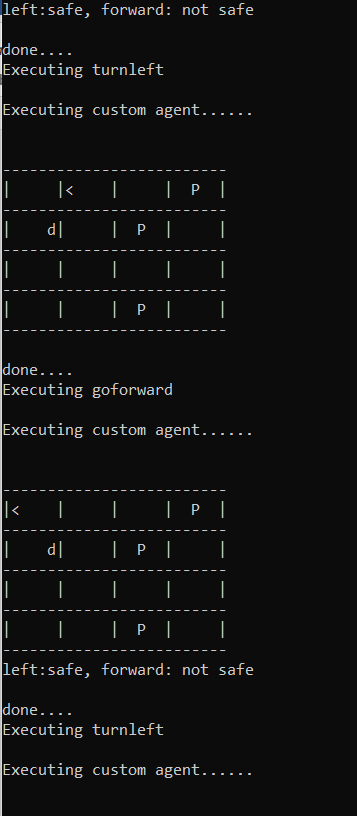
 

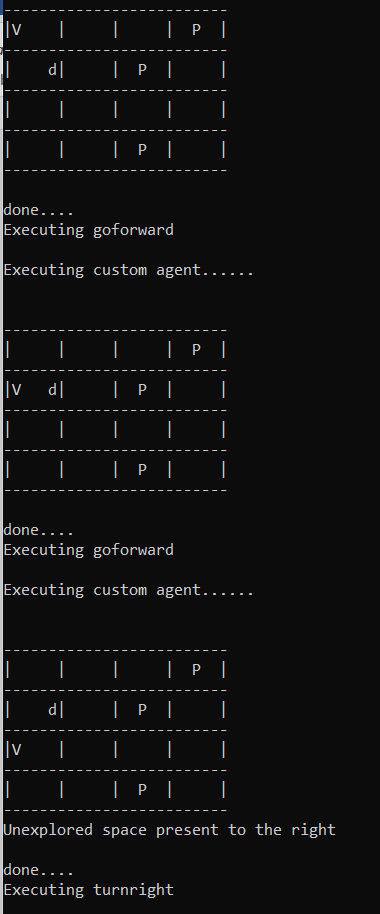
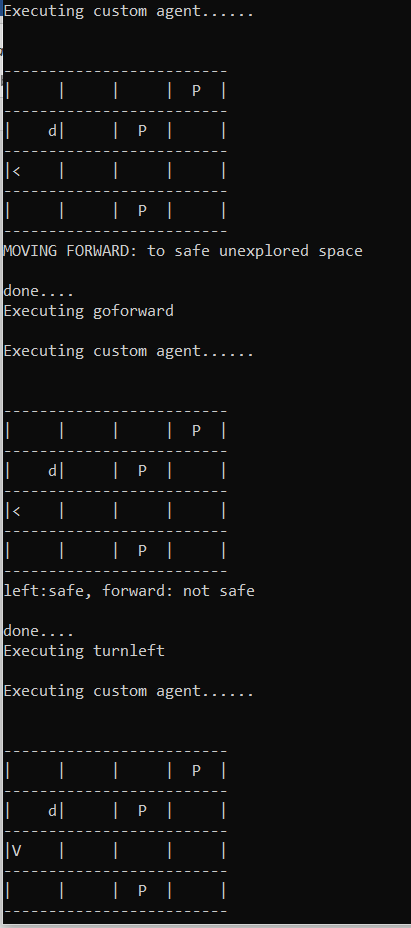
 

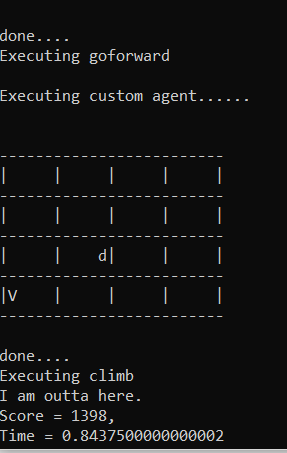
 

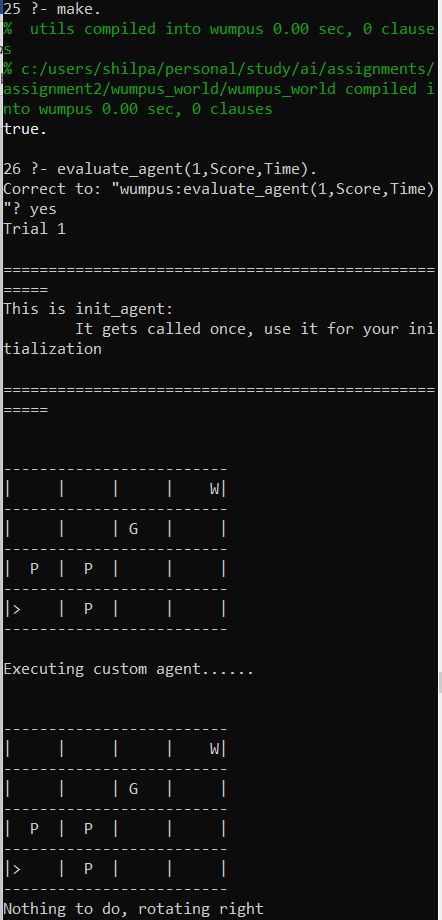
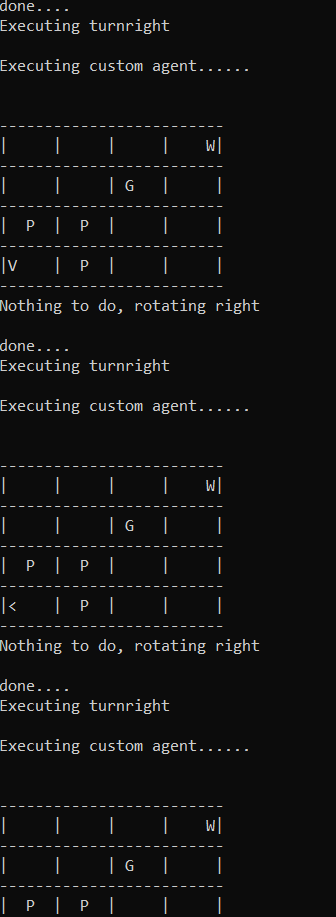
 

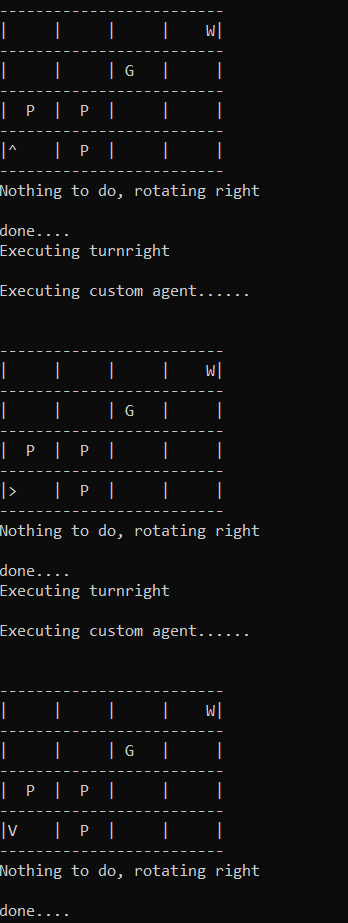
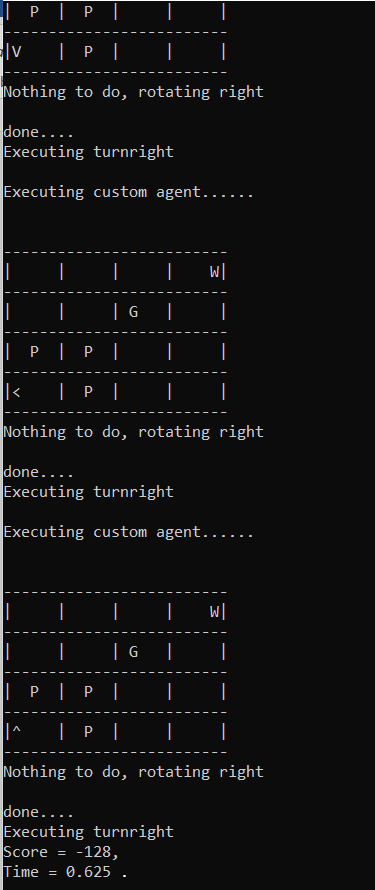
 



1. Map\_4

* The agent does not grab the gold or kill the Wumpus.
* The reason is that the agent is stuck between 3 pits and 1 wall. Hence it has nothing else to do than turning right. Therefore, the total score keeps going down and its negative.

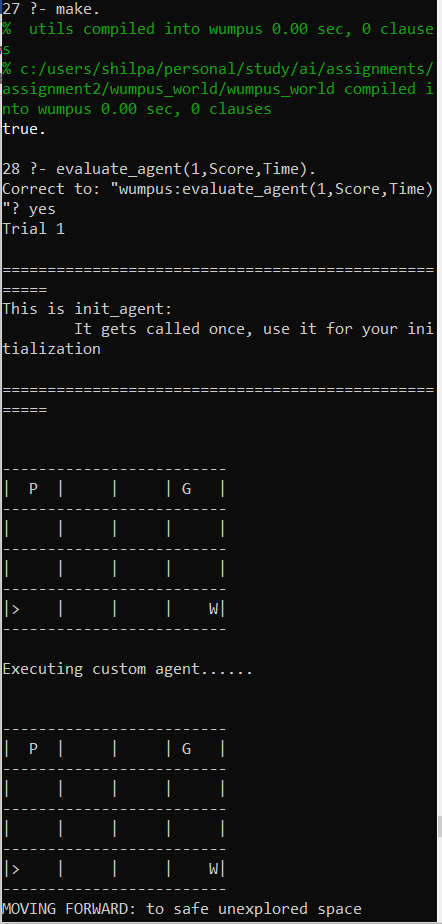
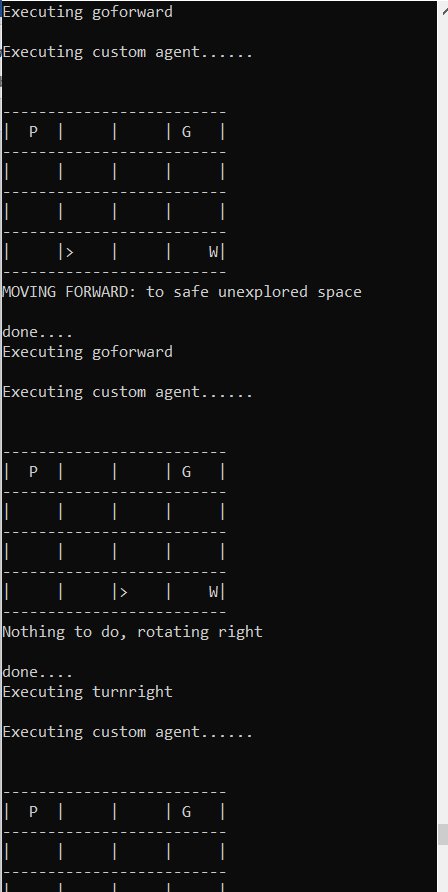
 

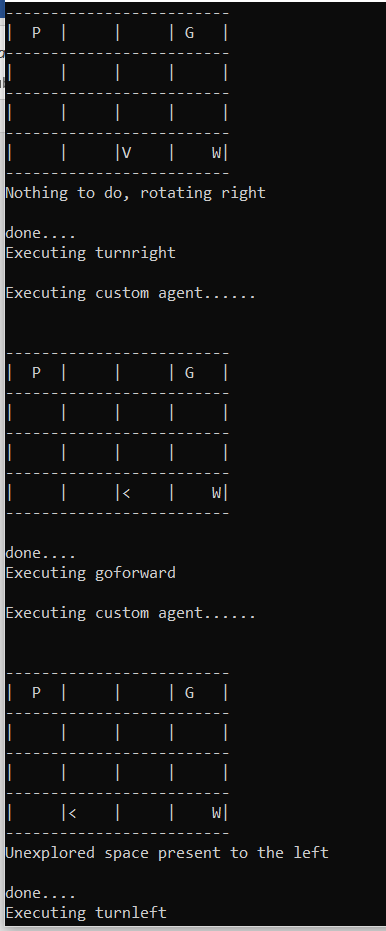
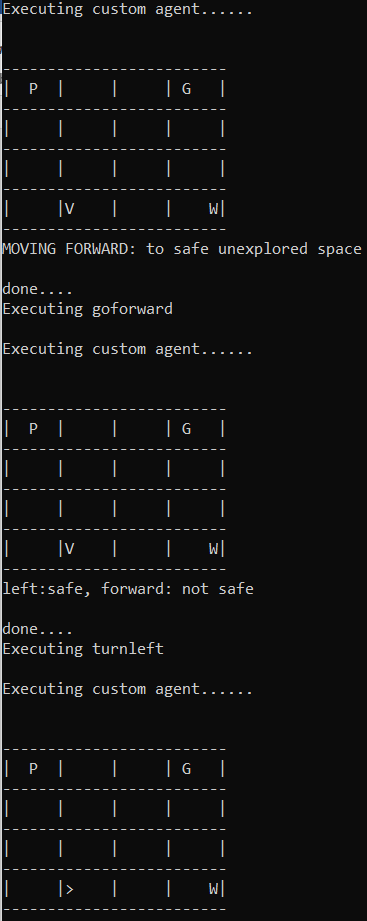
 

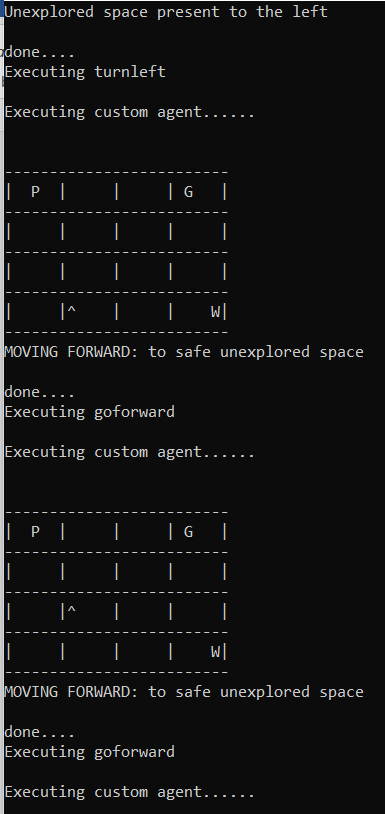
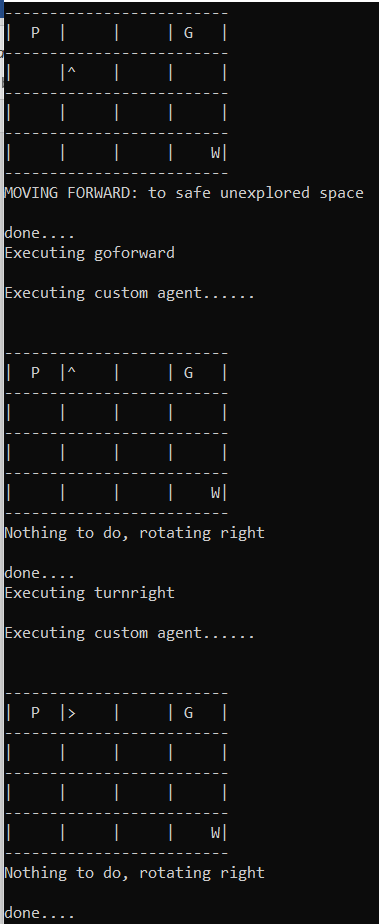
1. Map\_5

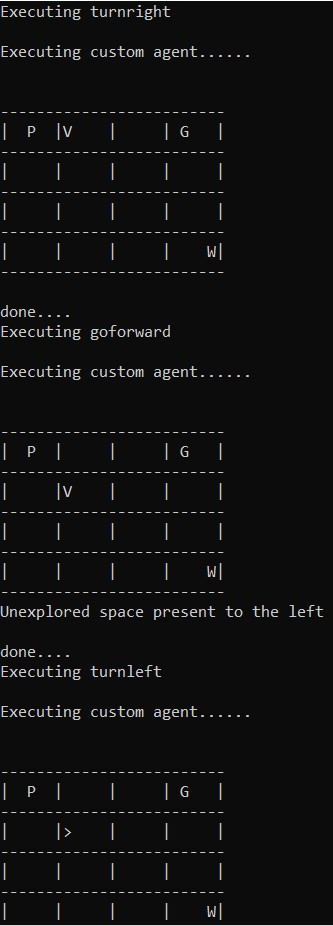
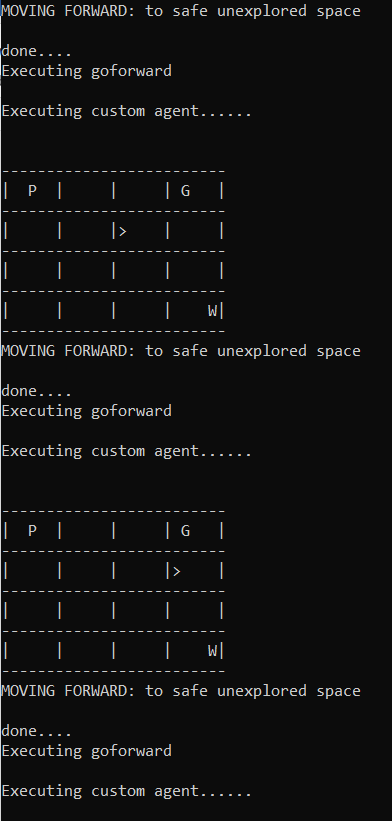
* The agent does not kill the Wumpus, but grabs the gold and then clims out.
* Inference Chain:

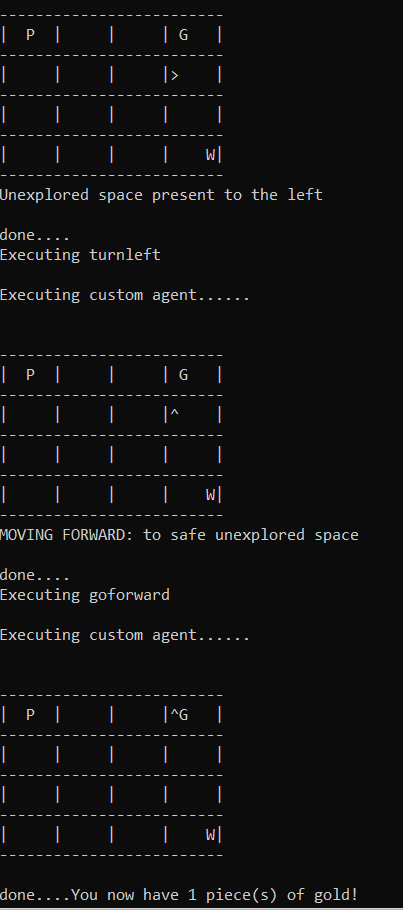
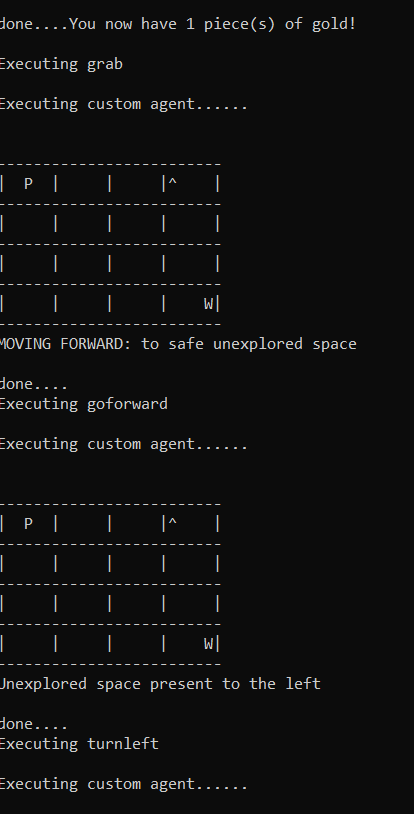
Go forward to safe unexplored space(2,1)-> go forward to safe unexplored space(3,1)-> observed stench, nothing to do, turn right-> not safe since wall encountered, turn right-> go forward to 2,1-> turn left to unexplored space-> wall encountered, turn left-> turn left to unexplored space-> go forward to 2,2-> go forward to 2,3-> go forward to 2,4->wall encountered, nothing else to do-> turn right-> turn right again to go opposite since breeze encountered-> go forward to 2,3-> turn left to unexplored space-> go forward to 3,3-> go forward to 4,3-> wall encountered, forward not safe, turn left to unexplored space-> glitter encountered, hence move forward to 4,4-> GRAB GOLD->forward is not safe since wall encountered, hence turn left-> go forward to 3,4-> turn right to unexplored space-> not safe since wall encountered, hence turn left-> move forward to 2,4-> turn right to unexplored space-> wall encountered, not safe, hence turn back left-> since there is a breeze and (3,4) is already safe, hence deduced pit is in 1,4, therefore turn left-> go forward to 2,3->turn right to unexplored space-> go forward to 1,3-> since wall encountered, turn left to unexplored space-> go forward to 1,2-> turn right to unexplored space-> wall encountered, not safe, hence turn left-> go forward to 1,1-> climb out.

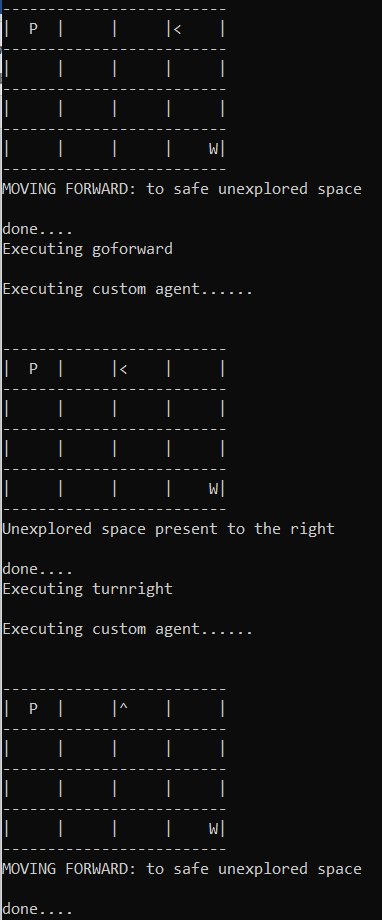
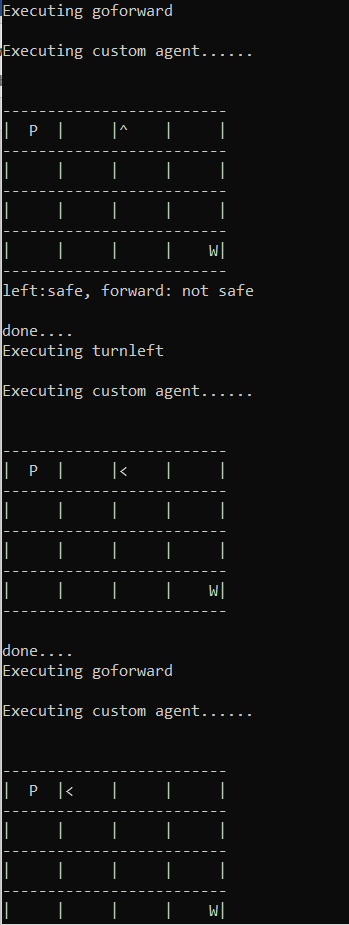
 

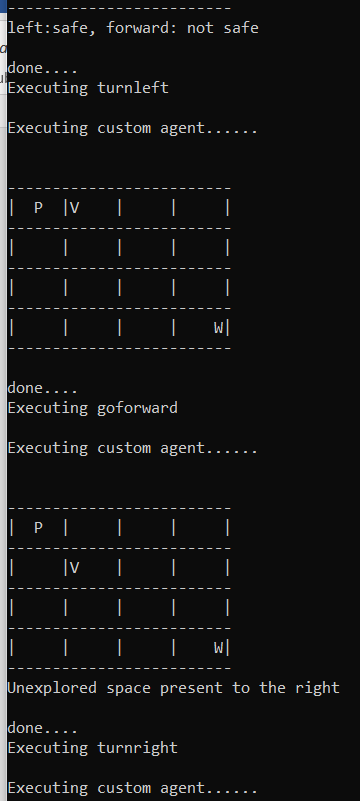
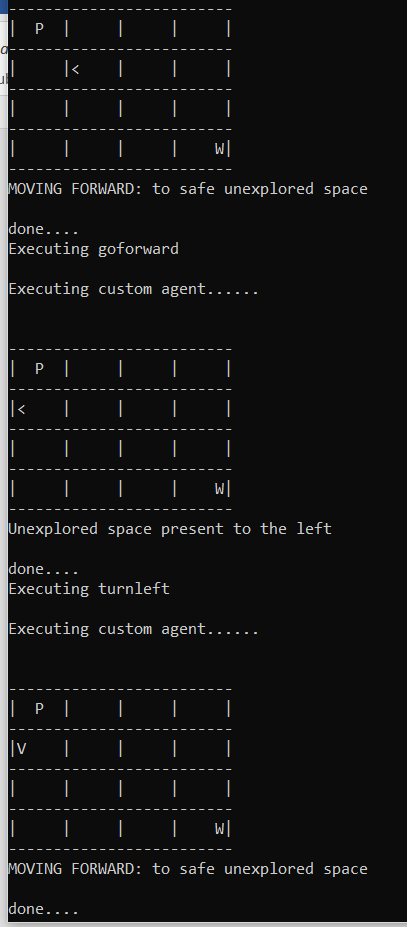
 

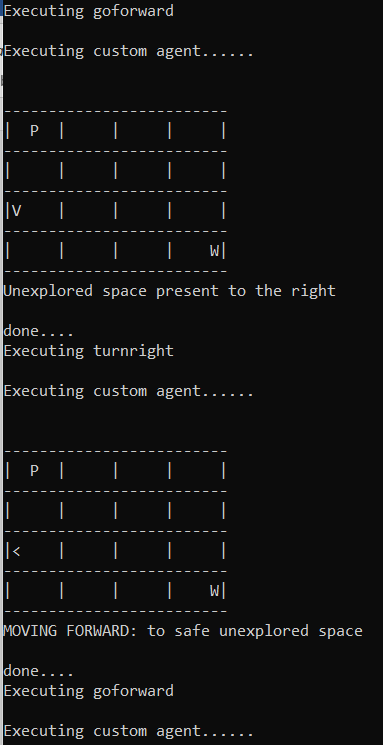
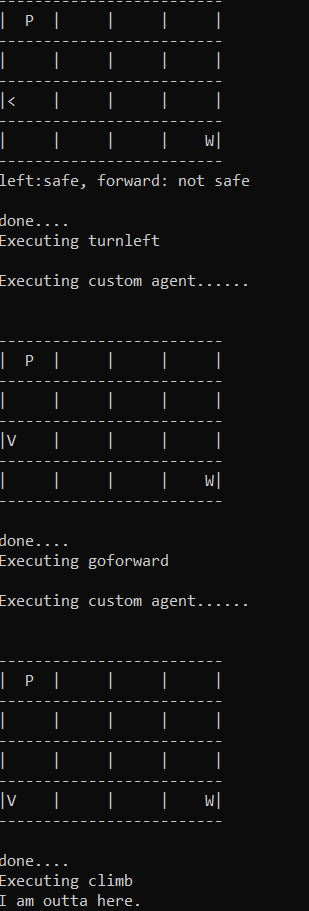
 

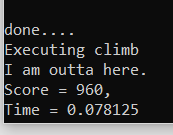
 

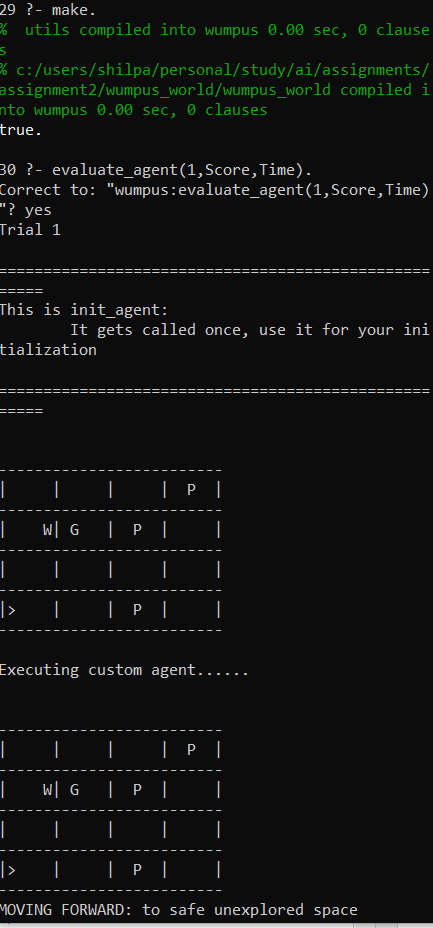
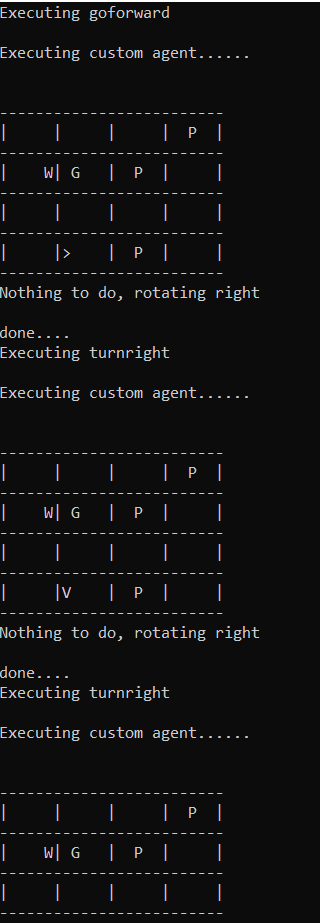
 

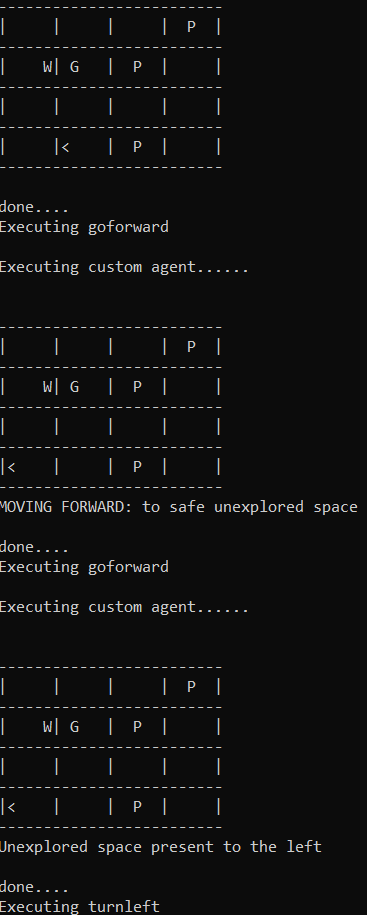
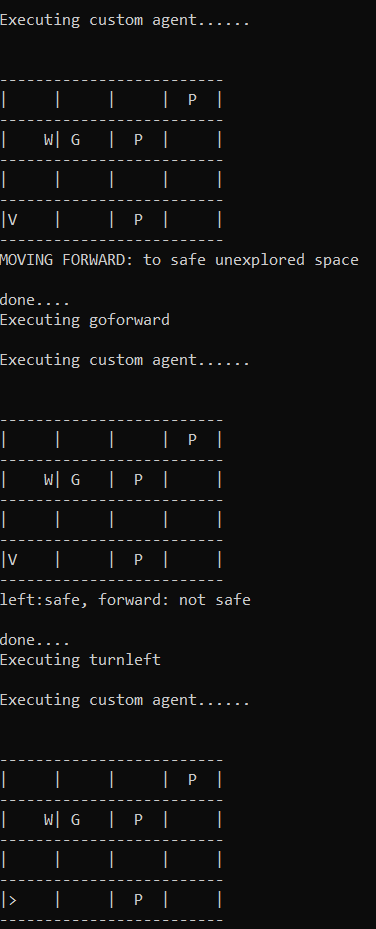


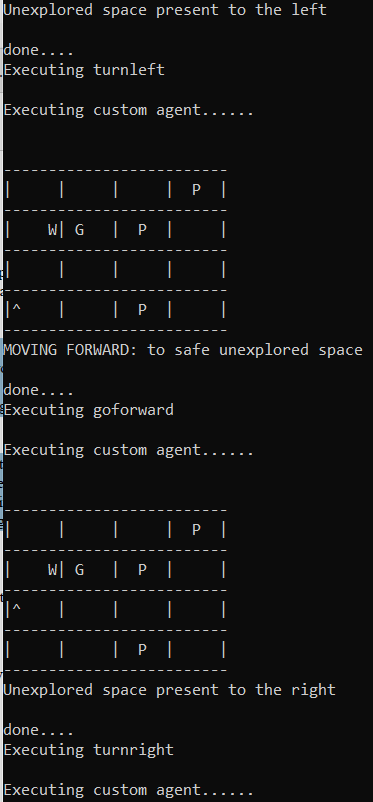
1. Map\_6

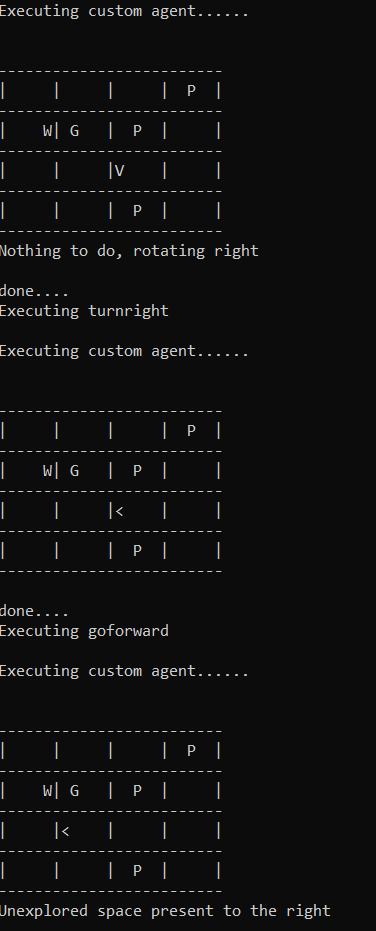
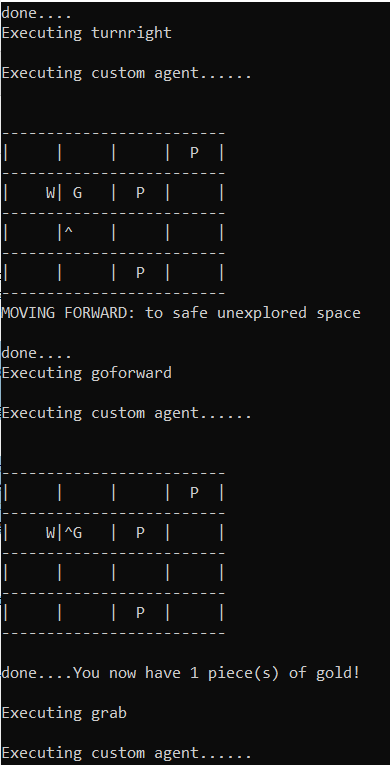
* The agent Grabs the gold and kills the wumpus
* Inference Chain:

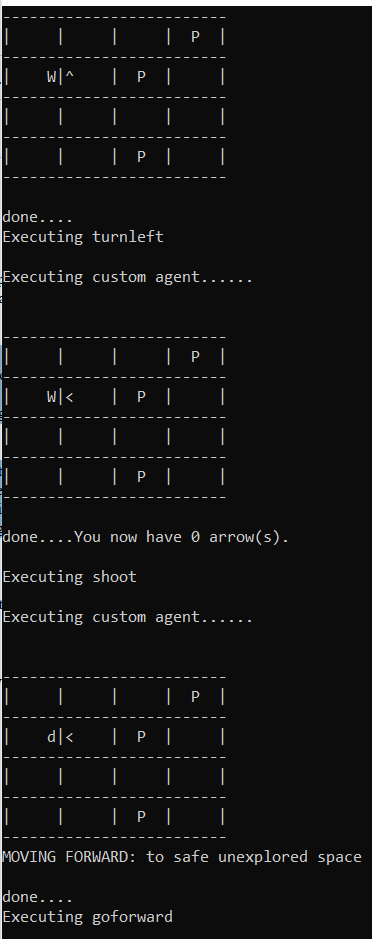
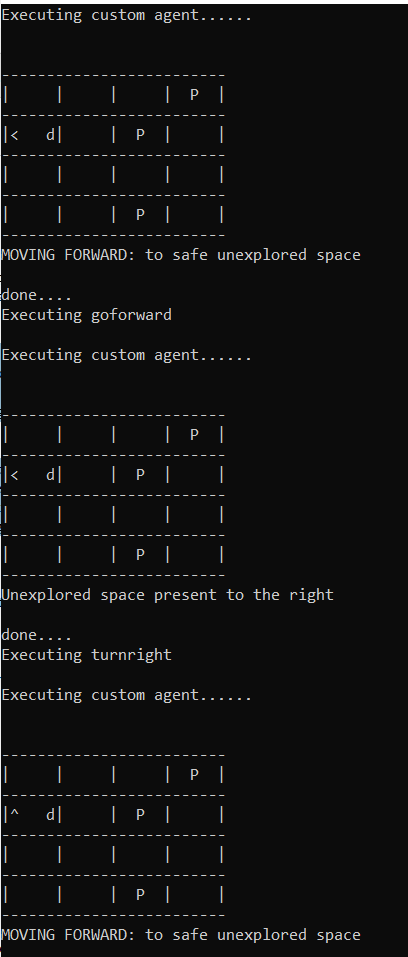
Go forward to safe unexplored space (2,1)-> experienced breeze, hence nothing to do, turn right-> excountered wall, not safe, hence turn right-> go forward to safe 1,1-> turn left to unexplored space-> wall encountered, not safe, turn left-> turn left to unexplored space again-> go forward to 1,2-> encountered stench, hence turn right to unexplored space-> move forward to safe space 2,2-> go forward to safe space 3,2-> experience breeze, turn right-> turnright to retrack back->go forward to 2,2-> turn right to unexplored space-> encountered glitter, hence go forward to 2,3-> Grab Gold-> turn left to unexplored space and also encountered stench and pit-> found wumpus, shoot wumpus-> go forward to 1,3-> wall encountered, not safe, turn right to unexplored space-> go forward to 1,4-> encountered a wall, turn left-> forward not safe since wall encountered, turn left-> unexplored space present to left, turn left-> go forward to 2,4-> go forward to 3,4 since its safe-> encountered breeze-> turn right-> turn right->go forward to safe space 2,4-> go forward to 1,4-> forward not safe since wall encountered, turn left-> go forward to 1,3-> already explored this space, hence move forward to 1,2-> turn right to unexplored space-> wall encountered, not safe, hence turn left-> go forward to safe space 1,1-> climb out.

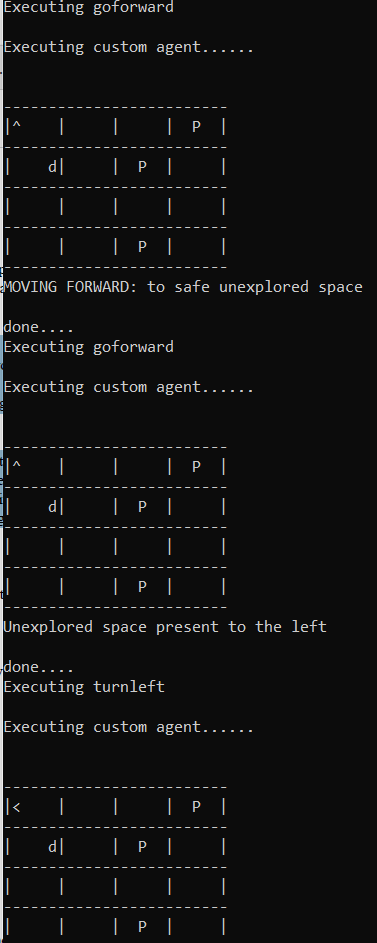
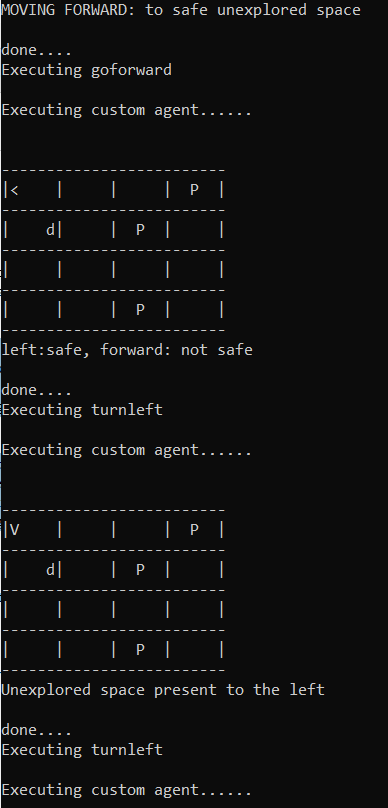
 

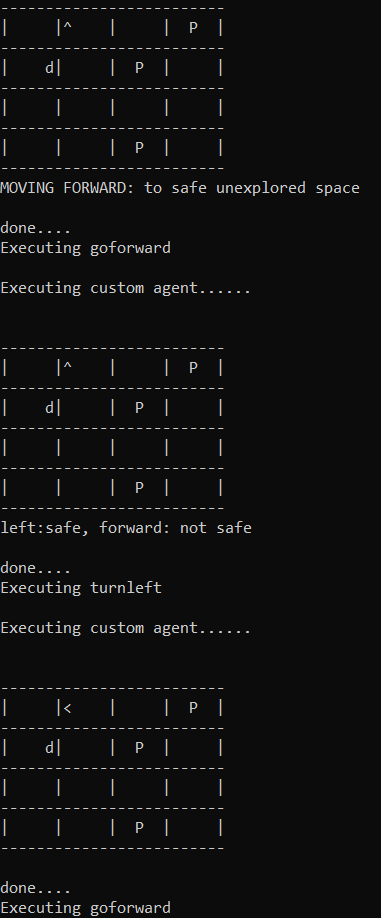
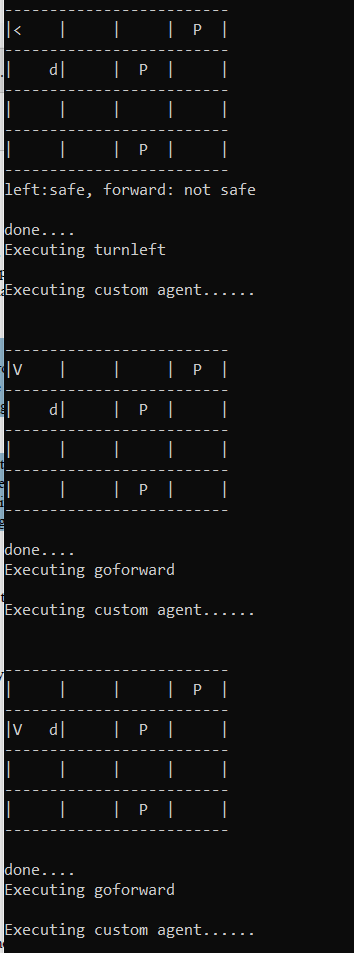
 

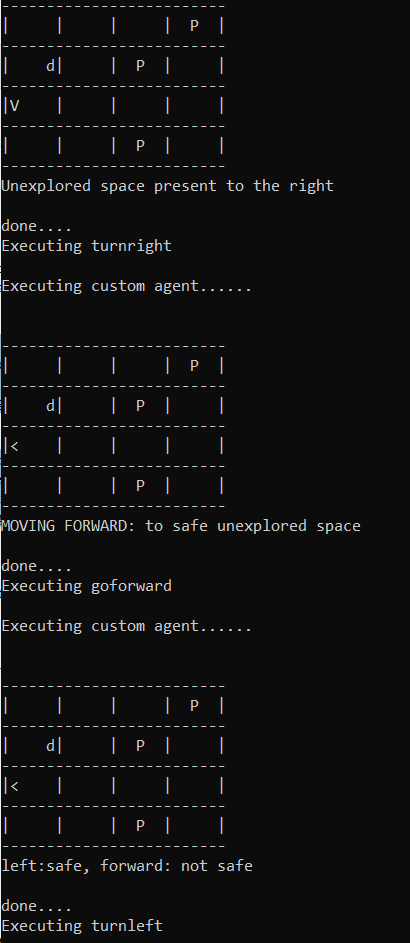
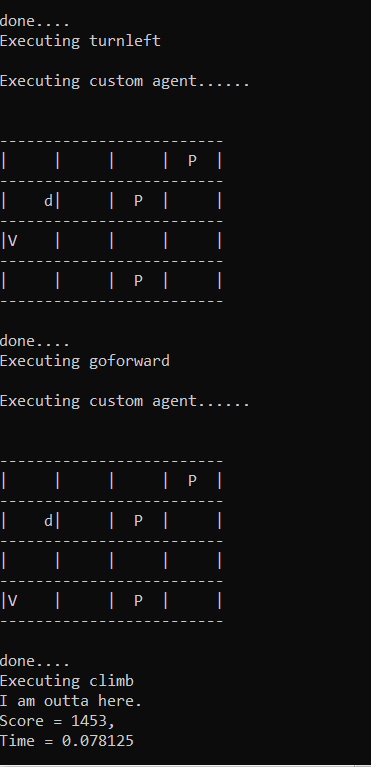
 

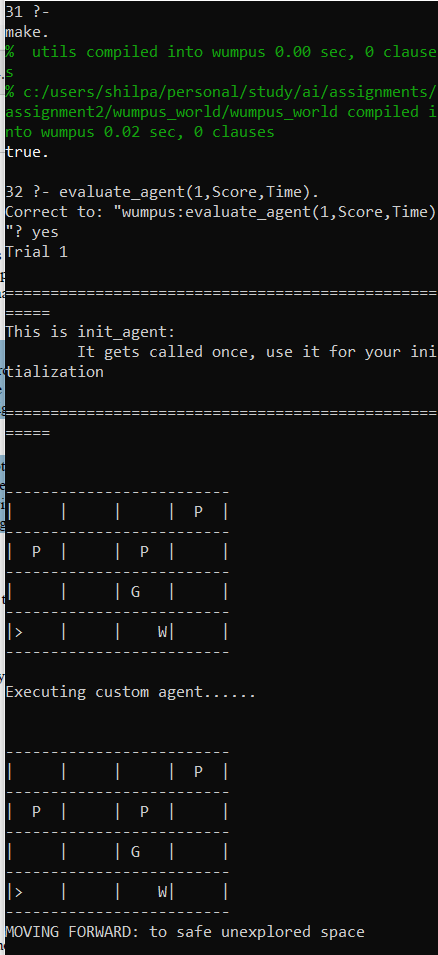
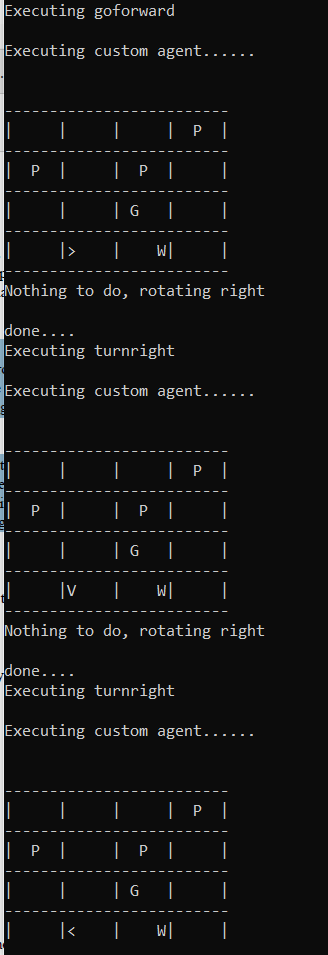
 

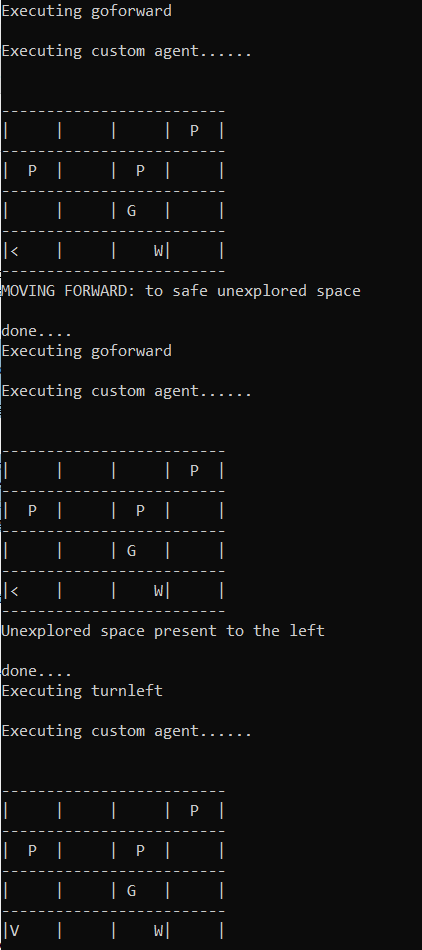
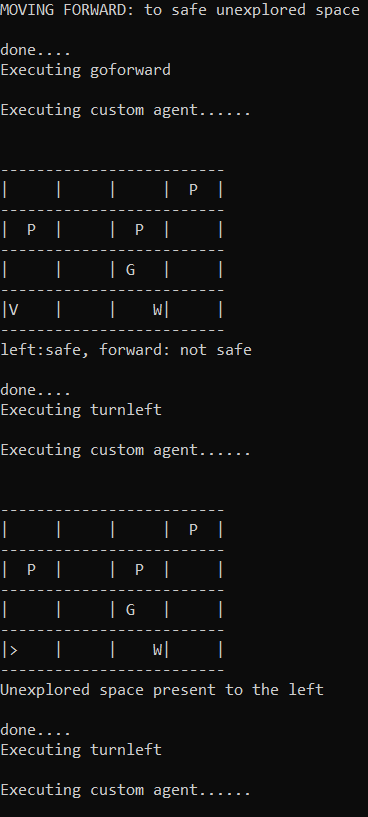
 

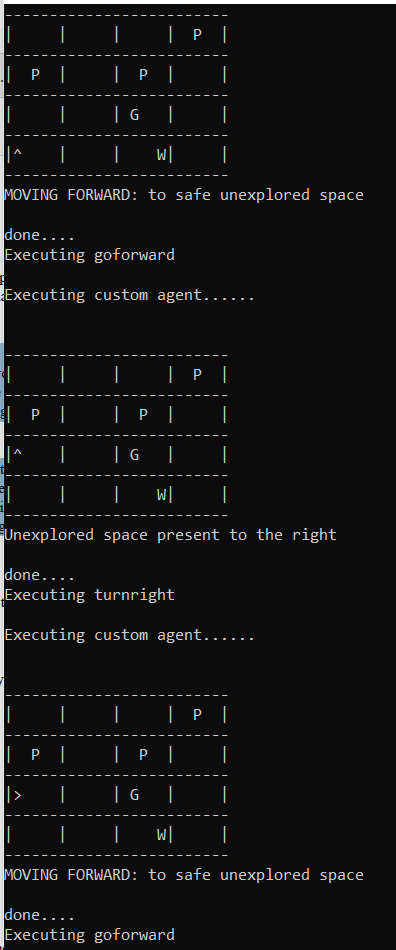
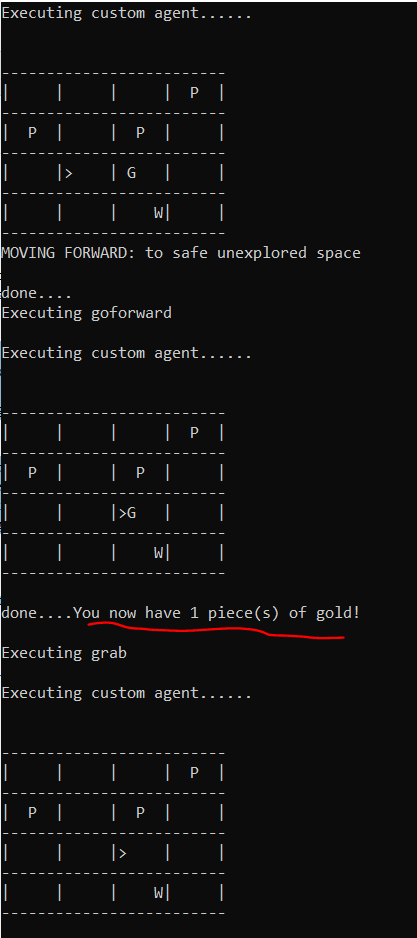
 

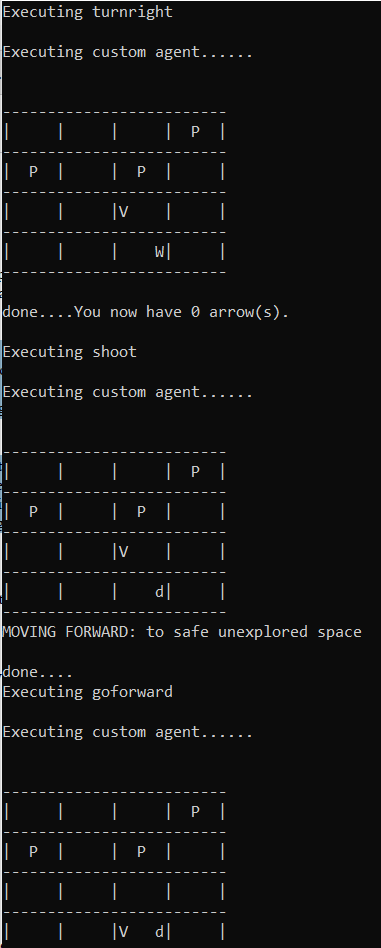
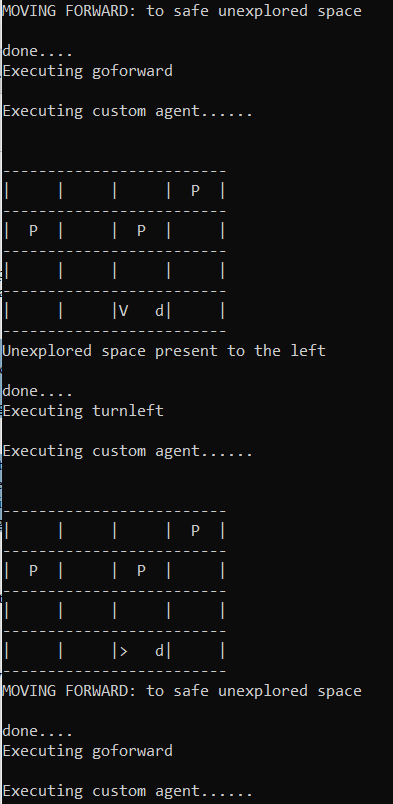
1. Map\_7

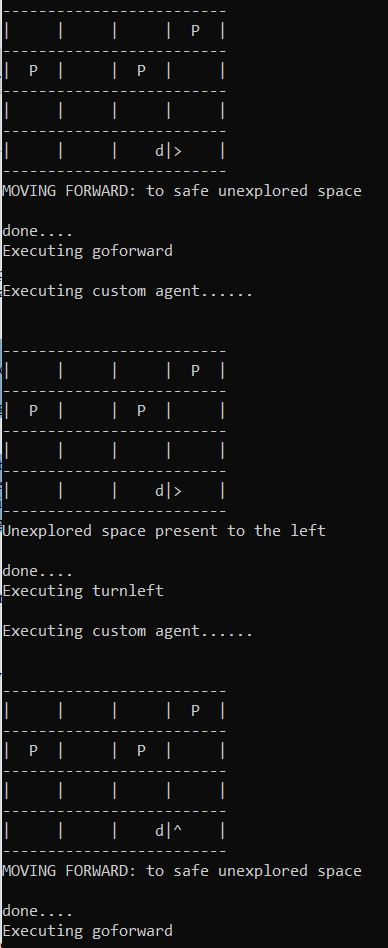
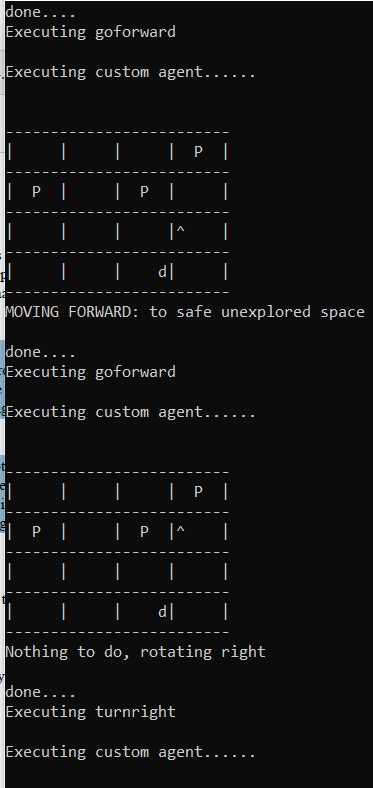
* The agent grabs gold and kills the Wumpus both
* Go forward to (2,1)-> encountered a stench, hence not safe to move forward, turn right to unexplored safe-> encountered wall, hence not safe, turn right again to safe space-> go forward to 1,1-> turn left to unexplored space since forward is not safe due to wall-> turn left since wall encountered again-> turn left to unexplored space->go forward to 1,2-> encountered breeze, right space is safe, hence turn right-> go forward to 2,2->encounters only glitter, hence move forward to 3,2->Grab Gold-> encounter stench,turn right-> shoot wumpus and kill-> move forward to 3,1 safely-> forward is not safe, since there is a wall, turn left to unexplored space-> go forward to 4,1-> forward is not safe, hence turn left-> go forward to 4,2-> go forward to safe space 4,3-> breeze felt, turned right to unexplored space-> not safe to move forward, turn right since nothing else to do-> go forward to safe space 4,2-> turn right to unexplored space->wall encountered, not safe-> turn left-> go forward to 4,3-> already seen and having breeze, turn right-> turn right-> go forward to 4,2-> go forward to 4,1->nothing to do, turn right-> go forward to 3,1-> go forward to 2,1-> turn left to unexplored space-> wall encountered, turn left-> go forward to 3,1-> go forward to 4,1-> turn left-> go forward to 4,2-> go forward to 4,3-> turn right nothing to do-> turn right nothing to do since all these moves are already seen and stored in database-> go forward to 4,2-> go forward to 4,1-> turn right since nothing else to do-> go forward to 3,1-> go forward to 2,1-> go forward to 1,1-> climb out.

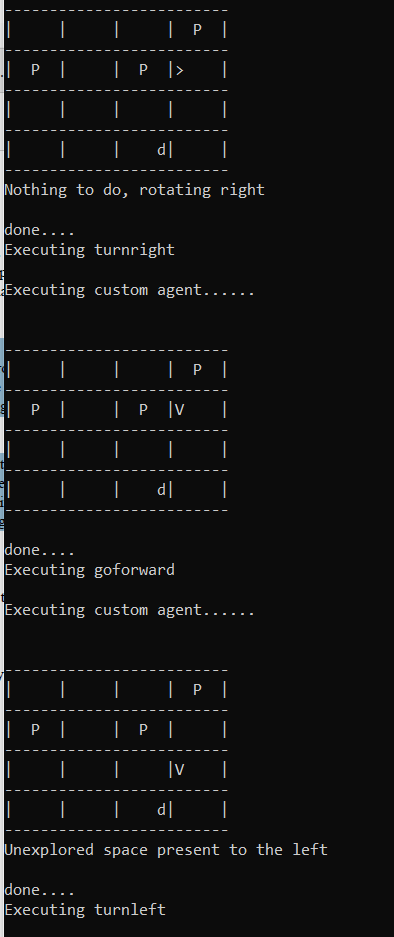
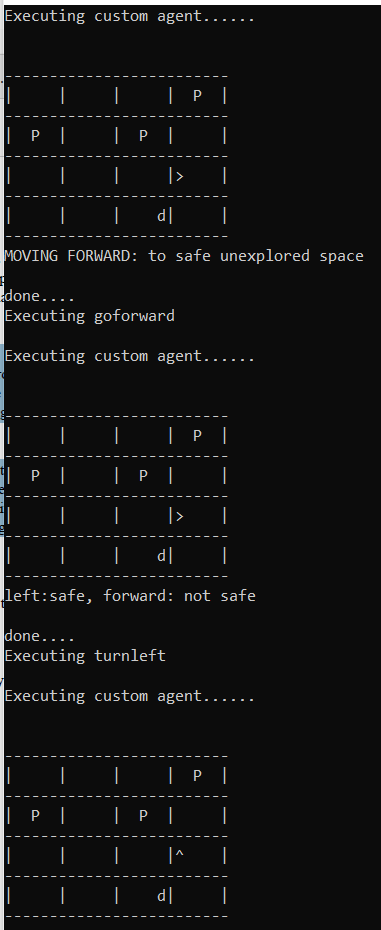
 

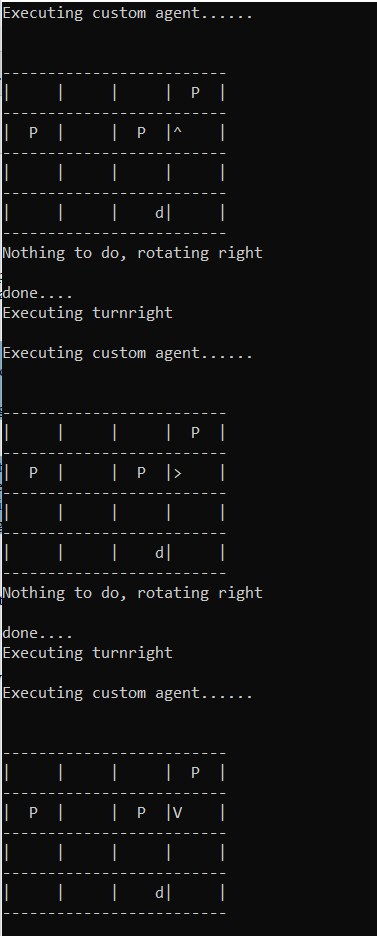
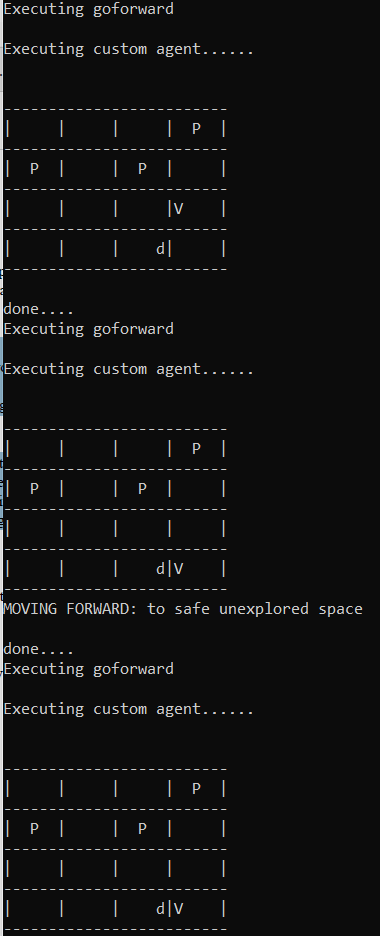
 

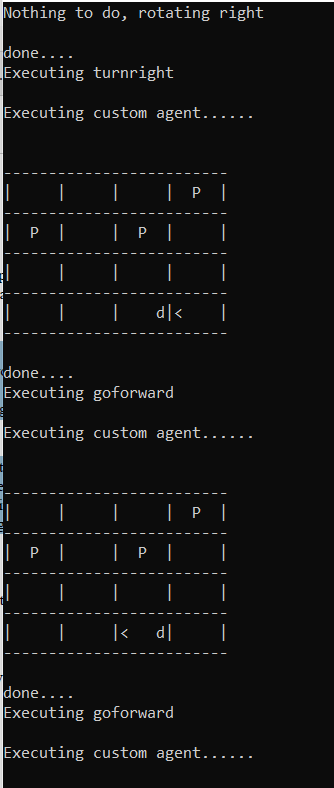
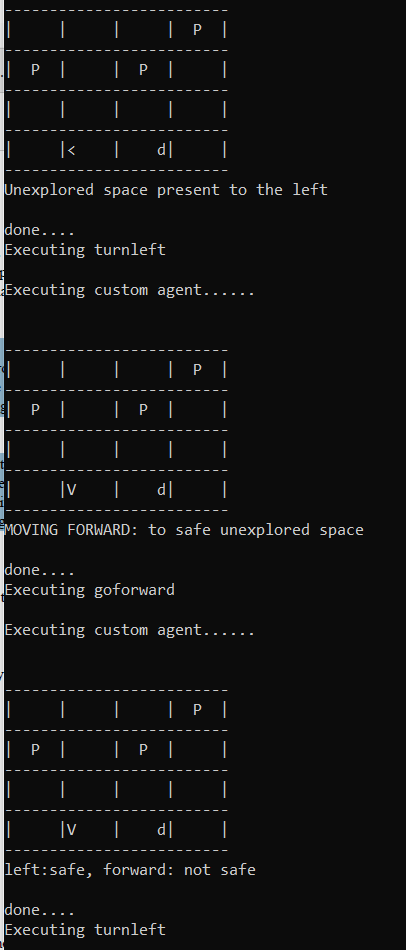
 

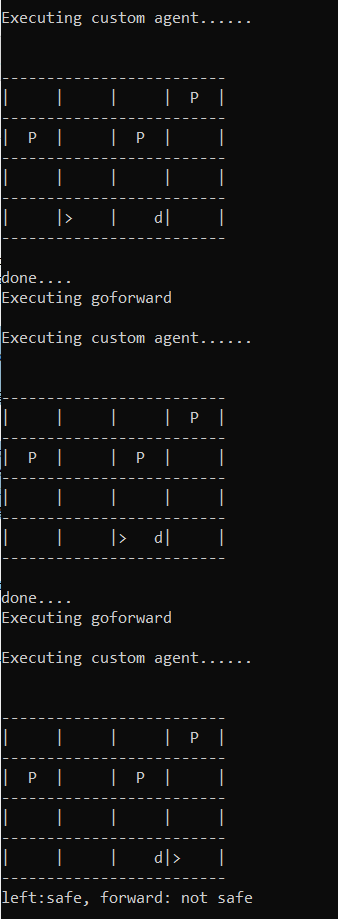
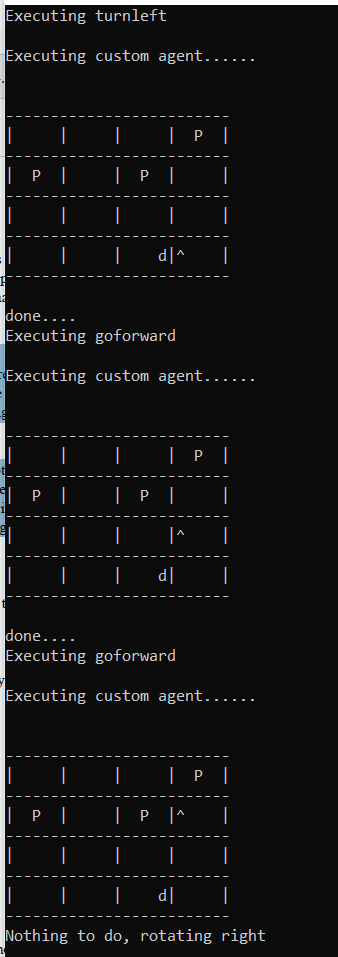
 

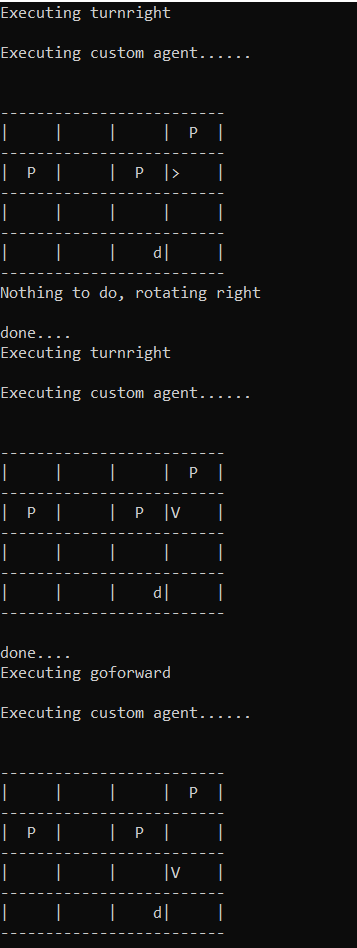
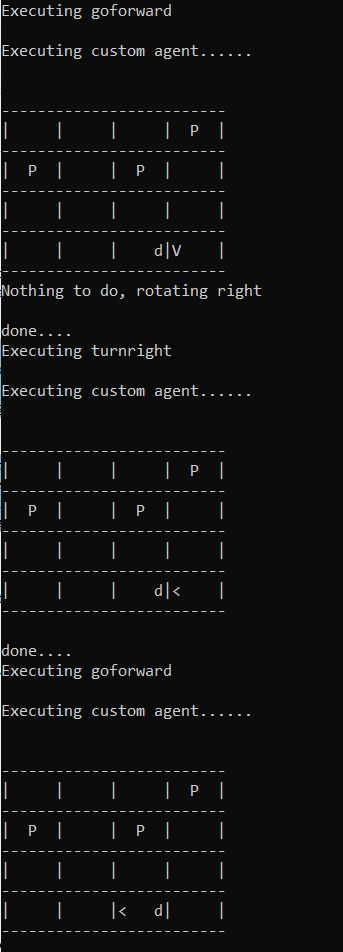
 



REFERENCES:

1. <https://softwareengineering.stackexchange.com/questions/84407/why-is-prolog-good-for-ai-programming>
2. Book: Prolog Programming for Artificial Intelligence 3rd Edition by Ivan Bratko
3. SwiProlog Documentations at <https://www.swi-prolog.org/pldoc>
4. <http://www.cs.trincoll.edu/~ram/cpsc352/notes/prolog/search.html>