

NANYANG TECHNOLOGICAL UNIVERSITY

SINGAPORE

CZ3005 Artificial Intelligence Lab Assignment 2

Team Name: Benjababe
Lab Group: SSP1

Member

Name	Matriculation No.
Goh Zongye Benjamin	U2022728L

Introduction

For all of the tests shown, the absolute map below will be used for the sake of consistency. Tests are also split into segments in a way I felt like the components are grouped up best. Relative maps are printed only for the last action in a sequence and all action/2 queries are printed. Order of tests in the report also follow the order in the driver printout.

```

===[Complete Absolute Map]===

-----

|###|###|###|###|###|###|###|###|###|###|
|###|###|###|###|###|###|###|###|###|###|
|###|###|###|###|###|###|###|###|###|###|
-----

|###|. . . T|. . .|. . . T|. . .|. . . T|###|
|###| ? | - O -| ? | - O -| ? |###|
|###|. . .|. . .|. . .|. . .|. . .|###|
-----

|###|. . .|. . . T|. . .|. . . T|. . .|###|
|###| ? | ? | ? | ? | ? |###|
|###|. . .|. . .|*|. . .|. . .|###|
-----

|###|. . .|. . .|. . .|. . . T|. . .|###|
|###| ? | ? | - ^ -| ? | ? |###|
|###|. . .|. . .|. . .|. . .|. . .|###|
-----

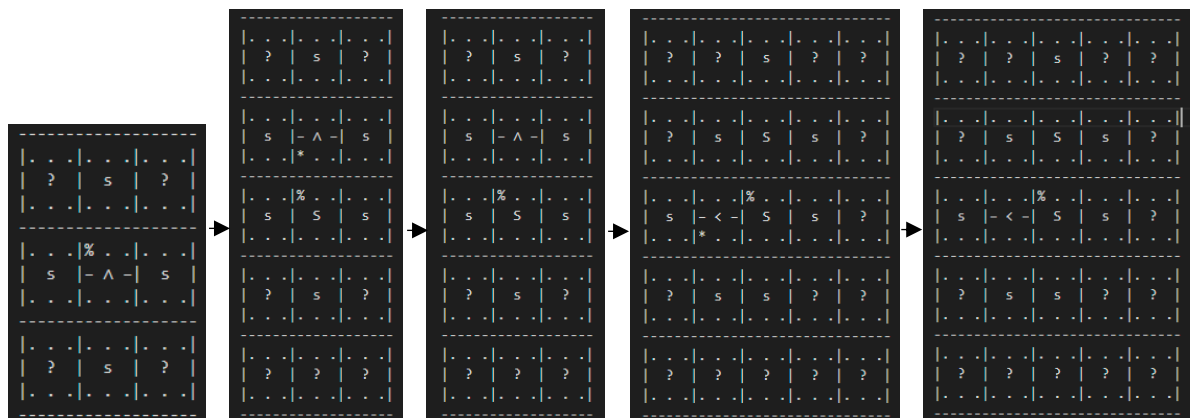
|###|. . .|. . .|. . . T|. . .|. . . T|###|
|###| ? | ? | ? | - O -| - W -|###|
|###|. . .|. . .|. . .|. . .|. . .|###|
-----

|###|###|###|###|###|###|###|###|###|###|
|###|###|###|###|###|###|###|###|###|###|
|###|###|###|###|###|###|###|###|###|###|
-----

```

Testing movement, pickup, glitter/2, safe/2 and visited/2

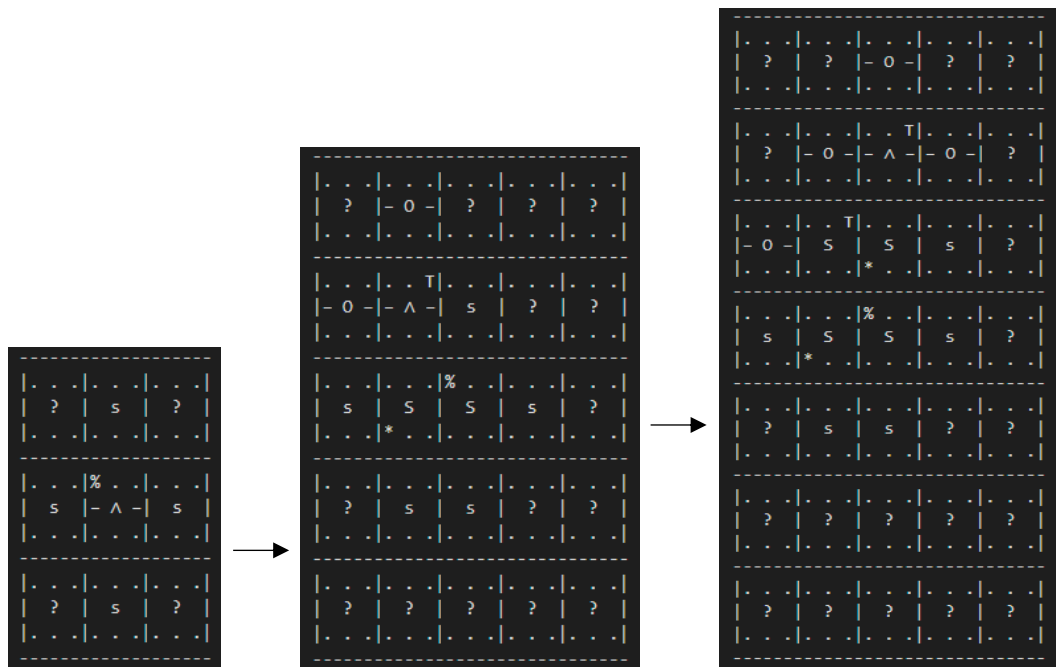
Initially, the agent spawns at (3, 3) on the absolute map which has 2 adjacent coins to it. The driver moves the agent to (3, 2) which has a coin and includes the glitter in the perceptions list. The relative map updates to include the new frontier cells and the glitter spot. Then agent sends the pickup action which removes the glitter from the cell. The agent is then moved to (2, 3) and the same action command is repeated. Sequence of images shows the relative map being updated accordingly.



Testing confundus/2 and tingle/2

The driver initially moves the agent to absolute (-1, 1) where there is a “tingle”. When the driver senses “tingle”, all its adjacent cells will be updated with a possible confundus portal location. This is except for any cells that have been visited or are deemed safe prior. The agent will also always check for all possible confundus locations, all visited cells adjacent also have a “tingle” to ensure that they are possibly portals.

The driver then moves the agent to (0, 2), another “tingle” cell, to show more updates for possible portal locations, maintaining the rules for a possible portal cell. Sequence of images show the map updates after the movements.

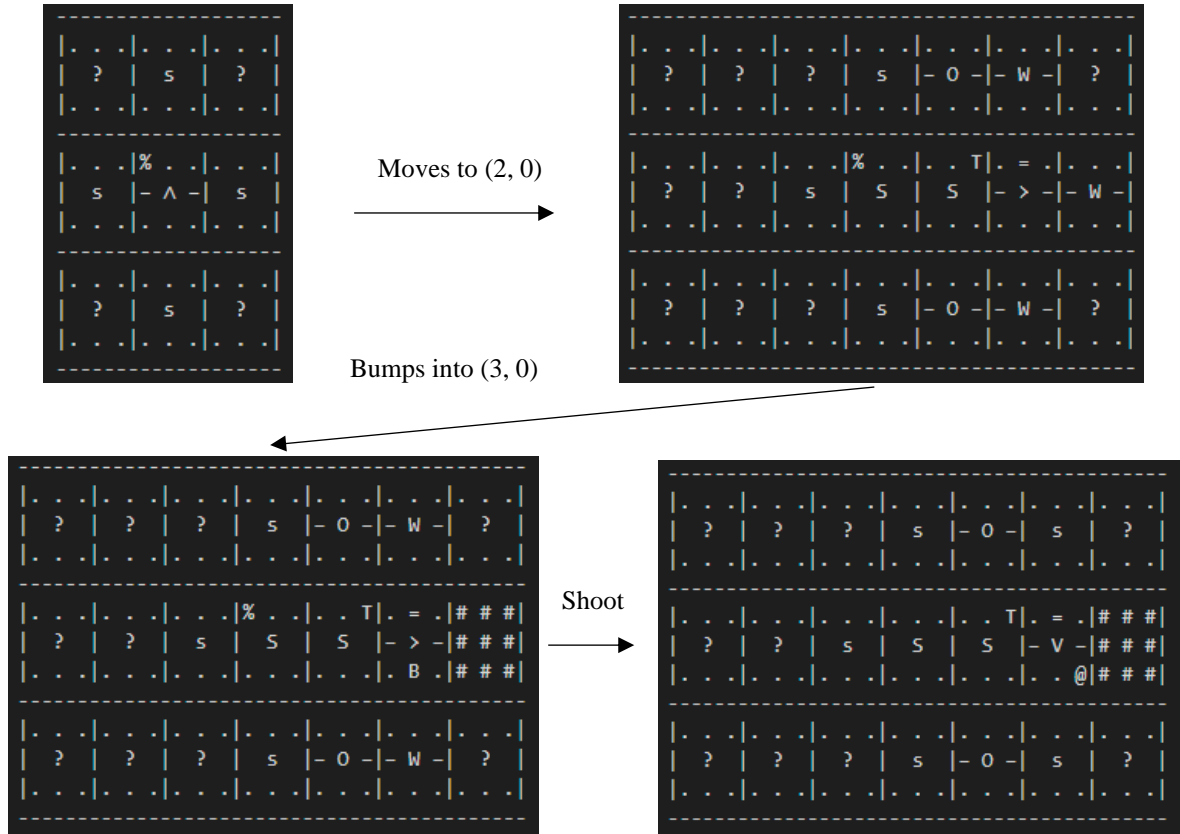


Testing stench/2, wumpus/2, wall/2, hasarrow/0, bump, shoot and scream

The driver moves the agent to absolute (2, 0) where there is a “stench”. When the driver senses “stench”, all its adjacent cells will be updated with a possible wumpus location. This is except for any cells that have been visited or are deemed safe prior, like the confundus portal requirements. The agent will also always check for all possible wumpus locations, that all visited cells adjacent also have a “stench” to ensure they are possibly wumpuses.

The driver then, while agent is facing absolute west, moves forward again to bump into the wall. The bump indicator will then show on the relative map and the possible wumpus for (3, 0) will be replaced with a wall. Driver then checks if the agent has the arrow and prints out the Boolean then turns right once to face the known wumpus spot, and shoots.

Upon shooting, the scream indicator will be sent to the agent as it hit the wumpus and all instances of stench/2 and wumpus/2 should be retracted from the agent. The relative map will update accordingly. Driver then checks for the arrow again and prints its Boolean value.



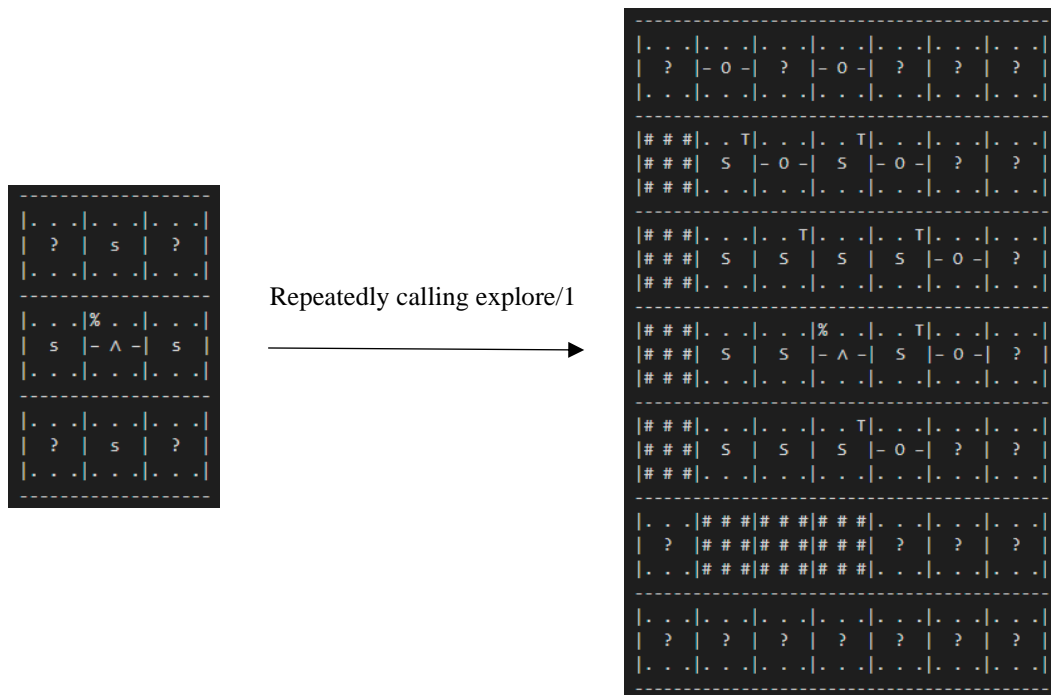
Testing reposition/1

For this test, the driver moves the agent to absolute (0, 2), ensuring both absolute and relative directions are west and steps the agent into the confundus portal. When the agent steps onto the portal, it will be repositioned to absolute (1, 4, west) and relative (0, 0, north), as well as most of the mapping gone from memory.

Any movement from the agent now should be oriented differently between absolute and relative maps (as if the relative map is absolute map rotated 90° counter-clockwise) and any cell updates should be consistent between them.

Testing explore/1

For explore/1, the driver can just simply keep calling it until it returns an empty list which will indicate that the agent has no free explorable paths and is at relative (0, 0). The final relative map should be a single cluster of visited cells bordered by walls, possible portal or wumpus cells.



Friend Driver

The friend driver has been obtained from team EasyClap and my own driver was ran on team EasyClap's agent.

Conclusion

From the grueling 4 weeks I have spent on this assignment, I have learnt that prolog is pretty useful as a logic programming language, having the innate ability to recursively search for variables to fit an input combination.

Getting started with prolog was probably the biggest learning curve as it had little similarities with functional programming and finding a group instead of doing this assignment by myself would've greatly relieved myself of some burdens.