ID	Name
18127259	Kiều Công Hậu
18127267	Trần Đình Sang

Introduction to Aritificial Intelligence

18CLC1

Project: Wumpus World

REPORT

I. Assignment plan:

No.	Assignments	Implement
1	Design and implement graphics.	Trần Đình Sang
2	Design maps.	Trần Đình Sang
3	Design and implement algorithms.	Kiều Công Hậu
4	Report	Kiều Công Hậu
5	Test.	Trần Đình Sang

II. Estimating the degree of completion level for each requirement:

No.	Requirements	Degree of completion
1	Implement code to explore the Wumpus World (using Proposional Logic).	100%
2	Output information about the search, including percepts at every room the agent enters, the content of or change in knowledge base after each new percept, and the action decided upon by the Agent.	100%
3	Load the world setup of Figure 1 in the problem description file.	100%

4	Finish problem successfully.	100%
5	Graphical demonstration of each step of the running process. (Console screen + Graphical library).	100%
6	Generate at least 5 maps with difference structures such as position and number of Pit, Gold and Wumpus.	100%
7	Report the algorithm, experiment with some reflection or comments.	100%

Because of the limit time, we just display on the graphical screen some essential actions and inferences of the Agent at every room:

- Perceive Stench and Breeze
- Shoot
- Kill Wumpus
- Detect Pit
- Grab gold
- Climb out of the cave

The whole actions and inferences at each room are listed carefully in the output text files in the folder ***Assets/Output/*** and printed onto the Console screen.

There are many actions in the output file texts:

- TURN_LEFT: the Agent turns left.
- TURN_RIGHT: the Agent turns right.
- TURN_UP: the Agent turns up.
- TURN_DOWN: the Agents turn down.
- MOVE_FORWARD: the Agent moves forward.
- GRAB_GOLD: the Agent grabs gold.

- PERCEIVE_BREEZE: the Agent perceives Breeze.
- PERCEIVE_STENCH: the Agent perceive Stench.
- SHOOT: the Agent shoot an arrow to the opposite room.
- KILL_WUMPUS: the Agent knows that he has killed the Wumpus.
- KILL_NO_WUMPUS: the Agent miss a shooting (the Agent don't know this).
- BE_EATEN_BY_WUMPUS: the Agent is eaten by the Wumpus.
- FALL_INTO_PIT: the Agent falls into the Pit.
- KILL_ALL_WUMPUS_AND_GRAB_ALL_FOOD: the Agent kills all of Wumpus and grabs all of Gold.
- CLIMB_OUT_OF_THE_CAVE: the Agent climbs out of the cave.
- DECTECT_PIT: the Agent detect a Pit at the opposite room (inference from KB).
- DETECT_WUMPUS: the Agent detect a Wumpus at the opposite room (inference from the KB).
- DETECT_NO_PIT: the Agent detect no Pit at the opposite room (inference from KB).
- DETECT_NO_WUMPUS: the Agent detect no Wumpus at the opposite room (inference from the KB).
- INFER_PIT: the Agent infers whether there is a Pit at the opposite room.
- INFER_NOT_PIT: the Agent infers whether there is no Pit at the opposite room.
- INFER_WUMPUS: the Agent infers whether there is a Wumpus at the opposite room.
- INFER_NOT_WUMPUS: the Agent infers whether there is no Wumpus at the opposite room.

About the Knowledge Base, there are some rules:

- A literal is represented as a number.
    - For example, to represent that there is Stench at room (1, 1), the literal will be 4001, 4 stands for Stench, 001 stands for room (1, 1).

- Another example, to represent that there is no Pit at room (2, 2), the literal will be -1012, -1 stands for no Pit, 012 stands for room (2, 2).
- Generally, 1 stands for Pit, 2 stands for Wumpus, 3 stands for Breeze, 4 stands for Stench, a nagetive sign stand for NOT operator, the last 3 digit stands for the room.

10	091	092	...	099	100
9	081	082	...	089	090
⋮	⋮	⋮	⁘	⋮	⋮
2	011	012	...	019	020
1	001	002	...	009	010
	1	2	...	9	10

- A clause is represented as a list of literals (list of numbers).
  - For example, to present there is a Pit at room (1, 1) OR there is a Breeze at room (2, 2), the clause will be [1001, 3012].
- A setences is represented as a list of lists (list of clauses).
  - For example, to present there is a Pit at room (1, 1) AND there is a Breeze at room(2, 2), the sentences will be [[1001], [3012]].

III. Directories structure:

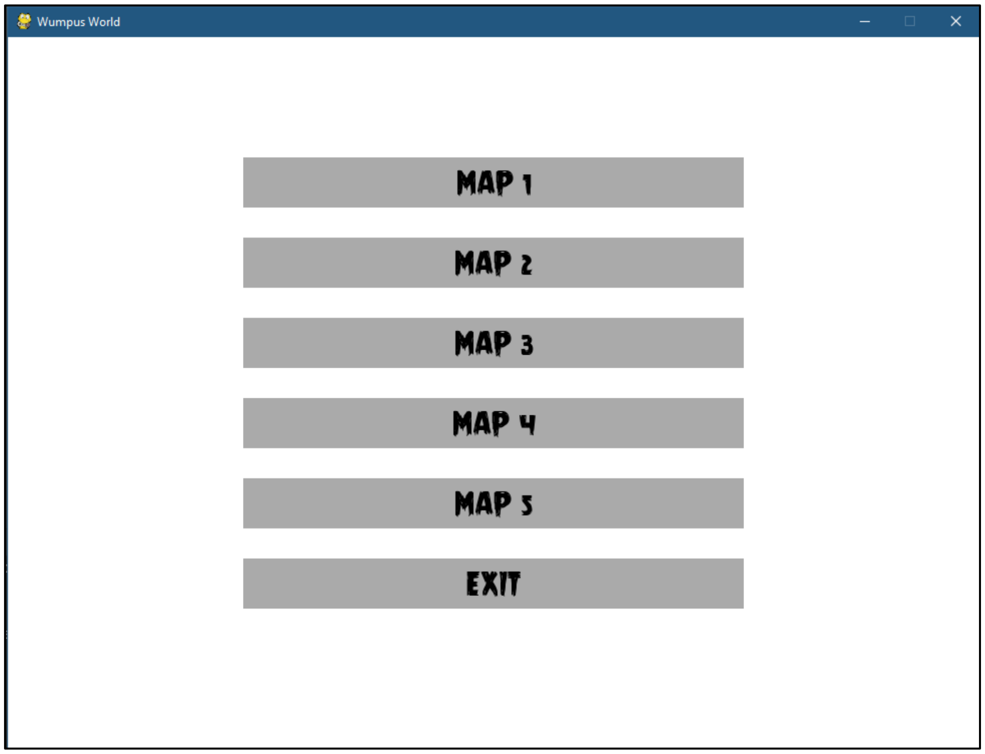
- ***Assets***: this folder contains images for GUI, map's inputs (text file), map' outputs (text file).
- ***Source***: this folder contains the source code.
- ***Report.pdf***: report.

IV. Enviroment:

- Language: Python (3.7.6).
- Library:
  - pygame
  - pysat.solvers
  - enum
  - copy
  - sys

V. Instruction:

- Step 1: Compile and build the program at file ***Main.py*** in folder ***Source***.
- Step 2: Choose 1 of 5 maps.



- Step 3: The Agent will automatically explore the cave, kill the Wumpus, grab Gold and climb out of the cave.

VI. Algorithms:

- To explore all of rooms without missing no room, we use the Back-tracking algorithm. For example, the Agent start at room (1, 1), then Agent move to room (2, 1), Agent perceives

Breeze, then Agent backtrack to the room (1, 1) and move to the room (1, 2), where the Agent perceives Stench. This algorithm works recursively.

Stench	
Agent	Breeze

- When the Agent moves to a room:
  - If there is a Wumpus or a Pit, the Agent dies.
  - If there is Gold, the Agent grab Gold.
  - If there is Stench or Breeze, the Agent add some rules to the KB:
    - This room has Stench and Breeze or not?
    - If this room has Stench, there is at least one of all adjacent rooms has a Wumpus.
      S ⇔ Wa OR Wb OR Wc OR Wd
    - If this room has Breeze, there is at least one of all adjacent rooms has a Pit.
      B ⇔ Pa OR Pb OR Pc OR Pd
    - Pit and Wumpus can not appear at the same room.
    - If this cell don't have Stench, all of adjacent rooms can not have a Wumpus.
    - If this cell don't have Breeze, all of adjacent rooms can not have a Pit.
  - If this room has Stench, the Agent will infer that whether there a Wumpus at one of the adjacent rooms. If the Agent detect a Wumpus, the Agent will shoot an arrow, the room which has a Wumpus which has been killed is safe now. But the Agent only knows that he has killed a Wumpus if this current room has no Stench anymore.
  - After some inference, if the current room still has Stench, the Agent will try to shoot at any directions till the Stench of this room disappears.
  - If this room has Breeze, the Agent will infer that whether a Pit at one of the adjacent rooms. If the Agent detect a Pit, the graphical screen will display it.
- After some inferences, if the Agent can explore the new rooms from this current room, the Agent will move to each room sequentially, else the Agent will backtrack to the previous room.

- Specially, if the Wumpus is killed by the Agent, the Agent must update the Stench states at the corresponding rooms as well as the Knowledge Base.
- To infer something, we tried to use the ourselves-implement pl-resolution algorithm but it run so slowly, so we decide to use the package Glucose3 from pysat.solvers to install the function to infer instead. The idea is if you want to infer alpha, you need to prove that the KB and not alpha is unsatisfiable, which means there is no models satisfy the KB and not alpha. g.solve() of Glucose3 will help us to find at least a model which satisft KB and not alpha, if yes, we can not infer alpha, else we can infer alpha from the KB.

```
- def infer(self, not_alpha):
      g = Glucose3()
      clause_list = copy.deepcopy(self.KB)
      negative_alpha = not_alpha
      for it in clause_list:
          g.add_clause(it)
      for it in negative_alpha:
          g.add_clause(it)
      sol = g.solve()
      if sol:
          return False
      return True
```

VII. Comments:

There is a special case that: if the initial room where the Agent start has only Breeze, the Agent will can not infer anything and climb out of the cave immediatelty. If at that room has Stench also, the Agent will cannot infer anything, then he will try to shoot at any adjacent rooms till the current room has no Stench anymore, then the last adjacent room the Agent shoot is the safe room, he can move to that room.

VIII. References:

- Source code of Problem 3 – Homework 03 (Puzzle game board) of this course.