

## CSC14003 – Artificial Intelligence

### PROJECT 02: LOGICAL AGENT

#### 1. Project description

The purpose of this project is to design and implement a *logical agent* that navigates through the Wumpus World, a partially-observable environment.

The Wumpus World presents the following key features:

- The environment is an underground cave with a network of interconnected two-dimensional rooms.
- A room may contain a *deadly pit*, signaled by a perceivable *breeze*, or a *fatal Wumpus monster*, detectable via a discernible *stench*.
  - The agent will die immediately when entering a room containing one of those harmful factors. No withdrawal is possible,
  - The percepts are available in the *four-neighborhood of the room* containing one of those harmful factors.
- The agent has an *arrow* to shoot in the direction he is facing.
- There is one *chest of gold*, located somewhere in the cave.
- Movement options: *forward, backward, left, or right by 90 degrees*.

The primary objective encompasses locating the gold. During the journey, the agent may need to kill Wumpus to pursue success in this environment.

Figure 1 demonstrates an example of the Wumpus world with key entities.

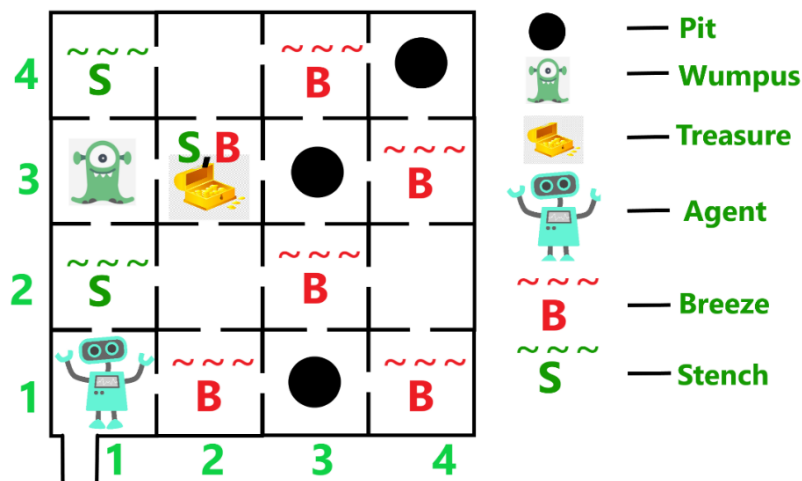


Figure 1. An example of the Wumpus world with key entities.

## 2. Wumpus World

Our Wumpus World for actual implementation slightly differs from the original problem description. The environment's specification is as follows.

- The world will be limited in a grid of  $10 \times 10$  rooms, instead of the  $4 \times 4$  one.
  - The coordinate of a room is represented as a pair of (x, y) values, in which x is the position in the horizontal axis (increasing from left to right) and y is the position in the vertical axis (going from bottom to top).
  - Room (1, 1) is the bottom-left one, while room (10, 10) the top-right one.
- Agent can start at any room (x, y) and he always faces to the right.
- **Room exit is room (1, 1). The exit door side is like the one in Figure 1**
- There may be any number of pits and chests of gold in the world.
- There is at least one Wumpus.
- The agent carries an infinite number of arrows.
- **When the Wumpus dies. It will scream and you will know this information.**
- **After Wumpus dies, the stench of that Wumpus will disappear (you should update the state of map after Wumpus dies)**
- The game will end when one of the following **two** conditions occurs:
  - The agent dies.
  - The agent climbs out of the cave (**exit the cave**).

The following activities can give the agent certain amounts of points.

Activity	Points
Pick up each chest of gold	<b>+1000</b>
Shoot an arrow	– 100
Killed by the Wumpus or Fall into a pit	– 10,000
Climb of the cave	+ 10
Move from one room to an adjacent room	– 10

## 3. Tasks

Your group must **set up a program to explore the Wumpus world and get the highest score possible for that world**, using either Propositional Logic or First-Order Logic (or both).

Your implementation should *output information about the search*, including the *percepts at every room the agent enters*, the *updates in the knowledge base after each new percept*, and the *action decided upon by the agent*. The program should also *output the score of the agent*.

Having your implementation generate worlds randomly can help you test your agent.

## 4. Specifications

**Input:** the given map is represented by matrix, which is stored in the input file, for example, map1.txt. The input file format is described as follows:

- The first line contains an integer N, which is the size of map.
- N next lines with each line represents a string. If room empty, it is marked by hyphen character (-). If room has some things or signal such as Wumpus(W), Pit(P), Breeze(B), Stench(S), Agent(A) or Gold(G). Between two adjacent rooms is separated by a dot (.)
- **Input only includes Wumpus(W), Pit(P), Agent(A) and Gold(G). You need to update information about Stench(S) and Breeze(B) on the map based on input data.**
- For example: -.-W.-.P.-.-.-.-

	S	W	BS	P	B				
--	---	---	----	---	---	--	--	--	--

**Output:** a result with path for agent, game point will be displayed on screen and/or written in output text file such as result1.txt.

## 5. Requirements

No.	Specifications	Scores
1	Finish problem successfully.	50%
2	Graphical demonstration of each step of the running process. You can demo in console screen or use any other graphical library.	10%
3	Generate at least 5 maps with difference structures such as position and number of Pit, Gold and Wumpus.	20%
4	Report your algorithm, experiment with some reflection or comments.	20%
<b>Total</b>		<b>100%</b>

## 6. Notice

This assignment will be completed in **groups**, as registered at the beginning of the course. To prepare, you will need to create a folder that contains various subfolders, including source, input, output, and document. The report must give the following information:

- Your detailed information (Student Id, Full Name)
- Assignment Plan
- Environment to compile and run your program.
- Estimating the degree of completion level for each requirement.

- References (if any)

Your team can use any programming language to be, but Python is encouraged

**You must design and code the Logic by yourself. Not use library.**

**Any plagiarism, tricks, or any lie will have 0 points for the course grade.**