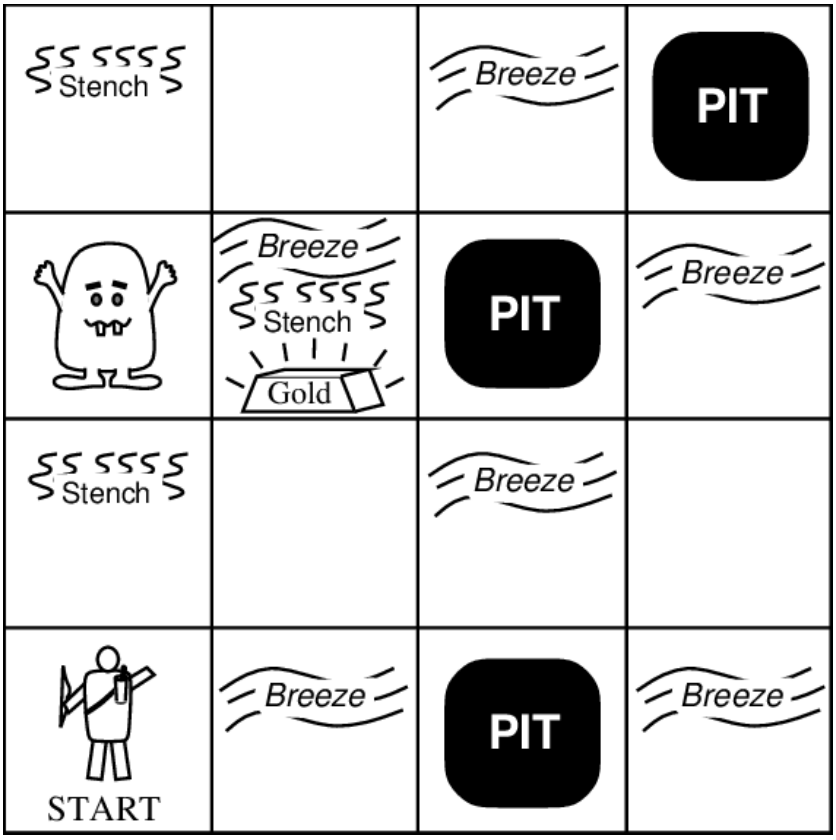
Project

Wumpus World

1. Description

The purpose of this project is to design and implement a logical search agent for a partially-observable environment. This will be accomplished by implementing an agent that navigates through the Wumpus World.



The basic properties of the Wumpus World are described in the textbook (Artificial Intelligent: A Modern Approach).

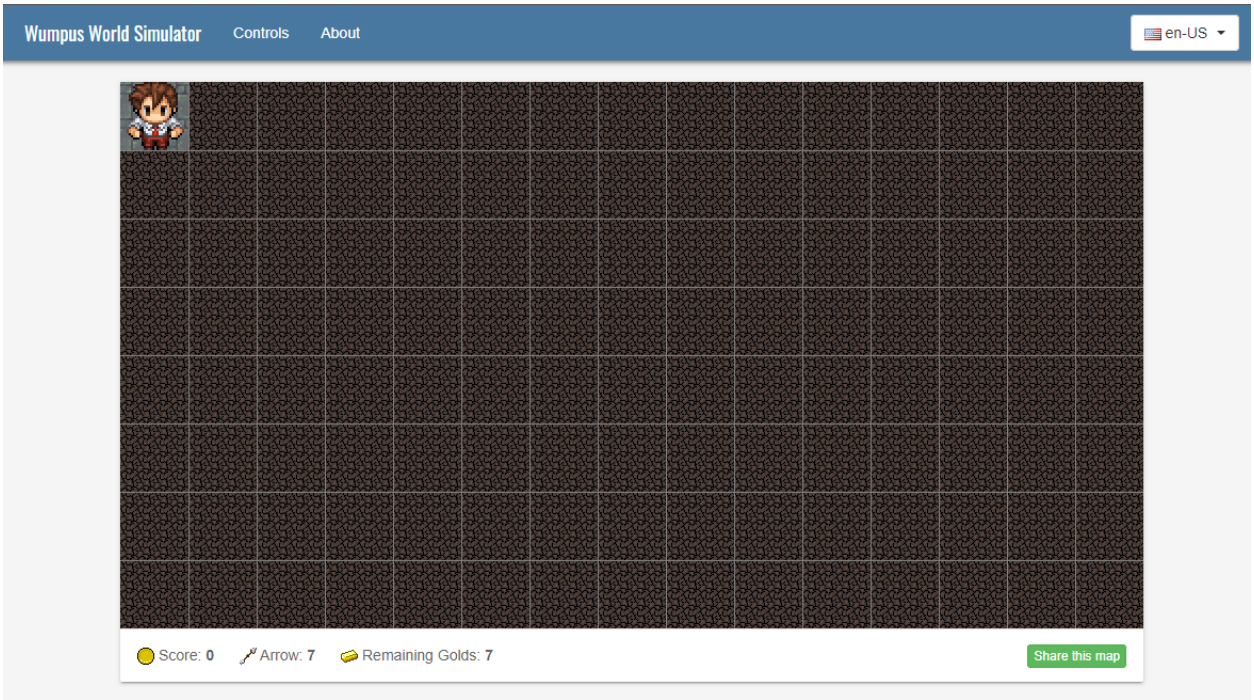
In summary, it is a game with:

- 2D cave connected by passages.
- Some rooms contain a pit into which we fall and perish.
    - We feel a breeze if near a pit.
- One room contains a Wumpus that will eat us.

  - We have one arrow that we can shoot in the direction we are facing.
  - We smell a stench if near the Wumpus.
- Somewhere, there is a pot of gold.
- We can move forward or backward, turn left by 90 degrees or right by 90 degrees.
- Find gold (if possible) and try and kill the Wumpus.

You can download and play game on http://www.dreamcodex.com/wumpus.php. Java RE must be installed in your computer. You should download zip and double-click the HTW.bat batch file.

You can also play game on https://thiagodnf.github.io/wumpus-world-simulator/ with a simpler version.



Note that: we just play game for understanding more about Wumpus World. Don't be downed in the game and forgot your mission.

2. New Wumpus World

We will modify the Wumpus world as such:

- The world will be limited in **10x10 instead of 4x4**. Room (1,1) will still be the bottom-left one, and Room (10,10) the top-right one. First number is room position in horizontal coordinate and second number is room position in vertical coordinate.
- Agent can **appear in any** Room (x_a, y_a) and always **facing to the right**. This room is the only room have the cave door.
- There may be **any number of pits and gold** in the world.
- There is **at least one Wumpus**.
- The agent carries an **infinite number of arrows**.
- The game will end when **one of the following three conditions** occurs:
    1. The agent dies
    2. The agent kills all of the Wumpus AND grabs all the gold
    3. The agent climbs out of the cave

The scores are as such:

- Add 100 points for picking up each gold.
- Reduce 100 points for shooting an arrow.
- Reduce 10000 points for dying (by being eaten by the Wumpus, falling in a pit).
- Add 10 point for climbing out of the cave.
- Reduce 10 points for moving from one room the the next.

Example:

G	G		B	P	B				
G	G	S	B	P	B			G	
	BS	W	BS	P	B				
B	P	BS		B					
P	B								
B						B			
					B	P	B		
						B	G	B	
							B	P	B
A								B	

Figure 1: The Wumpus World to test the agent for the report. The Agent's starting position is marked A. The other notes are: P = Pit, B = Breeze, W = Wumpus, S = Stench, G = Gold.

3. Tasks

Your group must implement code to explore the Wumpus World and get the highest score possible, using either Propositional Logic or First-Order Logic (or both).

Your implementation should output information about the search, including the precepts at every room the agent enters, the content of or change in the knowledge

base after each new precept, and the action decided upon by the agent. The program should also output the score of the agent.

Having your implementation generate Wumpus Worlds randomly can help you test your agent. However, it must also be able to load the world setup of Figure 1.

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 **W**umpus, **P**it, **B**reeze, **S**tench, or **A**gent, it is marked by first capitalized character in name of each type and written next to each other. Between two adjacent rooms is separated by a dot (.) For example:

	BS	W	BS	P	B				

  This row is represented: -.BS.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%
3	Graphical demonstration of each step of the running process. You can demo in console screen or use any other graphical library.	10%
4	Generate at least 5 maps with difference structures such as position and number of Pit, Gold and Wumpus.	20%

5	Report your algorithm, experiment with some reflection or comments.	20%
Total		100%

6. Notice

- This is a **GROUP** assignment. Each group has a maximum of 2 members.
- Duration: 2 weeks.
- Your group should use Python to do.
- Beside above requirements, report must also give the following information:
    - Your detail 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)
- Any plagiarism, any tricks, or any lie will have 0 point for course grade.