**Project 02**

**Wumpus World**

**Team member:**

Nguyễn Vũ Quang Minh 1651065

Phạm Thế Quyền 1651029

1. **Description**

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

1. **New Wumpus World**

We will modify the Wumpus world as such:

* The world will be limited in **10x10**.
* Agent can appear in any Room (**xa**, **ya**) 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 agent only has a limited time to explore the rooms, before the cave door collapses and the agent becomes trap inside forever. The agent only has time to visit **150** rooms.

The score 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, or being trapped inside the cave).
* **Add 10** point for **climbing out** of the cave.
* There is **no cost** for **moving** from one room to the next.

**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 represent the **N × N** map matrix. Each line contains M integers. The number at [i, j] (row i, column j) determines the state of rooms. 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 space (“ “).If room empty, it is marked by hyphen character (-).

**Output:**

**Level 1:** : Logical Search Agent

For this level, you must implement code to explore the Wumpus World and get the highest score possible, using First-Order Logic and resolution to solve it.

**Solution:**

The idea is every time the Agent enters a room, we build knowledge base (KB) based on state of the room the agent is inside.

**Case 1: room[i][j] = ‘-‘**

P\_i,j: This room has pit or not

W\_i,j: This room has wumpus or not

KB: P\_i+1,j = False ∧ P\_i-1,j = False ∧ P\_i,j+1 = False ∧ P\_i,j-1 = False ∧

W\_i+1,j = False ∧ W\_i-1,j = False ∧ W\_i,j+1 = False ∧ W\_i,j-1 = False

**Case 2: room[i][j] = ‘B’**

KB: P\_i+1,j = True ∨ P\_i-1,j = True ∨ P\_i,j+1 = True ∨ P\_i,j-1 = True ∧

**~**W\_i+1,j = True ∧ ~W\_i-1,j = True ∧ ~W\_i,j+1 = True ∧ ~W\_i,j-1 = True

Do the same with room[i][j] = ‘S’, ‘BS’. After that we will have a big list of knowledge base. We have a function to solve this KB and return a list of safe rooms for agent to enter.

We have a BFS function to find the way home from the current room to escape room with the opened room (the rooms have been passed)