

Project: Wumpus World Agent

1 Description

The Wumpus World is a classic example of a knowledge-based agent in AI. It involves reasoning, knowledge representation, and planning. This agent uses its world knowledge to make decisions and navigate safely in the given environment.

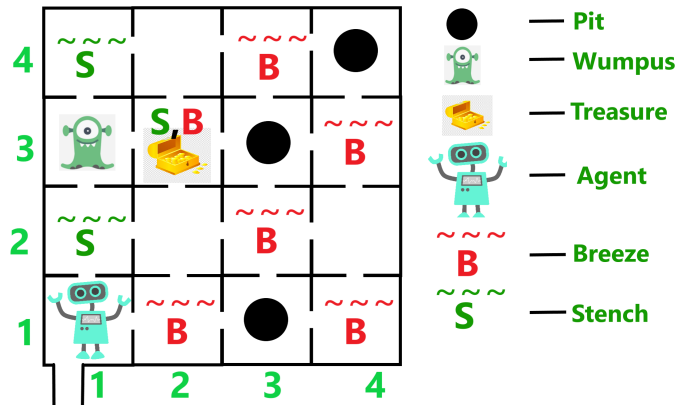


Figure 1: Example 4x4 Wumpus World map for illustration.

In this project, you are asked to build an intelligent agent for a Wumpus World environment as described below.

Environment

The environment is a static grid of size $N \times N$ (default $N=8$), with coordinates $(0,0)$ at the bottom-left corner. The X-axis increases to the right (East) and the Y-axis increases upward (North). Cells are bounded by walls; if the agent moves into a wall, it receives a **Bump** percept and remains in place. Each cell may contain:

- **Wumpus**: There are **K Wumpus** (default $K=2$) placed randomly. Wumpuses do not move, and their number K is known to the agent, but their positions are unknown at the start. If multiple Wumpus are aligned when shooting, only the first Wumpus hit is killed. When a Wumpus is killed, the environment emits a **Scream** and its stench disappears. A single room (cell) cannot contain more than one Wumpus.
- **Pit**: Appears in cells with a probability p (default $p=0.2$). Pits do not move, and the agent does not know the number or positions of pits.
- **Gold**: There is exactly **one gold**, placed randomly in a cell without pit or wumpus. Gold can appear at any cell, including $(0,0)$.

The agent starts at cell $(0,0)$ facing East (right). The starting cell is always safe (no pit or wumpus). It has one arrow to shoot in a straight line, which stops upon hitting a wall or killing the first wumpus encountered. The agent can perform the following actions:

- **Move Forward**

- **Turn Left / Turn Right**
- **Grab**
- **Shoot** Shooting (whether it hits or misses) costs -10 score.
- **Climb Out** (only at (0,0))

Note: The agent does not know the locations of gold, pits, or wumpus at the start; it must infer them through percepts as it explores.

Percepts

At each step, the agent perceives:

- Stench: if one or more wumpuses are in adjacent cells (north, south, east, west).
- Breeze: if one or more pits are in adjacent cells.
- Glitter: if gold is in the current cell.
- Bump: if it hits a wall when moving forward.
- Scream: if a wumpus is killed by its arrow.

Percepts for stench and breeze only indicate presence (true/false) but do not reveal the number or direction of wumpus or pits.

Objective

The agent's goal is to grab the gold and return to (0,0) to climb out safely, maximizing score. Killing the wumpus is optional and should only be done if it improves expected utility.

Action / Outcome	Score
Grab gold	+10
Move forward	-1
Turn left/right	-1
Shoot	-10
Die (fall in pit or eaten by wumpus)	-1000
Climb out (with gold)	+1000
Climb out (without gold)	0

2 Tasks

You are required to complete the following two main components:

A. Implementation

Develop a complete Wumpus World Agent System consisting of:

1. Environment Simulator:

- Randomly generate $N \times N$ maps with multiple wumpus (default $K=2$) and adjustable pit density (default $p=0.2$).
- Provide percepts (stench, breeze, glitter, bump, scream) to the agent based on its current state and action.
- Ensure visualization (console or graphical) showing:
 - Map state (known cells and agent knowledge)
 - Agent location, orientation, percepts, visited cells, and inferred safe/unsafe status of cells and action at each step

2. Inference Engine:

- Implement from scratch (no external logic inference libraries) using forward chaining or model checking.
- Determine whether each cell is safe, dangerous, or uncertain based on percepts and KB rules.

3. Planning Module:

- Implement A^* , Dijkstra, or another search algorithm to plan optimal paths considering cost, risk, and expected utility.

4. Hybrid Agent Integration:

- Integrate all modules (logic inference, planning) into a unified agent capable of achieving maximum score.

5. Random Agent Baseline:

- Implement a simple random agent for comparison in experiments.

6. Advanced Setting: Moving Wumpus Module:

In this advanced setting, the Wumpus is no longer stationary. Instead, each Wumpus will move once after every 5 actions performed by the agent.

Movement Rules:

- After the agent has performed 5 actions (of any type: move, turn, grab, shoot, climb), each Wumpus will:
 - Move to one of its valid adjacent cells (North, South, East, or West), chosen uniformly at random.
 - The Wumpus movement is not observable to the agent, only inferred from percept changes.
 - If the intended move would hit a wall, overlap with another Wumpus, or enter a pit, it stays in place.

Collision Rule:

- If a Wumpus moves into the agent's current cell during its movement phase, **the agent is immediately killed (eaten by the Wumpus)**.

Purpose: Because the Wumpus can move, the environment becomes **dynamic**. The agent's previous knowledge about Wumpus locations can become outdated after each movement phase. Therefore, the agent must:

- Continuously update its internal map based on new percepts.
- Act cautiously, as Wumpus positions change over time.

Requirements:

- Modify your environment simulator to:
 - Count the agent's total actions and trigger Wumpus movement after every 5 actions.
 - Implement Wumpus movement according to the rules above.
 - Detect and handle agent-Wumpus collision (agent death).
- Update your inference engine to: Maintain agent knowledge consistency with percept updates after each Wumpus movement phase.

This module challenges the agent to plan **risk-aware decisions** in a dynamic and partially observable environment.

B. Report

Prepare a comprehensive report (PDF) including:

1. Problem introduction and objectives
2. Formulation of Knowledge Base (KB) rules for breeze, stench, pit, wumpus, and gold using Propositional Logic, with explanation for multiple wumpus and multiple pits encoding.
3. System architecture and module design
4. Inference engine approach, including handling of moving Wumpus in the advanced setting.
5. Knowledge base update strategy to handle moving Wumpus percept changes.
6. Planning algorithm description
7. Experiment results:
 - Compare your agent with the random agent baseline in terms of success rate, average score, and decision efficiency.
8. Team member contributions (who did what)
9. An estimation of the completion level for each requirement (in %)
10. References and citations (if any)

3 Submission requirements

1. **Source code** (any language). Must be well-organized with clear instructions on how to run. Include a “testcases” folder with at least three different maps of different sizes, their input configurations, and output results (agent action log, final map state).

The source code must include a visualization module (console-based or graphical) that displays:

- Current map state (known cells and agent knowledge)
- Agent location and orientation
- Percepts and actions at each step

Your source code submission must include clear instructions on how to execute the program (e.g., a README file with installation requirements, commands to run, and expected outputs). Submissions without runnable instructions will be penalized.

2. **Video demo** (max 10 minutes) showing:

- Environment simulator
- Agent percepts and decisions
- Final outcome and performance analysis

3. **Report** (PDF) including:

- Problem introduction and objectives
- System architecture and module design
- Knowledge base encoding and inference explanation
- Planning algorithm
- Result of advanced setting
- Experiment results and evaluation
- Team member contributions (who did what)
- An estimation of the completion level for each requirement (in %)
- References and citations (if any)

4 Requirements and Grading

The project will be graded based on the following criteria:

No.	Criteria	Weight
1	Problem formulation and knowledge base design	10%
2.1	Environment simulator implementation	10%
2.2	Inference engine implementation	20%
2.3	Planning module implementation	10%
2.4	Hybrid agent integration	10%
2.5	Advanced moving Wumpus module	10%
3	Report quality and analysis	30%

5 Submission

Please follow the following submission guidelines:

- Your source code and report must be contributed in the form of a compressed file (.zip, .rar) and named according to the format StudentID1_StudentID2_....
- If the compressed file is larger than 25MB, upload it to Google Drive and share it via a link. **Absolutely no modifications are allowed after the deadline.**

Example details of the directory organization:

```
StudentID1_StudentID2_...
|-- Source
|   |-- (source code)
|   |-- README.txt (how to run source code)
|   |-- requirements.txt (libraries to be installed)
|-- Report.pdf (included demo video URLs)
```

6 Notices

Please pay attention to the following notices:

- This is a **GROUP** assignment. Each group has 3 - 4 members.
Groups with fewer or more members than the specified limit require lab instructor approval.
- Duration: about 3 weeks.
- All tools are not restricted; however, students should use them wisely. Lab instructors have the right to conduct additional oral interviews with random groups to assess their knowledge of the project.
- You are NOT allowed to use external logic solver libraries such as PySAT for the inference engine implementation. All inference modules must be implemented from scratch.
- Any submission found to have code similarity of 50% or higher with other groups or public projects on the Internet will be considered academic dishonesty and treated as plagiarism.
- Any form of plagiarism, dishonesty, or misconduct will result in a grade of zero for the course.