**UNIVERSITY OF SCIENCE, VNU-HCMC**

**FALCUTY INFORMATION TECHNOLOGY**

**CSC14003 - Introduction to Artificial Intelligence**

****

**Project 2  
Logical Agent**

**Instructors**: Nguyễn Trần Duy Minh, Nguyễn Ngọc Thảo,   
 Nguyễn Thanh Tình, Nguyễn Hải Đăng

**Class**: 22CLC06

**Group:** 06

**Members:**

22127022 – Võ Hoàng Anh

22127154 – Nguyễn Gia Huy

22127192 – Trần Gia Khiêm

22127210 – Phạm Anh Khôi

# Table of contents

[Table of contents 2](#_Toc174649910)

[1. Group information 3](#_Toc174649911)

[2. Work assignment 3](#_Toc174649912)

[3. Self-evaluation 7](#_Toc174649913)

[4. Algorithms 8](#_Toc174649914)

[4.1 General Algorithm 8](#_Toc174649915)

[4.2 Representing and Solving Propositional Logic with PySAT 9](#_Toc174649916)

[4.3 Knowledge Base 11](#_Toc174649917)

[4.4 Inference Engine 12](#_Toc174649918)

[4.5 Agent 13](#_Toc174649919)

[4.6 Environment 14](#_Toc174649920)

[5. Testing and Experiment 15](#_Toc174649921)

[5.1 Introduction 15](#_Toc174649922)

[5.2 Test Cases 15](#_Toc174649923)

[5.1 Test case 1: map1.txt 15](#_Toc174649924)

[5.2 Test case 2: map2.txt 16](#_Toc174649925)

[5.3 Test case 3: map3.txt 17](#_Toc174649926)

[5.4 Test case 4: map4.txt 18](#_Toc174649927)

[5.5 Test case 5: map5.txt 19](#_Toc174649928)

[5.3 Experiment Execution 20](#_Toc174649929)

[5.4 Results and Observations 21](#_Toc174649930)

[5.5 Reflections and Comments 22](#_Toc174649931)

[6. References 22](#_Toc174649932)

# 1. Group information

|  |  |  |  |
| --- | --- | --- | --- |
| **No.** | **Student ID** | **Full name** | **Gmail** |
| 1 | 22127022 | Võ Hoàng Anh | vhanh22@clc.fitus.edu.vn |
| 2 | 22127154 | Nguyễn Gia Huy | nghuy22@clc.fitus.edu.vn |
| 3 | 22127192 | Trần Gia Khiêm | tgkhiem22@clc.fitus.edu.vn |
| 4 | 22127210 | Phạm Anh Khôi | pakhoi22@clc.fitus.edu.vn |

Demo video: click vào [đây](https://youtu.be/Knj_VeQfNOY)

Google Drive storage: click vào [đây](https://drive.google.com/drive/folders/1FcUOhPCHu_RHFrXzH0oN1cIqpw0CUaow?usp=sharing)

# 2. Work assignment

The following table details the work assignments for each group member, specifying the tasks assigned to them and their respective completion rates. The work is divided into different aspects of the project, encompassing all four levels of complexity. The table ensures that all members contribute equally and cover every necessary component of the project.

|  |  |  |  |
| --- | --- | --- | --- |
| **Task** | **Description** | **Assigned Member** | **Completion Rate (%)** |
| Project Management | Coordination of tasks, meetings, and deadlines | Nguyễn Gia Huy | 100% |
| Environment Implementation | Develop the Environment class to build the map, process data, issue perceptions, and update the environment based on Agent actions | Võ Hoàng Anh | 100% |
| Agent Implementation | Develop the Agent class to receive environmental perceptions, process them, and take appropriate actions | Nguyễn Gia Huy | 100% |
| KnowledgeBase Implementation | Implement logic and algorithms to manage the agent's knowledge base and infer environmental conditions | Trần Gia Khiêm | 100% |
| InferenceEngine Implementation | Implement inference mechanisms to handle the agent's decision-making process and integrate it with the environment | Phạm Anh Khôi | 100% |
| Algorithm Documentation | Detailed write-up of the algorithms used in each level, including pseudocode and illustrative images | Nguyễn Gia Huy | 100% |
| Input/Output Handling | |  | | --- | |  |   Development of functions for reading input files and writing output files | Võ Hoàng Anh | 100% |
| GUI Development | Creating a graphical user interface to visualize the agent's exploration and interactions within the Wumpus World | Trần Gia Khiêm Phạm Anh Khôi | 100% |
| Testing | Creation of test cases and verification of algorithm correctness | Võ Hoàng Anh | 100% |
| Report Writing | Compilation and formatting of the final report | Nguyễn Gia Huy | 100% |
| Video Documentation | Recording and editing demonstration videos of the program in action | Nguyễn Gia Huy Phạm Anh Khôi Trần Gia Khiêm | 100% |

**Detailed Breakdown of Tasks:**

**1. Project Management:**

* Coordination of tasks, organizing meetings, and ensuring deadlines are met.
* Monitoring progress and ensuring all members are contributing equally.
* Overseeing the entire project lifecycle and addressing any issues promptly.

**2. Environment Implementation:**

* Develop the Program class to build the 10x10 grid map based on the input data.
* Implement functionality to process data, issue perceptions such as breeze, stench, and glow, and update the environment based on the agent’s actions.
* Ensure the environment reflects real-time changes as the agent interacts with it.

**3. Agent Implementation:**

* Develop the Agent class to interpret environmental perceptions (e.g., breeze, stench, whiff) and make decisions.
* Implement movement strategies (forward, turn left, turn right) and actions (grab, shoot, heal).
* Ensure the agent’s actions are based on the logic of the Wumpus World, including handling dynamic changes like Wumpus death and pit locations.

**4. KnowledgeBase Implementation:**

* Implement and extend algorithms to manage the agent’s knowledge base effectively.
* Ensure the knowledge base can handle the information from perceptions and update the map and decisions accordingly.
* Develop mechanisms to infer hidden details about the environment.

**5. InferenceEngine Implementation:**

* Implement inference mechanisms to handle complex decision-making processes for the agent.
* Integrate inference capabilities with the environment to enable the agent to navigate effectively and safely.
* Ensure the engine supports logical reasoning about environmental conditions and agent actions.

**6. Algorithm Documentation:**

* Write detailed descriptions of the algorithms used in the project.
* Include pseudocode and illustrative images to explain the logic and steps involved in the algorithms.
* Ensure clarity and comprehensiveness for ease of understanding and assessment.

**7. Input/Output Handling:**

* Develop functions to read from input files (e.g., map descriptions) and process the data to initialize the environment.
* Implement output functions to save results, including agent actions and final scores, in the specified format.
* Ensure robustness in handling various input cases and output formats.

**8. GUI Development:**

* Create a graphical interface to visualize the agent’s exploration of the Wumpus World.
* Implement real-time display of the agent’s path, percepts, and environmental changes.
* Design the interface to be user-friendly and informative, providing clear insights into the agent’s actions and environment.

**9. Testing:**

* Develop comprehensive test cases covering various scenarios, including edge cases.
* Run tests to verify the correctness, performance, and efficiency of the algorithms and agent behavior.
* Document test results and address any issues or bugs identified during testing.

**10. Report Writing:**

* Compile all sections of the report, ensuring proper formatting and organization.
* Include member information, work assignments, self-evaluation, descriptions of algorithms, and testing outcomes.
* Review and edit the report for clarity, completeness, and accuracy.

**11. Video Documentation:**

* Record videos demonstrating the program’s functionality for different test cases.
* Edit videos to highlight key features, successful runs, and notable aspects of the project.
* Upload videos and include URLs in the report to provide visual evidence of the program’s performance.

This revised work assignment ensures that each member has a clear role and responsibility, contributing effectively to the successful completion of the Wumpus World project.

# 3. Self-evaluation

|  |  |  |
| --- | --- | --- |
| **No.** | **Requirements** | **Completion Rate (%)** |
| 1 | Finish problem successfully. | 100% |
| 2 | Graphical demonstration of each step of the running process. You can demo in  console screen or use any other graphical library. | 100% |
| 3 | Generate at least 5 maps with difference structures such as position and number  of Pit, Gold and Wumpus. | 100% |
| 4 | Report your algorithm, experiment with some reflection or comments. | 100% |

# 4. Algorithms

## 4.1 General Algorithm

**Description:** The general algorithm outlines the primary flow of the agent's interaction with the environment in the Wumpus World. This algorithm forms the backbone of the project, guiding the agent through a series of steps from initialization to achieving its goal. The agent continuously perceives its environment, updates its knowledge base, chooses actions based on inferences, and interacts with the environment. This iterative process ensures the agent can navigate and make decisions in a partially observable and dynamic setting.

Main Flow of the Program:

A screen shot of a computer code

Description automatically generated

## 4.2 Representing and Solving Propositional Logic with PySAT

**Description:** This section describes how the problem is represented using propositional logic and solved using the PySAT library. Propositional logic is used to formalize the knowledge about the environment, with logical sentences representing the relationships and constraints of the Wumpus World. PySAT is a Python library that provides efficient SAT solvers for solving propositional logic problems. This combination allows the agent to infer new information from its knowledge base and make logical decisions.

**Propositional Logic Representation:**

* Each cell in the grid can have various properties (e.g., pit, Wumpus, breeze, stench, gold).
* Use propositional variables to represent the presence of these properties in each cell.

**Example:**

* P(x, y): There is a pit in cell (x, y)
* W(x, y): There is a Wumpus in cell (x, y)
* B(x, y): There is a breeze in cell (x, y)
* S(x, y): There is a stench in cell (x, y)

**Inference Rules:**

If there is a breeze in a cell, then there is a pit in an adjacent cell:

A white rectangular object with red and blue text

Description automatically generated

If there is a stench in a cell, then there is a Wumpus in an adjacent cell:

A white rectangular object with red text

Description automatically generated

Example code using***PySAT***:

A screenshot of a computer program

Description automatically generated

## 4.3 Knowledge Base

**Responsibilities:**

* Store and update the agent's knowledge about the environment.
* Manage propositional logic statements and use an inference engine to draw conclusions.

**Pseudocode:**

A screenshot of a computer program

Description automatically generated

**Description:** The KnowledgeBase class is responsible for maintaining and updating the agent's knowledge about the environment. It stores logical propositions derived from percepts and uses them to infer new information. This class interfaces with the SAT solver to handle the logical inference and ensure that the agent's decisions are based on the most up-to-date and accurate knowledge available.

## 4.4 Inference Engine

**Responsibilities:**

* Interface with the SAT solver to perform logical inference.
* Provide methods to query the knowledge base and derive new information.

**Pseudocode:**

A screen shot of a computer code

Description automatically generated

**Description:** The InferenceEngine class provides the mechanism for logical inference within the agent's architecture. It interacts with the KnowledgeBase to query and derive new information. This class enables the agent to make informed decisions by determining the truth values of propositions based on the current knowledge, ensuring the agent can infer safe paths and avoid hazards.

## 4.5 Agent

**Responsibilities:**

* Make decisions based on percepts and the current state of the knowledge base.
* Use the inference engine to determine safe actions and infer the presence of hazards.

**Pseudocode:**

A screen shot of a computer

Description automatically generated

**Description:** The Agent class encapsulates the behavior and decision-making capabilities of the logical agent. It uses the knowledge base and inference engine to choose actions based on percepts from the environment. The agent updates its knowledge with new percepts and makes decisions to navigate through the Wumpus World, aiming to find the gold and return safely.

## 4.6 Environment

**Responsibilities:**

* Maintain the state of the Wumpus World (positions of pits, Wumpus, gold, etc.).
* Provide percepts and elements to the agent based on its current position.
* Update the environment based on the agent's actions.

**Pseudocode:** **A screenshot of a computer program

Description automatically generated**

**Description:** The Environment class represents the Wumpus World in which the agent operates. It maintains the state of the world, including the positions of hazards and treasures. This class provides percepts and elements to the agent based on its current position and updates the world state in response to the agent's actions. It ensures the dynamic interaction between the agent and its environment.

# 5. Testing and Experiment

## 5.1 Introduction

Testing and experimentation are critical phases in the development and validation of the Wumpus World simulation. The goal is to ensure the logical agent functions correctly across a variety of scenarios, each with different configurations of pits, gold, and Wumpus. This section details the setup, execution, and results of experiments conducted on five different maps.

## 5.2 Test Cases

This section details the test cases created for each level of the project, along with the results obtained when running these test cases. Each level has different attributes to ensure comprehensive testing of the implemented algorithms.

### 5.1 Test case 1: map1.txt

A screenshot of a game

Description automatically generated

### 5.2 Test case 2: map2.txt

A screenshot of a game

Description automatically generated

### 5.3 Test case 3: map3.txt

A screenshot of a game

Description automatically generated

### 5.4 Test case 4: map4.txt

A screenshot of a game

Description automatically generated

### 5.5 Test case 5: map5.txt

A screenshot of a game

Description automatically generated

## 5.3 Experiment Execution

For each map, the logical agent was initialized in the starting position and tasked with navigating the environment to find the gold and return safely. The agent utilized its knowledge base and inference engine to make decisions based on percepts received from the environment. The main flow of execution for each test case is as follows:

A screenshot of a computer

Description automatically generated

**Table of experimental statistics for test-cases**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Test case** | **Start position** | **Gold** | **HP** | **Pit** | **Wumpus** | **Poisonous Gas** | **Actions** | **Score** |
| 1 | (5, 4) | 6 | 2 | 2 | 5 | 2 | 194 | 28060 |
| 2 | (0, 0) | 4 | 2 | 5 | 7 | 3 | 196 | 18050 |
| 3 | (0, 9) | 7 | 1 | 7 | 3 | 2 | 197 | 33040 |
| 4 | (9, 0) | 6 | 2 | 6 | 5 | 2 | 192 | 28080 |
| 5 | (9, 9) | 6 | 2 | 12 | 3 | 3 | 147 | 28530 |

## 5.4 Results and Observations

**Test Case 1:**

* **Path Taken:** The agent successfully navigated to the gold and returned to the starting position.
* **Score:** High, with no deaths or major obstacles encountered.
* **Comments:** The simple structure of the map allowed the agent to easily infer the safe path.

**Test Case 2:**

* **Path Taken:** The agent faced more complex paths but managed to find the gold and return.
* **Score:** Moderate, with some encounters with pits.
* **Comments:** The presence of multiple pits required careful reasoning to avoid hazards.

**Test Case 3:**

* **Path Taken:** The agent navigated the map efficiently, avoiding pits and the Wumpus.
* **Score:** High, with successful collection of health packs and gold.
* **Comments:** The addition of health packs provided an extra challenge but also a strategic advantage.

**Test Case 4:**

* **Path Taken:** The agent encountered multiple hazards but managed to find the gold and return.
* **Score:** Moderate, with a few close encounters with the Wumpus.
* **Comments:** The larger grid and increased number of hazards made this a challenging test.

**Test Case 5:**

* **Path Taken:** The agent demonstrated advanced reasoning and successfully navigated the complex map.
* **Score:** High, despite the dense distribution of pits and Wumpus.
* **Comments:** This test showcased the agent's ability to handle a high complexity environment.

## 5.5 Reflections and Comments

**Agent's Performance:** The agent's performance varied depending on the complexity and structure of the map. Simpler maps allowed for easier inference and quicker navigation, while more complex maps required sophisticated reasoning and handling of multiple hazards.

**Knowledge Base and Inference Engine:** The use of a knowledge base and inference engine was crucial for the agent's success. It enabled the agent to make informed decisions based on percepts and prior knowledge.

**Areas for Improvement:** Future iterations could improve the agent's handling of unexpected scenarios and optimize its pathfinding algorithm to reduce the number of steps taken.

**Overall success:** The logical agent successfully demonstrated its ability to navigate various environments, find gold, and avoid hazards, fulfilling the objectives of the Wumpus World.

# 6. References

1. *Boolean formula manipulation (pysat.formula) — PySAT 1.8.dev13 documentation*. (n.d.). <https://pysathq.github.io/docs/html/api/formula.html>

2. *SAT solvers’ API (pysat.solvers) — PySAT 1.8.dev13 documentation*. (n.d.). <https://pysathq.github.io/docs/html/api/solvers.html>

3. *knowledge-base for Wumpus World - Javatpoint*. (n.d.). www.javatpoint.com. <https://www.javatpoint.com/ai-knowledge-base-for-wumpus-world>

4. *The Wumpus world in Artificial Intelligence - Javatpoint*. (n.d.). www.javatpoint.com. <https://www.javatpoint.com/the-wumpus-world-in-artificial-intelligence>

5. *Pygame Front Page — pygame v2.6.0 documentation*. (n.d.). <https://www.pygame.org/docs/>