De La Salle University
Department of Software Technology

CSINTSY Major Course Output 2
Term 1, AY 2024–2025
Due before **25 Nov 2024 before 0800**
Submission via **AnimoSpace**

PROJECT SPECIFICATIONS

1 Project Description

In this project, students will apply the concepts and theories they have learned in the course to develop an artificial intelligence agent that assists a player in a puzzle game. Through this project, students should be able to demonstrate the following learning outcomes:

LO1. Design and evaluate informed search algorithms and knowledge representations for problem-solving

LO2. Collaboratively build systems that consider a number of paths or strategies in order to improve its performance in achieving its goal in less amount of computing time, or by some other metric of performance.

LO5. Articulate ideas and present results in correct technical written and oral English.

2 Project Specifications

2.1 Overview

In this project, you will design and implement an artificial intelligence algorithm for a bot. You must represent knowledge and rules about the world as a logic-based model, and apply the necessary techniques and algorithms to use this in assisting the player in completing his task in THE ADVENTURE WORLD.

2.2 The Adventure World

THE ADVENTURE WORLD is a simple world inspired by The Wumpus World described in Russell and Norvig (2020).

THE ADVENTURE WORLD is a $n \times n$ grid of cells. Some cells may have a Pit, which when the player falls into it, the player will not be able to climb out. Some cells have a **Gold coin**. The goal of the player is to collect at least 2 **Gold coin**s and return back Home and leave.

2.2.1 The World

- There is a cell called Home.
  - It is the starting position of the player.
  - It may be anywhere in the grid.
  - It does not contain a Pit.
  - It does not contain a **Gold coin**.
- There are $n - 2$ Pits scattered around the world.
  - The Pits are randomly placed.
  - There should be no 2 Pits that are directly adjacent to each other.
  - There are no Pits directly adjacent to the Home.
- There are $n - 1$ **Gold coin**s scattered around the world.
  - The **Gold coin**s are randomly placed.
  - There should be no 3 **Gold coin**s that are adjacent (vertically, or horizontally) to each other.

2.2.2 The Player

In the world, the player can either move, grab, or leave.

- The player can move left, move right, move up, and move down.
- The player can grab, to pick up the **Gold coin**.
- The player can leave, when on Home, to end the game.

2.3 Inference Engine

For this project, the students will use Prolog to implement the inference engine. PROLOG is a logic programming language that can do the following:

- represent a set of facts and rules in the knowledge base;
- query to determine whether a given formula can be derived from the knowledge base or not.

2.3.1 The Agent

The agent should be able to tell the player which cells are safe, which are unsafe (contain a Pit), or which are yet to be determined.

- When a player lands on a space, the agent may perceive: *breeze* and *glitter*. The knowledge base about the map is updated, and information about surrounding cells may be updated.
- In the cells directly adjacent to a Pit, the agent perceives a *breeze*.
- In the cells where the **Gold coin** is, the agent perceives a *glitter*.

2.4 The Application

For this project, you are to develop an application that allows the user (player) to navigate around The Adventure World, where $n = 5$. The agent tells the player which cells are safe, unsafe and unknown. An annotated map of the visited/known cells is displayed by the agent.

Assume that this is the map:

	Gold	*breeze*		Gold
Gold	*breeze*	Pit	*breeze*	
breeze	Pit	*breeze*	*breeze*	
	breeze	*breeze*	Pit	*breeze*
Home			Gold *breeze*	

Player's view. Annotated map based on knowledge of the agent.
(S=safe, B=Breeze, ?=unknown)

S				
Player	S			

Player's move: Up.

S				
Player S	S			
Home S	S			

Player's move: Right.

S	?			
S	**Player** S, B	?		
Home S	S			

The application ends with one of the following results:

- Mission accomplished! x coins collected.
- Mission failed! Only x coins collected.
- Mission failed! Player falls into a Pit.

The Inference Engine for the agent must be implemented in Prolog. You may opt to use Prolog, or other languages like Python, Java or C to implement the front end/interface to interpret and use the information queried from the Inference Engine.

2.5 Evaluation

You are expected to do an evaluation of your agent by testing it on different static maps (different positions for the Home, the Pits and the Gold coins). As you test the agent, focus only on whether it was able to provide an accurate description of the world as the player navigates. The application should terminate and correctly determine the results of the session.

Your agent should be able to handle $4 \leq n \leq 6$ maps. The maps may not be dynamically generated. You may create static maps, but your agent should be able to update and infer whether each cell is safe, unsafe, or unknown as the player navigates.

Your application works well on the map you prepared.	90%
Your inference engine is correct on a different/given map	10%

2.5.1 Performance Expectations (Grading)

3 Poster

In addition to the `SokoBot` codes, you are required to write a report documenting the implementation of the project. This report will be in the form of a one-page **poster**. You may visit this link for a guide about a poster presentations.

At the minimum the poster should contain the following:

Introduction A short but concise introduction describing the project.

The Inference Engine This section describes in detail the Inference Engine. Focus more on the project-specific details (e.g., how a knowledge is represented, how the rules are generated, how new formulas are added).

You may also use diagrams to show your strategies.

Evaluation and Performance A short introduction about how evaluation of performance is measures. This is followed by a description of the performance of the agent based on your own evaluation metrics.

This section must also highlight the strengths and weaknesses of the agent. Try to cite specific examples where the bot performs well / not well.

Challenges A discussion on the difficulties encountered when developing the application. What are the challenges encountered and how these were resolved?

Conclusion A conclusion section summarizing the process of implementing the project, including reflections, realizations, and insights obtained.

4 Deliverables

The deliverables for this project will be submitted via **AnimoSpace**. These should be submitted as listed below:

1. a zip file of the project directory, containing all files (source codes, etc.) needed to compile and run the project;
2. all source files written for this project;
3. a pdf file of the Poster;
4. a pdf file with a Table of Contributions showing the contributions of each member to the project.

5 Academic Honesty

Honesty policy applies. Please take note that you are NOT allowed to borrow and/or copy-and-paste in full or in part any existing related program code from the internet or other sources (such as printed materials like books, or source codes by other people that are not online). Violating this policy is a serious offense in De La Salle University and will result in a grade of 0.0 for the whole course.

Please remember that the point of this project is for all members to learn something and increase their appreciation of the concepts covered in class. Each member is expected to be able to explain the different aspects of their submitted work, whether part of their contributions or not. Failure to do this will be interpreted as a failure of the learning goals, and will result in a grade of 0 for that member.