RMIT University Vietnam

School of Science, Engineering & Technology

COSC2129/2723 - Artificial Intelligence

Assignment 1 Part 2: Multi-Agent

Due: **17:00 PM, August 07th, 2025, Week 6**

You must read fully and carefully the assignment specifications and instructions detailed in this file. You are NOT to modify this file in any way, except if instructed by the teaching staff in writing.

- **Course: COSC2129/2723** Artificial Intelligence, Semester 2, 2025
- **Deadline:** August 07th, 2025 @ 17:00pm (Week 6)
- **Course Weight:** 12%
- **Assignment type:** Individual
- **Submission method:** Upload zip file (see below for instructions)

The **aim of this project** is to get you acquainted with AI Multi-Agent search techniques. In this project, you will design agents for the classic version of Pacman, including ghosts. Along the way, you will implement both minimax and expectimax search and try your hand at evaluation function design.

The code base has not changed much from the previous assignment, but you should start with a fresh installation, rather than intermingling files from Assignment 1 part 1.

Your task

In this assessment, **your task** is to fully complete **5 Questions**, from **Q1 to Q5**, in the UCB's CS 188 Pacman Project 2: Multi-Agent Search (this link).

- You **must build and submit your solution** using the sample code we provide you with in **this Assignment**, which is **different** from the original UCB code base in the link above.
- **Files to Edit and Submit**: You will design agents for the classic version of Pacman and fill in portions of **multiAgents.py** for this assignment. You should submit **this**

file with your code and comments. Please do not change the other files in this distribution or submit any of our original files other than this file.

- Files you might want to look at:
  - *pacman.py*: The main file that runs Pacman games. This file also describes a Pacman GameState type, which you will use extensively in this project.
  - *game.py*: The logic behind how the Pacman world works. This file describes several supporting types like AgentState, Agent, Direction, and Grid.
  - *util.py*: Useful data structures for implementing search algorithms. You don't need to use these for this project but may find other functions defined here to be useful.

  Other supporting files you can ignore.

- Your code **must not contain any personal information**, like your student number or your name. If you use an IDE that inserts your name, student number, or username, you should disable that.

- You **must follow good SE practice** during your development; please refer to Marking criteria below.

- You are free to add additional testing scenarios under the `test_case/` folder.

Marking criteria

Question	Points
Q 1	2
Q 2	2
Q 3	2
Q 4	3
Q 5	3

The Autograder will be used to test your code for technical correctness (You can also use it to test your code). However, the correctness of your implementation - not the autograder's judgements - will be the final judgement of your score. The teaching staff reserves the right to run more tests, inspect your code manually, and **arrange a face-to-face meeting** for a discussion and demo of your solution when needed.

IMPORTANT: Submissions not compatible with the instructions above will attract zero marks and do not warrant a re-submission. We will not debug or fix your submission. Read carefully and ask for help (in forum or lab) if needed.

Important information

About this repo: You must ALWAYS keep your fork **private** and **never share it** with anybody in or outside the course, except your teammates (if it is a teamwork project), *even after the course is completed*. You are not allowed to make another repository copy outside the provided Classroom without the written permission of the teaching staff.

Please do not distribute or post solutions to any of the projects.
Corrections: From time to time, students or staff find errors (e.g., typos, unclear instructions, etc.) in the assignment specification. In that case, a corrected version of this file will be produced, announced, and distributed to you. Because of that, you are NOT to modify this file in any way to avoid conflicts.

Late submissions & extensions: A penalty of 10% of the maximum mark per day will apply to late assignments up to a maximum of five days, and 100% penalty thereafter (Extensions will only be permitted in *exceptional* circumstances under the University's rules).

Academic Dishonesty: This is an advanced course, so we expect full professionalism and ethical conduct. Plagiarism is a serious offense. Please **don't let us down and risk our trust**. Sophisticated *plagiarism detection* software via Codequiry may be used in this edition to check submitted code against other submissions in the class as well as resources available on the web. We will pursue the strongest consequences available according to the **University Academic Integrity policy**. In a nutshell, **never look at solution done by others**, either in (e.g., classmate) or outside (e.g., web, **AI tools**) the course: they have already done their learning, this is your opportunity!

Silent Policy: A silent policy will take effect **24 hours** before this assignment is due. This means that no question about this assignment will be answered, whether it is asked on the newsgroup, by email, or in person.

Code of Honour

We expect every RMIT student taking this course to adhere to the **Code of honour** under which every learner-student should:

- Submit their own original work.
- Do not share answers with others.
- Report suspected violations.
- Not engage in any other activities that will dishonestly improve their results or dishonestly improve or damage the results of others.

Submission

Submit your Assignment 1 Part 2 (Python files) in one zip file. File name is your student number, for example 1234567.zip if your student ID is s1234567. Do not submit unnecessary files.

Acknowledgements

This is Project 2: Multi-Agent Search from the set of UCB Pacman Projects. We are very grateful to UC Berkeley CS188 for developing and sharing their system for teaching and learning purposes.