COS30002 Artificial Intelligence for Games

Semester 1, 2023
Learning Summary Report

<Mai Tien Dat Tran> (<104207944>)

Declaration

I declare that this portfolio is my individual work. I have not copied from any other student's work or from any other source except where due acknowledgment is made explicitly in the text, nor has any part of this submission been written for me by another person or software service.

Signature: Dat

Self-Assessment Details

The following checklists provide an overview of my self-assessment for this unit.

	Pass (P)	Credit (C)	Distinction (D)	High Distinction (Low HD) | (High HD)	
Self-Assessment (please tick)	✔				

Self-assessment Statement

	Included? (tick)
Learning Summary Report	✔
Complete Pass (“core”) task work, approved in Canvas	✔

Minimum Pass Checklist

	Included? (tick)
Additional non-core task work (or equivalent) in a private repository and accessible to staff account.	
Spike Extension Report (for spike extensions) in Canvas	
Custom Project plan (for D and/or low HD), and/or High HD Research Plan document in Canvas (optional)	

Credit Checklist, in addition to Pass Checklist

	Included? (tick)
Custom Project Distinction Plan document, approved in Canvas	
All associated work (code, data etc.) available to staff (private repository), for non-trivial custom program(s) of own design	
Custom Project “D” level documents in Canvas, to document the program(s) (structure chart etc) including links to repository areas	

Distinction Checklist, in addition to Credit Checklist

	Included? (tick)
Custom Project “HD” level documents in Canvas, to document the program(s) (structure chart etc) including links to repository areas	

Low High Distinction Checklist, in addition to Distinction Checklist

	Included? (tick)
High Distinction Plan document, approved in Canvas	
High Distinction Report document, in Canvas, which includes links to repository assets	
All associated work (code, data etc.) available to staff (private repository) for your research work	

High High Distinction (Research) Checklist, in addition to D/Low HD Checklist

Introduction

This report summarises what I learnt in COS30002 AI for games. It includes a self-assessment against the criteria described in the unit outline, a justification of the pieces included, details of the coverage of the unit intended learning outcomes, and a reflection on my learning.

Overview of Pieces Included

01 - Lab - Repository Setup -- mark: completed: Show that you can build up the repository with all of my task submissions for the unit (code, report, etc.).

02 - Lab - FSM and Python -- mark: completed: Learn about the two most fundamental concepts in this unit: the finite state machine and the Python programming language.

03 - Lab - Tic-Tac-Toe -- mark: completed: Describe how to put together a basic TicTacToe simulation.
Learn about the loop structure of a game.

04 - Spike - Graphs, Search and Rules: -- mark: completed: Build efficient and/or successful TicTacToe bots by utilizing a Minimax Search-Tree algorithm.

05 - Lab - Graphs, Paths and Search -- mark: completed: Learn about the various kinds of search algorithms: Using a matrix, the Depth-First Search, Breadth-First Search, Dijkstra's, and A* algorithms simulation of an environment

06 - Spike - Navigation with Graphs -- mark: completed: To expand the current matrix environment simulation, implement thoroughly A*: More agents, a more complex map, and an environment.

07 - Lab - Goal Oriented Behaviour and SGI -- mark: completed: Describe the SGI and GOB and its limitations in resolving difficult problems.

08 - Spike - Goal-Oriented Action Planning (GOAP) -- mark: completed: Describe the GOAP and how it can use BFS to tackle the complicated problem that the GOB is unable to.

09 - Lab - PlanetWars -- mark: completed: Introduce the "Conquering" scenario, in which automated programs aim to outwit their rivals by directing their spacecraft to various locations on the map.
Use the bots to determine fleets to dispatch at random or only to win the game.

10 - Spike - Tactical Analysis with PlanetWars -- mark: completed: Create even more intricate bots for the current scenario, then assess how well each one does in battles against the other or the basic or random bots.

11 - Lab - Steering #1 - Seek, Arrive, Flee -- mark: completed: Learn about the steering simulation, in which an agent or agents may be able to search for, reach a location slowly, normally, or quickly, or escape from a assigned role

12 - Lab - Steering #2 - Wander and Paths: -- mark: completed:

Utilize the seeking, arriving, and fleeing features of the current steering simulation to implement:
searching for every vertex in a polyline that is created at random
glancing through the map

13 - Spike - Tactical Steering (Hiding): -- mark: completed:Utilize the searching, arriving, escaping, and wandering features of the current steering simulation to execute the following:

• Prey is an agent looking for a secure obstacle to hide in;
• Hunter is an agent pursuing the Prey;
• Obstacles that agents should avoid

14 - Spike - Emergent Group Behaviour -- mark: completed: Apply several group behaviors using the current steering simulation's searching, arriving, fleeing, and roaming features:

• Alignment: Agents move in the same direction;
• Separation: Agents tend to run away from one another;
• Cohesion: Agents tend to meet at one spot after a while.

15 - Spike - Agent Marksmanship -- mark: completed: Utilize the seeking, arriving, and fleeing features of the current steering simulation to implement:
• Prey that is able to transition between them

specified goals
• A stationary hunter with the ability to shoot prey by calculating its orbit

16 - Spike - Solider On Patrol -- mark: completed: Utilize the searching, arriving, escaping, and wandering features of the current steering simulation to execute the following:

• Preys are basic agents that simply wander the map at a leisurely pace.
• Hunters have two options: they can shoot the prey in various ways (determined by the FSM) or travel a random path with constantly changing ways of seeking and arriving.

22 - Doc - Learning Summary Report -- submitted: Provide a brief summary and demonstration of everything you learned in this unit, including the results I achieved on each portfolio activity.

My github:

https://github.com/tranmaitiendat/COS30002-104207944

Coverage of the Intended Learning Outcomes

This section outlines how the pieces I have included demonstrate the depth of my understanding in relation to each of the unit's intended learning outcomes.

ILO 1: Software Development for Game AI

Task 2: FSM & Python: The built Finite State Machine demonstrates the practical use of fundamental game development principles by explicitly defining states and transitions.
Tasks 3, 4: Graphs, Rules, and Tic Tac Toe: The AI code that is modular and the obvious game

The Tic-Tac-Toe loop design reveals depth by showing the relationships and effects that different software elements have on the behavior of the AI.

Task 6: Using Graphs for Navigation This well-organized codebase makes it simple to add or remove agents involved in the searching process, which is a perfect match with the extensibility principle of software development.

Tasks 9–10: Tactical Analysis and PlanetWars: The object-oriented codebase permits the extension in terms of various bots.
Jobs11,12,13,14,15, 16 - Steering: These tasks contain a world system that enables me to efficiently manage a variety of agents and objects of different kinds (prey, hunters, missiles, and obstacles).

ILO 2: Graphs and Path Planning

Task 4: Rules, Graphs, and Search The sophisticated bots developed for the game TicTacToe make use of Minimax's basic DFS-based search algorithm and its optimization, Alpha and beta prunning, to generate rudimentary graph traversal that may be both successful and efficient.

Tasks 5, 6 - Graphs, Paths, Search & Navigation: It demonstrated its ability to perceive and navigate a boxed environment using search techniques including Dijkstra's, A*, BFS, and DFS. This shows a basic knowledge of graph and path design techniques to account for varying terrain costs.

Task 8: GOAP: In this task, the problem that can be solved with just GOB and SGI is solved using the BFS algorithm as a GOAP solution.

Task 13: Tactical Steering: The prey uses strategic distance calculations to determine how far away it is from the hunter, itself, and any obstacles that can be hidden. It then plans the best hiding strategy.

ILO 3: Force-based Agent Movement

Task 11: Seek, Arrive, Flee: The agent(s) may use one of three steering modes to accomplish this task: escaping from a target, arriving at a target, or attempting to reach a target.
Task 12: Wander & Paths: The agent has two more steering options at its disposal: locating a route,

looking around the map
Task 13: Tactical Steering: The hunter can pursue the prey while they are in close proximity to one another (this feature was not incorporated into a distinct approach).
Task 14: Emergent Group Behavior: Put cohesion, separation, and alignment into practice at the size of a group of agents.
Tasks 15, 16: Make minor adjustments to the current functionality of seeking, arriving, fleeing, locating a path, and meandering in order to make it compatible with the new simulations.

ILO 4: Goals and Planning Actions

Task 4: Graphs, Search, and Rules: Develop a proficient bot that utilizes the Minimax algorithm to determine its next optimal move.

Task 10: PlanetWars Tactical Analysis: The bot TacticalAnalysis can determine its

the number of ships on both conquered and unconquered planets, as well as the distances between these planets, in order to determine which mode to use in its turn: "attack" or "defend"

Task 7 - GOB & SGI: Implement a GOB solution by simply selecting an action to solve a basic cost-based problem o Task 8 - GOAP: Implement a GOAP solution by using Breadth-First Search, solving the complex cost-based problem that GOB cannot solve

Task 13 - Tactical Steering: The Hunter determines the distances between other objects in the game and then creates its plan of hiding

ILO 5: Combine AI Techniques

Task 4-Graphs, Search, and Rules: Using the recusion technique, a Minimax bot with alpha-beta prunning is created to determine which move(s) will win the game of TicTacToe.

Tasks 5, 6: Graphs, Paths, Search, and Navigation: These tasks use various AI search techniques. algorithms that determine the path in the matrix environment, such as DFS, BFS, Dijkstra's, and A*. In task 6, the prey seeks to locate the green objective to win the game, and the hunters use A* to locate the preys.

Task 16: Soldier on Patrol: In order to decide which hunter mode should be active, this task makes use of an FSM. Additionally, it rotates the projectiles' acceleration using the 2D Rotation Vector Formula, enabling the hunter to shoot "spreadly," "massively spreadly," and "all-hit."

Reflection

The most important things I leant:

I gained a thorough understanding of artificial intelligence in games throughout the course. Among the most significant lessons were the use of goal-oriented action planning, the application of force-based agent movement, the integration of graph-based path planning approaches, and the production of games using the min-max algorithm. I also discovered how crucial it is to use these AI techniques in order to create games that are more engaging and dynamic. This educational experience surpassed my expectations and provided me with a thorough understanding of the applications and potential of artificial intelligence in game development.

The things that helped me most were:

The lab tasks in particular taught me a lot. Their practical examples and challenges helped me better understand the concepts and use the AI techniques. I was also able to increase my understanding of the material and get any uncertainty I was having cleared up via group discussions and teacher input.

I found the following topics particularly challenging:

Goal-oriented action planning (GOAP) and the integration of multiple AI methodologies were challenging, but once I understood and used them right, they also made me feel really accomplished.

I found the following topics particularly interesting:

I was particularly interested in how force-based agent movement was used in video games. Thanks to your material, I now have a new perspective on how to create realistic, dynamic characters for game settings.

I feel I learnt these topics, concepts, and/or tools really well:

I gained in-depth knowledge of force-based agent movement, graph-based path planning, goal-oriented action planning, and software development approaches for artificial intelligence in games. My understanding has improved as a result of applying these techniques to numerous lab tasks.

I still need to work on the following areas:

I need to improve my ability to combine different AI techniques in order to make games that are more intricate. Though I believe I have a fundamental understanding today, there's always room for improvement.

My progress in this unit was …:

Despite working and studying overseas for the final two weeks of the semester, I believe I improved consistently throughout this unit. My learning and final grade were greatly impacted by my consistent ability to turn in assignments on time and communicate with my tutor.

This unit will help me in the future:

The skills and knowledge I have acquired in this unit will come in very handy for my upcoming schoolwork and career aspirations. My grasp of AI techniques and how they relate to the creation of games will provide me a competitive edge in the growing gaming industry.

If I did this unit again I would do the following things differently:

Having recently balanced my schooling with an internship as an AI developer, I now understand the importance of effective time management and task prioritization. I would utilize these skills to arrange special projects and research for higher marks, as well as to begin lab work earlier and give myself more time to work through the complexities of the code.

Conclusion

In summary, I believe that I have clearly demonstrate that my portfolio is sufficient to be awarded a Pass grade.