COS30002 Artificial Intelligence for Games

Semester 1, 2023
Learning Summary Report

<HOANG BAO PHUC CHAU> (<103523966>)

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: Hoang Bao Phuc Chau

Self-Assessment Details

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

	Pass	Credit	Distinction	High Distinction	
	(P)	(C)	(D)	(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

This section outlines the pieces that I have included in my portfolio…

In this section, I want to include the following pieces:

- 01.P - Lab - Bitbucket Setup
- 02.P - Lab - FSM & Python
- 03.P - Lab - Tic-Tac-Toe
- 04.P - Spike - Graphs, Search & Rules
- 05.P - Lab - Graphs, Paths & Search
- 06.P - Spike - Navigation with Graphs
- 07.P - Lab - Goal Oriented Behaviour & SGI
- 08.P - Spike - Goal-Oriented Action Planning (GOAP)
- 09.P - Lab – PlanetWars
- 10.P - Spike - Tactical Analysis with PlanetWars
- 11.P - Lab - Steering 1- Seek, Arrive, Flee
- 12.P - Lab - Steering 2 - Wander and Paths
- 13.P - Spike - Tactical Steering (Hide!)
- 14.P - Spike - Emergent Group Behaviour
- 15.P - Spike - Agent Marksmanship
- 16.P - Spike - Soldier on Patrol

I want to include these pieces because I want to get Pass grade for this unit.

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

"Discuss and implement software development techniques to support the creation of AI behaviour in games"

- 01.P - Lab - Bitbucket Setup
  - In this task, I learnt to open a GitHub repository, commit, pull and push my code for maintenance.
- 02.P - Lab - FSM & Python
  - In this task, I first learnt about using Python. It was different from C++, which I used regularly, as it has shorter syntax and easier to use. Also, I learnt about how to implement FSM
- 03.P - Lab - Tic-Tac-Toe
  - In this task, I learnt how to implement basic AI (which was random AI) to play tic-tac-toe, which was not very intelligent and the structure of a basic game using Python.

ILO 2: Graphs and Path Planning

"Understand and utilise a variety of graph and path planning techniques."

- 04.P - Spike - Graphs, Search & Rules
  - In this task, I learnt about minimax algorithms for tic-tac-toe and how to improve the random search algorithm apart from the basic one. Minimax algorithms really analysed the graph and make good moves to take the winner position.
- 05.P - Lab - Graphs, Paths & Search

  - In this task. I learnt about 4 different search algorithms to analysed the graph and search for the best path to the destination, which are DFS, BFS, Dijkstra and SearchAStar. All of them have different logic compare to each other. However, to me, Dijkstra is the most favourite one, This search method is used for finding the shortest paths between nodes in a graph. Compared to BFS, it accounts for weighted paths, allowing it to determine the shortest path by distance, not simply the fewest edges.
- 06.P - Spike - Navigation with Graphs
  - In this task, I learnt how to create a navigation graph for a game, and use AI to analyse it, then find the best path. I created 4 agents moving with different speed and using different algorithms. This task gave me a deeper understanding and more confident of implementing Search Algorithms.

ILO 3: Force-based Agent Movement

"Create realistic movement for agents using steering force models."

- 11.P - Lab - Steering 1- Seek, Arrive, Flee
  - In this task, I drew an UML diagram to see which class contains which methods and properties, finish the flee function and be able to add additional agents. Also in this task, I saw how force affected the moving speed of the agent, also how the program updated and rendered.
- 12.P - Lab - Steering 2 - Wander and Paths
  - In this task, I analysed the path.py file to see what inside the Path class. Then, going inside the Agent class, I modified it so that it have some more properties and methods such as randomise_path(), path, wayponint_threshold, etc. Also, wander() move is also added using the guidelines in the Lab documentation. Both things are very important in the next assignments.
- 13.P - Spike - Tactical Steering (Hide!)
  - In this task, the goal is to create 2 agents, a prey and a hunter, and the prey must be able to hide from the hunter. Comparing to task 12, it has several differences. Firstly, a class named Obstacle has been created inside obstacle.py file and there are 3 types of circles has been created, also there is a function to check if the circle is safe. Inside Agent class, I had to create 2 classes which were inherited from Agent class, namely Prey and Hunter. However, there are some differences such as Prey utilise seek movement for hiding while Hunter use wander to look for Prey.
- 14.P - Spike - Emergent Group Behaviour
  - In this task, the implementation is somewhat simple, the only important thing to edit is the calculation() method of the Agent class. I successfully implemented Cohesion, Allignment and Separation for the Agent.

ILO 4: Goals and Planning Actions

"Create agents that are capable of planning actions in order to achieve goals."

- 07.P - Lab - Goal Oriented Behaviour & SGI
  - In this task, I first time learnt about GOB and SGI. I fixed the code inside the sample for it to work effectively. I also saw how stats and goals are related to each other and how they could be change rapidly.
- 08.P - Spike - Goal-Oriented Action Planning (GOAP)
  - For this task, using the knowledge learnt from past assignments, I upgraded the task 7 to a new level by making it more complex, divided it into different classes for easy maintenance, which are Graph and Node. Graph holds the data of goals and actions, while Node will be the one to represent all the states in the node and manage all the interactions between them. Also, I did use BFS search method to search for the best path.
- 09.P - Lab – PlanetWars
  - For this task, I learnt about the PlanetWars game, which illustrates GOB and SGI, the target is capture all of the planets and destroy the enemy. I successfully implemented the BestWorst bot for improving the logic and play the game more effectively.
- 10.P - Spike - Tactical Analysis with PlanetWars
  - For this task, I successfully implemented a new bot for analysing the game to get the best tactic (it also use GOB and SGI) The bot has the ability to think if it needs to defend or attack

based on the situation of the battle. After this task, I know GOB and SGI better and really love this game. (In the future I would try to make it better)

ILO 5: Combine AI Techniques

"Combine AI techniques to create more advanced game AI."

- 15.P - Spike - Agent Marksmanship
    - This task is challenging to me as it requires me to make a lot of math calculations for the projectile to work effectively. I used resources from the internet and ask the AI to get an idea how to implement it. Finally, the implementation of the firing mechanism is successful, with different stats for each type of ammo (however still follow the rule as the document required)
- 16.P - Spike - Soldier on Patrol
    - This task is the combination of all the Force-based movement task and FSM control. At the first place, it gave me some challenge as I have to figure the condition for FSM to effectively work. I managed to find out the condition based on the target destroyed, target left and ammo. However, implementation of burst mode and automatic mode took a lot of my time due to using wrong method for time delay

Reflection

The most important things I leant:

The most important I learnt is the search method such as BFS, DFS, Dijkstra, A Star, and Force Based movement, which is very important when I want to create a shooting game which heavily require those thing.

Also, I learnt the way to use pyglet to create a game, as all of the past time I only use C++

The things that helped me most were:

The things that helped me the most were lecture videos on Canvas, pyglet documentation, youtube and stackOverflow.

For example, pyglet documentation helped me to find the correct method to use, stackOverflow helped me to identify erros in my code, youtube helped me very much when implementing firing mechanism, etc.

I found the following topics particularly challenging:

Force-based movement would be the one who gave me then most challenge, due to the fact that it required me to study math, improve logical thinking. Also, I've never done such those things before, so it really gave me an initial impression.

I found the following topics particularly interesting:

To me, the Graphs and Path Planning is very interesting as it taught me about search methods, which I think it would be very important for me in the future.

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

I think I learnt Graph and Path Planning really well. For example, I implemented minimax algorithms for tic tac toe and also improved the random AI for playing better. I also implemented search Dijkstra for task 6 and create my own map to use with it.

I still need to work on the following areas:

Since force-based movement is the one gives me the most challenge, I still need to improve myself more and work on this to get to now it better.

My progress in this unit was …:

My progress in this unit was not very good and consistent. I did not consistently submit my work due to bad time management. However, I did engage with my tutor to get feedback on my work. Also, I think my progress affected my final grade quite a lot because Pass grade was not what I aimed for at the start of the semester.

This unit will help me in the future:

This unit will help me to implement AI to my own game in the future since I got some basic knowledge about common algorithms and the way to implement them used in various tasks.

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

If I did this unit again, I would spend more time doing tasks and complete them on time. Also, I would try to get at least a C grade for the unit instead of P.

Other…:

Conclusion

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