Spike Extension Report

Nguyen Khanh Toan - 104180605
COS30002 - AI for Games

Introduction

This Spike Extension Report serves as a detailed exploration of tasks undertaken to advance proficiency in AI for Games, in alignment with specific Instructional Learning Objectives (ILOs). Throughout this report, I delve into various extensions, each strategically chosen to improve understanding and implementation of key AI techniques within gaming contexts. The report not only outlines the tasks completed but also underscores their relevance to the specified ILOs. By implementing TicTacToe AI agents, Graph-based search algorithms, Goal-Oriented Action Planning (GOAP), tactical analysis in PlanetWars, tactical steering, and agent marksmanship, I aim to fulfill the outlined objectives comprehensively. Each task is to illustrate its contribution towards enhancing software development techniques for AI creation, utilizing graph and path planning, creating realistic agent movement, planning actions to achieve goals, and combining AI techniques to create more advanced game AI. Through this report, I aim to demonstrate a deeper understanding of AI integration within game development, aligning with the goal of skills in this dynamic field.

ILO's

1. Discuss and implement software development techniques to support the creation of AI behaviour in games
2 Understand and utilise a variety of graph and path planning techniques.
3 Create realistic movement for agents using steering force models.
4 Create agents that are capable of planning actions in order to achieve goals.
5 Combine AI techniques to create more advanced game AI

Spike Extension Alignment:

Spike Extension	ILO 1	ILO 2	ILO 3	ILO 4	ILO5
Lab 3 - TicTacToe	X	X			
Spike 4 - Graphs, Search and Rules		X		X	
Spike 8 - Goal-Oriented Action Planning (GOAP)	X			X	
Spike 6 - Navigation with Graphs	X	X			
Spike 10 - Tactical Analysis with PlanetWars	X			X	X
Spike 13 - Tactical Steering (Hide!)			X		X
Spike 15 – Agent Markmanship			X		X
Multi-Agent Custom project	X			X	X

Justification for Each Extension:

Extension 1: Lab 3 – TicTacToe

- *In this lab, I have implemented two AI agents.*
- *I have created a function to call which AI agent user want to play, or call both of the AI to compete with each other, or just simply a normal Tic Tac Toe game with two players.*
- *By utilise the OOP design, AI agent can now be easier to implemented and added into the program within the get_move() function.*

- *These help me to achieve ULO 1, which is implemented software development technique to support the AI creation in game.*

Extension 2: Spike 4 – Graphs, Search and Rules

- *In this Spike, I have implemented Alpha-Beta Pruning AI agent for Tic Tac Toe.*
- *This agent is more advance than Minimax agent, by cutting depth of a un-utilise move (further explanation in Spike 4 Report) to get more efficiency.*
- *This extension helps me to achieve ULO 2 and ULO 4. I can see how different states are represented as nodes and how transitions between states. Then searching the incoming states then decide to choose the best move with lower cost. This is fundamental and crucial for AI in specific and data structure and pattern in general.*

Extension 3: Spike 6 – Navigation with Graphs

- *In this Spike, I have created 4 types of agents in the Agent class with different cost.*
- *The Hunter agent will automatically change its target to the initial of the Prey and will try to catch the Prey Agent by its current box (Node) whenever user change planning plan by switch agent. Further detail in Spike 6 Report.*
- *This extension helps me to achieve ULO 1, ULO 2.*
    - *ULO 1: By implementing each individual agent in a different class (Agent class), this would help the program much easier to create a different AI agent in the future.*
    - *ULO 2: While working in this spike, I can understand what components it needs to complete the graph planning tree, to recreate a different agent that utilising the search algorithm.*

Extension 4: Spike 8 – Goal-Oriented Action Planning (GOAP)

- *In this Spike, I have implemented the GOAP agent in OOP and implement utility technique to always choose the best action that solve the solve the issue of SGI in the previous lab 7.*
- *The agent now can choose the best action with long term result by its calculating best utility (further details in the Spike 8 Report).*
- *This extension gets me to acquire the ULO 1, and ULO 4.*
    - *ULO 1: The utility AI agent implemented in OOP, which mean, it can utilise action with goal along with the structure of the program and can support further implementation of others AI agent, such as Rule-based Agent, …*
    - *ULO 4: This task extension has gain me understanding to make AI planning ahead instead of only choose what is best for current circumstance. This raise my awareness of creating an AI that can perform better in long term.*

Extension 5: Spike 10 – Tactical Analysis with PlanetWars

- In this Spike, I have create a new bot that use strategies to compete with the naive bot created in previous lab. The new bot, Disciplarmy has strategies to scouting, defending and attacking the planet scouted.
- Futher explanation in the Spike 10 Report.
- This extension gain me more understanding of developing an agent that working in an environment (UOL 1) and find out the solution to get the agent more rational about the game environment with optimise decisions to achieve goal (UOL 4). In addition, this extension task also gain me how to combine multiple AI technique (ULO 5) into a single

agent, for example, the agent only attack when the planet recognize enemy planet through scouting behavior, and defense the align planets.

Extension 6: Spike 13 – Tactical Steering (Hide!)

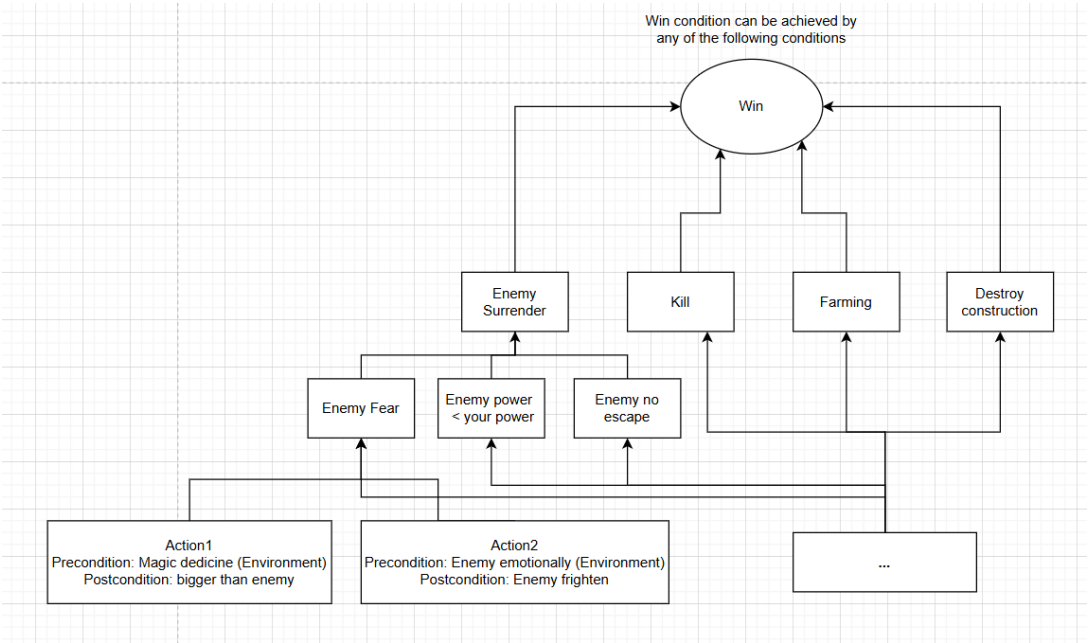
- In this Spike, I create a hunter that only chase prey when the prey is in the recognition range, and also prey has their own panic range. If the prey panic, it will hide behind the hiding object. And also while chasing, agent automatically flee from the hunter. And if nothing happen, prey will seek for the target(star)
- Further implementation in Spike 13 Report and Code.
- This spike task gain me further understanding how to applying multiple behavior into an agent to make it more complex and realistic.(ULO 3 & ULO 5)

Extension 7: Spike 15 – Agent Markmanship

- In this Spike, I have implemented a projectile acceleration calculation base on physics stimulation and I also solve that equation.
- Further explanation in the Spike 15 report and code.
- In this spike, I gain the understand of how to implement the agent with realistic movement into an AI game environment (ULO 3). And further more, the implementation not only about understanding the math/physics, but also how to translate it into programming language to make it happened in game.

Extension 8: Custom project on Multi-Agents (Spike 17-Folder in repository)

- In this project, I combined Goal-Oriented Action Planning (GOAP) with truth tables to create a system where AI agents can achieve long-term goals. The agents use a knowledge base (KB) that updates with changes in the environment.
- A truth table checks if the KB supports the goal. If it does, the GOAP agents start planning and taking actions to achieve the goal.
- This way, agents can adapt to changes and make informed decisions based on the updated knowledge.



Conclusion

In this Spike Extension Report, I have tackled a range of tasks aimed at enhancing my skills in AI for Games, aligning with specific Instructional Learning Objectives (ILOs). Through implementing TicTacToe AI agents, Graph-based search algorithms, Goal-Oriented Action Planning (GOAP), tactical analysis in PlanetWars, tactical steering, and agent marksmanship, I have dived deep into AI behavior in gaming environments. In addition, I also create a program that combine multiple AI agents to drive AI to the long term plan. These tasks have not only enhanced my understanding of software development techniques for AI creation but also support my proficiency in graph and path planning, realistic agent movement, and strategic decision-making within games. The alignment of each task with the designated ILOs underscores the comprehensive coverage of essential AI concepts essential for crafting sophisticated game AI. This report stands as my growth and proficiency in AI for Games, fulfilling the overarching objective of honing my game development skills through AI integration.