

Report

Contents

1	Introduction	2
2	Output Summary	3
3	Literature Review	4

Chapter 1

Introduction

Chapter 2

Output Summary

Chapter 3

Literature Review

My project focuses on creating a game in the souls-like genre where the enemies act in an eco-system where there is a food-chain hierarchy in which they can fight each other.

An intricate in-fighting system between multiple different AI agents is what I will be focusing on. One of the early examples of in-fighting was seen in the original Doom (1993)[1], where if an enemy-A got attacked by enemy-B, enemy-A will change it's target from player to enemy-B, this is a very simple system but offers many different gameplay opportunities, many levels were crafted with this in mind where the players would have to manipulate the AI by luring one enemy into another's line of fire. This idea was later expanded by Ubisoft in their later Far Cry games where different clans can combat each other in random events, and animals can also combat each other, combining these together, there are many random events where animals attack enemies that are already mid combat between you and another clan. In Far Cry Primal they expanded the animal system so that you could tame any animal in the world and make them your companion in gameplay.

This sort of systemic game design is created by making the different game systems aware of each other, and this in turn invites the player to use their creativity to make plans on how to execute their goals and creating unique experiences. This type of gameplay has been coined as emergent gameplay where complex systems *emerge* from the interaction on relatively simple game mechanics.[4]

There are different types of AI techniques that I could use, the following are the ones that I have looked into: Finite State Machines, Behaviour Trees, Goal Oriented Action Planning and Utility Based AI.

Finite state machines are the most rudimentary AI system, it is a system that I had implemented in my Advanced Games Tech Project, these are very easy and quick to implement for a simple AI system however it does not suit the complex AI behaviours I would like to build as when adding more complexity to a state machine, the "program flow becomes much more difficult to understand

and creates a debugging nightmare"[5].

Behaviour Trees were popularised by Halo 2[6] and are now the most common AI technique used in modern games, this is due to there being a streamlined flow/logic to the readability of the AI, this makes it easier to expand while keeping debugging simple. The design is built using nodes as modules, allowing nodes to be reused with minimal effort. There is also a 'blackboard' where shared knowledge is kept which the AI uses to make 'smarter' decisions while keeping memory usage to a minimum. This all happens while keeping a low computational overhead if implemented correctly. [7]

Goal Oriented Action Planning (GOAP) was originally implemented in a game in F.E.A.R (First Encounter Assault Recon) by Jeff Orkin [8]. It is a system that is made up of many actions, these actions comprise of objects, preconditions and effects; an agent will decide on an action to execute based on their current situation and then work backwards through the preconditions to create an efficient plan of executing the actions. This is done using an A* path finding algorithm, the actions have different weightings based on how easy they are to execute from the current state of the agent. This system can be very easy to scale if the ground work is set up properly as all would have to be done is add more actions in. Similar to behaviour trees (BT), a blackboard is used to store shared data which the AI can use to make decisions. This system can have performance issues, as goals and plans will have to be calculated for every active agent very frequently, so increasing the number of agents in the scene will have a performance hit. This was an issue when F.E.A.R came out as the rats in the game used the same AI system as the enemies, and once a rat had spawn, it would not be destroyed, and this would accumulate over a long play session and cause performance issues. This system can create novel AI behaviour which isn't seen easily in BT and FSM AI design. This is the system I would like to implement as it would be easier to create AI that seems to behave 'organically' like they do in F.E.A.R. [9]

References

- [1] id Software, id Software, (1993). DOOM. [computer game]. Available at: https://store.bethesda.net/store/bethesda/en_IE/pd/productID.5361563100/currency.GBP
- [2] Ubisoft.com. 2021. Far Cry franchise. [online] Available at: <https://www.ubisoft.com/en-gb/franchise/far-cry/> [Accessed 9 February 2021].
- [3] Thompson, T., 2018. Primal Instinct | Companion AI in Far Cry Primal. [online] Gamasutra.com. Available at: https://gamasutra.com/blogs/TommyThompson/20180906/325967/Primal_Instinct__Companion_AI_in_Far_Cry_Primal.php [Accessed 12 February 2021].
- [4] "Le Gameplay emergent (in French)". jeuxvideo.com. 2006-01-19. Available at: <https://www.jeuxvideo.com/dossiers/00006203/le-gameplay-emergent.htm> [Accessed 9th February 2021]
- [5] Buckland, M., 2010. Programming Game AI by Example. Burlington: Jones & Bartlett Learning, LLC, p.46.
- [6] Bungie, Microsoft Game Studios. Halo 2 [computer game]. 04/09/2004.
- [7] Simpson, C., 2014. Behavior trees for AI: How they work. [online] Gamasutra.com. Available at: https://www.gamasutra.com/blogs/ChrisSimpson/20140717/221339/Behavior_trees_for_AI_How_they_work.php [Accessed 12 February 2021].
- [8] Orkin, J., n.d. Goal-Oriented Action Planning (GOAP). [online] Alumni.media.mit.edu. Available at: <http://alumni.media.mit.edu/~jorkin/goap.html> [Accessed 12 February 2021].
- [9] Thompson, T., 2020. Building the AI of F.E.A.R. with Goal Oriented Action Planning. [online] Gamasutra.com. Available at: https://www.gamasutra.com/blogs/TommyThompson/20200507/362417/Building_the_AI_of_FEAR_with_Goal_Oriented_Action_Planning.php [Accessed 13 February 2021].