

Computer Games Development

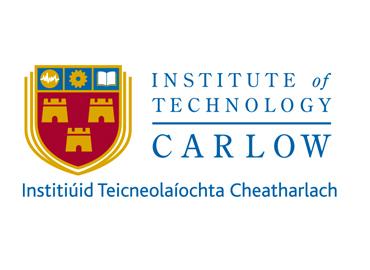
Project Report

Year IV

Brandon Seah-Dempsey

C00204076

[03/05/2020]



**Faculty of Computing and Networking Science**

**Open-Book and Remote Assessment Cover Page**

**Student Name: Brandon Seah-Dempsey**

**Student Number: C00204076**

**Lecturer Name: Lei Shi**

**Philip Bourke**

**Module: Project (CW 208)**

**Stage/Year: 4**

**Date: 03/05/2019**

**Declaration**

This examination/assessment will be submitted using Turnitin as the online submission tool. By submitting my examination/assessment to Turnitin, I am declaring that this examination/assessment is my own work. I understand that I may be required to orally defend any of my answers, to the lecturer, at a given time after the examination/assessment has been completed, as outlined in the student regulations.

**Contents**

[Project Abstract](#_heading=h.30j0zll) **4**

[Project Introduction and Research Question](#_heading=h.1fob9te) **4**

[Background](#_heading=h.3znysh7) **5**

[Literature Review](#_heading=h.2et92p0) **5**

[**Study**](#_heading=h.sajgxpr0vr23) **8**

[Project Description](#_heading=h.3dy6vkm) **9**

[**Project Milestones**](#_heading=h.lp4kpj4x5wsk) **11**

[Results and Discussion](#_heading=h.4d34og8) **12**

[Project Review and Conclusions](#_heading=h.2s8eyo1) **13**

[References](#_heading=h.17dp8vu) **14**

# 

# Project Abstract

AI has become an integral part of video games whether it’s having enemy entities act out different behaviours towards a player’s actions or having NPCs act out a routine to make them seem like a part of the game’s world; AI has become one of the key components in behaviour and entity interactions. Video games will use AI for many different kinds of entities with varying degrees of complexity; as such this project will try to incorporate an AI in a turn-based RPG in Unity to find it’s effectiveness in improving the difficulty and level of fun the AI will have on video games.

This project starts by investigating into different AI through various websites, documents, articles and online videos. Through my investigations, I’ve found different articles and sites discussing how AIs in video games function, the types of patterns and techniques used and some of the inner workings of their functionality such as adaptability. My research led me to find that certain games have different behaviours unlocking when certain conditions are met to achieve a form of adaptability, such as enemy searching nears locker if a player keeps hiding in lockers. My research had also led me to find ways of creating my own different behaviour patterns, such as a Behaviour Tree and Decision Tree. I then researched creating a turn-based battle system similar to Final Fantasy and found many suitable examples online, until I found an easy enough one to use and recreate myself on Unity.

As a result a video game was created using my research and investigations, allowing the creation of my chosen AI patterns and my desired battle system.

# Project Introduction and Research Question

Within video games, AI is an important component toward their development; whether it’s to make adaptive enemies and difficulty in a video game, or make NPCs seem like a real part of the world. AI is needed to ensure there is interactivity and life needed in a video game in order for any form of gameplay to exist.

The purpose of this project is to find how effective an AI model is in affecting the game in turn based styled RPG gameplay, similar to Final Fantasy; answering the question: “Can a Decision making AI be implemented into a turn-based RPG?”. This question will be answered through the use of AI models such as a Behaviour Tree and Decision Tree, and seeing the effectiveness of these models on the game’s enemies.

The potential impact of this will be to compare the different models and see their effectiveness in the situation there being put through. This study will see when is the better model in terms of gameplay enhancement and difficulty while still maintaining an enjoyable and challenging experience for its user.

# Background

The purpose of this project is to research the effectiveness of a decision making AI pattern implemented into the AI of Enemy characters in a turn-based RPG. When researching I first had to look up suitable AI patterns, such as a *Behaviour Tree* and *Decision Tree*, and find games that utilise this pattern to exemplify its effectiveness and my reason for using it, to back this research up with the appropriate documents and videos. A notable example was a video and online article detailing the functionality of the Alien’s behaviour tree from *Alien Isolation*; from it’s different states of behaviour, to its use of a Macro and Micro manager system for its tree, to the use of a “menace gauge” to affect and drive it’s passive behaviours. I would then research and look into existing Unity and C# documents, forums and videos with helping in creating and implementing both a turn-based RPG and the AI pattern I wish to use. After completing all of my research, I would then apply everything I’ve learned into creating a Unity project to test out the different AI patterns and then apply them to the Enemy and Player.

# Literature Review

AI has always been an essential part of video games; whether it is used for the behaviours of the different entities present in the game; used to perform certain techniques, such as pathfinding; or used in mechanisms not visible to the user, such as data mining. Ever since its first inception in the 1950s, the uses of AI have expanded greatly with developers of video games pushing to create games with more human-like AIs capable of adaptability. In this project, I will attempt to create an AI of my own using these techniques; when appropriate; for a Turn-Based RPG for Enemies and later Players; and will make certain decisions based on the available data like a form of adapting, i.e don’t use a Fireball spell anymore after learning that the Enemy is immune to Fire damage.In order to research into this topic I will have to look into various different documents and articles discussing the use and incorporation of AI in video games; but also use examples used in video games as well.

In terms of adaptive learning, my people have utilised different forms of it; whether through the use of an AI or using data of the Player to change the game based on it. A form of this is Dynamic game difficulty balancing (DGDB), which can be dated back to as early as 1975. In 1986 Compile developed *Zanac* (see Fig.1), the game that featured a unique adaptive AI that automatically adjusted the game’s difficulty according to the Player’s data, such as skill level, rate of fire and ship’s defensive status. This is done by measuring the aggressiveness and and game’s difficulty depending on the actions of the player, i.e data, attack pattern and skill level; and increases it for experts or decreases it for beginners. An instance of this would be if the player shoots with the main cannon frequently, collects power-ups, and fails to destroy bosses within the specified time limit, the AI will increase the number of tougher enemies appearing; or losing lives, starting a new level, or destroying reconnaissance planes will result in less enemies appearing.



Fig.1 - Zanac

A more modern example of this would be the AI of the Xenomorph in *Alien: Isolation* (see Fig.2) developed by Creative Assembly in 2014. When developing this AI the development team tried and tested AI techniques, designed in a way that breaks traditional game-design concepts. The Xenomorph’s AI uses a complex set of behavioural designs that slowly unlocks as it encounters the player; giving the illusion that the Xenomorph learns from each encounter and adjusts it’s hunting strategy. To do this the AI uses a two tier behaviour manager system, the *Macro*, director-AI, and the *Micro*, alien-AI.

The director-AI is used to observe the player throughout the entire game, always knowing where they are and maintaining a metric known as the ‘menace gauge’. The director’s job is to give a hint to the alien-AI about the player’s location periodically; while not actually telling the alien where the player is and allowing it to figure it out itself. As mentioned before the director maintains a metric called the ‘menace gauge’; similar to in approach to stress gauge from *Left 4 Dead* for prioritising enemy attacks; ‘menace gauge’ is used as a way of telling the alien to invoke menace by having it go to the player’s general location. The ‘menace gauge’ will increase depending on the player’s proximity to the alien; once the ‘menace gauge’ reaches its peak level the director will retreat and send the alien to neighbouring rooms or into the air vents. The purpose of this is to make the game scarier by having the alien closer to the player, but not for too long to allow them to achieve progress. This process is managed by a Utility AI or job system; dicating tasks to complete; i.e alien visits location and it’s priority of whether to finish what it’s doing, move towards doing this action or interrupt everything and execute right now! This job system also allows the alien to operate in two state active, where the alien sweeps area or specific locations due to a certain trigger, i.e noise; or passive, where ‘menace gauge’ is at its peak and the alien retreats to the vents, while moving ahead to where it thinks the player is going to be.

The alien-AI, however, is completely different as it’s behaviours are reliant on behavioural trees. This behaviour tree system contains over 100 different nodes, with around 30 top level nodes responsible for selecting the type of behaviour, it is about to execute and use and execute within large sub-sections of the tree that are responsible for sub-behaviours relating to specific tasks of the alien. What’s to note is that certain parts of the tree are locked off and gradually unlock when certain conditions are met depending on the player’s actions or points of the game’s campaign. This ensures that the alien will exhibit new behaviours the longer the player is playing the game, giving the impression that the alien is learning from experience based on the player’s behaviour and keeping up with them; to ensure fairness however, none of these conditions are met upon the player’s death. For example if the player keeps hitting the alien with the flamethrower the alien will either dodge back or to the sides to avoid it and rush the player or will ram into the player if they are set on fire; another example would be if the player keeps hiding in lockers or under desks the alien start searching more in the area of said objects. The alien will rely on a fast and efficient pathfinding when given specific jobs from the director, with heuristics that help dictate values of the alien’s sensors. These sensors allow the alien to identify and detect the player for a variety of different ways from the player’s footsteps to gunshots and even the player’s motion tracker when it is in close proximity to the alien. What should be noted is the alien also has a sensor behind it, acting as eyes on the back of it’s head to prevent the player from walking behind it to avoid detection. Yet despite all this, while the alien moves to a specific location due to the director it doesn't do so in an optimal fashion; as it is searching and hunting for the player. This is done by giving the alien areas of interest to explore, hand-tuned or dynamically generated, acting as areas the alien wants to visit or because the player has made too much noise. Using a perimeter of nearby locations, the alien will devise a path to visit them sub-optimally; prioritised based visibility of the alien; resulting in backtracking giving the illusions that the alien is double-checking. Locations for the alien’s movement are separated into search locations: those that the alien will visit, and spot locations: where it will stay where it currently is standing and then turn to look at the area of interest.



Fig.2 - Alien: Isolation

In conclusion, many will use different forms of adaptive learning in AIs, whether it is learning from the data values generated by the player or adapting based on the player’s behaviour and actions. And through implementing many will follow through traditional means of implementation, while others will break tradition but still produce incredible results with outstanding AIs.

# Study

During my investigations and studies into AI models, I had found different websites, articles and videos on topics and discussing the uses of the different models used, how they’re integrated into video games and how they function in general. Through these investigations lead me to try and integrate a Behaviour Tree and Decision Tree models into the AI of my project. In order to achieve this I first had to research forums and tutorial videos on how to create a turn based battle system in Unity and how to implement both a Behaviour and a Decision Tree, all in C#.

My research had allowed me to find an online video tutorial on building a turn-based battle system similar to Final Fantasy and later found articles on creating the AI models that I intended to use. To start of, I first created basic sprites and GUI to allow an easy to follow visual representation of my project; then following the tutorial I first created Base classes for both the Player and Enemy of the game to hold all of the data values used in the game, such as the max health, current health, speed, strength, ect. I then created three battle state managers, one for the Players and Enemy and one to manage both managers as well as handle both the turn handling and GUI interactions with the user. Afterward I added basic input into the project’s GUI and proceeded to make a start on creating the Behaviour Tree and Decision Tree. Following the forums and article guides I found, I created a Behaviour Tree that makes use of inheritance to create Action Nodes that will content game logic and run these actions through a sequence until all these action are fulfilled, return true, and will have a selector to decide which sequence runs based on certain conditions. A Decision Tree, that also uses inheritance to set up a tree(class) contenting different methods to represent different decisions and have them be selected based on certain conditions. Finally I had set up a Finite State Machine(FSM), which through the use of Enum states transitions between different states based on certain conditions, i.e Idle to Walking, Walking to Running, ect.

# Project Description

The finished product of my project should look similar to a battle scenario of Final Fantasy 6, with the Enemy of the left side, the Players on the right and the UI on the bottom of the screen. The gameplay will be similar to a turn-based battle of Final Fantasy 6 in which the Player and Enemy will have separate turns from each other, in other words they won’t be sharing the same attack queue. The UI will display the name, HP, MP and turn bar of all the Players; and will have panels to hold all the buttons used by the User, one for normal basic attacks and one for magic. When the attack button is pressed it will end that Player’s turn, deal damage to the Enemy and move onto the next one, if the magic button is pressed a new button panel is create to hold buttons for the Players magic attacks, should these buttons be pressed that same thing as the attack being pressed will occur. The Enemy will attack the Player after a set amount of time and will have three states to determine its decision making; Random, DT and BT. Random is self-explanatory, the Enemy will select a random Player and attack to use on the Player DT is where the Enemy will use a Decision Tree to decide which target and attack to use; and BT is similar to DT, but instead a Behaviour Tree is used instead. The Player and Enemy make use of a State Machine to determine the different state of their turns, with different methods running in each state.

A close up of a sign

Description automatically generated

In the end my project met all that I had desired except for the Player’s turn bar. The purpose of the turn bar is to have it’s scale X set to 0 and increase until it is at 1, once it is at the it would then be that Player’s turn. Once the User has selected one of the Player’s buttons the bar is set 0 and is increased again. If two Players have their bars at 1, the first one attacks first and the second once waits in the attack queue until the first Player has attacked. I was not able to achieve this because if a Player attacked before the other Players turn bar reached one, then the other turn bats stopped increasing and all the Players would be unable to attack. As well as that, the turn bar failed to increase in scale when it was set back to 0, making this unachievable at certain points. To resolve this I instead compromised and had the Players turn order be decided based on who’s bar reached 1 first, in other words if the turn bars reach 1 in this order: Thief, Mage and Warrior; then this was the entire turn order of all the Players. While this wasn’t exactly like Final Fantasy 6, this could still at least establish the order of Player attacks effectively in my project.

For my technical achievements, I have gained a better understanding of decision making AI structures on how they are established and how they should be properly built; a better understanding of how they are implemented and integrated into video games and how they affect the overall game; and how the functionality of these AI models should work accordingly. For my personal achievements, I was able to successfully create a turn-based battle system with a decision making aI model incorporated into it, something I had hope to possible before even starting my project and seeing the effectiveness of project, baring some results that I was hoping for and showing me what was needed to properly finalise this idea.

# 

# 

# Project Milestones

AI project’s milestones:

1. Simple game build with Game loop, update and render in Unity - (5 November 2019)

* General set up of the Game.
* Create entities and UI sprites to be used in the game.

1. Game character classes and GUI - (15 November 2019)

* Create Base Classes that hold all data used for Players and Enemy.
* Create UI elements and prefabs to be used.

1. Implement turn handling - (5 October 2019)

* Create a class that holds important variables used by both Player and Enemy, i.e target and chosen attack.
* Create State machine classes to handle the turn orders and actions of both the Player and Enemy.
* Create a Battle Manager class to manage and run both state machines during gameplay and handle the User’s input.

1. Basic user input - (7 December 2019)

* Create AddListener events to UI elements with appropriate test responses.
* Allows test responses to progress the Player’s turn and move onto next.

1. Test AI implementation - (22 December 2019)

* Create separate scenes and prefabs to test different AI models.
* Create tests for Decision Trees(DT), Behaviour Trees(BT) and Hierarchical Finite States Machines(HFSM).
* Tests should include using data to make decisions(DT), responses to inputs or events(HFSM), and changing values and checking if they reach a certain value(BT).

1. Final test AI implement- (23 January 2020)

* Finalise AI tests and save tests as scenes and prefabs.

1. Implement Enemy and Player interaction- (1 March 2020)

* Create Attacks and Abilities classes to be used to create interactions between the Player and Enemy.
* Through the use of the State machines and Battle manager, implement Enemy targeting Players and choosing attacks to interact with them by affecting their health.
* Add the same for the Players targeting the Enemy, but instead ensure attack choosing is done through the game’s UI.
* Then begin integration of AI decision making models.

1. Finalise GUI - (8 April 2020)

* Finalise by creating different button panels and button prefabs to but used by the Players.
* Create methods to adjust the active states of the panels and buttons, and take information from the Players to be applied to the buttons.
* Finalise the AddListener events to ensure Player interaction is done with Enemy through button presses, as well as turn progression.

1. Finalise Game - (29 April 2020)

* Make final changes, adjustments and additions to the project for final presentation and demo.

# Results and Discussion

The end results of my project were, while having a few problems, what I was hoping for in helping me find out the effectiveness of incorporating a decision making AI into a turn-based battle system.

Through my research and testing I was able to find and incorporate a Decision Tree and Behaviour Tree into the NPC enemy of a turn-based battle system and test the effectiveness of these decision making models in-game. From my tests I have found that models such as these can effectively be utilised in allowing the Enemy to decide on which target to attack and which attack to use on their target with the data given to the Tree. However, I found that due to the amount of data I used the decision making can be a bit finicky at times, for example sometimes it may not change it’s decisions of attack target. The main reason for this is due to giving each target and attack something similar to a weight value, with decisions being made based on the highest weight values of the target and attack.

A picture containing map

Description automatically generated

To avert this problem, I found that more data and more logic to my Trees decision making would be needed. Not only that, but fixing my Trees to make them more tree-like in structure to be more like the model they represent to improve their decision making options, giving them branching paths rather than looping then together. As well an expansion of the amount of data used by the Trees in their decision making.

A screenshot of a cell phone

Description automatically generated

In conclusion; while I was able to successfully incorporate a decision making AI model into an Enemy NPC, the results ranged from what I desired to finicky to due to lack of data or structure of models.

# Project Review and Conclusions

In the end; while it is possible to implement a decision making AI into a Turn-based battle system, the results depend entirely on the approach, implementation and data used in the decision making of the AI. With regards to my very own AI; while it was able to make decisions I wished for it to do there, there were times when the decision making would be a bit finicky and sometimes not make the intended decisions, mainly due to a lack of data or not having enough data to work with. If I were to start this project again, I would look more into incorporating more decision making models, such as a Rule Based System, create more data for the decision making to utilise and improve upon the AIs trees to make a more tree-like structure in their decision making. If someone wished to pursue a similar project, I would advise them to plan out what exactly they desire from their decision making AI and to plan out the decision it will make to get an understanding for well structured decisions with viable use of data. And with regards to my choice in technology I felt that the use of Unity was the right choice in my opinion, as I felt it Unity’s ease of use, easy to understand supporting software and variety it offers made development of my project easier to undertake and handle compared to using SFML or other development tools I am not familiar with.

In conclusion; it is possible to implement an effective decision making AI with learning capabilities, the next step in these approach should be to improve upon the current models used as well as their structures, while adding new one to increase the number of models that can be used, and to improve and widen the range of data these model should use in their decision making.

# References

|  |  |  |
| --- | --- | --- |
| **Referenced Publication** | **Citation** | **Reference** |
| Book | (David Gibson, Clark Aldrich and Marc Prensky 2007) | David Gibson, Clark Aldrich and Marc Prensky (2007). *Games and Simulations in Online Learning: Research and Development.* USA: Information Science Publishing. |
| Report | *(Integrated Machine Learning For Behavior Modeling in Video Games N/A*) | Ben Geisler (N/A). *Integrated Machine Learning For Behavior Modeling in Video Games.* (Radical Entertainment). |
| Website | (Tommy Thompson 2017)  (Natasha Mathur 2017)  (Code4Noobz 2019) | Tommy Thompson (April 4 2017). *The Perfect Organism — The AI of Alien: Isolation.* [Online]. (URL) <https://becominghuman.ai/the-perfect-organism-d350c05d8960> . (Accessed 8 November 2019).  Natasha Mathur (October 11 2018). *Building your own Basic Behavior tree in Unity (Tutorial).* [Online]. (URL) <https://hub.packtpub.com/building-your-own-basic-behavior-tree-tutorial/> . (Accessed 19 November 2019).  Code4Noobz (2019). *C# – Decision Tree – Simple Example.* [Online]. (URL) <https://code.4noobz.net/decision-tree-simple-example/>[/](https://hub.packtpub.com/building-your-own-basic-behavior-tree-tutorial/). (Accessed 20 December 2019). |

# 