

Computer Games Development CW208

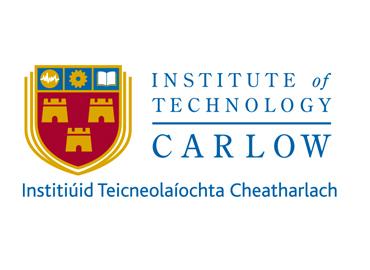
GDD and Project Report

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.4w8mmxazvssd)4

[**Project Introduction**](#_heading=h.k38kme690eh4)4

[**Background**](#_heading=h.ee84zki41ke6)5

[**Project Description**](#_heading=h.24ejcgazcgvs)5

[Game Overview](#_heading=h.tyjcwt) 6

[**Features**](#_heading=h.p2fclgfw190t)8

[Feature #1 - Entity Stats](#_heading=h.g6fgorqr6nxl) 8

[Feature #2 - Turn Handling](#_heading=h.xx8aqlngm5dc) 8

[Feature #3 - Special Attacks](#_heading=h.gvq525psx5yw) 8

[Feature #4 - UI](#_heading=h.pdijicf1g8cm) 8

[Feature #5 - Turn-Based Battle System](#_heading=h.bctocyxyai0y) 8

[Feature #6 - Decision Tree](#_heading=h.kb225y2be61b) 9

[Feature #7 - Behaviour Tree](#_heading=h.ye5n2tkev530) 9

[Feature #8 - Adjustable AI](#_heading=h.oasoiydebpv3) 9

[**Project Milestones**](#_heading=h.cynsv52l3bfo)9

[**Project Review and Conclusions**](#_heading=h.wkufb0i4wwv4)10

[**References**](#_heading=h.rl19o4q3xydp)11

[Book](#_heading=h.m2q4k91sf6i3) 11

[Report](#_heading=h.ra4tpig5mdm4) 11

[Web-site](#_heading=h.sln8a07mxsho) 11

# 

# 

# 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

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 an AI be implemented into a turn-based RPG for Enemies?”. 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 compared 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 to 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 look up various different documents, discussion and videos outlining the use and functionality of the different type of decision making AI models. I then looked into detail the way they are integrated and used within video games and see the overall impact on the games itself. Eventually I found suitable one to utilised and begin to look into them in considerable detail, such as a *Behaviour Tree* and *Decision Tree*, and found 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.

# 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.

# Game Overview

This game is a turn-based style RPG battle game, similar to the battle sequences of Final Fantasy 6. And just like it, it will consist of the Enemy being on the left side of the screen and the Players on the right, with the UI at the bottom of the screen. The right side of theUI will be used to control the Players during their turn, with the User pressing the button on the UI to determine the Players actions towards the Enemy, i.e attack will be a basic attack to the Enemy, magic will be to open the magic attack panel to select special attacks instead. On the left side the Players stats will be displayed, such as their name, HP, MP and turn bar; and will be updated if the Enemy attacks or the Players attack the Enemy. This game is an attempt to try and apply a decision making AI into the Enemy of a turn-based battle system, such as a Decision Tree and Behaviour Tree.

Before starting this project, I had been interested in seeing if turn-based battle systems could be improved upon through the use of an AI. The reason for this is because from my experience in playing these types of games, they tend to rely a lot on RNG or have event triggers based on the Enemy’s current HP, leaving the entire flow of these games left to chance and having its Users hoping for the best. Fundamentally what I wish to achieve with this is to test and see the effectiveness of the Enemy when it can make decisions depending on the data given to it’s decision making AI, without a reliance on RNG or event triggers as a form of difficulty. And to create a challenging experience through decision making AI rather than randomness, in hopes of improving turn-based RPGs.

While story and setting weren’t a major focus on my project, my idea for the character designs relied on shapes. With the Players consisting of squares and rectangles and the Enemy consisting of triangles. The reason for this was an idea that came to me in the form of a world where everything is made of shapes; from plants to animals and even people; with the idea being that each faction or nation in this world is represented by shapes. And just like in real life, the different nations have different and conflict views and as a result being at odds with each other or going to war. With the battle in-game taking place in the form of the conflict between these two nations, with the triangles being the tyrannical nation trying to conquer the world and the squares being the first in their conquest .

The User controls a party of three squares fighting against a triangle; the squares are essentially a rectangle block with two rectangle eyes and legs, all coloured black with a white outline. While the Enemy is a large triangle, with upside-down triangles as eyes and legs; coloured the same as the Players. The controls are very simple, the Users use the mouse to click on the buttons on the games UI to decide the Players actions. My reason for choosing this control scheme was to give the User an easy to understand control scheme without the need of a tutorial by simply having them push a button.

In this game, the party of squares goal is to defeat the leader of a group of triangles and stop them from trying to conquer their home; while the triangle’s goal is to defeat the squares line of resistance and lead the charge to conquer the squares nation. And in order to do so they must fight each other in a turn-based battle and fight until either the triangle’s HP hits 0 or all of the squares in the party’s HP hits 0.

Compared to games that try to attempt this, not many have actually attempted this approach as a result there's not many games that can be compared to my game. However, when compared to other turn-based RPGs; my approach is definitely suited towards making smart and difficult enemies, but compared to the bosses of these games the use of event triggers give a sense of increasing and added difficulty to them compared to my approach where the difficulty would be the same from the start of battle.

# Features

## Feature #1 - Entity Stats

This feature will allow all existing game entities to have adjustable stats that will affect how they will perform during gameplay of the game. This will include their maximum health, speed, strength ect.

## Feature #2 - Turn Handling

This feature will handle the turn order of the Player and Enemy, controlling when the Player and Enemy attacks, in what order will the Players attack in and handle what attacks or actions they will perform during their turn and how that will affect the other Players or Enemy.

## Feature #3 - Special Attacks

This feature will allow the creation and use of special attacks or abilities the Player and Enemy can use; each with their own unique values such as damage value, mana cost and different damage types.

## Feature #4 - UI

This feature allows for the User to give their input to the Players through a User Interface, these inputs will allow for the User to decide how the Players will respond and act towards the Enemy during battle.

## Feature #5 - Turn-Based Battle System

This feature, with the help of the Turn Handle feature, will allow the simulation of a turn based battle system similar to Final Fantasy, in which the Players will have their own turn order to decide their interaction to the Enemy while the Enemy has their own separate turn order to Pressure the Players to act quickly in order to defeat them

## 

## Feature #6 - Decision Tree

This feature will allow the Enemy to decide which Player to attack and which attack is the most effective to use based on various different data, such as how much health the Player has, the Enemy’s mana or if the Player can heal allies; through the use of a Decision Tree to control the Enemy’s decision making.

## Feature #7 - Behaviour Tree

This feature will allow the Enemy to decide which Player to attack and which attack is the most effective to use based on various different data, but instead the Enemy’s decision making is done through a Behaviour Tree.

## Feature #8 - Adjustable AI

This feature will allow for easy adjustment of the Enemy’s AI and decision making before gameplay, to allow the user to test out and play against the different form of AI used by the Enemy.

# 

# 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 them 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.

# Project Review and Conclusions

In conclusion; 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.

# 

# References

## Book

[1] David Gibson, Clark Aldrich and Marc Prensky (2007). *Games and Simulations in Online Learning: Research and Development.* USA: Information Science Publishing.

## Report

[2] Ben Geisler (N/A). *Integrated Machine Learning For Behavior Modeling in Video Games.* (Radical Entertainment).

## Web-site

[3] 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).

[4] 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).

[5] 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).