

Computer Games Development CW208

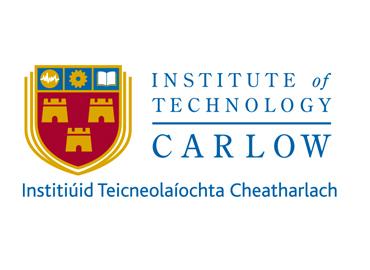
Technical Design Document

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

[Game Architecture](#_heading=h.gjdgxs) **3**

[**UML**](#_heading=h.db6caa2tqshs) **5**

[Class Diagram: Game](#_heading=h.wibqloclvjrd) 5

[**Features**](#_heading=h.n3lwtboxstyf) **6**

[Feature #1 - Entity Stats](#_heading=h.g6fgorqr6nxl) 6

[Feature #2 - Turn Handling](#_heading=h.xx8aqlngm5dc) 6

[Feature #3 - Special Attacks](#_heading=h.gvq525psx5yw) 6

[Feature #4 - UI](#_heading=h.pdijicf1g8cm) 6

[Feature #8 - Adjustable stats](#_heading=h.psdhrjqhy9ue) 7

[**CRC Cards**](#_heading=h.m62eprdyp9oy) **8**

[Class Name : Player](#_heading=h.8gfuc5tk929a) 8

[Class Name : Enemy](#_heading=h.feccv1uir2ld) 8

[Class Name : Battle Manager](#_heading=h.hkub0tqo28qi) 9

[Class Name : Turn Handle](#_heading=h.8vmyaiahxh6n) 9

[Class Name : Behaviour Tree](#_heading=h.396lzatgy284) 10

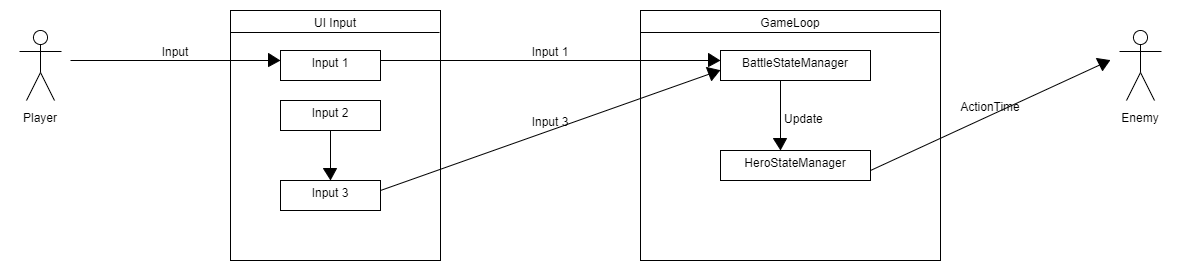
[Class Name : Decision Tree](#_heading=h.3i85y5ftcm2q) 10

[Class Name : Attacks](#_heading=h.ktxj5mdw5eyz) 11

[Class Name : Tree Data](#_heading=h.10ocnvtrqa5h) 11

[**References**](#_heading=h.1wg955vlrfx0) **12**

# **Game Architecture**



# **UML**

## ***Class Diagram:*** Game

# 

# **Features**

## Feature #1 - Entity Stats

Tasks:

1. Create a Player and Enemy Base Class scripts that will hold all data and variables used by both for gameplay.
2. Make classes public to ensure variables are accessible across all other scripts.
3. Attach scripts to game objects and set their values.

## Feature #2 - Turn Handling

Tasks:

1. Create a Turn Handle Class script that holds the data needed by both the Player and Enemy, such as their attack target or attack to use on their target.
2. Create State Machines for both the Player and Enemy that will handle different states of their turn, i.e Enemy has a ChooseAction state to decide its next course of action and an Action state to fulfil that action.
3. Create appropriate methods to be used in each of these states, that utilise the variables from the Base Classes of the Player and Enemy.

## Feature #3 - Special Attacks

Tasks:

1. Create a Base Class script that holds the necessary data and variables; such as attack power or damage type; for a normal basic attack, and special attacks and abilities to be used by the Player and Enemy.
2. Create Attack Class scripts that derive from the Base Class to be used by the Player and Enemy.
3. Add a Base Attack variable into the Player and Enemy Base Classes to hold a list of their attacks and abilities.

## Feature #4 - UI

Tasks:

1. Create sprites and use appropriate UI buttons to create an interactable UI similar in structure to Final Fantasy.
2. Through the Players State Machine, allow the UI buttons to access the Players data and interact with the Enemy.
3. Add methods in the Players State Machine for the input of different UI buttons, as well as creation of panels for different types of buttons.
4. Create button prefabs to be utilized by Players State Machine for inputs and creation of different panels.

**Feature #5 - Turn-Based Battle System**

Tasks:

1. Create a Battle Manager Class script to manage and run both of the State Machines of the Player and Enemy.
2. Have Battle Manager run both state machines separately through its own state machine, with the ability to access their data.
3. Ensure that Battle Manager’s state machine has states of the state of the game itself as well.
4. Ensure that the Battle Manager loops back to the start of state machines to repeat turns of Player and Enemy.

**Feature #6 - Decision Tree**

Tasks:

1. Create a base Decision class to be derived from and hold the main evaluate method.
2. Create a DecisionQuery class to run the Decision Tree and hold important variables.
3. Create a DecisionResult class to represent the final result of the Decision Tree.
4. Use the DecisionQuery to create a method that acts as the Tree and returns the result of that Tree.

**Feature #7 - Behaviour Tree**

Tasks:

1. Create a base Node class to hold the basic logic of the Behaviour Tree and derived from it to create the other functions of the Behaviour Tree.
2. Create a NodesStates enum to represent the different states of each Node in the Behaviour Tree.
3. Use Node to create a Selector, to select and run the Nodes within it.
4. Use Node to create a Sequence, to run all of it’s Nodes in a sequence until they all return SUCCESS.

## Feature #8 - Adjustable stats

Tasks:

1. Ensure that the stats of the Player and Enemy are public and always adjustable to change the flow of gameplay.
2. Ensure that the current AI state is public to allow an adjustable AI opponent.

# 

# **CRC Cards**

|  |  |
| --- | --- |
| ***Class Name : Player*** | |
| Subclasses : HeroStateMachine | |
| Superclasses : BaseHero | |
| Responsibilities | Collaborators |
| Interact with Enemy using Attack and lower their HP | Enemy |
| Use to run State Machine used in Player’s turn | Battle Manager |
| Provide Player data to Tree Data | Tree Data |
| Set values for attack target and attacks used | Turn Handle |
| Selects Attack Object to interact and use on Enemy | Attacks |

# 

|  |  |
| --- | --- |
| ***Class Name :*** Enemy | |
| Subclasses : EnemyStateMachine | |
| Superclasses : BaseEnemy | |
| Responsibilities | Collaborators |
| Interact with Player using Attack and lower their HP | Player |
| Use to run State Machine used in Enemy’s turn | Battle Manager |
| Use tree Data in conjunction with Behaviour Tree or Decision Tree | Tree Data |
| Set values for attack target and attacks used | Turn Handle |
| Decide which target and attack to use with the use of the Tree Data, through a Behaviour Tree | Behaviour Tree |
| Decide which target and attack to use with the use of the Tree Data, through a Decision Tree | Decision Tree |

# 

|  |  |
| --- | --- |
| Class Name : Battle Manager | |
| Subclasses : HeroStateMachine, EnemyStateMachine | |
| Superclasses : BattleStateManager | |
| Responsibilities | Collaborators |
| Run state machine to allow Player to interact with Enemy | Player |
| Run state machine to allow Enemy to interact with Player | Enemy |
| Use in conjunction with Enemy state machine to decide on attack and target, through a Behaviour Tree | Behaviour Tree |
| Use in conjunction with Enemy state machine to decide on attack and target, through a Decision Tree | Decision Tree |

# 

|  |  |
| --- | --- |
| Class Name : Turn Handle | |
| Subclasses : | |
| Superclasses : HandleTurn | |
| Responsibilities | Collaborators |
| Use to set values for Player’s target and attack | Player |
| Used to set Value on Enemy’s target and attack | Enemy |
| Used for handling interaction specific data of Player and Enemy | Battle Manager |
| Use evaluated data to decide target and attack for Enemy | Tree Data |

|  |  |
| --- | --- |
| Class Name : Behaviour Tree | |
| Subclasses : ActionNode, Selector, Sequence | |
| Superclasses : Node | |
| Responsibilities | Collaborators |
| Use Tree Data to decide on target and attack | Enemy |
| Select as target based off Tree Data | Player |
| Select attack based off Tree Data | Attacks |
| Use evaluated data to influence decisions | Tree Data |

|  |  |
| --- | --- |
| Class Name : Decision Tree | |
| Subclasses : DecisionQuery, DecisionResult, DecisionQueryTwo, DecisionResultTwo | |
| Superclasses : Decision | |
| Responsibilities | Collaborators |
| Use Tree Data to decide on target and attack | Enemy |
| Select as target based off Tree Data | Player |
| Select attack based off Tree Data | Attacks |
| Use evaluated data to influence decisions | Tree Data |

|  |  |
| --- | --- |
| Class Name : Attacks | |
| Subclasses : BlazingSmite, FireBall, FlameLance, PoisonBomb, StoneBullet, StoneFist, TriMagic, WaterBall, WindBreaker, WindSlash, BackStab, Bash, Caltrops, GreatSlash, KnifeThrow, ShieldBash, Slash, Stab, BlessedWater, GuidingLight | |
| Superclasses : BasicAttack | |
| Responsibilities | Collaborators |
| Use for interaction between Player and Enemy | Player |
| Use for interaction between Enemy and Player | Enemy |
| Select which Enemy attack to use based off data | Behaviour Tree |
| Select which Enemy attack to use based off data | Decision Tree |

|  |  |
| --- | --- |
| Class Name : Tree Data | |
| Subclasses : | |
| Superclasses : BTAttack, BTTarget, DTAttack, DTTarget | |
| Responsibilities | Collaborators |
| Stores data of Player | Player |
| Use data for decision making of Enemy | Enemy |
| Stores data of Enemy | Attack |
| Use evaluated data to influence decisions | Behaviour Tree |
| Use evaluated data to influence decisions | Decision Tree |

# **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).