SURNAME: Moodley
FIRST NAME: Tenisha

STUDENT NO.: 2105982

COURSE CODE: WSOA3003A

ESSAY TOPIC: Micro Project 1: Data Design of a Turn Based Combat Game

Plagiarism Declaration
I, **2105982 - Tenisha Moodley**, know and accept that plagiarism (i.e., to use another's work and to pretend that it is one's own) is wrong. Consequently, I declare that:

- The following essay is my own work.

- I have correctly acknowledged all direct quotes and paraphrased ideas. In addition I have provided a complete, alphabetized reference list, as required by the MLA method of referencing (described in the Student Handbook).

· I have not allowed, and will not allow, anyone to copy my work with the intention of passing it off as his or hers own work.

· I understand that the University of the Witwatersrand may take disciplinary action against me if there is a belief that this is not my own unaided work or that I failed to acknowledge the source of the ideas or words in my writing.

Signed: Tenisha Moodley

Date: 25 March 2021

Regarding the Prototypes Purpose and Requirements

Aspects that were regarded before processing this system were things like what a turn based system requires of players, in order for them to stay engaged with the system. It brings questions as to what the possible goal of the system is to the player. When the goal of winning against the other player or NPC is established, then the system requires the player(s) to engage strategically with the system in order to beat the system through combat or interaction. A combat game system ensures that there is a logical and most possibly a numerical way to win such as that the other player either maintains, gains or loses numbers along the system which leads to a final outcome.

The Intent Behind the Prototype

Prototyping systems help developers design games to the best of ones ability by integrating encouraged techniques such as iterative design and play testing. Therefore, the intent behind creating this prototype was to create, engage with and design the beginning core system of a turn based combat game in order to understand not only prototype smaller aspects of a greater gaming system but also how to observe how data design in turn based combat games require users to engage and interact with the system and in return, how the system then reacts to a user's interaction.

Essentially, this prototype will be used to create a system that manipulates data between a two character combat game. The term 'character' being used instead of 'player' in order to highlight that this system will not be a two player game, but will focus on how a player and a Non-Playable Character (NPC) react towards each other using data manipulation.

Furthermore, this prototype will be using a technique known as a state machine, a system that allows the prototype to move between the player and other system states, such as moving from the player turn state to the enemy turn state once the player has concluded their turn. This is essential to create the allusion of turn play within the turn based system. While the system itself stays the same, the 'state' of the system can change, and this manipulation of state change through data design and communication is the main objective of this prototype.

A table showing what the manipulated can effect in the prototyped system, and how or why that data is being manipulated

What the data, that is intended to be manipulated, can effect:	**How, or why, the data is being manipulated:**
- What state the system is in	- (The state would need to change in order to encourage interaction between the player and the NPC)
- How the system communicates state change to the player	- (Such as through having the players ability to interact with the game physically taken away or using dialogue. It's important to relay feedback to the player about state changes because the player needs to stay aware of how to strategically react to the NPC's reactions and the player won't be able to if they can't figure out what state the system is in)

What the data, that is intended to be manipulated, can effect:	How, or why, the data is being manipulated:
- If the player gets to live	- (If the player's defence statistic is greater than the enemy's attack statistic)
- If the player gets to win	- (If the player's life points are greater than the enemy's)
- What happens when player wins	- (Level increase)
- If the player is attacked	- (If the enemy's accuracy statistic is equal to or greater than the player's evasiveness/speed statistic)
- If attacks have missed	- (When accuracy is less than evasiveness/speed)
- If the attack or defence statistics have lowered or risen. - If accuracy or evasiveness statistics have lowered or risen	- (Attacks effect the statistics of the player or NPC)
- What happens when statistics are maxed out negatively or positively, such as that the player's life points can no longer be healed or if the enemy's accuracy has reached 0 percent accurate	- (Messages will appear to communicate such situations with the player, and the statistics won't proceed further than the minimum or maximum values. It's important to communicate this with the player because if the player doesn't get communication on this they could keep wasting turns instead of winning the system)

The Process recorded while Creating the Prototype

1. Creating a to-do list that entitles the UI, Environment, Game play and Logic behind the turn based combat system

User Interface (UI): Although, at this point of prototyping the UI is not all that important. This system will need to communicate to the player all of the aspects of the system in order to allow the player to engage with the system and interact with it willingly, and the use of functional UI is one of the most effective ways to ensure this. Therefore, the system will need Text box's and numbers being represented to the player in obvious forms of representation. Life bars will be used to represent life instead of just numbers, because numbers will be used primarily to represent the statistics of the player and NPC. The life bars will be used to avoid over whelming the player with numbers and disturbing the

communication with the player. UI buttons will be used to allow the players to interact with the system in order to manipulate data and reach the desired goal of the system.

Environment: will entitle the essential aspects or game bits of the system, such as that there are two characters, a player and an NPC, who will therefore need battle stations of some sorts to represent their position in the game (whether literal composition within the screen or strategic positioning within the system, both are important for communication between the system and the players). Backgrounds, and more detailed aspects will be disregarded for this prototype in the means of efficiency and functionality.

Game play: The alternation between game states are what is used to distinguish the system's progression. This games states would consist of; the start, player 1 turn, NPC turn, won, and lost. Turn based systems have combat, and combat are turn based. This is important to the creation of the system because if there is no form of conflict (violent or non violent) residing in a turn based game, then the game cannot proceed. And in the same way, if a combat game does not have a system which divulge away from the player, such as through a turn based system, then the combat game will not by definition be a combat game unless the one player is being represented through two mediums concerning the theme of the game or system. In order to create this turn based combat system, the system needs to move between data recognised as the states of the system, 1 turn at a time.

Logic: these are the core system scripts we will need to create, a unit script for the player and NPC's information, a battle system script for the game states, and a battle Head's Up Display (HUD) script for updating the unit UI.

2. Implementing What has been Intended of the System Prototype

The process started with establishing what UI would be necessary, to give the player the ability to communicate with the system both through dialogue and physical engagement such as through button actions and instructions through text, and then started writing the core system scripts around the UI. We do this to communicate to the player that their actions have effects on the system and what effects or actions take place after interaction.

Errors such as state changing while communicating to the player when they can or cannot interact with the buttons on screen became inaccurately depicted because in order to control the state machine, coroutines were used which means that the buttons could not disappear until it's the next states turn to appear, therefore granting the player to pose more than one action per turn.

Getting the buttons to disappear and reappear with the players turn became part of the solution, but getting the buttons to be non-interactive while the game gives the player time to process actions cannot be solved with coroutines (as tested). A final full solution to this

problem was eventually figured after constant exploring and play testing. Eventually, the decision to deactivate the interactivity of the buttons every time a button is pressed is the best way to stop the player from pressing more than one button, and therefore performing more than one action, per turn. This is important because the player should be aware that each turn means an action, unless the player gets more than one turn. This encourages the player to engage more with the system by interacting with cautiousness and strategy in mind.

The finalisation of the attacks given to the player in this prototype was an iterative process that discovered that the Smoke attack should lower speed but increase defence because logically smoke affects one's sight and therefore their agility, but increases their sense of cautiousness. While, the Doppelgänger attack lowers accuracy and increases attack damage, because having multiple targets and not knowing which is the real target can frustrate one into attacking more violently without a care for accuracy. The logic behind actions in the system communicate better to the players if they are logically accurate and communication between player and system data is the main goal for this prototype. Furthermore, this prototype's statistics will be static with each play through setup in order to not over scope this prototype and lose focus of the main objective.

The final data manipulated design aspect of this prototype with be that the NPC attacks the player according to the status of its own statistics, such as that if it's HP has dropped below half, then it will heal itself until it's HP is above half again or when certain statistics are greater than the other characters statistics, then data is manipulated to allow things to happen, such as that if the Enemy's speed is greater than that of the Player's accuracy, then the enemy will dodge the attack. This process of data design is important to this prototype because having an NPC that uses data manipulation to interact with the player encourages the player to interact with the NPC more. This form of data design is also the main area of exploration and research for this prototype which made the process necessary and interesting to explore.

Reflecting on the Process

After establishing my intent towards this prototyped system, which was to create a turn based combat system which explores the possibilities of data manipulation, I started the implementation of the system design process immediately by focusing on creating the UI and feedback designs to communicate the system to the players and afterwards working on the core systems or scripts of the system to accompany the UI. However, I realised that by doing this I over scoped the idea of my system.

This micro project took place over a period of a week so there was very little time to create a detailed data manipulated system which follows the accuracy, evasiveness/speed, attack and defence statistics of two players. Therefore, the game was changed from a two player

game that focused on manipulating 4 sets of data to a one player game with 4 sets of data manipulation in order to allow a more focused intent towards how the data between the attack, defence, speed, accuracy and life of the player manipulates.

It would have been nice to explore using a random number generator or dice rolling system to set the statistics with each restart to test the player (even if it was just the enemy that changed every restart) because of how interesting that would have made the data design as every restart of the game would have a different set of data to manipulate, but as previously mentioned, to prevent over scoping, this was not done.

The NPC was also over scoped. This is stated because the NPC could have simply attacked the player to gather the required data from this system prototype, however the NPC not only has more than one type of attack it can use on the player, but it also isn't a randomised selection of attacks. The NPC is wired to understand when it needs healing or when it should lower the players accuracy. Although, the NPC's system is not extremely advanced or particularly inventive, I still believe I ran out of time trying to finish this project with the scope as is.

The way the dodge mechanic was implemented was designed with simplicity in mind so it lacks various calculations to the data such as that the system calculates when a character dodges attacks based on a much more complex algorithm rather than just a simple 'if statement'. Due to lack of time and knowledge, and concerning the simplicity requirements for this prototype, the algorithm to decide how data is manipulated will definitely be looked at more thoroughly in future prototypes in order to make the system more efficient and readable to the players. How this will be done is still jot researched but as previously mentioned, a dice rolling system, even if invisible to the player, would be a great way to initialise the complexity of the system.

All previous designs of this prototype had included level data, but decided to disregard it in the final system because the issue became complicated with the decision of whether I wanted to save game play, allow statistics to change and introduce new player moves with each level increase. This all lead me to the conclusion that I was, again, over scoping.

Lastly, although the defence statistic has been implemented into the system, the defence does not actually work. For the time being it simply shows what data manipulates it but concerning the ability of a defence statistic, the defence does not actually weaken attack damage. This issue was realised late into play testing and the lack of a solution is due to insufficient time management used during the design of this prototype. A quick fix to this issue however, could have been to sightly increase the attack damage statistic of the opposing character whenever the other character's defence rose. This can be added into future prototypes, as it is essential that each aspect of the system has a purpose and demonstrates that purpose to the player accurately.