WSOA3003A Micro-Project 1: Data Design Reflection and Analysis

For the first Micro-Project a prototype had to be made using a turn-based system that focused on the data design. I created a prototype with a turn-based system where the player chooses a command from three choices to execute on their turn. These choices are randomized each turn between six unique commands. These commands are 'Attack', 'Defend', 'Heal', 'Cripple', 'Poison', 'Lifesteal'. To win player must simply reduce the enemies HP to zero and if the players HP is reduced to zero, they lose.

Intent

For this prototype I wanted to create a system that utilizes debuffs or lasting effects which impact the system in some way. This would change how the system handles the data depending on whether the debuff has been applied what the specific type of debuff is.

For this idea I took inspiration from many turn-based games I have played in in the past, mainly old Final Fantasy games and Pokémon. These games have multiple skills that have a long-lasting effect on gameplay either by reducing the stats that inhibits the unit's ability to use spells, hit enemies or even use their turn at all just to name a few. For the sake of a prototype similar skills to the ones mentioned before will be created but simplified just to showcase the potential and how the system would handle the data.

Process

To start, research was done in how to program and create a turn-based game. Through research the use of enumerations was used to control the states and flow of battle. These states would alternate between the player and the enemy until a certain condition was met which would result in either the players victory or defeat. Once these transitions between the turns was complete, the actual commands were created and implemented.

The first commands created were the bare basics needed for the game to function at a minimum and test that the turn system worked. The first was an attack command which just dealt damage to the enemy, this was then also given to the enemy just to allow the battle system to work and reach a potential end. Afterwards a defend command was added, this command when used would let the player take half the damage for that turn when attacked. The last basic command implemented was a heal command that healed a portion of missing health (but would not exceed the maximum health allocated to the unit). Note, at this point only the attack command is given to the enemy.

At this point, the game functioned well enough, and the turn-based system worked. Next the cripple command was implemented, this command cripples the enemy so that they deal less damage for the subsequent turns while the debuff is active. This would last a total of three turns and made the enemy deal one less damage per attack. If the cripple command were used again, the countdown would refresh and last again for three turns.

After the cripple command, the poison command was created. This command would deal a small amount of damage to the enemy on their turns and functioned the same as the cripple in the sense that it lasts three turns and refreshes if used again. There is a bug where if used on the very last turn, without the debuff being previously applied, and the enemies health then falls to zero it would not be seen as a defeated enemy

At this point I did not like having all the commands at the players disposal at every turn and decided to implement a form of randomness. Every turn three commands would be taken randomly from a list and given to the player to choose from (the same command could be given multiple times). I felt like this randomness gave the players more agency.

Lastly a lifesteal command was created which essentially dealt damage and healed the user for the same amount. I added this because it seemed fun and worked well in the turn-based setting.

After all commands were made, they would be implemented for the enemy as well however only the attack and defend commands would remain as the randomness of choice for the enemy was far too random.

Reflection

After this game was completed, I was happy with the result but there are many things which can be done and that I want to implement to better the game.

Firstly, the way in which commands are chosen and presented to the player. I want to make it in such a way that no duplicates are given and a potential deck system where the player can specify which commands and how many they want to enter battle with.

Another change would be the commands the enemy can use. Without a proper AI, the enemy is very basic and cannot use all the commands effectively. Therefore, with a proper AI battles would be more interesting with the enemy having a wider range of commands that also respond effectively to the players commands.

An obvious inclusion would be more variety of commands which would introduce more styles of gameplay affect the data in vastly different ways. This coupled with improved feedback, improved UI, and balancing would provide a much better experience.

All in all, the prototype succeeds in handling data and achieves the intended goals, with this in mind I consider the prototype to be a success overall.