Written Analysis – Data Design

Declan Thompson – 2167636

Intent:

The given assignment was to create a prototype for a game with data systems which interacted with each other, as well as show how the system changes due to this interaction. This took the form of a turn-based play system; I decided to model my prototype after the turn-based battle system from the Dragon Quest series of Japanese role-playing games. My intention was to create a simple yet effective turn-based system with a hidden level of depth or complexity to it – this is why I used Dragon Quest as a basis.

The Dragon Quest series is known for its very traditional and simple turn-based combat system – the player is given a small pool of options such as Attack, Defend, Magic, Items or Flee; they select one of these actions to perform in their turn, and once their turn is over, the enemy is allowed to take their turn. This process repeats until either the player or the enemies are defeated. On the surface, this system is incredibly simplistic, especially in the franchise's earlier titles. But with each new entry in the series, the combat system has been tweaked, modified, and improved to add new layers of hidden depth that may not be obvious to first-time or casual players.

To aid in my process, I used an article from Gamasutra written by John Harris called "Number Punchers: How Final Fantasy and Dragon Quest handle combat math" to further my understanding of Dragon Quest's Data Design systems. (Harris, 2018) Coupling this with my own knowledge of the in-game systems, my original intention was to create a system with the same levels of depth as Dragon Quest's, but I quickly realized that is not feasible for one person to do within the small-time limit we were given. I thus decided instead to set out a basic foundation for turn-based combat which shows the data interaction and manipulation loop, and that can also be expanded upon and improved at a later time.

Process:

I began my process by creating an incredibly simple turn-based combat system consisting of only one action: attack. I decided to start with this, and I progressively added more systems and layers of depth as I went on. To do this, I first created a script which would handle the statistics of both the player character and the enemy they would be fighting. Initially, the only stats included were Health and an Attack stat. The player's only action was to Attack: the value of their Attack stat would simply be subtracted from the enemies Health, and this would represent the amount of damage they took. After this, the enemy would attack in the same fashion, at which point the same thing would happen but to the player instead: their Health would subtract the value of the enemies Attack stat. This was inspired by Dragon Quest's Attack formula but is a simplified version, as my game does not have a Defence statistic built into it. (Harris, 2018) This would loop in the tradition of all turn-based combat systems.

I created a simple HUD system that provided just enough feedback to the player: an action menu on the left from which the player could select an action to perform during their turn, a section which displayed the player's name, level, Health (and Magic Points at a later step in the process), and a section underneath that one which would display Text based on the current situation. The enemy's stats were kept hidden in the tradition of games like Dragon Quest; the player is not aware of the enemies Health until they are defeated. Once the Attack system was proven to be functional, my next step was to add a Magic system which would allow the player two new possible actions: a spell to Heal themselves magic spell to damage the enemy.

I created a new stat for the player called MP (or Magic Points) – each time the player cast a spell, that spell's equivalent MP cost would be deducted from the player's MP pool. The Heal action was

designed to take a relatively low amount of MP so the player could potentially cast it multiple times during the battle if necessary, but the Magic Spell ability was designed to be more expensive so it could only be cast a few times before the player would run out of MP – this was done as the spell was also designed to do more damage than a regular attack; this would force the player to weigh their options and resources during the battle in order to strategize. Do they cast the spell for extra damage but risk not having enough MP for a necessary heal on their next turn?

My next step was to add a Flee action; a commonly found action in turn-based RPG combat. The player has a 1 in 3 chance of fleeing successfully: if they are successful, they flee the battle and the game ends. If they are unsuccessful, they waste their turn, and the enemy gets to take their turn. Based on this chance mechanic, I added a few more chance elements to the already existing actions: normal attacks now had a 20% chance to land a critical strike and do double damage, but also a 10% chance to miss completely, which would essentially waste the turn – again inspired by Dragon Quest's flat 1/32 chance to miss but simplified (Harris, 2018). Spells also had a 10% chance to fail, and the enemy was also given the possibility of missing the player or landing a critical strike. Other than this, the enemy's only action during their turn was to attack – for simplicity, I did not allow the enemy to heal or flee.

Reflection:

Reflecting on the outcome of this assignment, I feel that I was able to make a system that is simple but also effectively showcases the interaction of the data within it, as well as the way the system changes based on that data interaction. While my system is incredibly basic, even when compared to the first Dragon Quest game from 1986, I feel that I was able to implement the most vital elements of that system: the ability to attack, cast Magic, Heal, and Flee; the presence of basic stats for both the player and the enemy; and a system which switches between the turns of the player and their foe.

I also feel that the implementation of "hidden features" such as the chance to miss an attack or land a critical (two elements found within the Dragon Quest games) represent a small portion of the hidden levels of depth found within Dragon Quest's game system which I mentioned earlier – these are not explicitly stated in-game mechanics; the player is only made aware of them by witnessing them happen. They serve as extra mechanics on top of the game's core systems and add more complexity and uncertainty to the in-game battles, and I feel I was able to represent that aspect of the game's data systems in some small capacity. Of course, Dragon Quest has other systems such as Items, Equipment, Skills and the Tension mechanic which add much more complexity to the game's system which I was not able to showcase in my small prototype.

Other mechanics and systems were planned: at one point, there were other stats such as Defence – the player's Attack would do Damage dependent on the enemy's Defence stat instead of their Health, but I could not quite get this to work. I wanted other stats such as Agility and Luck which would affect the player's Critical chance, Dodge change and Flee chance; as well as stats which would affect the strength of their Heal and Fireball spells, but I also could not find a way to implement these within the time limit. I also wanted to make the player's Attack do Damage within a range of numbers but when I attempted to implement this, it conflicted with the rest of my systems.

Overall, I feel that I was able to create a basic system which serves its purpose of displaying the intricacies of its Data Design. It is a system which appears simplistic on the surface and is relatively easy to use but has layers to it which allow the player to engage it more strategically. While I was not able to implement everything I wanted to, the prototype as it is serves as a basic showcase for what the system is, how it works and what it can be; as well as a foundation for a more complex system more resembling that found within Dragon Quest. I have created a system which I personally want to expand upon further in the future without the pressures of time constraints – I would like to eventually add most or all of the previously mentioned features which were planned but not implemented.

References:

1. Harris, J. (October 4, 2018) *Number Punchers: How Final Fantasy and Dragon Quest handle combat math.* URL: https://www.gamasutra.com/view/news/325537/Number_Punchers_How_Final_Fantasy_and_Dragon_Quest_handle_combat_math.php