Assignment 1: Data Design Analysis: Turn-Based Combat System

Rayyan Gafoor, 1839637

Abstract

This report documents the design process for the prototype combat game, using a turn-based mechanic. The purpose of this prototype is for the designer to focus on Data Design and implement a mechanic that manipulates this data. This report will discuss the intentions of the designer, how the designer implemented the mechanics and finally the reflection on the final product. The mechanics all underwent different iterations to incorporate a variety of data.

Intention

Given the assignment brief, the designer decided to explore the idea of a turn-based combats system. To do this the designer decided to build on the basic idea and combat of Pokémon. The designer aimed to improve this idea by combining it with mechanics from games recently played, such as: Costume Quest 1& 2 and Genshin Impact. Both these games are completely different, so the designer decided to implement one mechanic form each. From Costume Quest, the designer enjoys using the buffs provided to the player during combat, and from Genshin Impact the designer was influenced to use the idea of different elements and the mechanic to combine these different elements to form new attacks. To do this and build on the turn-based system, the designer would have to develop different formulas, using the data sets created, and test which would work best with each other. The designer also saw this as a learning opportunity for the following reason: To be able to switch turns during the combat, the designer had to learn what are and how to use state machines. The following sections will discuss the design process, iterations and the designer's reflection on the assignment. Overall, the designer believes that using these two ideas together would be a great addition to the turn-based combat system.

Process

The designer began by doing research on the different types of elements, that could be used in the game. Using 7 basic elements from Genshin Impact with an edition of two more. The elements are as follows: Pyro, Hydro, Aero, Geo, Electro, Cryo, Dendro and the two new elements are: Aether and Necro.

Each of these elements were then planned to react differently with each when combined with the characters respective element (for the purpose of this prototype we will be observing the elements reactions when combined with a hydro typed character). In order to use the ideas previously mentioned together, the designer decided to combine the idea of buffs with the idea of elemental combinations. This was done by creating a new set of formulas that would apply to the characters attack when a buff is activated, in this case the buff were now the elements. For example, when we would apply the Cryo Buff to the Hydro characters normal attack, it would cause freeze which would stun the opponent for one round. Please refer to Appendix A for the elemental reaction buffs.

Moving one the designer also had to develop formulas for the basic mechanics that are used in the game. During the planning phase the designer came up with a variety of data for each character to house. The different data sets allowed the designer to develop more formulas, which all had different effects and allowed for more mechanics and actions. The designer decided to use certain data sets for the purpose of this prototype (each action has its own formula in which these data sets have been used). As a result the following actions/ mechanics are provided to the player: Attack (this action use takes in to consideration the opponents health and defence and the character health), Heal (this action use the characters intelligence data in order to determine the new health level), Stun (this action use both the character's and opponent's intelligence and uses it to calculate the chance of the stun attack landing successfully) and finally the Special Attack (this action determines whether or not the characters

elemental damage can be applied to the opponent by looking at their elemental types). Please look at appendix B for a table displaying the formulas used for these actions and the iterations they underwent for balancing. Some Iterations have not been successfully implemented, hence the designer decided to use a simpler formula for the purpose of this prototype.

Reflection.

As the designer, the broad brief for the project allowed me to challenge myself, by using methods and developing a game in a genre I am not familiar with. This served as a new learning experience for the designer, as the designer has never used these types of methods or developed a game of this genre before. Using the discussions from recent sessions and engaging with previous years content I was able to successfully develop the prototype and have gained more experience regarding game state machines. The designer has successfully learned how to use state machine and hopes to continue using it to better current and future projects.

During the planning phase of the project, I was at ease and enjoyed developing different formulas for different outcomes. Although trying to find the perfect balance between different combination of formulas proved to be more difficult than expected. As I progressed into the implementation phase, I realised that certain formulas would either give me errors or would be to complex to be used with some of the more basic formulas. Certain Buff formulas would buff the character to an extent of always winning or some would cancel out the effects of the other. Another effect that could be the reason for this, is that the data given to each character has not been properly assigned, as it has been mostly allocated randomly. As a result, certain data sets had to be removed. Although the idea of using the elements as buff work out well, aside from the minor imbalances, and I believe it can still be turn into a great mechanic if improved on.

For now, I am happy with the outcome of the prototype. The formulas are not to complex so the user can actually determine the effects it will have on their opponent. I believe my use of data sets are efficient enough to satisfy the requirements of the project, although I could have used a more diverse set of data for more actions, as I planned to do in the beginning of the project. As an assignment I found this fun and interesting to build, experimenting with the different data sets and coming up with new ideas, that I hope to implement and improve in a later stage.

Recommendation

The prototype does make use of data manipulation, although it could be more balanced in terms or the value each data set carries as well as the actual formulas that allow for the manipulation. As the designer I found that coming up with new data sets for new formulas was easy, but the actual implementation and uses of these data sets were found to be challenging, therefore developing new mechanics/ actions such as new forms of buffs would also improve game play and allow for more data variety. By adding more data sets, these formulas could also be improved on and could also lead to better combat between characters and allow for more strategy to be developed as a dynamic.

APPENDIX A: Table showing Elemental Buff Reactions For Hydro Slime Character

Base Element	Elemental Buff	Elemental Buff Result
Hydro	Aether	"Life Gain"- player gains 30% heath.
Hydro	Necro	"Life Steal"- Enemy loses 5HP, player gains 5HP.
Hydro	Pyro	"Vaporized"- Enemy loses an additional 30% of HP.
Hydro	Hydro	"Heal"- Player gains 5HP.
Hydro	Aero	"Energy Recharge"- Player gains 1 ENG.
Hydro	Electro	"Shock"- Additional 6 DMG is applied to the enemy and causes a stun.
Hydro	Dendro	"Heal" - Player gains 2 HP
Hydro	Cryo	"Freeze"- Additional 2 DMG is applied to the enemy and causes a stun.
Hydro	Geo	"Stun" -Players DEF gains 2 points and a stun is applied to the enemy.

APPENDIX B: Table showing action formulas and iterations

Action	Iteration 1 (currently in use)	Iteration 2	Iteration 3
Attack Damage	Attack- Defence	Basic: Health – Attack	(Attack/Defence) x (C)
Heal	CurrentHealth +(INT/2)	CurrentHealth+ (Health x INT/100)	CurrentHealth+ INT
Stun	Stun Chance= K+ (Player INT /100) Stun Block= K+ (Enemy INT/100)	K +(Player INT/ Enemy INT) x 0.01	Player INT- Enemy INT
Special	Attack + AttackBoost [if elemental opposites are present] Or Attack + Attack (5%)	Attack+AttackBoost-Defence	(Attack+ AttackBoost-Defence) x(C)

Where: K = Random.value (returns a value between 0 and 1), C is a buff multiplier and AttackBoost= (ElementalDamage x Attack)/ 100.