

MDA Analysis

The intent behind the design of this prototype was to create a turn-based system with data manipulation and interaction in mind, the said data was to be used to control the resulting changes in the system itself through player input, what the player can do and when they can do it. The system is intended to pit the player avatar against different four enemies, where player and enemy take turns lowering their health (hp), each enemy's health being affected differently depending on what ability the player used and declaring a winner when health reaches zero.

The following are the steps and techniques the designer used in the process of designing this system:

- For changing turns, the state design pattern was used, where I created two possible states, a state for the player and a state for the enemy. Within each respective state there are functions being called that contain what the player or enemy can do when either one of their states is called or rather, when it is their turn. The state switches to enemy turn after getting data that player has shot an ability to the enemy and has caused damage, then back to player turn after enemies deal damage to player.
- Player avatar. The player can move the avatar left and right to select an enemy to fire an ability at, this is done using C# coding where if the player presses A, D or left and right arrow keys the avatars positional data is changed to correspond with the direction the player wants to go to. After selecting an ability, the player can fire it to an enemy by clicking input button spacebar.
- Attacking and defending. Through C# coding and Unity UI I made it so the player has the choice to either heal or attack, if they click the attack button, the button calls a function to activate the available abilities. Should they choose heal the button will call a function to skip the player's turn and increase their health.
- Abilities. There are three elemental abilities, fireball, lightning, and a fighting ability. Since abilities are shot at enemies like bullets, depending on what ability the player chose or rather the data that the bullet carries, the different enemies will have their hp affected differently. When a fireball hits a normal enemy it carries more damage than if it were to hit a rock enemy, it causes more if it hits an ice enemy. Lightning carries more damage data if it hits a normal enemy than if it were to hit an ice enemy, it

causes far less if it hits a rock enemy. The fighting has more damage data against rock and ice enemies than against normal enemies.

The system makes use of player input data and data within the system itself that the player can manipulate to affect the result from the system. Upon finishing the prototype for this system most of my intentions for it were met: There is data interaction between the player and system, if it detects that the player has chosen to attack it will show abilities each with their own data that change another set of data, if they choose to heal the data that is affected is player health and their turn ends, during the enemy turn, data that represents player health is affected as well.

However, I was not able to make proper use of coding design patterns and regret not planning a structure to follow beforehand. What I can improve upon this system would be to have a smoother interaction of the data especially between player and enemy states, ideally, I intended to have each enemy attack player one by one but the way this is coded for enemies attack at once the second it is their turn and immediately switch to player turn afterwards. I also would have added *spells* as another mechanic players can make use of where players can increase their attacks, increase defence, decrease enemy accuracy and more.

In summary data interaction in this system occurs between the data that controls the switching of turns, input data from player, player health, enemy health, data that manipulates enemy health through player input, and manipulation of player health through enemies.