INTENTION OF DESIGN

The intention behind the micro project is to create a system that is directly affected by the data implemented in the code. The inspiration for this micro project was Darkest Dungeon which makes use of character base stats and manipulates these stats with abilities and items. This data does not need to be visible to the player however it always needs to be working in the background. As mentioned above the most important data is the character's base stats, these include health, damage, crit amount and health amount. Having four simple stats that are easily manipulatable made it easier to focus on how to implement them in game to show the data interaction within the micro project as well as to manage them throughout every change made in the game.

PROCESS OF DESIGN

Designing the game started with a focus on the initial combat system in its simplest form. Initially worked on the base stats such as health and damage and implementing them in game. In order to reduce the amount of work, a unit script which holds the base stats for any game object that would hold the script was made for both the player and the enemy. As most of the stats are public and can be manipulated in the inspector, this was convenient as some stats can be put to zero for one unit but not for the other, for example, the enemy does not have any crit chance whereas the player has crit chance [1].

Initially a battle system script was made to create the combat in game. The script made use of enumators to create five states which were START, PLAYER, ENEMY, WIN, LOSS. These would be used to keep track of which turn the game is in. At the start of the game the system would be in the START state and the set up function would be called to initiate all the important information the game needs to function. It would spawn the player and enemy in their respective positions, assign the correct stats where they were needed and update the player and enemy HUD, this will be explored further. Once the game was done setting up the battle, it would then move to the PLAYER state which held functionality for the player turn. The game would then loop between the PLAYER and ENEMY states until the player either loses or wins.

In order to see whether the system works, combat was the first area worked on. With only the health and damage stats being used, the attack functionality was added to both the player and the enemy. The player would be able to control their attack via an attack UI button whereas the enemy would attack whenever it is their turn. When this functionality was successfully

added, some modification was made to the player's attack function so as to give the player a secondary option during their turn.

The final function added to the game was the special attack or crit hit. This was another option only the player had. However, unlike the attack and heal option the player has, this function added a chance mechanic. In the battle system script a randomizer function was created. This is a bool function that randomly chooses a number between one and twenty. Depending on the outcome of the random.range, the function would either return a true or a false. This function would then be called whenever the player chose to use their special attack. If they did, the crit amount variable would be passed into the function and await the result. If the player hit, the damage would be a result of a calculation [2] but if the player missed, they would hit for zero damage.

REFLECTION OF DESIGN

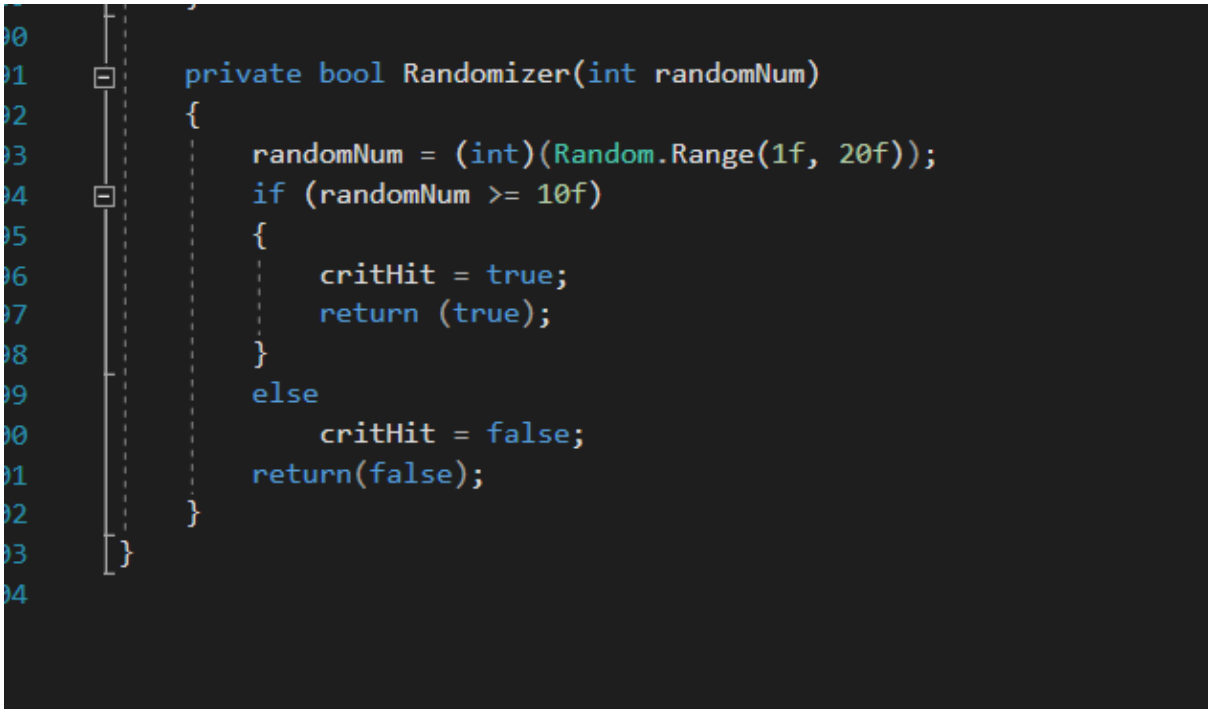
There was functionality that I was not able to implement in the game. The first was a proper randomization function. The one in game serves its purpose however at times it returns the same result constantly, this in turn made the player always hit their special attack or always miss it. Another functionality that was not added was a heal for the enemy based on their health. The idea was that the enemy would always attack until it reached half health. At that point there would be a 50% that the enemy would either attack or heal. If the enemy was at 25% health or lower, the heal chance would be increased to 75%. This idea would have showcased the interaction of data much more than what the game currently presents as the most interesting interaction of data mostly appears when the player uses their special attack.

APPENDIX

[1] Player and enemy stats.

	A	B	C	D	E	F	G	H	I	J	K	L
1	TURN BASED COMBAT GAME DATA											
2												
3	PLAYER STATS	1st	2nd	3rd				ENEMY STATS	1st	2nd		
4	Health	10	50	30				Health	100	200		
5	Damage	10	5	5				Damage	3	3		
6	Crit Amount	5	2	4				Crit Amount	3	2		
7	Heal Amount	5	5	10				Heal Amount	10	15		
8												
9	FORMULAS											
10	crit modifier	damage * crit amount										

[2] Randomizer function



```
90
91      private bool Randomizer(int randomNum)
92      {
93          randomNum = (int)(Random.Range(1f, 20f));
94          if (randomNum >= 10f)
95          {
96              critHit = true;
97              return (true);
98          }
99          else
00              critHit = false;
01          return(false);
02      }
03  }
04
```