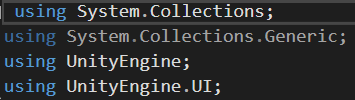
Battle System

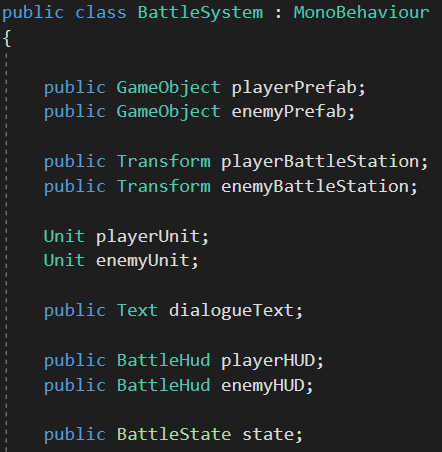
This first script of the Turn Based Mechanics is the largest one. It uses Unity UI systems and all the visuals of the component are held by the UI in Unity. This scrip is going to be attached to an empty object in the scene.



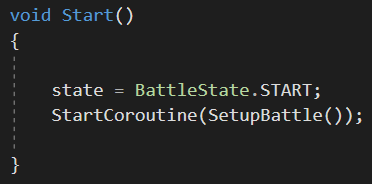
In this class is used IEnumerator system which allows to create different states of the program. As in chest game that would follow as Start of the battle, Player Turn, Enemy Turn, Win state and a Lost state.



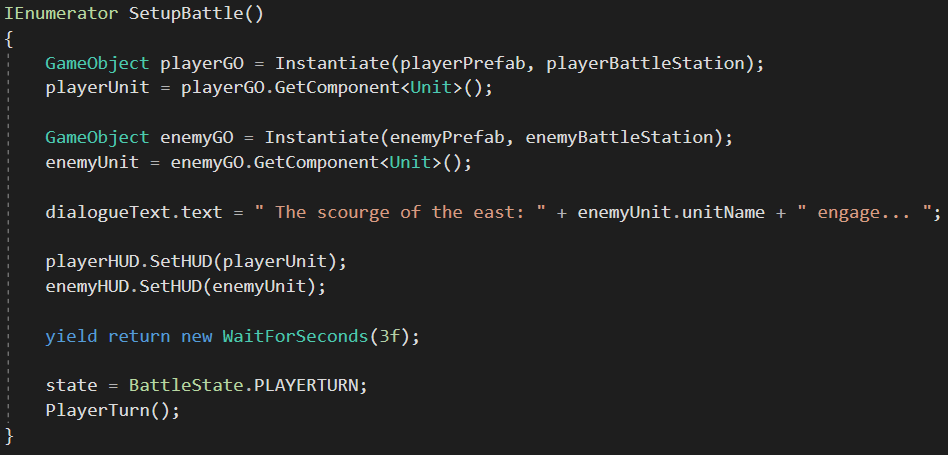
In the beginning of this class are declared references that would appear in the Inspector. It is going to be able to refer to the player and the enemy Prefabs containing 2D visual assets, the Battle Stations of both that will define their exact locations on the screen, player and enemy Units that will hold information for the stats of the characters. Also, a reference to a text box containing short dialogue information, the player and the enemy HUDs’ that would refer to elements in the UI in the final tutorial. And last is a reference to the Battle States.



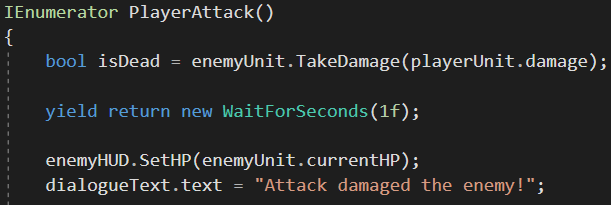
In the void start is the first Battle State and a Coroutine that will set up the battlefield.



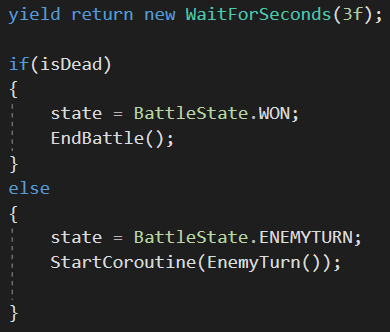
In this IEnumerator the program instantiates the player and the enemy Prefabs on top of their Units in the UI. The short text will appear in the dialogue text box. The information of the characters is loaded in their HUDs. Then the program will wait for 3 seconds, the player can orient in the situation and read all the information. Right after that the Battle State is switched to the Player’s Turn.

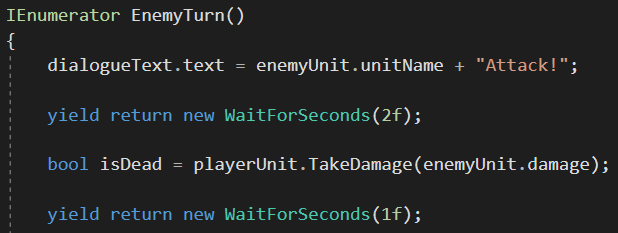


During the Player’s Attack is set a Bool that will check if the Enemy is dead and then wait for one more second, the Hit Points of the Enemy are reduced and in the text box is displayed a message that indicates that Enemy had been hit.

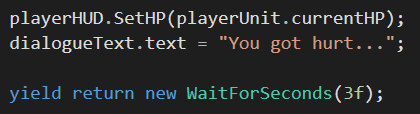


The Player’s Turn will pause for three more seconds, note that those pauses of the program are set by me in a matter of Game Design and can be adjusted by preference. Right after those three seconds the Bool is being checked and if the Enemy is dead that would End the Battle. And if not, the program would switch to Enemy’s Turn, with starting the next Coroutine.

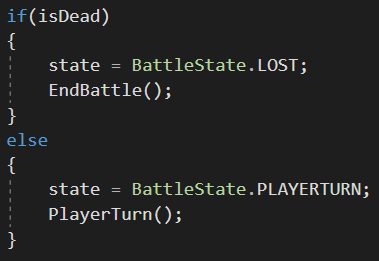


The Enemy’s Turn starts with displaying its name and the phrase “Attack!” and the program will pause on that moment for two seconds. Just like in the previous State here is set a ne Bool that will be checking if the Player is Dead or not and wait for another one second. 

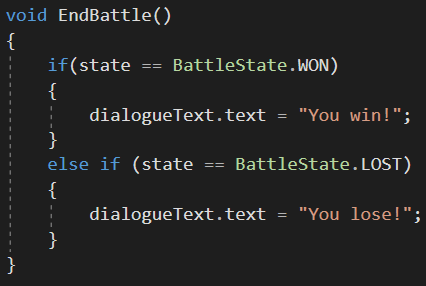
Right after that the Player’s HUD is updated and a new message appears into the text box in the UI saying that the player got hurt and wait for another three seconds.



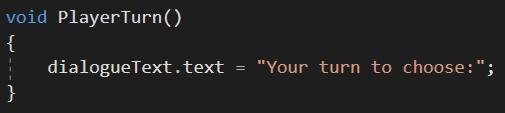
Just like in the previous State, the Bool is being checked and if the Player is dead that would discontinue the battle, or switch to Player’s Turn again.



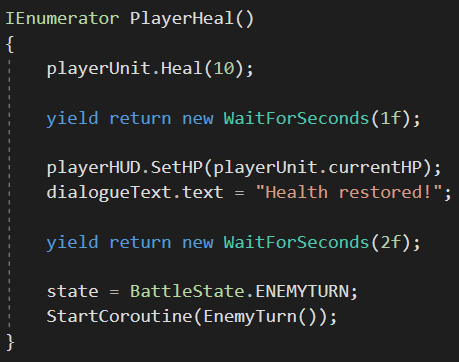
In the next line is created a condition for the end of the battle, which simply will display a message into the text box saying, “You win” or “You lose”.



The next lines display “Your turn to choose” every time that this cycle repeats.



For the purposes of this project I have added the option for the Player to restore its own Hit Points in a form of a Heal function, just to make it more interesting and a bit more complex. The function takes time during the Player’s Turn and adds 10 Hit Points, note that this number is just a matter of Game Design and can be adjusted. As a normal Player turn the program would wait for one second and then update the Player’s HUD and display a message into the text box saying, “Health restored”, then wait for two seconds and switch the Enemy’s Turn.



The final lines of this script refer to the Buttons in the UI for Attack and Heal, saying that those would be available for only during the Player’s Turn and will execute the correct Coroutine.

