Minh Nguyen

CS 162 Final Project

Design

For the final project, I am going to making some very drastic changes from my previous assignment. Firstly, I will make my abstract Monster class have virtual functions for each attack, defense, damage etc.. functions. This is needed since I will be creating dynamic objects based on user input when they are choosing their fighting line up. This will allow me to call the elements in each vector that I am creating as the STL container to hold the monsters.

There will be a total of 3 vector containers, 2 for each player's line up, and a losing container that holds the monsters that are defeated. I will not be implementing any sort of recovery attribute after each fight, but will allow the fighters to recover to full health.

I will use the same basic algorithm for the actual fighting deductions as before, except for in the places of the object calling, I will call it using the vector's position. IE, position 1 monster from player 1's line up will fight position 1's monster from player 2's line up.

After each fight, I will move the winner into the back of their respective line up using the vector swap function, and also move the loser into the Losers vector where in which I will hold to announce their places after the tournament. When the last monster is killed and only one remains, the while loop will end and I will print out the monsters order in which they entered the losing container.

Test

To test my program, I already know that my algorithm for the fighting schematic works. So to add onto testing for this program, I started by only adding one element to each player's container and seeing if I can get the objects of the classes to call their defense / attack etc. Once this worked, I replaced the battling algorithm with the vector's objects instead of the class objects as I had before. Upon confirming that this works, Then I add more elements to the container and see if the next fight continues. When this proved that it worked, I created the first dynamic class by user input, and stuck it into the container depending on which player is choosing. Then when the fight begins, I will call each element directly in the vector to represent the monster they are fighting.

Reflection

Upon completing this assignment, I was left with a few requirements of the lab that I could not complete. Firstly, I was unable to allow the winning creature to return to the end of it 's respective container, therefore, my program was forced to be truncated down to 3 fights, and only have 3 losers that enter the losing container. I did my best to work with user inputs and containers, but I wasn't able to code my program in such a way as to allow for repetitive use of the monsters in the containers when they win. I was however successful in moving the losing monster into the losing container.