

Author: Ryan Wells Farran
Course: CS 162 400 S2019 (Intro to CS II)
Date: 05/27/2019 | Sunday, May 27th 2019

Project 4 Reflection Document

I started Project 3 late and so it was fairly sloppy; so, before beginning this project I went back and cleaned up all of my code and console output. This was a great way to defamiliarize myself with the code because I had not looked at it since I had turned it in. I made the mistake of jumping in and coding my containers before stopping to think of how the program would work.

I at first was designing the class for the linked list to have a second pointer to `headOfLoserList` and I had code to add the players to the new head pointer when they died, but upon further developing my game I realized that it would have made way more sense to just have a third generic list and use the same add functions to them. I also made the mistake of using a stack instead of a queue, and by the time I realized this, it was sort of too late and I was too demoralized to go back and change everything; another great reason to have planned better.

I had issues with my memory and destructor. I ended up basically commenting out most of the delete statements to get my code to be able to compile and finish an entire tournament, although because of this you can not restart the game. My program also did not print the loser list correctly, but since I was already turning this in late to get more time on it, I decided to cut my losses.

Test Plan:

The `inputValidation` classes did not change, so the input/output was the same for this project as with the last.

The program could handle multiple fighters being the same type of character no problem.

The program could also handle different quit combinations such as player one choosing a character and player two quitting, vice versa.

This is another example of me struggling to complete these projects on time because of poor planning. If I had more time outside of school I would have loved to really take my time on this project and iron out all of the bugs.

Class Hierarchy

