

Sean Reilly

Reflections --- Assignment 3 --- Create a fantasy combat game

Design Description/Documentation

My original design called for 2 separate headers to break down the fighting and the characters. Once I started getting into the thick of it I couldn't think of a good way to incorporate the dice aspect that was needed to simulate the battles. I tried to add it to the fight.h but I needed a new constructor and it got too complicated for me to even understand what I was writing so I broke it off to its own class and header. It getting too complicated was actually a blessing in disguise as I could put the dice wherever it was needed in a more efficient way. A more detailed description of what is in the headers/main is as follows:

Dice.h – this is what I used to set up the random rolling aspect of the game. Has a default constructor, a set number of faces, and die sides which can be changed easily as needed. Separated it out from main as I realized I needed to use it in both type.h and main.cpp. I had also written this same file earlier in the class and just used that pretty much verbatim.

Fight.h – this was basically created to store the damage statistics of what happened when the characters were fighting. It's also where you can add and remove characters and stores the character list which was stored in a vector.

Type.h – this is essentially the meat of the program. Contains all the character types. Uses inheritance for each specific class of character. Has all the stats on characters armor, strength, etc.

Main.cpp – menu for the game. Allows you to create your character, remove a character already created or fight. Every header is fed into this.

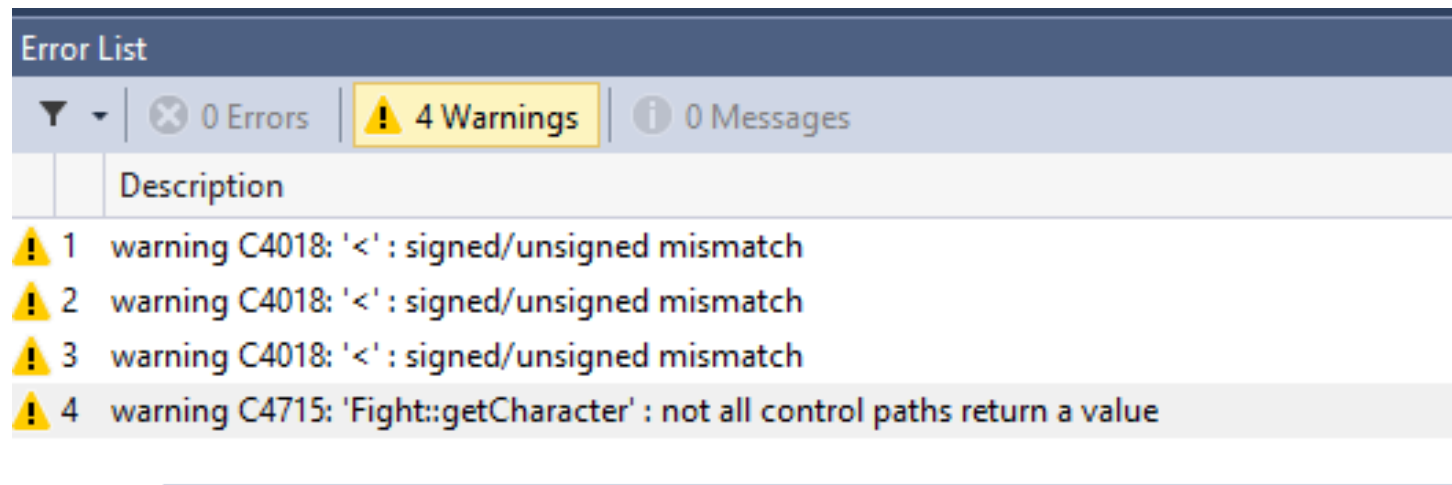
Test Plan

I did the same thing I've done in the past – broke everything down and test individually before incorporating with the rest of the program. The first thing I did was I wrote the main.cpp just to get my menu working. I thought this would be the best way as I could comment out the switch statements that weren't needed and run it without having interference. This idea worked as planned, but my code was a bit shaky which I'll get into more below. The next test was to be adding characters as I could not test the fighting aspect of the game without the characters. Once this was done I then tested the fighting sequence.

Test Results

My testing initially went rather poorly. I'll review in the order of the test plan above. In the main I didn't realize I could use a switch within a switch so it took me forever to figure out how to do what I wanted it to do. Other than that, it was a standard menu that I have used many times in the past and was not an issue. I could've essentially copy/pasted from another program and just changed the dialogue if I needed to. Next was the character type file. I ran into trouble using pointers as I usually do

--- still don't fully understand the concept to be honest. I also didn't originally have constructors for all the characters initially which was really dumb on my part and caused me some major headaches. The dice part gave me no trouble other than the few times where I'd run it and get the same value back on attacking and defending but after repeated testing this seems to be just an anomaly. Once all the classes were working it was easy to go back in and add the Achilles part, which I added on the end. I finally got everything to work and when all was said and done I finally got it to look like this:



On line 19, 30, 83, and 37 in fight.cpp. I was not able to upload to flip but it compiled for me. Hopefully will not lose points for this as I wanted to check but don't know how. Extremely worried that this will cause it to fail on flip. But it did compile for me and works quite nicely I might add!