Daniel Olivas

11/20/2016

Assignment 4 Reflection

Design

For my creature classes, I only need added 2 functions to them. I added a pure virtual function heal and pure virtual getName function to Creature class. I designed my healing function to add 3 strength points to the winner of the battle. If the 3 points put strength points above original strength, I just restored the creature's strength to starting strength. The getName function was added to have a way of displaying the creature's name.

I added Queue and Stack classes to handle holding the creatures. The Queue class and the Stack class are virtually identical to my lab 6 assignment. Queue pushes and hold each team's creatures and pops them off for battle. The winner is then popped back to the queue. Stack pushes the loser into a loser pile and is popped to display the losers after the tournament has concluded. I added an isEmpty function to both Queue and Stack. IsEmpty is a bool function that return true as long as list is empty.

Much of the Tournament Class was able to be carried over from assignment 3. I made a FighterQueue function that gets the player names, number of creatures for the teams, and pushes the creatures into the queue. I have a cMenu function that displays the creature list with associated number for user to select creature.

The battle function in Tournament class is largely the same as assignment 3. Aside from popping creature from queue, the only changes made were cosmetic in nature. I added text to display the winner of each round. I also added an option in between rounds to for the user to select to see the current score. I made a separate currScore function that displays the current score. For scoring, I made a simple winner gets 1 point, loser gets 0 per round.

Winner function is called in main and displays the winner with how many points they had compared to the loser. Loser function in Tournament class is called in main if user wishes to see loser pile.

Test Plan

From assignment 3, I knew that my creatures tested were good and my battle was also sound. The major items I need to test were my Queue and Stack classes. After I had finished had tried implemented my Queue and was popping the creatures into the battle, I had no way getting out of battle sequence once one team all creatures lose. I was getting to the "Queue is empty" and then shutting down. To fix this, I made the isEmpty function to return false once the queue is not empty, thus leaving the while loop. I anticipated the same would happen once I started testing my Stack class and made the same function there as well.

While selecting the creatures for each team, I had originally had the menu being displayed between each creature selection. I found that to be a little to clunky, so I changed it so the menu would only be displayed once for the user to select creatures. This was fine for small teams, but when testing larger teams, the menu would go out of scope. I added an option for the user to press x to display the menu again.

To test my program, rather than have teams of random creatures, I decided the best way to test would be to have teams of the same creatures similar to how assignment 3 was tested. I would test a team of 10 Blue Men vs 10 Barbarians to ensure that the results were proportionately the same as assignment 3 results.

Test Results

The results ended up, more or less, in line with the results of assignment 3. While I ran 1000 of each battle in assignment 3, I decided to scale back my test to just 10 battles for each test.

Win Percentages:

Blue Men: 100%

Harry Potter: 65%

Vampire: 49%

Barbarian: 31%

Medusa: 11%

Barbarian Vs Blue Men

Barbarian: 0

Blue Men: 10

Vampire Vs Blue Men

Vampire: 0

Blue Men: 10

Blue Men Vs Harry Potter

Blue Men: 10

Harry Potter: 0

Medusa Vs Harry Potter

Medusa: 0

Harry Potter: 10

Barbarian Vs Harry Potter

Barbarian: 2

Harry Potter: 10

Blue Men Vs Medusa

Blue Men: 10

Medusa: 0

Vampire Vs Medusa

Vampire: 10

Medusa: 2

Vampire Vs Harry Potter

Vampire: 4

Harry Potter: 10

Vampire Vs Barbarian

Vampire: 10

Barbarian: 3

Barbarian Vs Medusa

Barbarian: 10

Medusa: 3

Vampire Vs Vampire

Vampire 1: 10

Vampire 2: 9

Barbarian Vs Barbarian

Barbarian 1: 10

Barbarian 2: 9

Blue Men Vs Blue Men

Blue Men 1: 5

Blue Men 2: 10

Medusa Vs Medusa

Medusa 1: 10

Medusa 2: 8

Harry Potter Vs Harry Potter

Harry Potter 1: 10

Harry Potter 2: 8