Richard Moot
November 23, 2015
CS 162 - 400

Assignment 4 – Containers

Design

My design for this assignment will be to take my previously implemented combat and creature classes and use them within a tournament class. The tournament class will be responsible for handling creature selection, selecting a lineup size, determining winners, hold the containers for our player's lineups in queues, and also have the losers in a stack. I anticipate to minimally alter the original creature and combat classes, since they were previously implemented in a fairly modular manner. I would only have to modify my battlePhase function to handle determining which creature lost combat, which was previously handled by my driver that was in Main for the previous assignment.

I anticipate that the Tournament class will do most of the heavy lifting on this, and will leave a cleaner main function as a result. My goal here is ensure that the creatures are properly moved between containers so that combat could be smoothly facilitated along with determining winners of each battle.

Reflections

My initial reflection on this assignment was that time management is very key to delivering your project, since hurdles can end up being very time consuming which will draw out your anticipated delivery time. I had struggled greatly with several segmentation faults that ultimately were the result of improperly accessing container data and reaching too far. Once I was able to remedy these, the rest of the assignment was able to fall into place. My winner board would finally display all winners with their correct names.

I had decided as a result of time constraints to implement the simplest version possible of most things (i.e. scoring is based of wins, ties are decided by coin toss). Ideally, I would have preferred to implement a more robust scoring system, since it would seem fairer for certain creatures to get more points for defeating others. Beyond that, it has further helped in increasing my comfort in working with pointers and greatly reduced my compile time errors (as well as better help me to troubleshoot segmentation faults).

Testing

1. Basic Testing
   a. Lineup sizes from 2 to 10
   b. Creatures of same type correctly complete combat
   c. Character winning placement is correct
   d. Naming displays correctly
   e. Winner properly declared
2. Advanced Testing
   a. Special trigger effects that *aren't* supposed to trigger do not trigger
   b. Special characters trigger their effects

Testing Results

1a. This was the first test the elicited that elements in the containers were not being properly accessed and caused segmentation faults (too many ->next's). Ultimately the program could handle up to 10 creatures for each lineup.

1b. Caused no issues, expected results of combat resolving normally worked.

1c. Demonstrated that too many calls to removeLoser() were being made, which caused the name to be incorrect in the standings. I resolved this by caching the removed loser and call method upon it.

1d. Was done using displayRoster functions, worked as anticipated.

1e. After fixing the naming issue with a previous test, this had worked as anticipated.

2a. Testing for this is not definitive, since you are proving a negative, but it works be creating large Goblin rosters on each side and letting the fight (around 10 on each side). Ultimately worked as anticipated.

2b. Can only be tested through excessive battle between Goblin and Shadow to see Achillies and Dodge trigger.