Programming Project Report

Name: Jan Bobda
Date: 11/19/2018

Academic Integrity Statement: I pledge that I have neither given nor received unauthorized help on this programming assignment.

Problem Statement:

The goal in project 6 was to work on creating a small text-based dungeons and dragons inspired game by focusing on designing and implementing a monster class with a simple combat system. For inputs, the program takes in the two indices of the monsters in the monster array and for the outputs the program runs a battle simulation of the two monsters until one of the monsters dies. For error handling I added clamps to the constructors and set methods to prevent textfiles being read in that have invalid values. I also made it so that a monster cannot battle itself.

Design:

For design decisions, I followed the instructions for the class design which involved creating and encapsulating five member variables. I also made the attack method of the monster class only do one attack rather than going through the entire battle simulation. The only data structure that I used was the standard C-style array which stored monster objects read in from the monster.txt file. The pros of using the attack method to only simulate one attack rather than the whole battle was that it simplified the code and made it make more sense in my eyes. One pro of using that standard C-style array was that it was simple and easy to use and got the job done without any real memory overhead.

Implementation:

I started the program with no code provided so I had to follow the instructions to build the monster class. I started by simply creating the class header file with the function headers and the basic variable declarations and making sure that the class compiled correctly on my machine. From there I moved to implementing some of the methods and gradually testing them to make sure that they all worked properly. After this, I began working on making the actual battle simulation by first creating the monster array by reading in stats from the monsters.txt file. Finally, I created the simulation loop that runs through the simulation until the battle between the two monsters is done.

Testing:

I tested the program on several different inputs to make sure that it worked for the error handling that was required in the program. The tests are included in a separate file marked for testing output.

Conclusions:

Overall the program seems to be able to deal with the typical use cases and can react to them pretty well. The error checking that is implemented is effective and meets the guidelines set in the original assignment. The programming project was a success because it achieved everything it need to and the code was relatively clean and well commented.If I could do something different, I would have used some more advanced techniques to create more elegant code. This project could have used some changes that would have made the design more elegant. Instead of using a C-style array to store the monsters, I believe a vector would have been much better and more dynamic. The project, including the documentation, took roughly 3 and a half hours to complete.