Arjun Kahlon

May 8th 2018

Project 3 Reflection

This project had me build upon concepts from the previous project as well as from the previous Lab 4. Similar to both these assignments, I was tasked with creating a base class which would derive into multiple sub-classes. In this case, I created a Character class which had the attributes of attack, defense, armor, and strength. The attack and defense functions were generated from a dice roll. These functions were virtual and were overridden in the sub-classes. Most of the characters had a special ability. The Vampire had the ability of Charm, which prevented the enemy attacker from doing damage based on a 50/50 roll. This was implemented in the defense() function for the Vampire. The BlueMen would lose a die every time they reached a certain strength threshold. This was also implemented in the defense() function for BlueMen. Harry Potter ability allowed him to revive after dying for the 1st time. This was implemented in the defense() function for HarryPotter. Medusa would use Glare when she rolled a 12. This ability was interesting in the testing stages as the chances of it occurring seemed quite rare. It took me about 50 simulations for Medusa to roll a 12 (a roll of 6 on both dies) and perform Glare. I outputted status messages to alert the user when these abilities occurred. I would definitely say that the hardest part of this project was implementing these special abilities and ensuring that they worked correctly through numerous tests. I also struggled to implement a way to take in a string for a dice roll such as “2d6” and break it up into two separate int variable for numOfDies and numOfSides. I eventually settled on the stoi() function which worked perfectly in my IDE but not on the Flip Servers as they lack support for C++ 11. I had to go from stoi to atoi and use a c-string conversion in order to correctly store my int values for numOfDies and numOfSides. I also ran into an issue where after running 1 battle iteration, my program would neglect to ask for user input after the user entered in 1 for play again. I noticed that I had to reset my character objects to NULL at the beginning of my game loop. I had into a few other issues in the process of making this Project but overcame them through debugging with cout statements and hours spent researching the problems online. Overall, I felt I learned a lot from this project and feel confident in my understanding of inheritance as well as polymorphism.

Character (base)

Barbarian   Vampire   Blue Men   Medusa   Harry Potter   (derived classes)

main.cpp

Test Table

Input	Location	Expected Result	Actual Result
Entered 1 for Play Again	Main.cpp	The game loop would loop back to the beginning and ask the user to choose a Character to Character1.	The program neglected to ask the user for input and proceeded to the game. I realized that I had to reset my character objects to NULL at the beginning of the loop to fix this.
Entered 2 to Quit	Main.cpp	The program would exit the game loop and the execute to completion.	The program successfully exited as I expected.
Entered a character 'a' when my program asked the user to enter in an int value from 1 to 5 (representing different character choices)	Main.cpp	My getInput() function would correctly verify that the userInput is out of range/incompatible and reprompt the user to enter a value from 1 to 5.	The getInput() function correctly determined that the 'a' character was invalid and prompted the user to enter an int value from 1 to 5.
Tested Harry Potter's Hogwarts ability in a battle with Blue Men	Main.cpp HarryPotter.cpp	If Harry's strength were to drop to 0 or below, he would revice and his strength would be at 20.	Harry's strength correctly went up to 20 after he lost his first live.
Placed cout << "Decrement Die" in the BlueMen defense() function for each time their defense die decremented	BlueMen.cpp Main.cpp	The program would correctly output "Decrement Die" when the Blue Men reached a strength of 8 as well as when they reached a strength of 4.	The program correctly outputted these status messages.
Hard Coded Medusa's attack roll to 12 to test the effect of Glare.	Medusa.cpp Main.cpp	Medusa rolling a 12 would kill the opponent enemy unless they were Harry Potter and he was on his 1st life.	Medusa correctly defeated each and every enemy. Harry Potter revived but lost on his 2nd life.