Arjun Kahlon

Project 4 Reflection

CS 162

This project has us build upon the classes we created in Project 3. In Project 3, we created a base Character class and 5 derived classes: Barbarian, Vampire, BlueMen, Medusa, HarryPotter. In that Project, I conducted a 1 on 1 battle in which a sole winner was declared before the program returned to the menu. Project 4 had us implement containers. In this case, I had two teams with user declared number of team members. I created a LinkedList class, from which I derived a Queue class and Stack class. In the main, created a two Queues to hold the team members, and a stack to hold the loser pile. In the beginning of the program, the user defines how many team members they would like per team. The program then loops to fill this team. New members are inserted to the back of the queue. I then implemented a battle() function which removes the Character being stored at the front each queue to battle each other (similar to Project 3's battle). The winner of this battle received a Recover boost which is randomly generated based on a percentage of their starting Health. This Recover function was implemented in the parent Character class and inherited into the derived classes. The winner is then stored at the back of the queue while the loser is pushed to the front of the stack. This game loop continues until one of the queues is empty. After the game loop ends, the program prompts the user to view the loser pile. If the user enters yes, the program displays the stack which is already ordered from most recent loser to first loser. The program then prompts the user on whether or not they want to play again. If they choose yes, the main loop reiterated. If they choose no, the program exits. Input verification is done in my getInput() function and ensures that the value the user enters is correct. Overall, I found this Project to be my favorite one thus far in this course. I loved that we got to build upon a previous Project. My understanding of Queue and Stack has grown immensely in the past week.

The user input in Project 4 was very limited and thus I felt there were few test cases I could experiment with.

Test Cases:

What I did	**Location**	**Expected**	**Actual**
Entered a negative number (-4) for number of fighters	Main.cpp -main() function -getInput() function	My getInput function would verify that the negative parameter is out of bounds and prompt the user to enter in a positive number	The program correctly determined that -4 was out of range and had the user reenter the number.
Enter 1 to start game loop	Main.cpp	The game loop would enter correctly	The program unexpectedly exited. I realized that I was incorrectly storing 1 into my bool play variable instead of the int playGame variable. This lead to me storing false as my Boolean value and the loop not entering. (Loop ran when play = true).
Entered 3 in playAgain prompt	Main.cpp	Program would correctly prompt user to enter a value of either 1 or 2	Program successfully prompted user to enter in a value of 1 or 2
Entered character name	Main.cpp	Loop would successfully iterate to the next character entry	A ‘\n\ was stored in the input buffer that ended up not allowing the user to choose a number between 1-5 for the next character entry. Had to switch to getlinee to fix this issue.