

Thanhbinh Truong

Project 4; truontha@oregonstate.edu

5/29/17

933233558

The design of this program was to implement program from assignment 3 and make it into a tournament. This is created by implementing the assignment 3 to be able to take in two users that are allowed to select creatures of their own to fight against each other. Each player's characters will fight against each other until one player is left as the winner. At the end of each round, the winner will be displayed on the screen. After all the rounds, each round will be tallied up and the screen will display the overall winner of the tournament.

The tournament class was designed to run the entire tournament instead of just a few rounds like program 3 did. This was done by implementing a few functions such as ones to simulate the battle and select the creatures for each team. The creature list class was added to create a list of each player's creatures. Some modifications were made from the old program. Modifications from the old programs were made such as remaking the attack function and defend function for all the creatures.

This project at first seemed to be a piece of cake since it was just an addition from program 3. By making all the creatures and a basic implementation of the game, I thought I was going to be able to go through this with ease. Unfortunately, I was wrong. The biggest obstacle that I ran into was being able to ask the users how many characters they wanted to have on each team. It was frustrating trying to figure this out because I wasn't able to figure out how to create a new instance of each character depending on how many the user wanted. I tried as best as I could but ended up having to allow only a fix number of 4 creatures to be made to make sure my program ran thoroughly. An obstacle that I faced was being able to stop the rounds after the necessary number of rounds were complete. For some reason during each test, the rounds would continue to run. This was very frustrating as I felt like I did everything right but the error would continue to show up. To fix problem, I began taking a closer look at all of the for loops and noticed that I made multiple errors in the loops which would create complications during the running of the program, ultimately resulting in the errors.

## Project 1

### - Creature "Combat Game Tournament"

↳ 2 players; 1 user

↳ ~~Each~~ player will get ~~to~~ to choose how many creatures they want each team to have.

↳ Each player will take turns choosing who they want on their team from a list of creatures.

### - Rounds:

- Each round, creatures will go up against each other

- The winner will return back in the regular line-up while the loser will join the loser's container.

- Winner/loser will be displayed at the end of each round.

\*\*\*TEST PLAN/RESULTS\*\*\*

Test Plan/Results

<u>Test</u>	<u>Expected</u>	<u>Results</u>
① Enter <del>name</del> name of creature (eg xan)	Put "name" wanted to creature	Stored name to the creature
② Enter choice outside of menu ( $1 > x < 5$ )	- cout "error"	- Displayed that user entered wrong selection.
③ Enter choice inside of menu ( $1 \leq x \leq 5$ )	- Selects creature	- Creature is selected
④ Enter illegal char	- cout "error"	- Displayed that user entered wrong selection
⑤ Run program through	- Winner p1 or p2	- p1 or p2 ended up winning