

## *PROJECT 3 – FANTASY COMBAT*

### **Program Design**

*Problem:* The problem to be solved is implementing a program of Fantasy Combat with Vampire, Barbarian, Blue Men, Medusa, and Harry Potter. Users will choose what characters they want to fight each other. One character will attack the other and fight until one of them dies.

#### *Structure:*

Main Function:

Introduce the game and call the menu function from the GAME class.

User will enter their choice. They will either view the rules and character attributes, start the game, or exit (number between 1 and 3).

showRules Function – shows rules for the game and implements a while loop so the user can cycle through each of the characters attributes.

getCharacter functions – asks each player to choose a character they want to play as

Game Class:

This will implement the main game functions to get characters to fight one another

Character Class:

This will implement getName, getArmor, getStrength, dead, data (constructor like function), defense, attack, and diceRoll functions for the other classes to inherit

Vampire Class:

Inherit all attributes from Character and adds special attribute CHARM (rand function). No damage was done. Print armor and strength, but do nothing (implement in attack function).

Barbarian Class:

No special ability, inherits attributes from Character class

BlueMen Class:

Inherit all attributes from Character and adds special attribute MOB. Strength is less than or equal to 8 and greater than or equal to 4, which starts the MOB function and then implements the MOB into diceRoll function.

Medusa Class:

Inherit all attributes from Character and adds special attribute GLARE. Sets damage to max value (implemented in attack function).

Harry Potter Class:

Inherits all attributes from Character and adds special attribute HOGWARTS. Restores Harry Potter's life and resets his strength to 20 (implemented in defense function).

**Reflection**

This was probably the most fun I've had in this class so far.

All functions appear to be working properly, but I started with an entirely different way to implementing the dice functions. I ended up adding different dice functions after the way I was initially writing them did not compile the function.

I mostly copied and pasted functions for each of the characters and then went through hours of debugging to try to figure out how to edit each one to get it to do what I wanted it to. Mostly, these functions stayed constant with a few edits here and there.

I initially no game class, but added this when the program was running really slow with all my other implementations in my main.