Assignment 3 Reflections, Design, and Test Plan Document

Student: Joaquin Saldana

Design

The purpose of the assignment is to create a program that implements the fundamentals of polymorphism.

As stated in the assignment, I created a base abstract class that is titled “Character”. This the Character class will be the base class for different types of fighters which will be used to simulate a fantasy combat game.

The following functions are base class functions which will be shared by all classes inheriting from the base class:

Private member variables:

1. Int damage – will hold the damage inflicted by the character that will be passed to the combatant
2. Int strength – this variable will hold the character’s overall strength initiated by the constructor. This value will change as the game progresses and until ultimately the character reaches a strength of 0 and is “defeated”
3. Int armor – initiated by the constructor. Every character will have their own respective armor which deduces the opponents attack. This value is static and never changes.

The damage formula:

Figher 1 Damage = (Fighter 1 Defense Roll – Figher 2 Attack) – Figher 1 Armor

Public Functions:

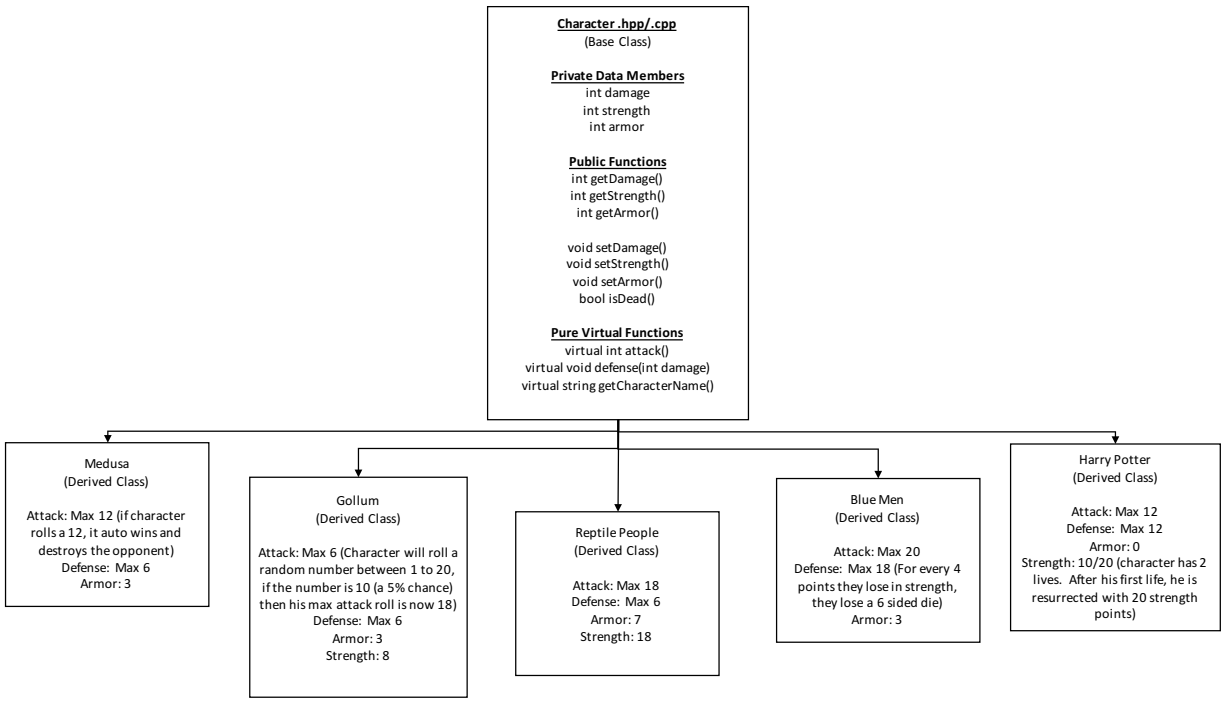
1. Standard set and get methods
    a. getDamage()
    b. getStrength()
    c. getArmor()
    d. setDamage()
    e. setStrength()
    f. setArmor()
2. bool isDead() – function returns true if the fighters strength has reached 0 or less

Public Pure Virtual Functions:

1. virtual int attack() = returns an int which is produced based on the character’s max attack.

2. Virtual void defense(int damage) = function takes an int as a parameter which is the opponents total attack. From there the character's defense generates a number which is applied to the damage formula which calculates how effective the opponent's attack was
3. Virtual string getCharacterName() = return the character's name which is a string

Below is a diagram which reflects the inheritance hierarchy:



Below is the list of the derived classes which will inherit from the character class:

1. Medusa
    a. Attack: max 12 (if character rolls a 12, it auto wins and destroys the opponent
    b. Defense: max 6
    c. Armor: 3
    d. Strength: 8
2. Gollum
    a. Attack: max 6 (Character will roll a random number between 1 to 20. If the number rolls is 10 (a 5% chance) then his max attack roll is now 18
    b. Defense: max 6
    c. Armor: 3
    d. Strength: 8
3. Reptile People
    a. Attack: max 18
    b. Defense: max 6

    c. Armor: 7
    d. Strength: 18
4. Blue Men
    a. Attack: max 20
    b. Defense: max 18 (for every 4 points they lose in strength, they lose a 6 sided die)
    c. Armor: 3
    d. Strength: 12
5. Harry Potter
    a. Attack: max 12
    b. Defense: max 12
    c. Armor: 0
    d. Strength: 10/20 pts (character has 2 lives. After his first life, he is resurrected with 20 strength points)

Main function will introduce the user to the program and explain in detail what it will ask of them. User will be prompted to choose who he wishes to fight. The program will then initiate a loop detailing the rounds, attack, and defense, and ultimate winner.

The program will continue until the user enters the number 0 to exit the program.

Also, I created a Dice class to simulate the roll of a six and ten sided dice. The Dice object was created as a public member of the character class so all derived classes will have access to it in order to use it for their attack and defense functions.

The combat will continue until one of the character's isDead() function returns a value of true. In essence, the rounds can extend past the conventional 3 to 5 rounds in a combat figher. However they should not exceed anything longer than a few double digit rounds and that itself should be rare. Most should end in the single round numbers.

Reflections

1. I originally was creating a Dice object in each of the attack and defense functions of the character classes and although this worked I found a better solution to this.
    a. I created a Dice object in the character header file and declared it public so all classes derived from the Character class will have access to it.
        i. The dice object has two functions
            1. One to simulate the roll of a 6 sided die
            2. The other to simulate the roll of a 10 sided die
2. In an effort to help me identify which character is fighting whom, and for the purpose of console output, I also included a pure virtual function of type string to identity the character's name.
    a. The function returns a string and was defined as a pure virtual function in the abstract class of Character.
    b. In each character derived class that follows I ensured the function was implemented and returned a string to represent the Character's name type ex: "Blue Men", "Reptile People", and more
3. For the Medusa character was trying to figure out how to allow her to auto win. I decided if she rolls a 12, then her attack function will return a very large number ensuring the opponents defense is depleted.
    a. In the scenario she faces a character like Harry Potter, I decided on letting Harry revive and continue fighting and not allowing Medusa to automatically WIN.
        i. I came to this conclusion because allowing for the match to discontinue is an automatic disadvantage to the Harry Potter character who's special attack is revival to 20 points in defense.
            1. And allowing another character to disable or serve another character's special attack ineffective is unfair. I believe each character should be allowed to use their own special attack if applicable.
        ii. In addition, there's nothing from keeping Medusa rolling another 12 in the 2nd life of the Harry Potter character meaning she will win automatically again.
4. When testing the program I noticed at times certain opponents strength were not depleting. I double checked the formula and found some the opponent characters were not generating a strong enough attack as a result the fighters strength went unaffected.
    a. To visually see when the defense went unaffected I placed a else statement to print when the fighter's strength went unaffected.
    b. Also noticed the Medusa character was the stronger character of the others when compared to winning.
5. Noticed there was the possibility that a draw can occur if both character's return an attack that is equally devastating to the other opponent.
    a. Created a Boolean function at the end of the while statement that verifies if both characters are dead in which case the program outputs a cout statement notifying the user of the draw.

TEST PLAN

Test Case	Input Values	Driver Functions	Expected Outcome	Observed Outcomes
Ensuring the attack and defensive functions are returning random number	Cout stmts in main and	Cout stmts in main	Random number's for each function on each test run (more than 1)	Various random number's within their respectable number value range. Everything worked fine
Validating the special attacks for each character is properly working. This includes the Medusa SA, Gollum SA, and Blue Men and Harry Potter Strength SA.	Multiple selections of the characters, and especially choose character's facing an opponent of the same character	Cout statement. When the Special attack was activated I created print statements in the attack and defensive functions to notify the user the SA has been activated	For console to notify the user when the special attack was activated	Worked correctly with the Harry Potter, Blue Men, Gollum, and Medusa characters.
Ensure the code is validating user input when choosing which characters	While statements and my own utility class functions	While statements and prompts	The function to properly detect when the user is entering a string of chars or a number outside the number range (1 through 5)	The function properly detected and informed the user of the error when the user entered the incorrect value
Characters vs the same character. Checking to see if there's any infinite loop's that need to be addressed	Choose the same character's against each other Gollum v Gollum, Reptile People vs. Reptile People, etc.	While stmts with terminating conditions	For the combat to end but with higher number rounds	No infinite loops were found and the number of rounds was unpredictable but not long.

Since the program only terminates after the user enters 0, checking for memory leaks when each pointer is “deleted”	Using pointers to figher 1 and fighter 2 in the main function	Function from my utility class, and while statements	For the program to terminate with no memory leaks	Program ran correctly, prompting the user at the end of each combat to enter 0 or any other number to continue. After testing several combat simulations, anywhere from 5 to 10 combats between multiple character’s, the valgrind program found no memory leaks and all dynamic memory was properly deallocated