Assignment 3(Reflection Document)

Woohyuk Yang

yangwo@oregonstate.edu

1. Requirements:

The requirement is to make a creature combat program. I have to make at least 6 classes; one is Creature class(base class) and the others are Medusa, Barbarian, Baba Yaga, Blue men, Harry Potter(derived classes).

Each type of creatures have their own ways of attack, and defense, and features like Armor and Strength Points as common. But the values of those common features are different from characters.

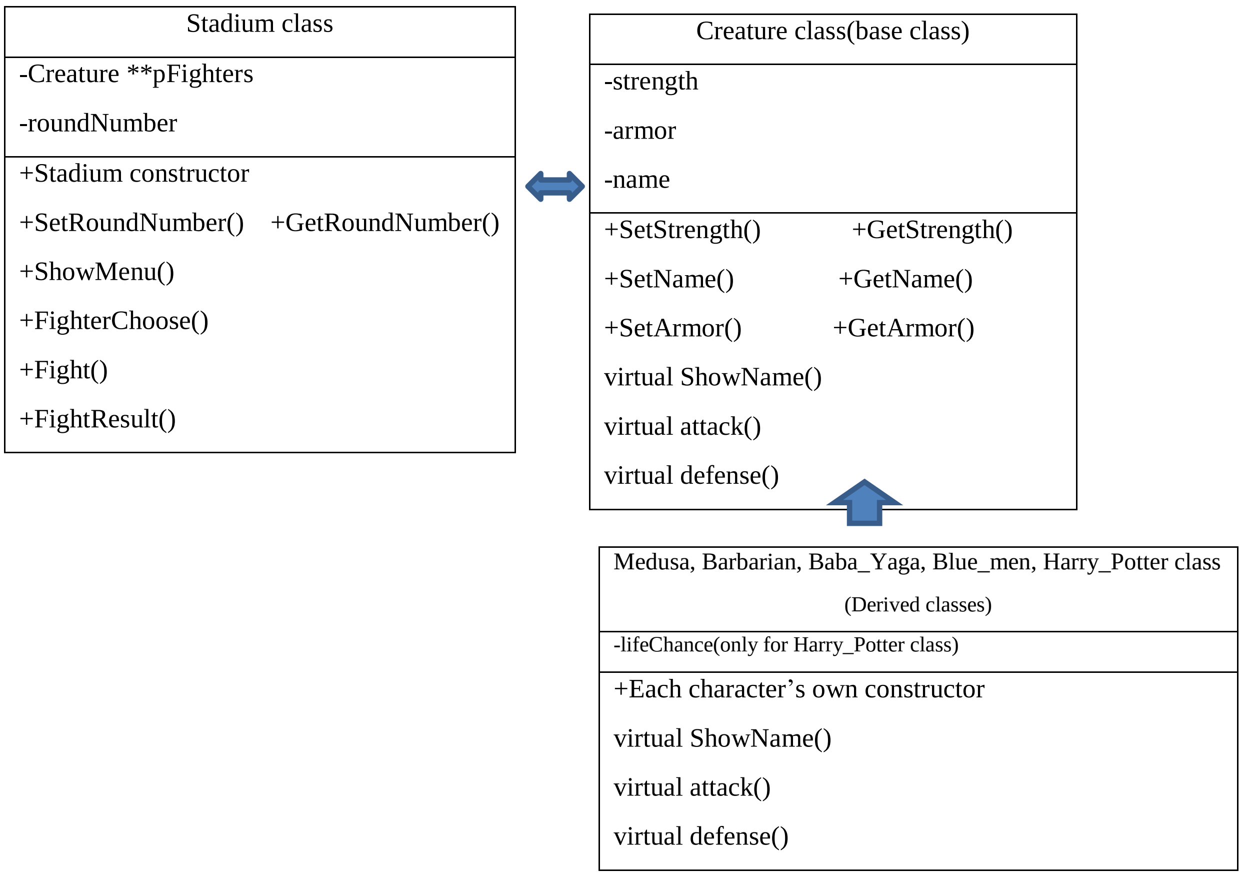
Type	Attack	Defense	Armor	Strength Points
Medusa	2d6* Glare	1d6	3	8
Barbarian	2d6	2d6	0	12
Baba Yaga	2d6*Soul	1d10	3	12
Blue men	2d10	3d6*	3	12*Mob
Harry Potter	2d6	2d6	0	10/20* Hogwarts

*3d6 is rolling three 6-sided dices. 2d10 is rolling two 10-sided dices.

Each creatures have their own way of attack as it is described above. Glare means when medusa rolls a 12 in attack then the target turns in to a stone points and medusa wins over the fight. Soul means when Baba yaga attacks the opponent then it receives 1/3 of attack points she inflicted(rounded down) as an additional Strength Points. Mob means every 4 points of damages blue men loose one defense dice. For example when the blue men reaches Strength Points of 8 then it only has 2d6 as its defense tool. Hogwarts lets Harry Potter revive one time. Once harry revives then he gets 20 as his Strength Points.

Once the classes are made, the deriving program has to let user choose two characters to fight and let them fight displaying their attack points and defense points on the screen also with two creature’s strength point.

2. Class Design

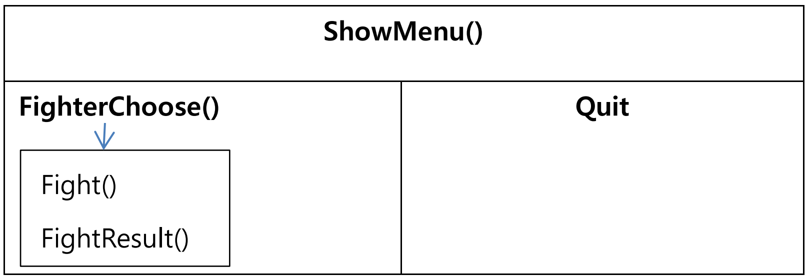


Creature class has some features like strength Points, armor, name. So other derived classes can inherit those features. This base class also contains some functions as member.

The class has accessor and mutation functions for strength and name variables. And ShowName() function which shows the name of each creatures.(access function for variable name) To use polymorphic features of C++ language, attack() function and defense() function were designed to be virtual functions. So even from derived classes like Medusa, Barbarian, Baba_Yaga, Blue_men, Harry_Potter classes can attack or defense in their own ways as the program is required.

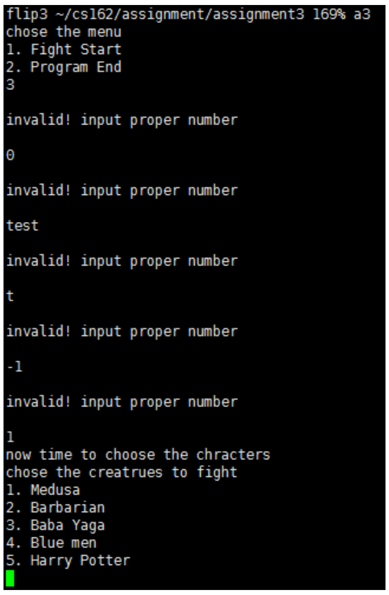
Stadium class is the class which allows two chosen creatures fight each other. They are composed of functional functions deriving fight simulation. In its constructor it makes an array of pointers to Creature class objects. To support that, the class needs double pointer variables for Creature class. The class member functions are composed of 4 functions. The first one is ShowMenu() function. This function lets user to start the fight or end the program. This will display the first show menu for the user. The second one is FighterChoose() function. This comes up once a user choose "fight start" menu before. This lets user to pick two creatures to fight. The third one is Fight() function which makes two creatures fight till one of them wins or tie. The last function FightResult() function is the function which decides who won over whom.

3. Test Plan

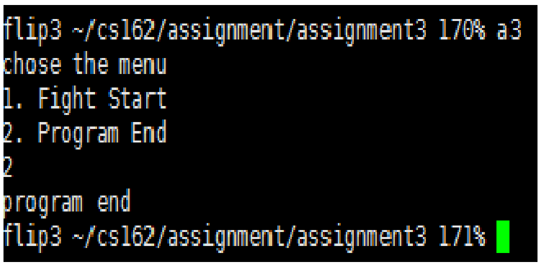


I illustrated the structure of the program to test the program how exactly it works with my intention.

When the program starts the user will see some menus he/she can choose the number. The user is only supposed to input ‘1’ or ‘2’ because ‘1’ is for staring fight and ‘2’ is to end the program. they type characters or invalid number then the program could crash. So I put validating process. As results below, the validating process worked fine and once I input ‘1’ then I can choose the characters and if I input ‘2’ then I can end the game.

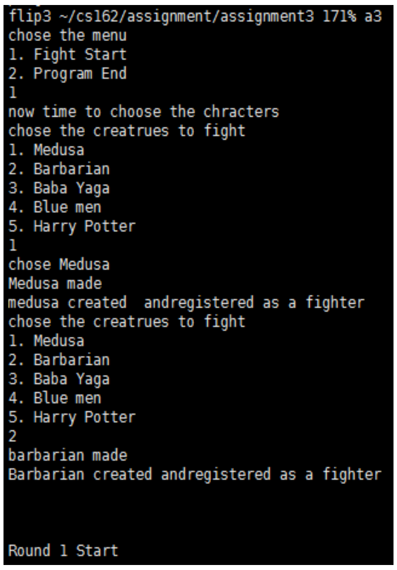


```
flip3 ~/cs162/assignment/assignment3 169% a3
chose the menu
1. Fight Start
2. Program End
3

invalid! input proper number

0

invalid! input proper number

test

invalid! input proper number

t

invalid! input proper number

-1

invalid! input proper number

1
now time to choose the chracters
chose the creatrues to fight
1. Medusa
2. Barbarian
3. Baba Yaga
4. Blue men
5. Harry Potter
```



```
flip3 ~/cs162/assignment/assignment3 170% a3
chose the menu
1. Fight Start
2. Program End
2
program end
flip3 ~/cs162/assignment/assignment3 171%
```

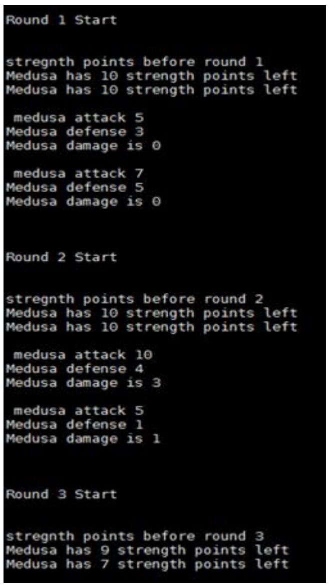
firstly when I input “1” and want to start the fight then I have to choose two characters. I tested this and as the picture below I can choose two characters and when two characters were selected the fight did start. I put my same validating process for numbers here also. The user can only type ‘1’ to ‘5’ for this section. And I tested whether user can pick same creatures twice as I intended, and it was able also.



```
flip3 ~/cs162/assignment/assignment3 171% a3
chose the menu
1. Fight Start
2. Program End
1
now time to choose the chracters
chose the creatrues to fight
1. Medusa
2. Barbarian
3. Baba Yaga
4. Blue men
5. Harry Potter
1
chose Medusa
Medusa made
medusa created  andregistered as a fighter
chose the creatrues to fight
1. Medusa
2. Barbarian
3. Baba Yaga
4. Blue men
5. Harry Potter
2
barbarian made
Barbarian created andregistered as a fighter


Round 1 Start
```

I tested every cases when it comes to fight. There are 15 cases and I made it in a table. Some areas are black colored because there is same test plan for that.

VS.	**Medusa**	**Barbarian**	**Baba Yaga**	**Blue men**	**Harry Potter**
Medusa	success	success	success	success	success
Barbarian		success	success	success	success
Baba Yaga			success	success	success
Blue men				success	success
Harry Potter					success



```
Round 1 Start

stregnth points before round 1
Medusa has 10 strength points left
Medusa has 10 strength points left

 medusa attack 5
Medusa defense 3
Medusa damage is 0

 medusa attack 7
Medusa defense 5
Medusa damage is 0


Round 2 Start

stregnth points before round 2
Medusa has 10 strength points left
Medusa has 10 strength points left

 medusa attack 10
Medusa defense 4
Medusa damage is 3

 medusa attack 5
Medusa defense 1
Medusa damage is 1


Round 3 Start

stregnth points before round 3
Medusa has 9 strength points left
Medusa has 7 strength points left
```

The picture on the left is when I tested my first test case

Medusa vs. Medusa.

Medusa defense is randomly number of sum of dice rolled. And damage is the actual damage concerned with armor(each creature's own feature.)

I differentiated defense and armor to show that the Fight() function does good process with attacking and defensing and renewing Strength Points for each creature after each round.

I checked each cases do randomly result attack points and defense points, and calculate those well with armor and left Strength Points inside the function.

I had to check each character's special features. For medusa's glare case, I put integer '12' directly to the variable representing the attack point and tested whether it works and it worked well. For Baba yaga's soul, I checked the value is well calculated and I also tested if fighter's strength Points go over "certain point" then it makes the match draw. And if the round number goes over " certain number" then it makes the match draw also. I set certain number like 10 instead of 100 only for the test. The matter is the process works well not the number itself. Because there is no change of function itself based on number, so if it works with small number then it would work well with bigger number. For blue men's Mob feature, I put some "cout" statement inside each if statement to check on which condition the function works with. I could check it works fine following its strength points also. Harry's Hogwarts feature could be checked when I first set 0 to harry's strength point so harry potter revives again and used GetStrength function to see if harry's strength points has been assigned properly.

For FightResult() function, I checked whether each cases work properly with GetStrength() function. Threre are 6 cases here. The first is when both creatures dies, the second is the first creature dies and loose, the third is the second creature dies and loose, the fourth is both do not die, the fifth is strength points of both creatures go over certain number(100 in this program) so it makes the game draw. And the last one is the round number goes over certain number(100 in this program) so it makes the game draw. For the last two cases, I set smaller numbers to each to verify it works properly more easily but still accurately. The result was all worked successfully.

4. Reflection

I changed my design doing my implementation.

At first, I did not have any plan to make stadium class at all. I tried to call functions from main function. But to make my program more object oriented and efficient in the end. I invented to make another class dealing with creature's desired action. So I made stadium class. Like creatures can really fight in real life at stadium.

Stadium function was firstly made of two functions; ShowMenu() and Fight() for barbarian class. Because Barbarian class only required simple logic to implement. The first two functions were enough and I assumed I can make other classes work in desired way with only two functions. But as other classes needed to add more complex logics, I needed more functions to deal with those. I newly made FightResult() function to determine who has won and to display that on screen. In the end, added another function called FighterChoose(). Fight() used to contain menu for user to choose the creatures to fight but that made one function too long. Also, choosing fighters is totally different process from "fight" so I made another function and made Fight() function shorter and concise with better clarity.

After stadium class was made. I dynamically allocated every creature objects first. And made Fighters to make fighters wait before a fight. But I corrected to make Fighter directly not needing to allocating every creatures at first. So whenever a user chooses a creature to fight then the object of that very creature gets dynamically allocated to make my dynamically allocated memories lighter.

In real fight or sports, it is important who attacks first. At first I designed Fight() function to prompt user to pick one to attack first . But this could be unfair and I have ever played game before which two players can attack and defense at the same time before. So I implemented two creatures attack and defense in one round. So "A" creature attacks and "B" defenses, and "B" creature attacks and "A" creature defenses. Those actions consist of one round. The fight results comes after each round.

Baba yaga can increase her strength point as she keeps attack, and baba yaga vs baba yaga combat could cause infinite loop so I set if both of character's strength points go above 100 then I made them draw and end the match. And I added a variable which can indicate the round number(the fight is composed of "round" as I described above). Like every sports have limitation I set it to have limitation as well. I thought 100 round is enough for both fighters and watchers.

Dealing with medusa's glare, I set the game itself limited, I set medusa's glare attack get to be 1000, which means she makes the other opponent a stone. So limiting the round of fight can work with implementation of "Glare" like the way I mentioned. I made Hogwarts even could help harry even if medusa makes him a stone.