Sangyun Lee

Leesa6@oregonstate.edu

Nov 05 2019

Project 4

About - Modify a fantasy combat game using Linked List, Classes and Inheritance (polymorphism). This game allows users select characters between 2 and 5 for a team. And Team of selected characters will fight each other. Winner of the battle will be added to the team again for later battle. Losers will be added to loser section. At the end, users can see who are losers of each team.

Requirements

- Get users' response and validate it
- Created linked list to play game as tournament
- Allow users to set the name for characters
- Recovery skill end of the battle
- Calculates the points of each match game
- Check result

Challenges

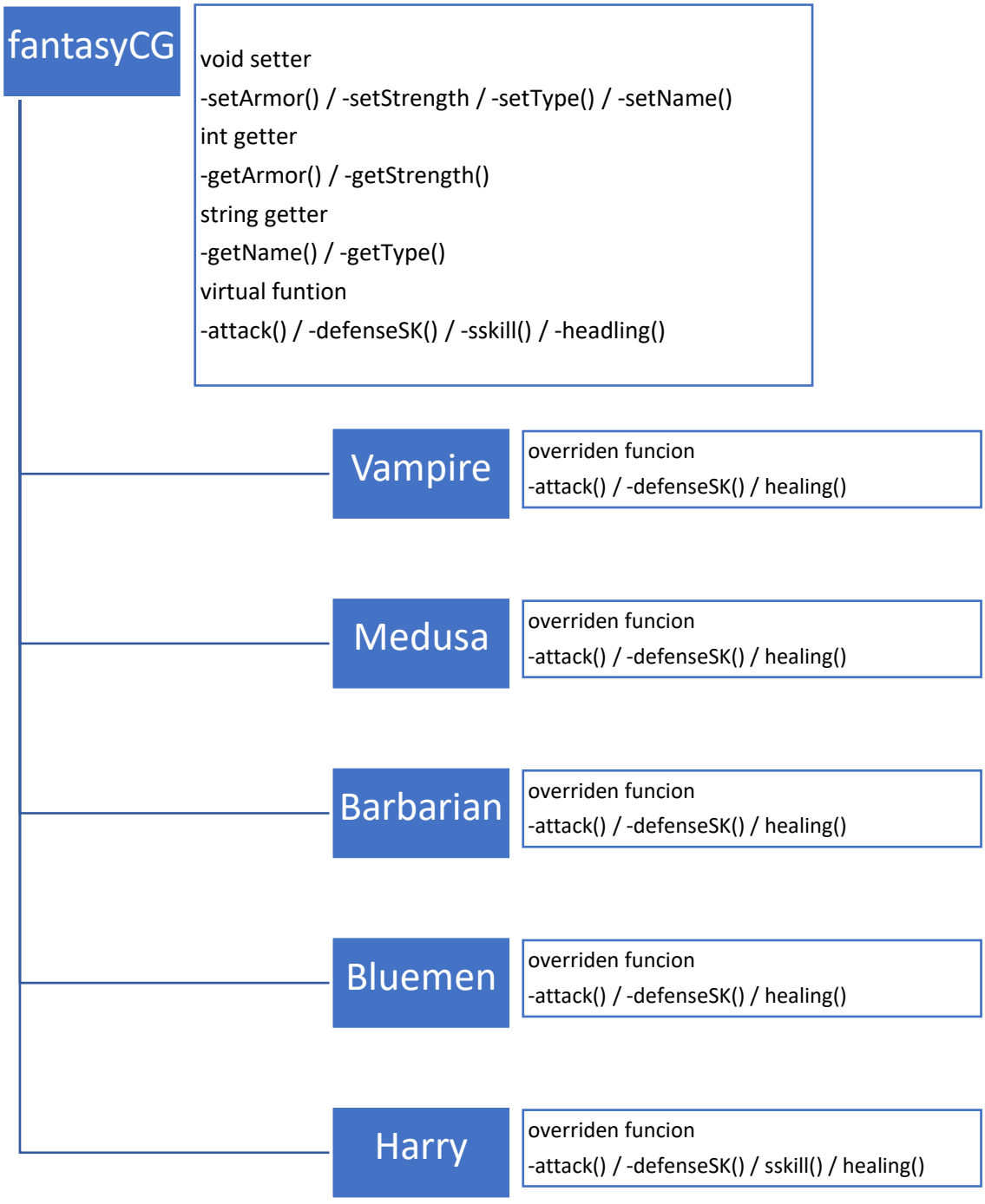
- How to create linked list for tournaments play
- How to properly implement recovery skill, and prompt result

How to solve the requirement and challenges

- How to create linked list for tournaments play
    - This is very similar to the lab I had done during week 6 and 7. I simply created structure with fantasyCG class which is parent class.
        - Struct Node {fantasyCG* variable name; Node* next} Node* head;
        - And created functions that can help to control the Linked list

        - It doesn't have to be doubly linked list which was requirement of lab, so I created singly linked list. It was much easier to link node

- How to properly implement recovery skill, and prompt result
    - Character name
        - Get user response using getline(std::cin, variable)
        - And initiate character class with user response as parameter
    - Recovery skill
        - I add another virtual function that implements the recovery skill. Also code it with srand(time(null)) and rand() % … in order to give randomness to the function. I split it into 4 categories. No healing portion, small portion, medium portion, large portion. And those have different healing effects.
    - prompt result
        - Since it is tournament, I will create the containers that informs the number of match. And prompts at the end with team points and winner. So, users are able to see the final result


fantasyCG
void setter
-setArmor() / -setStrength / -setType() / -setName()
int getter
-getArmor() / -getStrength()
string getter
-getName() / -getType()
virtual funtion
-attack() / -defenseSK() / -sskill() / -headling()
Vampire
overriden funcion
-attack() / -defenseSK() / healing()
Medusa
overriden funcion
-attack() / -defenseSK() / healing()
Barbarian
overriden funcion
-attack() / -defenseSK() / healing()
Bluemen
overriden funcion
-attack() / -defenseSK() / healing()
Harry
overriden funcion
-attack() / -defenseSK() / sskill() / healing()

Test Table – (I tested only new functions that added for project 4)

1. addPlayer() & disPlay() – I wanted to see whether the selected characters were added on list well or not. So, once I added all characters and display it with display()

Test Case	Input	Driver Func	Expected Output	Actual Output
When add 3 fighters	Vampire as vam, barbarian as bar, Harry porter as harry		vam(Vampire), bar(Barbarian), harry(Harry Potter)	Matches expected result
When add 2 fighters	Medusa as meme, Blue Man as bbl		meme(Medusa), bbl(Blue Man)	Matches expected result
When add 4 fighters	Barbarian as bbb, …… Vampire as vim		Display all characters like above	Matches expected result

2. getPlayer() / attack() / defenseSK() – get a fighter from listed link, and test whether It should generate random number for attack points and defense points in right range for the selected fighter.

Test Case	Input	Driver Func	Expected Output	Actual Output
Vampire	No input	Fighter->attack()	Random number between 1 ~ 12	Result within range
Harry	No input	Fighter->attack()	Random number between 2 ~ 12	Result within range
Vampire	No input	Fighter->defenseSK()	Random number between 1 ~ 12 or Charm	Result within range
Medusa	No input	Fighter->defenseSK()	Random number between 1 ~ 6	Result within range

3. healing() – Special ability to recover fighter's strength after battle ends

Test Case	Input	Driver Func	Expected Output	Actual Output
Harry potter lose		Function should not be triggered	No change	Matches expected result
Harry potter with str in 9	Intentionally recover her strength more than max and no Hogwarts used		Strength will be str 10	Matches expected result
Medusa win with str as 3		healing()	Strength will be recovered random points between 1 and 3	Result with in expected range
Medusa win with str as 7	Intentionally recover her strength more than max	healing()	Strength should 8 which is max strength of her	Matches expected result

4. menu(oneMore) – ask users to see that they want to play more game or not

Test Case	Input	Driver Func	Expected Output	Actual Output
Correct	1	Menu() & getInput()	"select a characterfor …"	Matches expected result
Correct	2	Menu() & getInput()	Display losers of each team and ask another game	Matches expected result
Correct	3	Menu() & getInput()	Program close	Matches expected result
Wrong	j	Menu() & getInput()	"Wrong input please…"	Matches expected result

Reflection

Throughout this project, I understood better about pointer and how strong pointer is. At first, I got confused how I can implement tournaments using linked list. However, once I understood pointer as just another data type that has address, I was able to handle the project 4 with the knowledge I learned from week 6 and week 7. At first, I was think about creating setter for characters' name, but, instead, I passed the user decided name as argument when class object is created. And return the name later with getter function.

When I designed the list or queue, I spent most of time to decide which linked lists I should use. Eventually, I used singly linked list and I create a new node with the same data for fighter and loser containers which is less elegant way.

One thing I had to pay attention was to tracking the variable name. Most of variables are pointers, so it was easy to make a mistake when I implemented the attack or defense even I made the variable names distinguishable for two teams. So, I had to double check it before move to the next steps.