CS1010E Programming Methodology

Semester 1 2020/2021

Assignment 07: Object Oriented Programming

Required Files

- `battle.py`
- `characters.py`
- `team.py`

Notes

- No library has been imported for your, if you need any additional standard library, feel free to add them yourself (*rejoice! there's no more turtle*)
- You are **NOT** allowed to change the input parameters to the functions (e.g., *number of parameters*).
- You are **NOT** allowed to change the function name.
- You may reuse the code written in earlier sub-task for your *subsequent* sub-tasks.
    - You may assume that the correct implementation is given.
    - If you wish to override the standard implementation of earlier sub-task, please copy the code for your earlier sub-task.
- Please remember to read through the section on **Additional Information** before you start working on the assignment.

Role Playing Game

There are lots of RPG games nowadays like Final Fantasy, World of Warcraft, or DOTA. In this assignment, you are going to implement the characters for turn-based RPG.

The game is a fight between two teams of adventurers formed by different types of characters (called "classes" but do not confuse these job classes with Python class, so to avoid confusion we will use types of characters). You are given a battle system that can let two teams fight against each other until one team wins by destroying all the characters in the other team. The good thing is that the battle system is already implemented for you. You only need to implement some new types of characters. You can choose to run the file `battle.py` to start the game and try it out.

Game System

- You are given some amount of gold. The default setting is 200 gold in the skeleton file.
- Every type of character has a cost, and you can recruit a team with any combination of them within your budget. Each team has to use up all the gold to form a team.
- A random team formed by `create_rand_team()` will create a team as your enemy. Then you can recruit your own team by choosing available types of characters by the function `user_choose_team()`. In the battle, you will be Team A and your enemy will be Team B.

Battle System

- Once the battle starts, each team will take turns to act. However, **only one character can perform an action for one turn for the entire team**. A character will be chosen at **random** within those that are still alive in the team and he will perform an action targeted at a **random** enemy that is alive selected using `rand_alive()`. However, the last class Necromancer has an ability to "target" your own team as well.
- So when a character acts, he will hurt a targeted enemy and the enemy will be hurt via `got_hurt()`. When the targeted enemy's hp (*hitpoint*) drops to zero, he will be dead and no longer acts … unless a Necromancer revives him afterwards.

A typical start of the battle is shown below. The file `sample.txt` stores the log of a full battle for your reference.

```
THE BATTLE STARTS!!!!!

Round 1
Team A:
Members:   |    Fighter|       Mage|Necromancer|    Fighter
Hitpoints: |       1200|        800|        800|       1200
Strength:  |        100|           |           |        100
Max. Mana: |           |         50|         50|
Currnet M: |           |         50|         50|

Team B:
Members:   |   ArchMage|  Berserker
Hitpoints: |        800|       1200
Strength:  |           |        100
Max. Mana: |         50|
Currnet M: |         50|

Team A member 0 Fighter acts
Hurt enemy 1 by damage 100.
Berserker hurt with remaining hp 1100.

Team A:
Members:   |    Fighter|       Mage|Necromancer|    Fighter
Hitpoints: |       1200|        800|        800|       1200
Strength:  |        100|           |           |        100
```

```
Max. Mana: |           |         50|         50|
Currnet M: |           |         50|         50|

Team B:
Members:   |   ArchMage|  Berserker
Hitpoints: |        800|       1100
Strength:  |           |        100
Max. Mana: |         50|
Currnet M: |         50|

Team B member 0 ArchMage acts
Strike enemy 3 with spell
Fighter hurt with remaining hp 800.

Round 2
Team A:
Members:   |    Fighter|       Mage|Necromancer|    Fighter
Hitpoints: |       1200|        800|        800|        800
Strength:  |        100|           |           |        100
Max. Mana: |           |         50|         50|
Currnet M: |           |         50|         50|

Team B:
Members:   |   ArchMage|  Berserker
Hitpoints: |        800|       1100
Strength:  |           |        100
Max. Mana: |         50|
Currnet M: |         30|

Team A member 0 Fighter acts
Hurt enemy 0 by damage 100.
ArchMage hurt with remaining hp 700.
```
(and the battle goes on….)

Task 1: Character Classes

We will have five different character classes, namely, `Fighter`, `Mage`, `Berserker`, `ArchMage`, and `Necromancer`. The first two classes are already implemented for you. The last three classes are to be implemented using the following idea:

- You are to use inheritance whenever possible.
- You are not to modify `Fighter` and `Mage` class.
- You should not override a method from superclass unless necessary.
- You should call superclass methods whenever possible instead of having your own implementation at sub-class.

Fighter

A `Fighter` has a maximum hp of 1200 and a strength of 100. He has a cost of 100 gold. During each turn in the battle, he will select a random enemy and inflict a damage equal to his strength (i.e., 100 hp).

Mage

A `Mage` has a maximum hp of 800, a mana of 50, and an intelligence of 400. During his turn, he will cast a spell and make a damage that is equal to his intelligence. However, casting a spell costs 20 mana. If his remaining mana is lower than that, he will take the turns to meditate and gain a recovery of 30 mana instead. In other words, he will not cast a spell to hurt an opponent during meditation. He has a cost of 200 gold.

Task 1A: Berserker

A `Berserker` is a `Fighter` but he costs 200 gold. If his hp is lower than or equal to half of his maximum hp, he will enter the "*berserk mode*". This doubles his strength to 200.

Sample Output

```
Team B member 1 Berserker acts
Berserk mode! Attack double!
Hurt enemy 3 by damage 200.
Fighter hurt with remaining hp 500.
```

Task 1B: ArchMage

An `ArchMage` is a `Mage` but he costs 600 gold. If he is the *only one alive* in his own team, he will cast the special spell `KABOOM` and it inflicts damage that is double of his intelligence on EVERY enemy alive. This `KABOOM` spell also requires 20 mana to cast.

Sample Output

```
Team A member 1 ArchMage acts
Cast KABOOM to every enemy!
Fighter died!
Fighter died!
```

Task 1C: Necromancer

A Necromancer is a `Mage` but he costs 400 gold. If there are some dead members in his own team, he can cast a spell `RAISE DEAD` costing 20 mana to revive a random dead team member and recover half of the revived member maximum hp rounded down instead of hurting his enemy this turn.

Sample Output

```
Team A member 1 Necromancer acts
Reviving member 0 with 400 hp
```

Files

`characters.py`

This is the main file where you should add your code. The two base types of characters (`Fighter` and `Mage`) are already implemented for you as `Fighter` class and `Mage` class respectively. Both are sub-classes of `Character` class.

There is a function `dprint()` defined at the top. We use this instead of `print()` to have a controlled printing (called debug print, or dprint for short). To turn the `dprint()` on and off, simply set `DEBUG_PRINT` global variable to `True` and `False` respectively.

`battle.py`

This file contains the "game engine". The whole system is already implemented for you and you do not have to make any big changes. The changes can be summarized below:

- Increase the value of `NUM_CHAR` to the number of types of characters already implemented.
- Uncomment the implemented type of characters within the function `create_char` and `user_choose_team`. It should have a preamble of `# UNCOMMENT ONCE YOU HAVE IMPLEMENTED THE NECESSARY CLASS`.
- Your game should work if you can use more than 600 gold.
- The second argument to the function `game_start` may be changed to True to allow for a pause after every turn. Otherwise, the engine will run to completion.

`team.py`

You should NOT change this file. You may read the implementation as it may contain useful helper functions.

Additional Information

Can I choose to redefine cast() instead of just inheriting it from the parent class?

You will be marked based on how well you grasp the concepts of OOP, including (but not limited to):

1) Inheriting from the correct class
2) Using super().__init__(), super().act(), super().cast() as extensively as possible to reduce redundancy in your code
3) Not defining redundant methods or attributes because you have inherited them from the parent class
4) Not hardcoding

In other words, even though you passed all the test cases, **you may be penalised** for some sections because you copied and pasted code from parent classes instead of using inheritance.

Can a necromancer revive another mage?

Yes. In fact, a necromancer can revive another necromancer. If the enemy team has two necromancers, god help us all. As for how much health the mage (archmage or necromancer) is revived with, the PDF stated that all mages have 800 HP, so they are revived with 400 HP.

Will the necromancer always cast revive?

If there is a dead member in the team, the necromancer will always try to revive a random dead member. This is unless the necromancer is out of MP, in which the necromancer must meditate to restore MP. The necromancer will only cast his / her normal attack when there are no dead members and there is sufficient mana to do so.

Does the berserker's attack go from 100 -> 200 -> 400 -> 800 or just 100 -> 200 -> 200 -> 200?

It will just go from 100 to 200. However, the statement "Berserk mode! Attack double!" will still be printed for every turn that the berserker's health is below 50%.

Will I lose marks if I add an extra space to the print statement?

If you deviate from the expected output, we will not penalise you for it. We are marking for your logic. The test cases will never check on the print. You can print any additional information you need. We will focus more on the inheritance and good practice part rather than what to print.

Does the archmage seriously cast "KABOOM" for every turn as long as she's the only one alive in the team?

Yes.

What if the archmage is the only team member, i.e. you formed a team with only one member?

Yep.

When the necromancer revives the teammate, does the teammate's mana also return to full?

If something is ambiguous and not in the project specifications, you can come up with your own strategy. We will not penalise you for not following anything that's not in the specifications.

Can I create my own definitions instead of using cast? For example for my archmage i will use a function death_to_all as the function to cast kaboom spell and for necromancer I will use the function i_am_here_to_save_you to cast the reviving spell.

Do not rename cast. You may write helper functions that can be called within cast.