CS205 C/C++ Program Design Assignment4

Introduction

In this assignment, we will use object oriented programming (oop) to model some problems.

Considering you're going to do a project, this assignment has only one task, so it won't take you too much time.

Here is the task:

Tasks

Card Game

In this assignment, we won't describe each task step by step in detail, you are expected to understand the problems by relating them to real life experience, codes written in oop are very self-explanatory and well-divided so you can even code directly without looking at the documentation sometimes.

Card Game

Codes of this task are provided in `assignment4/cardgame`

Let's implement a card game. There are two players. Each player has a hand of cards and a deck, and at the start of each round, each player draws a random card from their deck. Cards have a name, an attack value, and a defense value. Each round, each player chooses one card to play from their own hands. The card with the higher *power* wins the round. Each played card's power value is calculated as follows:

```
1  (player card's attack) - (opponent card's defense) / 2
```

A card may have an effect, which allows the card to make changes to the opponent's card, player's deck, opponent's deck, … and can be defined to be imaginative.

Effect example:

- reduce opponent card's attack and defense
- let the opponent discard three cards from hand
- let the player draw three cards from deck

Step1 Make Card

To play a card game, we're going to need to have cards, so let's make some! We're gonna implement the basics of the `{Card}` class first. The definition is in `card.h`, For convenience, all members are public. Here are the explanation of members:

- *name* - name of the card
- *attack* - attack value of the card
- *defense* - defense value of the card
- *Card* - Constructor
- *power* - return the power value
- *effect* - the effect of the card, a card has no effect by default
- *operator«* - format card in format `<name> <attack> <defense`

implement them in `card.cpp`

Step2 Make Player

Now that we have cards, we can make a deck, but we still need players to actually use them. We'll now implement `{Player}` class. The definition is in `player.h`, For convenience, all members are public. Here are the explanation of members:

- *deck* - the deck of the player, which is a card vector, the back of the vector is the top of the deck, player draws card from the top of the deck. We make sure that deck always have enough cards to draw.
- *name* - name of the player
- *hand* - cards in player's hand, players draw cards from deck to hand
- *Player* - constructor, when a player is initialized, he will draw 5 cards from deck
- *draw* - draw a card from deck to hand
- *play* - return the card in hand of given index, we make sure the index is in range
- *displayHand* - display cards in hand, one line for each card

implement them in `player.cpp`

Step3 Card Effects

We know that card has an effect but no effect by default. Let's make some cards with powerful effect.

public derive`{Card}` class and override *effect* method to make these cards, these two cards should have a constructor using *name attack defense* to initialize just like `{Card}` class:

- `ExchangeCard` - effect: exchange opponent card's attack and defense (add `exchange_card.cpp` and `exchange_card.h`)
- `BigBossCard` - effect: adds the attack and defense of the opponent's card to all cards in the player's deck, then removes all cards in the opponent's deck that share an attack or defense stat with the opponent's card (add `big_boss_card.cpp` and `big_boss_card.h`)

if you write a rule and turn around to let player and opponent play card and draw card, the card game can be played.

Test

I provide `main.cpp` for each assignment, but this file is not mainly used for test the correctness, this is used to check if your code can be compiled correctly, because I think the logic is simple and you should be able to test your code yourself.

What to submit

1. **your report in pdf format**
2. **your source codes**

code tree

the file tree of your final code should be:

```
assignment4
`-- cardgame
    |-- big_boss_card.cpp
    |-- big_boss_card.h
    |-- card.cpp
    |-- card.h
    |-- exchange_card.cpp
    |-- exchange_card.h
    |-- main.cpp
    |-- player.cpp
    `-- player.h
```