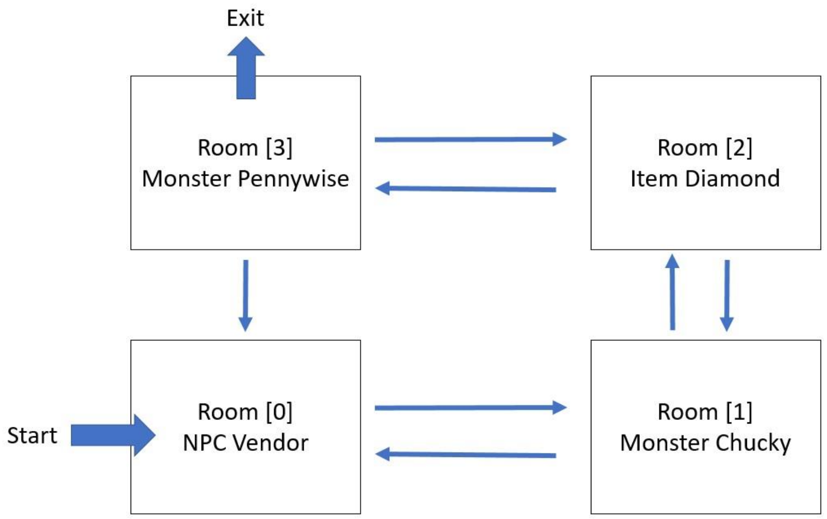
資料結構與物件導向程式設計

Dungeon Report

609001018 陳怡潔

*My Map:



Part I. Implementation Detail

1. General Game Logic:

Simply, I split the game into three fundamental parts:

<u>(1) Initializing:</u>

To control flow is mainly in *"main.cpp"* by implementing **Exception Handling.**

A. To start a new game happened in 2 conditions:

(a). If the player rejects loading the previous data:

Doing "throw" to "catch" part, constructing player, and calling "Dungeon::createMap(): void".

(b). Even if the player wants to load the previous status, the instruction isn't allowed when there is no stored data in either "Record_Player.txt" or "Record_Rooms.txt" (decided through "Record::loadToFile(Player*,Room rooms[]): bool" & "Record::checkisempty(ifstream&,ifstream&):bool"). Then, with the same action as (a), "throwing" to "catch" parts.

B. To load player and room's previous data:

If the player wants to load previous data and the command is consented, to initialize “Dungeon::player: Player” & “Dungeon::room[4]: Room” will be done through “ifstream” in “Record::loadPlayer(Player*,ifstream&): void” and “Record::loadRooms(Room therooms[], ifstream&): void.

(2) Playing Procedure:

Mainly, in *“Dungeon.cpp”*.

In “*Dungeon::runDungeon(): bool*”, there is a while loop to call *“Dungeon::enterRoom(Room*): void”*. Then it can decide which room the player is now relying on the **room index**.

I split rooms into 4 sorts in order to distinguish the specific actions of each room and call corresponding “*Dungeon::handleXXRoom(Room*): void*” (namely, handleXXRoom function) : Empty Room, Chest Room, Ghost Room and NPC Room.

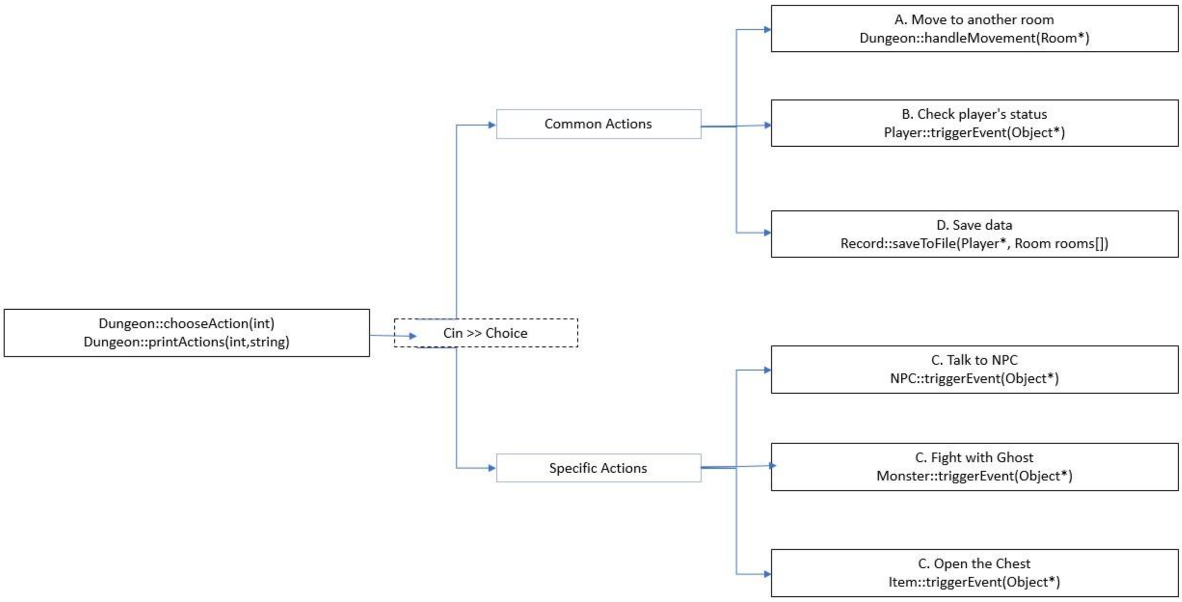
(3) End of Game:

In *“Dungeon::runDungeon(): bool”*, there are two conditions that it will return 0 to main.cpp and end the game.

A. The player is dead.
B. The player enter the last room (which isExit = true) , and the monster has been defeated by the player (or the last room is empty) .
C. Any time when the player wants to save data to file and exit.

2. Let’s talk about the details of actions in each room / with different *“handleXXRoom()”* functions:

Basically, there is a sequence of each *“handleXXRoom”* function:


Dungeon::chooseAction(int)
Dungeon::printActions(int,string)
Cin >> Choice
Common Actions
Specific Actions
A. Move to another room
Dungeon::handleMovement(Room*)
B. Check player's status
Player::triggerEvent(Object*)
D. Save data
Record::saveToFile(Player*, Room rooms[])
C. Talk to NPC
NPC::triggerEvent(Object*)
C. Fight with Ghost
Monster::triggerEvent(Object*)
C. Open the Chest
Item::triggerEvent(Object*)

Note:

1. Dungeon::handleNPCRoom(Room*):

   1-1. after talking to NPC and the transaction is done, the NPC won't be cleared. That is the player can interact with the NPC always.

2. Dungeon::handleChestRoom(Room*):

   2-1. After picking up the item in chest, the item will be cleared.

3. Dungeon::handleGhostRoom(Room*):

   3-1. After choosing to fight, the player has one chance to retreat (Dungeon::handleRetreat()). During the fight with ghost, the player cannot retreat anymore.

   3-2. After defeating the ghost, the ghost will be cleared.

   3-3. If the player enters a room with ghosts and chooses to move to another room rather than fights with them, the direction will only be allowed into the previous room.

4. Dungeon::handleEmptyRoom(Room*): the player is merely allowed to move to another room or save status into file.

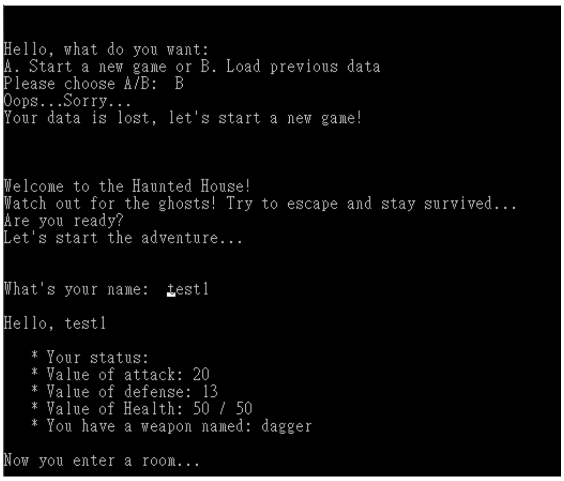
Part II. Results

The following are some particular results showing by screenshots:

1.Initializing:

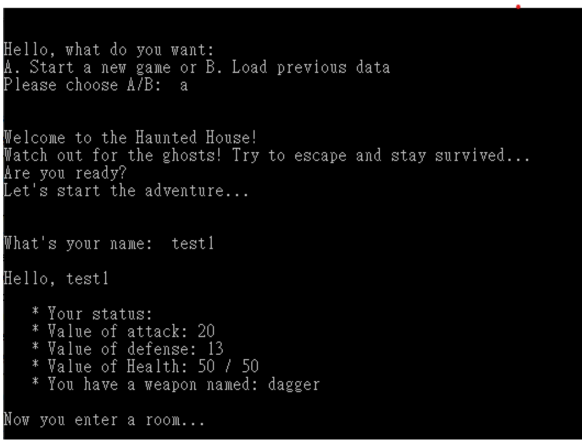
Three possible conditions:

(1) Wanna load the previous data, but there is no data existed:



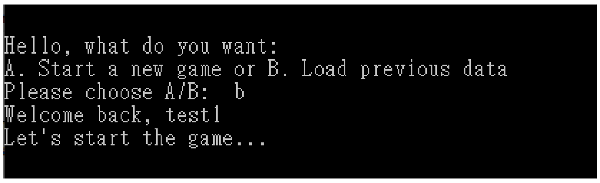
```
Hello, what do you want:
A. Start a new game or B. Load previous data
Please choose A/B:  B
Oops...Sorry...
Your data is lost, let's start a new game!



Welcome to the Haunted House!
Watch out for the ghosts! Try to escape and stay survived...
Are you ready?
Let's start the adventure...


What's your name:  test1

Hello, test1

   * Your status:
   * Value of attack: 20
   * Value of defense: 13
   * Value of Health: 50 / 50
   * You have a weapon named: dagger

Now you enter a room...
```

(2) Choose to begin a new game:



```
Hello, what do you want:
A. Start a new game or B. Load previous data
Please choose A/B:  a


Welcome to the Haunted House!
Watch out for the ghosts! Try to escape and stay survived...
Are you ready?
Let's start the adventure...


What's your name:  test1

Hello, test1

   * Your status:
   * Value of attack: 20
   * Value of defense: 13
   * Value of Health: 50 / 50
   * You have a weapon named: dagger

Now you enter a room...
```

(3) Load previous data successfully:

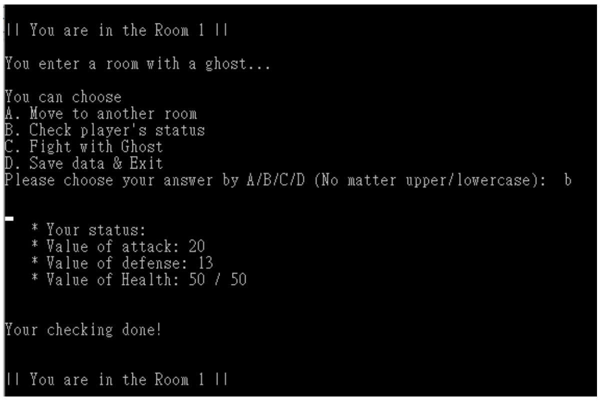


```
Hello, what do you want:
A. Start a new game or B. Load previous data
Please choose A/B:  b
Welcome back, test1
Let's start the game...
```

2. During the game:

*Common Actions:

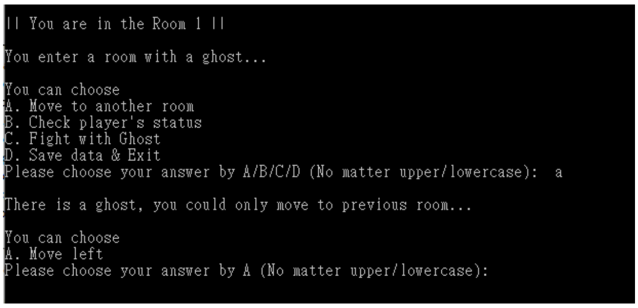
(1) Check player's status:



```
|| You are in the Room 1 ||

You enter a room with a ghost...

You can choose
A. Move to another room
B. Check player's status
C. Fight with Ghost
D. Save data & Exit
Please choose your answer by A/B/C/D (No matter upper/lowercase):  b


   * Your status:
   * Value of attack: 20
   * Value of defense: 13
   * Value of Health: 50 / 50


Your checking done!


|| You are in the Room 1 ||
```

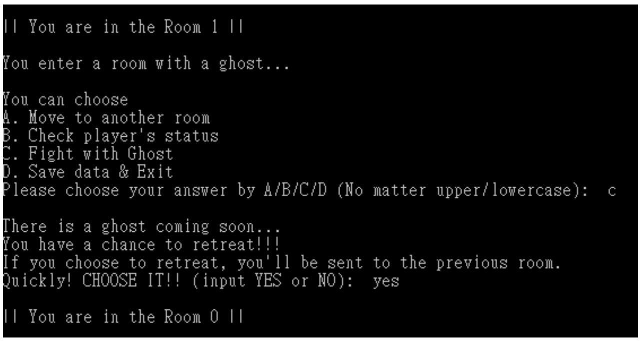
(2) Movement:

-The room with ghosts: the player could only return to the previous room



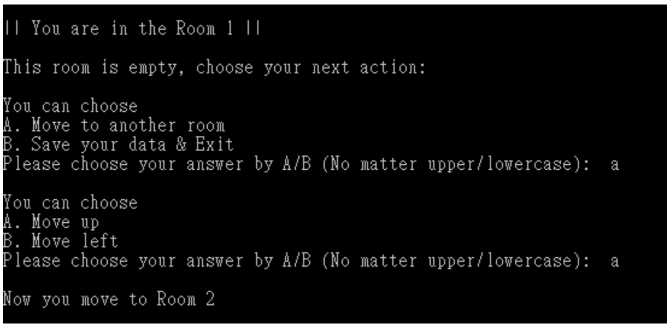
```
|| You are in the Room 1 ||

You enter a room with a ghost...

You can choose
A. Move to another room
B. Check player's status
C. Fight with Ghost
D. Save data & Exit
Please choose your answer by A/B/C/D (No matter upper/lowercase):  a

There is a ghost, you could only move to previous room...

You can choose
A. Move left
Please choose your answer by A (No matter upper/lowercase):
```

-Handle retreat: the player will be sent to the previous room:



```
|| You are in the Room 1 ||

You enter a room with a ghost...

You can choose
A. Move to another room
B. Check player's status
C. Fight with Ghost
D. Save data & Exit
Please choose your answer by A/B/C/D (No matter upper/lowercase):  c

There is a ghost coming soon...
You have a chance to retreat!!!
If you choose to retreat, you'll be sent to the previous room.
Quickly! CHOOSE IT!! (input YES or NO):  yes

|| You are in the Room 0 ||
```

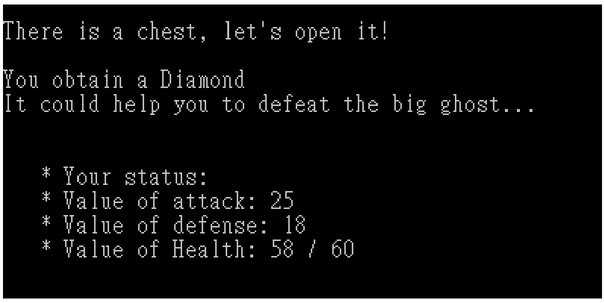
-Normal condition (depends on which room the player is):



```
|| You are in the Room 1 ||

This room is empty, choose your next action:

You can choose
A. Move to another room
B. Save your data & Exit
Please choose your answer by A/B (No matter upper/lowercase):  a

You can choose
A. Move up
B. Move left
Please choose your answer by A/B (No matter upper/lowercase):  a

Now you move to Room 2
```

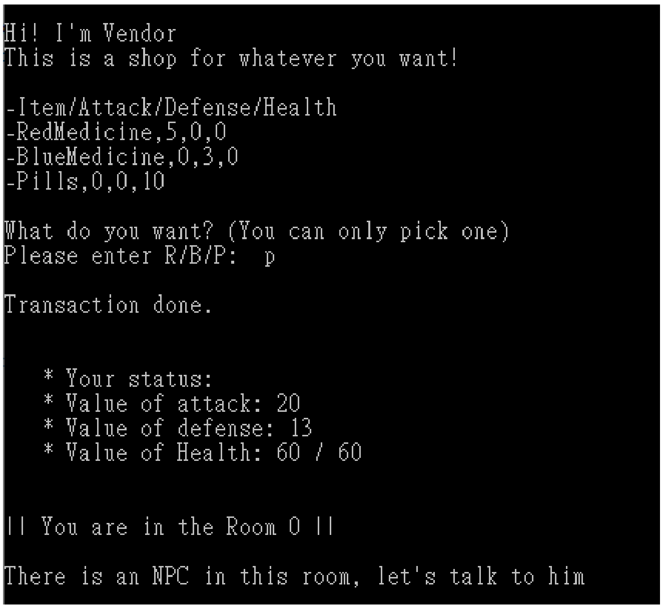
*Specific Actions:

(3) Open the Chest:



```
There is a chest, let's open it!

You obtain a Diamond
It could help you to defeat the big ghost...


   * Your status:
   * Value of attack: 25
   * Value of defense: 18
   * Value of Health: 58 / 60
```

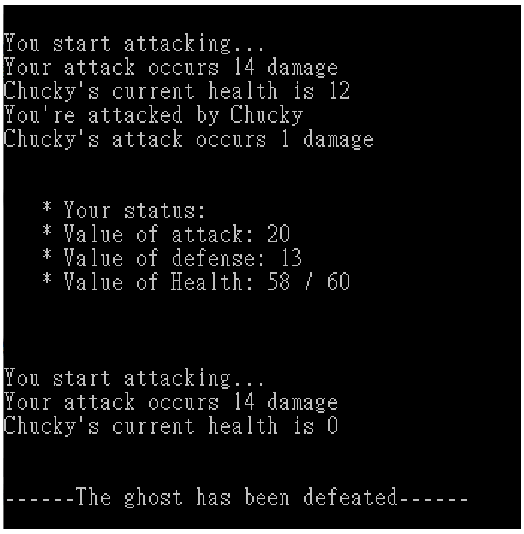
(4) Communicate with NPC: after transaction done, the NPC is still existed in the room!



```
Hi! I'm Vendor
This is a shop for whatever you want!

-Item/Attack/Defense/Health
-RedMedicine,5,0,0
-BlueMedicine,0,3,0
-Pills,0,0,10

What do you want? (You can only pick one)
Please enter R/B/P:  p

Transaction done.


   * Your status:
   * Value of attack: 20
   * Value of defense: 13
   * Value of Health: 60 / 60


|| You are in the Room 0 ||

There is an NPC in this room, let's talk to him
```

(5)Attacking Ghosts:

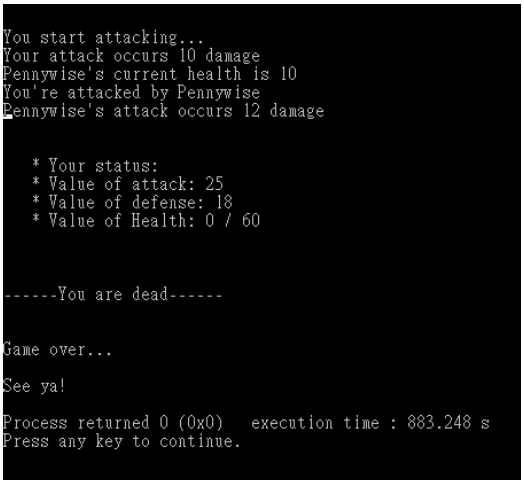
Ex. (Partially recorded)



```
You start attacking...
Your attack occurs 14 damage
Chucky's current health is 12
You're attacked by Chucky
Chucky's attack occurs 1 damage


   * Your status:
   * Value of attack: 20
   * Value of defense: 13
   * Value of Health: 58 / 60



You start attacking...
Your attack occurs 14 damage
Chucky's current health is 0


------The ghost has been defeated------
```

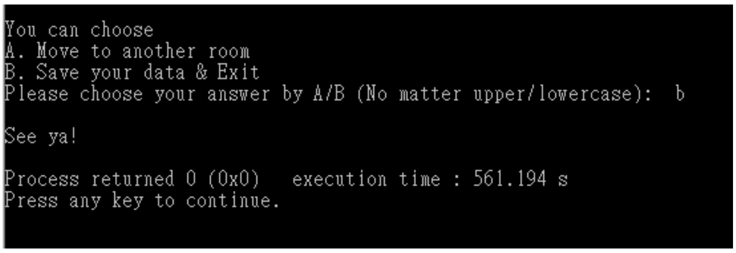
3.End of Game:

(1) Player is dead and Game over (Partially recorded)



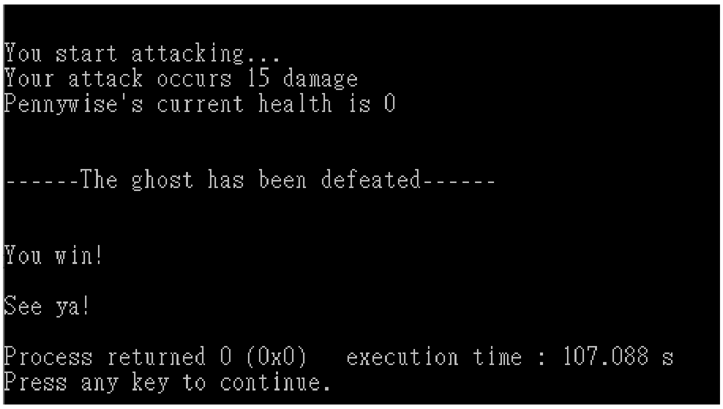
```
You start attacking...
Your attack occurs 10 damage
Pennywise's current health is 10
You're attacked by Pennywise
Pennywise's attack occurs 12 damage


   * Your status:
   * Value of attack: 25
   * Value of defense: 18
   * Value of Health: 0 / 60



------You are dead------


Game over...

See ya!

Process returned 0 (0x0)   execution time : 883.248 s
Press any key to continue.
```

(2) Save to File during the game and exit:



```
You can choose
A. Move to another room
B. Save your data & Exit
Please choose your answer by A/B (No matter upper/lowercase):  b

See ya!

Process returned 0 (0x0)   execution time : 561.194 s
Press any key to continue.
```

(3) Player defeated the last one ghost and arrived at the last room:



```
You start attacking...
Your attack occurs 15 damage
Pennywise's current health is 0


------The ghost has been defeated------


You win!

See ya!

Process returned 0 (0x0)   execution time : 107.088 s
Press any key to continue.
```

Part III. Discussion

在製作這份 Project 的過程中，最大的困難以及卡住最久的部分，主要是對於物件導向程式語言幾個特性(例如: Abstraction, Encapsulation, etc) 實作上的不理解，導致自己在思考程式邏輯上，腦袋也像抽象化般，無法找到各個 class 以及 inheritance 之間的連結，Dungeon class 的部分也一再地修改，總是百分之百的確定各個物件之間的運作結果，只能半信半疑地邊猜測邊查資料邊寫，因此也走過不少冤忘路。

而在定義 Virtual function (triggerEvent())也是使我一再無法讓程式順利執行的地方，因為程式總是停止在呼叫 Virtual function 的位置，讓我一直尋找 triggerEvent()定義上的 bug，然而尋求協助後，才發現我是在一開始 createMap()時，宣告各個房間中存取的物件，因為將其宣告成 local variables，而非以 pointer 的方式宣告 dynamic variables，存取至各個 room 中，以至於 createMap()結束時，連帶我的 objects 也一併消失，導致後續程式執行上發生連結上的錯誤。也讓我更加理解到物件之間交互作用下，牽一髮而動全身的重要性!

另外，在各種 debug 的過程中，最大的收穫大概就是很多時候一個很小的問題(例如未初始化變數，卻會導致結果有很大的出入，儘管 compilation 並未 error，卻反而更難找到問題所在，而這些情況有時候卻是不得不遇到後才會有所體悟，但只要能夠發現問題並解決，就是一段值得的 debug 過程!

Part IV. Conclusion

由於非本科系，這個 Project 算是我第一次完成的完整專案，儘管有助教們提供的資料與協助，由於自身的程式經驗不足，深深地體會到什麼叫「萬事起頭難」 以及 Debug 的辛酸血淚! 可能一個小小的錯誤就卡住個幾天，而很多時候的解決方法就是需要尋求協助，很慶幸能夠遇到很多不厭其煩願意協助我的人，不論是助教、同學、教授、別堂課的教授等。或許對於我這樣的初學者來說，一個人完成整份作業其實是挺吃力的，但當期望的結果能夠呈現出來的雀躍也是十分有成就感。

雖然很可惜由於自己在製作的過程中花了太多時間解決本身因能力不足產生的問題，沒有辦法把這個 project 做得更好，程式碼更精簡更簡單，但以此為開端，跳脫紙上談兵，真正完成這份作業後，不論是物件導向的概念或是程式設計上，都有跳躍性地成長，也期望可以從這份 Project 開始，對於程式設計有進一步更深層地認識!