

Dungeon

0816057 呂苾瑄

物件架構及功能：

Dungeon:生成玩家和地圖(包含房間、怪物、NPC 等)，控制玩家移動，判斷是否觸發特殊事件，判斷遊戲是否終止

Object

-GameCharacter

-**Player:**儲存玩家資訊(名字、血量、攻防、持有物品、所在位置等)，特殊事件:查看玩家資訊(印出目前血量及攻防)

-**Monster:** 儲存怪物資訊(名字、血量、攻防等)，特殊事件:戰鬥(玩家先攻，怪物後攻，印出玩家及怪物當前血量，判斷是否結束)

-**NPC:**儲存台詞，提供物品，特殊事件:提供物品(玩家可以選擇一個物品，更新玩家狀態，新增物品至玩家的持有物品欄，觸發 player 的特殊事件)

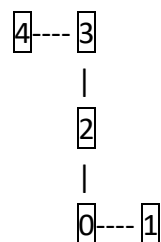
-**Item:**儲存物品資訊(名字、血量、攻防等)，特殊事件:增加玩家能力(將物品提供的屬性增加至玩家身上)

Room:儲存房間資訊(上下左右的相鄰房間、房間編號、是否為出口、房間內容物)，刪除已執行過的事件

Record:輸出或輸入各物件資訊

遊戲進行：

房間標號示意圖



```
Game start
Please enter your name.
```

遊戲開始時先輸入名字

```
Game start
Please enter your name.
sylvia
You're now at room 0
Some things in this room:
0. NPC
Choose an action.(enter the index)
```

玩家預設在 **room 0**

查看房間內的 **object**，選擇要做的事(這房間剛好只有一個 **object**)

```
Choose an item you prefer.(Please enter the index)
0. Sword, Health:0, Attack:2, Defense:0
1. Shield, Health:0, Attack:0, Defense:2
2. Magic_potion, Health:5, Attack:0, Defense:0
0
You become stronger.
Name:sylvia
Health:50 / 50
Attack:12
Defense:10
```

選擇 **NPC** 後，列出 **NPC** 的 **commodity**，玩家選擇其中一個後觸發 **item** 和 **player** 的 **triggerEvent**，分別是增加玩家能力及印出玩家資訊

```
Now you can move to other rooms.(1:up, 2:down, 3:left, 4:right)
1
Move to room 2
Some things in this room:
0. NPC
1. Little_monster2
Choose an action.(enter the index)
```

事件完成後被刪除，當房間的事件被清空之後可以進入下一個房間
此房間有兩個 **objects**

```
Now you can move to other rooms.(1:up, 2:down, 3:left, 4:right)
1
The room does not exist.
Go back to room 3
Now you can move to other rooms.(1:up, 2:down, 3:left, 4:right)
```

若是選擇的方向並沒有房間存在(例如從 **room 3** 往上走)，回到原房間重新選擇

```

monster's name:Little_monster2
Attack or retreat? (a/r)
a
You attack the monster.
Your health:50
Monster's health:9

The monster attack you.
Your health:49

Monster's health:9

You attack the monster.
Your health:49
Monster's health:3

The monster attack you.
Your health:48

Monster's health:3

You attack the monster.
Your health:48
Monster's health:0

You defeat this monster!

```

查看房間內的 **objects** 後選擇 **monster** 則進入戰鬥
 選擇 **attack** 後由玩家先攻擊怪獸，印出攻擊後雙方血量
 接著由怪獸攻擊玩家，一樣印出雙方血量
 其中一方被打敗(血量低於 0)戰鬥結束
 此範例為玩家獲勝

```

monster's name:Little_monster2
Attack or retreat? (a/r)
r
Go back to room 0
Now you can move to other rooms.(1:up, 2:down, 3:left, 4:right)

```

遇到怪物後選擇 **retreat** 退回前一個房間
 由於前一個房間已經清空，必須選擇移動的方向

出口在 4 號房間
 當 4 號房間的事件都被清除完時出現 **The end**，遊戲停止

討論

這個作業的 **function** 分散在各個不同的物件，有些彼此之間有繼承關係，在操作時必須注意他們之間的關係。

Object 中的 **virtual function** 讓 **GameCharacter (Player, Monster, NPC)**跟 **Item** 都有 **triggerEvent** 的能力，由於傳入的是 **object** 指標，必須要先 **downcasting** 確認要

呼叫誰的 `triggerEvent`。這個 `virtual function` 讓我們能分別定義在繼承同一 `parent` 的不同 `class` 之間各自的操作。

遊戲中的 `function` 有許多參數需要傳入 `class` 或 `vector` 等佔用比較多空間的東西，我們需要使用傳入 `pointer` 或傳入 `reference` 來避免這些東西被一再複製而佔用了許多記憶體位置。

結論

這個 `Dungeon` 還有一些待改進或是不太合理的地方(像是強迫推銷的 `NPC`)，但我從這個作業學到將這學期有關物件導向的知識進行實際的運用。像是將一個程式切割成許多物件，並分為 `.h` 和 `.cpp` 檔，可以讓我們在閱讀多行程式更加清楚明瞭，知道每個 `class` 包含哪些 `function` 和 `data`，以及彼此的繼承關係。將程式分成許多物件也同時意味著要將資料在各個 `class` 之間傳遞，讓我深深感受到了 `pointer` 的重要性。