

Paper Report of Dungeon

109550031

李旻融

Contents

Implementation detailed --- p. 2

- Player
- Map
- Item
- Monster
- NPC
- Action
- Showing Status
- Movement
- Game Logic
- Record System
- Optional Enhancement

Results --- p. 8

Discussion --- p. 12

Conclusion --- p. 13

[Implementation Detailed]

➤ Player :

假如是重新進行遊戲的話，一開始會先請玩家輸入名字和選擇角色，利用 Constructor 初始化每位玩家的戰鬥力。如果是回復上個遊戲設定，會將先前的值 set 進 player 裡。

以下是我所創造的角色和其戰鬥力：

1. Villager [MaxH=100, Att=100, Def=100, Mon=100]
2. Seer [MaxH=100, Att=150, Def= 50, Mon=120]
3. Hunter [MaxH=150, Att=100, Def= 50, Mon=100]
4. Witch [MaxH=150, Att= 50, Def=100, Mon=120]

➤ Map :

遊戲中共有十間房間，利用 vector 的 push_back 功能和 for 迴圈將房間加入原本是空的<vector> rooms，並且使用 linked-list (set up, down, left, right room)將每間房間串連起來，最後一間的 IsExit 要等於 true，這樣打完 BOSS 後才會離開。此外，在每個房間都創造一個 Object 型態的 vector，並將每個 item，monster, NPC 都 new 一個新的 Object 並加入其中。

➤ NPC :

整場遊戲中共有 4 個 NPC，每個 NPC 都有自己想講的話，會提供玩家三個 Item。其中有分需要花錢提升戰鬥力和賣掉戰鬥力以換取金錢，在 virtual function (triggerEvent) 中，NPC 會詢問玩家要哪一個，但玩家只能在不知道會發生什麼結果的情況下做決定。如果玩家選擇要這個 item 的話，加進 player 的 inventory，並利用 increaseStates 更改 player 的戰鬥力。

➤ Item :

整場遊戲中共有 24 個物件，其中 12 個 (money=0) 是可以被 player 撿起來的，但只能撿一次，另外 12 個 (money 有正負) 則是平均分配給四位 NPC。每個 Item 都有特定的功能替玩家提升戰鬥力，在 virtual function (triggerEvent) 中，假設玩家現在所在的房間有 Item 的話，會印出 Item 所提供的戰鬥力並詢問玩家是否願意配置此裝備。如果玩家選擇要這個 item 的話，加進 player 的 inventory，並利用 increaseStates 更改 player 的戰鬥力，而且 triggerEvent 是一個 bool function，會回傳 true，在 Dungeon 的 handleEvent function 裡，會把這個 item pop 掉，就不能再撿了。

➤ Monster :

5 隻怪獸(老虎也有伍茲)又分為 4 隻小怪，平均分散在不同房間，在最後一間則有 BOSS。每個 Monster 都有自己特定的戰鬥力，在 virtual function (triggerEvent) 中，會印出 Monster 的戰鬥力，玩家可以先確認自己現在的階級再決定是否願意與之對抗，如果願意則進行對打並顯示每回合剩餘血量，最後看是怪獸還是玩家勝出，顯示對戰結果，而且 triggerEvent 是一個 bool function，會回傳 true，在 Dungeon 的 handleEvent function 裡，會把這個 monster pop 掉，就不能再打她了。如果玩家想撤退，則會到先前的房間。

➤ Action :

先寫一個 for 迴圈來檢查房間有沒有怪獸，如果有怪獸會先跳出來，此時可以選擇要不要打怪，如果要的話，會跳到 monster 的 triggerEvent。如果是個沒有 monster 的房間，則會有五種可能的選項：

Showing Status, Movement, Logout,

Pick up the item, Talk to the NPC

如果房間沒有任何 item, monster, NPC 的話，玩家只能選擇移動。

➤ Showing Status :

這個功能在 player 的 virtual function (triggerEvent)，會 print 出玩家現在的階級，其中包括名字、血量、攻擊力、防禦力、儲蓄、現在位置。

➤ Movement :

我選擇將十間房間上下左右有通的方向一個個 print 出來，但後來發現似乎有更快的方法，可以先判斷是不是上下左右是不是 NULL，不是的話就 print 出那個方向，這樣的話程式應該可以簡單很多。然後利用 changeRoom 這個 function 移動到下個房間。

➤ Record System :

SAVE:

如果玩家在 choose action 的階段選擇登出，有兩個文字檔會分別記錄玩家和房間。紀錄玩家的檔案裡會有玩家姓名、血量、攻擊力、防禦力、現在房間 index 和前個房間 index、inventory 的所有 item，紀錄房間的檔案裡印出每個房間的 index 和上下左右房間的 index(沒有則 print 出-1)，再把房間裡剩下 object(選過的會被 pop 掉)的姓名、血量、攻擊力、防禦力都印出來。

LOAD:

玩家在遊戲剛開始的時候可以選擇要開始新遊戲還是要繼續先前的遊戲(如果上次有 save 的話)，分別 cin 玩家的姓名、血量、攻擊力、防禦力、現在房間 index 和前個房間 index、inventory 的所有 item 並且用這些值 set player，原先想把先前儲存的房間 index 和上下左右房間的 index 和剩下 object 的姓名、血量、攻擊力、防禦力都設定回去，成為一張嶄新的地圖，但好像出了一點小問題，又因為是 run-time-error，所以我找了好久還是沒辦法修正我的錯誤，最後，我想了一個新辦法，先重新 createmap 和最一開始一樣(所有 object 都有)，因為我有把 object 的名字都記錄下來，並創造一個 vector<string>，將剩下 object 的名字都存進去，所以我利用了 vector 的 find 功能，再利用 for 迴圈，將原先 map 裡頭的所有 object 都跑一次，只要有在這個 vector<string>裡頭就保留，否則 pop。

➤ Game Logic :

這是用來確認玩家死了沒，還沒的話就會一直進入 while 迴圈，叫玩家 choose action，如果玩家死了，就 print 出” GAME OVER”，並結束程式。

➤ Optional Enhancement :

1. 在遊戲一開始，玩家能夠依照自己喜歡的角色名稱或其戰鬥力，自行選擇角色。
2. Pick up the item 只能一次，但和 NPC 互動的次數則不受限制，NPC 提供了以下兩種道具，花錢提升戰鬥力和賣掉戰鬥力以換取金錢，和 item 不一樣的是，NPC 只會告訴你 item 的名稱，玩家只能在不知道 item 是否會對其造成傷害的情況下來做決定，或許一交易完，玩家的生命值就歸零也說不定。這就好像我們的人生，總是在做決定，也不知道這決定會給我們帶來怎樣的後果，但也就是因為隨機，才會讓人覺得生活處處充滿驚喜。
3. 玩家和怪獸打鬥的過程中，無論是誰進行攻擊，對方所受的傷害和剩餘血量都會被印出來，所以可以清楚的看到打鬥過程，而不只是結果。

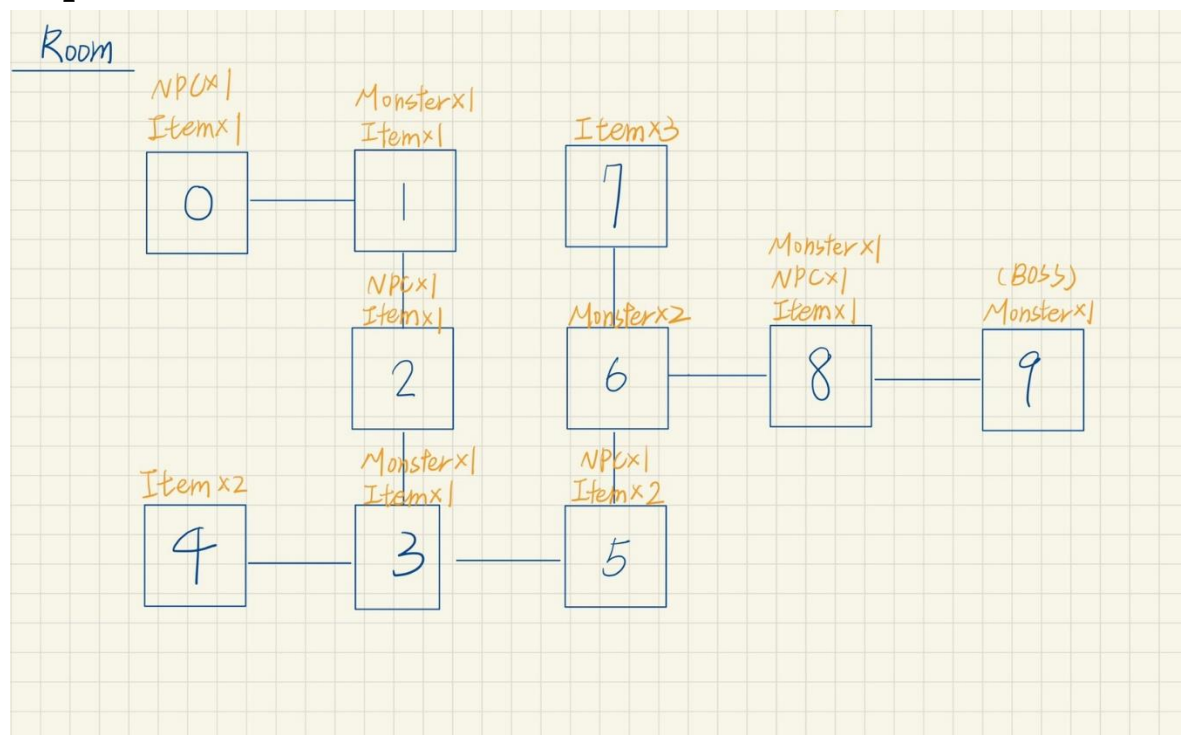
[Results]

➤ Player

```
Enter your name:
LON

[Choose the character]
1. Villager [ Maxhealth=100, Attack=100, Defense=100, Money=100 ]
2. Seer     [ Maxhealth=100, Attack=150, Defense= 50, Money=120 ]
3. Hunter   [ Maxhealth=150, Attack=100, Defense= 50, Money=100 ]
4. Witch    [ Maxhealth=150, Attack= 50, Defense=100, Money=120 ]
1
```

➤ Map



➤ Item

```
[Choose the action]
  1.Check the status
  2.Move to the other room
  3.Logout
  4.Pick up the item: Arrow
  5.Talk to the NPC: Lon
4

[Arrow]
  Health -30
  Attack +50
  Defense +10
EQUIP? [ 1.yes / 2.no ]
1

Arrow is yours now.
  [Current Health]  100--->70
  [Attack]         100--->150
  [Defense]        100--->110
```

➤ Monster

```
[Butterfly]: Health = 50 , Attack = 50 , Defense = 50
DO you want to fight with the monster? [ 1.check status / 2.yes / 3.retreat ]
1

  [Current Status]
  [Name]  LON
  [Health]  70
  [Attack]  150
  [Defense]  110
  [Money]  100
  You're in room 1!

DO you want to fight with the monster? [ 1.check status / 2.yes / 3.retreat ]
2

Butterfly: -140
Current Health:0
-----
LON: -28
Current Health:42
-----
Good! You beat Butterfly!
```

➤ NPC

```
[Choose the action]
  1.Check the status
  2.Move to the other room
  3.Logout
  4.Pick up the item: Helmet
  5.Talk to the NPC: Pipi
5

[Pipi] Wong! Wong!
[Which one do you want to equip?]
  1.Can
  2.Milk
  3.Cookie
2

[Current Health]  42--->22
[Attack]    150--->140
[Defense]   110--->100
[Money]    100--->160
```

➤ Action

```
[Choose the action]
  1.Check the status
  2.Move to the other room
  3.Logout
  4.Pick up the item: Arrow
  5.Talk to the NPC: Lon
4
```

➤ Showing Status

```
[Choose the action]
 1.Check the status
 2.Move to the other room
 3.Logout
 4.Pick up the item: Helmet
 5.Talk to the NPC: Pipi
1

[Current Status]
[Name] LON
[Health] 22
[Attack] 140
[Defense] 100
[Money] 160
You're in room 2!
```

➤ Movement

```
[Choose the action]
 1.Check the status
 2.Move to the other room
 3.Logout
 4.Pick up the item: Helmet
 5.Talk to the NPC: Pipi
2

[Choose the direction]
 1.go up
 2.go down
2
```

[Discussion]

Movement 的部分，一開始我選擇將十間房間上下左右有通的方向一個個 print 出來，但後來發現似乎有更快的方法，可以先判斷是不是上下左右是不是 NULL，不是的話就 print 出那個方向，這樣的話程式應該可以少很多行。

一開始打 record 的時候，我打了超久，大概兩百行左右，那時候我想在 save 的時候 cin 玩家的姓名、血量、攻擊力、防禦力、現在房間 index 和前個房間 index、inventory 的所有 item 並且用這些值 set player，原先想把先前儲存的房間 index 和上下左右房間的 index 和剩下 object 的姓名、血量、攻擊力、防禦力都設定回去，成為一張嶄新的地圖，但過程中好像出了一點小問題，又因為是 run-time-error，所以我找了好久還是沒辦法修正我的錯誤，到現在都沒找到，所以，我想了個新辦法，就是先重新 createmap 和最一開始一樣(所有 object 都有)，因為我有把 object 的名字都記錄下來，並創造一個 vector <string>，將剩下 object 的名字都存進去，所以我利用了 vector 的 find 功能，再利用 for 迴圈，將原先 map 裡頭的所有 object 都跑一次，只要有在這個 vector<string>裡頭就保留，否則 pop。

[Conclusion]

我覺得這次的Dungeon作業最難的地方不再把整個架構寫出來，是在Debug的時候，雖然要花很多時間，但做完的那一剎那真的很感動，而且在過程中真的可以學到很多東西，像我自己就學到了如何使用header檔和class的繼承功能，還有virtual function方法。

根據助教提供的header檔，可以得知 monster, npc, player要繼承 game character Game character, item, room要繼承Object。

首先，在handleEvent這個function裡頭，要先分辨這個物件是屬於哪一個種類(monster, item, npc)，接著才能與物件互動，所以要先在object(父繼承)設一個 pure virtual function(triggerEvent)，然後在每一個 monster, item, npc 的class中寫出各自的 triggerEvent。