

Dungeon Report

• Implementation detailed

Dungeon:

主要的遊戲架構，我透過 createPlayer 建玩家，再宣告一個 Room 型態的空陣列，透過 createMap1(連結各個房間)、createMap2(為房間設定物件)設定完整張地圖，這樣就完成遊戲前置。用 chooseAction 選接下來的動作，若選擇跟物件互動，會依房間有的物件呼叫該物件的 triggerEvent，已執行特定功能；若選擇移動，則會進入 **handleMovement**，**移動到玩家選擇的房間**，我在這裡完成了 **Movement** 的要求，每次選擇完都要跑 **checkGameLogic**，**看玩家是否勝利**，我在這裡完成 GameLogic 的要求。

main:

整個遊戲的流程，很簡潔只有呼叫各種 dungeon 裡面的函式而已。

Object:

是一個 abstract class，設定了基本的 name, tag, pure virtual function，以利其他物件繼承。

Item:

遊戲中的道具，繼承 Object，自己還有 health, attack, defense 等參數。**item** 的 **triggerEvent** 是用來撿起物品，然後加入 player 的

inventory 內，我在這裡完成 Pick up Items 的要求。

GameCharacter:

遊戲中的各種角色，繼承了 Object，比起 item 多了 type(屬性，附加功能), maxhealth, currenthealth

Player:

繼承 GameCharacter，會在一開始初始化，裡面有 inventory 負責記錄沿途得到的道具，player 的 triggerEvent 是用來印出 player 的各種參數以及他擁有的物品，我在這裡完成了 Showing Status 的要求。現在玩家所在的房間和前一個房間都是記錄在 player 內。

Monster:

繼承 GameCharacter，monster 的 triggerEvent 是進行戰鬥，我在這裡完成 Fighting System 的要求，若屬性相剋會由額外的加成，攻擊力要先扣掉防禦力才能構成有效攻擊(若扣完為負責沒有攻擊效果)，持續輪流攻擊(由玩家先)直到有一方死亡為止，若玩家遇到怪物選擇撤退，那他只能回到上一個房間(這樣才符合遊戲邏輯)，

NPC:

繼承 GameCharacter，每個 NPC 都有自己的台詞以及物品清單，我設定玩家只能選其中一樣，選完之後 NPC 就會消失，triggerEvent 就是讓 npc 說話、玩家選物品、放進 inventory、更新玩家狀態的

過程，我在這裡完成有關對 NPC 的要求。

Record:

我在這裡完成 record 的要求，分為兩大部分

1. **玩家紀錄**：紀錄玩家的各種參數，目前在哪個房間，還有 inventory 裡面每個物品的參數
2. **房間紀錄**：紀錄每個房間和其他房間的關係(上下左右是哪個房間)，房間裡面有甚麼物件，是不是出口等等的
記錄存進 GameRecord. text，讀檔的時候再把玩家跟房間都再次初始化並為他們設定原先儲存好的參數。

Room:

紀錄每個房間上下左右分別是哪個房間，房間有甚麼物件，是不是出口等等。

其他細節：

每次玩家跟物件互動完後視情況清除這個物件(我是採用將他的 tag 設定成 nothing 的方法代表清除)，在下一次進入 chooseAction 時就會沒有和物件互動這個選項。

inheritance & virtual functions:

我是照著範例的 UML 進行繼承，透過繼承讓我重複的參數和函式(像是 name, tag 等等)不用寫很多次，能直接使用祖先的，每個繼承

Object 的物件都有 override triggerEvent，並在裡面寫各自物件需要的操作(像是 item 的 triggerEven 就是用來撿道具，monster 的 triggerEvent 就是用來戰鬥)。

• Enhancement

我增加了屬性相剋的系統，讓使用者在創玩家時就先選擇屬性(有四大屬性，水火地風)，每個怪物也都有自己的屬性，如果相剋的話佔優勢的一方可以多加五點傷害，屬性參數宣告在 GameCharacter 內，monster 和 player 會在一開始的 constructor 就設定好屬性。屬性一樣可以被記錄跟存讀檔。

• Results

玩家在這個遊戲裡面可以自由選擇下一步的動作，像是移動、查看狀態、作存讀檔，透過撿道具，和 NPC 互動等方式增加自己的能力，以利跟怪物決鬥，能夠順利走到出口打贏最終 boss 就是贏家，若中途被怪物打死就輸掉遊戲。(在哪些函式裡符合要求的細節已經寫在上方的 Implementation detailed)

• Discussion

這個遊戲有蠻多擴充可能性，可以增加很多不同類型的物件(像是魔法師、同伴等等)，只要新增的物件繼承 Object 或 GameCharacter 就可以快速擁有一些基本性質，不過有個小缺點是沒有視覺化的狀

況下常常會不知道自己走到哪裡，有時候會迷路。

• Conclusion

很喜歡這次的作業，充分運用了很多 OOP 的重要觀念，像是繼承、virtual function、up/down casting 等等，做作業的過程中也學到很多實作中需要知道的細節，也更加了解物件導向。