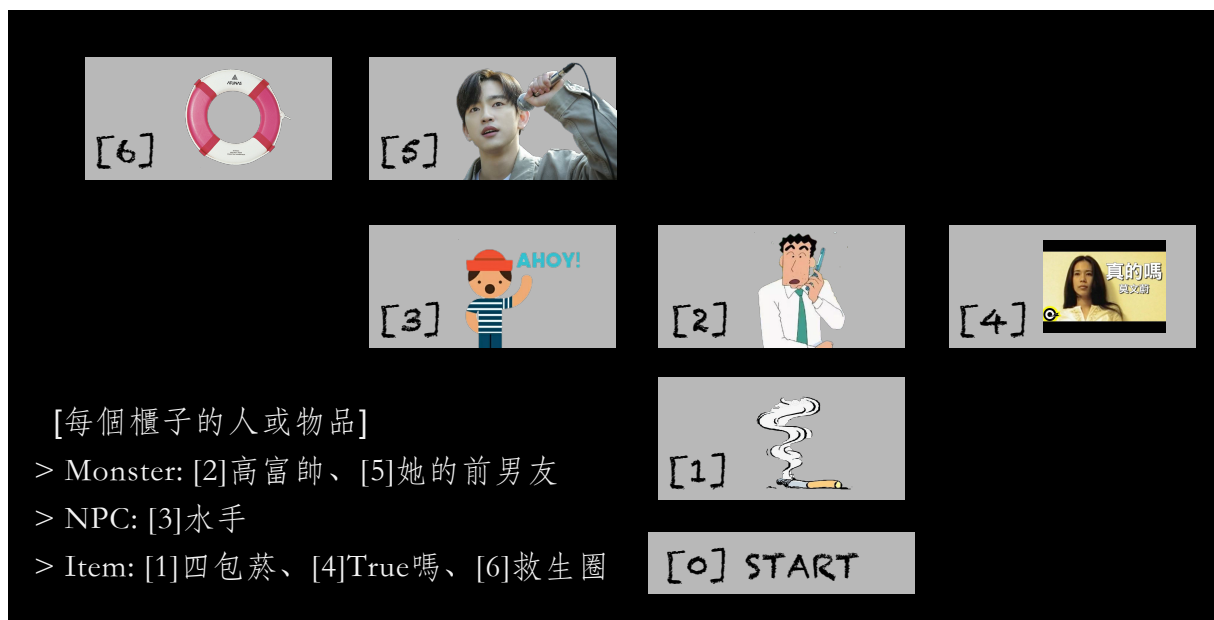


Dungeon

資工 A 林宇柔 109550069

設計基本概念介紹

我把遊戲名稱取名為「暈船勒戒所」(THE SEASICK REHAB)，玩家會先依據年齡被系統分配好角色，各角色有不同的預設健康值、攻擊力、防禦力、財力，在途中會遇到免費的道具、情敵、和一名走私商品的水手（要付錢），遊戲最終目標是要躲避情敵們的攻擊，拿到解救自己的救生圈 (lifebuoy)並逃出櫃子，成功下船。



▲ 地圖

Basic Functions - 程式碼想法分析

Movement (寫在 Dungeon.cpp 內)

```
/*創建地圖*/ void createMap();
```

我先決定了總共要建七間櫃子，然後運用 vector 先把每個櫃子要放的 Monster, NPC, Item 設定好，再用 object 的 push_back 把他們存進 object 的 vector。

接著一樣用 push_back 的方式建立櫃子（共七間），然後把每一間櫃子的初始值設定好，再把他們上下左右的櫃子直接用 setter 設定好。

```
/*玩家移動*/ void handleMovement();
```

這個函式會先判斷玩家現在在哪个櫃子，我用 switch-case 直接把每個櫃子可能繼續走的方向都寫出來，等玩家回答他們要走的方向（或存檔）之後，把 CurrentRoom 設定成要去的櫃子，然後現在在的櫃子就變成 PreviousRoom。

Showing Status（寫在 player.cpp 內）

```
/*查看狀態*/ bool triggerEvent(Object*);
```

我是直接把查看狀態當成 player 的 triggerEvent，用的是 this -> getter，會顯示玩家的名字、健康值、攻擊力、防禦力、存款、和現在待在的櫃子。

這個函式在 Dungeon.cpp 裡面是被 chooseAction 呼叫（玩家要選擇 "B. Check my status for me." 這個選項）

Pick up Items（寫在 item.cpp 內）

```
/*撿道具*/ bool triggerEvent(Object*);
```

選擇要不要撿道具我一樣是寫在 item 的 triggerEvent 裡面，一進櫃子遇到之後會先用 this -> getter 顯示道具的名字、健康值、攻擊力、防禦力，然後會問玩家到底要不要拿，如果要的話就用 addItem 把它搜集起來，然後用 IncreaseStates 把道具的能力加到玩家身上。

另外，我幫 Item 多設了一個叫 taken 的 boolean 變數，考慮到玩家有可能往回走，但我不想讓玩家一直拿一樣的 Item，所以如果在 triggerEvent 裡面玩家已經把道具拿走了，taken 就會變成 true。回到 Dungeon.cpp 裡面的 handleEvent 時就會判斷，如果 item -> getTaken 是 true 就算玩家說要拿也不能再拿一次，會直接問玩家接下來要去哪。

Fighting System（寫在 monster.cpp 內）

```
/*打架*/ bool triggerEvent(Object*);
```

打架的選擇也是放在 monster 的 triggerEvent 裡面，玩家可以選擇確認雙方狀況

、直接打架、或落跑。

情況一：如果確認完狀態就會再問一次玩家到底要不要打架。

情況二：選擇直接打架的話就是當雙方都還沒死掉（CurrentHealth > 0）就會用 takeDamage（健康值 - 對方攻擊力 + 自己防禦力）打出勝負，但因為我讓玩家先攻擊，所以只要怪物的 CurrentHealth 已經小於等於 0 了就會結束打架，玩家獲勝，反之玩家失敗，結束遊戲。

情況三：如果玩家要落跑就只能回到前一個櫃子，所以直接把 CurrentRoom 設回 PreviousRoom。

打死怪物之後，如果又回到同一個櫃子，就會在 Dungeon.cpp 裡面的 handleEvent 判斷，如果怪獸的 CurrentHealth 已經小於等於 0，會跟玩家說你已經把他打死了，直接問玩家接下來要去哪。

NPC（寫在 npc.cpp 內）

```
/*買賣商品*/ bool triggerEvent(Object*);
```

npc 的 triggerEvent 會先說他的招牌台詞，然後列出他有的三項商品，商品和道具差別就在商品要付錢，道具是撿免錢的，所以玩家先從三個裡面挑一個有興趣的出來之後，會給他看商品特質和價錢，再問他一次到底要不要，如果要的話就用 addItem 把它搜集起來，然後用 IncreaseStates 把商品的能力加到玩家身上然後扣掉玩家的存款。

一樣可能會出現撤退回同一個房間的問題，但如果是商品的話因為有三種，所以拿了一種走還是可以買另外兩種，我一樣利用 taken 註記有沒有拿過，但這次是直接 triggerEvent 裡面判斷。

商品 (commodity) 的設定是在 Dungeon.cpp 裡面，用 push_back 存進 Item 的 vector 裡面。

Game Logic（寫在 Dungeon.cpp 內）

```
/*判斷遊戲邏輯*/ bool checkGameLogic();
```

因為遊戲邏輯關係到遊戲是否結束，所以我在這個函式裡面判斷玩家的 CurrentHealth，如果小於等於 0 就是玩家死亡，會跑到 runDungeon 裡面讓遊戲結束，否則就會一直繼續遊戲直到玩家拿到救生圈，逃出櫃子。

```
/*玩家移動（逃出）*/ void handleMovement();
```

獲勝方法判斷在 handleMovement 的 case 6，也就是第六號櫃子的移動。我把救生圈想像成一個玩家自己都不知道是鑰匙的鑰匙，如果玩家進入六號櫃子但沒有拿救生圈的話，他就逃不出去，還只能往回走，直到他回六號房拿起救生圈才能逃出，遊戲結束（exit(0)）。

Record System（寫在 Record.cpp 內）

```
/*存檔*/ void saveToFile(Player*, vector<Room>&);
```

分別叫出 savePlayer 和 saveRooms 兩個 private 函式，savePlayer 就是一行一行把玩家的 Status 記錄下來，包含他拿到的道具和商品，saveRooms 則是存下有怪物的房間裡面的怪物的健康值。

```
/*讀檔*/ void loadFromFile(Player*, vector<Room>&);
```

分別叫出 loadPlayer 和 loadRooms 兩個 private 函式，savePlayer 就是一行一行把玩家存檔的 Status 設成現在的值，包含他之前拿到的道具和商品，saveRooms 則是照存下的值重設有怪物的房間裡面的怪物的健康值。

Inheritance and Virtual Functions

Inheritance

我沒有額外新增其他繼承，照助教給的提示分別為：Monster, Player, NPC 繼承了 Game Character, 而 Game Character, Item 繼承了 Object

Virtual Function

一樣是按照提示裡面，object.cpp 有個 pure virtual function (triggerEvent)，在繼承 object 的子類別裡面在寫出不同的 triggerEvent 內容。

Optional Enhancement 其他特殊設定

角色設定

我是請玩家輸入他們的年齡而不是名字，再依據年齡幫他們分配到適合的角色（已經設定好的四種角色，分別有不同的健康、攻擊力、防禦力、存款）

商品買賣

跟水手進行交易的時候要付錢，撿道具不用，如果破產了就不能繼續買東西。

撤退到有物件的房間

如果剛剛已經拿過道具了我就會當成那個房間空了，玩家就不能繼續撿，但如果是跟水手買東西的話，除了剛剛買走的東西，其他幾個只要有錢還是可以買。

其他：旁白 / 音樂

因為想要營造鋼鐵人的賈維斯（人工智能助理）的遊戲體驗，所以旁白問問題跟反應的句子都比較口語化，然後我 demo 的影片有配音樂跟音效，比較有趣。

Conclusion 結論

因為這次的 Dungeon 作業，我其實學會了很多原本不太擅長的 c++ 功能，像是繼承和 vector，而且 debug 的能力和效率我個人覺得也有很明顯的提升。寫作業的過程其實蠻好玩的（除了抓不出錯誤的時候），應該是實質上第一次真正運用程式之事做出一個成品。

覺得自己這次最可以改進的地方就是太晚開始做，導致有很多想寫的東西卻礙於時間有限而沒辦法發揮，除此之外覺得自己進步蠻多的。