Dungeon Project

Demo

可以在這裡看到各個功能的demo影片以及實際遊玩的影

Introduction

Game Outline

這是一款經典的地下城遊戲。在地下城中會遇到許多的道具及NPC， 玩家必須從不同的房間移動，擊敗怪物並升級。最後，打敗boss就可以找到地牢的出口，並逃離地下城

1. Basic functions

Action menu

根據每個房間的type去列出相對應的action menu

1. **Monster ⇒ 攻擊確認(attack/retreat)**
    - 若確認攻擊，則會觸發monster的triggerEvent並開始戰鬥
    - 若選擇retreat，則會回到上一個房間，並顯示default action選單
2. **NPC ⇒ 互動確認(talk/ignore)**
    - 若確認互動，則會觸發NPC的triggerEvent並開始互動
    - 若選擇ignore，則會跳出default action選單
3. **其他房間 ⇒ default action (move/status/shop/backpack/save)**
    - 列出基本功能的選單

Status

實作在player的triggerEvent裡，會列出所有玩家的數值屬性及狀態，包含飢餓、口渴值，武器、裝備、道聚，金幣、經驗值等等。

Movement

列出所有玩家在此房間可以走的方向，選擇方向後，利用Player class的changeRoom function改變currentRoom及記錄previousRoom

Pick up Item

利用Item的triggerEvent來讓玩家撿取Item，不同的type會分別做出不同的互動

1. **weapon、armor**
    - 會列出玩家現在裝備的屬性及觸發triggerEvent的Item的屬性，詢問玩家是否要更換weapon或armor
2. **potion 、 food 、 drink**
    - 印出Item名稱，並列出更新後玩家背包裡的Item

NPC

遊戲包含了四個NPC，分別是：

1. **merchant**
    - 與其互動可以購買藥水，分別是提升atk、def、hp的
    - 還有一種較特殊的是reborn potion((還沒寫!!!!!)) (見Item design)
2. **chief**
    - 若背包裡有打敗動物獲得的meat，將其拿給chief可以烹煮成BBQ ，能夠恢復較多hunger值，且吃了不會中毒
3. **weapon_master**
    - weapon master的房間在初始房間的正上方，會在一開始就給予玩家一把武器
4. **nurse**
    - 與其互動可以恢復生命值
    - 若currentHealth低於maxHealteh的80%，則恢復至80%
    - 若高於80%，則恢復至全滿

Fighting System

遊戲採用player攻擊一次、monster攻擊一次的回合制玩法

1. **玩家的回合，可以選擇三種action**
    a. attack
        - 傷害 = 攻擊者的攻擊力-攻擊對象的防禦力
        - 更新currentHealth，並在每個回合印出player與monster的剩餘生命值
        - 利用rand函數，在每次player攻擊時會有機率產生爆擊，造成1.5倍的傷害
    b. open backpack
        - 玩家可以利用potion來提升自身屬性，potion效果會在戰鬥結束後失效
        - 若在戰鬥中陷入中毒狀態，可以喝牛奶來解除中毒
    c. retreat
        - 玩家會被傳送回前一個房間
2. **戰鬥結束**
    a. 若玩家死亡，則遊戲結束
    b. 若monster死亡
        - 掉落gold及exp
        - 檢查更新後的exp是否可以升級，升級可以提升玩家的hp、atk、def，升級所需的exp也會跟著提高(見optional enhancement)
        - 呼叫player的triggerEvent印出結果顯示玩家戰鬥完的數值
3. **Boss的特殊技能**
    a. 使用rand生成亂數作為機率函式，Boss有機率可以免疫玩家攻擊
    b. Boss在攻擊玩家時可以恢復自身血量

Game Logic

判斷遊戲是否結束，若玩家死亡或玩家成功擊敗boss找到出口都會return false，使遊戲迴圈中斷，並印出結果。若玩家死亡則會輸出死因

2. Hunger system

hunger、thirst

1. **減少**
    - 在房間的移動會讓hunger及thirst減少2
    - 攻擊monster會讓hunger及thirst減少5
    - 在forest、desert房間則有不同的計算 (見Room system)
2. **hunger值或thirst值降至0**
    - 獲得 hungry或thirsty 的狀態
    - hunger或thirst的減少值會變成以生命值來扣
    - 若thirst值歸零，則遊戲會詢問是否要購買珍珠奶茶來解渴，恢復口渴值
3. **恢復**
    - 透過food、drink來恢復 (見Item design)
    - 在forest或desert中有小機率可以找到綠洲或湖泊
    - 在玩家升級時，可以選擇thirst或hunger其一來恢復至50

poison

1. **獲得中毒狀態**
    - 打敗動物會獲得meat，但若直接食用即會中毒
    - 遇到名為 “Gengar” 的monster時，他的攻擊有機率會附帶中毒效果
2. **中毒狀態**
    - 若有中毒狀態，則會在每一回合(移動、與Item互動、fighting中的每一回合)都扣除生命值
3. **恢復**
    - 玩家只能透過喝牛奶來解除中毒狀態

3. Room system

Desert

1. 玩家在這個房間內移動扣較多的口渴值(-5)
2. 有機率遇到沙塵暴及沙洲
    - 若遇到沙塵暴，則玩家口渴值-10
    - 有較低的機率遇到沙洲，可以恢復口渴值15

Forest

1. 玩家在這個房間內移動扣較多的口渴值(-5)
2. 有較低機率遇到湖泊(口渴值+15)
3. 森林裡會有動物，擊敗他們可以獲得meat

Swamp

在沼澤中的戰鬥無法使用武器及裝備，玩家會被強制脫下武器及裝備。

4. Item design

Weapon、Armor

1. **可以增加玩家的atk或def**
2. **取得途徑**

    與weapon master互動、撿取monster掉落物、商店購買
3. **種類**
    - weapon包含了sword、crysknife、bow、gun
    - armor包含了armor、shield

Potion

1. **限定在戰鬥中才可使用，會提升玩家戰鬥數值，戰鬥結束後失效**
2. **取得途徑**

    與merchant購買、撿取monster掉落物、商店購買
3. **種類**

- atk potion、def potion、hp potion分別會提升玩家的攻擊力、防禦力及生命值，藥水等級越高提升越多
- reborn potion可以讓玩家在死亡時以45%的生命值復活

Food

1. **可以恢復玩家的飢餓值**
2. **取得途徑**

   與chief互動、撿取monster掉落物、在empty kitchen房間中撿取、商店購買
3. **種類**
   - food的type包含了：bread、apple、BBQ、chocolate、rice ball、udon、curry rice
   - meat的type則是badFood，吃了會使玩家獲得中毒狀態，但可以拿去給chief烹煮獲得BBQ

Drink

1. **可以恢復玩家的口渴值**
2. **取得途徑**

   撿取monster掉落物、在empty kitchen房間中撿取、商店購買
3. **種類**
   - 包含了milk tea、coffee、cola、holoyoi
   - 若使用了milk則可以解除現有的中毒狀態
   - 在口渴值降為0時，可以透過購買bubble tea來恢復

5. Optional enhancement

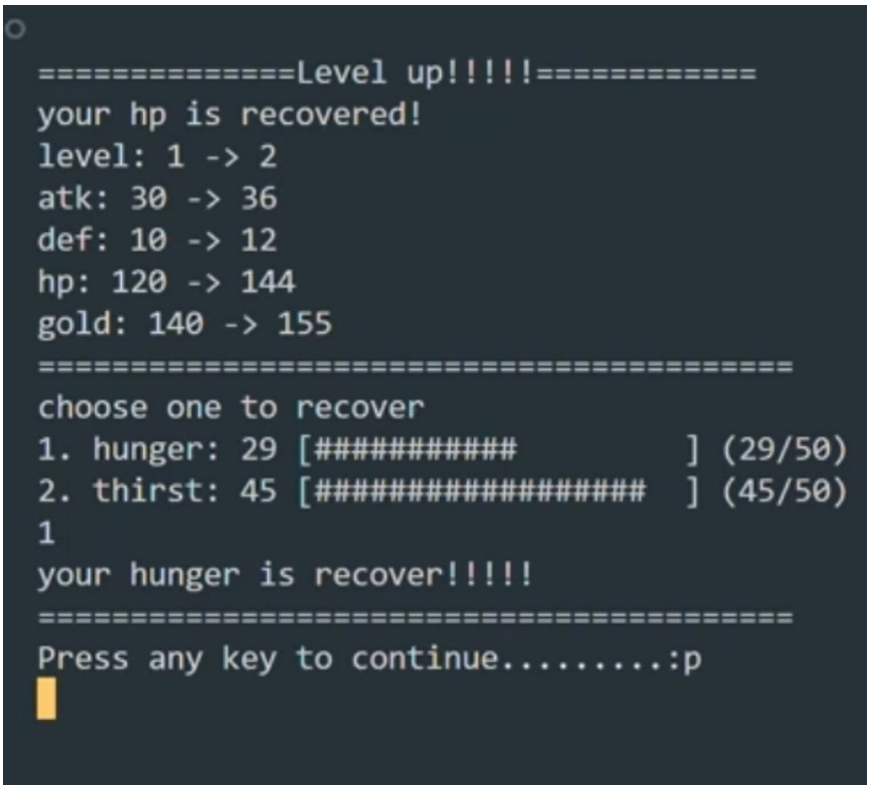
Level system

對玩家加上了等級系統

1. **如何升級**
   - 可以透過擊敗怪物獲得經驗值，每隻怪物提供的經驗值不同

- 若玩家現有的經驗值達到升級所需之經驗值即可升級

2. **升級後的數值更新公式**
    - 下次升級所需之經驗值 = (level ^ 3) + 15
    - atk、hp、def提升為1.2倍
    - 可選擇hunger、thirst其一恢復至全滿(50)



Critical attack

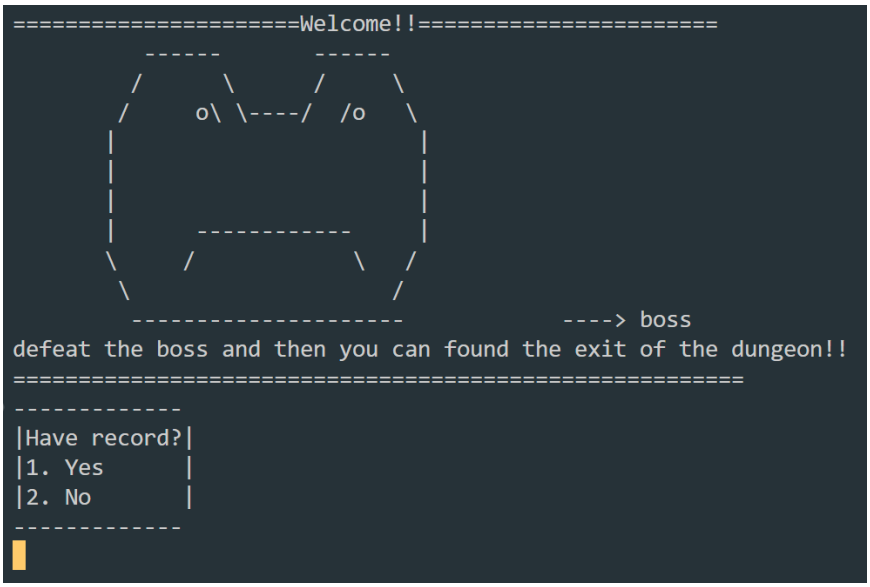
玩家在戰鬥中可能觸發暴擊

1. **如何觸發**
    - 玩家在每次攻擊時都使用一次rand函式產生一個在[0,1]中間的浮點數，若該浮點數≤0.35即可觸發爆擊
2. **暴擊傷害**
    - 暴擊可以產生攻擊力1.5倍的傷害
        - critical attack

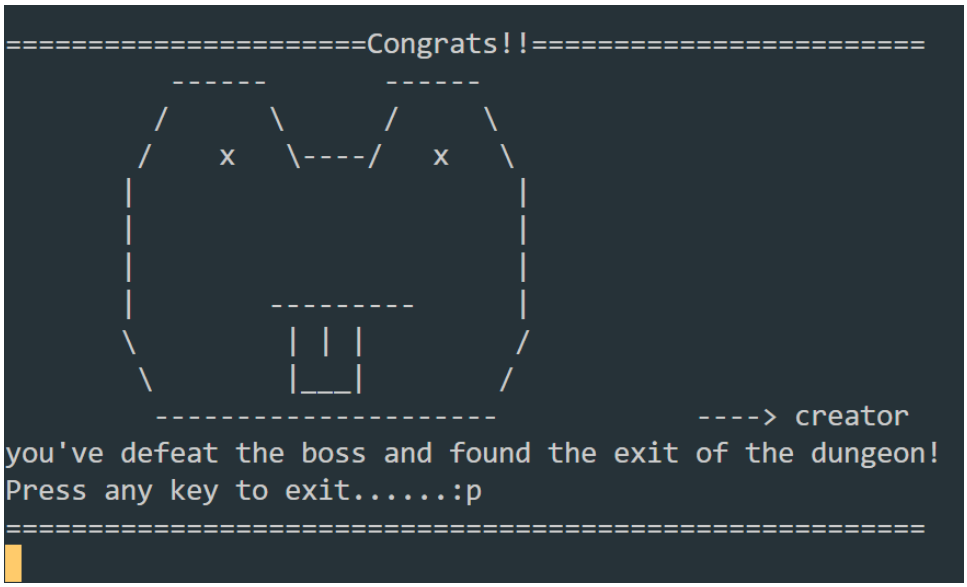
```
Round: 1 -------------------------
What do you want to do?
1. attack
2. open backpack
3. retreat
enter your action: 1

++++++++++++++++++
<Critical Attack!>
++++++++++++++++++
you cause 35 damage
bear's Hp: 50 -> 15
your Hp: 120 -> 120

Round: 1 -------------------------
bear attacks you and cause 15 damage
bear's Hp: 15 -> 15
your Hp: 120 -> 105
```

Visualize design

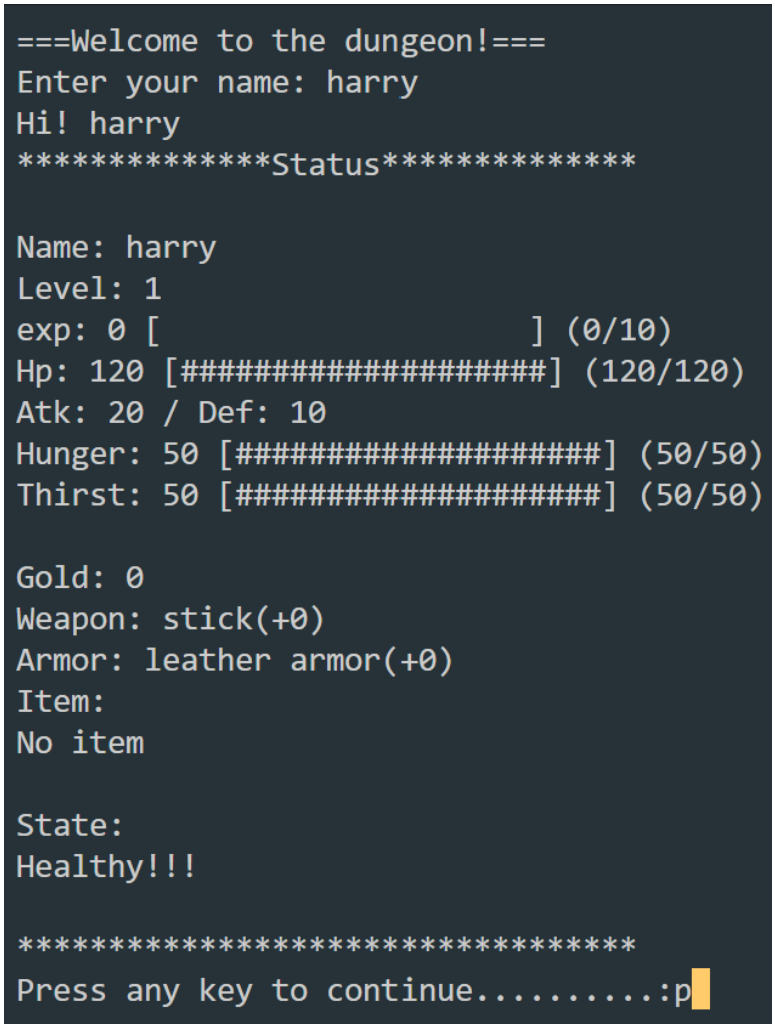
1. 加入一些視覺設計讓遊戲更豐富有趣
    - 起始畫面



- 玩家獲勝畫面



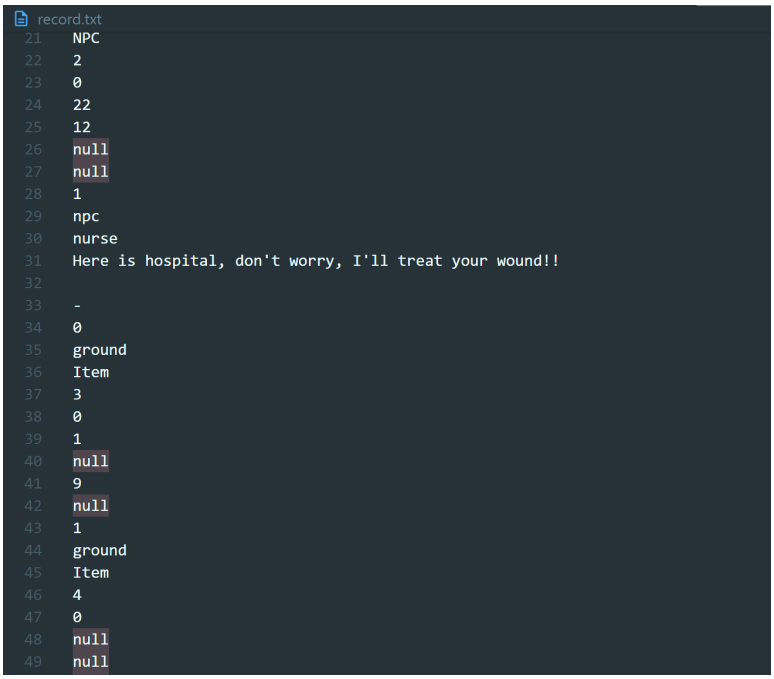
- 長條圖讓玩家能夠更直觀的看出數值變化



Record system

可以儲存玩家的進度並寫入record.txt，若有空指標，則輸出"null"作為讀取辨識用

1. **存檔**
    - 玩家在default action選單中可以選擇存檔
    - 因為不同object擁有的屬性不同，會針對不同object的tag做不同的輸出處理方式
        - Room：透過index紀錄每個房間的上下左右房間，並將房間的object(monster、NPC、item)呼叫對應各自tag的函式輸出
        - Monster、Player：將所有數值及Item輸出
        - NPC：將script以及Item輸出
        - Item：輸出type以及name，對於不同type做不同的處理
            - 若type為weapon、armor或potion，則記錄他們提升的atk、def以及hp
            - 若type為food、drink或badFood，則記錄他們恢復的hunger、thirst，以及造成的poison和poison_time
    - 資料會輸出至record.txt中儲存
        - record.txt



2. **讀取檔案**
    - 玩家在遊戲開始時可以決定是否要讀取先前的存檔
    - 與輸出相同，針對不同object的tag做不同的讀取處理方式，再利用setter函式設定資料

Conslusion

第一次嘗試這種將一個專題拆成許多元件完成的方式，但是在寫完dungeon這個專題後，對於物件導向設計又有更進一步的理解了，

在完成作業的過程中也遇到了一些程式設計上的問題是未來可以改進的，最主要的是Stack可能overflow的問題，在最後檢視自己的程式時，發現Dungeon.cpp中的defaultAction()這個函式的運作方式，不是return完結束函式再重新呼叫，而是在原本函式尚未結束的情況下又重新呼叫自己，造成call stack越疊越高，雖然因為現在只是一個小專案，對程式運行影響不大，但是未來應避免同樣的問題發生。