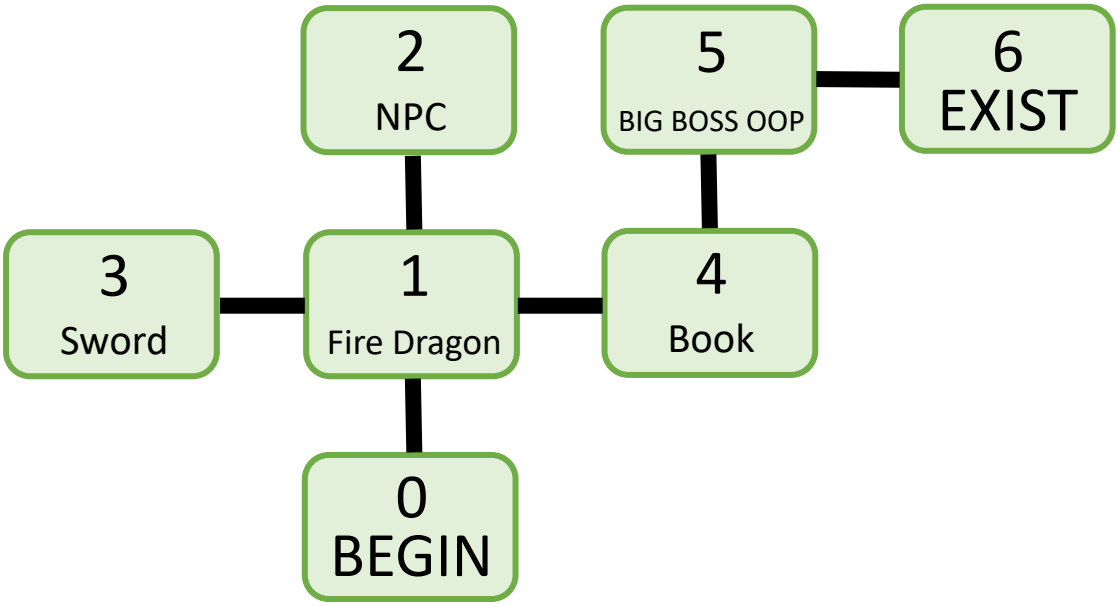
110704039 許甄芸

遊戲地圖設計:



1. Movement

   Implement: 我的地圖是使用 for 迴圈產生總共 7 個房間並放入 vector 中，分別用 class Room private 中的 index 變數當作房間編號，再用 link list 將各個房皆串在一起，並且使用 set_upRoom、set_downRoom、set_leftRoom、set_rightRoom 做出地圖的方位關係。

   在根據玩家選擇上下左右移動時，將 class player 中的 currentRoom 指向所選擇的房間、previousRoom 指向原本的房間，以便玩家遇到魔王時可以撤退至上個房間的設定。

   Result: 成功實現的功能有玩家可以選擇往上下左右，在地圖中任意穿梭，並且在遇到魔王時可以選擇撤退至前一個房間。(如以下運行截圖所展示)

```
----------------------
choose your direction:
1.go up 2.go down 3.go right 4.go left 5.save game progress and logout
```

```
------------------------------------------------------
Oh no! There is a monster!!!
Do you want to fight with fire dragon (Health: 80, Attack: 20, Defense: 10)?
1.yes! 2.escape to previous room 3.check my status
```

2. Showing Status

   Implements: 在 class player 中的函數 triggerEvent 中可以顯示玩家的 name、current health、current attack、current defense、current room。而當玩家在 room 0 或者遇到 monster 時若選擇 show status 便會執行此函數。(如以下程式碼所示)

```cpp
bool Player::triggerEvent(Object* object){
    Player* player = dynamic_cast<Player*>(object);
    cout<<"Your name is "<<get_name()<<endl;
    cout<<"Your Health: "<<get_currentHealth()<<endl;
    cout<<"Your Attack "<<get_attack()<<endl;
    cout<<"Your Defense "<<get_defense()<<endl;
    cout<<"You are in room "<<currentRoom->get_index()<<endl;
}
```

Result: 成功實現我想讓玩家在遊戲一開始知道自己所選擇之角色的各種屬性能力、以及在遇到 monster 不確定能否打敗且需不需要撤退時能夠查看自己目前的能力及狀態值的功能(如以下運行截圖所展示)

```
----------------------------------------
Your in room 0 now, what do you want to do next?
1.Show my status 2.move 3.save and logout
1
Your name is jenny
Your Health: 75
Your Attack 75
Your Defense 20
You are in room 0
----------------------------------------
```

```
------------------------------------------------------
Oh no! There is a monster!!!
Do you want to fight with fire dragon (Health: 80, Attack: 20, Defense: 10)?
1.yes! 2.escape to previous room 3.check my status
3
jenny ( Current health: 80, Attack: 20, Defense: 10)
Your name is jenny
Your Health: 75
Your Attack 75
Your Defense 20
You are in room 1
----------------------------
```

3. Pick up Items

   Implement: 於 Dungeon 的 createMap()中創建指向 Object 的 vector，並把 new 好的 Item 以.push_back 的方法裝入指定房間中。在玩家進入該房間時 class Item 中的函式 triggerEvent 會詢問玩家是否要裝備該 item，並且使用 get_name()、get_health()、get_attack()、get_defense()告訴玩家此 item 的名字與各屬性數值，若玩家選擇”是”則會利用 class player 中的 increaseStaes 將該 item 的 health、attack、defense 數值加入玩家的屬性數值中。(如以下程式碼所示)

```cpp
bool Item::triggerEvent(Object* object){
    int answer;
    Player* player = dynamic_cast<Player*>(object);
    cout << "------------------------------------------------------"<<endl;
    cout << "So lucky! You found an supplies over there!" << endl;
    cout << "Do you want to equip " << get_name() << "(" << get_health() << ","<<get_attack() << "," << get_defense() << ")?" << endl;
    cout << "1.Yes 2.No" << endl;
    cin >> answer;
    cout << endl;
    if(answer==1){
        player->addItem(*this);
        player->increaseStates(get_health(), get_attack(), get_defense());
        return true;
    }
    else
        return false;
```

Result: 可以讓玩家在房間探索時找到各式各樣的補給品，並且可以決定是否要裝備它，以便和怪物戰鬥時有更多的屬性數值好打敗他們。(如以下運行截圖所展示)

```
----------------------------------------
So lucky! You found an supplies over there!
Do you want to equip sword(100,25,5)?
1.Yes 2.No
1
```

4. Fighting System

   Implement: 於 Dungeon 的 createMap()中創建指向 Object 的 vector，並把 new 好的 Monster 以.push_back 的方法裝入指定房間中。

   在玩家進入該房間時 class Monster 中的函式 triggerEvent 會詢問玩家要與該 Monster 對戰還是逃回上一個房間或者查看玩家目前屬性數值，若選擇對戰則使用 if 判斷玩家及 Monster 的 current health 是否都大於 0，true 則以迴圈執行 class GameCharacter 中的 takeDamage 函式，分別將 Monster 及 Player 的 currentHealth 扣掉自身的(attack-defense)，直到一方的 currentHealth 小於 0。

   Result: 以玩家的防禦力及攻擊力相減為造成的生命值傷害量，與 Monster 不斷互相攻擊最終就會有一方死亡。(如以下運行截圖所展示)

```
----------------------------------------------------
Oh no! There is a monster!!!
Do you want to fight with fire dragon (Health: 80, Attack: 20, Defense: 10)?
1.yes! 2.escape to previous room 3.check my status
3
jenny ( Current health: 80, Attack: 20, Defense: 10)
Your name is jenny
Your Health: 75
Your Attack 75
Your Defense 20
You are in room 1
---------------------------
Now, decided again:
1.Fight! 2.Retreat
1
You beatfire dragon
```

5. NPC

   Implement: 於 Dungeon 的 createMap()中創建型態為 Item 的 vector，並把 new 好的 NPC 以.push_back 的方法裝入指定房間中。

   在玩家進入該房間時 class NPC 中的函式 triggerEvent 會呼叫 get_script()使輸出該 NPC 被設定的固定台詞，並且告訴玩家它所擁有的物品及其屬性，玩家可以選擇接受或不接受該物品，接受則呼叫 class player 中的 increaStates 函式，將物品屬性數值加入玩家屬性中。(如以下程式碼所示)

```
bool NPC::triggerEvent(Object* object){
    int answer;
    Player* player = dynamic_cast<Player*>(object);
    cout << "----------------------------------------" << endl;
    cout << "You found an old man standing over there." << endl;
    cout << ": Hello, " << object->get_name() << ". " << get_script();
    cout << " I have " << commodity[0].get_name() << "(" << commodity[0].get_health() << "," << commodity[0].get_attack() << "," << commodity[0].get_defense() << ")" << endl;
    cout << "1.yes 2.no" << endl;
    cin >> answer;
    if(answer == 1){
        player->addItem(commodity[0]);
        player->increaseStates(commodity[0].get_health(), commodity[0].get_attack(), commodity[0].get_defense());
        return true;
    }
    else if(answer==2){
        return false;
    }
```

Result: 成功實作出在遊戲中安排 NPC 的角色，並提供物品給玩家。而我也幫 NPC 加入了故事性讓遊戲帶入感更深。(如以下運行截圖所展示)

```
----------------------------------------
You found an old man standing over there.
: Hello, jenny. I've been stocked in here for a long time, now you are my only hope to escape.
I wish you can beat all these monster!
Do you want some supplies? I have medicine(150,15,30)
1.yes 2.no

1
----------------------
```

6. Game Logic

   Implement: 在 class room 中有個 bool 值 isExit，用來判斷玩家是否到了地圖的終點，若為 true 則輸出” CONGRATULATION　YOU　WIN !!!”並結束遊戲，或者在 Fighting System 中被 Monster 打敗了則會輸出"You were beaten by(Monster 名)QAQ"以及"**********GAME OVER**********"並結束遊戲。

   Result: 實作出了遊戲的輸贏邏輯。(如以下運行截圖所展示)

```
----------------------
choose your direction:
1.go right 2.go down 3.save game progress and logout
1

----------------------
CONGRATULATION  YOU  WIN !!!
```

```
------------------------------------------------------
Oh no! There is a monster!!!
Do you want to fight with BIG BOSS OOP (Health: 500, Attack: 100, Defense: 10)?
1.yes! 2.escape to previous room 3.check my status
1
You were beaten by BIG BOSS OOP QAQ
**********GAME OVER**********
```

7. Character Class Design

   Implement: 在 class Dungeon 的 createPlayer()中讓玩家選擇自己想要的角色，並使用 switch and case 將玩家所選擇的角色屬性值傳入 class Player 中，我設定屬性有三種分別為 class GameCharacter private 中的變數 currentHealth、attack、defense，而因為 class Player 有繼承 public GameCharacter 因此可以使用 class GameCharacter 中的 set_currentHealth、set_attack、set_defense 來設定屬性。(如以下程式碼所示)

```
void Dungeon::createPlayer(){
    string name;
    int choose;
    cout << "----------------------" << endl;
    cout << "Please enter your name" << endl;
    cin >> name;
    cout << "Choose your character: 1.Knight 2.Witch 3.Wsarrior "<<endl;
    cin>>choose;
    switch(choose){
        case 1:{
            Player player(name,75,75,20);
            player.set_maxHealth(500);
            this->player= player;
            break;
        }
        case 2:{
            Player player(name,100,50,10);
            player.set_maxHealth(500);
            this->player = player;
            break;
        }
        case 3:{
            Player player(name,150,25,15);
            player.set_maxHealth(500);
            this->player = player;
        }
    }
}
```

Result: 實作出讓玩家選擇自己想要的角色之功能，且不同角色有不同的生命值、攻擊力、以及防禦力(如以下運行截圖所展示)

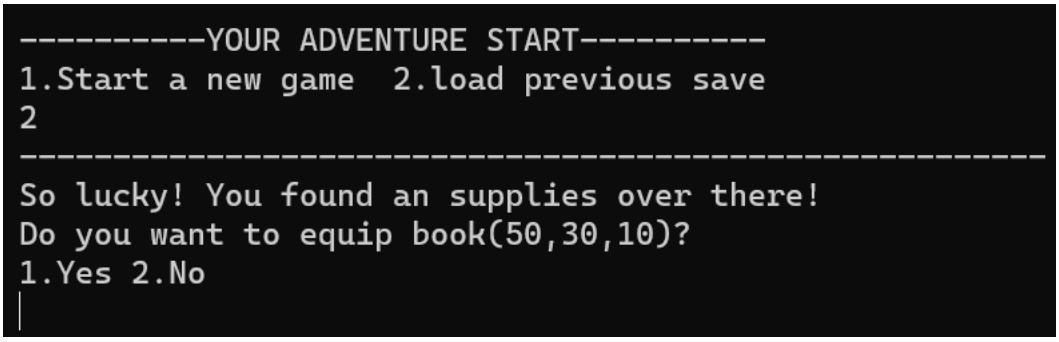
```
----------YOUR ADVENTURE START----------
1.Start a new game  2.load previous save
1
----------------------
Please enter your name
jenny
Choose your character: 1.Knight 2.Witch 3.Wsarrior
1
-----------------------------------------
Your in room 0 now, what do you want to do next?
1.Show my status 2.move 3.save and logout
1
Your name is jenny
Your Health: 75
Your Attack 75
Your Defense 20
You are in room 0
```

8. Optional Enhancement

Implement: 首先將 player 的所有數據存進 recordPlayer 的 txt 檔，包括 currentRoom、name、currentHealth、attack、defense 等等。再將 Monster 的資訊存也存到 recordrooms 的 txt 檔，因為再創建一個地圖的話除了

monster 的 currentHealth 以外都不會變，所以 recordrooms.txt 包括了 Monster 的 currentHealth。在 load 方面，在遊戲開始時會有 start a new game 跟 load previous save 兩個選項，若玩家選擇後者則將兩個 txt 檔的內容都存進程式中，繼續先前的遊戲進度。

Result: 成功實做出了若玩家想要暫時離開遊戲，下次仍可以重回遊戲進度的功能。(如以下運行截圖所展示)



Discussion:

在此遊戲中 class Object 有用到 pure virtual function: virtual bool triggerEvent(Object*) = 0;，並繼承給了 class Item、class GameCharacter、class Monster、class NPC、class Player，其中在 Item、Monster、NPC、Player 中有做 override 的動作。

實做下來我覺得它真的讓程式碼優化了許多，不需要在每個 class 重新做不同的 function，可以直接用 pure virtual function 做改寫的動作以符合每個類別所需，且還可以擁有自己的資料成員與成員函數。不只如此，我覺得它還讓定義清楚了許多，可以讓程式開發者在做修改的時候一下就知道這個 function 的功用就是觸發事件。

Conclusion:

因為之前從來沒有寫過程式的 project，都是一個程式檔案包含所有內容，因此習慣這個 project 的用法就花了我很多時間，包括在設計及 debug 的時候邏輯都要非常清晰，因為很多繼承的動作，要很清楚哪個檔案包含的是哪些功能，且在 debug 的時候只要有一個地方錯誤就很有可能要改很多的 class 內容。

然而在遊戲開發的過程中也是有找到一些樂趣，我開始會思考替這個遊戲加些甚麼功能可以讓它更有趣，但其實我想得很多功能後來都沒有實現，大概是因為我的程式能力還不夠好吧!很多都是寫一寫 debug 超~級久還是無法成功，最後就放棄並刪除這個功能了，希望這學期上完這門課我可以有所進步，然後到時候再寫出一個真的符合自己期望的遊戲!