封面

目錄

1. 簡介
2. 動機:

我們選的遊戲是Dead Cell「死亡細胞」，會選這個遊戲的原因是因為他是最近一年相當紅的2D橫向卷軸遊戲，但它獨特的地方是它具有最近很火紅的「魂系列」元素，是以高難度和輪迴的為特色的一種遊戲方式，流暢的角色操作和多元的裝備組合受到玩家的好評，我們認為是最近具代表的遊戲之一。

1. 分工:

廖彥澤:

怪物(惡魔)、裝備(弓、劍)、特效、地圖設計、UI、傳送門、終點、門、寶箱。

郭家佑:

主角、Boss、遊戲系統、地圖系統、顯示文字、按鈕、整合、障礙排除。

1. 遊戲介紹
   1. 遊戲說明
      1. 遊戲目標

在第一關打敗怪物獲取武器後，進入第二關打敗BOSS。

* + 1. 操作說明:

方向鍵:移動

Z:翻滾

X:攻擊

C:互動(開寶箱、撿東西、使用傳送門等)

空白鍵:跳躍

下+空白鍵:下跳

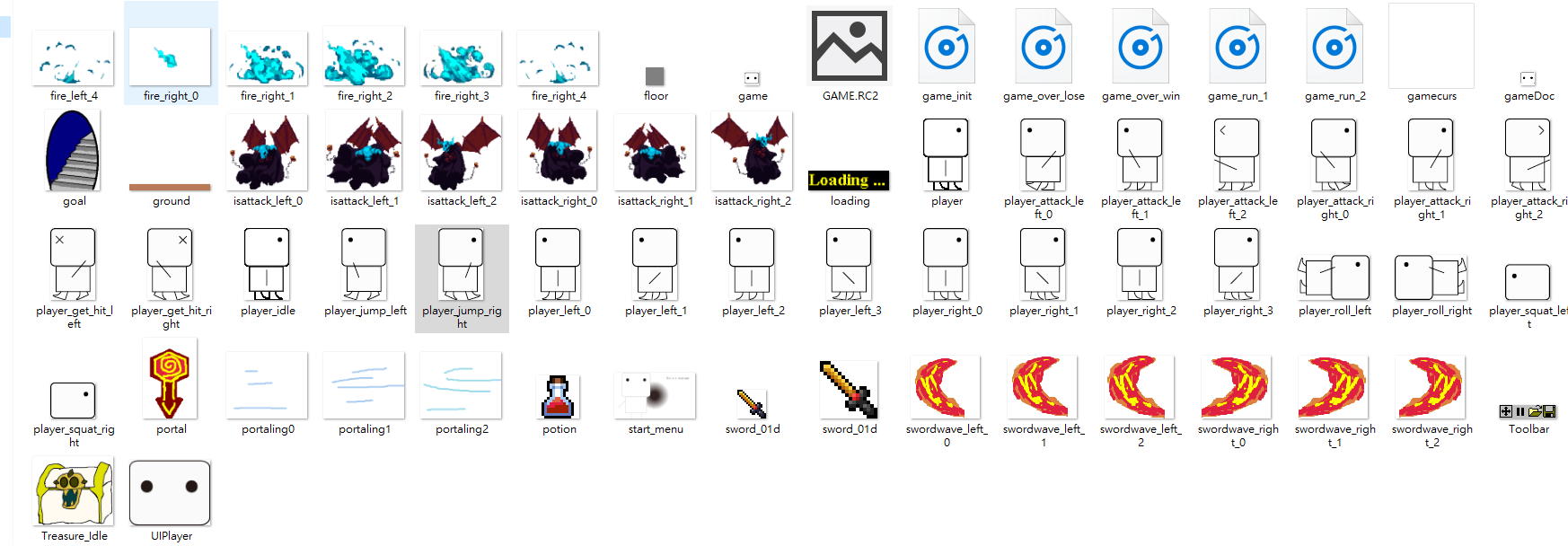
1:切換到武器1

2:切換到武器2

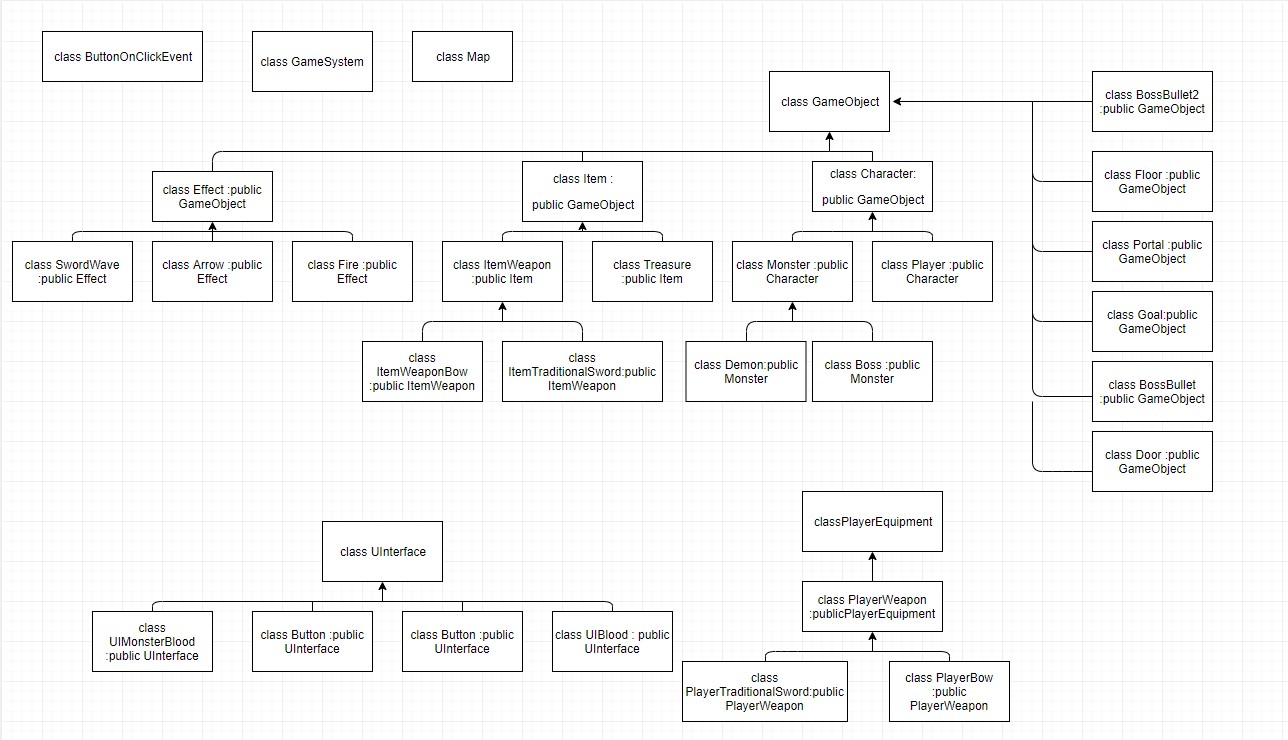
* + 1. 密技:

F:回血100

G:Boss扣血1000

* 1. 遊戲圖形
  2. 遊戲音效

|  |  |
| --- | --- |
| 代號 | 用處 |
| AUDIO\_GAME\_INIT | 開始畫面的音樂 |
| AUDIO\_GAME\_RUN\_1 | 第一關的音樂 |
| AUDIO\_GAME\_RUN\_2 | 第二關的音樂 |
| AUDIO\_GAME\_OVER\_LOSE | 遊戲失敗的音樂 |
| AUDIO\_GAME\_OVER\_WIN | 遊戲勝利的音樂 |

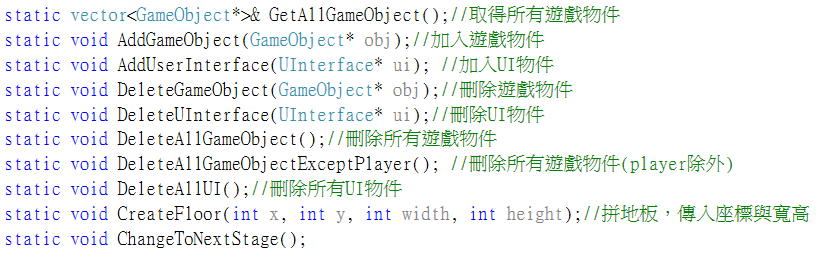
1. 程式設計
   1. 程式架構
   2. 程式類別

|  |  |  |  |
| --- | --- | --- | --- |
| 類別名稱 | .h檔行數 | .cpp檔行數 | 說明 |
| Arrow | 23 | 77 | 繼承Effect，弓箭射出的箭 |
| Boss | 88 | 377 | 繼承Monster，控制地圖上的Boss |
| BossBullet | 36 | 121 | 繼承GameObject，Boss發射的大子彈 |
| BossBullet2 | 31 | 80 | 繼承GameObject，Boss發射的小子彈 |
| Button | 25 | 49 | 繼承 UInterface，起始畫面的按鈕 |
| ButtonOnClickEvent | 33 | 64 | 當Button被點擊時，所觸發的事件 |
| Character | 77 | 298 | 繼承GameObject，具有角色基本的變數 |
| Demon | 24 | 93 | 繼承Monster，地圖上的小怪 |
| Door | 33 | 93 | 繼承GameObject，玩家擊破門時，怪物暈眩 |
| Effect | 38 | 154 | 繼承GameObject，武器、怪物的攻擊特效 |
| EquipedSlot | 18 | 49 | 繼承UInterface，顯示玩家目前的裝備 |
| Fire | 26 | 70 | 繼承Effcet，Demo攻擊得特效 |
| Floor | 19 | 41 | 繼承GameOject，地圖的地板和牆壁 |
| GameObject | 55 | 127 | 為地圖上所有物件的Base |
| GameSystem | 221 | 463 | 負責統一物件運行物件的動作(ShowAni..) |
| Goal | 11 | 21 | 繼承GameObject，到下一個地圖的門 |
| Item | 42 | 133 | 繼承GameObject，掉落物的Base |
| ItemTraditionalSword | 24 | 72 | 繼承ItemWeapon，只有圖片沒有武器功能(劍) |
| ItemWeapon | 27 | 30 | 繼承Item，具有武器素質的變數 |
| ItemWeaponBow | 14 | 67 | 繼承ItemWeapon，只有圖片沒有武器功能(弓) |
| Map | 39 | 110 | 紀錄地圖地形和大小，還有轉換座標的轉換 |
| Monster | 78 | 182 | 繼承Chacter，具有怪物的基本變數 |
| Player | 190 | 863 | 繼承Chacter，控制玩家 |
| PlayerBow | 21 | 48 | 繼承PlayerWeapon，有實質武器的功能(弓) |
| PlayerEquipment | 52 | 127 | 紀錄玩家身上的裝備 |
| PlayerTraditionalSword | 23 | 43 | 繼承PlayerWeapon，有實質武器的功能(劍) |
| PlayerWeapon | 18 | 15 | 繼承 PlayerEquipment，具有功能的Weapon |
| Portal | 22 | 56 | 繼承GameObject，地圖上兩點的傳送門 |
| Potion | 24 | 34 | 繼承Item，回血道具 |
| SwordWave | 19 | 65 | 繼承Effect，劍的攻擊特效 |
| Treasure | 17 | 57 | 繼承Item，地圖上的寶箱 |
| UIBlood | 19 | 59 | 繼承UIterface，顯示玩家血量 |
| UIMonsterBlood | 22 | 31 | 繼承UIterface，顯示怪物血量 |
| UInterface | 44 | 94 | UI的Base class |
| 總行數 | 1453 | 4266 |  |

* 1. 程式技術
     1. Static

遊戲中有系統與地圖兩個class，但整個遊戲中只會有一個系統與一個地圖，因此利用static來確保遊戲中使用到的是同一個系統與地圖。

* + - 1. GameSystem紀錄了所有遊戲中可供所有類別使用的函式或變數。例如:存放所有物件的gameObjectList、取得特定類別物件的函式GetGameObjectsWithType()等。
      2. Map設定了地圖大小，用以確保場景上的物件不會超出地圖邊界。還紀錄了所有地板的位置，可用來進行物件的碰撞判定。以及螢幕左上角的座標，用來換算螢幕上要看到的是地圖中的哪一部份。



▲使用static確保遊戲中使用到的都是同一個GameSystem

* + 1. Inheritance與polymorphism

我們寫了一個base class叫GameObject，裡面有x,y,width,height,layer等所有物件都會有的成員變數，以及SetX()、SetY()、ShowBitMap()等成員函式。場景中每個物件都會繼承GameObject，接著在mygame的OnShow()中，用for迴圈執行每個GameObject的ShowBitMap()，就可以用短短幾行來顯示全部的物件。

* + 1. Virtual destructor與memory leaks

有不少class有new pointer，沒有適當地delete將會造成memory leaks。由於我們每個物件都繼承了GameObject，因此可以透過virtual destructor來delete每個class中new出來的pointer。

* + 1. Layer

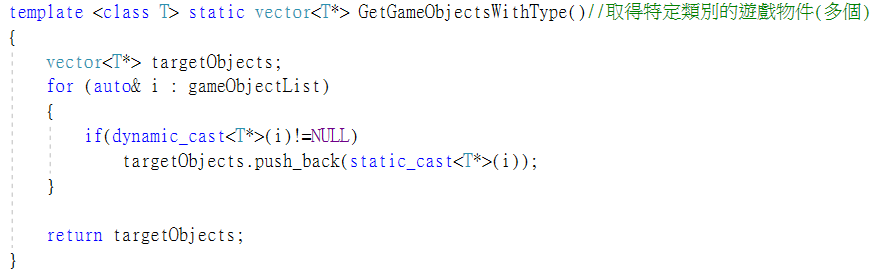
2D遊戲有圖層的問題，哪張圖要在哪張圖的前方是需要管理的。GameSystem中有一個vector叫gameObjectList，用來存放所有GameObject的指標。在gameObjectList插入新物件時，會根據每個GameObject的layer來決定該插入到哪個位置。圖層越高的物件會插入在越後面，這樣在OnShow()用for迴圈執行ShowBitMap()時就會比較晚顯示，就能顯示在越前方。

* + 1. Template與Dynamic\_cast
       1. Template

同一個函式可以利用template丟入不同型態的參數，省去重複撰寫的麻煩。GameSystem中的GetGameObjectsWithType，就是利用了template的特性，可以取得特定類別的所有物件。例如要取得所有的「怪物」，可以這樣寫:GetGameObjectsWithType<Monster>()，即可取得Monster、Demon、Boss等繼承了Monster的子類別物件。

* + - 1. Dynamic\_cast

在GetGameObjectsWithType中，利用了dynamic\_cast轉型失敗會回傳NullPtr的特性，可以找到特定類別的物件，以及繼承了這個類別的子類別的物件。用iterator對gameObjectList做迭代，接著做動態轉型dynamic\_cast<T\*>(iter)，若這個物件不是T的子類別，則不會回傳，反之則回傳。



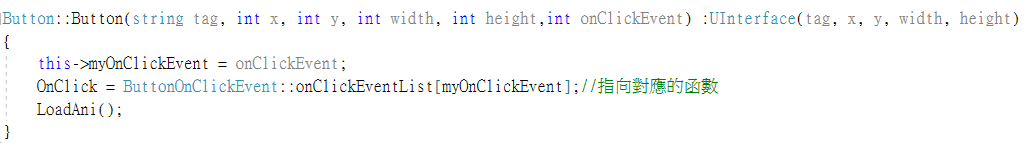
▲使用template與dynamic\_cast來取得特定類別的遊戲物件

* + 1. Enum

每個物件可能有多種動畫，例如左走、右走、上跳、攻擊等，因此將每個動畫存入vector，並透過enum來記錄每個動畫的編號，來決定該播放哪個動畫。

* + 1. Button、ButtonOnClickEvent與Function Pointer

每個Button都有x,y,width,height等基本的成員變數，但每種按鈕都有各自的OnClickEvent，所以除了Button外，還寫了一個class:ButtonOnClickEvent。ButtonOnClickEvent定義了每種按鈕的按下時的事件，而Button中定義了一個函式指標OnClick，會指向ButtonOnClickEvent中對應的事件。這樣子，Button的class就會很簡單，只需要x,y,width,height,以及OnClick，就能做出不同功能的按鈕。



▲Button只要在constructor指向對應的OnClick事件即可做出不同的效果。

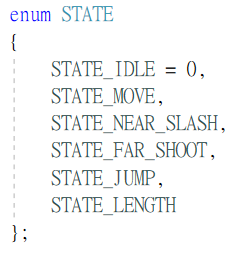
* + 1. Random

遊戲中有許多地方用到亂數，例如怪物掉落的裝備數值、每項道具的掉落機率等。亂數用到了C++的<Random>與<Functional>函式庫，利用這些函式庫就不用自己設定seed，減少了出錯的可能，而且產生的亂數真的很「亂」。

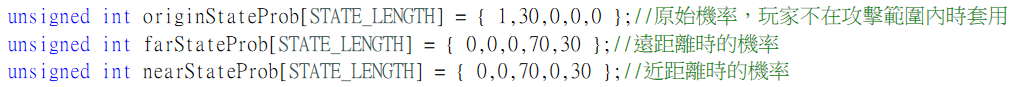
* + 1. Probability

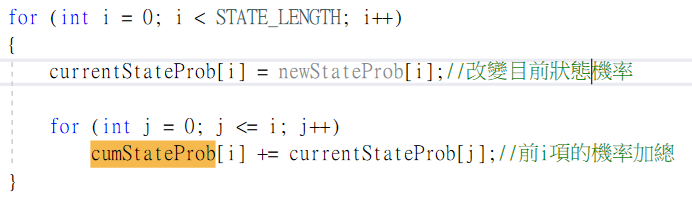
Boss的各項行動是隨機的，且每個動作的機率是不同的，這些機率又會因為與玩家之間的距離而產生變化。

首先定義Boss有幾種動作，再來用一個陣列紀錄每種動作各別的機率。接著再用一個陣列，將前i項的機率加總。最後使用Random產生0~機率總和之間的亂數，看這個亂數落在哪個區間，則執行哪個動作。

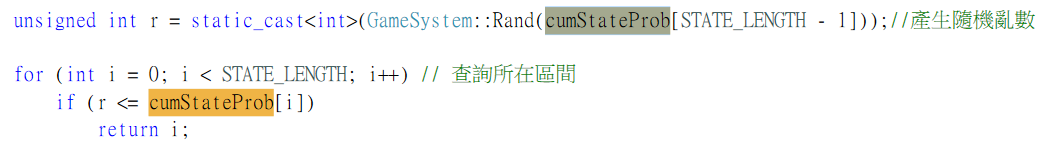


▲定義每種狀態

 ▲為每種狀態設定不同的機率，數字越大機率越高



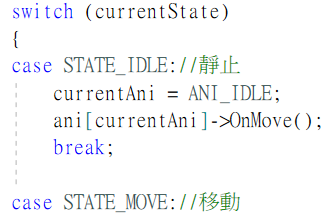
▲將前i項機率加總



▲查詢所在區間

* + 1. State Machine

狀態機能使設計變得簡單，Boss的各項動作便是使用狀態機。首先用enum定義各項狀態，接著再用另一個int變數currentState來記錄目前在哪個狀態，最後使用switch(currentState)來決定要執行哪個狀態。



▲透過switch(currentState)來執行各種狀態

1. 結語
   1. 問題及解決方法
      1. Template在.cpp會有無法解析的外部符號

將template寫到.h即可。

* + 1. 實體轉型會造成錯誤

將變數型態改為指標，即可轉型成功。

* + 1. 一個物件有很多種動畫，該怎麼儲存?

用Vector存動畫，用enum當編號。

* + 1. 碰撞偵測怎麼做?

用for迴圈判斷每一個x,y座標，有沒有在另一個物件的範圍內。

* + 1. 撿起裝備的方法?

裝備撿起後，裝備應該要「屬於」玩家，而不是「一件裝備」與「一個玩家」顯示在同一個座標。因此對地上的裝備道具按下互動鍵後，利用operator=，在玩家身上新增一個數值一模一樣的玩家裝備，並將地上的裝備道具刪除，即可讓裝備屬於玩家。

* + 1. 該怎麼設計按鈕的OnClick事件?

最初的按鈕的OnClick是設計成Virtual Function，想要有不同的OnClick就新增一個class繼承按鈕。但後來覺得，每個Button只有OnClick會不一樣，要因此而多寫一個class顯得有些多餘。最後決定將OnClick改為函式指標，然後再另外設計一個class ButtonOnClickEvent，讓函式指標指向其中的一個Function，即可只用兩個class做出不同的OnClick事件。

* 1. 時間表

|  |  |  |  |
| --- | --- | --- | --- |
| 週次 | 郭家佑 | 廖彥澤 | 說明 |
| 1 |  |  |  |
| 2 |  |  |  |
| 3 |  |  |  |
| 4 |  |  |  |
| 5 |  |  |  |
| 6 |  |  |  |
| 7 |  |  |  |
| 8 |  |  |  |
| 9 |  |  |  |
| 10 |  |  |  |
| 11 |  |  |  |
| 12 |  |  |  |
| 13 |  |  |  |
| 14 |  |  |  |
| 15 |  |  |  |
| 16 |  |  |  |
| 17 |  |  |  |
| 合計 | XXX | XXX |  |

* 1. 貢獻比例

郭家佑50%

廖彥澤50%

* 1. 自我檢核表

|  |  |  |  |
| --- | --- | --- | --- |
|  | 項目 | 完成否 | 無法完成的原因 |
| 1 | 解決 Memory leak | ☑已完成 □未完成 |  |
| 2 | 自定遊戲 Icon | ☑已完成 □未完成 |  |
| 3 | 全螢幕啟動 | ☑已完成 □未完成 |  |
| 4 | 有 About 畫面 | ☑已完成 □未完成 |  |
| 5 | 初始畫面說明按鍵及滑鼠  之用法與密技 | ☑已完成 □未完成 |  |
| 6 | 上傳 setup/apk/source 檔 | □已完成 □未完成 |  |
| 7 | setup 檔可正確執行 | □已完成 □未完成 |  |
| 8 | 報告字型、點數、對齊、行  距、頁碼等格式正確 | □已完成 □未完成 |  |
| 9 | 報告封面、側邊格式正確 | □已完成 □未完成 |  |
| 10 | 報告附錄程式格式正確 | □已完成 □未完成 |  |

* 1. 收穫
     1. 106590025 資工二 郭家佑

學到了template、dynamic\_cast、static、function pointer、random等，以前比較沒機會用到的語法。也更加熟悉polymorphism、inheritance等物件導向的精髓功能。

106590018 資工二　廖彥澤

學到如何與別人一起完成一個project，還有物件導向的概念，例如：

繼承、多型．．．等等，必須將一個物件包裝好，讓隊友能簡單使

用，才能保持project的架構和效率。

，

* 1. 心得、感想

106590025 資工二 郭家佑

透過這堂課，學會了多型與繼承的用法，實際地應用了物件導向的觀念與語法。也明白了如何解決memory leak，new完後要記得delete，若是在class的member variable做了new，那要記得在destructor進行delete。

由於是分組作業，因此還學會了github的用法，如何push、pull與merge。

也明白了與組員溝通協調的重要。如果自己負責的部分與組員衝突到，該怎麼辦?base class有功能可以新增，該由誰來更新?誰要負責寫哪個部分?不論如何，最重要的還是要互相溝通，不然最後兩個人的程式大雜燴，會使得程式不易維護。

106590018 資工二　廖彥澤

經過了這堂課的學習，在專案讓用到了許多繼承、多形……等等物件導向的觀念，除了在技術上的使用，在寫程式了邏輯也變得非常注重，

若程式不夠簡潔，就會寫出許多的垃圾code，在易讀性會大幅降低，

在維護上也會更加困難；在與隊友的合作、討論也相當重要，在大家初次練習的情況下，在遊戲初期，我們採取分開分工的方式，各自負責較不相關的class，讓遊戲快速有個架構，也比較不會混亂，在後期的中後期的時候再開始討論class之間該如何運作，整體來說算是相當順利。

* 1. 對於本課程的建議

Framework Bug很多，又很舊，請好好維護與更新。

附錄