

商管程式設計期末專案 第二組

經濟二 B07801008 吳謹妤

經濟二 B07303132 羅紹博

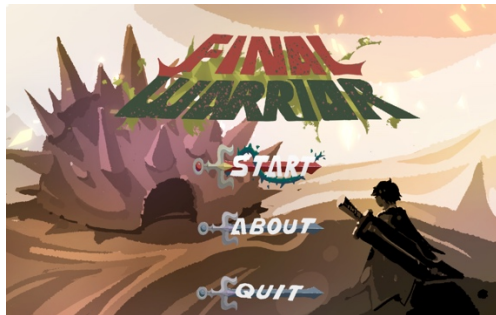
農經一 B08607017 黃子雍

經濟二 B07303058 張祥賢

經濟二 B07303024 李品樺

壹、 主題構思

我們的遊戲名稱叫做「Final Warrior」，是一款畫風細緻、介面精美、操作流暢的生存遊戲。當化身遊戲中的「Final Warrior」後，一場驚險又刺激的冒險就此展開，你能在末世魔王頻頻的猛烈攻擊中，一邊閃躲由四面八方襲來的猛烈砲火，一邊蒐集更高的積分嗎？在和魔王的對戰中快敗下陣來也沒關係，注意場上隨時出現的道具們，或許它們就是讓你絕處逢生的關鍵喔！



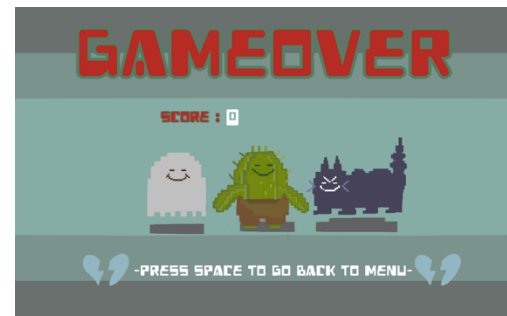
圖一：選單介面



圖二：操作介面



圖三：遊戲進行畫面

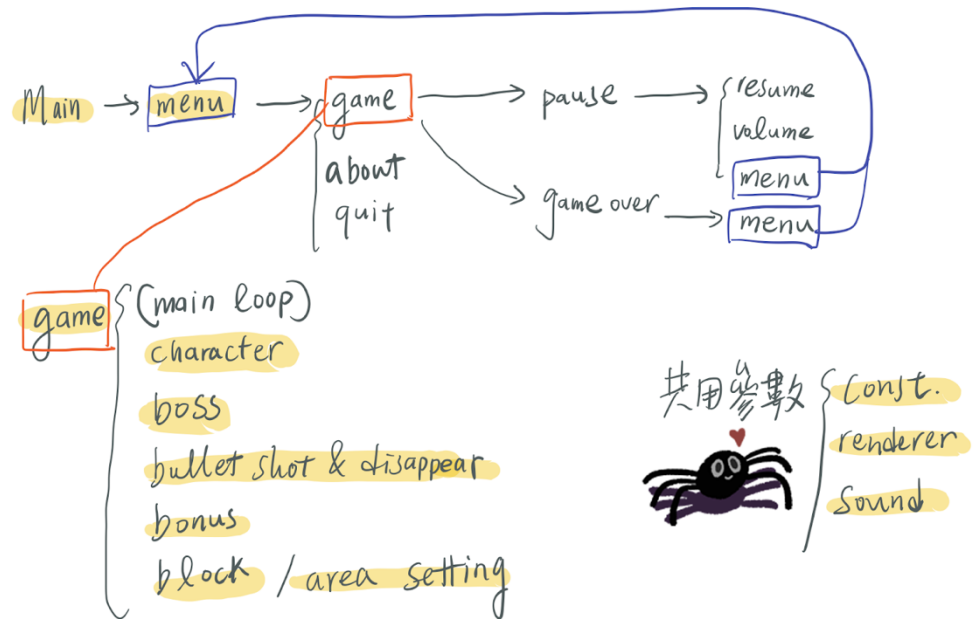


圖四：Game over 以及分數結算

貳、 系統設計

我們使用了以下兩個 python 模組來製作遊戲：

- **pygame**：視窗顯示、時間計算、遊戲主要架構。
- **random**：處理隨機事件。例如產生道具、子彈的軌跡等等。



為了增進分工以及除錯的效率，所以我們將遊戲的程式分成 11 個檔案。
各檔案的功能分述如下：

- **main**：用於呼叫 menu。
- **menu**：建立主選單。
 menu 為一個 class，在建立這個 class 時就會初始化 pygame，並且賦予 button、click 初始值 0。button 用於記錄使用者現在選到哪個選單，由於飛入的動畫只需要在回到主選單的時候出現一次，因此我們用 click 控制現在是否需要顯示飛入的動畫。若使用者什麼都還沒有按下的時候，就要顯示飛入動畫。在顯示選單位置的部分，我們 import 了 const 這個檔案，用於取得使用者目前的螢幕長寬比例。我們將三個選單按鈕水平置中，高度分別顯示在使用者螢幕的 2/5、3/5、4/5，以達到舒適且沒有壓迫感的顯示畫面。當使用者的按了鍵盤的「上」，click 便會減一，按了「下」，則會加一。我們利用此方法來判斷現在使用者應該是選到哪一個選單。的初始值其中有三個函數，分別是 main_menu、intro、intro_p2。一開始 menu 被呼叫到時會進入 main_menu 的函數中，並且顯示底圖以及飛入不同選單的按鈕。飛入動畫的部分在下一節的演算法會詳細介紹。當使用者選到 about 這個選項時，則會進入 intro 這個函數中。若在 intro 中按下右鍵，則會進入 intro_p2 函數中，也就是底圖顯示團隊成員的頁面。再按下 esc 時則會回到 main_menu 的函數中。
- **game**：遊戲的主程式。用於呼叫 character、boss 等等的檔案。
- **character**：利用 class 建立主角屬性和限制，並引入成 pygame 裡面的 sprite。主角需要設定的屬性有：
 1. 起始生命值

2. 從何處引入圖片，並且調整大小
3. 主角獲得的分數
4. 角色初始位置
5. 角色的移動及速度(不斷取得角色現有位置)。

限制：

1. 利用 `spritecollide` 來判斷角色是否到達邊界，如果有則速度歸零。
 2. 判斷角色現有血量是否等於零，如果低於零則遊戲結束。
- **boss**：利用 `class` 建立魔王的屬性。一樣利用 `pygame` 引入 `sprite`，並且利用 `pygame` 抓取圖片所在位置。在這裡只要記得設定魔王的 `x, y` 是變數，以利之後地圖滾動時，魔王位置的呈現。
 - **bullet_shot and disappear**：利用 `class` 建立子彈的屬性和各種移動的軌跡，並引入 `pygame` 裡面的 `sprite`，將子彈射為一個群組，並在這個檔案設定子彈移動模式。
 - **bonus**：建立 `bonus` 物件，道具有以下屬性：
 1. 隨機生成的座標
 2. 由不同權重決定的道具類型
 3. 消滅時間
 - **block**：建立 `block` 物件，並使用了 `pygame` 內建的矩形，屬性包含 `x` 座標、`y` 座標、高度、寬度，這個物件是為了底下的 `areaing_setting` 做準備。
 - **area_setting**：利用 `block` 物件，以矩形的方式拼接出邊界的範圍，如下圖所示，藍色部分為連接起來的邊界。

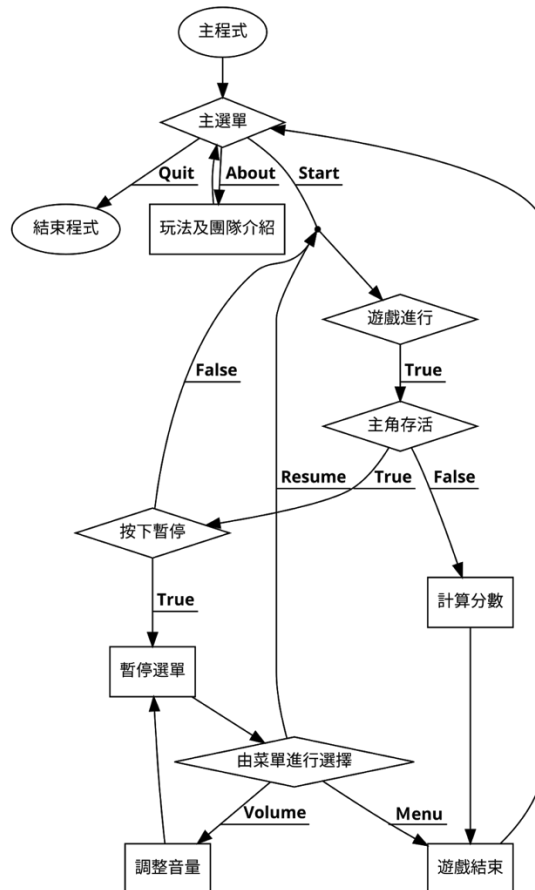


以下則為共用的參數：

- **const**：顧名思義，當中記錄了遊戲運行中的所有「常數」設定。因為遊戲以全螢幕遊玩，所以紀錄了使用者螢幕的長度和寬度，並且以這個長寬為資訊，計算出遊戲中的按鍵位置，例如開始遊戲、暫停鍵、血量等等的位置。另外常會使用到的色碼，我們預先透過「字典」的方式儲存，之後取用會更加直觀便利。遊戲主角的起始座標、子彈的速度等等，都在這邊賦予初始值。

- **renderer**：所有和「螢幕顯示」相關的函數都放在這個檔案當中，在主要的程式中取用，例如，「draw_....」的函數負責顯示各式各樣的圖片，例如人物血條、暫停鍵、分數、遊戲暫停、遊戲結束等等
- **sound**：利用 `pygame.mixer.music` 進行音訊處理。此檔案被呼叫時便會讀入各種音效、背景音樂。什麼時候需要播放音樂則是在 `menu` 和 `game` 中呼叫 `sound` 檔案，而並非從 `sound` 來控制，以避免發生互相 `import` 的問題。

以下是程式的運行流程圖：

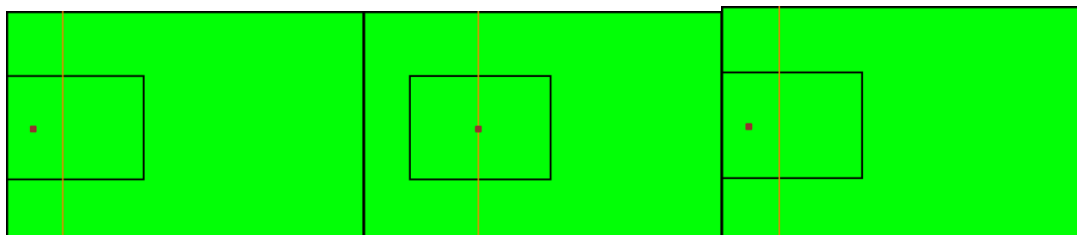


參、演算法

針對一些程式中比較有趣的功能，我們再深入闡述解釋它們的演算法。

● 地圖滾動

以下以 x 方向進行說明， y 方向的邏輯相同。紅點為玩家的座標，黑色方框為使用者的螢幕尺寸，而綠色方塊是背景圖片的實際大小。有以下三種情況，首先，若「螢幕的中線」和「左邊界」的距離小於「螢幕寬度的一半」，此時就從背景圖片的最左側開始顯示。其次，若「螢幕的中線」和「右邊界」的距離小於螢幕寬度的一半，此時就從圖片的最右側開始顯示。若不是以上兩種情況，就直接用人物的座標來判斷顯示的區域即可。



- 人物動畫效果

為了讓人物的移動能更加順暢的效果，我們利用動畫的原理，將人物的四個方向分別個畫了四張分鏡圖，並存在各自的 `list` 裡，在人物移動的過程中循環播放，以此達到流暢的移動效果，具體的移動方式如下：(為了方便說明，以下都以「向上」前進來說明，其餘三個方向的原理相同。)

由於按下 `event.key == const.key["up"]`時，會改變角色的移動速度(`change_y > 0` 往上)，我們利用 `change_x`，`change_y` 的正負 來判斷應該要從哪一個字典抓取圖片。當持續按上時，我們利用計時器 `tick % 4` 來決定是 `list` 裡的哪一張圖片。此外，為了讓圖片不會在放開方向鍵消失，我們做出一個會不斷刷新，內容只有一個的字典(每刷新一次，取代掉上一個 `value`)來記錄最後的位置。

- 選單按鈕的飛入效果

我們透過以下演算法的實踐，以達到選單飛入動畫的效果。

`while True:` # 在 `menu` 的 `main loop` 中執行選單滑入的動畫

`if` 使用者沒有按過鍵盤:

`if` 現在位置 > 最終位置: # 若現在的 `y` 座標不是最終的 `y` 座標

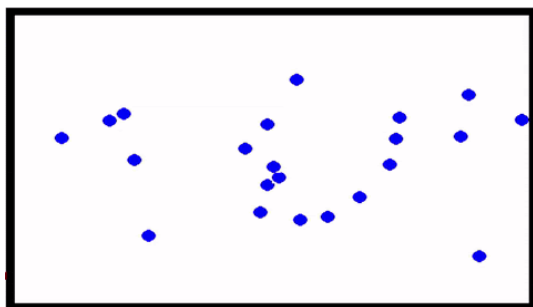
 位移 = (現在位置 - 最終位置) * 0.05 # 使每一次刷新的位移遞減

 現在位置 -= 位移

- 子彈的算法

- 天女散花：設定初始子彈移動角度，從 0 度開始每 45 度出現一顆。

- 隨機：固定初始位置，然後利用 `random` 和 `math` 的函數使得子彈每次出發的角度都不一樣，如下圖所示。



- 八卦陣：預先設定初始子彈移動角度，先做從零開始每 45 度一顆總共 8 顆，再一次出現向四周移動。

- 心型：將圓分成上下兩部分，各 0 度從上下加減 45 度一次出現兩顆，看起來就會像是心型。

- **bonus 出現機率的算法**

bonus 具有「**index**」的屬性，透過 **random** 模組隨機生成 0 到 120 間的數字。當 **index** 介於 0 和 60 之間，就生成「分數」這個 **bonus**；當 **index** 介於 60 和 80 之間，生成「加速」的道具；當 **index** 介於 80 和 120 之間，生成「補血」的道具。透過這種方式來給予三種道具不同的權重。此外，為了讓場上一直有「分數」的道具，當玩家吃掉一個分數之後，會立刻生成另一個「分數」道具；另外若人物目前是滿血的狀態，就不會生成「補血」道具。最後，為了增加遊戲難度，**bonus** 有內建一個計時的屬性，會隨著 **pygame** 內的時鐘計算時間，我們為每個 **bonus** 設定了兩個時間臨界值，當達到第一個臨界值，道具會開始閃爍，提醒玩家道具即將消失；當到達第二個臨界值，才讓道具真正消失。

- **碰撞測試**

pygame 內建很方便的碰撞測試分別有 **spritecollide**(物件，物件 **group**，**group** 要不要消除)、**groupcollide**(物件 1**group**，物件 2**group**、物件 1 要不要消除，物件 2 要不要消除)以及 **spritecollidencency**(物件 **group**，物件，物件要不要消除)。

遊戲中運用到碰撞測試的地方主要有五個。

1. 子彈碰撞到地圖邊界時，會自動消失，以此來降低 **cpu** 的負擔。
2. 主角被子彈碰到的時候，子彈消失且主角扣 0.5 格血。
3. 隨機生成的道具萬一出現在邊界的地方，要觸發碰撞並且消失。
4. 在碰到邊界時兩個物件都不要消失，但是回傳一個布林值讓主角速度歸零。
5. 主角在碰撞到隨機道具時道具要消失，並且回傳一個布林值觸發效果和音效

肆、 分工方式

展示影片錄製：李品樺、吳謹好、張祥賢、黃子雍、羅紹博

我們將整個程式簡略分為三個部分，並且使用 **GitHub** 進行版本控制：

1. 遊戲選單：李品樺、吳謹好
2. 主遊戲：張祥賢、黃子雍、羅紹博
3. 暫停選單：吳謹好

伍、 心得感想

吳謹好：一開始決定要寫遊戲時其實還蠻擔心的，因為這跟我們這學期所學的、所寫過的作業，是完全不同方面的應用，我們得自己先研究過 **pygame** 模組後才得以慢慢建構出一個遊戲.....不得不說這真的是一項十分浩大的工程啊（汗）。這次真的很幸運

可以跟四個很單的組員同組，很感謝有他們，我們才能順利寫出如此精緻又酷炫的射擊生存遊戲！在進行期末專案的這段期間，我覺得最困難的地方在於熟悉別人已經建好的模組，並使用這個模組寫出我們自己的程式。一開始為了弄懂 `pygame` 的運作，找了很多網路上的教學影片來看，但依舊似懂非懂，即便是負責最不複雜的部分，寫的時候依舊常常寫不出預期的效果，因此十分挫折。幸好經過每週的討論，以及大家的幫忙和分工，才讓我能順利完成我的部分，在這邊再次感謝其他四位組員！從小到大有很多打遊戲的經驗，但眼睜睜看著一個從無到有被創造出來倒是頭一次，因此至今依舊覺得十分不可思議！除了感到神奇之外，一想到光是用 `python` 寫出一個陽春的 `Final Warrior` 就讓人耗盡半條命了，更何況是那些我們常看到的電玩遊戲呢？這次期末專案的「洗禮」讓我完全了解要寫出一個遊戲是多麼不易的一件事，另一方面，我也開始打從心底尊敬遊戲工程師們，所以，請...請收下我的膝蓋吧！

羅紹博：感覺有點處於半夢半醒的狀況下完成了這門期末報告，因為跟之前的上課內容沾邊兒的地方好像只有 `class` 阿，唯一相同的只有平常被 `pdogs` 追殺的那種感覺阿。感覺最累的部分就是在事前學習如何使用 `pygame` 的功能和工作銜接了。因為 `pygame` 的功能太多，除了基本的介面架構以外，還有時候不太能找到自己需要的功能該如何去完成會比較有效率。工作銜接的部分則是因為，每個人在做的部分或多或少都有一點重和或需要配合的地方，但這時候如果引用的函數不太一樣就可能需要再花時間去了解到底為什麼要這樣寫，改動之後還要注意不會跟區塊的名稱不一樣(要一直檢查各個不同的 `py` 檔阿)。發現自己在 `debug` 方面非常的疲弱(其實是整個打 `code` 都很疲弱，非常感謝組員的包容和協助，有時候自己負責的功能做不太出來只能拿著打一半的 `code` 去求救的時候，真的是覺得十分汗顏。第一次嘗試做出遊戲，雖然自己只佔了非常微小的貢獻，但看到最後出來的成品還是覺得非常的開心，也看到了很多同學給出了正面的反饋，讓人有種一切都很值得的感覺。

黃子雍：這學期做得最開心的報告！一開始決定要做遊戲還有點小擔心，因為之前完全做過遊戲，也沒用過 `pygame` 這個函式庫，所以我們一開始花了不少時間在分頭熟悉模組的各種功能，我們一開始想，就各自找 `google` 解決被分配到的任務，最後再把所有功能一起加在一個起，不過我們馬上遇到一個困難是每個人寫的變數命名都是不一樣，而且因為檔案龐大、參數又各寫個的，以致在 `debug` 的過程就變得相當複雜而且難以執行，所以我們之後就決定改成要有共用參數跟螢幕顯示等會一起用的檔案最後再 `import` 到 `game` 使用這樣才解決問題～ 這次專案真的讓我理解寫一個好看的 `code` 有多麼重要 XD 然後每個組員都超 `carry`，每個人都能完成各自分工的工作，還能有很多創意的點子，最後才能產出這麼棒的作品，辛苦大家了，讚讚

張祥賢：這是第一次組隊做專案，儘管以前就學過 `python`，但以前沒有寫過這麼龐大的程式過，中間一直感受到強烈的厭世感。大家決定要用 `pygame` 寫遊戲，`pygame` 對我們來說是完全陌生的模組，所以我們原先的策略是讓組員分別實現被分配到的功能，

再來合併。但很快我們就發現程式碼出現很多重複的部分，討論的時間基本都花在合併和除錯，驚覺如果按照這樣的步驟寫下去之後，一定會 **debug** 到死去，所以後來我們重新梳理了程式的架構後，再把程式拆解成了許多檔案，透過引用檔案的，並盡可能地用 **function**、**class** 來組織我們的遊戲，才讓之後的流程順利了很多。學到最多的部分是，可以處理像這次專案有十幾個檔案、上千行的程式碼，對於一個程式的完整架構有更深的瞭解，此外過程中常常需要上網爬文學習不會寫的地方，對於這領域資訊檢索的能力進步了不少。最後想謝謝辛苦的組員們，一起熬夜爆肝做出完成度蠻高的作品。

李品樺：雖然之前已經有做過期末專案的經驗了，但是一開始在進行這次的期末專案的時候遇到了許多問題。由於我們原本是先各自分開寫檔案，大家都用了很多不同的方法在寫同一個功能，但是卻發現這樣在合併的時候會非常麻煩。所以後來我們才趕快重新想一個架構，才讓整體的脈絡比較清楚一點，也比較了解現在真正要做的事情是什麼。一開始總覺得很慌，感覺永遠都做不完，完成度也一直和我所預期的有點落差。後來再經過了許多個與大家奮鬥的夜晚之後，終於整體遊戲算是完整的了。到了後期再加上了美術設計的插圖以及整體的遊戲優化之後，感覺看起來就非常不錯了！在做期末專案的時候透過搜尋以及閱讀網路上許多的資料，以及學習向組員求救、溝通、討論，也讓我學習到許多課堂中得不到的知識和經驗。最後，很謝謝組員們的努力，大家都能夠配合有點小小強迫症的我趕在 **ddl** 的前一週全部做完，真的辛苦大家了～能夠有這次的合作經驗，做出一個自己也滿意的期末專案真的是一件很有成就感的事。