

國立台北科技大學

2023 Spring 資工系物件導向程式設計實習

期末報告

Survivor.io



第 32 組

110590450 歐佳昀 110590032 詹採晴

目錄

一、 簡介

1. 動機.....2
2. 分工.....2

二、 遊戲介紹

1. 遊戲說明.....2
2. 遊戲圖形.....4
3. 遊戲音效.....8

三、 程式設計

1. 程式架構.....8
2. 程式類別.....9
3. 程式技術.....9
 - (一) 關卡切分9
 - (二) 物件採取回收再利用概念10
 - (三) 物理引擎計算武器移動軌跡11

四、 結語

1. 問題及解決方法.....11
2. 時間表.....12
3. 貢獻比例.....13
4. 自我檢核表.....13
5. 收穫.....13
6. 心得或感想.....14
7. 對於本課程的建議、想說的話.....15

一、 簡介

1. 動機

Survivor.io 是一款現今十分熱門的生存類遊戲，特色在於其擁有多元的遊玩性。像是主線必須躲避殭屍吃到寶石，以及擁有相當豐富的武器攻擊方式，還有不同的 boss 攻擊方式。因此我們選擇了此遊戲作為我們這學期的遊戲實作主題。

2. 分工

本次專案組員有 110590045 歐佳昀、110590032 詹採晴，以下是工作分配表：

組員	工作內容
歐佳昀	主程式運行與架構、遊戲操作、部分敵人攻擊、遊戲控制功能、武器選擇介面等。
詹採晴	開始與結束界面、部分敵人攻擊、遊戲物件互動功能、UI 設計等。

二、 遊戲介紹

1. 遊戲說明

我們遊戲屬於生存遊戲類別，最主要的目的是打擊與閃避怪物攻擊，而當主角血量歸 0 後，進入死亡畫面，未歸零則可以繼續闖關直至遊戲結束。

遊戲主要使用滑鼠操作，點擊初始畫面左下角的對話框會彈出遊戲說明，進入遊戲後，拖曳畫面中象徵操縱桿的氣泡即可控制主角朝向各個方位移動，接觸寶石並達到一定數量後隨即觸發選擇武器功能界面，如遇上 Boss 關卡但未選擇武器則視為放棄機會，強制跳轉畫面。

遊戲一共分成三個大關與三個小關，由前至後依序為：

Stage[0] ▶ Boss[1] ▶ Stage[1] ▶ Boss[2] ▶ Stage[2] ▶ Boss[3]

其中，Stage 為一般小關卡，只要於時間內存活便可進入下一關；Boss 則屬於特殊關卡，需將主要目標打倒後，方可進入下一階段，遊戲中的武器種類如下：

- Lightning：呈現菱形狀軌跡移動，隨機出現不同的狀態，新增後會根據不同的時機呈現不同的狀態。
- Brick：呈現拋物線移動，具有四個拋擲方位，每次會隨機朝向其中一個方位移動。
- Guardian：以角色為中心呈現圓周移動。
- Heal elixir：增加 3500 的血量。

並且，我們有新增兩個作弊鍵，皆位於畫面右上角：

- 電腦圖標：點擊後出現藍色螢幕即可開啟不死模式，再次點擊可退出此模式。
- 右箭頭圖標：僅於 Boss 關卡內出現，點擊後即可跳關。

2. 遊戲圖形

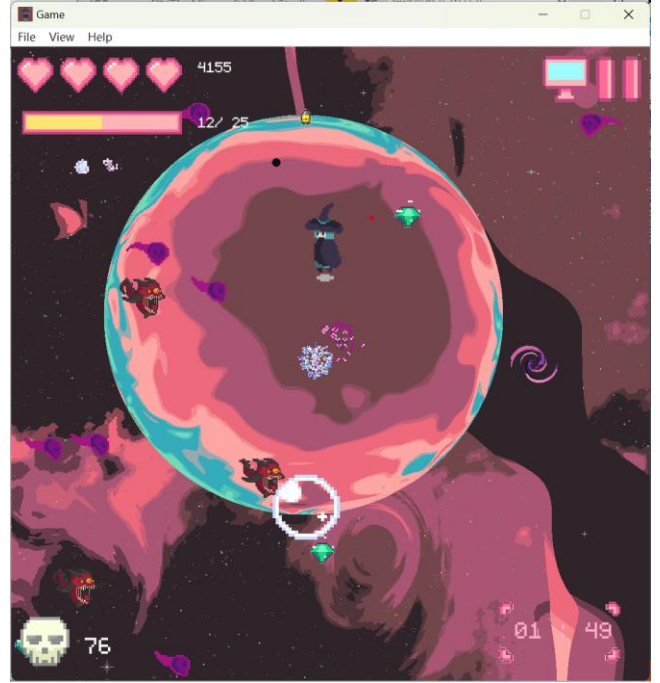
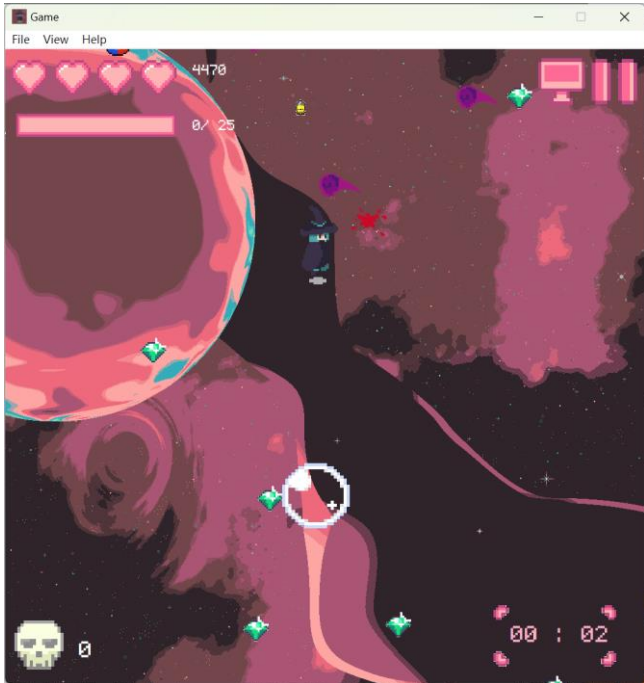
- 初始畫面（選擇背景畫面）



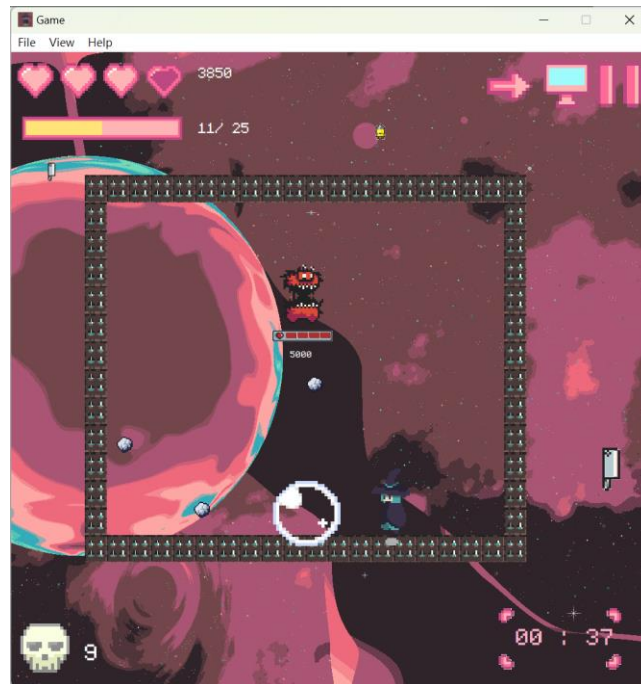
- 遊戲說明



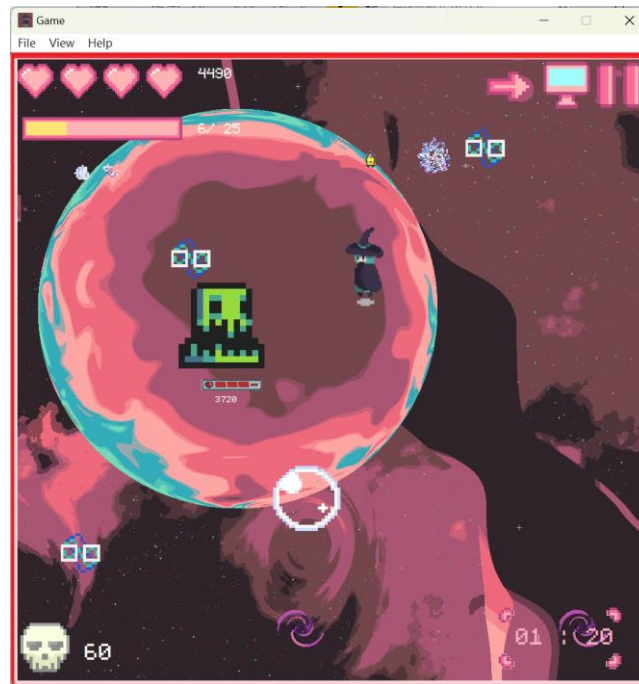
- Stage[0]、Stage[2] (節録)



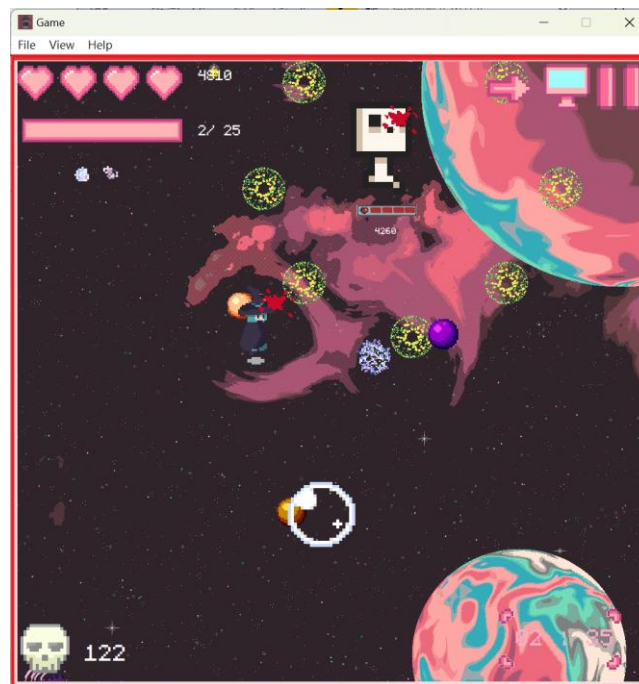
- Boss[1]



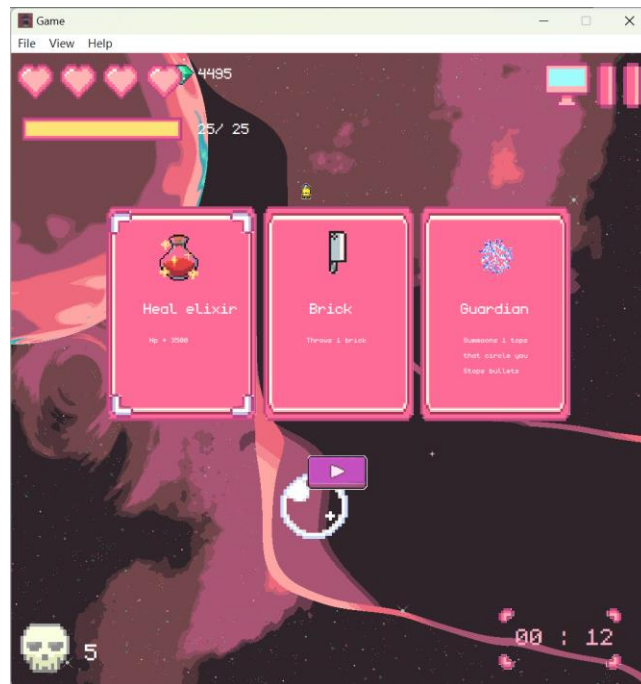
- Boss[2]



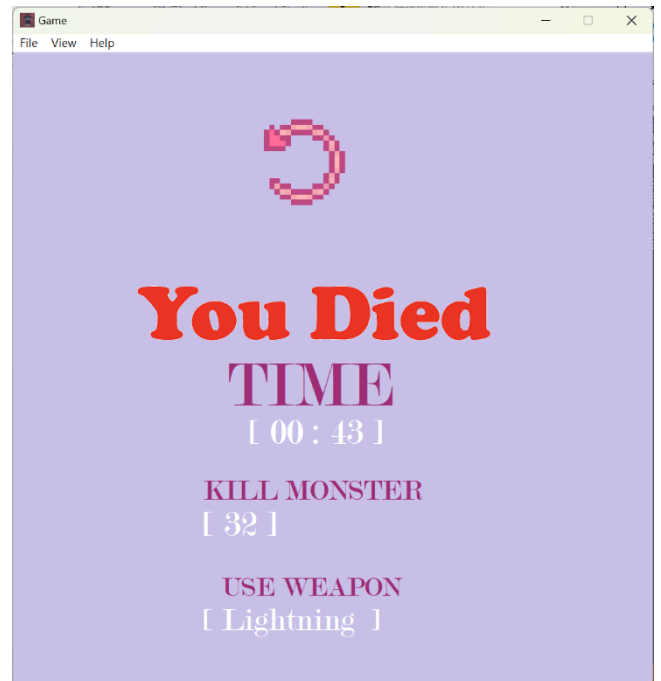
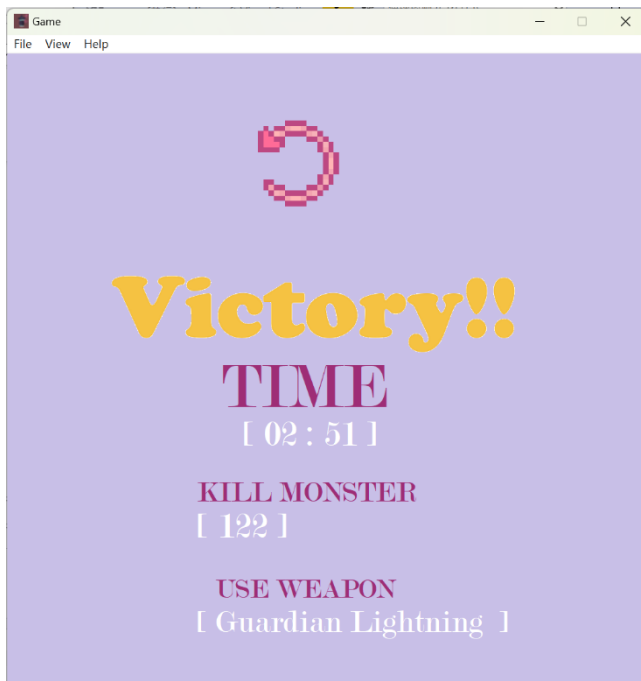
- Boss[3]



- 武器選擇介面（武器選擇隨機出現）



- 勝利畫面、死亡畫面



3. 遊戲音效

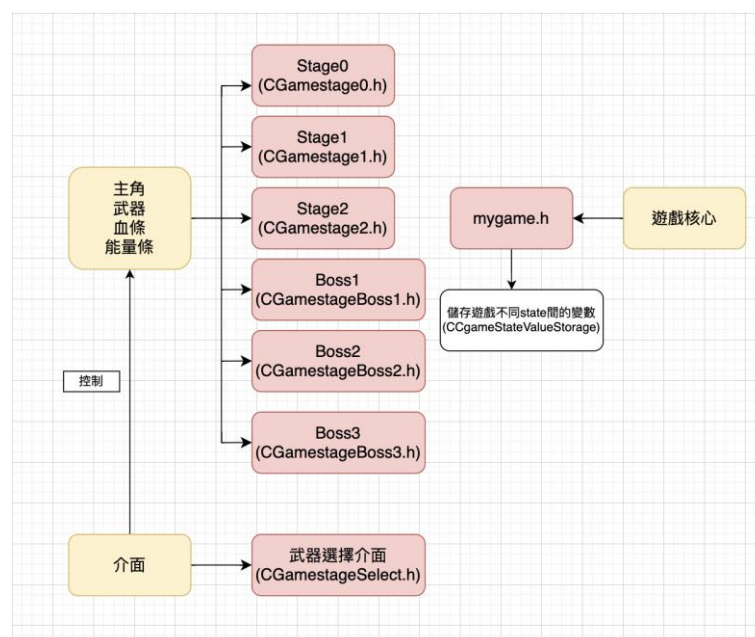
以下陳列在遊戲中使用到的音效。

音效類別	音效檔案
初始畫面	Select.wav
背景音樂： Stage[0], [1], [2] Game Over	Goblins_Dance_(Battle).wav
背景音樂： Boss	battle.wav
主角攻擊音效	bubbles.wav

三、程式設計

1. 程式架構

在這次實習，我們主要將類別分成角色及其相關物件（武器、血條、能量條）、介面、遊戲核心三大類。



2. 程式類別

類別名稱	.h 行數	.cpp 行數	說明
CGamestage0	134	935	實作 Stage0
CGamestage1	144	1094	實作 Stage1
CGamestage2	147	1204	實作 Stage2
CGamestageBoss1	100	558	實作 Boss1
CGamestageBoss2	105	641	實作 Boss2
CGamestageBoss3	110	707	實作 Boss3
CGamestageSelect	57	286	實作武器選擇介面
mygame	225	1058	遊戲主核心
CGameStateValueStorage	19	48	傳遞不同 state 之間的數值
總行數	1041	6531	

3. 程式技術

我們的遊戲已經還原了這個遊戲的幾個重要且具特色的攻擊模式，比如拋磚塊，閃電、飛鏢攻擊等，基礎功能像血量、能量計算、計時器等也已施工完畢。

而在此次實作中，我們在物件的切分關卡與行動軌跡上著墨較多，以下是我們使用到的幾個技術：

(一)關卡切分

因各個關卡所需要呈現的內容不同，故我們將每一個關卡包成一個類，而每個類分別掌管著一個關卡的怪物行動模式、出現物件等等，之後，我們將關卡物件交由遊戲核心迴圈掌控出現時間，可以有效掌控每個關卡應該出現的時機，以及其對應遊戲迴圈所使用的函式，像是 OnMove() 就會觸發對應關卡物件的 OnMove()，來呈現特定關卡物件

的軌跡移動。

並且，由於我們發現關卡物件之間會有需要資料流通的成員(腳色、武器……等)，故我們使用指標，以暫時出借記憶體位置的方式，賦予每個 class 可更改同一個物件數值的權利，同時，因為我們有控制好，同時間會對記憶體產生數值變更的關卡只會有一個，不會有兩個關卡物件互相打架的情況，故可以很好的使不同關卡物件間的同一個物件達到屬性一致，並減少了產生的物件數量。

步驟如下：

- 1) 於遊戲主程式的初始函式時先將須流通的物件初始化。
- 2) 呼叫其餘的關卡物件的初始函式。
- 3) 將於遊戲主程式建好的物件，位置透過函式傳送給旗下每一個需要的關卡，使其能擁有同樣的記憶體位置，並對其保有更改權力。
- 4) 往後當關卡需要對其物件更改數值時，只需將更改過後的數值傳送給以指標儲存好的物件記憶體位置便可。

(二)物件採取回收再利用概念

由於 survivor 的原遊戲特性，我們會產生相當多的怪物物件以及武器物件，並且為了方便管理，我們使用了 vector 來掌控我們物件，然而我們發現，如果將 vector 內的怪物物件直接 erase，並且需要新增怪物時才又重新生成一個物件，push 進 vector 的話，將會大大拖慢遊戲速度。

故我們採取資源回收的理念，將機制改為使用兩個 vector 互相管控正在行動的怪物與被殺死的怪物，並設置 reset 函式，當怪物的數量低於預期時，我們會從被殺死的怪物中取出一個，重新 reset 後放回另一個 vector，此舉大大優化了我們的遊戲速度。

(三)物理引擎計算武器移動軌跡

因為希望可以一邊熟習物件導向的各項能力，並同時鍛鍊自我思考與想像能力，故本遊戲的所有角色攻擊，我們幾乎全部使用物理引擎計算其軌跡，從中學到了許多數學公式以及轉換，對過去所學溫故知新。

我們使用到了非常多，諸如：向量、圓周、拋物線軌跡、極座標與笛卡爾座標轉換、物理反彈，來呈現不同的物體運動方式，更增加遊戲難度。

其中，我們運用向量的形式來轉換出物體向中心運動的方式，使用圓周來呈現物體繞圈的動作等等，將過去所學運用到此實作當中。

四. 結語

1. 問題及解決方法

在此次實作當中，我們有遇到不少困難，但都慢慢逐一擊破，對我們來說是很棒的學習經驗。

第一個為處理背景問題，因為原遊戲的背景沒有邊界，故我們使用了物理遠景移動慢，近景移動快的原理，改善了使用者體驗，達到地圖無邊界的效果。

第二個為在不同的 class 間互傳值遇到困難，我們先嘗試在 CGameStateInit 中寫 getter，在 CGameStateRun 中宣告類別是 CGameStateInit 的物件來提取參數。但宣告物件時會出現「沒有預設建構函式存在」的錯誤。

後來經過助教的建議，使用增加 class 並紀錄 static member 的方法，但過程中仍有遇到許多問題，像是原本只在 class init 裡寫 static 的宣告函式。但在 class run 繼承時，會出現重複繼承 CGameState 的問題，但把 CGameStateRun 把繼承的 CGameState 刪除時，會使虛擬的 class 不

能間接繼承，將 static 多寫成一個 class 的話，則會出現已在 obj 中定義的錯誤，後來經過上網查詢之後，才知道是把定義寫在.h 檔中的問題，因此新增一個 static 的.cpp 檔，解決了這個問題。

經上述除錯後，static 的問題解決，但是因為初始時已經先預設如果 static 值是多少，要 load bitmap，故出現要先 load bitmap before show bitmap 的錯誤訊息，因此改成預設要先 load，解決了這項問題。

而因為原本判斷 static 值是多少，才會 Load bitmap 這個動作寫在 OnInit 中，出現了背景不會隨值更換的問題，便加了一個 function 來執行這些判斷，才終於解決問題。

2. 時間表

週次	歐佳昀(小時)	詹採晴(小時)	說明
1	介紹	介紹	介紹
2~4 (3/3 – 3/17)	22	22	初始畫面、實作基礎操作
5 (3/17 – 3/24)	11	10	增加互動物件、整理函式
6 (3/25 – 3/31)	11	14	玩家、敵人物件移動函式
7	DEMO	DEMO	DEMO 1
8 (4/08 – 4/15)	24	28	修 bug、物件移動
9~10 (4/15 – 4/28)	12	8	重構遊戲、撿寶石動畫
11 (4/29 – 5/05)	14	14	優化與新增物件、切分關卡
12~13 (5/05 – 5/19)	15	18	優化與新增物件、更新素材 DEMO 2
14~15 (5/20 – 6/02)	20	16	優化與新增物件、優化關卡
16 (6/2 – 6/9)	9	5	優化與新增關卡
17	DEMO	DEMO	DEMO
總時間	138	135	

3. 貢獻比例

110590045 歐佳昀 50%

110590032 詹採晴 50%

4. 自我檢核表

	項目	是否完成	說明為何不能完成
1	解決 MemoryLeak 的問題	V	
2	自訂遊戲 Icon	V	
3	About 畫面	V	
4	遊戲中說明如何操作	V	
5	發佈 (Release) 版本	V	
6	承上項，該版本可以正常運作	V	
7	報告字型、點數、對齊、行距、頁碼等格式正確。	V	

5. 收穫

- 110590045 歐佳昀

透過這次實作，我了解了物件導向以及相關的各個觀念，像是繼承、類別、屬性、friend、static 等等的用法，並且也學習到了如何使用 git 來輔助專案開發與版本管控，也了解以後在開發大型專案時，我也會適時地去注意程式碼的簡潔度，並適當的加上註解來解釋程式碼。

當然，這堂課並不只教會我們 c++、物件導向，這堂課更讓我有機會做中學，學中做，雖然仍然有很多美中不足的地方，但我相信重要的是過程，此次實作學習的經驗便是最好的收穫。

- 110590032 詹採晴

我也體會到了「做中學」。把上學期的所學，實際應用出來，加深了我對上學期的學習過程的理解：包括基本的程式設計思維，再到 class 間的繼承、static member、friend 的使用。在做中學中，我也遇到了困難，但這也訓練我如何解決問題的能力。做中學是加深我們對學習過程的理解，還有

培養我們實際操作的能力。這次的經驗也讓我學習如何與隊友合作、溝通的重要，相信這份經驗對我們的未來一定有幫助的。

6. 心得與感想

- 110590045 歐佳昀

這次是我第一次和人合作一起寫遊戲專案，不管是分配工作等，還是進入狀況後將段落分配等，了解且學習到應該如何與人合作、溝通外，也增進了如何了解對方打程式的習慣與想法、改善程式碼等等的技巧與知識，並且，我深刻體會框架以及架構的重要性，可以大大減少程式碼的行數，並且可以避免過於繁雜的想法，更要清晰遊戲的組成與結構，而如果沒做好的話，程式碼容易變的繁複難懂，最重要的一點是，物件導向的概念真的非常的重要且便捷，透過此次實作，我也深可了解到物件導向的概念究竟有多需要被精熟。

而經過這次實作，累積這次的經驗，相信下次面對與人合作的專案開發時，我能更清楚了解自己應該做甚麼，怎麼做，這是我覺得此次實作最重要的地方之一。

- 110590032 詹採晴

這是我第一次寫這麼龐大的遊戲專案。從一開始的作業練習到後來是慢慢越加熟悉遊戲的框架，以及了解到不同的函式的作用。我也體會到要把程式碼的架構寫的簡潔、清楚，以及適當的加註解的重要。因為到後來程式越來越多時，如果要修改到之前寫過的程式，如果之前沒有寫註解，或是沒有把程式架構整理清楚的話，對於整個架構會感到十分複雜，也要花上加倍的時間解決。

這學期做的遊戲在過程中雖然有面臨困難，但也體會到了成功 debug 把功能做出來的成就感。也自己找素材包，創造出屬於我們風格的遊戲。整體來說這學期修的這門課是開心的。

7. 對於本課程的建議、想說的話

沒有，謝謝老師。