CMoney 菁英軟體工程師戰鬥營期中專題成果報告書

## 遊戲名稱：衝衝衝衝拉風

團隊成員:董周強、李昱緯

目錄

[一. 專案簡介 2](#_bookmark0)

[二. 開發環境 3](#_bookmark1)

[三. 需求分析 4](#_bookmark2)

[四. 系統架構 5](#_bookmark3)

[遊戲流程圖 5](#_bookmark4)

[物件圖 6](#_bookmark5)

[五**.** 團隊組成與分工 7](#_bookmark6)

[甘特圖及分工表 7](#_bookmark7)

一**.** 專案簡介

1. 故事背景：

身為太空艦長的玩家，意外讓船艦開到宇宙中最混亂且無序的地區當中，你該如何在最快的時間內脫離這個困境，但可千萬不要小看這個區域，你的船艦變得完全不受控制，它似乎有了…..生命?

## 遊戲介紹：

《衝衝衝衝拉風》為一款類益智遊戲、玩家警慎的思考每一步該如何前進，

帶領自己的船艦離開危險的區域。

* 1. 風險與報酬並存的操作方式

通過按壓能量條，可以使飛船有不同的前進距離，但按壓越久飛船的前進路線就越不穩定，在飛船邊離軌道時，只能透過滑鼠左鍵或右鍵讓飛船導回正軌，考驗玩家的反應力與決策力。

* 1. 緊張刺激的遊戲風格

這款遊戲希望將玩家的容錯率降到最低，玩家的每一步都至關重要，要求玩家在進行每一步時都專注在當下，若是失誤了飛船可會不知偏離去哪裡。

* 1. 豐富的遊戲元素

在遊戲中有多樣化的地形會影響者玩家的前進，也有各式各樣的道具可以讓玩家度過各式各樣的挑戰

二**.** 開發環境

## ※硬體環境

|  |  |  |
| --- | --- | --- |
| 規格 | ROG Zephyrus G14 | ASUS S410UQ |
| 處理器 | AMD Ryzen 7 4800HS | Intel Core i5-8250U |
| 作業系統 | Windows10 home | Windows10 |
| 記憶體 | 8GB DDR4 on board + 8GB DDR4-3200 SO-DIMM | 8GB\*1 DDR4 2133 |
| 資料儲存 | 512GB SSD | 1TB 5400轉 HDD |
| 顯示晶片 | NVIDIA® GeForce® GTX 1650Ti | NVIDIA® GeForce® GTX 940M |

※開發軟體

IntelliJ IDEA Community Edition 2020.2.2

※使用語言

Java

三**.** 需求分析

## **視覺畫面與使用者介面的直觀性**

因此遊戲主要需要玩家迅速的針對飛船的偏移作出相對應的操作，所以需要很即時的且準確地將飛船的偏移狀況回饋給玩家。飛船能量條的設計也非常重要，因為直接影響到了玩家的操作策略，所以要讓玩家直觀感受到能量條與移動距離的相關性。

## **合理的遊戲平衡**

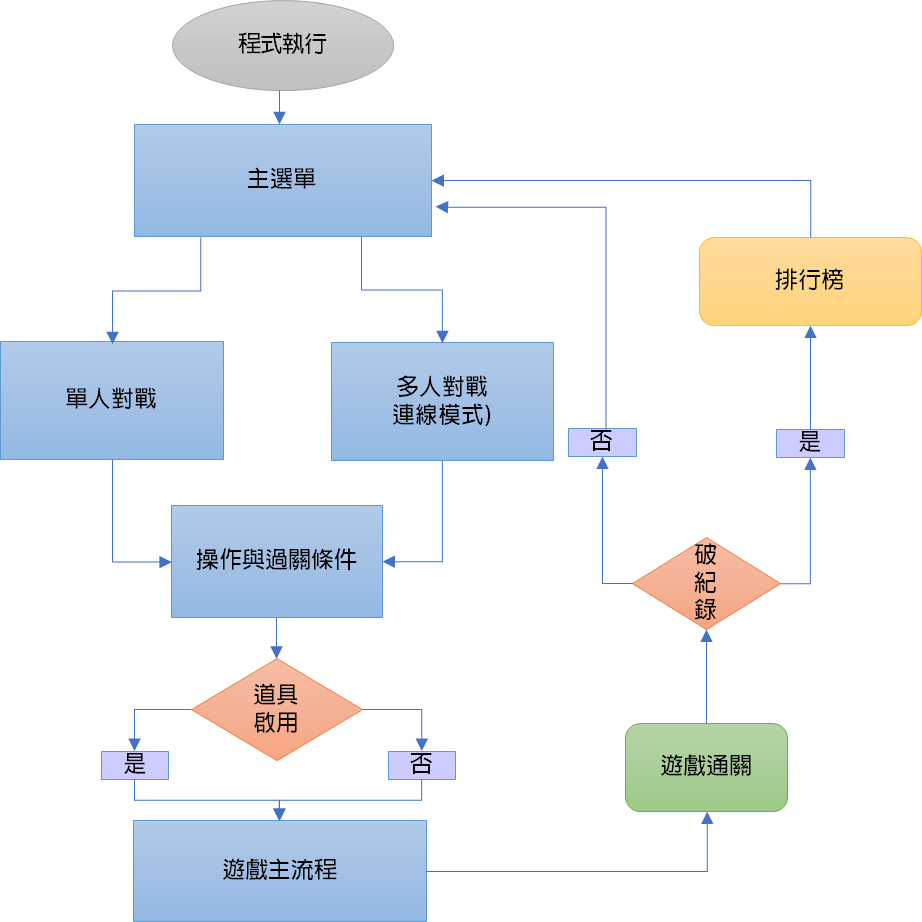
遊戲中，因為目標是以最少的次數去達到終點，我們必須合理的設計角色、隨機性、道具與地形的各式變數，若設計不平衡，將會使遊戲難以通關，抑或過於簡單通關， 喪失遊戲趣味性。

## **場景與腳色間的互動**

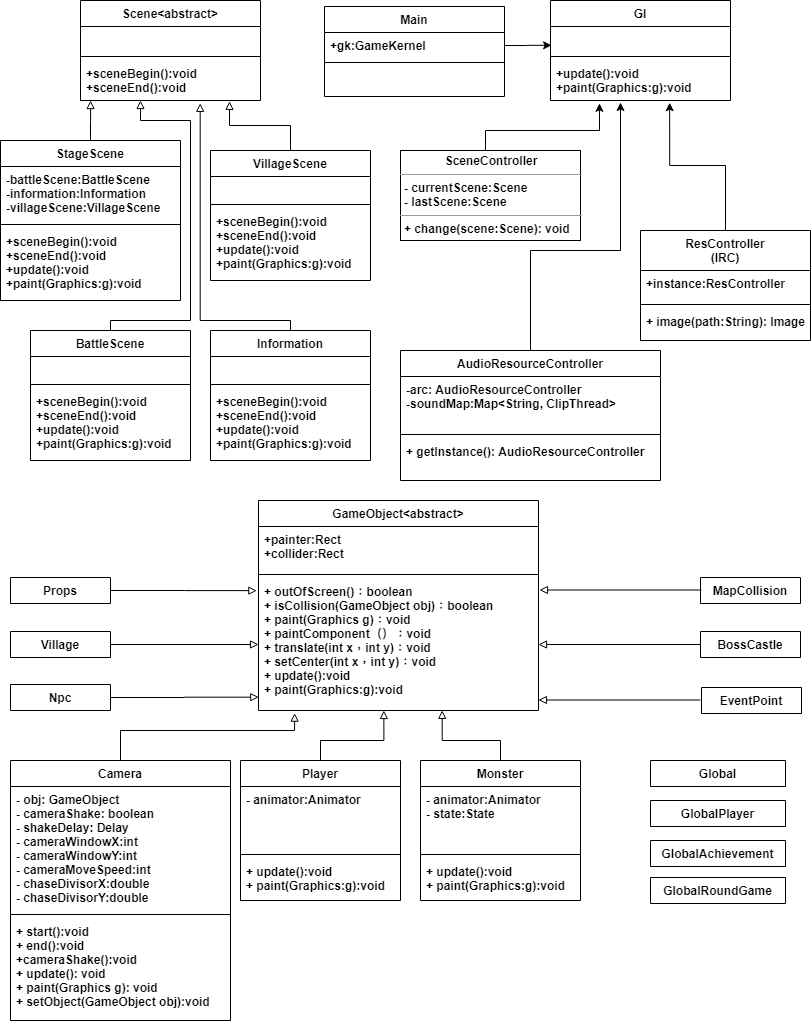
因本遊戲的核心在於做角色移動的RNG性，故在基礎操作已經有難度， 在設計場景中最困難之處在於說如何保有遊戲性之餘同時維持操作的可控性與合理性。

四**.** 系統架構

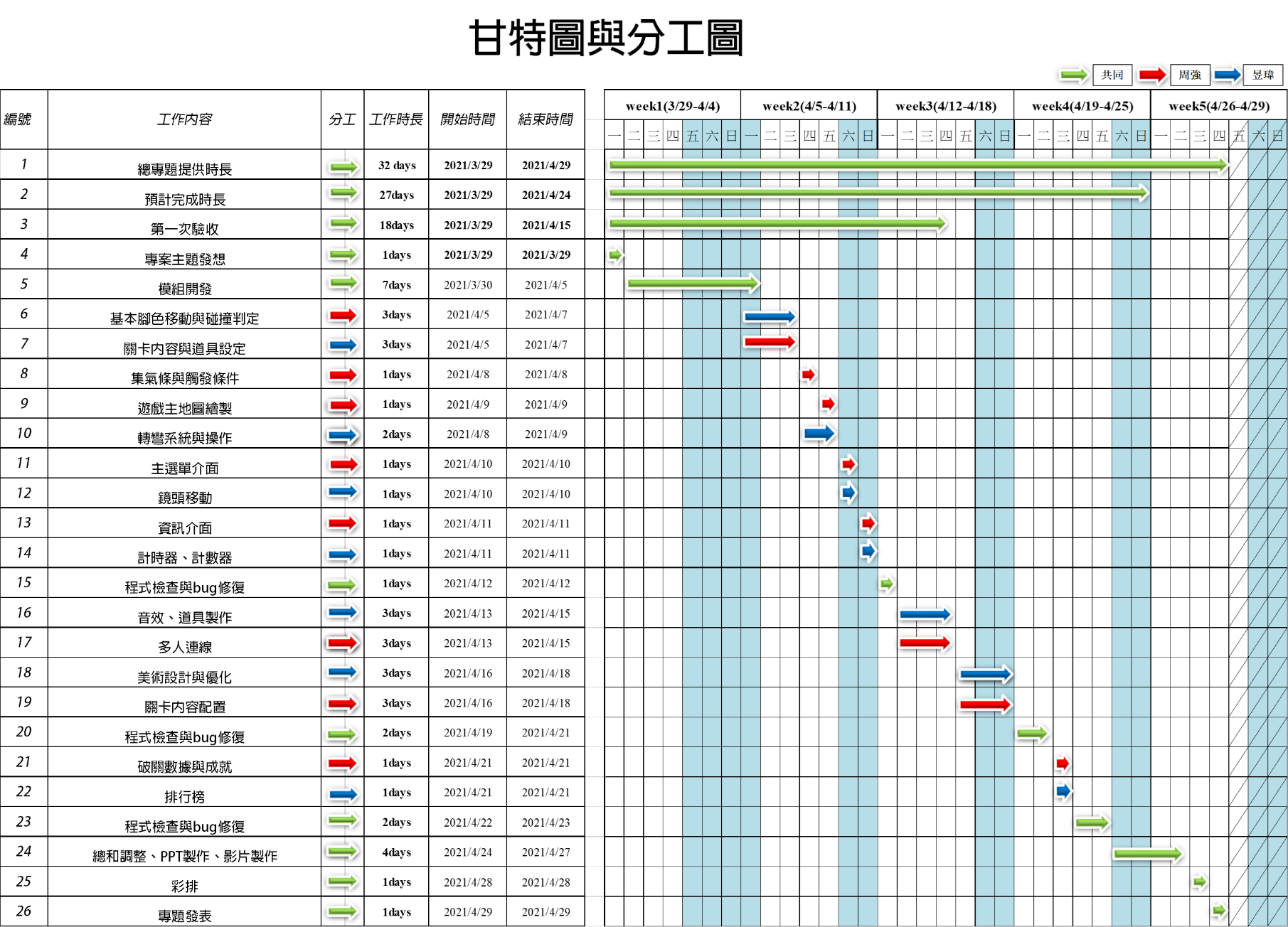
# 遊戲流程圖



# 物件圖



五**.** 團隊組成與分工



六**.** 實作方法

1. 造型商店及衣櫃系統

為了方便擴充，我們把造型的數量和資訊統一以一個 class enum 紀錄，並在造型商店及衣櫃的場景中分別增設一個內部類別，內含 2 個變數

（正在focus 的造型和其造型在場景中的展示序號），以記錄玩家目前的操作狀態，方便因應玩家的操作以執行不同的function。

1. 成就系統

為了方便管理，我們為成就系統開設了一個static class，內含一個class enum 以紀錄每個成就的內容和玩家的達成狀態，我們為每個成就編寫獨立function 以判斷成就是否被達成，開發者只需要把 function 安插在觸發位置，成就系統就能獨立運作。

1. 關卡中的場景切換

遊戲進行時，有三個場景需要一直進行切換，分別是 JungleScene、BattleScene 和 VillageScene。為了達到遊戲畫面順暢，我們讓JungleScene 主場景的物件會一直存在，避免重複載入而讓遊戲畫面停

頓，而BattleScene 和 VillageScene 將會在需要的時候才新增物件，其他時間將會處於 null 的狀態。而邏輯更新的部分，為減輕程式運作的負擔， 我們只會更新玩家所在的場景邏輯，其他兩個場景的邏輯將會停止更新， 直到玩家再次進入該場景。

1. 多人連線 – 邏輯更新

為了實現多人連線中的遊戲邏輯同步，我們給了每位玩家一個優先順位。當一號玩家在 JungleScene 時，將會由一號玩家負責更新遊戲邏輯， 並發送給其他玩家； 當一號玩家進入BattleScene 或 VillageScene 時， JungleScene 的邏輯更新將會按照優先順位，由二號玩家負責，以此類推。

1. 主地圖繪製與碰撞判斷

我們的主地圖上有多種特殊地形，且範圍廣大，若逐一完成，在繪製/產生地圖時會相當耗時，碰撞亦會難以處理。故我們反向操作， 單純繪製一張主地圖後，利用地圖產生器將不能行走的區域點出，在程式中將背景圖繪製上後，載入我們點出的物件疊上背景底圖以判斷碰撞，如此便能達成簡單達成判斷所有地形碰撞且節省大量時間。

1. 平衡調整

在調整平衡時，我們會預先設定我們想要看到的狀態，例如：角色在 20 等時穿戴 A 或 A 以上裝備才能打贏Boss1。並在事前計算傷害公式後決定各個裝備的性價比，以上述條件逐一調整角色在這個狀態前的成長、金錢掉落、及 Monster 的各能力數值設定。並在數次測

試後確保遊戲節奏符合預期後進行調整，隨著測試次數愈多、平衡微

調後，遊戲平衡也會愈趨穩定。

七**.** 專案成果

遊戲首頁

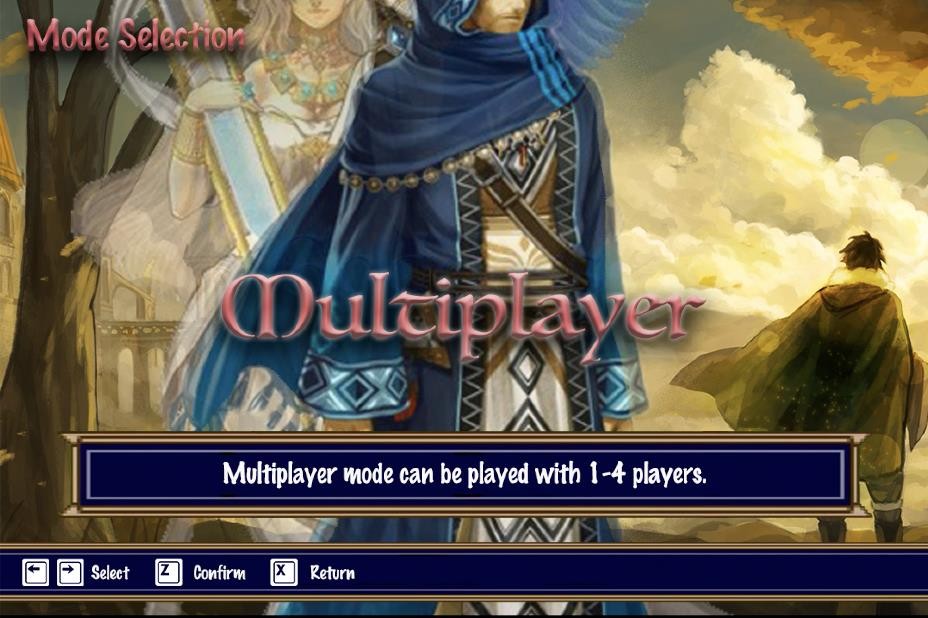
選單介面



選單介面 2



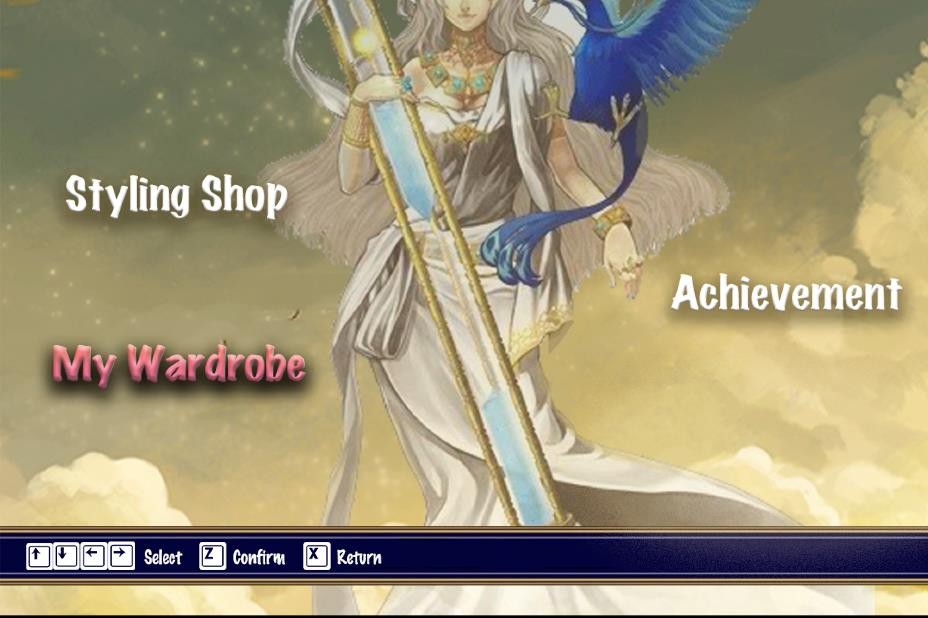
選單介面 3



單人模式 - 關卡選單介面



女神的房間 - 選單介面



造型商店



我的衣櫃



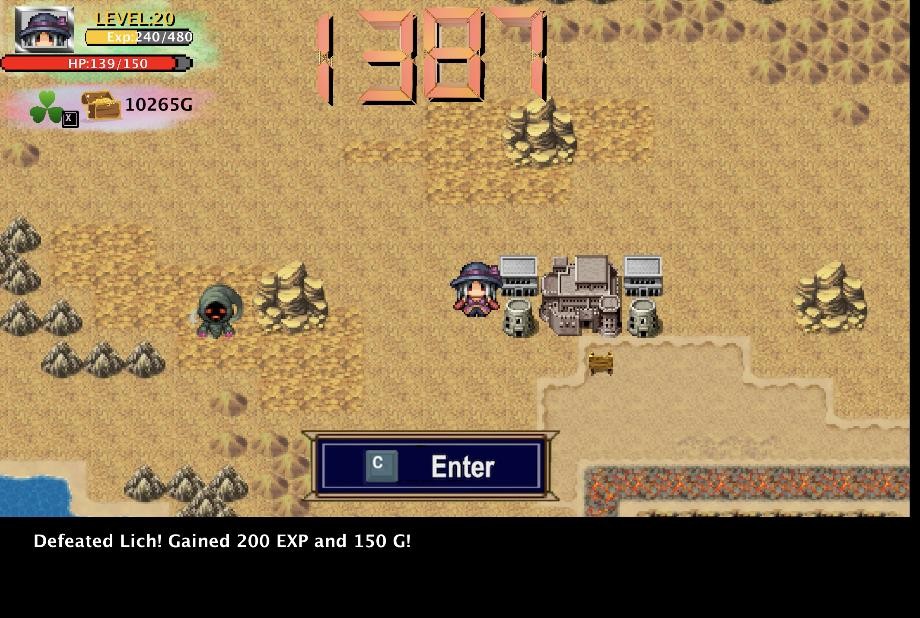
成就系統



多人模式 – 等待室



遊戲畫面





遊戲結算畫面





八**.** 討論與結論

透過編寫期中專題遊戲，讓我們的編寫程式語言基礎變得更紮實，提升了不少實作經驗，同時讓我們了解到我們在編程知識上的不足。

在本次的期中專題，很幸運的我們除了完成核心內容（單人模式）外，還超過預期，完成了很多額外內容（多人模式、成就系統），這些都需要感謝老師和同學們的協助，有了他們的建議和協助多人連線測試，才能讓我們在開發和除錯上更加順利。

而針對我們的遊戲，我們認為還有不少可以改進的地方：

1. 客製化操作鍵：在測試的過程中，我們發現不同的玩家對於操作鍵都會有不同的操作習慣，因此我們應該增設一個可以讓玩家按照自己的操作習慣設定操作鍵的功能。
2. 使用流程上的細節：我們目前的遊戲，有部分操作未能讓使用者直覺操作以及出現了多餘的操作步驟，這都是我們可以再進行改良的部分。
3. 未能在短時間內提升玩家的興趣：我們的遊戲屬於慢熱型，玩家需要遊玩兩次以上才能抓住遊戲的精髓，但基於現在的玩家有很多的遊戲選 擇，因此我們應該把遊戲改善成玩家能在更短的時間內抓住精髓，提高玩家的興趣。