



國立高雄大學資訊管理學系

2021 OOPII GROUP13_PROJECT

Checkpoint3 說明文件

資訊管理學系 113 級

A1093355 彭郁庭

A1093366 陳妍均

A1093348 張聿豐

2021/12/21



目錄

壹、摘要.....	1
貳、研究動機與研究問題.....	2
參、文獻回顧與探討.....	3
肆、研究方法及步驟.....	4~13
一、考量製作遊戲的方法.....	4
(一)OOA.....	4
(二)OOD.....	4~5
(三)OOP.....	5~6
二、遊戲架構設計.....	7~11
三、遊戲規則說明.....	12
四、遊戲操作方法.....	13
伍、預期結果.....	14
一、系統功能.....	14
二、系統特色.....	14
三、使用環境.....	14
四、預期研究效益.....	14
(一)加強民眾環保意識.....	14
(二)加深民眾對於各物種常識.....	14
陸、專案進度.....	15
柒、參考文獻.....	16



表目錄

表 4-1 使用者可派出的角色.....	6
表 6-1 專案進度排程表.....	8
.....	



圖目錄

圖 2-1 收入排名前 100 mobile 遊戲類型比.....	2
圖 2-2 收入排名前 100 mobile 遊戲流水收入.....	2
圖 4-1 遊戲程式架構圖.....	5
圖 4-2 UML 圖.....	8
圖 4-3 UML 圖.....	9
圖 4-4 UML 圖.....	9
圖 4-5 UML 圖.....	10
圖 4-6 UML 圖.....	10



壹、摘要

此份文件會講述開發此專案的研究動機，從使用者可能於遊玩時遇到的問題，及現行遊戲可加強的地方，做為我們進一步實作出遊戲的參考方向。會先由物件導向程式與現今多執行緒技術對於欲開發遊戲的重要性開始講述。以圖表的方式呈現我們的程式架構，採用 JAVA 作為主要開發工具的研究方法與步驟，最後是專案進度及程式執行成果。

透過本次的專案開發出一款保留原作戰鬥形式的同時，新增可召喚動物的多樣性及圖鑑功能，打造更貼近現實生態的策略遊戲。希望能讓使用者在遊玩時增長科普知識，進而將愛護環境的觀念深入使用者心中。

貳、研究動機與研究問題

我們想要設計一款以線上手遊「貓咪大戰爭」的基本遊戲架構，但更加貼近自然法則的遊戲，在原作中玩家只能召喚出不同型態的貓咪進行戰鬥。而我們參考了原有的戰鬥方式並新增其他可召喚物種、戰鬥時的年齡限制和收集物種圖鑑的功能來強調環保的重要性及寓教於樂的意圖，讓玩家在遊玩遊戲的同時也能獲得新知，一同為地球媽媽盡一份心意。

本遊戲是屬於塔防 MOBA 類的卡牌遊戲，在現今 MOBA 類遊戲風靡的時刻，由下面這兩張圓餅圖可知，卡牌及 MOBA 類型的遊戲，皆是遊戲類型裡佔收入的一大比例，因此可以得知卡牌類型加上 MOBA 類型，在市場裡面是受到歡迎的，並且消費者是願意投錢支持的。

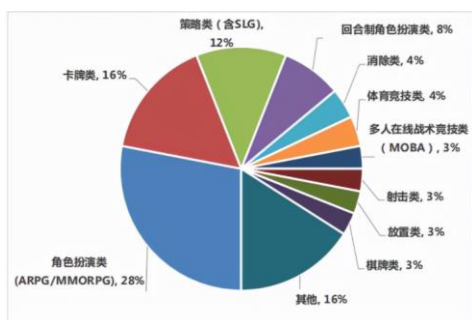


圖 2-1: 收入排名前 100 mobile 遊戲類型

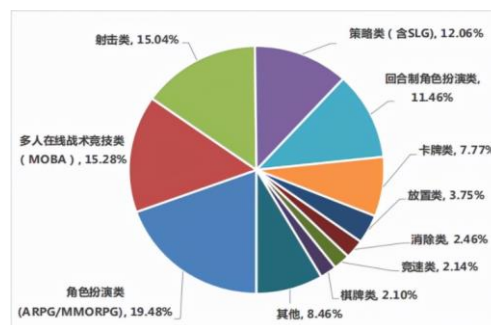


圖 2-2: 收入排名前 100 mobile 遊戲流水收入占比



參、文獻回顧與探討

現今 Java 的 GUI 工具包包含最原始的 AWT(Abstract Windows Tooljit)、試圖解決前者缺點的 Java Swing 和 SWT(Standard Widget Toolkit)。AWT 的優點是所佔 CPU 空間小、穩定性高，缺點則是擴展性、豐富性低。因為它是 JFC 的一部份，AWT 上的圖形函數與操作系統所提供的圖形函數之間存在著對等關係，這會導致同一程式在不同平台上的表現不一致，進而違背了 Java 主打「一次編寫，到處運行 (write once, run everywhere)」的語言特性。後來推出的 Swing 不僅支援了 AWT 提供的所有功能，並且是以純 Java 寫成，能跨平台執行加上成熟穩定又豐富靈活的特性，使其成為三者中最強大的工具包。然而也因為它實現了所有組件，需要消耗的空間自然更多，常因無法及時回收冗餘的組件而導致程式性能下降。最後的 SWT 可說是兩者的融合體，具有相對 AWT 更豐富的組件，消耗的內存較 Swing 少，卻也不夠穩定亦不可和 Swing 交互使用。我們將依照上述考量設計遊戲介面，優先考慮平台的移植性和組件的豐富性選擇 Swing 作為主要使用的工具包。

執行緒的部分由於 Java 支援多執行緒，能透過重寫 run()和實現 Runnable 介面來新增執行緒，通過呼叫 start()啟動。然而執行緒的執行順序是由作業系統排定，所以我們須透過不同的關鍵字來控制執行緒的狀態，確保資料不會出現互相競爭的情形。為了達到同步運算和介面更新的需求，我們會嘗試撰寫出執行緒安全的程式碼，提高 CPU 的效能，減少程式執行時間；音檔的部分也運用到多執行緒，藉由實作 Runnable，改寫 run()函數，藉由讀取音檔名稱，因為音訊是以 byte 來傳遞的，所以必須經過解碼存成 AudioInputStream 的 type，以 SourceDataLine 的形式讀取音樂，透過 start()函數來執行音樂。此次因為系統差別，在 X2GO 上播放的音檔會有斷斷續續的情形。



肆、研究方法及步驟

一、製作遊戲的方法：

OO 的考量

(一)OOA

本次使用 JAVA 語言作為該系統的主要開發工具，能以單機遊戲形式遊玩。

(二)OOD

- **InfoFileReader**
負責讀取地圖和角色檔案
- **Role**
負責初始化物種資料與戰鬥會使用到的動作
- **PlayerRole/EnemyRole**
負責儲存雙方戰士的資料和會各自會使用到的動作
- **Player**
負責更新與控制我方場上的已召喚物種
- **Enemy**
負責更新與控制敵方場上已召喚物種
- **Wallet**
負責控管戰鬥時的花費與收入
- **Game**
負責執行和控管遊戲進行
- **Run**
負責執行遊戲介面
- **MusicPlay**
負責背景音樂的播放
- **GameFrame**
負責戰鬥開始的介面
- **MapFrame**
負責地圖的介面
- **ArrangeFrame**
負責選取動物的介面



- **FightFrame**

負責打鬥中的介面

(三)OOP

- **Game**

為主程式，負責接收事件並創建 Player、Enemy、Wallet、InfoFileReader 物件利用各類別中定義的方法控管程式流程。

- **Role**

為 PlayerRole 和 EnemyRole 的父類別。建立傳入一個代表物種名稱的字串，和分別代表物種編號、血量、移動速度、攻擊力、懸賞金和壽命整數型態參數的建構子來初始化物件。定義移動方法的 move()、負責計算攻擊力的 attack()和遭受攻擊的 beAttacked()。

- **PlayerRole/EnemyRole**

此 class 繼承了父類別 Role。

- **Player/Enemy**

宣告二個陣列列表 armymember、movingArmy 分別用來存取戰士隊伍、派出的戰士，和整數型態變數 warfrontLocation 用來存取場上隊伍最前線的位置、enemyWarfrontLocation 用來存取敵方最前線的位置，定義呼叫新戰士進入場上隊伍的方法 call()、移動全部場上戰士的方法 move()、計算最前線戰士發動一次攻擊的方法 fight()、遭受敵方攻擊的方法 beAttacked()、清除場上已死亡戰士的方法 cleanArmy()、判斷我方最前線的位置的方法 judgeWarfrontlocation()和城堡物件被攻擊的方法 castleBeattacked()。

- **Wallet**

宣告整數型態變數 money、speed、level 分別用來存取錢包總金額、錢包產生前的速度、錢包的等級，定義召喚物種上場消耗金幣動作的方法 spend()，其中數值由 Animal_info 表格中的 Animal_price 來決定、依照錢包速度累積金幣的方法 makeMoney()、使錢包升級的方法 accelerate()以及當敵方角色死亡後可得到懸賞金的方法 addMoney()。

- **Run**

建立 GameFrame 的物件，並藉由 setVisible(true)，將介面呈現出來。



- **MusicPlay**

實作 Runnable，並創建建構子 MusicPlay() 裡面放入兩個參數，String 型態以及 boolean 型態，分別是 BGM 名稱以及判斷是否播放的布林值；startMusic() 是透過建立 Thread 的物件去執行 start()；setStop() 是傳入 boolean 型態，改變 stop 的布林值；run() 是執行 BGM 的函數，藉由判斷 excute 以及 stop 來判斷是否執行 BGM；doNothing() 藉由傳入毫秒，讓執行緒睡 1 秒鐘；music() 是傳入 BGM 檔案名稱，並藉由將 byte 解碼，讀取，藉由 start() 進行播放，最終呼叫 drain() 和 close() 關閉 SourceDataLine。

- **MapFrame**

繼承 JFrame 並實作 ActionListener。宣告 Button[] 陣列存放 9 張地圖的 buttons，以及創建 GameFrame 物件，藉由建構子設置 9 張地圖介面的擺設，並在 inner class ActionListener 裡給 button handler，經由點擊按鈕產生對應的地圖。

- **GameFrame**

建立 MapFrame 物件和 MusicPlay 物件，在建構子中設定起始遊戲介面的詳細內容和按鈕。

- **FightFrame**

建立 Game 物件用來控制遊戲進行過程，boolean 物件 exit、ready 用來設定音樂是否需播放，多個 JPanel 物件用來安排視窗版面，JButton 物件 dontCallRoleButton、attackButton、locationButton、accelerateWalletButton 分別用來使玩家每回合可不作動作、釋放攻擊技能、釋放退後技能，JButton 陣列用來使玩家派出動物角色，JLabel 物件 topLabel 用來顯示錢包、城堡資訊，整數變數 mapID、count 分別用來存取地圖編號和計算回合數，MusicPlay 物件用來控制音樂播放。
在建構子中設定戰鬥視窗的詳細內容和按鈕，並控制遊戲的進行。

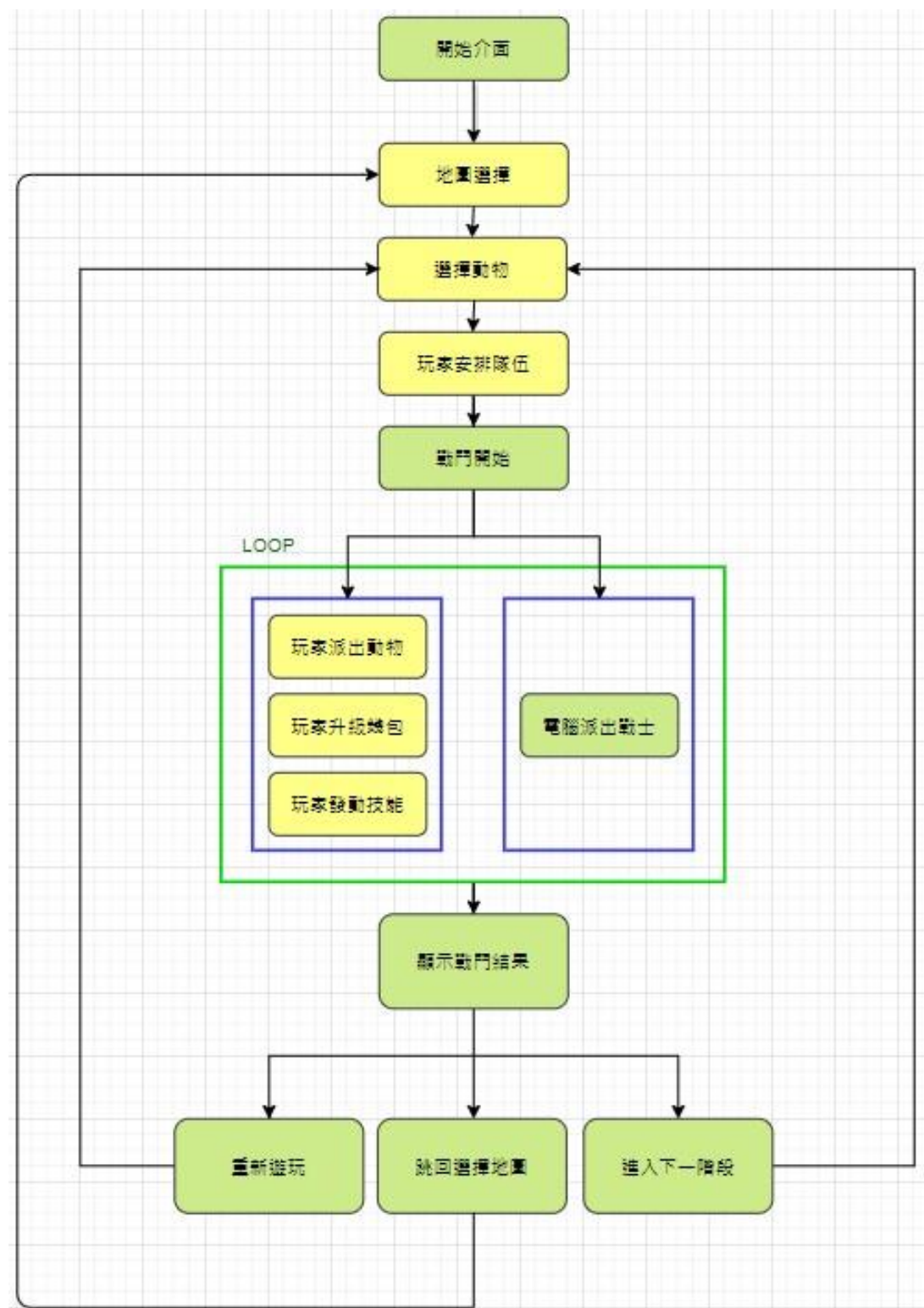
- **ArrangeFrame**

建立整數變數 mapID 用來存取地圖編號，InfoFileReader 物件 readAllPlayerMember 用來讀取檔案取得所需的資料，ArrayList playerTeam 用來存取玩家的隊伍，多個 JPanel 物件用來安排視窗版面，JButton 物件 putInButton、moreInfoButton、playButton 分別用來將角色放入隊伍中、查看更多資訊、進行遊戲，JButton[] 物件 playerTeamButton、memberLabel 分別用來存取所有動物和玩家隊伍的動物按鈕。
在建構子中設定安排隊伍視窗的詳細內容和按鈕。

二、遊戲架構設計

圖 4-1

遊戲程式架構圖



Uml 圖:

圖 4-2

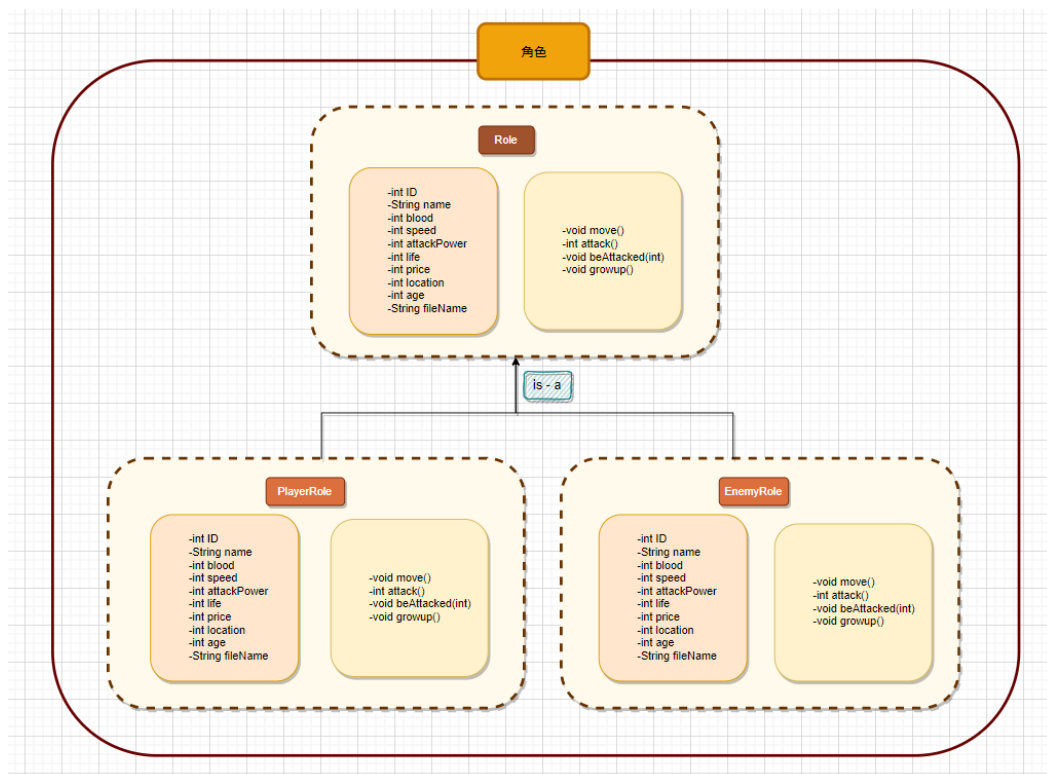


圖 4-3

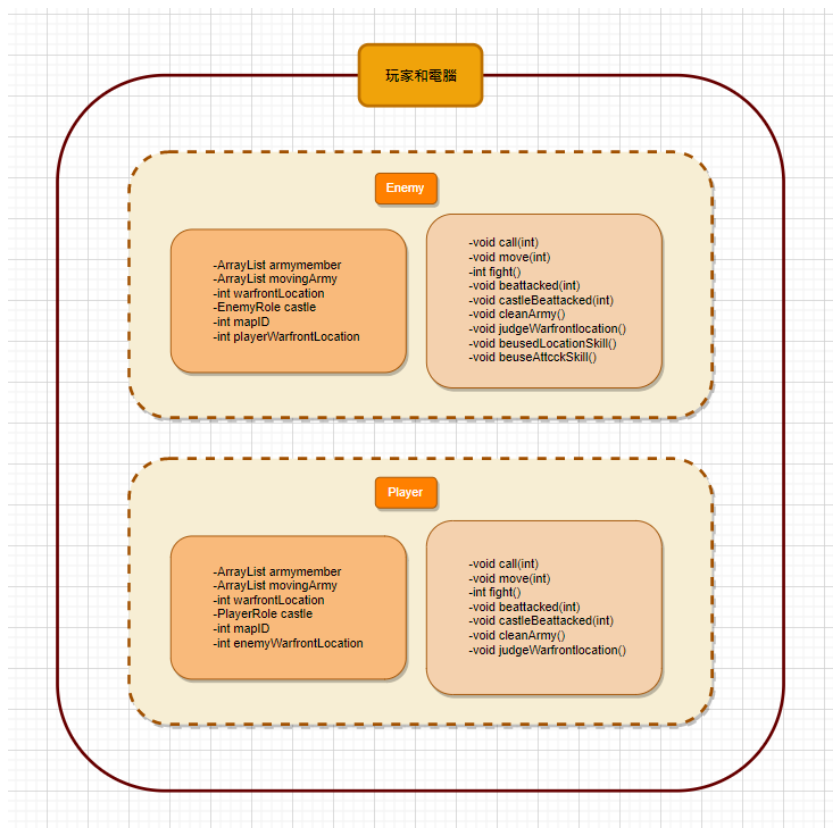


圖 4-4

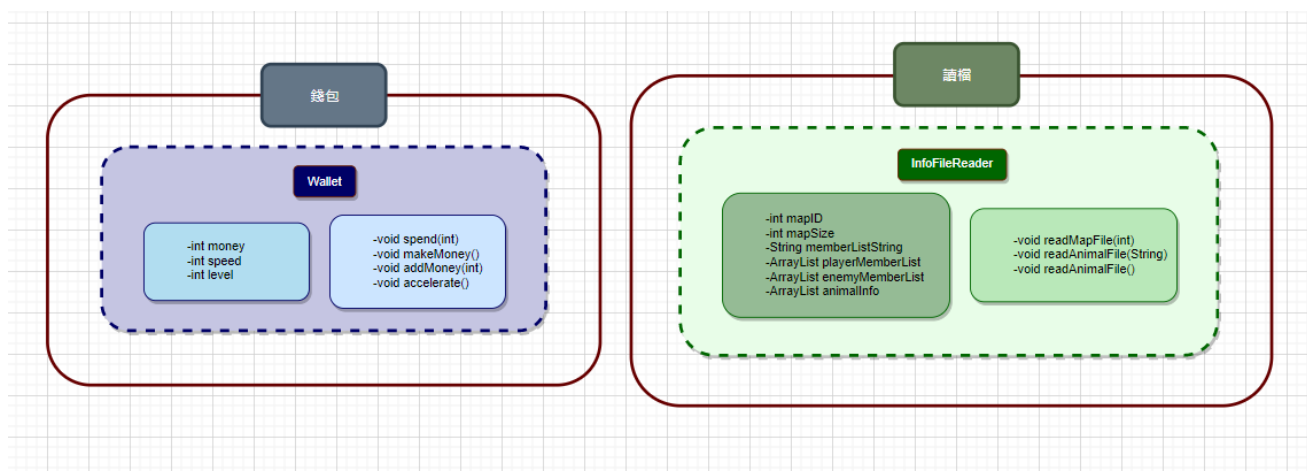


圖 4-5

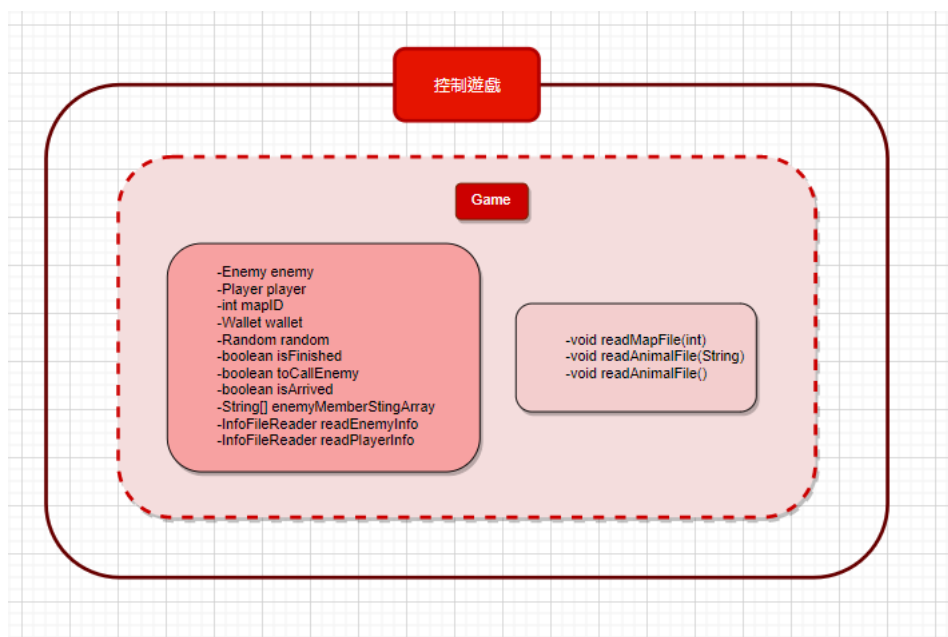
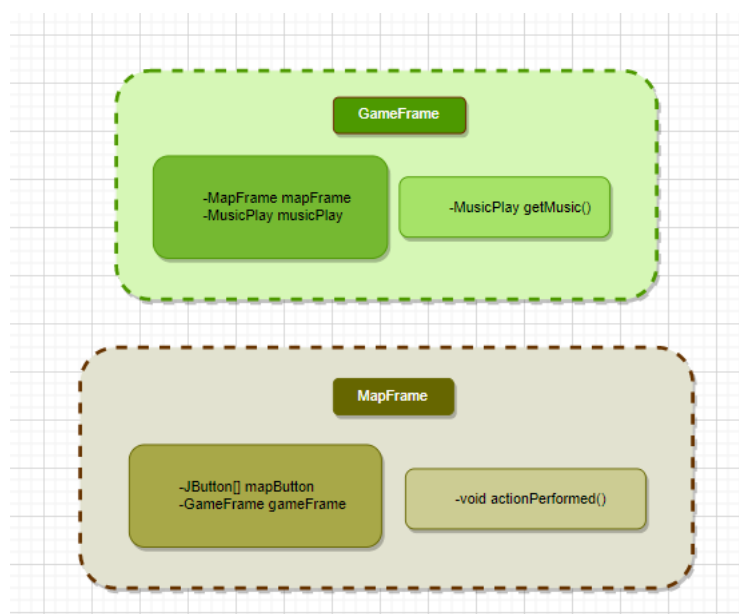
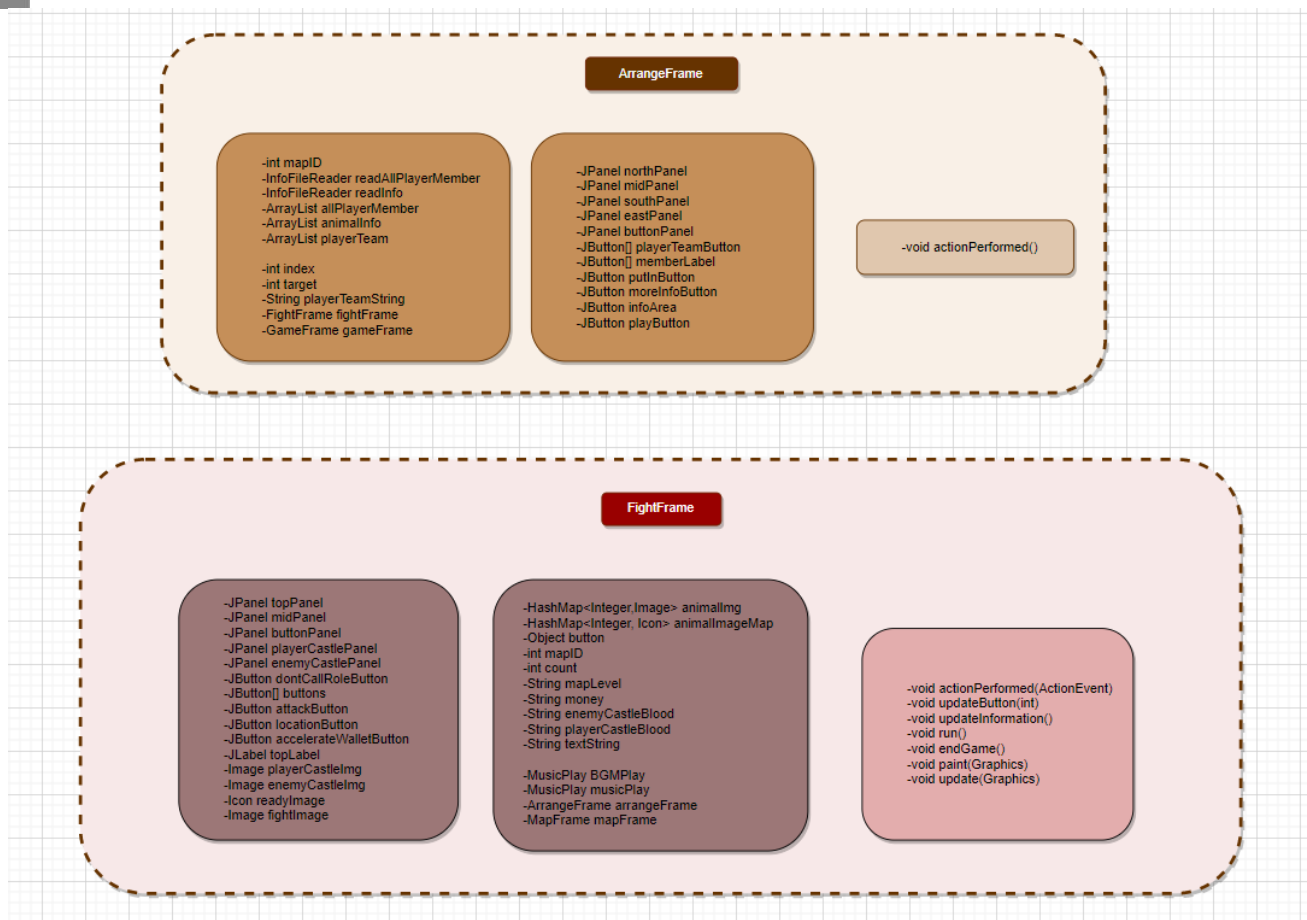


圖 4-6







三、遊戲規則說明

玩家面：

1. 玩家可以從 12 隻角色中挑選 1-5 隻動物，點擊 Start 按鍵，開始遊戲。
2. 每一回合只能點選其一種技能、Don't want to call、或派出一隻小兵。
3. 遊戲中會有錢包，每回合增加 50 元，可藉由購買 Accelerate Wallet 來生及錢包速度提升產出金錢的額度，最高只能升級到 6 等。
4. 若錢包金額不足以購買戰士或釋放角色，只能選擇 Don't want to call，結束這回合。
5. 可以由錢包購買小兵上陣與敵方對戰，以攻擊對方的主堡。
6. 擊殺敵方小兵可賺取懸賞金。
7. 可購買技能- Attack Skill，來攻擊場上所以敵方 300 滴血。
8. 可購買技能-Location Skill，使敵方敵人後退一小段距離。
9. 當敵方的主堡血量歸 0，玩家獲得勝利，若己方主堡歸 0，玩家失敗。

遊戲面：

1. 藉由玩家使用小兵，攻擊電腦方小兵，以攻擊電腦方主堡，使電腦方主堡血量歸 0。
2. 主堡血量歸 0 方失敗，另一方獲勝。
3. 電腦方兩回合會任意從每所分配的小兵中，派出 1 隻小兵。
4. 電腦方不會使用 Attack Skill、Location Skill、Accelerate Wallet 三項技能。



四、遊戲操作方法

首先點擊規則說明了解規則，了解完規則點擊 start 的 button 進入下一個步驟，會跳出另一個視窗介面，供玩家選擇 9 張地圖，每一張地圖皆有不同的背景、場景以及敵人，點選後，進入下一階段，選擇上場動物，每一場都可以選擇 1-5 隻動物，超過 5 隻或少於 0 隻會跳出錯誤訊息，每隻角色的個人資訊皆可透過點擊角色來了解，也可藉由點擊以選擇的動物進行刪除，選擇完本場上陣動物後，點擊 start 進入戰鬥，戰鬥中可以藉由玩家錢包的總金額判斷及選擇三種技能以及點選出場動物，藉由與敵方動物對戰和攻擊敵方主堡，當敵方主堡血量為 0，玩家獲取勝利，反之，玩家主堡血量為 0，敵方獲取勝利。

表 4-1

使用者可派出的角色名單

編號	動物名稱	速度	血量	攻擊力	價格
1	無尾熊	1	300	50	50
2	狐狸	1	300	100	100
3	熊貓	1	450	100	150
4	斑馬	1	450	150	200
5	非洲象	1	900	50	250
6	老鷹	1	450	200	250
7	伊蘭羚羊	1	600	150	250
8	長頸鹿	1	750	100	250
9	白鯊	1	300	250	250
10	花豹	1	450	250	300
11	抹香鯨	1	900	100	300
12	犀牛	1	750	200	350



伍、預期結果

一、系統功能

在系統功能方面主要有下列兩項功能：

1. 選擇關卡：
可選擇地圖關卡，總共有 9 關。
2. 編輯隊伍功能：
可更動隊伍編成。
3. 戰鬥功能：
可透過點擊召喚動物上場戰鬥和釋放技能直到分出勝負。

二、系統特色

1. 簡易性
透過友善的使用者介面和淺顯易懂的玩法能幫助使用者輕易上手。
2. 教育性
透過配合動物習性及壽命的設定來打造更貼近真實生態的情境，搭配日後將推出的圖鑑功能，幫助使用者得知每種動物的詳細資訊，能達到寓教於樂的效果。
3. 多樣性
伴隨日後的捕捉功能及更多關卡推出，能吸引玩家的蒐集慾望及遊玩動力。

三、使用環境

本次使用 JAVA 語言作為該系統的主要開發工具，操作簡單且可套用至不同裝置執行，主要以開發電腦版為目標。

四、預期研究效益

(一)加強民眾環保意識

我們會藉由遊戲畫面盡量真實的呈現現實的生態情況，並透過劇情描述環境正面臨的問題，進而讓玩家於遊玩過程中意識到為我們共同居住的星球努力，保留我們現今能觀察到的所有壯麗的自然景觀是值得且重要的一門課題。

(二)加深民眾對於各物種常識

透過遊戲介面的設計，玩家在幫捕捉到的動物進行升級的動作時，必須先進入到物種圖鑑的介面。這時就能大大增加玩家觀看物種簡介的機率，進而加深民眾對於各物種的基本常識。



陸、專案進度

表 6-1

專案進度排程表

編號	工作名稱	開始	完成	期間	九月		十月				十一月				十二月			
					第三週	第四週	第一週	第二週	第三週	第四週	第一週	第二週	第三週	第四週	第一週	第二週	第三週	第四週
1	資料收集	2021/9/23	2021/9/25	兩天	■													
2	專案訂定	2021/9/25	2021/9/27	兩天		■												
3	蒐集文獻	2021/9/27	2021/10/3	1W		■	■	■										
4	程式撰寫	2021/10/12	2021/12/10	8W			■	■	■	■	■	■	■	■	■	■		
5	功能測試	2021/12/12	2021/12/14	兩天												■		
6	系統測試	2021/12/14	2021/12/16	兩天													■	
7	文書撰寫	2021/12/10	2021/12/14	四天												■	■	
8	書面報告	2021/12/14	2021/12/16	兩天													■	
9	製作簡報	2021/12/16	2021/12/17	兩天													■	■
10	專題發表	2021/12/17	2021/12/27	1W													■	■



柒、參考文獻

✚ 維基百科—Swing(Java)

[https://zh.wikipedia.org/wiki/Swing_\(Java\)](https://zh.wikipedia.org/wiki/Swing_(Java))

✚ Java 多執行緒程式設計基礎知識彙總

<https://iter01.com/515915.html>

✚ 遊戲收入的圓餅圖

<https://min.news/game/bcbccda63aabf819a20eef6527a1146c.html>