**明新科技大學資訊工程系**

視窗程式設計

期 末 報 告

猜拳與擲骰子遊戲

學 生：

何嘉蕾 N12170031

陳達志 N12170036

中華民國114年6月3日

目 錄

[第一章 緒論 4](#_Toc199786297)

[1.1 動機 4](#_Toc199786298)

[1.2 問題陳述 4](#_Toc199786299)

[第二章 系統討論 5](#_Toc199786300)

[2.1 系統流程圖 5](#_Toc199786301)

[2.2 系統類別圖 6](#_Toc199786302)

[2.3 功能模組說明 7](#_Toc199786303)

[第三章 系統實現 8](#_Toc199786304)

[3.1 環境開發 8](#_Toc199786305)

[3.2 核心功能代碼解析 8](#_Toc199786306)

[3.3 關鍵技術實現說明 8](#_Toc199786307)

[3.3.1 遊戲動畫 8](#_Toc199786308)

[3.4 界面設計與使用者操作流程 9](#_Toc199786309)

[第四章 測試與驗證 11](#_Toc199786310)

[4.1 舊紀錄確認--OK 11](#_Toc199786311)

[4.2 載入紀錄確認--OK 11](#_Toc199786312)

[4.3 遊戲流程與結果判定--OK 12](#_Toc199786313)

[4.4 回傳結果--OK 13](#_Toc199786314)

[4.5 紀錄更新--OK 13](#_Toc199786315)

[第五章 結論與未來工作 14](#_Toc199786316)

[6.1 研究成果總結 14](#_Toc199786317)

[6.2 未來改進方向 14](#_Toc199786318)

[參考資料 15](#_Toc199786319)

# 第一章 緒論

## 1.1 動機

根據老師提供的題目參考裡面，選擇”猜拳與擲骰子遊戲”的原因一方面是因為這是兩個遊戲組成適合兩人一組的搭配，另一方面是這個題目在構想時直覺基本需要使用三個以上表單呈現，也可以藉完成此報告過程確認這學期對表單功能是否能正確運用。

## 1.2 問題陳述

“猜拳與擲骰子遊戲”為兩個遊戲的組合，玩家可以選擇想玩的遊戲，並且可以累積積分。”猜拳遊戲”如真實猜拳一樣，玩家選擇出拳的種類，對手則用隨機出拳。”擲骰子遊戲”因為骰子點數都無法決定，所以玩家與對手都將用隨機方式決定點數。

設計為有兩個遊戲的遊戲介面，所以兩個的遊戲積分要可以累績計算。

# 第二章 系統討論

## 2.1 系統流程圖

新建帳號或載入帳號

選擇遊戲

玩家選擇出拳

電腦隨機出拳

玩家與電腦隨機擲骰子點數

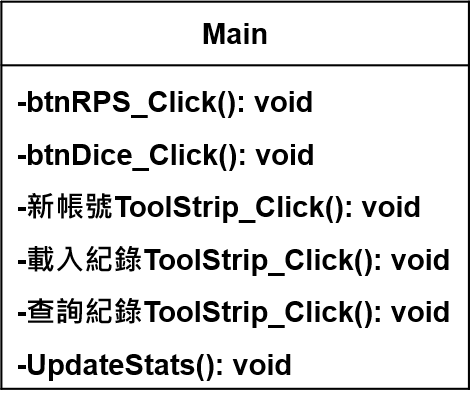
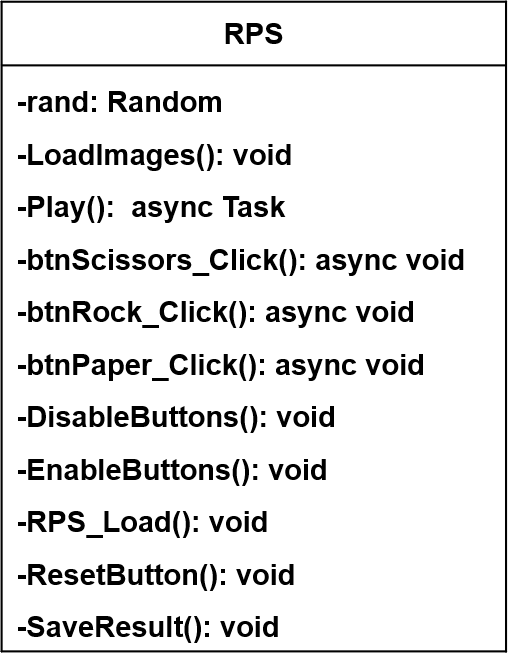
儲存紀錄

擲骰子

猜拳

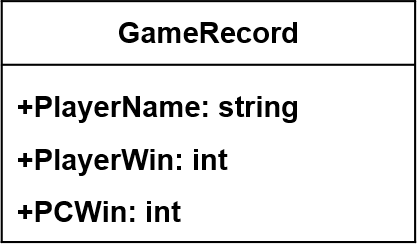
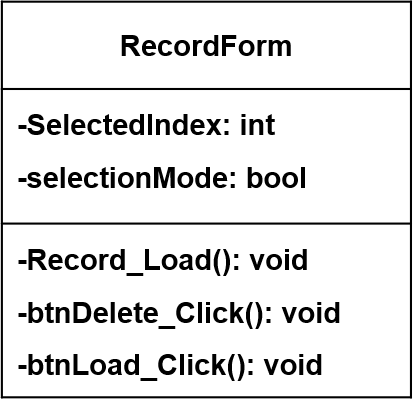
開始先建立新帳號，或選擇已經有紀錄的帳號，若選擇已經有紀錄的帳號則會載入儲存的紀錄；接著選取要進行的遊戲，每次勝負都會累加紀錄。離開遊戲時自動更新紀錄。

## 2.2 系統類別圖



**Association**

**關聯**



**Aggregation聚合**

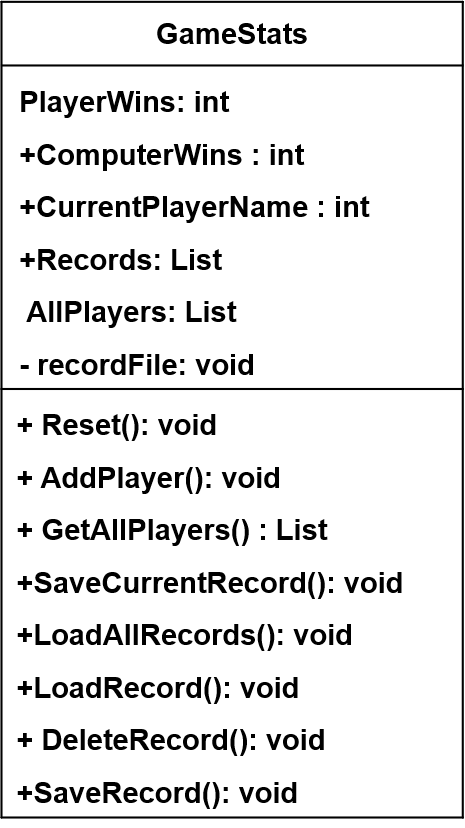
**Composition組成**

**Association**

**關聯**

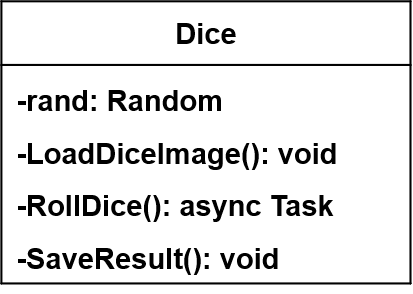
**Implementation**

**實作**



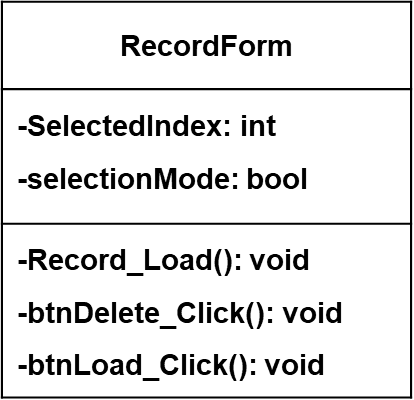
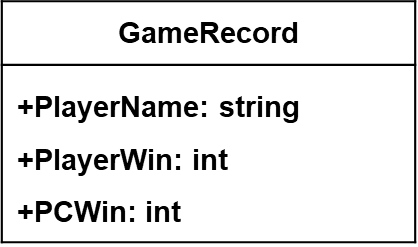
**Implementation**

**實作**



**Implementation**

**實作**



**Aggregation聚合**

**Composition組成**

# 

## 2.3 功能模組說明

Main.cs是一個表單應用，有”MenuStrip’控制項可以選擇新建帳號或者載入紀錄，另外也可以查看紀錄以及刪除紀錄。表單上會顯示玩家以及紀錄，點選猜拳遊戲可進入RPS.cs，點選擲骰子遊戲可以進入Dice.cs。

RPS.cs為猜拳遊戲表單，玩家點選剪刀、石頭或布圖示即啟動遊戲，PC會用亂數選擇所出的拳，然後判別勝負。會執行GameStats.cs紀錄結果。

Dice.cs為擲骰子比大小遊戲，玩家點選擲骰子按鈕可啟動遊戲，玩家與PC都會用亂數決定骰子點數進行比大小。相同執行GameStats.cs紀錄結果。

GameStats.cs 負責紀錄結果在RecordForm.cs 表單。

GameRecord.cs為定義所需紀錄項目。

# 第三章 系統實現

## 3.1 環境開發

a. Visual Studio 2022

b. .Net 8.0

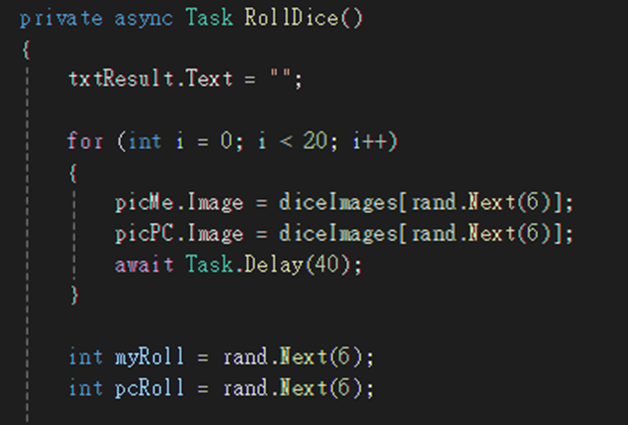
## 3.2 核心功能代碼解析

|  |  |  |
| --- | --- | --- |
| 功能名稱 | 主要類別 | 說明 |
| 建立/載入帳號 | Main、GameStats | 管理帳號名稱與初始化資料 |
| 執行猜拳遊戲 | RPS | 進行遊戲動畫、判斷勝負、儲存結果 |
| 執行擲骰子遊戲 | Dice | 進行遊戲動畫、判斷勝負、儲存結果 |
| 紀錄管理與查詢 | RecordForm、GameStats | 查詢、載入、刪除玩家紀錄 |
| JSON 檔案存取 | GameStats | 所有資料都儲存在 records.json |

## 3.3 關鍵技術實現說明

3.3.1 遊戲動畫

使用for 迴圈與await 搭配應用實現了圖案點時間輪流顯現，使用 await Task.Delay() 會讓畫面持續更新 但不會卡死介面呈現出遊戲中隨機選取的視覺感，有類似動畫的效果。



async：用來宣告 function定義一個非同步方法 (asynchronous method)。

await：用來標記 Coroutine 切換暫停和繼續的點。

async 修飾詞表示該方法是非同步的，也就是它可以使用 await 關鍵字等待耗時的操作（例如 I/O、延遲、網路、動畫等），而不會阻塞主執行程序。

非同步方法通常會回傳：

Task（無回傳值）

Task<T>（有回傳值）

void（通常用於事件處理器）

3.3.2 資料紀錄

使用”System.Text.Json” 套件，可以錄讀取 JSON 轉 List<GameRecord>呈現，也可以將結果存在records.json 保留。JSON（JavaScript Object Notation）是一種輕量級的資料儲存格式，常用來儲存與交換資料。records.json" 是用來儲存遊戲系統中所有帳號的紀錄資料。

*List<GameRecord> ↔ JSON 檔案*

## 3.4 界面設計與使用者操作流程

首先是Main 表單功能可以新建帳號或者載入帳號，以及察看或修改紀錄。Main表單上會顯示帳號以及紀錄，點選表單上的遊戲按鈕即可進入所選的遊戲。

進入猜拳遊戲(RPS)，表單上有三種拳的圖示，玩家選擇想出的拳即啟動遊戲，電腦方用動畫呈現隨機出拳，而後判定輸贏。關閉遊戲回到Main表單，輸贏的局數會更新。

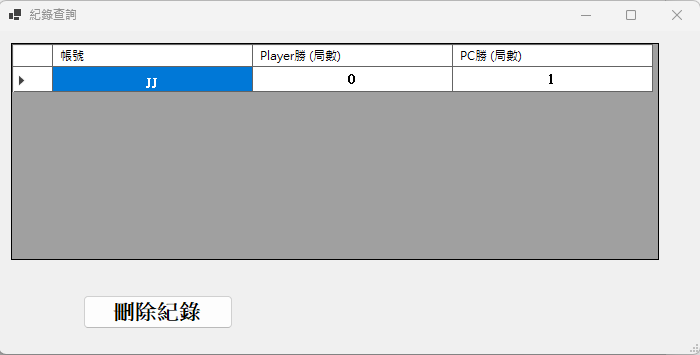
進入擲骰子遊戲(Dice)與猜拳遊戲流程類似，只是擲骰子遊戲點選擲骰子啟動遊戲，而雙方都會用動畫呈現隨機點數，而後判定輸贏。關閉遊戲回到Main表單更新局數。

Main表單更新輸贏的結果也會更新在記錄檔儲存。

# 第四章 測試與驗證

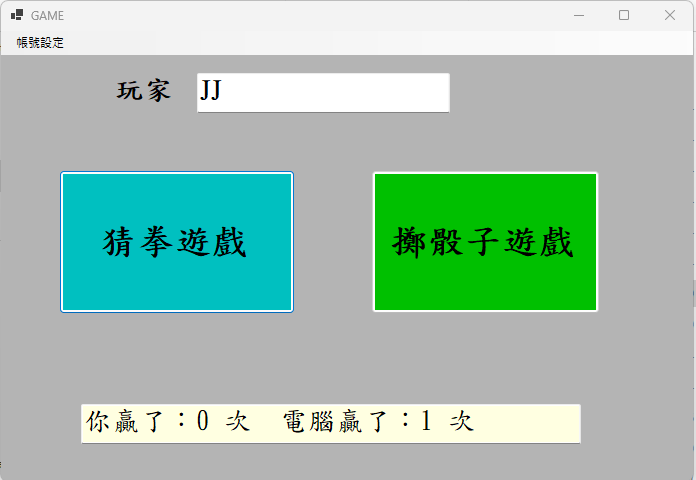
## 4.1 舊紀錄確認

資料正確。可以搜尋到之前的帳號以及紀錄。



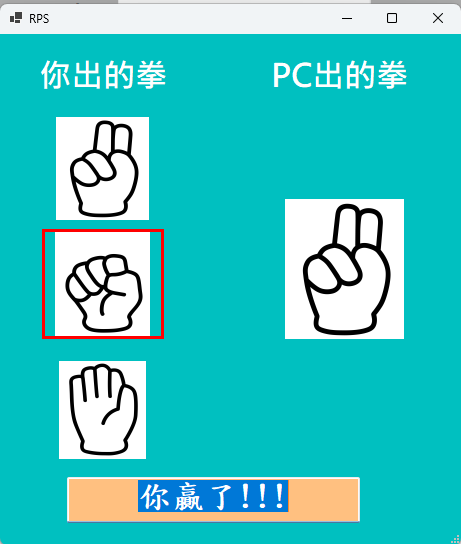
## 4.2 載入紀錄確認

執行正確。表單上顯示玩家名稱，以及紀錄顯示正確。

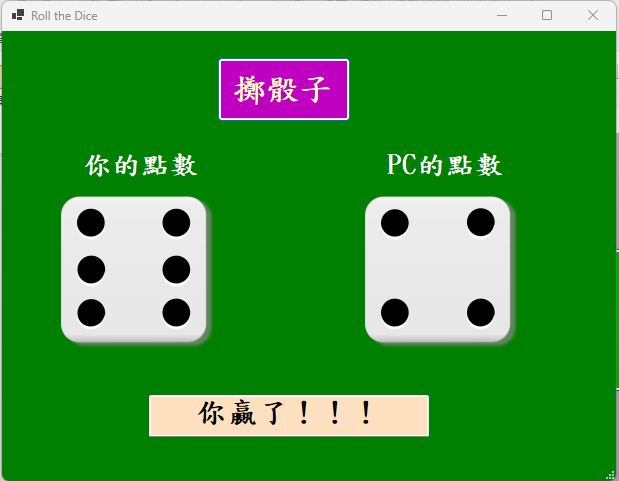


## 4.3 遊戲流程與結果判定

猜拳遊戲執行正確。有標示玩家選取的拳，PC也隨機顯示所出的拳，輸贏判別也正確。

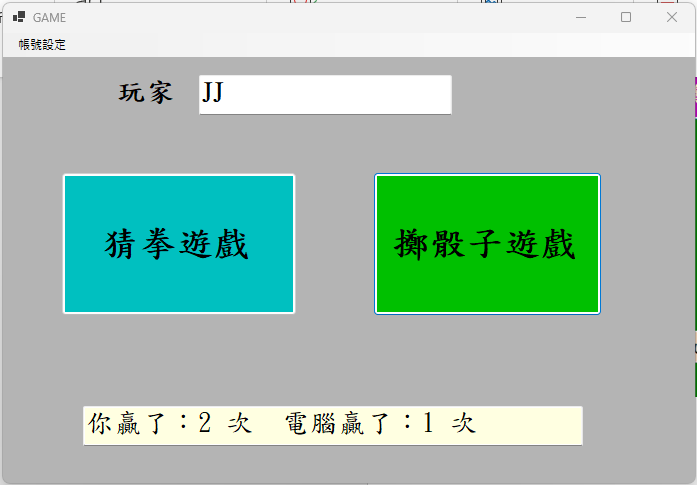


擲骰子遊戲執行正確。玩家和PC隨機出現點數，輸贏判別也正確。



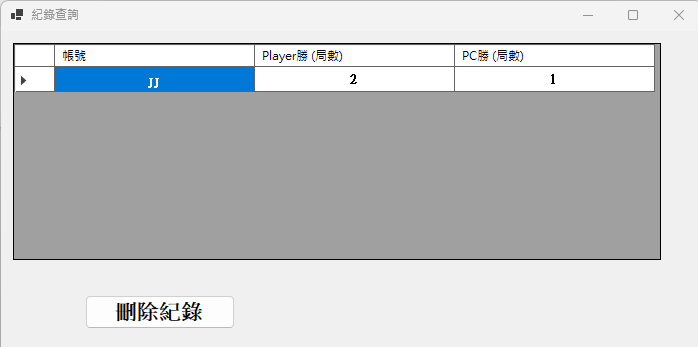
## 4.4 回傳結果

Main form顯示累計紀錄正確。



## 4.5 紀錄更新

紀錄查詢介面也有正確更新。



# 第五章 結論與未來工作

## 6.1 研究成果總結

最後執行結果均有達到預期目標，可以完整正確記錄結果、表單間轉換正常，也成功呈現動畫效果。對於每場遊戲的輸贏結果可以正確判定。表單畫面簡單並且容易了解，方便進行遊戲以及讀取修改紀錄。

## 6.2 未來改進方向

除了遊戲進行時的動畫效果，若能加上配音配合隨機時緊張氣氛，應該會有更佳呈現，並且增加遊戲進行時的樂趣。

再來是紀錄可以更細節查詢到每個帳號每次輸贏的遊戲項目以及輸贏的結果，應該對於未來設計更複雜遊戲會很有幫助。

# 參考資料

[1] https://ithelp.ithome.com.tw/m/articles/10187156

[2] https://blog.csdn.net/w3cwxl/article/details/142187343

[3] https://blog.csdn.net/LB\_0015/article/details/95487657

[4] https://blog.csdn.net/weixin\_43572477/article/details/100189441

[5] https://mur9mur.pixnet.net/blog/post/44937379

[6] https://www.youtube.com/watch?v=QpgcmHbGoo0

[7]<https://blog.cavedu.com/2012/10/05/%E6%95%99%E5%AD%B8-%E9%AA%B0%E5%AD%90%E6%AF%94%E5%A4%A7%E5%B0%8F%E9%81%8A%E6%88%B2/>

[8] <https://csharpkh.blogspot.com/2017/10/c-async-void-async-task.html>

[9] https://docs.python.org/zh-tw/3.12/library/asyncio.html