

逢 甲 大 學

資訊工程學系

專題研究報告

連線聲音控制遊戲
「好好說話」

指導教授：劉宗杰 教授

學生：D0782956 平祖安

D0746384 李筱文

中 華 民 國 一 ○ 九 年 十 二 月

摘要

本專題為 Google Chrome 小恐龍雙人連線遊玩版本，我們以 Swing 及 AWT 完成 GUI 呈現，以 TCP 協定通訊為基礎實作此遊戲的連線部分，並結合嵌入式系統，利用聲音感測玩家的音量，以操作角色移動。

目錄

摘要.....	2
目錄.....	3
圖目錄.....	4
第一章 緒論.....	5
1.1 遊戲介紹.....	5
1.1.1 遊戲角色.....	5
1.1.2 遊戲概述.....	5
1.1.3 操作方式.....	6
第二章 專題相關作品.....	7
2.1 Chrome 瀏覽器的小恐龍跳躍遊戲	7
2.1.1 主要功能.....	7
2.1.2 差異.....	7
第三章 研究方法.....	8
3.1 Arduino 與 Java 連線方法	8
3.2 物件關係圖.....	9
3.3 TCP 協定通訊方法.....	10
3.4 Server 端的功能.....	11
3.5 Client 端的功能.....	14
3.6 遊戲流程.....	15
第四章 實作結果.....	16
分工表.....	20
參考文獻.....	21

圖目錄

圖 1 選角畫面.....	5
圖 2 遊戲介面.....	5
圖 3 小恐龍遊戲畫面.....	7
圖 4 ESP32 開發版.....	8
圖 5 聲音感測器.....	8
圖 6 物件關係圖.....	9
圖 7 TCP 網路應用程式模型圖.....	10
圖 8 登入及註冊系統序列圖.....	11
圖 9 選角與配對序列圖.....	12
圖 10 遊戲服務序列圖.....	13
圖 11 遊戲流程示意圖.....	15
圖 12 登入介面.....	16
圖 13 註冊介面.....	16
圖 14 遊戲初始畫面.....	17
圖 15 選角畫面.....	17
圖 16 玩家等待畫面.....	18
圖 17 遊玩畫面.....	18
圖 18 勝負畫面.....	19
圖 19 Class Relationship Diagram	20

第一章 緒論

1.1 遊戲介紹

1.1.1 遊戲角色

正式開始遊戲前會要求玩家選取一個角色，如下圖，預設為左一的橘貓。由於操作方式為聲控，若在倒數結束前進行切換的動作，將倒數重設，當倒數結束時進入下個階段。



圖 1 選角畫面

1.1.2 遊戲概述

遊戲介面如圖，遊戲目標為操作角色避開障礙物，並盡量增加分數以獲得勝利。

背景將隨時間推移，並在地面隨機產生障礙物，天空隨機產生加分獎勵，玩家可藉由提高輸入音量，使角色跳過障礙物，或拿取加分獎勵，當玩家所控制的字符碰到障礙物時，判定為失敗，雙方都失敗時遊戲結束，分數高者獲勝。



圖 2 遊戲介面

1.1.3 操作方式

使用者對聲音感測器發出聲音，除了註冊和登入以外的功能都是依靠聲音進行延續。包含開始遊戲、選擇角色、遊玩。

1.2 研究動機

本遊戲靈感來自 chrome 瀏覽器提供的小恐龍跳躍遊戲，小恐龍的設計為單機遊戲，因此我們想製作一款可讓兩個玩家一起同樂的遊戲。為了增添趣味性，除了基本的避開障礙物，我們另設計了加分機制，讓兩個玩家競爭，並將操作方式改變成聲音輸入提升可玩性。

1.3 研究目標

當兩個玩家完成連線並開始遊戲後，即時更新遊戲畫面，此應用屬於隨機制的運作模式，參與的成員依照可隨機進行動作。

本專題以 Java 來撰寫，所設計的網路應用需分成 Server 端及 Client 端的模組，這兩模組的基本功能(要求)如下：

- Server 模組

- (1)Server 端一律以 Java 撰寫。
- (2)管理成員的加入、離開、…等作業。
- (3)負責初始化應用的作業。
- (4)扮演仲裁者的角色，判斷獲勝者。
- (5)維護各組運行中的資訊，偵測並處理 Client 不正常離線的情況。

- Client 模組

- (1)Client 端一律以 Java 撰寫並以圖形化介面來呈現。
- (2)成員的加入、離開…等介面。
- (3)從 Server 端接收其他 Client 的動作並顯示。
- (4)讀取參與者進行的動作，確認進行的合法性後，傳送給 Server 端。
- (5)偵測並處理與 Server 連線不正常中斷的情形。

第二章 專題相關作品

2.1 Chrome 瀏覽器的小恐龍跳躍遊戲

2.1.1 主要功能

遊戲畫面如下圖，玩家使用空白鍵或方向上鍵控制小恐龍跳躍，遊戲中地面及天空隨機出現障礙物，分數隨時間上升，碰到障礙物則計分停止，遊戲結束。

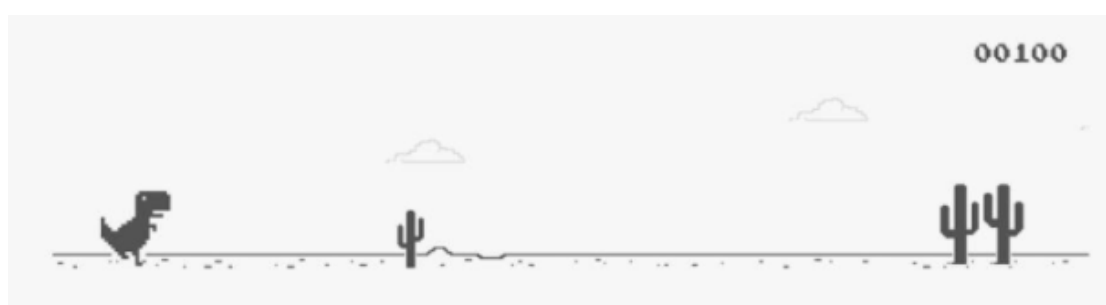


圖 3 小恐龍遊戲畫面

2.1.2 差異

小恐龍跳躍遊戲為 javascript 所寫的單機遊戲，本專題遊戲方式與其相像，與其相異之處在於，本專題為 java 所撰寫的雙人連線遊戲，遊戲設計加入了加分獎勵，操作方式為聲音控制。

第三章 研究方法

3.1 Arduino 與 Java 連線方法

此專題中，我們使用 ESP32 開發版，連接聲音感測器當作遊戲輸入，為了讓 Java 寫成的 Client 端接收到輸入，這裡使用了 Java 程式讀取 Serialport 的函式庫 jSerialComm，用監聽器的形式，只要輸入的音量大於門檻值，就會產生一個中斷。利用此中斷來觸發角色跳躍。

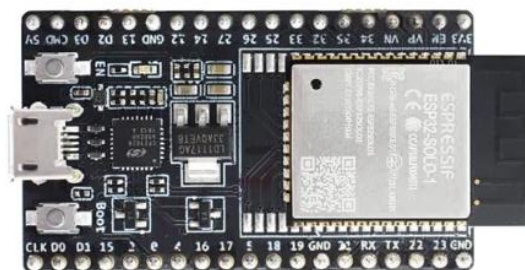


圖 4 ESP32 開發版



圖 5 聲音感測器

3.2 物件關係圖

從下圖可以看到，SingleGamingPanel 和 ConnectGamingPanel 幾乎都跟其他物件彼此有 Association 的關係，我們採用的方式是先完成單機版本，之後再加上連線的程式碼作出最後成品，因為設計上，我們採用互傳的方式，故連線版會少些許 attributes。

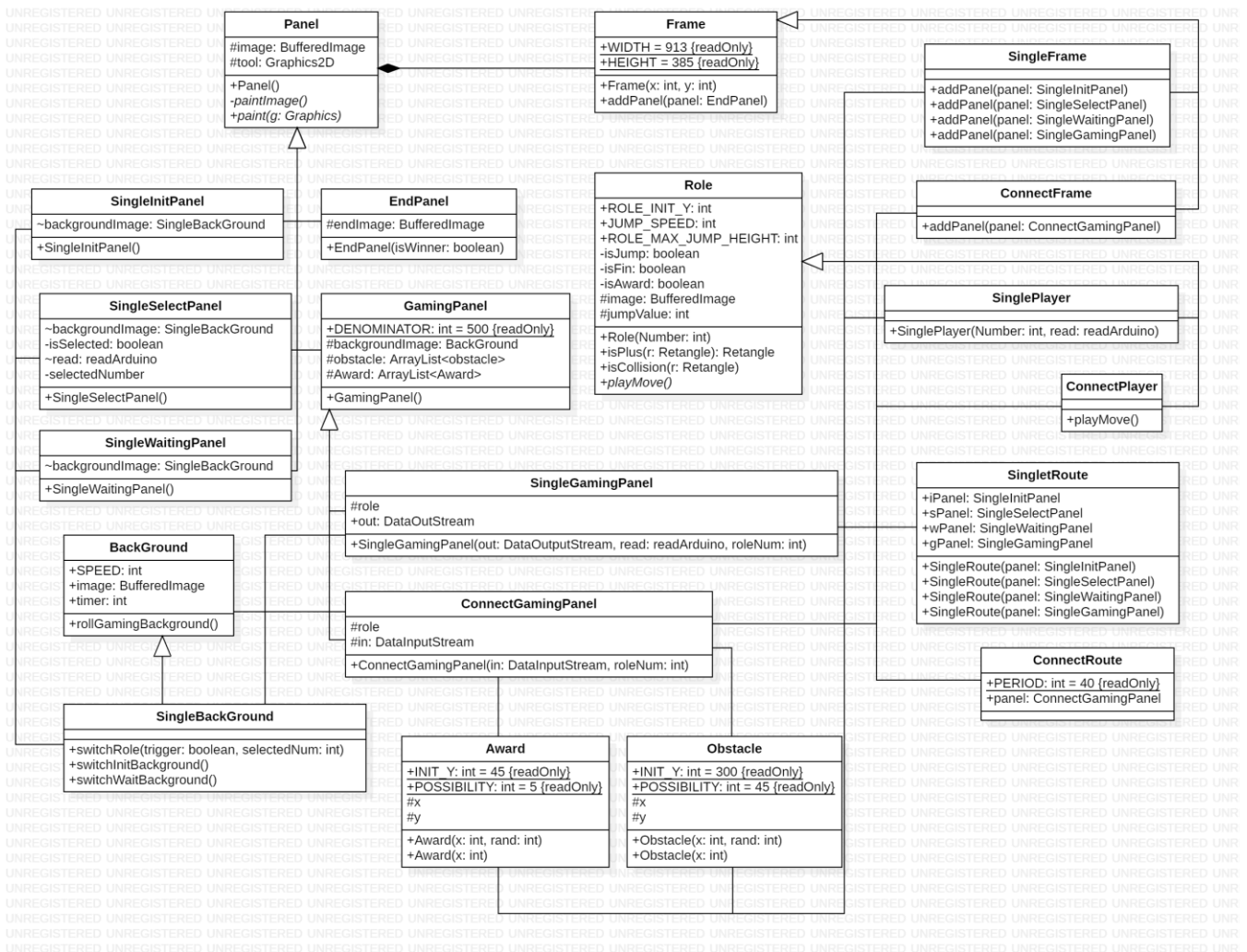


圖 6 物件關係圖

3.3 TCP 協定通訊方法

TCP (Transmission Control Protocol) 是一種連接導向的、可靠的、基於位元組流的傳輸層通訊協定。在 Internet protocol suite 中，TCP 層是位於 IP 層之上，應用層之下的中間層。不同主機的應用層之間經常需要可靠的、像管道一樣的連線，但是 IP 層不提供這樣的流機制，而是提供不可靠的封包交換。

應用層向 TCP 層傳送資料流，然後 TCP 把資料流分割成適當長度的報文段。之後 TCP 把結果封包傳給 IP 層，由 IP 層通過網路將封包傳送給接收端實體的 TCP 層。TCP 為了保證不丟失封包，就給每個封包一個序號，同時序號也保證了傳送到接收端的按順序接收。然後接收端實體對已成功收到的封包發回一個相應的確認 (ACK)。如果傳送端實體在合理的往返時延 (RTT) 內未收到確認，那麼對應的封包就被假設為已遺失，將會進行重傳。

相對於不穩定、有可能掉封包的 UDP 協定，TCP 更適合此專題目標。

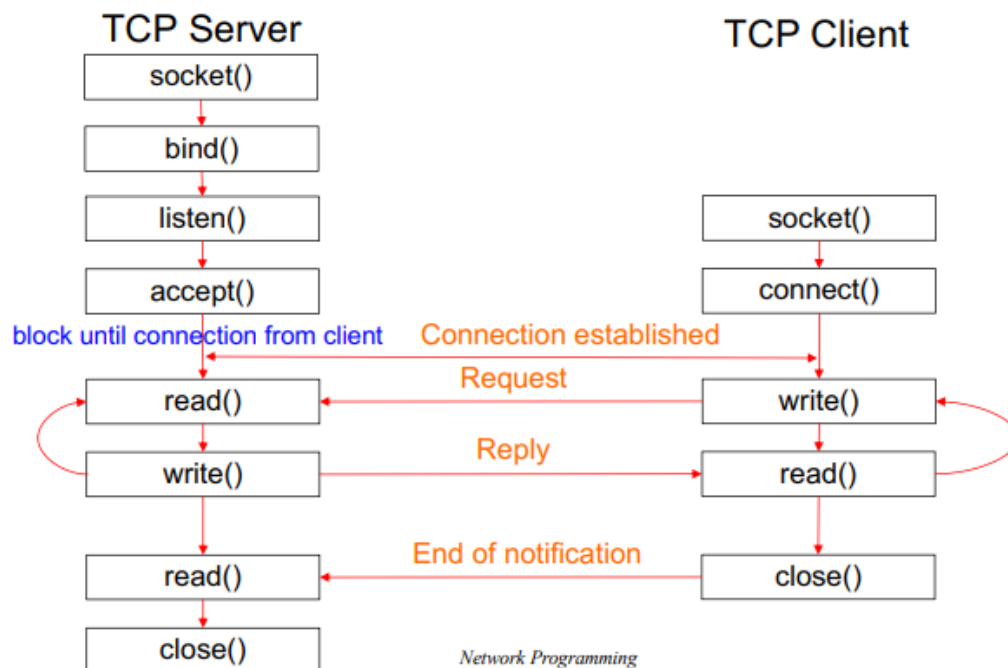


圖 7 TCP 網路應用程式模型圖

3.4 Server 端的功能

Login/Register:

負責接收 client 傳過來的帳號密碼，並確認是否有效。

Pair:

確認帳號有效後，server 將連線的 client 排入等待佇列，當佇列中有兩個 client，server 會將其配對，開始遊戲。

Gaming:

當遊戲開始時，server 產生 thread 負責傳遞兩方 client 的參數，讓玩家的視窗定期更新。

End:

若遊戲出現勝負或不正常中斷，server 的 thread 會將結果傳送至兩方，之後將連線關閉。

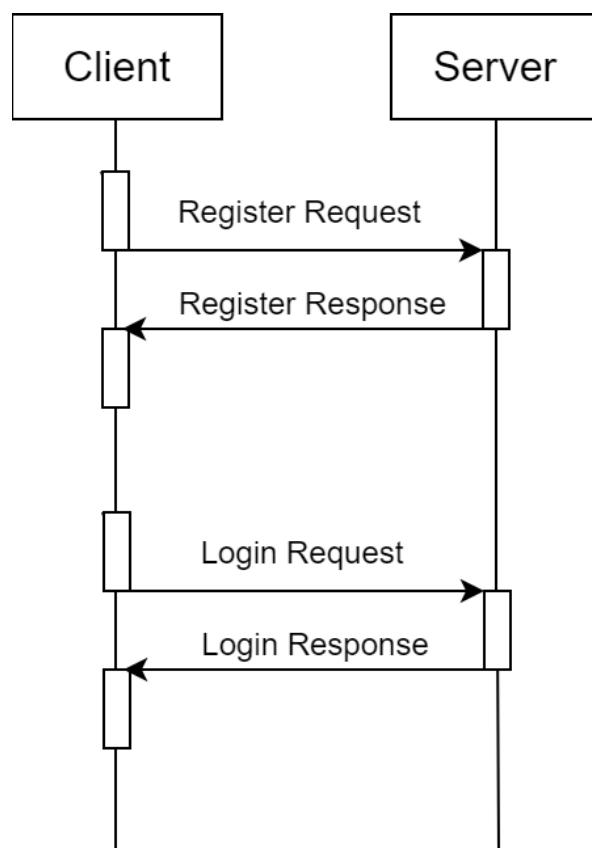


圖 8 登入及註冊系統序列圖

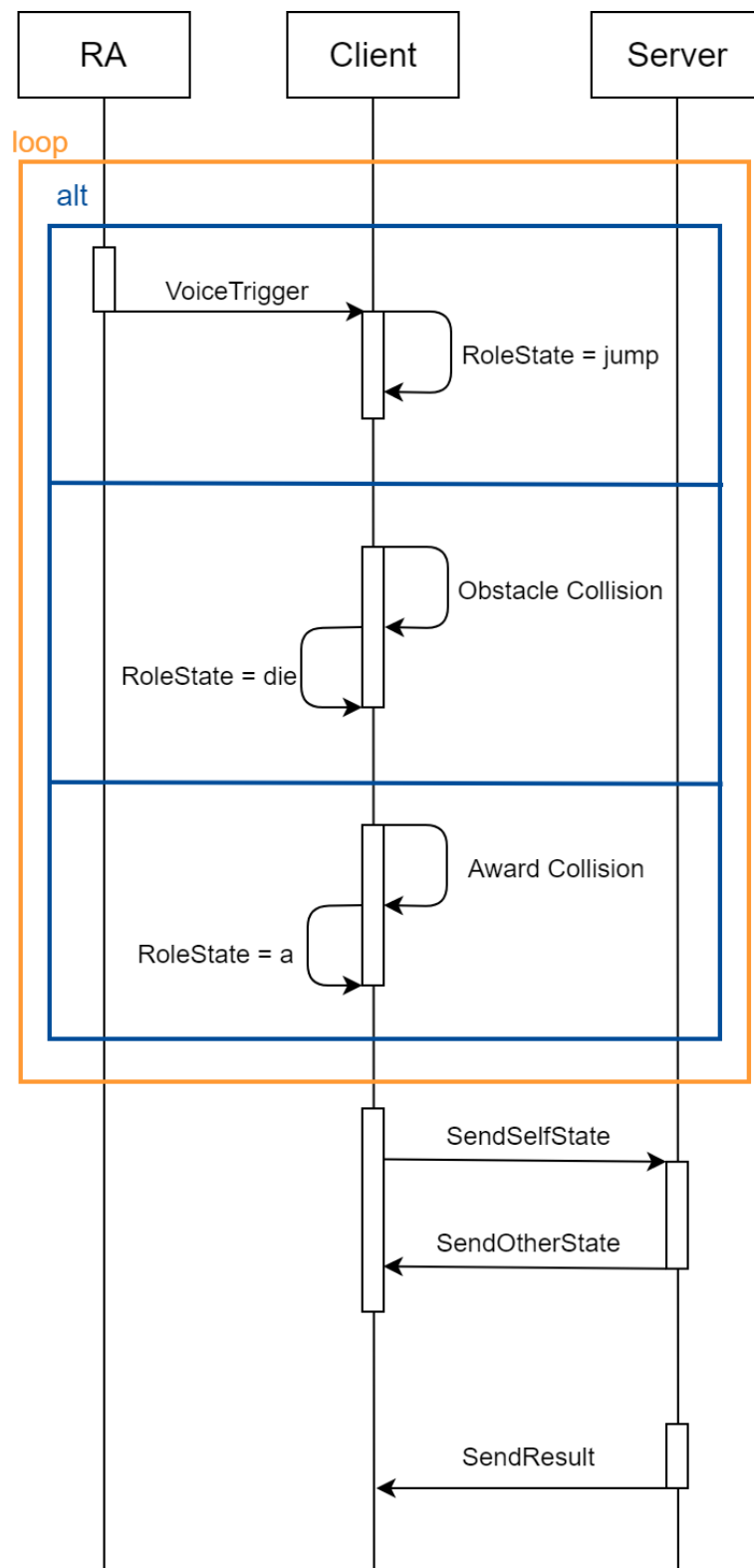


圖 9 選角與配對序列圖

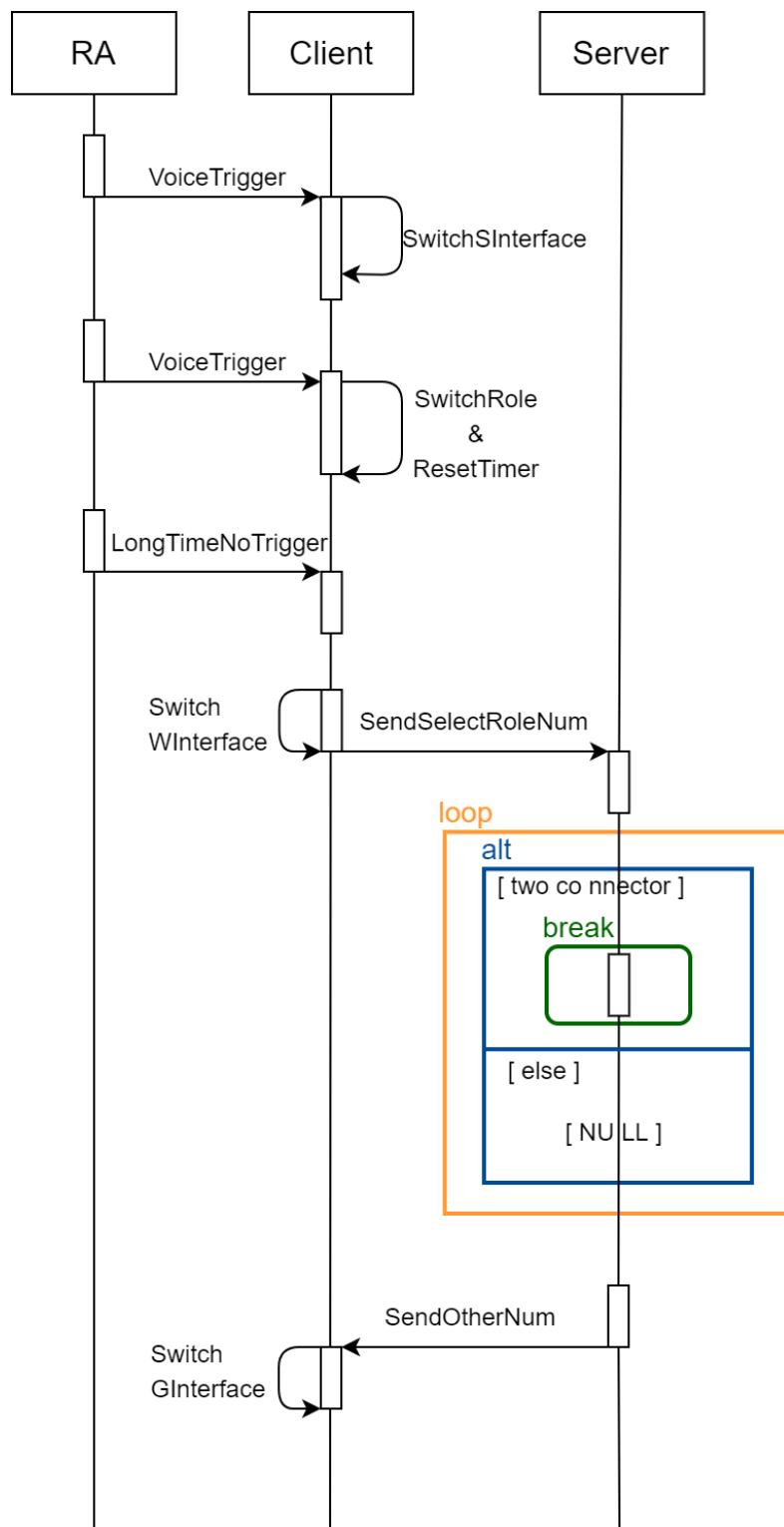


圖 10 遊戲服務序列圖

3.5 Client 端的功能

Phase Register:

使用者需輸入帳號和密碼，而系統會丟到 Server 進行檢查。

Phase Login:

使用者輸入帳號密碼，Server 進行檢查，成功即可進入 Phase Init。

Phase Init:

接收聲音，即可進入 Phase SelectRole，裡面附有動態文字，供使用者注意遊玩方式。

Phase SelectRole:

每接受一次聲音，即可切換角色，上面有倒數時間，若期間有重新選擇，會倒數重置，倒數結束進入 Phase Wait。

Phase Wait:

先選擇完角色者，需等待另一方參加者，此階段提供等待，當雙方都完成時，會暫留一下，接著進入 Phase Gaming。

Phase Gaming:

提供兩個畫面，上半部是自身遊玩狀態，且固定時間將送給對方自身狀況，下半部是對方遊玩狀態，持續接收對方狀態，並定期更新，當雙方狀態都結束時，進入 Phase End。

Phase End:

顯示誰勝誰負。

3.6 遊戲流程

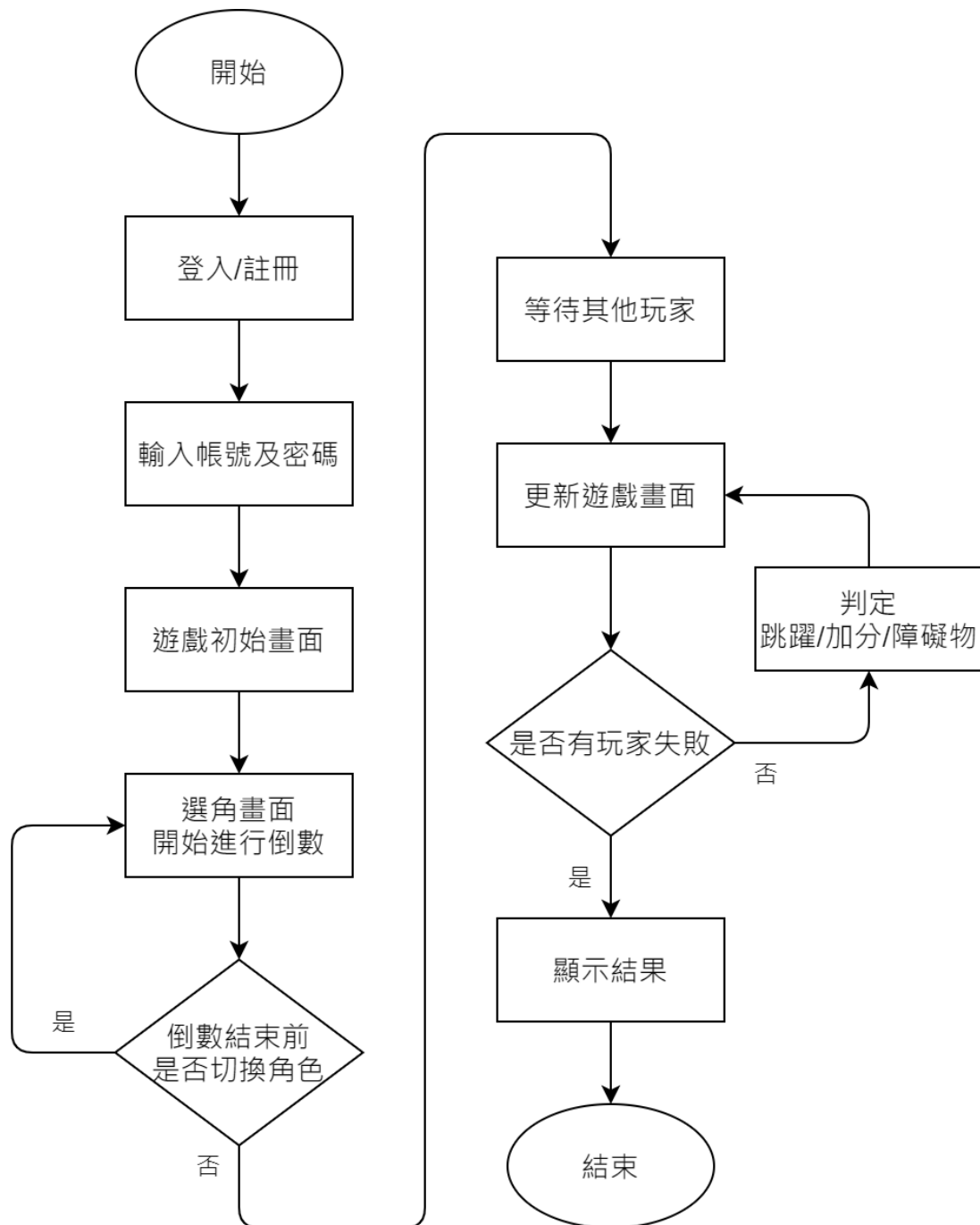


圖 11 遊戲流程示意圖

第四章 實作結果

4.1 註冊或登入

當按下 Register，進入註冊介面，當按下 Login，且帳號密碼 Server 有對應的一組時，進入開始畫面。

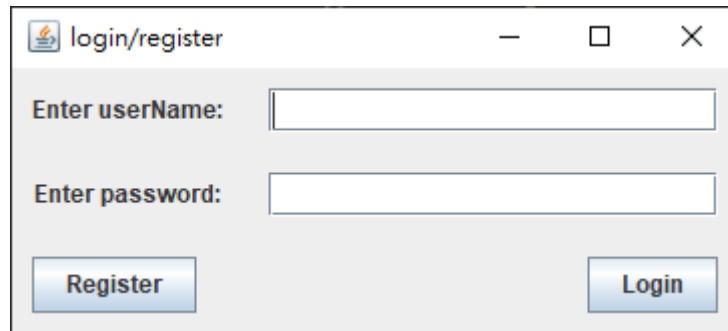
A screenshot of a Windows-style application window titled "login/register". The window has a standard title bar with minimize, maximize, and close buttons. Inside the window, there are two text input fields. The first is labeled "Enter userName:" and the second is labeled "Enter password:". Below these fields, there are two buttons: "Register" on the left and "Login" on the right. Both buttons are blue with white text.

圖 12 登入介面

4.2 註冊介面

當按下 Register，即可註冊一筆新的帳號密碼。


A screenshot of a Windows-style application window titled "Register". The window has a standard title bar with minimize, maximize, and close buttons. Inside the window, there are two text input fields. The first is labeled "Enter userName:" and the second is labeled "Enter password:". Below these fields, there is a single button labeled "Register" on the right side. The button is blue with white text.

圖 13 註冊介面

4.3 開始畫面

發出聲音即可進入選角介面，畫面紅色框起來的區域是動態的，希望可以加強使用者注意，以便知道如何進行遊戲。



圖 14 遊戲初始畫面

4.3 選角畫面

等待五秒時，自動切換至等待介面，發出聲音時，倒數重置，並同時切換角色，而選擇的角色即為上方有貓掌符號的貓咪。



圖 15 選角畫面

4.4 等待畫面

先到達的需等待另一方，而雙方都到達時，依然會有短暫的時間，提供使用者預備，並進入遊玩畫面。



圖 16 玩家等待畫面

4.5 遊玩畫面

使用者依靠聲音，控制角色上下，過程當中，雙方不斷透過 server 傳遞自己目前的狀態，以及接收對方的狀態。



圖 17 遊玩畫面

4.6 勝/敗畫面

判定勝負，給予相對應的畫面



圖 18 勝負畫面

分工表

平祖安:紅色部分，定義傳遞的資料，配合對方所需資料

李筱文:藍色部分，以最少參數運作，保持連線品質

其他:根據兩部分所做的基本物件

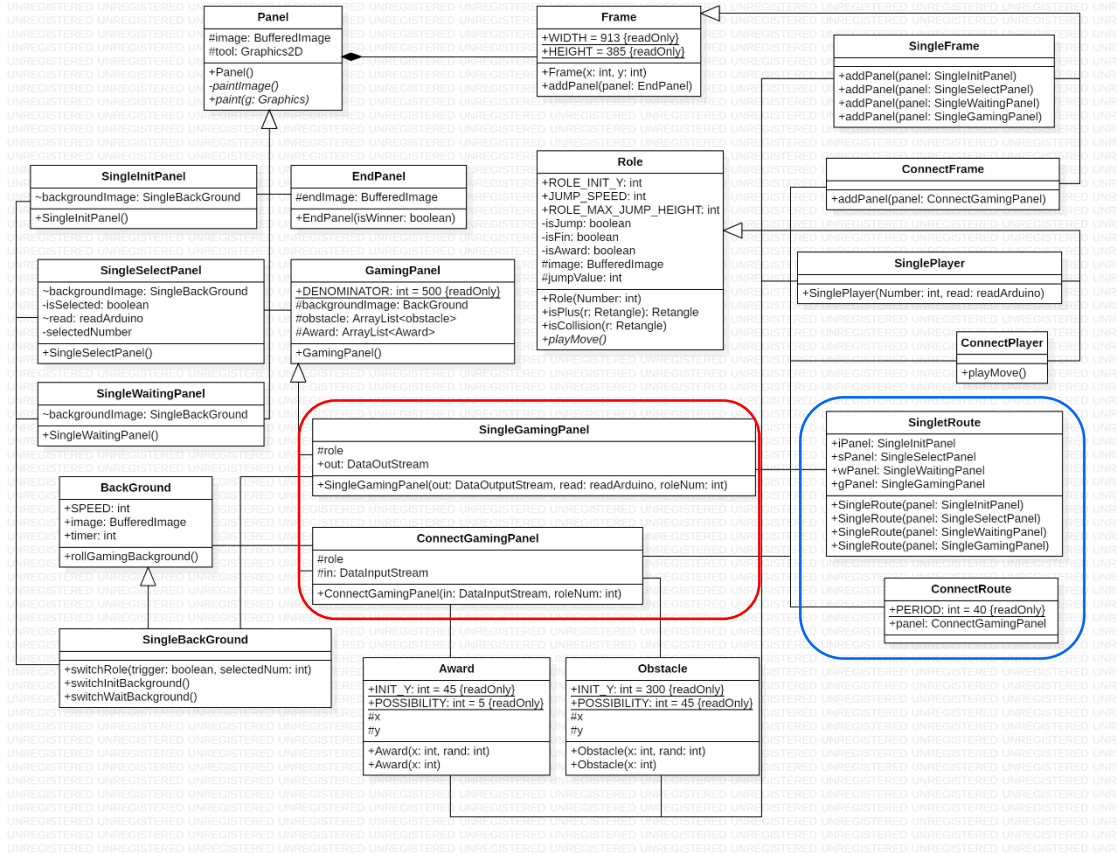


圖 19 Class Relationship Diagram

參考文獻

ESP32-開發環境

<https://makerpro.cc/2020/06/how-to-install-and-configure-esp32-development-environment/>

Socket 相關資料

<https://stackoverflow.com/search?q=socket&s=bd2a2f74-6da4-470b-9319-5c658b5aef15>