

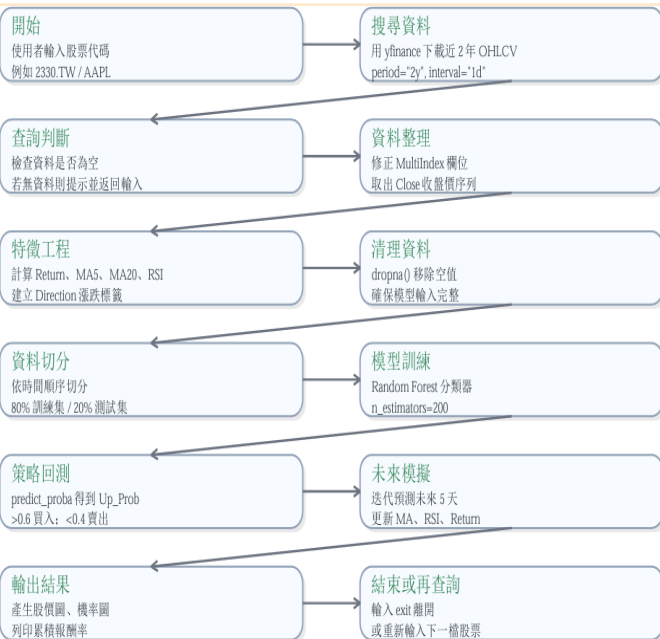
智慧科技學院程式設計專題海報

利用機器學習預測分析股票趨勢

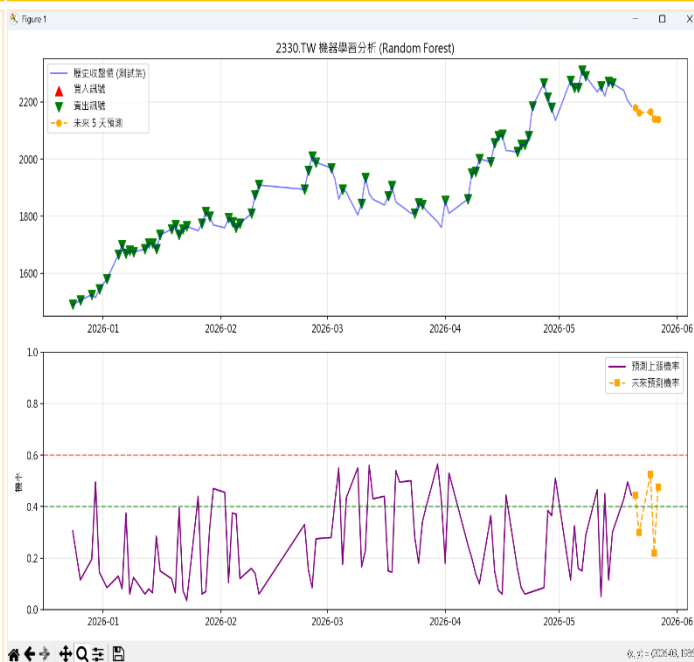
研究動機與目的

傳統技術分析多依賴人工指標與主觀經驗，難在瞬息萬變的金融市場中進行多維度的特徵動態整合。本專題建構了一個自動化的量化預測流水線 (Pipeline)，旨在透過機器學習演算法，將傳統技術指標轉化為具備機率預測能力的量化策略，以提升交易決策的客觀性。

程式流程圖



執行程式結果



執行成果說明

本專題提供標準的機器學習量化交易流程參考。從抓取股票資料、計算技術指標開始，透過時序切分資料，並以隨機森林模型結合機率濾網進行策略回測與未來五天預測，最終視覺化呈現累積報酬率。

電機系

指導老師: 張恩誌

學生: 林楷崴、盧胤輔、郭兆鵬、陶俊銘

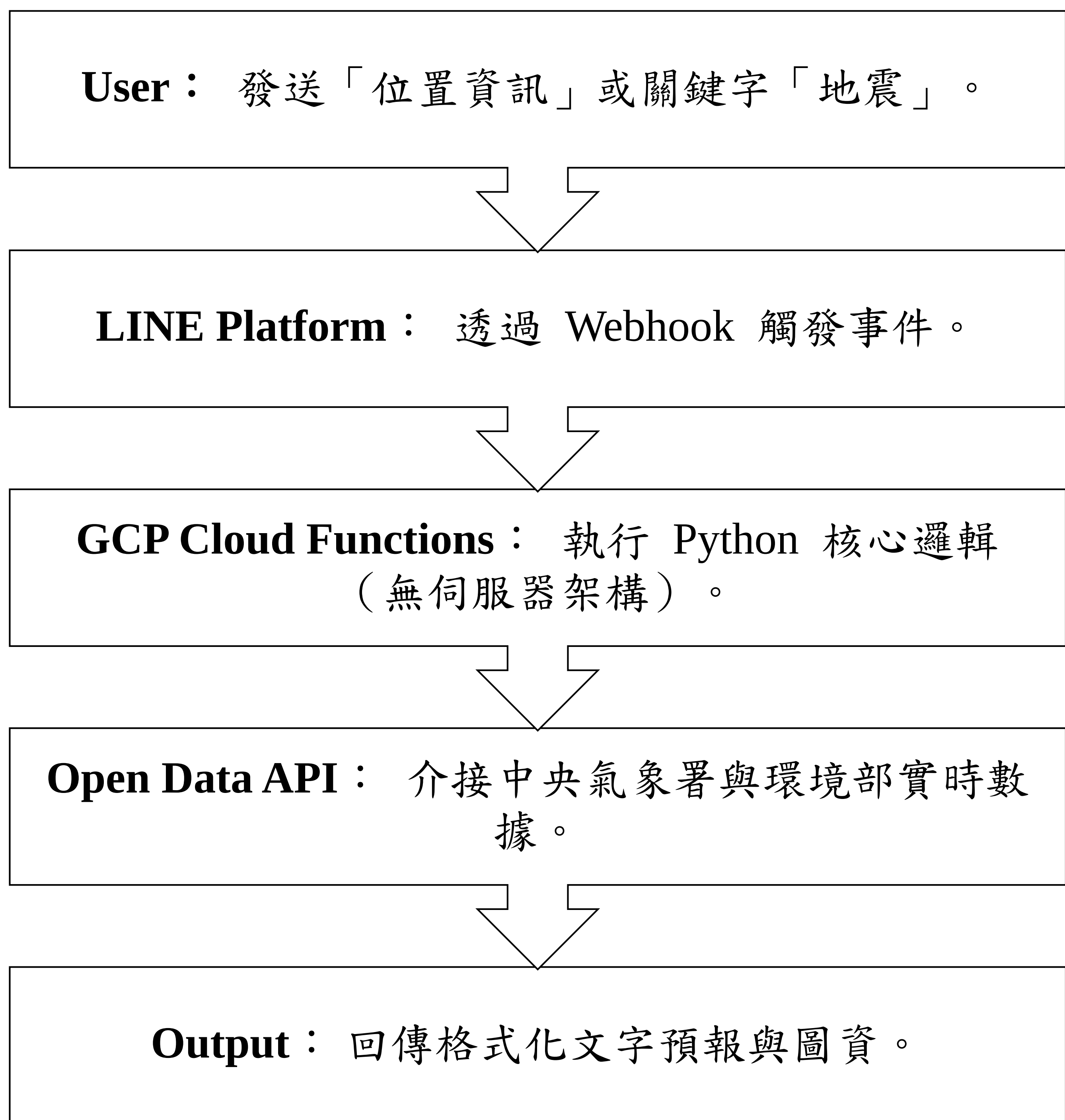
智慧科技學院電機系程式設計專題海報

題目：台灣天候與環境數據即時整合LineBot

一、研究動機

- 生活痛點：**現有氣象 App 資訊碎片化，查詢特定區域需多層點擊，且缺乏地震、空品整合。
- 技術轉型：**初期參考 Vocus 技術文件進行數據探勘，為提升現實應用價值，參考 OXXO Studio 實作指引，將 Code-based 腳本轉化為普及率高的 LINE Bot 介面。
- 目標：**打造一個「免下載、零學習、高強健性」的防災助手。

二、系統架構圖



三、核心演算法

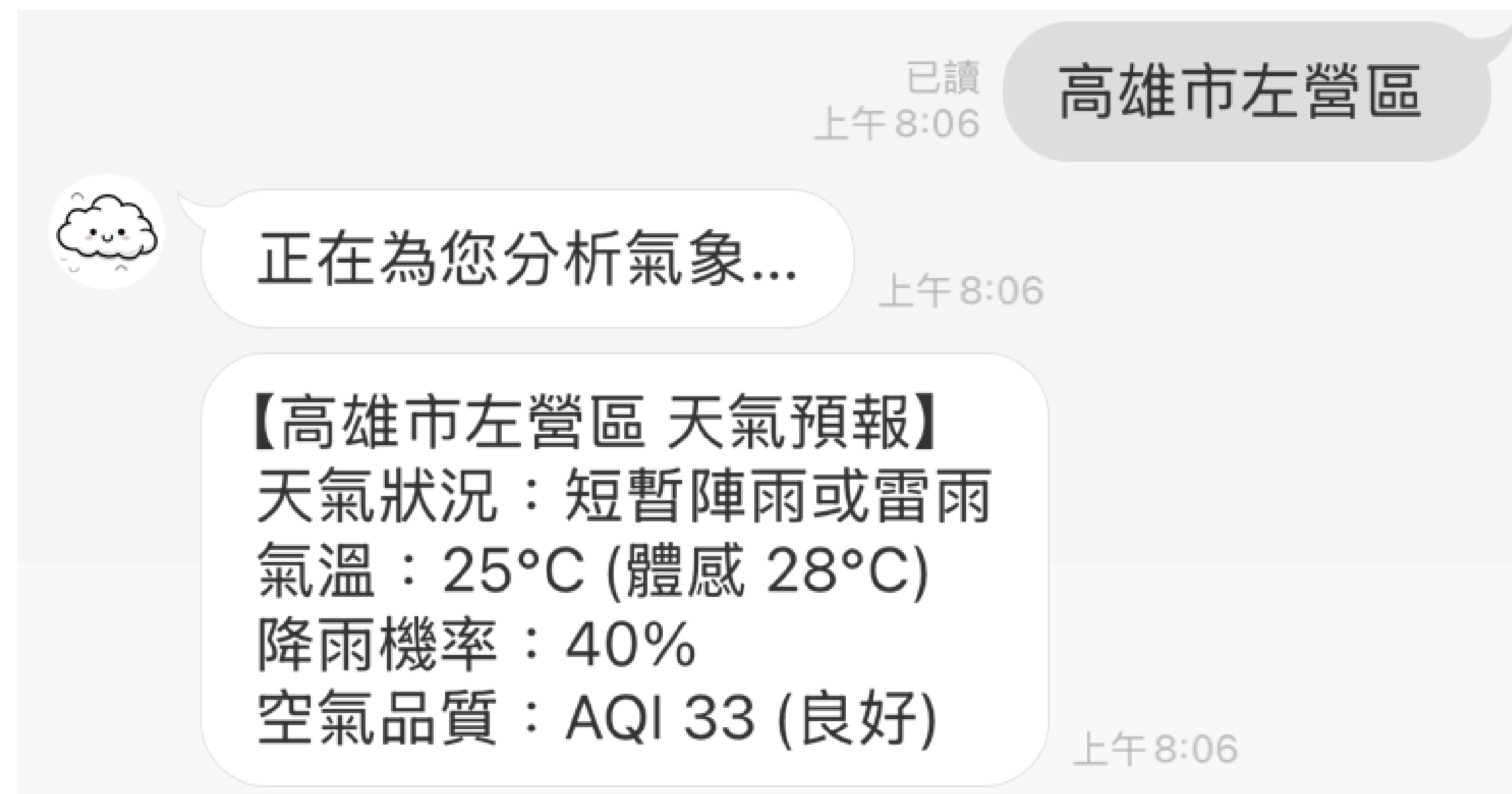
- 鄰近測站容錯演算法 (Fallback Mechanism)：**
為解決特定行政區查無 AQI 測站之問題，系統會自動搜尋同縣市最近「備份測站」，並標註 (鄰近測站：XXX)，確保數據回饋不中斷。
- 防禦性程式設計 (Defensive Programming)：**
- 型別動態檢查：**使用 isinstance() 處理 API 結構突變 (Dict 變 List)。
- 例外攔截：**透過 try-except 捕獲連線超時。

四、課本知識應用

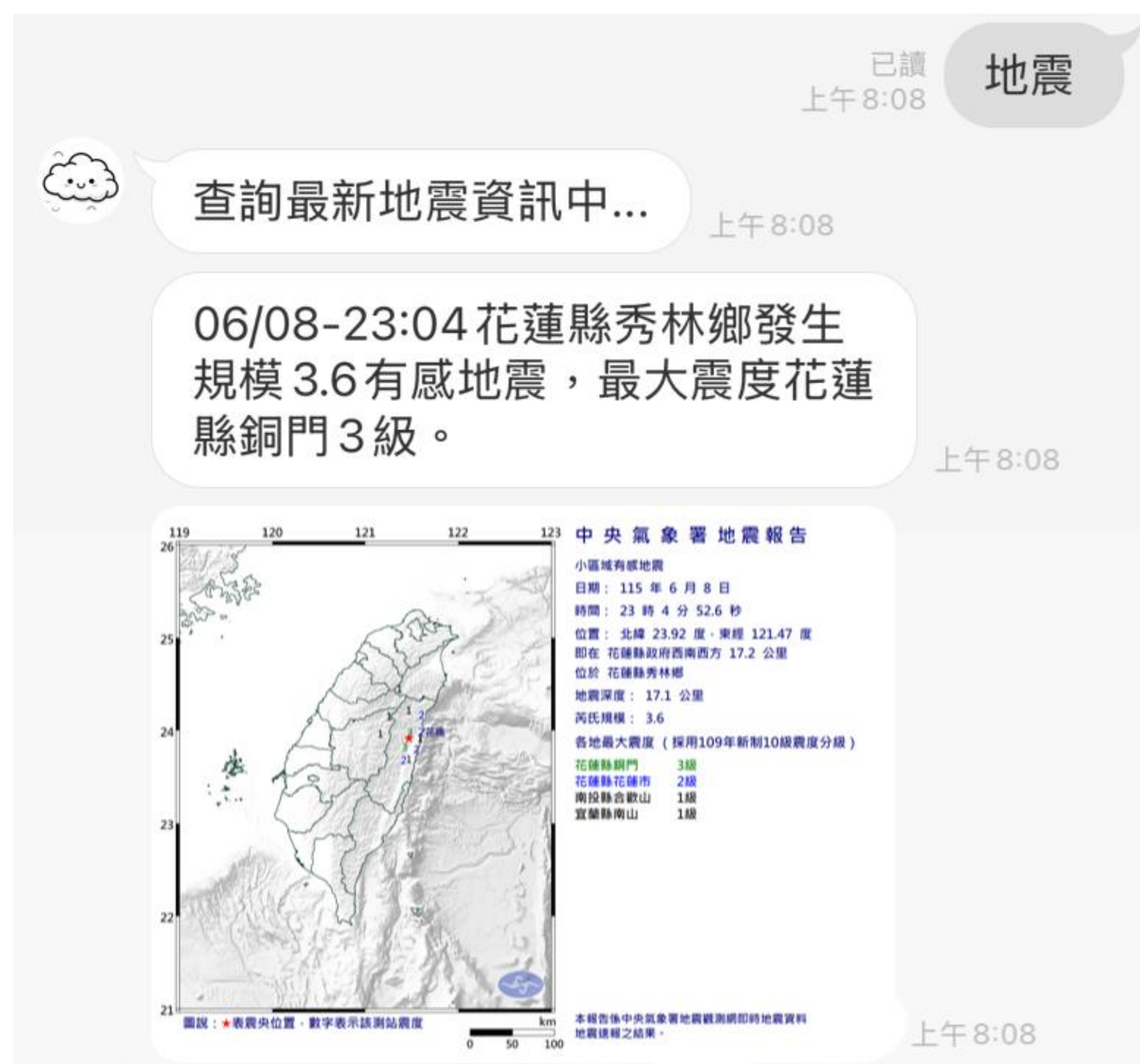
- Ch03-05：**判斷控制流程、迴圈遍歷 JSON 節點、字典與串列數據解析。
- Ch06：**函式封裝與參數傳遞，提升程式碼模組化。
- Ch08：**錯誤與例外處理機制，保障雲端環境穩定性。
- Ch12：**外部模組 requests 與網際網路資源整合。

五、呈現結果

- 即時天氣與空氣品質：**自動解析定位，整合氣溫、降雨機率與 AQI 於單一訊息。



- 防災視覺化：**抓取最新地震報告圖與雷達回波。



六、結論與展望

- 實作價值：**成功克服開放資料不穩定性，打造高可用性之災防工具。
- 未來展望：**預計導入 Cloud Scheduler 實現早晨主動預報功能，並針對極端氣象（豪雨/高溫）實作預警推送。

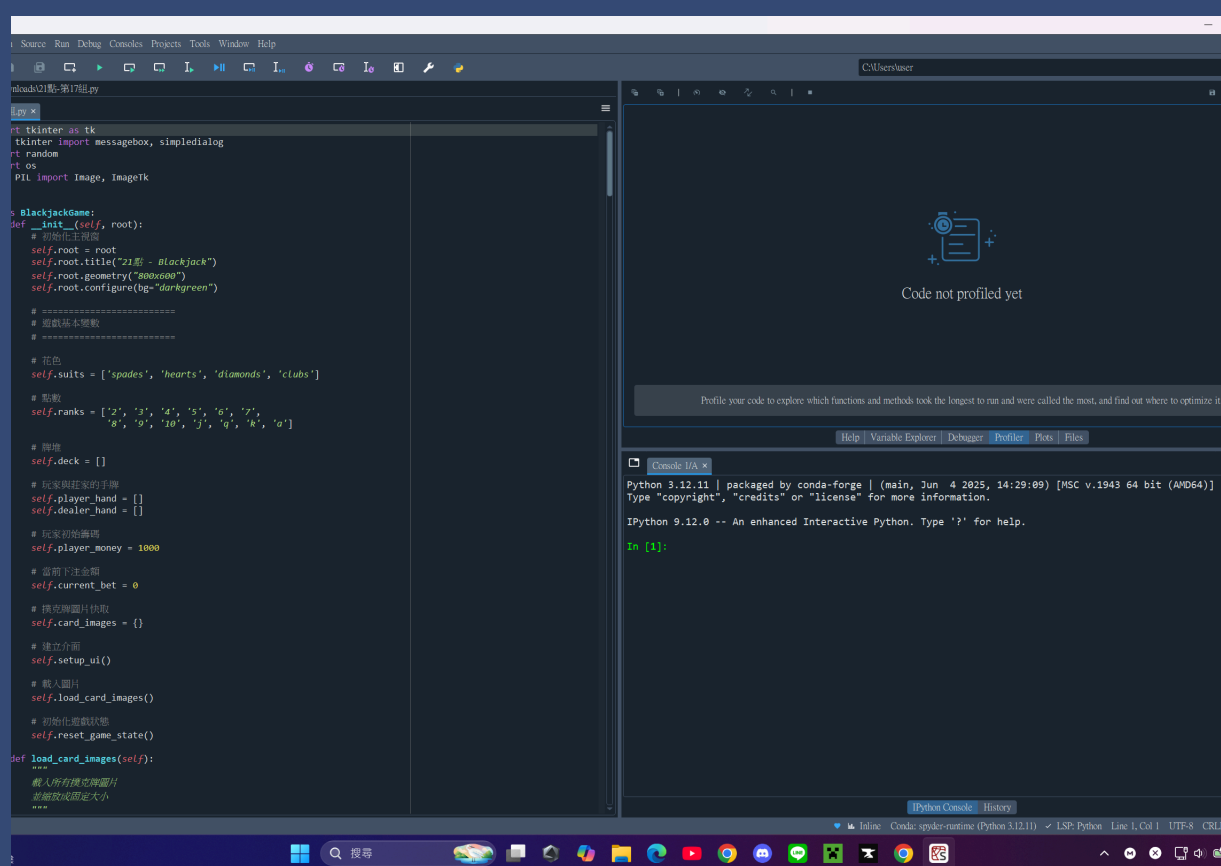
指導老師：黃蓮池 學生：賴雨葳、周芷瑜



智慧科技學院程式設計

21點

本專題運用Python 程式語言結合Tkinter圖形化介面，開發出一款具備下注機制的21點遊戲。系統採用物件導向架構(OOP)封裝遊戲資料，並用串列 (List) 與字典 (Dictionary) 等資料結構來精確管理牌組狀態與圖形素材；在遊戲流程上，也利用迴圈邏輯令莊家在未滿17點前必須自動補牌，模擬真實賭場的莊家行為。



指導老師:黃蓮池教授
學生:李翊聖、詹閔翔、郭承凱

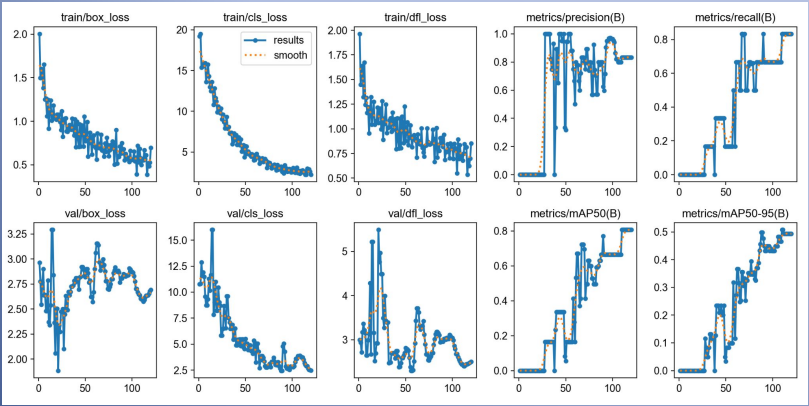
智慧科技學院程式設計專題海報

靜脈炎影像辨識

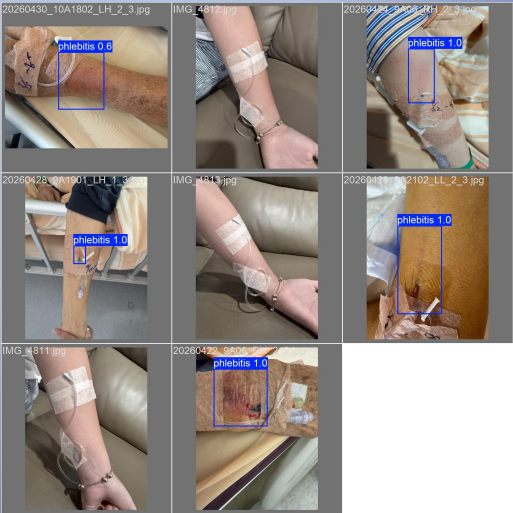
簡介

本專題「基於深度學習之臨床靜脈炎影像自動化辨識系統」旨在解決臨床護理中周邊靜脈留置針因引發靜脈炎，而過去高度依賴人力肉眼評估、容易有主觀經驗落差與照護負擔的痛點。為此，採用YOLO 系列物件偵測演算法，針對臨床留置針與周邊皮膚影像進行關鍵特徵偵測，自動精準定位並框選出靜脈炎(Phlebitis)發炎區域。相較於傳統需將整張照片丟入分類、無法指出確切病灶位置的「一般影像辨識模型」，本專題所使用的物件偵測技術，在臨床實用性上就像是「GPU 之於 CPU」：一般影像辨識(如 CPU) 擅長宏觀、序列化的通才分類，而物件偵測(如 GPU) 則具備強大的並行與精準聚焦能力，能在一瞬間從複雜的醫療背景中，同時精確鎖定並框選出多個細微的發炎座標。根據核訓練結果圖的數據顯示，模型在經歷約120 個 Epoch 的優化迭代後展現出優異的成效：其訓練集與驗證集的損失函數(Loss) 皆呈現穩定收斂且無明顯過擬合，而關鍵評估指標mAP50最終更衝上 0.8 (80%) 的高水準，精準率與召回率亦穩定維持在高點。本系統未來可整合至行動護理裝置，提供臨床即時輔助決策與早期預警，有效提升病人安全並減輕醫療人員的評估負擔。

圖一:YOLO 訓練結果圖



圖二:成果展示



資工系
指導老師：黃蓮池
學生：江叡騏

炸彈人遊戲

我想藉由這次課程來製作一個小遊戲，至於為甚麼選這一類型的遊戲，除了我小時候很愛玩這類的網頁遊戲，也是想用python挑戰複雜一點的遊戲

學員：吳致宏、游崇晏、劉定洲
科系：資訊工程學系
指導老師：黃蓮池

遊戲設計邏輯

遊戲以格子地圖為基礎，地圖會隨機產生不可破壞牆、可炸障礙物與空地。玩家和 AI 在格子間移動，放置炸彈後會倒數爆炸，火焰會往上下左右延伸。

AI 採用「保命優先」邏輯，會先判斷危險區並逃跑，確認安全後才追擊玩家或放炸彈。玩家可透過炸開障礙物取得道具，提升火力、增加炸彈數，或獲得護盾抵銷一次傷害。

遊戲玩法

玩家要在隨機地圖中控制角色(綠色史萊姆)，利用炸彈炸開障礙物、取得道具，並打敗三個 AI 對手。

操作方式

- WASD 或方向鍵：移動角色
- Space：放置炸彈
- P 或 Esc：暫停遊戲
- Enter：開始或重新開始

勝利條件

玩家成功用炸彈擊敗三個 AI，即可獲勝。

失敗條件

玩家被炸彈爆風擊中，或時間結束，就會失敗。

道具效果

- 爆炸範圍道具：增加炸彈火力
- 炸彈數量道具：增加可同時放置的炸彈數
- 護盾道具：可以抵銷一次傷害

程式設計邏輯

本遊戲使用 Python 製作，主要搭配內建的 tkinter 建立遊戲視窗與畫面。程式中使用 Canvas 繪製地圖、角色、炸彈、火焰、道具與介面。

主要用到的程式概念包含：

- class：建立玩家、AI、炸彈與道具物件
- if：判斷移動、爆炸、勝負與道具效果
- for：更新角色、炸彈、火焰與地圖格子
- list / dict：儲存角色、炸彈、道具與 AI 設定
- random：產生隨機地圖與道具
- tkinter：製作遊戲視窗與互動介面

這是遊戲畫面





太空生存戰

閃避隕石 · 收集道具 · 挑戰 Boss · 創造最高紀錄！

學員：蔡凱丞、余偉志、尹宇辰
科系：資訊工程學系
指導老師:黃蓮池



專題動機

現今許多人喜歡射擊與生存類遊戲,因此利用Pygame製作一款結合反應力、操作技巧的太空射擊遊戲。



遊戲玩法

#解壓縮檔案完後，點擊main.exe檔即可進入遊玩畫面

玩家使用鍵盤控制飛船：

- ← → 鍵控制飛船移動
- Space 鍵發射子彈
- 閃避隕石與敵方子彈攻擊
- 收集道具提升生存能力
- 挑戰Boss 並取得高分

當血量歸零時會消耗一條生命並重新復活無敵3秒。



遊戲邏輯設計

主要類別：

Player (玩家飛船)
Rock (隕石)
Bullet (玩家子彈)
EnemyBullet (Boss 子彈)
Boss (Boss 敵人)
Explosion (爆炸動畫)
Power (道具系統)

核心功能：

Sprite 群組管理
圓形碰撞偵測
排行榜系統
Boss 戰機制
道具掉落系統
無敵時間控制
血量與生命值
系統爆炸動畫與音效



遊戲規則與特色

初始值：

血量 100

生命數 3

隕石機制：

初始 8 顆隕石

擊倒boss前每增加 500 分隨機永久增加 1 ~ 5 顆隕石

擊倒boss後每增加 500 分隨機永久增加 1 ~ 10 顆隕石

道具系統：

護盾 (恢復 20 點血量)

閃電 (子彈數量變2發持續5秒)

無敵 (3 秒無敵、10000 分後出現)

Boss 戰當分數超過 20000 分：

攻擊有機率掉落道具

Boss 血量 500

發射扇形彈幕攻擊、擊敗 Boss 可獲得額外 10000 分



製作挑戰與未來展望

在《太空生存戰》的開發過程中，主要挑戰來自於遊戲平衡性設計與系統整合。包含 Boss 的攻擊模式設計、血量與彈幕機制、隕石生成與難度成長控制，以及排行榜資料的讀寫與排序，都需要反覆測試與調整，以確保遊戲能順暢運作並具有挑戰性。實際測試後，目前玩家最高分約可達 40,000 分，因此未來希望進一步優化遊戲平衡，使後期內容更具延展性與挑戰性。

未來規劃：

- 在 40,000 分後新增第二階段 Boss 挑戰，提升後期難度
 - 增加恢復生命的道具，使玩家具備更多生存策略
 - 建立遊戲商城系統，玩家可透過遊戲獲得金幣並進行消費
 - 加入造型與外觀系統，提升個人化與收集要素
- 期望透過後續優化，讓遊戲不僅具備挑戰性，也能增加長期可玩性與趣味性。



遊戲畫面

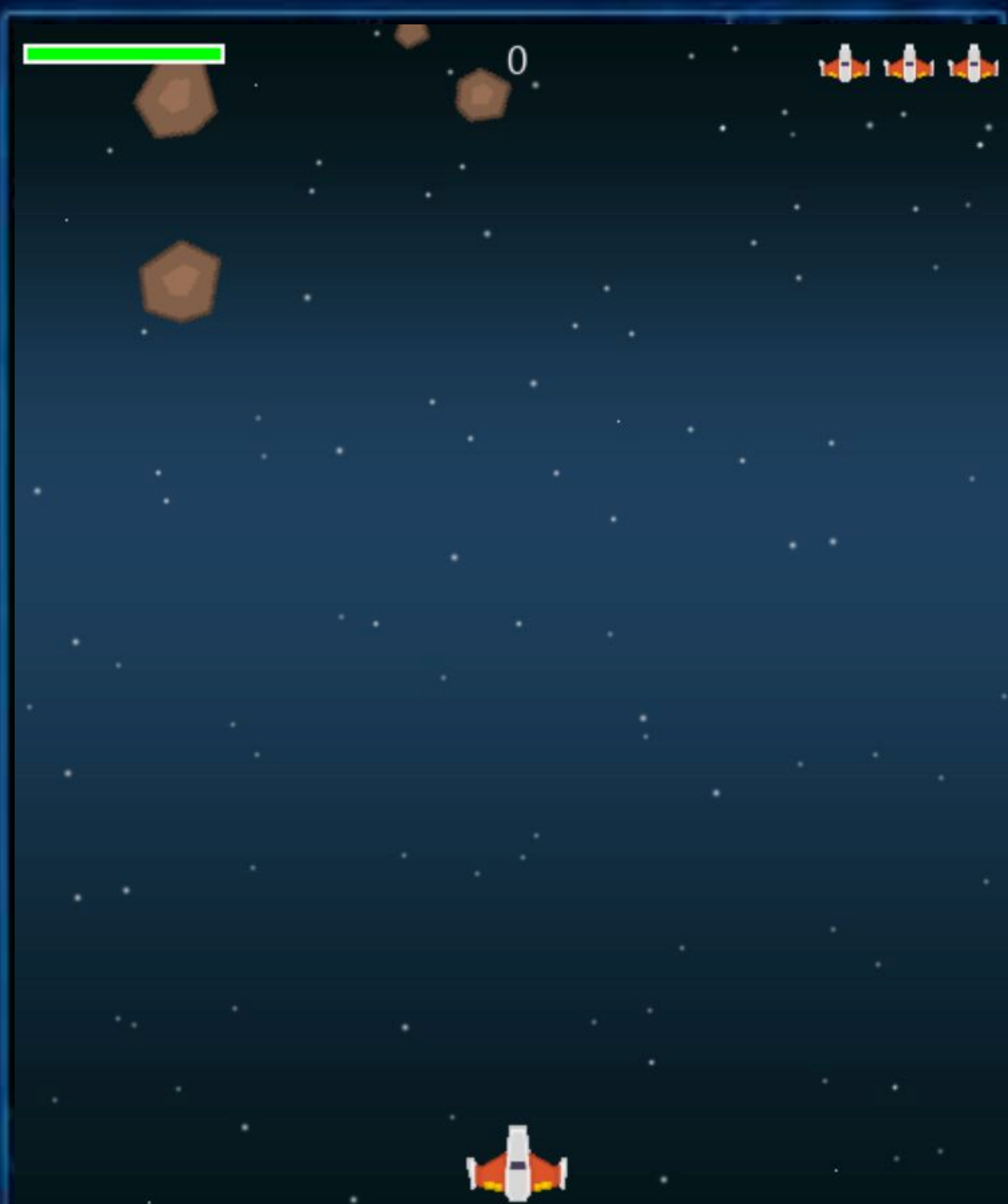
太空生存戰

← →移動飛船 空白鍵發射子彈~

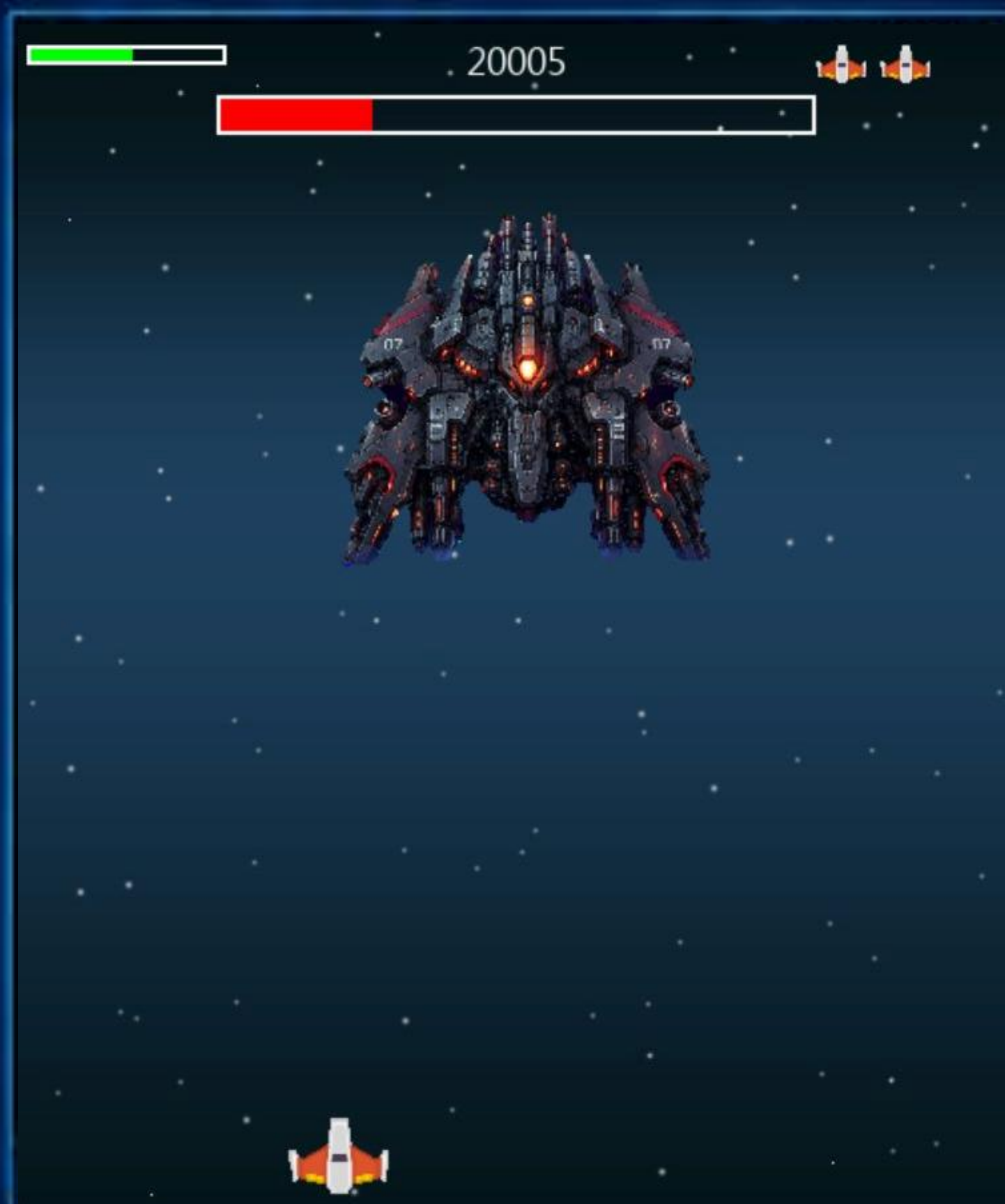
按鍵開始遊戲



遊戲開始畫面



遊戲進行畫面



Boss 戰畫面

名次	名字	分數
1	haha	39857
2	newtest	20043
3	o	20011
4	bbb	14933
5	b	10569
6	hhhhhhh	164
7	True	73
8	teat1	10
9	teat1	9
10	teat1	8

排行榜畫面

智慧科技學院程式設計專題海報

賭場模擬器

簡介：

我們結合事物的隨機性及大量的邏輯判斷，將 Python 基礎語法轉化為一個具備互動性的純文字賭場體驗，展現程式設計的邏輯與實際運用上的實用性！

```
=====
歡迎來到 義大賭場
=====
目前餘額：$1000
1. 21點
2. 擲骰子比大小
3. 猜大小
4. 不玩了！帶著錢走人（離開遊戲）
=====
請輸入指令（1-4）：
```

↑初始大廳:

可以輸入指定數字遊玩對應遊戲

```
=====
歡迎來到 押大押小 賭桌
=====
請輸入下注金額（目前餘額 $1000）:1000
請押注（輸入 B 代表大 / S 代表小）: b
已下注 $1000 押【大】，莊家搖骰盅中...

骰盅開啓：[6] [6] [2]
總共是14點！
結果是：【大】

===== 結算 =====
恭喜你猜對了！贏得 $1000！
```

↑遊戲3-押大押小實際遊玩畫面

```
=====
歡迎來到 幸運骰子桌
=====
請輸入下注金額（目前餘額 $1000）:1000
已下注 $1000，準備擲骰子...

輪到你擲骰子了...
你擲出了 [4] 和 [3]，總共是 7 點！

換莊家擲骰子...
莊家擲出了 [6] 和 [5]，總共是 11 點！

===== 結算 =====
莊家點數較大，你輸了這局。
```

↑遊戲2-骰子比大小實際遊玩畫面

```
=====
歡迎來到 21 點賭桌
=====
請輸入下注金額（目前餘額 $1000）:1000
已下注 $1000，發牌中...

你的手牌:[10, 1]，目前點數:11
莊家明牌:8，目前點數:?
要再抽一張牌嗎？（y/加牌 / n/停牌）y

你抽到了:1
你的手牌:[10, 1, 1]，目前點數:12
莊家明牌:8，目前點數:?
要再抽一張牌嗎？（y/加牌 / n/停牌）y
```

```
你抽到了:4
你的手牌:[10, 1, 1, 4]，目前點數:16
莊家明牌:8，目前點數:?
要再抽一張牌嗎？（y/加牌 / n/停牌）y
```

```
你抽到了:2
你的手牌:[10, 1, 1, 4, 2]，目前點數:18
莊家明牌:8，目前點數:?
要再抽一張牌嗎？（y/加牌 / n/停牌）y
```

```
你抽到了:9
你的手牌:[10, 1, 1, 4, 2, 9]，目前點數:27
莊家明牌:8，目前點數:?
你的點數已超過 21 點了，你輸了。
```

↑遊戲1-21點實際遊玩畫面

電機系

執導老師:黃榮生

學生:11401021A周呈峻

11401026A王脩元

PYTHON 實作 | 記帳系統

電機一 11401027A 趙晟竹

指導教授 洪湘欽老師

桌面應用 · CSV 儲存 · 圖表分析

專案概述

Project Overview

本專案以 Python 建立桌面端記帳系統，使用者可透過圖形化介面新增日常支出、即時查看消費明細、監控預算並產生類別圓餅圖。資料以 CSV 儲存在本機，兼具輕量、穩定、可攜，以及後續用 Excel 進行二次資料分析的便利性。

輕量化資料儲存 CSV 本機檔案

直覺式操作 Tkinter 圖形介面

可視化回饋 圓餅圖與提示訊息

核心功能

Core Features

圖形化介面

智能日期輸入

動態資料表

資料持久化

圖表視覺化

預算監控

精準刪除

DEBUG 防呆設計

Input Validation & Data Consistency

問題辨識

原系統若在金額欄位空白、輸入國字、英文字母或其他符號時，可能導致程式崩潰或寫入不完整資料。圖表分析也需要避免使用者自創類別造成分類混亂。

金額驗證

空白或非數字時立即跳出提醒，避免程式中斷。

類別標準化

支出類別限定為餐飲、交通、學習、娛樂、其他。

操作回饋

預算檢查與刪除成功視窗，提升使用者確認感。

CODE 開發流程

從資料讀取、資料新增、圖表分析到介面事件綁定

匯入模組

新增紀錄

預算/圖表

介面配置

資料表格

主程式

01 匯入模組與 CSV 初始化

02 新增資料與金額驗證

03 類別圖表與預算檢查

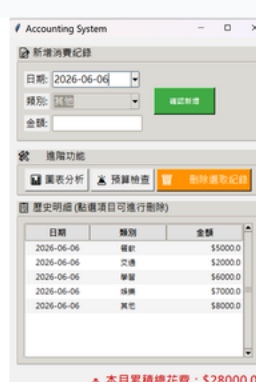
04 主視窗與欄位配置

05 Treeview 動態資料表

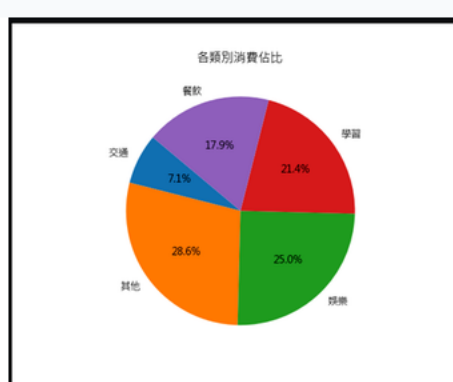
06 事件綁定與程式啟動

執行結果

Run Results



主畫面與消費明細



類別消費圓餅圖

預算檢查

☑ 狀態良好！還有 \$30000.0 可以花。

確定

預算檢查提醒

刪除成功

已刪除：2026-06-06 - 其他 (\$8000.0)

確定

刪除成功回饋

手機控制 Pac-Man 遊戲系統

學生：許祐晟、高浩程

指導教授：陳延華

研究動機

結合經典 Pac-Man 與手機控制技術，透過 QR Code 連線，使玩家能直接以手機作為遊戲搖桿，提升互動體驗。

系統特色

- QR Code 快速連線
- Flask 網頁控制器
- Tkinter 遊戲介面
- 幽靈 AI 與碰撞偵測
- 分數與道具系統
- 手機遠端操控

未來展望

- 多人連線
- 排行榜系統
- AI 追蹤幽靈
- 線上對戰模式

系統流程

手機掃描 QR Code



開啟網頁控制器



傳送方向指令



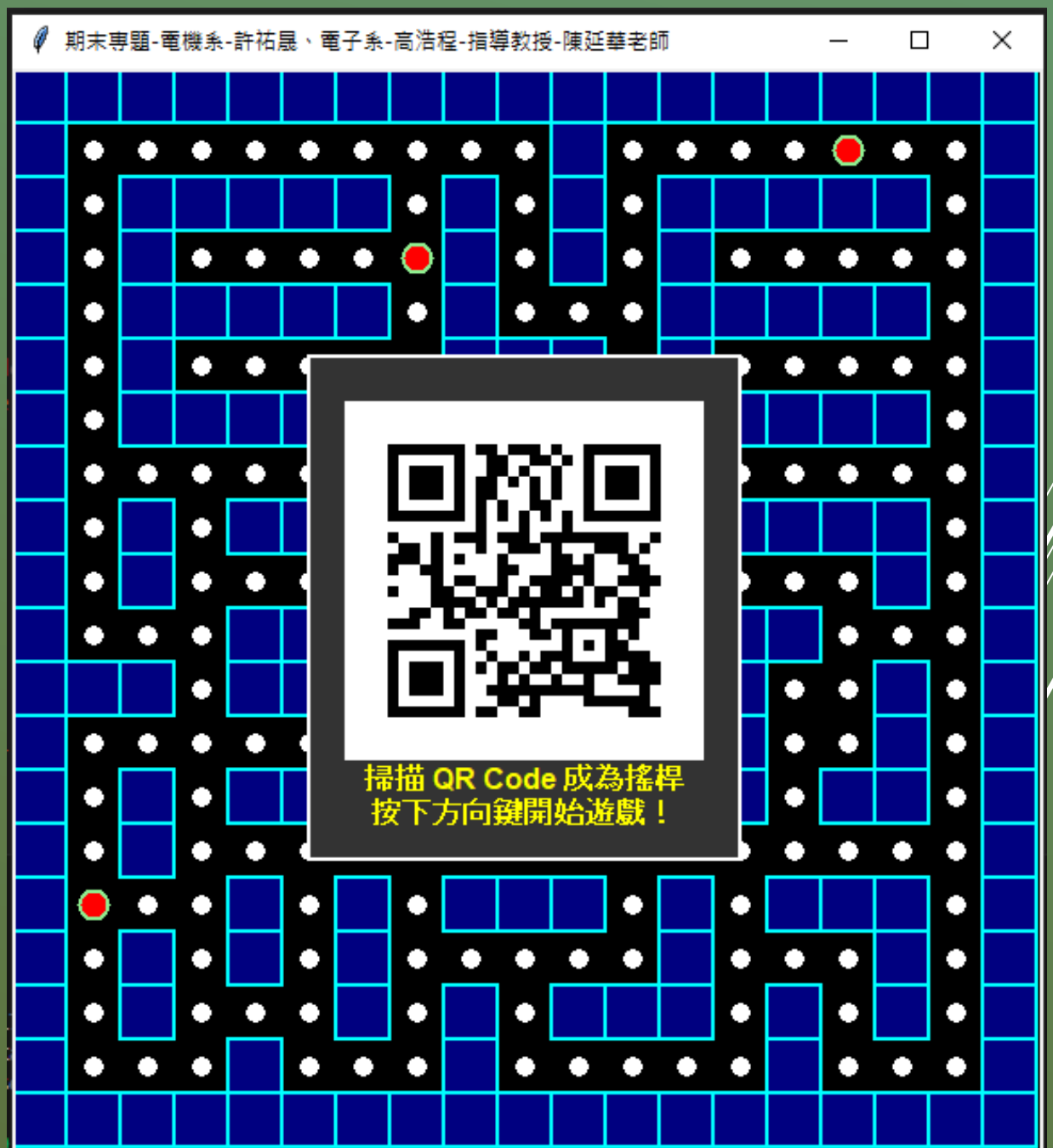
Flask 接收資料



Pac-Man 移動



分數更新與碰撞判斷



執行畫面

玩家掃描 QR Code 後即可利用手機控制 Pac-Man



智慧科技學院程式設計專題海報



全端 1A2B 猜數字大賽：

支援單機演算法與線上即時競技之遊戲平台

基於 Flask-SocketIO 與去中心化邊緣運算之混合式 Web 遊戲架構實作

組員: 鍾沛哲、李威潤

指導老師: 陳延華

1. 開發動機與背景 (Introduction)

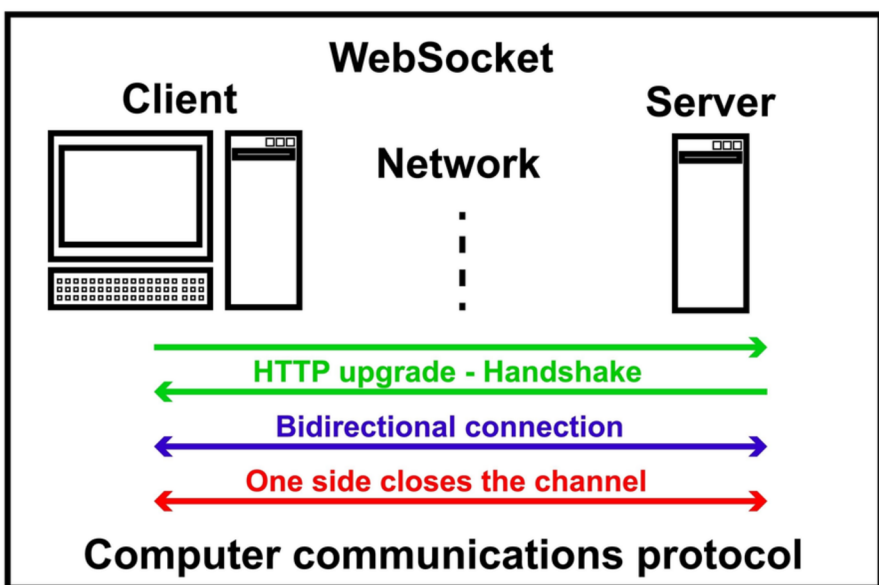
傳統網頁多採用「請求/回應 (Request/Response)」模型，面對「雙人連線、回合制」等高度即時互動的場景時，常面臨耗能、高延遲的技術痛點。本專題以經典邏輯遊戲「1A2B」為載體，探索現代網頁如何結合傳統 HTTP、WebSocket 以及邊緣運算，打造一套兼具高效能與資源最佳化的混合式網路對戰平台。

2. 系統核心三大亮點 (Core Highlights)

- 混合式通訊策略：靜態資源走 HTTP，連線對戰走 WebSocket。
- 邊緣運算思維：單人模式 100% 下放前端瀏覽器，伺服器算力零消耗。
- 高效廣播機制：放棄耗能的傳統輪詢 (Polling)，採用事件驅動 (Event-driven) 精準推送封包。

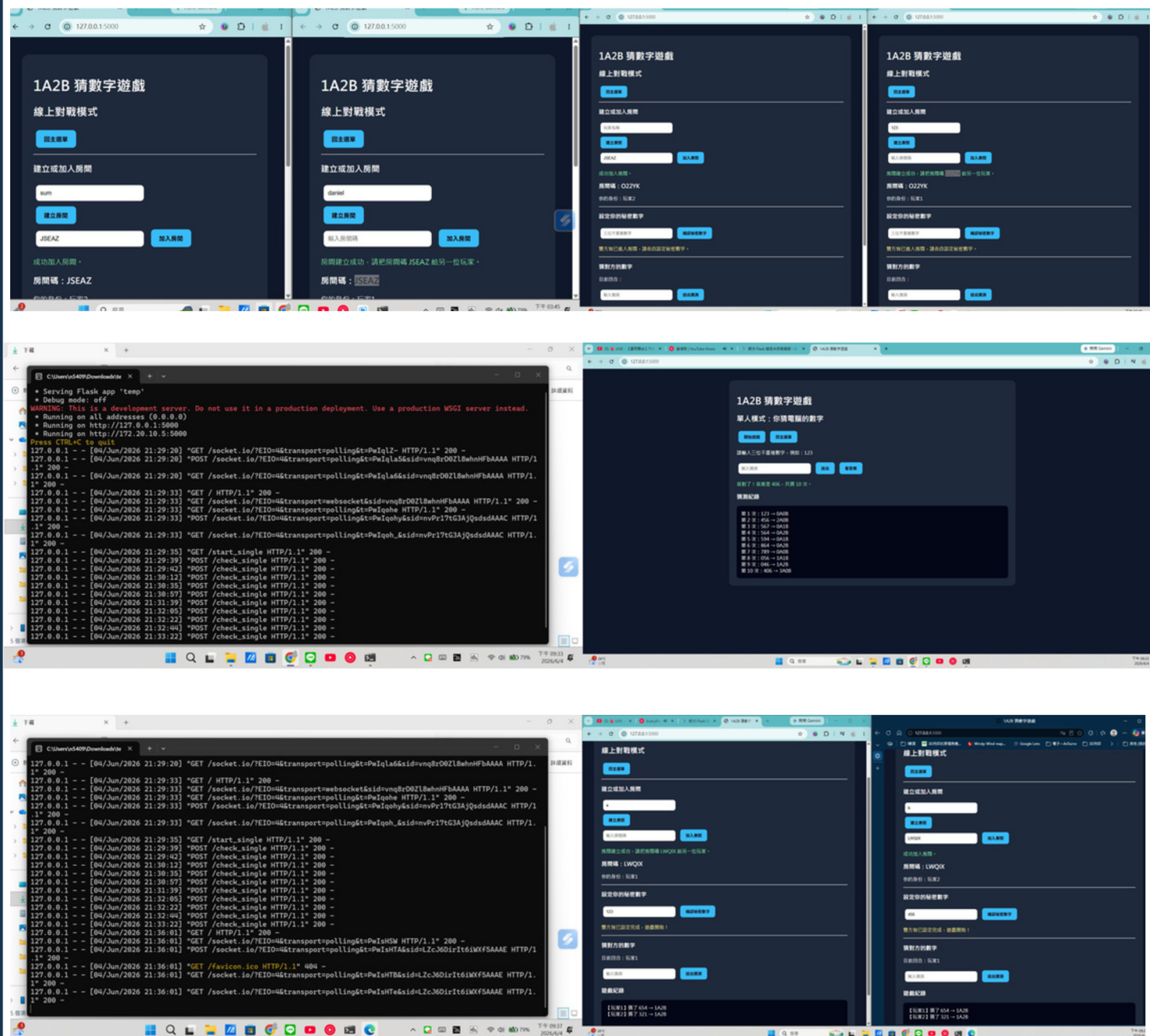
5. 系統網路通訊架構圖 (System Architecture)

(建議在這裡放置一張架構圖，呈現瀏覽器與後端 Flask-SocketIO 的關係)



網頁初始化時，瀏覽器向後端請求靜態資源。一旦進入對戰模式，前端透過握手 (Handshake) 升級為 WebSocket 全雙工 (Full-Duplex) 熱線，實現毫秒級無感延遲對戰。

8. 成果展示



3. 單機模式：去中心化演算 (Single Player Mode)

單人對戰不需要經過伺服器，利用用戶端 JavaScript 於本地端進行邊緣運算：

- 不重複驗證：利用數學「集合 (Set)」不重複特性，一行代碼 `new Set(n).size === 3` 完成防呆。
- 核心演算法：透過陣列高效過濾器 (`.filter()`) 動態比對答案：
 - A 判定：位置與數字皆符合。
 - B 判定：數字存在但位置錯誤。

4. 雙人線上對戰：Socket 通訊流程 (Real-time Match)

利用事件驅動機制，後端精準建立虛擬房間，通訊流程如下：

- `create_room / join_room`：隨機生成 5 碼大寫英數字房間代碼，並利用 Socket 房間機制隔離不同戰局。
- `set_secret`：雙方在不同客戶端設定秘密數字，後端進行就緒狀態偵測。
- `guess_online`：回合制交互攻守。後端作為「中立裁判」，比對答案後，將幾 A 幾 B 獨立推送至該房間所有成員的螢幕上。

6. 巢狀字典狀態機 (State Machine)

後端利用 Python 內建的高效巢狀字典 (Nested Dictionary) 動態管理多執行緒下的房間狀態，並在接收到請求時進行嚴格的安全性校驗：

- 狀態流轉：Waiting $\rightarrow$ Setting_Secret $\rightarrow$ Playing $\rightarrow$ Ended。
- 回合驗證：嚴格限制「非當前回合玩家」的惡意呼叫，防範前端透過控制台作弊，確保競賽公平性。

7. 專案開發效益與價值 (Benefits)

- 效能優化：將單人運算移至前端，使伺服器在面對萬人同時在線單機遊玩時，CPU 負載依然為 0%。
- 低開銷通訊：捨棄高頻率 HTTP 請求，大幅度節省伺服器網路頻寬 (Bandwidth) 與 I/O 開銷。
- 全端實務技能：完整實踐網頁前後端邏輯分離、即時通訊協定應用以及複雜狀態控制。

9. 未來展望 (Future Work)

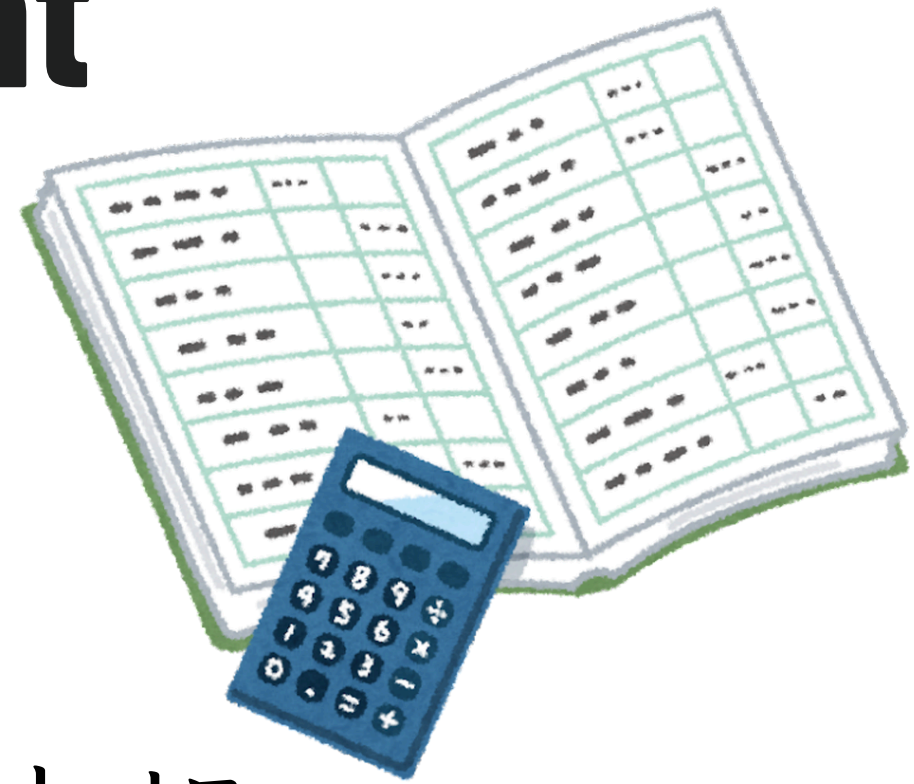
- 資料庫持久化：預計導入 PostgreSQL 與 Redis，建立會員勝率統計與「全球即時排行榜」。
- 動亂數媒合：開發後端佇列 (Queue) 演算法，讓線上玩家能一鍵自動配對。
- 互動式多媒體：利用既有 Socket 管道擴充「即時戰術語音/心理戰嘲諷貼圖」功能，提升遊戲趣味性。

成果心得:

本專題成功結合前端邊緣運算與後端 WebSocket，打造出高效能的全端對戰平台。單人模式 100% 下放瀏覽器 (JS) 執行，達成「零延遲」且伺服器算力零消耗；雙人連線則利用 Flask-SocketIO 建立雙向即時熱線，實現毫秒級無感延遲對戰。後端更導入嚴格的房間狀態機管理 (Waiting $\rightarrow$ Playing)，有效防範越權作弊，全面掌握了從「資源分配」視角優化全端網路服務的核心能力。

AI 記帳分析助手

AI Expense Analysis Assistant

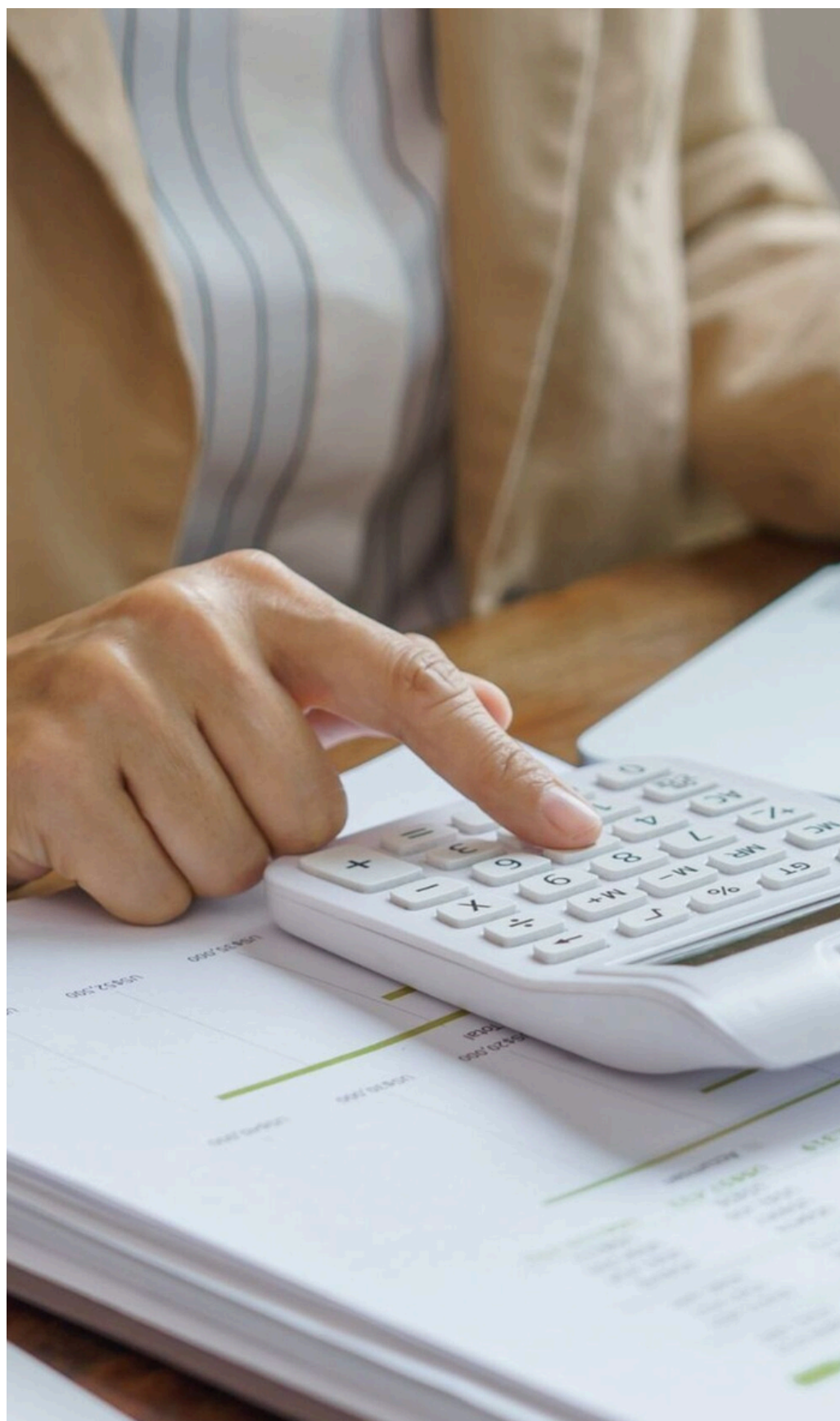


題目出處

本題目為自行發想Python 期末報告，結合記帳系統、資料分析與簡易AI 自動分類功能進行開發。

謝博宇

2026 6/1



題意說明

現代人每日消費頻繁且迅速，但往往無法清楚掌握自己的花費流向，一瞬間想購買的衝動 導致不可挽回的結果，因此我希望設計一套簡易 AI 記帳分析系統，協助使用者快速記錄支出並進行消費分析。

此系統透過 Python 開發，結合：

- GUI 圖形介面
- SQLite 資料庫
- Pandas 資料分析
- Matplotlib 圖表分析

實作一套具備：

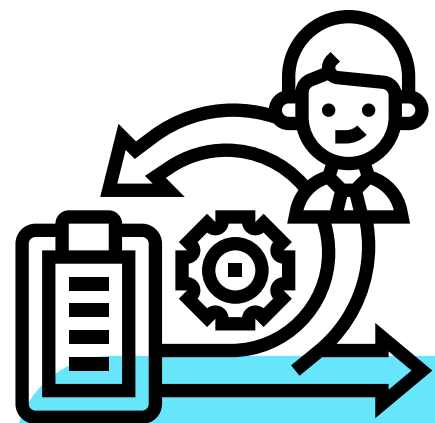
- 自動分類
- 統計總支出
- 消費分析
- 圖表視覺化
- 功能的智慧記帳系統。

(bout Artificial Intelligence)

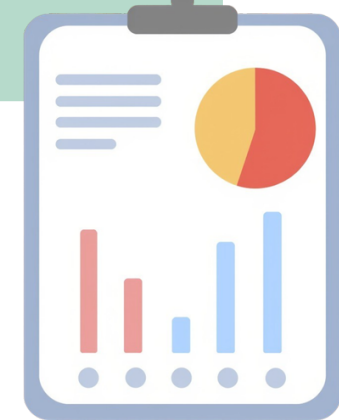
輸入消費資料



系統讀取資料



AI關鍵字分類



解題方式與內容

存入
SQLite
資料庫

Pandas
進行統計
分析

Matplotlib
產生圖表

finally
顯示分析
結果



完成分類後，系統會將資料存入 SQLite 資料庫。
之後使用 Pandas 進行資料分析與統計，再透過 Matplotlib 產生圓餅圖，顯示各類別消費比例。

tkinter

01 Python 內建的 GUI（圖形化介面）套件
功能：

建立 GUI 圖形介面。

用途：

- 視窗
- 按鈕
- 輸入框
- 標籤

負責

- 「新增記帳」按鈕
- 輸入項目框
- 金額輸入框
- 顯示總支出

sqlite3

02 Python 內建的輕量型資料庫（Database）
功能：

程式「永久保存資料」

建立本地端資料庫。

用途：

儲存使用者記帳資料

```
conn = sqlite3.connect("money.db")
cursor = conn.cursor()
```

cursor 「資料庫操作工具」

cursor 用來執行 SQL 指令，
進行資料庫操作。

conn connection（連線）

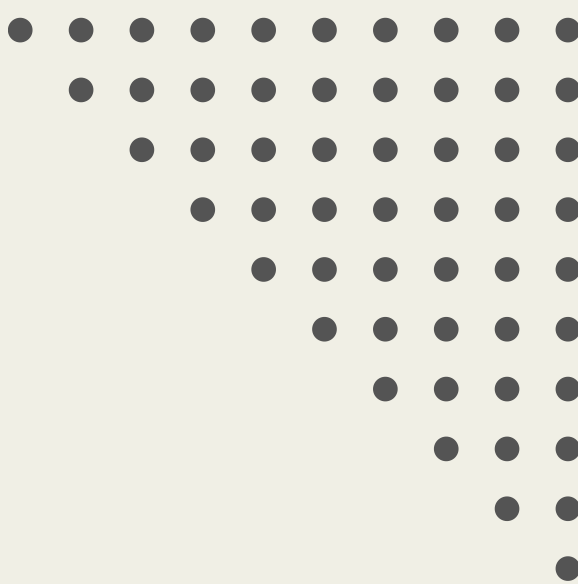
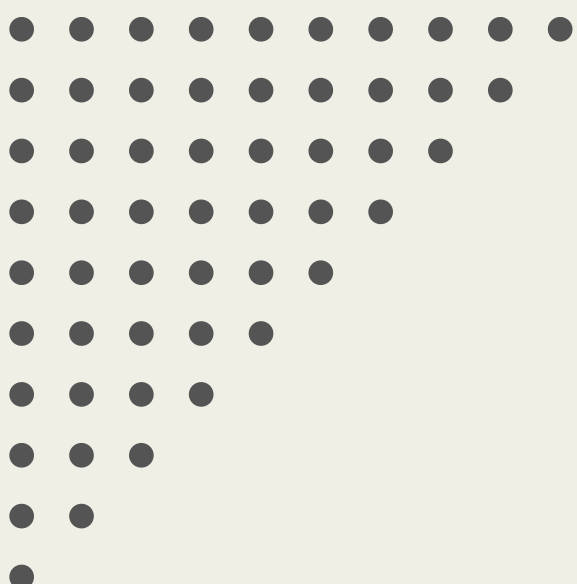
用來建立 SQLite 資料庫
並建立程式與資料庫的連線

題目出處：UVA題庫-11007

Mini Cube

學號：11403130A 姓名：楊巧雯

報告日期：6/1



題目說明

給 $2 \times 2 \times 2$ 魔術方塊的初始狀態，請求出將它還原所需的最少四分之一轉次數（即某一面旋轉 90 度算一次）。你可以假設輸入的初始狀態一定合法，因此一定存在解。當魔術方塊被還原時，顏色會像下列敘述一樣：白色（‘W’）與藍色（‘B’）相對，紅色（‘R’）與橘色（‘O’）相對，黃色（‘Y’）與綠色（‘G’）相對。

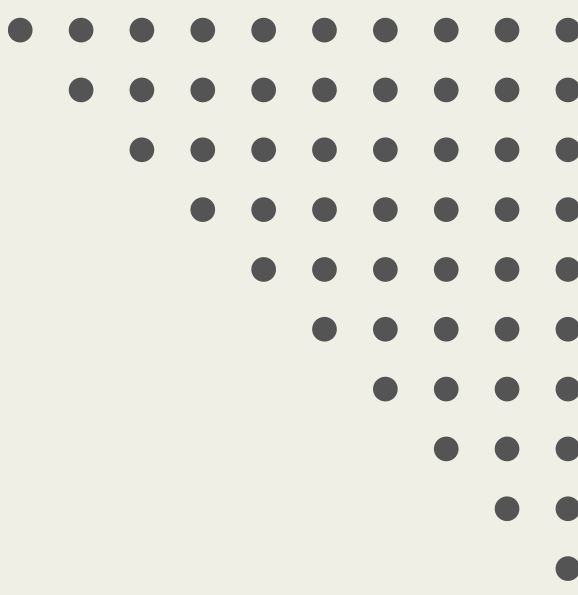
題目說明

輸入**第一行包含一個整數**，表示有多少個魔術方塊
需要你的程式求解（最多 100 個）。接下來每個
魔術方塊會用**6 行描述**，依序代表以下六個面：

上（T）、前（F）、右（R）、下（b）、後
（B）以及左（L）。

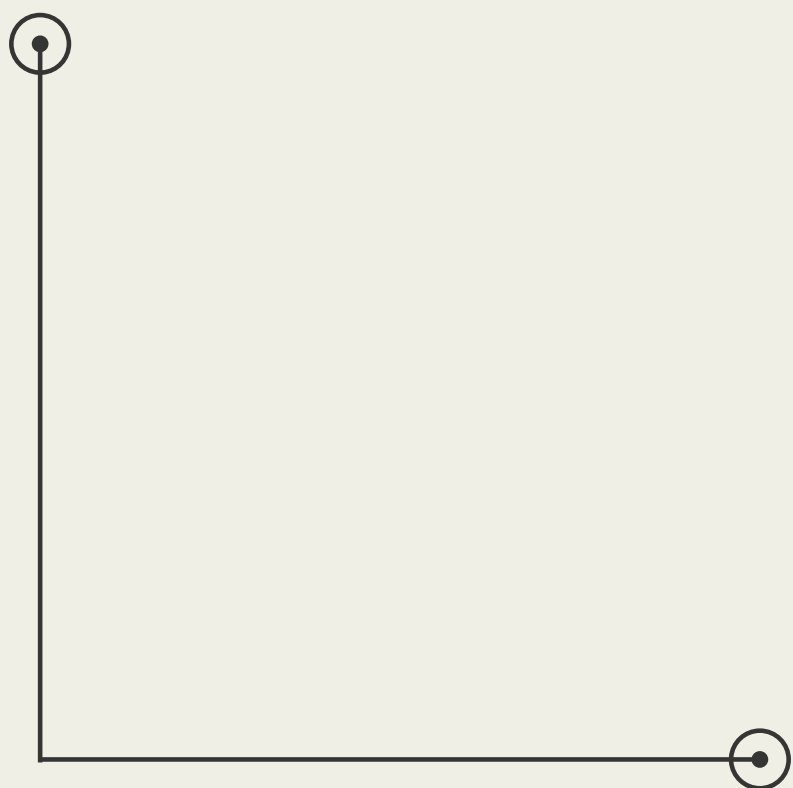
每個面會用**4 個字元描述其顏色**，排列順序如右
側展開圖所示。每組測資前面都會有一個空白
行。

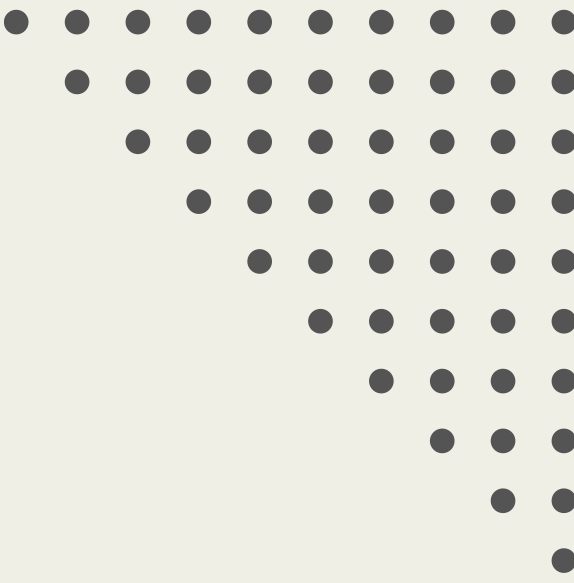
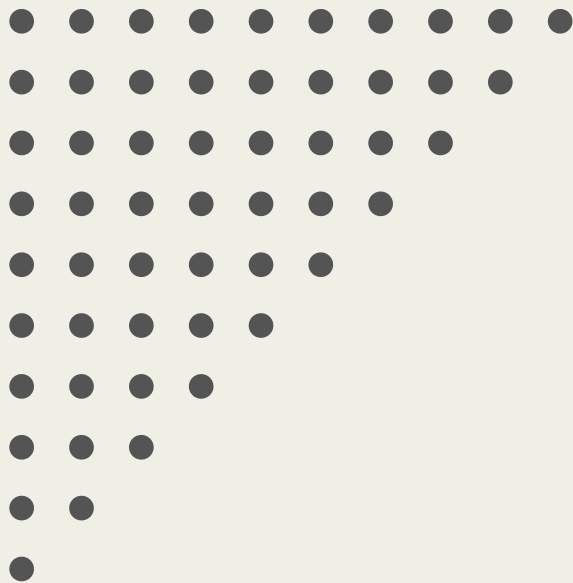
		B4	B3		
		B2	B1		
L3	L1	T1	T2	R2	R4
L4	L2	T3	T4	R1	R3
		F1	F2		
		F3	F4		
		b1	b2		
		b3	b4		



題目說明

對於每個魔術方塊，輸出一行，表示將它還原所需的
最少四分之一轉次數。





題目說明

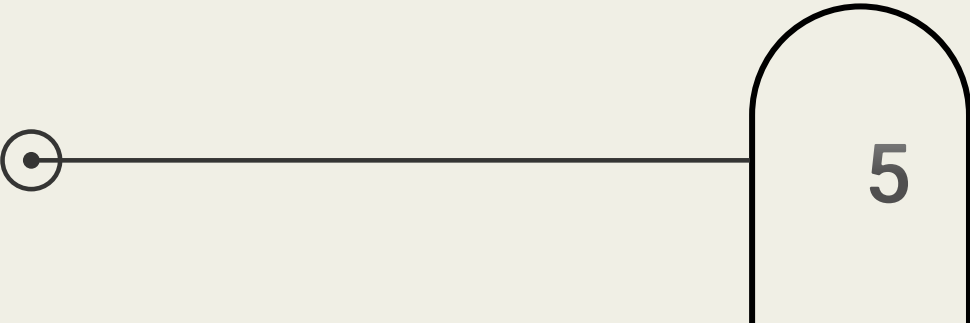
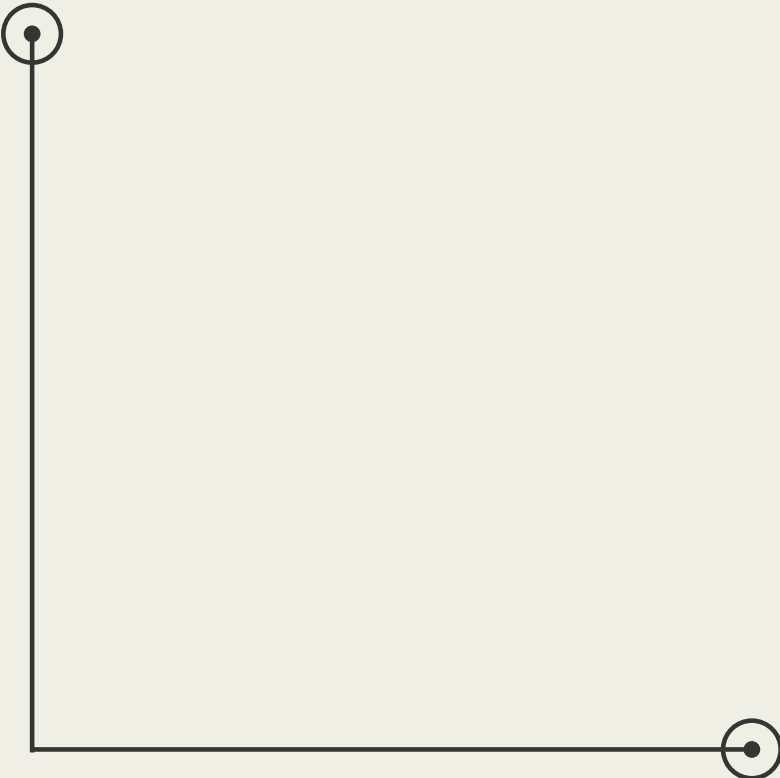
輸入:

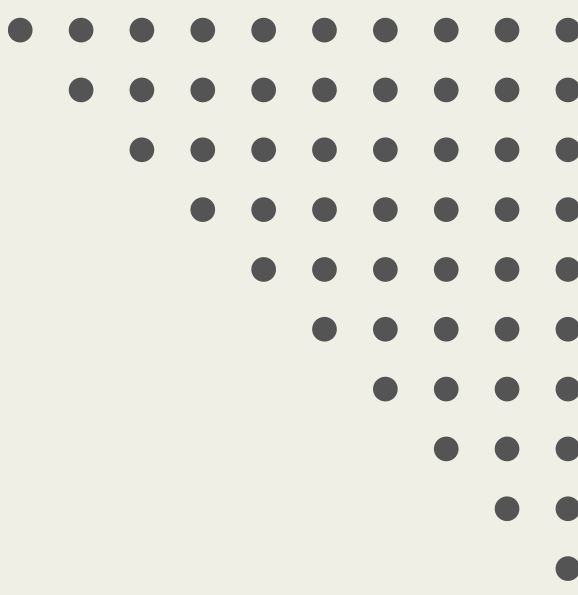
2
GGGG
WWBB
OORR
YYYY
BBWW
RROO

輸出:

2
6

WRRG
WORO
BGBW
YYOG
BORY
GYWB

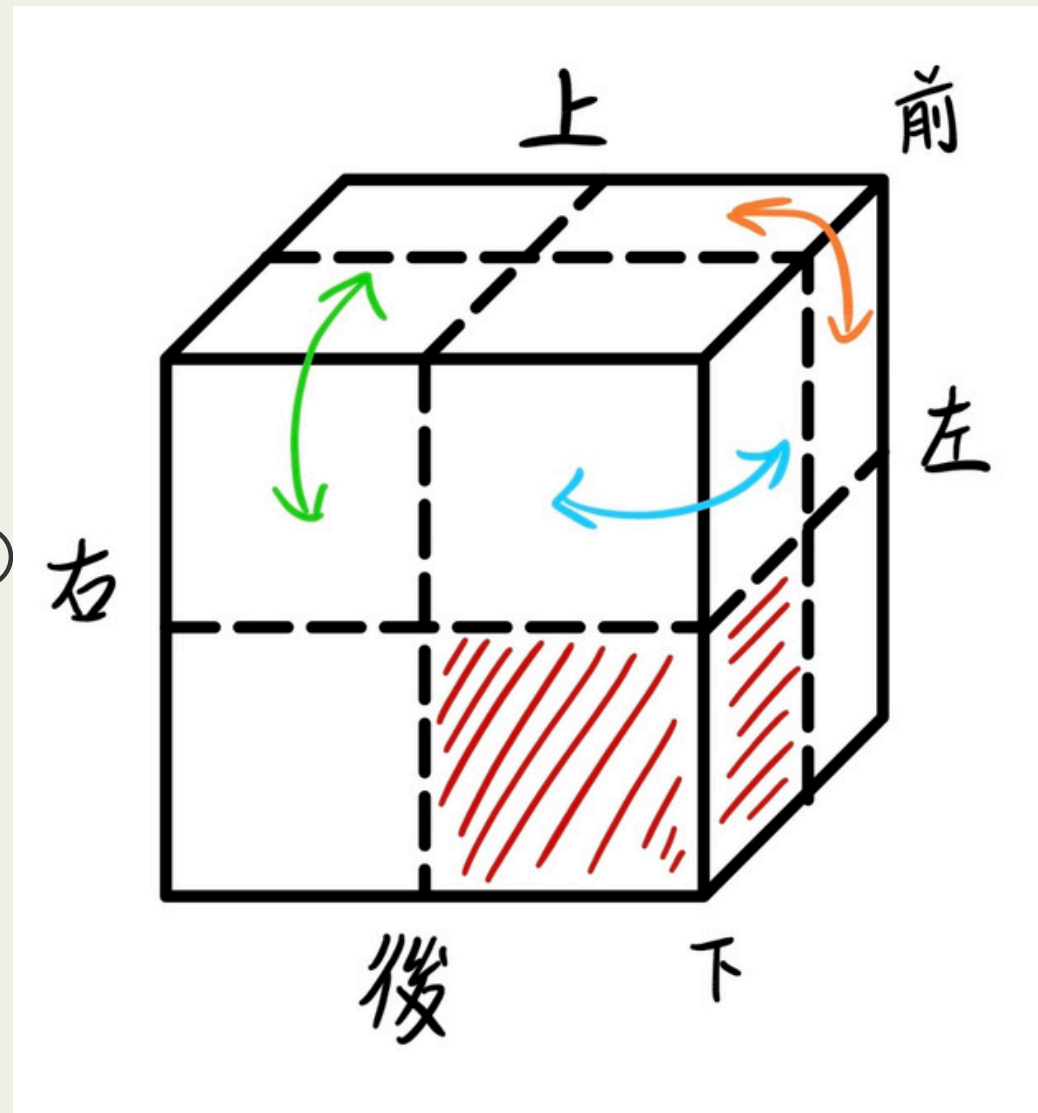




解題方法

1. 先找出一角做為基準
2. 制定一個旋轉規則
3. 利用雙向BFS去找答案

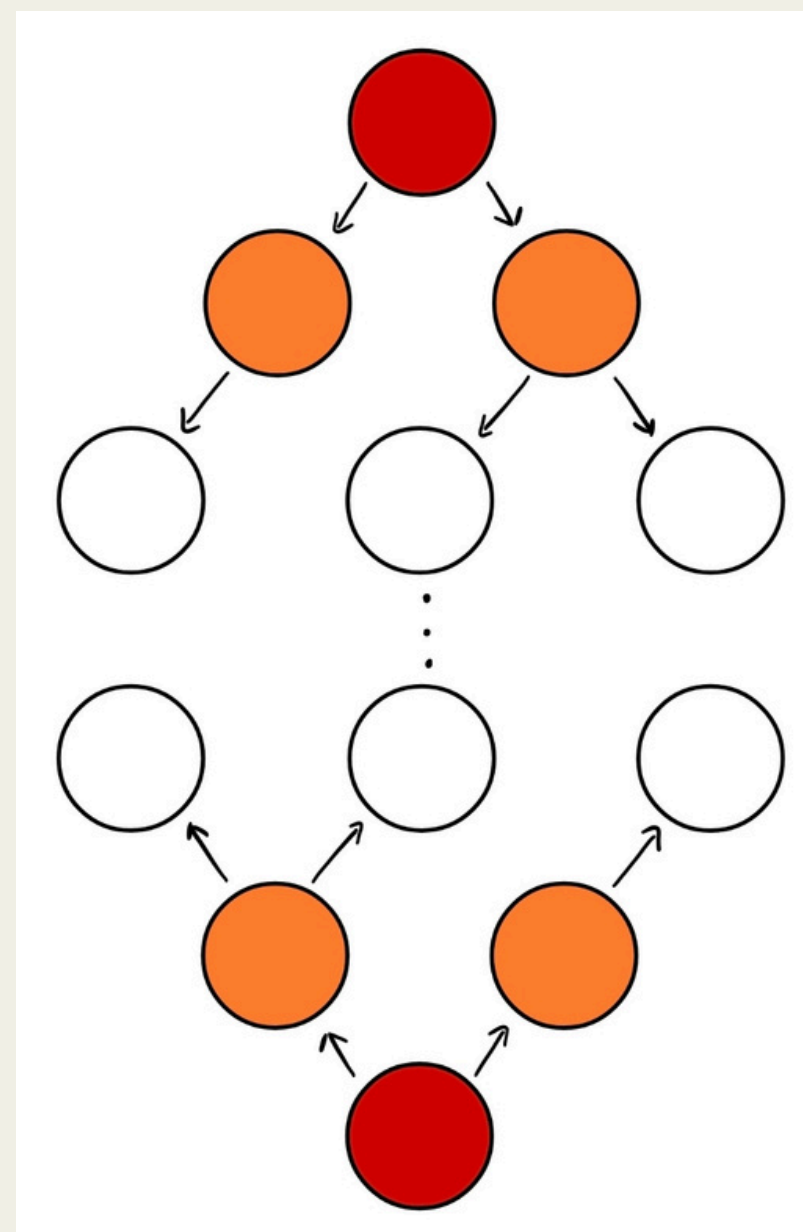
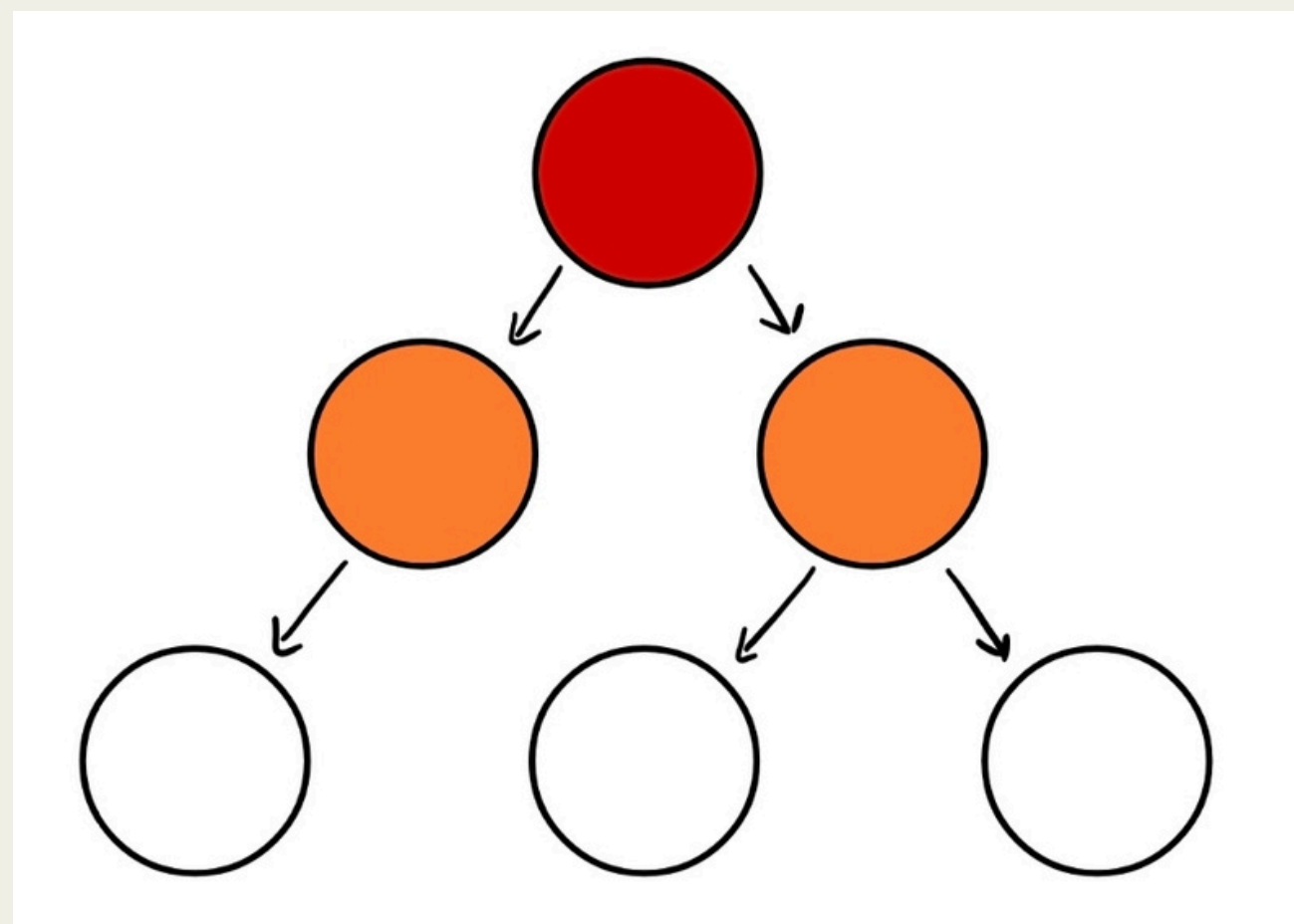
解題方法



		19	18		
		17	16		
22	20	0	1	9	11
23	21	2	3	8	10
		4	5		
		6	7		
		12	13		
		14	15		

		17	16		
20	0	1	9		
21	2	3	8		
	4	5			
		21	20		
4	2	0	17		
5	3	1	16		
	8	9			
		9	8		
		16	17		
		1	3	5	
		0	2	4	
		20	21		

解題方法



☀️ Python 即時天氣查詢與未來氣象提醒系統 ☀️

內容

本專題使用 Python 串接中央氣象署 OpenData API，建立一個即時天氣查詢系統。使用者可透過圖形介面選擇台灣不同縣市，系統會即時取得當地天氣資料，包含溫度、濕度與天氣狀態，並以圖示化方式呈現（☀️☔️☁️）。此外，系統會提供未來一小時的天氣預測與生活建議，提升使用者對天氣變化的掌握能力。

重點程式碼截圖

```
data = json.loads(response.read().decode("utf-8"))
stations = data["records"]["Station"]
```

API 串接：串接中央氣象署 OpenData API，取得並解析即時天氣資料。

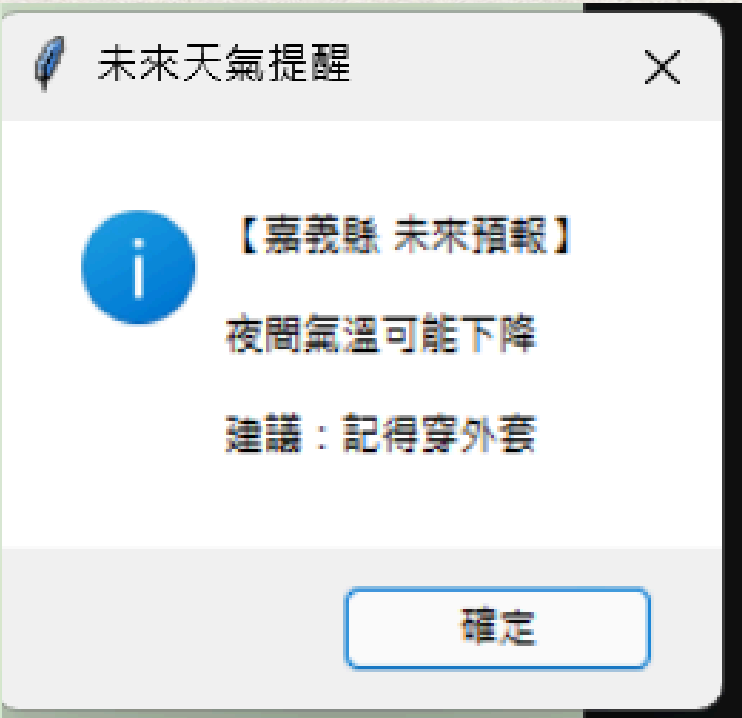
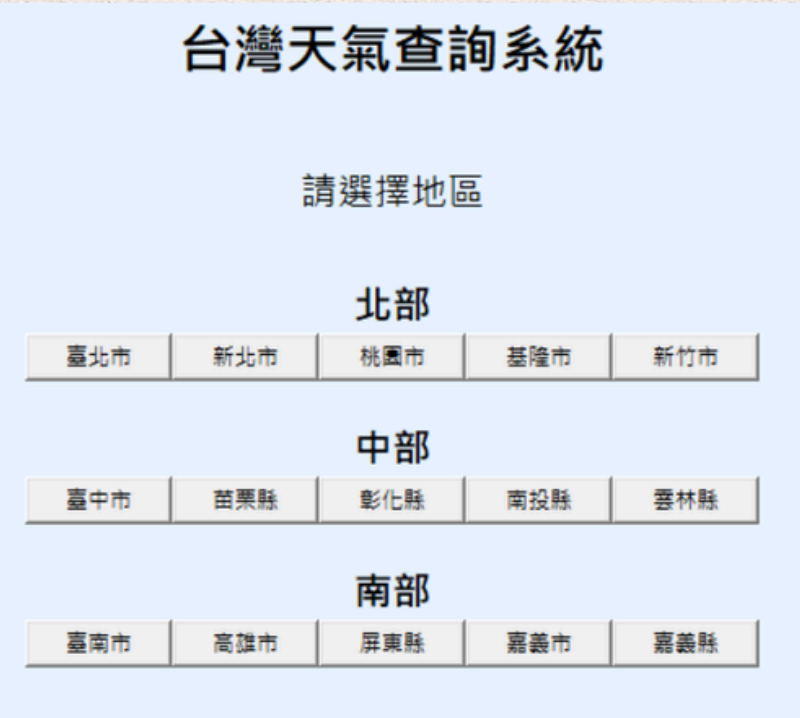
```
if "雨" in weather:
    icon = "☔️"
    reminder = "目前有降雨"
    forecast = "未來一小時可能持續降雨"
    advice = "建議攜帶雨傘"
```

天氣判斷與圖示化：依據天氣條件進行判斷分析，並以圖示化方式呈現天氣狀態。

```
for city in ["臺北市", "新北市", "桃園市", "基隆市", "新竹市"]:
    tk.Button(north_frame, text=city, width=10,
              command=lambda c=city: get_weather(c)).pack(side="left")
```

GUI 按鈕互動：建立互動式圖形介面，讓使用者透過按鈕查詢各縣市天氣資訊。

程式精采畫面截圖



404_NotFound 踩地雷

義守大學 智慧科技學院 資訊工程學系

成員：鄭艾可、李京穎、趙俞晴 指導老師：蔡賢亮

遊戲介紹

本專案是一個抽象的程式碼轉化為互動的遊戲實踐，利用數字線索展開關於邏輯推導與運氣的較量，也是童年常玩的遊戲。這次做踩地雷遊戲，我後台是用二維矩陣來做網格的座標管理，並用隨機演算法讓每場地雷隨機生成。其中寫出遞迴擴充機制，利用洪水填充演算法，讓玩家點到 0 的時候可以自動用遞迴掃描周圍 8 個格子、一次翻開大片空白的部分讓我們研究了一段時間。

畫面部分我們研究了 Tkinter 介面的事件驅動架構，去綁定滑鼠左右鍵的點擊和插旗動作，並動態去刷勝負條件，因為這全部都是用 Python 內建標準庫寫的，沒有用其他外掛，讓我在精簡的架構下把資料結構和邏輯學以致用，真的非常有成就感。

程式碼

```
game.py
class Minesweeper:
    def setup_board(self):
        self.board[r][c] = -1
        mines_placed += 1

        for r in range(self.rows):
            for c in range(self.cols):
                if self.board[r][c] == -1:
                    continue
                count = 0
                for dr in [-1, 0, 1]:
                    for dc in [-1, 0, 1]:
                        nr, nc = r + dr, c + dc
                        if 0 <= nr < self.rows and 0 <= nc < self.cols:
                            if self.board[nr][nc] == -1:
                                count += 1
                self.board[r][c] = count

    def create_widgets(self):
        for r in range(self.rows):
            row_btns = []
            for c in range(self.cols):
                btn = tk.Button(self.root, width=3, height=1, font=('Arial', 12, 'bold'),
                                relief="raised", bg="#D3D3D3")

                btn.bind('<Button-1>', lambda event, r=r, c=c: self.left_click(r, c))
                btn.bind('<Button-3>', lambda event, r=r, c=c: self.right_click(r, c))
                btn.grid(row=r, column=c)
                row_btns.append(btn)
            self.buttons.append(row_btns)

    def right_click(self, r, c):
        self.flags.add((r, c))
        self.buttons[r][c].config(text="🚩", fg="red")

    def reveal(self, r, c):
        if (r, c) in self.revealed:
            return
        self.revealed.add((r, c))
        val = self.board[r][c]

        colors = {1: "blue", 2: "green", 3: "red", 4: "darkblue", 5: "maroon", 6: "cyan", 7: "black"}

        if val > 0:
            self.buttons[r][c].config(text=str(val), bg="white", relief="sunken", fg=colors.get(val))
        else:
            self.buttons[r][c].config(bg="white", relief="sunken")

        for dr in [-1, 0, 1]:
            for dc in [-1, 0, 1]:
                nr, nc = r + dr, c + dc
                if 0 <= nr < self.rows and 0 <= nc < self.cols:
                    self.reveal(nr, nc)

    def show_all_mines(self):
        for r in range(self.rows):
            for c in range(self.cols):
```

第一步：鋪設地圖與算雷

(setup_board & create_widgets)

程式先在畫面上畫出一格一格的灰色按鈕。

接著偷偷在背景隨機放好地雷，並讓每個格子去數自己周圍 8 個鄰居有幾顆雷，把數字算好藏在裡面。

第二步：玩家操作與反應

(left_click & right_click)

點右鍵：玩家覺得這裡有危險，程式就在格子上插一隻紅旗。

點左鍵：玩家要翻開格子，程式會把按鈕按下去，秀出底下的數字和顏色。

第三步：自動爆開與連鎖反應如果玩家運氣好，踩到周圍完全沒有雷的「0」，程式就會啟動連鎖反應，自動把周圍的格子一次全部爆開，直到遇到有數字的邊界才會停下來。

遊戲介面



• 初始化遊戲



• 點擊任一方塊開始遊戲



• 繼續點擊(邏輯推導)



• 踩到地雷-遊戲結束

智慧科技學院程式設計專題海報

題目：pygame 末日機器人遊戲

指導教授：蔡賢亮

學生：資工一 吳冠廷、資工一 鍾皓丞、資工一 鄭郁懷

簡介

本作是一款 2D Roguelite 類生存射擊遊戲。玩家將在機械失控的末日世界，擊敗源源不絕的敵人獲得獎勵來不斷強化自身。每次挑戰都會遇到隨機的敵人與還有BOSS。透過靈活的衝刺走位、多重卡牌技能組合，以及隨機取得的傳奇武器，在槍林彈雨中殺出重圍。



角色系統

角色具備生命、護盾、體力與能量四種能量基礎屬性。消耗體力進行高速閃避，盡可能的躲避敵人的攻擊。



卡牌技能系統

每次升級會觸發三選一的隨機卡牌強化。技能組可以隨你任意組合。也可查看目前已取得的卡牌能力。



地表探索

進入大地圖後戰鬥正式開始。敵人會持續，玩家需要擊敗敵人來拾取物資、經驗與物資，同時提升角色能力，並在最後尋找情報點逃脫。



敵人與BOSS

包含近戰、遠程與自爆攻擊方式。敵人強度會隨玩家等級與難度模式有所不同。共有三隻BOSS具有不同的特性給玩家挑戰。



地堡基地

是遊戲的安全區域能消耗廢鐵作為貨幣升級、管理與物品、強化或調整武器屬性、裝備或切換武器。



任務系統

進入遊戲時會隨機發布任務，在地堡時也持續提示，且完成任務會獲得不同獎勵，獎勵會直接發佈到背包。