組別:team16

# Nothing Can Go Wrong – 期末遊戲專案

本專案是一款 2D 闖關遊戲，主題「Nothing Can Go Wrong…」。玩家控制角色在一個 5×5 的地圖中移動，並依次挑戰兩個小遊戲關卡：

1. 貪食蛇 (Greedy Snake)

2. 1A2B 猜數字

只要任一關卡失敗，整個遊戲就結束；必須同時通過兩個關卡才算整個遊戲勝利。以下說明:

#開發與執行環境

作業系統：Windows 10 / Windows 11

編譯器：MSYS2 MinGW-w64（GCC 15.1.0）

視窗繪製：raylib (需安裝 `mingw-w64-x86\_64-raylib`)

語言：C (C11 標準)

編譯與執行:

## 安裝與編譯步驟

以下以 \*\*MSYS2 MinGW 64-bit\*\* 為例說明如何安裝必要套件並編譯：

1.安裝 MSYS2

下載並安裝 MSYS2：https://www.msys2.org/](https://www.msys2.org/

安裝路徑請選擇無空格、純 ASCII（例如 `C:\msys64`）。

2.更新 MSYS2 環境

開啟「MSYS2 MinGW 64-bit」終端機 (提示字首會顯示 `MINGW64`)。

執行：

pacman -Syu

若出現「退出重啟」提示，請按任意鍵關閉視窗，再重新開啟「MSYS2 MinGW 64-bit」。

再次執行：

pacman -Su

直到顯示「沒有可更新套件」。

3. \*\*安裝 MinGW-w64 工具鏈與 raylib\*\*

仍在 `MINGW64` 終端機下，執行：

pacman -S --needed base-devel mingw-w64-x86\_64-toolchain

pacman -S mingw-w64-x86\_64-raylib

`base-devel` 會安裝 `make`、`autoconf`、`binutils` 等工具。

`mingw-w64-x86\_64-toolchain` 包含 `gcc`、`g++`、`gdb` 等。

`mingw-w64-x86\_64-raylib` 會安裝 Raylib 庫與標頭。

最後在終端輸入: gcc -std=c11 -O2 main.c map\_ui.c snake.c one2b.c -o game.exe -lraylib -lopengl32 -lgdi32 -lwinmm

-std=c11：使用 C11 標準

-O2：開啟最佳化

-o game.exe：輸出執行檔名稱為 game.exe

-lraylib -lopengl32 -lgdi32 -lwinmm：連結 Raylib 及 Windows 底層庫

編譯成功後，在同一終端執行：./game.exe即可啟動遊戲視窗。

操作說明

地圖介面 (Map Screen)

1. 遊戲開啟後，先在 Console (文字畫面) 輸入玩家姓名並按 Enter。
2. 正式進入 Raylib 繪製的 5×5 地圖視窗。
3. 使用鍵盤 W / A / S / D 控制角色上下左右移動：
4. 地圖上有兩個特殊格子：

(1,3) 紅色格 (Snake 關卡)

(3,1) 藍色格 (1A2B 關卡)

1. 角色走到紅格或藍格時，會跳出相應小遊戲 (Console 文字版)。
2. 如未通過該關卡，會顯示 Game Over 畫面，按 Enter 重新回到地圖。
3. 只要同時通過「貪食蛇 (Snake)」與「1A2B」兩個關卡，即顯示 Congratulations 畫面，並結束遊戲。

關卡一：貪食蛇 (Greedy Snake) 規則

1. 初始化

遊戲開始時，蛇長度初始為 3，位置預設放在地圖左上角 (0,0)、(1,0)、(2,0)，蛇頭為 (2,0)，蛇身向右展開。

地圖大小可選：預設為 20×20，但程式中可視需求修改為 10×10 或 15×15。

障礙物 (X) 與加成道具 (S=速度、R=縮身、L=多一命) 隨機生成。

1. 移動與操作

使用鍵盤 W/A/S/D 控制方向；蛇不可直接向相反方向移動（例如往上後立即往下會被忽略）。

每次移動後：

* + - 如果新位置是空格 (.)，蛇正常前進、尾部會跟進。
    - 撞牆或咬到自己，或碰到障礙且生命用盡 → 遊戲結束 (Game Over) 回傳失敗 (0)。

1. 食物 (F) 和加分

地圖上會隨機出現食物 (F)。蛇頭碰到食物：

* + - 得分 (state.score) +1。
    - 蛇長 length +1 (在下一帧生長一格)。
    - 地圖隨機再放一個新食物。
    - 每累計 10 分，蛇速度 (snake->speed) 減少 100,000 微秒，下限 200,000 微秒。

1. 障礙物 (X)

蛇頭碰到障礙：

生命 (state.lives) 減 1。

若生命變成 0 → 遊戲結束 (Game Over) 回傳失敗 (0)。

如果還有生命，當前障礙移除，並立即隨機生成新的障礙，但本次不再移動蛇頭 (需重新按方向)。

1. 加成道具 (Booster)  
   地圖上隨機放置三種加成：

S (Speed Booster) : 讓蛇的速度再提升一次 (減少 100,000 微秒)，最低可到 100,000 微秒。

R (Shrink Booster): 縮短蛇身兩格 (當前身長 > 3 才生效)。

L (Life Booster) ：額外 +1 條生命。  
每種加成道具只能使用一次 (用標記避免重複觸發)。

1. 勝利條件

當玩家的分數 (state.score) ≥ 10 或 蛇的長度 (snake->length) ≥ 15 時，視為通關 → 顯示「恭喜・通關」訊息，並回傳成功

1. 失敗條件

蛇頭撞牆、咬到自己；或碰到障礙且生命用盡 → 顯示 GAME OVER! 並回傳失敗。

關卡二：1A2B 猜數字 (1A2B Game) 規則

1. 角色選擇

進入關卡後，先顯示提示：

請選擇角色：1 = 出題者 (Questioner)，2 = 猜題者 (Answerer)

輸入 1 或 2：玩家輸入 1 表示自己當出題者，輸入 2 表示自己當猜題者。

1. 猜題者模式 (Answerer)

程式從 5040 組「4 位不重複數字」隨機選一組當秘密答案 (secret)。

玩家最多 7 次（MAX\_ATTEMPTS）去猜；每次需輸入一組「4 位不重複數字」。

範例輸入：1234

程式檢查合法性（長度 4、數字、互不重複）。

每次猜完後，程式計算並顯示 XAYB：

X (A)：位置＆數字都對的個數

Y (B)：數字正確但位置錯的個數

例如，答案 8123，玩家猜 1052 → X=0、Y=2，顯示 0A2B。

如果某次 X==4（猜中），立刻顯示「You Win!!」並回傳成功 (1)。

若 5 次都沒猜中，顯示「次數用完了，你失敗了。」並回傳失敗 (0)。

1. 出題者模式 (Questioner)

玩家當出題者，需先在 Console 輸入一組「4 位不重複數字」，作為秘密答案。

範例：1234

程式檢查：長度 4、皆為數字、互不重複。若不合規則，直接視為失敗 (0)。

程式預先生成所有 5040 組可能答案 (4 位不重複數字)。

電腦第一次隨機從 5040 組中抽一組作為第一個猜測 (pc\_guess)，並顯示：PC Guess: 5678請輸入提示(xAyB)

玩家需手動輸入 xAyB（例如 1A2B），表示有幾個 A、幾個 B。

如果玩家給錯提示（無法在現有候選集合中找到任何組符合該提示），程式立刻顯示「You Cheat!!」並回傳失敗 (0)。

若格式不符合 \dA\dB（數字字元 + 'A' + 數字字元 + 'B'），直接視為失敗 (0)。

程式透過「淘汰法」(Minimax-like) 在剩下的候選集合 (setarr[]) 中篩掉所有不符合該次 A/B 的組合，得到 newset[]。

如果 newset 數量為 0，表示玩家作弊 → 「You Cheat!!」→ 回傳 0。

如果 newset 數量為 1，代表只剩一種可能，下一次電腦直接猜這組。

否則程式呼叫 search\_next\_guess(all, newset, 5040, newsetcount)，從所有 5040 組中挑出一個「預期最均勻分布」的猜測作為下一個 pc\_guess。

每次電腦猜完，就算該猜測是否與玩家的答案完全相同：

如果 pc\_guess[] == answer[]，代表電腦猜中 → 顯示「PC Wins!」並回傳失敗 (0)。

否則繼續下一輪猜題。

電腦最多猜 5 次 (與猜題者模式相同限制)，若在第 5 次內都沒猜中玩家答案 → 玩家 (出題者) 勝利 → 顯示「恭喜！」並回傳成功 (1)。

4.作弊檢查

玩家在出題者模式若「亂給提示」，造成 newsetcount == 0，程式視為作弊並立即顯示「You Cheat!!」，關卡失敗、回傳 0。

分工:

主程式整合：411410076劉夏綱

貪食蛇關卡：411410076劉夏綱

1A2B 關卡：411410076劉夏綱

文件撰寫 (README、說明文件)：411410076劉夏綱

報告:411410076劉夏綱