

ASM Final Project :

主題：畫格子遊戲

組員：B94902002 葉耿甫 B94902036 徐名尉

一. 程式使用方式：(final project.asm)

1. 兩人對戰版本：

執行兩人對戰版本之 exe 執行檔。

出現遊戲畫面後，由 1P 開始以上下左右鍵移動游標並在想要畫線的位置按 enter。兩個玩家輪流執行此畫線動作。畫過的線不能被畫。

若一次畫線後使得任意一個格子的四邊皆被圍住，則將該格子填上畫上最後一筆玩家的顏色（1P 為紅色，2P 為綠色），並紀錄該玩家得一分。

且該玩家繼續畫線的動作（即：若任意玩家得分，則他必須再畫一筆。）

遊戲結束於所有的邊線都被畫過，此時統計分數並顯示哪個玩家勝利。程式結束。

2. 與電腦對戰版本：(僅完成部分) (final project2.asm)

執行與電腦對戰版本之 exe 執行檔。

遊戲方法與過程同 1，只是 2P 換為電腦控制，玩家只有一個並和電腦對抗。

二. Procedure :

1. main：主程序。

2. AddBlockR：增加 block 內的值，有幾個周圍的邊已被劃。(水平)(僅存於電腦對戰版本)

3. AddBlockC：增加 block 內的值，有幾個周圍的邊已被劃。(垂直)(僅存於電腦對戰版本)

4. Final：判斷結束與否，並判斷勝負結果。

5. RecordTime：紀錄按下 enter 之時間。

6. WriteTime：顯示思考時間。

7. WriteTurn：在玩家前顯示符號標示目前輪到的玩家。

8. WritePixel：畫點。(初始設定)

9. WrtieGrade：顯示分數。

10. WriteRecord：畫起始記分版。(初始設定)

11. MakeFrame：畫邊框。(初始設定)

12. WaitKey：等待並判斷使用者的輸入，並加以反應。

13. EnableBlanking：(未完成)讓游標會閃動。

14. WriteLineR：畫水平的線。

15. WriteLineC：畫垂直的線。
16. ReadPixelValueR：讀現在的 Pixel value。（水平）
17. ReadPixelValueC：讀現在的 Pixel value。（垂直）
18. DrawFill：對格子塗色。
19. DrawFillPosition：判斷塗色與否與塗色位置。
20. ComMove：控制電腦行為。（僅存於電腦對戰版本）

三. 電腦行為之演算法：

1. 基本演算法：

首先，電腦會判斷盤面是否具有”Score block”，”Score block”為已有三邊被畫過的格子。如果有，則進行填滿取分的動作。

在盤面不具”Score block”時，電腦會判斷盤面是否具有”Save edge”，定義為相關格子皆不是已填有 2 個邊的格子。如果有，則畫”Save edge”並將 turn pass 給另一玩家。

（註：實作僅至此，接下來的部分以任意取一線畫上代替。）

在盤面不具”Save edge”和”Score block”時，我們可以將盤面分為下面三種格子：

已填滿的格子。

已填 2 個邊的格子。

旁邊只有上面兩種格子的已填 1 或 0 個邊的格子。

此時電腦忽略已填滿的格子，並將所有 2 個邊的格子之間，具有共同未畫邊的設為同一個 set。

分完 set 後，從其中取出最小的 set 並畫其中一邊。

2. 基本演算法之證明：

Greedy 在這個遊戲中是適用的，首先因為得過一分還能再畫一筆這個性質使得有拿分機會必定要拿分這件事永遠不會錯。

接著，假設我們可以將每次的連續拿分視為很多個不同 block 的 set，我們可以把遊戲想成一個奇偶性的問題。當 block set 總數為奇時，後手開始讓對方得分是比較有優勢的。因為每個玩家都會盡可能讓對手拿到當時最小的區塊，後手玩家在此前提下會取得優勢。反之，當 block set 總數為偶時，先手開始讓對方得分會因為可以先進行選擇而獲得優勢。

由上分析所得知，盡可能不讓對手拿分也是很合理的，則在還有機會不讓對手得分時不讓對手得分總是對的。（無論在 block set 合為偶數或奇數時，讓對手先取得一個或數個 block set 並不會改善先後手的結果，有時甚至會更糟。）

3. 推廣：

將 block set 分為兩種，分別為開放式 block set 和封閉式 block set。

開放式的定義為 block set 的開端和結尾不是同一個 block，封閉式 block set 則是具有同一個開端和結尾。

關心他們邊與格數的關係，開放式 block set 內若有 a 個 block，則他的未填滿邊共有 a+1 個，而封閉式 block set 內若有 b 個 block，則他的未填滿邊共有 b 個。

假設一次遊戲的 block set 皆為開放式 block set，則在全部 safe edge 畫完後，畫過的邊數會有總邊數-block set 之未填滿邊總數。這個數字等於(block set 數+總 block 數)。

又，由於總邊數為偶數個，故總邊數-(block set 數+總 block 數)之奇偶性和 block set 數+總 block 數相同。

從這點可以知道，當總 block 數為偶數時，總 block 數和已畫邊數會有相同的奇偶性，這會使先手者必勝。反之，則後手者必勝。

接著討論有封閉性 block set 的情形，若將兩個開放式 block set 組合為一個開放式 block set，則減少一個未填滿邊卻也減少一個 block set 總數，未影響奇偶性。若將兩個開放式 block set 組合為一個封閉性 block set，則減少兩個未填滿邊卻只減少一個 block set 總數，將改變奇偶性。

故我們可得，此遊戲的目標等價於在總 block 數不利(即：全部皆為開放式 block set 時會敗)時創造奇數個封閉式 block set 改變奇偶性。反之則要維持偶數個封閉性 block set 維持優勢。

(此推廣只限於理論，實作方式還有待研究。)

四. 製作過程遭遇之困難

1. 分 Library 的困難

由於我們程式中的 procedure 大多沒有 reuse for other program 的價值，所以在計畫分 library 時費過一番腦筋。然而實在沒有辦法，也因此最後決定不分出 library。

五. 分工

B94902002 葉耿甫負責 GUI 編寫，人腦對人腦部分。

B94902036 徐名尉負責電腦行為部分。