

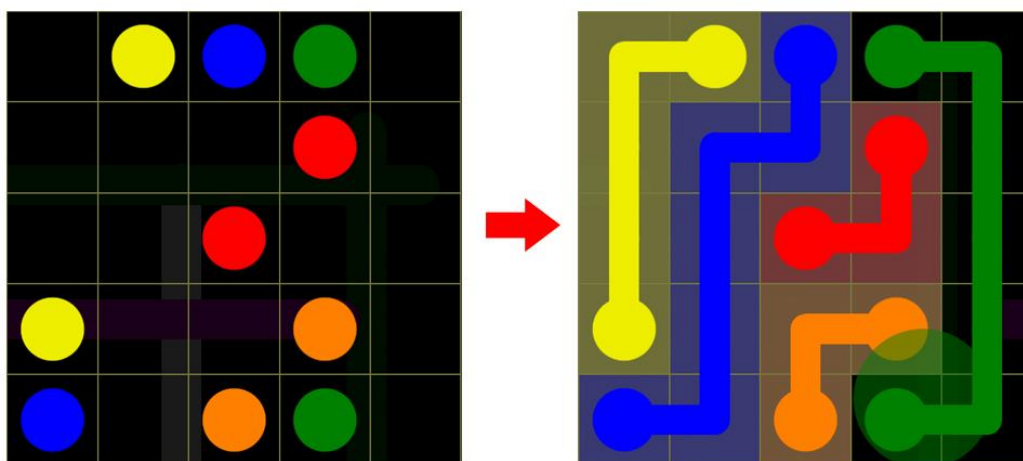
資料結構 Final Project 期末報告

電機四 A 110501529 鄭凱方

電機四 A 110501531 戴仕庭

一、題目介紹

Flow Free 是一款備受推崇的經典益智解謎遊戲，以其簡潔的設計和深刻的邏輯挑戰吸引了眾多玩家。遊戲的核心目標是在棋盤上，將相同顏色的起點和終點以線條相連，形成完整且不交錯的“管道”路徑。同時，所有路徑必須填滿棋盤，達成無縫覆蓋。在基本規則之外，遊戲更進一步加入了一項策略性極高的挑戰——在相鄰的四個色塊中，同一顏色的數量不得超過兩個，這一規則對玩家的規劃能力和邏輯思維提出了更高要求。



在 Flow Free 的 Final Project 中，採用雙人合作模式，以充分發揮團隊協作與個人創意。專案主要分為兩個核心部分：出題設計與解題實現。出題設計的重點在於如何根據使用者提供的棋盤尺寸，設計出既合理又具有挑戰性的題目。不僅需要確保題目符合遊戲規則，還可以通過設置額外限制（例如，相同顏色連線的最短距離）來提升遊戲的趣味性與難度。解題實現則聚焦於開發高效的演算法，以最短的時間正確解出給定尺寸的任意題目。這需要探索並實現先進的搜尋與優化技術，確保演算法的準確性和性能，特別是在面對複雜題目時能快速產生最佳解。

二、組員分工

- A. 戴仕庭: FLOW FREE 遊戲出題
- B. 鄭凱方: FLOW FREE 遊戲解題

三、實作方法&實際結果

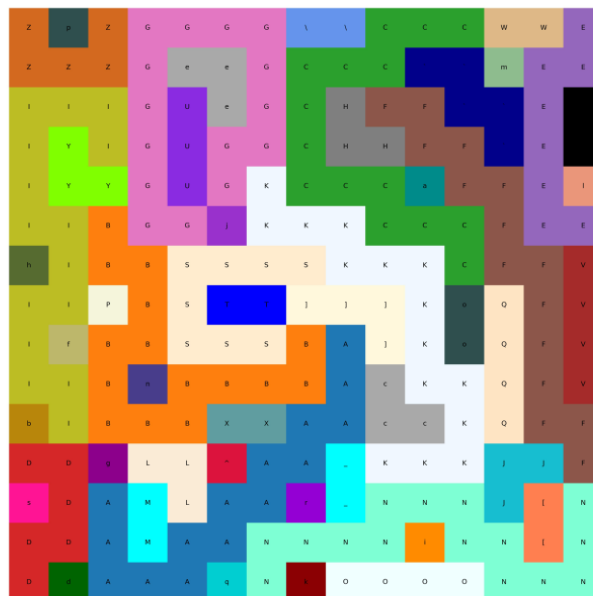
A. FLOW FREE 遊戲出題:

1. 使用者輸入: 棋盤的尺寸(長&寬)、單一顏色的最短長度
2. 程式輸出: _game.txt、_game.jpg、_solution.txt、_solution.jpg、_info.txt
3. C 語言程式: Flow_Free_Generator.c、Generator_IO.h(產生 txt 檔案)
4. 自動化程式: color_auto_process.py(txt 轉 jpg)、auto_process.bat
5. 資料結構使用: 2D Array and Linked List with dynamic memory allocation
6. 主程式運行流程:

主要為七個步驟生成圖片，為了確保每次圖片都有足夠的隨機性，所以用了大量的 random()，也導致程式運行時間難以估計。當棋盤越大或是最短長度限制越長，收斂的時間就會越長。以下程式介紹以功能介紹和各階段結果輸出為範例來說明。因為製作過程中輸出為一個字符代表一個顏色，但是由於電腦能輸出的字符有限，因此超出一定限制的 ASCII CODE 會導致程式錯誤，所以本程式仍有一定的棋盤尺寸限制。

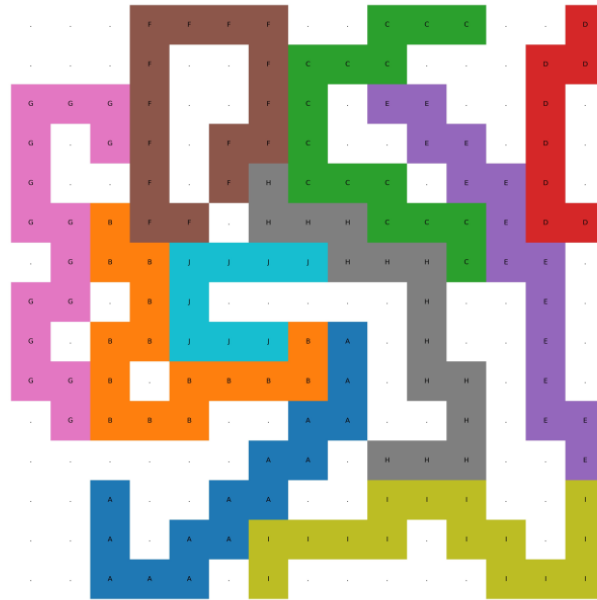
a. build_path()

先隨機從棋盤中挑選一個起點，在規則下(相同顏色不可相鄰)盡可能延長此顏色的管道，並將整個棋盤都佈滿顏色。此部分主要是為了在短時間內先構建一個初始解，讓後續更容易優化調整。



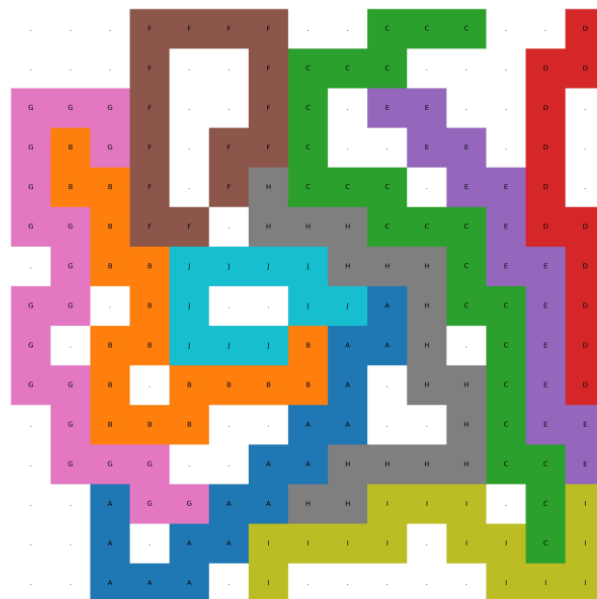
b. delete_short_path()

刪除棋盤上比最短長度限制還要短的顏色管道。由於前一步不會去管長度的問題，所以此部分是為了清空不成功的顏色布局，讓後續繼續根據長度限制去優化。



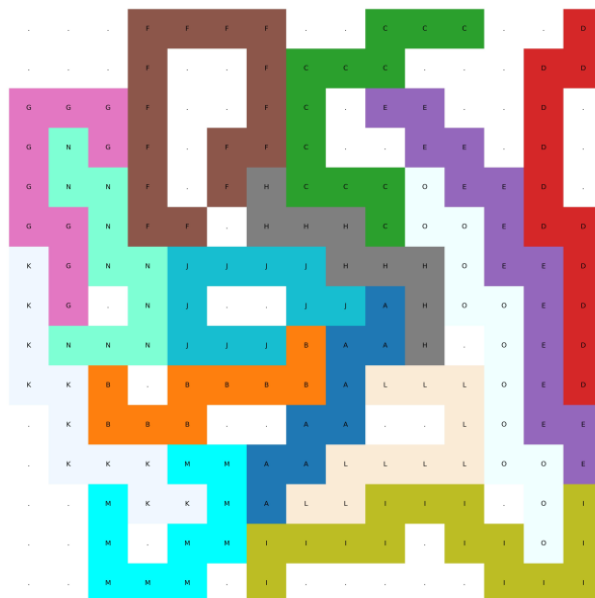
c. extend_the_head_of_path()

根據既有的顏色通道去優化，讓其盡可能延伸其起點端，此步驟是為了拉長原本的顏色，盡可能多把零碎空間去除，此步驟不會去延伸終點端是因為考量，終點端的線段若可以延長是由於原先早於該顏色的色塊佔據，但若延伸此部分，則容易導致空白的空間更加零碎，會增加進一步優化的難度。



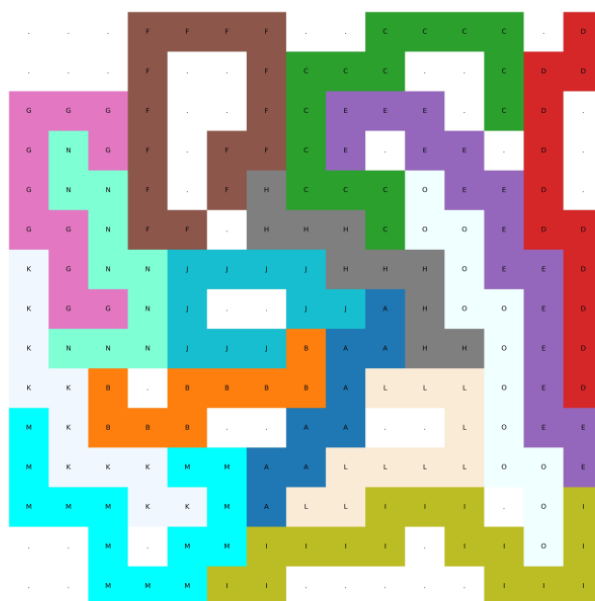
d. seperate_long_path()

此步驟會去將原先足夠長的顏色通道，根據其周遭的空白空間去切分為多段，讓其延伸到原先通道附近的空白空間。此部分主要是對長度做犧牲，由於越長的顏色通道對於生成題目的限制越大，因此這樣的調整可以讓題目生成速度提高。



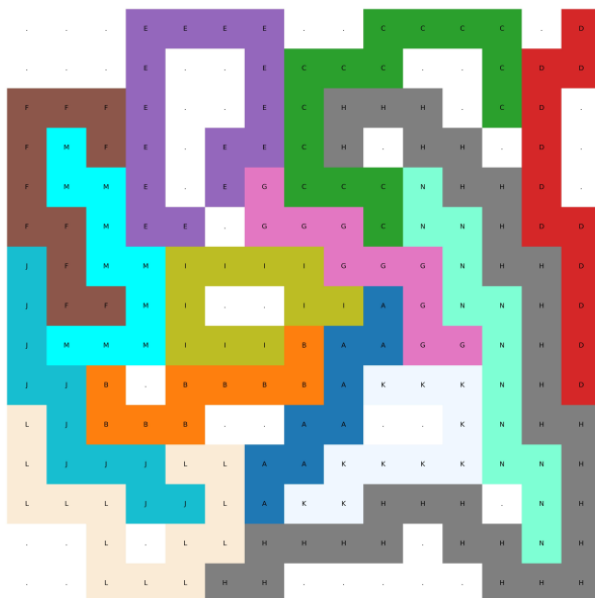
e. extend_two_ends_of_path()

在做完上述優化後，會把既有線段的兩端同時延長，將空白空間盡可能刪除。此步驟主要是考量到如果留下越多空白空間到最後一步，通常會需要更多時間才能收斂出結果。



f. `connect_two_paths()`

此步驟會將既有顏色的通道，如果滿足不相鄰的限制，且其兩顏色的其中一端有相鄰，將之相連。此步驟之目的在於減少顏色數量，由於在最後一步執行過程中容易產生大量顏色而出錯。



g. `fill_empty_cells()`

此為整個程式的最後一步，因此會使用較為激進的作法，通常也是造成程式運行時間較長的原因。此步驟會以空白空間為單位去優化，將空白空間附近的隨機一個顏色通道刪除，然後以原本空白空間為起點，重新產生顏色並盡可能延伸。而如果新的線段不夠長，則會被刪除，留下更多空白空間，來讓下一次的優化有更多的完整空間可以利用。

