

LabVIEW Final Project 說明文件

施孟成 陳昱丞 (第 7 組)

需求規劃

設計動機

Rush Hour Traffic Jam Logic Game 是一個滑塊類益智遊戲，於 1970 年代由蘆原伸之先生發明。



圖、實體遊戲照

遊戲是在一塊 6 x 6 平面方格的棋盤遊玩，共 36 個格子，車輛長度為 2 格的轎車或 3 格的卡車，紅車必定為 2 格，每台車都只能上下或左右，沿著車頭或是車尾方向移動，遊戲目標為將紅車移動到出

口。



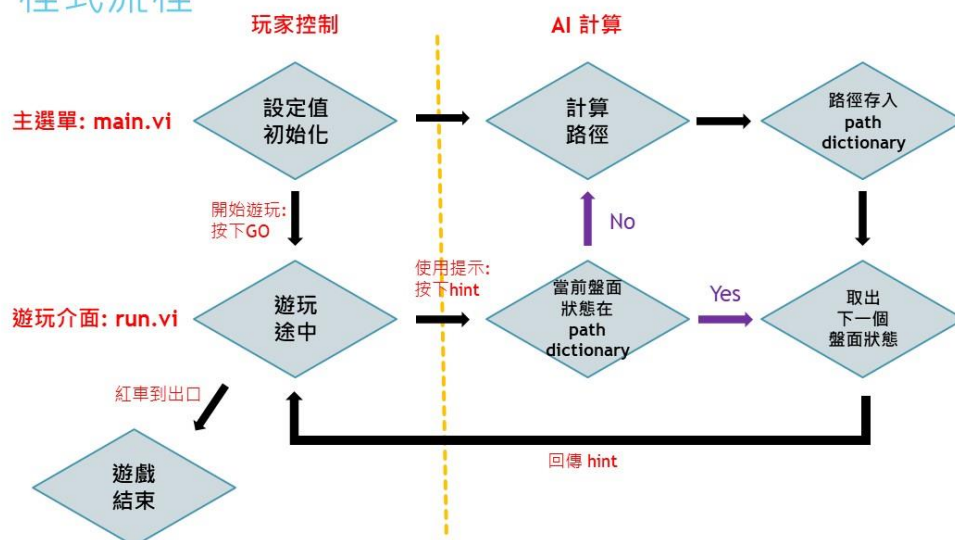
圖、出口必定在 (2, 5)，紅車必定在 row 2

需求分析

挑選它的原因在於它是小時候普遍玩過的益智遊戲之一。看似簡單，實則複雜。難度較高的關卡要在短時間少步數內解出並不容易。在我們設計的遊戲程式中包含有共五階段難度分級可自由選擇。大人小孩都適合拿來訓練思考，相當適合喜愛腦力激盪、想要訓練耐心的人。

程式流程規劃

程式流程



規則說明

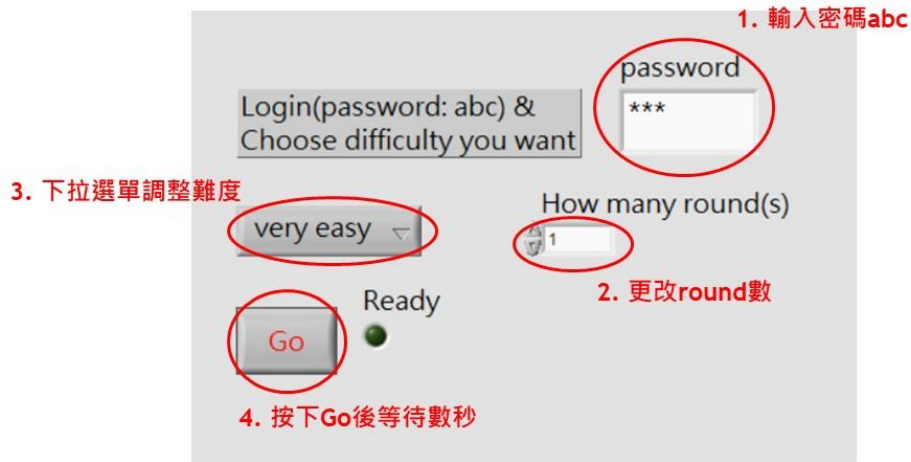
對弈方式(遊玩方式)

評斷標準有二，分別是：解題時間、解題步數。我們採用的是解題時間，較短者獲勝。在雙人模式當中，關主和挑戰者比較相同起始盤面移動至結束的較短時間者則獲勝。

接著說明遊玩方式

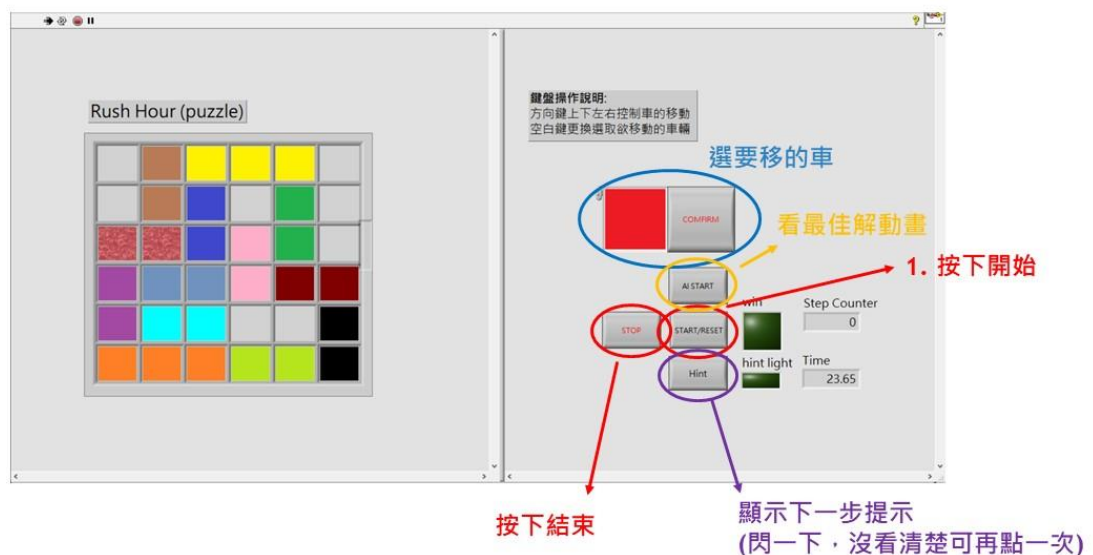
遊玩方式 - 單機模式 - 1

single-player/main.vi



無連線(單機)版本中，依序輸入密碼，選擇遊玩輪數，調整難易度，選擇完畢按下 Go 按鈕準備進入遊玩畫面。

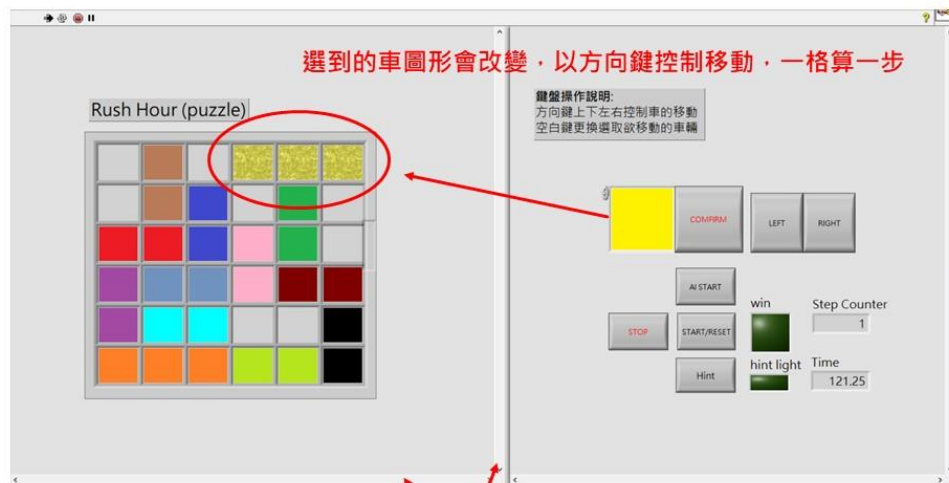
遊玩方式 - 單機模式 - 2



跳出遊玩畫面後，左邊會是遊玩的盤面，右邊是操控介面，按下

START/RESET 按鈕開始計時與計步數。AI START 按鈕會播放解題動畫。HINT 按鈕會根據遊玩畫面給予玩家下一步的提示。

遊玩方式 - 單機模式 - 3



PS: 若找不到盤面，可滾動或拉旁邊卷軸尋找

遊玩方式 - 單機模式 - 4



遊玩過程中，可在選單透過滑鼠選擇欲移動的車輛，也可透過鍵盤

空白鍵更換車輛，其中被選擇的車輛會呈現特殊色。可透過滑鼠點選螢幕按鈕或者鍵盤方向鍵選擇車輛移動方向。而當玩家移動步數相等於最佳解步數時，會有鈴聲(beep 一聲)提示。

連線(多機)版本中，提供兩支程式，分別有關主及挑戰者，兩者透過 UDP 連線實現，遊玩時由關主選擇遊戲難易度。以下會針對與單機版本不一樣的地方特別說明。

遊玩方式 - 連線對戰 - 1 (關主)

multi-player/server/main.vi



遊玩方式 - 連線對戰 - 2 (挑戰者)

multi-player/client/main.vi

1. 選擇單人或連線模式，若選one player則無連線；選two player則和關主連線



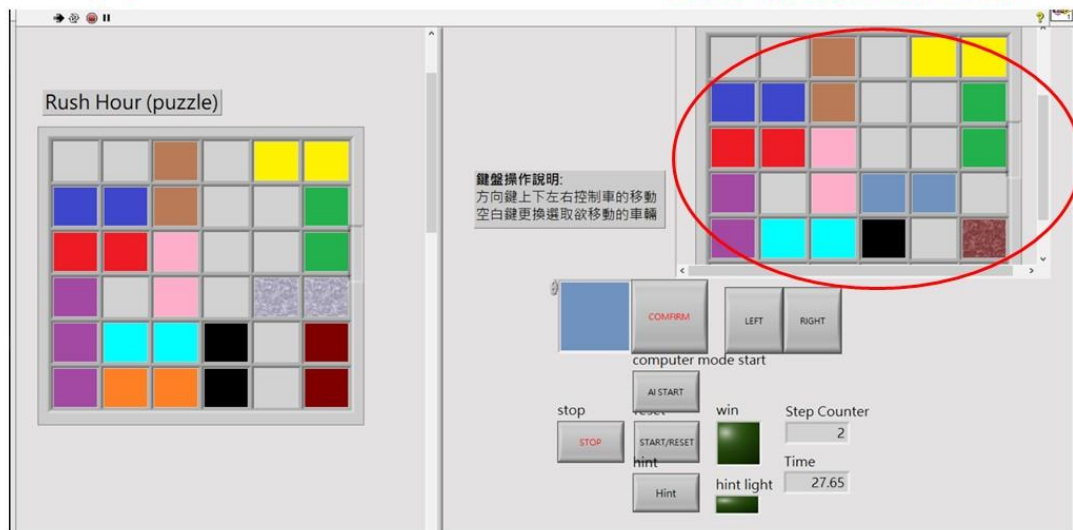
PS-1: 關主必須先開始執行程式 (server.vi)，挑戰者再在另一台電腦上執行 (client.vi)

PS-2: server.vi 和 client.vi 必須在同一個區域網路下，才能完成連線

遊玩方式 - 連線對戰 - 3 (挑戰者)

multi-player/client/main.vi

挑戰者可以看到關主的遊玩實況



遊玩規則與單機版本基本相同，在挑戰者遊戲畫面的右上角會有關主即時的遊玩畫面。

遊玩方式 - 連線對戰 - 4 (挑戰者)

multi-player/client/main.vi



分數計核規則、對弈結果判定

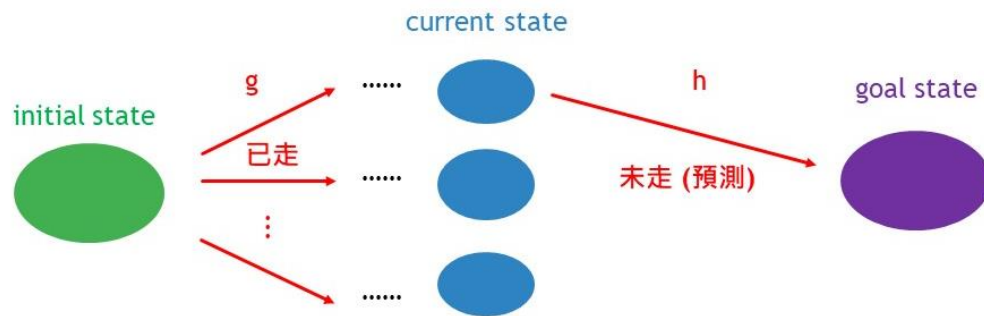
在單機版本中，會記錄不同輪之間的秒數，時間少者勝。

在連線(多機版)本中，會比較關主與挑戰者間的秒數，時間少者勝。

AI 玩家說明

利用當前玩家的盤面狀態，做 State Space Search，其中我們選用 A* 演算法，找到距離最短的 goal state。搜尋完成後，用回溯法找到最短路徑，即最少步數走法。我們將整條路徑存進一個 dictionary，方便玩家使用 HINT 按鈕查找下一步時，有機會可以不用重新搜尋一次。若當前 state 不在 dictionary 裡面，依然必須重新搜尋，並將

搜尋完成的路徑 push 進 dictionary。



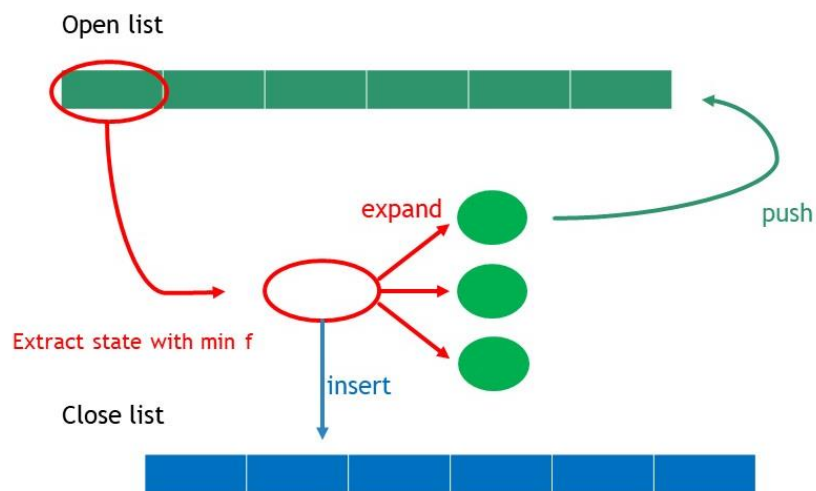
圖、A*演算法示意圖

$$f(n) = g(n) + h(n)$$

$g(n)$ ：從啟始點到目前節點的距離

$h(n)$ ：預測目前節點到結束點的距離

$f(n)$ ：目前節點的評價分數



圖、A* state-space search

Blocking Heuristic

紅車前面有幾台車，代表至少需要幾步才能將他們移開，要將他們

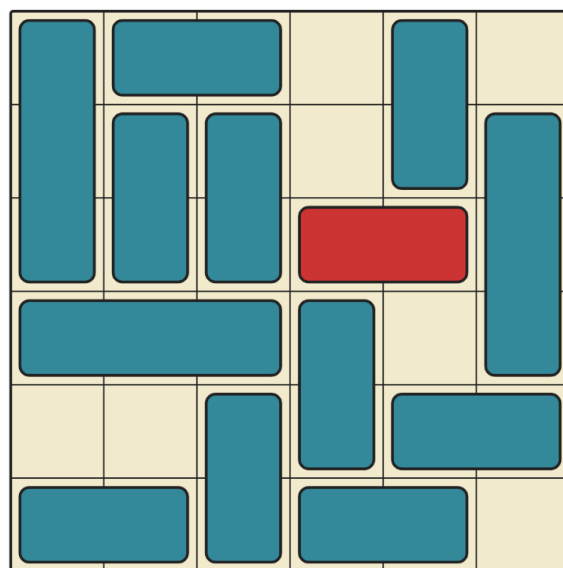
都移開紅車才能出得來。紅車出來的步數 $>$ 需移開的車子數 (擋在紅車前的車子數)，擋住的车子越多，距離 goal state 越遠。不會 over estimate。



圖、藍色車擋住了，需要移開

REFERENCES、相關研究

[1] Solving Rush Hour, the Puzzle - by Michael Fogleman
(<https://www.michaelfogleman.com/rush/>)



圖、遊戲畫面截圖

[2] On the Hardness of 6x6 Rush Hour - An exploration of the entire configuration space - by Jelle van Assema

[3] Programming Puzzle: Rush Hour Traffic Jam – by P.G. Baumstarck

APPENDIX

評分表說明：

7. 選單選擇 5 階段難度 (very easy.....very hard)
9. 用 UDP 實現區域網路內關主、挑戰者連線對戰
10. 玩家到達最短步數時會鈴聲提示
11. HINT 提示下一步、AI START 展示最佳解法動畫

AI-Game 普通版 評分表

編號	配分	功能需求說明
1.	(10%)	[文件]AI-G_需求規劃：描述此AI-G之設計動機、需求分析 程式規劃流程等內容；
2.	(10%)	[文件]AI-G_規則說明：描述此AI-G的對弈方式、分數核計 對弈結果之判定邏輯、以及任何參考資料；
3.	(10%)	[文件]AI-G_AI玩家說明：描述AI玩家的設計理念、預定導 入的方法、以及相關的參考資料；
4.	(10%)	(AI-G_人機介面設計)在 Front panel上實現此AI-G的操作介 面；
5.	(10%)	(AI-G_程式方塊圖撰寫)在Block diagram撰寫出各項程式功 能；
6.	(10%)	(AI-G_程式設計者說明文件) 在Block diagram撰寫說明文字
7.	(10%)	可手動調整遊戲難易度，增加其對應玩家數量；
8.	(10%)	Block diagram上，可採用 producer/consumer架構來實現；
9.	(10%)	可成功從單機(localhost)測試到多機(兩台PC以上)測試；
10.	(10%)	操作者玩遊戲時，可依據遊戲進度來播放(背景)音樂，建立 密碼權限確任模組；
11.	(20%)	實現電腦玩家(AI-based)，可啟動展示、快速獲分等功能！