

## Tzaar Game Requirements

### 1. Introduction

本文件簡述 Final Project 的需求，並說明整體 Tzaar 遊戲平台以及同學所設計撰寫的 Tzaar 策略程式在遊戲平台中的應有責任 (Responsibility)。由於同學彼此之間的策略將會互相進行對戰，所以進行策略實做時，必須遵守溝通介面的規定，以確保同學間的策略程式能互相溝通。本文件定義須遵守的溝通介面，請同學務必遵循溝通介面來完成策略程式的撰寫。

### 2. Final Project Requirement

請同學依照助教所規定之條件及溝通介面以 C programming Language 進行 **Tzaar 遊戲策略程式的設計及實作**，並且能夠使用該策略程式成功地在助教所提供的平台上進行遊戲。同學所設計撰寫的遊戲策略程式的執行週期為一次的**移動棋子決策**，以下詳述該策略程式的需求。

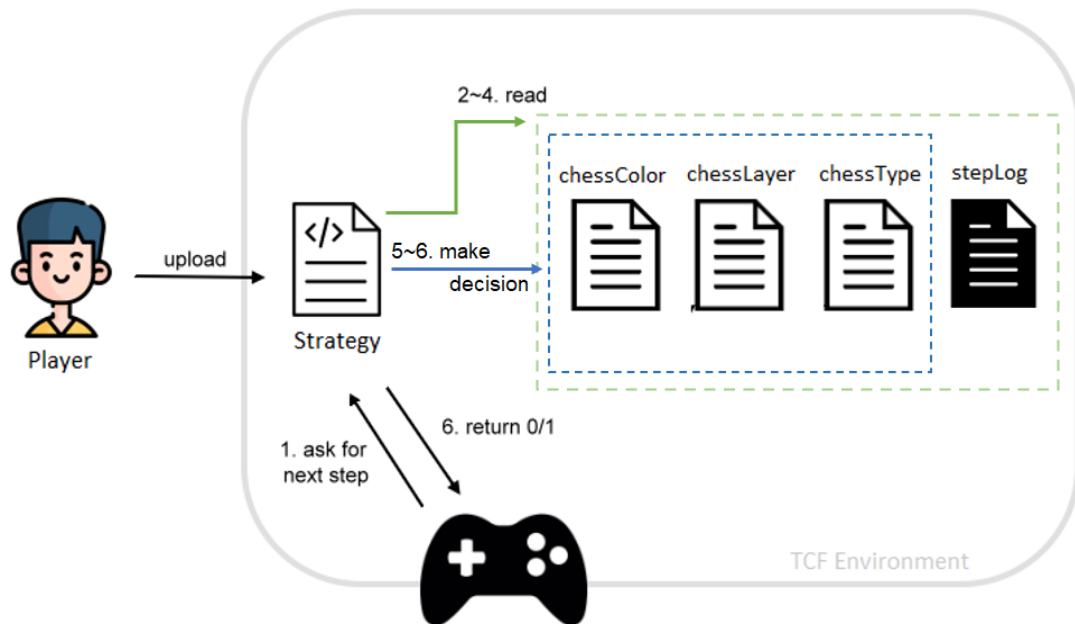
1. 本次 Final Project 的 Tzaar 遊戲，遊戲平台會讓兩位玩家使用各自的策略程式進行遊戲
2. 該策略程式會預期從執行時的執行參數得知棋子顏色分布狀況檔案 (chessColor)、棋子層數分布狀況檔案 (chessLayer)、棋子種類分布狀況檔案 (chessType)、玩家決策歷史紀錄檔案(steplog) 的路徑
3. 該策略程式會依棋盤目前狀態檔案路徑來讀取棋子顏色分布狀況檔案 (chessColor)、棋子層數分布狀況檔案 (chessLayer)、棋子種類分布狀況檔案 (chessType) 的內容，得知目前的棋盤狀態
4. 該策略程式會依據玩家決策歷史紀錄檔案路徑來讀取玩家決策歷史紀錄檔案 (steplog) 的內容，得知截至目前為止雙方玩家所有的決策紀錄
5. 依據棋盤目前狀態以及玩家決策歷史紀錄，該程式會決定此次步驟要執行的動作內容
6. 該策略程式會根據此次步驟決定要執行的動作內容，來變動棋盤目前的狀態，並依格式規定將更改後的棋盤狀況覆蓋寫入至棋子顏色分布狀況檔案 (chessColor)、棋子層數分布狀況檔案 (chessLayer)、棋子種類分布狀況檔案 (chessType) 中，最後傳回 0 (return 0) 結束程式

註 2: 策略程式不須且不能更改 steplog 中的內容

### 3. Game Rules

本節說明遊戲平台進行時的規則，讓同學了解遊戲平台中除了正常移動棋子規則的對戰規則之外 (Tzaar 遊戲規則請參考之前發佈的遊戲介紹文件)，還有違規情形的發生，違規詳情請參考先前發布的 Game Introduction 文件，而一旦有違規情形發生時，違規玩家將被強制判輸，由敵方玩家獲勝。

### 4. Overall Architecture



上圖呈現整體的 Tzaar 遊戲平台的運作方式，由助教提供的遊戲平台會負責整個遊戲進行的控制、遊戲規則的仲裁以及遊戲畫面的呈現(此部分由助教提供)，而 Strategy 則為學生所撰寫的遊戲策略程式。

由於 Tazzr 遊戲中，除第一回合外，每一個回合都有兩個步驟 (Tazzr 遊戲規則可參考先前發布的 Tazzr Game Introduction 文件)，因此在遊戲過程中，策略程式是以一個步驟為單位來進行決策的判斷。

在每個步驟的開始，遊戲平台會要求 Strategy 進行決策，並提供給 Strategy 最新的棋子顏色分布狀況檔案 (chessColor)、棋子層數分布狀況檔案 (chessLayer)、棋子種類分布狀況檔案 (chessType) 和玩家決策歷史紀錄檔案 (steplog)，Strategy 要依據此兩檔案所提供的資訊進行決策，並將執行該決策後的棋盤結果覆寫進棋子顏色分布狀況檔案 (chessColor)、棋子層數分布狀況檔案 (chessLayer)、棋子種類分布狀況檔案 (chessType) 中，最後再回傳 0 表示已完成決策。

完成決策後，遊戲平台會檢視棋盤目前狀態檔案 (chessColor, chessLayer, chessType) 中的檔案內容，並決定是否已分出勝負或是繼續進行遊戲。

## 5. Program Controlling Interface

同學的程式編譯後，在 Tzaar 遊戲平台中，會用以下指令在助教提供的 Server 環境下，以 command line 執行程式

```
[學號].out [Color] [Step] [chessColor] [chessLayer] [chessType] [stepLog]
```

- *[Color]*: 此策略程式所能操控的旗子顏色，資料值為 white 或 black。
- *[Step]*: 此策略程式目前將在哪一個步驟進行決策。資料值為 1 或 2。
- *[chessColor]*: chessColor.txt 的檔案路徑，此檔案描述當前棋盤上棋子顏色的分布狀況。
- *[chessLayer]*: chessLayer.txt 的檔案路徑，此檔案描述當前棋盤上棋子層數的分布狀況。
- *[chessType]*: chessType.txt 的檔案路徑，此檔案描述當前棋盤上棋子種類的分布狀況。
- *[stepLog]*: stepLog.txt 的檔案路徑，此檔案紀錄了到目前為止雙方移動棋子的歷史紀錄。

舉例來說

```
./E12345678.out white 1 ./chessColor.txt ./chessLayer.txt ./chessType.txt ./stepLog.txt
```

代表執行 E12345678 學生的策略進行一步棋步，其中

- 該策略控制的棋子為白方的棋子
- 此策略要在第 1 個步驟進行決策。
- chessColor.txt 的檔案路徑為 ./chessColor.txt
- chessLayer.txt 的檔案路徑為 ./chessLayer.txt
- chessType.txt 的檔案路徑為 ./chessType.txt
- stepLog.txt 的檔案路徑為 ./stepLog.txt

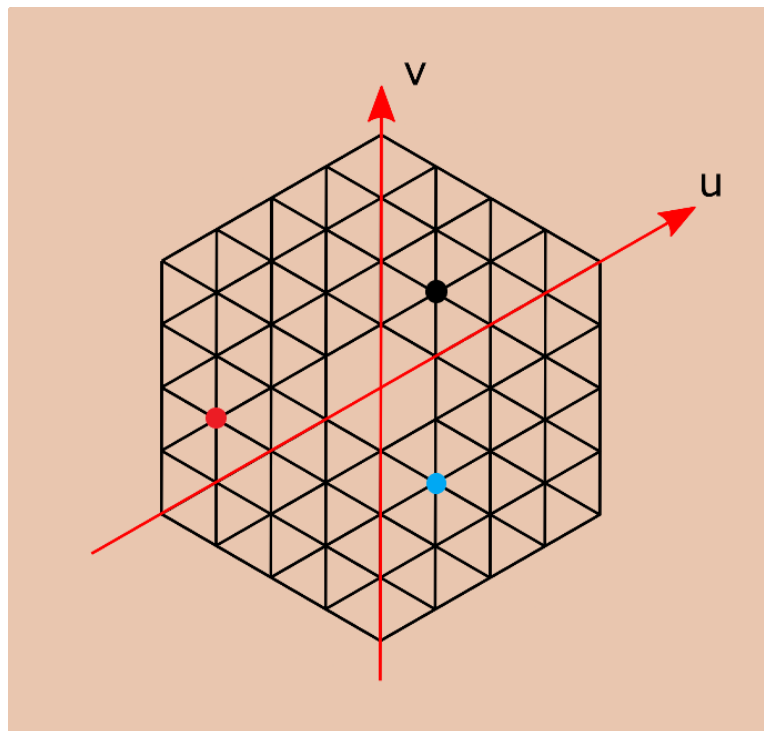
## 6. 棋子與棋盤

在遊戲進行時，遊戲平台會將棋盤中的棋子，利用座標的方式呈現，而棋子有種類以及層數之分，因此以下會介紹如何表達棋子和棋盤座標值的定義。

- Tazzr 中共有三種棋子，此三種棋子的編號分別為：

編號	代表意義
1	Totts
2	Tzarra
3	Tzaar

- 棋盤的座標軸表示為下圖所示



圖二、棋盤座標

棋盤以  $u$ 、 $v$  作為座標軸， $(u, v)$  作為座標點，以圖中三個不同顏色的點作為範例：

黑點座標為  $(1, 1)$

紅點座標為  $(-3, 1)$

藍點座標為  $(1, -2)$

在此座標系統下，會有不在棋盤上的點，如原點  $(0, 0)$ 。

## 7. Data Format

本節介紹遊戲進行時，遊戲平台會提供的四個檔案：chessColor.txt、chessLayer.txt、chessType.txt 和 steplog.txt，同學可依檔案所提供資訊，在策略程式中決定要進行什麼動作（吃子、併子或跳過）。

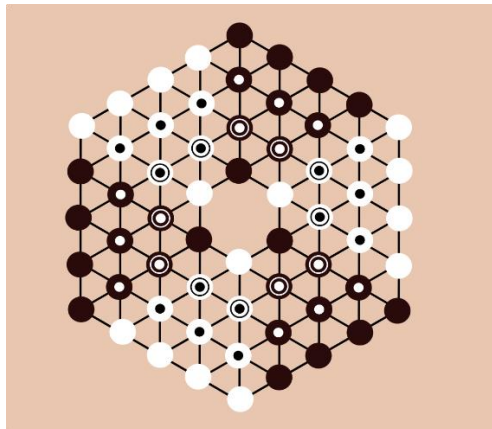
### (1) 棋子顏色分布狀況檔案格式 (chessColor.txt)

棋子顏色分布狀況檔案紀錄了目前棋盤上棋子顏色的分布狀況，並將棋盤座標轉換成直角坐標，以第 5 列為 x 軸，第 5 行為 y 軸，排列成 9\*9 的陣列（不含逗點），關於棋盤座標部分請參考圖二。

下表是 chessColor.txt 各個值所代表的意義：

值	代表意義
1	白色棋子
2	黑色棋子
0	在棋盤上但無棋子的點
-1	不在棋盤上的點

以下是描述開局棋盤分布的棋子顏色分布狀況檔案範例



圖三、開局棋盤

```

1  1,1,1,1,2,-1,-1,-1,-1
2  2,1,1,1,2,2,-1,-1,-1
3  2,2,1,1,2,2,2,-1,-1
4  2,2,2,1,2,2,2,2,-1
5  2,2,2,2,-1,1,1,1,1
6  -1,1,1,1,1,2,1,1,1
7  -1,-1,1,1,1,2,2,1,1
8  -1,-1,-1,1,1,2,2,2,1
9  -1,-1,-1,-1,1,2,2,2,2

```

chessColor.txt

**注意事項：**檔案最後不能有任何多餘符號（ex:換行符號）作為結束。

**(2) 棋子層數分布狀況檔案格式 (chessLayer.txt)**

棋子層數分布狀況檔案紀錄了目前棋盤上棋子層數的分布狀況，並將棋盤座標轉換成直角坐標，以第 5 列為 x 軸，第 5 行為 y 軸，排列成 9\*9 的陣列 (不含逗點)，關於棋盤座標部分請參考圖二。

下表是 chessLayer.txt 各個值所代表的意義：

值	代表意義
1 以上	數字的值即代表有幾層
0	在棋盤上但無棋子的點
-1	不在棋盤上的點

以下是描述開局棋盤分布的棋子顏色分布狀況檔案範例

```

1  1,1,1,1,1,-1,-1,-1,-1
2  1,1,1,1,1,1,-1,-1,-1
3  1,1,1,1,1,1,1,-1,-1
4  1,1,1,1,1,1,1,1,-1
5  1,1,1,1,-1,1,1,1,1
6  -1,1,1,1,1,1,1,1,1
7  -1,-1,1,1,1,1,1,1,1
8  -1,-1,-1,1,1,1,1,1,1
9  -1,-1,-1,-1,1,1,1,1,1

```

chessLayer.txt

**注意事項：**檔案最後不能有任何多餘符號 (ex:換行符號) 作為結束。

**(3) 棋子種類分布狀況檔案格式 (chessType.txt)**

棋子種類分布狀況檔案紀錄了目前棋盤上棋子層數的分布狀況，並將棋盤座標轉換成直角坐標，以第 5 列為 x 軸，第 5 行為 y 軸，排列成 9\*9 的陣列 (不含逗點)，關於棋盤座標部分請參考圖二。

下表是 chessType.txt 各個值所代表的意義：

值	代表意義
1	Totts
2	Tzarra
3	Tzaar
0	在棋盤上但無棋子的點
-1	不在棋盤上的點

以下是描述開局棋盤分布的棋子顏色分布狀況檔案範例

```

1  1, 1, 1, 1, 1, -1, -1, -1, -1
2  1, 2, 2, 2, 2, 1, -1, -1, -1
3  1, 2, 3, 3, 3, 2, 1, -1, -1
4  1, 2, 3, 1, 1, 3, 2, 1, -1
5  1, 2, 3, 1, -1, 1, 3, 2, 1
6  -1, 1, 2, 3, 1, 1, 3, 2, 1
7  -1, -1, 1, 2, 3, 3, 3, 2, 1
8  -1, -1, -1, 1, 2, 2, 2, 2, 1
9  -1, -1, -1, -1, 1, 1, 1, 1, 1

```

chessType.txt

**注意事項：**檔案最後不能有任何多餘符號 (ex:換行符號) 作為結束。

## (2) 玩家決策歷史紀錄檔案 (stepLog.txt)

玩家決策歷史紀錄檔案依序記錄了整個遊戲中玩家所進行的每一次玩家的決策，包含未移動棋子 (Pass)，檔案中每行即代表一次玩家的動作：

格式: [Color]\_[Round]\_[Step]\_[Action]\_[PathStart]\_[PathEnd]

- **Color**  
意義：執行此動作的顏色  
資料值：W (白色) 或 B (黑色)
- **Round**  
意義：執行此動作的所屬回合，每換一位玩家，回合數就會改變  
資料值：1, 2, ...
- **Step**  
意義：執行此動作的所屬步驟  
資料值：1 或 2
- **Action**  
意義：執行的動作為何  
資料值：Pass (跳過)、Capture (吃子)、Enhance (併子)
- **PathStart、PathEnd**  
意義：移動的起點 (PathStart) 與終點 (PathEnd)  
資料格式： u,v  
資料值：如果 Action 是 Pass，則值為 0,0

範例:

```
1 W_1_1_Capture_2,0_1,1
2 B_2_1_Capture_1,-2_0,-2
3 B_2_2_Enhance_0,-2_1,-3
4 W_3_1_Capture_-2,-1_-2,0
5 W_3_2_Pass_0,0_0,0
```

**注意事項:** 檔案最後不會有任何多餘符號 (ex:換行符號) 作為結束。

## 8. 實做注意事項

- 在執行策略程式時，遊戲資訊檔案不一定與策略程式置於同一工作目錄。為減輕同學判定遊戲資訊檔案位置之負擔，執行策略程式時，所傳入的檔案路徑參數將直接指定「檔案路徑」。
- 在決定移動的棋子後，策略程式只需要更新棋子顏色分布狀況檔案 (chessColor)、棋子層數分布狀況檔案 (chessLayer)、棋子種類分布狀況檔案 (chessType) 內容即可，**不需要更新玩家決策歷史紀錄檔案 (stepLog)** (遊戲進行中，助教開發的遊戲平台會負責更新該檔案內容)。