

Project Requirements

本文件描述了 Final Project 的需求，並簡述整個 Blokus Duo 遊戲平台以及同學所設計撰寫的 Blokus Duo 策略程式在遊戲平台中的應有責任 (Responsibility)，本文件亦詳述了平台中與同學撰寫的 Blokus Duo 策略程式溝通時的溝通介面，請同學務必遵循溝通介面來完成策略程式的撰寫。

Project Schedule

Date	Project Schedule
03/17	Final Project 正式公佈
04/14	Final Project 需求說明
04/28	Checkpoint 1 規定文件及測試平台公佈
05/19	Checkpoint 1 繳交期限
05/21	Checkpoint 2 規定文件及測試平台公佈
06/09	Checkpoint 2 繳交期限
06/10-06/16	Blokus Duo 大賽

Project Requirement

Project 的需求為：請同學依照助教規定之條件及溝通介面，以 C programming Language 進行 Blokus Duo 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拼圖擺放決策，以下詳述該遊戲策略程式的需求。

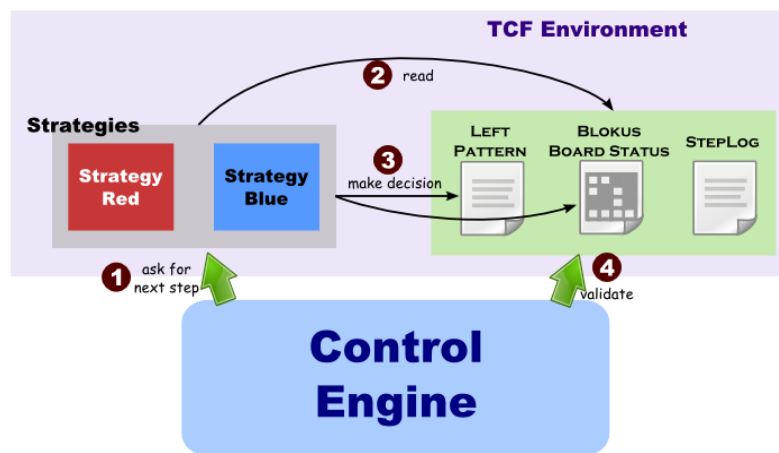
1. 該程式會預期從執行時的執行參數得知棋盤目前狀態檔案 (Blokus Board Status) 名稱、剩餘拼圖檔案 (Left Pattern) 名稱、棋步歷史紀錄檔案 (Step Log) 名稱，以及自身所代表的拼圖顏色
2. 該程式會依據棋盤目前狀態檔案名稱來讀取棋盤目前狀態檔案的內容，以得知目前的棋盤狀態
3. 該程式會依據剩餘拼圖檔案名稱來讀取剩餘拼圖檔案的內容，以得知目前自己所剩的可用拼圖
4. 依據自身所代表的拼圖顏色、目前棋盤狀態和剩餘可用的拼圖記錄、該程式會決定出下一個要擺放在棋盤上的拼圖以及棋擺放位置
5. 該程式會將拼圖擺放的決定，依格式更新棋盤目前狀態檔案以及剩餘拼圖檔案的內容，最後傳回 0 (return 0) 結束程式，或傳回 1 表示 PASS。

Game Rules

本節針對遊戲平台進行時的規則進行說明，以讓同學了解在遊戲平台中除了棋步規則之外的對戰規則（相關 Blokus Duo 遊戲棋步規則依之前發佈的 Project Introduction 文件為準）。

1. **PASS 後即無法繼續擺放拼圖**：玩家一旦 PASS，其後將不斷地 PASS 直到結束
2. **遊戲棋步思考時間為五秒鐘**：對戰過程中，若玩家 A 所撰寫的策略程式沒有在五秒內決定下一棋步並將下一個棋步的決定續寫至棋盤目前狀態檔案及剩餘拼圖檔案中，則判定玩家 A 決定 PASS
3. **不依規則進行拼圖擺放者視為落敗**：對戰過程中，若玩家 A 所撰寫的策略程式所決定的拼圖擺放違反棋步規則，則直接判定玩家 A 落敗
4. **任意修改棋盤目前狀態檔案或是剩餘拼圖檔案意圖影響棋局公平性者直接判定落敗**：對戰過程中，若玩家 A 竄改棋盤目前狀態檔案或是剩餘拼圖檔案意圖影響棋局公平性，則直接判定玩家 A 落敗

Overall Architecture



上圖呈現整體的 Blokus Duo 遊戲平台，**Control Engine** 負責整個遊戲進行的控制和遊戲規則的仲裁(此部分由助教提供)，而 **Strategy** 則為學生所撰寫的棋步策略。遊戲中，Control Engine 首先會要求在 TCF 上的 Strategy 進行下棋並提供給 Strategy 最新的棋盤目前狀態檔案 (**Blokus Board Status**)、剩餘拼圖檔案 (**Left Pattern**)以及棋步記錄檔案(**Step Log**、取得方式請參考第六頁相關說明)，Strategy 可依據這三種檔案所提供的資訊進行決策，並依據擺放的決定更新棋盤目前狀態檔案以及剩餘拼圖檔案(不用更新棋步記錄檔案)，再回傳 0 或 1 表示已完成下棋。完成下棋後 Control Engine 會檢視更新後的兩個檔案，並決定是否已分出勝負或是繼續進行遊戲。

Program Controlling Interface

同學們的程式在編譯過後，在 Blokus Duo 遊戲平台中會用以下指令在 command line 下控制或執行同學們的程式

```
[學號].out [Color] [Blokus Board Status] [Left Pattern] [Step Log]
```

- [Color]: 目前你所代表的顏色，可能為 *Red* 或是 *Blue*
- [Blokus Board Status]: 棋盤目前狀態檔案名稱
- [Left Pattern]: 剩餘拼圖檔案名稱
- [Step Log]: 棋步歷史紀錄檔案名稱

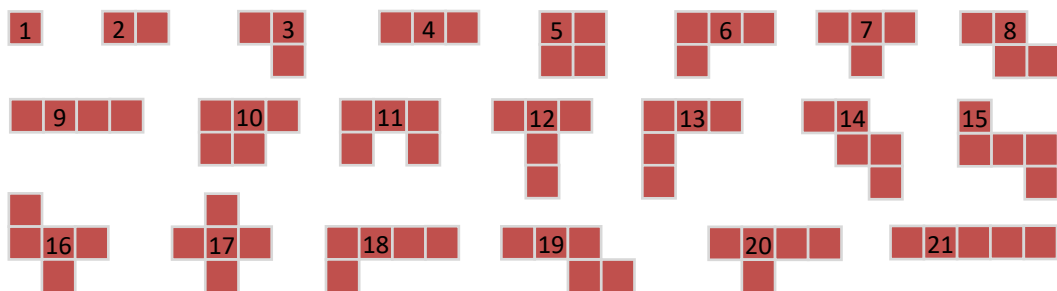
※註：除了 Main Control Interface 外，助教會在 checkpoint 1 和 checkpoint 2 定義其他的 testing/evaluation interface 以確保同學的程式階段性的符合助教所規定的 requirement，並協助同學掌握 project 的進度。相關的 testing/evaluation interface 會在之後的 checkpoint 規定文件中訂定。

Game Information Format

本節介紹遊戲進行時，遊戲平台與同學策略程式溝通時的各項資訊以及檔案格式。

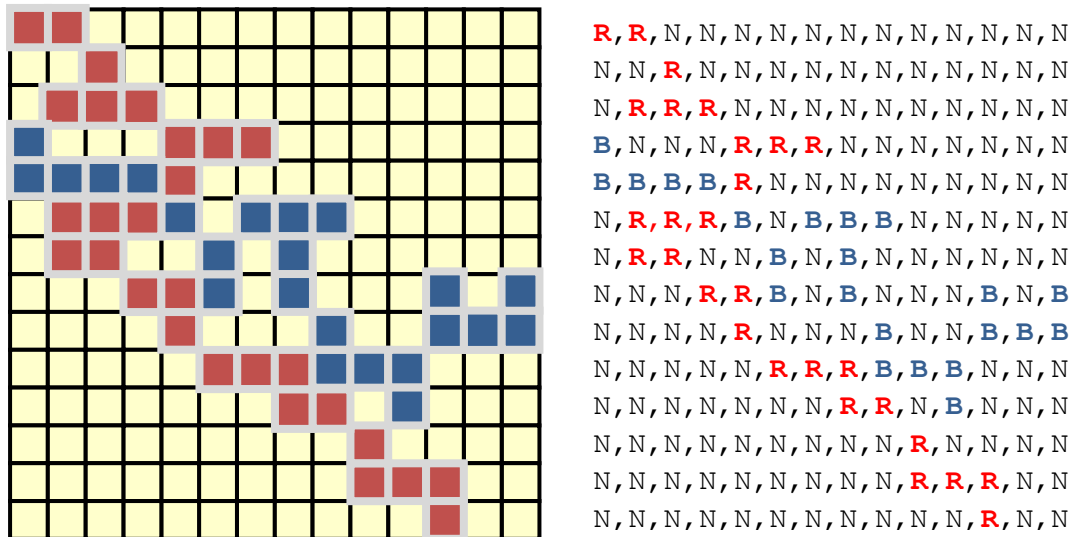
(1) 21 種拼圖的制訂與編號

在 Final Project 中，我們將 21 種拼圖予以編號，如下圖所示。在剩餘拼圖檔案中，我們將使用這些編號來代表每個拼圖。當拼圖被擺放到棋盤上時，是可以旋轉和翻面的。



(1) 棋盤目前狀態檔案格式 (Blokus Board Status)

Blokus Duo 棋盤規格是 14 x 14，在檔案 BlokusBoard.txt 中，我們利用一個 14 x 14 的矩陣來表示棋盤和棋盤上的狀態，下圖左方舉例呈現一個棋盤的狀態，而右方則表示相對應的 BlokusBoard.txt (右方檔案為方便說明，特別加上顏色，實際 BlokusBoard.txt 檔案呈現時並不會有文字顏色的不同)。



矩陣上的「N」代表尚未放上拼圖的格子，「R」代表 Red 的拼圖、「B」代表 Blue 的拼圖，這個矩陣的表示方法是在「期末 Blokus Duo 公開賽」中，做為彼此溝通的基礎，同學在程式內部可以另做資料結構的轉換，但是對於結果的呈現，請務必遵守此規範。

(2) 剩餘拼圖檔案格式 (Left Pattern)

檔案 LeftPattern_Color.txt 記錄特定顏色到目前為止剩下哪些拼圖，檔案利用一個序列代表剩子情況，序列位址代表該編號拼圖目前的狀態，「1」代表還沒有被擺到棋盤上；「0」代表已被擺到棋盤上。，以上圖棋盤為例，紅色玩家已依序將 6, 7, 2, 10, 3, 19, 15 號拼圖置於棋盤上，則 LeftPattern_Red.txt 檔案內容如下圖表示 (由上至下的每一行依序代表每次決定後剩下的拼圖)

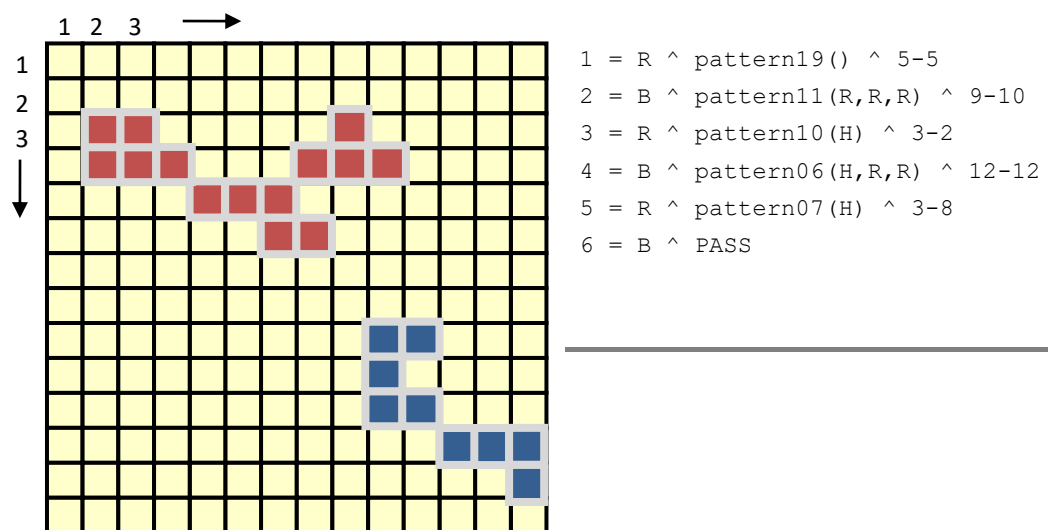
```
Player = E24980000
[1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1]
[1,1,1,1,1,0,1,1,1,1,1,1,1,1,1,1,1,1]
[1,1,1,1,1,0,0,1,1,1,1,1,1,1,1,1,1,1]
[1,0,1,1,1,0,0,1,1,1,1,1,1,1,1,1,1,1]
[1,0,1,1,1,0,0,1,1,0,1,1,1,1,1,1,1,1]
[1,0,0,1,1,0,0,1,1,0,1,1,1,1,1,1,1,1]
[1,0,0,1,1,0,0,1,1,0,1,1,1,1,1,1,0,1]
[1,0,0,1,1,0,0,1,1,0,1,1,1,1,0,1,1,1]
```

Player 將記錄執紅色拼圖玩家的學號，第二行開始的序列，由上至下每一行依序代表每次進行決定後，所剩下尚未置於棋盤上的拼圖。每次輪到紅色玩家時，其程式須讀取最後一行剩子情況，從還沒有擺放的拼圖中擇一放入，並將新的剩餘拼圖狀況續寫 (append) 回 LeftPattern_Red.txt 的最後一行。

(3) 棋步歷史紀錄檔案格式 (Step Log)

Step Log 將依序記錄每一位玩家擺放 Pattern 的編號以及擺放的位置於 StepLog.txt 中，StepLog.txt 會被 Control Engine 存放於 strategy 執行目錄的母資料夾中，同學可透過開啟 ../StepLog.txt 來讀取棋步歷史紀錄檔案。

下圖列出一個 StepLog.txt 的檔案內容做為例子，說明該檔案內容格式。



StepLog.txt 中的每一行代表每一位玩家所進行的動作，第一個數字為動作的順序，等號後接續的英文字母為玩家顏色代號 (R = 紅色, B = 藍色)，接著是擺放拼圖的編號以及其上下翻轉 (H)、順時鐘旋轉 (R) 的次數，最後是拼圖左上角擺放於棋盤的位置 (第一個數字為 y 軸位置, 第二個數字為 x 軸位置)。以檔案第 4 行為例，它代表了藍色玩家的動作，藍色玩家選擇放置 6 號拼圖，並將 6 號拼圖上下翻轉了 1 次，順時鐘旋轉 2 次，並將拼圖左上角放置於棋盤上 12-12 的位置。再舉第 5 行為例，紅色玩家上下翻轉 7 號拼圖，並將左上角放置於 3-8 的位置。

同學們可以利用此檔案瞭解對手擺放棋子的記錄、幫助自己調整擺放棋子的策略，但是此檔案將由助教的程式管理，同學們不需寫入任何資料至此。

Project 進行建議

1. 請及早進行 Project 的規劃及實作

雖然目前助教仍未講授 C 語言的 file I/O，但此次 project 的主要核心絕非 file I/O 部分，而是如何依據現有棋盤狀態，想出最好的棋步決策方式，並設計演算法讓電腦能夠幫忙進行運算，以獲得最後的勝利。在這樣的思維中，file I/O 以及 command line argument 皆為程式與遊戲平台的溝通介面且不會影響到決策的過程(亦即本 project 的主要核心)。

在 project 進行初期，同學現在已可以開始思考並實作棋步的決策演算法，並先自訂介面(同學可先利用 command line interface 做為程式的輸入以及輸出)來確保自己程式在決策上的正確性，同學亦可以先自行實作整體遊戲邏輯用以進行棋局策略的模擬。等之後同學瞭解如何使用 C 進行 file I/O 後，再將原先自訂介面切換為 file I/O 介面即可。

因為 project 的時程頗長且 project 的 load 較重，請同學儘早規劃 project 的進行時程，不要將 project 拖在 checkpoint deadline 或是 project deadline 前才開始進行，以免無法順利完成 project。

2. 請遵守 Project Checkpoint 所規定的進度

之後 release 的 project checkpoint 會在 project 進行中評量同學們的階段性程式碼是否有達成助教所規劃的進度，為了確保各位同學都能完成 project，checkpoint 的評量成果將會直接納入 project 成績的計算，因此請同學遵守 project checkpoint 的規劃，並在 project checkpoint deadline 前完成 checkpoint 中所規定的事項，以確保 project 成功。

3. 同學可以自訂資訊存取格式以更有效的管理遊戲資訊

Blokus Duo 遊戲平台不會限制策略程式在進行棋步決策時是否會產生或存取其他的檔案，但是遊戲平台只會對預設提供的三種檔案進行更新，若同學有定義遊戲資訊檔案來自行管理遊戲資訊，則必須自行從更新後的棋盤目前狀態檔案以及棋譜檔案中擷取想要的資訊後，再自行更新自身所定義的遊戲資訊檔案。