

Client 使用說明

陳志昌、許嘉銘、鄧鈞洲、鍾毓湘

December 17, 2019

前言

此程式主要目的是在於讓棋類程式開發者能有一個公開的平台對弈，互相測試棋力亦或是可以測試程式各版本間棋力的差異性，另外提供了許多方便的功能來減輕程式開發者的負擔。

- 基本功能使用：請詳閱 Section 1、2、3、4、8。
- 進階使用：請詳閱 Section 5、6、7。

Contents

1	檔案說明	2
2	環境設置	3
3	開啟方式	4
4	遊戲設定	5
5	房間介紹	10
6	盤面介紹	14
7	功能介紹	20
8	通訊協定	24
9	參考資料與聯絡資訊	40

1 檔案說明

1.1 資料夾

名稱	修改日期	類型	大小
Board	2019/8/31 下午 07:03	檔案資料夾	
Library	2019/8/31 下午 07:31	檔案資料夾	
LocalServer	2019/8/31 下午 06:29	檔案資料夾	
Search	2019/6/17 下午 08:09	檔案資料夾	
Setting	2019/8/31 下午 07:31	檔案資料夾	
WebServer	2019/8/31 下午 06:29	檔案資料夾	
Launcher.jar	2019/8/31 下午 07:42	Executable Jar File	329 KB

Figure 1: Client 檔案結構

- Board：遊戲的盤面紀錄檔案
- Library：遊戲的函式庫
- LocalServer：遊戲的本地伺服器
- Search：AI 程式檔案
- Setting：遊戲設定紀錄檔、房間設定紀錄檔
- WebServer：網頁伺服器
- Launcher.jar：Client 啟動器本體

注意：切勿隨便更動資料夾結構，會造成程式路徑錯誤而無法執行。

2 環境設置

完成環境建置之後，才能順利執行程式。

2.1 Java

程式為由 Java 撰寫而成，因此執行時需要 JVM，所以必須先安裝 jre。以下介紹 Windows 和 Ubuntu 安裝 java 8 的方法。

注意：此程式最低執行 Java 版本為 **Java 8**。

2.1.1 Windows

到官網下載並安裝 jre ([下載頁面連結](#))。

2.1.2 Ubuntu

開啟 Terminal 輸入以下指令

Ubuntu

```
sudo apt-get update  
sudo apt-get install openjdk-8-jdk
```

2.2 Browser

建議使用 Google Chrome 瀏覽器。

2.2.1 Windows

到官網下載並安裝 google chrome 瀏覽器 ([下載頁面連結](#))。

2.2.2 Ubuntu

開啟 Terminal 輸入以下指令

Ubuntu

```
wget https://dl.google.com/linux/direct/google-chrome-stable  
_current_amd64.deb  
  
sudo dpkg -i google-chrome-stable_current_amd64.deb
```

3 開啟方式

程式的 User Interface 分為 **GUI** (Graphical User Interface) 和 **CLI** (Command-Line Interface) 兩種，以下分別介紹兩種的使用方式。另外，在 Windows 和 Ubuntu 上使用方法相同。

因為 Ubuntu 有權限問題，所以需要給予執行權限。

Ubuntu

```
cd <the_path_to_ubuntu>  
chmod a+x -R ubuntu/
```

3.1 GUI

- 方法 1：雙擊檔案 Launcher.jar
- 方法 2：開啟 Terminal 輸入以下指令

Windows / Ubuntu

```
cd <the_path_to_Launcher.jar>  
java -jar Launcher.jar
```

3.2 CLI

開啟 Terminal 輸入以下指令

Windows / Ubuntu

```
cd <the_path_to_Launcher.jar>  
java -jar Launcher.jar -cli
```

4 遊戲設定

此節會介紹 Figure 2 中各個遊戲參數的欄位內容及意義。

Launcher Ver.5.2.1

Game Room Other

遊戲種類：DarkChess 遊戲版本：Ver.1.3.3 程式版本：Ver.5.2.1

帳號：Yahari 計時模式：Absolute

密碼：Yahari 秒數限制：900 (sec)

啟動模式：☒ 開房 ☐ 加房 ☐ 本地測試 單手制時間：0

斷線中盤：☐ 是 ☒ 否 ☐ 選擇盤面 幾手共用：0

連接模式：☐ 讀檔 ☐ 背景 ☒ DCTP ☐ 手動 長捉次數：3

重複次數：2 無吃翻次數：180

先手後手：☒ 先 ☐ 後

先後互換：☒ 是 ☐ 否 盤面路徑：選擇

Server IP：120.126.151.213 AI路徑：W_CDC_CLIENT\DCTP\dctp.exe 選擇

連線

Figure 2: GUI Client 設定頁面

- 遊戲種類 (Game Type)：此平台會支援多種遊戲，可以由此選項來選擇所需的遊戲類別。
- 帳號 (Account)：玩家帳號，需向管理者申請。**guest 帳號為 a0 至 a2000。**
- 密碼 (Password)：玩家密碼。**guest 帳號的密碼皆為 123。**
- 啟動模式 (Start Mode)：玩家選擇要以開房模式或加房模式進行遊戲。
 - 開房 (Open)：創建房間讓其他使用者加入。
 - 加房 (Enter)：加入其他使用者所創建的房間。
 - 本地測試 (Local Server)：本程式連接至本地伺服器，而非遠端伺服器。
- 斷線中盤 (MidBoard)：由開房模式的玩家進行遊戲**初始盤面**的設定。
 - 是 (Yes)：遊戲初始盤面由 Server 尋找該玩家**當日中盤斷線**的紀錄，以**最後的盤面**繼續進行遊戲。
 - 否 (No)：遊戲初始盤面為**預設初始盤面**，由 Server 端提供。
 - 選擇盤面 (Custom)：遊戲初始盤面為**盤面路徑欄位**所指向的檔案。玩家可透過此選項選擇中盤檔案進行中盤對下。
- 連接模式 (Connect Mode)：指 Client 端與程式的連接協定。
 - DCTP：使用 DCTP 協定。

- 背景 (Ponder)：使用背景協定。
- 讀檔 (ReadFile)：使用讀檔協定。
- 手動 (Human)：不使用程式，由玩家手動進行下棋。當雙方皆選擇手動模式，在遊戲中可以提和；任一方選擇手動模式在遊戲中可以認負、Abort。
- 重複局數 (Repeat Times)：此房間總共的對弈局數。
- 先手後手 (Host First Move)：開房玩家第一局為先手或後手。
- 先後互換 (Change First Move)：在每一局結束後，雙方玩家是否交換先後手。
- Server IP：要連接的 Server IP。預設為120.126.151.213。
- 計時模式 (Timer Mode)：將常見的計時模式統合起來，方便快速設定秒數限制、單手制時間和幾手共用欄位。
 - None：不設定時間限制，將不對遊戲時間進行限制。
 - Custom：由玩家自由設定秒數限制、單手制時間、幾手共用。
- 單手制時間 (One Ply Time)：玩家的思考時間 (即自由時間) 用完之後，要給予多少額外的思考時間。
- 幾手共用 (Ply Share)：在限時內，玩家必須下幾手。
- 長捉次數 (Long Catch)：最多盤面重複次數，超過即依棋規判定勝負。例如：暗棋作和、象棋長將作負。

長捉次數設為 3 b2-b1 為走子步	b2-b1, a1-a2, b1-b2, a2-a1, b2-b1, a1-a2, b1-b2, a2-a1, b2-b1, a1-a2, b1-b2, a2-a1	當走步如左， 稱為 4 步一循環， 循環三次，判和。
長捉次數設為 3 b2-b1 為吃子步	b2-b1, a1-a2, b1-b2, a2-a1, b2-b1, a1-a2, b1-b2, a2-a1, b2-b1, a1-a2, b1-b2, a2-a1	當走步如左， 因 b2-b1 為吃子步， 所以尚未達成三次循環。
長捉次數設為 3 b2-b1 為吃子步	b2-b1, a1-a2, b1-b2, a2-a1, b2-b1, a1-a2, b1-b2, a2-a1, b2-b1, a1-a2, b1-b2, a2-a1, b2-b1	當走步如左， 稱為 4 步一循環， 循環三次，判和。

- 無吃翻次數 (No Eat Flip)：最多無吃翻次數，超過即判和。
- 盤面路徑 (Board Path)：斷線中盤欄位選擇“選擇盤面”選項時，Client 端會到此路徑讀取中盤檔案，並傳送給伺服器當作起始盤面。

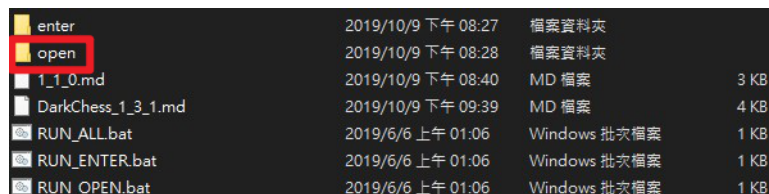
建議：使用絕對路徑。

- AI 路徑 (AI Path)：Client 端會到此路徑啟動電腦對局程式，並以連接模式中選擇的連接協定進行連接。

建議：使用絕對路徑。

4.1 開房

開房玩家需要設定以下的遊戲參數：斷線中盤、連接模式、重複次數、先手後手、先後互換、Server IP、計時模式 (秒數限制、單手制時間、幾手共用)、長捉次數、無吃翻次數、盤面路徑、AI 路徑。



enter	2019/10/9 下午 08:27	檔案資料夾	
open	2019/10/9 下午 08:28	檔案資料夾	
1_1_0.md	2019/10/9 下午 08:40	MD 檔案	3 KB
DarkChess_1_3_1.md	2019/10/9 下午 09:39	MD 檔案	4 KB
RUN_ALL.bat	2019/6/6 上午 01:06	Windows 批次檔案	1 KB
RUN_ENTER.bat	2019/6/6 上午 01:06	Windows 批次檔案	1 KB
RUN_OPEN.bat	2019/6/6 上午 01:06	Windows 批次檔案	1 KB

Figure 3: 開房資料夾

如 Figure 3 所示，進入 **open** 資料夾，執行 Launcher.jar (開啟方式參見 Section 3)。

4.1.1 GUI

在 Figure 2 啟動模式選擇**開房**，設定**遊戲參數**及**玩家資料**，然後**按下連線按鈕**與伺服器連線。

4.1.2 CLI

```
D:\Eclipse\workspace\number\out\open>java -jar Launcher.jar -cli
loading Library --- OK
loading GameType Setting File --- OK
loading Game Setting File --- OK

*** Launcher V5.2.1 ***
1) Show Settings
2) Change Game Type
3) Change Game Setting
4) Start Game
5) Quit
Please enter a number to continue: 3

*** Game Setting ***
1) Account [Yahari]
2) Password [Yahari]
3) Start Mode [Open]
4) LocalServer [No]
5) MidBoard [No]
6) Connect Mode [Ponder]
7) Repeat Times [2]
8) Host First Move [Yes]
9) Change First Move [Yes]
10) Server IP [120.126.195.84]
11) Timer Mode [Absolute (900/0/0)]
12) Long Catch [3]
13) No Bat Flip [180]
14) Search Path [Search\Search.exe]
15) Back
Please enter a number to continue:
```

Figure 4: CLI Client 開房設定頁面

```
D:\Eclipse\workspace\number\out\open>java -jar Launcher.jar -cli
loading Library --- OK
loading GameType Setting File --- OK
loading Game Setting File --- OK

*** Launcher V1.1.0 ***
1) Show Settings
2) Change Game Type
3) Change Game Setting
4) Start Game
5) Quit
Please enter a number to continue:
```

Figure 5: CLI Client 主頁面

Figure 4 為 CLI 開房選項範例。使用方法為：

1. 輸入 3 設定遊戲參數及玩家資料。
2. 回到主頁面 (Figure 5)，輸入 4 與伺服器連線。

4.2 加房

加房模式需要設定連接模式、Server IP、盤面路徑、AI 路徑。加房玩家只需要設定玩家資料即可，由開房玩家設定遊戲參數。

enter	2019/10/9 下午 08:27	檔案資料夾	
open	2019/10/9 下午 08:28	檔案資料夾	
1_1_0.md	2019/10/9 下午 08:40	MD 檔案	3 KB
DarkChess_1_3_1.md	2019/10/9 下午 09:39	MD 檔案	4 KB
RUN_ALL.bat	2019/6/6 上午 01:06	Windows 批次檔案	1 KB
RUN_ENTER.bat	2019/6/6 上午 01:06	Windows 批次檔案	1 KB
RUN_OPEN.bat	2019/6/6 上午 01:06	Windows 批次檔案	1 KB

Figure 6: 加房資料夾

如 Figure 6 所示，進入 enter 資料夾，執行 Launcher.jar (開啟方式參見 Section 3)。

4.2.1 GUI

在 Figure 2 啟動模式選擇加房，設定玩家資料，然後按下連線按鈕與伺服器連線。

4.2.2 CLI

```
D:\Eclipse\workspace\number\out\enter>java -jar Launcher.jar -cli
loading Library --- OK
loading GameType Setting File --- OK
loading Game Setting File --- OK

*** Launcher V5.2.1 ***
1) Show Settings
2) Change Game Type
3) Change Game Setting
4) Start Game
5) Quit
Please enter a number to continue: 3

*** Game Setting ***
1) Account [PupilDarkChess]
2) Password [PupilDarkChess]
3) Start Mode [Enter]
4) LocalServer [No]
5) Connect Mode [Dctp]
6) Server IP [120.126.195.84]
7) Search Path [D:\Eclipse\workspace\number\out\enter\Search\Search.exe]
8) Back
Please enter a number to continue: _
```

Figure 7: CLI Client 加房設定頁面

Figure 7 為 CLI 加房選項範例。使用方法為：

1. 輸入 3 設定玩家資料。
2. 回到主頁面 (Figure 5)，輸入 4 與伺服器連線。

5 房間介紹

5.1 房間設定

Figure 8: 房間頁面設定

- 盤面開啟方式：設定盤面檔案開啟方式。
 - 預設瀏覽器：由 Client 負責開啟預設網頁瀏覽器，來顯示盤面。
 - 遠端：由遠端連接 Client 來顯示盤面。
 - 不開啟：不顯示任何盤面。
- 輸出視窗捲軸：設定輸出視窗的捲軸參數。
 - 自動捲置底部：只要內容有更新，自動捲動至底部。
 - 不動作：不管內容是否更新，不移動捲軸位置。
- 輸出視窗背景顏色：可以更改輸出視窗背景顏色。
- 輸出視窗字體顏色：可以更改輸出視窗字體顏色。
- 輸出視窗字體大小：可以更改輸出視窗字體大小。
- 重新遊戲延遲時間：在每局遊戲之間的延遲時間。
- 時間自動倒數：玩家剩餘時間顯示功能參數設定。
 - 開啟：會自動倒數玩家剩餘時間。
 - 關閉：不會自動倒數玩家剩餘時間，只會在每手接收 Server 的剩餘時間並更新。
- 清除最後一場遊戲輸出和盤面：在此房間所設定的對弈局數結束之後，是否要清除輸出視窗和盤面顯示。
- 遊戲完成後自動關閉網頁：在此房間所設定的對弈局數結束之後，是否要關閉網頁。

5.2 房間頁面

5.2.1 遊戲參數

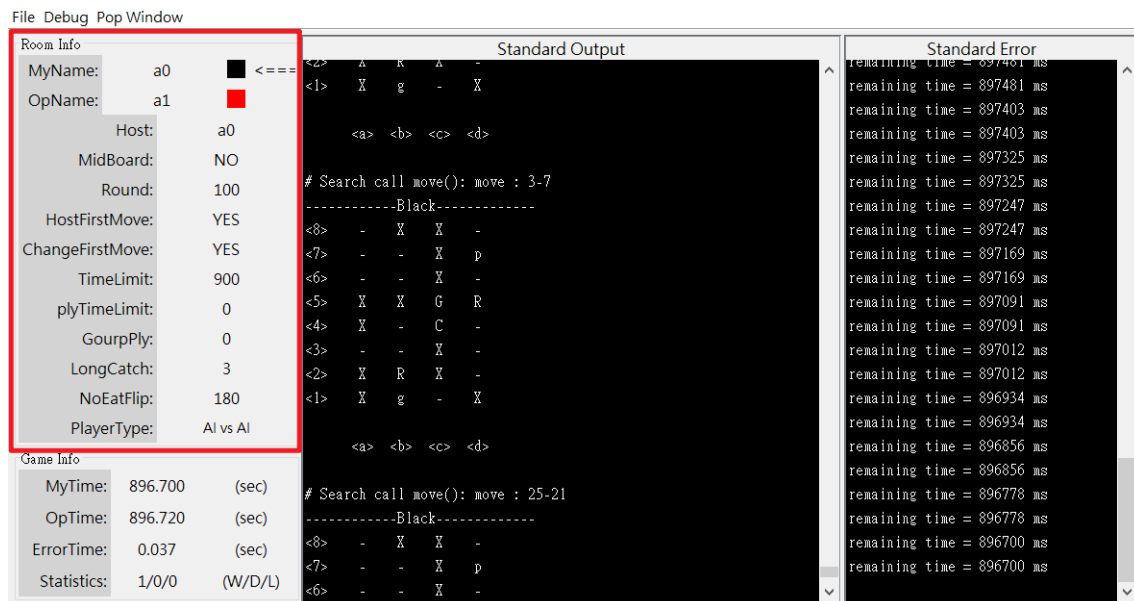


Figure 9: 遊戲參數顯示區域

在 Figure 9 的紅色方框中，上方為我方和對手的帳號名稱、持子顏色，和箭頭來指示輪到哪位玩家，下方為此房間內遊戲的各項參數。

5.2.2 遊戲狀態

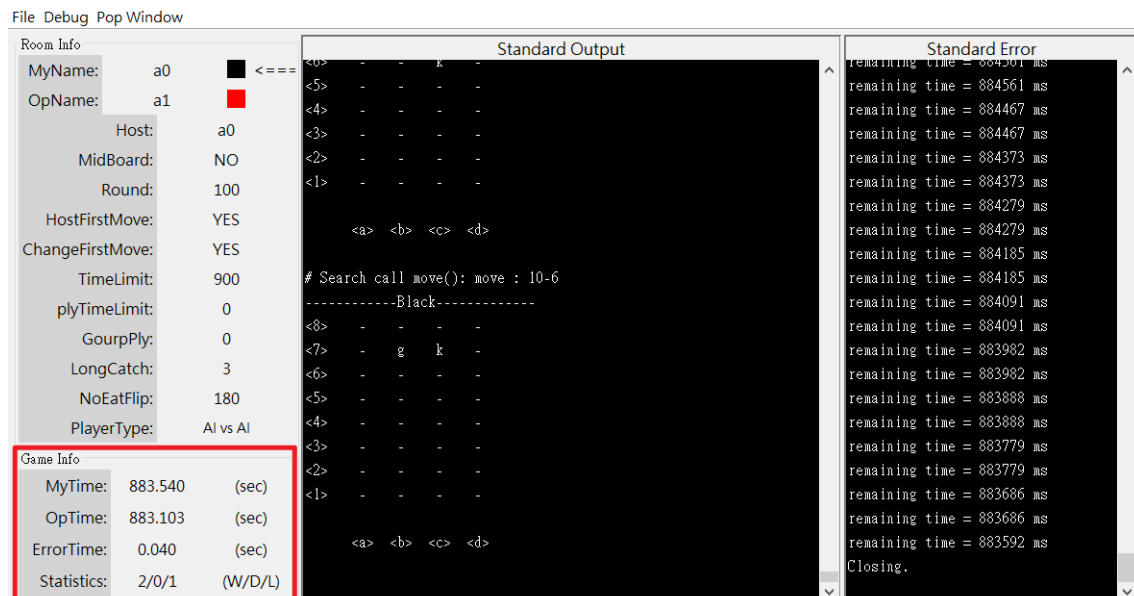


Figure 10: 遊戲狀態顯示區域

在 Figure 10 的紅色方框中，顯示雙方的剩餘時間，傳輸的誤差時間和勝負和局統計。

- MyTime：我方剩餘時間。
- OpTime：對方剩餘時間。
- ErrorTime：傳輸誤差時間。讓 X 和 Y 分別表示 Server 和 Client 紀錄的 AI 下一手所花費時間。計算公式為：

$$ErrorTime = \frac{(X - Y)}{2}$$

非真實網路延遲時間，但可作為網路穩定度參考。

- Statistics：勝/和/負的局數。

5.2.3 輸出視窗

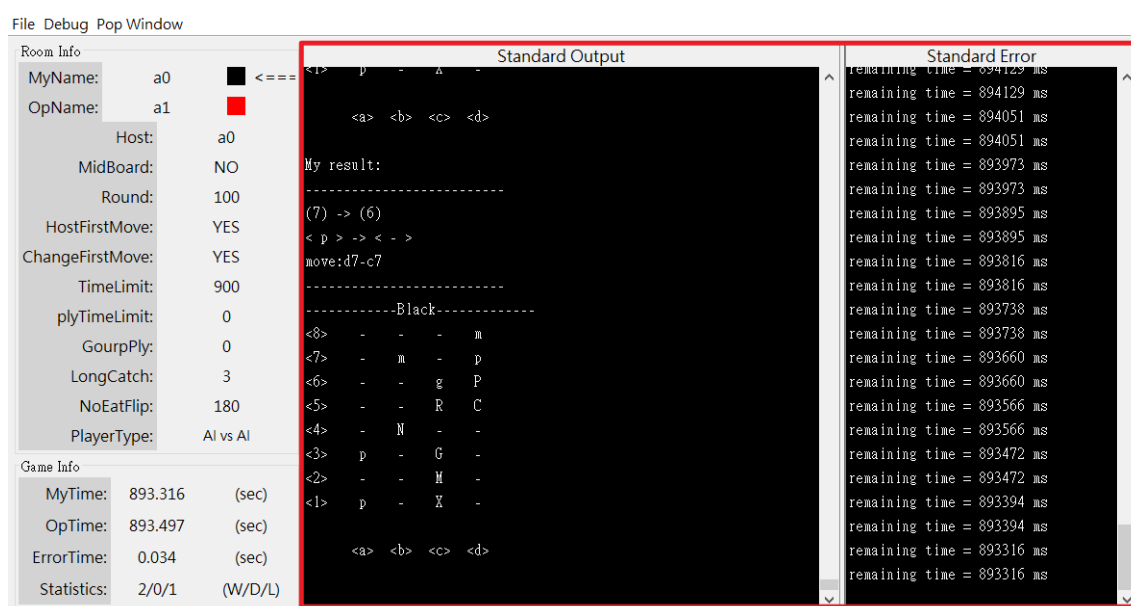


Figure 11: 程式輸出區域

在 Figure 11 的紅色方框中，左邊區塊顯示的是程式的標準輸出 (stdout)，右邊區塊則是標準錯誤輸出 (stderr)。

5.2.4 功能選單

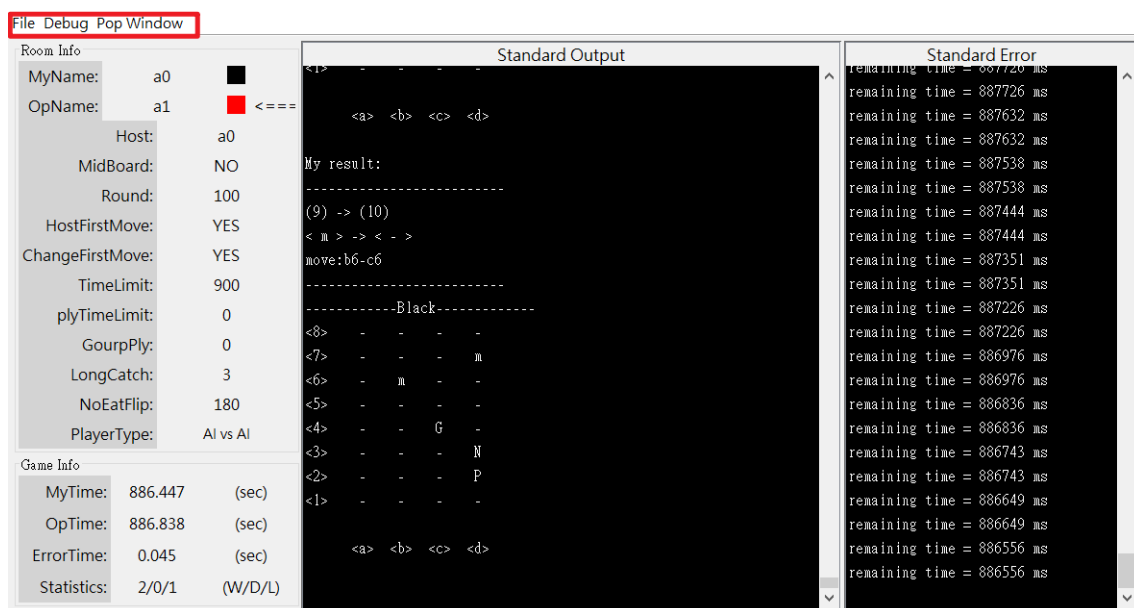


Figure 12: 功能選單區域

Figure 12 的紅色方框為功能選單。

- File
 - Room Statistics：單一房間內所有棋局的統計資料，參見 Section 7.7。
 - Game Record：單一棋局的棋局內容，參見 Section 7.8。
- Debug
 - AutoScroll：內容更新時，是否自動捲動捲軸到底部。
 - Increase Font Size：增加 Debug 區域字體大小。
 - Decrease Font Size：減少 Debug 區域字體大小。
- Pop Window
 - Pop Debug Window：將 Debug 區域的視窗獨立出來成一個頁面，方便使用者除錯。

6 盤面介紹

目前棋盤以網頁的方式呈現，以增加相容性，盤面選項使用預設瀏覽器的情況下，Client 會自動幫使用者開啟盤面檔案並連接，以下介紹盤面上的各個區塊和功能。

建議：使用 Google Chrome 瀏覽器。

6.1 遊戲盤面

- 開啟網頁後，會顯示棋子狀態、棋盤、歷史走步、與結果，如 Figure 13 所示。
- 結果區塊在“手動模式”下顯示是否為我方回合。

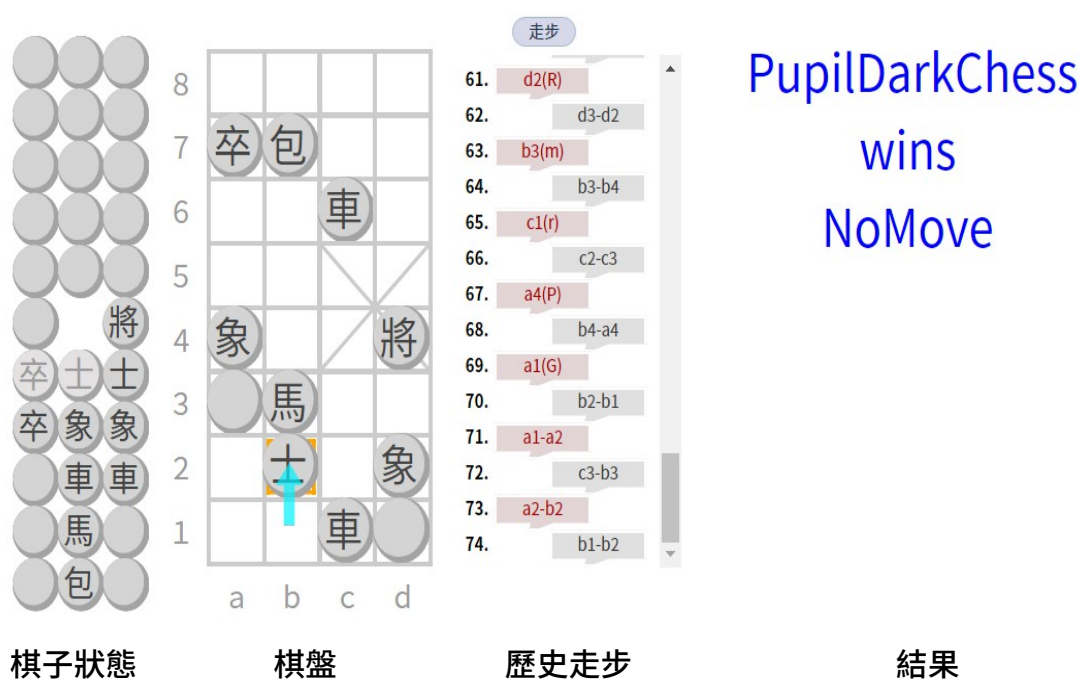


Figure 13: 顯示頁面

6.1.1 棋子狀態

- 網頁中左方會顯示目前棋子狀態，如圖 Figure 14 所示。



Figure 14: 棋子狀態

6.1.2 棋盤顯示

- 手動移動點選棋子，棋子底部會呈綠色，可移動的位置會以淺藍色方格顯示，如 Figure 15(a) 所示。
- 收到走步指令棋會從原本的位置延伸箭頭至目的地，目的地以橘色方格顯示，如 Figure 15(b) 與 Figure 15(c) 所示。
- 收到翻棋指令所翻開的棋子位置會有深藍色方格，如 Figure 15(d) 所示。

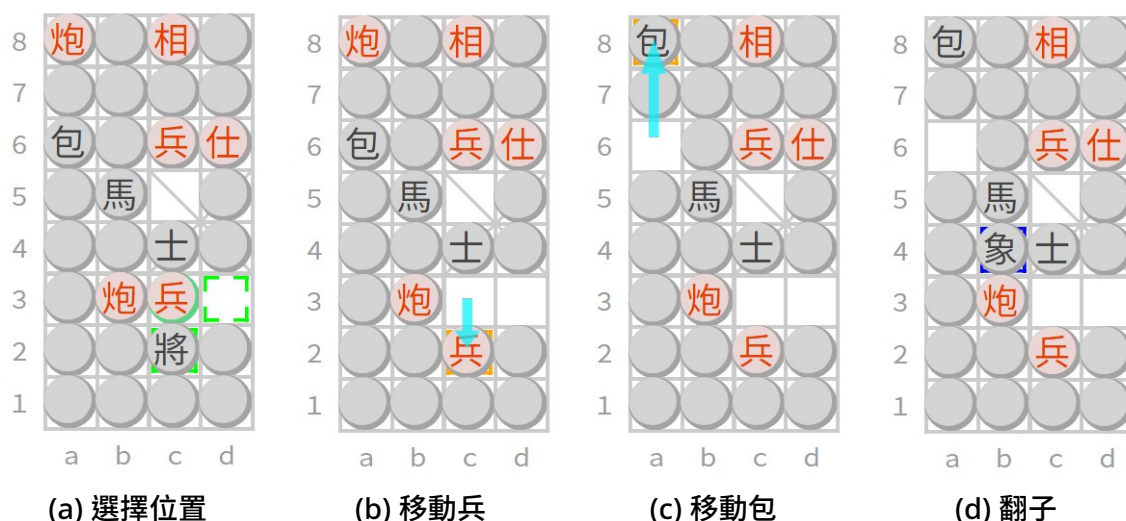


Figure 15: 棋盤顯示走步

6.1.3 手動模式

- 輪到我方，可提和、認負與悔棋，如 Figure 16(a)(b)(c)。
- 悔棋 (undo) 可利用歷史走步選擇回到哪一步，如 Figure 16(d)。
- 在手動模式中，結果的區塊會顯示 “Your turn” 表示輪到我方，字的顏色表示我方顏色。以暗棋為例，若尚未決定玩家持子的顏色，則 “Your turn” 顯示藍色，如 Figure 17(a) 所示；否則，則顯示紅色或黑色，如 Figure 17(b) 和 Figure 17(c) 所示。

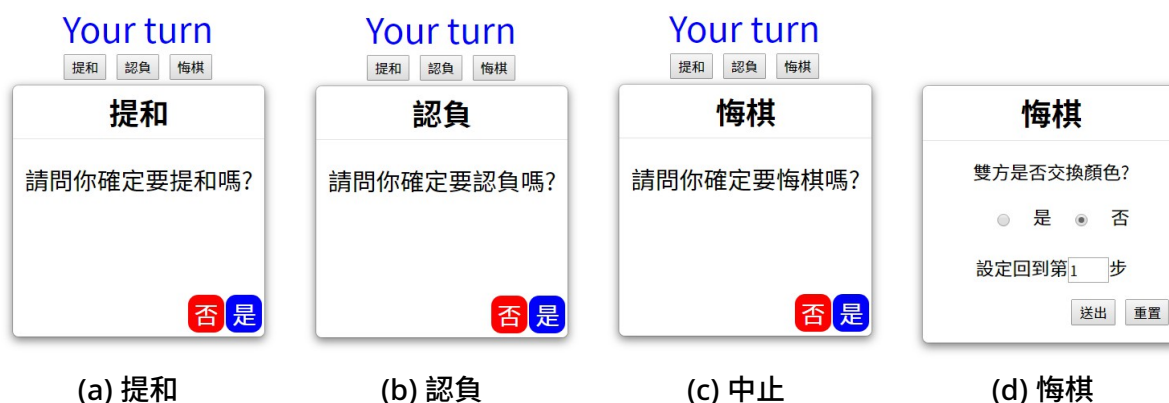
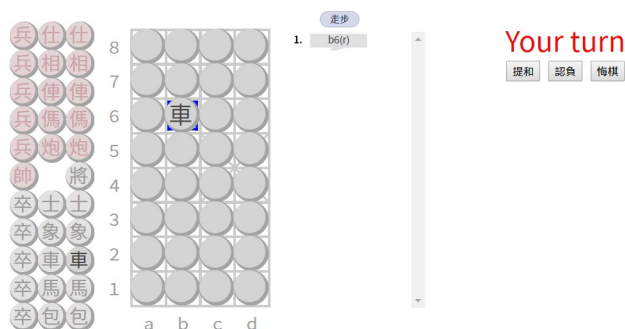


Figure 16: 輪我方時的選項



(a) 初始棋盤



(b) 紅方



(c) 黑方

Figure 17: 手動模式顏色說明

6.2 遊戲紀錄

依照 Section 7.8 的步驟進入遊戲紀錄。

6.2.1 使用說明

- 開啟頁面後，請選擇歷史檔案，如 Figure 18 所示。
- 在此頁面按鍵盤 'w', 's', 'a', 'd'，可控制走步往前或往後一手。
- 開啟檔案的先後手名稱與顏色，如 Figure 19 所示。歷史盤面如 Figure 20 所示。
- 可選擇開啟其它 board 檔，文字框輸入要儲存中盤檔名。上方文字框為使用者標準輸出，下方文字框為使用者標準錯誤輸出，會根據走步換行，預設是會自己換行，如果不想要換行可自行關閉，如 Figure 21 所示。

6.2.2 製作中盤檔案

- 製作中盤對下 (Section 7.5) 所需要的中盤檔案，檔案會記錄從起始盤面到選定盤面之間的所有走步。
- 使用方法為：如 Figure 22 所示。
 1. 選擇走步：請直接點擊走步來選擇中盤的最後一步。
 2. 輸入檔名：請輸入中盤的檔案名稱。
 3. 按下按鈕：請點擊 save 的按鈕來下載檔案。

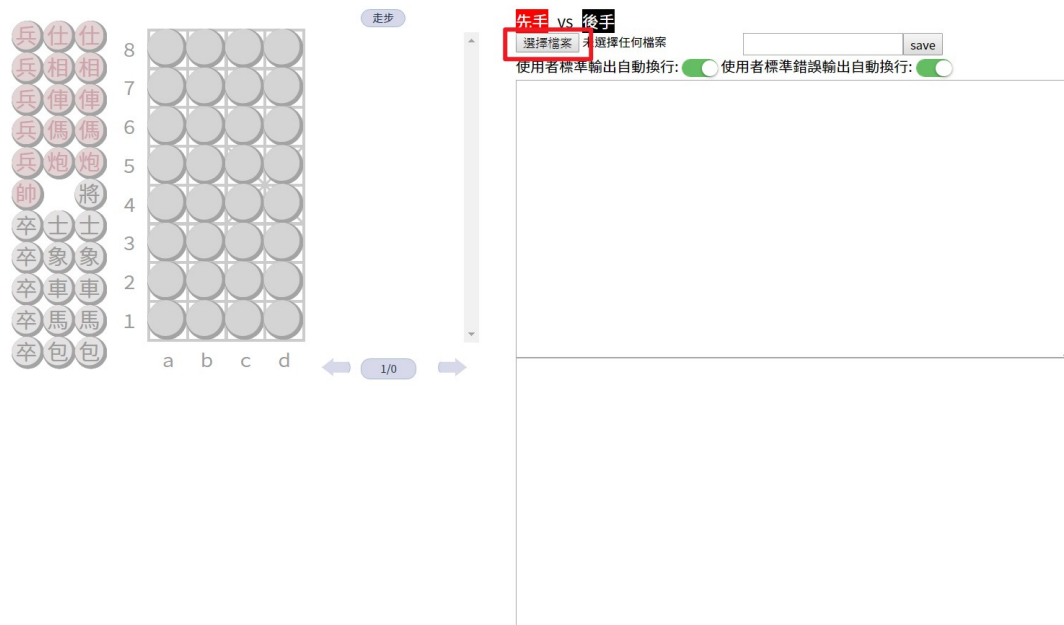


Figure 18: 顯示頁面



Figure 19: 先手 YAHARI 與後手 PUPILDARKCHESS

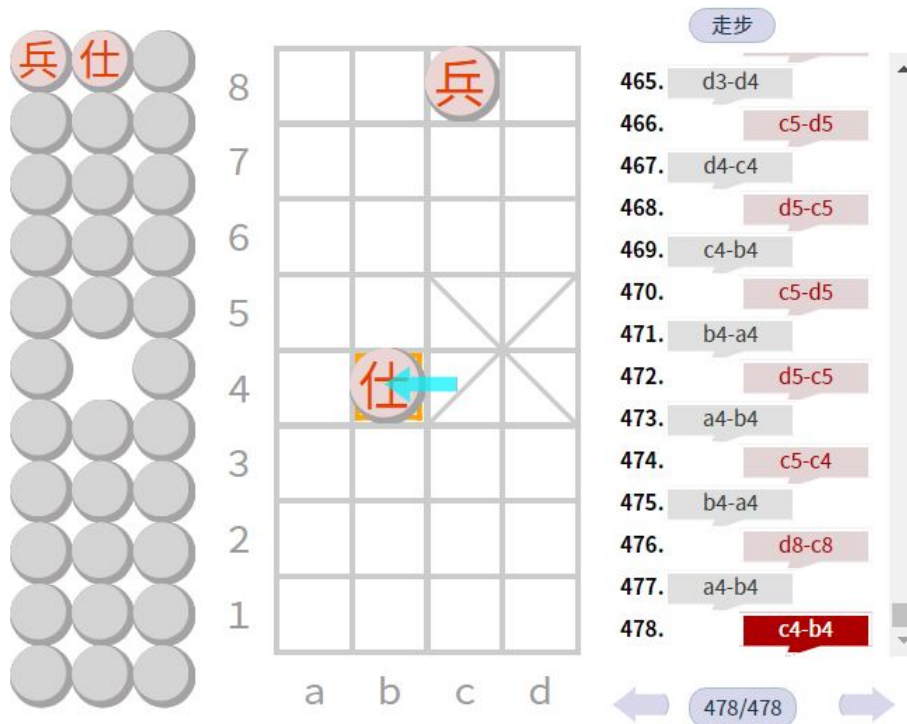


Figure 20: 盤面與歷史走步狀態

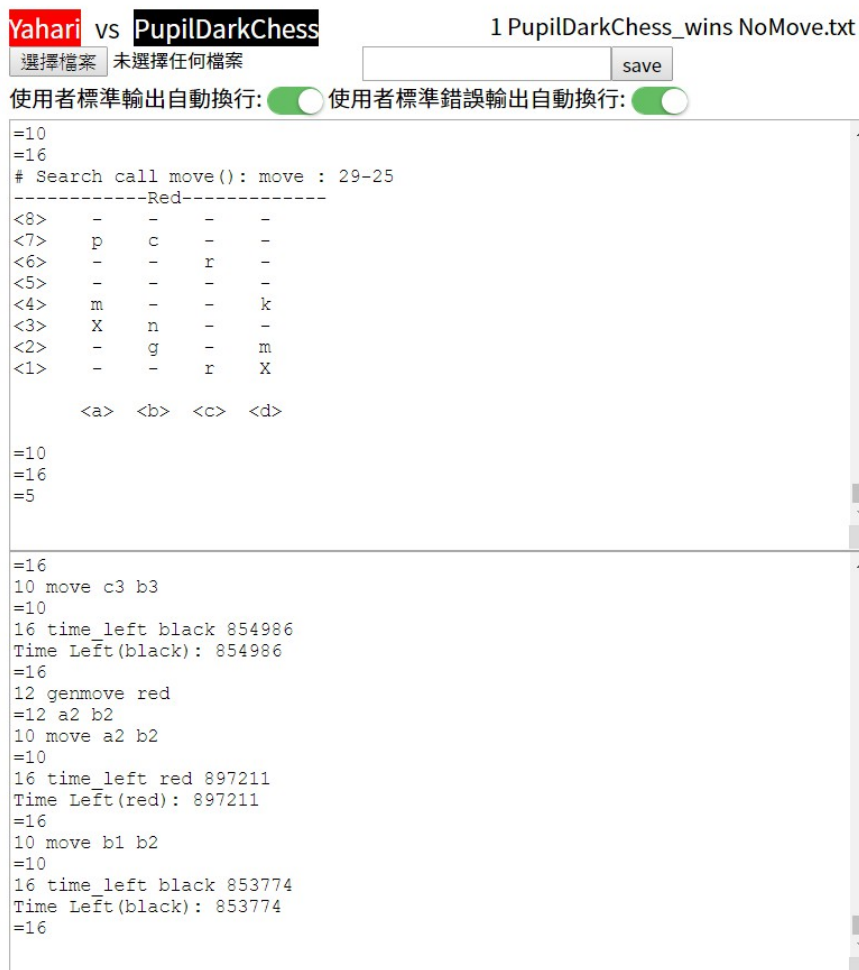


Figure 21: 使用者標準輸入與標準錯誤輸出

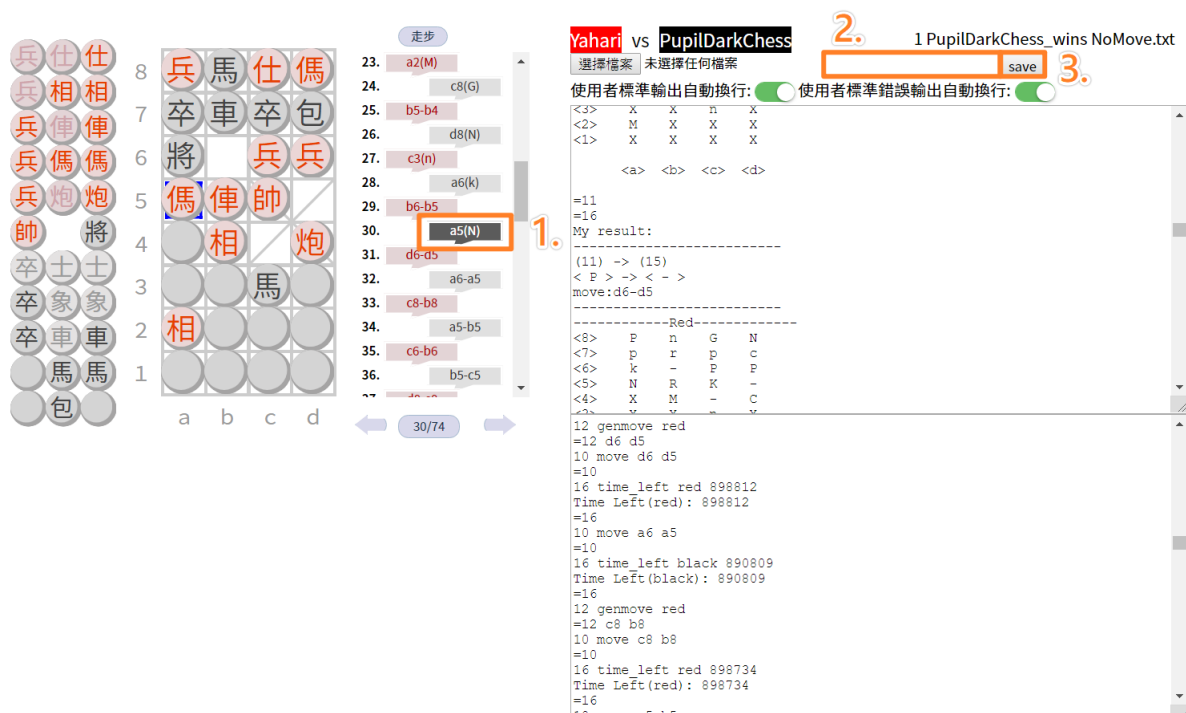


Figure 22: 製作中盤檔案步驟

6.3 遠端頁面

- 假設 Client 的 IP 是 120.126.151.3。在開房或加房之後，Client 端會出現 Figure 23 的介面。紅色方框顯示 Client 端的 Port，假設 Port 是 12079。
- 在觀戰的電腦上依照 Section 7.9 選擇遠端頁面，即出現 Figure 24 的介面。然後輸入 Client 的 IP: 120.126.151.3 和 Port: 12079 即可觀戰。

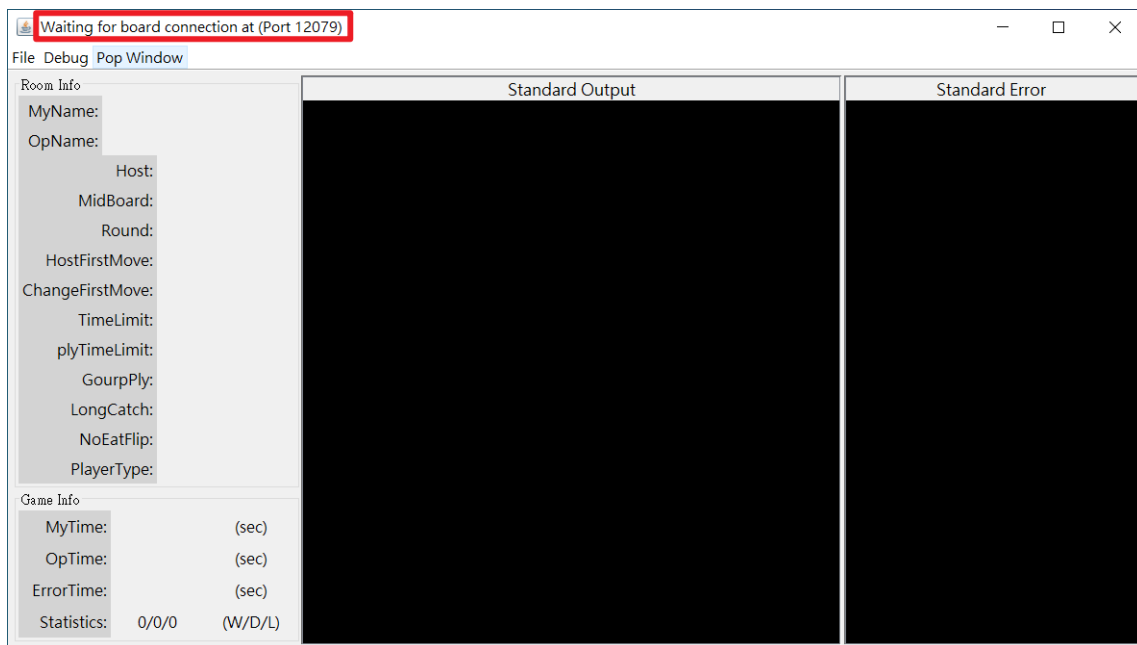


Figure 23: Port 顯示頁面

請輸入IP: PORT:

Figure 24: 遠端頁面

7 功能介紹

7.1 認負

當遊戲局勢大致已定的情況下，有時候會因為未達勝負條件，所以遊戲必須進行，造成時間上的浪費。此功能讓玩家或 AI 可以提前認負 (Resign) 結束比賽。方式分為兩種：

- 手動 (定義在 Section 6.1.3)
- AI (定義在 Section 8)

注意：認負功能在**所有模式**下皆可使用。

認負	線上伺服器	本地測試
AI vs AI	Yes	Yes
AI vs Human	Yes	Yes
Human vs Human	Yes	Yes

7.2 提和

遊戲進行中，輪到走子的玩家可以使用提和 (DrawOffer) 來詢問對方是否這局以和局論。若對方同意，則這局遊戲以和局結束；反之，則遊戲繼續進行。另外，如對方無回應，詢問方不需等待對方回應，可以直接走子。

提和的操作在**Section 6.1.3** 說明。

注意：提和過程**所耗費的詢問時間皆算在思考時間**上，請注意時間限制。

注意：基於公平性原則，只在某些情況下支援提和功能。

提和	線上伺服器	本地測試
AI vs AI	No	No
AI vs Human	No	No
Human vs Human	Yes	No

7.3 悔棋

悔棋 (Undo) 功能提供玩家一個方便的功能，讓玩家可以在下棋或是測試 AI 程式時，回到之前的走步，方便玩家學習或是除錯。

悔棋的操作在Section 6.1.3 說明。

注意：悔棋只能在**手動模式**下由玩家提出。

注意：基於公平性原則，只在某些情況下支援悔棋功能。

悔棋	線上伺服器	本地測試
AI vs AI	No	No
AI vs Human	No	Yes
Human vs Human	No	Yes

7.4 斷線中盤

斷線中盤提供玩家在網路不穩時發生斷線再次恢復盤面繼續遊戲的功能，但此功能**無法在本地測試時使用**。

Figure 25 為斷線中盤的設定範例，必須由開房玩家在**斷線中盤**中選擇**是**的選項。

The screenshot shows a game settings window with tabs for 'Game', 'Room', and 'Other'. The 'Game' tab is active. The settings include:

- 遊戲種類: DarkChess
- 遊戲版本: Ver.1.3.2
- 程式版本: Ver.1.2.0
- 帳號: yahari
- 密碼: yahari
- 計時模式: Absolute
- 秒數限制: 900 (sec)
- 單手制時間: 0
- 幾手共用: 0
- 啟動模式: ☒ 開房, ☐ 加房, ☐ 本地測試
- 斷線中盤: ☒ 是, ☐ 否, ☐ 選擇盤面
- 連接模式: ☐ 讀檔, ☒ 背景, ☐ DCTP, ☐ 手動
- 重複次數: 2
- 長捉次數: 3
- 無吃翻次數: 180
- 先後手: ☒ 先, ☐ 後
- 先後互換: ☒ 是, ☐ 否
- Server IP: 120.126.151.213
- 盤面路徑: (empty field) 選擇
- AI路徑: Search\Search.exe 選擇
- 連線: (button)

Figure 25: 斷線中盤、中盤對下、本地測試的設定範例 (此為 Figure 2 的介面)

7.5 中盤對下

中盤對下提供玩家可以從中盤開始遊戲，可用來對 AI 程式除錯，製作中盤方法在 Section 6.2.2 說明。

Figure 25 為中盤對下的設定範例。必須由開房玩家在**斷線中盤**中選擇**選擇盤面**的選項，並且在**盤面路徑欄位**中選擇中盤的檔案。

7.6 本地測試

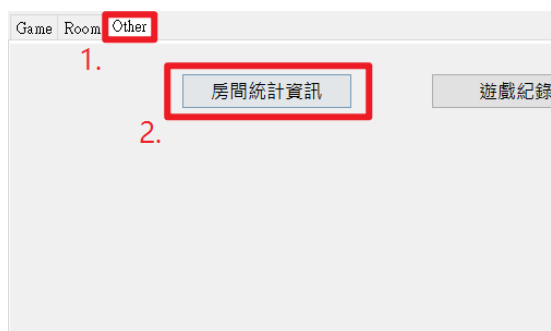
本地測試提供玩家可以不用連上網路即可測試 AI 程式的環境，且提供玩家許多除錯的功能，但有些功能受限或只在本地測試時提供，詳細規定都定義在各功能章節。

Figure 25 為本地測試的設定範例。記得**雙方玩家都要在啟動模式中勾選本地測試**方格。

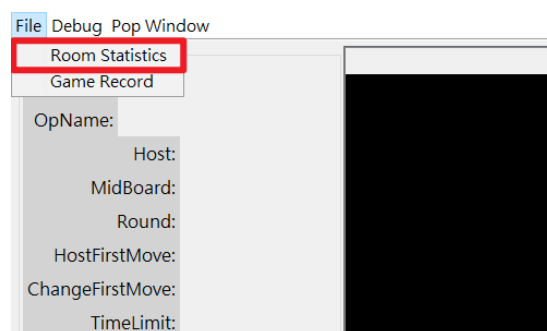
7.7 房間統計資訊

房間統計資訊會儲存一個房間內，各局遊戲的結果、勝負、時間等參數。開啟方式有兩種：

- 程式外部開啟：如 Figure 26(a) 中所示，在打開 Client 之後，切換 Tab 到 Other 頁面，按下房間統計資訊按鈕，即可在遊戲尚未開始時觀看房間統計資訊。
- 遊戲進行中開啟：如 Figure 26(b) 中所示，在房間頁面中，按下 File，選擇 Room Statistics 即可在遊戲進行中，隨時觀看當前或之前的房間統計資訊。



(a) 程式外部開啟



(b) 遊戲進行中開啟

Figure 26: 房間統計資訊頁面開啟方式

Figure 27 中，左方表格列出各局遊戲的資料，右方為雙方玩家勝場比率及和局比率的統計圖表，其下方的 Switch 可以在場數和比率之間做切換，而右下角的 Reload 可以重新讀取檔案，更新最新的遊戲狀況。

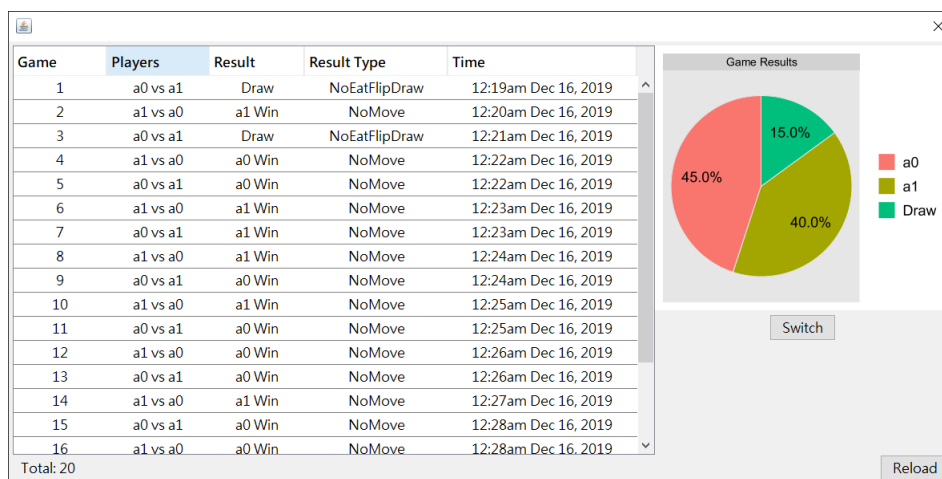


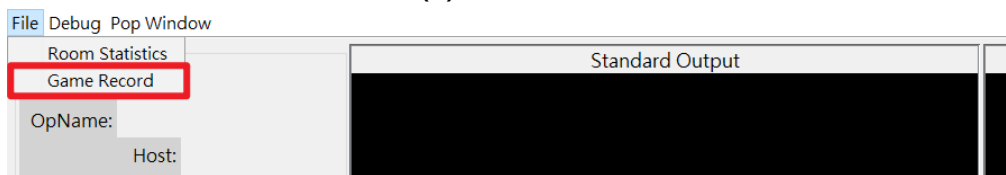
Figure 27: 房間統計資訊頁面

7.8 遊戲紀錄

提供玩家一個可以查看每局遊戲的詳細內容，包含走步、程式輸出、結果等。Figure 28 是開啟方式的操作說明。其餘說明在 Section 6.2 中詳述。



(a) 開啟方式 1



(b) 開啟方式 2

Figure 28: 遊戲紀錄網頁

注意：在觀看盤面並移動走步時，旁邊輸出視窗的捲軸移動位置會因為當時程式輸出緩衝區造成紀錄行數有些微的差異，但大致在正確位置附近。

7.9 遠端盤面

為因應有時觀看網頁和 Client 執行要在不同電腦上執行的需求，新增了遠端觀看盤面的功能，可以在遠端用瀏覽器觀看其它電腦的 Search 盤面。其餘說明在 Section 6.3 中詳述。



Figure 29: 遠端頁面開始方式

8 通訊協定

在程式與 Client 之間連接的時候，需要有一定的通訊協定來做溝通。暗棋共有三種通訊協定，分別為 DCTP、背景 (Ponder) 和讀檔 (ReadFile)，供使用者可以依所需來選擇通訊方式。

建議：使用 DCTP 通訊協定。

8.1 棋盤位置代號

各個通訊協定的棋盤位置皆如 Table 1 所示。由左而右分別為英文字母 a 到 d，由下而上為數字 1 到 8。

8	a8	b8	c8	d8
7	a7	b7	c7	d7
6	a6	b6	c6	d6
5	a5	b5	c5	d5
4	a4	b4	c4	d4
3	a3	b3	c3	d3
2	a2	b2	c2	d2
1	a1	b1	c1	d1
	a	b	c	d

Table 1: 棋盤位置代號

8.2 動作格式規定

各個通訊協定皆須符合 Table 2 之規定：

	規定	範例
走子	棋子移動前的位置和棋子移動後的位置應為不同	b1 ⇒ c1
翻棋	棋子移動前的位置和棋子移動後的位置應為相同	b6 ⇒ b6
認負	棋子移動前的位置和棋子移動後的位置皆為a0	a0 ⇒ a0

Table 2: 動作格式規定

8.3 棋子代碼

紅：帥 (K)、仕 (G)、相 (M)、俥 (R)、馮 (N)、炮 (C)、兵 (P)

黑：將 (k)、士 (g)、象 (m)、車 (r)、馬 (n)、包 (c)、卒 (p)

暗子 (X)、空棋 (-)

8.4 DCTP

DCTP 下，傳輸訊息採用標準輸入輸出 (StdIO)。

8.4.1 指令格式

在 DCTP 協定中，Client 傳送給 AI 程式的指令格式如 (1)。指令開頭以指令的數字編號 (id) 開始，後面加上一個**空格**跟指令名稱 (**command**)，接續是一個**空格**加上資料 (**arguments**)，最後加上一個**換行** (\n)。

$$[id] <command> <arguments> \n \quad (1)$$

Client 接收到指令後，要回覆結果，結果必須依照 (2)(3) 的格式。如果指令接收正確，則使用**等號** (=) 開頭進行回覆，並在後方加上傳送指令數字編號、**空格**、回覆資訊和**換行** (\n)，反之則使用**問號** (?) 開頭進行回覆，後方的回覆資訊則改為錯誤資訊。回覆的**指令數字編號** (id) 需要與傳送指令的數字編號 (id) 相同。

- 成功

$$= [id] <text> \n \quad (2)$$

- 失敗

$$?[id] <errmsg> \n \quad (3)$$

注意：請注意格式中**空格**和**換行**的部分。

8.4.2 指令列表

Table 3 為目前所有 DCTP 的指令，其中橘色部分的為目前所使用中的。

id	command	meaning
0	protocol_version	回傳通訊協定版本
1	name	回傳 AI 程式名稱
2	version	回傳 AI 程式版本
3	known_command	是否支援此指令
4	list_commands	列出所支援的指令
5	quit	要求程式終止
6	boardsize	設定盤面大小
7	reset_board	要求重置盤面
8	num_repetition	設定最多盤面重複次數
9	num_moves_to_draw	設定最多無吃翻次數
10	move	傳送走步
11	flip	傳送翻棋
12	genmove	要求產生走步
13	game_over	傳送遊戲結果
14	ready	詢問是否準備完成
15	time_settings	設定時間
16	time_left	傳送剩餘時間
17	showboard	顯示當前盤面

Table 3: DCTP 指令列表

8.4.3 指令參數

這邊介紹在各指令中，接收指令中的 <arguments> 和回覆指令中的 <text> 的相關規定。

0. protocol_version：回傳通訊協定版本

	參數	範例
接收	無	0 protocol_version\n
回傳	< 通訊協定版本 >	=0 1.0.0\n

1. name：回傳 AI 程式名稱

	參數	範例
接收	無	1 name\n
回傳	<name>	=1 yanyu\n

2. version：回傳 AI 程式版本

	參數	範例
接收	無	2 version\n
回傳	<version>	=2 5.13.2\n

3. known_command：是否支援此指令

	參數	範例
接收	<command>	3 known_command name\n
回傳	true or false	=3 true\n

4. list_commands：列出所支援的指令

	參數	範例
接收	無	4 list_commands\n
回傳	<command>\n<command>\n...	=4 protocol_version\nname\n...

- 請在每個指令中間加上換行(\n)。

5. quit：要求程式終止

	參數	範例
接收	無	5 quit\n
回傳	無	=5\n

- 請回傳回覆指令後自行關閉程式。

6. boardsize：設定盤面大小

	參數	範例
接收	<row> <column>	6 boardsize 8 4\n
回傳	無	=6\n

7. reset_board：要求重置盤面

	參數	範例
接收	無	7 reset_board\n
回傳	無	=7\n

8. num_repetition：設定最多盤面重複次數

	參數	範例
接收	<number>	8 num_repetition 3\n
回傳	無	=8\n

9. num_moves_to_draw：設定最多無吃翻次數

	參數	範例
接收	<number>	9 num_moves_to_draw 180\n
回傳	無	=9\n

10. move：傳送走步

	參數	範例
接收	<from> <dst>	10 move a1 a2\n
回傳	無	=10\n

- 走步位置格式定義在 Section 8.1。

11. flip：傳送翻棋

	參數	範例
接收	<location> <piece>	11 flip a3 G\n
回傳	無	=11\n

- 走步位置格式定義在 Section 8.1。
- 棋種格式定義在 Section 8.3。

12. genmove：要求產生走步

	參數	範例
接收	red, black or unknown	12 genmove black\n
回傳	<from> <dst>	=12 c5 c6\n

- unknown 出現在整局遊戲的第一手。
- 移動位置需符合**動作格式**之規定。
- 如需要**認負**，請回傳**a0 a0**，例如：=12 a0 a0\n。
- 走步位置格式定義在 Section 8.1。
- 棋種格式定義在 Section 8.3。

13. game_over：傳送遊戲結果

	參數	範例
接收	red, black or draw	13 game_over red\n
回傳	無	=13\n

14. ready：詢問是否準備完成

	參數	範例
接收	無	14 ready\n
回傳	無	=14\n

15. time_settings：設定時間

	參數	範例
接收	<time>	15 time_settings 900000\n
回傳	無	=15\n

- 參數 <time> 的單位為**毫秒**。

16. time_left：傳送剩餘時間

	參數	範例
接收	red or black <time>	16 time_left black 899999\n
回傳	無	=16\n

17. showboard :

	參數	範例
接收	無	17 showboard\n
回傳	<board>	=17 XXXX\nXXXX....

- 參數 <board> 的中間要加換行(\n)。

8.4.4 流程圖

Figure 30 為 DCTP 的流程圖，在收到結束指令之前，都由 Client 發送指令，而 AI 程式只要針對指令做出相對應的回應，不需要處理任何遊戲流程的問題。

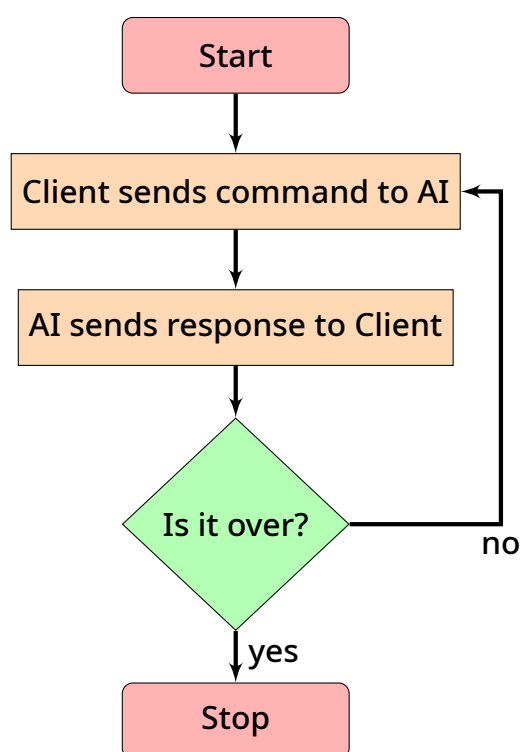


Figure 30: DCTP 流程圖

8.5 背景模式 (Ponder)

背景模式下，傳輸訊息使用 Socket，需要由 Search 先連上 Client 建立連線，方能進行溝通。此模式具備背景思考能力，可以在 AI 下出一手後馬上繼續思考。

注意：此版本有新增一些參數，因此**不適用舊版的程式碼和程式**，請使用新版的程式碼。

8.5.1 檔案介紹

在提供的 Demo 壓縮檔案中，會有如 Figure 31 檔案：

名稱	修改日期	類型	大小
Protocol.h	2019/12/5 下午 05:06	C/C++ Header	1 KB
Protocol.cpp	2019/12/5 下午 11:17	C++ Source	6 KB
MyAI.h	2014/6/22 上午 12:04	C/C++ Header	1 KB
MyAI.cpp	2014/12/22 下午 07:03	C++ Source	11 KB
main_clear.cpp	2014/12/22 下午 06:40	C++ Source	3 KB
main.cpp	2019/12/6 上午 01:43	C++ Source	4 KB
ClientSocket.h	2016/12/21 下午 05:31	C/C++ Header	1 KB
ClientSocket.cpp	2019/1/23 下午 11:08	C++ Source	5 KB

Figure 31: 背景模式檔案列表

- Protocol.h Protocol.cpp ClientSocket.h ClientSocket.cpp：Client 的 Protocol，要使用背景模式時，程式必須使用這些檔案。
- MyAI.h MyAI.cpp：隨機走步的 Demo 檔。
- main.cpp main_clear.cpp：Client 與 MyAI 之間的連接。

當你的程式要使用背景模式時，必須使用**ClientSocket.h**、**ClientSocket.cpp**、**Protocol.h**與**Protocol.cpp**，而流程部分僅需修改**main_clear.cpp**中，出現**//TODO**的程式碼片段。

8.5.2 編譯方法

範例程式包含 main.cpp、MyAI.h、MyAI.cpp。假設使用 g++，則編譯指令為：

Windows / Ubuntu

```
g++ -o search.exe main.cpp Protocol.cpp ClientSocket.cpp MyAI.cpp
```

假若你的程式為 YourAI.cpp，編譯指令為：

Windows / Ubuntu

```
g++ -o search.exe main.cpp Protocol.cpp ClientSocket.cpp YourAI.cpp
```

編譯完成後，設定 AI 路徑為此程式，再將思考模式選為背景 (Ponder) 即可。

8.5.3 程式碼說明

Listing 1: Class Protocol

```
enum PROTO_CLR{PCLR_RED, PCLR_BLACK, PCLR_UNKNOW};

class Protocol
{
private:
    ClientSocket m_csock;
    void Start(const char *state, int piece_count[14],
        char current_position[32], struct History &history,
        int &time);
public:
    Protocol();
    ~Protocol();
    bool init_protocol(const char *ip, const int port);
    void init_board(int piece_count[14], char current_position[32],
        struct History &history, int &time);
    void get_turn(bool &turn, PROTO_CLR &color);
    void send(const char src[3], const char dst[3]);
    void send(const char move[6]);
    void recv(char move[6], int &time, bool &isLast);
    PROTO_CLR get_color(const char move[6]);
};
```


1. 初始化 Protocol :

Protocol::init_protocol

```
void init_protocol(const char *ip, const int port);
```

GUI 介面透過參數傳遞 ip 和 port 到此來連接到 Server 取的初始資訊，在一場比賽開始時 Init_protocol 必須被呼叫。

Listing 2: Class Protocol

```
int main(int argc , char **argv)
{
    Protocol protocol;
    switch(argc){
        case 2:
        case 3:
            if(!protocol.init_protocol(argv[argc-2], atoi(argv[argc-1])))
                return 0;
            break;
    }
    ...
    return 0;
}
```

2. struct History :

struct History

```
struct History{
    char** move;
    int number_of_moves;
};
```

此為紀錄歷史走步的資料結構，可用來回復盤面。

在此定義 move 和 number_of_move :

- move :
 - 走步 : a3-a4
 - 翻棋 : c2(K)
- number_of_moves : 若 number_of_moves = 3, 則我們可以使用 move[0]、move[1] 和 move[2]。

如果你需要讀取歷史走步, 你可以在範例程式碼中如下程式區塊內實作 (在 TODO 的區塊)。

Listing 3: Class Protocol

```
struct History history;
protocol->init_board(piece_count, current_position, &history);
for (int i = 0; i < history.number_of_moves; i++) {
    // TODO: restore the history to your program.
}
```

3. 初始化盤面：

Protocol::init_board

```
void init_board(int piece_count[14],
               char current_position[32],
               struct History history);
```

當你呼叫 init_board 時你會取以下這些棋局資訊:

- piece_count[14]: 14 個兵種存活的個數, 棋子代碼可看 Section 8.3, 順序為: **K**、**G**、**M**、**R**、**N**、**C**、**P**、k、g、m、r、n、c、p。
- current_position[32]: 棋盤上各個位置的兵種, 順序為棋盤的左上角開始, 由左至右, 由上而下, 直到棋盤的右下角。
- history: 歷史走步。

Figure 32 為初始化盤面的範例：

- piece_count[14]: {1, 2, 2, 2, 2, 2, 4, 1, 2, 2, 2, 2, 2, 5}
- current_position[32]: {'n', 'X', 'X', 'X', 'k', 'X', 'X', 'X', 'X', 'c', 'X', '-', 'X', 'X', 'X', 'X', 'X', 'X', 'X', 'X', 'X', 'P', 'X', 'X', 'X', 'X', 'X', 'X', 'X', 'X', 'X'}

4. 取得先後手：

Protocol::get_turn

```
enum PROTO_CLR {PCLR_RED, PCLR_BLACK, PCLR_UNKNOWN};
void get_turn(bool &turn, PROTO_CLR &color);
```



Figure 32: 初始化盤面範例

- turn
 - true：先手。
 - false：後手。
- color
 - PCLR_RED：紅子方。
 - PCLR_BLACK：黑子方。
 - PCLR_UNKNOWN：未知。

當你從初始盤面開始比賽時, 你取得的顏色是 PCLR_UNKNOWN (需要透過第一手翻子來決定你的顏色)。若是從中盤對下, 則你取得的 color 為 PCLR_RED 或 PCLR_BLACK。

5. 傳送走步：

Protocol::send

```
void send(const char src[3], const char dst[3]);
void send(const char move[6]);
```

- 前者：將起始位置和目標位置分開。
 - 走步：例如從 d5 移動到 c5，則 src = d5 且 dst = c5。
 - 翻棋：例如翻開 d5 的棋子，則 src = d5 且 dst = d5 (src = dst)。
 - 認負：src = a0 且 dst = a0。
- 後者：將起始位置和目標位置合併。
 - 走步：例如從 d5 移動到 c5，則 move = d5-c5。
 - 翻棋：例如翻開 d5 的棋子，則 move = d5-d5。

- 認負：move = a0-a0。

注意：請確保動作代碼和移動位置符合**動作格式**之規定。

6. 接收走步：

Protocol::recv

```
void recv(char move[6], int &time, bool &isLast);
```

- move 是經由 Server 傳送過來的對手走步。
 - 走步：例如從 a3 移動到 a4，則 move = a3-a4。
 - 翻棋：例如翻開帥位於 c2，則 move = c2(K)。
- time 為我方玩家剩餘時間。
- isLast 表示這步走步是否為整局遊戲的最後一步。

7. 取得顏色：

Protocol::get_color

```
enum PROTO_CLR {PCLR_RED, PCLR_BLACK, PCLR_UNKNOWN};  
PROTO_CLR get_color(const char move[6]);
```

此函式回傳棋子的顏色。

- 翻開 a8(G)：

Listing 4: 翻開 a8(G)

```
PROTO_CLR color;  
char move[6] = "a8(G)";  
color = get_color(move); /* color == PCLR_RED */
```

- 翻開 d6(p)：

Listing 5: 翻開 d6(p)

```
PROTO_CLR color;  
char move[6] = "d6(p)";  
color = get_color(move); /* color == PCLR_BLACK */
```

8.6 讀檔模式 (ReadFile)

此模式下，Client 與 Search 採用檔案讀寫 (FileIO) 進行溝通。另外，此模式下 AI 程式不具備背景思考能力。

8.6.1 流程圖

Figure 33 為 ReadFile 的流程圖。Client 會在要求走步的時候呼叫 AI 程式，並等待 AI 程式關閉。此時，AI 程式需要讀取 board.txt，產生最佳走步，並將結果依照規定格式輸出至 move.txt，接著再由 Client 將結果傳送至伺服器。

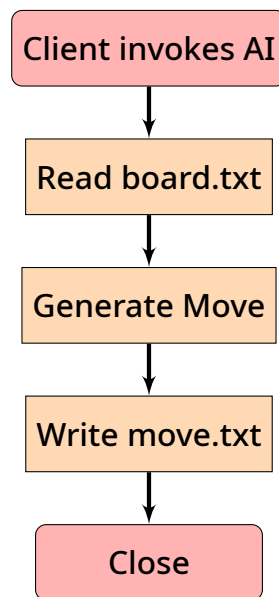


Figure 33: 讀檔模式流程圖

8.6.2 Board 檔

每一手 (ply) 的開始，將目前的棋局紀錄存於與AI 同一個資料夾內，使得 AI 程式讀取目前盤面，搜尋最佳步。board.txt 的格式如 Figure 34 所示，每個方框的意義如下：

1. 遊戲開始時間
2. 雙方玩家：先手在前，後手在後
3. 剩餘棋子狀態：照順序分別為 **K**、**G**、**M**、**R**、**N**、**C**、**P**、k、g、m、r、n、c、p
4. 初始盤面：棋子代碼定義在 Section 8.3。
5. 先手執子顏色
 - 0：先手執紅
 - 1：先手執黑
 - 2：尚未知曉 (出現在整局遊戲的第一手)

```

board.txt - 記事本
檔案(F) 編輯(E) 格式(O) 檢視(V) 說明
* 11:19pm Dec 7, 2019 1.
* Player1(a1) vs Player2(a0) 2.
* 1 2 2 2 2 2 5 1 2 2 2 2 2 5 3.
* X X X X 4.
* X X X X
* X X X X
* X X X X
* X X X X
* X X X X
* X X X X
* X X X X
* First 0 5.
* time 900 6.
* 1. b6(N) b4(K) 7.
* 2. a7(p) b2(p)
* 3. a3(N) d4(n)
* Comment 0 0
0. 1575760811020
1. 1575760811020
2. 1575760811239 8.
3. 1575760811317
4. 1575760811458
5. 1575760811536

```

Figure 34: 讀檔模式之 board.txt

6. 出手限制時間

7. 歷史走步

- 翻棋：b7(M) 表示在 b7 的位置翻出相
- 走棋：b5-b6 表示 b5 的棋子移動至 b6

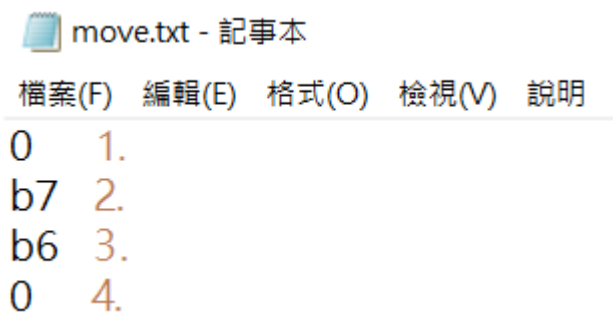
8. 每手時間紀錄：時間計算方式為伺服器收到走步的時間郵戳。時間郵戳為 1970/1/1 00:00:00 GMT 到現在所經過的毫秒數。

注意：在讀取 board.txt 的時候，需要讀取**初始盤面**(Figure 34 中淺綠色方框) 和**歷史走步**(Figure 34 中淺藍色方框)，才能讀取到最新的盤面。

注意：檔案名稱必須為**board.txt**，請確保能讀取到盤面資料。

8.6.3 Move 檔

當 AI 程式搜尋完畢，將該手 (ply) 的最佳走步存入 move.txt 並輸出至當前同一目錄下。move.txt 的格式如 Figure 35 所示：



```
move.txt - 記事本
檔案(F) 編輯(E) 格式(O) 檢視(V) 說明
0 1.
b7 2.
b6 3.
0 4.
```

Figure 35: 讀檔模式之 move.txt

1. 動作代碼

- 0：走步
- 1：翻棋
- 2：認負 (如需認負，棋子移動前的位置和棋子移動後的位置皆須設為a0。)

2. 棋子移動前的位置

3. 棋子移動後的位置

4. 保留

注意：請確保動作代碼和移動位置符合動作格式之規定。

注意：動作代碼中的認負功能是在本系統中新增的動作代碼，對於舊版的系統並不支援，請勿在舊版系統中使用。

注意：檔案名稱必須為 move.txt，切勿輸出錯誤導致 Client 讀取不到走步資料。

9 參考資料與聯絡資訊

1. Jr-Chang Chen, Gang-Yu Fan, Hung-Jui Chang, Tsan-sheng Hsu (2018). Compressing Chinese Dark Chess Endgame Databases by Deep Learning. *IEEE Transactions on Games* 10(4), 413–422.
2. Hung-Jui Chang, Jr-Chang Chen, Gang-Yu Fang, Chih-Wen Hsueh, Tsan-sheng Hsu (2018). Using Chinese Dark Chess Endgame Databases to Validate and Fine-Tune Game Evaluation Functions. *ICGA Journal* 40(2), 45–60.
3. Hung-Jui Chang, Jr-Chang Chen, Chih-Wen Hsueh, Tsan-sheng Hsu (2018). Analysis and Efficient Solutions for 2×4 Chinese Dark Chess. *ICGA Journal* 40(2), 61–76.
4. Chu-Hsuan Hsueh, I-Chen Wu, Tsan-sheng Hsu, Jr-Chang Chen (2018). An Investigation of Strength Analysis Metrics for Game-Playing Programs: A Case Study in Chinese Dark Chess. *ICGA Journal* 40(2), 77–104.
5. Chu-Hsuan Hsueh, I-Chen Wu, Wen-Jie Tseng, Shi-Jim Yen, Jr-Chang Chen (2016). An Analysis for Strength Improvement of an MCTS-Based Program Playing Chinese Dark Chess. *Theoretical Computer Science* 644(C), 63–75.
6. Jr-Chang Chen, Ting-Yu Lin, Tsan-sheng Hsu (2015). Equivalence Classes in Dark Chess Endgames. *IEEE Transactions on Computational Intelligence and AI in Games* 7(2), 109–122.
7. Shi-Jim Yen, Cheng-Wei Chou, Jr-Chang Chen, I-Chen Wu, Kuo-Yuan Kao (2015). Design and Implementation of Chinese Dark Chess Programs. *IEEE Transactions on Computational Intelligence and AI in Games* 7(1), 66–74.
8. Bo-Nian Chen, Bing-Jie Shen, Tsan-sheng Hsu (2010). Chinese Dark Chess, *ICGA Journal* 33(2), 93–106.

若仍有不清楚之處，煩請聯絡：

- Jr-Chang Chen (陳志昌), email: jcchen@gm.ntpu.edu.tw
- 許嘉銘, email: ldslds449@gmail.com
- 鄧鈞洲, email: stev0206@yahoo.com.tw
- 鍾毓湘, email: s710783117@gm.ntpu.edu.tw