

Geister 使用說明

陳志昌、薛筑軒、吳建德、周子馨、張可欣、楊淳亘、蘇亭仔

February 25, 2025

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 2、3、4、5、9。
- 進階使用：請詳閱 Section 6、7、8。

Contents

1	遊戲介紹	2
2	檔案說明	3
3	環境設置	4
4	開啟方式	5
5	遊戲設定	6
6	房間介紹	12
7	盤面介紹	16
8	功能介紹	24
9	通訊協定	31
10	Q & A	43
11	參考資料與聯絡資訊	46

1 遊戲介紹

Geister 是一種雙人遊玩的桌遊，包含 6x6 的棋盤、8 隻紅色幽靈與 8 隻藍色幽靈 (一位玩家各有 4 隻)。

1.1 遊戲規則



Figure 1: Geister 實體照片

- 準備階段：兩位玩家將自己的 8 隻幽靈分配在靠近自己的 4*2 位置。

注意： 玩家彼此並不知道對方的幽靈配置。

- 遊戲開始：每回合玩家可以移動自己的幽靈一次。
 - 方向：前、後、左、右。
 - 吃棋：若移動的位置有對方的幽靈，則吃掉該棋，此時可查看該棋顏色。
- 遊戲結束：滿足以下條件時，遊戲結束。
 - 吃完對手的四隻藍色幽靈，己方勝利。
 - 吃完對手的四隻紅色幽靈，對方勝利。
 - 己方藍色幽靈從對方地盤角落走出棋盤，己方勝利。

2 檔案說明

推薦在 120.126.151.216 ([頁面連結](#)) 下載/更新檔案。

2.1 Client 資料夾

名稱	修改日期	類型	大小
Board	2024/10/14 下午 11:59	檔案資料夾	
Library	2024/10/14 上午 12:52	檔案資料夾	
LocalServer	2024/10/14 上午 12:52	檔案資料夾	
Search	2024/10/14 上午 12:52	檔案資料夾	
Setting	2024/10/14 上午 12:52	檔案資料夾	
WebServer	2024/10/17 上午 12:48	檔案資料夾	
Launcher	2024/10/14 上午 12:52	Executable Jar File	4,477 KB

Figure 2: Client 檔案結構

- Board：遊戲的盤面紀錄檔案和指令檔案
- 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：網頁檔案
- Launcher.jar：Client 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。

3 環境設置

完成環境建置之後，才能順利執行程式。(Ubuntu 測試時的版本是 22.04.4。)

3.1 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/jdk。以下介紹 Windows 和 Ubuntu 安裝 java 8 64-bit 的方法。

注意：此程式最低執行 Java 版本為 **Java 8 64-bit**。

3.1.1 Windows

到官網下載並安裝 jre ([下載頁面連結](#))。

3.1.2 macOS

開啟 Terminal 輸入以下指令 (需事先安裝 [Homebrew](#))

```
macOS
brew tap adoptopenjdk/openjdk
brew install --cask adoptopenjdk8
```

3.1.3 Ubuntu

開啟 Terminal 輸入以下指令：

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

3.2 Browser

建議使用 Google Chrome 瀏覽器。

3.2.1 Windows / macOS

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

3.2.2 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
```

4 開啟方式

程式的 User Interface 使用 **GUI (Graphical User Interface)**，以下將介紹使用方式。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 / macOS

```
cd <the_path_to_ubuntu>
chmod a=rwx -R ubuntu/
```

4.1 GUI

直接雙擊資料夾中的 **Launcher.jar**，或使用 Terminal 輸入以下指令：

Windows / Ubuntu

```
cd <the_path_to_Launcher.jar>
java -jar Launcher.jar
# or use 'javaw' if you don't want a command prompt window to appear
javaw -jar Launcher.jar
```

macOS 可能遇到的狀況

- 未識別開發者無法打開：

1. 可參考 [打開來自未識別開發者的 Mac App](#)
2. 允許所有程式：

```
# turn off
sudo spctl -master-disable
# turn on
sudo spctl -master-enable
```

- 無法存取檔案 (Unable to access jarfile Launcher.jar)：

給予 java 完全取用磁碟權限

- Java 路徑：
/Library/Java/JavaVirtualMachines/openjdk-???.jdk/
Contents/Home/bin/java
- 可參考 [更改 Mac 上的「隱私權」偏好設定](#)

- 無法連接到伺服器：

檢查是否開啟防火牆，可參考 [更改 Mac 上的「防火牆」偏好設定](#)，如果有，則加入例外清單或關閉防火牆。

5 遊戲設定

此節會介紹 Figure 3 中各個遊戲參數的欄位內容及意義。

5.1 遊戲選項設定

Launcher Ver.5.4.3

遊戲 房間 其他 遊戲列表

遊戲種類： Geister 遊戲版本： Ver.1.0.0 程式版本： Ver.5.4.3

帳號： a1 計時模式： Absolute

密碼： 123 秒數限制： 900 (sec)

房間類別： 一般 比賽 單手制時間： 0

啟動模式： 開房 加房 本地測試 幾手共用： 0

斷線中盤： 是 否 選擇盤面 重複次數： 100

連接模式： TCP STUDIO 手動 和局手數： 200

先手後手： 先 後

先後互換： 是 否

Server IP： 120.126.151.213 盤面路徑： 選擇

AI路徑： 選擇

AI參數： (Empty)

連線

Figure 3: GUI Client 設定頁面

1. 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。
2. 帳號 (Account)：玩家帳號，需向管理者申請。guest 帳號為 a0 至 a2000。
3. 密碼 (Password)：玩家密碼。guest 帳號的密碼皆為 123。
4. 房間類別 (Room Type)：玩家可以選擇要進行一般模式的對打，或是參加比賽模式。
 - 一般 (General)：玩家自行測試，休閒的下棋模式。
 - 比賽 (Contest)：參加比賽，與其他玩家進行比賽和排名。
5. 啟動模式 (Start Mode)：一般模式可選擇，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。

6. 斷線中盤 (MidBoard)：一般模式可選擇，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。
7. 連接模式 (Connect Mode)：指 Client 端與程式的連接協定，詳閱 Section 9。
 - TCP：使用 TCP 協定。
 - STDIO (Standard Input Output)：標準輸入輸出。
 - 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。
8. 先手後手 (Host First Move)：**開房玩家第一局**為先手或後手。
9. 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**。
10. Server IP：要連接的 Server IP。預設為**120.126.151.213**。
11. 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。
12. 秒數限制 (Time Limit)：一局遊戲內玩家的**思考時間(即自由時間)**，不限制必須下幾手。
13. 單手制時間 (One Ply Time)：玩家的思考時間用完之後，**接下來每手的限制時間**。
14. 幾手共用 (Ply Share)：在**限時內**，玩家必須下**幾手**。
15. 重複局數 (Repeat Times)：一局**總共的對弈局數**。
16. 和局手數 (Max Move Number)：此房間每局**最多可走步數**，超過此數以和局計算。
17. 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。

18. 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。

19. AI 參數 (AI Argument)：Client 端會以此參數啟動**電腦對局程式**。

可以使用此欄位設定參數給 AI，抑或是可以自行使用 conf 或 ini 檔案來讓 AI 讀取並設定參數。

5.2 開房

開房玩家需要設定以下的遊戲參數：房間類型、啟動模式、斷線中盤、連接模式、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)、重複次數、和局手數、盤面路徑、AI 路徑。

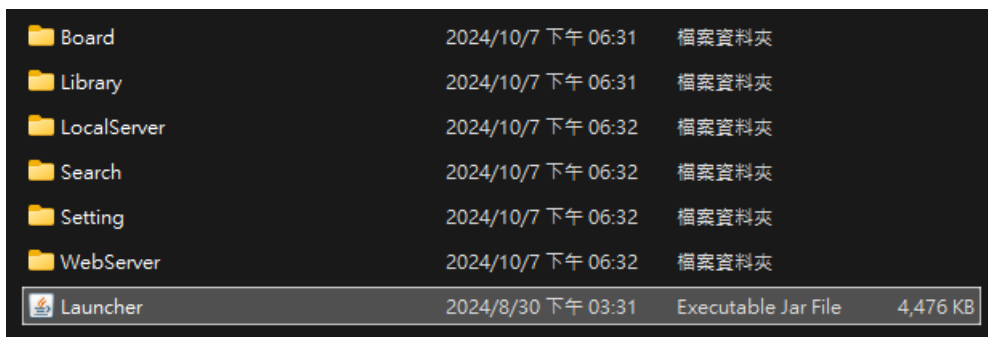


Figure 4: 開房資料夾

進入 open 資料夾，如 Figure 4 所示，直接執行 Launcher.jar。

5.2.1 GUI

在 Figure 3 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。

5.3 加房

加房模式僅需要設定房間類型、連接模式、Server IP、AI 路徑，其餘遊戲參數均由開房玩家設定。

進入 enter 資料夾，如 Figure 4 所示，再次執行 Launcher.jar 即可。

5.3.1 GUI

在 Figure 3 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。

與伺服器成功連線之後會出現加房選單，如 Figure 5，選擇要加房的房間並按下確認即可完成加房。

注意：加房者不能與開房者使用同樣的使用者名稱。

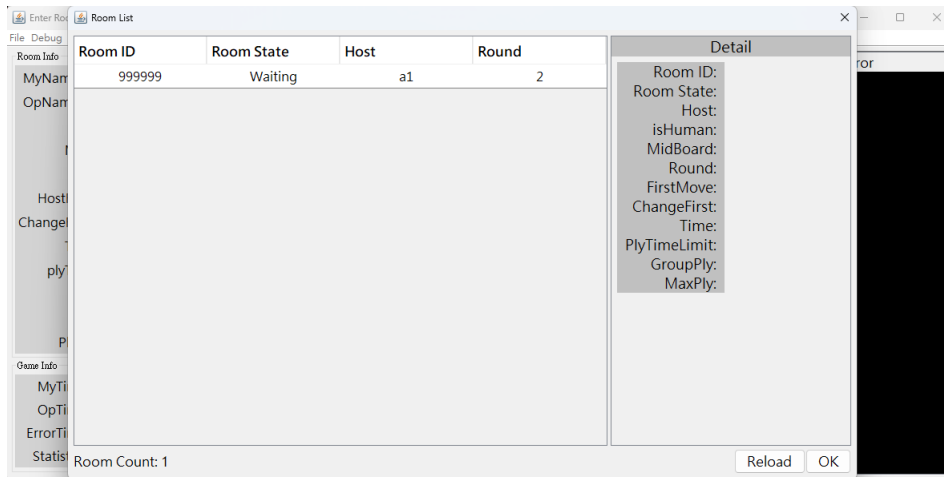


Figure 5: GUI Client 加房選單頁面

5.4 加入比賽

比賽模式需要設定房間類型、連接模式、Server IP、AI 路徑。統一由主辦方設定遊戲參數，玩家只需設定玩家資料即可。

如 Figure 4 所示，執行 Launcher.jar (開啟方式參見 Section 4)。

5.4.1 GUI

在 Figure 6 房間類型選擇**比賽**，設定**玩家資料**，然後**按下連線按鈕**與伺服器連線。



Figure 6: GUI Client 比賽設定頁面

與伺服器連線之後，會出現比賽列表，如 Figure 7 所示，請選擇要加入的比賽進入。

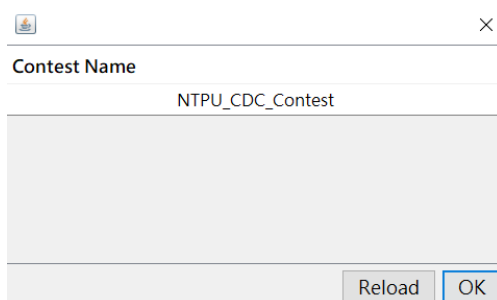


Figure 7: GUI Client 比賽列表頁面

成功加入比賽之後，會出現等待對手加入房間的提示，如 Figure 8 所示，假如對手已加入比賽，則會跳過此畫面。

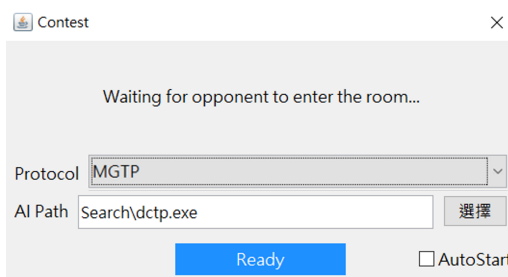


Figure 8: GUI Client 等待對手進入比賽頁面

注意：請勿隨意關閉程式，請耐心等待對手進入，如有任何問題皆可詢問相關人員。

在所有玩家都已經入比賽之後，請確認**比賽程式的參數**是否正確，如 Figure 9 所示，右下角有 AutoStart 方便玩家自動確認，使用後，在下次的比賽頁面將會倒數 10 秒之後自動確認。

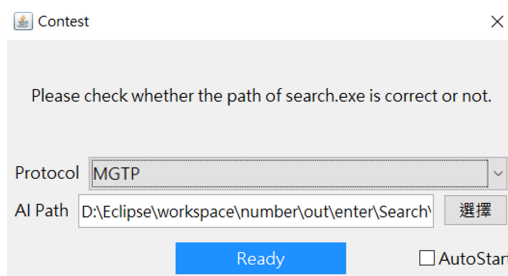


Figure 9: GUI Client 確認比賽程式設定頁面

注意：此項設定確認可在前述步驟內**先行確認**，不必等候到此提示出現後才進行。

在確認程式參數後，如果對手尚未確認完畢程式參數，會出現等候對手的提示，如 Figure 10 所示，假如對手已確認完畢，則倒數 5 秒後直接開始比賽。

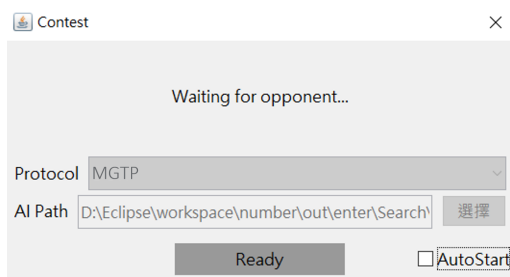


Figure 10: GUI Client 等待對手確認設定頁面

注意：請勿隨意關閉程式，請耐心等待對手確認，如有任何問題皆可詢問相關人員。

在所有玩家確認完畢程式後，則倒數 5 秒直接進行比賽，如 Figure 12 。

如果出現如 Figure 11 的提示，則代表這輪輪空。

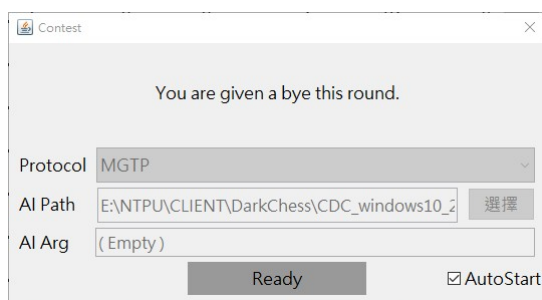


Figure 11: GUI Client 輪空畫面

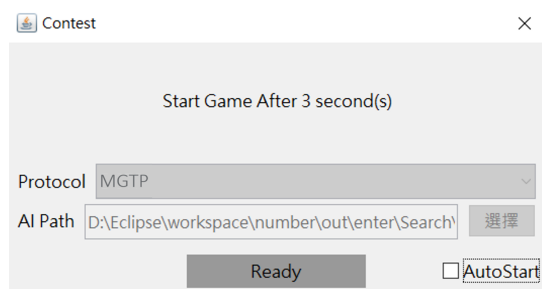


Figure 12: GUI Client 比賽前倒數頁面

6 房間介紹

6.1 房間選項設定



Figure 13: 房間頁面設定

1. 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。
2. 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。
3. 輸出視窗背景顏色：可以更改輸出視窗背景顏色。
4. 輸出視窗字體顏色：可以更改輸出視窗字體顏色。
5. 輸出視窗字體大小：可以更改輸出視窗字體大小。

6. 網頁斷線自動重開：偵測到網頁斷線後，自動重新開啟網頁。
7. 重新遊戲延遲時間：在每局遊戲之間的延遲時間。
8. 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。
9. 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。
10. 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。
11. 比賽模式自動開始：由程式倒數 10 秒後自動開始，玩家不須特地再確認一次。
12. 遊戲盤面紀錄檔案輸出：在 Board 資料夾輸出該場遊戲的紀錄檔案 (.txt)。
13. 遊戲指令紀錄檔案輸出：在 Board 資料夾輸出該場遊戲的指令紀錄檔案(.command)。
14. 遊戲房間統計檔案輸出：在 Board 資料夾輸出該場遊戲的統計檔案 (.info)。

6.2 房間頁面

6.2.1 遊戲參數

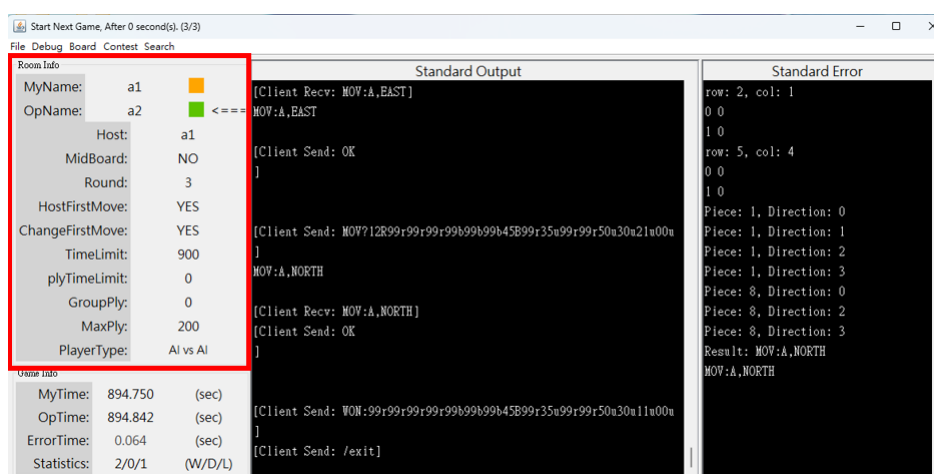


Figure 14: 遊戲參數顯示區域

在 Figure 14 的紅色方框中，上方為我方和對手的帳號名稱、代表顏色，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。

6.2.2 遊戲狀態

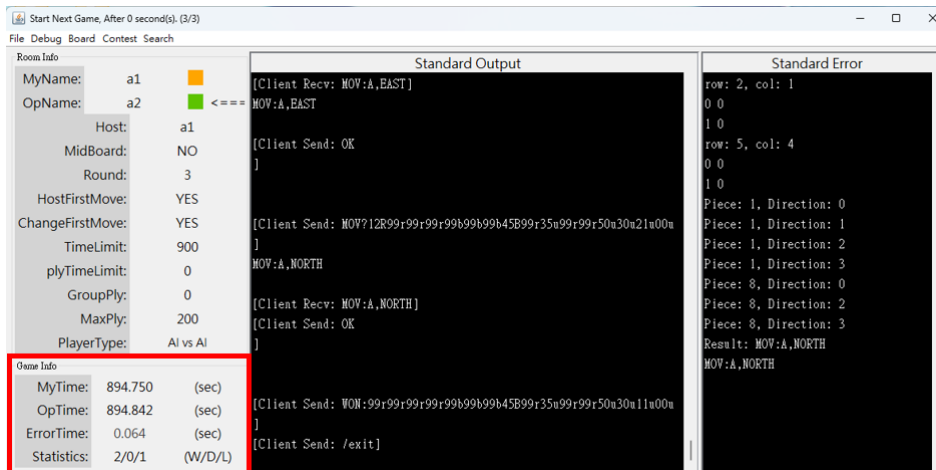


Figure 15: 遊戲狀態顯示區域

在 Figure 15 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 該手所花費時間。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。

6.2.3 輸出視窗

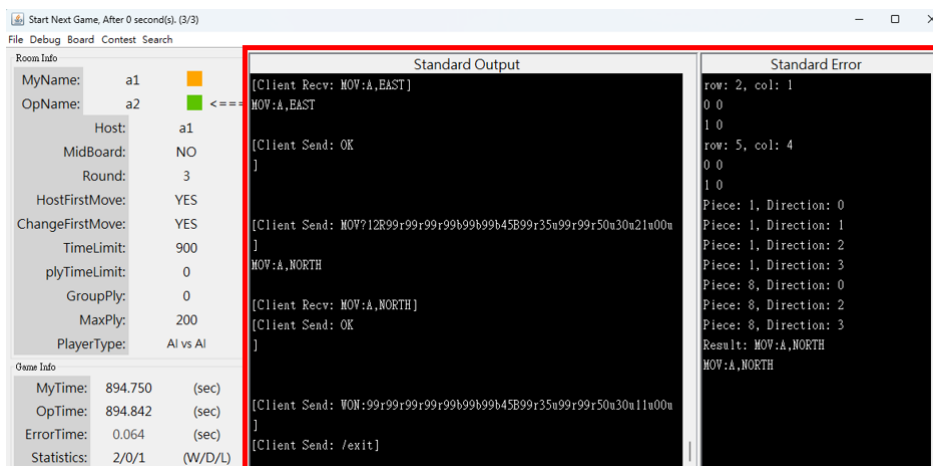


Figure 16: 程式輸出區域

在 Figure 16 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)。

注意：如果發現輸出每隔一段時間才一次全部輸出，無法即時輸出，請參考 QA 10.3。

6.2.4 功能選單

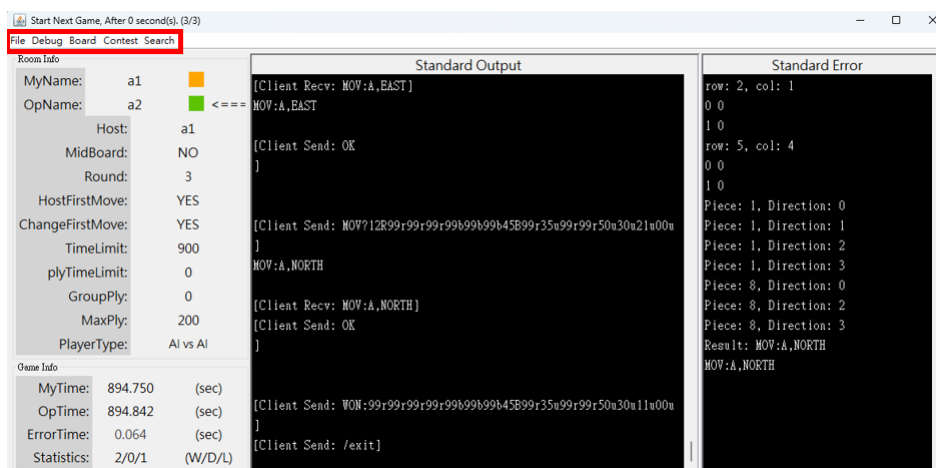


Figure 17: 功能選單區域

Figure 17 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 8.6。
 - Game Record：單一棋局的棋局內容，參見 Section 8.7。
- 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 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。
- Board
 - Open Board：開啟盤面網頁。
- Contest
 - AutoStart：比賽模式下，自動準備並開始比賽。
- Search
 - About：顯示有關目前 Search 的相關資訊。

7 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。

7.1 遊戲盤面介紹

盤面包含以下區域：棋子狀態、棋盤、歷史走步、結果，如 Figure 18 所示。



棋子狀態 棋盤 歷史走步 結果

Figure 18: 顯示頁面

7.1.1 棋子狀態

• 網頁中左方會顯示目前雙方的棋子狀態，如圖 Figure 19 所示。

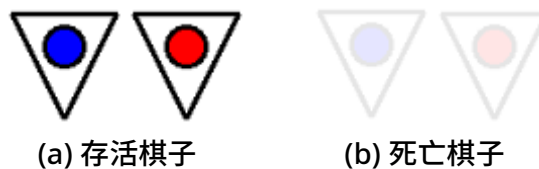


Figure 19: 棋子狀態

7.1.2 棋盤顯示

- 輪到自己的回合時，可動棋子底部會呈灰色，如 Figure 20(a) 所示。
- 選擇灰底棋子後，該棋可移動的位置會以淺綠色方格顯示，點選即可移動棋子，如 Figure 20(b) 與 Figure 20(c) 所示。

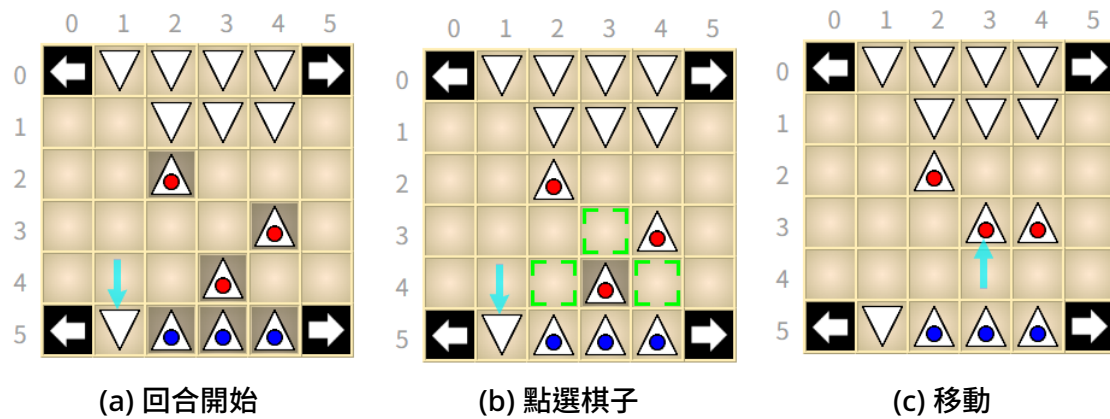


Figure 20: 棋盤顯示走步

7.1.3 手動模式

- **手動模式**下，遊戲開始前玩家需要設定己方的棋盤配置，畫面將如 Figure 21 所示。
- 在設定畫面，玩家點擊兩顆己方棋子可以互換顏色，調整自己的棋盤。
- 設定完畢後，需點擊「擺盤完成」(Set Finish) 按鈕進行確認。



Figure 21: 設定棋盤畫面

- 對手未配置完成的情況下，畫面將如 Figure 22 所示，處於等待狀態。
- 雙方準備完畢時，遊戲會自動開始。



Figure 22: 等待對手設定棋盤

- 輪到我方，可提和、認負與悔棋，如 Figure 23(a)(b)。
- 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。以幽靈棋為例，先手方 “Your turn” 顯示橘色，後手方顯示綠色，如 Figure 24(a) 和 Figure 24(b) 所示。

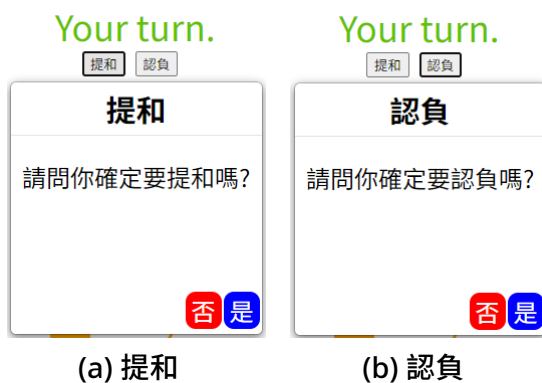


Figure 23: 輪我方時的選項



Figure 24: 手動模式顏色說明

7.2 遊戲紀錄

依照 Section 8.7 的步驟進入遊戲紀錄。

7.2.1 使用說明

- 開啟頁面後，請選擇歷史檔案，如 Figure25 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 26 所示。歷史盤面如 Figure 27 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 28 所示。



Figure 25: 顯示頁面

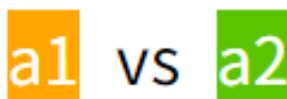


Figure 26: 先手 A1 與後手 A2

7.2.2 製作中盤檔案

- 製作中盤對下 (Section 8.4) 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。在此製作的檔案可作為自訂盤面開啟新遊戲，如 Figure 30 所示。
- 使用方法為：如 Figure 29 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。

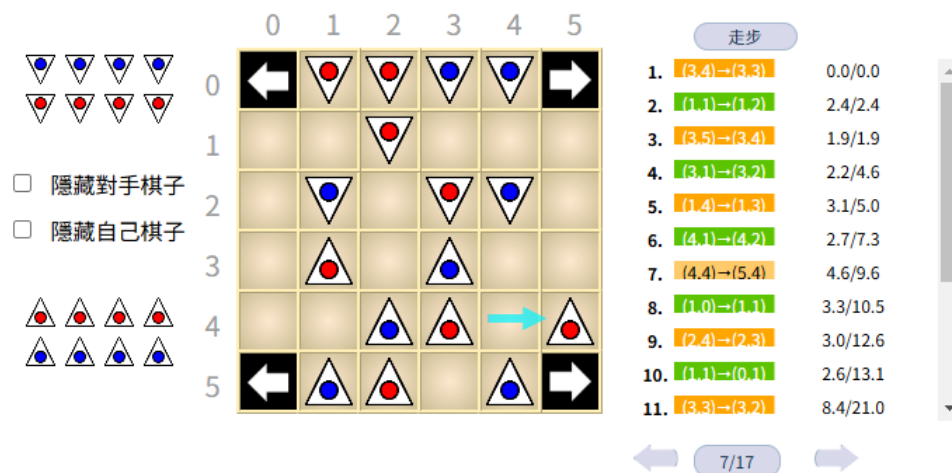


Figure 27: 盤面與歷史走步狀態

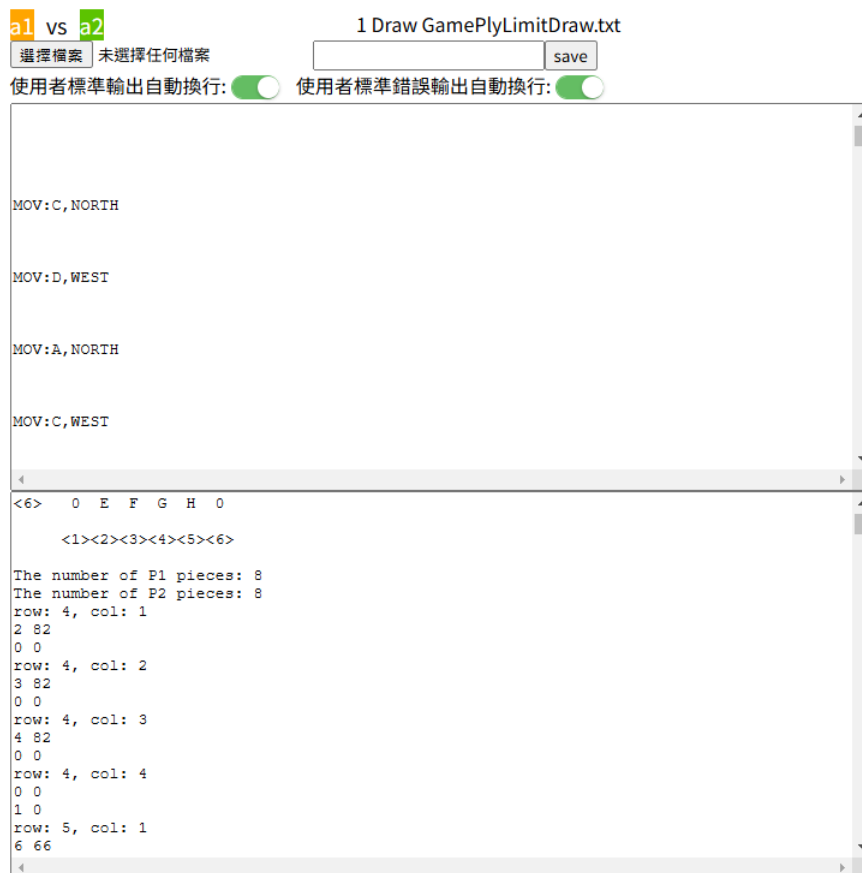


Figure 28: 使用者標準輸入與標準錯誤輸出

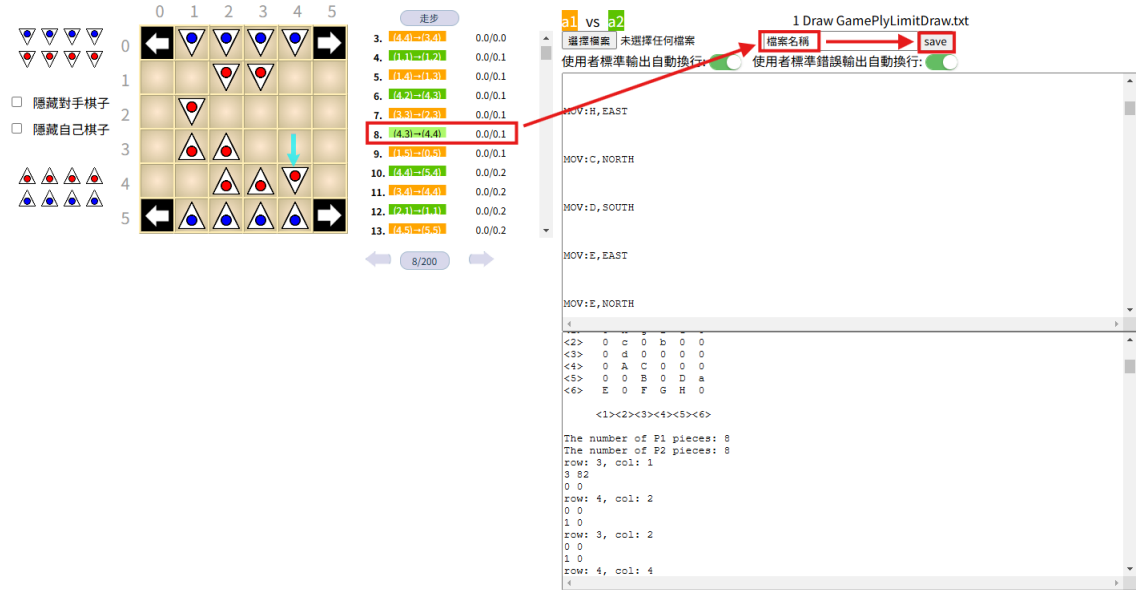


Figure 29: 製作中盤檔案步驟



Figure 30: 使用自訂盤面開啟新遊戲

7.3 遠端頁面

- 假設 Client 的 IP 是 120.126.151.3。在開房或加房依照 Figure 31 設定完畢後，Client 端會出現 Figure 32 的介面。紅色方框顯示 Client 端的 Port，假設 Port 是 50618。
- 在觀戰的電腦上依照 Section 8.8 選擇遠端頁面，即出現 Figure 33 的介面。然後輸入 Client 的 IP: 120.126.151.3 和 Port: 50618 即可觀戰。



Figure 31: 房間頁面設定

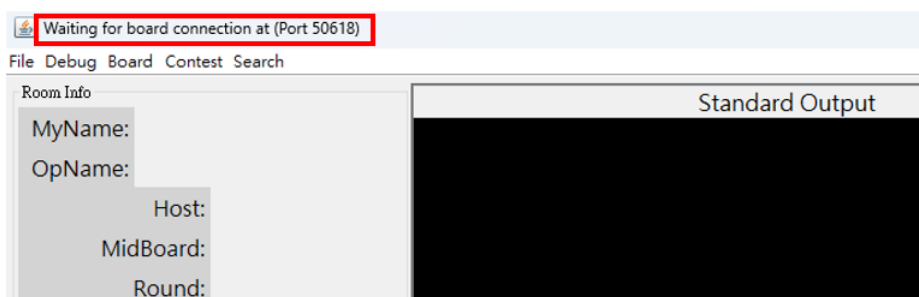


Figure 32: Port 顯示頁面

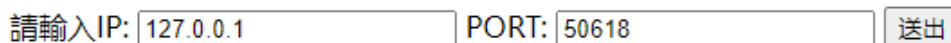


Figure 33: 遠端頁面

7.4 語言切換

遊戲盤面、遊戲紀錄頁面左下角均有語言選項，如 Figure 34、35 所示，可依使用者習慣更改為中、英、日三種語言。



Figure 34: Open、Enter 頁面語言選項

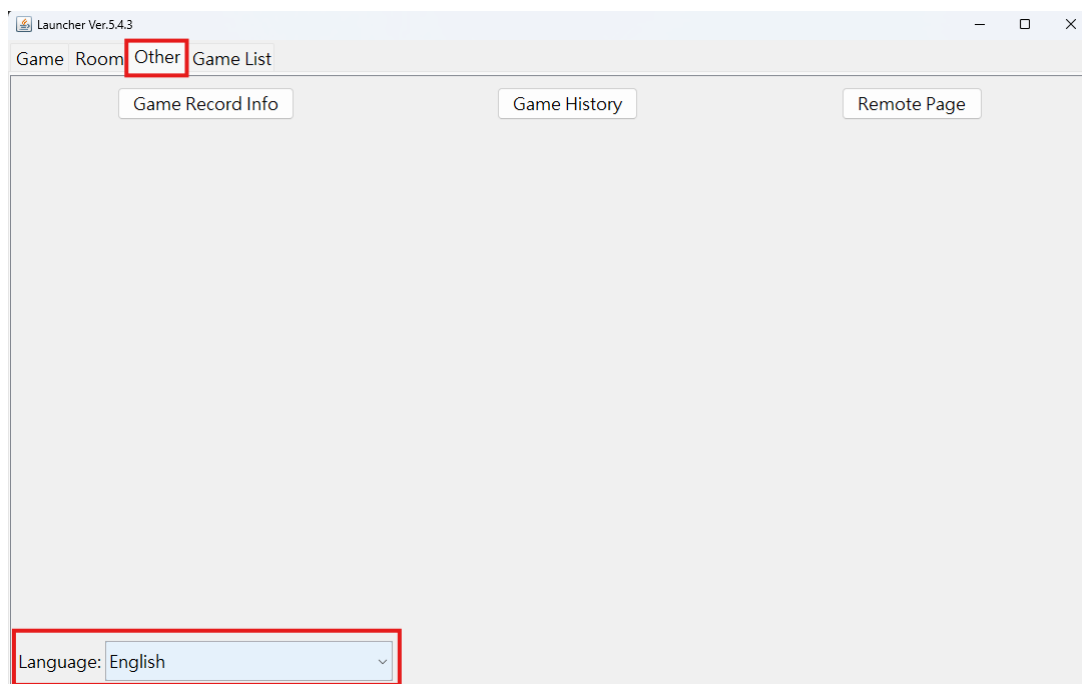


Figure 35: Launcher 語言選項

8 功能介紹

8.1 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 7.1.3)
- AI

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

8.2 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 7.1.3 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

8.3 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 36 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。

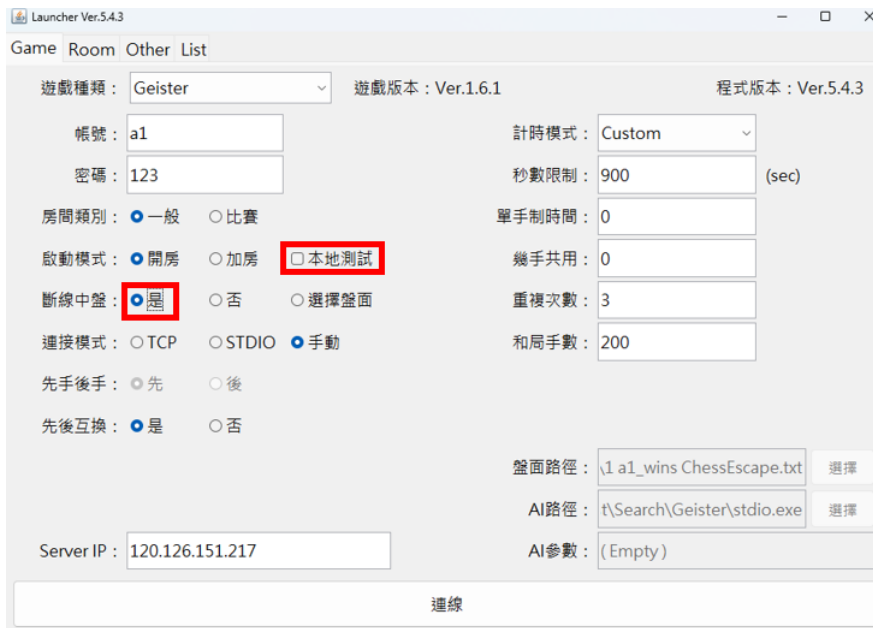


Figure 36: 斷線中盤設定範例 (此為 Figure 3 的介面)

8.4 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。

Figure 製作中盤方法在 Section 7.2.2 說明。

Figure 37 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。

8.5 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。

如 Figure 36 所示，雙方玩家都在啟動模式中勾選本地測試方格即可。

8.6 房間統計資訊

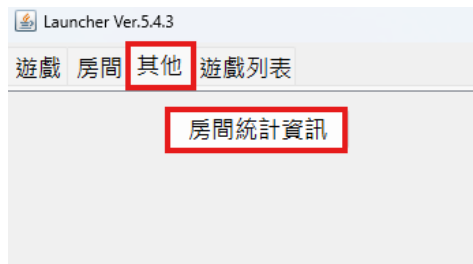
房間統計資訊會儲存一個房間內，各局遊戲的結果、勝負、時間等參數。開啟方式有兩種：

- 程式外部開啟：如 Figure 38(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 38(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。

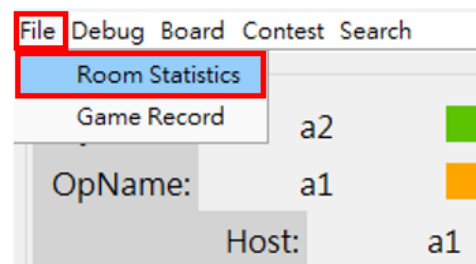
Figure 39 中，左方表格列出各局遊戲的資料，右方為雙方玩家勝場比率及和局比率的統計圖表，其下方的 Switch 可以在場數和比率之間做切換，而右下角的 Reload 可以重新讀取檔案，更新最新的遊戲狀況。



Figure 37: 中盤對下設定範例



(a) 程式外部開啟



(b) 遊戲進行中開啟

Figure 38: 房間統計資訊頁面開啟方式

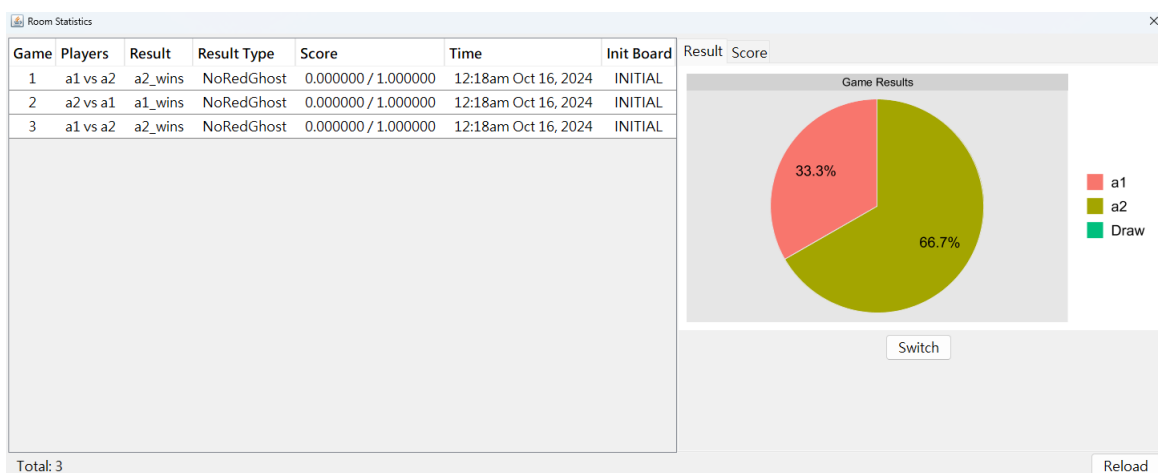
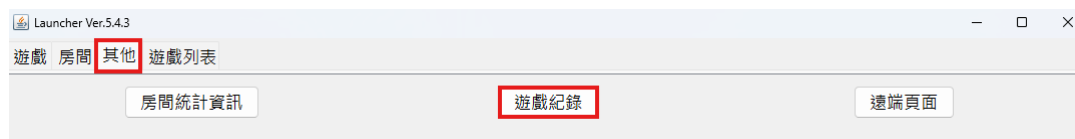


Figure 39: 房間統計資訊頁面

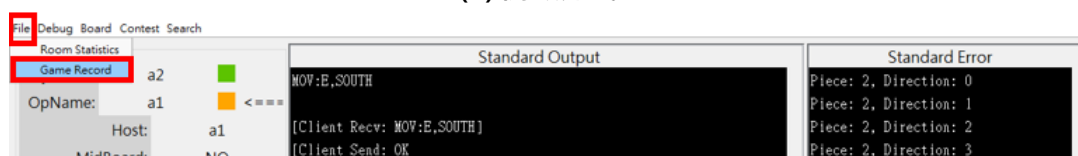
8.7 遊戲紀錄

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 40 是開啟方式的操作說明。

其餘說明在 Section 7.2 中詳述。



(a) 開啟方式 1



(b) 開啟方式 2

Figure 40: 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。

8.8 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 7.3 中詳述。



Figure 41: 遠端頁面開始方式

8.9 比賽模式

比賽模式提供一個比賽的環境，包含分配對手和紀錄勝負的部分，可在此模式中與其他對手一較高下。

開啟方法在 Section 5.4 中詳述。

注意：比賽模式只支援線上伺服器。

比賽模式	線上伺服器	本地測試
AI vs AI	Yes	No
AI vs Human	Yes	No
Human vs Human	Yes	No

8.10 斷線重連

在一般模式和比賽模式遊戲時網路斷線或 Client 關閉提供重新連回遊戲房間功能。

重新連線方法與一般加房或加入比賽相同。

注意：只支援線上伺服器。

斷線重連	線上伺服器	本地測試
AI vs AI	Yes	No
AI vs Human	Yes	No
Human vs Human	Yes	No

8.11 自訂盤面

提供使用者可以直接自訂初始盤面，而不需要依賴歷史走步才能覆盤該盤面。

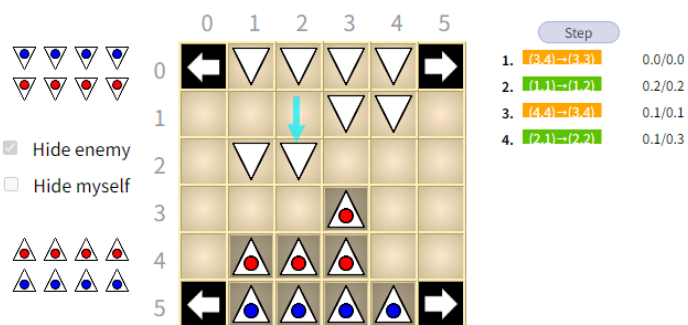
開房者必須依照 Section 8.12 製作合法的中盤檔案，並且將檔案中的剩餘棋子狀態、初始盤面和先手顏色依照需求來製作自訂盤面。

自訂盤面完成後，開房者自必須依照 Section 8.4 的操作來覆盤盤面。

```

This is TEMPORARILY file. You can use it for DEBUG.
* 12:18am Oct 16, 2024
* al vs a2
* 4 4 4 4
* - b b b b -
* - r r r r -
* - - - - -
* - R R R R -
* - D B B B -
* time 900
* 1. C,NORTH D,NORTH
* 2. D,WEST C,NORTH
* Comment 0 0
0. 1729037880601
1. 1729037880601
2. 1729037880757
3. 1729037880897
4. 1729037880991
0. 0
1. 0
2. 0
3. 0
# #
# #
###
    
```

(a) 自訂盤面檔案



(b) 範例盤面顯示

Figure 42: 製作自訂初始盤面範例

8.12 Board 檔案格式說明

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 43 所示，每個方框的意義如下：

```
This is TEMPORARILY file. You can use it for DEBUG.
* 12:18am Oct 16, 2024 1.←
* a1 vs a2 2.←
* 4 4 4 4 3.←
* - b b b b -
* - r r r r -
* - - - - - 4.←
* - - - - -
* - R R R R -
* - R R R R -
* time 900 5.←
* 1. C,NORTH D,NORTH 6.←
* 2. D,WEST C,NORTH
* Comment 0 0
0. 1729037880601
1. 1729037880601
2. 1729037880757
3. 1729037880897
4. 1729037880991 7.←
0. 0
1. 0
2. 0
3. 0 8.←

# #
# #
###
```

Figure 43: 讀檔模式之 board.txt

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：紀錄場上先後手分別的紅藍棋剩餘總數，照順序分別為 **先手紅棋**、**先手藍棋**、**後手紅棋**、**後手藍棋**。
4. 初始盤面：棋子代碼定義在 Section 9.2。
5. 出手限制時間，如果為 0，則表示時間**無限**
6. 歷史走步 (以當前玩家在下方的視角移動)
 - 1. C,NORTH D,NORTH
第一輪，先手將自己的 **C** 向 **北** 一步，後手將自己的 **D** 向 **北** 一步
 - 2. D,WEST C,NORTH
第二輪，先手將自己的 **D** 向 **西** 一步，後手將自己的 **C** 向 **北** 一步
7. 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。

8. 吃棋紀錄：根據 6. 紀錄每一步是否吃到棋，以及吃到棋的顏色。

- 0：沒有吃棋
- 1：吃紅棋
- 2：吃藍棋

注意：在讀取 board.txt 的時候，需要讀取**初始盤面**(Figure 43 中淺綠色方框) 和**歷史走步**(Figure 43 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。

注意：Figure 43 中淺藍色方框中，如果一行只有單個走步，**後方沒有多餘的空白**。

補充：若遊戲正常結束，Board 檔案會在 6. 的後面加入比賽結果。
格式為：< 結束方式 > < 獲勝方/平手 > < 比分 >

- 結束方式
 - NoRedGhost：獲勝方的紅鬼被吃完
 - NoBlueGhost：獲勝方吃完敵方的藍鬼
 - ChessEscape：獲勝方的藍鬼逃脫成功
 - GamePlyLimitDraw：平手
- 獲勝/平手
 - < 玩家 > wins：該玩家獲勝
 - Draw：平手
- 比分
 - 1,0：先手獲勝
 - 0,1：後手獲勝
 - 0.5,0.5：平手

```
* 52. B,NORTH B,WEST
* 53. H,NORTH
* NoRedGhost a2 wins 0,1
* Comment 0 0
0. 1729037926858
1. 1729037926858
```

Figure 44: 比賽結果範例

9 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。本程式共有三種通訊協定，分別為 TCP(Transmission Control Protocol)、STDIO (Standard Input Output) 和手動 (Manual)，供使用者可以依所需來選擇通訊方式。

9.1 棋盤位置代號

各個通訊協定的棋盤位置皆如 Table 1 所示。遵照日本 geister 伺服器座標格式（豎排-橫排），且藍色幽靈在敵方的角落（0-5、5-5 或 0-0、5-0）可以逃出並獲勝。
(參考 Section 11)

	0	1	2	3	4	5
0	0-0	1-0	2-0	3-0	4-0	5-0
1	0-1	1-1	2-1	3-1	4-1	5-1
2	0-2	1-2	2-2	3-2	4-2	5-2
3	0-3	1-3	2-3	3-3	4-3	5-3
4	0-4	1-4	2-4	3-4	4-4	5-4
5	0-5	1-5	2-5	3-5	4-5	5-5

Table 1: 棋盤位置代號

9.2 棋子代碼

遊戲開始時，棋子代碼固定如下分配，空白位置以 - 示意。

	0	1	2	3	4	5
0	-	h	g	f	e	-
1	-	d	c	b	a	-
2	-	-	-	-	-	-
3	-	-	-	-	-	-
4	-	A	B	C	D	-
5	-	E	F	G	H	-

Table 2: 棋子代碼配置

9.3 標準輸出

因為預設的標準輸出時會有緩衝區 (buffer)，造成 client 不會即時收到 search 的輸出，因此，請使用者在輸出回覆的時候，務必確保輸出不會停留在緩衝區，以便讓 client 收到，以下提供幾種 C/C++ 的參考程式碼。

1. 在輸出回覆之後，強制清除緩衝區。

Listing 1: Flush

```
// C
#include <stdio.h>
/* output response */
fflush(stdout);
fflush(stderr);

// C++
#include <cstdio>
/* output response */
std::cout.flush(); // or std::cout << std::flush;
std::cerr.flush(); // or std::cerr << std::flush;
```

2. 在程式一開始將緩衝區大小設為 0。

Listing 2: Set buffer size

```
// C
#include <stdio.h>
setvbuf(stdout, NULL, _IONBF, 0);
setvbuf(stderr, NULL, _IONBF, 0);
/* output response */

// C++
#include <cstdio>
std::cout.rdbuf()->pubsetbuf(0, 0);
std::cerr.rdbuf()->pubsetbuf(0, 0);
/* output response */
```

9.4 指令列表

各模式指令均與日本 geister 伺服器相同 (參考 Section 11)，隨遊戲進程排列如下：

1. 設定紅色幽靈

傳送方	指令	意義
Server	SET?	請求設定紅色幽靈
AI	SET:ABCD\r\n	設定ABCD為紅色幽靈
Server	OK 或 NG	回覆結果

- 根據 Section 9.2，輸入想設定為紅色幽靈的棋子代碼。
- 共有四個紅色幽靈，因此**必須輸入四個位置**。
- 雙方設定完畢後，Server 會向雙方傳送盤面資訊，並開始等待先手走步。

2. 傳送盤面資訊，並請求走步

傳送方	指令	意義
Server	MOV:14R24R34R ... 30u20u10u\r\n	盤面資訊、請求走步

- 根據 Section 9.1 描述各棋的位置。在特定狀態下，座標有不同規定，如下所述。
 - 該棋處於場上：其座標
 - 該棋已被吃：**(9,9)**
 - 該棋從角落逃脫：**(8,8)**
- 根據 Section 9.2，依照 ABCDEFGHabcdefgh 順序，描述各棋的顏色。
 - 未公開（己方場上）的紅/藍棋：R / B
 - 已公開（雙方被吃）的紅/藍棋：r / b
 - 未知顏色：u

3. 移動棋子

傳送方	指令	意義
AI	MOV:A,NORTH\r\n	走步，將A向北移動
Server	OK 或 NG	回覆結果（該移動未吃棋）
Server	OKR 或 OKB	回覆結果（該移動有吃棋）

- 根據 Section 9.1 指定移動的棋子。
- 以當前玩家在下方的視角，東南西北指定移動方向，可使用兩種指定方式：
 - NORTH、EAST、WEST、SOUTH
 - N、E、W、S

4. 遊戲結束

傳送方	指令	意義
Server	WON:14R24R34R ... 30u20u10u\r\n	盤面資訊，該玩家 勝利
Server	LST:14R24R34R ... 30u20u10u\r\n	盤面資訊，該玩家 落敗

9.5 Server 回覆意義

Server 接收到指令後，會回覆結果，格式如以下所示。

- 一般回覆：

指令	意義
OK \r\n	受理成功
NG \r\n	受理失敗

- 移動後吃棋回覆：

指令	意義
OKR\r\n	吃紅棋成功
OKB\r\n	吃藍棋成功

注意：請注意格式中 OK、NG 含有空格，但 OKR、OKB 不含空格。

9.6 流程圖

Figure 45 為通訊的流程圖，在收到結束指令之前，都由 Server 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。

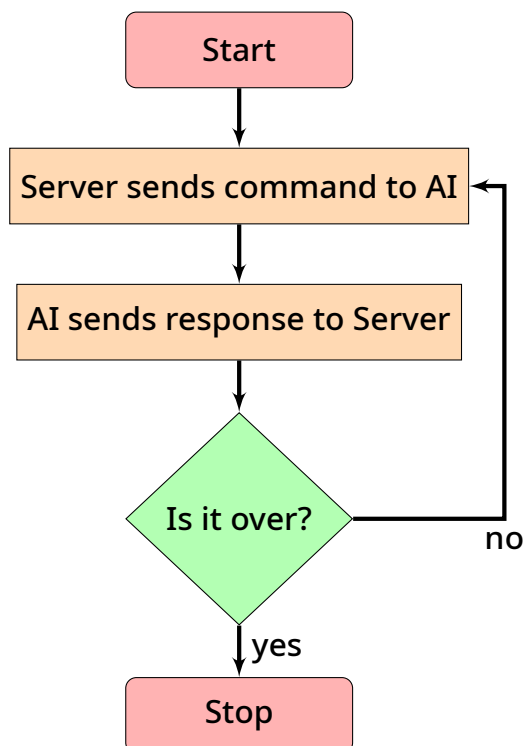


Figure 45: 通訊流程圖

9.7 TCP 模式介紹

TCP 模式下，傳輸訊息使用 Socket。使用者在 AI 路徑與 AI 參數欄位輸入 Search 程式對應的內容，Client 在啟動 Search 程式建立連線時，會根據輸入的 AI 路徑與 AI 參數自動加上 ip 與 port 執行。因此，使用者不需自行輸入 ip 與 port。

在連接模式選擇TCP，即可使用 Figure 46 範例程式的 tcp.jar。

9.7.1 AI 檔案介紹

在 Search 資料夾中，會有如 Figure 46 檔案：

名稱	修改日期	類型	大小
studio	2024/11/21 下午 06:55	應用程式	266 KB
tcp	2024/11/21 下午 06:55	Executable Jar File	40 KB

Figure 46: Search 資料夾檔案列表

需在 TCP 模式使用的 AI 為tcp.jar。

- tcp.jar：日本伺服器 geister.jar 當中的 randomPlayer AI。
 - AI 路徑：< java.exe 路徑 >
 - AI 參數：-jar < tcp.jar 路徑 >

如 Figure 47，選擇 tcp 模式，並參考 Figure 48 在右下角填入路徑與參數。

Launcher Ver.5.4.3

遊戲 房間 其他 遊戲列表

遊戲種類：Geister 遊戲版本：Ver.1.0.0 程式版本：Ver.5.4.3

帳號：a1 計時模式：Absolute

密碼：123 秒數限制：900 (sec)

房間類別： 一般 比賽

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： TCP STUDIO 手動

單手制時間：0

幾手共用：0

先後手： 先 後

重複次數：100

先後互換： 是 否

和局手數：200

Server IP：120.126.151.213

盤面路徑： 選擇

AI路徑：(x86)\Java\jre1.8.0_421\bin\java.exe 選擇

AI參數：ndows\open\Search\Geister\windows\tcp.jar

連線

Figure 47: 開啟 tcp AI 流程

AI Path:	C:\Program Files (x86)\Java\jre1.8.0_421\bin\java.exe	Select
AI Args:	-jar C:\Users\User\Desktop\windows\open\Search\Geister\windows\tcp.jar	

Figure 48: tcp 參數範例

輸入完成後，Client 程式會自行根據 ip 與可用的 port 加在指令尾端。

Client 實際呼叫 Search 的指令：< AI path 內容 > < AI Args 內容 > ip port

因此，使用者的 AI 程式需要讀取參數中的 ip 與 port 進行連線。
以 sample code 為例，此處 args[0] 為 ip，arg[1] 為 port。

```
public class HumanPlayer extends BasePlayer {  
  
    public String readLine() throws IOException {  
        BufferedReader br = new BufferedReader(new InputStreamReader(System.in));  
        return br.readLine();  
    }  
  
    public static void main(String[] args) throws Exception {  
        HumanPlayer p = new HumanPlayer();  
        p.init(args[0], Integer.parseInt(args[1]));  
        System.out.println("red ghosts? (ex. BCDE)");  
        System.out.println(p.setRedItems(p.readLine()));  
  
        GAME_LOOP: while (true) {  
            p.waitBoardInfo();  
            p.printBoard();  
            if (p.isEnded() == true)
```

9.7.2 範例程式

在 120.126.151.216 ([頁面連結](#)) 可以下載範例程式 (Sample code)。
在 TCP 資料夾中，會有如 Figure 49 與 Figure 50 檔案：

名稱	修改日期	類型	大小
BasePlayer	2024/11/25 下午 09:07	Java 來源檔案	7 KB
RandomPlayer	2024/11/25 下午 09:07	Java 來源檔案	2 KB

Figure 49: player 資料夾

名稱	修改日期	類型	大小
Board	2024/11/25 下午 09:07	Java 來源檔案	5 KB
Constant	2024/11/25 下午 09:07	Java 來源檔案	1 KB
Direction	2024/11/25 下午 09:07	Java 來源檔案	1 KB
Item	2024/11/25 下午 09:07	Java 來源檔案	4 KB
ItemColor	2024/11/25 下午 09:07	Java 來源檔案	1 KB
Player	2024/11/25 下午 09:07	Java 來源檔案	2 KB

Figure 50: utils 資料夾

TCP 範例程式均取自日本 [geister 伺服器 github](#)，網址標明於 Section 11。

9.8 STUDIO 模式介紹

STUDIO 全名為 Standard Input Output，代表傳輸訊息採用標準輸入輸出。

在 STUDIO 協定中，Server 與 AI 雖然透過 Client 進行溝通，但 Client 僅負責資料傳輸，不會修改 Server 與 AI 之間的指令內容。因此，本章節仍會介紹 Server 與 AI 之間的指令格式及其功能。

9.8.1 AI 檔案介紹

在 Search 資料夾中，會有如 Figure 51 檔案：

名稱	修改日期	類型	大小
studio	2024/11/21 下午 06:55	應用程式	266 KB
tcp	2024/11/21 下午 06:55	Executable Jar File	40 KB

Figure 51: Search 資料夾檔案列表

需在 studio 模式使用的 AI 為 `studio.exe`。

- `studio.exe`：隨機走步的 AI，若無法直接使用，可參考 Section 9.8.2 自行編譯製作。
 - AI 路徑：< studio.exe 路徑 >
 - AI 參數：無

如 Figure 52，選擇 studio 模式，並參考 Figure 53 在右下角填入路徑。

Launcher Ver.5.4.3

遊戲 房間 其他 遊戲列表

遊戲種類：Geister 遊戲版本：Ver.1.0.0 程式版本：Ver.5.4.3

帳號：a1 計時模式：Absolute

密碼：123 秒數限制：900 (sec)

房間類別： 一般 比賽

啟動模式： 開房 加房 本地測試

斷線中盤： 是 否 選擇盤面

單手制時間：0

連接模式： TCP STUDIO 手動

幾手共用：0

先後手： 先 後

重複次數：100

先後互換： 是 否

和局手數：200

Server IP：120.126.151.213

盤面路徑： 選擇

AI路徑：\\Search\Geister\windows\studio.exe 選擇

AI參數：(Empty)

連線

Figure 52: 開啟 studio AI 流程

AI Path:	C:\Users\User\Desktop\windows\open\Search\Geister\windows\stdio.exe	Select
AI Args:	(Empty)	

Figure 53: stdio 參數範例

9.8.2 使用範例程式製作 AI 檔案

在 120.126.151.216 ([頁面連結](#)) 可以下載範例程式 (Sample code)。
在 STDIO 資料夾中，會有如 Figure 54 檔案：

名稱	修改日期	類型	大小
main	2024/11/21 下午 06:55	C++ 來源檔案	3 KB
MyAI	2024/11/21 下午 06:55	C++ 來源檔案	10 KB
MyAI	2024/11/21 下午 06:55	C Header 來源檔案	2 KB
run.sh	2024/11/21 下午 06:55	sh_auto_file	1 KB

Figure 54: STDIO 資料夾

範例程式包含 main.cpp、MyAI.h、MyAI.cpp、run.sh。

直接執行 run.sh，或在資料夾內開啟 Terminal 輸入以下指令：

Windows/macOS

```
g++ -std=c++11 -o stdio main.cpp MyAI.cpp
```

即可成功編譯 stdio.exe，可將其放入 Search 資料夾供 client 搜尋檔案。

9.8.3 STDIO Sample code 程式碼說明

1. 接收訊息：

```
main.cpp

int main(){
    ...
    do{
        // read command
        if (fgets(read, 1024, stdin) == NULL) {
            fprintf(stderr, "Failed to read from stdin\n");
            break;
        }
        ...
        if (strstr(data[0], "MOV?") != nullptr){
            myai.Get(data, write);
        }
        else if (!strcmp(data[0], "/exit")){
            myai.Exit(data, write);
            break;
        }
        ...
    }while(true);
    return 0;
}
```

讀取標準輸入 stdin，並根據指令呼叫不同函式。

2. 設定紅色幽靈：

```
MyAI.cpp

void MyAI::Set( char* response){
    snprintf(response, 50, "SET:ABCD\r\n");
}
```

接收到 SET? 指令時會呼叫此函式，決定回應內容。

3. 盤面初始化：

```
MyAI.cpp

void MyAI::Ini(const char* data[], char* response){
    ...
    this->Init_board_state(position);
    snprintf(response, ... , position[15]);
}
```


呼叫 `Init_board_state(char* position)` 設定位置，決定回應內容。

4. 傳送走步：

MyAI.cpp

```
void MyAI::Get(const char* data[], char* response){
    ...
    this->Set_board(position);
    ...
    this->Generate_move(move);
    snprintf(response, 50, "MOV:%s", move);
}
```

- 呼叫 `Set_board(char* position)` 更新目前盤面位置。
- 呼叫 `Generate_move(char* move)` 生成走步，決定回應內容。

5. AI 生成走步：

MyAI.cpp

```
void MyAI::Generate_move(char* move){
    ...
    // get legal moves
    int move_count = this->get_legal_move(result);
    ...
    // randomly choose a legal move
    int rand_move = rand() % move_count;
    int piece = result[rand_move * 2];
    // determine the direction of the move
    const char* direction;
    switch(result[rand_move * 2 + 1]) {
        case 0: direction = "NORTH"; break;
        case 1: direction = "SOUTH"; break;
        case 2: direction = "WEST"; break;
        case 3: direction = "EAST"; break;
        default: direction = "UNKNOWN"; break;
    }
    ...
    // generate the new move string
    snprintf(move, 50, "%c,%s", 'A' + piece - 1, direction);
    this->Make_move(piece, direction);
    ...
}
```

- 呼叫 `get_legal_move(int* result)` 確認是否有可移動的棋。
- 隨機選擇其中一顆棋，並決定該棋移動的方向。（可修改此段 AI 判斷程式）
- 生成回應內容。

- 呼叫 `Make_move(const int piece, const char* direction)` 更新盤面。

6. 離開遊戲：

```

MyAI.cpp

void MyAI::Exit(const char* data[], char* response){
    fprintf(stderr, "Bye~\n");
}

```

接收到 `/exit` 指令時，輸出離開遊戲訊息。

7. 其他功能函式：

函式	意義
<code>Init_board_state(char* position)</code>	設定初始盤面
<code>Set_board(char* position)</code>	更新盤面狀態
<code>Print_chessboard()</code>	印出盤面
<code>get_legal_move(int* result)</code>	取得所有合法走步數
<code>referee(int piece, int* dst)</code>	取得指定棋子可移動的方向
<code>Make_move(const int piece, const char* direction)</code>	根據指定棋子和移動方向更新棋盤

10 Q & A

10.1 伺服器連接錯誤 (CONNECT SERVER ERROR)

按下連線按鈕後，發生如 Figure 55 的錯誤，顯示無法連線至伺服器。

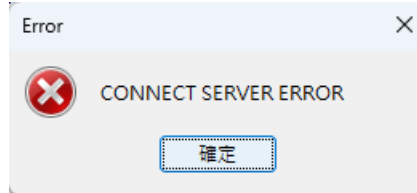


Figure 55: 伺服器連接錯誤 (CONNECT SERVER ERROR)

解決方法：請確認**Server IP**是否正確，如果依然無法解決，請聯絡相關人員。

10.2 登入錯誤 (LOGIN ACCOUNT FAILED)

按下連線按鈕後，發生如 Figure 56 的錯誤，顯示登入時發生錯誤。

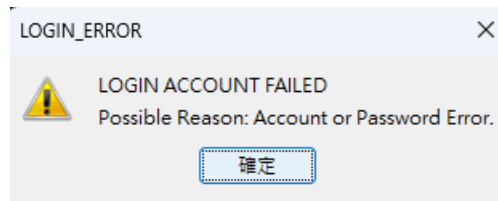


Figure 56: 登入錯誤 (LOGIN ACCOUNT FAILED)

解決方法：請確認**帳號**及**密碼**是否正確，如果依然無法解決，請聯絡相關人員。

10.3 AI 相關問題

相關 AI 的問題，像是 AI 無法執行、標準輸出等之類的問題。

- 明明我的 AI 路徑是正確的，但為何無法正常啟動?

解決方法：請改用**絕對路徑**試試，並確認程式**執行權限**，如果依然無法解決，請聯絡相關人員。

- 為什麼輸出都是隔一段時間才一次全部輸出，無法即時顯示?

解決方法：因為標準輸出有緩衝區，請確認在輸出後有**強制清除緩衝區**(flush)，亦或是**設置輸出緩衝區大小為 0**。

10.4 輸入參數錯誤

按下連線按鈕後，發生如 Figure 57 的錯誤，顯示某項欄位數值不合法，此項錯誤可能有多種發生的狀況，以下列舉。

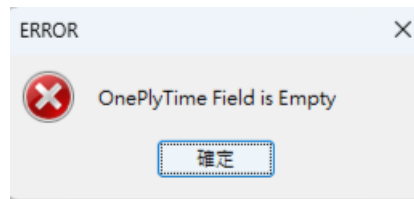


Figure 57: 輸入參數錯誤

- OOO 為空 (OOO is Empty)

例如：帳號欄位為空 (Account Field is Empty)

解決方法：請檢查該欄位**是否填寫**，如果依然無法解決，請聯絡相關人員。

- 請選擇 OOO (Please select the OOO)

例如：請選擇**啟動模式** (Please select the **Start Mode**)

解決方法：請檢查該項目**是否選擇**，如果依然無法解決，請聯絡相關人員。

- 不合法的 OOO (The Input of OOO is not a number)

例如：不合法的**重複次數** (The Input of **RepeatTime Field** is not a number)

解決方法：請檢查該欄位是否填入**數字**的字元，如果依然無法解決，請聯絡相關人員。

- OOO 數值過大 (Value of OOO Overflow)

例如：**重複次數**數值過大 (Value of **RepeatTime** Overflow)

解決方法：請檢查該欄位是否填入小於 2^{32} 的數字，如果依然無法解決，請聯絡相關人員。

- OOO 數值過低 (OOO must larger than zero)

例如：**重複次數**數值過低 (**RepeatTime** must larger than zero)

解決方法：請檢查該欄位是否填入**大於零**的數字，如果依然無法解決，請聯絡相關人員。

- 不合法的 OOO (The Input of OOO is not a Path)

例如：不合法的**盤面路徑** (The Input of **BoardPath Field** is not a Path)

解決方法：請檢查**盤面路徑**是否填入盤面檔案的**檔案路徑**，建議使用**絕對路徑**，如果依然無法解決，請聯絡相關人員。

- 無盤面檔案讀取權限 (The Input of BoardPath Field can not Read)

解決方法：請檢查**盤面檔案**是否具有**讀取權限**，如果依然無法解決，請聯絡相關人員。

- 無 AI 程式執行權限 (Search Permission denied)

解決方法：請檢查**AI 程式**是否具有**執行權限**，如果依然無法解決，請聯絡相關人員。

- 連接模式使用 HUMAN 下瀏覽器模式錯誤 (In order to use HUMAN mode, Please change the browser mode.)

解決方法：請檢查房間設定中**盤面開啟方式**是否設定為**遠端**或是**預設瀏覽器**，如果依然無法解決，請聯絡相關人員。

10.5 伺服器溝通錯誤

在遊戲比賽中途，發生如 Figure 58 的錯誤，顯示接收伺服器資料或者傳送資料給伺服器發生錯誤。

- 傳送資料錯誤 (SEND TO SERVER ERROR)
- 接收資料錯誤 (RECV FROM SERVER ERROR)



Figure 58: 接收伺服器資料錯誤

解決方法：請檢查**網路狀態**，盡量使用穩定度高的網路，如果依然無法解決，請聯絡相關人員。

10.6 STDIO 溝通錯誤 (STDOUT SEND DATA TO SEARCH ERROR)

在遊戲比賽開始時，發生如 Figure 59 的錯誤，顯示無法使用 Stdout 傳送資料給 Search。

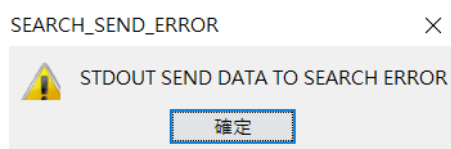


Figure 59: STDIO 溝通錯誤

解決方法：請檢查**連接模式**的選項，大部分原因為設定錯誤的連接模式，如果依然無法解決，請聯絡相關人員。

11 參考資料與聯絡資訊

1. 日本 geister 伺服器 github ([頁面連結](#))
2. 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.
3. 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.
4. 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.
5. 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.
6. 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.
7. 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.
8. 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.

9. 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
- 陳冠銓, email: penguinxdxd@gmail.com
- 吳建德, email: jimmy2053441@gmail.com