Geister Server Instructions

陳志昌、薛筑軒、吳建德、周子馨、張可欣、楊淳亘、蘇亭伃

February 25, 2025

Preface

The primary purpose of this program is to provide an open platform for board game program developers to compete, test their program's strength, or evaluate differences in strength between various versions of their programs. We also offers numerous convenient features to ease the burden on developers.

- Basic Features Usage: Please refer to Section 2, 3, 4, 5, 9.
- Advanced Usage: Please refer to Section 6, 7, 8.

Contents

1 Game Introduction

Geister is a two-player board game featuring a 6x6 board,8 red ghost pieces, and 8 blue ghost pieces (each player has 4 red and 4 blue).

1.1 Game Rule

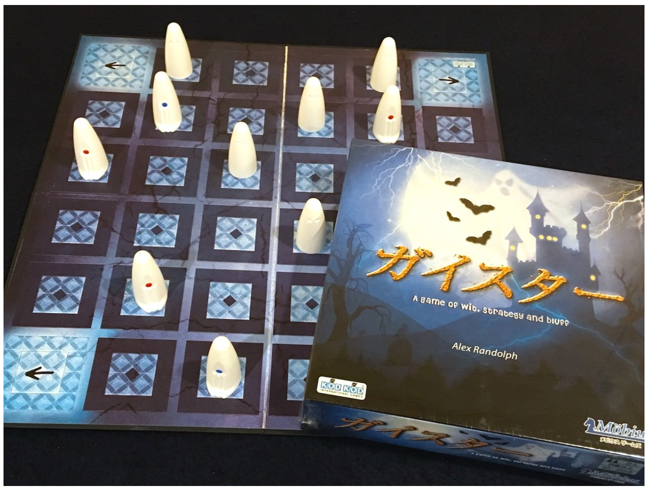


Figure 1: An Actual Photo Of Geister

- Setup: Both players place their 8 ghost pieces within the 4x2 area closest to them.

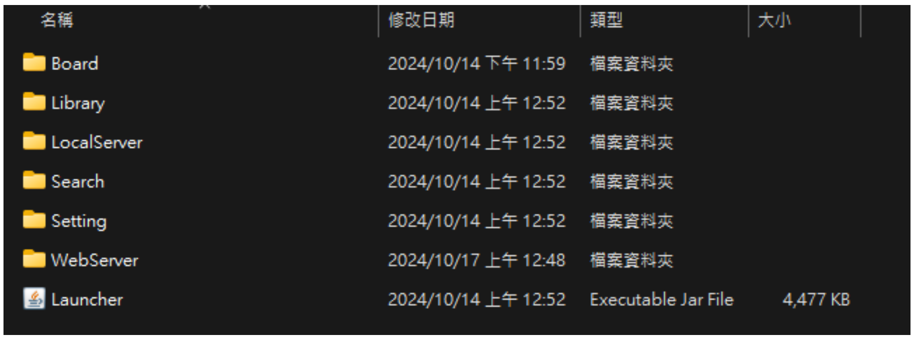
  > Note: Players cannot see each other's ghost color.

- GameStart: Each turn, a player can move one of their ghost pieces.
  - Able directions: front, back, left, right.
  - Capturing a Piece: If a player moves their ghost to a place occupied by an opponent's ghost, the opponent's piece is captured. Both players can know the color of the captured pieces.
- GameEnd: The game ends when one of the following conditions is met.
  - Win: Capture all four of your opponent's blue ghost pieces.
  - Loss: Capture all four of your opponent's red ghost pieces.
  - Win: One of your blue ghost pieces escapes through one of the corners of your opponent's side of the board.

2 File Description

Recommended to download/update the file at 120.126.151.216 (page link).

2.1 Folders



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 File Structure

- Board: Contains game board records and command files.
- Library: The game's function library.
- LocalServer: The game's local server.
- Search: Files for the AI program.
- Setting: Stores game settings and room configuration files.
- WebServer: Web-related files.
- Launcher.jar: The main Client application.

Note: Do not alter the folder structure arbitrarily, as it may cause path errors and prevent the program from running properly.

3 Environment Setup

You need to complete environment setup to run the program.

For Ubuntu, the version tested was 22.04.4, other versions should also work successfully.

3.1 Java

The program is written in Java, requiring a JVM to execute. Therefore, you must install JRE/JDK first. Below are instructions for installing Java 8 64-bit on Windows and Ubuntu.

Note: The minimum required Java version for this program is Java 8 64-bit.

3.1.1 Windows

Download and install JRE from the official website (Page Link).

3.1.2 macOS

Open the Terminal and enter the following command (Requires Homebrew to be pre-installed):

macOS

```
brew tap adoptopenjdk/openjdk
brew install –cask adoptopenjdk8
```

3.1.3 Ubuntu

Open the Terminal and enter the following command:

Ubuntu

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

3.2 Browser

We recommend using Google Chrome as the browser.

3.2.1 Windows / macOS

Download and install Google Chrome from the official website (Page Link).

3.2.2 Ubuntu

Open the Terminal and enter the following command:

Ubuntu

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

sudo dpkg -i google-chrome-stable_current_amd64.deb
```

4 Opening the Program

The program's User Interface uses a GUI (Graphical User Interface). Below is an introduction to how to use it. The usage method is the same for both Windows and Ubuntu. However, due to permission issues on Ubuntu, you need to grant execution permissions.

Ubuntu / macOS

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

4.1 GUI

Double-click Launcher.jar in the folder directly, or use the Terminal:

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
```

Possible Issues on macOS

- cannot be opened because it is from an unidentified developer:
  1. You can refer to instructions on how to open apps from unidentified developers.
  2. Allow all apps:

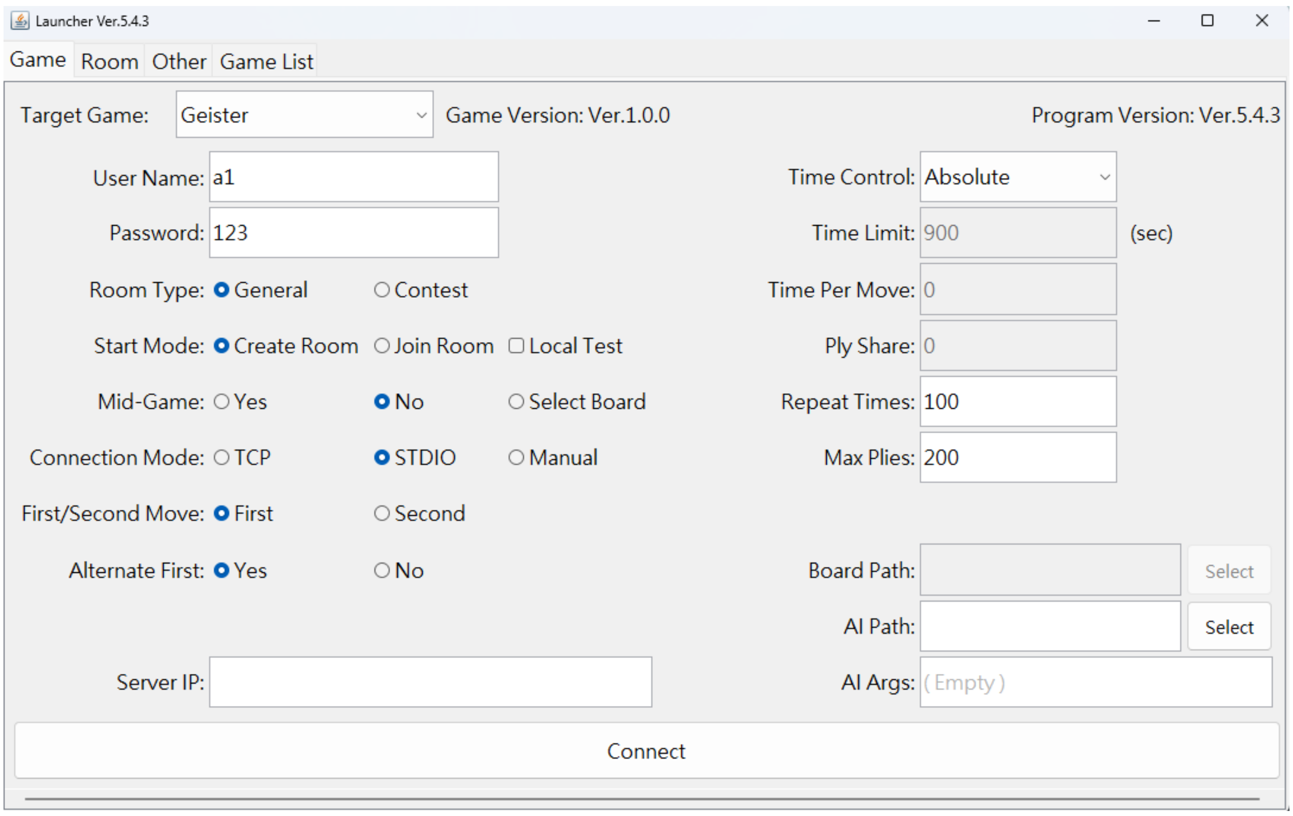
     ```
     # turn off
     sudo spctl –master-disable
     # turn on
     sudo spctl –master-enable
     ```

- Unable to access jarfile Launcher.jar:
  Grant Java full disk access
  - Java path:
    /Library/Java/JavaVirtualMachines/openjdk-???.jdk/Contents/Home/bin/java
  - You can refer to instructions on how to adjust the "Privacy" settings on macOS
- Unable to Connect to Server:
  Check if the firewall is enabled. You can refer to instructions on how to modify the "Firewall" preferences on macOS If the firewall is active, either add an exception or disable the firewall.

5 Game Setting

This section will explain the contents and meanings of the game parameters listed in Figure 3.

5.1 Game Options Settings



Target Game: Geister
Game Version: Ver.1.0.0
Program Version: Ver.5.4.3

User Name:	a1	Time Control:	Absolute
Password:	123	Time Limit:	900 (sec)
Room Type:	[✓ General] Contest	Time Per Move:	0
Start Mode:	[✓ Create Room] Join Room ☐ Local Test	Ply Share:	0
Mid-Game:	Yes [✓ No] Select Board	Repeat Times:	100
Connection Mode:	TCP [✓ STDIO] Manual	Max Plies:	200
First/Second Move:	[✓ First] Second		
Alternate First:	[✓ Yes] No	Board Path:	
		AI Path:	
Server IP:		AI Args:	(Empty)

Figure 3: GUI Client Settings Page

1. Game Type: This platform supports multiple types of games. You can select the desired game category from this option.
2. Account: Player's account, which must be applied for with the administrator. Guest accounts range from a0 to a2000.
3. Password: Player's password.Guest accounts have the password "123".
4. Room Type: Player can choose to engage in a general mode or participate in a contest mode.
   - General: A self-testing, casual mode for playing the game.
   - Contest: A competition mode for playing against other players and ranking.
5. Start Mode: Can be selected in general mode. Player chooses whether to create a room or join an existing one.
   - Open: Create a room for other users to join.

- Enter: Join a room created by another user.
- Local Server: Connects to local server instead of remote server.

6. Mid-Game: Can be selected in general mode. The game setup for the initial board, chosen by the player in Open Room mode.
   - Yes: The server will look for the player's disconnected mid-game records from that day and continue from the last board position.
   - No: The game will use the default initial board provided by the server.
   - Custom: The game will use the file from custom board path chosen by the player to resume the game.
7. Connect Mode: The connection protocol between the client and the program.
   - TCP: Use the TCP protocol.
   - STDIO: Use standard input and output.
   - Human: No program is used, and the player manually plays the game. If both players choose manual mode, a draw can be proposed; if either player chooses manual mode, they can resign or abort the game.
8. Host First Move: The host player will either play first or second in the first game.
9. Change First Move: Whether the players will swap the first and second moves after each game.
10. Server IP: The IP address of the server to connect to. The default is 120.126.151.213.
11. Timer Mode: Combines common timer modes for quick settings of time limits, per-move time, and shared move limits.
    - None: No time limit is set, no time restrictions will apply.
    - Custom: Player can freely set the time limit, per-move time, and shared move options.
12. Time Limit: The thinking time (free time) given for the player in a game.
13. One Ply Time: The time limit for each move after the player's thinking time runs out.
14. Ply Share: Number of moves player must make within the time limit.
15. Repeat Times: The total number of games to be played in the room.
16. Max Move Number: The maximum number of moves allowed in a game. If exceeded, the game will end in a draw.
17. Board Path: When the Mid-Game option is set to "Custom," the client will read the mid-game file from this path and send it to the server as the starting board.

    Recommendation: Use an absolute path.

18. AI Path: The client will start the AI program from this path and connect according to the selected connection protocol.

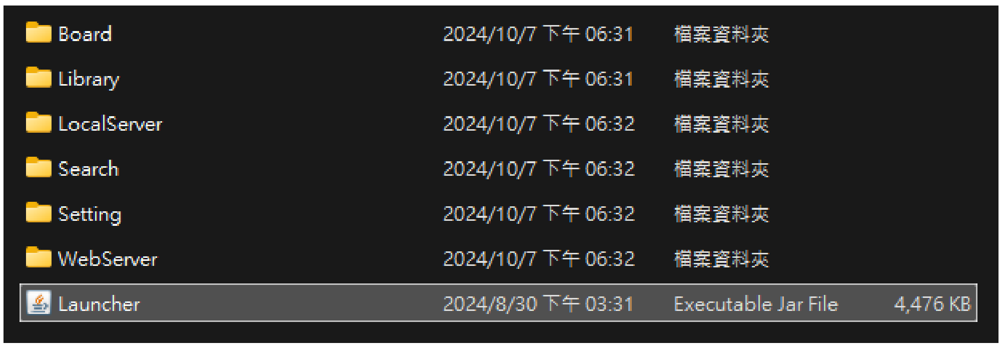
    Recommendation: Use an absolute path.

19. AI Argument: The client will start the AI program with this argument.

    This field can be used to set parameters for AI, or to use a conf or ini file to let the AI read and configure the parameters.

5.2 Create Room

The player creating the room needs to configure the following game parameters: Room Type, Start Mode, Mid-Game, Connect Mode, Host First Move, Change First Move, Server IP, Timer Mode(Time Limit, One Ply Time, Ply Share), Repeat Times, Max Move Number, Board Path, and AI Path.



Board	2024/10/7 下午 06:31	檔案資料夾	
Library	2024/10/7 下午 06:31	檔案資料夾	
LocalServer	2024/10/7 下午 06:32	檔案資料夾	
Search	2024/10/7 下午 06:32	檔案資料夾	
Setting	2024/10/7 下午 06:32	檔案資料夾	
WebServer	2024/10/7 下午 06:32	檔案資料夾	
Launcher	2024/8/30 下午 03:31	Executable Jar File	4,476 KB

Figure 4: open folder

Go to the open folder, as shown in Figure 4, and run Launcher.jar directly.

5.2.1 GUI

As shown in Figure 3, select ”Create Room” for the Start Mode, configure the game parameters and player information, then press the ”Connect” button to connect to the server.

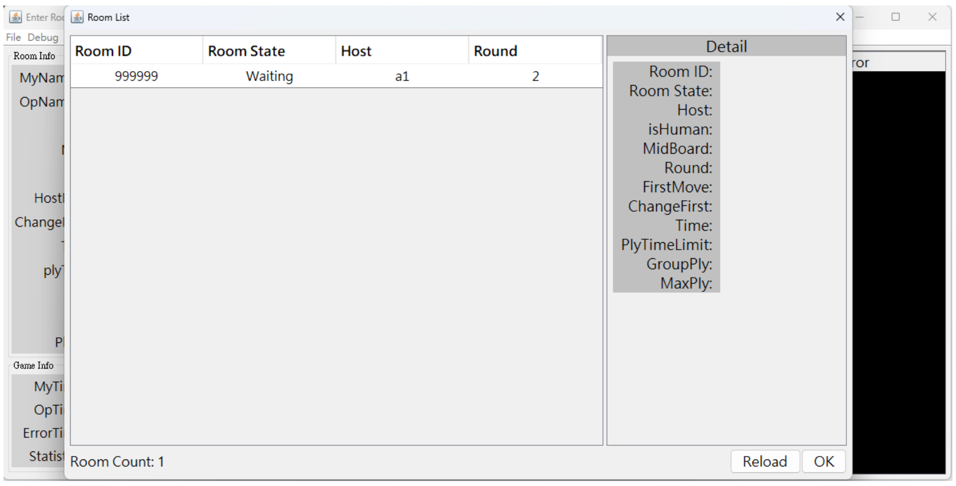
5.3 Join Room

For join mode, only the Room Type, Connect Mode, Server IP, and AI path need to be configured. Other game parameters are set by the player creating the room.

Go to the enter folder, as shown in Figure 4, and run Launcher.jar again.

5.3.1 GUI

In Figure 3, select ”Join Room” for the Start Mode, configure the player information, then press the ”Connect” button to connect to the server.



Room ID	Room State	Host	Round
999999	Waiting	a1	2

Room ID:
Room State:
Host:
isHuman:
MidBoard:
Round:
FirstMove:
ChangeFirst:
Time:
PlyTimeLimit:
GroupPly:
MaxPly:

Room Count: 1

Figure 5: GUI Client Join Room Menu Page

After successfully connecting to the server, the Join Room menu will appear, as shown in Figure 5. Select the room you want to join and click ”Confirm” to complete the process.

Note: The joining player cannot use the same username as the host.

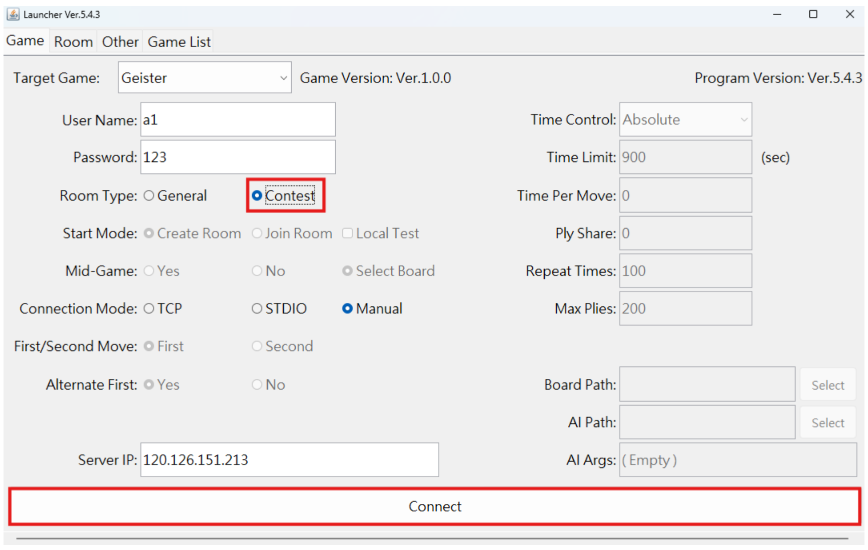
5.4 Joining a Contest

In contest mode, players need to configure the Room Type, Connect Mode, Server IP, and AI Path. The game parameters are uniformly set by the organizer, players only need to configure their player information.

As shown in Figure 4, execute Launcher.jar (refer to Section 4 for the opening method).

5.4.1 GUI

in Figure 6, select ”Contest” for the Room Type, configure the player information, and then press the ”Connect” button to connect to the server.


Launcher Ver.5.4.3

Game Room Other Game List

Target Game: Geister   Game Version: Ver.1.0.0   Program Version: Ver.5.4.3

User Name: a1   Time Control: Absolute

Password: 123   Time Limit: 900 (sec)

Room Type: General [✓ Contest]   Time Per Move: 0

Start Mode: Create Room Join Room Local Test   Ply Share: 0

Mid-Game: Yes No Select Board   Repeat Times: 100

Connection Mode: TCP STDIO [✓ Manual]   Max Plies: 200

First/Second Move: First Second

Alternate First: Yes No   Board Path: Select

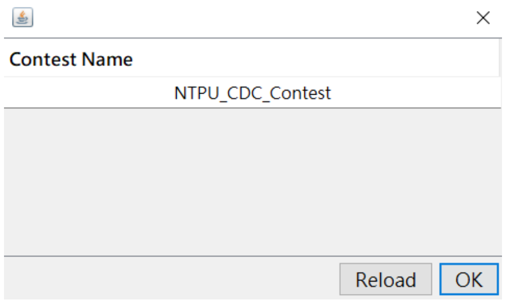
AI Path: Select

Server IP: 120.126.151.213   AI Args: (Empty)

Connect

Figure 6: GUI Client Contest Settings

After connecting to the server, a list of competitions will appear, as shown in Figure 7. Please select the competition you wish to join.

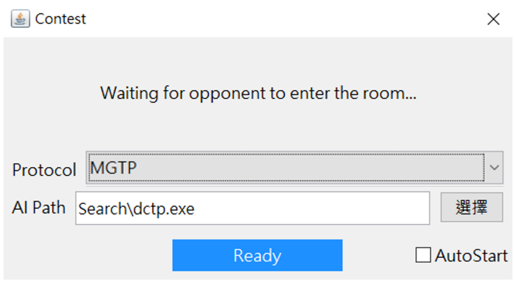

Contest Name

NTPU_CDC_Contest

Reload OK

Figure 7: GUI Client Competition List

After successfully joining a competition, a message will appear indicating that you are waiting for opponents to join the room, as shown in Figure 8. If the opponent has already joined, this screen will be skipped.


Contest

Waiting for opponent to enter the room...

Protocol MGTP

AI Path Search\dctp.exe 選擇

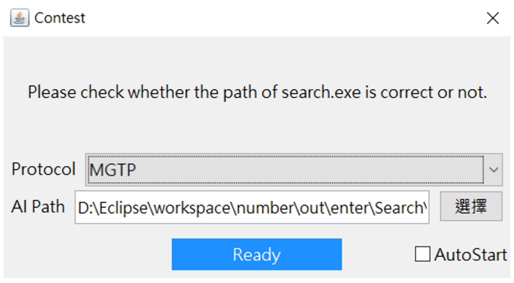
Ready AutoStart

Figure 8: GUI Client Waiting for Opponent

Note: Do not close the program arbitrarily. Please wait patiently for your opponents to join. If you have any issues, please ask relevant personnel.

Once all players have joined the competition, please confirm that the competition program's parameters are correct, as shown in Figure 9.

In the bottom-right corner, there is an AutoStart option, which allows players to automatically confirm. When enabled, the program will automatically confirm after a 10-second countdown on the next competition page.


Contest

Please check whether the path of search.exe is correct or not.

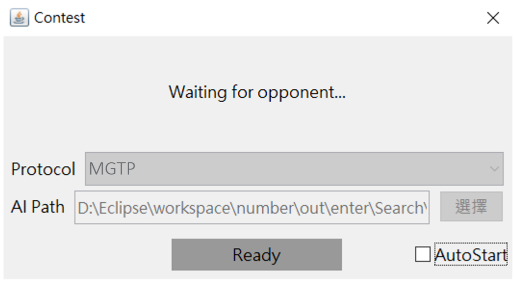
Protocol MGTP

AI Path D:\Eclipse\workspace\number\out\enter\Search\

Figure 9: GUI Client Confirming Competition Program Settings

Note: This setting can be confirmed during the previous steps and does not need to wait until this prompt appears.

After confirming the program parameters, if the opponents have not yet completed their parameter confirmation, a message will appear indicating that you are waiting for the opponents, as shown in Figure 10. If the opponents have already confirmed, the competition will start after a 5-second countdown.


Contest

Waiting for opponent...

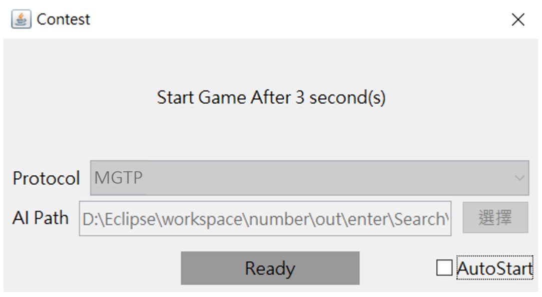
Protocol MGTP

AI Path D:\Eclipse\workspace\number\out\enter\Search\

Figure 10: GUI Client Waiting for Opponent to Confirm Settings

Note: Do not close the program arbitrarily. Please wait patiently for your opponents to confirm. If you have any issues, please ask relevant personnel.

Once all players have confirmed the program, the competition will begin after a 5-second countdown, as shown in Figure 11.



Contest

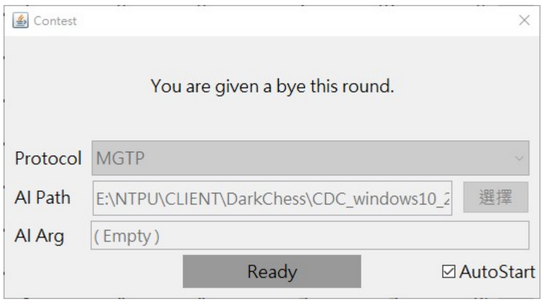
Start Game After 3 second(s)

Protocol: MGTP

AI Path: D:\Eclipse\workspace\number\out\enter\Search\

Figure 11: GUI Client Pre-Competition Countdown

If a message similar to Figure 12, it means you are skipped for this round.



Contest

You are given a bye this round.

Protocol: MGTP

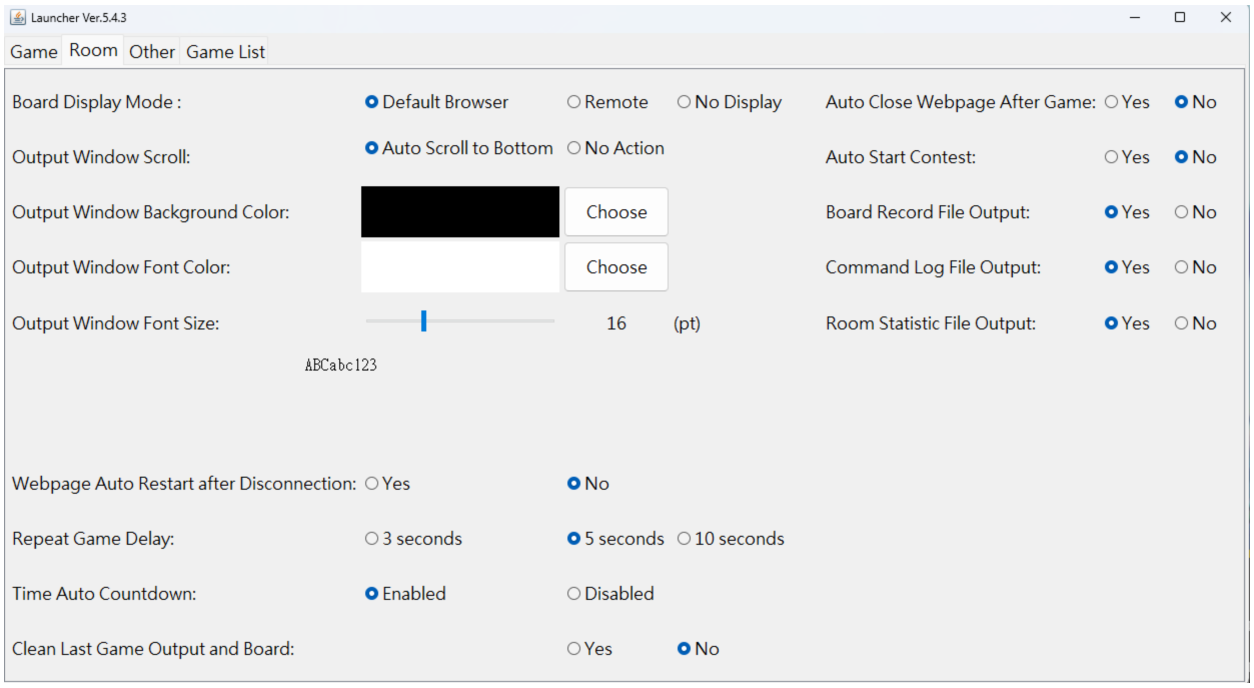
AI Path: E:\NTPU\CLIENT\DarkChess\CDC_windows10_2

AI Arg: (Empty)

Figure 12: GUI Client Skipped Round

6 Room Introduction

6.1 Room Options Settings



Board Display Mode : [✓ Default Browser] Remote No Display
Output Window Scroll: [✓ Auto Scroll to Bottom] No Action
Output Window Background Color: ■ Choose
Output Window Font Color: □ Choose
Output Window Font Size: 16 (pt)
ABCabc123
Webpage Auto Restart after Disconnection: Yes [✓ No]
Repeat Game Delay: 3 seconds [✓ 5 seconds] 10 seconds
Time Auto Countdown: [✓ Enabled] Disabled
Clean Last Game Output and Board: Yes [✓ No]
Auto Close Webpage After Game: Yes [✓ No]
Auto Start Contest: Yes [✓ No]
Board Record File Output: [✓ Yes] No
Command Log File Output: [✓ Yes] No
Room Statistic File Output: [✓ Yes] No

Figure 13: Room Settings Page

1. Board Display Mode: Set the method for opening the board file.
   - Default Browser: The Client opens the default web browser to display the board.
   - Remote: The board is displayed via a remote connection to the Client.
   - No Display: The board is not displayed.
2. Output Window Scroll: Configure the scrollbar behavior in the output window.
   - Auto Scroll to Bottom: Automatically scrolls to the bottom when content is updated.
   - No Action: Keeps the scrollbar position unchanged regardless of updates.
3. Output Window Background Color: Customize the output window's background color.
4. Output Window Font Color: Customize the font color in the output window.
5. Output Window Font Size: Customize the font size in the output window.

6. Webpage Auto Restart after Disconnection: Automatically reopens the webpage if a disconnection is detected.

7. Repeat Game Delay: Sets the delay time between games.

8. Time Auto Countdown:

   - Enabled: Automatically counts down the remaining time.
   - Disabled: Only updates the remaining time when receiving it from the Server for each move.

9. Clear Last Game Output and Board: Determines whether to clear the output window and board display after the specified number of games in this room.

10. Auto Close Webpage After Game: Determines whether to close the webpage after the specified number of games in this room.

11. Automatic Start contest: Starts the competition automatically after a 10-second countdown, without requiring further player confirmation.

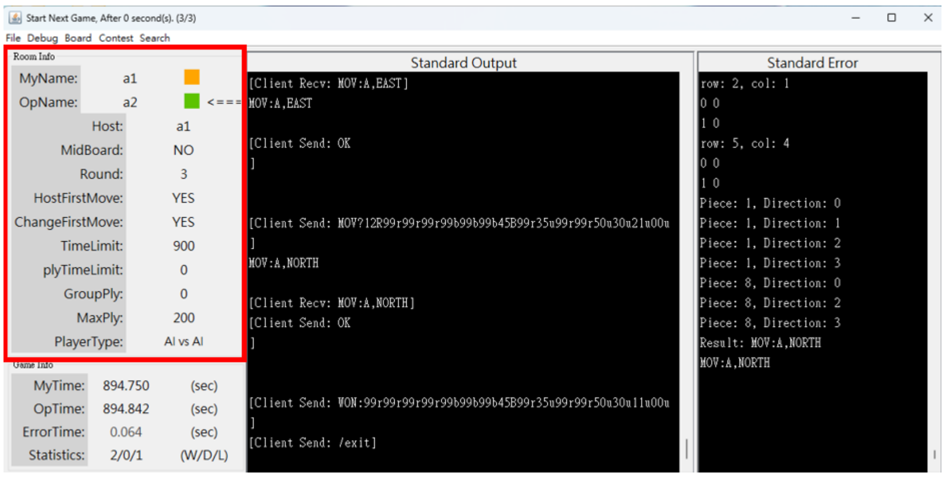
12. Board Record File Output: Saves a record file (.txt) of the game in the Board folder.

13. Command Log File Output: Saves a command record file (.command) of the game in the Board folder.

14. Room Statistic File Output: Saves a statistics file (.info) of the game in the Board folder.

6.2 Room Page

6.2.1 Game Parameters


Start Next Game, After 0 second(s). (3/3)
File Debug Board Contest Search
Room Info
Game Info
Standard Output
Standard Error

MyName:	a1	🟧
OpName:	a2	🟩 <===
Host:	a1	
MidBoard:	NO	
Round:	3	
HostFirstMove:	YES	
ChangeFirstMove:	YES	
TimeLimit:	900	
plyTimeLimit:	0	
GroupPly:	0	
MaxPly:	200	
PlayerType:	AI vs AI	

MyTime:	894.750	(sec)
OpTime:	894.842	(sec)
ErrorTime:	0.064	(sec)
Statistics:	2/0/1	(W/D/L)

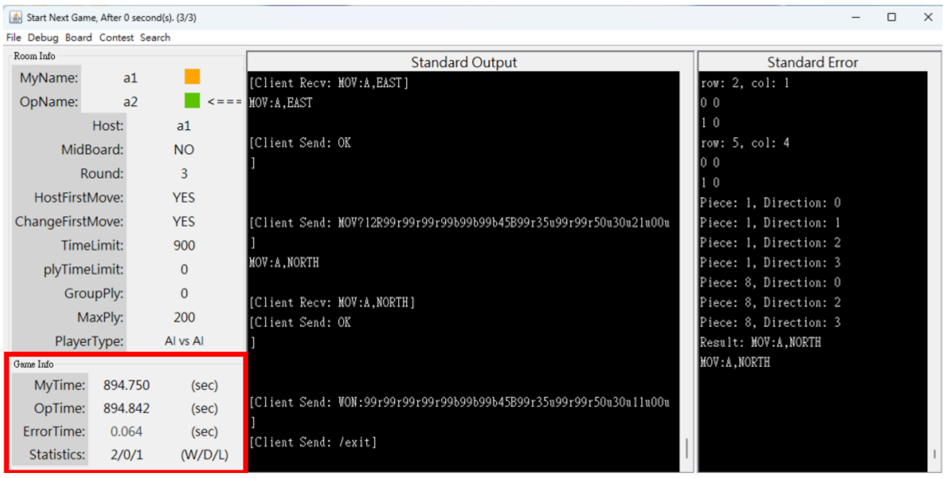
```
[Client Recv: MOV:A,EAST]
MOV:A,EAST

[Client Send: OK
]



[Client Send: MOV?12R99r99r99r99b99b99b45B99r35u99r99r50u30u21u00u
]
MOV:A,NORTH

[Client Recv: MOV:A,NORTH]
[Client Send: OK
]



[Client Send: WON:99r99r99r99r99b99b99b45B99r35u99r99r50u30u11u00u
]
[Client Send: /exit]
```

```
row: 2, col: 1
0 0
1 0
row: 5, col: 4
0 0
1 0
Piece: 1, Direction: 0
Piece: 1, Direction: 1
Piece: 1, Direction: 2
Piece: 1, Direction: 3
Piece: 8, Direction: 0
Piece: 8, Direction: 2
Piece: 8, Direction: 3
Result: MOV:A,NORTH
MOV:A,NORTH
```

Figure 14: Game Parameters Display Area

In the red box in Figure 14, the upper section shows the account names and representative colors of both players, and an arrow indicating whose turn it is. The lower section displays various game parameters for this room.

6.2.2 Game Status


Start Next Game, After 0 second(s). (3/3)

File Debug Board Contest Search

Room Info

MyName:	a1	🟧
OpName:	a2	🟩 <===
Host:	a1	
MidBoard:	NO	
Round:	3	
HostFirstMove:	YES	
ChangeFirstMove:	YES	
TimeLimit:	900	
plyTimeLimit:	0	
GroupPly:	0	
MaxPly:	200	
PlayerType:	AI vs AI	

Game Info

MyTime:	894.750	(sec)
OpTime:	894.842	(sec)
ErrorTime:	0.064	(sec)
Statistics:	2/0/1	(W/D/L)

Standard Output

```
[Client Recv: MOV:A,EAST]
MOV:A,EAST

[Client Send: OK
]


[Client Send: MOV?12R99r99r99r99b99b99b45B99r35u99r99r50u30u21u00u
]
MOV:A,NORTH

[Client Recv: MOV:A,NORTH]
[Client Send: OK
]


[Client Send: WON:99r99r99r99r99b99b99b45B99r35u99r99r50u30u11u00u
]
[Client Send: /exit]
```

Standard Error

```
row: 2, col: 1
0 0
1 0
row: 5, col: 4
0 0
1 0
Piece: 1, Direction: 0
Piece: 1, Direction: 1
Piece: 1, Direction: 2
Piece: 1, Direction: 3
Piece: 8, Direction: 0
Piece: 8, Direction: 2
Piece: 8, Direction: 3
Result: MOV:A,NORTH
MOV:A,NORTH
```

Figure 15: Game Status Display Area

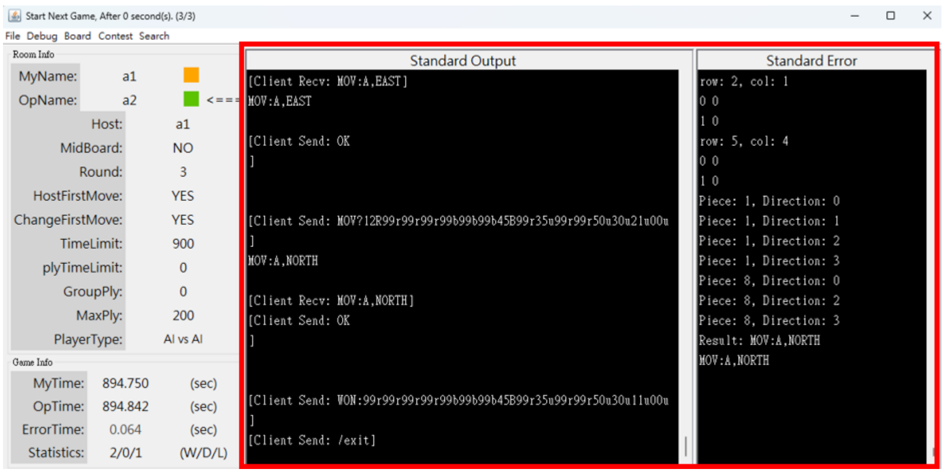
The red box in Figure 15 includes both players' remaining time, transmission error time, and win-draw-loss statistics.

- MyTime: Remaining time for the user.
- OpTime: Remaining time for the opponent.
- ErrorTime: Transmission error time. Let X and Y represent the time taken by the AI for the move as recorded by the Server and Client, respectively. The calculation formula is:

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

  This does not represent actual network latency, but can serve as a reference for network stability.
- Statistics: The count of wins/draws/losses.

6.2.3 Output Window


Start Next Game, After 0 second(s). (3/3)

File Debug Board Contest Search

Room Info

MyName:	a1	🟧
OpName:	a2	🟩 <===
Host:	a1	
MidBoard:	NO	
Round:	3	
HostFirstMove:	YES	
ChangeFirstMove:	YES	
TimeLimit:	900	
plyTimeLimit:	0	
GroupPly:	0	
MaxPly:	200	
PlayerType:	AI vs AI	

Game Info

MyTime:	894.750	(sec)
OpTime:	894.842	(sec)
ErrorTime:	0.064	(sec)
Statistics:	2/0/1	(W/D/L)

Standard Output

```
[Client Recv: MOV:A,EAST]
MOV:A,EAST

[Client Send: OK
]


[Client Send: MOV?12R99r99r99r99b99b99b45B99r35u99r99r50u30u21u00u
]
MOV:A,NORTH

[Client Recv: MOV:A,NORTH]
[Client Send: OK
]


[Client Send: WON:99r99r99r99r99b99b99b45B99r35u99r99r50u30u11u00u
]
[Client Send: /exit]
```

Standard Error

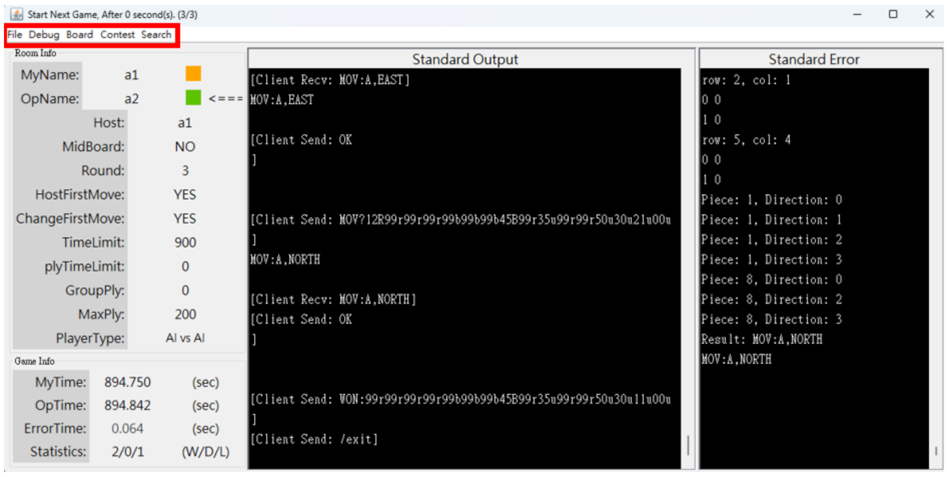
```
row: 2, col: 1
0 0
1 0
row: 5, col: 4
0 0
1 0
Piece: 1, Direction: 0
Piece: 1, Direction: 1
Piece: 1, Direction: 2
Piece: 1, Direction: 3
Piece: 8, Direction: 0
Piece: 8, Direction: 2
Piece: 8, Direction: 3
Result: MOV:A,NORTH
MOV:A,NORTH
```

Figure 16: Program Output Area

In the red box in Figure 16, the left section displays the program's standard output (stdout), the right section displays the standard error output (stderr).

> Note: If outputs appear in batches rather than in real-time, refer to QA 10.3.

6.2.4 Function Menu


Start Next Game, After 0 second(s). (3/3)
File Debug Board Contest Search
Room Info
Game Info
Standard Output
Standard Error

MyName:	a1	🟧
OpName:	a2	🟩 <===
Host:	a1	
MidBoard:	NO	
Round:	3	
HostFirstMove:	YES	
ChangeFirstMove:	YES	
TimeLimit:	900	
plyTimeLimit:	0	
GroupPly:	0	
MaxPly:	200	
PlayerType:	AI vs AI	

MyTime:	894.750	(sec)
OpTime:	894.842	(sec)
ErrorTime:	0.064	(sec)
Statistics:	2/0/1	(W/D/L)

```
[Client Recv: MOV:A,EAST]
MOV:A,EAST

[Client Send: OK
]



[Client Send: MOV?12R99r99r99r99b99b99b45B99r35u99r99r50u30u21u00u
]
MOV:A,NORTH

[Client Recv: MOV:A,NORTH]
[Client Send: OK
]



[Client Send: WON:99r99r99r99r99b99b99b45B99r35u99r99r50u30u11u00u
]
[Client Send: /exit]
```

```
row: 2, col: 1
0 0
1 0
row: 5, col: 4
0 0
1 0
Piece: 1, Direction: 0
Piece: 1, Direction: 1
Piece: 1, Direction: 2
Piece: 1, Direction: 3
Piece: 8, Direction: 0
Piece: 8, Direction: 2
Piece: 8, Direction: 3
Result: MOV:A,NORTH
MOV:A,NORTH
```

Figure 17: Function Menu Area

In the red box in Figure 17:

- File
  - Room Statistics: Displays statistical data for all games in a single room (see Section 8.6).
  - Game Record: Displays the content of a single game (see Section 8.7).
- Debug
  - AutoScroll: Determines whether the scrollbar automatically scrolls to the bottom when content is updated.
  - Increase Font Size: Increases the font size in the debug area.
  - Decrease Font Size: Decreases the font size in the debug area.
  - Pop Debug Window: Separates the debug area into an independent window for easier debugging.
- Board
  - Open Board: Opens the board webpage.
- Contest
  - AutoStart: Automatically prepares and starts the competition in contest mode.
- Search
  - About: Displays information related to the current Search feature.

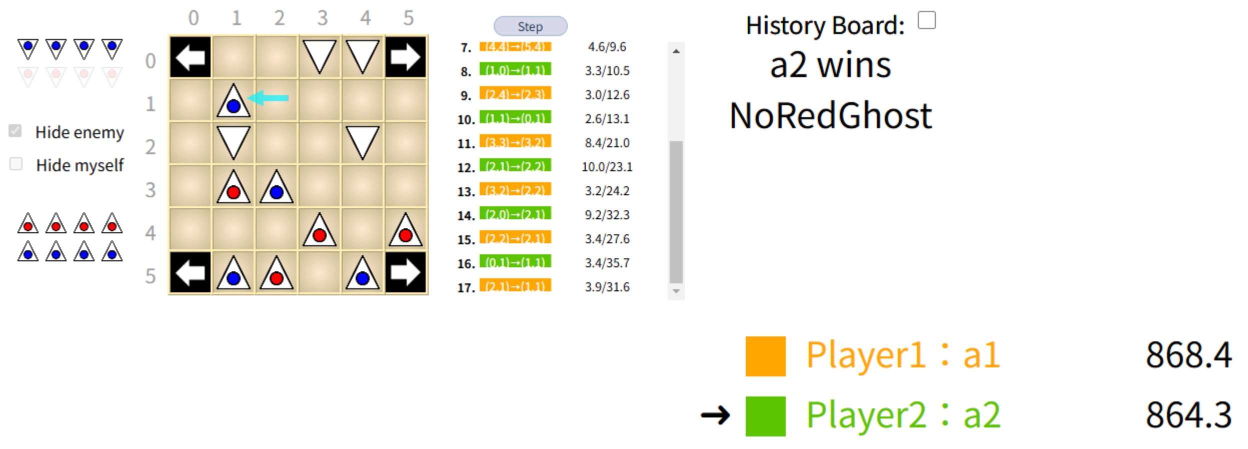
7 Board Introduction

Currently, the game board is presented via a webpage to enhance compatibility. When the board option is set to use the default browser, the Client will automatically open the board file and establish the connection for the user. Below is an introduction to the various sections and functionalities of the board.

Recommendation: Use Google Chrome.

7.1 Game Board Overview

The board includes the following sections: piece status, board, move history, and results, as shown in Figure 18.



7.	(4,4)→(5,4)	4.6/9.6
8.	(1,0)→(1,1)	3.3/10.5
9.	(2,4)→(2,3)	3.0/12.6
10.	(1,1)→(0,1)	2.6/13.1
11.	(3,3)→(3,2)	8.4/21.0
12.	(2,1)→(2,2)	10.0/23.1
13.	(3,2)→(2,2)	3.2/24.2
14.	(2,0)→(2,1)	9.2/32.3
15.	(2,2)→(2,1)	3.4/27.6
16.	(0,1)→(1,1)	3.4/35.7
17.	(2,1)→(1,1)	3.9/31.6

a2 wins

NoRedGhost

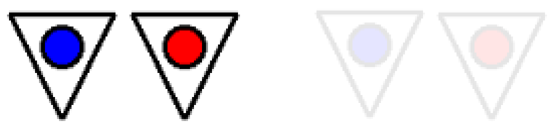
	🟧 Player1 : a1	868.4
→	🟩 Player2 : a2	864.3

Piece Status Board History Move Result

Figure 18: Display Page

7.1.1 Piece Status

- The left side of the webpage displays the current status of both players' pieces, as shown in Figure 19.

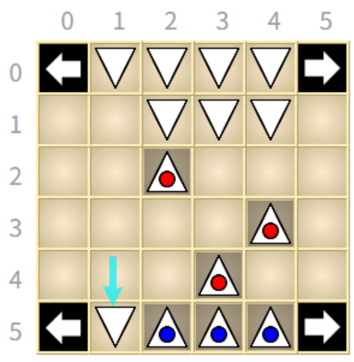


(a) Alive Pieces (b) Dead Pieces

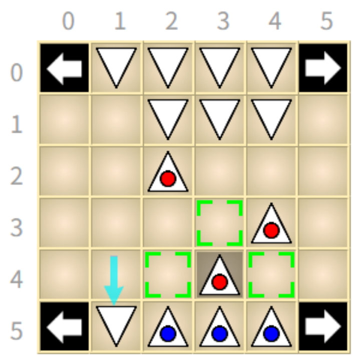
Figure 19: Piece Status

7.1.2 Board Display

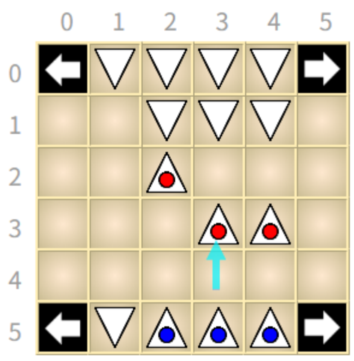
- During your turn, movable pieces will be highlighted in gray at the bottom, as shown in Figure 20(a).
- After selecting a gray-highlighted piece, its possible moves will be displayed as light green squares. Click a square to move the piece, as shown in Figure 20(b) and Figure 20(c).



(a) Start of Turn



(b) Select A Piece

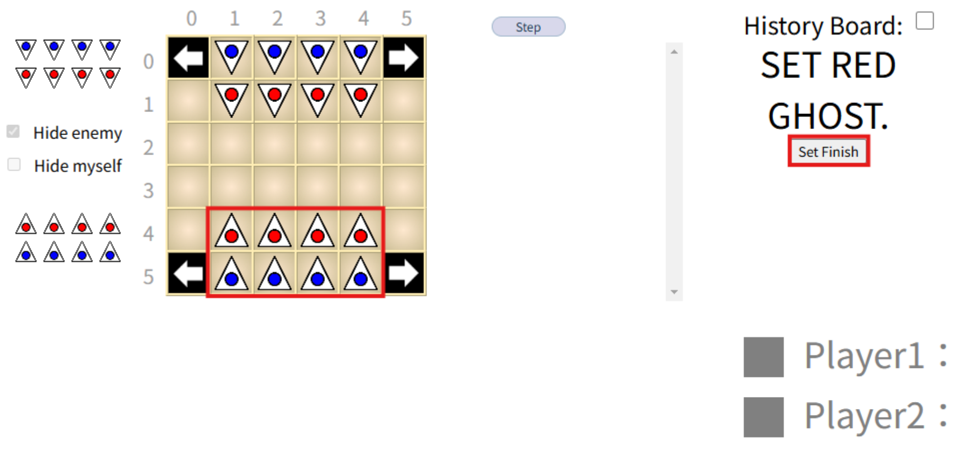


(c) Move A Piece

Figure 20: Board Display and Movement

7.1.3 Manual Mode

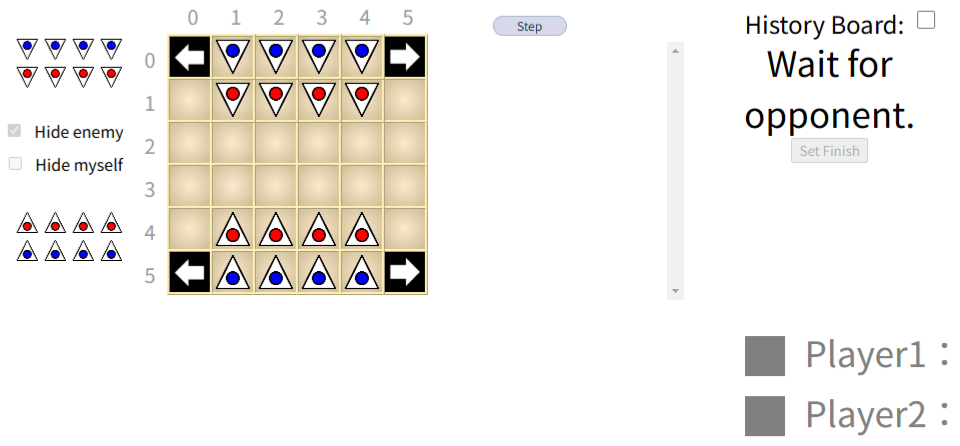
- In manual mode, players must configure their board setup before the game starts, as shown in Figure 21.
- Players can adjust their board setup by clicking two of their pieces to swap their positions.
- Once configuration is complete, click the ”Set Finish” button to confirm.



SET RED GHOST.

Figure 21: Board Setup Screen

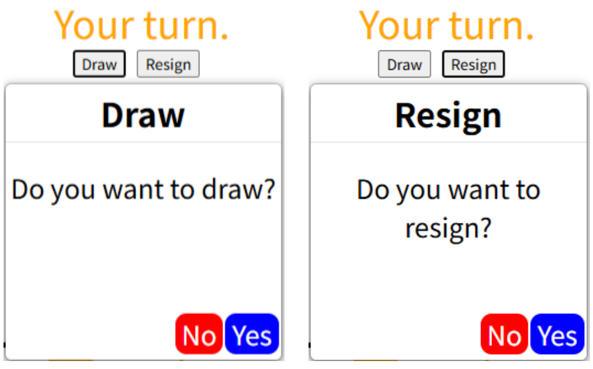
- If the opponent has not completed their setup, the screen will display a waiting status, as shown in Figure 22.
- When both players are ready, the game will start automatically.



Wait for opponent.

Figure 22: Waiting for Opponent to Set Board

- During your turn, you can propose a draw, resign, or undo a move, as shown in Figure 23(a)(b).
- In manual mode, the results section will display your turn with your assigned color. For example, in geister, the first player ("Your Turn") is shown in orange, and the second player is shown in green, as seen in Figure 24(a) and Figure 24(b).



Draw

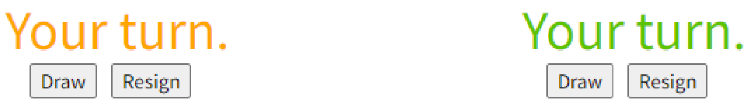
Do you want to draw?

Resign

Do you want to resign?

(a) Propose Draw (b) Resign

Figure 23: Options During Your Turn



(a) First Player (b) Second Player

Figure 24: Manual Mode Color Explanation

7.2 Game Records

Follow the steps in Section 8.7 to access game records.

7.2.1 Instructions

- After opening the page, select a history file, as shown in Figure25.
- Use the keyboard keys 'w', 's', 'a', 'd' to control forward and backward navigation through the moves.
- The names and colors of the first and second players are shown on the file, as displayed in Figure 26. The historical board is shown in Figure 27.
- The upper text box shows standard output, and the lower text box shows standard error output. These update with each move and can auto-wrap or be disabled, as shown in Figure 28.

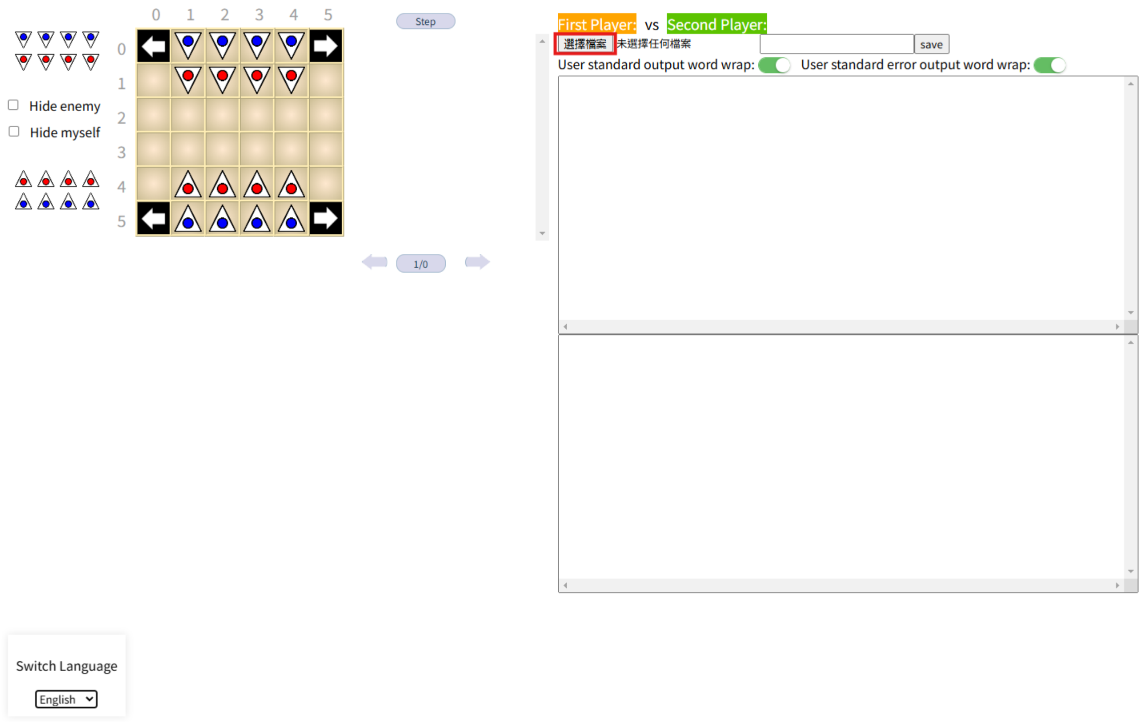


Figure 25: Display Page

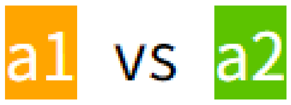
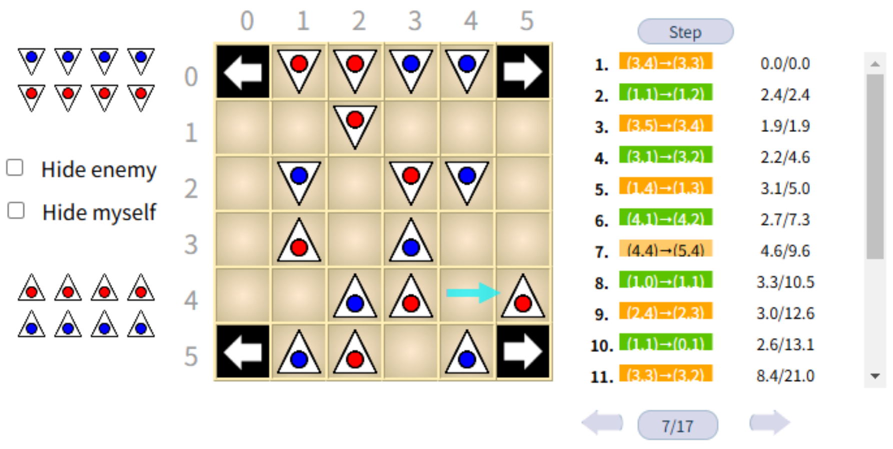
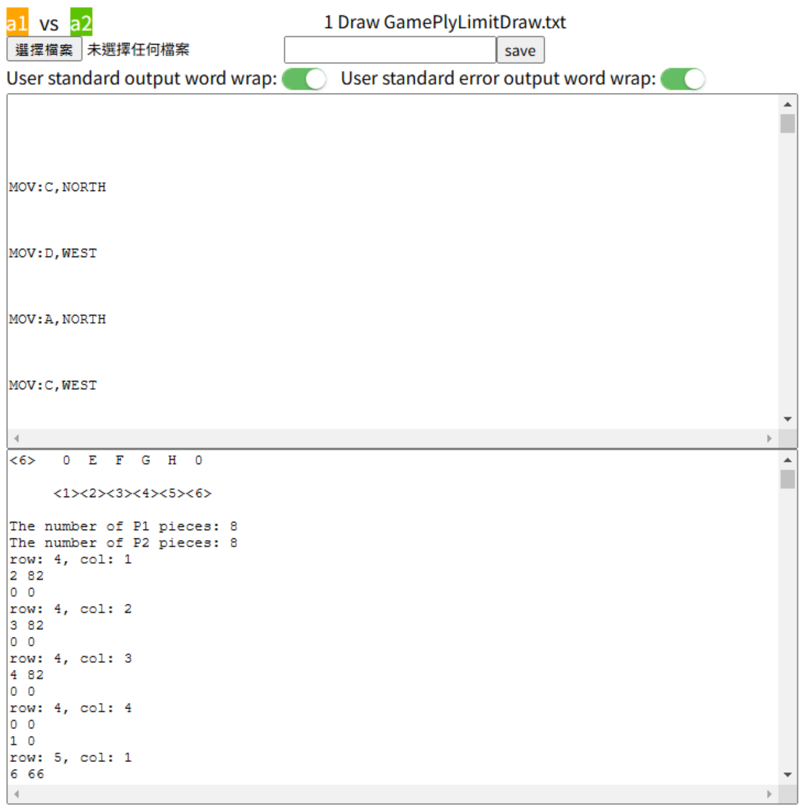


Figure 26: First Player A1 and Second Player A2



```
1.  (3,4)→(3,3)   0.0/0.0
2.  (1,1)→(1,2)   2.4/2.4
3.  (3,5)→(3,4)   1.9/1.9
4.  (3,1)→(3,2)   2.2/4.6
5.  (1,4)→(1,3)   3.1/5.0
6.  (4,1)→(4,2)   2.7/7.3
7.  (4,4)→(5,4)   4.6/9.6
8.  (1,0)→(1,1)   3.3/10.5
9.  (2,4)→(2,3)   3.0/12.6
10. (1,1)→(0,1)   2.6/13.1
11. (3,3)→(3,2)   8.4/21.0
```

Figure 27: Board and Move History



User standard output word wrap: ✓ User standard error output word wrap: ✓

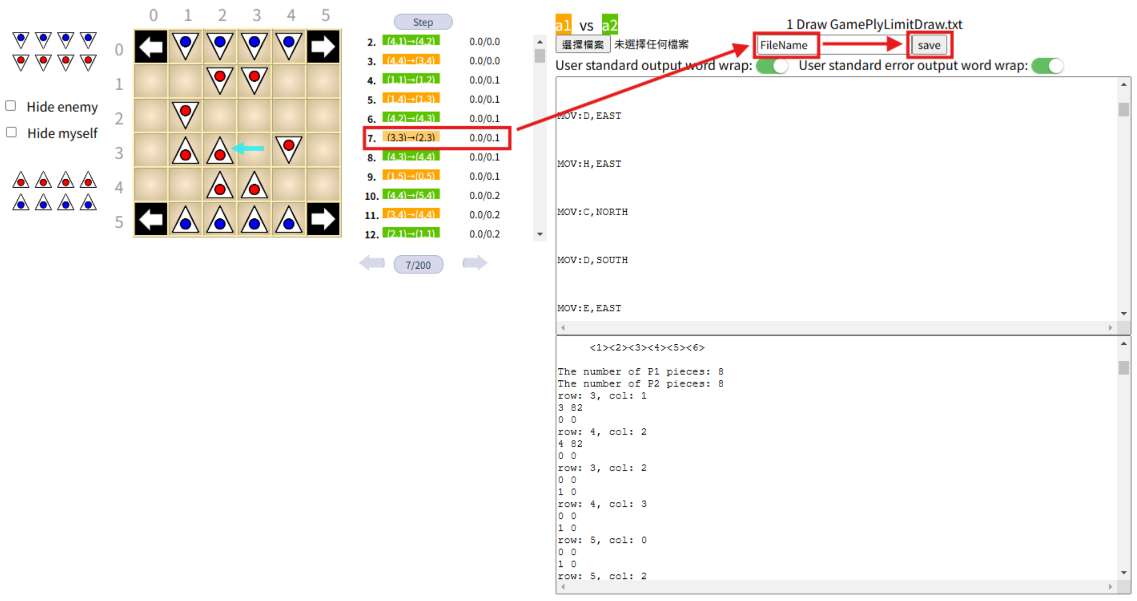
```
MOV:C,NORTH


MOV:D,WEST


MOV:A,NORTH


MOV:C,WEST
```

```
<6>   0  E  F  G  H  0

     <1><2><3><4><5><6>

The number of P1 pieces: 8
The number of P2 pieces: 8
row: 4, col: 1
2 82
0 0
row: 4, col: 2
3 82
0 0
row: 4, col: 3
4 82
0 0
row: 4, col: 4
0 0
1 0
row: 5, col: 1
6 66
```

Figure 28: Standard Output and Standard Error Output

7.2.2 Creating Mid-Game Files

- Create midgame files needed for midgame play (see Section 8.4). These files record all moves from the initial board to the selected board state. These files can be used to start a new game with a custom board, as shown in Figure 30.
- Steps: Shown as Figure 29.
  1. Select Move: Click a move to set it as the last step in the mid-game.
  2. Enter File Name: Provide a name for the mid-game file.
  3. Save File: Click the ”Save” button to download the file.



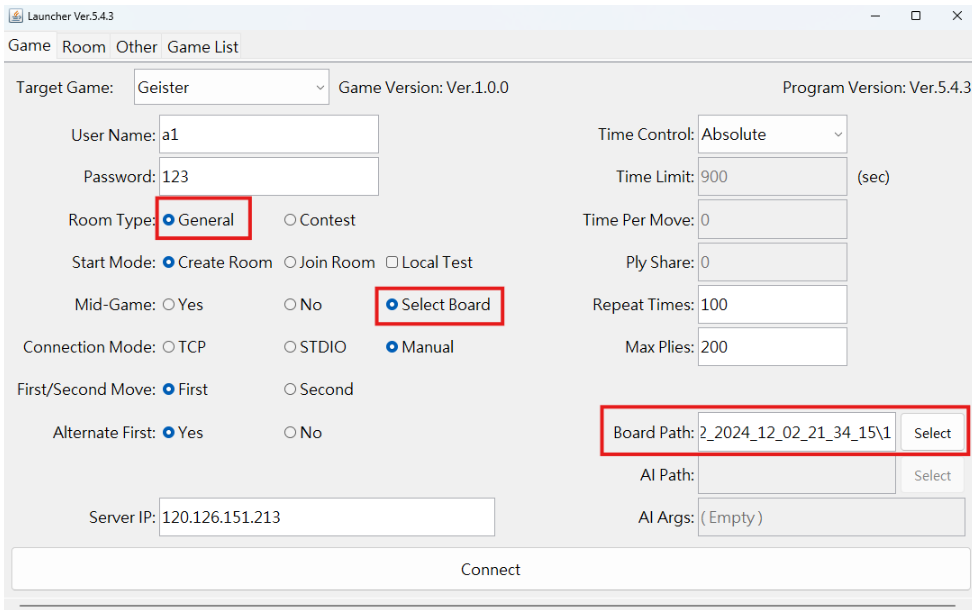
Hide enemy
Hide myself
0 1 2 3 4 5
Step
7/200
a1 vs a2
1 Draw GamePlyLimitDraw.txt
選擇檔案
未選擇任何檔案
FileName
save
User standard output word wrap:
User standard error output word wrap:

2.	(4,1)→(4,2)	0.0/0.0
3.	(4,4)→(3,4)	0.0/0.0
4.	(1,1)→(1,2)	0.0/0.1
5.	(1,4)→(1,3)	0.0/0.1
6.	(4,2)→(4,3)	0.0/0.1
7.	(3,3)→(2,3)	0.0/0.1
8.	(4,3)→(4,4)	0.0/0.1
9.	(1,5)→(0,5)	0.0/0.1
10.	(4,4)→(5,4)	0.0/0.2
11.	(3,4)→(4,4)	0.0/0.2
12.	(2,1)→(1,1)	0.0/0.2

```
MOV:D,EAST


MOV:H,EAST


MOV:C,NORTH


MOV:D,SOUTH


MOV:E,EAST
```

```
    <1><2><3><4><5><6>

The number of P1 pieces: 8
The number of P2 pieces: 8
row: 3, col: 1
3 82
0 0
row: 4, col: 2
4 82
0 0
row: 3, col: 2
0 0
1 0
row: 4, col: 3
0 0
1 0
row: 5, col: 0
0 0
1 0
row: 5, col: 2
```

Figure 29: Steps for Creating a Mid-Game File

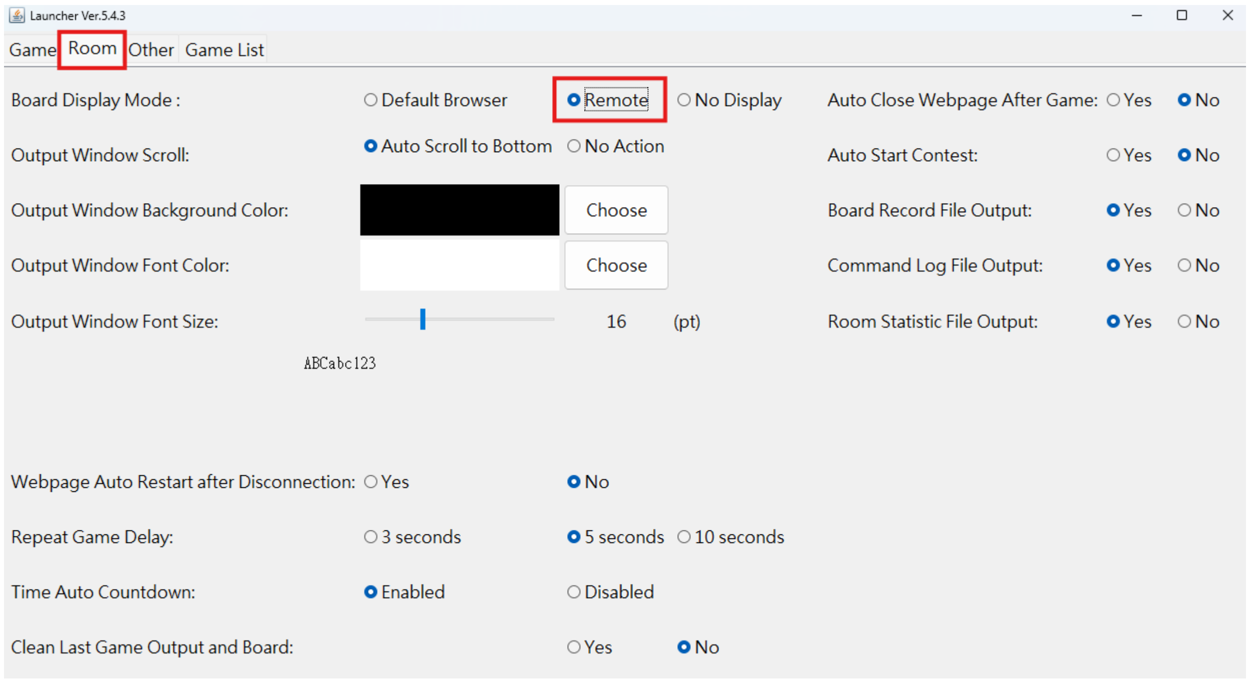


Target Game: Geister
Game Version: Ver.1.0.0
Program Version: Ver.5.4.3
User Name: a1
Password: 123
Room Type: [✓ General] Contest
Start Mode: [✓ Create Room] Join Room Local Test
Mid-Game: Yes No [✓ Select Board]
Connection Mode: TCP STDIO [✓ Manual]
First/Second Move: [✓ First] Second
Alternate First: [✓ Yes] No
Server IP: 120.126.151.213
Time Control: Absolute
Time Limit: 900 (sec)
Time Per Move: 0
Ply Share: 0
Repeat Times: 100
Max Plies: 200
Board Path: 2_2024_12_02_21_34_15\1
AI Path:
AI Args: (Empty)

Figure 30: Start a new game using a custom board setup

7.3 Remote Page

- Assume the Client's IP is 120.126.151.3. After creating or joining a room and completing the settings as shown in Figure 31, the Client interface will display as shown in Figure 32. The red box indicates the Client's Port, assumed here to be 50618.
- On the spectating computer, follow the instructions in Section 8.8 to select the remote page, which will display the interface shown in Figure 33. Then enter the Client's IP (120.126.151.3) and Port (50618) to spectate.



Launcher Ver.5.4.3

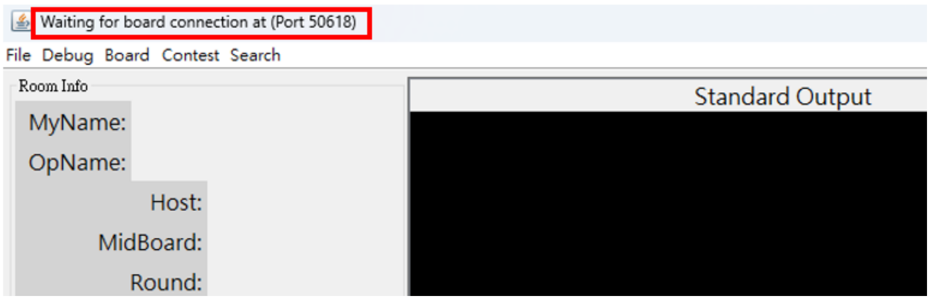
Game | Room | Other | Game List

Board Display Mode :	Default Browser [✓ Remote] No Display		Auto Close Webpage After Game:	Yes [✓ No]
Output Window Scroll:	[✓ Auto Scroll to Bottom] No Action		Auto Start Contest:	Yes [✓ No]
Output Window Background Color:	■	Choose	Board Record File Output:	[✓ Yes] No
Output Window Font Color:	□	Choose	Command Log File Output:	[✓ Yes] No
Output Window Font Size:	16 (pt)		Room Statistic File Output:	[✓ Yes] No

ABCabc123

Webpage Auto Restart after Disconnection:	Yes [✓ No]
Repeat Game Delay:	3 seconds [✓ 5 seconds] 10 seconds
Time Auto Countdown:	[✓ Enabled] Disabled
Clean Last Game Output and Board:	Yes [✓ No]

Figure 31: Room Settings Page



Waiting for board connection at (Port 50618)

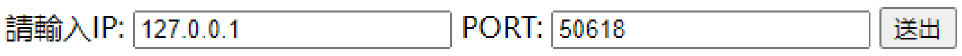
File Debug Board Contest Search

Room Info

MyName:
OpName:
Host:
MidBoard:
Round:

Standard Output

Figure 32: Port Display Page

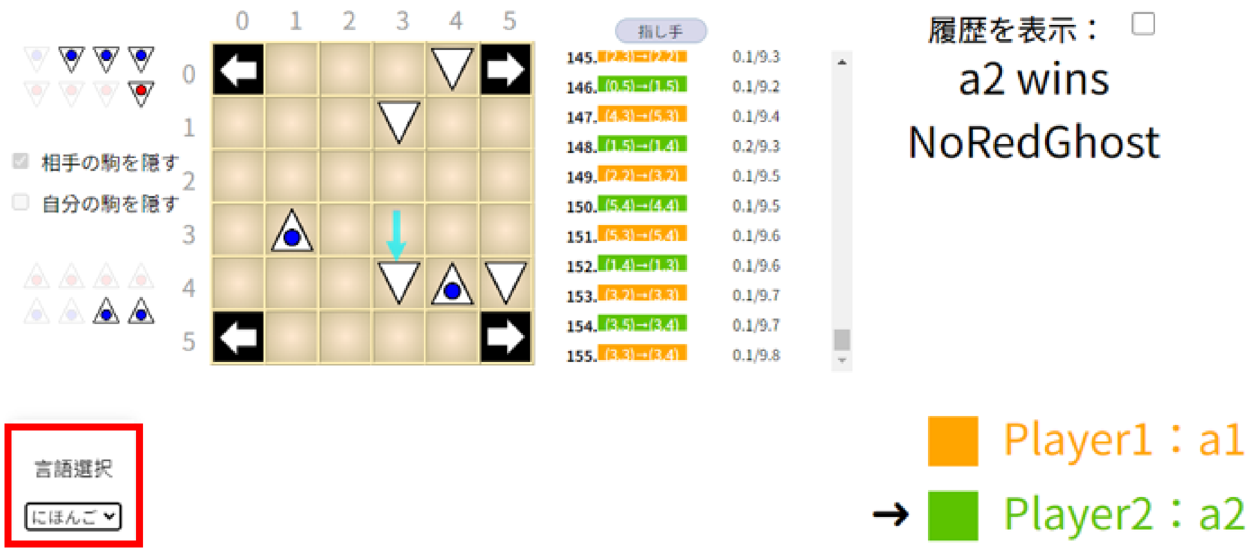


請輸入IP: 127.0.0.1 PORT: 50618 送出

Figure 33: Remote Page Interface

7.4 Language Switching

The game board and game record pages both have language options located at the bottom-left corner, as shown in Figure 34 and 35. Users can switch between Chinese, English, and Japanese based on their preference.



```
145. (2,3)→(2,2)   0.1/9.3
146. (0,5)→(1,5)   0.1/9.2
147. (4,3)→(5,3)   0.1/9.4
148. (1,5)→(1,4)   0.2/9.3
149. (2,2)→(3,2)   0.1/9.5
150. (5,4)→(4,4)   0.1/9.5
151. (5,3)→(5,4)   0.1/9.6
152. (1,4)→(1,3)   0.1/9.6
153. (3,2)→(3,3)   0.1/9.7
154. (3,5)→(3,4)   0.1/9.7
155. (3,3)→(3,4)   0.1/9.8
```

履歴を表示：

a2 wins

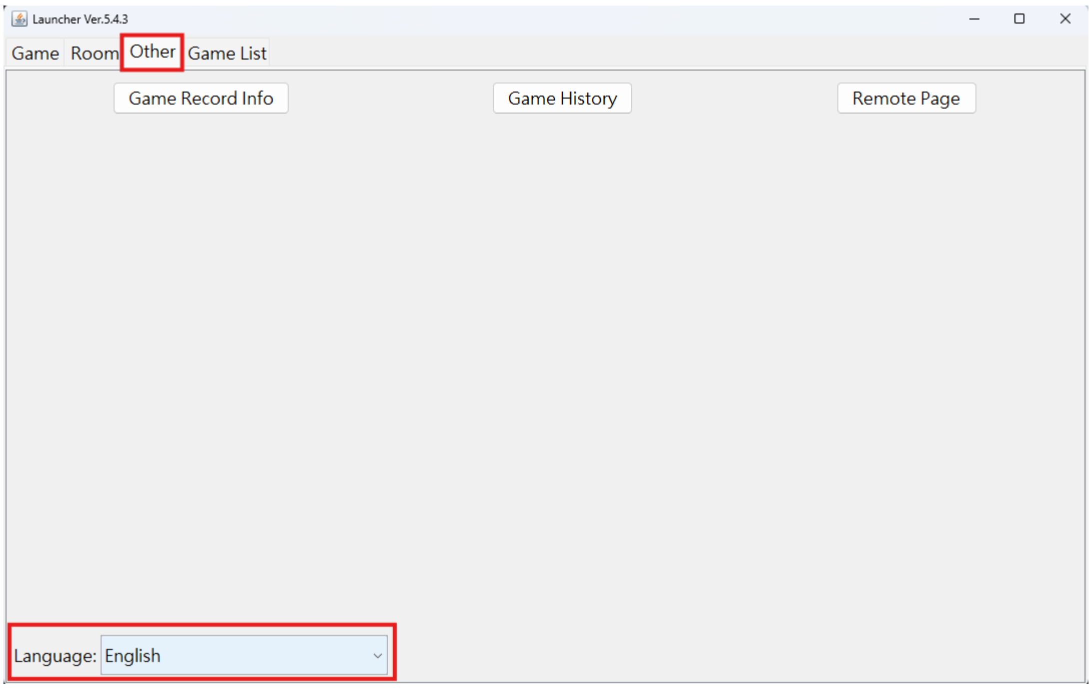
NoRedGhost

言語選択: にほんご

🟧 Player1：a1

→ 🟩 Player2：a2

Figure 34: Language Options on the Open and Enter Pages



Language: English

Figure 35: Language Options on the Launcher

8 Features Overview

8.1 Resign

When the game's outcome is nearly decided but the win condition is not yet met, continuing the game can waste time. This feature allows a player or AI to resign and end the match early. There are two ways to resign:

- Manual (defined in Section 7.1.3)
- AI

Note: The resign feature is available in all game modes.

Resign	Online Server	Local Testing
AI vs AI	Yes	Yes
AI vs Human	Yes	Yes
Human vs Human	Yes	Yes

8.2 Draw Offer

During the game, the current player can offer a draw by using the Draw Offer function to ask the opponent if they agree to a draw. If the opponent accepts, the game ends in a draw. Otherwise, the game continues. Additionally, if the opponent does not respond, the player offering the draw does not have to wait and can make their next move.

The operation of the draw offer is explained in Section 7.1.3.

Note: The time spent during the draw offer process counts towards the thinking time, so be mindful of the time limit.

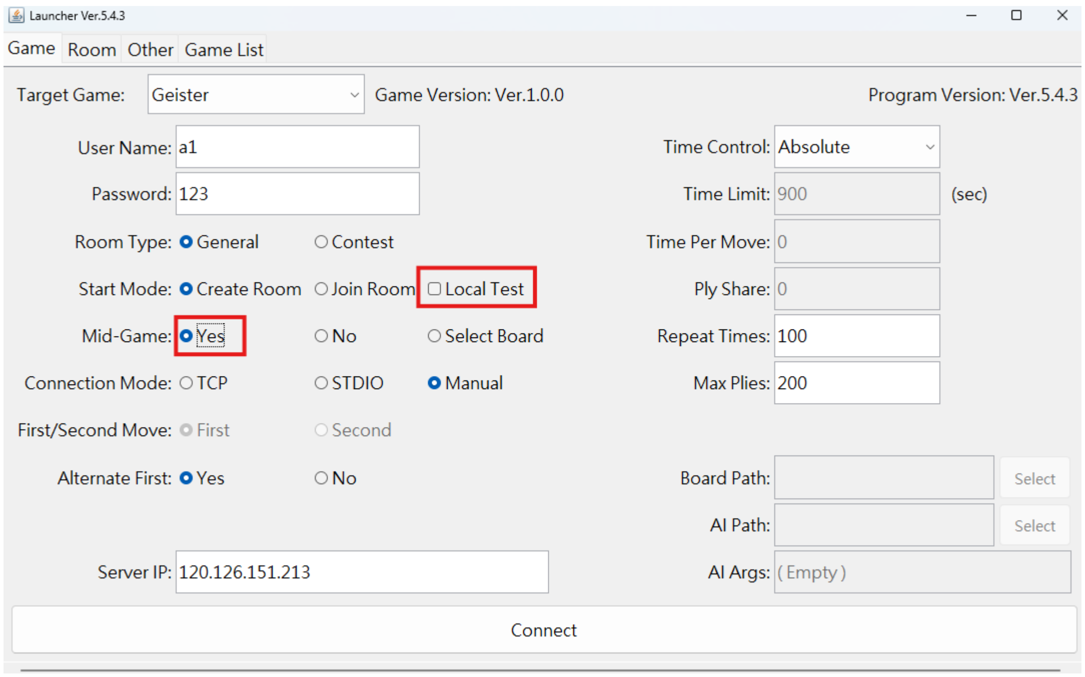
Note: For fairness, the draw offer feature is only supported under certain conditions.

Draw Offer	Online Server	Local Testing
AI vs AI	No	No
AI vs Human	No	No
Human vs Human	Yes	No

8.3 Mid-Game Recovery After Disconnection

The Mid-Game Recovery feature allows players to resume the game after a network disconnection. However, this feature cannot be used during local testing.

Figure 36 shows an example of the mid-game recovery settings, which require the room host to select Yes in the Mid-Game options.



Target Game: Geister
Game Version: Ver.1.0.0
Program Version: Ver.5.4.3
User Name: a1
Password: 123
Room Type: [✓ General] Contest
Start Mode: [✓ Create Room] Join Room ☐ Local Test
Mid-Game: [✓ Yes] No Select Board
Connection Mode: TCP STDIO [✓ Manual]
First/Second Move: First Second
Alternate First: [✓ Yes] No
Server IP: 120.126.151.213
Time Control: Absolute
Time Limit: 900 (sec)
Time Per Move: 0
Ply Share: 0
Repeat Times: 100
Max Plies: 200
Board Path:
AI Path:
AI Args: (Empty)

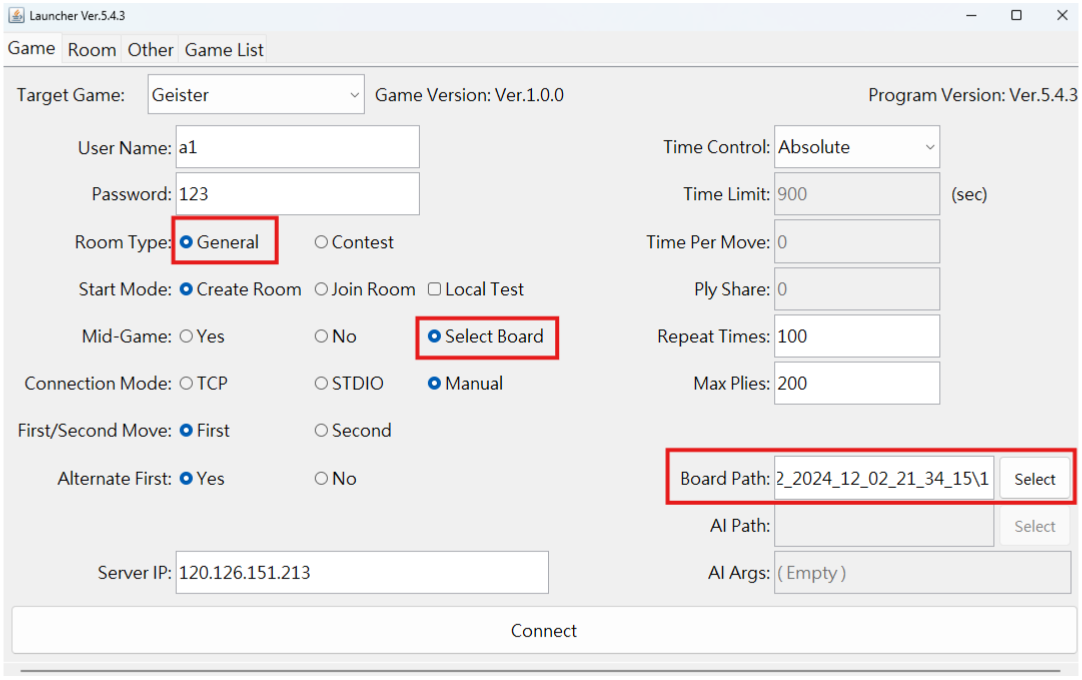
Figure 36: Example of Mid-Game Recovery Settings (interface from Figure 3)

8.4 Mid-Game Continuation

The Mid-Game Continuation feature enables players to start a game from a mid-game position, which can be used for AI debugging.

The method to create a mid-game position is explained in Section 7.2.2.

Figure 37 shows an example of mid-game continuation settings, where the room host must select the Select Board option in Mid-Game, and specify the path to the mid-game file.



Target Game: Geister
Game Version: Ver.1.0.0
Program Version: Ver.5.4.3
User Name: a1
Password: 123
Room Type: [✓ General] Contest
Start Mode: [✓ Create Room] Join Room ☐ Local Test
Mid-Game: Yes No [✓ Select Board]
Connection Mode: TCP STDIO [✓ Manual]
First/Second Move: [✓ First] Second
Alternate First: [✓ Yes] No
Server IP: 120.126.151.213
Time Control: Absolute
Time Limit: 900 (sec)
Time Per Move: 0
Ply Share: 0
Repeat Times: 100
Max Plies: 200
Board Path: 2_2024_12_02_21_34_15\1
AI Path:
AI Args: (Empty)

Figure 37: Example of Mid-Game Settings

8.5 Local Testing

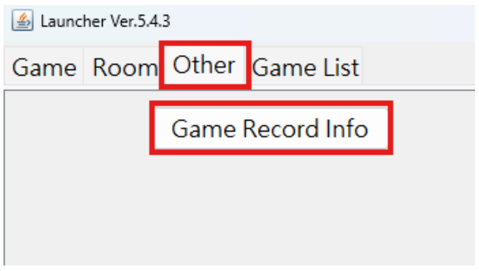
The Local Testing feature provides an environment where players can test AI programs without needing an internet connection. It also offers debugging tools, although some features are limited or only available during local testing. Details are defined in the respective sections for each feature.

As shown in Figure 36, both players must check the Local Test box in the start mode to enable local testing.

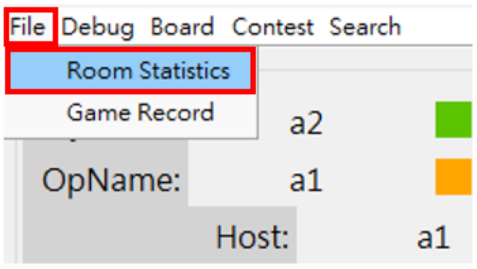
8.6 Room Statistics

Room statistics record information such as game results, win/loss, and time parameters for all games in a room. There are two ways to access room statistics:

- External Access: As shown in Figure 38(a), switch to the Other tab after opening the client and click the Room Statistics button to view room statistics before starting a game.
- In-Game Access: As shown in Figure 38(b), click File in the room page and select Room Statistics to view current or past room statistics at any time during the game.



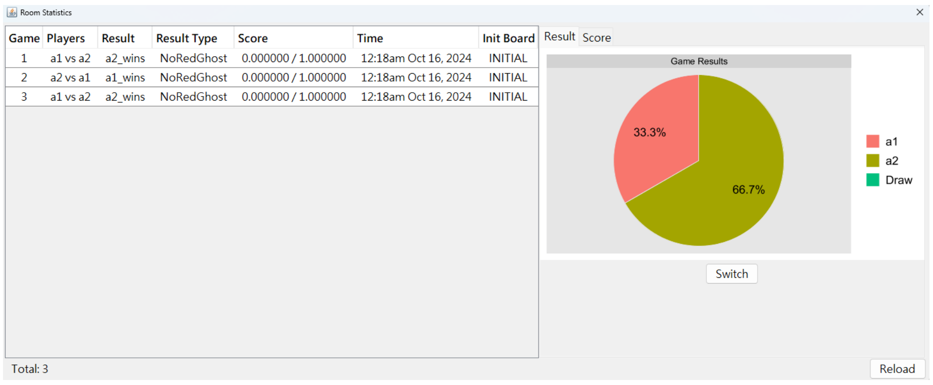
(a) External Access



(b) In-Game Access

Figure 38: Ways to Access Room Statistics

In Figure 39, the left table lists data for each game, while the right section displays statistical charts for win rates and draw rates between players. The Switch button allows toggling between game counts and rates, and the Reload button updates the data with the latest game information.



Game	Players	Result	Result Type	Score	Time	Init Board
1	a1 vs a2	a2_wins	NoRedGhost	0.000000 / 1.000000	12:18am Oct 16, 2024	INITIAL
2	a2 vs a1	a1_wins	NoRedGhost	0.000000 / 1.000000	12:18am Oct 16, 2024	INITIAL
3	a1 vs a2	a2_wins	NoRedGhost	0.000000 / 1.000000	12:18am Oct 16, 2024	INITIAL

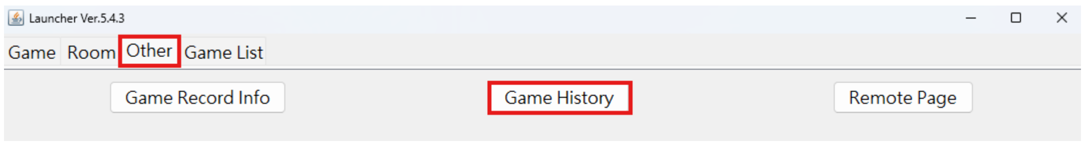
Total: 3

Figure 39: Room Statistics Page

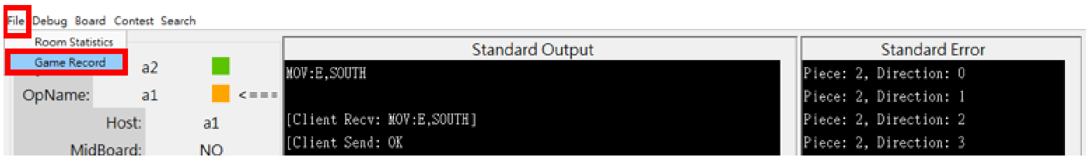
8.7 Game Records

This feature allows players to view detailed content of each game, including moves, program output, and results. Figure 40 shows instructions for accessing game records.

Additional details are explained in Section 7.2.1.



(a) Access Method 1



```
MOV:E,SOUTH

[Client Recv: MOV:E,SOUTH]
[Client Send: OK
```

```
Piece: 2, Direction: 0
Piece: 2, Direction: 1
Piece: 2, Direction: 2
Piece: 2, Direction: 3
```

(b) Access Method 2

Figure 40: Game Records Web Page

> Note: When viewing the board and moving through steps, the scrollbar in the adjacent output window may slightly misalign due to the program's output buffer, but it will generally be near the correct position.

8.8 Remote Board

To accommodate the need to view web pages and run the client on different computers, a Remote Board feature has been added. This allows viewing the search board of another computer through a browser.

Further details are provided in Section 7.3.

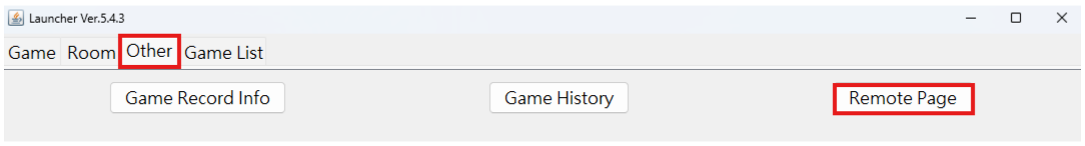


Figure 41: Starting Remote Board

8.9 Contest Mode

The Contest Mode provides an environment for contests, including opponent matching and win/loss recording. Players can compete with others in this mode.

The activation method is detailed in Section 5.4.

Note: Competition Mode is only supported on online servers.

Contest Mode	Online Server	Local Testing
AI vs AI	Yes	No
AI vs Human	Yes	No
Human vs Human	Yes	No

8.10 Disconnection Reconnection

In general and contest modes, this feature allows players to reconnect to a game room after a network disconnection or client closure.

The reconnection method is the same as joining a room or contest.

Note: This feature is only supported on online servers.

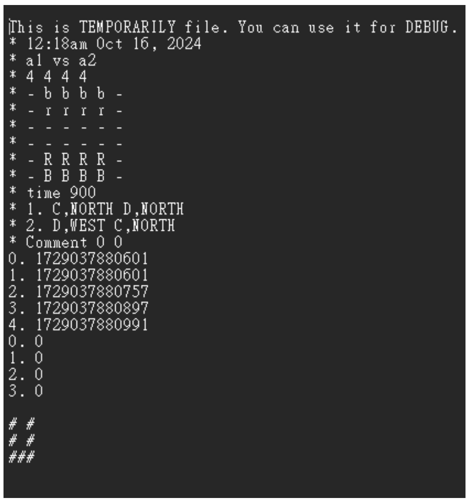
Reconnection	Online Server	Local Testing
AI vs AI	Yes	No
AI vs Human	Yes	No
Human vs Human	Yes	No

8.11 Custom Board

This feature allows users to customize the initial board directly without relying on historical moves to recreate the board.

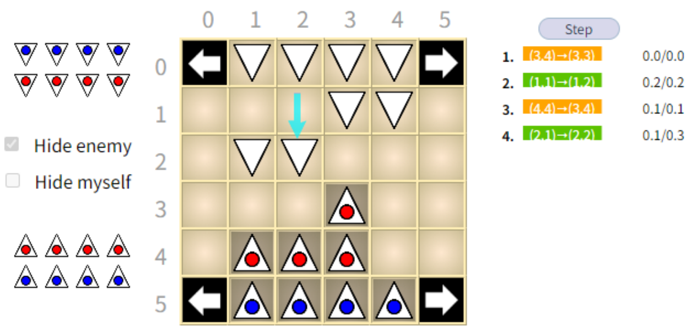
The room host must create a valid mid-game file as defined in Section 8.12, specifying the remaining pieces, initial board layout, and first-move color as needed.

After completing the custom board, the room host must follow the operations in Section 8.4 to load the board.



```
This is TEMPORARILY file. You can use it for DEBUG.
* 12:18am Oct 16, 2024
* a1 vs a2
* 4 4 4 4
* - b b b b -
* - r r r r -
* - - - - - -
* - - - - - -
* - R R R R -
* - B 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) Example Custom Board File



[✓] Hide enemy
Hide myself

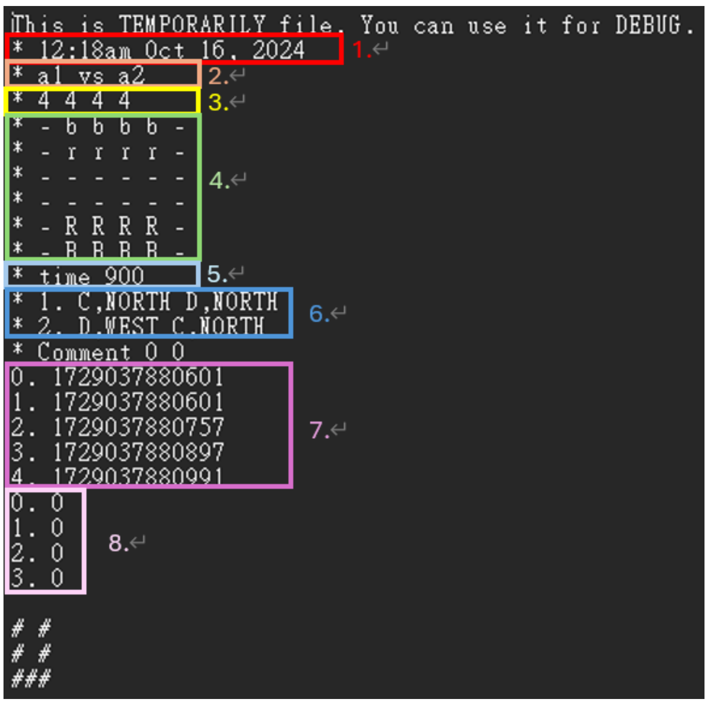
```
1. (3,4)→(3,3)    0.0/0.0
2. (1,1)→(1,2)    0.2/0.2
3. (4,4)→(3,4)    0.1/0.1
4. (2,1)→(2,2)    0.1/0.3
```

(b) Example Custom Board Display

Figure 42: Custom Board Example

8.12 Board File Format Description

Each ply begins with the current game state saved in the same folder as the AI program, allowing the AI to read the board and search for the best move. The format of board.txt is shown in Figure 43, with the meanings of each section as follows:



```
This is TEMPORARILY file. You can use it for DEBUG.
* 12:18am Oct 16, 2024
* a1 vs a2
* 4 4 4 4
* - b b b b -
* - r r r r -
* - - - - - -
* - - - - - -
* - R R R R -
* - B 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

# #
# #
###
```

Figure 43: board.txt Example for Read Mode

1. Game start time
2. Players: The first player is listed first, followed by the second player.
3. Remaining pieces: Records the total remaining number of both players' red/blue pieces. Listed in order: first player red pieces, first player blue pieces, second player red pieces, second player blue pieces.
4. Initial board: Piece codes are defined in Section 9.2.
5. Move time limit: If set to 0, there is no time limit.
6. Move history (from the perspective with the current player side at the bottom):
   - 1. C,NORTH D,NORTH
     In the first round, the first player moves their piece C north, and the second player moves their piece D north.
   - 2. D,WEST C,NORTH
     In the second round, the first player moves their piece D west, and the second player moves their piece C north.
7. Move time log: The server records the timestamp of each move as milliseconds elapsed since 1970/1/1 00:00:00 GMT.
8. Capture log: Records whether a piece was captured in each move and its color:

- 0: No piece captured
- 1: Red piece captured
- 2: Blue piece captured

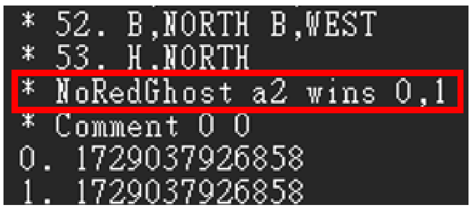
Note: When reading board.txt, both the initial board (highlighted in light green in Figure 43) and the move history (highlighted in light blue in Figure 43) are required to load the latest board state.

Note: The file name must be board.txt. Ensure the board data can be read.

Note: In the light blue box of Figure 43, if a line contains only a single move, there are no extra spaces at the end.

Supplement: If the game ends normally, the board.txt file will include the game result after 6.
The format is:<Ending Type> <Winner/Draw> <Score>

- Ending Type:
  - NoRedGhost: All red ghosts of the winning player were captured.
  - NoBlueGhost: All blue ghosts of the losing player were captured.
  - ChessEscape: The blue ghost of the winning player successfully escaped.
  - GamePlyLimitDraw: The game ended in a draw due to move limit.
- Winner/Draw:
  - <Player>wins: The specified player won.
  - Draw: The game ended in a draw.
- Score:
  - 1,0: First player won.
  - 0,1: Second player won.
  - 0.5,0.5: Draw.



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

Figure 44: Game Result Example

9 Communication Protocols

When connecting the program with the client, a specific communication protocol is required for interaction. This program supports three communication protocols: TCP(Transmission Control Protocol), STDIO (Standard Input Output), and Manual, allowing users to choose the desired communication method.

9.1 Board Position Notation

The board positions across all protocols follow the format shown in Table 1. Follow the Japanese Geister server's coordinate system (vertical-horizontal), where blue ghosts can escape and win from the opponent's corners (0-5, 5-5, or 0-0, 5-0).

(See Section 11 for details).

	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: Board Position Notation

9.2 Piece Codes

At the start of the game, piece codes are assigned as fixed values, with empty positions represented by - .

	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: Piece Code Assignment

9.3 Standard Output

Since the default standard output has a buffer that may delay the client from receiving the search output in real-time, users must ensure output is not held in the buffer when sending replies. The following are reference code examples for C/C++:

1. Flush the buffer immediately after sending the reply.

Listing 1: Flush

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

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

2. Set the buffer size to 0 at the beginning of the program.

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 Command List

Commands in all modes are consistent with the Japanese Geister server (see Section 11) and are used in the following order during the game:

1. Set Red Ghosts

Sender	Command	Meaning
Server	SET?	Ask to set red ghosts
AI	SET:ABCD\r\n	Set ABCD as red ghosts
Server	OK or NG	Response

- Based on Section 9.2, input the piece codes for the four red ghosts.
- You must specify four positions for the red ghosts.
- Once both players finish setup, the server sends the board information to both players and waits for the first player's move.

2. Send Board Information and Request Move

Sender	Command	Meaning
Server	MOV:14R24R34R ... 30u20u10u\r\n	board info, ask for move

- Use Section 9.1 for the positions of all pieces. Specific conditions apply:
  - On the board: Its coordinates
  - Captured: (9,9)
  - Escaped from corners: (8,8)

- Refer to Section 9.2 for describing piece colors using ABCDEFGHabcdefgh:
  - Unrevealed red/blue pieces (on player's side): R / B
  - Revealed red/blue pieces (captured by either player): r / b
  - Unknown color: u

3. Move a Piece

Sender	Command	Meaning
AI	MOV:A,NORTH\r\n	Move piece A north
Server	OK or NG	Response（No piece captured）
Server	OKR or OKB	Response（piece captured）

- Use Section 9.1 to specify the piece to move.
- From the perspective with the current player at the bottom, movement directions can be specified using:
  - NORTH, EAST, WEST, SOUTH
  - or N, E, W, S

4. Game Over

Sender	Command	Meaning
Server	WON:14R24R34R ... 30u20u10u\r\n	board info, the player win
Server	LST:14R24R34R ... 30u20u10u\r\n	board info, the player lose

9.5 Server Response Meaning

After receiving a command, the server replies with the result in the following format:

- General Replies:

Command	Meaning
OK \r\n	Successfully processed
NG \r\n	Processing failed

- Capture Replies:

Command	Meaning
OKR\r\n	Successfully captured a red ghost
OKB\r\n	Successfully captured a blue ghost

Note: OK and NG include spaces, but OKR and OKB do not.

9.6 Flowchart

Figure 45 shows the communication flowchart. The server sends commands continuously until a termination command is received. The AI program only needs to respond to commands without handling game flow logic.

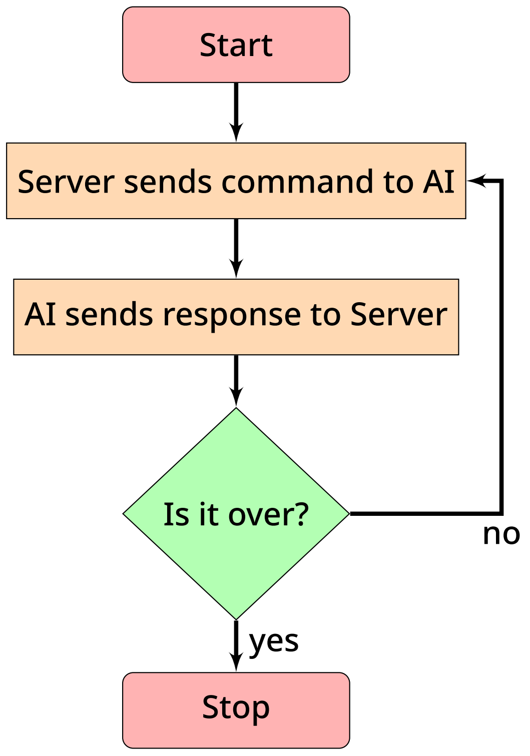


Figure 45: Communication Flowchart

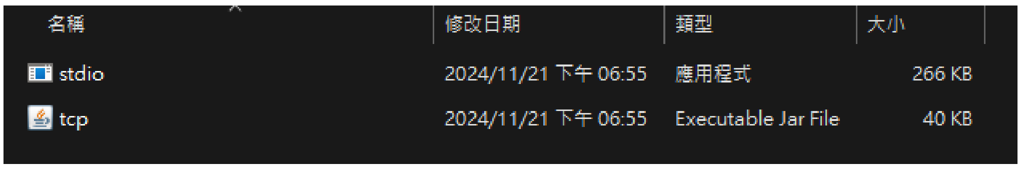
9.7 TCP Mode

In TCP mode, messages are transmitted using Sockets. Users input the corresponding content for the Search program in the AI path and AI arguments fields. When the Client starts the Search program to establish a connection, it automatically appends the IP and port based on the entered AI path and AI arguments. Therefore, users do not need to manually input the IP and port.

Selecting TCP connection mode allows the use of the sample file tcp.jar as shown in Figure 46.

9.7.1 AI Files

The Search folder contains files as shown in Figure 46:



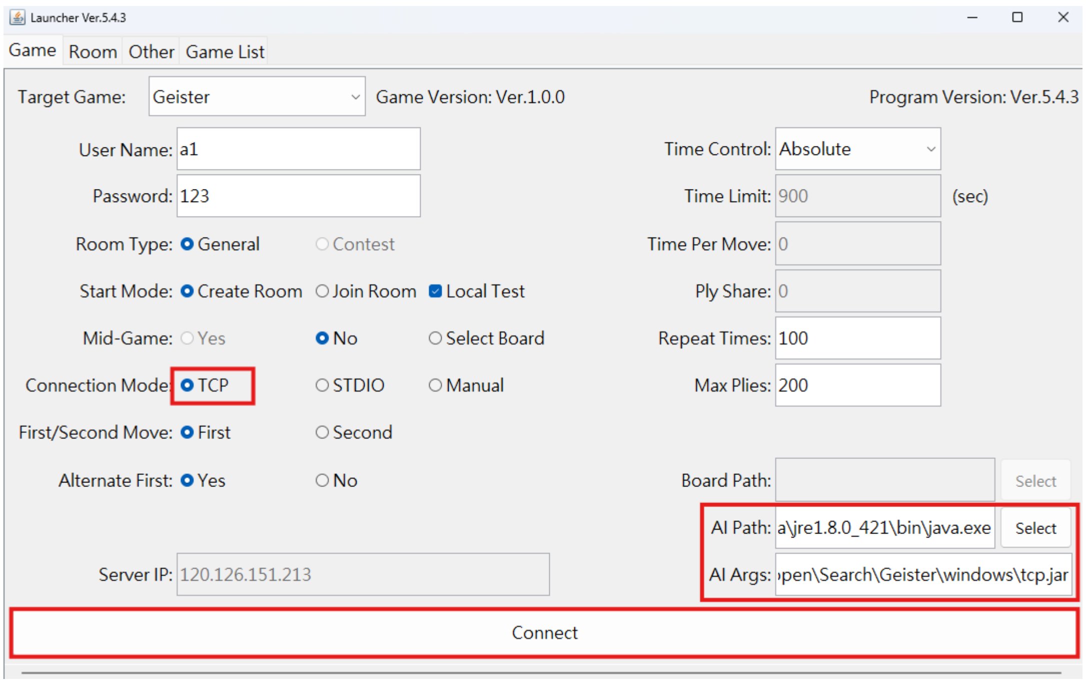
名稱	修改日期	類型	大小
stdio	2024/11/21 下午 06:55	應用程式	266 KB
tcp	2024/11/21 下午 06:55	Executable Jar File	40 KB

Figure 46: Search Folder File List

The AI that uses TCP mode is tcp.jar.

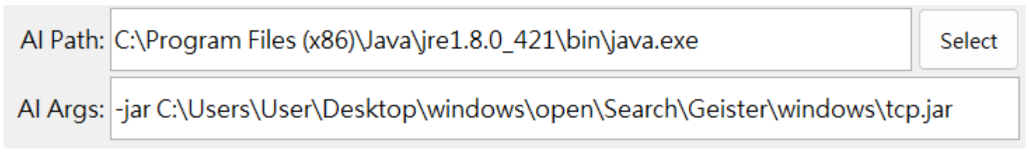
- tcp.jar: RandomPlayer AI from the Japanese Geister server (geister.jar).
  - AI Path: <java.exe path>
  - AI Arguments: -jar <tcp.jar path>

As shown in Figure 47, select the TCP mode and refer to Figure 48 to fill in the path and arguments in the bottom right corner.



Target Game: Geister
Game Version: Ver.1.0.0
Program Version: Ver.5.4.3
User Name: a1
Password: 123
Room Type: [✓ General] Contest
Start Mode: [✓ Create Room] Join Room [✓ Local Test]
Mid-Game: Yes [✓ No] Select Board
Connection Mode: [✓ TCP] STDIO Manual
First/Second Move: [✓ First] Second
Alternate First: [✓ Yes] No
Server IP: 120.126.151.213
Time Control: Absolute
Time Limit: 900 (sec)
Time Per Move: 0
Ply Share: 0
Repeat Times: 100
Max Plies: 200
Board Path:
AI Path: a\jre1.8.0_421\bin\java.exe
AI Args: ›pen\Search\Geister\windows\tcp.jar

Figure 47: Start the TCP AI


AI Path: C:\Program Files (x86)\Java\jre1.8.0_421\bin\java.exe

AI Args: -jar C:\Users\User\Desktop\windows\open\Search\Geister\windows\tcp.jar

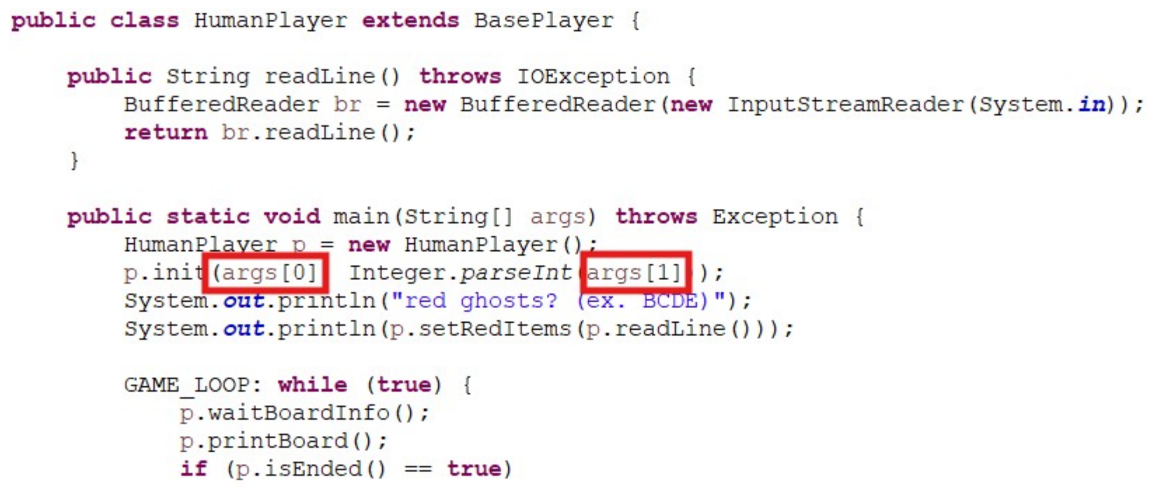
Figure 48: tcp path and args example

the Client program will automatically append the IP and an available port to the end of the command.

The actual command called by the Client for Search is: <AI Path content> <AI Args content> IP port.

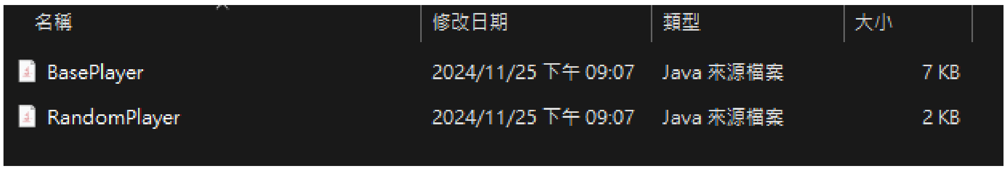
Therefore, the user's AI program needs to read the IP and port from the arguments to establish a connection.

For example, in the sample code, args[0] represents the IP and args[1] represents the 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)
```

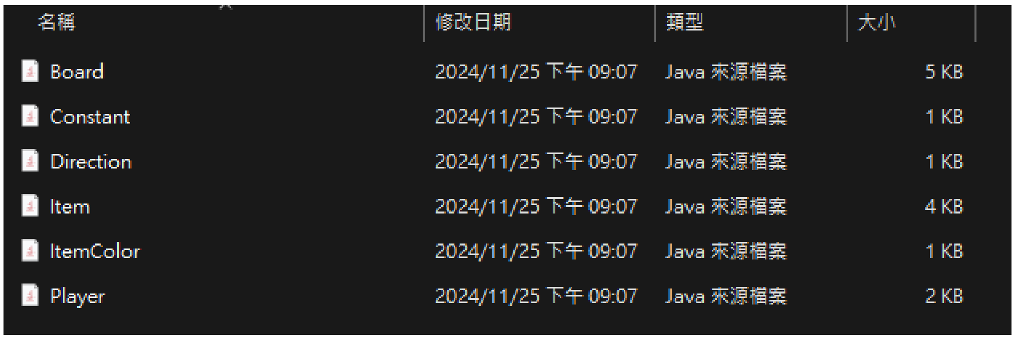
9.7.2 Sample Code

Sample code can be downloaded from 120.126.151.216 (Page Link).
The TCP folder contains files as shown in Figure 49 and 50:



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 Folder



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 Folder

The TCP sample code originates from the Japanese Geister server's GitHub (link in Section 11).

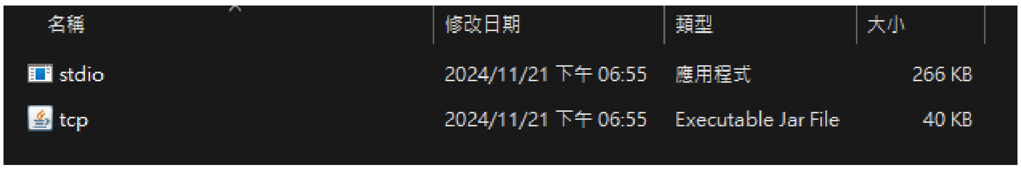
9.8 STDIO Mode

STDIO (Standard Input/Output) represents a protocol where communication occurs through standard input and output streams.

Although the server and AI communicate via the client, the client only handles data transmission without altering the commands between the server and AI. This section introduces the command format and functionality between the server and AI.

9.8.1 AI Files

The Search folder contains files as shown in Figure 51:



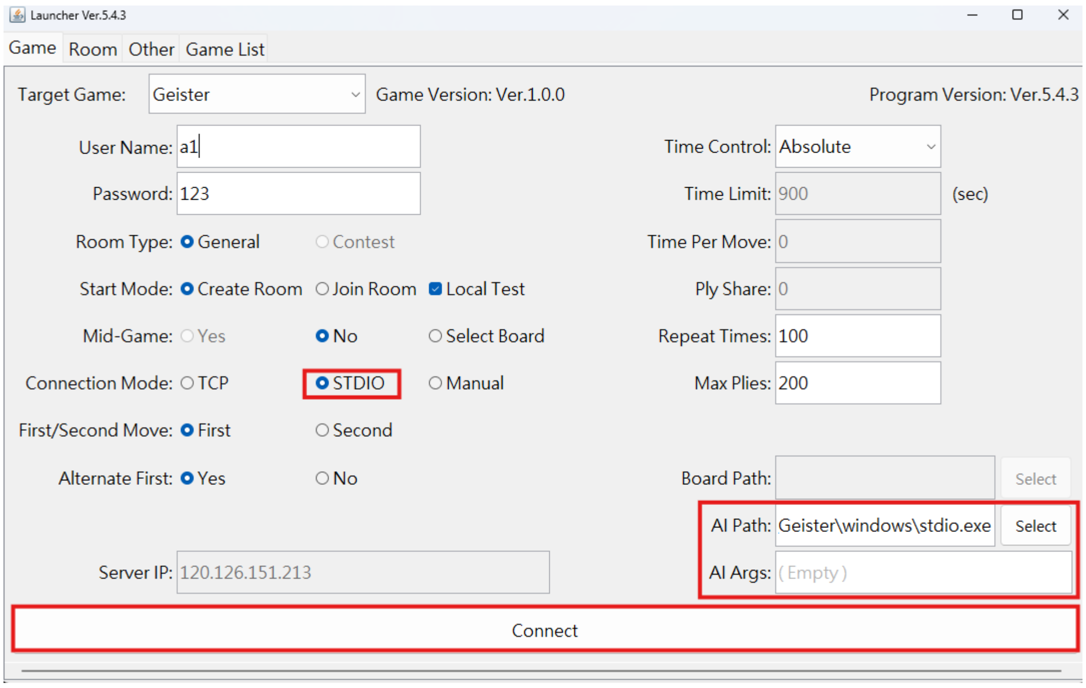
名稱	修改日期	類型	大小
stdio	2024/11/21 下午 06:55	應用程式	266 KB
tcp	2024/11/21 下午 06:55	Executable Jar File	40 KB

Figure 51: Search Folder File List

The AI that uses STDIO mode is stdio.exe.

- stdio.exe: A random-move AI. If it cannot be used directly, refer to Section 9.8.2 for compilation instructions.
  - AI Path: <stdio.exe path>
  - AI Arguments: None

As shown in Figure 52 select the STDIO mode and refer to Figure 53 to fill in the path in the bottom right corner.



Launcher Ver.5.4.3

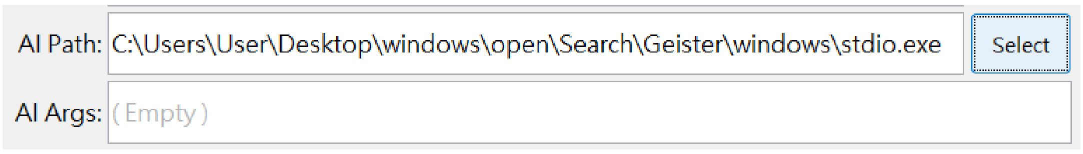
Game Room Other Game List

Target Game: Geister   Game Version: Ver.1.0.0   Program Version: Ver.5.4.3

User Name: a1
Password: 123
Room Type: [✓ General] Contest
Start Mode: [✓ Create Room] Join Room [✓ Local Test]
Mid-Game: Yes [✓ No] Select Board
Connection Mode: TCP [✓ STDIO] Manual
First/Second Move: [✓ First] Second
Alternate First: [✓ Yes] No
Server IP: 120.126.151.213

Time Control: Absolute
Time Limit: 900 (sec)
Time Per Move: 0
Ply Share: 0
Repeat Times: 100
Max Plies: 200
Board Path:
AI Path: Geister\windows\stdio.exe
AI Args: (Empty)

Figure 52: Start the STDIO AI

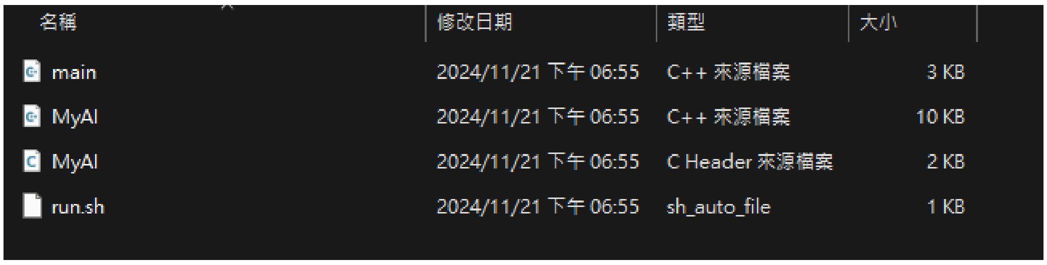

AI Path: C:\Users\User\Desktop\windows\open\Search\Geister\windows\stdio.exe

AI Args: (Empty)

Figure 53: stdio path example

9.8.2 Building AI Files with Sample Code

Sample code can be downloaded from 120.126.151.216 (Page Link).
The STDIO folder contains files as shown in Figure 54:



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 Folder

Sample code includes main.cpp, MyAI.h, MyAI.cpp, and run.sh.

Execute run.sh or open a terminal in the folder and enter:

Windows/macOS

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

This will successfully compile stdio.exe, which can then be placed in the Search folder for the client to locate.

9.8.3 STDIO Sample Code Explanation

1. Receive Messages:

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;
}
```

Read from standard input (stdin) and call different functions based on the command received.

2. Set Red Ghosts:

MyAI.cpp

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

When receiving the SET? command, call the corresponding function to determine the response.

3. Initialize the Board:

MyAI.cpp

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

Call Init_board_state(char* position) to set positions and determine the response.

4. Send Moves:

MyAI.cpp

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

- Call Set_board(char* position) to update the current board positions.
- Call Generate_move(char* move) to generate moves and determine the response.

5. AI Move Generation:

MyAI.cpp

```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);
  ...
}
```

- Call get_legal_move(int* result) to check for legal moves.
- Randomly choose a piece and determine its move direction (can modify this part to implement your AI logic).
- Generate response content.
- Call Make_move(const int piece, const char* direction) to update the board.

6. Exit Game:

MyAI.cpp

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

When receiving the /exit command, output an exit message.

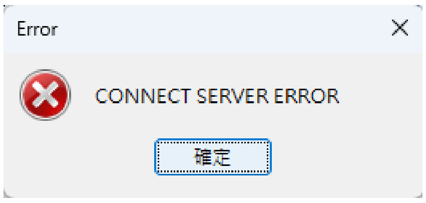
7. Other Functions:

Function	Meaning
Init_board_state(char* position)	Set initial board state
Set_board(char* position)	Update board state
Print_chessboard()	Print the board
get_legal_move(int* result)	Get all legal moves
referee(int piece, int* dst)	Get possible moves for a specific piece
Make_move(const int piece, const char* direction)	Update board with piece and direction

10 Q & A

10.1 Server Connection Error (CONNECT SERVER ERROR)

After pressing the connect button, an error appears as shown in Figure 55, indicating the server cannot be reached.


Error

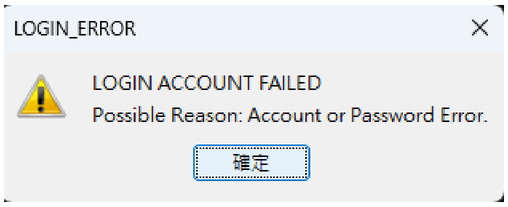
CONNECT SERVER ERROR

Figure 55: Server Connection Error (CONNECT SERVER ERROR)

Solution: Please verify that the Server IP is correct. If the issue persists, contact the relevant personnel.

10.2 Login Error (LOGIN ACCOUNT FAILED)

After pressing the connect button, an error appears as shown in Figure 56, indicating a login failure.


LOGIN_ERROR

LOGIN ACCOUNT FAILED

Possible Reason: Account or Password Error.

Figure 56: Login Error (LOGIN ACCOUNT FAILED)

Solution: Please verify that the account and password are correct. If the issue persists, contact the relevant personnel.

10.3 AI-Related Issues

Issues related to AI, such as failure to execute or standard output problems.

- My AI path is correct, but it fails to start properly.

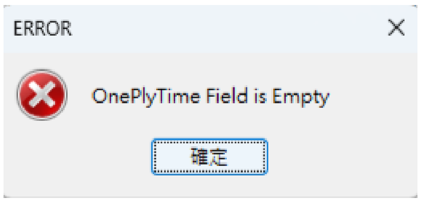
  Solution: Try using an absolute path and confirm the program execution permissions. If the issue persists, contact the relevant personnel.

- Why does the output appear in chunks after a delay instead of displaying in real-time?

  Solution: This is due to output buffering. Ensure you flush the output buffer after writing, or set the output buffer size to zero.

10.4 Input Parameter Errors

After pressing the connect button, an error appears as shown in Figure 57, indicating invalid field input. Various scenarios are listed below:


ERROR

OnePlyTime Field is Empty

Figure 57: Input Parameter Error

- OOO is Empty

  Example: Account field is empty

  Solution: Please ensure the field is filled. If the issue persists, contact the relevant personnel.

- Please select the OOO

  Example: Please select the Start Mode

  Solution Ensure the field is selected. If the issue persists, contact the relevant personnel.

- The Input of OOO is not a number

  Example: The Input of RepeatTime Field is not a number

  Solution: Ensure the field contains numeric characters. If the issue persists, contact the relevant personnel.

- Value of OOO Overflow

  Example: Value of RepeatTime Overflow

Solution: Ensure the value entered is less than 2^{32}. If the issue persists, contact the relevant personnel.

- OOO must larger than zero

  Example: RepeatTime must larger than zero

  Solution: Ensure the value entered is greater than zero. If the issue persists, contact the relevant personnel.

- The Input of OOO is not a Path

  Example: The Input of BoardPath Field is not a Path

  Solution: Ensure the field contains a valid file path for the board file. Use an absolute path if possible. If the issue persists, contact the relevant personnel.

- The Input of BoardPath Field can not Read

  Solution: Ensure the board file has read permissions. If the issue persists, contact the relevant personnel.

- Search Permission denied

  Solution: Ensure the AI program has execution permissions. If the issue persists, contact the relevant personnel.

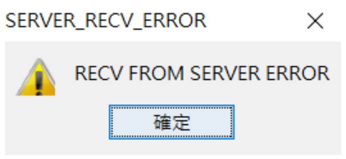
- In order to use HUMAN mode, Please change the browser mode.

  Solution: Verify that the Board Display Mode in the room settings is set to remote or Default Browser. If the issue persists, contact the relevant personnel.

10.5 Server Communication Error

During gameplay, an error appears as shown in Figure 58, indicating issues with sending or receiving data from the server.

- Send Data Error (SEND TO SERVER ERROR)
- Receive Data Error (RECV FROM SERVER ERROR)


SERVER_RECV_ERROR

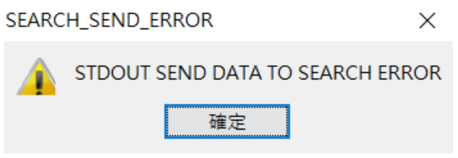
RECV FROM SERVER ERROR

Figure 58: Server Data Communication Error

Solution: Check the network connection and use a stable network. If the issue persists, contact the relevant personnel.

10.6 STDIO Communication Error (STDOUT SEND DATA TO SEARCH ERROR)

At the start of gameplay, an error appears as shown in Figure 59, indicating failure to send data via Stdout to Search.


SEARCH_SEND_ERROR

STDOUT SEND DATA TO SEARCH ERROR

Figure 59: STDIO Communication Error

Solution: Verify the connection mode settings, as this is often caused by an incorrect mode selection. If the issue persists, contact the relevant personnel.

11 References and Contact Information

1. Japan Geister Server GitHub (Page Link)

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.

If there is still anything unclear, please contact:

- 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