がイスタープログラム使用説明書

陳志昌、薛筑軒、吳建德、周子馨、張可欣、楊淳亘、蘇亭伃

January 20, 2025

前書き

本プログラムの主な目的は、ボードゲームなどのプログラム開発者が公開プラットフォーム上で対局し、互いの棋力をテストしたり、異なるバージョン間の棋力の違いを検証したりすることにあります。また、プログラム開発者の負担を軽減するための便利な機能も多数提供しています。

- 基本機能の使い方：Section 2、3、4、5、9をご参照ください。
- 応用機能の使い方：Section 6、7、8をご参照ください。

Contents

1　ゲーム紹介

ガイスター（Geister）は、2 人でプレイするボードゲームで、6x6 の盤面と、8 つの赤いゴースト、8 つの青いゴースト（各プレイヤーに 4 つずつ）を使用します。

1.1　ゲームルール

Figure 1: ガイスターの実物写真

- 準備段階：2 人のプレイヤーは自分の 8 つのゴーストを、手前側の 4x2 マスのエリアに配置します。

  注意：プレイヤー同士は相手のゴーストの配置を知ることはできません。

- ゲーム開始：各ターンごとに、プレイヤーは自分のゴーストを 1 回移動できます。
  - 移動方向：前、後、左、右。
  - 駒を取る：移動先に相手のゴーストがいる場合、そのゴーストを取ります。この際、取ったゴーストの色を確認できます。
- ゲーム終了：次の条件のいずれかを満たした場合、ゲームは終了します。
  - 相手の青いゴーストを 4 つすべて取った場合、自分の勝利。
  - 相手の赤いゴーストを 4 つすべて取った場合、相手の勝利。
  - 自分の青いゴーストが相手の陣地の隅から盤外に出た場合、自分の勝利。

2　ファイル説明

120.126.151.216 (ページリンク) からファイルのダウンロード/更新を推奨します。

2.1　Client フォルダ

Figure 2: Client ファイル構造

- Board：ゲームの盤面記録ファイルとコマンドファイル
- Library：ゲームのライブラリ
- LocalServer：ゲームのローカルサーバー
- Search：AI プログラムファイル
- Setting：ゲーム設定記録ファイル、ルーム設定記録ファイル
- WebServer：ウェブファイル
- Launcher.jar：Client 本体

> 注意：フォルダ構造を任意に変更しないでください。プログラムのパスが誤り、実行できなくなる原因になります。

3　環境設置

環境の構築が完了してから、プログラムを正常に実行することができます。(Ubuntu でテストされたバージョンは 22.04.4 です。)

3.1　Java

本プログラムは Java で作成されているため、実行には JVM が必要です。そのため、事前に jre/jdk をインストールしてください。以下では、Windows と Ubuntu における Java 64 ビット版のインストール方法を紹介します。

> 注意：本プログラムの最低動作環境はJava 8 64-bitです。

3.1.1　Windows

公式サイトから jre (ダウンロードページリンク) をダウンロードしてインストールしてください。

3.1.2　macOS

ターミナルを開いて、以下のコマンドを入力してください (事前に Homebrew をインストールしておく必要があります)。

macOS

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

3.1.3　Ubuntu

ターミナルを開いて、以下のコマンドを入力してください。

Ubuntu

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

3.2　Browser

Google Chrome ブラウザの使用を推奨します。

3.2.1　Windows / macOS

公式サイトから Google Chrome ブラウザをダウンロードしてインストールしてください (ダウンロードページリンク)。

3.2.2　Ubuntu

ターミナルを開いて、以下のコマンドを入力してください。

Ubuntu

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

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

4 起動方法

プログラムのユーザーインターフェース（User Interface）は GUI (Graphical User Interface) を使用しています。以下は使用方法の説明です。なお、Windows と Ubuntu での使用方法は同じです。

Ubuntu では権限の問題があるため、実行権限を付与する必要があります。

Ubuntu / macOS

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

4.1 GUI

フォルダ内のLauncher.jarを直接ダブルクリックするか、ターミナルで以下のコマンドを入力してください：

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 で遭遇しうる問題

- 「開発元が未確認のため開けない」：
  1. 未確認の開発元の Mac アプリを開く方法を参照してください。
  2. すべてのアプリケーションを許可する：

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

- 「Launcher.jar ファイルにアクセスできません（Unable to access jar file）」：
  java にフルディスクアクセス権限を付与してください。
  - Java パス：
    /Library/Java/JavaVirtualMachines/openjdk-???.jdk/
    Contents/Home/bin/java
  - Mac の「プライバシー」設定を変更する方法をご確認ください。
- 「サーバーに接続できない」：
  ファイアウォールが有効になっているか確認してください。必要に応じて、Mac の「ファイアウォール」設定を変更し、例外リストに追加するか、ファイアウォールを無効にしてください。

5　ゲーム設定

本セクションでは、Figure 3 に示された各ゲームパラメーターのフィールド内容とその意味について説明します。

5.1　ゲームオプション設定

Figure 3: GUI クライアント設定ページ

1. ゲーム種類 (Game Type)：このプラットフォームでは複数のゲームをサポートしており、このオプションで実験したいゲームを選択できます。
2. ユーザー名 (Account)：プレイヤーアカウントは管理者に申請してください。ゲストアカウントは a0 から a2000 まで利用可能です。
3. パスワード (Password)：プレイヤーパスワード。ゲストアカウントのパスワードはすべて 123 です。
4. ルームタイプ (Room Type)：プレイヤーは一般モードで対局するか、コンテストモードに参加するかを選択できます。
   - 一般 (General)：プレイヤーが自由にテストプレイできるカジュアルなモード。
   - コンテスト (Contest)：コンテストに参加し、他のプレイヤーと対局してランキングに挑みます。
5. 開始モード (Start Mode)：一般モードで選択できます。プレイヤーはルームを作成するか、他のルームに参加するかを選択できます。

- ルーム作成 (Open)：ルームを作成し、他のユーザーが参加できるようにします。
- ルーム参加 (Enter)：他のユーザーが作成したルームに参加します。
- ローカルテスト (Local Server)：このプログラムをローカルサーバーに接続します（リモートサーバーではありません）。

6. 中盤 (MidBoard)：一般モードで選択できます。ルーム作成者がゲーム開始時の盤面を設定します。
    - はい (Yes)：ゲーム開始時の盤面は、サーバーがその日の中断記録を検索し、最後の盤面から再開します。
    - いいえ (No)：ゲーム開始時の盤面はサーバーから提供される初期盤面です。
    - ボードを選択 (Custom)：ゲーム開始時の盤面は指定されたファイルパスの盤面を使用します。このオプションを使用すると、プレイヤーは中盤ファイルを選択して対局を開始できます。
7. 接続モード (Connect Mode)：クライアントとプログラムの接続プロトコルを指します。詳細は Section 9を参照してください。
    - TCP：TCP プロトコルを使用します。
    - STDIO (Standard Input Output)：標準入力出力を使用します。
    - 手動 (Human)：プログラムを使用せず、プレイヤーが手動で対局を行います。両方のプレイヤーが手動モードを選択した場合、ゲーム中に引き分けを提案できます。いずれか一方が手動モードを選択した場合、ゲーム中に投了が可能です。
8. 先手後手 (Host First Move)：ルーム作成者が第 1 局で先手を取るか後手を取るかを設定します。
9. 先後交代 (Change FIrst Move)：各局終了後に先手と後手を交代するかどうかを選択します。
10. サーバー IP (Server IP)：接続するサーバーの IP アドレス(デフォルト：120.126.151.213)。
11. 計時モード (Timer Mode)：よく使われる計時モードを統合し、時間制限、着手ごとの時間、手数共有設定を簡単に行えるようにします。
    - None：時間制限を設定せず、ゲーム時間を無制限にします。
    - Custom：プレイヤーが自由に秒数制限、1 手制限時間、複数手共有設定を行います。
12. 時間制限 (Time Limit)：1 ゲーム内のプレイヤーの思考時間（つまり自由時間）。何手指すかの制限はありません。
13. 着手ごとの時間 (One Ply Time)：プレイヤーの思考時間を使い切った後に追加される思考時間。
14. 手数共有 (Ply Share)：制限時間内に指さなければならない手数を設定します。
15. 繰り返し回数 (Repeat Times)：このルームで行われる総対局数を設定します。
16. 最大手数 (Max Move Number)：1 ゲーム内での最大手数を設定し、超えた場合は引き分けとなります。

17. 局面パス (Board Path)：中盤設定で「ボードを選択」を選んだ場合、クライアントはこのパスから中盤ファイルを読み取り、サーバーに初期盤面として送信します。

> 推奨：絶対パスを使用すること。

18. AI パス (AI Path)：クライアントはこのパスからAI 対局プログラムを起動し、選択された接続モードのプロトコルを使用して接続します。

> 推奨：絶対パスを使用すること。

19. AI 引数 (AI Argument)：クライアントはこの引数を使用してAI 対局プログラムを起動します。

> この欄を使用して AI に引数を設定するか、または conf ファイルや ini ファイルを使用して AI が引数を読み取れるよう設定してください。

5.2 ルーム作成

ルームを作成するプレイヤーは、以下のゲームパラメーターを設定する必要があります：ルームタイプ、開始モード、中盤、接続モード、先手後手、先後交代、サーバー IP、計時モード（時間制限、着手ごとの時間、手数共有）、繰り返し回数、最大手数、局面パス、AI パス。

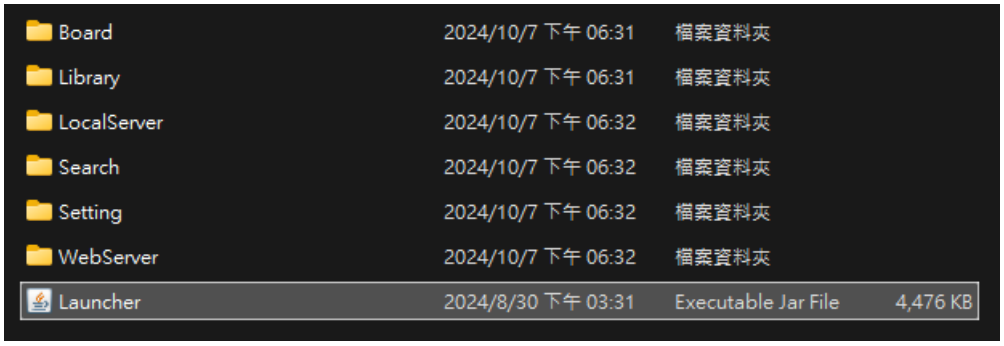


Figure 4: ルーム作成用フォルダ

open フォルダに入り、Figure 4 のように Launcher.jar を直接実行します。

5.2.1 GUI

Figure 3 の開始モードでルーム作成を選択し、ゲームパラメーターとプレイヤー情報を設定した後、接続ボタンを押してサーバーに接続します。

5.3 ルーム参加

参加モードでは、ルームタイプ、接続モード、サーバー IP、AI パスを設定する必要があります。それ以外のゲームパラメーターはルーム作成者が設定します。

enter フォルダに入り、Figure 4 のように Launcher.jar を再び実行します。

5.3.1 GUI

Figure 3 の開始モードでルーム参加を選択し、プレイヤー情報を設定した後、接続ボタンを押してサーバーに接続します。

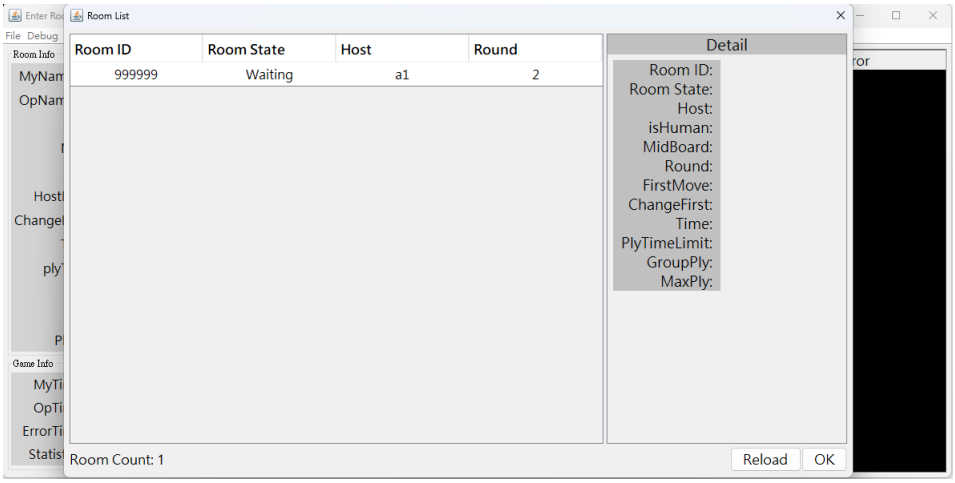


Figure 5: GUI Client ルーム参加メニュー

サーバーに接続成功後、参加可能なルーム一覧が Figure 5 のように表示されます。参加したいルームを選択して確認を押すと、ルームへの参加が完了します。

> 注意：ルーム参加者は、作成者と同じユーザー名を使用できません。

5.4 コンテストに参加

コンテストモードでは、ルームタイプ、接続モード、サーバー IP、AI パスを設定する必要があります。ゲームパラメーターは主催者が一律で設定されるため、プレイヤーは自身の情報を設定するだけで済みます。

Figure 4 のように Launcher.jar を実行してください（起動方法は Section 4 を参照）。

5.4.1 GUI

Figure 6 のルームタイプでコンテストを選択し、プレイヤー情報を設定した後、接続ボタンを押してサーバーに接続します。

Figure 6: GUI Client コンテスト設定ページ

サーバーに接続後、Figure 7 のようなコンテスト一覧が表示されます。参加したいコンテストを選択して進みます。

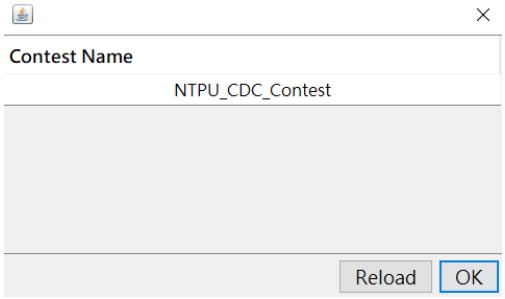


Figure 7: GUI Client コンテストリストページ

コンテストへの参加が成功すると、対戦相手がルームに入るのを待つ画面が表示されます（Figure 8 参照）。対戦相手がすでにコンテストに参加している場合、この画面はスキップされます。

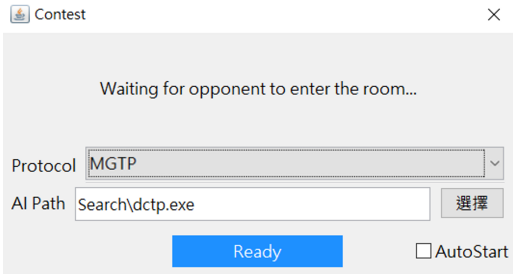


Figure 8: GUI Client 対戦相手参加の待機ページ

注意：プログラムを勝手に終了しないでください。対戦相手が入室するまでお待ちください。問題が発生した場合は、関係者にお問い合わせください。

コンテストに参加後、Figure 9のような確認画面でパラメーター設定内容を確認してください。右下の「AutoStart」機能を使用すれば、自動確認後に次回の試合で 10 秒カウント後に自動的に確認されます。

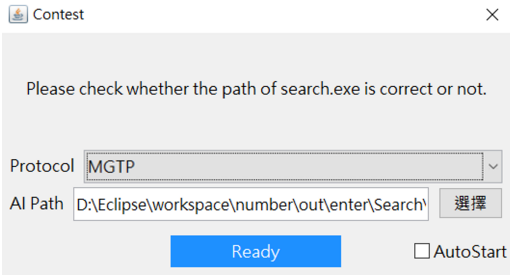


Figure 9: GUI Client コンテスト設定確認ページ

注意：この設定確認は前述の手順内で事前に行うことができます。この画面が表示されるのを待たずに確認を済ませてください。

パラメータを確認した後、対戦相手がまだ確認を終えていない場合、Figure 10 に示すように対戦相手を待機する画面が表示されます。対戦相手が確認を終えた場合は、5 秒のカウントダウン後にコンテストが開始されます。

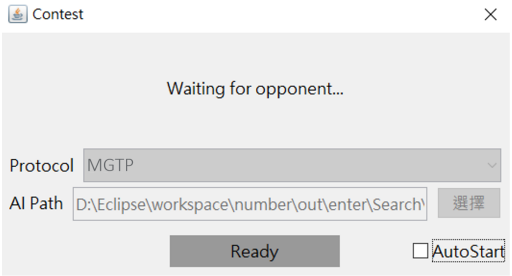


Figure 10: GUI Client 対戦相手の設定確認待機ページ

注意：プログラムを勝手に終了しないでください。対戦相手の確認が終わるまでお待ちください。問題が発生した場合は、関係者にお問い合わせください。

全参加者がプログラムの確認を完了した後は、5 秒のカウントダウンの後に試合が開始されます（Figure 12 参照）。

Figure 11 に示すような画面が表示された場合、その試合では自分が不戦勝（輪空）となります。

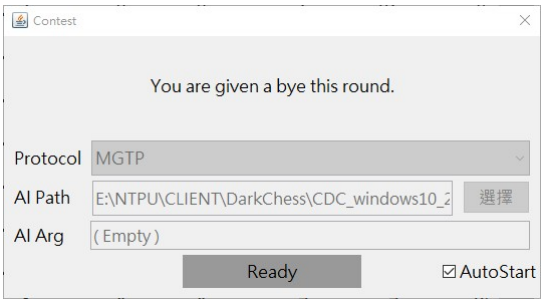


Figure 11: GUI Client 不戦勝の場合

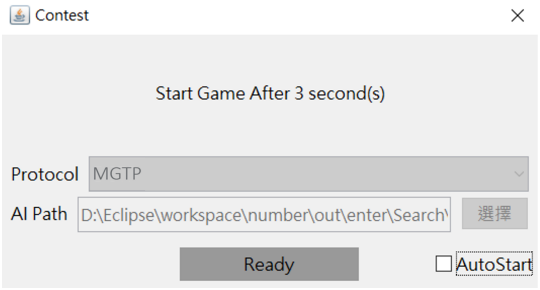


Figure 12: GUI Client 試合開始前のカウントダウン

6　ルーム紹介

6.1　ルームオプション設定

Figure 13: ルーム設定ページ

1. 局面表示モード：盤面ファイルの表示方法を設定。
    - デフォルトブラウザ: Client がデフォルトのウェブブラウザを起動して盤面を表示。
    - リモート: リモート接続の Client が盤面を表示。
    - 非表示: 盤面を表示しない。
2. 出力ウィンドウスクロール：出力ウィンドウのスクロール動作を設定。
    - 自動スクロール: 内容が更新されるたびに、ウィンドウが自動的に最下部までスクロール。
    - 何もしない: 内容が更新されてもスクロールしない。
3. 出力ウィンドウ背景色：出力ウィンドウの背景色を変更可能。
4. 出力ウィンドウフォント色：出力ウィンドウの文字色を変更可能。
5. 出力ウィンドウフォントサイズ：出力ウィンドウの文字サイズを変更可能。

6. ページ切断時自動再起動：ウェブが切断された場合、自動で再接続する。
7. ゲーム再開遅延：各ゲーム間の遅延時間を設定。
8. タイム自動カウントダウン：残り時間のカウントダウンを設定。
    - 有効：自動でプレイヤーの残り時間をカウントダウン。
    - 無効：手ごとに Server から残り時間を取得して更新する。
9. 最終ゲーム出力と盤面をクリア：ルームで設定された試合がすべて終了した後、出力ウィンドウと盤面表示をリセットするかどうかを設定。
10. ゲーム終了後ページを自動閉じる：ルームで設定された試合がすべて終了した後、ウェブページを閉じるかどうかを設定。
11. コンテストモード自動開始：プログラムが 10 秒のカウントダウン後、自動で試合を開始。プレイヤーの確認は不要。
12. 盤面記録ファイル出力：Board フォルダにゲームの記録ファイル（.txt）を出力する。
13. コマンドログファイル出力：Board フォルダにゲームコマンドログファイル（.command）を出力する。
14. ルーム統計ファイル出力：Board フォルダにルーム統計の記録ファイル（.info）を出力する。

6.2 ルームページ

6.2.1 ゲームパラメータ

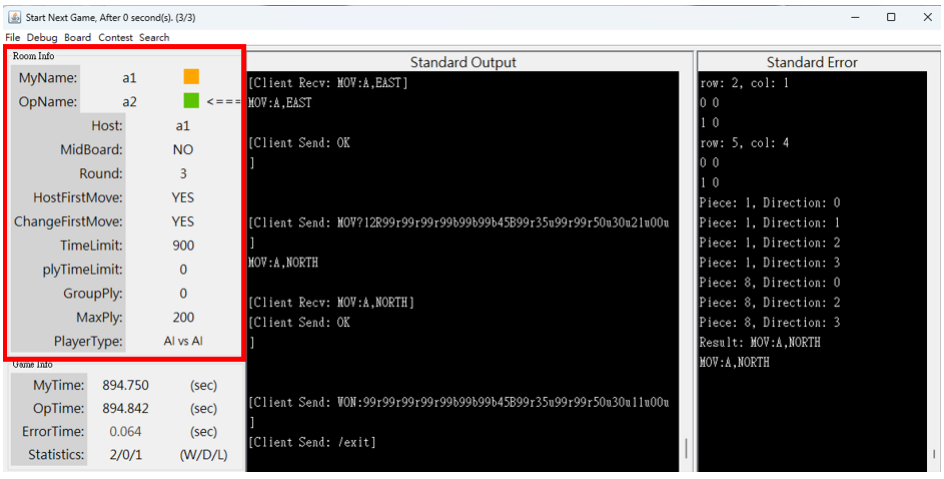


Figure 14: ゲームパラメータ表示エリア

Figure 14 の赤い枠内では、上部に自分と相手のアカウント名、シンボルカラー、矢印で指示された手番が表示されます。下部にはルーム内での各種ゲームパラメータが表示されます。

6.2.2　ゲーム状況

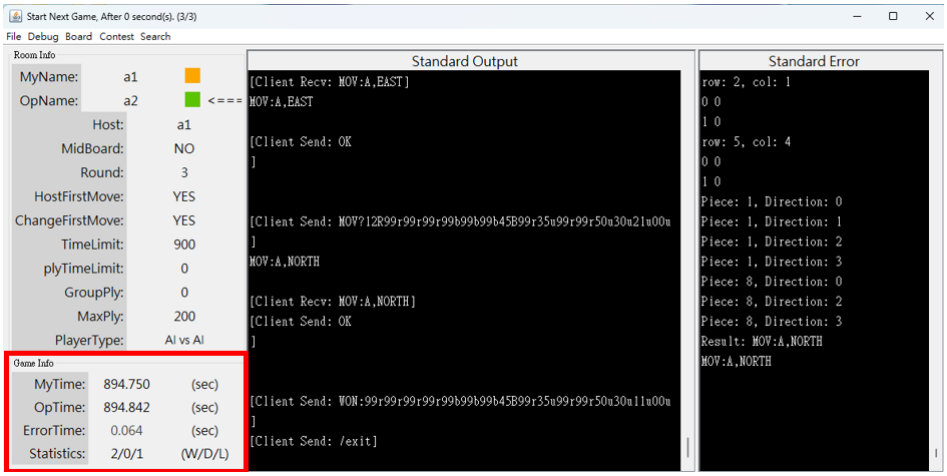


Figure 15: ゲーム状況表示エリア

Figure 15 の赤い枠内には、両者の残り時間、通信遅延時間、勝敗および引き分けの統計が表示されます。

- MyTime：自分の残り時間。
- OpTime：相手の残り時間。
- ErrorTime：通信遅延時間。X と Y をそれぞれ Server と Client が記録した AI のその手に費やした時間とします。計算式は次の通りです：

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

  リアルタイムの通信遅延ではありませんが、ネットワークの安定性の参考になります。

- Statistics：勝利/引き分け/敗北のゲーム数。

6.2.3　出力ウィンドウ

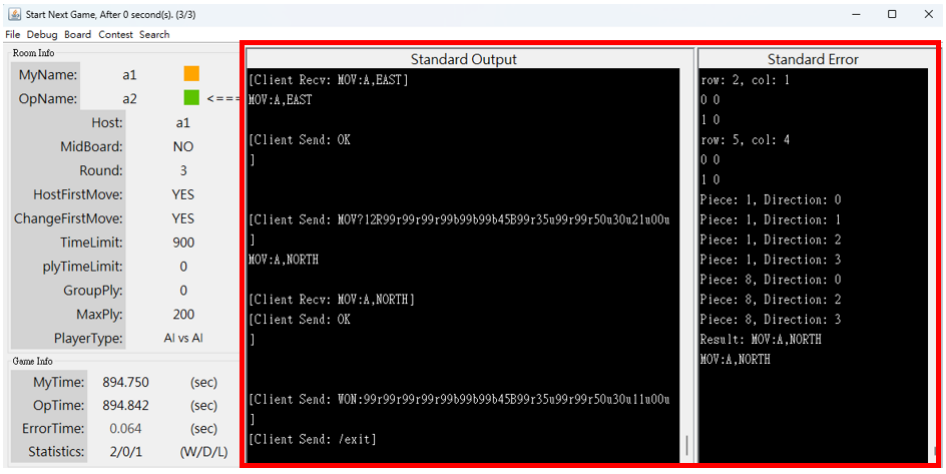


Figure 16: プログラム出力エリア

Figure 16 の赤い枠内では、左側に標準出力 (stdout)、右側に標準エラー出力 (stderr) が表示されます。

> 注意：出力がリアルタイムではなく、一定間隔でまとめて表示される場合は、QA 10.3 を参照してください。

6.2.4 機能メニュー

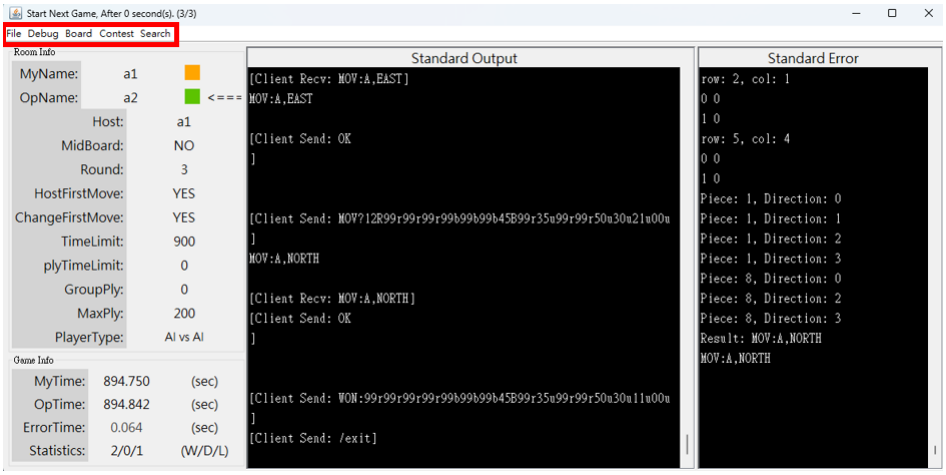


Figure 17: 機能メニューエリア

Figure 17 の赤い枠内には以下の機能メニューがあります。

- File
  - Room Statistics：ルーム内の全試合の統計情報を表示（Section 8.6 を参照）。
  - Game Record：単一試合の内容を表示（Section 8.7 を参照）。
- Debug
  - AutoScroll：内容更新時にスクロールを自動で最下部に移動するかを設定。
  - Increase Font Size：Debug エリアの文字サイズを大きくする。
  - Decrease Font Size：Debug エリアの文字サイズを小さくする。
  - Pop Debug Window：Debug エリアを独立したウィンドウとして表示し、デバッグを容易にする。
- Board
  - Open Board：盤面ウェブページを開く。
- Contest
  - AutoStart：コンテストモードで自動的に準備して試合を開始。
- Search
  - About：現在の Search に関する情報を表示。

7　盤面紹介

現在の盤面は互換性を高めるため、ウェブページ形式で表示されています。局面表示モードでデフォルトブラウザを使用する設定の場合、Client が自動的に盤面ファイルを開き、接続してくれます。以下は、盤面上の各エリアと機能の紹介です。

> 推奨ブラウザ：Google Chrome。

7.1　ゲーム盤面の紹介

盤面は以下のエリアで構成されています：駒の状態、盤面、履歴、結果（Figure 18 参照）。

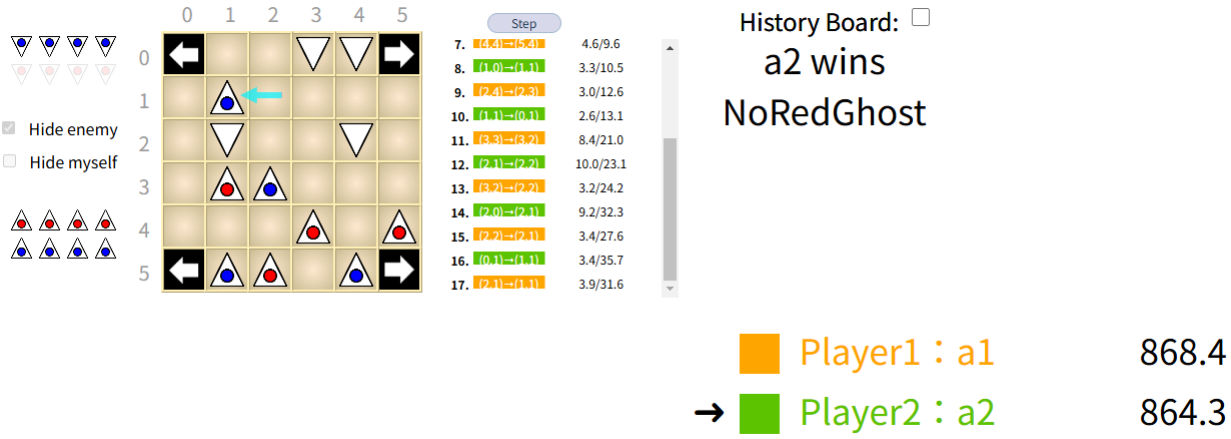


Figure 18: 表示ページ

7.1.1　駒の状態

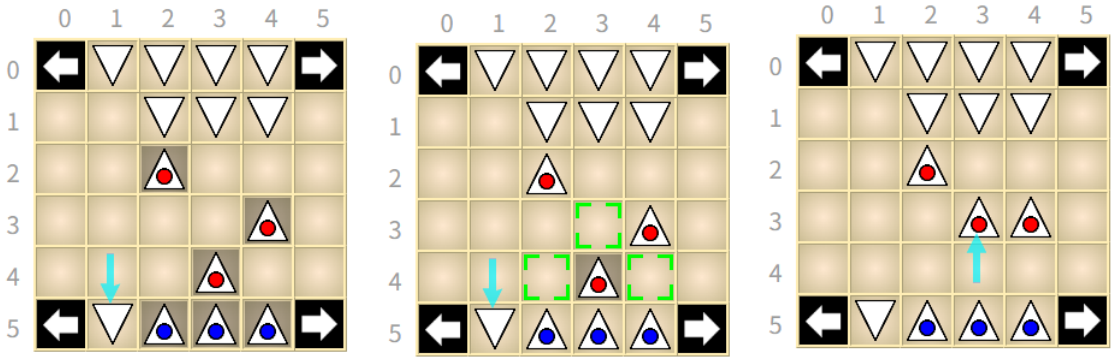
- ウェブページの左側には現在の両プレイヤーの駒の状態が表示されます（Figure 19 参照）。

(a) 生存駒　　(b) 死亡駒

Figure 19: 駒の状態

7.1.2　盤面の表示

- 自分のターンの時、動かせる駒の底部が灰色で表示されます（Figure 20(a) 参照）。
- 灰色の駒を選択すると、その駒の行ける位置が薄緑色のマスで表示され、クリックすることで駒を移動できます（Figure 20(b) および Figure 20(c) 参照）。



(a) ターン開始　(b) 駒を選択　(c) 移動

Figure 20: 盤面の移動表示

7.1.3　手動モード

- 手動モードでは、ゲーム開始前にプレイヤーが自分の盤面配置を設定する必要があり、画面は Figure 21 のように表示されます。
- 設定画面では、プレイヤーが自分の駒をクリックして色を交換し、盤面を調整できます。
- 設定が完了したら、「盤面設定完了」(Set Finish) ボタンをクリックして確認します。

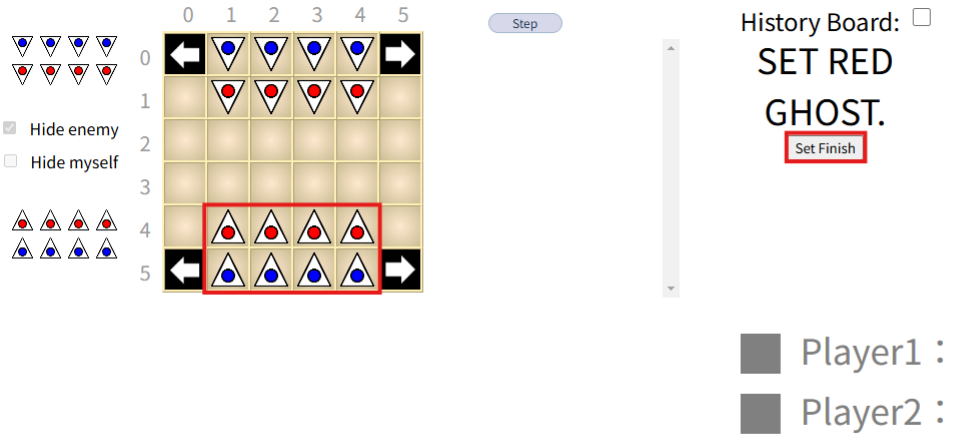


Figure 21: 盤面設定画面

- 対戦相手が設定を完了していない場合、画面は Figure 22 のようになり、待機状態となります。
- 両方が準備を完了すると、ゲームが自動的に開始されます。

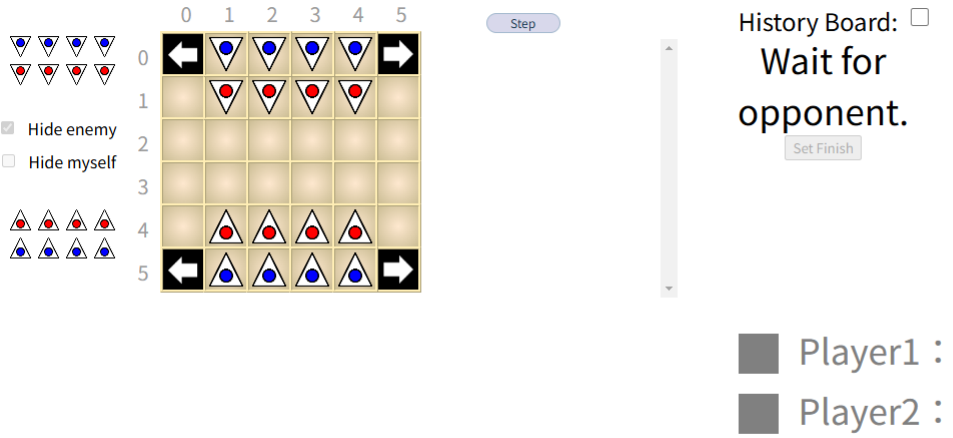


Figure 22: 対戦相手の盤面設定待機状態

- 自分のターン中は、引き分け提案、降参（投了）、やり直しを行うことができます（Figure 23(a)(b) 参照）。
- 手動モードでは、結果エリアに “Your turn” が表示され、自分のターンであることを示します。文字の色は自分の代表色を表します。ガイスターの場合、先手の “Your turn” はオレンジ色、後手は緑色で表示されます（Figure 24(a)(b) 参照）。

(a) 引き分け提案　(b) 降参

Figure 23: 自分のターン時の選択肢

(a) 先手　(b) 後手

Figure 24: 手動モードのシンボルカラー

7.2　ゲーム記録

Section 8.7 の手順に従い、ゲーム記録にアクセスします。

7.2.1　使用説明

- ページを開いた後、履歴ファイルを選択します（Figure 25参照）。
- このページでキーボードの ‘w’, ‘s’, ‘a’, ‘d’ を押すことで、一手ずつ進行または戻ることができます。
- 開いたファイルには、先手と後手の名前および色が表示されます（Figure 26 参照）。履歴盤面は Figure 27 に示されています。
- 別の board ファイルを開くことも可能で、テキストボックスに保存する中盤ファイル名を入力します。上部のテキストボックスにはユーザーの標準出力が、下部のテキストボックスには標準エラー出力が表示され、手順に応じて改行されます。デフォルトでは自動で改行されますが、不要な場合は手動でオフにできます（Figure 28 参照）。

Figure 25: 表示ページ

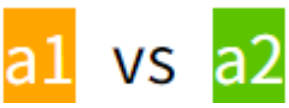


Figure 26: 先手 A1 と後手 A2

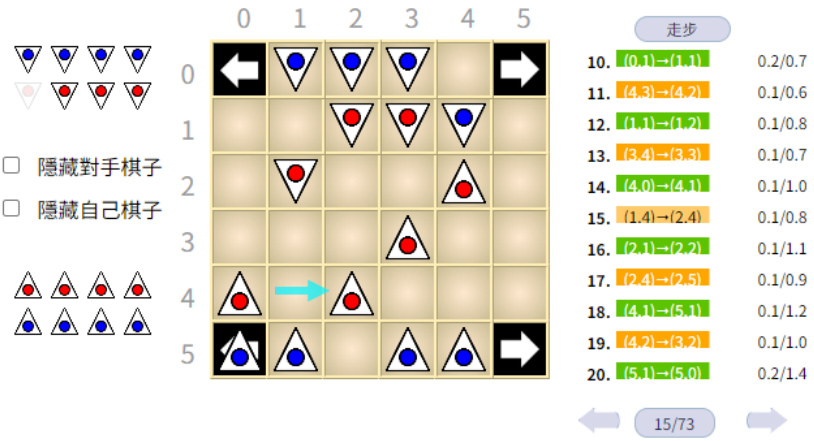


Figure 27: 盤面と履歴の状態

Figure 28: ユーザーの標準入力と標準エラー出力

7.2.2　中盤ファイルの作成

- Section 8.4) に必要な対戦用の中盤ファイルを作成します。ファイルには開始盤面から選択盤面までの全手順が記録されます。作成したファイルは新しいゲームのカスタム盤面として使用できます（Figure 30 参照）。
- 使用方法は以下の通りです（Figure 29 参照）。
  1. 指し手を選択：指し手を直接クリックして中盤の最終手を選択します。
  2. ファイル名を入力：中盤ファイルの名前を入力します。
  3. ボタンを押す：save ボタンをクリックしてファイルをダウンロードします。

Figure 29: 中盤ファイル作成手順

Figure 30: カスタム盤面で新しいゲームを開始

7.3 リモートページ

- Client の IP が 120.126.151.3 の場合、ルームを作成または参加した後、Figure 31 の設定に従うと、Client 側に Figure 32 のインターフェースが表示されます。赤い枠には Client 側の Port が表示され、Port は 50618 と仮定します。
- 観戦用のコンピュータで Section 8.8 に従いリモートページを選択すると、Figure 33 のインターフェースが表示されます。Client の IP: 120.126.151.3 と Port: 50618 を入力することで観戦が可能です。

Figure 31: ルームのページ設定

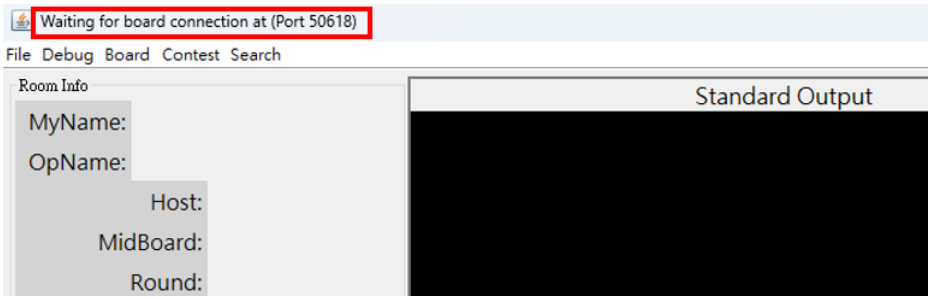


Figure 32: Port 表示ページ

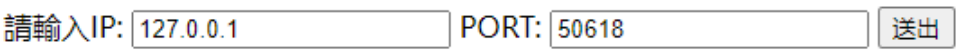


Figure 33: リモートページ

7.4　言語切り替え

ゲーム盤面とゲーム記録ページの左下には言語オプションがあり、Figure 34 および Figure 35 に示されています。ユーザーの好みに応じて中国語、英語、日本語の三言語に変更できます。

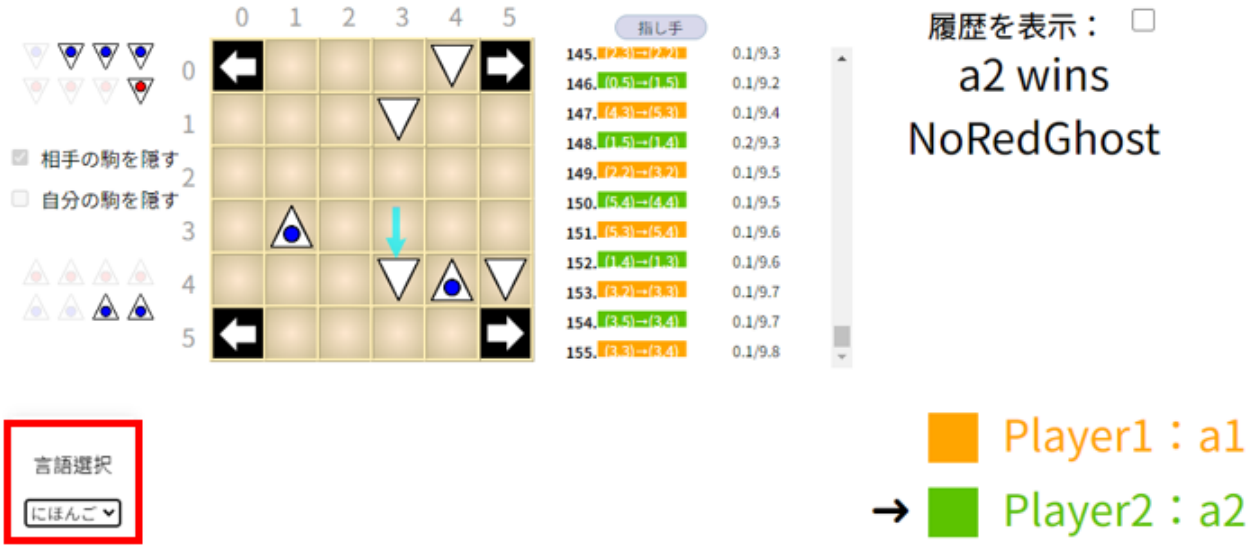


Figure 34: Open、Enter ページの言語オプション

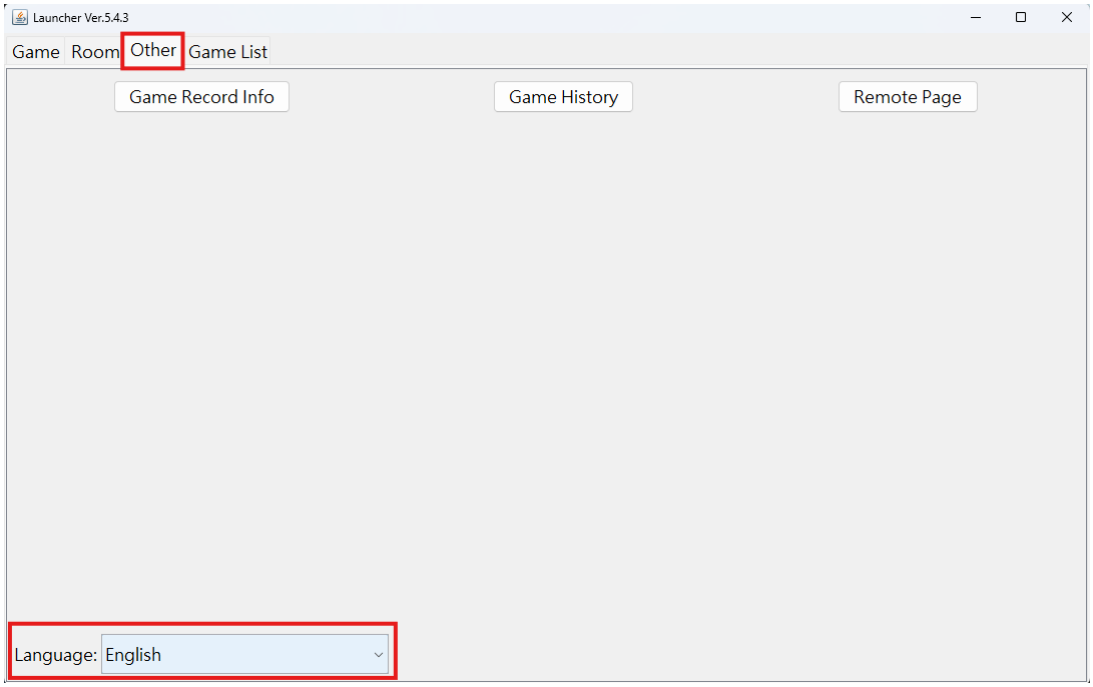


Figure 35: Launcher の言語オプション

8 機能紹介

8.1 降参

ゲームの局勢がほぼ決まっている場合でも、勝敗条件を満たしていないためにゲームを進める必要があり、時間の浪費を招くことがあります。この機能により、プレイヤーまたは AI が事前に降参 (Resign) して試合を終了することができます。方法は以下の 2 種類です：

- 手動 (Section 7.1.3 で定義)
- AI (Section 9 で定義)

注意：降参機能はすべてのモードで使用可能です。

降参	オンラインサーバー	ローカルテスト
AI vs AI	Yes	Yes
AI vs Human	Yes	Yes
Human vs Human	Yes	Yes

8.2 引き分け提案

ゲーム進行中、手番のプレイヤーが引き分け提案 (DrawOffer) を使用して、相手にこの局を引き分けとするかどうか尋ねることができます。相手が同意した場合、この局は引き分けで終了します。反対の場合はゲームが続行されます。また、相手が応答しない場合、提案者は相手の返答を待たずに手を進めることができます。

引き分け提案の操作は Section 7.1.3 で説明されています。

注意：引き分け提案の過程で消費される時間はすべて思考時間に含まれるため、時間制限に注意してください。

注意：公平性の観点から、特定の状況でのみ引き分け提案機能をサポートします。

引き分け	オンラインサーバー	ローカルテスト
AI vs AI	No	No
AI vs Human	No	No
Human vs Human	Yes	No

8.3 中盤再開

接続切断時の中盤再開機能は、ネットワークが不安定な場合に再接続してゲームを続行できる機能を提供します。ただし、この機能はローカルテストでは使用できません。

Figure 36 は中盤再開の設定例です。この設定は、ルームを作成するプレイヤーが中盤で「はい」を選択する必要があります。

Figure 36: 中盤再接続の設定例 (これは Figure 3 のインターフェースです)

8.4 中盤開始

中盤開始は、プレイヤーが中盤からゲームを開始できる機能で、AI プログラムのデバッグに活用できます。

中盤開始の方法については Section 7.2.2 に記載されています。

Figure 37 は中盤開始の設定例です。ルームを作成するプレイヤーは、中盤で「ボードを選択」のオプションを選び、局面パスフィールドで中盤ファイルを選択する必要があります。

Figure 37: 中盤開始設定例

8.5 ローカルテスト

ローカルテストは、インターネットに接続せずに AI プログラムをテストできる環境を提供し、多くのデバッグ機能を備えています。ただし、一部の機能には制限があり、またはローカルテストでのみ使用可能です。詳細な規定は各機能の章に定義されています。

Figure 36 に示すように、両方のプレイヤーが起動モードでローカルテストチェックボックスをオンにするだけで有効になります。

8.6 ゲーム記録情報

ゲーム記録情報は、ルーム内で各試合の結果、勝敗、時間などのパラメータを記録します。開く方法は以下の 2 通りです：

- プログラム外部で開く：Figure 38(a) に示すように、クライアントを開いた後、タブを「Other」（その他）ページに切り替え、「Game Record Info」（ゲーム記録情報）ボタンを押すことでゲーム開始前に統計情報を確認できます。
- ゲーム中に開く：Figure 38(b) に示すように、ルームページで「File」をクリックし、「Room Statistics」を選択することで、ゲーム中に現在または以前のゲーム記録情報をいつでも確認できます。

(a) プログラム外部で開く　　(b) ゲーム中に開く

Figure 38: ゲーム記録情報ページの開き方

Figure 39 において、左側の表には各ゲームのデータがリストされており、右側には双方プレイヤーの勝率および引き分け率の統計グラフが表示されています。下部の「Switch」ボタンで試合数と比率を切り替えることができ、右下の「Reload」ボタンでファイルを再読み込みし、最新のゲーム状況を更新できます。

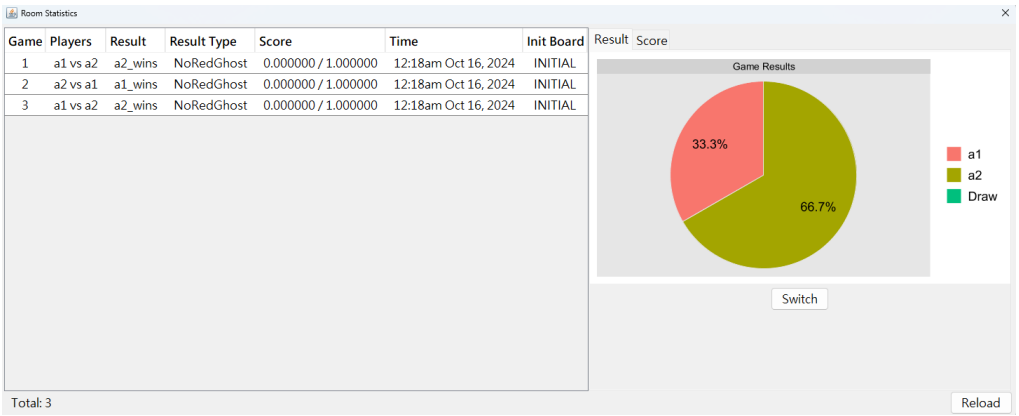


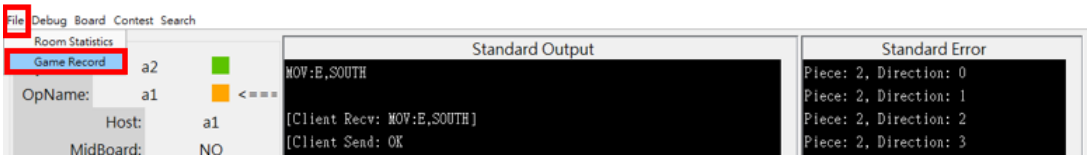
Figure 39: ゲーム記録情報ページ

8.7　ゲーム履歴

ゲーム履歴では、各ゲームの詳細な内容（手順、プログラム出力、結果など）を確認できます。Figure 40 は開き方の説明です。

その他の詳細は Section 7.2 に記載されています。

(a) 開き方 1



(b) 開き方 2

Figure 40: ゲーム履歴ページの開き方

注意：ボードを確認しながら手を進める際、隣の出力ウィンドウのスクロール位置は、プログラムの出力バッファによって記録行数にわずかな差異が生じることがありますが、おおむね正確な位置付近にあります。

8.8　リモートページ

ウェブページと Client 実行を別のコンピュータで行う必要がある場合、リモートで他のコンピュータの Search ボードを閲覧する機能を提供します。

詳細は Section 7.3 に記載されています。

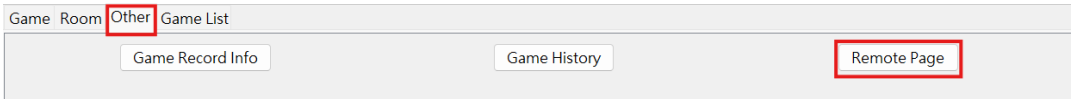


Figure 41: リモートページの開き方

8.9　コンテストモード

コンテストモードは、対戦相手を割り当て、勝敗を記録する競技環境を提供します。このモードで他のプレイヤーと対戦できます。起動方法は Section5.4 で詳述されています。

開き方は Section 5.4 で詳述されています。

注意：コンテストモードはオンラインサーバーのみ対応しています。

コンテストモード	オンラインサーバー	ローカルテスト
AI vs AI	Yes	No
AI vs Human	Yes	No
Human vs Human	Yes	No

8.10 接続再試行

通常モードおよびコンテストモードで、ネットワーク切断やクライアントの終了時に再びゲームルームに接続する機能を提供します。

再接続の方法は、通常のルーム追加やコンテストへの参加方法と同じです。

注意：オンラインサーバーのみ対応しています。

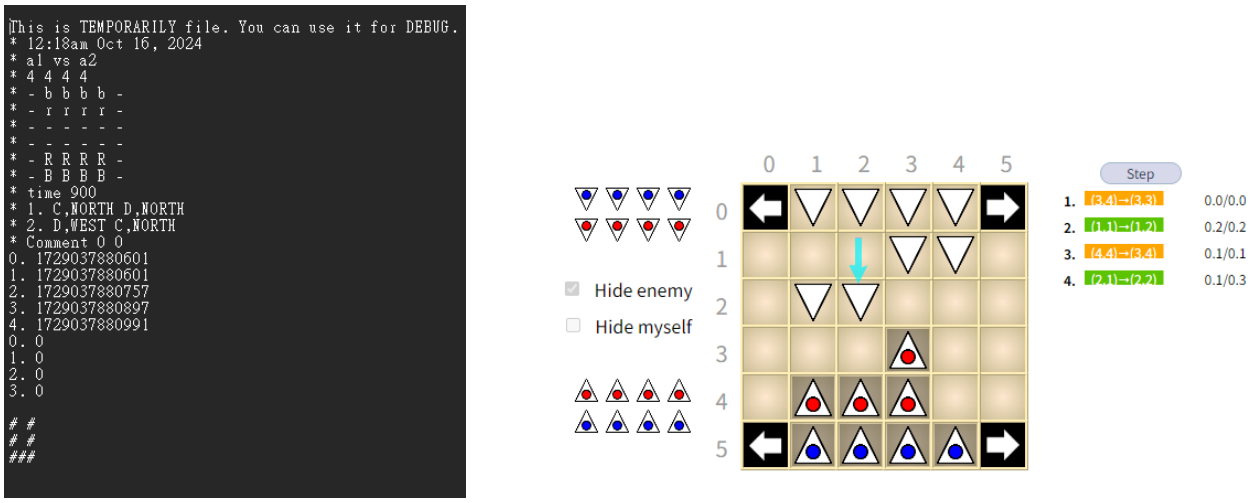
接続再試行	オンラインサーバー	ローカルテスト
AI vs AI	Yes	No
AI vs Human	Yes	No
Human vs Human	Yes	No

8.11　カスタム盤面

ユーザーが直接初期盤面をカスタマイズできる機能を提供します。これにより、履歴の手順を依存せずに盤面を再現できます。

ルーム作成者は、Section 8.12 に従って合法的な中盤ファイルを作成する必要があります。ファイル内の残り駒数、初期ボード、先後手をニーズに応じてカスタマイズされたボードを作成します。

カスタム盤面が完成した後、ルームを作成したプレイヤーは、Section 8.4 の操作に従って盤面を再現する必要があります。



(a) カスタム盤面ファイル　　(b) 例盤面の表示

Figure 42: カスタム初期盤面作成例

8.12 Board ファイルフォーマットの説明

各手（ply）の開始時に、現在の盤面の記録がAI と同じフォルダに保存され、AI プログラムが現在の盤面を読み込み、最適な手を検索できるようになります。board.txt のフォーマットは Figure 43 のようになっており、各ボックスの意味は次のとおりです：

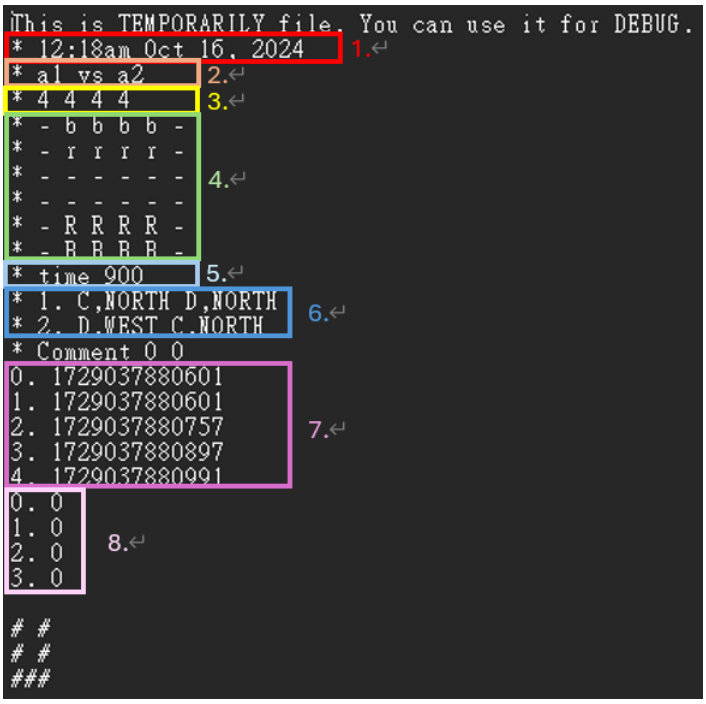


Figure 43: 読み込みモードの board.txt

1. ゲーム開始時間
2. プレイヤー：先手が前、後手が後
3. 残り駒の状態：場に先手後手それぞれの赤/青の駒の数を記録します。順番は、先手の赤駒、先手の青駒、後手の赤駒、後手の青駒です。
4. 初期盤面：駒のコードは Section 9.2 に定義されています。
5. 手の制限時間。0 の場合、時間制限はありません。
6. 履歴の手順 (現在のプレイヤーが下側にいる視点で移動します)：
   - 1. C,NORTH D,NORTH
     最初のターンで、先手は自分の C を 北 に 1 歩、後手は自分の D を 北 に 1 歩動かします。
   - 2. D,WEST C,NORTH
     2 ターン目で、先手は自分の D を 西 に 1 歩、後手は自分の C を 北 に 1 歩動かします。
7. 各手の時間記録：時間計算方法は、サーバーが受け取った手のタイムスタンプに基づきます。タイムスタンプは、1970 年 1 月 1 日 00:00:00 GMT から現在までの経過ミリ秒です。

8. 駒を取った記録：6 の手順に基づき、各手が駒を取ったか、どの色の駒を取ったかを記録します。
    - 0：駒は取られていない
    - 1：赤駒を取った
    - 2：青駒を取った

注意：board.txt を読み込む際、初期盤面(Figure 43 の淡い緑色ボックス) と履歴の手順(Figure 43 の淡い青色ボックス) を読み込む必要があり、最新の盤面を取得するためにはそれらが必要です。

注意：ファイル名はboard.txtである必要があります。盤面データが正しく読み込めることを確認してください。

注意：Figure 43 の淡い青色ボックスで、1 行に 1 つの手しか記録されていない場合、後ろに余分な空白がありません。

補足：ゲームが正常に終了した場合、Board ファイルには 6 の後ろに試合結果が追加されます。
フォーマットは：< 終了方式 >< 勝者/引き分け >< スコア >

- 終了方式
    - NoRedGhost：勝者の赤いゴーストが全部食べられた
    - NoBlueGhost：勝者が敵の青いゴーストを全部食べた
    - ChessEscape：勝者の青いゴーストが逃げた
    - GamePlyLimitDraw：引き分け
- 勝者/引き分け
    - < プレイヤー > wins：そのプレイヤーが勝った
    - Draw：引き分け
- スコア
    - 1,0：先手が勝利
    - 0,1：後手が勝利
    - 0.5,0.5：引き分け

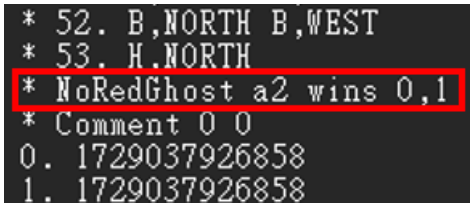


Figure 44: 試合結果例

9　通信プロトコル

プログラムとクライアントの接続には、特定の通信プロトコルが必要です。本プログラムには以下の 3 種類の通信プロトコルがあり、ユーザーは必要に応じて通信方法を選択できます：TCP(Transmission Control Protocol)、STDIO (Standard Input Output) 、手動 (Manual)。

9.1　ボード位置コード

各通信プロトコルのボード位置は Table 1 に示しています。日本の Geister サーバー座標形式（縦-横）に従い、青のゴーストが敵陣のコーナー（0-5、5-5、または 0-0、5-0）から脱出すると勝利となります。
（詳細は Section 11を参照）

	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: ボード位置コード

9.2　駒コード

ゲーム開始時、駒コードは以下のように固定されています。空白位置は「-」で表されます。

	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: 駒コード配置

9.3 標準出力

デフォルトの標準出力にはバッファ（buffer）があるため、クライアントがサーチの出力を即時に受信できない場合があります。このため、ユーザーは出力を返答する際に必ずバッファに出力が残らないようにし、クライアントが適切に受信できることを確認してください。以下は C/C++ での参考コードです：

1. 出力後に強制的にバッファをクリアする。

Listing 1: Flush

```c
// 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. プログラム開始時にバッファサイズを 0 に設定する。

Listing 2: Set buffer size

```c
// 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 コマンドリスト

各モードのコマンドは日本の Geister サーバーと同じです（詳細は Section 11を参照）。ゲーム進行に従って以下の順で使用されます：

1. 赤いゴーストの設定

送信元	コマンド	意味
Server	SET?	赤いゴーストの設定要求
AI	SET:ABCD\r\n	ABCDを赤いゴーストとして設定
Server	OK または NG	結果の返答

- Section 9.2 に従い、赤ゴーストとして設定したい駒コードを入力します。
- 赤いゴーストは合計 4 つであるため、4 つの位置を入力する必要があります。
- 両方が設定完了後、サーバーは双方にボード情報を送信し、先手の手番を待ちます。

2. ボード情報の送信と移動要求

送信元	コマンド	意味
Server	MOV:14R24R34R ... 30u20u10u\r\n	ボード情報、移動要求

- Section 9.1 に基づき、各駒の位置を記述します。特定の状態では座標の規定が異なります。
  - 駒が盤上にいる場合：その座標
  - 駒が取られた場合：(9,9)
  - 駒がコーナーから脱出した場合：(8,8)

- Section 9.2 に従い、ABCDEFGHabcdefgh の順に各駒の色を記述します：
  - 非公開（自陣の盤上）の赤/青駒：R / B
  - 公開済み（双方が取った）の赤/青駒：r / b
  - 色不明：u

3. 駒の移動

送信元	コマンド	意味
AI	MOV:A,NORTH\r\n	Aを北に移動
Server	OK または NG	移動結果（駒を取らなかった場合）
Server	OKR または OKB	移動結果（駒を取った場合）

- Section 9.1 に従い、移動する駒を指定します。
- 現在のプレイヤーが下側にいる視点で移動します。東南西北の方向指定は以下のいずれかを使用できます：
  - NORTH、EAST、WEST、SOUTH
  - N、E、W、S

4. ゲーム終了

送信元	コマンド	意味
Server	WON:14R24R34R ... 30u20u10u\r\n	ボード情報、そのプレイヤーの勝利
Server	LST:14R24R34R ... 30u20u10u\r\n	ボード情報、そのプレイヤーの敗北

9.5　サーバーの返答の意味

サーバーがコマンドを受信すると結果を返答します。フォーマットは以下の通りです：

- 一般的な返答：

コマンド	意味
OK \r\n	正常に受理
NG \r\n	受理失敗

- 移動後の駒を取る返答：

コマンド	意味
OKR\r\n	赤駒を取ることに成功
OKB\r\n	青駒を取ることに成功

> 注意：フォーマット中の OK、NG には空白が含まれますが、OKR、OKB には含まれません。

9.6　フローチャート

Figure 45 は通信のフローチャートです。終了コマンドを受信するまで、すべてのコマンドはサーバーが送信し、AI プログラムは指示に対応する返答をするだけで、ゲーム進行に関する処理は必要ありません。

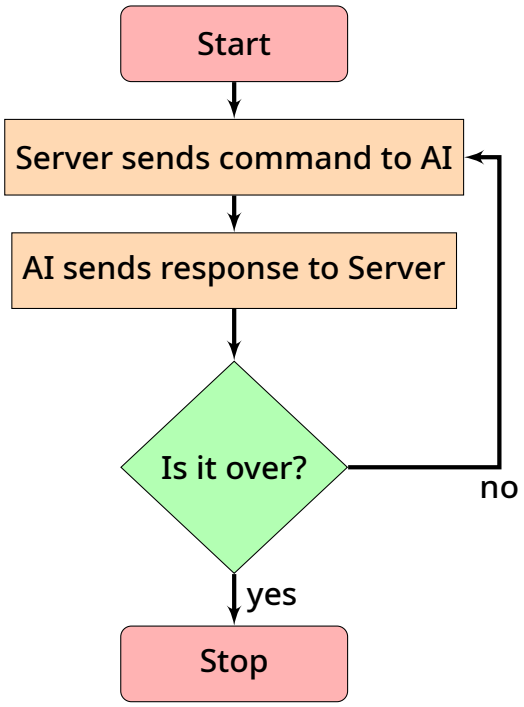


Figure 45: 通信フローチャート

9.7　TCP モードの紹介

TCP モードでは、メッセージの送受信に Socket を使用します。ユーザーは AI パスと AI 引数欄に Search プログラムに対応する内容を入力し、Client が Search プログラムを起動して接続を確立する際、入力された AI パスと AI 引数に自動的に IP とポートを追加して実行します。そのため、ユーザーは IP とポートを自ら入力する必要はありません。

TCPを選択することで、Figure 46 に示されているサンプルの tcp.jar を使用可能です。

9.7.1　AI ファイルの紹介

Search フォルダには以下のようなファイルが含まれます（Figure 46 参照）：

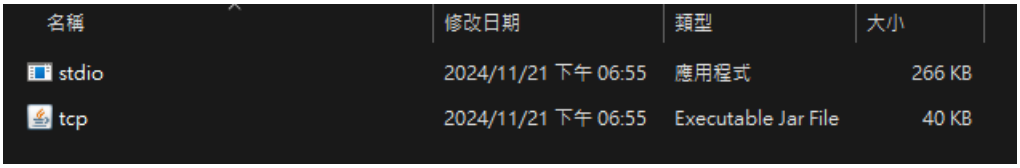


Figure 46: Search フォルダ

TCP モードで使用する AI はtcp.jarです。

- tcp.jar：日本の Geister サーバー geister.jar に含まれる randomPlayer AI。
  - AI パス：< java.exe のパス >
  - AI 引数：-jar < tcp.jar のパス >

Figure 47では TCP モードを有効化する手順が示され、Figure 48 では右下にパスと引数を入力する例が示されています。

Figure 47: TCP AI の起動プロセス

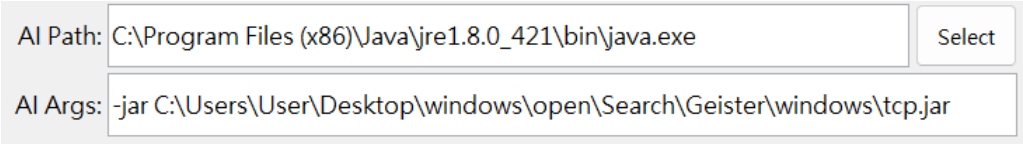


Figure 48: TCP 引数例

ユーザーが入力を完了すると、Client プログラムが自動的に ip と使用可能な port を指令の末尾に追加します。

> Client が実際に Search プログラムを呼び出すコマンド例：< AI path 内容 > < AI Args 内容 > ip port

そのため、ユーザーの AI プログラムは、引数で提供される IP とポートを読み取り、それを基に接続を確立する必要があります。

以下はサンプルコードの例です。args[0] がip、arg[1] がportとして渡されます。

```
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 サンプルコード

120.126.151.216 (リンク) からサンプルコード (Sample code) をダウンロードできます。

TCP フォルダには Figure 49 と Figure 50 のようなファイルが含まれます：

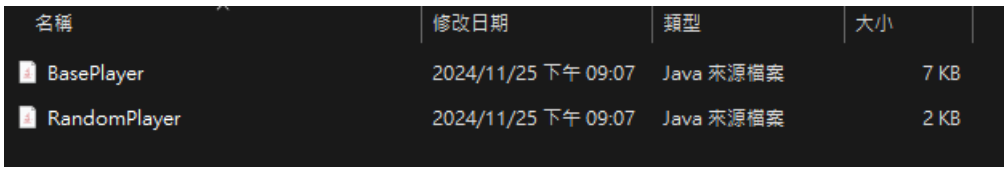


Figure 49: player フォルダ

Figure 50: utils フォルダ

TCP のサンプルコードはすべて日本の Geister サーバーの GitHub から取得されており、その URL は Section 11 に記載されています。

9.8 STDIO モードの紹介

STDIO の正式名称は Standard Input Output で、メッセージの送受信に標準入力出力を利用します。

STDIO プロトコルでは、Server と AI は Client を介して通信を行いますが、Client はデータ転送のみを担当し、Server と AI 間の指令内容を変更しません。そのため、本章では Server と AI 間の指令フォーマットおよび機能を説明します。

9.8.1 AI ファイルの紹介

Search フォルダには以下のようなファイルが含まれます（Figure 51 参照）：

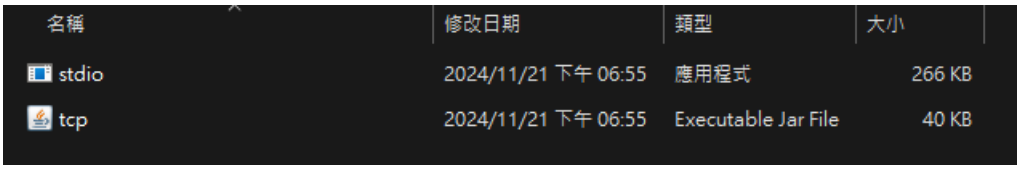


Figure 51: Search フォルダ

stdio モードで使用する AI はstdio.exeです。

- stdio.exe：ランダム移動を行う AI。直接利用できない場合は、Section 9.8.2 を参照し、自身でコンパイルしてください。
  - AI パス：< stdio.exe のパス >
  - AI 引数：なし

Figure 52 のように stdio モードを選択し、Figure 53 の右下隅にパスを入力してください。

Figure 52: stdio AI の起動プロセス

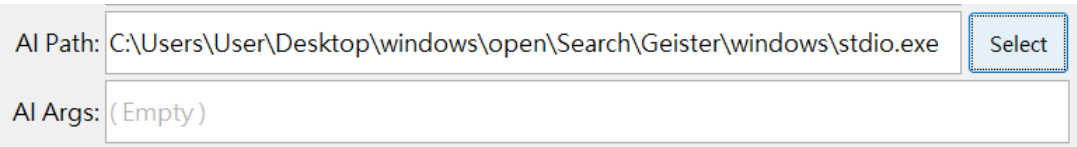


Figure 53: stdio 引数例

9.8.2 サンプルコードを使って AI ファイルを作成

120.126.151.216 (リンク) からサンプルコード (Sample code) をダウンロードできます。

STDIO フォルダには以下のようなファイルが含まれています（Figure 54 参照）：

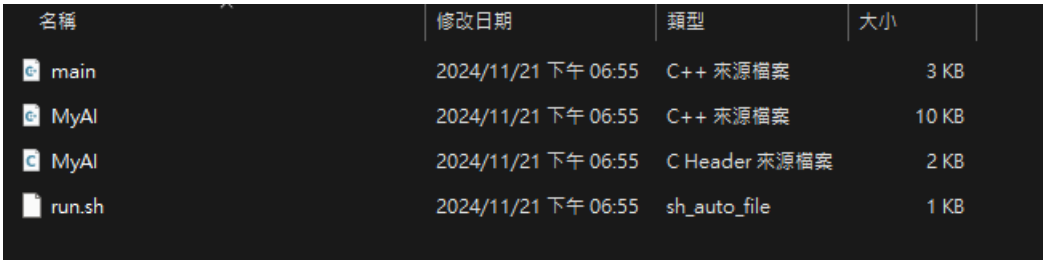


Figure 54: STDIO フォルダ

サンプルプログラムには main.cpp、MyAI.h、MyAI.cpp、run.sh が含まれています。

run.sh を直接実行するか、フォルダ内でターミナルを開いて以下のコマンドを入力します：

Windows/macOS

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

これで stdio.exe のコンパイルが成功します。このファイルを Search フォルダに置き、Client が AI ファイルとして利用できるようにします。

9.8.3 STDIO Sample code コードの説明

1. メッセージの受信：

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;
}
```

標準入力（stdin）を読み取り、受信したコマンドに応じて異なる関数を呼び出します。

2. 赤いゴーストの設定：

MyAI.cpp

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

SET? コマンドを受信した際にこの関数が呼び出され、応答内容を決定します。

3. 盤面の初期化：

MyAI.cpp

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

Init_board_state(char* position) 関数を呼び出して初期位置を設定し、応答内容を決定します。

4. 移動の送信：

MyAI.cpp

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

- Set_board(char* position) を呼び出して現在の盤面位置を更新します。
- Generate_move(char* move) を呼び出して移動を生成し、応答内容を決定します。

5. AI の移動生成：

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);
  ...
}
```

- get_legal_move(int* result) を呼び出し、移動可能なコマを確認します。

- 移動可能なコマの中からランダムに 1 つ選び、そのコマの移動方向を決定します。（この部分の AI ロジックは編集可能です）
- 応答内容を生成します。
- Make_move(const int piece, const char* direction) を呼び出して盤面を更新します。

6. ゲームから退出：

MyAI.cpp

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

/exit コマンドを受信すると、ゲーム退出メッセージを出力します。

7. その他の機能関数：

関数	意味
Init_board_state(char* position)	初期盤面を設定
Set_board(char* position)	盤面の状態を更新
Print_chessboard()	盤面を表示
get_legal_move(int* result)	すべての合法的な移動数を取得
referee(int piece, int* dst)	指定コマの移動可能な方向を取得
Make_move(const int piece, const char* direction)	指定コマと移動方向に基づいて盤面を更新

10　Q & A

10.1　サーバー接続エラー (CONNECT SERVER ERROR)

接続ボタンを押した際、Figure 55 のようなエラーが発生し、サーバーに接続できないと表示されます。

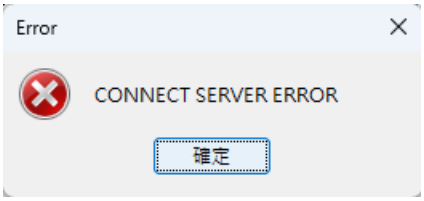


Figure 55: サーバー接続エラー (CONNECT SERVER ERROR)

解決方法：Server IPが正しいか確認してください。それでも解決しない場合は、関連担当者に連絡してください。

10.2　ログインエラー (LOGIN ACCOUNT FAILED)

接続ボタンを押した際、Figure 56 のようなエラーが発生し、ログインエラーと表示されます。

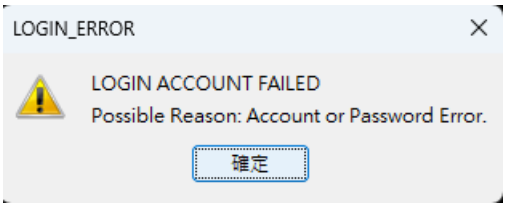


Figure 56: ログインエラー (LOGIN ACCOUNT FAILED)

解決方法：アカウント名とパスワードが正しいか確認してください。それでも解決しない場合は、関連担当者に連絡してください。

10.3　AI 関連の問題

AI 関連するの問題、例えば AI が実行できない、標準出力に関する問題などです。

- AI のパスが正しいにもかかわらず正常に起動しない。

  解決方法：絶対パスを使用して試してください。そして、実行権限も確認してください。それでも解決しない場合は、関連担当者に連絡してください。

- 出力が遅延して一度にまとめて表示される。

  解決方法：標準出力（stdout）のバッファを強制的にクリア(flush)、またはバッファサイズを 0 に設定してください。

10.4　入力パラメータエラー

接続ボタンを押した際、Figure 57のようなエラーが発生し、フィールドの値が不正であると表示されます。

Figure 57: 入力パラメータエラー

- OOO が空 (OOO is Empty)

  例：アカウントフィールドが空 (Account Field is Empty)

  解決方法：該当フィールドが入力されているか確認してください。それでも解決しない場合は、関連担当者に連絡してください。

- OOO を選択してください (Please select the OOO)

  例：起動モードを選択してください (Please select the Start Mode)

  解決方法：該当項目が選択されているか確認してください。それでも解決しない場合は、関連担当者に連絡してください。

- OOO が不正な値 (The Input of OOO is not a number)

  例：繰り返し回数が不正な値 (The Input of RepeatTime Field is not a number)

  解決方法：数値が入力されているか確認してください。それでも解決しない場合は、関連担当者に連絡してください。

- OOO の値が大きすぎる (Value of OOO Overflow)

  例：繰り返し回数の値が大きすぎる (Value of RepeatTime Overflow)

解決方法：値が 2^{32} 未満であるか確認してください。それでも解決しない場合は、関連担当者に連絡してください。

- OOO の値が小さすぎる (OOO must larger than zero)

  例：繰り返し回数の値が小さすぎる (RepeatTime must larger than zero)

  解決方法：値が0 以上であるか確認してください。それでも解決しない場合は、関連担当者に連絡してください。

- OOO が不正なパス (The Input of OOO is not a Path)

  例：盤面パスが不正なパス (The Input of BoardPath Field is not a Path)

  解決方法：盤面パスに盤面ファイルのパスが正しく入力されているか確認してください。絶対パスの使用をお勧めします。それでも解決しない場合は、関連担当者に連絡してください。

- ファイルに読み取り権限がない (The Input of BoardPath Field can not Read)

  解決方法：ファイルの読み取り権限を確認してください。それでも解決しない場合は、関連担当者に連絡してください。

- AI プログラムの実行権限がない (Search Permission denied)

  解決方法：AI プログラムに実行権限があるか確認してください。それでも解決しない場合は、関連担当者に連絡してください。

- HUMAN 接続モード使用時のブラウザモード設定エラー (In order to use HUMAN mode, Please change the browser mode.)

  解決方法：局面表示モードをリモートまたはデフォルトブラウザに設定してください。それでも解決しない場合は、関連担当者に連絡してください。

10.5 サーバー通信エラー

ゲーム中にサーバーへのデータ送信または受信でエラーが発生（Figure 58 参照）。

- データ送信エラー (SEND TO SERVER ERROR)
- データ受信エラー (RECV FROM SERVER ERROR)

Figure 58: サーバー通信エラー

解決方法：ネットワーク状況を確認し、安定した回線を使用してください。それでも解決しない場合は、関連担当者に連絡してください。

10.6 STDIO 通信エラー (STDOUT SEND DATA TO SEARCH ERROR)

ゲーム開始時に Figure 59 のエラーが発生し、Stdout が Search にデータを送信できない。

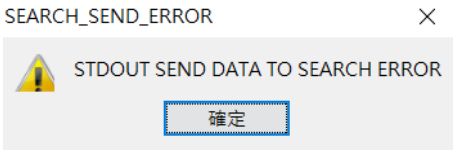


Figure 59: STDIO 通信エラー

解決方法：接続モードの設定を確認してください。多くの場合、接続モードの設定ミスが原因です。それでも解決しない場合は、関連担当者に連絡してください。

11 参考資料と連絡先

1. 日本 geister サーバー GitHub (リンク)
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.

もしまだ不明な点がございましたら、担当者にご連絡ください：

- 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