**Hidden Random Chess**

Hidden Random Chess is a proof-of-concept online chess game where each player's back rank is randomized (inspired by Chess960) and opponent pieces remain hidden until a capture occurs. The project uses Node.js with Express and Socket.io on the server side, integrates chess.js for move validation and game state management, and includes chessboard.js via CDN for potential UI enhancements.

**Features**

* **Randomized Back Rank:**  
  Each player's back rank is randomized using Chess960-style rules.
* **Hidden Opponent Pieces:**  
  Opponent pieces are shown as generic tokens (●) until they are revealed by a capture.
* **Real-Time Multiplayer:**  
  Socket.io enables real-time communication between two players.
* **Move Validation & Game State:**  
  Uses chess.js for verifying legal moves and maintaining game state.
* **Extensible UI:**  
  Although the demo uses custom HTML/CSS for board rendering (to support the hidden information mechanic), chessboard.js is included via CDN for potential future enhancements.

**Project Structure**

php

Copy

hidden-chess/

├── package.json # Project configuration and dependencies.

├── server.js # Main server file (Express and Socket.io setup, game logic).

└── public/

└── index.html # Client-side code (HTML, CSS, and JavaScript for the game interface).

**Installation**

**Prerequisites**

* [Node.js](https://nodejs.org/) (v10 or later recommended)
* npm (Node Package Manager)

**Setup Steps**

1. **Clone or Create the Project Folder**

Open a terminal and create or clone the repository:

bash

Copy

mkdir hidden-chess

cd hidden-chess

1. **Initialize the Project**

Initialize a new Node.js project:

bash

Copy

npm init -y

1. **Install Dependencies**

Install Express, Socket.io, and chess.js by running:

bash

Copy

npm install express socket.io chess.js

**Note:**

* + You may see messages about funding or npm updates. These are informational and do not affect your project.
  + If you receive syntax errors related to chess.js, consider downgrading it:

bash

Copy

npm install chess.js@0.10.3

Alternatively, update your project to use ES Modules as described in the project documentation.

1. **Verify the File Structure**

Ensure your project folder contains:

* + package.json
  + server.js
  + A folder named public with index.html inside it.

**Usage**

**Running the Server**

From the project root, run:

bash

Copy

node server.js

You should see a message like:

arduino

Copy

Server running on port 3000

**Testing the Game**

1. Open your web browser and navigate to http://localhost:3000.
2. The first window will display "Waiting for an opponent…".
3. Open a second browser window or tab and navigate to the same URL.
4. The two players will be paired into a game.
5. Click on one of your pieces, then click on the destination square to move. Opponent pieces will remain hidden (displayed as ●) until a capture reveals them.

**Troubleshooting**

* **SyntaxError / Unexpected Token:**
  + This error may be caused by incompatibility with the module type used in chess.js.
    - **Option 1:** Downgrade chess.js using:

bash

Copy

npm install chess.js@0.10.3

* + - **Option 2:** Convert your project to ES Modules by adding "type": "module" to your package.json and updating your require statements to import statements.
* **Dependency Issues:**
  + Ensure you are running commands from the correct project directory and that all dependencies are installed (check for the node\_modules folder).
* **npm Update Notice:**
  + The message about a new npm version is informational. You may update npm using:

bash

Copy

npm install -g npm

but this is optional.

**Future Enhancements**

* **Enhanced Move Validation & Game Rules:**  
  Integrate full chess rules with chess.js (support for check, checkmate, en passant, castling, and pawn promotion).
* **Improved UI Using chessboard.js:**  
  Replace the custom board rendering with chessboard.js for a more polished visual interface. Custom modifications can allow the hidden-information mechanic to persist.
* **Robust Game State Management:**  
  Implement persistent sessions, better handling of disconnects, and game state cleanup.

**License**

This project is provided as a proof-of-concept and is free to modify and extend. Feel free to contribute or provide feedback.

**Acknowledgments**

* [Express](https://expressjs.com/)
* [Socket.io](https://socket.io/)
* [chess.js](https://github.com/jhlywa/chess.js)
* [chessboard.js](https://chessboardjs.com/)

Feel free to modify and extend this project to suit your needs. Enjoy building and playing Hidden Random Chess!

-ZZ

A screenshot of a game

Description automatically generated

A screenshot of a game

Description automatically generated

