**Two-Player Strategy Game**

Welcome to the **Two-Player Strategy Game**! This is a real-time, turn-based strategy game that you can play in your browser. The game is built using HTML, CSS, JavaScript, and WebSockets.

**Features**

* **Real-time Multiplayer:** Play against another player in real-time using WebSockets.
* **Turn-Based Gameplay:** Players take turns moving their pieces on a 5x5 grid.
* **Simple Controls:** Easy-to-use controls for moving pieces across the board.
* **Automatic Game Start:** The game begins automatically when two players join.

**How to Play**

**Objective**

The goal of the game is to outmaneuver your opponent by moving your pieces strategically across the board.

**Setup**

1. Clone this repository to your local machine:

bash

**git clone https://github.com/your-username/two-player-strategy-game.git**

1. Navigate to the project directory:

bash

**cd two-player-strategy-game**

1. Install the necessary dependencies:

bash

**npm install**

1. Start the server:

bash

Copy code

node server.js

1. Open two separate browser windows (or tabs) and navigate to http://localhost:3000 in both. Each window will be assigned as either Player A or Player B.

**Controls**

* **L:** Move left
* **R:** Move right
* **F:** Move forward
* **B:** Move backward
* **FL:** Move forward-left diagonally
* **FR:** Move forward-right diagonally
* **BL:** Move backward-left diagonally
* **BR:** Move backward-right diagonally

**Gameplay**

* **Player A** and **Player B** are assigned when the first and second browsers connect.
* The game will start automatically when both players have joined.
* Players take turns moving their pieces.
* The current player’s turn is displayed at the top of the game board.

**Winning the Game**

The game continues until one player meets the winning conditions, which are defined by the specific rules of your game.

**Project Structure**

* **index.html:** The main HTML file that sets up the game UI.
* **script.js:** The JavaScript file that handles game logic, user interactions, and WebSocket communication.
* **server.js:** The Node.js server file that manages WebSocket connections and game state.
* **styles.css:** The CSS file that styles the game board and controls.

**Dependencies**

* **Node.js:** The server-side environment to run the WebSocket server.
* **ws:** A WebSocket library for Node.js.

**Acknowledgments**

* Inspired by HitWicket Problem Statement.
* Built with love and lots of debugging