Here’s a `README.md` file for your Reversi game project:

Reversi

About Reversi

Reversi, also known as Othello, is a classic strategy board game played on a grid. Players take turns placing their pieces on the board, with the goal of having the majority of their pieces displayed at the end of the game. This implementation of Reversi allows you to play against an AI opponent (MinMax Agent).

How to Run the Game

To run the game, use the following command:

```
run
```

Customization

Adjusting the Board Size

You can adjust the size of the game board by modifying the `ROWS` and `COLUMNS` constant in the `constants.h` file. For example:
rows and columns must be equale

```
#define ROWS 6  // Change this to 8 for an 8x8 board
```

Game Difficulty

The difficulty of the AI opponent depends on the **depth** parameter in the `MinMax` algorithm. You can adjust the depth in the `MinMax.c` file the depth controller function:

```
// i : number of moves played in the game;
int Depth_Controller(int i){ // Increase for a harder AI, decrease for an easier AI
    int Depth;

    if (i < 10)
    {
        Depth = 4;
    }
    else{
        Depth = 5;
    }
    return Depth;
    }
```

A higher depth value makes the AI think further ahead, increasing its difficulty but also its computation time.

Game Accuracy

The accuracy of the AI's moves depends on the **evaluateBoard** function in `MinMax.c`. This function determines how the AI evaluates the board state. You can tweak this function to improve or modify the AI's decision-making process.

```c
int evaluateBoard(int board[BOARD_SIZE][BOARD_SIZE]) {
    // Add your evaluation logic here
}
```

Project Structure

- **`constants.h`**: Contains constants like `ROWS`.
- `MinMax.c`: Implements the MinMax algorithm with alpha-beta pruning for the AI.
- **`reversi.c`**: Contains the game logic.
- **`GameLoop.c`**: Contains the game loop.
- **`main.c`**: To choose the game loop type like `MinMaxAgentWihtRandomPlayer`.
- **`README.md`**: This file, providing an overview of the project.

Requirements

- A C compiler (e.g., GCC).
- A terminal or command prompt to run the game.

How to Play

1. Run the game using the `run` command.
2. Follow the on-screen instructions to place your pieces.
3. The game ends when the board is full or no valid moves are left. The player with the most pieces wins.

Contributing

Feel free to contribute to this project by:

- Improving the AI logic.
- Adding new features (e.g., GUI, multiplayer mode).
- Fixing bugs or optimizing the code.

License

This project is open-source and available under the MIT License. See the `LICENSE` file for details.

Enjoy playing Reversi! Let me know if you have any questions or need further assistance.