virtual 1v1 checkers fight

1. Title

Implementation of virtual 1v1 checkers fight in C++

2. Abstract

This project involves creating a console-based virtual 1v1 checkers fight in C++. The game supports two players, providing a platform for strategic moves, captures, and king promotions. The primary goal is to develop an interactive and functional implementation of the classic virtual 1v1 checkers fight

3. Objectives

- **Game Logic:** Implement the core rules and logic of the Checkers game, including legal moves, captures, and king promotions.
- **Data Structures: 2D array**
- link list
- Stack
- Queue
- Back tracking:
- **User Interface:** Create a user-friendly console interface for players to input moves and view the game board.
- **Error Handling:** Implement robust error handling to ensure valid user inputs and prevent runtime issues.
- **Game Termination:** Develop a mechanism to determine and display the winner or a draw at the end of the game. When opponents is finished only one player is left

4. Features

- **Board Representation:** Utilize a 2D array to represent the game board, with different symbols denoting player pieces, kings, and empty spaces.

Player Class: Implement a `Player` class to store player information, including name and piece type.

- **Move Validation:** Develop functions to validate user moves, ensuring adherence to the game rules and preventing illegal moves.
- **Capture Mechanism:** Implement the ability to capture opponent pieces by leaping over them.
- **King Promotion:** Allow regular pieces to be promoted to kings when reaching the opposite end of the board.
- **User Input:** Create an input mechanism for players to specify their moves during their turns.

- **Game Over Check:** Implement a function to check for the end of the game based on the remaining pieces and declare a winner.

5. Conclusion

This project will result in a functional 1v1 checkers fight implemented in C++, providing a valuable opportunity to apply programming skills, understand game development principles, and enhance problem-solving abilities. The combination of data structures, user input handling, and game logic implementation will contribute to a comprehensive programming experience.