Project Overview: Game & Data Structures

Project Description

This project implements a game involving strategic moves, utilizing advanced data structures like linked lists and trees. The game logic is carefully structured to handle player moves, maintain game state, and optimize decision-making processes.

Key Components

1. Game Logic

Files: game (1).c, game (1).h

Description: Defines the core mechanics of the game, including initializing game boards, handling player moves, and ensuring proper memory management. The logic includes functions to evaluate possible moves and update the game state accordingly.

2. List Management

Files: Lists (1).h, Lists (2).c

Description: Implements linked list structures to manage sequences of moves or game states. Provides essential list operations like insertion, deletion, and traversal, ensuring efficient data handling.

3. Tree Structures

Files: Trees (1).c, Trees (1).h

Description: Uses tree structures to represent possible moves in the game, allowing for strategic decision-making. Functions are included to build, traverse, and manage these trees, representing the game's decision-making process.

4. Main Program

File: main (2).c

Description: Serves as the entry point for the program, coordinating the interaction between game logic, list management, and tree structures. Ensures a smooth flow of the game from start to finish.

Features

- Strategic Gameplay: The project uses advanced data structures to create a game that requires strategic thinking.

- Modular Design: Each component (game logic, lists, trees) is modular and well-organized, making the code easy to maintain and extend.

- Efficient Memory Management: Careful allocation and deallocation of memory to prevent leaks and ensure smooth performance.

Technologies Used

- Programming Language: C

- Data Structures: Linked Lists, Trees

- Memory Management: Dynamic memory allocation