Institute of Information Technology

University of Dhaka

Course Code: CSE 3117

Course name: Artificial Intelligence

Project Report: MiniChess (6×5 Board) with AI Opponent

Submitted by:

MD Morshaline Mojumder (BSSE-1504)
MD Rony Rahman (BSSE-1509)
MD Israfil Hossain (BSSE-1508)

Submitted to:

Dr. Ahmedul Kabir
Associate Professor,iit
Gmail: kabir@iit.du.ac.bd

GitHub Repository: `https://github.com/morshaline1504/MINICHESS`

1. Introduction

Artificial Intelligence (AI) plays a crucial role in developing intelligent agents capable of making optimal decisions in competitive environments. One of the most classical applications of AI is adversarial game playing, where an agent must reason about its own moves as well as the possible responses of an opponent. Such games are typically modeled as two-player, fully observable, deterministic, zero-sum games.

This project presents the design and implementation of a **MiniChess** game played on a **6×5 board** with an intelligent AI opponent. MiniChess is a simplified version of standard chess that reduces board size and complexity while preserving strategic depth. The AI opponent uses the **Minimax algorithm** combined with **Alpha-Beta pruning**, a heuristic **evaluation function**, and an **early stopping mechanism** to provide efficient and challenging gameplay.

The primary objective of this project is to demonstrate core Artificial Intelligence concepts taught in the course, including game tree search, heuristic evaluation, optimization techniques, and interactive AI systems.

2. Game Description and Implementation

2.1 Board Setup and Rules

The MiniChess variant implemented in this project uses a **6×5 board**. Each player starts with a reduced set of chess pieces while maintaining standard chess movement rules.

Initial Board Configuration:

- Row 0 (Black): Rook, Knight, Queen, King, Bishop
- Row 1 (Black Pawns): Five pawns
- Rows 2–3: Empty squares
- Row 4 (White Pawns): Five pawns
- Row 5 (White): Rook, Knight, Queen, King, Bishop

Standard chess rules apply, including capturing, check, checkmate, and pawn promotion (to queen). One bishop per side is removed to fit the smaller board size.

2.2 Game Modes

The following game modes are supported:

- Human vs Human

- Human vs AI
- AI vs AI

Players can choose their color and adjust AI difficulty from depth 1 (Beginner) to depth 4 (Expert).

3. System Architecture and Design

The system follows a modular architecture to ensure clarity, maintainability, and extensibility.

3.1 Core Components

- **Board Representation:** 2D array representing the 6×5 board
- **Move Generator:** Generates all legal moves
- **Game State Manager:** Tracks turns and terminal states
- **AI Engine:** Minimax with Alpha-Beta pruning
- **User Interface Module:** Handles rendering and interaction

4. Minimax Search Algorithm

The AI decision-making is based on the **Minimax algorithm**, which assumes optimal play from both players.

4.1 Strategy

- Black pieces act as the maximizing player
- White pieces act as the minimizing player

The algorithm recursively explores future board states and assigns evaluation scores to determine the optimal move.

4.2 Terminal Conditions

The search terminates when:

- A predefined depth limit is reached
- A king is captured (checkmate)
- The move count exceeds 100 (draw)

5. Alpha-Beta Pruning

Alpha-Beta pruning improves minimax efficiency by eliminating branches that cannot affect the final decision.

5.1 Pruning Logic

- Alpha (α): Best value for the maximizing player
- Beta (β): Best value for the minimizing player
- Pruning occurs when $\beta \leq \alpha$

This optimization reduces node evaluation by approximately 50–70%.

6. Evaluation Function Design

The evaluation function estimates board strength using multiple heuristics.

6.1 Material Values

- Pawn: 100
- Knight: 320
- Bishop: 330
- Rook: 500
- Queen: 900
- King: 20000

6.2 Positional and Tactical Factors

- Pawn advancement bonuses
- Knight centralization
- King safety
- Center control (+10)
- Check penalty (-50)
- Checkmate score ($\pm$100000)

7. Early Stopping Mechanism

To prevent excessive computation time, a time-based cutoff is applied.

- Time limit: 5000 ms per move
- Iterative deepening ensures a valid move is always available

8. User Interface

The interface is built using HTML, CSS, and JavaScript with the following features:

- Visual 6×5 chessboard
- Unicode chess symbols
- Highlighted legal moves
- Move history and undo support
- Responsive design for desktop and mobile

9. Testing and Performance

Testing shows:

- Depth 1–2: Beginner friendly
- Depth 3: Balanced gameplay
- Depth 4: Strong and challenging AI

At depth 3, the AI evaluates approximately 1800 nodes with an average decision time of 800 ms.

10. Conclusion and Future Work

This project successfully demonstrates adversarial search using minimax with alpha-beta pruning. The AI plays competitively while maintaining responsive performance.

10.1 Future Improvements

- Transposition tables
- Improved move ordering
- Opening books and endgame databases
- Machine learning-based evaluation

Tools and Technologies

- JavaScript
- HTML, CSS
- Minimax with Alpha-Beta Pruning
- GitHub