



Incredible Chess Board

Course Name : Artificial Intelligence Laboratory
Course Code : CSE -4110
Date of Submission : 04.09.2024

Submitted to

Md. Shahidul Salim
Lecturer
Department of Computer
Science and Engineering
KUET
Khulna -9203, Bangladesh.

Most. Kaniz Fatema Isha
Lecturer
Department of Computer
Science and Engineering
KUET
Khulna -9203, Bangladesh.

Submitted By

S M Zobayer Abedin
Roll : 1907045
Group : A1
Year : 4th
Semester : 1st

Md Kawsar Ahmed Kazol
Roll : 1907063
Group : A1
Year : 4th
Semester : 1st

Contents

1. Introduction.....	3
2. Theory	3
Game Rules	3
3. Algorithm used in game	5
Alpha-Beta Pruning for "Incredible Chess"	5
How A* Search Works in "Incredible Chess"	5
Benefits in "Incredible Chess"	6
4. Features	6
5. Pseudocode	8
I. Minimax with Alpha-Beta Pruning	8
II. AI Move Selection.....	8
6. Game scenario	9
7. Discussion	11
I. Key Aspects.....	11
II. Challenges	12
8. Conclusion	12

1. Introduction

"Incredible Chess" is an innovative and distinctive variant of the traditional chess game, offering a fresh and unique take on one of the most enduring and beloved board games in history. Unlike conventional chess, where players manage a variety of pieces with different movement patterns, "Incredible Chess" simplifies the game by featuring only pawns. However, these pawns come with a twist: they possess the ability to move forward or backward by any positive number of steps, as long as they do not jump over any other piece. This new movement dynamic adds a layer of strategy and complexity to the game, challenging players to think several moves ahead and consider multiple variables in their decision-making process.

This report delves into the development of "Incredible Chess," which has been brought to life through the use of Python and the Pygame library. The choice of Python as the programming language, combined with Pygame's powerful capabilities for game development, provides an ideal platform for creating a game that is both visually appealing and functionally robust. The game's design focuses on delivering an intuitive user experience, ensuring that both seasoned chess players and newcomers can enjoy the novel gameplay mechanics without a steep learning curve.

In summary, "Incredible Chess" is more than just a variation of a classic game; it is a thoughtful reimagining that introduces new strategic elements and offers a fresh challenge for chess enthusiasts. This report will explore the technical details of the game's development, focusing on the programming and algorithmic strategies employed to bring this unique chess variant to life. Through this exploration, the report aims to provide insights into the complexities of game development, particularly in creating an AI that can competently compete against human players in a game as strategically rich as chess.

2. Theory

Game Rules

I. Board Setup

"Incredible Chess" is played on an 8x8 chessboard. Each column contains one white pawn and one black pawn. Initially, white pawns are positioned in the lower half of the board (the first row of each column), while black pawns are placed in the upper half (the last row of each column). This setup places the pieces directly opposite each other, creating immediate strategic interactions in each column.

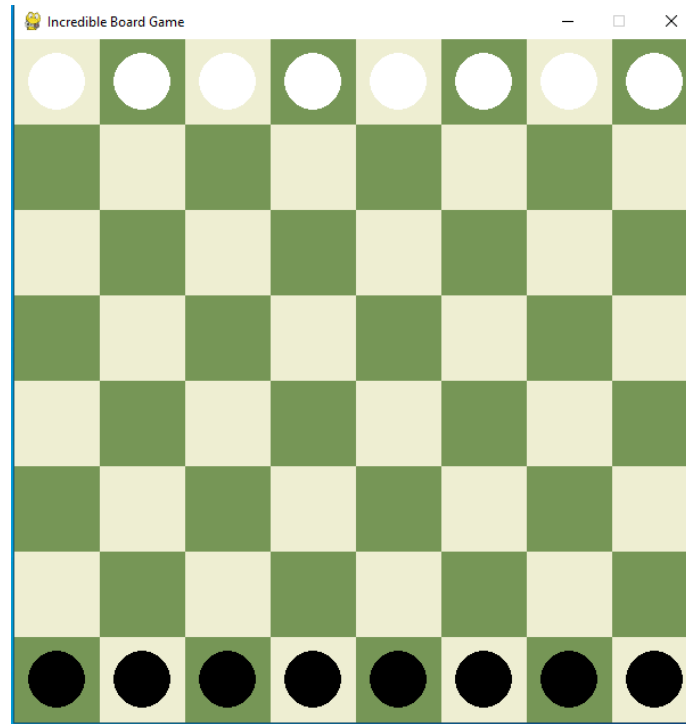


Figure 2.1: chess board

II. Pawn Movement

In "Incredible Chess," pawns move vertically forward or backward by any number of squares within their column. Unlike standard chess, pawns cannot jump over other pieces and must navigate around any obstacles. This movement flexibility adds strategic depth, requiring players to carefully plan their moves to both advance their position and block their opponent's options.



Figure 2.2 : Pawn

III. Objective and Winning Conditions

The goal of "Incredible Chess" is to strategically restrict your opponent's moves. Players take turns moving their pawns according to the rules, and the game ends when one player has no valid moves left. This could be due to blocked paths or disadvantageous positions. The player with no remaining valid moves loses, and their opponent is declared the

winner. The game emphasizes strategic control and movement restriction, offering a unique twist on traditional chess.

3. Algorithm used in game

Minimax algorithm for “incredible chess”

The Minimax algorithm in "Incredible Chess" evaluates all possible moves for each player, simulating future game states to choose the optimal move. It minimizes the potential maximum loss by assuming the opponent plays optimally, using alpha-beta pruning to eliminate irrelevant branches and enhance decision-making efficiency.

Alpha-Beta Pruning for "Incredible Chess"

How Alpha-Beta Pruning Works in "Incredible Chess"

- I. **Alpha and Beta Values:**
 - a. **Alpha** represents the best value that the maximizer (AI) can guarantee at that level or above. Initially set to negative infinity, it increases as the AI finds better options.
 - b. **Beta** represents the best value that the minimizer (opponent) can guarantee at that level or below. Initially set to positive infinity, it decreases as the opponent finds better options.
- II. **Pruning Process:**
 - a. **Alpha Cutoff:** If the value of a node is greater than or equal to beta, the minimizer (opponent) will not allow this branch to be chosen, so the remaining branches under this node are pruned.
 - b. **Beta Cutoff:** If the value of a node is less than or equal to alpha, the maximizer (AI) will not consider this branch, and further exploration is unnecessary.

How A* Search Works in "Incredible Chess"

1. **Search Initialization:**
 - The algorithm starts with the current state of the game (the board setup, including the positions of all pawns).
 - The goal is to find the best move that will maximize the AI's advantage or minimize the player's potential to win.
2. **Node Representation:**
 - Each possible move or board state is considered a "node" in the search graph.
 - A node represents the game state after a potential move by a pawn, including the new positions of all pieces on the board.
3. **Heuristic Function (h(n)):**

- The A* algorithm uses a heuristic function to estimate the cost (or distance) from the current node to the goal.
 - In "Incredible Chess," this heuristic might evaluate factors such as how close a pawn is to restricting the opponent's moves, or how well the current pawn positions can defend against the opponent's next move.
4. **Cost Function ($g(n)$):**
- The cost function calculates the actual cost to reach a particular node from the start node.
 - In the context of "Incredible Chess," this might involve counting the number of moves made so far or the strategic advantage gained.
5. **Evaluation Function ($f(n)$):**
- The evaluation function combines the heuristic and cost functions: $f(n) = g(n) + h(n)$.
 - A* uses this combined score to prioritize which nodes to explore, always selecting the node with the lowest $f(n)$ value, meaning the most promising path towards the goal.
6. **Pathfinding and Move Selection:**
- A* expands nodes (possible moves) by exploring their neighboring nodes (subsequent possible moves).
 - It continues this process, updating the $f(n)$ scores for each node, until it finds the move that leads to the best possible outcome for the AI.
 - Once the best path is identified, the AI executes the move, positioning its pawns optimally based on the analysis.
7. **Backtracking and Pruning:**
- If a path turns out to be less promising (higher $f(n)$), A* backtracks and prunes it from further consideration, focusing on more promising paths.
 - This ensures the AI doesn't waste time on suboptimal moves.

Benefits in "Incredible Chess"

- **Strategic Decision-Making:** A* ensures that the AI chooses moves that not only provide immediate advantages but also set up future success.
- **Efficiency:** By focusing only on the most promising paths, A* reduces the number of nodes (moves) the AI needs to evaluate, speeding up decision-making.
- **Adaptability:** The heuristic can be tailored to specific strategies in "Incredible Chess," such as prioritizing moves that block the opponent or improve pawn positions.

4. Features

I. Graphical Interface

- The game features an intuitive graphical interface developed using Pygame, allowing seamless interaction. The chessboard is clearly displayed with alternating colored squares, making it easy for players to select pawns with mouse clicks. The visual design ensures a user-friendly experience.

II. AI Opponent

- The AI opponent uses the Minimax algorithm with alpha-beta pruning, playing as white and always taking the first move. It evaluates all possible moves to maximize its chances of winning, providing a challenging experience for the player.

III. Pawn Movement

- Pawns in "Incredible Chess" can move vertically forward or backward by any positive number of steps but cannot jump over other pieces. This rule adds strategic depth as players must avoid blocking their own pawns while outmaneuvering the opponent.

IV. Valid Move Indicators

- When a pawn is selected, the game highlights all valid moves, helping players make informed decisions and reducing the likelihood of illegal moves. This enhances the overall gameplay experience.

V. Turn-Based Gameplay

- The game alternates turns between the player (controlling black pawns) and the AI (controlling white pawns), ensuring a balanced and fair game flow.

VI. Endgame Detection

- The game detects when a player has no valid moves left, declaring that player the loser. The winner is displayed on the screen, providing a clear conclusion to the match.

VII. Interactive Input

- Players can interact using both mouse and keyboard. Pawns are selected by clicking, and valid moves are highlighted. Arrow keys can be used to move selected pawns, making the game accessible and easy to control.

VIII. Scalability and Modifiability

- The game's modular code structure allows for easy modifications and scalability. Developers can add new features or change rules with minimal rework, making it an adaptable foundation for further development.

5. Pseudocode

I. Minimax with Alpha-Beta Pruning

```
function minimax(board, depth, isMaximizing, alpha, beta):
    if depth == 0 or terminal state is reached:
        return evaluate(board)

    if isMaximizing:
        maxEval = -infinity
        for each move in get_all_moves(board, 'W'):
            boardCopy = copy(board)
            apply_move(boardCopy, move)
            eval = minimax(boardCopy, depth - 1, False, alpha, beta)
            maxEval = max(maxEval, eval)
            alpha = max(alpha, eval)
            if beta <= alpha:
                break
        return maxEval
    else:
        minEval = infinity
        for each move in get_all_moves(board, 'B'):
            boardCopy = copy(board)
            apply_move(boardCopy, move)
            eval = minimax(boardCopy, depth - 1, True, alpha, beta)
            minEval = min(minEval, eval)
            beta = min(beta, eval)
            if beta <= alpha:
                break
        return minEval
```

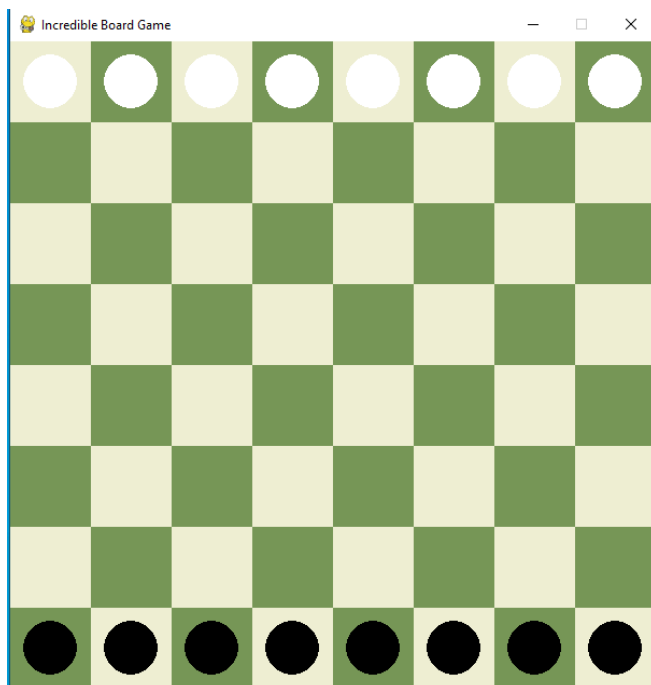
II. AI Move Selection

```
function computer_move(board):
    bestScore = -infinity
    bestMove = null
    for each move in get_all_moves(board, 'W'):
        boardCopy = copy(board)
        apply_move(boardCopy, move)
```



```
score = minimax(boardCopy, 3, False, -infinity, infinity)
if score > bestScore:
    bestScore = score
    bestMove = move
apply_move(board, bestMove)
```

6. Game scenario



This is the first scenario of our game where Black is AI pawn and White is player's pawn. Firstly player should turn his move .

Figure 6.1 : First scenario

It shows the indicator in red color where the pawn can move .Pawn can go One ,two or three move at a time .

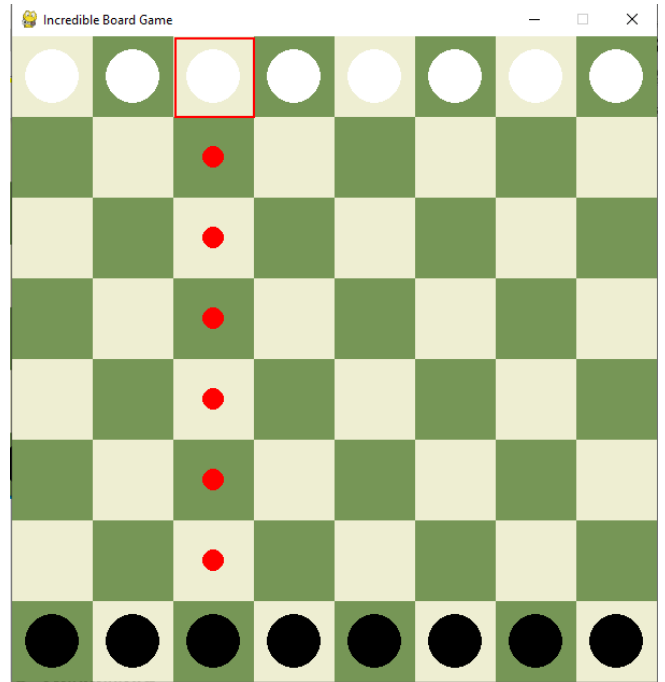


Figure 6.2 :Player turns his move

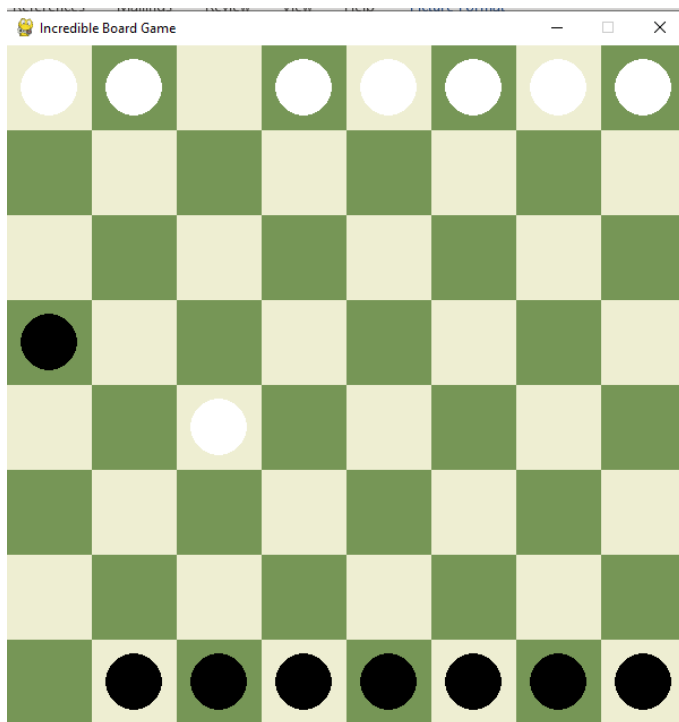


Figure 6.3 : Ai move its turn

When white plays his move then AI also plays the black pawn . White moves first, followed by the AI-controlled black pawns, aiming to restrict opponent's moves.

After playing the game who will be winner it will be displayed in the console.

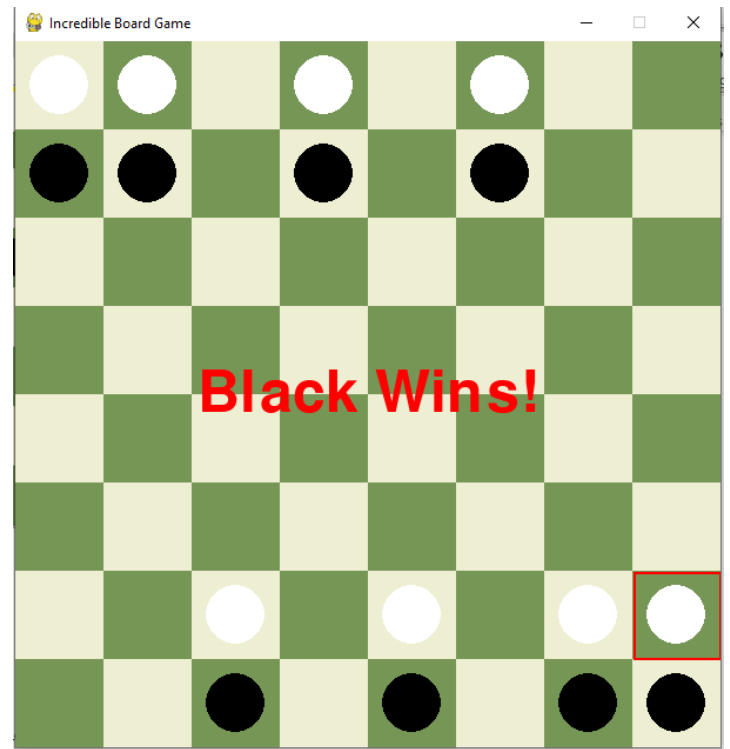


Figure 6.4 : Last scenario of game

7. Discussion

The implementation of "Incredible Chess" using Pygame provides an interactive and visually appealing game experience. The integration of the Minimax algorithm with alpha-beta pruning enables the AI to make strategic decisions, offering a challenging gameplay experience.

I. Key Aspects

- Efficiency:** Alpha-beta pruning significantly reduces the number of nodes evaluated in the game tree, making the AI more efficient.
- Strategy:** The AI considers all possible moves for each pawn and selects the optimal move, ensuring competitive gameplay.

- c. **User Interaction:** The graphical interface allows players to interact with the game intuitively, making it accessible and engaging.

II. Challenges

- a. **Complexity:** Implementing an efficient Minimax algorithm with alpha-beta pruning requires careful consideration of all possible moves and game states.
- b. **Performance:** Ensuring the AI makes decisions in a reasonable timeframe, especially with increased depth in the Minimax algorithm, can be challenging.

8. Conclusion

"Incredible Chess" is a compelling reimagining of traditional chess that introduces a fresh and challenging gameplay experience through the use of only pawns with unique movement capabilities. The integration of an AI opponent, powered by the Minimax algorithm with Alpha-Beta Pruning, significantly enhances the strategic depth of the game, allowing the AI to make well-calculated moves that challenge players at every turn. The Pygame library plays a crucial role in creating a visually appealing and interactive interface, making the game accessible and enjoyable for a broad audience. This project not only highlights the potential of combining artificial intelligence with game development to create innovative and engaging experiences but also serves as a practical example of how AI techniques can be applied to enhance gameplay in traditional board games. Looking ahead, "Incredible Chess" offers numerous opportunities for future enhancements, such as introducing varying difficulty levels, multiplayer modes, or additional game pieces, all of which could further enrich the gameplay and extend its appeal. Ultimately, "Incredible Chess" demonstrates the exciting possibilities that arise when classic games are infused with modern AI technology, resulting in a unique and entertaining application that both challenges and delights players.