Project Title: Exploding Checkers AI Game

Submitted By:

- Muneeb Maqsood Khan (22k4481)
- Hashir Ahmed Khan (22k4419)

Course: Artificial Intelligence

Instructor: Miss Ravia Ijaz

Submission Date: 16 May 2025

---

1. Executive Summary

Project Overview:
This project aimed to design and implement an AI-powered version of a modified checkers game called *Exploding Checkers*. The traditional gameplay of checkers has been modified to include "exploding" mechanics when a piece is captured. The AI component, powered by the Minimax algorithm with Alpha-Beta pruning, was developed to enable intelligent move selection against a human player.

---

2. Introduction

Background:
Checkers is a classic two-player strategy board game where players move their pieces diagonally and capture opponent pieces by jumping over them. The game is known for its simplicity and strategic depth. We chose this game to explore the integration of AI in a turn-based environment and added unique "exploding" mechanics to increase complexity and decision-making depth.

Objectives of the Project:

- To design and implement a playable version of Exploding Checkers.
- To integrate an AI using the Minimax algorithm with Alpha-Beta pruning.

- To test the AI's performance against human players in various scenarios.

---

3. Game Description

Original Game Rules:
In traditional checkers, each player controls 12 pieces on an 8×8 board. Players move pieces diagonally forward and capture opponent pieces by jumping over them. The game ends when one player captures all opponent pieces or blocks all possible moves.

Innovations and Modifications:

- Introduction of an "exploding" mechanic: when a piece is captured, it causes surrounding pieces (in adjacent squares) to be removed as well.
- Simplified version with a smaller board size for faster testing and development.
- AI-enabled mode allowing a human player to compete against a Minimax-powered opponent.

---

4. AI Approach and Methodology

AI Techniques Used:
We used the Minimax algorithm combined with Alpha-Beta pruning to optimize move selection for the AI player. The algorithm simulates all possible future game states and prunes branches that don't affect the final decision, enhancing efficiency.

Algorithm and Heuristic Design:
The AI uses a custom evaluation function that considers the number of pieces, control of the center board, and the risk posed by nearby opponents. Exploding mechanics were integrated into the evaluation logic to account for collateral effects of a capture.

AI Performance Evaluation:
The AI was tested in multiple matches against human players. Metrics included win rate, average decision time per move, and adaptability to different board states. The AI demonstrated strong performance, with a win rate exceeding 70% and an average move decision time under 1.5 seconds.

---

5. Game Mechanics and Rules

Modified Game Rules:

- Capturing an opponent's piece causes it to "explode", removing adjacent pieces (both friendly and enemy).
- A simplified board size (e.g., 6×6) to accommodate faster gameplay.
- Traditional movement and capture mechanics are preserved.

Turn-based Mechanics:
Players alternate turns. On each turn, a player may move a piece diagonally forward or capture an opponent. If a capture occurs, the explosion logic is triggered, and the game state is updated accordingly.

Winning Conditions:
The game ends when one player has no remaining pieces or cannot make a legal move. The player with remaining pieces is declared the winner.

---

6. Implementation and Development

Development Process:
The game was implemented in Python, focusing first on rule design and gameplay logic, followed by AI integration. Extensive testing was done to handle edge cases arising from the explosion mechanic.

Programming Languages and Tools:

- **Programming Language:** Python
- **Libraries:** `random`, `pygame`, `sys`, `math`

Challenges Encountered:

- Designing explosion logic that didn't break the game loop or introduce bugs in chain reactions.
- Optimizing the Minimax algorithm to handle deeper lookahead within a reasonable time frame.

- Balancing the AI's evaluation function to avoid overly aggressive or passive behavior.

---

7. Team Contributions

- **Muneeb Maqsood Khan (22k4481):**
  Responsible for AI algorithm development, including Minimax and Alpha-Beta pruning logic. Also worked on performance tuning and heuristic function design.

- **Hashir Ahmed Khan (22k4419):**
  Implemented game logic, explosion mechanic, and board state transitions. Also handled user interface and game interaction elements.

---

8. Results and Discussion

AI Performance:
The AI showed excellent decision-making capabilities, maintaining a consistent edge in gameplay against human opponents.

- **Win Rate:** 70%+

- **Average Move Time:** ~1.3 seconds

- **Strengths:** Strategic captures, optimal sacrifices, handling explosion logic well

- **Weaknesses:** Occasional vulnerability to bait strategies due to aggressive evaluation heuristics

---

9. References

- Russell, S., & Norvig, P. (2021). *Artificial Intelligence: A Modern Approach*.

- Online articles on Minimax and Alpha-Beta Pruning from GeeksforGeeks.

- Pygame documentation