Exploding Checkers: A Strategic AI Board Game

Group Members: Muneeb Maqsood Khan(22K-4481), Hashir Ahmed Khan(22K-4419)
Course: AI
Instructor: Miss Alina Arshad / Miss Ravia Ijaz

1. Project Overview

Project Topic:

Exploding Checkers is an innovative twist on the traditional game of checkers, introducing a **controlled explosion mechanic** where players can strategically detonate pieces to alter the board state. This adds an extra layer of strategy, requiring players to balance **normal movement, capturing, and explosive tactics.**

Objective:

The main goal of this project is to **develop an AI capable of strategically playing Exploding Checkers** by implementing the **Minimax algorithm with Alpha-Beta pruning** and heuristic evaluation. The AI will analyze board positions, assess the benefits of explosions, and predict potential chain reactions to optimize decision-making.

2. Game Description

Original Game Background:

Checkers (also known as Draughts) is a two-player game played on an 8x8 board. Each player starts with **12 pieces** and moves diagonally. Pieces capture opponents by jumping over them, and reaching the last row promotes a piece to a **King**, which can move both forward and backward. The game ends when a player loses all their pieces or has no valid moves.

Innovations Introduced:

- **Controlled Explosions:** Each player has a limited number (e.g., 3) of **explosions** they can trigger instead of a normal move.
- **Explosion Mechanics:** When activated, the selected piece **self-destructs**, removing all adjacent pieces (diagonally & orthogonally).
- **Chain Reactions:** If an explosion eliminates another **Explosive Piece**, it triggers a secondary explosion.

- **Strategic Traps:** Players can manipulate board positioning to force opponents into explosive traps.

Impact on Gameplay:

- Enhances **tactical depth**, requiring players to balance normal moves and explosive actions.
- Introduces **new winning strategies** beyond traditional checkers gameplay.
- Encourages **risk vs. reward decision-making**, as explosions can eliminate multiple pieces at once.

3. AI Approach and Methodology

AI Techniques to be Used:

- **Minimax Algorithm:** Evaluates potential moves and selects the optimal one.
- **Alpha-Beta Pruning:** Optimizes Minimax by eliminating unnecessary calculations.
- **Heuristic-Based Evaluation:** AI assigns a score to different board states based on:
    - Number of remaining pieces.
    - Positional advantage (center control, piece safety, king pieces).
    - Explosion potential and chain reactions.

Complexity Analysis:

- Minimax with Alpha-Beta Pruning reduces the computational cost compared to brute-force search.
- Explosion chains introduce **nonlinear state expansion**, requiring **optimized depth search limits** to ensure AI efficiency.
- Trade-offs between **exploding vs. normal moves** increase decision complexity, making heuristic evaluation critical.

4. Game Rules and Mechanics

Modified Rules:

1. Each player starts with **12 pieces**, as in regular checkers.

2. Players can **move, capture, or trigger an explosion** on their turn.
3. Explosions clear all adjacent pieces (diagonal & orthogonal).
4. Players have **3 explosions per game** (can be adjusted for balance).
5. If an explosion removes another **Explosive Piece**, it triggers a chain reaction.
6. Kings still follow standard movement rules but can **also be detonated** for a larger explosion radius.

Winning Conditions:

- A player **wins by eliminating all opponent's pieces** or **trapping them with no legal moves.**

Turn Sequence:

1. Player decides to **move, capture, or explode**.
2. AI evaluates board state and selects an optimal move.
3. The game continues until a winning condition is met.

5. Implementation Plan

Programming Language:

- **Python** (due to strong AI libraries and easy GUI development with Pygame)

Libraries and Tools:

- **Pygame** (for GUI implementation)
- **NumPy** (for data structures and game state management)
- **Scikit-learn / TensorFlow** (if reinforcement learning is explored)

Milestones and Timeline:

Week	Task
1-2	Finalize game rules and board mechanics
3-4	Develop Minimax AI with Alpha-Beta pruning
5-6	Implement and test game mechanics with GUI
7	Integrate AI with explosion strategy

Week	Task
8	Final testing and report preparation

6. References

- "Artificial Intelligence: A Modern Approach" by Stuart Russell and Peter Norvig.
- Research papers on **Minimax and Alpha-Beta Pruning** for board games.
- Online resources on **Monte Carlo Tree Search (MCTS)** for AI decision-making.