Project Report

TOPIC: Strategic AI Checkers: Human vs AI with Minimax and Alpha-Beta Pruning

Submitted by:

Umer Qureshi 22K-4706

# Project Report

# Strategic AI Checkers: Human vs AI with Minimax and Alpha-Beta Pruning

## 1. Introduction

This report presents the design and implementation of an AI-powered Checkers game. The project involves creating a classic Checkers game with a graphical interface using Pygame, and integrating an AI opponent powered by the Minimax algorithm with Alpha-Beta pruning. The aim is to offer a challenging opponent to human players while demonstrating strategic decision-making and AI capabilities.

## 2. Game Design and Features

### 2.1 Game Overview

Checkers is a two-player strategy board game played on an 8x8 grid. Each player starts with 12 pieces placed on the dark squares of the three rows closest to them. The objective is to capture all the opponent's pieces or block them from making a legal move.

### 2.2 Features Implemented

- Interactive game board using Pygame  
- Human vs AI gameplay  
- AI decision-making using Minimax with Alpha-Beta pruning  
- Visual cues for legal moves and captures  
- Turn-based game progression  
- Support for kinged pieces and multiple jumps

## 3. AI Design and Strategy

### 3.1 Minimax Algorithm

The AI uses the Minimax algorithm to simulate and evaluate possible game states. It recursively explores possible moves for both the AI and the opponent to determine the optimal move that maximizes the AI’s advantage while minimizing the opponent’s chances.

### 3.2 Alpha-Beta Pruning

To enhance performance, Alpha-Beta pruning is used to reduce the number of nodes evaluated by the Minimax algorithm. This allows deeper searches within reasonable time limits, making the AI more responsive and effective.

### 3.3 Heuristic Function

The heuristic function evaluates game states based on several factors:  
- Number of pieces remaining  
- Number of kinged pieces  
- Positional control of the board  
- Potential to perform captures  
These metrics guide the AI in choosing the most advantageous move.

## 4. Implementation Details

The project is implemented in Python using the following tools and libraries:  
- Pygame for the graphical interface and input handling  
- Custom Minimax and Alpha-Beta pruning implementations for AI decision-making  
- Modular code structure for board logic, AI logic, and game loop

## 5. Challenges and Solutions

- Ensuring smooth real-time interaction between AI and human players: Resolved by optimizing search depth.  
- Designing an effective heuristic: Achieved through iterative testing and fine-tuning weights.  
- Handling multiple jumps and king promotions: Implemented custom rules logic to handle special cases accurately.

## 6. Results and Observations

The final game performs well with smooth gameplay and effective AI decision-making. The AI offers a challenging experience for human players. The project demonstrates key concepts of adversarial search and heuristic evaluation.

## 7. Conclusion

This project successfully showcases the integration of classic board game logic with strategic AI. By using Minimax and Alpha-Beta pruning, the AI delivers intelligent moves and ensures competitive gameplay. The combination of game development and AI implementation provides a practical demonstration of theoretical concepts in artificial intelligence.

## 8. References

- https://en.wikipedia.org/wiki/Checkers  
- https://www.geeksforgeeks.org/minimax-algorithm-in-game-theory-set-1-introduction/  
- https://www.pygame.org/docs/  
- https://github.com/Gualor/checkers-minimax