**Project Title:** 10x10 Checkers with AI using Minimax, Alpha-Beta Pruning  
**Submitted By:** Ateeb Azam, Muhammad Ali Zuberi, Huzaifa Bin Khalid  
**Course:** AI  
**Instructor:** Sir Abdullah Yaqoob  
**Submission Date:** 11th May 2025

**1. Executive Summary**

**Project Overview:**  
This project aimed to develop a strategic AI capable of playing an enhanced version of Checkers on a 10x10 board. The game increases complexity over traditional Checkers, requiring smarter and more efficient decision-making. The AI uses the Depth Minimax algorithm enhanced with Alpha-Beta Pruning for optimal move evaluation and supports depth-limited search to balance performance. The final product includes a playable GUI and optional Computer vs. Computer mode.

**2. Introduction**

**Background:**  
Traditional Checkers is an 8x8 board game involving diagonal movement and piece captures. Our project modifies the board to 10x10 to introduce greater tactical depth. This project was selected to explore AI performance in a complex, strategic, and turn-based environment.

**Objectives of the Project:**

* Develop a working Checkers game on a 10x10 board.
* Design and integrate a Minimax-based AI with Alpha-Beta Pruning.
* Enable efficient decision-making in a larger search space.

**3. Game Description**

**Original Game Rules:**  
Checkers is a two-player game played on an 8x8 grid. Players move pieces diagonally and capture opponents by jumping over them. A player wins by capturing all opponent pieces or blocking all their legal moves.

**Innovations and Modifications:**

* Increased board size from 8x8 to 10x10.
* New win condition: capture all opponent pieces or block all legal moves.
* Integration of AI using Minimax and Alpha-Beta Pruning.

**4. AI Approach and Methodology**

**AI Techniques Used:**

* Minimax Algorithm for strategic decision-making.
* Alpha-Beta Pruning for reducing the search space.
* Depth-limited search to manage computational load.

**Algorithm and Heuristic Design:**

* Heuristics prioritize number of pieces.
* Higher scores assigned to positional advantage.
* States evaluated using a custom scoring function.

**AI Performance Evaluation:**  
The AI achieved a win rate of 60% against human opponents. It demonstrated strong performance in blocking and capturing strategies.

**5. Game Mechanics and Rules**

**Modified Game Rules:**

* Game played on a 10x10 board.
* Movement and capture mechanics remain consistent with standard Checkers.

**Turn-based Mechanics:**

* Players take alternate turns.
* AI uses depth-limited Minimax with Alpha-Beta Pruning to decide moves.

**Winning Conditions:**

* A player wins by either capturing all opponent pieces or rendering them immobile.

**6. Implementation and Development**

**Development Process:**  
The development followed an 8-week schedule starting from basic board setup to AI logic integration and testing. The game logic was developed first, followed by GUI and AI integration.

**Programming Languages and Tools:**

* Programming Language: Python
* Libraries: Pygame, random
* Tools: None beyond standard Python setup

**Challenges Encountered:**  
The main challenge faced was implementing and debugging the AI logic using Minimax and Alpha-Beta Pruning due to the large search space of a 10x10 board.

**7. Team Contributions**

* **Muhammad Ali Zuberi:** Game and GUI implementation.
* **Huzaifa Bin Khalid:** Game and GUI implementation.
* **Ateeb Azam:** Designed and integrated AI logic into the game.

**8. Results and Discussion**

**AI Performance:**  
The AI demonstrated a 60% win rate in test runs against human players. Its decision-making balanced speed and efficiency due to depth-limited Minimax and Alpha-Beta Pruning. The heuristic scoring significantly improved move quality in mid-to-late game stages.

**9. References**

* YouTube tutorials on Minimax and Checkers AI development.