**Project Proposal: AI-Enhanced Checkers Game**

**Project Title:** AI-Enhanced Checkers Game  
**Submitted By:** Malaika(22k-5059) & Muhammad Asim( 22k-5066)  
**Course:** AI  
**Instructor:** [Ms. Ayesha Almas Ansari]  
**Submission Date:** [06-03-25]

### **1. Project Overview**

#### **Project Topic:**

This project aims to modify the traditional Checkers game by integrating an AI opponent. The existing game currently supports two human players. The innovation involves replacing one player with an AI bot that can make strategic decisions based on the game state.

#### **Objective:**

* Develop an AI opponent capable of making intelligent moves.
* Implement decision-making algorithms such as Minimax with Alpha-Beta pruning.
* Ensure smooth integration of AI into the existing game framework.
* Provide an engaging user experience by balancing difficulty levels.

### **2. Game Description**

#### **Original Game Background:**

Checkers is a classic two-player strategy game played on an 8×8 board where each player moves diagonally and captures opponent pieces by jumping over them. The goal is to eliminate all opponent pieces or block them from moving.

#### **Innovations Introduced:**

* One player is replaced by an AI bot that makes strategic moves based on the Minimax algorithm.
* AI difficulty balancing through heuristic evaluation of board positions.
* Optimized AI decision-making using Alpha-Beta pruning.
* Enhanced user experience by enabling AI-vs-human gameplay.



### **3. AI Approach and Methodology**

#### **AI Techniques to be Used:**

* **Minimax Algorithm** for decision-making.
* **Alpha-Beta Pruning** to optimize move calculations.
* **Heuristic Evaluation** for evaluating board states and move prioritization.

#### **Heuristic Design:**

The AI will evaluate board states based on:

* Number of remaining pieces.
* Positioning advantages (safe zones, mobility, and board control).
* Potential jumps and captures.

#### **Complexity Analysis:**

The AI will evaluate possible future moves in a tree structure. The time complexity will be approximately **O(b^d)**, where b is the branching factor and d is the search depth. Alpha-Beta pruning will optimize performance by reducing unnecessary evaluations.

### **4. Game Rules and Mechanics**

#### **Modified Rules:**

* The game will allow a human vs. AI mode.
* AI follows the same movement and capturing rules as a human player.
* AI moves will be automatically computed and displayed.

#### **Winning Conditions:**

* The game ends when one player (AI or human) eliminates all the opponent’s pieces or blocks them from making legal moves.

#### **Turn Sequence:**

* Players take alternate turns.
* The AI will compute and execute its move when it's its turn.

### **5. Implementation Plan**

#### **Programming Language:** C++

#### **Libraries and Tools:**

* Standard Template Library (STL) for data structures.
* GCC Compiler for execution.
* AI decision-making algorithms implemented from scratch.

#### **Milestones and Timeline:**

| **Task** | **Duration** |
| --- | --- |
| Code Analysis | 1 week |
| AI Algorithm Implementation | 2 weeks |
| Integration & Debugging | 2 weeks |
| Final Testing & Report Writing | 1 week |

### **6. References**

* Research papers on Minimax and Alpha-Beta pruning.
* Existing open-source Checkers implementations for reference.

**Prepared by:**  
Malaika & Asim