**Project Title:** Chinese Checkers

**Submitted By:** Yusra Asim (k22-4378), Lamia Asif (k22-4622)

**Course:** AI

**Instructor:** Abdullah Yaqoob

**Submission Date:** 10th March 2025

**1. Project Overview**

● **Project Topic:**

2 - 4 player Chinese Checkers with an extra rule for playing piece to block opponent's piece until next turn.

● **Objective:**

The objective of the game is to be the first player to move all of your pieces into the opposite triangular "goal" zone, occupying the exact positions of the opponent’s starting formation. Players must balance between advancing their own pieces and using the Blocking Piece to slow down or obstruct their opponents.

**2. Game Description**

● **Original Game Background:**

Chinese Checkers is a strategic, turn-based board game played on a hexagram (six-pointed star) board with up to six players. Each player controls a set of pieces of one color, starting from one of the triangular “home” zones on the board. The goal is to move all of your pieces across the board into the opposing triangle, following specific movement rules. The board consists of 121 spaces arranged in a grid, allowing pieces to move in six possible directions.

● **Innovations Introduced:**

Modified Rule: an additional piece that can not move itself and blocks opponent's moves until next turn (opponent's piece can not jump over blocking piece)

This will add a new strategic layer and allow players to choose between moving their own piece or blocking opponent

**3. AI Approach and Methodology**

● **AI Techniques to be Used:**

○ Minimax Algorithm to evaluate player's moves and plan optimal path

○ Alpha-Beta Pruning to optimize the Minimax search

○ Reinforcement Learning to train AI for optimal path finding

○ Other Techniques: nil

● **Heuristic Design:**

○ Evaluates game states by rewarding proximity to the goal, blocking opponent pieces, and mobility, while penalizing opponent progress and vulnerable positions. Will combine all of these factors into a single score to guide AI's decision making.

● **Complexity Analysis:**

○ O(b^d) for Minimax, and O(b^(d/2)) with Alpha Beta pruning where

b = branching factor (average number of possible moves per turn).

d = depth of the search tree (number of turns to look ahead).

**4. Game Rules and Mechanics**

● **Modified Rules:**

○ **Standard piece:**

- Move one space in any direction into an adjacent empty spot.

- OR jump over one adjacent piece (of any color) into an empty space directly on the other side.

- Multiple jumps are allowed in a single turn if possible.

○ **Blocking piece:**

- Each player has one Blocking Piece.

- It cannot move itself.

- It blocks the opponent’s pieces from moving through or jumping over its position.

- Opponent’s pieces cannot jump over the Blocking Piece

- applicable till one turn

● **Winning Conditions:**

○ A player wins by moving all their pieces into the opposite triangle on the

board.

● **Turn Sequence:**

○ Players can only move one piece per turn (no multiple jumps in one

turn).

○ Diagonal moves are allowed, but only if the adjacent space is

occupied.

○ Jumps can only occur over a single adjacent piece into an empty

Space.

○Jump can only occur if a blocking piece is not placed in the direct

path.

**5. Implementation Plan**

● **Programming Language:** Python

● **Libraries and Tools:**

○ Pygame

○ NumPy

○ TensorFlow

● **Milestones and Timeline:**

○ **Week 1-2:** Game design and rule finalization

○ **Week 3-4:** AI strategy development (Minimax and heuristics)

○ **Week 5-6:** Coding and testing the game mechanics

○ **Week 7:** AI integration and testing

○ **Week 8:** Final testing and report preparation

**6. References**

● [Research Paper - On Strongly Solving Chinese Checkers](https://webdocs.cs.ualberta.ca/~nathanst/papers/sturtevant2019chinesecheckers.pdf)

* [Research Paper - THE SHORTEST GAME OF CHINESE CHECKERS AND RELATED PROBLEMS](https://www.emis.de/journals/INTEGERS/papers/jg1/jg1.pdf)
* [Efficient Learning in Chinese Checkers: Comparing Parameter Sharing in Multi-Agent Reinforcement Learning](https://arxiv.org/pdf/2405.18733)