National University
of computer and emerging sciences

Artificial Intelligence

Proposal

"Mini-Checkers with Obstacles"

NATIONAL UNIVERSITY OF COMPUTER AND EMERGING SCIENCES

KARACHI CAMPUS

Group Members:

Muhammad Muzammil	**22K-4267**
Asim Majeed	**22K-4535**
Ayan Hasan	**22K-4367**

Instructor : Talha Shahid

1. Project Overview

Project Topic:
A mini‑checkers variant on a 6×6 board with random “obstacle” squares that block movement and captures, creating new choke‑point tactics between one human player and an AI opponent.

Objective:
To develop a strategic AI for this obstacle‑augmented mini‑checkers using a Minimax algorithm with alpha‑beta pruning and an obstacle‑aware heuristic, demonstrating how random barriers change optimal play.

2. Game Description

Original Game Background:
Standard checkers is played on an 8×8 board, each player with 12 men that move diagonally and capture by jumping. Men are promoted to kings when they reach the opponent’s back rank, gaining backward movement.

Innovations Introduced:

- **Board Size Reduction:** 6×6 grid, each player starts with 6 men on two rows.
- **Random Obstacles:** A variable number of squares are randomly marked impassable—no piece can move onto or jump over them. These squares are reassigned as the game progresses.
- **Impact on Strategy:** Obstacles create variable choke points, forcing rerouting and the need to account for the randomness of the obstacles.

3. AI Approach and Methodology

AI Techniques to be Used:

- **Minimax Algorithm:** Standard Minimax for two-player zero-sum play.
- **Alpha-Beta Pruning:** To cut off branches that won't influence the final decision, improving search depth.
- **Optional Reinforcement Learning:** Self-play to fine-tune heuristic weights (piece value, mobility around obstacles).

Heuristic Design:

- **Material Balance:** +1 per man, +2 per king.
- **Mobility:** Number of legal moves, heavily penalizing pieces trapped by obstacles.
- **Obstacle Proximity:** Count of paths to promotion that pass near obstacles; favor pieces with clearer routes.

Complexity Analysis:

- **Branching Factor:** ~4–6 (depending on available jumps)
- **Search Depth:** Target depth of 6 plies with α–β, keeping worst-case nodes around $O(6^3)$ after pruning.
- **Challenges:** Move generator must skip obstacle squares and recalculate jump sequences that might be interrupted by obstacles.

4. Game Rules and Mechanics

Modified Rules:

1. **Board Layout:** 6×6 dark‑square checkers board. Obstacles occupy a certain number of fixed dark squares randomly on the board, with their positions changing over time.
2. **Setup:** Each player places six men on their three nearest rows (dark squares only).
3. **Movement:**
   - Men move diagonally forward one square (never onto obstacles).
   - Captures are mandatory: jump diagonally over an adjacent enemy into the next square, unless that square is an obstacle or occupied.
4. **Promotion:** A man reaching the opponent's back rank becomes a king and may move/capture diagonally both forward and backward.
5. **Winning Conditions:** Capture all opponent pieces or leave them with no legal moves.
6. **Turn Sequence:** Players alternate turns, beginning with the human.

5. Implementation Plan

Programming Language: Python

Libraries and Tools:

- **Pygame:** GUI for board and piece rendering.
- **NumPy:** Board‑state representation and fast move computations.
- **Optional:** TensorFlow or PyTorch for reinforcement‑learning experiments.

Milestones and Timeline:

- **Week 1–2:**
  - Define board representation (6×6 array with obstacle flags).
  - Implement basic move and capture generation, enforcing obstacles.
- **Week 3–4:**
  - Build Minimax with alpha‑beta pruning.
  - Design and integrate the obstacle‑aware heuristic evaluation.
- **Week 5–6:**
  - Develop Pygame UI (render board, handle user input)
  - Test AI vs. human in console mode for correctness.

- **Week 7:** Connect AI engine to GUI; conduct playtesting and performance profiling.

- **Week 8:** Final debugging, parameter tuning, and prepare report with analysis of obstacle impacts.