Project Report: Checkers AI Game

1. Title Page

Project Title: Checkers AI Game
Submitted By: Uzair Ali (22k-4695)
Group Members: Muhammed Ahmed (22k-4725) Huzkail Waseem (22k-4678)

Course: Artificial Intelligence
Instructor: Miss Mehak Mazhar
Submission Date: 8 March 2025

2. Abstract

This project presents the design and implementation of an AI-enhanced Checkers game featuring three distinct gameplay modes: two-player (human vs. human), easy AI (random move generator), and hard AI (Minimax algorithm with heuristic evaluation). The primary aim is to showcase how classical AI techniques can be applied to a well-known board game, offering varying levels of challenge. We describe game mechanics, AI methodology, implementation details, and a planned development timeline.

3. Introduction

Checkers (also known as Draughts) is a strategic two-player board game played on an 8×8 grid where diagonal movement and capturing are key mechanics. Upon reaching the opponent's baseline, a piece is "kinged," gaining bidirectional movement. Our project enhances this classic game by introducing AI opponents of adjustable difficulty, thereby providing both educational insight into search algorithms and an engaging experience for players of all skill levels .

4. Game Description

4.1 Original Game Background

Board and Pieces: Standard 8×8 checkerboard with 12 pieces per player.

Movement: Diagonal moves to adjacent unoccupied squares; captures by jumping over an opponent’s piece into an empty square.

King Promotion: A regular piece becomes a king upon reaching the opponent’s last row and may then move diagonally forward and backward .

4.2 Innovations Introduced

1) AI Gameplay Modes:

Two-Player Mode (human vs. human)

Easy AI Mode (random move selection)

Hard AI Mode (Minimax algorithm with depth-3 search)

2)Visual Enhancements: King pieces are visually distinguished by a yellow ring.

Heuristic Evaluation: The AI’s evaluation function weights king pieces at 1.5× regular pieces to prioritize strategic king-making moves .

5. AI Approach and Methodology

5.1 AI Techniques

Random Move Generator: Utilizes Python's random library for the easy AI mode.

Minimax Algorithm: Depth-limited (d=3) search to explore possible move trees and select the optimal move in hard AI mode.

Heuristic Evaluation Function:

Computes (value of AI pieces) – (value of opponent pieces).

Regular piece = 1 point; king piece = 1.5 points.

Encourages moves that produce kings and capture opponent's kings .

5.2 Complexity Analysis

The branching factor bb in Checkers varies but averages around 7–10 possible moves per turn. With a depth d=3d = 3, the worst-case time complexity is approximately O(bd)≈O(103)=1,000O(b^d) \approx O(10^3) = 1{,}000 node evaluations per AI turn. Practical pruning via terminal state detection and simple move ordering reduces actual search effort.

6. Game Rules and Mechanics

6.1 Modified Rules

All standard Checkers rules apply.

King pieces gain backward movement upon promotion.

Hard AI mode biases toward king promotion through heuristic weighting.

6.2 Winning Conditions

Capture Victory: One player captures all opponent pieces.

Blockade Victory: One player blocks all legal moves of the opponent.

6.3 Turn Sequence

Player 1 moves (human or AI).

Player 2 moves (human or AI).

Repeat until a winning condition is met .

Programming Language & Tools

Language: Python

Libraries:

`pygame` for graphical interface and event handling

`random` for easy AI mode

Custom Minimax implementation for hard AI mode

---

8. References

Pinterest. “Checkers Game Rules.”
Retrieved from
https://www.pinterest.com/pin/345792077637457088/