



NATIONAL UNIVERSITY OF COMPUTER AND EMERGING SCIENCES

(KARACHI CAMPUS)

FAST School of Computing

Spring 2025

Project Title:

AI Based Checkers Game

Submitted By:

Maryam Hasan (22k-4701)

Muhammad Arham (22k-4707)

Muhammad Hassan Baig (22k-4696)

Course: AI

Instructor: Ms. Mehak Mazhar

Submission Date: 8 March 2025

Project Overview

Project Topic

The project is a Checkers Game with AI that utilizes the Minimax Algorithm to implement an AI opponent.

Objective

The objective of this project is to develop an AI-powered Checkers game where a player can compete against an AI that makes optimal moves using the Minimax algorithm.

Game Description

Original Game Background

The project is based on Checkers (Draughts), a two-player strategy board game played on an 8×8 grid. The game involves moving pieces diagonally across the board, capturing opponent pieces by jumping over them, and "kinging" pieces when they reach the opposite side.

Innovations Introduced

- **AI Opponent:** Uses the Minimax Algorithm to determine the best move for the AI player.
- **Graphical Representation:** The game is built using Pygame for interactive visuals.
- **Automatic Move Execution:** The AI plays automatically when it is its turn.
- **Move Highlighting:** The game visually indicates valid moves to the player.

AI Approach and Methodology

AI Techniques Used

- **Minimax Algorithm:** Used for game decision-making by simulating all possible moves.

Heuristic Design

- The evaluation function considers:
 - **Number of Pieces:** A simple scoring system where more pieces mean a better position.
 - **Kings' Advantage:** Kings (pieces that reached the opponent's side) are given extra weight in evaluation.

Complexity Analysis

- **Time Complexity:** The Minimax algorithm with depth d has a complexity of $O(b^d)$, where b is the branching factor (number of possible moves).

Game Rules and Mechanics

Modified Rules

- The game follows traditional Checkers rules.
- The AI automatically plays when it is the AI's turn.

Winning Conditions

- A player wins when the opponent has no remaining pieces.

Turn Sequence

- **Human Move:** Player clicks on a piece and selects a valid move.
- **AI Move:** The AI computes the best move using Minimax and executes it.

Implementation Plan

Programming Language

- Python

Libraries and Tools

- **Pygame** – For GUI rendering.
- **Deepcopy** – To clone board states for Minimax calculations.

Milestones and Timeline

Week	Task
1-2	Game design, board setup
3-4	Implement Minimax algorithm
5-6	Integrate AI with the game
7	Testing and improvements
8	Final testing and report preparation

References https://youtu.be/ipExjmyd6cc?si=mgA_e1k1l5YYeQ-c