AI Lab Project Report

# Project Title:

AI-Powered Connect Four with Minimax Algorithm

# Submitted By:

M. Soubanullah (22k-5002)  
M. Ismail Memon (22k-5005)  
Asadullah Vohra (22k-5121)

# Course:

Artificial Intelligence

# Instructor:

Ms. Almas Ayesha Ansari

# Submission Date:

25-03-2025

# 1. Introduction

This project implements the classic Connect Four game integrated with an AI opponent. The AI uses the Minimax algorithm, enhanced with Alpha-Beta pruning, to make optimal decisions and offer varying levels of challenge. The aim is to provide an interactive, intelligent gameplay experience where users can play against a capable virtual opponent.

# 2. Objectives

* Develop a fully functional Connect Four game using Python and Pygame.
* Integrate a decision-making AI using the Minimax algorithm.
* Optimize performance using Alpha-Beta pruning.
* Design a heuristic evaluation function to evaluate the board states.
* Implement three difficulty levels: Easy, Medium, and Hard, based on search depth.

# 3. Game Description

## Game Rules

* The board consists of 6 rows and 7 columns.
* Players alternate turns dropping colored discs into columns.
* The first player to connect four of their discs in a row—horizontally, vertically, or diagonally—wins.

## Innovative Features

* AI Opponent:
* Uses Minimax algorithm.
* Alpha-Beta pruning reduces computation time.
* Heuristic function evaluates board positions.
* Difficulty Modes:
* Easy: Random valid moves.
* Medium: Minimax with search depth = 3.
* Hard: Minimax with Alpha-Beta pruning and depth = 5.

# 4. AI Methodology

## Algorithms Used

|  |  |
| --- | --- |
| Technique | Purpose |
| Minimax | Evaluate all possible game states to maximize AI’s advantage |
| Alpha-Beta Pruning | Eliminate branches that won’t influence the final decision |
| Heuristic Evaluation | Score board positions based on strategic criteria |

## Heuristic Evaluation Criteria

* Immediate Win/Loss: Assign +∞ if AI wins, -∞ if human wins.
* Three-in-a-Row Potential: Assign high score if AI or opponent is close to a win.
* Center Column Control: Preference for central columns due to strategic value.

## Complexity

Without Pruning: O(b^d), where b is branching factor, d is search depth.

With Alpha-Beta Pruning: Effective reduction by ~50%.

# 5. Implementation Plan

## Tech Stack

Language: Python

Libraries:

* pygame (GUI rendering)
* numpy (board representation and computation)

## Development Timeline

|  |  |
| --- | --- |
| Week | Tasks |
| 1 | Implement game board and core mechanics |
| 2 | Integrate basic Minimax algorithm |
| 3 | Add Alpha-Beta pruning and heuristic evaluation |
| 4 | Final testing and GUI enhancements |

# 6. Future Work

* Integrate full Minimax implementation with Alpha-Beta pruning.
* Finalize heuristic tuning for optimal AI play.
* Add and test difficulty levels.
* Improve UI/UX with animations and indicators.