Connect Four Game Using Minimax Algorithm

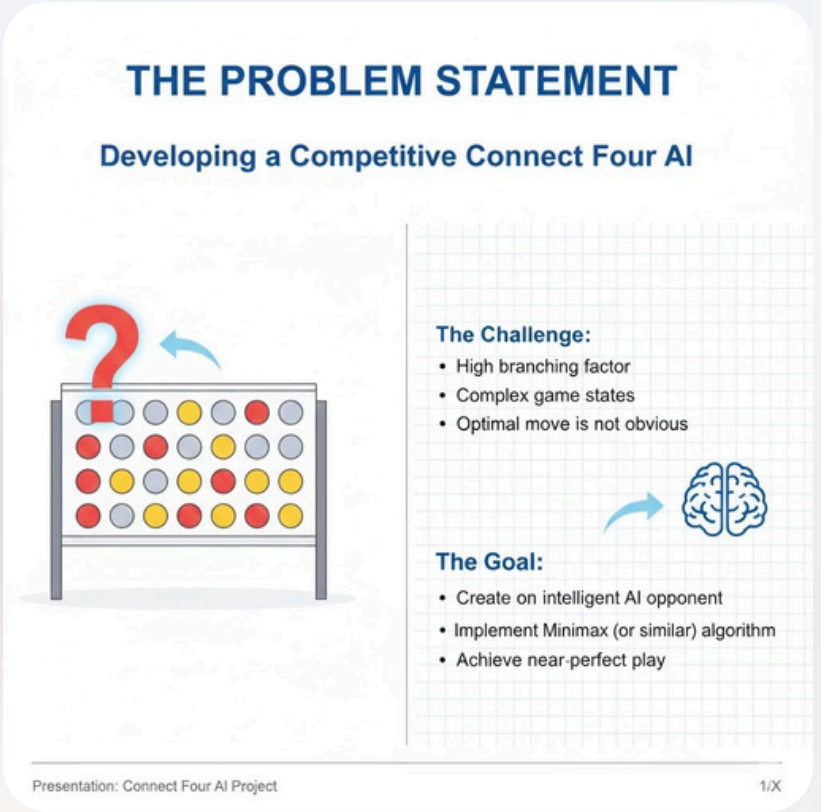
Presented By:

Mayank Kumar Jain (1BF24CS170)

Mayank Verma (1BF24CS172)

Mohammed Nawaalur Rahaman (1BF24CS177)

Muhammed Razan (1BF24CS182)



Problem Statement

Game Challenge

Connect Four requires strategic reasoning and prediction of opponent moves, but manual gameplay lacks intelligent automation and optimal decision-making capabilities.

Design Objective

Create a Java-based Connect Four game where human players compete against an AI opponent using the Minimax algorithm for optimal move selection.

System Requirements

The system mustenforce game rules correctly, evaluate game states efficiently, and detect win, loss, and draw conditions with complete accuracy.

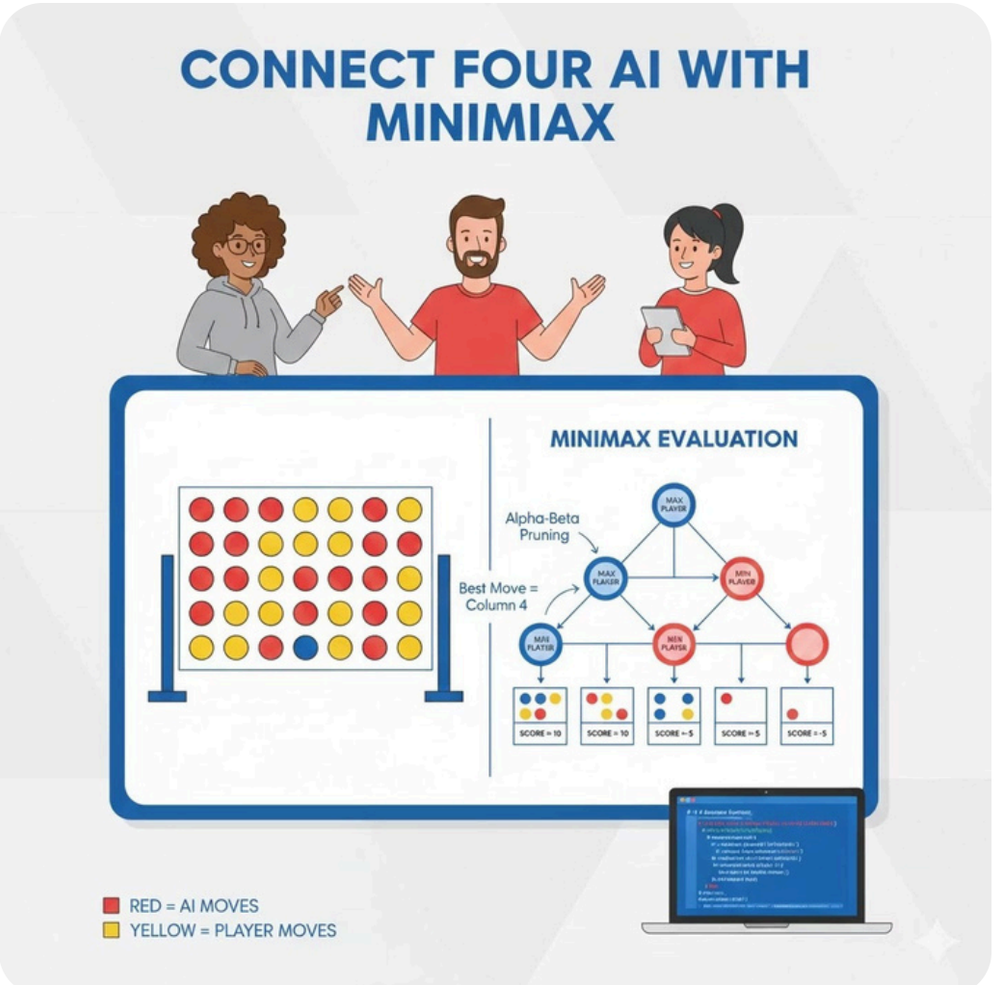
Introduction to Connect Four & AI Integration

Game Fundamentals

Connect Four utilizes a 7-column × 6-row grid where players alternately drop colored discs. The objective is to connect four discs consecutively in horizontal, vertical, or diagonal arrangements.

Artificial Intelligence Role

AI transforms gaming experiencesby enabling intelligent decision-making in turn-based strategy games, providing challenging and adaptive opponents.



This project demonstrates the powerful integration of AI concepts with Object-Oriented Programming principles in Java, creating an intelligent game system that challenges human strategic thinking.

Project Architecture Overview

Player Input

Human interaction through intuitive UI

Game Logic

Rule enforcement and state management

Minimax Algorithm

AI decision-making engine

AI Move

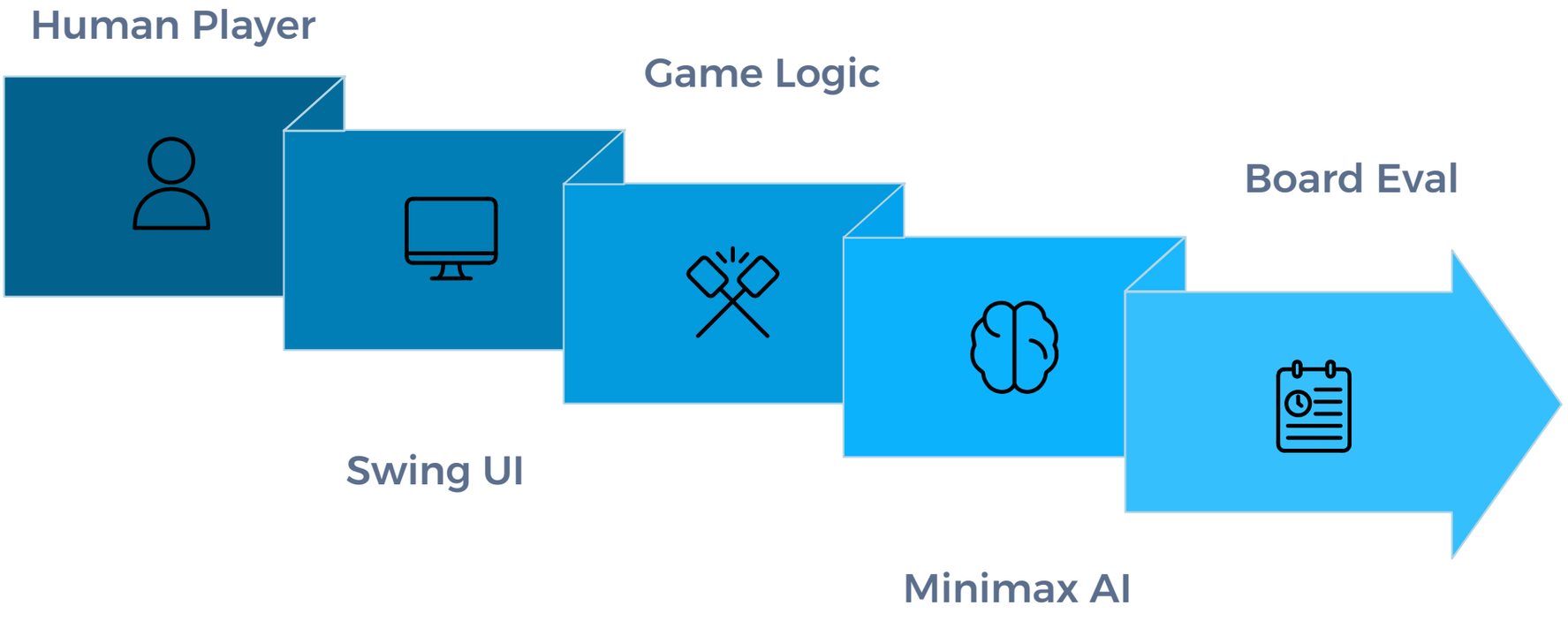
Optimal move execution

Result Check

Win/loss/draw evaluation

The project architecture emphasizes **separation of concerns** using OOPS principles, dividing functionality into logical modules: User Interface, Game Logic, AI Decision Making, and Result Evaluation.

System Block Diagram



The system architecture demonstrates clear data flow from user input through game logic processing, AI evaluation, and final result determination, ensuring modular and maintainable code structure.

Tools & Technologies Used

Programming Language

Java

Robust, platform-independent language with strong OOPS support

Development Paradigm

Object-OrientedProgramming

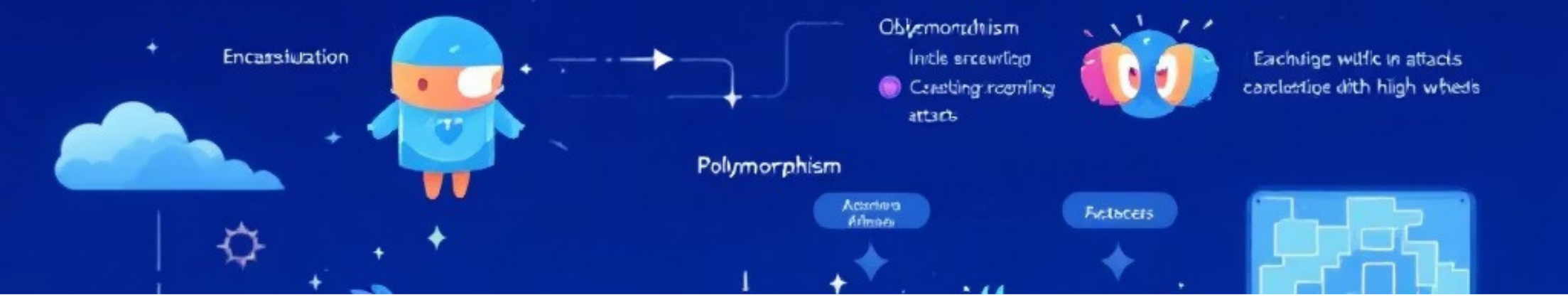
Encapsulation, inheritance, polymorphism, and abstraction

AI Algorithm

Minimax Algorithm

Recursive decision-making for optimal game strategy

- **IDE:** Visual Studio Code
- **JDK Version:** JDK 8 or above
- **UI Framework:** Java Swing
- **Operating System:** Windows / Linux



OOPS Concepts Implementation

Encapsulation

Game State class encapsulates board data, moves, and game status with controlled access through methods like move(), undo(), and getters, ensuring data integrity.

Inheritance

Board Drawing extends JComponent, utilizing Swing class hierarchy (JFrame, JPanel) for building sophisticated user interfaces with inherited functionality.

Polymorphism

Method overriding including paintComponent(Graphics g) and event handling using overridden listener methods for flexible behavior implementation.

Abstraction

UIlayer separated from game logic, allowing UI to interact with GameState without knowing internal implementation details, promoting modularity.

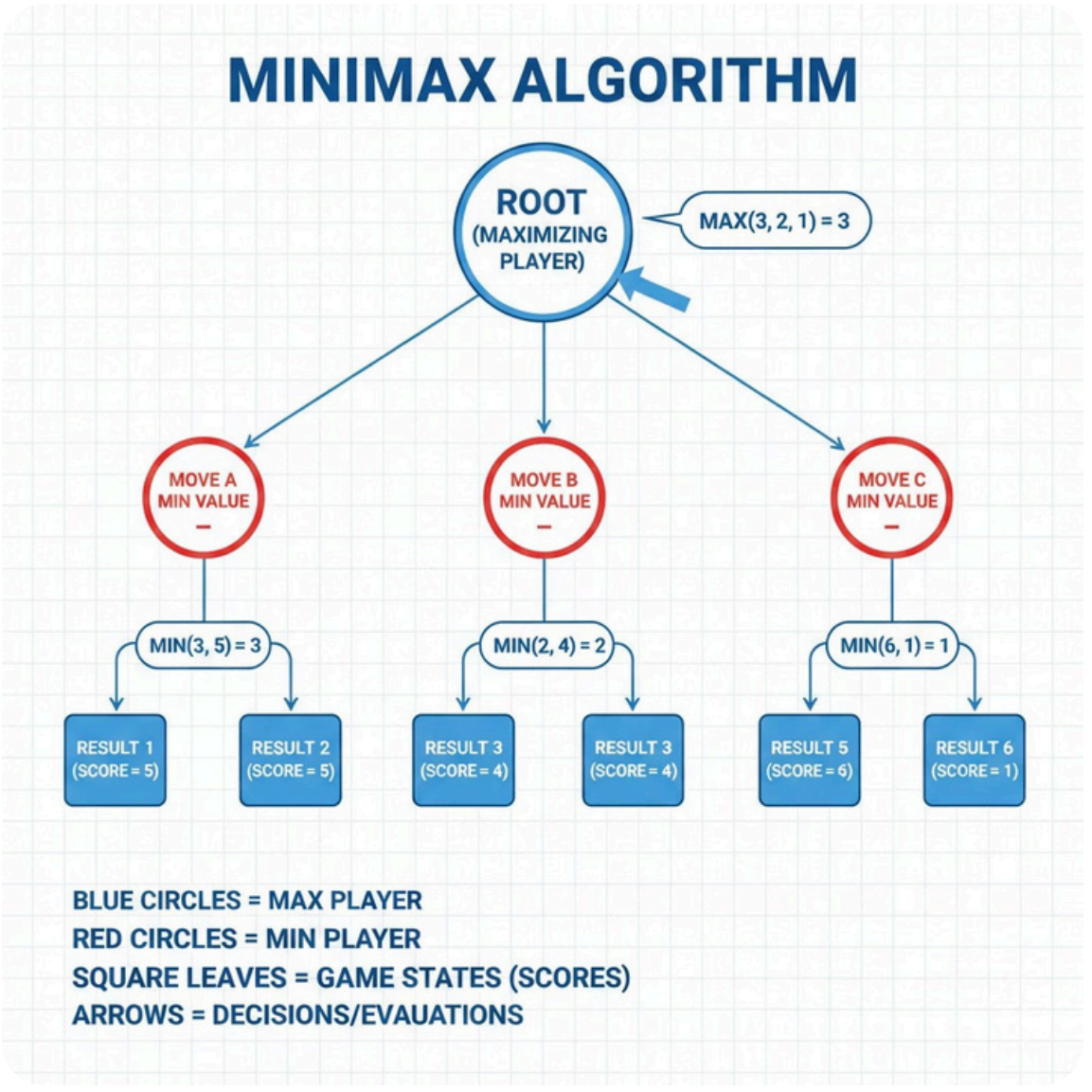
Minimax Algorithm: Core AI Engine

Algorithm Principle

Minimax is a recursive decision-making algorithm for turn-based games, assuming both players play optimally to maximize their advantage.

Player Roles

- **Maximizer:** AI player attempts to maximize score
- **Minimizer:** Human player minimizes AI advantage



01

Generate all valid moves

02

Simulate each move recursively

03

Evaluate opponent responses

04

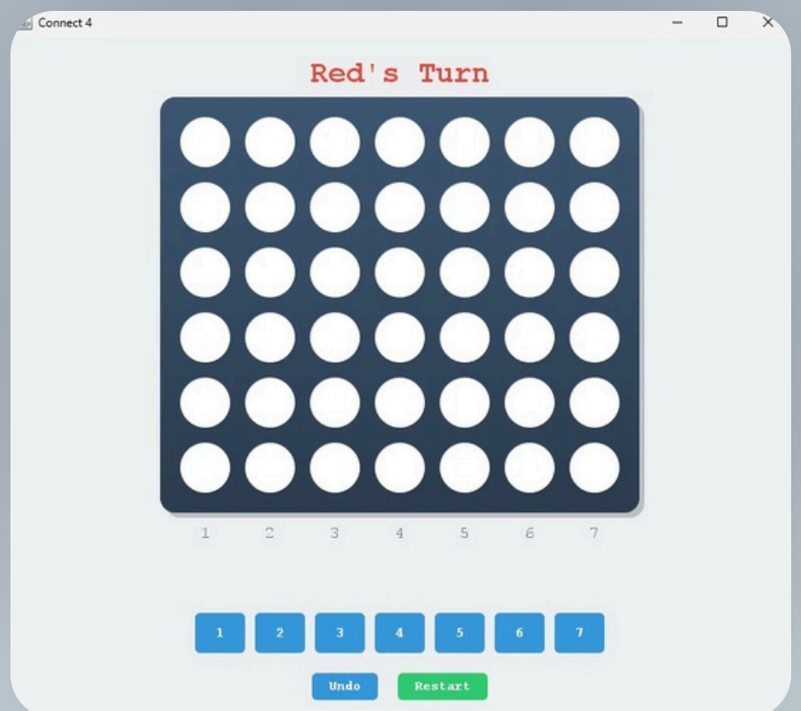
Identify terminal game states

05

Assign evaluation scores

06

Select optimal move



Board Evaluation & System Integration

1

Evaluation Scores

Winning state receives highpositivescore,losingstate receives high negative score, and draw state receives zero score for balanced decision-making.

2

Depth Limitation

Recursion depthis limited to improve computational performance while maintaining strategic depth and decision quality.

3

AI Integration

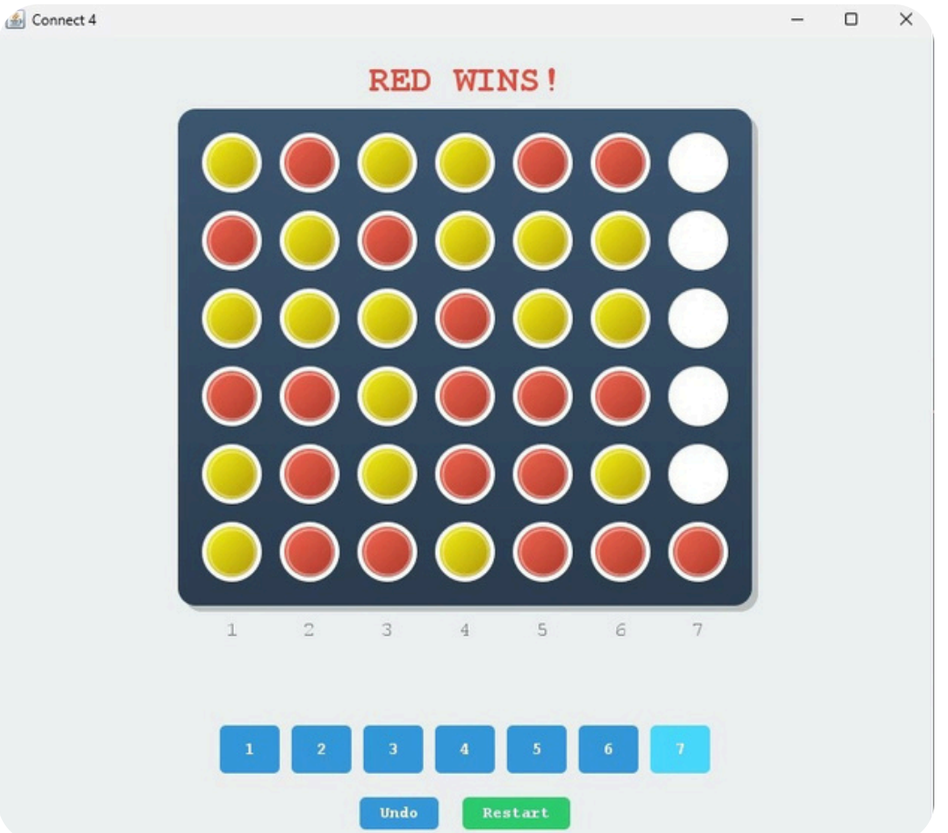
Minimaxis invokedduring AI turn, analyzing possible moves and selecting the best column based on comprehensive evaluation scores.

The evaluation function considers multiple factors including potential winning patterns, blocking opponent strategies, and controlling center positions for maximum strategic advantage.

Results & Conclusion

Project Results

- Gameexecuteswithout runtime errors
- AI effectively blocks opponent strategies
- Correct detection of horizontal wins
- Accurate vertical win identification
- Precise diagonal win recognition
- Proper draw condition handling



Technical Achievement

Successfully integrated artificial intelligence with Object-Oriented Programming principles in Java, creating a robust and intelligent gaming system.

Practical Application

Demonstrates practical implementation of Minimax algorithm in real-world gaming scenarios with optimal decision-making capabilities.

Learning Outcomes

Project significantly enhances understanding of game AI, software design patterns, and the integration of algorithmic thinking with OOPS concepts.

THANK YOU