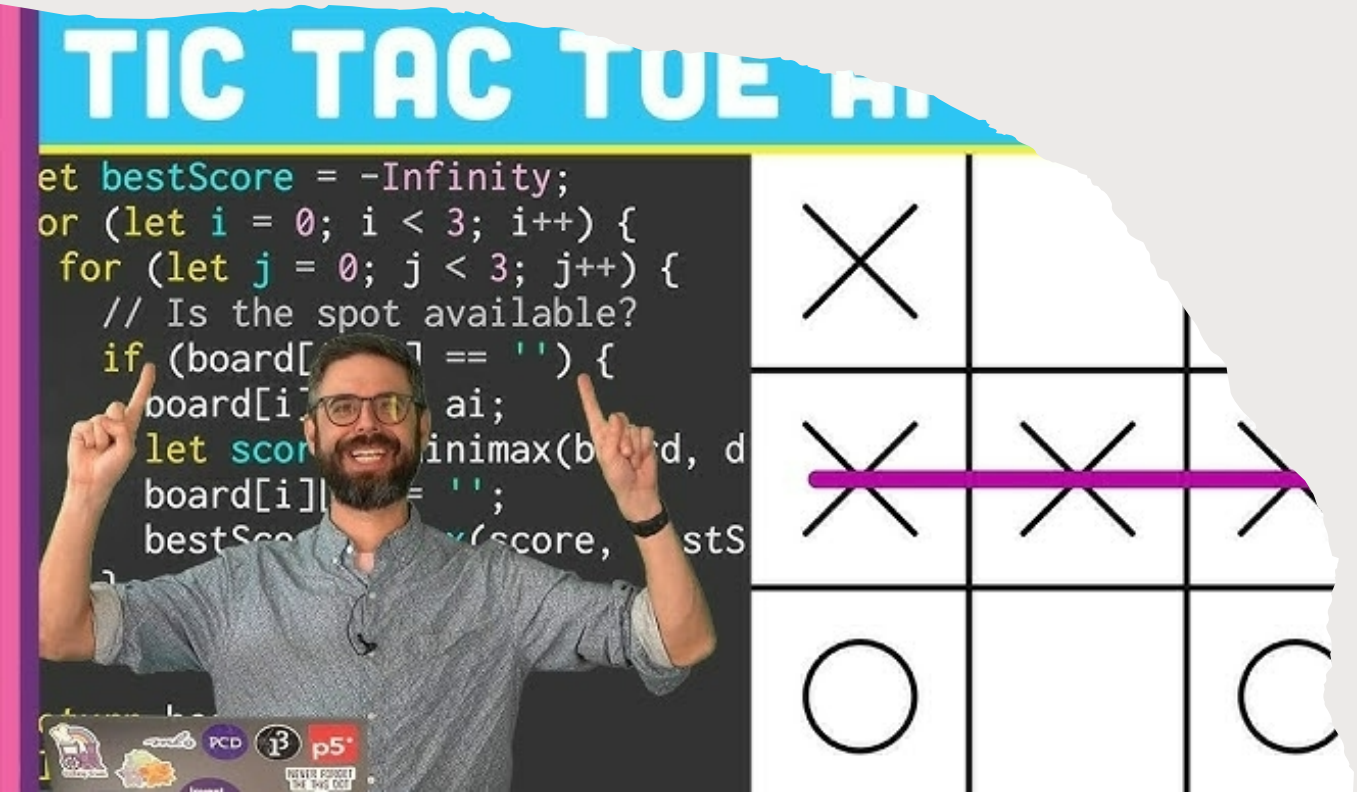
Implementing and Comparing Adversarial Search and Reinforcement Learning in Tic Tac Toe

Sarika Burugu - 00876552

Chinthakuntla Teja Reddy - 00868032

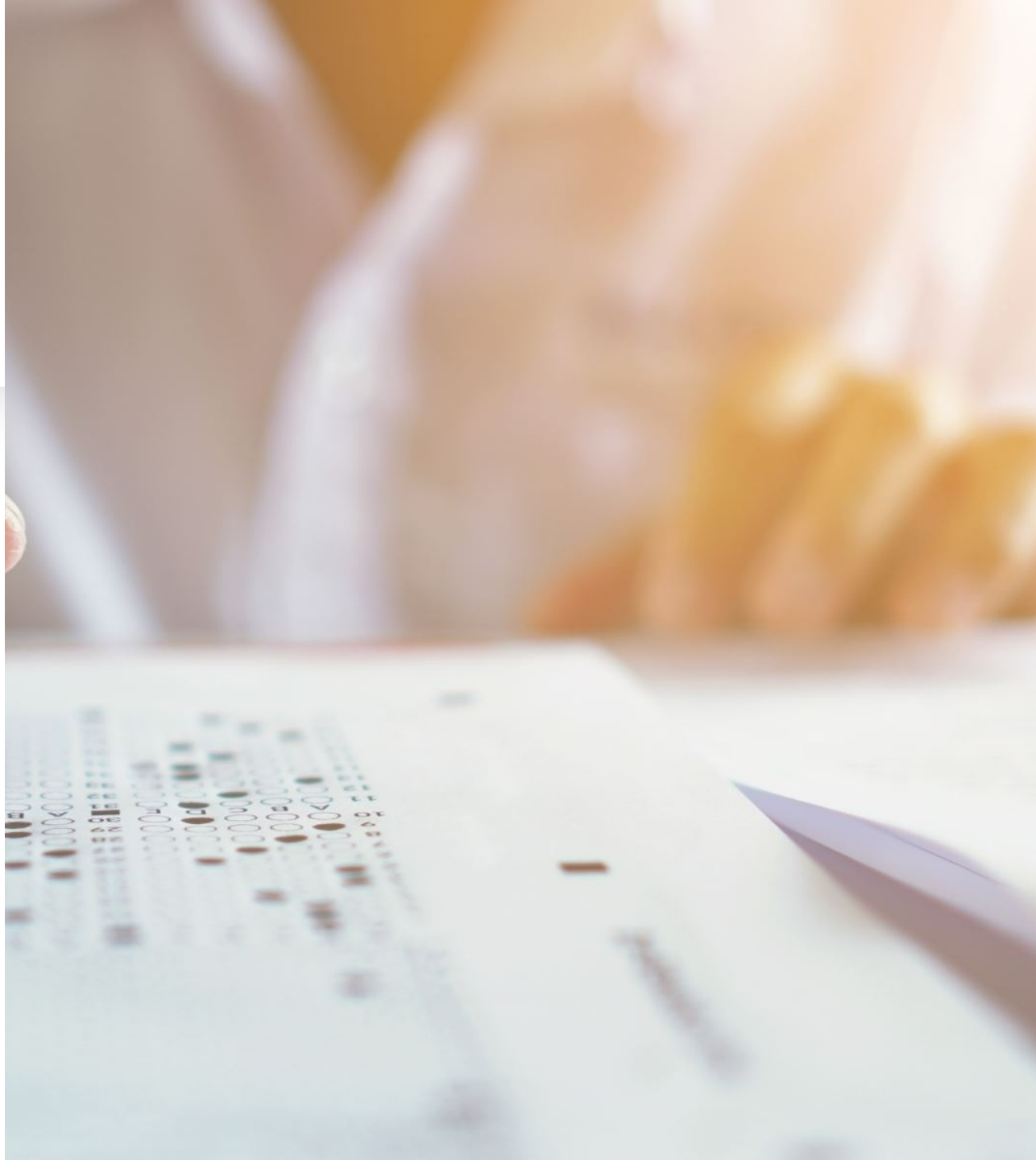
Project Objectives



- To develop adversarial search algorithms and reinforcement learning agents to solve the Tic Tac Toe game.
- To implement three types of Artificial Intelligence agents:
- 1. Q-learning (from Reinforcement Learning)
- 2. Min-Max Algorithm (from Adversarial Search)
- 3. Alpha-Beta Pruning (from Adversarial Search)
- To test these agents by having them play multiple rounds of Tic Tac Toe against each other and identify the most efficient algorithm.

Deliverables

- **User Documentation:** A model that explains the Tic Tac Toe game implementation using the three AI agents (Min-Max, Alpha-Beta, and Reinforcement Learning).
- **Codebase:** Python (.py) files for the AI agents.
- **GitHub Repository:** A link to the repository containing the code and related files.
- **Demonstration Video:** A YouTube video showcasing the implementation along with slides.



Approach

- **Reinforcement Learning Approach:** Q-Learning
- This is our first approach, where the agent learns the optimal strategy through interactions with the game environment.

- **Adversarial Search Approaches:**

  Min-Max Algorithm: A decision-making strategy for two-player zero-sum games that considers all possible moves.

  Alpha-Beta Pruning: An optimization technique that reduces the number of nodes evaluated in the Min-Max algorithm.

- **Technology Stack:**
- Programming Language: Python 3

Evaluation Methodology

- The project will be assessed based on the successful implementation of the AI agents.
- The performance of each agent will be evaluated by counting the number of wins each agent achieves when playing against each other.
- A comparison table will be generated, showing the moves and corresponding scores for each agent, along with their win rates.
- Graphs will be used to visually represent the performance of each agent.
- Time and space complexity of each algorithm will be compared to assess their efficiency.