TEAM CHARGERS
Term Project – Mid Status Report

1. **Research Question**: I hypothesize that reinforcement learning, Q-Learning agent can improve the performance of Tic-Tac-Toe.

2. **Introduction**: Tic-Tac-Toe is a game played on a 3x3 grid, with two players taking turns placing their marks (either “X” or “O”) on the board. The game ends when one player has three marks in a row (horizontally, Vertically, or Diagonally) or when the board is full and there is no winner (a tie game).

   The primary objective of this project is to develop an AI agent capable of playing Tic-Tac-Toe optimally using two approaches: Q-Learning and Minimax with Alpha-Beta pruning. The project aims to test the effectiveness of these techniques, measure their performance, and explore the trade-offs between them, with a focus on strategy, exploration, and decision-making.

3. **Related Literature:**

   - **Q-Learning for Tic-Tac-Toe:** In "An Efficient Q-learning Strategy for Tic-Tac-Toe" (https://www.researchgate.net/publication/369096697_Reinforcement_Learning_Playing_Tic-Tac-Toe), Q-Learning was applied to learn optimal strategies for Tic-Tac-Toe. Like our approach, the document discusses the use of Q-tables for state-action evaluation. However, our implementation includes advanced reward tracking and performance metrics.

   - **Minimax and Alpha-Beta Pruning:** The classical approach of Minimax, as explored in "Alpha-Beta Pruning in Minimax Algorithm for Tic-Tac-Toe" (https://www.researchgate.net/publication/358907355_Tic_Tac_Toe_by_Minimax_Alpha-Beta_Pruning_Using_Arduino?_sg=Z4d5yPt9kUJG8szOBcFxF6k6Sh9xeH88gp5kEqX0zMkNWqlyHfotSZbZ8EvOU5KQP7O3DwfcXDzNw_M&_tp=eyJjb250ZXh0Ijp7ImZpcnN0UGFnZSI6InB1YmxpY2F0aW9uRGlyZWN0b3J5IiwicGFnZSI6Il9kaXJlY3QifX0), is used to optimize decision-making by pruning unnecessary branches in the decision tree. While the original paper uses a fixed depth, our solution introduces dynamic depth control and a choice between Minimax and Alpha-Beta pruning based on user input.

   - **Comparison with Other Approaches:** The work "A Comparison of Classical AI Techniques for Tic-Tac-Toe" (https://arxiv.org/pdf/1708.06799.pdf) compares the performance of various AI algorithms for Tic-Tac-Toe. Our research work extends this by integrating reinforcement learning and comparing the performance of two distinct AI models (Q-Learning and Minimax/Alpha-Beta).

4. **Approach:**
We have implemented two AI techniques for playing Tic-Tac-Toe:

- **Q-Learning Agent:** The agent learns optimal strategies by updating Q-values based on the rewards received during gameplay. A balance between exploration and exploitation is maintained using an epsilon-greedy approach.
- **Minimax with Alpha-Beta Pruning:** The agent evaluates possible future moves and selects the best move by simulating game outcomes through recursion. Alpha-Beta pruning optimizes this process by eliminating branches of the game tree that do not affect the final decision.

5. **Achievements & Challenges:**
Achievements:

- The Q-Learning agent has been trained successfully against a random agent, and its performance metrics (Total mistakes per turn, Average accumulated rewards, win rate, and draw rate) have been tracked. We were able to achieve this point.
- The Minimax with Alpha-Beta pruning agent successfully selects optimal moves and performs faster due to pruning.

Challenges & Workarounds:

- **Q-Table Representation:** Initially, we faced challenges with representing the game states in the Q-table. The states were stored as strings, which caused issues during updates. We then resolved this by converting the state representation to tuples, ensuring compatibility with NumPy operations.
- **Agent Exploration vs. Exploitation:** Balancing exploration and exploitation in the Q-Learning agent was tricky. We then adjusted the epsilon value dynamically based on the agent's performance, ensuring that exploration decreased as the agent learned.
- **Minimax Depth Limitation:** Minimax, with its recursive nature, was causing performance bottlenecks. To solve this, we implemented a depth limit and dynamic pruning strategies.

6. **GitHub Repository Link:**

- **https://github.com/sureshnalajala/Tic-Tac-Toe.git**