Tic-Tac-Toe with Alpha-Beta Pruning

1. Title

Project Title: Tic-Tac-Toe

Name = Vikrant Baliyan

Roll no. = 202401100400211

2. Introduction

Tic-Tac-Toe is a classic two-player game where players take turns marking spaces on a 3x3 grid. This project implements an AI opponent using the Minimax algorithm with Alpha-Beta Pruning. The AI is designed to make optimal moves, ensuring a challenging experience for the human player. This report documents the methodology, implementation, and results of the project.

3. Methodology

Algorithm Used: Minimax with Alpha-Beta Pruning

The Minimax algorithm evaluates all possible moves and chooses the best outcome for the AI.

Alpha-Beta Pruning optimizes the Minimax algorithm by eliminating unnecessary branches, improving efficiency.

The game alternates turns between the AI ('X') and the human player ('O'). Game Flow:

The AI ('X') makes the first move.

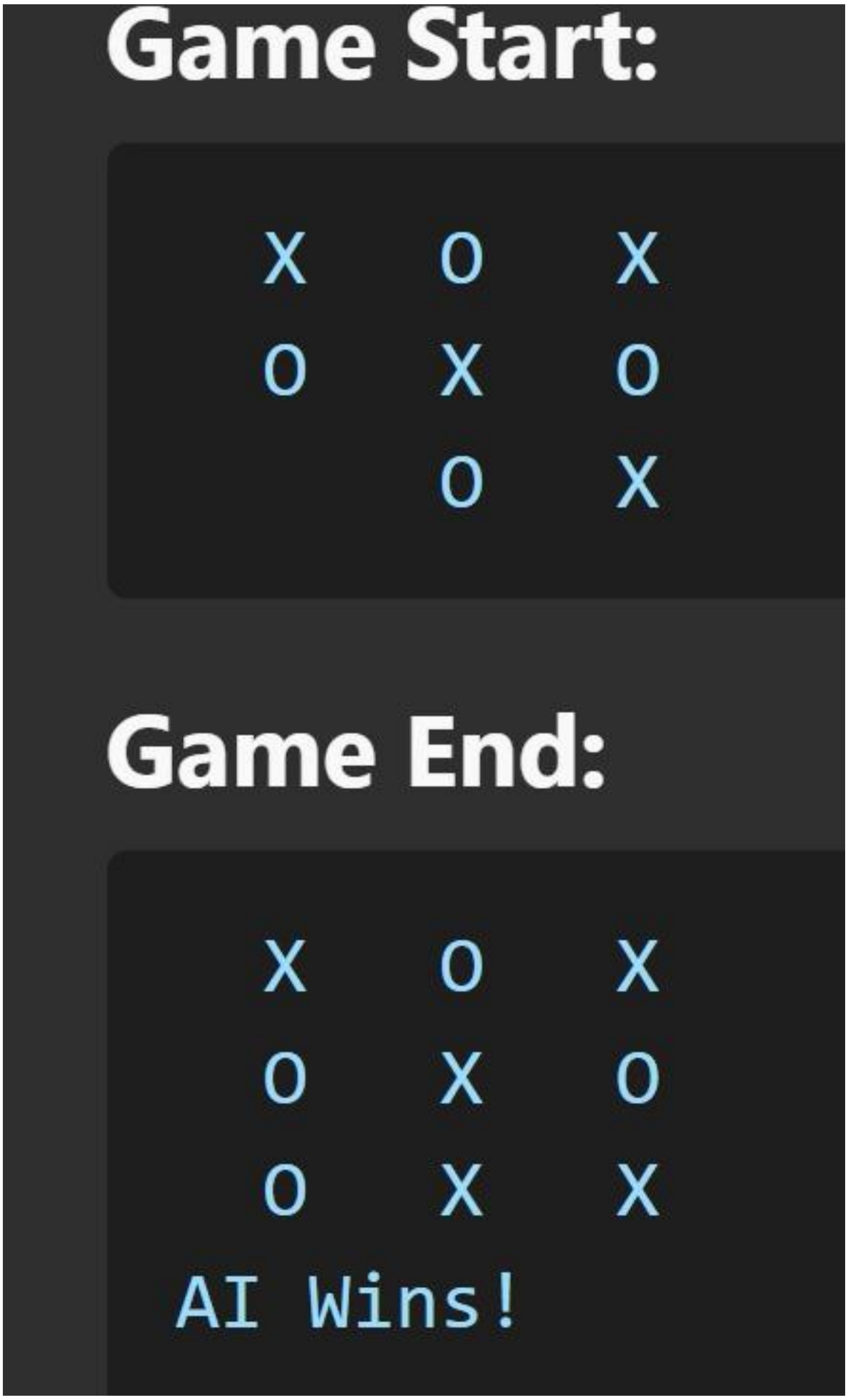
The player ('O') enters their move by specifying row and column indices (0-2).

The board updates and displays after each move.

The game continues until a winner is found or the board is full.

The result is displayed at the end (Win, Lose, or Draw). import math

```
# Function to print the Tic-Tac-Toe board
def print_board(b):
    for row in b:
        print(" ".join(row))
    print()

# Function to evaluate the board state
def evaluate(b):
    for r in b + list(zip(*b)) + [[b[i][i] for i in range(3)], [b[i][2-i] for i in range(3)]]:
        if r[0] == r[1] == r[2] != ' ':
            return 10 if r[0] == 'X' else -10
    return 0

# Minimax function with Alpha-Beta Pruning
def minimax(b, depth, is_max, alpha, beta):
    score = evaluate(b)
    if score != 0 or not any(' ' in row for row in b):
        return score
    if is_max:
        best = -math.inf
        for i in range(3):
            for j in range(3):
                if b[i][j] == ' ':
                    b[i][j] = 'X'
                    best = max(best, minimax(b, depth + 1, False, alpha, beta))
                    b[i][j] = ' '
                    alpha = max(alpha, best)
                    if beta <= alpha:
                        break
        return best
    else:
        best = math.inf
        for i in range(3):
            for j in range(3):
                if b[i][j] == ' ':
                    b[i][j] = 'O'
                    best = min(best, minimax(b, depth + 1, True, alpha, beta))
                    b[i][j] = ' '
                    beta = min(beta, best)
                    if beta <= alpha:
                        break
        return best

# Function to find the best move for AI
def best_move(b):
    best_val = -math.inf
    move = (-1, -1)
    for i in range(3):
        for j in range(3):
            if b[i][j] == ' ':
                b[i][j] = 'X'
                move_val = minimax(b, 0, False, -math.inf, math.inf)
                b[i][j] = ' '
                if move_val > best_val:
                    move = (i, j)
                    best_val = move_val
    return move

# Main function to play the game
def play():
    board = [[' ']*3 for _ in range(3)]
    for turn in range(9):
        print_board(board)
        if evaluate(board):
            break

        if turn % 2 == 0:
            print("AI's Turn (X)")
            x, y = best_move(board)
        else:
            while True:
                try:
                    x, y = map(int, input("Your Turn (O). Enter row and column (0-2): ").split())
                    if board[x][y] == ' ':
                        break
                    print("Invalid move. Try again.")
                except (ValueError, IndexError):
                    print("Invalid input. Enter two numbers between 0 and 2.")

        board[x][y] = 'X' if turn % 2 == 0 else 'O'

    print_board(board)
    result = evaluate(board)
    if result == 10:
        print("AI Wins!")
    elif result == -10:
        print("You Win!")
    else:
        print("It's a Draw!")

if __name__ == "__main__":
    play()
```



Game Start:

```
X   O   X
O   X   O
    O   X
```

Game End:

```
  X   O   X
  O   X   O
  O   X   X
AI Wins!
```