Tic Tac Toe

ARTIFICIAL INTELLIGENCE PROJECT

Introduction

ABOUT THE GAME

- Tic-tac-toe is a classic paper-and-pencil game played on a grid of 3x3 squares.

- The game is also known as "noughts and crosses" in some parts of the world.

- The goal of tic-tac-toe is to get three of your own symbols in a row, either horizontally, vertically, or diagonally, before your opponent does.

Rules

- The game is played on a grid of 3x3 squares.

- Two players participate, usually referred to as "X" and "O." Player X typically goes first.

- Players take turns placing their symbol (X or O) in an empty square on the grid.

- The game continues until one player achieves three of their symbols in a row (horizontally, vertically, or diagonally) or there are no more empty squares.

- If a player successfully gets three of their symbols in a row, they win the game.

- If all squares are filled and no player has won, the game is a draw (also known as a tie or a stalemate).

```
board = [' ' for _ in range(9)]
```
To Create the 3 X 3 square Tic Tac Toe board

```
winning_combinations = [
    [0, 1, 2], [3, 4, 5], [6, 7, 8],  [0, 3, 6], [1, 4, 7], [2, 5, 8],   [0, 4, 8], [2, 4, 6] ]
```
To Define the winning combinations

```
def print_board():
```
 Function to print the board

```
def is_board_full():
```
 Function to check if the board is full

```
def has_player_won(player):
```
 Function to check if the player has won

```
play_game()
```
Start the game

Code Explanation

```python
def make_player_move():
    Function to make the player's move

def make_ai_move():
    Function to make the AI's move

    for move in available_moves:
        To Check if AI can win in the next move

    for move in available_moves:
        To Check if player can win in the next move

    ai_move = random.choice(available_moves)
    Choose a random move

def play_game():
    Main game loop
```
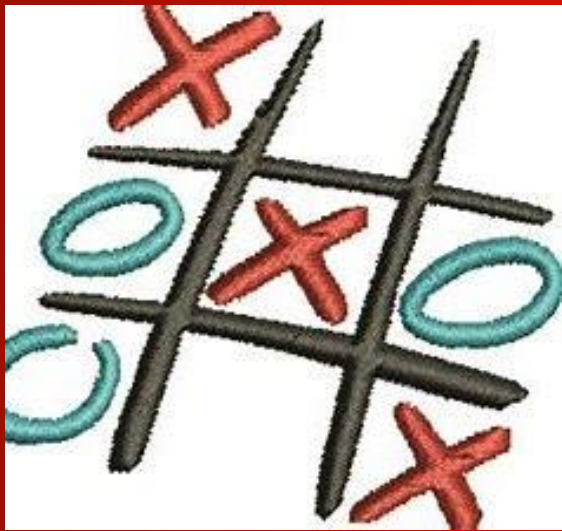
Code Explanation

In this code, the player plays as 'X', and the AI opponent plays as 'O'. The player and AI take turns making moves until one of them wins or the game ends in a tie. The AI opponent uses a simple strategy to prioritize winning moves and blocking the player's winning moves whenever possible.

```
IDLE Shell 3.11.3                                                    —  □  ✕

File  Edit  Shell  Debug  Options  Window  Help

Python 3.11.3 (tags/v3.11.3:f3909b8, Apr  4 2023, 23:49:59) [MSC v.1934 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
= RESTART: C:\Users\Krithinka.M\AppData\Local\Programs\Python\Python311\game.py
Welcome to Tic Tac Toe!
---------
|   |   |   |
---------
|   |   |   |
---------
|   |   |   |
---------
Enter your move (0-8): 0
---------
| X |   |   |
---------
|   |   |   |
---------
|   |   |   |
---------
| X |   |   |
---------
|   |   |   |
---------
| O |   |   |
---------
Enter your move (0-8): 1
---------
| X | X |   |
---------
|   |   |   |
---------
| O |   |   |
---------
| X | X | O |
---------
|   |   |   |
---------
| O |   |   |
```
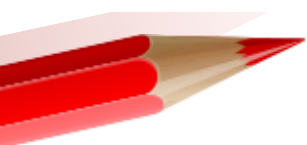
```
|   |   |   |
---------
---------
| X |   |   |
---------
|   |   |   |
---------
| O |   |   |
---------
Enter your move (0-8): 1
---------
| X | X |   |
---------
|   |   |   |
---------
| O |   |   |
---------
---------
| X | X | O |
---------
|   |   |   |
---------
| O |   |   |
---------
Enter your move (0-8): 7
---------
| X | X | O |
---------
|   |   |   |
---------
| O | X |   |
---------
---------
| X | X | O |
---------
|   | O |   |
---------
| O | X |   |
---------
You lost! Better luck next time.
>>>
```

```python
import random

# Create the Tic Tac Toe board
board = [' ' for _ in range(9)]

# Define the winning combinations
winning_combinations = [
    [0, 1, 2], [3, 4, 5], [6, 7, 8],   # Rows
    [0, 3, 6], [1, 4, 7], [2, 5, 8],   # Columns
    [0, 4, 8], [2, 4, 6]               # Diagonals
]

# Function to print the board
def print_board():
    print('---------')
    for i in range(0, 9, 3):
        print(f'| {board[i]} | {board[i+1]} | {board[i+2]} |')
        print('---------')

# Function to check if the board is full
def is_board_full():
    return ' ' not in board

# Function to check if the player has won
def has_player_won(player):
    for combo in winning_combinations:
        if all(board[i] == player for i in combo):
            return True
    return False

# Function to make the player's move
def make_player_move():
    while True:
        move = int(input("Enter your move (0-8): "))
        if move >= 0 and move < 9 and board[move] == ' ':
            board[move] = 'X'
            break
        else:
            print("Invalid move. Try again.")

# Function to make the AI's move
```

Project code

File Edit Shell Debug Options Window Help

```
Python 3.11.4 (tags/v3.11.4:d2340ef, Jun  7 2023, 05:45:37) [MSC v.1934 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.

= RESTART: C:/Users/Krithinka.M/Downloads/AI_PROJECT.py
Welcome to Tic Tac Toe!
---------
|   |   |   |
---------
|   |   |   |
---------
|   |   |   |
---------
Enter your move (0-8): |
```

Project demo

10

Thank You!