People's Democratic Republic of Algeria
Ministry of Higher Education and Scientific Research
Saad Dahleb Blida 01 University
Department of computer science

MASTER'S INTELLIGENT SYSTEMS ENGINEERING

ARTIFICIAL INTELLIGENCE

REPORT

TIC TAC TOE GAME IMPLEMENTATION

–

PYTHON GAME IMPLIMENTATION USING MIN-MAX ALGORITHM

Made by :

ABDELATIF MEKRI
IMAD EDDINE BOUKADER

Academic year : 2023-2024

Contents

KEY WORDS :

XO , XOXO , TIC-TAC-TOE , USER-VS-MACHINE , PYTHON ,MINMAX ALGORITHM , TKINTER

1. Introduction

The Tic Tac Toe project is a Python implementation of the classic game using the Tkinter library for the graphical user interface. The primary goal was to create a functional game with a player-vs-Computer mode, featuring a basic AI like opponent that makes optimal moves using the minimax algorithm.

2. Project Overview

The project consists of a graphical user interface (GUI) built with Tkinter, a TicTacToe class to manage the game state, and a computer opponent capable of playing against the player. The game allows the player to choose their symbol (X or O) and features a reset button for restarting the game.

3. Implementation Details

3.1. 'TicTacToe' Class

The heart of the game is the TicTacToe class, which maintains the game state and logic. Here's an excerpt:

```
class TicTacToe:
    def __init__(self, player_symbol):
        # Initialization of the game board, current player, and game-over status
        # ...

    def apply_move(self, move):
        # Method to apply a move to the game board
        # ...

    def toggle_player(self):
        # Method to toggle the current player
        # ...
    def minimax(self, depth, maximizing_player):
        # Minimax algorithm for move calculations
        # ...
```

3.2 GUI Setup

The graphical user interface is set up using Tkinter. Here's a snippet showcasing the GUI layout:

```
def choose_symbol():
        # Method to choose the player's symbol , taking X as defult
        # ...

    def set_player_symbol():
        # Method to set the player's symbol
```

```
        # ...
        # Styling the radiobuttons for symbol choice
# GUI Setup
app = tk.Tk()
        # Setting app title and configuration
        # ...
game = TicTacToe(X)
# Create GUI labels for the Tic Tac Toe board and associate them with the
player_move function and adding buttons
```

3.3. Game Logic

a. The Player

Player moves are handled through the player_move function, which applies the move, updates the GUI labels, and checks for a winner or a tie.

```
def player_move(move):
    game.apply_move(move)
    update_board_labels()

    # Check for game outcome
    if game.is_winner():
        # Display winner/loser message and reset the game
        # ...
    elif game.is_board_full():
        # Display tie message and reset the game
        # ...
    else:
        # Computer makes a move
        make_computer_move(game)
        update_board_labels()
        # Check for game outcome after move
        # ...
```

b. The computer

```
def minimax(self, depth, maximizing_player):
    # Minimax algorithm implementation
    # ...
def make_computer_move(game):
    # AI move using minimax algorithm
```

4. Usage

To use the TIC-TAC-TOE game :

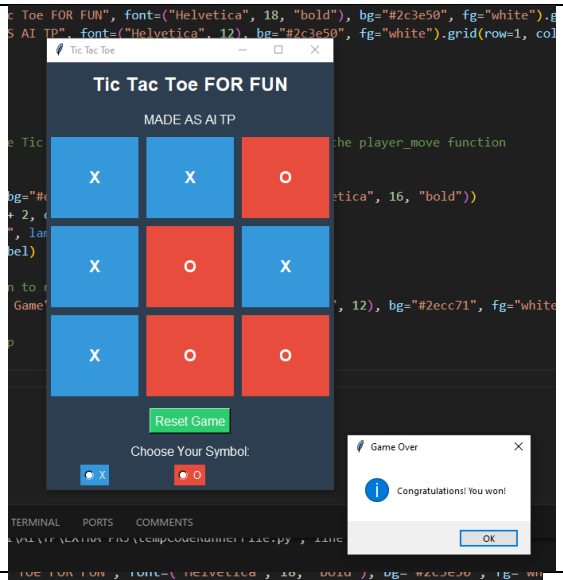
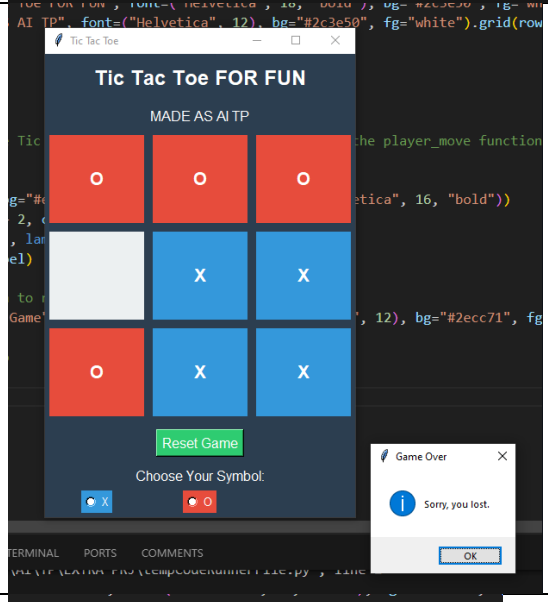
-Clone the repository to your local machine.

-Open the project on VSC (for example) .

-Run the game : **python XOXO.py**

5. GUI example

Example of the User Interface Implimentd :

The Player Wins	
The Player Loses	
A Tie	

6. Future Considerations

While the project meets its current goals, there are opportunities for improvement. Future enhancements could include

- Saving the player's score , whoch might help refine the algorithm,

- Adding additional features,

- Improving the overall user interface for a more polished gaming experience..

7. Conclusion

In conclusion, the Tic Tac Toe project successfully implements a functional game with a player-vs-Computer mode. The use of the minimax algorithm adds a challenging opponent. The GUI provides a user-friendly experience, allowing players to enjoy the classic game with an intuitive interface.