

VISVESVARAYA TECHNOLOGICAL UNIVERSITY

Jnana Sangama, Belagavi - 590018



Mini project

On

Python Programming Laboratory (21CSL46)

“TIC TAC TOE GAME USING tkinter LIBRARY”

By

PRANAV SHIRALI

4MT21CS104

SULAIMAN PILOOR

4MT21CS081

PRANSHU SHARMA

4MT21CS106

NAKWA ABDUL REHAN

4MT21CS091

Mentor

Mrs. Amrutha

Assistant Professor

Department of Computer Science & Engineering



DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

(Accredited by NBA)

MANGALORE INSTITUTE OF TECHNOLOGY & ENGINEERING

Accredited by NAAC with A+ Grade, An ISO 9001: 2015 Certified Institution

(A Unit of Rajalaxmi Education Trust®, Mangalore - 575001)

Affiliated to VTU, Belagavi, Approved by AICTE, New Delhi

Badaga Mijar, Moodabidri-574225, Karnataka



2022-23

MANGALORE INSTITUTE OF TECHNOLOGY & ENGINEERING

Accredited by NAAC with A+ Grade, An ISO 9001: 2015 Certified Institution

(A Unit of Rajalaxmi Education Trust®, Mangalore - 575001)

Affiliated to VTU, Belagavi, Approved by AICTE, New Delhi

Badaga Mijar, Moodabidri-574225, Karnataka

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

(Accredited by NBA)

CERTIFICATE

This is to certify that the mini project entitled “Tic Tac Toe Game Using Tkinter Library” is a Bonafede work carried out by **PRANAV SHIRALI(4MT21CS104)**, **SULAIMAN PILOOR(4MT21CS081)**, **PRANSHU SHARMA(4MT21CS106)** and **NAKWA ABDUL REHAN(4MT21CS091)** in partial fulfilment for the requirement of 4th semester **Python Programming Laboratory (21CSL46)**. The mini project has been approved as it satisfies the academic requirement in respect of the 21CSL46 prescribed for the 4th Semester B.E in Computer Science & Engineering Program by the **Visvesvaraya Technological University, Belagavi** for the academic year 2022 – 2023.

.....

Mrs. Amrutha

Assistant Professor
MITE, Moodabidri

Abstract

The Tic-Tac-Toe Game with GUI is a fun Python project that offers a contemporary take on the traditional game of Tic-Tac-Toe in a digital format. The game may be played interactively on computer displays thanks to the project's usage of the tkinter package to construct a graphical user interface. Players take turns marking their preferred symbols ('X' or 'O') by clicking on the buttons on a 3x3 grid that serves as the game board. The implementation has game logic to check for winning combinations in rows, columns, and diagonals as well as to determine whether the game is tied. Players receive real-time feedback through the GUI, which shows win/tie messages in pop-up message boxes. The project provides instruction in data representation using dictionaries, GUI construction, event management, and game logic implementation. It offers a fun and engaging method to play Tic-Tac-Toe and is a great learning tool for Python aficionados wishing to improve their coding abilities.

INTRODUCTION

Noughts and Crosses, another name for the popular paper-and-pencil game of tic-tac-toe, is played by players of all ages all over the world. In this two-player game, players alternately mark a 3x3 grid with the sign ('X' or 'O') of their choice. The goal of the game is to place three of your symbols in a line, column, or diagonal on the grid; the person who does it first wins the round. When every square in the grid is filled but no player has managed to make a winning combination, the game is deemed a tie.

The actual pen and paper are normally used to play Tic-Tac-Toe in its classic form, but as technology has advanced, computerized versions of the game have grown in popularity. the previous

Tic-Tac-Toe is normally played with a real pen and piece of paper, but as technology has advanced, computerized versions of the game have grown in popularity. The aforementioned code displays a graphical user interface (GUI)-based interactive Python implementation of Tic-Tac-Toe.

Project Description

By the use of the tkinter package to generate a graphical representation of the game board, the goal of this Python project is to give users a contemporary and entertaining method to play Tic-Tac-Toe. Players can now engage with the game through a user-friendly GUI on their computer screen rather than the conventional pen and paper.

Learning outcomes and user experience

Players can command their movements by clicking on the GUI buttons directly, making for an exciting and dynamic experience. Playing is made more enjoyable by receiving immediate feedback on whether they have won or tied the game.

For prospective Python coders, this project provides numerous significant learnings:

1. **GUI Development:** Learning about GUI development will help students understand how to use the Tkinter library to build a GUI application. They'll know how to organize and develop a user-friendly interface for increased user engagement.
2. **Event Handling:** The project emphasizes the significance of event management in GUI applications. The ability to link functions to certain events and react to them will be understood by learners.
3. **Game Logic:** The Tic-Tac-Toe game logic is used to show how to check for winning combinations and decide how the game will turn out.
4. **Data Structures:** The use of dictionaries to represent the Tic-Tac-Toe board showcases the effectiveness of data structures in handling game states.

IMPLEMENTATION

The implementation of the Tic-Tac-Toe game with a GUI involves creating a graphical representation of the game board and implementing game logic for player moves and win/tie conditions. The board is represented using a Python dictionary where each key corresponds to a position on the 3x3 grid, and the value represents the player's move ('X' or 'O') or an empty space (' ') for unmarked positions. The GUI is built using the tkinter library, with buttons representing the board positions. When a player clicks on a button, the corresponding event handler function is triggered, marking the position on the board and updating the button label accordingly. The game logic checks for winning combinations after each move and displays a message box to announce the winner or a tie. The project emphasizes the interaction between game logic and the GUI, enabling an engaging and user-friendly gaming experience.

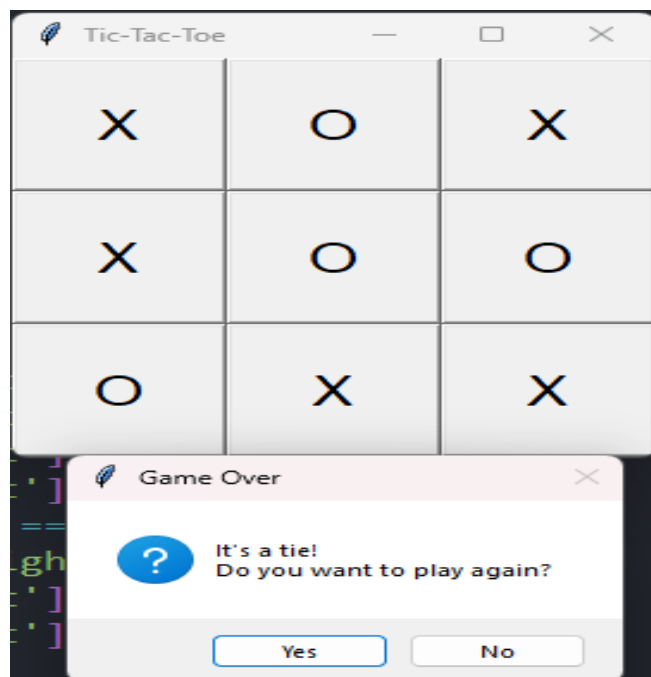
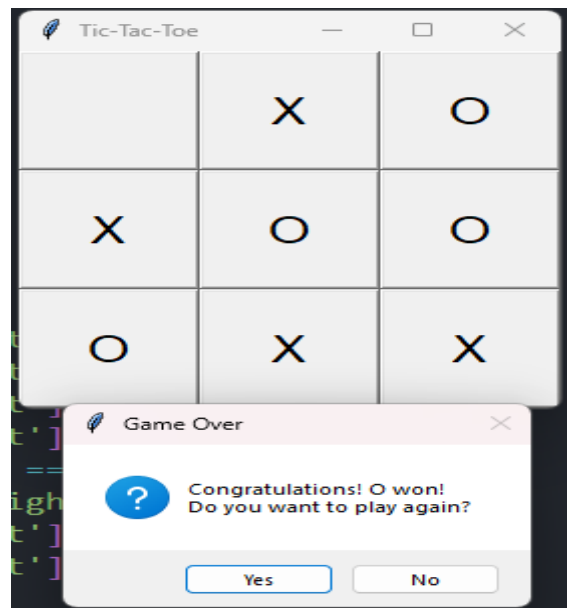
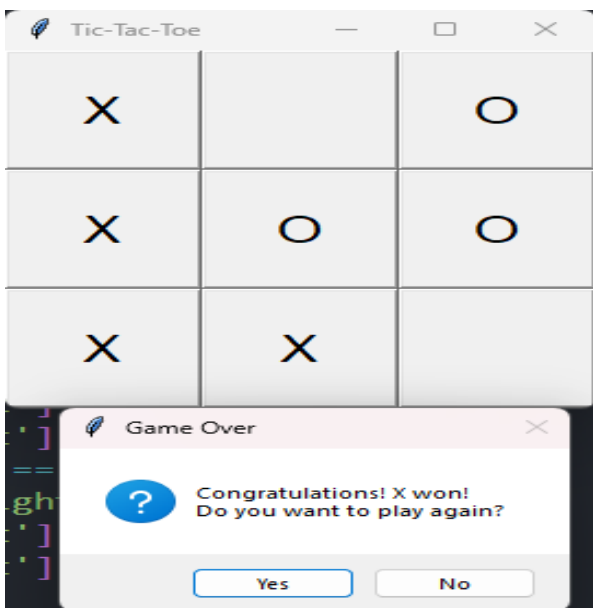
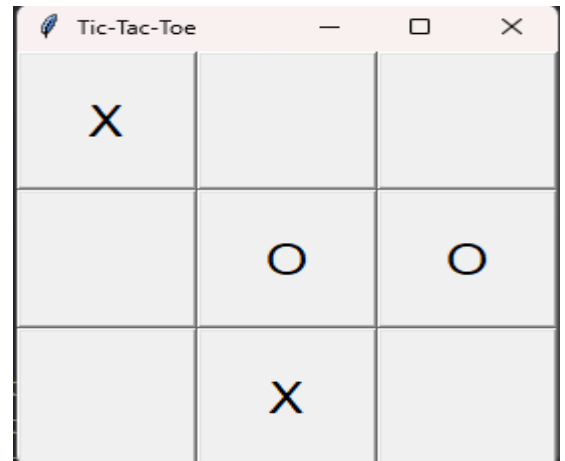
Code for this project is attached in next page:

```

1  import tkinter as tk
2  from tkinter import messagebox
3
4  theBoard = {
5      'topleft': ' ', 'topmid': ' ', 'topright': ' ',
6      'midleft': ' ', 'center': ' ', 'midright': ' ',
7      'botleft': ' ', 'botmid': ' ', 'botright': ' '
8  }
9
10 def check_win(player, board):
11     # Check rows, columns, and diagonals
12     for row in range(3):
13         if (board[f'topleft'] == board[f'topmid'] == board[f'topright'] == player or
14             board[f'midleft'] == board[f'center'] == board[f'midright'] == player or
15             board[f'botleft'] == board[f'botmid'] == board[f'botright'] == player or
16             board[f'topleft'] == board[f'midleft'] == board[f'botleft'] == player or
17             board[f'topmid'] == board[f'center'] == board[f'botmid'] == player or
18             board[f'topright'] == board[f'midright'] == board[f'botright'] == player or
19             board[f'topleft'] == board[f'center'] == board[f'botright'] == player or
20             board[f'topright'] == board[f'center'] == board[f'botleft'] == player):
21             return True
22     return False
23
24
25 def play_again():
26     global turn
27     for key in theBoard:
28         theBoard[key] = ' '
29         button_dict[key].config(text=' ')
30     turn = 'X'
31
32 def on_click(key):
33     global turn
34     if theBoard[key] == ' ':
35         theBoard[key] = turn
36         button_dict[key].config(text=turn)
37         if check_win(turn, theBoard):
38             if messagebox.askyesno("Game Over", f"Congratulations! {turn} won!\nDo you want to play again?"):
39                 play_again()
40             else:
41                 root.quit()
42         elif ' ' not in theBoard.values():
43             if messagebox.askyesno("Game Over", "It's a tie!\nDo you want to play again?"):
44                 play_again()
45             else:
46                 root.quit()
47         else:
48             turn = 'O' if turn == 'X' else 'X'
49
50 position_to_indices = {
51     'topleft': (0, 0), 'topmid': (0, 1), 'topright': (0, 2),
52     'midleft': (1, 0), 'center': (1, 1), 'midright': (1, 2),
53     'botleft': (2, 0), 'botmid': (2, 1), 'botright': (2, 2)
54 }
55
56 root = tk.Tk()
57 root.title("Tic-Tac-Toe")
58
59 button_dict = {}
60 for key, (row, col) in position_to_indices.items():
61     button_dict[key] = tk.Button(root, text=' ', font=('Helvetica', 20), height=2, width=5, command=lambda k=key: on_click(k))
62     button_dict[key].grid(row=row, column=col)
63
64 turn = 'X'
65 root.mainloop()

```

RESULT



CONCLUSION AND FUTURE ENHANCEMENT

The Tic-Tac-Toe Game with GUI small project is a success in achieving its goal of giving players a fun and engaging gaming experience. The project develops a user-friendly interface that enables players to participate in the classic game on their computer screens using Python programming and the tkinter package. Smooth and simple gameplay is made possible by the GUI implementation, which uses a 3x3 grid of buttons to represent the places on the board.

The project's code employs a well-organized structure, separating game logic from GUI implementation. The game logic ensures that players can play the game with standard Tic-Tac-Toe rules, with immediate feedback on whether they win or the game ends in a tie. The use of a Python dictionary to represent the board positions efficiently handles the state of the game.

Furthermore, the mini project serves as an excellent learning resource for both novice and intermediate Python programmers. Learners can gain insights into GUI development, event-driven programming, data structures, and game logic implementation. Additionally, the project fosters problem-solving skills and encourages creativity in exploring new features and improvements.

For future enhancements, several avenues can be explored to enrich the project. Some possible improvements include:

1. **Enhanced User Interface:** Incorporate graphics and visual effects to make the GUI more appealing and engaging for players.
2. **Player Modes:** Implement different player modes, such as playing against an AI or enabling multiplayer functionality.
3. **Difficulty Levels:** Add varying difficulty levels for the AI player, catering to players of different skill levels.
4. **Score Tracking:** Introduce a score tracker to keep records of wins, losses, and ties for different players.
5. **Board Size Options:** Offer customizable board size options, allowing players to play Tic-Tac-Toe on larger grids for added complexity.
6. **Undo/Redo Feature:** Implement an undo/redo feature to allow players to revisit previous moves during gameplay.
7. **Online Multiplayer:** Integrate network capabilities to enable players to compete against others online.

In conclusion, the Tic-Tac-Toe Game with GUI mini project serves as a testament to the versatility of Python and the capabilities of the tkinter library for GUI development. The project's successful implementation, along with the potential for future enhancements, demonstrates its value as an educational and entertaining Python project. By continuing to build upon the project, learners can further sharpen their programming skills and create a more immersive gaming experience for players worldwide.

REFERENCES

1. Automate The Boring Stuff by Al Sweigart
2. https://www.tutorialspoint.com/python/python_gui_programming.htm
3. <https://www.geeksforgeeks.org/python-gui-tkinter/>
4. <https://docs.python.org/3/library/tkinter.html>
5. <https://medium.com/pythoneers/building-a-tic-tac-toe-game-in-python-with-tkinter-e9060345f629>
6. **GitHub link to this project:**
<https://github.com/pranavshirali/AutomateBoringStuff/blob/master/TicToeGameGUI.py>