

# Project Topic 01 Tic Tac Toe Game

Date: 4th March 2024

## Group 2 - Team Members

1. Vivek Sharma
2. Muhammad Zaid
3. Dang Ngoc Tai Tran
4. Munasinghe Arachchige Chandina Nilukshi
5. Malinda Bandara Weragama Kahanda Walawwe

Our goal is to create a simple and interactive Tic-Tac-Toe game with user interface (UI) in Jupyter notebook.

The game should allow two players to make moves, detect a winner, and end with a clear result.

Import the module for UI development as follows

```
In [1]: import tkinter as tk
```

This is used to get the message box to print the result

```
In [2]: from tkinter import messagebox
from tkinter import scrolledtext
import markdown
```

Create the board and, to place the buttons correctly we use two loops

```
In [3]: def create_board(master, on_button_click):

    buttons = [[None]*3 for _ in range(3)]
    for i in range(3):
        for j in range(3):
            buttons[i][j] = tk.Button(master, text="", width=8, height=3,
                                      font=('Times New Roman', 20, 'bold'),
                                      command=lambda row=i, col=j: on_button_click(row, col))
            buttons[i][j].grid(row=i, column=j, padx=5, pady=5)
    return buttons
```

It places "X" or "O" inside the button depending on the current player. In this function, the function to check the winner, function to check if it's a tie and reset the board and if the game isn't ended yet the function to switch player is executed.

```
In [4]: def on_button_click(row, col):
    if buttons[row][col]['text'] == "":
        buttons[row][col]['text'] = current_player
        board[row][col] = current_player
        if check_winner(row, col):
            messagebox.showinfo("Result", f"{current_player} Wins The Game Congratulations !!! ")
            reset_game()
        elif check_tie():
            messagebox.showinfo("Result", "It's a tie! Try Again")
            reset_game()
        else:
            switch_player()
```

This is the function to check the winner.

```
In [5]: def check_winner(row, col):

    if all(board[row][i] == current_player for i in range(3)):
        return True

    if all(board[i][col] == current_player for i in range(3)):
```

```

        return True

    if all(board[i][i] == current_player for i in range(3)):
        return True

    if all(board[i][2 - i] == current_player for i in range(3)):
        return True
    return False

```

This is a function to check if it is a tie.

```

In [6]: def check_tie():
        return all(board[i][j] != ' ' for i in range(3) for j in range(3))

```

This is the function to switch the player.

```

In [7]: def switch_player():
        global current_player
        if current_player == "xXx":
            current_player = "0o0"
        else:
            current_player = "xXx"

```

This is the function to reset the game by clearing the board.

```

In [8]: def reset_game():
        for i in range(3):
            for j in range(3):
                buttons[i][j]['text'] = ""
                board[i][j] = " "
        switch_player()

```

This is the function that set up the GUI window including text box.

```

In [9]: def main():
        global buttons, board, current_player

        root = tk.Tk()
        root.title("Tic-Tac-Toe Game By Group 2")
        root.resizable(width=False, height=False)
        root.geometry("450x560")

        frame = tk.Frame(root)
        frame.grid()

        current_player = "xXx"
        board = [[' ']*3 for _ in range(3)]
        buttons = create_board(frame, on_button_click)
        markdown_text = ""

```

Project Assignment by Group 2  
 Tic Tac Toe Game with User Interface in Jupyter notebook  
 Created by  
 1. Vivek Sharma  
 2. Muhammad Zaid  
 3. Dang Ngoc Tai Tran  
 4. Munasinghe Arachchige Chandina Nilukshi  
 5. Malinda Bandara Weragama Kahanda Walawwe  
 Special Thank for Teacher Melinda Varo

```

"""
    html_text = markdown.markdown(markdown_text)

    text_widget = tk.Text(root, width=52, height=10)
    text_widget.insert(tk.END, html_text)
    text_widget.configure(state='disabled')
    text_widget.grid(row=4, column=0)

    root.mainloop()

```

This code ensures that the script is run directly.

```

In [10]: if __name__ == "__main__":
        main()

```

## Implementation:

We plan to use functions to handle different aspects of the game, such as printing the board, checking for a winner, and determining a tie. The game loop will allow players to make moves

until there is a winner or a tie, and the final result will be clearly displayed.

## Conclusion:

The Tic-Tac-Toe game with user interface is now completed ! It provides a simple and interactive experience for two players.

The code checks for winning conditions and ties, making it a fully functional game.

### GitHub link

<https://github.com/Viveksharma2000/projectworkgroup2.git>

In [ ]:

Loading [MathJax]/jax/output/CommonHTML/fonts/TeX/fontdata.js