

# Assignment - 5

28.02.2025

group - 10

## Team Members

- Ayesha Subhashanee
- Yamong Phoo
- Nethmi Nayanathara Egodawaththa Arachchige
- Ama Nethmini Anjana Wathukarage
- Tharindu Kavindya Silva Hettiyadura

Group 10 - Trello Board Project link: <https://trello.com/b/wfK43SUK/future-project>

GitHub Repository link: <https://github.com/Ayesha-Subhashanee/Group-project>

## Tic - Tac - Toe

### Descriptions

Tic\_Tac\_Toe game is two players game. It plays in a 3 x 3 grid.

Each player take marking in the square with their respective symbols like "X" or "O" which are common use while playing this game.

**The first player who align of three of marks in a horizontal or vertical or diagonal is the winner.**

### Rules

- Two player game with taking turn is played in a 3 x 3 grid.
- Each player choose their respective symbol. Example; Player 1 choose "X" and Player 2 choose "O".
- Players take turn and mark their respective symbol in an empty square cell.
- Each player try to mark with respective symbol not to win the other player.
- Players also try to align their symbol in a horizontal or vertical or diagonal.
- If 9 all cells are filled, there is no winning player. The game is end in a draw.

## Main Window Setup (UI Components)

```
In [ ]: import tkinter as tk

class TicTacToe:
    def __init__(self, root):
        self.root = root
        self.root.title("Tic-Tac-Toe")
        self.root.geometry("400x450")
        self.root.configure(bg="#f8c8c8")

        self.current_player = "X"
        self.board = [""] * 9

        self.status_label = tk.Label(self.root, text="Player X's Turn", font=("Arial", 16), bg="#f8c8c8")
        self.status_label.pack(pady=10)

        self.buttons = []
        self.create_grid()

        self.restart_button = tk.Button(self.root, text="Restart", font=("Arial", 14), bg="#4CAF50", fg="white",
                                         command=self.reset_game)
        self.restart_button.pack(pady=10)

    def create_grid(self):
        frame = tk.Frame(self.root)
        frame.pack()
        for i in range(9):
            btn = tk.Button(frame, text="", font=("Arial", 20), width=5, height=2,
                             command=lambda i=i: self.player_move(i))
            btn.grid(row=i // 3, column=i % 3, padx=5, pady=5)
            self.buttons.append(btn)

    def player_move(self, index):
        if self.board[index] == "" and not self.check_winner():
```

```

self.board[index] = self.current_player
self.buttons[index].config(text=self.current_player)

if self.check_winner():
    self.status_label.config(text=f"Player {self.current_player} Wins!")
    self.disable_buttons()
elif self.check_draw():
    self.status_label.config(text="It's a Draw!")
else:
    self.current_player = "O" if self.current_player == "X" else "X"
    self.status_label.config(text=f"Player {self.current_player}'s Turn")

def check_winner(self):
    win_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]
    ]
    for combo in win_combinations:
        if self.board[combo[0]] == self.board[combo[1]] == self.board[combo[2]] != "":
            return True
    return False

def check_draw(self):
    return "" not in self.board and not self.check_winner()

def reset_game(self):
    self.current_player = "X"
    self.board = [""] * 9
    self.status_label.config(text="Player X's Turn")
    for button in self.buttons:
        button.config(text="", state="normal")

def disable_buttons(self):
    for button in self.buttons:
        button.config(state="disabled")

if __name__ == "__main__":
    root = tk.Tk()
    game = TicTacToe(root)
    root.mainloop()

```

## Conclusion

In this project, we successfully implemented a Tic-Tac-Toe game using Python and Tkinter. The game allows two players to take turns marking "X" or "O" on a 3x3 grid, with the objective of aligning three marks in a row, column, or diagonal to win. If all spaces are filled without a winner, the game ends in a draw.

In [ ]:

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