

**Multiplayer Tic-Tac-Toe**

*ENGI316 Programming in Matlab for Engineering*

***Submitted To***

*Asst. Prof. Dr. Neyre TEKBIYIK ERSOY*

***By***

*Lesley Esenei Ajone Lafen – 21901004*

*Department of Software Engineering*

***and***

*Yusuf Ahmed Yusuf* – *21813138*

Department of Computer Engineering

**Multiplayer Tic-Tac-Toe**

INTRODUCTION

Tic Tac Toe is a board game played by two players on a 3x3 grid who alternately put either X or O in the grid spaces until the game results in a winner or a draw. In the case of this application, a background color of blue and red are used as player marks in place of X and O. To obtain a win, a player must place their marks on a straight horizontal, vertical, or diagonal line. If all spaces in the grid are occupied and no winner is attained up to this point, the game results in a draw. Another minor but necessary consideration is ensuring a player can’t play in a grid space that is occupied. At the end of each round, the result is added to an excel file “game\_record.xlsx” and a graph of player 1 vs player 2 scores is plotted.

# Problem Definition and Formulation:

* Explanation of the problem: The problem is to create a Tic-Tac-Toe game where two players take turns placing their symbols (Player 1: 'X' and Player 2: 'O') on a 3x3 game board. The goal is to get three of their symbols in a row, either horizontally, vertically, or diagonally. The game continues until one player wins or the game board is full and results in a tie.
* Definitions of the variables used in the problem:
  + gameMatrix: A 3x3 matrix representing the game board, initialized with zeros.
  + isPlayer1Turn: A boolean variable indicating whether it is Player 1's turn or not.
  + pos: A variable storing the input position (row, col) where a player wants to place their symbol.
  + winner: A variable indicating the player who wins the game (1 for Player 1, 2 for Player 2, 0 for a tie).
* Descriptions of the tasks that need to be performed to solve the problem:
  1. Initialize the game matrix and create the game board.
  2. Prompt the players for their moves and validate the input.
  3. Update the game matrix with the player's move.
  4. Check if there is a winner or if the game results in a tie.
  5. Display appropriate messages and update game records accordingly.
  6. Repeat the above steps until the game ends.

Solution Method:

* The calculations required to solve each task of the problem are as follows:
  + Task 1: Initialize the game matrix with zeros.
  + Task 2: Prompt the players for their moves and validate the input using the **input** function and the **validate\_input** function.
  + Task 3: Update the game matrix with the player's move by assigning the corresponding element in the matrix to the player's symbol (1 for Player 1, 2 for Player 2).
  + Task 4: Check if there is a winner or if the game results in a tie using the **check\_win** and **is\_board\_full** functions.
  + Task 5: Display appropriate messages using the **disp** function and show message boxes using the **msgbox** function. Update game records using the **add\_game\_record** function.
  + Task 6: Repeat the above tasks until the game ends using a while loop.
* The equations used to solve the problem are:
  + No specific equations are used in this problem.
* The symbols used in the equations are:
  + None, as there are no specific equations in this problem.