TicTacToe Part 1

Contents

Time required: 60 minutes

1. Create a MATLAB script named **TicTacToe.m**
2. Include your name and date at the top of the script file as comments.

Tic Tac Toe

A classic game programming problem.

CTRL C is Your Friend

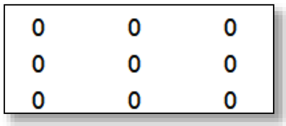
If you want to exit or break out of a MATLAB program early:

1. Hold down the **CTRL** key
2. Type **C**
3. Let go of the **CTRL** key

Step 1: Create the Game Board

Tic Tac Toe requires a 3 x 3 matrix.

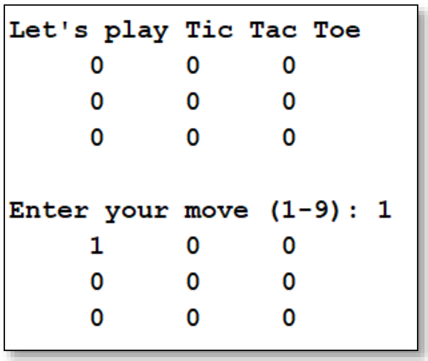
1. Create a 3 x 3 matrix named board that contains 0’s.
2. Create a function that displays the game board with the board matrix as an argument.



```
    0     0     0
    0     0     0
    0     0     0
```

3. Display the game title.

4. Get the player’s input.

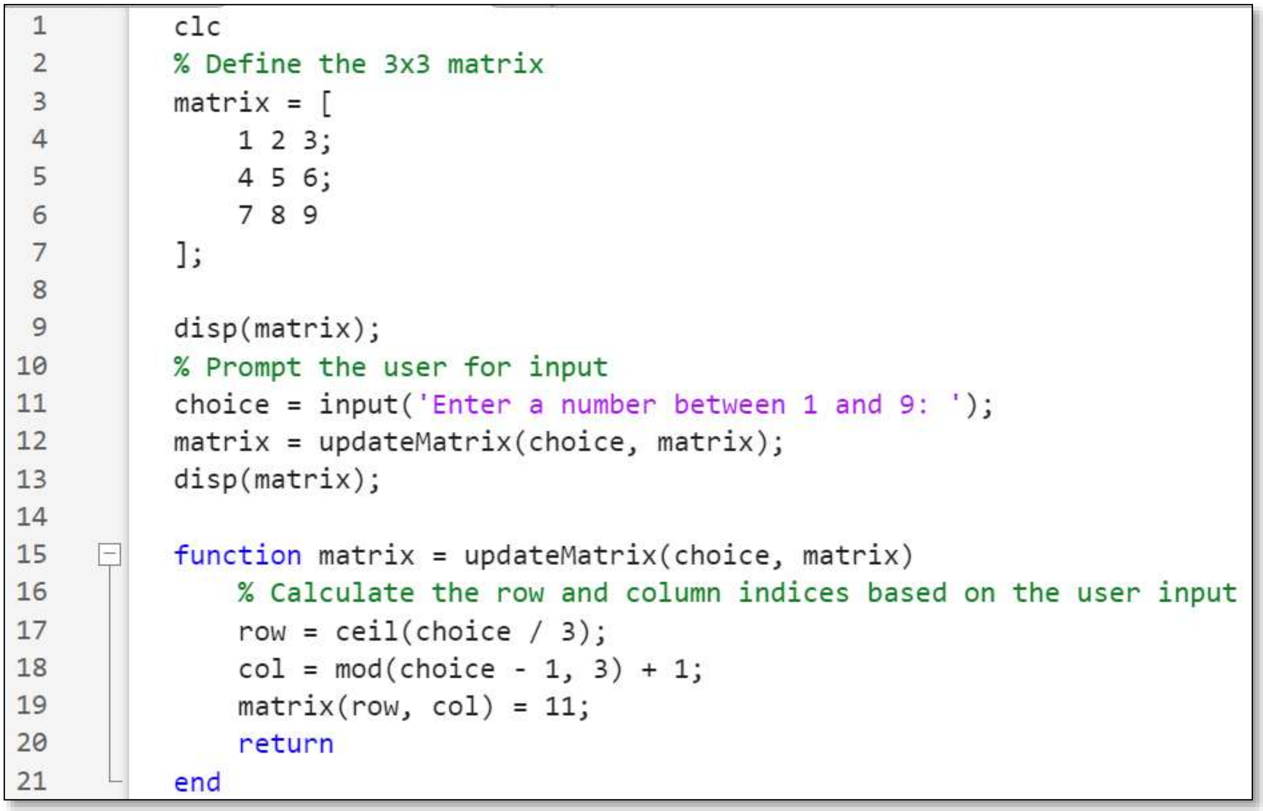


```
Let's play Tic Tac Toe
     0     0     0
     0     0     0
     0     0     0

Enter your move (1-9): 1
     1     0     0
     0     0     0
     0     0     0
```

Convert from 1-9 to Matrix Coordinates

The user is typing in 1-9. We want to return the matrix coordinates.



```matlab
clc
% Define the 3x3 matrix
matrix = [
    1 2 3;
    4 5 6;
    7 8 9
];

disp(matrix);
% Prompt the user for input
choice = input('Enter a number between 1 and 9: ');
matrix = updateMatrix(choice, matrix);
disp(matrix);

function matrix = updateMatrix(choice, matrix)
    % Calculate the row and column indices based on the user input
    row = ceil(choice / 3);
    col = mod(choice - 1, 3) + 1;
    matrix(row, col) = 11;
    return
end
```

row = ceil(choice / 3);

- choice / 3: Divides the user's input by 3. This helps determine which row the input corresponds to.
- ceil(...): The ceil function rounds up to the nearest integer. This ensures that any fractional result from the division is rounded up to the next whole number, giving us the correct row index.

col = mod(choice - 1, 3) + 1;

- choice - 1: Subtracts 1 from the user's input. This adjustment is necessary because MATLAB indices start at 1, but the modulo operation works more intuitively with a zero-based index.
- mod(..., 3): The mod function calculates the remainder of integer division when the adjusted input is divided by 3. This gives us the column index within the range of 1 to 3.
- + 1: Adds 1 to the result of the modulo operation to shift back to MATLAB's 1-based indexing.

Example

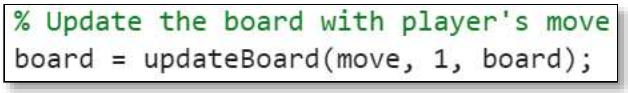
If the user enters 5:

- **Row Calculation**: row = ceil(5 / 3) = ceil(1.6667) = 2
- **Column Calculation**: col = mod(5 - 1, 3) + 1 = mod(4, 3) + 1 = 1 + 1 = 2

The matrix location for the input 5 is (2, 2).

Step 2: Update Board

1. Create a function that updates the board named updateBoard.
2. updateBoard has three arguments
   a. move - This is 1-9 indicating where in the matrix the play was made
   b. player - 1 is the player, 2 is the computer
   c. board - This is the game board.
3. This is an example of a call to the **updateBoard** function. We are hard coding the **updateBoard** function with the player’s number for now.

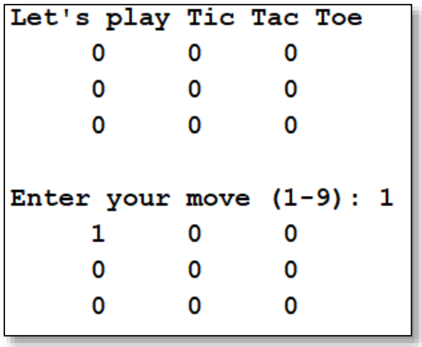


```
% Update the board with player's move
board = updateBoard(move, 1, board);
```

4. Replace the 0 with a 1 for the current players move.
   Use the above program Convert 1-9 to MATRIX

5. Redisplay the game board.

Example run:



```
Let's play Tic Tac Toe
     0     0     0
     0     0     0
     0     0     0

Enter your move (1-9): 1
     1     0     0
     0     0     0
     0     0     0
```

Assignment Submission

1. Submit properly named and commented script files.

2. Attach a screenshot of the Command Window showing the successful execution of each script.

3. Attach all to the assignment in Blackboard.