

**Department of Computer Science**

**CSCL2201 – Computer Organization and Assembly Language**

**Tic Tac Toe**

# Project Report

**Semester: Fall 2024**

**Group members:**

1. **Alisha Anwar (2312105)**
2. **Hira Niaz Malik(2312113)**
3. **Muhammad Yaseen Ahmed (2212153)**
4. **Areeba Zulfiqar(2312393)**
5. **Aimen Nisar(2312129)**
6. **Laiba Jabbar(2312394)**

***Submitted to:***

**Sir Saqib Sadiq**

***Lab Engineer***

**Sir Hurrar Hassan Rizvi**

## Table of Content:

* PSEUDOCODE
* FLOWCHART
* SOURCE CODE
* SCREENSHOTS
* DEBUG LOG

## PSEUDOCODE

### Pseudocode for Assembly Program

1. **Initialization**

Load the data segment (DS) to access our variables.

1. **Title Screen**

Display a warm welcome using the title string, Wait for the user to press any key to continue.

1. **Explain the Game Rules**

Show the rules one line at a time to make them easy to follow, Again, wait for user input to proceed.

1. **Set up the game**

Reset all important variables:

1. **Game Loop**

a. Display the game board  
b. Check if the game is over  
c. Prompt the current player for input  
d. Update the board  
e. Switch turns

1. **End Game if Win or Draw**
2. **Ask to play again**.
3. **Exit**

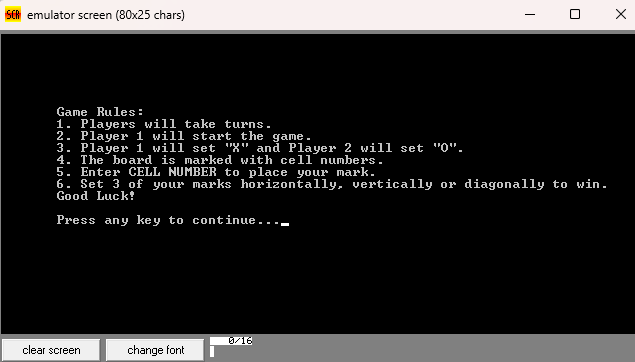
## FLOWCHART

## SCREENSHOTS

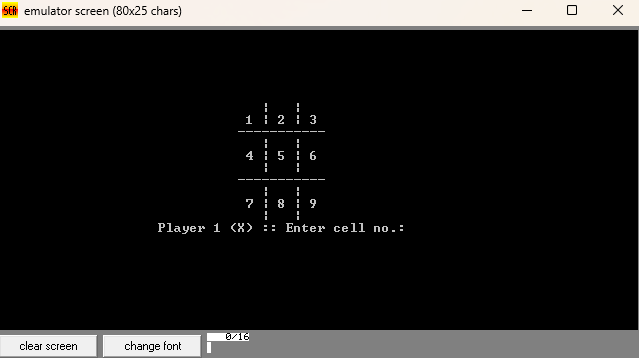
#### Output

**Menu**

****

**Rules**

**Display Board**

****

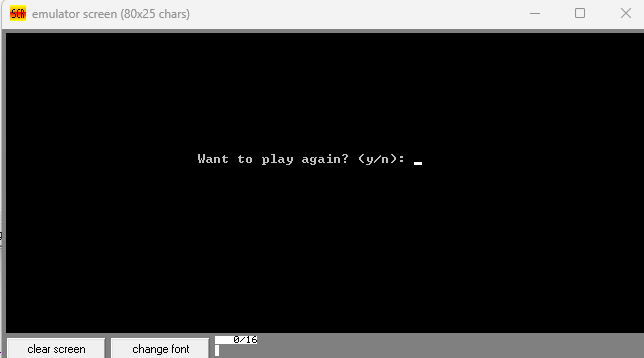
**Wrong Input**

****

**Game Result**

****

**Try Again**

****

## SOURCE CODE

PRINT MACRO P1

LEA DX,P1 ;PRINT

MOV AH,9

INT 21H

ENDM

INPUT MACRO

MOV AH, 7 ; INPUT WITHOUT ECHO

INT 21H

ENDM

INECHO MACRO

MOV AH, 1 ;INPUT WITH ECHO

INT 21H

ENDM

.MODEL SMALL

.STACK 500H

.DATA

; LINES T1, T2, T3 AND T4 ARE USED TO BUILD THE TIC - TAC - TOE LOGO

T1 DB 4, 4, 4, 4, 4, 32, 4, 32, 4, 4, 4, 4, 32, 32, 32, 4, 4, 4, 4, 4, 32, 32, 4, 4, 32, 32, 4, 4, 4, 4, 32, 32, 32, 4, 4, 4, 4, 4, 32, 32, 4, 4, 32, 32, 4, 4, 4, 4, '$'

T2 DB 32, 32, 4, 32, 32, 32, 4, 32, 4, 32, 32, 32, 32, 32, 32, 32, 32, 4, 32, 32, 32, 4, 32, 32, 4, 32, 4, 32, 32, 32, 32, 32, 32 , 32, 32, 4, 32, 32, 32, 4, 32, 32, 4, 32, 4,'$'

T3 DB 32, 32, 4, 32, 32, 32, 4, 32, 4, 32, 32, 32, 32, 32, 32, 32, 32, 4, 32, 32, 32, 4, 4, 4, 4, 32, 4, 32, 32, 32, 32, 32, 32 , 32, 32, 4, 32, 32, 32, 4, 32, 32, 4, 32, 4, 4, 4, 4,'$'

T4 DB 32, 32, 4, 32, 32, 32, 4, 32, 4, 4, 4, 4, 32, 4, 32, 32, 32, 4, 32, 32, 32, 4, 32, 32, 4, 32, 4, 4, 4, 4, 32, 4, 32 , 32, 32, 4, 32, 32, 32, 32, 4, 4, 32, 32, 4, 4, 4, 4,'$'

;--------------------------------------------------------------------

; ----------------- STRINGS USED THROUGHOUT THE GAME -----------------

PTC DB 'Press any key to continue...$'

; GAME RULES

R DB 'Game Rules:$'

R1 DB '1. Players will take turns.$'

R2 DB '2. Player 1 will start the game.$'

R3 DB '3. Player 1 will set "X" and Player 2 will set "O".$'

R4 DB '4. The board is marked with cell numbers.$'

R5 DB '5. Enter CELL NUMBER to place your mark.$'

R6 DB '6. Set 3 of your marks horizontally, vertically or diagonally to win.$'

R7 DB 'Good Luck!$'

; CELL MARK FOR PLAYERS

PC1 DB ' (X)$'

PC2 DB ' (O)$'

; BOARD LINES

L1 DB ' | | $'

L2 DB '-----------$'

N1 DB ' | $'

; CELL NUMBERS

C1 DB '1$'

C2 DB '2$'

C3 DB '3$'

C4 DB '4$'

C5 DB '5$'

C6 DB '6$'

C7 DB '7$'

C8 DB '8$'

C9 DB '9$'

; PLAYER NO. , MOVES AND CHECK FLAGS FOR IF THE GAME IS WON OR DRAWN

PLAYER DB 50, '$'

MOVES DB 0

DONE DB 0

DR DB 0

; INPUT SECTION PROMTS

INP DB 32, ':: Enter cell no. : $'

TKN DB 'This cell is taken! Press any key...$'

; CURRENT MARK (X/O)

CUR DB 88

; FINAL MESSAGES

W1 DB 'Player $'

W2 DB ' won the game!$'

DRW DB 'The game is draw!$'

; TRY AGAIN PROMPT MESSAGES

TRA DB 'Want to play again? (y/n): $'

WI DB 32, 32, 32, 'Wrong input! Press any key... $'

; THIS LINE IS USED TO OVERWIRTE A LINE TO CLEAN THE AREA

EMP DB ' $'

;===============================================================

.CODE

MAIN PROC

MOV AX, @DATA

MOV DS, AX

; --------- DISPLAY THE TITLE SCREEN ---------

TITLESCREEN:

; --------------- LOGO DISPLAY START------------------------

; SET CURSOR

MOV AH, 2

MOV BH, 0

MOV DH, 6

MOV DL, 14

INT 10H

PRINT T1

; SET CURSOR

MOV AH, 2

MOV DH, 7

MOV DL, 14

INT 10H

PRINT T2

; SET CURSOR

MOV AH, 2

MOV DH, 8

MOV DL, 14

INT 10H

PRINT T3

; SET CURSOR

MOV AH, 2

MOV DH, 9

MOV DL, 14

INT 10H

PRINT T2

; SET CURSOR

MOV AH, 2

MOV DH, 10

MOV DL, 14

INT 10H

PRINT T4

;----------------- LOGO DISPLAY END -----------------

; SET CURSOR

MOV AH, 2

MOV DH, 12

MOV DL, 22

INT 10H

; SET CURSOR

MOV AH, 2

MOV DH, 13

MOV DL, 24

INT 10H

PRINT PTC ; PRESS ANY KEY

INPUT

; CLEAR SCREEN

MOV AX,0600H

MOV BH,07H

MOV CX,0000H

MOV DX,184FH

INT 10H

JMP RULES

; ----------- DISPLAY GAME RULES --------------

RULES:

; SET CURSOR

MOV AH, 2

MOV BH, 0

MOV DH, 6

MOV DL, 7

INT 10H

PRINT R

; SET CURSOR

MOV AH, 2

MOV DH, 7

MOV DL, 7

INT 10H

PRINT R1 ; RULE 1

; SET CURSOR

MOV AH, 2

MOV DH, 8

MOV DL, 7

INT 10H

PRINT R2 ; RULE 2

; SET CURSOR

MOV AH, 2

MOV DH, 9

MOV DL, 7

INT 10H

PRINT R3 ; RULE 3

; SET CURSOR

MOV AH, 2

MOV DH, 10

MOV DL, 7

INT 10H

PRINT R4 ; RULE 4

; SET CURSOR

MOV AH, 2

MOV DH, 11

MOV DL, 7

INT 10H

PRINT R5 ; RULE 5

; SET CURSOR

MOV AH, 2

MOV DH, 12

MOV DL, 7

INT 10H

PRINT R6

; SET CURSOR

MOV AH, 2

MOV DH, 13

MOV DL, 7

INT 10H

PRINT R7

; SET CURSOR

MOV AH, 2

MOV DH, 15

MOV DL, 7

INT 10H

PRINT PTC ; PRESS ANY KEY

INPUT ;INPUT WIHTOUT ECHO

; ---------- DISPLAY GAME RULES END ---------

; ---------- INITIALIZE ---------------------

INIT:

MOV PLAYER, 50 ; INITIALIZING ALL VARIABLES

MOV MOVES, 0

MOV DONE, 0

MOV DR, 0

MOV C1, 49

MOV C2, 50

MOV C3, 51

MOV C4, 52

MOV C5, 53

MOV C6, 54

MOV C7, 55

MOV C8, 56

MOV C9, 57

JMP PLRCHANGE

; ---------- INITIALIZATION ENDS --------------

; ------------ VICTORY ------------------------

VICTORY:

PRINT W1

PRINT PLAYER

PRINT W2

; SET CURSOR

MOV AH, 2

MOV DH, 17

MOV DL, 28

INT 10H

PRINT PTC ; PRESS ANY KEY

INPUT

JMP TRYAGAIN

; ------------ DRAW ------------

DRAW:

PRINT DRW

; SET CURSOR

MOV AH, 2

MOV DH, 17

MOV DL, 28

INT 10H

PRINT PTC ; PRESS ANY KEY

INPUT

JMP TRYAGAIN

; ------------ CHECK IF WINNING CONDITION IS MET -----------

CHECK: ; THERE ARE 8 POSSIBLE WINNING COMBINATIONS

CHECK1: ; CHECKING 1, 2, 3

MOV AL, C1

MOV BL, C2

MOV CL, C3

CMP AL, BL

JNZ CHECK2

CMP BL, CL

JNZ CHECK2

MOV DONE, 1

JMP BOARD

CHECK2: ; CHECKING 4, 5, 6

MOV AL, C4

MOV BL, C5

MOV CL, C6

CMP AL, BL

JNZ CHECK3

CMP BL, CL

JNZ CHECK3

MOV DONE, 1

JMP BOARD

CHECK3: ; CHECKING 7, 8, 9

MOV AL, C4

MOV BL, C5

MOV CL, C6

CMP AL, BL

JNZ CHECK4

CMP BL, CL

JNZ CHECK4

MOV DONE, 1

JMP BOARD

CHECK4: ; CHECKING 1, 4, 7

MOV AL, C1

MOV BL, C4

MOV CL, C7

CMP AL, BL

JNZ CHECK5

CMP BL, CL

JNZ CHECK5

MOV DONE, 1

JMP BOARD

CHECK5: ; CHECKING 2, 5, 8

MOV AL, C2

MOV BL, C5

MOV CL, C8

CMP AL, BL

JNZ CHECK6

CMP BL, CL

JNZ CHECK6

MOV DONE, 1

JMP BOARD

CHECK6: ; CHECKING 3, 6, 9

MOV AL, C3

MOV BL, C6

MOV CL, C9

CMP AL, BL

JNZ CHECK7

CMP BL, CL

JNZ CHECK7

MOV DONE, 1

JMP BOARD

CHECK7: ; CHECKING 1, 5, 9

MOV AL, C1

MOV BL, C5

MOV CL, C9

CMP AL, BL

JNZ CHECK8

CMP BL, CL

JNZ CHECK8

MOV DONE, 1

JMP BOARD

CHECK8: ; CHECKING 3, 5, 7

MOV AL, C3

MOV BL, C5

MOV CL, C7

CMP AL, BL

JNZ DRAWCHECK

CMP BL, CL

JNZ DRAWCHECK

MOV DONE, 1

JMP BOARD

DRAWCHECK:

MOV AL, MOVES

CMP AL, 9

JB PLRCHANGE

MOV DR, 1

JMP BOARD

JMP EXIT

; ------------ PLAYER ----------

PLRCHANGE:

CMP PLAYER, 49

JZ P2

CMP PLAYER, 50

JZ P1

P1:

MOV PLAYER, 49

MOV CUR, 88

JMP BOARD

P2:

MOV PLAYER, 50

MOV CUR, 79

JMP BOARD

; ------------- BOARD -------------

BOARD:

; CLEAR SCREEN

MOV AX,0600H

MOV BH,07H

MOV CX,0000H

MOV DX,184FH

INT 10H

; SET CURSOR

MOV AH, 2

MOV BH, 0

MOV DH, 6

MOV DL, 30

INT 10H

PRINT L1

; SET CURSOR

MOV AH, 2

MOV DH, 7

MOV DL, 30

INT 10H

MOV AH, 2

MOV DL, 32

INT 21H

; --------------------------------

; CELL 1

PRINT C1

PRINT N1

; CELL 2

PRINT C2

PRINT N1

; CELL 3

PRINT C3

; ---------------------------------

; SET CURSOR

MOV AH, 2

MOV DH, 8

MOV DL, 30

INT 10H

PRINT L2

; SET CURSOR

MOV AH, 2

MOV DH, 9

MOV DL, 30

INT 10H

PRINT L1

; SET CURSOR

MOV AH, 2

MOV DH, 10

MOV DL, 30

INT 10H

MOV AH, 2

MOV DL, 32

INT 21H

; --------------------------------

; CELL 4

PRINT C4

PRINT N1

; CELL 5

PRINT C5

PRINT N1

; CELL 6

PRINT C6

; ---------------------------------

; SET CURSOR

MOV AH, 2

MOV DH, 11

MOV DL, 30

INT 10H

PRINT L1

; SET CURSOR

MOV AH, 2

MOV DH, 12

MOV DL, 30

INT 10H

PRINT L2

; SET CURSOR

MOV AH, 2

MOV DH, 13

MOV DL, 30

INT 10H

PRINT L1

; SET CURSOR

MOV AH, 2

MOV DH, 14

MOV DL, 30

INT 10H

MOV AH, 2

MOV DL, 32

INT 21H

; --------------------------------

; CELL 4

PRINT C7

PRINT N1

; CELL 5

PRINT C8

PRINT N1

; CELL 6

PRINT C9

; ---------------------------------

; SET CURSOR

MOV AH, 2

MOV DH, 15

MOV DL, 30

INT 10H

PRINT L1

; SET CURSOR

MOV AH, 2

MOV DH, 16

MOV DL, 20

INT 10H

CMP DONE, 1

JZ VICTORY

CMP DR, 1

JZ DRAW

; ------------ END OF BOARD -------

; ------------ INPUT --------------

INPUT:

PRINT W1

MOV AH, 2

MOV DL, PLAYER

INT 21H

CMP PLAYER, 49

JZ PL1

PRINT PC2

JMP TAKEINPUT

PL1:

PRINT PC1

TAKEINPUT:

PRINT INP

INECHO

INC MOVES ; INCREMENTING MOVES COUNTER BY 1

MOV BL, AL

SUB BL, 48

MOV CL, CUR

; CHECKING IF INPUT IS BETWEEN 1-9

CMP BL, 1

JZ C1U

CMP BL, 2

JZ C2U

CMP BL, 3

JZ C3U

CMP BL, 4

JZ C4U

CMP BL, 5

JZ C5U

CMP BL, 6

JZ C6U

CMP BL, 7

JZ C7U

CMP BL, 8

JZ C8U

CMP BL, 9

JZ C9U

;---------------------------------

; IF INPUT IS INVALID

DEC MOVES ; DECREMENTING MOVES BY 1, SINCE IT WAS INVALID

; SET CURSOR

MOV AH, 2

MOV DH, 16

MOV DL, 20

INT 10H

PRINT WI ; WRONG INPUT MESSAGE

INPUT

; SET CURSOR

MOV AH, 2

MOV DH, 16

MOV DL, 20

INT 10H

PRINT EMP ; CLEARING THAT LINE

; SET CURSOR

MOV AH, 2

MOV DH, 16

MOV DL, 20

INT 10H

JMP INPUT

TAKEN:

DEC MOVES

; SET CURSOR

MOV AH, 2

MOV DH, 16

MOV DL, 20

INT 10H

PRINT TKN ; DISPLAY THAT THE CELL IS TAKEN

INPUT

; SET CURSOR

MOV AH, 2

MOV DH, 16

MOV DL, 20

INT 10H

PRINT EMP ; EMPTY LINE TO OVERWRITE ANOTHER LINE TO CLEAN THE SPACE

; SET CURSOR

MOV AH, 2

MOV DH, 16

MOV DL, 20

INT 10H

JMP INPUT

; ADJUST

; SETTING BOARD POSITION AS INPUT MARK

C1U:

CMP C1, 88 ; CHECKING IF THE CELL IS ALREADY 'X'

JZ TAKEN

CMP C1, 79 ; CHECKING IF THE CELL IS ALREADY 'O'

JZ TAKEN

MOV C1, CL

JMP CHECK

C2U:

CMP C2, 88 ; CHECKING IF THE CELL IS ALREADY 'X'

JZ TAKEN

CMP C2, 79 ; CHECKING IF THE CELL IS ALREADY 'O'

JZ TAKEN

MOV C2, CL

JMP CHECK

C3U:

CMP C3, 88 ; CHECKING IF THE CELL IS ALREADY 'X'

JZ TAKEN

CMP C3, 79 ; CHECKING IF THE CELL IS ALREADY 'O'

JZ TAKEN

MOV C3, CL

JMP CHECK

C4U:

CMP C4, 88 ; CHECKING IF THE CELL IS ALREADY 'X'

JZ TAKEN

CMP C4, 79 ; CHECKING IF THE CELL IS ALREADY 'O'

JZ TAKEN

MOV C4, CL

JMP CHECK

C5U:

CMP C5, 88 ; CHECKING IF THE CELL IS ALREADY 'X'

JZ TAKEN

CMP C5, 79 ; CHECKING IF THE CELL IS ALREADY 'O'

JZ TAKEN

MOV C5, CL

JMP CHECK

C6U:

CMP C6, 88 ; CHECKING IF THE CELL IS ALREADY 'X'

JZ TAKEN

CMP C6, 79 ; CHECKING IF THE CELL IS ALREADY 'O'

JZ TAKEN

MOV C6, CL

JMP CHECK

C7U:

CMP C7, 88 ; CHECKING IF THE CELL IS ALREADY 'X'

JZ TAKEN

CMP C7, 79 ; CHECKING IF THE CELL IS ALREADY 'O'

JZ TAKEN

MOV C7, CL

JMP CHECK

C8U:

CMP C8, 88 ; CHECKING IF THE CELL IS ALREADY 'X'

JZ TAKEN

CMP C8, 79 ; CHECKING IF THE CELL IS ALREADY 'O'

JZ TAKEN

MOV C8, CL

JMP CHECK

C9U:

CMP C9, 88 ; CHECKING IF THE CELL IS ALREADY 'X'

JZ TAKEN

CMP C9, 79 ; CHECKING IF THE CELL IS ALREADY 'O'

JZ TAKEN

MOV C9, CL

JMP CHECK

; --------------------------------

; ----------- TRY AGAIN -----------

TRYAGAIN:

; CLEAR SCREEN

MOV AX,0600H

MOV BH,07H

MOV CX,0000H

MOV DX,184FH

INT 10H

; SET CURSOR

MOV AH, 2

MOV BH, 0

MOV DH, 10

MOV DL, 24

INT 10H

PRINT TRA ; TRY AGAIN PROMPT

INECHO

CMP AL, 121 ; CHECK IF INPUT IS 'y'

JZ INIT

CMP AL, 89 ; CHECK IF INPUT IS 'Y'

JZ INIT

; IF INPUT IS 'Y'/'y' THEN REPEAT THE GAME

CMP AL, 110 ; CHECK IF INPUT IS 'n'

JZ EXIT

CMP AL, 78 ; CHECK IF INPUT IS 'N'

JZ EXIT

; IF INPUT IS 'N'/'n' THEN EXIT THE GAME

; IF INPUT IS INVALID

; SET CURSOR

MOV AH, 2

MOV BH, 0

MOV DH, 10

MOV DL, 24

INT 10H

PRINT WI ; WRONG INPUT MESSAGE

INPUT

; SET CURSOR

MOV AH, 2

MOV BH, 0

MOV DH, 10

MOV DL, 24

INT 10H

PRINT EMP ; EMPTY LINE TO OVERWRITE ANOTHER LINE TO CLEAN THE SPACE

JMP TRYAGAIN ; PROMPT THE TRY AGAIN

; ----------- END OF INPUT --------

EXIT:

MOV AH, 4CH

INT 21H

MAIN ENDP

END MAIN

## DEBUG LOG

