2021

Project Report

# Group Members SUBMITTED TO:

SUMAYYA KHALID (02-131192-057) SIR REHAN BAIG

ADIL WAHEED (02-131192-082)

RABYA ESSANI (02-131192-081)

ABDULLAH IQBAL (02-131192-076)

**TIC TAC TOE GAME**

**COMPUTER ARCHITECTURE & ORGANIZATION**

**22-01-2021**

BSE -3A

BAHRIA UNIVERSITY

(KARACHI CAMPUS)

Table of Contents

[Table of Figures 2](#_Toc534082437)

[Abstract: 3](#_Toc534082438)

[Code and Explanation: 4](#_Toc534082439)

[Step 01: 4](#_Toc534082440)

[Step 02: 5](#_Toc534082441)

[Step 03: 7](#_Toc534082442)

[Step 04: 11](#_Toc534082443)

[Interfaces: 13](#_Toc534082444)

# Table of Figures

[Figure 1Interface 1 13](#_Toc534082393)

[Figure 2Interface 2 14](#_Toc534082394)

# Abstract:

This is a Multiplayer game having 3 rows and 3 columns. A player must tick or cross in the boxes of a game by his turn. There are 8 possibilities of winning a game for e.g: If player has towin, he must have to tick in consecutive 3 boxes horizontally in rows, vertically in columns or both diagonally. If any of the player successfully made a3 number line then he/she will be declared as the winner.Each player name is mentioned in each turn. If the game is over then it will ask for another game or exit. This game is aConsole based game.

# Code and Explanation:

## Step 01:

In this step we are defining different labels which will be used in the game for their particular purpose and we are initializing 9 registers ($t1…$t9) for allocating memory of 9 boxes of tic tack toe table. We are also saving the three labels in temporary saved register ($s1,$s2,$s3) for printing board, asking player to input move and telling them who won.

.data

board: .asciiz " 1 2 3\n1 | | \n ---+---+---\n2 | | \n ---+---+---\n3 | | \n"

askMove: .asciiz "Player insert your play (column|line):"

invalidMove: .asciiz "\*\*Invalid Move\*\*"

occupiedSpace: .asciiz "\*\*Space already occupied\*\*\n"

x: .asciiz "X"

o: .asciiz "O"

won: .asciiz "\nPlayer Won! \n"

tie: .asciiz "\nTie!!!"

gameMenu: .asciiz "\n\nChoose an option:\n[1] New Game\t[99] Quit\nOption: "

clean: .byte ' '

.text

.globl main

main:

li $t1, 0

li $t2, 0

li $t3, 0

li $t4, 0

li $t5, 0

li $t6, 0

li $t7, 0

li $t8, 0

li $t9, 0

li $s0, 0

li $s5, 0

## Step 02:

In this step we are assigning values to $s1 and creating links of printing board, assigning two players link(player X and player 0) and also creating the link in which a label ($s6) is taking input from the user.

la $s1, board

la $s2, askMove

la $s3, won

lb $a1, clean

sb $a1, 15($s1)

sb $a1, 18($s1)

sb $a1, 22($s1)

sb $a1, 40($s1)

sb $a1, 44($s1)

sb $a1, 48($s1)

sb $a1, 66($s1)

sb $a1, 70($s1)

sb $a1, 74($s1)

PrintBoard:

li $v0, 4

la $a0, board

syscall

beq $s5, 9, Tie

add $s5, $s5, 1

rem $t0, $s0, 2

add $s0, $s0, 1

bnez $t0, Player0

PlayerX:

lb $a1, x

sb $a1, 7($s2)

sb $a1, 8($s3)

j Play

Player0:

lb $a1, o

sb $a1, 7($s2)

sb $a1, 8($s3)

Play:

li $v0, 4

la $a0, askMove

syscall

li $v0, 5

syscall

move $s6, $v0

beq $s6, 11, J11

beq $s6, 21, J21

beq $s6, 31, J31

beq $s6, 12, J12

beq $s6, 22, J22

beq $s6, 32, J32

beq $s6, 13, J13

beq $s6, 23, J23

beq $s6, 33, J33

li $v0, 4

la $a0, invalidMove

syscall

j Play

## Step 03:

In this step we are just defining the rows and columns of tic tack toe table.

J11:

bnez $t1, Occupied

bnez $t0, O11

X11:

li $t1, 1

sb $a1, 14($s1)

j CheckVictory

O11:

li $t1, 2

sb $a1, 14($s1)

j CheckVictory

J21:

bnez $t2, Occupied

bnez $t0, O21

X21:

li $t2, 1

sb $a1, 18($s1)

j CheckVictory

O21:

li $t2, 2

sb $a1, 18($s1)

j CheckVictory

J31:

bnez $t3, Occupied

bnez $t0, O31

X31:

li $t3, 1

sb $a1, 22($s1)

j CheckVictory

O31:

li $t3, 2

sb $a1, 22($s1)

j CheckVictory

J12:

bnez $t4, Occupied

bnez $t0, O12

X12:

li $t4, 1

sb $a1, 40($s1)

j CheckVictory

O12:

li $t4, 2

sb $a1, 40($s1)

j CheckVictory

J22:

bnez $t5, Occupied

bnez $t0, O22

X22:

li $t5, 1

sb $a1, 44($s1)

j CheckVictory

O22:

li $t5, 2

sb $a1, 44($s1)

j CheckVictory

J32:

bnez $t6, Occupied

bnez $t0, O32

X32:

li $t6, 1

sb $a1, 48($s1)

j CheckVictory

O32:

li $t6, 2

sb $a1, 48($s1)

j CheckVictory

J13:

bnez $t7, Occupied

bnez $t0, O13

X13:

li $t7, 1

sb $a1, 66($s1)

j CheckVictory

O13:

li $t7, 2

sb $a1, 66($s1)

j CheckVictory

J23:

bnez $t8, Occupied

bnez $t0, O23

X23:

li $t8, 1

sb $a1, 70($s1)

j CheckVictory

O23:

li $t8, 2

sb $a1, 70($s1)

j CheckVictory

J33:

bnez $t9, Occupied

bnez $t0, O33

X33:

li $t9, 1

sb $a1, 74($s1)

j CheckVictory

O33:

li $t9, 2

sb $a1, 74($s1)

j CheckVictory

## Step 04:

In this step we are checking if the space is occupied or not and we are checking for victory here.

Occupied:

li $v0, 4

la $a0, occupiedSpace

syscall

j Play

CheckVictory:

and $s7, $t1, $t2

and $s7, $s7, $t3

bnez $s7, Victory

and $s7, $t4, $t5

and $s7, $s7, $t6

bnez $s7, Victory

and $s7, $t7, $t8

and $s7, $s7, $t9

bnez $s7, Victory

and $s7, $t1, $t4

and $s7, $s7, $t7

bnez $s7, Victory

and $s7, $t2, $t5

and $s7, $s7, $t8

bnez $s7, Victory

and $s7, $t3, $t6

and $s7, $s7, $t9

bnez $s7, Victory

and $s7, $t1, $t5

and $s7, $s7, $t9

bnez $s7, Victory

and $s7, $t7, $t5

and $s7, $s7, $t3

bnez $s7, Victory

j PrintBoard

Victory:

li $v0, 4

la $a0, board

syscall

li $v0, 4

la $a0, won

syscall

j MenuNewGame

Tie:

li $v0, 4

la $a0, tie

syscall

MenuNewGame:

li $v0,4

la $a0, gameMenu

syscall

li $v0,5

syscall

bne $v0, 99, main

li $v0, 10

syscall

# Interfaces:

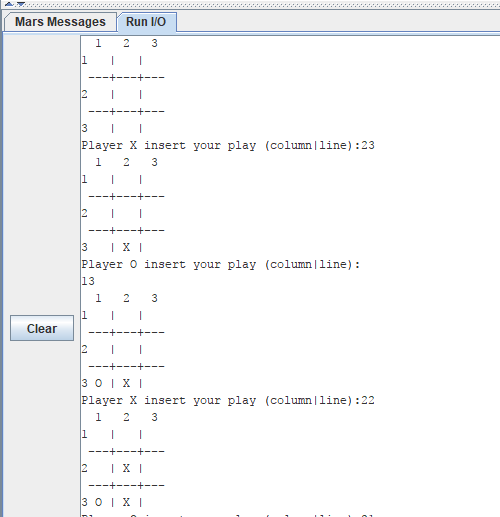


Figure Interface 1

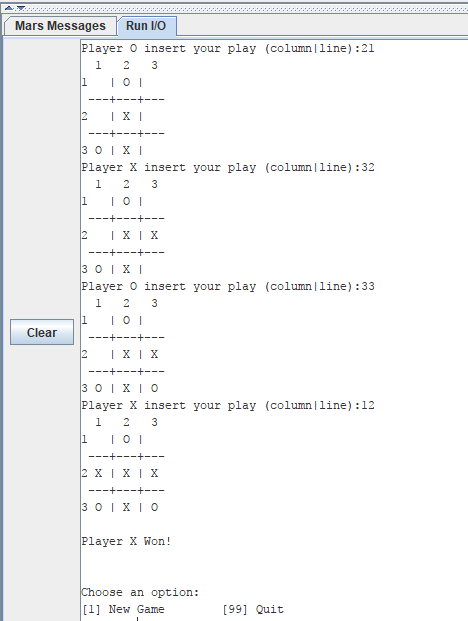


Figure Interface 2