**PROJECT REPORT**

Subject: **CAO Lab BSE 3-A**

**Submitted to:**

Sir Muhammad Rehan Baig

Project Title: **Tic tac toe game using Mips**

**Group Members :**

1. SAMI ULLAH KHAN 02-131192-071
2. Muhammad Saad Kamal 02-131192-075
3. Seemab Shafqat 02-131182-071

**Tic-tac-toe**

Tic-tac-toe (noughts and crosses) is a game for two players, X and O, who take turns marking the spaces in a 3×3 grid. The player who succeeds in placing three of their marks in a diagonal, horizontal, or vertical row is the winner. It is a solved game with a forced draw assuming best play from both players.

**Introduction:**

In order to win the game, a player must place three of their marks in a horizontal, vertical, or diagonal row.

The following example game is won by the first player, X:

Game of Tic-tac-toe, won by X

Players soon discover that the best play from both parties leads to a draw. Hence, tic-tac-toe is most often played by young children, who often have not yet discovered the optimal strategy.

Incidence structure for tic-tac-toe.

Because of the simplicity of tic-tac-toe, it is often used as a pedagogical tool for teaching the concepts of good sportsmanship and the branch of artificial intelligence that deals with the searching of game trees. It is straightforward to write a computer program to play tic-tac-toe perfectly or to enumerate the 765 essentially different positions (the state space complexity) or the 26,830 possible games up to rotations and reflections (the game tree complexity) on this space. If played optimally by both players, the game always ends in a draw, making tic-tac-toe a futile game.

Tic-tac-toe is the (3,3,3)-game.[3] Harary's generalized tic-tac-toe is an even broader generalization of tic-tac-toe. It can also be generalized as a nd game. Tic-tac-toe is the game where n equals 3 and d equals 2.[4] It can be generalized even further by playing on an arbitrary incidence structure, where rows are lines and cells are points. Tic-tac-toe is the game given by the incidence structure shown to the right, consisting of nine points, three horizontal lines, three vertical lines, and two diagonal lines, each line consisting of at least three points.

**#Code**

.data

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

askMove: .asciiz "Player insert your play (Column|Row):"

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

occupiedSpace: .asciiz "\*\*Space filled already\*\*\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

la $s1, board

la $s2, askMove

la $s3, won

lb $a1, clean

sb $a1, 14($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

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

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