

8
3 rows
3 col-4

	a	b	c
1	0	1	2
2	3	4	5
3	6	7	8

array win [0, 4, 8]
[3, 4, 5]

Pseudo Code Tic-Tac-Toe

10/5

State

25

playerTurn = 'x' or 'o'

- track moves?

(8) win Conditions:

*use int os to 10 event to each event to each

- change player?

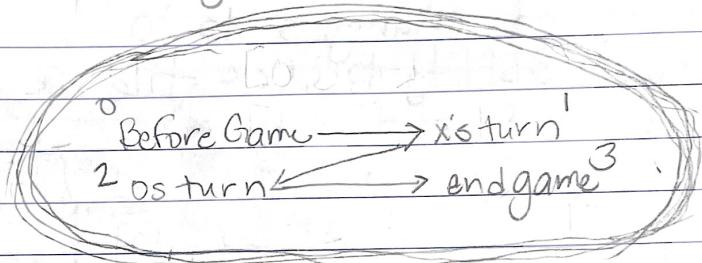
boardState = [] length=9

- make checked box inactive?

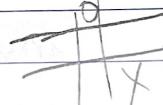
gameActive =

winner:

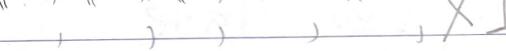
players:



1. who's turn? X



2. X plays → [, , , , , , , , , X]



3. O's turn

4. O plays → [, O, , , , , , , X]

→ Click event toggles from X to O

functions

checkWinConditions ['0', '0', '0', ' ', 'x', ...] ChangePlayer()

checkWinCond()

gameResult()

init()

drawBoard()

setState()

eventHandler, handleClick()

updateSpace()

finishGame()

resetState()

makeMove()

Bootstrap

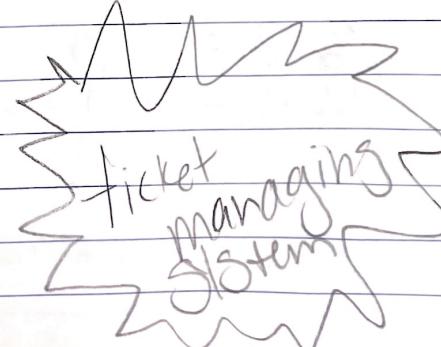
board

3 rows

3 col-4

3 '0' 'x' 'o'

?



on click
in HTML
on each tile

init()

- start game button

↳ onclick drawBoard() ← also use for reset

↳ get gameState()

render HTML

First click is 'X' (player one)

tile clicked gets an 'X' - update DOM(state) - Check win state

Player turn changes to 'O'

Remove ability to click tile selected (event listener)

init()

drawBoard() - game says it's X's turn (player 1 go!)

index /

boardState

[0, 1, 2]

3 4 5
6 7 8

Player Turn
Player State

Player turn
Change away

turnCount

if turnCount % 2 == 0 → player 2

if turnCount % 2 == 1 → player 1

= 'setting'
= == 'checking'

playerTurn = 1 → Player 1 ('X')
PlayerTurn = 2 → player 2 ('O')

divs-addEL

let state = {
boardState: []},

winConditions: [[]],
3

1 X 4 O 3 X
↓ 5 X 6 O
2 O 7 X

let state =
player = X
boardState = [empty]
playerTurn = 1
winCond: [C1, C2, ...]

reset button

- play = X ✓
- function O ✓
- boardState ✓
- checkWin ✓
- changePlayer ✓
- CL - play 2 go! ✓
- playerTurn = 2 ✓
- play = O

1. Button functions 0-8

2. check winCond.

3. ChangePlayer

* Press Space
- space gets 'X' or 'O'
- functions run
- function #()
- checkWin() → words show
- changePlay()
- disabled space