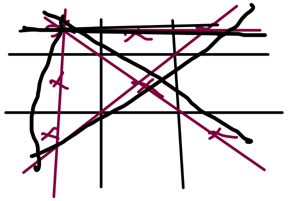
PROJECT-01

TIC TAC TOE

$\boxed{x}$ $\boxed{0}$

players
↳ p1
↳ p2



→ 3x3

→ Rowwise
→ Columnwise
→ Diagonalwise
→ Anti diagonalwise

o	o	o
x		x
	x	

p1 → Ram → x
p2 → Ravi → 0

X	X	O
O	X	X
X	O	O

⟹ Draw

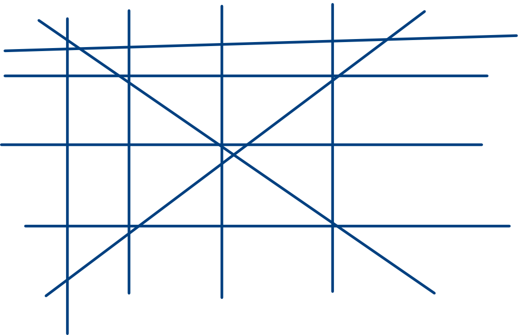
p1 → Ram → x
p2 → Ravi → 0

O		
O	X	X
O		

↳ Ravi won

X	X	X
	O	
O		

→ Ram won



↳ Design Tic Tac Toe

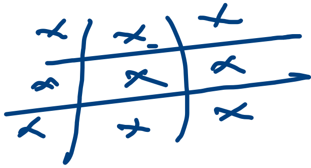
↳ Requirements → 1 board → define → size of board (n x n)

→ 2 Players (A, B)

→ Rules

↳ if any row, column, diagonal, anti diagonal is filled with same symbol

↳ considered as winner



→ Outcomes

↳ Player 1 → winner

↳ player 2 → winner

↳ Draw

→ Board (5 mins)

↳ Board class

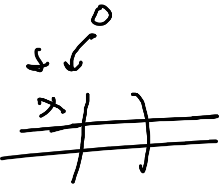
↳ int size → user

↳ character matrix

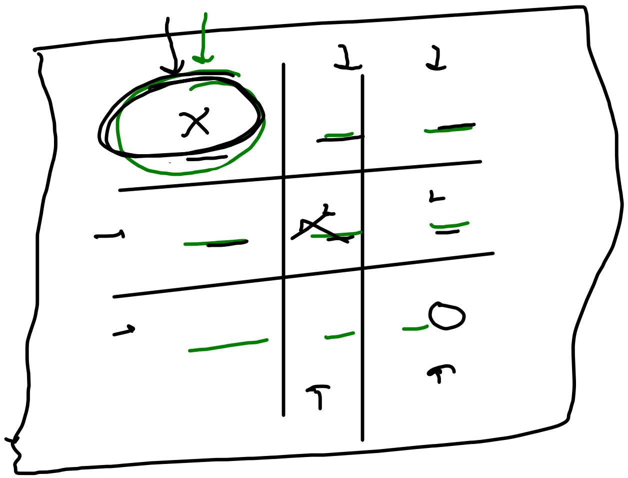
↳ default characters

↳ fn → default characters

↳ constructor



→ if(default character)



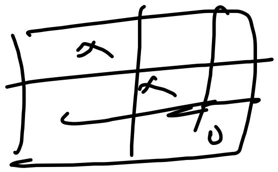
p1 →

if (⊟

p2

if (−)

↳ fn. printBoard Config



→ Player

→ player class

(5- 6) mins

↳ ① name of the player

↳ ② age →

↳ ③ address of the player

↳ ④ player contact number

↳ ⑤ player email id

↳ ⑥ player symbol

Ravi
23
UP
x — ___
R ___

↳ awhiling
initialise
↳ (constructor)

↳ fn → set
↳ setting the properties of a player

↳ fn → get → name, age, add, contact, email id, symbol
↳ purpose is to return the details of a player.

↳ set → name, symbol, contact, email id
↳ set → name, symbol

fn → get → name, age, symbol
fn → get → name, symbol

↳ validation of player details

→ Game rules

↳ Game class