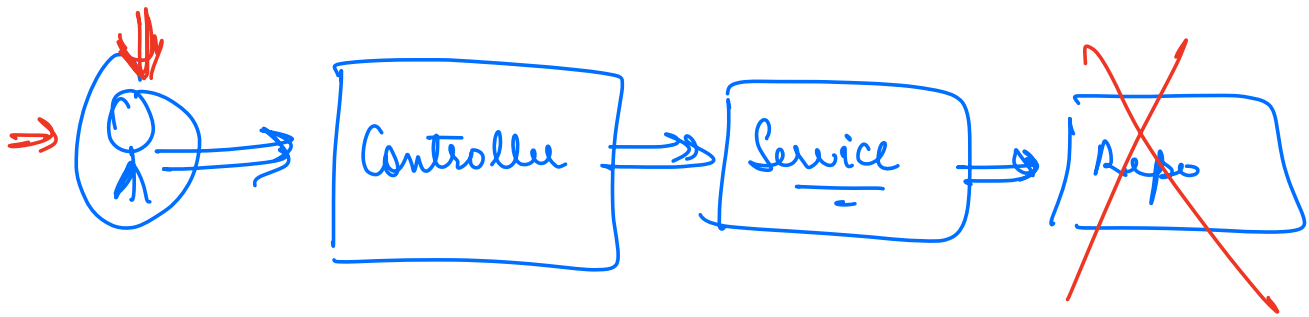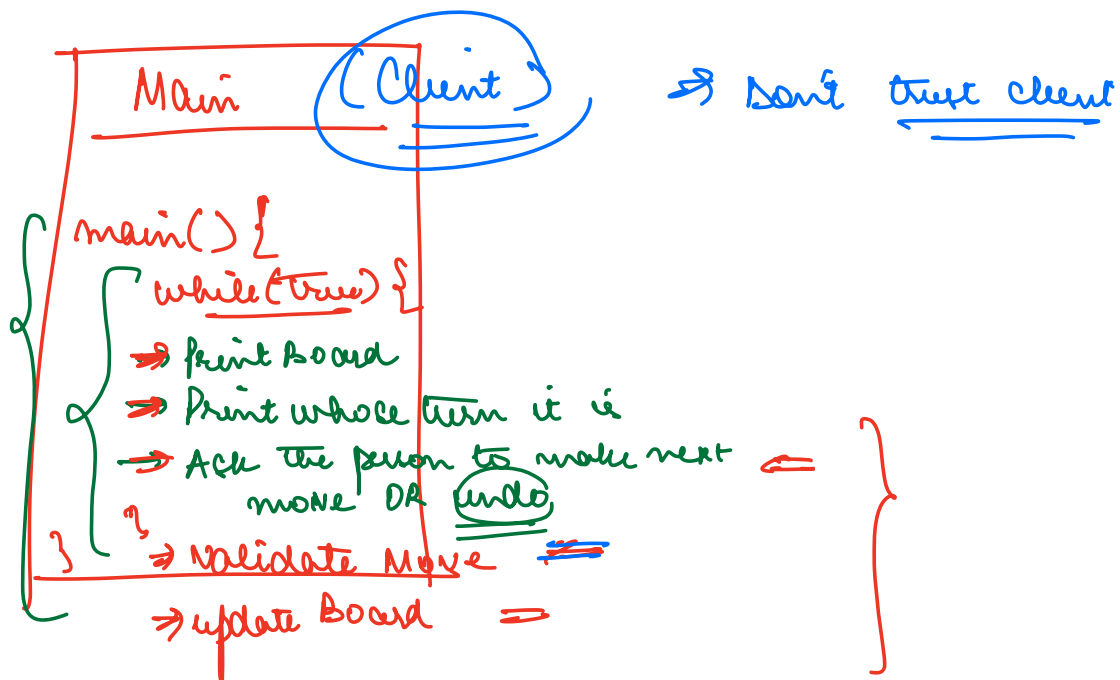Will start at 9:10

Code TicTacToe
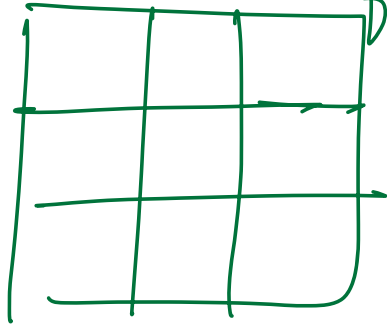

→ We are building an interactive app[n]
   → no need of Services and repo.
→ Simple desktop app[n]

Main (Client)   → Don't trust client

```
main() {
   while(true) {
   → Print Board
   → Print whose turn it is
   → Ask the person to make next
      move OR undo
   → Validate Move
   → update Board
}
```

-- Game is starting -- · -- --

-- The current state of Board is -- --
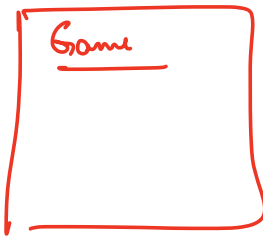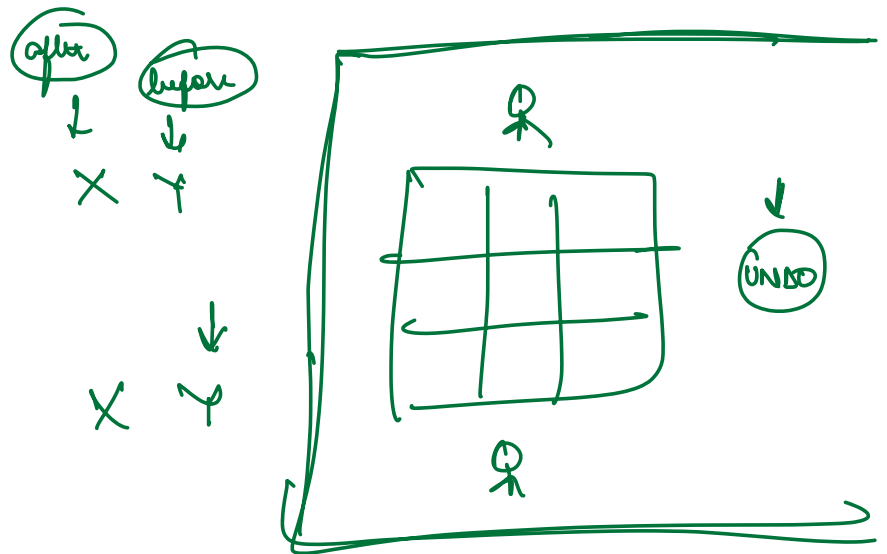

Y
undo

-- Does anyone want to undo (y/n) --------

↓ n

-- It is [X]'s turn. Please tell row and col where you want to make move

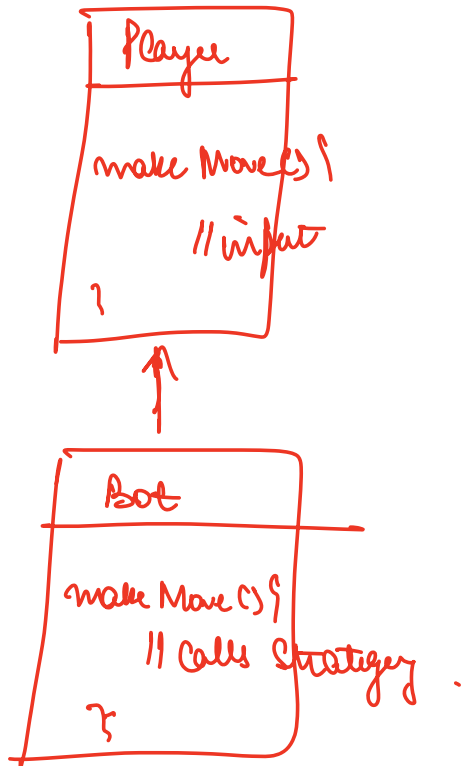⇒ take input

if bot: Bot Playing Strategy will tell row and col.

else: take input from command line

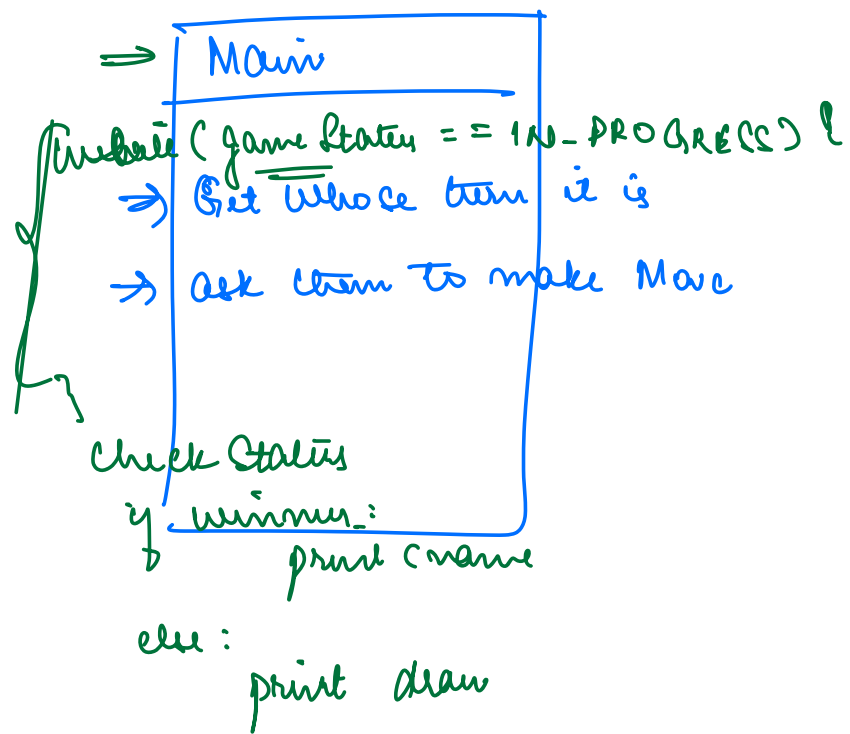Main {
    if (type ==)
    elif (type ==)
}

after
before

X
Y

X
Y

UNDO

Game

Player   currentPlayer = ____ ;

currentPlayer. make Move ( )

Player

make Move(s)
    // input
}

Bot

make Move ( ) {
    // Calls Strategy .
}

⇒ **Main**

while (game Status == IN-PROGRESS) {

⇒ Get whose turn it is

⇒ ask them to make Move

}

Check Status

if winner:

print Cname

else:

print draw

```
class  Main  {        // Client

        main() {

                Game g = _____;
                while (& g. status() == (N-PROG) {
                        g. print Board() .
                        Player p = g. get Next Player()
                        p. make Move()
                }
        }
                Player p = g. get Winner();
}
```
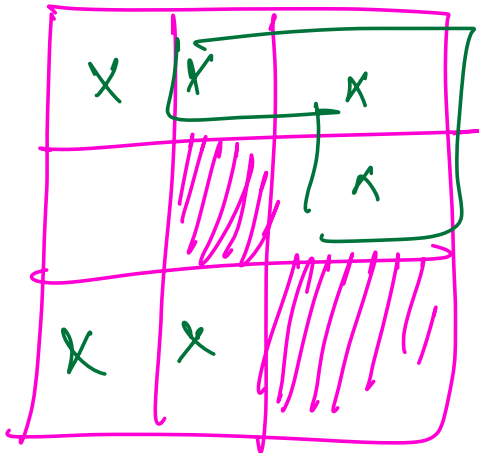

2 together winning
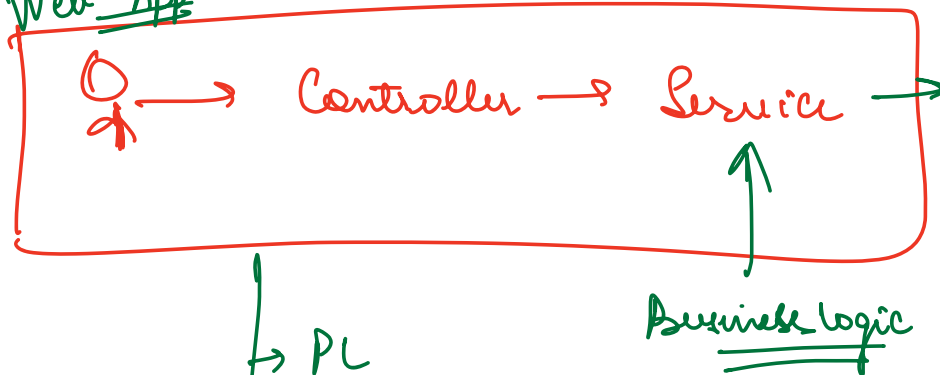Arrow winning Strategy

Client code Should be dumb.

Currently, | Client $\longrightarrow$ Models

$\Rightarrow$ Any method client is using should only be in controller

User $\longrightarrow$ Controller $\longrightarrow$ Models

Business logic

Game / Interactive App[n]

TTT

Web App[n]

User $\longrightarrow$ Controller $\longrightarrow$ Service $\longrightarrow$ Repo $\longrightarrow$ Models

Business logic

PL

BMS

S/W
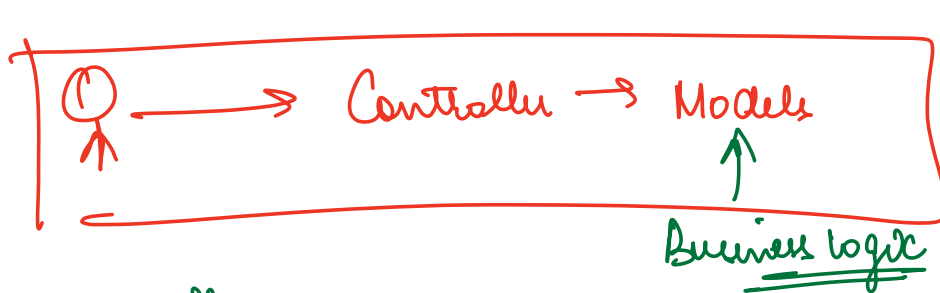
No Business logic

$\rightarrow$ Simple class with Attr

Only entity def[n]

Game Controller (gc) (1)

π {
  (gc.) make Move (g1)

  gc. make Move (g2) } T2

  gc. make Move (g3) } T3

Break till 10:18