

AGENDA

- > class diagram TTT
- > decide winner in $O(1)$
- > code

start by 9:07 PM IST

NOUNS

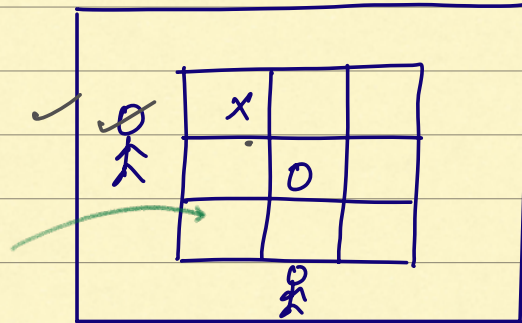
✓

VISUALIZATION

✓

✓

*] CLASS DIAGRAM:



~~Player~~

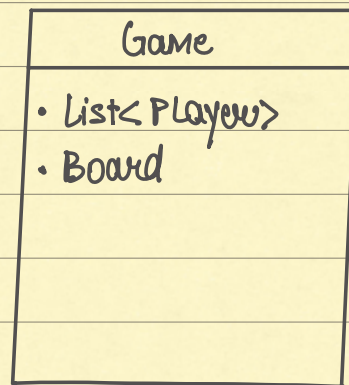
~~symbol~~

~~Board~~

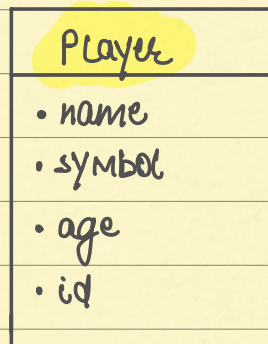
~~cell~~

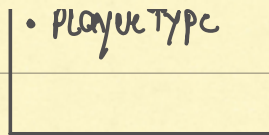
~~Game~~

①



②

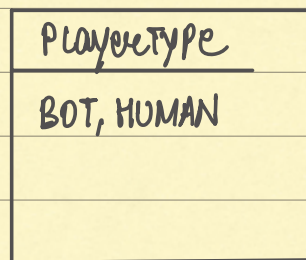




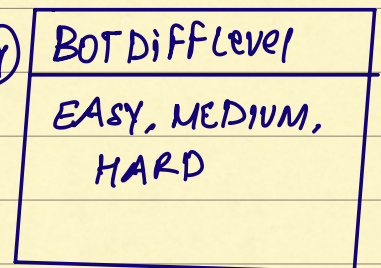
3



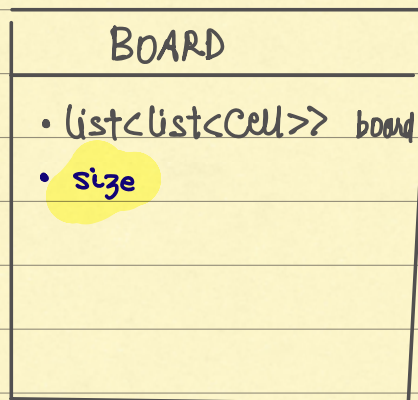
4



9

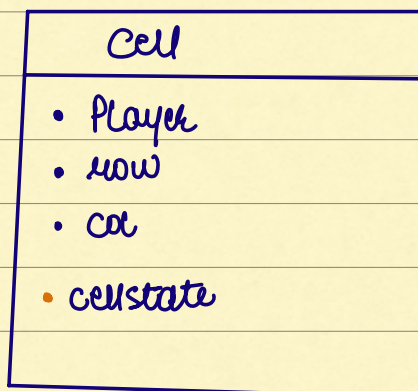


5

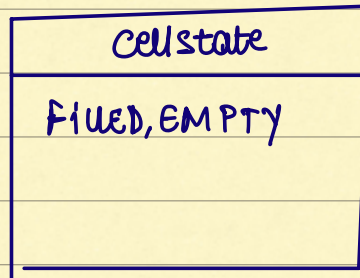


b.size(),

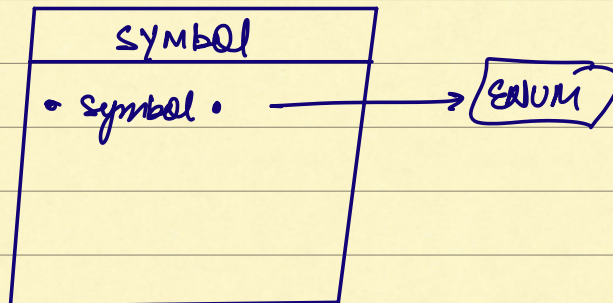
6



⑦

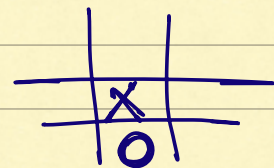
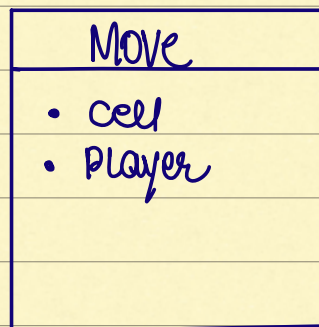


⑧

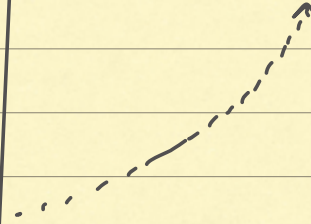
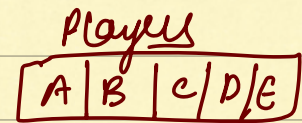
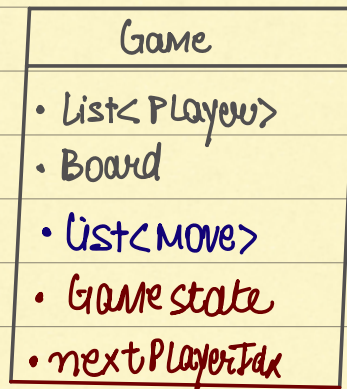


-) nextPlayer index
-) GameState
-) Bot strategy

⑨



①



②



- all classes / attributes / relations
- design Patterns..

*) DESIGN PATTERNS:

1.) Strategy

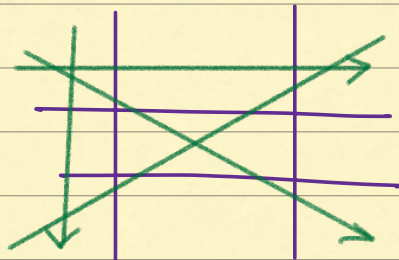
- finding Bot difficulty
- Rule winning strategy
- Player strategy



2.7 Game object \rightarrow validation pass... }
+ many attributes

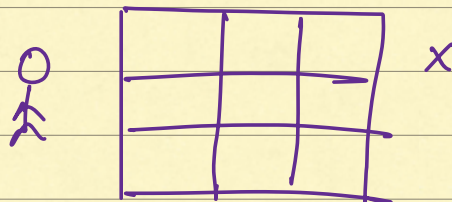
GameBuilder

*1) FINDING WINNER IN O(1) time:



1 full row	same symbol
1 full col	same "
1 full DIAG	same ..

Goal: to find winner

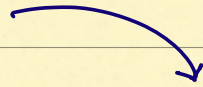




(N-1)

1.) BRUTE FORCE WAY:

$O(N^3)$



checkWinner(Board board) {

for every Player:

for every Row:

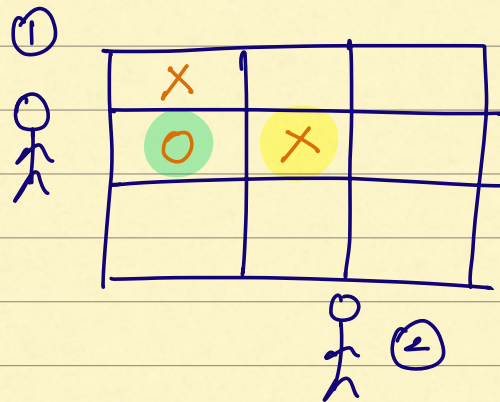
for every col:

if (all(symbol) in any Row ||
all same symbol in col ||
diag)

return true;

return false

}



* Better Approach

checkWinner(Board board, Player lastMovePlayer) {

for every Row:

for every col:

if (all (symbol) in any Row ||

all same symbol in col ||

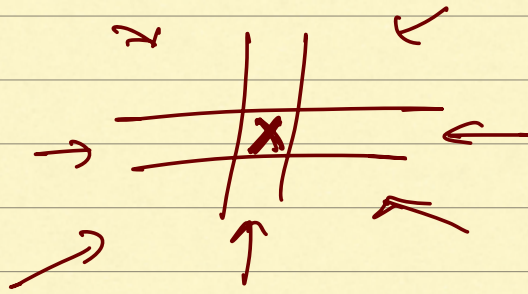
diag)

return true;

return false

✓ $O(N^2)$

* More Better Approach



checkWinner(Board board, Player lastMovePlayer) {

① for loop \rightarrow Row

② for loop \rightarrow Col

if (diagonal) {

For loop for diagonal...

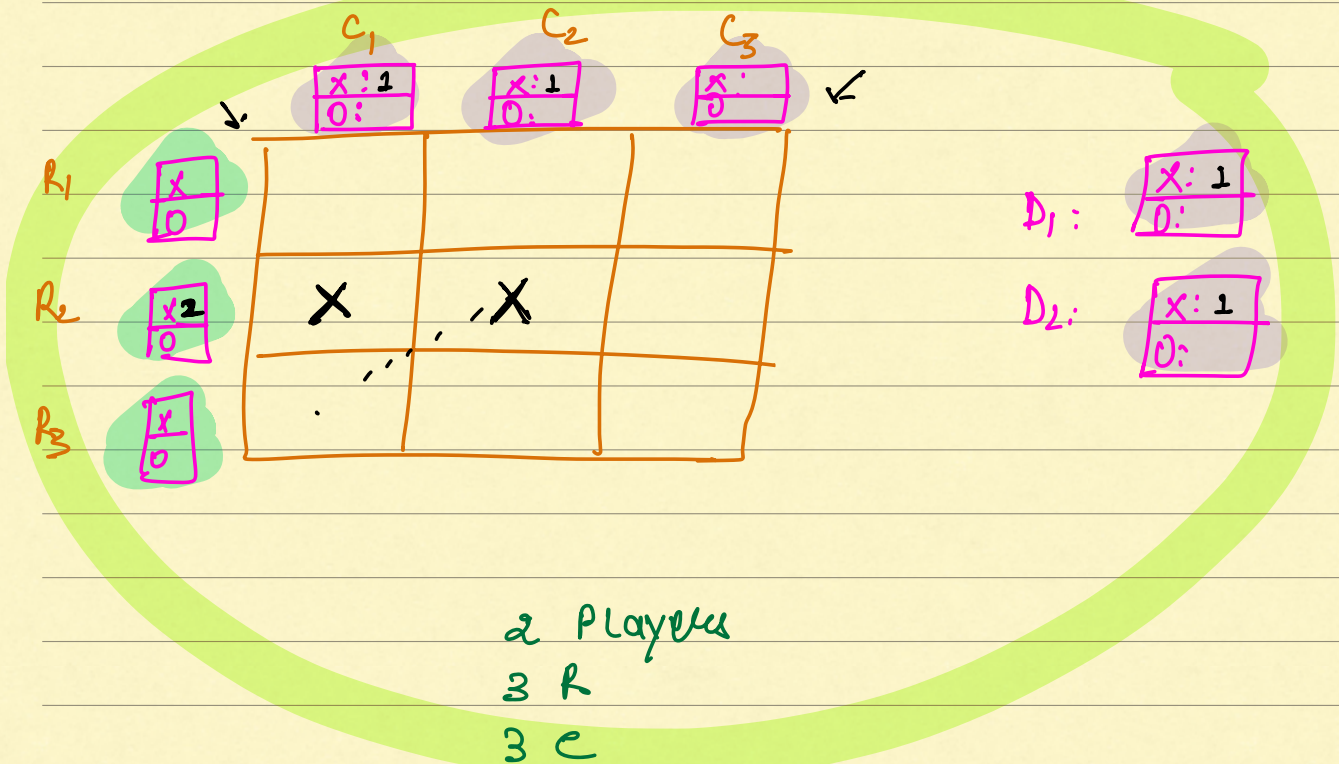
}

return false

$O(N)$

*) BEST APPROACH: $O(1)$

$O(N) \rightarrow O(1)$
space



2 Players
 3 R
 3 C

P_1
 3M R
 3M C
 2 D
8 X.

→ After Every move; Insert/ into Map...
 update