1. Good Evening

2. Lecture begins at 9:05 pm

3. Topic → Design TTT

Agenda

✓ 1. Schema Design Guidelines for Machine Coding interview

2. Project Structure Guidelines for Machine Coding interview

3. Design Tic-Tac-Toe

4. Discussion on strategies to do Undo

5. Homework

(Parking lot)

1. Schema Design Guidelines

Pen → Class Diagram

TTT → Interactive System

BMS, SW, PL → Schema Design

Step1 → Make a table for every entity class in class diagram

(Model, entity, POJO, DTO)

Step2 → Make columns for all primitive attributes in the class

(Strings & date time)

class Ticket ✓

int id ✓

Date tob ✓

List < Seat > Seats

User bookedby

int amount ✓

Ticked

id	tob	amount

Ticket : Seat

1 : m

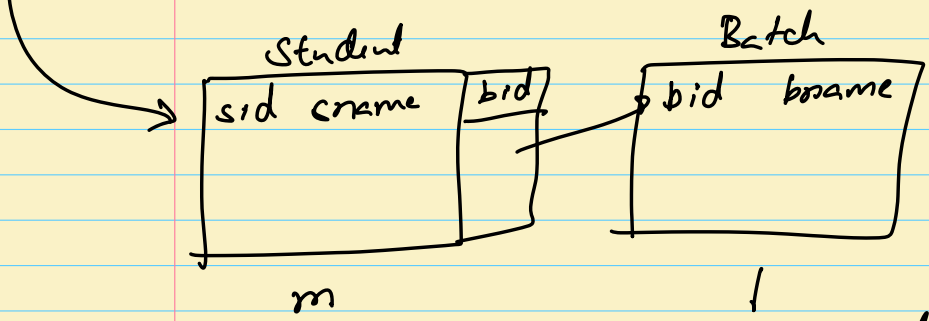
Step3 →

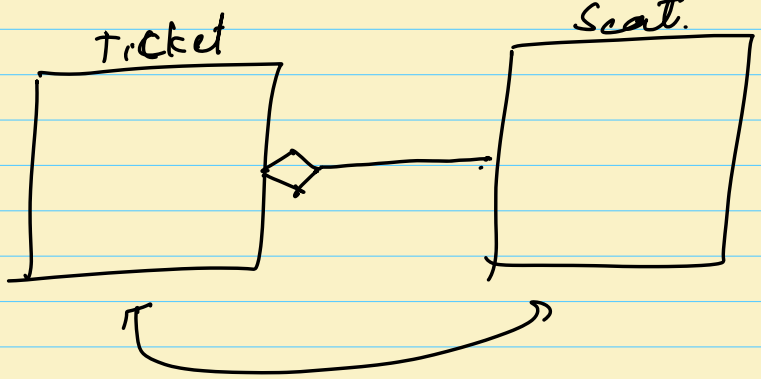
1. Identify cardinality of relation

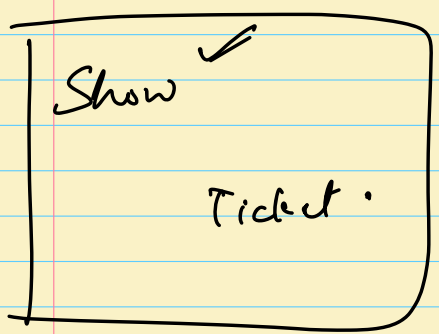
→ 1 : 1 → Add Fk on either side.

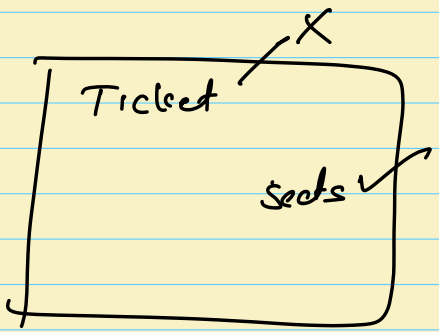
→ 1 : m or m : 1 → Add Fk on many side

→ m : m → Mapping Table





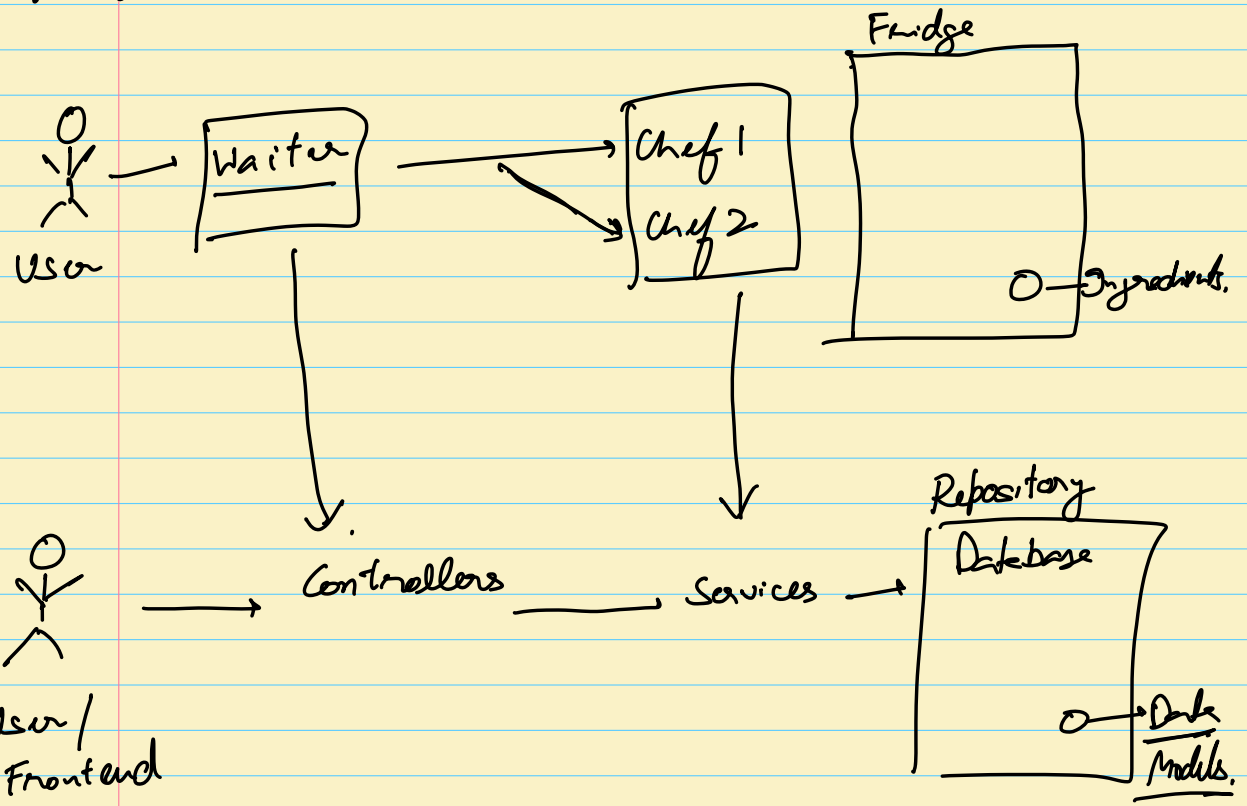




2. Project structure Guidelines

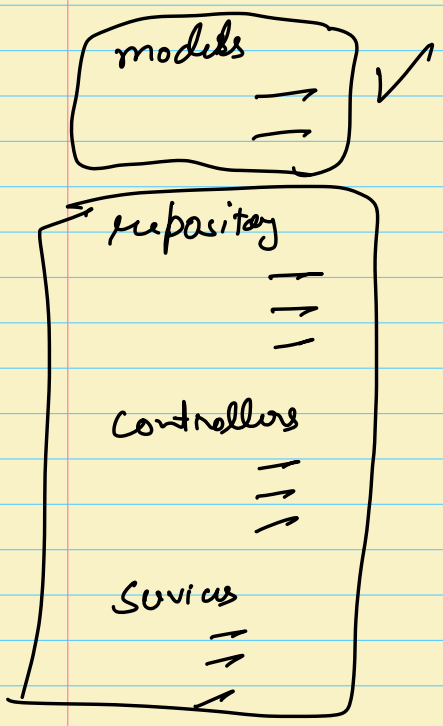
What ~~is~~ are metrics of evaluation?

1. Project structure
2. How many requirements could we complete end to end & bring it to working state.



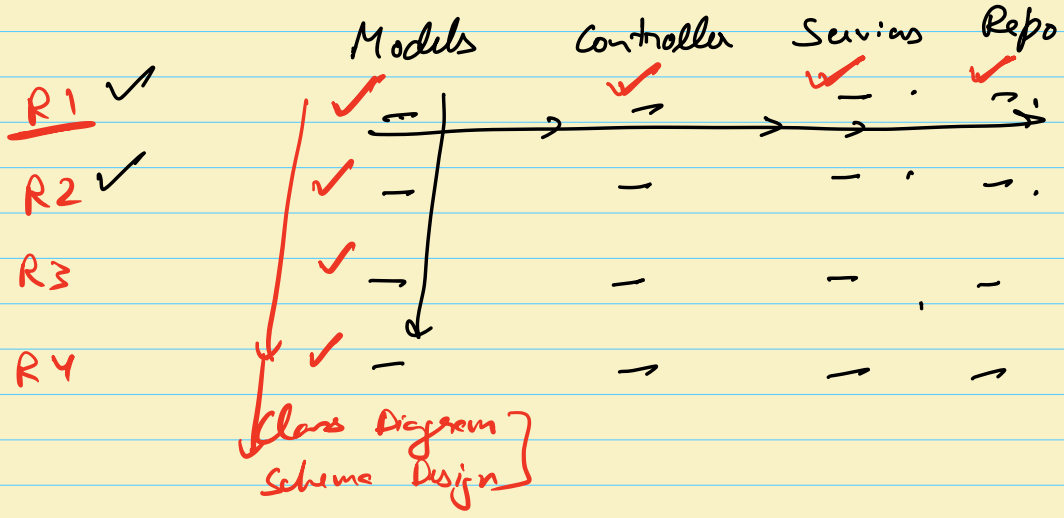
BMS, SW

1. Controller — Take request from user & fwd to services.
2. Services — Business logic
3. Repository — To speak to the database
4. Models — Entities in class diagram.



Advice

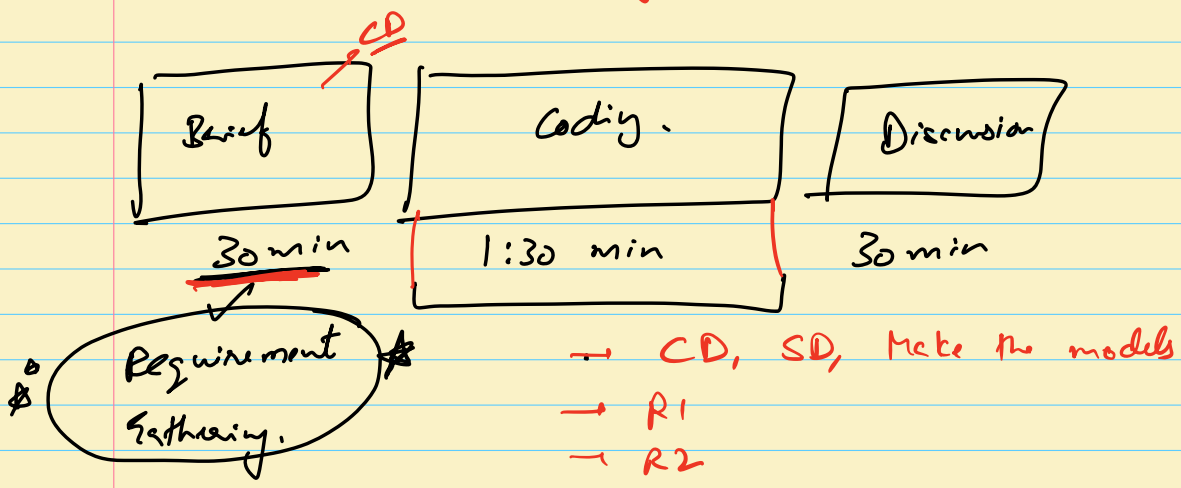
1. Class Diagram
2. Finish all the models
3. Pick requirements 1 by 1
   > finish all 4 layers for that.



→ Entity

→ Interactive applications

↘ Web APIs → Schema Design
→ Project structure



→ CD, SD, Make the models
→ R1
→ R2

1. Problem Statement : Design Splitwise

Design Tic-Tac-Toe

1. Problem Statement
2. Overview →
3. Gathering Requirements.
4. Coding
5. Discussion.

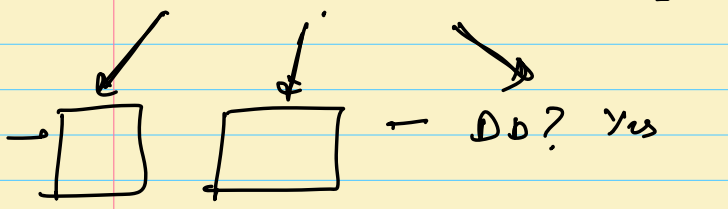
2 minutes

[Overview of problem

Aware ✓ Not aware →

Focus

1. Align with the interviewer
2. Identify type of system
   - Entity
   - Interactive System
   - Web APIs



2. Requirement Gathering

1. Size of board N*N ✓ [3*3]

2. How many players? → N-1 players. [2]

3. Each player should've a unique symbol defined at the game start.

10:33 to 10:45

4. How does the game end? → Draw
   → 1st person wins.

5. Are there more than one way to win?
   - Row ✓
   - Column
   - Diagonal
   - Corners

Bots Bots level	Undo Rewatch

6. Who starts > players play in which order?

A	B	C	D

B	D	A	C

7. Will there be bots? Yes

8. Bots could be of different levels → Easy, Medium, Difficult

strategy

- Leaderboard
- Timer
- Pause / Play, Exit

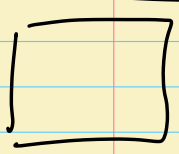
9. Undo & Reset

10. Rewatch. (or Replay)

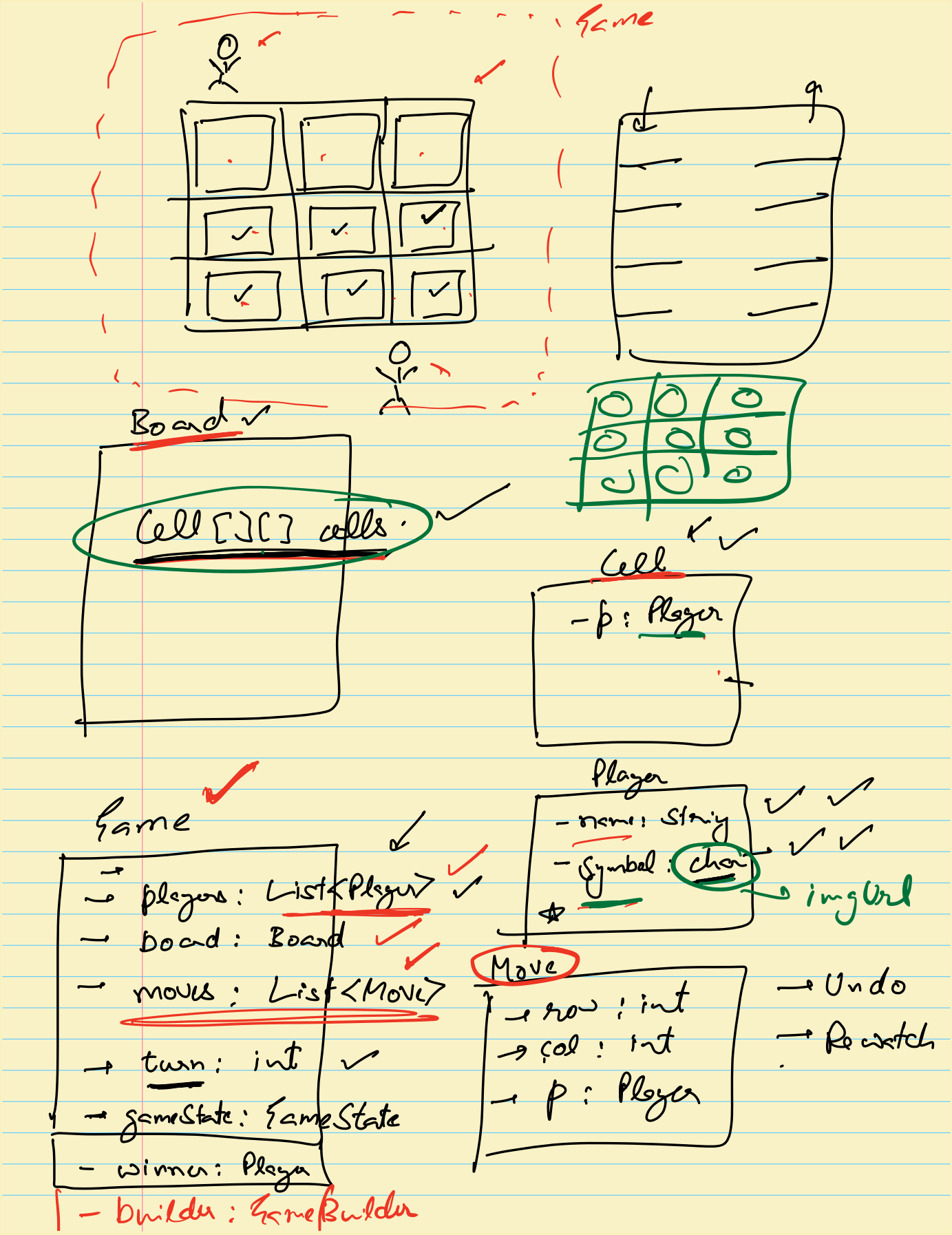
Break → 10:38 to 10:45

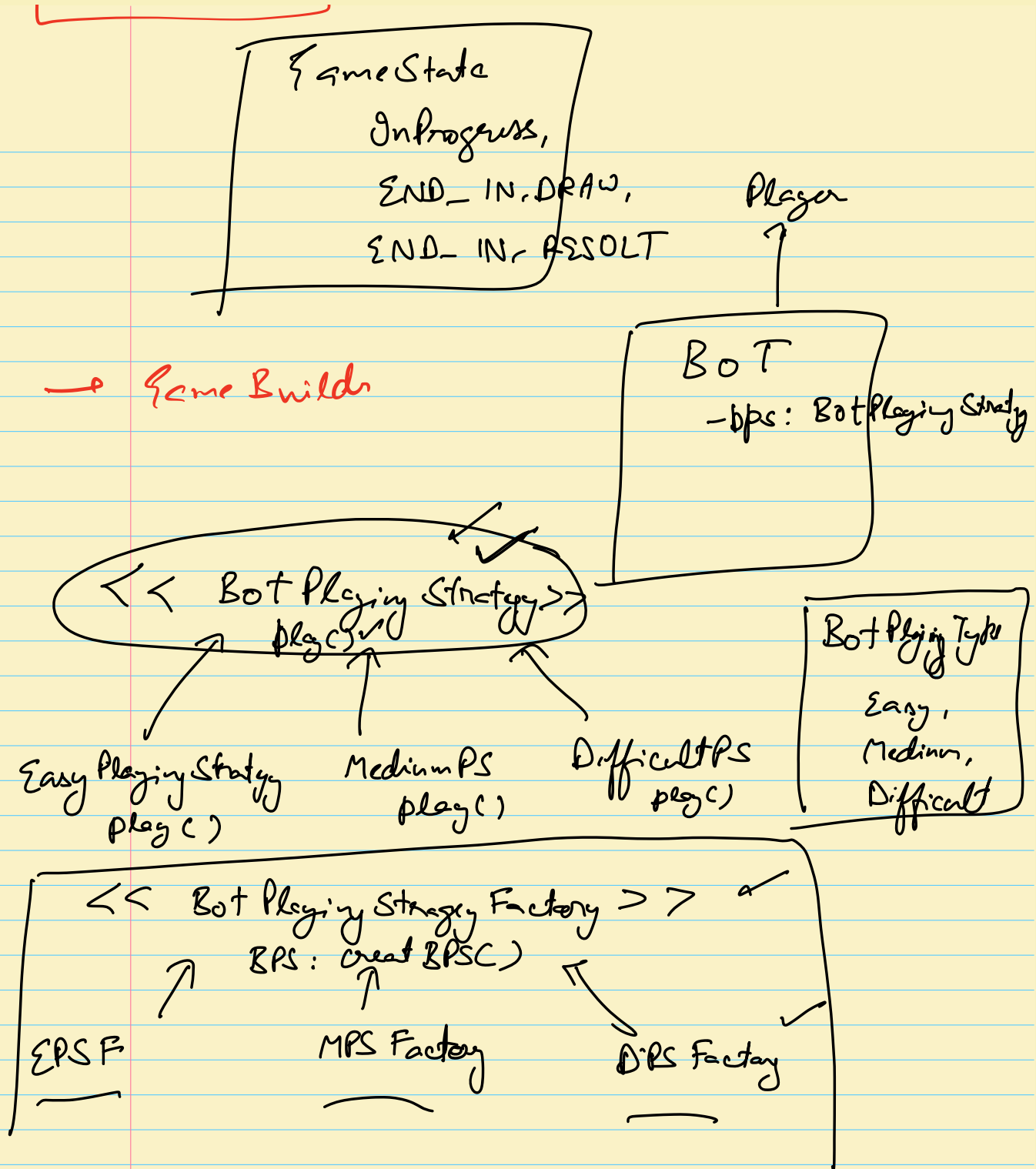
→ Class Diagram { identify all the classes }

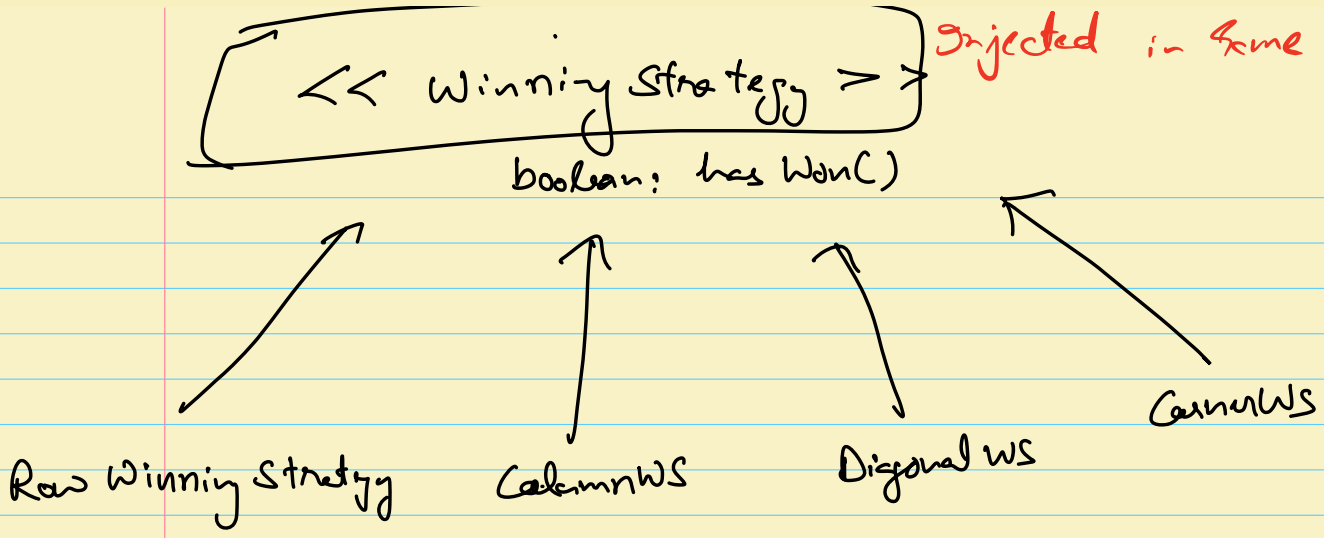
Reg → Classes

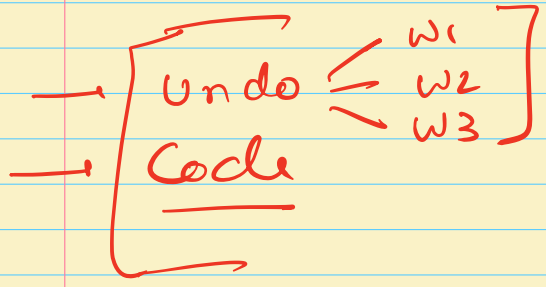


1. Nouns
2. Sketch & ~~make~~ think of user story.


Game
Board
Cell[][] cells
Cell
-p: Player
Player
-name: String
-Symbol: char
imgUrl
Game
players: List<Player>
board: Board
moves: List<Move>
turn: int
gameState: GameState
-winner: Player
-builder: GameBuilder
Move
row: int
col: int
p: Player
Undo
Restart


GameState
InProgress,
END_IN_DRAW,
END_IN_RESOLT
Player
BoT
-bps: BotPlayingStrategy
GameBuilder
<< BotPlayingStrategy >>
play()
Easy PlayingStrategy
play()
MediumPS
play()
DifficultPS
play()
BotPlayingType
Easy,
Medium,
Difficult
<< BotPlayingStrategyFactory >>
BPS: creatBPS()
EPSF
MPS Factory
DPS Factory





Homework

1. Code

2. How can you decide the winner in O(1)