

Lecture 33: Build Tic-tac-toe Game

classmate

Date _____
Page _____

Requirements

- size of the board should be scalable
- There are standard game rules and be further extendible.
- Allow app notifications for Tic-Tac-Toe moves, wins, draws, etc.

UML Design

- Creating it using top-down approach
- Start with Game Class which will be our main class.

Hum kaise develop korege using top-down approach

① Humne Game class banaya and then uske ek ek variables, class reference yaa methods banaye and then uske according sub classes banaya.

② Jaise where we play game \Rightarrow board so make board class then board mein grid hogा so usse banane ke liye a 2D array for grid and grid mein hum kya store karte hai \rightarrow create symbol

③ Aise hi hum small objects banate hain and using main class.

<<abstract>>

```

Rules
    isValidMove(Board b, r, c);
    checkWin(Board b, Symbol s);
    checkDraw(Board b);

```

<<abstract>>

```

IObserver
void update(string msg);

```

```

Game
Board board;
deque<Player> players;
Rules rules;
vector<IObserver> observers;
bool GameOver;
addPlayer(Player p);
addObserver(IObserver o);
notify(string msg);
play();

```

```

ConsoleNotifier
update(string msg);

```

```

Player
int pd;
string name;
Symbol symbols;
int score;
placeMark(row, col, symbol s);

```

```

Board
vector<vector<Symbol>> grid;
char mark;
getMark();

```

```

StandardRules
isValidMove(Board b, r, c);
checkWin(Board b, symbols);
checkDraw(Board b);

```

```

Symbol
char mark;
getMark();

```

```

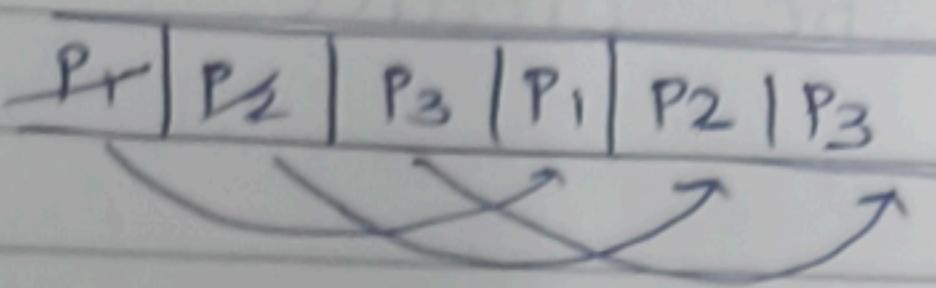
GameFactory
createGame(GameType t, int size);
getCell(row, col);
display();

```

Final UML Diagram

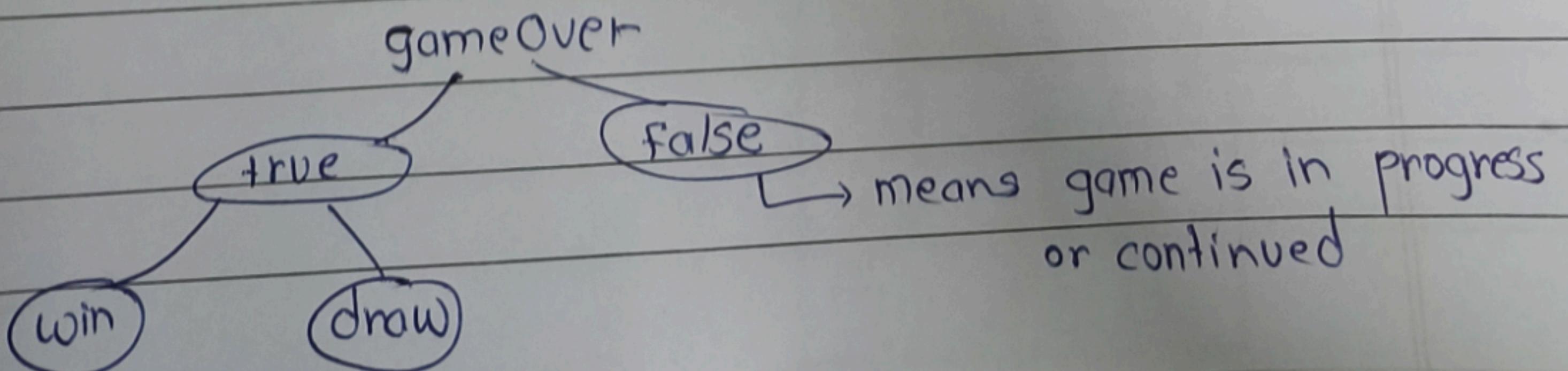
- ⑯ GameFactory class bhi banayegi (optional)
→ it is preferable as it manage when & where object is to be created.

so we will use deque it would manage turns of players efficiently



turn is
⇒ This is how + manage

- ⑨ Player class banayege
- ⑩ Ab humara last requirement raha haf i.e. On App Notifications aur notifications yaane observer pattern
- ⑪ On App Notifications yaane players ko unke phone pr game through message ayega naaki 3rd party like SMS, Email, etc. aur yeh msg both saare players ko dikhega.
- ⑫ so Humara User (i.e. Player class) observer nhi hoga, par koi dusra class hoga whose work is to display notifications on screen
Class Name : consoleNotifier
- ⑬ Hum multiple observer bana skte hai aur player class ko bhi bana sakte hai for email / SMS notification
- ⑭ Ab game khatam ^{hua} ~~hoga~~ yaa nhi uske liye hum boolean variable gameOver ~~hoga~~ use krege



④ Toh grid mein a cell mein em empty bhi ho sakte hoi aur hum waha null nhi store kar sakte hai coz null exception aayega ∴ we choose '-(underscore)' symbol to represent empty cell.

⑤ Board follow SRP coz it has only one responsibility ⇒ to manage board only; usse game uske rules ke baare mein info nhi hai. Board sirf board manage krne ke hi method rakhega.

⑥ Ab rules ke liye hum rule class banayegi having methods -

- checkWin(Board b, symbols); → checks whether anyone win or not

- checkDraw(Board b); → checks whether game is draw / not. (How? ① if board is full)

- isValidMove(Board b, r, l); → checks whether move is valid or not

- ① Row/Col empty hai yaao nhi

- ② If yes then invalid

- ③ If No then valid

⑦ Rule class abstract hogi and concrete class will have different rule strategy

⑧ Ab humme players banane padegi but hum list nhi use kar sakte coz hum game scalable bana rhe hain and no. of players vary in future i.e. 2, 3 or 4.

∴ We turns

⑨ Player

⑩ Ab on obse

⑪ An pho

3r

⑫ :

13

14