Tic Tac Toe

Game

Board board
List<Player> players
Player winner
List<Move> moves
GameState gameState
int currentPlayerIndex

GameState

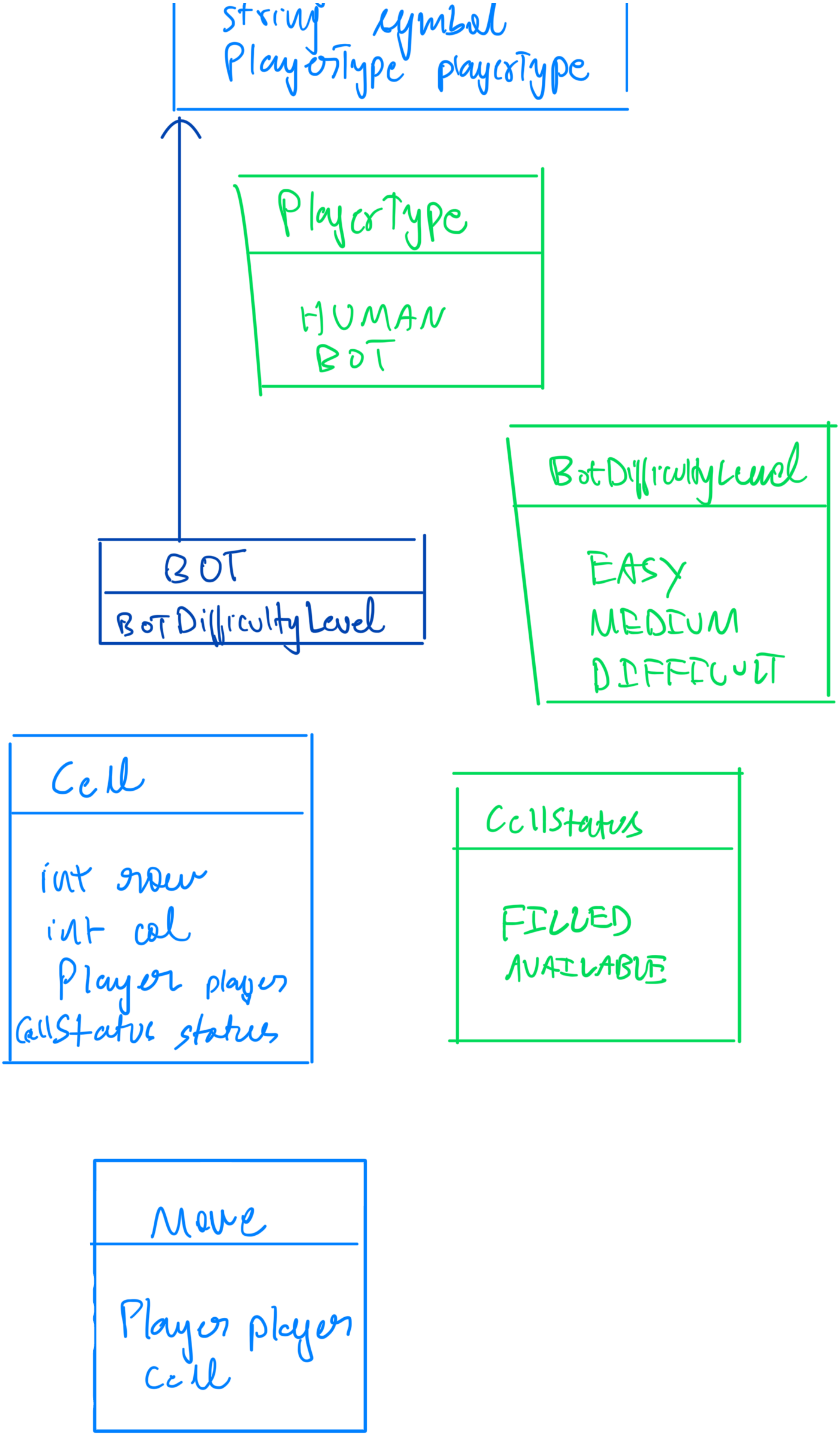
NOT_STARTED
IN_PROGRESS
ENDED
DRAW

Board

int size
List<List<Cell>>

Player

string name



We want to track the flow of a [illegible]

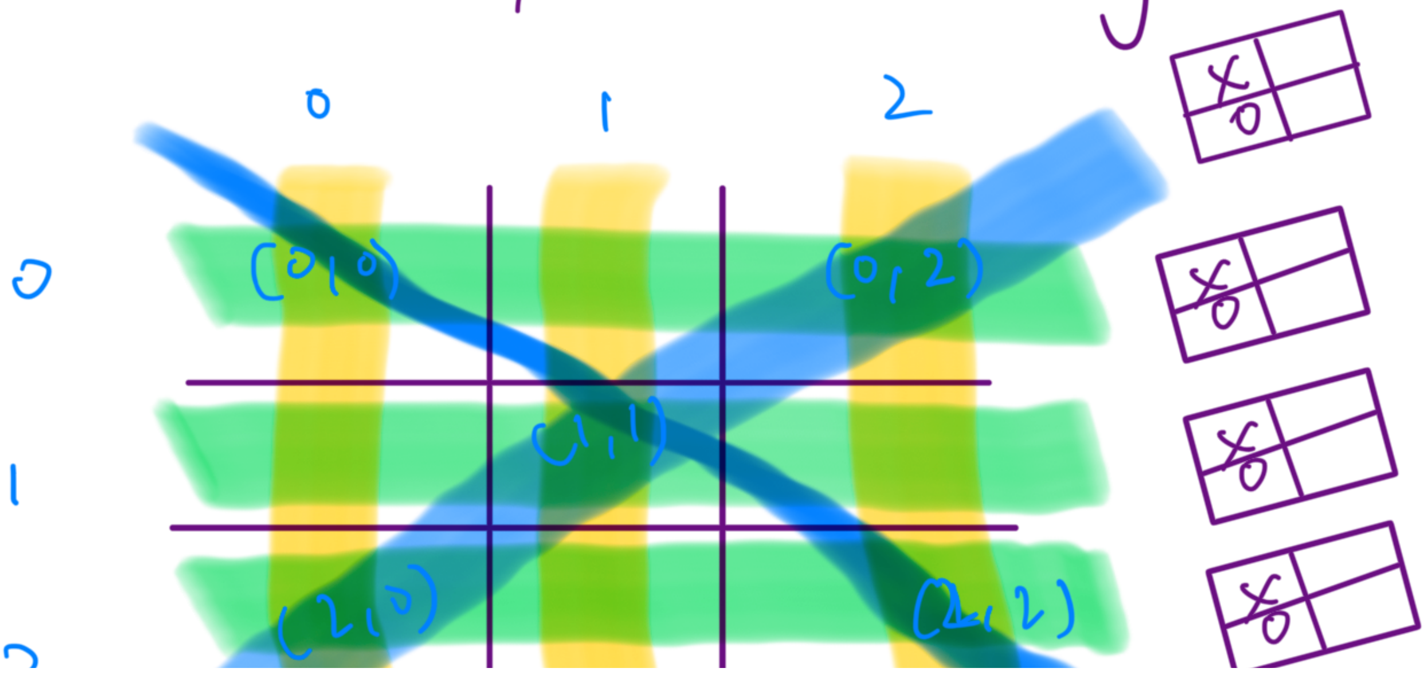
We want to treat the game as a service and the GameController as a controller, also, the Game class acts as a model class as well.

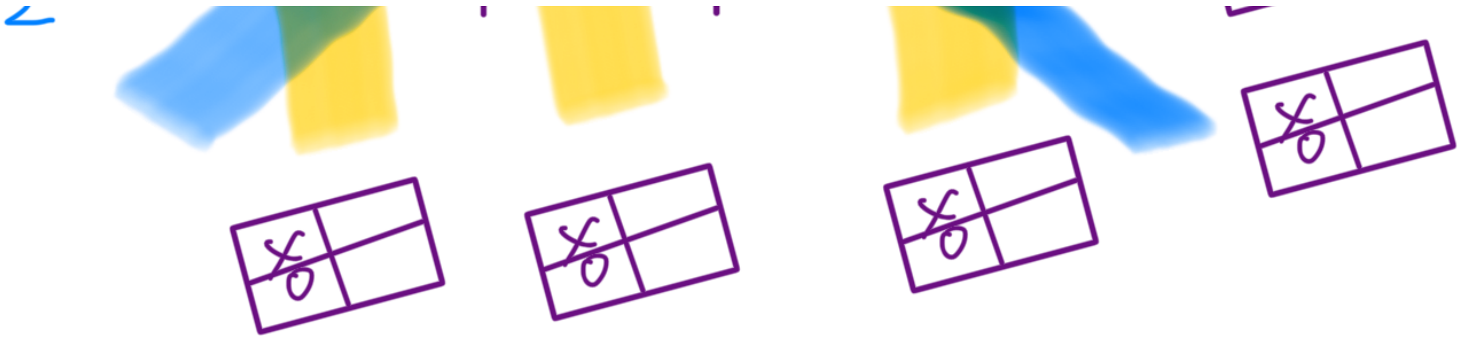
But, when we want to make a move, then, it should be routed via a controller, which is a GameController here.

By default, when the cell is intialised, we are making sure that the cellStatus is

So, we create a new Cell, or, do we not create a new cell,

For checking the winner, we can use the strategy design pattern because, there can be multiple ways in which a player can win like row trio, column trio, or even a diagonal trio.





We can maintain a hashmap for each row and column which can find the winner in O(1)

Since we call this after every move, we can also update this hashmap then.

Also, we dont need to worry about multiple object creations, we can just restrict only one object creation each for the strategy for a Game.