Board verification logic

To store the state of the board and track the moves we have created 1D - array and called it as **state**. Initial value of state = [-1, -1, -1, -1, -1, -1, -1, -1, -1].

Index	0	1	2	3	4	5	6	7	8
Value	-1	-1	-1	-1	-1	-1	-1	-1	-1

Here each index represents a cell. Ex: state[0] means cell A1, state[2] means cell A3, state[7] mean cell A8.

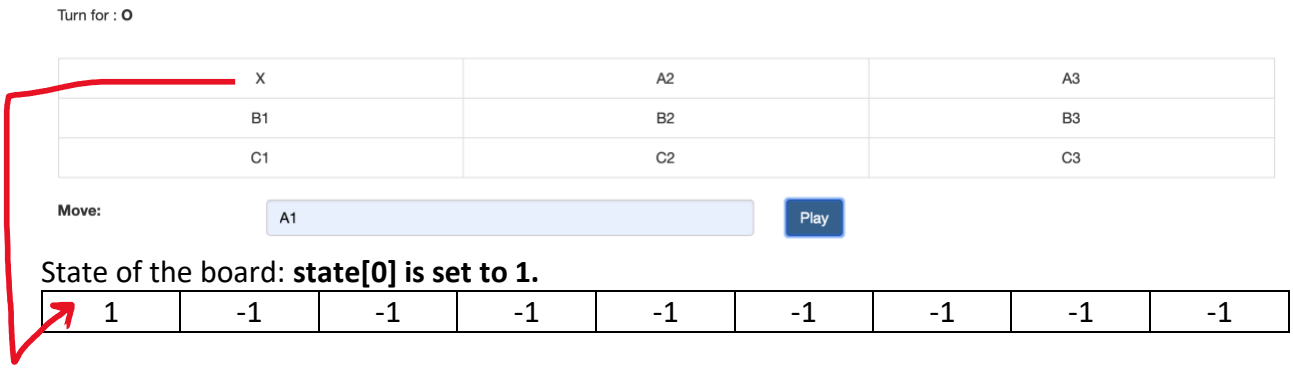
A1	A2	A3
B1	B2	B3
C1	C2	C3

Cells have been named from ["A1", "A2", "A3", "B1", "B2", "B3", "C1", "C2", "C3"] as shown above. For convenience the cells have the same dom id as represented in the image.

How the state changes by player moves: Example - When two moves are made 1st by player 1(X) and then player 2(O), the **state** of the board changes like below.

MOVE Number 1:

Move by 'X' at cell A1 --- (The image is taken after the move was made. So, Turn becomes: O)



State of the board: **state[0] is set to 1.**

1	-1	-1	-1	-1	-1	-1	-1	-1

MOVE Number 2:

Move by 'O' at cell C2 --- (The image is taken after the move was made. So Turn becomes: X)

State of the board: **state[7] is set to 0**.

1	-1	-1	-1	-1	-1	-1	0	-1

How to play tic-tac-toe? https://en.wikipedia.org/wiki/Tic-tac-toe

How to decide winner?

To decide the winner, we just need to check if any of the triplets (as shown above) either row wise or column wise or the diagonal wise has the elements set to either 1 or 0.

INDEX	0	1	2	3	4	5	6	7	8
VALUE	1	1	1	-1	-1	-1	-1	-1	-1

So, after every play you need to verify if someone won or not. This can be done by iterating over all the possible winning combinations. Essentially by checking if any of the combination has either 1 or 0 in it and accordingly deciding the winner.
The combinations will be:

```
[
[0, 1, 2],
[3, 4, 5],
[6, 7, 8],
[0, 3, 6],
[1, 4, 7],
[2, 5, 8],
[0, 4, 8],
[2, 4, 6],
];
```