

(qlebnah f function li qbl)  
circleClicked, db howa li  
clacka ad tfaeal  
circleClicked

array d circles id d circle

id d current player

export const isWinner = (gameBoard, currentMove, currentPlayer) => {

let board = [...gameBoard];

board[currentMove] = currentPlayer;

copy d gameBoard

, bash maywqaechi mutation, manbdlochi original  
hada emlnah gha l test w sf

const winLines = [  
[0, 1, 2, 3],  
[4, 5, 6, 7],  
[8, 9, 10, 11],  
[12, 13, 14, 15],  
[0, 4, 8, 12],  
[1, 5, 9, 13],  
[2, 6, 10, 14],  
[3, 7, 11, 15],  
[0, 5, 10, 16],  
[3, 6, 9, 12],  
];

≡≡≡

1111

X

for (let i = 0; i < winLines.length; i++) {  
const [c1, c2, c3, c4] = winLines[i];

if (  
board[c1] > 0 &&  
board[c1] === board[c2] &&  
board[c2] === board[c3] &&  
board[c3] === board[c4]

ykon nefis raqm matalan 1 1 1 1 f winLines w maykonshi 0

) {  
return true;

matalan f c1=pos4 (1) c2=pos5 (1)  
c3=pos6 (1) c4=pos7 (1)

}  
return false;

};

khash tashi circle maybqa fih 0, w winner gaema tfaelat qbl zaema

export const isDraw = (gameBoard, currentMove, currentPlayer) => {

const board = [...gameBoard];

board[currentMove] = currentPlayer;

reduce ki apply function ela chaque element f array w ki jem3om f resultat  
wahda, x===0 ki returner true false, 1 0, ida x=== 0 true 1 ( kayn 0 f circle)  
ghaybqa yjem3om, khas ykon 0, yaeni hta circle ma fih 0, htashi x ma 0

let count = board.reduce((n, x) => n + (x === 0), 0);

console.log(`count:\${count}`);

return count === 0;

↑  
n=0

konna nqedro neemlo  
for  
w ida board[i] ===0  
n++

};

const getComputerRandomMove = (gameBoard) => {

let validMoves = [];

for (let i = 0; i < gameBoard.length; i++) {  
if (gameBoard[i] === 0) validMoves.push(i);  
}

pon

let rndMove = Math.floor(Math.random() \* validMoves.length);  
return validMoves[rndMove];

};

const getPosition = (gameBoard, moveChecks) => {

for (let check = 0; check < moveChecks.length; check++) {

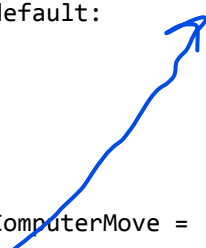
for (let i = 0; i < moveChecks[check].max; i += moveChecks[check].step) {  
let series =  
gameBoard[i + moveChecks[check].indexes[0]].toString() +  
gameBoard[i + moveChecks[check].indexes[1]].toString() +  
gameBoard[i + moveChecks[check].indexes[2]].toString() +  
gameBoard[i + moveChecks[check].indexes[3]].toString();

switch (series) {  
case "1110":  
case "2220":  
return i + moveChecks[check].indexes[3];  
case "1101":  
case "2202":

```

        return i + moveChecks[check].indexes[2];
    case "1011":
    case "2022":
        return i + moveChecks[check].indexes[1];
    case "0111":
    case "0222":
        return i + moveChecks[check].indexes[0];
    default:
    }
    }
    }
    return -1;
};

```



```

export const getComputerMove = (gameBoard) => {
    let moveChecks = [
        // vertical
        {
            indexes: [0, 4, 8, 12],
            max: 4,
            step: 1,
        },
        // horizontal
        {
            indexes: [0, 1, 2, 3],
            max: 16,
            step: 4,
        },
        // diagonal
        {
            indexes: [0, 5, 10, 15],
            max: 16,
            step: 16,
        },
        // diagonal
        {
            indexes: [3, 6, 9, 12],
            max: 16,
            step: 16,
        }
    ];

    let position = getPosition(gameBoard, moveChecks);
    if (position > -1) return position;

    return getComputerRandomMove(gameBoard);
};

```