```
let canvas = document.getElementById('canvas');
let ctx = canvas.getContext("2d");

canvas.width = window.innerWidth;
canvas.height = window.innerHeight;
//set gamewidth and gameheight to scale with window size
let gameheight;
let gamewidth;
if (canvas.width >= canvas.height * 7 / 6) {
    gameheight = canvas.height / 2;
    gamewidth = gameheight * 7 / 6;
}
else {
    gamewidth = canvas.width / 2;
    gameheight = gamewidth * 6 / 7;
}
ctx.fillStyle = "pink";
ctx.fillRect(0, 0, canvas.width, canvas.height);
// drawing outside rectangle for the board
ctx.strokeRect((canvas.width - gamewidth) / 2, (canvas.height - gameheight) / 2, gamewidth, gameheight);
ctx.beginPath();
// drawing the lines within the rectangle to seperate the columns and rows
for (i = 1; i < 7; i++) {
    ctx.moveTo((canvas.width - gamewidth) / 2 + i * gamewidth / 7, (canvas.height - gameheight) / 2);
    ctx.lineTo((canvas.width - gamewidth) / 2 + i * gamewidth / 7, (canvas.height - gameheight) / 2 + gameheight);
}
for (i = 1; i < 6; i++) {
    ctx.moveTo((canvas.width - gamewidth) / 2, (canvas.height - gameheight) / 2 + i * gameheight / 6);
    ctx.lineTo((canvas.width - gamewidth) / 2 + gamewidth, (canvas.height - gameheight) / 2 + i * gameheight / 6);
}
ctx.stroke();
// writing numbers above columns
for (i = 1; i < 8; i++) {
    ctx.font = `${gameheight / 10}px Times New Roman`;
    ctx.fillStyle = "black";
    ctx.textAlign = "center";
    ctx.fillText(i, (canvas.width - gamewidth) / 2 + i * gamewidth / 7 - gamewidth / 14, (canvas.height - 1.1 * gameheight) / 2);
}

let finish = false;
let turn = "Red";
let move = 0;

function setstatus(text) {
    // clear previous status
    ctx.fillStyle = "pink";
    ctx.fillRect(0, 0, canvas.width, canvas.height / 6);
    // set new status
    ctx.font = `${gameheight / 10}px Times New Roman`;
    ctx.fillStyle = "black";
    ctx.textAlign = "center";
    ctx.fillText(text, canvas.width / 2, canvas.height / 10);
}
```

```
// defining list that contains the positions of the pieces
let board = [[], [], [], [], [], [], []];

function placepiece(keypressed) {
    // checks if input is valid
    if (parseInt(keypressed) && parseInt(keypressed) < 8) {
        column = parseInt(keypressed) - 1
        // check if the column is filled
        if (board[column].length < 6) {
            setstatus("You pressed the " + keypressed + " key. " + turn + " piece was placed in column " + parseInt(column + 1))
            // add turn to the board
            board[column].push(turn);
            // color the appropriate box the color of the turn
            ctx.fillStyle = turn;
            ctx.fillRect((canvas.width - gamewidth) / 2 + gamewidth * ((column) / 7) + 1, (canvas.height - gameheight) / 2 + gameheight * (6 - board[column].length) / 6 + 1, gamewidth / 7 - 2, gameheight / 6 - 2)
            // increment move number
            move++
            // win condition checking
            for (i = -3; i < 1; i++) {
                let row = board[column].length - 1;
                // horizontal check
                if (column + i >= 0 && column + i <= 3) {
                    if (board[column + i][row] == turn && board[column + i + 1][row] == turn && board[column + i + 2][row] == turn && board[column + i + 3][row] == turn) {
                        finish = true;
                        setstatus(turn + " is the winner!")
                    }
                }
                // vertical check
                if (row + i >= 0 && row + i <= 2) {
                    if (board[column][row + i] == turn && board[column][row + i + 1] == turn && board[column][row + i + 2] == turn && board[column][row + i + 3] == turn) {
                        finish = true;
                        setstatus(turn + " is the winner!")
                    }
                }
                // top right bottom left diagonal
                if (row + i >= 0 && row + i <= 2 && column + i >= 0 && column + i <= 3) {
                    if (board[column + i][row + i] == turn && board[column + i + 1][row + i + 1] == turn && board[column + i + 2][row + i + 2] == turn && board[column + i + 3][row + i + 3] == turn) {
                        finish = true;
                        setstatus(turn + " is the winner!")
                    }
                }
                // top left bottom right diagonal
                if (row - i <= 5 && row - i >= 3 && column + i >= 0 && column + i <= 3) {
                    if (board[column + i][row - i] == turn && board[column + i + 1][row - i - 1] == turn && board[column + i + 2][row - i - 2] == turn && board[column + i + 3][row - i - 3] == turn) {
                        finish = true;
                        setstatus(turn + " is the winner!")
                    }
```

```
                }
            }
            // toggle the turn
            if (turn == "Red") {
                turn = "Blue";
            }
            else {
                turn = "Red";
            }
            // check if board is filled
            if (move == 42 && !finish) {
                finish = true;
                setstatus("It's a draw!")
            }
        }
        else {
            // tells player the column is filled
            setstatus("You pressed the " + keypressed + " key. That column is already filled!")
        }
    }
    else {
        // tells player to press a numerical key between 1 and 7
        setstatus("You pressed the " + keypressed + " key. Please press a numerical key between 1 and 7.")
    }
}

function refreshinfo() {
    ctx.fillStyle = "pink";
    ctx.fillRect(0, canvas.height * 3 / 4 + 1, canvas.width, canvas.height / 4);
    ctx.font = `${gameheight / 8}px Times New Roman`;
    ctx.fillStyle = "black";
    ctx.textAlign = "center";
    ctx.fillText(turn + ", it is your turn!", canvas.width / 2, canvas.height * 33 / 40);
    ctx.fillText("Move: " + move, canvas.width / 2, canvas.height * 9 / 10);
}

refreshinfo();

document.addEventListener("keyup", e => {
    if (!finish) {
        // try to place piece corresponding to key pressed
        placepiece(e.key);
        // display move number and whose turn it is
        refreshinfo()
    }
})
```