Checkerboard

Demo

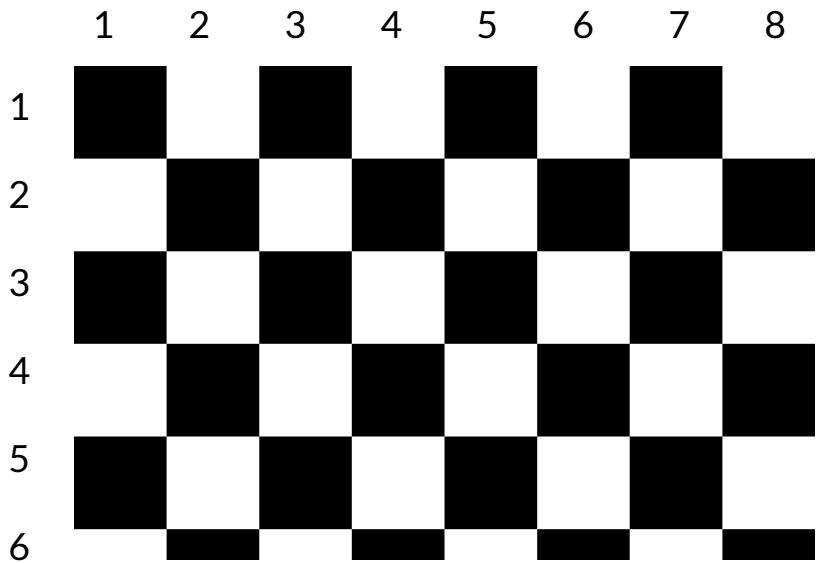
Csaba Hadnagy
Green Fox Gymnocercus EGO
2020-2021

27.11.2020

600 px

The classic checkerboards has 8 columns & rows and they are square-shaped.
But we have here a rectangle… ☹ *...so we gotta deal with it.*

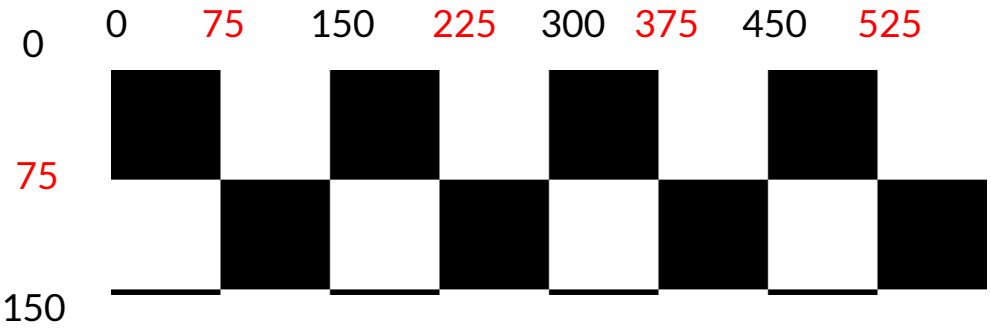
400px



8 columns
600 / 8 = 75, so
the side of the square = 75

400 / 75 = 5.3333… so
6 rows

The ‘x’ and ‘y’ coordinates will be the multiples of 75 including 0.



Odd row :
x- coordinates: 0, 150, 300, 450
y-coordinates: 0 , 150, 300

Even row :
x- coordinates: 75, 225, 375, 525
y-coordinates: 75 , 225, 375

2 types of row to emulate.

- *They are shifted to each other.*
- *Coordinates should mirror this.*

We need altogether 4 loops.

That's our matrix.

```
let side = 75;

let board = [
  [0, 150, 300, 450],      //x1
  [75, 225, 375, 525],     //x2
  [0, 150, 300],           //y1
  [75, 225, 375]           //y2
]
//y1 runs in x1 ----> odd rows
//y2 runs in x2 ----> even rows
```

*the **'x' arrays** are defined so:*

board [0] = x1
board [1] = x2

*while **'y' ones** so:*

board [2] = y1
board [3] = y2

```
//odd rows
for (let i = 0; i < board[0].length; i++) {

  for (let j = 0; j < board[2].length; j++) {
    ctx.fillRect(board[0][i], board[2][j], side, side)
  }
}

//even rows
for (let i = 0; i < board[1].length; i++) {

  for (let j = 0; j < board[3].length; j++) {
    ctx.fillRect(board[1][i], board[3][j], side, side)
  }
}
```

We need altogether 4 loops.

That's our matrix.

```
let side = 75;

let board = [
  [0, side * 2, side * 4, side * 6],  //x1
  [side, side * 3, side * 5, side *7],//x2
  [0, side * 2, side * 4],            //y1
  [side, side * 3, side * 5]          //y2
]

//y1 runs  in x1   ----> odd rows
//y2 runs  in x2   ----> even rows
```

*the **'x' arrays** are defined so:*

board [0] = x1
board [1] = x2

*while **'y' ones** so:*

board [2] = y1
board [3] = y2

```
//odd rows
for (let i = 0; i < board[0].length; i++) {

  for (let j = 0; j < board[2].length; j++) {
    ctx.fillRect(board[0][i], board[2][j], side, side)
  }
}

//even rows
for (let i = 0; i < board[1].length; i++) {

  for (let j = 0; j < board[3].length; j++) {
    ctx.fillRect(board[1][i], board[3][j], side, side)
  }
}
```


Thank you
attention!!!
FOR YOUR
That's all Folks!