

# Chessboard Condition

Jonathan Lam

August 18, 2018

The only reason I'm writing this is because I saw other people do some really complicated stuff with their if statements, or using some weird long method. It's actually really simple, so I thought I'd write this to explain my method.

Here's my very simple if statement.

```
if ((i/64 + j/64) % 2 == 0)
```

This takes advantage of the fact that  $i/64$  acts as the floor function, so we have mapped every point of the form

$$(x, y), \quad 0 \leq x, y < 512$$

into a point

$$(x, y), \quad 0 \leq x, y < 8.$$

Note that we can compute a sum table as below:

	0	1	2	3	4	5	6	7
0	0	1	2	3	4	5	6	7
1	1	2	3	4	5	6	7	8
2	2	3	4	5	6	7	8	9
3	3	4	5	6	7	8	9	10
4	4	5	6	7	8	9	10	11
5	5	6	7	8	9	10	11	12
6	6	7	8	9	10	11	12	13
7	7	8	9	10	11	12	13	14

We can take these sums mod 2.

	0	1	2	3	4	5	6	7
0	0	1	0	1	0	1	0	1
1	1	0	1	0	1	0	1	0
2	0	1	0	1	0	1	0	1
3	1	0	1	0	1	0	1	0
4	0	1	0	1	0	1	0	1
5	1	0	1	0	1	0	1	0
6	0	1	0	1	0	1	0	1
7	1	0	1	0	1	0	1	0

And the chessboard pattern is now clear within this table. We just take  $x + y$  and check whether it is 0 or 1 mod 2.

The full function is on the next page.

```
void drawChessboard(pixel pixels[BOARD_SIZE][BOARD_SIZE]) {
    for (int i = 0; i < BOARD_SIZE; i++) {
        for (int j = 0; j < BOARD_SIZE; j++) {
            if ((i/64 + j/64) % 2 == 0) { // this is the line which is different
                // black
                pixels[i][j].red = 0;
                pixels[i][j].green = 0;
                pixels[i][j].blue = 0;
            } else {
                // white
                pixels[i][j].red = 255;
                pixels[i][j].green = 255;
                pixels[i][j].blue = 255;
            }
        }
    }
}
```