|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 00 | 01 | 02 | 03 | 04 | 05 | 06 | 07 | 08 | 09 |
| 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 |
| 20 | 21 | 22 | 23 | 24 | 25 | 26 | 27 | 28 | 29 |
| 30 | 31 | 32 | 33 | 34 | 35 | 36 | 37 | 38 | 39 |
| 40 | 41 | 42 | 43 | 44 | 45 | 46 | 47 | 48 | 49 |
| 50 | 51 | 52 | 53 | 54 | 55 | 56 | 57 | 58 | 59 |
| 60 | 61 | 62 | 63 | 64 | 65 | 66 | 67 | 68 | 69 |
| 70 | 71 | 72 | 73 | 74 | 75 | 76 | 77 | 78 | 79 |
| 80 | 81 | 82 | 83 | 84 | 85 | 86 | 87 | 88 | 89 |
| 90 | 91 | 92 | 93 | 94 | 95 | 96 | 97 | 98 | 99 |

So the first digit is the row index and the second digit is the column index. The issue is when we check the cells around a specific cell to see if they’re dead or alive to determine if the specific cell should be dead or alive in the next generation. Ex:

//array t is a two dimensional array filled with values of 0s and 1s, 0 for dead and 1 for alive. You can //access an element by t[i][j], it represents the current generation.

//array n is the next generation two dimension array.

//to check if a neighbor cell is alive, we must make sure they exist as well. Checking existence is important if the cell we are at is on a border. One method is to make sure the row and column index of the neighbor cell is >= 0. Be sure to check existence first before we try to access the contents.

Var alive\_count=0;

For(var I = 0; I < rowlength; i++){

For(var j = 0; j < collength; j++){

//if (top cell alive) alive\_count++;

//if (bottom cell alive) alive\_count++;

//if (left cell alive) alive\_count++;

//if (right cell alive) alive\_count++;

//if (lefttop cell alive) alive\_count++;

//if (righttop cell alive) alive\_count++;

//if (leftbottom cell alive) alive\_count++;

//if (rightbottom cell alive) alive\_count++;

If (t[i][j] == 1 & alive\_count < 2) n[i][j] = 0;

If (t[i][j] == 1 & alive\_count > 3) n[i][j] = 0;

If (t[i][j] == 1 & (alive\_count >= 2 or alive\_count <= 3) ) n[i][j] = 1;

If(t[i][j] == 0 & alive\_count==3) n[i][j] = 1;

}

}