Finite State Machines (Cellular Automata)

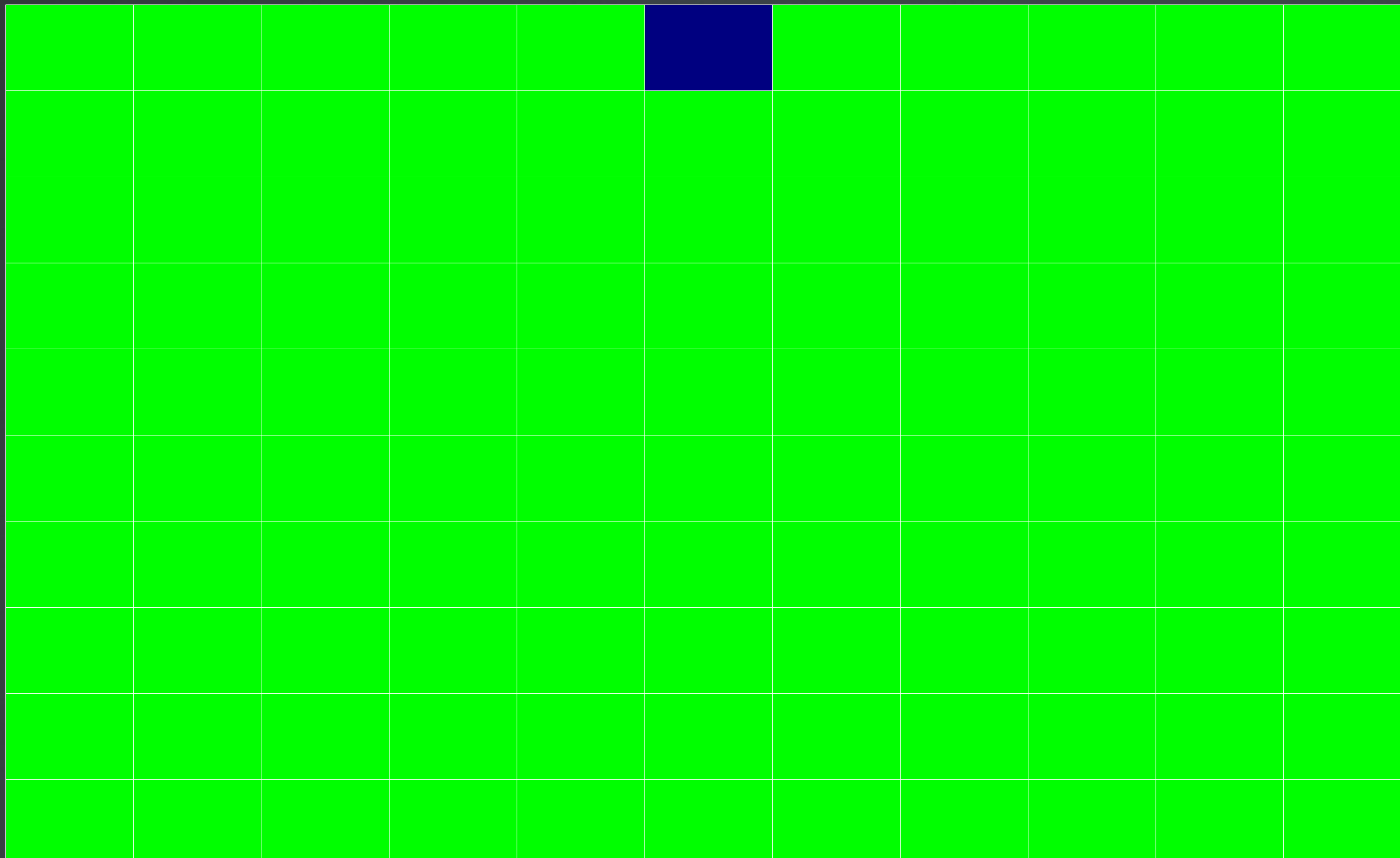
- What they are
- What they do
- Uses
- Implications

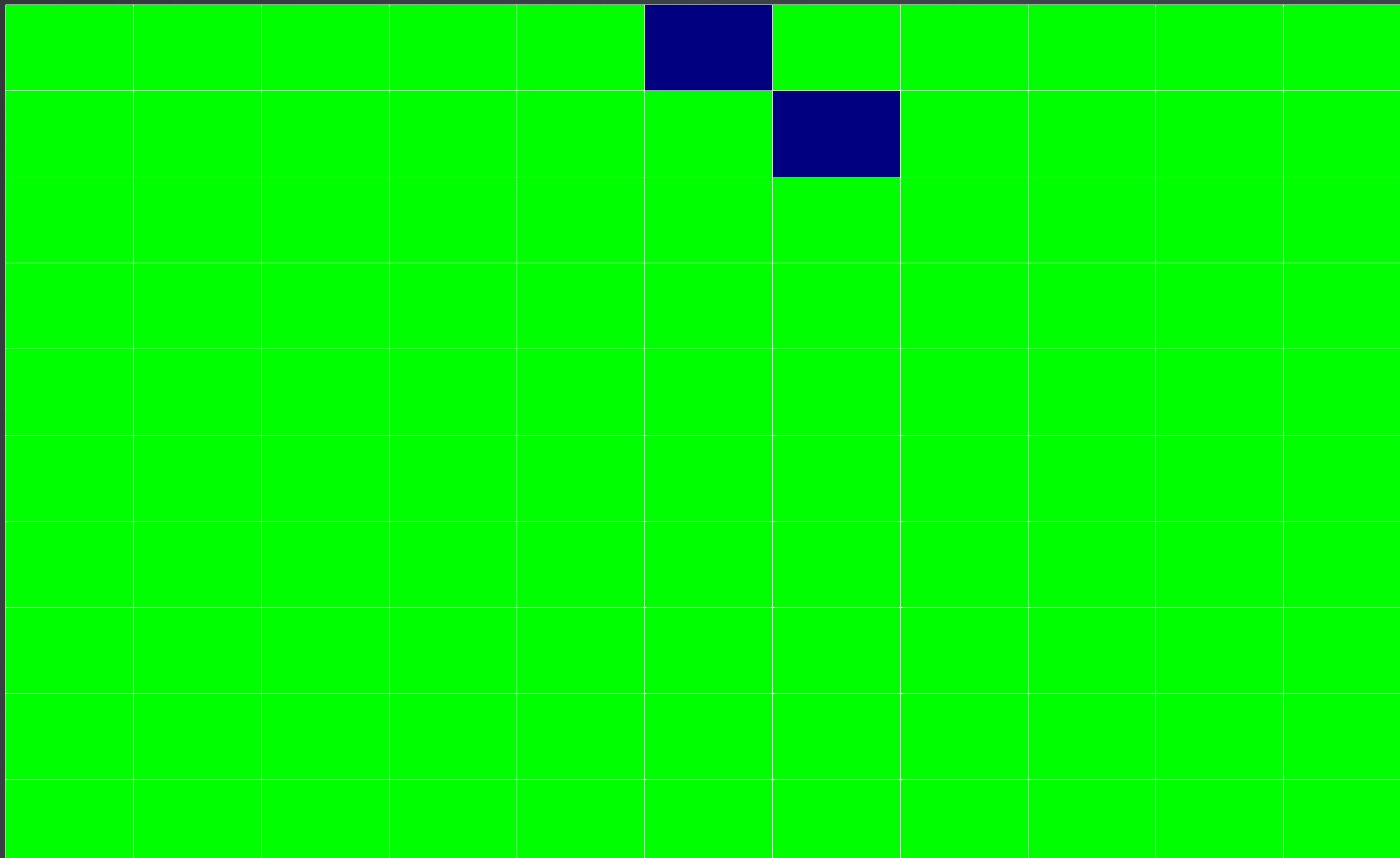
What they are

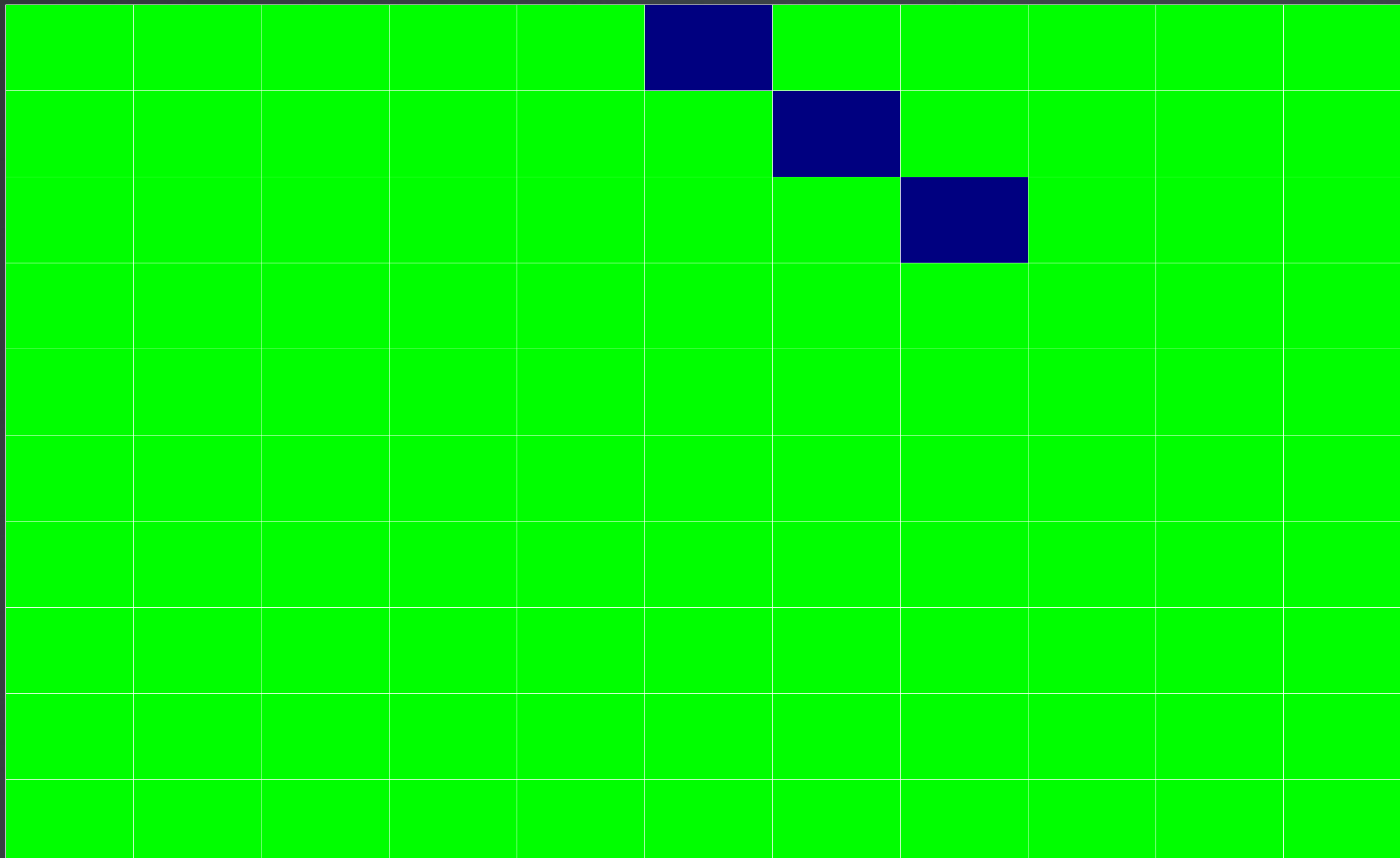
- An array of “cells”
  - 1 dimensional, 2 dimensional, or even more
  - May be of fixed size or “infinite”
- Each cell is in a state – think of it as colour
- At each time step the states change
  - New state depends on this cell and its immediate neighbours
  - Simple rules control the state changes

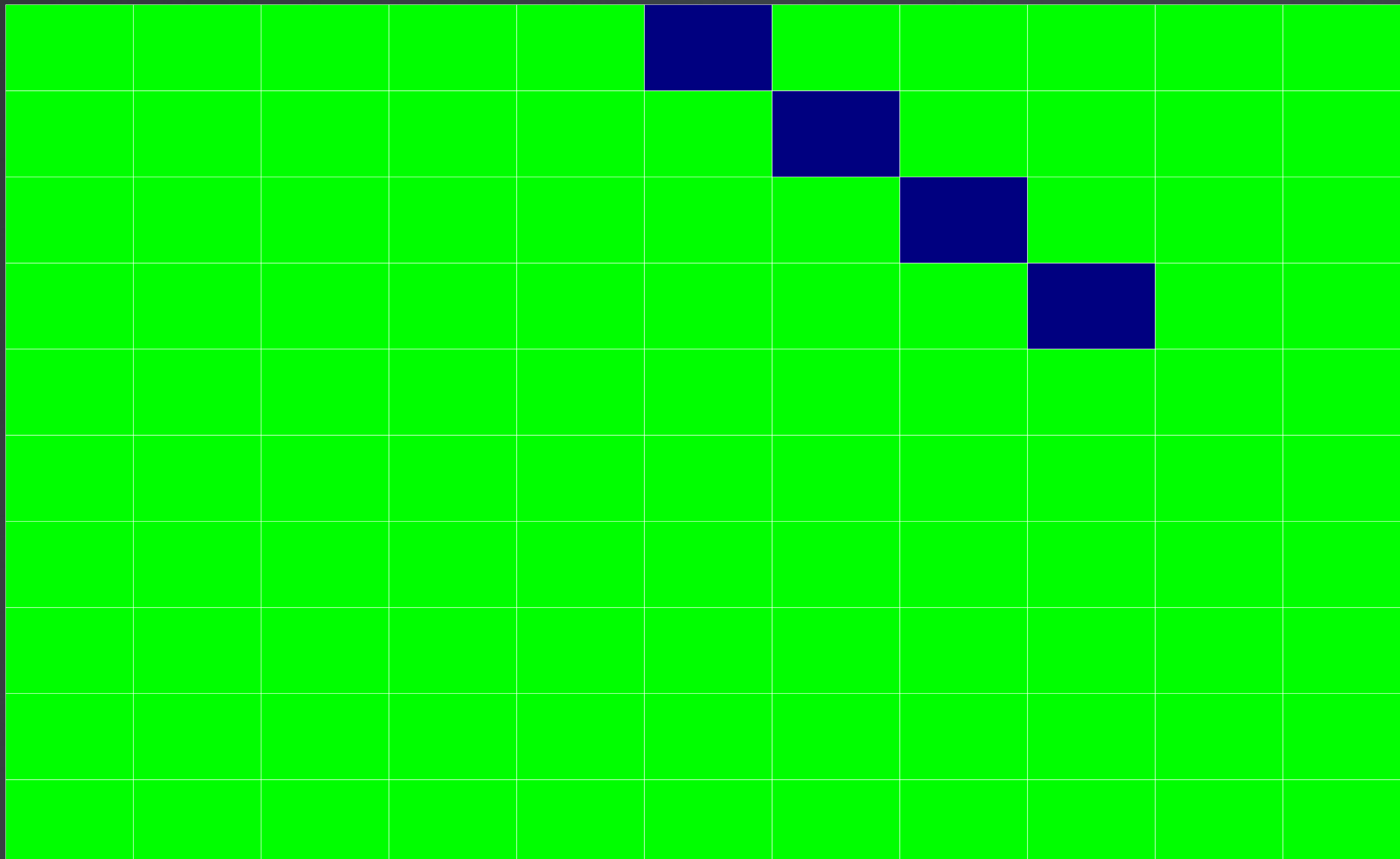
A Sample

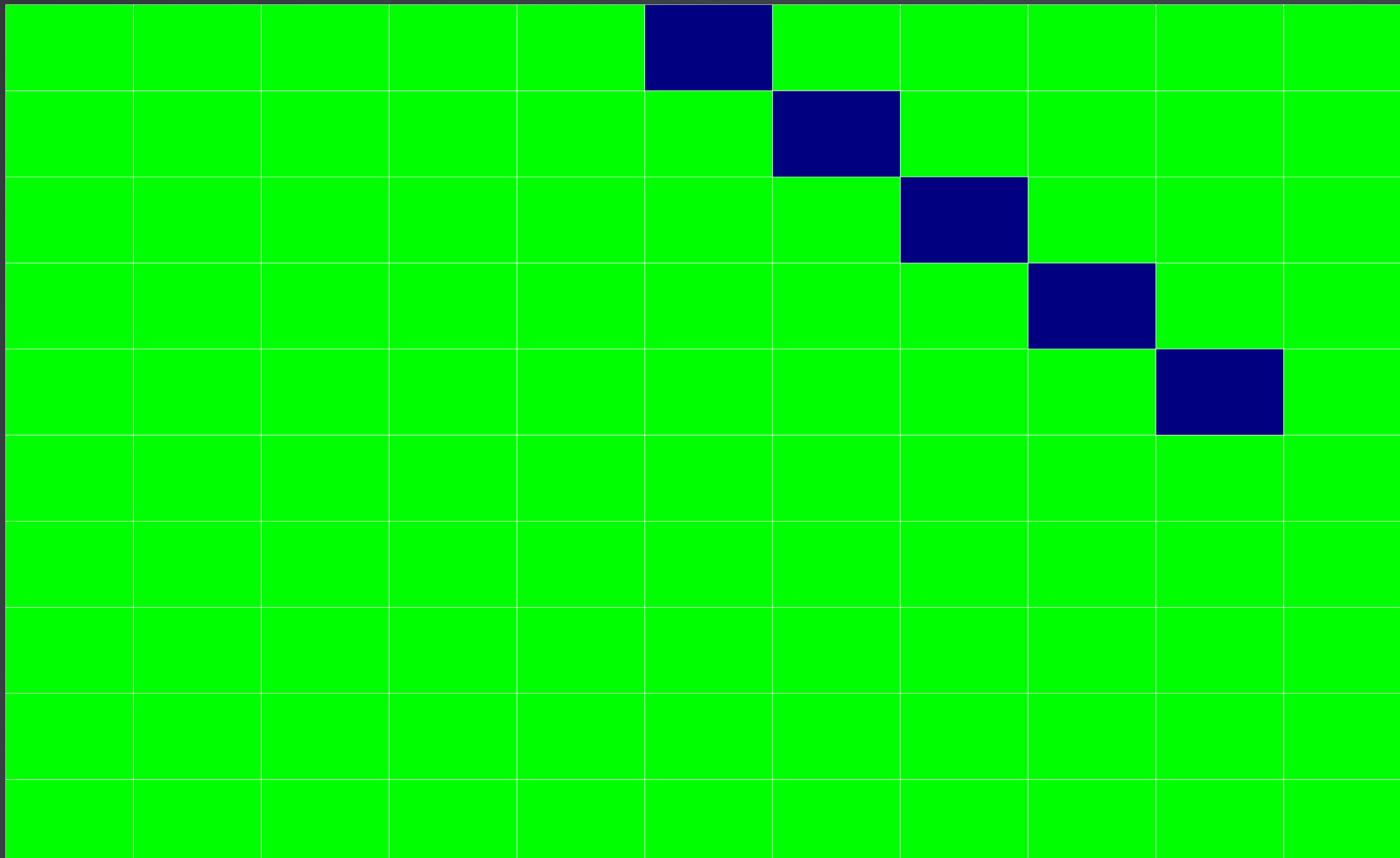
- Imagine a row of lets say 11 cells
  - In one of two states – e.g. black/white
  - All the cells are intially white, except the centre one.
- Transition rules:
  - A cell is white unless the cell to its left is black, then it is black
- We can draw the cells in a row across, and show the next state in the line below them

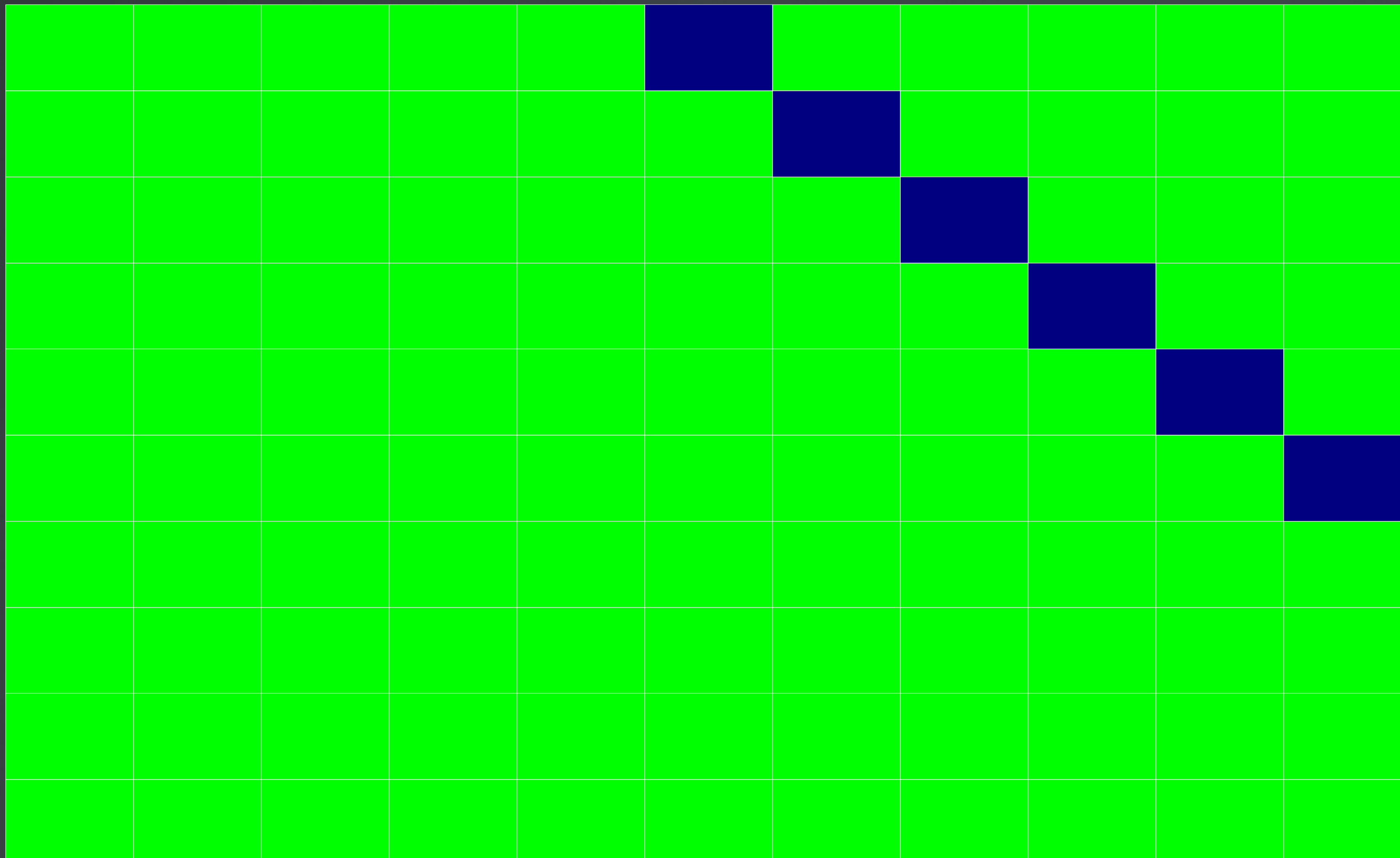












Pretty simple, really.

The state of a cell depends only on its own state and its (two) immediate neighbours.

There are exactly eight possible combinations:

```
0 0 0  → 0
0 0 1  → 0
0 1 0  → 0
0 1 1  → 0
1 0 0  → 1
1 0 1  → 0
1 1 0  → 0
1 1 1  → 0
```

If we consider all possible sets of rules:

```
0 0 0  → ?
0 0 1  → ?
0 1 0  → ?
0 1 1  → ?
1 0 0  → ?
1 0 1  → ?
1 1 0  → ?
1 1 1  → ?
```

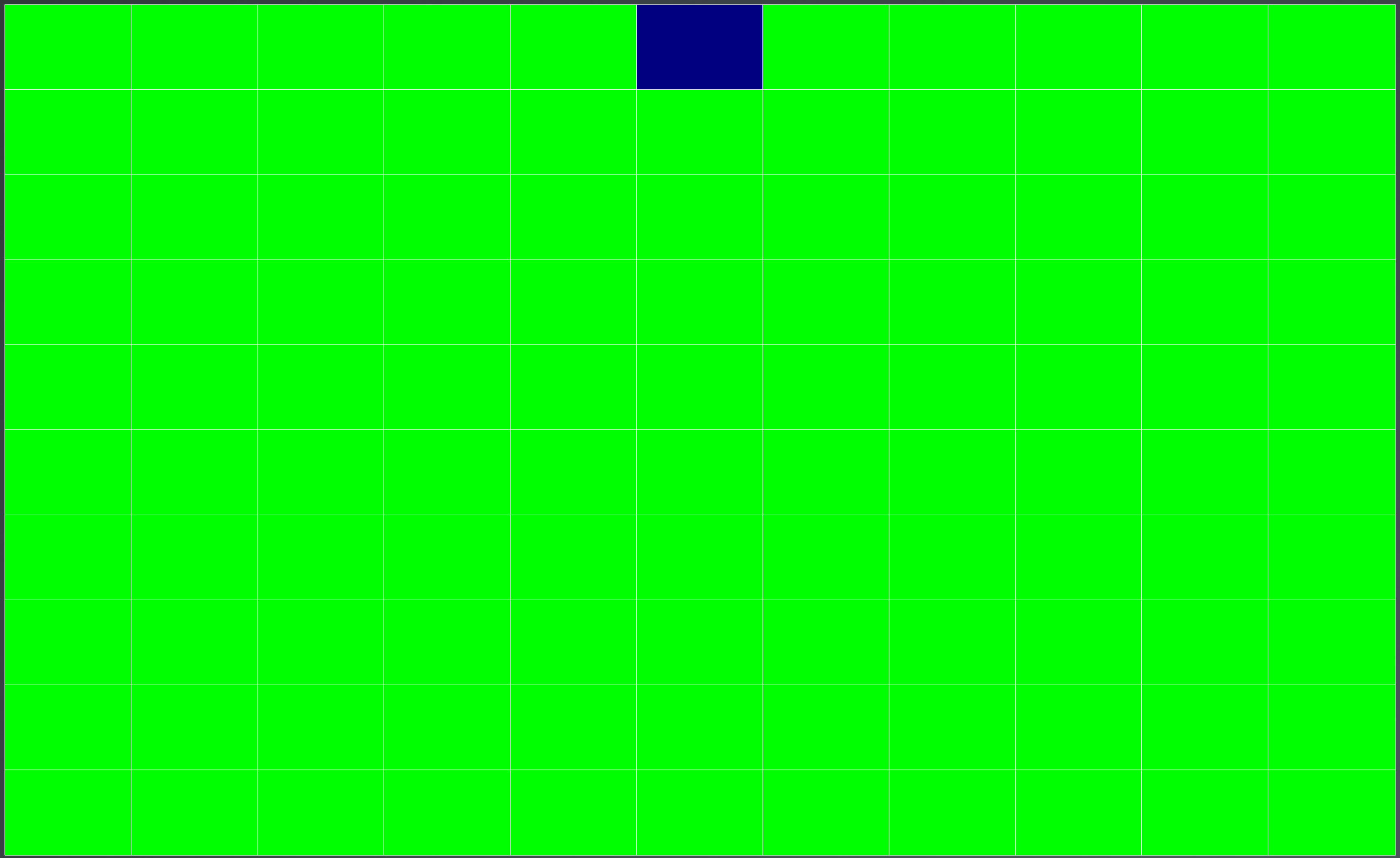
We can see that there are just 256 possible sets

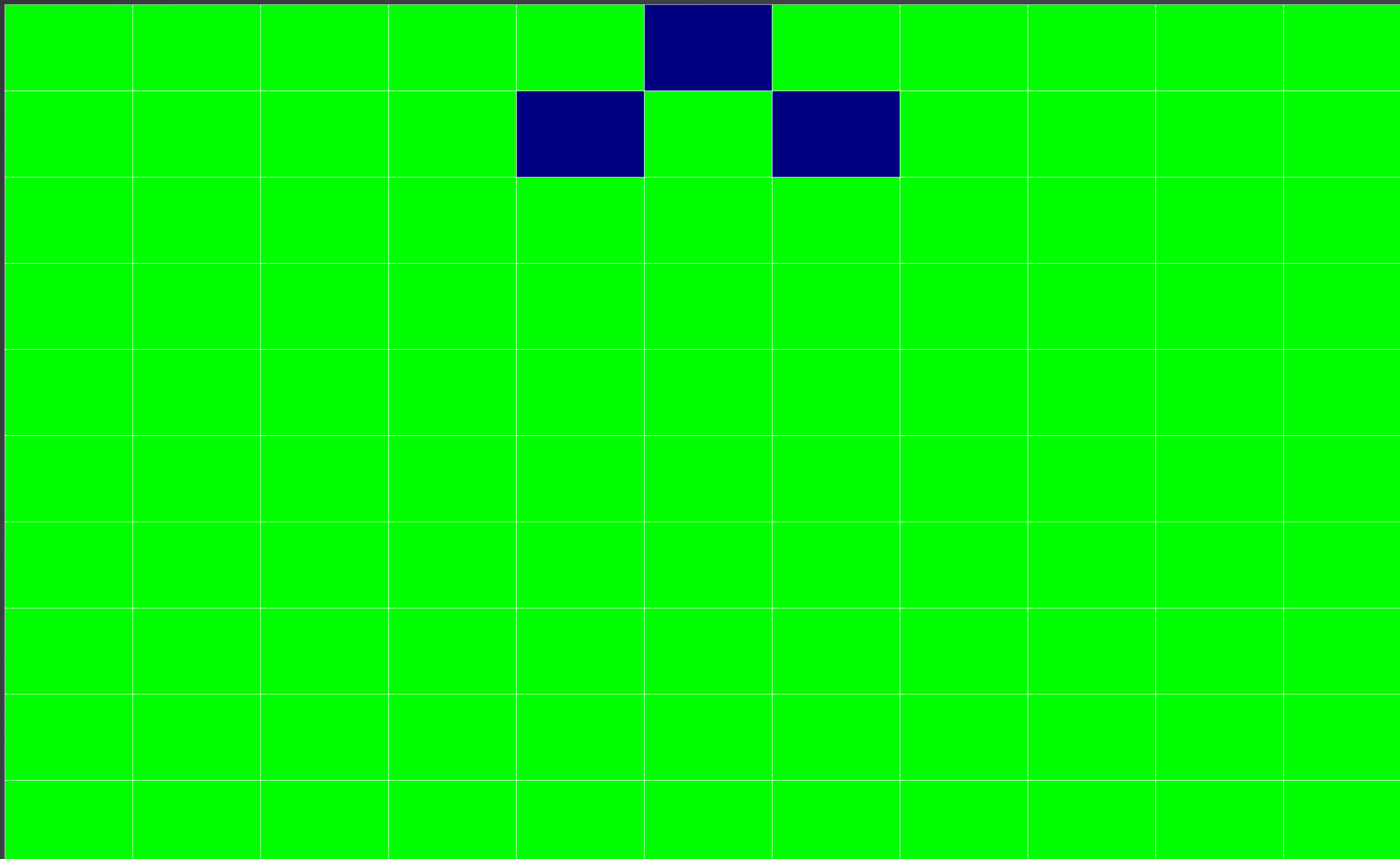
Let us look at this example:

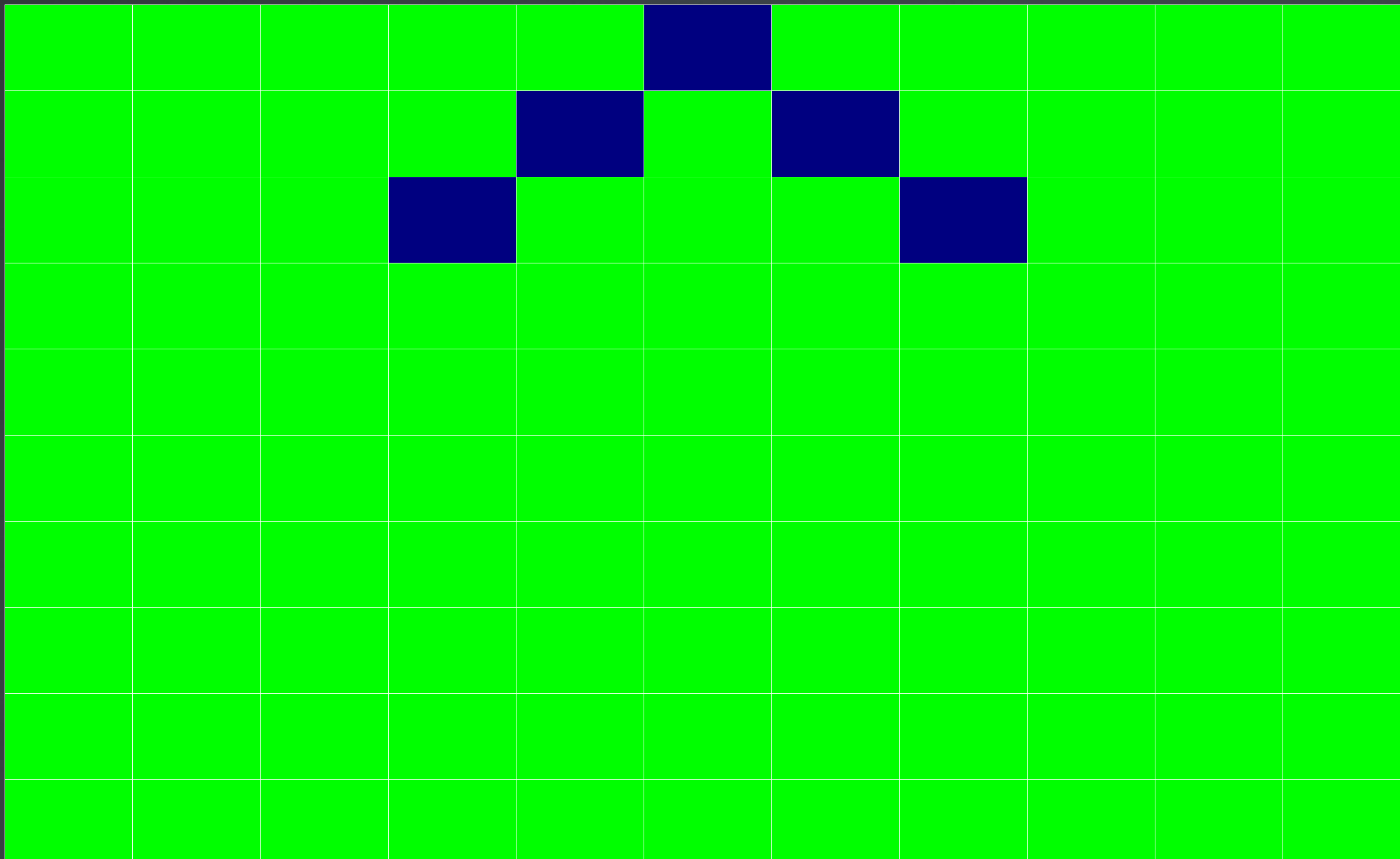
```
0 0 0  → 0
0 0 1  → 1
0 1 0  → 0
0 1 1  → 0
1 0 0  → 1
1 0 1  → 0
1 1 0  → 0
1 1 1  → 0
```

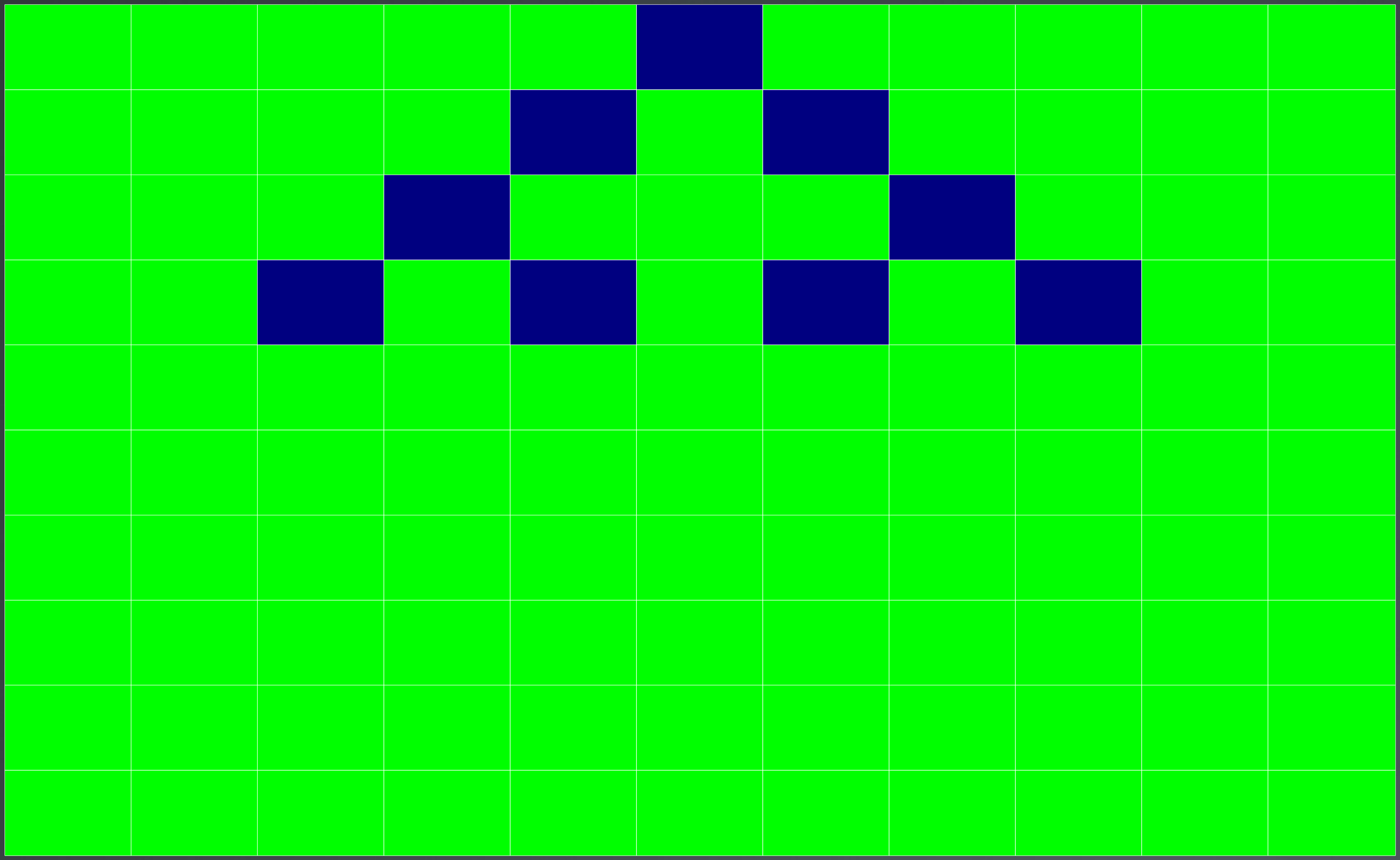
“A cell becomes black if just the one above and to the left or right is black”

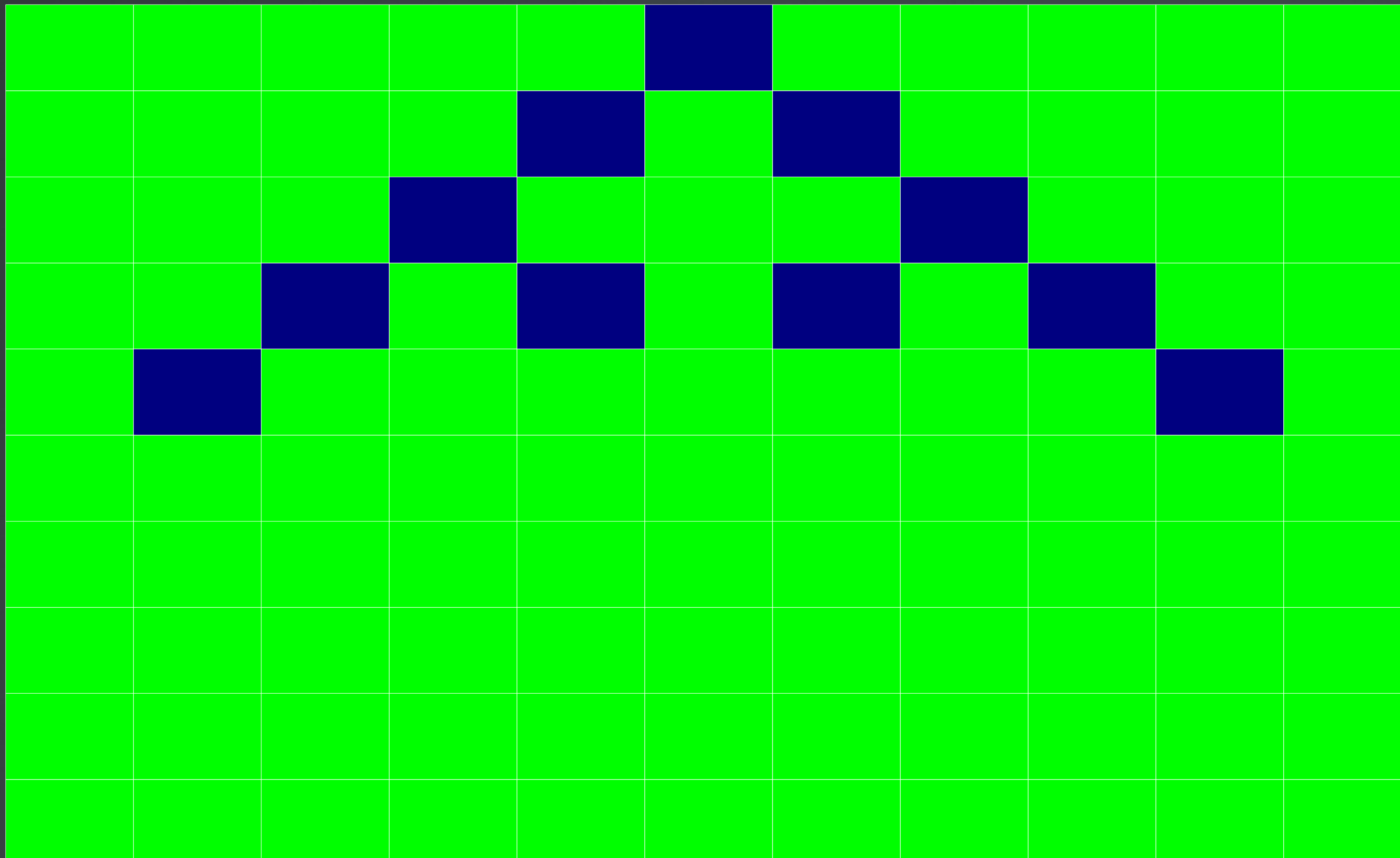
- So it is all very simple, and we could write a computer program to do the calculation, and actually look at every possible case.
- Question:
  - Are any of the patterns generated more ”interesting” than this one?

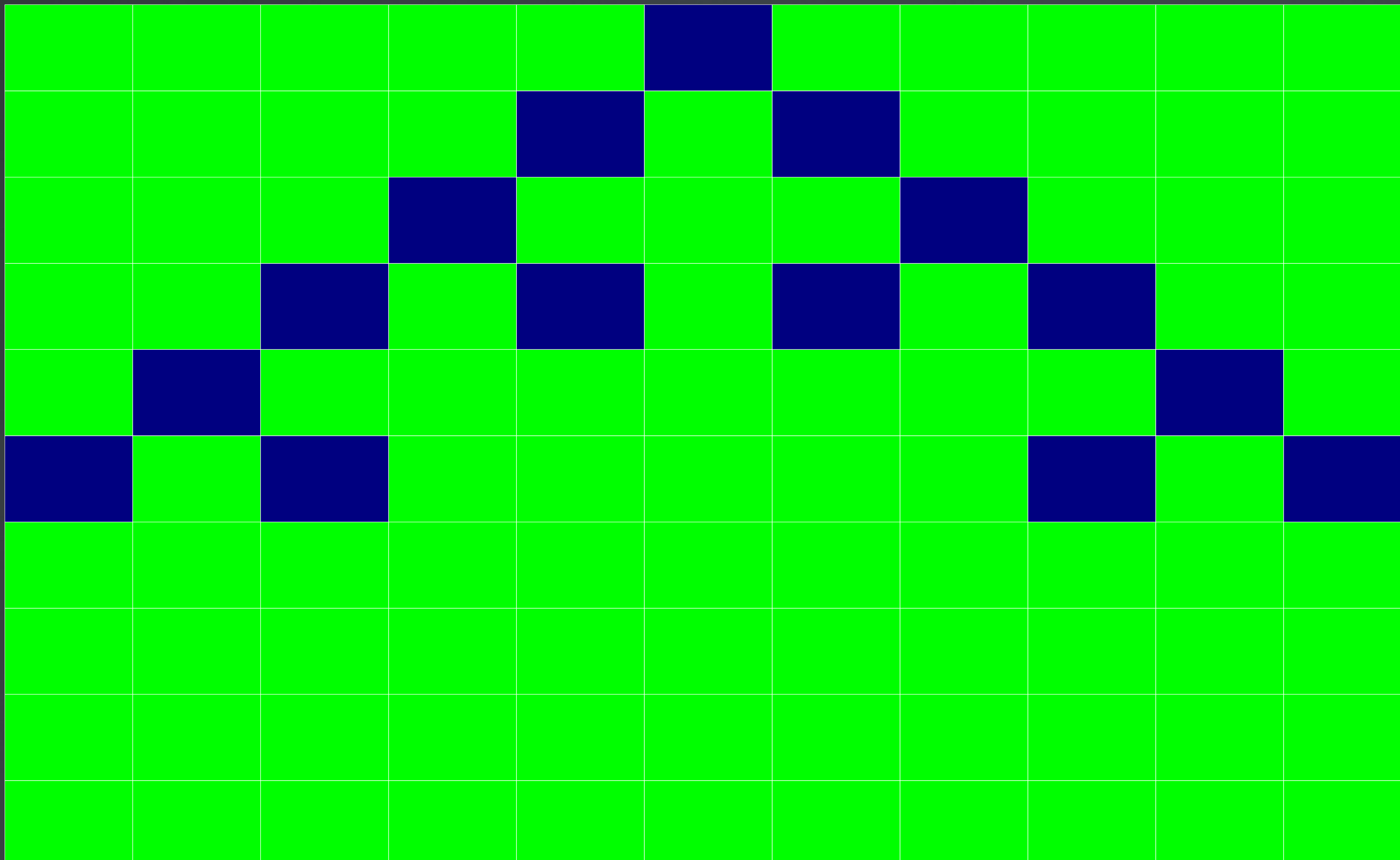


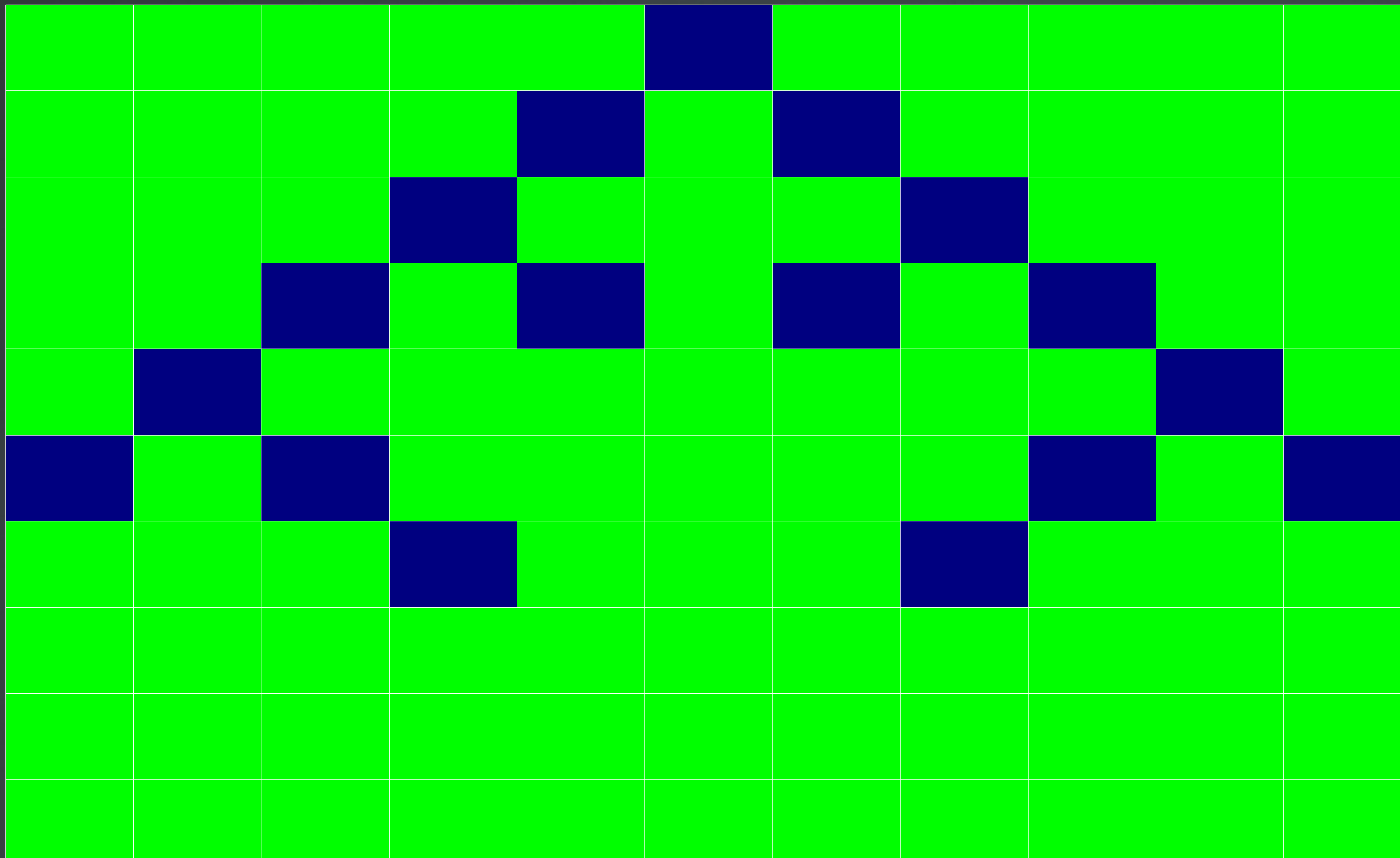


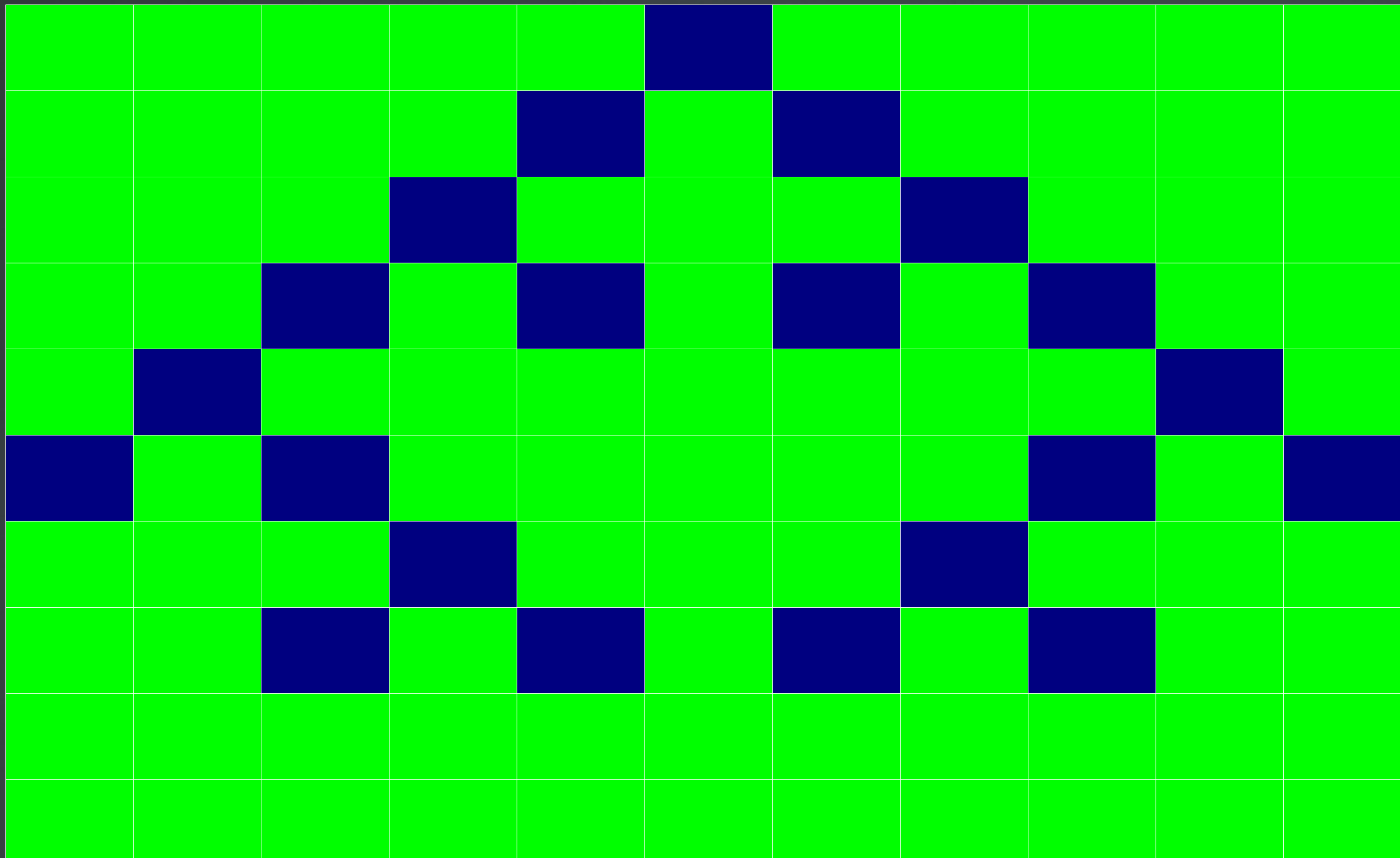


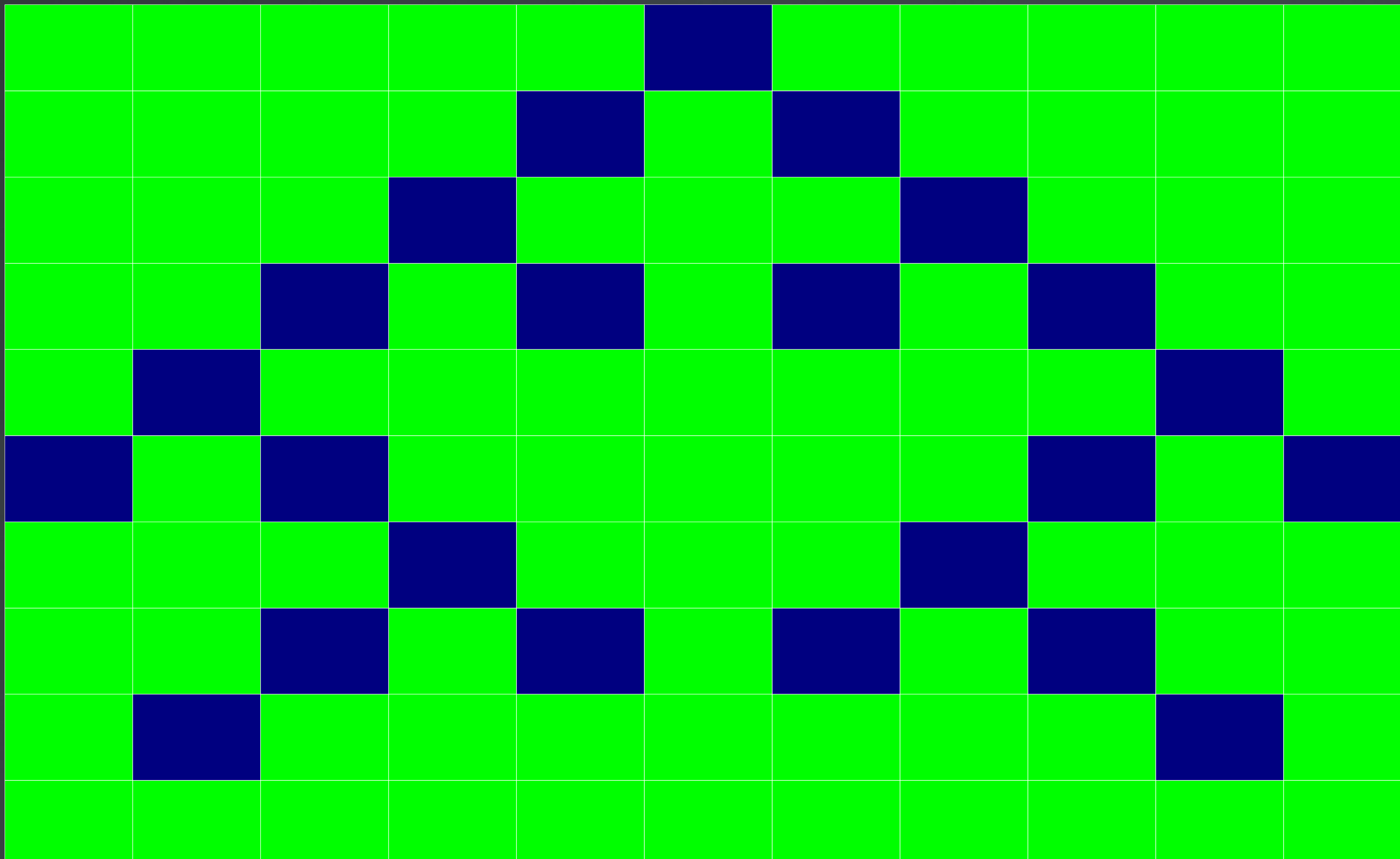


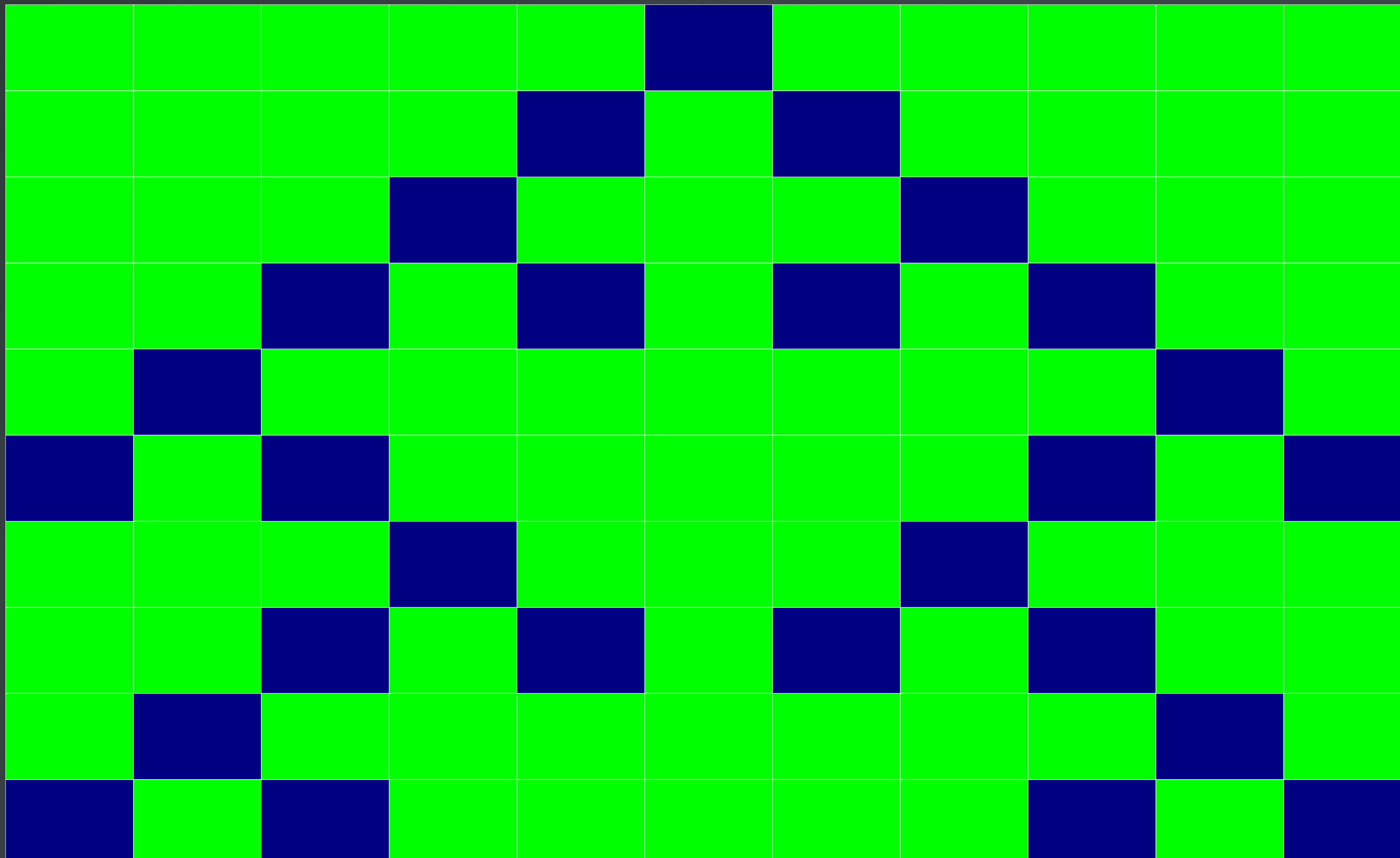












The pattern would be even more complex if our array of cells were larger

Results also depend on the starting configuration we chose a very simple one with one black cell

In fact just the simple machines we have described can generate patterns ranging from trivial to attractive symmetric ones like this, to complex ones and even ones showing non-repeating “random” ones.

Higher Dimensions

In 1970, British mathematician John Conway invented "life" - a two-dimensional finite state machine.

- A cell is "alive" or "dead"
- An "alive" cell stays "alive" if it has two or three "alive" neighbours, otherwise it "dies"
- A "dead" cell becomes "alive" (spawns) if it has exactly three "live" neighnours.

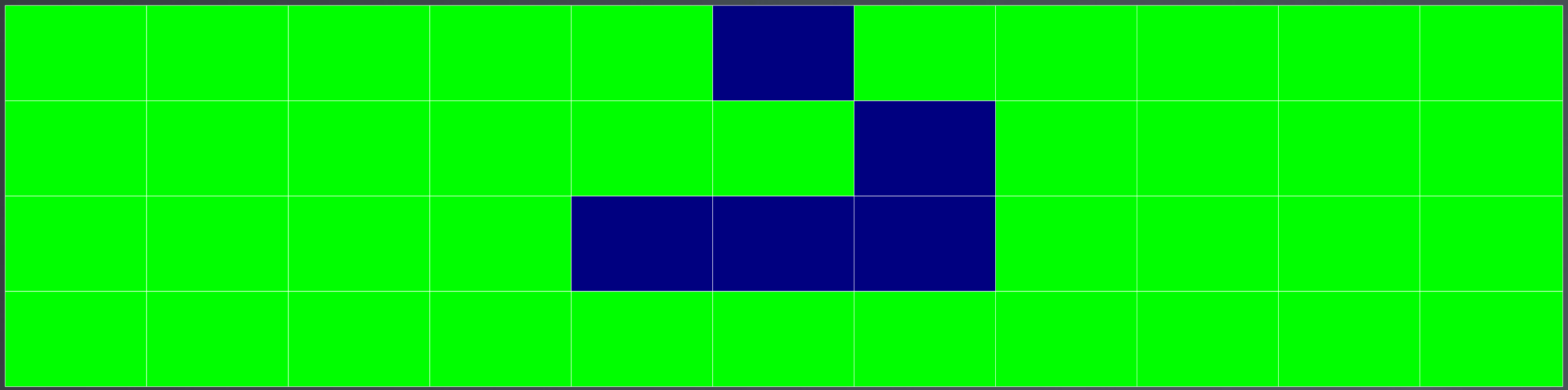
A cell has 8 immediate neighbours, so there are 2^9 (512) rules when expressed as we have been doing, and 2^{512} (about 1.3×10^{154}) possible rule sets.

This is a very complex machine.

Most starting configurations quickly degenerate to nothing, to a few fixed blobs, or a few “blinkers” - patterns that cycle through a few states – or combinations of these.

This is one of the simpler “interesting” patterns

- It is called a glider
- Over successive cycles, it will move down and to the right (until it falls off the endge of the world).



Conway's game is in some sense a universal calculator

- It is possible (but not trivial) to set up initial conditions that will perform “computation”
- For example, you can use blocks of four cells (a stable configuration) to represent numbers, and active elements to move them together, thus doing addition.

Uses of Finite State Machines

2D machines are regularly used in image processing

- Eroding and dilating of masks
- Filling in of gaps
- Smoothing of edges

Speculation

Stephen Wolfram (author of the computer program Mathematica) has suggested that cellular automata offer a better model of the universe than the current continuous mathematical ones.