

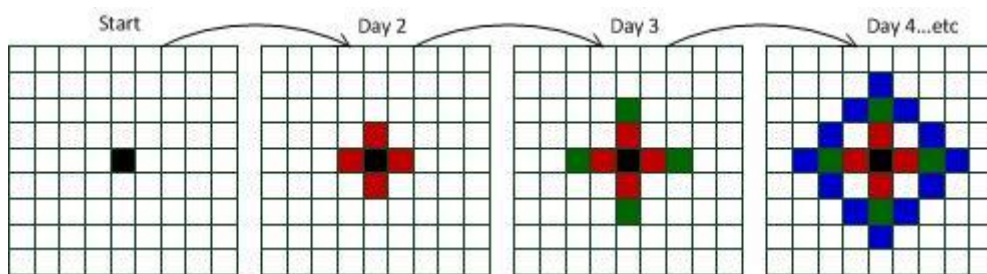
# Lectorial 8: Zombie Apocalypse!

**Fractals** are complex, symmetrical patterns produced by mathematical models that infinitely and recursively generate increasingly smaller patterns that each relate to each other. One of the most famous fractals is the **Mandelbrot Set** named after the mathematician Benoit Mandelbrot.

In this group lectorial task you will be modelling the spread of a 'virus' and showing the spread pattern visually using MATLAB figure windows.

To show this in MATLAB:

- Create a 100 x 100 zeros matrix, **each element of this matrix will represent a person** who can be 'infected' with the virus (i.e. 10,000 people).
- Somewhere in this matrix change a value to '1' to represent '**patient zero**' of our pandemic.
- This infection must spread each day to matrix elements **directly north, south, east and west** of another infected element (elements diagonal to an infected element can't catch the virus from that element).
- Infection must only spread to elements with **one infected neighbour only**, if an element has two or more infected neighbours it cannot be infected (see visual below in Figure 1).



**Figure 1:** Progression of virus for 4 days.

For this lectorial task we will only work on getting the **basic mathematical model** working, in weeks to come we'll look at **viral lifetime** (e.g. contagious for a finite number of iterations/days) and **immunity** (elements that have been infected once cannot be re-infected). Next week we'll also look at statistically analysing the spread of the virus and its distribution over time.

## Hints and tips:

- Use the gridded paper on the next page to help you **design your algorithm by hand**, take note of each index in the grid that you change on each iteration.
- Think about the **logic needed** to determine if an element is to be changed (infected) or left alone (if statements), how will you simulate each day (for/while loops?).
- A **user-defined function** should be used to determine the indices of the elements to be changed, the function should then return these indices to the main program which will then deal with changing them.
- Start simple; perhaps get your function to **just create a line** from the infected element extending out to the right, then build this so the function **generates a cross shape** (extends elements out north, south, east and west) before you **build up to the full pattern**.

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
1															
2															
3															
4															
5															
6															
7															
8															
9															
10															
11															
12															
13															
14															
15															

Write logical steps and notes here: