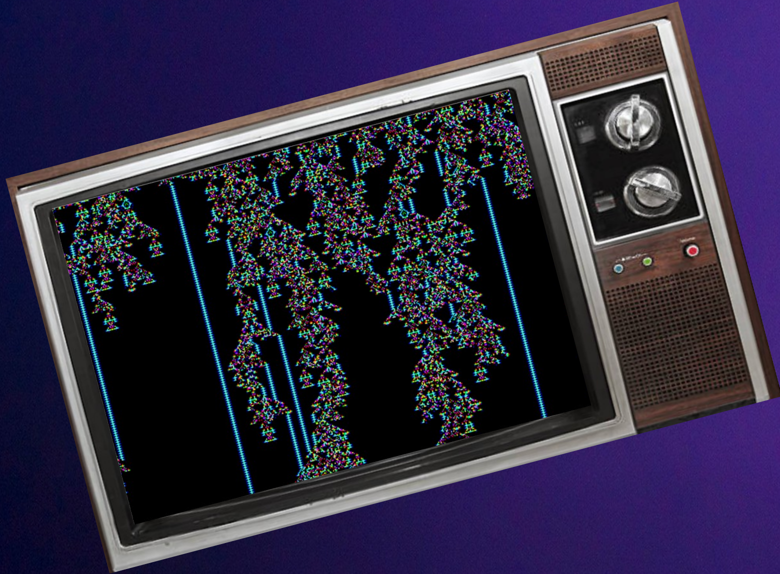


NIHR | Applied Research Collaboration South West Peninsula

@penCHORD_UoE
@peninsula_ARC

**Module 6 : Modelling Behaviour**

Session 6A : Cellular Automata

Dr Daniel Chalk

"Life is the name of the game"

HSMA 5

#hsma5isalive

# Note to Dan

Dear Dan,

Share screen optimised for video for this session!

You forgot didn't you?

I knew it.

Yours,

Dan

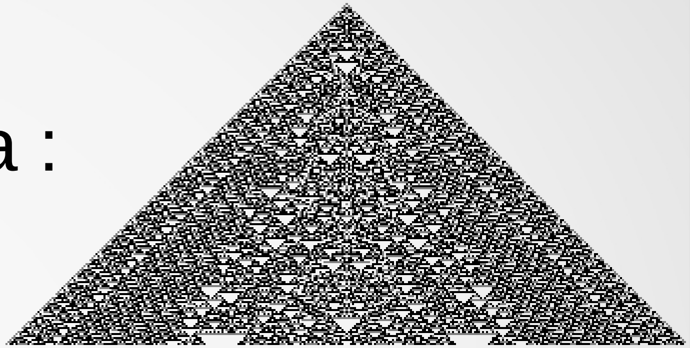
# Modelling Behaviour

Modelling approaches such as Discrete Event Simulation focus on the *processes* within a system.

But sometimes behavioural aspects within a system are important.

In this module, we'll look at ways we can model behaviour using *Agent Based Simulation (ABS)*. In this session, we'll start by looking at a type of computational model known as a *Cellular Automaton* (plural = Cellular Automata), which introduces many of the core concepts of Agent Based Simulation.

# Cellular Automata



Core components of Cellular Automata :

- n-dimensional grids of *cells*
- each cell is in one of a number of different *states*
- time advances in time steps called *generations*
- the states of cells may change in each generation according to *rules*
- the rules that determine the state of a cell in any given generation typically depend on the state of the cell in a previous generation and / or the states of the cells around it (referred to as the *neighbourhood*).

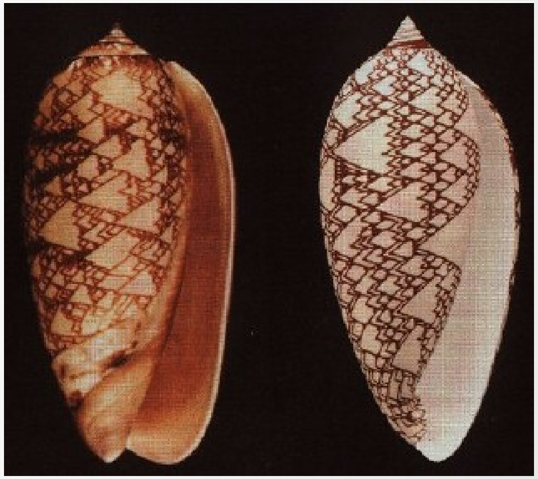
# Why Use Them?

In the natural world, there are phenomena that cannot be explained using conventional mathematics.  Examples include the formation of a snowflake, or the patterns on a seashell.

In addition, we can find examples in human systems, such as the formation of traffic jams, and the spread of crime, that have been modelled using Cellular Automata.

Such phenomena typically arise *emergently*.

In other words, relatively simple rules consistently applied at an individual level can lead to emergent *dynamics* or *behaviours*.



# Ecological Modelling

In ecological modelling, we often know much more about the individual behaviours of animals, but less about the population-level dynamics.

For example, we may know that an animal is an efficient forager that needs to fill up with resource as quickly as possible (e.g. a bumble bee) but less about what the population-level dynamics look like given these individual behaviours (e.g. at the colony level).

Note : in ecology, Agent Based Simulation is often referred to as *Individual Based Modelling (IBM)*.

# From Animals to People

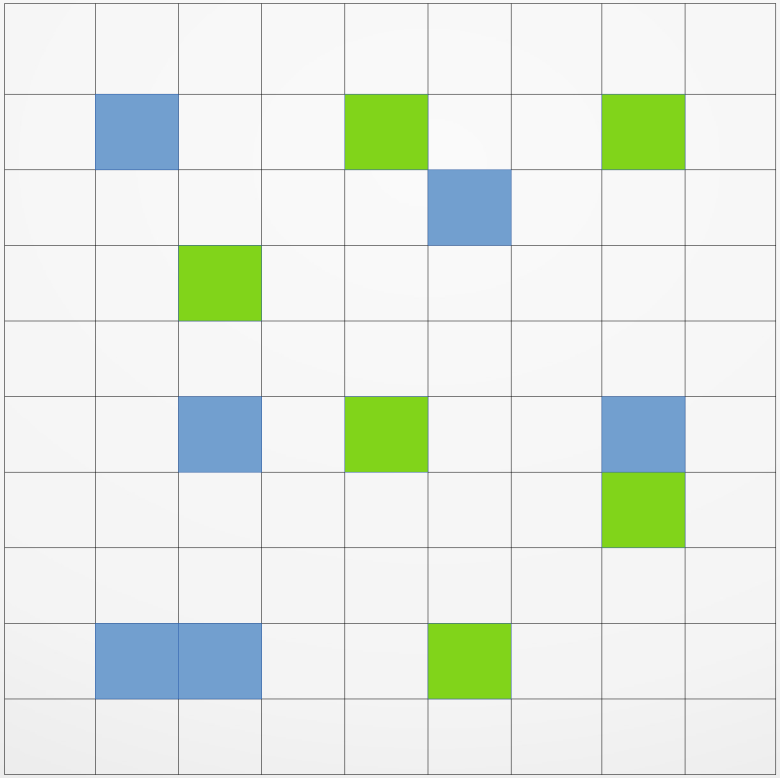
Such applications are not limited to ecology. For example, we may be able to model rules that represent the individual behaviours, motivations and interactions of humans in a system, and want to understand what the emergent dynamics look like at a population / system level.

Can anyone think of a rather topical example that fits that description…?

# Cellular Automaton Structure

Let’s look at an example of what a Cellular Automaton (CA) might look like.

*Grid of cells in 2-dimensional CA*



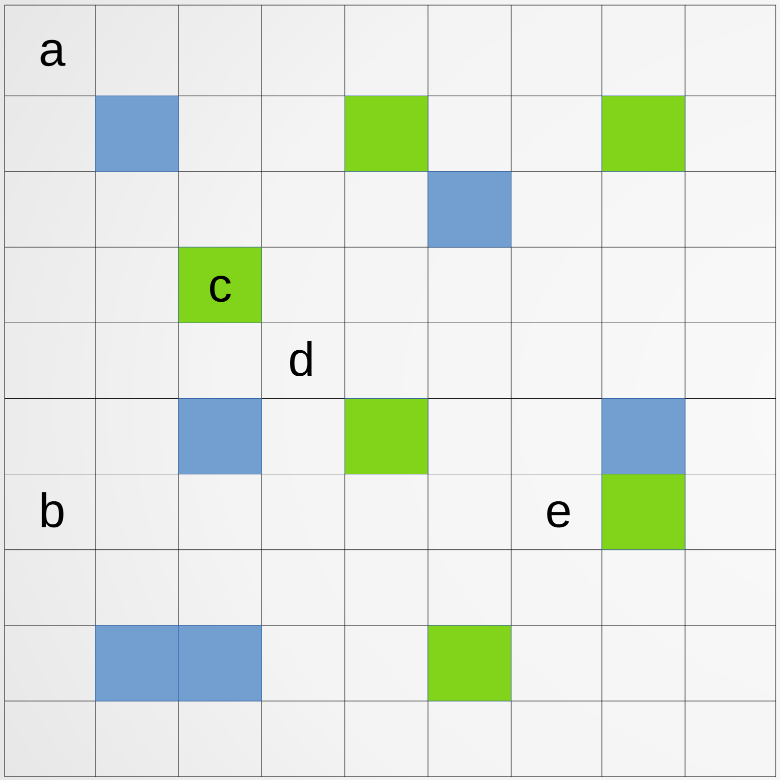
| | | |
|---|---|---|
| n | n | n |
| n | c | n |
| n | n | n |

Neighbourhood = *Moore*
(n cells + the c cell are the neighbourhood of cell c. n cells are the neighbours)

3 states :
- Off
- Blue
- Green

Rules :
1. If cell off in generation *t*, and green and / or blue cells present in neighbours, switch on to Blue or Green depending on which cells are most prevalent in neighbours (breaking ties randomly) in *t+1*
2. If cell Blue or Green in generation *t*, switch off in *t+1*

# Cellular Automaton Structure




What will happen to these cells in the next generation?

- a?
- b?
- c?
- d?
- e?

Rules :
1. If cell off in generation *t*, and green and / or blue cells present in neighbours, switch on to Blue or Green depending on which cells are most prevalent in neighbours (breaking ties randomly) in *t+1*
2. If cell Blue or Green in generation *t*, switch off in *t+1*

# Dimensionality

Cellular Automata can be multi-dimensional beyond 2 dimensions, but most CA are either 1-dimensional or 2-dimensional.

In 2-dimensional CA we have a 2D grid of cells.  In a 1D CA, we have a single row of cells that is typically visualised in 2 dimensions by showing the evolution of the CA over time using the y-axis.

# Elementary Cellular Automata

An *Elementary Cellular Automaton* is a special type of 1D CA in which there are *2 possible states* (0 and 1), and the state of a cell in the next generation depends only on its own state and that of its two immediate neighbours.

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| | | | $n_1$ | c | $n_2$ | | | |

An Elementary CA where the state of cell c in $\text{generation}_{t+1}$ depends solely on the states of cells c, $n_1$ and $n_2$ in $\text{generation}_t$

If each CA has a unique ruleset, then there are 256 possible Elementary Cellular Automata :

- There are 8 possible state configurations for a cell and it's immediate neighbours (e.g. 101, 111, 001 etc) – $2^3$ (2 valid states for each of 3 cells)
- Each of these 8 configurations will output either a 0 or 1 for the cell in question. Therefore, a unique ruleset will specify whether each of the 8 configurations outputs a 0 or a 1. This gives us $2^8$ possible rulesets, where each ruleset can be specified by an *8-bit array* (an array of 8 binary values).

# Elementary Cellular Automata

Stephen Wolfram, a computer scientist famed for his research into Cellular Automata in the 80s, proposed the *Wolfram Code* – in which each ruleset of an Elementary Cellular Automata is allocated a number between 0 and 255.

Some of these rulesets have been shown to have special properties, such as generating complex emergent behaviour. One ruleset – 110 – has been shown to be capable of *universal computation*.

We're not going to go into the details of this, but the key point is that the power of Cellular Automata is that *very simple rules* can lead to *complex emergent behaviour*.

# Elementary CA Simulator

Let’s have a look at a 1D Elementary CA Simulator :
https://bit.ly/3jiSlwz

We’ll check out a few rules, including rule 110 and also rule 30. Rule 30 is an example of a *chaotic* CA – one which produces complex, seemingly random patterns from simple, well-defined rules.

Rule 30 has been used as a random number generator (RNG) and for cryptography. A pattern strongly resembling that generated by Rule 30 can be seen on the shell of the cone snail *Conus textile.*

# The Edge of Chaos

As well as Elementary CA, we can also have other 1D CA.  Imagine a 1D CA that has a number of states (one of which represents a *dead* state, the rest different *alive* states), and a neighbourhood of a given size.

Let's imagine this CA has a rule that says if a cell and it's neighbours are dead, that cell can't come back to life in the next generation.

However, beyond this rule, we will randomly generate all other rules.  We will have a parameter called λ (*lambda)* that we can adjust, which represents the probability of one of the randomly generated rules leading to an alive state (as opposed to a dead state).

In other words :
- as λ tends towards 0, more cells die quickly (because there are few rules that lead to life)
- as λ tends towards 1, more cells continue living (because most of the rules lead to life)

# The Edge of Chaos

There is an abstract concept known as *The Edge of Chaos* which describes the points that lie between order and disorder (chaos). It says that in the edge of chaos, there is great instability as there is a constant interplay between order and disorder.

In terms of Cellular Automata, computer scientist Christopher Langton conducted numerous experiments into the concept of The Edge of Chaos in 1D CA. He found that there is a transitional period in complexity in emergent behaviour in 1D CA that sits between the move from generating patterns (emergent behaviour) that are uninteresting because they are highly ordered, to generating patterns that are uninteresting because they're too chaotic. He put forward the idea of using λ to explore the emergent complexity space and find The Edge of Chaos – an area in which cellular automata can be used for universal computation.

Let's look at how this works :
https://math.hws.edu/eck/js/edge-of-chaos/CA-info.html

# 2D Cellular Automata

2D Cellular Automata share the principles of 1D CA, but there are greater possibilities in terms of neighbourhood configurations.

The two most commonly used neighbourhoods in 2D CA are :

| | | |
|---|---|---|
| | n | |
| n | c | n |
| | n | |

The **von Neuman** neighbourhood, in which the neighbourhood of a cell c is defined as the central cell c and the immediate neighbours in the four cardinal directions (North, South, East, West).

| | | |
|---|---|---|
| n | n | n |
| n | c | n |
| n | n | n |

The **Moore** neighbourhood, in which the neighbourhood of a cell c is defined as the central cell c and all of its surrounding cells.

# The Game of Life

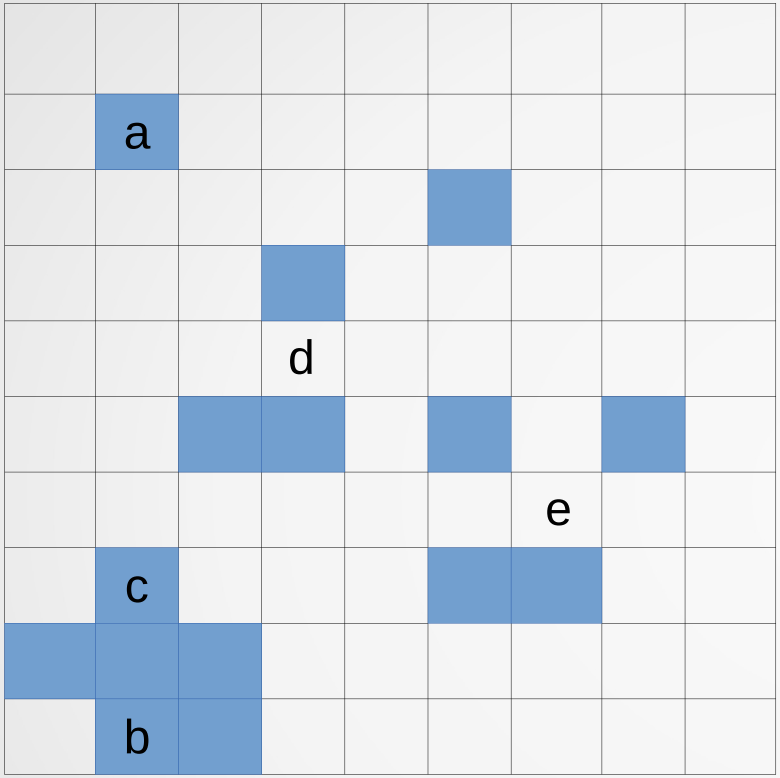
Probably the most famous Cellular Automaton is *Conway's Game of Life* (sometimes referred to simply as *Life*). It was devised by the mathematician John Conway.

Life is a 2D Cellular Automaton, in which each cell can be in one of 2 states – *alive* or *dead*. It uses a *Moore Neighbourhood.*

Life consists of 4 simple rules :

1. If a cell is alive and has fewer than 2 alive neighbours, the cell will die in the next generation (underpopulation)
2. If a cell is alive and has either 2 or 3 alive neighbours, the cell will live on in the next generation
3. If a cell is alive and has more than 3 alive neighbours, the cell will die in the next generation (overpopulation)
4. If a cell is dead and has exactly three alive neighbours, the cell will be "born" (become alive) in the next generation (reproduction)

# Example




What will happen to these cells in the next generation?

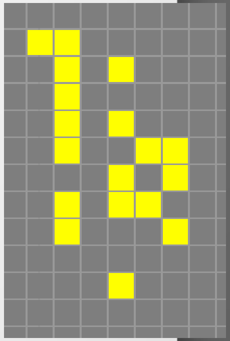
- a?
- b?
- c?
- d?
- e?

1. If a cell is alive and has fewer than 2 alive neighbours, the cell will die in the next generation (underpopulation)
2. If a cell is alive and has either 2 or 3 alive neighbours, the cell will live on in the next generation
3. If a cell is alive and has more than 3 alive neighbours, the cell will die in the next generation (overpopulation)
4. If a cell is dead and has exactly three alive neighbours, the cell will be “born” (become alive) in the next generation (reproduction)

# Patterns

We see that Life consists of only 4 very simple rules.  But out of these simple rules, we find complex behaviour emerging.  Specifically, a number of *Patterns* (or *Life Forms*) have been found to emerge when playing Life.  These patterns fall into different categories, with the main three categories being :

**Still Lifes** : these are patterns that do not change from one generation to the next



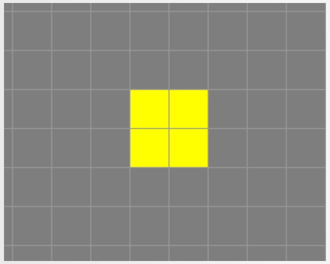
**Oscillators** : these are patterns that repeat after a certain number of generations (known as the *period* of the pattern)

**Spaceships** : these are patterns that move themselves across the grid

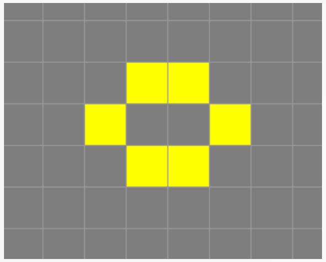
Before we look into these patterns in more detail, let's have a look at Life in action and see what we can spot : https://playgameoflife.com/ (we'll also have a look at https://beltoforion.de/en/game_of_life/)

# Still Lifes

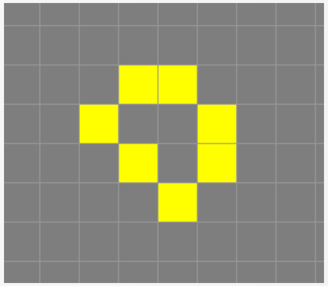
A Still Life pattern does not change from one generation to the next, and is therefore *stable*.  Some common Still Life patterns you'll see emerge are as follows :



**Block**



**Beehive**

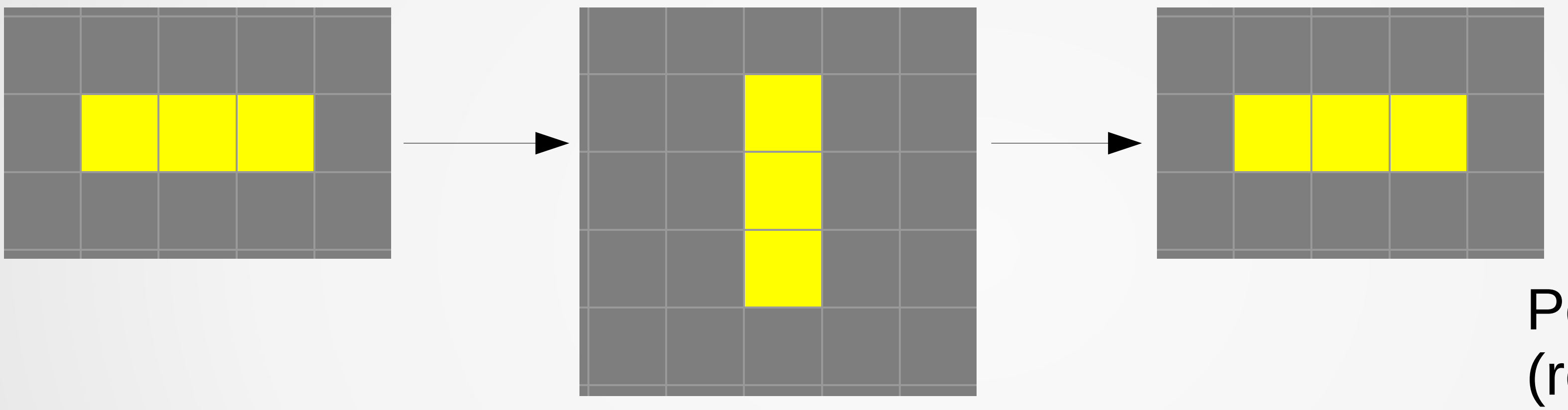


**Loaf**

Take a moment to think about why these patterns are Still Lifes – why do they not change from one generation to the next?

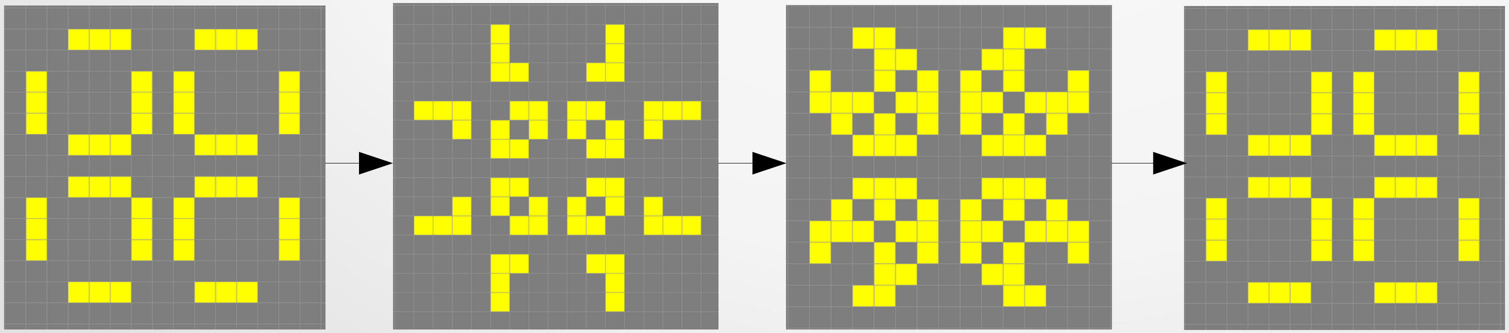
# Oscillators

An Oscillator pattern repeats after a certain number of generations (the *period*). A couple common patterns you'll see emerge are :



**Blinker**

Period = 2
(repeats after 2 generations)

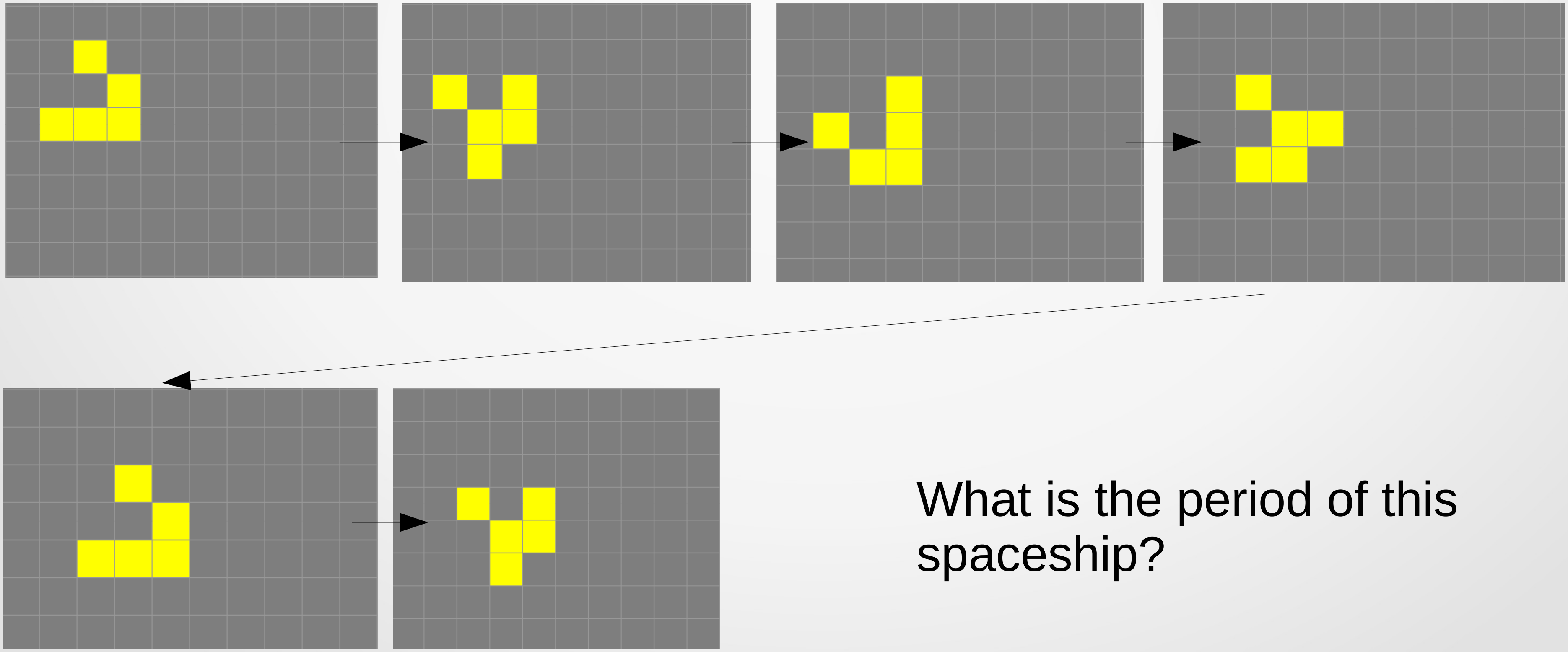


**Pulsar**

Period = 3
(repeats after 3 generations)

# Spaceships

A spaceship is a pattern that moves itself across the grid via a pattern that repeats after a certain period, but in a different location.  Therefore, they can be thought of as Oscillators that move location.  The most common spaceship you'll see emerge is a **Glider** :



What is the period of this spaceship?

# Eaters

Eaters are Still Life patterns that can interact with certain other patterns without suffering *damage*.  It appears that the Eater *consumes* the other life form whilst retaining its own integrity.

Let’s look at an example : https://playgameoflife.com/lexicon/eater

# Pattern Discovery

Patterns have been (and continue to be) discovered.  There is an annual contest called the Pattern of the Year contest in which people submit interesting patterns that they have discovered in Life and the submissions are judged.

You can find out more (including looking at previous entries) here :
https://www.conwaylife.com/wiki/Pattern_of_the_Year

# So What?

So, why should we care about any of this (other than I think it's fascinating!)? Why are patterns important? Well…

- Patterns demonstrate emergent (and self-organising) behaviour at a population level from simple rules at the individual level

- Patterns demonstrate how order can emerge from (apparent) chaos

- If we use a Cellular Automaton to represent a real world system, patterns could help us understand emergent behaviour that may help us design interventions to encourage or combat

# Exercise 1

After a 10 minute comfort break, you'll work in your groups for **80 minutes**. I want you to:

1. Have a play with the 1D Elementary CA Simulator here : https://bit.ly/3jiSlwz and exploring the Edge of Chaos phenomenon here : https://math.hws.edu/eck/js/edge-of-chaos/CA-info.html
2. Play with the Game of Life either here : https://playgameoflife.com/ or here : https://beltoforion.de/en/game_of_life/. See what patterns you can get to emerge, and explore some of the pre-determined patterns (see the "Lexicon" button in the first link, or the different pattern buttons in the second link).
3. Have a look at some other 2D Cellular Automata (such as *Star Wars* and *Brian's Brain*) in the Cellular Automata Simulator here : https://robinforest.net/post/cellular-automata/
4. Once you've had a chance to play around with all this, I want you to, as a group, come up with a design for a Cellular Automaton that could represent a real world problem. You should start by discussing potential application areas where you think CA could potentially help – (think creatively, CA is an abstract approach). Then, once you've come up with some ideas, pick one for which you'll work up a design. You'll need to think about :

- What your grid will represent, and the dimensionality of the grid
- How many states you'll have, and what they'll represent
- What neighbourhood you'll use
- What rules you'll use (remember – the key with CA is *simple rules* leading to *complex emergent behaviour*, so consider how you can distil behavioural rules into simple "rules of thumb")
- What question(s) you want the CA to answer, and how you might use it to inform intervention(s)

At the end of the exercise, I'll ask each group to present their ideas.