Cellular Automata

A crash course intro discrete computational modelling.

Basics

Rules

Rules define how the cells "live" in the game.

Rules are:

1. Simple
2. Conversions
3. Local

Often multiple rules are used in combination to form complex models in the game.

Cells

The game is made up of 2 dimensional matrix of cells.

Each cell has a state.

The state changes by on the rules and the environment.

Environment - other cells.

Examples: Conway's Game of Life

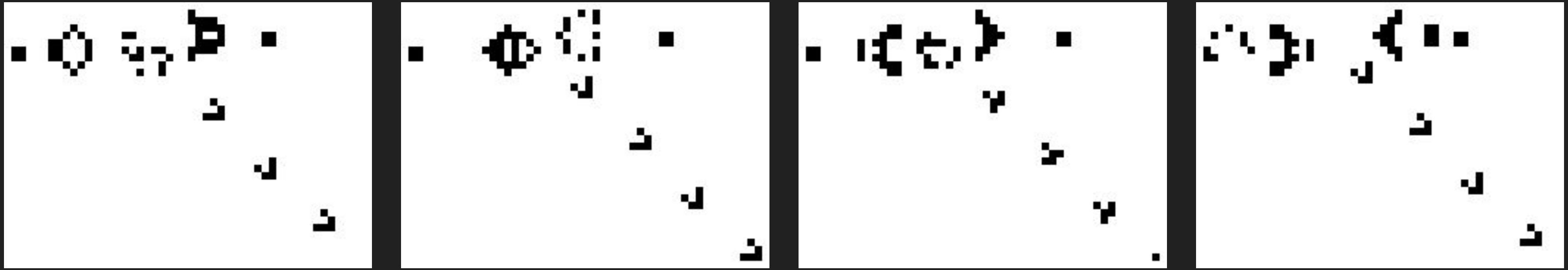
Conway's game of life is based around 2 major rules.

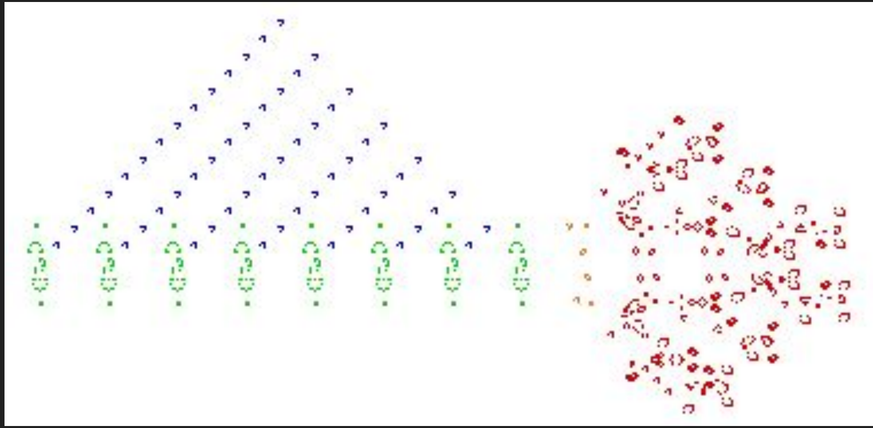
If there are 3 cells alive nearby, bring the cell back to life.

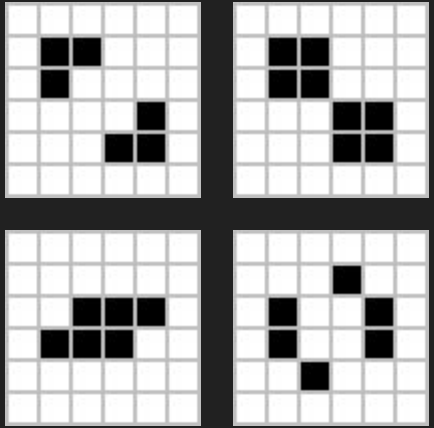
Else if there aren't 2 cells alive nearby, kill the cell.

This allows for some complex and diverse patterns to be formed and is a simple representation of population.

Conway's Game of Life in action







Real world Examples

Yes, cellular automata can also model real world scenarios.

Real World Examples: Virus infection Simulation

4 main states in the grid that represent a community:

Healthy, Infected, Immune, Dead.

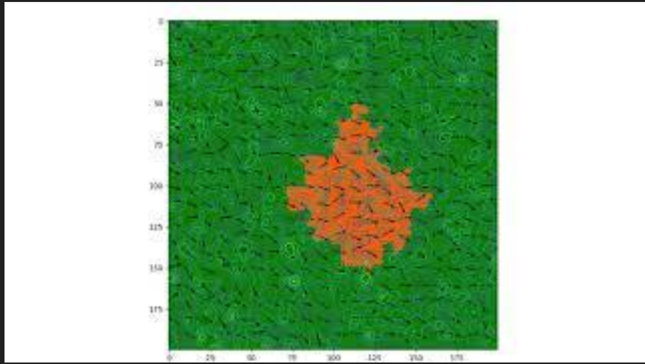
We can develop rules such as:

- If a 'normal' is surrounded by n 'Infected' patients, there is a n / 8 chance of changing to 'Infected'
- A 'Infected' has a ⅛ chance of changing to Immune.
- A 'Infected' has a ¼ chance of changing to Dead.

The variables allow us to simulate different viruses.

Real World Examples: Fire Spreading in Woods

- What happens during a fire?
- Wood catches on fire
- Fire spreads
- How do we simulate fire spreading through Cellular Automata?



1. Normal state
2. Ignition state
3. Spread state
4. Charred state

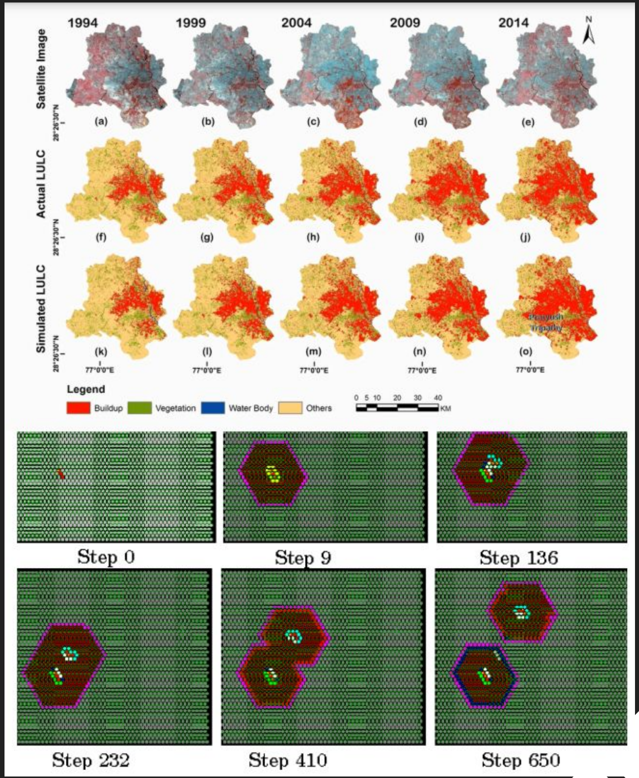
What would we need to do about these states?

More Specific Details

- Ignition
  - If cell is a ‘Tree’ AND neighbouring cell is burning, tree cell becomes burned
- Spread
  - Cell must be a ‘Tree’ cell
  - Cell must be horizontally perpendicular, being 1 cell away
- Burnout
  - If cell has been in ignition state for X step, cell has burned out.

Further Real World Examples

- Terrain generation
- Cell Division (mitosis)
- Predicting ant colony movement
- Flow of blood through vessels
- Consumer behaviour in markets
- Growth of trees and plants
- Any other examples of deterministic growth.
- etc.



Your turn!

Now you have a chance to code your own states and rules

Your Turn

Go to ca.wyli.tech
(custom deployment)

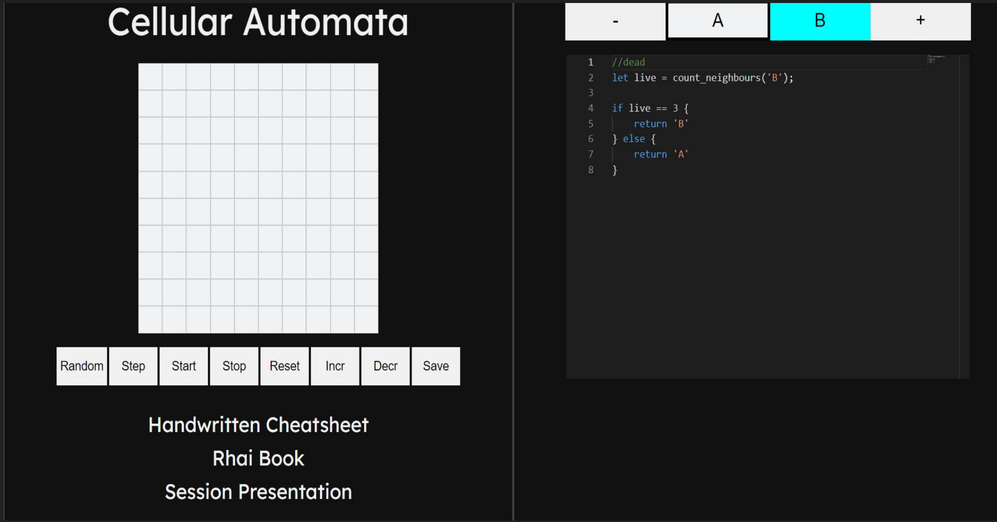
You will be greeted with a page that looks like this:

The white grid represents the cells

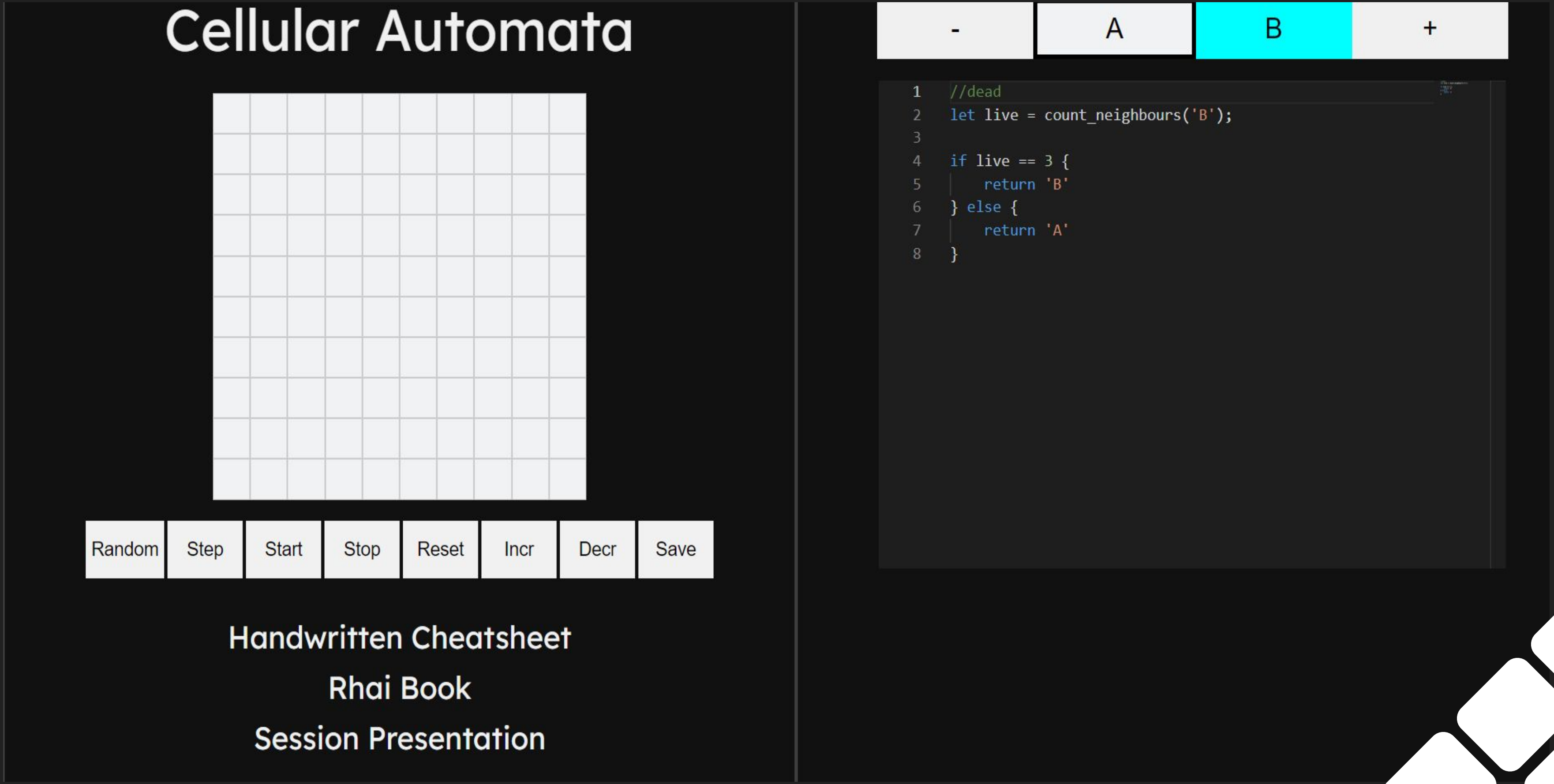
Underneath are controls

The green box on the side is where you script the rules

On top of that are the available states controls.



UI Breakdown



Syntax breakdown

To change the state of a cell, return the state it should change to. `(return 'A')`

To get a random state, invoke `rand_state()`.

invoke `rand(a, b)`, to have a `a/b` chance to return true.

invoke `neighbours()`, which returns a list of neighbour states as chars in a clockwise order from the top right.

invoke `count_neighbours(state)` which returns a integer representing the number of neighbours with the same state.

Implement Game of Life

Try implementing Conway's Game of Life based on the following 2 rules:

1. If there is 3 cells alive nearby, bring the cell back to life.
2. Else if there aren't 2 cells alive nearby, kill the cell.

Implement Spreading of Virus

Try implementing the spread of virus based on the following
1 rules:

1. If infected neighbours is less or equal to 4, 12.5% chance of cell getting infected.
2. If infected neighbours is greater than 4,50% chance of cell getting infected

Implement Fire Spread Simulation

Implement the fire spreading simulation you saw earlier. You must spawn a fire cell on a grid.

1. If there is tree 1 cell away from a burning cell, tree cell burns
2. Some cells cannot be burned
3. After certain amount of time, burning cell becomes burnt out.

Some important notes

- Decrease the grid size so that the change in states from one to another will be easier to see
- If the buttons do not respond, refresh the page
- Save your code once in a while in case of crashes.
- If you call D, make sure you do not <u>delete it</u> — otherwise it will cause a crash!