Conway's Game of Life

Krishna

November 2019

1 What is Conway's Game of life?

Conway's game of life is a simple game whose next state is determined by the pre-existing state of the board.Each cell on the board can be alive(white) or dead(black).To reach the next state of the board, the following rules must be followed:

1. Any live cell with fewer than two live neighbours dies, as if by underpopulation.
2. Any live cell with two or three live neighbours lives on to the next generation.
3. Any live cell with more than three live neighbours dies, as if by overpopulation.
4. Any dead cell with exactly three live neighbours becomes a live cell, as if by reproduction.

After Randomly generating an initial game state, each cell must determine its fate after the next cycle. The game is usually performed over an infinite grid with infinite cycles. Here, we have limited ourselves to a 100 by 100 board and only a few game states. However, this is sufficient enough to notice the different and interesting patterns that begin to emerge.

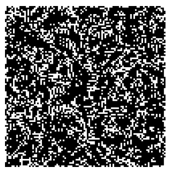

Figure 1: An example board

2 Implementation

"I have chosen to use sagemath here as I am fairly comfortable with it among other functionalities it supports that are required, such as the ones listed below:

1. Ease of use of 2-D arrays
2. Packages that allow for the randomisation of data(for the initial state)
3. Packages to manipulate images using pixels, and also save them pattern-wise.
4. Ease of recursion

The .ipynb and .pdf file contains the code, and the gif has been generated by importing images as layers in gimp and exporting it as a .gif file

Note: In order to deal with confusion on the edge tiles, I have assumed that the all tiles beyond the boundary are dead

3 Explanation of the code

First, I have randomly generated an input state. 0 implies the cell is dead while 1 implies the cell is alive. Next, I have printed the input state using the function img.show(). One thing to note here is that I have magnified the height and width of the image by a factor of 6 each as the output looked too small. Then next block of code recursively generates the next board and then prints it under the name t-'s' Where s is the image number. Finally using gimp, I exported it as a gif.