Testing Center vs. Periphery Theory using Cellular Automata

LING 4411
Benjamin Gillott

Note: All relevant code for this project is accessible through this link:

https://github.com/ben-gillott/conway-and-the-snail

Introduction

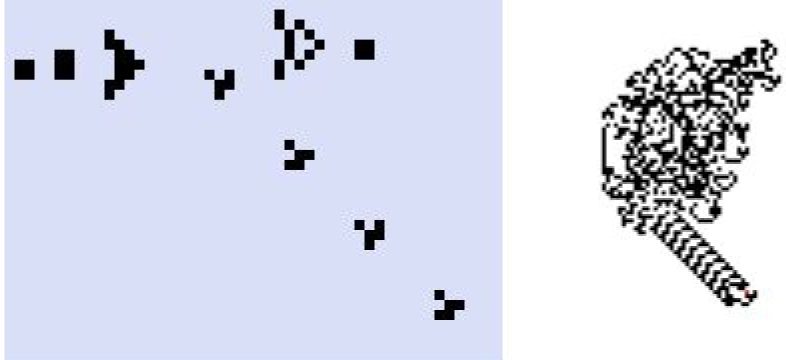
A language can be seen as a form of a living organism, at least in some regards (1). The goal of this paper and project is to model specific language change over time using a simplified simulation of life.

Since we cannot capture an actual life form in a simulation, we will need a simple replacement. One method of simulating life is cellular automata, which will serve as our basic approximation of a living organism. A cellular automaton is a grid of “cells”, with each cell possessing a certain state. Over time the state of each cell changes based on the state of its surrounding cells (6).

The goal of this research was to simulate linguistics change in Japanese over time. Specifically, to compare results generated by a cellular automata simulation with Yanagita Kunio’s “Theory of the Snail”. This is also known as the center versus periphery theory, which describes a ripple pattern in language change. The end goal is to see if the results of the two models differ, and if so in what ways.

Cellular Automata

Cellular automata can be loosely described as any simulation that operates based on a set of cells, or values, and rules governing their changes in state (1). Some famous examples of cellular automata include Conway's Game of Life and Langton's Ant, both of which are pictured below.



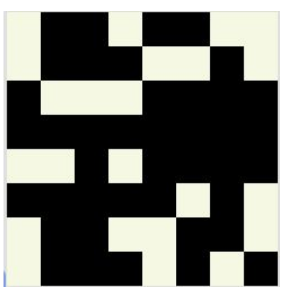
(left: Conway's Game of Life [7], right: Langton's Ant [8])

Both of these automata function through the basic premise of changing the state of a cell over time. The resulting state always depends on the state of the surrounding cells.

Cellular automata has previously been used in linguistics to model multilingualism in Spain [1]. In this study very similar rules to Conway's Game of Life were used, however cell states represented speakers of Catalan, primary Spanish, or bilingualism. The simulation was set up to represent current Spain, and see how the two languages might interact over time. While the goals of that paper were rather different from this project, it is a reassurance that cellular automata can be used to model linguistic change if done correctly.

Simulation

The implementation of our cellular automaton for this project stores values for each cell in a two dimensional array of values. The program defines a per cell value calculation function, which is the critical logic of the simulation. A time stepping function applies the value function to each element of the array, and creates the next “time step” of the automaton. The resulting values are drawn to the screen. Below is an example of a state from a Conway’s Game of Life simulation, with the two values represented by black and white.



Programming Languages

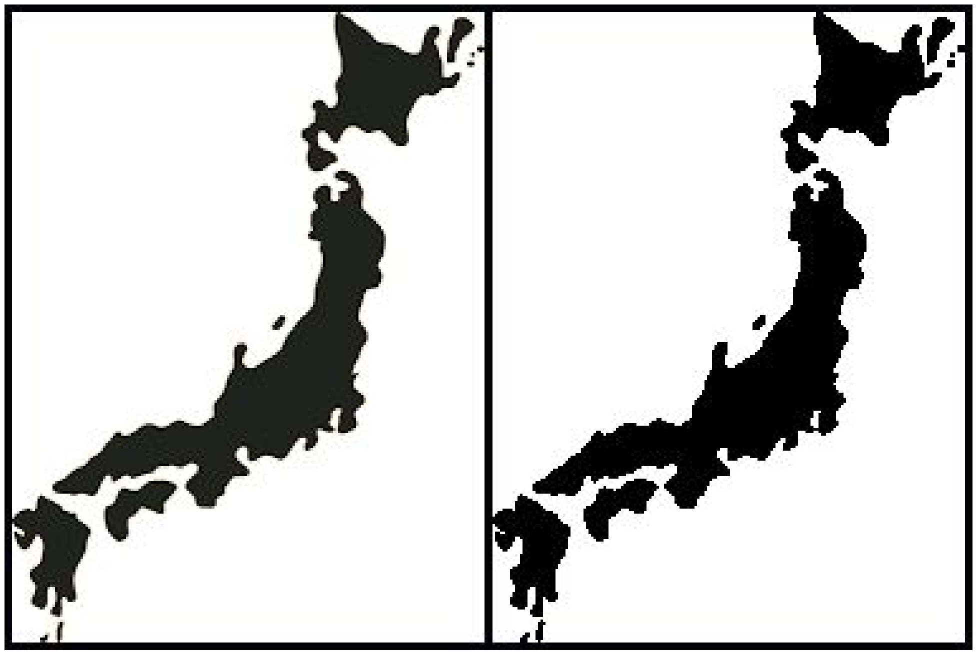
I used a combination of the HTML and JavaScript (JS) languages for this project. These languages are useful as it is relatively simple to run as a website, and therefore lends itself to rapid prototyping. HTML handles the frontend and simulation display, while the JS code runs the “logic” of the simulation.

There are some limitations to this choice of languages. The primary setbacks are that they are not very efficient, and do not offer control over low level computer architecture. This means that the simulation needs to be somewhat limited in its level of detail to reduce processing strain. While an efficient language such as C or C++ could make a more detailed simulation, it seemed worth the trade off, as that would have greatly increased the difficulty of development.

Program Setup

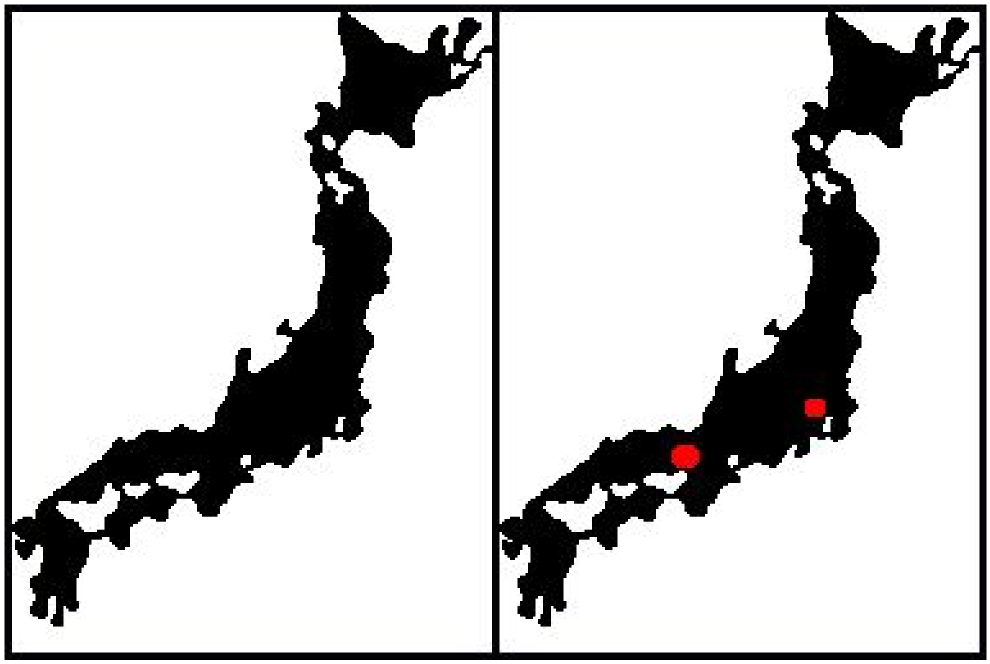
Before beginning the simulation some setup work needed to be done outside of the actual cellular automata logic. The main issue was mapping the shape of Japan's land mass onto the simulation. I did this by generating a map image of Japan and storing reading each pixel of it to get necessary information.

I first used a photo editing tool to remove anti-aliasing, which is a technique used to blur lines using middling color values, in order to make sure the map only had a finite set of colors when reading it.



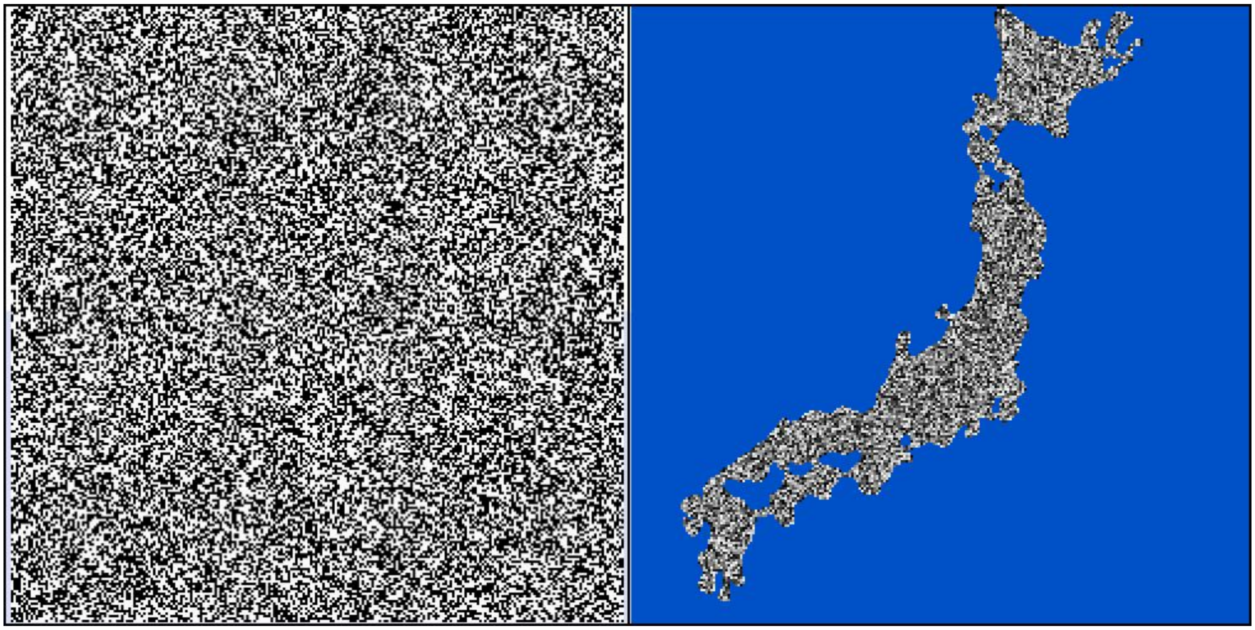
(left to right, a: with anti-aliasing, b:anti-aliasing removed)

I next added “bridges” between all landmasses to ensure the simulation can reach all parts of the map (c), as well as red pixels to mark areas near Kyoto and Tokyo, in order to later test the theory of the snail (d), as these areas would be used as starting points for linguistic change.



(left to right: c: bridges added, d: centers marked)

Using the map I then sampled each of its pixels using JavaScript to get a corresponding “map value” for the simulation. I was then able to read from these “map values” in order to draw the simulation correctly. Below is randomized black and white cells both before (e) and after adding the map values (f).



(left to right: e: random 250x250, f: mapped 250x250)

Before proceeding to the cell functions there is an important clarification to make on how values are displayed. A pixel's color is represented by three values, [R,G,B], representing red green and blue. Each of these values range from 0 to 255. In most cellular automata there are a small number of states a cell can take on, such as the 1 or 0 in Conway's Game of Life (7), which are drawn as white or black. However a pixel can easily represent more complex values, such as by drawing a "greyscale" value, between white and black.

In the simulations below black is used to denote a low value, while colors closer to white mark higher numbers.

Linguistic Assumptions

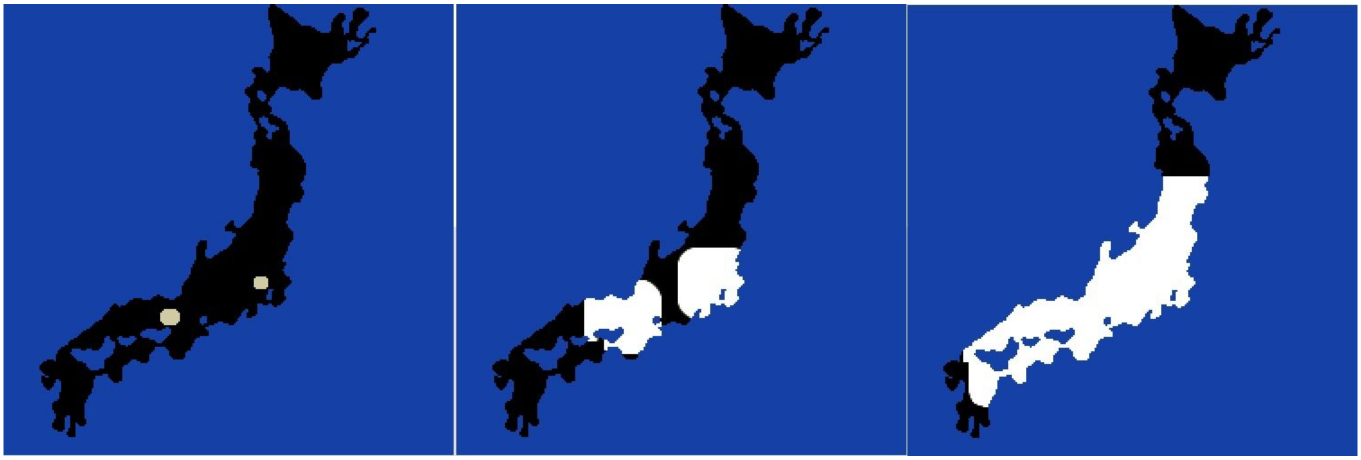
To design the basic simulations shown below, I made somewhat aggressive linguistic assumptions to facilitate easier programming. It could be much more interesting to see simulations without these assumptions, however I was unable to in the given time window.

Firstly, language state was represented as a number, with low values representing an early state of the language, and higher values representing later state. This differs from the theory of the snail (2), as that had a small handful of states, representing the number of terms available for use. It would actually be much more interesting to run the simulation with a small handful of states, as this would be closer to Yanagita Kunio's theory, and it would be likely to produce a more varied progression.

It is also assumed that there is only one "strain" of language, and that it can only develop linearly. It cannot regress or deviate. It would be possible to add more strains to enable a rendition of parallel evolution of language.

Basic Simulation Results

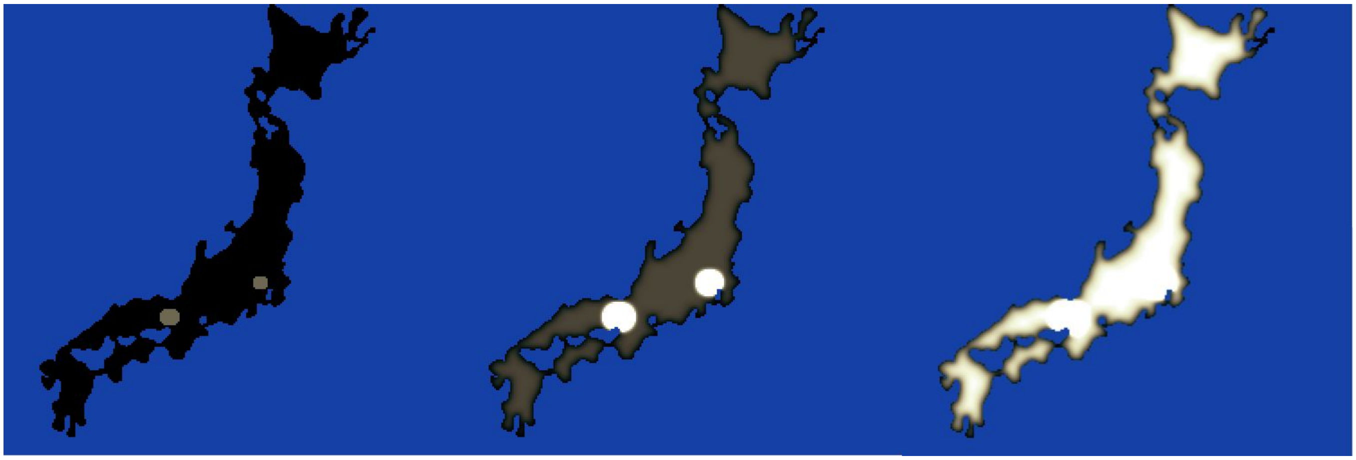
The first simulation had each cell immediately adapt to the highest value of the cells around it. This resulted in a very clean and linear progression, with a boundary of the change expanding outward. This is a very stable and oversimplified representation of progression. However it does give some insight into how new changes merge together when coming from two different points, as can be seen in the south of the third time step below.



(g: linear, time progression from left to right)

The next rendition had a smoother “diffusion” algorithm. This meant that each cell looked at the average of the values of neighboring cells, and evolved based on that sum and a constant growth factor. It also enabled language development outside of Kyoto and Tokyo regions, but with a much lower starting growth.

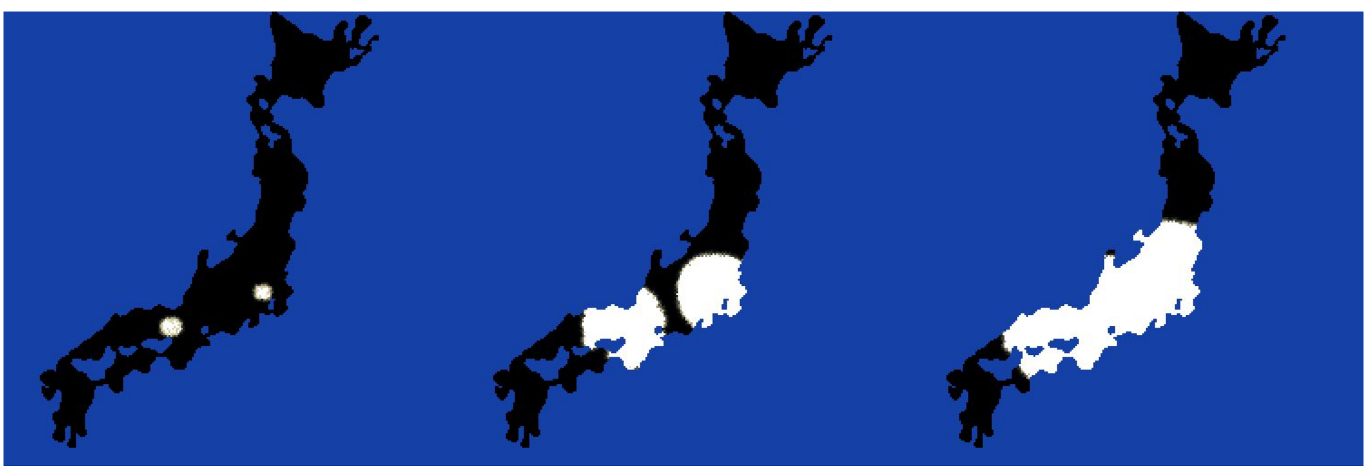
This was a much smoother and more interesting simulation. There are issues in how quickly language develops outside of the given centers, which makes it difficult to see progression. However this could be quite interesting and accurate if implementing the multiple strains of language idea mentioned above, as it would allow new strains to develop simultaneously.



(h: diffusion, time progression from left to right)

The final simulation run had a cell growth equation similar to the one seen in diffusion. It was different as it removed the growth in areas aside from Tokyo and Kyoto, and added a randomization factor that affected how well a cell's language state would spread to a neighboring cell.

This resulted in a smooth, pseudo-random progression across the landmass. If combined with some more advanced assumptions, I believe this simulation would be the most effective in testing the theory of the snail.



(i: randomized, time progression from left to right)

Future Directions:

Next steps include having fewer states rather than a gradient, and adding parallel evolution of language as mentioned above. It also be useful to improve the visualization tool using more color than a simple greyscale, as it would provide much more information. Finally, the end goal after creating a more interesting simulation would be to look at actual data, such as the maps of the use of ででむし or other terms for snail, and see how the simulations compares to historical evidence.

Results:

Thus far it is difficult to draw conclusions from the simulations run. It is possible to see elements of circular spreading as described in the center vs. periphery theory (2) in the third simulation (i). However, without the addition of more advanced rulesets on cell growth, it is a somewhat meaningless comparison.

This research has the potential to show interesting results. A majority of the time spent on this project went into setting up the basic program, with about 25-30 hours of coding and bugfixing, and I was not able to complete as much of the more advanced linguistic analysis I was hoping to. However with this initial setup complete new methods can be developed much more quickly. I intend to continue developing over the next few months, and the program and updates will be available at the link at the beginning of this paper.

Bibliography

Note: The images used in this paper were created by me (Benjamin Gillott), unless cited otherwise.

[1] Beltran, Francesc S., Herrando, Salvador, Ferreres, Doris, Adell, Marc-Antoni, Estreder, Violant and Ruiz-Soler, Marcos (2009). 'Forecasting a Language Shift Based on Cellular Automata'. *Journal of Artificial Societies and Social Simulation* 12(3)5 <http://jasss.soc.surrey.ac.uk/12/3/5.html>.

[2] Center Versus Periphery (2019, February 24) Retrieved from https://en.wikipedia.org/wiki/Center_versus_periphery

[6] Cellular Automaton (2019, December 16) Retrieved from https://en.wikipedia.org/wiki/Cellular_automaton

[7] Conway's Game of Life. (2019, December 17). Retrieved from https://en.wikipedia.org/wiki/Conway's_Game_of_Life

[8] Langton’s Ant (2019, December 15) Retrieved from https://en.wikipedia.org/wiki/Langton%27s_ant