Interim Report - Sustained Complexity of Intelligent Systems

Eli Stone, Lucas Saldyt

The universe is governed by a set of semi-deterministic rules that result in the evolution of complex systems[1]. Our simulation attempts to do the same: beginning with a simple rule set, create a complex system. Ideally, the number of rules should be minimized, while the complexity of the simulation's output should be maximized. This project is intended as an expansion on Conway's game of life[2] where cell positions are *not* discrete. However, direct interventions (Like SproutLife's[3] conversion of 2x2 squares to r-pentomino shapes) with the system should be avoided, as this kind of intervention is too heavy-handed and situation specific. Instead, rules should be general.

By analyzing the data generated by openlife, the initial rule set can be adjusted to maximize complexity[4]. By tweaking parameters individually, those which are volatile (increase complexity) can be kept, and those that are insignificant (don't affect complexity output) can be discarded.

We have created simulation in C/C++ for study. Our simulation creates an initial random distribution of "cells" (semi-intelligent agents of our system), and saves their movement over time. These cells interact with each other in an attempt to maximize their lifespan through simple rules[5]. If a cell group is too crowded, move away from other cells, and if the cells are too sparse but close enough to see others, move towards the center of mass of these other cells. This produces emergent complexity as cells multiply and spread out until running out of area, then die out. Simulation data is recorded after a certain number of timesteps and then analyzed.

Our analysis is done through a set of python scripts that focus on networks between cells: Networks are built based on the proximity of cells, and then their size and count are logged to estimate complexity. The effectiveness of our simulation is based on achieving a high level of complexity for a long period of time. We intend to expand on this testing software and also find the system's entropy[6] and other variables in this network.

We want our simulation to have a minimal rule set, but still produce a large degree of complexity over an indefinite time period. By changing parameters in our simulation, we will find which factors (and which rule sets) affect maximum complexity the most. We will change our simulation by discarding non-influencing factors and adding influencing factors.

Bibliography

[1] Aaronson, Scott. "Quantifying the Rise and Fall of Complexity in Closed Systems: The Coffee Automaton." http://www.scottaaronson.com/papers/coffee2.pdf

[2] Stanford University. "Game of Life." 07 Dec. 2016.
http://web.stanford.edu/~cdebs/GameOfLife/

[3] Shapiro, Alex. "Sproutlife, evolving of conway's game of life"
https://github.com/ShprAlex/SproutLife/blob/master/README.md

[4] Ferreira, P. "*Tracing Complexity Theory*". *Powerpoint*. : Stanford, Oct. 2001. Ppt.
http://web.mit.edu/esd.83/www/notebook/Complexity%20Theory.ppt

[5] Grobstein, Paul, and Laura Cyckowski. "Deterministic and Non-deterministic Emergence." *Deterministic and Non-deterministic Emergence | Serendip Studio*. Serendip Studio, 11 Dec. 2008. 07 Dec. 2016. http://serendip.brynmawr.edu/exchange/ca

[6] Bhatia, Aatish, "*Entropy Explained, With Sheep: From Melting Ice Cubes to a Mystery About Time.*" https://aatishb.github.io/entropy/