TJHSST Computer Systems Lab Senior Research Project
Playing God: The Engineering of Functional Designs in the Game of Life
2007-2008

Liban Mohamed

May 28, 2008

Abstract

Conway's Game of Life is a set of rules in a two dimensional cellular automata grid. This ruleset was specifically chosen by John Conway for the ability to create stable patterns as well as the difficulty of creating patterns which grow without bound. This difficulty was rather quickly overcome by Bill Gosper's glider gun, which opened up the ability to create binary computational devices such as logic gates. As soon as the possibility of binary computational devices in the Game of Life was discovered, it was realized that patterns could be designed in the Game of Life which could symbolically carry out computations. This project endeavors to facilitate in the design and creation of functional patterns in the Game of Life.

1 Introduction

1.1 Scope of Study

The first portion of this project involves the design and coding of a high functioning Game of Life interface. The purpose of the creation of this interface is the assistance it is able to provide with the design and creation of functional patterns in the Game of Life, and thus represents the bulk of this project. This section further breaks down into two parts: the design, and the creation. The creation of patterns in the game of life is assisted by a highly flexible Game of Life editor, with features so abundant they're falling out of its ears. The second portion of the interface is a program which assists in the design of functional patterns.

The second part of this project involves using the previously created interface to create a computation device in the Game of Life, and explore the time and space limitations of such a computational device. This second part of the project is what necessiates the first part; it would be virtually impossible to create a computational device in the Game of Life without first having created an extremely high functioning interface and then creating the search programs. Specifically, the second objective of this project is to develop a calculator implemented with the Game of Life. This will require the development of algorithms useable and relatively efficient in the Game of Life. After a four function calculator is created, functions such as exponentiation will be added, though one major failing of any Game of Life implementation is the difficulty of representing numbers not in the whole number set. This is possible, however, and will be another extension of the computing device.

1.2 Type of research

This project involves pure applied research. None of the problems that this project attemts to address have not been tackled before and the trick will to find, using computer searches, patterns which have the functionalities that will be required. This will all be new to me, but it is the nature of cellular automata that fundamental understanding is impossible to achieve: cellular automata are noteable precisely for the ability to defy prediction.

2 Background

As previously mentioned, a man by the name of Paul Rendell spent a number of years developing a finite Turing machine extensible to universality in Conway's Game of Life. His project is the extreme of complexity, but others have created logic gates, most notably Andrew Adamatzky's LogiCell, presented in his *Collision Based Computing.* With respect to Rendell, two rather large extensions are possible on his work. Firstly, he created his Turing machine with extensiblity to universality specifically in mind, meaing one could use his Turing machine as a template to create a universal Turing machine. In addition to that, Rendell is currently working on create a stack cell generator, which would generate tape for his Turing machine, effectively giving it infinite tape.

This project differs from the other projects in that it endeavors to create a product, a calculator, which must have a useable user interface. My approach differs from that of Adamatzky's because I will endeavor to create one multipurpose design which will perform all of the functions and give understandeable outputs, as opposed to creating a number of different circuits, the outputs of which would hinge on the state of a specific cell at some arbitrary time.

3 Development

The current edition of my program functions to run any 2 dimensional cellular automata setup using von Neumann neighborhoods. The inputs may be either clicked into the grid or loaded from a text file of on and off states. In the interface, a number of useful functions have been implemented, including the selection and running of sub-grids, copying and pasting rectangular areas of the grid, clearing selected areas, and the aforementioned saving and loading of patterns. The program is highly efficient, running at up to 200Hz in the Systems Lab computers.

Although the next portion of the interface - the portion which assists in the design of functional patterns - is not currently functioning, one significant development is the ability of the user to interact with the graphical interface by selecting, clearing, copying, and pasting cells on both the input and the output grids. This is not, of course, the final form that the project will take, but the expectations for the rest of the time are detailed elsewhere.

Analysis for the functioning of my interface and search programs will be done by evaluating the speed of the programs. The frequency of generations given the activity in a grid is the only thing that can be quantitatively measured about a program that acts a simple cellular automata interface. For the search program, the analysis will be carried out by evaluating the effect of the addition of each constraint. Different constraints will have varying utilities, and the number of unnecessary, redunant, or unwanted results that are eliminated by modifying constraints will be tabulated to compare their effectiveness.

Finally, analysis of the pattern which I use my programs to design within the Game of Life will be done by observing the amount of time and space that the various parts of the computational pattern require.

4 Results

At the end of this project, I can expect a program which functions to assist in the design and creation of functional patterns in the Game of Life and one such functional pattern whose design has been assisted by my previously created programs.

4.1 Discussion

The first purpose of this project was to create a Game of Life interface that exhibits efficiency and flexibility. Although the program was written in Java, which to some extent limits the efficiency of the interface, the program is able to run at up to 200Hz while displaying every change in state. For the flexibility aspect, the interface excels; the following functions have been implemented: the selection and running of sub-grids, copying and pasting rectangular areas of the grid, clearing selected areas, and the saving and loading of patterns. In addition to loading patterns from a text file to paste onto a grid, it is possible to load a full size grid.

4.2 Conclusion

coming when I have concluded the project

4.3 Recommedations

The first recommendation that I have for extensions of this project is the extension of the project outside of the Game of Life. Though the Game of Life is very useful and was selected for its traits, it is hardly the only useful set of rules for cellular automata, and for many simulations is entirely useless. The extension into other sets of cellular automata rules could include the creation of a program similar to the one that has been created in this project to assist in the design of functional patterns, but instead would assist in the selection of rule sets.

5 Appendices

none yet

References

[1] www.rennard.org/alife/english/logicellgb.html

[2] Collision Based Computing, by Andrew Adamatzky