3D Cellular Automata

Philip Davies
Assistant Director HE
Bournemouth and Poole College
Bournemouth, UK
pdavies@bpc.ac.uk

Abstract

A cellular automaton (CA) is a set of rules which determines the state of individual cells on a grid, based on neighbourhood relations. CAs have been used by researchers to model a wide range of systems from cell growth to cosmology to universal computation. However nearly all such models have been on one or two dimensional grids. This article provides a brief history of the development of CAs and then extends the models to three dimensions using open source software; Blender and Python. New 3D rules are examined and the development of 3D cell configurations explored and visualized.
Keywords – Cellular Automaton, Game of Life, 3D

I. Introduction and Brief History

Cellular Automata were invented in 1947 by John von Neumann who began by asking the question, “What kind of logical organization is sufficient for a system to be able to reproduce itself?” [1]

Von Neumann constructed a two-dimensional square grid of cells where each cell had one of 29 possible states. A set of rules (state-transition function) determined how the states transform into other states. According to the set of rules, one cell changes its state depending on the state of its neighbouring cells. All cells undergo transition synchronously, in step with a universal "clock" which determines the passing generations.

With this simple grid of cells, and some state-transition rules von Neumann was able to create a Universal Constructor that was the first self-reproducing system.

Improvements followed in 1968 when Edgar Codd, of relational database fame, showed that it was possible to make a self-reproducing machine with fewer states. In his CA only 8 states were required instead of 29 [2].This was further improved in 1972 when Edwin Roger Banks produced a 4-state CA. [3]

In 1974 John Conway introduced a new two-state, two-dimensional cellular automaton which he called the Game of Life and which became the most widely known of all the cellular automata through its publication by Martin Gardner in Scientific American. We will return to this version shortly.

In 1984, Christopher Langton extended Codd's cellular automaton and created what was called Langton's Loops[4], which used only a small number of cells compared to previous attempts.

However, the signs were evident some years earlier that cellular automata were being taken more seriously by the physics community and considered as more than just a mere computational amusement. In 1969, the German computer pioneer Konrad Zuse published the first book on digital physics with his work on Calculating Space, which proposed for the first time that the physical laws of the universe are discrete by nature, and could be modelled as the output of a giant cellular automaton

This was followed in 1983 by Stephen Wolfram, the author of Mathematica, who published the first of a series of papers systematically investigating the most basic class of all cellular automata – one dimensional arrays. Despite the simplification of operating in one dimension, Wolfram showed that even this space demonstrated evidence of unexpected complexity. Wolfram followed this up in 2002 with a vast 1280-page book called “A New Kind of Science” in which he painstakingly catalogues the behaviour of one and two dimensional cellular automata and seeks to draw inferences and analogies with physics, biology chemistry and other sciences. He argues that the study of cellular automata is rich enough to merit its own discipline of science where the complexities found in nature may be due more to these mechanisms and modelled better by CA than by differential equations in some instances.

Cellular Automata continue to be explored and it even has its own Journal of Cellular Automata. Here we seek to add to the growing body of knowledge on Cellular automata by extending the work into three dimensions – a realm which has had very little exploration, possibly because of the vastness of the subject area.

The structure of this paper is as follows: Section 2 examines in more detail one particular cellular automata, Conway's Game of Life as this has proved the most investigated system in two dimensions. In Section 3, we show how this can be extended into three dimensions and the system visualised using open source software Blender and Python. In Section 4, we start to engage in a simple classification of some of the most primitive component in the 3D system. Section 5 is a conclusion and discussion of future work.

II. CONWAY'S GAME OF LIFE

It might be justly argued that the most significant step forward in cellular automata came in 1974 when John Conway introduced a new two-state, two-dimensional system which he called the Game of Life and which became widely know through its popularization by Martin Gardner in his Scientific American column.

The rules of The Game of Life were as follows: The two states of a cell were conveniently depicted as black and white. There are three state rules

1. If a black cell has 2 or 3 black neighbours, it stays black.
2. If a black cell has less than 2 or more than 3 black neighbours it becomes white.
3. If a white cell has 3 black neighbours, it becomes black.

Despite the simplicity and strict determinism of the rules, the system is able to demonstrate a remarkable variety of behaviours which range from apparent randomness to strict order.

A tabulation of life forms and terms has been constructed.

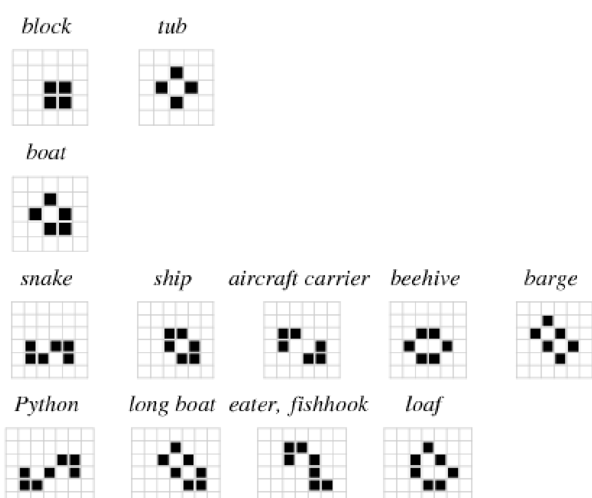


Figure 1: Basic Game of Life shapes (from Wolfram http://mathworld.wolfram.com/Life.html)

One of the most significant features of the Game of Life in particular and CAs in general is the generation of seemingly independent structures that have an existence and permanence and which exhibit a set of unexpected and unpredictable behaviours. Five types of pattern are evident:

1. A pattern which does not change from one generation to the next is called 'still life'
2. A pattern that flips or oscillates between one state and another is called an oscillator.
3. A pattern which expands and then dies out after a finite number of generations is a transient
4. A pattern which expands forever and produces an infinite number of cells is called a garden of Eden
5. Finally, and possibly most interestingly a pattern may show a kinematic permanence and move across the grid. One such configuration being the Glider.

Group 5 which shows kinematic permanence provides entrance to a new level of description. At this 'configuration level', new rules of behaviour identified with each configuration are found. Gliders and other similar patterns can be arranged so that dynamic physical principles can be modelled. Furthermore, it has been shown that the gliders can be made to interact in such a way as to perform computations. More recently, it has been shown that the Game of Life can emulate a universal Turing machine which means that it can model any computation on any computer.

III. EXTENSION TO THREE DIMENSIONS

With this universality in mind, the Conway Game of Life has, in this paper, been taken as the starting point for the extension into three dimensions. Exactly the same rules as the two dimensional case have been adopted and using the Python scripting language have been developed for visualization using the open source 3D modelling application Blender. This provides the means to explore the development of 3D patterns in a powerfully visual way with full 3D access to the structures created. What is does not do is provide a fully dynamic system since each generation needs to be run separately. The grid was limited to 50x50x50 cells as larger grids were not easily computable within Blender which could not cope with larger cell groups.

Random population

The first run began by populating the 50x50x50 grid with 1% random scattering of cells. This was then run through a number of iterations or generations with snapshots taken at appropriate points. A maximum of 20 generations was run as an apparent equilibrium was reached before that point. Figures 2 to 6 show significant generation points with the last figure showing an internal view of the structure of the cells. The figures show a characteristic development of random distribution moving into a sparse but ‘clumpy’ arrangement which then generates outward until it fills the whole space to an equilibrium density of around 20%. After this although there is continual local fluctuation, there is no global change evident. A further run of 0.5% population seen in appendix 1 reaches the same position but simply takes longer to do so.

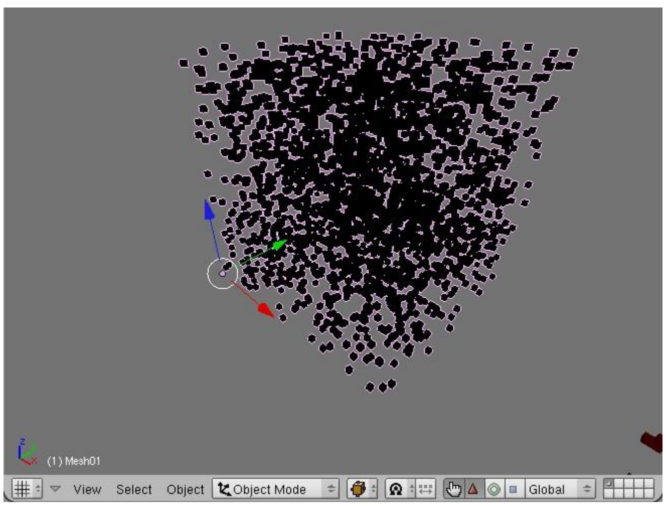


Figure 2: Random cells (50x50x50 grid)

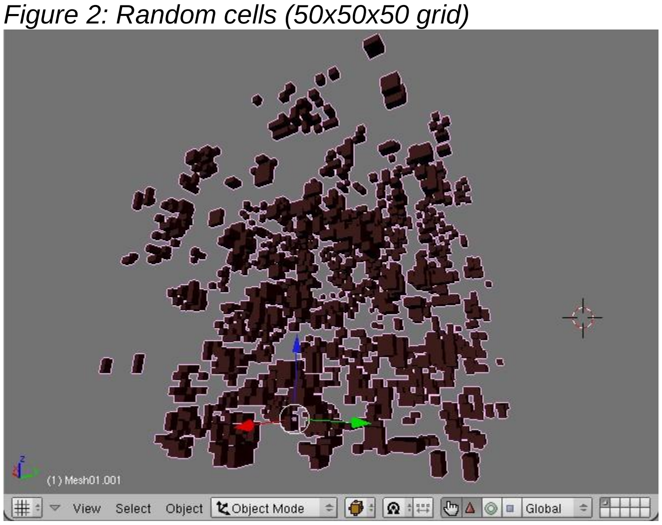


Figure 3: 1st Iteration (50x50x50 grid)

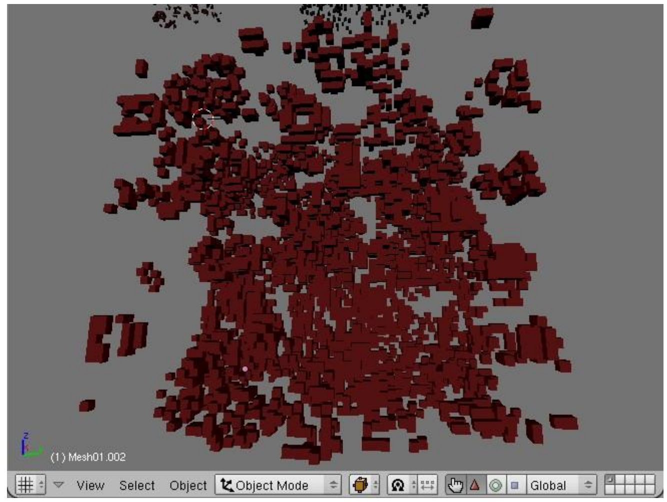


Figure 4: 2nd Iteration (50x50x50 grid)

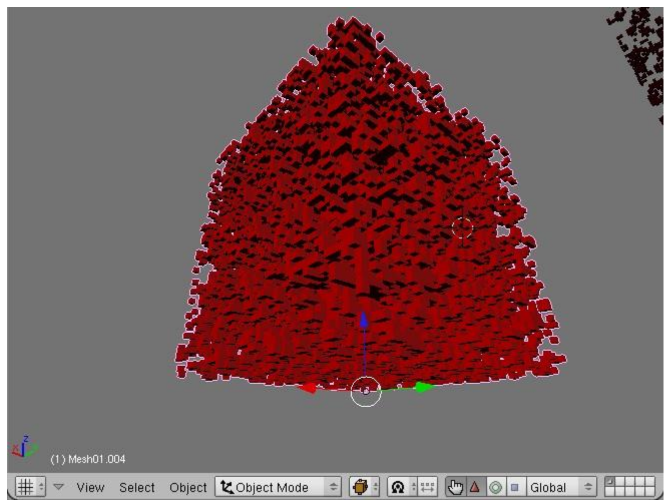


Figure 5: 20th Iteration (50x50x50 grid)

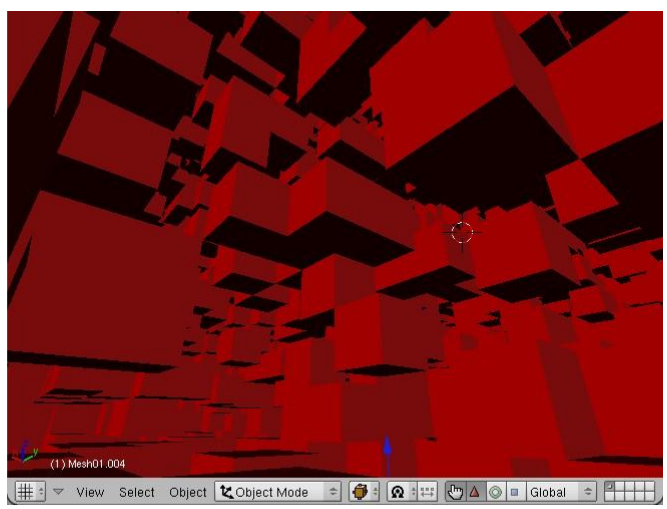


Figure 6: 20th Iteration - Internal (50x50x50 grid)

IV. CATALOGUING SIMPLE STRUCTURES

The evolution of basic structures within the system was examined. A simple column of cells of varying length from 1 to 10 cells was systematically examined. All sets of cells showed transience from 3 to 24 generations. Columns of 1 and 2 cells respectively last only one generation by definition. Figures 7 to 14 show the evolutionary stages of each primitive shape.

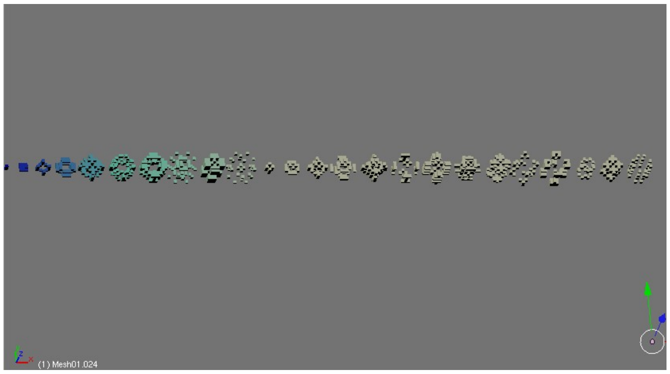


Figure 7:Three Column Evolution

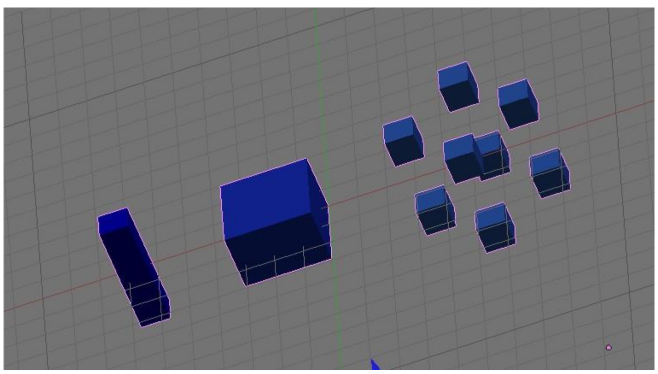


Figure 8: Four Column Evolution

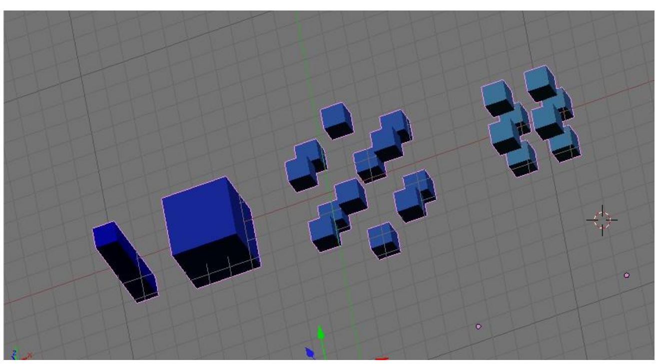


Figure 9: Five Column Evolution

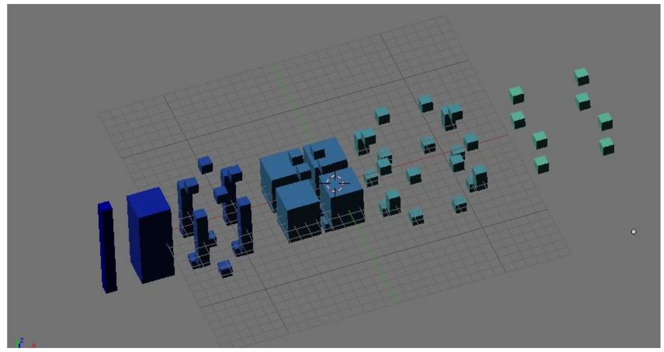


Figure 10: Six Column Evolution

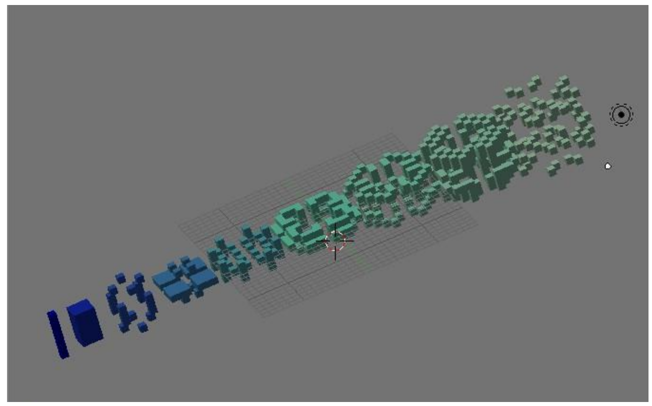


Figure 11: Seven Column Evolution

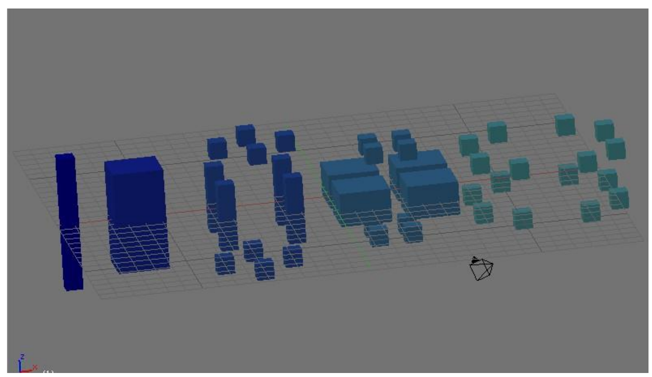


Figure 12: Eight column evolution

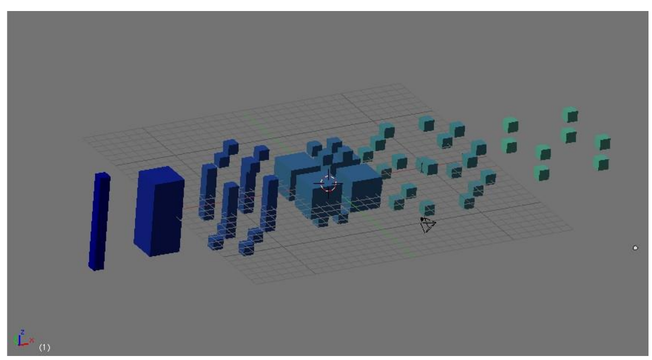


Figure 13: Nine Column Evolution

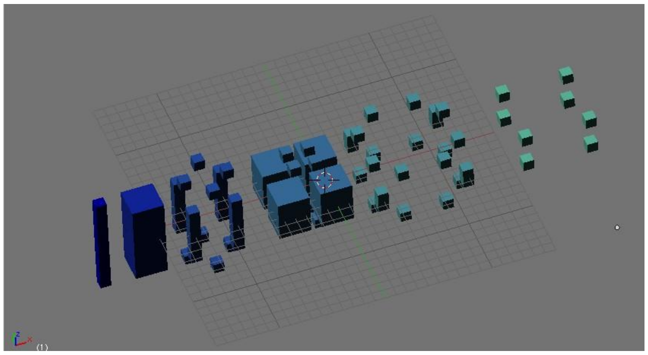


Figure 14: Ten Column Evolution

There are some interesting formations. A simple three column block has a lifespan of 24 iterations after which it self-destructs as seen in figure 7. Whereas a simple 4 column block has an even shorter lifespan of only 3 iterations. And a 5 column block has a lifespan of 4 iterations as seen in figure 9 and a six column also has a lifespan of 4 iterations. The full results are tabulated in Table 1

Column length	Lifetime (iterations)
1	1
2	1
3	24
4	3
5	4
6	4
7	9
8	5
9	6
10	6

Table 1: Lifetime of primitive column formations

Significantly a column of three cells has by far the longest transient life of 24 generations, which compares starkly with a column of 4 cells that only has a 3 generation life.

V. Alternative Rules

The rules which we have used so far are the same as 2D life and it is likely that these rules lead to an over population in 3D as many more neighbours are available, 26, in 3D compared to 8 in 2D. This means that nearly all configurations are 'Garden of Eden' types and lead to a maximal density as depicted in figure 6.

The original rules specify minimum of 2 cell and maximum of 3 to sustain life while a minimum of 3 and maximum of 3 are required to create life. Using these numbers in this order, this rule can be conveniently categorized as the (2333) rule. A particular rule was chosen out of the infinite number which are available. The (5766) rule was chosen as being closest in neighbourhood ratio for 3D to the (2333) rule in 2D. In other words (2333):8 corresponds to (5766):26. A simple 1% random distribution grid was used to enable a comparison with the results of the same distribution under the (2333) rule. The results of the run are seen in appendix 2. It was found that this rule did not generate exponentially and reach a limit. Instead this distribution came to a simple scattering of still life and oscillator patterns which more closely matched the results expected in the normal 2D Game of Life.

VI. Conclusions and Future Work

The techniques and technologies needed to create a 3D visualization using open source software has been established and explored. Research issues opened with this line of enquiry are extensive and would require detailed work before they could be fully explored. Initial findings suggest a useful architecture can be initially developed making use of 3D Cellular Automata and that this is a system worth further investigation. However the question of whether the non-dynamic use of blender is the best vehicle for this remains open. The biggest drawback of this software is that it cannot be run in real time. It also has serious limitations of grid-size and is not scalable.

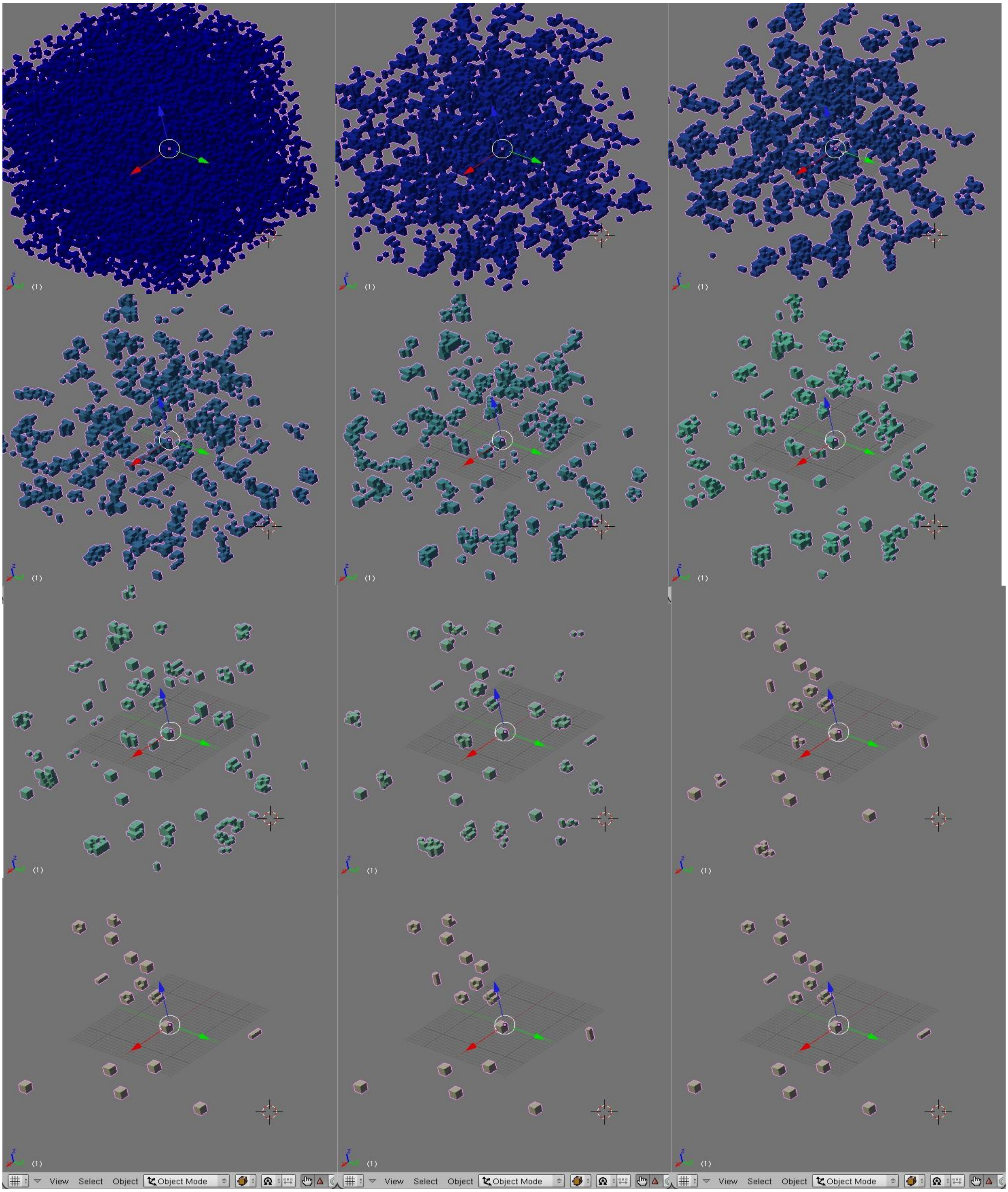
For the future, a number of promising research directions in addition to those in this work is under scrutiny. There are a large number of unanswered questions raised by this paper. How do the evolution of patters compare from one rule set to another? Are gliders present in the 3D (5766) domain as they are in the 2D (2333) domain? And consequently can this system also be shown to be equivalent to a Universal computation machine. At present these and other questions must remain open.

ACKNOWLEDGMENTS
The author would like to thank Nick Rowe for his expert assistance and advice with Blender and Python scripting and for his encouragement of this work.

References

1. von Neumann, John; Burks, Arthur W. (1966). "Theory of Self-Reproducing Automata." www.walenz.org. http://www.walenz.org/vonNeumann/index.html.
2. Codd, Edgar F. (1968). *"Cellular Automata"*. Academic Press, New York.
3. Banks, Edwin (1971). Information Processing and Transmission in Cellular Automata. PhD thesis, MIT, Department of Mechanical Engineering. http://www.bottomlayer.com/bottom/banks/banks_commentary.htm.
4. Gardner, M (1970) Scientific American 223 (October 1970): 120-123.

APPENDIX 2
The (5766) Rule on a 1% random distribution



Appendix 1

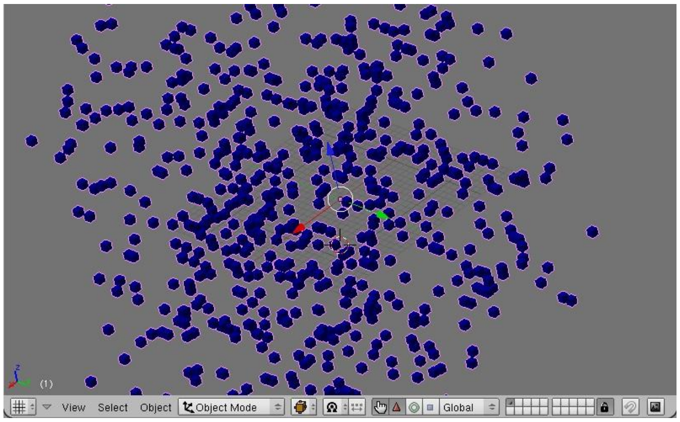


Figure 8: random 0.005 (50x50x50) 0 iteration

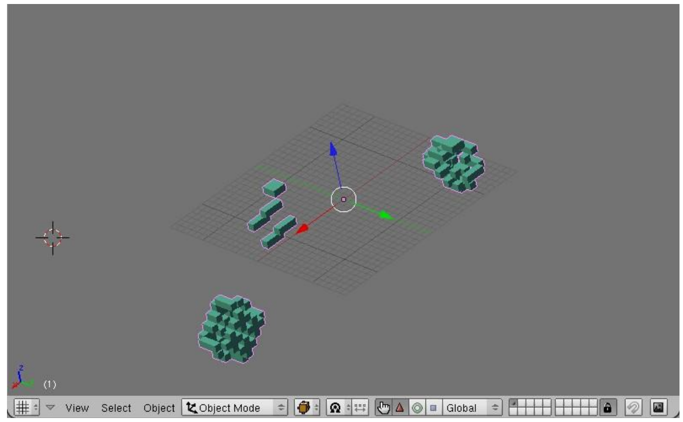


Figure 4: random 0.005 (50x50x50) 5 iteration

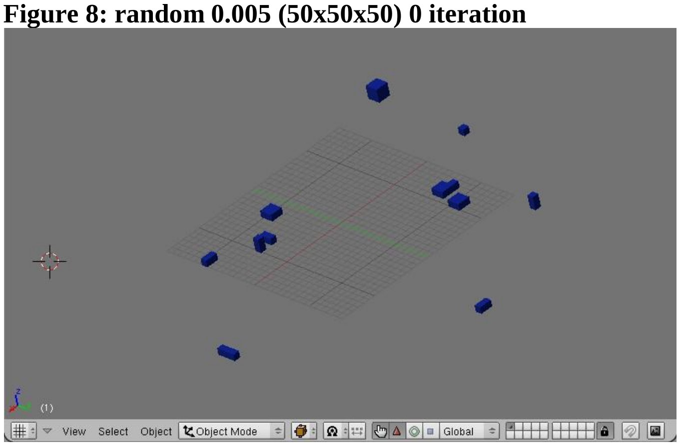


Figure 7:random 0.005 (50x50x50) 1 iteration

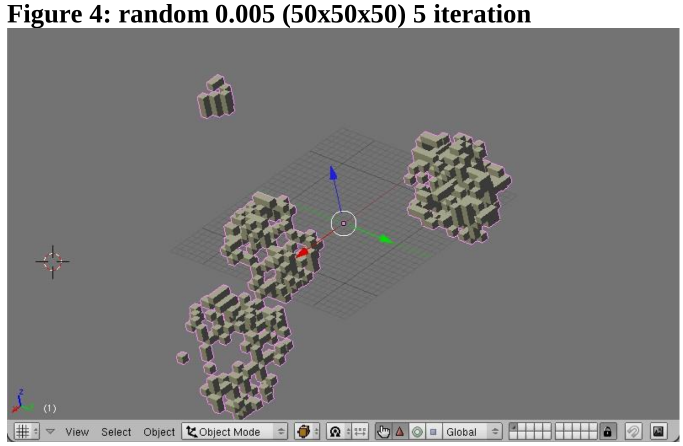


Figure 3:random 0.005 (50x50x50) 10 iteration

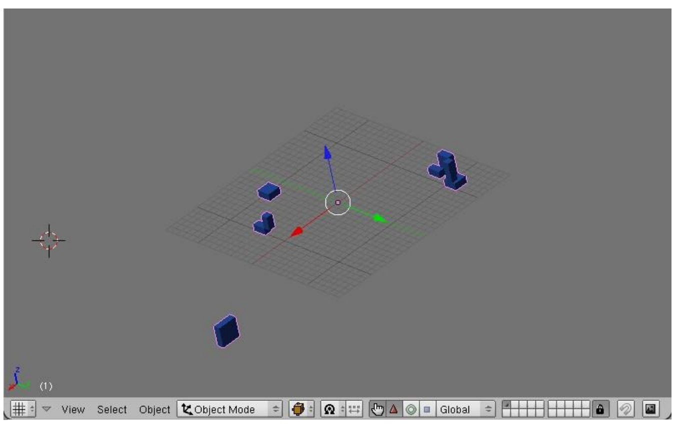


Figure 6: random 0.005 (50x50x50) 2 iteration

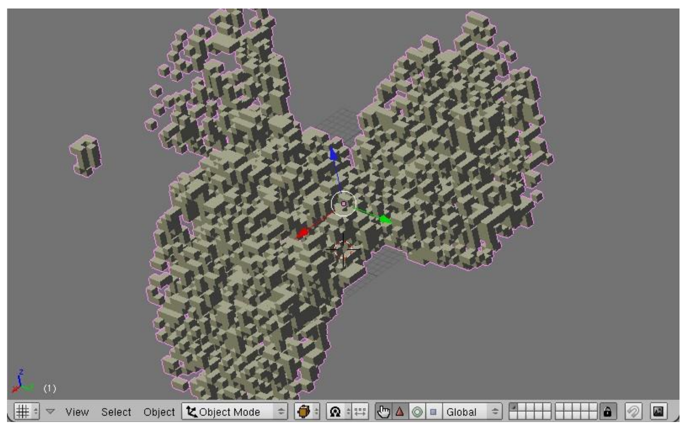


Figure 2:random 0.005 (50x50x50) 20 iteration

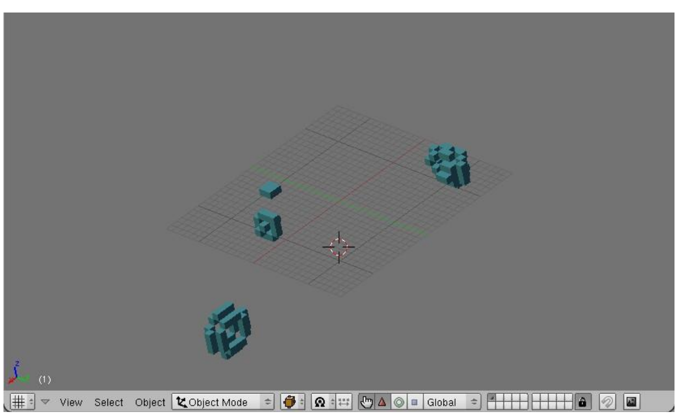


Figure 5:random 0.005 (50x50x50) 4 iteration

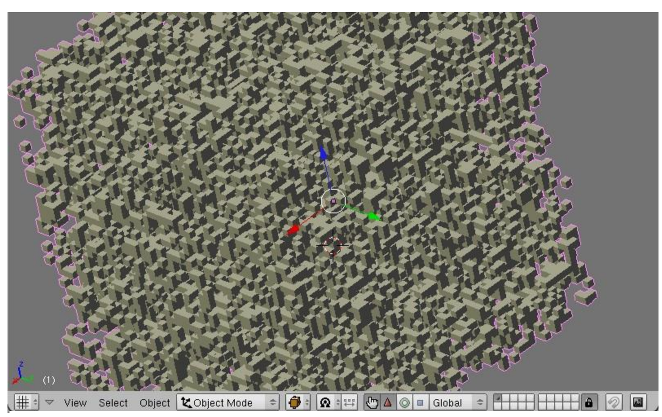


Figure 1:random 0.005 (50x50x50) 50 iteration