Simulation of Life Game Based on Cellular Automata

Jiacheng Huang, Yong Peng

College of Internet of Things Engineering, Jiangnan University, Wuxi, China
Email: 1628407015@stu.suda.edu.cn

How to cite this paper: Huang, J.C. and Peng, Y. (2021) Simulation of Life Game Based on Cellular Automata. *Journal of Computer and Communications*, **9**, 44-58.
https://doi.org/10.4236/jcc.2021.91005

Received: December 21, 2020
Accepted: January 19, 2021
Published: January 22, 2021

Abstract

Vs2010 is used as the development environment, so as to realize the visual programming of Game of Life, and explore the life evolution process of cell group in different sizes and states. According to cell forms such as circulation and disappearance, it reflects the complex changes of Game of Life. Setting different initial states through the code and observing the final generated graphics, you can see that the complex and simple initial states can achieve the same result. It is also concluded that a suitable initial state can reach the final state in fewer steps, which can greatly simplify the evolution process. The entire system is completely closed and has certain limitations. Meanwhile, the evolution process of symmetrical initial state is also symmetrically distributed. I introduce random quantities into the system to make the simulation results closer to the actual situation. By setting a random initial state to make the chaotic and disorderly situation simple, the concept of "determinism and randomness" can be better expressed. In the process of change, some local structures remain fixed, and some local structures present periodic cycles. These structures interact in complex ways to understand the concept of "whole and part". The game of life enlightens us: the simplest logical rules can produce complex and interesting activities, and a complex system may be iterated by simple rules.

Keywords

Cell, Game of Life, Complexity

1. Introduction to Cellular Automata

The spatial model composed of a series of cells with the same properties is called cellular automata [1]. Cellular Automata are discrete models which are now widely used in scientific simulations and researches. They consist of some cells, which will change over time according to specific rules [2]. It is a dynamic system that is discrete in time and space. Each cell scattered in the regular grid takes a finite discrete state, follows the same action rules, and updates synchronously according to certain local rules. A large number of cells form the evolution of a dynamic system through simple interaction. Different from general dynamic models, cellular automata are not determined by strictly defined physical equations or functions, but are constituted by a series of model construction rules. Any model that satisfies these rules can be regarded as a cellular automata model [3]. For example, cellular automata are proved to be an abstract computational mathematical model, and cellular automata can be used as models or theories for physical systems and equipment. This method can abstract all the details of the basic physical system, but is still faithful to the basic physical reality it describes [4].

In the 1950s, J. Von Neumann first proposed the concept of cellular automata, so as to study the phenomenon of self-replication of machines, which is a computer model describing the process of reproduction and evolution, but it is too complicated, with thousands of cells and 29 states [5] [6]. Therefore, it is difficult to be implemented. In the 1980s, there was a new breakthrough in the theoretical research of cellular automata. Researchers have successfully discovered the phenomena of bifurcation, attractor, self-replication and self-organization by using the existing simple rules. Since then, cellular automata have become an important tool for studying, solving and applying complex systems [7].

Hereafter, Stephen Wolfram has extended the theory of cellular automata and recorded all his ideas in the 1200-page "A New Kind of Science" [8]. In this book, he deepened the research on cellular automata and complex systems, believing that all natural laws can be created by cellular automata, from simple addition, subtraction, multiplication, and division operations to complex stock fluctuations can be explained by this theory, namely when a set of rules is set up, the universe likes a machine constantly calculating to produce our own world, thus creating our own world [9]. Cellular automata use simple methods to solve problems that cannot be solved by complex methods in traditional science, and use simple, local regular and discrete methods to describe complex, global and continuous systems. It is possible to conduct simulation studies on urban growth [10] [11] [12]. It can monitor and simulate land use changes [13] [14]. It can also be applied to artificial life research, which has been widely used [15].

The basic structure of two-dimensional cellular automata is cell, cell space, neighbor, cell evolution rule and cellular state, which is a simple calculation method [16] [17]. Each cell is in a certain position and state. Each cell surrounds the other cells, and there are only two states, either alive or dead [18]. Humans are used to assuming that the environment will greatly affect the cellular state. Assuming that the environment will affect the cellular state, we will narrow the scope of the assumption to infinitesimal and think that the state of the cell is only affected by its current state and the state of the adjacent cell, which is similar to a calculus equation dividing a rectangle into countless infinitesimal rectangles. In this sense, differential equations and cellular automata are a pair of relative calculation methods [19].

The initial state assumes that there is a natural cell group, and each cell has only one initial state, and then the cell group will change, reproduce or annihilate. When the time is n, the positions and states of all cells will be refreshed once. Then, according to the rules, the current state of each cell and the state of neighboring cells, the state of this cell is inferred. When the time is $n + 1$, all cells change according to the same rules. It seems very simple, but a simple calculation method is difficult to be analyzed. The characteristic of cellular automation is that under certain rules, it is difficult to predict what changes will take place in the cell under any initial conditions. However, it is also the charm of cellular automata, namely unpredictable, unmeasurable and unpredictable. A seed can grow into a big tree in various forms or even an oasis. There is a square-mesh cellular automaton, and the initial state contains a cell group. By setting a simple rule, only the surrounding cells are changed. After dozens of changes, the whole graph changes greatly, leaving only a small group of cells. This rule successfully depicts the internal features of the graph, as shown in Figure 1. Each cell uses only local information, but successfully changes the big environment, which means that local rules can generate global patterns.

A lot of scientific research has developed the theory of cellular automata, which enables us to better think and understand related concepts such as "determinism and randomness", "process and state", "simple and complex", "continuous and discrete", "parallel and serial" etc. These views provided us with a good idea and brought many inspirations.

The novelty of this article lies in the following aspects:

1) Successfully demonstrated part of the work of the predecessors with the software programmed by myself, demonstrated cellular automata that recognizes internal features, game of life, etc.

2) Discuss the evolution process of the game of life, understand the characteristics of the whole change through different initial states, and better understand the concepts of "process and state", "simple and complex". Introduce random quantities into the system to make the simulation results closer to the actual situation. By setting a random initial state to make the chaotic and disorderly situation simple, the concept of "determinism and randomness" can be better expressed.

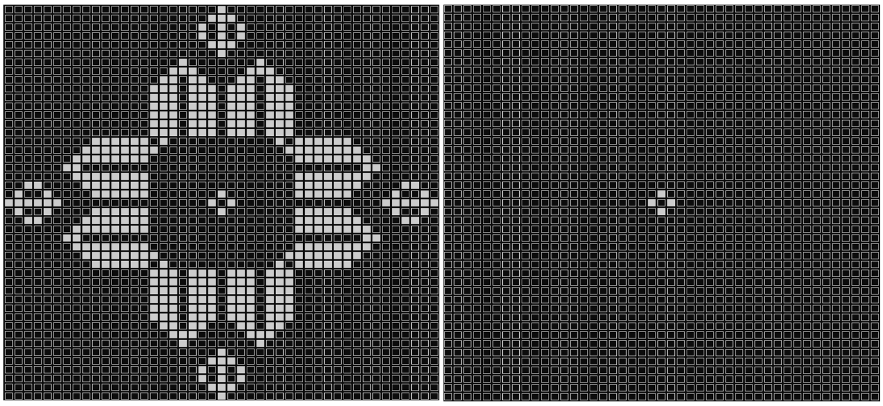


Figure 1. Cellular automata that recognizes internal features.

3) In the process of change, some local structures remain fixed, and some local structures present periodic cycles. These structures interact in complex ways to understand the concept of "whole and part".

2. Introduction to Game of Life

Conway's Game of Life is the most typical cellular automata model in the development of several decades [20]. Around 1970, Conway and his students launched Game of Life after a lot of experiments. Its appearance has aroused the extensive interest of people, and cellular automata has become popular all over the world [20] [21]. The rules of Game of Life are very simple, but they approximately describe the survival and reproduction rules of biological groups and are most widely used. Game of Life is also a kind of image matrix transformation, which brings decades of exploration and discussion to the world [22] [23].

The rule of cell state evolution in the game of life is the core part of the entire program [24]. Space is represented by a two-dimensional rectangular grid, each grid represents a cell, and each cell has two states: dead or alive. The Moore neighbor approach is adopted here, as shown in **Figure 2**.

The Moore neighbor approach means that each cell has eight adjacent cells, and the state of each generation of cells depends on the number of cells around it. By calculating the number of living cells in the eight cells adjacent to itself, the evolution is carried out according to the following three rules:

1) If the number of cells around the living cell is less than 2 or more than 3, death will occur;

2) If the number of cells around the living cell is 2 or 3, it can continue to survive;

3) If the number of cells around the dead cell is exactly 3, it will be resurrected.

Game of Life is composed of numerous local changes, and none of the cells knows the changes in the big environment. Simple graphics can be constantly changed and gradually complicated. However, complex graphics can also be continuously simplified under the rules. In fact, Game of Life leaves us more thinking-whether there is a model that can grow indefinitely? Is it possible for a small cell to create larger objects or even the whole world by a simple rule?

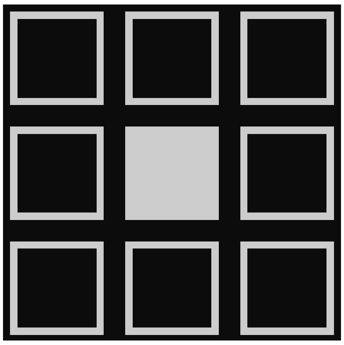


Figure 2. Moore-type neighbor method.

3. Model of the Game of Life

In this paper, Visual Studio 2010 is used to simulate the evolution process of cell group. The basic data structure is matrix, which is convenient to simulate a large number of cell states. The two-dimensional matrix A [i][j] is used to define the initial state of the cell, in which the life state of the living cell is 1, represented by white squares, and the life state of the dead cell is 0, represented by black squares. Considering the authenticity and efficiency of the simulation, Moore-type neighbor method is still adopted to study, and C language is used to express the evolution rules of cell.

First judge the survival status of cell neighbors, and count the number, the implementation code is as follows:

```
Count = board[ROWS - 1][COLS - 1] + board[ROWS - 1][COLS]
+ board[ROWS - 1][COLS + 1] + board[ROWS][COLS - 1]
+ board[ROWS][COLS + 1] + board[ROWS + 1][COLS - 1]
+ board[ROWS + 1][COLS] + board[ROWS + 1][COLS + 1];
```

Then change the state of the cell according to the rules, board[i][j] indicates the initial state and temp[i][j] indicates the state after the change:

```
switch (Count)
                {
                case 3:
                    temp[ROWS][COLS] = 1;
                    break;
                case 2:
                    temp[ROWS][COLS] = board[ROWS][COLS];
                    break;
default:
                     temp[ROWS][COLS] = 0;
                }
```

Finally, assign the result state to the current state and prepare to enter the next change:

```
board[i][j] = temp[i][j];
```

The simulation interface of the whole software is shown in **Figure 3**. There are two small menus. The "Options" menu is used to select the display and running status of the cell group. First of all, the display size is selected. There are three interfaces to select, and then the initial state is selected. There are also three setting methods. The first is code setting. C language is used to describe the location of the cell group, which is suitable for large quantities. The second is random setting and the third is manual setting. After selecting a good state, you can click "START", and the initial state will change continuously according to the rules. During the change process, you can click "STOP" to view the current life state, press "START" and start the change again, and "EXIT" is used to exit the

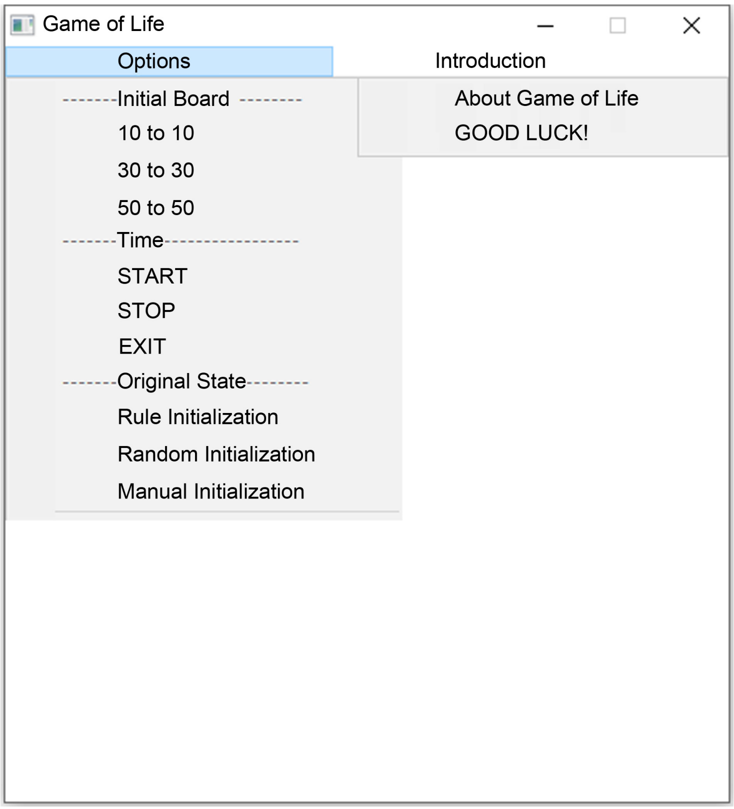


Figure 3. Interface of software.

program. The "Introduction" menu is made up of two parts. It is used to briefly introduce information related to the program, as shown in Figure 4.

The flow chart of the entire program is shown in Figure 5.

4. Graphical Simulation and Result Analysis

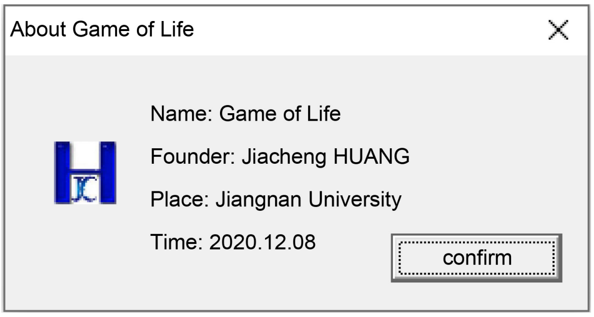
The simulations are performed on three different sizes of interfaces. The evolution of the game of life is more complicated, and the typical changes are selected for analysis.

4.1. Cells Change on the 10 × 10 Interface

For the 10 × 10 interface, I chose a cyclical mode to illustrate, as shown in Figure 6, there are a total of 5 states, from the outside to the inside, and then from the inside to the outside to achieve a periodic cycle.

There are many types of cyclical cells. Here I will list two basic cells with a period of 2, as shown in Figure 7 and Figure 8, respectively. They will constantly exchange forms after the system is stable.

When the number of cells is large, the entire system can be divided into countless parts, and in the final stable state, it is also composed of these basic cell combinations. So I expanded the initial number of cells, as shown in Figure 9. After dozens of changes, the final state is composed of basic fixed cells and periodic cells. The entire cell group has a period of 2 and is constantly changing, as


About Game of Life

Name: Game of Life

Founder: Jiacheng HUANG

Place: Jiangnan University

Time: 2020.12.08

Figure 4. Interface of introduction.

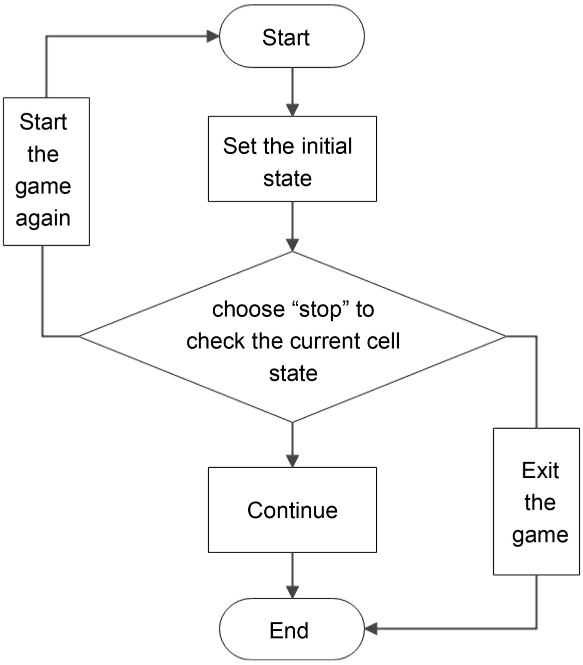


Figure 5. Flow chart of program.

shown in Figure 10.

4.2. Cells Change on the 30 × 30 Interface

For the 30 × 30 interface, I chose a vanishing mode to illustrate. As shown in Figure 11, they correspond to two different initial states. After dozens of changes, they gradually become more complicated. In the final state, all cells die. These are just two ways of cell death, and the changes in these two ways are completely different. I think that when the magnitude of the cell expands dozens of times or even infinity, there will also be certain situations that make the number of cells clear. Therefore, it can be concluded that an appropriate initial state can greatly simplify the evolution of the entire system, and can change from complexity to simplicity, or even nothingness.

4.3. Cells Change on the 50 × 50 Interface

For the 50 × 50 interface, I chose two different initial conditions as shown in

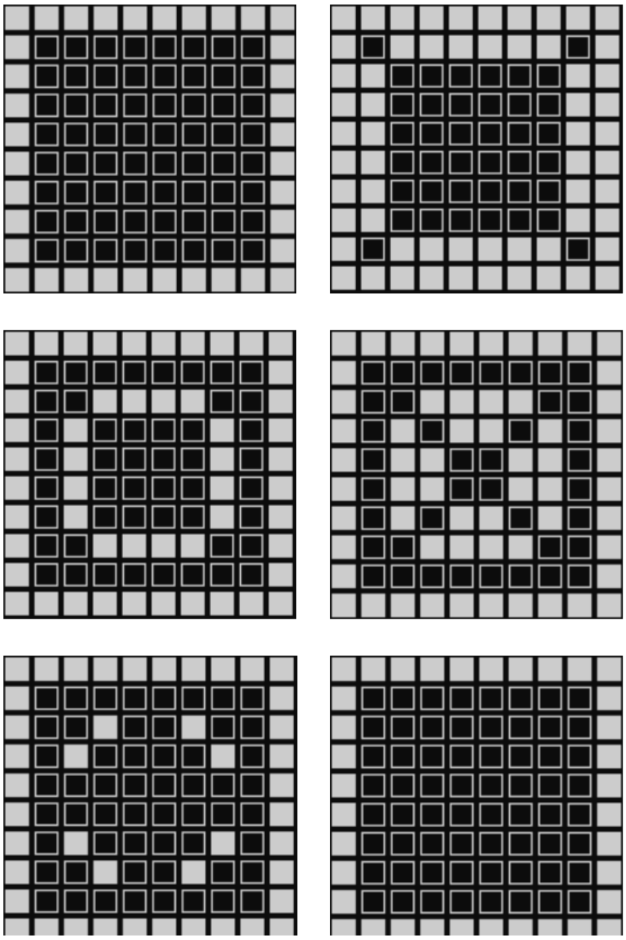


Figure 6. Cyclic cell.

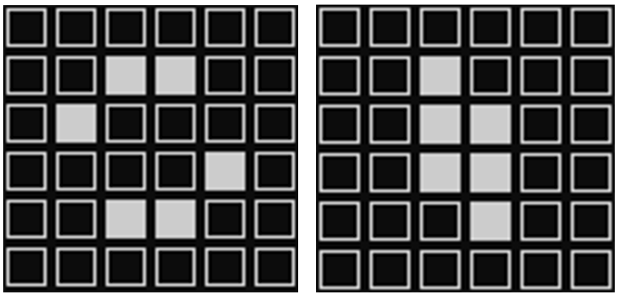


Figure 7. A cell state with period 2.

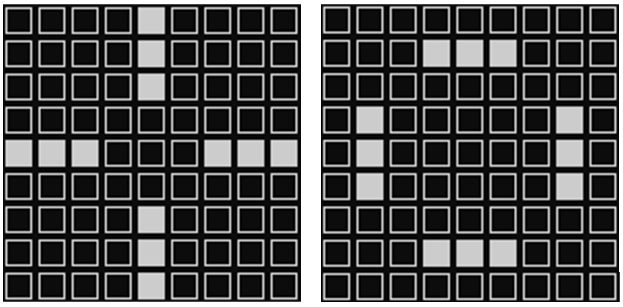


Figure 8. Another cell state with a period of 2.

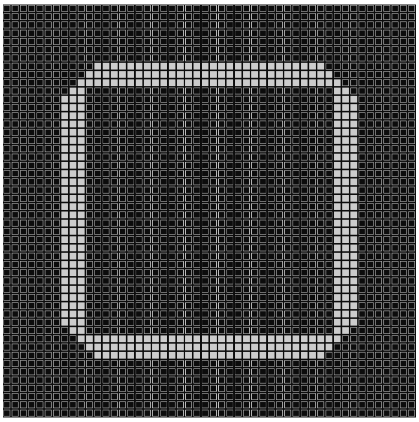


Figure 9. Initial state.

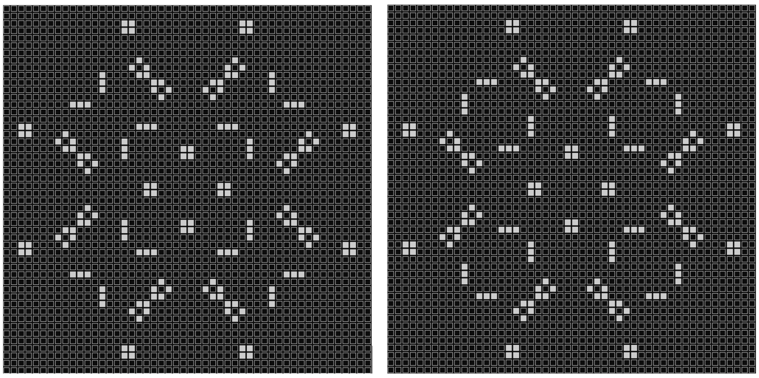


Figure 10. Two final states.

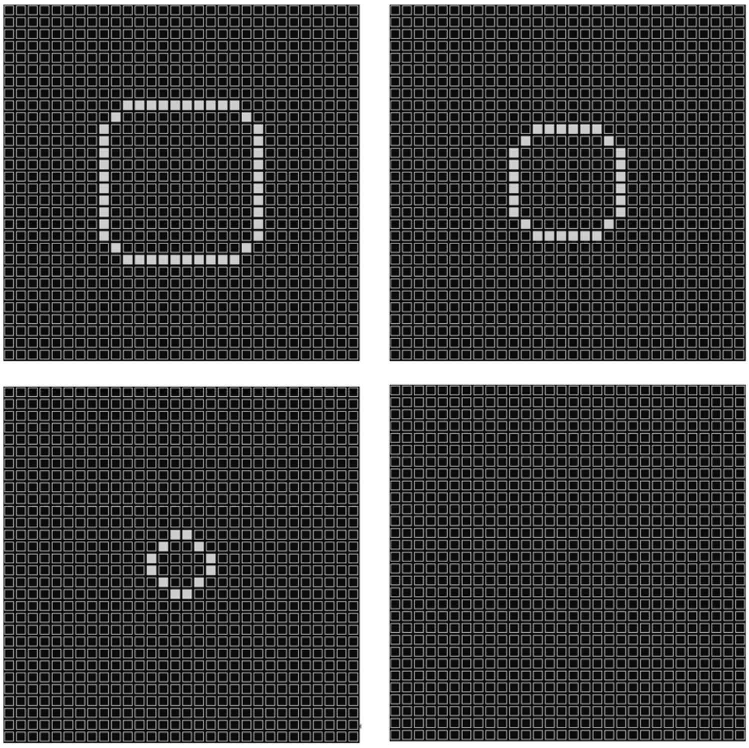


Figure 11. Vanishing type of cell.

Figure 12. After 69 and 67 transformations, the final state is exactly the same, as shown in Figure 13.

Observing on the 30 × 30 interface and changing the initial number of cell groups, you will find that similar situations exist. The two initial states are shown in Figure 14. After dozens of changes, the final state is the same, as shown in Figure 15.

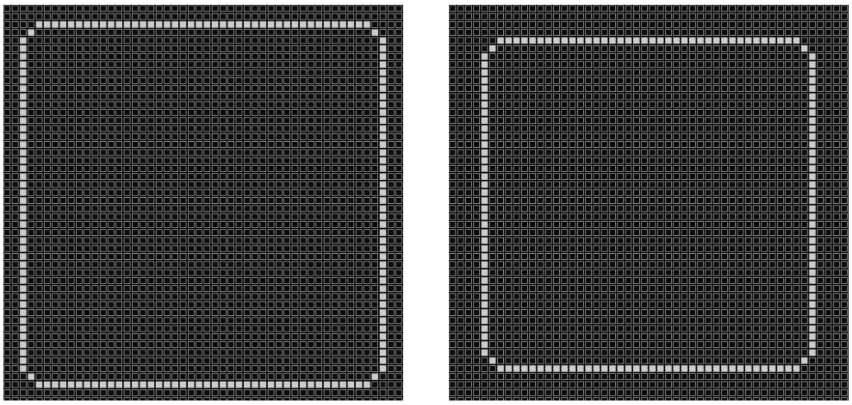


Figure 12. Two initial states with different numbers and positions.

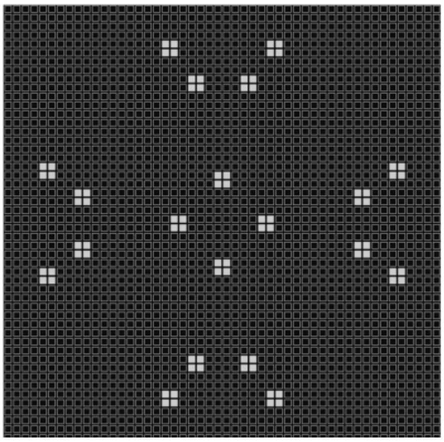


Figure 13. Final state.

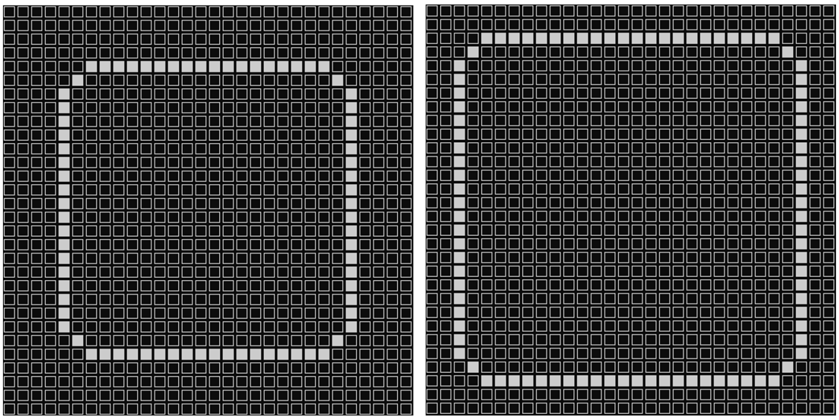


Figure 14. Two initial states on the 30 × 30 interface.

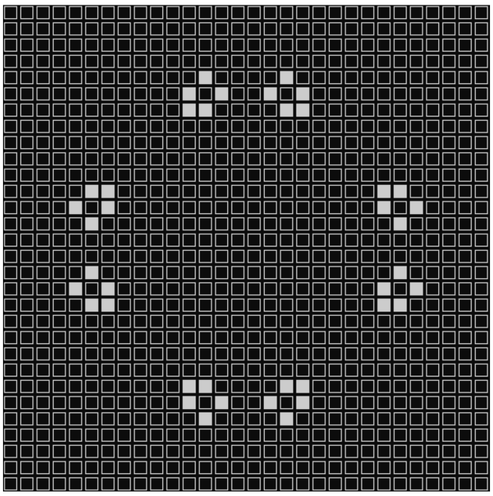


Figure 15. Final state.

Similar situations also exist on other size interfaces, so it can be concluded that in the cell system, there may be several or more ways to reach the same result. You can try to change the initial state of the cells group to achieve the ultimate goal. Although the difference between the two states of change is only a dozen cells in the current limited environment, when the environment is infinite, one more choice may greatly improve efficiency and make the evolution process easier.

In all the typical changes listed above, it can be clearly seen that when the initial state is symmetrically distributed, the entire process of evolution is symmetrical.

4.4. Cells Are Randomly Generated on the Interface

The random variable is introduced into the system, which can make the simulation results closer to the actual situation [25]. Therefore, we observe how the random initial state will change. The initial state is very complex, as shown in Figure 16. And the system tends to be stable after 988 changes. During the whole change process, the first 500 cell groups decreased continuously, and the system became complicated again at 650 times, leaving only fixed and periodic cell groups at the end, as shown in Figure 17.

Because cellular automata adopts a closed system and does not have openness, a single cell at the edge of the system will not die, which is a disadvantage. I think it is the limited environment that causes the temporary complexity of the system, similar to the sputtering of water droplets. When the cell group encounters the cell on the edge, it will emit more splash and become complicated. By comparing random cell groups on interfaces of different sizes, I hold that when the environment is infinite, the cell group will always change, but in the end it will tend to be simple. The simplified way of the whole process is to spread to all sides, which is somewhat similar to Chinese Go. The right move of a chess can revitalize the overall situation, and the complex situation will be suddenly clear

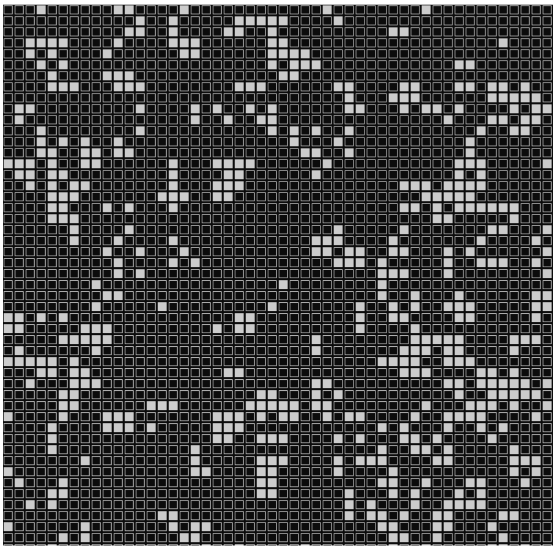


Figure 16. Initial state.

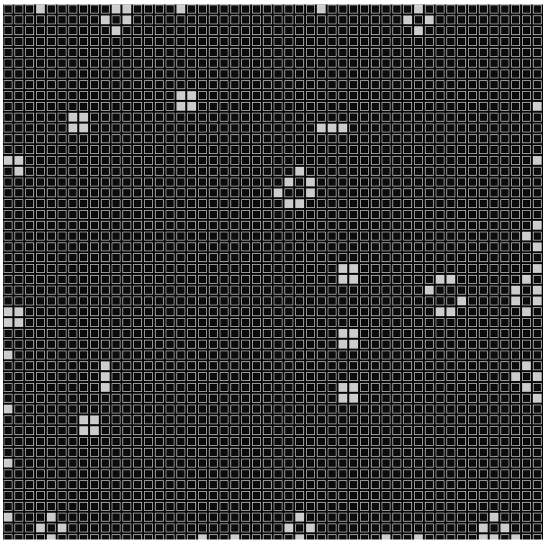


Figure 17. Final state.

in an instant.

This is very consistent with the principle of chaos theory. It tells people that "the original state of everything is a bunch of seemingly unrelated fragments, but after this chaotic state ends, these inorganic fragments will organically merge into a whole".

5. Conclusions

The entire simulation uses visual studio2010 as a development tool to realize the visual programming of the game of life, and specifically analyzes the life evolution process of the initial state, such as cycle, disappearance, and random. The simulation results show that the game of life is complicated. Composed of countless parts, it is born from simplicity, passing through the small cells of different parts, and then combined with its own large environment and passed to each level.

All simulations are carried out in a limited environment, which cannot truly reflect the changes of the system, and there is no openness. But there are some useful conclusions that can be drawn: a suitable initial state can greatly simplify the evolution process, and different initial states can achieve the same result, which can provide more selectivity. At the same time, when the initial states are symmetrically distributed, the evolution process is also symmetrical.

The use of cellular automata to simulate complex systems can explain many internal mechanisms of complexity. The simulation method of cellular automata has become a new method for studying complex systems [26]. When traditional scientific methods do not work, computer simulation is often the only way to understand the situation of things. It not only compensates for the weaknesses of human thinking, but also eases the limitations of human research tools. As Paul Humphreys [27] expresses in the title of the book, it is to help us "extend ourselves". Learning to use computer simulations can help us come up with new theories, models and hypotheses.

The game of life enlightens us: the simplest logical rules can produce complex and interesting activities, and a complex system may be iterated by simple rules.

Conflicts of Interest

The authors declare no conflicts of interest regarding the publication of this paper.

References

[1] Kemeny, J.G. (1967) Theory of Self-Reproducing Automata. John von Neumann. Edited by Arthur W. Burks. University of Illinois Press, Urbana, 1966. 408 pp., illus. $10. *Science*, **157**, 180. https://doi.org/10.1126/science.157.3785.180

[2] Gong, Y.M. (2017) A Survey on the Modeling and Applications of Cellular Automata Theory. *IOP Conference Series: Materials Science and Engineering*, **242**, No. 1. https://doi.org/10.1088/1757-899X/242/1/012106

[3] Giitsidis, T. and Sirakoulis, G.C. (2016) Modeling Passengers Boarding in Aircraft Using Cellular Automata. *Journal of Automatica Sinica*, No. 4, 365-384. https://doi.org/10.1109/JAS.2016.7510076

[4] Cheung, D. and Perez-Delgado, C.A. (2013) Cellular Automata as a Model of Physical Systems.

[5] Goldstine, H.H. (2008) The Computer from Pascal to von Neumann. Princeton University Press, Princeton. https://doi.org/10.1515/9781400820139

[6] Von Neumann, J. (1956) The General and Logical Theory of Automata. In: Newman, J.R., Ed., *The World of Mathematics*, Vol. 4, Simon and Schuster, New York, 2070-2098.

[7] Waters, D.P. (2012) Von Neumann's Theory of Self-Reproducing Automata: A Useful Framework for Biosemiotics? *Biosemiotics*, **5**, 5-15. https://doi.org/10.1007/s12304-011-9127-z

[8] Worfram, S. (2002) A New Kind of Science: Champaign Illinois. Wolfram Media,

Champaign.

[9] Wolfram, S. (1984) Computation Theory of Cellular Automata. *Communications in Mathematical Physics*, **96**, 15-57. https://doi.org/10.1007/BF01217347

[10] White, R. (1997) The Use of Constrained Cellular Automata for High-Resolution Modelling of Urban Land-Use Dynamics. *Environment and Planning B: Planning and Design*, **24**, 323-343. https://doi.org/10.1068/b240323

[11] Wu, F. and Webster, C.J. (1998) Simulation of Land Development through the Integration of Cellular Automata and Multicriteria Evaluation. *Environment and Planning B: Planning and Design*, **25**, 103-126. https://doi.org/10.1068/b250103

[12] Clarke, K.C. and Gaydos, L.J. (1998) Loose-Coupling a Cellular Automaton Model and GIS: Long-Term Urban Growth Prediction for San Francisco and Washington/Baltimore. *International Journal of Geographical Information Science*, **12**, 699-714. https://doi.org/10.1080/136588198241617

[13] Newkirk, R.T. and Wang, F. (1990) A Common Knowledge Database for Remote Sensing and Geographic Information in a Change-Detection Expert System. *Environment and Planning B: Planning and Design*, **17**, 395-404. https://doi.org/10.1068/b170395

[14] Li, X. and Yeh, A.G.O. (1998) Principal Component Analysis of Stacked Multi-Temporal Images for the Monitoring of Rapid Urban Expansion in the Pearl River Delta. *International Journal of Remote Sensing*, **19**, 1501-1518. https://doi.org/10.1080/014311698215315

[15] Lewin, R. (1993) Complexity: Life at the Edge of Chaos. *American Journal of Physics*, **61**, L627-L633. https://doi.org/10.1119/1.17163

[16] Nandi, S., Kar, B.K. and Chaudhuri, P.P. (1995) Theory and Applications of Cellular Automata. *IEEE Transactions on Computers*, **43**, 1346-1357. https://doi.org/10.1109/12.338094

[17] Ermentrout, G.B. and Edelstein-Keshet, L. (1993) Cellular Automata Approaches to Biological Modeling. *Journal of Theoretical Biology*, **160**, 97-133. https://doi.org/10.1006/jtbi.1993.1007

[18] Pimienta, P.J.P., Garboczi, E.J. and Craig, C.W. (1992) Cellular Automaton Algorithm for Surface Mass Transport Due to Curvature Gradients Simulations of Sintering. *Computational Materials Science*, **1**, 63-77. https://doi.org/10.1016/0927-0256(92)90008-W

[19] Wolfram, S. (1983) Statistical Mechanics of Cellular Automata. *Review of Modern Physics*, **55**, 601-644. https://doi.org/10.1103/RevModPhys.55.601

[20] Oxman, G., Weiss, S. and Be'ery, Y. (2014) Computational Methods for Conway's Game of Life Cellular Automaton. *Journal of Computational Science*, **5**, 24-31. https://doi.org/10.1016/j.jocs.2013.07.005

[21] Heudin, J.-C. (1996) A New Candidate Rule for the Game of Two-Dimensional Life. *Complex Systems*, **10**, 367-381.

[22] Liu, D.-D., Zhang, W., Yu, H. and Zhu, Z.-L. (2018) An Image Encryption Scheme Using Self-Adaptive Selective Permutation and Inter-Intra-Block Feedback Diffusion. *Signal Processing*, **151**, 130-143. https://doi.org/10.1016/j.sigpro.2018.05.008

[23] Chai, X.L., Zheng, X.Y., Gan, Z.H., Han, D.J. and Chen, Y.R. (2018) An Image Encryption Algorithm Based on Chaotic System and Compressive Sensing. *Signal Processing*, **148**, 124-144. https://doi.org/10.1016/j.sigpro.2018.02.007

[24] White, R. and Engelen, G. (1993) Cellular Automata and Fractal Urban Form: A Cellular Modelling Approach to the Evolution of Urban Land-Use Patterns. *Envi-*

ronment and Planning A, **25**, 1175-1199. https://doi.org/10.1068/a251175

[25] Li, X. and Gar-On Yeh, A. (2000) Modelling Sustainable Urban Development by the Integration of Constrained Cellular Automata and GIS. *International Journal of Geographical Information Science*, **14**, 131-152. https://doi.org/10.1080/136588100240886

[26] Qi, L.L. and Zhang, H.X. (2007) Reason Out Emergence from Cellular Automata Modeling. *Studies in Computational Intelligence*, **64**, 147-159. https://doi.org/10.1007/978-3-540-71986-1_8

[27] Humphreys, P. (2004) Extending Ourselves. Oxford University Press, Oxford. https://doi.org/10.1093/0195158709.001.0001