LABORATORY INSTRUCTION PAGES

Laboratory Experiments With
Artificial Life Studies
CELLULAR AUTOMATA
John Conway's Game of Life

Geology 200 - Evolutionary Systems
James Madison University
Lynn S. Fichter and Steven J. Baedke

Purpose

Experimenting with "Local Rules/Global Behavior"

We began the semester with Cellular Automata experiments, but did not tell you anything about them, just encouraged you to observe how the system behaved. By now in lecture we have explored the principles of cellular automata and the purpose of this lab is to explore the behaviors of the Game of Life with new appreciation.

Cellular Automata are the most basic Artificial Life systems, and the most simple. But, for all their simplicity they are amazingly complex. The core theoretical principle here is Local Rules/Global Behavior, the principle that the large scale patterns we see in a system are not the result of some overall design, but emerge from the working out of individual rules whose influence and effect is local and does not extend beyond its immediate surroundings.

The theoretical and philosophical significance of this idea is that complex outcomes emerge from the action of the simple rules, creating emergent properties that seemingly flow from purpose and design. But, they do not require purpose and design, and thus in the larger picture we begin to nibble away at the idea that the only way complex systems can be explained is through arguments from design - i.e. vitalism and finalism.

The Game of Life was invented by the British mathematician John Conway, and described by Martin Gardner in his Mathematical Games in Scientific American (223(4), October, 1970, pp 120-123). The game was originally played by hand with counters, but implementation on a computer greatly increased the ease of exploring patterns. Since its invention and popularization by Gardner the Game of Life has become a widespread research tool with applications in many disciplines, including, for example, game theory and sociology. A wide variety of cellular automata discussions and computer programs are available on the web; search on "John Conway's Game of Life" and you get a long list.

The Game of Life is, of course, not a game in the traditional sense where the goal is to follow top-down rules to achieve some predetermined goal (win the game, either against the game itself or an opponent.) Instead the Game of Life is about the emergence of unexpected patterns and behaviors though bottom up processes.

We are interested in exploring four ideas here:

1. Self organization from large random arrays.
2. Classifying the behavior cellular automata.
3. Messing with the local rules; the behavior a variety of Survival/Birth rule sets.
4. Avalanche Behavior: Analyzing information flow in cellular automata.

We expect you will be focused on doing the experiments. However, as you proceed it is important to watch how cellular automata systems are behaving in terms of the principles of chaos and complexity.

The Instructions for the opening the program and for the experiments are in these pages. Data sheets for recording these results of your experiments are in a separate packet titled "Record of Experimental Results"

1. Self Organization in Large Random Arrays

Using David Bunnell's Life3000 Program[1]

❑ **Life3000** is available only in the Geology Department computer lab, Miller 232. Click on the icon in the Evolutionary Systems desktop folder.

Life3000.lnk

❑ **Life3000** is a Windows based program. Observe the tool bar at the top. Move the cursor across FILE, **EDIT, WORLD, OPTIONS** to see what is there.

❑ Under **WORLD** click the **GLOBE** and peruse the options available. We use them later.

❑ Place the cursor on the large blank screen on the right. Click the left button and a pixel will light up; it is said to be "alive." Click the same pixel with the right button and it will become "dead."

❑ Hold the left button down and swirl the pointer across the screen. Pixels will light up as you go.

- ▸ Clicking **GENOCIDE** to the left clears the screen of everything.
- ▸ **EDIT: UNDO** will get your last screen back.

1. Create a Large Random Array of Life Cells.

Hold the left mouse button down and swirl the pointer across the grid. The array should not posses any obvious patterns, but should have at least 200 live cells.

☞ Once you have created an array you like . . ..

- ▸ Go to **FILE: SAVE AS** and save the array in your folder on the S drive.

☞ Click **PROCREATE** and observe the outcome.

1. Experimental Record One – A Large Random Array

Record your results on the Record of Experimental Results pages. Describe what happened, the final outcome (sketch samples) and how many generations it took. Note that Pattern Test under OPTION will, if turned on, tell you when life is Stable, Extinct, or Repeating, and at how many generations.

2. Create and Run A Second Large Random Array of Livee Cells.

[1] Life3000, along with other A-Life programs came with the book Artificial Life Playhouse by Stehen Prata: Waite Group Press. The book is now out of print, although similar programs are available on the www.

Create and Run a second random array and save it following the procedure above.

2. Experimental Record Two – A Second Large Random Array

Record your results on the "Record of Experimental Results" pages.

3. Create and Run A Third Large Random Array of Live Cells.

Create and Run a third random array and save it following the procedure above.

3. Experimental Record Three – A Third Large Random Array

Record your results on the "Record of Experimental Results" pages.

4. Create and Run A Fourth Large Random Array of Live Cells.

Create and Run a fourth random array and save it following the procedure above.

4. Experimental Record Four – A Fourth Large Random Array

Record your results on the "Record of Experimental Results" pages.

5. Self Organization - A Thought Experiment.

5. Experimental Record Five – Describe what is meant by Self Organization in Cellular Automata.

Record your results on the "Record of Experimental Results" pages.

6. Perturbing the System with World Wrap.

Ok, we observe that cellular automata are capable of self-organizing, and they do so just by following the local rules. No one has programmed in these outcomes. But how robust is this?

☞ Open one of your original random files, any one that pleases you for the following experiments.

☞ Go to **World** and click the **Globe**.

- ▸ Go to **World Edge Wrap** and turn it on. The default condition of the world (grid) is that the edges are hard, and any live cell encountering, and attempting to go beyond it, becomes dead. Turning on World Edge Wrap connects the left edge with the right, and the top edge with the bottom. Any live cell crossing the edge shows up on the opposite side. In effect the world becomes spherical with no edges to fall off of.

6. EXPERIMENTAL RECORD SIX – PERTURBING THE SYSTEM WITH WORLD WRAP.

Describe how this run compares with the original; the final outcome, and the number of generations it took.

7. Perturbing the System with Mutations

Return to the original file used in the last run.

☞ Go to **WORLD: GLOBE** and turn **WORLD EDGE WRAP** on.

☞ Go to **OPTIONS: PATTERN TESTS** and turn them all off.

☞ Go back to **WORLD: GLOBE** and turn **MUTATIONS** on to 1 mutation (per generation).

Run the simulation again.

7. EXPERIMENTAL RECORD SEVEN – PERTURBING THE SYSTEM WITH MUTATIONS.

Describe how this run compares with the original; the final outcome, and the number of generations it took. Since mutations are random try going back to the original random file and running the system several times for comparison.

8. Changing Survival Neighbors.

Return to the original file used in the last run.

☞ Turn **WORLD EDGE WRAP** on, **PATTERN TESTS** on, and **MUTATIONS OFF.**

☞ Go to **WORLD: GLOBE: BIRTH/SURVIVAL NEIGHBORS**.

- ▸ Record the default settings below.
- ▸ Change Local Rules: Survival Neighbors to 3-3 and record below right.

	Original (default)Local Rules			Changed Local Rules to the Following	
	Birth Neighbors	Survival Neighbors		Birth Neighbors	Survival Neighbors
Minimum	□	□	Minimum	□	□
Maximum	□	□	Maximum	□	□

8. EXPERIMENTAL RECORD EIGHT – EFFECT OF CHANGING SURVIVAL NEIGHBORS.

Describe how this run compares with the previous two.

9. Changing Birth Neighbors.

Return to the original file and World conditions used in the last run.

- ▸ Record the default settings below.
- ▸ Change Local Rules: Birth Neighbors to 2-3 and record below right.

Original (default) Local Rules			Changed Local Rules to the Following		
	Birth Neighbors	Survival Neighbors		Birth Neighbors	Survival Neighbors
Minimum	□	□	Minimum	□	□
Maximum	□	□	Maximum	□	□

9. Experimental Record Nine – Effect of Changing Birth Neighbors.

Describe how this run compares with the previous two.

10. In Conclusion.

10. Record Ten – In Conclusion

Based on your observations, describe, rationalize, account for, justify, or explain behavior that originates from Local Rules/Global Behavior.

How do each of the three parameters affect the behavior of the system?

World Wrap?

Mutations?

Birth/Survival Neighbors?

Why and/or how does self organization occur in a Cellular Automata?

2. Classifying the Behavior of Cellular Automata

Introduction

Cellular Automata are open, dissipative structures, and what they dissipate is information. You are already aware from previous experiments that changing information flow (Birth, Death, Mutations, World Wrap) changes how the cellular automata behave. They may become point attractors, limit cycle attractors, strange attractors, or chaotic (vernacular) attractors.

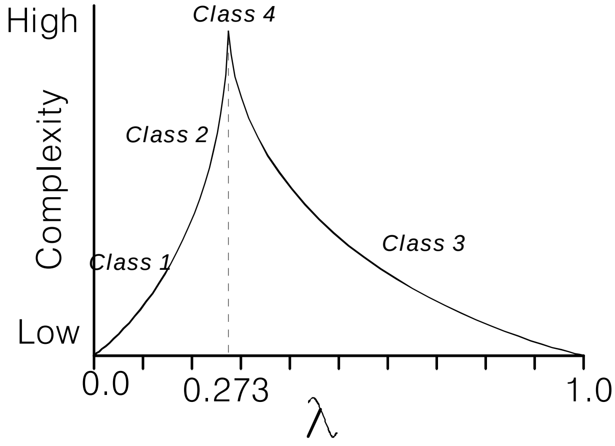
But for us, cellular automata are a model for all open systems. If we can understand information flow here in the relatively simple world of cellular automata, then we can better understand information flow in much more complex systems, including economies, ourselves, and our brains.

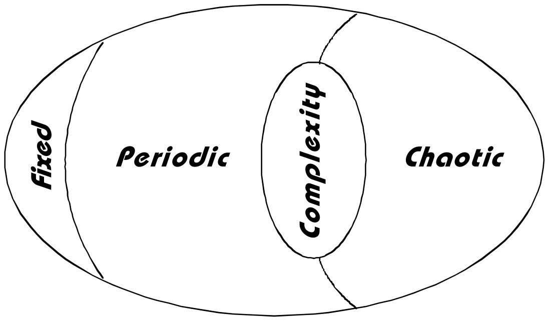
So, this is another example of universality - the principle that the underlying properties and behaviors of a large class of problems are essentially the same, and therefore, can be studied in principle without delving into the particular details of each system. The resulting analyses, however, can be applied to the understanding of any system which posses those properties. In Per Bak's words (the developer of self-organized criticality):

> *"We hope that important features of large-scale phenomena are shared between seemingly disparate kinds of systems, such as a network of interacting economic agents, or the interactions between various parts of the crust of the earth. This hope is nourished by the observation of the ubiquitous empirical patterns of nature - fractals, 1/f noise, and scaling of large events among them. Since these phenomena appear everywhere, they cannot depend on any specific detail whatsoever. . . this gives the option of selecting the simplest possible system belonging to that class for detailed study. One hopes that a system is so simple that it can be studied effectively on a computer, or maybe laws of nature can be derived by mathematical analysis, with pen and paper, from that stripped-down description or model."*

Purpose

Two methods exist to classify the ease of information flow in cellular automata, Chris Langton's lambda (λ) values (below left) and Steve Wolfram's classes one through four (below right).





The description of each of Wolfram's classes are:

- **Class One -** Rules that produce dull universes; all dead dells or all living cells. A point attractor.
- **Class Two -** Rules that produce stable, repetative configurations. A limit cycle attractor.
- **Class Three** - Rules that produce chaotic patterns.
- **Class Four** - Rules that produce complex, locally organized patterns. A strange attractor.

In this experiment you are to classify the behavior of some cellular automata in a similar way we did in class. It is good practice, and we want you to get to the point where you can classify these systems quickly and easily.

Remember intermediate classifications, and evolutionary classifications are possible. For example: 3 evolving to 2 [3 ⇨ 2], or a class between 1 and 2 *[1(2)]*.

Both the Instructions for the experiments and the spaces for recording your experimental results are contained here.

Working With The Life32 CA Program

❑ Life32 is another Cellular Automata program available at the address below. It is the most versatile, powerful, and sophisticated of the Game of Life programs we have found. This also makes it more complicated to use.

Shortcut to Life32.exe.lnk

http://psoup.math.wisc.edu/Life32.html

❑ *Open Program:* Look for the Life32 icon on the desktop under Evolutionary Systems. If it is not there then through Explorer Go To:

T:\Alifeprog\Life32\Program\Life32.exe double click the .exe file

❑ *Run Program:* **Life32** is a Windows based program with a tool bar across the top. Click on each of the categories: **GAME, EDIT, ANIMATION, SCREEN, OPTIONS, VIEW** to see what is there.

☞ Observe the keystroke shortcuts; some are very useful.

- Enter - toggles program on and off
- / - zooms out
- *Ctrl** - zooms in
- *Ctrl F* - opens the next file in the directory of files. With the program toggled ON each new file opens running.

❑ *Find Files:* Files are also on the T drive under the four folders **dblifelib-2**, **Guns**, **Guns2**, **LifeP**. There are over 2000 program files here you can explore later if you want. For now the *LifeP* folder contains a nice sampling of a wide variety of types.

☞ Go To: **GAME: OPEN** and it should open on the **LIFEP** folder. If not Go To: T:\LifeP and load the first file Acorn. Toggle the *Enter* key to start the program if it is not running.

☞ Now strike *Ctrl F* and the next file will open, etc. The files you examine are in alphabetical order below. You can *Ctrl F* to them, or go through **GAME: OPEN**.

11. Classifying CA behavior Using Steve Wolfram's Classes.

Below is a table for recording you analyses of the CA files. Run each of the files as often or as long as you need to accurately classify the system.

- Rule refers to Survival/Birth neighbors, shown bottom center of screen for each system.

11. Experimental Record Eleven – Steve Wolfram's Classes

File Name	Rule S/B	Wolfram Class.	Description of Initial State And Behavior
acorn			
AK47			
breeder			
bship			
btrack			
crystal			
eaters			
forerake			
fuses			
gun30			
hwssgun			
linepuf			

lonedots			
makefuse			
max			
mirage			
oscsp5			
rabbits			
slr			
spark2pi			
still			
tubtrax			

12. Sensitive Dependence

☞ CA's at S23/B3 are sensitive dependent; one of the easiest ways to detect this is to take a working pattern and try to change a pixel (live to dead, or vice versa) without destroying the pattern. Guns are especially good for this since they behave like limit cycle attractors; any disruptions shows up quickly.

12. Experimental Record Twelve – Sensitive Dependence

Go to **Games: Open: Guns** and Ctrl F through the files to find a relatively simple example. Stop the simulation and add or subtract single pixels somewhere in the pattern (**Options: Draw Mode**).

How close does a new pixel have to be to already existing pixels before the system is disrupted?

Once disruption occurs is the system able to recover?

3. Messing With The Local Rules and Global Behavior

Notice in the last experiment that all the systems have the same rules S23/B3. There is something magical about this rule set. It tends to produce rich but constrained behavior. If a natural system could evolve by naturally selecting a rule set from all the sets available this is probably the set that would survive. Late we explore different kinds of selection principles. But, may rules are possible, some produce gibberish and some behavior as rich as the default rules S23/B3. We want to explore this a little.

Note that playing around with rules to explore behavior variations is one of the biggest time sinks you can imagine. There are lots of wonderful things to discover, but it can take lots of time. We want to limit it here.

Try This: **GAME: OPEN: ROTO8.**

- Observe this is a limit cycle attractor.
- Now Go To **RULES: S1357/B1357.** You will have to quickly tap the **/** key to back away since under this rule set Roto8 expands fast.
- If you Go To: ***Game: Open: Open: Roto8*** (file will still be selected), then Go To **GAME: RULES** you can select a new rule set. This is a quick way to try a new rule set.
- Try these rule sets; compare and observe what happens (nothing to record, just observe).

– **S1357/B1357** (first one you did)
– **S235678/B3678**
– **/B234**
– **S12345/B3**
– **S34/B34**
– **S5678 /B35678**

13. Experimenting With Rule Sets

☞ Find one of the files you experimented with above, one that you think would be fun to change the rules on. Of course, you could start with something simple to see what happens, or something complex, or something boring to see if it can be made more interesting, or something intricate to see if it can be made more or less intricate. . . gee, this is a complicated decision.

☞ As you did with Roto8 above cycle through the various rules and observe how the behavior changes.

13. EXPERIMENTAL RECORD THIRTEEN – EXPERIMENTING WITH RULE SETS

Of course, we don't want you to do this blindly, but to contemplate the meaning of it all. So, in the spaces below record your observations:

☞ Record: How ***Interesting*** the pattern is: high, intermediate, low (subtler gradations if you can)(a value judgement, of course).

☞ Record: ***Comparative "r"*** value of each rule set, say on a 1-4 scale, like in X_{next} (also a value judgement).

☞ Record: the ***Attractor*** the system it is in.

Name of file you are working with.

How Interesting?	Rule Set	Relative r value:	Attractor	Brief description of behavior.
	/2			
	/234			Brief description of behavior.
	12345/3			Brief description of behavior.
	125/36			Brief description of behavior.
	1357/1357			Brief description of behavior.
	1358/357			Brief description of behavior.
	23/36			Brief description of behavior.
	235678/3678			Brief description of behavior.
	235678/378			Brief description of behavior.

	238/357			Brief description of behavior.
	245/368			Brief description of behavior.
	34/34			Brief description of behavior.
	34678/3678			Brief description of behavior.
	45678/3			Brief description of behavior.
	5/345			Brief description of behavior.
	5678/35678			Brief description of behavior.

14. Screen Captures

Micrografx Designer.lnk

☞ Find the 3, or 4, or 5 rule sets from the above experiment that show the most interesting behavior, and that also exhibits the most diverse behavior of the system under different rules sets. Do screen captures for each one. Paste into Designer, or some other program, and print them as a record of this experiment.

14. Experimental Record Fourteen – Print Screen Captures

4. Self Organized Criticality In Cellular Automata

(Cellular Automata as Sandpiles)

If we were to do this experiment fully it would involve a lot of very tedious record keeping. And none of the programs we have are set up to easily provide the information we need. Thankfully, someone has already systematically explored this out to millions of computer runs. So, we just want to gain a sense of how a CA can behave like a sandpile.

Preparing Life3000

Life3000.lnk

☞ Open the program **Life3000** (not the Life32 used in the last experiments).

☞ **Edge Wrap:** Go to World and turn **World Edge Wrap** on.

- ▸ The CA world now behaves as a globe with the top edge connected to the bottom, and left to right.

☞ **Pattern Tests:** Go to **Options: Pattern Tests: Repetitions** and set them for 5.

☞ Create a random array of at least 200 cells and save it to the S drive (or you may use a previous array).

15. A Cellular Automata as a SOC (Self-Organized Criticality) World.

▸ Data table is on the next page. Note that Generation 1 is in the lower right and you proceed up the page. Recording data this way will make it easier to do the arithmetic.

▸ **General Procedures:**

1. Click **Procreate** and watch until the program stops and one of two boxes appears: "Possible Repeating Pattern" or "Life Stable in Generation ___." Remember the number and close the box.

2. Record the Generation number in the table.

3. Add 1 pixel to the grid, and click **Procreate**. In an actual experiment the pixels would be place at random - try to mimic that.

4. When the program stops again, record the Generation number and calculate the difference between 1st and 2nd generations and record in the space.

5. **Repeat** Steps **1-4** for each new generation.

15. Experimental Record Fifteen – A CA as a SOC World

☞ Data table on the next page

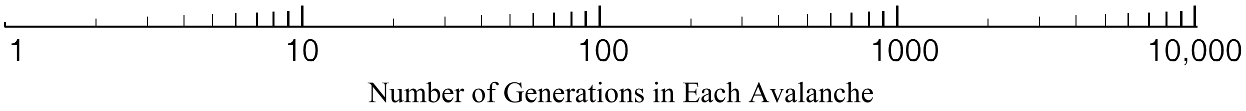
NOTE: Begin recording data in the lower right and work upward.

	Generation Number	Generation Difference		Generation Number	Generation Difference
24th Generation			12th Generation		
Difference 23rd and 24th=			Difference 11th and 12th=		
23rd Generation			11th Generation		
Difference 22nd and 23rd=			Difference 10th and 11th=		
22nd Generation			10th Generation		
Difference 21st and 22nd=			Difference 9th and 10th=		
21st Generation			9th Generation		
Difference 20th and 21st=			Difference 8th and 9th=		
20th Generation			8th Generation		
Difference 19th and 20th=			Difference 7th and 8th=		
19th Generation			7th Generation		
Difference 18th and 19th=			Difference 6th and 7th=		
18th Generation			6th Generation		
Difference 17th and 18th=			Difference 5th and 6th=		
17th Generation			5th Generation		
Difference 16th and 17th=			Difference 4th and 5th=		
16th Generation			4th Generation		
Difference 15th and 16th=			Difference 3rd and 4th=		
15th Generation			3rd Generation		
Difference 14th and 15th=			Difference 2nd and 3rd =		
14th Generation			2nd Generation		
Difference 13th and 14th=			Difference 1st and 2nd=		
13th Generation			1st Generation		
Difference 12th and 13th=			Begin Down Here		

16. Plotting the CA Data Logarithmically

- Below is a rule with a logarithmic scale. Take the 23 "Generation Difference" values you have and plot them along the rule with an "X". If you have a value more than once, stack the "X's" vertically.

16. Experimental Record Sixteen – Plotting the CA Data Logarithmically



Now, you do not have nearly enough data to draw any logically or mathematically sound conclusions. Nor did we really plot the single pixels at each run randomly, which biases our data. But other people have run this experiment randomly, for millions of data points, and we can rely on them for the rigor.

Nonetheless, our data should roughly follow a power law distribution. That is, many data points with low values, and only a few data points with high values. A similar power law distribution exists for total number of births/deaths with each random pixel setting.

To what degree does your data seem to follow a power-law distribution?

17. Combining Avalanche Data

We are going to combine avalanche data from everyone's experiment to increase our data base. It is not really a valid thing to do since everyone began with a different random array, but since it is avalanche frequence we are after the data should be roughly comparable.

17. Experimental Record Seventeen – Combining Avalanche Data

I will provide a master log chart for the plotting. Everyone plot their avalanche data.

Observe the Following:

☞ How sensitive the CA can be. Sometimes adding a single pixel causes avalanches of many generations of change, sometimes, only a few generations of change. This is analogous to dropping single grains of sand onto the sand pile. Some land in the right place and cause major avalanches; others just accumulate, building up the criticality.

What was the largest avalanche you had?

What do you think might be the upper size limit of avalanches?

Think back over where your placed your pixels. What kinds of locations produce little change, and which produce major avalanches?

18. What Else is Critical?

We spent time in an earlier experiment examining the effects of changing Survival and Birth neighbors, that is, information flow. Let's briefly revisit that again.

☞ Reload file of random pixels you saved on the S drive.

☞ Then redo the experiments you have just completed with the Survival/Birth neighbor combinations listed below.

18. Experimental Record Eighteen – What Else is Critical?

Go to **World: click the globe: Birth/survival Neighbors** and set the **Survival Neighbors** to **3/3.** Leave the Birth neighbors at the default of 3-3.

☞ Run the experiment as you did for number 3 above, recording your early results in the table below.

	Generation Number	Generation Difference		Generation Number	Generation Difference
6th Generation			3rd Generation		
Difference 5th and 6th=			Difference 2nd and 3rd=		
5th Generation			2nd Generation		
Difference 4th and 5th=			Difference 1st and 2nd=		
4th Generation			1st Generation		
Difference 3rd and 4th=					

Go to **WORLD: CLICK THE GLOBE: BIRTH/SURVIVAL NEIGHBORS**. Click **RESTORE DEFAULTS**, and then set the **Birth Neighbors** to **2/3.** Leave the Survival neighbors at the default of 2/3.

☞ Run the experiment as you did for number 3 above, recording your early results in the table below. (Go Back to the Globe and change Birth/Survival to the defaults while the simulation is running and observe the outcome)

	Generation Number	Generation Difference		Generation Number	Generation Difference
6th Generation			3rd Generation		
Difference 5th and 6th=			Difference 2nd and 3rd=		
5th Generation			2nd Generation		
Difference 4th and 5th=			Difference 1st and 2nd=		
4th Generation			1st Generation		
Difference 3rd and 4th=					

Go to **WORLD: CLICK THE GLOBE: BIRTH/SURVIVAL NEIGHBORS**. Click **RESTORE DEFAULTS**, and then set the **Birth Neighbors TO 2/2.** Leave the Survival neighbors at the default of 2/3.

☞ Run the experiment as you did for number 3 above, recording your early results in the table below.

☞ Click **SUSPEND** add a pixel at random. Does the system exhibit sensitive dependence?

	Generation Number	Generation Difference		Generation Number	Generation Difference
6th Generation			3rd Generation		
Difference 5th and 6th=			Difference 2nd and 3rd=		
5th Generation			2nd Generation		
Difference 4th and 5th=			Difference 1st and 2nd=		
4th Generation			1st Generation		
Difference 3th and 4th=					

Do these results surprise you, considering all your other experimental work with cellular automata?

What do you think is the likelihood that another critical set of conditions (Birth/Survival) exists?

☺ F ☺ i ☺ n ☺ i ☺ s ☺