CS 7A - Fall 2015 - John Conway's Game of Life. Due 9/28/15

Background Theory

The following is from Martin Gardner's book, *Wheels, Life and Other Mathematical Amusements.*

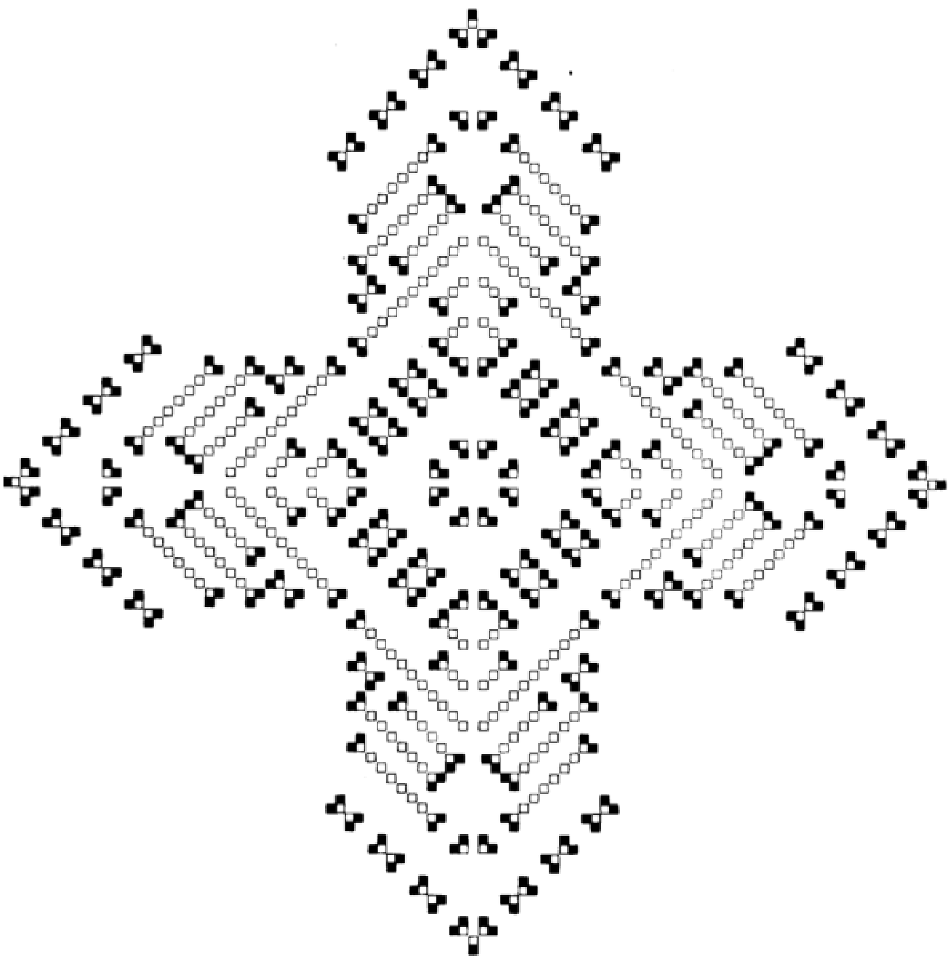
Cellular automata theory began in the mid-nineteen-fifties when John von Neumann set himself the task of proving that self-replicating machines were possible. Such a machine, given proper instructions, would build an exact duplicate of itself. Each of the two machines would then build another, the four would become eight, and so on. (This proliferation of self-replicating automata is the basis of Lord Dunsany's amusing 1951 novel *The Last Revolution.*) Von Neumann first proved his case with "kinematic" models of a machine that could roam through a warehouse of parts, select needed components and put together a copy of itself. Later, adopting an inspired suggestion by his friend Stanislaw M. Ulam, he showed the possibility of such machines in a more elegant and abstract way.

Von Neumann's new proof used what is now called a "uniform cellular space" equivalent to an infinite checkerboard. Each cell can have any finite number of "states," including a "quiescent" (or empty) state, and a finite set of "neighbor" cells that can influence its state. The pattern of states changes in discrete time steps according to a set of "transition rules" that apply simultaneously to every cell. The cells symbolize the basic parts of a finite-state automaton and a configuration of live cells is an idealized model of such a machine. Conway's game of "Life" is based on just such a space. His neighborhood consists of the eight cells surrounding a cell; each cell has two states (empty or filled), and his transition rules are the birth, death and survival rules I explained in the previous chapter. Von Neumann, applying transition rules to a space in which each cell has 29 states and four orthogonally adjacent neighbors, proved the existence of a configuration of about 200,000 cells that would self-reproduce.

The reason for such an enormous configuration is that, for von Neumann's proof to apply to actual automata, it was necessary that his cellular space be capable of simulating a Turing machine: an idealized automaton, named for its inventor, the British mathematician A. M. Turing, capable of performing any desired calculation. By embedding this universal computer in his configuration, von Neumann was able to produce a universal constructor. Because it could in principle construct any desired configuration by stretching "arms" into an empty region of the cellular space, it would self-replicate when given a blueprint of itself. Since von Xeumann's death in 1957 his existence proof (the actual configuration is too vast to construct and manipulate) has been greatly simplified. The latest and best reduction, by Edwin Roger Banks, a mechanical engineering graduate student at the Massachusetts Institute of Technology, does the job with cells of only four states. Self-replication in a trivial sense-without using configurations that contain Turing machines-is easy to achieve. A delightfully simple example, discovered by Edward Fredkin of M.I.T. about 1960, uses two-state cells, the von Neumann neighborhood of four orthogonally adjacent cells and the following parity rule: Each cell with an even number of live neighbors (0, 2, 4) at time t becomes or remains empty at time $t+1$, and each cell with an odd number of neighbors (1, 3) at time t becomes or remains live at time $t+1$. It is not hard to show that after 2^n ticks (n varying with different patterns) any initial pattern of live cells will reproduce itself four times-above, below, left and right of an empty space that it formerly occupied. The four replicas will be displaced 2^n cells from the vanished original. The new pattern will, of course, replicate again after another 2^n steps, so that the duplicates keep quadrupling in the endless series $1, 4, 16, 64, \ldots$ Figure 136 shows two quadruplings of a right tromino. Terry Winograd, in a 1967 term paper written when he was an M.I.T. student, generalized Fredkin's rule to other neighborhoods, any number of dimensions and cells with any prime number of states.

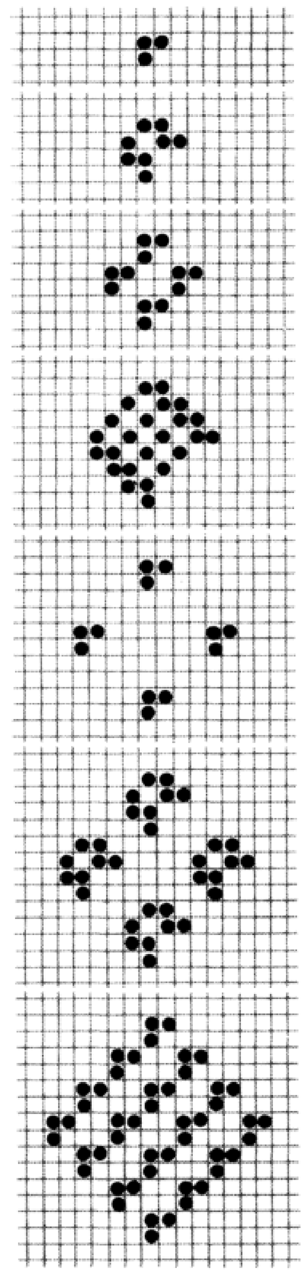
Ulam investigated a variety of cellular automata games, experimenting with different neighborhoods, numbers of states and transition rules. In a 1967 paper "On Recursively Defined Geometrical Objects and Patterns of Growth," written with Robert G. Schrandt, Ulam described a number of different games. Figure 137 shows generation 45 of a history that began with one counter on the central cell. As in Conway's game, the cells are two-state, but the neighborhood is that of von Neumann (four adjacent orthogonal cells). Births occur on cells that have one and only one neighbor, and all live cells of generation n vanish when generation $n+2$ is born. In other words, only the last two generations survive at any step. In Figure 137 the 444 new births are shown as black cells. The 404 white cells of the preceding generation will all disappear on the next tick. Note the characteristic subpattern, which Ulam calls a "dog bone." Ulam experimented with games in which two configurations were allowed to grow until they collided. In the ensuing "battle" one side would sometimes wipe out the other; sometimes both armies would be annihilated. Ulam also explored games on three-dimensional cubical tessellations. His major papers on cellular automata are in *Essays on Cellular Automata*, edited by Arthur W. Burks.

Figure 137



Generation 45 in a cellular game devised by Stanislaw M. Ulam

Figure 136



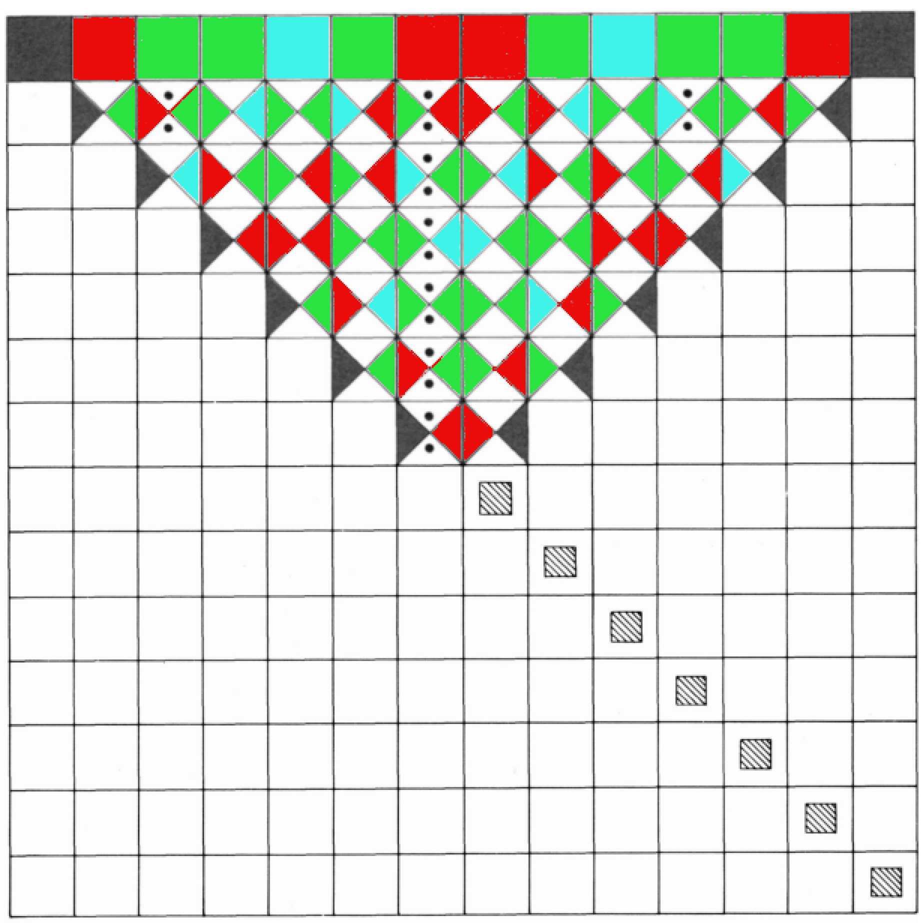
The replication of a tromino

Similar games can he devised for triangular and hexagonal tessellations but, although they look different, they are not essentially so. All can be translated into equivalent games on a square tessellation by a suitable definition of "neighborhood." A neighborhood need not be made up of touching cells. In chess, for instance, a knight's neighborhood consists of the squares to which it can leap and squares on which there are pieces that can attack it. As Burks has pointed out, games such as chess, checkers and go can be regarded as cellular automata games in which there are complicated neighborhoods and transition rules and in which players choose among alternative next states in an attempt to be first to reach a certain final state that wins.

Among the notable contributions of Edward F. Moore to cellular automata theory the best-known is a technique for proving the existence of what John W. Tukey named "Garden of Eden" patterns. These are configurations that cannot arise in a game because no preceding generation can form them. They appear only if given in the initial (zero) generation. Because such a configuration has no predecessor, it cannot be self-reproducing. I shall not describe Moore's ingenious technique because he explained it informally in an article in *Scientific American* (see "Mathematics in the Biological Sciences," by Edward F. Moore; September, 1964) and more formally in a paper that is included in Burks's anthology.

Alvy Ray Smith 111, a cellular automata expert at Kew York University's School of Engineering and Science, found a simple application of Moore's technique to Conway's game. Consider two five-by-five squares, one with all cells empty, the other with one counter in the center. Because, in one tick, the central nine cells of both squares are certain to become identical (in this case all cells empty) they are said to be "mutually erasable." It follows from Moore's theorem that a Garden of Eden configuration must exist in Conway's game. Unfortunately the proof does not tell how to find such a pattern and so far none is known. It may be simple or it may be enormously complex. Using one of Moore's formulas, Smith has been able to calculate that such a pattern exists within a square of 10 billion cells on a side, which does not help much in finding one.

Smith has been working on cellular automata that simulate pattern-recognition machines. Although this is now only of theoretical interest, the time may come when robots will need "retinas" for recognizing patterns. The speeds of scanning devices are slow compared with the speeds obtainable by the "parallel computation" of animal retinas, which simultaneously transmit thousands of messages to the brain. Parallel computation is the only way new computers can increase significantly in speed because without it they are limited by the speed of light through miniaturized circuitry. The cover of the February, 1971, issue of *Scientific Arnerican* [reproduced in Figure 138) shows a simple procedure, devised by Smith, by which a finite one-dimensional cellular space employs parallel computation for recognizing palindromic symmetry. Each cell has many possible states, the number depending on the number of different symbols in the palindrome, and a cell's neighborhood is the two cells on each side.



Cellular automaton

Smith symbolizes the palindrome TOO HOT TO HOOT with four states of cells in the top row. T, 0 and H are represented by red, green and blue respectively, and black marks the palindrome's two ends. The white cells in the other rows are in the quiescent state. The horizontal rows below the top row are successive generations of the top configuration when certain transition rules are followed in discrete time steps. In other words, the picture is a space-time diagram of a single row, each successive row indicating the next generation.

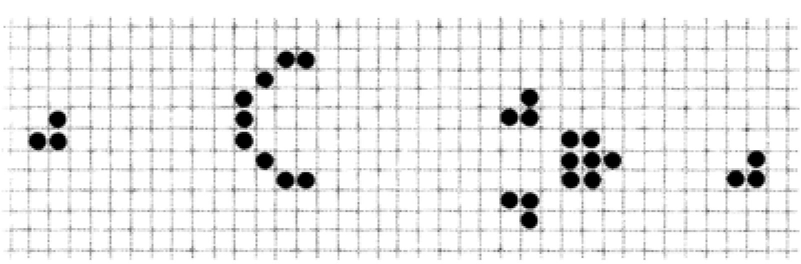
In the first transition each shade travels one cell to the left and one cell to the right, except for the end colors, which are blocked by black; black moves inward at each step. Each cell on which two colors land acquires a new state, symbolized by a cell divided into four triangles. The left triangle has the color that was previously on the left, the right triangle has the shading previously on the right. The result of this first move is shown in the second row. When an adjacent pair of cells forms a tilted square in the center that is a solid shading, it indicates a "collision" of like shadings and is symbolized by black dots in the two white triangles of the left cell. Dots remain in that cell for all subsequent generations unless a collision of unlike shadings occurs to the immediate right of the dotted cell, in which case the dots are erased. When collisions of unlike shadings occur, the left cell of the pair remains undotted for all subsequent generations even though like shadings may later collide on its right.

At each move the shadings continue to travel one cell left or right (the direction in which the shaded triangles point) and all rules apply. If the palindrome has n letters, with n even as in this example (the scheme is modified slightly if n is odd), it is easy to see that after $n/2$ moves only two adjacent nonquiescent cells remain. If the left cell of this pair is dotted, the automaton has recognized the initial row as being palindromic. Down the diagram's center you see the colliding pairs of like shadings in the same order as they appear on the palindrome from the center to each end. As soon as recognition occurs the left cell of the last pair is erased and the right cell is altered to an "accept" state, here symbolized by nested squares. An undotted left cell would signal a nonpalindrome, in which case the left cell would become blank and the right cell would go into a "reject" state.

A Turing machine, which computes serially, requires in general n^2 steps to recognize a palindrome of length n. Although recognition occurs here at step $n/2$, the accept state is shown moving in subsequent generations to the right to symbolize the cell-by-cell transmission of the acceptance to an output boundary of the cellular space. Of course it is easy to construct more efficient palindrome-recognizing devices with actual electronic hardware, but the point here is to do it with a highly abstract, one-dimensional cellular space in which information can pass only from a cell to adjacent cells and not even the center of the initial series of symbols is known at the outset. As Smith puts it anthropomorphically, after the first step each of the three dotted cells thinks it is at the center of a palindrome. The dotted cells at each end are disillusioned on the next move because of the collision of unlike shadings at their right. Not until generation $n/2$ does the dotted cell at the center know it *is* at the center.

Now for some startling new results concerning Conway's game. Conway was fully aware of earlier games and it was with them in mind that he selected his recursive rules with great care to avoid two extremes: too many patterns that grow quickly without limit and too many that fade quickly. By striking a delicate balance he designed a game of surprising unpredictability and one that produced such remarkable figures as oscillators and moving spaceships. He conjectured that no finite population could grow (in number of members) without limit, and he offered $30 for the first proof or disproof. The prize was won in November, 1970, by a group in the Artificial Intelligence Project at M.I.T. consisting of (in alphabetical order) Robert April, Michael Beeler, R. William Gosper, Jr., Richard Howell, Rich Schroeppel and Michael Speciner. Using a program devised by Speciner for displaying life histories on an oscilloscope, Gosper made a truly astounding discovery: he found a glider gun! The configuration in Figure 139 grows into such a gun, firing its first glider on tick 40. The gun is an oscillator of period 30 that ejects a new glider every 30 ticks. Since each glider adds five more counters to the field, the population obviously grows without limit.

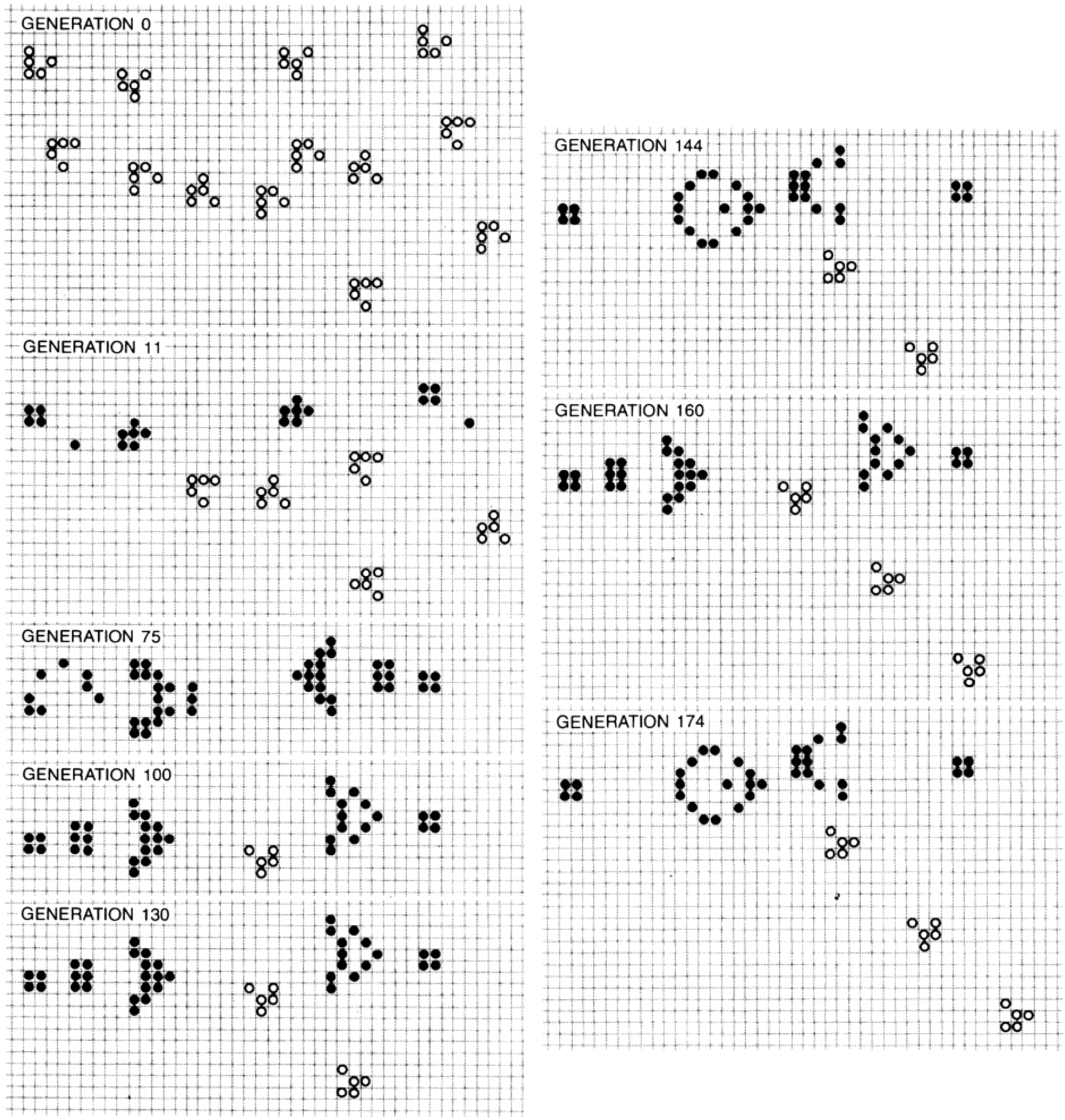
Figure 139



A configuration that grows into a glider gun

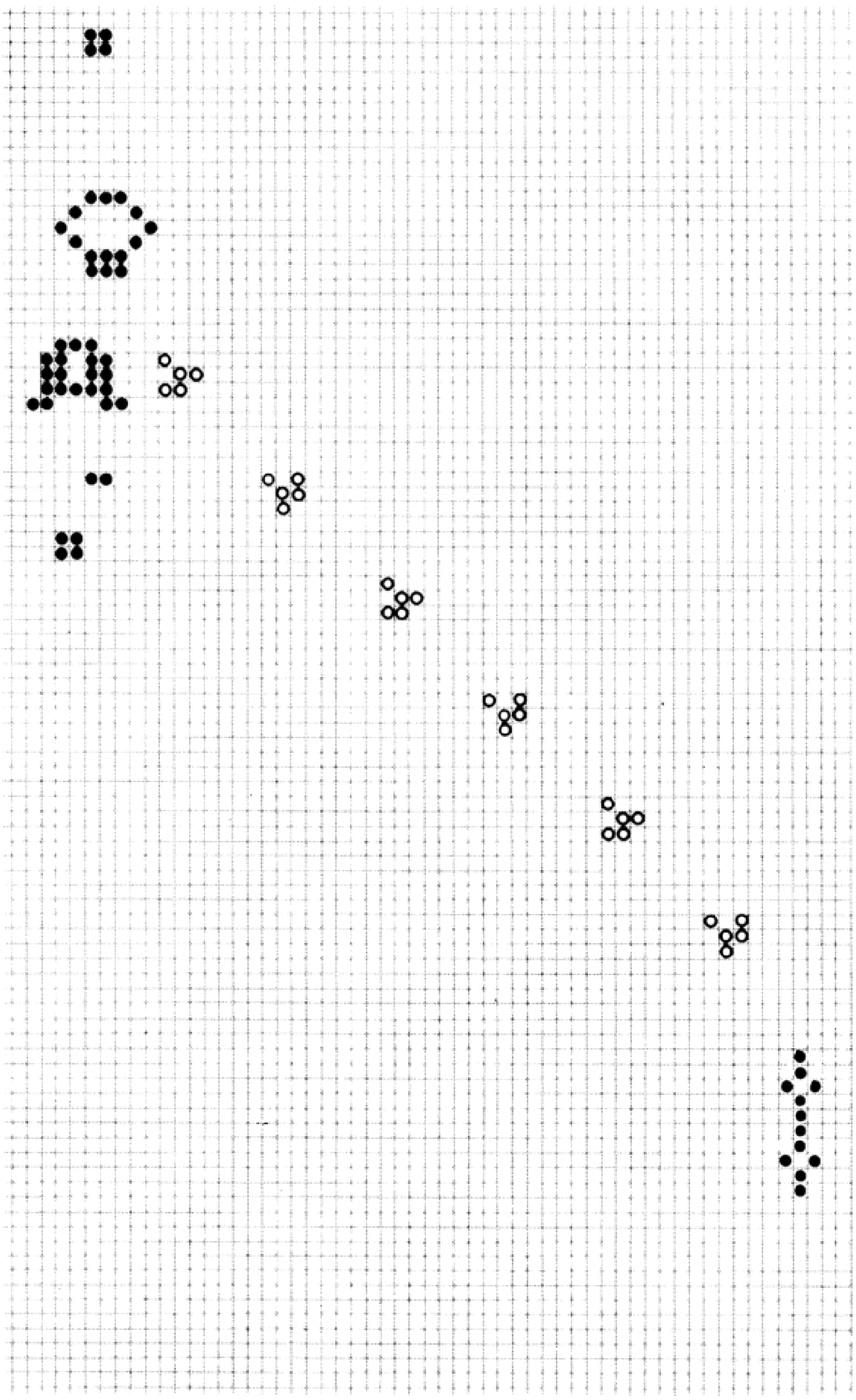
The glider gun led the M.I.T. group to many other amazing discoveries. A series of printouts (supplied by Robert T. Wain- wright of Yorktown Heights, N.Y.) shows how 13 gliders crash to form a glider gun [see Figure 1401. The last five printouts show the gun in full action. The group also found a way to position a pentadecathlon [see Figure 1411, an oscillator of period 15, so that it "eats" every glider that strikes it. A pentadecathlon can also reflect a glider 180 degrees, making it possible for two pentadecathlons to shuttle a glider back and forth forever. Streams of intersecting gliders produce fantastic results. Strange patterns can be created that in turn emit gliders. Sometimes collision configurations grow until they ingest all guns. In other cases the collision mass destroys one or more guns by shooting back. The group's latest burst of virtuosity is a way of placing eight guns so that the intersecting streams of gliders build a factory that assembles and fires a middleweight spaceship about every 300 ticks.

Figure 140



Here and on the facing page 13 gliders crash to form a glider gun (generation 75) that oscillates with a period of 30, firing a glider in each cycle

Figure 141

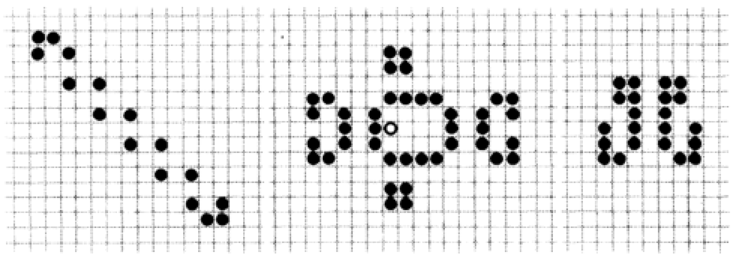


Pentadecathlon *(bottom right)* "eats" gliders fired by the gun

The existence of glider guns raises the exciting possibility that Conway's game will allow the simulation of a Turing machine, a universal calculator capable in principle of doing anything the most powerful computer can do. The trick would be to use gliders as unit pulses for storing and transmitting information and performing the required logic operations that are handled in actual computers by their circuitry. If Conway's game allows a universal calculator, the next question will be whether it allows a universal constructor, from which nontrivial self replication would follow. So far this has not been achieved with a two-state space and Conway's neighborhood, although it has been proved impossible with two states and the von Neumann neighborhood.

The M.I.T. group found many new oscillators [see Figure 142]. One of them, the barber pole, can be stretched to any length and is a flip-flop, with each state a mirror image of the other. Another, which they rediscovered, is a pattern Conway's group had found earlier and called a Hertz oscillator. Every four ticks the hollow "hit" switches from one side of the central frame to the other, making it an oscillator of period 8. The tumbler, which was found by George D. Collins, Jr., of McLean, Va., turns upside down every seven ticks.

Figure 142

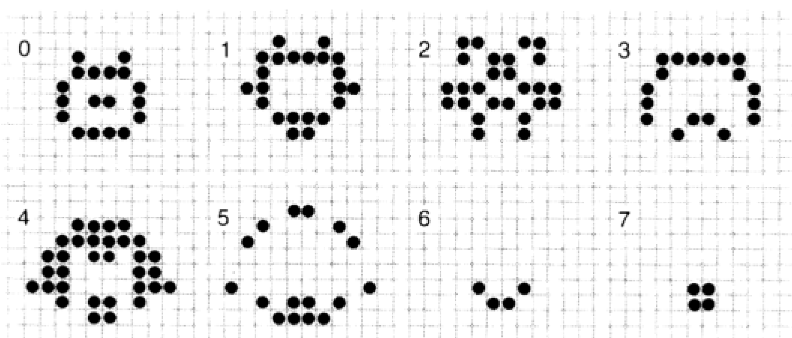


Barber pole *(left)*, Hertz oscillator *(middle)*, and tumbler *(right)*

The Cheshire cat [see Figure 143] was discovered by C. R. Tompkins of Corona, Calif. On the sixth tick the face vanishes, leaving only a grin; the grin fades on the next tick and only a permanent paw print (block) remains. The harvester was constructed by David W. Poyner of Basildon in England. It plows up an infinite diagonal at the speed of light, oscillating with period 4 and ejecting stable packages along the way [see Figure 144]. "Unfortunately," writes Poyner, "I have been unable to develop a propagator that will sow as fast as the harvester will reap."

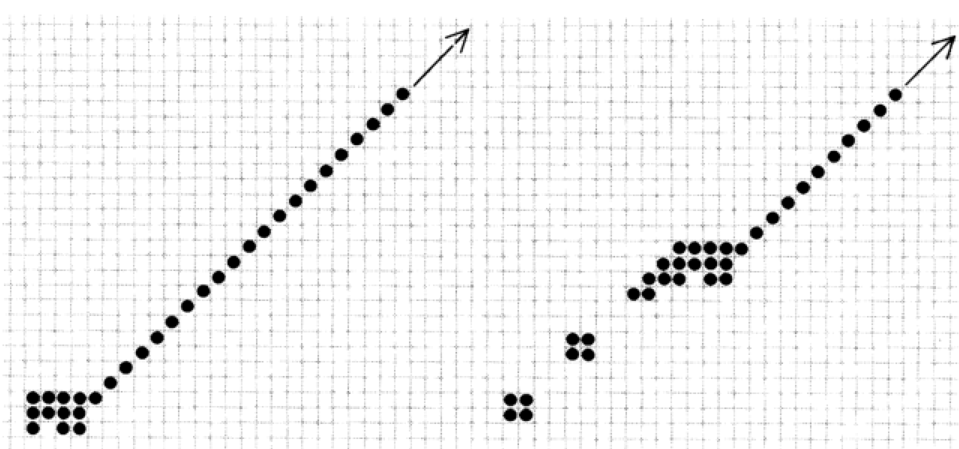
Wainwright has made a number of intriguing investigations. He filled a 120-by-120 square field with 4,800 randomly placed bits (a density of one-third) and tracked their history for 450 generations, by which time the density of this primordial soup, as Wainwright calls it, had thinned steadily to one-sixth.

Figure 143



The Cheshire cat (0) fades to a grin (6) and disappears, leaving a paw print (7)

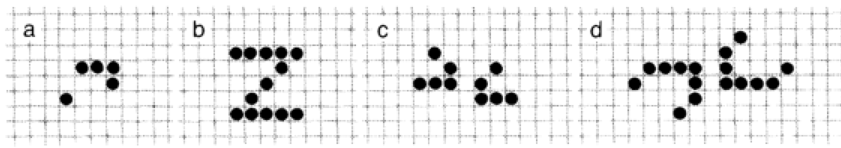
Figure 144



The harvester, shown at generations (0) *left* and 10 *(right)*

Whether it would eventually vanish or, as Wainwright says, percolate at a constant minimum density is anybody's guess. At any rate, during the 450 generations 42 short-lived gliders were formed. Wainwright found 14 different patterns that became glider states on the next tick. The most common pattern to produce a glider on the next tick is shown [*a in Figure* 145]. A Z-pattern found by Collins and by Jeffrey Lund of Pewaukee, Wis., after 12 ticks becomes two gliders that sail off in opposite directions [*b in Figure* 145]. Wainwright and others set two gliders on a collision course that causes all bits to vanish on the fourth tick [*c in Figure* 145]. Wallace W. Wagner of Anaheim, Calif., found a collision course for two lightweight space-ships that also ends (on the seventh tick) in total blankness [*d in Figure* 145].

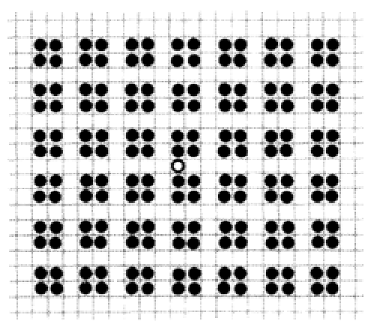
Figure 145



Two spawners of gliders and two collision courses

Wainwright has experimented with various infinite fields of regular stable patterns, which he calls agars-rich culture mediums. When, for instance, a single "virus," or bit, is placed in the agar of blocks shown in Figure 146 so that it touches the corners of four blocks, the agar eliminates the virus and repairs itself in two ticks. If, however, the alien bit is positioned as shown (or at any of the seven other symmetrically equivalent spots), it initiates an inexorable disintegration of the pattern. The portion eaten away contains active debris that has overall bilateral symmetry along one axis and a roughly oval border that expands, probably forever, in the four compass directions at the speed of light.

Figure 146



Agar doomed by a virus

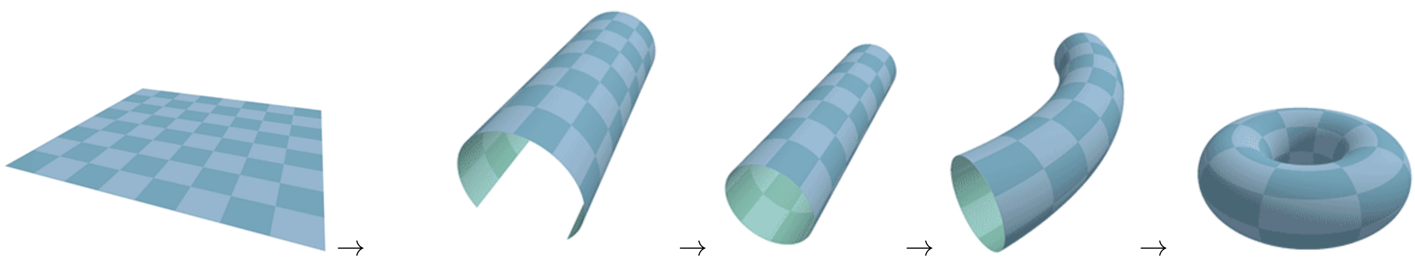
The most immediate practical application of cellular automata theory, Banks believes, is likely to be the design of circuits capable of self-repair or the wiring of any specified type of new circuit. No one can say how significant the theory may eventually become for the physical and biological sciences. It may have important bearings on cell growth in embryos, the replication of DNA molecules, the operation of nerve nets, genetic changes in evolving populations and so on. Analogies with life processes are impossible to resist. If a primordial broth of amino acids is large enough, and there is sufficient time, self-replicating, moving automata may result from complex transition rules built into the structure of matter and the laws of nature. There is even the possibility that space-time itself is granular, composed of discrete units, and that the universe, as Fredkin and others have suggested, is a vast cellular automaton run by an enormous computer. If so, what we call motion may be only simulated motion. A moving spaceship, on the ultimate microlevel, may be essentially the same as one of Conway's spaceships, appearing to move on the macrolevel whereas actually there is only an alteration of states of basic space-time cells in obedience to transition rules that have not yet been discovered.

Projects due by Wed. 9/14.

1. A simple example of self-replication uses two-state cells (dead and alive), the von Neumann neighborhood of four orthogonally adjacent cells (up, down, left and right) and the following parity rule: Each cell with an even number of live neighbors (0, 2, 4) at time t becomes or remains empty at time $t+1$, and each cell with an odd number of neighbors (1, 3) at time t becomes or remains live at time $t+1$.

   Modify the life code you wrote to allow the user to choose whether to use the Conway rules or the von Neumann rules. How does that change the dynamics of the evolution of the starting configurations we looked at (R pentomino, H, etc.)? Write a brief description of how each changes.

2. Ideally the grid the life patterns evolve on has no boundaries, but as a practical coding consideration, that's pretty difficult (especially if you're working with text, rather than graphics.) To get rid of the boundaries without getting a grid too large to disply, we can introduce the toroid topology, the grid is transformed into a torus, like so:

   Allow the user to choose the dimension of the `width` by `length` rectangle (instead of being confined to a square) and then allow the user to (optionally) use a doughnut topology by making cell in column `pos==k*width` a side-by-side neighbor of `pos==k*width+width-1` (and so in the Conway rules, `pos==k*width` would also have neighbors `pos==k*width-1`and `pos==(k+2)*width-1`above and below, and so on.) That produces a cylindrical playing field (which acts like cylinder, but we still view it as a rectangle.) To complete the doughnut, associate each cell in the to row with the cell in the same column in the bottom row, so that these are neighbors. So that cell `pos%k` is a neighbor of `pos%k+(length-1)*width` and also the cells to the right and left of that.

   How does that change the dynamics of the evolution of the starting configurations we looked at (R pentomino, H, etc.)? Write a brief description of how each changes.

Here's a Youtube showing we aren't the first to consider this... life on a doughnut