Cellular Automata

CAMUS

Cellular Automata Music

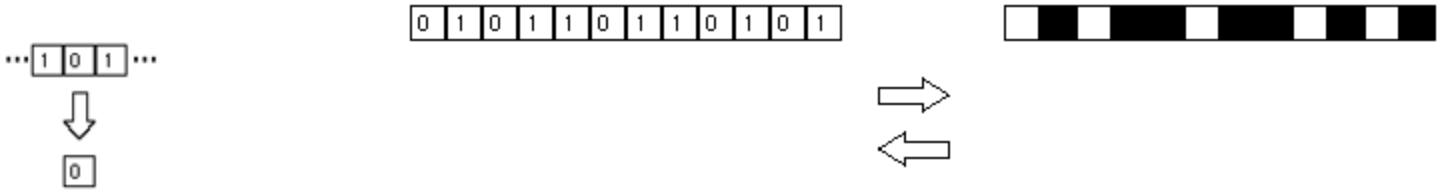
By Eduardo Miranda

Generalized CA Model

- Discrete model in Mathematical Abstraction
  - Regular grid of *Cells*
  - Each cell in one of a finite number of states
  - Time is discrete
  - The state of a cell at time *t* is a function of the states of a finite number of cells (its *neighborhood*) at time *t-1*
  - All cells have the same rule for updating
  - Each time the rules are applied to a grid, a new *generation* is created

Illustration

- A simple example
  - An array of 12 cells
  - Each cell can value either 0 or 1
  - The right side displays as colours: 0=white and 1=black.
- *CA rule in action*
  - At each tick of the clock, the values of all 12 cells change simultaneously according to a set of rules
  - If both neighbours are the same (0s or 1s), then this cell equals zero in the next stage, otherwise it becomes a 1



Computation with CAs

- John Conway's Game of Life (1970s)
  - Gliders – arrangements of cells which move themselves across the grid
    - Can interact to perform computations
- Stephen Wolfram's Rule 101
  - Turing Complete
    - Can perform universal computation
    - Every Turing-computable function

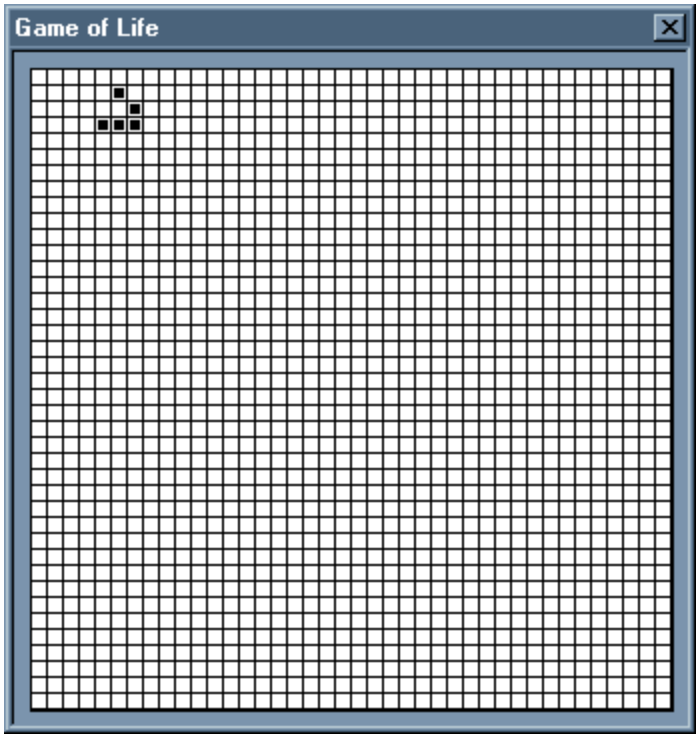
Game of Life

- A cell can be either alive (black) or dead (white)
- Conway's Rules
  - For a space that is 'populated'
    - Each cell with one or no neighbors dies, as if by loneliness
    - Each cell with four or more neighbors dies, as if by overpopulation
    - Each cell with two or three neighbors survives
  - For a space that is 'empty' or 'unpopulated'
    - Each cell with three neighbors becomes populated
- Rules are applied simultaneously to all cells of the lattice

Game of Life

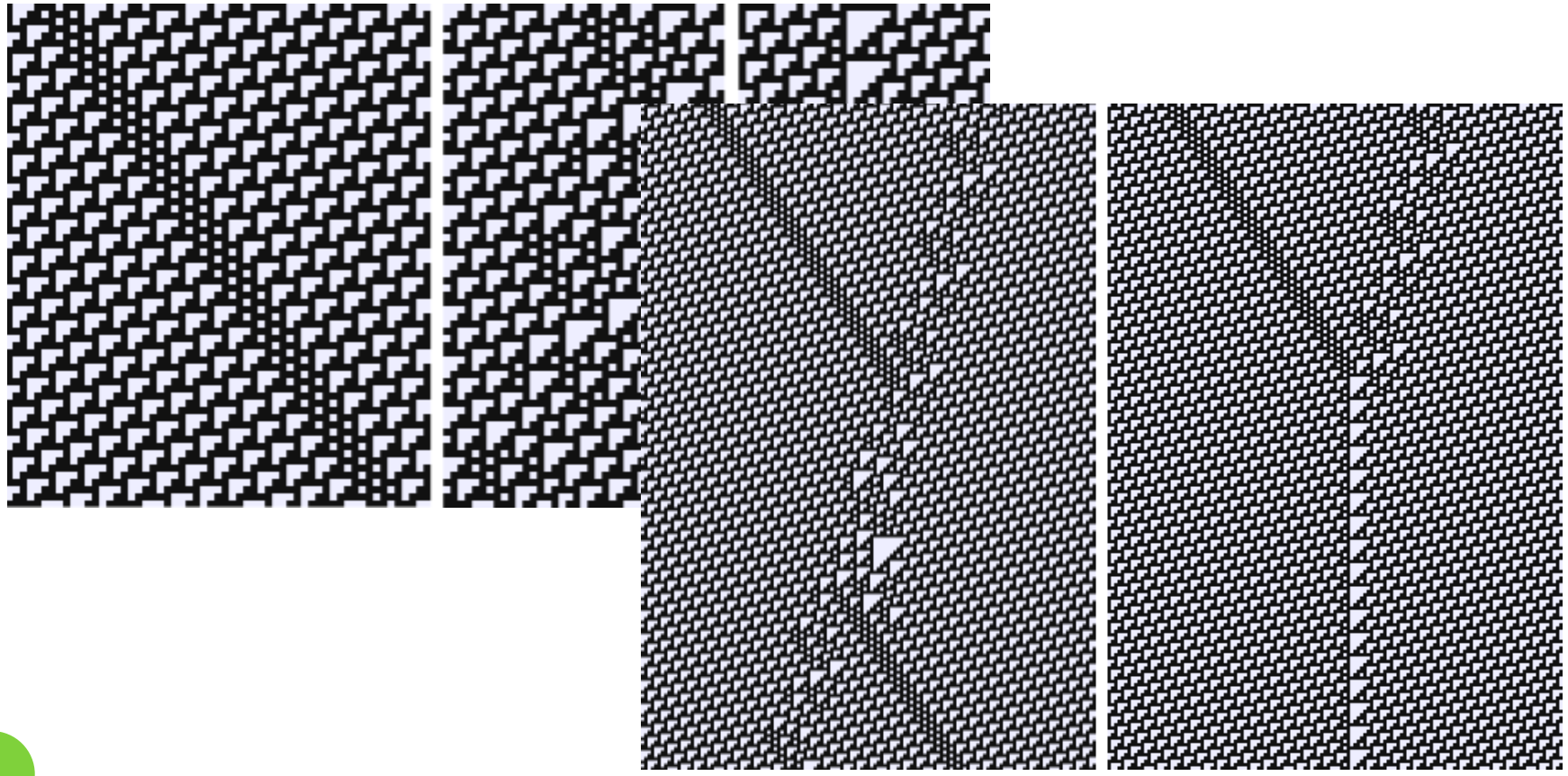
http://www.bitstorm.org/gameoflife/

Application



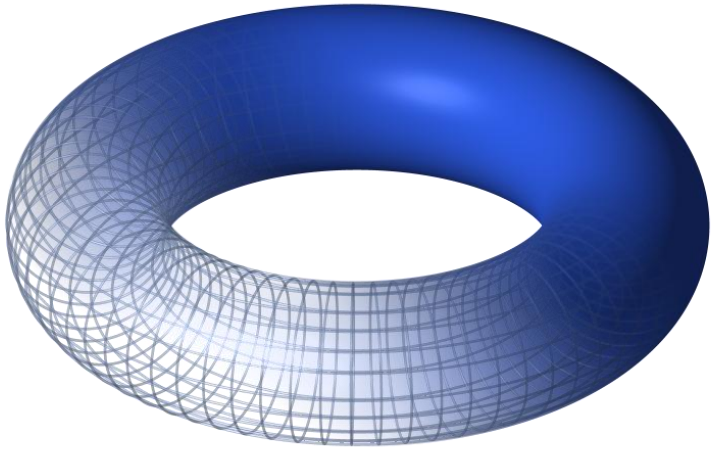
Rule 101 Computation

current pattern	111	110	101	100	011	010	001	000
new state for center cell	0	1	1	0	1	1	1	0



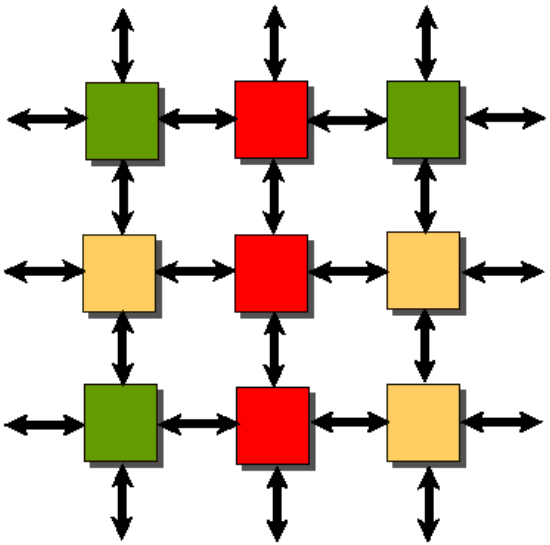
Toroidal Space

- A plane would require edge rules / boundary conditions
- Most CA therefore utilize a toroidal field



More Sophisticated

- Multi-state CA configurations
  - Can assume values other than 0 and 1
  - Transition rules normally compute four (von Neuman) or eight (Moore) neighbours



Project Questions & Motivations

- Can computers create new kinds of music rather than just mimic performers?
- Difficult to judge computer generated compositions because they lack the cultural references we normally use when appreciating music
- Design a study on whether cellular automata that exhibit pattern propagation behaviour could be used or adapted to model the propagation of musical patterns

CAMUS

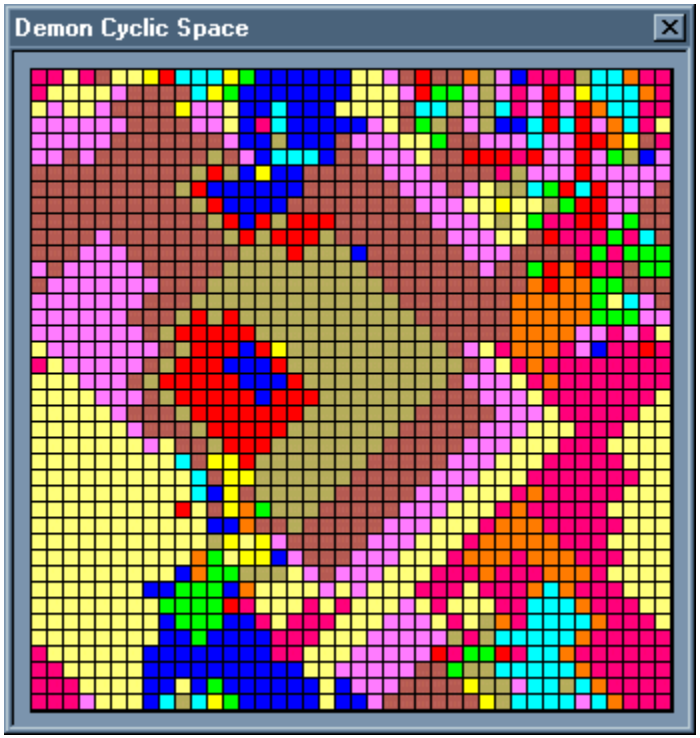
- A Cellular Automata MUSic Generator
  - Cellular automata produce large amounts of patterned
  - Music composition is based on pattern propagation and formal manipulation of its parameters
  - CAMUS maps CA into a music representation in order to generate compositional material
- Algorithms used:
  - Game of Life (invented by John Horton Conway)
  - Demon Cyclic Space (designed by David Griffeath)

Demon Cyclic Space

- Initialized as a random distribution of coloured cells
- Always end up with stable, angular spirals reminiscent of crystalline growths

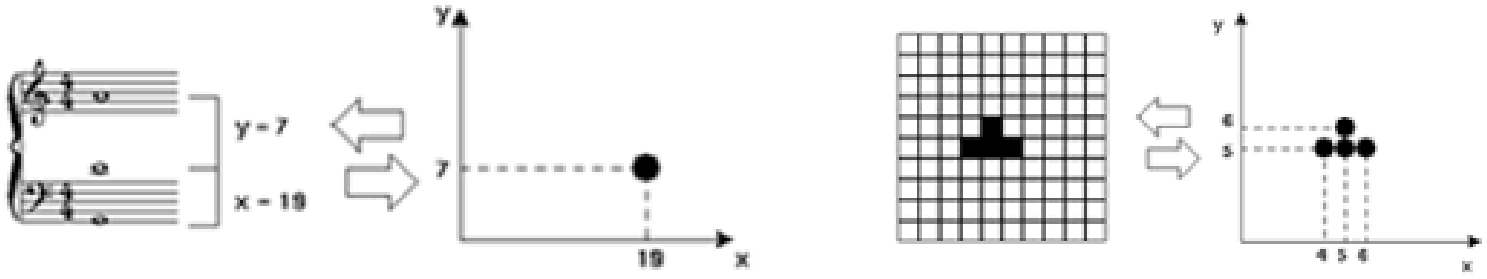
- Each of the n possible states for a cell is represented by a different colour and numbered from 0 to *n*-1
- A cell that happens to be in state *k* at one tick of the clock dominates any adjacent cells that are in state *k*-1
  - Adjacent cells change from *k*-1 to *k*.
  - This rule resembles a natural chain in which a cell in state 2 can dominate a cell in state 1 even if the later is dominating a cell in state 0.
  - Since the automaton is cyclic, the chain has no end and a cell in state 0 dominates its neighbouring cells that are in state *n*-1 .

Demon Cyclic Space



The musical engine of CAMUS

- The system uses both automata in parallel to produce music.
- Game of Life
  - CAMUS uses a Cartesian model in order to represent a triplet; that is, a set of three notes. The model has two dimensions, where the horizontal coordinate represents the first interval of the triple and the vertical coordinate represents its second interval.



The musical engine of CAMUS

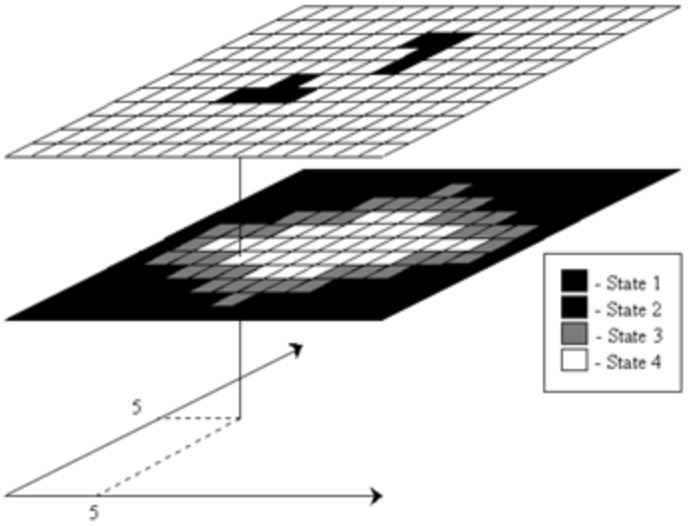
- Demon Cyclic Space
  - DCS automaton determines the "orchestration" of the composition.
  - Each colour corresponds to an instrument (MIDI) designated to perform the notes generated by a specific cell.
  - Each musical cell has its own timing, but the notes within a cell can assume different durations and can be triggered at different times.

Automaton Initialization

- To begin the music process:
  - The Game of Life automaton is set up with a starting configuration
  - The Demon Cyclic Space automaton is initialized with random states
  - Both are set to run
- At each time step, the co-ordinates of each live cell are analyzed and used to determine a triple which will be played at the corresponding time in the composition.
- The state of the corresponding cell of the Demon Cyclic Space automaton is used to determine the orchestration of the piece.

Illustrated Example

- The cell in the Game of Life at (5, 5) is alive, thus constitutes a sonic event (that is, a set of three notes).
- The corresponding cell in the Demon Cyclic Space is in state 4, the sonic event would be played by instrument number four (e.g., using MIDI channel 4)
- The co-ordinates (5, 5) describe the intervals in a triplet: the fundamental pitch, the note five semitones above the fundamental, and the note ten semitones above the fundamental.



Timing & Shape

- Temporal positioning of the cell (x, y)
  - Value being **1** if the cell is alive and **0** if it is dead:

$$\boldsymbol{a} = \text{cell}\,(x, y - 1) \qquad \boldsymbol{b} = \text{cell}\,(x, y + 1)$$

$$\boldsymbol{c} = \text{cell}\,(x + 1, y) \qquad \boldsymbol{d} = \text{cell}\,(x - 1, y)$$

$$\boldsymbol{m} = \text{cell}\,(x - 1, y - 1) \qquad \boldsymbol{n} = \text{cell}\,(x + 1, y + 1)$$

$$\boldsymbol{o} = \text{cell}\,(x + 1, y - 1) \qquad \boldsymbol{p} = \text{cell}\,(x - 1, y + 1)$$

- Form four 4-bit words, ***abcd***, ***dcba***, ***mnop*** and ***ponm*** and perform the bitwise inclusive OR

$$\boldsymbol{Tgg\ (trigger) = abcd \mid dcba}$$

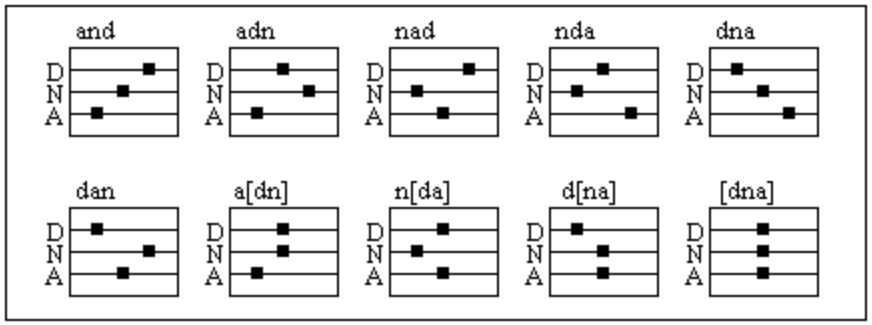
$$\boldsymbol{Dur\ (duration) = mnop \mid ponm}$$

  - Assigned random values with bounds determined by the pattern associated to their AND values

Time and Shape Cont.

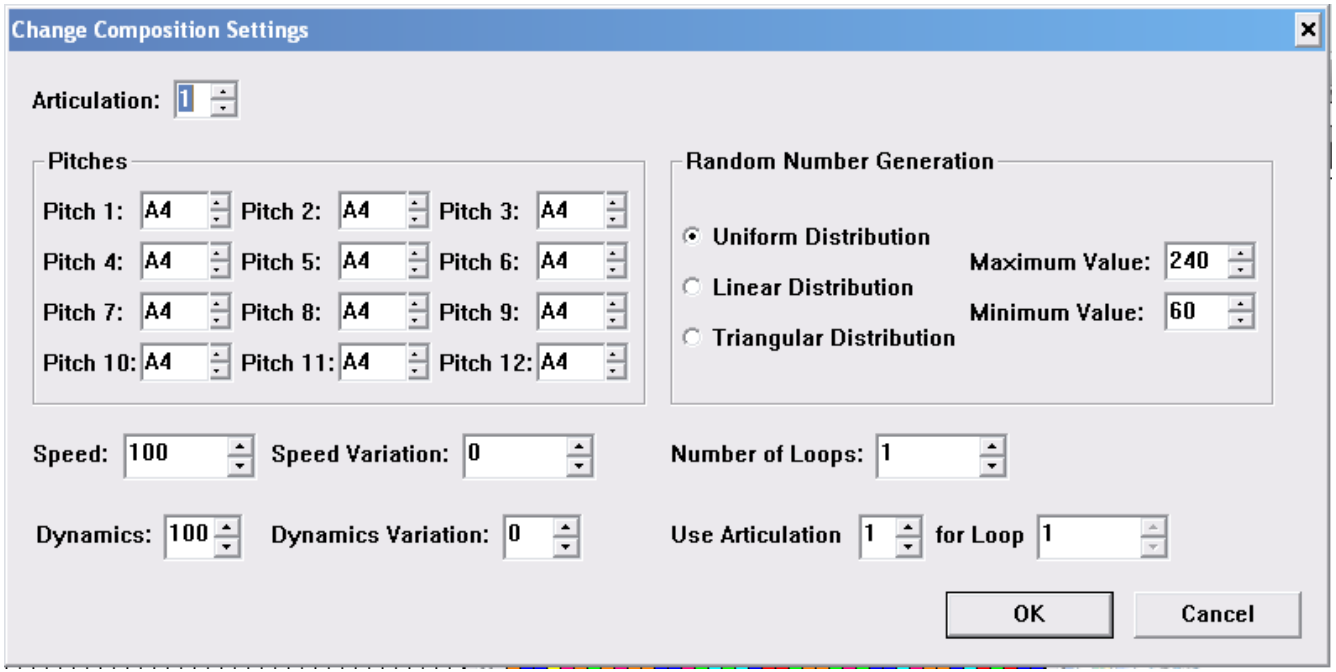
- Associate a code, known as an **AND**
  - (cellul**A**r ge**N**etic co**D**e)
  - **A** denoting the lower pitch, **N** the middle pitch, and **D** the upper.
  - Square brackets are used to indicate that the note events contained within that bracket occur simultaneously.

0000 : a[dn]	0001: [dna]	0010 : adn	0011 : dna	0101 : and
0110 : dan	0111 : nad	1001 : d[na]	1011 : nda	1111 : n[da]



Articulations & Looping

- Articulations are arrangements of variable assignment
- Different articulations can be assigned to different loops
- Users can set up to 9999 loops
- When a sequence of loops has completed, CAMUS starts at the first loop again



Commentary and Results

- CAMUS deals with musical forms at two levels:
  - Internal organization of triplets
    - AND-code
  - External organization of triplets in time
    - Game of Life
- Musical representations mapping was arbitrary
- Concluded: Cellular automata are appropriate for generating musical forms and CAMUS can produce interesting musical sequences
  - Validation: Two pieces composed by CAMUS for chamber orchestra and electro-acoustic ensemble were performed and well received by audiences

Demonstration

- *Entre l'Absurde et le Mystère* 
- *Wee Batucada Scotica*
- *CAMUS as Source Material* 
  - *Original Pitch used, rhythm and instrumentation adjusted*