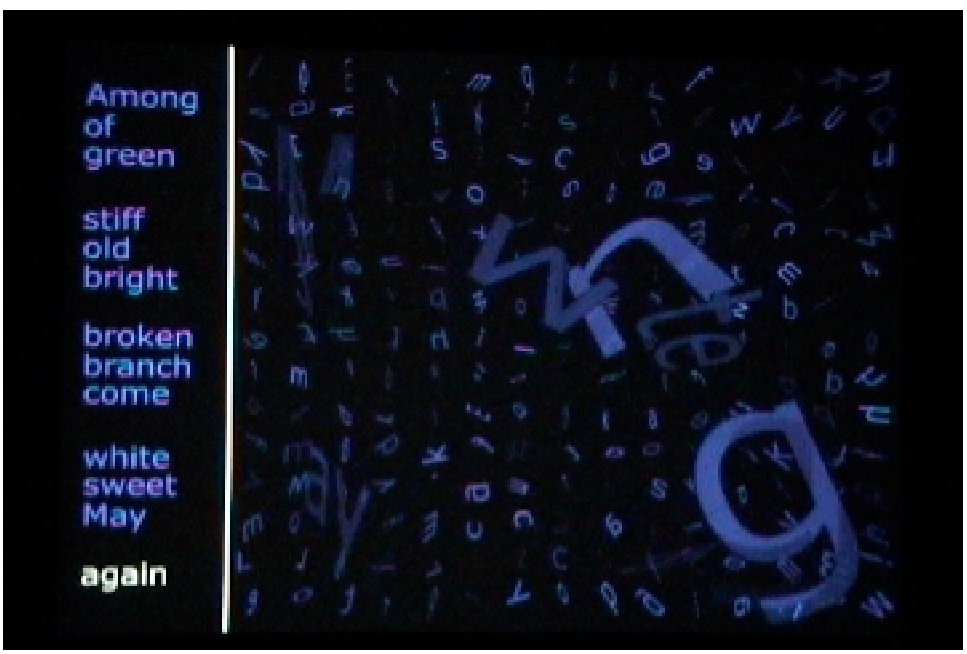
Jason Freeman

The Locust Tree In Flower

An Interactive, Multimedia Installation

Based on a Text by William Carlos Williams



INTRODUCTION

The Locust Tree in Flower is an interactive multimedia installation by composer Jason Freeman, based on a poem by William Carlos Williams. The installation invites a single person at a time to create and perform a musical setting of the poem by simply reading it. A short piece of music is generated in real time by applying digital processing, mixing, and looping to the user's voice.

The installation makes use of **new synthesis algorithms** which were developed by the composer. These algorithms use techniques adapted from a field of mathematics known as **cellular automata**. While the overall form and structure of the piece is pre-determined, all of the sounds themselves are produced in real time, and they vary according to the pitch, timbre, and inflection of the user's voice.

A common audio processing technique called **phase vocoding** is also used in the installation to repeat highly compressed copies of the sounds a user generated earlier in the piece. These **compressed repetitions** highlight phonetic relationships within the text.

The Locust Tree in Flower is intended to give people a meaningful role in creating and performing a short piece of music, whether or not they have a formal musical education. It also highlights the richness and diversity of the human voice.

The installation was premiered on July 14, 2000, at an event jointly presented by the **Computer Music Center at Columbia University** and the **Lincoln Center Festival**.

The following pages explore aspects of the installation in depth, including the text, synthesis algorithms, and user experience.

Text

The Locust Tree in Flower is based on a poem of the same title by William Carlos Williams (1883-1963). The poem was originally published in his collection *An Early Martyr* (1935).

The Locust Tree in Flower

Among
of
green

stiff
old
bright

broken
branch
come

white
sweet
May

again

Non-exclusive rights to use and reprint the text in the U.S. and Canada have been obtained by the composer from New Directions Publishing Corporation.

User Experience

The Locust Tree in Flower is used by a single person at a time; the entire user experience lasts approximately five minutes.

The user walks up to a microphone. A sensor mat detects that someone is ready to interact and displays brief instructions to the user on the video screen. The screen then splits into two parts:

- The left side displays the text of the poem. Each new word is highlighted in white.
- The right side displays fragments from this highlighted word, usually one or two letters at a time.

As the word fragments appear on the right, the user speaks them into the microphone. The installation then processes the voice in real time to create all of the music. In addition, background images are drawn on the screen based on the computer's analysis of the sound it produces.

Users waiting to interact may listen to the music on one of the extra sets of headphones.

Cellular Automata Synthesis

The Locust Tree in Flower processes users' voices using new digital signal processing techniques, developed by the composer, which are based on cellular automata, a field of mathematics which draws on ideas such as chaos theory, artificial intelligence, and artificial life.

About Cellular Automata

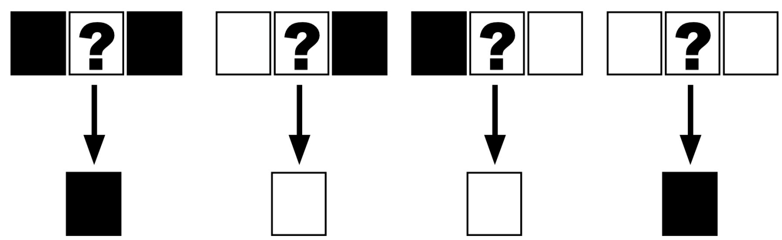
A cellular automata (CA) is a collection of evolving cells which change in each generation according to pre-determined rules.

The best way to understand CAs is to look at a simple example:

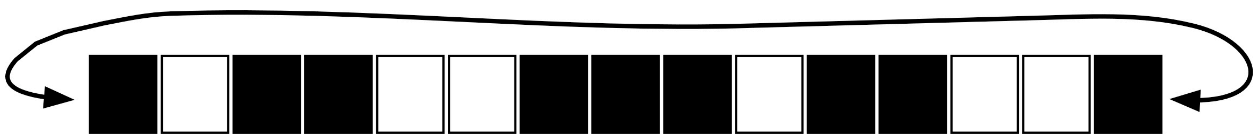
- Assume we have a **row of n cells**, each of which is either **dead** (black) or **alive** (white).

- For each cell, we flip a coin to decide if it starts out dead or alive.

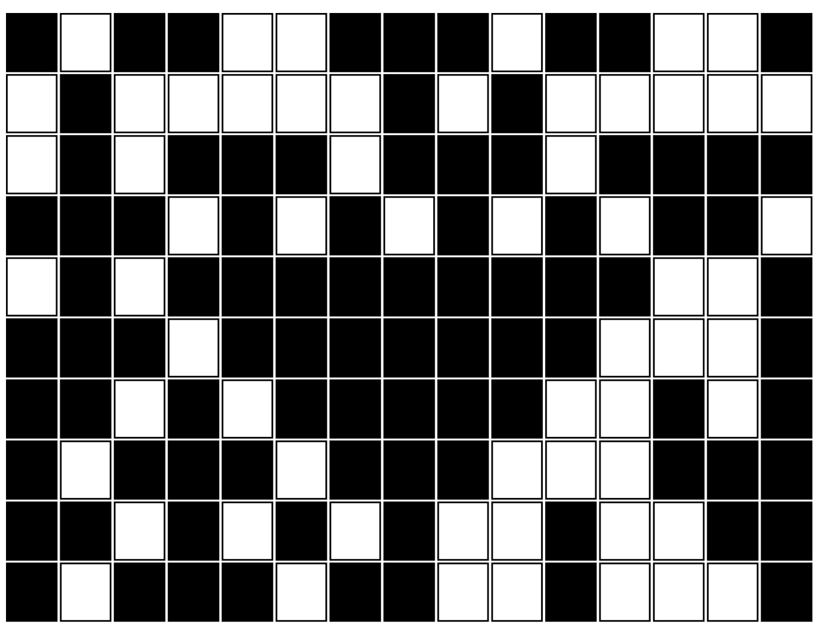
- We also set up **transition rules** which determine whether each cell will live or die in the next generation. We will consider each cell one at a time, and we will decide its fate based on the states of the **neighbor cells** to its immediate left and right:
  - If exactly one of its neighbor cells is alive, then the cell will be alive in the next generation.
  - If none or both of its neighbor cells are alive, then the cell will be dead in the next generation.



- Since the first cell in our row doesn't have a left neighbor and the last cell doesn't have a right neighbor, we need to define special **boundary conditions**. We'll simply turn our row of cells into a loop; the first cell's left neighbor becomes the last cell and the last cell's right neighbor becomes the first cell.



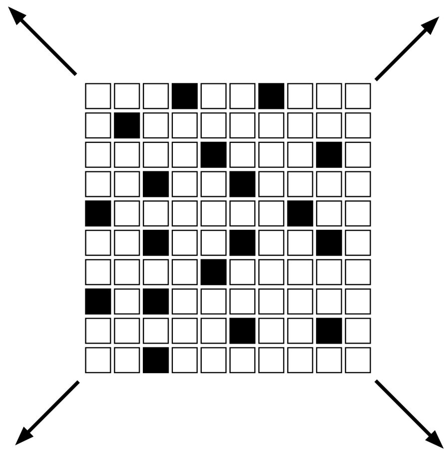
These very simple rules can create complex and unexpected results. Here is visual representation of the first ten generations of a CA based on this example:



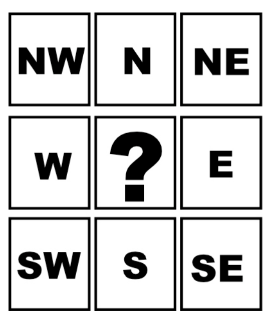
About the Game of Life

The best-known example of a CA is John Conway's Game of Life. It works like this:

- Instead of a single row of cells, there is a **two-dimensional grid of cells** that extends infinitely in all directions.



- The transition rules consider each cell and look at its **eight neighbors** (N, NE, E, SE, S, SW, W, NW):
  - If exactly three of its neighbors are alive, then the cell will be alive in the next generation.
  - If exactly two of its neighbors are alive and it is also alive, then the cell will be alive in the next generation.
  - Otherwise, the cell will be dead in the next generation.

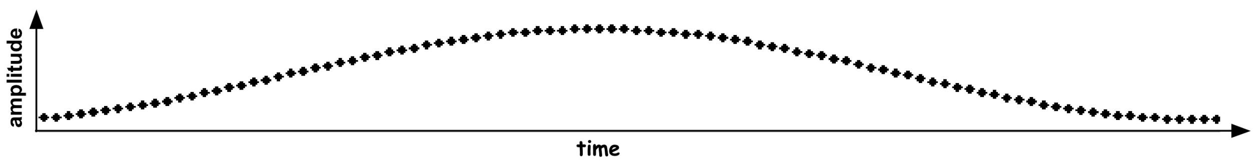


The Game of Life generates even more surprising and complex results than one-dimensional CAs.

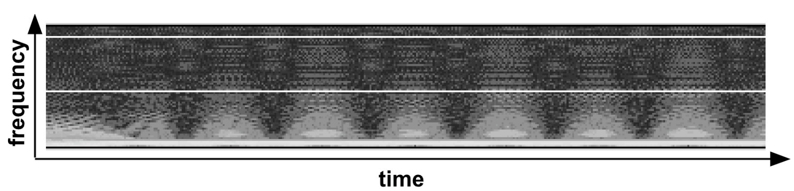
Key Digital Audio Concepts

Before we apply the ideas of CAs to processing sounds, it is important to understand some key ideas of digital audio:

1) **Digital audio consists of discrete samples**. While an analog device (i.e. a record player) stores sound continuously (i.e. grooves on a record), a digital device (i.e. a computer or CD player) stores sound as a set of discrete samples. These samples are so close together (usually 44,100 samples per second) that we hear them as a continuous sound. But digital sound is really just a one-dimensional grid of numbers.



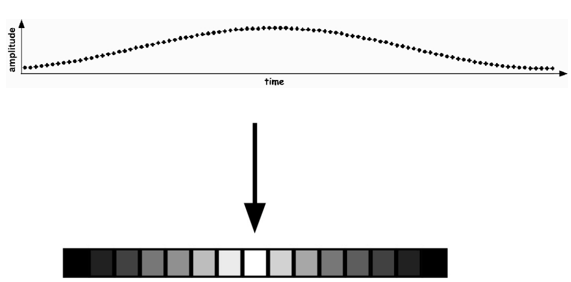
2) **Sound can also be represented on a grid of frequency and time**. A technique called the **Fast Fourier Transform** enables computers to represent sound as the intensity of different frequencies (i.e. pitches) as they change over time. This data is often represented on a two-dimensional grid, where the horizontal axis is time, the vertical axis is frequency, and the color of the plotted data is the intensity.



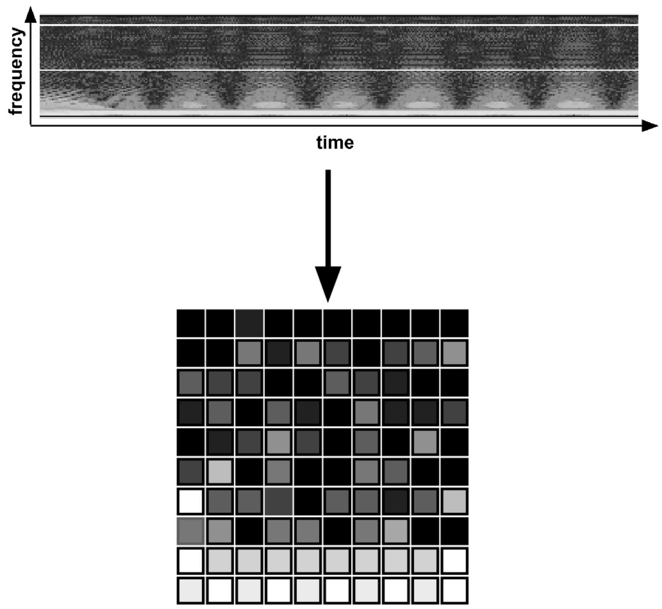
Cellular Automata Synthesis Algorithms

In the installation, the user's voice is processed in three ways:

1) **One-dimensional waveform**: A short sound is repeated over and over again. The discrete samples which make up the sound become the initial configuration of a one-dimensional CA, and each repetition of the sound becomes a new generation. The system is similar to our example of a one-dimensional CA, except that the cells can take on many different values representing degrees of life and death.



2) **Two-dimensional Game of Life**: A short sound is repeated over and over again. The two-dimensional frequency/time grid of the sound becomes the initial configuration of a Game of Life. Again, each cell can take on many different values (i.e. frequency intensities) which represent various degrees of life and death.



3) **One-dimensional granular transformation**: This algorithm processes a continuous stream of sound; in this installation, it always processes the sound produced by one of the previous two algorithms. The sound is divided into chunks of equal length, and these chunks become the initial configuration of a one-dimensional CA. As the CA evolves, it controls the relative volumes of subsequent chunks of audio.

Throughout *The Locust Tree in Flower,* these algorithms are used with different parameters and in different combinations to process sound in a variety of ways. And since small changes in the initial configurations of cells cause drastic changes in the outcomes of the algorithms, the timbre, frequency, and inflection of the user's voice has a profound influence on the evolution of the sound.

ABOUT PHASE VOCODING

A phase vocoder is a powerful tool for transforming digital sound based on its frequency content over time. Two of the most basic (and popular) applications of phase vocoding are to change the pitch of a sound without changing its speed and to change the speed of a sound without changing its pitch.

Normally, speeding up a sound makes its pitch correspondingly higher, a result known as “munchkinization.” (Think, for instance, of playing a 33 RPM record at 45 rpm.) Phase vocoding enables these two domains to be transformed independently of the other.

The Locust Tree in Flower uses phase vocoding in order to highlight phonetic relationships in the poem. As a user generates music, the computer records it. Then, whenever the user says a sound which has already occurred in the poem (i.e. the ‘o’ in ‘of’ is a repetition of the ‘o’ in 'among'), the phase vocoder repeats all of the sound it recorded between those two places in the poem. The playback of these sounds is sped up significantly, sometimes by as much as a factor of 30.