Structure & Simulation: Growth and Evolution in Generative Music

Dane Filipczak

Follow this and additional works at: https://dsc.duq.edu/etd

Part of the Composition Commons

Recommended Citation

This Worldwide Access is brought to you for free and open access by Duquesne Scholarship Collection. It has been accepted for inclusion in Electronic Theses and Dissertations by an authorized administrator of Duquesne Scholarship Collection. For more information, please contact phillipsg@duq.edu.
STRUCTURE & SIMULATION:
GROWTH AND EVOLUTION IN GENERATIVE MUSIC

A Thesis
Submitted to the Mary Pappert School of Music

Duquesne University

In partial fulfillment of the requirements for
the degree of Master of Music

By
Dane Filipczak

May 2017
Structure & Simulation: Growth and Evolution in Generative Music (2017) by Dane Filipczak is licensed under a Creative Commons Attribution 4.0 International License and is available from http://ddc.duq.edu/etd/116.
STRUCTURE & SIMULATION:
GROWTH AND EVOLUTION IN GENERATIVE MUSIC

By

Dane Filipczak

Approved April 6th, 2017

Lynn Emberg Purse, M.M.
Associate Professor of Music Technology,
Area Coordinator of Composition
(Committee Chair)

William Purse, M.M.
Professor of Music Technology & Guitar,
Chair of Contemporary Music Media and Jazz
(Committee Member)

Judith Bowman, Ph.D.
Professor of Music Education & Music Technology
(Committee Member)

Seth Beckman, D.A.
Dean, Mary Pappert School of Music

Stephen Benham, Ph.D.
Director of Graduate Studies
Associate Dean, Mary Pappert School of Music
ABSTRACT

STRUCTURE & SIMULATION:
GROWTH AND EVOLUTION IN GENERATIVE MUSIC

By
Dane Filipczak
May 2017

Thesis supervised by Lynn Emberg Purse, M.M.

This thesis describes a series of original interactive computer music compositions that are structured around simulations of natural phenomena. The relationship between simulation and music composition is considered and found to be a likely site of high creative potential due to both domains’ concern with process and becoming as opposed to static artifacts. The medium of the web application is considered as an ideal site for contemporary musical experimentation and a portrait is given of the current affordances and constraints of browser-based applications. Five musical systems are described within the context of their simulation and design techniques and some reflections are made to suggest further lines of experimentation.
DEDICATION

To Charles Darwin
ACKNOWLEDGEMENT

This work is conceptually indebted to many individuals, but would not have been possible without the complementary influences of Neri Oxman and Manuel DeLanda.

I’d also like to thank...

Ang Urban, for their encouraging questions and instruction in the lightness of being.

Djordje Ivkovic, for his sanity and stabilizing companionship.

and Adam Ziel, for getting and keeping me interested in doing strange things with software.
TABLE OF CONTENTS

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Abstract</td>
<td>iv</td>
</tr>
<tr>
<td>Dedication</td>
<td>v</td>
</tr>
<tr>
<td>Acknowledgement</td>
<td>vi</td>
</tr>
<tr>
<td>List of Figures</td>
<td>viii</td>
</tr>
<tr>
<td>Music and Simulation</td>
<td>1</td>
</tr>
<tr>
<td>The Internet and Virtual Galleries</td>
<td>1</td>
</tr>
<tr>
<td>I. Evolution and Adaption</td>
<td>4</td>
</tr>
<tr>
<td>Ia. Earthworm</td>
<td>9</td>
</tr>
<tr>
<td>Ib. Passerine</td>
<td>11</td>
</tr>
<tr>
<td>II. Forces and Growth</td>
<td>13</td>
</tr>
<tr>
<td>Ila. Caasi</td>
<td>13</td>
</tr>
<tr>
<td>IIb. Cyerce</td>
<td>15</td>
</tr>
<tr>
<td>IIc. Circadian</td>
<td>19</td>
</tr>
<tr>
<td>Reflections</td>
<td>23</td>
</tr>
<tr>
<td>Works Cited</td>
<td>28</td>
</tr>
</tbody>
</table>
# LIST OF FIGURES

<p>| | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>1.</td>
<td>Earthworm ................................................................. 9</td>
</tr>
<tr>
<td>2.</td>
<td>Passerine ......................................................................... 12</td>
</tr>
<tr>
<td>3.</td>
<td>Caasi ............................................................................... 14</td>
</tr>
<tr>
<td>4.</td>
<td>Cyerce ........................................................................... 16</td>
</tr>
<tr>
<td>5.</td>
<td>Circadian ......................................................................... 21</td>
</tr>
</tbody>
</table>
Music and Simulation

This project is a series of generative music compositions, in each of which real-time computer simulations inform various structures. A simulation is the computation of a mathematical model over time and is an essentially dynamic process. Music as well is essentially dynamic and concerned with development and change through time. I became interested in computer simulations of natural phenomena because they allowed me to explore, pervert and modify natural beauty. While learning to program I was struck by the extent to which the processes I was expressing in code were similar to musical processes. This project hopes to expose and exploit this similarity.

After some introduction and discussion of relevant concepts, five generative compositions are described under two classes. The first class in concerned with simulations of evolution and contains the pieces Earthworm and Passerine. The second class is concerned with simulations of forces and growth and contains the pieces Caasi, Cyerce and Circadian. All five pieces can be accessed from the project url dane.audio/sands or directly from dane.audio/name_of_piece.

The Internet and Virtual Galleries

I chose to implement this project as a series of web based applications. There are several key differences between a computer program running locally on a desktop or mobile device and a program running in a web browser. Both the native and web paradigms offer different affordances and constraints, which are worth briefly detailing.

A web application can be differentiated from a native application by the location of its source code. The code and supporting media required to run a native application are
stored in memory of the device in which they’ve been installed, while the code and media required for a web application are accessed only upon navigation to their hosting webpage. After materials have been accessed they may be stored in a temporary cache within the browser. This avoids having to repeatedly access identical materials for often-visited web locations, which becomes significant when large media files are in use.

A crucial distinction between the source code of native and web applications is that native applications are written using device-specific protocols, that is, techniques that are specific to the architecture or operating system of the device they are intended for. A native application written to run on an Android tablet will not run on a Windows PC because the difference in operating system and circuit architecture between the two devices requires that the programmer choose tools, programming languages and techniques that respect the affordances and constraints of the device. In contrast, the technologies of the Internet are intended to be device-agnostic. JavaScript is the single programming language that is used for client-side interaction across all web browsers, and a program written in JavaScript will, with few exceptions, be able to function identically across different devices. In addition and complementary to a common programming language, recent browsers are equipped with standardized technologies for working with rich media. Three of these recent technologies that have been used extensively in this work, given in order of descending maturity, are:

- The HTML5 `<canvas>` element, which allows two-dimensional pixel-based drawing and animation.
- the Web Audio API, which supplies tools for audio analysis, synthesis, and processing.
- WebGL, which allows the browser to communicate with a device’s Graphical Processing Unit, a specialized piece of hardware that affords the speed necessary to render three-dimensional graphics in real time.

The emergence of these and other standardized web technologies means that the browser is becoming ever more capable of providing an interface to rich applications and experiences rather than the simple information it was initially intended to circulate. The youth of these technologies communicates that this work has only recently become possible in an Internet medium.

Despite innovations in web technology and the future promise of this area, there are still many good reasons to write a native application. Because native applications are designed with the specificities of the device, the process of accessing device-specific hardware like sensors and displays is relatively streamlined. Most importantly for many programmers, there is a difference in performance speed between native and browser code that is sometimes vast. This performance difference is due to programming techniques possible in native code that optimize the efficient allocation of device memory and resources. In practice, this means that there is a very tangible limit on the processing power required of programs that can be reasonably executed in a web browser. Several times during this project I’ve had to scale down or design around performance limits imposed by the browser medium.

The development of diverse musical practices depends on diverse environments. Byrne (2007) illustrates how different material contexts provide the affordances and constraints that allow heterogeneous styles to emerge. While the web browser has recently become a primary site of musical encounter, a distinct genre hasn’t yet evolved
to fill the niche offered by the medium. This project is, in part, an attempt to adequately formulate the question: what could the music created in Internet space look like?

I. Evolution and Adaption

The first two pieces to describe are based on simulations of biological evolution. One broad conception of musical form is that it is a series of techniques to bridge the gap between the two extremes of unrelieved repetition and unrelieved difference (Scholes, 1977). In this understanding, the most fundamental musical structure is the theme and variations and the most fundamental musical gesture is development. The proliferation of life by evolution follows a similar process; small differences are developed and refined over time amidst a vast sea of repetitions.

I first became interested in programming computers as a result of exposure to the concept of evolutionary computation, which is a class of algorithms that model the process of biological evolution in pursuit of a variety of ends. The first section of this thesis describes compositional systems designed according to these models. To understand how evolutionary computation works, let’s first review the basics of biological evolution. A necessary requisite is an information structure, DNA, which encodes the instructions for developing an organism. In nature, this development isn’t strictly deterministic but relies on many contingencies (affordances and constraints) of the environment in which development takes place. As a result, no two organisms are exactly alike even if they are clones of each other and thus share the same DNA (as an identical twin, I’m personally invested in affirming this wonderful fact). In contrast, computer simulations of development are abstractions that usually disregard the aleatoric
effects of environmental influence so that two identical DNA encodings necessarily develop into identical artifacts.

DNA is the informational apparatus by which traits get passed from one generation to the next. Sometimes DNA is passed directly to the next generation in asexual reproduction. However, we mammals are acquainted with a more complicated approach wherein DNA arrives in the next generation as a result of the ‘shuffling’ or crossover of the DNA of two individuals, called sexual reproduction. Either un-shuffled asexual DNA or shuffled sexual DNA usually arrives in the next generation intact, without any errors. Very rarely, however, mistakes are made in the copying process resulting in novel encodings that didn’t occur at all in the previous generation. These are called mutations and most are either neutral to the organism, having no affect on its life process, or harmful, causing it to die before it can reproduce and pass the mutation to the next generation. A small fraction of mutations, however, are beneficial to the organism and allow it to reproduce more than it would have otherwise; these benevolent mutations can quickly spread throughout a population and become permanent features of a species.

When a mutation increases the rate at which an organism reproduces, we say that it increases its fitness. In nature, fitness is determined in one of two ways. The first is natural selection, wherein an organism’s navigation of environmental pressures such as nutrient availability and predators determines how long and how robustly it will survive and reproduce. The second is sexual selection, wherein the rate at which an organism reproduces depends on its success at convincing a member of the opposite sex that it is a suitable partner for mating. Both sexual selection and natural selection can be facilitated or hindered by random mutations in DNA.
Thus biological evolution follows a simple recursive procedure:

1. Replicate with variation DNA with greatest fitness.
2. Evaluate fitness of variants.
3. Repeat.

The extraordinary simplicity and elegance of this procedure make it easily replicable in software. Echoing the steps just listed, the general process of an evolutionary program proceeds as follows. First, create a population of virtual DNA with the ability to replicate. Next, devise a method to test the DNA for its fitness. Finally, create a new population that passes only the fittest DNA into the next generation, allowing for it to mutate slightly. Over time, the system will tend to find DNA that achieves higher and higher fitness scores.

While traditional evolutionary algorithms are usually deployed upon well-defined problems with easily quantifiable solutions, I’m interested in their application to ill-defined aesthetic goals and the potential of this application to spawn variations of materials that are applicable to musical form as outlined above. There is an analogy between judging fitness by aesthetic goals and natural selection; both approaches lack a sense of teleology or of a clearly perceptible goal. Biological evolution isn’t aiming to achieve greater complexity or perfection but is rather blindly searching a space of possibilities in complete ignorance. Moreover, the filter that determines which possibilities are viable is constantly changing. Music composition is reminiscent of this purposeless development within fuzzy boundaries. I’ve developed two systems that apply evolutionary computation to musical materials in unique ways that are intended to pursue aesthetic instead of technical solutions. Before detailing these systems, it will be helpful
to consider in more depth the process of bio-digital development by which digital representations of DNA become perceivable artifacts.

**Parametric Design and Feature Space**

Evolutionary programming requires the tools of a discipline that is usually known as *parametric design* when encountered separately from evolutionary processes. This approach requires that every detail of an artifact be specified by a mutable parameter. The task of the designer isn’t to specify the exact form of the artifact, but rather to determine the relevant parameters of the artifact and establish plausible ranges within which interesting variation can be achieved within the constraints of the design goals. In programming applications, the list of parameters that determine an artifact is commonly represented as a series of floating-point numbers from 0 to 1, but this is an arbitrary convention based on ease of manipulation. Many more complex methods of encoding artifact features are possible.

Consider a simple parametric design system with three parameters, each of which is an integer from 0 to 255 that corresponds to the amount of red, blue, or green present in the shading of a cube. Now visualize the cube in three dimensions with an edge length of 255 where each of the three spatial dimensions (width, depth, and height) corresponds to the range of either the red, green, or blue parameter. Any possible set of values for our three parameters can be represented as a unique point within this cube, and the set of all points lying within the geometry of the cube represent all possible colors our system is capable of producing. Points that are nearby each other represent similar colors with
similar parameter values, and points very far from each other represent very different parameters and colors.

What we’ve just described is a three-dimensional feature space corresponding to parametric system with three variables. As systems become more complicated, their amount of variables increases along with the dimensionality of its feature space. Humans aren’t very good at visualizing higher dimensional spaces but have nevertheless been describing them with mathematics and relying upon them in scientific theories for centuries (DeLanda 2002). Any potential artifact produced by a parametric system is represented by a single point in $n$-dimensional space where $n$ is the number of parameters determined by the designer. One of the great joys of working parametrically is that, even though the designer is aware of the potential range of each parameter, there are almost certainly possibilities latent in the feature space that are foreign to the creator’s imagination. The challenge of designing such a space is determining which are the relevant features of the artifact to be parameterized, after which the range of potential outcomes can be explored.

A variety of navigational techniques can be implemented in order to discover what forms lie in various regions. One approach is to move randomly through feature space, either through Brownian motion (steps in random directions) or a more subtle process like Perlin or Simplex noise. Another approach is to develop a criterion to gauge the fitness of each point in feature space and move purposefully in the direction of increased fitness. The general term for such a purposeful movement is optimization, and evolutionary computation can be understood as one possible method of optimizing locomotion through high-dimensional feature space.
Ia. Earthworm

Titled and styled after the creature that was supposedly Darwin’s favorite, Earthworm is an evolutionary system wherein the fitness of the artifact is determined by interactive selection. It’s a tool that is designed to assist with the composition of melodies by a human artificer - a creativity pump that catalyzes effective melodic writing. The primary goal of Earthworm was to develop a method of parametrically encoding musical phrases that could generate plausible melodies within a modal/diatonic idiom. A secondary goal was to craft an interface intuitive enough to be understood and used effectively by individuals of all ages and abilities.

![Earthworm Diagram](image)

**Figure 1. Earthworm**

Horizontal worm shapes represent melodies. Placing the mouse over a melody plays it according to the tempo set by a slider on the control panel at the right. Other sliders set the population size, mutation rate, and note length of the melodies. Clicking on a melody changes its color and adds it to the mating pool. Clicking on a portrait of
Darwin breeds a new generation, either sexually (with crossover) or asexually (without crossover), depending on user settings. Melodies can be bred chromatically, diatonically, or pentatonically. The diatonic setting can yield any of the seven church modes except locrian. Locrian was excluded from the model because of its rarity and confliction with the built-in perfect fifth drone accompaniment, a feature implemented in order to easily establish key centricity.

In my experience, the craft of writing melody is rarely quick and frictionless. More often, it involves rapid iteration and incremental modification, much of which isn’t consciously deliberated in the composer’s mind. A system like Earthworm allows the user to tease out form by way of intuition in a similar manner to how I perceive the workings of my ‘natural’ composition process, a sort of groping in the dark that’s nevertheless directional. I’ve found the tool to be particularly effective when integrated into more traditional practice. For example, one can sit at the keyboard with the application open and breed a few generations until a suitable tune fragment emerges, upon which the fragment can be iterated and adapted within the keyboard. I have found that my practice as a composer has helped me develop an acute ability to modify, adapt, and synthesize melodic fragments. However, the raw melodic materials do not always immediately present themselves. Certainly they sometimes do (inspiration isn’t always a fantasy), but on some occasions my mind isn’t robustly able to produce raw melodic material on which to procedurally transform. On these occasions I’ve found Earthworm to be helpful because it allows me to use intuitive cognitive processes in order to obtain the raw seed on which to work.
**Ib. Passerine**

Passerine was my second attempt at applying genetic programming for musical ends. DNA in Passerine encodes the instructions to procedurally generate the bodies of virtual creatures that exist within a simulated physical environment. Music is generated indirectly from the genetic sequence as a result of collisions of those creatures with stationary objects in their environment. Different creatures have different extensive properties like number and length of limbs, body layout, and overall size but also different intensive properties like friction and restitution that can vary across the limbs of the creature. In addition, some joints can act as motors, spinning in a certain direction with a certain torque. The system is capable of generating diverse creatures with very different locomotion patterns.

While in Earthworm the fittest specimen of each generation is determined by user intervention, in Passerine each creature has a virtual metabolism that is capable of killing the creature when it runs out of energy. Energy is acquired by colliding with note objects in a certain order determined by user input. Imagine a sequence of abstract scale degrees [5, 3, 4]. A collision with 4 immediately after spawning will have no effect on a creature’s overall energy, while a collision with 5 and subsequent collision with 3 will each increase a creature’s energy by a set amount. It’s only by colliding with notes in a certain order that a creature can keep its energy levels high enough to avoid death. Most individuals will run out of energy quickly, but their genes will endow some with the capacity to collide with notes in the order required to remain alive. The individual who remains alive the longest is the fittest individual of its generation and becomes the sole parent of every individual in the subsequent generation. In this case, reproduction is
always asexual, that is, never involves crossover between two heterogeneous genetic sequences but rather always proceeds by mutation of a single sequence.

Figure 2. Passerine

The musical texture of Passerine is primarily polyphonic, with each individual producing a monophonic series of pitches. Differentiation between voices is achieved through varying orchestration and articulation. The ADSR envelope for each voice is randomized in each generation to provide subtle variation to the resulting orchestra. Each voice is bussed to a master reverberation node, the parameters of which vary slightly with each generation in order to contribute to the effect of gradually shifting environment. The only locus of interaction for the user is the establishment of the initial sequence of notes that the population is evolving towards; all subsequent variation and musical interested in generated purely through the evolutionary process as the population becomes more and more adapt at playing the sequence of notes.

While Passerine relies upon a simulated physics-world in order to enable the embodiment of a set of virtual actors, the physical forces at play in its world are static and
don’t contribute directly to the musical outcome. The next set of projects demonstrates applications of forces that are more dynamic and integral to the musical events.

II. Forces and Growth

The following three pieces are based on simulations of physical forces and biological development, between which there is a complicated relationship. The most ubiquitous application of simulation in consumer products such as video games and graphic animation is the simulation of physical forces. Gravity, air resistance, friction, and other factors can be implemented in order to create a sense of being within an embodied environment. Simulated forces can be either accurate to those present in the terrestrial milieu or else entirely surreal. The next set of projects utilizes simulated forces in the service of musical ends. Forces serve in the following as either transcendent environmental containers or immanent interactions among discreet agents.

IIa. Caasi

Caasi, its title an anagram of Isaac (Newton), is an attempt to sonify variations in global physical forces. The user clicks the screen to deploy particles of random masses, each of which is immediately caught within a simulated physics world. By holding down a key and swiping the mouse the user can generate force vectors. Sliders control the global force of gravity and extent of air resistance. Tones are cued when particles collide with the edges of the screen.
Ontologically, Caasi lies somewhere between an instrument and piece of music. It requires continuous user input in order to make any sound at all. However, the pitch content of the sounds it makes when prompted are rigidly pre-determined and organized according to a specific aesthetic goal. On the other hand, its instrumentation is flexible within certain limits and left up to the user. I think the work is most successful as a conceptual tool for developing an intuition of what the atmospheric forces on heterogeneous planets might feel like. After experimenting with the program one develops a sense of ‘low-gravity sound’ versus ‘high-gravity sound’ that, while quantifiable, is difficult to express in words. Additionally, Caasi is a promising tool for physics education because it renders the abstract concepts of gravitational force and wind resistance tangible.

The forces in Caasi are all calculated according to global, environmental variables. This treatment of forces as top-down, hierarchical constraints is analogous to their function in classical dynamics and most other areas of pre-20th century physics. The
subsequent examples build on the concept of force-governed movement but calculate their forces according to immanent interactions among constituent parts of the system. This bottom-up approach is analogous to \textit{biological} dynamism, wherein development of a system proceeds according to internal rather than external constraints.

\textbf{IIb. Cyerce}

Cyerce, titled after the sea slug \textit{cyerce nigricans}, was my first exploration of a system animated by bottom-up, immanent forces. I was introduced to this concept by an article by the Norwegian generative artist Inconvergent, who describes an algorithm for generating two-dimensional patterns that’s inspired by the growth of organisms like coral, flowers and cabbage. The process begins with a collection of autonomous nodes arranged in a ring. Each node is attracted to its two neighbors in the ring while also being repelled from every node in the ring. Whenever two connected nodes get beyond a certain distance apart from each other, the edge between them splits and a new node appears halfway between them. These three simple rules in combination lead to interesting emergent patterns that morph over time in a biomimetic manner. This style of digital morphogenesis can be termed \textit{differential growth} because the form is determined by different rates of growth happening simultaneously in various parts of the simulated organism.

The differential growth algorithm described above seemed to be an ideal starting place for an investigation of musical form governed by bottom-up forces because of its relative simplicity and limited spatial dimensions. I was not able to locate many examples of similar work, nor any projects that intuitively conveyed the dynamic nature of the system over time through interaction. Inconvergent’s work, while interesting and perhaps
pioneering, is limited to static representations of a final product – the digital counterpart of a biological corpse. Tracking the development of the shape over time would allow for its dynamism to be harnessed in service of a musical process.

![Image of Cyerce](image)

**Figure 4. Cyerce**

The musical structure of Cyerce is based on two overlapping rhythmic cycles that create polyrhythm. The first cycle is independent of the simulated organism, while the second is dependent on the temporal and spatial structure of the organism. The second cycle gets cues to sound a tone when an edge splits and a new node appears. This tone does not sound immediately, but is quantified to the rhythmic grid described below. The pitch of the tone is determined by the location of the node within Cartesian space where the origin (0, 0) is the center of the organism. The absolute value of the x-axis determines the pitch while the absolute value of the y-axis determines the instrumental timbre. Each organism is assigned a random subset of four instruments out of eight possibilities in order to keep the orchestration surprising as different organisms arise and pass away.
**Bjorklund Rhythms**

The rhythmic cycles in Cyerce are determined through a generative technique known as alternatively as either Bjorklund or Euclidean rhythms. The defining characteristic of a Bjorklund rhythm is that its attacks $k$ are distributed as evenly as possible among metric divisions $m>k$. Toussaint (2004) showed that algorithms described by Euclid circa b.c.e 300 and Bjorklund circa 2003 are structurally identical; however, Bjorklund’s method results in binary lists of symbols that are useful for representing either notes and rests or accented and unaccented notes. For Cyerce, I implemented a function $bjork(k, m)$ that evenly distributes $k$ attacks among $m$ divisions and returns a list of zeros and ones that is then used to schedule rhythmic events. A few examples should help to clarify this.

\[
\begin{align*}
bjork(4, 8) & = [1, 0, 1, 0, 1, 0, 1, 0] \\
bjork(3, 8) & = [1, 0, 0, 1, 0, 0, 1, 0] \\
bjork(4, 7) & = [1, 0, 1, 0, 1, 0, 1] 
\end{align*}
\]

Cyerce utilizes a single temporal cycle that is divided at any given instant in two distinct manners by the results of two Bjorklund functions being fed random values for $k$ and $m$. To achieve limited rhythmic consistency over time, $m$ changes less frequently than $k$ and is restricted to integers less than 12. What results is a generative cascade of polyrhythms that vary in rhythmic dissonance. Two patterns with $m$ values of a relatively consonant ratio, such as 9:3, can produce tightly interlocking parts. Patterns with relatively dissonant ratios between their $m$ values such as 11:5 produce more rhythmically dense and jarring effects that retain metric coherency because of their shared cycle length. One
advantage of automating polymeter in this way is that it had led me to some very strange
and profound combinatorial possibilities that I wouldn’t have explored otherwise.

**Immanent Harmonic Transformations I**

There are many techniques for procedurally generating harmonic progressions. Most rely on a top-down approach wherein tones are organized as members of a hierarchical key. In Cyerce, I wanted to handle harmonic progressions in a manner that would reflect the immanent, bottom-up approach consistent with the project. I drew upon contemporary neo-Riemannian music theory to achieve this goal. The neo-Riemannian body of work seeks to understand harmonic progressions by relating harmonies directly to each other, without reference to an overarching tonic or scale. A transformational account of harmony is articulated wherein the relatedness of triads is judged in accordance with either the commonality of their tones or the smoothness of the voice leading required to transform each to the other. One of the goals of these theorists is to conceptualize harmonic function in the music of the late Romantic period, when common practice notions of key and scalar hierarchy began to be challenged. Cohn (2012) claims that the transformational conception of harmony was indigenous to the way that nineteenth-century composers approached their craft but has since been “stripped from music theory’s inheritance.” (ix) The bottom-up approach can be difficult to grasp because top-down diatonic hierarchy has been institutionalized in the names of notes, their position on the staff, and the system of key signatures (Cohn 8 2012). However, such an approach is valuable for generative music because it can be formalized
algorithmically to produce tonal harmonic progressions that don’t rely on a strict key or hierarchy.

The transformational approach was implemented in Cyerce by emphasizing common tone relationships. Only the four basic triad qualities (major, minor, diminished and augmented) were utilized. The algorithm proceeds in four steps:

a. Select randomly a tone of the current triad.

b. Select randomly the structural role (root, third or fifth) that this tone will play in the next triad.

c. Select the quality of the next triad stochastically, based on the quality of the current triad and reducing the relative probability of a series of dissonant triads.

d. Calculate the pitches of the two remaining tones in the next triad given the common tone and its triadic quality.

This approach leads to a generative cascade of triads linked by common tones. To achieve a greater sense of tonality, triads transform no more than three times per ten-second span. This establishes a limited macro-harmony within any given expanse of time.

Cyerce can be interacted with by dragging the mouse across the screen to create new nodes. These nodes are immediately incorporated into the organism at the closest possible point and begin to be influenced by the same growth forces as the others. The ability to interact with a growth simulation in real time creates an experience of ‘growth painting’, or perhaps ‘audiovisual gardening’.
IIb. Circadian

Circadian is an extension and reinterpretation of the basic concept of Cyerce into three dimensions. This presented many additional challenges over the two dimensional version. Circadian retains the three basic rules of Cyerce:

1. A collection of interlinked nodes exert repulsion against every other node
2. Nodes are attracted to every node immediately connected to them.
3. If the edge between any two linked nodes lengthens beyond a certain threshold, a new node is inserted halfway between them.

The main difficulty with implementing these rules in a three-dimensional system is that nodes are situated within the two-dimensional surface of a three-dimensional shape as opposed to a one-dimensional edge of a two-dimensional shape. When an edge splits on a line, it’s simple to determine which nodes the newly created mode should be linked to because every node has only two links with its immediate neighbors in the line. On a two dimensional surface a node must be able to flexibly connect to other nodes while maintaining the mesh structure of the surface, and this is not a simple problem.
The transformation of harmonic material in Circadian proceeds in a manner opposite and complementary to Cyerce. While in Cyerce a single note of a triad is selected in order to serve as a common tone by which to pivot to the next triad, Circadian selects a single note to be altered while the two remaining pitches of the triad remain the same. Circadian adopts a more liberal scope of definition for the triad, allowing thirds to be either major, minor, or augmented and fifths to be either perfect, diminished or augmented. Thus, six types of triad are possible (presented with their abbreviations): major (M), minor (m), augmented (A), diminished (d), suspended fourth (sus4) and major with diminished fifth (Mb5). There are constraints applied to each type of triad that determine which notes may be selected to transform and by what amount, halftone (t) or tone (w). Not all possible transformations are utilized; some are discarded based on aesthetic preference. The following table details for each type of triad the chosen notes
which are capable of moving, in what direction, by what degree, and the type of triad that results from the movement (regardless of inversion).

Table 1. Triadic transformations in Circadian

<table>
<thead>
<tr>
<th>↑</th>
<th>M</th>
<th>m</th>
<th>d</th>
<th>A</th>
<th>sus4</th>
<th>Mb5</th>
</tr>
</thead>
<tbody>
<tr>
<td>1h→d</td>
<td>3w→sus4</td>
<td>5h→m</td>
<td>1h→m</td>
<td>1h→Mb5</td>
<td>5h→M</td>
<td></td>
</tr>
<tr>
<td>3h→sus4</td>
<td>5h→M</td>
<td>5w→M</td>
<td>3h→m</td>
<td>5h→m</td>
<td>5w→A</td>
<td></td>
</tr>
<tr>
<td>5h→A</td>
<td>5w→d</td>
<td>5h→m</td>
<td>5h→m</td>
<td>5w→M</td>
<td></td>
<td></td>
</tr>
<tr>
<td>5w→m</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>↓</th>
<th>1h→m</th>
<th>1h→A</th>
<th>1h→M</th>
<th>1h→M</th>
<th>3h→M</th>
<th>1h→sus4</th>
</tr>
</thead>
<tbody>
<tr>
<td>1w→d</td>
<td>1w→M</td>
<td>1w→m</td>
<td>3h→M</td>
<td>3w→m</td>
<td></td>
<td></td>
</tr>
<tr>
<td>5h→Mb5</td>
<td>5h→d</td>
<td>5h→M</td>
<td>5h→M</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>5w→M</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

One consequence of this system is that a triad is never able to transform into a triad of its same type, generating harmonic interest within a decidedly tonal vocabulary. An interesting observation is that the classically dissonant triads, diminished and augmented, are only able to transform into classically consonant triads.

Voicing

In Cyerce, the percussive nature of the sounds led me to disregard voicing of the chords because concurrent sounds were too brief for their spacing to be perceptually important. In Circadian, the sustained wall of sound required a more nuanced approach to the voicing of triads. I implemented a classic voice spacing plan that roughly corresponds to the structure of the acoustic overtone series, wherein lower tones are distributed farther
apart than higher tones and spacing becomes smaller as pitch rises. This classic voice
distribution arises as a consequence of the limited range of human hearing. Small
intervals in low notes are much more likely to produce dissonant clashes in the upper
overtones of the notes that are within the range of human hearing. Small intervals are
more tolerable in higher notes because dissonances in their upper overtones are likely to
occur at higher frequencies than are perceivable by humans.

**Reflections**

Each of these pieces represents an attempt to craft generative music from a
simulation of a natural phenomenon. In many cases, this led to interesting musical form
that was enhanced by visual corollary. Simulations are effective ways of producing form
over time through computation, and it has been demonstrated that this form can be
effectively mapped onto the musical milieu.

Historically, the composition of art music has been characterized by the
imposition of the intentions of composer. Borrowing a technical term from philosophy,
we can describe this concept of design as *hylomorphic* – consisting of an inert material
(tones), upon which a form is imposed. Using a spatial metaphor, the hylomorphic model
can be described as a top-down approach. In contrast, the generative music in this project
is characterized by a *materialist* concept of design, wherein form emerges from
immanent interactions among materials that are governed by their own diverse logics
independent of any transcendent hierarchy. The materialist model can be described as a
bottom-up approach.
Human designers have historically been most intimate with the top-down approach, usually to the extent of ignoring the existence of bottom-up design altogether. However, hylomorphic design is in fact a rarity on the Earth that only came into existence with the emergence of complex human minds with the capacity and desire to impose their artistic intensions on the outside world. All the beauty of the natural world is achieved by materialist, bottom-up design operating at different scales:

- Geological beauty emerges from the immanent logic of mineral systems.
- Biological beauty emerges from the immanent logic of genetic replicators.
- Celestial beauty emerges from the immanent logic of elements in space.

These examples have in common the absence of an intelligent designer who can impose their intention hylomorphically on the materials. They also have in common temporal scales incompatible with normal human perception. The process by which a mountain, species or galaxy comes to be formed is too slow to be perceivable under normal human conditions – the disembodied techniques of scientific empiricism are required for these design processes to become “visible”. In absence of this extended scientific vision, it’s quite understandable that earlier generations of humans assumed natural beauty to be the artifact of a hylomorphic designer – such was the only concept of design readily available to their perception. Computer simulations greatly behoove the student of materialist design because they allow it to be studied at temporal and spatial scales compatible with human perception.

Because hylomorphic human artists have been influenced by natural beauty throughout history, there has long been a complicated interaction between hylomorphic and materialist designs in human artistic practice. In music, this interaction came to be
explicitly problematized with the advent of aleatoric postmodernism. Aleatoric composers rejected hylomorphic intention completely, throwing their materials into a chaotic flux. Any beauty that emerges from aleatoric music is bottom-up, created through immanent materialistic interactions among tones. Randomness is also a necessary component to many forms of natural materialist design; however, it never the only component of the process. Postmodern composers rejected intention but failed to perceive that chaos isn’t the only alternative to hylomorphism, as shown by the above discussion of materialist design. The ambient music pioneered by Brian Eno develops aleatoric postmodernism by emphasizing the novel form that may arise from the aleatoric combination of given tone sequences, providing another realization of materialistic composition at a higher structural level. This project is a continuation of the materialist formal concept initiated by aleatoric and ambient music.

We’ve just identified two broad movements in art music history, the hylomorphic and the materialist. We should now ask whether one is more aesthetically ‘correct’ than the other in order to address the disdainful schism often perpetuated by practitioners of either camp. To stereotype the problem, materialists see the hylomorphists as narrow-minded while the hylomorphists see the materialists as anti-artistic. Recalling the above discussion of natural beauty can help to neutralize this dichotomy. Hylomorphic design is a distinctive capacity of the human mind; nevertheless, materialist design is so outstandingly competent that for millennia humans misattributed its natural manifestations to a hylomorphic designer. Materialist design is the norm of natural beauty, though its processes often fall outside the scope of human perception.

Hylomorphic design is a manifestation of materialist design specifically adapted to
human minds that often draws inspiration from the snapshots of materialistic processes available to the limited perceptual capacities of human bodies. When the postmodernists rejected hylomorphism outright, they were dissociating themselves from the embodied human experience, and this was an important step in the evolution of aesthetics. However, the failure of the postmodern attitude is that it is founded on a basis of social constructivism and therefore ignores the embodied perceptual capacities of the human experiencer. A more scientific project is to first question the assumptions of top-down human design by exploring bottom-up naturalistic alternatives while making sure to adapt the resulting forms to a scale suited to human perceptual capacities.

Materialistic music as explored in this project is a way of questioning hylomorphic tradition, of pushing it outside itself. It’s a necessary compliment to top-down composition in the same way a walk in the woods is a necessary compliment to a beautiful panting of a landscape. The arrangement of tones using bottom-up design has been less explored than analogous approaches in the visual arts for the simple reason that the very production of tones is almost always the result of an intentional act of a human mind – with few rare exceptions, stable tones suitable for combination with other tones are only produced by humans and their artifacts. Because tonal music is so correlated with humans, it’s not a surprise that their hylomorphic design strategies historically monopolized its production. The introduction of materialist design into tonal music shouldn’t replace hylomorphic design but augment and queer it. The two approaches, generative and static, should be able to coexist within a single work with the materialist understood as a more primal superset of the hylomorphic just as the animal is a superset of the human. The embodied return to human perceptual scales after flights to non-human
domains is the aesthetic movement that interests me the most. Deleuze and Guattari have offered a description of artistic practice copasetic with my vision of the interaction between generative and static approaches to composition:

The artist begins by looking around him- or herself… adopting an earthbound position, the artist turns his or her attention to the microscopic, to crystals, molecules, atoms, and particles, not for scientific conformity, but for movement, for nothing but immanent movement; the artist tells him- or herself that this world has had different aspects, will have still others, and that there are already others on other planets; finally, the artist opens up to the Cosmos in order to harness forces in a “work” (without which the opening onto the Cosmos would only be a reverie incapable of enlarging the limits of the earth). (1980, 372)

Combining bottom-up and top-down design concepts requires a medium capable both of generating novel form in real time through procedural means and displaying fixed pre-composed sequences. This is where the computer excels as a context for aesthetic experience; it’s uniquely able to catalyze opportunities to enlarge the limits of the earth by re-scaling nature to suit our perception. I look forward to creating music that combines these two concepts of design to create interesting and new experiences.
Works Cited


Toussaint, G. The Euclidean algorithm generates traditional musical rhythms.
