Eat Your Grains: Artificial Ecosystems, Granular Synthesis and Generative Music

By Gabe Webb

Thesis Advisors Edwin van der Heide, Rob Saunders

Abstract

Composers and scientists have used algorithms to create new pieces of music for centuries, and they have often been inspired by natural phenomena for modeling their work. The advancement of recording technology and the development of the digital computer in the mid-to-late 20th century allowed musicians, artists and researchers to apply statistical models derived from natural sciences to musical composition and manipulate sounds on timescales that were previously impossibly small.

In this work, we propose Audion, a multi-agent software system for granular synthesis and real-time musical exploration. First, we provide the historical, artistic and scientific contexts for current approaches in generative music, paying special attention to methods inspired by cellular automata, multi-agent systems and artificial ecosystems. We then discuss the development of Audion, drawing from design criteria derived from the work of Gordon Pask.

Audion allows a user to interact with a field of “birds” by placing digital objects in an on-screen environment. These objects modify internal states of the birds: hunger, fear and attraction. Individual birds and their internal states trigger and modify individual grains of audio, and ecosystem-wide variables control parameters of the resultant granular sound. The audio output references the ecologically inspired granular synthesis approach of Barry Truax and Damian Keller. Based on the stated criteria and provided context, we suggest improvements for Audion, such as applying the simulation's emergent structures to the structure of audio output and creating environmental pressures that are independent of the user's manipulations.

Graduation Thesis
Media Technology M. Sc.
Leiden University
July 2020

“Most of all, the world is a place where parts of wholes are described.”

The Books, “Smells Like Content”

I. An overture, of sorts

When composer and music producer Brian Eno spoke about generative music at the 1996 Imagination Conference in San Francisco, he recalled a visit to a Bay Area science museum nearly two decades earlier. He saw a computer displaying a simulation known as "Conway’s Game of Life."

“Life was the first thing I ever saw on a computer that interested me,” he said, “Almost the last, actually, as well.” [1] Life is a relatively simple program. Within a grid of squares, a square can either be alive or dead. Each square, or cell, is affected by its eight surrounding cells. At each time step, depending on the number of adjacent cells that are dead or alive, a cell will become alive, stay alive, die or stay dead.

“The rules are very, very simple, but this little population here will reconfigure itself, form beautiful patterns, collapse, open up again, do all sorts of things,” said Eno. Life exemplified his approach to composition on prior projects, such as Ambient 1: Music for Airports, and future music: If sets of parameters are passed through a process of relatively simple rules, the results can be varied almost endlessly. [2] Eno also praised composers Steve Reich and Terry Riley for their pieces “It’s Gonna Rain” and “In C.” Their work, he argued, represented a new idea of composition in which the composer had “a packet of [musical] seeds.” These musical seeds, much like the seeds of tomato plants or sunflowers, grow into the same type of thing when they are planted. However, importantly, “they turned into a different version of that piece every time.” [3]

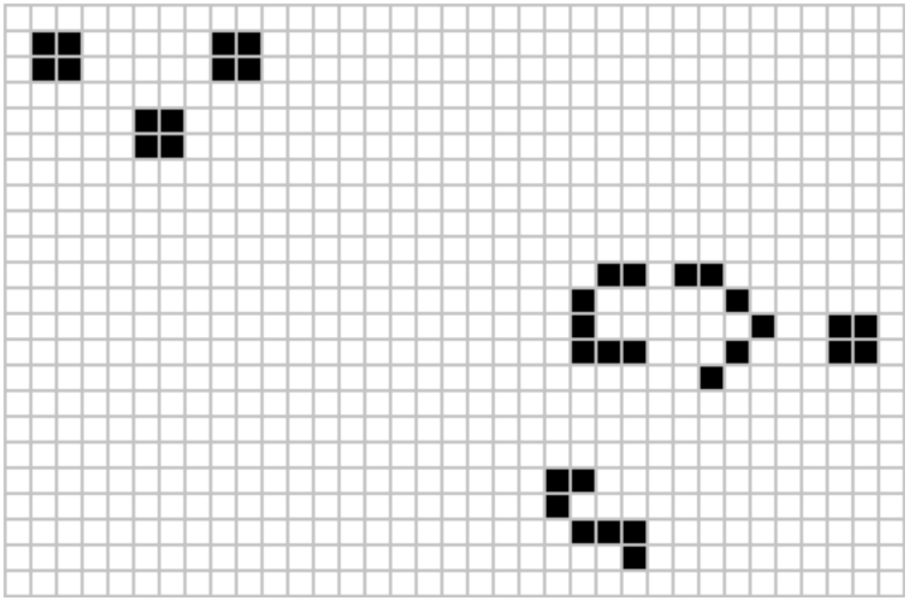

Fig. 1: A frame from the computer simulation “Conway’s Game of Life.” [4]

It is apt that Eno draws on natural metaphors to explain his fascination with generative music. Nick Collins summarized Eno's definition of generative music as "algorithmic music that happens to produce output in real time," and that is how we will approach the term here. [5] For centuries, scientists and musicians have connected music to natural phenomena, both intuitively and empirically. The Greek philosopher Socrates considered music to be an essential discipline for the study of philosophy. Music and astronomy, the ancient Greeks believed, were physical counterparts to arithmetic and geometry, and music was so fundamentally linked to the natural world that until medieval times, scholars would write about “The Music of the Spheres” in reference to planetary motion. [6] Similarly, medieval Islamic scholar Ibn Sina argued that the quality of a piece of music was derived from its timbre, organization and rhythm—and how these attributes reflected the natural world. [7]

Twentieth century composers, musicians and scientists have maintained the study of music in proximity to natural science. Artistic and scientific experiments with nature-inspired algorithms or complex statistical analysis have sometimes hewed quite close to Eno's notions of a plant's seeds and the resultant individuals. As Jaime Serquera and Eduardo Reck Miranda noted, cellular automata simulations like Life "are of interest to computer musicians because of their emergent structures … This is attractive, especially for those synthesis techniques that demand large amounts of control data over time." [8]

Miroslav Spasov discussed the origin of the phrase "emergent properties," coined by C.D. Broad and echoed in Eno's lecture. Emergent properties are attributes that appear "at a certain level of complexity, but do not exist at lower levels." Originally, the phrase was meant to link the "non-symbolic cognitive processes" in the brain to more "complex" social systems such as language and music. [9] Any set of connected neurons is unremarkable on its own, but collectively they structure our consciousness.

We might reasonably connect the original context of emergent properties in cognition and Eno's fascination with Life. We hear it in Eno's own words: the population will "reconfigure itself." While he almost certainly did not mean to in a strict, formal sense, Eno effectively ascribes a sense of agency to the cells of Life. These cells have a goal (perpetuation) and a means to act (the rules of the simulation), and the forms of clusters are made unpredictable by the interactions of the many constituent cells. Conway's Game of Life could be understood as a three-way tension between agents, a stated order and environmental variables. This tension is part of much of Eno's ambient work, and also his "Oblique Strategies" cards. [3]

The idea of agency, we argue, is particularly interesting to generative music, because a perception of agency might enrich the experience of the musician. Spasov noted the importance of agency as a concept essential to interactive systems for musical performance and criticized interactivity that was based on "a user-controlled system that reacts in a more-or-less prescribed way." An ideal generative music system, he argued, "allows agent technology to create content, and even structure, based on the dynamic interplay of parts." Importantly, he added that the interplay between a user and such a system (as opposed to the interplay between computer agents within a system) was another point for observing emergent properties. [9]

We must also remember that the previously discussed works of Eno, Reich and Riley exist within specific technological contexts. When Reich recorded "It's Gonna Rain," inexpensive tape recorders were fairly novel, and until decades into Eno's career, access to computers was mostly for academic institutions or businesses. Recent music history offers many examples of technological limitations fostering new techniques, or even whole genres, but in looking at Eno and Reich, we ask what the technologies of this particular moment enable that were not previously easily realized? In precis: What's new?

In a lecture at McGill University's Centre for Interdisciplinary Research in Music, Media and Technology, Miller Puckette pointed out that computers only recently became capable of real-time, musical stage performance. In response, he wondered: How might we approach computers as instruments? "Perhaps a way to answer that is, well, what is it that you would like to do [with a computer] that you can't do with a banjo?" [10]

Many computer music environments, Puckette observed, are capable of roughly the same set of outputs, but distinguished from each other by "the efficiency by which they would make one subset of the set of all possible sounds." [10] If we understand the goal of a musical tool as enabling a user to move from the imagining of a sound to its production, then we might view Puckette's approach as one defined by the friction between user and the realization of an idea. He expressed that the creator of a musical software environment is, at their worst, a censor who obstructs musical ideas. [10]

By concluding that musical software stands in the way of output through its relative ease or difficulty of use, we presuppose that a particular "subset of sounds" is the goal. However, in generative music, the goal of the composer is often tangential to, or in opposition of, Puckette's proposed efficiency. In many generative systems we might begin with a subset of sounds or musical phrases and then seek to extrapolate them into a great multitude of new sounds.

To answer Puckette's question with respect to generative music specifically, we might say that we want a computer-as-instrument that transforms our input algorithmically. An essential facet of the computer instrument as discussed here is the ability to translate input to output in real time, and one component of our ability to interpret these outputs is the appearance of emergent structures. Perhaps the instrument, then, is a combination of real-time permutation and the perceivable structures of agency.

We contend that advances in interactive systems have fulfilled much of the promise of agency for generative music. Furthermore, the ubiquity and portability of the computer hardware now allows for generative, agent-based systems to be incorporated into live performance or improvisation. We will consider these positions specifically within the long history of looking to nature to inform musical composition. We argue here that the interactivity offered by agent-based systems, when applied to live performance and improvisation, might realize more fully the premise of Eno's "musical seeds."

By evaluating the strengths of existing systems, we will then be able to propose our own framework and system for real-time music improvisation and performance.

II. Background

In Johnathan Swift's satirical story Gulliver's Travels, the protagonist recalls a tour of the scientific academy of the fictitious city of Lagado. There, he sees a professor surrounded by pupils. The students stood at the edges of a large matrix of a wooden cubes, and words from Lagado's language were painted on the faces of each cube. Upon the professor's instructions, large cranks would be turned, and a new combination of words would be formed on the cubes' visible faces. The output would be recorded in a large ledger. The professor said that "by this contrivance, the most ignorant person, at a reasonable charge, and with a little bodily labor, might write books in philosophy, poetry, law, mathematics and theology, without the least assistance of genius or study." [11]

Eleanor Selfridge-Field considered the academy's machine when she wrote about the history of combinatorics in music. [12] Combinatorics is the "branch of mathematics studying enumeration, combination and permutation of sets of elements, and the mathematical relations that characterize their properties." [11] By considering the Legado machine, Selfridge-Field raised interesting questions about the influence of algorithmic design on the output of a generative system.

For our discussion, we will place algorithms for generative music into three broad categories: combinatorics, statistical models, and nature-inspired algorithms. By examining the historical use of each approach, we might inform the design of a compelling system for musical improvisation and real-time performance.

Limits of Endless Loops

Algorithmic processes have been applied to creative endeavors for millennia, and Karlheinz Essl pointed to combinatoric outputs in sources as diverse as the Jewish Kabbalah and the musical dice games of Joseph Haydn when he argued that algorithms are "a method of perceiving an abstract model behind the sensual surface, or in turn, of constructing such a model in order to create aesthetic works." [14] Historical examples such as musical dice games or Music for Airports represent a combinatoric approach that utilizes groups of phrases from a particular composer, and as a result, those exercises convey many of that individual's stylistic traits.

At this point, it is useful to consider what Tero Parvianinen called "generative method" and "generative product," and consider how these two ideas relate to some of the combinatoric approaches already discussed. [15] Generative methods include Steve Reich's modification of tape playback for "It's Gonna Rain" and Eno's overlapping tape loops for Music for Airports. The designed setup will produce something that cannot be (easily) predicted beforehand, but it will produce essentially the same output each time. Importantly, there is no component of indeterminacy within the system once it begins producing its music. As Collins noted, strictly classifying generative approaches is contentious—as are the implications of classification for authorship—but examining these broad categories provides some useful context. [5]

Music as a generative product is perhaps akin to Eno's musical seeds: a procedure for music generation is defined, but the process includes variables so that even the same inputs might produce significantly different products. Parvianinen offers Riley's "In C" as an example of a generative product. The piece consists of 53 sequential musical phrases written by Riley, and each can be played by a musician as many times as they wish, so long as all the musicians stay within a certain range of phrases. [16]

As Eno said in his lecture: "[Generative music pieces] simply don't ever re-configure in the same way again … The considerations that are important, then, become questions of how the system works and, most important of all, what you feed into the system." [1] This points to two potential limitations of generative methods: Although the initial setups are alterable in a way that might produce different ordering of the elements, the variety of the output is still limited by 1) the physical set-up at the outset of the generative process and 2) a listener's conception of how different an output would need to be to constitute something genuinely new.

In the case of "It's Gonna Rain," if we view the input as the tape loop, and the system as the tape decks—modified by Reich putting "[his] thumb on the recording reel of one of the machines"—we see how changing the input slightly is unlikely to change a listener's perception of the piece. [17] Certain recording artifacts might become more or less pronounced at certain moments, or the words might combine differently, but most listeners probably do not evaluate the piece as many distinct pairings of the recorded loop, played in various degrees of synchronization. Instead, our attention is drawn to the gradual shifts of phase, the realignment of the words and the recording's ambient sounds. If this is how we perceive the piece, it is difficult to say that we can change it on a fundamental level without either significantly altering the input or changing the system.

In contrast, Riley’s “In C" introduces a strong element of variability. Musicians can choose how long they play any given pattern, and are encouraged by Riley’s instructions to change based primarily on personal musical preference. While the separate decision making of its performers introduces randomness into the piece, its static input material (“the same page of 53 melodic patterns” played “within 2 or 3 patterns of each other [sic.])”) limits the output of the system in a way that a meaningfully different output of the system might rely on other choices, such as instrumentation. [18]

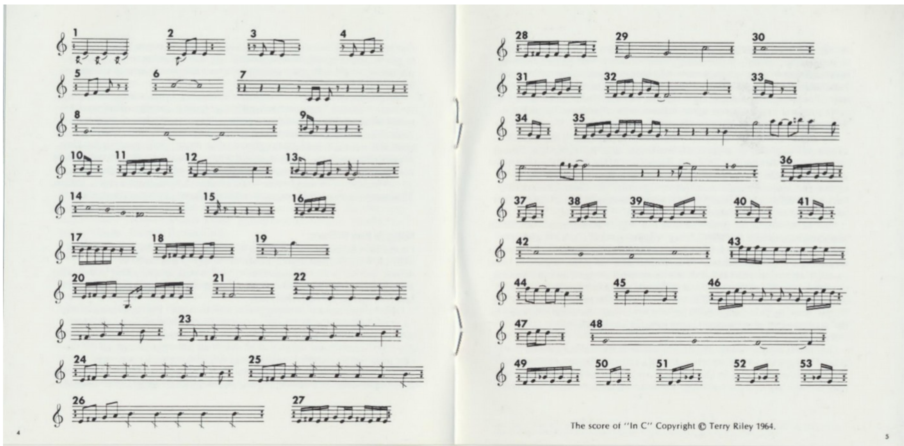

Fig. 2: The fifty-three musical phrases that comprise the score of Terry Riley's "In C." [19]

While the generative methods and generative products described above produced influential musical ideas and recordings, there is a reasonably transparent link between the inputs and outputs. For our line of reasoning, however, it is notable that what might be the most reconfigurable work of generative music discussed, "In C," is also one in which the performers are granted a degree of freedom—of agency—to interact with the set of instructions.

That is not meant to suggest that obfuscation between input and output is the hallmark of generative-musical excellence. Rather, this analysis of "In C" is meant to support the notion that agency is a compelling component for generative systems for the indeterminacy it offers.

However, in the mid-20th century, digital computers emerged as a powerful tool for musical exploration. Beginning with early experiments and continuing to contemporary neural networks, many methods of computer-assisted composition are guided by statistical models. With this in mind, we should consider the role that these models have historically played, and how that might inform the design of a future system.

Markov Walk with Me

While algorithmic creativity has a long history, Essl argued that it is only since the 1950s and the introduction of computing that algorithmic composition has “drastically” changed our understanding of the roles of composers. He wrote that, since then, the composer has moved away from the image of a “demiurge who controls every tiny detail.” As a way of conferring control outside the composer, some of the earliest entries in computer-assisted composition made use of random choices cast within statistical frameworks, “in order to create music which is not limited to a fixed appearance.” [14]

In the 1940s and 1950s, Markov chains gained a significant amount of interest from theorists analyzing existing music. In 1957, University of Illinois researchers Lejaren Hiller and Leonardo Isaacson used Markov chains in the first documented attempt to generate music from them, as part of the Illiac Suite. Two years later, Greek-French composer Iannis Xenakis would use simultaneous Markov chains to create his works Analogique A, Analogique B and Syrmos. [20]

Randomness was also a source of inspiration for composers outside of computing, such as John Cage. His piece Imaginary Landscape No. 4 (March No. 2) is scored for 24 performers, adjusting the tuning and volume on 12 radios. Cage hoped to remove all predictability from his piece and said that he wanted to erase "all will and the very idea of success." [21] This, Essl wrote, confronts what Cage thought was a "primary obstacle" to music—the temporal. [14]

Here, when it is suggested that time is an obstacle, that does not mean a composition ought to be infinitely long, but rather, it should contain infinite possibilities. No matter how rapt, the people in an audience have babysitters waiting at home, house plants to water, and parking meters to feed quarters before their cars get towed. We might, more appropriately, think of this temporal "obstacle" as a constraint on the composer. Composers can only spend so much time writing their scores, and randomness is one possibility for how they can produce the greatest possible output from the least amount of input.

In the statistically guided Markov walks of Xenakis, Lejaren and Hiller, and in the environmental randomness of Cage, we see the limitations of chance decisions for generative music. As Herremans, Chuan and Chew wrote, "Melodies are more than a just movements between notes," and that the note sequences produced by Markov chains and other statistical models "typically [do] not enforce patterns that lead to long-term structure." [22] In the case of Imaginary Landscape No. 4, Cage was consciously challenging these structures, but even that piece is scored with very specific instructions. Environmental variables (How clear is a radio's reception? What is playing on a particular station?) are volatile, and how we respond to them is influenced by the context of the performance. Furthermore, certain structural variables, such as the movements of the radio dials written into the score of the piece, are inflexible.

The information theorist Joel E. Cohen also pointed out that there are certain aesthetic considerations that apply to statistical analyses. He argued that by selecting "a few 'master works'" that would serve as the informational basis for computer-generated compositions, we ignore the important external relationships between the audience and those works. These preconceptions are "important, if not dominant, in determining the 'value' of a work." In order to base music generation on a corpus of existing works, we must also assume that "no letter not already known in the alphabet can occur." [23]

Neural networks—computer programming models inspired by the connections between the neurons of the brain—have shown significant promise for composing music. A variant of Google's Magenta neural network, trained on 10,000 hours of piano music and "inspired" by Satie piano pieces, produced an album of one-minute tracks in January 2020. [24] However, neural networks do not eschew the problems Cohen or Selfridge-Field suggested. A network's output is determined by the choices made in curating its training data.

Before we write off the use of statistical modeling for understanding music and composition, we might de-emphasize its use for making many discrete decisions, and instead focus on how probabilities play a role in our comprehension as listeners. David Temperly analyzed the ways in which probability affects a listener's understanding of music:

[We] hear a pattern of notes and we draw conclusions about the underlying structures that gave rise to those notes: structures of tonality, meter and other things. These judgments are often somewhat uncertain, and this uncertainty applies not just at the moment that the judgment is made, but to the way it is represented in memory ... Certain note patterns are probable, others are not; and our mental representation of these probabilities accounts for important musical phenomena such as surprise, tension, expectation, error detection and pitch identification. [25]

We do not understand music as any single, given event. Rather, it is a set of events, whose importance relative to each other is in flux. To return to information theory: "In the cultural signs system of melody, for example, the notes, intervals, motifs, phrases, periods, sections and movements all constitute letters of different alphabets." [23] Markov models and other stochastic methods offer utility for generating discrete events, but other means are necessary to implement longer-term cohesion in creating music.

The Natural Order of Things

While musicians do not typically expect an instrument or a tool for making music to impart structure into the resulting music, we can see how this might be useful for generative music: A system that can bring structure to generative music—and especially one that is intended to function in the context of live performance—might allow a musician the ability to take advantage of the system's inherent structure, or use the system's interactive properties to confound its structure. Just as Eno perceived a connection between the artificial agents of Life and musical possibility, we see an emerging body of scientific work applying nature-inspired algorithms or biological computing methods to music composition.

Fittingly, Essl wrote that "the principle of indeterminacy and the statistical organization of mass structures can also be found in nature—'natural events such as the collision of hail or rain with hard surfaces, or the song of cicadas in the summer field.'" [14] Nazmul Siddique and Jojja Adeli wrote that nature-inspired computing is apt for situations when "the problem is complex and nonlinear and involves a large number of potential solutions or has multiple objectives" or "a diversity of solutions is desirable." [26] While it may sound strange to describe music-generation as a "problem," the point stands: composition is a field with many intersecting goals, and possible solutions that can be assessed on many levels, simultaneously.

Flocking algorithms have also garnered significant attention as generative systems. These algorithms use computer agents with simple sets of rules—not unlike the cells in Life—to recreate the effects of groups of birds in flight (or schools of fish swimming, etc.). As Jan Schacher, Daniel Bisig and Philippe Kocher wrote: "Such phenomena are characterized by a structural organization, which emerges from processes of self-organization, and combines regular and chaotic properties." [27]

However, the emergent property of self-organization and coalescence near an optimum should not be construed as inherently advantageous for generating music. Schacher, Bisig and Kocher pointed out that it might be "tempting to use swarm simulations … as a generative 'ready-made'" and argued that simply processing numerical output of a flocking simulation to meet existing, specific criteria for its output undermines the point of using the algorithm in the first place. "A fundamental challenge in generative art and composition" they wrote, "relates to the establishment of meaningful and traceable mapping relationships between underlying algorithmic processes and the resulting aesthetic output." [27]

Maximos Kaliakatsos-Papakostas, Andreas Floros and Michael Vrahatis reflected on the difficulties of defining optimization and reducing user passivity while generating music with particle swarm algorithms. Their research proposed a system of real-time feedback from listeners to steer the movements of particle swarms. The positions of agents represented attributes such as rhythm, tone, polyphony and intensity, and the favor a listener assigned to a particular musical phrase generated by the system would adjust a genetic algorithm that controlled the agents. However, one of their most interesting conclusions could be that:

> Human users may not be certain about the ideal features that they require from a music piece … some potential 'ideal feature' a human rater may have in mind at some point during the simulation may be influenced by a melody that she/he hears during a rating round. [28]

The changing desires of a user might have posed difficulties for those particular researchers, but those changes have significant implications for generative music. Systems that are able to adjust to the changing input of a user—while simultaneously providing output for the user to evaluate—create an interactivity that has been absent from many of the previously examined historical approaches.

The historical disconnect between composer and interactive system might be the result of the relative newness of the computer hardware that makes such a feedback loop possible. The timely exchange between composer and computer would have been unthinkable with the computer that created the Illiac Suite. In the way that the ability to easily record music to tape—and then modify those tapes and machines—led to new ideas in both musical creation and performance, we should understand computerized agents as having the same effect, and design systems for expression accordingly.

III. An Ecosystem

In many of the preceding examples, we see a tension between what we will describe as the poles of fidelity and novelty. Here, we do not mean fidelity as a technical measure of faithful reproduction between a recording and its source material. Rather, we use it to describe the relationship between a musical work and its underlying premise. Novelty here means not only if a piece of music is new, but if it also introduces distinct ideas that represent a break with or reinterpretation of musical tradition. But, as Temperly and Cohen both noted, we interpret music through our prior exposures to it, but we seek it—at least partially—for unexpected delight. [23, 25]

With this in mind, we are able to see how the concepts of fidelity and novelty apply to live performance: A virtuoso musician could be lauded for both her ability to reproduce the notes of a cello sonata accurately and her ability to inflect those notes with personal affect. Even highly improvisational styles such as jazz represent this dynamic between fidelity and novelty. How skillfully a musician draws from a body of influences and imagination—and how that interpretation engages the composer's material—can weigh quite heavily in the audience's evaluation of the performance.

We can apply the fidelity/novelty paradigm to musical instruments, too. The relationship between the intended input and output of an instrument is quite predictable, by design. While certain guitars are prized for a particular timbre, the guitarist also probably prizes the instrument's ability to produce an F# when he presses down on the second fret of an E string. The 20th century's "Algorithmic Revolution" might be characterized as undoing the expectation that musical scores exist as static objects. [14] What happens, then, when the relationship between the input and output of an instrument is also no longer static? And what mechanisms might we use to ensure that the relationship between input and output is not so obscure that it becomes meaningless?

We have discussed some of the limitations of pure randomness in generative art—namely that, when more decisions are made randomly, the output increasingly risks becoming noisy, inscrutable monotony. But we are also reminded of Essl's observations about the rhythms of hail stones, and the emergent structures of cellular automata. Composer Daphne Oram noted such phenomena quite poetically:

> Have you ever tried musing in front of a flickering coal fire? The coals form fascinating, grotesque shapes, some fiery red, some sullen black. Tongues of flame, blue and yellow, create crazy rhythms as they dance. You cannot predict what will happen next, yet you feel beneath it all a consuming pulse. [29]

Jon McCormack examined nature-inspired algorithms in the context of creativity, and he did so in terms very close to the proposed concepts of fidelity and novelty. McCormack wrote that generative work should be "novel and appropriate … to the particular aesthetic domain." [30] To that end, he also noted the artistic applications for ecosystem models: "Evolutionary synthesis is a process capable of generating unprecedented novelty. The aim is to structure these artificial ecosystems in such a way that they exhibit novel discovery in a creative context rather than a biological one." [30]

We do not mean to suggest here that algorithms inspired by nature offer intrinsic creative solutions, or that a non-ecological model is inherently not creative. Rather, nature-inspired synthesis excels at producing novel output because it is adaptive or evolutionary. While we can understand the factors that created a particular output, re-engineering it might be impossible. This is why natural simulations, and especially ecosystems, are of particular interest for musical improvisation.

And to return to our original question—What happens when we embed an active system for novelty in the instrument?—we first have to understand how such a system might be implemented.

Prior Proposals

In proposing an instrument or system for nature-inspired generative music, agent-based models hold significant appeal. Flocking algorithms have been used extensively for composition. Schacher, Bisig and Kocher wrote that the use of flocking algorithms for generative art can bring the algorithms' "structural organization, which emerges from processes of self-organizations and combines regular and chaotic properties." This give-and-take between organization and chaos is apt for the balance between structure and surprise we seek in making music. [27]

We can also adapt a flocking model to have multiple levels of organization. At the simulation level, flocking agents can form structures or oscillating patterns in aggregate. A second, inter-agent structure is also apparent: a simulated bird in a flocking algorithm often has rules that guide its individual movements with respect to other birds. Finally, we consider an intra-agent layer, consisting of the motivators for each agent's actions in each time step of the simulation. The separation between system, inter-agent and intra-agent states need not be stark because the interaction between them is important.

We might think of these levels of abstraction for a flocking simulation as corresponding to structures within music, such as movements, phrases and notes. Musicality could be said to be an emergent property of individual notes and their relationships to each other in time and pitch. In order to translate a simulation into more complex musical forms, then we might experiment with a simulation whose agents have meaningful relationships to each other. Robin Gras, Didier Devaurs, Adrianna Wozniak and Adam Aspinall wrote that careful modeling of individuals' behaviors using a map of competing desires can allow for a more precise emergence of the system's properties than starting with calculated birth and death rates, as are often used in "classical approaches." [31]

But how should a user interact with such a system adapted for generating music? Gordon Pask provided four primary considerations of interactivity for his Musicolor environment. First, a sufficient—but not overwhelming—amount "controllable novelty." Second, the ability for a user to interpret various levels of abstraction. Third, cues that "guide the learning" through various levels of abstraction. Fourth, the system's ability to "adapt its characteristics" to engage the user. [32]

Pask's considerations are useful for evaluating the relationship between the user and the artificial system, and he also proposes a system of "mobiles," or small autonomous vehicles, to guide the process of interaction among agents in a system. Agents' goals "should be partially incompatible" to create competition. "Some goals," Pask wrote, should require more than one agent to achieve. Each goal should have discrete sub-goals. Both main goals and sub-goals should have some element of cooperation. Finally, the pursuit of goals should be "embedded in each mobile." [32]

"The really interesting issue is what happens if some human beings are provided with the wherewithal to produce signs in the mobile language," Pask wrote. His mobiles would produce "complex auditory and visual effects" on their own, but the human would use the rules of the system and the tools for communication to produce a desired outcome. [32] Pask's notions might be echoed by McCormack, who saw the potential for a "recursive coupling between system and environment" with certain attributes of fitness evolving at a system-wide level. [30] While Pask did not consider an element of artificial evolution as part of his mobiles' design, the ability for a system to select its constituent agents' for fitness also offers significant potential for novelty.

These are the Rules

From Pask's and McCormack's work, incorporated with the previously mentioned ideas of complex internal states, we propose our own framework for an interactive, multi-agent system for musical improvisation and performance:

1. We should be able to prompt the system, not control it. The system and its agents should respond to the users' actions in ways that are predictable in principle, but not in the particular execution of any given instance.

2. The system should be able to be interpreted at several, separate levels. Each level should be readily apparent to the user (visually and audibly), and the user should be able to manipulate each level.

3. The system should adapt to input, but also according to its own rules. If user input is minimal or stagnates, the system should continue to evolve. This does not necessarily mean the system should be self-sustaining. Total collapse should be a possible outcome.

4. The system should have a low barrier for entry, and a high bound of exploration. This is arguably the most subjective of these parameters, but it is worthwhile to consider existing metaphors for interactivity (piano keyboards, the patch cables of a modular synthesizer, etc.).

These criteria guide the creation and ongoing iteration of a software tool we call Audion. While the project is ongoing, its presented form is offered as a starting point for real-time musical exploration with multi-agent systems.

IV. Designing an Expressive System

The design process for Audion began by considering Eno's fascination with Conway's Game of Life, and translating that simulation directly into a musical interface. This initial step was envisioned as an initial exploration, to help prioritize elements of control and implementation, rather than as an end unto itself.

The program's design was inspired by the Tenori-On—a digital musical instrument created by Toshio Iwai for Yamaha—and other two-dimensional step sequencers common in live, electronic music performance. [33] This particular metaphor was chosen for the visual similarities between the Tenori-On interface and graphical representations of Life.

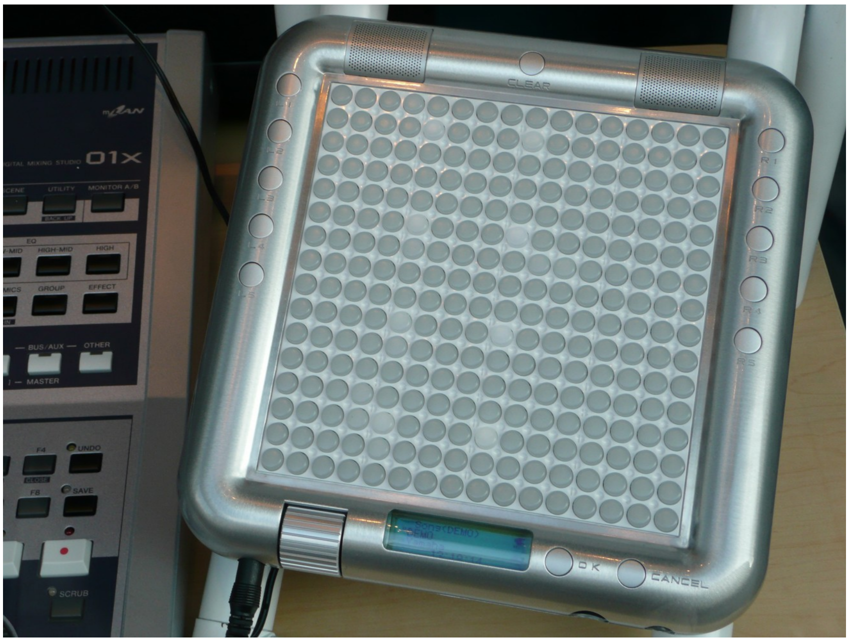

Fig. 3: The Tenori-On, an electronic instrument created by Toshio Iwai. [33, 34]

The interfaces for this program and the proceeding iterations were realized in LÖVE, a 2D game engine for the Lua programming language. LÖVE was chosen for its perceived speed, simplicity, portability and active community of developers. [35] Additionally, the Pure Data visual programming language was used to handle audio manipulation and synthesis. [36]

In this program, the user can begin by setting initial parameters, such as the width and height of the grid in cells. The grid begins with all cells dead, and by clicking on individual cells, the user can activate them either prior to starting the simulation (to initialize the simulation with a desired pattern) or as the simulation runs (to interact with or disrupt patterns that form). When the simulation is started, the scan line moves from the top of the grid to the bottom at each time step. The simulation's cells also advance once each time step. Users can adjust the frequency of the time step in beats per minute.

Each horizontal row represents a time step, and each vertical row represents a note of the C major scale. During each time step, the living cells' corresponding notes are played. By "drawing" several configurations of cells that move in the plane but oscillate in their form, we hear patterns of note intervals and rhythms emerge from the simulation.

As previously stated, this implementation was intended to function as a starting point for exploration. While this version meets some of our stated criteria—its mechanics are quite simple, and its cells will carry on or die without user input—we might struggle to suggest it is unpredictable or that its agents' movements represent multiple levels of meaning.

In an attempt to increase the level of interaction and unpredictable behavior in the system, the ability to entirely eliminate any cell from the simulation was added. When a cell is eliminated, it functions as a dead cell that cannot become alive. This adds the ability to control which notes are not played, but doing so has the consequence of forcing the cells' patterns to adjust around the gaps in the simulation.

We are interested in the emergent properties of this (or any) system, but we can only affect those properties by changing the conditions of the simulation's current time step. In order to produce a desired structure-wide result, we have to predict how our immediate modifications will affect distant time steps. It is this relationship between between desire and outcome that might create the interest for both the listener and the user, and it is these types of changes that we attempted to implement in later iterations of the program.

Using Agent Motivations to Guide Action

As previously discussed, multi-agent systems that model properties such as age, adaptability or desire for shelter of their constituent individuals have advantages over models that are predicated on assumptions such as birth and death rates. [31] While this might lead to a "more detailed" simulation, that is not meant to suggest that these models will lead to more inherently interesting output than a similarly sonified, but less complex, simulation. Indeed, Manuel Rocha-Iturbide warned that focusing on the complexity and accuracy of a simulation for sound design runs the risk of rendering the simulation "more important than ourselves, and we become passive, content to merely observe the results." [qtd., 8]

As a test of this premise, a simulation with two types of agents was created. Within a two-dimensional space, a population of "birds" and "trees" was initiated. Periodically, each tree produced a random pattern of offspring within a certain radius, with the number of offspring trees selected at random within a normal distribution. The birds would forage, searching for trees to eat. Each bird had a randomly generated field of vision, and if a bird was hungry and a tree was within its field of vision, it would move toward that tree.

Additionally, all birds were programmed with a sense of affinity toward other birds. For instance, bird A would would approach other birds that were in its field of vision, if bird A was not too hungry. Otherwise, it would prioritize food. Each bird also had an individual lifespan and maximum speed at which it could move.

For translating the simulation into audio, the linear velocities of birds were used to control band-pass filters applied to white noise. In contrast to the grid-based input of the previous simulation, this approach allowed for a range of outputs beyond prescribed musical tones. This was a decision made to test the system's capacity for diverse outputs, in line with the goal of a high barrier for exploration. By interpreting the state of a bird (its velocity) that is the result of internal goals (to seek food or peers), we also begin to see layers of meaning established, although admittedly not ready to be manipulated. Succinctly, if our first exercise was one that leaned heavily on exploring user inputs, this example was meant to examine the benefits and possibilities of layering complexity within a system.

When the simulation is run, we hear noise (as to be expected) as the birds forage. Their foraging movement is semi-random, and as such, the frequencies of the corresponding band-pass filters are shifted back and forth. However, periodically, we hear certain frequencies—perhaps not unlike the sound effects of a cartoon flying saucer—as certain behaviors happen. As birds accelerate, their increasing velocity as they move across the plane to food or peers results in the band-pass filters shifting up in frequency, in tandem.

It is important to note that the parameter of each bird that is mapped to sound, its linear velocity in the X- and Y- directions, was selected for its visibility. That is, unless we have a way of illustrating or gleaning information about internal states of the birds, we cannot interpret the multiple layers of meaning. While we can surmise some details about a bird's target based on the trajectory of its movements, the translation of movement into sound means that we only hear the resultant action of an internal state, rather than the process of state change or the relationship between states.

We might also argue that when we turn our attention to the information gleaned from this system's audio, we find that most of the audio output contains little or no meaning. In this simulation, a bird's foraging movements are random until it sees a tree to eat or another bird to join, so the filter it applies to the white noise generation is accordingly random. While aggregate randomness could be said to be a form of information—that is, we can glean from unstructured noise that no bird is moving toward a target—that information is still muddled by the fact that it is similarly transformed by separate events. A bird approaching food and approaching another bird, or several birds flocking together, perform essentially the same transformation on the audio. Additionally, a flock of birds that might emerge is a notable event within the system, but cannot be conveyed because its constituent birds continue to move mostly randomly, albeit in close proximity to each other. The individual corrective actions taken to maintain a flock are quite small, so the accompanying audio cues are not perceived as the flock moves.

This desire to transform the events of our system into discrete, informative sounds is reminiscent of what Damian Keller and Barry Truax called an ecological approach to composition. They argued that rather than using prescriptive musical forms, "time be parsed into informationally relevant events … attention-based processes are triggered by organized transformation, not by redundancy or randomness." [37] Keller presented his ecological approach to composition in connection to J.J. Gibson's notion that "information is structure that specifies an environment to an animal." [38] A loop forms between environment, information and animal, and this gives rise to compositional structure. "Actually occurring events" form the basis of the structure rather than arbitrary measures of time, and as a result, change "is not simply the fluctuation of variables: it dictates how these variables are observed." [37]

In our simulation, we can see an event-driven relationship between agents and the environment as birds move, trees multiply, and birds eat. The changes in the environment and resultant actions of birds gives rise to the structure of the sound. But if we analyze this system in reference to the computer as instrument, we see that we have decoupled the musician's interpretation of informative events from a method of input to the system. If we want to engage all four points of the previously described framework, we might then focus on McCormack's idea of "interactive evolution" through "environmental modification," and allowing a user to impart their intentions back into the system using knowledge gleaned from the system. [30] But what would this look like?

Audion

For the next (and, with respect to this writing, final) iteration of Audion, the focus was incorporating all four of the stated design criteria, and conveying information between the user and the system and developing the expressive capabilities of the program.

This iteration was an expansion of the previous birds-and-trees metaphor, however the trees were removed as separate agents. At startup, the user is once again presented a two-dimensional space with an initial group of birds whose parameters are randomly instantiated. Each bird has three internal states: fear, hunger and desire to mate. The level of each desire (assigned between one and 100) fluctuates with time and in response to environmental variables that the user can manipulate. Each bird's internal states are also given individual weights, so some might respond more strongly to fear than hunger, etc. Each bird also has a gender, maximum speed, a lifespan and a field of vision that are assigned within a normal distribution.

Each state has a corresponding, discrete action. When a bird reaches its threshold for hunger and a piece of food is within its field of vision, it eats. When a bird reaches its threshold for desire to mate and can "see" another bird, it moves to mate. When a bird reaches its threshold for fear, it will "flee." Eating and mating return the levels of hunger and desire to 0, respectively. When a bird flees, its level of fear decreases in proportion to the distance between it and the threat.

In order to manipulate the environment and affect the birds accordingly, the user can place four different object types into the simulation. Each object has a radius of efficacy, and each object can be connected to an audio file for sampling grains. The grains are triggered by a bird eating, mating or fleeing within the object's radius. The four objects and their functions are explained in **Table 1**.

Table 1: User-placed Objects in Audion

Object	**Icon**
The **"feeder"** object disperses the pieces of food that the birds need to survive, and its audio is triggered by birds eating.	
The **“watcher”** object accelerates the increase of a bird’s desire to mate, and its audio is triggered when a pair of nearby birds mates.	
The **"hunter"** object increases a bird’s level of fear, and its audio is triggered when a bird flees from it.	
The **"plauge"** object has no effect on the states of individual birds and produces no sound directly, but a percentage of birds are killed when they linger in its effective radius.	

Motivation for the use of granular synthesis

We mentioned the work of Truax and Keller in analyzing the relationship between environmental event and audio event within the previously described agent-based simulation, and their work with granular synthesis heavily influenced the final proposal for Audion. Granular synthesis is the production of "complex sounds ... based on the production of a high density of acoustic events called 'grains.'" [39] In the case of Audion, these grains are drawn from user-selected audio files.

Manipulating grains of audio allows for the transformation of sound on multiple time scales. Curtis Roads wrote of the "macro," "meso" and "micro" time scales of music, which we also see in the work of Truax and Keller. [37, 40] An individual grain, typically measured in milliseconds, represents the micro time scale. The meso time scale represents collections of sounds into phrases, and the macro time scale encompasses larger structures of musical forms. A micro-scale grain has qualities imparted by its audio source and amplitude envelope, and the combination of grains can be manipulated to create rhythmic patterns, tonality and noise on the meso time scale. [40]

Truax and Keller used physical models derived from natural events to control grains on the meso and macro time scales, and they wrote that through "interaction of the local waveforms with meso-scale time patterns … the [audio] output is characterized by the emergent properties, which are not present in either global or local parameters." [37, 38] Notably, Truax attempted to evoke the physical movement of water when designing a system for controlling grains in his 1986 composition *Riverrun*:

> "Riverrun creates a sound environment in which stasis and flux, solidity and movement co-exist in a dynamic balance … The fundamental paradox of granular synthesis—that the enormously rich and powerful textures it produces result from its being based on the most 'trivial' grains of sound—suggested a metaphoric relation to the river whose power is based on the accumulation of countless 'powerless' droplets of water." [39]

Likewise, we could see a metaphorical connection between granular synthesis and the foraging birds of our model. Any single agent, in response to the system's configuration and user inputs, triggers its representative grain, but the character of the individual and combined grains changes over time. By recombining and manipulating existing sounds, we generate new music, but the added possibility for interaction and real-time adjustment, as well as the unpredictable motion of agents, might ideally create a system for highly varied explorations.

Interaction Design

While the sonic component of any audio environment is obviously important, we must also address how directly it engages the proposed design criteria.

First, in order to allow a user to steer but not directly control the system, the user-placed objects addressed a specific parameter of the individual birds, but also the system-wide movement. By amplifying the birds' attraction, for instance, we increase the total number of birds, and thus the number of grains that might be played simultaneously. Higher levels of mating desire also increase the pitch of the grain played. Mating among the birds, however, passes along the "mother's" attributes to the offspring, within a certain allowance for random mutation. The ultimate result is that, depending on the combination of birds' attributes, as well as factors like the availability of food and the number of user-placed agents in the plane, we wind up with complex interactions that can be foreseen but not produced with absolute certainty.

Second, we should consider the levels of information available within the system and our ability to manipulate them. The system has at least three levels of control for users to interact with: the individual bird and its states, the interaction between agents in the system (bird-and-bird, bird-and-object, bird-and-food), and the ecosystem's properties as a whole (population, object placement).

It is also important that each of these levels can be interpreted and manipulated. To convey how the bird will interact with other agents, we can interpret its field of vision, dominant state and gender. The field of vision indicates whether or not a bird "sees" a bird or object to encounter, and its dominant state will determine how it responds. To this end, the bounding shape of a bird reflects its motivating internal state, as well as the boundaries of its field of vision; the inner shape of a bird indicates its gender.

To change the system's population level, we can place watcher or plague objects, and, to some degree, control the number of birds present through general trends of growth and death. The relative abundance of food can also lead to population decay or stability.

Third, we should consider the system's adaptability. While there is no explicit fitness function, as is common in many evolutionary algorithms, the competitive aspects of the simulation lead to some forms of evolution. Because much of a bird's ability to interact is dictated by its field of vision, over time, the simulation tends to produce birds with larger fields of vision. Similarly, because faster "female" birds tend to move between "males" more quickly to mate, there is typically an increase in the average speed of birds.

Expressive Quality

Finally, we should consider the ways in which a user can express musical ideas with the tools available in the system. Audion was designed in such a way that the user can shape facets of the timing, pitch and timbre of the audio output, but that is not to say the examination offered here is an exhaustive list of possibilities. For a discussion of manipulating these qualities, we will ignore the user's choice of audio file from which grains are extracted because this choice —while important—is arbitrary, and Audion's agents will behave similarly no matter the file selected.

Perhaps the most apparent way to control timing in Audion is by controlling the discrete actions that trigger sound grains. How this is achieved depends on the types of objects a user has placed in the environment. For instance, when a bird eats, it triggers the grains of a respective feeder object, so we are able to trigger those grains more frequently by manipulating the radius of the feeder object and/or the number of "seeds" it drops as food. Similarly, the radii of the watcher and hunter objects contribute to the frequency with which their grains are triggered.

If we consider the perceived movement of the audio through panning as an expression of timing (i.e. movement is a change predicated upon time), we can also manipulate the timing through the X-positions of objects in the field. The sound produced by an object is panned relative to the bird triggering the grain. For instance, if a pair of birds mates to the left of a watcher object, the sound triggered will be panned proportionally to the left.

Pitch control also has multiple methods for manipulation. Similar to the way in which the X-positions of a user-placed object control the panning as birds act near it, the object's Y-position manipulates the pitch of a grain, with a lower Y-position (with the origin in the upper left of the field) corresponding to a lower pitch.

Some sound parameters are meant to have a level of intuitiveness. Higher levels of attraction and fear for individual birds means the grains they trigger will be more erratic: a bird with a higher desire to mate will randomly select from a wider range of possible values to shift the pitch of the grain. Hunger is also intended to have an intuitive meaning: a bird with a higher level of hunger will trigger a grain with a longer duration.

Modifying the system's output as a whole is done with aggregate, system-wide variables derived from all the birds. The average hunger of the population controls the system's level of reverberation, with a hungrier population leading to a more lively reverberation. The average desire to mate controls a resonance filter applied to the audio output. As the average desire to mate increases in the simulation, the cutoff frequency of the resonance filter also increases.

From here, we can make the connection between grain size and the previously discussed manipulation of timing: The layering of grains and resultant polyphony emerge as a function of both actions taken by individual birds and the system-wide variables affected by the objects the user has placed in the simulation. This circular connection is part of the intent behind Audion. Much like the use of granular synthesis, this feedback loop between objects, agents and states acting at various levels within the system is intended to evoke Keller's notion that time in composition is derived not from time as independent from "actually occurring events," but rather as dependent on events happening in relation to each other. However, as Keller notes, an important element of this ecological model for composition "is the organization of spectrally complex samples into feasible meso-temporal patterns." These "meso and macro" patterns should then "[unveil] new properties resulting from the interaction of these levels." [37]

V. Evaluation

We might be tempted to end the evaluation of a system like Audion with user testing or by examining how well Audion meets the outlined design criteria. We hope that the software proves to be a useful system, certainly, but an evaluation that hinges exclusively on the usability of an interface would be incomplete or even unfitting. Ultimately, Audion is presented as an instrument for creating generative music, and it seeks to explore the possibilities of a real-time loop of information between the user and algorithm. It is imperative, then, that we consider the system on those terms.

In documenting the design and implementation of Audion, we have introduced ideas from a number of voices within music and computer science. An exhaustive analysis of how Audion relates to each system introduced might be prohibitively lengthy, so we will focus on three particular elements of Audion and the systems that inspired it: the ability to transform input into novel output, the translation of the system's emergent properties into audio, with a particular focus on the role of the user's input in both aspects.

Transformation of Input

Transformation is an inherent part of our working definition of generative music: an algorithmic process manipulates input into a modified or expanded output. If we feel compelled to state an explicit ideal standard of transformation in a generative music system, it might be what Eno called "music for free, in a sense": A system by which we can produce infinite, meaningfully different output from finite input. [1]

To evaluate Audion with respect to the ways it transforms input, we could pose two questions. First, how different would the output of the system be between uses if the initial system parameters were the same? Second, how different can the produced outputs be? In answering these two questions, we will set aside the choice of the audio sampled for grains, and instead discuss Audion's transformation of sound in more functional terms. If we were to discuss the system's capability for transformation while also considering the sampled audio, then the answers to both questions become an unsatisfying infinitude—a user could choose any sound.

If we were to instantiate the system several times with the same user-selected parameters and let it run with a uniform configuration of user-selected objects and audio files, what we might quickly see is that the transformation applied to sounds is highly dependent on user interaction. As a result, the transformations of the system in isolation are quite limited. The initial parameters of a trial's first birds are produced by a random-number generator in the software's underlying game engine. If we run the system without any form of user interaction, only the stochastic variables in the system—the placement of food from a feeder object, the parameters of the birds spawned at startup, or the birds eliminated by a plague object—influence the transformation. However, de facto measures of fitness that are in place mean that birds with traits such as large fields of vision or faster maximum speeds will, over time, drive the proliferation of those traits in the population as a whole.

Another contributing factor to the limitations of the system's transformation is the implementation of limits on birds' attributes. While individual birds have a maximum speed, for instance, a system limit is implemented which keeps any bird, no matter how many generations unfold, below a certain speed threshold. These limits were intended to keep the system usable, but their ultimate effect is that the birds tend to converge toward their hard-coded maximum values. This convergence means that the transformation of sound by the system in necessarily linked to user interaction, which reintroduces randomness into the system.

The differences in outputs are less a function of the simulation than a function of the user's manipulation of the environment. As we have previously indicated, there are parameters of the sound that are explicitly mapped to certain actions by birds or measures of the environment. While the ways in which a sound can be manipulated are limited by the hard-coded connections between environment and sound qualities, the precise levels of each parameter can be manipulated based on the knowledge of the user.

The effect of a system that tends to produce uniform results when left alone and more varied output with user skill might then be a more direct comparison to a musical instrument. However, a more successful iteration of Audion might enable a high degree of variability in its isolated transformations while enabling more skilled navigation of the system.

Translation of Emergent Structures

The choices of granular synthesis and an agent-based simulation were motivated by the idea that both forms express emergent structures. The various timescales of music that Roads defined were intended to be counterparts to the metaphorical events in the Audion environment. [40] At the level or grains operating on the micro timescale, we have the discrete, single action of a bird choosing to eat, flee or mate—but the meso and macro timescales are less clearly connected to the structure of the simulation.

The audio files selected for sampling into grains could be said to correspond to a macro timescale: As grains are triggered for any particular object, the position in the audio file from which grains are selected is advanced forward in time. However, depending on the choices of the user and parameters of the simulation, it might be more appropriate to view the progression of audio files or the life cycles of birds on the meso time scale. A version of the simulation that continues through several repetitions of an audio or a user that modifies objects frequently would effectively be creating constituent phrases of sounds, rather than guiding a broad structure of the output.

It also becomes challenging to define these two larger timescales without a context for the presentation of Audion's output. This is especially true if we view Audion as akin to more conventional instruments. The banjo, for instance, does not have any embedded notion of musical structure with respect to time. Rather, a banjo is used to present music of a structure that has been determined by its player or a composer. That structure could be the indefinite strumming of a random sequence of notes, or a jig.

What Audion might lack, then, is the option for its user to choose to impart structures on the macro and meso time scales into the system. While the birds have goals to achieve, states to motivate their actions and lifespans for which they inhabit the simulation, the birds also exist within a linear progression of time under uniform pressure. There is no context for time in the simulation. Contrary to the ways in which Truax and Keller imparted structure to granular compositions by modeling natural events, Audion places responsibility for structure on the actions of a user. [40] An advantage of Audion's open-ended approach is that it becomes suited for live experimentation, however, without a mechanism for imparting structure beyond the micro timescale, we perhaps lose some of the system's potential by forcing the user to accept real-time interaction at every level.

Another structural limitation might come from the simulation itself. While a user can adjust the objects in an environment to control individual grains, the lack of additional pressure within the simulation leads to uniformities that might hinder the translation of the simulation's emergent properties into meaningful structure to the output. A predator agent for the birds, for instance, could lead to meaningful rises and falls in populations or the system rewarding agents that give highest priority to their state of fear.

We might also consider the use of the sampled audio itself as a source of structure for proposed environmental pressures. The amplitude envelope of an audio file could be used to structure the mating patterns of birds, for instance. While this particular example is quite arbitrary, we might consider this idea framed within our earliest notion of musical "seeds": a single, small piece of information being transformed and translated into many, many permutations.

VI. Conclusion

Audion is an attempt to create a system that is distinct from conventional instruments by disrupting the connection between a particular user input and a predictable output. While there are a plethora of historical and contemporary examples of generative systems, Audion is distinguished by a focus on real-time exploration, the use of granular synthesis, and a mode of interaction based on manipulating the environment of an ecologically inspired simulation.

Audion's development began with a series of experiments focused on translating the activity of agent-based systems into a musical tool. Agent-based systems were chosen because of their ability to transform relatively little user input into high levels of output, and they do so while potentially offering emergent structures that are useful for musical expression. The use of state-motivated agents with goals independent of the musician's goals presents an interesting level of interactivity. The incorporation of granular synthesis was inspired by the work of Truax and Keller, and the writing of Roads, which drew connections between units of sound and various timescales of musical information.

In order to produce desirable results in the realm of music, a user must also understand Audion's simulation. By disrupting the execution of a musical idea, we attempt to invite the musician using Audion to correct, incorporate or oppose the system's output. This co-evolution is the crux of the system.

The focus of further development should become the clarity between user action and system behavior, so that the user can make more informed choices to affect the audio output. Additionally, the system would benefit from more precise manipulation of timescales. As an open-ended instrument, Audion can be effective at producing interesting sounds, but the use of the emergent structures of the system's agents is underutilized.

Acknowledgments

I would like to presonally thank Edwin van der Heide and Rob Saunders for their guidance; Zach Barr and Jae Perris for offering valuable feedback and volunteering their time as critics; and my partner Kees, and my family and friends for their loving support during this project.

References

[#]	Source
[1]	Brian Eno, "Generative Music: A Talk Delivered in San Francisco, June 1996." *In Motion Magazine* (July 1996).
[2]	Brian Eno, *Ambient 1: Music for Airports.* PVC (1978).
[3]	John Lysaker, "Turning Listening Inside Out: Brian Eno's Ambient 1: Music for Airports." *The Journal of Speculative Philosophy* 31 No. 1 (2017): 155-176.
[4]	User liambdonegan01, "Conway's Game of Life." Wikipedia. Accessed 5 March 2020.
[5]	Nick Collins, "The Analysis of Generative Music Programs." Organised Sound 13 No. 3 (2008): 237-248.
[6]	Peter Pesic, *Music and the Making of Modern Science.* Cambridge,

Massachusetts: The MIT Press (2014).

[7] Fadlou Shehadi, *Philosophies of Music in Medieval Islam*. Leiden, The Netherlands: E.J. Brill (1995).

[8] Jaime Serquera, Eduardo Reck Miranda, "Histogram Mapping Synthesis: Cellular Automata-based Technique for Flexible Sound Design." *Computer Music Journal* 38 No. 4 (Winter 2014): 38-52.

[9] Miroslav Spasov, "Music Composition as an Act of Cognition: ENACTIV—Interactive Multi-Modal Composing System." *Organised Sound* 16 No. 1 (2011): 69-86.

[10] CIRMMT, "Miller Puckette - Design choices for computer instruments and computer compositional tools." YouTube. Video. Accessed 20 July 2020.

[11] Johnathan Swift, *Gulliver's Travels*. Project Gutenberg website. Accessed 12 Nov. 2019.

[12] Eleanor Selfridge-Field, "Composition, Combinatorics and Simulation: A Historical and Philosophical Enquiry." *Virtual Music: Computer Synthesis of Musical Style*. Ed. David Cope (2001).

[13] Eric Weisstein, "Combinatorics." MathWorld—A Wolfram Web Resource. Accessed 15 April 2020.

[14] Karlheinz Essl, "Algorithmic Composition." *The Cambridge Companion to Electronic Music*. Ed. Nick Collins (2007).

[15] Tero Parvianinen, "How Generative Music Works: A Perspective." Blog. Accessed 4 March 2020.

[16] Allan Kozinn, "A Classic Minimalist Score, Played at Maximal (and Electronical) Length." *The New York Times*. 10 Nov. 2009.

[17] Tom Huizenga, "Fifty Years of Steve Reich's 'It's Gonna Rain.'" National Public Radio website. Accessed 17 March 2020.

[18] Terry Riley, Ictus, Blindman Kwartet, *In C: Terry Riley*. Cyprès Records: 2000.

[19] Terry Riley, Members of the Center of the Creative and Performing Arts in the State University of New York at Buffalo, *Terry Riley: In C*. CBS Records: 1968.

[20] Charles Ames, "The Markov Process as a Compositional Model: A Survey and Tutorial." *Leonardo* 22 (1989): 175-187.

[21] "Imaginary Landscape No. 4 (March No. 2)." The John Cage Trust website. Accessed 6 March 2020.

[22] Dorien Herremans, Ching-Hua Chan, Elaine Chew, "A Functional Taxonomy of Music-generation Systems." *ACM Computing Surveys* 50 No. 5 (Sept. 2017).

[23] Joel E. Cohen, "Information Theory and Music." *Behavioral Science* 7 No. 2

(1962): 137-164.

[24] Sebastian Macchia, “Making an Album with Music Transformer.” Magenta blog. Accessed 8 March 2020.

[25] David Temperly, *Music and Probability*. Cambridge, Massachusetts: The MIT Press (2007).

[26] Nzamul Siddique, Jojja Adeli, “Nature-inspired Computing: An Overview and Some Future Directions.” *Cognitive Computing* No. 7 (2015): 706-714.

[27] Jan C. Schacher, Daniel Bisig, Philippe Kocher. “The Map and the Flock: Emergence in Mapping with Swarm Algorithms.” *Computer Music Journal* 38 No. 3 (Fall 2014): 49-63.

[28] Maximos A. Kaliakatsos-Papakostas, Andreas Floros, Michael N. Vrahatis, "Interactive Music Composition Driven by Feature Evolution.” *SpringerPlus* No. 5 (2016).

[29] Daphne Oram, *An Individual Note of Music, Sound and Electronics*. London: Galliard Ltd. (1972).

[30] John McCormack, “Creative Ecosystems.” *Computers and Creativity*. Ed. John McCormack and Mark d'Inverno (2012).

[31] Robin Gras, Didier Devaurs, Adrianna Wozniak, Adam Aspinall. “An Individual-based Evolving Predator-prey Ecosystem Using a Fuzzy Cognitive Map as the Behavior Model.” *Artificial Life* 15 No. 4 (2009): 423-463.

[32] Gordon Pask, “A Comment, a Case History and a Plan.” *Cybernetic Serendipity*. Ed. Jasia Reichardt (1970).

[33] Paul Nagle, "Yamaha Tenori-On." *Sound on Sound* (Feb. 2008).

[34] Rich Lem, “Tenori-On.” Wikipedia. Accessed 8 June 2020.

[35] LÖVE, love2d.org.

[36] Pure Data, puredata.info.

[37] Damian Keller, Barry Truax, “Ecologically Based Granular Synthesis.” International Computer Music Conference Proceedings (1998).

[38] Damian Keller, "Compositional Processes from an Ecological Perspective." *Leonardo Music Journal* 10 (2000): 55-60.

[39] Barry Truax, Riverrun. Simon Fraser University website. Accessed 10 June 2020.

[40] Curtis Roads, *Microsound*. Cambridge, Massachusetts: The MIT Press (2001).