Meta-Xenakis

New Perspectives on Iannis Xenakis's Life, Work, and Legacies

Edited by Sharon Kanach and Peter Nelson

https://www.openbookpublishers.com

Sharon Kanach and Peter Nelson (eds), *Meta-Xenakis: New Perspectives on Iannis Xenakis's Life, Work, and Legacies*. Cambridge, UK: Open Book Publishers, 2024, https://doi.org/10.11647/OBP.0390

Digital material and resources associated with this volume are available at https://doi.org/10.11647/OBP.0390#resources

ISBN Paperback: 978-1-80511-224-2
ISBN Hardback: 978-1-80511-225-9
ISBN Digital (PDF): 978-1-80511-226-6
ISBN Digital ebook (epub): 978-1-80511-227-3
ISBN HTML: 978-1-80511-229-7
DOI: 10.11647/OBP.0390

Cover image: Iannis Xenakis at the C.R. MacIntosh Museum, Glasgow, Scotland, 1987. Photo by Henning Lohner, courtesy of CIX Archives, Lohner collection.

Cover design: Jeevanjot Kaur Nagpal

With generous support from: Centre Iannis Xenakis, Xenakis Project of the Americas/The Brook Center, Université de Rouen: BQRI, IRIHS, GRHis, CÉRÉdI.

39. The Process of Creating a Computational System: A Collective Audiovisual Composition

Hugo Solís, Mizky Bernal, Diego Jiménez, Guillermo Leonardini, and Eunice Pérez

Introduction

In the year 2021, the Music Technology Group of SUICREA (University Seminar for Research in Artistic Creation at the National Autonomous University of Mexico) was created as a pedagogical platform focused on the exchange of computational ideas and collaborative work. Directed by Dr. Hugo Solís, this space aims to imagine and seek creative solutions in the field of music technology and computation from a practical perspective, where the sonic imagination guides the developmental processes and computational implementations.

In 2021, the team developed a web application that emulates, in a simplified form, the original UPIC system of Iannis Xenakis. Subsequently, in 2022, instead of expanding and detailing the system, the group decided to create an audiovisual web application that combined the compositional concerns and interests of each of the course members into a single project. The unifying element of the proposals was the integration of autonomous agents that resulted in the use of the "flocking behavior" proposed by Craig Reynolds.[1] This flocking behavior is directly related to the approaches and concepts of agency, intelligences, and emergent phenomena. Another original element of the system is the correlation mechanics between the 3D geometric space and the sound parameters, since the latter are derived from free projections of the rotation of this geometric space.

Finally, a model of synchronization and collective organization of the rhythmic pulse was implemented based on experiments and research related to synchronization

1 Reynolds, 1987.

 https://doi.org/10.11647/OBP.0390.41

in groups of agents including Yoshiki Kuramoto's (b. 1940) model, among others.[2] The behaviors and visual actions give rise to a reading of parameters that controls a bank of frequency modulation (FM) oscillators that, as a whole, generate the sound complexity and the evolution of the work.

Background

SUICREA's Music Technology Group was created within Music Creation Laboratory (LaCreMus) under the direction of Julio Estrada (b. 1943), who suggested the creation of a space for computational research. Since this initiative arose during the times of the pandemic, this work team integrated a Latin American community, and allowed its members to be located in Mexico and Bolivia. This made it possible for the group to work online throughout the process.

In 2021, the group proposed their first group project: to create a simplified version of Xenakis's UPIC (within the Replit platform for writing code on the web).[3] We implemented this through a programming platform with audiovisual capabilities that we called UpicMX.[4] The purpose of this first project was to introduce the teaching of the programming language for audiovisual purposes; the aim was to create a graphical sound system to learn the basics of computing.

Xenakis devised a computerized music creation tool: UPIC (Unité Polyagogique et Informatique du CEMAMu), developed at the Center for Studies in Mathematics and Musical Automatics (CEMAMu) based in Paris, France, the first prototype of which was completed in 1977. The UPIC was a system consisting of a processor with a digital vector display, in the form of a drawing table, on which music was "drawn." With the help of an electronic pen, a curve or series of curves were drawn, which the machine would convert into sound; the X plane corresponded to time, and the Y plane to pitch. By means of controls on the board, some other variables such as register and intensity could be defined; a television screen reproduced the drawings, while the resulting music was listened to by means of amplifiers after having been processed in a synthesizer.[5]

The UPIC opened enormous possibilities for musical creation and revolutionized the field of electronic music, implying a paradigm shift in drawing music: it created possibilities for creative impulse and continuous movement, overcoming the previous restrictions of a fragmented musical thought that was largely based on a limited writing. The new system was an innovation because it also broadened the perspective of music in close relation to other disciplines such as geometry and its spatiality,

2 Kuramoto, 1975. Explained in "Synchronization," *Ichi*, https://ichi.pro/es/sincronizacion-la-danza-sama-de-la-naturaleza-20453629169086

3 *Replit*, https://replit.com/

4 *Soundflock*, https://soundflock.site/

5 For a fuller description of UPIC see Marino et al., 1993.

mathematics, drawing, and electronics.

The UpicMX project was successful in achieving two different programming solutions to convert traces into sound. The first solution was "plastic sonification," which consists of a graphical approximation of the reading of the traces; the function obtains, according to a process of conversion, the time and frequency of the pixels of the drawing in order to assign each pixel an oscillator in real time. Each of the oscillators is superimposed on the next one for a fraction of a second, constantly generating an additive synthesis in frequencies so close that beats are produced that enrich the resulting timbre (see Code 39.1).

```
function sonoficarPlastica() {
  loadPixels(); // Loads all pixels in an array. Function of
p5
  var filasConNegro = []
  var frequencias = []
  //tiempoApixel converts the pixels into a time
relationship
  var timeActualPix = int(min(tiempoApixel() + 3, width));
  // The following code goes through all pixels column by
column
  // in each column it scrolls through all the rows and if a
  // pixel in black adds it to a list where the
  // coordinates with black pixels.
  for (var cont = 0; cont < height; cont++) {
    var pixCanalRojo = pixels[(cont * width + timeActualPix)
* 4 + 0];
    if (pixCanalRojo == 0) {
      filasConNegro.push(height - cont);
    }
  }
  //The frequency corresponding to each coordinate is
obtained.
  //that has black in frequency format.
  frequencias = filasConNegro.map(pixNegro => {
    let midi = map(pixNegro, 0, height, notaGrave,
notaAguda);
    //midi2freq converts the midi note to its corresponding
frequency
```

```
    return midi2freq(midi);
  });
  // For each frequency a 20 millisecond oscillator is
generated.
  for (var i = 0; i < frequencias.length; i++) {
    var osc = new p5.Oscillator(frequencias[i], 'sine');
    // amp, start and stop are functions of the webaudioapi.
    osc.amp(0);
    osc.start();
    osc.amp(0.1, 0.05);
    osc.amp(0, 0.1, 0.05);
    osc.stop(0.2);
  }
}
```

Code 39.1

The second type of "sonification," called "parametric," reads and stores the trace information in time and frequency parameters, then assigns it to a single oscillator per line that modulates its values over time using envelopes in its frequency and amplitude parameters. When the "touch" button is pressed, the oscillators are activated and change their frequency according to the movement of the base generator trace (see Code 39.2).

```
function sonificarParametrica() {
  destruyeOsc(); // Release all active generators with stop
  //del webaudioapi.
  osciladores = [];
  // compositionTF is an array of arrays in which
  //each internal array contains the stroke points.
  for (var i = 0; i < composicionTF.length; i++) {
    var gestoTMP = composicionTF[i];
    var osc = new p5.Oscillator(composicionTF[i][1][0],
'sine');
    osc.start(composicionTF[i][0][0]);
    // A single oscillator is generated whose frequency
changes
    // discretely but at a speed that gives us
```

```
    //a perceptual glissando.
    osc.amp(0.0, 0, composicionTF[i][0][0]);
    osc.amp(0.1, 0.1, composicionTF[i][0][0]);
    // nesting for that goes through each point of each
stroke and assigns
    //frequency at the corresponding moment (time).
    for (var j = 1; j < gestoTMP[0].length; j++) {
      var tTmp = composicionTF[i][0][j];
      var fTmp = composicionTF[i][1][j];
      var tTmpAnterior = composicionTF[i][0][j - 1];
      osc.freq(fTmp, tTmp - tTmpAnterior, tTmpAnterior);
    }
    // On the last point of each stroke the volume is turned
down and off.
    var indxtUltimo = gestoTMP[0].length - 1;
    osc.amp(0, 0.1, gestoTMP[0][indxtUltimo] - 0.1);
    osc.stop(gestoTMP[0][indxtUltimo]);
    // All oscillators are added to one array
    //to be able to turn them off in another external
function.
    osciladores.push(osc);
  }
}
```

Code 39.2

Another feature of the project is the incorporation of a grid in both the X and Y axes that can be activated or deactivated, depending on the accuracy with which you want to manage the times and heights (pitches) in the traces.

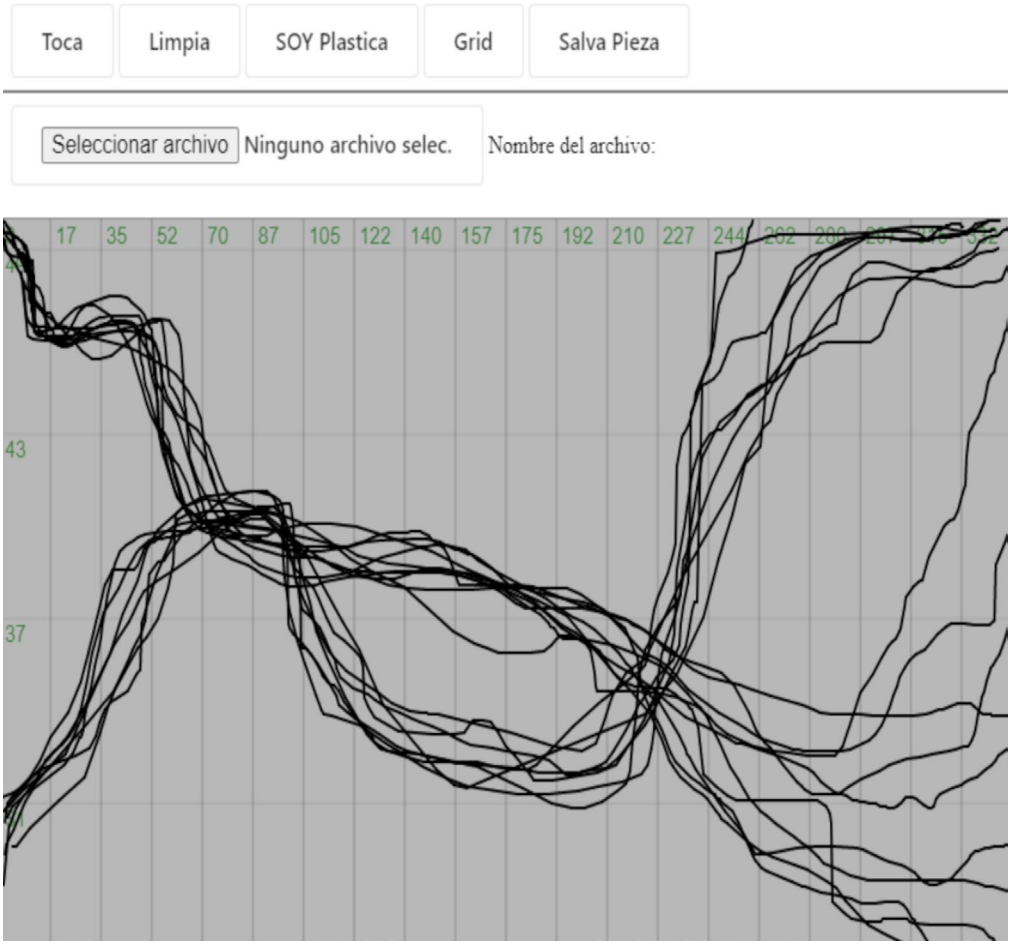

Fig. 39.1 Graphical representation where X is time and Y is frequency. Figure created by authors (2022).

Derived from this first exploration, a second project was generated throughout 2022, which we named Sound Flock. We will now jointly present these projects below.

The Collective Imaginary as a Triggering Process

In January 2022, a meeting was held within the course to propose the characteristics of the project that would be developed throughout that year. This project was the conjunction of the individual imaginations and ideas that interested the members of the course, which were unified to produce a single project integrating the points in common while preserving the particularities of the students' proposals. Each member of the team proposed specific needs, both in the field of sound and in the field of usability, user interface, possibilities, and control ranges. Throughout this

stage, the common needs were amalgamated, and the individual particularities were incorporated into the resulting product, leaving out some non-compatible elements. Among the common ideas, the following stand out:

1. The design of a system where the user could interact and modify parameters of the environment—in this way, the user is both interpreter and spectator of the work.[6]
2. The design of a 3D space where there were moving objects (agents).
3. The idea that the behavior of these objects/agents would concretely imitate the movement of fireflies, which resulted in two important characteristics (items 4 and 5 below).
4. These agents had to follow certain patterns/interactions when moving, which have to do with flocking behavior where, broadly speaking, the individuals of the group have reciprocal influences among themselves without any hierarchies.
5. The agents would have an individual light pulse where the flash of each one would seem independent of the others at first, but after a certain amount of time, a synchronization of all of the light pulses would be provoked.

Within the 3D space, the principle that the absolute interpretation of the positions of the agents is accompanied by a reading from any viewing angle, so that the user may rotate and move within the 3D environment at will. It is important to note that the user may also activate planes from any point in space, moving towards the background.

Consequent on (4) above, different simultaneous readings can be made of the behavior of the agents. These planes advance on their relative Z axis, which corresponds to time—the transit speed of which can be defined by the user—and where the X and Y axes, also relative to each plane, correspond to different parameters of the rhythm/sound, also defined by the user. Since the parameters respond to the position of the space, by changing the reading point, it is possible to generate multiple, ever-changing versions of the sound creation.

These five characteristics gave birth and shape to the entire project and guided the decision making and computer code development over the following months, resulting in the Sound Flock project, which is described below (see Figure 39.2).[7]

6 For the sake of clarity, we use the term "user" despite its possible reductionist connotation. However, we think that, given the nature of the platform, the person using the tool will be at different levels a creator, performer, player, explorer, researcher and interpreter.

7 The functional version can be found at https://soundflock.site/. The source code can be found at "hugosoli/soundFlock," *GitHub*, https://github.com/hugosoli/soundflock

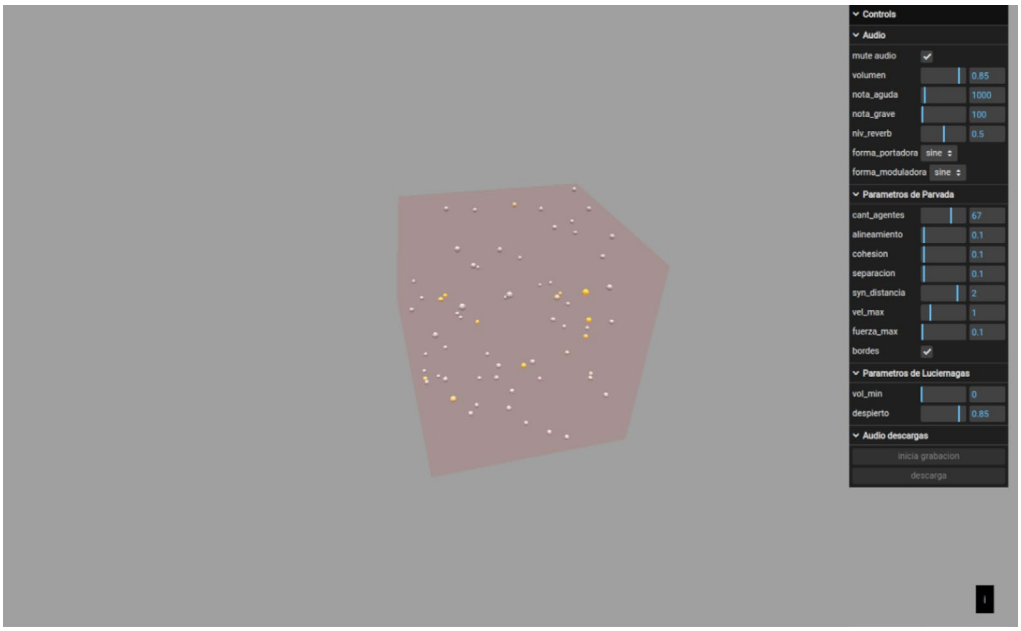

Fig. 39.2 Screenshot of the system, the sand-colored and orange dots correspond to the agents. Figure created by authors (2022).

Development of the Work

The development of the work was guided by the vision raised in the investigation with the aforementioned factors: agentiality, flock behavior, and interference of planes on the space. The general purpose was to have an audiovisual experience in which the events, and their ranges and parameters, were manipulated by means of a control panel. In this way, a resultant was planned that could produce "compositions" to be downloaded, with the objective of preserving them in fixed format for their future and possible diffusion.

In order to develop and achieve the proposed objectives, it was essential to work step by step with the following elements:

1. Define the 3D space.

The first decision was to establish a 3D space, which we called "external space," which corresponds to the Cartesian absolute coordinates of the digital graphic system. Within this virtual space, a cube was placed whose function is to be the container for the graphical representation of the agents.

2. Characteristics of the agents.

During this stage, work was done on the graphic representation of the agents and the following was determined:

a. That they should be confined inside the cube, in a spherical representation where the agents have the capacity to flash with cyclic pulses as real fireflies do.

b. That they should demonstrate a capacity of movement in three dimensions with different degrees of randomness.

3. Agent behavior.

The main objective at this stage was to get the agents to have a flocking behavior. Several examples of this movement were studied in order to adapt them to the goals of our proposal. To organize the computational code, several classes were designed, among which the Boid class and the Flock class stand out.[8] The first class contains all the behaviors of each agent, and the second one groups the collection of agents. Within these two classes it is possible to completely define the behavior that the agents would have, clarifying that their movement is given through acceleration, position, velocity, attraction, and distance between all of them.

Sound Behavior

At this stage, it was decided that each of the agents would emit a sound generated through Frequency Modulation synthesis. The sound process was linked to the behavior of the agents in such a way that generated an association between the parameters of the virtual physical space and the sound parameters. The modulation index was associated with the position on the Z axis, the intensity with the position on the Y axis, the intensity with the position on the X axis. It is important to mention that the intensity not only depends on the position but is also modulated by the cyclic pulsations of the flashes.[9]

1. Sound process recording.

During the development of the project, the importance of making recordings of the sound actions for their possible preservation and/or editing became clear. Therefore, the necessary code was implemented to generate files in WAV format; to achieve this, codes found on the web that took advantage of the Web Audio API for recording digital files were adapted.

2. Sound capture by intersecting planes.

At this stage, a mechanism of reading by means of blueprints was developed in relation to the position of the agents. This reading is generated when the plane crosses the cube, reading the relative position of each of the agents, adding a new sound.

8 We refer to the concept of Class, within object-oriented programming (in English OOP).
9 Estrada, 2001.

This procedure was performed by identifying the relative coordinates of each agent in space.

3. Control panel.

Within the original conception, one of the goals was to allow the user to interact with the system and thus achieve a more interactive experience. With this objective in mind, a control panel was implemented that allows the manipulation of a large number of parameters and ranges that help to control the sound results.

Emerging Behaviors

Flocking Behavior

Flocking behavior is present in groups of small individuals: it is most commonly found in flocks of birds, schools of fish, groups of insects, etc. Emergent behavior is considered to be the phenomenon that arises from the decisions made by each individual following simple rules, where the most important are the relationships of each individual with its environment. There are many works that have simulated this type of behavior. The one that has been followed as a model for the project is the work of Craig Reynolds, implemented in the P5.js library by Daniel Shiffman.[10] During the course, this code was adapted to the Three.js library for modeling in three dimensions.[11]

In this model, an agent-object is created, named "Boid," which has the following properties: a velocity and acceleration, a maximum velocity and force, a pulsation velocity together with its phase, and a "step" which consists of the amount that its phase is advanced.[12]

As stated, flocking behavior is defined by the "individual decisions" of each agent where such decisions are defined with three functions: separation, alignment, and cohesion, as established by Reynolds.

The separation function consists of how agents avoid colliding with each other, avoiding and pushing each other according to a given range. The function has two important local variables: "desiredseparation," which consists of the level of separation of the agent from the rest (in other words, this impulse indicates what the range of action of the function is), and "steer," which indicates a vector with the future direction and position of the agent. To know which of the agents are in the range of action of the function, the distance between each agent and the rest must be calculated. If the

10 "processing/p5.js," *Github*, https://github.com/processing/p5.js; "The Nature of Code" (2024), *Github*, https://github.com/nature-of-code/book-website-2nd-edition

11 "mrdoob/three.js," *Github*, https://github.com/mrdoob/three.js/

12 In the following paragraphs sections of the source code are explained, so it is suggested to accompany the reading with a review of the source code.

distance is greater than the "desiredseparation" then three operations are applied: the vector of the agent must be subtracted from the vector of the nearby agent, the subtraction normalized, and then divided scalarly with the value of the distance. Finally, this value is added to the variable "steer." This process is repeated with each of the agents, as long as their distance from the agent is less than the "desiredseparation."

The second part of the function consists of dividing the overall "steer" by the number of times that the previous operation was performed; i.e., normalize the value of "steer," and finally adapt this value to the general properties of the agents (in other words, adapt to the maximum speed, the current speed, and the maximum force). Thus, we finally have the "steer" to be applied to the agent in question.

The alignment function instructs each agent to change its direction of movement according to the position of the rest of the agents, causing groups of agents to move in the same direction. Like the separation function, the alignment function has two important variables: "neighbordist" and "sum." "Neighbordist," is the distance at which the function will come into effect, and "sum" is a vector that will indicate the direction and final position of the agent. For each agent within the range of "neighbordist," the current velocity of each of them will be added to "sum."

The second part of the function consists of taking the value of "sum" (the sum of all velocities) and dividing it by how many agents are within the range of action of the function; i.e., normalize such division and adapt it to the general properties of the agent and, finally, return the value of "sum." In case there is no agent within the range of action, a vector with values of 0 in each axis must be returned. In this way, the behavior of the agent will not be affected in any way.

The cohesion function acts on the agents to calculate the average position of nearby agents and instructs them to move to the average position. Like the separation and alignment functions, the cohesion function has two parts: in the first part, a variable (again defined as "sum") is obtained that sums all the positions within the action range (as in alignment, the action range is defined by the variable "neighbordist"). In the second part, this sum is divided into the number of agents within the "neighbordist." Before returning the value of "sum," the secondary function "seek" must be applied, which adapts the value to obtain the final result.

Finally, these three functions are added together with the "applyForce" function, to indicate to each agent its next position in each interaction of the animation. In this way, one has a new position that summarizes how separated and aligned the agent will be with respect to the others and if its direction has cohesion with the rest.

According to different configurations of these functions, movements will be obtained that are either more synchronous or more chaotic. It should be noted that some functions can override the effect of others; for example, in the case of alignment with separation, a high level of separation and cohesion will generate a hesitant movement of the agents.

Synchronization

Two different strategies were used to achieve synchronization: pulse synchronization and phase synchronization. For this purpose, two different functions were used: "bpmsync(),"which is in charge of synchronizing the pulses, and "syncronization(),"which is in charge of synchronizing the phases (see code 39.3).

In the "pmsync()" function, there is a local variable named "sigma," which is the sum of the differences between the sine of the variable "step" of each of the agents. Then we proceed to normalize "step" where the value is multiplied with a predetermined value that controls the speed of synchronization. Kuramoto proposes that each agent listens to the rest of the nearby agents when they press and, therefore, each time any agent presses, the rest of the nearby agents will slightly advance the speed of their pulse, being affected by the collectivity.[13]

On the other hand, in the function "syncronization(),"a counter was assigned to each agent with the variable "step," which generates cycles between 0 and 1. If the "step" state is less than 1, then the firefly has an OFF state; on the contrary, when "step" is equal to 1, the firefly's state is AWAKE, generating a light and timbre pulse. In turn, if "step" is equal to 1, then each phase of each agent will be advanced by a small amount the variable "step;" that is, every time any agent presses, the other nearby agents will begin to synchronize, equalizing their phases with each other.[14]

This procedure helps to ensure that the groups that are close in space gradually grow closer in their phases; that is to say, are constantly synchronizing with each other. It is possible to control the number of agents, and also how far away they can sense other agents in their "flashes." With these variants, it is possible to accelerate the synchronization process or, on the contrary, to generate asynchronous pulses independent of each other (see Code 39.3).[15]

```
bpmsync(boids) { //Kuramoto
  let sigma = 0;
  for (let i = 0; i < boids.length; i++) {
    sigma += Math.sin(boids[i].paso - this.paso);
  }
  let term2 = 1 / boids.length * sigma;
  this.paso += term2 * 0.001; // Value that reduces the
speed at which synchronization is done.
 }
```

13 See Nicky Case, *Fireflies*, https://ncase.me/fireflies/
14 Ibid.
15 Ibid.

```
  syncronization(boids) {
    if (this.estado == "DESPIERTO") {
      for (let i = 0; i < boids.length; i++) {
        let d = this.position.distanceTo(boids[i].position);
        // If the distance is greater than 0 and less than
an arbitrary amount (0 when you are yourself).
        if ((d > 0) && (d < SYNC_DISTANCIA)) {
          if (boids[i].estado == "DESPIERTO") {
            this.fase += this.paso;
          }
        }
      }
    }
  }
```

Code 39.3

Individual Comments

Mizky Bernal, Bolivia

I recently started to explore computer music creation. This exploration allowed me to learn technical aspects of programming languages, the study of the possible tools provided by the implementation of programs, and the scope of working in the various specialized virtual platforms—all within the framework of the possibilities and limitations that I still find because of my limited training in this area.

This approach made me consider everything that the subject produces in my person from a creative perspective. From this reflection I understood that my creative approach could be modified at certain points, since the tools provided by technology are quite broad. That is how I began to speculate about the possibilities of sound control, the incursion into multidisciplinarity, the work on exploratory listening, possible audiovisual inquiries, the manipulation on semiological-musical management, and many other interesting elements. Undoubtedly, the development of this course boosted my curiosity towards the exploration of the virtual world.

Diego Jiménez, Mexico

Throughout the course I was able to realize the creative and technical possibilities of sound creation by computer, as well as the benefits of programming in general, both

to organize my thoughts and to enrich my creative process. This type of creation helps to structure thinking, by the very nature of programming languages and by the need to give the computer clear and precise indications about what it has to execute. This puts the programmer in an uncomfortable situation, because it forces him to specify absolutely everything that the machine—which does not have the capacity to make decisions—needs to do; no matter how basic or logical the idea is, it must be described. Therefore, it is necessary that all instructions be given in a conscious way in order to be declared. On the other hand, computer creation requires a high degree of imagination and organization, since there cannot be an answer without a question and, therefore, it is not possible to have a direction without a problem to solve. You must first imagine, in order to understand the problems and implications of what you want to do, and then design possible solutions to achieve that goal.

As for my artistic work, the creation of sound by computer has implied a significant enrichment since it has forced me to consider paradigm changes. For example, I now treat the computer as just another instrument or as an extension of traditional instruments. This allows me to materialize in a more reliable way what I need. On the other hand, programming and its field of action invites me to consider areas of knowledge that are not necessarily limited only to sound, so it can be a useful tool to create multidisciplinary works in which the artist's perceptive capacities are completed, without restricting them to a single sense.

Guillermo Leonardini, Bolivia

During the time that I have studied the subject of musical computing I have learned several things. The first one is that technology functions as a tool at the service of one's fantasies, instead of as a mere technical exercise, where the achievement of connecting some devices would be enough to justify the existence of the proposal. I find more interesting the position that technology, as a tool, allows a faster understanding of the phenomena studied and, therefore, drives creativity more effectively. On the other hand, I have learned that every imagination has a second process: that of translating the execution of the imagined into the real plane, to order that which is in the mind, to decipher the tools for its execution, to glimpse the steps to concretize it, and to achieve the construction—in the luthier's sense—of the material and the device to finally fulfil one's imagination. If something cannot be done, it can be codified, and it can be built.

The concept of work/tool within a digital platform is appealing to me. I had never worked on such a project before. One continually thinks about how the audience—or perhaps, in this case, the solitary virtual surfer—perceives the work; but one also thinks about how the composer operates the tool to get something to their liking. Experimenting with "the form of the work equals the utility of the tool" or "the final version of the work equals the audience's experimentation with the tool" were attractive approaches during the process. One never knows how far the audience will

take the proposal. Will it achieve what was imagined? Can one say that there can be a misuse of the tool? Delegating decisions to someone else is always a bit of a mystery.

Eunice Perez, Mexico

Programming languages are generally complex. They require a great capacity for abstraction, a gradual process of learning and constant practice, but, as rich as their complexity is, so is their scope. It is fascinating that the ideas and fantasies of a creator can take concrete form in the programs or works created through them. Programming works from the root, at the origin of sound objects, offers the ability to decide their acoustic essence and their behavior in space; it is a more radical process than working with pre-established objects such as musical instruments.

My artistic praxis is enriched by these tools and, at the same time, I find new paths to follow, not only in the field of audiovisual creation, but also in the area of artistic research, as a means of auditory or visual exploration. Electronic media offer possibilities for handling information and sound and visual objects that are alien to the analogical world. It is possible to transform or move them at will, which facilitates the study of the researcher-artist's personal interests. The collective creation of a program such as the one we have worked on in the two projects within the Music Technology Group is a way of sharing processes, concerns, and interests with colleagues.

Hugo Solís, Mexico

Guiding a process of technological creation under the premise that the most important thing is to give materiality to a development of the collective imagination is a challenge and, in itself, a process of creativity. Questions such as where to put the emphasis, whether in the technological development or in the sound result, in the learning process or in the pristine result, are solved in the moment, during each session. The trade-offs and decisions were made on a meeting-by-meeting basis and, as a whole, were influenced by the profile of the group. As a supporter of "learning by doing," the methodology of teaching programming along with the creation of a specific project has its particularities and challenges. However, this is how we humans acquire language, and it was in this sense that I tried to guide each of the sessions. We learned through listening, repetition and, above all, under the impulse and the need to communicate. It is under this premise that, over two semesters, the process of designing first a web version of the UPIC, and later a unique approach to collective audiovisual creation was carried out. The first semester functioned as a stage to learn the basic principles of computer programming where concepts such as variables, arrays, cycles, functions, and logical operators were gradually used, as they were required within the code designed in a multi-user platform that allowed everyone, to view and manipulate a shared code from their computers. During the second semester, assuming that the basic principles

had been assimilated, we had the opportunity to delve deeper into various concepts, at creating new code and, being able to review the open-source code that, in this case, offered us parts of the solution to implementation. However, the nature of such codes required specific adaptation or translation from another computer language to JavaScript. On certain occasions, it was necessary to make sketches or code fragments in other computer languages including PureData's graphical language. At a certain point, when it was decided to implement a recording system within the application, it was necessary to dedicate several sessions to the theory of signal processing and the theory of digital sound synthesis, in addition to mentioning topics such as Frequency Modulation synthesis. I must confess that some of the processes were smoother than others; one that took a significant amount of time was writing the code that allowed us to perform the 2D projections of the 3D plane from a specific camera position view.

In relation to the distance learning dynamics, it is important to mention that half of the group is not in Mexico, so the course could not have been given in any other way. However, from a deeper point of view, I consider that teaching with the shared code platforms and the electronic whiteboards facilitated the teaching-learning process, given the predisposition of the group to maintain an active and attentive attitude during the sessions.

In this way, the result described in these pages was achieved: a digital system for audiovisual sound composition, created collectively. Undoubtedly, this result can grow and be refined in its details, but, at the same time, I celebrate that a functional system was achieved, derived from an intention of collective creation that served as a learning process. With this, I can say that we obtained a real, tangible audiovisual system, as well as a sound reality.

Sound Examples Available

The generated audio materials are available on SoundCloud,[16] where each of the materials presents different ranges of parameters translated into particular sonorities and evolutions. However, the reader-listener-receiver-user, above all, is invited to use the tool and to exploit it according to his/her own needs and under the impulse of his/her particular listening.

Conclusions

This chapter presents different perspectives on a collaborative process, a technological result, an audiovisual creation proposal, and a learning process mediated by remote distance learning. Undoubtedly, like any evolving digital object, the system is perfectible and improvable. However, perhaps it is born as a mutable, changing

16 Guillermo Leonardini, "console.log(ban.da[da])" (2022), *SoundCloud*, https://soundcloud.com/user-511292729/consolelogbandada

proposal in a constant state of transformation, and it is alive. That is to say, the system was not born to be finished, or to acquire such a static state; but rather remain a system in evolution and adaptation, a laboratory of changing sound experimentation where the subject changes. On the other hand, this version—with its very specific restrictions, possibilities, and particularities—is the reflection of a kind of serendipity, an aggregation of individual searches and needs put together, through sessions of discussion, reflection, and decision making, including elimination and negotiation of tasks within the limits of knowledge, time, resources, and computational creation skills. We must say, it is attractive that this proposal is the result of collective thinking and collective imagination. After the first semester was over, the possibility was raised for each member of the group to develop his or her own audiovisual solution; however, it would have would have been complicated to guide different ideas in parallel. It was in this way that the joint proposal started from each of the individual proposals—not from the sum of isolated ideas, but rather under the reflection on how to achieve their unification, their fusion, and the dialogue between them. Some elements of the individual visions were discarded, and others were incorporated once the proposal acquired unity and clarity.

In the territory of sound, we manage to control and listen to processes that are at a particular boundary, as they are neither deterministic, on the one hand, nor chaotic, on the other. The tool is at a point where control of the resulting sonorities is maintained, although several of the processes are not directly controlled by the user. At this stage the user can suggest ranges, dynamics, and interaction situations, but the particles have the same control over the resulting sonorities and a certain "creative independence."

What is presented here is the materialization of one possibility among many others. It is the sum of thoughts, needs, and curiosities of the teacher and the students who, in a collective search for a digital composition tool, arrived at a specific synthesis which allows a particular—and hopefully original—exploration for the creation of sound structures with the particularities described in this chapter. Therefore, this chapter is a description of a methodological work guided by the fantasies and curiosities of all the members of the group in an analogous way to the agents of the system.

Future Work

As part of the development and improvement of this project, several points are planned to achieve a greater sonic and perceptual richness and, thus, to grow and improve the users' experiences. Among the points to be implemented in the future, the following stand out:

1. Change the reading of the agent values on the axes X,Y, Z from the center to the ends to achieve a better sound balance.
2. Incorporate the binaural and/or multichannel audibility of the system, and consequently enable the download of a recording with such characteristics.

3. Clean and refine the reading of the plans on the 3D space.
4. To be able to change the waveform of the agents, as well as to incorporate the option of uploading an audio file to be used as a sample for sound generation.
5. Consider the possibility of incorporating not only a cube as a delimiter of the agents' space, but also other geometric figures.
6. To have the possibility of having delimiters within other delimiters, such as, for example, cubes inside cubes.

References

ESTRADA, Julio (2001), "The UPIC, from Xenakis to Its Alternative Development in MUSIC and 3D," *Academia.edu*, https://www.academia.edu/8313338/

KURAMOTO, Yoshiki (1975). "Self-Entrainment of a Population of Coupled Non-linear Oscillators," in H. Araki (ed.), *International Symposium on Mathematical Problems in Theoretical Physics*, Lecture Notes in Physics, 39, New York, Springer-Verlag, p. 420–2.

MARINO, Gérard, SERRA, Marie-Hélène, and RACZINSKI, Jean-Michel (1993), "The UPIC System: Origins and Innovations," *Perspectives of New Music*, vol. 31, no. 1, p. 258–69, https://doi.org/10.2307/833053

REYNOLDS, Craig (1987), *Computer Graphic*s, vol. 21, no. 4, July, p. 25–34.