

COMPOSING WITH IANNIX



JULIAN SCORDATO
info@julianscordato.com

Composer, sound artist and
music technologist

Lisbon
Computation
Communication
Aesthetics
& X

Abstract

This paper focuses on the usage of lanniX graphical sequencer for the creation of digital art. As currently in the written literature there are only few examples which address more or less extensively the creative processes in relation to lanniX (Scordato 2017), three case studies from my recent production will be examined throughout the paper. Each considered work exploits a specific functionality of lanniX. By providing technical and aesthetic information in relation to software programming, this article is intended as a useful resource to disclose features, possible strategies, and issues that other-wise would remain "trade secrets" of advanced software users of interaction experiences.

Keywords

lanniX
Graphical Sequencer
Digital Art
Multimodality
Interaction

1. INTRODUCTION

IanniX is a real-time “graphical open source sequencer for digital art” (“What is IanniX? | IanniX.”), based on Xenakis’s *UPIC* (Bourotte 2012). Through various communication protocols, it synchronizes punctual events as well as continuous data to external environments (e.g., Pure Data, Processing) and hardware such as MIDI devices and microcontroller boards.

Its graphical interface shows a representation of a multidimensional and multi-format score that is programmable via GUI, JavaScript, and third-party applications that use a compatible network protocol (OSC, Raw UDP, TCP, HTTP).

The interface is based on three types of abstract objects—triggers, curves and cursors—to be placed in a virtual three-dimensional space. Triggers and curves represent isolated events and spatial trajectories, respectively. Cursors are time-based elements (playheads) that move along curves in order to read a specific sequence of space-limited events. In this sense, IanniX proposes a three-dimensional and poly-temporal sequencer that runs independent of any audio synthesis engine. Therefore, it is suitable for a broad variety of applications.

For its capability of interfacing various media and devices, as it has been already clarified in the software documentation (Scordato 2017), IanniX represents a useful tool for digital artists, designers, and beyond.

2. IANNIX SCORES

In relation to their functionality, and to the interaction mode with third-party software and compatible devices, six types of scores have been recognized (Co-duys and Jacquemin 2014):

Control score. Autonomous, reproducible and determinist, once the control score has been set, the sequencer produces a data output—messages—for controlling further processes (e.g., sound synthesis, sampling and spatialization).

Reactive score. Reacting to external stimuli without generating any output of control data, the primary purposes of the reactive score are visualization and graphical representation of data received from programming environments and devices (e.g., a 3D trajectory detected by a motion capture device, or the deformation of a curve expressed by a parametric equation).

Generative score. Produced by algorithms written in JavaScript, the generative score can either cause the result to evolve over time or generate materials in a predetermined way; therefore, the script generates the score as output.

Stochastic score. Using JavaScript and third-party libraries, the user can create a stochastic score in which the global process is predictable while single events are aleatoric; even in this instance, the script produces the output.

Interactive score. Being based on human-machine or machine-machine interaction, the interactive score involves the cooperation between various entities. In this context, IanniX may act as a mapping device but also introduces compositional and temporal dimensions; a bidirectional data stream is involved.

Recursive score. IanniX can control itself, that is to say that the output related to an object can be reinserted into the system as an input command to control either the sequencer or another object; in some cases, this may imply feedback or deformation of the score as a function of time.

3. PROJECT REPORTS

Three recent works from my personal production will be examined. Each of them exploits a specific functionality of IanniX: in *Constellations* (graphical sequencer and electronics, 2014), electronic sounds are generated by a control score; in *Study for a network* (interactive audiovisual installation, 2015), an interactive score serves both as graphical user interface and control of the sound processing system; while *Pulsion X* (generative audiovisual art, 2016) is based on a reactive score that receives parameters for the generation of graphics in real-time.

3.1. Constellations (2014)

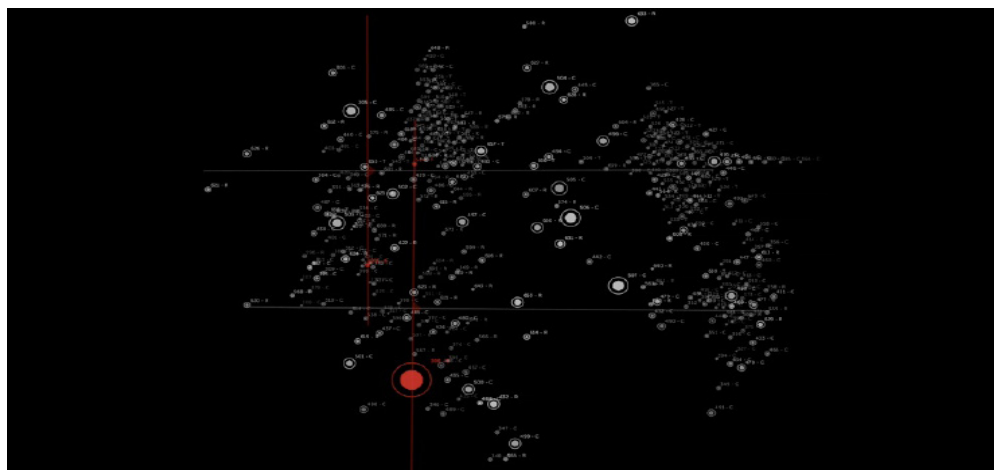
Constellations is a composition for graphical sequencer and electronics. It was created between March and April 2014. The world premiere occurred on October 1, 2014 at the CCRMA (Center for Computer Research in Music and Acoustics) of the Stanford University, in the context of their annual Transitions concert series. Further performances and screenings took place within numerous international festivals and exhibitions. Program notes have been provided as follows:

This work begins from the exploration of an imaginary celestial space, which is translated into sound space. How does each celestial sphere—starting from its manifestation as a unit—interact with the cosmos where it belongs? How does it react to its law? How does it transform itself integrating with the system, until the loss of identity? In contrast with that process, the constellations act underlining the bodies in their uniqueness by means of creation of symbolic links: beyond the sense, they stand as a classification and articulation device of the individual inside the system. (“Showcase | IanniX.”)

The composition is based on the reading of a IanniX score constituted by 356 punctual events and two drones. Due to such amount of information, a text report of the full score would be extremely redundant; however, the script has been generalized in Appendix A.

Fig. 1

Excerpt from the control score of *Constellations* (<https://youtu.be/dCVK-5wHX6To>).



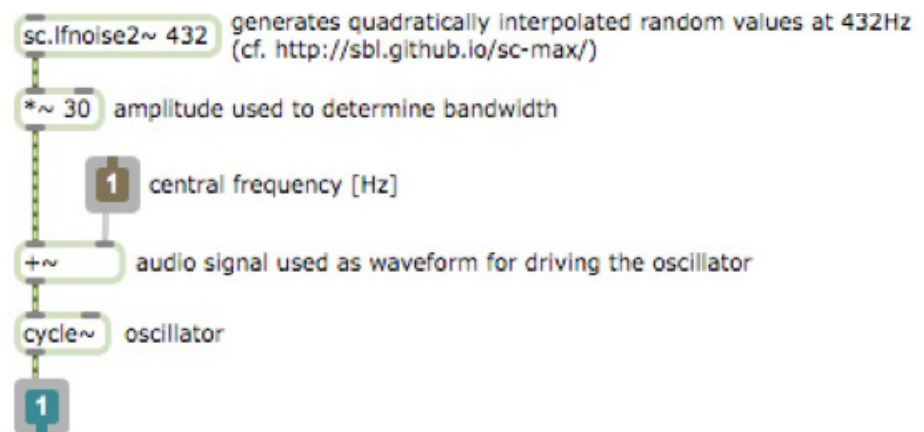
Two straight curves define a double reading path along the entire score, from left to right and vice versa; such reading technique can be considered like a sort of musical counterpoint. Cursors serve both as trigger activators and mes-

sage-emitting objects for controlling the central frequency [Hz] of band-limited noise; values are therefore transmitted via OSC protocol (Wright and Freed 1997) to Max/MSP programming environment. Similarly, through their messages, triggers convey a series of attributes for the definition of punctual sound events:

- trigger duration → sound duration [s];
- trigger position on Y axis → fundamental frequency [Hz];
- trigger group → linear amplitude (from 0 to 1);
- trigger label → timbre (preset);
- custom value → activation status of sustain (0 or 1).

All messages produced by the IanniX score are received in Max through the *udpreceive* object. Trigger values control the parameters of a simple additive synthesis ("MSP Tutorial 7: Additive Synthesis."), which has been included in a *~poly* object in order to allow for a polyphony of up to 30 voices ("MSP Tutorial 21: Using the *~poly* Object."); when the custom value is set to "1", individual synthesized sounds are processed by an algorithm that simulates a sustain effect: a decay curve proportional to the sound duration [s] is applied to the output of *freeze.pfft* external object (Charles 2008). Instead, mapped cursor values—related to *cursor_value_y* variable—control the following algorithm:

Fig. 2
Max/MSP implementation
for the generation of
band-limited noise.



Starting from these simple elements for sound generation, the composition assumes a certain complexity through the interaction between elementary electronic sounds in a feedback network capable of processing them both synchronically and diachronically. Thus, sound elements no longer exist just as intrinsic and independent entities; they become instead strongly characterized by global processes that transform them as part of the network. More specifically, in the final result, timbre and temporal development often depend more on such processes than the actual nature of the sound sources. In the context of a personal artistic research, a processing system with said features—subsequently named SPAN—was already designed in 2011 for the previous *Axon* (Scordato 2011).

Additive synthesis sounds and drones are used as separate inputs for feeding SPAN, which is based on eight interconnected processing units that perform band-pass filtering (*fffb~*) and pitch shifting processes (*gizmo~*). Inputs are routed to the processing units according to eleven presets that also establish the following parameters: number of active band-pass filters (from 3 to 17) for each unit, number of transpositions (from 1 to 3), and latency factor [ms] in the transmission of audio signals in the network (passing through *delay~* objects). The particularity of the audio transmission within SPAN is that is position-dependent: delays are

calculated according to the spatial arrangement of the processing units, considering that each output is routed to a loudspeaker in an eight-channel system.

Finally, the gain of each unit is controlled in real-time by a performer who therefore alters the balance of the network. An animated score of *Constellations* was implemented in December 2015 on the occasion of Living Lab Music 6, a showcase of research results by the Sound and Music Processing Lab of the Conservatory of Padua, Italy (“Living Lab Music 6—SaMPL.”). In this new version, the position of the lanniX viewport changes according to commands (zoom, center, and rotate) stored in a *coll object* (“Max Basic Tutorial 18: Data Collections.”) and sent opportunely to lanniX via OSC.

3.2. Study for a network (2015)

Study for a network is an interactive audiovisual installation created in October and November 2015 for a collective exhibition at the Centre Cívic Convent de Sant Agustí, Barcelona. This work was accessible to the public on November 17–21, 2015 during Recorreguts Sonors, an annual event that brings together artists and creatives with different perspectives who deal with sound as expressive medium (“Recorreguts Sonors: Centre Civic Convent Sant Agusti.”). A work description has been provided as follows:

Study for a network projects the Barcelona metro network topology in the audiovisual space of the installation by means of voice and words.

Through hand gestures, users can move, zoom in and out on portions of the metro map displayed on a graphical interface, also controlling the virtual listening point: from a clear voice announcing the stops of a metro line, to a ubiquitous listening perspective (i.e. the unintelligible set of announcements of the entire network). (“Showcase | lanniX.”)

The lanniX score comprehends a metro map of Barcelona on which 11 curves and 148 triggers have been hand-drawn (cf. Appendix B). These objects represent metro lines and metro stops, respectively. Each curve is travelled by a looping cursor that moves on it at a predetermined speed. In their path, cursors come across triggers aimed at starting the playback of sound samples in Max/MSP.

Also, they report their current position for sound spatialization in an immersive projection system (e.g. quadrasonic or octophonic setup).

In Max, trigger messages—that convey `cursor_ID` and `trigger_ID` values received from *udpreceive object*—are routed to 11 polyphonic sub-patches (*poly~*) based on sample playback (“MSP Tutorial 13: Recording and Playback.”) with gain control (*gain~*). Thus, each sampler is set to play the pre-recorded vocal announcements related to a specific metro line. Cursor messages, instead, provide the position of sound sources—identified again by `cursor_ID`—to be spatialized with Ambisonics through the *ambipanning~ object* (Schacher 2010); general control syntax is:

```
xyz <source ID> <x> <y> <z> <source ID>
```

Hence, metro line announcements are spatialized as follows:

```
xyz <cursor_ID> <cursor_xPos> <cursor_yPos> 0. <cursor_ID>
```

As reported in the synopsis, user interaction consists in the exploration of the score through hand motion, with the consequence of changing the listening perspective according to the current position of the viewport.

Fig. 3

Excerpt from the interactive score of *Study for a network* (<https://youtu.be/ozw2VTatWDE>).

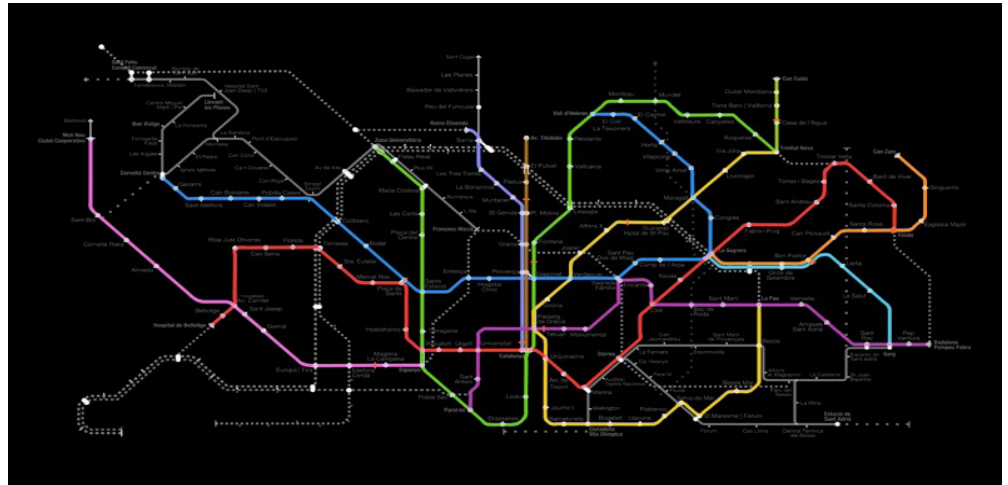
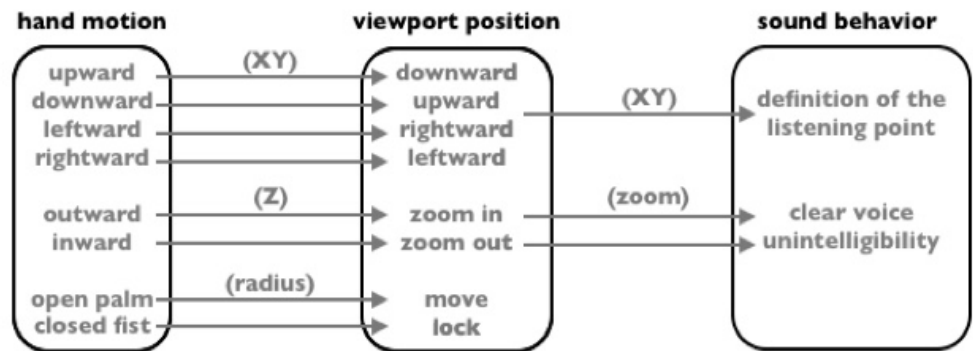


Fig. 4

Human-machine interaction in *Study for a network*.

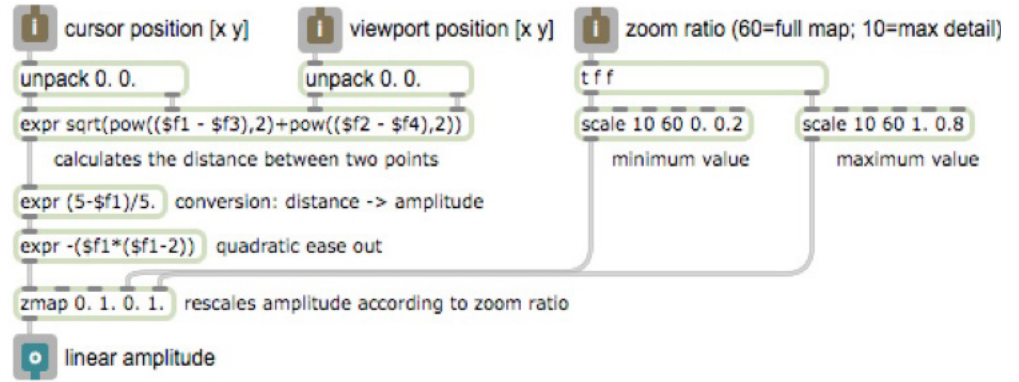


For the implementation of user interaction, a Leap Motion device has been positioned in front of the video projection area, as it offers a low-budget motion tracking sensor for mapping hand data, including palm position and size (van Canneyt 2013). Motion-related data are imported into Max/MSP through the *leapmotion* object (Françoise 2014):

- `palmposition <x> <y> <z>` → 3D position;
- `sphereradius <value>` → palm size.

When the palm size exceeds a threshold value, Leap Motion data are processed (*line* and *scale*) and forwarded to *lanniX*, to Ambisonics spatialization, and to sampling sub-patches. In *lanniX*, palm position values serve as arguments for center (XY) and zoom (Z) commands that control the viewport position (cf. Fig. 4). While in the spatialization system, X and Y values—rescaled according to a zoom ratio—are used to shift the position (`cursor_xPos` and `cursor_yPos`) of sound sources. Even the gain control of the samplers is motion-dependent: linear amplitude values are calculated by considering the distance between the moving cursor position and the viewport shift caused by the hand motion; additionally, the output is rescaled in order to foster an increased intelligibility of vocal announcements as the user focuses on details of the metro map. Max implementation is described by the following figure:

156 Fig. 5
Distance-amplitude
conversion.



3.3. Pulsion X (2016)

Pulsion X is a generative audiovisual work created in March-April 2016 with the intention to explore the graphic capabilities of lanniX through the definition and the manipulation of complex curves. It has been premiered on June 3, 2016 at the Cinéma de la Maison de la Culture de Bourges. A brief note has been released:

X is an abstract object: the unknown; the point as a generating element. X is also operation: the multiplication; the negation as a function of resistance. Through generative audiovisual processes, *Pulsion X* introduces the reticular form: nodes get connected to each other and the microform explodes. Globally, the chain of generated explosions describes an ever-changing texture. Nevertheless, nodes manifest themselves in an apparent atemporality, in an actual immobility.

Generative algorithms are implemented in Max/MSP programming environment, while the lanniX score only reacts by redrawing its objects according to commands received in real-time via OSC. In this case, sequencing functions in lanniX are completely unused, as the score only contains information about visualization (cf. Appendix C): six three-dimensional curves are defined as mathematical equations with three parameters (param1, param2, and param3); through the setEquationParam command, parameter values are changed, and curves are instantly redrawn.

In *Pulsion X*, real-time operation is based on three processing layers that are passed simultaneously to an output stage (cf. Fig. 7): stochastic audio reproduction, audio processing through SPAN system (cf. Chap. 3.1), and visual representation in lanniX.

As regards the audio reproduction, six short audio buffers (*buffer~*) are played continuously forwards and backwards at different speeds (*play~*) with muting functionality controlled by the average of the audio signal itself (*average~*): for each player, when the average amplitude value (*snapshot~*) falls below a lower threshold, there is a 1/3 chance to be muted; while, if the same value exceeds an upper threshold, the player has 1/7 chance to be unmuted. Audio output is on six independent channels.

157 **Fig. 6**
Excerpt from the reactive score of Pulsion X (<https://youtu.be/-OQxB15EtnA>).

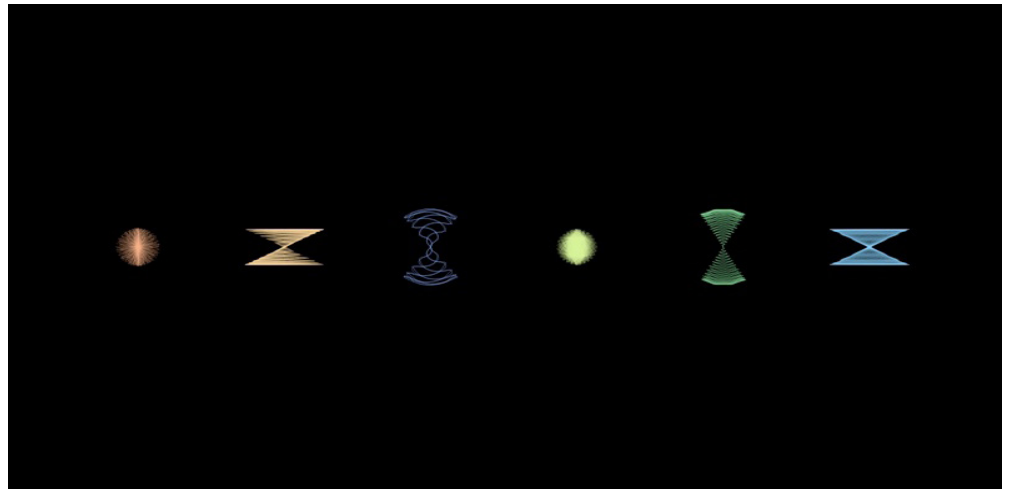
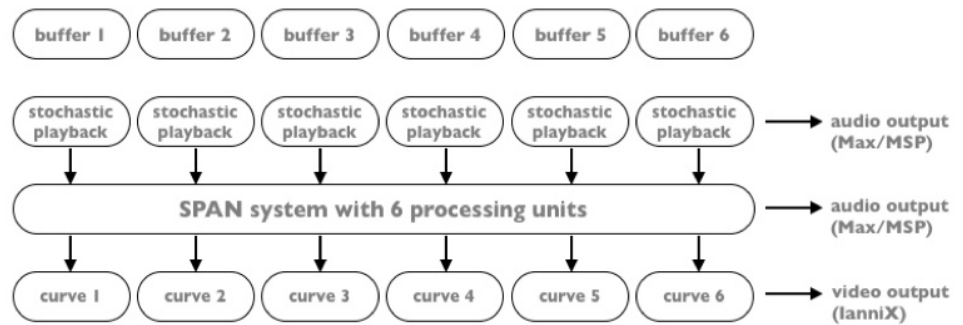


Fig. 7
Output stages in *Pulsion X*.



The second processing layer is constituted by SPAN system, already described in Chap. 3.1. In this specific case, the audio feedback network is made of six interconnected processing units used to transform the audio signals received from the stochastic players. System parameters change according to thirty global presets (*coll*) that are randomly recalled at the activation of the previously mentioned unmute function.

In the last layer, audio parameters—amplitude [dB] and fundamental frequency [Hz]—extracted with *fiddle~* objects (Puckette, Apel and Zicarelli 1998) from each SPAN output are used to control the visualization of the curves through the following commands:

```
setEquationParam <ID> param1 <rescaled amplitude [dB]>
setEquationParam <ID> param3 <rescaled frequency [Hz]>
setColorActiveHue <ID> <rescaled frequency [Hz]> 130 250 27
```

Ultimately, curve activation status and positioning commands (*setActive*, *setPos*, *zoom* and *rotate*) are stored into the global presets (*coll*) and transmitted to *lanniX* when any audio player is unmuted.

4. CONCLUSIONS

With its openness, flexibility, and a strong level of abstraction, *lanniX* may reserve interesting applications in the future. Even if its current interface contains symbolic limitations that does not make it as flexible as a graphic design application, in the context of experimental notation and sequencing programs, *lanniX* is capable to provide new ways of drawing sound events and performing graphic information (Georgaki 2006; Bourotte 2012; Gottfried 2015); also, the ability to import external graphics amplifies the representational possibilities of basic

objects. In general, the system is able to manage digital information bidirectionally, as it results from the score types described in Chap. 2. Therefore, the use of IanniX is not limited to digital art but is also applicable in design and mixed technology contexts.

Being the result of the work by a single author, the case studies described in Chap. 3 actually offer a limited perspective over practical IanniX applications, as they are strongly influenced by a personal artistic approach and creative tools chosen for contingent reasons (e.g. available equipment and context of presentation). Nevertheless, they represent singular examples of creative process sharing that might clarify certain aspects concerning the software usage.

REFERENCES

- Bourotte, Rodolphe.** 'The UPIC and Its Descendants: Drawing sound 2012.' *Proceedings of the International Symposium Xenakis*, Paris, France, May 23-25, 2012.
- Canneyt, Michaël.** 'Programming with the leap motion.' Accessed March 25, 2017. <http://www.freepascal.org/~michael/articles/leap1/leap1.pdf>
- Centre Cívic Convent de Sant Agustí.** 'Recorreguts Sonors: Centre Cívic Convent Sant Agustí.' Accessed March 25, 2017. <http://conventagusti.com/blog/category/recorreguts/>
- Charles, Jean-François.** 'A Tutorial on Spectral Sound Processing Using Max/MSP and Jitter.' *Computer Music Journal* 32-3 (2008): 87-102.
- Coduys, Thierry, and Guillaume Jacquemin.** 'Partitions retroactives avec IanniX.' *Actes des Journées d'Informatique Musicale*, Bourges, France, May, 21-23, 2014.
- Cycling '74.** 'Max Basic Tutorial 18: Data Collections.' Accessed March 25, 2017. <https://docs.cycling74.com/max5/tutorials/max-tut/basicchapter18.html>
 — 'MSP Tutorial 7: Additive Synthesis.' Accessed March 25, 2017. <https://docs.cycling74.com/max5/tutorials/msp-tut/mspchapter07.html>
 — 'MSP Tutorial 13: Recording and Playback.' Accessed March 25, 2017. <https://docs.cycling74.com/max5/tutorials/msp-tut/mspchapter13.html>
 — 'MSP Tutorial 21: Using the ~poly Object.' Accessed March 25, 2017. <https://docs.cycling74.com/max5/tutorials/msp-tut/mspchapter21.html>
- Françoise, Jules.** 'Leap Motion skeletal tracking in Max.' Accessed March 25, 2017. <https://julesfrancoise.com/leapmotion/>
- Georgaki, Anastasia.** 'The grain of Xenakis' Technological thought in the Computer Music Research of our days.' *Proceedings of the 1st International Symposium Iannis Xenakis*, Athens, Greece, May 18-20, 2005.
- Gottfried, Rama.** 'SVG to OSC Transcoding: Towards a platform for notational praxis and electronic performance.' *Proceedings of the 1st International Conference on Technologies for Music Notation and Representation*, Paris, France, May 28-30, 2015.
- IanniX Association.** 'Showcase | IanniX.' Accessed March 25, 2017. <https://www.iannix.org/en/projects>
 — 'What is IanniX? | IanniX.' Accessed March 25, 2017. <https://www.iannix.org/en/whatisiannix/>
- Puckette, Miller, Theodore Apel and David Zicarelli.** 'Real-time audio analysis tools for Pd and MSP.' *Proceedings of the 24th International Computer Music Conference*, Ann Arbor, USA, October 1-6, 1998.
- Schacher, Jan C.** 'Seven years of Ambisonics Tools for MaxMSP—A brief report.' *Proceedings of the 2nd International Symposium on Ambisonics and Spherical Acoustics*, Paris, France, May 6-7, 2010.
- Scordato, Julian.** 'Axon'. Udine, Italy: *Taukay Edizioni Musicali*, 2011.
 — 'IanniX software documentation' Accessed April 15, 2017. <https://www.iannix.org/download/documentation.pdf>
- Sound and Music Processing Lab.** 'Living Lab Music 6—SaMPL.' Accessed March 25, 2017. http://www.sampl-lab.org/en/index.php/Living_Lab_Music_6
- Wright, Matthew, and Andrew Freed.** 'Open SoundControl. A New Protocol for Communicating with Sound Synthesizers.' *Proceedings of the 23rd International Computer Music Conference*, Thessaloniki, Greece, September 25-30, 1997.

```
/*
    Constellations, for graphical sequencer and electronics, 2014
    Copyright (C) 2014-2016 - Julian Scordato

    Generalization of the IanniX score.
*/

//syntax for two curves
run("add curve <ID>");
run("setpos current <x> <y> 0");
run("setpointat current 0 <x> <y> 0"); //point 1
run("setpointat current 1 <x> <y> 0"); //point 2
run("setsize current 3"); //curve thickness
run("setactive current 0");
run("setcolorinactive current 255 255 255 102");
run("setcolormultiply current 255 255 255 102");

//syntax for two cursors
run("add cursor <ID>");
run("setcurve current lastCurve");
run("setsize current 3");
run("setwidth current 3.3");
run("setdepth current 0.1");
run("setpattern current 0 0 1 -1 0"); //single roundtrip
run("setspeed current auto 210"); //total duration is 420 s
run("setmessage current 20, osc://ip_out:port_out/cursor cursor_id cursor_value_y");
//last variable controls pitch
run("setboundssourcemode current 1");
run("setcoloractive current 255 0 0 110");
run("setcolormultiply current 255 0 0 110");

//syntax for 356 triggers
run("add trigger <ID>");
run("setpos current <x> <y> 0");
run("setsize current <value>"); //for visualization purposes
run("settriggeroff current <duration [s]>");
run("setgroup current <linear amplitude value>");
run("setlabel current <r/c/t>"); //sound timbre
run("setmessage current 1, osc://ip_out:port_out/trigger trigger_duration trigger_
value_y trigger_group_id trigger_label <0/1>"); //last value controls sustain
run("setcoloractive current 255 255 255 <transparency>");
```

```

/*
    Study for a network, interactive audiovisual installation, 2015
    Copyright (C) 2015-2016 - Julian Scordato

    Generalization of the IanniX score.
*/

//load metro map as background image
run("registertexture background -4 2.01881 4 -2.01881 <filename>");

//syntax for 11 hand-drawn curves made through IanniX GUI
//curves retrace the metro lines such as in the background image
run("add curve <ID>");
run("setpos current <x> <y> 0");
var <name> = [ <points> ];
for(var i = 0 ; i < <name>.length ; i++)
    run("setpointat current " + i + " " + <name>[i].x + " " + <name>[i].y + " " +
<name>[i].z + " " + <name>[i].c1x + " " + <name>[i].c1y + " " + <name>[i].c1z + " " +
<name>[i].c2x + " " + <name>[i].c2y + " " + <name>[i].c2z); //set points of a smooth
curve
run("setsize current 7"); //curve thickness
run("setcolor current <r> <g> <b> 0"); //color such as in the image

//syntax for 11 cursors
run("add cursor <ID>");
run("setcurve current lastCurve");
run("setsize current 3");
run("setwidth current 0.05");
run("setdepth current 0.1");
run("setpattern current 0 0 1 -1"); //loop on support curve
run("setspeed current 0.07");
run("setmessage current 20, osc://ip_out:port_out/cursor cursor_id cursor_xPos
cursor_yPos"); //values control sound spatialization and amplitude

//syntax for 148 triggers
run("add trigger <ID>");
run("setpos current <x> <y> 0"); //position of the metro stops
run("setsize current 0.1");
run("settriggeroff current 1"); //for visualization purposes
run("setmessage current 1, osc://ip_out:port_out/trigger cursor_id trigger_id"); //
values control sound sampling
run("setcolor current _curve_active");

```

APPENDIX C: PULSION X – SCRIPT

```

/*
  Pulsion X, generative audiovisual art, 2016
  Copyright (C) 2016 - Julian Scordato

  Full IanniX score.
*/

//6 three-dimensional curves defined as parametric equations
run("add curve 1");
run("setequation current polar (400*param1+param2)*sin(7000*para-
m3*t*2*PI),(1000*param1+70*param2)*(70*param3*t*2*PI),(300*param1+param2)*(100*para-
m3*t*2*PI)");

run("add curve 2");
run("setequation current polar 0.5*(param1+5*param2)*sin(50*para-
m3*t*2*PI),(700*param1+param2)*atan(7000*param3*t*2*PI),
sqrt(300*param1+param2)*tan(4000*param3*t*2*PI)");

run("add curve 3");
run("setequation current polar 0.5*(param1+5*param2)*sin(50*para-
m3*t*2*PI),(700*param1+param2)*sin(7000*param3*t*2*PI),(300*param1+param2)*-
cos(4000*param3*t*2*PI)");

run("add curve 4");
run("setequation current polar (300*param1+param2)*sin(100*para-
m3*t*2*PI),(1000*param1+70*param2)*(70*param3*t*2*PI),(400*param1+param2)*(7000*par-
am3*t*2*PI)");

run("add curve 5");
run("setequation current polar 0.5*(param1+5*param2)*sin(50*para-
m3*t*2*PI),(300*param1+param2)*cos(4000*param3*t*2*PI),(700*param1+para-
m2)*sin(7000*param3*t*2*PI)");

run("add curve 6");
run("setequation current polar 0.5*(param1+5*param2)*sin(50*para-
m3*t*2*PI),(700*param1+param2)*atan(7000*param3*t*2*PI),(300*param1+para-
m2)*(4000*param3*t*2*PI)");

run("setpos all 0 0 0");
run("setsize all 1"); //curve thickness
run("setequationparam all param1 0.001"); //variable parameter
run("setequationparam all param2 0.483"); //fixed parameter
run("setequationparam all param3 0"); //variable parameter
run("setequationnbpoints all 2279");
run("setcoloractivehue all 64 0 255 0"); //invisible at start
run("setcolorinactivehue all 255 255 255 0"); //invisible if inactive

```