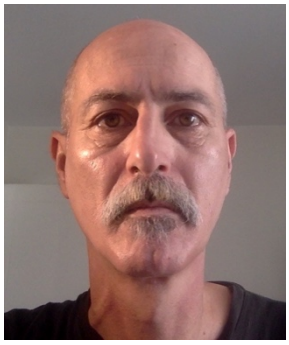
Riding the audio wave

A7, a custom-generative software for visual performance

Guillaume Rochais, PhD candidate and artist-programmer.
ACCRA, Université de Strasbourg, France
e-mail: guillaume.rochais@gmail.com



1 Introduction

1.1 Presentation

This paper presents the philosophy and main features of A7, a custom software for real-time visual accompaniment of musical performances.

Made with TouchDesigner, this work includes three distinct parts: the program itself, the generative and audio-reactive modules that feed it, and the graphical user interface that allows interaction with both the program and the visuals.This interface is similar to a musical instrument, and the performance to an interpretation placed under the sign of improvisation.

The three parts of this work are closely linked, each one coming, over the course of iterations, to question the others. A7 is therefore a work in progress, constantly rethought, refined and enriched.

1.2 VJing, live visuals & A/V performances

The practice we are developing here is close to what is commonly referred to as VJing, a term born from an analogy with DJing in the musical world, and which seems to have originated at the end of the 1970s in the PepperMint Lounge [1], a New York nightclub, before being popularized by MTV in the 1980s [2].

The Video Jockey would therefore be the visual equivalent of the Disc Jockey and his role would be to select, manipulate and mix live video sources of which he is not necessarily the author, most often with existing tools.

The VJ is thus above all associated with the world of clubbing, his performance being regarded as mainly decorative,like a simple update of the light shows and disco balls of the 1970s.

Like many performers today, we do not find ourselves in this definition. Indeed, our approach is based, on the one hand, on the exclusive use of visual elements programmed by us and generated in real time, and, on the other hand, on a home-made software, as much for the

audiovisual processing as for the graphic interface.

Beyond these technical aspects, our approach is also distinguished by the context in which it is deployed. We don't work in nightclubs with DJs, but with electronic musicians playing live their own compositions in a concert configuration, and just like them, we claim the singular nature of our creations and their full belonging to the artistic field.

To define our practice, we borrow the terminology proposed by Steve Gibson and al [3]. When the visual performance is not, or very little, prepared with the musician, we will use the term *Live visuals*. When the work presented is the result of a real collaboration with the musician, we will rather speak of *A/V performance*.

1.3 Jonathan Noé aka Coloscope

This presentation of A7 aims to inform the demonstration that will be made during this conference. This performance will include a concrete implementation of the tool: the creation of live visuals for the Strasbourg-based electronic musician Jonathan Noé. Under the name *Coloscope*, he proposes an electro-punk

> "torn between the first transgressions of krautrock and the synth lines of John Carpenter's films […] an entirely live and largely improvised electro set that reconnects with the DIY, precarious and minimalist approach of punk and underground medicine." [4]

Jonathan Noé built his instrument entirely with Pure Data, both for sound synthesis and sequencing. The program is mainly based on FM synthesis and excludes the use of samples. During the live performance, it is executed on a Raspberry Pi model B nano computer equipped with a sound and MIDI card Pisound, a strong technical constraint, chosen and claimed, which imposes sobriety in the use of resources and denotes an approach anchored in FLOSS and DIY.

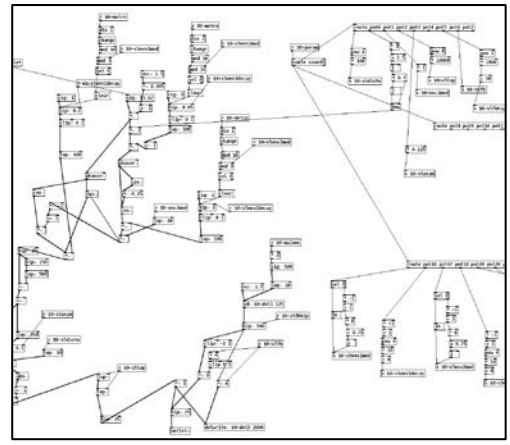


Fig.1 Screenshot of a small part of Jonathan Noé's Pure Data network.

In concert, the Raspberry works in an autonomous way, without a screen, nor mouse or keyboard. It is only driven by two MIDI controllers: a BCF 2000 control surface featuring faders, knobs and rotary encoders and a Nektar Impact LX25 two-octave keyboard equipped with knobs and rotary encoders.

Fig.2 Jonathan Noé playing while we are handling the visuals. ChaosLab 2, October 2022, Strasbourg.

1.4 TouchDesigner

A7 is designed with TouchDesigner [5], a visual programming language developed by the Canadian company Derivative. It allows the production of a wide variety of projects, from application prototyping to interactive or immersive installations, including audiovisual performance, virtual reality, mapping or concert light shows.

Similar to software such as Pure Data, vvvv or Max/MSP, TouchDesigner has the particular advantage of using Python as a scripting language, the software being delivered with a custom Python build.

Although the software is proprietary, TouchDesigner is offered in a free version for non-commercial use, with some limitations in functionality that nevertheless leave ample possibilities for creation.

2 Path, intentions and philosophy

The work undertaken with A7 follows on from various experiments carried out since 2017 in the field of Live visuals, with from the outset two parties that are still relevant today: the design and fabrication of a custom software, and the exclusive use of visuals generated in real time. The idea was to design a program, accessible through a user interface, allowing the manipulation of pre-written generative and audio-reactive visuals in real time.

Several versions followed one another. The previous one, which I used and refined for three years, was based on the use of a McMillen QuNeo MIDI controller which was not without programming and ergonomic issues.

It was this point that triggered the desire to completely redesign the program by replacing the MIDI controller with a touch screen, at the moment a Samsung Galaxy Tab A7 Lite tablet.

While the previous user interface was largely conditioned by the controls available on the QuNeo, the GUI accessible on the tablet is completely custom. It also removes a whole software layer: on-screen transfer of Quneo controls, MIDI feedback to the controller, LED lighting.

This choice implies radical changes, a simple update of the previous project was not possible. A7 was therefore created from scratch. However, thanks to the modular architecture at the heart of TouchDesigner, some elements, in particular the generative visuals, could be reused with some adaptations.

A major evolution of A7 is the use of two instances of TouchDesigner running on the same machine: one running the GUI, the other the main program, both communicating through a message system specific to TouchDesigner via the TCP/IP protocol.

The performance gain is significant, as two instances make a better use of the CPU cores without the complexity that a direct thread management would impose. In addition, the GUI is only displayed at 30 frames per second, which saves resources to display visuals at 60 frames per second.

The practice of Live visuals requires a lot of work beforehand, which is a combination of generative art, pure programming and UX design. However, if there are three distinct parts to this work, both in terms of the skills they require and the issues they raise, and if they take place at different times, these three parts

are closely linked. Over the course of the iterations, each one modifies the others, revealing possibilities, underlining gaps or problems. This work is therefore in constant evolution, constantly rethought, refined and enriched.

3 Architecture

3.1 Program

The program has a modular design and is inspired by video and audio mixers. It allows two generative modules to run and to be manipulated simultaneously.

Whether they generate texture or geometry, these modules deliver a rasterized image at the output, which first passes through a post-processing module, then a compositing module.

The resulting video signal finally passes through a finalization module to adapt the projected image to the projection conditions.

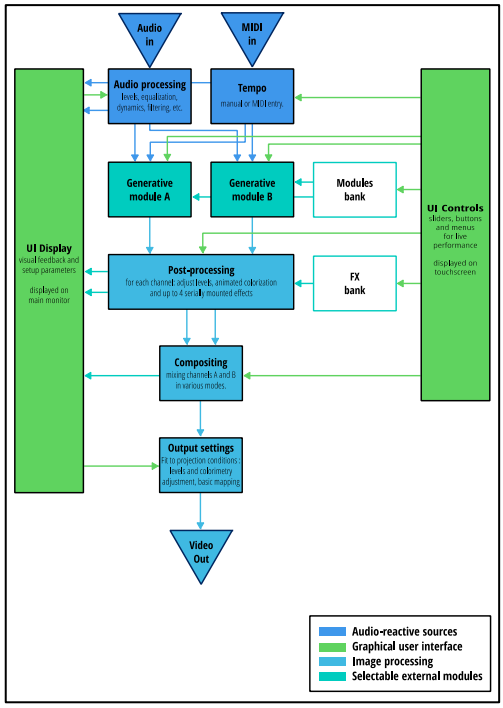


Fig.3 A7 Flow chart

While this design is reminiscent of the A/B model used by DJs, which is widely used in VJing, our practice is not very oriented towards mixing or switching sources. These two channels are generally used in the Preview/Program logic of a video mixer: while a visual runs in a certain configuration on one channel, the other allows to prepare the next evolution.

A recently added module offers a texture bank to feed generative modules that need it. It allows in particular to inject the image produced by a generative module in the one loaded in the other channel. For the moment, this remains rather limited, and needs to be refined and enriched.

The program also integrates a sound processing module that delivers different signals—reference audio, bass, midrange and treble—reprocessed by resampling and filtering to be used as visual parameter controls, as well as the corresponding RMS powers.

In addition, there is a tempo module that allows you to beat the bar manually or to obtain this information via MIDI messages. It produces from the tempo different types of periodic signals enabling to introduce a rhythmic dimension in the visual variations.

The possibility of receiving information from the musician via MIDI is a new feature of A7 compared to its predecessor. For the moment, it is limited to this tempo information, but the long-term collaboration with Jonathan Noé allows us to envisage a more refined work on the relationship between sound and image, an evolution from Live visuals to A/V performance.

The program also includes automation and randomization system, which allows for a partial or total autopilot operation. The development of richer and more complex automation is the main axis of development envisaged in the short term.

3.2 Interface

The design of the interface implies thinking about a use, a "way of playing". Its goal is to offer a maximum of freedom during the performance while preserving the handling. It is therefore necessary to identify the interesting parameters to be manipulated, but also to imagine the ergonomics that is the most appropriate to the performative practice. It is then easy to understand that interface and performance influence each other, the former conditioning the possibilities of the latter, and the latter revealing the lacks or superfluous functions of the former.

Concretely, the interaction with the program is done through two devices. The Display module is displayed on the computer screen and provides visual feedback and access to basic settings. The Controls module is displayed on the touch screen and presents all the useful controls during the performance: selection of visuals and effects, manipulation of generative modules, compositing…

3.2.1 Display

The Display module offers four video monitors: the final image, the two channels, with the possibility of highlighting the alpha layer and viewing the image before or after post-processing, and a preview of the selected compositing mode.

There are also two groups of three tabs. In the first group, the Audio section allows you to view and adjust the sound level of the input, bass, midrange and treble, to adapt the smoothing of the signals, and to select the audio input and output.

The Tempo section allows you to switch between tempo entered manually or obtained by MIDI message. It also provides visual feedback of the rhythm in bars and beats, and of the multiples of the beat used by the generative modules.

Finally, several useful controls during startup and initialization are grouped in the Utils section. A lock allows to block the functionalities in order to avoid any manipulation error during the performance.

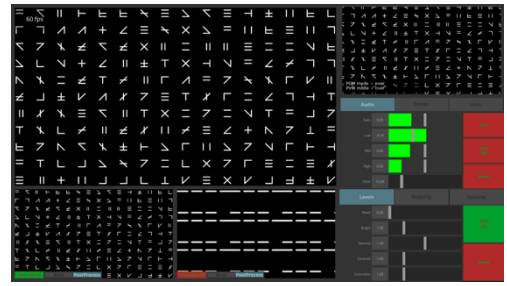


Fig. 4 Screenshot of Display UI

In the second group, the Levels tab allows you to adjust the levels of the image transmitted to the projector, and the Mapping tab to adapt the geometry. These controls are set according to the projection conditions before the performance, and do not affect the feedback displayed on the computer screen.

Added to this is a panel for previewing and selecting the textures provided as input to each generative module.

3.2.2 Controls

The Controls module is composed of three sections: on the left and right, the controls of each channel, in the center

the shared controls. In each section, the most used live features are stacked from bottom to top to optimize their accessibility and r educe the risk of unintentional manipulation.

3.2.2.1 Channels A and B

For each channel, there is a menu at the top to load the generative modules from the library. They are initialized during this loading, but incidents may occur depending on the use of resources that is then made. An error notification system will indicate this by superimposing it on the video feedback of the channel concerned. The Reinit button on t he left of the menu will then allow you to correct the problem.

Fig. 5 Screenshot of the A and B channels section of the Controls UI

At the bottom of the section, eight faders and eight buttons can be used to act directly on t he generative modules. The parameters controlled are specific to each module, the labels of the controls being updated when they are loaded. The buttons can have three different behaviors: toggle (alternately on and off), momentary (on when pressed, off otherwise) or radio (buttons in a group of which only one can be on). These behaviors are specific to each generative module and are updated during loading.

The section above allows you to navigate between the twenty presets saved for each generative module. Presets include, of course, the sixteen parameters that can be manipulated by the sliders and buttons, but also other parameters that are specific to each generative module and that are not accessible in a performance situation. The Save button is used to save the current configuration. To avoid any manipulation error during the performance, it requires a long press.

Then come the last controls acting directly on t he generative modules: four buttons that permit to multiply or divide the frequency of the periodic signals coming from the tempo that the modules use. In order to maintain synchronism, the multiplication coefficient is only applied at the beginning of the bar. It appears grayed out as long as it is not active.

The last section includes four tabs to access the post-processing and automation controls. The first offers four controls analogous to the Aux Sends of an audio mixer. They control whether and how much each effect is applied to the image.

The second tab can be used to adjust the image levels by setting four parameters: black level, brightness, gamma and contrast. The third tab is a dynamic colorization tool. It allows you to adjust the mix of the effect and the source image, the base color of the colorization, the range of the color variation around this base and the speed of this variation. It also has a button to convert the image to black and white.

The Autopilot tab currently offers few features. The Filter slider controls the interpolation between the previous and next value of a pa rameter when it is changed manually or by loading a new preset. The Filter value indicates the duration of this interpolation in seconds. The P Dur slider sets the display time of a preset when Auto Pilot or Auto PST modes are active.

The Auto PST button automatically and randomly changes the preset. The Auto-Pilot button does the same thing, but also changes the Filter and P Dur parameters on each pull. The Random PST button simply draws a preset at random. Finally, the Random Pars button generates a random configuration of all the custom parameters of a gener ative module. It is only used in the search phase, and is disabled in performance situations as its results are so unpredictable.

3.2.2.2 Shared controls

The Shared section, in the center of the Controls, is composed of three blocks of commands: at the top, the effect management, below the manual tempo adjustment and below the compositing functions.

The effects section has four tabs corresponding to the four effects that can be used simultaneously. A drop-down menu allows you to load the desired effect from the dedicated library. Two groups of eight radio buttons are used to select, for each channel, one of the eight available presets.

In the Tempo section, the Tap button is used to beat the bar manually, and t he Reset button to synchronize the periodic signals with the beginning of the musical bar. Four buttons are then used to adjust the resulting tempo in steps of +/-1 or +/- 0.1 bpm.

Fig. 5 Screenshot of the Shared section of the Controls UI

In the Compositing section, the horizontal slider controls the crossfading from one channel to another. On either side, a button allows instant switching. The two horizontal sliders are used to control the replacement of the alpha layer by black for each channel, which can have a significant impact on t he result of compositing operations.

Above the horizontal slider, four buttons are used to select the compositing mode for both channels. Two buttons allow you to navigate through the forty-five available modes, and a third reverses the order of compositing operations for modes where this order is important. The resulting setting is previewed in the Display window. The last button validate the selection.

A final series of controls is used to activate or deactivate compositing, and to adjust the level of mix between the source image and t he composite. Added to this line is a F ade to black button whose duration is adjustable. It is normally only used at the beginning and end of the performance.

3.3 Generative Modules

The generative visuals are at the heart of the program, they are the artistic material from which the improvisation will be built. They are conceived upstream in a generative approach—putting parameters in relation, using chance—, and integrate an audio-reactive dimension.

The characteristics of the music—frequencies, levels, tempo—influence the visuals produced to propose a synesthetic experience. The objective is to continuously compose new ones in order to enrich the palette available during the performance.

The creation of the visuals often takes place initially outside the main program. From a c oncept, a t echnique or an i dea of rendering, a pr ocess of research and experimentation begins, leaving a large place to serendipity. This work continues until it begins to settle, to find a satisfactory outcome.

In order to be inserted into the main program, and to be manipulated in real time, these generative modules must then be normalized. Each one has audio, RMS and tempo signals as input, as well as a texture, and provides a video image as output. They all have their own parameters, sixteen of which can be modified by the interface: up t o eight controllable by sliders, and eight others by buttons. When the module is loaded, the labels of these controls are updated with the parameters they modify.

The possible combinations of these controls lead to a w ide variety of renderings, and it can sometimes be difficult to retrieve a par ticular configuration.

This is why I added a system that allows to save and load twenty presets for each of the generative modules, as many milestones set along the exploration of their potentialities, whether it is done during the performance or before.

3.4 Effects

The post-processing effects are built on the same logic as the generative modules: they have their own parameters, their control by the interface is standardized, and they can be loaded on the fly from a dedicated library. Some are extremely simple, such as the feedback or edge detection effect, others are more complex, using the input image as material applied to 3D scenes, or including an audio-reactive dimension.

Each one has eight presets for ease of use. A possible evolution would be to allow a finer parameterization of the effects during the performance, without bringing too much complexity that could harm the ergonomics and the expression of spontaneity.

4 Concepts and theoretical approach

4.1 Generative approach

The generative approach is at the heart of our work. It translates into the massive use of systems aiming to introduce autonomy at all stages leading to the final visual, to set up a complex system of interactions and interferences constantly updating the visual configuration.

These can be simple random draws. They are then generated by the Random module of Python, paying particular attention to the management of the seeds so that no pa rameter uses the same one.

We also use noise generating algorithms (Sparse, Brown, Perlin, Simplex…) or sets of logical-mathematical operations trying to simulate complex systems, i.e. perfectly deterministic systems, but whose sensitivity to initial conditions makes their evolution unpredictable.

Or even systems of interactions between periodic signals which, although perfectly predictable from a theoretical point of view, generate extremely varied and often unexpected configurations as soon as they are sufficiently complex and t he frequencies used are close.

If audio signals are most often used to establish identifiable interactions between music and visuals, they can also be used as a b asis for generative systems as soon as we remove, by various processing operations, the characteristics recognizable by the ear.

This is the case, for example, for the charatcter strings used for the visuals of Jonathan Noé's concert. The audio signal is resampled to the desired number of characters. The last three digits of the numerical value of each sample are transposed into the choosen interval — from [0-1] to [0-254] — each sample thus generating one of the 255 possible signs included in a font we made before by generating all the combinations of eight fundamental lines.

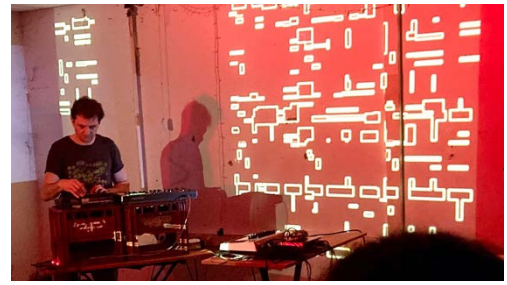


Fig. 6-9 Four views of the A7 in action with Jonathan Noé.

4.2 Control and Manipulation

Pure randomness has only a very limited visual interest, producing images that are always different, but nevertheless largely similar and only exceptionally offering a singular character. It is therefore necessary to tame it, to dose it, to constrain it in an ordered system. For it is indeed the articulation of chance and rule, the implementation of a "programmed chance" that is at the heart of the generative approach.

Although the interface offers multiple possibilities for manipulating the displayed configuration, the vast majority of the parameters determining the visual, either static or dynamic, are fixed in advance. The choice of those that can be modified in real time is crucial, as are the limits within which they can vary. It is on these controls that the performance will be built, in particular those of the generative modules whose progressive deployment in accordance with the music is the foundation of our approach.

4.3 Composition, Improvisation, Comprovisation

The practices of VJing, Live visuals and A/V performance are all performative. The distinctions that we have previously introduced are based in particular on the degree of preparation of these performances and of collaboration with the musician, i.e. the balance that they establish between composition and improvisation.

These two terms are generally perceived as opposites. To improvise is to seek out or provoke the unexpected (improvisus), whereas to compose (compono) is to put together elements, to build a whole. This opposition refers to an outdated conception of the work of art considered as a finished object, under the total control of its creator, a conception that composers such as John Cage have shattered since the 1950s [6].

The two notions now seem more complementary than opposed. How can one improvise without first having a musical experience made of long hours of practice or theoretical learning? How can one compose without being able to invent new motifs in the moment? Improvisation is built on experience, composition organizes the moment's emergences. Both coexist in musical performance, which leads Sandeep Bhagwati to introduce the notion of "comprovisation":

"The terms "improvisation" and "composition" can be useful as mental constructs but they can confuse and cloud the realities of music making mainly because they suggest that they are somehow dualistic, even antagonistic entities-where in reality they constitute points along a continuum. Using the term "comprovisation" can make us aware of

the contingent nature of this continuum: the fact that the term is so blatantly mongrel immediately leads to the question: how much and w hat is composed and how much and what is improvised in a given performance? And this uncertainty may prompt us to listen and look more closely at the individual moment of performance." [7]

These observations apply perfectly to the various forms of visual performance we have discussed, and are entirely consistent with the directions we have chosen and the way we look at our practice.

5. Conclusion

Like musical improvisation, visual improvisation requires regular practice in order to develop sufficient familiarity with the program and the interface to master its productions, to anticipate the result of the manipulations carried out in order to organize the emergence of the unexpected without letting oneself be surprised. It is a question of acquiring a form of virtuosity that allows one to unfold the thread of a v isual narrative inspired by music, in the perspective described by Pierre-Paul Lacas, for whom:

> "in improvisation, composition and execution coincide: the fingers execute what the mind seems to decipher on a secret score that would be printed at the very moment the instrument expresses it".[6]

While the preliminary stages allow for the reworking of every detail to get as close as possible to the desired result, while the unfortunate error is easily made up for, the performance implies risk-taking.

Concentration and attentive listening are essential to let oneself be carried by the music while making the visual production evolve, and c onstantly correct, by small touches of control, an un stable equilibrium in order to ride the sound wave.

Notes and references

[1] SHAUGHNESSY, Adrian, 2006. Last night a VJ zapped my retinas. In : *D-FUSE, VJ: Audio-Visual Art and VJ Culture*. Laurence King Publishing.

[2] CREVITS, Bram, 2006. The roots of VJing. In : D-FUSE, *VJ: Audio-Visual Art and VJ Culture*. Laurence King Publishing.

[3] GIBSON, Steve, ARISONA, Stefan, LEISHMAN, Donna et TANAKA, Atau (ed.), 2022. *Live Visuals: History, Theory, Practice*. London : Routledge.

[4] description by the artist himself

[5] https://derivative.ca/

[6] BOESWILLWALD, André-Pierre, FÉRON, Alain et LACAS, Pierre-Paul, [sans date]. *Improvisation musicale*. Encyclopædia Universalis.

7] BHAGWATI, Sandeep, 2013. Comprovisation – Concepts and Techniques. In : FRISK, Henrik et ÖSTERSJÖ, Stefan (ed.), *(Re)thinking improvisation: artistic explorations and conceptual writing*. Malmö : Malmö Academy of Music, Lund University. pp. 99-104.