DIGITAL ACCESS TO SCHOLARSHIP AT HARVARD
DASH.HARVARD.EDU

HARVARD LIBRARY

Placeholders for [failure?] — A music creation and technology research

Citation

Vincenot, Julien. 2024. Placeholders for [failure?] — A music creation and technology research. Doctoral dissertation, Harvard University Graduate School of Arts and Sciences.

Link

https://nrs.harvard.edu/URN-3:HUL.INSTREPOS:37379863

Terms of use

This article was downloaded from Harvard University's DASH repository, and is made available under the terms and conditions applicable to Other Posted Material (LAA), as set forth at
https://harvardwiki.atlassian.net/wiki/external/NGY5NDE4ZjgzNTc5NDQzMGIzZWZhMGFlOWI2M2EwYTg

Accessibility

https://accessibility.huit.harvard.edu/digital-accessibility-policy

Share Your Story

The Harvard community has made this article openly available.
Please share how this access benefits you. Submit a story

HARVARD
Kenneth C. Griffin

GRADUATE SCHOOL
OF ARTS AND SCIENCES

DISSERTATION ACCEPTANCE CERTIFICATE

The undersigned, appointed by the

Department of Music

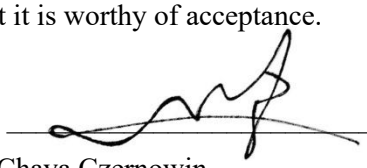
have examined a dissertation entitled

Placeholders for [failure?] — A music creation and technology research

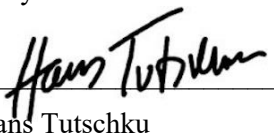
presented by

Julien Vincenot

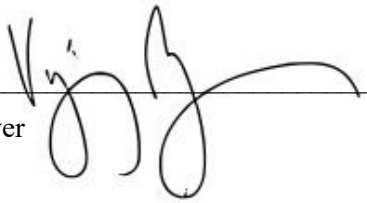
candidate for the degree of Doctor of Philosophy and hereby certify that it is worthy of acceptance.

Signature 

Professor Chaya Czernowin

Signature 

Professor Hans Tutschku

Signature 

Professor Vijay Iyer

Date: September 2, 2024

Placeholders for [failure?]

A music creation and technology research

A dissertation presented by

Julien Vincenot

to

The Department of Music

in partial fulfillment of the requirements for the degree of

Doctor of Philosophy

in the subject of

Music

Harvard University

Cambridge, Massachusetts

September 2024

© 2024 Julien Vincenot

All rights reserved.

Abstract

Placeholders for [failure?]

A music creation and technology research

This dissertation describes how I developed my practice as a composer between 2017 and 2024. I will first discuss my background and value system, then elaborate on several ideas and techniques for composition, interleaved with more specific examples of works, finished and unfinished. In the last years, my practice took different shapes and turns between mixed music composition (i.e. instrumental music with live computer), acousmatic music (or fixed media), improvised performances, audiovisual generative or interactive installations, and hybrid projects with generative scores. Those different categories of work, in which I aspire to keep expressing myself, reflect my ever-changing interests, and I hope they make obvious the necessity to keep extending the scope of what we call « composition ».

My research process has been strongly influenced by the development of dedicated software for computer-aided composition (CAC), in particular the *MOZ'Lib* package (a satellite of the *bach* ecosystem for MaxMSP) which I actively maintain since 2015. I will discuss it partially in the middle of this text, trying to address its most salient features and hoping to trigger the curiosity of fellow composers, musicians and musicologists to explore it by themselves.

Finally, a modest collection of scores will end this portfolio of artistic and research projects. During the time of this research, my interest has switched from "ordinary scores" to fully software-based generative scores, and it would hardly make sense to reduce my latest projects to a series of pages. Some of these projects were initially intended as live mixed music, before solidifying into fixed media. When available, I will provide images of excerpts or screen recording videos for reference. I will provide a recording (live, studio or a simulation) of some of the projects discussed, when available, as well as a couple of pieces of mixed music predating my time at Harvard, to give more context on my personal trajectory in the last ten years.

Table of Contents

Title Page i

Copyright ii

Abstract iii

Table of Contents iv

Acknowledgements v

Foreword 2

Part I **3**

Memory, writing & code 3

Hiding my intentions : conceptual vs. technical framework 6

(Mixed) Music and Magic 9

CAC and augmented imagination 13

Mixed music production : Cinema vs. Theater? 25

Interlude I — A small catalog of (loosely related) techniques 29

Placeholders and sprites 41

Interlude II — MOZ'Lib 49

Part II **61**

Graph theory or catastrophe theory? 61

Silent cacophony and a few broken mechanics 75

Keeping the score in a liquid state 87

After CAC? 95

Portfolio **101**

Acknowledgements

Before getting into the body of my work, I cannot avoid to mention the eruption of Covid-19 pandemic in my third year on Harvard campus. At this time the most substantial part of my research was about to take off, and as a result it made my creative life rather difficult and painful. From this point onwards social anxiety, which felt already at home, kept growing. The work never really stopped, but my music practically ceased to exist publicly, and I took very few projects to completion. The very last years have been a struggle, catching up with missed opportunities, rebuilding the tools that I lost, and reimagining my whole practice. Most of all, I wanted to restore my drive to compose again, and today I feel I have partially succeeded.

For all these reasons I want to extend my warmest thanks to both my advisors Hans Tutschku and Chaya Czernowin, for their enduring patience and support for my work, despite the challenges of distance and long periods of silence. I also want to express my gratitude to Vijay Iyer, who accepted to serve as a reader on my dissertation committee. I really appreciate his interest and decisive suggestions when my projects felt the most stalled.

A special "thank you" goes to Claire Chase, whose inspiring workshop for flutists/composers in 2018 sparked my exploration into generative scores. I also want to thank Ray Evanoff and Evan Johnson for the two exceptional analysis classes they gave to our small group of composers. Those were the highlights of my time at Harvard, and I will remember them dearly.

For the chance I had teaching at their side, I want to thank Katherine Pukinskis, Joseph Jakubowski and of course Hans Tutschku. In spite of rather perilous times and significant technical constraints — most classes were taught online at the height of the pandemic. If there is one thing the last years didn't damage, it is my continued willingness to teach and thus keep learning.

I must also express my gratitude to Seth Torres, Technical Director of HUSEAC, for his invaluable guidance with a multitude of musical and technical matters, from mastering to concert set-up.

Finally I cannot forget to thank Nancy Shafman and Eva Kim for their continuous help during this long period on and off campus, and for their kindness in answering any questions.

Closer to my research, I want to say some very special thanks to Andrea Agostini and Daniele Ghisi. Without their enormous work around *bach* for the last 14 years, my own research would have been simply impossible.

I want to extend my heartfelt gratitude to Örjan Sandred for his constant support and inspiration. Our exchanges around music and constraint algorithms have been among the most demanding I have had, and it is a real privilege to be able to present some of the truly musical tools he designed as an integral

part of MOZ'Lib.

Many thanks to Jean-Baptiste Barrière for his immense support to MOZ'Lib since the very beginning, for his valuable insights to improve the project, and the opportunity to experiment and learn with two captivating historical libraries of computer-aided composition (CAC), his own system *Chréode* and Kaija Saariaho's *Transkaija*.

I want to express my gratitude and pride for the few loyal friends who attended the CAC online workshops I held regularly for the last 2 years, in particular Billy Newman, Hervé Zénouda, Leonid Zvolinsky, and last but not least Juan Sebastián Vassallo whose help and feedback with MOZ'Lib has been invaluable.

Big thanks again to my HGNM friends, especially Adi Snir, Zeynep Toraman, Sonja Mutić, Julio Zuñiga, Elena Rykova, Christopher Lock, Lorenzo Troiani, James Bean… again in spite of the distance and missed opportunities, for their inspiration, rich exchanges and kindness.

And to all the dear friends of my cohort, Uri Schreter, Ganavya Doraiswamy, Edwin Li, Samantha Jones, Sarah Koval, Jacob Sunshine, Nina Baratti, Anna Yu Wang, and Elaine Fitz Gibbon. I always felt a little clumsy surrounded by such a group of brilliant people. Now that we almost all graduated, and starting new exciting chapters in our lives, I am feeling nostalgic for the brief period when we were the closest, standing together against many shared challenges.

To my old friends of *Unmapped*, I all miss dearly. Despite us being scattered around the world, I never completely abandon the idea of us all playing together again.

To Rafaël Carosi and John Bardakos, for our countless hours of discussions and experimentations working on *Game Theory*, for sharing that dream together, even for a short while.

To Jacopo Baboni Schilingi, and of course the whole team of the composition cursus I had the privilege to study with in Montbéliard, Gaja Maffezzoli, Lorenzo Bianchi, Giacomo Platini and Frédéric Voisin, along with my fellow students there in particular the famous « Groupe des Cinq ».

I want to thank my whole family for their continued encouragement and love, even though I keep moving farther and farther away from them. And in particular my father Luc Dubos, my first music teacher, who always slipped me the book, the film, the disc… or the instrument I needed just at the right time, and who taught me the real value of *laziness*.

Finally, this work is dedicated to my dearest Shan 倪杉, my partner in life, my true light in the dark, for her endless love and support, for making space for me in Wuhan which I can now call my second home, for bearing with me in the lowest lows — and also now my closest collaborator, for her constant feedback and help making impossible choices, for the weirdest things we will create together.

what is music, if I forget everything (I think) I know about sound and art?

music is not space

music is not energy, it is pure information
a hallucination

something that my brain fabricates (something it can do)
and that is shaped by my attention
invoked/revoked by my judgement (and beliefs)

something that envelops me, either caressing or with sharp edges
like an animal
or a virus

something that might die before me, something that I let live then,
but which might still try to kill me

music is an idea of a life form

Foreword

The following text is going to be relatively technical, hopefully not in the obscure or cryptic sense of the word, but in the sense that I need to discuss my practice, tools and ideas in a precise way. The challenge is trying to describe these within a broader process, but somehow accessible for everyone without the full background. Technique is at the core of my way of thinking and working, so I cannot artificially divide this text into three distinct sections such as : Part I) my ideas vs. Part II) my tools vs. Part III) my music. Instead I will try to blend them together to maintain conceptual unity. My ideas fuel my tools constantly, and my tools are my ideas actualized at the same time into something localized, (fully or partially) static like fossils, and something dynamic — a network in which other ideas and tools can grow and navigate, which I can operate when I compose music. Together they form my *technical milieu*[1], an "artificial envelope" superimposed between me and the world. Protecting me from the world one might say.

Therefore when describing those techniques and ideas I will try to remain complex yet hopefully digestible and not overly complicated. As I strive to do in my own teaching, I aim to prevent the perpetuation of any monopoly over technical knowledge. Sharing ideas and techniques, trying to demystify a discipline which is clearly of minor public interest today, yet I feel can be of vital importance, is for me a way to repay my masters. It is also a modest contribution to try and defend a form of *technodiversity*[2] in the field of computer music, in a mono-technological culture whose attention is suddenly entirely focused on the machine-learning-driven AI paradigm.

One term will recur frequently throughout this text, the main area of techniques and thinking called « computer-aided composition ». It will almost systematically be abbreviated into CAC (similarly to the French CAO for « composition assistée par ordinateur »). I will also use the common abbreviation of DAW for « digital audio workstation », the most common working environment used when composing electronic music, such as Reaper, Cubase, Logic, etc. Please note that English is my second language, so my writing may be straightforward and unflourished. However, I will strive to be as clear and precise as possible.

---

[1] The term was coined by archaeologist and anthropologist André Leroi-Gourhan, specialized in the relation between technics, culture and aesthetics. cf. André Leroi-Gourhan, *Milieu et Techniques*, Albin Michel, 1945-1973

[2] This concept was proposed recently by philosopher Yuk Hui : « Instead of a universal history describing one technology with various stages of development, we can step back for a moment and instead describe technological development as involving different cosmotechnics. I call this technodiversity. Here, we must revisit the question of locality […] in terms of systems of knowledge. Michel Foucault called knowledge systems epistemes and understood them as ways of life — ways of sensing and ordering experience, producing in turn certain forms of knowledge. » cf. Anders Dunker, « On Technodiversity: A Conversation with Yuk Hui », in *Los Angeles Review of Books*, June 2020. See also Yuk Hui, *The question concerning technology in China: an essay in cosmotechnics*, Urbanomic, 2018

Part I

Memory, writing & code

The initial tension in my activity of music making is between the *written* and *non-written*. My first instrument was *arpa llanera*, which belongs to a musical culture in which I had no family ties (if not of heart), mostly of oral tradition and with strong propensity for improvisation. As a result, I did not start learning how to read music until I was 17 — a quite painful process — under the guidance of a jazz piano teacher. I didn't study counterpoint and barely touched classical harmony, which I forgot instantly when I fell in love with the colors and theories of modal jazz.

My practice of music creation truly began during my undergraduate years in Paris with the *Unmapped* collective, which I co-founded in 2007. There I started to use the computer as my primary instrument, and abandoned the harp and piano entirely. At that time, I was just getting familiar with acousmatic music composition, computer music and programming with CSound and Max[3]. The same year, I suddenly became fascinated by the sub-discipline of computer-aided composition, which I discovered with a software called PWGL. For the barely literate musician I was, this new and mysterious tool felt like the key to a forbidden kingdom I would never dream of entering. It is only after a few years of arduous study, with PWGL serving as my primary sandbox for experimentations, that I started to produce some instrumental music, such as *Ascidiacea* for bass flute or *MOrphism* for small ensemble. Since 2011, I have fixed my attention on mixed music as a way to coalesce all my interests : electronic music, instrumental composition, algorithmic composition, computer-aided analysis, programming in general…

It may seem paradoxical, given the little confidence I have in my relationship with written music, but I do believe writing can do certain things that non-written music, memorized or improvised, cannot. I see writing here in the broadest sense of the continuum — text, score, recording, code… — independently from any music genre. As demonstrated by Jack Goody[4], writing first emerged as the accountant list, reporting agricultural produce, prices and debts for the ruling classes. The music score is an updated

---

[3] Max (also known as "MaxMSP") is a visual programming environment conceived in 1985 by Miller Puckette while working in Ircam (Paris). Since 1990, Max has been developed and commercialized by private company Opcode and later by Cycling'74, under the direction of David Zicarelli. Concurrently, Puckette has developed *PureData* (Pd), an independent, open-source environment with a similar philosophy.
Max allows musicians, artists and researchers to create their own programs, called *patches*, with minimal programming knowledge, by simply connecting building-blocks with virtual "patch cords". The environment is highly extensible by design : the large community of users often enhances the base language with third-party libraries or "packages", including the *bach* ecosystem.

[4] cf. Jack Goody, *The domestication of the savage mind*, Themes in the social sciences, Cambridge University Press, 1977

version of the same technology : essentially a horizontal list of those things we attach by convention to certain sounds, textures, gestures, figures. A list (or worst, a *spreadsheet*), and nothing more…

And yet, in the eye of those who know how to use it, it becomes so much more than that : a powerful *dispositif* allowing them to operate music, and above all that helps them think music. The score can be traversed in multiple directions, ordered or reversed, cut into sub-lists, filtered, analyzed, compared with another scores, etc., just like the list of the accountant thousands of years ago. It is powerful, but limited, basically limited in that its execution is bound to the eye movements of the reader, the tension-latency of everything happening behind those eyes, and the hand movements of the writer, who is in our case the same person.

I believe the computer brings an extension of the writing power of the music score, just like the score was an extension of the expressive power of memory and improvisation. By that I do not mean computer-based engraving, but the encapsulation of improvisation and writing processes into some higher order apparatus, analog to the continuum described by Katherine N. Hayles in her text *Speech, Writing, Code*[5]. All three co-exist and overlap, yet correspond to three distinct realities. Beyond speech, I think the core idea from the initial system was "working from memory" or being able to operate on-the-fly "in working memory", to articulate musical entities old and new. Above all, the computer is a *machine à écrire*, which translates literally to "typing machine" but which I prefer to think as *writing machine*, *machine that writes*, *machine to write*…

These three dimensions of music kept co-existing in some capacity in my practice. Until 2017, when I moved from France to the U.S., there was always a hermetic distinction between improvisation and composition in my work. This may sound obvious, but composing was always a solitary activity for me, while improvising was always a collective one. In a sense, I believe "improvising around" a score I wrote with musicians in a rehearsal situation does belong to the second category. In the first one, I always felt more free, but also weaker, undecided, while in the latter, I tended to naturally blend into the collective, feeling protected and somehow stronger. Consequently, I would allow myself very different things when I composed than when I improvised with *Unmapped*. It seemed that moving from one to the other, the list of my preoccupations, even what I would call my "style", would shift dramatically. But

---

[5] « Speech, writing, code: these three major systems for creating signification interact with each other in millions of encounters every day. Each, as it has been theorized, comes with its own worldview, associated technologies, and user feedback loops. In the progression from speech to writing to code, each successor regime reinterprets the system(s) that came before, inscribing prior values into its own dynamics. Now that the information age is well advanced, we urgently need nuanced analyses of the overlaps and discontinuities of code with the legacy systems of speech and writing, so that we can understand how processes of signification change when speech and writing are coded into binary digits. » cf. N. Katherine Hayles, *My Mother was a Computer: Digital Subjects and Literary Texts*, Part I.2 « Speech, Writing, Code: Three Worldviews », The University of Chicago Press, 2005

ideas and techniques kept circulating between the two contexts, fueled not only by my own practice but also by those of my fellow members in the collective.

In a sense, my activity felt *balanced* during that time, dividing my life between those two complementary alternatives. I would compose less frequently, but each piece served as a new *marker* along the road, both a challenge and a summary of what had happened since the previous one. On the other hand, improvising was a more consistent ritual of experimentation throughout the year, maintained by the pure pleasure of spending time with friends and playing music together. I kept growing as a musician within this group for the 10 years that followed its inception. However, when I moved to Cambridge, most of *Unmapped* members were already scattered around the world and this fragile balance was already gone. Since then, I have found myself improvising only on rare occasions. However, this aspect of my practice resurfaced in a different form within my research, as part of hybrid projects such as *Game Theory*, which I will describe in the second half of this text.

Hiding my intentions : conceptual vs. technical framework

All that I attempt to express, the *things* that I put in a given piece, can be summed up by a network of « semantic » connections between my own life experiences, body states, personal struggles, relations with loved ones, and various sources of knowledge I am interested in at the time[6]. They are first hinted at by a title, which I generally need to choose before I can start composing. I can identify a couple of invariants since my earlier works — forms of life and their behavior, memories and their corruption, etc. — but they tend to dilute over the years. After the title, this network manifests itself by developing for each piece a set of narrative ideas, key words and imitative behaviors that are usually quite archetypal, *banal* perhaps, with inspirations ranging from philosophy, neurosciences, politics, cinema, experimental poetry, science fiction novels, video games, etc.[7]

I believe those things are my personal concerns, and only mine to know fully. They are *private matter*. The only persons with whom I think I can share about them are very close to me, people who I can trust will not judge how naive those ideas can be. This often includes the musicians I collaborate with on a given project, when conditions allow to develop a deeper understanding. The "secret" has to be shared with them, so they can know intimately what the piece is about, and try to inject it into the work in non-verbal, often unconscious ways. Apart from this small circle, I usually prefer not to display those intentions, those concepts and metaphors publicly. Beyond the risk of momentary embarrassment, it is also a matter of time : it often takes me a few years to understand what a piece is really about, and it

---

[6] I must say I am not particularly interested in the idea of the survival of contemporary music as a *subject* for my music. By that I mean, writing music that aims at enhancing the audience's listening abilities, or transmitting « new » notions or metaphors about its parameters, is not my motor for composing. As a neophyte, I don't think many examples of such music attracted me to this repertoire in the first place. They were ways of musicking — musics, plural — which behaved in certain (unusual) ways, and seemed to tell me something about the world I lived in, and myself in it. Pieces that impacted me the most often drew connections with unexpected areas of knowledge, but I don't think their poetic goal was to fill gaps in the audience's incomplete sensitivity.

[7] Beyond titles and semantic networks, a more marginal interest of mine has been to try and draw connections between different projects. I have been fascinated by the idea of setting up *cycles* of pieces since I discovered Gérard Grisey's *Les Espaces Acoustiques* and especially *Quatre chants pour franchir le seuil*. To be honest, I was never able to complete any of the few cycles I started, but my interest remains. Cycles allow to bring a certain sense of global narration, consistency and perceptible trajectories across different works — a medium-long term design somehow analog to the "concept album" in popular music.
This is what I attempted to do with one of my first acousmatic pieces (*Croisement Irréversible*, 2007), composed during the terminal illness of my mother. Later I returned to this idea with a diptych of mixed music pieces, *Mémoire de l'eau* and *Fire&Forget*, which I hoped to turn into a full cycle called 五行 Wuxing, referring to the five movements or "elements" in Chinese cosmology. During my first year at Harvard, I discovered the fascinating practice of the gloss in Islamic philosophy, accumulating centuries of interpretation and disputes on the same page like layers of tree bark. I planned a cycle for string quartet called *Hashiya*, that would grow like commentaries around its core, the *Matn* (composed in 2018, see portfolio of scores). More recently, two projects I also intended as cycles were *Object-oriented trilogy* for trio (which became the fixed media piece *Objets de désir illicites*) and *Game Theory*, one could imagine as a hyper-cycle or cycle of cycles, project I abandoned in 2021.

usually diverges significantly from my first intentions. Even then, I strongly believe my position as the artist doesn't guarantee I have the right interpretation of the work I am trying to create, on the contrary. I think the piece itself, its title (and when one cannot avoid it, its program note), the recording and score, and a few clues hidden here and there, are largely sufficient.

On the opposite, I could keep describing over and over all the technical problems that a given project represented for me, how I tried to solve them, when I failed to do so, the tools I developed along the way, etc. In other words, my practice[8]. Now this probably can feel as a limitation in my way of thinking, reducing my music to "technicalities" and neglecting the conceptual dimension. I don't think so. Whenever I compose a new piece, or when I try to analyze pre-existing music, I necessarily take a technical point of view, and that includes conceptual aspects as well. I want to defend this posture. Even though it might not seem "clean" or "noble" to reveal exactly how music is made, I believe that understanding the *how* is essential before discussing the *why*. Without this initial step, any discussion may lead only to misunderstandings. Speaking right away about abstract concepts, "intentions" and "discourses", even "the form", all those things remain for me terribly vague. I wish we took more time to determine some form of technical framework to share beforehand. This is why I always ask colleagues to see their scores, before I say anything about their music. This is for me the strict minimum to be able to follow and understand the music better, as I listen. And yet very often this simple request is evaded or welcomed with some kind of worry, even suspicion.

I believe techniques are not only tools to produce a result. They are life forms, bearing

---

[8] Here I would like to share a larger excerpt from a public lecture by linguist, philosopher and former president of Sorbonne University, Georges Molinié (1944 - 2014), who I assume is not well known in the English-speaking world. Specialist of semiotics and French stylistics, he gave the following, rather unusual speech to an audience of musicians / pedagogists, where he defended the *practice* of creation over artistic *result* with some interesting arguments :
« The first assertion that I will stand by is that in my theory, art does *not* exist. This, in fact, will make our lives easier, you will see. […] For me, it is a question about language, or languages. What in other realms we call semiotic activity, *semioses*, or whatever, thus languages. It is all about languages, whether these languages are made from extremely structured, delimited materials — the organs of phonation for what we call verbal language, or different kinds of sound processing, even though that's also sound — whether it is made with color, movement… These are all semiotic materials, which define behaviors and these behaviors can be called languages. [...] In my approach, there is only what is *artistisable*, ultimately there is no art *per se*.
With the word 'creation,' I feel a bit more comfortable. […] For me, 'creation' would oppose '*œuvre*' (the work as piece, *opus*). It would oppose 'object' and it would even oppose 'result'. I will consider 'œuvre', 'object', and 'result' as fixed, given, produced elements, but what interests me is *how* we get there. We can call it creation, or *poiesis*, why not, the term is not bad. [...] It seems to me that only then, we would have a thought of action. Not a thought of the thing, but a thought of action, a thought of *praxis* [...] If we return to a Marxist reading of *praxis*, the idea of *travail* (the work as labor) and the idea of transformation. [...] Learning is about know-how, and know-how is about gestures, to reproduce or to produce. [...] Thus, creation, as *praxis*, consists in labor, therefore something laborious, painful, which is an effort, truly an activity that unfolds over time, and thus transforms, but transforms what? [...] It is a labor that consists in, laboriously, with a lot of effort and attempts, to transform the relationship of subjectivity to externality. [...] Necessarily, the horizon of this *praxis* is materialization. It is materialization, it is sensitization, but we should insist on the '-tion'. It is something that is *in the process* of being materialized, *in the process* of becoming sensible. »
cf. Georges Molinié, « L'enseignement de l'art à travers la création - une perspective philosophique », Rencontres *Créer ce que l'on joue*, Public intervention in MPAA, March 8-9, 2010.

personality. They can generate or transform materials according to set expectations, give rise to singular properties, a “coloration” (or a “flavor”) that reflects a specific reduction in our perception. They are also full-fledged perceptive systems, able to filter our own perceptions in manners unique to each of them : ways of listening or seeing later become ways of segmenting, structuring a form. They evolve, disappear one day… unless they can be preserved and keep evolving — including all the means they can control us with. As such I think they should always be disclosed.

(Mixed) Music and Magic

Now I would like to explain why mixed music remains my preferred means of expression as a composer. By this term I mean any piece of music combining in the same acoustic space instrumental sound and some form of fixed, generative, or interactive real-time electronics. In my case it corresponds essentially to music for instruments and computer, and can exist either as a live performance or as a recording. In the latter case, it amounts to being a subset of acousmatic music including some instrumental layers.

At first, I had a tendency to explain this inclination by referring to time scales. As a younger composer I often felt my electronic music — i.e. my first few attempts at pure acousmatic works — missed something that prevented it to function properly. Constantly zooming in and out in my DAW session, I was free to explore the whole continuum of time scales in which sound objects could exist, from micro to macro[9]. Yet I always found myself unable to compose with them in a convincing way, precisely at the more restrained scales traditionally inhabited by instrumental music. I needed certain music entities or *identities* — rhythms, figures, textures, gestures or higher-order combinations — to exist through instruments rather than electronic sounds.

I recently came across an article concerning the last album of electronic music artist Chris Clark, whom I admired for many years. The following commentary by Thom Yorke, well-known leader of Radiohead, caught my attention :

> *Chris wrote me to say he'd started singing, looking for feedback/advice or whatever, cuz it was kind of new shark-infested waters for him.* […] *To me the way he approached it all wasn't the usual singer/songwriter guff thank god; it mirrored the way he approached all his composition and recording, but this time it had a human face. His face.*[10]

This idea of music "having a face" completely resonated with my own work. Perhaps my music did need a face, even a rather caricatural mask resembling a face, to exist among others. This I believe I found in the practice of mixed music, which mainly brought me a sense of presence, or frontal identity[11]

---

[9] cf. Curtis Roads, *Microsound*, MIT Press, 2004

[10] cf. M.I. Ruiz, « Clark Announces New Album Executive Produced by Thom Yorke », news report on *Pitchfork.com*, January 2023 (accessed June 2024)

[11] This intuition later came into resonance with the video essay *Peak Face* by M.C. Abbott, María Buey González and Carl Olsson : « Terraforming is the process of intelligence transforming its surroundings at the scale of the planet. The *Facial Phase* proposes that the terraforming has a period that has been dominated by the face which began around the Cambrian Explosion (~541 mya) and has lasted until now. The face began as an evolutionary adaptation that collected sensory organs at the front-ends of animals. From this perspective, the face is primarily a platform for changing the environment and only secondarily an index of human identity. » Project developed as part of Benjamin H. Bratton's research program *The Terraforming 2021*, Strelka Institute for Media, Architecture and Design in Moscow, Russia. cf. https://www.youtube.com/watch?v=rcfp25LaWjI (accessed June 2024)

that I couldn't find without instruments. This brings me back to Pierre Schaeffer's concept of the *allure* of a sound, later summarized by Michel Chion as « what reveals what the agent of [a sound's] energy is, and whether this agent is living or not »[12]. The allure is essentially an indicator of behavior, either pure predictable regularity (machine, program), supple periodicity (living agent) or unpredictable irregularity (natural phenomenon). Just like acousmatic music, mixed music works around that tension between the three poles of allure ; but by design, it is also accentuated by the irruption of the living agent — usually human musician(s) — into the fictional space.

Another interesting notion to discuss, seldom put in parallel with mixed music I believe, is that of *augmented reality*. If we look at the different usages of this term today either in digital arts or main stream media and technology, they can feel extremely close phenomenologically speaking with what composers have been doing with mixed music for nearly 80 years. That is, if one allows me a very approximate description, adding a layer of fiction on top of physical reality, environments, objects or human agents. Obviously the distinction is quite debatable if one considers the reality of projecting sounds via loudspeakers, when compared with the more "natural" projection of a voice or instruments — both can be seen as equally real or fictional depending on the point of view[13]. When working with mixed music, we are never really far from setting up a magic trick. We want the audience to believe in some kind of acoustic and conceptual illusion, either a seamless continuity between the physical world and the virtual world, or an artificial dialogue between an instrument and its own shadow. For instance, capturing the sound of an instrument and freezing it indefinitely while the musician already moved to the next phrase, if well executed, remains one those satisfying tricks in the toolbox of mixed music.

In fact, mixed music is often criticized for these very reasons, because it seems to simplify, even caricature the relations which would normally happen between human agents, i.e. between individual voices or groups of an ensemble, between a soloist and an orchestra, etc. In the case of my own music, I do have a tendency to remain safely in a *concerto-like* type of hierarchical and causal relationship

---

[12] cf. Michel Chion, *Guide to Sound Objects*, translation by John Dack and Christine North, 1995-2009. Text available freely on monoskop.org (accessed June 2024)

[13] Digital artist Maurice Benayoun seems to argue for a continuum between those ontological layers : « *This is not cut off from the world* — The same technologies which have enabled the immersion of the visitor have also made possible *fusion* between fiction and reality. After the sudden appearance of reality in fiction (reality shows, news shows, documentary dramas etc) it is fiction which has slipped into reality. Beyond the extremes of representation, of reality shows, of excessive broadcasting of events, the physical space, often situated in town, has become a support/medium to be worked on in detail. Mixed methods, increased reality, geo-localisation and urban action have become artistic playgrounds on a totally new scale. » cf. Maurice Benayoun, « Art after technology », *in MIT Technology Review*, 2008

between instruments and electronics. This is a personal inclination, most likely geometric[14], and from it results a set of more or less predictable behaviors. For example in the case of real-time electronic music, when the apparatus is hardcoded in a certain way, it can only react *to* (and not interact *with*) actions made by a human musician, not the other way around. In that context, playing into a reverberation effect or some elaborate improvising machines can be considered analog. For some expert listeners, this is a non-satisfying trick. But it is a *writing* problem, i.e. a momentary state of fact resulting of aesthetic choices and time spent programming, not a weakness specific to mixed music as a medium. For a given instance of music, perhaps the composer didn't think (or want) to make the instrument capture some electronic sound they heard and freeze them indefinitely, as a reciprocal gesture towards the machine. The possibility exists — a human composer could also write such predictable causal behaviors between two instrumental voices — but the necessity to adopt it is debatable. In fact one might find some hidden ideology there : the human agent must shine, and get out somehow "victorious" of the interaction, while the artificial agent, the technological apparatus (fixed media, patch or more elaborate) must act as in a state of servitude, verging on servility. All in all, it is a matter of awareness, which is easy, and choice, which is always more delicate.

I must confess that while these parameters are more important to me when I set up an improvisation patch for a performance, they do not play a determining role in my written music. Of course they do exist to some extent, and I am aware of them, but they are clearly not my main motor for composing. When I describe my mixed music, I almost always start with the instrumental part, the score aspect of it, and I often neglect the electronic side. This is perhaps because I seldom felt the need to theorize on this aspect of my practice — working in Max or in a DAW are both a place of total freedom for me, or at least I like to perceive them as such[15], and I prefer to avoid compromising this belief with a surplus of theoretical tension. Because I feel it is more precarious and fragile, more problematic and also more personal, the composition side of my work in general demands me much more analytical and technical energy, also much more time. In a way, the electronic side of my mixed music inherits from my

---

[14] I do believe the perception of symmetry in electronic composition and especially in mixed music can play a role to some extent in setting up hierarchical relations in the composition itself. If one thinks about placing a sound object or a soloist in the middle of a set if speakers (usually organized in even numbers), there is not an infinity of possible combinations. Just like the relative bilateral symmetry of our bodies may require some kind of a conversion or synthesis from *stereo* perceptions into *mono* concepts, I suspect that for some composers like myself, returning eventually to the center can amount to a spatial equivalent of the perfect cadence. For a most remarkable illustration of this, I would refer to Mark Fell's short track *2-A* from his album *Multistability* (Raster-Noton, 2010)

[15] For a discussion on the impact of the digital workstation as a *dispositif* for music creation — « visual processors as much as sound processors » — a critique of its underlying politics and in particular the illusion of democratization (which I came from) and resulting "uberization" of music production, cf. Michael Terren, « The Hegemony of the DAW », for *Disclaimer.org.au*, 2021 (accessed June 2024)

improvisational character, trying to capitalize on the (supposedly) solid architecture I first establish with the instrumental part, flowing around it with reduced supervision. When I compose in a DAW my degree of confidence is much higher, because I can listen to the music over and over, and without ambiguity. The repeated exposure allows me to reach the deeper intimacy I need with the materials in order to establish connections between them. The final result is an accumulation of modifications performed until the critical moment where I honestly believe it cannot be improved — or until someone else proves me wrong by *showing me* how to make it better.

On the opposite, I doubt 100% of the time when writing a score. What does this pitch sound like? Is that duration really what I think? Even assembling some instrumental part from MIDI snippets or improvised recordings in the DAW, I feel often undecided and powerless[16]. When I zoom out here, is this phrase not kitsch? Is that sound really appropriate here? Won't I cover myself in public shame by using this chord after that one…? This is why using the computer to assist me — even sometimes guide me — through the writing process, was always the initial condition for me to be able to successfully produce mixed music.

[16] A more extreme example of this was the composition process of *Hyperlinks — asset(s) failed to load properly*, one of the most difficult projects I ever encountered. After three months elaborating the instrumental part in the fall 2023, and a couple of weeks of recording sessions with the flutist and guitarist in France, I came back to China in January 2024 with more than 30 hours of materials. It took me more than five months to assemble the instrumental part only. Certain sections required rather meticulous work, following the score very closely, which was time consuming but manageable. But for others which I anticipated to be more free and spontaneous, closer to my own improvisational / acousmatic practice, I felt the most paralyzed I ever was in front of a DAW session. It was only thanks to the regular interventions of my wife Shan, re-discussing my intentions, helping me make choices and delete irrelevant materials, that I was finally able to make progress. This is a clear example of a blind spot in my composition framework, where I couldn't think of a way for the computer to help solve my situation, only the care of human connection could.

CAC and augmented imagination

Between 2014 and 2017 I reached a state in my work where I was feeling, for some time at least, a real balance between the few poetic ideas I had and my capacity to produce patches that could make music in a timely manner before my little energy towards those ideas faded away. I wrote few pieces at that time but they were exactly what I wanted. During that period my highest level of pleasure when writing an "instrumental part" was when done in a manner equivalent to writing tape music : assembling and editing MIDI and audio clips in the DAW, with immediate auditory feedback.

Obviously, being unable to compose without hearing what I am doing reduces drastically the amount but also the *kind* of music that I can actually produce. Very few virtual instruments available today are satisfying enough, trying to imitate real instruments, that I can "suspend my disbelief" and trust what I am actually doing *could* be played by a human musician. It limits my possibilities in term of instrumental sounds (i.e. playing techniques), in term of chaining those sounds in a realistic way, and in general in receiving what I would call a *quasi-expressive* response to my experimentations[17].

Yet I remain a child of the *home studio* generation — I felt my first emotions as a "composer" in front of *Garageband* and *Reason* in 2004, with a MIDI keyboard connected to an iBook G3. Before that, as a late learner of piano and solfège, all I felt was latency and heaviness. No matter how much respect I have for people writing scores at the table, I could never learn to imagine music with only paper and pencil after that. Yet I found another kind of latency and heaviness that would allow me to reach a higher level of consciousness and production for my music, by adding CAC tools in the equation.

What I call "computer-aided composition" today is a sub-discipline of computer music, with programming environments and tools allowing to calculate musical structures and to represent them into some kind of actionable notation. Many early CAC environment such as Ircam's CHANT/FORMES, as well as PatchWork and its successors[18] were based on a programming language called Lisp. The *bach* package for Max, a more modern approach of CAC, doesn't rely on Lisp but borrows a lot of its concepts

---

[17] In my experience sample-based libraries, which treat practically every sound from any given instrument as an isolated "percussive instrument" (like *Contimbre* or *Ircam instruments*), cannot produce an image I can trust enough to compose with. So far, my only luck was with virtual instruments based on physical-model synthesis, which is still extremely rare in the industry (see Sample Modeling, Modartt, etc.), and offers a small amount of instruments with a limited range of playing techniques. This is one of the reasons why a big portion of my instrumental music is mostly made of notes, and it will remain so until more open, expressive and flexible tools of "score to sound" translation appear in the public, perhaps in the near future as generative AI keeps developing.

[18] *PatchWork* was created by Mikael Laurson (Sibelius Academy, Helsinki, Finland), then further developed in Paris' Ircam by Camilo Rueda and Jacques Duthen. The environment was later refactored by the Ircam Musical Representations team into what would become *OpenMusic*. In 2002, Mikael Laurson and Mika Kuuskankare introduced *PWGL* as a modern alternative to OpenMusic, with a strong emphasis on constraint programming.

from previous Lisp-based environments. Lisp was designed by John McCarthy and his team at MIT in the late 1950s. The language was originally designed to perform higher level symbolic computation — LISP stands for "List Processor" — which means it could generate and manipulate non-numeric data more easily than other languages at the time. By design it was particularly suited to write recursive processes (functions calling themselves indefinitely), and the symmetry between its code and data[19] made it easy to write programs capable of generating other programs. The language was widely used in the first decades of artificial intelligence research, before its decline during the 1990s around the time of the second "AI winter". It is not entirely clear why Lisp became popular in the computer music world in that period, and why it remained around for so long, but one might suppose its inherent facility to represent complex, nested musical structures was one of the main reasons.

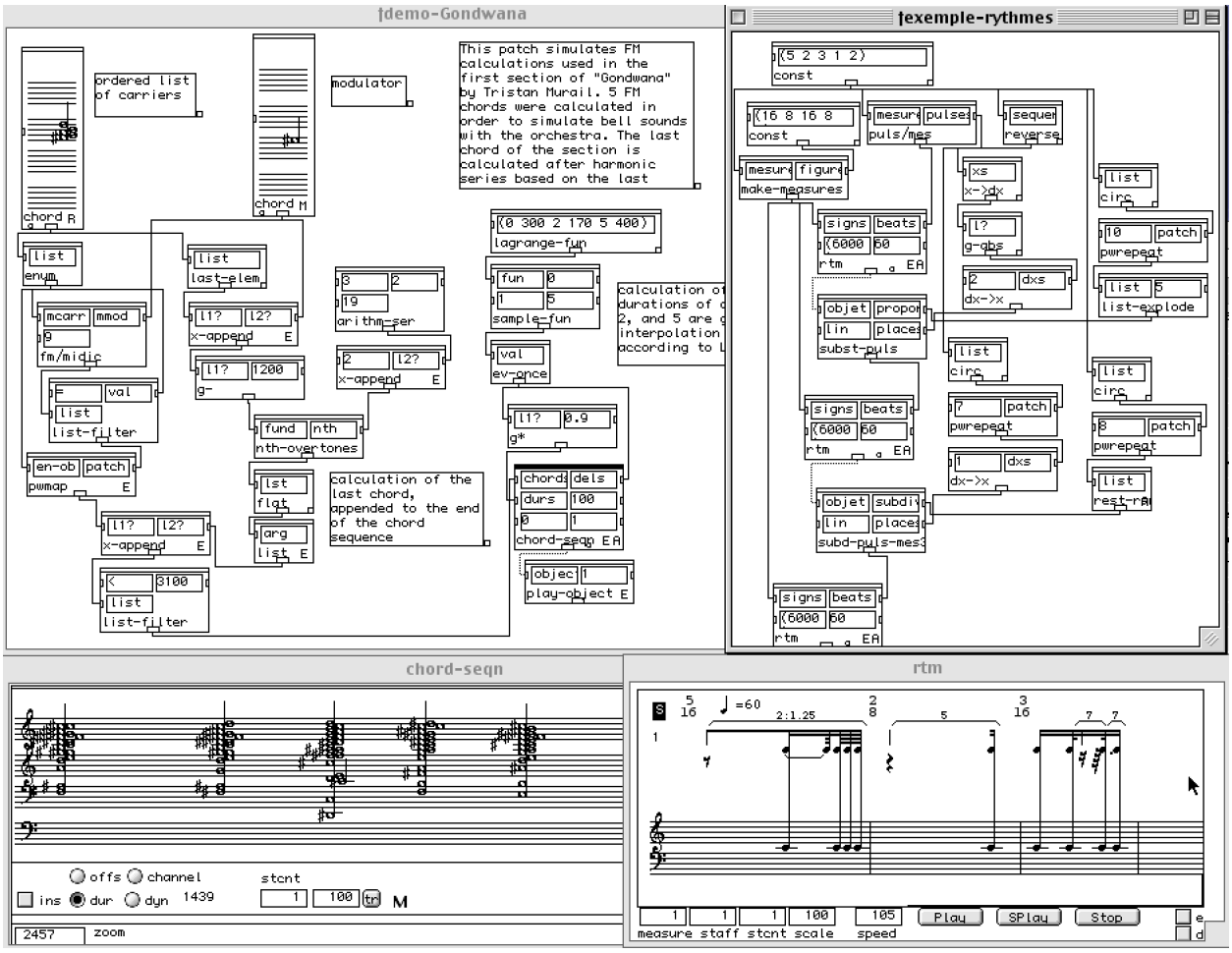
In the early 1980s, programs like CHANT/FORMES required the user to interact with pure code, command line inputs, and only rudimentary graphical interfaces would allow to display a musical result. More often than not, composers needed the assistance of professional programmers, if not for day to day usage, but certainly to produce Lisp programs exceeding intermediate levels of complexity. By the end of that decade, PatchWork allowed composers to build their own CAC programs by connecting bits of Lisp code visually, with patches similar to Max or PureData. Programming was not only made more easy and accessible to composers of various aesthetic and technical backgrounds : these environments gave rise to radically new ways of interacting with music ideas. As I already wrote elsewhere[20], I count myself among those who learned composition with the help of such CAC tools. More specifically, I studied composition in parallel of studying CAC, while composers of my generation or younger usually study composition for many years before they even hear about the most famous option, *OpenMusic*. When I was a student in the Conservatory of Montbéliard, CAC was not merely a music calculator for the composer's toolbox. In these classes, students were encouraged to think together with the help of CAC. The software, PWGL in this case, was a hyper-expressive means to learn how to conceive and describe our musical ideas. Patches were for us a way to share and confront ideas, and to solve musical problems lying in front of us that we might not have been able to express in words at first. The key question there was : "How do I write this?"

---

[19] The most obvious difference between Lisp and other programming languages is its systematic use of parentheses to structure the code like a tree. This makes Lisp a rather strange language, considered as more difficult to learn. But it also brings its greater strength : code and data are interchangeable. For instance the following defines a function to calculate the factorial of a given number n :
`(defun factorial (n) (if (= n 0) 1 (* n (factorial (- n 1)))))`
so that `(factorial 4)` returns 24 and `(factorial 10)` returns 3628800.
Similarly, here is how PatchWork would represent a rhythmic tree :
`((1 (1 1 1 1)) (1 (1 1 2)) (1 (2 1 1)) (1 (1 2 1)) (1 (3 1)) (1 (1 3)))`

[20] cf. Julien Vincenot, « On 'slow' computer-aided composition », in *OpenMusic Composer's Book Vol. 3,* Éditions Delatour – Ircam-Centre Pompidou, 2016

CAC made me reconsider the way of thinking about detail while creating music : written detail did not necessarily have to be realized entirely by the composer's hand, because its properties could be declared algorithmically. Multiple aspects of the music could be modeled before it even existed. A given idea was not merely an instance written on music paper, but a small system that could be explored interactively, with immediate visual and auditory feedback. On the other hand, more space and time was left for compositional detail, making artistic decisions on a multitude of time scales and musical parameters, just like in acousmatic composition.



Examples of composition techniques realized in the original *PatchWork*
after ideas by Tristan Murail (*Esquisse* library) and Brian Ferneyhough (*Combine* library)
(screenshot taken on MacOS 9.2 via *QEmu* in 2024, thanks to Paulo Raposo's immense archival work)

The link between CAC and electronics is in fact very strong traditionally, as many composers (some related to the spectral school) used it as a sort of unification tool, specifically to help bring together instrumental writing and electronics. For instance, one might want to control sound synthesis with the same compositional principles as the one governing some orchestral situation. In my own case, I must say

I rarely use CAC algorithms to generate electronics. Both sides do happen mostly in the same environment, MaxMSP (along with *bach*), and there might be some porosity and mutual influence between the two, but I do not consider them to be the same activity. I make a distinction between working with electronics on the signal level — or sub-symbolic level, were synthesis and sound processing take place — and the "coarser" symbolic level of the score. Again because I am interested in different typologies of materials and time scales in both domains, the language used to control them is fundamentally different. If I see CAC as a tool for unification, as we will see in later chapters, it is to bring together the different aspects of my practice of music creation. While all the softwares of the PatchWork family were "deferred-time" environment, poorly suited for interactive settings, *bach* is a fully-fledged CAC environment available in the a real-time context of Max. For this simple reason, it gives much more possibilities for the composer to create projects at the crossroads between instrumental / mixed music, improvisation, interactive installation, and collaboration with other media (video, text, etc.). All kinds of such hybrid projects were practically possible before, but real-time CAC brought the tools to express higher level music representations and behaviors, some "syntactic intelligence" which I believe was previously lacking in a software like Max.

I really want to emphasize on the existential quality of the relation I have working with CAC. I never felt any particular pleasure in front of a blank piece of paper score, or even in front of a blank document of a notation software for that matter, if not vertigo and confusion. But for more than 15 years now I was always excited by the idea of making a patch, testing a given idea, seeing how a result I couldn't write otherwise looked like, listening to it, all leading to more ideas, etc. Writing a patch is the only way I know to apply the necessary pressure on myself to overcome the infinite difficulty of writing a score for human musicians, with more or less conventional notation[21]. Like some of my colleagues, who might express a form of *graphomania* or uncontrollable impulse to write music on paper, I am a graphomaniac of patches and code. The difference is that as a composer I cannot really exist socially by

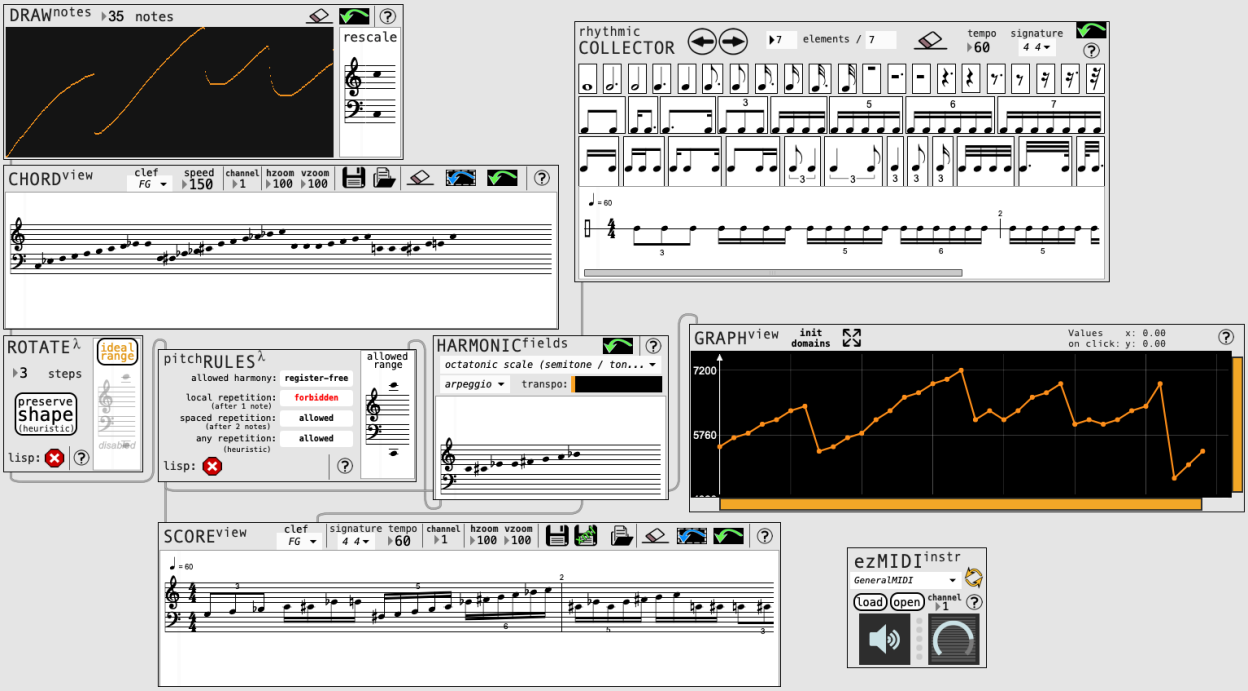
[21] Computer-aided composition is generally not considered to be the most appropriate tool to produce "unconventional" notations. Just like more mainstream engraving softwares like Sibelius or Dorico, it is always possible to "cheat" to some extent, and I have done so in quite extreme ways with *bach* when working on *Game Theory*, but it is and will remain limited in scope. Of course it is the *language* in the background, the encoding chosen by the composer which allows to manipulate certain typologies of music structures or not : *any* kind of music can be represented and manipulated that way. And in a sense, being able to produce a direct display as notation is accessory. But translating code into visuals by hand is always a lengthy process, and it is reasonable to want to reduce it with the help of the computer. When working on my unfinished piece *all the memories are rigged* in 2019, I made some promising experiments generating more free and elaborate notations with CAC systems via the SVG format. The results could then be easily exported to professional vector graphics softwares, such as Illustrator, Omnigraffle, etc. I did abandon this direction for the time being : composing with less conventional kinds of notation for instrumentalists implies a very different set of skills that I do not feel the necessity to develop for my music anymore, in particular a deeper knowledge of instruments' organology and idiomaticity for which I have only a limited interest.

publishing patches, first because only a minuscule portion of the audience might appreciate them, technically or musically, but also because many of my patches do not “make music” exactly in the same sense that a score “makes music” when given to a musician. Most of them probe a tiny portion of compositional space, hoping to find a vein of “something interesting” — but many fail to do so, and I keep the others in mind for the next “serious” project. I do not feel the absolute necessity of making a score, i.e. a piece of art to be showed in public, when testing any single idea. In many cases, making a CAC patch was actually enough. In fact the opposite is also true. When composing a piece, especially as a younger student, I would explore a multitude of small ideas with different patches : I would discard many of them along the way, finish to realize some others « by hand » in a notation software, but actually use only a few of them to generate materials algorithmically[22]. In fact very often I spend the time to make patches as a way to collect enough informations on an idea, so to speak, before I realize it is *not* what I need. Or perhaps I do use them eventually, only much slower, and less directly. Many programs I wrote, while playing with them, observing their behavior, like little machines trying to imitate “musical life”, infused my way of thinking about music, and I would perhaps return to them months or years later in another project.

A lot of my time over the last 7 years was spent trying to restore this pleasure I felt, originally, when composing with PWGL. This software was my main laboratory for music exploration since 2007, but for several reasons it began to slowly wither in 2016 and suddenly disappeared from the internet in 2020. This is mostly what the MOZ’Lib package for Max has been about, a salvaging project. I honestly did not anticipate to invest so much time and energy maintaining a software of this size, slowly porting all the tools I cared about from the PatchWork and PWGL eras, adapting them and documenting their usage specifically in the new context of Max. It is only since the last couple years that I am slowly starting to experience some *return* on that invested time, trying to revitalize my composition practice with those renewed tools.

---

[22] Jean-Baptiste Barrière describes Kaija Saariaho’s ways of interacting with her personal library *Transkaija* in the CHANT/FORMES environment, in the mid-1980s, in rather similar terms :
« At that time, Kaija used the program mostly to generate musical material, a 'pretext' for instrumental music writing. Very freely, she would often modify the result by hand, either for musical purpose or to follow instrumental constraints. For synthesis, on the other hand, she would be able to use the data much more strictly to control the Chant program (cf. pieces such as *Io*, *Amers*, etc.). On occasions, she would even juxtapose both approaches to create 'friction' between instruments and electronics. » In 2016, a portion of Transkaija’s code was reconstructed and re-actualized for real-time use, under the direction of J.-B. Barrière. It is available today as part of the official MOZ’Lib distribution. cf. Documentation in the MOZ’Lib Overview, under *PWforMax Guest libraries*.



Max patch using MOZ'Lib modules (cf. dedicated chapter below)

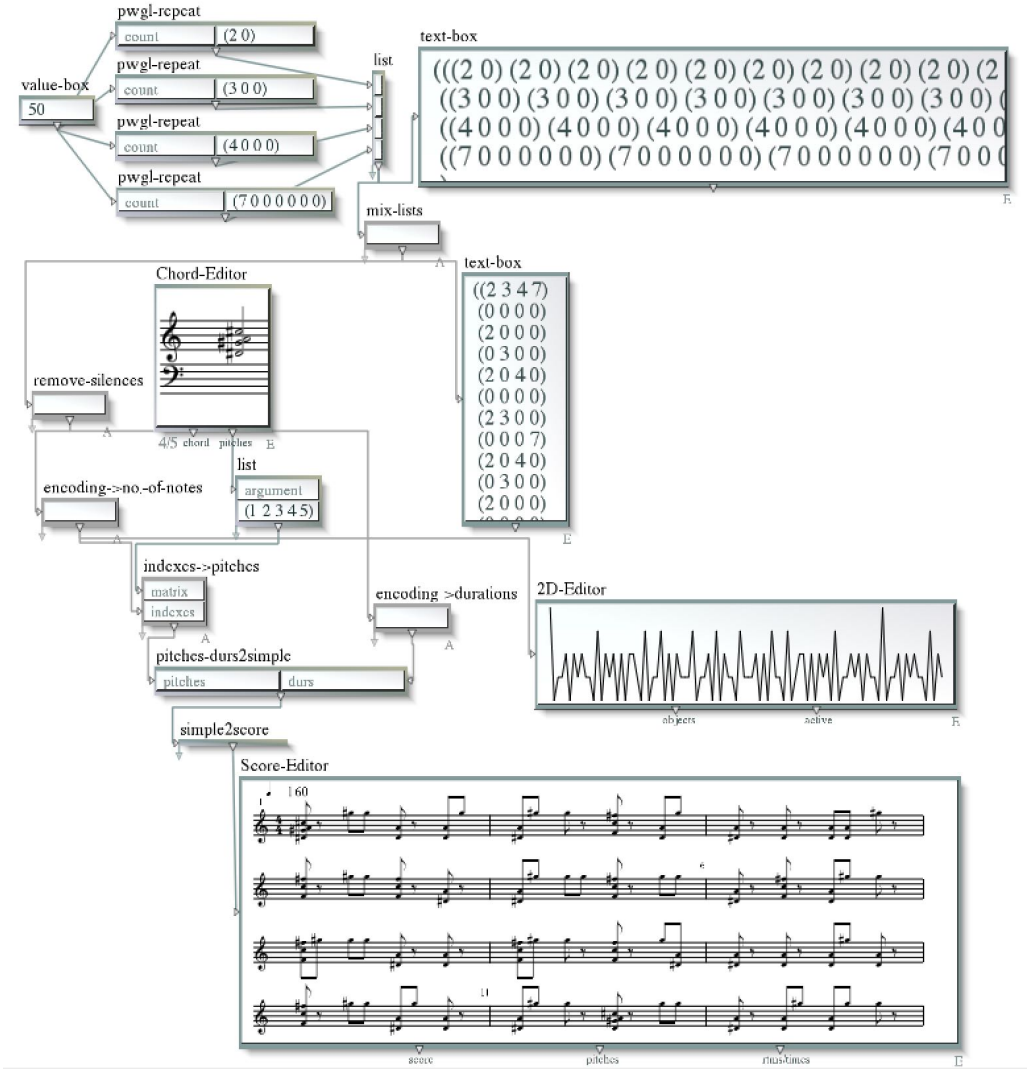
After spending several years teaching CAC on the side of my creation and research activity, including the continuous development of MOZ'Lib, I realized that the most important aspect to pass on was to be able to *represent* a problem accurately, to determine how its parameters should be *encoded*. First, the question of representation is fundamental in my opinion to be able to compose, and at the core of what CAC tools have to offer : being able to visualize an idea under a different angle is often enough to understand how to realize it sonically, or how to draw connections between different materials. This includes the display of some intermediary materials or results in somewhat traditional score notation. It can also mean being able to represent some of the properties of a music entity as some sort of graph notation (or BPF, "break-point" function on a X-Y axis), to highlight its behavior and directionality for instance, thus taking a more analytical distance with the materials. There is also the possibility to arrange a bigger corpus of data into some abstract space, sorted according to some given parameters (pitch register, duration, density or more complex perceptual descriptors like inharmonicity, noisiness, etc.) or by criteria of similarity described by the composers themselves. In this case, one not only understands better the *compositional space* they are working with, but also compose directly by navigating inside that space, making the writing process much closer to improvisation. Another way to represent the compositional space we should not forget, even if it might seem trivial, is the patch itself, similar to the

topological organization of music sketches on paper. If a typical real-time Max patch is usually given some kind of control interface, analog to an synthesizer or custom *instrument*, a CAC patch in my experience is also a space to lay out materials and abstract ideas, building up contextual and operational memory, like augmented music paper.

The most important facet of environments like PWGL or the *bach* ecosystem is that they bring a *language* to express ideas, to describe specific problems, invent isolated techniques or whole systems. In my opinion, they should not be seen as a simple set of predefined tools allowing composers to apply certain techniques to their own materials, without critique. Let me illustrate this with an example. As a student, I was fascinated by the idea of interpolation or metamorphosis — that is, transforming a given music entity A (chord or spectrum, melody, rhythmic motif, or else) into another one B in a certain amount of steps. This is a very typical example of CAC, analog to the *morphing* technique in computer visuals. Many composers of various obediences — like Tristan Murail, Brian Ferneyhough, Kaija Saariaho, Magnus Lindberg, Jacopo Baboni Schilingi and others — have proposed their own interpretations of this idea. I took some of those implementations for granted at first, even tried to use them in my music without success on a few occasions. But as fascinated as I was, I realized that I was not entirely satisfied with any of them, and started to make my own propositions of what "musical metamorphosis" could be, what it meant for me[23].

Finding the proper way(s) to describe a problem is the most difficult step. It requires to think about how a given *encoding* or *re-encoding* of the data could highlight the (musical, perceptual…) properties we are interested in, thus clarifying the relations we want to develop in the music. Very often in my experience, the most elegant solutions to a programming problem comes from changing the point of view, the way of representing the musical data that we try to generate or transform. And as a side effect, it often leads to some kinds of "epiphany", where one realizes unexpected connections between certain ideas of music making.

[23] I already described in details the reasons of my dissatisfaction with previous models, along with some concrete music examples. cf. Vincenot, 2016

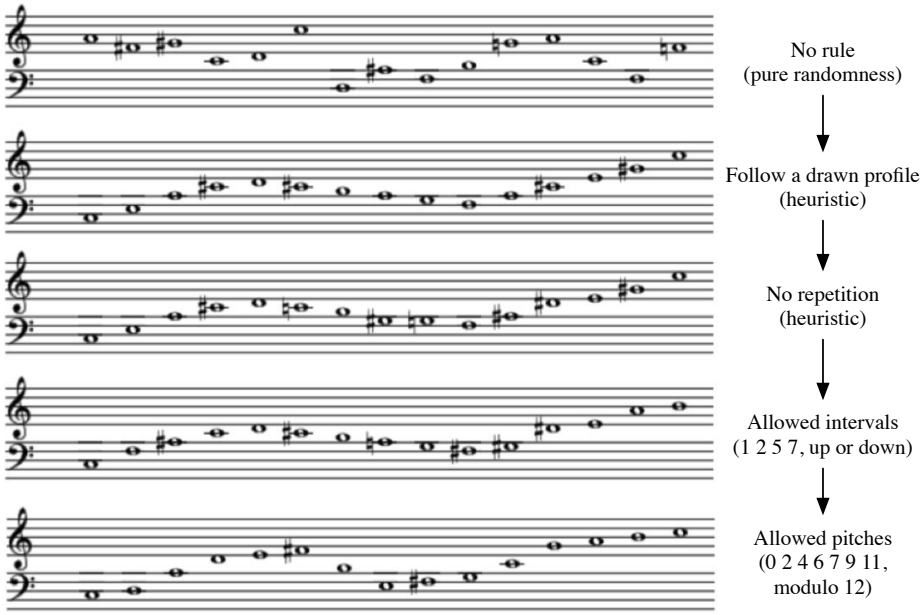


Simple polymetric process based on some initial, "music-agnostic" encoding
(PWGL patch programmed in 2011 with Dionysios Papanikolaou for his piece *Memoria*…)

If we move suddenly into the domain of constraint-based music generation — one of the most exciting aspects of PWGL I later ported to MOZ'Lib — writing a rule also requires developing the formal awareness to be able to describe a situation accurately. A constraint engine like PMC[24] or Cluster-

[24] The PMC engine (Pattern-Matching Constraints) or PWConstraints was created by Mikael Laurson as part of the original Patchwork environment. cf. M. Laurson. PatchWork: A Visual Programming Language and some Musical Applications. Studia musica no.6, doctoral dissertation, Sibelius Academy, Helsinki, 1996., chapter 5.1

Engine[25] works by generating random values based on a pool of candidates in order to build a music sequence or — in the case of polyphonic score with meter, multiple voices with rhythm, pitch, etc. — an overlay of such sequences. For each new value picked randomly, the system will check against the totality of rules defined by the composer, and will accept or refuse candidates accordingly. To be more precise, two kinds of rules exist in such systems. The first, most common category is called *true/false* or strict rules. A logical test is applied to each potential candidate, returning "true" (T) if the conditions described by the test are met — the property (of a single variable) or the relation (between multiple variables) is established — and "false" (NIL) otherwise. The second category, called *heuristic* or soft rules, does not reject candidates that fail their test. In fact they do not return a truth value, but a numerical value, an *intensity* so to speak, that gets higher if the property or relation we are looking for is stronger, and lower if it is weaker. By design more flexible than *true/false* rules and often compared to some sort of *wish*, heuristic rules will order all candidates based on that intensity, building sequences that amplify a situation, even if this amplification if minimal in front of stronger, contradictory rules. In other words, heuristic rules are applied *as much as possible* in the given logical context.



Simple example of generating a pitch profile with constraints with PMC engine.
(Search space of 15 notes chosen inside a chromatic scale from 48-C3 to 72-C5)

[25] The Cluster-Engine library was developed by Örjan Sandred for PWGL in 2010, after successive developments under various forms, *OMCS* for OpenMusic then *PWMC* for PWGL. Since 2020, it is officially featured in the core library of MOZ'Lib as its most advanced music constraint solver.
cf. « Constraint-Solving Systems in Music Creation » in *Handbook of Artificial Intelligence for Music*, Springer, Switzerland, 2021

My own approach with constraints has been similar to the work of a sculptor or ceramist. If some approaches to composition could be seen as essentially *additive*, I could define my own as mostly *substractive*. In the beginning at least, I need to find a block of material with the right properties, then only I can start cutting inside, molding it, looking for a shape. Then I collect as many of those shapes as I can, and I start looking for connections between them. I often start *without* a very clear image of what I am looking for, just setting up the minimal conditions for the music to exist — the *candidates*, how many instruments, their allowed range, allowed rhythmic values, melodic motifs, etc. — then I start generating and listening. At this point, results are extremely uninteresting musically, basically information noise without any kind of structure whatsoever. I keep interacting with the engine by adding rules, checking and listening to some intermediate results, adding new rules, solving unexpected contradictions between my demands, etc. When the type of situations I obtain becomes satisfying, I start amassing a large quantity of results with strongest identities based on my taste, my intentions for behaviors in the piece, poetic ideas, etc. I keep interacting like this until I collected enough (likely *too many*) materials.

The whole process aims to establish the necessary conditions to *believe* these isolated materials could, in fact, become music. In the same way, after I generated a sufficient amount of those instances, and I start editing them "by hand" in some kind of a sequencer, I need to keep believing but at a higher scale. I must invest emotionally into a meso/macro temporality which didn't exist in the materials, and try to solidify relations of analogy, causality and necessity out of those micro-meso materials, like reconstituting an imaginary jigsaw puzzle. In that sense, I feel this part of my practice using CAC has to do to some extent with the well-known idea of *pareidolia*, a perceptive phenomenon were a subject would recognize a pattern or shape within "nebulous" or random stimuli. Just like spotting a *figure* within a *texture*. The main property of a texture is that it can be segmented in multiple ways, where all possible segmentations can be considered equivalent. Therefore it seems that pareidolia occurs on the edges between "full texture" and areas were perception is made less equivocal by some local saliences or other (gestalt-like) morphogenetic properties. *Music pareidolia* could amount to hearing (or perceiving through other senses) some "musical forms" were they normally shouldn't be — in a recording of water dripping, the unusual gesture of an instrumentalist, a video-art piece, some game mechanics or a cloud of random notes generated by computer. In that sense, many (or *any?*) creation frameworks were the artist "goes fishing" for materials in some kind of chaotic structure could be seen as analog to pareidolia[26] : first looking for a block of "marble", then hoping that some form is hiding inside, and that the stars of

[26] I should also mention the more general concept of *apophenia*, a phenomenon analog to pareidolia, which would amount to drawing connections between *a priori* unrelated properties, perceptions, concepts, key-words, etc. in order to generate some amount of poetic complexity and depth. This is also a phenomenon I use continuously when setting up the poetic entourage of a piece, which eventually guides me towards the final form.

randomness and interpretation will somehow come to align, even for an instant. When I generate materials with CAC, I am usually facing shapeless, disappointing textures. But now and then, the initial conditions I established allow the advent of lucky accidents, where I end up believing I finally dug out some actual music from the mud. This state of belief — which can be deeply damaged by the environment — is what I am looking for most of the time, and need to protect, even if the gold I found is merely shiny litter.

Perhaps this reality of my practice clashes with the prejudice that using machines to make art should necessarily help make art faster and better, lead to some overproduction of music, thus trivializing both the artistic process and result. Technology seems to evolve in the direction of enabling us to do more, and more easily. But cumulatively, we can do more complex tasks with the same limited amount of physical and cognitive energy. If we end up doing less than we would without it, the technique is unadapted and needs to be redesigned — which often happens with new computer-based techniques and in particular prototypes made by artists themselves. We know that the "prophecies" of AI-based art, when deconstructed, reveal the impasse that machines can only (re)make existing art faster — and also in quantities that immensely exceed human capacities to consume art. On the other end of the spectrum, while I have always hoped (and somehow still do) that machines could give me access to creation, and would potentially make me "save time" while composing, I can only confirm that using technology my music-making remains slow and sporadic[27]. My decisions remain painful to make, and it is still difficult to evaluate on my own if those decisions are of better or worst quality. Working with CAC is not about

---

[27] Recently on two different occasions (one musical, one not), I suddenly switched between two technologies with the same function, allowing me to save considerable amounts of time. I realized in that moment that I might actually not care all that much about saving time. The time I save, using a faster algorithm or computer, I will keep losing to mine — generate or analyze — more data, go deeper, scrape the bottom. I think this is related to a form of *play*, toying with those machines without real purpose, but also with what Pierre Boulez called on some occasion the "rendement" of a technique, one could translate by its output capacity or performance.
If I spend a considerable amount of time creating an algorithm for a given piece, but end up using it very little, it may generate a certain frustration. An actual waste of time, but also a waste of a good program which I may never need to use again. This often leads me to make dubious choices, like producing more music than I actually need (here the "rendement" is apparently limitless), losing more time trying to use everything in the piece, etc., until I lose interest in its results and start deleting everything compulsively.
« I much prefer a precarious system with high performance ("rendement") to a perfect system with performance that is not guaranteed. » Interview given by Pierre Boulez in Ircam (2001), published in Jacopo Baboni Schilingi, *La signature. Douze propositions pour la création* de la musique au XXIe siècle, Éditions Mimesis International, 2024.

saving time, it is about restraining as much as possible the area of composition that CAC cannot deal with, and trying to understand better what Bernard Stiegler called the *incalculable*[28].

[28] This reminds me of anecdotes reported by certain composers about their paper score writing rituals. One of them would purposefully draw every single shape with a ruler (or other cumbersome devices) just in order to slow themselves down while composing. Losing time on purpose. It was never entirely clear to me, slowing themselves *from what*? From the speed of their own ideas, untrustworthy impulses of pure imagination? Or perhaps from the speed of deduction resulting from their own generative devices, i.e. the whole palette of analog *pre-CAC* tools they inherited from the XXth century? Or are these two actually the same thing?

Mixed music production : Cinema vs. Theater?

To conclude this first part, I would like to discuss what has become a dilemma in my practice of mixed music over the past few years, this time from the perspective of its actual mode of production. This may be due to some contradictions between my expectations — what I want to hear in my music, what I have actually been able to produce, and my personal difficulty in handling live music projects.

My pieces are typically structured by alternating sections which, beyond their specific musical properties, exist somewhere inside the continuum between fully written, semi-improvised and fully improvised. The latter are often rather short situations, either used as punctuation or to bring local contrast against some more written build-ups. In my most recent projects — few declinations of *Game Theory*, 藏镜人—*puppetmaster*, *Hyperlinks—asset(s) failed to load properly* — this continuum between written and improvised is further developed, or turned into a two-dimensional surface, by introducing the generative score. I am very interested in that tension between what is purely written (either by hand or generated / CAC-based, in any case very strict formally) and what is partly free or unpredictable. I consider this somehow analog to Schaeffer's idea of *allure* I mentioned before, i.e. the continuum between machine-like, living-like and nature-like.

Introducing improvisatory elements in my music, not as a single motor of expression but by smaller touches, was originally a way for me to explore this topic with musicians who, precisely, did not have a tendency to improvise, or even felt completely paralyzed by the word itself. One of my goals was to try and make musicians take improvisatory decisions, even without noticing, either by providing vague constraints (approximate durations, global directions, allowed harmony, pools of playing techniques, etc.) or through a more specific generative score. And when it is not yet obvious to them, to make them realize how artificial and arbitrary the rift between interpretation and improvisation is, often ingrained since the early years of their training. Paradoxically, improvised sections in my mind are more "relaxed" and less demanding in a way, at least physically, if not cognitively.

On the other hand, if I consider the more written portions of my music, those are clearly less ambiguous and allow fewer opportunities for deviation. I expect them to be more *machine-like*, in a way. They need a lot more time and commitment to come through, which is often incompatible with how new music production is usually organized, as a system.

One clearly visible aspect of these sections, among others, is that they generally aim towards the pole of *as fast as possible*. Just like for heuristic rules I already mentioned, this can be seen as a wish on my part, something that is always decided in consultation with the musicians. As fast as possible does not

mean "*the* fastest speed", it means a narrow range between the impossible and "slightly too slow" to be efficient, to be exciting for me as a listener. Obviously, it is also a matter of physical capabilities and endurance for the musicians, something that I always tried my best to calibrate with them, and with the help of constraints, to stay within the realm of the feasible. Yet in spite of my best intentions, the reality is more unforgiving… Several pieces I wrote after *Mémoire de l'eau* (2015) are extremely difficult or simply impossible to perform as is, with the level of expectation I develop as I get attached to the studio simulation. There, the range I just mentioned above seems too narrow for the music to exist in the physical space. And yet I keep trying to explore this area, in fact I must confess I really enjoy spending my composing time in those regions.

This has often led me to perceive my own music in a quite precarious and unflattering light, that it might be in fact *fake music*. My recordings tend to value artificiality without restraint, even on the instrumental level. When I share them with other composers, instrumentalists, or anyone else, what they hear is usually *what the piece could be,* not what it became in the live situation. There the conditions are rarely or never met, probably because my demands are too unrealistic, non pragmatic. I even wondered on occasions if they were in fact a form of self-sabotage… But even if that were true, it would only be partially so.

In the last 7 years I have produced only a few concert pieces, which are always more demanding for me logistically and emotionally. My production was already quite modest before I started this research, but I did consider myself happy if I managed to produce one piece every year. As a result, I have often been shamed for composing so little compared to others in the new music world. All these issues made me question my place in the area of music creation today, and in particular why live performance should always be considered the final criteria for judgement, or at least why it should be my personal horizon. Obviously the concert brings a much stronger aura and historical weight than the recording, a sense that it is "the natural way" of exhibiting music to an audience. It feels more direct and more "authentic", carrying a certain idea of "truth". And yet we know concerts are also a concentrate of artificiality, from the curtain rise to the applause.

This raises a question that has been on my mind since I began improvising and composing, and even more since I invested most of my attention towards mixed music : where should I situate myself on the axis between live performance and pure recording? For years, being stuck in the middle, I have chosen a hybrid solution, writing mixed music for concert, i.e. *mostly fixed* electronics on top of a live performance. I already mentioned the idea of the magic trick : creating the illusion of a connection with chance synchronicity, blind algorithms, etc., is for me an integral part of the work of mixed music. I am

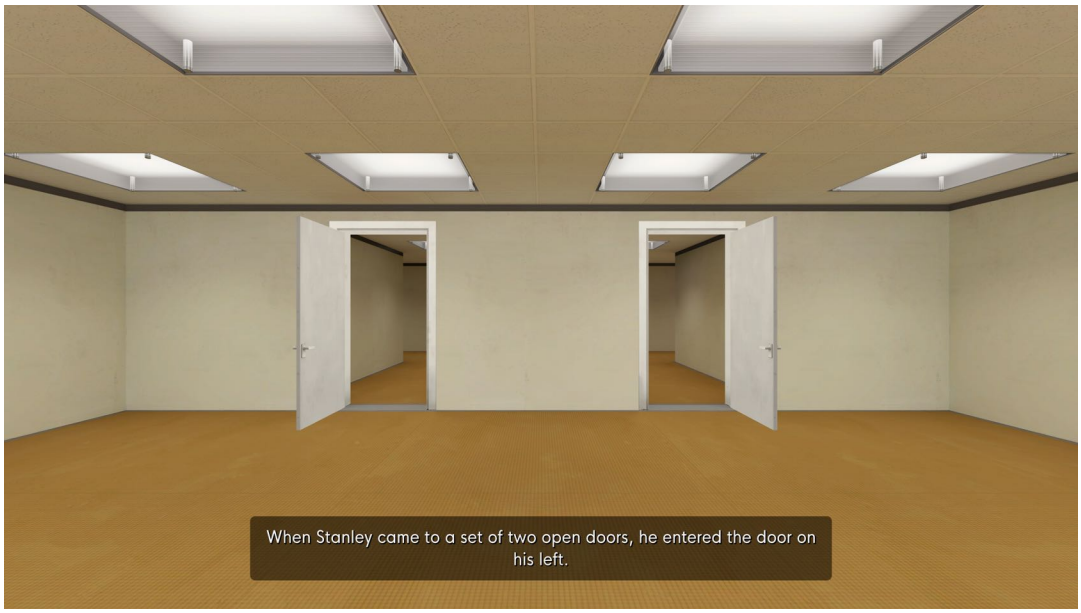
clearly comfortable with this idea of trickery, but as we have seen my problem lies elsewhere : my scores rarely pass the test of instrumental reality, in the concert venue. Our common value system revolves around the idea of presenting music that can actually be performed with pure skills, on the spot, by human individual(s)… and the occasional help of some *deus ex machina*. In other words, this is *theater*. This will perhaps allow me to scan my problem a little differently, and if readers will allow me the analogy : I am personally more drawn to *cinema*. But not necessarily purely abstract cinema — we often present acousmatic music as "cinema for the ear". By that I mean : cinema with *actors*, presence, skills or even (supernatural) virtuosity, able to do illogical things, de-multiply themselves, fly around the scene while distorting space and time. In other words, cinema *for movement* (kínēma) — that values high virtuosity and extravagant artificiality, along with one crucial element : precision.

My knowledge of the theories and discourses on interpretation of the classical repertoire is extremely small, and essentially boils down to two examples usually considered as the anomaly : Sergiu Celibidache's phenomenological approach leading to spectacularly slow performances, and Glenn Gould's decision to end his career as a live performer to focus exclusively on recording. Today I sincerely admire both, but the latter always left a stronger impression on me, first in my short time as a very approximate pianist, and later as a (home-)studio composer. It is tempting for me to transpose Gould's polemical posture against the current reality of performance, and try to follow and exaggerate his later approach with the studio recording. Creating — or faking — performances that surpass physical limitations, yet remain plausible, can give more occasions to develop certain categories of music expression which I feel are often overlooked in our field : figures (opposed to gestures and textures), rhythmic continuity and perception of speed (opposed to duration and slowness), etc.

But if I choose to follow this direction exclusively, a rather down-to-earth, *economic* problem remains : how can a composer exist from producing recordings alone, in the mid 2020s? In which artistic and economic network would that kind of music exist? In the present day, I don't think there is much possibilities left anymore :

— either I go left, I settle for *music as recording*. This seems quite a comfortable cocoon to stay in, almost radical in a way, closer to the core of what I believe music should be.

— or I can go right, keep following the *impossible* dream of unifying my interests in some kind of hybrid approach to live performance. It is most risky and very unlikely to succeed, also much more difficult to define. At the periphery of music itself…



Illusion of choice in the first minutes of the *The Stanley Parable*

Interlude I — A small catalog of (loosely related) techniques

Now I would like to list a few categories of techniques that I am interested in and use somewhat consistently in my music. My goal here is not to be comprehensive, but to examine from a higher ground the kinds of properties that I am looking for when using them.

• Morphological analysis

When I started to study composition, one of my goals was to try using CAC to compose music *from* video, and later from gestures recorded with various sensors. My posture was similar in a way with the idea of « composition with models »[29], a creative approach at the crossroads between composition and computer-aided analysis. But unlike composers close to the spectral school for instance, after the initial curiosity of my student years I didn't keep a strong interest in using models based on sound — recordings of instruments, sound objects or even pre-existing music — to structure my work[30].

My idea was to use the computer to highlight interesting properties in a given model, and use these as materials that would somehow influence the music's formal behavior, either locally or globally. Essentially, I wanted to use the computer to help me investigate any object, any kind of *data* in which I perceived some latent form of musicality, inner rhythm or interesting behavior that I would like to transpose into the musical domain.

I was never interested in direct *mapping*, that is taking the numerical values of the data extracted from the other media and applying them directly to music parameters, in other words *the value for x goes up, therefore the pitch goes up*. Instead I was always looking for ways to use the computer to perform deeper analysis on the object, try to identify its specific properties : temporal (or spatial) proportions ; intensity (quantitative or symbolic) and local/global directionality ; structure of contrast, "morpho-functional" elements (such as figure vs. background, etc)[31]. Then only I would try to apply these properties to musical materials, at any time scale I saw fit, with the help of CAC tools like constraints.

---

[29] This idea of composition with models was defended by the composer François-Bernard Mâche, student of Olivier Messiaen, whose practice left a strong impression on the younger composers of the spectral school like Gérard Grisey and Tristan Murail, cf. *Music, Myth and Nature, Or The Dolphins of Arion,* Taylor & Francis, 1992.

[30] I am aware that this idea can be considered a heresy by certain composers : « We are musicians, and our model is sound, not literature, sound, not mathematics, sound, not theater, visual arts, quantum physics, geology, astrology, or acupuncture ! », cf. Gérard Grisey « La musique : le devenir des sons » (1986), *Écrits ou l'invention de la musique spectrale,* Paris, MF, 2008

[31] Plenty of details on this research can be found in my Master thesis, cf. Julien Vincenot, *Analyse morphologique d'objets extra-musicaux (geste, vidéo,…) et morphogenèse en écriture instrumentale assistée par ordinateur*, Université Paris VIII, 2011

For several reasons, I left this part of my practice behind me for a long while, after I wrote *silent_data_corrupt* (2014) for alto saxophone and live computer — and thus began a significantly different period in my work. Later, as I already explained, most of my CAC tools slowly became unavailable, including the ones dedicated to analysis. It is only recently that I returned to it as a composition technique : when collaborating with Carmen Diez Salvatierra for *Objets de désir illicites*, we experimented together with a keyword-based analysis of each one of her 28 "cut-out shapes"[32], to determine the overall structure and evolution of the piece. Even if I mostly don't feel the need to follow this method anymore, it remains a conceptual framework that influences concretely the way I think about music today.

• Symbolic metamorphosis

The traditional approach to music interpolation takes place in the strictly numerical space : algorithms typically calculate intermediary states between two music entities A and B, by drawing n data points between the values describing those entities — pitches, durations, dynamics, or else. For instance a 5-step interpolation between the notes C and E returns : C, C#, D, D#, E, which amounts to a glissando in low resolution… By symbolic metamorphosis on the other hand, I mean an approach where the distance between entities A and B is expressed in terms of their similarity, as defined by the composer according to a variety of criteria. These criteria of resemblance can be based on the measure of perceptive properties or sound descriptors (the *harmonic centroid* or the *roughness* of a chord are well-known examples), but can also result from a more creative and arbitrary process of analysis on the music data.

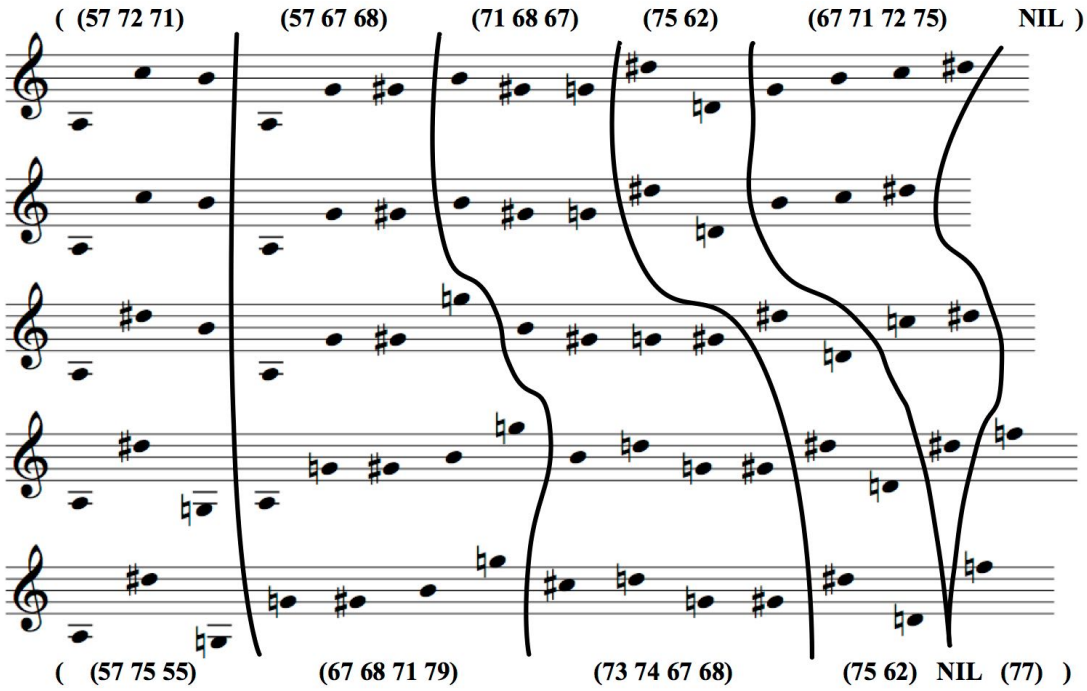
One simple criterion I find particularly interesting is the Levenshtein distance (or *edit distance*), which measures the difference between two entities by determining the smaller amount of elementary operations (such as adding, deleting and substituting) to transform A into B. For instance, to transform the word "horse" into "shore" requires to 1) add an "s" (horse → shorse) and 2) delete an "s" (shorse → shore). I have proposed a collection of interpolators in PWGL, now partly included in MOZ'Lib, which can perform similar transformations on any sequence of symbols, either melodic, rhythmic, harmonic, composite representations, etc. I used this technique to generate rather easy processes of metamorphosis between pure melodic profiles[33] in my pieces *Mémoire de l'eau* for accordion, and *Freeze, Flight or Fight*

---

[32] *Objets de désir illicites* (2022) is based on a collection of "short poetic devices" in French by Carmen Diez Salvatierra, *28 formes découpées*, Les Presses du Réel, 2019.

[33] Interpolation in the rhythmic domain with this method is promising, but becomes complicated when moving between motifs belonging to very different metric structures. To this day I haven't found a satisfactory solution to this problem, and usually stick to a hybrid approach between numeric and symbolic. I have started to develop another approach, based on *dynamic time warping* — an algorithm for measuring similarity between time sequences which may vary in speed locally — but I haven't used it in my music yet.

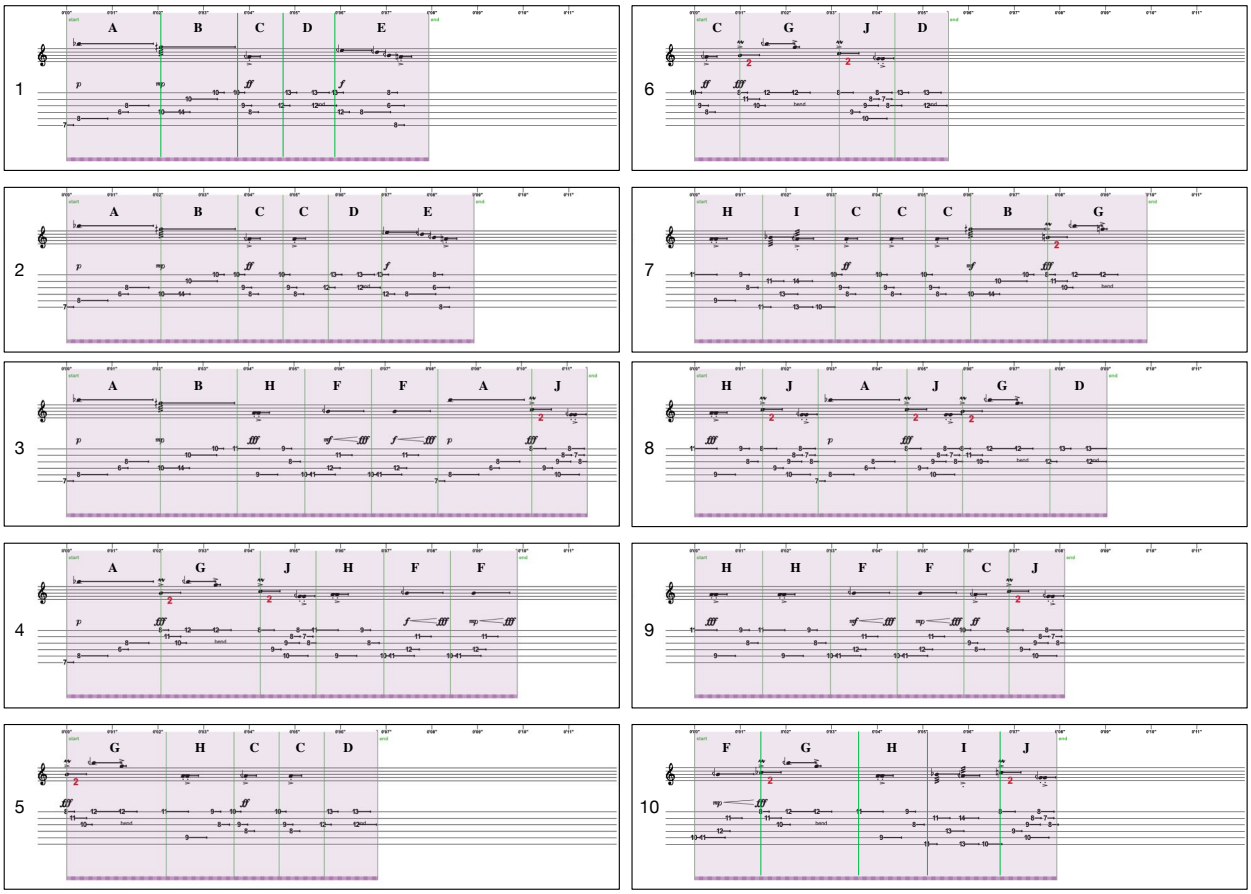
for saxophone and percussions. I believe they can produce more convincing results, closer to the idea one can have of a metamorphosis, because they do not invent foreign intermediate values between A and B. Instead, they rely exclusively on elements that already belong to at least one of them. Doing so, the intermediate motifs work by combining recognizable subsets of the originals, helping to maintain their identities throughout.



Simple example of symbolic interpolation between two melodic profiles

I have often experimented with another approach, similar to symbolic interpolation, which allows to navigate inside a pre-existing corpus of music entities, inspired by Frédéric Voisin's *FV-Morphologie* library for PWGL. By computing all the editing distances between elements of this corpus, one can produce an interesting category of graph called a *minimum spanning tree*. This mode of representation does not correspond to a traditional euclidian space (with X and Y axis), but rather to a topological reduction where all the connections between entities are always the shortest path to move between two given entities. That is, instead of generating artificially the steps that could exist between A and B, like an interpolation, one uses similar entities that already exist in the corpus as intermediary steps. I have used this method when working on the unfinished *Salt & Iron* (2022), shortly after for 藏镜人—*puppetmaster* (2023), and for *Hyperlinks — asset(s) failed to load properly* (2023-2024), in particular to help the decision-making process, by navigating in spaces of pre-existing melodic and harmonic materials.

For the latter piece, I also developed a variation of the symbolic interpolation idea, but this time applied to an arbitrary segmentation of fully-constituted music entities. This idea was inspired by a demonstration of Hans Tutschku's composition process for his piece *Irrgärten* (2010) for two pianos and live-electronics, which left a strong impression on me at the time. I started from a small collection of pre-existing score excerpts, initially generated using rather naive random processes, and then modified manually. I segmented these excerpts according to my own perception and understanding, placing markers along them. At this point, each sub-segment can be represented by a number or a word, thus creating a sort of vocabulary which can be invoked to create recombinations freely, according to arbitrary rules or formal processes. I have experimented with a blend of such techniques — permutations or random scrambling, sorting with some specific criteria (i.e. "alphabetical" order, ascending-descending by register, brightness, complexity, etc.), symbolic interpolation between two sequences, etc. — to create a couple of sections in *Hyperlinks—asset(s) failed to load properly*. My goal was to produce a sense of semantic drifting, swinging constantly between repetition and contrast.



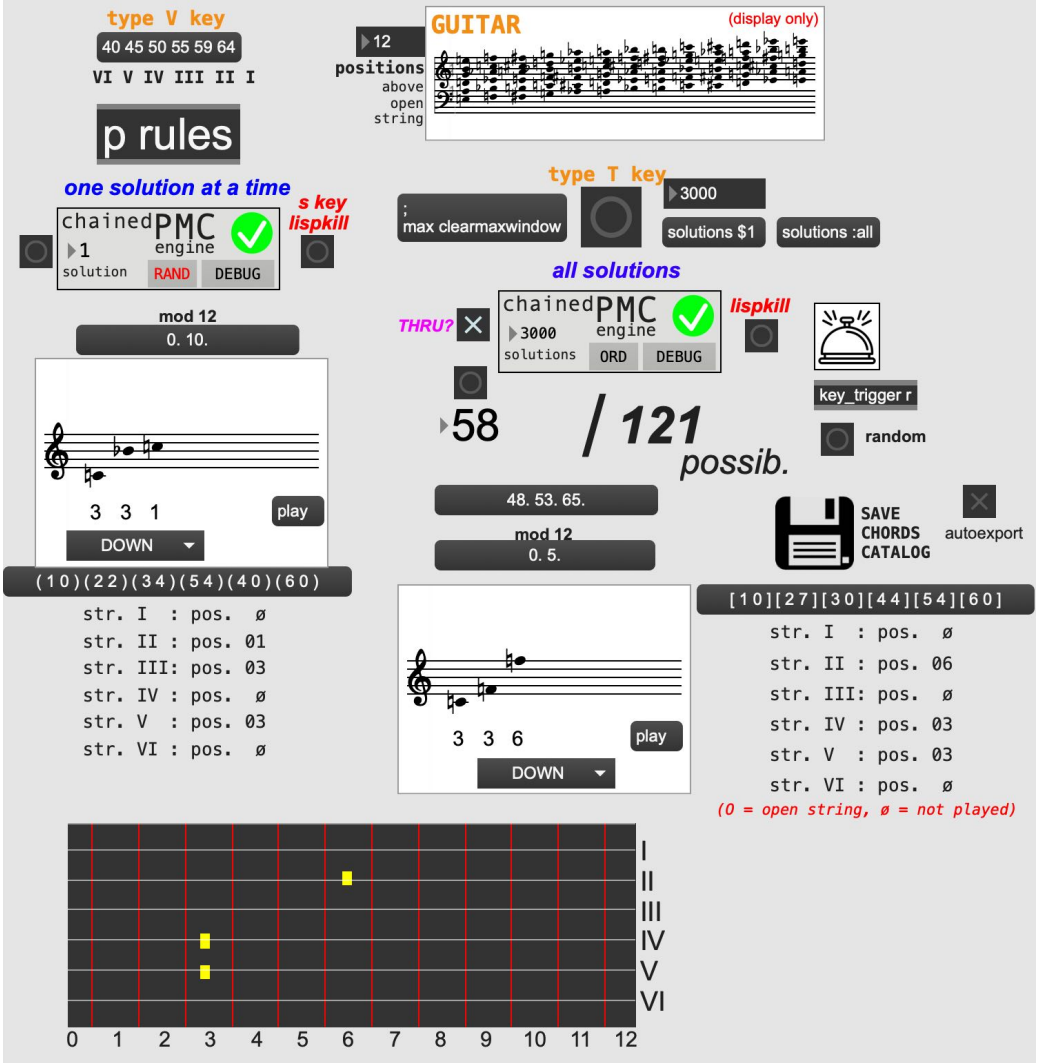
Example of symbolic process in *Hyperlinks — asset(s) failed to load properly*
("Placeholder A" for baroque flute and electric guitar)

• Instrumental constraints

One consistent use I made of constraints in my music was to figure out ways to help me generate scores that would be more easily playable by musicians. This is now a necessary condition for me, when working on a project in collaboration with musicians, to ensure that whatever is generated falls somehow within the realm of the feasible. I always need to spend some time with the musicians to discuss the most important limitations of their instrument, what is strictly impossible to perform for them, in particular the relation between certain gestures and more extreme rhythmic values. Then I try to turn those into a set of rules, and produce test results so the musicians can help me verify the system's plausible, until we agree on a common ground. This way, even if some difficult bits of music still 'slip through the cracks,' I can keep interacting freely with the constraint engine with peace of mind, and avoid wasting time getting attached to completely unusable materials.

My first experiment with this idea was writing rules to generate playable chords on various harmonies with the PMC engine, for the piano part of my early piece *MOrphism*, by controlling the distance between fingers. Similarly, this time in close collaboration with Yohann Juhel for *Mémoire de l'eau,* I determined certain conditions to build idiomatic positions and chords for the accordion, then further constrained those to a specific rhythmic context when used as candidates for the Cluster-Engine. More recently, when composing *Hyperlinks*, the flutist Irène Blanc helped me to specify rules to generate music for the baroque flute, in particular by excluding the most difficult "fork fingerings".

The most advanced set of tools I developed in this vein was specifically designed to generate idiomatic fingerboard positions when composing for string instruments. This project originally started in collaboration with Jean-Baptiste Barrière for violin and cello. I created a patch based on the PMC engine to build finger positions with reasonable stretch, in a way completely agnostic of tuning and resulting notes, focusing only on the topology of the fingerboard. Those rules were later combined with more musical rules concerning register, allowed pitches, micro-intervals, etc. This first patch was used to generate collections of plausible positions with four fingers, which could later be imported into a second patch, now based on the Cluster-Engine. The second step makes it possible to generate an actual music score favoring chains of positions with smaller jumps, compatible with certain allowed harmonic progressions and potentially interacting with other musical voices. An updated version of this system was developed for the guitar, in collaboration with Rafaël Carosi for *Hyperlinks*, and used in various situations of the piece where the instrument was present.



Guitar chord positions generator used for *Hyperlinks*, based on MOZ'Lib's constraint modules

• Post-processing on materials generated with constraints

When I started working with constraints more consistently to generate materials for my music — I first used the Cluster-Engine for *silent_data_corrupt* in 2014 — I did not really restraint the amount of music I could generate until I was satisfied and began editing. At this point, I was still discovering how Sandred's system worked, and what could be done with it musically. But as things developed, I often got overwhelmed by the sheer amount of interesting materials I could collect, and the quantity of manual work necessary to make sense out of them for a single music project would get out of control. Identifying what "worked" for me in those sequences, which ones to keep, and how to connect them, would sometimes take weeks to produce a single minute of music. Certain pieces in that time — for instance *Freeze, Flight or Fight* for saxophone and percussions, hardly digestible for me today — keep the mark of that indecision, of wanting to include "everything" into the final work.

When I started to write 藏镜人—*puppetmaster* then *Hyperlinks—asset(s) failed to load properly*, I adopted a different strategy. First, I needed to be much more specific about my expectations from the beginning, by spending considerably more time defining the search space and the rules. Then I would do my best to generate fewer sequences, and only keep a few of them as my starting point. Secondly, I would try to establish a more balanced "economy of materials" and explore each of them more in depth. For example, by pre-segmenting and then recombining them, allowing for more internal repetitions ; or by listening to them in loop from various perspectives, scanning freely with the loop boundaries to create unexpected patterns. For 藏镜人—*puppetmaster*, I also experimented with Markov chains in a rather quirky way : by listening repeatedly to random chains of a large collection of unrelated segments, I trained the system myself by adding weights only to the transitions that sounded the most natural to me. The results were promising and used for the live-generative part of the piece, but the experiment itself was terribly time-consuming, and truly felt like I was in fact the one being trained by the machine, not the other way around…

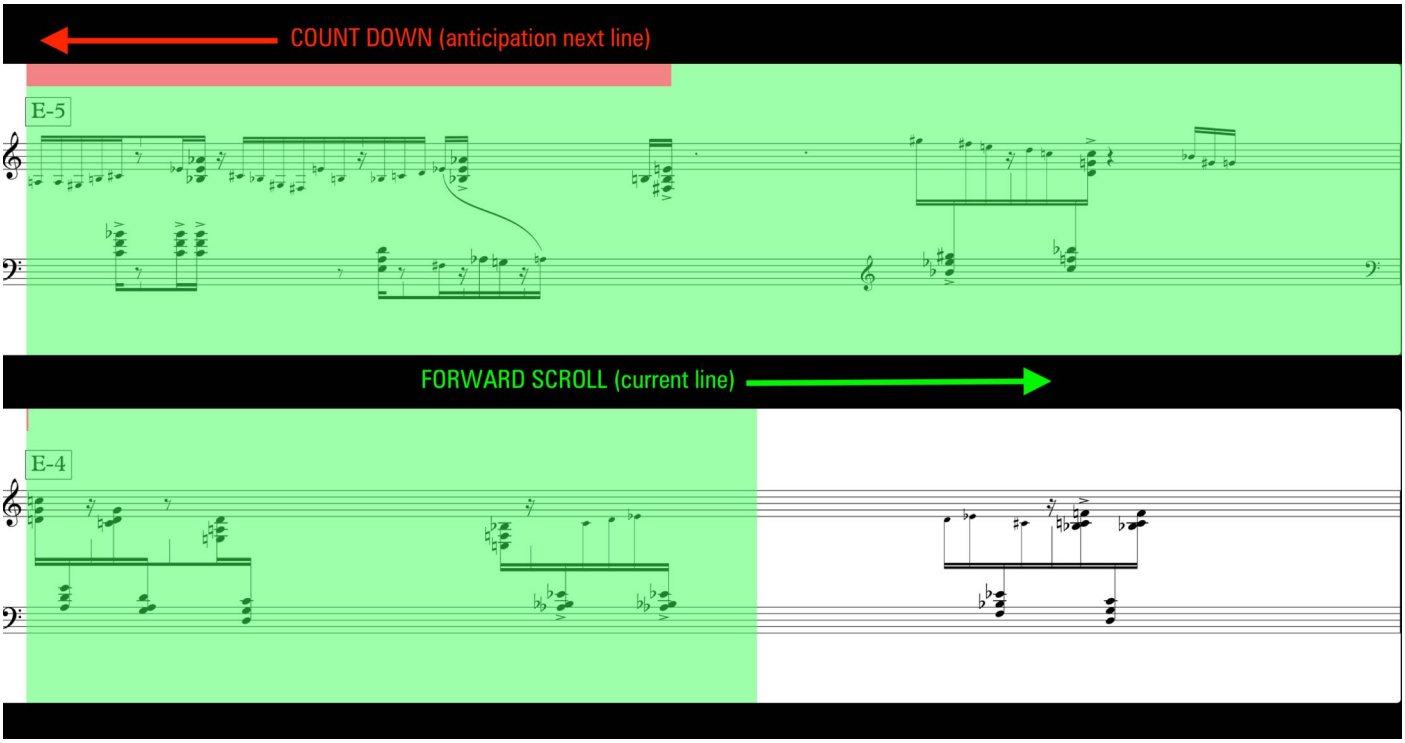
• Variations on the progress bar…

This last category is not a CAC technique *per se*, but clearly it has influenced my production of mixed music consistently since *Esprits-Animaux* (2012). The progress bar (or loading bar) is an essential element of visual grammar and computer interfaces[34]. As a younger composer I quickly became obsessed with it when I realized it was perfectly aligned with my conception of time in music, which was more chronometric in nature. While I am able to think with (and manipulate) traditional tempo-based / metric notations in slower situations, I find myself extremely clumsy when working with them in real time. Using a progress bar to convey to the musicians the exact time needed to execute a musical phrase, always felt more direct and reliable to me, closer to the way I composed acousmatic music in the DAW. I am perfectly aware this is not common practice, nor immediately effective for the musicians themselves, but in my experience it becomes quite natural to them after a little adjustment time.

I originally used Max to display the score for musicians with the progress bar logic for my piece *Esprits-Animaux* for violin and cello. The patch was borrowed from my teacher Lorenzo Bianchi's piece *Ouest profond* (2009) which had a great influence on me at the time. I kept expending and refining this idea for *silent_data_corrupt*, *Mémoire de l'eau* and *Fire&Forget,* all pieces for solo instrument and live

[34] While the digital progress bar we know today appeared around the late 1970s and mid. 1980s, I recently discovered its ancestor was most likely conceived in the year 1896 by engineer and economist Karol Adamiecki, and predates the invention of computers. Called the *harmonogram*, it was developed along Adamiecki's "law of harmony in management", most likely in a logic of control and surveillance in industry, agriculture and mining.

computer. In the latest form, the system displayed isolated segments of the score by alternating between two frames A and B, always pre-loading the next phrase in advance in the opposite frame. By alternating this way between two frames, the musician is able to anticipate constantly what comes next in the score, without worrying about page turns. The condition I usually give to musicians is that, while they are free to distort the phrase's time however they see fit[35], between the beginning and end of the frame, they should do their best to arrive on time for the end of the loading bar. I tend to use the beginning and end of each segment as moments of re-synchronization with the electronic part, which in turn participate in structuring the overall piece. To help musicians even further and enhancing synchronization, I often added a secondary, *reversed* progress bar positioned under the opposite frame, so that the musician could switch earlier to the next phrase and get ready to start playing it on time — like a count down.



Max patch for automatic score display in *Mémoire de l'eau* for accordion and live computer (2015)

[35] Even though I allow musicians to approach local time in this kind of notation rather freely, I had the tendency to suggest a somehow even and mechanical rendering of each phrases, following the more mechanical, MIDI-based auditory feedback I was composing with. This led to a rather unnatural way of engraving my scores. While traditionally the rhythmic spacing is not exactly equal, but follows some sort of "smoothing rule" developed empirically by engravers to improve readability, I always preferred to force a strictly proportional spacing were a whole note equals two quarter notes, one quarter note equals two eighth notes, etc. In the case of *Fire&Forget*, where the rhythmic notation is more involving, this led to some extreme situations were the readability of the score was far from optimal, making it much more difficult for readers to follow.

More than a practical choice and a personal preference, the loading bar refers to a specific kind of temporality[36]. In the domain of computer interfaces, it is a promise, a fiction, most likely a lie — but usually a lie that we can rely on. It has its own energy, its own behavior, its own *whims*… It is re-assuring, a confirmation that things are working properly in the background, even if they don't, until they don't… In a way it is quantified success, or failure. These ideas of the loading bar as a non-functional, non-reliable tool, came in the last years when I began to think of having screens in front of musicians not only as a pragmatic, *organologic* part of a live apparatus, but also as an integral part of the poetic proposition of a given work, to be seen by the musicians but also potentially by the audience.

I developed some prototypes in that vein, first in the early stages of *Game Theory - En Rafale*, composed during Claire Chase's workshop for flutists / composers in 2018, then the unfinished piece *all the memories are rigged* for ensemble (2019). There I imagined certain sections where the score display would start to become glitchy (almost stroboscopic), the progress bar would slow down or accelerate unpredictably, or even would start going backward, inviting the musicians to scan through materials, revisit them non-linearly. At a predetermined moment in the piece, the screens of all the musicians would suddenly dim automatically — leaving the whole performance space in the dark while the musicians continued playing from memory — then gradually return to normal brightness for the conclusion of the piece. Most of these ideas, I must confess, I didn't dare to use in the final version of the works. At the time I came up with these ideas, in the years 2018-2020, I probably didn't trust anyone (myself included) to actually have the time to make sense of them musically, and in ways that would not be a "caricature of new music". More recently in the early sketches of *Hyperlinks*, I briefly considered to keep developing this idea, where the score followed the erratic behavior of an imaginary internet browser with unstable bandwidth — unfortunately it became clear quite early that the piece would never happen in live, and therefore I abandoned the idea.

I was planning to re-use and develop some of these ideas when working on preparatory sketches for my bigger project *Game Theory* for ensemble and live computer. The main generative score patch for the performance, which is the only part of the project I mostly took to completion, was also inspired by internet and social media. It was presented not as a fixed frame with overlaid progress bar, but became the progress bar itself. The score was generated on the fly in a single bach.roll object, printing elements as they were being calculated, and progressively shifted to follow the intended playback speed. This time, the musicians could focus on a single area on the left side of the screen, near a fixed playback cursor of sorts, which allowed them to better anticipate the music arriving from the right side. In this context, the

---

[36] As shown by John Maeda, the design of the progress bar and its behavior can have a strong impact on the perception of time itself, cf. J. Maeda, *The Laws of Simplicity*, MIT Press, 2006

score became an infinite scroll of generated music, like a bottomless feed of overwhelming information capturing the musicians' attention until exhaustion. Additionally, calculating the score for *Game Theory* was extremely demanding even for my then rather powerful computer, to the point that the scrolling would oftentimes slow down and accumulate delay for a while, then suddenly accelerate to catch up, creating quite surprising and comical situations for the musicians of Listen ensemble who tested it with me in December 2019. At one point in the course of the piece, we even imagined that this main score would malfunction entirely, and suddenly be replaced by a chaotic display of animated GIFs, popular internet memes of various durations playing in loop, whose behavior and inner rhythm the musicians should try to imitate.

… and a digression on audio scores

Before I close this chapter, I would like to mention some exceptions to my usual practice, projects in which I relied not on the screen but on *sound* to conduct the progression of musicians during a given piece. This is a quite important detail to me, because one of the most common practices in mixed music, that is using a click track or in-ear metronome, is precisely the one that feels the most unnatural to me. I used it in only three occasions, for *Esprits-Animaux* (2012) in combination with the screen display, then for *Freeze, Fight or Flight* (2016, never performed) and *Hashiya: Matn* (2018). Choosing the click track felt like a logical solution for all of them, because of several sections that were more involved rhythmically. This is also most likely because they were not solo pieces, and I couldn't think of another solution to help multiple musicians to synchronize so precisely with the electronic part. Yet my experience was increasingly dissatisfying for several reasons, among which some extremely bad luck with wireless transmitters dropping off during the performance, or the click itself becoming audible to the audience accidentally in more quieter passages[37]. Most of all, I always found difficult to accept the idea that, for the sake of rhythmic accuracy, musicians should be forced to endure such a dry and non-musical layer to disrupt their immersion in the musical experience they were supposed to be a part of.

More recently I made a slightly different experiment while composing both 藏镜人— *puppetmaster* and *Hyperlinks*, two pieces for more than one musician that involved some form of a generative score, which I will describe more in detail in a later chapter. Both pieces were somewhat involved rhythmically, and eventually featured an unusual amount of repetitions compared with my previous production. This is why, in addition to the display of *bach*-based scores and the occasional progress bar, I took the gamble of *playing* the score itself to the musicians instead of a click-track, in the

[37] I admit the latter problem could be solved quite easily, yet both cases remained equally upsetting memories for me, especially in such contexts were I had to deal with plenty of pressuring matters.

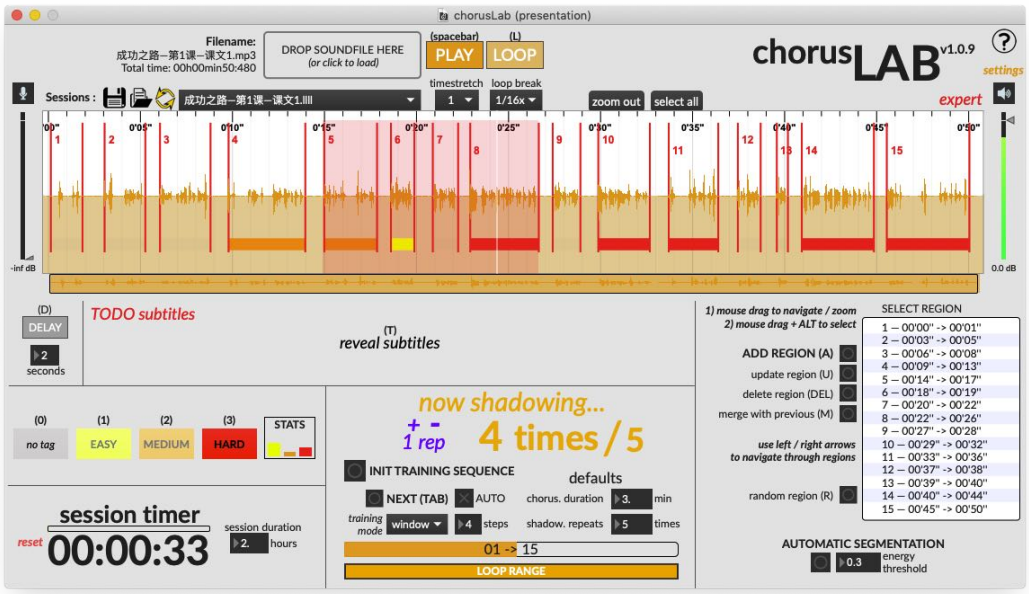
exact timing they would be expected to perform it, with the help of the same virtual instruments that I used to compose. I understand this idea can come as shocking to certain readers, due to the obviously limited and artificial quality of such simulation, but I believe this was an interesting solution nonetheless, in a way similar to the idea of the *audio score*.

In fact I have used the principle of the audio score only on one occasion, for certain parts of my string quartet *Hashiya: Matn*, commissioned by Ina-GRM. Unfortunately, the string players who were expected to perform the premiere in Radio France were not available to collaborate with me beforehand, and I was left with no other option than experimenting myself with borrowed string instruments. This felt like an exciting idea at first, and yet a very poor, makeshift solution, since I never studied any of those. I managed to produce an audio score based on a dense collection of recordings I made myself during a residency at the GRM studios in Paris, then assembled in the DAW. I gave the final recordings to the musicians, accompanied with an imperfect transcription of those different passages in standard notation, based on vague memories of the gestures I recorded months before. Clearly the method could have been improved, but I was hoping that the musicians could focus on the audio score instead of the paper score, and they were in fact able to reconstitute the image I had in mind, at least partially. I only realized later the biggest flaw in my idea : musicians, in particular professional ones under limited paid rehearsal time, would never be able to engage enough with such a detailed audio score, simply because listening time is much more demanding than reading time.

If I now return to the idea behind 藏镜人—*puppetmaster* and *Hyperlinks*, I did integrate the idea that the musicians would not be able to spend much time with the audio score, or even with the paper score for that matter. I started from the hypothesis that they would not have much time to practice the music at all before the concert, or that they would discover some of it during the concert — which is in fact the reality of many new music productions nowadays. Therefore, I would better incorporate the learning phase as a structuring element to the piece itself.

This was in fact inspired by a part of my daily routine of Chinese practice since 2021 : systematically *chorusing* with, and *shadowing* behind audio recordings of sentences of increasing complexity, i.e. repeating them out loud to improve pronunciation and internalize syntax. Prior to using this technique in music, I had developed some tools with MaxMSP exclusively for the purpose of language practice, which culminated with a prototype named *chorusLab*, in fact loosely based on *bach* and CAC techniques[38]. Without this personal experience of actually repeating sentences over and over for

[38] *chorusLab* allows to segment a large audio file freely or automatically, then practice each segment in turn, inspired by spaced-repetition algorithms (SRS) such as Anki or Glossika. It was developed as a pet project in the fall 2021, then beta tested by a small community of Chinese learners. To this day it was not officially released.



hours every week, to the point of forgetting meaning and focusing only on sound, rhythm and prosody, tones and consonants, I probably would not have dared to propose this experiment to actual musicians.

This was an interesting solution for me, because parts of the score were partially or entirely generated in real time; in some other parts, the order of appearance of score segments was not determined in advance, but picked inside a large pool of candidates by following a Markov process. This is why providing this additional layer of information to the musician, and giving more weight to their auditory memory and ability to imitate, felt like a reasonable solution to try. To some extent, I would consider it analog to coming to a choir rehearsal unprepared, and trying to catch up by listening to one's neighbors. In the case of 藏镜人—*puppetmaster* which was performed live on two occasions, the results were interesting but globally mixed, and I hope I can keep refining this idea in a different situation, with another piece. For *Hyperlinks* on the other hand, the idea of *chorusing* proved to be a valuable tool not for the live situation, but in the context of intensive studio recording. In particular, it helped the two musicians record large quantities of rhythmically demanding music with increasing levels of accuracy and expressive quality, yet with limited preparation and rehearsal time.

Placeholders and sprites

Here I would like to take a sidetrack to explore the idea of the placeholder, a sort of conceptual *ritournelle* that has been lingering at the back of my mind for the past few years. Most often encountered in linguistics, we can talk about placeholder words or names in a sentence for instance, i.e. generic words temporarily holding the fort for the actual *whatsitsname* we are currently missing, forgot or didn't decide yet. It is *something* that temporarily occupies a space, until something better, more appropriate or efficient, replaces it. It is not a flattering position. Placeholders are often elements that are *required* by the rules of the system in which they exist, like syntactic scaffolding, yet they may carry minimal or no semantic information at all. In mathematics and computer programming, this idea is even more pervasive : encapsulating a piece of code (as a function) or data (as a variable) can be seen as an explicit form of placeholder, indispensable for programmers to achieve a higher level of expressivity and complexity. We name a function or a variable, so that it can conveniently stay in place, in multiple places even, and thus propagate any modification to its content. All those examples seem to define a placeholder as something extremely *local*, an element in a sentence or in a code expression that can eventually be replaced. But just as an arbitrary symbol can hold huge amounts of data or even hidden layers of actionable code, when one zooms out in the chain of abstractions, placeholders can in fact be found at any scale.

I really hope readers can bear with me and still appreciate the irony of focusing on such an *empty* [ ] concept. If I see placeholders everywhere, are they even worth mentioning? Why are they so fascinating to me? I want to take the occasion of writing this text to try and unfold this, primarily for myself. I would like to determine if it is merely a tautological description, some "semantico-poetic" inspiration, or if it can truly become a practical, conceptual tool when elaborating CAC software and for my practice of music creation. I need to *put my finger on it*, develop my understanding of how they work, what they imply, so I can manage them more effectively later on. Because I feel placeholders, especially the ones I assign to a given function in a work, quickly become *walls* I need to tear down to be able to move forward.

As the son of a graphic designer, watching my father work on a new book or advertisement layout, one thing always piqued my curiosity : the *Lorem ipsum*, a nonsensical pseudo-Latin text used as a placeholder until the final text is determined, serving to incarnate a design proposal and perhaps help justify a choice of fonts. Yet it is not just some random gibberish : to the non-latinist (or the terrible student of Latin I was) it is *very convincing* gibberish. What I find fascinating is how the text, in that con-*text*, only exists to be forgotten, how it becomes *texture*, yet retains the appearance of the well-known, a

fictitious idea of the past. I now wonder how much this idea influenced my music without me noticing. Certain typologies of sections which return quite often in my music, seem to resonate with that description of the *Lorem ipsum* : most likely generated, maybe "a little bit old fashioned" (music of notes rather than sound for sure) ; dense and potentially scrolled through as fast as possible by musicians ; leaving only some shape or color behind them, details most likely overlooked, unless one revisits them repeatedly[39].

LOREM IPSUM DOLOR
SIT AMET CONSECTETUR

LOREM ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.
Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehen-

It seems that a placeholder is not always strictly local but can become a surface, or a blend of texture and segmentable figures, but a surface nonetheless, or once again… a wall. A placeholder wall. If I may jump suddenly to a very different domain, this reminds me of certain video games, where walls or background textures were accidentally forgotten by programmers and automatically replaced by a default texture, or a note to self to return later and… leave the job unfinished. A slightly different example is the invisible wall representing the boundaries of the virtual space that the player is allowed to explore. Often not very subtle, the avatar is simply forbidden to cross over, but in some extreme cases, the limit of the game itself becomes another space, this time infinite and empty, from which the player can escape only

[39] More broadly, the quality of the "scriptural" is a recurring inspiration in my work. In the last years, I have developed an interest for the concept of *scriptio continua*, a style of writing without spaces or punctuation marks. Common in Greco-Roman antiquity and still in use in several scripts around the world, it was progressively abandoned in the West most likely to minimize the ambiguity resulting from parsing certain texts. I started working with this idea in the unfinished *Salt&Iron* for three flutists (2022), where I imagined completely "flat" musical sequences which the musicians would need to interpret in depth to deduce the appropriate rhythm, pauses and tone, evoking my own journey learning to read Chinese writing.

by… turning off the game. Both cases signal the unavoidable limitations and necessities of the medium, often becoming an object of mockery from players, and the occasional “Easter egg” hidden there by developers, who are still often deprived of leaving a signature on their own work.

In terms of density, the moment when a placeholder is not merely a delimited object anymore, but becomes the “whole world” — an undifferentiated surface without clear contour, is probably my own personal limit, a property I want to avoid in my music. If the "whole world" here is a music piece, I am not sure if the word “placeholder” is even appropriate in that context anymore. I cannot help but see some connection with the online collaborative fiction called *The Backrooms*, which has emerged on the *4chan* platform in 2019. The premise came from an anonymous user’s post[40], accompanied by a single picture, and mentioning the possibility of “falling from reality” into a parallel dimension that is manifestly uniform, without contrast, and infinite. Large internet communities transformed this initial prompt into the basis for generating a maze of countless worlds — called *levels* — each with its own rules, often inspired by tropes of popular culture, horror, and video games. All of them share, in theory, the common rule of describing some kind of a liminal space or *non-place*, an anonymous zone of transition aiming to generate a feeling similar to the *uncanny valley*. I find the resulting mass of content available today around this topic fascinating, for its lack of identified author, its “oulipian” inventiveness and dimensions — supposedly, several hundred thousands of texts. For the new reader who discovers it, the total lack of direction to approach it is vertiginous, at the image of internet itself, or at least what the internet used to be. This is clearly one of the inspirations for wanting to revisit the idea of the hypertext in music, which I will discuss in the next chapter.

The original picture of *The Backrooms*

[40] « If you're not careful and you noclip out of reality in the wrong areas, you'll end up in the Backrooms, where it's nothing but the stink of old moist carpet, the madness of mono-yellow, the endless background noise of fluorescent lights at maximum hum-buzz, and approximately six hundred million square miles of randomly segmented empty rooms to be trapped in
God save you if you hear something wandering around nearby, because it sure as hell has heard you » Anonymous, 4chan (May 13, 2019)

When I first heard about *The Backrooms*, I could not avoid drawing a connection with the "green walls" I see pretty much everywhere I go since I arrived in China two years ago. Usually serving as enclosures around the numerous construction sites of the country, they often differ in material and texture, but the common denominator is always a green carpet-like, artificial grass quality. I found these walls in the most unexpected places — around shopping malls, at the docks, in front of old restaurants, in a public park, on a mountain site… — which led my wife and I to a recurring joke : « Well, this area has not been programmed yet… »

Even in the Cuihua mountains (翠华山), a placeholder wall…

This is I think a great representation of the idea of "failure" in art — or how failing in following our plans ends up usually being the ladder to actually end up somewhere completely different — and of the limit of systems, including digital ones. As we all know computers are terribly literal, one cannot get more from them than what was actually encoded, and one cannot escape from that enclosure[41].

I need to address the relationship between this concept of placeholder and a potentially artificial dualism between form and content, a theoretical problem which troubles me since I encountered it in my

[41] It was only very recently that I discovered the work of artist 徐冰 Xu Bing, through his spectacular installation 引力剧场 (*Gravitional Arena*, 2022) in Shanghai's Museum of Art Pudong. One of his oldest projects, 天书 (*A Book from the Sky*, 1988), also left a strong impression on me. Both works feature abstract symbols designed to resemble traditional Chinese characters, yet are completely meaningless to native speakers. Xu's work immediately resonated with the image of the *Lorem ipsum*, but also with its "placeholder wall" variant. Several online resources I consulted mentioned an early critique of Xu's *A Book from the Sky*, labeling his work as 鬼打墙 (guǐdǎqiáng). This famous idiom can be loosely translated as "ghosts building walls", a supernatural phenomenon where travelers repeatedly find themselves on the same path, regardless of which direction they took. As a metaphor, it could be interpreted as building invisible obstacles around oneself, or deliberate complexity for the sake of creating confusion.

research nearly 15 years ago. Specifically, I am concerned with the recurring critique of trying to predetermine local or global forms in a given music context, then merely "filling" them with musical content so to speak. This may have to do with some misconception about structuralism — and by extension all structuralist-inspired music approaches — as embracing a purely abstract, formalist approach disconnected from reality, ignoring the *flesh* of the objets it studies[42]. I disagree with such a simplification, and especially in regards to artistic creation. In my view both approaches, top-down (form to detail) and bottom-up (detail to form) are equally valid, and can generate uniquely interesting properties and expressive effects. A composer may naturally lean towards one approach or the other, and may even alternate between them in the course of the composition process.

As for the "real nature" of form, whether form predates matter, or the other way around, I feel very incompetent to develop a detailed argumentation. I can only rely on pre-existing theories of specialists of form I have been exposed to, ranging mostly from Gestalt psychology to the works of René Thom and Jean Petitot in a variety of disciplines — from mathematics, visual perception, phonetics, semiology, neurosciences. According to René Thom's catastrophe theory, which inherits some elements of *Gestalttheorie* : « the inherent characteristic of all form, of all morphology, is to express itself through a discontinuity of the medium's properties », but also : « matters aspires to form »[43]. Regarding perception, Petitot explains : « Many specialists of perception (for example Stephen Grossberg) have explained that the perception of scenes with objects involves complex neural algorithms for segmentation, through the construction of discontinuous contours, that separate objects from the background and define homogeneous regions ("merging") filled with sensory qualities ("filling-in"). »[44] In the more global sense, Petitot also remarks that : « The emergence of morphologies within matter is a *hard problem*, analog to the explanatory gap between neurosciences and consciousness. »

Beyond that impasse, my interest for the concept of the placeholder perhaps reveals that when I compose, I do not pay so much attention to "content", especially in the beginning of a project, but I am much more attracted to manipulating forms, or representations of forms. This kind of posture has been

---

[42] In tracing the genealogy of what he calls a *philosophy of form*, drawing from Goethe's botany, the works of D'Arcy Thompson (*On Growth and Form*), Alan Turing (*The Chemical Basis of Morphogenesis*), and René Thom (*Structural Stability and Morphogenesis*), Jean Petitot describes the organicist influences of Claude Lévi-Strauss's structural thought : « On the theoretical level, identifying this genealogy is of critical importance because it attests that structuralism is a biologically inspired *naturalism* and not, as is too often believed, a logicist inspired *formalism*. Structures are conceived there as dynamic forms in development, as morphodynamically (self-)organized and (self-)regulated totalities. » cf. Jean Petitot, *Morphologie et esthétique*, Maisonneuve et Larose, 2003

[43] cf. René Thom, *Structural Stability and Morphogenesis*, InterÉditions, Paris, 1972

[44] cf. Jean Petitot, « Trois remarques sur les modèles morphologiques de René Thom », Public intervention in *1923-2023 - Centenaire de René Thom*, Institut des Hautes Études Scientifiques (IHES), 2023. https://www.youtube.com/watch?v=S3wHYEPt8CA (accessed June 2024)

criticized in the past, and I think rightly so, by someone like Tristan Murail in his famous article *Spectra and sprites* :

> *There is a conceptual error from the very beginning : a composer does not work with 12 notes, x rhythmic figures, x dynamic markings, all infinitely permutable ; he works with sound and time.*
> *The sound phenomenon has been confounded with its representations, and we work with these, with symbols. Since these symbols are limited in number, we quickly come up against the wall…*[45]

Interestingly, we reach yet another *wall…* Today I would argue the conceptual error was twofold. First, the idea that composers only had symbols representing notes, rhythmic figures and dynamic markings to work with. Perhaps this was still the case in the 1980s, but such limitations have been dismantled by the explosion of computing capabilities, and the complete redefinition of modes of interaction and representation over the last decades. So much that today, we can generate complex texts, images, and sounds with rudimentary prompts expressed in natural language.

Secondly, I believe « sound and time » is a rather limited portion of what music is about. In fact, I am tempted to think that music is not (really) about sound[46]. That sound is yet another placeholder — a powerful instance which happens to perfectly align (most of the time) with what my attention is looking for when I compose or listen to music. But a placeholder nonetheless, for *something* that is not essentially musical, or sound-based, but can cross multiple areas of expression (*inter-semiotic* or *trans-artistic* in a way), similar to the definition of *rhythm* given by Molino and Nattiez[47].

Placeholders are symbols which all mean « come back and teach me more ». They hide entities for us to care for, that can grow and evolve… until we stop paying attention. If I take the context of software creation dedicated to art making in general, working with placeholders means navigating a chain of decisions and design choices, while managing the technical complexity and inherent time required to

---

[45] cf. Tristan Murail, « Spectra and sprites », 1984

[46] A similar position is defended in the following article with some interesting arguments. cf. Eric Chasalow « Electroacoustic Music is Not About Sound », *NewMusicBox* — NewMusicUSA.org, 2016

[47] « […] rhythm is not exclusively the property of music, and it is not enough, when speaking of musical rhythm, to merely note that it can be found elsewhere. We must not adhere to a conception of musical rhythm that is separate from its anthropological foundations and must recognize that rhythm, as such, possesses an autonomy that we readily attribute to music: rhythm is not an abstract organization of time, the surface configurations of which are sufficient to study, but an ensemble of behaviors involving the body as well as society. This is why we prefer to speak, on the most general level, not of rhythm but of rhythmic experience. […]
This experience is omnipresent, from physiological rhythms to movements and gestures, from sexual rhythms to technical rhythms, and even to the collective rhythms of work, play, sport, combat, ritual, and festivity: it constitutes an ensemble as complex and diverse as music. The relationships between rhythm and music should therefore not be viewed as dependent relationships between music and one of its parameters, but rather as relationships between two 'families' united by multiple intertwined networks. » cf. J. Molino et J.-J. Nattiez, « Typologies et universaux », pp. 362-363, in *Musiques - Une encyclopédie pour le XXème siècle*, vol. V, Actes Sud / Cité de la Musique, (Paris, 2007)

implement each brick locally. This is even more the case when using CAC, which requires to constantly juggle between composition level and programming level : considering the time required to make such a brick functional, one often has no choice but to leave it in a suboptimal state, to be able to keep the thread of compositional ideas intact. Simultaneously, we raise fragile, not fully satisfying elements to support the whole structure, all while maintaining sight of the ever-changing horizon we are aiming to reach. In other words, placeholders are to programming and (I believe also) to art making what scaffoldings are to construction : a necessary evil, supposed to prevent the work from collapsing (yet).

Closer to composition, I often see a placeholder as some local "decision-making" device whose complexity has been voluntarily / temporarily overlooked in order to keep working on the bigger picture. For instance, some parts of a decisive sketch or CAC patch can be left with some unsatisfying parameters such as : « use those two scales for now, just to check the engine works », « pick *n* elements randomly in that short list, I will replace this with something more interesting tomorrow », or even « I really don't know what to think about this element yet, I will need a few more months (or years) thinking about this, let's do it more easily for now ». These are often not the most intelligent or expressive ones, by a long shot, but they do allow me to get the interaction loop up and running. I need to *forget* some of those local choices on purpose, voluntarily along the way, otherwise I cannot make progress. Obtaining some initial musical output, then maintain my attention[48] and interest for the process are the priority.

But this can lead to a scenario where I would play with (and feel) a system for an extended period of time, and entirely forget about some placeholders I left somewhere inside the tree of the program. While interacting with it, I would get used more and more to the result as a whole, because every new brick added to it would get me closer to my goal. Or I might have made a surprising discovery along the way, and tried to obtain more similar results with the same initial conditions. For instance, some passages of my scores would display a silly consonant harmony — sections in the middle of 藏镜人— *puppetmaster* may be the strongest example — because I was not really paying attention to harmony when programming it. Perhaps I was focusing on rhythmic behavior instead, or other musical properties — or worst, non-musical parts of the program, making sure the patch would never crash during the concert. Eventually, I reach a point where the overall result — materials with a strong identity, aligning

---

[48] Not every artist is born with equal focus, determination and discipline, yet in the attention economy, we are all collectively targeted by the competing "weapons of mass distraction" described by Yves Citton. My enthusiasm for music creation truly blossomed during the last years of the "web 1.0", where it seemed paradoxically much easier to enjoy than today. But I only existed "as a composer" with mass social media, streaming platforms and some form of algorithmic surveillance in the background. These challenges are likely even more pronounced for the younger generations, who have been immersed from the outset in this intense regimen of information and attention capture. This is why a discussion on the ecology of attention is crucial for both artistic creation and pedagogy today. We also need to critically evaluate and perhaps (re)design our tools and techniques accordingly, alternatively, to reclaim some of our capacity to act. cf. Yves Citton, *The ecology of attention*, Polity, 2017.

with my original vision — is what matters.

I told on occasions to some colleagues that my music was almost entirely generated and I absolutely *could not* write it by myself. And yet some of them would not believe me : « I am sure you could still write it by hand » is basically how the conversation ended. Yet I know for certain that I could not. Quite amusingly enough, a couple section I called “Placeholders" in my last piece *Hyperlinks*, are in fact based on a form of “MIDI painting” : some initial random generation, then a long process of manual editing, copy/pasting, transforming all while listening in loop. In other words, I wrote them mostly by hand, although the final result’s structure was partially determined mechanically by some symbolic interpolation device. A few of my pieces present similar moments — the middle section of *silent_data_corrupt*, a short section right after the introduction of *Fire&Forget*, the whole opening section of 藏镜人—*puppetmaster* — which are as much “handwritten” as can be. And when I listen to them again years later, they always feel a little clumsy to me, shameful even, out of place. Just like if I left them there by accident, and I *should have* replaced them with something better if I had the time, a better idea, or the right patch… I believe now they are necessary, they could not *not be there*. They are the proof of what I can really do, without CAC. They are my witnesses, an incarnation of my true self to justify the rest which is (perhaps) not *really* me : the amplified, magnified, computer-enhanced (*cyborg*), what I wish I could be. A point of reference to keep myself on the ground.

If placeholders were likely present in my music from the beginning — some readers might relate to this in their own work — they become much more challenging in larger projects like *Game Theory*. Placeholders are at the same time a guarantee of success — succeeding in building something — and a guarantee of failure. A wooden wall is a placeholder for a concrete wall, a concrete wall is a placeholder for a window or a door, a green wall is a placeholder for future, *un-seeable* parts of a city… The more time and effort I invest in deploying a placeholder for any significant purpose, the harder it becomes to replace it. Until the point, perhaps, where I decide the work is *done*…

Interlude II — MOZ'Lib

MOZ'Lib is an external package for MaxMSP which I started developing in 2015. It can be considered a non-official satellite of the *bach* ecosystem, created by composers Andrea Agostini and Daniele Ghisi in 2010. First a personal hobby project, MOZ'Lib has been completely refactored in its current form in the summer 2019, and fully documented in English. The package is distributed as open source on Github since January 2020[49], and regularly updated. Originally a tool exclusive to macOS, it has been ported to Windows with the help of Matteo Marson and Juan Sebastián Vassallo in 2021.

MOZ'Lib can be divided into two distinct halves, with significantly different purposes : the original MOZ'Lib, a set of pedagogical tools designed to explore musical writing, creation and computer programming ; and *PWforMax*, an expert system allowing to generate and run Lisp code in Max, and thus to work in real time with any library from the PatchWork (PW) / OpenMusic / PWGL family.

The impulse for the creation of the pedagogical side of the project was a commission from the project *Ariane#*, directed by the DRAC (Regional Directorate of Cultural Affairs) of Franche-Comté region in the east of France, and piloted by the Conservatory of Montbéliard. It resulted from lengthy discussions with composer and pedagogist Gaja Maffezzoli, who specialized in pedagogical approaches of electroacoustic music intended for younger children, first in Florence's Tempo Reale center, then in Montbéliard. Its initial goal was to introduce a young musicians in training to music theory and instrumental composition with the help of digital tools. The library had to be accessible, and above all, reactive and interactive. This is why, while we originally considered using PWGL to develop these tools, we eventually opted for *bach*. Another choice made early on was to avoid making closed tools, but instead produce a more flexible environment, combining ideas like a chain of *plug-ins* in a DAW. The solution was inspired by the BEAP and Vizzie modules introduced in 2014 by Max 7 : a library of bpatchers, i.e. large boxes with embedded user interfaces, that could be connected to each other like regular Max objects to build more complex systems with limited programming knowledge. My own ambition was to create the easy CAC environment I wish I had as a younger composer.

The main library is a collection of more than 30 modules to generate, transform, display and listen to written music materials. Modules can be simply connected with patch cords and transmit lists of pitches (or midicents, where 6000 is the central C), rhythm (represented as rational numbers), purely symbolic data or other music parameters. Unlike PatchWork and its successors, where the programming logic was so-called bottom-up or *demand-driven*, MOZ'Lib inherits the same approach as Max and *bach*

[49] cf. https://github.com/JulienVincenot/MOZLib/

where data follows the tree of the patch from top to bottom — i.e. *data-driven*. This logic is in fact quite unnatural to users familiar of classical CAC environments. This resulted in some necessary design choices like the addition of a "callback" button on each module, forcing data to traverse the patch again and update the whole tree, for instance when a new module is introduced in the chain.

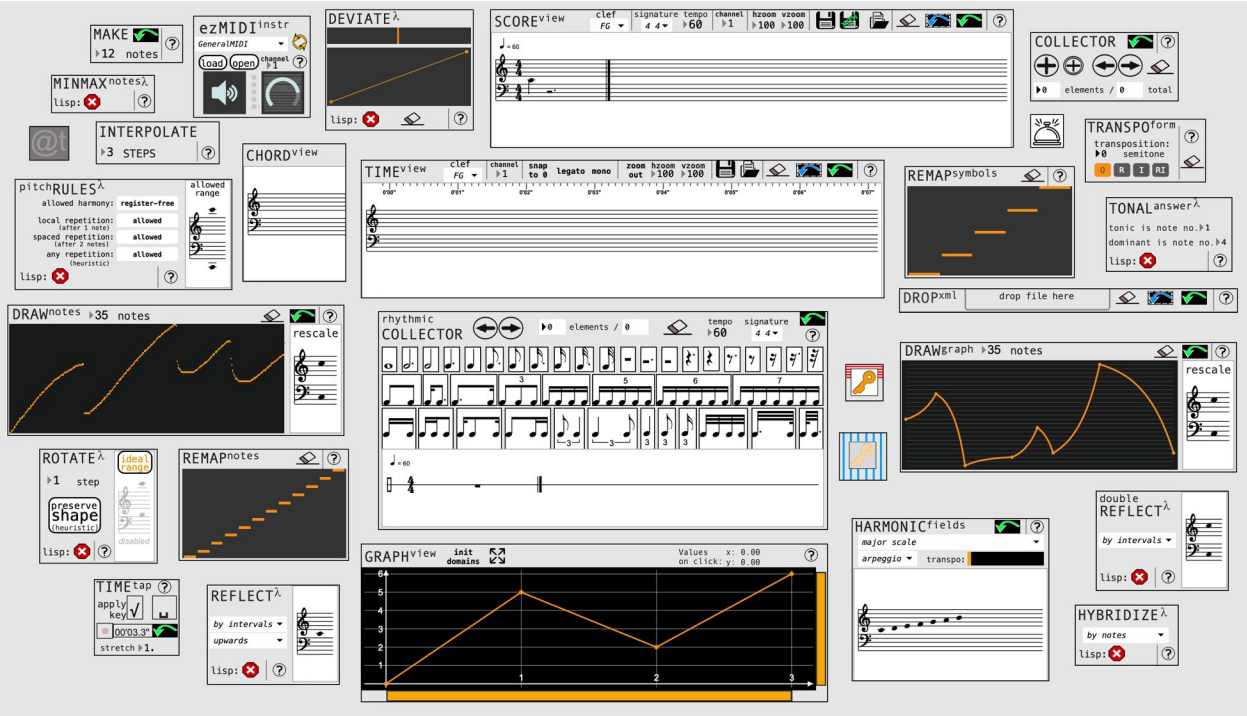
In the side bar of a Max patcher window, the snippets menu allows to easily access the palette of modules arranged in different categories :

1) *MOZ Generators* — modules to create simple musical entities from scratch by drawing, recording rhythm (with key tap), or building up a new sequence by selecting from a palette of pre-determined options (rhythmic cells, harmonic fields, etc.). These modules provide some input for all the other modules, therefore they are usually the first ones to add in a patch.

2) *MOZ Editors* — a small set of modules to display musical entities according to different modes of representation : flat melodic profile or list of chords (inspired by PWGL's chord editor), sequencer-like chronometric score (bach.roll), traditional mensural score (bach.score) or more abstract representations (break-point function or graph).

3) *MOZ Transformers* — a selection of classical CAC techniques such as transposition, retrograde/ inversion, rotation, reflection, distortion, interpolation, non-linear remapping, basic constraints…

4) *MOZ Utilities* — a collection of smaller tools to facilitate programming and building of interfaces, MusicXML import or playback of scores.



As an advanced user of PWGL, I did not anticipate to use any of those tools in my day-to-day work. But as the MOZ'Lib package kept growing and improving, I found myself using them more and more in a variety of situations, to accelerate my workflow when testing ideas, to display scores more easily, etc. Some modules were in fact inspired by my own workflow with PWGL, re-actualized for *bach*.

The *MOZ Transformers* collection of techniques was the most tricky part of the project. One of my goals in particular was to emulate certain functions from the *JBS-CMI* and *JBS-Profile* libraries by Jacopo Baboni Schilingi, which were my entry points in PWGL almost 10 years before. Many of them were in fact quite complex Lisp programs, and I quickly realized that it was impossible to recreate them only with Max or *bach* objects. This was the initial seed to develop a system allowing to transplant techniques of the PatchWork family into Max.

The initial prototype of *PWforMax* was the result of a collective effort during the 2015 summer meeting of the international PRISMA research group, held in Ircam, in particular with Örjan Sandred, Hans Tutschku and Johannes Kretz[50]. In this system, the Lisp code is initially produced in Max, then evaluated outside of Max — via terminal commands to the *shell* object — by SBCL, an independent Lisp implementation. After the calculation is completed, the result is returned back inside Max with imperceptible latency, and can then be further developed or displayed with regular *bach* interfaces[51].

The system is in fact quite simple but was critical to be able to implement the small catalog of techniques needed for my pedagogical modules, with a minimal amount of work. But it soon became a much bigger endeavor, that is preserving decades of CAC research and compositional knowledge which was about to disappear. When I discovered PWGL, it was not a "naked" tool : a few dozen of *user libraries* developed by fellow composers and researchers were passed from hand to hand. I learned how to compose by experimenting with those tools, discussing with their authors, studying their code, etc.

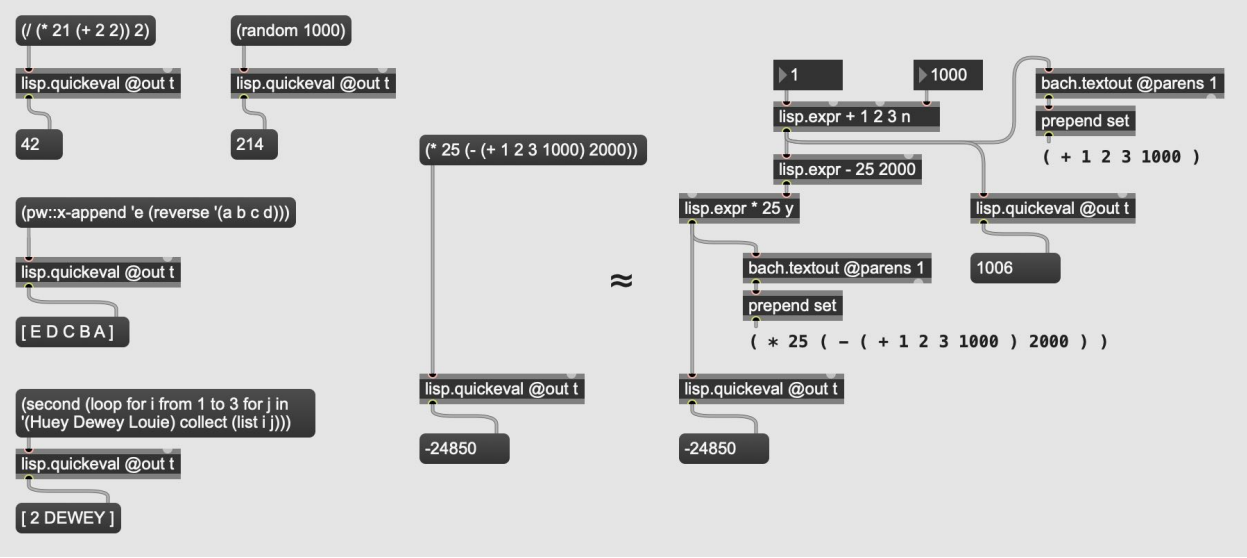
---

[50] The dedicated workshop in PRISMA was triggered after a hypothesis I made during my time in the Ircam cursus, the previous year, in discussion with Jean Bresson who was then the main developer of OpenMusic. Both PWGL and OpenMusic were programmed using a proprietary implementation of Common Lisp called *LispWorks*. I realized both environments came with strict limitations on *heap size* in recursive functions. This prevented users from performing heavier calculations, for instance when analyzing a large corpus of data, or when generating longer results with a constraint engine. This limitation did not exist in several open source Lisp implementations such as SBCL, so I proposed the (rather absurd) solution of performing more demanding calculations outside of PWGL or OM, with SBCL working in the background when necessary. The idea was prototyped directly in Max by PRISMA members, and immediately adopted for MOZ'Lib.

[51] In the first prototypes, traditional Max messages were used to generate the Lisp code on the fly, with Max's *sprintf* object, but quickly the well known size limit of such messages brought a number of issues. They were practically unusable in the long term. It was thanks to a critical insight by Jean-Baptiste Barrière that I eventually started to use *bach* objects themselves to generate Lisp code without limitation, thus making *PWforMax* fully stable and operational. For more technical details about the implementation of the system, cf. Vincenot, « LISP in Max: Exploratory Computer-Aided Composition in Real-Time », in International Computer Music Conference (ICMC) proceedings, Shanghai, 2017.

They became an integral part of my working environment. Over the years I have ported one-by-one a certain amount of libraries from PWGL to MOZ'Lib, often in collaboration with other composers. This was made possible only by collecting a large amount of historical functions on which all those libraries relied. Most of them, common to PatchWork and PWGL, were in fact borrowed from the source code of OpenMusic, and thus became the main kernel of *PWforMax*.

First hidden in the background of certain MOZ'Lib modules, *PWforMax* is now fully documented and accessible to users who need to work with Lisp in parallel of standard *bach* objects. In particular, it comes with dedicated tools to facilitate the manipulation of Lisp code directly inside of Max. The few objects I call *Lisp scripting boxes* allow users to copy PWGL or OpenMusic patches with minimal efforts directly into Max. This allows not only to preserve their previous work, programs and sketches, in a more enduring environment like Max, but also to transform radically the way they interact with those tools by bringing them into the real-time world.



Some basic examples of Lisp evaluations in *PWforMax*, including a same s-expression given both as 1) a pure Max message and 2) a dynamic patch with Lisp scripting boxes

A large amount of PWGL and OpenMusic functions are available directly in this system, but most of them are poorly documented for lack of time or necessity. For this reason, one of the most common critique *PWforMax* received in its early years, was that it felt like a "black box", making it quite difficult to learn even for advanced users. This is why that I made several efforts in the recent years to better communicate what was available to use, in particular a dedicated module to explore the kernel by keywords, called *LispLore*.

Certain libraries have received a special treatment and are now showcased as *PWforMax Guest Libraries*. They required not only to adapt their Lisp code to work in the new environment, but also to design custom interfaces more adapted for the context of Max, and a full documentation with plenty of example patches. I see this selection almost like a work of curation, highlighting different approaches to music thinking through CAC that I find especially compelling, and in particular some libraries created by friends from the PRISMA group. I hope to keep bringing new libraries to this collection as often as possible, when time allows and depending on users' requests.

Here is a panorama of some of the most important libraries available in MOZ'Lib :

• Jean-Baptiste Barrière's *Chréode* and Kaija Saariaho's *Transkaija*

This section brings together the reconstitution of two historical libraries, both developed in the 1980s in *CHANT/FORMES* environments[52]. Realized in close collaboration with Barrière shortly after I presented him with early prototypes of MOZ'Lib in 2016, this project was the first serious exploration of the system to facilitate the porting of Lisp tools into Max. The original code for both libraries was written in *Le Lisp*[53], a now seldom-used dialect of Lisp. Therefore it was not possible for us to experiment directly with the original source code during the reconstruction process, and we had to make a lot of hypotheses from the original code, some intermediary versions for OpenMusic and PWGL, and of course Barrière's own memories.

Chréode was conceived with the ambition to realize « a grammar of formal processes, a morphogenesis »[54], that could determine the whole behavior of complex sound synthesis as well as instrumental scores. For this purpose, the system works with a single unifying paradigm : the accumulation, interference and downsampling of damped sine waves, inspired by formant wave synthesis. In a logic similar to MOZ'Lib's pedagogical modules, *Chréode* now consists of a palette of bpatchers, but

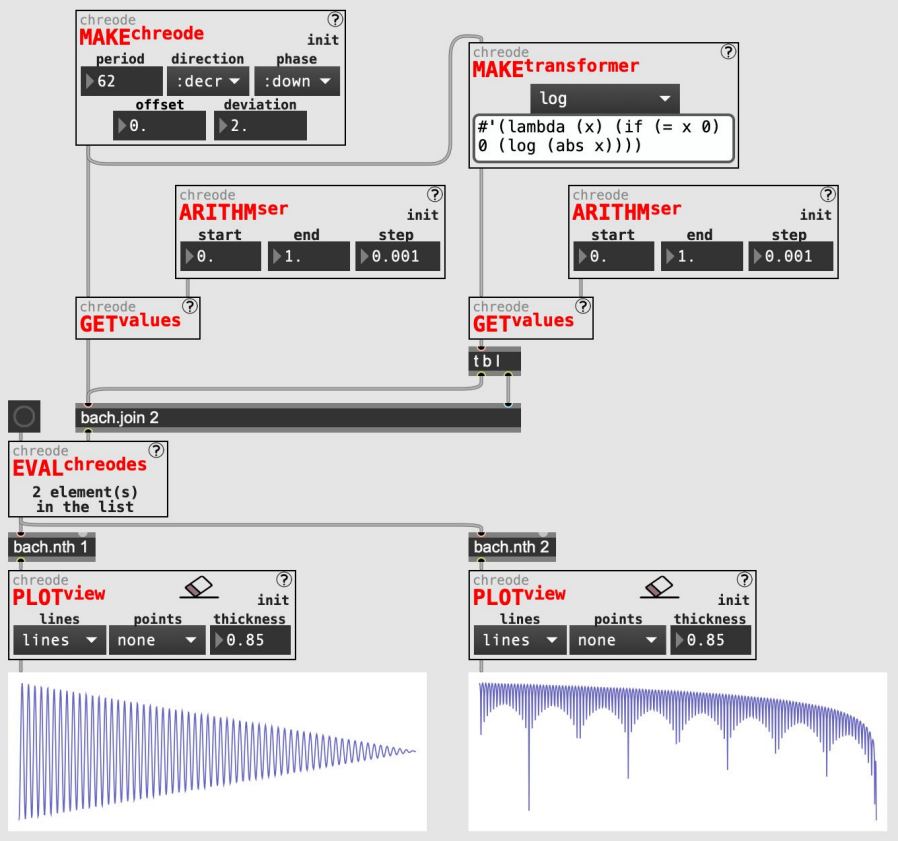
---

[52] Contributors to the original code of Chréode and Transkaija include Pierre Cointe (Formes), Jacques Duthen and Pierre-François Baisnée (Esquisse). A translation of Chréode from Le Lisp into CommonLisp was made by Kilian Sprotte in 2010-11, when porting the system into OpenMusic and PWGL. I relied mostly on Sprotte's version (*OM-Chréode*) as a reference, helped by further research with original documents provided by Barrière, to realize the version presented in MOZ'Lib.

[53] *Le Lisp* was created in 1980 by the French Institute for Research in Computer Science and Automation (INRIA). At the time I realized this port, it seemed that Le Lisp was being deprecated by INRIA. Very recently it was brought to my attention that INRIA agreed to migrate its source code to FreeBSD License in 2020, and today a large subset is available openly on Github. Hopefully this will facilitate the access to more CAC tools developed in that dialect.

[54] cf. Jean-Baptiste Barrière, « L'informatique musicale comme approche cognitive: simulation, timbre et processus formels », in La musique et les sciences cognitives, Ed. S. McAdams & I. Deliège, Pierre Mardaga, Brussels, 1988.

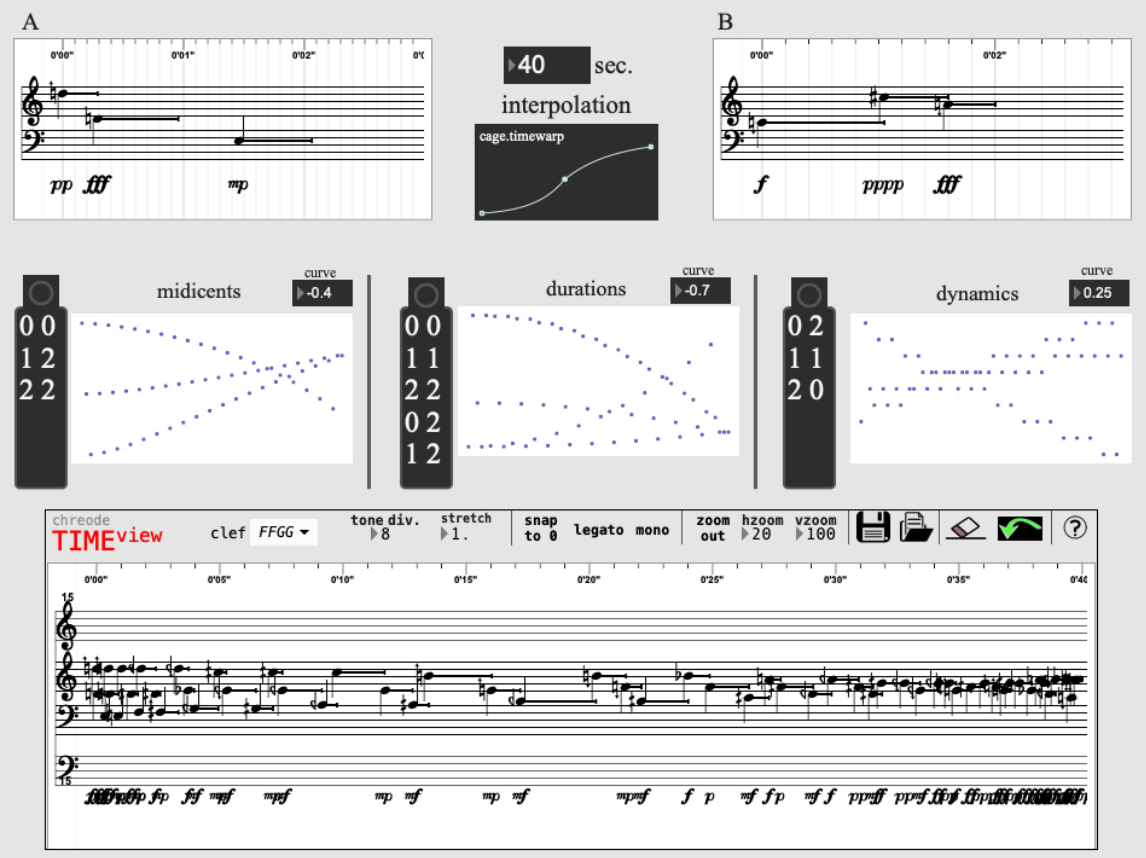
this time used for *visual coding*[55]. Each of them generates a piece of Lisp code dynamically, and the user needs to connect them according to the original syntax of the system. The resulting code is sent for evaluation on the fly, as the user changes the modules' parameters, and the results can be observed either as graphs or *bach* scores in real-time.



Transkaija is a smaller, more specialized library, revolving around a single Lisp function and some of its variations. Unlike *Chréode,* I had to perform a full reconstruction working for the original sources in *Le Lisp* (almost undecipherable for me) and some academic publications by Saariaho herself in that period. I already mentioned my interest on music interpolation, and today *Transkaija* remains a powerful example of the "debate" around this idea among composers in the early days of CAC. If one

[55] As pointed out by Barrière in the beginning of our collaboration on *Chréode*, as he was getting more familiar with the system, patching with code generating modules can easily bring confusion between the semantic and pragmatic role of a given abstraction. While patching in PWGL seemed very similar to Lisp code represented as a patch tree (the lexicon and rules are broadly the same), each box in that environment was able to return its own intermediary result with one key stroke. On the other hand, code generators in MOZ'Lib return nothing but nested Lisp code, which is evaluated later down the line. This makes direct patching more cryptic, one of the reason why I later created the *lisp.quickeval* abstraction as a way to evaluate Lisp code anywhere in a patch more easily.

looks at common implementations of music interpolations, I believe the most *literal* ones — based on the regular sampling of a line drawn between two values — were the most problematic. They generally ignore the notion of musical function : for instance, an isolated note inside a chord will not be connected with the closest note in the opposite chord, but with the one that has the same position in the list. *Transkaija* takes a more subtle approach. First, it allows to interpolate in parallel between several parameters of fully formed music entities, i.e. not only chords or pitch profiles, but also durations and dynamics. Secondly, the user can determine explicitly which elements in the two phrases should point to one another, thereby generating much more original and musically precise scenarios. If we return to chord interpolations, one can decide which note in the first chord will be linked to which one in the second chord ; a single note can even be split progressively into multiple targets, and vice versa[56].



[56] For more information on this algorithm and its musical applications, literal or not, cf. Kaija Saariaho, « Shaping a compositional network with computer », in *International Computer Music Conference ICMC Proceedings*, University of Michigan Publishing, 1984 and Landon Morrison, « Encoding Post-Spectral Sound: Kaija Saariaho's Early Electronic Music at Ircam, 1982–87 » in *Music Theory Online, Journal of the Society for Music Theory*, 27-3, 2021.

• Mikael Laurson's *PWConstraints*

Not a "guest library" per se, *PWConstraints* was provided in the kernel of MOZ'Lib since the very beginning, and used for some of the pedagogical modules. *PWConstraints* was originally created in 1995 by Mikael Laurson, PatchWork's original creator, then ported to PWGL. Possibly for lack of trust after OpenMusic succeeded to PatchWork, Laurson preferred to keep the sources of PWGL closed, even though the software itself was distributed freely. This is why the version of *PWConstraints* that is now included with MOZ'Lib was based on a port realized by Örjan Sandred from PatchWork to OpenMusic in 1999. This version may not be the most up-to-date iteration of Laurson's system, but luckily it was still available as open source and fully functional.

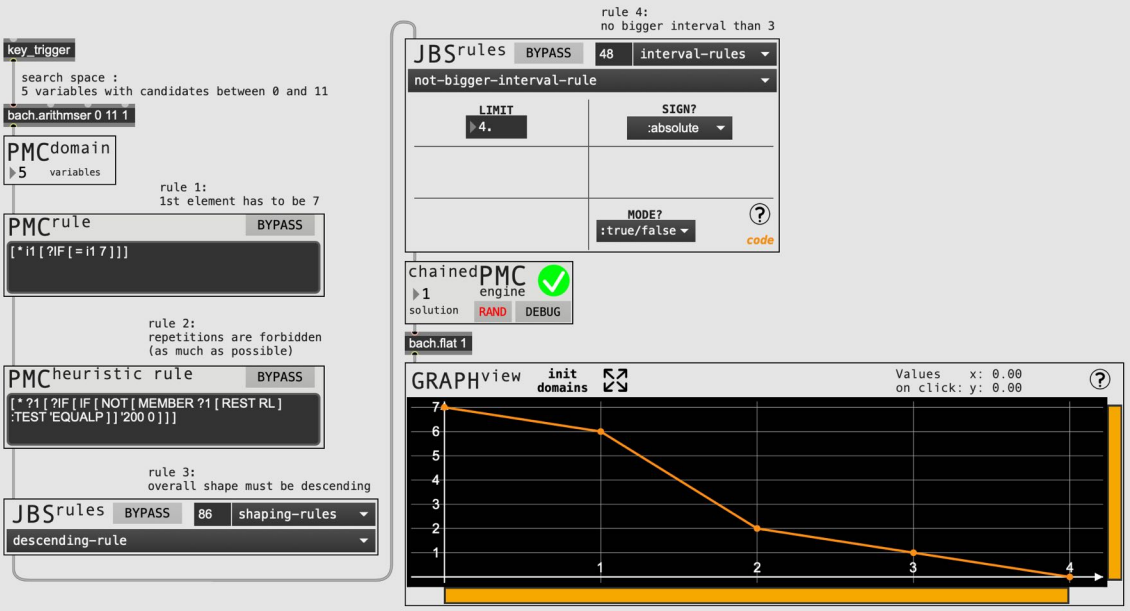
The library revolves around the PMC engine, a function searching for potential solutions to a musical problem described by the user with a set of rules. Solutions are generally a sequence of pre-determined size for a *single* parameter at once, i.e. either pitch, rhythm, or any other symbolic representation that the user wants to work with. The rules are written with the help of a dedicated syntax[57] based on the Lisp language, which makes it quite a demanding system for newcomers[58]. To facilitate the approach with *PWConstraints*, some pedagogical tools have been developed in particular Jacopo Baboni Schilingi's library *JBSconstraints* and the *JBSrules* module, presented below.

More specifically in MOZ'Lib, the mode of patching and interaction with the library has been largely redesigned to fit the usual practices of Max users. I created a small set of bpatchers dedicated to *PWConstraints*, inspired by initial discussions with Örjan Sandred when porting together his Cluster-Engine library. Instead of a central engine receiving different rules from multiple locations in the patch, which was more common in PWGL, Sandred proposed the idea of a *chained* engine. There, the candidates (defined in *PMCdomain*) visually traverse a string of rule modules (*PMCrule*, *PMCheuristic-rule*, *JBSrules* or else) acting metaphorically like filters, and eventually reach the *PMCengine* module where the computation really takes place.

---

[57] For a thorough description of the PMC syntax, one can refer to the documentation of MOZ'Lib, as well as the fifth chapter of Mikael Laurson's doctoral dissertation, which is entirely dedicated to PWConstraints and some musical examples. cf. PatchWork*: A Visual Programming Language and some Musical Applications*, Studia musica no.6, Chapter 5 (pp.145-186), Sibelius Academy, Helsinki, 1996

[58] Learning how to use PMC syntax was the main reason why I started to learn Lisp in the first place. For anyone interested in becoming familiar with this language in a reasonable amount of time, I strongly recommend the following book, available for free on the author's website, in particular its first chapters. cf. David S. Touretzky, *Common Lisp: A Gentle Introduction to Symbolic Computation,* 1st edition 1984, revision 2013, Dover Publications. Also available for free on the author's website : https://www.cs.cmu.edu/~dst/LispBook/book.pdf



Generating a simple sequence of numbers with MOZ'Lib's PMCengine module —
Four rules written using PMCrule (true/false), PMCheuristic-rule and JBSrules (*JBSconstraints* library)

• Jacopo Baboni Schilingi's *JBScmi*, *JBSprofile* and *JBSconstraints*

I studied composition and CAC in parallel with Jacopo Baboni Schilingi in the Conservatory of Montbéliard between 2008 and 2013. When we met, I immediately expressed to him my interest in CAC and yet my difficulties learning with PWGL's rather sparse documentation. He made me understand that I would not go very far without external libraries, and gave me all the ones he had, including his own[59]. Those three libraries were in fact the first ones that I studied in depth, and I later contributed to their documentation under Baboni Schilingi's supervision. Developed between the early 1990s and late 2000s, the three combined propose a vast panorama of approaches to CAC. Some of these functions have become "classics" of CAC, and can be found in multiple environments of the PatchWork family. I believe they also show an interesting evolution in term of aesthetics and areas of interest during the composer's career.

JBScmi is a non-homogeneous collection of personal techniques for generating and manipulating musical structures, in particular dealing with matrices, complex permutations, groupings and segmentations. In my opinion they correspond to a more "standard" approach of CAC, developed in the earlier years of Baboni Schilingi's trajectory.

JBSprofile is primarily focused on controlling the *shape* or contour of any kind of music

[59] For reference, I maintain a public repository of more than 50 user libraries which were circulating in CAC circles until PWGL's development was interrupted. cf. https://github.com/JulienVincenot/PWGL-community-library/

parameter, including melodic profiles, according to a variety of geometric transformations : perturbations, reflections, derivations / integrations and interpolations. The library, initially written in collaboration with Mikhail Malt, reflects the interest of the composer for the topic of music morphology, which he also explored with the *Morphologie* library in collaboration with Frédéric Voisin.

Finally, *JBSconstraints* is a large collection of rules written in PMC syntax, and compatible with all the *PWConstraints* modules. In Max, the library was implemented as a single bpatcher named *JBSrules*, where the interface and parameters dynamically change depending on which rule is selected. Rather complex scenarios of music generation can be realized by combining multiple instances of this module, even with minimal knowledge of Lisp and PMC syntax. For this reason it constitutes the best entry point to experiment with music constraints before studying more complex approaches.

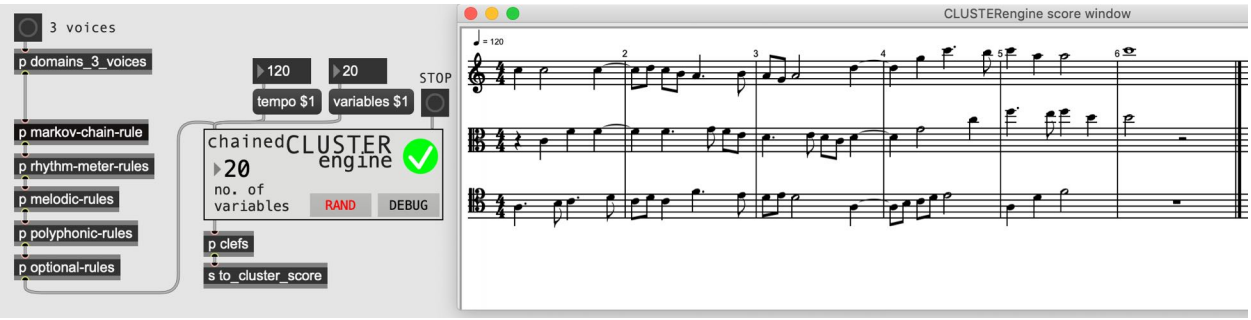
• Örjan Sandred's *CLUSTERengine*

Initially created in 2010 as a user-library for PWGL and fully ported to MOZ'Lib in 2020, the Cluster-Engine is a system for solving musical constraint satisfaction problems. Its name refers to the idea of using a cluster of smaller search engines (similar to Laurson's PMC) that collaboratively search for a solution, similar to the idea of a cluster of computers. Results take the form of a full score which satisfies a set of rules defined by the composer. The Cluster-Engine breaks down the musical score into several sequences of variables, each being managed by a different engine. A single voice is represented by two sequences : one for pitches and one for durations (including rests). One additional sequence represents the metric structure of the score, shared by all the voices.

I believe the Cluster-Engine is a particularly expressive tool for two main reasons. First, it brings a lot of flexibility when defining the initial domain or *search space* : candidates can be either single values or *motifs*, i.e. predefined sequences of durations, pitches or chords, intervals, etc. A carefully designed search space can have a strong impact in generating musical results, even before adding a single rule. The second reason is the great variety of ways in which rules can be written, how they can affect properties or relations not only in a single parameter of the score, or in a single voice (between pitch and rhythm), but also between a voice and the metric structure, between different voices, etc. The engine also introduces sophisticated approaches to *back-tracking*[60], making it possible to generate complex results with minimal time. This makes the interactive loop when composing much more direct and intuitive in my experience.

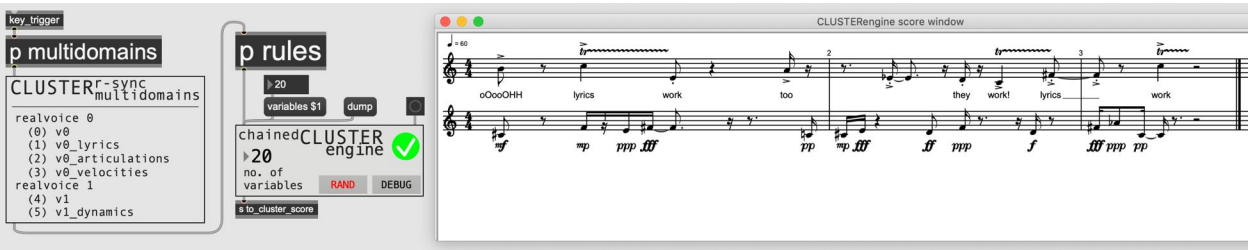
---

[60] Back-tracking is the way in which a constraint engine, when building a solution, is able to retrace its steps so to speak, when it encounters a logical dead end, and keep looking for allowed solutions. This was traditionally performed in a deterministic way, exploring all the branches of the tree of solutions, thus wasting a lot of intermediate computations. Sandred introduced a more efficient solution with the idea of *back-jumping*, allowing to the engine to fix an earlier portion of the score without sacrificing all the music that was generated in between.



Example of polyphonic constraint generation by O. Sandred, attempting to imitate certain rules from 16th century vocal style — including Markov chain analysis of the rhythmic behavior in some Palestrina excerpt

The CLUSTERengine interface I designed for MOZ'Lib is also made of a large collection of bpatchers. They further develop the idea of the *chained* engine introduced with *PWConstraints*. One can build a functional generator of polyphonic scores by first chaining any number of *voice-domain* modules (each dealing with pitch and rhythmic candidates for a given voice) and a *metric-domain* module in direction of the main *CLUSTERengine* module itself. Rules can be introduced in between the domain and the engine. A rule typically requires two things to work properly, an *accessor* and a code part. First, the accessor or rule applicator determines which music parameter (or relations between those parameters) will be visible to the rule. The code part will check the properties of musical parameters made visible to it by a given accessor, it is "the rule itself" so to speak. Rules are written as Lisp lambdas — either as pure Lisp code, or visually by using Lisp scripting boxes — and behave similarly to PMC *true/false* and *heuristic* rules. Depending on the accessor's scope, the rule will have an impact on rhythm, pitch, relations between different parameters in a given voice, harmonic or polyrhythmic relations, etc. One of the main limitations of the Cluster-Engine system is that it can only deal with pitch, rhythm and meter by default. When porting the system to Max, I proposed an experimental system called "Multidomains", allowing to generate scores with a variety of parameters by taking advantage of the different *slots* (articulations, dynamics, lyrics, etc.) assigned to any note in *bach* notation objects.



Example of using "Multidomains" on two separate voices — rules are applied not only on pitch, rhythm and meter, but also lyrics, articulations, velocities, dynamics, and their mutual relations

Finally, I would like to mention ongoing projects related to MOZ'Lib, as well as further developments I plan for this package.

First, I would like to keep extending the collection of available libraries, in particular those that I believe deserve a more substantial documentation, among which :

• Frédéric Voisin's *FVmorphologie*, a library dedicated to computer-aided analysis, classification, pattern recognition and reconstruction of any numerical profiles or symbolic sequences.

• Torsten Anders' *CLUSTERrules*, a collection of pre-written rules compatible with the CLUSTERengine library, and concerning various musical scopes (rhythm, melody, harmony, counterpoint, etc.), in a way similar to JBSconstraints' approach.

• my personal library *JVcomponents*, developed in PWGL between 2008 and 2015, presenting various CAC techniques revolving around analysis, supervised interpolation (numerical and symbolic), instrumental constraints, generative constraints, etc.

At the moment, as the main developer of MOZ'Lib my intervention is required whenever users want to add new libraries like these. To address this limitation I plan to implement a long overdue system allowing users of MOZ'Lib and PWforMax to import, develop and share more easily their own Lisp libraries.

I also plan to keep developing two chapters of the package that appeared rather recently :

• *MOZ Tools* : a selection of patches created collectively during the online workshops of *bach*-based CAC I lead regularly since the fall 2022, and which I believe can be useful to the community.

• MOZ Analysis : a relatively small collection of computer-aided analysis paradigms inspired by Jacopo Baboni Schilingi and Frédéric Voisin's *Morphologie* library, which I intend to continue expanding in the near future. My own practice of analysis has been strongly influenced recently by the two seminars led by Evan Johnson and Ray Evanoff at Harvard. During these seminars, I began to test these tools more rigorously — experiencing notable successes and failures — in the practical context of music analysis. I hope this will eventually foster broader discussions and provide more opportunities to practice what I call "creative computer-aided analysis", and perhaps even lead to a more ambitious project (book or other format) dedicated to this practice, as suggested by my colleagues during the seminars.

Part II

Graph theory or catastrophe theory?

Game Theory was my main creation project since the fall of 2018, but after investing substantial amounts of time and energy, both personally and with a few collaborators, it was de facto abandoned in the end of 2021. For this last section, I want to try and describe what this project was about, what I think was interesting about it, and perhaps what made it so difficult to get it out of the ground. Moving past the frustration of this one "failed" attempt, I also want to discuss how my overall practice has evolved in parallel of these reflections and experimentations. The project itself is definitely behind me, but I still hope to develop some of its key aspects in the future.

First of all, *Game Theory* was for me a speculation about format, and an attempt to express what the idea of "interactive music" really implied for me. And by that I meant not interactivity between instruments and machines — the usual suspect — but interactivity between the audience and the whole musical apparatus. When I first began to consider it, I intended to bring together my different areas of interest in music creation, beyond the usual mode of presentation as *a piece*. These discussions first emerged in 2016-17 with a few members of Listen ensemble, founded by guitarist Rafaël Carosi[61]. We really started developing the project in 2018-19, initially titled *Theory of Fun*, for five musicians — flute, clarinet, electric guitar, double bass and live computer — and only later organized for a public premiere in March 2020. Our goal was to see if we could "enhance" the typical mixed music performance with the generative score (and new possibilities of real time CAC brought by *bach*), somehow influenced by the audience via smartphone interfaces, and incorporating some degree of improvisational fluidity and flexibility. My main inspirations at the time were Umberto Eco's writings on the "open work"[62], and ideas borrowed from video game design — in particular Raph Koster's *A Theory of Fun for Game Design*[63]. Later in our discussions, we placed greater emphasis on the idea of interactivity and immersion, and the blurring between performance space and exhibition space, similar to what is commonly found in visual arts and multimedia installations.

---

[61] These discussions came soon after a residency at the Spatial Sound Institute of Budapest, in September 2016, in which I acted as artistic consultant and laptop performer. The program revolved around a performance of Terry Riley's *In C* by 12 musicians of the ensemble, remixed by electronic music producer Murcof (Fernando Corona) on the multichannel system designed by 4DSound studio. cf. https://www.youtube.com/watch?v=RWTpf8ZPOW8 (accessed June 2024)

[62] cf. Umberto Eco, *The Open Work*, Harvard University Press, 1989 (orig. 1962)

[63] cf. Raph Koster, *A Theory of Fun for Game Design*, 2nd Ed., O'Reilly Media (2014)

All of this led, somewhat "naturally" but without clear anticipation on my part, to a kind of fantasized "art total" project — one might say as an alternative to a format like opera, for instance — more in line with my own value system and music background. I now believe that the outbreak of the Covid-19 pandemic spared us from presenting in public a very fragile network of barely functional placeholders. As the project grew in ambition, it required more involving collaborations with different fields of expertise, for visuals and scenography, but also for the design and programming of smartphone interfaces which clearly exceeded my own skills. On the music side, I needed to compose (i.e. program) the live generative score, which was a rather new problematic for me, but also imagine an electronic part that could adapt to it in satisfying ways. I was also de facto in charge of the artistic direction, overseeing all the other production aspects, for which I had no clear vision and understanding. I learned the hard way that leading such a multifaceted project was far too ambitious, even with a small team surrounding me — a total of 10 persons were involved at one point or another, including musicians.

As we imagined it, *Game Theory* was purely and simply unrealistic : too big and too expensive, but also *technically* unfeasible with the time and resources at our disposal. Beyond questions of funding, the artistic expectations we had set for the performance were clearly too high. For almost two years, we elaborated countless prototypes, but also conceptual and technical hypotheses (including the imagined behavior of the audience) that were extremely difficult to verify. This resulted in endless discussions, many of them hardly productive, which in turn generated plenty of misunderstandings and frustration for team members including myself. Apart from Rafaël Carosi, with whom the creative exchange, dedication, and friendship remained constant from beginning to end, everyone else involved in Game Theory could only afford limited time to commit to it at various points. To be feasible, I imagine such a project would require a stable creative team working together consistently for at least 3 or 4 years, similar to a small independent video game production. I now know for certain that I will never be able to secure such substantial funding for a creation project in my own name, unless I invest my own money or transform it into a stronger collaborative effort. Today, I tend to believe that it would be more appropriate to develop this as a collaboration between multiple composers and artists, potentially in a pedagogical context, which would allow to share the workload more evenly and efficiently.

A big part of my interest for this project was theoretical : I wanted to find a good approach to composing music in a non-linear way, whether that involved local behaviors of the score or more global decisions for how the music can be segmented and evolve during such a performance. This is not a trivial problem for me, simply because as composers we commonly think of the score as a sequence of events following a pre-determined order, from beginning to end. Just imagining that each section of a piece

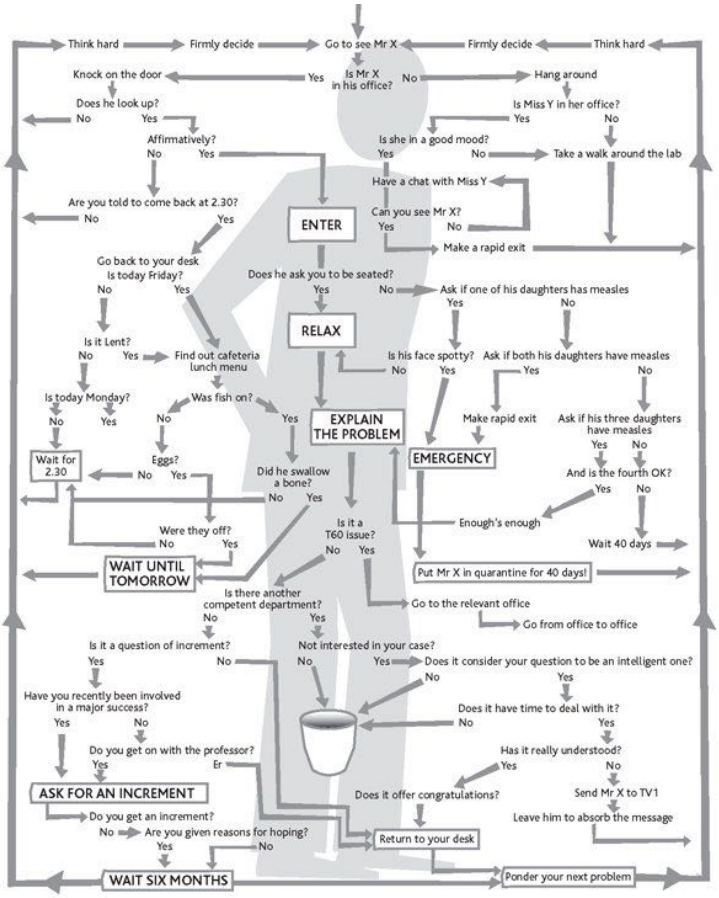
could have a satisfactory way to transition to any other section, including between very different coexisting music languages, is a challenging issue in itself. As a whole, this is a large writing problem, which depends on the limitations of what can be encoded in a given programming language. One cannot easily program a system that will indefinitely (i.e. recursively) extrapolate intelligent or expressive musical decisions from a constantly evolving context[64]. We needed to restrain things a little in order to keep them intelligible, and to be able to manipulate them effectively. In other words, I had to compose a family of *placeholders* which could be called or recalled on demand, in any order, share some memory of what had happened before, and take decisions accordingly. This is why, quite early in the project, I started to explore systems written as an interweaving of discrete *graphs* or flowcharts.

Always a passionate reader of works from the Oulipo movement, I was fascinated by how Georges Perec used this kind of graph to determine his 1968 book titled *The Art of Asking Your Boss for a Raise*[65]. In my first reading, I was struck by the rhythmic quality of the text, resulting from the tightly-knit branches in the flowchart that served as its model. As the story progresses, the reader realizes that the narrator, as he gets closer and closer to a meeting with his boss, is in fact kept away from the raise by the shape of the graph itself. It appeared that the book — 70 pages written as a single phrase — was the result of a thorough, systematic exploration of the whole graph, from easier to more complex paths. Years later, after a deeper analysis, I realized that the author first *seemed* to exhaust all the possibilities given by the graph — the so-called *stratégie de l'épuisement* — but later in the text introduces some points of rupture. For instance, Perec exploits local loops (or traps) in the graph as some sort of "rhythmic accumulators", progressively increasing tension and a sense of helplessness. In these moments, the system begins to crack under the absurdity of the administrative machine — in other words it *crashes* completely, like a computer caught in an infinite loop. This is a property I tried to play with in a small experiment during Claire Chase's workshop for flutists / composers in 2018 — *snails are an excuse to eat butter and garlic,* a hybrid between a graphical score and music theater. This was for me a playful yet very theoretical experiment, in the sense that I would never present this to performers, and certainly not dare to perform it myself.

---

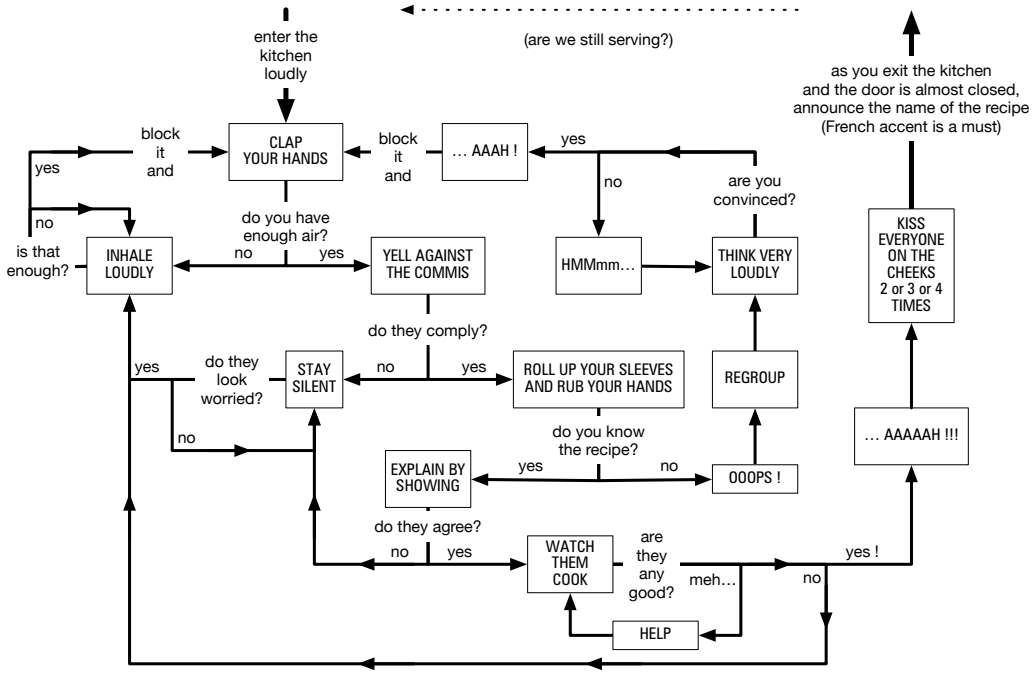
[64] I need to mention that, at the time when I was developing *Game Theory*, the very idea of tools like ChatGPT or Midjourney was completely unimaginable. And yet at the time I am writing this, we are still awaiting the advent of operational equivalents to those in the music domain, especially for the symbolic level of the score.

[65] cf. Georges Perec, *L'art et la manière d'aborder son chef de service pour lui demander une augmentation*, Hachette-Dunod, 1968



Georges Perec's flowchart for *The Art of…*

SNAILS ARE AN EXCUSE TO EAT BUTTER AND GARLIC



This is probably due to a lack of knowledge on my part, but my first impression is that composers rarely use this mode of graphical / conceptual organization for the purpose of creating more open, non-linear music situations. I want to put aside electronic music here, where composers are already more often in the position to design instruments, pieces or installations *as graphs* — if only in the form of a patch or code. For instrumental music on the other hand, I have seen on occasion composers use graph-like representations in paper sketches, but only as a space to set up and explore by themselves. An expressive tool for thinking the music, possibly a structural mechanism to organize materials, establish chains of events or the evolution of music properties, but rarely more. In the end, they would give musicians or the audience one single modality to experience these systems, and that is within a « protected path » so to speak, the particular instance they chose for the piece. I believe this is a missed opportunity**,** especially in a music world where composers tend to create very different, personal languages and systems. Musicians and the audience could much better internalize the behavior of those systems, live inside of them, and potentially appreciate the music better and more in-depth, if they were able to explore those more freely, interactively. And also, potentially, in ways that the composers themselves didn't envision, similar to the phenomenon of "emergent gameplay", where complex situations and modes of interaction arise unexpectedly from even simple game mechanics.

Here I need to make a distinction between *analog* open works and *digital* ones. Of course, important examples of non-linear or open scores exist throughout the recent repertoire of instrumental music. The most well-known are in fact the easiest and perhaps most anecdotal ones, for instance asking musicians to scramble pages of the score before each performance. More complex generative, behavior-based or rule-based works, often categorized as *game pieces*, bring much more depth to the idea. I could mention Iannis Xenakis' *Duel*[66], John Zorn's *Cobra* or Georges Lewis' *Artificial Life 2007* as some remarkable examples I have encountered during my research. There, the system is truly composed as such, departing considerably from the traditional score, and often based on a very compact amount of instructions. Yet again, I need to clearly differentiate this family of (analog) works and interactive pieces using digital technologies, in a way similar to how we might separate board games and video games. The gap between them is not just a change in medium, and in fact the rules in both domains can be very

---

[66] I was only recently made aware of a remarkable experiment, conducted by Marco Liuni and Davide Morelli in 2006, based on a software reimplementation of Xenakis' game pieces. This seems to go precisely in the direction I am interested in, in the sense that the interactivity component is not reserved to the musicians, but was experienced by audience members. « Liuni & Morelli rendered *Duel* as a live installation in which members of the public could take the role of the conductors. Their movements were analysed by computer vision algorithms to drive the score, which was, in turn, visualised on a large screen. This rendition concentrated on the interactive participation of the human players, and the musical events prescribed in the original score were realised using audio samples instead of real orchestras. » cf. Stefano Kalonaris, « On Xenakis's Games of Musical Strategy », Musmat.org, 2022

similar, even identical. Yet there is a crucial difference : in video games, rules are enforced automatically, *in the background*. A player's enjoyment is usually not based on remembering (or even understanding) those rules, but instead on experiencing situations *emerging* from the rules via a complex of input devices and sensory feedback. If we return to music, and open pieces conceived in the "analog" domain : with the exception of participative works of course, the audience has usually no direct experience of the openness of the system, which concerns only the performers. What the audience members perceive is a result, the surface of what the system can offer — and in fact if they hear the piece only once, as far as they know it is perceptually equivalent to a fixed score.

While the torture of the infinite scroll, more representative of the Web 2.0 and modern social media, emerged naturally as a metaphor when developing the score of *Game Theory*, I have recently been more intrigued by the rougher structures of the "old Web" and its HTML mazes. This leads me to another idea which I feel has not been much explored in music — or at least I was not able to find much traces of it : the concept of *hypertext*. Today we are all familiar with this idea to some extent — although traditional websites are increasingly replaced by more opaque, dynamic web applications. To simplify greatly[67], we can define the hypertext as a network (or graph) of texts in any language or media, interconnected by uni- or bi-directional *hyperlinks*. Originally, a hypertext is only limited by the size and design of the graph that is offered, and reading becomes de facto a non-linear, creative experience for the reader-visitor. If some of the more ancient websites could be traversed comprehensively — not the encouraged mode of exploration, obviously — most modern platforms take content generation to the point that there is no clear, readable graph structure to follow anymore.

I tend to associate this idea of hypertextuality with a short period of recent art history known as *web art* or net art[68], which I feel had its most flourishing period between the mid-1990s and the early

---

[67] A much more precise definition of the notion of hypertext in music, more theoretical and not related to a particular technical implementation, is given by composer Nicolas Jacquot : « Hypertextuality is the aspect of transtextuality that designates the transformative links existing between a source *hypotext* A and a *hypertext* B. Gérard Genette specifies that the nature of the transformations potentially at play ranges gradually between two modalities : (morphological) derivation on one hand, and (stylistic) imitation on the other. […] In a symmetrically inverse way, it means : "saying the same thing differently / saying something else similarly". Regardless of the mode of operation, the existence of hypertexts in music is undoubtedly as old as music itself, whether transmitted orally or in written form. Nevertheless, it is of course the intrinsic properties of writing, linked to the high manipulability of the media, that make it the favored medium for hypertextual operations and organizations. »
cf. Nicolas Jacquot, *Métalepse - Réflexion de la figure dans le domaine de la composition*, Master's Thesis, University of Reims Champagne-Ardenne, 2016

[68] For a clarification on the terms, cf. Jon Ippolito, « Ten Myths of Internet Art », in *Leonardo* 35(5):485-498, 2002

2000s[69], until the coagulation of the whole internet around a few proprietary platforms. I was too young myself to experience much of these works at the time they were created[70], and today a lot of my interest probably results from both theoretical / technical curiosity and fantasizing what those works actually could have been. And in particular, I wonder about musical applications of these ideas at the time — if they existed, they probably did — inside but also outside of the traditional concert hall or exhibition space. By that I do not mean *works as systems* or *works as programs* in the larger sense — the more *generative* behavior we might expect today from a MaxMSP-based installation or a modern WebAudio-based real-time application — but really this sort of discrete organization of time and space that the quite limited, *slow* web art of the origins was relying on. Combined with the idea of the score as a fractalized object, I particularly wonder what this would have entailed to be technically and musically viable, whether as a solitary or collective experience for musicians or non-musicians alike.

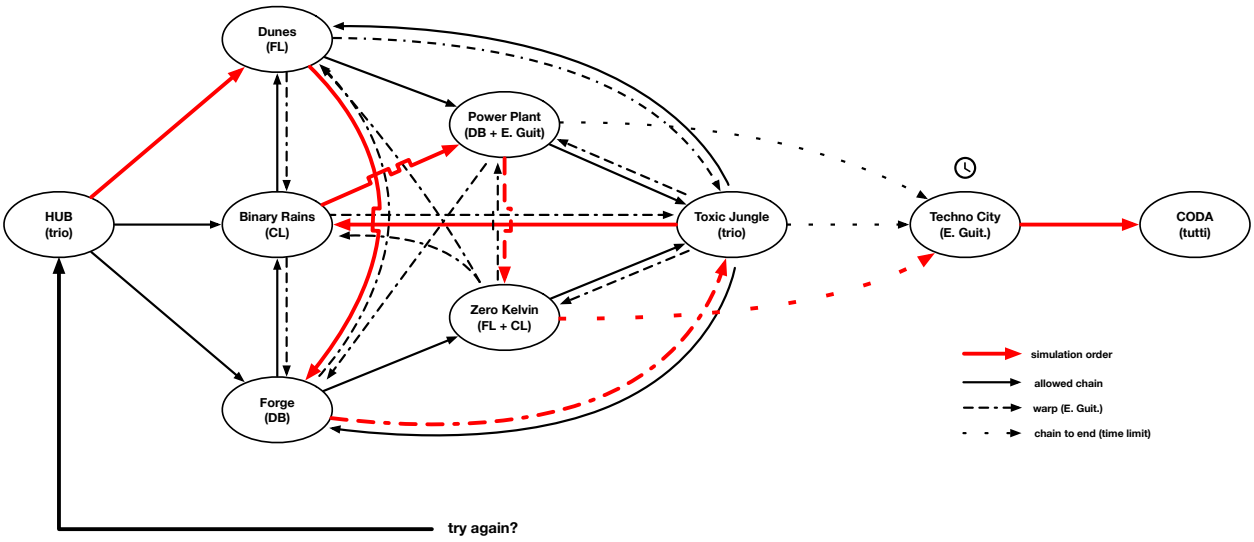
My original goal was to explore this idea with the piece *Hyperlinks - asset(s) failed to load properly*, written in collaboration with Irène Blanc (flutes) and Rafaël Carosi (guitars). I imagined this first as an experiment, a more manageable sub-set of what I wanted *Game Theory* to be. There, the musicians would freely explore a network of scores — or "pieces of music" in general — both fixed and generative, during the live performance. Unfortunately, a series of events during the conception of the piece led me to scale down my ambitions. At some point, I considered the possibility of transforming this piece into a genuine *web-based* interactive object, building upon the large amount of materials we were able to record with the musicians in January 2024. Yet I feel the idea came a little too late, and I did not find the time and energy to reconfigure the project so deeply — perhaps as a future, interactive version. However, I would still need to establish a clear conceptual and technical framework for this kind of works to exist, along with my main CAC preoccupations.

---

[69] « This will be the story of a generation, now lost and forgotten: the first to invest the Web in the mid-90s, to the point of making it a mode of existence experimenting with other identities, other genres, other relationships, other fraternities, other works. […] The Web was never a technical means of expressing our subjectivities. It was the art of intensifying what each and every one of us secretly felt, namely the vacillation of so-called identities, our shifting sands, where we formed the project of infinite HTML labyrinths in which we lost ourselves and in which Internet users could drift without ever seeing the end of it. […] We dispensed with the traditional prescribers of the art world – critics, theorists, curators, collectors or galleries – because we could publish directly on the network, addressing ourselves to one and all. At least, we had this illusion, this desire. »
cf. Grégory Chatonsky, « Jimmy Owenns, Kennedy James, Govy: the existence of the network », Published on the artist's personal website chatonsky.net, 2023.

[70] The little I effectively knew, and yet my strongest memories of this area of experimentation, existed mostly in the form of Flash-based creations, such as the *Modern Living / Neurotica* series (1998–2001) by artist Han Hoogerbrugge. A considerable amount of net art works created during that period have been lost since Adobe Flash (formerly Macromedia Flash) was discontinued in 2020. For a panorama of works that are still accessible and (laboriously) maintained as functional experiences, cf. for instance the Net-Art (net-art.org) and Rhizome (rhizome.org) archives.

After this digression on the hypertext, I want to return to *Game Theory* for a moment, and describe a little more in depth how its internal rules and behavior were organized. The main patch I composed in the fall and winter 2019-2020 was a combination of two graphs, each controlling one aspect or layer of the generative score. The higher layer, somewhat more simple to enhance readability and feasibility, was trying to introduce openness into the global behavior of the performance. Instead of being divided into sections, we imagined to segment the project in different *levels* or worlds, an allusion to the chaptering of 1980s video games. Each of them developed a very distinct musical behavior, showcased a different soloist (or subset) within the ensemble, a "temperature range" (cf. below), and of course was associated with specific electronic and visual situations.

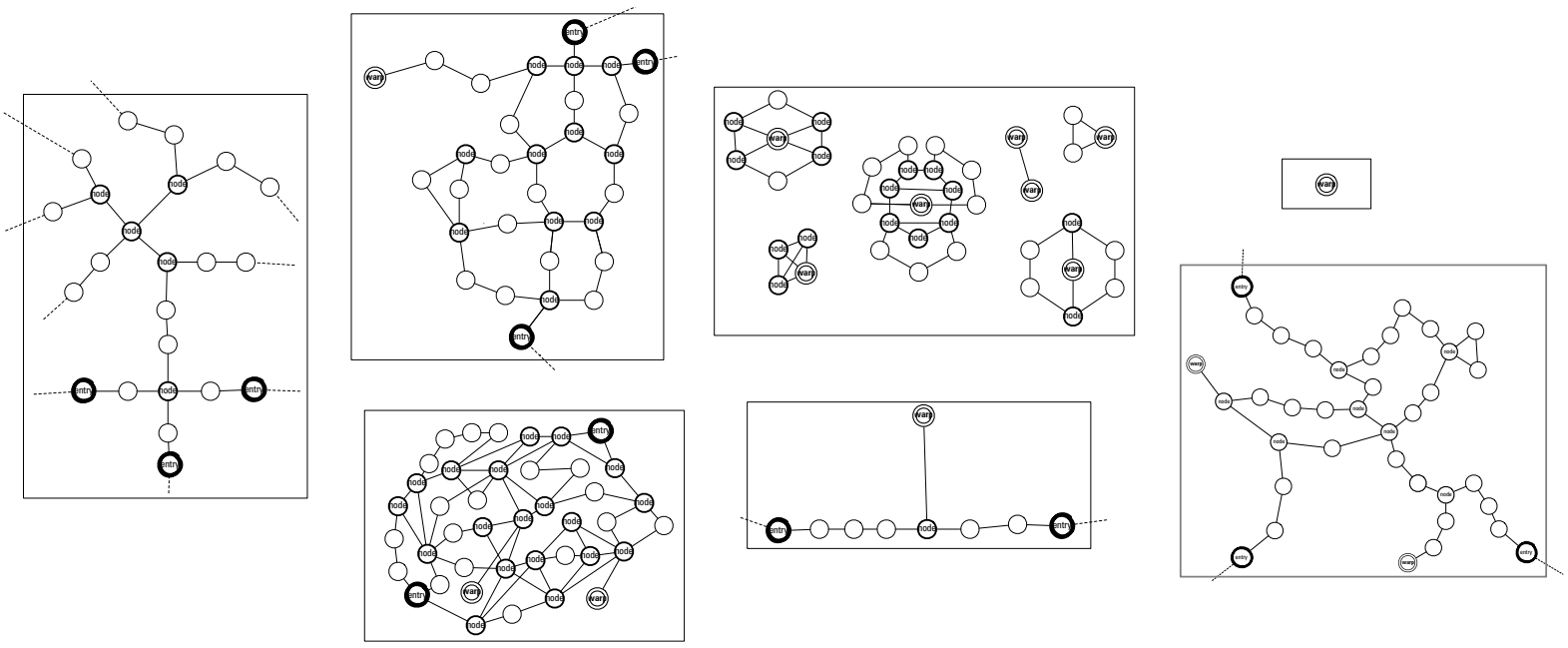


Tentative "global graph" for the 2020 version of *Game Theory*

First the audience would enter the performance space with the apparatus already started — similar to an installation : a sort of "hub" or neutral level of the piece. After a while, musicians would take place and start performing, following the score in one of a few entry levels, decided randomly. Levels were partially interconnected, in such a way to foster at the same time a certain unpredictability and yet orient the overall performance to a certain direction. For instance, not all but certain levels were to remain inaccessible for a while, until certain conditions — audience interaction, minimum duration — would unlock them. Each performance would traverse the graph in a different order, potentially repeating certain levels — and therefore taking the occasion to revisit and complexify them. Moving between two levels in the main graph would either result in an immediate shift or in isolated transition sections called "warp zones". Imagined as "teleportation tunnels" to reach remote parts of the graph — like the secret passages popularized by *Super Mario Bros*. in 1985 — these warp zones served as an additional segmentation

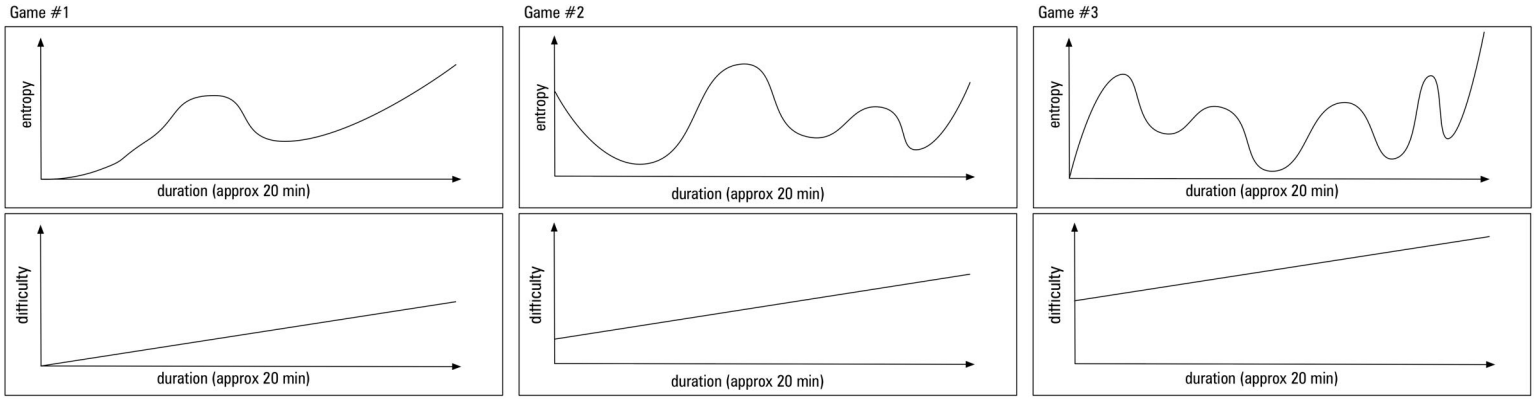
device, as well as a formal attractor towards the end of the performance. These rather short sections — moments of sudden acceleration of the music, before returning to "normal" navigation — would give a special role to the most important soloist (and perhaps antagonist?) of the piece : Rafaël Carosi's electric guitar. In the course of several warp zones potentially happening in a same "game", the guitar would progressively gain prominence, evolving from sparse improvisation to dense (quasi-"free jazz" like) virtuosity, all in anticipation of the final level which were to revolve entirely around the instrument.

In the lowest layers of the system, each *level* (or section) would be associated with its own dedicated local graph, each of them with very different shapes. I imagined those to control the music locally almost like a board game. The cursor would move along the graph in any direction, based on certain probabilities : each edge of the graph would be given a certain weight — translatable into time proportions — while each new node would correspond to a new "symbol" in the local structure of contrast. A possible sequence would be : A for 5 sec., then C for 12 sec., then H for 1.5 sec. before returning suddenly to A for 7 sec., then D, etc. Whenever a symbol returned, something similar should occur, based on the memory of the system. This simple structure would influence every aspect of the score that would be deduced down the line : soloist vs. tutti, polyphonic density, rhythmic motifs, playing techniques, etc., but also was planned to influence contrast and trigger events in the electronics part.



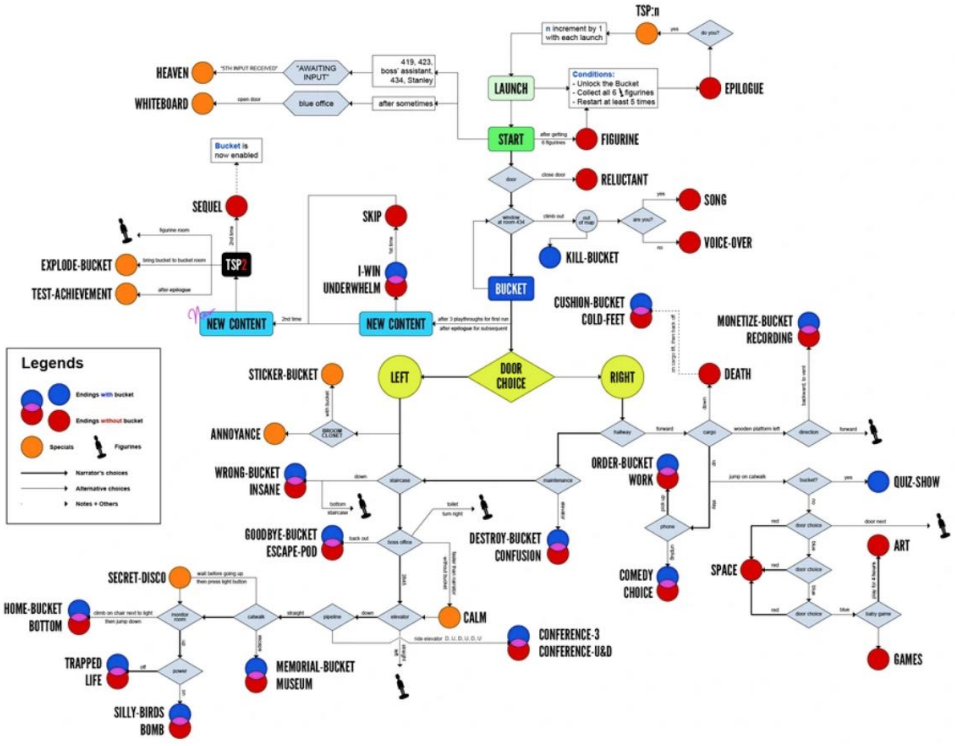
Original sketches for different typologies of "local graphs" in *Game Theory,*
including the starting hub, cyclic level, maze, archipelago, simple corridor,
all interconnected by direct links or probability-based warp zones ("anti-level").

I called “temperature” of the system the main parameter I imagined would be derived *somehow* from the audience’s behavior with their smartphones, and which would in turn influence several aspects of the score. Globally, if the temperature remained too low for too long, it would potentially accelerate the transition to another, “warmer” level. Temperature had also a strong impact on temporality when traversing the local graph (multiplying the weight of the edges), thus on the overall speed and contrastive rhythm of the performance — in a sense, analog to harmonic rhythm. The memory of the system, i.e. what each symbol of the local graph meant, was accumulated on the fly and kept until the global graph would “decide” to move to another level, thereby creating an entirely new situation. The temperature mostly remained beyond my control as the composer, yet I attempted to modulate it using an additional parameter I named “entropy”. This parameter was set to follow a fixed trajectory during a given session, and potentially across sessions in a same evening. I applied this idea of entropy in parts of the program which I believed could influence the perceived “difficulty” of the piece, for the musicians of course, but also how the audience perceived their effort throughout. To follow the metaphor, the interaction between (audience) temperature and (pre-determined) entropy shaped almost every musical decisions I took. Overall, I imagined these would manifest by changes in *states of matter,* between “solid”, “liquid” and “gas”, on multiple aspects of the performance : from the choice of rhythmic density, playing techniques and sounds qualities for the electronics, to what I would call the “cohesiveness” of the audience, that is how decisions would be taken, individually or collectively.



Entropy and difficulty curves in three potential successive sessions of *Game Theory*

Another example of graph-based organization of time which had a strong influence on *Game Theory*, was a video game called *The Stanley Parable* (2013), created by Davey Wreden[71]. The player explores a series of office corridors and rooms, accompanied by the voice of an omniscient narrator who constantly criticizes or mocks their choices. The real interest of this proposition lies in the exploration of all possible pathways within the game's space, each leading to a different outcome — a total of 42 "endings" in its latest edition — before returning to the starting point. Naturally, the game's "plot", essentially expressed through its ironic narrator, focuses on the illusion of freedom and lack of real choices for the avatar who remains a prisoner in this closed system, no matter how many time he attempts to escape it. This leads me to an interesting aspect of game design : *replayability*, which is usually considered by players as something desirable, especially for the most expensive games. This includes elements like multiple narrative branches and endings — and how they build our memory inside the game — or depth in the game's systems and mechanics (vectors of "fun"), which can convince players to restart and keep exploring it long after their first play-through. Even in a game based on a rather limited graph, a common trait in the experience of numerous players today seems to emerge in connection with this idea



Flowchart describing all possible endings of *The Stanley Parable*

[71] Interestingly, Wreden's game develops very similar themes to Perec's *The Art of…*, first of all the obvious kafkaïan administrative alienation, to the point that one of the game's most obvious branches is identical to the book's : finding access to the boss's office. Just like Perec, he uses local loops or traps very often, locking up the player temporarily, as moments to accumulate rhythmic and narrative tension — whether comedic or dramatic.

of replayability, namely "completionism" — the urge to explore a game in its smallest details. This aligns with a widespread (and rather invasive) practice in today's game industry, that is to quantify every aspect of the player's progress in order to increase artificially the time they spend in front of their screen. As a result, more and more players strive to reach the "100% completion" at all costs, and get rewarded with so-called "achievements" or trophies for the most absurd and repetitive challenges[72].

Now let's consider a slightly different example : non-linear interactive films such as Netflix's *Black Mirror: Bandersnatch* (2018), where the narrative is segmented by a series of choices between two or three options, leading to slight variations in the plot. These are extremely complex projects to put together, technically, and yet in spite of gigantic budgets they are not always considered convincing. A possible reason, is that they fail to maintain this illusion of freedom that most video games provide. The simple constraint of filming every sequence with actors, with conditions analog to the film industry, limits naturally the amount of pathways and choices that can be offered to the audience. Yet more recently, as I was watching a similar interactive film called *Breakout 13* (2023) by Beijing-based independent studio ALT Lab, I had a very different, more satisfying experience. There, the authors decided to provide an explicit representation of the rather complex network of alternate events. Navigating this graph, players are allowed to pick a point in the narrative and replay a situation by behaving differently, trying to unlock hidden portions of the story. To highlight the process of revealing the whole graph, a percentage value — in a way, a progress bar — is displayed at all times. "You only saw 78% of this chapter…". I believe *this*

Narrative tree of *Breakout 13*, with completion rate on the bottom left

[72] The author of *The Stanley Parable* pushed the absurdity of such "achievements" to the extreme, in particular with a trophy named "Go outside", which could only be unlocked by not playing the game for 10 consecutive years.

was the main vector of tension and interest in the game, which pushed me to keep exploring, not the plot itself : another game on top of the more visible, "Choose your own adventure"-type part of the experience.

Why is this important for *Game Theory*, beyond the video game influence I already declared? My experiences with both *The Stanley Parable* and *Breakout 13* make me lean toward an hypothesis : when talking about non-linear works, the audience's appreciation doesn't come only from choosing a path, which is not always guaranteed to be interesting for itself. In addition, it comes from contemplating the architecture of the graph, and exploring all of its ramifications. The more time is spent inside, the more one can feel the depth of the system. This may seem like a banality, but I also find interesting to reconsider what this idea of replayability could mean to us in terms of music appreciation. Composers and instrumentalists know very well that nowadays the probability of *reprise* of a piece after its premiere is extremely low, even for composers of high notoriety. As for *encores*, that is a piece replayed entirely or partially on a same day program, the idea has practically become legendary in our field. Apart from the performers and composers themselves (and perhaps colleagues) attending rehearsals, most music created in these situations will be heard only once, and then disappear. When one is lucky, a recording remains for the archive. Most likely, the only judgement the audience will ever have of months (or years) of work, condensed in a few minutes (or hours), will be based on that single glance, which can easily lead to misunderstandings. I am perfectly aware that demanding more today is unrealistic, for a variety of reasons, and yet I want to believe there are technical and/or artistic ways to try and improve the situation. This idea of replayability was in fact a determining aspect of *Game Theory* from the very beginning[73], and it irrigated practically every aspect of the project from the smallest composition details to larger production design[74]. One instance of the piece was expected to last 15 to 25 minutes, but we imagined several scenarios where the audience would be facing a "game over" (or fake ending) during the course of an evening. The audience would have 10 seconds to vote if they wanted to let it go or "try again" — insert

---

[73] During my first collaboration with Listen ensemble, performing Terry Riley's *In C* in Budapest's Spatial Sound Institute in 2016, the audience's feedback led the ensemble to perform an *encore* of the entire piece. This meant extending the concert by an extra one and half hours of music, which no one — in the ensemble or venue team — had anticipated. Some readers might argue that the piece in question is not a representative example, that it would be hardly applicable for more "serious" and challenging works. Yet, it remains one of the most memorable concerts of my career, and it was one of the reasons that pushed Listen and I to collaborate again, hoping to replicate it.

[74] When discussing the project with Listen ensemble, this aspect of replayability was immediately considered in interaction with practicality and budget of course. This led us to design *Game Theory* to be as modular as possible, especially dealing with so many different bricks in the multimedia continuum. We thought this approach — it remains a hypothesis, difficult to verify today — would enhance the portability of the project and its adaptability.

another coin in the machine, so to speak — potentially leading the ensemble to restart from the beginning, looking together for a better outcome[75].

[75] I already mentioned how I attempted to incorporate the learning phase as a structuring element in my last pieces 藏镜人—*puppetmaster* and *Hyperlinks - asset(s) failed to load properly.* In a similar way, my intuition for *Game Theory* was that replayability would give the occasion to musicians to acquire a higher level of fluency with the system — through rehearsals of course, but also in the course of the performance itself, multiple iterations or "games", and internally by revisiting specific levels. I also hoped that the musicians in the ensemble who were less familiar with improvisation would gradually gain more confidence in their ability to take spontaneous musical decisions, with or without a score in front of them. This was manifest in what we imagined could be a "typical" *Game Theory* evening, in terms of evolution of difficulty and complexity : 1st game as "tutorial", beginner level (medium duration) ; 2nd game, intermediary level (shorter duration) ; 3rd game, "expert" level (longer duration)…

Silent cacophony and a few broken mechanics

This leads us to a well-known question in game studies, namely the tension between *ludology* and *narratology*. We could summarized this as the dynamic between the game as interactive system — its mechanics or *gameplay*, as experienced actively by the player(s) — and the game as narrative or discourse, what is told to the player through those mechanics but also by other means such as text, visuals, sounds, etc. This can often turn into a situation of so-called "ludonarrative dissonance"[76] where both aspects seem to contradict each other. For *Game Theory*, this manifested by distinguishing two competing layers. On the one hand, everything that could be considered as "fake game" or staged elements, that could not be influenced by the audience : scripted events, illusions of choices, forced directionality, pre-recorded materials, etc., in other words, everything that would be pre-composed. On the other hand, what could be effectively considered as "game-like" — the generative score of course but also generative electronics, projected visuals, visuals on the smartphones, the network orchestrating the whole apparatus, all aspects that needed to *function* in parallel, and primarily the interactive channels open to the audience.

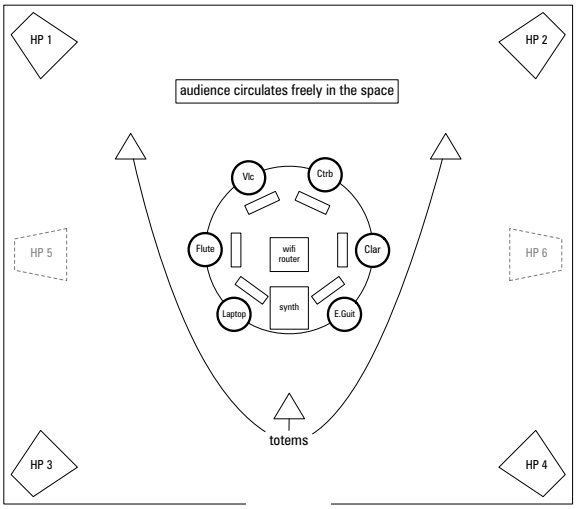
When I proposed this typology of project, I did not mean to pretend I would become myself a game designer. Despite my deep interest for these ideas, this is a very different profession for which I have not been trained. Over the last few years, I learned as much as a could about this topic, hoping at the very least to understand what I wanted to avoid. My main focus remained on expanding my expressive palette as a composer, especially in terms of form and, as I already mentioned, format or ways of *monstration*. As a composer of mixed music, more accustomed to control than full openness, I tend to lean towards the *narrative* axis to ensure that the performance would be efficient and interesting as a whole, artistically, and not sacrifice the music for the sake of the idea. Yet I had to find the right balance, with a sufficient amount of "gameplay", otherwise I could hardly justify the existence of such project. I will try to present some hypotheses we made for this layer, how we imagined the audience could interact with the apparatus and, probably among those, some reasons that led me to abandon the project, at least for the time being.

Early in the development of *Game Theory*, it was imagined that the audience would interact with

---

[76] A typical example would be a storyline that constantly reminds the player of the urgency of "saving the world", while in practice, the game encourages taking all kinds of detours and engaging in activities or quests non-essential to the main narrative. Some game creators and theorists have argued against this, placing more value on works where the mechanics themselves reinforce the discourse, and vice versa. Others believe that such inconsistency may hold expressive interest by momentarily pulling the player out of the immersive experience. In other words, ludonarrative dissonance can serve as a reminder that they are playing a video game... or attending a music performance.

the system through various interfaces on their own smartphones, which would evolve dynamically according to the musical and narrative context during the performance. In addition, projections on the walls would combine pure visuals with some informations — real or fictitious — about the system's behavior. The smartphone seemed like the most obvious and affordable solution at first, considering how ubiquitous those devices have become[77]. Another reason is that the audience would be encouraged to circulate freely around the performers, arranged in circle towards the center, and could keep interacting with the system from anywhere in the space. Later in the development, we considered to diversify the interactive layer, to allow visitors to interact without a smartphone. Different options were explored : placing several "totems" around the performance space, i.e. tablets mounted on stylized structures (mirroring the smartphone interfaces) or a constellation of smaller interactive objects and sensors, potentially based on Arduino or Raspberry Pi.



I cannot stress enough how *vague* my own ideas for audience interactions were throughout the process. My initial choice of using the smartphone was mostly conceptual and practical, but beyond a non-structured patchwork of "game-like" tropes I could think of, I had no idea how to actually implement them or how to connect them effectively. Yet one thing was clear for me from the beginning, is that I did *not* want this performance to entertain a fiction where the audience would become the composer or a part of the ensemble. Above all, I did not want the smartphone to become a *controller* in the instrumental

---

[77] When I first imagined this project in 2016-17, the smartphone was already omnipresent in our lives but did not seem as powerful a tool of mediation for the audience as it is today. Over the last few years, in every exhibition of painting, video art, installation I have attended — not even mentioning concerts — visitors are documenting the whole experience with their phone. To the point that the work now mostly exists through this fractalization process, via a multiplicity of artificial retinas and digital memory. It has become a full part of receiving and appreciating art as a cultural practice. This is why today I would probably consider this initial choice of the smartphone more carefully, and at least try to address these problematics more frontally in the work.
cf. Aja Romano, « TikTok has transformed the concert experience », in Vox.com, September 2023

sense[78], like the MIDI devices we use when improvising with a Max patch. If there was indeed a form of control by the audience, I wanted it to be decorrelated from the music as much as possible, and instead develop its own autonomy within the apparatus, without relying artificially on sound or music metaphors[79]. As a result, visitors would be practically unable to trigger sounds — to avoid the *machine-gun* effect and similar gimmicks — or modulate any music parameters, at least consciously.

Instead I was hoping to reinforce the audience *as* audience, to create an occasion for listeners to *resist* the music so to speak, apply some pressure back on it, and generate a form of pseudo-haptic feedback, helping to orient and transform their journey inside the work. I imagined this almost like an evaluation of the work as it was being made, where the audience's judgement could influence its (re)writing on the fly. In particular, this was inspired by a well-known series of experiments on listener's perception, conducted by John McAdams around Roger Reynolds' piece *The Angel of Death*[80]. At first, I imagined to give the audience a direct access to such qualitative, meta-parameter — "interest" vs. "boredom", perceived contrast, or my favorite, "temperature" — in the form of a cursor moving up or down along an arbitrary axis, similar to the ones used during McAdams' experiment. But I later realized it would be much more interesting, or at least less artificial, to deduce such information from all kinds of interactions with the game layer — insistence on a button, speed of reaction to a prompt, movement detection by accelerometer or gyroscope, etc., i.e. all measures of activity or effort obtained by means of "game mechanics", from each member in the audience simultaneously. As a consequence, the main

---

[78] Musicians and artists have been provided for decades with increasingly accessible tools to create so-called "interactive" digital media situations, such as MaxMSP, Supercollider, Processing, TouchDesigner, and recently even video game engines like Unity or Unreal. And yet in the music domain, I am rarely convinced by such productions — including what I can imagine myself, for that matter — in terms of interaction design towards the general audience.
A possible reason is that we are reluctant to deviate from what we consider the most natural solutions *as musicians*, where a patch generally adds up to some kind of "instrument". I believe we could benefit from setting aside this predominant organological approach, and experiment with different metaphors, affordances, user experiences, etc.

[79] I must confess that I was never really attracted to the genre of "music video games" myself, such as the popular *Guitar Hero* or *Taiko no Tatsujin*. This is probably why I wanted instead to try and inject video game ideas into music in the first place.

[80] I only heard about these years later, but the idea left a strong impression on me. When I first started developing *Game Theory* I contemplated a similar experiment, but looped on itself : where the judgement of the listeners would not only be analyzed, but influence in turn the music, then keep influencing the listeners experience, and so on.
« The experiment consisted of collecting continuous ratings on two scales: Familiarity and Emotional Force. For the Familiarity Rating (FR) scale, listeners were to continually estimate how familiar what they were currently hearing was to anything they had heard from the beginning of the piece on a scale from "Completely New" to "Very Familiar". For the Emotional Force (EF) scale, they were to continually rate the force of their emotional reaction to the piece at each moment on a scale from "Very Weak" to "Very Strong". »
cf. Shlomo Dubnov, Stephen McAdams and Roger Reynolds, « Structural and Affective Aspects of Music From Statistical Audio Signal Analysis » in *Journal of the American Society for Information Science and Technology*, Volume 57, Issue 11, 2006, and Roger Reynolds, « Epilog: Reflections on Psychological Testing with The Angel of Death » in *Music Perception*, Vol. 22, No. 2, 351–356, Winter 2004.

limitation I would need to establish is restraining immediate auditive or musical feedback as much as possible. The obvious reason is that the sum of individual feedbacks would soon saturate the space entirely, and make the situation completely unreadable. Instead, I would need to rely almost exclusively on *visual* feedback — from the same smartphone interfaces — and reserve auditory ones for the most meaningful, collective situations.

At this point, a clear distinction needed to be made between direct or quasi-instantaneous interactions, on the one hand, and slower, medium or long-term interactions, on the other hand. The first category — either triggers, switches or modulations of sorts — if they were to have a more immediate, perceptible impact, would not only affect the music, but the whole audio-visual apparatus. Consequently, they would help to highlight meso or macro-structural changes rather than merely trigger extremely local events.

To maintain a level of control, the system would need to exhibit a form of "resistance" towards the audience, for instance by limiting how frequently each audience member could interact. Rather than constantly soliciting everyone to tap on their screen, the system would try to keep these moments of direct action rare, and display passive visuals or less consequential interactive objects instead, thereby encouraging focus on the performance. For the same reasons, elements like timed animations or progress bars could also be used to measure time and anticipate formal shifts or the next opportunity to interact. I also imagined situations which would combine such timed elements with direct actions, for instance by suggesting the audience to press the screen for long enough, until it was completely filled with color. By calculating a different ramp time for everyone, depending on when they started to press, one could orchestrate (i.e. somehow manipulate) the audience into believing they all triggered a same, unique musical event…

Likewise, we considered establishing certain "good" or "bad" behaviors[81] as basic mechanics, both for practical, "game" reasons and narrative reasons. In particular this was our first solution to handle potential "trolls" in the audience, who would interact with their phone mostly to test the limits of the

---

[81] Although I have participated in programming several interactive installation projects for other artists, and also composed music for some fixed, non-generative installations (cf. *Parameter / Perimeter* with Pauline Shongov in 2019), it was only recently that I had the opportunity to create an interactive installation as a personal project : 湮未没—*pas encore rien*, created in collaboration with NI Shan. In this context we only had a few days to program the patch, but we did try to implement some "good behavior" mechanics.
The main point of interaction was a motorized, AI-based camera, able to rotate and follow visitors as they entered the room. If the visitors spent only a short amount of time with the installation, the system would simply display a *façade* layer so to speak, without much contrast, then return to "sleep" after they left. If they spent enough time and interacted a minimum, they would probably be able to experience a full cycle — with sound and images slightly different each time, yet composed sequentially. Finally, if the visitors stayed even longer, and behaved in certain ways — engaging with the AI camera — they would be rewarded for their patience and interest with a hidden section.

system, or even try to break it. According to the narrative, such abnormal behaviors or patterns could have been detected and either punished — losing control momentarily — or (closer to our hearts) rewarded with "Easter eggs" under certain conditions[82].

The original, most obvious mode of interaction I considered since the beginning, was asking the audience to reply to some questions or prompts, i.e. to make explicit choices between two or more options. This was directly inspired by *The Stanley Parable*'s narrative tree : "which door to take : left or right?", "should we move on now : yes or no?" etc. The audience would vote to decide when to start and finish the piece, when to introduce significant events, which section to explore next… only after a certain amount of votes had been collected, or a time limit had been reached. There, the audience interactions would be much less connected to the music, and more to the global narrative behind the performance. As a scenographic device, the system could either make a very sparse, local usage of such messages, or suddenly bombard the audience with alerts, absurd requests or taunting remarks about their compulsive phone usage… At the time the project was abandoned, only (rather naive) hypotheses had been made about this part, or the explicit narrative which could have structured the performance. Yet, after attending

[82] The idea of rewarding individual members of the audience for unpredictable behaviors was inspired after attending the immersive theater production *Sleep No More* in New York, 2019. There, my partner and I realized that imitating the rest of the audience — i.e. following performers wherever they went — had a very limited interest in the long run. On the other hand, abnormal behaviors — arriving last in the elevator, spending time observing details in the environment, or sitting alone in a room, motionless — were often rewarded with hidden bits of meaning or unexpected interactions. For *Game Theory*, we imagined those rewards to take different forms, musical or visual — either on one's smartphone or on the main projections — or a more material form such as a paper print of a generated score excerpt, or even a small 3D print representing their impact on the performance.

a public presentation of a project with slightly similar ambitions to *Game Theory* in 2021, it became obvious for us that the entire performance could *not* rely only on such devices to reach our artistic goals[83].

For the specific category of "direct" interactions, the very few strictly musical ones that I still considered using, albeit with considerable restraint, would be :

• on/off switches for musicians : Certain audience members would momentarily be given the power to "censor" the score of one or more musicians, either by making it unreadable, pixelated or by turning it off entirely. This would potentially lead to entirely silent moments in the performance, used as occasions to trigger a narrative and/or musical shift.

• drawing shapes : At certain intervals, certain players would be chosen randomly by the system to draw an arbitrary profile or shape on their screen. This would be almost instantly transmitted to the screen of one musician among the ensemble, on top of the regular generative score. The performer would have a limited time window to make sense out of it — musically, sonically, gesturally… — in the current context being developed. Potentially, a few simultaneous drawings could be synchronized, resulting in collective gestures within the ensemble.

• firefly swarm : Finally, an idea I imagined almost exclusively for the beginning of the performance — i.e. the hub, or the initial installation-like situation when entering the space — involved

---

[83] During the 2021 edition of Ircam's *ManiFeste* festival, Rafaël Carosi and I attended a performance which shared a lot of similarities with our own project : *Music of Choices* by Alexandros Markeas. For about an hour, the composer improvised alone on the piano and keyboards, along with live electronics. Audience members were invited to connect to a web application with their phone, in order to influence the performance.
After hearing only this premiere many years ago, I would be completely incapable to judge of the music value of the proposition. Some technical hiccups did happened, but otherwise everything seemed to work as planned. On the other hand, I would notice a few elements in the interactive apparatus which I feel were debatable.
First, a single mode of interaction was proposed, in the form of an almost uninterrupted flow of questions, to be answered with two or more possible choices. Results were projected in real time on a screen next to the musician, visible to everyone. These choices seemed to have only a limited impact on the performance, on the few key moments when the composer looked at the results on the screen. Since I was constantly solicited by my phone, or looking at the main screen, it is possible that I misunderstood the situation.
Secondly, instead of remaining secret until everyone replied, poll results were constantly updated in real time, as the choices were being taken, which had for effect to systematically influence the group to change their mind and follow the majority. This, I believe, could have been very useful in the beginning of the piece, as a tutorial so to speak, to feel one's impact on the collective. But as a general solution, it turned the whole concert into a comical "arm-wrestling" competition, between completely arbitrary groups within the audience. The process kept restarting from zero with every new question, and without many occasions to rest.
Finally, the majority of choices were about the music performance itself — which instrument to use, which character to play with, should the sound be louder or softer… — before drifting quite artificially into more comical or metaphysical topics. As a result, the interactive system felt more like a "gadget", distracting listeners from the music itself.
Yet, attending this performance was for us extremely valuable, since we were never able to take *Game Theory* in front of a live audience ourselves. Even with clearly different artistic premisses, this was the occasion to "test" ideas that were very similar to some of our own, to see how they could behave in time with a large audience, what could be improved, and what should be avoided. We also noticed some conceptual and technical issues which could have been prevented by solutions we already imagined.

using simple taps from the audience on their screen to trigger soft and short instrumental or electronic sounds, in any case some very delimited, pointillistic objects. My goal was to create a tutorial-like situation where the audience could realize not only their impact on the system, but also the influence from other audience members. But this was also intended as an opportunity to encourage responsibility : with every new tap, the volume would considerably decrease, mitigating any attempt to invade the space with unnecessary sounds.

Slower modes of interaction, on the other hand, would allow to focus on continuous and broader *tendencies* in behaviors among the audience, as a metaphor for tensions arising within the collective, and potentially a way to control the accumulation of musical, electronic and visual layers. They could also influence the piece in the medium-long term : another idea I was particularly attached to was to let the audience place "memory markers" during the performance. These would be used to identify specific time intervals in the main generative score, which the audience would have considered as memorable : passages which would later have a higher probability to be repeated, or transformed, thus allowing to establish a collective memory within the performance.

I feel that, among all the directions I have considered for smartphone interactions, the ones that stuck with me the most were all collective in nature. By that I mean either emerging collective behaviors, or situations where audience members would be somehow invited to collaborate more explicitly — actions requiring a certain effort and precision from each individual to fulfill a certain goal. The latter aspect was inspired in particular by a couple of remarkable, collective control-oriented social experiments I came across during my research. The first, more ancient, was the so-called "Pong experiment"[84] organized by Loren Carpenter (co-founder of Pixar) at the SIGGRAPH conference in 1991. There, a few hundred audience members were invited to play the original Pong game collaboratively on a large screen, by ways of colored paddles detected by a camera. The second and more recent example is "Twitch Plays Pokémon"[85], started originally in February 2014 and often re-organized, where more than a million

---

[84] I discovered this rather spectacular experiment in Adam Curtis' documentary series for the BBC, *All watched over by machines of loving grace* (2011), named after Richard Brautigan's poem. The latter was part of a list of potential "Easter eggs" expected to be disseminated in the performance. cf. https://vimeo.com/78043173 (accessed July 2024)

[85] During this experiment, now largely documented on the internet, the high complexity of navigating certain parts of the game collectively made necessary to implement different voting mechanics to overcome different obstacles. This, along with John Cage's *Les chants de Maldoror pulvérisés par l'assistance même* — another vote-based performance piece — and John Zorn's *Cobra* — where the so-called *guerilla mode* could lead musicians to rebel against their leader — made us realize very early that we needed to compose with group dynamics in mind. In particular, we imagined a progression along the continuum between "authoritarian" and "democratic" poles in the course of the performance. cf. Lindsey Marley-Vincent, « The Politics of Pokémon. Socialized Gaming, Religious Themes and the Construction of Communal Narratives », in *Religion in Digital Games Reloaded. Immersion into the Field*, *Online - Heidelberg Journal of Religions on the Internet*, Vol. 7, 2015

participants managed to collaborate to control (by votes) and eventually complete the game in about 16 days. The first prototype I made along those lines was a simple Max patch called the “Clap-o-meter”, aimed at simulating a cloud of simple inputs from the audience with various degrees of “impatience”. From there, I calculated the “density” of this cloud by counting how many inputs would be received within different time scales, either very small (~100 milliseconds) or extremely long (30 to 60 seconds) intervals. This would lead to very different shapes, either sudden changes in density — potentially leading to immediate triggers of local events, like the “firefly swarm” scenario I mentioned — or much slower evolutions for parameters such as the “temperature”.

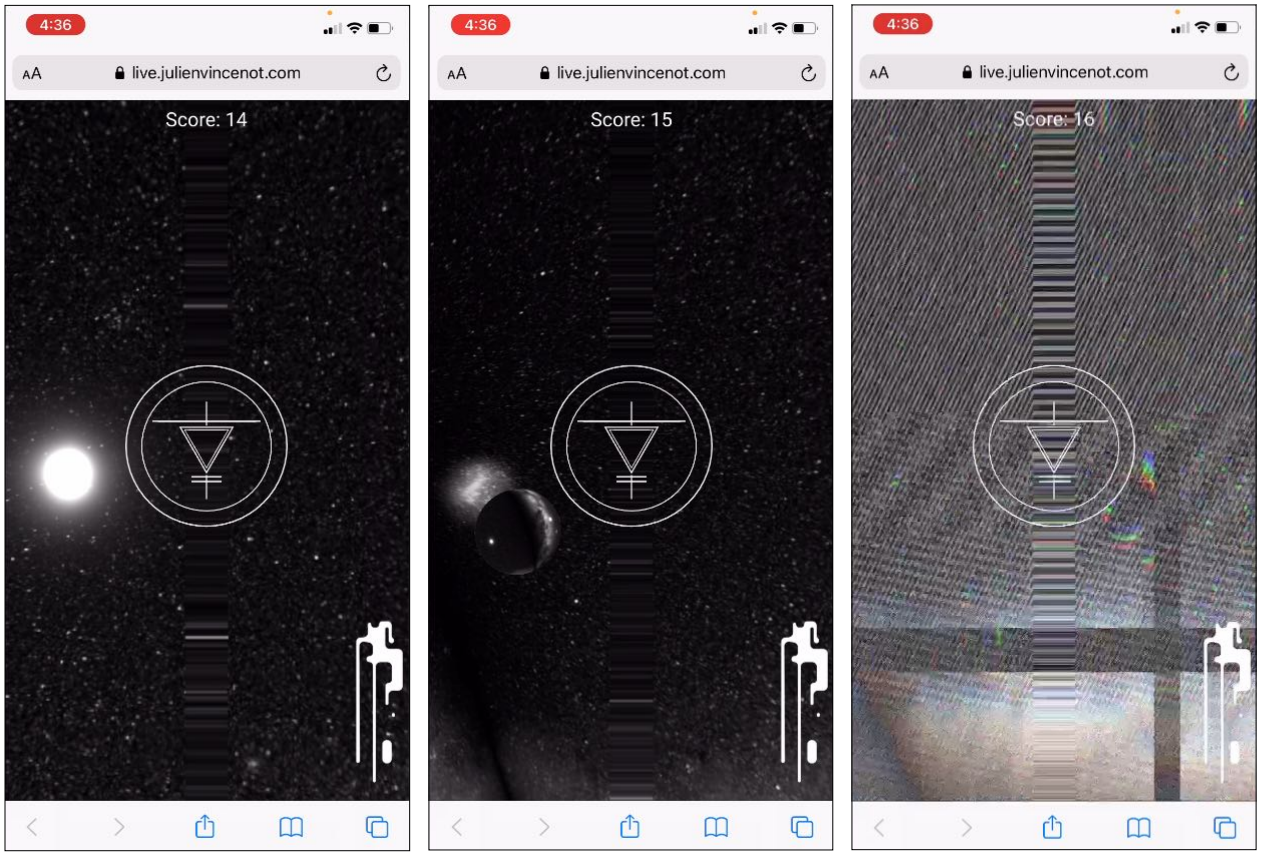
Most of the ideas I have listed in this section were not exactly game mechanics per se, but more variations around the same idea : pushing buttons. This was one of the main pitfalls in our project’s discussions, one we found particularly unsatisfying and difficult to overcome. The reason for this could be twofold. On the one hand, in an interactive context, pressing buttons can have an infinity of effects and meanings, especially within a complex narrative. I think it goes without saying that the majority of video games rely solely on pressing buttons… but we preferred to avoid imitating the obvious “gamepad” archetype, which is more suited to control an avatar in a figurative space than more abstract scenarios. On the other hand, imagining more complex modes of interaction on a smartphone screen, even those that might seem the most “natural” to our fellow smartphone users, would mechanically imply a much more complex and potentially costly execution.

And yet, it is in this direction that I found probably the most promising solutions. Beyond tapping and sliding fingers on the screen, smartphones come pre-bundled with tools allowing to work with *augmented reality* (AR) : a screen, one or more cameras, and sensors like accelerometers and gyroscopes. On paper, we believed exploring this rather new mode of interaction — i.e. overlaying generative visuals

and interfaces on top of the camera feed of a smartphone — could have a lot of expressive potential for the concert situation[86]. Unfortunately, when we were exploring this topic in 2020, it did not seem possible to reach reliable AR on a low-budget, cross-platform application — especially for a live performance happening partially in the dark. For this reason, in the last year the project was still active, we mostly experimented with an idea of "pseudo-AR" : generating 3D scenes (and potential games) observed from a fixed point in the virtual space, which could be rotated freely in 360 degrees — by changing the device's orientation.

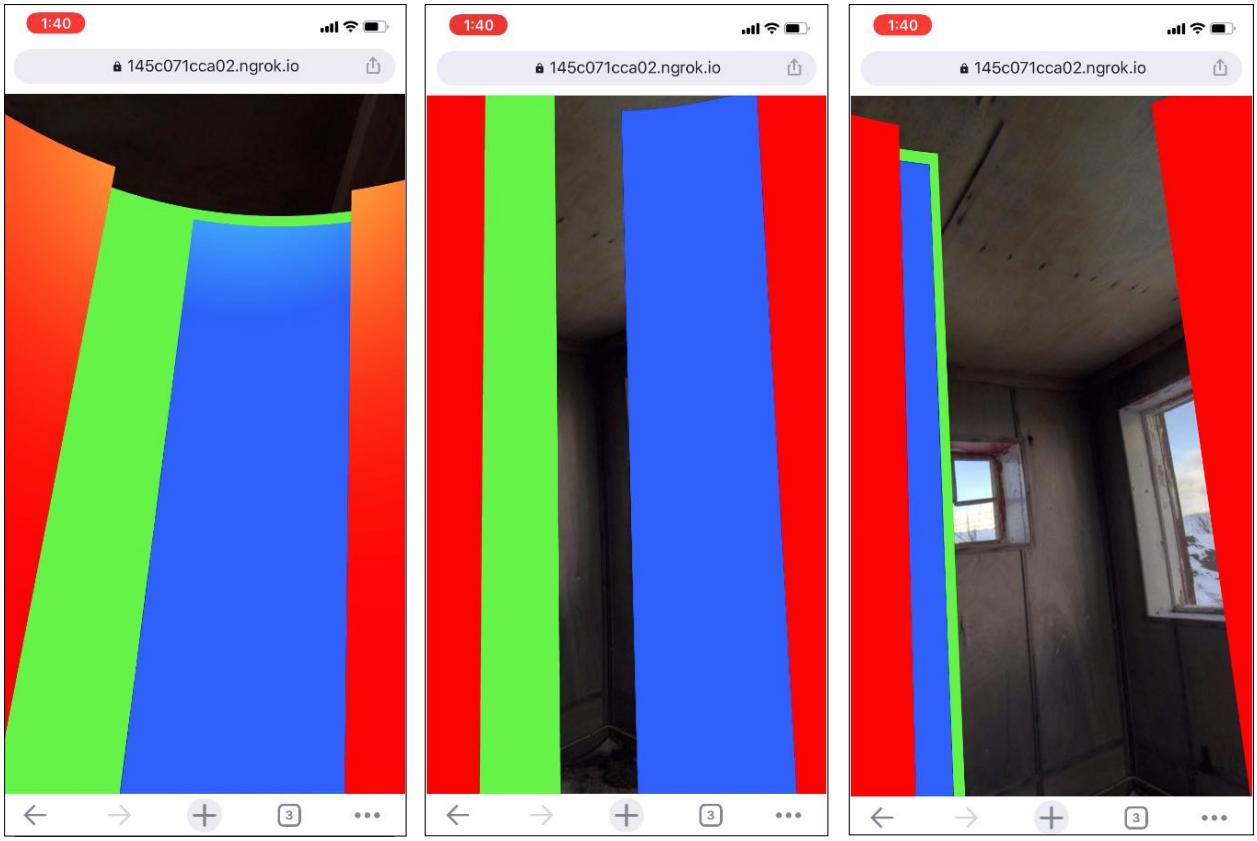
This idea was developed in two directions, or "mini-games" :

• *Moon hunting in the dark* : A single "moon" would constantly gravitate around the player in the virtual space. By rotating the phone in every direction, one could sometimes find the moon and "capture" it with the central visor on the screen — triggering a new event — before it reappeared elsewhere, randomly. This first little game prototype was implemented during a residency in Toulouse's Studio Éole (summer 2021), and tested with a very small audience.



[86] I believe it could be even more impactful for a mixed music performance — in a sense, already a form of "musical AR" — to incorporate smartphone-based AR, rather than relying solely on projected visuals around the audience. This could support the "fictional layer" of electronic music atop the live performance, with visual elements emanating directly from the musicians themselves in the performance space. For a similar purpose, we also considered a more direct solution, of projecting such visuals on semi-transparent panels around the ensemble.

• *Gyroscopic maze* (or “Finding *the* point of view”) : This was imagined as a “multiplayer” experience, in which every audience member was surrounded by multiple layers of circular walls — one rotating wall for each player — seen by everyone from the center. Each player’s wall would have a large open dent, right in the middle of their field of vision. The goal of the game would be for everyone to rotate and find the one, common opening, finding the “exit” together so to speak. From there, it would become easy to calculate the distance between the current state of the game and this expected “good behavior” (all the angles aligned in the same direction), and thus transpose this distance by a (musically) perceptible increase in tension towards this common goal. A partial prototype of this game was realized in 2020, but the multiplayer part of the system remained unfinished[87].



[87] This aspect of multiplayer and network management was and remains the most difficult for me technically, far beyond my learning capabilities. Since the last time I attempted to tackle the subject in 2021, several companies have created libraries to facilitate this aspect for aspiring game designers. I will turn to these resources if I have another opportunity to work on such issues.

Yet in both examples, a significant problem remained : this solution was interesting for a solo experience or even with a small audience, but a larger audience would bring the risk of players potentially colliding with each other…[88]

This series of reflections and hypotheses collided with the harsh reality of its technical implementation. Despite our repeated attempts, we were never able to find someone capable of assisting us with this part of the project, especially on a rather limited budget. Beyond pure technical skills — to design and program such interfaces or game experiences for smartphones — we also needed someone with an artistic sensibility, who would be curious and flexible enough to join and collaborate on such an unusual project. When the pandemic hit, the March 2020 premiere was immediately cancelled and the original team was disbanded : only remained Rafaël Carosi and myself, trying to reorganize the whole project for a future opportunity — which never really came. These uncertain times suddenly opened a window to learn and experiment, and I resolved to take this extra workload on myself, at least temporarily. But developing video game-like experiences, beyond pure code, immediately revealed one of my biggest shortcomings : I almost didn't know anything about digital graphics, but also my visual skills and sensibility in general are absolutely appalling. To be able to restart the project, we needed to collaborate with a visual artist familiar with digital media, technology and generativity. Luckily, this is when our small team was joined by multimedia artist John (Iannis) Bardakos, who brought his strong visual personality and theoretical insights to the project. For more than a year, the three of us would discuss this new version of the project regularly, yet by the distance. As Rafaël was looking for fundings and venues in this new, difficult world, John would elaborate *Game Theory*'s visual identity, and I would attempt to turn it into functional programs.

I first explored possibilities of the video game engine *Unity* — freely accessible and largely documented — but in 2020 it was still far to be the easiest route for mobile applications. We needed something much lighter to start making prototypes, and we opted for the library *Three.js* which allows to develop interactive 3D applications directly in the web browser, making it de facto portable to any modern smartphone. Based on Javascript, programming with this new tool was rather accessible to me — if not perfectly natural — and very easy to connect in real time with MaxMSP. I trained myself with the language during the fall 2020, and after half a year of hiatus, I travelled back to France in the spring. I resumed my experiments on Three.js — and learned React.js to orchestrate them dynamically — as the

[88] A similar approach was chosen by Nadya Lev and Star St. Germain for their mixed reality narrative game, *HoloVista*, released in the end of 2020. To avoid the risk of nausea and collision with the environment, they opted to combine the "pseudo-AR" idea with more classical controls, allowing the rotate the virtual space both by device orientation and by sliding a finger on the screen.

trio prepared to gather in person for our first (and last) residency at Studio Éole, in the summer 2021.

There I did manage to produce a small, partially functional prototype, that could be ran on multiple smartphones and interact with the audiovisual apparatus. But after several months of learning and experimenting with these tools, I ran again into the obvious fact that I am *not* a game designer or programmer. Even if I was able to solve a very large number of technical difficulties along the way, I never became fluent enough to create a game system that could be flexible and robust enough to work alongside my own musical systems. And in spite of my constant study of the topic, my game design imagination remained very modest. To be able to turn a project like *Game Theory* into a convincing reality, would require either an opportunity to collaborate with specialists of game design, or to practice myself on smaller game projects for as long as necessary.

Keeping the score in a liquid state

Before closing this section, I would like to briefly discuss the generative score itself, as a technical object in the context of *Game Theory*. To be perfectly transparent, I don't think I would be able to describe it in much detail musically, beyond the formal organization in two graph layers I already explained. I composed the main patch almost 5 years ago, and during more than 3 years, some successive updates in MaxMSP and the *bach* package made the score entirely inaccessible — the patch would crash systematically on opening, and became impossible to fix[89], to the point that I have lost faith in the project partially. This score remains a nebulous object : apart from a short residency where I was able to test it with musicians, I never had the opportunity to step back and fully evaluate it in a performance context. It has remained, as one could say, *lettre morte*. Yet, I will try to describe some of the ideas and properties I can remember, and in particular, what such a score represents for me today.

First of all, in a context where the audience could circulate freely around performers, I imagined the live-generative score would need to acquire a certain intensity or "presence". It would not be projected on the walls to be seen at all times — which I always found to be a somewhat superficial solution — but would be an integral part of the staging nonetheless. This would be emphasized by developing the score as a more "stylized" visual object — white-grey inscriptions on a black background in particular, seemed an interesting solution to integrate with the lighting design. Standard musical notations would be combined with more discreet, visual elements only addressed to the audience, to reinforce the narrative. As a repurposed object, I wanted the score to maintain a certain form of ambiguity : the fiction that an "authentic" composition process was actually taking place, almost like a parody of this often-mystified work of the composer. Ironically, the patch running in the background to generate this behavior, this fiction, did result from a form of musical rationalization — i.e. CAC-like formalization — or at least an attempt to represent some reducible parts of the composition process, perhaps to highlight what would still be missing in the end. Just "as if" a score existed…

Not so long ago, a composer friend told me : « What I am doing right now doesn't really require notation, it doesn't really look like [what] a composer [would do], haha… » This made me think about

---

[89] Only very recently in the fall 2023, and with the help of Andrea Agostini, I was finally able to re-open the generative score patch on a more modern machine with Apple silicon chip. On this occasion, I discovered that the score generation and scrolling was a much smoother process on the new platform than it was on the Intel machine with which I initially composed the patch. Such a generative score patch, beyond the computing power required to calculate the score structure and music content — mostly by manipulating complex list representations, similar to Lisp — is also extremely demanding in terms of graphics power, which the Intel computer couldn't really provide in parallel. In other words, the powerful and expensive machine I purchased especially for *Game Theory* in the fall 2019 made the project impossible to run correctly in the first place.

what "looking like a composer" could mean today, the different clichés that this idea might carry, especially when writing a score. I couldn't help but drawing a parallel with my recent projects involving generative scores : entertaining the illusion of a score « that looks like a score », behaves like a score, but is (perhaps) *not* a score… As a matter of fact, many of the results I would obtain from the main generative patch, programmed in 2019, contained elements or ways of notating which were quite remote from what I would normally use in my mixed music. Instead they could be seen as a wink to some forms of graphical mannerism I have observed in contemporary scores over the years — and sometimes even imitated myself in my earliest compositions[90].

When I programmed the very first prototype for solo flute, *Game Theory #0 - En Rafale*, in the fall 2018 — the score worked with all the music parameters available in *bach* notation objects, including pitches which were generated quite naively, with probabilities on selected harmonic fields. Claire Chase, who was leading the workshop, proposed that removing pitches from the equation would help the flutists avoid sight-reading the score too literally, like they were taught, and instead focus on what the score suggested in terms of gestures and morphologies. I agreed this was a very important point, and I immediately disabled staff lines and accidentals in the final output. I left the simple pitch system in place, working in the background, only to determine overall registers. I kept this idea in the next patch I composed a year later for Listen ensemble, but this time I didn't want to sacrifice harmonic control entirely. For this reason, I elaborated a system which would analyze the electronic part — partly generative and fixed, partly improvised — in real time in order to determine the harmonic context for the musicians to work with[91]. This whole set of allowed pitches or frequencies would be partially re-orchestrated and distributed to each instrument according to their respective range, and displayed right next to the reading cursor.

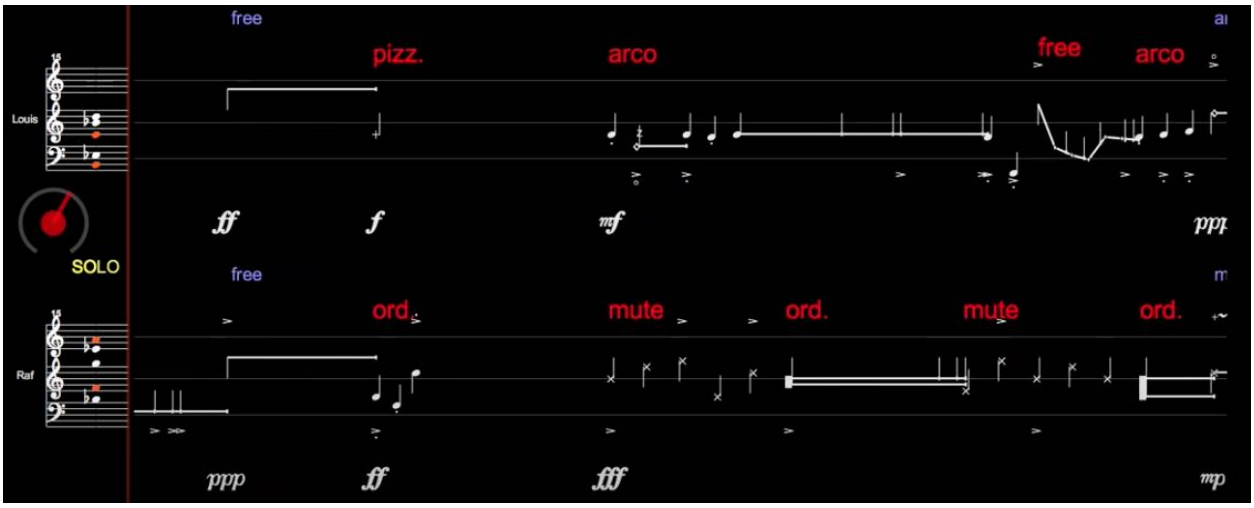
At the core of the generative system, I established a hierarchical database to determine what the instrumentalists would be able to play, inspired by object-oriented programming. Each instrument would have a set of possible playing techniques, defined by specific notation elements (note heads, articulations, additional glyphs, etc.) and in turn each of those would have a limited range for pitches, durations and rhythmic density, dynamics, and other constraints for how to chain them. This system would allow to create new instruments quite easily, but also to invoke any combination of instruments I would need for

---

[90] Retrospectively, I tend to believe the behavior of the score I programmed for *Game Theory* was strongly inspired by the score of *all the memories are rigged*, a piece for 9 musicians (without electronics) I wrote in the spring 2019 for a residency with WasteLAnd ensemble but never completed.

[91] I found this idea interesting because it took the opposite approach to what is typically the "norm" since spectral music, i.e. generating sound synthesis that closely follows the spectral content of an ensemble or orchestral performance. I also wanted to be able to freely explore the whole continuum between spectral fusion and fission.

future iterations of *Game Theory*[92]. In addition to this global instrumentarium, the local graph for each "level" of the piece would be associated with a list of more specific musical properties, which would further restrain what the chosen instruments could actually do in a given context.



Ultimately, what I was the most dissatisfied with in this system was centralizing everything in a single notation object (*bach.roll*), but also trying to calculate all the music with the same paradigm in the background. My initial goal with this score was to push the limits of what I could express with *bach* objects, in terms of music language and decision-making. I wanted to reach something very general, a motor that (admittedly) was not very smart musically — I couldn't use constraints generation for instance — but still, an automaton able to generate very different, plausible music situations (at least visually) which could trigger a meaningful musical response from the musicians. In spite of some promising results, this approach inherently limited the system's outputs, particularly in terms of control and degree of detail, especially when trying to achieve a very specific result I would imagine[93].

---

[92] Right after the premiere of *Game Theory #0 - En Rafale*, it became clear that the next patch would need to be as modular as possible, to avoid the inherent complexity of reprogramming the entire system from scratch for each new instrument. The patch I started in the fall 2019 would allow to accommodate any set of instruments, from a piece for 5 flutes and 7 guitars to a more diverse ensemble, depending on the opportunity. Whichever instrumentation was chosen, the whole patch would be able to adapt the overall musical behavior accordingly.
This proved to be a great help during a residency with Listen ensemble in December 2019, when we tested the generative score for the first time. Originally, the project was planned for five musicians: flute, clarinet, cello, double bass, and electric guitar—not counting myself on the live computer. The cellist planned to participate with us had to drop out of the project on very short notice, and we didn't have time to organize a replacement. Luckily, the patch adapted to this change without much trouble.

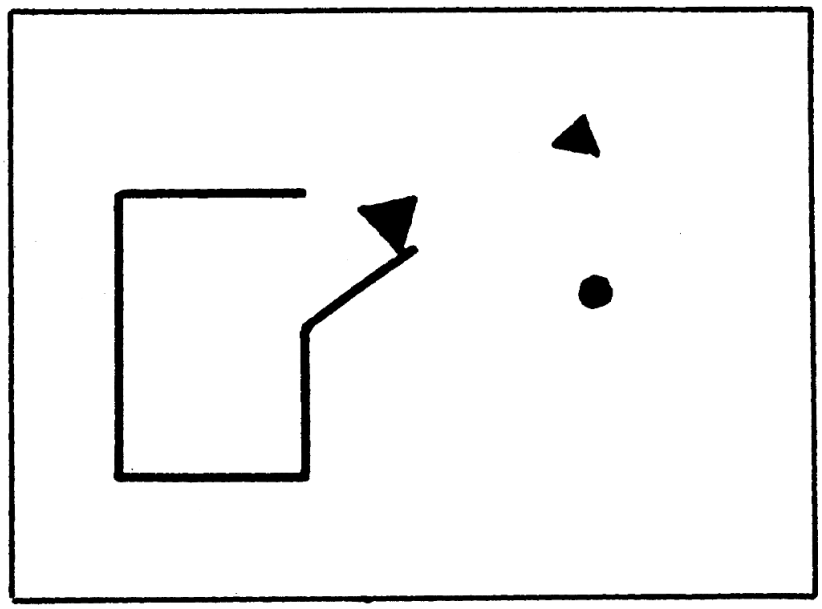
[93] This is why in my most recent, partly generative mixed music pieces like 藏镜人—*puppetmaster* and *Hyperlinks,* I tried to develop as many different approaches to score generation as I could — the same way I would work "normally" with CAC before, creating different sketches or patches for different sections. After this first experimental phase, I would try to combine them as seamlessly as possible in a concert patch — often by overlapping display of scores coming from different paradigms for different musicians to play at the same time.

On the side of the more *plausible* elements of music notation handled by *bach*, I imagined adding an explicit "gauge" referring to the parameter of "temperature" I mentioned before — possibly reinforced by changing the background color of the score along with it. This was for me a way to densify the score as visual, narrative device, but also to open a direct channel from the audience (the "game layer") to the musicians' screens. This gauge served for me as a supplement of information that the score could not provide : a sort of non-verbal (yet polysemic), dynamic "expression marking", that the musicians could (re-)interpret as they wished in any musical context, most likely to suggest the perceived amount of tension or agitation, instability, inharmonicity, roughness, friction, etc.

If I continue to spin this metaphor of temperature one last time, I could say that my goal was to work with the state of matter of the score itself so to speak, turning it from solid to liquid, and trying to maintain it that way as long as possible. Unfortunately, I feel that the generative score I produced for *Game Theory* always felt more "viscous" or "muddy" than liquid… The problem I was battling against, and which I never managed to fix entirely, was a form of *inertia*, much like trying to pilot a heavy ship on the sea. Whenever the temperature changed, it would often take 7 to 15 seconds to effectively *stir* the algorithm in the new direction, and really start observing changes in the score. This was partly because of the inherent complexity of the patch itself, which I started to compose almost like an "ecosystemic", self-organizing piece, with some metaphors possibly inspired by the music of Agostino Di Scipio or Phivos-Angelos Kollias for instance. This resulted in countless layers of feedback between different parts of the patch depending on each other, which significantly increased the overall computational load, and also made my life much more difficult whenever I tried to debug it. The perceived inertia was also a side-effect of the solution I chose initially to help the musicians anticipate the music better. I always needed to prepare some "extra music" in advance — coming from the right of the screen, to the reading cursor on the left — thus creating a large perceptive gap between any interaction and the resulting score. Instead, I would have needed more diversity between slower influences — acting on the more complex parts of the score — and more direct, perceptible reactions to the audience's behaviors — for instance with isolated gestures.

On top of this main musical "feed" and the temperature gauge, I imagined — but didn't get to fully implement — a constellation of non-obviously musical elements to enrich the generative score. I hoped these could provide more diverse and granular ways to convey musical ideas to the instrumentalists on the spot. Among those, I already mentioned the progress bars, narrative pop-ups, animated GIFs, but also all kinds of blinking icons, glitches and other real time processes applied to the score itself as a visual layer (or Jitter matrix). I also made some experiments on the side with "animated" scores — inspired in particular by the short film used by Simmel and Heider for their famous experiment on apparent behavior

of inanimate objects[94]. This last option in particular was interesting as a way to counteract the "viscosity" of the generated score, by making notations more lively or unstable, visibly reactive. All those ideas were part of the toolbox I developed for the project — hoping to resort to them, eventually, during a creation residency — but I never got to effectively combine them in a final system.



Simmel and Heider's experiment with animated geometric figures (1944)

Now that I have addressed all the main aspects of this project, from the non-linear form to the score, and including the layer of interaction with the audience, I want to conclude on my main intentions for *Game Theory*, and also discuss an element that I have kept somehow elusive in the background, namely the electronics parts.

The biggest problem for me when developing *Game Theory* was to keep the musicians in a state of constant discovery of the music. I did not want to rely on too many pre-determined bricks, which seemed like taking the easy way out. This was a constraint I needed to give myself, otherwise I don't think I would have remained so interested in the project in the first place. This was both a technical and compositional challenge : trying to imagine a piece that would be interesting — by standards similar to mixed music pieces I composed before — and still *feasible* with constraints of a score that was discovered on the fly. Generally, the system could be seen as an overly precise invitation to improvise as freely as possible. This sort of *in-between* approach required giving as much attention to written details as to listening and, of course, choosing when would be more meaningful to remain silent.

---

[94] cf. Simmel, Marianne and Heider, Fritz, « An experimental study in apparent behavior », in *The American Journal of Psychology*, Volume 57, 1944

Obviously this complexity implied a substantial amount of preparatory work with the musicians themselves, ensuring that everyone had as clear an understanding as possible of what the system could do, even before the audience intervened. During the first residency testing the system with the musicians of Listen ensemble in 2019, I was impressed by their receptiveness to the project's concept, and how quickly they started to appropriate the score. They all spontaneously grasped that the state of "stress" and concentration induced by the constant bombarding of musical data could lead them — after some initial adjustment — to develop highly expressive situations. I remember they produced during that week a great diversity of outputs, strong polyphonic events, clear moments of contrast and directionality while improvising with the — then somewhat *clunky* — generative score. The playfulness in their approach, engaging with the patch's still modest notational ramblings, was particularly touching. For this reason, I will keep trying to explore this direction as often as I can.

The second aspect of this project that would constitute a much bigger technical and compositional challenge was the electronics part. That is, creating a system where pre-recorded and generative sounds alike would follow the structure I elaborated for the score, in a way that would be as meaningful as possible. The most difficult for me was trying to find a compromise between the level of control I usually set for my mixed music — mostly studio-based — and the more unpredictable, fluctuating quality I could work with for instance when improvising in real time with *Unmapped*. Unlike the generative score, which now seems much easier to formalize by comparison, programming a system that could adapt to any situation and orchestrate a flow of events that would sound consistent and natural throughout felt like a daunting task. For this purpose, I needed to elaborate tools which would allow to generate flexible electronics situation by a concatenation of elementary bricks, based on pre-recorded sounds or synthesis and their transformations. Those bricks would be controlled individually not by myself of course, but by some kind of overarching formal grammar. I made several prototypes in that vein, where I would first generate a preliminary score (following the main instrumental score) — where each symbol was associated with certain keywords and descriptors of those elementary bricks in pseudo-"natural language" — then use this score to generate a new instance of each of them, on the fly. I developed these two systems side-by-side — generative score and generative electronics — to the point where it felt I had all the tools I needed to execute the idea before the premiere in March 2020. But even if it had effectively taken place, I don't know for sure that I would have been able to do so.

This brings me back to some questions I already touched upon partially in the last two chapters. My initial impulse for this project was figuring out *how* to write interactive and generative, non-linear music for instruments and electronics. I think I provided some possible answers to this question, but even

with the arsenal of conceptual and technical tools I developed to do so, I still feel that my imagination remained insufficient, too limited to realize it. Technically speaking, executing a single idea for the project was extremely time-consuming — 4 or 5 months for the first version of the generative score, or for a single prototype of the "game layer" — and yet I often ended up with systems that were not as flexible as I had envisioned for the project. As soon as they were completed, they almost instantly felt like fossilized. I couldn't move quickly enough within them, or between different elements — lacking the flexibility and agility required to keep the project in motion.

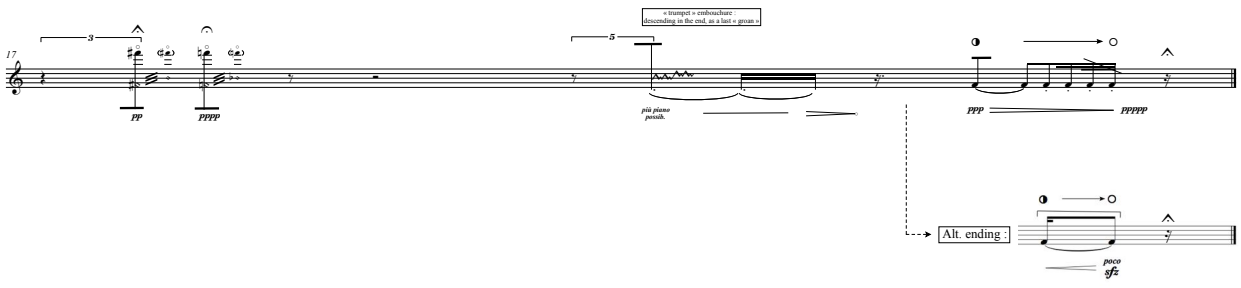
On paper, *Game Theory* remained a very exciting project for me all along — even today. But when in front of the blank page — or the "blank patch", the weight of uncertainty brought by the very nature of such project made it very difficult to reach something satisfying. The already intense struggle I had when making decisions for "linear" music has extended into the non-linear realm : especially when dealing with the electronics part, I felt paralyzed by the myriad of possibilities, where each new choice would lead to exponentially more numerous decisions to take, and materials to produce accordingly. Composing a piece of mixed music in the DAW would become more manageable as I continually listened and jumped between time scales, progressively sculpting the whole situation until I got a clear picture. But in the case of non-linear generative composition, as considered here, achieving this overall perception became practically unattainable, always resulting in a fragmented and incomplete experience.

In that situation, it became increasingly complicated to imagine myself in the position of the audience. And because this problem of perception and reception of the work remained, I kept being torn between two extremes : on the one hand, trying to offer a different, rather unpredictable experience to the listeners, and on the other hand, trying to "keep them with me" so to speak, in terms of accessibility, intelligibility, attention, etc., all the while being solicited by the smartphones and projected visuals. In spite of my sincere desire to extend the limits of how musical systems, materials, and situations can be explored by musicians and listeners, I need to acknowledge that some internal tension continues to pull me back towards more conventional narratives, rooted in linear or teleological time. After the last 5 or 6 years of reflections and experimentations in this direction, I am still not sure how to solve this contradiction, which is clearly a stumbling block in my research. Perhaps this is not obvious for every reader, but even video game designers must choose where they stand between linearity and non-linearity, "handcrafted" design and generative (or procedural) design, but in any case, they cannot do so at the expense of readability and consistency. Both for technical and aesthetic reasons, my views on this matter have evolved in the last years. Rather than the fully open work I hoped for initially, which may involve multiple beginnings, paths, and endings, combined with high granularity and unpredictability at all scales, I shifted my interest more and more towards a compromise or "semi-open" situation : relatively enclosed

space, with clear landmarks and directional structure, ordered (if not permutable) sections injected locally with various degrees of indeterminacy. This would seem like a more manageable horizon, something I could reasonably aim for, and gradually progress toward, over the course of a dozen projects perhaps…

Yet since the last couple years, I cannot help but wonder about the technological timing : what if *Game Theory* came slightly too early to be feasible, and only needed to wait a couple years, or just a few more months… for the technological framework to be ripe for it? If I think back to early works of non-linear music or game pieces I have mentioned, it seemed they all became imaginable because they appeared in a *technical milieu* where the production of musical materials was partially automatized via certain techniques — analog (on paper) or digital — potentially combined with the improvisational potential of the performers. We could even mention the so-called "musical dice games" (some examples attributed to Mozart) which were not possible without the inherent generative potential of tonal music itself. Most likely, a project of the dimensions imagined for *Game Theory* could not be "handwritten", even with dozens of composers on the assembly line, relying heavily on the whole spectrum of analog and digital proliferation techniques imagined in the XXth century… Perhaps truly interactive, generative, non-linear works become possible only by reaching a certain degree of abundance of materials, a tight network of connections ensuring consistency, causality and necessity between them, and of course… some robust decision-making devices, in place of my own indecisions.

This somehow takes me back to an old memory, with my friend the composer Carlo Ciceri, who sadly passed away in 2022. Carlo was one of the first who encouraged me to present my music in public, and I really looked up to him. At the end of the score for my short solo flute piece *Ascidiacea* (2010), I originally left the last gesture undecided between two options, whether a more inconclusive gesture, or a more decisive one. For Carlo, I *had* to choose the first one : the second option was too cliché anyway, and I guess branching out like this, right before the last sound, seemed like a lack of conviction. And it was true, at the time none of my ideas were fixed, I doubted about everything I did. I still do. But yet for me, it was also a way to give the piece some depth, some unpredictability, an option for the performer only to choose depending on "how the piece went" (to their ears), an embedded judgement I couldn't take. An X — a placeholder — or a last twist before the exit : which of these two doors takes us out? How to end…?



After CAC?

As I am writing this text around the summer of 2024, I cannot conclude without touching briefly on a subject I have mostly avoided until now — namely, the machine learning (or *deep learning*) based artificial intelligence techniques which have spectacularly irrupted in our lives over the last few years. During my time in the Conservatory of Montbéliard, I was lucky to be exposed to some rudimentary examples of these techniques by musician-researcher Frédéric Voisin, who was among the first to experiment with artificial neural networks (ANNs) in the context of music creation[95], at a time where the field of computer music didn't seem to take those very seriously. It was only around 2012 that a sudden increase of interest (and investment) led to the current AI boom. For a reminder, the decisive match between Lee Sedol and *AlphaGo* took place in March 2016 ; large language models like ChatGPT and prompt-based models for image generation like Stable Diffusion or Midjourney only came to widespread awareness in 2022-23. We were all caught off guard by the speed of this new arms race, and obviously any discussion about specific tools, in this context, would instantly become obsolete. This is clearly not the place for me to try and develop a definitive opinion on a topic I do not fully master. And I will not pretend to pick sides in the profound conflicts that these technological leaps keep triggering in our societies, in particular among artists, largely disoriented in the face of emerging AI art practices and the whims of multi-billion dollar entities we have no control over.

The main question I need to address is generativity in art, and music in particular. This is obviously one of the main topics of the present research I started in 2017, but in general one of my main interests since the late 2000s. Before 2022, generativity was not a subject in the mainstream, public space, but with the advent of those platforms — I mean *platforms* rather than tools — it suddenly became 1) a topic of constant emergency and 2) it was immediately confiscated by the AI sphere. As if generativity did not exist in art before that… This is, and will remain, a significant cultural problem we will need to work against in the years to come, otherwise a vast expanse of artistic heritage may fall into oblivion, as if erased by backward masking effect.

During my time on Harvard campus, I did exchange with a few specialists about recent topics and trends in machine learning, and since then I have kept informing myself about the subject frequently. But clearly in terms of my actual artistic practice — apart from short experiments with Ircam's *RAVE* system

---

[95] In collaboration with Robin Meier, Frédéric Voisin resorted to supervised (multilayer perceptrons) and unsupervised (self-organizing maps) neural networks to generate musical structures and sound synthesis for the installations *Caresses de Marquises* (2004), *Symphonie des Machines* (2006) and *Last Manœuvres in the Dark* (2008).
cf. Frédéric Voisin & Robin Meier, « Playing Integrated Music Knowledges with Artifical Neural Networks », in *Journées d'Informatique Musicale* Proceedings, Ircam-Centre Pompidou, 2004

for real time sound generation based on deep learning — I feel I have stayed on the side of the road in this race, perhaps like many of my colleagues. For my personal case, I could guess multiple reasons for this : implementations tied to specific hardware (GPU) were not useable on macOS platforms until very recently ; still very few flexible implementations for musicians beyond "vanilla" AI techniques ; a very limited interest of AI communities towards (score-based) symbolic control compared with audio, which seems always more appealing to the industry ; in comparison with text and images, the great lack of operational, annotated data on music (written or recorded), beyond the typical categories of the market place (genre, BPM, mood, instrumentation, key, etc.).

For all these reasons, and probably others I didn't identify, it appears that we are still far — though this may well be false tomorrow… — from a prompt-based, ChatGPT-like or Midjourney-like system for written composition. We can only speculate about what such tools could mean for the future of CAC, if CAC still exists — or if it was merely a blink in the history of music composition techniques. But when observing the way we interact with those AI tools today in other domains, I am still struck by their proximity with how I would interact myself with the machine when composing with constraint engines[96]. In fact I should remind readers that machine (or deep) learning and ANNs are only a small portion of the whole history of artificial intelligence, which includes very different branches like expert systems, evolutionary algorithms, as well as… constraint programming.

The main difference between a modern prompt-based approach and musical constraints, as presented in MOZ'Lib for instance, is the implementation of rules. While the first allows to express an idea in a very direct way, via natural language, the second approach leaves us with a rather dry, often complex and time consuming programming interface to describe the desired result. Prompt-based systems

---

[96] In the following excerpt, artist Grégory Chatonsky describes four traits of what he calls "artificial imagination" in his practice, which I believe is very similar to my own, except for the very last one (big data) :
« First of all, there is the question of *hyper-production*, that is the ability to produce a very large number of files, a quantity that surpasses the logic of industrial production. We move away from the industrial model, indeed, it is no longer about subtractive production — by that I mean using a mold. Very concretely, industry is about using molds, with which the same object is produced in large numbers. But instead, it becomes additive production, as in the case of 3D printing, where each copy is unique and there are no economies of scale […]
Secondly, *infinitude*, that is the feeling that we cannot have access to everything. […] Yet this inaccessibility could constitute the definition of a world that exceeds our horizon. By definition, a world is what exceeds my horizon of expectations.
Thirdly, the *autonomy* of machines that are indeed programmed, but operate without the artist's intervention during production. The artist can be absent, or watch the production process passively, the outcome of which may surprise them.
And lastly, from a "détournement" (rerouting, hijacking) of *big data*, that is the use of data that are massively recorded in databases thanks to the participation of internet users. So it quickly becomes a matter of quantities, of the web, of an autonomous machine that resembles a world. »
cf. Grégory Chatonsky, *Qu'est-ce que l'imagination (artificielle)?*, Public intervention for ENS Post Digital, 2017
https://www.youtube.com/watch?v=t6Uh5d9-hnY (accessed July 2024)

explore latent spaces with up to several hundred or thousand dimensions and can produce results that are spectacular in terms of surface quality — they appear extremely plausible to us. However, they remain difficult to control very precisely : each new prompt usually destroys the previous result, and few of these tools allow to "block" clear properties of previous results and propagate them in the next ones. Meanwhile, constraint-based approaches make a clear distinction between what is a simple wish (heuristic rules) and properties which must be scrupulously respected in the result (true/false rules). In a sense, ANNs behave a lot like heuristic rules — and they are, clearly, much better at it than constraints — yet they are *essentially* heuristic. As such, I believe the constraints approach is still a more precise tool to work with, in the sense that it allows me to obtain exactly what I ask for — if it exists in the logical space, of course. But also in the way I interact with it : as I add more and more rules to specify a musical situation, a constraint engine gradually stabilizes itself, to the point where its results practically won't "move" anymore. I find it much easier to follow a machine that points toward the "perfect" solution, than one that constantly returns different answers to the same question.

Meanwhile, is CAC as we know it already obsolete? I don't think it is — at least for the time being. This is not only a technical problem, but a matter of *praxis*, as well as a cultural and pedagogical challenge. For instance, I have seen colleagues on social media mock the fact that ChatGPT couldn't convert frequencies into MIDI values accurately — a trivial task in any CAC environment. Joke aside, we need to be more aware about what those tools are, what they can and cannot do. The most fascinating for us I think is the way we can *interact* with them, rather than what they actually can produce, in "logical" terms so to speak. To the point that some AI artists today are already feeling nostalgic for the artifacts generated by imperfect AI models last than 2 years ago… I really hope, when the time comes of a hypothetical, modern AI-based CAC environment, that this new system will not be completely amnesic, but will somehow incorporate previous generative systems, including older AI approaches deemed "out of fashion"[97], so we can keep revisiting them if we so desire.

If the visual arts seem to benefit from a large technological advance on these topics, perhaps we need to see what they have to say already to better prepare for it, and compare it with what we already know after decades of generativity in music. What I find the most interesting about those new practices, is what they tell us about ourselves as artists — or *artistising* beings, and in particular by re-introducing the question of imagination, supposedly opposed to artificial imagination. CAC is not obsolete because of AI, for exactly the same reasons that human creation is not obsolete because of AI. Because artistic creation

[97] Online repositories already compile hundreds of discontinued AI projects, including artistic and musical efforts, cf. for instance the *AI Graveyard* archive https://dang.ai/ai-graveyard (accessed July 2024)

and imagination were inseparable from technique and artificiality from the beginning. As Bernard Stiegler taught us, there is no imagination without "exosomatisation"[98], i.e. externalization of biological and cognitive functions outside of the body. This is why I tend to prefer the term of "augmented imagination" in my work, rather than "artificial imagination" which is not exactly my concern. Obviously, AI-based generativity and imagination — as production of *images* (including "musical images" like scores or recordings) — is not the issue here. I would even say that using ANNs trained on all the art in the world, without the artists permission, to generate more art is not really the issue either — human art has been stuttering long before ANNs. What I think is the real risk today for artists is to become more and more tied to platforms as an ersatz for tools of creation. My position is that platforms are not tools, they are merely apparatus of capture, what Grégory Chatonsky calls a "consumerism of creativity"[99]. Outside of these platforms, I can only support artists to appropriate and divert these tools as much as they can — as they are already doing, in the open source spheres.

As a composer, occasional performer and musician-researcher in general, I am mostly curious about how the generalization of these tools will impact the audience's relationship with art. Moreover, what will be the function of artists in this new equation? We have many reasons to be both very optimistic and pessimistic alike… I now tend to imagine that many of us will become the *preparators of tools* for the

---

[98] « Bernard Stiegler, in his *Technics and Time* vol. 3, attempts to show that Kant's three syntheses, namely, apprehension in the intuition, reproduction in the imagination, and recognition in the concept will not be able to complete their functions without a fourth synthesis, which is artifact, since recognition implies that something is kept in memory, while this memory will also depend on external memory […] That is, imagination is not exclusively limited to the human mind, but rather it involves material supports without which the human mind will be limited to repetitive activities ; for example, one would not be able to understand the infinite if one had to count it. […] With the advancement of digital technology, the fourth synthesis constitutes a computational hermeneutics in which human and machine relation must be conceived as a recursive coupling, where the latter has more and more capacity to determine the former. We are not talking about humans on the one hand and recursive algorithms on the other ; the reality is that we are all integrated into gigantic recursive operations, for example, Google Search. […] Even if we follow the above argument that imagination is fundamentally artificial, we still need to clarify the role of "artificial imagination" in machines since machines are no longer just a passive medium of support but also actively participate in the cognitive process of imagination. However, this is not to say that artifacts before the computational age had no capacity to constitute the imagination. Art would, of course, be the counterexample since the question of imagination is fundamental in artistic, that is, artificial, creation and expression. In view of the banality that is given to the term artificial imagination, namely that machines will be able to replace artists in terms of art making, it will be the task of art to reopen the question of imagination by giving it a new role and going beyond the limits of industrial expectations. »
cf. Yuk Hui, « Imagination and the Infinite - A critique of artificial imagination », in *Balkan Journal of Philosophy*, 2023

[99] This may have already been partially the case, even before AI, for instance with the ubiquitous DAWs in music production, the quasi monopolistic position of certain software companies in the digital graphics world, and even to some extent with some of the programming environments which have become standards in our field, like MaxMSP. I am aware of that contradiction, but I don't think I am equipped to produce a better solution than invoking Bernard Stiegler's stance (after Jacques Derrida's) that every technique is a *pharmakon*, i.e. both a remedy and a poison.

art-making of others[100] — if not merely the fuel that these tools need to burn to operate. But has this ever been different? I feel that our "older" tools of artistic creation — in the broadest sense, technical and conceptual frameworks — used to be inherited at a much smaller time scale, or a much slower interpersonal pace. When I started composing, my first real "artworks" (which I consider so) could not have existed without the tools and ideas that some of my teachers *prepared for me*. Not only the skills they transmitted, but actually artifacts of their own practices, which branched out into mine and became my own practice. And of course, I am not even trying to list all the other influences which were digested unconsciously since I was born… As for whether these new tools will truly give rise to new, deep parallel artistic practices (perhaps "meta-artistic" would be the word?), or merely add up to more superficial cultural practices, confined to what can already be observed on social media, I sincerely cannot say…

If I circle back now into what was discussed in the last chapter, it is obvious that projects like *Game Theory* would become suddenly *much* more easy to manage with the high abundance of materials promised by generative AI — while more "classical" approaches to generativity seemed insufficient. If the current state of generative AI as platform-based, technological block is clearly non-desirable for me to take part in, I will need to investigate their open source pendants in the near future. In particular, the most seductive idea for me would be to explore ways to produce musical "in-painting". By that I mean, generating missing portions between pre-existing materials — in a way similar to Luciano Berio's work on *Rendering* (1989/1990) with fragments of Schubert's *D major symphony* — an idea I attempted several times to realize in the past with constraints, unsuccessfully. This would clearly solve one of the many issues of *Game Theory* — both for its instrumental score and electronics part — which was how to handle numerous unpredictable transitions between sections, and potentially offer alternate versions for each of them, to avoid strict repetitions when revisiting a "level" of the piece. Although I would clearly need to be careful about the timing to restart such a project : I can easily imagine a point, slightly too late, when technology would have completely trivialized the process, making it somehow unstimulating or too banal for me to carry through.

---

[100] « Looking to the future, we can expect the interplay of art and technology to deepen. Artists who embrace this ever-evolving landscape will contribute their unique perspectives to the development of machine learning and shape our collective relationship with it. This time around, the revolution will also extend to the audience's role. Passive viewers become participants in the making and remaking of the art. The very same AI systems that empower artists can empower the audience. Audience members can utilise AI tools to generate art, even if they do not have traditional artistic skills. AI democratises the art-making process, making it accessible to anyone with access to the technology. » cf. Thomas Hajdu, *From Duchamp to AI - the transformation of authorship in art,* theconversation.com, August 2023

One last thought before closing this research, is perhaps expressing what I feel is the most pressing issue for my own practice, that is to find ways to continue enjoying making music the way I used to, and still be able somehow to call myself an artist. After a few years thinking almost exclusively about generative scores, theorizing on how they could happen technically, what they meant musically, what inherent difficulty they brought, etc., I have perhaps hit a saturation point. In fact, over the last year, it actually did feel more natural to me to compose that way, and the new sensations I am finally getting out of it are slowly getting closer to the ones I had reached after my first years learning and experimenting with PWGL. That period didn't last very long in the end — 2011-2017 was the most "productive" zone for me — and I hope this new one will be more enduring. In any case, I believe that generative scores have become too integral to my practice to completely turn back now, and above all, they have become more than just an unusual apparatus for live performances. It is not just *the* final patch that I give to musicians for rehearsals and the final concert, but an augmented framework for my whole composition process, allowing me to prototype a piece in a more fluid or elastic way. This requires the musicians' participation not only in the last stretch, but helping me fine-tune the system for them, tightening the grip until everything is settled. For instance, I can always make assumptions about the difficulty of performance for a specific musical situation, but I will need to gauge the actual difficulty while observing the musicians in front of the score — perhaps adjusting some parameters on-the-fly to lighten their load, for a little while, and raising the bar progressively until the premiere. I want to keep composing that way, but also try to forget my scores are the way they are. This process needs to become "normal" for me, just like an extension of my previous ways of composing. In other words : forget generativity and keep composing, generatively.

Portfolio

All the files will be kept available on Google Drive under **this link**, or on simple request by **email**.

Mixed music (traditional scores) :

• ***Fire&Forget*** for piano and live computer (2017)
score and MIDI simulation recording (additional recordings by Marina Pizzi)

• ***Hashiya: Matn*** for string quartet and live computer (2018)
score, audio simulation & live recordings by Mivos Quartet (Paine Hall)
and Quatuor Léonis (Radio France, Studio 104)

(and two earlier works for reference)
• ***silent_data_corrupt*** for alto saxophone and live computer (2014)
score and live recording by Nikita Zimin (Ircam)

• ***Mémoire de l'eau*** for accordion and live computer (2015)
score and studio recording by Yohann Juhel

Mixed music (computer scores) :

• ***Game Theory #0 - En Rafale*** for flute and live computer (2018)
video score (silent) and live recording by Taiga Ultan (flute)

• ***all the memories are rigged*** for ensemble (2019 - unfinished) — excerpts video score (silent)

• ***Game Theory*** (2019-2022 - unfinished) — video score excerpts and interaction prototypes

• 藏镜人 — ***puppetmaster*** (2023) — video score (silent) and MIDI simulation recording

Acousmatic music :

• ***Virology*** (2018) — stereo recording

• ***Objets de désir illicites*** (2019-2022) — stereo recording
based on Carmen Diez Salvatierra's poems *28 formes découpées* (Les Presses du Réel, 2019)
and recordings by Carmen Diez Salvatierra (voice), Madison Greenstone (clarinet),
Alexandria Smith (trumpet) and Kathryn Schulmeister (double bass)

• ***Hyperlinks — asset(s) failed to load properly*** (2023-2024) — stereo recording and video score excerpts
Irène Blanc, flutes & Rafaël Carosi, guitars

Installations :

• ***Parameter // Perimeter*** (2019), installation diptych with Pauline Shongov

• 湮未没—***pas encore rien*** (2023), interactive and generative multimedia installation with NI Shan 倪杉

Improvisation :

• ***For SAAAAM***, live performance with Uri Schreter
(recorded at Harvard's Paine Hall, May 2019)

• **Unmapped -** ***Unamerika***
with João Fernandes, William Goutfreind, Jérémy Labelle, Gabriel Peraza Peraza
(recorded in Villiers-sur-Orge, EP published by ELLI Records, 2018-2020)

• ***Neglected Auguries***, duo with Kevin Gironnay
(recorded in Montreal, EP published by ELLI Records, 2019-2020)

(and an earlier recording for reference)
• **Unmapped -** ***Fulgurite***, with Emanuele Battisti, João Fernandes and Kevin Gironnay
(recorded in Paris, EP published by ELLI Records, 2013-2015)