

Universität für künstlerische und industrielle
Gestaltung Kunsthochschule Linz
Institut für Medien - Interface Cultures

Masterarbeit zur Erlangung des akademischen Grades Master of Arts

Live coding as a performative act
What is really alive?

Sule Suarez Leguizamón

Betreut von:
Univ. Prof. Dr. Manuela Naveau

Datum der Masterprüfung:

Unterschrift des Betreuers/der Betreuerin:

Linz, 2025

Contents

I	Abstract	3
II	Acknowledgements	4
III	Introduction	5
IV	Live Coding and liveness	7
1	Crafting through the time	7
2	The grammatic of performative coding	8
3	Liveness and DSLs: Domain Specific Languages for performing live	8
4	Algorithm thought	8
V	Show your Screen	10
1	Dusk of an electric forest	11
2	Empty Beginning	15
3	Toplap Linz	19
4	Art as a series of steps	23
5	Alternative archive: Not dead but not alive	24

"The machine makes the music, but I created the machine I don't know where responsibility lies in that situation"

AUTECHRE'S SEAN BOOTH

"Algorithms Are Thoughts, Chainsaws Are Tools"

TOPLAP MANIFESTO

I Abstract

Live coding is the practice of constructing and interacting with algorithms during a performance. It aims to convey algorithmic thinking, real-time composition and networked collaborations to the audience.[1] A distinctive quality of improvisation is the temporal element, which involves interacting with tools and the capacity to rethink rules and conditions. There is no concern with optimisation and efficiency, which characterise the technology industry, and instead there is a focus on embracing error, experimentation and open processes rather than specific outcomes.

In this thesis, I will review different experiences I have had in relation to my own live coding practice. I want to share the methodologies I have learned and discovered along my journey as a live coder. I will delve into my research on time in live coding, its playful nature, and how the community plays a fundamental role in consolidating an open source and diverse practice that aims to explore the computer as an expressive interface.

In the first part, I will talk about different performances and conceptual and technical concepts, as well as my experience co-organizing Toplap Linz, a community dedicated to the exchange of knowledge and experimentation in live coding.

Then, in the second part, I will go into more depth on certain technical and theoretical aspects of live coding, as well as a brief overview of the history of live coding and the most recent contributions made by the community in terms of developing new tools, experiments, and connections between different disciplines such as cinema.

Finally, I would like to share the documentation of the performances, photographs, and videos, as well as an interactive page that includes the code for some of the performances, where you can experience an excerpt of what happened and also access the code, allowing visitors to the website to copy, use, modify, and perhaps understand it.

Key words Algorithmic thinking, performance, artistic methodologies, process art, real-time composition, livecoding community.

II Acknowledgements

III Introduction

My interest in programming was a coincidence. I had finished my bachelor's degree in fine arts and had mostly worked with "analog" techniques (which it means that computer art was something foreign to me). After my time at the art academy, I found myself with the terrifying freedom of no longer being in an institution. It was at that moment that I began working with Juana Prieto, and the two of us started this journey into the field of creative coding.

We had done our undergraduate studies together and had formed a friendship since then. Juana had the idea to make a GIF movie for the web — the text would lead to new places, a linear narrative that ultimately became a labyrinth of moving images and blinking text. Neither of us had extensive knowledge of programming, but I joined the project because the idea presented itself as a challenge that combined things that intrigued me, like the relationship between image and text, where both are equally important.

We spent months working with the image and text archives we each had, merging narratives and creating labyrinths together. Those were months of discovery for both of us. We had received support from the Cinemateca of Bogotá and were working in the institution's labs. For that project, we used Twine, which is a mix of HTML and its own CSS-like style. As the weeks went by and we read the documentation, we began using JavaScript within the program to add some logic and elements that made sense within the web aesthetic — like decision boxes or having the audience answer questions.

From that moment on, programming stopped being an unattainable mystery reserved for engineers and computer scientists. I began to explore the potential of this language used to communicate with machines. While we were at the Cinemateca in Bogotá, we came across Toplap Bogotá, a live coding community. RafroBeat (who is also a member of Toplap) was giving a workshop on Tidal Cycles at the time, which marked the beginning of my journey with code as a performative act.

The week after that workshop, Malitzin Cortés and Iván Abreu, two Mexican artists who work with code, architecture, and expanded cinema, gave a workshop where we used film archives as a base to create live coding performances. That was my first experience performing in front of an audience, and the first time Juana and I performed together.

Together, we created the collective "Vestidas y Alborotadas" and became active members of the Toplap community. We performed numerous times in Bogotá, Medellín, and Manizales. Together, we questioned the presence of the body on stage. And from that question, we decided to become monstrous beings and embark on this path of learning and experimentation.

In each performance, we embodied a different monster — and different rhythms too — and our intention was always to make people dance with fast-paced sounds,

sampling our own voices and surroundings. We used our archive of photos and videos to create visuals. With each performance, we wanted to offer the audience a glimpse into the world of each monster — their own sounds and images — creating a narrative around these fictional beings.

When I arrived at the Kunsthochschule Linz, I found an exciting musical and artistic scene in the city. The art scene, both inside and outside the university, thrilled me with its openness — the spaces and the community are genuinely willing to see and listen to what you do. Over time, together with Gorka Egino (also a student at Kunsthochschule Linz and a livecoder), and with the support of the Noise Meetup Lab, we decided to create a live coding platform in Linz called Toplap Linz. Through a series of workshops and jam sessions, we aimed to build a space for sharing knowledge and fostering community.

The creative or inventive capacity of this language resides in its more human side [2], in contrast to the process-optimization mindset we usually associate with code. Here writing code is not about debugging but about expression and performativity of algorithmic thought. While it is a language that requires precision to be executed, its potential to expand the written word is unique.

Live coding is a practice in which the programmer writes their own writing machine. These are open machines, supported by a community. There are always collaborations, rewritings, updates, and improvements. A program is not programmed to be used later, but is used by programming it[3].

But going back to the basics, What is live coding? One of the most common definitions is that it is a practice of programming in front of an audience, where the code is executed in real time, and the performer interacts with the code as part of the performance.[4] This definition, however, does not encompass all the nuances and possibilities of this practice. We would have to go deeper into the unfolding of the code within the performative act, the characterization of some Domain Specific Languages (DSLs), the playfulness within the scene, and the community that has been built around this practice, and its open source philosophy.

This thesis is an attempt to explore the liveness on live coding, whether in a solo or collaborative performance, on site or online, and the way in which the code is written and executed in real time. How does time exist in this on the fly programming? How does invention and intervention lead to playfulness?

IV Live Coding and liveness

1 Crafting through the time

Now I would like to start with Ursula Franklin's concepts of Holistic Technologies and Prescriptive Technologies. Holistic technologies can be directly associated with artisanal processes in which the artisan has mastery of the material from start to finish and leaves a personal mark on the final result. Prescriptive technologies, on the other hand, are technologies that break the process down into well-structured, hierarchical steps, making the results predictable and scalable.[1]

When talking about technology, we should not limit ourselves to devices, machines, and gears. Technology is, above all, a way of doing things that encompasses ideas, mindsets, and procedures. And within these technologies, we can find groups of people who identify themselves by a common practice and are defined by it. Technology is linked to culture in that it establishes values and defines ways of doing things.[5]

So, what can we say about live coding as a practice and technology? Since its inception, it has been characterized by being open source, by using the computer as an expressive interface, and it has been sustained over time by a community that has decided to take care of the software and expand its practice to different disciplines. The first live coding performances and practices can be traced back to the early 21st century, particularly in the context of electronic music and digital arts. Live coding, at its core, is a form of real-time improvisation that uses programming as a creative medium.

2 The grammatic of performative coding

Some Manifiestos to understand the live coding practice, where does it come this idea of liveness Philosophy behind live coding

Generative code Manifiesto

Toplap manifesto

NEOKHIPUKAMAYOQ Manifiesto

3 Liveness and DSLs: Domain Specific Languages for performing live

The concept of liveness in live coding is closely tied to the use of Domain Specific Languages (DSLs). These languages are designed to facilitate specific tasks and often come with their own set of rules and structures. In the context of live coding, DSLs can enhance the performative aspect by allowing for more intuitive and expressive coding practices. The liveness of the code is not just about its execution in real time, but also about how it can be manipulated and transformed on the fly, creating a dynamic interplay between the performer and the code.

struggles and learnings Laptop jamming is a practice that combines live coding with collaborative improvisation, where multiple performers use their laptops to create music or visual art in real time. This section is a collection of my own experiences and learnings as a live coder, focusing on the practical aspects of live coding, including techniques, tools, and strategies that I have found useful in my practice.

Toplap experience, Pastagang Coming soon

4 Algorithm thought

The first livecoding performances and practices can be traced back to the early 21st century, particularly in the context of electronic music and digital arts. Livecoding, at its core, is a form of real-time improvisation that uses programming as a creative medium.

Among the most significant milestones in the development of livecoding as an artistic practice, we can mention ChucK, developed by Ge Wang in 2002. ChucK is a real-time audio programming language that allows precise manipulation of sound and time. Besides the tool itself, it is important to highlight the performances that began to be carried out around its use, such as "On-the-fly Counterpoint: A piece for two laptops and two laptop projectors" by Ge Wang and Perry Cook in 2003[6].

In 2004, Toplap (Temporary Organisation for the Promotion of Live Algorithm Programming) was founded in London, marking a significant step in the livecoding community. Toplap's mission is to promote live coding as a performance practice and to create a space for collaboration and knowledge sharing among artists and programmers. The community has since grown globally, with local chapters forming in various cities, including Toplap Linz, which I co-founded in 2024

V Show your Screen

In live coding performances, one of the basic principles is live improvisation: creating what you hear and see from scratch. Algorithms are written and operated on in real time[7]. Another key element is demonstrating to the audience that what they are witnessing is being created live, which is why the performer's screen is typically projected, revealing everything, including mistakes.

However, not all live coders start from a 'blank page', demonstrating exceptional skill and speed; some use prepared code. I have chosen to do this on many occasions during solo performances as it makes it easier to control the sounds and images, and prevents the programmes from becoming overloaded, crashing or producing poor sound quality. Although I don't entirely agree with purist live coders who disapprove of prepared code in live performances, I would argue that 'from scratch' performances have a special quality in terms of time, disposition towards the live coding tool (DSL), and interaction with the audience and other artists, particularly when there is more than one on stage.

Before a performance, I spend time rehearsing and planning what I want to do, the sounds I am going to use and the images or textures that will be on stage. This process is similar to painting: you prepare a canvas, select a colour palette and choose a motif. However, other experiences within the practice escape this type of prior ideation and are subject to the moment itself and the unpredictability of the people and elements involved in the performance. Examples include jam sessions, collective performances and 'from scratch' performances.

I will talk about some of my own performances and those I have taken part in. I will identify the different working methods and discoveries that appeared during the process. I would also like to explore the different qualities of time in various performance modes and address the live aspect of live coding, considering its potential for experimentation. I would also like to talk about the importance of community in live coding and how this can lead to new approaches to time and performance.

1 Dusk of an electric forest

'Dusk of an Electric Forest' was a performance piece that I created for the 'Sound Fever' event at the Ars Electronica Campus in 2024. The call for submissions for that edition invited reflection on questions such as:

- What is the sound of the Anthropocene? How do we listen to the Anthropocene?
- How do we listen to ourselves in the Anthropocene?
- How does the past resonate in the future? What are the future sounds of our digital and physical selves?
- How can music alter narratives and adapt to the constantly shifting ecosystem in which it is created? Is it possible to change the listener's attention into a shared space and an imaginative space?
- Can we decentralize the human realm, amplifying the voices of non-human agents?

When conceptualising the performance, I used the sounds of a hyper-technological society as my starting point: noise, the incessant hum of construction, machines that shake the ground, the saturation of cities and silence — the decline or absence of other species.

The Anthropocene has impacted numerous ecosystems, and the concept of human expansion and growth has led to a constant state of emergency. We are suffering the consequences of our own actions, and paradoxically, our surroundings become quieter amid our own noise in these environments that we have modified.¹

I would like to mention the work of Bernie Krause on biophony and soundscapes. In the late 1960s, he began researching and documenting the sounds of natural ecosystems around the world. His archive includes ecosystems that have since been wiped out by human activity. He has collected over 5,000 hours of recordings from natural environments, including at least 15,000 terrestrial and marine species from around the world [8].

He took his orchestra idea to a fellow scientist, the late Stuart Gage. Together, they developed the acoustic niche hypothesis, which posits that living beings that share an ecosystem evolve to produce sounds with different rhythms and tones so as not



¹You can see the video documentation of the performance here:

to interfere with each other [9]. This is important for finding mates, food and water. It is a harmonious system in which different species coexist alongside their diverse sounds, which are astonishingly beautiful.

Listening to Bernie's early recordings from the 1970s instantly evokes a sense of nostalgia for a distant, highly diverse world that is far removed from city life. Although many ecosystems remain intact, many are under threat. The sound of cars or air traffic makes it difficult for many species to hear each other, leaving them vulnerable to predators or making it difficult for them to mate. And as for us, our noise pollution is often unbearable. While cities may have an interesting soundscape, rapid growth or change leads to noise becoming predominant.

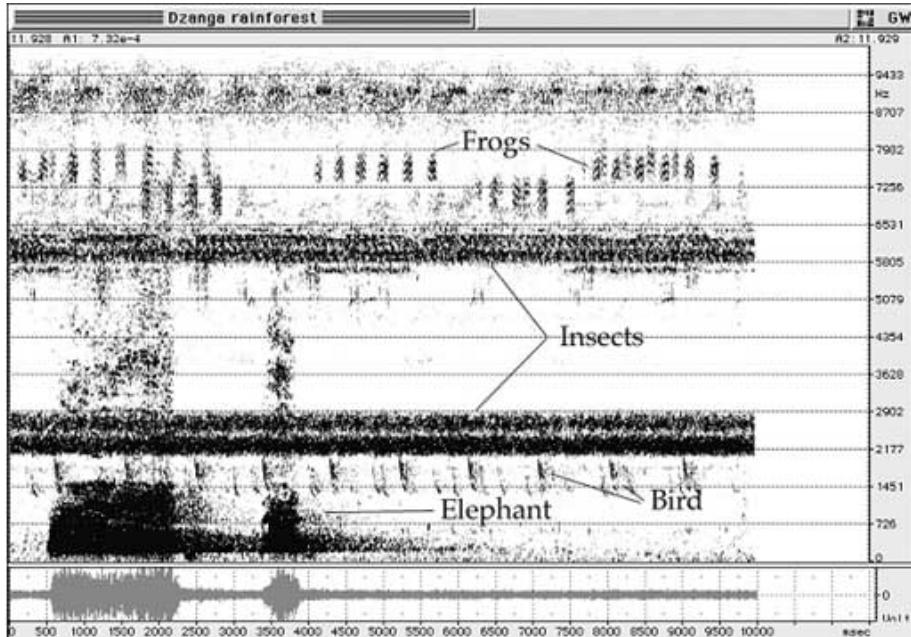


Figure 1: Dzanga Rainforest, Central African Republic. Bernie Krause. Spectrogram showing different species of animals.

Krause made his first sound recording of Sugarloaf State Park in Sonoma Valley, California, with the intention of documenting the environment near his home. From then on, he regularly recorded that park. In 2011, the area suffered a severe drought that lasted for years. As can be seen in Figure 2, the spectrogram shows how the soundscape gradually falls silent: birds no longer sing as they once did because they have to conserve energy due to food shortages, while the insect population declines and the vegetation dries up. Finally, in 2017, the forest fire broke out, affecting much of the area.

Silence is also a homogeneous mass of noise. In *Dusk of an Electric Forest*, I explored the idea that silence is not an absence of sound or a pause, but an overlapping mass of noise. To achieve this, I recorded sounds from construction sites in the city, such as the noise of drills, the movement of heavy machinery across asphalt, and car alarms. I also recorded the Traun River flowing and insects singing, for example.

During the performance, moments of unintelligible, grainy murmurs can be heard, which then become more rhythmic and repetitive. The performance begins with the

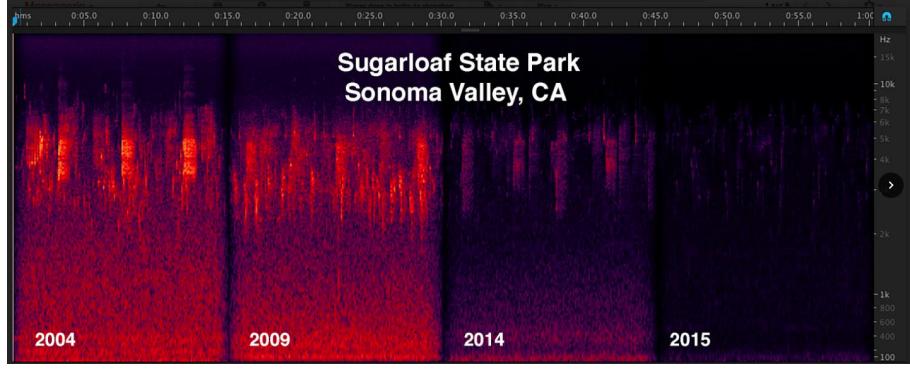


Figure 2: Spectrogram of the soundscape of Sugarloaf State Park, California, from 2011 to 2017. Bernie Krause.

sounds of insects moving from one source to another in stereo. Samples of drills and hammers are mixed with the continuous murmur of water.

The visuals feature a glitch grid that resembles landscapes or neon cartographies. As the performance progresses, these small squares change position and colour, and the textures rotate and become increasingly grainy, in line with the sound.

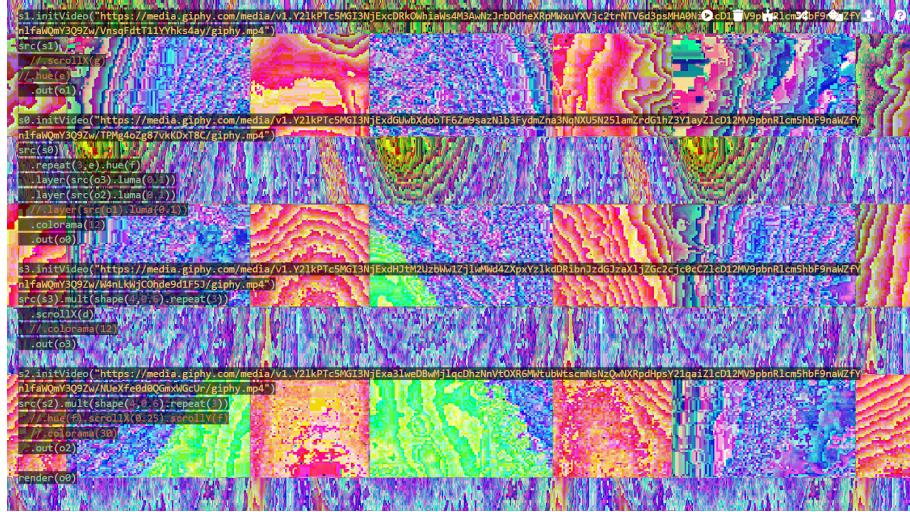


Figure 3: Hydra visual from Dusk of an Electric Forest.

Drawing inspiration from the mythical figure of the snake woman or the goddess Coatlicue, as described by Gloria Anzaldúa in *Borderlands: The Frontier*, the new mestiza is a being that bridges destruction and fertility [10], and is resistance itself. A writhing snake moves through these hyper-glitched landscapes, falling and approaching the audience with its mouth open and fangs exposed.

In pre-Hispanic cosmogony, the snake symbolises fertility and acts as a guide between life and death. For instance, there is the legend of Bachué, the woman who founded the villages of the Bogotá savannah. Once her children were ready for her departure, she plunged into the river, transformed into a snake, and disappeared forever.

Just as snakes shed their skin, we are constantly changing and leaving behind the ruins of our past selves. However, it is interesting to consider where we are going

each time we shed our old skin. While we are capable of chaos, we are also capable of achieving balance. This female figure also symbolises rebellion and care.



Figure 4: Picture from Dusk of an Electric Forest performance. Photo by Flavia Somarriba

For me, Anzaldua has been a key figure in feminism and philosophy. In her writing, she reimagines Mexican and Chicano spirituality from the perspective of historical and political violence and domination. She is also a key figure when it comes to issues of identity related to mestizaje and borders. Gloria Anzaldua explores forms of resistance by critically and constantly reconfiguring our identity formations and ontological categories, with the aim of achieving positive social change and global justice.

For this performance, I aimed to transform the chaotic sounds of the Anthropocene into repetitive, rhythmic noise by turning them into grainy, elongated and distorted sounds. I selected sounds that typically accompany urban growth, such as construction work, alarms, and machinery. These sounds are very prevalent today, as megaprojects often take precedence over architectural and landscape conservation. Despite all this noise, other small animals, such as insects and birds, can still be heard, finding a place in our fast-paced way of life.

But how can we think about the silence of the Anthropocene when it is spreading and dominating other species and natural cycles? Perhaps one day our species will no longer have this desire to spread without limits, and our sounds will once again become diverse.

Another key aspect of this performance was considering those symbols that resist Western homogenisation and colonial processes. We should ask ourselves what we can learn from cosmogonies that seem so distant to us, how identity can be built from miscegenation, mixtures, and exchanges, and how we can update our own worldviews.

Although we are aware of the environmental emergency we are living in, it seems that our relationship with accelerated change remains unchanged. Perhaps in the future, we will either become a homogeneous mass of noise or make room for other elements to emerge more organically.

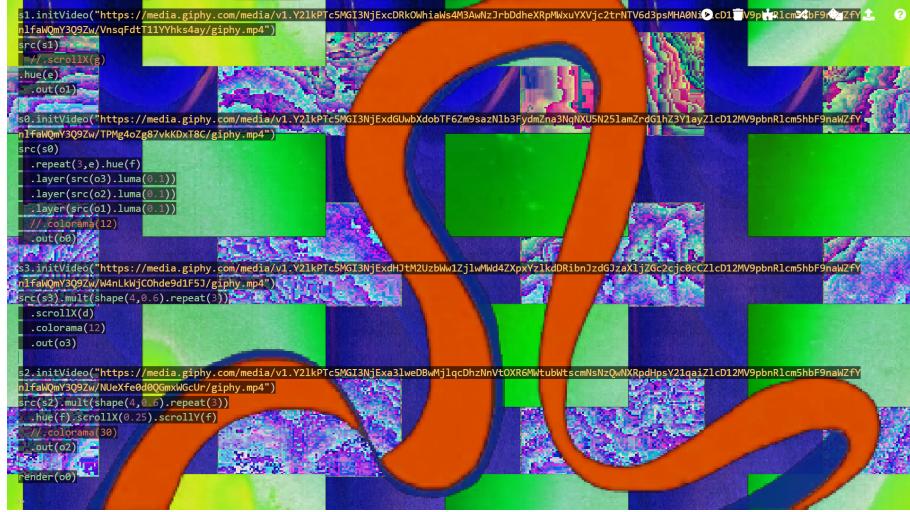


Figure 5: visual code from Dusk of an Electric Forest.

I chose the order of the sounds to create the atmosphere I wanted and had some effects and parameters I wanted to implement in the code. As for the visuals, I had three analog photographs that I had taken some time ago and modified with methylene blue, so they had become colored spots, and a public domain animation of a snake. My intention with the visuals was to create several layers that would gradually become saturated and grainy, until finally the snake appeared on top of them all.

On stage, I sought order once again, rediscovered the sounds I had planned, modified the code I had already written, ran it, commented on the sections I wanted to mute or might need later, changed parameters, and also gave myself the freedom to improvise with the palette I had selected.

Finally, to mention some technical information, in this performance I used two live coding environments: Hydra for the visuals and Tidal Cycles for the sound. Hydra is a tool inspired by analog image synthesizers, and Tidal Cycles mainly uses samples and synthesizers as its main resources. In chapter 2, I will discuss these two environments in more detail, including some technical specifics and a little about their history.

2 Empty Beginning

The 'from scratch' approach is playful and embraces error. There is no pre-established order, and a musical composition is created in real time. Starting with a 'blank sheet', the performance itself then develops.

At the same time, the audience participates and witnesses the performer developing

their ideas, with algorithms serving as the primary notation. Audience members do not need to be familiar with programming to enjoy the performance. The visible code reveals artistic thought 'in action', embedded in the materiality and mediality of the performance itself [11].

For me, the first few minutes of a performance 'from scratch' can be overwhelming, with that initial silence and an audience that is both expectant and sometimes confused. Then movement appears on the screen: the performer begins to write, and from that moment on, the performance starts in earnest. The live coder continues to write, and as the text transforms, so do the sounds and images.

During my live coding practice, the presence of a collective and a community that organises events, innovates in the field, experiments and teaches has been crucial. Toplap is a collective founded in Hamburg in 2004 that has been dedicated to promoting and practising live coding ever since. Toplap has expanded worldwide in the form of local nodes that organise events, workshops, concerts and meetings. Anyone can start a new Toplap node.

This global live coding community organises events that seek to connect artists from different parts of the world. Numerous online events are broadcast live, including Livecodera, Toplap Solstice and SuperCollider meetups. The most important event is the International Conference of Live Coding, which is organised every year in a different location, bringing together performers and academics to share their research and artistic projects.

When Gorka Egino and I participated in the Toplap Solstice in December 2024, we were in different locations. For this performance, we decided not to have any prior ideas or plans for our session. I was in charge of the sound, while Gorka was in charge of the visuals. We used Flok, a collaborative, web-based live coding editor that allows several people to connect to a session via the same link.

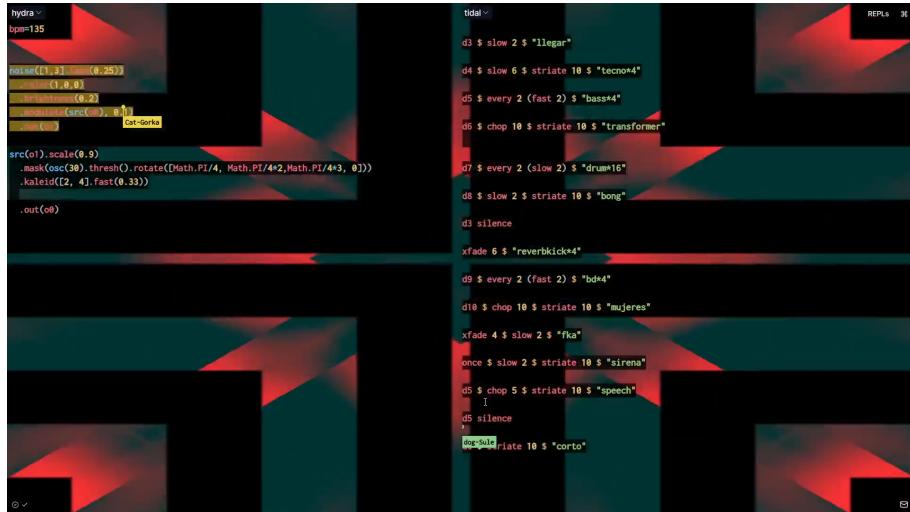


Figure 6: Screenshot of the YouTube recording of the performance by Sule and Gorka at Toplap Solstice, December 2024.

In the YouTube recording of the performance (see footnote), you can clearly see how the code develops. In it, the cursor embodies the artist's body in a certain way,

travelling across the screen and writing, hesitating, correcting, pausing, modifying and activating the code.

This type of event is an interesting format for those who are new to or curious about the practice, as it provides an opportunity to see in detail how different artists use the tools, offering new ideas and introducing lesser-known live coding platforms.

A variety of musical genres are also showcased, ranging from the most experimental (noise and ambient) to the more conventional (techno, jazz and cumbia). Toplap Solstice events last 40 hours continuously in winter, enabling people in different time zones to connect and watch performances.

Another performance I would like to mention is my participation in the AMRO Festival in 2024. For this set, I used the Algorave format, which has also brought live coding to more mainstream venues such as clubs and discos. While the sounds may be diverse, the ultimate goal is to dance and have fun.

AMRO (Art Meets Radical Openness) is a festival, platform and community that celebrates art, hacktivism and open cultures. Since 2008, it has been organised by servus.at in collaboration with the Department of Time Based Media at the University of Art in Linz [12]. The event brings together artists, activists, developers, researchers and hacktivists. Workshops, talks, performances and an exhibition covered topics such as community building in digital and physical spaces, ethical uses and critical AI analysis, open tools, free and open-source software, and the environmental impact of nowadays technology.



Figure 7: Performance at AMRO Festival 2024. City Landscapes by Sule Suarez Leguizamón. Photo by Martin Bruner

I titled my performance 'City Landscapes', for which I planned to perform a live-coded techno set using my visual archive of different cities, recorded with my Samsung PL20 compact digital camera. I have had this camera since the early 2000s and its resolution is lower than that of many current digital devices. When record-

ing video, it produces noticeable texture and pixelation, and the colours can appear pastel depending on the lighting conditions. I created a sequence using some clips I took with this camera, which I then used for the visuals during the performance. Within the Hydra environment, I modified this sequence using code, which allowed me to create modulations and experiment with time, textures and colours.

For this performance, I primarily used samples from the SuperCollider Tidal Cycles sample library. This library contains sounds from iconic drum machines, such as the Roland TR-808 and TR-909, which I used in City Landscapes. I created a sequence of drum rhythms, a fast bass line and kicks with reverb, as well as claps and snare. I recorded myself saying short phrases, divided them into small parts, sped them up and randomised them to fit the composition. It was minimalist techno, retaining the essential elements to make people dance and creating some polyrhythms so that it wasn't monotonous.

Figure 8 shows the code I used for Tidal Cycles and Hydra. I used these two environments locally in the Pulsar code editor. In the image, you can see how the previous frame freezes in the visuals and remains on the screen for a few seconds, scrolling sideways, and also how the image blurs in different directions.

In this performance, improvisation did not start from scratch: sounds and images had been curated in advance, and the potential course of the performance had been planned and rehearsed. However, there was room for error, and I did not intend to be entirely governed by a strict plan and schedule. Although we can consider the computer and live coding to be contemporary musical instruments, the concept of a musical score becomes blurred in this context.

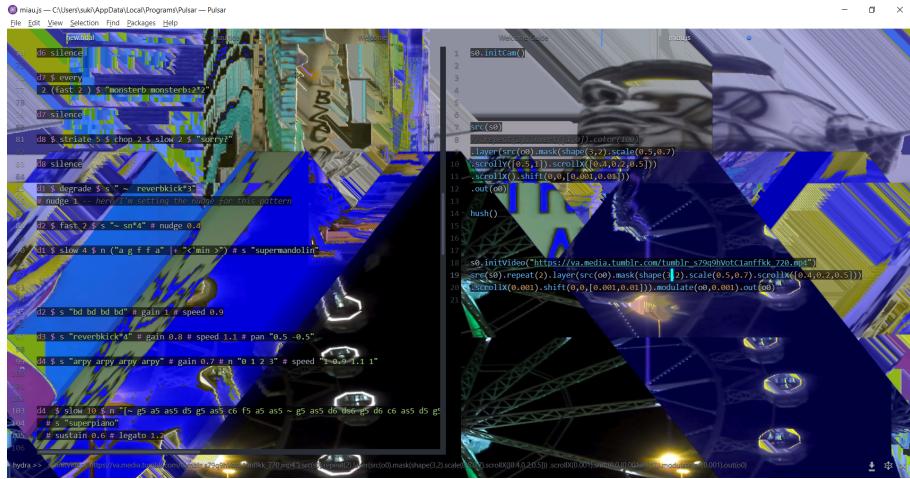


Figure 8: Visual and sound code from AMRO Festival performance.

During the festival, I had the opportunity to meet other live coders, including Timo Hoogland (NL) and Lina Bautista (CO) (also known as Linalab). They performed together with Flok and Mercury, and also led a collaborative music workshop. I also met Rémi Georges (FR), whose performance incorporated family archives and live coding, and Martin Gius (AT), who ran a Networked Improvisation workshop in TidalCycles and led a live coding improvisation session, connecting people in Linz on the day of the event with members of Toplap Vienna. This session provided me

with ideas on how to create a live coding session with lots of participants, bearing in mind that this can often descend into chaos, but there are still ways to mediate the participants' experience.

3 Toplap Linz

In June 2024, I had the opportunity to teach a workshop on Tidal Cycles with the NoiseMeetup collective. This was my first time teaching a workshop on this tool, and I invited Gorka Egino to collaborate with me. We used Flok.cc to connect all the participants to the same platform. First, we provided an introduction, covering how the tool works, how to create initial sounds and how to generate compositions using code. The second part consisted of playing together using Flok.

From that first experience, we learned that in order to start a jam, certain rules must be established, as having more than 15 people play together on the same interface can quickly lead to chaos. This workshop also led to the creation of Toplap Linz. Founded in 2004 in the UK, TOPLAP (Transnational Organisation for the Parsimony of Live Art Programming) is an organisation that explores and promotes live coding. This collective functions like a rhizome: anyone with the necessary motivation and knowledge can start a Toplap group and establish a community spread all over the world.

TOPLAP Linz is a TOPLAP node located in Linz, Austria. It was founded on 2024 as part of the broader international TOPLAP community to explore and promote live coding practices. It is aligned with the open-source philosophy and one of its core pillars are being inclusive and accessible.

As a result of the creation of Toplap Linz, we also started organising meetings to answer questions, share knowledge and try new things. However, these meetings were not as well received, so after a few attempts, we began to wonder if our reach was limited, or if the invitations were ambiguous or unappealing.

Some time later, in collaboration with Noise Meetup, we held a second two-day workshop. We repeated the workshop and jam format, and this time, as well as teaching Tidal Cycles, we included Hydra. This event was held at DH5. We dedicated the first day to workshops and the second to a computer jam. We had more participants this time, and overall it was a successful event — we were all writing code and having fun by the end!

This time, we made sure to install Tidal Cycles on the computers of those attending for the first time, despite knowing it would be time-consuming and potentially cumbersome. We felt it was important for people to have the tool ready to use outside the workshop, in other contexts. We also wanted to provide a space for group installation, as this process can be confusing if you don't have experience with operating system command applications. Our goal was to make the experience as enjoyable as possible from the outset.

Then, we got down to business and shared a Flok.cc session where everyone could

connect at the same time and review the code. We also created a cheat sheet for Tidal Cycles and Hydra with guidelines to help participants start exploring the tool. It's a simple guide offering an overview of the possibilities of these two environments for experimenting with sound and visuals respectively (see fig. 9). From my experience of learning to use different software, I have personally found it quite useful to be able to follow the process on my own computer. This allows you to interact more effectively with the tools when someone explains something in a masterclass.



Figure 9: Flyer for workshop on Tidal Cycles and Hydra by Toplap Linz, March 2025, Linz.

On the second day of the workshop, we devoted ourselves solely to computer jam sessions. We timed each session, starting with five minutes, then moving on to ten, twenty and thirty minutes. In the longer sessions, we worked with themes. For instance, in one session, we used the symbol '520', which is used in China to say 'I love you'. The visuals and sound were often humorous and corny. In another session, someone suggested using horses. Interesting things happened in this dynamic and we were all able to collaborate. Sometimes, after a jam, questions arose about how to create a particular rhythm or apply an effect to the visuals. We realised that, in the jam dynamic, people dared to try more complex things.

As a general rule, we decided not to copy code from sources outside our Flok page, such as the official website or online tutorials. There were two main reasons for this. The first is that it is difficult to predict the result of mixing pieces of code that are not understood — it may not work at all. The second is that if you join someone else's code to yours, it will most likely break and stop working. Therefore, we focused on the process of crafting the code, adopting a slower but more deliberate approach to ideation and elaboration. We always started from scratch and developed the jam together.

Another important principle that applies to any type of jam session is the importance of active listening. This involves understanding each other while doing something

together and knowing when to try a sound from the library and how to introduce it. When you are unsure of the sound you want to introduce, it is often better not to go in at full volume. You could also choose to play divisions and rhythms similar to those of others, or simpler divisions, before moving on to polyrhythms.

In a group of more than six people improvising simultaneously, it is important to understand that there are times when it is better to give others space and remain silent. These pauses are useful for taking a step back and observing what is happening without the pressure of having to keep programming. They also give you a moment to think about how you can contribute something new to the jam, such as another sound, rhythm, modification of the existing code, image or effect.

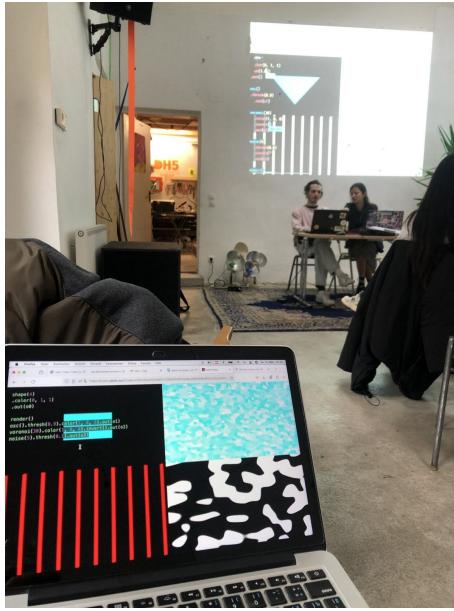


Figure 10: Picture from Toplap Linz meeting, March 2025. Photo by Mascha Ilich

This experience makes the idea of starting from scratch and programming on the fly feel even more real. Jam sessions with others greatly increase the level of spontaneity, as it's impossible to predict what each person will contribute to the performance. This is enjoyable and enables you to connect with others by creating a shared audiovisual experience. The absence of control enables you to achieve the unexpected. Ideas are conceived and carried out almost simultaneously.

Some time after this workshop, we collaborated with Servus.at, Stadtwerkstatt Linz, and Ljudmila (Ljubljana) to organize a satellite event taking place in Linz for the ICLC International Conference of Live Coding, which was held in Barcelona from May 27 to 30, 2025. We invited live coders who were traveling to the ICLC event and making some stops along the way to give concerts. Some members of Toplap participated: Te-En Chen, map(h) collective (Sule Suarez and Gorka Egino), Confirm Humanity (Cecilia Bojanik), and incredible live coders visited us: c-robo(EU), Blaz Pavlica (SVL), and Shelly Knotts (AR)



TIDAL CHEATSHEET

CONTROL PARAMETERS (with #)

PARAM	info
s / sample	folder name
n	sample select
FX	range
gain	0.1
pan	0.1
note	whatever
lpf	50..20000
room	0.1
size	0.1
krush	0.5

RESTRUCTURE PATTERNS

rev, rot, bite p

MODIFY FUNCTIONS:

every n f, sometimes f, juxtapose f...

MINI-NOTATION (between "")

notation	meaning	function
[]	grouping	-
a * n	faster by n	fast n play n
a / n	slower by n	slow n
[<a b>]	go choosing...	-
~	silence	mask b
a? a?n	degrade	degrade, degradeBy n
a(n, m)	euclidean rhythm	-

EUCLIDEAN RHYTHMS
 (2,5) : A thirteenth century Persian rhythm.
 (3,4) : Cumbia from Colombia.
 (3,5,2) : Rumanian folk-dance rhythm.
 (3,7) : Bulgarian folk-dance rhythm.
 (3,8) : The Cuban tresillo pattern.
 (4,9) : The Akak rhythm of Turkey.
 (4,11) : Frank Zappa: Outside Now.
 (5,7) : The Nawakhat pattern, a popular Arab rhythm.
 (5,16) : The Bossa-Nova rhythm necklace of Brazil.
 (7,8) : A typical rhythm played on the Bendir (frame drum).
 (7,12) : A common West African bell pattern.

SAMPLES

PERC	kit	house (8), ccpu (25), ccpu2 (31), gretsch (24)
	cymbal	808cy (25)
	clap	clap (8)
	snare	snare (90)
	tom	lt (16)
	conga	808ic (5)
	kick	909 (1), hardkick (6), kick (21)
	hand drum	hi (8)
SYNTH	extended	can (24), ulgab (5), ul (10)
	dsynth	dsynth (3), arpy (11), notes (15)
	bass3	bass3 (11), dbass (12)
	ACOUSTIC	pluck (17)
GLITCH	procshort	procshort (8), yeah (31), glitch (8), wind (10), haw (6), psr (30)
	FOLEY	foley (270)
	VOICE	alphabet (24), mouth (15)

Figure 11: Tidal Cheatsheet, Toplap Linz. 2025

It was a night where we all collaborated. The guest live coders only planned to do audio, so the performers and members of Toplap Linz did the visuals for those who only planned to do a sound intervention. In my experience at these types of events that are solely focused on live coding, everyone is attentive to technical support, open to helping other artists if they need a cable, an interface, or if the interfaces crash and troubleshooting is required.

It was a night full of a variety of rhythms and tools. Everyone was using different live coding environments and software, such as SuperCollider, Strudel, TidalCycles, and C-robo also made a connection between a live coding interface and Ableton, which is a DAW (Digital Audio Workstation). It is interesting to learn about different notations and their own possibilities in terms of expressiveness and complexity.

Each artist has a specific background and different approaches to the same medium. When you are a spectator at a live coding performance, a whole particular construction of thought is revealed. A meticulous construction of algorithms that create sounds and images and are displayed to the public. It is a work of technological crafting, it takes time to develop, and the changes are gradual and depend on the direction decided by the artist and the temporal space of the performance.

People from the audience approached us after the performance to ask if we were doing everything with code, if the visuals and sound corresponded to the text that was seen in the projection, changing, becoming longer, or being eliminated. In live coding, there is a need to state that what is happening is processual, that we are not working with fixed technology, and that the process is the protagonist of the performance. There is also a desire to engage in participatory practices, making it clear that there is an element of improvisation and that both the audience and the performer can be equally surprised by the paths the performance may take.



Figure 12: Algorave, Toplap Linz. 2025. Photo by Lina Pulido

4 Art as a series of steps

In this chapter, I have taken a journey through different experiences within the practice of live coding, trying to cover different methodologies of conceptualization and realization. There have been different degrees of improvisation, planning, and execution. This has also been a journey of my own to understand what brings a practice such as programming to “life” and what happens when it collides with the living arts.

For me, the performative practice of live coding is a field of experimentation in which I invite the audience to observe how I construct what they are hearing and seeing, in which the audience’s gaze is active, they are witnesses to constant decision-making, they can see when there is doubt, errors, skill, audacity, etc. The computer contains the body. The body extends the computer. In this exercise of response and interaction, the body is not passive towards the interface; it is the body that activates it dynamically.

In my own practice, there has been ambivalence in terms of score, choreography, and improvisation. In some cases, such as my performance Dusk of an Electric Forest, there is a concept and a prior curation of images and sounds, the code is constructed from an established frame, and there is more or less an idea of different moments in the performance.

However, even in this type of experience, the performance is not limited to the mechanical act of activating and deactivating lines of code. It involves working with a limited color palette, but with several parameters that remain open and escape any planning. Although in Dusk of an Electric Forest, I had tried different effects, rhythms, and patterns. When the moment of the performance arrived, things were different. Things that I thought would work suddenly didn’t sound as I expected. Adrift, I had to find new ways to construct the sensations I wanted to convey and

discard some pre-established ideas. A space of uncertainty opens up, and being flexible is the key to enjoying and constructing the performance.

Conditions change radically when you try or practice something alone, without an audience, and when you face the stage. Things look and sound different on speakers than on headphones, or on a computer screen and a projector. The acoustic and visual conditions are totally different, which means you have to be able to adapt.

5 Alternative archive: Not dead but not alive

I have talked about my own methodologies and a few experiences within live coding performance. I have also been thinking lately about how to create an archive of something that is so difficult to archive, such as performance. So I did an archaeology of my own files and coding sessions and found sound recordings and excerpts of visual codes.

I created a brief documentation on a web page to help you visualise some of the performances I have mentioned. In the process of excavating my files, I also found a lovely composition I was creating using samples of electromagnetic field recordings I took in the countryside. This now exists as a bonus track on this page a sketch of a performance that hasn't happened yet.



Figure 13: Screenshot of the web page archive "Not dead but not alive".

On the right, you can click to visit the archive from Dusk of an electric forest, In the middle you can find City Landscapes. On the left is the electromagnetic field composition Bonus track.

There is a button in the top right corner for finding more information. In the middle, you can activate the sound. In the top left corner, there is a button that shows the code for the visual. My idea is that you will understand this text better, and you will also have the opportunity to experiment with the code that I am sharing with you. In the best-case scenario, it may also inspire you to see how it works and give you the opportunity to modify it.

I have written about my own methodologies and events, and in this section, I want to make the archive more tangible. I'm aware that this documentation is really

incomplete, that it works like a photograph, just capturing a precise moment from a whole event. Here I had focussed on performance practices which have an intricate relation with time and ephemeral, therefore is practice that face challenges in terms of archiving.

Performer artist Stuart Brisley points to the temporal paradox on performance: “The issue is not one of the ephemeral versus the permanent. Nothing is forever. It is the question of the relative durations of the impermanent”. A performance has qualities of singular, irreducible, unrepresentable but still the event will be repeated in thought, record and memory.[13]

Understanding of performance lives in two instances: one where the event unfolds and brings different senses and meanings to everyone present, and another where a sense of knowing is built through revision. This understanding can only be grasped by inhabiting the specific qualities of duration and time. The second instance is the after-moment, where revision occurs and a sense of knowing and reflection is built.[14]

So, think of this section as a small journal of processes. Here, the grammar is a fragment of code running inside this page.

Bibliography

- [1] Alan F Blackwell, Emma Cocker, Geoff Cox, Alex McLean, and Thor Magnusson. *Live coding: a user's manual*. MIT Press, 2022.
- [2] Matt Pearson. *Generative Art*. Manning Publications Co., USA, 2011.
- [3] Alex McLean, Julian Rohrhuber, and Renate Wieser. The meaning of live: From art without audience to programs without users, April 2023.
- [4] Thor Magnusson. Herding cats: Observing live coding in the wild. *Computer Music Journal*, 38(1):8–16, 2014.
- [5] Ursula Franklin. *The real world of technology*. House of Anansi, 1999.
- [6] Nick Collins. Live coding of consequence. *Leonardo*, 44(3):207–211, 06 2011.
- [7] Golan Levin and Tega Brain. *Code as creative medium: a handbook for computational art and design*. MIT Press, 2021.
- [8] François Ribac. Bernie krause's the great animal orchestra—an exhibition. *Sound Studies*, 2(2):201–204, 2016.
- [9] Bernie Krause. *Wild soundscapes: discovering the voice of the natural world*. Yale University Press, 2016.
- [10] Gloria Anzaldúa. *Borderlands/La frontera: la nueva mestiza*. Capitán Swing Libros, 2021.
- [11] Emma Cocker. Performing thinking in action: the melete of live coding. *International Journal of Performance Arts and Digital Media*, 12(2):102–116, 2016.
- [12] AMRO Art Meets Radical Openness. Amro festival dedicated to art, hacktivism and open culture, 2024.
- [13] Kiff Bamford. Working with fragments: A performance art's archive. *Journal of Writing in Creative Practice*, 7(3):557–567, 2014.
- [14] Amelia Jones and Adrian Heathfield. *Perform, repeat, record: Live art in history*. Intellect Bristol, 2012.