How Live is Live Coding? The case of Tidal's Longest Night.

Raphaël Maurice Forment
ECLLA, Université Jean Monnet (UJM) - Saint Étienne
raphael.forment@gmail.com

Alex McLean
Then Try This, Sheffield/Penryn
alex@slab.org

ABSTRACT

From the 21st to the 23rd of December 2021, more than one hundred live coders from the TidalCycles [1] community and beyond coordinated to produce non-stop algorithmic audiovisual performances broadcasted openly on streaming platforms [1] . For the analyst or software designer, this event represents an extraordinary opportunity in that it allows to study a wide and diverse range of performers incorporating live coding techniques in their work from around the world. In this article, we present a discussion about the particular *liveness* [2] of live coding based on our observations of these performances. We focus on code editing practices and on the general approach to the medium – the text editor and the source code – to better understand how live coders from the event approached the performing act and how they designed live performances through the mediation of code. More specifically, this article explores the reasons that could explain why, despite the fact that live coding libraries such as TidalCycles are built to encourage free improvisation and free exploration of musical patterns, the majority of live coders do not start their performance *from scratch*.

1. INTRODUCTION

The practice of live coding in musical and audiovisual performance is now an established part of computer music research. More than two decades of research and artistic creations have lead to popularisation of live coding techniques among a larger audience thanks to robust audio coding platforms such as SuperCollider [3], Max, Pure Data, ChucK [4], ExTempore [5] and to various musical libraries or software exploring different models of live computation (ORCA [2] , Sonic Pi [6], Gibber [7], TidalCycles, etc...). This popularity emerged from the founding of the TOPLAP collective in 2004 [8], and subsequent initiatives such as the live.code.festival in Karlsruhe, /* vivo */ festivals in Mexico City [9], and Algorave algorithmic dance music events founded in London and quickly spreading across hundreds of cities and festivals worldwide [10].

Musical live coding can now be found in academia [11] as a tool for experiments on new instruments, digital art, computer languages, HCI and pedagogy among many fields of musical, graphical and choreographic research. It has also garnered the interest of enthusiast electronic musicians willing to rethink the role of computers in modern music-making setups, searching for ways to complement – or supplement – complex hardware and/or digital audio workstations. Beyond the numerous possibilities of use, live coding is also perceived by its community as a codified but flexible approach to electronic-music making promoting open-source initiatives, collaboration and values usually associated with the *hacker* ethic and FOSS (Free/Open Source Software) communities [12].

The event we hereby study is by no means unusual for its duration and scale in the live coding world. Indeed, the TOPLAP 15th anniversary (February 19th 2019 [3]) or New Moon (August 19th 2020 [4]) could have provided similar corpuses with even more performances to analyse. This could easily be explained by the fact that having deep roots in *net-art*, creative coding and underground computer music scenes, live coding has always used the internet as the medium *by default* for exchanging knowledge, ideas, software and to create digital meeting grounds. Years before the systemic shift caused by the COVID-19 pandemic and its impact on live music performances, live coding related research already held interest for the topic of network collaboration and synchronisation at its core: Troop [13], Flok [5] , Estuary [14].

In this article, we draw on the mass of performances from the Longest Night stream, in order to nourish a discussion about the presentation of *liveness* in live coding practices and about how different approaches to the medium lead to very different artistic outcomes and performance practices. While all live coders share a common appeal for *liveness* (as far as we know there were no pre-recorded performances during the stream), how the ultimate form in which coding takes place on stage can dramatically shift, from complex pre-written performances to from-scratch improvisation of sound synthesis algorithms.

[1] https://night.tidalcycles.org/: event website.

[2] https://github.com/hundredrabbits/Orca : By David Mondou-Labbe, an *esoteric programming language designed to quickly create procedural sequencers*, following an unusual bi-directional programming paradigm similar to the Befunge programming language.

[3] https://toplap.org/toplap-15th-anniversary-stream-14-17th-february-2019/: event link.

[4] https://sun.tidalcycles.org/: event website.

[5] https://github.com/munshkr/flok: *Web-based P2P collaborative editor for live coding music and graphics*, by Damián Silvani.

2. EVENT DESCRIPTION

Figure 1. Longest Night flyer, made by Luluganeta, in charge of the graphical chart.

The *Longest Night* was so called because it took place over the weekend of the December solstice, the moment when the earth finishes one oscillation and begins another. The focus on the solstice is therefore intended to unify an international line-up of people contributing performances (although of course in the Southern hemisphere, this was the *shortest* night). This was a volunteer-run event, organised by members of the international TidalCycles community: Bernard Gray (*Cleary*), Thomas Grund (*MrReason*), Scott Fradkin (*Sfradkin*), Renzo Torrisi (*Ritchse*) and many others active under nicknames: *paulchannelstrip*, *luluganeta*, *Givo29*, *kit-christopher*, *arethusa*, etc... The event emanated from a community with close ties to the software, almost all of the performances were making at least some use of Tidal [6]. Nonetheless, some performers have chosen to use other live coding environments/software such as Gibber, Sonic Pi, ORCA or even, in some cases, custom-made tools and libraries.

The constraints linked to the organisation of an international remote event were handled by leveraging several web technologies, the most important one being *muxy* [7], a server-side tool for multiplexing video streams developed by Tidal community member Damián Silvani, with a web front-end developed for the event by Thomas Grund. A great deal of preparatory work was done to automate the inscription and distribution of stream tokens across multiple time zones, allowing a performer to broadcast and perform during a strict 20 minute slot, before an automatic relay is established with the next performer. There was no vetting of performances, anyone could freely sign up for a performance slot. Technical monitoring and archive recording was ensured for the total duration of the event. Documents and details related to the organisation and technical management were compiled and stored by Bernard Gray to ensure the later reproducibility of the event [8].

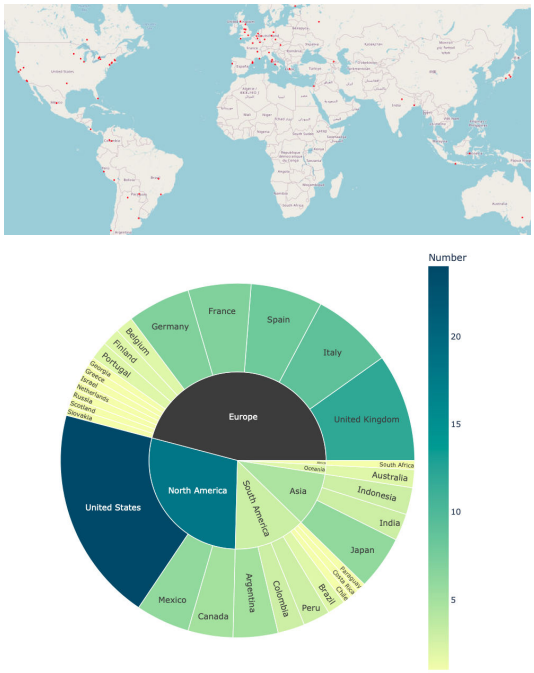

Figure 2. Approximated location of Tidal's *Longest Night* performers based on personal stream descriptions.

With the organisation team taking care of such technicalities, performers were left with the task to connect using audiovisual streaming software and were interacting with each other through various chat systems: YouTube, Discord, Telegram, Twitch or personal communications on social media. Communication could also happen directly through code, the artists using code comments as a way to communicate while performing or even, in some cases, chats directly integrated in the editing environment.

3. METHODOLOGY

An access to recorded archives of the event was provided by Bernard Gray, and has since been archived on Archive.org as well as Youtube [9]. Due to the uncertainties of internet networks, and some failures to connect and perform, nine of the total of 120 performances are missing (7.5%). The accessible footage represents approximately 50Gb of raw unedited video files. The quality of each video varies according to the broadcast and encoding format chosen by each artist. It can sometimes present audiovisual artefacts related to the transmission (missing sound, glitches). Rarely, videos have been archived even though they are, accidentally, partially truncated. Incomplete performances have been included in our analysis, as long as they communicate the gestures and techniques of a given live coder.

[6] We here use *Tidal* as a diminutive for all TidalCycles related live coding environments: TidalCycles, David Ogborn's MiniTidal, etc...

[7] `https://github.com/munshkr/muxy`: GitHub project.

[8] `https://drive.google.com/drive/folders/1JgCY6M0X4uab__HbVFG8ebvTgIMUXfRZ`: organisation documents, courtesy of Bernard Gray.

[9] See `https://archive.org/details/toplap` and `https://youtube.com/eulerroom`: Volunteer work from Scott Fradkin, assuming the role of an unofficial community archivist.

Strategies had to be found to deal with the profusion of footage to consider (112 * 20 minutes performances). A table – whose structure is detailed below – was used by Raphaël to compile generic information about each performance. It has been laid out as a quantitative analysis tool, allowing us to gather basic and general thematic information about a large number of performances (*see table*). As such, notes have been taken on various general topics as a ground for later inductive reasoning, confronting our experiences and potential biases to quantitative observations. The document will serve, for the discussion, solely as a reminder and visual help. In opting for this method, we hoped not to focus too keenly on specific performances or topics but rather to uncover general trends and patterns in the way the performative act is being approached by the Tidal community. Data from the event will be used as a basis for broader discussion.

Studying performances without the explicit consent of performers leads to ethical concerns. To address this, we have drawn only anonymised, general conclusions about our observations of the field. We only use data to the extent that it identifies a statistical/generic group of users sharing a specific approach to live coding. Live coders using their own tool will be identified only where their work has been made public through code source releases, academic papers or public communication.

	— Identification —
Priv.	Artist and description
Public	Location (country/town), lost/recorded, audio/visual, solo/collaborative
	— Code Editing —
Public	Text editor, editing paradigm (regular, Emacs, Vim, custom), editing speed, code comments, lisibility
	— Customisation —
Public	Bespoke tools, libraries and extensions. Physical controllers, external instruments and synthesizers
	— Musical Content —
Public	BPM, harmonic, melodic, rythmic and timbral features. Short musical description

Figure 3. Diagram of the observation grid used for each performance slot, thought to be respectful of anonymity.

Some biases in our analytic approach can already be acknowledged and considered. Naturally, studying performances on a surface level, focusing on a performer's interactions with live coding environments impair the possibility to provide deeper and nuanced example-based analyses, detailing how, for each performer, interaction relates to an artistic practice. We accept, for the time being, that this is not the scope of this article. Our intention, hereby, is to review and ponder on current live coding practices, focusing on the larger and *public* picture as opposed to the details. By consequence, it must be dully noted that we have consciously chosen to focus almost exclusively on observing the editing technique and manual execution of the performance, and certainly do not attempt to make value judgements on the code or sonic materials (audio samples, patches) being used or the musical outcomes.

One must also acknowledge our own involvement in the TidalCycles community, as the instigator of the TidalCycles project and its community Alex, as a performer, technical documentation editor and doctoral student for Raphaël. The writing of this article is motivated by the desire to deepen the understanding of the practice of live coding. Confronting TidalCycles as a research product [15] allows us to observe some of the complex interactions between technology and community and the emerging craft practices that emerge to bind them together.

4. PARADOX AND QUESTIONS

Historically, live coding has been presented as a scene for improvisation and experimentation in computer music performance [16] [17], proposing an alternative to skeuomorphic software designs and closed-source black-boxes (such as Digital Audio Workstations (DAWs), commercial software and digital synthesizers). This proposition was explored through a cathartic return to the simplicity and affluence of plain text, programming languages and grammars [18]. From the perspective of HCI (Human-Computer Interaction) research, live coding was used as a means to circumvent idiomatic patterns in music technology, such as the omnipresent piano-roll or timeline representation, reminiscent of early *musique concrète* or electro-acoustic studio practices that were sometimes felt as hindrances to direct musical expression on the laptop. More explicitly, live coding research has been focusing on exploring the computer – here considered as a fully-fledged musical environment – and computation itself as a valuable live musical expression framework, considering the control structures and internal logic of programs as compositional tools and suitable mental environments for the live performer [19] [4].

However, the provocative title of this article points to an apparent paradox: if live coding is an improvised practice, why do many performers not work *from scratch*? Approximately one quarter of performances during the *Longest Night* stream played from scratch, with the majority instead working on a continuum from working with a well-prepared musical structure, to adapting pre-prepared code, or to tweaking fully composed tracks. We approach this paradox to try to understand the nature and importance of improvisation in this particular art form.

Our analysis will be split in two sections: the first one exploring from-scratch practices, before considering the other largely dominant approaches to live coding performances. In doing so, we will argue that idiomatic patterns in the usage of live coding libraries or text editors are emerging which influence the way improvisation is approached by live coders.

5. ANALYSIS: NUANCES OF LIVENESS IN OBSERVED LIVE CODING PRACTICES

5.1 The Historical Promethean Live Coder

The liveness underpinning *live coding* as an art form and practice can be seen in the proclamations given by the 2004 TOPLAP "draft manifesto" for live coding. Although originally authored in a haphazard manner via largely anonymous contributions to the TOPLAP wiki [20], this document has often been used by musicians and artists as a practical guide to understand the gesture and spirit that unites the live coding scene, although more tempered and measured documents have since been published (*e.g.* the community guidelines for holding an Algorave [10]). The manifesto's significance for live coding development and legacy can easily be measured by the number of academic publications referring to it as the *de facto* introduction to the topic, despite its satirical, emphatic and at times self-contradictory style. The manifesto calls openly for the "access to the performer's mind, to the whole human instrument" and to the practice of live exploration of algorithms: "Insight into algorithms [...] No backup", stressing the necessity to lay bare the inner workings of the music being performed: "Show us your Screens [...] Code should be seen as well as heard, underlying algorithms viewed as well as their visual outcome" in order to invite musicians to expose, on the same level, their instruments, gestures (key strokes) and musical thoughts (algorithms). We should note that while this is ostensibly a *draft* manifesto, and is still hosted in a wiki form editable by anyone and therefore adaptable to changing practices, it has not been significantly edited for many years.

Contrary to the Algorave guidelines, the TOPLAP manifesto also emphasizes the "manual dexterity", "mental dexterity" and the thrilling danger associated with the act of live algorithmic programming, inviting musicians to take risks. This Promethean live coder figure, the one of the experimentalist, might be a key explanation to the global perception among researchers and artists alike of live coding as a particularly experimentation and improvisation-driven art form dedicated for computer savvy programmers/musicians. However, the reality is often more complex, and veterans or beginners alike can be brought to try coding from scratch in an event of the nature of the *Longest Night*. Knowledge of this art form's history and legacy is not of paramount importance for many, and from-scratch is not appearing as a significant marker between experienced veteran live coders and new users.

During the event, 24.8% (28 out of 113) of the slots were starting from scratch, with musicians presenting a blank slate on their text editor without any visible backup or prepared materials [11]. From-scratch has been attempted by solo and synchronised performers, both for audio and for video live coding, most often using Olivia Jack's JavaScript-based Hydra visual synthesizer [12]. Multiple environments have been used for this purpose including Tidal, SuperCollider (also hosting Tidal's sound engine, SuperDirt [13]), Sam Aaron's Sonic Pi [6] and bespoke live coding environments.

5.1.1 *From Scratch? How and Why?*

From scratch performances could perhaps be perceived as echoes of the original TOPLAP manifesto, with artists learning to adhere to the radical form of improvisation it described, a position embodied in the past by artists such as Click Nilson, and studied by Andrew Sorensen [21]. However, it is interesting to note that during the Longest Night, from-scratch performances were particularly prominent in performers from Latin America, continuing the strong community practice that developed largely independently in Mexico City [9]. While TOPLAP was initially founded in Europe, it was Mexico City that hosted the first regular community live coding meetings, where participants placed greater importance on from-scratch performance than their European counterparts through 9-minute performances sometimes referred to as "Mexican roulette".

Among the cohort of *Longest Night*'s performers, from-scratch live coders are the ones who correspond the most to the stereotypical figure of the *Promethean live coder*, but are also, by many aspects, the minority. That said, the dividing line between from-scratch coding and other forms of live coding all along the event is, more often than not, difficult to draw. At most, this form of performance can be described as a performative intention, forcing the musician to type actively. Editing speed, familiarity with efficient code editing practices (indentation, code snippets, fast navigation), deep knowledge of the libraries or languages being used (custom functions or DSP-oriented code) were not key factors motivating musicians to engage in this demanding performing technique, for novice users seemed to try their hand at it.

For the specific case of Tidal, we argue that it is quick to learn, where beginners quickly reach musical results [22]. Nonetheless, from scratch performances on Tidal, even for the observed veterans, tended to be more idiomatic, centered around the software's functionality for algorithmic manipulation of rhythmic patterns. These performances are necessarily more open to particular economic patterns of use: emphasis on repetition, creative audio mangling, interference between combined sequences, as well as fast tempos and minimal usage of parametric envelopes or complex audio effects. This observation is analogous to ones made by Thor Magnusson, presenting DMIs as "epistemic tools" embodying their own musical theory and their own usage patterns [23].

5.1.2 *Celebration and collaboration*

From-scratch collaborative live coding is a practice that has always been present in the community [24], and was explored during 7 different performance slots for networked collaborative jams. These slots were occupied by performers willing to celebrate the Longest Night by coding among

[10] `https://github.com/Algorave/guidelines/blob/master/README_en.md`: Algorave collaborative guidelines.

[11] We took into account performers willing to start from an initial bare-bone skeleton as long as it stays basic in design.

[12] `https://github.com/ojack/hydra`: GitHub repository.

[13] `https://github.com/musikinformatik/SuperDirt`: Tidal audio engine, GitHub repository.

Figure 4. A new created live coding session on David Ogborn's Estuary platform, with six performer slots.

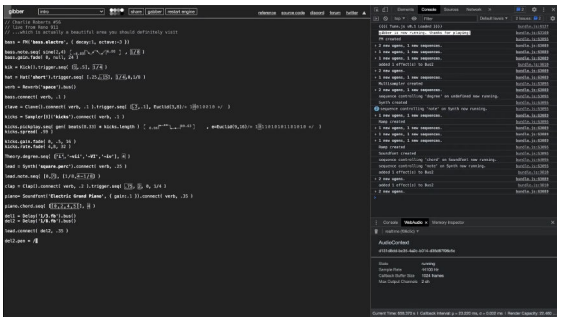

Figure 5. Charlie Robert's performance using Gibber, a JavaScript browser-based audio-visual live coding environment from-scratch during Tidal's *Longest Night*. [7]

others using David Ogborn's MiniTidal and Estuary platform [14], or by connecting privately using Flok (TypeScript and NodeJS peer-to-peer collaborative editor) or the Python-based Troop [13].

In this context, we observed that the emphasis on fast typing speed and occasional angst associated with solo from-scratch performance tend to fade. The prevalence of the format might find sources in the fact that it exposes musicians to a particularly joyful form of collaborative improvisation typically found in many other improvised musical cultures and circles, here redeployed for code-based performance. Single performers alone can already – if they do not refrain from it – have a dramatic impact on the music being generated (long samples, high gain, rhythmic density, etc...). This leads to musical behaviours already well known such as call and response, different 'band-members' focusing on different timbres/voices, or collaboration towards building complex sonic or visual algorithms. From-scratch collaborative coding exposes the musician to a group dynamic, where personal artistic intent or even self-identity as a performer is temporarily dissolved, encouraging experimentation, radical improvisation and a touch of letting go. The possibility of sharing the program state and tempo with other musicians appear to slowly lead to the establishment of this form of live coding performance as a separate sub-genre, typically showcased in networked events like the *Longest Night*. However, as is the case with environments not previously bent towards specific musical practices and styles, a certain factor of homogenisation is notable, given the limited pool of available audio samples, synthesizers and local networking capabilities given by the containerised web application.

5.1.3 Solo From Scratch

Twenty one of the from-scratch slots were occupied by solo performers, an approach to live coding that is perhaps most commented on in the literature despite being in the minority of performances during events like the Longest Night. Challenges associated with this practice are pertaining to the creation, in limited time/energy, of a complete performance complex enough to hold the audience's attention. It is also the one that more deliberately exposes the musician to errors, bugs and to the "decoupling of embodiment" [25] associated with writing code for further evaluation. On the other hand, it is also appreciated as a form of expression that offers chance events and surprises to occur, with improvisers inviting the audience to enter into an impromptu exploration of system affordances. It is not surprising to acknowledge that the technique has been used, during the *Longest Night*, to produce a wide range of distinct sonic results: ambient sample-based landscapes, breakcore/IDM music, free-form improvisation, noise, live DSP or external synthesizer-based performances. Adding to the variety of music, this form of from scratch allows live coders to showcase custom environments, or detach themselves from the most idiosyncratic type of usages, as seen through Alex performance using Vortex (Python-based Tidal rewriting) or Charlie Roberts using Gibber (Fig. 5).

As previously said, the diversity of profiles starting from-scratch exceeds our initial expectations. These performers share few common characteristics, whether it is by the environment or the interface used (from 'vanilla' to heavily modified or custom), by the nature of the code itself (musical patterns and/or lower-level DSP generative code) or by the degree and speed of interaction with code (from hardly any interaction at all to extreme velocity of edits). It is true that from-scratch coding appear to be of interest for live coders deeply versed in live digital signal programming (e.g. with JITlib [26]), hacking of custom live coding systems [27] and control of digital or hardware setups. It is also true to note that a greater familiarity with coding, programming languages, complex sound synthesis software (such as SuperCollider and its JITLib library, perhaps the most common system used for synthesis in live-coding) and text editors can be found among performers attempting this type of performance. Nonetheless, they only represent a small percentage of the performances being displayed during the *Longest Night*.

5.1.4 From scratch: some conclusions

Our observations reveal the participation of a surprisingly broad range of from scratch practices. The dangers and technical issues naturally arising from such performances, incarnated by the mythical figure of the Promethean live coder, did not act as a hindrance but, on the contrary, may have encouraged a number of musicians to embrace live

coding as a particularly radical form of computer-based musical experimentation. We were particularly surprised by the various technical profiles of musicians who lend themselves to the exercise, the practice not being limited to the most experienced live coders or computer programmers.

This phenomenon might be explained by the prominence given by live coding language designers and by live coding research in general to the development of terse and rich DSLs [14] for music expression, allowing temporal semantics, generative sequencing capabilities and precise parameter control to be expressed in a terse and legible manner. From-scratch is often made possible by the close coupling between live coding libraries and bespoke audio backends, alleviating the need for complex preparation and technical setup [15] , we noticed a stronger tendency towards musical homogenisation regarding performances produced using the from scratch non-prepared environments, linked to idiomatic patterns of software usage and livecoders relying on reflexes and procedural memory [28].

5.2 Distributing Liveness

5.2.1 Algorithmic gardeners

The rest of the performances (*approx.* 75%, 85 out of 113) featured prearranged setups and musical/video materials. The nature of the live interaction in these performances differed greatly compared to the from-scratch performers, whether they were coding collectively or solo, audio or audio/visual performances. However, it does not follow that these artists were not live coders. Rather, we could argue that this majority of the cohort prefer to distribute liveness to different stages of their work, that is, to disseminate live coded interactions more privately, across composition and preparation as well as music performance, one example being the showcase of *Wags*, a new Tidal-inspired but not Haskell-based live coding environment, by Mike Solomon [29]. This suggests a nuanced and difficult to pinpoint spectrum of live coding practices, ranging from almost totally fixed performance to paving a way, beforehand, to a more personal improvisational process, in a safer and perhaps less threatening private space.

The potential of a compositional and / or preparational phase to live coding connects it with the world of more classical generative and procedural music, where code is written and then left to produce music autonomously. In conversation with David Toop, Brian Eno explains that "generative music is like trying to create a seed". He relates generative music to gardening, "it responds to conditions during its growth and it changes and it's different every year." [30]. The assumption here is that the algorithms generating music are akin to nature, but are separate from us. Concerning live coding, one should be cautious with the generative-gardening analogy, which could be seen as separating the author from their work, where the former is cast as the genius designer of the latter.

[14] *Domain Specific Language*: specialized computer language for a particular application domain (*i.e.* algorithmic music, live expression).

[15] In particular, one could argue that the live coding library itself acts as a non-neutral *prepared code* for the live-coder to perform with.

In Eno's notion of generative music, there is distance between musicians and their algorithms. However, the gardener is also alive in the garden, working in the weather, never sitting back to watch plants grow. In *Farming as performance*, Dominic Glover builds a view based on the observation that a crop is not the outcome of a plan [31]. The farmer does not simply plant a seed and sit back. Rather, a crop is the result of an improvised performance, with both plants and farmers as performers, where any plan evolves over time. He follows the observation of Paul Richards [32] that the eventual layout of a field is not a design, but rather a partial historical record of continual responses that the farmer and their crops make to changing weather. When a live coder brings pre-written code a performance, we could see that code as a similar, living historical record.

Live coding performers, even for the case where prepared materials or setups are presented, challenge the assumptions of generative music by working hands-on with code, while it runs. At any point, the state of the code gives a snapshot of the music being generated at the present moment, allowing a performer to react through code comments, and a few keypresses might result in dramatic shifts in musical structure. By performing with code in this way, the live coder is compelled into humility, because just as with the weather, or plant life, code operates in ways and at scales beyond human control. The human performer develops heuristics for working with their code, but in a very real sense, the code also acts upon them. Prepared performances from Tidal's *Longest Night* showcase a type of improvisation much akin to what, in other terms, is described by Palle Dahlstedt as "systemic improvisation" [33].

Glover's analogy of farming-as-performance throws light on comparisons of from-scratch and prepared performances. Whereas from-scratch performers like to plant and interfere with the genetics of a seed while it grows, pre-prepared live coders instead like to bring plants to the party, and see how they fare in the performance environment, here akin to the farmer responding to weather. The key difference between algorithmic/generative music and distributing liveness in live coding performances is therefore to be found, in our opinion, in what the Longest Night allows us to observe: the observable presence of source code as a communicative medium, as a social link between the musician and their community, and as a testimony of one's implication in the unfolding algorithmic process.

5.2.2 Handling obstacles to live algorithmic performance

There are also pragmatic potential reasons for the *Longest Night* musicians tendency to pre-prepare musical materials, which go beyond and sometimes influence, afterhand, the aesthetic discourse on the practice. We observed that most of the performers belonging to this category tend to consider algorithmic code as a malleable interface, limiting their interventions to actions such as commenting lines, editing variables, crafting transitions between patterns or making terse edits to previously written materials. Reasons for doing so could be related to relative musical inexperience; limited by text editing speed, by the fear of running into bugs and performance-stopping issues, or to being ex-

posed to complexity not manageable in realtime.

On the other hand, an experienced musician may simply be unwilling to give up the levels of control they are used to in more conventional software (DAWs, mixing/mastering tools), wanting to spend significant time mastering samples and adjusting the mix of their otherwise live coded compositions before performing them. For most, our external observations are insufficient to assert the true intention behind the artists' actions, and further investigation is needed, through field interviews.

One other area for future investigation is the amount of customization which artists apply to different parts of the default Tidal system of text editor, pattern language and sound synthesizer and samples. Some take time to get deep into the default configuration of the Atom programmer's editor, working with more advanced features such as multiple cursors when coordinating a change across multiple patterns at once. Others explore alternative approaches to text editing [34], picking more obscure editors, or experimenting with live visualisation. The influence of the personalisation of the work environment on music playing for the case of the live coding scene has, until now, been too rarely addressed.

Due to the open and welcoming nature of the Longest Night event, many participants will be doing their first ever live coding performance, or even first ever performance of any kind, and still managing to find new juxtapositions of sounds and patterns transformations in the default install and sample set. There could be some concerns that the large number of performances which appeared visually similar, where artists have chosen to keep similar editor themes, could run counter to live coding's promoted ideals for freedom from constraints, that we saw in the TOPLAP manifesto. Nonetheless, the combinational possibilities offered by a live coding environment like Tidal, which offers a large number of pattern transformations which can be quickly combined to find new possibilities, means that a singular shared performing environment does not always result in similar musical results.

Pre-arranged performances from the Longest Night can prove particularly disconcerting for the analyst, as they reveal a whole spectrum of performing practices ranging from private studio work presentation to loosely arranged improvisation sets. The nature of this performing art form pushing musicians to work, perform and share with the public using mostly plain-text, improvisational and studio practices often appear blended in plain-sight, something that is, to our knowledge, not often the case for computer-based audiovisual performances.

6. CONCLUSIONS

As can be seen from our observations from the *Longest Night*, the divide between from-scratch performances and pre-arranged performances only acts as a surface typology, usable for work purposes only. From an analytic standpoint, its validity holds to the mere fact that it allows us to uncover, upon closer examination, finer distinctions between different user groups sharing the same approach to source code and live performance.

While a certain taste for liveness, demonstration and algorithmic unfoldings unites the field of live coding performance, an event of this sort appear to be a good tracker to track the evolution and diversity of live coding practices, starting from the idealized 2004 promethean live coder figure to the contemporanean and much different networked live-coder here observed during the event. Further investigation centred around the question of liveness (*e.g.* Auslander [35], Emmerson [36], Croft) will undoubtedly help us to include live coding performances in the larger field of electronic music performance.

Acknowledgments

I (Raphaël) would like to thank Laurent Pottier (ECLLA, Université de Saint Étienne) and Alain Bonardi (MUSIDANSE, IRCAM, Université Paris 8) for their advices and for their trust in this research project. I would also like to thank the 3LA doctoral school (ED 484) from the University of Lyon and my laboratory, the ECLLA from the Université of Saint Étienne, for funding this research. I would also like to thank Alex McLean for diving with me in the complex issue of analysing live coding practices.

I would like to thank Raphaël for inviting me to contribute to this paper and for his tireless support of the TidalCycles community. My work here was supported by a UKRI Future Leaders Fellowship [grant number MR/V025 260/1].

7. REFERENCES

[1] A. McLean, "Making programming languages to dance to: live coding with tidal," in *Proceedings of the 2nd ACM SIGPLAN international workshop on Functional art, music, modeling & design*, 2014, pp. 63–70.

[2] P. Rein, S. Ramson, J. Lincke, R. Hirschfeld, and T. Pape, "Exploratory and live, programming and coding," *The Art, Science, and Engineering of Programming*, vol. 3, no. 1, pp. 1–1, 2018.

[3] S. Wilson, D. Cottle, and N. Collins, *The SuperCollider Book*. The MIT Press, 2011.

[4] G. Wang, P. R. Cook, and S. Salazar, "Chuck: A strongly timed computer music language," *Computer Music Journal*, vol. 39, no. 4, pp. 10–29, 2015.

[5] A. C. Sorensen, "Extempore: The design, implementation and application of a cyber-physical programming language," Ph.D. dissertation, College of Engineering and Computer Science, The Australian National University, 2018.

[6] S. Aaron and A. F. Blackwell, "From sonic pi to overtone: creative musical experiences with domain-specific and functional languages," in *Proceedings of the first ACM SIGPLAN workshop on Functional art, music, modeling & design*, 2013, pp. 35–46.

[7] C. Roberts and J. Kuchera-Morin, "Gibber: Live coding audio in the browser," in *ICMC*, vol. 11, 2012, p. 6.

[8] A. Ward, J. Rohrhuber, F. Olofsson, A. Mclean, D. Griffiths, N. Collins, and A. Alexander, "Live algorithm programming and a temporary organisation for its promotion," in *readme - Software Art and Cultures*, Jan 2004, pp. 243 – 261.

[9] H. Villaseñor-Ramírez and I. Paz, "Live Coding From Scratch:The Cases of Practice in Mexico City and Barcelona," in *Proceedings of the 2020 International Conference on Live Coding (ICLC2020)*. Limerick, Ireland: University of Limerick, Feb. 2020, pp. 59–68.

[10] N. Collins and A. McLean, "Algorave: Live performance of algorithmic electronic dance music," in *Proceedings of the International Conference on New Interfaces for Musical Expression*, 2014, pp. 355–358.

[11] A. McLean and R. T. Dean, *The Oxford handbook of algorithmic music*. Oxford University Press, 2018.

[12] C. Alexandra, "What is algorave?" https://media.ccc.de/v/rc3-2021-chaoszone-345-what-is-algorave, Dec 2021.

[13] R. Kirkbride, "Troop: a collaborative tool for live coding," in *Proceedings of the 14th Sound and Music Computing Conference*, 2017, pp. 104–9.

[14] D. Ogborn, J. Beverley, L. N. del Angel, E. Tsabary, and A. McLean, "Estuary: Browser-based collaborative projectional live coding of musical patterns," in *International Conference on Live Coding (ICLC)*, vol. 2017, 2017.

[15] A. McLean, "Research products," Mar 2021. [Online]. Available: https://slab.org/research-products/

[16] K. Burland and A. McLean, "Understanding live coding events," *International Journal of Performance Arts and Digital Media*, vol. 12, no. 2, pp. 139–151, 2016.

[17] N. Collins, "Live coding of consequence," *Leonardo*, vol. 44, no. 3, pp. 207–211, 2011.

[18] D. Stowell and A. McLean, *Live Music-Making: A Rich Open Task Requires a Rich Open Interface*. London: Springer London, 2013, pp. 139–152.

[19] T. Magnusson, "Algorithms as Scores: Coding Live Music," *Leonardo Music Journal*, vol. 21, pp. 19–23, 12 2011.

[20] C. Haworth, "Technology, Creativity and the Social in Algorithmic Music," in *The Oxford Handbook of Algorithmic Music*, ser. Oxford Handbooks. Oxford University Press, pp. 557–582.

[21] A. R. Brown and A. Sorensen, "Interacting with generative music through live coding," *Contemporary Music Review*, vol. 28, no. 1, pp. 17–29, 2009.

[22] A. McLean and R. Bell, "Pattern, code and algorithmic drumming circles," in *Proceedings of the Fourth International Conference on Live Coding*. Madrid, Spain: Medialab Prado / Madrid Destino, Jan. 2019, p. 353. [Online]. Available: https://doi.org/10.5281/zenodo.3946174

[23] T. Magnusson, "Epistemic tools: the phenomenology of digital musical instruments," Ph.D. dissertation, University of Sussex, May 2009. [Online]. Available: http://sro.sussex.ac.uk/id/eprint/83540/

[24] J. Rohrhuber, A. de Campo, R. Wieser, J.-K. van Kampen, E. Ho, and H. Hölzl, "Purloined Letters and Distributed Persons," in *Music in the Global Village Conference 2007*.

[25] S. Knotts, "Live coding and failure," *The Aesthetics of Imperfection in Music and the Arts: Spontaneity, Flaws and the Unfinished*, p. 189, 2020.

[26] J. Rohrhuber and A. de Campo, "Just in time programming," *The SuperCollider Book. MIT Press, Cambridge, Massachusetts*, 2011.

[27] B. Bacot and C. Canonne, "Musique et hacking : de l'éthique aux pratiques."

[28] T. Sayer, "Cognition and improvisation: some implications for live coding," in *First International Conference on Live Coding (ICLC2015)*, 2015.

[29] M. Solomon, "Functional reactive programming and the web audio api," in *Proceedings of the International Web Audio Conference*, ser. WAC '21, L. Joglar-Ongay and S. et al., Eds. Barcelona, Spain: UPF, July 2021.

[30] D. Toop, *Haunted Weather: Music, Silence, and Memory*, main edition ed. Serpent's Tail, 2005.

[31] D. Glover, "Farming as a performance: A conceptual and methodological contribution to the ecology of practices," vol. 25, no. 1. [Online]. Available: http://journals.librarypublishing.arizona.edu/jpe/article/id/2066/

[32] P. Richards, "Agriculture as Performance," in *Farmer First: Farmer Innovation and Agricultural Research*, R. Chambers, A. Pacey, and L. A. Thrupp, Eds. Intermediate Technology Publications, pp. 39–42.

[33] P. Dahlstedt, *Action and Perception: Embodying algorithms and the extended mind*. United Kingdom: Oxford University Press, 2018, pp. 41–66.

[34] J. Armitage, A. McPherson *et al.*, "The stenophone: live coding on a chorded keyboard with continuous control," in *International Conference on Live Coding*, Morelia, Mexico, 2017.

[35] P. Auslander, *Liveness: performance in a mediatized culture*, 2nd ed. London ; New York: Routledge, 2008, oCLC: ocn144770575.

[36] D. Peters, G. Eckel, and A. Dorschel, Eds., *Bodily expression in electronic music: perspectives on reclaiming performativity*, 1st ed., ser. Routledge research in music. New York: Routledge, 2012, no. 2, oCLC: ocn666242840.