

Discovering Creative Commons Sounds in Live Coding

ANNA XAMBÓ SEDÓ 

Music, Technology and Innovation – Institute for Sonic Creativity (MTI²), De Montfort University, Leicester, UK
Email: anna.xambo@dmu.ac.uk

This article reports on a study to identify the new sonic challenges and opportunities for live coders, computer musicians and sonic artists using MIRLCa, a live-coding environment powered by an artificial intelligence (AI) system. MIRLCa works as a customisable worldwide sampler, with sounds retrieved from the collective online Creative Commons (CC) database Freesound. The live-coding environment was developed in SuperCollider by the author in conversation with the live-coding community through a series of workshops and by observing its use by 16 live coders, including the author, in work-in-progress sessions, impromptu performances and concerts. This article presents a qualitative analysis of the workshops, work-in-progress sessions and performances. The findings identify (1) the advantages and disadvantages, and (2) the different compositional strategies that result from manipulating a digital sampler of online CC sounds in live coding. A prominent advantage of using sound samples in live coding is its low-entry access suitable for music improvisation. The article concludes by highlighting future directions relevant to performance, composition, musicology and education.

1. INTRODUCTION: SOUND-BASED LIVE CODING AND AI

Live coding was initially characterised as a new form of expression in computer music based on ‘real-time scripting during laptop music performance’ (Collins, McLean, Rohrer and Ward 2003: 321). Live coding has greatly evolved, becoming an established artistic and cultural practice (Blackwell, Cocker, Cox, McLean and Magnusson 2022), and that welcomes underrepresented communities to be involved in music technology, including women, non-binary individuals (Armitage and Thornham 2021) and disabled identities (Skuse 2020).

We find a flourishing use of artificial intelligence (AI) algorithms, unlocking its musical potential, as previously studied by the author (Xambó 2022). Machine learning (ML) algorithms allow the creation of computer programs that can learn and make predictions from experience through the training of datasets, which can be supervised or unsupervised by humans, to build models that can make predictions when faced with new data. Despite the musical potential, ML in live coding adds a layer of complexity. Interactive machine learning (IML)

(Fails and Olsen Jr 2003) is a human-centred approach to human–computer interaction (HCI) that allows users to tune the results of the training process towards their expectations. Efforts towards bringing IML concepts into the design of digital musical instruments as creative musical tools have been made by Rebecca Fiebrink and colleagues (Fiebrink and Caramiaux, 2018). Notably, the Wekinator (Fiebrink, Trueman and Cook 2009) allows artists to understand ML algorithms and embrace them in their practice.

Sound-based music has been identified as an inclusive approach to making music with novel technologies. The concept was envisioned by Leigh Landy as ‘sound-based music 4 all’, with a lower barrier of entry than traditional note-based music (Landy 2009), where ‘people of all ages, abilities and backgrounds will be able to share sounds and sound-based works as well as participate together in sound-based performance’ (ibid.: 532). By analogy, the use of sound samples in live coding can also lower the entry point to this coding musical practice.

Among the common strategies to lower the entry access to live coding is the use of design constraints. Design constraints in digital musical systems were considered by Thor Magnusson as a mechanism to promote creativity (Magnusson 2010). A remarkable example of a constrained live-coding system is Magnusson’s *ixi lang* (Magnusson 2011), a system built in SuperCollider (McCartney 2002) that allows the user to manipulate musical patterns by using a syntax that is simple to operate and understand.

In this vein, the author has developed the live-coding system Music Information Retrieval in Live Coding (MIRLC), a SuperCollider extension that offers a constrained and easy-to-use live-coding environment (Xambó, Lerch and Freeman 2018a). The code is publicly available.¹ The module MIRLCRep accesses the online sound database Freesound (Font, Roma and Serra 2013) in real time.

Freesound started in 2005 and has currently more than 500,000 sounds. Freesound has been designed to promote the use of sounds among sound researchers,

¹<https://github.com/axambo/MIRLC>.

developers and artists who can both upload and download sounds. The types of sounds include recorded and created sounds (e.g., soundscapes, ambience, electronic, loops, effects, noise and voice). The sounds are licensed under Creative Commons (CC) (Lessig 2004), which promotes the remix culture.

Accessing Freesound in live coding can lower the barrier of entry to live coding because it allows the live coder to focus on the live-coding experience of manipulating sounds with no need for sampling. However, this approach can also have drawbacks related to the heterogeneous nature of the sound database. The most prominent challenge is to retrieve undesired sounds. To overcome this issue, the author has developed the follow-up live-coding system Music Information Retrieval in Live Coding Auto (MIRLCA).

MIRLCA is another SuperCollider extension that allows users to customise a worldwide sampler by training models to retrieve only ‘desired’ sounds from Freesound using a constrained live-coding interface. This approach promotes a hands-on understanding of ML and IML. The system has been used in international workshops along with performances and was developed following participatory design methodologies (Xambó, Roma, Roig and Solaz 2021). The code is publicly available.²

This article identifies the new sonic challenges and opportunities brought to live coders, computer musicians and sonic artists by the use of MIRLCA, a live-coding environment powered with an AI system that works as a customisable sampler of CC sounds. Previously, we analysed two workshops and a concert with two performances using the system (Xambó et al. 2021). We found that the workshop participants and live coders took ownership of the code as well as trained and used the models. Here, we complete the analysis with two more workshops, two more concerts and four impromptu performances. Our analysis centres on (1) identifying the advantages and disadvantages of this live-coding approach, and (2) examining different compositional strategies involving manipulating a digital sampler of online CC sounds in live coding. Overall, we found this to be a novel approach for live coding.

2. THE WORLD AS A SAMPLER

This section revises foundational work on how the use of sounds from crowdsourced libraries and the internet have influenced performance practice.

2.1. The turntable as an instrument

Musique concrète, pioneered by Pierre Schaeffer and others, established the use of sound samples as raw materials. This was realised by manipulating turntables in the 1950s and continuing later with tape recorders and digital techniques (Schaeffer [1966] 2017: 38). A follow-up relates to the creative use of turntables and other devices in Jamaican dub in the late 1960s–early 1970s (Toop 1995: 115–18) and early hip hop in the 1970s–1980s (White 1996) to produce popular music. However, it was not until the dawn of affordable digital samplers in the 1980s that we find a popularisation of the use of sound samples as a common musical practice (Rodgers 2003; Harkins 2019). This was generally linked to the production of pop music and later electronic music.

Harkins provides a useful working definition of digital sampling as ‘the use of digital technologies to record, store, and reproduce any sound’ (Harkins 2019: 4). Tara Rodgers describes the processes involved in electronic music production as ‘selecting, recording, editing and processing sound pieces to be incorporated into a larger musical work’ (Rodgers 2003: 313). Rodgers also points to the characteristic of electronic musicians of taking the dual role of ‘producers–consumers of pre-recorded sounds and patterns that are transformed by a digital instrument that itself is an object of consumption and transformation’ (ibid.: 315).

2.2. The world as an instrument

Sound maps connect sound samples with their geolocation and have been widely explored since the advent of online databases and geolocation technologies. A prominent precursor is Murray Schafer’s acoustic ecology (Schafer 1977) and the related World Soundscape Project (Schafer 1977; Truax 2002) in the 1970s at Simon Fraser University in Vancouver, Canada. This collective brought international attention to the sonic environment and raised awareness about noise pollution through soundscapes and environmental sounds.

Portable digital audio recorders and the internet brought about the possibility of creating crowd-sourced sound maps of specific locations. In 2006, the composer Francisco López presented ‘The World as an Instrument’ at MACBA in Barcelona, a workshop where different artists’ work was introduced to reflect the new popular practices of soundscape composition using the ‘real world’ as a sonic palette.³

Embedded devices have allowed the creation of worldwide open live streamings, such as Locustream Open Microphone Project, which developed

²<https://github.com/mirlca>.

³www.macba.cat/en/exhibitions-activities/activities/world-instrument.

streamboxes and mobile apps for streaming remote soundscapes (Sinclair 2018). Started in 2005, the project offers a live sound map, the *Locusstream Soundmap*,⁴ showcasing a collection of live open microphones across the globe. Liveshout (Chaves and Rebelo 2011) is a mobile app that turns the phone into an open, on-the-move mic that can broadcast and contribute to the Locus Sonus soundmap with live collaborative performances (Sinclair 2018).

2.3. Audio Commons for music composition and performance

The so-called Web 2.0, also known as the social web, represented a tipping point in the history of the internet, when, in the early 2000s, websites began to emphasise user-generated dynamic online content instead of traditional top-down static web pages (Bleicher 2006). This included a range of online services that provide access to databases with hundreds of thousands of digital media files. The legal framework of CC licences (Lessig 2004) was devised to promote creativity by legally allowing for new ways of managing this digital content (e.g., sharing, reusing, remixing). This has led to a new community of prosumers, still relevant today, who both produce and consume online digital content (Ritzer and Jurgenson 2010), which aligns with the acknowledged presumption existent in the electronic music culture (Rodgers 2003). Navigating through these large databases is not an easy task and requires suitable tools.

The Audio Commons Initiative was conceived to promote the use of open audio content as well as to develop relevant technologies to support an ecosystem of databases, media production tools and users (Font et al. 2016). The use of CC sounds in linear media production has been discussed with use cases in composition and performance (Xambó, Font, Fazekas and Barthet 2019).

Online databases such as Freesound offer an application programming interface (API) or software interface to allow communication between software applications. The Freesound API⁵ allows developers to communicate with the Freesound database using a provided set of tools and services to browse, search and retrieve information from Freesound, which works in conjunction with the audio analysis of the sounds using *Essentia* (Bogdanov et al. 2013). Freesound Labs⁶ features several projects that foster creative ways of interacting with the CC sound database.

⁴<https://locusonus.org/soundmap>.

⁵<https://freesound.org/docs/api>.

⁶<https://labs.freesound.org>.

We can also find musical instruments that leverage cloud computing and CC sounds. For example, the smart mandolin generates a soundscape-based accompaniment using Freesound (Turchet and Barthet 2018). Smart musical instruments (SMIs), such as the smart mandolin, were defined by Luca Turchet as ‘devices dedicated to playing music, which are equipped with embedded intelligence and are able to communicate with external devices’ (Turchet 2018: 8948). SMIs are an instance of using AI principles in new interfaces for musical expression (NIMEs).

The use of CC sounds in hardware digital samplers has previously been explored. For example, SAMPLER allows users to load and play sounds from Freesound (Font 2021). As with any traditional music sampler, users need to take several design decisions (e.g., query the sound, shape the sound with an envelope or trim the start/end) before playing samples. Although it is possible to work with multiple sounds at once, there is a limit to the sounds that can be loaded due to the hardware capacity.

These hardware limitations are overcome with the system presented in this article by using software. The system takes a reactable-like approach (Jordà, Geiger, Alonso and Kaltbrunner 2007), whereby there is no distinction between designing a sound and performing with it. The sound is shaped while performed, aligning with process music proposed by Steve Reich, in which the process becomes the piece of music (Reich 1965). Another key difference is that, compared with hardware samplers, which tend to be connected to musical instrument digital interface (MIDI) notes mapping, such as in SAMPLER, our approach explores other avenues. This way, the creativity is not confined to music-note input but, instead, the samples can lead to other musical spaces with their own rhythms, an approach inspired by Gerard Roma and colleagues’ seminal work on the Freesound Radio (Roma, Herrera and Serra 2009). Our focus is on providing easy-to-use software with a live-coding interface that fosters sound-based algorithmic music fed by CC sounds.

2.4. CC sounds in live coding

There exist multiple live-coding environments, which typically support different ways of performing audio synthesis, including sample operations. However, access to online crowdsourced databases of sounds is less common. Gibber is one browser-based live-coding environment that allows the live coder to perform audio synthesis with oscillators, synthesisers, audio effects and samples, among others (Roberts, Wright and Kuchera-Morin 2015). Amid the different options for manipulating samples, it is possible to retrieve, load, play back and manipulate sounds by

using the object Freesound (ibid.). EarSketch is a web-based learning environment that takes a sample-based approach to algorithmic music composition using code and a digital audio workstation (DAW) interface (Freeman, Magerko, Edwards, Mcklin, Lee and Moore 2019). With an emphasis on musical genres, students can work with curated samples, personal samples or sounds from Freesound (ibid.). The audio repurposing of CC sounds from Freesound using music information retrieval (MIR) has been explored by the author and colleagues (Xambó et al. 2018a) and forms the foundation for this work.

While the manipulation of sounds from crowdsourced libraries in live coding has potential, it also has its limitations (see section 4). The main challenge is navigating unknown sound databases and finding appropriate sounds in live performance. One approach to overcoming the limitations is by combining personal and crowdsourced databases, which was explored by the author and colleagues obtaining promising results (Xambó, Roma, Lerch, Barthet and Fazekas 2018b). Ordiales and Bruno investigated the use of CC sounds from RedPanal.org and Freesound combined with sounds from local databases using a hardware interface for live coding (Ordiales and Bruno 2017). Another approach uses AI to enhance the retrieval of CC sounds (see section 3). It is out of the scope of this article to review the use of AI in live coding. An overview of several approaches to live coding using AI was presented by the author in a previous publication (Xambó 2022).

3. THE ENVIRONMENT OF AI-EMPOWERED MUSIC INFORMATION RETRIEVAL IN LIVE CODING

This section presents the research context, research question and research methods that guide this research as well as the nature of the data collection and data analysis from workshops, work-in-progress sessions, concerts and impromptu performances.

3.1. The MIRLCAuto project

In a nutshell, the system MIRLCAuto (MIRLCA)⁷ was built on top of MIRLCRep (Xambó et al. 2018a), a user-friendly live-coding environment designed within SuperCollider and the Freesound quark⁸ to query CC sounds from the Freesound online database applying MIR techniques. A detailed technical account of MIRLCA and some early findings were published in 2021 (Xambó et al. 2021).

⁷<https://mirca.dmu.ac.uk>.

⁸<http://github.com/g-roma/Freesound.sc>.

MIRLCA uses supervised ML algorithms provided by the Fluid Corpus Manipulation (FluCoMa) toolkit (Tremblay, Roma and Green 2022). Based on the live coder's musical preferences, the system learns and predicts the type of sounds the live coder prefers to be retrieved from Freesound. The aim is to offer a flexible and tailorable live-coding CC sound-based music environment. This approach allows 'taming' the heterogeneous nature of sounds from crowdsourced libraries towards the live coder's needs, enhanced with the algorithmic music possibilities brought about by live coding.

Started in 2020, MIRLCA was built on SuperCollider and is in ongoing development. The author is the main developer, informed by conversations with the live-coding community, typically in the form of workshops, following a participatory design process. Its development has been also informed by observing and discussing its use by 16 live coders, including the author, as early adopter users in work-in-progress sessions, impromptu performances and concerts.

3.2. Research question and research methods

This article aims to identify the new sonic challenges and opportunities brought to live coders, computer musicians and sonic artists by the use of MIRLCA, a live-coding environment powered by AI working as a customisable sampler of sounds from around the globe. Here, a reflective retrospection is undertaken to look at the challenges and opportunities of manipulating CC sounds in live coding focusing on the (1) advantages vs disadvantages, and (2) live-coding compositional strategies.

We analysed text (e.g., interview blog posts, workshop attendees' feedback) and video (e.g., work-in-progress sessions, concerts, impromptu performances), with most of the information publicly available in the form of blog posts and videos (see References section). A total of four workshops, three concerts with eight performances, four work-in-progress video sessions and four impromptu performances with four groups of one solo, two duos and one trio of live coders were analysed (see sections 3.3 and 3.4). These online and onsite activities involved more than 60 workshop participants and 16 live coders, including the author. We sought permission from the individuals named in the article.

To identify patterns of behaviour, the research methods are inspired by qualitative ethnographic analysis (Rosaldo 1993) and thematic analysis (Clarke and Braun 2006). Given the full-length video material of the concerts, work-in-progress sessions and impromptu performances, we also used video analysis techniques (Xambó, Laney, Dobbyn and Jordà 2013).

This research is a follow-up of our previous findings from two of the workshops and one concert with two performances (Xambó et al. 2021). While in our previous publication, we presented a behind-the-scenes look at the system and explored the concept of personal musical preferences referred to as ‘situated musical actions’ (Xambó et al. 2021), here we focus on the sonic potential to live coding that this novel approach entails.

3.3. The workshops and the work-in-progress sessions

Overall, we organised four workshops, inviting both beginners and experts in programming. Three workshops were carried out online while the fourth workshop was carried out onsite. Altogether, more than 60 participants attended the workshops.

The workshop ‘Performing with a Virtual Agent: Machine Learning for Live Coding’ was delivered three times in an online format. The three workshops had originally been planned as onsite with local participants but ended up becoming online due to the COVID-19 pandemic, which also allowed the inclusion of participants from around the world. The workshop was co-organised and delivered by the author together with Sam Roig in collaboration with three different organisations and communities: IKLECTIK (London), l’ull cec (Barcelona, Spain) and Leicester Hackspace (Leicester, UK).

The first workshop was held in December 2020 and it was organised in collaboration with IKLECTIK,⁹ a platform dedicated to supporting experimental contemporary art and music. The other two workshops were organised in January 2021. One was organised in collaboration with l’ull cec,¹⁰ an independent organisation that coordinates activities in the fields of sound art and experimental music and TOPLAP Barcelona,¹¹ a Barcelona-based live-coding collective. The last one was organised in collaboration with Leicester Hackspace,¹² a venue for makers of digital, electronic, mechanical and creative projects.

The purpose of these hands-on online workshops was to allow the participants (1) to explore the MIRLCRep2 tool (a follow-up version of MIRLCRep), and (2) to expose them to how ML could help improve the live-coding experience when using the MIRLCa tool. By the end of the workshops, the participants were able to train their ML models using our system’s live-coding training methods.

We offered tutorial sessions to help the workshop participants adapt the tool to their own practice.

⁹www.iklectik.org.

¹⁰<https://lullcec.org>.

¹¹<https://toplap.cat>.

¹²<https://leicesterhackspace.org.uk>.

Thanks to the support of l’ull cec, TOPLAP Barcelona and Phonos,¹³ a pioneering centre in the fields of electronic and electroacoustic music in Spain, we documented a series of four interviews and work-in-progress videos related to our workshop in Barcelona, featuring Hernani Villaseñor, Ramon Casamajó, Iris Saladino and Iván Paz.

In January 2022, the author together with Iván Paz co-organised the onsite workshop ‘Live Taming Free Sounds’. The workshop was part of the weekend event *on-the-fly: Live Coding Hacklab* at the Center for Art and Media Karlsruhe (ZKM) in Germany.¹⁴ In the Hacklab, we acted as mentors for the topic of ML in live coding (Figure 1).

The purpose of this hands-on workshop was to allow the participants (1) to get a quick overview of some different approaches to applying ML in live coding, (2) to do a hands-on inspection of how to classify CC sounds to use as a live digital worldwide sampler, and (3) to carry out an aesthetic incursion into sound-based music in live coding. By the end of the workshop, the participants were able to perform in a group with their ML models using our system’s live-coding training methods, as explained in the next section.

3.4. The concerts and impromptu performances

As a follow-up to the online workshops, we adapted our original idea of hosting three public onsite concerts to do what was possible under the pandemic circumstances. Consequently, the first concert, ‘Similar Sounds: A Virtual Agent in Live Coding’, hosted by IKLECTIK in December 2020 in London, was delivered online. The concert consisted of two solo performances by Gerard Roma and the author, followed by a Q&A panel with the two live coders together with the live coder expert in live coding and AI Iván Paz, and moderated by Sam Roig.

The second concert, ‘Different Similar Sounds: A Live Coding Evening “From Scratch”’, hosted by Phonos in April 2021 in Barcelona and organised in collaboration with TOPLAP Barcelona and l’ull cec, was delivered with an audience limitation of 15 people due to the pandemic restrictions. The concert comprised four live coders associated with TOPLAP Barcelona (Ramon Casamajó, Roger Pibernat, Iván Paz and Chigüire), who used MIRLCa ‘from scratch’, adapting the library to their particular approaches and aesthetics. The concert ended with a group improvisation ‘from scratch’ by the four performers.

The third concert, ‘Dirty Dialogues’, was organised by the Music, Technology and Innovation – Institute

¹³www.upf.edu/web/phonos.

¹⁴<https://zkm.de/en/event/2022/01/on-the-fly-live-coding-hacklab>.



Figure 1. Video screenshot of the workshop ‘Live Taming Free Sounds’ at *on-the-fly: Live Coding Hacklab* on 29–30 January 2022, ZKM, Karlsruhe, Germany. Video by Mate Bredan.

for Sonic Creativity (MTI²) in collaboration with l’ull cec. This concert was pre-recorded in May 2021 due to the COVID-19 restrictions and premiered online. The concert was an encounter of 11 musicians from the Dirty Electronics Ensemble led by John Richards together with Jon.Ogara and the author in a free music improvisation session (Figure 2). Apart from an online release of the performance and interview with the musicians, a live album was also released in October 2021 on the Chicago netlabel pan y rosas discos (Dirty Electronics Ensemble, Jon.Ogara and Xambó 2021).

In January 2022, and as part of the workshop at ZKM during the *on-the-fly: Live Coding Hacklab* weekend, the workshop attendees spent the last two hours of the workshop creating teams and preparing a showcase event, the impromptu performances (Figure 3). In total, there were four groups (one individual, two duos and one trio) together with the presentation of Naoto Hieda’s Hydra Freesound Auto,¹⁵ a self-live-coding system. Impromptu ‘from scratch’ sessions were performed by beginners in live coding together with the expert live coders Lina Bautista, Luka Frelih, Olivia Jack, Shelly Knotts and Iván Paz. The ‘from scratch’ sessions typically consisted of playing live for 9 minutes starting from an empty screen and ending with the audience applause (Villaseñor-Ramírez and Paz 2020).

4. MANIPULATION OF ONLINE CC SOUNDS IN LIVE CODING

Our approach to live coding embraces the tradition of digital sampling in an idiosyncratic way. First, it works

with sounds from a crowdsourced library, which can include sounds from the live coder uploaded to the online database. Generally, most of the sounds are recorded by others, and hence the magnitude of sounds available is much larger compared with personal sound collections. Second, the typical activities involved in digital sampling include the selection, recording, editing and processing of sounds, in which the process of collecting and preparing the samples tends to be especially time-consuming (Rodgers 2003: 314). While our approach centres on the live curation, editing and processing of the sounds, resembling a DJ’s mix. Third, the use of a digital sampler operated via live-coding commands, as opposed to hardware interface buttons, shapes the ways of interacting with the sounds. Small new programmes can be written relatively fast, and new computational possibilities can emerge.

Table 1 outlines some of the advantages and disadvantages of manipulating CC sounds in live coding experienced from the use of the MIRCRep library (Xambó et al. 2018a, 2018b) for SuperCollider. One salient advantage is the low-entry access to digital sampling and live coding due to the use of a constrained environment with high-level live-coding commands and an emphasis on manipulating ready-to-use sounds. Second, although often live coders embrace the unknown in their algorithms, here the unknown is embraced through the sound itself making the discovery of sounds an exciting quality. Third, noting the use of CC sounds by showing the metadata of the sound in real time (e.g., author, sound title) and generating a credit list at the end of a session raises awareness about digital commons as a valuable resource. Fourth, the sounds available are not restricted anymore to the data memory of the

¹⁵<https://labs.freesound.org/apps/2022/02/09/hydra-freesound.html>.



Figure 2. A moment of the performance ‘Dirty Dialogues’ with the Dirty Electronics Ensemble, Jon.Ogara and Anna Xambó on 17 May 2021 at PACE, De Montfort University, Leicester, UK. Photo by Sam Roig.



Figure 3. A ‘from scratch’ session with Olivia Jack (left) and the author (right) live coding with MIRLCA at *on-the-fly: Live Coding Hacklab* on 30 January 2022, ZKM, Karlsruhe, Germany. Photo by Antonio Roberts.

digital sampler but to the number of sounds available on the online sound database (e.g., more than 500K in Freesound). Fifth, the live-coding interactive access to the online database with content-based and semantic queries allows the live coder to achieve more variation together with certain control. Sixth, similar to other live-coding environments, it is possible to make music

and improvise immediately and to build narratives through the use of semantic queries, assuming that the required software libraries have been installed. Finally, tinkering with the code and tailoring the environment to each live coder’s needs is possible due to hosting the environment in the free and open-source SuperCollider software.

Table 1. Manipulation of CC sounds in live coding

Pros	Cons
• Live code using samples – easy entry point for beginners in live coding and digital sampling	• The sound results are not always as expected from a heterogeneous database of sounds
• Discovery of digital commons sounds on-the-fly and with a surprise factor	• The system requires being connected to the internet
• ‘Infinite’ sounds at your disposal	• The system is more computationally expensive than using audio synthesis from a local computer
• Interactive access to an online database of sounds	• The system depends on external libraries and APIs, which often can get modified and can affect your environment
• Improvise with the tool right away and construct semantic narratives	• Any changes in the structure of the code require technical knowledge
• Easily collaborate with others using the provided presets/ tool or adapt it to your needs/integrate it into other environments	• There is not an easy way to achieve rhythmic synchronisation among samples

However, this approach has also disadvantages. One prominent issue is that the retrieved sounds from the queries may not always be as desired, thus disrupting the musical flow or the live coder’s intention. Second, a crowdsourced sound database tends to have a wide range of sounds of different qualities, captured by different means, which makes it heterogenous by nature. Third, the constant downloading of sounds can become computationally more expensive than working with sound synthesis, yet nowadays it might be less noticeable with a standard laptop. Fourth, the technical requirements increase with the need to be connected to the internet in order to search and download the sounds in real time. Yet, in an increasingly connected world, this may be a minor issue. Fifth, other technical prerequisites are the software dependencies on external libraries, which may require certain tech-savvy knowledge. This demand also applies if the live coder wants to customise the environment of their creative production workflow. However, the online documentation should be helpful for those who are just starting out with live coding or digital sampling. Finally, although collaboration is possible, synchronisation support between live coders’ computers has not been implemented yet. This is not seen as a priority feature given the potential for interesting rhythmic structures emerging from the combination of the sounds without synchronising.

To mainly deal with the issue of obtaining unexpected sound results in live performance, we devised the use of ML to train a binary classifier to return results closer to a serendipitous choice instead of a random choice (Xambó et al. 2021). Table 2 illustrates the pros and cons of manipulating CC sounds in live coding enhanced with ML from our

experience of using MIRLCA (ibid.). Considering that MIRLCA is an environment still in ongoing development, we focus here on the overall approach and disregard particular missing features or existent failures that are expected to be addressed in the future. Thus, we talk about a generic classifier without specifying the number of categories supported. For example, there are plans for the binary classifier to be expanded to more than two categories to make it less limited.

On the one hand, the most visible advantages of this approach include the potential of customising the environment to be prompted towards serendipity by training the system to learn and predict from a provided set of categories, such as ‘good’ versus ‘bad’ sounds. Second, it is also possible to train the system under different musical contexts or ‘situated musical actions’, such as training a session that can predict ‘good’ rhythmic sounds versus ‘bad’ rhythmic sounds for a part of a live-coding session. Last, to customise the system and prior to the performance, it is possible to create a small annotated dataset and to generate a neural network model for a particular use, which for the live coder can become an easy entry access point to ML concepts.

On the other hand, several drawbacks arise from this approach. Although effort can be made to customise the system by training the live coder’s own neural networks, the sound results can still be perceived as unwanted. Second, the musical proximity with the sound-based music helps, but it is also true that the live coder will need some time to grasp the workflow of the system and to obtain interesting sonic subspaces. Moreover, the training of the neural network model can take a certain amount of time

Table 2. Manipulation of curated CC sounds in live coding using ML

Pros	Cons
<ul style="list-style-type: none"> • Train/customise your digital sampler to play sounds by categories, e.g. ‘good’ sounds (prompt towards serendipity) 	<ul style="list-style-type: none"> • The sound results might still not be as expected
<ul style="list-style-type: none"> • Train and use different neural network models (according to different <i>situated musical actions</i>) using a live-coding approach 	<ul style="list-style-type: none"> • Training/customising your sampler can take time (on average at least 1 hour), which reduces the immediacy of improvisation
<ul style="list-style-type: none"> • Get to grips with a classifier and IML – easy entry point for beginners in ML 	<ul style="list-style-type: none"> • The learning curve can become steep with the addition of ML

(on average at least one hour), which requires some planning and preparation. This could affect the improvisational spirit of live coding. However, the system works with a provided model so there is no need for training if the live coder does not want to. Altogether, while the learning curve can become steep, the potential of live ‘taming’ the free sounds should be worth it.

5. LIVE-CODING STRATEGIES

We identified several different live-coding compositional strategies from manipulating a digital sampler of online CC sounds related to the themes: *from scratch*, *algorithmic soundscape composition*, *embracing the error*, *tailoring and DIY*, *the role of training the models* and *collaborative constellations*.

5.1. From scratch

The ‘from scratch’ live-coding technique commonly used in the live coding scenes in Mexico City and Barcelona has been defined as ‘to write code in a blank document against time’ (Villaseñor-Ramírez and Paz 2020: 60), and more particularly as ‘a creative technique that emphasises (visualises) real-time code writing’ (ibid.: 67).

Many of the live coders started ‘from scratch’ with empty canvases in their live-coding sessions with MIRLCA. For example, Hernani Villaseñor and Iván Paz preferred to keep simple code programs, aligned with the intention of a ‘from scratch’ performance in order ‘to find more efficient code structures and syntax (e.g., concise, compact, succinct), to, with the least amount of characters, achieve developing a complete piece’ (ibid.: 66). Needless to say, one of the key principles of the TOPLAP manifesto¹⁶ is ‘Obscurantism is dangerous. Show us your screens.’

In both work-in-progress sessions, Hernani Villaseñor and Iván Paz started from a blank canvas and generally worked with two or three groups of two

or three sounds each. The sounds were a reminder of everyday sounds, which were processed and assembled in creative ways ‘à la tape music’. At first, the sounds were typically retrieved randomly with the function `random()`. Then, similar sounds were searched with the function `similar()`. After that, the sounds’ sample rates were modified with the functions `play()` or `autochopped()` to speed the sounds up or down for a period. The following example illustrates a code excerpt of this approach using two groups of sounds:

```
a = MIRLCA.new
a.random
a.similar
a.play(0.5)

b = MIRLCA.new
b.random
b.similar
b.play(0.2)
b.autochopped(32, 2)
b.delay
```

Iris Saladino took a different approach to ‘from scratch’ in her work-in-progress session. Her approach was to combine two software systems. First, she used MIRLCRep2 to search and download sounds from Freesound using tags (e.g., ‘sunrise’, ‘traffic’, ‘sunset’) that she saved in a local folder. Second, she processed the sounds ‘from scratch’ using TidalCycles,¹⁷ a live-coding environment designed for making patterns with code. The musical result resembled generative ambient music.

5.2. Algorithmic soundscape composition

Many of the live coders took advantage of the bespoke functions available in MIRLCA as well as the built-in functions from their live-coding environments to generate algorithmic music processes. As discussed in the previous section, the tandem `random()` and `similar()` functions are often used to retrieve

¹⁶<https://toplap.org/wiki/ManifiestoDraft>.

¹⁷<https://tidalcycles.org>.

sounds. Here, an initial random sound is retrieved, which is followed by a similar sound to maintain a consistent sonic space. Starting a session with a random sound expresses openness and uncertainty, as well as uniqueness, because the likeliness of two performances starting with the same sound is small. Arguably the combination of random sounds with other ways of retrieving sounds, such as similar sounds or sounds by tag, shows that building a narrative from random sounds is possible, despite this being questioned by Barry Truax when discussing using a random collage of environmental sounds for sound-scape composition (Truax 2002: 6).

SuperCollider supports algorithmic composition with a wide variety of built-in functions. Gerard Roma and Roger Pibernat started their sessions ‘from scratch’, and combined the provided algorithmic functions of MIRLCA with algorithmic functions or instructions from SuperCollider. Roger Pibernat used TDef in his performance hosted by Phonos to create time patterns that changed parts of the code at a constant rate. For example, the sample rate of a group of sounds was instructed to change every four seconds by randomly selecting an option from a list of three values: `a.play([0.25, 0.5, 1]).choose`. In his performance at IKLECTIK, Gerard Roma accessed the buffers of the sound samples to apply SuperCollider functions to them. Using JITLib,¹⁸ a customised unit generator `PlayRnd` randomly played five buffers of sounds previously downloaded using the `tag` and `similar` functionality in MIRLCA. The following example shows the code:

```
p = ProxySpace.push
p.fadeTime = 10
~x.play
~x.source = {
  0.2*mix.ar(PlayRnd.ar((1.5),
  0.5, 1))!2;
}
a = MIRLCA.new
a.tag("snow", 5)
a.delay(10)
a.similar
a.printbuffers
```

The MIRLCA functions `autochopped` or `play-auto` for playing randomly assigned sample rates or `similarauto` for automatically obtaining similar sounds from a target sound are also used frequently to give some autonomous and algorithmic behaviour to groups of sounds.

5.3. Embracing the error

Learning how to embrace errors is part of the live-coding practice. As Ramon Casamajó mentioned about his performance organised by Phonos: ‘the error is part of the game’. These include minor errors, such as the system not finding any candidate sound from a query, and major errors, such as getting an unwanted sound. As Iván Paz reflected from his concert hosted by Phonos: ‘methods such as `similar` can produce unpleasant surprises that you have to embrace and control the performance on-the-fly’. In turn, this can prompt free improvisation, as Chigiire commented after their performance hosted by Phonos: ‘There was some degree of unpredictability that made me feel comfortable with less preparation. I didn’t have any concrete idea in mind, I felt much freer to improvise.’

The aesthetics of imperfection in music and the arts has been discussed (Hamilton and Pearson 2020). The values of spontaneity, flaws and unfinished are highlighted as relevant to the improvisatory nature of the creative work. Shelly Knotts remarks that live coding is an error-driven coding activity (Knotts 2020: 198) and points out how the unknown, errors and mistakes can become a sociopolitical statement in live coding: ‘By resisting strongly defined boundaries and idealized forms of musical practice based on technical accuracy, live coding remains an inclusive, open-ended and evolving practice grounded in social and creative freedom’ (ibid.: 189–90). The MIRLCA system opens up new dimensions in engaging with the unknown because the live coder cannot have entire control of the incoming sounds that emerge from the real-time queries. Instead, the live coder is prompted to sculpt the incoming new sounds.

5.4. Tailoring and DIY

In the live-coding community, each live coder has their set-up and environment. Some instances illustrated how the live coders combined MIRLCA with their usual software for live coding. Hernani Villaseñor superimposed ‘two editors: Emacs with a transparent background running MIRLCA and, underneath, Atom running Hydra¹⁹ for capturing the typing with a webcam’. Ramon Casamajó and Iris Saladino used MIRLCA/MIRLCRep2 in combination with TidalCycles. Ramon Casamajó combined downloaded sounds from the internet using MIRLCA with sounds from TidalCycles stored in a local drive: ‘I’ve approached the Tidal side more rhythmically, recovering some patterns I had written for past performances. On the MIRLCA side, I’ve looked for vocals for the first part of the piece and noisy textures for

¹⁸<https://doc.scode.org/Overviews/JITLib.html>.

¹⁹<https://hydra.ojack.xyz>.

the end.’ As a result, we could experience two sonic spaces or two parallel digital musical instruments. Beyond software, Jon.Ogara explored MIRLCA combined with hardware using the Kinect sensor and Max.

5.5. The role of training the models

The role of training the models for the performance connects with the notion of specific musical contexts, what we termed ‘situated musical actions’ (Xambó et al. 2021), which can help understand the training of a new model. For example, Gerard Roma trained a model ‘to favour environmental sound and reject human voice and synthetic music loops, which are common in Freesound’ (ibid.: 11). This was done in order to then use it in performance to control the resulting types of sounds: ‘In each section, the results from the tag searches were extended by similarity search and the model was used to ensure that the results remained within the desired aesthetic. This allowed a more open attitude for launching similarity queries with unknown results, encouraging improvisation’ (ibid.: 11).

Likewise, Iván Paz also agreed about the improvisational nature of using a trained model, and the curatorial role of the live coder, commenting that the result is ‘like knowing what types of sounds you will receive and trying to organise/colour them on-the-fly’. Iván Paz commented about the trade-off relationship between unexpected results and training accuracy: ‘There’s probably a sweet balance between surprises and consistency within the training accuracy.’ Indeed, we are only at the beginning of the possibilities that this musical perspective can bring.

Some of the live coders trained multiple models for different groups of sounds. For example, Jon.Ogara envisioned a long-term project of a diary of neural nets (snapshots of a musical biography or life journey) based on how he reacts to events using singular words. In the collaborations with more than one live coder, each live coder used their trained model/s, which is discussed in the next section.

The IML approach for the training process with MIRLCA as a live-coding session blurs the division between offline and on-the-fly training. In the on-the-fly event at ZKM, Luka Frelüh performed a ‘from scratch’ training session for algorave sounds, including canvassing the audience’s opinion to help him label the sounds as ‘good’ or ‘bad’. The brief for the training was to create music that you can dance to. After obtaining a decent training accuracy, the model was tested live. The proof of the success of the model was that some people indeed danced.

5.6. Collaborative constellations

We observed five collaborative performances that were all improvisational by nature. In the five collaborations, each live coder had trained their own model previously and then performed live with their trained models.

In the concert hosted by Phonos, the four live coders ended by performing a ‘from scratch’ session altogether. The layout was configured in such a way that the live coders created an outer ring with the audience inside. Each live coder had two pairs of speakers configuring an 8-multichannel system, a laptop running MIRLCA and a projector. Ramon Casamajó mentioned here the importance of listening to each other: ‘I tried to listen to quite a bit of the rest and look for reactive or complementary sounds to theirs, trying to leave empty spaces as well.’ Although the conversation could be difficult sometimes. Chigüire thought about it ‘as a conversation at a loud bar’, and Iván Paz found that ‘it was very challenging to synchronise all the sounds.’

In the concert at MTI², we explored collaboration with a combination of analogue and digital instruments, acoustic and electronic materials, as well as live coding and DIY sound-making techniques. Both Jon.Ogara and the author performed with MIRLCA, the former as part of an acoustic and electronic ensemble while the latter performed in a live-coding style. The improvisation was an exercise of listening to each other and making a suitable call–response. For example, the improvisation started with three Dirty Electronics Ensemble members performing with different DIY circuits and found objects producing incremental tides of noise, while Jon.Ogara slowly faded in a female voice sound sample that whispered ‘come back alive’ and the author retrieved a repetitive sound sample of a printer printing.

In the showcase at ZKM, there were two duos and one trio who performed ‘from scratch’. Each live coder had a projected screen and could connect to a mixing desk with stereo output. The music ranged from algorave beats to soundscape, to glitch music, with some contrasting sounds that were handled as they appeared in the scene. In the ensembles, there were also a combination of expert live coders and beginners. For beginners, working with sound samples seemed like a suitable low-entry access to start with live coding, because they could refer to familiar sounds, apart from sharing the performance space with experts seemed to be an optimal learning scenario.

Offering a performer-only audio output (via headphones) would be an option to allow the live coder to test the new incoming sound before launching it. Although this feature has been explored in collaborative musical interfaces (Fencott and Bryan-Kinns

2012), it would not favour the flow of process music and the surprise factor brought by MIRLCA, where the live coder shapes the new incoming sounds that emerge unexpectedly. This connects with the remix culture already anticipated with dub music and what it meant to dub a track, ‘as if music was modelling clay rather than copyright property’ (Toop 1995: 118). The tension between control and surprise seems to work well with the MIRLCA system, which promotes freedom and openness commonly found in music improvisation.

6. CONCLUSION

In this article, we introduced a new approach to live coding and digital sampling that promotes the on-the-fly discovery of CC sounds. We presented a bespoke system, MIRLCA, a customisable sampler of sounds from Freesound that can be enhanced with ML, and highlighted several challenges and opportunities.

We presented the feedback of live coders who tested the system and how they used the tool. The customisation of the sampler using ML invites the live coder to train new models. Although customised training models can reduce unwanted results from online sound databases, it is still an uncertain space that might not always bring the desired serendipitous sound results. In performance, the system has proven to be suitable for free improvisation and shown that it can be used in heterogeneous ensembles.

The combination of the sampler functionalities with coding results in a novel approach to deal with ‘infinite’ sounds that emerge with a certain autonomous level. This distinctive behaviour brings a risk for the unknown, a singular characteristic that aligns well with values found in music improvisation such as freedom, openness, surprise and unexpectedness. This approach has potential but can be inconsistent sometimes given the wilderness nature of crowd-sourced online sound databases.

Despite this article focusing on live-coding practice, we can foresee the benefits of this approach in other areas. For example, the sampler could be used in both modes of performance or training, to discover sounds based on semantic enquiries. This can work well with tasks that entail sound-based music composition or sound design. From a musicological standpoint, the present article contributes a detailed account of the collaborative nature of the live-coding community and describes how the knowledge is openly shared and embraced, including the musical aesthetics from the use of CC sounds. We also envision that this approach can have benefits in education, by bringing digital commons and music improvisation to the classroom using a

creative and constrained environment that provides low-entry access and a flexible approach to using sound samples.

Acknowledgements

This project was funded by the EPSRC HDI Network Plus Grant (EP/R045178/1). The author would like to thank the anonymous reviewers for their time and thoughtful suggestions. The author gives special thanks to Will Adams for proofreading and Gerard Roma and Iván Paz for their help during the writing. The author is grateful to the workshop attendees and early adopters of the tool for their participation in the project and positive insights. The author thanks all the people and organisations involved in this project who helped immensely in making it a reality. The analysed footage from ZKM was filmed by Mate Bredan during the ‘on-the-fly: Live Coding Hacklab’ at ZKM | Center for Art and Media Karlsruhe in January 2022. Finally, the author thanks the live-coding and Freesound communities.

REFERENCES

- Armitage, J. and Thornham, H. 2021. Don’t Touch My MIDI Cables: Gender, Technology and Sound in Live Coding. *Feminist Review* **127**(1): 90–106.
- Blackwell, A. F., Cocker, E., Cox, G., McLean, A. and Magnusson, T. 2022. *Live Coding: A User’s Manual*. Cambridge, MA: MIT Press.
- Bleicher, P. 2006. Web 2.0 Revolution: Power to the People. *Applied Clinical Trials* **15**(8): 34.
- Bogdanov, D., Wack, N., Gómez, E., Gulati, S., Herrera, P., Mayor, O., et al. 2013. Essentia: An Open-Source Library for Sound and Music Analysis. *Proceedings of the 21st ACM International Conference on Multimedia*. New York: ACM, 855–8.
- Chaves, R. and Rebelo, P. 2011. Sensing Shared Places: Designing a Mobile Audio Streaming Environment. *Body, Space & Technology* **10**(1). <http://doi.org/10.16995/bst.85>.
- Clarke, V. and Braun, V. 2006. Using Thematic Analysis in Psychology. *Qualitative Research in Psychology* **3**(2): 77–101.
- Collins, N., McLean, A., Rohrhuber, J. and Ward, A. 2003. Live Coding in Laptop Performance. *Organised Sound* **8**(3): 321–30.
- Fails, J. A. and Olsen Jr., D. R. 2003. Interactive Machine Learning. *Proceedings of the 8th International Conference on Intelligent User Interfaces*. Miami, FL: Association for Computing Machinery, 39–45.
- Fencott, R. and Bryan-Kinns, N. 2012. Audio Delivery and Territoriality in Collaborative Digital Musical Interaction, *The 26th BCS Conference on Human Computer Interaction*, Birmingham, UK, 69–78.

- Fiebrink, R. and Caramiaux, B. 2018. The Machine Learning Algorithm as Creative Musical Tool. In R. T. Dean and A. McLean (eds.), *The Oxford Handbook of Algorithmic Music*. Oxford: Oxford University Press, 518–35.
- Fiebrink, R., Trueman, D. and Cook, P. R. 2009. A Meta-Instrument for Interactive, On-The-Fly Machine Learning. *Proceedings of the International Conference on New Interfaces for Musical Expression*, Pittsburgh, PA, 280–5.
- Font, F. 2021. *SOURCE: A Freesound Community Music Sampler, Audio Mostly 2021*. New York: ACM, 182–7.
- Font, F., Brookes, T., Fazekas, G., Guerber, M., La Burthe, A., Plans, D., et al. 2016. Audio Commons: Bringing Creative Commons Audio Content to the Creative Industries. *Audio Engineering Society Conference: 61st International Conference: Audio for Games*, Audio Engineering Society.
- Font, F., Roma, G. and Serra, X. 2013. Freesound Technical Demo. *Proceedings of the 21st ACM International Conference on Multimedia*. New York: ACM, 411–12.
- Freeman, J., Magerko, B., Edwards, D., Mcklin, T., Lee, T. and Moore, R. 2019. Earsketch: Engaging Broad Populations in Computing through Music. *Communications of the ACM* **62**(9): 78–85.
- Hamilton, A. and Pearson, L. (eds.) 2020. *The Aesthetics of Imperfection in Music and the Arts: Spontaneity, Flaws and the Unfinished*. London: Bloomsbury.
- Harkins, P. 2019. Introduction. In *Digital Sampling: The Design and Use of Music Technologies*. Abingdon: Routledge, 1–14.
- Jordà, S., Geiger, G., Alonso, M. and Kaltenbrunner, M. 2007. The reacTable: Exploring the Synergy between Live Music Performance and Tabletop Tangible Interfaces. *Proceedings of the 1st International Conference on Tangible and Embedded Interaction*, New York, 139–46.
- Knotts, S. 2020. Live Coding and Failure. In A. Hamilton and L. Pearson (eds.), *The Aesthetics of Imperfection in Music and the Arts: Spontaneity, Flaws and the Unfinished*. London: Bloomsbury, 189–201.
- Landy, L. 2009. Sound-based Music 4 All. In R. T. Dean (ed.), *The Oxford Handbook of Computer Music*. Oxford: Oxford University Press, 518–35.
- Lessig, L. 2004. The Creative Commons. *Montana Law Review*, 65. <https://scholarworks.umt.edu/mlr/vol65/iss1/1>.
- Magnusson, T. 2010. Designing Constraints: Composing and Performing with Digital Musical Systems. *Computer Music Journal* **34**(4): 62–73.
- Magnusson, T. 2011. The ixi lang: A SuperCollider Parasite for Live Coding. *Proceedings of the International Computer Music Conference*. Huddersfield, UK: ICMA, 503–6.
- McCartney, J. 2002. Rethinking the Computer Music Language: SuperCollider. *Computer Music Journal* **26**(4): 61–8.
- Ordiales, H. and Bruno, M. L. 2017. Sound Recycling from Public Databases: Another BigData Approach to Sound Collections. *Proceedings of the International Audio Mostly Conference*, Trento, Italy.
- Reich, S. 1965. Music as a Gradual Process. In S. Reich (ed.), *Writings on Music*. Oxford: Oxford University Press, 34–6.
- Ritzer, G. and Jurgenson, N. 2010. Production, Consumption, Prosumption: The Nature of Capitalism in the Age of the Digital ‘Prosumer’. *Journal of Consumer Culture* **10**(1): 13–36.
- Roberts, C., Wright, M. and Kuchera-Morin, J. 2015. Music Programming in Gibber. *Proceedings of the International Computer Music Conference, ICMA*, 50–7.
- Rodgers, T. 2003. On the Process and Aesthetics of Sampling in Electronic Music Production. *Organised Sound* **8**(3): 313–20.
- Roma, G., Herrera, P. and Serra, X. 2009. Freesound Radio: Supporting Music Creation by Exploration of a Sound Database. Paper presented at Computational Creativity Support Workshop CHI09, Boston, MA.
- Rosaldo, R. 1993. *Culture & Truth: The Remaking of Social Analysis*. Boston, MA: Beacon Press.
- Schaeffer, P. [1966] 2017. *Treatise on Musical Objects: An Essay across Disciplines*. Oakland, CA: University of California Press.
- Schafer, R. M. 1977. *The Soundscape: Our Sonic Environment and the Tuning of the World*. Rochester, VT: Destiny Books.
- Sinclair, P. 2018. Locustream Open Microphone Project. *Proceedings of the International Computer Music Conference*. Daegu, South Korea: ICMA, 271–5.
- Skuse, A. 2020. Disabled Approaches to Live Coding, Crippling the Code. *Proceedings of the International Conference on Live Coding*. Limerick, Ireland: ICMA, 5: 69–77.
- Toop, D. 1995. *Ocean of Sound: Aether Talk, Ambient Sound and Imaginary Worlds*. London: Serpent’s Tail.
- Tremblay, P. A., Roma, G. and Green, O. 2022. The Fluid Corpus Manipulation Toolkit: Enabling Programmatic Data Mining as Musicking. *Computer Music Journal* **45**(2): 9–23.
- Truax, B. 2002. Genres and Techniques of Soundscape Composition as Developed at Simon Fraser University. *Organised Sound* **7**(1): 5–14.
- Turchet, L. 2018. Smart Musical Instruments: Vision, Design Principles, and Future Directions. *IEEE Access*, **7**: 8944–63.
- Turchet, L. and Barthet, M. 2018. Jamming with a Smart Mandolin and Freesound-Based Accompaniment. *23rd Conference of Open Innovations Association (FRUCT)*, IEEE, 375–81.
- Villaseñor-Ramírez, H. and Paz, I. 2020. Live Coding From Scratch: The Cases of Practice in Mexico City and Barcelona. *Proceedings of the 2020 International Conference on Live Coding*. Limerick, Ireland: University of Limerick, 59–68.
- White, M. 1996. The Phonograph Turntable and Performance Practice in Hip Hop Music, Ethnomusicology OnLine 2. *umbc*. www.umbc.edu/eol/2/white/ (accessed 30 December 2022).
- Xambó, A. 2022. Virtual Agents in Live Coding: A Review of Past, Present and Future Directions. *eContact!* **21**(1). https://econtact.ca/21_1/xambosedo_agents.html (accessed 19 September 2022).

- Xambó, A., Font, F., Fazekas, G. and Barthet, M. 2019. Leveraging Online Audio Commons Content for Media Production. In M. Filimowicz (ed.), *Foundations in Sound Design for Linear Media*. Abingdon: Routledge, 248–82.
- Xambó, A., Laney, R., Dobbyn, C. and Jordà, S. 2013. Video Analysis for Evaluating Music Interaction: Musical Tabletops. In S. Holland, K. Wilkie, P. Mulholland and A. Seago (eds.), *Music and Human-Computer Interaction*. Cham, Switzerland: Springer, 241–58.
- Xambó, A., Lerch, A. and Freeman, J. 2018a. Music Information Retrieval in Live Coding: A Theoretical Framework. *Computer Music Journal* 42(4): 9–25.
- Xambó, A., Roma, G., Lerch, A., Barthet, M. and Fazekas, G. 2018b. Live Repurposing of Sounds: MIR Explorations with Personal and Crowdsourced Databases. *Proceedings of the International Conference on New Interfaces for Musical Expression*. Blacksburg, VA: Virginia Tech.
- Xambó, A., Roma, G., Roig, S. and Solaz, E. 2021. Live Coding with the Cloud and a Virtual Agent. *Proceedings of the International Conference on New Interfaces for Musical Expression*. Shanghai, China: NYU Shanghai.

INTERVIEWS

- Roig, S. and Xambó, A. 28 January 2021. Different Similar Sounds: An Interview with Hernani Villaseñor. <https://mirlca.dmu.ac.uk/posts/different-similar-sounds-interview-with-hernani-villasenor>
- Roig, S. and Xambó, A. 4 February 2021. Different Similar Sounds: An Interview with Ramon Casamajó. <https://mirlca.dmu.ac.uk/posts/different-similar-sounds-interview-with-ramon-casamajo>
- Roig, S. and Xambó, A. 18 February 2021. Different Similar Sounds: An Interview with Iris Saladino. <https://mirlca.dmu.ac.uk/posts/different-similar-sounds-interview-with-iris-saladino>

- [dmu.ac.uk/posts/different-similar-sounds-interview-with-iris-saladino](https://mirlca.dmu.ac.uk/posts/different-similar-sounds-interview-with-iris-saladino)
- Roig, S. and Xambó, A. 19 March 2021. Different Similar Sounds: An Interview with Iván Paz. <https://mirlca.dmu.ac.uk/posts/different-similar-sounds-interview-with-ivan-paz>
- Xambó, A. and Roig, A. 14 May 2021. An Interview with Jon Ogara. <https://mirlca.dmu.ac.uk/posts/interview-with-jon-ogara>
- Xambó, A. 28 September 2021. Different Similar Sounds ‘From Scratch’: A Conversation with Ramon Casamajó, Iván Paz, Chigüire, and Roger Pibernat. <https://mirlca.dmu.ac.uk/posts/different-similar-sounds-from-scratch-a-conversation-with-ramon-casamajo-ivan-paz-chiguire-and-roger-pibernat>

DISCOGRAPHY

- Dirty Electronics Ensemble, Jon.Ogara and Xambó, Anna. 1 October 2021. Dirty Dialogues. pan y rosas discos, pyr313. www.panyrosasdiscos.org/pyr313-dirty-electronics-ensemble-jon-ogara-and-anna-xambo-dirty-dialogues

VIDEOGRAPHY

- Different Similar Sounds: A Live Coding Evening ‘From Scratch.’ September 2021. <https://youtu.be/IDVsawECK2Y>
- Dirty Dialogues – The Performance. October 2021. <https://vimeo.com/626477944>
- Dirty Dialogues – The Interview. October 2021. <https://vimeo.com/626564500>
- Similar Sounds: A Virtual Agent in Live Coding. December 2020. <https://youtu.be/ZRqNfgg1HU0>