Street Code - live coding in public space

Alexandra Cárdenas

Independent
tiemposdelruido@gmail.com

ABSTRACT

Live coding is a technique for creating music through the computation of algorithms in real time. It includes a set of philosophical and technological characteristics that have the potential to unchain deep and urgent questions regarding the aesthetic, politic, technical and social aspects of music. Space, as a common thread that deals with each one of these aspects, has been an important subject of questioning and exploration by live coders throughout the three decades of development of this practice. Live coding is intimately related to hacker ethics and appropriates current technological advancements, bringing a new set of tools and aesthetic approaches to music creation. Moreover, thanks to a particular collaboration between music composition and computer science, live coders are endowed with the curiosity and the capability to overcome given cultural rules to create new spaces and contexts for their performances.

Public space offers live coders an immense territory for exploration through the possibility of the recontextualisation of artistic codes, the reappropriation of the social relations of music, and the diversification of the conception of space. Live coders can offer distinct contextualisations of aesthetic experiences in the urban space unexplored thus far. Therefore, in this paper, I intend to bring attention to the public space as a consequent stage for live coded music.

1. INTRODUCTION

In this paper, I discuss the possibilities that the live coding of music can bring to the exploration of the public space, and the possibilities that the exploration of public space can bring to the live coding of music. I describe my ideas and perceptions from the point of view of a composer of electroacustic music whose creative focus lies in live electronics and specifically, in live coding.

Street code is a series of performances of live coded electroacoustic music in the public space, premiered in Berlin on the 4th of February of 2017. In this project, I look for connecting points between the various aspects of my work: composition, improvisation, instrument building, live electronics, programming, pattern thinking, performance, and live coding, and their relationship to space. During the process of description of these connecting points, new connections emerge towards other disciplines and fields that deal with public territory, the poetics of code and the recontextualization of the social and cultural aspects of music and sound art.

The art of sound in the public space has been the domain of sound art in the last decades. Usually, when referring to music in the public space, what comes to mind are the various types of popular music festivals or the different expressions of street music. Appearances of western academic music in the public space are mostly narrowed to the greatest hits of famous composers from the past. Contemporary music has been, for the most part, kept secluded in its usual venues. Electroacoustic music has had an important tradition of exploration of the space, but only a minor part of this focus has gone into the public space.

Technological advances in computer technology, and therefore in computer music, have instigated the mobility of the electroacoustic studio. The laptop (as the quintessential representation of the portable electroacoustic studio) and the increased portability and potency of electronic equipment allow the possibility to explore space in ways that were unthinkable for computer music before. When released out of the confines of the non-portable electroacoustic studio, computer music faces a whole new horizon for the exploration of space, both in the physical sense and in the social sense.

With *Street Code*, I propose to take the laptop to the public space for the creation of electroacoustic music. But more specifically, I want to bring attention to the possibility of live coded electroacoustic music in the public space. This, due to the intrinsic characteristics of live coding that make it a highly reactive, inclusive and experimental practice.

2. LIVE CODING ELECTROACOUSTIC MUSIC

"In the current technological condition, media formats need not be linear, deterministic, and static; their real-time rendering allows for interpretation, interaction, and change of the type we find in mobile apps, generative music, and games. Performance contexts will increasingly reflect the dynamic nature of modern media, where composers, performers, and audience (or any amalgamation thereof) are able to write or re-write notation, scores, or any other instructions and are able to design or re-design software, hardware, or other machinery in a more open and dynamic way." (Thor Magnusson, 2014)

Live coding was born in the academia in the hands of computer musicians who looked for new means to create music in a more unrestricted set of conditions. This has represented the opening of a limitless horizon for electroacoustic[1] music. Many live coders are composers themselves, and different expressions of composed music for live coding in many combinations are to be found in their pieces. From solo free improvisations in the frame of an electroacoustic concert to mixed pieces with acoustic instruments, live coded scores for soloists to network ensembles, electroacoustic music has seen the appearance of new forms and possibilities thanks to live coding.

The networked capabilities of the computer have allowed composers to investigate new possibilities of interaction since the 1970s. Most recently, the duo "very long cat", for example, composed by David Ogborn and Shawn Mativetsky, a live coder and a tabla player respectively, perform via the Internet making use of eclectic technologies with the main goal to explore the impositions of the idea of "zero latency" in a live coding ensemble. Audio signals require time to move from one place to another and be decoded and transcoded again (Ogborn & Mativetsky, 2015). It is in the nature of the impossibilities of the musical technology where live coders find the artistic value to be worked with.

"With the development of social networking, interface design and algorithms have become integral to our social interplay with data collection, content personalisation and behaviour prediction mediating the potential interactions " (Knotts, 2016). Electroacoustic music creation also reflects the nature of society's interaction with data and communications, and this is the structural point of exploration by OFFAL (Orchestra For Females And Laptops), who play over the internet making use of systems that allow the members of the orchestra to vote and democratically decide the evolution of the musical piece.

[1] I make use of the term "electroacoustic" in this paper, to differentiate academic music from the popular music "electronic" term.

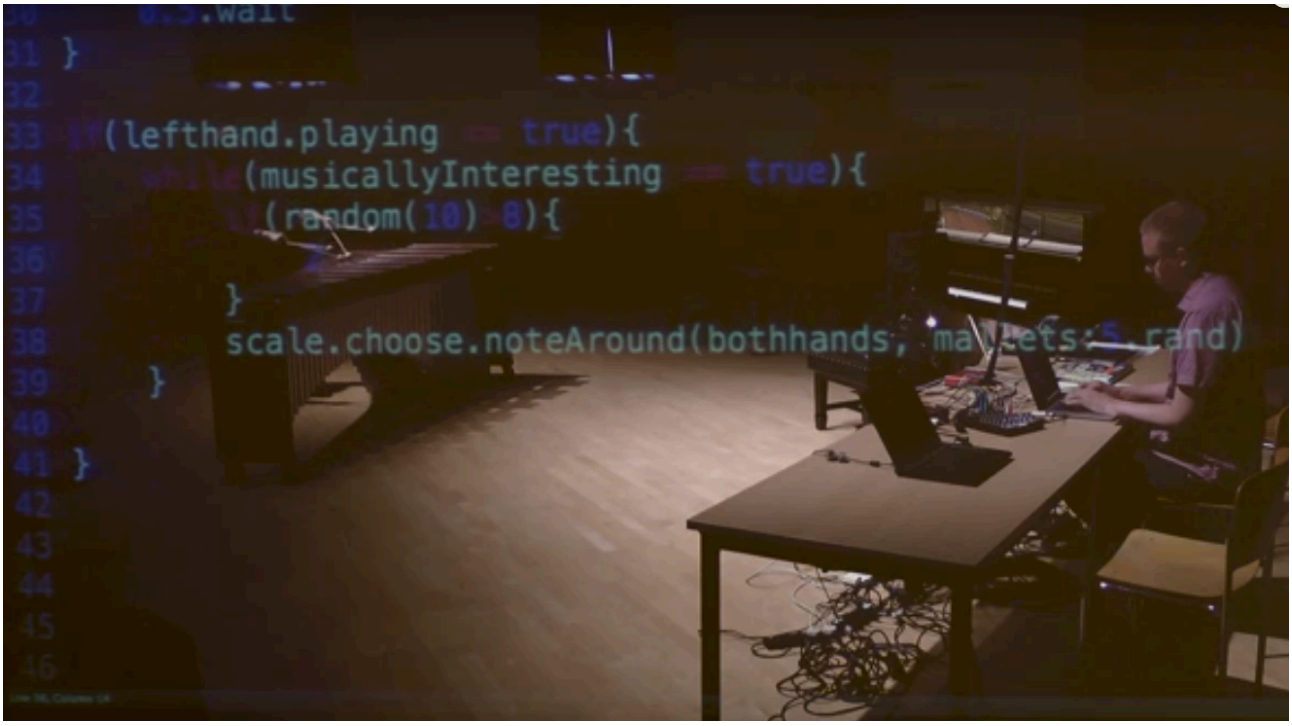
The band SharedBuffer is an improvisation quartet composed by Eldad Tsabary, Alex McLean, Alexandra Cárdenas and David Ogborn who are globally distributed and play over the internet by a networking system called Extramuros, written by David Ogborn. This software allows them to work together on a piece of code by using a browser connected to the Internet. "The performers are connected via various pieces of software for Internet-mediated sharing and collaboration, including parts developed for this ongoing project. Shared Buffer Study #X is part of a series supported by the research project "Live Coding and the Challenges of Digital Society" (McMaster University Arts Research Board)." (Ogborn, 2014)

The screen for every member of SharedBuffer, Piksel 2014

Networking has brought a new idea of space to electroacoustic music. Musicians don't require to be physically together in order to be able to perform as an ensemble. Location can be transposed to the cyberspace. One live coding musical instrument can be located on the Internet and each member can have a personal instance of the instrument in their own physical space. Cyberspace is now a new space for electroacoustic music.

Networking abilities and their relationship to space are being explored in other ways. For instance, when live coding a Disklavier that is also being played by a pianist. In their Improvisation for Pianist (Disklavier) and Live Coder, Juan A. Romero and Anne Veinberg played at the first ICLC. "The improvisation consists of mutual feedback using the Disklavier as an acoustic and mechanical interface between the pianist and the live coder. Phrases and notes will be recorded and algorithmically transformed and played back through the same device interfering and interacting with the player. Some sound synthesis will be also live coded to expand and adapt the sound of the piano and the playing style of the perfomer."[2]

[2] Juan A. Romero's description of the piece



Encoding the marimbist, Thor Magnusson, 2015

In *Encoding the Marimbist*, Thor Magnusson composed a piece for live coding and marimba in which the instructions for the marimbist where live coded. This is another example of the live coding instrument without using loudspeakers. The score of the piece was live coded in real life by the composer and projected so the audience and the instrumentalist could follow it. “In this performance the marimbist Greta Eacott will perform real-time generated musical notation in the form of code. The coding language called CMU (Code Music Notation) is a notational language for human interpreters and thus different from traditional CUI's (Code User Interfaces) written for machine interpreters. CMU is an object oriented programming language with a C-family syntax and dot notation, also supporting functional approaches, such as first class functions and recursion.”[3]

[3] Description of the piece by the composer

Type a personality, by Nick Collins is another good example of the possibilities of live coding in electroacoustic music. “Type a Personality is a score for pianist with interwoven typing as well as live coded synthesis engine electronics part. The control sequence for the electronics part is a direct result of Anne’s scripted typing; the inevitable errors in typing certain keys under time pressure, and the inter-key time intervals, directly set the state and memory values of the audio engine. The exact text to type out is different for each performance, but watch out for a possible extract from the TOPLAP manifesto.”[4]

Type a personality, Nick Collins, 2015

The pianist, Anne Veinberg, has to type a text in a computer keyboard while playing the musical score for the piano. The different levels of difficulty imposed on the pianist trigger mistakes and typos that will define the result of the piece in all aspects.

[4] Description of the piece by the composer

Grenzgänger 2
premiered in Berlin in 2015 as part of Unerhörte Musik

My own piece for accordion and live electronics “Grenzgänger 2”, is a mixture of fixed structures and improvisational passages for both the accordionist and the live coder. In this completely written score, there is still flexibility to explore the space and the instrument during the performance, both for the accordionist and the live coder. This duo creates a whole meta-cyber-instrument that no matter how many rehearsals have taken place beforehand, random patterns in the synthesis of the sound and in the behaviour of filters and other effects used on the accordion will create a surprising and unexpected musical result with different challenges for the performers every time the piece is performed.

3. LIVE CODING GENERATIVE MUSIC

"From the standpoint of Taoist philosophy natural forms are not made but grown, and there is a radical difference between the organic and the mechanical. Things which are made, such as houses, furniture, and machines, are an assemblage of parts put together, or shaped, like sculpture, from the outside inwards. But things which grow shape themselves from within outwards—they are not assemblages of originally distinct parts; they partition themselves, elaborating their own structure from the whole to the parts, from the simple to the complex" (Alan Watts, 1958)

Generative music is a place where the emotion of music meets the practicality and logic of programming. And live coding is the practice of making generative music in real-time. In live coding we take strict and logical machine processes and subvert them to create illogical, expressive, and unexpected results. Generative music is not composed exactly in the way music was composed before, since it is no longer built to fit a prescribed musical form. Generative music is grown from a set of initial variables and conditionals that will develop and take shape, in most cases, outside of the capacity of cognition of the composer. The composer plants a seed, with the required information for the computer (a machine with enormous computation capacities) to generate a musical piece that resembles more a sentient being with its own life and consciousness than an inert piece of fixed material.

Generative music can only be described, not prescribed. It is self-structuring, much like organic life. This music doesn't conform to established designs or pre-approved musical forms. This music creates its own forms. When the electroacoustic composers mix in their creations the chaos and the unpredictability of music with its emotions, sounds, and structures with the apparently cold world of programming, they are giving birth to a holistic cyber-instrument that can create a type of music that can only be pertinent for the post-digital era: a music that is alive, ever changing, with flexible and infinite, multidimensional structures that would be almost impossible to predict or even explain by unaided human minds, but that carries in itself enormous possibilities of becoming well beyond the limits of any idea of fixed musical form.

The apparent dichotomy between the natural and the artificial, the logical and the emotional, and the simple and the complex, is seen under a new light in generative music. They are not opposites, but interact in different ways. Life itself evolves between entropy and order, and music, in order to reflect life itself, must do the same.

In generative music we take the programming tools, which are usually invested with a boolean categorisation: things are right or things are wrong, and use them for what they were not designed for. For a generative musician there are no rights and wrongs, it's all a matter of personal taste. Generative music is there to break the rules. Except for the software and hardware rules. Those rules can't be broken or even bended, but they can be subverted. And this is what live coding is about.

In generative music the composers strive to express their individuality, and also the chaotic characteristics of the processes they have created and that finally grow out of their control. Generative musicians are artists of chaos, and live coders create and control this chaos as a performance. To achieve this, they must embrace the unexpected, overcome the fear of failure and harness both the unexpected and the fear in order to create unparalleled works.

Here lies the healthy tension between programming and arts. Both in traditional music and traditional programming practices, the error and the unexpected are not welcome. In live coding, they are not only welcome but needed as part of the creative process. Live coders learn to be comfortable with the lack of control over their work and use it as fuel for their creations.

"The machine makes the music, but I created the machine... I don't know where responsibility lies in that situation."[5] (Sean Booth, 2010) Much has been said about whether programming a machine to make music is or not composition. Live coders have understood that designing the initial conditions of a program is a composition. Generative music can be made without computers, therefore, we can say that 4'33" is Cage's composition, even though he only set the initial variables for the piece. 4'33" is a generative piece of music. We can go even further to explain that when programmers write a piece of software for making music, they become composers themselves. All the music that will be played using these softwares, will be influenced by the frame of mind and the election of the conditionals and variables of their writers. Live coding questions one more time the clear line between composer, technician and performer.

The generative music composer is a designer of the initial settings of the piece, a curator of its behaviours, and the biggest parameter to control the piece is the aesthetic judgment of the composer. The machine does all the hard work of realising computations and following the rules at incredible speeds. The composer commands the intention of the piece, not the exact result.

This role won't be changed until the machines themselves can develop an aesthetic sense, but this is soon approaching, as Artificial Intelligence and Machine Listening are starting to be part of the design of future live coding systems (Collins, 2016). But even if the machines become composers themselves, the human composer will always be at the heart of the systems.

The seeds of generative music are algorithms. They are part of the natural world and their essence transcends any medium, hence, algorithms remain the same, no matter the technological advancements.

With the speedy advancements in technological equipment, computer musicians face a new problem: the permanence of their tools. A violin was developed for centuries until it reached the pinnacle of its evolution. During that time, a big repertoire and a big tradition of codes and contexts were transmitted for generations. Music that was written centuries ago, can be played in a contemporary violin. But what happens when music that was written for a certain computer instrument a couple of decades ago, can't be played any longer in the most recent computers? The subject of obsolescence and impermanence is a delicate one for computer musicians. Therefore the transmission of ideas goes beyond the mere writing of a score… codes and algorithms are shared in order to transmit the ideas successfully, in the hopes that they can be recreated with the tools of the future.

---

[5] Autechre in interview, 2010

Live coding provides a unique perspective on the generative landscape, especially, because the generative algorithms are presented in code, and this supports the interaction with the machine during a live performance. (Brown and Sorensen, 2009) In the creation of a live coded performance, it is very efficient to design generative processes that create music in an autonomous way, allowing the artists to put their attention in the modification or creation of different parts of the musical discourse. The generative processes in live coding should be succinct, computationally efficient, responsive and modifiable.

The way a live coder deals with code demands reflexion in action, and this has proved to be very valuable in creative tasks. Generative processes with computers facilitate this in unprecedented ways. Through generative processes, live coders can make use of the computing power of the laptop by creating substantial musical turnouts produced by algorithms. "Typically, there is considerable leverage in this process where the effort of describing the algorithm is minimal compared to the effort that would have been required (if it were even possible) to describe directly the musical material that is produced. " (Brown and Sorensen, 2009).

In a live coded performance, an algorithm is rarely left alone. It is constantly revisited by the live coder to be modified, changing then the development of the music. The results of generative processes may not be predictable, therefore the interaction with algorithms is also improvisational itself.

In this example of a live coded rendering of *Piano Phase* by Steve Reich, we can see in the end result the algorithmic simplicity of the piece: two instruments play the same material, but one of them is shifted every once in a while. Here, the shifting process was automated with the function "future".

The generative process exists in two levels in live coding: the unfolding of the algorithms themselves and the interaction of the live coder with the algorithms. The complexity of the algorithms plays a determinant factor in their use in a live coded performance. If an algorithm is too complex, the performance may suffer by the amount of time the live coder needs to write it, and also by the increased possibility to make a typing mistake. Moreover, this complexity may reduce the possibilities of future modifications of the algorithm.

Abstraction is another element that plays a big role in the live coding performance. Software written for live coding are not written at machine-level, but are abstractions that allow the programmer to write music without dealing with the complexity of the operational system. Each live coder makes personal decisions about how to get involved with abstraction. Most of these decisions are not taking having the audience in mind, but they are taken with the idea of being able to express musical ideas in code.

Efficiency, related to the constraints of the musical instrument, plays yet another big role in the writing of algorithms in live coding. Every computing system has its own demands that must be taken into account. The live coder must be aware of the capacities of the system and how the algorithms may interact with it.

4. EMERGENCE AND STOCHASTIC PROCESS IN LIVE CODED MUSIC

"The machinery of the world is far too complex for the simplicity of men" Jorge Luis Borges

Emergence is a natural phenomenon very well studied by complexity theory at the end of the 20th century, though this term was coined by Georg Henry Lewes in the mid-19th century. In this phenomenon, a simple rule set at a low level, creates complexity at a higher level. It can be easily observed in the movements of a flock of birds, where each individual behaves following only three rules, yet the movement of the whole flock describes an enormous total complexity.

Emergence happens in generative music too. Complex musical pieces can be created with a set of simple rules. Simple algorithms can create an enormous complexity. In live coding, this complexity is mediated and modified by the live coder, who even in the impossibility of description of the result of the algorithms that are being used, may interfere and make decisions based on his aesthetic sense and the overall understanding of the system he's controlling.

Take for example the software TydalCycles[6] written by Alex McLean. This software is one of the few written with functional language that are used for live coding (most of them are object-oriented languages). The main focus of this software is to create and manipulate musical patterns in a live performance. Therefore, the internal structure of TidalCycles can be thought as patterns that behave in a circular way. Much as classical Carnatic music is conceived. A typical process of creation and manipulation of a pattern in this software begins with a very simple description that is being changed step by step into patterns so big and complex that at some point it is not possible for the live coder to know what will be the exact results. Still, the live coder knows the general behaviour of the pattern and this is enough for making music.

An example of a typical possible progression of pattern modification in TidalCycles:

d1 $ sound "bd"

d1 $ sound "bd sn"

d1 $ sound "[bd sn sn] cp"

d1 $ sound "[bd sn sn] cp"

[6] https://tidalcycles.org/

```
d1 $ sound "[bd bd] [bd [sn [sn sn] sn] sn]”

d1 $ sound "[bd [sn sn]*2]/2 [bd [sn bd]/2]*2"

d1 $ rev (sound "[bd bd] [bd [sn [sn sn] sn] sn]”)

d1 $ density 4 $ sound "bd*2 [bd [sn sn*2 sn] sn]”

d1 $ every 2 (|=| speed "2") $ sound "bd*2 [bd [sn sn*2 sn] sn]" |=| speed “1"

d1 $ every 3 (|-| up "3") $ every 2 (|+| up "5") $ sound "bd*2 [bd [sn sn*2 sn] sn]" |=| up "2 4 5"
```

In this way of programming, every sounding element follows a set of simple instructions on its own, yet the full resulting compound of sounds is too complex to be precisely described. The cleverness of the software resides also in the possibilities it offers live coders to design very complex algorithms within a comparatively simple syntax.

When patterns behave in such a complex fashion from very simple rules dictated by the composer, the emergence phenomenon appears in unexpected ways, creating yet another level of musical patterns that are by no means under the direct control of the composer, but under the control of what could be perceived as randomness. Events can coincide, creating moments of insurmountable beauty or unexpected silences or even glitches, none of them being specifically planned by the live coder. With simple algorithms, a live coder can create complex emergent behaviours in music.

Alan Turing’s final work was on the subject of morphogenesis: how patterns formed in nature through emergent rules. It was considered dangerously leftist in the 50s, because it implies that humans could become a self organised collective with no need of governmental institutions. It is only now, that thanks to the Internet and our new perceptions on organisational distributed nodes, that emergence is regaining its political importance (Pearson, 2011).

In the mid-20th century, Iannis Xenakis introduced the use of stochastic processes in musical composition. Not long afterwards he started using the computer as a suitable tool to compute and automate the enormous amount of operation these processes require. A few years later he started using probabilistic methods to synthesise sound, and finally in the 80s he developed the stochastic synthesis. Xenakis was clear about his intentions at creating these processes: he wanted to offer an alternative to the linearity of serial music and the possibility to create and articulate sound masses inspired by the musicality of natural events throughout the recognition of the stochastic laws that govern them.

Stochastic processes have been of enormous importance in computer music, by allowing it to become non-linear and generative. Computers can make decisions in two ways: deterministic and stochastic: deterministic process leads to an output after carrying out fixed tasks which don't involve random selection, a stochastic process integrates random choices. Stochastic processes can be used in different levels, at the sound synthesis level, it can help with giving the computer sounds a more "human" feel. These processes help with the efficiency of the computational process by reducing the amount of calculations needed to achieve a certain result.

Aleatoric are the most simplistic of the stochastic processes. They generate a random number between given values. By changing these values over time, one can create "tendency masks", which are widely used for granular synthesis techniques. Live coders can make use of tendency masks to control the behaviour of musical patterns in real time.

Probability distributions are methods also used for live coding. In this way, the artist can be sure that certain outputs occur more often than others. One can make suggestions to the system on the occurrence of the musical events, but the unexpected element is always a possible presence, making the dialogue with the machine a more organic one.

Different types of noise generators and Markov chains, automata, chaotic systems, and fractals are stochastic tools that live coders implement in their music. This shows an overview of the possible way of dealing with algorithms in computer music for a live performance. Many complex and simple algorithms can work independently and are mediated by the live coder.

5. LIVE CODING AND HACKER ETHICS

a. Hacking sounds

"Computers are bringing about a situation that's like the invention of harmony. Sub-routines are like chords. No one would think of keeping a chord to himself. You'd give it to anyone who wanted it. You'd welcome alterations of it. Sub-routines are altered by a single punch. We're getting music made by man himself: not just one man." John Cage, 1969

The first hackers were a group of dorky train enthusiasts at the Massachusetts Institute of Technology. Once they got in contact with a computer, they made history by exploring its computational capabilities. From then on, the term "hacker" is used to designate someone who is curious, talented and passionate about technological systems. Far from its "Hollywoodesque" rendition of a savvy criminal, the hacker is, in reality, more a very talented programmer that is excellent at finding solutions and at developing new systems for individual and communal expression.

“A program in execution is a program in motion. Control flows around the program, taking data with it. Data flows into the program, and is breathed out again. A hacker staring intently into her screen is probably turning somersaults in her mind “ (McLean, 2007). Each programmer has their own style and approaches the act of programming in an individual way. In the same way, the way they share their code with the world involves a personal decision. Still, programs are written in a collaborative manner, the programmer is indebted with the writers of the operative system he uses and of the environment he is working on. A programmer can be seen as a creative individual, rather than a technician, it is then when the computer stops being a lifeless box and becomes an environment made for and by human expression.

Hackers and artists share the deep passion of their ideas and the commitment to finding the most creative solutions for their personal expression. It is not a coincidence that live coders have been greatly influenced by the ethics of the hackers, shaping the philosophy of live coding in every level.

“Hackers believe that essential lessons can be learned about the systems of the world from taking things apart, seeing how they work, and using this knowledge to create new and even more interesting things. They resent any person, physical barrier, or law that tries to keep them from doing this.” (Levy, 1884). Hacker ethics are well formulated by Steven Levy in his book *Hackers: Heroes of the Computer Revolution* of 1984. He summed them up in 6 important points that I will compare with live coding next:

1. **Access to computers should be unlimited and total**

This tenet is not limited to computers. Access to any system and anything that might be a source of knowledge should be unlimited and total. This is also how live coders see their systems. Starting from the hardware, live coders are looking for ways to facilitate reach of live coding by using Raspberry Pi as a hardware to install software like TidalCycles or SonicPi. This means everyone can start live coding without the need to invest a lot of money in the hardware.

But it’s in the software where live coders have taken the biggest steps granting access to their systems, by making them open source and creating communities to develop them. Anyone interested can at any time learn to live code, make a live coding musical instrument for themselves and join the conversation of the live coding community.

2. **All information should be free**

Live coders are a community of hackers that share information in an unlimited way. Sharing their codes making use of repositories like GitHub, being active in forums and e-mail listings, and organising workshops, live coders are well known for sharing information.

3. **Mistrust authority - promote decentralisation**

Live coders' community has developed around the world in a decentralised manner, much in the way emergent communities are created. With decentralised nodes and self-organised systems, the toplap organisation has is present in different cities that organise activities around live coding without the need for hierarchical structures. Each live coder is their own authority, thus ensuring that empowered individuals are inspired to participate in the community by their own interest, and not because they are forced to do so.

The use of free software is another way to ensure that the artistic interests of live coders are not under the authority of commercial software and the music industry.

4. Hackers should be judged by their hacking, not bogus criteria such as degrees, age, race, or position.

The live coding community is recognised by their values of respect and inclusion. They welcome every hacker that wants to learn and participate without regard to their backgrounds or current situations. The respect for each other is palpable in the forums[7] of live coders, where it can be noticed how they are willing to help each other in a respectful way. In contrast with other technological communities, where noobs (newcomers) are mocked and mistreated, the live coding community is based on respect, which attracts people of all genders, ages, races, and abilities. This is also why live coding is a practice that doesn't approve of competition, recognising that each individual is different and their inputs equally important.

5. You can create art and beauty on a computer.

This is more than self-evident for live coders who have based their practice on this statement. However, for live coders, as well as for hackers, beauty lies not only in the resulting art but in the beauty of the program. The code itself becomes an aesthetic expression to be contemplated and valued in artistic ways. Live coders take pleasure not only in the resulting music but also on the thoughtful contemplation of the algorithms that their peers create, use and modify during the performance.

6. Computers can change your life for the better.

Such refined and powerful tools, when used for personal empowering and development, can do nothing but change the hackers' lives for the better. Computer music has been benefitted by varied aspects of the live coding practice, including openness and the hacking of musical systems that could have never been born under commercial standards. Live coding then, has liberated computer music from the restraints of commercial systems.

---

[7] Lurk is one of the most important forums for the live coding community: https://lurk.org/groups/livecode

The hands-on imperative, an important statement of the hacker ethics, refers to the freedom to understanding and exploring the technologies around us. This leads us to the importance of the use of open source software, which each individual can explore and change as they please.

The community is the home of the hackers. Whether this community is local or located in the cyberspace, the live coder community shares the common value of collaboration. There's not such thing as a lonely live coder. Each performance represents the collaboration between different hackers and will, in turn, represent a new platform for development for the community.

6. URBAN EXPLORATION

b. Dérive, Flânerie and Parkour as urban subversions

"Through the Unknown, we'll find the New" Charles Baudelaire, Les Fleurs du Mal

Dérive, Flânerie and Parkour share the same principle of subversion, challenging the hegemonic discourses of discipline and control in the urban environment. Dealing with the physicality and the embodiment of the human movement in the cities, they act as metaphors that question the relationship individual/society.

Flâneur is the Parisian stroller of the 19th century that explores his environment through observation and contemplation. His aim is not to live the city as the regular man, is not to use it as it was intended for, but to wander it aimlessly. Charles Baudelaire explored this figure in his writings, describing it as a dandy, an aesthete who primarily enjoys the many manifestations of the modern city life looking and listening attentively. Christopher Butler describes the flâneur: "the city's modernity is most particularly defined for him by the activities of the flâneur observer, whose aim is to derive 'l'éternel du transitoire' ('the eternal from the transitory') and to see the 'poétique dans l'historique'('the poetic in the historic')."[8]

[8] Christopher Butler, 'Early Modernism: Literature, Music and Painting in Europe 1900 – 1916'

In the 20th century, Walter Benjamin returned to Baudelaire's description of the flâneur as a starting point to study the impact of the modern life in the individual's psyche. Baudelaire's *Les Fleurs du Mal* is the starting point of Benjamin's study on modernity. The fact that Benjamin was studying a figure from the past reflects his idea that all human manifestations are transient as is modernity. In his work *The Arcades Project*, Benjamin uses two concepts to explain the human response to the modern city life: *Erlebnis* and *Erfahrung*. Erlebnis describes a shock state that the individual experiments due to the overwhelming of the senses, while Erfahrung refers to a more positive experience of wandering and enjoying what the city has to offer. Benjamin proposed Baudelaire's poetry as a mean to turn Erlebnis into Erfahrung. Memory is an important concept in Benjamin's work and is also related to the transmutation of Erlebnis into Erfahrung, in that it is in the physical spaces of the city, rather than in the psyche of the inhabitants where it can be found.

The flâneur is the observer of the city, the one who takes aesthetic pleasure in getting lost in it, wandering it aimlessly, abandoning himself in the artificial world of the capitalist civilisation. Benjamin's flâneur is the critique and the observer of the city from a Marxist point of view.

Flânerie is a way to re-imagine the city. A way to re-appropriate liminal states that humans enjoyed before the arrival of modernism. It is a ritual, a transition between states of mind. Between the daily life one, the normal one, where the subject is controlled and basically abducted, to one where the individual regains his psychological and spiritual will. Benjamin saw this heightened state achieved through flânerie as one of transcendence, one where one enters the world of advertising, but where it is stripped of its intended purposes. This was the only way to overcome the effects of modernity, at least temporarily.

In *The Arcades Project*, Benjamin writes about mirroring spaces. Observing these highly commercial buildings, he mulls over the mix of interior and exterior spaces, wondering for instance if the outside café tables inside an arcade were really indoors or outdoors. His focus on the spatial made him suggest that the flâneur experiences the city as if he was indoors. The flâneur transcends the modern imposition of feeling unprotected outside and in this way, he re-appropriates the city by changing the focus of his perception.

For Benjamin 'dream spaces' are bookstores, shops, and cafés that act as extensions of the street. Gambling was for him a leisure occupation that also invested the individual with the promise of a utopian dream. The flanêur is the ultimate observer of the modern city, of the commodity-obsessed marketplace. He doesn't challenge the system, but he knows how to overcome it. His tool for achieving the transmutation from Erlebnis to Erfahrung is Einfühlung.

Flâneurs ignore the rush hour, they instead hang in places where they are not supposed to stay. They represent the resistance of the daydreamer to an authoritarian system that finds a free will individual subversive. Their role is mainly symbolic. The wandering spirit of the flâneur can be seen in the intellectual spirit of the bohemian intellectual. The Parisian 20th-century cafés were the meeting point for flâneurs and bohemians.

Benjamin opened a path that moved from the early modernists, to the surrealists and moved on to the situationists. As part of the Situationist International, Guy Debord developed the notion of the Dérive. A dérive is an unplanned walk through the city in which the individual can achieve a psychogeography, that is, the walker creates a mental map of the city in which he is drawn into events and situations.

Debord looks to bring together his Marxist ideas with Benjamin's ideas and in doing so, the spirit of the flâneur has been kept alive through the impulse of the situationists. In the 21st century, the citizen still faces threats of shock states due to the overwhelming over stimulation of the city life. The city is negated to the citizens. The public spaces are seen as a commodity by the corporations who see the citizens as potential customers ignoring their primal needs and investing them with artificial ones. Many battles have been fought around the world for citizens to regain the right to appropriate the cities. But is mainly through philosophical movements that advances have been made in this subject.

Parkour, the art of movement, developed in the late 20th century in France, is an increasingly well know practice that nurtures from flânerie and dérive and takes the ideas of transcendence of the capitalist system into the physical realm. The city is not only the collection of spaces, memories, and events to be observed or photographed, it is now the playground where individuals move, create physical displacements within an aesthetic and philosophical intention of strength, individuality, and empowerment. The citizen touches the floor of the city, becomes one with the pavement, enjoys the walls, the stairs, the ups and downs of the architectural space that was created with completely different intentions. In this way, the traceur[9] re-appropriates the city and circumscribes himself in an environment that can be used at his will through his creativity and his physical strength. In Parkour, movement is the central idea. Through movement, the borders between the individual and the objects in the environment are blurred. Parkour opens possibilities to reshape given configurations of places through its relationship to movement. Parkour is a deterritorialising practice that challenges the conceptions of the capitalist urban space and its authoritarian restrictions, allowing the traceur to reclaim the case of the space and time in the city following nothing but their will, creativity, and physical and mental capacities.

[9] the male practitioner of parkour is referred to as 'traceur' while the female is a 'traceuse', meaning 'the one who traces' a route in space.

Recent attempts of reterritorialization of the practice (recognizing it as a sport in the UK), will not take its inherent subversive character away. Parkour is a mainly a transitional urban artistic expression whose main philosophical aim is the strengthening of the individual to become a useful member of society. “Be strong to be useful” is the motto of parkour. But what is useful in this subversive practice? the idea of a useful individual for parkour is quite different from the idea of a useful individual in the capitalist system. A traceur is one that is able to overcome obstacles to reach his desired goals. The traceur shares then, important philosophical views with the hacker. Traceurs create also communities, both physically and online, to learn from each other and help each other to become stronger, to share with each other information on interesting spots for training and techniques to overcome obstacles. Obstacles are the fuel of the traceur, they are his muses, his enemies to be conquered and his raison d’être. The traceur is the hacker of the public space. The one that finds his relationship to his environment as the key to the knowledge of the world. And in the same way as the hacker, one that feels that knowledge must be accessed freely.

The traceur trains not only if physical prowess, but also in self-responsibility. A traceur is a responsible individual whose interest and passion make him part of his collective. Parkour is also an emergent community. Traceurs share with the hackers the curse of the “hollywoodesque” interpretation by the media. Traceurs are seen as the thieves and trespassers, the quintessential criminal of the capitalist system. In an authoritative system, a group of individuals that awaken from the state of shock is perceived as a threat. Traceurs and hackers also share the generosity and the inclusivity of their communities. The mediatic display of parkour has certainly helped to portray the Parkour practitioner as a strong and young male. But the truth is that, like the hackers, Parkour is practiced by people of all genders, ages, races and backgrounds.

Traceurs enter in a dialogue with their environment. They explore the city in the look for the perfect spot to practice their repertoire of movements, invent new ones, share with other traceurs or be alone with their bodies in the city. Parkour is not an extreme sport. In its origin, Parkour is not even a sport, it is a highly precise and physically demanding practice that is self-described as an art. It is about finding the perfect balance between comfort and challenge, safety and risk, fear and motivation. Traceuses activate the endless potential of the spots they find, being they abandoned places, or busy plazas. The name Parkour derive from the military practice of obstacle course called ‘parcours de combatant’, and this appropriation, from a military technique to a subversive one, contextualises parkour in a deterritorialising practice from its conception.

Parkour looks for alternate ways to deal with architecture. Architecture stops being merely visual and prescriptive, and its function becomes one of a dialog offerer through tactile interactions. Architecture is to be felt, touched and used to develop our physicality, to heighten our senses and develop new ones through connections that were not possible to discover before. The looks of bewilderment and awe from non-practitioners, and the frowns upon by authorities, make the traceurs proud and invest them with a sense of inspiration.

Kat, Mexican traceuse

A Parkour practitioner perceives the city in a different way, with an intimate sense of belonging and the certainty of possibility. Every space can be conquered, every obstacle can be met, at least in in speculations. The traceur discovers the city through a need of freedom and self-empowerment and transcends the authoritarian environment through an active observation and interaction with the city, much as a post-human flâneur.

7. STREET CODE

Street code is a series of performances of electroacoustic music in the public space. It deals with the live coding action in an urban environment. It raises questions regarding the abilities of live coding to explore urban environments through its visual and aural characteristics. In the spirit of flânerie, of transcending the rigours of our capitalist environment, and the hacker vision of deterritorialisation and reappropriation, through *Street Code* I look for strategies to generate spaces for the recontextualisation of codes in the urban environment, shedding light into the public space as an ideal environment for the exploration of live coded music through its spaces, its codes and its subversions.

https://youtu.be/0Vh7xLZRI78

Street Code logo by Rodrigo Velasco

c. Live coding and spatial explorations

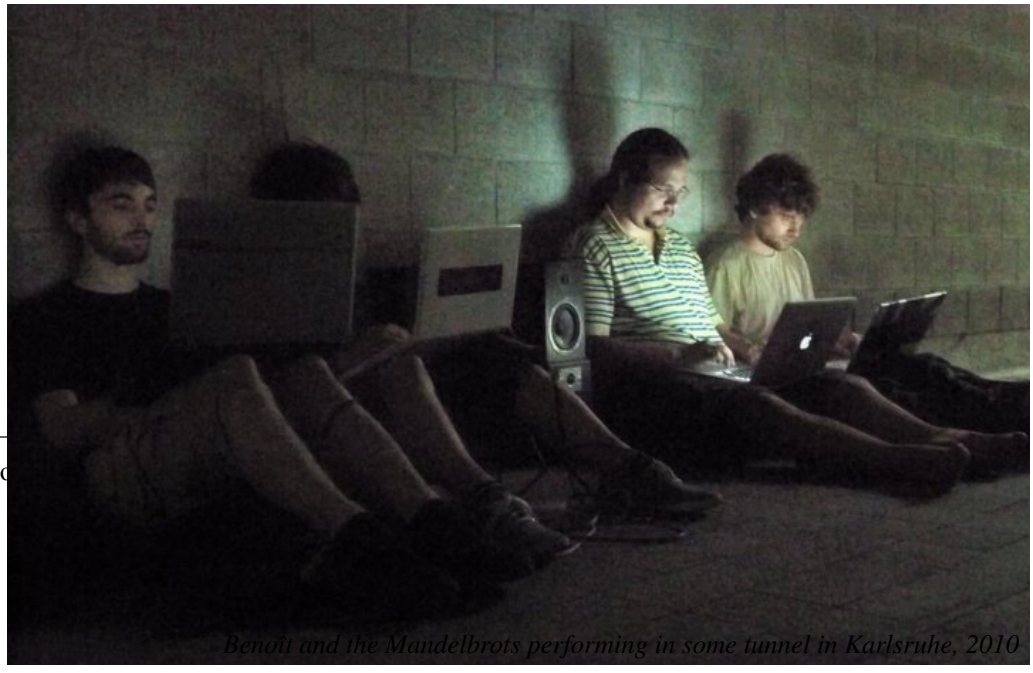
Antecedents of live coders exploring the space in their performances have accompanied the practice since its beginning. A very important factor to keep in mind is that the cyber-space is now one to be considered in electroacoustic music and much of this is caused by the live coders' explorations. The social space that an internet community share to develop and create together is material of extensive investigation from the musicological and ethnographical points of view. For now, suffice to say that the fact of live coding being born and developed in the cyber-space, endows live coders with an intrinsic curiosity and challenging way to explore space.

Surveying live coders in the past months with the intention to discover prior examples of live coding in the public space for *Street Code*, I have found a varied list of examples of exploration of space in live coding:

• Marije Baalman, 2015. As part of a multidisciplinary action called *Metatopia 1.0* taking place in Madrid, Marije live coded music in the Plaza Reina Sofía while dancers and visualists improvised too. Marije hanged a laptop from her neck, strapped a small projector to her arm, and was connected to portable speakers. This is possibly the first time live coding has taken place in the streets.

Marije Baalman, live coding in the Plaza Reina Sofía in Madrid, 2015

• Powerbooks unplugged[10], the ensemble for algorithmic network music, has explored space in many forms. One of his members, Julian Rohrhuber, wrote me when asked about live coding and space antecedents, on a live coding forum in the live coding Slack channel “"as powerbooks unplugged we played in the garden of a mountain primary school, a museum, of course on trains, and – most importantly! At home.”



Benoît and the Mandelbrots performing in some tunnel in Karlsruhe, 2010

[10] https://t

It is worth mentioning that bars have been one of the unusual venues for electroacoustic music that have been explored by live coders. With the Algorave (algorithmic rave), the goal is to live code music for dancing. Therefore bas and clubs have hosted different live coding events.

- Benoît and the Mandelbrots, a live coded band from Germany, has done varied explorations in space, including the semi-public space.

- Alex McLean and Shelly Knotts, 2015. As Alex McLean described it to me in the Slack channel "Shelly and I live coded in a quarry with curious passersby and sheep"[11]. Live coders have performed in nature, in camps, in outdoor festivals, in tents…

live coding projections in nature

- The Cybernetic Orchestra from Canada did a 1 hour "Algoskate" at the Hamilton's Harbour front in 2015

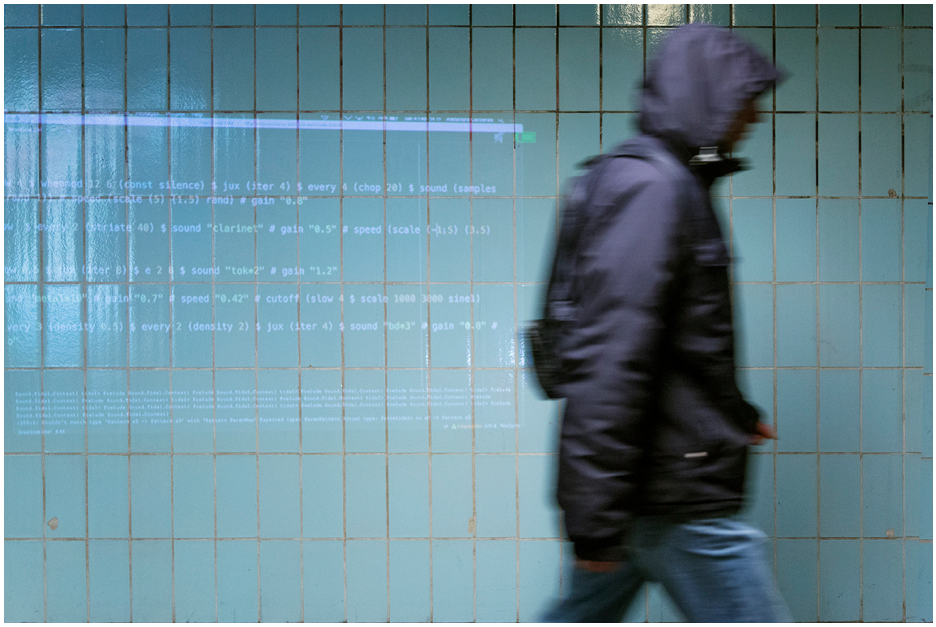
- Joseph Wilk, 2014-2015. He explained his explorations on the Slack channel: "I tried some busking live coding in Berlin. Just some mini-rigs and a pretty crappy projector. Was fun if not very successful and interesting people :slightly_smiling_face: I also tried performing in odd places in Berlin, more an effect on myself performing than of effecting people around me. I'm afraid I don't have a very good art explanation of what I was doing, just that music is all about places and memories to me so I wanted to explore that. I also took some silly photos outside performing :slightly_smiling_face:"

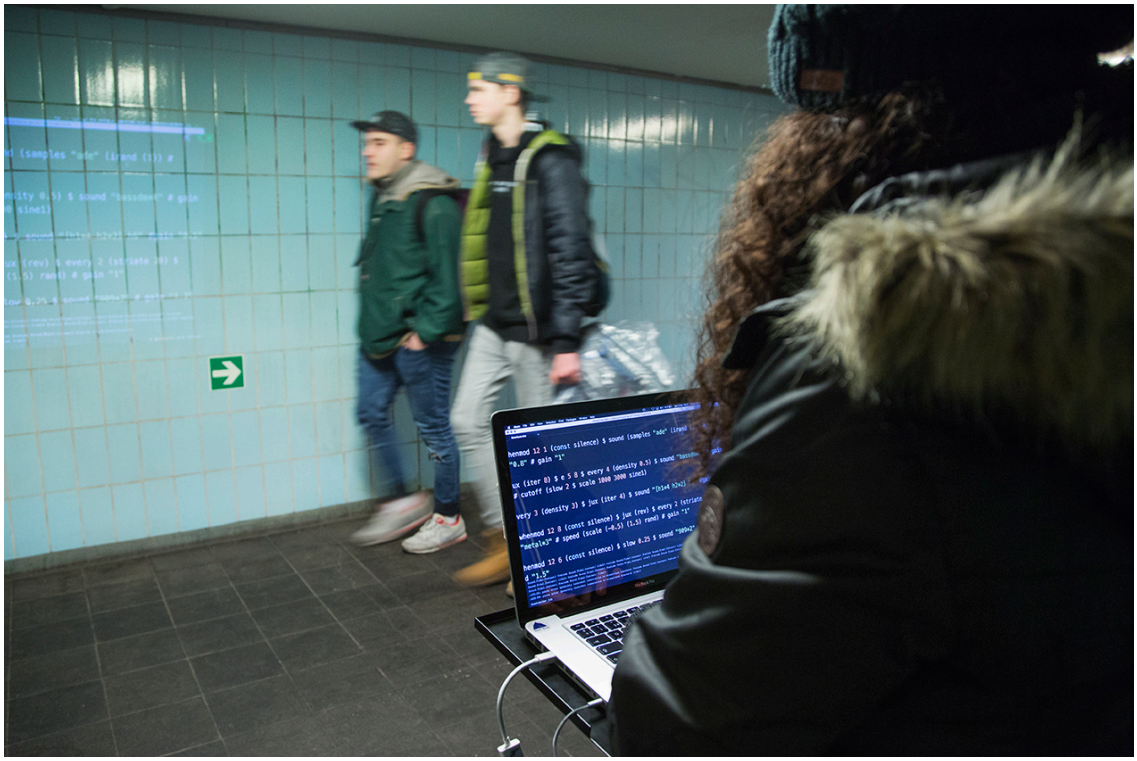
[11] http://inhabitingthehack.github.io/2015/09/07/uncanny-valley/

John Wilk doing live coding in public spaces in Berlin

Pictures of the premiere of Street Code

Alexandra Cárdenas, Berlín, February 2017





8. CONCLUSION

Live coding music performance has the ability to affect the public space and be affected by it through the recontextualisation of artistic codes and relations in our society. Audience can engage in different levels that are proposed by the poetics of code that live coding deal with.

Through the subversion of the use of the public space, I propose live coding as a healthy practice to re-appropriate the public space and to revisit stalled aspects of the exploration in electroacoustic music. I propose the performance of live coded music as a transitional form of sound installation and as a creative way to de-territorialise the urban architecture.

In the spirit of the hacker philosophy of sharing, transparency and inclusion as means to reach the new, I propose new aesthetic perceptions of the music performance using current technology and media as tools to expand our knowledge and explorations of the urban environment. In a practice that puts together what has been at times considered as disparate, live coding allows for art and science, movement and architecture, academic music and street music, sound installation and randomness to propose new tools to reconsider the contextualisation of art and with it the perception of our own environment.

9. REFERENCES

- Collins, N., McLean, A., Rohrhuber, J. & Ward, A. "Live Coding Techniques for Laptop Performance", *Organised Sound*, vol. 8 (3). 2003. pp. 321–30.
- Collins, N. "Live coding of Consequence", *Leonardo Journal*. Vol. 44 (3). 2011.
- Magnusson, Thor. Improgramming: http://improgramming.wordpress.com
- Collins, N., McLean, A., Rohrhuber, J. & Ward, A. "Live Coding Techniques for Laptop Performance", *Organised Sound*, vol. 8 (3). 2003. pp. 321–30.
- Anderson, Benedict. 1991. Comunidades Imaginadas. Reflexiones sobre el origen y la difusión del nacionalismo. Mexico. Fondo de Cultura Económica.
- Stallman, Richard. The Hacker Community and Ethics: An Interview with Richard M. Stallman, 2002 http://www.gnu.org/philosophy/rms-hack.html
- Levy, Steven. *Hackers: Heroes of the Computer Revolution* (ISBN 0-385-19195-2)
- McLean, Alex. Making Programming Languages to Dance to: Live Coding with Tidal. 2014
- Born, Georgina. Music, Sound and Space. Cambridge University Press, 2013.

- Turner, Victor 1969. “Liminality and Communitas”. The Ritual Process: Structure and Anti-Structure. Chicago. Aldine Publishing.

- McLean A. (2011).Artists-Programmers and Programming Languages for the Arts. Goldsmiths University of London: Londres.

- Wilson, S., Cottle, D. Y Collins, N. (ed.). (2012). The SuperCollider Book. Massachusetts:The MIT Press.

- Baalman, Marije. Embodiment of code, 2016

- Mooney, James. HughDavies’s Electroacoustic Musical Instruments and their Relation to Present-Day Live Coding Practice:Some Historic Precedents and Similarities, 2016

- Bell, Renick and Parkinson, Adam. Deadmau5, Derek Bailey, and the Laptop Instrument – Improvisation, Composition, and Liveness in Live Coding, 2015

- Magnusson, Thor. Confessions of a live coder, 2011

- Knotts, Shelly. Algorithmic Interfaces for Collaborative Improvisation, 2017

- Brown, Andrew and Sorensen Andrew. Interacting with generative music through live coding. 2009

- Collins, Nick and d’Escriván, Julio. The Cambridge companion to Electronic Music. Cambridge University Press. 2007.

- Zeilinger, Martin. Live coding the law: Improvisation, code and copyright. 2014

- Magnusson, Thor. Algorithms as scores, coding live music. 2011

- Blackwell, Allan and Collins, Nick. The programming language as musical instrument. 2005

- Magnusson, Thor. Affordances and Constraints in Screen-Based Musical Instruments. 2006

- Cascone, Kim. THE AESTHETICS OF FAILURE: 'PostDigital' Tendencies in Contemporary Computer Music. 2017

- Baranski, Sandrine. Música y complejidad: la voz de la red según Chris Brown.

- Debord, Guy. Theory of the Dérive.

- Boutin, Aimée. Rethinking the Flâneur: Flânerie and the Senses. 2012

- Birkerts, Sven. Walter Benjamin, Flâneur: A Flanerie. 1983

- Pearson, Matt. Generative art. 2011

- McLean, Alex. Hacking sound in context
- Baudrillard, Jean. Simulacra and Simulations. 1988
- Cage, John. Silence. 1961
- Xenakis, Iannis. Formaized Music. 1992
- Baranski, Sandrine. La utilización de la emergencia en la creación musical contemporánea como fenómeno psicoacústico complejo
- Luque, Sergio. Stochastic Synthesis. Origins and Extensions. 2006
- Supper, Martin. Elektroakustische Musik und Computermusik. 1997
- Brunner, Christoph. Nice-looking obstacles: parkour as urban practice of deterritorialization. 2008
- Ferrero, Carlos. Lo sublime de la acción: riesgo, límite y frontera a través del parkour. 2015
- Klein, Georg. Sound situations - creating perception in sound art. (2012)
- Capell, David. On Experience in the art of Janet Cardiff and George Bures Miller. 2006
- Guattari/Deleuze. A thousand plateaus. 1987
- Ablinger, Peter. No transgression.
- Cox, Georg and McLean Alex. Speaking code. Coding as aesthetic and political expression. 2013
- Columbia, David. The Cultural Logic of Computation. 2009
- Kekou, Eva and Matteo Marangoni. A New Sense of City through Hearing and Sound. 2010
- Blesser, Barry and Salter, Linda-Ruth. Spaces Speak, Are You Listening? Experiencing Aural Architecture. 2005
- Kwon, Miwon. ONE PLACE AFTER ANOTHER. Site-specific art and locational identity. 2002
- Williams, Joseph. Busking in Musical Thought: Value, Affect, and Becoming. 2016

Alexandra Cárdenas compositora/live coder

BIO
June, 2018

Composer, programmer and improviser of music, Alexandra Cárdenas has followed a path from Western classical composition to improvisation and live electronics. Using open source software like Super Collider and TidalCycles, her work is focused on the exploration of the musicality of code and the algorithmic behaviour of music. An important part of this exploration consists of the practice of live coding, including performances at the forefront of the Algorave scene and live coded electroacoustic music. Based in Berlin, where she completed the Masters Sound Studies at the University of the Arts, Alexandra travels the world giving concerts, teaching live coding of music and creating works in diverse artistic residencies. She has received support and funding from emblematic institution for the arts like CMMAS, National Center for the Arts, and FONCA in Mexico, as well as the Tokyo Wonder Site in Japan, the National Council for the Arts in Norway and most recently, Ibermusicas, institution that supported her most recent tour in South America during 2018 called HYPERSONICA. In 2017 Alexandra was the Chair of the International Live Coding Conference in Morelia, Mexico. She has collaborated with diverse artists, like Olivia Jack, computer scientist and artist from USA, Norwegian accordionist Camilla Vatne Barrett-Due, Mexican performance artist Erick Diego, Mexican percussionist Raul Tudon, computer scientist and musician AlexMcLean from the UK, with whom

she takes part of the legendary live coding band SLUB, and is part of the OFFAL collective. Besides writing music for ensembles, orchestra, and soloists of contemporary music, Alexandra has made different projects with theatre companies in Mexico, Belgium and Germany. Alexandra was born in Bogota, Colombia in 1976. Studied Music Composition at the Los Andes University, and lived in Mexico City from 2001. She currently lives in Berlin since 2013.

http://cargocollective.com/tiemposdelruido

https://soundcloud.com/tiemposdelruido https://www.youtube.-com/user/tiemposdelruido/videos https://