

# Deadmau5, Derek Bailey, and the Laptop Instrument – Improvisation, Composition, and Liveness in Live Coding

Adam Parkinson  
EAVI, Goldsmiths  
[a.parkinson@gold.ac.uk](mailto:a.parkinson@gold.ac.uk)

Renick Bell  
Tama Art University  
[renick@gmail.com](mailto:renick@gmail.com)

## ABSTRACT

Dance music superstar Deadmau5 and the improvising guitarist Derek Bailey represent, through their writing and practice, two very different approaches to performing live. By critically considering the practice of live coding in relation to these divergent approaches, we discuss live coding with regards to where the liveness lies and how the laptop and software are treated as a musical instrument. Each practice uses the laptop as a musical tool in a very different way. Live coding uses the laptop as an instrument in a manner that draws upon the techniques and strategies of free improvisation, in contrast to Deadmau5’s notion of laptop as playback device and the live performance as spectacle. We discuss Deadmau5’s practice in relation to Francisco Lopez’s notion of the possibilities of electronic performance, and ideas about labour and liveness.

## 1. Introduction

In this paper, we will explore the practice of live coding in relation to two very different but related practices: the performance practice of dance music “superstar” Deadmau5, and the “non-idiomatic” free improvisation discussed in particular by British guitarist Derek Bailey in his book *Improvisation: its Nature and Practice in Music* (Bailey 1993). For the former, the “live” show is a synaesthetic, perfectly coordinated, precisely pre-planned immersive audio-visual experience in which the performer simply presses “play”, whereas for the latter, the “live” show is continually composed in real-time. Sketching a continuum between these two practices provides a way of framing the practice of live coding. Doing so shows live coding as having more in common with ‘traditional’ instrumental practices which focus on an instrument and a performer on stage, engaged in real-time composition, than the live performer as a channel for a spectacle that is epitomised by the practice of Deadmau5. In particular, approaches to instrumentality found in free improvisation are present in live coding.

Deadmau5 is of interest to us given his prominent place in contemporary, popular electronic music combined with his controversial stance on the role of the musician in live performance, while Bailey’s performance method and writing about free improvisation, although not electronic, match the declared intentions of many live coders and therefore provide a background for understanding improvisation and its relationship to the “live” in live coding. We see Bailey as an authority on what a certain ‘pure’ idea of liveness and improvisation, which provides a valuable contrast with Deadmau5’s idea of live performance. Deadmau5 realises different notions of live electronic music practices that combine an almost Wagnerian spectacle with a dismissal of many traditional notions of virtuosity and instrumentality that might be construed as unnecessary in electronic music. In a sense, Bailey and Deadmau5 strive for opposite aims: Deadmau5 wants complete predictability and repeatability, a play button that always plays the same perfectly composed thing, whereas for Bailey each performance is a real-time composition seeking freedom from idiom and even the whole of musical history, and his practice rallies against that hypothetical play button. Neither of these aims is ever achievable, but they provide useful extremes within which to understand the practice of the live coding laptop musician.

These two views of a live performance form poles on a continuum of attitudes and ideologies of liveness, performance and musical instruments, upon which we can place and analyse live coding. Drawing on this, we suggest the following. Live coding situates the laptop as an instrument and locus of real-time composition bearing many similarities to ‘traditional’ instruments, and with the explorations of these instruments practised in relatively free improvisation. Live coding typically bears few similarities to the practices of Deadmau5, who treats the computer in a performance as a reproduction tool rather than a compositional one, despite possible aesthetic similarities (and frequently a relationship to dance music and dance music culture) that the two share.

## 2. Liveness

By examining this aforementioned continuum, it can be seen that while a designation of “live” in either case indicates performer activity as an event is underway, the type of activity in which the performer engages – composition or facilitation of spectacle – varies tremendously. The events we discuss can all said to be “live” yet in very different ways, and thus a worthwhile discussion of “liveness” can take place. Auslander has written an important contemporary text on liveness, which draws on Baudrillard, using the word “mediatized” in a fashion he describes as:

loosely... to indicate that a particular cultural object is a product of the mass media or media technology. “Mediatized performance” is performance that is circulated on television, as audio or video recordings, and in other forms based on in technologies of reproduction. Baudrillard’s own definition is more expansive: “What is mediatized is not what comes off the daily press, out of the tube, or on the radio: it is what is reinterpreted by the sign form, articulated into models, and administered by the code”. (Auslander 2008, 5)

This supports the view that rather than being a polar opposite to “mediatized” (as in Auslander (Auslander 2008; Auslander 2000; Auslander 2006)), designations of “live” are used to declare that a performer is active in some way in front of an audience, with the following consequences:

1. Liveness can be based on the prior perception of performer activity or decision-making.
2. Liveness and mediatization can co-occur. Live laptop music involves the performance of the mediatized. Mediatization may in fact amplify perceptions of liveness. From this viewpoint, audiences call something ‘live’ when they feel aware of performer decisions, typically but not always manifest in explicit physical activity in the moment of the performance... (Bown, Bell, and Parkinson 2014, 18)

While older music technologies like violins might be construed as not being mediatized (even though they may be used to repeat and mediate a score), digital music technology is almost completely entangled with the technology of reproduction. Live coding is nearly always mediatized, explicitly so according to Baudrillard’s definition, in so far as the laptop and code are commonly conceived and used as technologies of seamless reproduction. Software abstractions are precisely tools for the reproduction of certain behaviours in the computer. In the typical case, it uses the media of code, samples, synthesizers, and projection, yet its practitioners and audience have little doubt about it being characterised as “live”. Considering liveness then from the view of activity, it remains to determine what type of activity is taking place. The liveness in live coding is fulfilled through a performer’s activity in generating the sound, rather than a performer’s presence as a figurehead in a spectacle.

## 3. Deadmau5 and Liveness in Contemporary Electronic Dance Music

Deadmau5, despite the name, is neither dead nor a rodent, but a very successful DJ and producer, known for his large-scale immersive live shows, over which he presides wearing a huge mouse head. For many, Deadmau5 may be seen as the epitome of everything that is wrong with Electronic Dance Music, and the embodiment of a particularly North American strand of ‘stadium’ dance music that is far away from the evasive “underground” spirit that many value. However, whilst practitioners involved might claim there is a great stylistic and artistic gulf between Deadmau5 and live coding, there are also many similarities. Both an algorave live coding event and a Deadmau5 concert are events held at night, often in clubs, where one of the performer’s main intentions will be to make an audience dance, through the production and manipulation of electronic sounds triggered or synthesised by a laptop (Collins and McLean 2014). Regardless of any value judgements about the aesthetics or practice of Deadmau5, we believe that he cannot be dismissed and in his practice and blog posts we find a very interesting idea about liveness in contemporary electronic music.

On his tumblr (Zimmerman 2013), Deadmau5 famously declared “We all hit play”, going on to say, “its no secret. when it comes to “live” performance of EDM... that’s about the most it seems you can do anyway.” He explains the mechanics of his live set-up, informing us that he can do some ‘tweaking’, but structurally most of his set is predetermined, premixed and tied in with queues for lights and video:

heres how it works.... Somewhere in that mess is a computer, running ableton live... and its spewing out premixed (to a degree) stems of my original producitons, and then a SMPTE feed to front of house (so tell the light / video systems) where im at in the performance... so that all the visuals line up nicely and all the light cues are on and stuff. Now, while thats all goin on... theres a good chunk of Midi data spitting out

as well to a handful of synths and crap that are / were used in the actual production... which i can tweak *live* and whatnot... but doesn't give me a lot of "lookit me im jimi hendrix check out this solo" stuff, because im constrained to work on a set timeline because of the SMPTE. Its a super redundant system, and more importantly its reliable as FUCK! [...] so thats my "live" show. and thats as "live" as i can comfortably get it (for now anyway)<sup>1</sup>

He trades off spontaneity for reliability: the best laid plans of Deadmau5 rarely go awry. In doing so, he diminishes the possibility to demonstrate his own personal virtuosity, but interestingly to us this seems to be of little concern to him, and he even goes on to be actively dismissive of claims and displays of virtuosic live electronic performance;

Im just so sick of hearing the "NO!!! IM NOT JUST DOING THIS, I HAVE 6 TABLES UP THERE AND I DO THIS THIS AND THIS" like... honestly. who gives a fuck? i dont have any shame in admitting that for "unhooked" sets.. i just roll up with a laptop and a midi controller and "select" tracks n hit a spacebar.

For Deadmau5, what is important is not the ability to virtuosically do these things live and in real time, but the artistic processes involved in the composition and production of these tracks, and the immersive live show involving all the spectacles of lights and videos:

my "skills" and other PRODUCERS skills shine where it needs to shine... in the goddamned studio, and on the fucking releases. thats what counts.[...] you know what makes the EDM show the crazy amazing show that it is? you guys do, the fans, the people who came to appreciate the music, the lights, all the other people who came, we just facilitate the means and the pretty lights and the draw of more awesome people like you by our studio productions. which is exactly what it is.

We could link this to the electroacoustic "tape music" tradition, where a "live" performance traditionally involves the playback of fixed media with some live spatialisation for the event; we could also link it to rock or even opera through the focus on the spectacle. Deadmau5 declares:

"im not going to let it go thinking that people assume theres a guy on a laptop up there producing new original tracks on the fly. because none of the "top dj's in the world" to my knowledge have. myself included."

Deadmau5's approach is met with suspicion by many, especially those who want to see a musician actively doing something - effectively labouring - on stage, and for whom "liveness" might be signified by perceived performer activity in the area of composition. This does beg the question: at precisely what historical point did we expect our composers to be adept performers of their music anyway? The divisions between composer and instrumentalist are still alive in well in the world of classical music, and whilst there may be many problematic value systems and divisions of labour that accompany this, at least the role of composition is valued.<sup>2</sup>

Deadmau5's live sets blur the already murky distinctions between a "live" set and a "DJ" set (distinctions which we cannot fully explore here), and Deadmau5 suggests that most large scale electronic dance music performances operate along similar lines. The artist plays their original compositions, perhaps mixing and modifying parts, but doesn't compose new material in real time, and whilst the well established and traditional musical spectacle is alive and well, traditional ideas of instrumental virtuosity are discarded.

#### 4. Derek Bailey and Improvisation

At the other end of the spectrum to Deadmau5's joyously unashamed play-pressing, we find the musical practice of free improvisation. The free improvisation to which we refer emerged primarily in the United Kingdom in the 1960s from a nexus of workshops and regular concerts. These include the regular sessions by the "Joseph Holbrooke" trio of Derek Bailey, Tony Oxley and Gavin Bryars above the Grapes in Sheffield, the improvisation workshops run by John Stevens in Ealing College, West London, the London Musicians Collective, the work of AMM and Cornelius Cardew, and the Spontaneous Music Ensemble sessions at the Little Theatre with John Stevens, Trevor Watts and Paul Rutherford. Free

---

<sup>1</sup>We have not corrected the spelling or grammar used in the blog post.

<sup>2</sup>The issue of labour is treated further in the last section of this paper.

improvisation drew on diverse musical sources including but not limited to American free jazz (in particular Ornette Coleman), the American avant-garde of Cage and La Monte Young along with the European avant-garde of Stockhausen. Histories of the influences on free improvisation, its early days and its position in Western music history are traced in Sansom (Sansom 2001) and Prevost (Prévost 2001).

Free improvisation is clearly not fully represented by the ideas and practice of the sole figure of Derek Bailey, but as a practitioner, and one who wrote about and spoke articulately about his practice and his lifelong relationship with a musical instrument (the guitar), he remains a pioneer and a useful figurehead for the purposes of our discussion. Bailey's book *Improvisation: its Nature and Practice in Music* was first published in 1980 and remains a canonical and widely referred to text on the subject of improvisation. It is still one of relatively few texts discussing the ideas and ideals of "free" improvisation (as opposed to jazz improvisation, though the scope of Bailey's book does also go beyond said "free" improvisation).

Bailey introduces improvisation in the following way:

Defined in any one of a series of catchphrases ranging from 'making it up as he goes along' to 'instant composition', improvisation is generally viewed as a musical conjuring trick, a doubtful expedient, or even a vulgar habit (Bailey 1993).

He goes on to say of "free" improvisation that: "It has no stylistic or idiomatic commitment. It has no prescribed idiomatic sound. The characteristics of freely improvised music are established only by the sonic musical identity of the person or persons playing it." (Bailey 1993, 83). He calls it the oldest musical practice - for the earliest attempts at sound producing by humans presumably had no idiom or structure to be constrained by - and describes it as a recent movement emerging from a certain questioning of musical rules.

Whilst in the early days of free improvisation a lot of work was undertaken to escape musical idioms and convention - experimenting with escaping meter or regular tonal structures, for instance - we are drawing on free improvisation because it is a practice which foregrounds process and an immediacy with instruments and musical materials. Real time interactions and an explorations of the sonic possibilities of whatever musical materials are at hand take importance over the strict following of a score and the ability to reproduce seamless, identical versions of a piece of music that has already been written. We will see ways in which a similar focus on process, interaction and exploration of materials is also brought to the fore in live coding.

In free improvisation, the instrument is re-imagined as a pure expressive tool, capable of generating a whole range of sounds unbounded by notions of what music should or shouldn't be, and how the instrument should or should not be used. This might be achieved through practice leading to general mastery and the honing of advanced techniques such as circular breathing amongst woodwind players, but it might also be achieved by an amateur who is lucky enough not to be conditioned by years of training to think that, for example, a guitar shouldn't be played with a toothbrush. Bailey speaks of "the instrumental impulse", which is quite separate from the act of playing in terms of score-following that many musicians might do. Discussing instrument-playing amongst non-improvisers, he notes, "[creating] music is a separate study totally divorced from playing that instrument" (Bailey 1993, 98). Improvisation explores the musical affordances of an instrument and begins with the instrument, not the score. For Bailey, the instrument "is not only a means to an end, it is a source of material, and technique for the improviser is often an exploitation of the natural resources of the instrument (Bailey 1993, 99)". When Bailey says "[the] accidental can be exploited through the amount of control exercised over the instrument, from complete - producing exactly what the player dictates - to none at all - *letting the instrument have its say*" (italics our own) (Bailey 1993, 100), he could quite easily be talking about live coding, the use of generative algorithms, and the exploration of what a laptop and coding environment has to 'say' musically. These are the "natural resources" of this novel instrument.

## 5. Improvising with Computers

Free improvisation and the practice of Derek Bailey may seem an unusual bedfellow for computer music. The polemical biography of Derek Bailey uses the computer as an example of the antithesis of the permanent revolution of Bailey's playing. We are told that "what he plays is more consistently surprising than anything constructed on a computer: it happens in real time." (Watson 2004, 1) and that "nothing is quite as dull and boring and dead as knowing precisely what is going to happen - whether that is listening to computerised electronica..." (Watson 2004, 2). Elsewhere it is clear that the author has a more nuanced view of computers and creativity, but still, we see the computer being situated against the freedom and unpredictability of Derek Bailey. As we will argue, live coding makes specific efforts to reframe the computer as an instrument, as opposed to a machine of mediation, construction and predictability, and there is a whole history of using computers as unpredictable musical tools and even improvising agents.

George Lewis has been working on improvising with computers since the 1980s (Lewis 2000; Lewis 1999). Lewis's work places particular emphasis on the computer as improviser, moving more into the realms of artificial intelligence and researching "independent computer agency" (Lewis 1999, p.108) than many live coders. To an extent, his work explores whether we can treat the computer as a musician rather than as musical instrument. As he notes, in his most well known computer music piece Voyager "the computer system is not an instrument, and therefore cannot be controlled by a performer. Rather, the system is a multi-instrumental player with its own instrument," (Lewis 1999, p.103). Nonetheless, he is a pioneer of bringing computers into the sphere of improvised music and offering an alternative view of the musical role of the computer to that of the predictable sequencer. Other examples of improvising with computers which predate the live coding on which we focus in this paper would include the work of "The Hub" (Gresham-Lancaster 1998), Robert Rowe's interactive systems (Rowe 1999) and the work of Joel Chadabe (Chadabe 1984), to name but a few. Whilst these are all important in a historical framing of laptop improvisation and live coding, we will not discuss them at any length here as we wish to remain focused on the specifics of live coding: namely, real-time implementation and writing of code, and the projection of this code so the audience can see it, for it is here that the unique qualities of the practice emerge and we see the way in which the instrumental capacity of the laptop is explored.

The computer-as-instrument, and ways for expressively improvising with computers, is also widely explored in what we refer to as the New Interfaces for Musical Expression (NIME) field, where attempts are often made to extend the computer allowing for gestural control through accelerometers, biosensors and other means. One survey of this is found in (Miranda and Wanderley 2006). This is a field to which whole conferences (such as the New Interfaces for Expression Conference, the proceedings of which are another reference for any further investigations) are dedicated, and live coding often finds a comfortable place within this field. Again, it is outside the scope of our writing here to engage with much of this research, and we focus on live coding where very little attempts are made to extend or improve the input capabilities of the computer to make them more expressive: instead, the inherent, unadulterated affordances of keyboard, mouse and code are harnessed. Audience legibility is explored less through gestural legibility than through the projection of the performer's screen.

## 6. Live Coding

We will now draw on the practices of Deadmau5 and Bailey to examine some aspects of the practice of live coding. We define live coding as "the interactive control of algorithmic processes through programming activity as a performance". This practice is not homogenous, and live coding realistically refers to a range of practices and approaches with differing degrees of liveness, explored, for instance, by Church et al (Church, Nash, and Blackwell 2010). However, these practices are united by principles found in the draft manifesto (<http://toplap.org/wiki/ManifestoDraft>) (if we are to take Toplap as being representative of the movement, and for the sake of this paper, we will).

Live coders uses many of the same tools as Deadmau5 - that is, at the heart of the musical production set up there lies a laptop - but these tools are used in quite different ways, even though we might draw connections between the sonic outcomes. However, as we have seen, Deadmau5 does not believe that any of the "top djs in the world" use this tool for "producing new original tracks on the fly". Either Deadmau5 had not encountered live coding, or perhaps he was not considering live coders amongst these world class DJs, for it is exactly the idea of "producing new original tracks on the fly" that motivates much of if not all live coding. Peter Kirn, in one of the slew of aftermath articles that followed Deadmau5's blog post, brings attention to the multiplicity of practices in creating live electronic music that are "more live" than Deadmau5. (Kirn 2012) In this sense, live coding begins to have more in common with the liveness of composition in Bailey than the liveness of execution in Deadmau5.

Live coding nearly always involves - and more often than not actively encourages - improvisation and unexpected musical occurrences. Live coder Thor Magnusson declares, "Live coding emphasizes improvisation" and "[the] default mode of live coding performance is improvisation." (Magnusson 2014). There are multiple ways in which live coding remains an unpredictable musical environment to operate in. The frequent inclusion of randomness or algorithms with levels of complexity to generate musical materials which prevent the coder from being able to fully anticipate the musical output of their coding. Beyond the unpredictability of a program's output, live coders often execute their code in real-time in response to the preceding stream of sound, the sound of other players, the response of an audience, and so on. At the extreme end of the liveness in live coding is live coding "from scratch", which could be seen as an impossible ideal due to the underlying layers of abstraction on which a performance relies. The "from scratch" appellation signifies the coder's intention to perform in the moment or with their immediate presence, precisely the concept of "live" identified in Bown et al. (Bown, Bell, and Parkinson 2014) and in many senses an epitome of non-idiomatic free improvisation.

However, live coding is, in other respects, not the "free" improvisation as described by Derek Bailey. It is often working decidedly within or at least in a conversation with idioms, and trying to obey musical rules - often in situations where obeying of musical rules, such as the rhythmic structures of house and techno, is frowned upon (for instance, at ICMC).

Early free improvisation groups, such as Bailey's Joseph Holbrooke trio, were attempting to escape "the dogma of the beat", whereas live coding (for those generating dance music, at least) could be seen as trying to embrace the dogma of the beat. Nonetheless, free improvisation serves as a crucial reference point because it encapsulates an approach to instruments which explores their sonic affordances, such as we see in live coding. It also embodies the uncertainty which live coders cultivate, the uncertainty that makes it 'live' coding, and the uncertainty which distances it from the aspirations of total control that we find in Deadmau5. The approach to the laptop in live coding envisages it as a musical instrument quite different to the studio tool of Deadmau5.

## 7. The Instrument and the Stage in Live Coding

The manifesto on Toplap declares, "Show us your screens." Through projecting the performer's screen during performance, coding environments and the laptop itself are presented as an instrument, drawing on a traditional conception of instruments that might be closer to thinking about a guitar or piano than the way in which Deadmau5 and his audience imagine his laptop. Toplap notes that "[it] is not necessary for a lay audience to understand the code to appreciate it, much as it is not necessary to know how to play guitar in order to appreciate watching a guitar performance".

The showing of screens potentially allows for the demonstrations of instrumental virtuosity - what Deadmau5 mockingly describes as the "lookit me im jimi hendrix check out this solo" elements of performance - that are precisely what is absent in the performances of Deadmau5 and precisely what Deadmau5 is not interested in. This virtuosity might be perceived differently to the virtuosity of a guitarist or other traditional instruments which might seem more connected to the limits of the performer's body. Even to one unfamiliar with the playing of guitar, the sheer speed at which a performer can play can be impressive and indicative of virtuosity. The shared experience of a human body between performer and listener and an awareness of the physical limitations of that body can provide a common ground for understanding of virtuosity. Similarly, a long sustained note on a woodwind or brass instrument, perhaps achieved through circular breathing, might be perceived as virtuosic through its direct connection to the perceived or known limits of a body. The issues of speed and physical limitations in live coding is discussed by Sorensen and Brown, referring to their duo aa-cell:

One problem that all live programmers must deal with is how to physically type the required code within a reasonable period of time; reasonable for both the audience but, probably, more importantly, to assist the performer to more rapidly realise ideas (Sorensen and Brown 2007)

It might be that the virtuosity in live coding is not anchored to the performing body in the same way that it might be for a guitarist and that touch typing will never be as impressive as Van Halen. Virtuosity on the part of a live coder might instead manifest itself through the use of difficult or less-common functions, complexity of code written live, or elegance of expression in coding. Within live coding, this transparency of the projected activity is interrogated: Green writes about the potential lack of transparency in code with low role-expressiveness (among other problems of notation), and work is being done to explore legibility in programming languages (Green 1989; McLean et al. 2010).

In the framing of laptop as instrument, with a focus on the legibility of that instrument, live coding could be seen as re-asserting elements of a concert hall tradition in electronic music. Describing the concert hall tradition of musicians on stage, electronic musician Francisco Lopez notes:

From my perspective, electronic music doesn't need this. Of course it can have it, it can develop its own versions of it (as indeed it does). But it's not inherent to it, it's not a natural consequence of the practices and essential manners of the operations of electronic music, but rather a symbolic acceptance of a tradition of a very different nature (in this regard, probably an opposite nature). What is more important, I believe, is that by blindly following this tradition it wastes the potential for strengthening a most important breakthrough in music of perhaps historical proportions (López 2004).

Lopez discusses "the visible intricacy of instrument playing" that rock and classical traditions share, and that we might see live coding beginning to share, too. He notes:

"While in the previous tradition of instrumental music each kind of sound corresponds to a certain gesture and to a specific physical instrument, in electronic music every possible sound is produced with the same click of a mouse, pushing of a button or turning of a knob. I don't find anything interesting in showing/contemplating these actions (if they are visible at all)." (López 2004)

Live coding has found a way around this: the interesting interaction happens not in the clicking of a mouse, but in the interactions on the screen and through code. “Show us your screens” becomes a way of either re-asserting an instrumental tradition in computer music or of increasing the transparency of the production. In the projection, electronic music performance returns to the stage. Live coding contrasts with the view of Lopez in that it finds value in the means of production, particularly its relation to the present.

It should be said that the highly visual nature of a Deadmau5 performance also differs from a Lopez performance which takes place in a completely darkened room, and there is strong link to operatic and rock spectacles in Deadmau5’s performances. Still, in some senses Deadmau5 might be construed as being closer to the radical abandonment of the stage that Lopez dreams of than the instrumentality of live coding in that a Deadmau5 performance, like a Lopez performance, discards the presentation of instrumental activity.

However, live coding ultimately avoids “blindly following” the tradition that Lopez rejects: if we are suggesting that live coding frames the laptop as an instrument in quite a traditional sense - more like the piano or the guitar than the studio tool of Deadmau5 - it should be noted that this is done in quite a radical way. It is a vital experiment that draws on the explorations of instruments in free improvisation, using the unique affordances of code and computer to create a live instrument in which ideas of score and composition are blurred, a set of extended techniques for drawing unexpected modes of expression out of a sophisticated adding machine. This serves to open up expressive possibilities in human-machine interaction, rather than closing them down through smuggling outdated musical value systems in through the back door. When the growing market for gestural controllers makes it all the more easy for a laptop performer to express virtuosity through physical gestures, drumming on pads, and navigating arrays of blinking lights, live coding is interrogating new languages of expression on stage whilst continually adapting its own instruments. Coding languages are developed and extended through new libraries, for instance, and the laptop-as-instrument develops, becoming a different machine in the hands of every programmer-performer.

## 7.1. Compositional and Instrumental Strategies in Live Coding

We have proposed that live coding quite explicitly treats the computer as an instrument, but it is a unique instrument begot by the very specific affordances of laptops and coding environments. A live coding performance will combine composition and improvisation, and the temporal nature of typing and evaluating code adds to the blurring between improvisation and composition within a live coding.

Composition in live coding ranges from the direct authoring of lists of durations and notes to meta-composition in which processes are defined which then determine the final sonic details. It blurs the distinction between instrument and score (A. Blackwell and Collins 2005, 3). Its instrument contains the score, and the score is manipulated in a performance to produce the media stream. The improvisational and compositional activity typically takes place in this range.

In the pre-performance activities of a live coder we see similarities to the studio composition of Deadmau5, and also the practicing, exploring and honing of new techniques before a concert by an improviser such as Bailey. Like Deadmau5’s studio work, the live coder might make libraries in advance of a performance that might not be altered during a performance. On the other hand, like Bailey the live coder builds chops which are stored not in the muscle but in text files. The typically modular nature of these code tools and their spontaneous usage in a performance allows them to be used improvisationally rather than in inflexible arrangements like a pre-arranged Deadmau5 set. In other words, they are used live for compositional purposes rather than perfunctory execution.

The laptop-as-instrument in live coding is a novel instrument that in many instances is being continually refined and developed by practitioners in the field. Different musical challenges are met and, on occasion, tackled, a tactic we find in free improvisation where “a large part of most improvising techniques is developed to meet particular situations encountered in performance.” (Bailey 1993, 99)

For instance, there is often a ‘conversational’ element of improvisation, as players will ‘exchange’ sounds, and to play in this manner one must be able to quickly move between sounds. For a variety of reasons, moving quickly between sounds is not always easy when performing with a computer. The difficulties of sudden changes in sound are discussed in Sorenson and Brown; Hession and McLean (Hession and McLean 2014) discuss attempts on McLean’s part to develop his coding language to go beyond fixed tempos and time signatures and explore changes in performance, enabling a more ‘conversational’ improvisation.

The musical exploration of laptop and code is similar to the unconventional interrogations of instruments through ‘extended technique’ and other strategies that are common in free improvisation. Keith Rowe started playing his guitar flat on a table, Rhodri Davies plays his harp with a fan, and Sachiko M plays a sampler without loading sounds into its memory. The hidden sonic affordances of an instrument are revealed through unconventional uses. A paper from the early days of musical live coding describes a similar strategy at play:

“With commercial tools for laptop performance like Ableton Live and Radial now readily available, those suspicious of the fixed interfaces and design decisions of such software turn to the customisable computer language. Why not explore the interfacing of the language itself as a novel performance tool, in stark contrast to pretty but conventional user interfaces?” (Collins et al. 2003)

Live coding frames the laptop as a real time musical instrument, different to the laptop as studio tool or composition device, which draws on some of the more traditional ideas of instruments and in some senses places it closer to acoustic instruments and traditions of stages and concert halls. However, this is a new instrument, a sonic exploration of a space of possibilities created by laptops and coding environments, and it is in a state of flux and development.

## 8. Labour, Composition, and Live Coding

Rather than debate the value of the performance ethics of Deadmau5, Bailey or live coders, a caveat is mentioned in order to show that given a continuum, it is possible to imagine performances which exist at different points along that continuum. Having argued this far, it now remains for a live coder to make a spectacle from some kind of perfect reproduction of programming. This could be a real-time presentation of the symbols of code without any compositional activity actually taking place, like executing pre-written code according to a script. It could be said that recent live performances of Ryoji Ikeda and Robert Henke performing as Monolake do something in this direction. Ikeda could be said to fetishise data or revel in its code-like appearance. The recent Monolake spectacle “Lumière” features the text “love” and “code” in a tightly synced live AV show (author 2014).

A live coder executing pre-written blocks of code without editing them in a pre-sequenced order could also be said to bear significant similarity to Deadmau5. It might also be argued that writing a “for” loop or even re-implementing a well-known algorithm in a performance puts the performer closer to the Deadmau5 pole if it can be said that there is little that is compositional in those two activities.

This brings into focus the issue of liveness and “labour”. It seems some may want to see a performer working or engaging in physical activity, precisely the definition of “liveness” argued for here. This may stem from pressure of ‘concert hall’ and other older performance practices as mentioned in the discussion of Lopez above. Whether there is value in the non-compositional labour involved in writing out for loops or dealing with the requirements of the syntax of a language remains to be argued in a future paper. The visible screen of the live coder at least assures the audience that the labour being undertaken is appropriate to the task in hand and the performer’s fee (if they were lucky enough to get one), and they are less likely to be checking their Facebook, filing their tax return, replying to emails or submitting conference papers.

In a sense, Deadmau5 epitomises a new model for labour-free “live” computer music that earlier theorists have anticipated, one that embraces the very unique affordances of the computer as a servile musical tool. On the other hand, the physical spectacle of Deadmau5 simultaneously places him quite firmly in rock or operatic traditions. In contrast, live coding reveals the compositional labour to the audience.

## 9. Conclusion

Through looking at the very different practices of Derek Bailey and Deadmau5, we have described a continuum. They provide opposing poles on this continuum of liveness with regards to composition and thinking about instruments.

The aspirations towards transparency of activity in live coding contrast with the opaqueness of Deadmau5’s activity in terms of its effect on the sound. Live coding values the performer’s compositional activity in the performance whereas Deadmau5 does not, placing the emphasis on the live activity of the lights and the audience as co-performers and in the non-live compositional activity in the studio prior to the concert.

Live coding uses the laptop and software as a musical instrument. In some ways, this is quite traditional compared to Deadmau5, but perhaps more importantly we find an instrumental practice that relates to free improvisation, especially through the unbounded exploration of the musical potential of materials.

## References

Auslander, Philip. 2000. “Liveness, Mediation and Intermedial Performance.” *Degrés: Revue de Synthèse À Orientation Sémiologique* (101).



- . 2006. “Music as Performance: Living in the Immaterial World.” *Theatre Survey* 47 (02): 261–269.
- . 2008. *Liveness: Performance in a Mediatized Culture*. Routledge.
- author, unknown. 2014. “Robert Henke: Lumière.” Product site. *Ableton.Com*. <https://www.ableton.com/en/blog/robert-henke-lumiere-lasers-interview/>.
- Bailey, Derek. 1993. *Improvisation: Its Nature And Practice In Music*. Da Capo Press. <http://www.worldcat.org/isbn/0306805286>.
- Blackwell, Alan, and Nick Collins. 2005. “The Programming Language as a Musical Instrument.” *Proceedings of PPIG05 (Psychology of Programming Interest Group)* 3: 284–289.
- Bown, Oliver, Renick Bell, and Adam Parkinson. 2014. “Examining the Perception of Liveness and Activity in Laptop Music: Listeners’ Inference About What the Performer Is Doing from the Audio Alone.” In *Proceedings of the International Conference on New Interfaces for Musical Expression*. London, UK.
- Chadabe, Joel. 1984. “Interactive Composing: An Overview.” *Computer Music Journal*: 22–27.
- Church, Luke, Chris Nash, and Alan F Blackwell. 2010. “Liveness in Notation Use: From Music to Programming.” *Proceedings of PPIG 2010*: 2–11.
- Collins, Nick, and Alex McLean. 2014. “Algorave: A Survey of the History, Aesthetics and Technology of Live Performance of Algorithmic Electronic Dance Music.” In *Proceedings of the International Conference on New Interfaces for Musical Expression*.
- Collins, Nick, Alex McLean, Julian Rohrerhuber, and Adrian Ward. 2003. “Live Coding in Laptop Performance.” *Organised Sound* 8 (03): 321–330. doi:10.1017/S135577180300030X. <http://dx.doi.org/10.1017/S135577180300030X>.
- Green, Thomas RG. 1989. “Cognitive Dimensions of Notations.” A. Sutcliffe and Macaulay, Editors, *People and Computers V*: 443–460.
- Gresham-Lancaster, Scot. 1998. “The Aesthetics and History of the Hub: The Effects of Changing Technology on Network Computer Music.” *Leonardo Music Journal* 8: 39. doi:10.2307/1513398. <http://www.jstor.org/discover/10.2307/1513398?uid=3738328&uid=2134&uid=4578002247&uid=2&uid=70&uid=3&uid=4578002237&uid=60&sid=21102875061697>.
- Hession, Paul, and Alex McLean. 2014. “Extending Instruments with Live Algorithms in a Percussion/Code Duo.” In *Proceedings of the 50th Anniversary Convention of the AISB: Live Algorithms*.
- Kirn, Peter. 2012. “Deadmau5, Honest About His Own Press-Play Sets, Misses Out On ‘Scene.’” <http://createdigitalmusic.com/2012/06/deadmau5-honest-about-his-own-press-play-sets-misses-out-on-scene/>.
- Lewis, George E. 1999. “Interacting with Latter-Day Musical Automata.” *Contemporary Music Review* 18 (3): 99–112.
- . 2000. “Too Many Notes: Computers, Complexity and Culture in Voyager.” *Leonardo Music Journal* 10: 33–39.
- López, Francisco. 2004. “Against the Stage.” *Unpublished Article*. Retrieved May 8: 2009.
- Magnusson, Thor. 2014. “Improvising with the Threnoscope: Integrating Code, Hardware, GUI, Network, and Graphic Scores.” In NIME.
- McLean, Alex, Dave Griffiths, Nick Collins, and Geraint Wiggins. 2010. “Visualisation of Live Code.” *Proceedings of Electronic Visualisation and the Arts 2010*.
- Miranda, Eduardo Reck, and Marcelo M Wanderley. 2006. *New Digital Musical Instruments: Control and Interaction Beyond the Keyboard*. Vol. 21. AR Editions, Inc.
- Prévost, Eddie. 2001. “The Arrival of a New Musical Aesthetic: Extracts from a Half-Buried Diary.” *Leonardo Music Journal* 11: 25–28.
- Rowe, Robert. 1999. “The Aesthetics of Interactive Music Systems.” *Contemporary Music Review* 18 (3): 83–87.
- Sansom, Matthew. 2001. “Imaging Music: Abstract Expressionism and Free Improvisation.” *Leonardo Music Journal* 11: 29–34.
- Sorensen, Andrew, and Andrew R. Brown. 2007. “Aa-Cell in Practice: An Approach to Musical Live Coding.” In *Proceedings of the International Computer Music Conference*.
- Watson, Ben. 2004. *Derek Bailey and the Story of Free Improvisation*. Verso.
- Zimmerman, Joel. 2013. “We All Hit Play.” <http://deadmau5.tumblr.com/post/25690507284/we-all-hit-play>.