Revealing Timelines: Live Coding and its Gestures

Joanne Armitage
ICSRiM, School of Music,
University of Leeds
j.l.armitage@leeds.ac.uk

ABSTRACT

As a highly mediatized, laptop-based improvisation practice, the human gesture of live coding is often just small motions. The physical connection of the performer and their interface is limited to a small surface area of skin making singular temporal connections, one by one, with a computer keyboard. Live coders have often been somewhat teased for the lack of ‘expressivity’ in their physical performance. As a response, practitioners have combined a range of visual stimuli with their code as a visual distraction from their limited gestures.

In this paper I consider the key presses of a live coding in the context of expressive gesture, and present a technological approach to amplifying, or highlighting them in live performance as auditory and non-auditory stimuli. I will then describe the application of the system in two performances (three for the final paper), and conclude with some thoughts as to how this idea could progress and expand.

1. INTRODUCTION

If we consider live coding within traditional musical performance paradigms interesting critical questions arise as to the nature of live coding as a musical performance practice. As a performer/improviser, moving from instrumental practice to a text-based laptop performance interface, the most striking difference must certainly be the addition of the typing timeline and how that gesture connects (or not) with the sound being produced.

In music, gesture is often considered to hold the expressive qualities of a performance. With stochastic processes so prevalent, even in acoustic music composition practice, the need to identify an expressive instrument to ‘deliver’ algorithmically generated sounds seems counterintuitive. Despite the idea of musical gesture feeling particularly redundant in the context of a machine performance, an audience desire for some form of explicit, exposed and intelligible process brings gesture into question.

As an instrument, the laptop is so often considered an obstacle - abstracting the laptop performer and lessening audience-perceived ‘tangibility’ of a sound. Whilst orchestral performers occlude each other’s bodies visually and sound auditorily, and the delay on the guitarist can confound the temporal relationship between finger movements and sound, the live coder shows their screen. We run risk of exposing ‘too much’ of our desktop, or a private email, or an intimate photograph with just an ill-considered search or accidental click. I’m sure that most live code performers are sick of the “checking your email” comments.

The discussion of the placement and location of the instrument in live coding practice is a debated topic. As a performer who was exploring live coding of hardware, from my personal perspective, the location was clearer. Although, I expect users of samples would not necessarily consider the sample bank they address as their ‘instrument’, and would perhaps consider there to be no (clear) instrument in their performance. What these considerations bring to light are the tensions and nuances that lie within a dynamic field of interdisciplinary practitioners. These ideas that challenge notions of traditional instrumental performance practice present an exciting discourse in the field.

The relationship between sound, gesture and expression is an expansive area of research where the connections between a performer, their instrument and the sound it produces are analysed. Alongside that we have another body of research that looks at these ideas through a digital lens at electronic music performers and their interactions with their machines. This paper looks to these fields, but
discusses work that celebrates the minute and subtle actions of the live coder as a valid form of expression in performance, and considers ways in which we can reveal this to the listener.

In short, this paper will first briefly consider gestures in performance and then look towards an analysis of that in the context of live coding, informed by discussions around digital performance practice. It will then present an improvisation system, *key*, that I have developed to render the keystrokes of a live coding performance as vibrations - amplifying a coder's gestures, whilst revealing the temporal of flow of a their performance.

2. CONTEXT

The relationship between instrumentalist gesture and the sound is produces has been a long considered topic of research in musicology. Gesture has been traditionally described in these terms as a visceral force inherent in sound as a result of characteristics of a motion. As Dobrian (2012) suggests, current technologies such as sensors and cameras facilitate a more quantifiable measure of a gesture. Setting poeticism aside, it is incredibly easy to measure aspects of live coding gesture - our keystrokes result in a flow of values that occur across time. Aspects of the hand’s movements across the keyboard could be captured with combinations of flex sensors, accelerometers or cameras; however, they have little direct effect on the sound.

 Whilst in acoustic instrument performance practice the body connects with the instrument in multiple dimensions that have an immediate effect on sound, the key presses of live coding generally occur prior to a change in sound. The temporal detachment/disconnect within live coding movements, when viewed as a performance gesture, is something I personally find fascinating. Not only the notion that the performer reveals their plans prior to their inception, but the disconnect inherent in our temporal flow being mediated by the machine. We can try to release ourselves from the visual absorption of the screen, looking up to survey audience response, but a critical aspect of what makes a live coding a performance is each finger manoeuvring the keyboard, individually pressing keys.

The keystroke has previously been considered in a number of works, most famously *The Typewriter* (1950) by Leroy Anderson: A novel work that uses the mechanical sound of the typewriter as a rhythmic device. Notable in this context is a computer rendition of the work by Alexander (2004) that brings it to the (then) contemporary office space, capturing keystrokes with a microphone.

In terms of live coding practice keystrokes have been previously explored in a number of works. Baalman (2015) discusses the work, ‘Live Code’ which brings the auditory elements of typing into the performance via the internal microphone of the laptop that is being coded on. The captured audio is used as the main auditory element of the performance whilst the code, characters of the keystrokes themselves and additional sensors manipulate and affect it. Another example of a performance bringing keystrokes to the fore is *SK_Computer* (Knotts, 2014) where they are visualised during the performance.

In a recent lecture, Herndon (2016) suggests the laptop to be our most intimate of instruments. If the laptop embodies our digital lives (work and personal) then these timelines are still active whilst we are engaging with a machine as part of our performance. At a time when musicians are increasingly supported by revenues from live performance, we have to question our tools as means of delivering ‘authenticity’. The excitement of liveness and the instability of our machines, delivers an uncertainty and potential for failure. In live coding, much of what is considered to be an authentic performance is informed by the Toplap manifesto. This manifesto suggests an open process whereby screens are shared and code grows from a blank screen. Could this be considered to reflect a ‘true’ unfolding of a performance timeline? Syntax occludes non-coding and coding audience members alike, much like flailing arms and bows. Further, the ephemerality of code is lost in the duration of performance, where unsounding ‘switched off’ text lies redundant visually yet active.

3. KEY, CONCEPT AND DESIGN

‘Key’ is my first response to the idea of coding as a performance practice that could explore physical embodiment via haptic technologies. The initial concept behind this work was to investigate ways in
which we could explicitly expose the underlying timelines, and relatively 1 dimensional gestures of live coding to listeners. The work explores a simple idea whereby the key presses produced by a live coder are ‘haptified’ or translated into vibrations felt by the listener. By doing this, I hoped to:

- Highlight awareness of typing gestures
- Increase the listener’s ‘connection’ with the performance
- Augment audience experience with additional sensory stimuli

Technologically, ‘key’ comprises of hardware vibrating motors that are controlled via an Arduino microcontroller, further details of this setup can be found in Armitage and Ng (2015). Keystrokes are captured in SuperCollider, mapped to vibration parameters, and sent via the Serial port to the Arduino for rendering. When a key on the keyboard is pressed it triggers all the motors to vibrate, the duration of that vibration and the intensity (pulse width modulation) is mapped depending on character pres-ets.

4. PERFORMANCES USING KEY

4.1 di.stanze, Leeds, 14th November 2014

Around the time I conceived of ‘key’, I had began to send SysEx messages to a Juno 106 synthesizer. These are strings of hexadecimal numbers that can control most parameters on the instrument. I was fascinated at how I could address the instrument using code in this way, and appreciative of the often criticised MIDI implementation in the instrument. I did find a slight problem with this new way of interfacing with my instrument. The SysEx messages on the Juno 106 are strings of 7 hexadecimal values. In terms of what I then believed to be an ‘authentic’ blank screen live coding performance, I felt I could must type them out at least once.

Using the motors allowed me to represent my performance to listener in a more immediate way; “honestly, I am working towards producing sound!” Not only that, but it made the typing aspect performative, allowing me to connect with the audience. I setup three motors on the seats in front of me, with instructions – “Hold me, I am safe.” Although the concert went terribly, as my first time live coding from scratch, solo. Post-event commentaries online included the following; “The haptic element of Joanne Armitage’s performance (a buzzer held during the performance) ultimately drew attention to changes that she made in the music by typing (which didn’t always signal changes in sound).”

This comment reflects how the work essentially placed what were errors in typing at the foreground of my performance. Feeling that things didn’t go to plan, and a little self-conscious as a new live coding practitioner, I began to think the concept was unsuccessful. Although, I have recently altered the system and rejuvenated it as the ideas of gesture and alternate temporal flow became more pertinent in my practice. Further to this, I refined the idea and presented two other performances of the work.

4.2 Deep Hedonia, Liverpool, 18th March 2016.

With some slight changes of the mapping to ‘amplify’ when I executed code in SuperCollider, and an addition of 13 vibration motors, I employed the system again in performance (see Figure 1). There is video documentation of this performance available online (https://vimeo.com/162125985) as well as the full audio recording (link in the video description).

As an alteration, I mapped the vibration intensity of the motors, which roughly translates to vibration amplitude, as the time different between keystrokes. In this attempt to reveal my flow better to the listener, vibrations were strong when I typed characters in quick succession, becoming weaker as I slowed down. The weaker vibration for slow typing was set to reflect the uncertainty, in performance. It

---

1 A recording of the first performance can be heard here https://soundcloud.com/joannnne/key_v001_liveatdistanze_ccch-midisysex

2 Lauren Redhead’s blog discussion of the di.stanze event is available here http://weblog.laurenredhead.eu/post/101608521587/distanze-festival-modes-of-listening
ended up reflecting my debugging process quite interestingly - as I found the mistake(s), corrected and worked my way back towards performing 'confidently'.

Performing an ambient set, an audience member described the sensation of closing their eyes, whilst knowing that the performance was still active. I think this brings an interesting connection between the temporal dissonance of live coding and the authenticity and liveness of its performance, discussed in (Parkinson and Bell, 2015).

5. TO CONCLUDE

This paper has discussed the work key in the context of gesture in live coding performance practice. It has presented a quick overview of the technologies involved and discussed its application in two (by final submission 3) performances. As I reflect now on this work, I reconsider my initial thoughts. If we consider our code as a form of 'live score' (Magnusson, 2011), our typing is not an additional timeline; it is a relocation of the compositional design process to the performance. Thinking from this perspective, it becomes less clear as to where our expressive performance gesture exists, or whether it does at all.

With live coding still a somewhat emergent practice, there is space for greater discussion around the performative effect of typing as a timeline in a performance, and how the editing code relates to the sound. This idea raises critical questions as to how coders produce sound and then how their typing responds to it. In the future I hope to introduce the use of this system to other live coders and consider how we can use an amplification of their performance gestures to better represent the flow of their performance. For example, sensors could be employed to capture the gesture ‘detail’, performer engagement with the screen, and we could also capture movements of the mouse/track pad.

Live coding practice can consider projection of screens as a way to counteract and disambiguate the abstraction of digital technologies. Despite this transparency, there is still an interesting phenomenon whereby the majority of the physicality of the performance gesture is somewhat removed from the
sound being produced. This temporal disconnect subverts traditional performance ideas of ‘playing in time’ and relinquishes the processing of temporal control to the computer.

Although many of us spend considerably more time typing than we have ever spent focussing on an instrument, our typing itself would not be considered virtuosic. When we consider how errors of inaccurate typing can affect flow, we open up the potential for a challenging discussion as to how typing affects sound. When considering the vulnerability of the live coder, accuracy becomes pertinent. Could audience vibrations be employed as a signifier or flow? The perceived ‘value’ of that to the listener is something that warrants further investigation.

Acknowledgments

A special thanks to Alex McLean and Kia Ng.

REFERENCES


