

Reflections on live coding collaboration

Alex McLean

Live coding collaborations

Through my practice as a live coder of music, I have enjoyed varied collaborations with percussionists, live artists, performance artists, dancers and choreographers, as well as other live coders. In the following short paper I will reflect on a number of these collaborations, including with/within algarave, live Jazz improv, performance art and choreography practice. Some focus will be on the role of time and language, as core themes embedded in “live coding”, but I will also consider wider cultural issues, and of the role of collaboration in making live coding meaningful. I conclude by considering how technology could better support close collaboration in the future.

Throughout, informal reports by collaborators and audience members are drawn from, as well as reflections as a live coding performer. This is not intended as a research output, or practice-led research, but as reflections on practice which could inform later research hypotheses.

Making collaboration visible: Inter- and intra-technology

When performing with technology on stage, there can be a lingering feeling that some aspect of the performance is invisible. Slub have projected screens since inception in the year 2000 (Collins et al. 2003), a habit which has been taken up by the live coding community at large (Ward et al. 2004). Slub consists of Adrian Ward, Dave Griffiths and myself in various combinations, but our collaboration does not take place in our technology, but through the musical and sonic structures we produce. We do make a network on stage, but this is only to create a shared clock so that we may coordinate tempo changes, and share the same down beat. Our systems are otherwise decoupled, our collaboration being between our different systems rather than through the same system. This is not clear to all audience members however, who through informal post-performance discussion have occasionally revealed an assumption that we are working on different parts of a technological machine, rather than working on our own machines and collaborating as musicians within a laptop ensemble.

Audience assumptions are perhaps less important than they might appear, particularly where music is considered to be a space for audience members to explore, rather than a medium for performer-audience communication. However, audience perception of balance between performers has always been a key issue when collaborating with instrumentalists and dancers. The question is, how can such collaborations be staged to show them as balanced exchanges between two or more creative individuals? Reflecting the general role of computation in culture, one assumption might be that the laptop operator is somehow *controlling* the other performer, or at least processing their sound or movements in some way. Another assumption might be that the laptopist is carrying out mundane operations, while an instrumentalist or dancer is contributing the real creativity to the performance through ‘authentic’ gesture.

Of course in many technology oriented performances, such assumptions as described above are actually true, and great imbalance between laptopist and a more ‘physical’ collaborator is not always seen as an important consideration. However, the collaborations I have taken part in have always looked for balance. Kate Sicchio and I are developing a live code and live choreography performance as a confluence of our practices, setting up a feedback loop between choreography, the body, code, sound and back into choreography (McLean and Sicchio 2014). We are ambivalent about the success of this piece, our experience as performers connecting our two notations has been very good, but the physical strain put on Kate on her side of the loop led one audience member to report feeling that

I (as programmer) was torturing Kate (as dancer). In the piece, Kate's movements interfere with my code, but any torture felt by me is solely cognitive, and so less visible. Kate is herself a technologist as well as choreographer, and has been instrumental in the recent conceptual development of live coding (Sicchio 2014), but it can be difficult to get the nature of our collaboration, as an exchange between reflective technologists, across. Where we have agreed our performance has really worked, is where we have explained and discussed it first.

As an aside, this work carries a key problem when experimenting with collaborative performance; such performances are set up to fail; ideas collide and we learn from the pieces. All we can really do is embrace the risk, and hope that audience members perceive some of the possibility that we are reaching for, and often miss.

Back to the question of audience perception; how can collaboration through body and code be made more visible? One collaboration with live coder and drummer Matthew Yee-King as Canute¹ looks for way of sharing data between an instrumentalist and live coder. Matthew produces probability distributions of hits on his drum kit, visualising them and sending them to me as Tidal patterns (McLean 2014), which I then transform through live coding with further visualisation. Three performances in, audience response has so far been positive, although perhaps more responding to the musical end result (in particular Matthew's frenzied drumming), and less on the conceptual basis of the piece at this stage.

A more directly interventionist approach has been found in collaboration with performance artist Susanne Palzer. Susanne curates a series of "Open Platform" happenings based on the idea of "Technology without Technology", exploring notions of digital art outside of the normal frame. She has developed a series of performance pieces where she steps on and off a (wooden) platform, sometimes with lights also switching on and off, exploring the digital in performance. We have collaborated on two performances so far, in the first "Binary Transmission" (Access Space, 6th December 2013), I knitted while Susanne stepped on and off and around her wooden platform, a knit for every on, and a pearl for every off. In this way her discrete, binary movements were transduced into the binary pattern of fabric. In our second collaboration "On-Gaku" (Bloc Sheffield, 12th July 2014), I used a laptop rather than knitting needles, and did my usual live coding with Tidal. However, we hooked up a pressure sensor to Susanne's platform, so that my screen was only projected while Susanne was 'on'. I worked using a wireless keyboard, and using the projection as my screen, so had to cope with only seeing the code I was editing for fleeting moments. In joining our individual practices in this way, our difficulty was more visible on both sides. In my case, I could not see my screen for most of the time, and in Susanne's, her physical exertion was clear.

It is perhaps telling that my collaborations often end up looking for ways of balancing difficulty and friction in interwoven performance practice by deliberately introducing new difficulties and struggles. This works well within a performance art context. It is worth noting however that my musical collaborations with instrumentalists, including with drummer Paul Hession (Hession and McLean 2014), are far less troubled in terms of the nature of collaboration. When the collaboration is on the shared basis of sound, technology has less of a bridging role, and therefore has less of an overbearing influence on audience reception of a piece. However, none of this is to say that our technology should in any way become invisible or seamless.

Percussion - generation at the speed of gesture, and freedom from the grid

A primary motivation for the development of Tidal over the years has been collaboration with percussionists. This began with a number of sessions and performances with Alex Garacotche in 2004, including the Ultrasound festival in Huddersfield. At the time I was using my feedback.pl editor for live coding with the Perl programming language, which included an interesting user interface application for self-modifying code. However, it was unwieldy, and when live coding "from scratch", I might be a minute into a performance before I started making sound.

By switching to the Haskell programming language I have been able to develop Tidal as an embedded Domain Specific Language (eDSL), for composing pattern as higher order structures with highly economical syntax (McLean 2014). This allows me to respond to changes introduced by co-performers within seconds. As well as speed of reaction, it has also been important to develop an expressive approach to time. While 16 step dance music is a passion of mine, Tidal allows me to quickly express complex metric subdivisions, and layering time signatures on

¹See <http://canute.lurk.org/> for information about Canute.

top of each other to create polyrhythm. Tidal represents time using rational numbers, and patterns as functions rather than sequences.

Tidal is certainly not without its constraints, but the freedom which this representation of time offers me has allowed me to collaborate within free improvisation. My primary exploration in this area has been with drummer Paul Hession, who has honed his practice over many years, including through collaborations and more recently solo play. Paul has now extended his drum kit with a range of analogue, digital and physical techniques (Hession and McLean 2014), and interestingly has explored collaborations with unsupervised ‘live algorithms’ alongside his occasional work with me as live coder (Hession and McLean 2014).

My conclusion here is that while code necessarily distances the live coding musician from the physical production of sound, live coding technology, including my own, has succeeded in reducing latency between action and reaction. This in turn has allowed myself as a live coder to collaborate closely with live instrumentalists, including in free jazz situations. In this sense, live coding has genuinely brought programmers closer to the people around them.

Community growth and genre

Collaboration in music extends beyond co-performers, but also with audience members and in the broad Musicking sense (Small 1998) where every activity around music culture is seen to be part of music-making. There is an argument that music has become formulaic and backward-looking over the past decade, lacking revolutions comparable to rock ‘n’ roll and rave in the late 20th century (Fisher 2014). It is too early to say whether live coding will have real cultural resonance as an agent for change in mass media, but perhaps there is some potential shown in the media reaction to Algorave music (e.g. Cheshire 2013).

Algorithmic music has been present in dance music culture for some time, but Algorave has provided a new common ground for us to explore together (Collins and McLean 2014). Algorave is a collaboration without clearly defined edges, a space initially created by live coders such as Nick Collins, Dan Stowell, Matthew Yee-King and myself, and (I think crucially, in terms of establishing identity) graphic designer David Palmer. Creating this space has in some respects janitorial, helping shape that identity in the background, while leaving space for organisers, performers and (perhaps most importantly) revellers to define what Algorave really means. What started as a joke of sorts has become unexpectedly successful - many people in many countries (e.g. UK, Mexico, Australia, Germany, Peru, Belgium, Canada) have felt able to make Algoraves for their own, without asking anyone for permission. Some have been organised by practitioners and professional promoters, and quite a few by academic conferences, a collaboration which spans research and practice.

Closer

Live coding is by now proven as a reasonable means to make music, both within small engaged live coding communities, and within larger enthusiastic, dancing audiences in the hundreds. Perhaps the next leap is to see how live coding can bring us closer together, and unearth modes of interaction that could take us further away from software engineering, towards closer shared experience of code. From the perspective of music technology, the most recent leaps in shared programming environments are a decade old; the *Republic* live coding environment for SuperCollider (Rohrhuber et al. 2007), and the *Reactable* tabletop instrument (Jordà et al. 2005). The former explores conversational, shared live coding style, and the latter simultaneous editing of a sonic dataflow network by collaborators around a circular table.

My feeling is that a further leap is overdue, and the results could take live coding further away from the well established applications for programming languages, into radically different ones. In particular, environments aimed at creative, shared exploration through abstraction, aimed at shared experience rather than end results.

Bibliography

- Cheshire, Tom. 2013. "Hacking Meets Clubbing with the 'Algorave'." *Wired Magazine (UK)* (September): 85+.
- 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 Rohrhuber, and Adrian Ward. 2003. "Live Coding in Laptop Performance." *Organised Sound* 8 (03): 321–330.
- Fisher, Mark. 2014. *Ghosts of My Life: Writings on Depression, Hauntology and Lost Futures*. Paperback; Zero Books.
- 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*.
- Jordà, Sergi, Martin Kaltenbrunner, Günter Geiger, and Ross Bencina. 2005. "The reacTable." In *In Proceedings of the International Computer Music Conference (ICMC 2005)*, 579–582.
- McLean, Alex. 2014. "Making Programming Languages to Dance to: Live Coding with Tidal." In *Proceedings of the 2nd ACM SIGPLAN International Workshop on Functional Art, Music, Modelling and Design*.
- McLean, Alex, and Kate Sicchio. 2014. "Sound Choreography <> Body Code." In *Proceedings of the 2nd Conference on Computation, Communication, Aesthetics and X (XCoAx)*, 355–362.
- Rohrhuber, Julian, Alberto de Campo, Renate Wieser, Jan-Kees van Kampen, Echo Ho, and Hannes Hölzl. 2007. "Purloined Letters and Distributed Persons." In *Music in the Global Village Conference 2007*.
- Sicchio, Kate. 2014. "Hacking Choreography: Dance and Live Coding." *Computer Music Journal* 38 (1) (March 1): 31–39.
- Small, Christopher. 1998. *Musicking: the Meanings of Performing and Listening (Music Culture)*. First. Paperback; Wesleyan.
- Ward, Adrian, Julian Rohrhuber, Fredrik Olofsson, Alex McLean, Dave Griffiths, Nick Collins, and Amy Alexander. 2004. "Live Algorithm Programming and a Temporary Organisation for Its Promotion." In *Read_Me — Software Art and Cultures*, edited by Olga Goriunova and Alexei Shulgin.