

Algorave: Dancing to algorithms

Cambridge Companion to Music and Digital Culture, 1000 words

Alex McLean, Research Institute for the History of Science and Technology, Deutsches Museum, Munich

Abstract:

A brief reflection on the founding and development of *Algorave*, events where musicians and visualists create music using algorithms, usually through *live coding*: live manipulation of algorithms as code. The now traditional role of projected code at such event is reflected upon, as well as the experience of performing at such an event, and the role of code as creative material in embodied music improvisation.

Keywords:

Live coding, algorave, textility of code, making, algorithmic music, performing arts.

50 word bio:

I am an artist-programmer and interdisciplinary researcher based in Sheffield UK. I co-founded the Algorave and TOPLAP live coding movements, several international conferences/festivals, and the TidalCycles live coding environment. I work as post-doctoral researcher on the PENELOPE project at Deutsches Museum, Munich, investigating the structures of ancient weaves.

Algorave is a movement that I co-founded with Nick Collins, Matthew Yee-King and Dan Stowell, focussed on the conspicuous involvement of algorithms in the generation of electronic dance music, which has quickly developed since the first event in 2012. At first, Algorave often seemed imaginary. With events promoted as algoraves often poorly attended, sometimes in inappropriate settings such as brightly lit rooms with rows of seating (Collins and McLean, 2014). The 'rave' in algorave suggests mass dancing as one, but in the beginning, this was rare. This may well be indicative of the academic roots of computer music being poorly spliced with the history of electronic dance music (Parkinson and McLean, 2014).

More recently, algorave has taken hold as a distributed network of thriving scenes, with events organised by experienced promoters finding large audiences in club and festival venues, or adopted by local musicians putting on parties in small rooms with big sound systems. However, we can say that all of the over 150 Algorave events so far have been experimental, pushing at the boundary between improvisatory and danceable. By embracing the value of experiment we have to accept that they will not always 'work'. While some artists have toured around them, these events have each developed their own local flavour, having taken place in dozens of cities across Europe, Australia, Japan and both North and South America. Unlike creative franchises such as MakerFaire and TED, the algorave brand is purposefully unprotected: anyone is free to host one and there are no particular constraints. But what ties them together?

There are a range of approaches at play, but the majority of performances at algoraves are live coded, meaning that the language of computer code is used as a medium for creating music by describing it. This code is made visible for audiences through projection, often throughout the space, potentially creating a sense of being inside the code. The programmer creates and/or modifies code while that code generates music, creating a continuous creative feedback loop through code and sound that is an amalgam of composition via notation and music improvisation.

The notion of dancing to algorithmic music is evocative of sci-fi but has a history in the here-and-now. Accomplished musicians have employed algorithms in their work for many years, in the case of electronic music duo Autechre for over two decades, pushing the boundaries of dance music to widespread critical acclaim. There is of course a far longer history of composers formalising their creative approach. Indeed, rather than signaling technological progress, I would argue that algorave instead signals an unravelling of technology, stripping back years of interface development to re-expose computers as language machines. Words are a very human mode of articulation, and the words of source code compose together to define the computational procedures of everyday life. So, in the spirit of Christopher Small's conception of music as representing wider cultural relationships (Small, 1998), the visible presence of code in algoraves not only allows us to reflect upon the role of code in our lives, but also to reimagine that role. We can imagine coding as a true craft, shared and culturally legitimate, by focussing on the role of coding as just one step in a live and very human process of becoming.

Virtuosity and code comprehension are often discussed in live coding literature, which situates the programmer as a virtuoso and audience members as passive listeners who comprehend musical processes by reading code while listening. In my opinion, neither of these presumed roles work well at an algorave. Firstly there is the name (can you really take yourself seriously as an 'algorave virtuoso')? More than that, algorave's combination of experimental freedom with accessibility seems closer to punk than contemporary Western classical music, with programming languages like Ixi lang and TidalCycles perhaps being as easy to learn as three guitar chords. In both of these systems, the ability to create techno music is only a few keypresses away, and genre-twisting transformations just a few more. While live coding dominates algoraves (although other approaches to algorithmic music are welcome), the traditional projections of code mean it is hardly possible to read them while dancing. Simply witnessing the broad outlines of coding activity, and the derivatives of code complexity growing and waning with that of the music, is more important to most algorave participants than close reading or understanding, although just as some like to crowd behind a DJ to watch their technique, so participants are free to stand and read into the technique of the live coder.

Perhaps more controversially, I think the live coder's code comprehension is also in doubt. In Tidal, which is embedded in the strictly typed language Haskell, just about everything is a pattern, or a function involving one. It is therefore straightforward to introduce pattern transformations at points within a piece of Tidal code, without understanding the whole. My introspective hunch is that this property of the programming language allows me to make music with TidalCycles without really knowing what my code is 'doing'. In fact, because TidalCycles is highly declarative, in notating what is to be done rather than how, it isn't really doing anything but rather describing an outcome across several layers of abstraction. The meaning therefore is not understood in terms of code, but in terms of the musical results. In other words, live coding becomes much more about listening, as well as deciding when to make a change, than it is about understanding the code for itself. I feel like I am guided around my code based on what happened to the music last time I made a change. This is what I like to refer to as the 'textility of code' after Tim Ingold's textility of making (2010), which is closely related to the idea of Bricolage programming explored by Turkle and Papert (1992). Rather than seeing a programming language as a means to efficiently express a thought, I think it is much closer to the truth to think of it as an environment in which to think through code as material.

Down to earth, the experience of live coding at an algorave feels physical rather than disembodied: as the live coder, you are still working with code as abstract material, but your focus is on both the physical experience of listening and the moments at which each code edit is evaluated, in time with the movements of people dancing in front of you. Though in an apparent state of flow (Csikszentmihalyi 2008; Nash and Blackwell 2011), you become hyper-aware of the passing of time, as you work with or against expectations held by club audiences of regular change, at multiples of four by continually counting, the pace of edits intertwined with the pace of musical change. Who knows where this strange experience will lead digital music culture?

Acknowledgments

This work was completed during the PENELOPE project, with funding from the European Research Council (ERC) under the Horizon 2020 research and innovation programme of the European Union, grant agreement No 682711.

Collins, N. and A. 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*.

Parkinson, A. and A. McLean (2014). Interfacing with the night. In *Proceedings of the 2nd International Conference on Live Interfaces*.

Small, C. (1998, June). *Musicking: The Meanings of Performing and Listening (Music Culture)* (First ed.). Wesleyan.

Ingold, T. (2010, January). The textility of making. *Cambridge Journal of Economics* 34 (1), 91-102.

Turkle, S. and S. Papert (1992, March). Epistemological pluralism and the revaluation of the concrete. *Journal of Mathematical Behavior* 11 (1), 3-33.

Csikszentmihalyi, M. (2008). *Flow: the psychology of optimal experience*. HarperCollins.

Nash, C. and A. F. Blackwell (2011). Tracking virtuosity and flow in computer music. In *Proceedings of International Computer Music Conference 2011*.