

## Personal Take 9 Algorave: Dancing to Algorithms

ALEX MCLEAN

*Algorave* is a movement I co-founded with Nick Collins, Matthew Yee-King and Dan Stowell, focused on the conspicuous involvement of algorithms in the generation of electronic dance music, which has developed quickly since its inception in 2012 (Collins and McLean 2014). At first, algorave often seemed imaginary, with some ‘algorave’ events poorly attended or in inappropriate settings such as brightly lit rooms with rows of seating. The ‘rave’ in algorave suggests mass dancing as one, but this was rare in the beginning. 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. All of the 150-plus algorave events so far have been experimental, pushing at the boundary between improvisatory and danceable. By embracing the experiment we have to accept that the events 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 few constraints. But what ties them together?

There is 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. This code is made visible for audiences through projection throughout the space, potentially creating a sense of being *inside* the code. The programmer creates and/or modifies code while it 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, as in the case of electronic music duo Autechre who push 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 signalling 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 (1998) conception of music as representing wider cultural relationships, 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 focusing 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, yet, neither of these presumed roles work well at an algorave. First there is the name (can you really take yourself seriously as an 'algorave virtuoso')? Beyond that, algorave's combination of experimental freedom with accessibility seems closer to punk than 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 keystrokes away, and genre-twisting transformations just a few more. While live coding dominates algoraves, 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 read into the technique of the live coder.

Perhaps more controversially, I think the live coder's code comprehension is also in doubt. In *TidalCycles*, 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. Meaning is not understood in terms of code, but in terms of

177 *Personal Take 9*

musical results. Live coding becomes more about listening, and deciding *when* to make a change, than it is about understanding the code 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 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 more accurate to think of it as an environment in which to think through code as material.

The experience of live coding at an algorave feels physical rather than disembodied: as the live coder, you are 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 and the pace of edits intertwined with the pace of musical change. Who knows where this strange experience will lead digital music culture?