Routledge
Taylor & Francis Group

International Journal of Performance Arts and Digital Media

ISSN: 1479-4713 (Print) 2040-0934 (Online) Journal homepage: http://www.tandfonline.com/loi/rpdm20

Writing with shaky hands

Thor Magnusson & Kate Sicchio

To cite this article: Thor Magnusson & Kate Sicchio (2016) Writing with shaky hands, International Journal of Performance Arts and Digital Media, 12:2, 99-101, DOI: 10.1080/14794713.2016.1234689

To link to this article: http://dx.doi.org/10.1080/14794713.2016.1234689

INTERNATIONAL JOURNAL OF PERFORMANCE ARTS AND DIGITAL MEDIA, 2016
VOL. 12, NO. 2, 99–101
http://dx.doi.org/10.1080/14794713.2016.1234689

Routledge
Taylor & Francis Group

EDITORIAL

Writing with shaky hands

> I am no doubt not the only one who writes in order to have no face. Do not ask who I am and do not ask me to remain the same: leave it to our bureaucrats and our police to see that our papers are in order. At least spare us their morality when we write.
>
> (Foucault, 1972, 17)

In his introduction to the *Archaeology of Knowledge*, Foucault objects to be defined as a writer of a solid identity in order to be able to move, change, and explore the unknown – labyrinths and underground passages – admitting that such writing is at times done with a 'shaky hand'. This journal issue explores live coding, a performance practice that operates in a similar spirit of writing as an adventure and exploration, of deliberately rejecting definitions, of being heterogeneous in nature, and continually challenging its self-understanding through the practice of writing and rewriting – of defining and redefining – as a public performance.

Although live coding attempts to escape definition, there are denoting statements that have traditionally described its activities, such as: writing software in real-time; changing a programme whilst it's running; projecting the screen for the audience to participate in; writing as an improvisatory practice; composing live using textual notation; changing rules whilst following them; conversing with the computer in its own native language; thinking in public; creating and using bespoke systems tailored for on-the-fly performance.

Live coding is an improvisatory performance method that has been applied in the diverse art forms, but historically it has been very prominent in music, perhaps due to the importance of the musical score, computer music, and the technological foundations of musical practice in general (as demonstrated by the history of musical instruments). Live coding can be used as a compositional technique in the studio without an audience, but in this issue we focus on live coding as a performance method where code is typically projected on the wall for the audience to engage with – observing the thought process and the composition as it happens, as well as partaking in interpretation (understanding how sounds, visuals, or movements derive from the notation).

Live coders programme, they write in public (Greek: *pro-graphein*) – but they also *pre-gramme*, that is, their algorithmic writing is conditioned by a system that has already been designed with careful considerations of expressivity, constraints, interface, and other concerns of human-machine interaction and performer-audience communication. Most live coders adopt or develop a system best matched to the ways they wish to communicate the writing of their work – manifest as music, visuals, or other art forms. It is a writing of a writing that will happen at a later stage, on a different stage! The initial multimodal system thus offers creative solutions to the specific problematics of the stage, the real-time actuality of the performance, and the openness of improvisatory practices. Whilst the live coding system is pregrammed, the subsequent performance act of programming processes digital media as fluid entities whose functionalities are subject to development and change.

The aim of this live coding special issue for the *International Journal of Performance Arts and Digital Media* is to present live coding in its broader historical and aesthetic context; the issue offers a set of theoretical papers supplemented with focussed artistic statements exemplifying

diverse live coding practices. Emma Cocker's article *Performing Thinking in Action: The Meletē of Live Coding* applies the Greek term *meletē* to describe live coding as a meditative thought experiment; one that is publicly shared, a form of 'thinking-in-action'. Live coding is here framed as a site for experimentation and negotiation between the spontaneous and the planned, the human and the machine. In *Setting Live Coding Performance in Wider Historical Contexts*, Sally Jane Norman traces live coding practices of projecting liveness and poetic modelling back to historical examples involving the Sun King, Louis XIV, amongst others. A key theme in the article explores how live events are crafted as uniquely artistic material, overriding habitual frames of reference. This is a theme picked up by Tim Sayer, whose article *Cognitive Load and Live Coding: A Comparison with Improvisation using Traditional Instruments*, explores the cognitive mechanisms applied in improvisation, specifically comparing the improvisational practices of live coding to those of instrumental music.

Several articles address audience perception in live coding, defining it as a practice where most commonly artists choose to 'show their screens'. In *Understanding Live Coding Events*, Burland and McLean provide a study that aims to 'explore the motivations, experiences, and responses of live coding audiences and to examine their perceptions of the role and impact of the projected source code during live coding events'. This is an area that has not been investigated previously in any depth within the live coding literature, but live coders' gesture of projecting code sets up a relationship between performer and audience that is rather unique in the performing arts.

The connections of live coding with live art are also found within this special issue, with Hester Reeve reflecting on her work collaborating with live coders, and placing it into philosophical discourse. Reeve's *Live Code, Live Art and the BwO Dissection* draws out similarities in the practices of live coding and the practices of live art, and in *<en>Coding Performance: From Analogue to Digital*, Hannah Allan looks to Fluxus performance work in order to find text-based pieces which echo work found in the live coding community.

In conceiving this issue, we decided to call for artist statements in order to establish a more concrete grounding to some of the philosophical arguments presented in the longer articles. These statements focus on issues arising directly from performance, including designing performance systems, playful interpretations of code, and pedagogical implications. The creation of a live coding environment that serves equally for virtuoso performance and as an educational tool is the subject of Sam Aaron's article. In *Sonic Pi – Performance in Education, Technology and Art*, Aaron discusses the development of Sonic Pi and argues that teaching is by nature a performance – any good educational tool should reflect that fact. Andrew Brown's artist statement, *Performing with the Other: The Relationship of Musician and Machine in Live Coding*, delves deeper into the performer-machine relationship with a phenomenological discussion of his own performance works. For Brown, the live coding system is not a mere instrument, but the live coder's *other* – an entity that can respond to the live coder and exert its own character during performance.

Other artist statements explore audience engagement through a variety of live coding practices. The Mexican live coding community's contribution to this issue is a report authored by Mauro Herrera Machuca, Jaime Alonso Lobato Cardoso, José Alberto Torres Cerro and Fernando Javier Lomelí Bravo, reflecting on their performance system that merges natural language and code. For Herrera et al. have created framework that explores the understanding of code as literary commands aimed at audiences who may not have a computer programming background. Chris Kiefer discusses his web-based experiment *ApProgXimate Audio* as a way to invite participants to live coding practices through a system that applies simple commands to explore complex sound synthesis. Kiefer wants his work to be accessible to lay audience and novice live coders and he therefore applies the Web Audio API, an audio synthesis framework

that runs in most modern browsers. Charlie Roberts's *Gibber* system is also written in the Web Audio API, and it enables live updating of code, where the text can respond to sound through the events it defines as notation. Roberts also discusses public engagement through the use of visual elements to reinforce audience attention throughout performances. Finally, lOhannes m zmölnig critically explores the issue of how text is understood by the audience on screen, discussing his multi-layered representation of code, simultaneously hiding and revealing, thus lending itself to multiple readings by audience members.

In reflecting on the articles of this live coding special issue of IJPADM, the culture of live coding emerges as one that is not overly concerned with the act of programming itself, but rather with the techno-social aspects of its performative systems, representations of composition, rule-based improvisation, and performance activity. These critical discussions are not on technology per se, but rather on how audiences, artists and learners read, engage, and create with code that is developing in real-time, on the fly, or live.

Live coding is maturing as an artistic performance method. This history has been documented from its beginnings as a focussed practice in the early 2000s (see Collins et al. 2003), to Magnusson's (2014) discussion of products, practices, experiments, and performances drawing on 10 years of hindsight. The latter article speculates on live coding as a method that will become increasingly ubiquitous in diverse art forms, to the degree that we may not need the term *live coding* in the future. As this special issue demonstrates, the activities currently denoted as *live coding* are methods that can be seen to have historical precedents in the arts, but they also relate to problems studied in computer science, often under the term live programming (Tanimoto 2013). Presenting live coding in these broad terms, relating it to past practices and related research fields, makes it increasingly hard to incisively define. It appears that the popularity of live coding can be explained by its appeal to people growing up with interactive media who reject being subjectified as passive consumers of technology. Live coding represents a mindset of sorts that conceives of technology as fluid phenomena, offering an openness to change and exploration, such as live redesign of instruments, live rewriting of scores, or indeed of writing with 'shaky hands' in order to have no face.

References

Collins, Nick, Alex McLean, Julian Rohrhuber, and Adrian Ward. 2003. "Live Coding in Laptop Performance." *Organised Sound* 8 (3): 321–330.

Foucault, Michel. 1972. *The Archaeology of Knowledge and the Discourse of Language*. New York: Pantheon Books.

Magnusson, Thor. 2014. "Herding Cats: Observing Live Coding in the Wild." *Computer Music Journal* 38 (1): 8–16.

Tanimoto, Steve. 2013. "A Perspective on the Evolution of Live Programming." LIVE 2013 symposium proceedings, ICSE conference, San Francisco.

Thor Magnusson
University of Sussex, Brighton, UK
t.magnusson@sussex.ac.uk

Kate Sicchio
New York University, MAGNET (NYU Media and Games Network), Brooklyn, NY, USA
sicchio@nyu.edu