

Performing Thinking in Action: The Meletē of Live Coding

Emma Cocker

Reader in Fine Art

School of Art and Design

Nottingham Trent University

Dryden Street

Nottingham

NG1 4GG

0115 84 88654

emma.cocker@ntu.ac.uk

Emma Cocker is a writer-artist and Reader in Fine Art at Nottingham Trent University. Operating under the title Not Yet There, her research addresses the endeavour of creative labour focusing on models of (art) practice and subjectivity that resist the pressure of a single, stable position by remaining wilfully unresolved. Cocker's recent writing has been published in *Failure*, 2010; *Stillness in a Mobile World*, 2011; *Drawing a Hypothesis: Figures of Thought*, 2011; *Hyperdrawing: Beyond the Lines of Contemporary Art*, 2012; *On Not Knowing: How Artists Think*, 2013; *Reading/Feeling*, 2013 and *Cartographies of Exile: A New Spatial Literacy*, 2016. *The Yes of the No* is Cocker's first collection of writing published by Site Gallery, 2016. She is currently a key researcher on the project *Choreo-graphic Figures: Deviations from the Line*, 2014 – 2017.

Performing Thinking in Action: The Meletē of Live Coding

Within this article, live coding is conceived as a meletē, an Ancient Greek term used to describe a meditative thought experiment or exercise in thought, especially understood as a preparatory practice supporting other forms of critical — even ethical — action. Underpinned by the principle of performing its thinking through ‘showing the screen’, live coding involves ‘making visible’ the process of its own unfolding through the public sharing of live decision-making within improvisatory performance practice. Live coding can also be conceived as the performing of ‘thinking-in-action’, a live and embodied navigation of various critical thresholds, affordances and restraints, where its thinking-knowing cannot be easily transmitted nor is it strictly a latent knowledge or ‘know how’ activated through action. Live coding involves the live negotiation between receptivity and spontaneity, between the embodied and intuitive, between an immersive flow experience and split-attention, between human and machine, the known and not yet known. Moreover, in performing ‘thinking-in-action’, live coding emerges as an experimental site for reflecting on different perceptions and possibilities of temporal experience within live performance: for attending to the threshold between the live and mediated, between present and future-present, proposing even a quality of atemporality or aliveness.

Keywords: meletē, artistic research, thinking-in-action, process, praxis

Within this article, I address live coding as a dynamic model of both ‘performing thinking’ in action *and* the performing of ‘thinking-in-action’, a practice based on the principles of timing and timeliness, of invention and intervention. The reflections are based on my experiences — observations and conversations — as a critical interlocutor within two live coding research projects funded by the UK Arts and Humanities Research Council *Digital Transformations* theme: *Live Notation — Transforming Matters of Performance*, 2012, led by co-investigators Alex McLean and Hester Reeve, for exploring the possibilities of relating live coding (performing with programming languages) and live art (performing with actions) (Fig. 1 – 2); and *Weaving Codes / Coding Weaves* (2014 — 2016), led by co-investigators Alex McLean and Ellen Harlizius-Klück, working with Dave Griffiths and Kia Ng, for exploring the historical and theoretical points of resonance between ancient loom weaving and computer programming, the insights gained if we bring these activities together (Fig.3 – 4).

My intent is to explore *live coding in performance arts* not through attending to the techniques and technicalities of this emergent practice, through specific reference to examples of live coding as performance *production*. Without explicit expertise in either programming or musical/audio-visual improvisation, my intent is not to differentiate live coding as a new form of artistic expression, through elaboration on the particularity of its processes, languages and methods of public presentation. Instead, I explore how the process-oriented performativity of live coding relates to wider debates concerning the epistemic — even aesthetic-epistemological — potential of artistic endeavour. Here, one might ask, not only what are the new possibilities afforded to artistic performance through live coding, but also how might this practice contribute to the burgeoning field of artistic-performance *research*, and vice versa? Demonstrating a multi-modal model of sense-making emerging between the lines of musical-rhythmic, linguistic-verbal, spatial-visual and numerical-logical intelligences (Gardner, 1983), how might live coding offer insight into the commonalities, complementarity *and* differences between ways of knowing within science(s) and the arts? More specifically, I am interested in how the aesthetic enquiry within certain live coding practices can be conceived as a contemporary form of *meletē*, an Ancient Greek term used to describe a meditative thought experiment or exercise in thought, especially understood as a preparatory practice supporting other forms of critical — even ethical — action.

Whilst these ideas — including some elaboration on the notion of *meletē* — will be explored later as the article progresses, I want to begin by reflecting on how live coding ‘performs its thinking’ as a live event. Broadly speaking, live coding describes the improvisatory real-time composition of predominantly computer-generated audio-visual material, where the writing of code itself (or other executable instructions) is presented as a live event for an audience. Significantly, the writing of code is not just the means for generating audio-visual material, but additionally, this process is often folded into the audio-visual experience itself. Alongside witnessing the coder engaged in the live act of coding (labouring at their laptop), the code itself is also presented — often projected — real-time as it is being worked on as a visible part of the performance. Live coding asserts value for the *liveness* of its performance, moreover, the principle of ‘making visible’ the process of its own procedural unfolding. For live coders including the collective *slub* (Alex McLean, Dave Griffiths and Adrian Ward) ‘the preferred option for live coding is that of interpreted scripting languages, giving an immediate code and run aesthetic’ (Collins et al, 2003, 321). Here, a running command line is presented live to the audience whilst it is being written, real-time feedback affording the live coder the possibility of modifying and re-

writing the code synchronous to its execution. Live coding involves the presentation of a textual or graphic interface — using existing, hacked or homemade programming languages — through which the liveness of the *working out* within coding is revealed. Performing its thinking through the ‘showing of the screen’, live coding is a revelatory practice intent on showing its source code or operational principles *as* they are being written.ⁱ To expose the inner workings of a practice foregrounds the process itself, emphasising the methods and mechanics of production, the durational ‘taking place’ of something happening (live). It is a means for rendering communicable to an audience the dynamic experience of decision-making, the navigation of competing forces, the activity of working with and through obstacles or of ‘figuring’ something out.

Live coding wilfully unveils the underlying operational layer of activity beneath the more familiar, readable gestures of computational performance. The projected code reveals the presence of the programming language ‘behind’ the audio-visual effects witnessed within the live coding performance, moreover, makes tangible the process through which the live coder modifies and reworks this material, from the construction of the language itself, to the changing of its variables, parameters and functions. However, these modifications are not just conceptual workings, the logical manipulation of an abstract notational system. Rather there is an inherent physical, kinetic dimension to the live writing of code, a movement vocabulary of micro-adjustments, changes and shifts performed in the frantic shuttling of the cursor around the screen, in the flash points of activation and execution. Live coding as a live and present practice, the real-time toggling back and forth of the cursor, decisions made from *within* the continuity of a process (a running command line) rather than through a figured-out design applied from without. Live coding gives texture to the textual screen, transforming it into a stage of sorts upon which the cursor-coder’s thinking makes its moves. The cursor: live site of action, of backtracking and retraction, of addition and subtraction, cut and paste, deletion and erasure, insertion and manipulation. Analogous to the pulsing live body within other forms of performance, the flashing cursor marks the point of decision-making — of consciousness perhaps — within the live programming of code; the movement of the coder’s thinking as it oscillates between sense-making through the discontinuous, abstract notational form of code and the ‘continuous’ — even sensuous — experience of coding as a lived experience. Accordingly, the thinking that is performed within live coding is multi-modal; a linguistic-numerical computational logic combined with a musical-rhythmic ‘know-how’ of a more tacit (Polanyi, 1958) or even embodied kind, involving the tactility of active exploration or haptic perception which in James

Gibson's terms describes, 'The sensibility of the individual to the world adjacent to his body by use of his body' (1966, 97).

Whilst live coding draws on the tacit principles of 'know-how', the showing of the process — the 'making visible' of the operational thinking and working out — seems predicated less on a model of 'how to' (apply logical, algorithmic methodologies); and is instead performed in the key of 'try this', a democratising gesture or call towards the development of further exploratory, playful practice. Certainly, the revelation and live reworking of digital code through its performance is inherently advocatory, even political, involving showing and sharing the unfolding logic of a language so instrumental to contemporary life, but in which still few are fluent. Indeed, rather than simply 'users' of existing software products (whose source code remains hidden, inaccessible, undisclosed) for some live coders the production of the programming language itself is often an intrinsic part of practice and performance. However, the commitment to making the operational principles of coding 'visible' or even 'tangible' within the context of a performance practice often seems less about direct explication or instruction, an approach based on the pedagogical transmission of a *specific* method, its technics and techniques. In one sense, the act of revelation appears to be performed more as an invitation, even *incitement*. Here, the live coding performance has a constitutional function, helping to augment and inaugurate a community of existing and future live coders, through the revelation of a shared or 'common' performance language. In these terms, the 'thinking' shared in a live coding performance does not make *claim* to knowledge, but is made visible to others for them to actively modify, build upon and creatively develop themselves. However, the audience for live coding performances might comprise individuals familiar with and therefore capable of 'reading' the code as it is being written, alongside individuals who are code-illiterate for whom the unfolding code remains illegible at the level of its language content. Some live coder's remain relatively ambivalent to the audience's capacity for deciphering the code itself, which suggests that there is something *else* or *beyond* the revelation of a readable working method communicated through the 'showing of the screen'.

The thinking performed live and made visible within live coding shows something *beyond* its linguistic or methodological content. Here, following Paolo Virno, one might even argue that, 'what really counts is the act of enunciating and not the text of the enunciation' (2015, 35). In *Epistemologies of Aesthetics*, Dieter Mersch attempts to differentiate 'aesthetic thought' from the 'classical philosophical ideas about thinking thought, in particular equating thought and concepts or, since the linguistic turn, the

appropriation of thought by language’ (2015, 8). He argues how discursiveness — ‘making a statement or formulating an “argument” in the form of sentences’ (Mersch, 2015, 8) — and methodology — based upon a ‘scientific, i.e. methodological, research process’ (9) — have ‘advanced to become the main criteria for the production of epistēmē’, neither of which he claims are ‘particularly suited to artistic practice’ (9). Against this context, Mersch asks what ‘thought in other media’ might mean, where ‘thought is understood as a practice, as acting *with* materials, *in* materials, or *through* materials ... or *with* media, *in* media or *through* media’ (2015, 9 —10). Interestingly, Mersch is keen to avoid ‘favouring tacit knowledge as is the trend in science studies and the history of science’ (2015, 9 —10) in an attempt to differentiate an artistic or rather aesthetic mode of thought beyond a vocabulary of linguistic discursivity and process methodology, where the specificity — even alterity — of an aesthetic epistemology is made explicit. For Mersch, artistic thought ‘reveals itself in the form of those practices that “work in the work”, the “becoming” of the processes themselves. For this reason we speak here of thought as praxis and as performance’ (2015, 11). Moreover, he argues that the ‘decisive epistemic modus’ through which art performs, presents and exhibits its ‘work in the work’ is always one of *showing*: ‘we are dealing with “showings” that in equal measure reveal something and show themselves while in showing, hold themselves back [...] their métier is not representation, but presence’ (Mersch, 2015, 170). Additionally for Mersch, ‘aesthetics focus on the singular, on this-here or something that can be shown. Yet it is never clear what (*quid*) it is, only that (*quod*) it is’ (2015, 118 —119). Conceived in this sense, the performative act of ‘showing the screen’ within live coding might be considered less a device for revealing to an audience *what* or *how* something is unfolding (where meaning is legible at the level of linguistic or methodological reasoning), but purely that something *is* unfolding, the revelation of artistic thought ‘in action’ *with*, *in* and *through* the materiality and mediality of the performance itself.

Significantly for Mersch, the process of artistic thought involves ‘the stimulation of effects or leaps rather than directional intentions or calculated efforts that follow a precise plan and aim for closure in a manner imagined at the work’s inception’ (2015, 11). Here then, it could be argued that the performing of artistic or even aesthetic thinking within live coding happens only when it is *with*, *in* and *through* the materiality and mediality of the performance, through ‘surrender(ing) to the event and its experience’ (Mersch, 2015, 52). Admittedly, my own interest is in relation to those examples of live coding performance that are not so much concerned with the planning and designing of a ‘script’ in advance, but instead are *practised* through a contingent, improvisatory process, with the live running code

a site for testing the possibilities of *this* or *this* or *this* or *this*, for exploring the potential of ‘what if’.

Indeed, for Erin Manning and Brian Massumi, ‘Potential is not of the if-then. Potential is allied to what-if’ (2014, 41). They assert that performing in the key of what-if ‘is never a question of formally working something out in advance’ (Manning and Massumi, 2014, 43), but rather ‘a movement precise with training but still open to regeneration’ (44). Live coding pressures the ‘if-then’ thinking of computational logic towards the ‘what-if’ of speculative experimentation. Live coding is a hybrid — even liminal — practice, operating at a critical interstice between different disciplines, oscillating between a problem-solving modality and a problematising, questioning, even obstacle-generating tendency. More than a *strictly* ‘logical’ procedure, live coding involves the investigative practice of doing something as a way of knowing how something is done, moreover, for knowing how it might be changed, *swerved*, taken in a different direction. Live coding seemingly deviates a computational logical-numerical-linguistic mode of thinking (algorithmically) towards artistic application, involving not only a musical-rhythmic intelligence or sensibility, but also some of the more intuitive, haptic knowledge(s) commonly associated with creative thinking. Or alternatively, it draws attention to how an algorithmic approach to dynamic thought and action is *already* creative, underpinned by many of the principles considered intrinsic to artistic practice and research. Whilst grounded in a commitment to the *practise* of coding (a familiarity with and fluency in computational language), live coding performance is not one of simply showing one’s expertise (as rehearsed and scripted), but also seeks to create the germinal conditions wherein something unplanned for or unanticipated might arise.

The principle of improvisation — of beginning a performance without pre-set plan or script, without knowing where it will lead — necessarily embraces a sense of risk and uncertainty, the stepping off or away from what is known or certain towards a state of ‘not knowing’. Indeed, receptivity to the experience of ‘not knowing’ is necessary for invention and intervention within artistic enquiry.ⁱⁱ For Elizabeth Fisher and Rebecca Fortnum,

‘Artists often begin something without knowing how it will turn out. In practice, this translates as thinking through doing. Some of the methods used by artists to drive the creative process or work out problems can seem counter-intuitive or irrational: distraction or relinquishing control, embracing chance and collaboration, following a hunch rather than a rationale; privileging the senses over the intellect’ (2013, 7).

They argue that whilst such principles of ‘thinking through doing’ are often associated with a ‘largely negative lexicon’ (the *uncertain, invisible, incomprehensible*), at other times, ‘not knowing is not only to be overcome, but sought, explored and savoured; where failure, boredom, frustration and getting lost are constructively deployed’ (Fisher and Fortnum, 2013, 7). I would argue that like other forms of artistic enquiry, some live coders actively deploy these ‘tactics for not knowing’ (Cocker, 2013a) as a means for generating new ways of working (live). Rather than striving to successfully execute an already known and rehearsed ‘script’ (the repetition or revisitation of what already exists), the criticality of a performance could be conceived in relation to the possibility therein for encountering something new; in the challenge, provocation and even pleasure of the unexpected. Here, new forms of operating are provoked into being through the encounter with a situation unlike what has come before. Indeed, Sarat Maharaj differentiates between innovation (conceived as the ‘improvement and incremental adding to what is already there’) and a species of creativity that is ‘about discontinuity, about rupture, about production and emergence, and the spasmodic appearance of something entirely unexpected and new’ (2012). He uses the term *benightedness* to describe the creative ‘journey into the unforeseeable’ (Maharaj, 2012). For Maharaj, not knowing does not describe a state of ignorance, but rather the temporary *suspension* — a letting go, relinquishing or momentary knocking back — of fixed ideas and ways of operating (the *already* known) in order to allow in the possibilities of surprise.

Live coding performances *actively disclose* to an audience their moments of not knowing, of trial and error, of testing something out (through endless subtractive and additive procedures, the testing of constants and variables), moreover, the promise of not knowing (and the risk, uncertainty and sometimes even messiness therein) is arguably part of live coding’s improvisational performativity. Live coding also has a capacity for a level of self-reflexivity resonant with artistic enquiry, the concern therein not just for the production of performance, but also for the simultaneous disclosure of its ‘thinking process’. This self-reflexivity can also be seen in the folding back of live annotation into the live coding performance itself, presented through commentary, notes and marginalia arising in and through a given performance that are embedded within the textual frame of the code. Here, the operational language of code becomes interwoven with an expositional language intent on contemplating the terms of its own coming into being. Live coding is a practice alert to the circumstances or ‘occasionality’ of its own making, capable of reflexively documenting the unfolding of its own emergence. However the exposition of process (within live coding) is not always concerned with explication, but also has the capacity for

adding layers of complexity, further enriching the work itself. Indeed, within live coding (and artistic enquiry more broadly) the process of ‘making visible’ one’s reasoning is not always about transparency, but can sometimes be used (willfully) to create further opacity, desirable confusion and bewilderment (or even sometimes within live coding for humorous effect). Again, for Mersch, aesthetic reflexivity is performed in the key of *showing* rather than *saying*, a reflexivity which is not simply the practice of “‘reflecting on”, the foundation of which would be intentional consciousness’ (2015, 15). Rather, he asserts, ‘it is something that happened *through* making art [...] reflexivity takes place as an event within constellations and their composition in order to, *through* them, draw something out that could not otherwise be elicited’ (Mersch, 2015, 15).

Within live coding the principle of ‘making visible’ its process also extends beyond what might usually considered the work of art, the performance itself. The audience are often witness to the processes of preparation as much as what might be conventionally considered to be the ‘performance’. Referring to Andrew Sorensen’s practice, Stephen Ramsey states that what we witness at the start of the live coding performance, ‘is the equivalent of watching a symphony warm up’ (2016). I would argue that there is something more radical at play, where the revelation of preparation is less akin to the symphonic process of tuning in or up (which might often be witnessed in theatres but still not considered as part of the performance), but instead that for many live coders the back(stage) and front(stage) of the process appear somewhat indivisible. Within some live coding performances, there is no concealment of preparation or setting up, no cuts to be made after the fact — all is visible, all part of the work. Preparation becomes folded into the practice itself, is part of (and not prior to) the performance. The ‘making visible’ of the back-stage of the performance process (the activities of warming-up, of setting up the parameters of one’s environment, defining of functions), alongside the revelation of the script (as it is being modified synchronous to its execution or actualisation) differentiates live coding as a critical practice engaged in the interrogation and exposition of its own means of production and working processes. In this sense, one might situate live coding within a wider lineage of post-1960s performance practices, which reject virtuosity (and its wilful concealment of the practise or effort), instead exposing the mechanisms of production, including the incorporation of rehearsal and preparation into the space-time of the performance itself. For example, in Yvonne Rainer’s choreographic performances from the 1960s onwards (as Catherine Wood observes), ‘the “making” of the work is simultaneous with and dependent

upon its “doing” as performance; it is process *and* action. Tautologically, the work (of art) is simultaneous with the work (effort) of its execution’ (Wood, 2007, 27).

Significantly, as Paolo Virno states, what characterises the ‘work of the performing artist’ is that their ‘actions have no extrinsic goal. They don’t create a lasting product, since they aim only at their own occurrence. They don’t create new objects, but rather a contingent and singular event [...] the purpose of their activity coincides entirely with its own execution’ (2015, 22). Indeed, live coding performs its ‘code working’, which as Geoff Cox states, ‘draws attention to the work involved in writing code, as well as the work that code does once executed’ (2013, 39). Live coding is composed in front of the audience *through* its performance, unlike conventional forms of scripting for performance that are ‘decomposed’ or disappear as they are performed (Hall, 2007, p.6). Within live coding, the challenge seems less one of responding with learnt behaviour or an already rehearsed script, but rather in learning how to harness the potential unique to every contingent situation. As Nick Collins et al. note, live coders ‘work with programming languages, building their own custom software, tweaking or writing the programs themselves as they perform’ (2003, 1). Code is written as it is performed, a practice often referred to as ‘coding on the fly’ or ‘just-in-time coding’ or what I have named ‘kairotic coding’ (Cocker, 2013b). Live coding involves cultivating a working knowledge of a process such that it becomes ingrained in mind and muscle, activated at the fingertips, live and emergent to the situation. In this sense, live coding is a practice not based on knowing how to deal with a situation in advance, but rather a knowledge born of the moment, from having confidence. Confidence is the knowledge that the right decision will be made when required; it involves trusting — *tôi kairôi*: ‘to trust the moment’ (Hawhee, 2002, 18) — that a response will be performed appropriately and with skill at the propitious time. This is not about placing faith in a form of tacit knowledge, however, if this describes an already embodied *know-how*. Instead, what is activated is a *known-not* knowledge closer perhaps to Maharaj’s articulation of the flux of *no-how*, ‘distinct from the circuits of *know-how* that run on clearly spelled out methodological steel tracks. It is the rather unpredictable surge and ebb of potentialities and propensities’ (2009).

What emerges through live coding is a form of knowledge capable of working within situations that are newly encountered and not already known, or rather a mode of knowing that is activated and emerges simultaneous to the situation in which it is found, and that alone is adequate to the task of responding to that situation. Arguably, what is performed within live coding is qualitatively closer to the ‘immanent *intensification*’ of thinking that Alain Badiou (following Nietzsche) asserts ‘is not effectuated

anywhere else than where it is given — thought is effective in situ, it is what ... is intensified upon itself, or again, it is the movement of its own intensity' (Badiou, 2005, 58 — 9). This mode of knowledge as 'intensification' refuses the paradigm of commodification which is all too easily reabsorbed by capital but rather as Alain Pottage argues, its 'power exists only *en acte*, or *in actu*, (which) is to say — in language closer to Spinoza ... that is 'is' only in the process of its exercise' (1998, 22). For Manning and Massumi, 'thinkings-in-the-act' or 'thought in the act' refers to the immanent movement of thought as it is performed (2014). Not a knowledge that is easily banked and transferable then, but rather acquired *through* practice, moreover, activated only in-and-through practice. In these terms, the epistemological significance of live coding is not just in its 'performing thinking' in action (showing its thinking as a live event, the 'making visible' of the 'work of work'), but also in how it performs 'thinking-in-action' (where the specificity of its thought remains immanent and not anterior to the performance itself).

The 'thought in live coding' takes place *in* and *through* the materiality and mediality of the performance itself, a 'thinking-in-action' emerging synchronous to the live situation as it unfolds. In one sense, the 'occasionality' of this mode of 'thinking-in-action' might be conceived as *kairotic*, based on the principles of timing and timeliness, of invention and intervention. In other articles, I have reflected on the *kairotic* dimension of live coding (Cocker, 2013b, 2016) drawing specifically on the Ancient Greek rhetorical conceptualisation of the term, where *kairos* is often taken to mean 'timing' or the 'right time', a 'decisive' critical moment whose fleeting opportunity must be grasped before it passes. Here, *kairos* can be conceived as both a temporal 'opening' or critical moment (a 'nick' in time) *and* in Eric Charles White's terms, a 'will-to-invent' capable of responding to this opening: 'Kairos thus establishes the living present as point of departure or inspiration for a purely circumstantial activity of invention' (White, 1987, 13). However, a different accentual form of the term — *kairòs* — is used by philosopher Antonio Negri for describing the event of creating an adequate epistemology simultaneously to the knowledge/being that it attempts to describe. Negri designates the term *kairòs* to the 'restless' instant where naming and the thing named attain existence (in time), for which he draws example from the way that the poet 'vacillating, fixes the verse' (2003, 153). For Negri, '*kairòs* is the modality of time through which being opens itself, attracted by the void at the limit of time, and it thus decides to fill that void' (2003, 152.) He conceives the limit experience of *kairòs* as one of "being on the brink", as "being on a razor's edge", a point of rupture and of necessary decision (Negri, 2003, 152). Here, *kairòs* describes the radical temporality of the very moment of something new coming into being unique to that very moment, a mode

of immanent (and imminent) invention. Following Negri, live coding's thinking-in-action involves an improvisatory tendency performed through leaning into the void of the 'to come', a *kairotic* thinking-in-action wherein the unfolding future of the performance emerges simultaneously to its imagining.

However, it might be tempting to focus on the 'just in time' nature of live coding's decision-making processes as one that is necessarily connected to speed of thought urgent in the now of the present as it is *seized*. However, the performativity of timing and timeliness within *kairos* is not performed in haste or without due care and attention, but also relies on the *dual principles* of slowness and speed.

Paradoxically perhaps, the critical opportunity within the 'opening' of *kairos* (ready to be seized) might only be discerned through a slowing down of habitual flows and rhythms, thereby producing the necessary quality of 'attention'. Indeed, against the privileging of real-time performance — and narrowing of the feedback loop between coding and its execution through technological advancement — I advocate critical value for the gaps and lags within live coding performance conceived as reflective intervals for biding one's time and for deciding when to act. Indeed, live coding seems to be a practice not so much performed in the singular temporal present, as an experimental practice capable of extending, expanding, even stretching the sense of how the present is experienced. Daniel Stern points towards the concept of the 'three part present' developed by various philosophers from Husserl (1964) to Varela (1999). He notes how for Husserl the 'three part present' comprised: 'a present-of-the-present moment (not so different from the present instance of *chronos*, the passing point of moving time)', a 'past-of-the-present moment' (*retention*) and a 'future-of-the-present moment' (*protention*) (Stern, 2004, 27). Within this model, as Stern asserts, the 'future-of-the-present moment is part of the experience of the felt present moment because its foreshadow, even if vague, is acting at the present instant to give directionality and, at times, a sense of what is about to unfold' (2004, 27). It could be argued that live coding performances seemingly give live articulation to this sense of three-part present, where the performer's attention appears split between the present-of-the-present moment (the *as is*) and a 'future-of-the-present moment' (the yet-to-come). Within the live and present moment, the live coder is both attending and improvising in the moment (an experience perhaps synonymous with 'flow states') whilst also setting up the conditions for what will come next. Moreover, live coders are often engaged in the production of scaffolding, code thread written in the present that will be activated at a future point. As such, whilst improvising in the present-present and planning for a future-present, the coder is also attends to the "past-of-the-present moment", through backtracking and reactivating code lines already written, but presently dormant. In this

sense too, live coding is both live and *alive*, its code immediately executable and also capable of being put on hold, suspended until an auspicious point for bringing it to life.

Within the live coding performance, the experience of time can be felt as the continuous, indivisible flux of flow states or of immersion, whilst at the same time experienced also as tangibly discontinuous, comprising an infinite number of intervals whose fleeting potential is either to be seized or lost. Live coding's unfolding thinking-in-action takes place through the experience of a complex multi-temporality. Moreover, it is a practice that requires some prior knowledge of a process (what can be predicted or anticipated *before* or in advance); embodied 'know-how' activated *during, in* and *through* its performance, and a *kairotic* knowledge (a 'know when' yet arguably *known-not* knowledge that emerges simultaneous to — unique and in complete fidelity to — the emergent situation, through a process of leaning into the 'to-come'). My focus on the temporality within live coding in this article places emphasis on the immanent quality of thinking-in-action that happens through embodied engagement with the materiality and mediality of performance as it unfolds as a live event rather than through prior scripting in advance. Additionally, the future-oriented tendency of its improvisation — it perpetual leaning into the 'to-come' — is one that is underpinned by a sense of 'not knowing', harnessing the potential of chance and unpredictability. Somewhere between the known and the not yet known, between the predictable and serendipitous, live coding demands a level of process fluency that comes only with practise, at the same time as retaining the desire for unexpected combinations and possibilities arising from within a seemingly familiar language, a receptivity to encountering something new. For Elizabeth Grosz, a future-oriented approach to time affirms its potential as an 'open-ended and fundamentally active force — a materializing if not material — force whose movements and operations have an inherent element of surprise, unpredictability or newness' (1999, 4). She asserts that, 'Only if we open ourselves up to a time in which the future plays a structuring role in the value and effectivity of the past and present can we revel in the indeterminacy, the becoming, of time itself' (Grosz, 1999, 11). For Grosz, this practice of leaning into the future has epistemological as well as ethics implications — she asks:

'What would a science or a body of knowledge be like that, instead of invoking the criteria of repeatability and the guarantee of outcomes ... sought to endlessly experiment without drawing conclusions? Would such a science approximate the singularity and uniqueness of art or intuition? Could experimental techniques themselves be derived experimentally, artistically, inventively, nonteleologically?' (1999, 11).

Furthermore, she asks, how might such practices ‘provoke new modes of knowing, being, responding, and initiating’ (Grosz, 1999, 11). In one sense, Grosz could be speaking of live coding and its wider implications, where its experiments are neither repeatable or teleological as such, its knowledge capable of cultivating new modes of being in the world, rather than only to be applied as an expedient mode of agile problem-solving.

Thus far, I have drawn on various references in order to propose and explore a distinction between the performing-of-thinking in action within live coding (a making visible of its working process including the states of uncertainty and not knowing) and its performing of thinking-in-action (which refers more explicitly to the immanent and even *kairotic* nature of a mode of thought emerging synchronous to the performance itself). However, my own interest is also in how the performing of thinking practised within live coding might even augment other forms of critical action, cultivating the potential of a more ethical — perhaps even resistant — mode of human agency and subjectivity. Here, live coding’s thinking-in-action not only supports the unfolding of its own performance, but arguably has a capacity for opening up the possibility for new modes of being and behaving underpinned by the human qualities of attention, cognitive agility and tactical intelligence that are practised therein. It is in this sense that I am interested in how the performing of thinking in action within live coding might be conceived as a form of *meletē* or thought experiment. In his late work on the care of the self and hermeneutics of the subject, Michel Foucault observes how within Ancient Greece there were a number of terms — *meletē*, *meletan*, *meditation*, *meditari* — which roughly translate as ‘meditation’, referring to the philosophical exercise in or practice of thought acting on thought. For Foucault, the process of *meletē* involved a mode of internal reflection, the ‘preparation of thought on thought, of thought by thought — which prepares the individual to ... improvise’ (2001, 454). Conceived as a *meletē*, the improvisatory tendency of live coding might be disentangled from its resulting product and considered more as an end in itself, a site for practising the relation for example, between spontaneity and receptivity, between yielding and control, between the agency of the self and the forces of the external world. Indeed, I would argue that there are a range of qualities cultivated within the performance of live coding, where it could be approached as an exercise in thought for practising heightened attention; a sense of *que vive*, alertness, vigilance or watchfulness; for attuning to the experience of present-ness or perhaps even for becoming sensitised to the possibilities of a ‘three-part present’; for developing an ethical and elective rather than obedient relation to the rule; for working with limits as points of creative leverage rather than of control; for

converting chance and contingency into opportunity, for practising a ‘fall’ from what is known in order to encounter the as-yet-unknown.

Whilst these various qualities and capacities are performed in action within live coding, my assertion is that if practised in the key of a *meletē* they also have the potential to be applied to other situations and actions, indeed to the living of a life. The mode of thinking involved in the practice of *meletē* is not abstract contemplation but rather has a direct relation to action. As Foucault states, the *meletē* required real activity and attention: ‘It is not just a sort of withdrawal of thought playing freely on itself. It is real exercise ... whose basic function is to prepare the individual for what he will soon have to do’ (2001, 425). Furthermore, the *meletē* of live coding is not a private, introspective act, but rather one whose performativity requires the presence of others, the institution or perhaps even constitution of a ‘public’ that is both *witness* to its performing of thinking-in-action, whilst also actively part of the critical conditions that gives rise to the specificity of its performance. In this sense, the mode of thought performed through live coding is inherently relational and contingent. For Paolo Virno, ‘Contingency, instability, absence of purpose, inseparability between the ‘product’ and the actions that realise it, necessary institution of a public sphere: all of these define ethical and political conduct (and before that, game playing)’ (2015, 23). It is in these terms I would argue, that the *meletē* of live coding can be conceived explicitly as a *praxis*, where its mode of thought is less concerned with the development of theoretical knowledge (*theoria*), nor solely with a mode of making or production (*poiēsis*), but rather its thinking-in-action is inherently related to the enactment and exercise of a politics *of action*, moreover, of ethical-political action.ⁱⁱⁱ

References

- Arendt, Hannah. 1958. *The human condition*. Chicago: University of Chicago Press.
- Badiou, Alain, and Alberto Toscano. 2005. *Handbook of inaeconomics*. Stanford, California: Stanford University Press.
- Cocker, Emma. 2013b. Tactics for not knowing: preparing for the unexpected. In: Fisher, Elizabeth and Rebecca Fortnum, 2013. *On not knowing: how artists think*. London: Black Dog Publishing.
- Cocker, Emma. 2013a. Live notation: reflections on a kairotic practice. *Performance Research*. 18 (5): 69-76.
- Cocker, Emma. 2016. Weaving codes / coding weaves: Penelopean mētis and the weaver-coder’s kairos. *Textile: Journal of Cloth and Culture*.

- Collins, Nick, McLean, Alex, Rohrhuber, Julian, and Ward, Adrian. 2003. Live coding techniques for laptop performance. Available at <http://eprints.sussex.ac.uk/1295/>. Accessed on 15 May 2016.
- Cox, Geoff, and Alex McLean. 2013. *Speaking code: coding as aesthetic and political expression*. Cambridge, Mass: The MIT Press.
- Foucault, Michel, Frédéric Gros, François Ewald, and Alessandro Fontana. 2005. *The hermeneutics of the subject: lectures at the Collège de France, 1981-1982*. New York: Palgrave-Macmillan.
- Gardner, Howard. 1983. *Frames of mind: the theory of multiple intelligences*. New York: Basic Books.
- Gibson, James J. 1966. *The senses considered as perceptual systems*. Boston: Houghton Mifflin.
- Grosz, Elizabeth. 1999. *Becomings: explorations in time, memory and futures*. New York: Cornell University Press.
- Hall, John, 2007. *13 ways of talking about performance writing*. Plymouth: Plymouth College of Art Press.
- Hawhee, Debra. 2004. *Bodily arts: rhetoric and athletics in Ancient Greece*. Austin: University of Texas Press.
- Polanyi, Michael. 1958. *Personal knowledge; towards a post-critical philosophy*. Chicago: University of Chicago Press.
- Manning, Erin and Brian Massumi. 2014. *Thought in the act: passages in the ecology of experience*. Minneapolis: University of Minnesota Press.
- Mersch, Dieter. 2015. *Epistemologies of aesthetics*. Zurich and Berlin: diaphanes.
- Maharaj, Sarat. 2009. Know-how and No-How: stopgap notes on “method” in visual art as knowledge production. *Art & Research*, 2 (2). Available at <http://www.artandresearch.org.uk/v2n2/maharaj.html>2009, accessed on 15 May 2016.
- Maharaj, Sarat. 2012. Six impulses on art research. SARN Conference "We, the public", 26th-27th of April 2012. Available at <https://vimeo.com/60841270>, accessed on 15 May 2016.
- Pottage, Alain. 1998. Power as an art of contingency: Luhmann, Deleuze, Foucault. *Economy and Society*, 27, 1, 1998.
- Ramsey, Stephen. 2016. Algorithms are thoughts, chainsaws are tools. Available at <https://vimeo.com/9790850>. Accessed on 15 May 2016.
- Stern, Daniel, 2004. *The present moment in psychotherapy and everyday life*. New York: W. W. Norton & Company.
- TOPLAP manifesto. Available at <http://toplap.org/wiki/Manifesto>. Accessed on 15 May 2016.
- Virno, Paolo. 2015. *When the word becomes flesh: language and human nature*. South Pasadena, California: Semiotext(e) 2015
- Wood, Catherine. 2007. *Yvonne Rainer: the mind is a muscle*. London: Afterall.

i The TOPLAP manifesto makes the intention explicit: ‘Give us access to the performer’s mind, to the whole human instrument. Obscurantism is dangerous. Show us your screens ...

Code should be seen as well as heard, underlying algorithms viewed as well as their visual outcome'. Available at <http://toplap.org/wiki/Manifesto>. Accessed on 15 May 2016.

- ii Significantly, Mersch differentiates between the role of the experiment(al) within art and science: 'The scientific system is the medium through which experiments are first constituted ... In the arts, this relation is turned around; it is the *experiri* of the *experimentum* that is the medium *through* which artistic research takes place' (2015, 52).
- iii *Theoria*, *praxis* and *poiēsis* are the three Aristotelian concepts related to the 'activities of man' as outlined in Aristotle's *Nicomachean Ethics*. The notion of *praxis* is explicitly recuperated and developed by Hannah Arendt (1958).