

## **Publication: Routledge Companion to Music, Technology and Education (Editor Andrew King)**

### **Chapter Title: Learning from live coding**

**Authors:** Professor Pam Burnard, Franziska Florack (Faculty of Education, University of Cambridge), Professor Alan F. Blackwell, Dr. Sam Aaron (University of Cambridge Computer Laboratory), and Carrie Anne Philbin (Education Pioneer Raspberry Pi Foundation).

The pace of engagement with new digital technologies in music education is proving to be slow. A key issue – and a major barrier to entry is teacher skills and confidence to work with technology creatively. Following a concerted campaign from the technology industry, in Spring 2013, the UK Government agreed to introduce the subject of computing into the national curriculum for all children the age of five. This was in response to a significant fall in the number and quality of students applying to study computer science at degree level. Following the Henley Review of Music Education (2011), a process of change was underway with new music education hubs being set the challenge of keeping pace with and increasing digital skills and engagement with live coding performance in music.

### **Learning about the nature of live coding performance in music**

Live coding music performance is a growing international phenomenon. Live coding can be viewed as a way in which computer programming is used to communicate the musical intentions of the live coder to the computer, which then produces sound as output. Programming skills are embodied in the code, design, abstraction and implementation. Mostly this happens in the mind of the live coder. The live coder writes code to produce sounds in real time. It could be argued that live coding has its foundations in participatory<sup>1</sup> musical performance because live coding involves real-time aesthetic decisions, judgements and feedback while editing code (see narratives of improvised live coding performance practices discussed by Andrew Brown and Nick Collins in Burnard, 2012). Like jazz, live coding can fuse the practices and creativities of performance, improvisation and (pre-)compositional elements. It can be enacted in the immediate moment (constituting the ‘liveness’) in which the creator codes or uses the syntax of a particular programming language, which may or may not be adapted by the performer (Magnusson, 2013). The coding style may have an important role in inspiring certain musical and creative behaviours, communications and interactions employed in live coding performance as a musical and creative practice, but how is the processes of learning to code music defined and located? How do we evaluate the exciting notion of ‘liveness’? What are the outcomes for learning that can / should be assessed in music education settings?

Live coded performances of music can range from a curated piece to engaging /

---

<sup>1</sup> According to Thomas Turino, different cultural cohorts value certain musical fields and properties of music over others. He differentiates between two types of performance, with different social functions and responsibilities, and thus different sound features that make them work. Participatory and presentational performances differ fundamentally in that with participatory performance there are no artist-audience distinctions. The members of ensembles specialise in presentational performance.

motivating an audience to dance (such as at a nightclub or on a beach) or to both solo and ensemble live coding shared by jazz and electronic and laptop musicians in diverse performance spaces, with the projection of code onto a screen being an important and significant feature of both the performer's programming and non-programming actions. Live coding performances can occur in a variety of concert settings, from underground venues to galleries, and often feature projection of the code as part of the visual performance space. Live coding practices share similarities to live DJ practices where the coding involves a physical manipulation of live performance/compositional parameters, rather than textual/coding manipulation.

Equally, live coding music does not necessarily resemble DJ practices where the projection of the code as part of the visual performance space is central. Because there is both coding and computational thinking involved, the Sonic Pi platform or environment is built on computational data structures and musical data structures, both of which impact on the learning and expressive processes as has been shown in teaching computational thinking through musical live coding (Ruthmann, et al 2010). The coding activity may display curatorial skills, such as in the creation of a new piece from existing recordings, an arrangement of an old piece, or a newly improvised piece performed at a concert. Because, in many cases, most of the musical events of a performance takes place in a here-and-now context, each is a distinctive form in which individuals come together in order to explore a new angle on live improvised music (Burnard, 2012:176).

This raises questions as to whether a particular real time performance brings with it a set of ascribed values, or whether the creativity involved is a peculiar quality of the act or is something along a continuum involving a risk-laden to risk-free act of real time programming. Live coding introduces the driving force of change in activity as the notion of 'liveness' with composition and improvisation happening in the immediacy of performing and experimenting simultaneously.

This also raises the potential for claiming coding / syntax errors as: (a) a source of creativity; (b) a temporary obstacle to which the audience may be privy; or (c) an act of strategic improvisation (see Burnard, 2012 for a discussion of errors / mistakes as interesting 'collisions'). However, the very complex features and the unclear nature of what live coding performance is (Magnusson, 2013), the complexities and diversity of what constitutes aesthetic evaluation of live coding actions (Bell, 2013), what learners make of, and how they become engaged in, learning to code (Aaron *et al.*, 2015; Philben, 2013), and how audiences appreciate and evaluate live coding, whether in informal contexts (e.g. at home) or in formal, institutionalised contexts (e.g. at school) place new demands on music educators and music education.

Developing appropriate practices and the criteria used to assess live coding performance is a particular 'ultimate' and major challenge for music education (Finney and Burnard, 2006). It is one of the aims of the Sonic Pi: Live & Coding (SPL&C) project described and featured later in this chapter. Work samples are used to exemplify each category for each criterion in rubrics which offer forms of co-constructed assessment practices with learners (see web-link to Sonic Pi Live and Coding Teacher Toolkit (Launched 4<sup>th</sup> Nov 2014), a toolkit of resources developed to support delivery of the SPL&C model, including lesson plans plus

guidance notes, a set of short films, inspirations works by artists and Sonic Pi v2.0: <http://www.sonicpiliveandcoding.com>).

### **Partnership and collaboration models: the key to effective teacher learning?**

Another key factor in the SPL&C project (discussion to follow) is the role of outsourcing creative arts engagement through creative collaboration and partnerships with non-teaching artist professionals. Educational discourses offer a body of evidence on the key features of teachers' continuing professional development learning (CPDL - Cordingly, 2013) and, in particular, the evidence for the use of research and enquiry. Cordingly (2013) summarises the factors identified in research studies as being key to effective teacher learning. These are:

- Using specialist external expertise in a sustained way
- The giving and receiving of structured peer support within the school
- A professional dialogue rooted in evidence
- Mastering the art of 'unlearning'
- Embracing the act of performing, composing and experimenting / improvising simultaneously
- A focus on *why* things do / don't work as well as on *how* they work. This is an integration of theory and practice
- Sustained enquiry-oriented learning, usually undertaken over two terms or more
- Learning from observing others practice
- The use of exploratory tools and protocols to help learning

These elements are confirmed in the 2013 Royal Society for the Arts (RSA) / British Educational Research Association (BERA) inquiry into the impact on teachers of engaging in enquiry-based approaches to teacher learning and practice. The conclusion drawn from examining this body of evidence was that we need to build an approach based on enquiry-oriented learning, learning by doing and talking, and advancing engaged leadership roles to peripatetic teachers who are teaching arts professionals of popular music performance. As such, the Sonic Pi: Live & Coding project (SPL&C) sought to develop a model for arts led partnership that could transform music learning by exploring the creative potential of live coding to provide new pathways for young people into digital music. The research centered process worked across arts, technology and education partners with young people to develop software program Sonic Pi so that when used with a Raspberry Pi computer, it would become a musical instrument using code for composition, timbre and interaction. The SPL&C partnership came together in May 2013 and its implementation was seen to have the potential to create the conditions for enquiry-rich, collaborative learning and teaching with an outsourcing of creative arts engagement and the development of new

digital and technological practices in music curricula through ‘creative partnership’ models for non-teaching artist professionals in collaborative partnerships.

Despite the possibilities that digital technologies are acknowledged to offer computer science, computational education and music education, the incorporation of new technological practices into music curricula remains a challenging political, social and practical task. In many cases, ‘school music’ practices are presented in an abstract manner, where goalposts are based on competencies with schematised rubrics for measuring everything of value in the development of formal learning, much of which is removed from the everyday experience of young people (Harris, 2014). Because of this they can lack the authenticity of engagement.

In music education, skills *and* knowledge are considered central to the ‘subject culture’, which is implicated as something that involves both participatory and presentational events such as contemporary pop and classical music concerts. Digital musicking and live coding performances are potentially spaces for, and practices of, empowerment which are not presently a central or featured part of the curriculum in music education. Exploring live coding practices, liveness, running code and changing variables within a prewritten programme in real-time are simply not part of the subject culture, as yet. Incorporating Raspberry Pi<sup>2</sup> into music lessons, teaching and learning and responding to live coding, in the age of digital technology, is an important task for educators keen to build bridges between the smartphone-toting learners and their teachers.

Within the literature which explores live coding, authors highlight the importance of technical skill of the performer (Brown and Sorensen, 2007, 2009), the notions of audience enjoyment and the sharing of the performance (Collins, 2011; Hall, 2013; McCallum, 2011; Ogborn, 2014; Ward et al., 2004), the complexity of the manipulation (Collins, 2011), the immediate transfer of meaning (Roberts, 2014), the diversity of the tools involved (Collins, 2011; Ogborn, 2014) and the playful ‘liveness’ of the performance (Sorensen, Swift and Riddell, 2014). How do these ideas relate to the pathways and models for assessing coding and what can be learned from the diverse practices of live coders and how they can inform the field of music education?

This paper will address the issues of:

- a) how ‘liveness’ is defined and located in live coding music performance;
- b) how learning the practice of live coding is made explicit; and
- c) how collaborative partnerships provide rich opportunities for young people to develop digital music making

Insights on emerging computing-based learning environments based on collaborative partnerships that have developed as a result of this kind of work and the ways in

---

<sup>2</sup> The Raspberry Pi is a low cost (£25) credit-card sized computer invented in Cambridge and launched on 29 February 2012 with the aim of encouraging children and young people all over the world to learn programming. It was deliberately produced at very low cost in order to be accessible to all and to encourage exploration.

which coding can create new learning pathways, are discussed next.

### **Introducing the ‘context’ of Sonic Pi: Live & Coding – an Research and Development (R&D) project**

As mentioned earlier, Sonic Pi is a new open source software tool and platform for the Raspberry Pi computer, created by Sam Aaron. It was designed as a coding programme to enable school children to learn programming by creating music and learn to create digital music by programming.

In this section we share insights from a research and development project called ‘Sonic Pi: Live & Coding’ (funding was awarded by the Digital R&D Fund for the Arts by Nesta, AHRC and the Arts Council). The project was delivered in three distinct phases for the purpose of bring developing new practices to enable, empower, inform and inspire students to engage in opportunities to create digital music and develop new digital creativities in music. The project opens up and provides new ways of working that open up new educational and arts-led partnerships and, potentially, new digital creativities. The broad partnership comprises: two music education hubs (Cambridgeshire Music and Hertfordshire Music Service), two Bridge organisations (Norfolk and Norwich Festival Bridge and Royal Opera House Bridge), the project lead and arts partner (Cambridge Junction), the technology partner (the Raspberry Pi Foundation with the University of Cambridge Computer Laboratory) and the research partner (University of Cambridge, Faculty of Education).

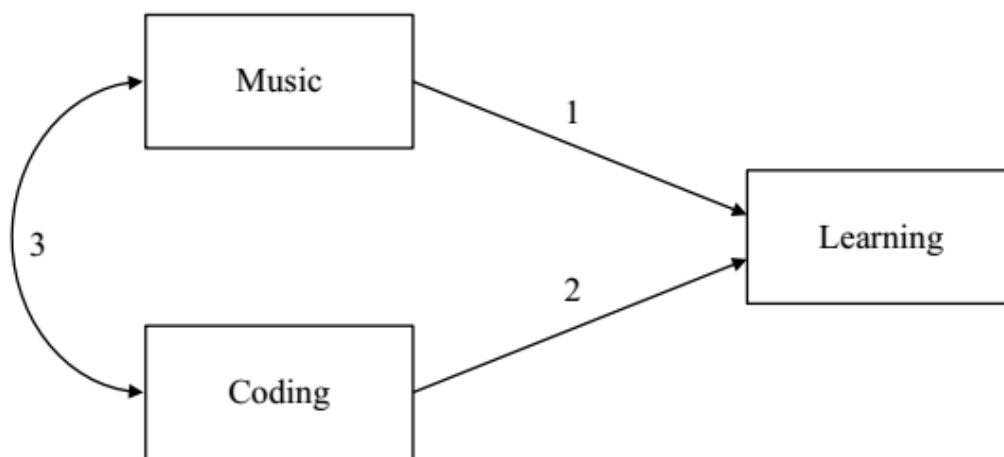
### **FIGURE 1: SPL&C partnerships GOES HERE**

All programming languages present the user with a conceptual model of the underlying behaviour of the computer that is to some extent fictive - a ‘black box inside a glass box’. The lexicon of a given language reflects the specific conceptual model of that language (sometimes referred to as a ‘programming paradigm’). The teaching curriculum in computer science, as with all subject curricula, presents a sequence of concepts in mathematical computer science, whose presentation is ordered by considerations such as familiarity from children’s everyday experience, mutual dependency, symbolic complexity and so on. There are some programming languages where the lexicon and habitus of practical programming resemble quite closely the concepts of mathematical computer science – especially and unsurprisingly those that are used in the work of mathematical computer science researchers.

However, in the case of the more pragmatically designed programming languages used in professional software engineering, there is often a degree of mismatch between the conceptual models useful for engineering and those of mathematical computer science that might form the basis of a formal curriculum. This fundamental

tension is resolved with a degree of compromise in ‘teaching languages’ that deliver some degree of practical functionality while also retaining a lexical correspondence to mathematical computer science. Many teaching languages are designed for use in application domains such as computer games, which are motivational for students, but also happen to be domains in which the tension can be ignored, in this case because there is no school syllabus for a rigorous and formal theory of computer games that might come into conflict with that of mathematical computer science. In the case of Sonic Pi, however, this regular compromise in computer science pedagogy becomes apparent because the conceptual progression of mathematical computer science does not straightforwardly correspond to the conceptual progression in music pedagogy.

We should consider the context of Sonic Pi in relation to digital music genres and how ‘liveness’ can be used as a unique functionality through which to establish diverse and variegated forms of music performance. As with many other features of Sonic Pi, this raises the question of whether the goal of the system is to express and support a conceptual progression in mathematical computer science pedagogy, or a progression in the understanding of fundamental concepts from music pedagogy. In fact, as Sonic Pi has become more ‘musical’ over the course of the Sonic Pi: Live & Coding project, the syntax and lexicon of the language have also become more musical. Research on the relationship between coding, music (and its pedagogy) and learning – as illustrated in Figure 2 – is still young and exploratory, and it is surely an interdisciplinary challenge.



**FIGURE 2: Relation between coding, music and learning GOES HERE**

This chapter is intended to advance the discourse with a focus on the role that Sonic Pi might play in order to enable and facilitate new pedagogies and better learning in both formal and informal educational contexts.

### **A sociocultural perspective on learning and computing in classrooms**

The theoretical framework of this study is a sociocultural perspective drawing on and related to Lev Semenovitch Vygotsky’s (1978) and post-Vygotskian scholars who played a critical role in the development of human learning and development and the

development of Cultural Historical Activity Theory (CHAT). In the field of digital design and related to human-computer interaction, and computer-supported collaborative work communities, Engestrom's (1999) work on activity systems offers a theoretical concepts that are critical for researchers and practitioners interested in new digital learning environments. Using a sociocultural approach, this research adopts a multiple case study methodology in order to identify and describe how learners engage with and learn from digital technology in formal and informal learning settings. Data was gathered over twelve weeks in three different intervals as shown in Table 1 below.

**TABLE 1: Design of the study GOES HERE**

**PHASE 1 -  
DEVELOPMENT**  
Creative workshops and  
training sessions

Data collection (observations, interviews, artefacts) took place during two six-week interventions, each at a Cambridgeshire and Hertfordshire community secondary school, followed by a week long Summer School at the Cambridge Junction. In the school settings, for two hours a week, 28 Year 8 students and 26 Year 9 students were shown how to code music in the Sonic Pi software. The course, which took place during the regular music lessons, was led by the peripatetic instrumental music



teacher, with assistance from the classroom music teacher, and two artists from the Juneau Project from Birmingham. The Sonic Pi: Live & Coding unit of work sought to address the learning of live coding performance and culminated in live coding paired / group performances. Nine students of different engagement levels were interviewed on the school day following the performance. The adolescent learners (secondary school students) featured in this article / presentation will be referred to as learners from School A.

Two weeks later, 60 children aged 10 to 16 took part in a coding Summer School at the Cambridge Junction. During this time, students were taught how to use the Sonic Pi software on the Raspberry Pi computer and they came into contact with a series of external coding artists. The week ended in a series of performances by the students, some of which included live coding. During the week, daily observations took place and all students completed written and filmed diaries in order to be eligible for the Arts Award. All students of this group will be referred to as Summer School participants. Seven groups of students were interviewed; three of the interviews became case studies in terms of observations in the last two days due to the students' high engagement (Case Study Groups A, B and C). The research focused on the topics of experience, learning and engagement.

### **Perspectives of the learners - What learning took place?**

The evidence showed learning in three different areas: music, coding and live coding. Subcategories which emerged included: (1) students displaying an increase in enthusiasm, skill and definition of music; and (2) students showing a better understanding of the technicalities of coding and an increase in their willingness to experiment and learn from mistakes. However, some of the students of lower ability, with limited engagement, said that they found coding 'difficult'. Looking at learning about live coding there was a third subcategory (3) which revealed themes of risk and the engagement and the thrill of the performance. Live coding music differs from traditional coding as code is produced and executed live on stage in front of an audience in order to create music. This chapter will particularly consider the themes which were supported by the strongest evidence through the triangulation of: (A) the different musical skills students felt they were learning by coding, (B) the students' increased willingness to experiment and (C) the students' perceived risk of live coding.

### ***Learning in music***

Students felt that they learned a range of musical skills during their engagement with Sonic Pi coding; however, they found it hard to express what these were exactly. During an interview, a student from the summer school noted that: 'I think it's not affecting my music skills that much but I am using my music skills to do this. [...] I think this helps my music and my music helps this' – effectively contradicting his opinion immediately. In which ways the course was affecting his skills, he could not describe. From observations and interviews it emerged that students learned about matching of instruments, the speed of music, how music is divided into beats, composing and that they were using and improving technical musical terms. Especially noteworthy were discoveries such as '[I learned that] not all music is made

up of notes' as they showed a growing understanding of the complexities of music and its characteristics. Students rarely used musical terms to describe the work they were engaged in: during interviews the word 'composition' was almost absent from students' recollection of their experiences.

### ***Learning in coding***

Although students were able to explain some of the perceived learning that they felt they were engaged in with regards to coding (such as, in the interview with Case Study Students Group A, '[I am proud of] how far we have come because we can now write a good decent song and use a cool instrument; we only learned how to use the program a few days ago', most students commented on their experience of *proactively* learning to code. They were not afraid to make mistakes, but valued the fact that the program would help them to improve, that mistakes would go unnoticed, that they could learn by asking questions and were able to work independently. Their pleasure at improving their programs and experimenting with code was very noticeable. The most evident echo of this attitude was a student from Case Study Students Group B, who described the process as extremely satisfying: 'I keep going and I want more of what I am doing. And it's really because of trial and error. I do something and then it breaks and then I try again and then I learn something. And that just keeps happening over and over'.

### ***Learning about live coding***

Although some more able students revelled in the idea of producing new code live on stage (Case Study Students Group B, during the interview: 'I liked the live coding battles and learning cool things to do' (of day 2)), a few distinct groups found the concept of live coding intimidating. Case Study Students Group C, two young girls, commented several times on the dangers of the code breaking and the resulting humiliation by the audience. This worry also extended to other people who were live coding on stage. During the interview one of the girls noted that: 'When we were doing the coding war thing someone, when it didn't work or something, everyone said "boo!" and the people on stage, I felt bad for them'. Noticeably, this sentiment was not echoed by members of the other focus groups who seemed to have more confidence in their coding ability. However, some students of School A seemed to agree with the girls, if for slightly different reasons. On day three of the course, the teacher asked a student whether he had tried changing 'things in real-time'. Clearly worried about the functionality of the code, the student answered that: 'I don't want to mess up my code'.

Analysing how, or the extent to which, live coding practices were evaluated or assessed in the practice of the learners is less clear than with the teachers. We observed one performance-oriented assessment practice at School A and engaged in an informal process of evaluation following the Summer School performances. We know the challenges teachers face. The language arising from conflicting ideologies in the field of music has an impact on the extent to which teachers see live coding of music as an assessable practice; whether they have the ability to articulate a comprehensive understanding of what live coding music is as a field of practice assessment; and what it might encompass as a community.

### ***Building collaboration and partnership***

As argued earlier, how learning is defined and located, what is learned, and how learning occurs, depends largely upon the learning culture of any classroom. The daily practices, which include the objects and outcomes of activity involved in teaching and learning in classrooms, have been found to be crucially important in terms of documenting, interpreting and theorising collaboration, participation, learning, and achievement. Engestrom (1998) is among several activity theorists who have attempted to make learning cultures visible by means of mapping or making explicit the hidden boundaries of time, space, divisions of labour, categorisations and assigned tasks and routines in and across learning (as an activity system)<sup>3</sup> and settings.

As shown in Table 1, in the design of the study, Phase 1 involved training sessions, Phase 2a the delivery in two schools and Phase 2b the delivery of Sonic Pi learning activity in a Summer School. The four settings represent four cases involving the learning object (to learn and engage in live coding using Sonic Pi; hence the project title ‘Sonic Pi: Live & Coding’). Each case represents an activity system involving a teaching team of peripatetic teachers, artists, computer scientists and music education hub facilitators. The object of the activity was for teachers (and artists) to first learn the skill of coding music and then to teach young people to learn and engage in the performance of live coding. The activity systems developed across the four settings where Sonic Pi was used to facilitate learning coding music performance are interconnected. The object of the activity, and the motive in all four settings, was to learn to live code music. The four training sessions were led by the computer scientist and the project manager. The six school-based sessions were led by peripatetic instrumental teachers, supported by classroom music teachers and IT teachers / technicians. They were also supported by artists delivering the Sonic Pi unit of work in School A and School B. The 5-day Summer School represented a consolidation of practices, hosted at a cultural venue, the Cambridge Junction; activities were led by the artists and assisted by the teachers. The software, Sonic Pi, offered a constellation of new pathways for coding music performance to the *subject* (being learners and how they elaborate, refine and concretise the concept of coding music performance using Sonic Pi). So, we see the formulation of learning by learners / teachers / artists / scientists. The work activity of the schoolteachers working collaboratively with the artists is called ‘teaching’.

The classroom trials, each of six weeks in duration, took place with a class of Key Stage 3 students. A Year 9 class at Freman College, Hertfordshire, took place in two 1 hour lessons per week for six weeks. A year 8 class at Coleridge Community College, Cambridgeshire, took part in six 1-hour-and-40-minute lessons. These lessons were led by the instrumental teacher with assistance from the classroom music teacher. The lead artists, Juneau Projects led on one of the lessons, midway through the intervention. In these sessions they focused on the performance side of live coding, returning in the final week to lead a final public performance with the class. Juneau Projects introduced design elements to enhance and support the performance aspects coding with the use of bespoke controllers to aid learning in how to get to grips with

---

<sup>3</sup> See Burnard and Younker (2010) for an elaboration of the use of activity theory in the analysis of classroom music learning.

Sonic Pi and coding music.

The engagement in coding gained a motivating force that gave a different shape and direction to activity when the setting was changed to the Summer School, where the object of activity was not about school-going formal learning but learning within a voluntary and fee paying Summer School informal learning setting.

The five-day Summer School was led by the Juneau Projects and held at Cambridge Junction. The venue was fully opened up to the 57 young 10-14 year-olds, giving them the freedom of the venue and its facilities. Tickets were £25 for the week. The Summer School participants encompassed children with a range of backgrounds and interests who perceived themselves at the outset as either musicians or coders.

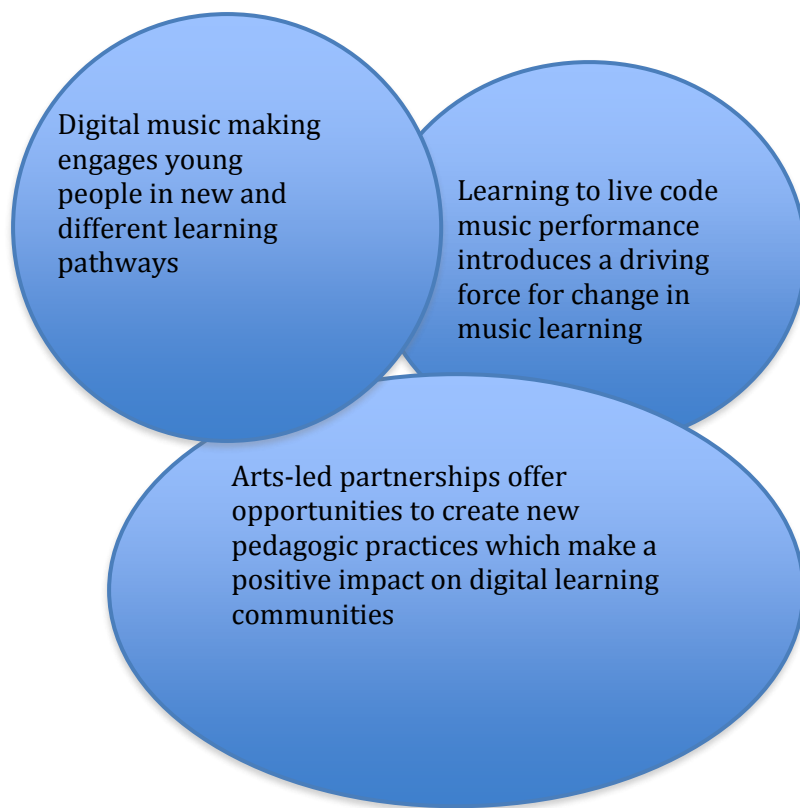
The Summer School started with an introduction to Raspberry Pi and Sonic Pi and included a series of guest live coders. The final performances at the end of the week demonstrated a range of possibilities in coding and use of Sonic Pi.

## **TABLE 2: Summer School Concert Items – Engaging digital music opportunities GOES HERE**

Experimenting with live coding informally in the Summer School was, for 57 young people, a new way of composing and performing music on computers. Summer school students considered themselves performers and creators of new music composing their own music via new pathways enabled by computing and programming.

Whilst this section only discusses some of the enabling learning pathways that entail the writing and editing of computer code as an integral part of digital music making and creating, it tries to isolate the preliminary themes which appeared across different phases of the project, including school-going and non-school-going settings, hence offering the strongest triangulation. It draws on learners', teachers', artists' and scientists' own perspectives as well as researcher observations on what kind of learning took place during the professionals' training, the 6-week school-based music curriculum course and the Sonic Pi Summer School. As the object of the activity was to learn live coding performance, the essential peculiarity of in-school classroom learning cultures sees the object and subject and division of labour reversed. The object of learning gains a motivating force, as do the possible goals and actions, in the Summer School setting where performance of live coding was emphasised.

Overall, it can be said that learning took place in the areas of music and coding, but that the evidence rather supported statements about the students' *attitudes to learning* (confusion, engagement, worry, curiosity, excitement), as opposed to *what* they were actually learning. Students seemed to struggle to identify specific skills (beyond statements like this, from interview day 2: 'I've learned how to make the songs and how they can sound good') but clearly showed some competence during practices and performances. What digital skills have been acquired can only be judged after an analysis of the students' code. From the analysis of the teachers and scientists it was found that questions, defining and solving problems, developing and refining new learning (as illustrated in Figure 3) were communicated through, and evidenced throughout the project.



**FIGURE 3: New learning GOES HERE**

## Final thoughts

The implications of these findings for teacher education are clear. We will require radical change in order to enable confident teaching of computing and coding generally, along with digital music making specifically. We need more experimentation and openness to risk taking, where music and computing teachers working in a fast-changing society can responsively co-create effective teaching strategies that empower and utilise the digital knowledge and skills that their students bring with them to the contemporary classroom. We will also require to build teacher confidence in a digital learning environment that encourages the students to equally participate, explore and experiment, engage and learn from digital music making.

Collaborations between experts, artists, teachers and learners and between individual learners and fellow learners can enable development of new pedagogic **practices** (findings corroborated with Hall, Thomson and Russell, 2007). There is immense potential for collaborative teaching of live coding and for digital music making to act as **a catalyst for educational change**.

## References

- Aaron, S., Blackwell, A. & Burnard, P. (in review) The development of Sonic Pi and its use in educational partnerships: Co-creating pedagogies for learning computer programming. *The Journal of Music Technology and Education*. Special Issue 'Live Coding in Music Education'.
- Amabile, T.M. (1996) *Creativity in Context*. Boulder, CO: Westview Press.
- Bell, R. (2013). Towards useful aesthetic evaluations of live coding. In *Proceedings of the International Computer Music Conference*, Perth, Australia. <http://renickbell.net/pdf/renick-bell-icmc-2013.pdf> (six pages)
- Black, P. & Wiliam, D. (2003). *Assessment for Learning: Putting it into Practice*. Maidenhead: Open University Press.
- Brown, A. R. & Sorensen, A. (2009). Interacting with generative music through live coding. *Contemporary Music Review*, 28(1), 17-29. doi:10.1080/07494460802663991
- Brown, A. R. & Sorensen, A. C. (2007). aa-cell in practice!: an approach to musical live coding. In *Proceedings of the International Computer Music Conference* (pp. 292-299). Copenhagen: International Computer Music Association. Retrieved from <http://www.computermusic.org/>
- Burnard, P. (2012). *Musical Creativities in Practice*. Oxford: OUP.
- Burnard, P. and Younker, B.A. (2010) Towards a broader conception of creativity in the music classroom: A case for using Engestrom's Activity Theory as a bases for research and characterizing group music-making practices. In R. Wright (Ed.) *Sociology of Music Education*, pp. 165-191, Aldershot: Ashgate.
- Collins, N. (2011). Live coding of consequence. *Leonardo*, 44(3), 207-211.

doi:10.1162/LEON\_a\_00164

Cordingly, P. (2013). 'The role of professional learning.' In C. McLaughlin (Ed.) *Teachers Learning Professional Development and Education*. (pp. 19-38). Cambridge: Cambridge University Press

Engestrom, Y. (1998). Reorganising the motivational sphere of classroom culture. In F. Seeger, J. Voigt & U. Wascgesio (Eds.). *The Culture of Mathematics Classrooms* (pp. 76-103). Cambridge. Cambridge University Press.

Engestrom, Y. (1999) 'Activity theory and individual and social transformation', in Y. Engestrom, M. Reijio and R.L. Punamaki (Eds) *Perspectives on Activity Theory*, pp. 20-40). Cambridge, New York: Cambridge University Press.

Finney, J., and Burnard, P. (Eds). (2008) *Music Education with Digital Technology*. London: Continuum.

Hall, T. (2013). Sharing electronic music performances. In *Visualise: Making Art in Context* (pp. 60-63). Cambridge: Anglia Ruskin University. Retrieved from <http://www.ludions.com/texts/2013b/>

Hall, C., Thomson, P. & Russell, L. (2007). Teaching like an artists: The pedagogic identities and practices of artists in schools. *British Educational Research Journal*, 37(5), 935-954.

Harris, A. (2014) *The Creative Turn: Toward a New Aesthetic Imaginary*. The Netherlands: Sense Publishers.

Hebert, D. G. (2010). Ethnicity and music education: sociological dimensions. In R. Wright (Ed) *Sociology and Music Education* (pp. 93-109). Aldershot: Asghate.

Henley, D. (2011) *Music Education in England: A review by Darren Henley for the Department for Education and the Department for Culture, Media and Sport*. Department for Education and Department for Culture, Media and Sport. <https://www.gov.uk/government/publications/music-education-england-a-review-by-darren-henley-for-the-department-for-education-and-the-department-for-culture-media-and-sport>.

Magnusson, T. (2013). The threnoscope: A musical work for live coding performance. In *First International Workshop on Live Programming* in conjunction with ICSE 2013 May 18-26<sup>th</sup>, 2013. San Francisco, CA. <http://liveprogramming.github.io/2013/>

McCallum, L. (2011). *Show Us Your Screens*. Retrieved from <http://vimeo.com/20241649>

Ogborn, D. (2014). Live coding in a scalable, participatory laptop orchestra. *Computer Music Journal*, 38(1), 17-30. doi:10.1162/COMJ\_a\_00217

Roberts, C. (2014). *Immediacy In Creative Coding Environments*. University of California at Santa Barbara.

Philben, C. A. (2013). *Adventures in Raspberry Pi*. London: Wiley.

Ruthmann, A., Heines, J.M., Greher, G.R., Laidler, P, and Saulters, C. (2010) Teaching computational thinking through musical live coding in scratch. <http://scratched.gse.harvard.edu/sites/default/files/sigcse2010paper-heines.pdf>.

Sorensen, A., Swift, B. & Riddell, A. (2014). The many meanings of live coding. *Computer Music Journal*, 38(1), 65-76.

Turino, T. (2008). *Music as Social Life: The Politics of Participation*. Chicago: University of Chicago Press.

Ward, A., Rohrhuber, J., Olofsson, F., McLean, A., Griffiths, D., Collins, N. & Alexander, A. (2004). Live algorithm programming and a temporary organisation for its promotion. In O. Goriunova & A. Shulgin (Eds.). Presented at the read\_me --- Software Art and Cultures. Retrieved from <http://art.runme.org/1107861145-2780-0/livecoding.pdf>

Vygotsky, L.S. (1978) *Mind in Society: The Development of Higher Psychological Processes*, Cambridge, MA: Harvard University Press.