Show Them My Screen: Mirroring a Laptop Screen as an Expressive and Communicative Means in Computer Music

Sang Won Lee
Computer Science Department
Virginia Tech
sangwonlee@vt.edu

ABSTRACT

Modern computer music performances often involve a musical instrument that is primarily digital; software runs on a computer, and the physical form of the instrument is the computer. In such a practice, the performance interface is rendered on a computer screen for the performer. There has been a concern in using a laptop as a musical instrument from the audience's perspective, in that having "a laptop performer sitting behind the screen" makes it difficult for the audience to understand how the performer is creating music. Mirroring a computer screen on a projection screen has been one way to address the concern and reveal the performer's instrument. This paper introduces and discusses the author's computer music practice, in which a performer actively considers screen mirroring as an essential part of the performance, beyond visualization of music. In this case, screen mirroring is not complementary, but inevitable from the inception of the performance. The related works listed within explore various roles of screen mirroring in computer music performance and helps us understand empirical and logistical findings in such practices.

Author Keywords

Laptop Performance, Live Coding, Audience Communication

CCS Concepts

•**Applied computing → Sound and music computing;** Performing arts;

1. INTRODUCTION

Modern computer music performances often involve a laptop performer, a musician sitting behind a laptop playing music on stage. Oftentimes, the musical instrument is a computer program, and the physical form of the instrument is the computer. Since its emergence, a laptop performance has become a prominent trend in computer music, creating novel musical practices such as Laptop Orchestra and Live Coding [9, 13, 37]. In this setting, the performance interface is rendered on a computer screen primarily for the performer to get visual feedback from the input. However, there has been a concern from the audience's perspective that having a laptop performer sitting behind the screen with visual fixation on the screen makes it difficult for the audience to understand what the performer is doing [36, 37]. One response to such a complaint is to reveal (mirror) the performer's screen on a projection screen in the performance space. This proves that the performer is not merely checking their email, but is instead engaged with a live performance [7]. This tradition has become Canonical especially in live coding music [9].

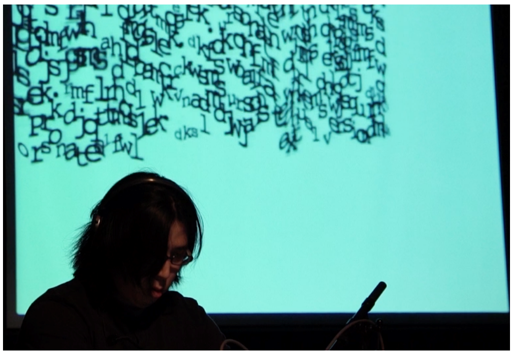

Figure 1: Mirroring the screen in *Live Writing: Gloomy Streets* - What is on the projection screen is the interface that the performer is playing and seeing

While the idea of mirroring a computer screen as-is passively addresses this concern, researchers and musicians have been actively using screen mirroring as an opportunity to enhance the expressivity of their performances and to develop a novel music practice. One widely used approach involves overlaying a visualization on top of a performance interface (e.g., live coding editors), using a programming environment as a canvas for generative visuals as well as a performance interface for generative music. In this paper, I introduce other types of laptop-based music performance in which screen mirroring is inevitable from the inception of the performance; this is consistent with my own practice as well.

The scope of musical works covered in this paper includes only those cases in which visuals on the screen are mirrored; that is, the projected screen is exactly the same as the screen the performer is seeing and using (See Fig. 1). It should be noted that this is not a comprehensive survey of musical performances that involve mirroring a computer screen, particularly because of the lack of specific documentation needed to confirm screen mirroring in others' artworks. In addition, this paper does not cover visual generative art, including visual live coding and visualization of music, because the goal is instead to discuss the mirroring screen technique beyond the ways in which it has been used in live coding and it is relatively well documented in [17, 27, 31]. It is my hope that this paper can serve as a useful reference

NIME'19, June 3-6, 2019, Federal University of Rio Grande do Sul, Porto Alegre, Brazil.

for researchers and practitioners understand the challenges and opportunities of mirroring a computer screen, both in practice and in the future music performance system design.

2. RELATED WORKS

Contemporary computer-based musical performances create tension between shifting to aural performativity [36] and reinvigorating visual performativity [33]. This paper suggests that simply revealing the laptop screen to the audience can support the latter approach.

2.1 Hiding Behind the Screen

While software is dominant in today's music-making processes, it seems that musicians have been avoiding revealing purely software-based musical instruments. One can argue that this is because of the discomfort the audience has when the performer behind a laptop screen does not convey any apparent relationship with their aural experience, rather asserting that the audience should now change [36]. The audience may have trouble in understanding the performer's activity in audio-only, laptop-based performances, depending on their experience in computer music, and their background knowledge of software [5]. It is not common to find musical performances in which musicians mirror the screen of their laptop, which runs an audio programming environment (e.g. MAX/MSP, Pure Data, SuperCollider) or digital audio workstation (Ableton Live, Logic), considering how widely such software is used in live performances; one exception to this tendency is live coding, which we will discuss in the following section. Even when software is used for music performances, it is often observed that musicians consider visual and corporeal aspects of the performance, typically combining a laptop performance together with other performers, or other types of external, gesture-based interface — a MIDI controller, for example. Alternatively, there can be visuals that are separate from the performance interface used by performers. It seems that musicians are hesitant to share purely software-based musical instruments with their audiences as-is, even though a graphical user interface on the screen can also supply a visual aspect of the performance.

This tendency of avoiding GUI-based musical instruments remains for (purely) laptop-based instruments. Even when researchers find expressivity in the native input devices of a laptop, they often want the audience to recognize the physical gestures of using a laptop, rather than limiting the visible aspect of the performance to a graphical user interface. For example, in the case of *Clix*, keypresses on laptop keyboards are the primary gestures; there is no visual feedback associated with that action on the screen beyond simple logging [13]. A set of design principles was suggested to design laptop-based instruments that avoid a need for visual fixations [12]. This often yields an eccentric approach; *Laptop Accordion* is one of such examples, using the gesture of opening and closing a laptop for sound generation [30]. While this can certainly be effective in audience communication, performers are not allowed to use a laptop in the way that they are most proficient with using native input technologies [13]. However, there is one exception in this tendency: the live coding musical practice compels musicians to reveal their musical instrument, which is a programming language, and to share their screen as-is.

2.2 Live Coding: Show Us Your Screens

Live coding is a musical practice in which performers are guided to reveal their screens [9]. This principle is encapsulated well in a statement from *TOPLAP's*[1] (a live coding community) manifesto: "Obscurantism is dangerous. Show us your screens." In this practice, it is typical to project program code text in the performance space for the audience to better understand the algorithmic and generative nature of the music-making process. The highly visible nature of the coding process makes the performance aspect of programming explicit to the audience. Live coding researchers have attempted to take advantage of projection screens to promote audience communication. One such attempt involves developing a graphical programming language, with the motivation of offering visual representations of algorithms that communicate to audiences more effectively [25, 26, 28, 29]. Another approach involves separate live coders (or teams of live coders) for visuals and music, respectively [11, 17].

The requirement of mirroring live coders' computer screens has been diversified, evolved, and reinterpreted in various directions. For example, projecting multiple computer screens at once is limited by the projection screens and projectors available. This led one live coding ensemble to create a dedicated visualization technique, which is technically not screen mirroring [14, 40, 24]. For certain styles of live-coded music — such as electronic dance music — projecting a computer screen that shows a code editor may even have little to no effect on an audience, given that the audience is expected to dance, rather than watch a screen under a dazzling light show [8].

Mirroring a musician's screen will be a continuing practice in live coding performances. The purpose of this paper is to introduce how the author's practice of mirroring a laptop screen, influenced by the live coding community, has been used to enhance audience communication and to motivate novel audiovisual music performances.

3. SCREEN MIRRORING FOR AUDIENCE COMMUNICATION

In this section, I discuss musical performances, including my own, in which the performance interface is mirrored on a projection screen and designed to help communicate with the audience.

3.1 Screen-mediated Audience Participation

In interactive music performances involving audience participation, mirroring the performer's interface on a projection screen can help to share information and facilitate participation. In particular, revealing the performer's interface is an effective method of manifesting how a performer is connected to the audience members, accounting for the ways that the audience contributes to the sonic outcome.

(Manifesting the distributed nature) In one audience participation piece, *echobo*, a projection screen is used to mirror the interface seen by an on-stage performer, who is using a tablet [23]. The audience are guided to play a mobile musical instrument on their smartphones, and the only sound of the piece comes from their devices. While the on-stage performer alone cannot make any sound, the aggregate sound generated by a large audience's smartphones creates a background harmonic structure composed of a dense, stochastic mass of notes in a selected scale. The performer's interface shows a set of blocks in a 2D arrangement, and the performer presses arrow keys to move a square-shaped cursor and select a chord block in the space(See Fig. 2). Audience members then use their smartphones to play instruments whose key (e.g., a given minor or major scale) is controlled by the on-stage performer. In this case,the

[1] http://www.toplap.org/

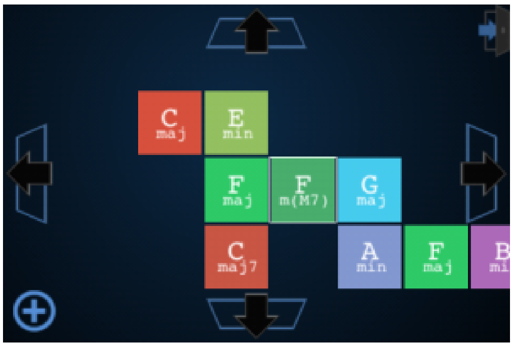

Figure 2: The performer's interface of *echobo* projected on the screen helps the audience create a sense of a connected ensemble.

performer's interface is designed to effectively convey the relationship between the on-stage performer's control over the chord structure of the music and the sound the audience generates; the visual representation of the chord can help the audience understand the idea of chord progression. If the performer's interface were not shared with the audience, the aggregate sonic outcome, which changes unpredictably, could confuse and interrupt the audience. The audience's ability to associate the performer's control with the synchronized chord changes helps to create a sense of a connected ensemble.

(Facilitating Social Interaction) A projection screen can be used to facilitate social interaction among audience members. In *Crowd in C[loud]*, the performer's interface rendered on a projection screen shows collaborative or competitive information in real time, and such information can encourage social interaction to sustain audience participation [10, 19]. For example, audience members can find their screen names and see how many other users liked their melody ("liked"), and how many of them are playing that melody at the moment ("crowded"). In addition, the most-liked screen name and the most-crowded screen name are displayed for recognition. In another audience participation framework, *massMobile*, displaying the collective outcome of audience responses effectively facilitates the emergence of group behaviors, with audience members changing their votes to join the leading group and thus collectively effecting a change [38].

(Tools for Complementary Design and Textual Communication) Information displayed on the projection screen can serve as a shared visual context to help simplify the audience's interface design. User interfaces for audience use must be fairly limited, since audience members do not have enough time to become familiar with a complex user interface, given the limited duration of the performance [19, 38]. It is thus crucial to make the instrument as compact as possible. Using the projection screen to display additional information can effectively convey a complex idea, with richly expressed textual and visual modalities, without interrupting the performance by offering additional features to the audience that are not central to the performance. Given that only non-verbal communication is allowed in a musical performance, this is a very effective method of engaging an audience. For example, a live coder can type a chat message to communicate with the audience and elicit a verbal response [20], or annotate displayed graphics with handwritten text [32][2].

3.2 Visualaudio (not audiovisual) performance

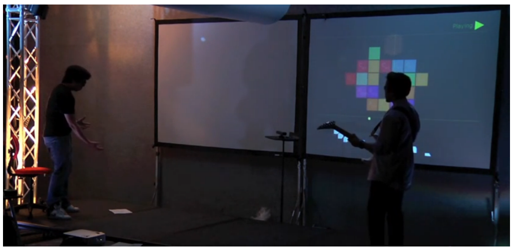

Figure 3: Mirroring the screen in *Crossole* - the performer (left) directly manipulates virtual objects on the projection screen

Audiovisual performance practices include visualization; visuals are either generated in response to sound, or created separately from of the sound. However, audio and music can similarly be a response to the visuals produced by a musician; this is a performance practice in which visuals precede music in the conceptual sound generation mechanism. *Live Coding YouTube* is one such case, in which manipulating videos is the primary gesture of musical performance [18]. The sound of the music comes exclusively from playing YouTube videos — by live coding them — without any audio processing techniques: retrieving a video, playing it, and specifying the time from which playback should begin. Both the performer and the audience need to be able to see the videos, as this is both the core concept and the only sound generation mechanism used in the performance.In this setting, visuals connect a performer's input to the sonic outcome (code => visual => music), and mirroring a live coder's performance interface is not only about revealing the input, but also about revealing the sound itself.

Live Writing is another case in which visuals are conceptually embedded in a musical performance (See Fig. 1) [22]. In this performance practice (or writing practice), a performer writes a poem, and the poem is the only input that advances the music and triggers sonic events. The generated music is a composed sonification of the poem, and the composition process cannot be detached from the content of the poem. Naturally, the audience experience is an integrated experience of three different modalities: textual (reading the poem), audible (listening to the sonification of the poem and the writing activities), and visual (visualization of sound on top of the poem text, connecting two different modalities) [21]. In these two cases, it is impossible to convey the idea of the performance practice without showing the performer's interface, as the visual nature of the composition and performance in its inception.

3.3 Virtual Representation of Music Control

NIME researchers and practitioners often create their own screen-based musical interfaces, which are often graphical, interactive applications. In this regard, mirroring the performer's screen can be an effective (and arguably inevitable method) of helping the audience become immersed into the virtual world that a performer is in.

Crossole is one example of a musical instrument designed with a metaphor of controlling virtual objects [34]. The idea originates from a well-known sci-fi movie, *Minority Report*, which inspired a number of research projects involving mid-air gestural control[3] [1, 39]. The instrument is designed for building a chord progression structure resembling a crossword puzzle; the performer navigates through chord blocks (See Fig. 3). In such a situation, it is natural for a performer to share the screen that they control with the audi-

[2] https://youtu.be/Tdj5e82nPHQ

[3] https://youtu.be/NwVBzx0LMNQ

ence. Sometimes, screen mirroring is not enough to account for a performer's activity, especially when it involves touch screen input — as may be the case with mobile devices or a tabletop interface. In these contexts, musicians often choose to capture a top-down view of the physical screen and project this image instead, so that the audience can see the screen as well as the performer's hands [32]. In addition to graphically represented interfaces, many game-based music-making applications follow the practice of mirroring a player's or observer's screen, where the musician becomes a game player and game activity is sonified as a live music piece [2, 6, 15, 16].

4. SCREEN MIRRORING IN PRACTICE

In this section, I introduce some of the practical issues that I have experienced, particularly regarding screen mirroring.

4.1 Where Do I Stand When I Need to Be Shown?

An interesting practical issue arises when a laptop performance incorporates a gesture-based interface and the screen is mirrored: where the performer stands, and which direction the performer faces. A performer typically stands either facing the projection screen or facing a laptop, which in turn often faces the back of the stage.In *Crossole*, the performer must stand a few meters away from the depth camera (Kinect) with no objects in between, which forces the laptop to be placed behind the camera [34]. This makes the choice of a performer performing on virtual elements shown in a projection much more plausible for both performer and audience, instead of controlling tiny virtual blocks on a laptop screen several meters away from the performer and invisible to the audience. However, this results in an awkward position in which the performer must stand with their back turned to the audience. In addition, the audience has no anterior view of the performer, making it likely that they will miss out on rich, non-verbal communication (e.g., facial expressions). In the concert where *Crossole* premiered, the performer and stage crews had a brief debate about how to locate the camera, which direction the performer would face, and whether the performer should perform watching the projection screen or the laptop screen; none of these are common problems in stage production.For the *Crossole* performance, this resulted in a strange stage setup in which performers stood on the side of the stage, half-facing the projection screen (See Fig. 3)[4].

Indeed, it seems that there exists some tension between facing the projection screen (with the audience behind the performer) and facing the audience, especially for pieces that incorporate gesture-based interfaces. In the ICMC performance of *Carillon*[5], the composer and performer decided to simply place the laptop facing the projection screen, performing with the audience behind his back. This creates an intuitive mapping between the performer's gestures and their results, for the audience to better understand the idea of controlling music in a virtual world. However, the audience did not have an anterior view of the performer, and the performer did not have a view of the audience. This situation limits communication between the two. Another performance of the same piece maintained the configuration in which the performer faces the projection screen, but the performer was moved to stand behind the audience[6]. At first, this sounds like a resolution to the problem, but one quickly realizes from the performance video that audience members cannot see the performer and the projection screen at the same time. This may ruin the perceived intimacy between the audiovisuals on the screen and the performer's gestures.

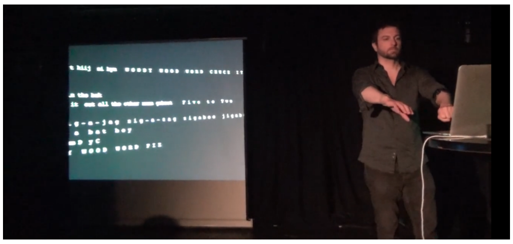

Figure 4: Mirroring screen in *Live Writing: Jimmy Raps* - The performer faces the laptop and the audience sees the mirrored screen. The visuals on the projection screen are laterally inverse to the laptop screen, so the performer's gestures cause the visuals to moves in the "wrong" direction.

Facing the audience and staying behind the (mirrored) laptop screen do not resolve the issue either. In the second iteration of Live Writing practice, *Jimmy Raps*, the composer and performer decided to type a poem with a gesture-based interface, MTO, controlling both the visuals and the sound[7] [35]. The musician chose to use a laptop both for input (keyboard) and output (visuals on the laptop screen, mirrored on the projection). The laptop and laptop stand block the audience's view of the composer's gestural expressions, prompting him to take a few steps back from the laptop for parts that are dedicated to gestural control. In addition, the visuals are controlled in a direction that appears to be opposite to that of his arm gestures (See Fig. 4). Due to the nature of *Live Writing* practice, simply flipping the image horizontally is not a solution, as the poem text would become unreadable. It seems that there is no perfect solution yet, and the constraints of various performance spaces (audience seats, reconfigurability, location and scale of projection screen) need to be taken into account on a per-piece basis. This makes it challenging for practitioners using such interfaces to figure out these details until they visit the performance space. Given the rise of software-based instruments and VR-based instruments, in which a screen needs to be shared with the audience, novel opportunities in perspective sharing for computer music performances may arise for future research in performing technology, virtual reality, and augmented reality. In [3], Berthaut, et al. pioneered in creating a mixed-reality performance environment in which visuals are projected on a transparent display in which a performer can face the audience, both audience and the performer can confront the virtual objects rendered on reflective transparent surfaces, and the audience can see the performer through the transparent surface. In addition, many of the performance setups explored in this section can be attributed with a set of dimensions suggested in [4].

4.2 Other practical issues in screen mirroring

When a musician decides to reveal their screen for the benefit of audience communication, a set of poorly documented technical and logistical issues arises — including problems as small as the need to warm up the projector before the performance starts. While performance spaces typically provide projection equipment, the technical level of the equipment provided can vary drastically, ranging from a simple projector and projector stand to a high-quality, in-house projection system, in which case a stage crew at the mixer

[4] https://youtu.be/Vy3Z4XRsFv4
[5] https://youtu.be/_hxka3PJL34
[6] https://youtu.be/AG6TiKvhpsk
[7] https://vimeo.com/220087603

has complete control over the projection. Therefore, it is important to understand the capabilities afforded by the performance space prior to the concert.

4.2.1 Know What They Will See

A musician needs a well thought-out plan as to what the very first visual presented to the audience will be. Without some foresight, a performer may end up beginning a performance with a log-in screen, waking up the laptop screen. Or, this open nature can be intentionally selected. For example, in the case of *Live Coding YouTube* and *Crowd In C[loud]*, I have initiated the pieces from a clean desktop computer screen — which could have been rather embarrassing in other performances — and begin by launching a web browser and typing the URL to the application. This choice was made for the audience to realize that both performance environments are built entirely on a web browser, and that there is no additional software involved (such as a digital audio workstation, local web server, or audio programming environment). This shows off the advances made in web audio technology, and the affordance that anybody can visit the URL to perform a piece, which they may not have known otherwise.

Similarly, a musical performance involving a mirrored computer screen may be interrupted by unexpected incidents, including normal operating system behavior. For example, a notification rendered on top of other software (e.g., to inform the user of an email, calendar events, or system update) can detract from the audience and the performer's immersion in the performance. In fact, in a previous Live Writing performance, the whole audience experienced the tension of a performance failure when the OS displayed a low battery notification, requiring the performer to stop typing and click the "Close" button [8]. While what to reveal to the audience depends on stakeholders' preferences, it is generally better to plan, rehearse, and control what will be shown on the screen in advance with the stage crew.

4.2.2 Responsive Design and Responsive Rehearsal

For a musician who intends to mirror their screen in a performance space, preparations must be made to accommodate the variety of possible screen resolutions of the projector used. Connecting a projector to a computer or a laptop may force a particular screen resolution, potentially one smaller than that of the development environment — this is typical of smaller and older projectors. If this is not discovered until immediately before the performance, this can cause the performer to panic. This is particularly problematic for a web-based application, in which a third-party library (such as Bootstrap[9]) may respond to a lower-resolution screen differently. In general, it is recommended to know the supported screen resolutions prior to the performance and to test the application using a secondary screen or projector. In addition, it is useful to consider the significance of readability of the information shown on the projection, if it matters, in relation to the context and the style of music, screen resolution, visual content, performance space configuration, and the size of the projection.

5. SHOW THEM MY SCREEN

In this paper, I have reviewed previous works that incorporate mirroring a performer's interface running on a computer in the performance space for audience communication. My personal trajectory of starting a computer music practice from live coding influenced me to consider it natural to reveal my computer screen to the audience. Arguably, revealing a computer screen to the audience can compel artists to perceive this as an opportunity of make the visual performativity stronger. This principle I tried to adhere to has compelled me to develop novel performance environment, as opposed to settling for one meta music-making environment (e.g. MAX/MSP). Subsequently, I have utilized the rich expressivity and communication enabled by sharing screens to facilitate various types of novel audiovisual performance. These performances incorporate audience participation, music generated from visuals, and gesture-based interfaces which control virtual objects on the screen. I wish that this reflection could contribute to nurturing the culture of open-screen performances and transparent design in NIME. Eventually, when this idea of enhancing audience communication in live performances becomes prominent, audiences can approach computer music performances with a common ground of communicative, compelling, and engaging visuals on the screen.

6. REFERENCES

[1] F. Bérard, J. Ip, M. Benovoy, D. El-Shimy, J. R. Blum, and J. R. Cooperstock. Did "minority report" get it wrong? superiority of the mouse over 3d input devices in a 3d placement task. In *IFIP Conference on Human-Computer Interaction*, pages 400–414. Springer, 2009.

[2] F. Berthaut, H. Katayose, H. Wakama, N. Totani, and Y. Sato. First person shooters as collaborative multiprocess instruments. In *Proceedings of the International Conference on New Interfaces for Musical Expression*, pages 44–47, Oslo, Norway, 2011.

[3] F. Berthaut, D. Martinez, M. Hachet, and S. Subramanian. Reflets: Combining and revealing spaces for musical performances. In E. Berdahl and J. Allison, editors, *Proceedings of the International Conference on New Interfaces for Musical Expression*, pages 116–120, Baton Rouge, Louisiana, USA, May 2015. Louisiana State University.

[4] F. Berthaut, V. Zappi, and D. Mazzanti. Scenography of immersive virtual musical instruments. In *VR Workshop: Sonic Interaction in Virtual Environments (SIVE), 2014 IEEE*, Minneapolis, United States, Mar. 2014.

[5] O. Bown, R. Bell, and A. Parkinson. Examining the perception of liveness and activity in laptop music: Listeners' inference about what the performer is doing from the audio alone. In *Proceedings of the International Conference on New Interfaces for Musical Expression*, pages 13–18, London, United Kingdom, 2014. Goldsmiths, University of London.

[6] M. Cerqueira, S. Salazar, and G. Wang. Soundcraft: Transducing starcraft 2. In *Proceedings of the International Conference on New Interfaces for Musical Expression*, pages 243–247, Daejeon, Republic of Korea, May 2013. Graduate School of Culture Technology, KAIST.

[7] N. Collins. Generative music and laptop performance. *Contemporary Music Review*, 22(4):67–79, 2003.

[8] N. Collins and A. McLean. Algorave: A survey of the history, aesthetics and technology of live performance of algorithmic electronic dance music. In *Proceedings of the International Conference on New Interfaces for Musical Expression*, pages 355–358, London, United Kingdom, 2014. Goldsmiths, University of London.

[9] N. Collins, A. McLean, J. Rohrhuber, and A. Ward. Live coding in laptop performance. *Organised Sound*,

[8] https://youtu.be/hAHxM8a-0UI?t=543

[9] https://getbootstrap.com/

8(03):321–330, 2003.

[10] A. D. de Carvalho Junior, S. W. Lee, and G. Essl. Understanding cloud support for the audience participation concert performance of crowd in c[loud]. In *Proceedings of the International Conference on New Interfaces for Musical Expression*, volume 16 of *2220-4806*, pages 176–181, Brisbane, Australia, 2016. Queensland Conservatorium Griffith University.

[11] D. Della Casa and G. John. Livecodelab 2.0 and its language livecodelang. In *Proceedings of the 2nd ACM SIGPLAN international workshop on Functional art, music, modeling & design*, pages 1–8. ACM, 2014.

[12] T. Edwards and R. B. Sutherland. Eyes off the screen! techniques for restoring visual freedom in leo performance. In *Proceedings of the 1st Symposium on Laptop Ensembles & Orchestras (SLEO)*, pages 33–40, 2012.

[13] R. Fiebrink, G. Wang, and P. R. Cook. Don't forget the laptop: Using native input capabilities for expressive musical control. In *Proceedings of the 7th International Conference on New Interfaces for Musical Expression*, NIME '07, pages 164–167, New York, NY, USA, 2007. ACM.

[14] J. Freeman and A. Troyer. Collaborative textual improvisation in a laptop ensemble. *Computer Music Journal*, 35(2):8–21, 2011.

[15] R. Hamilton. Building interactive networked musical environments using q3osc. In *Audio Engineering Society Conference: 35th International Conference: Audio for Games*. Audio Engineering Society, 2009.

[16] R. Hamilton. Sonifying game-space choreographies with udkosc. In *Proceedings of the International Conference on New Interfaces for Musical Expression*, pages 446–449, Daejeon, Republic of Korea, May 2013. Graduate School of Culture Technology, KAIST.

[17] S. Lawson and R. R. Smith. What am i looking at?: An approach to describing the projected image in live-coding performance. In *Proceedings of International Conference on Live Coding*, Medialab Prado, Madrid, Spain, 2019.

[18] S. W. Lee, J. Bang, and G. Essl. Live coding youtube: Organizing streaming media for an audiovisual performance. In *Proceedings of the International Conference on New Interfaces for Musical Expression*, pages 261–266, Copenhagen, Denmark, 2017. Aalborg University Copenhagen.

[19] S. W. Lee, A. D. de Carvalho Jr, and G. Essl. Crowd in c[loud]: Audience participation music with online dating metaphor using cloud service. 2016.

[20] S. W. Lee and G. Essl. Live coding the mobile music instrument. *Ann Arbor*, 1001:48109–2121, 2013.

[21] S. W. Lee and G. Essl. Web-based temporal typography for musical expression and performance. In E. Berdahl and J. Allison, editors, *Proceedings of the International Conference on New Interfaces for Musical Expression*, pages 65–69, Baton Rouge, Louisiana, USA, May 31 – June 3 2015. Louisiana State University.

[22] S. W. Lee, G. Essl, and M. Martinez. Live writing : Writing as a real-time audiovisual performance. In *Proceedings of the International Conference on New Interfaces for Musical Expression*, Brisbane, Australia, 2016.

[23] S. W. Lee and J. Freeman. echobo: A mobile music instrument designed for audience to play. *Ann Arbor*, 1001:48109–2121.

[24] S. W. Lee and J. Freeman. Real-time music notation in mixed laptop acoustic ensembles. *Computer Music Journal*, 37(4):24–36, Dec 2013.

[25] T. Magnusson. ixi lang: a supercollider parasite for live coding. In *Proceedings of International Computer Music Conference 2011*, pages 503–506. Michigan Publishing, 2011.

[26] T. Magnusson. The threnoscope: a musical work for live coding performance. In *LIVE Workshop, International Conference of Software Engineering*, 2013.

[27] A. McLean, D. Griffiths, N. Collins, and G. Wiggins. Visualisation of live code. In *Proceedings of the 2010 International Conference on Electronic Visualisation and the Arts*, EVA'10, pages 26–30, Swindon, UK, 2010. BCS Learning & Development Ltd.

[28] A. McLean, D. Griffiths, N. Collins, and G. Wiggins. Visualisation of live code. *Proceedings of Electronic Visualisation and the Arts 2010*, 2010.

[29] A. McLean and G. A. Wiggins. Texture: Visual notation for live coding of pattern. In *ICMC*, 2011.

[30] A. Meacham, S. Kannan, and G. Wang. The laptop accordion. In *Proceedings of the International Conference on New Interfaces for Musical Expression*, volume 16 of *2220-4806*, pages 236–240, Brisbane, Australia, 2016. Queensland Conservatorium Griffith University.

[31] C. Roberts. Code as information and code as spectacle. *International Journal of Performance Arts and Digital Media*, 12(2):201–206, 2016.

[32] S. Salazar, A. Piepenbrink, and S. Reid. Developing a performance practice for mobile music technology. In T. M. Luke Dahl, Douglas Bowman, editor, *Proceedings of the International Conference on New Interfaces for Musical Expression*, pages 59–64, Blacksburg, Virginia, USA, June 2018. Virginia Tech.

[33] W. A. Schloss. Using contemporary technology in live performance: The dilemma of the performer. *Journal of New Music Research*, 32(3):239–242, 2003.

[34] S. Şentürk, S. W. Lee, A. Sastry, A. Daruwalla, and G. Weinberg. Crossole: A gestural interface for composition, improvisation and performance using kinect. In *Proceedings of the International Conference on New Interfaces for Musical Expression*, Ann Arbor, Michigan, 2012. University of Michigan.

[35] D. A. Stewart and S. W. L. Lee. Live writing: Jimmy raps, 2017. Music Performance, the International Conference on New Interfaces for Musical Expression (NIME).

[36] C. Stuart. The object of performance: Aural performativity in contemporary laptop music. *Contemporary Music Review*, 22(4):59–65, 2003.

[37] D. Trueman. Why a laptop orchestra? *Organised Sound*, 12(2):171–179, 2007.

[38] N. Weitzner, J. Freeman, Y.-L. Chen, and S. Garrett. massmobile: towards a flexible framework for large-scale participatory collaborations in live performances. *Organised Sound*, 18(01):30–42, 2013.

[39] A. D. Wilson. Touchlight: an imaging touch screen and display for gesture-based interaction. In *Proceedings of the 6th international conference on Multimodal interfaces*, pages 69–76. ACM, 2004.

[40] S. Wilson, N. Lorway, R. Coull, K. Vasilakos, and T. Moyers. Free as in beer: Some explorations into structured improvisation using networked live-coding systems. *Computer Music Journal*, 38(1):54–64, 2014.