Visualising a Live Coding Arts Process

Anon1
-
-
-

Anon2
-
-
-

Anon3
-
-
-

ABSTRACT

This paper describes an empirical study of source code visualisation as a means to communicate the programming process in "live coding" computer music performances. Following an exploratory field study conducted during a live coding performance at an arts festival, two different interaction-driven visualisation techniques were incorporated into a live coding system. We then performed a more controlled lab study to evaluate the visualisations' contributions to the audience experience, with emphasis on the (self-reported) experiential dimensions of *understanding* and *enjoyment*. Both software visualisation techniques enhanced audience enjoyment, while the effect on audience understanding was more complex. We conclude by suggesting how these visualisation techniques may be used to enhance the audience experience of live coding.

Categories and Subject Descriptors

H.5 [**Information Interfaces and Presentation**]: Miscellaneous

General Terms

Experimentation, Design

Keywords

Software visualisation, live coding, musical performance

1. INTRODUCTION

"Show us your screens... Code should be seen as well as heard", declares the draft manifesto of "TOPLAP" [8], an international organisation devoted to the artistic performance practice of "live coding". In live coding, computer code is written in front of a live audience to generate music and visuals in real time. The "show us your screens" rhetoric underscores the need for authenticity to distinguish this artform from similar (but non-live) computational arts practices.

OZCHI '14, Dec 2-5, 2014, Sydney, Australia

But what is the benefit of the live coder showing their screen? In a live coding performance, non-expert live coding audience members spend much of their time staring at raw (usually text-based) computer code. Until now, little empirical study has been undertaken to gauge an audience's understanding of that computer code and whether, from an audience perspective, code really should be "seen as well as heard".

Traditional approaches to source code visualisation (see [6] for a review) often focus on structure of the source code (e.g. visualising complex object/class relationships) rather than the *process* of programming. In a process-oriented activity such as live coding, different code visualisation techniques are necessary [4, 3]. However, these academic treatments of code visualisation in live coding adopt a survey-based approach, and the techniques discussed have not been subject to empirical evaluation.

In this paper, we examine the audience's experience of the displayed code during live coding performances and to see whether code-driven visualisations might improve both the audience enjoyment and the audience understanding of these performances. This exploration takes place initially through the results of an exploratory field study at a contemporary arts festival, and subsequently through a lab-based follow-up user study.

2. EXPLORATORY FIELD STUDY

After a live coding performance at the [name withheld for blind peer-review] arts festival in [place and date withheld], audience members were asked to fill out a survey regarding their perception of and response to the projection of the computer code during the performance. Each audience member was asked to indicate which of a number of curves/trajectories best represented their *enjoyment* and *understanding* of the performer's actions in typing the code through the performance. These trajectories allowed for "high", "medium", and "low" levels of enjoyment/understanding for the (self-determined) "beginning", "middle" and "end" of the performance. Other survey questions addressed their sense of "liveness" of the performance (c.f. [1]) and whether the projected code was confusing.

2.1 Field Study Results

Of the thirteen survey responses received, six audience members showed a high level of enjoyment throughout the whole performance, while the remaining seven responses showed alternating levels of enjoyment. No audience members indicated a low level of enjoyment throughout the performance.

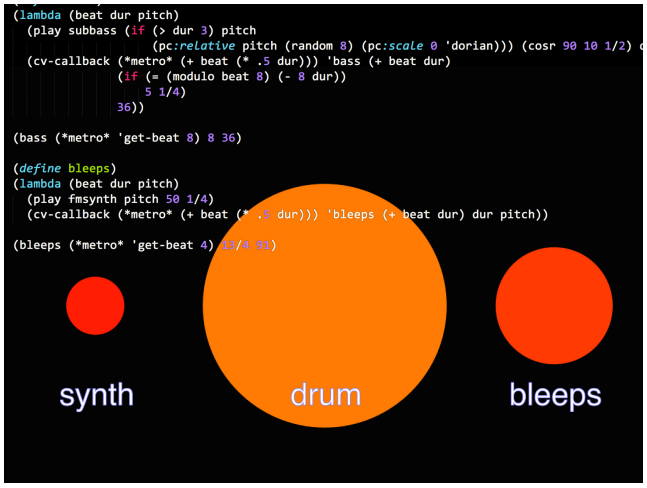

Figure 1: An example didactic visualisation (all figures best viewed in colour).

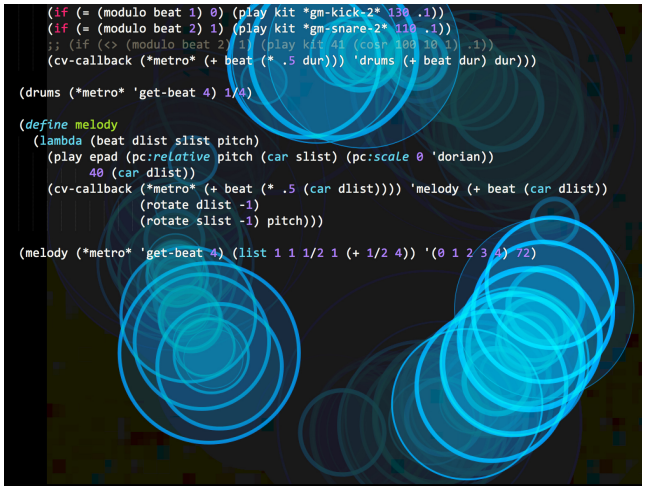

Figure 2: An example aesthetic visualisation.

Only two of the thirteen respondents indicated that they understood the relationship between the code projections and the music throughout the performance. Three of the six respondents who reported a high level of enjoyment throughout the performance also indicated an increase in understanding (from low to high) as the performance progressed, although a Chi-square analysis revealed no significant relationship between enjoyment and understanding due to the small sample size. Nine of the thirteen respondents stated that the code projection provided a sense of liveness to the performance and the remainder stated that viewing the code had no effect on their sense of liveness. Four respondents felt that the code projections were confusing, five felt that they were not confusing, and four did not answer the question.

Taken as a whole, the results of this small field study were salutatory towards the benefit of "seeing as well as hearing" code during a live coding performance, especially as far as the general public is concerned. The majority of the audience felt that they made the performance seem more "live". However, a minority stated that they found the projections confusing and only a very small number of respondents claimed to have actually understood what the programmer was doing. We were quite intrigued by the small cohort of respondents whose understanding increased through the performance and whose enjoyment remained high, and we wished to test whether augmenting code projections with additional visualisations might increase the understanding and enjoyment of the audience in live coding.

3. LAB STUDY

A second lab study was conducted to test the impact of additional visual feedback (beyond the raw source code) on audience understanding and enjoyment in live coding. Music visualisation is an extremely rich and open-ended task, so to guide the development of the visualisations for our lab study, we used the concepts of understanding and enjoyment from the initial survey to develop two new code visualisations: a *didactic* one and an *aesthetic* one.

The didactic visualisation (shown in Figure 1) attempted to communicate *information* about the actions of the programmer, prominently displaying the *names* of the active (source code) functions and the "time until next execution" for each function (which is particularly relevant in a time-sensitive programming context such as music making). Bright colours and solid shapes were used to ensure constant visibility and to communicate the intention of the underlying code. The didactic visualisations proceeded through four stages, with phase changes made depending on the number of active functions (instruments).

The aesthetic visualisation technique, on the other hand, was designed to react to the programmer's activity in a more abstract way, to maximise aesthetic appeal [2] and to engage the audience's interest. Although still based on the source code and the livecoder's edits, the generation of shapes was driven by instrument volume and synchronised with the musical beat. The emphasis for the aesthetic visualisation was on the artistic appeal of the visuals (see Figure 2), including more variety in visual structure and colour. As in the didactic condition, the aesthetic visualisations proceeded through four stages, based on the number of active functions (instruments), but these visuals had no textual labels and they moved and interacted with each other over the entire projected scene.

Our hypothesis was the didactic visualisation approach would result in enhanced audience understanding, and a reduction in audience confusion through the performance. In contrast, we predicted that the aesthetic visualisations would positively influence audience enjoyment, both overall and over the course of the performance.

3.1 Lab Study Experimental Design

To assess the impact of these two visualisation techniques on audience understanding and enjoyment, we conducted a lab study. Two independent audiences ($N = 19 + 22 = 41$) recruited through an on-campus advertisement each watched a live coder perform two ten-minute "sets": one accompanied by the didactic visuals, and one with the aesthetic. The order of presentation of the two visual conditions was swapped between the groups. The improvisational nature of a live coding performance makes "controlled" experiments difficult, but the live coding artist attempted (as much as possible) to do the same two performances for each group.

Over the course of these performances, each audience mem-

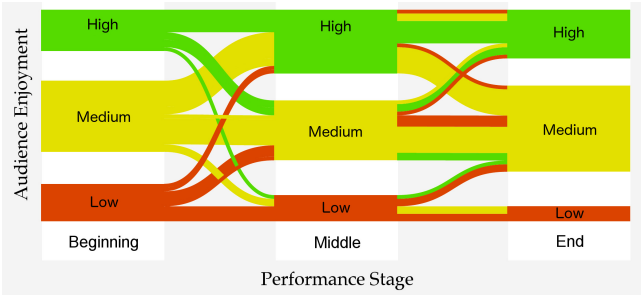

Figure 3: Audience reported enjoyment during the beginning, middle and end of the performance for the didactic condition.

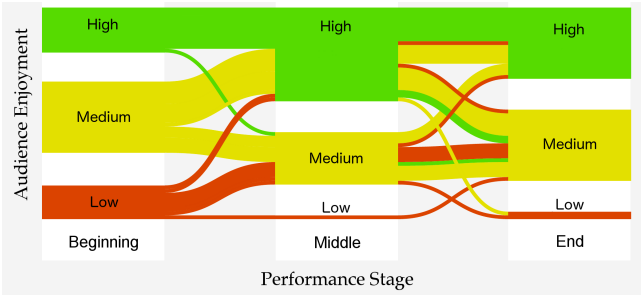

Figure 4: Audience-reported enjoyment level during the beginning, middle and end of the performance for the aesthetic condition. Line width at each stage indicates proportion of the audience reporting high, medium or low enjoyment, and line colour is determined by the enjoyment level at the *beginning* of the performance.

ber completed a survey consisting of four sections: demographic information, their opinion of the first piece, their opinion of the second piece and questions about the performance overall. Similar to the first field trial, the questionnaire primarily focussed on self-reported levels of "enjoyment" and "understanding" related to the visualisations specifically and also to the performance more generally. There was also a free-form question for suggested improvements to the visualisations.

After the lab study performance, a video-cued-recall [7] interview was conducted with the live coder using a video of the performance.

3.2 Lab Study Results

Of the 41 audience participants 66% were male, 76% were aged between 18 and 32 and 78% of the participants had never seen a live coding performance before.

The audience-reported enjoyment and understanding responses from the questionnaire were evaluated for the two visualisation conditions as described below. A significance level of 0.05 was used for the Chi-squared analysis.

3.2.1 *Enjoyment*

Overall, the majority of the participants reported that both visualisation conditions had a positive effect on their **enjoyment** of the performance: 76% stated that the aesthetic visualisations improved their enjoyment and 56% stated

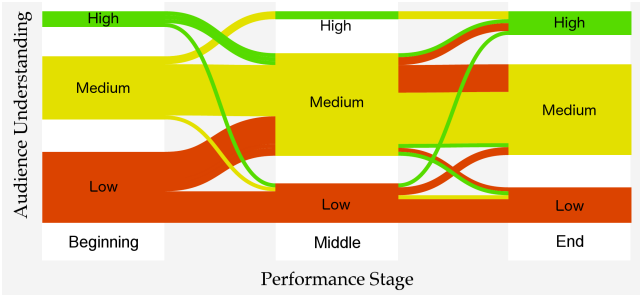

Figure 5: Audience reported understanding during the beginning, middle and end of the performance for the didactic condition.

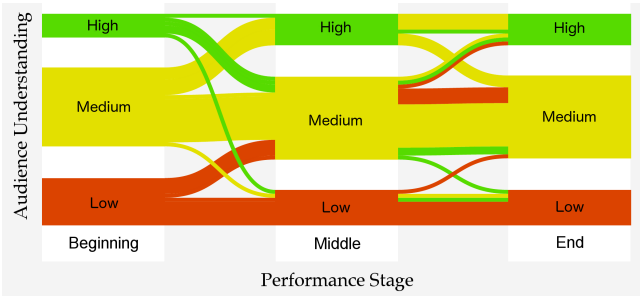

Figure 6: Audience reported understanding during the beginning, middle and end of the performance for the aesthetic condition.

that the didactic visualisations improved their enjoyment. No significant difference between the visualisation types regarding enjoyment was found ($\chi^2 = 3.7733, df = 2, p = 0.1516$).

Participants were asked to rate their enjoyment during the (self-determined) "beginning", "middle" and "end" of the performances (see Figure 4 and Figure 3). During the didactic performance, 15% of the audience stated that their enjoyment *increased* from the beginning of the performance and was steady thereafter. By contrast, 24% of the audience reported this pattern of enjoyment during the aesthetic performance. Approximately 30% of the audience of all (aesthetic and didactic) performances stated that their enjoyment remained steady throughout.

3.2.2 *Understanding*

In response to a specific survey question, 37% of participants stated that overall, the didactic visualisations "helped them to **understand** the code", compared to 12% of participants for the aesthetic visualisations. This was a significant difference between the visualisation conditions ($\chi^2 = 7.1986, df = 2, p = 0.02734$).

Again, participants were asked to rate their understanding during the (self-reported) "beginning", "middle" and "end" of the performance (see Figure 6 and Figure 5). During the didactic and aesthetic performances, 49% and 44% respectively of the participants stated that their understanding *remained the same* throughout the performances. During the didactic performance, 10% of the audience reported a level of understanding that *trended downwards* (eg. high to low) compared to 20% of the audience during the aesthetic performance. However, this reported advantage of

the didactic visualisations was offset by the reported audience understanding at the beginning of the performance where 44% indicated a low understanding with the didactic visualisations compared to only 30% with the aesthetic visualisations.

Overall, the questionnaire results for audience understanding are complex, and reported levels of understanding fluctuated during the performances. Dramatically, Figure 5 shows that a very small proportion of the audience reported high understanding during the middle of the performances. One interpretation of this result might be that it took audience members some time to work out what the didactic visualisations were actually showing, and that this conflicted with the first impressions of what some audience members (hence the decrease in levels of understanding from beginning to middle). However, once they finally understood the graphics some audience members were then able to better understand the live-coding performance.

3.3 Discussion

The overall enjoyment of the visualisations was high, for both the aesthetic and didactic visualisations. Reported enjoyment of the aesthetic visualisations was higher than for the didactic visualisations but the trends across Figures 4 and 3 are complex.

As discussed above, the small number of high responses for understanding during the middle of the didactic performances, and the decreasing trend from high to middle level understanding from beginning to middle of the performances perhaps indicates a higher cognitive load for understanding the didactic visualisations themselves. In fact, features of the didactic visualisation were reported to confuse some members of the audience, despite their stated aim of *assisting* audience understanding. One audience member even stated that they "found them distracting" and that they "preferred just to read the code".

The video-cued-recall interview indicated that the experience of the visualisations of the live coder and the audience was fundamentally different. While many members of the audience reported that they drifted between focussing on the music, focussing on the visualisations and focussing on the code, the live coder reported that their focus was purely on the code and the music, rarely drifting. In one particular section of the interview, the live coder stated: "I definitely wasn't paying attention to them [the visualisations] on the day. In fact I tune them out as best I can because I am just trying to focus on the code". By contrast, one audience member stated that "you could see the code being written and the visualisations helped to show when a piece of code started working". Another audience member stated that "the visualisations were interesting but distracting". When asked if the visualisations were distracting the live coder stated: "Ah, no. In general I'm just so focussed on the code".

4. CONCLUSION

In this first empirical study of audience perception of code visualisation in live coding, we have identified an opportunity for real-time code visualisations to help improve the audience experience of a live coding computer music performance. With few exceptions, our initial survey of a live coding performance at an arts festival revealed a generally low to medium level of audience understanding throughout that performance (although almost half the survey respondents indicated a high level of enjoyment throughout).

In a subsequent lab study, a comparison of two prototype code visualisations indicated that both visualisations seemed to help with enjoyment. Significantly more audience members reported that our didactic visualisations helped with understanding but overall trends for both enjoyment and understanding throughout the performances were complex. There are indications of a higher cognitive load for the didactic visualisations than the aesthetic visualisations and this may have influenced audience responses to them.

In a future extension of this work, design lessons from both visualisation types could be combined together to produce live coding driven visualisations which targeted both the aesthetics as well as a greater understanding of the live coding process. These visualisations could then be compared with the baseline "no visualisation" condition in an audience experiment. There are also opportunities to vary the nature of the visualisations over the course of a performance.

Over 60 years ago, the media theorist Marshall McLuhan stated that "The business of art is no longer the communication of thoughts or feelings which are to be conceptually ordered, but a direct participation in an experience. The whole tendency of modern communication... is towards participation in a process, rather than apprehension of concepts." [5] Our hope is that future developments in visualisations for live coding may bring audiences further into the *process* of a highly-skilled live coding artist.

5. REFERENCES

[1] P. Auslander. *Liveness: Performance in a Mediatized Culture*. Routledge, second edition, 2008.

[2] N. Cawthon and A. V. Moere. The Effect of Aesthetic on the Usability of Data Visualization. *2007 11th International Conference Information Visualization (IV '07)*, pages 637–648, July 2007.

[3] T. Magnusson. Algorithms as Scores: Coding Live Music. *Leonardo Music Journal*, 21:19–23, 2011.

[4] A. Mclean, D. Griffiths, and N. Collins. Visualisation of live code. *Visualisation and the Arts*, pages 1–5, 2010.

[5] M. McLuhan. Letter to Harold Adam Innis, March 14 1951. In E. McLuhan and F. Zingrone, editors, *Essential McLuhan*, page 73.

[6] R. L. Novais, A. Torres, T. S. Mendes, M. Mendonça, and N. Zazworka. Software evolution visualization: A systematic mapping study. *Information and Software Technology*, 55(11):1860–1883, Nov. 2013.

[7] L. A. Suchman and R. H. Trigg. Understanding practice: video as a medium for reflection and design. pages 65–90, 1992.

[8] Toplap. Toplap Manifesto. http://toplap.org/wiki/ManifestoDraft, 2010.