Project Plan

28th March 2014

Arrian Purcell

Description

Currently missing within the research surrounding live coding is a visualisation of the code that represents the artists intent. Previous visualisation techniques present an abstract and often disjoint representation from the associated code. Missing within this context is a formal analysis of how to best represent the artist's intent visually and a move towards formal analysis of the target audience.

Missing within the literature is an elaboration of dynamic software visualisations and a sensible taxonomy of music visualisations. This study would aim to develop a taxonomy of music and software visualisations that could be used as a basis for further analysis and research.

The study will approach the examination of music and software visualisations from two angles. These would include examining live coding from the audience's perspective to gain insight into how to design visualisations to support their existing mental model and examining live coding from the artist's perspective, gaining insight into what the artist wants to convey and the best method to approach this. This will be investigated through systematic experimentation and analysis of qualitative and quantitative data.

Plan

The project structure would consist of:

1. A field study surveying the existing mental model of a live coding audience (completed)
2. A study investigating a variety of live coding visualisation techniques
3. Analysis and refinement of these visualisation techniques
4. A study investigating the refined visualisation techniques

See the following page for the project timeline.

Date	Description
6 April	Begin preparing the first stage visualisations. The design and implementation of the first stage of visualisations will begin. By this stage a substantial amount of literature will be analysed and first stage visualisation mockups will be prepared. Furthermore, these visualisations will take advantage of the data analysis from the initial survey and will tak advantage of the developed taxonomy.
15 April	Preparation for the first stage experimental lab study will begin. This will take into account the progress with the visualisations and will include organising the lab study location, time, ethics etc.
2 May	First stage visualisations prepared. Three or four visualisations will be completed, ready for use within the experimental study.
12 May	First experimental lab study. On or around this date the first stage visualisations will be examined in an experimental 'lab' study. This study will likely take place during a CSSA event.
13 May	Analysis of the data from the first experiment will conducted. This will go on to inform the second stage of visualisations.
16 May	Begin preparing the second stage visualisations. The design and implementation of the second stage of visualisations will begin. This will build on the first stage visualisations and will take into account the analysis of the data from the first lab experiment.
30 May	Preparation for the second stage experimental lab study will begin. Again, this will include organising the lab study location, time, ethics etc.
16 June	Mid-term seminar. A 20 minute presentation will describe research progress to date and a description of the topic in detail.
23 June	Second stage visualisations prepared. Refinement of the first stage of visualisations will be completed.
4 July	Second lab study. On or around this date the second stage visualisations will be examined in an experiment very similar to the first study.
1 August	Begin preparing the CHI paper. This will summarise the more interesting research conducted.
3 October	Draft thesis to be presented to Henry for review. This will be a very close to final thesis.
13 October	Final seminar presenting the results of the research project over 25 minutes.
25 October	Thesis submission deadline at 3pm.

Reading List

28th March 2014

Arrian Purcell

Aaron, S., Blackwell, A. F., Hoadley, R. and Regan, T. 2011. A principled approach to developing new languages for live coding. Nime, (June), pp. 381-386.

Amiri, F. 2011. Programming as design: The role of programming in interactive media curriculum in art and design. International Journal of Art & Design Education, 30 (2), pp. 200-210.

Auslander, P. 2005. At the Listening Post, or, do machines perform?. International Journal of Performance Arts and Digital Media, 1 (1), pp. 5-10.

Baecker, R. 1996. The Early History of Software Visualization.

Bennedsen, J. and Caspersen, M. 2008. Exposing the programming process. Reflections on the Teaching of Programming, pp. 6-16.

Bernsen, N. 1997. Defining a taxonomy of output modalities from an HCI perspective. Computer Standards & Interfaces, 18 pp. 537-553.

Blackwell, A. and Collins, N. 2005. The programming language as a musical instrument. Proceedings of PPIG05 (Psychology of Programming Interest Group), pp. 1-11.

Boyer, K. E., Dwight, A. A., Fondren, R. T., Vouk, M. A. and Lester, J. C. 2008. A development environment for distributed synchronous collaborative programming. ACM SIGCSE Bulletin, 40 (3), p. 158.

Brown, A. and Sorensen, A. 2007. aa-cell in practice: An approach to musical live coding. Proceedings of the International Computer Music Conference.

Brown, A. R. and Sorensen, A. 2009. Interacting with Generative Music through Live Coding. Contemporary Music Review, 28 (1), pp. 17-29.

Cameron, D. and Carroll, J. 2011. Encoding Liveness: Performance and Real-Time Rendering in Machinima. The Machinima Reader.

Card, S. and Mackinlay, J. 1997. The structure of the information visualization design space. Information Visualization, 1997.

Chen, C., Weng, M., Jeng, S. and Chuang, Y. 2008. Emotion-based music visualization using photos. Advances in Multimedia, 4903 pp. 358-368.

Chi, E. 2000. A taxonomy of visualization techniques using the data state reference model. Information Visualization, 2000. InfoVis 2000. IEEE, 94301 (Table 2).

Collins, N. 2011. Live Coding of Consequence. Leonardo, 44 (3), pp. 207-211.

Collins, N. and Mclean, A. 2003. Live coding in laptop performance. Organised Sound, 8 (3), pp. 321-329.

Correia, N. 2012. AVVX - a vector graphics tool for audiovisual performances. Leonardo Electronic Almanac, 19 (3).

Desmond, M., Storey, M. and Exton, C. 2006. Fluid source code views for just in-time comprehension. Workshop on Software Engineering, pp. 1--4.

Dewitt, T. 1987. Visual Music: Searching for an Aesthetic. Leonardo, 20 (2), p. 115.

Dulic, A. and Hamel, K. 2009. Visual music instrument. International Journal of Arts and Technology.

Dunker, P., Nowak, S., Begau, A. and Lanz, C. 2008. Content-based mood classification for photos and music: a generic multi-modal classification framework and evaluation approach. Proceedings of the 1st ACM, pp. 97-104.

Ferguson, S. and Cabrera, D. 2005. Seeing Sound: Real-Time Sound Visualisation in Visual Feedback Loops Used for Training Musicians. Ninth International Conference on Information Visualisation (IV'05), pp. 97-102.

Foote, J. 1999. Visualizing music and audio using self-similarity. Proceedings of the seventh ACM international conference on Multimedia (Part 1) - MULTIMEDIA '99, pp. 77-80.

Foote, J. and Cooper, M. 2001. Visualizing musical structure and rhythm via self-similarity. Proceedings of the 2001 International Conference on Computer Music.

Freeman, T. Outsourcing the VJ. Leonardo Electronic Almanac, 19 (3), pp. 124-133.

Fronk, A., Gude, D. and Rinkenauer, G. 2006. Evaluating 3D-visualisation of code structures in the context of reverse engineering.

Gerber, A. and Brown, A. 2006. Visualising Music with Impromptu. pp. 38-43.

Gravino, C., Tortora, G. and Scanniello, G. 2010. An empirical investigation on the relation between analysis models and source code comprehension. Proceedings of the 2010 ACM Symposium on Applied Computing, pp. 2365-2366.

Green, M. 1998. Toward a perceptual science of multidimensional data visualization: Bertin and beyond. ERGO/GERO Human Factors Science.

Gulwani, S. 2010. Dimensions in program synthesis. Proceedings of the 12th international ACM SIGPLAN.

Heer, J. and Agrawala, M. 2006. Software design patterns for information visualization. IEEE transactions on visualization and computer graphics, 12 (5), pp. 853-60.

Hendrix, D., Cross, J. I. and Maghsoodloo, S. 2002. The effectiveness of control structure diagrams in source code comprehension activities. Software Engineering, IEEE Transactions on, 28 (5), pp. 463-477.

Hockley, N., Bahlmann, F. and Chasin, M. 2010. Programming hearing instruments to make live music more enjoyable. The Hearing Journal, 63 (9).

Huber, D. and Healey, C. Visualizing Data with Motion. IEEE Visualization 2005 - (VIS'05), pp. 67-67.

Hundhausen, C. D. and Brown, J. L. 2007. What You See Is What You Code: A “live” algorithm development and visualization environment for novice learners. Journal of Visual Languages & Computing, 18 (1), pp. 22-47.

Hyrskykari, A. 1993. Development of Program Visualization Systems. Symposium of Visual Aspects of Man-Machine Systems.

Kelemen, C. and Turk, E. Active Visual Scaffolding. pdf.aminer.org, pp. 1-12.

Kot, B. 2005. Information Visualisation utilising 3D Computer Game Engines.

Levin, G. 2000. Painterly interfaces for audiovisual performance.

Magnusson, T. 2011. Algorithms as scores: Coding live music. Leonardo Music Journal, 21 pp. 19-23.

Mailman, J. Improving Synesthesia. Leonardo Electronic Almanac, 19 (3).

Mankoff, J., Dey, A. K., Hsieh, G., Kientz, J., Lederer, S. and Ames, M. 2003. Heuristic evaluation of ambient displays. Proceedings of the conference on Human factors in computing systems - CHI '03, (5), p. 169.

Manovich, L. The Language of New Media.

Mazzola, G., Walsh, D., Butler, L. and Polukeyev, A. Visualization Technologies for Music , Dance, and Staging in Operas. Leonardo Electronic Almanac, 19 (3), pp. 272-283.

Mccarthy, L. Gathering Audience Feedback on an Audiovisual Performance. Leonardo Electronic Almanac, 19 (3).

Mclean, A. and Wiggins, G. 2010. Live coding towards computational creativity. Conference on Computational Creativity.

Mclean, A. and Wiggins, G. 2010. Tidal—pattern language for the live coding of music. Proceedings of the 7th Sound and Music Computing Conference.

Mclean, A., Griffiths, D., Collins, N. and Wiggins, G. 2010. Visualisation of live code. Visualisation and the Arts, pp. 1-5.

Mclean, C. 2011. Artist-programmers and Programming Languages for the Arts.

Milgram, P. and Kishino, F. 1994. A taxonomy of mixed reality visual displays. IEICE TRANSACTIONS on Information and Systems, 77 (12), pp. 1321-1329.

Mitroo, J., Herman, N. and Badler, N. 1979. Movies from music: Visualizing musical compositions. ACM SIGGRAPH Computer Graphics.

Moore, A. 2002. Authenticity as authentication. Popular Music, 21 (02), pp. 209-223.

Muller, H. A., Wong, K. and Tilley, S. R. Dimensions of Software Architecture for Program Understanding.

Murphy-Hill, E. and Black, A. 2010. An interactive ambient visualization for code smells. International symposium on Software visualization.

Myers, B. 1986. Visual programming, programming by example, and program visualization: a taxonomy. ACM SIGCHI Bulletin, (April), pp. 59-66.

Myers, B. 1990. Taxonomies of Visual Programming and Program Visualization. Journal of Visual Languages & Computing.

Nilson, C. 2007. Live coding practice. Proceedings of the 7th international conference on new interfaces for musical expression, p. 112.

Nosek, J. 1998. The case for collaborative programming. Communications of the ACM, 41 (3), pp. 105-108.

Oda, T., Nakakoji, K. and Yamamoto, Y. 2013. Somethingit: A prototyping library for live and sound improvisation. Live Programming (LIVE), pp. 13-16.

Paxton, J. 2002. Live programming as a lecture technique. Journal of Computing Sciences in Colleges, pp. 51-56.

Phelan, P. 2003. Performance, Live Culture and Things of the Heart. Journal of Visual Culture, 2 (3), pp. 291-302.

Price, B., Baecker, R. and Small, I. 1993. A principled taxonomy of software visualization. Journal of Visual Languages & Computing.

Price, B., Small, I. and Baecker, R. 1992. A taxonomy of software visualization. System Sciences, 1992.

Rajlich, V. and Wilde, N. 2002. The role of concepts in program comprehension. Program Comprehension, 2002. \ldots, pp. 271-278.

Reason, M. 2006. Young audiences and live theatre, Part 2: Perceptions of liveness in performance. Studies in Theatre and Performance, 26 (3), pp. 221-241.

Robillard, M. Representing concerns in source code.

Roman, G. and Cox, K. 1993. A taxonomy of program visualization systems. Computer, 26 (12).

Ruthmann, A., Heines, J. and Greher, G. 2010. Teaching computational thinking through musical live coding in scratch. Proceedings of the 41st, pp. 351-355.

Sa, A. How An Audio-Visual Instrument Can Foster The Sonic Experience. Leonardo Electronic Almanac, 19 (3).

Sakellion, Y. and Da, Y. Structured Spontaneity. Leonardo Electronic Almanac, 19 (3).

Smith, S. and Williams, G. 1997. A Visualization of Music. Visualization'97, Proceedings, pp. 499-503.

Stowell, D. and Mclean, A. 2013. Music and Human-Computer Interaction. Springer London, pp. 139-152.

Swift, B., Sorensen, A., Gardner, H. and Hosking, J. 2013. Visual code annotations for cyberphysical programming. Live Programming, pp. 27-30.

Torrens, M., Hertzog, P. and Arcos, J. 2004. Visualizing and Exploring Personal Music Libraries. ISMIR.

Tory, M. and Moller, T. 2002. A Model-Based Visualization Taxonomy. School of Computing Science, Simon Fraser University, (CMPT-TR2002-06), p. 4.

Tory, M. and Moller, T. 2004. Rethinking Visualization: A High-Level Taxonomy. IEEE Symposium on Information Visualization, pp. 151-158.

Van Wijk, J. 2005. The Value of Visualization. IEEE Visualization 2005 - (VIS'05), p. 11.

Ward, M. 2002. A taxonomy of glyph placement strategies for multidimensional data visualization. Information Visualization, 1 (3-4), pp. 194-210.

Zuk, T., Schlesier, L., Neumann, P., Hancock, M. S. and Carpendale, S. 2006. Heuristics for information visualization evaluation. pp. 55-60.

Survey

15th March 2014

Arrian Purcell

Purpose

The purpose of this interview was to gain insight into the audience's current understanding and enjoyment of the live coding process. Additionally, the relationship between enjoyment and understanding was to be examined. It was hoped that the examination of these factors would further inform the development of visualisations within live coding.

Method

Survey questions (see Appendix A) were distributed following a live coding performance. Both an online and paper copy were distributed.

Results and Discussion

A total of thirteen survey responses were received. Of these, 77% regularly listen to music and 54% perform regularly. 38% of the respondents have high exposure to programming through work, study or their hobbies, as opposed to 31% who have no experience with it. Of the respondents, 69% had never been to a live coding performance before.

Enjoyment was measured according to the relative change in enjoyment through the performance from the beginning to the end. 46% of survey respondents had high enjoyment throughout the performance. The results for enjoyment are summarised in Table 1. The results suggest an overall high level of enjoyment of the performance. No respondents chose low enjoyment throughout the performance.

	Flat	High	Low	High to Low	Low to High	Unsure
Count	2	6	0	2	1	2

Table 1 : Enjoyment through the performance

Similarly, understanding was measured according to the relative change in understanding through the performance from the beginning to the end. 31% of survey respondents had no change to understanding through the performance The results for understanding are summarised in Table 2. Overall, understanding is spread out more than enjoyment with only 15% suggesting that they could understand the relationship between the visuals and the music throughout the performance. There is no statistically significant relationship ($p > .05$) between music listening habits and understanding nor is there a statistically significant relationship ($p > .05$) between coding experience and understanding.

	Flat	High	Low	High to Low	Low to High	Unsure
Count	4	2	0	2	3	2

Table 2 : Understanding through the performance

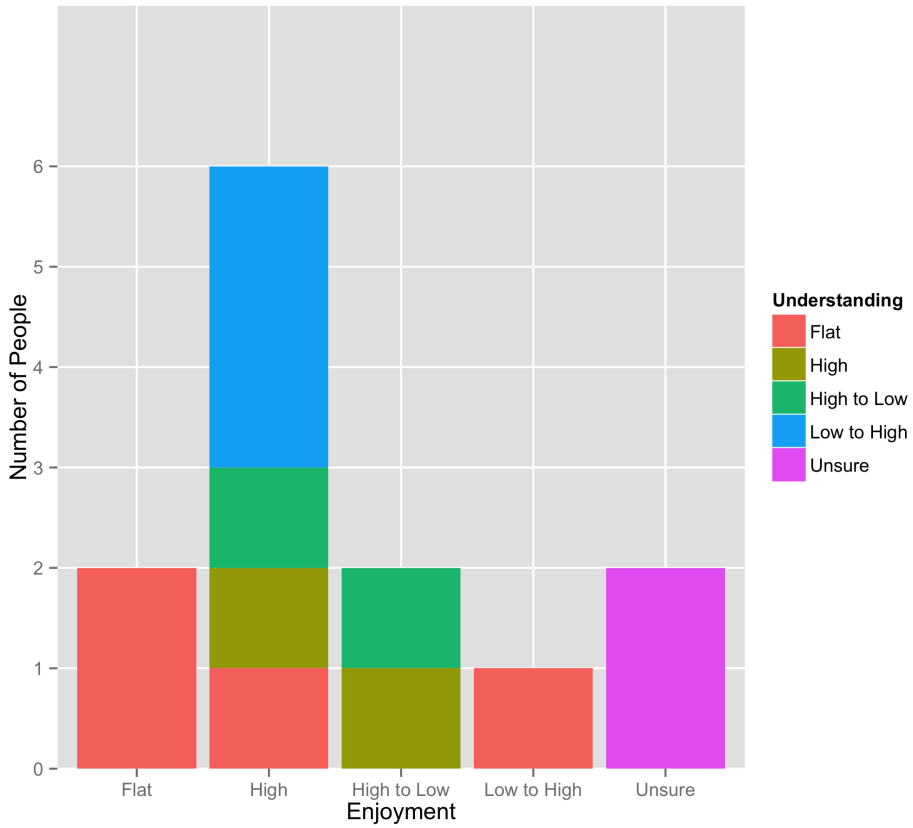
The relationship between enjoyment and understanding can be seen in Figure 1. Notably, three respondents who had high enjoyment throughout the performance were the only respondents who had a pattern of low to high understanding. However, the relationship between enjoyment and understanding is not statistically significant ($p > .05$).

69% of respondents stated that the visuals provided a sense of liveness to the performance. The remained 31% stated that they had no effect on their sense of liveness. There were no responses stating that the visuals negatively impacted the sense of liveness.

In terms of confusion, 38% suggested that no aspects of the visuals were confusing, though 31% did not respond to the question.

Supplementary observations of the performance are available in Appendix B.

Figure 1 : Enjoyment and Understanding



Appendix A - Survey Questions

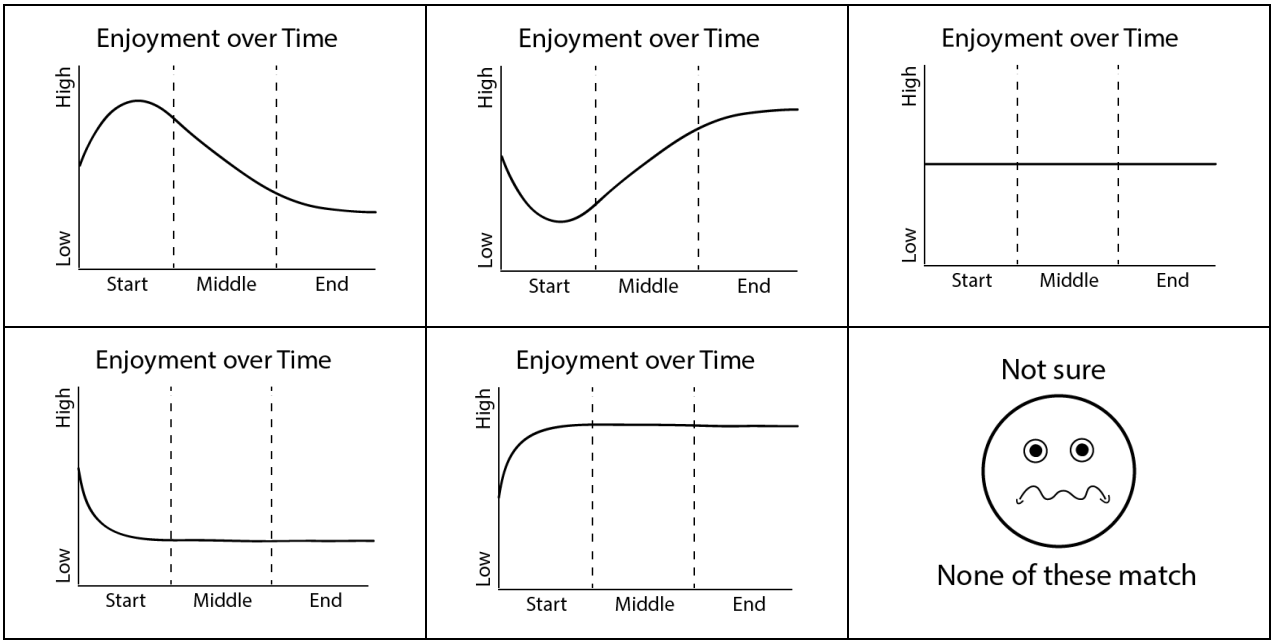
1. How many live coding performances have you been to?
    A. None - This is my first one
    B. Less than five
    C. Five or more

2. How much music do you listen to?
    A. Hardly any
    B. A little
    C. A large amount

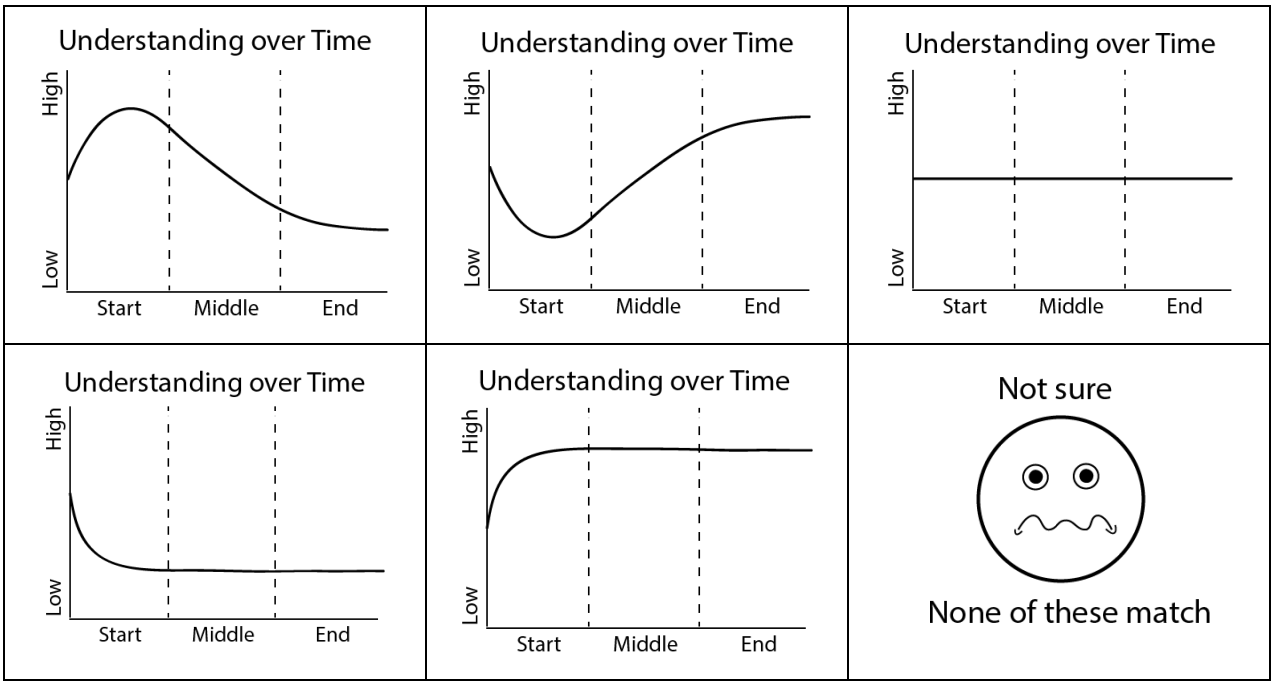
3. Do you play an instrument or sing?
    A. No - I would not consider myself a musician or singer
    B. Occasionally or I have in the past
    C. Yes - I play or sing regularly

4. How much experience do you have with programming?
    A. I have no experience with it
    B. I have some experience programming
    C. I currently program for my study/hobby/work

5. Please circle the image below that best represents your **enjoyment** through the performance.



6. Please circle the image below that best represents your **understanding** of the relationship between the **visuals** and the **music** through the performance.



7. This was a live performance! What effect did the visuals have on your sense of 'liveness' of the performance?

A. The visuals were distracting
B. They had no effect on my sense of 'liveness'
C. They helped

8. Were there any aspects of the visuals that you found confusing?

Appendix B - Supplementary Observations

Overall, the live coding performance went smoothly. The room layout was perhaps not optimal for the performance and the display screen was very dim leaving some parts of the source code unreadable. Additionally, the projection surface was not flat further reducing readability.

An estimated 20 people were in attendance at the start of the performance. This grew to an estimated 30 people by the end of the performance, over a time period of about 20 minutes.

The logistics of handing out paper surveys did not suit the venue layout, however most people had the ability to access the survey through their phone. Online survey attrition was about four people, though the accuracy and reason behind this can not be determined from the data. As suggested by Henry, it may be easier just to hand out the QR code on a piece of paper after the performance.

Henry also noted that the other performer used his visuals to tell a story and this story may or may not relate to the music being played.

Follow-Up Interview

20th March 2014

Arrian Purcell

Purpose

The purpose of the follow-up interviews was to gain an in-depth view of what some of the audience believed they would like to understand more about the performance.

Additionally, it was anticipated that some audiences may prefer not to understand more about the technical details of the performance but would be more interested in focussing on the music throughout the performance. The extent of this sentiment was to be examined.

Method

Two questions were asked of three audience members a number of days after a performance. These two questions were:

1. What did you "understand" about what was going on with the code being projected? In particular, what did you understand about the relationship between the code and the music?
2. Would would you like to understand more about the code in order to enjoy the performance more?

Results and Discussion

A total of three responses were received. interviews are available in Appendix A.

All three respondents referred to the initial stage of developing loops to be played and two of these referred to the silence preceding the music suggesting an understanding within the initial stages of the performance. Two interviewees mentioned that they recognised changes to the code and changes to the music separately but could not see the link between the two.

All three interviewees suggested that it would be nice to know a little about the code or the concepts but it is not essential and may even be detrimental to enjoying the music. One interviewee mentioned switching between observing the code and listening to the music. This transition was described as ‘switching off’ suggesting that the code takes mental effort to understand whereas the music does not.

Appendix A - Interviews

Respondent 1

1) What did you "understand" about what was going on with the code being projected? In particular, what did you understand about the relationship between the code and the music?

The code sets up a set of nested loops which are then modified by the composer in real time. This immediately leads to the danger of repetitive loops. It may be useful to have rhythms of 4 or 8 bar repetition as in Africa. Actually, this musical form lends itself to that type of rhythm and music. As soon as an organ comes in I am reminded of Mike Oldfield and am anxious that someone will say “slightly distorted guitar”. IN jazz one improvises on standards so there is a very strong form (AABA etc) which, in general, is respected allowing the audience to deduce where they are in the piece. I had the feeling that the present way the code is used limited the music

2) Would would you like to understand more about the code in order to enjoy the performance more?

Yes, I think that if the audience were told what was happening or the ideas behind the constructs then I would be happier. Compared to jazz it is not note by note improvisation so an explanation of the limits and advantages would be useful.

Respondent 2

1) What did you "understand" about what was going on with the code being projected? In particular, what did you understand about the relationship between the code and the music?

In the beginning, I could tell from the silence and the live coding that it was being build, and sound by sound line by line was being added to as the piece grew. When Ben went back in the code to change beats or melodies, I could tell something was being changed but wasn't tracking what or how.

2) Would would you like to understand more about the code in order to enjoy the performance more?

I feel like I already understood a rudimentary amount [...] which was enough to enjoy it. I feel like if I had more knowledge about code I would focus on that to the detriment of the music; and if I knew more about music then I may have focused on that to the detriment of my attention on the code. Considering my education, if I had not had the exposure to code and music through [...], then some basic rudimentary knowledge of code would have been good.

Respondent 3

1) What did you "understand" about what was going on with the code being projected? In particular, what did you understand about the relationship between the code and the music?

I understood that the music was being made from scratch and this was evident in the long silence before any sound is heard. I understand what sounds are being made based on the code names and that some of the numbers represent timing, volume and pitch. I still don't quite understand when the code is "ready" and starts working to make music. It has something to do with the highlighting the text, but that also confuses me. (I understand that most of the music is stored in the program as "sound bites" of real instruments, but sounds can also be made from scratch as mathematical wave functions - but I don't think I would know this if Ben hadn't have told me). Sometimes the coder scrolls up and down the screen to much and I get lost, I don't have a big picture of what all the code looks like.

2) Would you like to understand more about the code in order to enjoy the performance more?

In some ways I would like to understand a little more about the code. It would be nice to have a director's commentary of what's going on behind the scenes, just so I can follow along with the changes that I can hear in the music as they are occurring. But I think I more enjoy just listening to the music, knowing broadly that a livecoder is manipulating code to make the sounds that I hear. I don't often like reading the code for the whole performance, maybe for a few minutes at a time, but then I like to switch off and just focus on what the musician is playing. I more often like to listen to the music and guess what the livecoder has done to make that changes (which is kind of backwards). I wouldn't mind having more understanding of the code on hand, but I probably wouldn't use the details of it during the whole performance every time.

Taxonomy

20th March 2014

Arrian Purcell

Code Visualisation

Visuals based on Code Augmentation (eg. infographics, annotations, sparklines)
Visuals based on Abstraction (eg. scheme bricks, gource, code flow)
Domain Visualisations (eg. fluid source code views, indentation, class diagrams)

Music Visualisation

Generative Visualisations (eg. frequency wave, VLC/iTunes visualisations)
Associative or Emotive Visualisations (eg. video art, sampled video)
Domain Visualisations (eg. ableton, sheet music)