

Visualising Live Code Study Report

Arrian Purcell

June 27, 2014

For most of its history source code has been displayed as simple text due to the expressiveness of this format, despite its inefficiencies. It is only recently, due to ever increasing programming language complexity and increasing computational power, that code annotations and syntax highlighting have become more commonplace. Nevertheless, these visual enhancements rarely provide information beyond the basic grammar of the language they are intended to augment. The limitations of this approach are becoming ever more apparent as programming languages and interactive programming environments move towards the need for real-time comprehension and a need to understand the source code within the context of a running program.

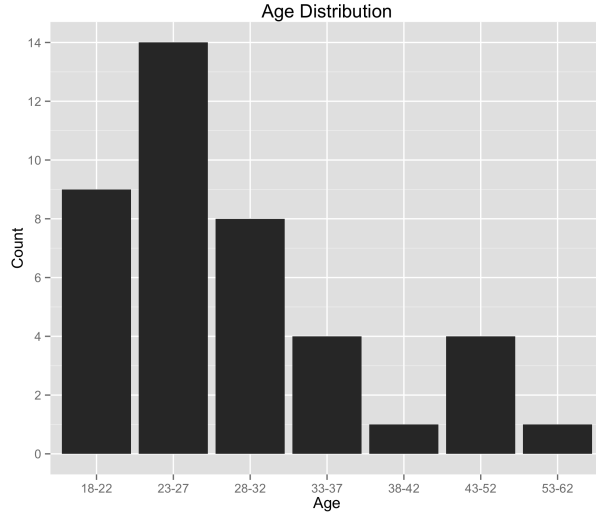
The problem of understanding code within a running environment is complex. The fact that programming concepts are fundamentally different to human understandable concepts [1] presents challenges in communicating the complex nature and intention of the program. The fundamental complexity lies within the translation from code to the brain; the programming concepts to the human understandable concepts.

The human brain is highly proficient in pattern recognition [2] and there is evidence to suggest that visualisations can take advantage of this proficiency to enhance understanding [3]. It would be informative to investigate this translation using visual representations to translate and assist in the understanding of program concepts.

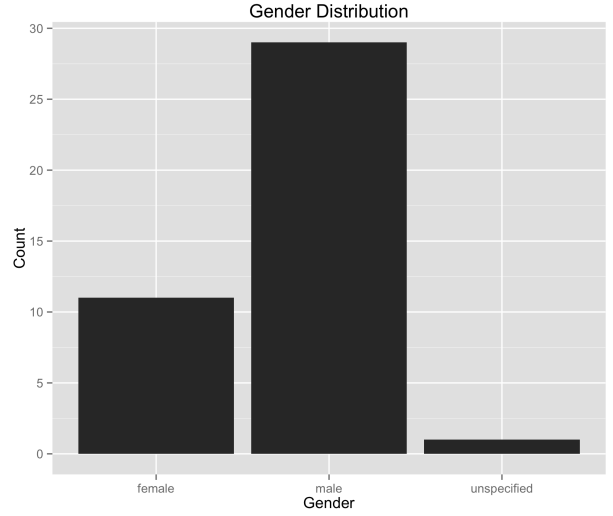
To this end, two code visualisations have been analysed in a live coding context to determine effective presentational and educational features. The two visualisations evaluated included a visualisation targeting aesthetic appeal and a visualisation with a more didactic approach. The goal was to determine differences and desirable aspects of the two visualisation approaches in order to inform future live coding visualisations.

The set of didactic visualisations predominantly focussed on the relationship between the live coding active processes and their behaviour. The visualisations prominently displaying the names of the active functions with visual indication of the number of functions running and their callback time. Bright colours and solid shapes were used to ensure constant visibility and communicate the intention of the underlying code. An example of this approach can be seen in Figure 5b. Overall, four visualisations were presented with each introduced depending on the number of active functions.

The set of aesthetic visualisations focussed less on the programmatic aspects of the live coding performance, rather intending to provide additional visual interest to the projected code thereby increasing retention. More variety was used in visual structure and colour. An example of this approach can be seen in Figure 5a. Again, four visualisations were presented, varying the visualisation based on the number of active functions.

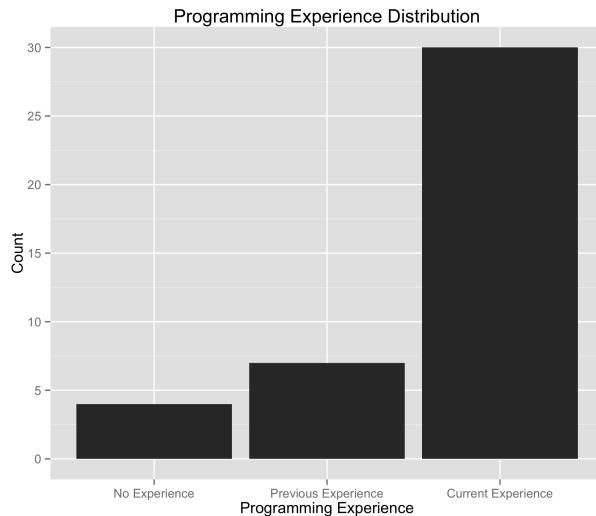


(a) Age Distribution

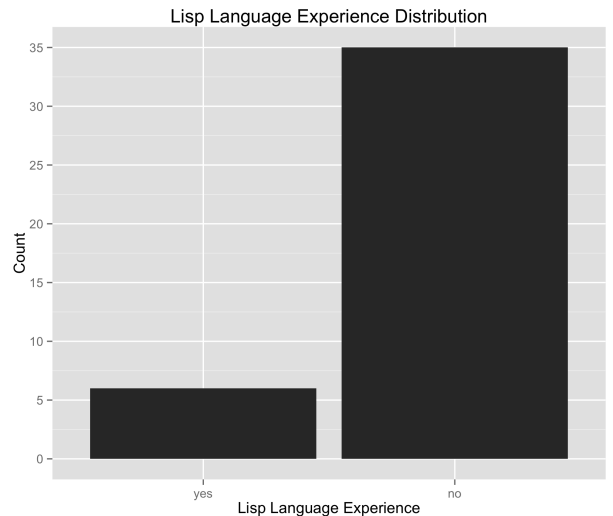


(b) Gender Distribution

Figure 1: Basic Demographics



(a) Programming Experience Distribution



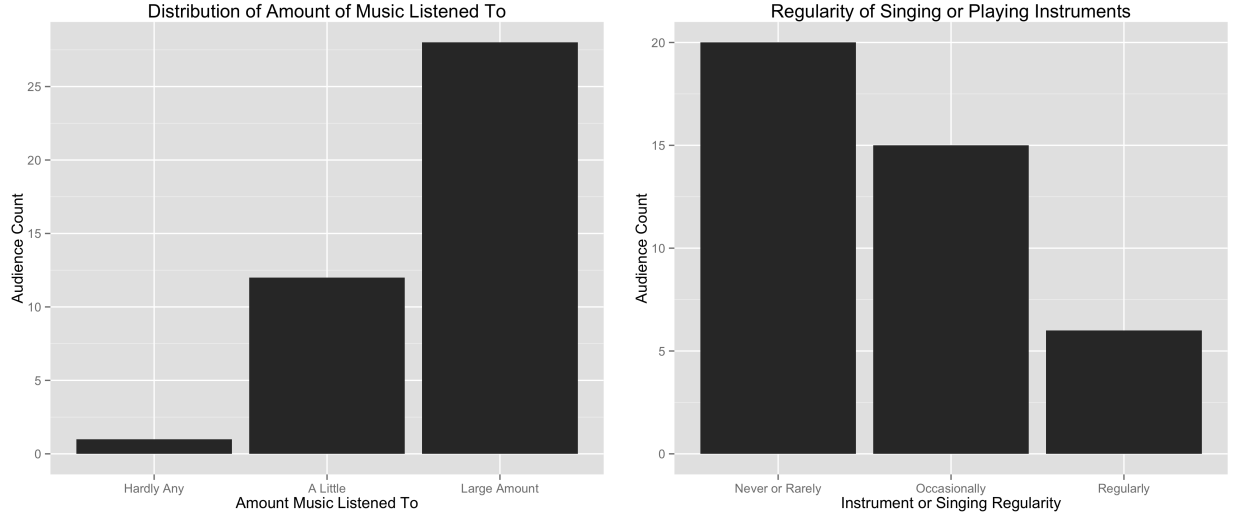
(b) Lisp Experience Distribution

Figure 2: Programming Demographics

1 Method

Two sets of visualisations were presented to an audience during a live coding performance and surveys were distributed. The performance was run as two ten minute improvised songs with each song demonstrating one of the two experimental visualisation conditions, either the didactic condition or the aesthetic condition. The experiment was run twice, with two separate participant groups, swapping the order in which the two sets of visualisations were presented to the audience.

The audience members completed a survey consisting of four sections over the course of the performance. These sections included a demographic information section, opinion of the first visualisation, opinion of the second visualisation and a section investigating the audience's overall opinion of the performance. The sections regarding the visualisations predominantly focussed on the enjoyment and understanding related to



(a) Listen to Music Regularity Distribution

(b) Playing Instrument or Singing Regularity Distribution

Figure 3: Musical Demographics

the visualisation while the final section focussed on eliciting potential improvements.

After explaining to the participants the structure of the experiment and allowing the participants to complete the initial demographic section of a survey, the live coder began the first performance utilising one of the two visualisations. Following this first set, the second section of the survey was conducted. A second set was then played using the alternative visualisation. Following this second set, the same survey questions were administered again, again asking the participants specifics about their enjoyment and understanding related to the specific visualisation demonstrated. A final survey question was then asked relating to their opinion regarding the whole performance and suggested improvements.

2 Participants

A total of 41 participants took part in the study. Over the two performances, 19 participants observed the first performance and 22 participants observed the second performance. The demographic makeup between the two performances was very similar.

66% of the participants stated that they were male (see Figure 1b) and most participants were aged between 18 and 32 (76%, see Figure 1a). As the study was conducted within the Computer Science Department, a large proportion of the participants were experienced with programming with 90% having current or previous experience with it (see Figure 2a). Nevertheless, only 15% of participants had previous experience with any of the Lisp style of languages (see Figure 2b).

Of the participants, 68% stated that they listened to a large amount of music (see Figure 3a) though only about 15% of participants stated that they played and instrument or sung regularly (see Figure 3b). Only 22% of participants had seen a live coding performance before.

3 Results

3.1 Aesthetic vs Didactic Visualisation

Understanding and enjoyment were evaluated against the didactic and aesthetic visualisations. Results and statistical analysis of the differences between the aesthetic and didactic conditions follows. Note that for the following statistical analysis a significance level of 0.05 was used with the chi-squared test for independence.

3.1.1 Understanding

Overall, 37% participants stated specifically that the didactic visualisations helped them to understand the code, whereas 12% participants stated that the aesthetic visualisations assisted in understanding the code.

H_0 : There is no difference between the aesthetic visualisations and didactic visualisations in terms of understanding.

H_1 : There is a difference between the two visualisations in terms of understanding.

A significant difference between the visualisations effect on understanding was found ($\chi^2 = 7.1986, df = 2, p = 0.02734$).

3.1.2 Enjoyment

Overall, for both visualisations, a large proportion ($> 50\%$) of the participants stated that the visualisations helped their enjoyment of the performance. Of the participants, 76% stated that the aesthetic visualisations helped their enjoyment compared to 56% participants that stated the didactic visualisations helped their enjoyment.

There was indication that any decreases in enjoyment may have been postponed when shown the aesthetic visualisation and that there was overall an increasing enjoyment during the middle of the performance when this visualisation was shown. See Figure 4a and Figure 4b for the distribution of changes to enjoyment between the aesthetic and didactic visualisations.

H_0 : There is no difference between the aesthetic visualisations and didactic visualisations in terms of enjoyment.

H_1 : There is a difference between the two visualisations in terms of enjoyment.

No significant difference between the two visualisations effect on enjoyment was found ($\chi^2 = 3.7733, df = 2, p = 0.1516$).

3.1.3 Liveness

Participants were asked to discuss how each visualisation influenced or impacted the liveness of the performance. Concepts discussed included the positive and negative aspects of the didactic visualisations, the positive and negative aspects of the aesthetic visualisations, source code discussion and statements indicating an understanding between the visuals and the source code. A summary of the references and coverage of these concepts is available in Table 1. The references column indicates the number of times the concept was mentioned and the coverage indicates the total amount of discussion of the concept.

3.2 Improvements

The audience was asked what improvements to the visualisations they would like to see. 40% of the audience suggested improvements that indicate a desire to see more relationship between the visualisations and the music, with suggestions such as matching the visualisations to musical rhythms or pitch.

Other suggestions included increasing the readability of source code, increasing uniformity between the visualisations and the use of images to assist with the understanding of high level concepts.

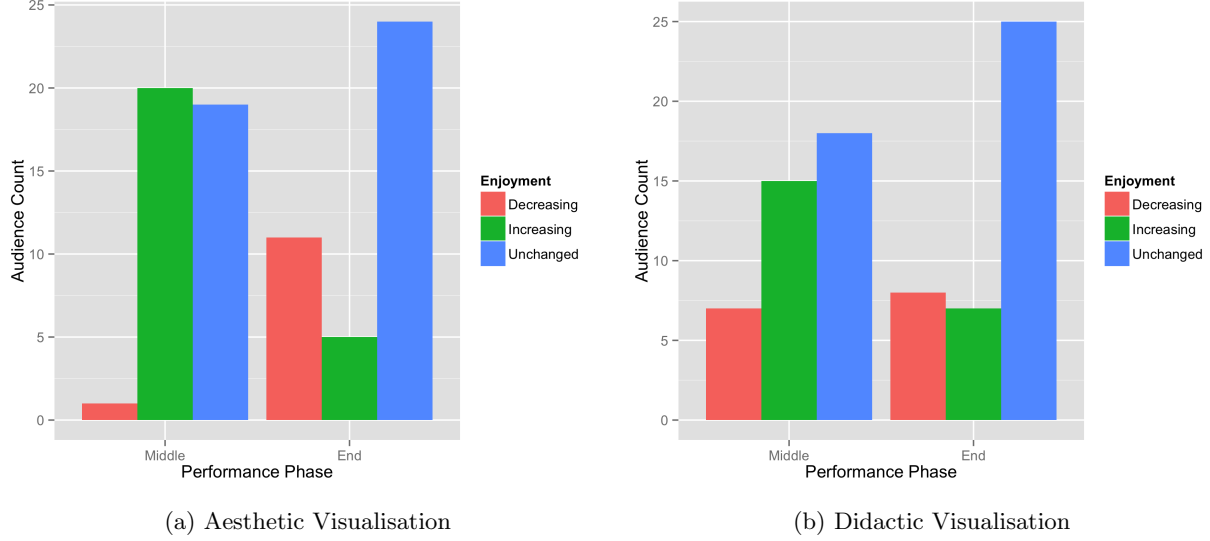


Figure 4: Change of enjoyment through specific phases of the performance for aesthetic and didactic visualisations

Table 1: Liveness Concept Coverage from Audience Feedback

Concept	References	Coverage
Negative Didactic Aspects	26	17%
Positive Didactic Aspects	19	12%
Negative Aesthetic Aspects	21	11%
Positive Aesthetic Aspects	14	8%
Source Code	40	28%
Demonstrating Understanding	6	5%

4 Discussion

Overall enjoyment of the visualisations was high. This was the case for both the aesthetic and didactic visualisations. However, it is interesting to note that enjoyment of the aesthetic visualisation was higher and generally increased during the earlier stages of the performances suggesting that the aesthetic visualisations held the audience’s attention for longer (as shown in Figure 4a). The didactic visualisation on the other hand had a more constant decrease in enjoyment. The audience improvements offer some insight into this suggesting that the visualisations were competing with the projected code.

Although the results surrounding the understanding afforded by the visualisations are unsurprising with the didactic visualisations having a clear educational benefit over the aesthetic visualisations, this will provide a baseline for future visualisations.

A number of participants stated that the didactic visualisations were distracting, too abrupt or took away from the code in other ways. On the other hand, some said that the didactic visualisations gave the performance a sense of being ‘too polished’. The general consensus, however, was that the visualisation should either better reflect the code, better reflect the music or have more variety.

The overall reponse indicates that there is much potential to improve the visualisations, developing both the aesthetic aspects of the visualisations in addition to the educational aspects of the visualisations. Future visualisations will focus on taking advantage of the prolonged attention afforded by the aesthetic visualisa-

tions and combine it with the more educational aspects of the didactic visualisations demonstrated within this experiment. With these results there is some progress towards an effective interactive programming environment visualisation.

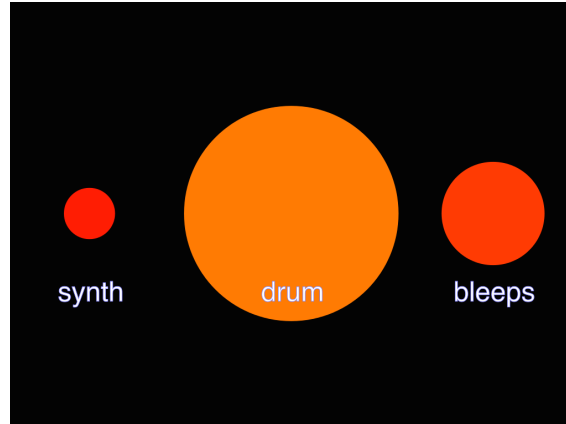
References

- [1] TJ Biggerstaff. Program understanding and the concept assignment problem. *Communications of the ACM*, 1994.
- [2] Colin Ware. *Information visualization: perception for design*. Elsevier, 2013.
- [3] Lawrence J Najjar. Principles of educational multimedia user interface design. *Human Factors: The Journal of the Human Factors and Ergonomics Society*, 40(2):311–323, 1998.

Appendix A



(a) Aesthetic Visualisation



(b) Didactic Visualisation

Figure 5: Visualisation Examples