Understanding through Code Visualisation

Arrian Purcell

A subthesis submitted in partial fulfillment of the degree of
Bachelor of Software Engineering (Honours) at
The Department of Computer Science
Australian National University

March 2014

Typeset in Palatino by TEX and $\LaTeX\,2_\varepsilon$.

Except where otherwise indicated, this thesis is my own original work.

Arrian Purcell
19 March 2014

To fluffy the cat.

Acknowledgements

Here is where you thank the people who helped you along the way…

Abstract

Live coding is a method of performance that presents audio and visual content to audiences through programming. Often ″showing the code″ is a fundamental part of the performance in order to retain the attention of the audience and provide a measure of authenticity.

Currently missing within the research within 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 formal analysis of the target audience.

Contents

Acknowledgements **vii**

Abstract **ix**

1 Introduction **1**

1.1 Summary - remove 1

1.2 Background 1

1.3 Theoretical Framework 2

2 Literature Review **3**

2.1 Live Coding 3

2.1.1 Music vs Visualisation 3

2.2 Music Visualisation 3

2.2.1 Taxonomy 3

2.2.2 Live Performance 3

2.3 Software Engineering Practice 3

2.3.1 Application of Software Engineering to Live Coding 4

2.3.1.1 Design 4

2.3.1.2 Coding 4

2.3.1.3 Comprehension 4

2.3.2 Application of Live Coding to Software Engineering 4

2.3.2.1 Dissemination of Code Understanding 4

2.3.2.2 Multidisciplinary Cohesion 4

2.3.2.3 Visualisation Framework 4

3 Survey **5**

3.1 Purpose 5

3.2 Hypothesis 5

3.3 Materials 5

3.4 Method 5

3.5 Observations 5

3.6 Results 5

3.7 Supplementary Observations - remove 6

4 Visualisation Experiment 1 **7**

4.1 Rationale 7

4.2 Procedure 7

4.3 Results 7

5 Visualisation Experiment 2 **9**

5.1 Rationale 9

5.2 Procedure 9

5.3 Results 9

6 Conclusion **11**

A Survey Results **13**

B Visualisations **15**

Bibliography **17**

Chapter 1

Introduction

-code is often difficult to quickly understand -some observers may lack the experience to understand the software or the programming process

-how can we improve source code comprehension? -how can we aid understanding of the programming process? -better yet, how can we better communicate the programmers intention?

-techniques such as modelling or code documentation arent dynamic or flexible -dont allow for close to realtime understanding -an effective technique is the use of visualisations -it would be valuable to use visualisations as a means to communicate the programmers intention

1.1 Summary - remove

-this thesis will explore code visualisations -specifically, it will investigate visuals within the combination of the domains of software and music -will be using Iive coding as a platform and case study for this (will discuss later) -will develop and test code visualisations on audiences with audiences of varied levels of experience with programming, addressing code comprehension

1.2 Background

Live Coding -live coding is a platform for bridging these two domain visualisations -what is live coding? -method of programming in front of an audience for artistic or informative purposes -the live coder displays their screen to an audience, showing their code as they are working on it building a functional program -makes use of interactive programming environments -program running while changes are being made -often focusses on improvisation - the programmer often has to think on their feet

-what does live coding achieve? -gives the audience insight into the programming process - ill be taking advantage of this

1.3 Theoretical Framework

Chapter 2

Literature Review

2.1 Live Coding

(focus on developing a narrative concerning what needs to be done within live coding to achieve the software engineering goals and what needs to be done to develop successful visualisations within the field of live coding)

Live coding describes the process of exposing the programming process to a live audience. -talk more about what live coding is

Live coding history... . -talk more about history of live coding

There exists much discussion within the live coding research body (eg. ...) about the potential for live visual manipulation and examination of the current progress within the field to achieve this.

2.1.1 Music vs Visualisation

In addition, there has been a move towards manipulation of the visuals in synchronisation with the ....

2.2 Music Visualisation

2.2.1 Taxonomy

2.2.2 Live Performance

2.3 Software Engineering Practice

As the field of live coding develops, the relevance of both the application of software engineering practice to the field and the relevance of live coding to the field of software engineering has become highly apparent...

2.3.1 Application of Software Engineering to Live Coding

The application of software engineering to live coding... ([Blackwell and Collins 2005] paper incl. requirements analysis, design, coding, project management, reuse, debugging, documentation, comprehension and maintenance)

2.3.1.1 Design

Design...

2.3.1.2 Coding

Coding...

2.3.1.3 Comprehension

Comprehension...

2.3.2 Application of Live Coding to Software Engineering

The application of live coding to software engineering...

2.3.2.1 Dissemination of Code Understanding

2.3.2.2 Multidisciplinary Cohesion

2.3.2.3 Visualisation Framework

Chapter 3

Survey

3.1 Purpose

An intial survey was conducted to analyse an audiences existing understanding of the live coding process.

Describe why you are doing the experiment.

3.2 Hypothesis

Describe what you think will happen.

3.3 Materials

List special materials you used.

3.4 Method

Write a step by step description of what you actually did, identifying the different variables and how you controlled them. Describe what things you changed (variables you manipulated).

3.5 Observations

1.Using all your senses, collect measurable, quantitative raw data and describe what you observed in written form. 2. Reorganise raw data into tables and graphs if you can. 3. Don't forget to describe what these charts or graphs tell us! 4. Pictures, drawings, or even movies of what you observed would help people understand what you observed.

3.6 Results

1.Based on your observations, what do you think you have learned? In other words, make inferences based on your observations. 2.Compare actual results to your hypothesis and describe *why there may have been differences. 3.Identify possible sources of errors or problems in the design of the experiment and try to suggest changes that might be made next time this experiment is done. 4.What have experts learned about this topic? (Refer to books or magazines.)*

3.7 Supplementary Observations - remove

Chapter 4

Visualisation Experiment 1

4.1 Rationale

4.2 Procedure

4.3 Results

Chapter 5

Visualisation Experiment 2

5.1 Rationale

5.2 Procedure

5.3 Results

Chapter 6

Conclusion

Appendix A

Survey Results

Appendix B

Visualisations

Bibliography

BLACKWELL, A. AND COLLINS, N. 2005. The programming language as a musical instrument. . . . *of PPIG05 (Psychology of Programming . . .* , 1–11. (p. 4)