Research Proposal: Serializing Live coding performances on an massively-on-line web platform.

Dario Villanueva

January 14, 2015

1 Introduction

Live coding is an art form. Be it musical, or more recently, audio-visual performances, these enthral and entertain audiences across the world. The pieces themselves exist in their purest form only momentarily, while they are executed by the artist. As such, are usually are usually recorded as videos, effectively "freezing" them into a format that doesn't afford easy analysis.

Due to the relative infancy of live coding as an art form and the "raster" nature of the current preferred mode of storage, the study of these performances is practically limited to manual, ad-hoc work. However, code, the very medium used to create these pieces is inherently serialisable. While many artists divulge the results of these performances on-line as annotated source code files, they lack a key aspect: *a representation of the process undergone by the artist when creating the piece*. This is key, and relates to the first point of the TOPLAP manifesto [1]:

> *Give us access to the performer's mind, to the whole human instrument*

By recording the code and keystrokes inserted by the performers as they play, we enable these pieces to be stored and replayed in an interactive manner. This in turn would allow for analysis and evaluation, encouraging the creation of derivative works. Furthermore, the ability to formally study and

analyse the creative process of these artists would enable more potential live coders to learn from more adept artists, as well as gaining insights on the nature of this novel artform.

We propose the creation of an online platform that would combine the performance ability of environments such as Gibber [2], SuperCollider [3] or LiveCodeLab [4], but which allows performances to be:

- Broadcast and enjoyed live without needing to be in the same physical space as the performer.
- Stored in a serialisable manner, rather than the current audio/video rasterised format.

It is expected that this platform would also encourage collaboration between artists. Once the system is built, it would be trivial to create a multi-performer mode, so that more than one artist could have input at a given time. It would also be relatively trivial to set up the different *game modes* and competitive scenarios described by Nilson [5]: *Scrabble*, *Tetris Challenge*, *Root War*, etc.

By making the system open source, it is expected that the collaborative nature of the live coding scene would encourage other individuals to improve and create more modes - perhaps allowing for competitive "laptop battles" happening on-line for thousands to witness, enjoy and learn from.

2 Research Questions

- Creating a technique to serialise, store and broadcast live coding performances to an audience as they are performed, storing them on a server and executing them locally as they are received.
- How would the replayability and transparency of a live coding piece help lower the high entry barrier to live coding. Can an on-line platform help others learn how to livecode more effectively?
- Are there any common patterns on live coding performances that can be identified? Are there different *schools* or *styles* that can be quantitatively identified? Could this on-line platform help create and spread these styles?

- Given an online competitive arena, can live coding become a game, perhaps a sport?

3 Process

There are currently several open-source web live coding environments which a great number of artists are already familiar with. Leveraging these systems, the platform would integrate a web sockets system to connect several instances of these live coding environments, in either performer or audience form.

The server orchestrating the performance, relaying the keystrokes and code changes to the all of the clients, would also be in charge of storing these; and on demand, it would allow for retrieval of the piece by a client at any point, usually by calling on a known URL route. When playing back a performance, the user will be able to know when different pieces of code were triggered, and when the code was altered. In playing back the performance, the user will be able to easily alter it, creating derivative forms.

One of the major challenges to overcome in order for this platform to work would be the creation of a format that efficiently captures the actions of the performer, while remaining lightweight. Furthermore, a visual language needs to be developed to communicate and visualise these changes, either as a diagram notation or some other form of musical score.

References

[1] Various. *The TOPLAP Manifesto (draft)*. URL: `http://toplap.org/wiki/ManifestoDraft`.

[2] Charlie Roberts. *Gibber (About)*. URL: `http://charlie-roberts.com/gibber/about-gibber/`.

[3] James McCartney. *Supercollider*. URL: `http://supercollider.sourceforge.net/`.

[4] Davide Della Casa and Guy John. *LiveCodeLab*. URL: `http://livecodelab.net/`.

[5] Nick (Click Nilson) Collins. “Live Coding Practice”. In: *University of Sussex. Department of Informatics* (2005). URL: http://www.sussex.ac.uk/Users/nc81/research/livecodingpractice.pdf.