Research Proposal: Turning Live coding performances into an online audience game

Darío Villanueva

January 20, 2015

Introduction

Live coding is an art form. Current audiovisual performance languages [1, 2, 3, 4, 5] put code at the artist's disposal to act as a bridge between the analog and the digital [6]. These talented individuals create pieces that enthral and engage audiences across the world. However, these experiences exist momentarily, only whilst being executed by the artist. They are usually recorded as videos, which effectively "freeze" them into a format that doesn't easily afford rework, study or analysis. Authors have outlined techniques to improve and practice livecoding, in the form of games and competitive scenarios such as *laptop battles*, *Scrabble*, *Tetris Challenge* etc [7, 8].

I propose the creation of an online game that allows live coding performers to compete in front of an audience. Its objective is to encourage collaboration, play, study and practise. The platform will offer different game modes and competitive scenarios where performers can compete against themselves, an AI or each other.

The platform will operate in two modes: Performer or Audience. In performer mode, one or more artists can collaborate or compete in any of the game modes, creating graphics and music on a browser tab. In Audience mode, any number of people will be able to watch a given peformance or competition - The server will distribute code and keystrokes as the performer(s) play and execute it locally.

The server orchestrating all this will connect the Performer(s) with the Audience via websockets. Performances will be serialised and stored in a searchable database. When Audiences play back these performances, they will be able to pause, replay and alter (*fork*) them, allowing for the creation of derivative works. The audience watching a competitive performance will also be able to give kudos to either of the contestants, which will decide which of the performers will win that particular "battle".

Research Questions

- Creating a technique to serialise, broadcast, store, visualise and reproduce live coding performances to an audience.
- Creating an online platform that leverages the aforementioned technique in which artists can perform, interact, collaborate and compete in different game modes and events.
- How will this newly afforded replayability and serialisability help lower the high entry barrier to live coding. Can an on-line platform help others learn how to livecode more effectively?
- Given an online competitive arena, can live coding become a game, perhaps an sport? Can this platform spark the creation of novel musical games?

References

[1] Charlie Roberts. *Gibber*. URL: `http://charlie-roberts.com/gibber/about-gibber/`.

[2] Alex McLean. *Tidal*. URL: `http://tidal.lurk.org/`.

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

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

[5] Dave Griffiths. *fluxus*. URL: `http://www.pawfal.org/fluxus/`.

[6] Alex Mclean. "The Textural X". In: *Computational Communication Aesthetics and X. Bergamo, Italy.* (2013).

[7] Nick Collins et al. "Live coding in laptop performance". In: *Organised Sound* (2003).

[8] 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`.