Stir bugs

GERARD ROMA, University of West London, United Kingdom

Additional Key Words and Phrases: live coding, agent-based models, corpus-based performance

ACM Reference Format:
Gerard Roma. 2023. Stir bugs. 1, 1 (May 2023), 3 pages.

1 PROGRAM NOTES

Stir bugs is an exploration of live algorithmic control in corpus-based performance. A community of computational agents confined to a two-dimensional square prison cell is live-coded into collective madness.

Agents are controlled by simple code functions that define navigation in a *terrain* made of a collection of electronic noise samples. Each agent is also associated with a sound playback/synthesis function. The performance embraces the complexity emerging from quickly coding a multiplicity of behaviours in a shared sonic space.

2 PROJECT DESCRIPTION

In previous work [3] a framework for the visualization of sound collections was presented, including an implementation in the SuperCollider language[1]. This allows for representing a sound corpus as a 2D grid which is computed through audio analysis and dimensionality reduction. Each individual sound is represented as an icon which shows the temporal evolution of loudness and spectral centroid. The visualization can then be used for corpus-based composition and performance [4].

One way to navigate the corpus is by using live coded artificial agents. This approach, inspired by agent-based modelling [1, 5], can be described as agent-based music live coding (ABMLC). The conceptual model and implementation are explained in more detail in [2]. Each agent is defined by a short code snippet which controls the position in the visualization. Several functions, such as patterns, lines, random walks and orbits are explored. The visual animation improves the readability of the performance with respect to the raw code.

Mob[2] is a bespoke live coding environment written in SuperCollider, which allows live coding of visual agents on a 2D surface. The *terrain* (data available to agents in the shared space) may be generated in different ways, including the visualization of a sound corpus. Each agent is live-coded through its corresponding text box in plain SuperCollider language. An agent is also associated with one of several synth definitions which have been coded in advance. In addition to the position and the buffer of the sound associated with it, synths receive an amplitude parameter which is controlled through a hardware MIDI slider. Figure 1 shows a screenshot of the interface. A random colour is associated with each agent, which is used both in the plot and in the code editor tab.

[1] https://github.com/flucoma/FluidCorpusMap2
[2] https://github.com/g-roma/Mob

Licensed under a Creative Commons Attribution 4.0 International License (CC BY 4.0). Copyright remains with the author(s).

Music Proceedings of the International Conference on New Interfaces for Musical Expression
NIME'23, 31 May — 2 June 2023, Mexico City, Mexico

In *Stir bugs*, this system is used to explore the continuum between repetitive and random structures. The corpus was created through experimentation with a small hardware modular sound processor. The musical form is improvised through live coding, but familiarity with the sound corpus and its visualisation provides the performer with some leverage in the negotiation with the emerging collective artificial agency.

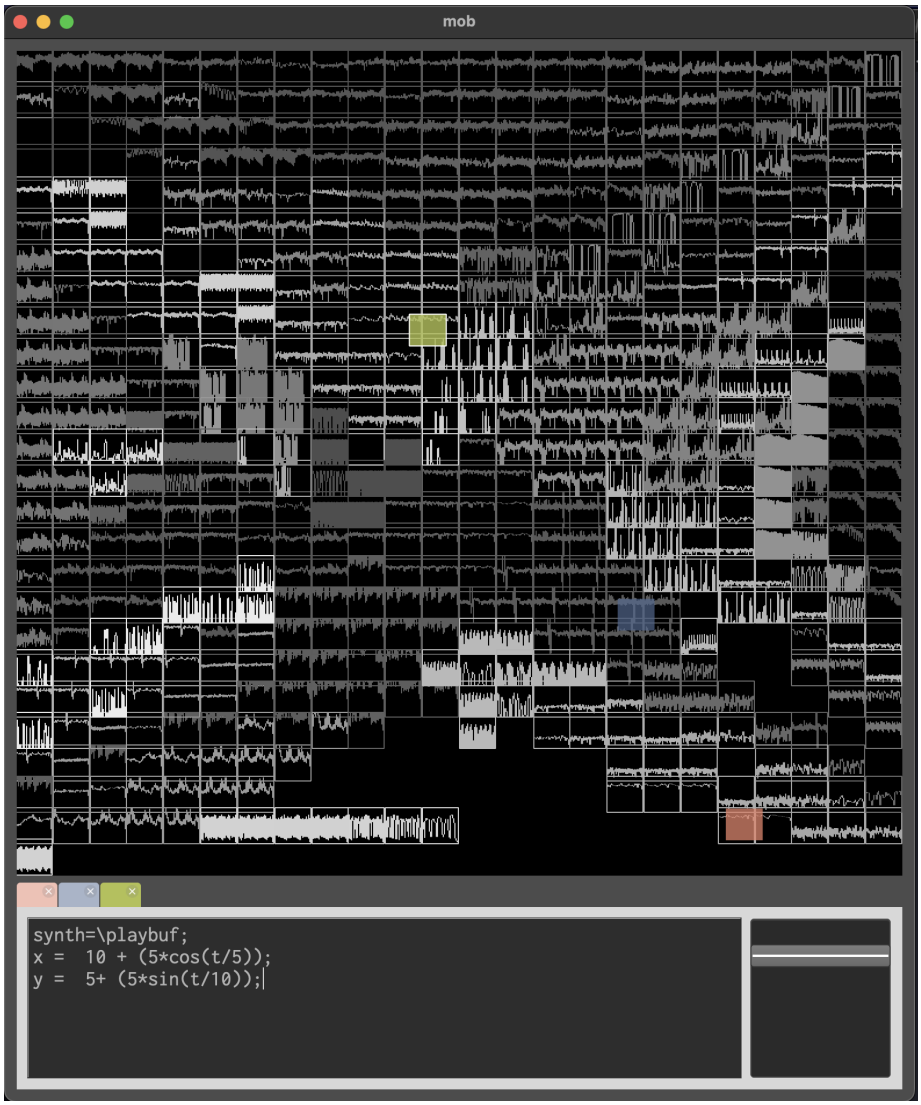

Fig. 1. *Mob* user interface

3 PERFORMANCE NOTES

The performance is expected to last 12-15 minutes and will be presented remotely as an audio-visual solo live coding improvisation using *Mob*.

REFERENCES

[1] Mitchel Resnick. 1997. *Turtles, Termites, and Traffic Jams: Explorations in Massively Parallel Microworlds.* MIT Press.

[2] Gerard Roma. 2023. Agent-based Music Live Coding: Sonic Adventures in 2D. *Organised Sound* 28, 2 (2023). (In press).

[3] Gerard Roma, Anna Xambó, Owen Green, and Pierre Alexandre Tremblay. 2021. A General Framework for Visualization of Sound Collections in Musical Interfaces. *Applied Sciences* 11, 24 (2021), 11926.

[4] Diemo Schwarz, Grégory Beller, Bruno Verbrugghe, and Sam Britton. 2006. Real-time corpus-based concatenative synthesis with catart. In *9th International Conference on Digital Audio Effects (DAFx)*. 279–282.

[5] Seth Tisue and Uri Wilensky. 2004. Netlogo: A simple environment for modeling complexity. In *International Conference on Complex Systems*, Vol. 21. Boston, MA, 16–21.