Browser-based Collaborative Live Coding with Glicol: A Graph-oriented Live Coding Language Written in Rust

Qichao Lan
RITMO
Department of Musicology
University of Oslo
qichao.lan@imv.uio.no

Alexander Refsum Jensenius
RITMO
Department of Musicology
University of Oslo
a.r.jensenius@imv.uio.no

ABSTRACT

In this workshop, participants will be invited to try out Glicol, a graph-oriented live coding language written in Rust. Participants will get familiar with the syntax of Glicol, as well as its browser-based environment developed with WebAssembly, AudioWorklet and SharedArrayBuffer. In the browser-based interface, a new form of interaction in collaborative live coding will be introduced too. After that, participants can brainstorm new features together and learn how to customise the language. In addition, there will be a scheduled live coding performance with Glicol at the conference, and participants of the workshop can choose to join as co-performers.

1. INTRODUCTION

Glicol[1] (an acronym for graph-oriented live coding language) is a new computer music language primarily designed for music live coding. As its name suggests, the quintessence of Glicol is to use a graph-oriented syntax to represent the audio graph in a straightforward manner during live coding performances, bypassing common programming paradigms such as object-oriented programming or functional programming. Glicol has some prominent features as follows:

- **Glicol is fast.** Glicol has both its language and signal engine written in Rust, a system-level language that aims at memory safety and efficiency. With the 'zero-cost abstraction' of Rust, Glicol can minimise the performance loss from converting the code string to audio float stream.
- **Glicol is good at error handling.** Taking advantage of the robust error handling mechanism in Rust, Glicol can capture and feedback the errors such as typos or misspelling during a live coding performance while continuing the music based on previous error-free code.
- **Glicol is intuitive.** We adopt the WYSIWYG (What You See Is What You Get) design pattern in Glicol interaction, which means that the code on the screen is exactly what is converted to the music sound. Muting a track is equivalent to commenting out a line of code. Lots of work has been done under the hood such as using the longest common subsequence (LCS) algorithm every time the graph updates. Details of the implementation can be found on Glicol's GitHub repository [2] and the presentation at the conference.
- **Glicol is friendly towards collaboration.** Using WebAssembly [3] and AudioWorklet[1], Glicol can run on its browser-based interface which also supports collaborative live coding. More information can be found on its newly launched official website.

2. WORKSHOP PLAN

This workshop will consist of two sessions. In the first session, participants will get familiar with the Glicol language and its web-based run-time environment. Given the virtual setup of the conference, we encourage participants to browse Glicol's website in advance to make this session more efficient.

The second session will begin with a discussion to reflect on the previous session and brainstorm new features. For example, we can learn how to prototype new nodes on Glicol's web interface and implement it in Rust using a given Macro template. To illustrate, the `plate` node in Glicol that wraps the Dattorro's reverb effect [2] is written with the Glicol syntax in Rust. Also, participants are welcomed to give suggestions on the tutorial compilation and the collaborative interface design of Glicol.

Last but not the least, this workshop is linked with a scheduled collaborative performance at the conference, in which the participants of the workshop can choose to join as performers as well. Thus, we may view the workshop, especially the first session, as an online rehearsal of the performance. We expect to see an iterative process in the workshop where we can find issues, improve the language and yield knowledge.

3. AUTHORS SHORT BIOS

Qichao Lan is a computer-musician and researcher in music technology, specialising in audio programming, live coding, new instrument design, and music AI. He is also publishing open-source software and performing live coding music under the name 'chaosprint'. In 2018, Qichao gained his master's degree in Sonic Arts at the University of Sheffield.

[1] https://glicol.org

[2] https://github.com/chaosprint/glicol

Web Audio Conference WAC-2021, July 5–7, 2021, Barcelona, Spain.

Now, he is a doctoral research fellow funded by the Nordic SMC organisation and works for the RITMO Centre at the University of Oslo. He is currently focusing on developing a new programming language called Glicol which can be used for collaborative live coding music performance, music AI research and STEM education.

Alexander Refsum Jensenius is a music researcher and research musician. His research focuses on why music makes us move, which he explores through empirical studies using different types of motion sensing technologies. He also uses the analytical knowledge and tools in the creation of new music, with both traditional and very untraditional instruments. As chair of the NIME steering committee, he is a leading figure in the international computer music community. From 2017 he co-directs RITMO Centre for Interdisciplinary Studies in Rhythm, Time and Motion, an interdisciplinary centre of excellence at the University of Oslo. As a member of the EUA Expert Group on Science 2.0/Open Science, he is also involved in pushing for modernising the way research is conceived and conducted.

4. REFERENCES

[1] H. Choi. Audioworklet: the future of web audio. In *ICMC*, 2018.

[2] J. Dattorro. Effect design, part 1: Reverberator and other filters. *Journal of the Audio Engineering Society*, 45(9):660–684, 1997.

[3] A. Haas, A. Rossberg, D. L. Schuff, B. L. Titzer, M. Holman, D. Gohman, L. Wagner, A. Zakai, and J. Bastien. Bringing the web up to speed with webassembly. In *Proceedings of the 38th ACM SIGPLAN Conference on Programming Language Design and Implementation*, pages 185–200, 2017.