

# LOLC for Laptop Music Ensemble

Jason Freeman, Sang Won Lee, Shannon Yao, Aaron Albin

School of Music

Georgia Institute of Technology

840 McMillan Street

Atlanta, Georgia 30332-0456

+1 404 385 7257

jason.freeman | sangwonlee | syao31 | aalbin3@gatech.edu

## ABSTRACT

This statement describes LOLC, a text-based collaborative music improvisation environment for laptop ensemble developed at Georgia Tech and presented in performance by the authors at the Creativity and Cognition conference.

## Author Keywords

music, improvisation, text, collaboration, laptop

## ACM Classification Keywords

J.5. Computer Applications. Arts and Humanities. Music.

## General Terms

Experimentation, Human Factors.

## LOLC OVERVIEW

In LOLC [1], the musicians in a laptop ensemble use a textual performance interface, developed specifically for this piece, to create and share rhythmic motives based on a collection of recorded sounds. The environment encourages musicians to share their code with each other, developing an improvisational conversation over time as material is looped, borrowed, and transformed.

The LOLC software environment comprises server software that synchronizes pattern definitions across the ensemble and maintains time synchronization; client software that presents an integrated development environment (IDE) style interface for text-based performance and collaboration; and a visualization of performers' activities that is projected for the audience to watch.

LOLC's design is inspired both by live-coding systems and by traditional forms of ensemble improvisation, particularly in jazz and art music. Musicians type in simple one-line code statements to create rhythmic patterns based on sound files, to play and loop those patterns, and to transform and combine their own patterns and those of other ensemble members.

Our goal with LOLC was to create an environment that encouraged a conversational style of improvisational

creativity within a large laptop ensemble, using the idea of borrowing and transformation as the core paradigm. Network functionality serves to make that collaboration easier, since this interaction paradigm can otherwise be difficult in a laptop ensemble, where each member would need to manually recreate the musical content or underlying algorithm in order to borrow from other musicians.

LOLC is also designed to be accessible to musicians with no experience in computer music or computer programming. The environment's easy learning curve and simple syntax guarantees that almost anyone can learn to perform competently with the environment after a few hours of practice, and we have successfully staged performances with musicians who have no technical background whatsoever.

In performances of LOLC, audiences are also able to relate to the performers via the projected visualization. The projection does not merely display code fragments, it also displays chat messages performers use to plan and comment upon the music as it unfolds. It also visualizes the collaborative relationships among the performers as they borrow and transform each other's material.

## CREATING MUSIC WITH LOLC

Pre-recorded sound files serve as the base musical element in LOLC. Although any sound file can be used, short and percussive sounds tend to work best. Sounds are loaded as follows:

```
mySound : "sound.aif"
```

Another way to create a pattern is to specify the rhythmic repetition of an existing pattern through bracket syntax. For example, the following pattern definition plays mySound as two eighth notes at *fortissimo*, a quarter-note rest, one quarter note at *mezzo-forte* and four sixteenth notes at *pianissimo*:

```
myPattern : mySound[e.ff, e.ff,  
              q.n, q, s.pp, s.pp, s.pp, s.pp]
```

Patterns can also be nested. In the following example, *myPattern* is played twice:

```
myNested : myPattern[w,h]
```

Each time the pattern is played, its note durations are stretched or compressed to match the target duration. In this example, the patterns remain unchanged for the whole note iteration and are halved for the half-note iteration.

The last but most important way to create a new pattern is to transform or combine existing patterns. Thirteen operations currently supported. These operations were chosen because of their importance in studies on improvisational interaction and musical pattern manipulations. The syntax for all transformations follows this example:

```
pattern1 : sound1[w,h,h]
pattern2 : sound2[q,q,q,q]
myConcat:cat(pattern1, pattern2)
myTrunc  : trunc(pattern2, 2)
```

The operation `cat` places two patterns in succession, so `myConcat` combines patterns based on two different audio-file sources. The operation `trunc` removes the final `n` items from a pattern, so `myTrunc` removes the final two quarter notes from `pattern2`.

Patterns are not played immediately upon creation. Instead, musicians must use an LOLC scheduling expression to determine when and how they play. Typically, patterns are scheduled for playback at the next beat or measure:

```
play myPattern @nextBeat
play myPattern @nextMeasure
```

LOLC supports additional scheduling methods: `preview` will preview the pattern over headphones and `loop` will play it repeatedly:

```
loop myPattern @nextMeasure ~16
```

This example loops the pattern sixteen times. Additional mechanisms support shorthand notations, scheduling at specific points of time in the future, and prematurely killing playback. Scheduling commands are shared with other clients via the chat-inspired interface (Figure 1), but they are played solely on the local client machine.

Through LOLC's novel text-based, networked architecture for improvisatory music-making, we are exploring new mechanisms for musicians to collaborate with each other to create music in performance as well as new techniques for revealing their creative processes to audiences.

### ACKNOWLEDGMENTS

LOLC is supported by a grant from the National Science Foundation as part of a larger research project on musical improvisation in performance and education (NSF CreativeIT #0855758). Additional documentation, software downloads, and videos are available at <http://www.jasonfreeman.net/lolc/>.

### REFERENCES

1. Freeman, J. and Van Troyer, A. Collaborative Textual Improvisation in a Laptop Ensemble. *Computer Music Journal* 35, 2 (Summer 2011), 1-14.

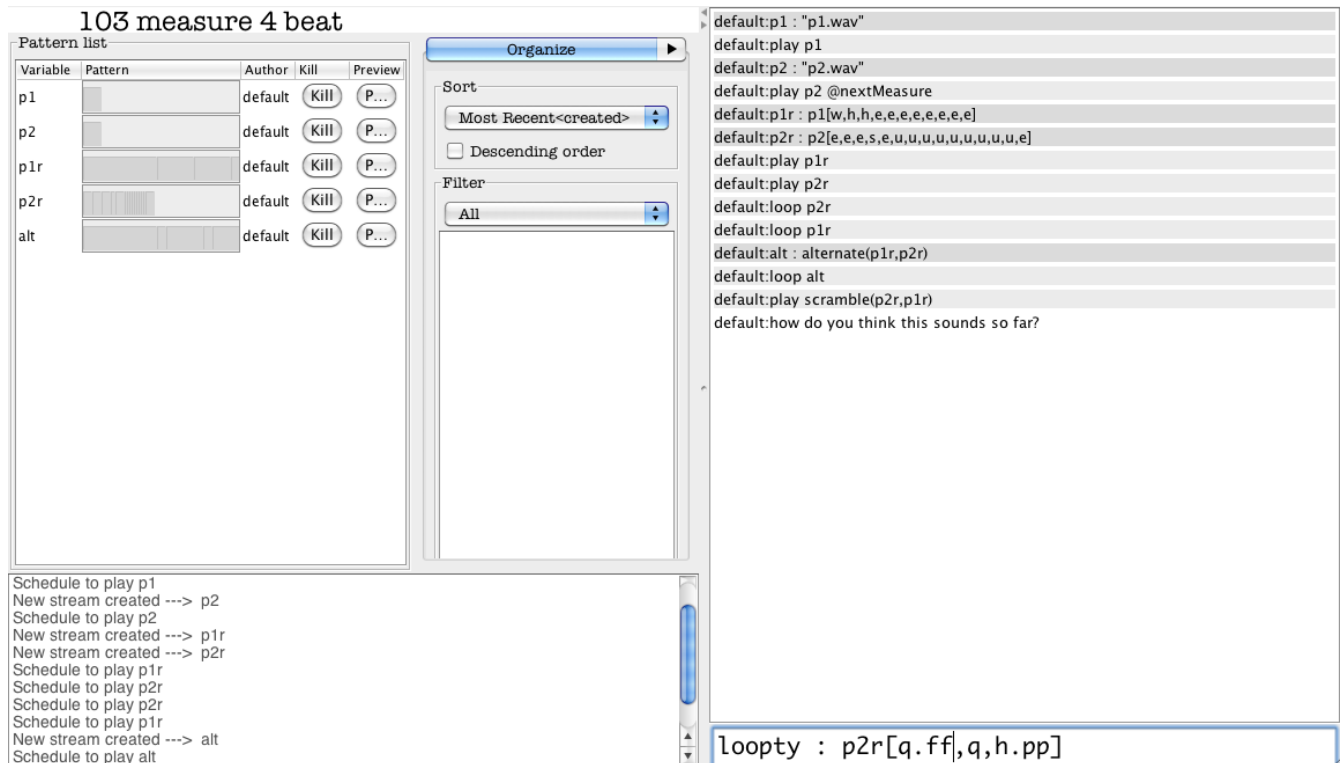


Figure 1. Musicians use an Integrated Development Environment that is inspired by instant-messaging clients to create and share music and messages with LOLC.