Jason Freeman, Akito Van Troyer, Andrew Colella, Sang Won Lee, and Shannon Yao

LOLC (2010)

for laptop orchestra

Performance Guide

About the Piece

In LOLC, the musicians in the laptop orchestra 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. 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). License for JSyn synthesis library provided by Phil Burk of Mobileer Inc.

About the Creators

Jason Freeman, an assistant professor of music at Georgia Tech, oversees all development of LOLC. Together with Akito Van Troyer, he designed the LOLC language and its network and development architecture. Jason also developed the real-time music notation architecture and language constructs for LOLC.

Akito van Troyer, a recent graduate of Georgia Tech's MS program in Music Technology (and current student at the MIT Media Lab), had the initial inspiration to combine live coding and laptop orchestra that led to LOLC. He implemented the initial version of LOLC, including its user interface, compiler, sound engine, and video projection.

Andrew Colella, another recent graduate of Georgia Tech's MS program in Music Technology, refined implementation of the sound engine and real-time notation engines.

Sang Won Lee, a current MS candidate in music technology, has worked on back-end compiler and rendering support for real-time music notation, on the user interface, and on user studies of LOLC.

Shannon Yao, an MS candidate in digital media at Georgia Tech, created new visualizations of LOLC patterns and performances both for the video projection and the client application.

Duration

The duration is entirely up to you. Practically speaking, we expect most performances to last between 10 and 15 minutes.

Instrumentation

Any number of laptop musicians can play LOLC. Practically speaking, we expect most performances to include between 5 and 10 musicians.

Technical requirements

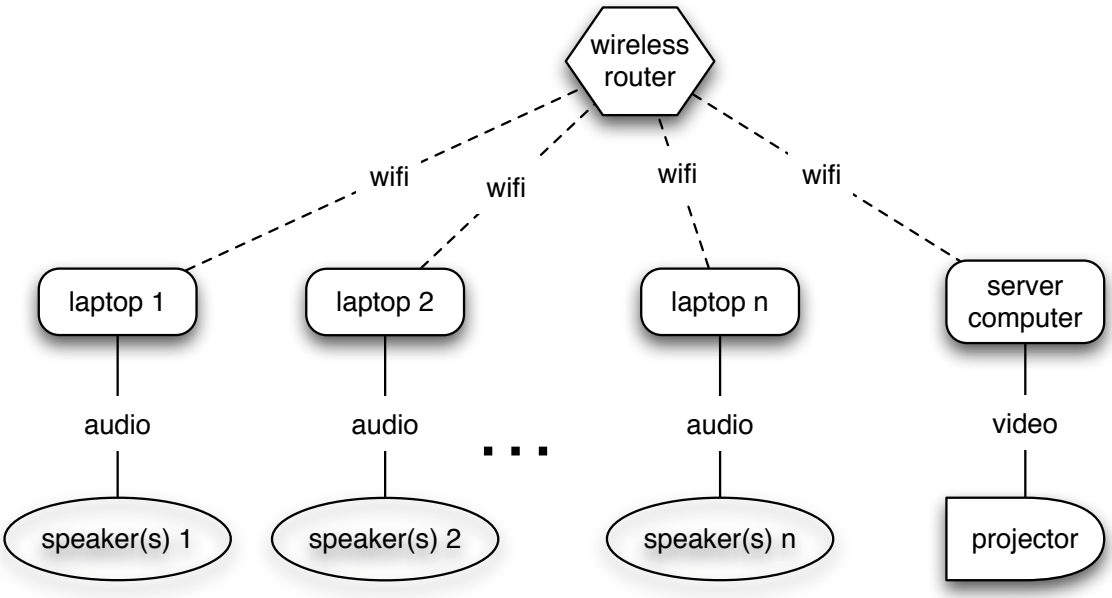
Each musician in the ensemble must have:

- A Mac or Windows laptop
- One or more speakers connected to her laptop in a mono or stereo configuration
- An external audio interface (optional)
- Headphones (optional)

The following additional equipment is required:

- A wireless network router. The router need not be connected to the Internet.
- A Mac or Windows computer that acts as the server. We prefer that this be a separate machine, but if need be, one of the laptops in the ensemble can also operate as the server.
- A projector (optional).

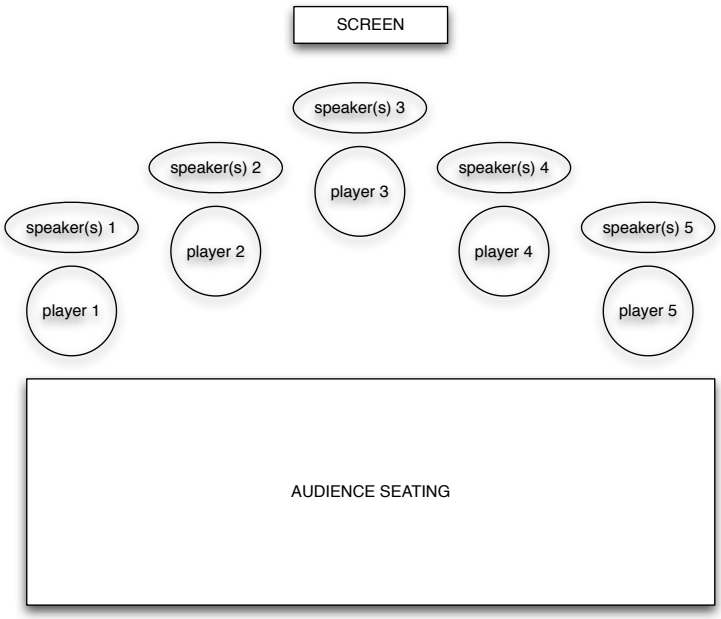
Wiring diagram



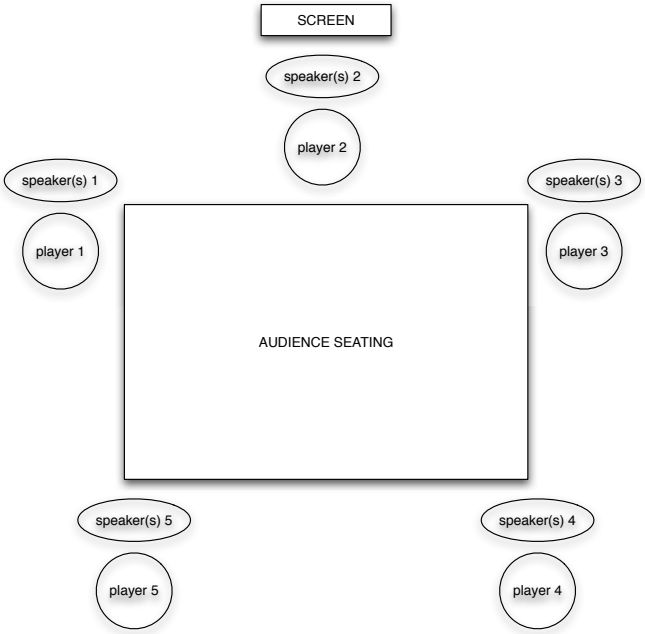
Performance Layout

The arrangement of musicians in performance is entirely up to you. We only ask that each musician be located as close to her respective speaker(s) as possible. Here are two setups that we have found to be effective.

Semicircle: The musicians sit in a semicircle on stage, with the projection behind them.



Surround: The musicians set up all around the performance space, surrounding the audience, with the projection remaining in the front.



Installing the LOLC Software

All musicians in the ensemble must run the LOLC Client software on their laptops. The server computer must run the LOLC Server software.

To install and run this software, follow these steps:

1. Everyone must install a JMSL license (see instructions below).
2. Download the LOLC software from http://www.jasonfreeman.net/lolc/.
3. Double-click the download to decompress it.
4. Double-click the client or server application files to launch. It may take a minute for the software to launch; please be patient.

Installing the JMSL License

LOLC was developed using Nick Didkovsky's Java Music Specification Language (JMSL) and Phil Burk's JSyn audio architecture. You must install a license for JMSL on your laptop in order for LOLC to function properly.

To install the license, follow these steps:

1. Register for a free 30-day JMSL license at http://www.algomusic.com/algomusiclicense/request_jmsl_demo_license.html
2. Download the JMSL software.
3. In the JMSL folder, double-click the JMSL_License_installer.jar file.
4. Drag the JMSL.lic file you receive via e-mail onto the gray area to install the license.

If you wish to continue using LOLC after 30 days, please contact us at http://www.jasonfreeman.net/contact/ to get a free, non-expiring license to continue using JMSL with LOLC.

We apologize for the hassle of installing this license and are currently working with Algomusic on a new solution that will eliminate these additional installation steps.

Learning LOLC

The best way to learn LOLC is to read through the LOLC tutorial and try out the examples it includes. The LOLC tutorial is included with the LOLC client. After launching LOLC, choose Help->Tutorial from the menu.

The application also includes a quick command reference (Help->Reference).

Configuring LOLC

Individual Practice

When using LOLC by yourself, there is no need to launch the LOLC server. Just launch the LOLC client, close the network dialog box, and start making music. (Note, though, that initialization variables in startup scripts only load if you are connected to a server.)

Network Configuration

When using LOLC in an ensemble, follow these steps:

1. Connect all computers (client and server) to your local wireless router.
2. Note the IP address of the server computer. The IP address is listed in System Preferences->Network on Macs.
3. Launch the LOLC server on the server computer.
4. Launch the LOLC client on each of the musicians' computers.
5. In the network dialog box in each LOLC client, enter the server's IP address (see step 2), a username of your choice, and (if using projection) a unique identifying color for your user. Then hit connect.
6. Verify the configuration by checking the "Performers" list in the LOLC server window or on the projection. If all usernames are listed, then configuration is complete.

If an LOLC client cannot connect to the server, try these troubleshooting tips:

- Make sure that the client machine is running on the same wireless network as the server.
- Make sure that you entered the IP address of the server correctly.
- Make sure that you are running the same version of LOLC as the server.

LOLC client Options and Features

Chat Pane

The chat pane is the main interface through which to interact with LOLC in performance. You can type in LOLC variable definitions and scheduling operations in the bottom-right text field. To execute them, hit shift-enter. To enter a chat message that the other musicians can read, hit enter.

The chat pane includes several features to help you type faster. Clicking on any previous message in the chat view will copy it into the text field so you can quickly edit and re-execute it. Pressing the up arrow will cycle through a history of the previous code you have typed. And as you type, LOLC auto-completes your words: just click on a suggested word or hit tab to accept the default suggestion.

Pattern Library Pane

The pattern library helps you to understand the variables created within the ensemble and to borrow and transform material created by others.

The Pattern List shows all currently-defined variables. The colors of blocks indicate the sound files they play, their brightness indicates their dynamic, the lengths of the blocks indicate the rhythmic duration of each event within the pattern, and their height indicates spectral centroid (how high or low the sound is). Since this list can quickly become long, numerous options on the "Organize" tab help you to sort and filter the list of variables to find specific patterns based on their author, the sounds they use, when they were created, and other criteria.

The "Currently Playing" tab helps you track the music playing within the ensemble, showing which specific variables are currently playing.

The "File names in use" tab explains the color coding of patterns in the pattern list, linking colors to the sounds they represent.

Audio Configuration Dialog

The Audio Configuration dialog, which is displayed on startup, lets you assign audio channels to the main audio output and to the preview output.

We encourage you to set up preview functionality so that you can hear patterns on headphones before scheduling them for playback over speakers. You can do this if you use a multi-channel audio interface, or on the Mac you may also create an aggregate device in Apple's Audio MIDI Setup in order to use the built-in headphone jack for previewing.

Note that to change your audio settings, you must restart LOLC.

Console Pane

The console displays status and error messages. Pay special attention to the console for syntax errors when you execute pattern definition and scheduling expressions.

Sounds

LOLC ships with a default set of 80 sounds, created by Jeff Snyder, for use within the environment.

We encourage you to use your own sounds with LOLC. You can import any sound files into the system by following these steps:

1. Save all of your sound files as WAV files.
2. Copy them into the LOLC sound folder (lib/sound inside of the main LOLC folder on Windows; or right-click the application to Show Package Contents->Contents/Resources/lib/sound on Mac). You must copy the sound files onto each client machine and onto the server.

Music Notation

Recent versions of LOLC include preliminary support for real-time music notation, supporting performance environments in which laptop musicians generate notation on-the-fly for instrumental musicians to sight-read in performance. For more information about this feature, refer to the LOLC tutorial file.

LOLC server Options and Features

Visualization

The main window of the server shows a visualization of LOLC performers, status, code, collaboration, and chat messages. We encourage you to project this screen so that audience members can see it during performance.

To show the visualization in full-screen mode, hiding the dock and menu bar, hit the space bar.

In performance, you should configure LOLC so that the placement of each performer on screen corresponds to her physical location in the performance space. To do so, edit "stageLayout.xml" inside of the main LOLC folder on Windows (or right-click the application to Show Package Contents->Contents/Resources/lib/code on Mac). Follow the example file to create an entry for each performer that identifies their user id and x/y placement on screen.

Sounds

LOLC ships with a default set of 80 sounds, created by Jeff Snyder, for use within the environment.

We encourage you to use your own sounds with LOLC. You can import any sound files into the system by following these steps:

1. Save all of your sound files as WAV files.
2. Copy them into the LOLC sound folder (lib/sound inside of the main LOLC folder on Windows; or right-click the application to Show Package Contents-

>Contents/Resources/lib/sound on Mac). You must copy the sound files onto each client machine and onto the server.

Startup

When LOLC server launches, it loads a startup script to define variables for use within the environment. When clients connect to the server, they receive this same set of variables.

By default, the startup script defines variables corresponding to each of Jeff Snyder's default sound files (e.g. b1 : "b1.wav"). In theory, the startup file can include any legal LOLC variable expressions.

To modify the contents of the startup script, edit the startup.lolc file in any text editor (including the LOLC editor). It is located in lib/code/ (inside the application package on Mac or the main folder on Windows).

It is not necessary to copy new versions of startup.lolc to each client machine. The startup file only needs to reside on the server.

Console

The console shows server status, error messages, and network connection activities (e.g. client connects and disconnects).

Config

The config dialog, which is displayed on startup, provides control over basic server functions. You can change the tempo (bpm), meter (beats per measure), and beat unit (by default a quarter note).

Menu

"Reload All Variables" will clear all variables out of the server and reload the default variables from the startup file.

"Dump actions to file" will log all code and chat messages to a text file. Select it once to begin recording and a second time to stop recording. The file is stored in lib/stash/.

Multi-Location Performances With LOLC

While we prefer that all musicians in the ensemble are co-located in a single performance venue, LOLC does support performances in which the musicians are divided among two or more performance locations.

To set up a multi-location performance, follow these steps. In this example, we assume two locations: location A and location B.

1. At both locations, make sure that all computers are connected to the Internet, not just to a local-area network.
2. Set up the server computer in location A only. The server (and any firewall it is connected to on the network) must allow incoming connections on port 36785.
3. At both locations A and B, set up an extra LOLC client computer, called a "ghost" computer, that will reproduce the sounds created by users at the remote location(s). When launching the "ghost" LOLC client at location A, enter a comma-delimited list of all the musicians at location B in the network dialog box. When launching the "ghost" LOLC client at location B, enter a comma-delimited list of all the musicians at location A in the network dialog box.
4. Musicians from both locations A and B should enter the IP address of the server at location A into the network dialog box.

With this setup, all of the sounds created at location A will be reproduced on the "ghost" computer at location B, and all of the sounds at location B will be reproduced on the "ghost" computer at location A. Players across both locations will see the same command and chat messages and will see the same patterns in the pattern library. The measure / beat counters will be synchronized across all machines in both locations.

Please note that the projection created by the server will only be available in location A, where the server is physically located. If you wish to reproduce the server's projection at location B, we recommend you use a standard screen-sharing tool such as VNC to display the server's screen in location B.

Preparing a Performance of LOLC

There is no "standard" way in which to perform LOLC. That is why we call this document a performance guide instead of a score. It is up to your ensemble to decide upon the content and structure of your performance. And since LOLC is about collaborative improvisation, we expect that content and structure to vary every time you perform the work.

Sounds and Startup Files

Here are some key questions to think about as you determine the core musical content of your performance:

- What audio files will you use? Will you use the default sounds that come with LOLC or will you create your own? Will you use all of the default sounds or only a particular subset? Will you record your own sounds, sample them, synthesize them, and/or find them in databases such as Freesound?

- What variables will you predefine before the performance? Will you solely define shortcut variables for the audio file names? Or will you also define core rhythmic material and larger motives that you will use during the performance?

See previous sections for details on how to use your own sound files in LOLC and to predefine your own variables in the startup script.

Musical Structure and Roles

As you rehearse the piece, we encourage you to experiment with a variety of different organizational strategies. See what works best for your ensemble.

Here are some key questions and some ideas to get you started:

- Is there a pre-determined large-scale structure? For instance, you may define certain sections based on their audio, rhythmic, or textural content and then move from section to section either gradually or suddenly. You may create a gradual build-up of texture by starting with a solo musician and adding additional members of the ensemble one by one. Or you may choose to make no such decisions in advance and just see where the improvisation leads you.

- Will different musicians take on different roles within the ensemble? For instance, some musicians may focus on creating new variables while others focus on scheduling existing variables for playback. Some musicians may focus on creating looping background textures while others focus on creating episodic foreground material. Some musicians may focus on percussion and bass while others focus on melodic motives. And one or more musicians may have "conducting" roles, cueing the ensemble with chat messages to move from section to section or adopt particular strategies and roles during the performance.

Common Questions

Everything seems to work fine but I do not get any sound. What's going on?

Be sure you followed the instructions above and installed the JMSL license.

I am connecting my laptop's built-in audio output to my speakers — rather than an external audio interface — and I hear no sound. What's going on?

In some cases, you may need to switch the playback channel in Lolc-client's Audio Configuration window to channel 1 (instead of the default channel 0) when using built-in audio.

Some of the keyboard shortcuts don't work. What's going on?

Some of the keyboard shortcuts, such as open and save, only work when a Code Editor window is in the foreground. Other command-key shortcuts, such as kill, work no matter what window is in the foreground.

The visualization on LOLC Server looks funny. What's going on?

The visualization needs to be in full-screen mode to display properly. Hit the space bar to toggle between the small view and the full-screen view. Also, we have experienced some intermittent problems with full-screen display of the server visualization on Windows. We believe we have resolved these problems but want to hear from you if anything doesn't look quite right.

Why doesn't LOLC work on Linux?

LOLC may indeed work on Linux, but we have not tested it and do not support it.

Getting Help and Getting in Touch

Please do not hesitate to contact us with technical questions, bug reports, or other issues. We would also love to hear about your experiences with LOLC and the music you perform with it.

We can be reached via:

http://www.jasonfreeman.net/contact/

Most importantly, have fun with LOLC!

Jason Freeman, Akito Van Troyer, Andrew Colella, Sang Won Lee, Shannon Yao

Atlanta, Georgia

April 22, 2010

rev. January 31, 2013

Appendix A: A Sample Musical Structure for a Performance

We have used the general structure outlined below in several performances of LOLC. We provide it here as a potentially useful starting point for your rehearsals.

Tempo: 120 Meter: 4/4 Approximate duration: 300-350 mm.

Description	Example Code Fragments

Additive Entrance
One by one, each player loops a new pattern based on a single "p" sound. There is at least a two-measure delay between successive entrances. Each pattern's length is an integer multiple of 4 beats.
	`ex1 : p1[q.f,q,q,e,e]`
	`loop ex1 @ m`
	`ex2 : p2[e,q.p,eh]`
	`loop ex2 @ m`
	`ex3 : p3[wh.ff, h.p]`
	`loop ex3 @ m`

Transform and Combine
Players may use any operations they want to transform and combine patterns or create new ones. But all patterns must continue to use only "p" sounds and must have a length that is an integer multiple of 4 beats. Gradually increase intensity and density.
	`tc1 : shuffle(cat(ex1, ex2, ex3))`
	`loop tc1 ~4`
	`tc2 : merge(tc1, cat(pp4, pp5, pp6))`
	`loop tc2 @ 73`
	`tc3 : spread(ex3, 4, 7, 0.5)`
	`loop tc3 ~8`

More Categories (Optional)
Players add in one or two other categories of sounds (e.g. s, w). Total pattern duration must still be a multiple of a whole note. Gradually increase intensity and density to first peak.
	`mc1 : cat(s2[h], s3[eq], s4[e])`
	`play mc1 @ m`
	`mc2 : s5[e,e,e,s,s,qe,t,t,t,t]`
	`play mc2 @ m`
	`mc3 : alternate(mc1, mc2)`
	`loop mc3`

Additive Bass
One by one, each player loops a new pattern based on a "b" sound. There is at least a two-measure delay between successive entrances. Patterns can have any length and need not line up with measure boundaries. As soon as the first player has started their loop, all other players issue a kill * command to immediately stop playback of material from the previous three sections.
	`kill *`
	`vv1 : b1[ww.f]`
	`loop vv1`
	`vv2 : b4[h,hs.p]`
	`loop vv2`
	`vv3 : b2[qe,qs.pp]`
	`loop vv3`

Anything Goes
Players may use any sounds and operations they wish. Gradually increase intensity and density to second peak.
	`ag1 : s5[u,u,u,u,u,w.n]`
	`ag2 : spread(cat(tp4,tp5,tp6), 3, 7, 0)`
	`ag3 : rotate(ag1, 2)`

Ending All players agree via chat message on an ending measure and create a subito ending by killing all of their patterns on that measure.	`kill * @ 350` `kill *`