Making Music with Linux Multimedia Studio

MUSIC BUILDER

Linux has truly started to compete with Windows and MacOS as a platform for professional sound applications. Linux Multimedia Studio (LMMS) is a Linux sound tool that packs a variety of impressive features into a neat bundle. **BY TOBIAS DOERFFEL**

www.photocase.de

Not too long ago, multimedia and Linux were separate worlds, but – thank goodness – the situation has changed dramatically. Linux no longer lags behind Windows and MacOS even in the field of music productions: Audacity brings sound editing power to Linux, Rosegarden handles Midi, and Hydrogen gives you the ability to put beats and rhythms together. Last but not least, Csound adds customized software synthesizers and effects.

As you will no doubt have noticed, there is a separate tool for each task, although many users would prefer an integrated system instead of individual components for really creative work. This led to the launch of the Linux Multimedia Studio (LMMS) in spring last year. LMMS combines all the functionality of the programs I just mentioned in a single interface that looks a bit like Fruity Loops on Windows.

Installing and Setting Up LMMS

You man be able to find binary packages for your distro at the LMMS website [1]. Users with distros that aren't represented by the binary packages will need to build LMMS from the source code to install it. You can start off by downloading the source code from [1].

As LMMS is a realtime application that needs a lot of CPU power, you should enable any appropriate optimization options. The “Major Compiler Options” box lists important compiler options for the Intel platform.

You can use *configure* options to disable some LMMS features and make the program even quicker. For example, it doesn’t make much sense to build surround support into LMMS if your sound card doesn’t support the option. Without surround support, LMMS only needs to handle half the volume of data in some places, and this pays dividends in the form of much quicker performance.

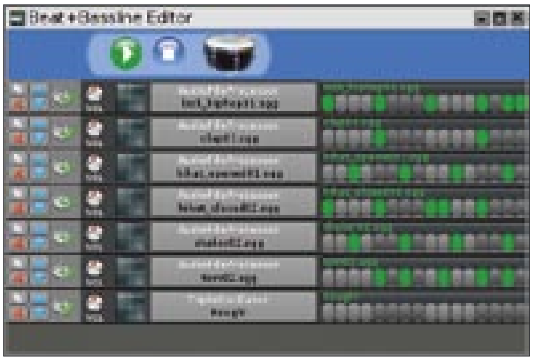

Figure 1: In LMMS, you can put beats together with a few clicks. The program comes with a collection of samples you can use for your own songs.

You can disable surround by specifying *--disable-surround ab*. And if you can do without genuine realtime, because you do not have a midi keyboard, you can specify *--with-latency=VALUE*, where *VALUE* is a power of 2, that is, 64, 128, or 256. The default value is 256, which gives you a latency time (that is, a delay between event and sound output) of 6 milliseconds. The smaller the value, the lower the latency, and the closer to realtime you get.

To specify all these options for the build, you could configure the source code with the following command: *./configure -prefix=/usr CXXFLAGS="-O2 ..."*

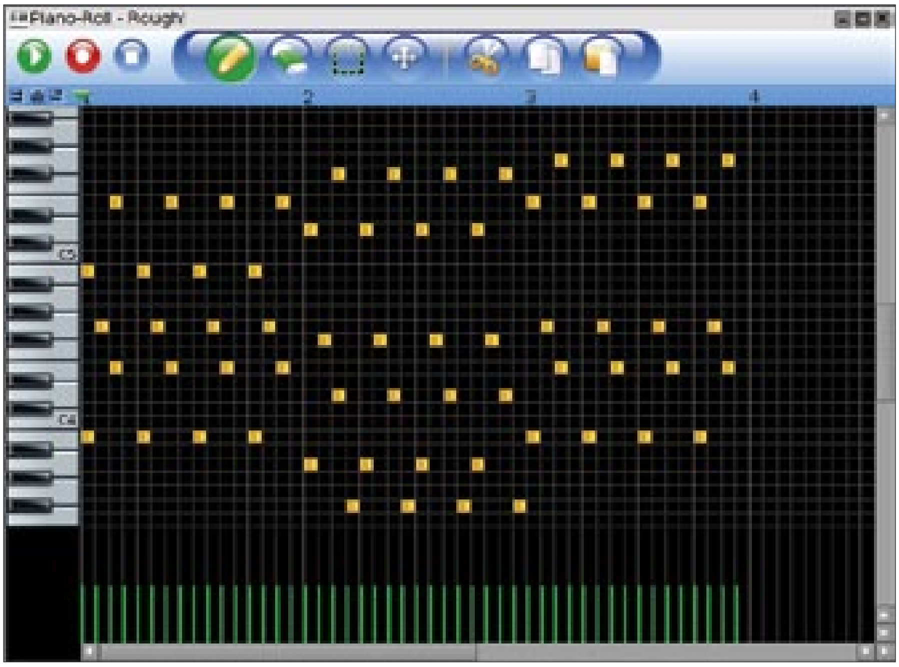

Figure 2: Using the Piano Roll to edit basslines. Clicking allows you to add, remove, or change the values of notes.

[--disable-surround] [--with-latency=VALUE]. Then run *make* to build an LMMS binary and *make install* to install the required files on your system.

make rpm gives you an RPM package, typically below */usr/src/*. You can then install the package by entering *rpm -i FILE.rpm*. However, this disables the options that you set at the *configure* stage; instead, you have to set the environmental *RPM_OPT_FLAGS* variable.

After building and installing, you can type *lmms* to launch the program. This launches a wizard to walk you through the initial setup.

First Steps

When you launch the program, the LMMS interface appears with the following areas and windows. The most important window, the Song Editor, contains the playlist with the various tracks and a toolbox. The Beat + Bassline Editor helps you create beats and basslines, which you can drop into the Song Editor later. The Project Notes window is where you can add notes to describe the project. On the left is a sidebar with various task areas, such as the Project and Sample browser.

Let's start off by creating a simple beat. To do so, open the Sample Browser (*My Samples*), open the *drums* folder, and double click the *kick_hiphop01.ogg* file. A new track appears in the Beat + Bassline Editor, along with a new channel containing the selected sample. The *AudioFileProcessor* plug-in, visible in the same window, is used for loading samples. *AudioFileProcessor* allows you to set up a number of output parameters.

Now is a good time to try out the virtual piano keyboard, to see if you are getting any sound output and if LMMS has successfully loaded the sample. If you can hear a few thudding noises at this point, everything is well, and you can close the channel window.

Time to press on with our first steps. To create steps, click the gray buttons in the corresponding pattern in the Beat + Bassline editor; this changes the gray buttons to green. You can then click on the play button above the tracks to listen to the initial results.

Of course, a kick can get monotonous, but you can double click to add snares, hi-hats, and so on. Figure 1 shows an example of a completed beat.

Melodic Moments

We want LMMS to give us some real music rather than just a couple of drum samples. Luckily, LMMS has its own software synthesizer, dubbed Triple Oscillator, which comprises three oscillators, filters, envelopes, and low frequency oscillators (LFOs).

As newcomers are typically overwhelmed by this feature-richness, LMMS defines a few presets. Preset files contains specific settings for a single channel. They store information on the plug-in used (such as Triple Oscillator, Audio File Processor), the settings, and the filters and envelopes.

Loading a preset avoids the need to tinker with complex sound generation parameters. At the same time, presets are the ideal jump-off point for your own sounds; you can twiddle the buttons to get them just right. To access the preset browser, click on the yellow asterisk on the left-hand side. Each plug-in resides in its own folder.

Before we go on, let's set up a new bassline. To do so, click the *Add Beat/Bassline* button in the Song Editor toolbox, and then open the *Rough!* preset for this example.

As we will be doing more than just creating beats with this preset, we need to edit the pattern in the piano roll. You can double click the pattern (above or below the step buttons) in the Beat + Bassline Editor to open the piano roll tool. Once in the Piano Roll (Figure 2), you can simply click to add notes,

Major Compiler Options

- *-O2* or *-O3*: This option specifies the optimization level the compiler should use. For newer processors with lots of cache, try using *-O3*.
- *-mcpu=IHRECPU*: This tells the compiler which processor it should optimize the code for. The possible values for *YOURCPU* are: *pentium2, pentium3, pentium4, k6, k6-2, k6-3, athlon, athlon-tbird, athlon-4, athlon-xp*. The GCC manpage has a few other options. Check */proc/cpuinfo* to find out which CPU you have.
- *-msse, -msse2*: This tells the compiler to use SSE commands for floating point operations. This considerably improves the performance of LMMS, however, only the current crop of AMD and Intel CPUs (Pentium 3 or later) actually support these commands.
- *-mfpmath=sse,387*: If your processor does support SSE instructions, you can use this option to enable the SSE unit and the internal co-processor.

More adventurous readers looking for more power might like to try the Acovea [2] program, which uses genetic algorithms to discover optimum tuning options.

change their positions, or change their lengths. Drag the light end of the note to do this. Right clicking with the mouse removes individual notes.

In the Song Editor, you can now link up the beat and the newly created bassline. Left click to create playlist objects that define when and for how long a beat or bassline will play.

As shown in Figure 3, the playlist can also comprise channels (the Triple Oscillator is shown in the figure.) This makes sense if your patterns do not contain basslines that continually repeat, but genuine melodies that occur just once in the whole song. You can left click on a channel track to create a pattern of this type, and then double click to edit.

Sample Tracks

Besides the channel and Beat/Bassline tracks, LMMS also supports sample tracks. You can use these to create longer samples such as raps or guitar solos that you have recorded. The whole sample track technology is still under developing, and functionality is restricted to loading samples at this time of writing. In future, LMMS will have recording and simple cutting functions.

Envelopes, LFOs, Filters, Chords

The Channel window (Figure 4) has more than just one plug-in: besides the *Plugin* tab, there are *Envelopes/LFO/Filter* and the *Arp/Chord* tabs. *Envelopes/LFO/Filter* (Figure 4) allows you to define a filter for an active channel and set parameters such as the cutoff frequency and resonance. You can control these two parameters using envelopes and low-frequency oscillators (LFOs).

Envelopes represent the temporal progression of the value in question, for example, quick attack and slow decay. The bigger the value the *Amount* button is set to, the more the envelope affects the original value. A low frequency oscillator changes the value periodically. You will need to experiment with this technology to get used to the effects you can create.

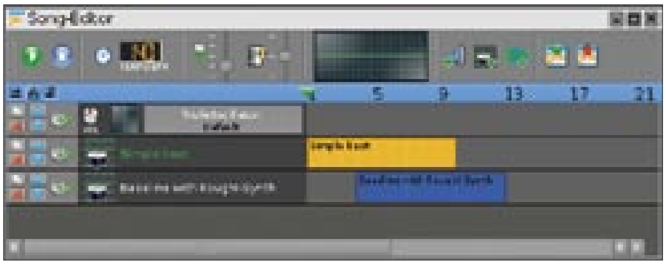

Figure 3: The Song Editor combines beats and basslines to form a playlist.

The *Arp/Chord* tab helps you define settings for automatic arpeggios and chords. If you select a major chord and a range of two octaves for a tone, LMMS can automatically add the matching major chord over a range of two octaves. The arpeggio function allows you to play the individual notes in the chord sequentially.

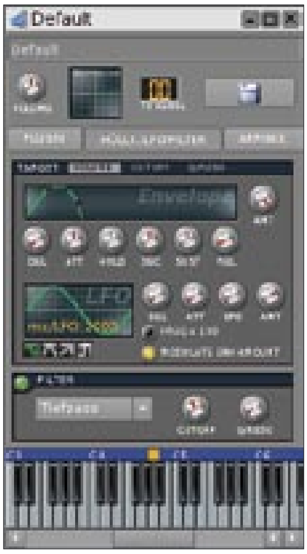

Figure 4: Each channel can be controlled using envelopes, low frequency oscillators, and filters.

Pattern freezing allows users to create complex, polyphone songs, even if your CPU may not have enough power to play them live. Clicking on *Freeze* in the context menu for a pattern tells LMMS to play the pattern once internally and store the results in a buffer.

When you play the song live, the software will not attempt to recalculate the data for this pattern; instead it will just read the buffer and patch it into the output buffer. Changes to the frozen pattern are not implemented until you unfreeze and refreeze it.

You can also control LMMS via a midi keyboard. Pressing a key on the keyboard tells LMMS to play the matching note on the current channel. Additionally, you can map any LMMS button to your keyboard's pitch-wheel by selecting *Map to MIDI device* in the context menu for the button. The button will respond to pitch-wheel commands after doing so, but make sure you set the pitch range to maximum on your keyboard.

After completing a song, you can export the finished product to a Wav or Ogg file. To do so, select *Project* | *Export* or press [Ctrl] + [E]. After entering a filename and specifying the format, a window with options for the song export appears. Now click on *Export* to tell LMMS to write the song to the output file, which you can then burn onto a CD or play back in your favorite music player.

Big Plans

The LMMS roadmap is full of major extensions to the program. For example, LMMS does not give you any options for applying effects (reverb, echo, distortion, chorus, equalizers, and filters, just to name a few.) Work on effect management is scheduled to start following version 0.1.0, and will include a LADSPA interface among other things. LADSPA (Linux Audio Developer's Simple Plugin API) gives audio software developers a standardized interface for effect and synthesizer plugins.

A system that would allow users to control all buttons and switches in LMMS dynamically is equally as important. The LMMS developers are currently looking for a way to give users the ability to automatically modify the cutoff frequency of a filter by drawing a curve. Additionally, it might be possible to control periodic detuning of an oscillator by applying an LFO.

As LMMS still has a fairly restricted sound spectrum, more plugins are planned for the future. Besides a LADSPA plugin hoster, the developers are also working on a VST hoster, which would allow you to use VST plugins.

The LMMS project is still in development, and the roadmap for these features is still fairly vague. In contrast, good progress is being made on a traditional note editor. LMMS already supports MIDI file import, and it looks certain that LMMS will be able to import other file types in the near future. ■

INFO

[1] LMMS homepage: *http://lmms.sourceforge.net*

[2] Acovea: *http://www.coyotegulch.com*