

## The monthly GNU column

# Brave GNU World



Following on from last month's column, we start by taking a look at a graphical sound editor before moving on to way to keep many versions of an application available, a dancing game and a software directory.

Welcome back to another issue of the Brave GNU

World. This issue picks up where the last stopped. In

the large field of audio applications. **BY GEORG C.F. GREVE**

## LAoE

LAoE [1] is a graphical sound editor by Olivier Gäumann. After working on it since July 2000, he recently made it available under the GNU General Public License (GPL).

Prior to starting this project, Olivier had been looking for an appropriate application that would fit his needs since January 2000, unfortunately without success. So he decided to scratch his own itch and began developing. As he lacked experience in professional music, development encountered some dead ends and difficulties, but these were resolved over time.

The design of LAoE has taken inspiration from the Free Software graphics application GIMP [2] from which it adopted the layer-based concept with numerous plugins and effects. Olivier hopes to create a sound editor that is about as complete as state-of-the-art graphical programs.

LAoE can read mp3, wav, au, aiff and gsm files and supports output in wav and aiff formats. The implementation of a native laoe-format, in which the different layers and settings will be preserved, is planned.

Although the performance of the hardware puts a practical limit to the number of channels and layers, LAoE theoretically supports an unlimited number. To ensure maximum precision, all operations are floating-point based. Besides standard functions like copy, paste, cut and move, LAoE also provides a rather sophisticated and unlimited undo system that allows going back to any former state. Selecting and modifying areas for editing is possible in multiple ways and with different functions.

Besides analysis tools, LAoE contains waveform generators and different functions for amplitude modification. Transduction into and editing in the frequency domain by means of Fast-

Fourier-Transform (FFT) is supported as well as filtering by different parameters and frequencies. Aside from that, the project offers many different effects, e.g. a delayed echo, pitch manipulation, disharmony plugin, time manipulation and so on. Also it is possible to display and edit spectrograms.

Olivier also spent some time improving usability, so LAoE offers "mouse stroke shortcuts" that for instance allow calling the undo history plugin by drawing a simple "u" with the mouse. Should that not be enough, you are of course welcome to provide your own effects and expansions in the form of plugins.

With development being in a fairly advanced stage by now, Olivier was planning to focus more on documentation – currently a short reference/menu card already exists – in order to improve usability and win more users for the project. At the moment, the incoming stream of suggestions and wishes for improvements is pretty strong, though, so it seems possible that further documentation may be somewhat delayed.

The main problem of LAoE is that isn't usable for pieces of music longer than roughly 5 minutes, as the whole piece needs to be read into memory, which provides for strong technical limitation.

Also the programming language used was Java; with all the problems described in detail in last month's issue [3]. So LAoE is still dependent upon a proprietary platform.

Whether the future will bring a Free Java or use of another language – possibly also to avoid technical problems – it can be said that LAoE is definitely a very interesting project that took another important step in the field of professional audio applications.

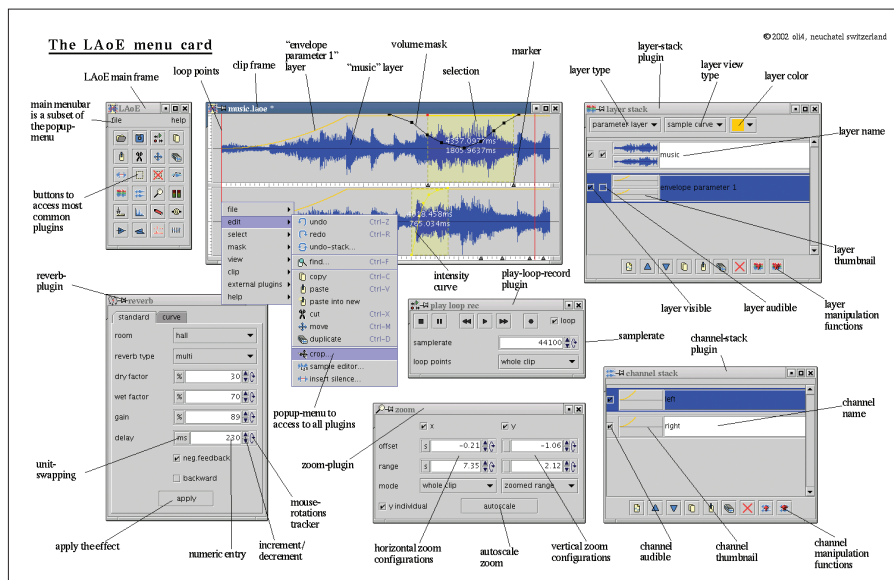


Figure 1: LAoE menu card

Although according to his own sentiment Olivier did not receive a lot of support from others, that does not mean he wouldn't appreciate it now. Quite the opposite, he explicitly is looking for volunteers to help him work on this project.

More information about the project as well as contact details can be found on the LAoE home page. [1]

## Apvsys

Sometimes it is necessary or at least useful to have different versions of the same program installed simultaneously so they can be used alternatively or even at the same time. Possible reasons for this are different versions not being backwards-compatible any more or other programs requiring different versions of the same application.

The different operating systems have different approaches to solve this problem. Under MS Windows this is handled by means of the centralized Registry, which only solves this problem extremely unsatisfactorily.

The operating system Plan 9 has the concept of so-called "views" in combination with union file systems. This allowed storing all applications in the /bin directory and by means of the view, the user can determine which version is being used. Under Unix such a choice / installation is usually done with symbolic links and the environment variables PATH and LD\_LIBRARY\_PATH.

Given the case that there is just a new version to be installed by hand outside the scope of the distribution without harming the maintainability of the system, there are tools like GNU Stow [5],

which can make your life easier (see previous issue [6]). Should several different versions be usable at the same time, this does not help you much, however.

People try to solve these problems with wrappers and shell scripts, but these solutions are often fragile and require intensive care and maintenance. Apvsys [4] by Arnaud Bertrand seeks to provide a better solution.

The different versions of all applications are all installed into different sub-directories and managed by Apvsys. Users only need to add a single directory to their PATH variable, which could also happen automatically.

Afterwards, the APVSYS\_TOOLSPEC\_FILES allow configuring different versions for specific applications and environments. Commandline tools allow checking the used versions and administrating the configuration. If no particular version is specified, the default version is used. All configuration is done by means of simple text files that are administrated through Perl applications. The Apvsys shell itself was written in C.

A few words should be said about the background. Apvsys is the reimplementation of a program written in 2001 by Alex Farell using Perl. This could not be used anymore because Alex Farell had written it during his working time and so the rights were with the employer.

Apvsys made sure not to run into the same problems and is available as Free Software under the GNU General Public License (GPL) Version 2.

By switching to C Arnaud Bertrand sought to increase the speed of the application, while making sure to keep the

format of the ASCII configuration database to make the shift as easy as possible. This seems to have been rather successful, as Apvsys is successfully being used in three big design centers in Belgium and France as part of their production environment since 2002.

So the core of Apvsys can definitely be considered stable, which of course doesn't mean that there are no problems or room for improvement. Sometimes different tools with identical names cause problems and should be treated with care, for instance. Also there are sometimes problems when the paths are hard-coded into the applications.

Future plans for Apvsys are to add a graphical user interface and a packaging system. The decision of whether the packaging system will be chosen from those already existing or be newly implemented is still not clear, though.

In any case, volunteers are very welcome to participate in Apvsys development and possibly make suggestions about the interface or packaging system. Another area in which volunteers can help easily is platform independence. Apvsys is known to run on GNU/Linux, Solaris, HPUX and Cygwin. It should also run on other platforms, but that needs to be tested.

The last suggestion by Xavier Duret, who filled out the Brave GNU World questions on Apvsys, refers to the GNU Hurd. [7] In case this hasn't been done already, he would like to encourage a volunteer to create a Hurd translator based upon a Union filesystem that could provide equivalent functionalities to Apvsys or Plan 9.

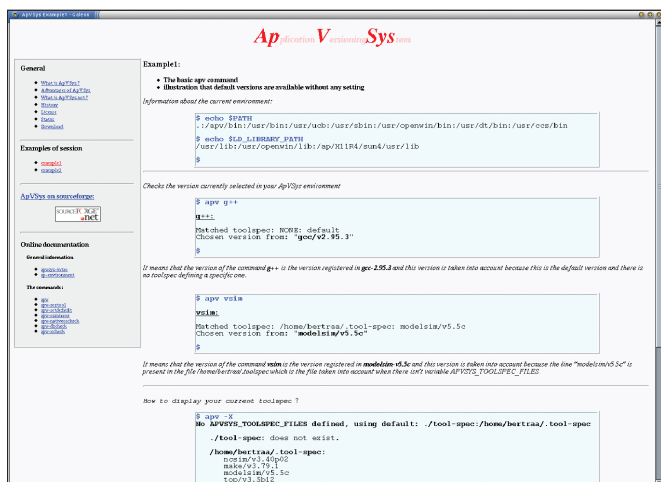


Figure 2: Avsys home page

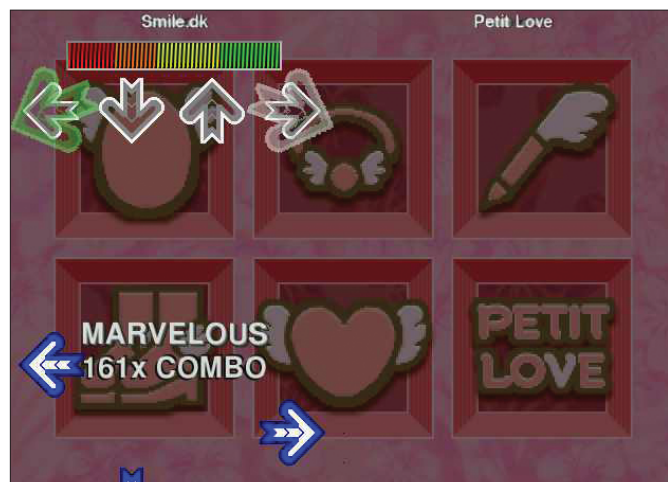


Figure 3: Dancing to the pyDDR beat

## pyDDR

At least in the northern hemisphere, summer is approaching fast. Therefore even geeks should think about getting into shape. To make sure that they don't have to leave their beloved computer for this, there is this next project.

pyDDR [8] is a Python clone of the game DDR ("Dance Dance Revolution"), in which the player under full bodily engagement has to step on the right arrows of a sensor mat at the right time. Sounds simple, and it is. But only if you're watching. Even though these games have not yet spread very far in Europe and the America, they have been popular for some time in Asia and in many arcades people are dancing or, for another game, beating the drums.

Thanks to the work done by Brendan Becker, who implemented pyDDR as Free Software under a license similar to the X11 license, you can now also do this at home and so during waiting time – like a long compilation – do something for the body. Libraries used were the PyGame Module [9] and the Simple DirectMedia Layer (SDL) [10].

Brendan got support for his work through several volunteers like Joe Wreschnig and John Bafford; he'd also like to encourage others to get involved. This includes financial contribution, Brendan emphasizes, as he would rather spend more time on games than doing his "normal" job.

pyDDR was originally written for GNU/Linux, but there are reports that it

is successfully being made to run on Windows NT, XP, 9X as well as BSD, MacOS X and BeOS. Even on a Playstation 2 running GNU/Linux somebody got it to run – this can only be recommended to the technically savvy, though.

Of course the full game experience requires getting a sensor mat, or, if you intend to play against each other, even two. But you can also replace one sensor mat with the keyboard, so it is at least possible to get an idea of the game without having to get new hardware.

But even if you don't believe in physical movement like dancing, the game is still worth a look because it looks very professional with good graphics and sound, providing an idea of what Free Software games can look like.

## Free Software Directory

With a growing amount of Free Software applications, it gets increasingly harder to keep the overview. Portal sites are trying to address this problem.

Unfortunately some portals, like Freshmeat, are only of limited use, as they are not capable of ensuring the quality of their entries and also often mix proprietary and Free Software.

To solve this problem, the Free Software Foundation began working on the Free Software Directory, [11] a database for Free Software, in September 1999 and entrusted Janet Casey with maintaining it.

Every entry in the directory is checked for proper licensing by looking at the

web page and into the source code to make sure that the license is what it seemed to be. Therefore users can place more trust into the information from the Free Software Directory.

In April 2003, the United Nations Education, Scientific and Cultural Organization (UNESCO) [12] joined forces with this FSF project and supports it officially, so now it is the "FSF/UNESCO Free Software Directory."

If you are looking for solutions or would like to know whether an application exists, the Free Software Directory offers a good reference point. Authors of new and existing projects should make sure their work is listed correctly in this directory. Of course, you should still contact Brave GNU World. [13] with project information as well.

## End

If you have questions, comments, ideas or suggestions for cool projects, please don't hesitate to send me email. [13] ■

The screenshot shows the FSF/UNESCO Free Software Directory website. At the top, there is a search bar and a logo of a ram. Below the search bar, it says "2,209 packages indexed". On the left side, there is a navigation menu with categories like Accessibility, All GNU Packages, Audio, Business and Productivity, Database, Design and Manufacturing, E-commerce, Education, Email, Games, Graphics, GUI, Hobbies, Live Communications, Localization, Miscellaneous, Network Applications, Printing, Science, Security, Software development, Software Libraries, System administration, Telephony, Text creation and manipulation, Video, and Web Authoring. The main content area contains a description of the directory, a list of licenses, and a section titled "Ten most recently updated entries in the directory..." with entries like abcm2ps, AutoUpdate, Ayzm, Bugzilla, Caltree, Checkuris, Chronos, Circus Linux, and CvsKnit.

Figure 4: Maintaining an overview with the Free Software Directory

## INFO

- [1] LAoE home page: <http://www.oli4.ch/laoe>
- [2] GIMP home page: <http://www.gimp.org>
- [3] Brave GNU World, Linux Magazine, Issue #32 / July 2003, p91, <http://brave-gnu-world.org/issue-49.en.html>
- [4] Application Version System (Apvsys) home page: <http://www.apvsys.org>
- [5] GNU Stow home page: <http://www.gnu.org/software/stow/>
- [6] Brave GNU World, Linux Magazine, Issue #19 / April 2002, p88, <http://brave-gnu-world.org/issue-36.en.html>
- [7] GNU Hurd home page: <http://www.gnu.org/software/hurd/>
- [8] pyDDR home page: <http://icculus.org/pyddr/>
- [9] PyGame home page: <http://pygame.org>
- [10] Simple DirectMedia Layer (SDL) home page: <http://www.libsdl.org>
- [11] Free Software Directory: <http://www.gnu.org/directory/>
- [12] United Nations Educational, Scientific and Cultural Organization (UNESCO): <http://www.unesco.org>
- [13] Home page of Georg's Brave GNU World: <http://brave-gnu-world.org>

Send ideas, comments and questions to: [col-umn@brave-gnu-world.org](mailto:col-umn@brave-gnu-world.org)