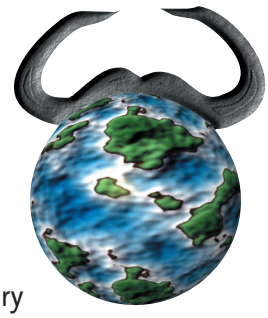
The monthly GNU column

Brave GNU World



Welcome to this issue of the Brave GNU World. Brave GNU World celebrates its four year anniversary with this issue and there would be no better way to help us with out celebrations than by taking a look at, even helping with, some of these projects.

BY GEORG C. F. GREVE

Even though some projects are probably of primary interest to developers, we still hope that less technical readers will also be able to draw new perspectives and be inspired by the projects shown here.

MagiConf

One of the points raised frequently by critics of GNU/Linux is that configuration of hardware is too complicated and not simple enough for beginners.

MagiConf [5] by Marc Boris Dürner, Ramesh Panuganty and Richard Ibbotson seeks to close this gap for the Debian GNU/Linux [6] distribution, which is the favorite of many people in the community, but not exactly known for its "end user suitability."

Thanks to a library written in C/C++, significant part of system components can be administrated easily by means of graphical user interfaces (GUI) based on GNOME/GTK or KDE/Qt.

Currently supported is configuration of the network card, sound card, mouse and keyboard for the XFree86, as well as connected PCMCIA and USB devices.

It should be said that in terms of sound cards, only "Open Sound System" (OSS) drivers are currently supported. The more modern and soon to be standard part of the Linux kernel "Advanced Linux Sound Architecture" (ALSA) [7] is not yet supported. Also configuring the graphical desktop (XFree86) is not yet possible. However, it is most likely it is just a matter of time until these will be supported.

If things are not developing fast enough for your liking, you can certainly speed things up by offering your support. Today, MagiConf is already supporting the hardware detection system Discover 2.0 [8] and this will certainly allow a high degree of automation and providing useful presets to the users.

The project began more than two years ago as a pure Qt project. After the essential functionality had been moved into a library, the authors were quick to provide a Gtk+ front-end, so both of the major desktops are equally supported. As often found, projects and distributions only support one desktop or another well, so the authors have some reason to be proud of this.

As a programming language, C++ was used for both the KDE and via GTKmm for the GTK clients. Licenses used in the project are the GNU Lesser General Public License (LGPL) for the library and the GNU General Public License (GPL) for the clients. So the project is entirely Free Software.

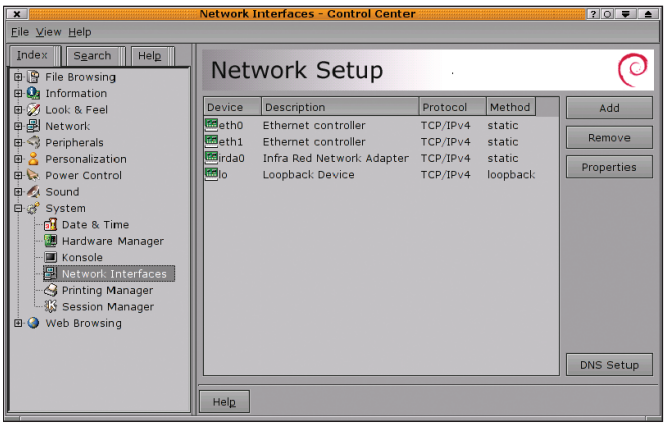


Figure 1: Magiconf showing GTKmm Frontendand the Network Controller Setup dialog

Moon-Buggy

Recently I introduced you to an ncurses based ASCII clone of the console classic "Space Invaders," here is some new food for friends of this genre.

Moon-Buggy [9] by Jochen Voss is an ASCII clone of the game which was very successful on the Commodore C64 and Amstrad CPC 464 [10].

The player is given the task to drive a Moon-Buggy across a crater-ridden moon surface without crashing it, so the concept is not very hard to grasp. Like so many simple games, this one is also highly addictive.

Proof for that is the growing Moon-Buggy fan community, which even went so far to write robots that could play Moon-Buggy for them. This came out when following up on a bug report [11] in which a player reported that the high-score list can only handle 5 digit scores.

That bug report did cause some surprise because no one expected that anyone might be able to get that far. Since it takes the robot two days of continuous play to reach it, that assumption is quite understandable.

The game was written around Christmas 1998, when Jochen Voss had some time to waste. Since then there have been several improvements and a Debian package by Christian T. Steigies; the project itself is now essentially stable.

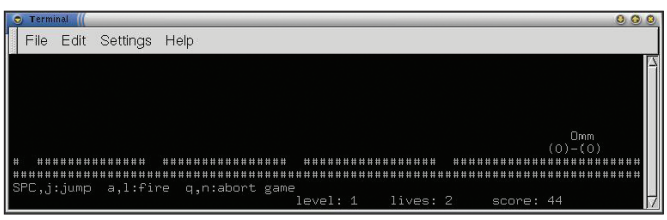


Figure 2: Moon-Buggy in action

It was written in C with ncurses and runs on pretty much all POSIX compliant Unix systems on the shell and console as well as over ssh connections.

Provided he finds time for it, Jochen plans to make some final improvements and maybe even add the UFO, which is present in the original, but missing in his program. Then it should finally be ready for the version 1.0 release.

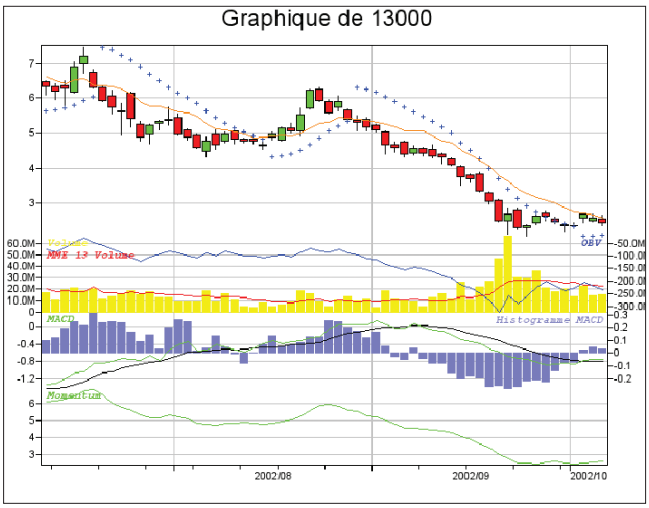


Figure 3: GeniusTrader graphing portfolios

cdcover

Nowadays, CD writers are spread widely and found in a large number of households, so many people will know the problems associated with trying to avoid losing track of all their CDs.

The program cdcover [12] by Roland Schäuble allows for the creation of covers and inserts for CD jewelcases in an interactive, comfortable and easy way. Provided fields are: title, subtitle, text on both front and back, footnote and date.

Based upon templates, the program generates postscript files from that data, which can either be printed directly or viewed in ghostview.

The project is written in Python; a logical consequence of Roland's motivation to learn Python by starting this project. He was especially careful to preserve the platform independence of Python and make cdcover usable on GNU/Linux (Unix) systems and Win32 alike.

His personal experience with this was that even when using Python it is not very easy to retain platform independence. As an example, he has so far failed to get the Python-CDDB module to run on Win32.

The binding to the CDDB database is currently only available on GNU/Linux and other Unices, allowing these platforms to import the artist, title and index of audio CDs from the internet.

This project, which is released as Free Software under the GNU General Public License (GPL), is a family project in some way, as the web page [12] was created by Roland's son, Michele Schäuble.

Roland sees possible room for improvement of the project especially in the configuration, which is still done by means of a ".cdcover" text file.

Working on this as well as adding support for saving data in a more portable format, such as XML, is part of Roland's plans for the future.

Any offers of help are quite welcome, especially in terms of more users and feedback, as well as more templates.

GeniusTrader

There are many niches that have not yet been freed by Free Software and sometimes projects set out to change that. GeniusTrader [13] by Fabien Fulhaber is such a project.

Supported by volunteers like Oliver Bossert, Fabien has begun creating a toolset for creating and testing automatic trade systems. His project will benefit "people who wish to get rich on the stock exchange by following signals generated by a computer."

Even if the stock fever has cooled down somewhat as a result of the current world economy, the project remains fascinating. Comparable software only exists as very expensive proprietary solutions, which are not really accessible to normal people.

GeniusTrader offers those who are interested the chance to simulate and evaluate trade strategies. So, even without the risk of real financial losses a lot about the stock exchange can be learned.

The project began in 1998, but has been declared dead twice in the course of its existence, every time because the approach was too broad, with no end in sight. Fabien made the third approach in 2002 and focused on the trade system, around which everything was assembled. He sees the largest weakness in its exclusive usage of the commandline.

The project is written in Perl, which requires users of the program to be proficient in the language to put it to any serious use, to modify the module for access to the database, for example.

Recently some structural changes have been made, so the current focus is on adapting the remaining modules. Fabien hopes to be able to implement a Web front-end.

For this he is still seeking help in form of user feedback and development contributions by others.

Enough

Enough Brave GNU World for this month, as usual I'd like to encourage comments, ideas, questions and suggestions by mail, [1]. ■

INFO

[1] Send ideas, comments and questions to Brave GNU World: *column@brave-gnu-world.org*

[2] Home page of the GNU Project: *http://www.gnu.org/*

[3] Home page of Georg's Brave GNU World: *http://brave-gnu-world.org*

[4] "We run GNU" initiative: *http://www.gnu.org/brave-gnu-world/rungnu/rungnu.en.html*

[5] MagiConf home page: *http://magiconf.sheflug.co.uk*

[6] Debian GNU/Linux: *http://www.debian.org*

[7] ALSA Project home page: *http://www.alsa-project.org/*

[8] Discover 2.0: *http://archive.progeny.com/progeny/discover/*

[9] Moon-Buggy home page: *http://www.mathematik.uni-kl.de/~wwwstoch/voss/comp/moon-buggy.html*

[10] Amstrad CPC 464: *http://www.obsoletecomputermuseum.org/amstrad/*

[11] The Moon-Buggy *Highscore-Bug" http://bugs.debian.org/75275*

[12] cdcover home page: *http://cdcover.sf.net/*

[13] GeniusTrader home page: *http://www.geniustrader.org*