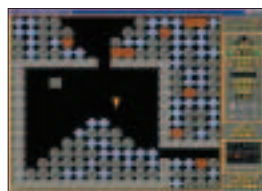


The monthly GNU column

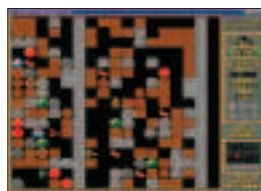
BRAVE GNU WORLD



Welcome to another issue of Georg CF Greve's **Brave GNU World**. This month we'll look at some more ways to creatively free yourself of free time



Rocks'n'Diamonds in Boulder Dash mode



Rocks'n'Diamonds in Emerald Mine mode



Rocks'n'Diamonds in Supaplex mode



Editor mode for Rocks'n'Diamonds

Rocks'n'Diamonds

Rocks'n'Diamonds, by Holger Schemel, is a game which bears a striking resemblance to classics like Boulder Dash (Commodore 64), Emerald Mine (Amiga) and Supaplex (PC). This shouldn't be overly surprising since it was written by a great fan of all these games.

For the unenlightened: the point of this classic 2D arcade game is to collect diamonds without suffering a premature end. To achieve this, you can (among other things) move rocks, drop bombs and fool monsters.

The game was written in C with an eye on portability. It runs on pretty much any flavour of Unix – given that X11 is supported – and also under MacOS X, DOS and Windows. With smooth scrolling of levels, joystick support and a freely customisable keyboard binding it can also be tailored to the preferences and circumstances of its users.

On top of the quite original-looking graphics, the overall feel is increased by support for sound effects and music on all operating systems that support it.

Another point in the game's favour is its networking support under Unix, which allows up to four players to take on levels together. It also has a local multi-player mode that lets players solve levels as a team on a single machine.

To make sure there will be no boredom the game has literally thousands of levels that need solving and once you've finished those, there's still the level editor.

Holger Schemel publishes it as Free Software under the GNU General Public License.

Although the game is already very mature,

development is still active. According to Holger, emulating the different game engines of Boulder Dash, Emerald Mine, Supaplex and Sokoban is still posing some problems since it is not yet good enough to play all their original levels. If you would like to help, you're welcome to do so. Also contributions in terms of graphics, help on porting it to another platform or new levels are all very welcome.

Holger himself experienced some of the interesting aspects of such a co-operation in the mid 90s when the German nuclear research centre Julich told him his game was crashing one of their AIX servers. Since he wasn't able to reproduce that bug, he was provided with Telnet access to the affected system and tracked the problem down to a faulty X11-system call. He was thus able to fix the bug while hoping he did not have too much of a negative impact on some nuclear gear.

Fortunately that is not very likely, but this little story nicely shows how the connecting and co-operative spirit of Free Software can sometimes bring you to interesting places.

Mirror Magic

Like the last game, Mirror Magic was also written by Holger Schemel. By way of context, the game itself was written in 1988 and distributed commercially and as proprietary software under the name Mindbender for the Amiga. Holger then ported it to Unix around 1994 and published it under the GNU General Public License as Free Software.

The goal of the game is to guide a laser beam out of its emitter into a detector. Given a couple of player-adjustable mirrors this could be easy but is

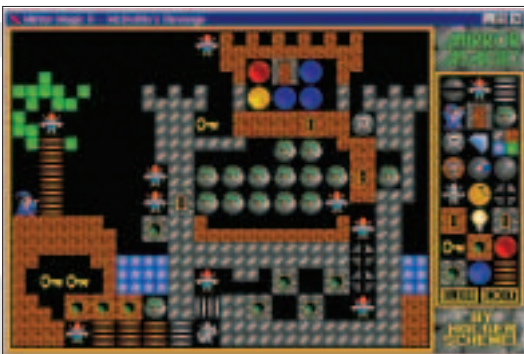




Mirror Magic

made increasingly difficult by all kinds of obstacles that can either be circumvented or destroyed by laser power. In some positions, the mirrors will cause the beam to feedback into the emitter, causing the laser to overheat and eventually explode; which is a rather undesirable outcome.

Simple principles can often lead to a lot of fun and Mirror Magic is no exception to this rule. Just like Rocks'n'Diamonds, Mirror Magic also provides nice graphics, sound effects and music. In fact both games have a suspiciously similar look, which is no coincidence since Rocks'n'Diamonds is based on the engine of Mirror Magic. Their relationship goes as far as the version number, which – at the time this column was written – is 2.0.1, released on March 19th 2002 for both games.



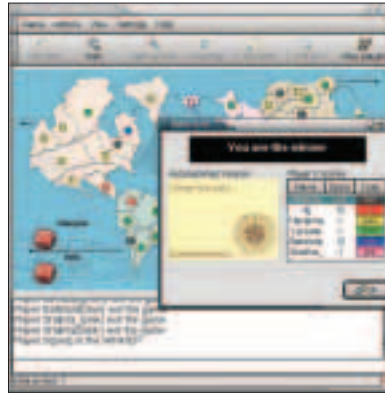
Mirror Magic's level editor

T.E.G.

Many things that would raise complaints in real life can be enjoyed freely in virtual space. Achieving world domination is one of them. T.E.G. ("Tenes Empanadas Graciela") is a clone of the well-known game Risk and was started by Ricardo Quesada in 1996.

The game concept of Risk shouldn't need much explanation, but in case some readers are not yet familiar with it: Risk is a board game in which the players compete with their armies in taking over control of certain regions or the whole world. Winning depends on tactical ability and some luck rolling the dice.

Although the project was declared dead many



Conquer the world the T.E.G. way



Different map with playing cards for T.E.G.

times during its development, it has been under continuous development since early 2000. It currently contains three maps, is capable of network play and has translations into Spanish, French, German and Polish.

Further plans focus on the generation of different rule systems, better maps, increased intelligence of robots and a meta-server. Help with these tasks by interested graphics designers or developers is certainly welcome.

T.E.G. was written in C using the GTK+/GNOME libraries and is published under the GNU General Public License.

J-TEG

Should the choice of C and GTK+ not suit your individual taste, you could try J-TEG by Jef De Geeter and Yves Vandewoude. It is a Java implementation of TEG although the codebase and development are entirely independent.

This project is also published under the GNU General Public License and since it uses the same networking protocol as TEG, both games can communicate with each other. In terms of translations, J-TEG currently offers Dutch, French, German and Italian.

Thanks to using Java, J-TEG should be able to run on almost all platforms supporting Java 1.3 or higher. But of course this also means it will have the Java-related problems. It would be good if Sun showed more interest in making Java a fully Open language and support Free reference implementations.

GNU Chess

GNU Chess is among the oldest projects of the GNU system, as its development began way back in 1984. It is under maintenance and development until today and should also find its place in the Brave GNU World.

The game Chess itself shouldn't need any explanation. Even non-players often know the rules and many people have had contact in school when taking numbers to the nth power was explained with



Connecting for network play with JTEG



Language choice in JTEG

a chessboard and grains of rice.

Given the origin and age of the project the choice of GNU General Public License and C as the programming language can hardly surprise anyone. The most active current developers of GNU Chess are Simon Waters, and Lukas Geyer. Stuart Cracraft, who maintained the project for many years, still helps them with advice and occasional replies to bug-reports, though he is slowly pulling out of GNU Chess. Kong-Sian should still be mentioned since he contributed the major part of the GNU Chess version 5 codebase.

Simon sees the focus of current development on maintaining and further expanding the high portability and on implementing an end-game database and an analysis mode. The analysis mode in particular is something he considers important since in his experience, complex programs can profit enormously from such a mode. Along with the already finished code clean up, the analysis mode should also help further increase the playing strength of GNU Chess.

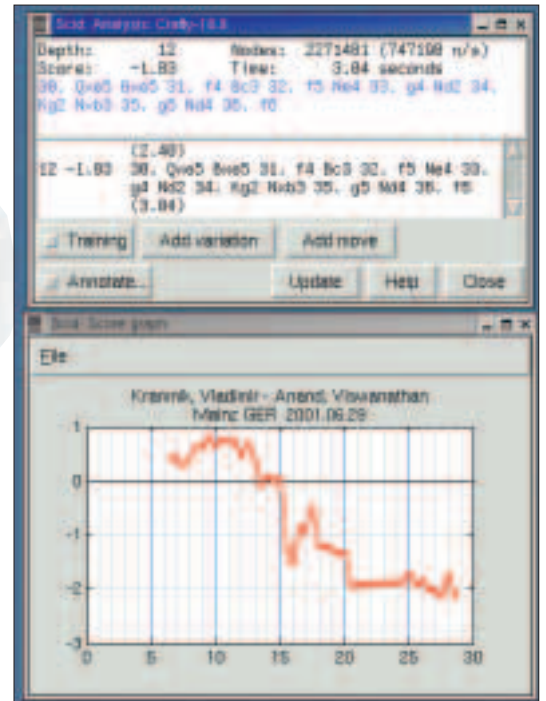
Even if there are many gratis Chess programs, some of which you can even get the source code for, GNU Chess seems to be very popular with maintainers of Web sites and authors of graphical chess programs who need a Chess engine that will give them the necessary freedom to port and integrate it easily and efficiently.

According to Simon's experience the freedom offered by GNU Chess is a major advantage that is very much valued, as he was able to discover by the large amount of patches sent in.

If you just want to play GNU Chess, you can of course do so, but a graphical front-end would probably be useful. The best known front-end is probably XBoard by Tim Mann, which is published under the GNU GPL.

Scid

Those who know Chess from playing will certainly know the value of a good Chess database – at least once you reach a



Scid analysis

certain level of skill. Scid (Shane's chess information database") is such a database, developed by Shane Hudson under the GNU General Public License.

Games can be easily and quickly entered in Scid to check in the database for certain search parameters. Since the use of any database is largely determined by the capabilities of maintaining it, this part of the functionality has been given a lot of the attention.

The possibility to train your own playing strength was also important to Shane and with the help of a WinBoard-compatible Chess engine you can even use it for analysing games. In an area dominated by proprietary and predominately expensive programs, Scid not only supports Windows but also runs under Unix and has – according to its author – a much easier, cleaner interface.

Of course the size of the accompanying game database is also an important factor for the usefulness of such a project. Some proprietary programs have more than 1,000,000 games in their database. From the Scid Web site, you can download a high-quality database with over 500,000 games of master level.

The development of Scid began in 1999 and today it is clearly a stable project with translations into 12 languages. As a program language, Shane used C++



XBoard front-end



Scid database

with Tcl/Tk for the graphical user interface. Therefore the most important tasks are maintenance of the help pages and the creation of a tutorial for new users. So if you like spending time pondering over Chess, you should risk a look at Scid.

GNU oSIP Library

RFC2543 describes the Session Initiation Protocol (SIP), a protocol to initiate, modify and terminate multimedia sessions. SIP was invented as a lightweight replacement of H323 in order to allow, in particular, hardware and software Internet telephones. Among other things it allows for proxies as gateways between networks and registrars to locate dynamic users.

The protocol bears a – desired – resemblance to the MAIL and HTTP protocols and just as it is possible today to mail me with the “mailto:greve@gnu.org”, SIP will make it possible to call me over the “sip:greve@gnu.org” URL one day.

Given that more and more companies are shifting from H232 to SIP and given that release five of the UMTS protocol is based on SIP, it is becoming increasingly important to implement this protocol freely.

The GNU oSIP (“Omnibus SIP”) library by Aymeric Moizard is one such Free implementation under the GNU Lesser General Public License and has recently become part of the GNU-Project. oSIP was written in C, deliberately limiting dependencies to the libc6 so that it may be used on as many systems as possible. This allows the use of oSIP in embedded devices and creates the foundation of mobile Internet phones based on Free Software.

The major advantages of oSIP compared to proprietary projects are that it is very small, flexible and Free. To the knowledge of Aymeric, there is also no other Free Software SIP C stack that is comparable to oSIP.

It is quite possible that SIP-based internet-telephony will completely replace the current telephone with its well-known players. If one combines the social and economic importance of communication by phone with the tendency of proprietary software to create monopolies, it immediately becomes apparent that communication needs to be possible with Free Software in order to help preventing a global monopoly on telecommunication. As such it is easy to understand why oSIP is a seminal contribution to the GNUMCOMM project by Aymeric.

libferris

Like the previous project, the Ferris library is also something that most “normal users” will never come in direct contact with. But even if its features are only immediately useful to developers, I believe it’s useful to have a certain understanding for what’s happening “behind the curtains” – even for pure users.

The libferris is a Virtual File System (VFS) running in user-space. Its function is to provide transparent, easy and consistent access to many different sources of data for programs – and therefore the users.

The libferris allows users to make databases, relational databases, XML, mailboxes, FTP-accounts, sockets, compressed and rpm archives, and SSH2 servers available as transparent directory structures. It also allows the direct extraction of certain information/data out of different file formats like ID3, MPEG2 and all image formats supported by the Imlib2 or ImageMagick libraries.

Users gain the advantage that it becomes irrelevant where or in which format data is stored, as all such details are handled by the libferris. Developers don’t have to worry about supporting dozens of file formats and transport layers anymore – all they need to write is a binding to libferris. Ben Martin began working on the libferris in April/May 2001 and had already released the first version in June 2001.

As the programming language for the project Ben chose C++, since he wanted to make extensive use of the C++ “Standard Template Library” (STL), which is also the reason why he decided against expanding the gnome-vfs for his purposes.

The object and steam orientation of libferris makes its expansion pretty easy and enables users to write their own modules to make other sources or formats accessible through the libferris.

Ben is seeking help in form of modules that allow the inclusion of different protocols or extracting information out of previously unsupported file formats. But he also wouldn’t mind being provided with nice, fast hardware as he pointed out.

Until the next time

That’s enough Brave GNU World for this month. As in every issue, I’d like to encourage you to send mail with ideas, feedback, comments and interesting projects to the usual address.

Info

Send ideas, comments, questions to Brave GNU World	column@brave-gnu-world.org
Homepage of the GNU Project	http://www.gnu.org/
Homepage of Georg’s Brave GNU World	http://brave-gnu-world.org
Rocks’n’Diamonds homepage	http://www.artsoft.org/rocksndiamonds/
Mirror Magic homepage	http://www.artsoft.org/mirrormagic/
T.E.G. homepage	http://teg.sourceforge.net
J-TEG homepage	http://ljteg.sourceforge.net
GNU Chess homepage	http://www.gnu.org/software/chess
XBoard and WinBoard homepage	http://www.tim-mann.org/xboard.html
SCID homepage	http://scid.sourceforge.net
RFC2543	http://www.ietf.org/rfc/rfc2543.txt
GNU oSIP library homepage	http://www.gnu.org/software/osip/
GNUMCOMM homepage	http://www.gnu.org/software/gnumcomm/
libferris homepage	http://witme.sourceforge.net/libferris.web