FOSSpicks

Sparkling gems and new releases from the world of Free and Open Source Software

Mike Saunders has spent a decade mining the internet for free software treasures. Here's the result of his latest haul…

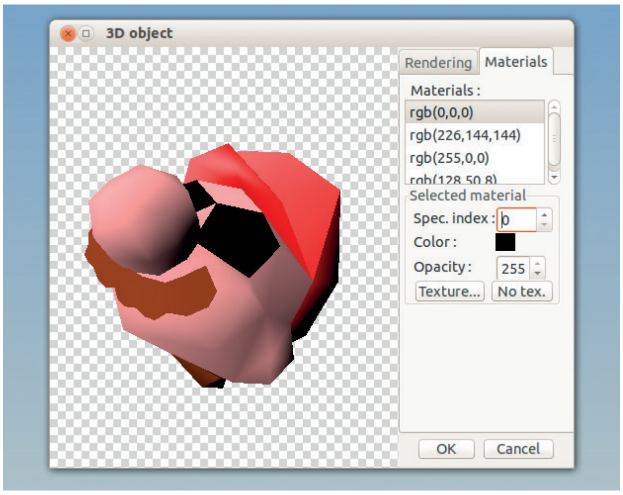
Image editor

LazPaint 6.2

LazPaint is an image editor somewhat akin to *Paint.NET* on Windows. Its name stems from the IDE used to build the program – *Lazarus* – which lets developers write applications in Object Pascal. Unusually, *Lazarus* can build the same program using different graphical toolkits, so in the downloads section for *LazPaint* (hosted on SourceForge) you can find pre-compiled binaries using *Gtk* and *Qt*. This helps the app to fit in better with your desktop, so it won't look ugly in Gnome, Xfce or KDE.

We used the *Gtk* version, and our main gripe with the interface is the tiny toolbar buttons and other controls. On a high-resolution display, they're almost comically small and hard to make out. Still, they all have decent tooltips, so after a few minutes of hovering the mouse and making mental notes, the interface becomes usable.

Like *Gimp*, *LazPaint* is a bitmap image editor with various tools for creating shapes, transformation effects (such as rotation and resampling) and multiple-level undo/redo. Extra dialogs for managing layers, colours and additional tools are available via the View menu – and from here, you can also enable a grid that's displayed over the image when zoomed in. This is particularly useful if you're making per-pixel edits to an image, as is common in video game sprite work.



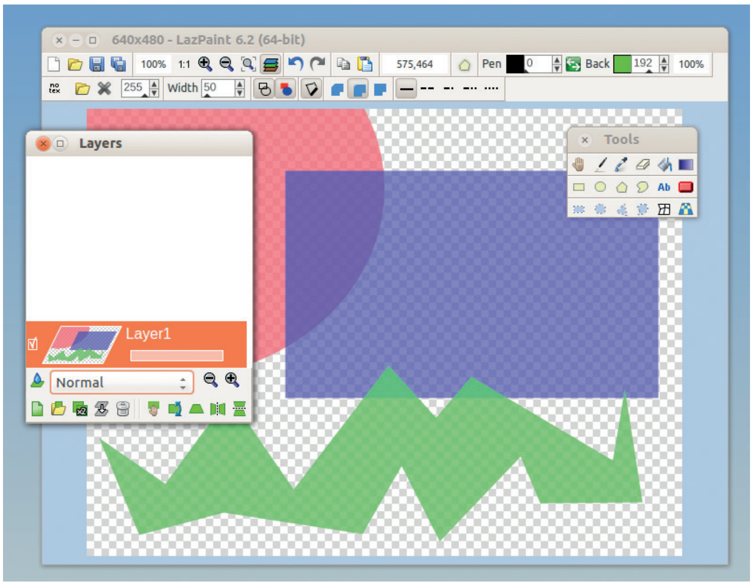
Materials:
rgb(0,0,0)
rgb(226,144,144)
rgb(255,0,0)
rgb(128,50,8)
Selected material
Spec. index: 0
Color:
Opacity: 255

LazPaint can import 3D models, such as this sinister Mario head.

Also like *Gimp*, *LazPaint* has a bunch of filters including blurs, sharpen, emboss and contour. The range isn't as extensive as *Gimp*'s, and some of them threw up bizarre error messages in our testing ("Access violation" anyone?) but on the whole they work well. It's also possible to render various textures from inside the program, and even import 3D models. Along with its native **.lzp** format, *LazPaint* can also save images to PNG, JPEG, TGA, TIFF, BMP and other formats.

> "Like Gimp, LazPaint is a bitmap image editor with various tools for creating shapes and effects."

In all, *LazPaint* isn't as feature-packed as *Gimp*, which might lead many to ask: what's the point of it? Well, if you're happy with *Gimp*'s interface, fair enough. But we know that a lot of people don't like how *Gimp* looks and works, and *LazPaint* provides a great deal of the same functionality with a simpler (and arguably more approachable) design. So we're glad there's some more variety in our choice of Linux image editings.



The teensy-tiny toolbar is really fiddly to work with at first, but at least the buttons have tooltips.

PROJECT WEBSITE
www.facebook.com/LazPaint

Operating system

ReactOS 0.3.16

For all its successes, Free Software is especially good at cloning existing projects. This doesn't mean there's a lack of imagination – just that it's often better to re-use an existing design rather than re-invent the wheel every time. Richard Stallman took this exact approach when creating GNU, which is famously "not Unix" but based very closely on the design and structure of that OS.

Now, ReactOS is a Free Software clone of Windows, aiming to be compatible with that OS's applications and hardware drivers. It's debatable whether Windows is a sensible design to copy, but in any case, the goal is that we'll all have a free (as in freedom and beer) OS on which to run legacy Windows applications. ReactOS has been in development since the late 90s, and we take a peek at it every few years to see how it's shaping up.

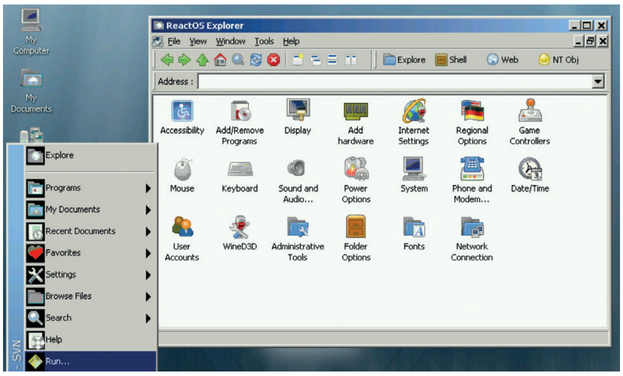
Helpfully, the development team makes it available in various formats: a live CD image, an installation CD image, and virtual hard drive images for *VMware* and *VirtualBox*. We took the latter for a spin, which worked fairly well, although it booted into a terrible 16-colour mode that looked completely pants. A quick tweak in the Control Panel soon sorted that out, however. By and large, ReactOS looks and feels like Windows 2000, albeit with plenty of rough edges and various missing pieces.

Most of the standard Windows tools are available though: a command line prompt, text editor, *Paint*-like program, Control Panel and so forth. And while many of these have been created from scratch for ReactOS, a great deal of code has been taking from the *Wine* project, which helps to cut down on duplication of effort.

How's your luck?

So, the big question is: how good is the compatibility? It's a mixed bag, and programs that don't run well on *Wine* (see **https://appdb.winehq.org**) tend to break on ReactOS too, in our experience. You'll have more luck with early-2000s applications, as support for the Windows API of that time is more complete – many recent programs simply won't start at all. Still, ReactOS isn't short of software, thanks to an Application Manager that downloads (mostly open source) Windows programs from the web and installs them. So you can get *Firefox*, *AbiWord*, chat clients, various games and other system tools with just a few clicks.

> "ReactOS isn't short of software, thanks to Application Manager that downloads programs from the web."



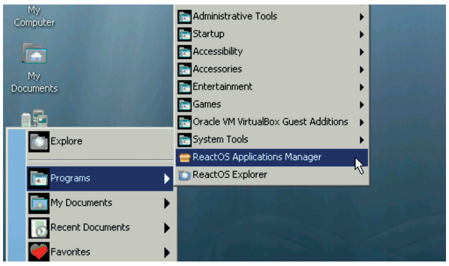
In many parts, ReactOS's interface is a pixel-perfect clone of Windows 9x/2000, which could get Microsoft grumbling one day.

In the last couple of years, the ReactOS team has tried to boost interest and contribution from the wider open source community with fundraising projects. Ultimately, we think it would progress more quickly with some solid commercial backing – but we can understand that many companies don't want to touch it with a 50ft barge pole, in case Microsoft tries to stomp it down with legal action.

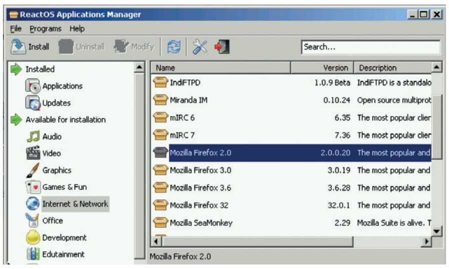
Nonetheless, ReactOS is an impressive effort and deserves more fame. Hopefully it will reach a level of sufficient stability and completeness to run most older Windows XP programs without problems, to help those who don't want to "upgrade" to Windows 7/8/10 and can't switch to Linux for whatever reasons.

PROJECT WEBSITE
www.reactos.com

How it works: Adding applications



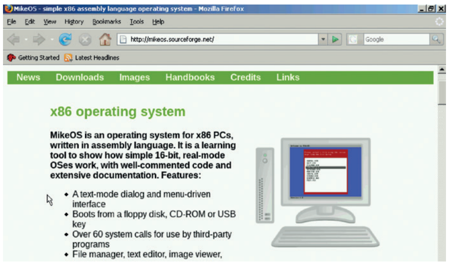
1 Menu
ReactOS isn't supplied with a web browser, so go to Start > Programs > ReactOS Applications Manager to bring up the software exploration tool.



Name	Version	Description
IndiFTPD	1.0.9 Beta	IndiFTPD is a standalo
Miranda IM	0.10.24	Open source multiprot
mIRC 6	6.35	The most popular clier
mIRC 7	7.36	The most popular clier
Mozilla Firefox 2.0	2.0.0.20	The most popular and
Mozilla Firefox 3.0	3.0.19	The most popular and
Mozilla Firefox 3.6	3.6.28	The most popular and
Mozilla Firefox 32	32.0.1	The most popular and
Mozilla SeaMonkey	2.29	Mozilla Suite is alive. T

Mozilla Firefox 2.0

2 Searching
Go into the Internet & Network category on the left. Various versions of *Firefox* are available – in our experience, the older ones tend to work more reliably.


MikeOS - simple x86 assembly language operating system - Mozilla Firefox
File Edit View History Bookmarks Tools Help
http://mikeos.sourceforge.net/
Google
Getting Started
Latest Headlines
News
Downloads
Images
Handbooks
Credits
Links

x86 operating system

MikeOS is an operating system for x86 PCs, written in assembly language. It is a learning tool to show how simple 16-bit, real-mode OSes work, with well-commented code and extensive documentation. Features:

- A text-mode dialog and menu-driven interface
- Boots from a floppy disk, CD-ROM or USB key
- Over 60 system calls for use by third-party programs
- File manager, text editor, image viewer,

3 Install
After you've selected a version, click Install in the top-left and the program will be retrieved from the web. You'll then find it in the Start menu.

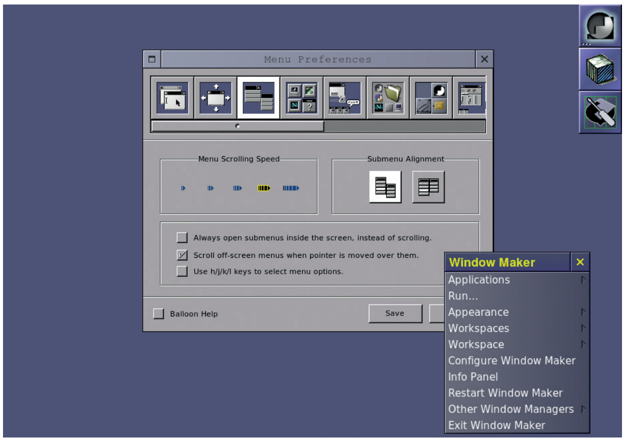
Window manager

Window Maker 0.95.6

Steve Jobs will be remembered for many things: shiny iGadgets by most people, and a purveyor of walled garden "digital prisons" by some in the Free Software camp. But one of Jobs's lesser known creations is Next, a computer company he built after being booted out of Apple. Next sold tremendously powerful (and equally expensive at $9,999) workstation machines that never achieved widespread popularity, but left their mark on the industry. The first ever web browser was written on a Next box, for instance.

Anyway, the Next operating system had an attractive and novel interface that's significantly different to the usual taskbar-plus-program menu approach we're all familiar with. *Window Maker* apes this very closely, while providing extra graphical fluff including themes and gradients. This window manager hasn't seen a great deal of activity in recent years, but given that it was one of our favourites in the early 2000s, we're glad to see it's still receiving minor updates.

Window Maker is all about the dock in the top-right corner. Try right-clicking on the desktop to bring up a program menu, launch something, and then drag its icon (usually found in the bottom-left of the screen) onto the dock. This will save it as a launcher for later. You can slide the dock around by clicking and dragging on the top icon, and change launch settings for the icons via right-click context menus. A configuration tool is provided for setting up the window manager – so you can customise it without having to manually edit config files.



Menu Preferences

Menu Scrolling Speed

Submenu Alignment

☐ Always open submenus inside the screen, instead of scrolling.

☑ Scroll off-screen menus when pointer is moved over them.

☐ Use h/j/k/l keys to select menu options.

☐ Balloon Help

***Window Maker* is highly customisable, and easy to tweak thanks to the supplied config tool.**

Window Maker is fast, fluid and attractive, providing a genuinely fresh approach to the desktop. It's way less demanding on the RAM banks than the big desktops such as KDE and Gnome, but is still more approachable and easy to customise than the ultra-minimal window managers. We hope it stays around for years to come.

PROJECT WEBSITE
www.windowmaker.org

Trimmed-down systemd alternative

uselessd 2

Despite the fact that *Systemd* has been adopted by almost every major distro, internet debates are still raging about it. Proponents say it simplifies and streamlines the Linux boot process, providing extra features for process isolation and logging. Conversely, critics say it violates long-time Unix principles and it's swallowing up too much functionality that should be left in other components.

Uselessd, which could be described as "useless" or the daemon that "uses less" according to the developer, is a fork of *Systemd* that aims to bring it back to basics. The idea is that *Systemd* is actually good for some things – namely booting the system, starting services, managing dependencies between them and making sure they don't exceed their resource limits – but that's it. Unlike *Systemd*, *Uselessd* won't keep growing and taking over other parts of the core system, such as logins and network management.

There are some other changes too. *Uselessd* doesn't use the controversial *Journald* system by default, instead logging to plain text files in the usual Unix way. Yes, some *Journald* supporters say that the binary format is actually better, as much more metadata is stored and you can perform complex searches without a zillion grep commands and regular expressions. But many still prefer the plain text syslog. In addition, *Uselessd* aims to be more portable, working with other C libraries than just *Glibc*. The goal is to purge *Systemd* of "GNUisms" – in other words, things that lock it deeply into the GNU/Linux ecosystem.

"Uselessd is a fork of Systemd that aims to bring it back to basics."

Like much of the project, *Uselessd*'s logo mocks the original systemd, here implying that the latter includes the kitchen sink.

PROJECT WEBSITE
http://uselessd.darknedgy.net

Nintendo Entertainment System emulator

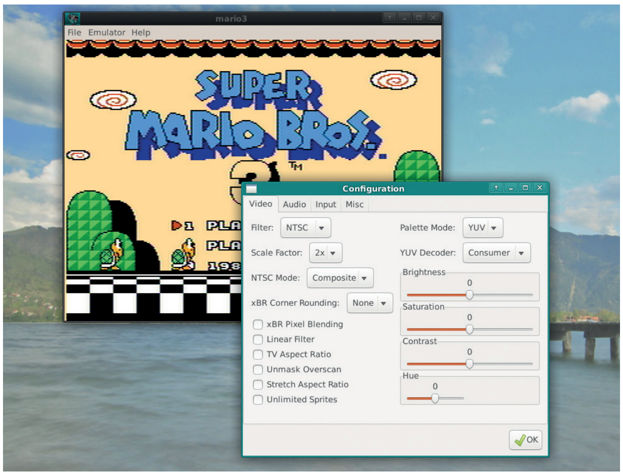
Nestopia 1.46.1

There are two types of console emulator user: those who just like to indulge in nostalgia for a moment, and hardcore players who demand that everything is exactly like the original – no exceptions. *Nestopia* is geared towards the latter group of users, with an intense focus on emulating the NES hardware as accurately as possible. You may have played other NES emulators before, and not noticed anything odd, but some emulators take barely-noticeable shortcuts to improve performance.

Nestopia, however, strives to be "cycle accurate"; in other words, with every emulated 6502 CPU cycle, it emulates the activity of all the other hardware components, to keep the virtual NES running in perfect sync. Most of us wouldn't notice this, but for some NES fanatics, it's vital for delivering the most realistic emulation. But this comes at a cost – *Nestopia* consistently used 30% of our CPU (emulating *Super Mario Bros 3*), which is a fair chunk on a 2.5GHz Core i5 machine.

Outside of this focus on accuracy, *Nestopia* is a fine all-round emulator too. It's easy to configure input devices and video options, while save states are also supported. You can even record gameplay in NSV format, and convert it to a more mainstream format using external tools. This is great if you're rather hot at a certain game, and want to upload your skills to YouTube.

"Nestopia aims to provide 'cycle accurate' NES emulation."



Filter: NTSC
Scale Factor: 2x
NTSC Mode: Composite
xBR Corner Rounding: None
□ xBR Pixel Blending
□ Linear Filter
□ TV Aspect Ratio
□ Unmask Overscan
□ Stretch Aspect Ratio
□ Unlimited Sprites
Palette Mode: YUV
YUV Decoder: Consumer
Brightness 0
Saturation 0
Contrast 0
Hue 0

With some tweaks to the video output settings, you can create the lovely old fuzzy CRT TV effect.

Cheats are supported, using Game Genie and Pro Action Rocky codes, and there's an impressive range of options for configuring the video output. The defaults are fine for regular NES emulation, but you can also smooth out jaggedness between pixels using filters – try 2xSal or HqX to see how it looks.

PROJECT WEBSITE
https://github.com/rdanbrook/nestopia

Vim status bar upgrade

Vim-airline

Some people are put off from learning *Vim* because of its horrendously steep learning curve, which is a shame, because it's a fantastic editor when you've mastered it. Others have tried battling through *Vimtutor* but still found the editor horrendously bare and terse – like it doesn't even try to make you feel welcome. Yes, *Vim* out-of-the-box isn't very pleasant, but once you've added the ruler, status line, syntax highlighting, line numbers, highlighted searches and other features, it quickly becomes more livable.

Vim-airline is a plugin that goes a step further, replacing the stock status line with one that's much prettier and more informative. You can install it using a variety of *Vim* plugin managers – such as *Pathogen*, *Vundle* and *VAM* – or install it manually by copying its files into **.vim** in your home directory. Because we were already using *Pathogen*, we just needed to enter:

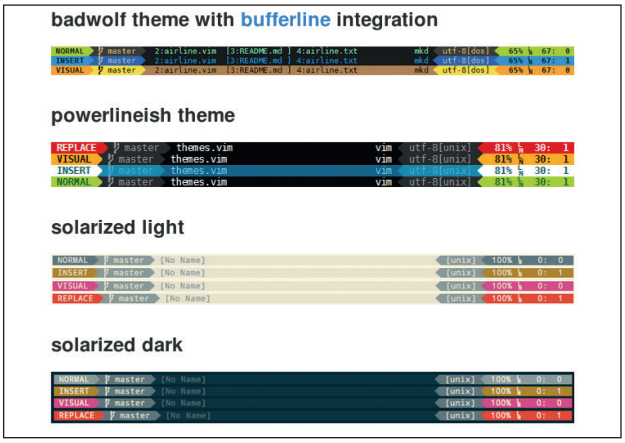
```
git clone https://github.com/bling/vim-airline ~/.vim/bundle/vim-airline
```

After starting *Vim*, however, we saw the new status line but it didn't have any colour (in *Xfce-Terminal*). If you have a similar problem, add this to the start of your .vimrc:

```
set t_Co=256
```

Now you should see *Airline* in all its glory. The default theme is rather garish and might not work well with your colour scheme, so to fix this, go into **~/.vim/bundle/vim-airline/autoload/airline/themes/** and take a look around. To activate the theme 'powerlineish', for instance, add this to your **.vimrc**:

```
let g:airline_theme = 'powerlineish'
```



badwolf theme with bufferline integration

```
NORMAL  master  2:airline.vim [3:README.md ] 4:airline.txt   mkd  utf-8[dos]  65%  67: 0
INSERT  master  2:airline.vim [3:README.md ] 4:airline.txt   mkd  utf-8[dos]  65%  67: 1
VISUAL  master  2:airline.vim [3:README.md ] 4:airline.txt   mkd  utf-8[dos]  65%  67: 0
```

powerlineish theme

```
REPLACE  master  themes.vim            vim  utf-8[unix]  81%  30: 1
VISUAL   master  themes.vim            vim  utf-8[unix]  81%  30: 1
INSERT   master  themes.vim            vim  utf-8[unix]  81%  30: 1
NORMAL   master  themes.vim            vim  utf-8[unix]  81%  30: 1
```

solarized light

```
NORMAL   master  [No Name]             [unix]  100%  0: 0
INSERT   master  [No Name]             [unix]  100%  0: 1
VISUAL   master  [No Name]             [unix]  100%  0: 0
REPLACE  master  [No Name]             [unix]  100%  0: 1
```

solarized dark

```
NORMAL   master  [No Name]             [unix]  100%  0: 0
INSERT   master  [No Name]             [unix]  100%  0: 1
VISUAL   master  [No Name]             [unix]  100%  0: 0
REPLACE  master  [No Name]             [unix]  100%  0: 1
```

Various *Airline* themes in action. Solarized works well with the *Vim* colour scheme of the same name.

Airline shows your current mode (and changes colour depending on the mode), the file you're editing, its encoding, along with your position in the file (line, column and percentage). If you're using tabs, you can also spruce up your tab bar by adding this to your **.vimrc**:

```
let g:airline#extensions#tabline#enabled = 1
```

PROJECT WEBSITE
https://github.com/bling/vim-airline

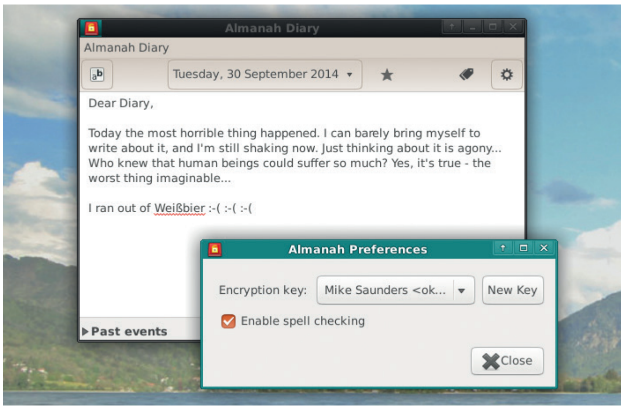
Diary editor

Almanah 0.11.0

Dear diary, today I spent several hours trying to compile programs from source code, frequently coming up against missing dependencies, compilation errors and segfaults. Just when I thought one program had built correctly, I started it and received around 3,000 lines of Python error messages before the whole thing crashed out spectacularly. Why, diary, does it have to be this way? Why can't programs be nice and easy to install, like *Almanah*?

Yes, this is a diary editing program – essentially a custom text editor with journal-keeping-related features. To build it you need **libsqlite3-dev**, **libcryptui-dev** and **libgpgme11-dev**. With those in place, installing it was a cinch with the **./configure**, **make** and **make install** (as root) procedure.

After starting the app, you're presented with an empty entry for today. The single button in the top-left provides some basic formatting facilities – bold, italic and underline – but there's no undo or redo for text editing operations, annoyingly. Next to the formatting button is a date; click that and you can jump to other diary entries (if you've written them). There's also a star button that you can use to mark certain diary entries as especially important.



Dear Diary,

Today the most horrible thing happened. I can barely bring myself to write about it, and I'm still shaking now. Just thinking about it is agony... Who knew that human beings could suffer so much? Yes, it's true - the worst thing imaginable...

I ran out of Weißbier :-(:-(:-(

Encryption key: Mike Saunders <ok...

☑ Enable spell checking

Don't want the world to see your heart-pourings? Encrypt your database using your PGP key.

To help organise diary entries, you can apply tags to them, and even attach links to files and web addresses via the cog menu. It's possible to export the diary in plain text format, with each day stored in a separate file (using YYYY-MM-DD filenames), or print it out.

Of course, if your diary contains plenty of juicy secrets you'll want to keep it safe: to help with this, *Alamanah* can encrypt it using your PGP key. Click the Almanah Diary menu and then Preferences to set it up.

> "Almanah can encrypt your diary entries using your PGP key."

PROJECT WEBSITE
https://wiki.gnome.org/Apps/Almanah_Diary

Web browser

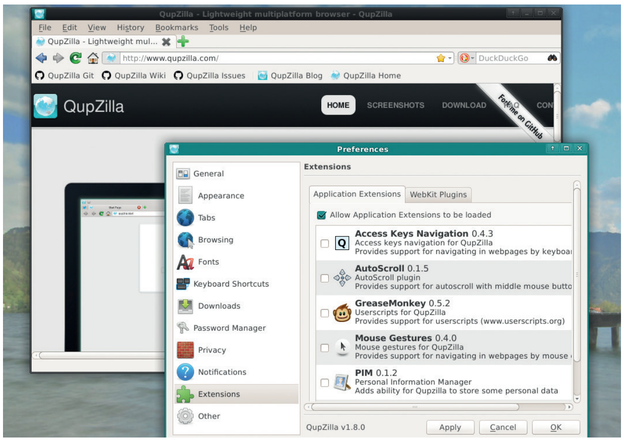
QupZilla 1.8.0

QupZilla is a *Qt*-based browser using *WebKit* as its rendering engine, designed to provide a good out-of-the-box experience without the need for extra plugins and extensions.

To build it from the source code, you'll need development libraries for *Qt* (at least version 4.7) and OpenSSL installed. Then run **qmake** followed by **make**, and if it compiles successfully, run **make install** as root to place it in your filesystem. Then you can start it by just entering **qupzilla**.

Interface-wise, it's pretty standard fare; there's nothing especially original that leaps out at you, and the speed dial-type interface on new tabs has been done before. Ex-*Firefox* users who were miffed by the Australis UI revamp will be happy with *QupZilla*'s layout – it doesn't try to chase *Chrome* in every direction. In terms of memory consumption, it was on a par with *Firefox* in our testing, and slightly lighter with many tabs open.

QupZilla is supplied with a gaggle of useful extras, such as an advert blocker. This uses the EasyList from Adblock Plus – so it's very effective. There's also a combined history, bookmarks and RSS view, which can be brought up with Ctrl+Shift+H, and having these things together in one place works surprisingly well. Another handy feature is the ability to take a screenshot of an entire web page, and save it in various formats – useful if you want to show a web developer where something is broken.



☑ Allow Application Extensions to be loaded

□ **Access Keys Navigation** 0.4.3
Access keys navigation for QupZilla
Provides support for navigating in webpages by keyboa

□ **AutoScroll** 0.1.5
AutoScroll plugin
Provides support for autoscroll with middle mouse butto

□ **GreaseMonkey** 0.5.2
Userscripts for QupZilla
Provides support for userscripts (www.userscripts.org)

□ **Mouse Gestures** 0.4.0
Mouse gestures for QupZilla
Provides support for navigating in webpages by mouse

□ **PIM** 0.1.2
Personal Information Manager
Adds ability for Qupzilla to store some personal data

QupZilla is bundled with a handful of extensions, including the ever-useful *GreaseMonkey*.

By default, *QupZilla* uses DuckDuckGo for web searches, and also enables the Do Not Track (DNT) header. The latter is a bit controversial though: if all browsers simply enable DNT by default, many users won't hear about it and all websites will just ignore it. But if users are told about the feature and turn it on themselves, they'll expect better conformity from sites. At least, that's the idea...

PROJECT WEBSITE
www.qupzilla.com

FOSSPICKS Brain Relaxers

Space shooting/racing larks

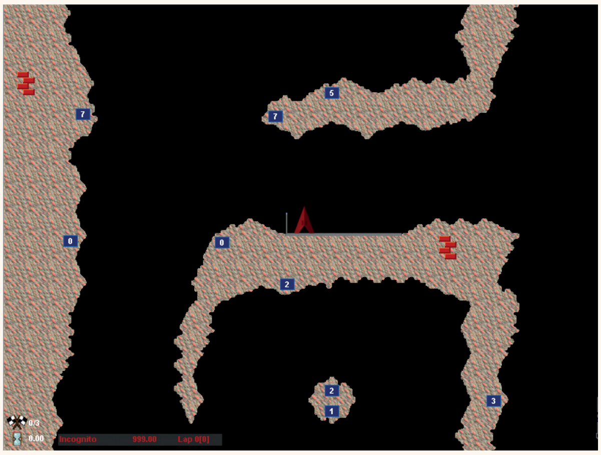
Galaxy Forces 1.82

Galaxy Forces is essentially a souped-up version of *Thrust*, a 1986 classic that appeared on pretty much every 8-bit home computer. And it's all about control here: you move a spaceship around by hitting the cursor keys to rotate it, and up to blast your rear thruster.

However, your ship doesn't simply fly in the direction it's pointing, due to gravity and momentum, so you have to plan your movements in advance (or perform some nifty rotate-and-thrust manoeuvres to slow yourself down). It takes a while to get the hang of, and you'll find yourself crashing into the walls at first. But don't give up – once you've mastered the controls, you'll be wallowing in self-satisfaction as you cruise gracefully around a level, avoiding obstacles with pixel-perfect precision.

Galaxy Forces has 50 levels; the first bunch are purely for network-based multiplayer combat games, while the others can also be played alone. These latter levels involve racing around a course (trying to beat the best times, as listed on the game's website) or carrying cargo in missions. So there's plenty to do, and while the game selection interface is incredibly clunky and the music is annoyingly repetitive, the gameplay itself has been well-crafted.

Note that you need to connect to IP address 127.0.0.1 (ie your local machine) to start the game, and then click the New button to choose a level. Helpfully, the developers have made available a pre-compiled binary package, with executables for both 32-bit and 64-bit machines.



This might look simple to navigate, but when your craft has so much momentum it's hellishly difficult.

PROJECT WEBSITE
www.galaxy-forces.com

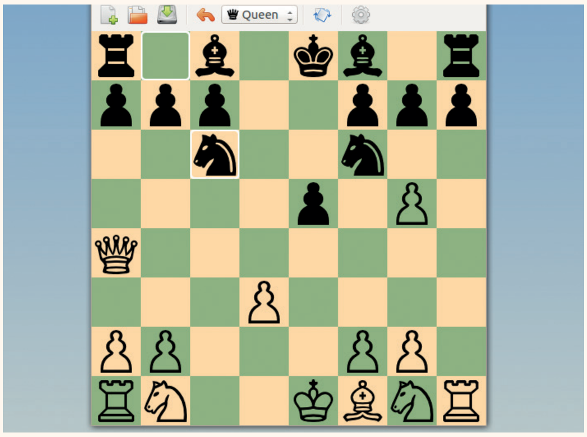
Chess game

Gambit 1.0

Running untrusted scripts from the internet, especially when they ask you to enter your user or root password, is a bad idea. Down this path lie Windows-esque levels of madness. However, it's the simplest way to get *Gambit* up and running: no distro-specific packages were available at the time of writing, so the developer recommends you grab **gambit-autobuild.sh** from the website, make it executable, and run it. And it works – the script grabbed dependencies, retrieved the source code, built it, and placed the resulting program in **Gambit** in the home directory. So to play, you just need to run **./gambitchess** inside it.

You're thrown straight into a game, controlling the white pieces, on a difficulty level of 5 (out of 5). We're not skilled enough in chess to judge exactly how challenging this level is, but we'd rather *Gambit* defaulted to something in the middle – or at least prompt the player before the game starts. From there, it's like most other chess games, where you click and drag pieces to make your move, and the computer thinks and responds.

Various colour schemes are available for the board, along with animations as the pieces move. Excellently, *Gambit* supports saving and loading games in the Portable Game Notation (PGN) format, which has been around for decades and is the *de facto* standard for storing chess matches. You can even find archives of PGN files on the net for classic battles between famous players.



Gambit's difficulty level 1 isn't hard to beat – even with our limited skills, we soon snaffled the CPU's queen.

PROJECT WEBSITE
http://gambitchess.sourceforge.net