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…

Create images from ASCII diagrams

Asciidia 0.3.2

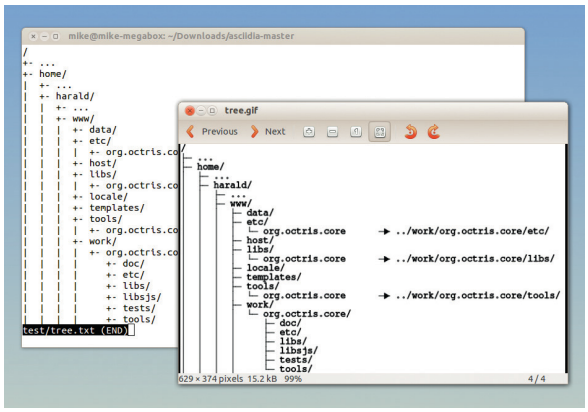
Plain text is excellent, and much more versatile than a lot of people give it credit for. Take Markdown syntax, for one simple example:

```
Heading
=======
Some **bold** text
* This
* is a
* list
```

Although this is good old plain text, the extra formatting here is easy to understand, and the Markdown tools convert this text into decent HTML with just one command.

Asciida does a similar job, but it focuses on creating diagrams from plain text files. In other words, you create images in a plain text editor using regular characters, run Asciida, and get a proper vector (or bitmap) version.

As Asciida is written in PHP, you'll need the **php5-cli** package installed to run it. Extract the tarball, **cd** into the resulting directory, and there you'll see the program: **asciida.php**. The best way to learn how Asciida works is with an example file, and fortunately a few are supplied in the test directory. Have a peek at **test/diagram.txt**, for instance – it looks like classic ASCII art (see the screenshot), but Asciida is clever enough to work out the shapes and signs contained therein.



Here's **diagram.txt** being viewed in a plain text terminal, and Asciida's SVG conversion shown in Firefox. Nicely done.

Words and pictures

To convert **test/diagram.txt** into a vector (SVG) image, enter this:

```
./asciidia.php -t diagram -i test/diagram.txt -o svg:diagram.svg
```

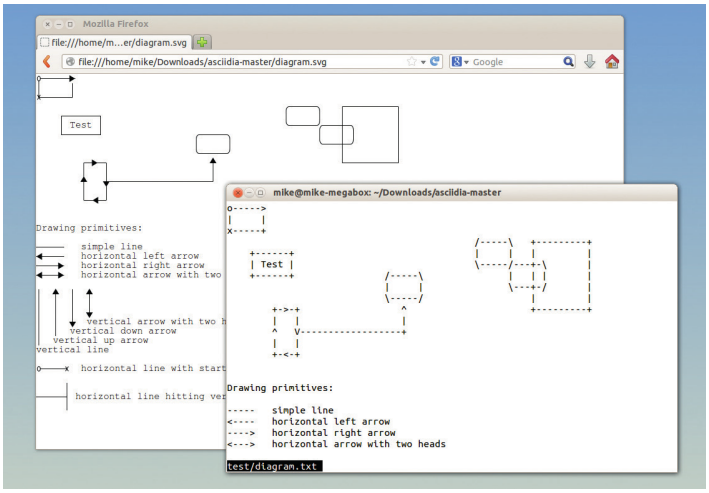
Asciida doesn't provide much in the way of feedback, but its silence at the command line shows that the conversion has worked. Now open **diagram.svg** in a vector graphics editor such as Inkscape – or you can even open it in Firefox if you don't have a vector editor to hand. And *voilà*: there's a fancier version of the ASCII art diagram, with everything in its right place.

> **"Asciida does a similar job to Markdown, but focuses on creating diagrams from plain text files."**

Asciida can also generate bitmap images providing that the **convert** tool from ImageMagick is installed. For some reason, on our Ubuntu 13.10 test box Asciida complained that the **MAGICK_HOME** environment variable wasn't set, so we had to run the program like this:

```
export MAGICK_HOME=`which convert`
./asciidia.php -t diagram -i test/diagram.txt -o png:diagram.png
```

Note the **-o png** option here to produce PNG output. Asciida can generate files in other formats too – see **README.md** for the details.



Asciida can convert other types of plain text diagram, such as directory trees.

PROJECT WEBSITE
https://github.com/aurora/asciidia

Image manager

Phototonic 0.93

Zawinski's Law of software development, from Mozilla and XEmacs hacker Jamie Zawinski, states: "Every program expands until it can read mail." While this law is used in jest to mock programs like Emacs that have stretched way beyond their original purposes, it makes a good point: bloat and feature creep are everywhere in free software.

In contrast, we love it when a program has a very clear goal, as in the case of Phototonic. It manages your images – nothing else. It doesn't try to link to social media websites, or provide advanced editing facilities, or wash your car. All it does is help to view and organise your pictures, and consequently it's fast and reliable.

Phototonic is written in C++ with Qt 4 as the front-end, so you'll need the **libqt4-dev** and **qt4-qmake** packages installed (that's in Debian and Ubuntu – in other distros they may have different names). After extracting the tarball and **cd**ing into the new directory, enter:

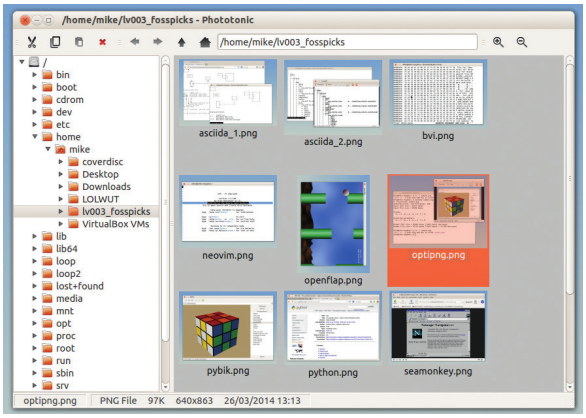
```
qmake-qt4
make
```

This builds the source code, and you can enter **sudo make install** to copy the binaries into your filesystem. Then just enter **phototonic** to start the program.

At first glance, Phototonic looks somewhat like a regular file manager, with a tree view of the filesystem down the left, and a pane containing thumbnails on the right. Navigate into a directory containing images, and you'll see them on the right-hand side. You can right-click on directories and images to rename them; this is how you're meant to organise your images in Phototonic. It doesn't try to do anything fancy with tags or metadata, but trusts that you can achieve what you want with a good old-fashioned directory tree.

Click on an image to highlight it (11 file formats are supported), and information about the image's file format, file size and dimensions is displayed in the bottom status bar. Double-click a picture to view it close-up, and press Escape to go back to the main screen.

Under the View menu you'll find options to change the sizes of the thumbnails along with the sort order, while the toolbar at the top of the window lets you navigate like in a web browser.



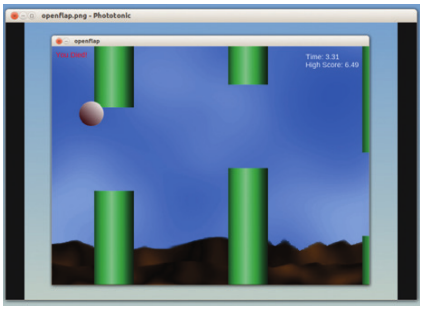
Use the plus and minus magnifying glass buttons in the top-right of the window to change the thumbnail sizes.

Phototonic includes basic image editing facilities (rotating, flipping and cropping) along with a slideshow view that shows a new image every five seconds. This default duration, along with other aspects of the program including default keybindings, is configurable in the Preferences panel. In all, it's exactly what a standalone image manager should be: simple, fast, stable and not overloaded with buggy features that should be implemented in separate apps.

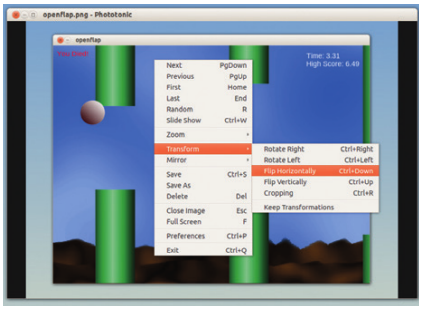
"Photonic is exactly what a standalone image manager should be: simple, fast and stable."

PROJECT WEBSITE
http://oferkv.github.io/phototonic

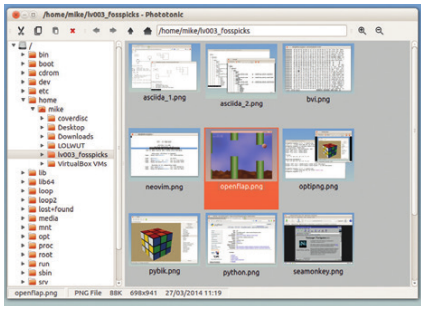
How it works: Editing an image



1 Select
Double-click on an image to display it in full. In this view, use the Page Up and Page Down keys to navigate through images.



2 Transform
Right-click on the image and go into the Transform menu. There you'll find options for rotating, flipping and cropping the image.



3 Save
Right-click again and choose Save, then press Esc to return to the thumbnail view. Hit F5 to refresh the thumbnails to reflect your changes.

Vi-like hex editor

Bvi 1.4.0rc

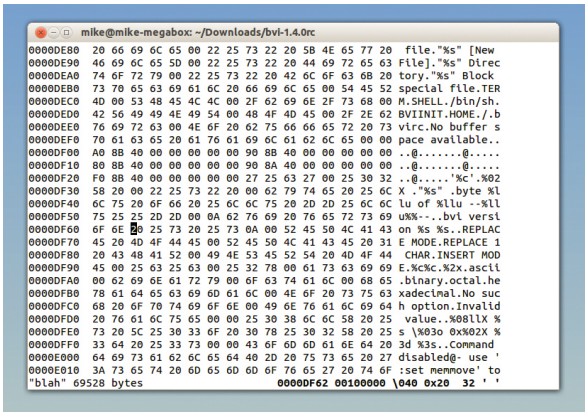
Nobody forgets their first encounter with the Vi(m) text editor. Compared with most "normal" editors, where you can simply type in text and press a key combination to save your work, Vi initially seems bizarre, with its arcane system of modes and commands like **:wq**. Yet after spending a while with Vi, many people find it to be incredibly efficient and powerful – hence its huge army of dedicated fans.

If you're one such Vi supporter, you might like Bvi: it's a similar editor geared towards working with binary files. To install it, you'll need GCC and the ncurses development files (eg the **libncurses5-dev** package in Debian/Ubuntu). After you've built the source code you can open a file with **bvi filename**.

Now, Bvi splits the screen into four sections: the left-hand panel shows the addresses (ie locations) of bytes in the file, in hexadecimal (base 16) format. The middle panel shows the contents of those bytes in hexadecimal format, while the right-hand panel shows the ASCII representations of those bytes. Finally, a status line at the bottom shows the filename along with the currently highlighted byte in binary, octal, hex and decimal formats.

Vi-like commands

Move around using the cursor keys (or in more traditional Vi style, H/J/K/L), and hit Tab to switch between the hex and ASCII panels. To replace a byte, hit R. By default you can't insert or delete bytes - use the **:set memmove** command to enable these operations, and then **i** and **d** to do them. Saving a file and quitting is just like in Vi as well, with **:w** and **:q** respectively.



It's not pretty, but it's extremely lightweight (68k for the executable) and very useful when you need to poke around inside binary files.

Bvi has many more commands taken from Vi, along with a few useful extras such as the ability to edit a specified range of bytes in a file, instead of the whole file. Its minimal requirements mean that, like Vi, it will run almost everywhere, so it's on our list of "things to install by default in a new distro" now.

PROJECT WEBSITE
http://bvi.sourceforge.net

PNG file compressor

OptiPNG 0.7.5

Many of the new Free Software programs that get released are easy to ignore, performing piffling little jobs that perhaps only the developer finds useful. At first glance, one such is OptiPNG. Wow, so it reduces PNG file sizes by 10–15%; who cares? In this day and age of terabyte hard drives and blazingly fast internet connections, do such trivial reductions really matter?

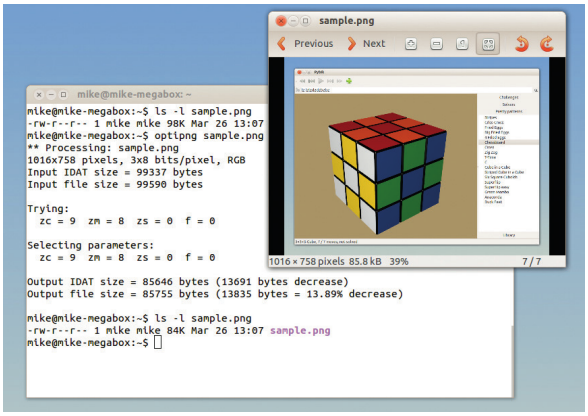
Well, they do in some circumstances – and they matter a lot. If you're hosting a very popular website serving up thousands of PNG images every minute, even the smallest reductions can add up over time, making your website faster and ultimately reducing your bandwidth costs.

OptiPNG performs lossless compression on PNG images; that is, it tries to reduce their file sizes using various compression algorithms and without removing any pixel data. The end result looks exactly the same as the original image. Using it is very simple:

```
optipng file.png
```

OptiPNG spits out some information as it works, overwrites the original file with the smaller one, and shows you the reduction percentage. We did a bunch of tests using screenshots from this very FOSSpicks section, as generated by Gimp. Occasionally we saw impressive reductions of 25–30%, but by and large the shrinkage was in the 10–15% range.

"Even the smallest reductions can make your website faster."



Here's the FOSSpicks screenshot from the Pybik review, now 14% smaller thanks to OptiPNG.

While the default options do a decent job, OptiPNG has some extra settings for choosing the optimisation level and PNG delta filters. The program's website also has an excellent explanation of how PNG optimisation works under the hood: **http://tinyurl.com/a9wprt**.

PROJECT WEBSITE
http://optipng.sourceforge.net

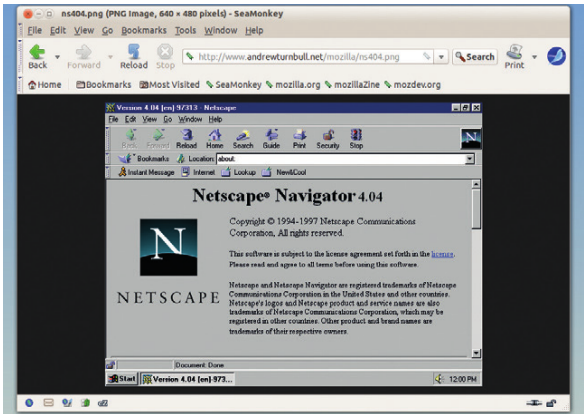
Internet suite

SeaMonkey 2.25

If you were around on the web in the late 90s, you'll certainly remember Netscape, a suite of programs including a browser (Navigator), email client and web page editor. After Netscape's demise, much of its source code was refactored by the Mozilla project, and today we see the results most famously in the form of Firefox. But while Firefox is a standalone browser, the internet suite project has also continued under the name SeaMonkey. In recent years it looked like SeaMonkey development was stagnating, but as more users become dissatisfied with Firefox the suite is getting more attention.

And it's really easy to try: grab the 33MB **.tar.bz2** file from the SeaMonkey website, extract it, jump into the resulting directory and run the **seamonkey** executable inside. That's it – you don't need to install it system-wide if you don't want to. (Note that the program stores its data in **.mozilla/seamonkey/** in your home directory.) Prepare for a blast of nostalgia when you first start the app, because the interface has hardly changed since Netscape 4. A few things have been removed or clumped together, but otherwise it's quite similar.

Because SeaMonkey shares the same underlying HTML and JavaScript engines as Firefox, it works largely the same for web browsing, albeit with a more traditional interface. But we like the fact that it has been consistent over the years, not whimsically integrating every questionable change from self-styled "user experience designers" (ugh).



Here's SeaMonkey from 2014, showing a screenshot Netscape 4.04 from 1997. Not a lot has changed, has it?

New features in version 2.25 include VP9 video decoding, support for Opus audio in WebM, and the Gamepad API (so that web games and apps can access USB or Bluetooth joypads).

"We like the fact that SeaMonkey has been consistent over the years."

PROJECT WEBSITE
www.seamonkey-project.org

Editor reborn "for the 21st century"

Neovim 2014-Mar-23

There's something of a Vi(m) theme to this month's FOSSpicks, what with Bvi's appearance on the facing page. Now we have Neovim, a fork of the regular Vim editor. But given that Vim is under active development, has masses of fans and a well respected lead developer, what kind of madman would want to fork it?

Well, that madman is Thiago de Arruda Padilha from Brazil, and he has some compelling arguments: Vim's codebase is old, complicated and full of cruft that could be removed, he says. It's difficult for new contributors to get involved. So it's time for Vim to undergo a major source code overhaul.

This isn't an easy task, and Padilha has set up a crowdfunding project to get started. At the time of writing, he had raised over $32,000 to fund his work on Neovim – an impressive sum, given that his original goal was $10,000.

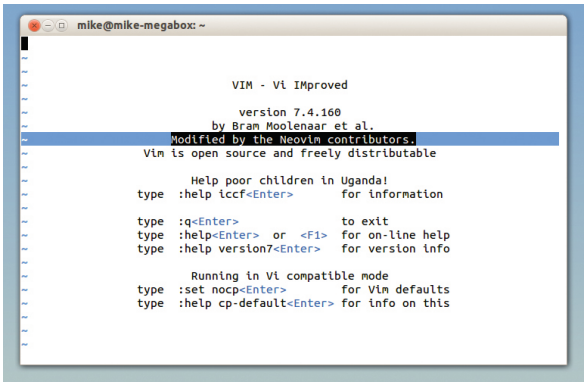
Plus ça change...

Although there's still a huge amount of work to be done on the editor, you can try it out right now to get a feel for it. To get the dependencies on Debian/Ubuntu:

```
sudo apt-get install libtool autoconf
automake cmake libncurses5-dev g++
```

Then grab the master **.zip** file from GitHub (**https://github.com/neovim/neovim**), extract it and run **make cmake && sudo make install** in the resulting directory. After compilation, you'll be able to run the editor with **nvim**.

Right now it looks, feels and smells like regular Vim – and that's the intention. Neovim won't look too different on the surface, as all of the important work will take place under the hood. Padilha wants Neovim to have a simpler build system, more platform-independent code, and a more versatile plugin system. It should also be easier to embed the editor into other programs, and create GUI front-ends for it on multiple platforms.



So far there's not much to distinguish Neovim from the original version, apart from an extra line on the startup screen.

PROJECT WEBSITE
www.neovim.org

Programming language

Python 3.4

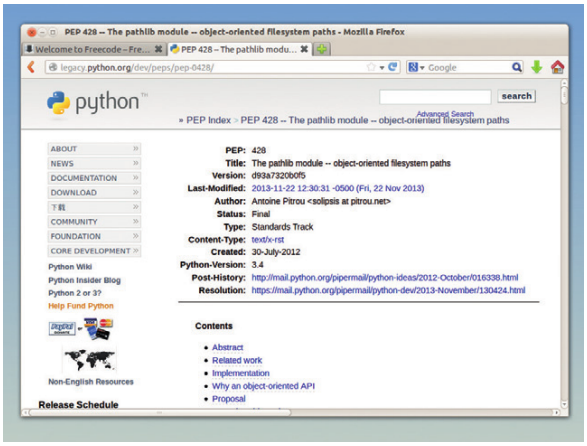
Maintaining a programming language is a tricky business. Python 3 was a bold step, breaking compatibility with Python 2 programs, but it was regarded as a necessary move to clean away a lot of cruft that had built up. There was plenty of controversy at the time, but Python 3 is maturing well, and now we have the 3.4 release. A bunch of new modules have been included, such as **pathlib**, which provides an object-oriented API for filesystem paths. For instance:

```
p = Path('/home/mike/foo/bar.py')
```

With this, **p.name** contains **bar.py**, **p.suffix** contains **.py**, and **p.parts** contains a tuple with every element in the path. You can join, split and compare paths, and query them (eg to find out if a path is relative). If you're doing cross-platform coding, there are ways to handle Windows paths as well, including the drive letter and colon combinations at the beginning.

Another new module is **asyncio**, which provides asynchronous I/O via a pluggable event loop and coroutines. Then there's **enum** (provides support for enumerated data types), **tracemalloc** (a debugging tool to trace memory allocation) and **ensurepip** (a cross-platform way to install the PIP package manager into an existing Python setup).

Many security improvements have been implemented as well: for instance, there's now TLS 1.1 and 1.2 support in the SSL module. These are just the main new features in 3.4, and lots of work has been done on other modules to squish bugs and add general improvements. The changes have been very well documented too – something that's often lacking with new language release, so kudos to the Python team.



Many Python features begin life as PEPs, or "Python Enhancement Proposals", which are discussed and reviewed by the community.

> **"Python 3 is maturing well, and now we have the 3.4 release."**

PROJECT WEBSITE
www.python.org

Information organiser

TreeLine 1.9.4

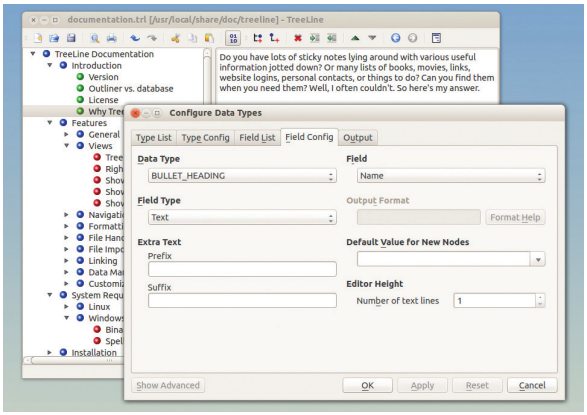
At the start of this issue's FOSSpicks we heaped spoonfuls of praise on plain text, but sometimes it can get out of hand. Take **notes.txt** for instance, the text file used for planning this section of the magazine: it started off well, with clearly defined sections and tidy presentation. But after a while it ended up as a morass of ideas, links, compilation instructions and other random bits and bobs. What we really need is a souped-up note taking and list compilation app – and thank $DEITY, TreeLine provides it.

Essentially, TreeLine could be described as an outliner, note-taker or PIM program. It's hugely versatile and isn't designed to work with specific types of information; it will handle anything you can put in a tree-like structure.

TreeLine 1.9.4 is a development release on the road to 2.0, and is a complete rewrite of the earlier 1.4 series. It's written in Python with Qt providing the interface, so you'll need PyQt installed to run it. When you start the app, you're presented with a two-pane view: the left-hand side contains the tree of items, and the right-hand side shows the data for individual items.

It's in the trees! It's coming!

The best way to understand TreeLine is to open the documentation, which, brilliantly, is made in TreeLine itself. Click Help > Full Documentation in the menu, and a new TreeLine window will open. Click the arrows on the left-hand side to open up nodes in the tree, and green or red items to read them on the right.



TreeLine stores its data in (optionally encrypted) XML format, and can export to HTML and plain text.

TreeLine supports custom data types with a range of fields (eg text, numbers, dates) so you can easily customise it for nigh-on any type of information. It's hugely configurable and didn't exhibit any major bugs in our testing, so from now on it's our go-to app for FOSSpicks planning.

PROJECT WEBSITE
http://treeline.bellz.org

FOSSPICKS Brain Relaxers

Flappy Bird-like game

Openflap 1.0

Frontier: Elite II provided a giant universe to explore, together with open-ended gameplay where you could trade, mine, fight or work for the military. It provided months of fantastic entertainment – and all on the Amiga. That's an incredibly rich game running on a 7MHz CPU backed with 1MB of RAM.

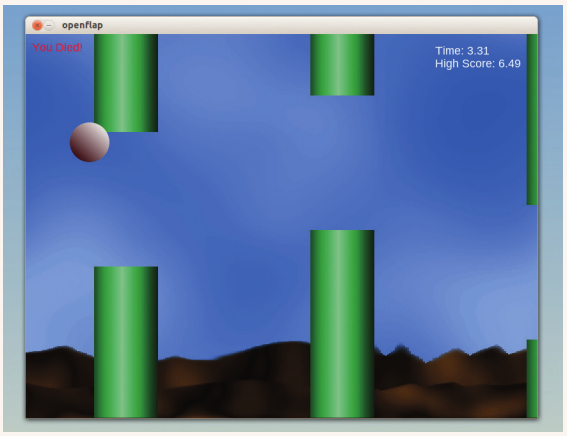
Today, people carry mobile phones that are several thousand times more powerful than the Amiga 600, and yet the most popular mobile games are so utterly trivial it makes us rage. Take Flappy Bird, for instance: it's brain-shutdowningly tedious, and yet it was making its author $50,000 a day at its peak.

Anyway, in true FOSS style, someone has come up with a free clone: Openflap. The gameplay is equally as minimal, and compiling it actually takes more brainpower (see the README file for instructions). The only dependencies you need are SDL and its various bolt-on libraries – so once you have it installed, just enter **openflap** to play.

Tapping tedium

If you've never seen Flappy Bird (or its million clones) before, here's how you play: tap Space. A ball is falling from the top of the screen, and tapping space bounces it upwards. But! The ball is also moving to the right, and you have to tap space to help it through gaps in pipes. And that's all there is to it.

Your high score is shown on the right, as an incentive to keep playing, but what really prompted us to include Openflap is the source code. It's clear and easy to read – a good resource for snippets and ideas if you want to make your own C++/SDL game.



Bounce the ball, miss the gaps – that's it. But the code is useful for learning how to do graphics, sound and input in SDL.

PROJECT WEBSITE
https://github.com/jazztickets/openflap

Rubik's Cube game

Pybik 1.1.1

Some people find Rubik's Cubes maddenly frustrating; others find them a good form of grey-matter stimulation. If you enjoy the odd session of "cubing" (as the cool kids call it) but you don't want to have a cube by your desk in case your boss thinks you're wasting time, then you can get by with a computerised version.

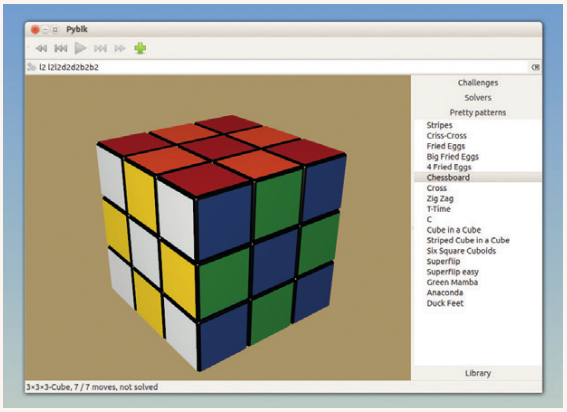
Pybik is written in Python, with its interface provided by Qt. So you'll need **PyQt4** to install it – see the **INSTALL** file in the tarball for full instructions.

Once you've built it, you can run it in place with **./pybik**. The presentation is gorgeous: the cube is rendered in shiny 3D, with lovely light effects and impressive detail to show the plastic parts onto which the coloured labels are stuck. OK, you might argue that visual frills are unimportant in a game like this, but if you're going to be staring at the cube for a while as you solve it, why not make it look good?

Pretty colours[3]

Click and drag on the cream area around the cube to rotate it, and hover the mouse cursor over a piece to see how it will rotate when you click it (the mouse pointer changes direction).

A number of challenges are available, eg prompting you to solve a cube in under 10 moves, while the "Pretty patterns" list generates funky-looking cube layouts for you to solve. There's also a library of moves along with some solving algorithms. It's a remarkably good substitute for a real cube, although you can't throw it against the wall when you get annoyed with it. Oh well.



The lighting effects are a nice touch, although at some angles the glare is a bit over the top.

PROJECT WEBSITE
https://launchpad.net/pybik