How often has someone commanded you to “Fetch me a shrubbery!” only to find that you’re all out of bushy vegetation? Well, now you never need to be caught short again, because you can grow your own.

It can all be done at a moment’s notice as well, so there’s no need to be fiddling around with genetic resequencing and John Innes compost Number 3 either. The ngPlant suite (for there is more than one executable here) creates and renders 3D models from a kind of fractal seed. You control the limits of its leafiness, the straightness of its stems and the fluffy frondy bits if you’re going to have those. The plant is grown from a stem (which has its own set of controls) to which are added branches, and sub-branches, should those be required of your creation.

Parameters for properties such as axial deviance, uprightness and suchlike can be tweaked until your vegetation is just how you like it – use the tree view (haha!) to select the component to change, then browse through the various available options using the tabs you’ll find at the top-right of the ngPlant interface.

It may be difficult to get the hang of at first, but the source also includes a samples directory with a few ideas in it, including the TGA texture files. Load them up and twiddle with the parameters (click and drag the red blobs for adjusting the curves) and you can turn the cactus into a fir tree in no time.

Seeding randomly
Adjusting the random seed on the main settings panel creates as many variants as you could wish for, and you’re then able to export the result in .obj format so that you can include your creations in other rendering software packages, such as Blender.

Unfortunately, it has to be said that building ngPlant isn’t particularly easy – or at least, it wasn’t for us. It uses the Python SCons system, which is no bad thing, but we did have to tinker with the script files to make it link to the appropriate libraries. If you don’t like the sound of this, you may be better off sticking with the stable version, which does build more reliably and is available from the same place on the project’s download pages.
Packet generator

Ostinato

Version 0.1.1 Web http://code.google.com/p/ostinato

Should you have a desperate need to find out what on Earth is going on on your network, then you need to look no further than the excellent packet sniffer, Wireshark. Highly configurable, it’s a useful tool for spotting intrusion attempts, finding out what network users are doing, mapping usage and debugging services.

Sometimes though, you need the exact opposite of something. If you find yourself thinking, “That Wireshark is a really great and awesome tool and all, but what I really need is the exact opposite of it,” you’ll shortly be very glad that you’re reading this fine and perhaps prescient page of quality magazine journalism. Because yes indeed, Ostinato is exactly that opposite you seek. Instead of snooping on packets sent from near or far, this useful tool actually generates them.

This is just as handy for debugging applications, testing that your network defences are working or finding out what happens in your workplace if you launch a DoS attack on the local Outlook server (but only try this if you already have a good job to go to).

Tidy packet

It actually integrates with Wireshark if you have it available — we suggest you install it now if you haven’t already got it. You can easily capture the exchange of packets to and from your target. A wide range of protocols and packet types are supported (you can even build your own if you want), which make it suitable for testing anything from web apps to network hardware. It would take too long to explain all the operations of the software here, but fortunately the author has created a getting started video tutorial, which you can see by visiting: www.youtube.com/watch?v=On64IYQEFYI.

A quick tip for building this app from source — you need to run qmake on the ost.pro file to generate a makefile, but make sure you use the Qt 4 version of qmake (if you have multiple ones installed) or else you won’t be able to build anything!

Checksum generator

PySum

Version 0.5 Web http://pysum.berlios.de/en

What do you do when you really need to make a hash of something? Top points if you answered “ask Mike to do it”, but what if you actually wanted to generate a hash checksum of some file? Checksums are very useful, primarily for authenticating and validating files. Depending on what distribution you’re using, you probably already have some form of hash-checking in your package manager anyway – it’s pretty standard kit, and we ran a tutorial about MD5, the most commonly used checksum generator, in LXF123.

But what if you want to send a file to someone and validate that the data arrived safely at its destination, or that it hasn’t been interfered with along the way? This simple little Python tool comes valiantly to your rescue, generating MD5, CRC32 and various flavours of the SHA algorithm.

Why do we need more than one hashing function? Well, if you just want to check that a file hasn’t been accidentally interfered with, you don’t, and the simplest hash function will do.

Seeding ran

However, a distinct problem arises when you want to find out whether a file has been intentionally altered (like, say, a distro package being doctored to contain some spyware or something like that). With a simpler hash scheme, such as MD5 for instance, it isn’t actually that difficult to generate an altered file and tweak it so that the hash keys match.

With more complex schemes, such as SHA, this becomes harder to do, and in the case of the higher bit-levels, almost impossible (though if someone wants to spend hundreds of hours of supercomputer time to have a go, I guess they could).
Everybody dreams of creating their own world, with lakes just where you want them, mountains that are just rugged enough, and maybe some fjords with fiddly bits. Well, maybe not everyone, but enough of you to keep downloading every terrain generator that ever thrusts itself forth on to the internet. This one is more worthy of your attention than most efforts.

The goal has been to create a kind of Linux version of the highly successful Terragen commercial software (for Windows and Mac OS X), which has been used in countless films, TV shows and print media. That software became popular due to the stunning images it could create with just a smattering of settings in a stylish and easy to follow GUI. Picogen might not have quite the same depth of functionality, but it certainly hits the mark on the swish interface and the relative ease of use.

Powerful rendering software is never going to be completely intuitive – there are just too many parameters to be tweaked and settings to be, er, set. Picogen has its fair share of tables full of double precision numbers with quasi-intelligible labels. Fortunately there are a few good tutorials included. Click on the appropriate button in the top-right of the panel and follow the instructions to create your own new-world masterpiece.

The objects with which you can fill your world are relatively simple for now. Well, simple but complex nonetheless. Both the ground objects and the water plane object can be altered by procedural code to generate fractal surfaces. Liberal use of 3D noise functions enables quite realistic and complicated sculpting behaviours, and you can superimpose objects to achieve your desired effect if necessary (say, rolling green plains with spiky mountains in the distance), or fiddle around with the shaders to create the desired effect.

**Render me good**

The render settings need some delving into. The default preview render is fine for a quick overview of the scene, but just increasing the resolution is going to give rather sketchy results, as most of the surface objects will contain a lot of noise. Erm, yes, they’re made up of noise, but you don’t want unintended surface noise. The trick here is to up the number of iterations. Combined with the larger image, this does mean huge render times – a 1024x768 image can easily take an hour or more to render on a 3GHz machine at good quality. Fortunately, the built-in tutorial has some guidance on the production render settings (and, of course, you can also have as many different render profiles as you like).

Also buried in the tutorial pages are some example scripts for creating different terrain. The syntax isn’t immediately explained, but simple experimentation and fiddling around with the values here should yield some knowledge of what is possible. If you want to mess around some more, the code is actually written in a language the author calls Jux, which is sort of similar to Scheme. There is some useful guidance in the ‘References’ section of the documentation, so check it out.

There is also a standalone editor for messing around with code, as well as a pretty swish editing environment – it’ll compile and run your code to produce a simple face-mapped sample of the output, so you can tweak fractal functions and see what they look like without too much of a time-delay.

Building Picogen is not altogether straightforward, but there are binary packages and builds for RPM and Deb, so you don’t have to worry about it too much. You will, however, need the latest version of libnoise. The release version of this hasn’t made it to all distros yet, but you can find the source here: http://libnoise.sourceforge.net.

Picogen isn’t the most amazing landscape rendering software we’ve ever laid eyes on, but it certainly has a lot of potential and deserves the effort it takes to understand it. And what’s not to like about building a world?

“A 1024x768 image can take an hour to render on a 3GHz machine.”

This looks like the place in Morocco where I lost my wallet.
Optical character recognition

Cuneiform

Version 0.1.0  Web http://tiny.cc/4dgyp

Back in LXF133 we took a look at Abbyy Finereader, probably the best known name in optical character recognition. Abbyy is a proprietary solution aimed primarily at business users, but if your needs are less exacting, it may be interesting to check out the Cuneiform project. The cuneiform code was originally proprietary, and developed by the Russian company Cognitive Technologies. Way back in 2008, the engine code was released under the BSD licence and, since then, development has continued as open source. The main code is still very Windows-oriented, but there is a Linux-loving spinoff that compiles the engine code, though not the graphical interface, so it’s command line only.

The results from Cuneiform are pretty impressive, if different from ‘conventional’ OCR. It tries to recognise letter shapes in a variety of different ways. The switch options to process dot-matrix or fax images really do improve the accuracy of the results, so remember to use them where appropriate. There are some sample images included for you to try out, but obviously it’s much better to try your own. A variety of input file formats are supported, but if you have the ImageMagick development files installed, Cuneiform will build against that and support practically any input type. Output can be HTML, RTF, plain text, or RTF with embedded Latex. A number of languages are also supported, but it defaults to English.

Building the app is via cmake, so you have to observe the usual construction of a build directory and run the cmake commands from there. Although this seems like a bulky app, it compiles very quickly and without much fuss.

If you’re interested in open source OCR software, it would be remiss of us to not mention the Tesseract project as well, which has been given a boost by a great deal of Google effort being put into the code.

“The results are pretty impressive, if different from conventional OCR.”

Temperature sensor

Psensor

Version 3.3.1  Web http://tiny.cc/05gw9

Now, many of you may not have had the chance to study thermodynamics, so let me explain some of the underlying concepts for you. Things that do work usually get hot (the corollary of this is that Mike often wears a jumper). You might think that heat is a good thing. Certainly in the bleak midwinter it’s cosy to toast your feet by the warmth of a few Linux boxes. But for the processors inside, heat is a very bad thing that they desperately want to get rid of. In fact, if they don’t, they heat up so much that they quite literally melt or burn out, perhaps taking out other expensive components at the same time. That’s why you have fans isn’t it? Yes, but fans fail. They get stuffed up with dust, coffee grinds, toenail clippings or whatever, or they just wear out. And suddenly, usually in less than a minute, your system is critical.

Many CPUs and GPUs have thermal shutdowns, as do some motherboards. So maybe you don’t need to be forever sticking a thermometer in the case to find out what’s going on. Of course, there are various monitoring systems available that hook into the sensors engineered into various bits of hardware, but they’re often bulky, process-slowing bits of code. This effort is a slimmed-down but functional temperature monitor that’ll draw a moving graph of any temperature component picked up by lm-sensors for example, or RTF with embedded Latex. A number of languages are also supported, but it defaults to English.

Building the app is via cmake, so you have to observe the usual construction of a build directory and run the cmake commands from there. Although this seems like a bulky app, it compiles very quickly and without much fuss.

If you’re interested in open source temperature sensor hardware, you’ll need to compile from source with extra options to enable support for NVidia sensor hardware.

“The results are pretty impressive, if different from conventional OCR.”

Psensor is a slimmed-down but functional temperature monitor.”

www.tuxradar.com

October 2010 LXF136 | 71
You cannae change the laws of physics. But you can use them to do some cunning things, like heat up your lunchtime noodles. You can also make a game out of them, as many people have done with varying degrees of success. The archetype of the genre is Kevin Ryan’s The Incredible Machine, which came out for MS-DOS back in 1992, and it’s saying something that the mechanics and playability of that title have rarely been bettered by its long line of imitators.

The Butterfly Effect is one such contender. The basics of the game are just the same – you have to arrange a series of everyday objects and trust the laws of physics (gravity being the one that plays the biggest part) to help them perform some goal. In this game, the goals are all very friendly and family-safe, so if you want to introduce children to their subservience to the gravitational constant, this is a good choice. The challenges in the game are introduced by notes you’ll find on-screen at the beginning of the level. Sometimes these provide helpful hints, but, to be honest, most of the included challenges are straightforward in terms of goals, if not exactly in terms of their solution. Thirty levels are included in the games, which get progressively harder as they involve more objects and more complicated scenarios. Sometimes the solutions are more simple than they appear, and sometimes you can get lucky with a shortcut of your own. The levels are built around XML files, so if you wanted to create your own it isn’t beyond the realms of possibility.

A prebuilt binary of the latest milestone release is available and ran OK on the systems we tried it on, so there’s no great need to build this game from source – just run the installer as a normal user and have it installed to your home directory.

There are a lot of bowling balls in this game, but very few skittles, sandwiches or pints of beer.

To be honest, at first glance I thought this game was called Violentland. How wrong I was. It’s about a nice young girl called Violet, so it’s actually not Violentland at all. Instead, it’s about helping a pretty young lady fire her multitude of weapons at an advancing horde of zombies, spiders and other denizens of this place, while skipping through their blood-soaked corpses to pick up powerups. How foolish of me!

There isn’t actually much more to the game than that. Move around with the W, S, A and D keys (her hips swivel a full 360, which I guess is very useful if zombies are coming at you from all sides), and try to keep a certain amount of distance between your character and the bad guys. Shooting them, for which you’ll need the mouse, helps immensely, and pressing F1 will show you some other useful options. If you manage to survive long enough, there are powerups and skills to be had, which make staying alive a tad easier. I really wish I could tell you more about them, but those pesky spiders have a habit of sneaking in from off-screen and nibbling my ankles. There’s one very useful piece of technology you should be aware of, though. Young Violet has managed to get hold of some futuretech in the form of a limited teleport device. Press Q and then use the mouse to indicate your desired location, and the laws of the universe will be torn asunder.

Made with SDL, this tiny game is actually quite fun. It’s still very much in a nascent stage of development, but the crew are a friendly bunch and will appreciate your feedback and any contributions you might make.
Windows API implementation

Wine

Version 1.2 Web www.winehq.org

Wine isn’t an emulator, but it does fulfil more or less the same function. It enables you to install and run software that was written for a rather different operating system. Behind the scenes it’s very clever – it implements (in clean, non-copyright code) all the API functionality of Windows. As far as any software you’re running knows, it’s actually running on Windows, so the great majority of system-friendly applications will run without any problems.

You probably already have Wine installed, and if not, it’s almost certain to be available as a package for your distro. But if you have a special need for running Windows software, this milestone release is well worth checking out. Among other features is the ability to run 64-bit Windows apps – a major achievement that has taken a fair bit of development work. Other new features include better support for bi-directional fonts, extra support for the newest MS system libraries and dozens of cosmetic improvements that make for a better and faster all-round experience.

Wine used to be just for boring commercial apps that you had to run to do some specific task, but the number of improvements in recent releases, especially this one, make it just as excellent for running fun stuff too. If you use Wine at all, you really need to get this version right now.

Google hasn’t ported Sketchup to Linux? Never mind, have some Wine instead…

GIF animator

GiftedMotion

Version 1.20 Web www.onyxbits.de/giftedmotion

Whether you believe the web would be more wondrous without them or not, it’s a fact of modern life that at some point you’ll have to create an animated GIF file for some website or other.

The fundamentals of animating a GIF aren’t tricky – the image data is just stored as successive deltas with some timing information. But putting one together can be a bit of a pain. You can do it, but loading all those layers in is a bit of a pain and, if it doesn’t work quite how you thought, changing it is a pain.

This very simple Java application makes a simple and straightforward job just that. Run it and load in the frames of your image; you can easily rearrange them and see a preview of what you have so far in the main panel. Shuffle them around, change the timing and when you’re happy, just press the big red button to save out the animation. It doesn’t really need complicated controls – it’s a simple function that it performs and it does it pretty painlessly. If you do need to bash out the occasional animation, it’s well worth downloading this teensy tiny app to save you some time.

This is a straightforward Java app, supplied as a JAR file so there’s no need for compiling or any fancy installing tricks – you can just run it from the commandline with java -jar giftedmotion-1.20.jar.

Knit one, Perl… I mean Purl one…

PyQt 4.7

Important update to the Python binding for Qt adds support for new Qt features.

www.riverbankcomputing.com/software/pyqt/intro

Pinot 0.96

D-BUS enabled indexer with a GTK interface for file searching.

http://pinot.berlios.de

gpsim 0.25.0

Simulator for Microchip PIC microcontrollers now supports even more models.

http://gpsim.sourceforge.net/gpsim.html

ClamTK

The GUI for the ClamAV antivirus engine gets an overhaul.

http://clamtk.sourceforge.net

New and updated software that also deserves a look...

Poppler 0.14.1

Indispensable PDF rendering library gets tweaked performance for ObjectStream docs.

http://poppler.freedesktop.org

Dream Pie 1.1

Interesting new take on a Python interactive shell.

http://dreampie.sourceforge.net

Coffee Saint 3.38

Nagios helper app now includes LiveStatus support.

http://vanheusden.com/java/CoffeeSaint

Paludis 0.48.3

Package manager for Gentoo gets some serious updates.

http://paludis.pioto.org

Knitter 0.54

Included in our last issue, now with major interface updates, including ‘stitch-in-place’.

http://sourceforge.net/projects/knitter

PyQt 4.7

Important update to the Python binding for Qt adds support for new Qt features.

www.riverbankcomputing.com/software/pyqt/intro

Pinot 0.96

D-BUS enabled indexer with a GTK interface for file searching.

http://pinot.berlios.de

gpsim 0.25.0

Simulator for Microchip PIC microcontrollers now supports even more models.

http://gpsim.sourceforge.net/gpsim.html

ClamTK

The GUI for the ClamAV antivirus engine gets an overhaul.

http://clamtk.sourceforge.net