Full Circle

THE INDEPENDENT MAGAZINE FOR THE UBUNTU LINUX COMMUNITY

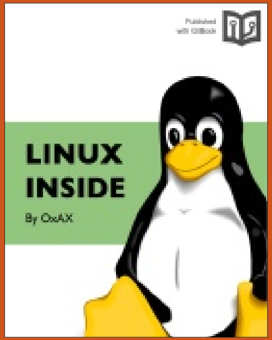
ISSUE #152 - December 2019

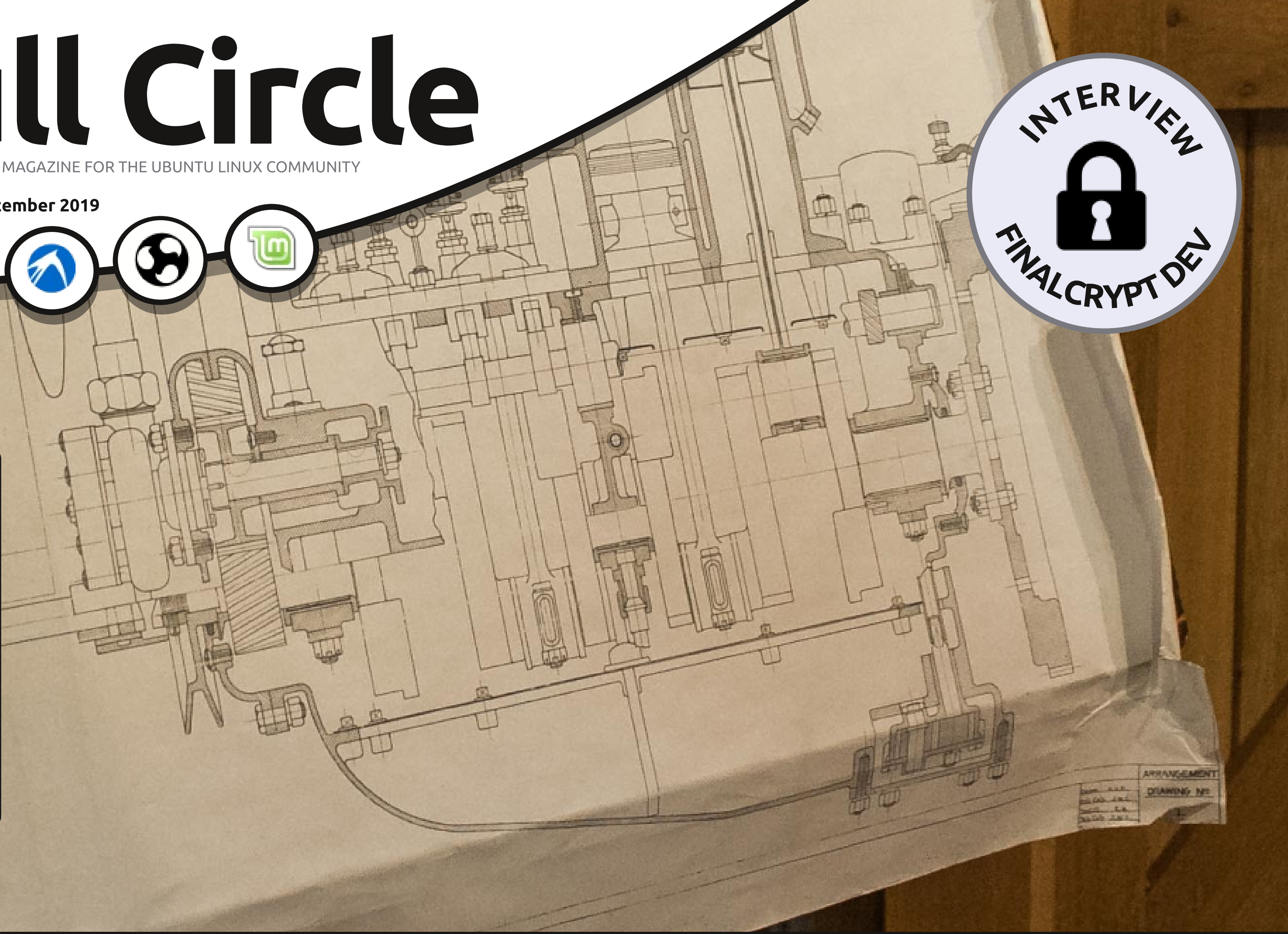
INTERVIEW

FINALCRYPT DEV

Photo: Jez (Flickr.com)

BOOK REVIEW





COMPUTER AIDED DESIGN

SETTING CAD SIZES FOR PRINTING

Full Circle Magazine is neither affiliated with, nor endorsed by, Canonical Ltd.

Full Circle

THE INDEPENDENT MAGAZINE FOR THE UBUNTU LINUX COMMUNITY

SOME RIGHTS RESERVED

The articles contained in this magazine are released under the Creative Commons Attribution-Share Alike 3.0 Unported license. This means you can adapt, copy, distribute and transmit the articles but only under the following conditions: you must attribute the work to the original author in some way (at least a name, email or URL) and to this magazine by name ('Full Circle Magazine') and the URL www.fullcirclemagazine.org (but not attribute the article(s) in any way that suggests that they endorse you or your use of the work). If you alter, transform, or build upon this work, you must distribute the resulting work under the same, similar or a compatible license.

Full Circle magazine is entirely independent of Canonical, the sponsor of the Ubuntu projects, and the views and opinions in the magazine should in no way be assumed to have Canonical endorsement.

EDITORIAL

Welcome to the last issue of Full Circle*

Yes, it is indeed the last issue of FCM*. But fear not we have the usual suspects this month; Python, Darktable, and Inkscape, and not forgetting the continuation of editing photos with Krita. Yes, Krita. Not GIMP.

This month also marks a combo I couldn't have planned if I even tried. Goodbye to Lucas. This month is his last article. I wish him all the best, and he's obviously more than welcome to submit articles when he has free time. Congratulations to Greg. This month is where he reaches 100 Python articles! I've no idea how he's managed to put up with me for this long. Here's to another 100!

Last month I put forth the idea of removing the news section from the magazine. Surprisingly, people did actually email me. So, the news section gets a stay of execution. Y'see, I've no idea what you folks like/dislike unless you email me and tell me what you like/dislike.

All the best to you and yours for 2020!
Ronnie
ronnie@fullcirclemagazine.org

* last issue of 2019 that is

FCM PATREON : https://www.patreon.com/fullcirclemagazine

This magazine was created using :

Find Full Circle on:

facebook.com/fullcircle magazine

twitter.com/#!/fullcirclemag

http://issuu.com/fullcircle magazine

https://mastodon.social/ @fullcirclemagazine

Weekly News:

http://fullcirclemagazine.org/feed/podcast

http://www.stitcher.com/s?fid=85347&refid=stpr

http://tunein.com/radio/Full-Circle-Weekly-News-p855064/

NEWS

Submitted by ErikTheUnready

New version of Zulip released:

Zulip 2.1 provides the benefits of real-time chat, while also being great at asynchronous communication. Inspired by email and working like slack, Zulip aims to be your all-in-one communication application for teams. Highlights in the new version include: a new public archive tool and a Digital Ocean one click installer. There are new data import tools, new export tools and it has been updated for newer distro versions, while removing support for EOL Ubuntu 14.04 Trusty.

https://blog.zulip.org/2019/12/13/zulip-2-1-released/

Russian police raid the offices ofNginx.

Nginx creator Igor Sysoev, who was an employee of Rambler almost 20 years ago, at the time wrote the code for what would become the open-source Nginx web server platform. He claims he wrote the software in his spare time, and thus it belongs to him, though Rambler appears to disagree and has claimed ownership of the code. Apparently the statute of limitations in Russia is 15 years, so no-one is sure what is going on. Igor was arrested along with Maxim Konovalov. They have since been released. The search ruling indicated that Nginx is the intellectual property of Rambler, which was distributed unlawfully as a free product, without the knowledge of Rambler, and as part of a criminal intent. The damage from the publication of Nginx is estimated at 51 million rubles. BTW, Nginx is now owned by F5 networks. You can follow the story as it unfolds on twitter: @AntNesterov

https://www.zdnet.com/article/russian-police-raid-nginx-moscow-office/

Version 6.1 of Cocoss Reaper is out:

A new version of Reaper is out that now supports theming.Reaper has experimental native support for Linux, and the Windows version works well with WINE.It is great that commercial music creation entities are now supporting Linux. Reaper supports lots of plug-in's and it could become your favourite DAW. The new version comes with a ton of improvements.

http://reaper.fm/index.php

Nethack 3.6.3 released:

If you are into gaming at the terminal, you will be glad to know there is a new version of nethack released. Those that don't know,NetHack is a single player dungeon exploration game that runs on a wide variety of computer systems, with a variety of graphical and text interfaces all using the same game engine. The aim being exploration, not hacking everything to pieces. Over 190 bug fixes and over 22 game enhancements and community contributions made since the release of 3.6.2 in May this year.

https://nethack.org/v363/release.html

Vim 8.2 is released:

Vim 8.2 is a minor release, though a lot of bugs have been fixed and the documentation was updated. There are a few interesting new features and a silly game, featuring the ugliest sheep you have ever seen. It is becoming more like Emacs every day.

https://www.vim.org/news/news.php

DX9VK gets a new release:

Version 0.40 - Croakacola with bug fixes galore and features like implementation of the ability to use more than 4GB VRAM on 32-bit applications.This helps greatly in modded Skyrim/Oblivion, etc.

Performance fixed in Risen and Legend of the Heroes: Trails of the Sky, as well as minor performance tweaks under the hood.

https://github.com/Joshua-Ashton/d9vk/releases/tag/0.40

Flowblade 2.4 released:

Flowblade has transitioned to Python 3. The editor provides new tools for cropping clips to the accuracy of individual frames, processing them using filters and multi-level image composition for embedding in video. It is possible to arbitrarily determine the order of application of tools and adjust the behaviour of the timeline.Work was done to increase the quality of images and the availability of tools in compositing.

https://github.com/jliljebl/flowblade/releases/tag/v2.4

Virtualbox brings us 6.1:

Packages are available for Linux (Ubuntu, Fedora, openSUSE, Debian, SLES, RHEL in assemblies for the AMD64 architecture), Solaris, macOS, and Windows.The list of changes are as long as my arm, but a lot of work has been done on the display, CPU support, USB and paravirtualization. You can read more on the Oracle virtualbox website as this is a major update.

https://www.virtualbox.org/wiki/Changelog-6.1

Release of Qt 5.14 Framework and Qt Creator 4.11.0 Development Environment:

With QT6 on the horizon, QT 5.14/15 should be the last in this line of the current architecture, with better Wayland support and integration. At the same time , the release of the integrated development environment Qt Creator 4.11.0 , designed to create cross-platform applications using the Qt library, was released. It supports both the development of classic C ++ programs and the use of the QML language, which uses JavaScript to define scripts, and the structure and parameters of interface elements are set by CSS-like blocks.

https://www.qt.io/blog/qt-5.14-has-released

New release of Xine 1.2.10:

The latest version of xine-lib 1.2.10, a multi-platform library for playing video and audio files, as well as a set of related plug-ins, has been released. The library can be used in a number of video players, including xine-ui, gxine, kaffeine.It now supports EGL and Wayland as well as Android. New features like multithreading and new decoders for new codecs are included in this release.

https://sourceforge.net/projects/xine/

Apache SpamAssassin 3.4.3 has been released!

Apache SpamAssassin 3.4.3 contains numerous tweaks and bug fixes. There are a number of functional patches, improvements as well as security reasons to upgrade to 3.4.3. In this release, there is also one new plugin and there are bug fixes for two CVEs: 12420 / 11805 of this year.

SpamAssassin developers also announced the preparation of the 4.0 branch, which will implement full-fledged embedded processing of UTF-8. On March 1, 2020, the publication of rules with signatures based on the SHA-1 algorithm will also be stopped (in release 3.4.2, the SHA-256 and SHA-512 hash functions replaced the SHA-1).

https://spamassassin.apache.org/news.html

Codeweavers Crossover 19 released:

Most of this version’s code was dedicated to MacOS, as

Apple have said they would be dropping 32-bit support. Jeremy White announced: “I am excited that we have released CrossOver 19 and we are providing support for 32 bit Windows applications on an operating system with no 32 bit libraries - our own Christmas Miracle.” For us Linux users, CrossOver 19.0 has better handling of Microsoft Office and other improvements but not as prominent as the Apple macOS improvements.

https://www.codeweavers.com/about/blogs/jwhite/2019/12/10/celebrating-the-difficult-the-release-of-crossover-19

Open CASCADE Technology 7.4.0:

Open CASCADE Technology (OCCT) is a software product with a twenty-year history, combining a set of libraries and software development tools focused on 3D modeling, especially computer-aided design (CAD) systems. OCCT is primarily the core of geometric modeling. Open CASCADE Technology is the core or an important component of programs such as FreeCAD, KiCAD, Netgen, gmsh, CadQuery, pyOCCT and others. Open CASCADE Technology 7.4.0 includes more than 500 improvements and fixes compared to the previous version 7.3.0, which was released a year and a half ago. All of the shiny new stuff is presented in a PDF document @ https://www.opencascade.com/sites/default/files/documents/release_notes_7.4.0.pdf

https://www.opencascade.com/

OpenVSP 3.19.1:

OpenVSP is a free parametric CAD system for design and analysis of aircraft geometry (CFD, FEM). The program is developed by NASA Langley Research Center employees and is included in the NASA Software Catalog software list. Most of the changes were aimed at improving cross-platform, accuracy of calculations and stability. The OpenVSP development branch 3.19.x includes three of the most anticipated innovations: VSPAERO 6.0.0 , Generic XSec Editor and auto- generated API documentation using Doxygen. In addition, extensive work was carried out to improve and correct errors. A significant part of this work was carried out by the ESAero team, sponsored by the US Air Force Research Laboratory.

In addition to all the changes in the program, Ubuntu users 18.04 can now download the DEB package (thanks to Cibin Joseph for the work done for packaging), and a 64-bit EXE is also provided for Windows users.

http://openvsp.org

Exim 4.93 release

The release of the Exim 4.93 mail server, was announced. This represents ten months of hard work Exim, that has twice the user base of Postfix, so it is quite popular. Besides fixing bugs and improving stability and performance a bunch of new features were added. (like router variables, ehlo transport event, JSON lookups, and more).

https://lists.exim.org/lurker/message/20191208.213349.3407a963.en.html

SuperTux 0.6.1 Release

After a year of development, the release of the classic platform game SuperTux 0.6.1 , reminiscent of the style of Super Mario, is available for download . The game is distributed under the GPLv3 license and is available in assemblies for Linux (AppImage), Windows and macOS. Other than Bug fixes and optimizations, there is an improved story mode. In the story mode, the Ghost Forest level has been added. There are 3 new bonus worlds for you to enjoy also. There are also new enemies. Listen to the new music and enjoy the new backgrounds when you play this fun platformer.

https://www.supertux.org/news/2019/12/15/0.6.1

QEMU 4.2 emulator release:

As an emulator, QEMU allows you to run a program compiled for one hardware platform on a system with a completely different architecture, for example, run an application for

ARM on an x86-compatible PC. In virtualization mode in QEMU, the performance of code execution in an isolated environment is close to the native system due to the direct execution of instructions on the CPU and the use of the Xen hypervisor or KVM module. The project was originally created by Fabrice Bellard in order to enable Linux compiled for the x86 platform to run Linux executables on architectures other than x86. Over the years of development, support for full emulation for 14 hardware architectures was added, the number of emulated hardware devices exceeded 400. In the preparation of version 4.2, more than 2200 changes from 198 developers were made.

https://lists.nongnu.org/archive/html/qemu-devel/2019-12/msg02579.html

Interlink Version 7282:

New release based on the Palemoon 28.8.0 release was announced. Massive MailNews Core refresh with over 100 changes including security, stability, and performance. The update is mainly under-the-hood, as most of them are library updates. Included in this update is support for Illumos, the Solaris-like operating system.

https://lists.nongnu.org/archive/html/qemu-devel/2019-12/msg02579.html

Release of Mesa 19.3.0:

The first release of the Mesa 19.3.0 branch has been released - after the final stabilization of the code, a stable version 19.3.1 will be released. Mesa 19.3 provides full support for OpenGL 4.6 for Intel GPUs (i965, iris drivers), support for OpenGL 4.5 for AMD GPUs (r600, radeonsi) and NVIDIA (nvc0), as well as Vulkan 1.1 support for Intel and AMD cards. Changes to support OpenGL 4.6 were also added to the radeonsi driver, but they were not included in the Mesa 19.3 branch. The 19.2 branch will receive it's last update this year.

https://lists.freedesktop.org/archives/mesa-announce/2019-December/000563.html

Vulnerability in NPM that could modify arbitrary files during package installation:

The update of the NPM 6.13.4 package manager, included in the delivery of Node.js and used to distribute modules in JavaScript, eliminated three vulnerabilities (CVE-2019-16775 , CVE-2019-16776 and CVE-2019-16777), which allow to modify or overwrite arbitrary system files when installing a package prepared by an attacker. As a workaround, you can install with the option --ignore-scripts, which prohibits the execution of built-in handler packages. NPM developers analysed the packages available in the repository and did not find traces of using the identified problems to carry out attacks.

https://blog.npmjs.org/post/189618601100/binary-planting-with-the-npm-cli

Jonathonf's PPA's removed in protest:

In a statement on launchpad, he said: "I will be removing most of my PPAs from public access due to continued and persistent abuse by companies using these packages for commercial gain with flagrant disregard to the knowledge and effort required to maintain them."

Now we are not sure if Jonathon is unaware of corporate greed or does not understand GPL v3. The GPL does not prohibit you from charging corporate clients.

https://launchpad.net/~jonathonf

Krita Receives Epic MegaGrant:

Epic Games donated $ 25,000 to the Krita graphics editor, which is being developed for artists and illustrators. The editor supports multi-layer image processing, provides tools for working with various color models and has a large set of tools for digital painting, sketching and texture formation. The money will be spent on developing the next stable release of Krita. The donation was made as part of the Epic MegaGrants initiative, a $ 100 million grant fund for game developers, content creators and toolkit developers related to the Unreal Engine or open source projects useful to the 3D community. Previous donations include donations to Blender and Lutris.

https://krita.org/en/item/krita-receives-epic-megagrant

Gentoo developers are considering preparing binary builds of the Linux kernel:

Gentoo developers are discussing the provision of generic Linux kernel packages that do not require manual configuration during assembly and are similar to the kernel packages provided in traditional binary distributions. As an example of the problem that arises when using manual tuning of kernel parameters practiced by Gentoo, there is a lack of a unified set of default options that guarantees operability after updating. The advantages of manual kernel tuning include the ability to fine-tune performance, eliminating unnecessary components during assembly, reducing assembly time and reducing the size of the resulting kernel.

https://blogs.gentoo.org/mgorny/2019/12/19/a-distribution-kernel-for-gentoo/

GNUnet 0.12, a framework for building secure P2P networks:

Gnunet is designed to build secure, decentralized P2P networks. Networks created using GNUnet do not have a single point of failure and are able to guarantee the inviolability of users' private information, including eliminating possible abuses by special services and administrators with access to network nodes. The release is marked as containing significant protocol changes that violate backward compatibility with versions 0.11.x. GNUnet supports the creation of P2P networks over TCP, UDP, HTTP / HTTPS, Bluetooth and WLAN, and can work in F2F (Friend-to-friend) mode. NAT bypass is supported, ncluding using UPnP and ICMP. A distributed hash table (DHT) is used to address the location of data. Means for deploying mesh networks are also provided.

https://gnunet.org/en/

Mozilla will switch from IRC to Matrix and add a second DNS-over-HTTPS provider to Firefox:

Mozilla decided to switch to using a decentralized communication service for developers built using the open Matrix platform . It was decided to launch the Matrix server using the Modular.im hosting service. The technical obsolescence of the IRC protocol was noted. We can also note the addition to Firefox of an alternative provider for DNS over HTTPS (DoH, DNS over HTTPS). In addition to the previously proposed default CloudFlare DNS server, NextDNS service will also be included in the settings, which also offers a proxy of the same name for DoH. You can select the provider in the network connection settings.

https://discourse.mozilla.org/t/synchronous-messaging-at-mozilla-the-decision/50620

Linux Mint 19.3 released:

This is the second update of the Linux Mint 19.x branch, based on Ubuntu 18.04 LTS and supported until 2023. The distribution is fully compatible with Ubuntu, but differs significantly in the approach to organizing the user interface and the selection of default applications. Linux Mint developers provide a desktop environment that matches the classic canons of organizing the desktop, which is more familiar to users who do not like the GNOME 3 interface. Mate, Cinnamon and XFCE flavours are available for download.

https://blog.linuxmint.com/?p=3834

NetworkManager 1.22.0 released:

A new stable interface release has been published to simplify network settings - NetworkManager 1.22 . Plugins for supporting VPN, OpenConnect, PPTP, OpenVPN and OpenSWAN are developed as part of their own development cycles.

https://wiki.gnome.org/Projects/NetworkManager

Full Circle Weekly News

Join our new host Leo Chavez as he presents you with a short podcast (<10min) with just the news. No chit-chat. No time wasting. Just the latest FOSS/Linux/ Ubuntu news.

RSS: http://fullcirclemagazine.org/feed/podcast

AUDIO MP3

contents ^

COMMAND & CONQUER

Written by Lucas Westermann

Rust

This month marks the final Command & Conquer article I'll be writing. For more details on why, you may want to look at last month's article. That being said, I wanted to do something a little different for the last article. The first part of the article will be dedicated to some articles I'm most proud of having written, and the second half will be dedicated to writing a GraphQL API to track my Go games. So if you're interested in one but not the other, you know where to jump to.

Part 1

I've been writing for FCM since issue #21 - 131 issues ago! Over that time I've written some articles that are, as of now, obsolete, and some that hold up to the test of time. Below you'll find a list of my favorite articles that I've written, and what issue they appeared in:

- CLI Cookbook - FCM #76. I'm most proud of this one because we managed to get the community involved and actually created something together. I can't guarantee that all the commands are still accurate, but I'm sure there are still a good few ideas that are valid. The actual PDF/LaTeX documents can be found here: https://github.com/lswest/cli-cookbook
- Flexbox Stylus - FCM #92. This was another fun little project I wrote for myself that yielded a great article. I built a set of helper functions for Stylus to easily create/manage Flexbox settings. Not terribly useful in this day and age, but still fun.
- Tailwind CSS - FCM #134. This article introduced my readers to a tool that completely changed my approach to designing and styling websites, and is a method I still use to this day. Definitely a worthwhile read to anyone who's interested in web development.
- My web development articles. I won't list all the issues I had web development focused articles in (though there will be a few at the end of this item). The reason I'm proud of these articles is quite simple - I both enjoyed the topic, and used the knowledge in my professional life (I still do!). In writing those sorts of articles, I always hoped to make the entry into new web technologies easier for beginners. Noteworthy articles: Gatsby Multi-Language (151), AMP (127), CSS Grids (125), Static Site Generation (103).

There are other articles on a wide range of topics - guitar, note taking, virtualization, etc. Unfortunately, I don't have a complete list of articles anywhere for easy browsing. If any readers have something like that, they're welcome to email it to me (address below).

Part 2

Now, on to other topics near and dear to my heart: Go & GraphQL.

For anyone not familiar with Go, it's an ancient chinese board game (estimated at over 2500 years old), played with black and white stones on a 19x19 grid. It's also known as Baduk or Weiqi in Korea and China, respectively.

GraphQL is a (much) more recent invention. It's a query language for APIs that define a schema of data, and allow flexible querying for information. Basic example - you could define a schema for a book and an author, and keep track of things like ISBN, number of pages, publishing date, author, title, etc. Anyone who has access to the API can, using the same URL, selectively query only the data they want (i.e. title, author, and cover page) instead of getting everything back every time. It's the backend to Gatsby's static site generation (controlled via the gatsby-node.js file), and is extremely powerful. Ever since using it for the first time, I've wanted to create my own GraphQL API to replace my aging Ruby on Rails application that I use for tracking movies and video games I want to see/buy. I have since converted the information I already had (stored in a sqlite database from Rails) into mongodb, and written the API to the point where it can access and create entries in the database. Now it's time to expand the

contents ^

functionality - adding in my Go games. I will not be covering the frontend aspect (planned to be a Gatsby PWA that hydrates data on load), as it's not been completed yet, and GraphQL is flexible enough that you can access it from pretty much anything.

All code has been placed into a Gist here: https://gist.github.com/lswest/d2118f4fa0225b80993acb7337fdefc2

I will be linking to individual files throughout the article! So there's no need to grab them all now.

The Basics

I set up my API using Express.js, mongoose, apollo-server, and apollo-server-express. Most aspects will remain the same regardless of implementation, but the actual connection to the database will differ.

How does a GraphQL API work? You define a few schemas (think of it as a class definition) for queries, types, and mutations. Mutations are the create/update/delete aspect of CRUD, and queries are the "read" aspect. I won't go into detail on the mutations, just a basic create function.

GraphQL then takes your defined schema and uses it for validation, typing, and for understanding the requests sent to it. The schemas also control which fields from your database are available in the API.

Basic folder structure:

```
/src/models/
/src/schemas/
/src/resolvers/
/src/index.js
/package.json
```

Requirements

Make sure you've installed NodeJS (the LTS should be sufficient if you don't want to be on the faster moving stable branch), mongodb (or your database system of choice), and have some test data prepared (for example a JSON block to import into mongodb or to hard-code into the app).

To get the project up and running, you can do the following (if you prefer npm, all yarn commands have npm equivalents):

```
yarn init

yarn add -D nodemon
@babel/core @babel/node
@babel/preset-env
```

Create a .babelrc file with:

```
{
 "presets": ["@babel/preset-
env"]
}
```

yarn add mongoose express graphql apollo-server apollo-server-express

Add the following script to your package.json:

```
"dev": "nodemon --exec babel-
node src/index.js"
```

Using Compass or mongo's CLI, be sure to create a database to store your data in if you want to use a database.

Step 1 Mongoose Schema

For a mongodb implementation with mongoose, you define a mongoose.Schema (separate from the GraphQL Schema). Here you're essentially defining the document structure to be stored/loaded from the collection(s).

My Schema for Go looks like this:

/src/models/goGames.js: https://gist.github.com/lswest/d2118f4fa0225b80993acb7337fdefc2#file-models-gogames-js

Basic explanation

I defined fields for a 'go' game to include Title (i.e. Lucas VS George), the date played (currently defined as a String, as I haven't yet figured out how to make dates work correctly), what server it was played on (KGS, IGS, FGS, online-go, etc), Black and White player names, Komi (the points given to White for going second), Result in the traditional notation - i.e. B+Res, and MyWin which tracks if I won this game (for statistics later on) - if I were to add someone else's game, I'd simply leave this as false, and SGF. I tend to download my games' SGF files and store them somewhere on my PC. While I won't necessarily link them all on a web server, I can at least track the name. If I do eventually add them in as static files, I can then just update them to links.

The collection defines what I want the collection to be called in mongodb (currently, the collection does not exist - so I could have

contents ^

chosen anything here). You then apply the schema to a model, and export the resulting variable to use later on.

Step 2 GraphQL Schema

Once we've defined our mongodb server, we need to define our GraphQL schema. You should base the schema off your database definition, but it does not have to be a one-to-one match.

The GraphQL Schema I defined looks like this:

/src/schemas/goGames.js: https://gist.github.com/lswest/d2118f4fa0225b80993acb7337fdefc2#file-schemas-gogames-js

The GoGame type is a match for the mongoose Schema, and the createGoGame mutation takes pretty much all the fields.

The queries, however, are specialized. The first query (goGame) can only be filtered by ID and/or title, as it returns a single instance it makes sense to be as restrictive as possible to avoid weird results. The allGoGames query can be filtered using pretty much all fields except Komi and Result. As my goal for this API is to track my own games, I'm more likely to search for games where I was black or white, and perhaps define if it was a win or a loss. I don't think I'll ever search for all games where Komi was 0.5, for example. If I end up needing this, I can simply add it in as an option. Similarly, I won't necessarily be filtering by result, as I'll never (at that point) know which player was which. The field is important for a quick overview, but shouldn't be very useful when filtering what I want to see. I also added a Limit field to the allGoGames, to limit the number of results returned.

Step 3 Resolvers

Okay, we've now defined our schemas and given some thought to the options available in a query. However, until we define our resolvers, the query won't work. A resolver is a function that defines what happens with the parameters we defined in our schema. For my Go games, it looks like this:

/src/resolvers/goGames.js: https://gist.github.com/lswest/d2118f4fa0225b80993acb7337fdefc2#file-resolvers-gogames-js

Admittedly, almost all my resolvers look like this, with the only difference being variable names and the models used. The goGame resolver is the simplest - I take any of the args passed through (Title or _id), and then run a findOne on the collection.

The allGoGames resolver is more complicated. I pass in all the args, including a field called Limit. The idea behind 'limit' is to set a maximum number of results (ie. if I want a top 10). As this field doesn't exist in the mongodb document, it will never yield results if it's just passed in that way. Instead, I check if args has a property 'Limit'. If it does, I create a copy of the object and delete the 'Limit' property. I then adjust the mongodb command to pass in the remaining arguments and use args.Limit in the .limit() function. If args.Limit doesn't exist, I just run a find() on all the args.

The createGoGame resolver takes all the arguments I specified in the GraphQL Schema. However, it also needs an id. Instead of forcing the user or client to generate one, I instead add an _id field to the object using mongoose.Types.ObjectId() before creating the item.

Step 4 - Putting it all together

The first thing I would recommend you do is create an index.js file in both /src/schemas and /src/resolvers. This file will serve as an aggregator for all your schemas and resolvers once you have more than one.

/src/schemas/index.js: https://gist.github.com/lswest/d2118f4fa0225b80993acb7337fdefc2#file-schemas-index-js

/src/resolvers/index.js: https://gist.github.com/lswest/d2118f4fa0225b80993acb7337fdefc2#file-resolvers-index-js

Now for the heart of the server:

/src/index.js: https://gist.github.com/lswest/d2118f4fa0225b80993acb7337fdefc2#file-src-index-js

Be sure to replace {MONGO_URL} with your actual mongodb connection string (most likely mongodb://localhost:27017/{database}), where {database} is whatever database name you defined manually in step 1.

Step 5 - Trying it out

Once you've started the server with yarn dev, the server should be running on localhost:5000. However, the root doesn't return anything as we only defined the path "/graphql". So head on over to http://localhost:5000/graphql and have a play around on your graphiql instance.

To create items:

```
mutation {

createGoGame(Title:"Example",PlayedDate:"2019-12-06",Server:"Fox",Black:"Player1",White:"Player2",Komi:"7.5",Result:"B+Res",MyWin:false,SGF:"2019-12-06 - example.sgf") {
    _id
   }
}
```

The above will generate an entry and return the id for you to use in a goGame query.

To query items:

```
{
   goGame(_id: “id from above”) {
        Title
     }
}
```

To see them all, you can also run:

```
{
   allGoGames {
        Title
     }
}
```

So, I hope this last article has gotten you enthused for GraphQL. To all my avid readers - thank you for your time and interest over these years! As always, if you want to send me a message you can reach me at lswest34+fcm@gmail.com. Especially if you happen to have a good list of my articles and what issues they appeared in!

Lucas has learned all he knows from repeatedly breaking his system, then having no other option but to discover how to fix it. You can email Lucas at: lswest34@gmail.com.

Python In The REAL World Pt.100

Written by Greg Walters

Lately, I've been doing some work with a computer book publisher on Machine Learning and Python. It's a very interesting subject, and I really enjoy the learning process on all of the various modeling methods.

You want pickle with that?

One thing that I've found is that when you need to (or want to) save some of the data in the middle of a process, often the pickle library is used. I've known about pickle for a long time, but have really never messed with it much, so I thought I'd explore some.

Pickle or olive?

An olive is a wonderful thing in a martini. It doesn't do anything for Python code. It is, however, a type of non-venomous Python mainly in Australia.

Pickle on the other hand, is a method to serialize and deserialize Python object structures. If you know all there is to know about Pickles, feel free to jump forward in the article. If not, or if you trust me to teach you something, keep reading.

Serializing means to take an object from memory, convert it into a stream of bytes that can be stored on disk. De-serializing is the reverse of the process. Let's say you have a dictionary. You can't just dump it to disk from memory. You have to convert it into a format that lends itself to being a disk file. JSON, XML, HTML all jump to mind. Pickles are just another way to do this. There's a library that handles all the tough stuff for you.

What can you pickle? Well most Python objects can be pickled, but there are a few that can't. While you can pickle simple objects (Integers, floats, complex numbers and strings), you normally would pickle Tuples, Lists, sets and Dictionaries that are built from most objects. However, things like generators, lambda functions and defaultdicts can not be pickled. There are some workarounds, but this is pretty much the rule of thumb.

Pickling Process

I want to thank my friend Halvard Tislavoll from Norway for the idea and the code for this part.

Let's say that you want to create a dataset of colors that can be used when creating a GUI. Tkinter (as well as other GUI toolkits) allows you to use hex codes as well as color names. There are many web pages that show all the colors along with the color name for quick reference, but what if you needed all of them along with their hex values?

Under Ubuntu, there is a file located in the /etc/X11 folder called rgb.txt. Make a copy of it and put the copy into a working folder.

Please note that Halvard's coding style is not the same as mine and I'm sure that it's not the same as yours. However, I'm sure that you will be able to understand his code.

Now we can get started creating a program to convert this text to a dictionary and pickle it. Name your program file "rgb2pickle.py". First, you need to import the pickle library

```
import pickle
```

Now, let's define an empty list and empty dictionary...

```
myList  = []
myDict  = {}
```

Next, we'll want to create some support functions. First the function that will read the rgb.txt file and return the contents in a list (see next page, top right).

This function takes the decimal value and returns the hex value using the above helper functions (next page, bottom right).

This function takes the list and puts everything into a dictionary...

```
def txt2dict(myList):
    i = 0
    aDict = {}
    for item in myList:
        newList = []
# pick name, clean it for tab
        name =
item[13:].lstrip("\t")
        name = name.rstrip()
# pick RGB values (dec)
        R =
item[0:3].lstrip()
        R = emptystring(R)
        G =
item[4:7].lstrip()
        G = emptystring(G)
        B =
item[8:11].lstrip()
        B = emptystring(B)
```

Now, convert the RGB values to a hex value…

```
# convert dec2hex value
H1 = dec2hex(R)
H2 = dec2hex(G)
H3 = dec2hex(B)
# build a hash value
H = f"#{H1}{H2}{H3}"
```

And append this to the list structure and update the dictionary…

```
# make a list with a new
column structure
        newList.append(name)
        newList.append(R)
        newList.append(G)
        newList.append(B)
        newList.append(H)
        aDict[i] = newList
        i += 1
    return aDict
```

```
def read_file_textlines(filname):
    """read text from file and return the file content in a list of lines"""
    import io
    signal = None
    try:
        with open(filname, "r") as f:
            text_lines = f.read().splitlines()
            signal = True   # Done.
            return signal, text_lines
    except IOError:
        signal = False   # Something went wrong!!
        return signal, ''
```

Next is a function that checks to see if a section of text is empty and if so, replace it with '00'...

```
def emptystring(txt):
    if txt == '':   # txt == '0'
        txt = '00'
    return txt
```

Here is a routine that makes sure that the text-only values are properly set up as hex...

```
def onelychrs(txt):
    if len(txt) < 2:
        txt = f"0{txt}"
    return txt
```

This is another function that returns a valid "00" hex value…

```
def chkzero(txt):
    if txt == '0':
        txt = '00'
    return txt
```

Now that all of the helper functions are done, let's put them all together (next page, top right).

The terminal output is very uninteresting and responds very quickly. Now, we have a pickle file, what do we do with it?

```
def dec2hex(txt):
    txt = hex(int(txt)).lstrip("0x")
    txt = txt.upper()
    txt = onelychrs(txt)
    txt = chkzero(txt)
    return txt
```

DEPICKLING A PICKLE

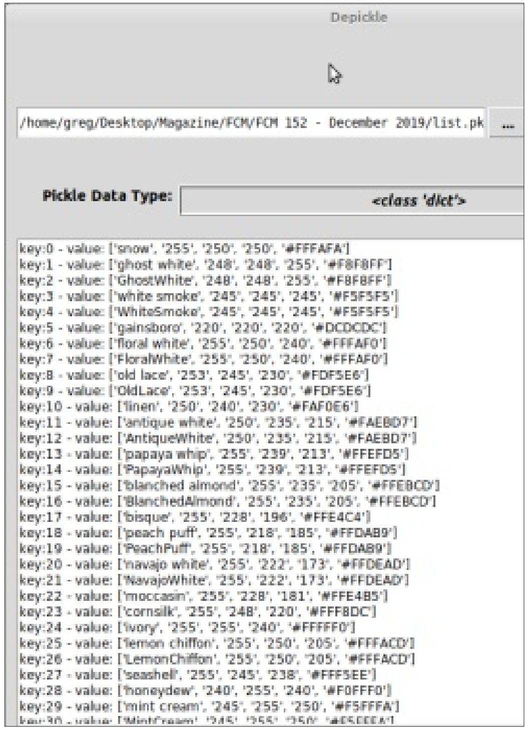
Depickling (or deserializing) is just as easy as it was to create the pickle file once you have the data ready.

Here is a quick code snippet that you can use in a CLI application…

```
with open(filename, 'rb') as
f:
    data = pkl.load(f)
```

This works well only if the pickle file was created with Python 3. If, however, it was created with Python 2 and the cPickle routine, it will probably error out. An easy workaround for this is (shown bottom right).

```
with open(filename, 'rb') as f:
    try:
        data = pkl.load(f)
    except UnicodeDecodeError:
        f.seek(0)  # in case the offset of the file is not in the beginning
        data = pkl.load(f, encoding='bytes')
```



/home/greg/Desktop/Magazine/FCM/FCM 152 - December 2019/list.pk

<class 'dict'>

```
key:0 - value: ['snow', '255', '250', '250', '#FFFAFA']
key:1 - value: ['ghost white', '248', '248', '255', '#F8F8FF']
key:2 - value: ['GhostWhite', '248', '248', '255', '#F8F8FF']
key:3 - value: ['white smoke', '245', '245', '245', '#F5F5F5']
key:4 - value: ['WhiteSmoke', '245', '245', '245', '#F5F5F5']
key:5 - value: ['gainsboro', '220', '220', '220', '#DCDCDC']
key:6 - value: ['floral white', '255', '250', '240', '#FFFAF0']
key:7 - value: ['FloralWhite', '255', '250', '240', '#FFFAF0']
key:8 - value: ['old lace', '253', '245', '230', '#FDF5E6']
key:9 - value: ['OldLace', '253', '245', '230', '#FDF5E6']
key:10 - value: ['linen', '250', '240', '230', '#FAF0E6']
key:11 - value: ['antique white', '250', '235', '215', '#FAEBD7']
key:12 - value: ['AntiqueWhite', '250', '235', '215', '#FAEBD7']
key:13 - value: ['papaya whip', '255', '239', '213', '#FFEFD5']
key:14 - value: ['PapayaWhip', '255', '239', '213', '#FFEFD5']
key:15 - value: ['blanched almond', '255', '235', '205', '#FFEBCD']
key:16 - value: ['BlanchedAlmond', '255', '235', '205', '#FFEBCD']
key:17 - value: ['bisque', '255', '228', '196', '#FFE4C4']
key:18 - value: ['peach puff', '255', '218', '185', '#FFDAB9']
key:19 - value: ['PeachPuff', '255', '218', '185', '#FFDAB9']
key:20 - value: ['navajo white', '255', '222', '173', '#FFDEAD']
key:21 - value: ['NavajoWhite', '255', '222', '173', '#FFDEAD']
key:22 - value: ['moccasin', '255', '228', '181', '#FFE4B5']
key:23 - value: ['cornsilk', '255', '248', '220', '#FFF8DC']
key:24 - value: ['ivory', '255', '255', '240', '#FFFFF0']
key:25 - value: ['lemon chiffon', '255', '250', '205', '#FFFACD']
key:26 - value: ['LemonChiffon', '255', '250', '205', '#FFFACD']
key:27 - value: ['seashell', '255', '245', '238', '#FFF5EE']
key:28 - value: ['honeydew', '240', '255', '240', '#F0FFF0']
key:29 - value: ['mint cream', '245', '255', '250', '#F5FFFA']
key:30 - value: ['MintCream', '245', '255', '250', '#F5FFFA']
```

```
# Main code starts here...
signal, myList = read_file_textlines('rgb.txt')
print(f'Signal is: {signal}')

This is the part that creates the pickle file...
myDict = txt2dict(myList)
fn = 'list.pkl'
outfile = open(fn, 'wb')
pickle.dump(myDict, outfile)
outfile.close()

print('Finished!')
```

Anyway, for my purposes I wanted to be able to actually see the data in its raw form, direct from the pickle file. I threw together a quick Page form and threw in a very little code (THANK YOU Page!) and here is the result…

I won't bore you with the details of how to do this in Page, since there's only three buttons, an entry widget, two labels, and a scrolled text widget. We've already covered that. However, I will show you the code that is important in the _support file. I really didn't worry about any error checking for the simple project.

We'll look at the callback for the "get filename" button (the one that has "…" as its text) first (middle right). Basically, this simply calls a tkinter askopenfilename filedialog and puts the selected filename and path into the entry widget for display.

```
def on_btnFile():
    # print('depickle1_support.on_btnFile')
    # sys.stdout.flush()
    global fn, progpath
    title = "Select Pickle File"
    ft = (("Pickle files", "*.pkl"), ("all files", "*.*"))
    fn = filedialog.askopenfilename(initialdir=progpath,
                                    title=title,
                                    filetypes=ft)

    entry_var.set(fn)
```

Next is the callback for the "GO" button. This is where the real work is done. The logic is to:

- Clear the text box.
- Open the file that the user has selected in the routine above.
- Depickle it.
- Determine the type of data structure and display it.
- Fill the text widget with the data from the structure (if possible).

```
def on_btnGo():
    #
print('depickle1_support.on_b
tnGo')
    # sys.stdout.flush()
    clear_stw()   # Clear the
text widget
```

Here is the actual code for depickling (next page, top right).

Lastly, based on the type of data we have, fill the text widget (bottom right).

The function to clear the text widget is really simple…

```
def clear_stw():
    # Clear the ScrolledText
Widget

w.Scrolledtext1.delete('1.0',
'end')
```

And the very last thing, just to be complete, is the Page provided function that defines the tkinter variables that allow easy setting of the text for the various widgets.

```
def set_Tk_var():
    global pickleType
    pickleType =
tk.StringVar()
    pickleType.set('')
    global entry_var
    entry_var =
tk.StringVar()
```

That's it.

Here is a good website that can help you understand the pickling process if you still want to learn more... https://www.datacamp.com/community/tutorials/pickle-python-tutorial

```
    global fn
    with open(fn, 'rb') as f:
        try:
            data = pkl.load(f)
        except UnicodeDecodeError:
            f.seek(0)  # in case the offset of the file is not in the beginning
            data = pkl.load(f, encoding='bytes')    # encoding='latin1')
    print(type(data))
    pickleType.set(type(data))
```

```
    if type(data) is dict:
        for key, val in data.items():
            # print(f'key: {key} - value: {val}')
            w.Scrolledtext1.insert('end', f'key:{key} - value: {val}\n')
    elif type(data) is list:
        for d in data:
            w.Scrolledtext1.insert('end', f'{d}\n')
    else:
        w.Scrolledtext1.insert('end', f'{data}\n')
```

I've put the source files for the rgb2pickle.py file from Halvard on pastebin at: https://pastebin.com/s09mp72G

And the python source for the Depickle GUI program there as well at: Depickle.py - https://pastebin.com/wsUMqk1F; Depickle_support.py - https://pastebin.com/TgmgngxJ

Until next time, I hope you have a wonderful New Year and remember to keep coding!

Greg Walters is a retired programmer living in Central Texas, USA. He has been a programmer since 1972 and in his spare time, he is an author, amateur photographer, luthier, fair musician and a pretty darn good cook. He still is the owner of RainyDaySolutions a consulting company and he spends most of his time writing articles for FCM and tutorials. His website is www.thedesignatedgeek.xyz .

Written by Wil Lake

CAD To Scale

There are a number of good books & articles on how to use Computer Aided Design software, including Alan Ward's FreeCAD published in Full Circle, but I have found it difficult to find out how to set up CAD software to print out on a fixed size of paper to a specific scale. The following is how I have been able to achieve this.

A BIT OF CAD BACKGROUND

A number of sources maintain that Computer Aided Design dates back to the late 1950's, when Dr. Patrick J. Hanratty developed a numerical control program that was called PRONTO. In 1960, Ivan Sutherland, who worked in the Lincoln Laboratory of MIT in the USA, created SKETCHPAD, which was more like the graphical design systems we use today.

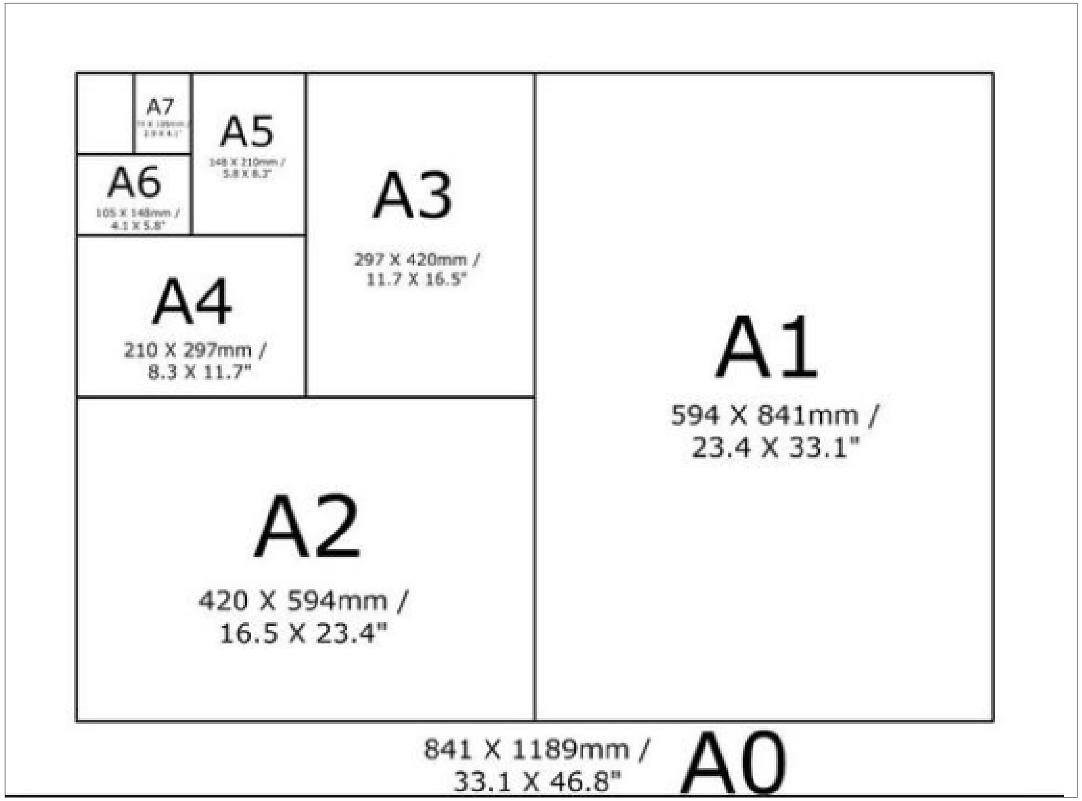
In the past, designers used pen / pencil, and drew on linen / paper, to produce a design that they wished to be made or stored for future use. This original would be reproduced using a contact print process onto light-sensitive paper. The process was introduced in the 19th century and allowed accurate reproduction of the original design document. The process produced a negative of the original with light coloured lines on a blue background. Thus it became known as the blue-print to be given to the people who would make the required design.

Some of the past designers would produce their work on paper, full size. In the car industry, they had large drawing boards as well as large tables and wall areas to draw their designs to full size.

To this day, we also draw designs full size, but use computers to produce files that we can edit, save, publish, and print. But, we do not usually reproduce the print full-size. Sometimes we want to reproduce the print of a large physical design on a smaller scale or, if the physical design is small, we may want to reproduce the print on a bigger scale. So that we all understand what is being reproduced on the print, we have standards that we use for both paper sizes and representative scales.

These standards are organized by 'International Organization for Standardization' (ISO). Because 'International Organization for Standardization' would have different acronyms in different languages (IOS in English, OIN in French for Organisation internationale de normalisation), the founders decided to give it the short form ISO. ISO is derived from the Greek isos, meaning equal. Whatever the country, whatever the language, ISO is always used.

There are a number of ISO paper sizes from A0 ('A' Zero) to A7. Most home printers take the ISO paper size of A4, which is 297mm high by 210 mm wide. The ISO 'A' paper sizes; one could say



that it starts with A0 which is 1189mm by 841mm. To get to the next size down (A1), one folds the A0 paper in half which becomes 841 by 594. Again to get to the next size down (A2), one folds the A1 paper again in half which becomes 594 by 420. S

To enable designs to be reproduced on paper, the full-size design has to be scaled to fit. There are a number of standard scales that are used which include 1 to 50, 1 to 100, 1 to 200, etc, and they are usually written as 1:50, or 1:100, or 1:200. The unit of 1 on the printed drawing would represent 50 or 100 or 200 units of the design.

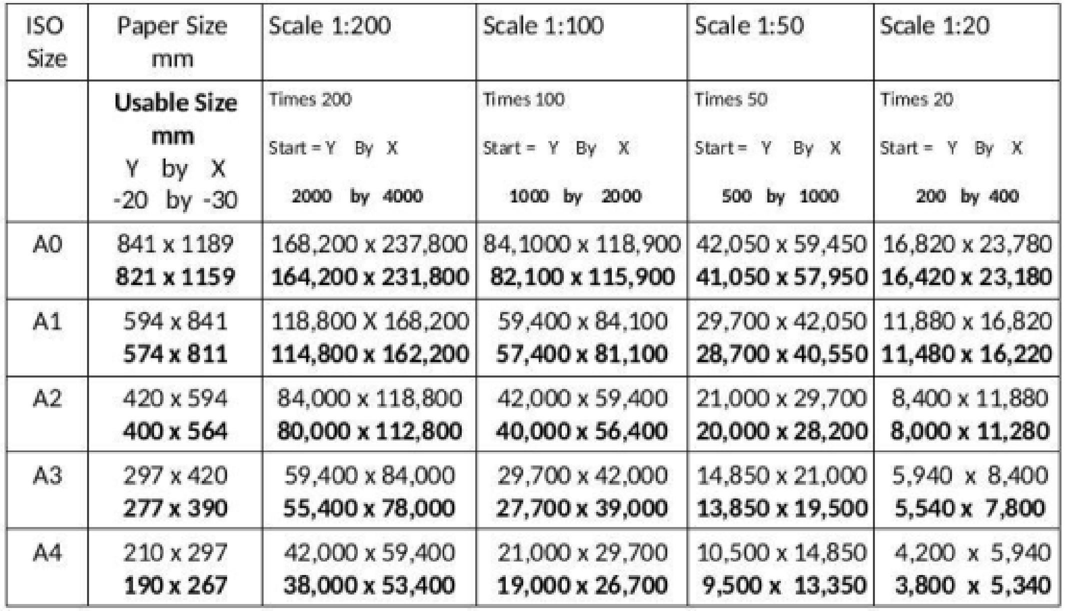
The scales can also be shown as 1:50 = 0.02, or 1:100 = 0.01, or 1:200 = 0.005.

Deciding on paper size and scale

After deciding what you are going to draw, you need to decide on how and to whom you are issuing the drawings. This could be a carpenter or builder to make something for you, or it could be a gardener or landscaper to lay out something for you.

First decide how big, in full size, your design is going to be. Next decide on which size of paper you want to issue your drawing design on.

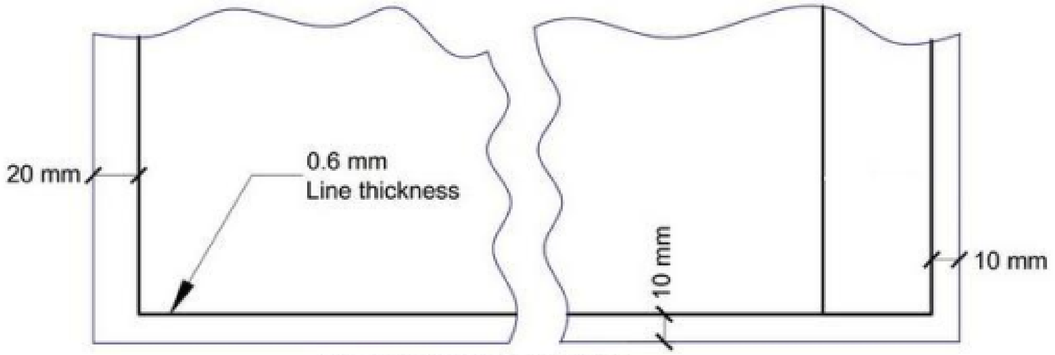
All drawings require a border. The standard line thickness of the border is 0.6mm. The top, right & bottom margins should be 10mm and the left margin should be 20mm. This applies for all sheets sizes A0, A1, A2, A3, A4. Notice that the left margin is larger to allow space for binding a drawing set.



ISO Size	Paper Size mm	Scale 1:200	Scale 1:100	Scale 1:50	Scale 1:20
	Usable Size mm Y by X -20 by -30	Times 200 Start = Y By X 2000 by 4000	Times 100 Start = Y By X 1000 by 2000	Times 50 Start = Y By X 500 by 1000	Times 20 Start = Y By X 200 by 400
A0	841 x 1189 **821 x 1159**	168,200 x 237,800 **164,200 x 231,800**	84,1000 x 118,900 **82,100 x 115,900**	42,050 x 59,450 **41,050 x 57,950**	16,820 x 23,780 **16,420 x 23,180**
A1	594 x 841 **574 x 811**	118,800 X 168,200 **114,800 x 162,200**	59,400 x 84,100 **57,400 x 81,100**	29,700 x 42,050 **28,700 x 40,550**	11,880 x 16,820 **11,480 x 16,220**
A2	420 x 594 **400 x 564**	84,000 x 118,800 **80,000 x 112,800**	42,000 x 59,400 **40,000 x 56,400**	21,000 x 29,700 **20,000 x 28,200**	8,400 x 11,880 **8,000 x 11,280**
A3	297 x 420 **277 x 390**	59,400 x 84,000 **55,400 x 78,000**	29,700 x 42,000 **27,700 x 39,000**	14,850 x 21,000 **13,850 x 19,500**	5,940 x 8,400 **5,540 x 7,800**
A4	210 x 297 **190 x 267**	42,000 x 59,400 **38,000 x 53,400**	21,000 x 29,700 **19,000 x 26,700**	10,500 x 14,850 **9,500 x 13,350**	4,200 x 5,940 **3,800 x 5,340**

The table lists the ISO sizes including the actual paper measurements and the corresponding full size drawing measurements under a number of different scale options, note all units are in millimetres. The tables values are for non-uniform borders, i.e. 20mm to the left and 10mm on the other edges.

Let us do a dry run. Say that you want a gardener / landscaper to alter your back garden. First measure your garden width & length. At this stage, all one needs to do is pace out the measurements, i.e. one step to be approximately one metre. If you record nine steps wide and fourteen steps long, using your A4 home printer we may be able to use a scale of 1:50.



Drawing borders and margins

The 1:50 scale on an A4 paper has a usable area equal to a full size measurement of 10 metres by 14.35 metres. Depending on what changes you want to show on your drawing, there may be enough space. If there is a lot of detail, and some written explanations needed on the drawing, the A4 paper can still be used but use a scale of 1:100 which has a usable area

contents ^

equal to a full size measurement of 20 metres by 28.7 metres. An alternative would be to draw on a paper size of A3 using a scale of 1:50, then save the drawing full size as a pdf file and get a printing company to print it out full size for you. Another alternative would be to print out the saved pdf file on your home printer on A4 paper, but the drawing would not be to scale.

Let us try another dry run. You want to draw up a building (it could be a shed / outbuilding) that is 3 metres wide by 5 metres long and has a top roof height of 4 meters. Usually when designing buildings, there are a number of views recorded on the one drawing. These include a front view elevation, up to two side view elevations, a back view elevation, and a plan view.

With the plan view in the middle, with a front elevation below, a back elevation above, and a side elevation both left & right of the plan, we can work out the maximum real size needed. The plan would be a 3 x 5 metre oblong, the front elevation would be a 4 x 5 metre oblong as would the back elevation, the two side elevations would be 3 x 5 metre oblongs. Thus the overall size would be 13 metres wide by 11 metres high, but only if the elevations butted up to each other. To enable the drawing to be viewed easily we would have at least 2 metres, or more, between each elevation, thus the overall size becomes 17 metres wide by 15 metres high. But to have enough space to add comments, lists of parts, title boxes, and other information, the width and height would need to be increased. The best size of paper would be an A3 and the scale would be 1:100.

LibreCAD

LibreCAD is Free and Open Source Software, available for everyone to use, share and modify, and produced by a worldwide community of hundreds of developers. LibreCAD can be used by the three platforms of Windows, Apple Mac, and Linux, i.e. Ubuntu.

Using LibreCAD version 2.0.9 running under Ubuntu 16.04, the following actions were used to set up a drawing to print on A3 paper to a scale of 1:100. The drawing can also be saved as a PDF file and taken to a printing company for them to print it out full size.

Open LibreCAD and set the paper size to A3 by clicking on the headings: Edit, Current Drawing Preferences (to open a sub window), Drawing Preferences, and under the Paper tab, change the Paper Format to A3 Landscape, and click OK.

Add two new Layers by clicking on the heading: Layer, Add Layer, or by clicking on the add [+] sign under the Layer List heading to open a sub window named Layer Settings. Name the first new layer 95-BK-Border, and have the colour Black/White, the Width 0.6mm (ISO) and Line type Continues. For the second new layer, use the name 97-Y-Paper, and have the colour Yellow, the Width 0.25mm (ISO), and Line type Dash.

With the 97-Y-Paper layer highlighted, click on the Command input line, which should show that it is active by the Command line title changing to blue.

On the Command line enter:-

`rec <enter>`
{shortcut for the rectangle command}
`0,0 <enter>`
{specifies first corner}

`42000,29700 <enter>`
{specifies second corner}

`za <enter>`
{shortcut for Zoom Auto command}

[highlight the 95-BK-Border Layer and enter at the Command line]

`2000,1000 <enter>`
{specifies first corner of the Border line}

`39000,27700 <enter>`
{specifies second corner of Border line}

Let's look at what the drawing would look like.

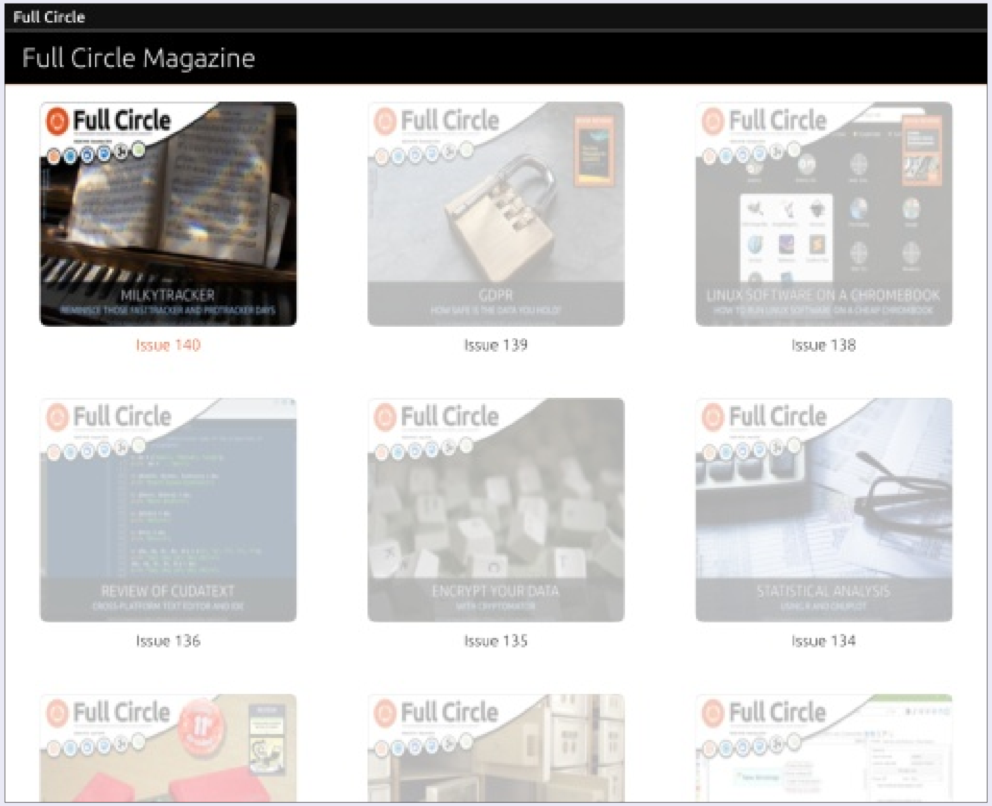
Click on File, Print Preview to view print preview, then move the cursor over the preview area which displays a numbered box and three icons. Click on the numbered box and change the number to '1:100' and press <enter>. Then click on the centre icon to 'centre to page'. The preview screen will change showing the Yellow Page Line & Border Lines are in sync with the A3 paper. Unfortunately, if we do not edit the Yellow Page line settings, the print process will try to print the Yellow Dashes on the edge of the paper. To overcome this problem, with the Layer 97-Y-Paper highlighted, click on the printer icon next to its name to switch off the print option.

To save the drawing, click on File, Save as – to open a sub window Save Drawing As. Edit the ‘Look In’ box to show the path to your required storage folder, i.e. /home/username/Documents/CAD-files/ Name the drawing in the File Name box to A3-1-100-Empty, and click on Open to save the file.

This drawing file can then be used as your source file to be copied, renamed, and used to draw your shed / outbuilding in full size.

Wil Lake is a retired time-served mechanical engineer who retrained as an IT engineer when Microprocessors entered the mainstream. He has worked in the IT industry for over 30 years and lives in the Cotswolds, England, UK, and has spent years renovating a cowshed.

The Official Full Circle App for Ubuntu Touch - UPDATED!

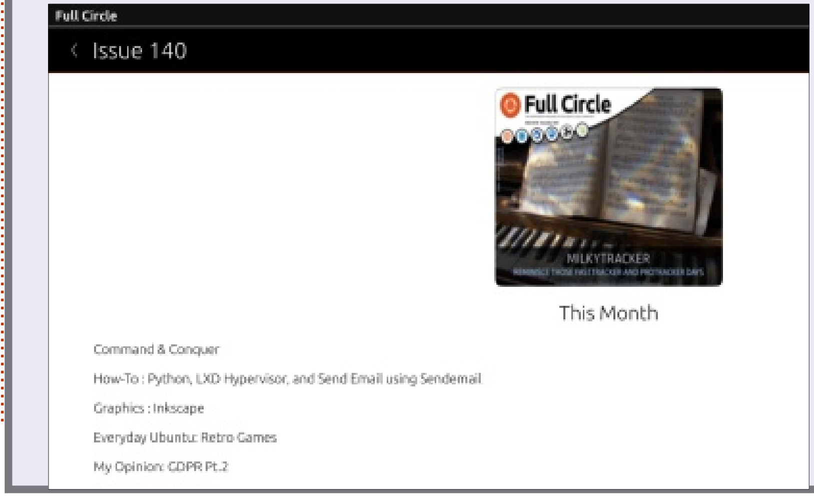


Brian Douglass has updated his FCM app for Ubports Touch devices that will allow you to view current issues, and back issues, and to download and view them on your Ubuntu Touch phone/tablet.

Install

Either search for 'full circle' in the Open Store and click install, or view the URL below on your device and click install to be taken to the store page: https://uappexplorer.com/app/fullcircle.bhdouglass

HUGE thanks to Brian for this.



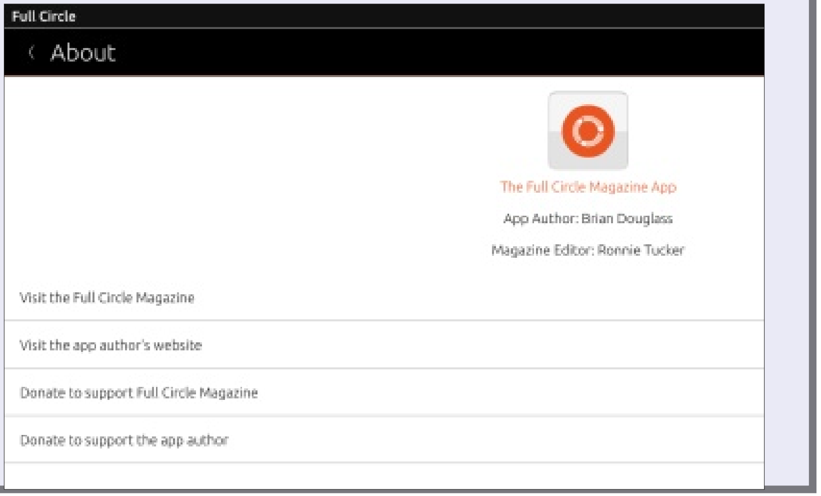
Command & Conquer

How-To : Python, LXD Hypervisor, and Send Email using Sendemail

Graphics : Inkscape

Everyday Ubuntu: Retro Games

My Opinion: GDPR Pt.2


Full Circle
About
Visit the Full Circle Magazine
Visit the app author's website
Donate to support Full Circle Magazine
Donate to support the app author

The Full Circle Magazine App

App Author: Brian Douglass

Magazine Editor: Ronnie Tucker

HOW-TO

Written by Erik

Darktable - Pt10

If you have been following our Darktable tutorials, you will know we are working with the latest version, not the version that is bundled in the Ubuntu Software centre.

Disclaimer: In no way am I an expert in photo manipulation, I just know a guy who had a dog that gave me some fleas. I thought I may help you scratch an itch also.

As promised, this issue we look into the wonderful world of LUTs.

This is what we currently have (pre Darktable 2.6) : https://www.darktable.org/2016/05/color-manipulation-with-the-colour-checker-lut-module/

But, in the newer versions (Darktable 2.7+), we have LUT 3D. But what is it you may ask? Well LUT is short for Look-Up Table. Why do we need it? LUTs allow us to change color spaces. That said, technically, you have more than one LUT type. We have the Technical / Scientific LUT for precise color shifts, and we have Creative / Artistic LUTs that you see in something like Snapseed on your phone. Now you will read that LUTs are mathematically precise, but that holds true only for the first type. Think of the two as serious and fun; you just have to decide where your work space is. It's also not a 'one size fits all' scenario, that is why there are so many. Because it is a mathematical formula, it gets applied to your exposure 'x' so the result where 'x' changes, it changes the outcome. There are a lot of variables, so you will need to put some effort in.

So what is a 3D Lut? Well think of it as an array in 3 dimensions. Read more here: https://www.darktable.org/2019/05/New%20module-lut3d/

Again, I am no expert, but people are visual creatures and sometimes we need to picture something to understand it. Not all of us need to be experts in everything, and that is OK, you just need to know enough to enjoy something.

To understand LUTs, let's look at their origins. LUTs started because of the need to match the output of different cameras. Though it is still used for this, it has become so much more.

Think of LUTs as a quick way to apply a color preset, like Snapseed on your phone does. That said, it is not a complete automation, as you still need to denoise or sharpen manually. LUTs apply to colors. There are basically two types us n00bs need to be aware of – the type they use in film production (CUBE FILES), and the type we use in digital photography (HALDCLUT FILES). So be sure you download the correct one, though you *can use the cube files, the latter is preferred. Should you see 3DL files or MGA-files; these are LUTs too, but just be aware that not all types of LUT files are supported by every photo manipulation application. If you have used RawTherapee or Photoflow, you will know which you need. All well and good, but where can you get LUTs? Here is a starting point: https://freshluts.com/, or we can borrow from RawTherapee: http://rawtherapee.com/shared/HaldCLUT.zip

If you have metered internet, know that some of these preset packages can be quite large.

Once you have some LUTs, you need to set the path. In Darktable, open the settings, go to the core settings tab and scroll all the way down. Choose your LUT root folder as per the image:

Add the 3D LUT module to your module arsenal. With the new search bar it is as easy as typing '3D' if you do not see it. Click the little folder to open the path you set previously. If this does not work, go back and set the path again. Make sure the file types are set to *your LUT type at the bottom of the new window that opened. (Remember CUBE and HALDCLUT files?). You should then be presented with your presets. Double-clicking any of them will apply it to your picture.

The only issue here is that if you have lots of LUTs, you will need to click on that mini folder again to open your list, and double-click each one in succession. There is, unfortunately, no quick way to run through them that I know of. If there is, or if you know of a faster way, please let us know.

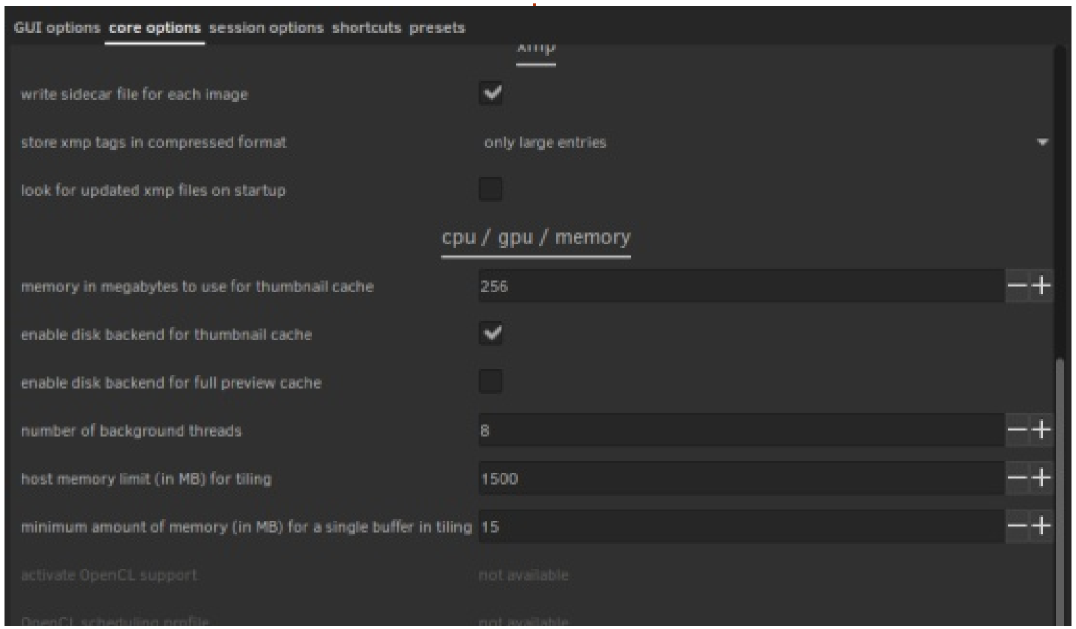
LUTs are applied at 100% opacity. You can change that. Feel free to play around with that as it can create some interesting effects. How do you do this, you may ask? Well, next to the big X is an O that you click. (No, we are not playing Tic Tac Toe). If you hover your mouse pointer over this, you will see a tooltip that says “uniformly”. Applying a LUT is not the be-all and end-all of your color correction. You can still go and adjust every aspect of the applied LUT with the other modules. Think of it as a cheat, to get you 90% of the way there (color correction). The differences between some LUTs are very subtle. Do not be afraid to take snapshots to compare them. There is no great skill required to use LUTs. Open one of your LUTs in your file manager and view it with mousepad or leafpad or gedit (whatever your choice of notepad application is), and you will see that a LUT is just text. Lots and lots of numbers.

I say this, as there are lots of free LUTs available on the internet, but you do not want a binary. Always make sure that the so-called LUT you just downloaded is actually a simple text file with numbers like these. Malware on Linux is a thing. Rather be safe than sorry. Though I have seen .EXE-files parading as LUTs, know that a LUT file is simply a lookup table that changes color values. Delete the binaries and find your LUTs elsewhere. Generally, LUTs will work on your Windows and MAC computers too, as they are text.

Now, about our two types of LUTs I mentioned in the beginning. The Technical LUT will make a drab picture look like it should, or bring life to it. The Artistic LUT is one you add to an already good looking picture to give it style or flair. Be aware of how you use each one.

Happy editing and should you have any questions or corrections, yes everyone makes mistakes, email us at: misc@fullcirclemagazine.org

Erik has been in IT for 30+ years. He has seen technology come and go. From repairing washing machine sized hard drives with multimeters and oscilloscopes, laying cable, to scaling 3G towers, he's done it.



write sidecar file for each image ☑

store xmp tags in compressed format: only large entries

look for updated xmp files on startup □

memory in megabytes to use for thumbnail cache: 256

enable disk backend for thumbnail cache ☑

enable disk backend for full preview cache □

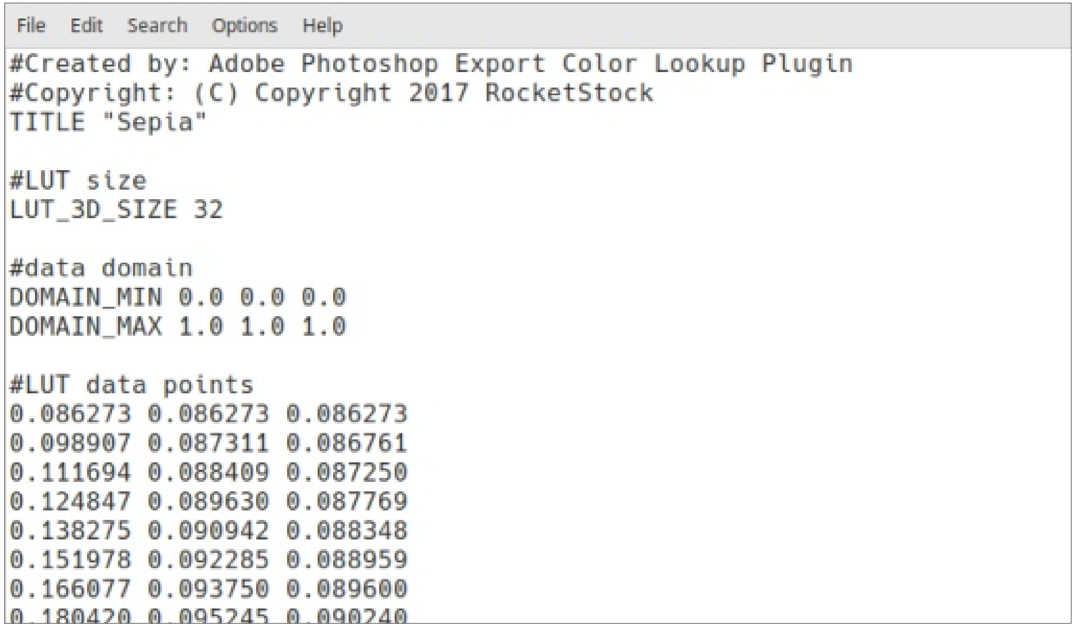
number of background threads: 8

host memory limit (in MB) for tiling: 1500

minimum amount of memory (in MB) for a single buffer in tiling: 15

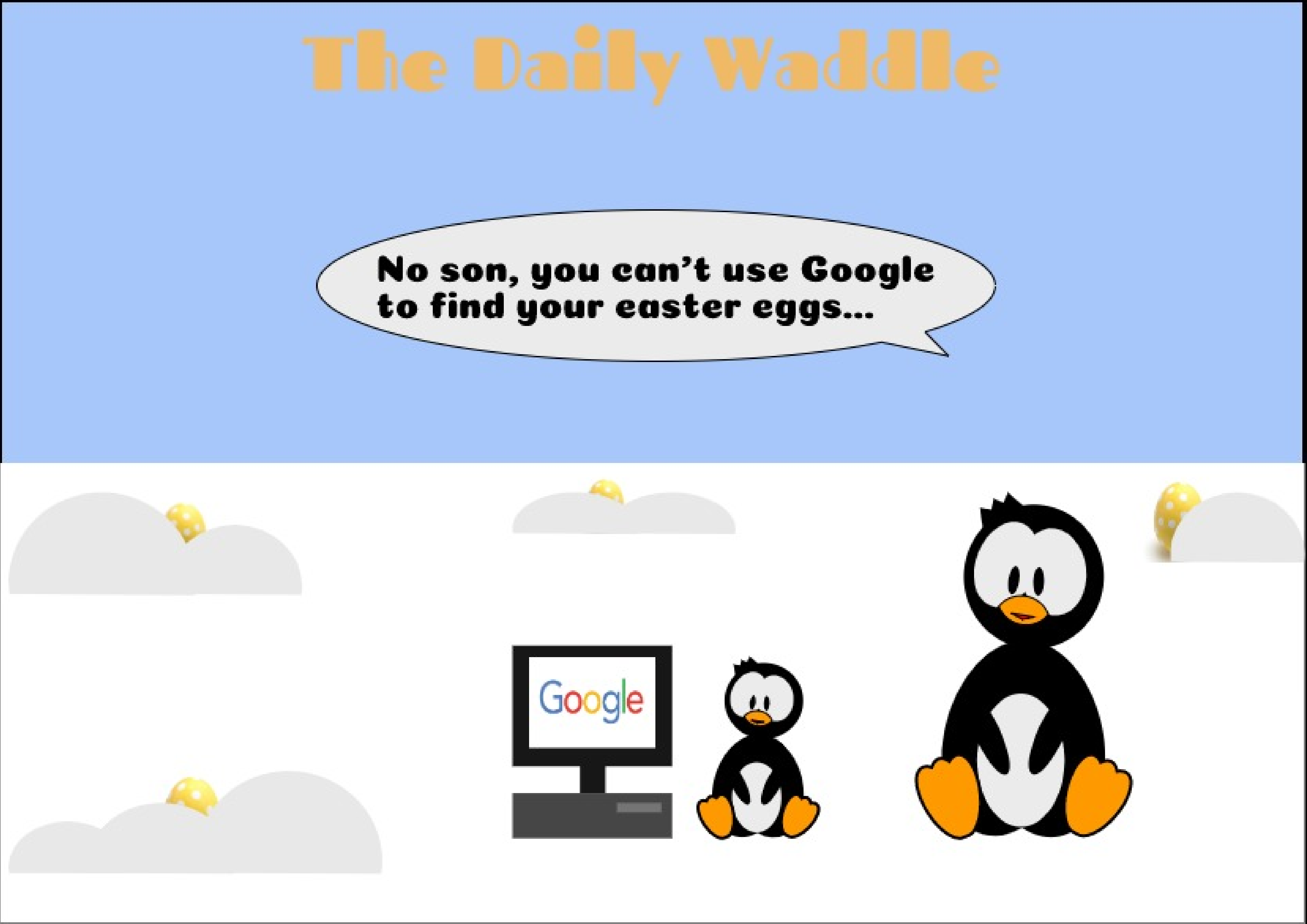
activate OpenCL support: not available

OpenCL scheduling profile: not available



```
#Created by: Adobe Photoshop Export Color Lookup Plugin
#Copyright: (C) Copyright 2017 RocketStock
TITLE "Sepia"

#LUT size
LUT_3D_SIZE 32

#data domain
DOMAIN_MIN 0.0 0.0 0.0
DOMAIN_MAX 1.0 1.0 1.0

#LUT data points
0.086273 0.086273 0.086273
0.098907 0.087311 0.086761
0.111694 0.088409 0.087250
0.124847 0.089630 0.087769
0.138275 0.090942 0.088348
0.151978 0.092285 0.088959
0.166077 0.093750 0.089600
0.180420 0.095245 0.090240
```

The Daily Waddle



HOW-TO

Written by Mark Crutch

Inkscape - Part 92

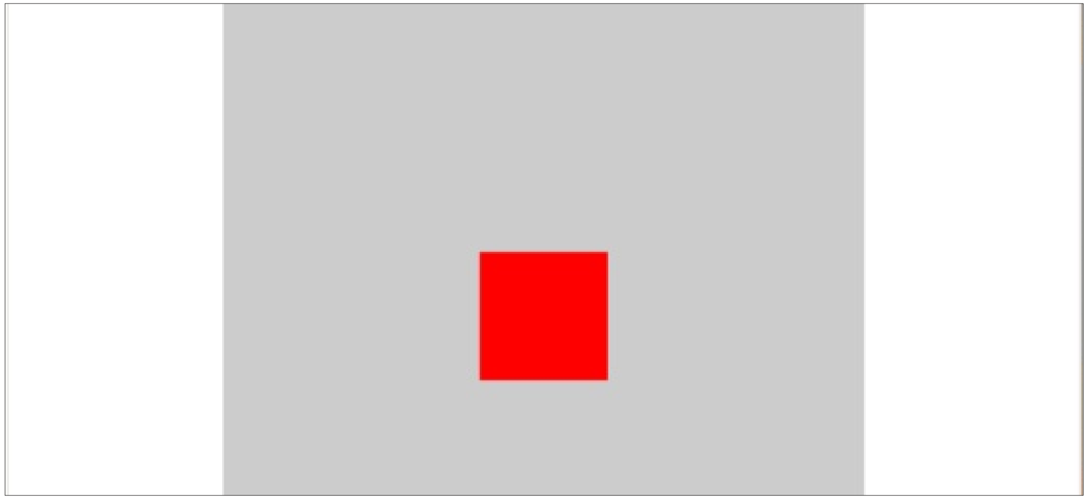
Last time, we looked at the "transform" attribute, particularly with regard to its frequent appearances in Inkscape files. In this instalment, and the next, we're going to combine this information with some of the JavaScript we've covered in previous months to show how you might animate the moving, turning, skewing and scaling of an SVG element.

Once again we'll start off with a very simple SVG file (top right).

As usual, we've got a viewbox of 100×100 units, but, this time, I've added a background <rect> to make it clearer where the extents of the canvas are when the file is loaded into a web browser. The element we're going to animate is the red rectangle, which is inside a group (<g>). The group has a transform attribute on it, containing a series of functions that will be applied to its content. In this case, they each have no effect – a rotation of 0 doesn't turn the shape at all, whilst a scale factor of 1 leaves it at its original size. Strictly speaking, they're not necessary at all at this point, but they are present to show you the format of the string we'll be creating with our animation code.

The rectangle itself bears a little explanation. In an SVG document, the y-axis runs down the page, and the x-axis runs from left to right. So the origin – the 0,0 point in the image – is at the top-left of the page. The position of a <rect> is also based on its top-left corner. Therefore, setting the "x" and "y" attributes to 0 would position the rectangle at the top left of the screen. But, for this code, I want to pretend that the "x" and "y" coordinates correspond to the center of the rectangle. This means offsetting its position by half the width to the left, and half the height upwards. Since it's 20 units on each side, the transform attribute with a translate(-10, -10) does the job nicely. Basically, we've moved the rectangle's reference point from the top-left corner to the center. With that manipulation in place, using 50 for the "x" and "y" coordinates puts it right in the

```
<svg
  xmlns="http://www.w3.org/2000/svg"
  viewBox="0 0 100 100">

  <script href="transform.js" />

  <rect x="0" y="0"
    width="100" height="100"
    fill="#ccc" />

  <g id="g1"
    transform="
      translate(0,0)
      scale(1)
      rotate(0, 50, 50)
      skewX(0)
      skewY(0)">

    <rect
     transform="translate(-10,-10)"
      x="50" y="50"
      width="20" height="20"
      fill="red" />
  </g>
</svg>
```



middle of the image.

There's one other line in the file – a <script> tag that references an external document. For the sake of simplicity, we're going to keep our code in a separate JavaScript file, avoiding problems with character escaping, and meaning that we don't need to make any further changes to the SVG file. The only requirement is that the JS file is kept in the same directory as the SVG file, since I've used only a filename, not a relative path or absolute URL. The JS file itself, “transform.js”, contains just a single line at this point:

```
alert("Hello World");
```

By loading the page in a web browser we can now easily confirm that we have a red square in the middle of a grey square, and that a message pops up from our JavaScript file to indicate that it's being loaded correctly. Now we can move on to some real code.

Based on the approach from part 90 of this series, we're going to create a single function that updates the transform element for each frame that the browser renders. The function will receive a timestamp, and use that to determine how long the animation has been running, and therefore what values should be put into the transform element's functions for that particular point in time.

To begin with, we'll just animate one of the properties: rotation. Replace the alert() call in the JavaScript file with this (right).

Much of this looks similar to code we've seen previously, but there are enough differences to warrant a step-by-step walkthrough.

We start by declaring a global variable called “group”. Previously, we've used the “var” keyword to do this, but modern JS has mostly replaced that with “let” (for variables that will change), and “const” (for those that won't). We've used “var” when working in the console as it won't throw an error if you try to run the same line twice – as “const” would do. But, as we're creating a separate JS file here, we'll stick to convention. In this case, the “group” variable will eventually hold a reference to the <g> element, but as that's not necessarily available as soon as the page loads, we'll declare it using “let” and update the value later.

```
let group;
requestAnimationFrame(initialise);

function initialise(ts) {
  group = document.querySelector("#g1");

  group.animProperties = {
    startTime: ts,
    rotationDuration: 5   // Time to turn 360 degrees
  }


  // Start the animation running
  requestAnimationFrame(animate);
}


function animate(ts) {
  const props = group.animProperties;


  const runningTime = (ts - props.startTime) / 1000;
  const anglePerSecond = 360 / props.rotationDuration;
  const totalAngle = anglePerSecond * runningTime;
  const angle = totalAngle % 360;


  group.setAttribute("transform", `
    rotate(${angle}, 50, 50)
  `);


  requestAnimationFrame(animate);
}
```

Later doesn't take long to arrive. The very next line of code causes our initialise() function to run just before the next frame is drawn, giving the browser time to render the content so that our <g> element actually exists in the document structure before we use it. The initialise() function itself does these things:

- Set our “group” global variable as a reference to the <g> element.
- Set up some JS properties on the element. First is the initial timestamp, which will be used as the basis for calculating the

timings in the animate() function.
- The only other property we're creating for now is the time it should take for the square to do one rotation, in seconds.
- Finally, we have another call to getAnimationFrame() which will start the actual animation running.

Now we get to the animation code itself, in the form of the animate() function. We begin by getting a reference to the animation properties we set up previously, and storing it with a more convenient name. We can use "const" instead of "let" here as the value we assign doesn't get changed within this function.

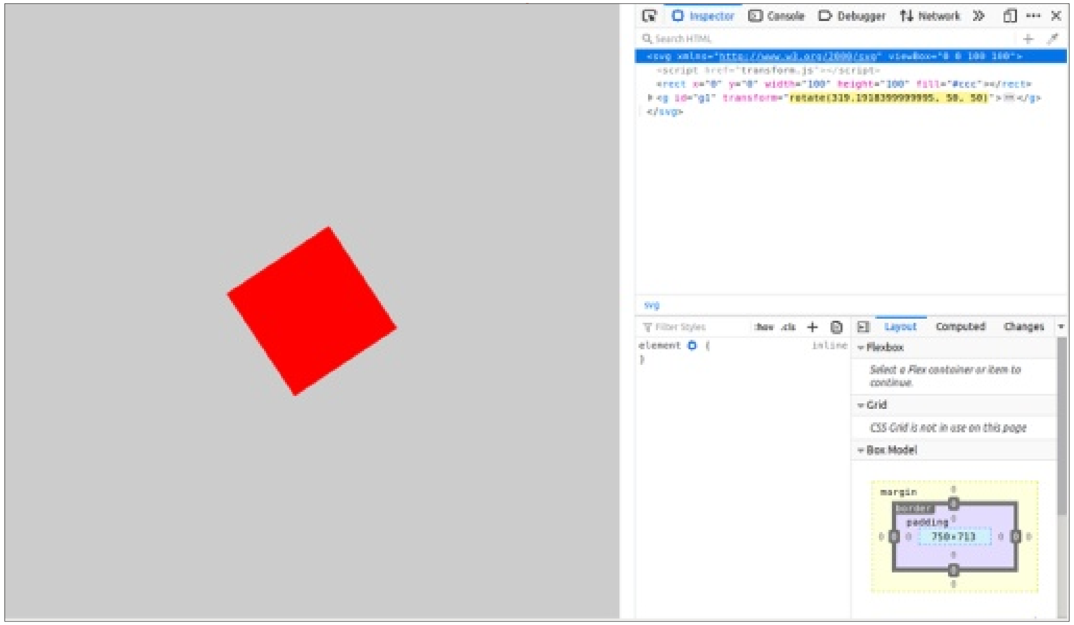
The next group of lines just calculates the value, in degrees, that we need to rotate the square by. We get the amount of time the animation has been running for, by subtracting the initial timestamp from the current one, then divide the value by 1000 to convert from milliseconds to seconds. By dividing 360 by the desired rotation time we find the amount of rotation we need to perform every second; multiplying that value by the amount of time we've been running for gives a total value for the number of degrees to rotate by.

After the first rotation has completed, the calculated value will be larger than 360. That's not actually a problem – the browser will happily do the right thing for you in this case – but I prefer to be a little explicit about what's happening. That explains the last line of this block, where we use the modulus operator (%) to get the value that remains after dividing the total angle by 360. This has the effect of normalising the rotation angle so it never goes above 360, which can make it easier to see what's happening if you need to log the value out, or if you view it live in the developer tools.

The penultimate line uses setAttribute() to update our transform attribute with a new value. The value itself is a template string, delimited by backticks (`…`). They're not always as easy to spot in code as the more usual quotes and double-quotes, so make sure you don't overlook them. Within a template string any content inside a ${} block will be evaluated as JavaScript, and the result will be placed into the string. For our purposes, this means we can use ${angle} to inject the value of the "angle" variable into the string, without having to perform a lot of concatenation. We're using the three-value form of rotate() in order to provide coordinates for the center of rotation (50, 50) – without it the square will rotate about the top-left corner of the page.

The last line simply queues up another call to the animate() function, as we've seen previously. Load the SVG file into a web browser and, if everything is correct, you should see the square spinning around in the middle of the page. Press F12 in the browser to open the developer tools, and select the tab labelled "Inspector" (Firefox) or "Elements" (Chrome/Chromium) – it's usually the leftmost tab on the bar. You should see the structure of your SVG file, with the <g> element visible, and the value of the transform attribute updating as the square rotates. Notice how the first number never goes above 360; try replacing the ${angle} tag with ${totalAngle} and see what difference it makes to the rotation and to the attribute's value.

Rotation is pretty straightforward because we have to deal with only an ever increasing number. If we exceed a full rotation then we either normalise the number, or let the browser do



```
<svg xmlns="http://www.w3.org/2000/svg" viewBox="0 0 100 100">
  <script href="transform.js"></script>
  <rect x="0" y="0" width="100" height="100" fill="#ccc"></rect>
▸ <g id="g1" transform="rotate(319.1918399999995, 50, 50)">…</g>
</svg>
```

```
element {
}
```

Select a Flex container or item to continue.

CSS Grid is not in use on this page

it for us. The other transform functions are a little more tricky: skewX and skewY expect a value between -90 and +90 (though the extreme ends of the range distort the object so much that they're not very useful); translate can take any number, but there's only a limited range that makes sense within the confines of our 100×100 viewbox; scale has a similar practical limit. For all these transform functions, therefore, we want to animate back and forth between two values. This means creating three properties for each thing we want to animate, for the lower limit, upper limit and duration. Here's how the group.animProperties object might be extended to also include skewX, for example (note the addition of a comma after the rotationDuration property, as this is no longer the last item in the object). Shown top right.

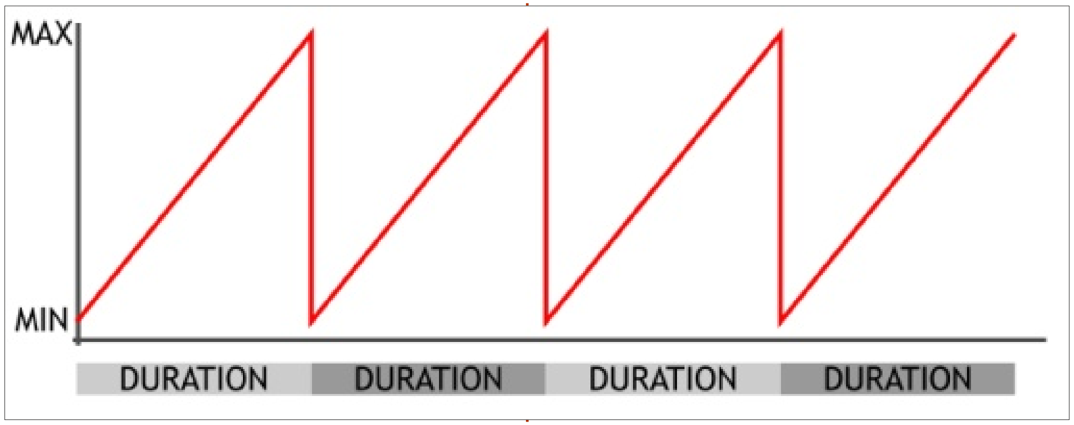
```
group.animProperties = {
  startTime: ts,
  rotationDuration: 5,    // Time to turn 360 degrees

  skewXDuration: 10,
  skewXMin: -60,
  skewXMax: 60
}
```

To go with the new property, we'll also need an extra group of lines in the animation function, just after the corresponding lines for rotation, but before the call to setAttribute() (bottom left).

```
const skewXDur = props.skewXDuration;
const skewXRange = props.skewXMax - props.skewXMin;
let skewXPosition = (runningTime % skewXDur) / skewXDur;
const skewXDirection = runningTime % ( skewXDur * 2);
if (skewXDirection > skewXDur) {
  skewXPosition = 1 - skewXPosition;
}
const skewXAmount = (skewXRange * skewXPosition) + props.skewXMin;
```

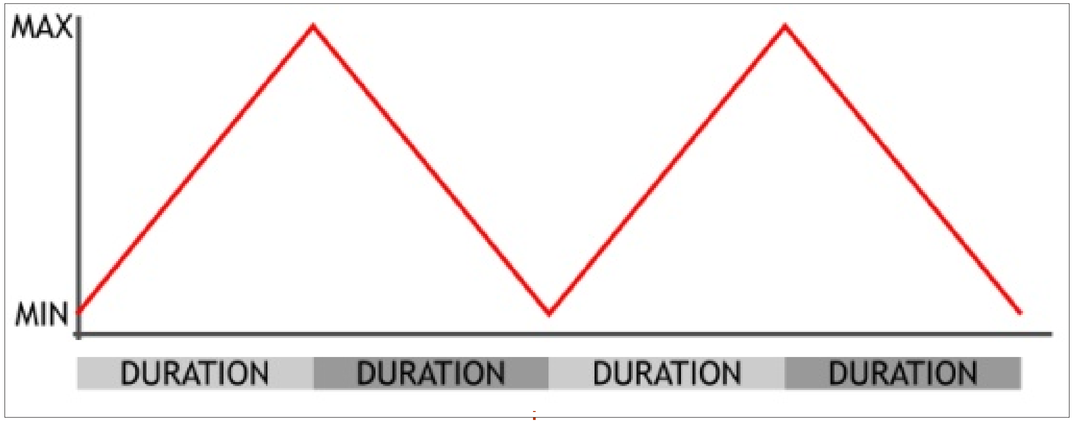
We start by assigning props.skewDuration to a local variable, for no other reason than it gets used a lot, so we've given it a more convenient name. The second line subtracts the minimum value property from the maximum, to give us the total amount of possible skew. We'll use this to work out what the current skew amount should be at any given timestamp.

The third line calculates the "position" along the animation for the current timestamp. We do this by taking the running time (calculated earlier, in the previous block), dividing it by the duration for this animation, then taking the remainder. This gives us a value that runs from zero to the duration value, then jumps back to zero before ramping up again on each iteration. Rather than running from zero to "duration" it's more useful if we adjust this value to be a decimal from 0 to 1, which is achieved by dividing by the total duration.



If we were to comment out the next few lines and jump to the last one, we would find that the animation cycles repeatedly from the minimum value to the maximum, jumping straight back to the minimum on each iteration. Plotting the values over time results in a "sawtooth" chart.

For our animation, however, we want the value to transition linearly both up and down, without the sudden jump between iterations. What we want is a triangle wave:



As you can see, on odd numbered iterations we want the animation to proceed as usual, but on the even numbered ones we want the position value to step downwards rather than upwards. In the code above this is done by creating a “skewXDirection” variable which holds the modulus of the current running time when divided by twice the duration. This value will ramp up from zero at the start of an odd iteration, through the duration value at the end of the odd iteration, continuing up to twice the duration value at the end of the subsequent even iteration. We’ve created another sawtooth wave, but this time running from zero to duration×2 over the course of two iterations.

The “if” statement that follows checks to see if this direction value is greater than the duration: if it is then we must be on an even cycle. In that case the “skewXPosition” variable (which, if you recall, ranges from 0 to 1) is subtracted from 1, so as the animation progresses the final position value first steps upwards, then steps downwards, before the cycle repeats in a triangle wave pattern.

The last step in calculating the actual value is to multiply the current position in the cycle by the total range of the animation, then add the minimum value to move the final result into the right range of numbers.

Phew! That was a lot to take in for a few lines of code. If you find it easier to follow, try adding some console.log() lines amongst the code so you can see how the values change in the developer tools.

With our final value calculated, the last step is to update the transform attribute to hold both the rotate() and the skewX() functions. Extend the previous template string to this:

```
group.setAttribute("transform", `
  rotate(${angle}, 50, 50)
  skewX(${skewXAmount})
`);
```

Loading the file into the browser, you should see the square being skewed as it rotates. But you’ll also notice that our simple, constrained rotation in the middle of the screen has turned into a whirling dervish that swoops out of the bounds of our image before flying back in and then setting off into the distance once more. Next time we’ll discuss why this is happening, and finish this little animation by adding scaling and translation. In the meantime why not test your own understanding of this code by adding the necessary lines to make the shape also skew in the Y direction, at a different frequency to the skewX() effect.

Mark uses Inkscape to create three webcomics, 'The Greys', 'Monsters, Inked' and 'Elvie', which can all be found at http://www.peppertop.com/

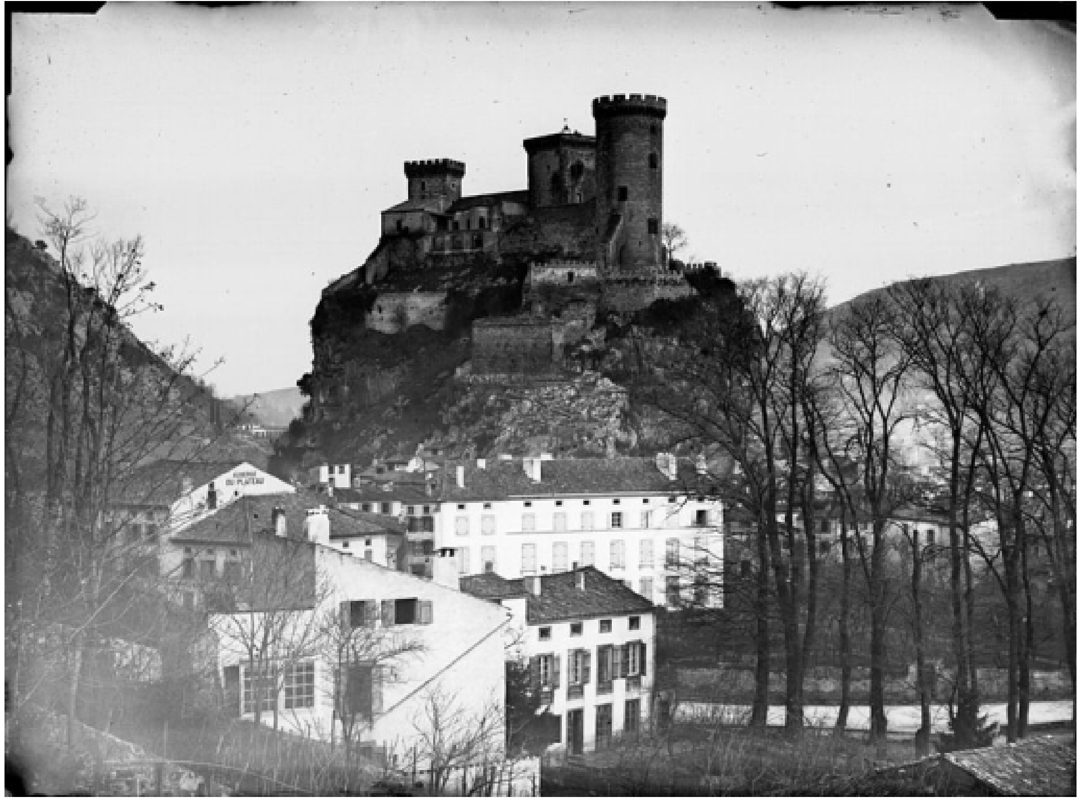
Written by Alan Ward

Krita To Rework Old Photos Pt.2

As discussed previously, this series is aimed at learning to make something of the old photos in my possession, and others in the public domain, due to their age. You, the reader, are welcome to tag along, and, I hope glean some small insight and perhaps an idea or two from time to time. No promises are made as to the quality of the content, or potential errors and omissions. I am a computer scientist, not a true artist or a professional on image restoration. So please take all this as a best effort, but with no firm guarantees - much as is the case of most open-source software.

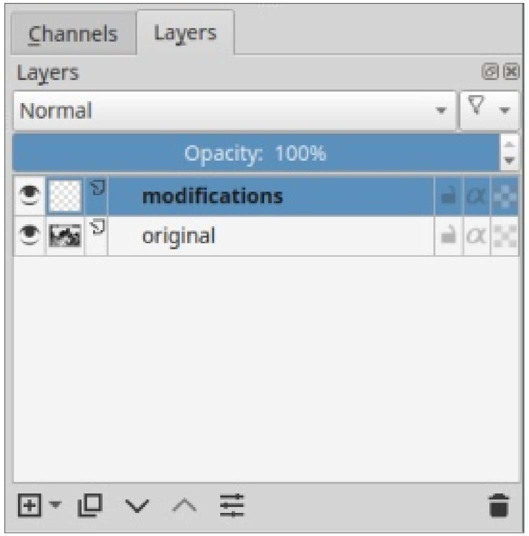
In this part of this series, we will commence work with a simple landscape, a photo of the castle of Foix in southern France about the turn of the century. With the passage of time, this photo is now stated to fall within the public domain, has already been digitized by the Rosalis project of the municipal public library of Toulouse, and may be downloaded from Wikicommons at address: https://commons.m.wikimedia.org/wiki/File:Ch%C3%A2teau_en_ruines_(8056081904).jpg .

A cursory visual inspection shows us that this picture has some of the rather typical imperfections often associated with photographs of its time: more than a century old, the original was no doubt a glass plate using the dry (gelatin) process. These include: physical loss of parts of the image around the edges due to wear-and-tear while handling or storing the image, a completely washed-out sky that shows no relief at all, an improbably darkened castle and parts of the surrounding vegetation as well as the trees to the right of the image, and a patch of light-colored discoloration in the left-bottom corner. Luckily, the image itself is sharp and well-focused. Details are easily visible, except where excess brightness has washed them out such as on light-colored window shutters. Some dark defects (spots) may be seen, mostly against the clear sky.



Once we have this file opened in Krita, the first thing I like to do is to rename and save it in Krita's internal format. We will be using layers, which are not supported by the picture's original JPEG format. By changing the format, we can ensure that layers are kept separate (and not fused together). We also make sure we are working on a new copy and not on the original file, which may be stored for further reference if needed — or if disaster strikes for whatever reason.

Returning to layers, I then set up a new layer. It will, initially, be transparent. All modifications that imply adding color -- painting, spraying -- will be done to this layer, and not to the original image. If any serious errors are made, they can be wiped off on the new layer, without affecting the original. Giving each layer a (significant) name may be of help. Please note in the screen capture that the new layer is placed above the original, so that changes will be readily apparent. Also, note that any changes made will be to the layer that is currently active — the new layer in this capture.



Normal

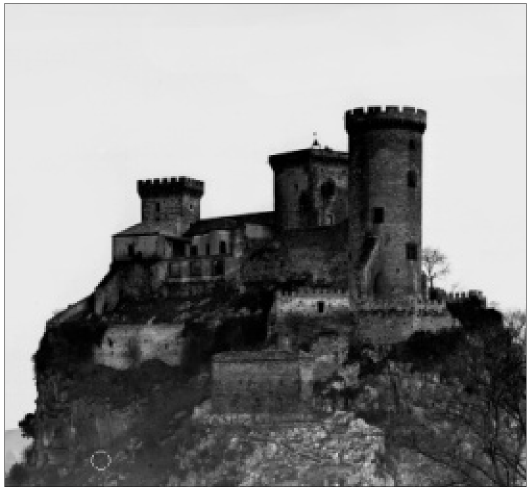
Opacity: 100%

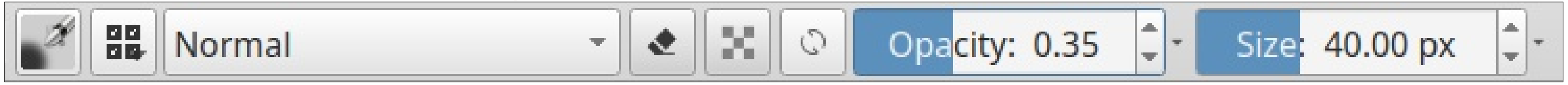
modifications

original

Let us begin with the picture borders. Most defects are to be seen around the top edge, and affect the sky. We can cover them with a paint color similar to the sky itself, making them disappear. Choose any brush you feel comfortable with; Krita has a large selection, comprising both pencils, felt-tips, and an airbrush. This latter is my personal choice, with a low opacity setting and relatively small size — both of these are configurable with horizontal bars set just beneath the top menu bar.

Begin by right-clicking on a suitable piece to sky to copy that color and make it your brush's active color, and then proceed by applying small touches. Brushing ever more lightly and with many applications to build color is the ticket. In passing, the same technique may be used to wipe the sky of its ungainly black dots. The end result is not perfect, with its very uniform appearance, but it is still rather more pleasing than in its original state.

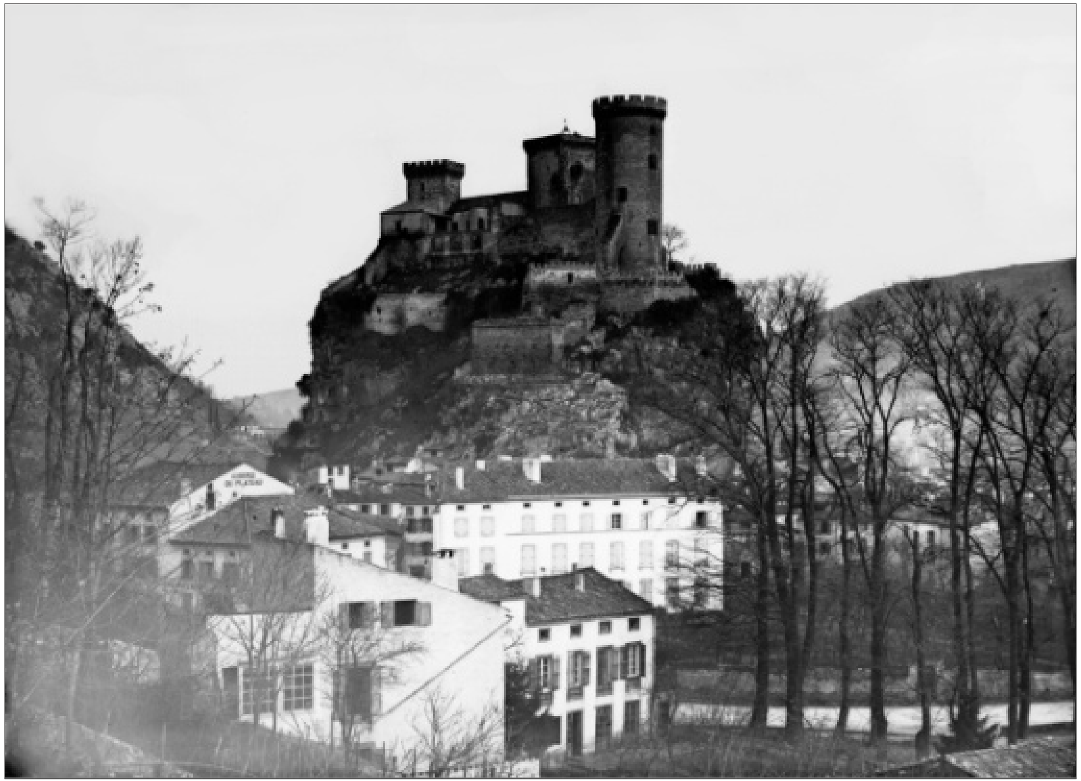




Normal

Opacity: 0.35

Size: 40.00 px

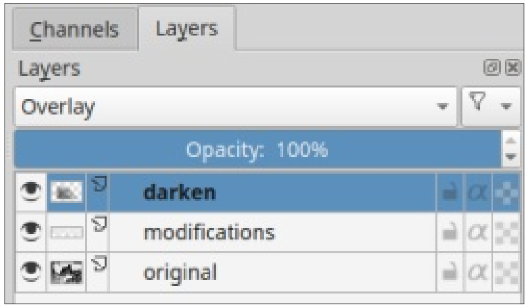


As can be seen above, the extreme left of the image has a thin row of dark pixels running from top to bottom. Partly an artifact, and partly due to the original photographic process, this defect can be resolved in part by painting in details, thus reconstructing the scene by extending the visible part of the image left towards the edge. However, this technique is quite time-intensive, and for little gain: just one or two rows of pixels at the very leftmost edge of the picture. For this reason, it may be more effective just to crop the image, cutting off these stray leftmost pixels.

All along the lower edge of the image, we find a similar dark strip that can also be cut out by cropping. We are left with a rather better edge all round, except for the dark spot in the extreme lower left corner, and the lighter blemish around it.

To clear up the light-colored

blemish, a natural tendency would be to use some dark color on it. But it is clear that applying color directly would only create a dark mass, with all details lost beneath it. To avoid this mishap, we can create a third layer. This time, its mode of combination with the layers beneath it should not be "Normal", but either "Darken" or "Overlay". In this particular application, the latter gave me the best results.


Channels | Layers

Layers

Overlay

Opacity: 100%

darken

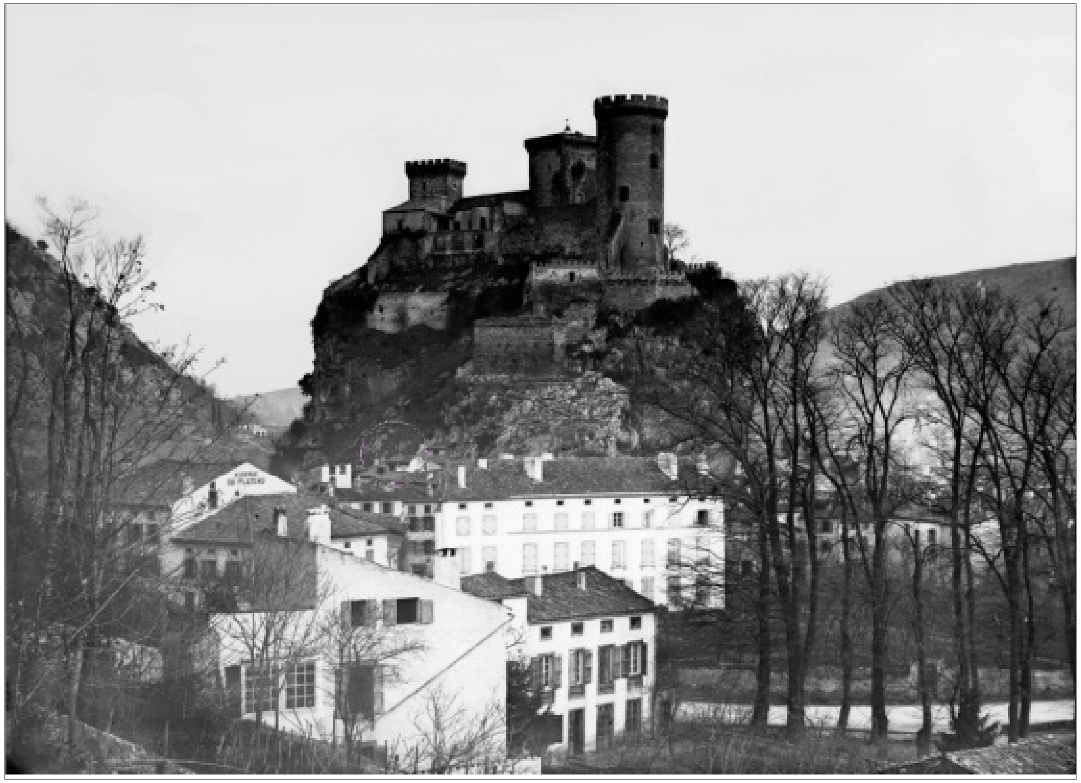
modifications

original

Now, inside this new layer, select a dark color -- or even flat black -- and slowly, lightly, use a low opacity setting to airbrush some color over the area affected. As you proceed, you should see how details such as tree limbs or the fences regain some contrast. With some application, a satisfying result can be obtained. The whitish blemish is no longer quite as apparent.

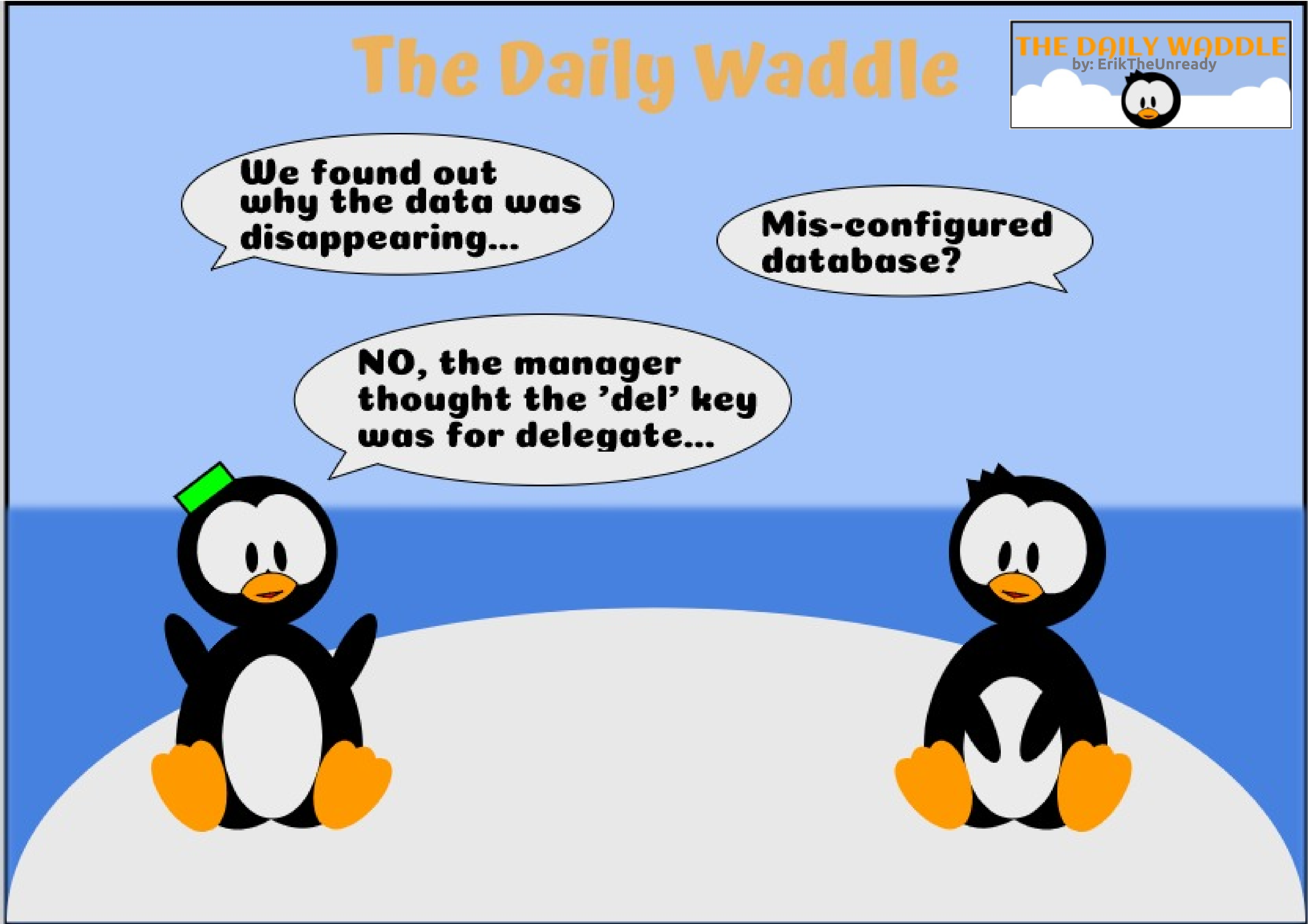
Unfortunately, the area beneath the small black patch in the extreme lower left corner has lost all of its detail. Very little can be done here to remedy this loss of information, so it may be best just to leave it as it is. Since the light-colored blemish has been adjusted, the difference in tone between both areas is lessened, and the eye is not drawn to the dark patch quite as easily as before.

In this part of our series, we have set up our image in Krita, and used two new layers, one a covering layer to hide defects beneath new paint, and the second a darkening or overlay layer to darken the original details and set the overall density of a specific area back to an acceptable average. I encourage the reader to practice some or all of these techniques on this image, or any other of your preference, and we will be back in the next part to try solve the issue of the dark castle and trees to the right of the photo. Until then, take care!



Alan holds a PhD. He teaches comp sci and eng at Escola Andorrana de Batxillerat. He has given GNU/Linux courses at the Uni of Andorra and taught GNU/Linux systems admin at the OU of Catalunya.

contents ^


The Daily Waddle
THE DAILY WADDLE
by: ErikTheUnready
We found out why the data was disappearing...
Mis-configured database?
NO, the manager thought the 'del' key was for delegate...

LINUX LOOPBACK

Written by S. J. Webb

MAYBE NEXT MONTH

SJ Webb is a researcher coordinator. When he is not working, he enjoys time with his wife and kids. He thanks Mike Ferarri for his mentorship.

contents ^

Written by Richard Adams

Return To Retrogaming

We're back for a palate-cleansing revisit to Linux Retrogaming this month, after our last two months of recipe management (see what I did there?).

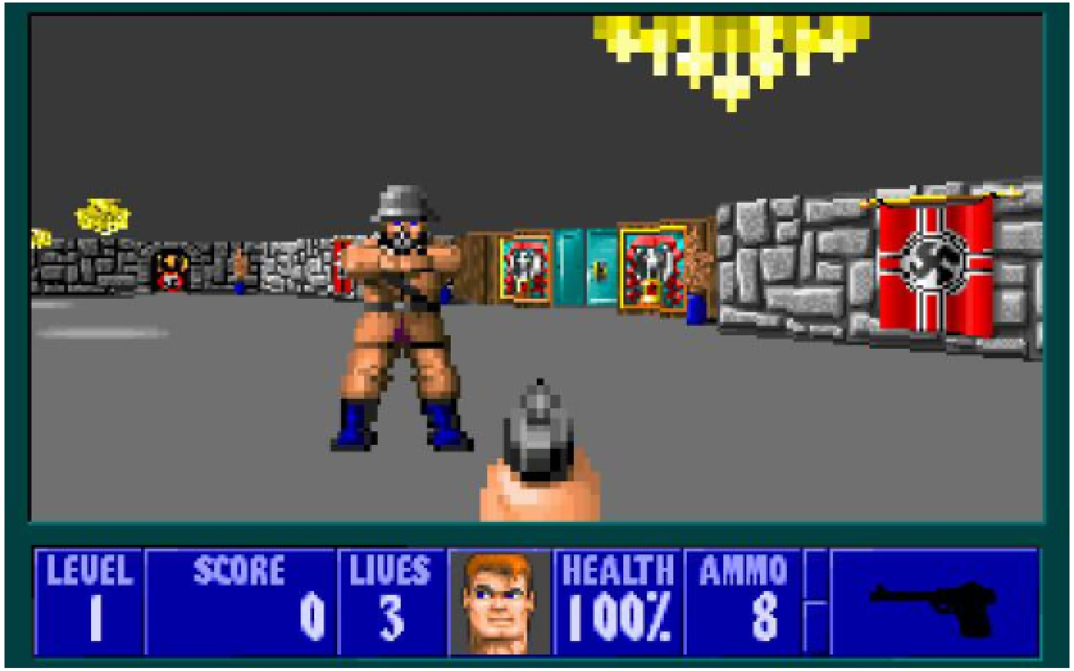
Prepare To Meet Your (Chocolate) Doom!

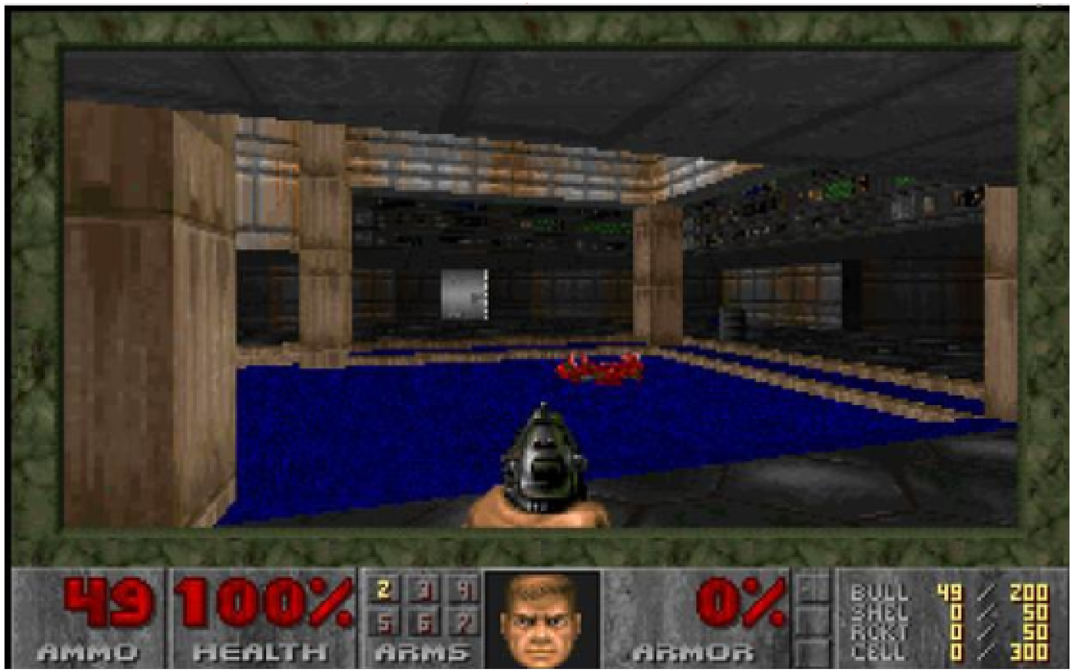
Back in 1992, the rage in computer gaming was the impressively immersive experience of playing Wolfenstein 3D. Created by id Software and distributed by Apogee, it was probably the first really successful first person shooter, or FPS, game:

Cartoony though it may be by today's standards, in 1992 it was absolutely revolutionary. As World War II spy hero B. J. Blazkowicz, you had to escape your German captors and fight your way past Nazis and Wehrmacht guards (and even Hitler himself!) to freedom. I had a close friend who refused to play the game because you would occasionally have to shoot German Shepherds. But they were, after all, NAZI German Shepherds, so I was personally OK with that part of it.

But id was not content to rest on its laurels. Returning to work from lunch break one 1993 afternoon, I stopped briefly at a used computer store (you know, one of those places that EvilBay helped put out of business) and, on a whim, picked up a shareware diskette for something called … DOOM. "Kind of a dumb name," I thought, as it didn't really tell you much. I got home that evening and installed it, then started it up. Amid the roar of angry monsters, I goggled at the game's attract mode. It was FAR more real looking than Wolf3D, with weathered metal walls, barrels of toxic slime, zombified marines and demonic imps, and there I was with just a lowly pistol (in that respect, just like Wolf3D).

Doom took the gaming world by storm and laid the groundwork





for most of today's gaming, as the FPS shooter has become far and away the dominant gaming genre. Although it's over 25 years old, Doom's timeless gameplay, devilishly clever level designs, and undeniable immersive quality, mean it's STILL fun to play. Observe this delightful Fine Brother's Entertainment React video, where they show the game to modern-day teens and elicit some entertaining reactions: https://www.youtube.com/watch?v=GC7Onhm3tto

Doom was famously ported to Windows 95 as a proof-of-concept prior to Win95's release, and it has been ported to every platform imaginable, including insane ones like the TI-83+ calculator and certain Kodak digital cameras. It may be the most widely ported piece of commercial software ever, and (of course) it is available, in multiple forms, for Linux. The one I use is Chocolate Doom, but there's also gzdoom, prboom, freedoom, and probably others that I'm not even aware of. I also have the full Ultimate Doom on the PC side, which I purchased from GOG.com (which I recommend as a good site to find Linux games, as they have a pretty nice variety). You can install Chocolate Doom on Linux from the Synaptic Package Manager. Refer to Everyday Ubuntu from Full Circle Magazine #130 for more details on installing from Synaptic. Briefly, you can search in Synaptic for 'Chocolate', click the boxes on chocolate-doom, chocolate-common, and doom-wad-shareware, then hit Apply. Synaptic will then download and install Chocolate Doom for you.

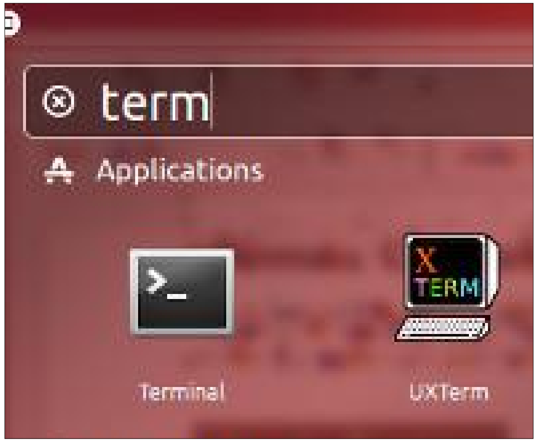
Doom was originally distributed using the then-popular 'shareware' marketing model, something much less common today. When you downloaded Doom for free or paid a nominal fee like I had for the shareware diskette, you got what was called Episode One: Knee-Deep in the Dead. The full game at that time consisted of three 'episodes', the next two of which you could purchase directly from id, keeping their marketing costs down and increasing their profit margins substantially.

If you install Chocolate Doom, you get Episode One only, the shareware version. The game's assets were included in a single file, with the extension WAD (Where's All the Data?), which is what we just installed from Synaptic Package Manager as doom-wad-shareware. The salient difference between the shareware and full version was simply a different WAD. Given this, if you have a full version of Doom, which is pretty affordable at only $5.99 on GOG.com, you can use that game's WAD with Chocolate Doom to play the full version of Ultimate Doom on Linux. In addition, there are thousands of WADS available for free download that do interesting things like turn the enemies into purple dinosaurs (DIE, Barney!), or convert the whole game into a Star Wars game, or an Aliens game, or many, many others.

Copying to a Restricted Location

If you don't have access to a Windows/DOS PC, you will need to have someone who does to install the Ultimate Doom package, if you go the GOG.com route. Once it's installed, the game's DDOM.WAD can be found in C:/GOG Games/DOOM/. You will need to transfer this file onto your Linux machine. Probably the easiest way to do so is via a USB thumb drive. After transferring the WAD file from Windows, put the USB stick in your Linux machine. We now need to invoke the File Manager using elevated privileges to give us access to a file location that Ubuntu will not normally let us use. We do this by invoking Nautilus (the File Manager) as a Super User.

Go to the terminal (usually the third icon from the top on the Launcher that runs down the left-hand side of the screen), or go to the Dash – top icon on the Launcher on screen left and type in term, then click the terminal icon):

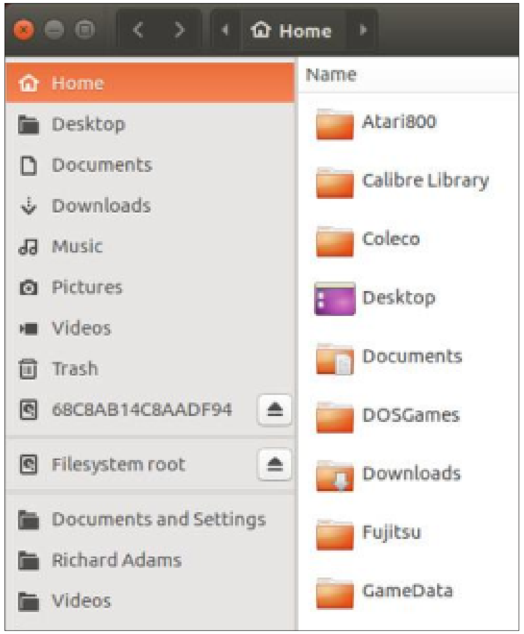


and from the terminal type:

```
sudo nautilus
```

then enter your Superuser or admin password that you set up during Linux's initial installation.

This will start a SuperUser File Manager session that will allow you to copy the DOOM.WAD file to the necessary location. First, find the DOOM.WAD on your thumb drive using the shortcuts on the left to navigate, then right-click the file in the right-hand pane and Copy. Now, go to the filesystem root directory. You will see it on the left:



Click it, then double-click USR on the right. Scroll down to SHARE, double-click, scroll to GAMES and double-click, then double-click on DOOM. Right-click on an empty part of the right pane and Copy.

This is a VERY valuable thing to know how to do in Ubuntu, since a disadvantage of the Ubuntu distro is that it's much harder to do admin-level file management than in many other Linux distros, so be sure and file this information away for potential future use!

Ultimate Chocolate Doom!

Chocolate Doom does support command-line parameters that allow us to customize various aspects of how it runs, including selection of WAD files. If we want to run Ultimate Chocolate Doom with our Ultimate Doom WAD file, we can invoke the terminal (again, go to the Dash – top icon on the Launcher on screen-left and type in term, then click the terminal icon).

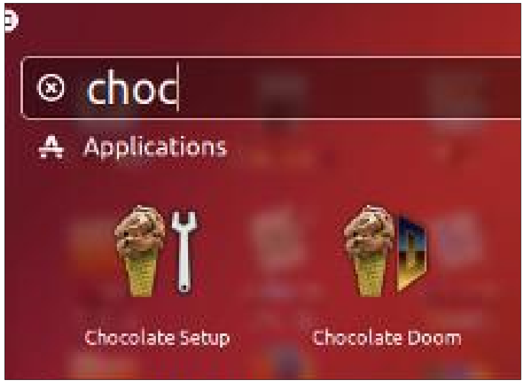
From within the terminal type:

```
chocolate-doom -iwad DOOM.wad
```

then hit <Enter>, and Ultimate Chocolate Doom will start, but that's not really convenient, is it? Let's use a little Linux knowledge to make things easier.

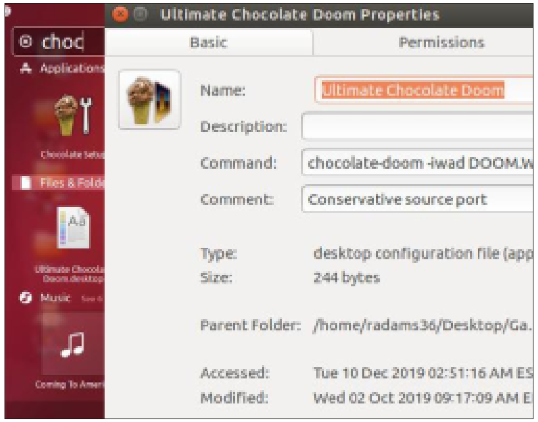
Ultimate Chocolate Doom Desktop Setup

First, use the Dash again and search for 'chocolate', this should find Chocolate Doom. Click and hold the Chocolate Doom icon, then drag and drop to an open area on the desktop. This will create a new desktop Chocolate Doom icon:



The icon is actually an SLK, or Symbolic Link, file. Symbolic Links can be very useful in Linux (for example, you can save space by using SLKs in multiple file locations instead of multiple copies of the same file), and here it will give us an Ultimate Chocolate Doom launcher so we don't have to always type in the command-line.

Right-click the Chocolate Doom icon on the desktop and go to Properties:



Name: Ultimate Chocolate Doom
Description:
Command: chocolate-doom -iwad DOOM.W
Comment: Conservative source port
Type: desktop configuration file (app
Size: 244 bytes
Parent Folder: /home/radams36/Desktop/Ga.
Accessed: Tue 10 Dec 2019 02:51:16 AM ES
Modified: Wed 02 Oct 2019 09:17:09 AM E

Now, let's rename this SLK to Ultimate Chocolate Doom and put in the new command-line. Use the same command-line as before:

```
chocolate-doom -iwad
DOOM.wad.
```

Now we can launch Ultimate Chocolate Doom routinely from our desktop and conveniently save humanity (yet again) from the demonic hordes that always impose in the computer gaming world!

Richard 'Flash' Adams spent about 20 years in corporate IT. He lives in rural northwest Georgia, USA, with his adopted 'son', a cockatiel named Baby.

contents ^

UBPORTS DEVICES

Written by UBports Team

COMING SOON

When the project manager is looking for someone to write the specs...

Written by SJ Webb

Mopria and Google Cloud Printing

Recently, Google announced the death of Google's Cloud Printing (GCP). The service ends in 2020. Surprisingly, several people are disappointed in this decision. A few network managers are reliant on GCP at their worksite. I personally never had GCP work reliably on my devices.

At one time, GCP was needed for the Chromebook printing support. The early Chrome OS did not have access to the Android Apps from Brother, HP, Epson and other printer vendors. Using an Epson printer and Epson App, I can print locally from my Chromebook reliably. Yet my Brother laser jet could not connect and print from my Chromebook. I can understand the decision to kill GCP.

Not all printer vendors are in the Chromebook platform. I know HP and Epson are working well. Yet Brother is a dismal result. One solution being offered by Google is the Mopria Print Service app.

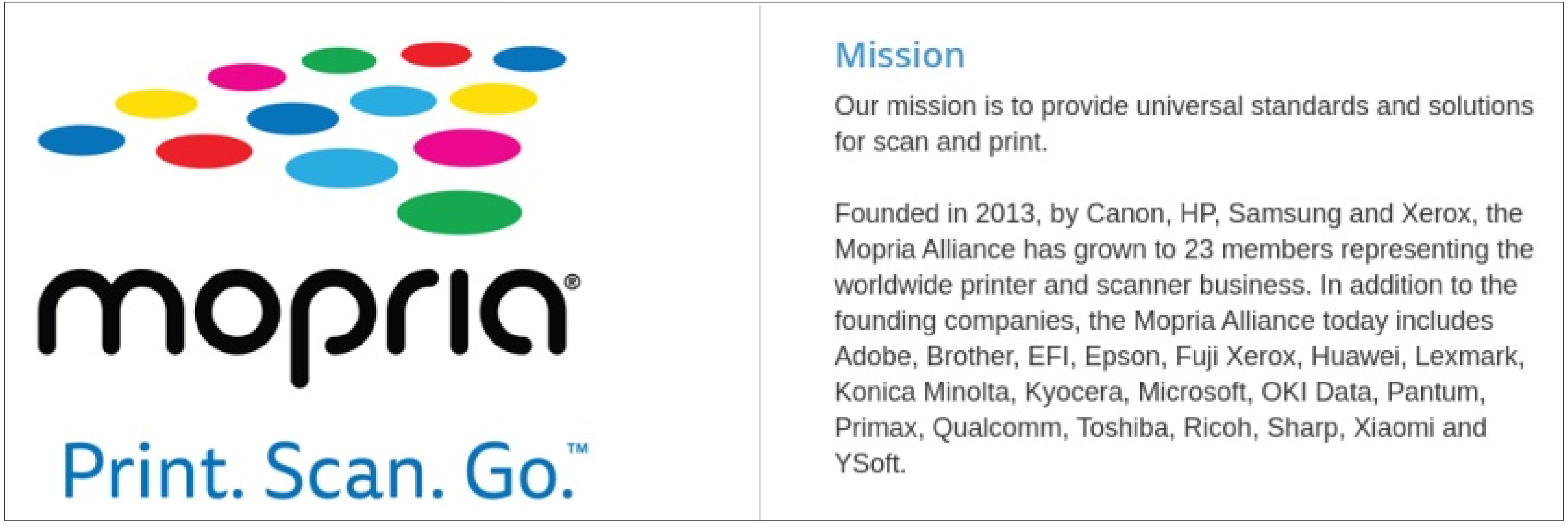
The Mopria Alliance is a non-profit that provides universal standards and solutions for scanning and printing. It is industry supported by the major technology names like Samsung, Epson, and Xiaomi.

Apparently the Mopria App is cross-platform on major printers and scanners.

I read the online reviews, and the app is pretty rock solid for newer printers. It connects to your wireless printer using your local private home router. The app has over 19,000 reviews and is produced by the corporate vendors; I doubt this app would be malicious. It appears Mopria works with only Android devices, Chromebooks, and the Windows OS.

The app will not work if you leave your local network. GCP would print over any network. The main gripe about the Mopria App is the local wifi printing only. Yet many users are fine with these limitations. I see this being a possible solution for enterprises that have onsite employees. Yet I am not a network manager, and my opinion is possibly clouded by ignorance.

SJ Webb is a researcher coordinator. When he is not working, he enjoys time with his wife and kids. He thanks Mike Ferarri for his mentorship.



Mission

Our mission is to provide universal standards and solutions for scan and print.

Founded in 2013, by Canon, HP, Samsung and Xerox, the Mopria Alliance has grown to 23 members representing the worldwide printer and scanner business. In addition to the founding companies, the Mopria Alliance today includes Adobe, Brother, EFI, Epson, Fuji Xerox, Huawei, Lexmark, Konica Minolta, Kyocera, Microsoft, OKI Data, Pantum, Primax, Qualcomm, Toshiba, Ricoh, Sharp, Xiaomi and YSoft.

contents ^

Written by Ronnie Tucker

Write For Full Circle Magazine

Guidelines

The single rule for an article is that **it must somehow be linked to Ubuntu or one of the many derivatives of Ubuntu (Kubuntu, Xubuntu, Lubuntu, etc).**

Rules

• There is no word limit for articles, but be advised that long articles may be split across several issues.

• For advice, please refer to the **Official Full Circle Style Guide**: http://bit.ly/fcmwriting

• Write your article in whichever software you choose, I would recommend LibreOffice, but most importantly - **PLEASE SPELL AND GRAMMAR CHECK IT!**

• In your article, please indicate where you would like a particular image to be placed by indicating the image name in a new paragraph or by embedding the image in the ODT (Open Office) document.

• Images should be JPG, no wider than 800 pixels, and use low compression.

• Do not use tables or any type of **bold** or *italic* formatting.

If you are writing a review, please follow these guidelines :

When you are ready to submit your article please email it to: articles@fullcirclemagazine.org

Translations

If you would like to translate Full Circle into your native language please send an email to ronnie@fullcirclemagazine.org and we will either put you in touch with an existing team, or give you access to the raw text to translate from. With a completed PDF, you will be able to upload your file to the main Full Circle site.

Reviews

Games/Applications

When reviewing games/applications please state clearly:

- title of the game
- who makes the game
- is it free, or a paid download?
- where to get it from (give download/homepage URL)
- is it Linux native, or did you use Wine?
- your marks out of five
- a summary with positive and negative points

Hardware

When reviewing hardware please state clearly:

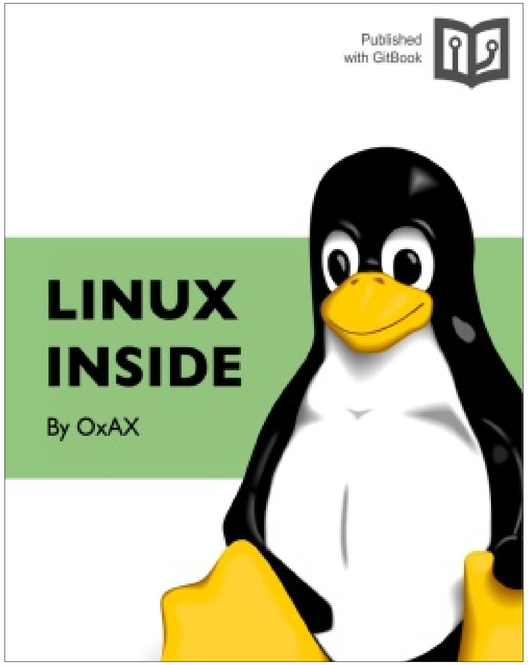
- make and model of the hardware
- what category would you put this hardware into?
- any glitches that you may have had while using the hardware?
- easy to get the hardware working in Linux?
- did you have to use Windows drivers?
- marks out of five
- a summary with positive and negative points

You don't need to be an expert to write an article - write about the games, applications and hardware that you use every day.

BOOK REVIEW

Written by Erik

Linux Inside



Linux Inside
Author: 0xAX (Alex)
Author's twitter account:
https://twitter.com/0xAX
Web page:
https://0xax.gitbooks.io/linux-insides/
License: Creative Commons

This is a FREE resource with multiple contributors on Github. If you feel you can contribute, sign up for a Github account and help.

Do you want to know more about the Linux kernel? Would you like some insights into what goes on behind the scenes, so to speak? Well, then this book is for you. The book tackles low level programming in the kernel. If ever you were curious about your CPU and how it works, well, there is some of that in here too. If you are unsure if this is something you would like to read, I suggest reading the introduction. Bear in mind that you would have to have a fleeting knowledge of Assembler and C. That said, the explanations are clear, so you do not need to know the in-and-outs of these languages to follow along. There are also references given to books you can read if at any time you want to brush up your knowledge on any of the subjects covered in this book.

The step-by-step approach is covered in detail and you immediately know that the author has experience at every one of these steps. Speaking of every step, the author provides links after each of these steps for you to engross yourself in. This is the reason it has taken me so long to read this book as it speaks to the ADHD part of my brain, and SQUIRREL! Yes, if you are like me and do not have ADHD, all these interesting links will have you flitting across sections of the book like a butterfly in a field of flowers. I also found myself Searching the internet for terms I did not know. Good thing this is an online book! If English is not your first language, keep another browser tab open. This, again, will contribute to your ADHD. The third thing to make you feel like an ADHD kid is there are Wikipedia links embedded into the text at all levels. The nice thing is that at any stage you can highlight something and click the plus-sign to add a comment.

If all this is too distracting for you, you have the option to download the book as a PDF-file. Do not be fooled, there is no light reading here. These are facts and code cut right to the bone. If you have ever read – I am using that term VERY lightly, as nobody reads them – CPU developer references, you may understand why this book exists. It is a calm patch in a sea of turmoil. There is a road you can follow that takes you from start to finish, even though there are "side quests". It feels structured and well thought out. Staying with my RPG metaphor, when you level up your Linux knowledge, this is the way to do so. Do not feel bad having to re-read anything. I am actually making an appointment in my calendar to re-read this.

Since this book is free, I encourage you to get it, read it, forget about it, and read it again. The reason I am saying this is because the book is not 100% complete. You will find sections that still have to be written. If this is your thing, then jump right in, I look forward to reading your insights. Otherwise, there is plenty to keep your mind occupied in the meantime.

Since the book is incomplete, I will not rate it, but so far, it is a must-have book for any Linux enthusiast!

REVIEW

Written by SJ Webb

Stadia

My last console was a Playstation 3. After years of game play and streaming, it finally gave the red light. I was not too enthused with purchasing a new console, so I sold my games. I used the money to buy my first Roku to keep streaming. The price of the current mainstream consoles at that time were greater than $250.00. I could not justify buying a console at that time.

I purchased the Stadia from the Google Store. I placed my order in June 2019. I paid $150.00 for the Founder's Edition. The required hardware is a Chromecast Ultra and the Midnight Blue controller.

The instructions were 90% accurate. The first step is to connect the Chromecast Ultra to your TV using the HDMI port. You download the Google Home App to an Android phone or tablet. You then use the Android device to activate the Chromecast Ultra. The next step is to download the Stadia App. You use the app to activate the controller. During the process, you enter a custom serial number and activation code from Google to activate the Stadia. (The custom serial number could be an effort from Google to prevent resale). The controller is the console in some respects and it runs on Linux!

"*This [Google Stadia] starts with our platform foundations of Linux and Vulkan, and shows in our selection of GPUs that have open-source drivers and tools. We're integrating LLVM and DirectX Shader Compiler to ensure you get great features and performance from our compilers and debuggers.*"

- Dov Zimring, Stadia Developer Platform Lead

I use the Stadia app to purchase the games. The Stadia comes with 2 games and 3 free months to the Stadia Pro subscription. The games are Samurai Shodown and Destiny 2. I mostly play Destiny 2. My router is within 8 feet of my router, and I have high speed internet. I do not have any lag or noticeable pixelated game streaming. Overall I am quite happy with my purchase.

However Google had a poor debut. The builtin Google Assistant is not active. There are accounts of people not getting their activation code. There is game lag if you have poor connectivity. The 4k game play is not true 4k. I believe the Google game stream is 1080 or 780, and then upgraded by the Chromecast Ultra to 4k. Currently there are only 22 games coming out in the next year. Yet Google promises more titles in the next year.

I agree the launch is less than perfect, but you are paying a lower price for a console. So I really disagree with the mob mentality. This is a cheaper solution to an expensive console. Additionally, cloud gaming is new. I am sure with Google's expertise in the cloud, the experience will improve. Personally, I am quite happy with the Stadia. It has a relatively small footprint in my house. Plus I game when the kiddos and wife are in bed. If you are looking for cutting edge game play, go with a console or a Linux-Steam combo.

INTERVIEW

Written by Erik

Ron de Jong - Finalcrypt Dev

Website: http://www.finalcrypt.org

From the website: "*Today's cyber espionage comes from hidden spyware waiting for you to unlock your drive. Disk Encryption no longer protects! Even when you're logged on, unopened files have to remain encrypted. Only File Encryption stops spyware reading your files. Also, most crypto software uses broken AES or asymmetric crypto soon broken by The Shor's Algorithm with Quantum Computers. This cyber espionage pandemic has to be stopped by unbreakable One-Time Pad File Encryption. That is why 'FinalCrypt' was built.*"

Finalcrypt is, by all accounts, something amazing. It follows the Unix philosophy of doing one thing and doing it well. The main problem of cryptography is getting your "key" from one person to another, without it being intercepted. Hence our current shared key cryptography. What if you do not need to give the key to anyone? Now this becomes a real secret. (A secret is something only you know). One-time pad, (https://en.wikipedia.org/wiki/One-time_pad) is what you are looking for, and this is what Finalcrypt is all about. Finalcrypt is the brain-child of Rob de Jong, who is also the programmer and project maintainer.

Continuing our series of interviews with open source heroes, in this issue we would like to introduce you to the brilliant creator behind Finalcrypt, Ron de Jong. I asked Ron for an interview and he very kindly accepted to answer my questions.

Q: Ron, thank you very much for your time. First, can you tell us something about yourself? How old are you? Where do you live? What do you do for a living? Where did you study?

A: In September, I'll be celebrating my 50th birthday; living in Zaandijk (close to Amsterdam) in The Netherlands. At 13, I started programming (Atari 600XL) and became fascinated and obsessed with programming, but dropped out of school at 15 when my father suddenly passed away. Later, I picked up and finished Telematics & LAN Management education, and started working for 15 years as a UNIX Systems & Software Engineer for international Telecom, Internet & IT companies. A period of intense work and training, combining programming and systems engineering, allowing me to start contracting from 2006. After 2008, things went downhill, and, in 2012, I ended my paid career realizing I could no longer function without severe stress in bright light, noisy, social and commercial environments, and in 2013 I was diagnosed with (severe) Autism (Asperger), and decided to retreat in rest-contributing society from home (in an autism friendly way) – developing Free OpenSource (Human Rights) Software.

Q: Encryption is a very complicated subject, when and how did you become interested in it?

A: Throughout my career, encryption always played a part, but more as something you'd use to hide and protect confidential information – assuming that encryption algorithms were as secure as the authorities claimed. Over the years, more and more signals came out that crypto algorithms weren't as secure as claimed, and vulnerabilities were deliberately exploited to expand espionage on civilians. Cyber security news became jaw dropping during the past decade. Thanks to people like Edward Snowden, Julian Assange, and established news-media, we now know we can't trust national security agencies to respect privacy. I finally came to realize that One-Time Pad Encryption is the only truly unbreakable encryption. A straightforward algorithm that can't be reversed or brute-forced – its encryption comes from only irreversible random bit patterns.

Q: Would you care to share some insights into what goes into coding something like finalcrypt?

contents ^

A: Being a crypto-sceptic, I didn't want to rely on any cryptographic library, so I decided to build encryption from the ground up – starting at the bit level. Some years back, my initial idea was to XOR (toggle) data-bits with correlating personal key-bits coming from personal images or videos, because even the fastest cluster of supercomputers can't brute-force (render all bit combinations) until it matches a picture (or video) of me and my cat. So FinalCrypt started out without a key generator, and using existing pictures or videos as keys. Then I got into discussion online with crypto experts who claimed that FinalCrypt actually was One-Time Pad Encryption, but broke OTP rules and could therefore not be 100% unbreakable. Then, in version 2.6.0, I built-in a FIPS140-2 & RFC1750 compliant True Random Number Generator to generate OTP keys allowing manual OTP encryption, and, in version 5.0.0, I added Automatic Key generation, which allowed FinalCrypt to (batch) encrypt all files with One-Time Pad security by default.

Q: Finalcrypt obviously was created in response to something? Care to share?

A: After my autism diagnosis I tried to understand my social limitation in an attempt to understand what social behaviour really is, but couldn't find objective answers (not even from academic sociologists) so I started to philosophize about social behavior and how evolution evolved social behavior to increase our chances of survival, and concluded that group hunting behavior is about hunting competences of stealth behavioral observation, covert communication, cooperation, cunningly planning deception and attack, overthrowing opponents and enemies. This primal instinct that we humans still have and evolve – keeping us on top of the food chain. Unfortunately, we humans also use these competences against each other to dominate and exploit. Mostly for economic reasons. From that moment on, all pieces of the puzzle fitted together and I was determined to design and build unbreakable encryption from the ground up – not trusting any ClosedSource linked encryption libraries in widely accepted encryption standards.

Q: I see you opted for the creative Commons license; was this always your goal?

A: Starting off, I chose GPL3, but soon after I realised that GPL3 allowed anyone to build-in backdoors, and publicly release it, so I changed to Creative Commons License not allowing anyone releasing derivative works. Changes for personal use are allowed.

Q: Do you have any metrics on how many people use this software?

A: Yes, the website currently counts 1917 different users (unique ip addresses). This number grows with 5 - 50 users / day, counting from 2019-04-01. This can be measured because FinalCrypt (at start-up) automatically checks for updates once every 24 hours.

Q: Can one use finalcrypt within an encrypted volume like LUKS or a Veracrypt volume?

A: Yes. FinalCrypt (like most applications) has no notion of any underlying disk encryption, and just creates files and writes file-data to the underlying physical (or logical presented) file-system. Just tested FinalCrypt on LUKS successfully (with different file-systems).

Q: What difference does the underlying file system make when you have to code encryption software?

A: Mainly file-system meta-data attributes like what timestamps does the file-system support, cloning modification timestamps from source to target file, and what file-system permissions are valid or invalid as data source and data target, and whether directory & file links should be ignored preventing double-directory looping and therefore double encryption attempts.

Q: Since the software is free, how can someone support you or say thank you?

A: I appreciate every "thank you", but FinalCrypt really depends on publicity as I don't have the financial means to advertise, so users can really thank me by sharing online.

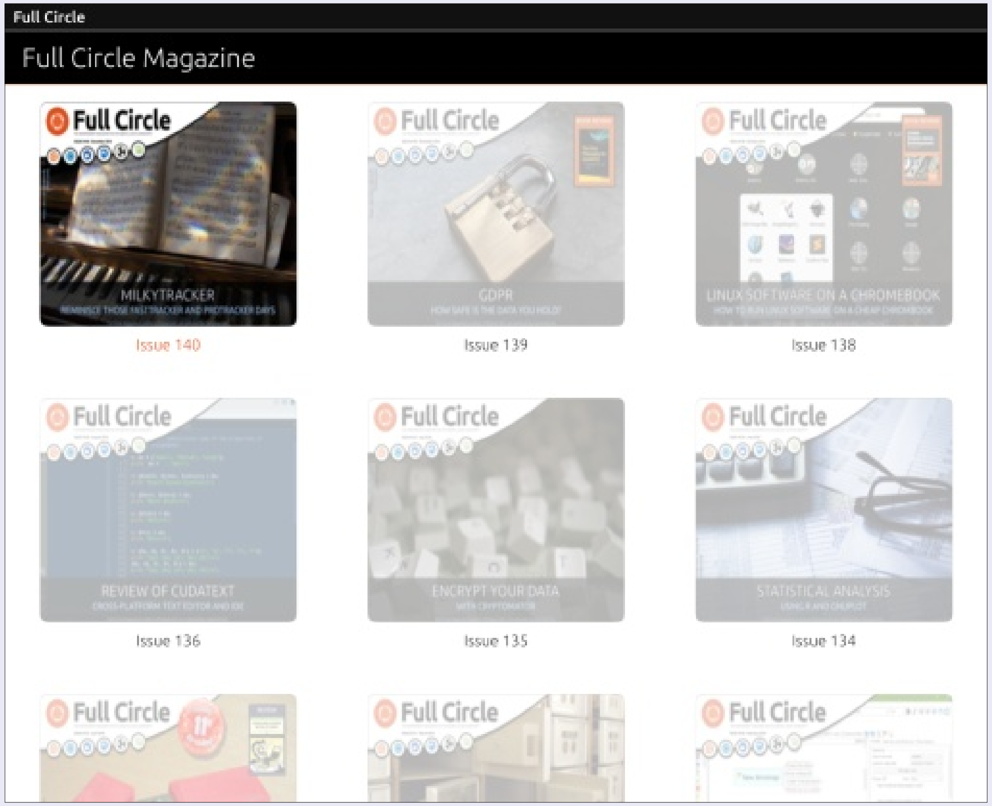
FinalCrypt is also a political statement telling “Big Brother” that we demand privacy.

Outtro

We have dropped some of the more technical questions about Finalcrypt, but if you would like more, drop us an email on misc@fullcirclemagazine.org.

Finalcrypt seems to be an answer to the current work being done by government agencies on elliptical curve encryption, ‘coz, let’s face it, the threat to privacy is internal these days too. (Like when the GCHQ hacked Belgain telecoms). Cryptography is almost the only thing that protects your digital privacy, and, if you care, you should encrypt everything. If you want to keep your data truly safe, you need OTP and thus Finalcrypt. Ron has put in tremendous effort to make your world a safer place. Not all heroes wear capes.

The Official Full Circle App for Ubuntu Touch - UPDATED!

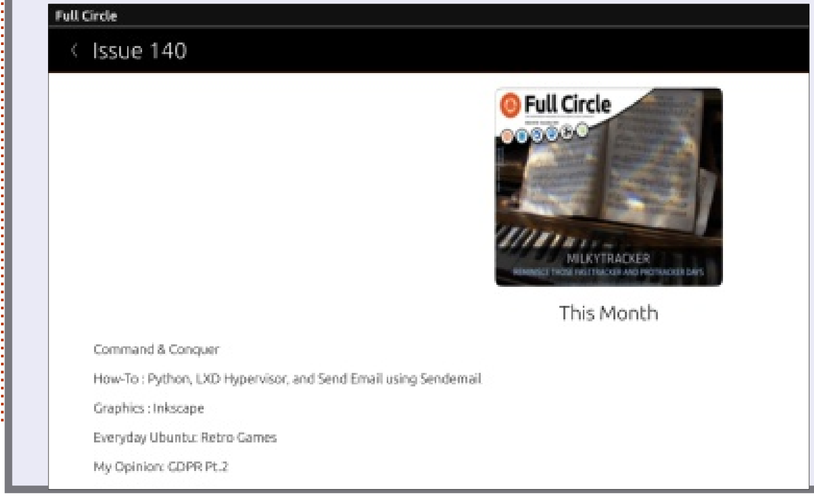


Brian Douglass has updated his FCM app for Ubports Touch devices that will allow you to view current issues, and back issues, and to download and view them on your Ubuntu Touch phone/tablet.

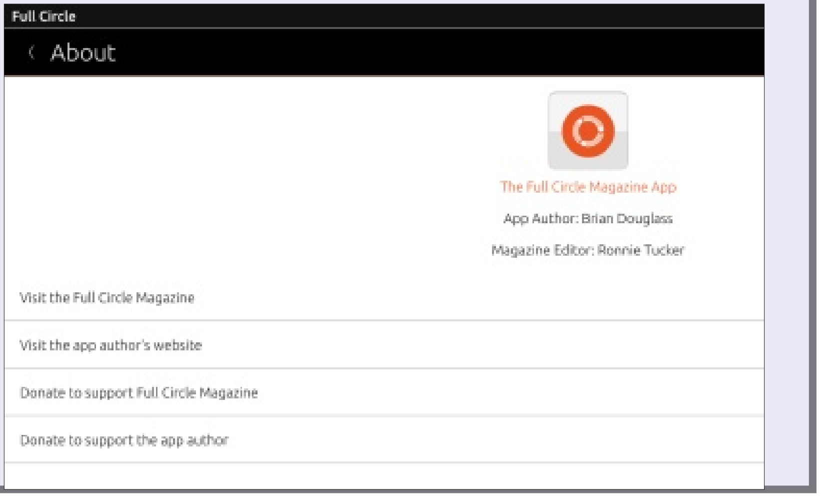
Install

Either search for 'full circle' in the Open Store and click install, or view the URL below on your device and click install to be taken to the store page: https://uappexplorer.com/app/fullcircle.bhdouglass

HUGE thanks to Brian for this.



Command & Conquer
How-To : Python, LXD Hypervisor, and Send Email using Sendemail
Graphics : Inkscape
Everyday Ubuntu: Retro Games
My Opinion: GDPR Pt.2



The Full Circle Magazine App
App Author: Brian Douglass
Magazine Editor: Ronnie Tucker

Visit the Full Circle Magazine
Visit the app author's website
Donate to support Full Circle Magazine
Donate to support the app author

LETTERS

If you would like to submit a letter for publication, compliment or complaint, please email it to: letters@fullcirclemagazine.org. PLEASE NOTE: some letters may be edited for space.

Join us on:

goo.gl/FRTMl

facebook.com/fullcircle magazine

twitter.com/#!/fullcirclemag

linkedin.com/company/full-circle-magazine

ubuntuforums.org/forum display.php?f=270

DistroWatch.com

Put the fun back into computing. Use Linux, BSD.

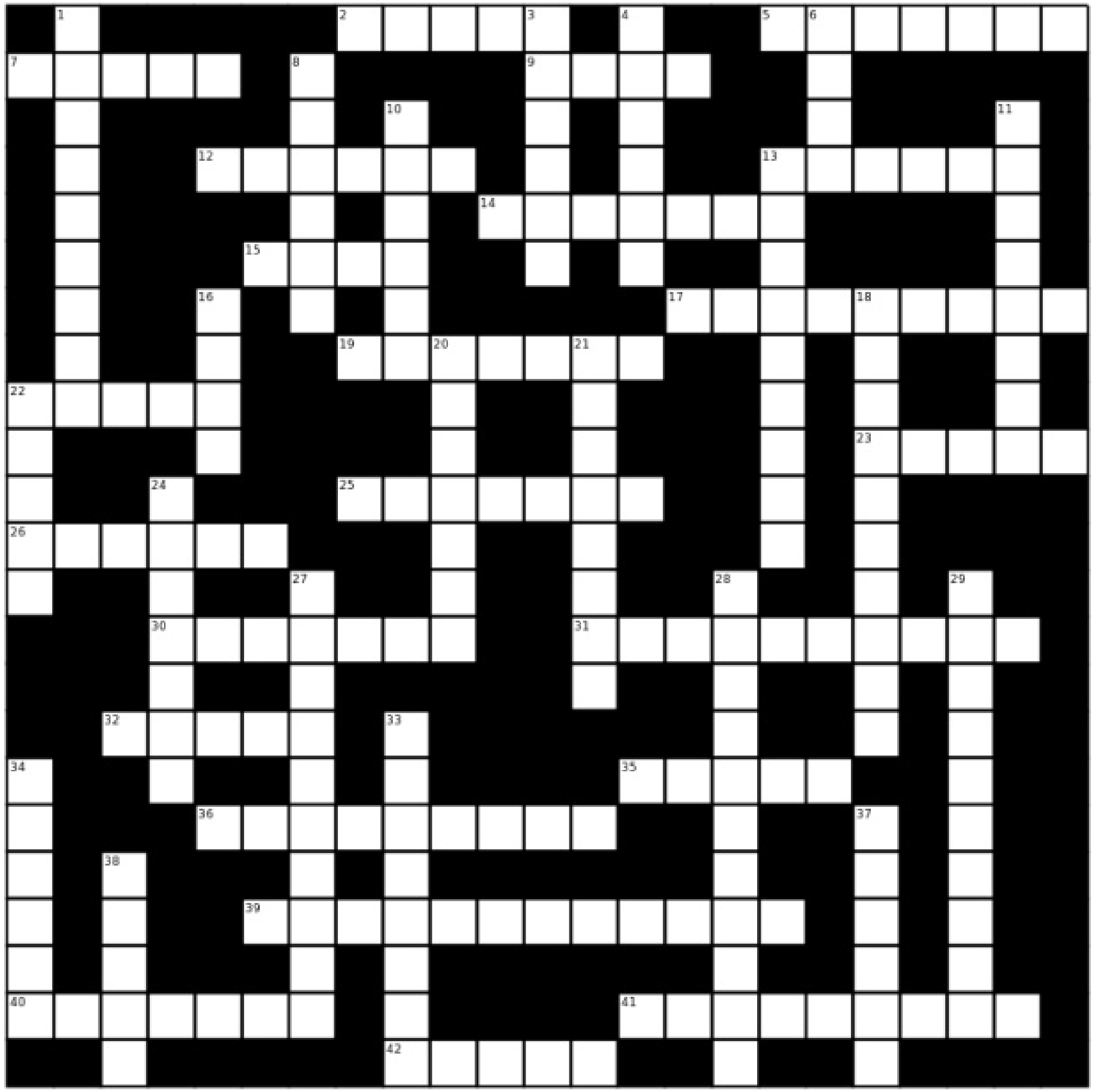
Full Circle Needs You!

Without reader input **Full Circle** would be an empty PDF file (which I don't think many people would find particularly interesting). We are always looking for articles, reviews, anything! Even small things like letters and desktop screens help fill the magazine.

See the article **Writing for Full Circle** in this issue to read our basic guidelines.

Have a look at the last page of any issue to get the details of where to send your contributions.

CROSSWORD - BIG DISTRO CROSSWORD



Compiled by Erik

Answers are elsewhere this issue.

ACROSS

2A: From the Latin, to mean alone...
5A: National park in Venezuela...
7A: A coastal resort town in Sierra Leone...
9A: A steam powered aircraft named after a Roman god...
12A: This was initially created by Daniel Robbins...
13A: A town in Somalia... (and 5 in Numerology)
14A: This OS claim to fame is with access to all Google services...
15A: The second largest Hawaiian Island...
17A: The catch phrase this year was: “Trim the fat”...
19A: When OpenSUSE looks after your data...
22A: The apprentice in “Of the madman and gentleman hunter”...
23A: A fat-free additive found in foods like potato chips...
25A: Known for the taskbar across the centre of the screen...
26A: Has “obcore”, “obextras” and “observice” repositories... (how obscure?)
30A: Like Bit.trip runner or your mother-in-law’s ailments...
31A: Claims to be the "world’s first Dual Layer Live DVD"...
32A: A Japanese poem...
35A: Played by Christopher Walken in the Bond thriller, “a view to a kill”...
36A: The recently dropped the “inux” in Linux...
39A: The Pakku-GUI package manager gives this one away...
40A: Not quite the mixtape you play in the taxi...
41A: Use this to analyse social networks...
42A: Lots of animals have them...

DOWN

1D: If two and two give you four...
3D: When you have only an operating system and a service, and need a name...
4D: Relating to high in the mountains...
6D: In GOT there was a Stark with that name...
8D: A system call operates in exactly the same way as a command...
10D: A city in Southern Italy, that used to be a Capitol of a country...
11D: This is the brainchild of Gaël Duval...
16D: The plural of your self esteem...
18D: Not a rough structure that carries a load...
20D: Named after the sun...
21D: It is a wildcard...
24D: The project was started in October 2005 by Olivier Cochard-Labbé...
27D: This was founded in 2009 by Igor Seletskiy...
28D: This Lüneburg company makes UTM’s...
29D: Usually provided at the end of a meal...
33D: “Get off my cloud” whose name is based on a meme...
34D: Grandmother or old woman in Xhosa...
37D: Another Arch-distro that did not make it out of the gates...
38D: a “resonably secure” operating system...

Q&A

Compiled by ErikTheUnready

If you have a Linux question, email it to: questions@fullcirclemagazine.org, and Erik will answer them in a future issue. Please include as much information as you can about your query.

Welcome back to another edition of Questions and Answers! In this section, we will endeavour to answer your Ubuntu questions. Be sure to add details of the version of your operating system and your hardware. I will try to remove any personally identifiable strings from questions, but it is best not to include things like serial numbers, UUID's or IP addresses. If your question does not appear immediately, it is just because there is such a lot, and I do them 'first-come-first-served'.

I often have to deal with Dell support, and my time zone puts me over to India. I am an HP certified tech, IBM certified tech, Lenovo certified tech, Siemens, etc, but I often get up against that Dell service agent wall. (I will not log a call if there is nothing wrong, my time is more valuable to me). I have also been in the business long enough to have experience. So it is quite a frustration to me when call-centre agents treat you like an end user. "Have you tried turning it off and on again?" To which I reply: "No, as the power supply is broken, this is not possible". This is usually followed by that fake politeness, "Yes sir, I hear what you are saying, but I am going to need you to reboot it". I will log a call for a dead hard drive on a server, then get the runaround, where they do not want to do anything until you put the drive back in the server, reboot the server, and give the error code. "But the drive does not spin up, I am telling you what the problem is as I cannot just willy-nilly reboot the server". Up comes the wall: "but Sir, I cannot be helping you if I do not have that code"... grrrr. But Since India is 3 hours out, I wait until it is 5pm in India before logging the call. That way, I get service from someone elsewhere who is not a robot. Sometimes you need to get a little crafty in IT.

Errata: In FCM#150, I answered Komal incorrectly. This was pointed out by Michael, and indeed, when I updated my Ubuntu I could no longer use .jnlp files either. (This is not the first .jnlp file question we had, so guys take note).

"In issue 150 you had a question regarding jnlp files not working (search for 'N-Able'). I think you will find that support for jnlp was withdrawn from Java. JavaWs and Iced Tea are not included in the V.11 and above.

This is an extract from https://www.oracle.com/technetwork/java/java-se-support-roadmap.html:

The Web Deployment Technology bundled with the Oracle JRE, consisting of the Java Plugin and Java Web Start has a shorter support lifecycle: only five years of Premier Support. The deployment stack was marked as deprecated, and flagged for removal in Java SE 9 and Java SE 10. Oracle Java SE 11 and later versions do not include the Deployment Stack."

———————————————————

Q: This question might be beneath your level, but how do I search for a package?

A: Full Circle Magazine is all for the N00bs! We will not leave you hanging. You can use "apt search" or "apt list" eg. 'apt search nano' or 'apt list nano' - you can also use wildcards.

Q: Hey, my computer is having a problem with the network port. I have been told to update the BIOS to version A17, but how do I know what it is now?

A: Two ways come to mind. One – when booting your computer, pay attention to the first screen that pops up. It will either be at the top or at the bottom, very seldom in the centre or surrounds. Two, open a terminal window and type:

```
sudo dmidecode -s bios-version
```

Q: I can't seem to delete a cron entry off my server that points to pastebin. Obviously, it is reading it from somewhere else and keeps replacing it. I am too much of a newbie to Linux to know where, can you please assist me? I have just started web programming and need my server for a project.

A: Did something not work correctly and you changed permissions to 777? You have malware my friend. You can remove it, but my advice is copy your data off and reinstall. Once a server is compromised and you are new to Linux, finding and shutting all the holes may be a long process for you. You can read up here: https://askubuntu.com/questions/1150346/crypto-miner-malware.

Q: So help me. I want to be a programmer, but where should I start. I mean c++ is what everyone uses, right, but I heard Python is easy. But I don't want to learn something I can't get a job for. Must I learn Java? What is the fastest?

A: Your reasoning is flawed. Programming is not like cars, it doesn't matter if you have a Ferrari when the roads you can drive on allow only 40mph. I suggest finding the Harvard CS50 channel on YouTube and learn along that path. Start with 'scratch' if you have no idea. Once you can program, and do it well in any language, it should not be hard to transition to another.

Q: How can I debloat Ubuntu and have just what I want ?

A: I do not know about "bloat" but you can uninstall anything you do not like. The other option is to come from the other side and install Ubuntu core and add only the packages you want?.

Q: I tried to install focuswriter on a shared computer, but it won't install. Every time I try to install it, it says that I don't have authorisation. Is there any other way to do it?

A: As a user, you have your user folder to yourself. You can download the AppImage of Focuswriter in your user profile and run it from there.

Q: I want to use Rambox as a portable app, rather than installing it, as I want it to run off my USB stick so that I can plug it into any of the computers and run it from there.

A: As per my reply above, have you tried downloading the AppImage and running it from your thumb drive?

Q: I went thrift-store bargain hunting and picked up a sony vaio laptop for $5. Everything worked fine with Ubuntu 19.10, but today the wifi is just gone. I can't enable it at all.

A: The Vaio, iirc, had a hardware switch on the side. Just slide it completely to the "on" position.

Q: Ubuntu is eating too much ram on my laptop. What to do?

A: As you have given me very little to go on, I am going to assume you have vanilla Gnome installed. If you have 2GB of memory, the usage may seem a tad high, you can switch to another variant like Kubuntu that will use less.

Q: I was wondering how do I put the new Chrome snap into firejail? I am using Ubuntu 19.10. On an i5 with 8GB of RAM.

A: As far as I know, snap packages are already sandboxed, but I may be wrong. I do not use snaps, and I do not see anything on the internet, maybe one of our readers has the answer?

Q: I really liked the screenshots of this program - https://github.com/eNkru/electron-xiami . My problem is that it is all in Chinese. How can I make this program in English for my Ubuntu 18.04? If I try Google translate on the page, it won't translate the page, saying that it is already in English.

A: You can try the standard make routine. Nevermind. Try this: https://www.appimagehub.com/p/1323201/ - while I am on the subject.

Q: Guys, I follow https://www.youtube.com/watch?v=In9tvv_ge9I for my Xubuntu. It is not working in Xubuntu 18.04. This is a problem for me.

A: I had a look and tested it the way they did it, and can confirm it does not work. A much easier way would be to edit your launcher to read:

```
env GTK_THEME=Adwaita:dark
geany %F
```

(You can do this as the video suggests, or in your whisker menu launcher, which does not require root permissions.)

Q: How come Ubuntu doesn't need antivirus?

A: Many will disagree with me, but I reckon it does. Viruses are *usually* written for Windows as it has like 80% of the desktop market. (Not being able to walk into a shop and buying a machine with Linux or getting refunded if you do not want Windows has a lot to do with it). However! Linux malware is becoming more prevalent as most users do not have antivirus software.

Q: My system freezes randomly. Full freeze. I dual-boot with Windows 10. It is a 'cheap as chips' laptop, with a Celeron CPU and 2GB of memory. HP, but I can't find the model number. I use Ubuntu 16.04 still, as my apps don't run on 18.04. Since it freezes, I can't even see dmesg output.

A: Freezing can be multiple issues. Try this first: https://askubuntu.com/questions/803640/system-freezes-completely-with-intel-bay-trail

Q: Can my Voyager (XUBUNTU 14.04) run 4k video?

A: If your hardware supports it, yes.

Q: Help me. I did something stupid. I copied all of American horror stories series to my hard drive and it became 100% full. Now I can't use it. Ubuntu can't start Gnome.

A: You will have to free us some space. Try this: https://www.omgubuntu.co.uk/2016/08/5-ways-free-up-space-on-ubuntu

Q: I am booting Ubuntu from my lexar usb drive with high read/write speed. Sometimes it is fast, and sometimes it is slow, on my Lenovo G50. I have persistent space, so I can load drivers. This is my first foray into desktop linux. This fast/slow nonsense is nowhere to be found on Google. Thank you for your time.

A: Some laptops like the G50 series have many USB-2 and only one USB-3 port. The USB-3 port should have a blue interface. See: https://www.lenovo.com/us/en/laptops/lenovo/g-series/g50/

Q: Erm. My lubuntu shows my swap file is in 100% use. How do I fix that?

A: Find the problem, use top or htop. Browsers like eating lots of memory. Run the command: free -h - to see if your main memory is full.

Q: Okay, my Ubuntu dock is showing only on my laptop display and not my external display. Can I have it on there too? I don't want to crane my neck all the time between the two screens.

A: Yes, you actually just need to set that in the settings under "dock".

Q: I don't use bluetooth at all, but it is on in Ubuntu 18.04, all the time. I have to keep turning it off. If I use the wifi switch, it comes back on with wifi. I just want it to stay off.

A: Turn it off in your BIOS and it will not bother you any more.

Q: Can you help me out? I switched to Ubuntu Studio 19.10 and everything is great. However, I have noticed that since updating, I can't scrobble any more. I use Clementine and VLC, but neither scrobble. LastFM shows I scrobbled months ago, but I listen almost every day. <removed> is me on LastFM. <image> <image> This is my setup.

A: I have not used Clementine or VLC in years, but I can try to point you in the right direction. Disconnect Clementine on LastFM and get a new api key. Then run your songs through MusicBrainz Picard, and allow it to rename your

contents ^

files. A good rule of thumb is when a thing does not work across multiple applications, it is the thing. Try another application like Banshee / Amarok, and, if it does not work, take a common American song, named correctly, tagged correctly, and play it. If it scrobbles, your file names and tags may need attention. (Things often change in the background, scrobbling may need a better filename than you have there). You can also fire up nethogs in a terminal, and see if your players actually try to contact LastFM.

Q: Is there a better way to display uname -a to show the distribution details? There is no 'pipe for' column, but I am a newbie to the terminal.

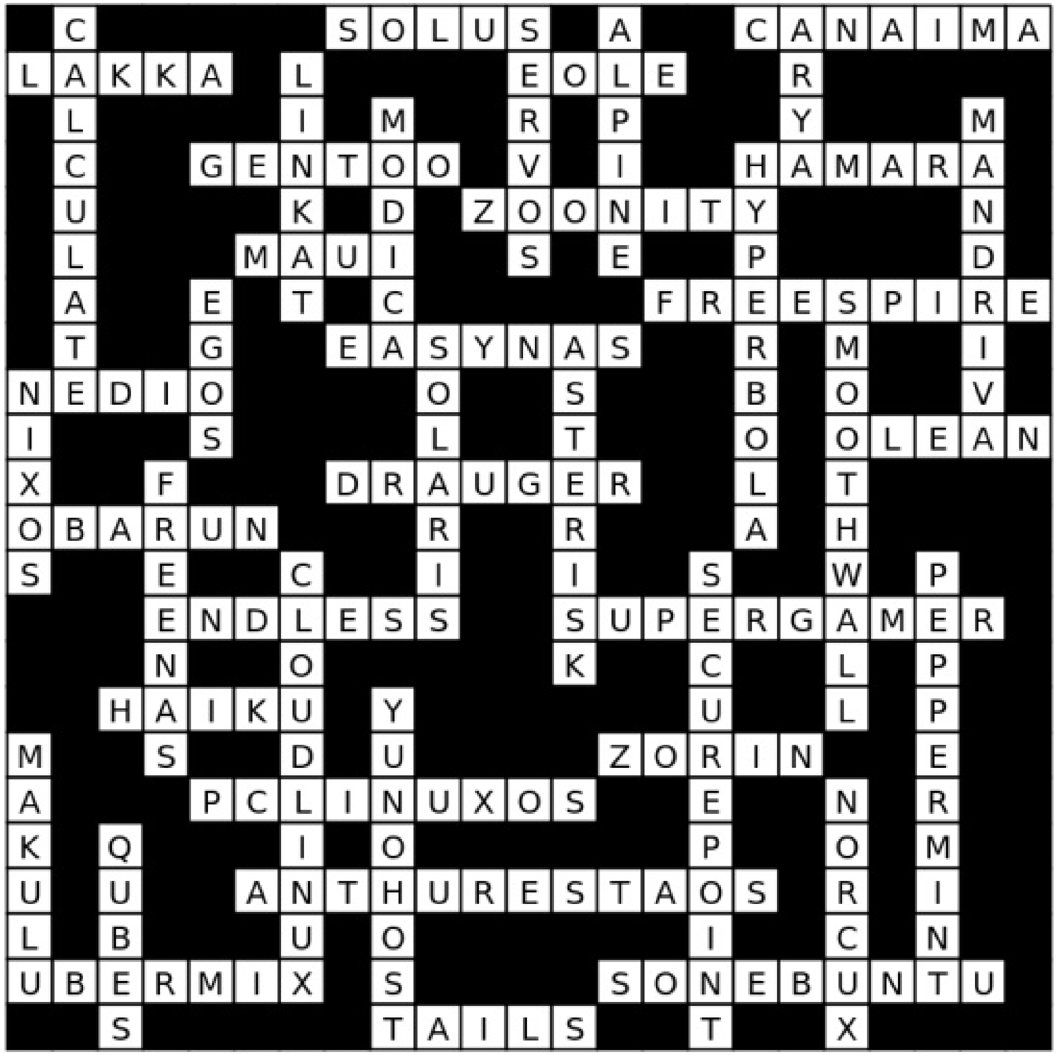
A: It depends on what you mean and what info you require. Have you considered, lsb_release -crid ?? Otherwise there is inxi, or even neofetch, or screenfetch, to do that for you.

Q: I sort of understand what apt-fast does, but I am not clear on the details. Can you help me?

A: Look here: https://askubuntu.com/questions/52243/what-is-apt-fast-and-should-i-use-it

Erik has been in IT for 30+ years. He has seen technology come and go. From repairing washing machine sized hard drives with multimeters and oscilloscopes, laying cable, to scaling 3G towers, he's done it.

Crossword answers:



	C						S	O	L	U	S		A			C	A	N	A	I	M	A
L	A	K	K	A		L					E	O	L	E			R					
	L					I		M			R		P				Y				M	
	C			G	E	N	T	O	O		V		I			H	A	M	A	R	A	
	U					K		D		Z	O	O	N	I	T	Y					N	
	L				M	A	U	I			S		E			P					D	
	A			E		T		C						F	R	E	E	S	P	I	R	E
	T			G			E	A	S	Y	N	A	S			R		M			I	
N	E	D	I	O					O			S				B		O			V	
I				S					L			T				O		O	L	E	A	N
X			F				D	R	A	U	G	E	R			L		T				
O	B	A	R	U	N				R			R				A		H				
S			E			C			I			I			S			W		P		
			E	N	D	L	E	S	S			S	U	P	E	R	G	A	M	E	R	
			N			O						K			C			L		P		
		H	A	I	K	U		Y							U			L		P		
M			S			D		U					Z	O	R	I	N			E		
A				P	C	L	I	N	U	X	O	S			E			N		R		
K		Q				I		O							P			O		M		
U		U			A	N	T	H	U	R	E	S	T	A	O	S		R		I		
L		B				U		O							I			C		N		
U	B	E	R	M	I	X		S					S	O	N	E	B	U	N	T	U	
		S						T	A	I	L	S			T			X				

LINUX CERTIFIED

Part 9

Written by ErikTheUnready

Before we start , here is a new link to test yourself: http://www.penguintutor.com/quiz/index.php

Where we are: https://www.lpi.org/our-certifications/exam-201-objectives (Topic 201: The kernel). In your LPIC study guide, mine is the second edition, this will be chapter 3 (pages 94 – 132). If you are using the free book above, it's chapter 1 (page 15-44).

Welcome back learners, and a nod to those "just interested".

Some distributions make it easier than others to compile a kernel, some not so much. What you need to know is the gist of the thing, meaning almost a generalization. LPIC does not cover a specific distribution, but it wants you to have the tools and knowledge, so you can read the documentation on "distro x" and be able to follow the steps. It is like following a recipe, you are free to change it so you end up with a chocolate cake instead of a vanilla cake, for instance. So let's discuss the recipe. Now, as with any recipe, if you do not have the tools, you probably are going to fail. You will be installing a lot of tools, source, documentation and libraries, so make sure you have enough disk space free. If you have a system you have built on before, it is a good idea to clean up before you build your custom kernel. Keeping with our cake analogy, 'make clean', 'make distclean' and 'make mrproper' are the commands you can run to clean your workspace.

Configuring a kernel takes a lot of reading. Do not attempt if you are not prepared to read a lot of recipes and directions. Information is everywhere, so you need to pay attention.

```
make menuconfig
```

(this requires ncurses) or:

```
make xconfig (QT)
```

or:

```
make gconfig (GTK)
```

Tip: copy your .config-files to a USB thumbdrive before you change anything. This is especially true when you are working with a kernel that is the same version as your current one. Mistakes will be made and accidents do happen (If you do not have a copy VM!).

'Excluded' is the most confusing. Excluded means it is not loaded in this kernel. This does not mean you cannot load it via a module after the fact. So be aware of this when you fire up: make menuconfig. The rest are self explanatory.

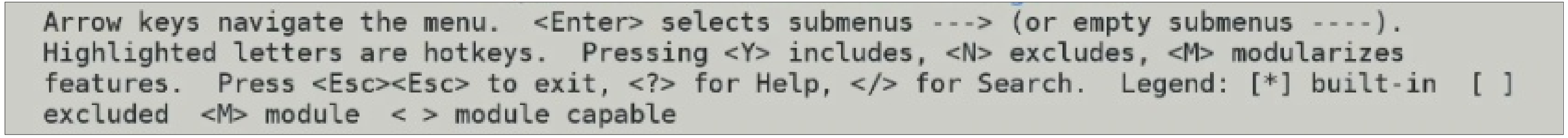
Tip: When you practise this, do it in a VM you already have a copy of.

When you hit save in make menuconfig, it overwrites your .config files!

It is a good idea to do compilation in a safe spot. What I mean by this is that the power must be stable; if not, use a UPS. Also keep children away, and make sure you can't trip over the power cable. Usually one compiles a kernel for systems with low resources to speed them up. The downside is that the lower the resources, the longer it takes to compile. The last time I did this was installing Gentoo on a PII Gateway machine. It took three days to install.

Let's look at our first target: make bzImage

This will make the base. When you run the command, go make some tea and sandwiches; it may take a while. This is where leveraging the power of virtual machines comes in. If you want to do it again, you need to wait for it to complete, but with a bunch of



```
Arrow keys navigate the menu.  <Enter> selects submenus ---> (or empty submenus ----).
Highlighted letters are hotkeys.  Pressing <Y> includes, <N> excludes, <M> modularizes
features.  Press <Esc><Esc> to exit, <?> for Help, </> for Search.  Legend: [*] built-in  [ ]
excluded  <M> module  < > module capable
```

VM's you can do it a few times in a row to get the hang of things. Change it up, remove the parallel port in one and the floppy in another. Go wild, see what it takes to break it. Breaking and fixing something is a great way to learn.

Our next target: make modules

Now that the kernel image has been built, the next step is to look at modules. Do not leave the source folder. When you run this command, you now have the opportunity to go wash those dishes from earlier. You can read the section in the book if you have not done so.

Our next target: make modules_install

This is what we need to map our modules we just made. If you were watching the previous make commands run, you may have noticed that they make files like "parport.o". Those output files now need to be sorted and polished and whatever else is needed for the modules to install. You don't need to know any of this in-depth.

Don your hard hat and let's move the bzImage. For the exam, be aware which trees use gzip and which use bzip. When you move your bzImage to your boot folder, you can rename it to vmlinuz<kernel version.arch> Now we go to the next step, mkinitrd /boot/initrd<kernel version>.img <target> You need to know Ubuntu uses mkinitramfs not mkinitrd.

So now we have a recipe and we understand the flow. If you are doing this in a flavour of Ubuntu or Red Hat, make sure you read that documentation to understand all the quirks. You can still find a copy of CentOS5 and Debian5 to play with.

If you would like to jump ahead, or test your skills at an LPI exam, do a test paper here: https://www.itexams.com/exam/117-201 (These were once valid LPI questions and will ease you into the format).

* The new exam number is 201-400, and 117-201 was the old one.

*The site does require signing up, but sign up with temporary email – not your real email.

***DO NOT learn these questions parrot fashion, as they are probably not real exam questions.**

Let us know how you did; good or bad – it does not matter. Good means you are ready to write and confident in your skills. Bad means you will be learning new stuff!! Yay! There is no downside here. If you do not know why an answer is the way it is, contact us.

Erik has been in IT for 30+ years. He has seen technology come and go. From repairing washing machine sized hard drives with multimeters and oscilloscopes, laying cable, to scaling 3G towers, he's done it.

Written by Erik

Oxygen Not Included

Web page: https://www.klei.com/games/oxygen-not-included

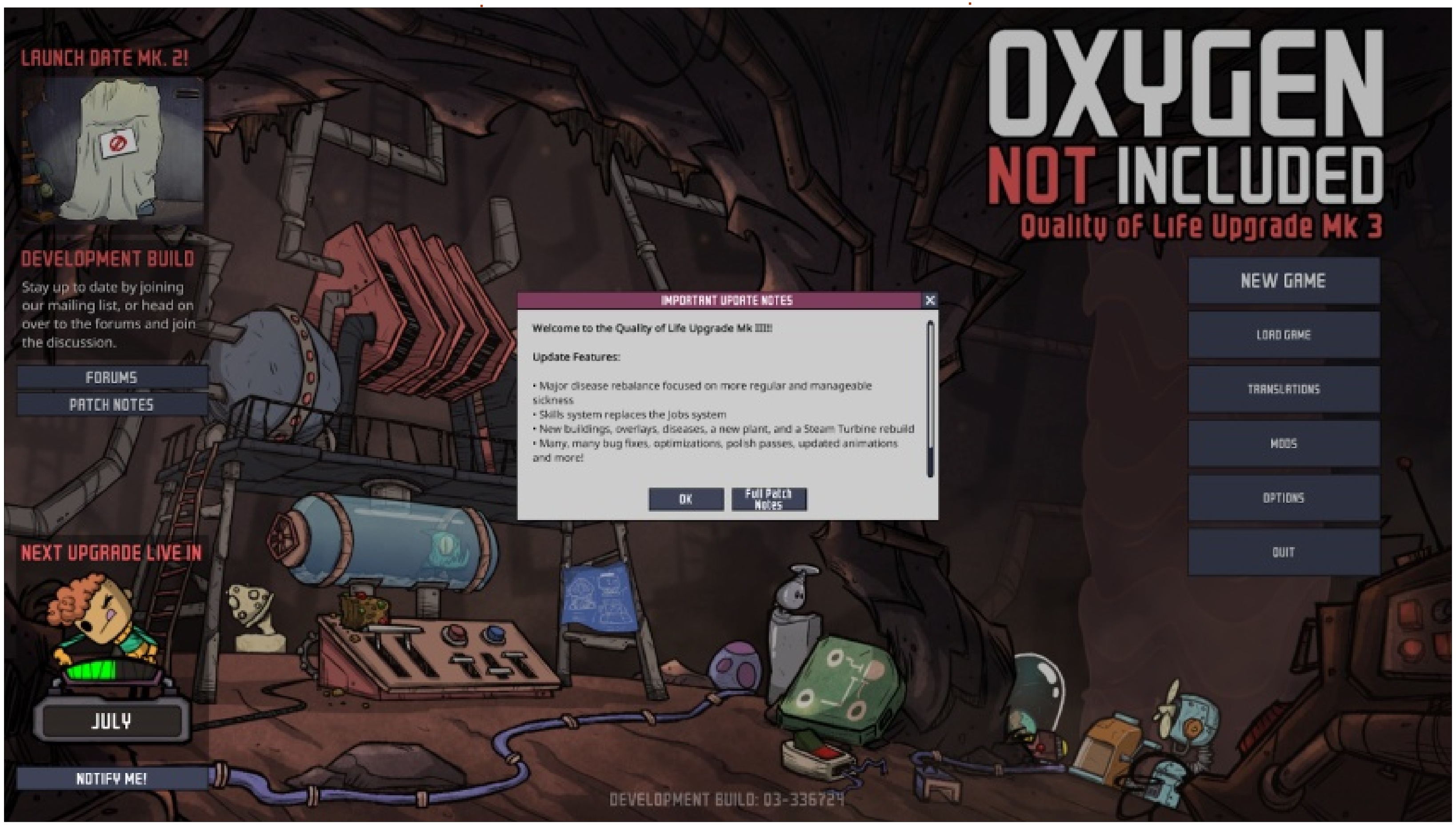
Price: Game is in early access at the time of writing.

Blurb: "*Oxygen Not Included is a space-colony simulation game. Manage your colonists and help them dig, build and maintain a subterranean asteroid base. You'll need water, warmth, food, and oxygen to keep them alive, and even more than that to keep them happy.*"

This is not a fallout shelter-type game, even though the screenshots may look alike. These little dirty settlers will piddle on your floor and sleep in it. No wonder they get sick and infect everyone else. Yes, this game is all about mitigating one crisis after the next, rather than being a straight 'colony sim'. There does not seem to be a goal here, other than survival. Your characters are called "dupes" or duplicates, each has traits which can be positive or negative and will increase over time as they use said skills. The overall look and feel is cartoon-y, but the graphics will grow on you as the animations are really well done. The cartoon exterior hides a very intricate interior, with lots of management systems waiting for you. One of the cool things is that you can name your "dupes" – which helps a lot in remembering who is who and also furthers your immersion into the game. The game is made in Unity, and now that Unity is available natively for Linux, we should see a lot of improvements. Yes, the game has been in early access since 2016.

Graphics.

At first I thought the graphics were horrible. These cartoons look like they are wearing Halloween (as in serial killer) masks. My assumption, at first, was that it was a kid's version of Fallout



Stay up to date by joining our mailing list, or head on over to the forums and join the discussion.

Welcome to the Quality of Life Upgrade Mk III!!

Update Features:

- Major disease rebalance focused on more regular and manageable sickness
- Skills system replaces the Jobs system
- New buildings, overlays, diseases, a new plant, and a Steam Turbine rebuild
- Many, many bug fixes, optimizations, polish passes, updated animations and more!

contents ^

Shelter. Boy, was I wrong. One of my first “dupes” had a flatulence problem, and would go around in a green haze, rather than just some named / numbered ‘stat’. What takes this game to the next level is the animations. Characters will go to the loo and strain can be seen on their faces, followed by relief when they exit. They will fall asleep and drool. They will concentrate with their tongues out. There is a lot more than meets the eye at first! For me the fun part was the different animations when things go wrong, they are hilarious and this just rounds out the cartoon theme.

Sound.

The music is very ambient and the sounds are fitting. There are some tracks that are enjoyable all on their lonesome. As I mentioned, the game is early access, so there is no OST for you yet. Also, since this is a Unity game, there are no sound files to listen to as it is all packed into resource files so you will have to wait for the final game.

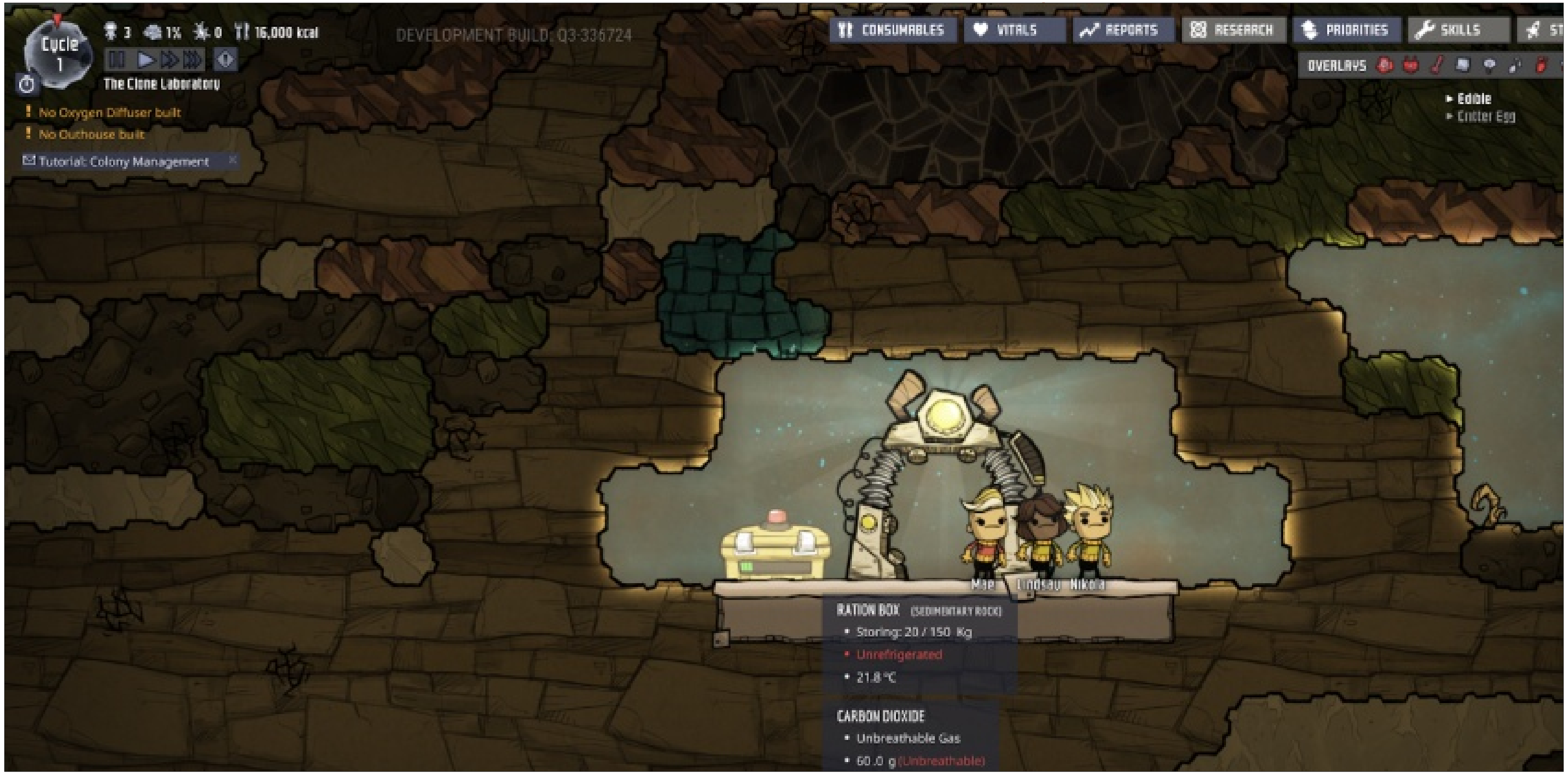
Game-play.

The game is a little confusing at first and there is a steeper than average learning curve, but do not let this put you off. This game will feed your hungry brain cells. You have to plan your days and decide if it is a success on a daily basis. For instance, day one should be getting water and sanitation in. Day two would be barracks (sleeping quarters) and electricity, and so on. At no point does it feel like grind or repetitive, and every day brings its own crisis for you to manage. You can set priorities on everything, right down to the kitchen sink! You can pick and choose your dupes, so, say one wanted to exclude all Gingers, you can do that. (Hey I want dupes with souls!). Keep an eye on that oxygen though!

Overall.

The game feels responsive, even on an Intel potato graphics card. It does tax my core i5 a bit, but not overly so. Memory usage seems average for a Unity game at around two gigabytes. Beware! This game can be a time-sink of note. As with any early access game, expect bugs. Currently the pause button un-pauses, and un-pause pauses. Now and then your dupes may not do what they are told. Still, it is quite enjoyable.

As this game is in early access, there is no rating, but try it anyway. It has a VERY positive rating on steam already. You may find yourself buying this title in early access, even though they want $25...



! No Oxygen Diffuser built
! No Outhouse built
Tutorial: Colony Management

RATION BOX (SEDIMENTARY ROCK)
- Storing: 20 / 150 Kg
- Unrefrigerated
- 21.8 °C

CARBON DIOXIDE
- Unbreathable Gas
- 60.0 g (Unbreathable)

PATRONS

Monthly Patrons

Alex Crabtree
Alex Popescu
Bill Berninghausen
Brian Bogdan
CBinMV
Darren
Dennis Mack
Devin McPherson
Doug Bruce
Elizabeth K. Joseph
Eric Meddleton
George Smith
Henry D Mills
Hugo Sutherland
Jack
Joao Cantinho Lopes
John Andrews
John Malon
John Prigge
Jonathan Pienaar
JT
Kevin O'Brien
Lee Allen
Leo Paesen
Linda P
Mark Shuttleworth
Norman Phillips
Oscar Rivera
Paul Anderson
Paul Readovin
Rob Fitzgerald
Roy Milner
Scott Mack
Sony Varghese
Tom Bell
Tony
Vincent Jobard
Volker Bradley
William von Hagen
Taylor Conroy

Single Donations

2019:
Floyd Smith
Jack Hamm
aram v nathan
Joachim Haupt
Hari Zafiriadis
Glenn Heaton
Adam Gwizdz
George Parker
Linda Prinsen
Frank Dinger
Graig Pearen
Stefano Giancarli
Raymond Meyer
wil van schaik
J.J. van Kampen
James Flanagan
Brian Kelly
Giulio De Chiara
Frits van Leeuwen
Lee Whitehead
Peter Swentzel
Peter Leemann
Zoltan Borsos
Alin Hanghiuc
Mogens Dreier
Ronald Eike

The current site was created thanks to **Lucas Westermann** (Mr. Command & Conquer) who took on the task of completely rebuilding the site, and scripts, from scratch, in his own time.

The Patreon page is to help pay the domain and hosting fees. The yearly target was quickly reached thanks to those listed on this page. The money also helps with the new mailing list that I set up.

Several people have asked for a PayPal (single donation) option, so I've added a button to the right side of the website

A big thank you to all those who've used Patreon and the PayPal button. It's a HUGE help.

patreon

https://www.patreon.com/fullcirclemagazine

Donate with PayPal

https://paypal.me/ronnietucker

donorbox

https://donorbox.org/recurring-monthly-donation

HOW TO CONTRIBUTE

FULL CIRCLE NEEDS YOU!

A magazine isn't a magazine without articles and Full Circle is no exception. We need your opinions, desktops, stories, how-to's, reviews, and anything else you want to tell your fellow *buntu users. Send your articles to: articles@fullcirclemagazine.org

We are always looking for new articles to include in Full Circle. For help and advice please see the **Official Full Circle Style Guide**: http://bit.ly/fcmwriting

Send your **comments** or Linux experiences to: letters@fullcirclemagazine.org
Hardware/software **reviews** should be sent to: reviews@fullcirclemagazine.org
Questions for Q&A should go to: questions@fullcirclemagazine.org
Desktop screens should be emailed to: misc@fullcirclemagazine.org
... or you can visit our **site** via: fullcirclemagazine.org

FCM#153
Deadline:
Sunday 12th Jan. 2020.
Release:
Friday 31st Jan. 2020.

Full Circle Team

Editor - Ronnie Tucker
ronnie@fullcirclemagazine.org

Webmaster - Lucas Westermann
admin@fullcirclemagazine.org

Editing & Proofreading
Mike Kennedy, Gord Campbell, Robert Orsino, Josh Hertel, Bert Jerred, Jim Dyer and Emily Gonyer

Our thanks go to Canonical, the many translation teams around the world and **Thorsten Wilms** for the FCM logo.

For the Full Circle Weekly News:

You can keep up to date with the Weekly News using the RSS feed: http://fullcirclemagazine.org/feed/podcast

Or, if you're out and about, you can get the Weekly News via Stitcher Radio (Android/iOS/web): http://www.stitcher.com/s?fid=85347&refid=stpr

and via TuneIn at: http://tunein.com/radio/Full-Circle-Weekly-News-p855064/

Getting Full Circle Magazine:

EPUB Format - Most editions have a link to the epub file on that issue's download page. If you have any problems with the epub file, email: mobile@fullcirclemagazine.org

Issuu - You can read Full Circle online via Issuu: http://issuu.com/fullcirclemagazine. Please share and rate FCM as it helps to spread the word about FCM and Ubuntu.

FCM PATREON : https://www.patreon.com/fullcirclemagazine