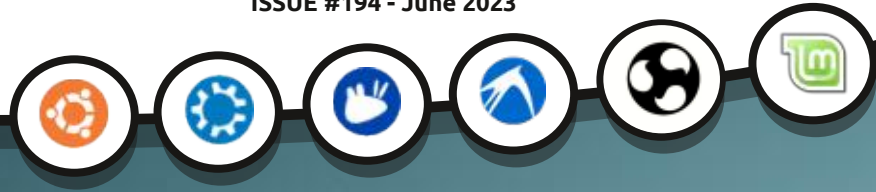




Full Circle

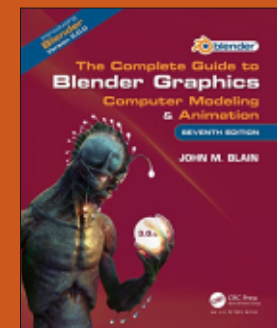
THE INDEPENDENT MAGAZINE FOR THE UBUNTU LINUX COMMUNITY

ISSUE #194 - June 2023



Kubuntu 23.04 REVIEW

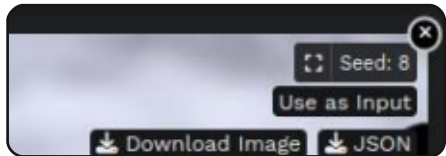
BOOK REVIEW



HowTo



Python p.17



Stable Diffusion p.24



Latex p.26



FreeCAD p.33



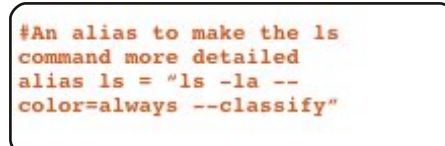
Inkscape p.29

Graphics



Full Circle

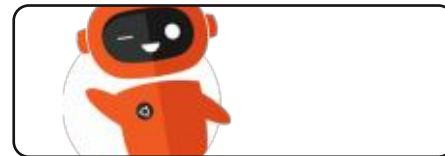
THE INDEPENDENT MAGAZINE FOR THE UBUNTU LINUX COMMUNITY



Command & Conquer p.15



... p.XX



Ubuntu Devices p.XX



The Daily Waddle p.34



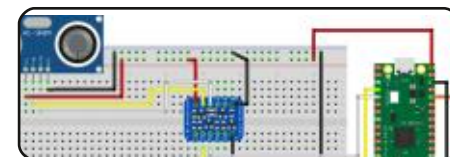
My Story p.43



Letters p.52



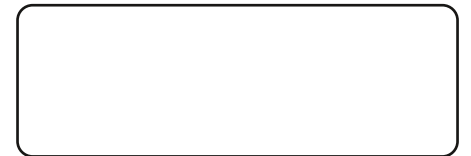
Q&A p.53



Micro This Micro That p.36



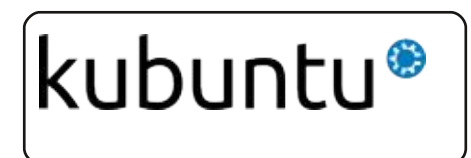
Linux News p.04



Everyday Ubuntu p.XX



Book Review p.50



Review p.46



Ubuntu Games p.57



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.



WELCOME TO THE LATEST ISSUE OF FULL CIRCLE

For your pleasure this month, we continue with; Python, Latex, FreeCAD, Stable Diffusion, some more Micro This Micro That, and Inkscape.

Under review we have Kubuntu 23.04, the rather hefty Blender Graphics book, and the beautiful game APICO. Bee-utiful. See what I did there? It's OK. Never mind.

In other news: I managed to sort out my font issues, so we're back to 'full fat' FCM as Ian (of EPUB fame) called it. **I finally have proper bold fonts again!** Ahhh. All is good in the world again.

For my PC at work I installed Linux Mint on it (rather than Ubuntu which I have on this here home PC) to see what it's like these days. I have to say: I'm thinking about putting it on my home PC too. Mint reminds me of the good old days of Ubuntu with the Synaptic package manager, task bar, and no snaps.

Don't forget: we have a Table of Contents which lists every article from every issue of FCM. Huge thanks to **Paul Romano** for maintaining: <https://goo.gl/tpOKqm> and, if you're looking for some help, advice, or just a chinwag: remember that we have a **Telegram** group: <https://t.me/joinchat/24ec1oMFO1ZjZDc0>. I hope to see you there. Come and say hello.

All the best for 2023!

Ronnie

ronnie@fullcirclemagazine.org



This magazine was created using :



Find Full Circle on:



facebook.com/fullcirclemagazine



twitter.com/#!/fullcirclemag



<https://mastodon.social/@fullcirclemagazine>

Weekly News:



<https://fullcirclemagazine.org/podcasts/index.xml>



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



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

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



RELEASE OF GNU LIBMICRO LIBRARYHTTPD 0.9.77:

29/05/2023

The GNU project published the libmicrohttpd 0.9.77 library, which represents a simple API for embedding the functionality of a HTTP server into applications. Supported platforms include GNU/Linux, FreeBSD, OpenBSD, NetBSD, Solaris, Android, macOS, Win32 and z/OS. The library is distributed under the LGPL 2.1+ license. Once built, the library occupies about 32 KB.

The library supports the HTTP 1.1, TLS protocol, incremental processing of POST requests, basic and digest authentication, IPv6, SHOUTcast, various multiplexing methods (select, poll, epoll) and multithreading models (e.g., flow pool or connection flow can be used). To reduce the overhead costs that occur when the context between the kernel and the user space, the number of system calls in the process is minimized.

<https://www.mail-archive.com/info-gnu@gnu.org/msg03188.html>

AFTER A TEN-YEAR HIATUS - GOLDENDICT 1.5.0:

30/05/2023

GoldenDict 1.5.0, an application for working with dictionary data, supporting various formats of dictionaries and encyclopedias, and is able to display documents in HTML using the WebKit engine has a new release. The project code is written in C++ using the Qt library and distributed under the GPLv3+ license. It supports Windows, Linux and macOS platforms.

You can note a graphical interface based on tabs with tooltips to explain the meaning or translate words allocated or placed in the clipboard. When searching, morphology and the ability to reproduce the correct sound pronunciation of words are available. Among the supported dictionaries formats are StarDict, Babylon, Lingvo, Dictd, AARD,

MDict and SDict, and integration with Wikipedia and MediaWiki-based sites is supported.

The release of GoldenDict 1.5.0 is notable for the fact that it was published 10 years later than the originally scheduled release date (the last stable release was formed in 2010, and version 1.5.0 was planned to be released in September 2013).

<https://github.com/goldendict/goldendict/releases/tag/1.5.0>

CANONICAL PREPARES UBUNTU DESKTOP VERSION CONTAINING ONLY SNAP PACKAGES:

31/05/2023

Developers from Canonical intend to start next year on the delivery of additional Ubuntu Desktop builds based on the Ubuntu Core platform and include only applications designed with Snap packages. Similar to Fedora

Silverblue and Endless OS that use Flatpaks. The first experimental builds of the new version of Ubuntu Desktop is planned for the spring of next year. The delivery of classic Ubuntu Desktop builds with deb-packages will continue without changes.

Ubuntu Core means the delivery of an indivisible monolithic image of the base system, which does not use a breakdown into separate deb packages and uses the mechanism of atomic renewal of the entire system. Ubuntu Core components, including the basic system, the Linux kernel, system add-ons and additional applications, are supplied in snap format and controlled by snap tools. The Snap components are isolated using AppArmor and Seccomp, which creates an additional milestone to protect the system in case of compromised individual applications. The base file system is mounted in read-only mode. Updates of the base environment are delivered in OTA mode (over-the-air), include only changes (delta-updates) and synchronized



with the current LTS release of Ubuntu.

<https://www.omgubuntu.co.uk/2023/05/immutable-all-snap-ubuntu-desktop>

RELEASE OF ARMBIAN 23.05:

01/06/2023

The release of Armbian 23.05, which provides a compact system environment for various single-board computers with processors based on ARM, RISC-V and x86 architectures, including various models of Raspberry Pi, Orange Pi, Banana Pi, Banana Pi, Helios64, pine64, Nanopi and Cubieboard based on Allwinner, rockchip etc, was announced.

Debian and Ubuntu packages are used for builds, but the environment is completely reassembled using its own build system with the inclusion of optimizations to reduce size, increase productivity and use of additional protection mechanisms

The project supports more than 30 Linux kernel builds for different

ARM and ARM64 platforms. To simplify the creation of their system images, a SDK is provided. When you use SSH, an option is provided for the use of two-factor authentication. The release includes the box64 emulator, which allows you to run programs built for processors based on the x86 architecture. "We offer ready-made packages to launch custom environments based on KDE, GNOME, Budgie, Cinnamon, i3-wm, Mate, Xfce and Xmonad."

<https://www.armbian.com/newsflash/armbian-23-05-suni/>

RELEASE ANGIE 1.2.0:

01/06/2023

A release of a high-performance HTTP server and the Angie 1.2.0 multi-protocol proxy server, branched from Nginx, by a group of former project developers who had resigned from the F5 Network, was announced. The original Angie cose is available under a BSD license.

The development is supported by the company "Webserver", formed last fall and received an investment of \$1 million. Among

the co-owners of the company is: Valentin Bartenev (leader of the team that developed the product Nginx Unit), Ivan Poluyanov (former head of frontland developers Rambler and Mail.Ru), Oleg Mamontov (head of the technical support team NGINX Inc) and Ruslan Yermilov (Rus.FreeBSD.org).

<https://github.com/webserver-llc/angie/releases/tag/1.2.0>

RELEASE OF TRUENAS CORE 13.0-U5:

02/06/2023

The release of TrueNAS CORE 13.0-U5, a distribution for the rapid deployment of network storage (NAS, Network-Attached Storage), which continues the development of the FreeNAS project, is out. TrueNAS CORE 13 is based on the FreeBSD 13 codebase, features integrated support for ZFS and the ability to manage through a web interface built using the Django Python framework. FTP, NFS, Samba, AFP, rsync and iSCSI are supported to provide storage access, software RAID (0.1.5) can be used to

authorize clients with LDAP/Active Directory support. The size of iso-image is 1 GB (x86_64). In parallel, the TrueNAS SCALE distribution is developing, using Linux instead of FreeBSD.

<https://www.truenas.com/blog/truenas-13-0-u5-maximizes-quality-and-your-storage-experience/>

RELEASE OF KALI LINUX 2023.2:

02/06/2023

The release of the Kali Linux 2023.2 distribution, based on Debian and designed to test systems for vulnerabilities, auditing, analysis of residual information and detection of the consequences of attacks by attackers, is here. All original developments created within the distribution are distributed under the GPL license and are available through the public Git repository. Several variants of iso-images, 443 MB, 2.8 GB and 3.7 GB have been prepared for download. Images are available for i386, x86_64, ARM (armhf and armel, Raspberry Pi, Banana Pi, ARM Chromebook, Odroid). By default, the Xfce

desktop is offered, but the KDE, GNOME, MATE, LXDE and Enlightenment e17 are optionally supported.

Kali includes one of the most complete sets of tools for computer security professionals, from web-based software testing and wireless network intrusion to data readers from RFID chips. It includes a collection of exploits and more than 300 specialized security checks, such as Aircrack, Maltego, SAINT, Kismet, Blueggbuer, Btcrack, Btscanner, Nmap, p0f. In addition, the distribution includes means to speed up password selection (Multihash CU Brute Forcer) and WPA keys (Pyrit) through the use of CUDA and AMD Stream graphics cards, which allow the use of GPU graphics cards NVIDIA and AMD for computing operations.

<https://www.kali.org/blog/kali-linux-2023-2-release/>

RELEASE OF APACHE NETBEANS 18:

03/06/2023

The Apache Software Foundation introduced the integrated development environment - Apache NetBeans 18, which provides support for programming languages like Java, Java EE, PHP, C/C++, JavaScript and Groovy programming languages. The finished builds are formed for Linux (snap, flatpak), Windows and macOS.

<https://blogs.apache.org/netbeans/entry/announce-apache-netbeans-18-released>

UPDATING SENDMAIL 8.17.2:

03/06/2023

Almost two years after the last update, the release of the Sendmail 8.17.2 SMTP server is available. In addition to bug fixes, the new version has improved support for email addresses with symbols of national alphabets (EAI, Email Address Internationalization). They also added support for new versions of macOS, improved DANES (DNS-based Authentication of Named Entities), to prevent the performance of the MaxMimeHeaderLeng parameter by default is set to 2048/1024.

<https://marc.info/?l=sendmail-announce&m=168578185221545&w=2>

BLINK, HIGH-PERFORMANCE SYSTEM EMULATOR:

05/06/2023

The first major release of the Blink project, which allows you to run statically and dynamically assembled Linux applications in a virtual machine with an emulated processor, is out. With the help of Blink, x86-64 Linux programs can be run in other POSIX-compatible operating systems (macOS, FreeBSD, NetBSD, OpenBSD, Cygwin) and on equipment with other hardware architectures (x86, ARM, RISC-V, MIPS, PowerPC, s390x). The project code is written in C (ANSI C11) and is distributed under the ISC license. Only dependency required is libc (POSIX.1-2017).

To ensure high performance, a JIT compiler is used, converting the original instructions into machine code for the target platform on the fly. Directly run executables in ELF, PE (Portable Executables) and bin (Flat executable) formats collected



DistroWatch.com

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

with standard C-libraries
Cosmopolitan, Glibc and Musl
formats are all supported.

<https://github.com/jart/blink/releases/tag/1.0.0>

RELEASE OF OWNCAST 0.1.0:

05/06/2023

Owncast 0.1.0, a server for streaming video and chatting with the audience, was announced. The server runs on the user's equipment and unlike Twitch, Facebook Live and YouTube Live services, allows you to fully control the broadcasting process and set your chat rules. Management and interaction with users is carried out through the web-interface. The project code is written in Go and distributed under the MIT license.

The new release is notable for the complete rewriting of the frontend code responsible for displaying the web-interface. The new interface is noticeably faster, better adapted for mobile devices, supports the insertion of its Javascript code and provides opportunities for customizing style

and design. In the chat, participants can change colours.

<https://owncast.online/releases/owncast-0.1.0/>

PLANE:

06/06/2023

Plane 0.7, that provides tools for project management, error tracking, work planning, product development support, task building and coordinating their implementation, is available. A platform that can be deployed in its own infrastructure and not dependent on third-party suppliers is developing as an alternative to proprietary systems as JIRA, Linear and Height. The project is still under development and is preparing for the first stable release. The code is written in Python using the Django framework and is distributed under the Apache 2.0 license. PostgreSQL is used as a DBMS, and for a fast storage - Redis. The Web interface is written on TypeScript using the Next.js library.

Plane supports different types of workflows and allows you to

separately track tasks (ToDo), to-do list (backlog), tasks and completed tasks. The system is designed for the use of cascading (waterfall) and flexible (agile) methods of project development. In the cascading model, the development is seen as a continuous flow, consistently undergoing planning, design, implementation, testing, integration and support.

<https://github.com/makeplane/plane/releases/tag/v0.7-dev>

ASAHI LINUX:

06/06/2023

The developers of the Asahi project, aimed at porting Linux to work on Mac computers equipped with ARM chips developed by Apple, prepared a June update of the distribution (560 MB and 3.5 GB) and published a report on the implementation in the support of OpenGL 3.1. Asahi Linux is based on Arch Linux, includes a traditional set of programs and comes with the KDE Plasma desktop. The distribution is built using regular Arch Linux repositories, and all specific changes, such as the kernel,

installer, loader, auxiliary scripts and environment settings, are placed in a separate repository.

To support the GPU AGX chips on Apple M1 and M2, they develop two drivers that work in pairing with each other: DRM driver (Direct Rendering Manager) drmm-asahi for the Linux kernel written in Rust language, and the driver asahi for Mesa, written in the C language. The core-level driver is initially developed taking into account the future support of the Vulkan API, and the software interface for interaction with the user space and is designed with an UAPI view, provided by the new Intel Xe driver. Since Apple M1/M2 chips use its own GPU, which runs on closed firmware and uses quite complex shared data structures, they had to reverse engineer macOS drivers.

<https://asahilinux.org/2023/06/opengl-3-1-on-asahi-linux/>

RELEASE OF OPENSUSE LEAP 15.5:

07/06/2023

After a year of development, the openSUSE Leap 15.5

distribution was released. The release is based on a single set of binary packages with SUSE Linux Enterprise 15 SP 5 with some custom applications from the openSUSE Tumbleweed repository. The use of the same binary packages in SUSE and openSUSE simplifies the transition between distributions, saves resources for building packages, distributing updates and testing, unifies the differences in spec files and allows you to move away from analyzing error messages for different architectures. To download, a universal DVD build, 4 GB (x86_64, aarch64, ppc64le, 390x), a mini image (200 MB) and Live-builds with KDE, GNOME and Xfce (900 MB).

Updates for the openSUSE Leap 15.5 branch will be available until the end of 2024. Initially, version 15.5 was expected to be the latest in the 15.x series, but the developers decided to generate another release of 15.6 next year before the planned transition to the ALP (Adaptable Linux Platform) as the basis of openSUSE and SUSE Leap.

<https://www.opensuse.org/>

POSTMARKETOS 23.06:

07/06/2023

The release of postmarketOS 23.06, a Linux distribution for smartphones, based on Alpine Linux, the standard Musl C-library and a set of BusyBox utilities, has been published. The goal of the project is to provide a Linux distribution for smartphones, independent of the life cycle support of official firmware and not tied to the standard solutions of the main industry players who set the development vector. The builds are prepared for the PINE64 PinePhone, Purism Librem 5 and 29 community-supported devices, including the Samsung Galaxy A3/A5/S4, Xiaomi Mi Note 2/Redmi 2, OnePlus 6, Lenovo A6000, ASUS MeMo Pad 7 and even the Nokia N900. Limited experimental support is provided for more than 300 devices.

The postmarketOS environment is as unified as possible and takes all the device-specific components in separate packages, all other packages are identical to all devices and are based on Alpine Linux packages. In the builds,

where possible, the Linux vanilla kernel is used, otherwise the kernels from firmware prepared by device manufacturers are used. KDE Plasma Mobile, Phosh, GNOME Mobile and Sxmo are available as the main custom shells, but it is possible to install other environments, including MATE and Xfce.

<https://postmarketos.org/blog/2023/06/07/v23.06-release/>

RELEASE OF CINNAMON 5.8:

08/06/2023

After 7 months of development, the Cinnamon 5.8 user environment was released. The Linux Mint distribution community develops a fork of GNOME Shell, Nautilus file manager and Mutter window manager aimed at providing an environment in the classic GNOME 2 style with support for successful interaction elements from GNOME Shell. Cinnamon is based on GNOME components, but these components are supplied as a periodically synchronized fork not associated with GNOME external dependencies. The new Cinnamon release will be proposed

in the Linux Mint 21.2 distribution, which is scheduled to be released at the end of June.

<http://cinnamon.linuxmint.com/>

PROJECT FCIV.NET DEVELOPS 3D VERSION OF THE STRATEGY GAME

FREECIV:

09/06/2023

The Fciv.net project is developing a 3D version of the turn-based strategy game Freeciv, where the gameplay resembles the Civilization games. The game can be run in a web browser that supports HTML5 and WebGL 2. You can play in multiplayer mode and individual rivalry with bots. Fciv.net continues to develop the Freeciv-web project codebase and is characterized by the use of WebGL and the Three.js 3D engine, as well as some advanced capabilities such as the ChatGPT-based AI assistant. The project code is distributed under the AGPLv3 license.

<https://github.com/fciv-net/fciv-net>

PUZZLEFS FILE SYSTEM FOR LINUX KERNEL:

09/06/2023

Cisco has offered a new PuzzleFS file system implemented as a module for the Linux kernel written in the Rust language. The FS is designed to be used for the placement of isolated containers and continues to develop the ideas proposed in the 'Atomfs' FS. The implementation is still in the prototype stage and is open under the Apache 2.0 and MIT licenses.

The project is aimed at circumventing the restrictions that occur when using container images in the OCI (Open Container Initiative). PuzzleFS solves such problems as effective storage of duplicate data, the ability to directly mount, repetitive image builds and memory safety.

<https://lore.kernel.org/rust-for-linux/20230609063118.24852-1-amiculas@cisco.com>

RELEASE OF DEBIAN 12 "BOOKWORM":

10/06/2023

After almost two years of development, Debian GNU/Linux 12.0 (Bookworm) was released, available for nine officially supported architectures: Intel IA-32/x86 (i686), AMD64/x86-64, ARM EABI (armel), ARM64, ARMv7 (armhf), mipsel, mips64el, PowerPC 64 (ppc64el) and IBM System. Updates for Debian 12 will be available for 5 years.

Installation images are available (the publication of images is delayed) installation images that can be downloaded by HTTP, jigdo or BitTorrent. For the amd64 and i386 architectures, they developed a LiveUSB, available with GNOME, KDE, LXDE, Xfce, Cinnamon and MATE, as well as a multi-architecture DVD that combines packages for the amd64 platform with additional packages for the i386 architecture.

The repository features 64,419 binary packages, which is 4,868 more packages than proposed in Debian 11.

<https://www.debian.org/News/2023/20230610>

RELEASE OF EASYOS 5.4:

11/06/2023

Barry Kauler, founder of the Puppy Linux project, has published the EasyOS 5.4 distribution, combining Puppy Linux technology with container isolation to run system components. Distribution management is carried out through a set of graphic configurators developed by the project. Image size 860 MB.

<https://bkhome.org/news/202306/easyos-kirkstone-series-version-54-released.html>

THE FREEBSD PROJECT IS 30 YEARS OLD:

19/05/2023

The FreeBSD project celebrates its Thirtieth anniversary. The birthday of the project is considered on June 19, 1993. On this day, an unofficial set of patches to 386BSD, a 4.3BSD fork was created, where it was planned

to develop a fast, stable and reliable OS for systems with i386 processors. When choosing the name of the project, FreeBSD, BSDFree86 and Free86BSD were considered as options. As a result, it was given to the FreeBSD version, which was proposed by David Greenman. In November 1993, based on 4.3BSD-Lite 'Net/2' and 386BSD 0.1, the first release of FreeBSD 1.0 was released.

<https://www.freebsd.foundation.org/national-freebsd-day/>

BCACHEFS IN THE LINUX KERNEL:

19/05/2023

Kent Overstreet, author of the Linux kernel caching systems of block devices on BCache SSD-drives, in his speech at the LSFMM 2023 (Linux Storage, Filesystem, Memory Management & BPF Summit) summed up the results of work on the transfer of the Bcachefs file system to the main composition of the Linux core and talked about plans for the further development of this FS. In May, an updated set of patches with the

implementation of the Bcachefs FS was proposed for reviewing and inclusion in the main composition of the Linux kernel. FS Bcachefs has been developing for about 10 years. The readiness to review the implementation of Bcachefs before the inclusion in the core was announced at the end of 2020 and in the current version of the patches took into account the comments and shortcomings identified during the previous review.

The purpose of the development of Bcachefs is to achieve the level of XFS in performance, reliability and scalability, while providing additional features inherent in Btrfs and ZFS, such as the inclusion in the partition of several devices, multi-layer layouts, replication (RAID 1/10), caching, transparent data compression (LZ4, gzip and ZSTD modes), status cuts, and the ability to record the amount of the data (used by ChaCha20 and Poly1305). In terms of performance, Bcachefs ahead of Btrfs and other FS based on the Copy-on-Write mechanism, and demonstrates a working speed close to Ext4 and XFS.

<https://news.ycombinator.com/item?id=36366002>

SysLinuxOS 12:

20/05/2023

The SysLinuxOS 12 distribution has been published, built on Debian 12 and aimed at providing a bootable live-environment optimized for system integrators and administrators. To download, builds with GNOME (4.8 GB) and MATE (4.6 GB) DE's have been prepared.

It includes a selection of pre-installed applications for monitoring and diagnosis of the network, tunneling traffic, launching a VPN, remote access, intrusion detection, security checks, simulating networking and traffic analysis, which can be used immediately after downloading the distribution to a USB drive.

Applications included like: Wireshark, Ettercap, Ettercap, PackETH, Pktsender, Putty, Nmap, GNS3, Lssid, Packet Tracer 8.2.1, Wine, Virtualbox 7.0.2, Teamviewer, Anydesk, Remmina, Zoom, Skype, Pktsender, Sparrow-Wifi, Angry Ipner, Fast-cli,

Speed, ,, Firewalk, Firejails, Cacti, Icinga, Monit, Nagios4, Fail2ban, Wireguard, OpenVPN, Firefox, Chrome, Chromium, Microsoft Edge and Tor Browser.

Unlike Debian 12, in SysLinuxOS, the GRUB downloader returned other installed operating systems via the os-prober package. The Linux kernel has been updated to version 6.3.8. Implemented a more understandable name for network interfaces (eth0, wlan0, etc.). The environment works in Live mode, but also supports installation on a disk with the help of the Calamares installer.

<https://syslinuxos.com/syslinuxos-12-for-system-integrators/>

RELEASE OF THE SDL 2.28.0:

21/05/2023

After seven months of development, the release of SDL 2.28.0 (Simple DirectMedia Layer) library is out. It is aimed at simplifying the writing of games and multimedia applications. The SDL library provides such tools as hardware-reusable 2D and 3D

output, input processing, sound reproduction, 3D output via OpenGL/Vulkan and many other related operations. The library is written in the C language and distributed under the Zlib license. To use the capabilities of SDL in projects in a different programming language, the necessary bindings are provided.

The release of SDL 2.28.0 mainly offers bug fixes, the innovations note the addition of functions `SDL_HasWindowSurface()` and `SDL_DestroyWindowSurface()` to switch between the API `SDL_DL_Renderer` and `SDL_Surface`, the new event `SDL_DISPLAYEVENT_MOVED`, the main-screen `S.Leo_SCREEN_KEYBOARD` to control the display of the on-screen keyboard.

<https://discourse.libsdl.org/t/announcing-sdl-2-28-0/44341>

SUSE LINUX ENTERPRISE 15 SP5:

21/05/2023

After a year of development, SUSE has released - SUSE Linux

Enterprise 15 SP5. Based on the SUSE Linux Enterprise platform, products such as SUSE Linux Enterprise Server, SUSE Linux Enterprise Desktop, SUSE Manager and SUSE Enterprise High Performance Computing have been formed. The distribution can be downloaded and used for free, but access to updates and fixes is limited to a 60-day trial period. The release is available in builds for aarch64, ppc64le, s390x and x86_64 architectures.

SUSE Linux Enterprise 15 SP5 supports full binary package compatibility with the openSUSE Leap 15.5 distribution community, which was released two weeks ago. The high level of compatibility is achieved through the use of a set of binary packages in openSUSE, instead of reassembling source packages. It is assumed that users can first form and test a working solution using openSUSE, and then switch to the commercial version of SUSE Linux with full support, SLA, certification, long release of updates and advanced means for mass implementation without unnecessary complications

<https://www.suse.com/news/SUSE-Advances-Capabilities-to-its-Comprehensive-Software-Stack-to-Help-Customers-Accelerate-Digital-Transformation/>

CENTOS STREAM WILL BE THE ONLY PUBLIC SOURCE OF RHEL PACKAGES:

21/05/2023

Red Hat has announced a change in the approach to publishing the source code packages of Red Hat Enterprise Linux and discontinuing the publication of packages in the Git repository git.centos.org. The only publicly available source of the RHEL packages will now be the CentOS Stream repository. For Red Hat's customers and partners, they can download the package code corresponding to RHEL releases through the company's client portal, which requires an account to access.

For CentOS and CentOS Stream projects, the new source distribution model will not lead to noticeable changes, but third-party distributions, such as AlmaLinux, Rocky Linux, Oracle and EuroLinux,

created by reassembling RHEL packages, will have to significantly revise its development processes or bypasses to access the package code from RHEL releases.

<https://www.redhat.com/en/blog/furthering-evolution-centos-stream>

PEERTUBE 5.2:

22/05/2023

The decentralized platform for video hosting and video broadcasting, PeerTube 5.2 was announced. PeerTube offers an independent alternative to YouTube, Dailymotion and Vimeo, an independent of individual providers, using a content distribution network based on P2P communications and linking visitors browsers. The project is distributed under the AGPLv3 license.

"This version comes with a small technical challenge that we're proud to have overcome! This new feature won't be as visible as a graphical change, but it will make hosting a PeerTube platform easier, more resilient and cheaper."

<https://joinpeertube.org/news/release-5.2>

INTEL AND BLOCKADE LABS HAVE RELEASED A MODEL FOR THE SYNTHESIS OF 3D IMAGES:

22/05/2023

Intel and Blockade Labs have jointly developed a LDM3D (Latent Diffusion Model for 3D) machine learning model to generate images and related depth maps based on text descriptions in natural language. The development resembles a Stable Diffusion image synthesis system, but allows you to form three-dimensional visual content, such as spherical panoramic images that can be viewed in 360-degree mode. On the practical side, the model can be used in games and virtual reality systems for the interactive formation of three-dimensional environments.

To download, a ready-made model for machine learning systems is available, which can be used with PyTorch and code designed to generate images using

models from the Stable Diffusion project. The model is distributed under the Permissive Creative ML OpenRAIL-M license, which allows use for commercial purposes. Dissemination under an open license allows researchers and interested developers to improve the model according to their needs and optimize it for highly specialized applications.

<https://www.intc.com/news-events/press-releases/detail/1630/intel-labs-introduces-ai-diffusion-model-generates>

RELEASE TOR BROWSER 12.5:

23/05/2023

After 8 months of development, a major release of the specialized Tor Browser 12.5 is presented. It continues to develop functionality based on Firefox 102 ESR. The browser is focused on ensuring anonymity, security and privacy, and all traffic is redirected only through the Tor network. Tor Browser builds are prepared for Linux, Android, Windows and macOS.

To provide additional protection, Tor Browser includes a HTTPS Only setting that allows you to use traffic encryption on all sites where possible. To reduce the threat from attacks using JavaScript and blocking the plugins by default, the NoScript add-on is available. Fteproxy and obfs4proxy are used to combat traffic locking and inspection.

<https://blog.torproject.org/new-release-tor-browser-125/>

RELEASE OF PROXMOX VE 8.0:

23/05/2023

Proxmox Virtual Environment 8.0, a specialized Linux distribution based on Debian GNU/Linux, aimed at deploying and maintaining virtual servers using LXC and KVM, and able to replace products such as VMware vSphere, Microsoft Hyper-V and Citrix Hypervisor, is out. Installation size is 1.1 GB.

Proxmox VE provides means to deploy virtual servers of industrial grade with a web-based interface designed to manage hundreds or

even thousands of virtual machines. The distribution has built-in tools for backing up virtual environments and available out-of-box support for clustering, including the ability to migrate virtual environments from one node to another without stopping the work. Among the features of the web-interface: support for secure VNC consoles; control of access to all available objects (VM, storage, nodes, etc.) role-based; support for various authentication mechanisms (MS ADS, LDAP, Linux PAM, Proxmox VE authentication).

<https://forum.proxmox.com/threads/proxmox-ve-8-0-released.129320/>

RELEASE OF FLOWBLADE 2.10:

23/05/2023

After two and a half years of development, the release of a multi-track system of non-linear video editing, Flowblade 2.10, is out. It allows you to compile films and videos from a set of individual videos, sound files and images. The editor provides a means for trimming clips with accuracy down

to individual frames, processing them with filters and multi-level layout of images for embedding in the video.

The project code is written in Python and distributed under the GPLv3 license. The builds are prepared in Flatpak format. The MLT framework is used for video editing. The FFmpeg library is used to process various video, sound and images formats. The interface was built using PyGTK. For mathematical calculations, the NumPy library is used. PIL is used for image processing. You can use plugins with the implementation of video effects from the collection of Frei0r, as well as sound plugins LADSPA and G'MIC image filters.

<http://jliljebl.github.io/flowblade/>

OPENSNIATCH 1.6.0:

24/05/2023

The release of OpenSnitch 1.6, developing an open analogue of the proprietary dynamic firewall, Little Snitch, has been published. The application allows you to interactively monitor your network activity of user applications and

block unwanted network traffic. The project code is written in the Go language (GUI in Python and PyQt5) and is distributed under the GPLv3 license. Packages in rpm and deb formats have been prepared for installation.

OpenSnitch displays a dialogue where you decide on the continuation of the network operation or blocking network activity. The program allows you to set access rules that allow you to consider applications, users, target hosts and network ports. Access can be provided both on a permanent basis and limited only to the current process or temporary.

OpenSnitch also allows you to configure the batch filter through the graphical interface, keep network activity statistics and log it too.

<https://github.com/evilsocket/opensnitch/releases/tag/v1.6.0>

BROWSERBOX PRO:

25/05/2023

Dosyago has opened the source code of the BrowserBox Pro platform, which implements a remote browser isolation system (RBI, Remote Browser Isolation) under the GPLv3 license. BrowserBox Pro provides an additional layer of client systems protection when navigating the Web, implemented through the removal of web-page processing functions to an isolated environment that can be performed on another host or in the cloud. The platform code is written in JavaScript using the Node.js platform.

BrowserBox Pro provides a kind of proxy for viewing sites, taking on the function of drawing web pages. For this, Chrome browser is used, running in a separate isolated environment in headless mode. The user is provided with a web-interface that reproduces the browser interface. The code (JavaScript, HTML and CSS) of sites opened through this web-interface is processed in an isolated environment, and the client only receives by the already drawn

content.

<https://github.com/dosyago/BrowserBoxPro>

THE

VIRTUALBOX NETWORKING PRIMER

Connecting and Configuring
Virtual Machines



Robin Catling

The VirtualBox Networking Primer

Connecting and Configuring Virtual Machines

The VirtualBox Networking Primer is a no-nonsense guide for the VirtualBox user taking their next steps into virtual networks.

While Oracle VM VirtualBox is a great free tool, the real power of virtualisation comes when you start connecting virtual machines to each other and to the wider world. Software development, sales, education and training are just some of the areas in which network access to virtual machines offers endless opportunities. But the world of computer networks is filled with complex technical jargon.

Complete with principles, practice, examples and glossary, The VirtualBox Networking Primer takes the frustration and confusion out of connecting real-world projects.

Author: **Robin Catling**

Publisher: Proactivity Press

ISBN13 : 9781916119482

Amazon US link:

[https://www.amazon.com/dp/1916119484?
ref=pe_3052080_397514860](https://www.amazon.com/dp/1916119484?ref=pe_3052080_397514860)

Amazon UK link:

[https://www.amazon.co.uk/VirtualBox-Networking-Primer-
Connecting-Configuring/dp/1916119484/ref=sr_1_1?
dchild=1&keywords=virtualbox+networking+primer&qid=1600253
699&s=books&sr=1-1](https://www.amazon.co.uk/VirtualBox-Networking-Primer-Connecting-Configuring/dp/1916119484/ref=sr_1_1?dchild=1&keywords=virtualbox+networking+primer&qid=1600253699&s=books&sr=1-1)

Kobo:

[https://www.kobo.com/us/en/ebook/the-virtualbox-networking-
primer](https://www.kobo.com/us/en/ebook/the-virtualbox-networking-primer)



COMMAND & CONQUER

Written by Erik

Web Services

Though I try to keep our CnC as n00b friendly as possible, I may not explain as well as I should, so if you get stuck, let us know on misc@fullcirclemagazine.org

Apache is free and open source. It is simple to set up. On the official Ubuntu page, you can find the instructions: <https://ubuntu.com/tutorials/install-and-configure-apache#2-installing-apache>

I suggest you open the page and pop your terminal next to it.

Once completed, move to the next page: <https://ubuntu.com/tutorials/install-and-configure-apache#3-creating-your-own-website>

```
www: bash — Konsole
New Tab Split View
ed@neon9: /var/www$ ls
html
ed@neon9: /var/www$ sudo mkdir gci
[sudo] password for ed:
ed@neon9: /var/www$ ls
gci html
ed@neon9: /var/www$
```

```
~: bash — Konsole
New Tab Split View Copy Paste Find
ed@neon9:~$ sudo apt install apache2
[sudo] password for ed:
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
Starting pkgProblemResolver with broken count: 0
Starting 2 pkgProblemResolver with broken count: 0
Done
The following additional packages will be installed:
  apache2-bin apache2-data apache2-utils libapr1 libaprutil1 libaprutil1-dbd-sqlite3 libaprutil1-ldap
  liblua5.3-0
Suggested packages:
  apache2-doc apache2-suexec-pristine | apache2-suexec-custom
The following NEW packages will be installed:
  apache2 apache2-bin apache2-data apache2-utils libapr1 libaprutil1 libaprutil1-dbd-sqlite3
  libaprutil1-ldap liblua5.3-0
0 upgraded, 9 newly installed, 0 to remove and 558 not upgraded.
Need to get 2,457 kB of archives.
After this operation, 8,216 kB of additional disk space will be used.
Do you want to continue? [Y/n] y
Get:1 http://archive.ubuntu.com/ubuntu jammy/main amd64 liblua5.3-0 amd64 5.3.6-1build1 [140 kB]
Get:2 http://security.ubuntu.com/ubuntu/ubuntu jammy-security/main amd64 libapr1 amd64 1.7.0-Sububuntu0.22.04.1 [108 kB]
Get:3 http://archive.ubuntu.com/ubuntu jammy-updates/main amd64 apache2-bin amd64 2.4.52-1ubuntu4.5 [1,345 kB]
Get:4 http://security.ubuntu.com/ubuntu/ubuntu jammy-security/main amd64 libaprutil1 amd64 1.6.1-Sububuntu4.22.04.1 [92 .6 kB]
Get:5 http://security.ubuntu.com/ubuntu/ubuntu jammy-security/main amd64 libaprutil1-dbd-sqlite3 amd64 1.6.1-Sububuntu4 .22.04.1 [11.3 kB]
```

You can make the folder name meaningful or keep that gci as per the instructions. For simplicity, I will do that too.

Please continue: <https://ubuntu.com/tutorials/install-and-configure-apache#4-setting-up-the-virtualhost-configuration-file>

Just follow along and make the changes. This is not the part we will be discussing as it is all laid out very plainly.

```
GNU nano 6.2
<VirtualHost *:80>
# The ServerName directive sets the request
# the server uses to identify itself. This i
# redirection URLs. In the context of virtua
# specifies what hostname must appear in the
# match this virtual host. For the default v
# value is not decisive as it is used as a l
# However, you must set it for any further v
#ServerName www.example.com

ServerAdmin misc@fullcirclemagazine.org
DocumentRoot /var/www/gci
ServerName gci.example.com

# Available loglevels: trace8, ..., trace1,
# error, crit, alert, emerg.
```

As you can see, I am shadowing to make sure the tutorial still works.

And of course it did not work. You see it's DNS, it's always DNS.... XD

So to make the final step work, you will need to add an entry to your hosts file.

So:

`sudo nano /etc/hosts -and add an address`

I made mine 127.0.0.5

```
--: sudo nano — Konsole
New Tab Split View
GNU nano 6.2 /etc/
# Standard host addresses
127.0.0.1 localhost
::1 localhost ip6-localhost ip6-loopback
ff02::1 ip6-allnodes
ff02::2 ip6-allrouters
# This host address
127.0.1.1 neon9
127.0.0.5 gci.example.com
```

Save it and type

`gci.example.com`

into your browser and voila!

We also have Nginx, as an alternative to Apache.

Before we go on to installing Nginx, let's talk about Apache. I looked this up in Wikipedia, so not everything may be 100%. The

Apache HTTP Server was created by Robert McCool in 1995 and has fallen under the Apache Software Foundation since 1999.

Because of this popularity, Apache benefits from great documentation and integrated support from other software projects. (It was the most popular server on the internet from at least 1996 through 2016.) That said, it is by no means perfect, so in 2002, Igor Sysoev began work on Nginx, as Apache could not handle more than ten thousand concurrent connections. Nginx was released to the public in 2004, and was able to deliver on serving more than ten thousand concurrent connections. Nginx has since become more popular than Apache, due to its small size and its ability to scale on minimal hardware. We will talk more on this later in this CnC series, I just want you to know the basic difference. (We will get much more in-depth when we talk security) You will also hear people talking about static and dynamic content and file-based and URI-based, but this is beyond our n00b CnC for now.

For those of you who just want to play, you have a web server on

your Ubuntu machine now. Now's a good time to learn basic HTML & CSS.

Let's move over to Nginx.

OK let's not, I will go over my space in the magazine, but we will continue in the next issue.

[Editors note: see all this blank space? Don't believe him. He's slacking. Nah, just kidding.]



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.



Last month, we looked at playing some tricks on the ttk TNotebook widget with the style manipulation Python module that I wrote called “mystyles_dark.py”. I also promised that I would focus more on that module to provide some insight on how it all works.

Since then, I’ve modified the theme file so it isn’t quite as dark, and to do some other things that the original couldn’t. The new file is called “mystyles_notsodark.py”, and that’s what we’ll concentrate on in this month’s article. The files will be available in my repository, which I will share with you at the end of the article.

The original code was the starting point of my notsodark Tcl theme. I use the styling module as my testbed to make sure that I could make things work using Python, since my knowledge of Tcl/Tk was and still is limited. I looked at almost every Tcl theme I could, including the source for the four Linux “default” themes alt, clam, classic and of course default. I have

to admit, it took me a while to figure out how to convert much of the Tcl code into Python. I had various copies of my beginning Tcl books open at the same time as I was trying to code the Python file and I had many false starts, but I eventually figured it all out. Eventually, I got the color schemes all worked out to something that I liked.

That’s enough history, now let’s get started.

The first thing we’ll look at is the styling module, “mystyles_notsodark.py”. The imports section is where I always start when examining a python module, so that’s where we’ll begin today.

```
import sys
import shared
import tkinter as tk
import os.path
```

Since we are making a “standalone” module, we need to import sys and tkinter. The use of os.path is needed since we are going to make the module operating system agnostic (able to run on Windows, Linux, and Mac), and we want to be able to get to the Assets folder no matter where the user has installed our module; we can use the os.path.join method to concatenate the correct directory separator (“/” or “\”) in the correct places. The other thing we need to import is an empty file called shared.py. We normally use this to enable various modules of a

program to communicate between each other. However, there is a secondary benefit of this, which will become clear in a little bit.

We put the next three lines just after the import section so that the values are, by scope, implicitly global. This means that they are available to any functions with the module as long as they aren’t changed within any of the functions. We have to be careful of this (see below).

We can use the location variable to provide the proper base path when we call the os.path.join statements later on.

Next, we need to do some definitions of color variables that we will need in our various

```
_script = sys.argv[0]
location = os.path.dirname(_script)
version = "2.05.9 "
```

The sys.argv[0] call returns the full path and name of the python script. The os.path.dirname(_script) call returns just the path. On my machine, the returned values are:

```
_script: '/home/greg/Desktop/Experiments/styleModules/demo1/mystyles_notsodark.py'
location: '/home/greg/Desktop/Experiments/styleModules/demo1'
```

functions. I'll show only a few of the definitions, but you can look

```
_bgcolor = "#919191"
_fgcolor = "black"
_activebgcolor = "gray66"
_activefgcolor = "gray5"
_troughcolor = "gray43"
_barcolor = "gray82"
_darkcolor = _bgcolor
_lightcolor = _fgcolor
_disabledcolor = "gray42"
_disabledfgcolor = "gray3"
_fieldbgcolor = "gray20"
_fieldfgcolor = "gray79"
_bordercolor = "dimgray"
```

into the actual module.

`_bgcolor` stands for background color, `_fgcolor` is the foreground color.

Now that we have all of our global definitions made, we can start with our first function (top right).

The `create_styles` function takes just one parameter, an instance of `ttk.Style()`. Since I like to save as many keystrokes as I can, I named that instance `sty`.

The first `configure` statement creates a set of color definitions that are duplicated from the global definitions. These, however, are supposed to be used for the root style of the theme, which sets the

color sets for every `ttk` widget used in the Toplevel form. It doesn't seem to actually work in Python like this and the actual colors that can be used are those from the global definition.

The next thing is to declare a map for the root style that controls how the widget colors will react to the various states like disabled, active, in focus, not in focus, and so on. Each item must be a list of tuples containing the state followed by the color, even if there

```
def create_styles(sty):
    # =====
    # Apply a "generic" Toplevel style
    # =====
    sty.configure(
        ".",
        bgcolor=_bgcolor,
        fgcolor=_fgcolor,
        activebgcolor=_activebgcolor,
        activefgcolor=_activefgcolor,
        troughcolor=_troughcolor,
        barcolor=_barcolor,
        ...
```

is only one state that you are defining (bottom right).

Now we can start defining the styles for each widget (middle). I'll show you only a few choice

```
# =====
# Style for ALL TButton widgets
# =====
sty.map(
    "TButton",
    background=[("disabled", "#d9d9d9"), ("active", _activebgcolor)],
    foreground=[("disabled", _disabledfgcolor), ("active", _activefgcolor)],
)
sty.configure(
    "TButton",
    foreground=_fgcolor,
    background=_bgcolor,
    padding=[4, 4, 4, 4],
    font="Ubuntu 12 bold",
)
```

```
sty.map(
    ".",
    background=[("disabled", _disabledcolor), ("active", _activebgcolor)],
    foreground=[("disabled", _disabledfgcolor)],
    selectbackground=[("!focus", _selectbackground)],
    selectforeground=[("!focus", "white")],
    embossed=[("disabled", 1)],
)
```

definitions to give you an idea how it's done. Here's the style for the Tbutton.

As before, there can be a section for the map of the state to color, and then a section for the "normal" look. If you want to create a style for all the widgets of a particular class, as in the above snippet, you must use the default style name. That almost always begins with a "T".

When we want to set the attributes that we can't normally set, we need to use the `style.configure` method. This allows us to set things like background and foreground colors, the padding (a list containing up to four integers for left, top, right and bottom in that order), font and so on.

The TButton also has a special style built in called Toolbutton. This, as well, can be overridden by using the same configure and map settings (top right).

As always, when using images, we have to keep a "temporary" copy of the image object, so Python's garbage collection doesn't happily delete it. Now you

```
sty.configure (
    "Toolbutton",
    anchor="center",
    background=_bgcolor,
    foreground=_fgcolor,
    relief="flat",
    highlightcolor=_bordercolor,
    shiftrelief=2,
    highlightthickness=2,
    padding=2,
    font="Veranda 12 bold",
)
sty.map(
    "Toolbutton",
    background=[
        ("disabled", _bgcolor),
        ("active", "gray49"),
        ("selected", _tvwindow),
    ],
    foreground=[("disabled", _disabledfgcolor), ("active", _activefgcolor)],
)
```

It took me a while to figure out how to get the whole custom image thing to work. First, we need to define the images we want to use.

```
# =====
# Style for ALL TCheckbutton
# =====
shared.con_image = tk.PhotoImage(
    file=os.path.join(location, "Assets", "chk24x16.png")
)
shared.coff_image = tk.PhotoImage(
    file=os.path.join(location, "Assets", "unchk24x16.png")
)
shared.cdis_image = tk.PhotoImage(
    file=os.path.join(location, "Assets", "blank24x16.png")
)
```

can see how we use the `os.path.join` function to define the location of the image we want and the need to use the shared.py file.

Next, we need to create a custom element that holds the

information on which image is used for which state. This, of course, will change for different widgets and for various element parts. You can create images that will be used to control the shape of the widget and so on (next page, top right).

Finally, we have to define the layout. This is the REALLY difficult part. Typically, a widget consists of one or more elements. For the TCheckbutton, there is a border, a padding, a focus and a label. This varies by widget class (next page,

```

sty.map(
    "TCheckbutton",
    background=[
        ("disabled", _bgcolor),
        ("pressed", _activebgcolor),
        ("active", _activebgcolor),
        ("hover", _activebgcolor),
        # ("selected", tvwindow),
    ],
    foreground=[
        ("disabled", _disabledfgcolor),
        ("pressed", _activefgcolor),
        ("active", _activefgcolor),
        ("hover", _activefgcolor),
    ],
)

```

```

sty.element_create(
    "custom.CBindicator",
    "image",
    shared.coff_image,
    ("selected", shared.con_image),
    ("disabled", shared.cdis_image),
)

```

Then we can handle the basic widget configuration of colors and so on.

```

sty.configure(
    "TCheckbutton",
    background=_bgcolor,
    foreground=_fgcolor,
    indicatormargin=[6, 6, 6, 6],
    padding=[6, 6, 6, 6],
    font="Ubuntu 12 bold",
)

```

bottom right).

Finally, we can provide the map for the colors for each state (top left).

The styling module also provides a custom TRadiobutton definition.

I won't go through all the various ttk widgets that the styling module handles. I tried to deal with every standard ttk widget available.

In the process of creating the actual Tcl theme, I added a function that looked important, but I didn't really know what it did.

```

sty.layout(
    "TCheckbutton",
    [
        (
            "Checkbutton.padding",
            {
                "sticky": "nsw",
                "children": [
                    ("custom.CBindicator", {"side": "left", "sticky": ""}),
                    (
                        "Checkbutton.focus",
                        {
                            "side": "left",
                            "sticky": "e",
                            "children": [("Checkbutton.label", {"sticky": "nse"})],
                        },
                    ),
                ],
            },
        ),
    ],
)

```

HOWTO - PYTHON

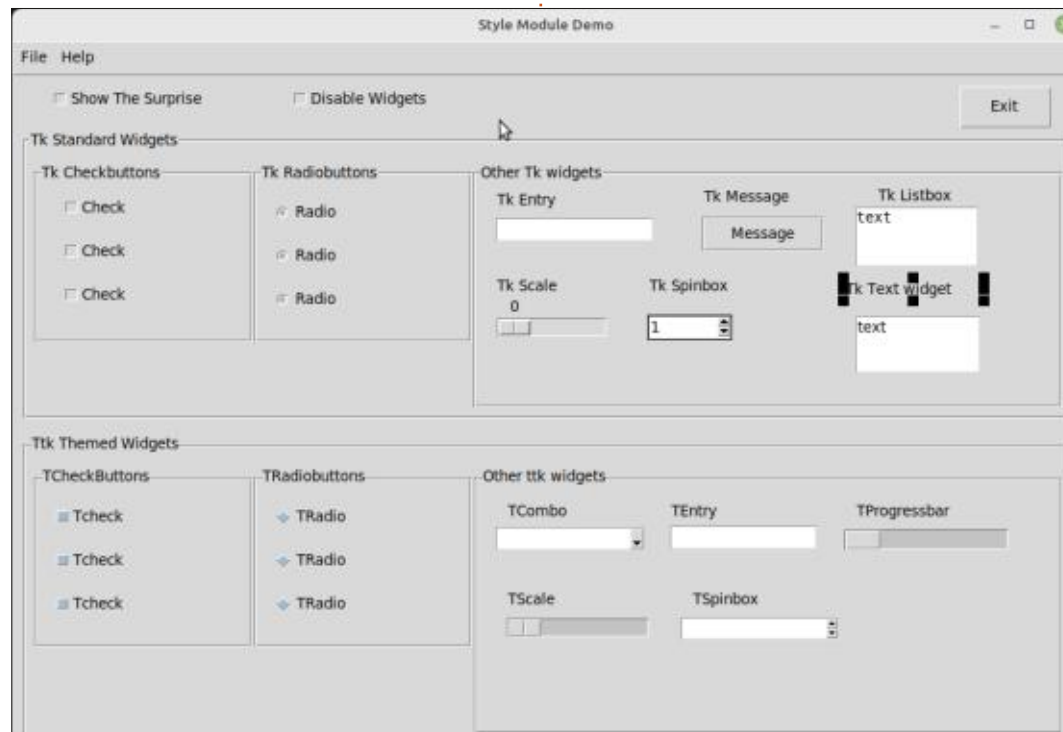
It didn't cause anything to throw any errors, so I just left it in. Eventually, I was trying to clean out some of the unneeded code and I came back to the function that I didn't know what it did. It wasn't in all the themes that I looked at in my learning process so I decided to look up what it was for. Here is the code (top right).

It turns out that the `tk_setPalette` function was created to allow for the normally non-themeable Tk widgets to get a pseudo theme applied by letting the `tk_setPalette` function override

the normal database for the Tk widgets. We'll see more about this when we get into the demo program code, which we'll do now.

Below is what the PAGE designer version of the demo program looks like.

You can see that I've created the program to use two Labelframes, one for some "generic" Tk widgets and one for the ttk versions of them (where available). Now we'll look at the code for the demo program.



```
def set_palette(toplevel):
    # https://www.tcl-lang.org/man/tcl8.6/TkCmd/palette.htm
    toplevel.tk_setPalette(
        activeBackground=_activebgcolor,
        activeForeground=_activefgcolor,
        background=_bgcolor,
        disabledForeground=_disabledfgcolor,
        foreground=_fgcolor,
        highlightBackground=_bgcolor,
        highlightColor=_fgcolor,
        InsertBackground=_fgcolor,
        selectColor=_selectbackground,
        selectBackground=_selectbackground,
        selectForeground=_selectforeground,
    )
```

In our demo program, we will skip all the normal imports and jump right into the startup function (bottom right).

You can see, we define our `sty` instance of the `ttk.Style` object, then we call the `create_styles` function of the styling module. Unfortunately, the `ttk.TLabels` don't normally like to be changed via a style, so we use the function `fix_TLabels` to set the proper

foreground and background colors. Then I put some data into the `Tk.Listbox` and `Tk.Textbox` and finally start the `TProgressbar` which was set to `Indeterminate` mode in `PAGE`.

Here is the `fix_TLabels` function. I get the background color from the styling module, then create a list of all the `TLabel` aliases. Then I simply walk through the list, applying the background color (next page, top right).

```
def startup():
    global sty
    sty = ttk.Style()
    os_default_theme = sty.theme_use()
    last_style = os_default_theme
    mystyles_notsodark.create_styles(sty)
    fix_TLabels()
    setup_list_and_text()
    _w1.TProgressbar1.start()
```

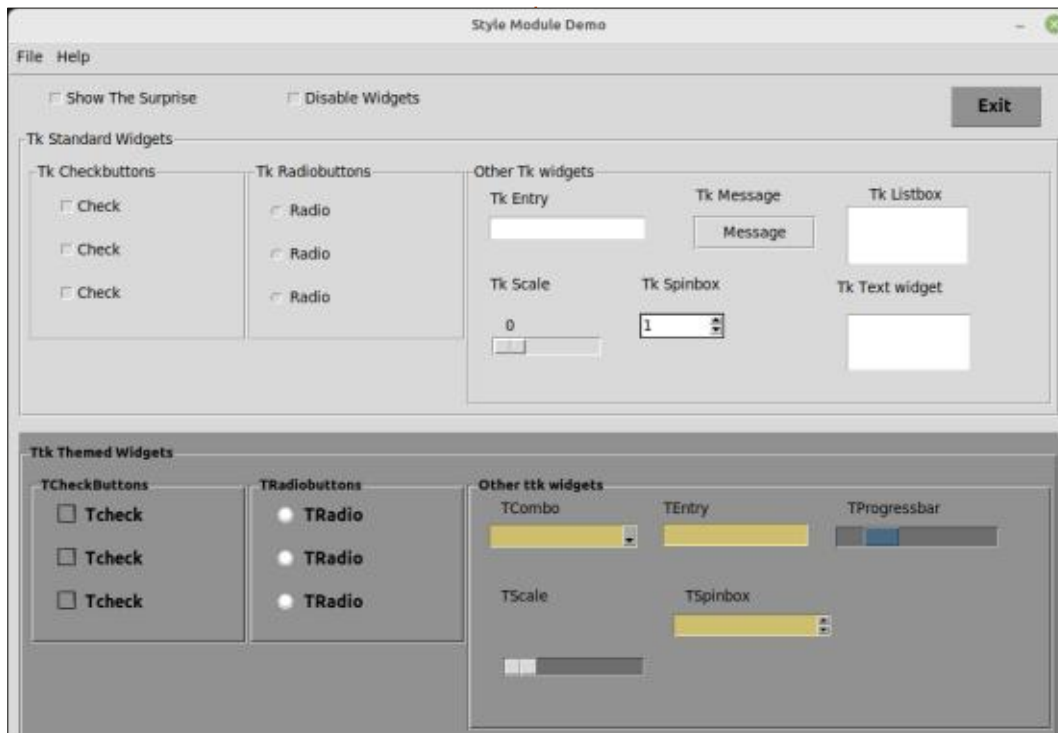
If you look back at the image of the program in the PAGE designer, you will notice two checkboxes near the top of the form near the left side. One is labeled “Show the surprise” and the other “Disable Widgets”. We’ll deal with the “Disable Widgets” callback first (bottom right).

Again, I’m going to show only a portion of the widget list. Just like I did in the `fix_TLabels` function, I created a list of the aliases of the widgets that I want to be able to control the disabled and normal

states. This way, I can demonstrate the way the widgets respond in the different states. The actual list contains both the Tk and ttk widgets that respond to being disabled. I use variable `che56`, which PAGE assigned for me to find out if the checkbox is checked or not. Based on that, I walk through the widget list and set each widget to the proper disabled or normal state.

Below is the program running.

Now you can see the styling module in “action”. The



```
def fix_TLabels():
    bgcolor = mystyles_notsodark._bgcolor
    print(f"{bgcolor=}")
    labelList = [_w1.TLabel1, _w1.TLabel2,
                 _w1.TLabel3, _w1.TLabel4, _w1.TLabel5]
    for label in labelList:
        label.configure(background=bgcolor)
```

```
def on_disableWidgets(*args):
    widgetList = [
        _w1.TCheckbutton1,
        _w1.TCheckbutton2,
        _w1.TCheckbutton3,
        _w1.TRadiobutton1,
        _w1.TRadiobutton2,
        _w1.TRadiobutton3,
        _w1.Radiobutton1,
        ...
        _w1.TSpinbox1,
        _w1.Listbox1,
        _w1.Text1,
    ]
    if _w1.che56.get() == 1:
        for widget in widgetList:
            widget.configure(state=DISABLED)
    else:
        for widget in widgetList:
            widget.configure(state=NORMAL)
```

TCheckbuttons and the TRadiobuttons are using the custom graphics, and the other ttk widgets now have the common style. I set the TCombobox, the TEntry widget, and the TSpinbox to have a complimentary background color that isn’t the standard bright white.

callback. The idea here is to call the `tk_setPalette` function to show the ability to use the style on the Tk widgets (next page, top right).

To be able to set the palette back to the “normal” Tk colors, I had to duplicate the `set_palette` function and then I set the colors back to the standard colors.

Now we can look at the “Show The Surprise” checkbox

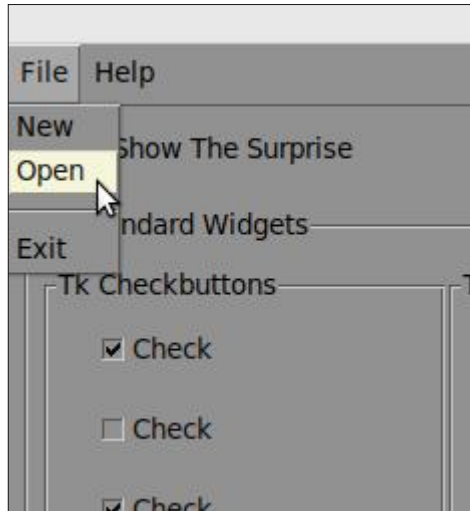
HOWTO - PYTHON

Below is what it looks like when we have the “Show The Surprise” checkbox and the “Disable Widgets” checkbox selected at the same time.

All the widgets now are not only themed, but disabled, and you can see that it’s pretty obvious that the widgets are disabled. Even if the Radiobutton or Checkbuttons are selected, you can still tell those that are selected.

One additional feature that the set_Palette function provides is to set the color for the menubar as

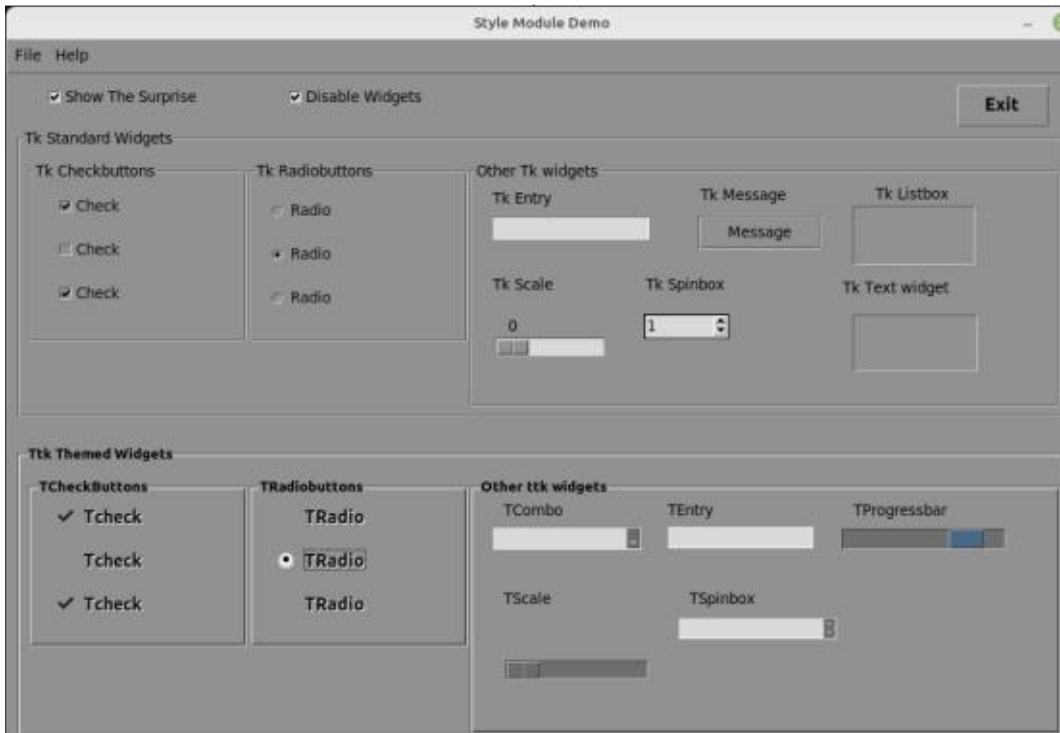
well.



```
def on_chk1(*args):
    if _debug:
        print("StyleModuleDemo_support.on_chk1")
        for arg in args:
            print("    another arg:", arg)
        sys.stdout.flush()
    if _w1.che50.get() == 1:
        mystyles_notsodark.set_palette(_top1)
    else:
        mystyles_notsodark.reset_palette(_top1)
        fix_TLabels()
```

All the files for this project are located in my repository at <https://github.com/gregwa1953/FCM194>.

Until next time, as always; stay safe, healthy, positive and creative!



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.thedesignedgeek.xyz.



HOW-TO

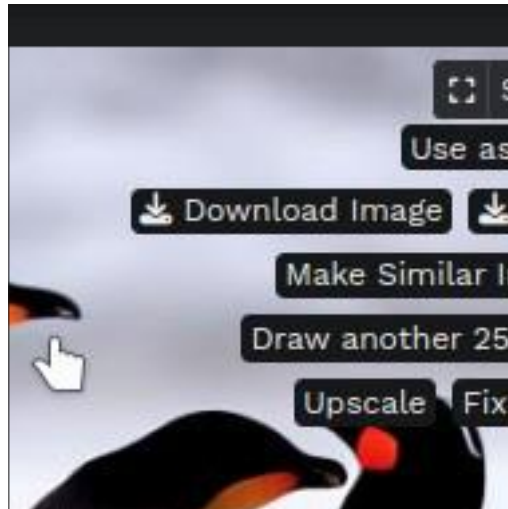
Written by Jon Hoskin

In this part of An Introduction to Stable Diffusion, we will learn to manipulate and edit an image we create with Inpainter. I started this article with 2.5.36 and several upgrades have occurred since, and it's currently 2.5.38. We begin by creating an image – “a photograph of four penguins dancing,” generating 20 images, and selecting the image below. The image generated is something I would like to keep with the exception of the fourth penguin's beak in the upper right, and I can remove that with Inpainter.

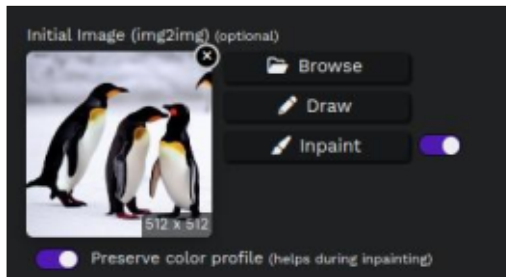


With the cursor on the image,

select “Use as input.” Then, in the area below the prompt, you will see the image selected.

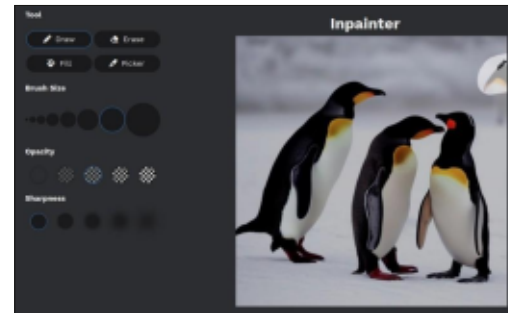


Next select Inpaint to the right of the image and the Inpainter window opens.

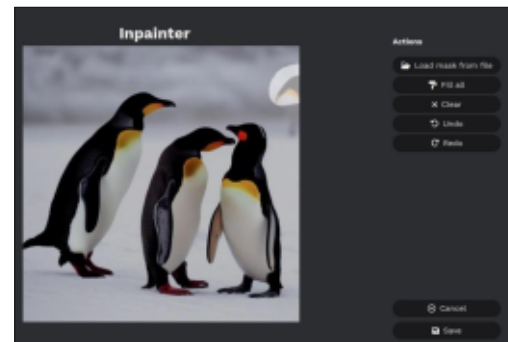


By default, the Draw option is selected and you are ready to go. You can change the Brush Size, Opacity and Sharpness of the

Inpainter brush. Although it works fine by default, you may want to change the Brush Size depending on what you want to remove. You can Erase (erroneously!) if you highlight too much.



When satisfied, select the save button at the bottom of the window and close the Inpainter window by clicking on the X in the upper right corner. Then generate new image(s) using the original parameters. The prompt may need to be changed to reflect what is desired in the final image.



Stable Diffusion Pt.4

The result should be something like what is shown below. As with any manipulation of graphics, the result depends on the limitation of the tools and the skill of the individual.



Using the same image, let's try adding some color. With the cursor over the new image, again select 'use as input', but, this time, select Draw instead of Inpaint. This time, you will see typical color and options. Draw is essentially the same as Inpaint except you are adding color rather than removing what already exists. I added some color to the right penguin, saved,

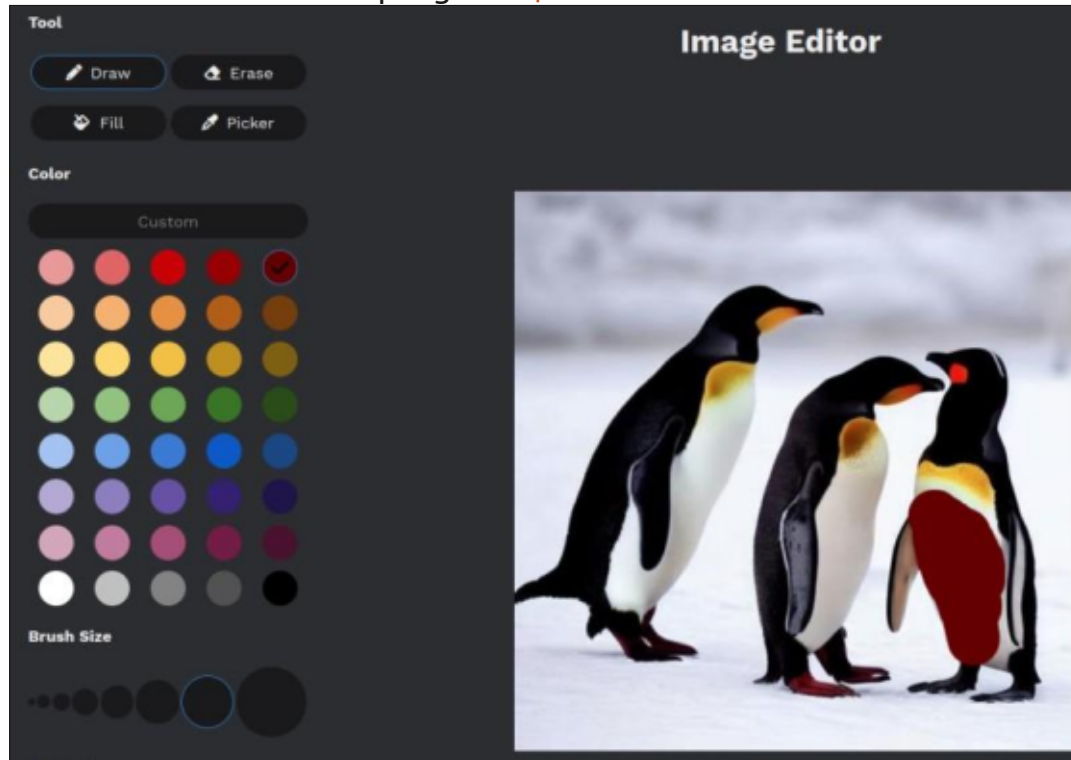
HOWTO - RESCUEZILLA FOR BACKUPS

and exited.

After generating another 20 images, several look like what I was expecting. I preferred the one on the right, but it has a few artifacts and while you may not see it, there appears to be some text-like addition at the bottom. Depending on the issue, using GIMP or other graphic editor may be easier in removing the unwanted text. Also, because I didn't paint the entire white area on the front of the penguin, one result had a jagged edge and there seems to be some bleed over onto the next penguin

on the second image generated. As with any modification, you may need several attempts to get the desired result.

Next time we will look at Outpainting followed by adding additional models.





HOW-TO

Written by Robert Boardman

For this project, I am going to put together a small book from several files. Since I like to cook, I also collect recipes, particularly recipes of Chinese dishes. So this book is going to be a cookbook. The techniques and procedures can be used for any book. I went over putting several files together to generate one PDF in FCM #190. This project will use the techniques discussed in that previous work and expand upon them.

The first thing we need to do is open a new document. The document class needs to be "book". The document class

Contents	
1 Bao	1
2 Chinese Steamed Buns (Bao Tze)	5
3 Simple Dipping Sauce	7
4 Dumpling Fillings	9
5 BBQ Pork	11
6 Steamed Bun Dough	13
7 Char Siu Bao - Chinese Steamed Pork Buns	15
8 ChaSioPao	17
9 Chinese Sesame Buns	19
10 Chung Steamed Buns	23
11 TangYuan - Bean Paste Dumplings	25
12 Boiled Dumpling with Chili Oil	27
13 ChaSiu Filling	29

"article" could be used but that will change the options available. For example, 'article' sets margins for printing one side only. The 'book' class is designed to produce a PDF that could be printed and bound as a book with two-sided printing. This means margins are not left and right, but inside and outside. It also means the text of the book starts on a right-hand page and that each chapter starts on a right-hand page. If a chapter ends with a right-hand, odd-numbered page, then a blank left-hand page is inserted before the next chapter starts. When using the document class 'article', pages are neither left nor right, text flows from one page to another as if being printed in one long column.

Top right is the code I will start with. I have put my annotations below the code.

All lines starting with a percent symbol are comments. All lines starting with a back slash are Latex instructions. \frontmatter, \mainmatter, \backmatter instructions work with the 'book'

```

\documentclass[letterpaper,11pt]{book}
%preamble
\usepackage{inputenc, cooking, cooking-units}
\author{Robert Boardman}
\title{Cookbook}
\begin{document}
\frontmatter
\maketitle
\tableofcontents
\mainmatter
\include{Bao}
\include{BaoMrsChiu}
\include{Buns}
\include{CharSuiBao}
\include{ChaSioPao}
\include{ChineseSesameBuns}
\include{ChungSteamedBuns}
\include{beanpastedumpling}
\include{boildumplingchilioil}
\include{ChaSuiFilling}
\backmatter
% bibliography, glossary and index would go here.
\end{document}

```

document class but are not allowed in the article document class.

\tableofcontents uses the \chapter instructions in each of the recipes to make the table-of-contents automatically. The \include commands bring the individual recipe files into the PDF. These files are in the same directory as this "master" file. That is not necessary. If they were in one or more different directories,

the directory names are added to the include, for example \include{recipe/Bao}. Latex assumes names of included files end with ".tex".

This revision of the cookbook does not have any back matter. A real cookbook would have a recipe name index and an ingredient index. Since these recipes include ingredients specific to Chinese food, I might also include a

glossary. Those are topics for another article (or two).

Top right is a sample of one of the included files.



Notes for the included file:

```
\chapter{chapter title}
\section*{section name} The *
means this section is not numbered
\begin{itemize} starts an
unordered or bulleted list
\item 3 1/2 cups flour
\item 1 tablespoon shortening
\end{itemize}
\section*{Instructions}
\begin{enumerate} starts a
numbered or ordered list
\item Put 3 cups of the flour into a
bowl.
...
```

```
\end{enumerate}
```

When the PDF is built, the instruction `\include{Bao}` is replaced by the contents of the Bao.tex file. That means the Bao.tex file does not have any of the introductory or concluding instructions. This particular recipe does have text after the list of instructions. It is important not to include the `\end{document}` instruction at the end of the included file.

This is my first edit of the first part of the cookbook. It requires more work. For example, the fraction in 3 1/2 cups of flour needs to look like a fraction. I could write it as a decimal, but that would cause confusion. “Does 3.5 mean 3 1/2 or 35 cups?” So I need to change the 1/2 so that Latex formats it as 1/2. (Any other fractions should be reformatted as well.) Ideally, all dry ingredients would be measured in grams, and all liquid ingredients in millilitres. That would eliminate the fractions but also make the cookbook unsaleable in Canada and the U.S.A.

There is at least one file currently included that has more

```
\chapter{Bao}
\section*{Ingredients}
\begin{itemize}
\item 3 1/2 cups flour
\item 1 tablespoon shortening
\item 1/4 cup sugar
\item 1 pkg active dry yeast (2 1/2 teaspoons)
\item 1 cup warm water
\end{itemize}
\section*{Instructions}
\begin{enumerate}
\item Put 3 cups of the flour into a bowl.
... more instructions
\end{enumerate}
Other text can be added after the instructions section.
Note: Leftover buns may be served after reheating by
steaming.
```

than one recipe in it. I want to separate each recipe into a different tex file and a different include, so I can arrange the recipes in the order I want. I will probably add “chapters” – which will be section dividers: perhaps meat, eggs, vegetables, soup, fish, etc. Or perhaps sections for dishes from different parts of China.

Cookbooks deserve to be illustrated with photos. Some recipes can benefit from images as well as words to show how a dish is put together.

I need to decide if I am happy with the way the book looks. Do I want to change the page size or the margins or the default fonts

being used? Do I want to add some color? Drop caps are nice if used judiciously. There is the end matter to build: index, indices, glossary. You can probably think of other changes you would make if you were writing, editing, publishing a book. We will work through some or perhaps all of these features in the next few columns.

KILOBYTE MAGAZINE

Kilobyte Magazine is a fanzine for 8bit enthusiasts. It covers consoles, computers, handhelds and more, as well as new games for old systems. If you grew up with Commodore, Atari, Sinclair or Amstrad, this magazine is for you.

<https://retro.wtf/kilobytemagazine/>



HOW-TO

Written by Mark Crutch

Over the previous few instalments, I've been looking at Inkscape's new multi-page feature, culminating in last month's analysis of how it's implemented within the SVG file. What I found was that the details of each page are stored in a proprietary 'page' element in the Inkscape namespace – appearing in the XML editor as `<inkscape:page>` – rather than in a standard SVG `<view>` element. This is a strange choice, in my opinion, as using the latter would have made the extra pages more accessible via a web browser.

Last month, I described a couple of methods by which a browser can be coaxed into showing the extra pages. The first used a little-known version of the URL fragment identifier to target each page by its `viewBox` values. This doesn't require the file to be modified, but

does need you to dig into the XML to extract the relevant numbers. The second approach modifies the file to manually add `<view>` elements, which is more work but does allow you to use a more semantic URL. If you haven't read the previous article, I strongly advise doing so before proceeding, as this one directly builds on that information.

Our aim here is to make it easier to access all the pages of an Inkscape document via a web browser. One thing browsers do, which Inkscape doesn't, is to run JavaScript. So, by adding a small JS function that will be executed when the file is opened in a browser, we can automatically create `<view>` elements from each of the `<inkscape:page>` elements, without the need to manually copy-and-paste coordinates. This is



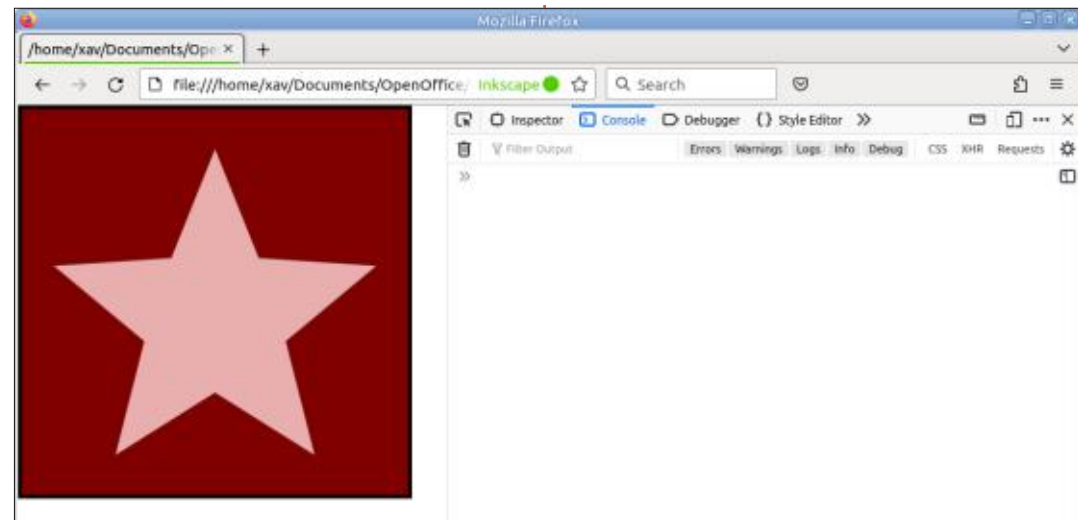
obviously useful when there are a lot of pages to consider, but even for files with just a couple of pages, the ability to just paste the same snippet of code into any file without having to modify it each time makes this approach arguably simpler than manually creating the `<view>` elements yourself.

We'll be using the same multi-page SVG file from last month, consisting of four pages arranged horizontally, with a different background color and content in each.

Having created a multi-page Inkscape file, the first thing to do is

to save it to disk and open it inside a web browser. This will display only the first page, as expected. As we progress through this code, we'll log some data to the developer console, so open the dev tools in the browser (typically by pressing F12), and switch to the 'Console' tab. Below is how it looks in Firefox on my Ubuntu Mate machine.

I've talked about adding JavaScript code to Inkscape files in the past, so I won't go into depth about the options here. For what we're trying to achieve, we just want a simple chunk of script embedded directly into the file



which will run at load time. The easiest way to create this within Inkscape is as follows:

- Open the File > Document Properties... dialog
- Switch to the 'Scripting' tab.
- Within that tab, choose the 'Embedded scripts' tab.
- Click the '+' button below the (empty) list of embedded scripts.
- You should see a new entry appear in the list with an arbitrary ID.
- Ensure that entry is selected.
- Write your code in the 'Content' box below.
- You can save (Ctrl-S) as you develop, without having to close the dialog.
- Whenever you save the file, manually reload in the browser (F5) to see the effect.

Let's begin by logging out the existing `<inkscape:page>` elements to the console using this code (top right).

It's not essential to fully understand the workings of this code in order to use it, but I'll describe it anyway for those who are interested. The first two lines simply set up variables for the namespaces we'll need. We won't be using the SVG namespace just

```
const inkNS = 'http://www.inkscape.org/namespaces/inkscape';
const svgNS = 'http://www.w3.org/2000/svg';
const pages = document.getElementsByTagNameNS(inkNS, 'page');

Array.from(pages).forEach((page, idx) => {
  console.log(idx, page);
});
```

yet, but we will require it soon, so this is a good time to introduce it. The third line uses one of the browser's built-in functions to find all the 'page' elements in the Inkscape namespace, and assign them to a variable as a 'collection'.

For historical and technical reasons, a 'collection' is very similar to a JS array, but not quite actually the same thing. These days, JavaScript's array functions are pretty powerful, so we would really like to create an array from the content of our collection so we can use those functions. The 'array.from(pages)' part does exactly that, iterating over each entry in the collection to build up a temporary array. We can then use the 'forEach' array method to execute a block of code for each entry in the array.

The `forEach()` method expects to have one parameter, and that parameter should be a function.

We could build a function elsewhere and pass its name in here, but it's more common in JS to see 'anonymous' functions used for small tasks like this one. In this case, the anonymous function is called once for each element in the array, and, each time it's called, it's passed the current element (in a variable we've called 'page') and the index of that element in the array ('idx'). The '=>' syntax is used for so-called arrow functions, and can be thought of largely as an alternative to the 'function' keyword you might be more familiar with if you haven't gone near JS for a while.

Inside curly braces ('{...}') we have the body of the function – just a single line that uses the `console.log()` function to print the `idx` and `pages` values to the developer console. Finally, the last line closes not only the function body, but also the end of the `forEach()` method.

If you're not very familiar with JavaScript, then do take a few minutes to try to understand the code above. It's particularly useful to examine where each pair of brackets ('(...)') and braces ('{...}') start and end, and what content is inside each one.

With that code in place, save the file and reload it in the web browser, and you should see something in the console looking similar to this:



The green numbers are the `idx` values, starting at zero because...

HOWTO - INKSCAPE

well, there are good reasons, but this is not the place to go into them. Suffice to say that most programming languages use zero-based indexing for things like arrays, and JavaScript is no exception.

Following each green number, you can see an XML representation of each SVG element in the array – the four `<inkscape:page>` nodes we’re interested in. Each node also has a whole load of other baggage attached to it in the JS world, and you can see much of that by expanding the small triangle next to each one. In practice, we don’t need any of that for our task, so feel free to leave that triangle unexpanded, or to collapse it back down again if curiosity does get the better of you.

So far, our code hasn’t really achieved very much – just printing the index, and the same nodes we can see in Inkscape’s XML editor. But now that we’ve got a way to grab a handle to each `<page>` element, we can start to pull them apart to get to the individual details we’ll need. We’re going to want to extract the `x`, `y`, `width`, and `height` values. These are stored as ‘attributes’ on the element, and

can be retrieved using the `getAttribute()` method. Let’s add four lines after the `console.log()` to retrieve these values, and assign each to a JS variable (`x`, `y`, `w`, `h`).

```
const x =  
page.getAttribute('x');
```

```
const y =  
page.getAttribute('y');
```

```
const w =  
page.getAttribute('width');
```

```
const h =  
page.getAttribute('height');
```

We could log them out at this point, but ultimately we’re going to want these formatted into a space-separated string for use in a `viewBox` attribute. One additional line of code will do this for us, and then we’ll log the result:

```
const viewBox = `${x} ${y} ${w} ${h}`;
```

```
console.log(viewBox);
```

Be very careful! The ‘`const viewBox`’ line uses a feature of JS called ‘template strings’. These are delimited by backticks (```) rather than normal quotes or apostrophes. Using this method lets us put our variables directly into the string using the `${}` notation, and the JS engine will

swap them out for the variables’ values when the code runs. If your log ends up containing the actual `${}` string, then you’ve used the wrong type of quotes, and will need to search your keyboard again for the easily-overlooked backtick character.

All those previous 6 lines should have been added after the existing `console.log()` and before the closing brace. If you’ve done it correctly, saving the file and reloading it in the browser should show something like this – similar to the previous output, but with the `viewBox` values displayed after each XML node (see image below).

Now we’ve got all the details we need, it’s time to create a new `<view>` element for each page. For this, we need to use the `document.createElementNS()` function – the NS on the end referring to the fact that this lets

us specify a namespace for our new element. This is where our earlier assignment of the `svgNS` variable will be used, ensuring that we end up with what is effectively an `<svg:view>` element, rather than an `<inkscape:view>` element, or anything else.

Each `<view>` element will also require two attributes. One is the ‘`viewBox`’ for which we’ve already prepared the value. The other is an ‘`id`’ attribute which will define the string we have to append to our URL to view this page. For the sake of simplicity, we’re just going to name the pages ‘`page-1`’, ‘`page-2`’, and so on, using another JS template string in which we’ll also add 1 to the value to rid ourselves of those pesky zero-indexed numbers. Therefore, to create our new element, and set both attributes, we’ll need these three lines of code:



HOWTO - INKSCAPE

```
const view =  
document.createElementNS(svgNS,  
S, 'view');
```

```
view.setAttribute('id',  
'page-${idx + 1}');
```

```
view.setAttribute('viewBox',  
viewBox);
```

There's just one thing left to do. Although we've created our new <view> element, it currently just lives as an object in the JS world, and needs to be inserted into the browser's internal model of the document. We'll insert each <view> as a child of the corresponding <inkscape:page> element in order to keep things neatly together. This last line will do the job:

```
page.appendChild(view);
```

With that, the final code should look like this – albeit that you can't actually see all the lines at once in Inkscape's unfortunately inflexible editor field (top right).

You can optionally remove the console.log() lines if you wish, as they're purely there for educational and debugging purposes, and have no effect on the actual operation of the code.

Now that the code is done, how do you actually use it? Simply load

the SVG file directly into your browser, and append '#page-2' to the end of the URL to view the second page. I'm sure you can work out the syntax for the other pages. Entering an invalid ID (e.g. '#page-22') simply causes the browser to show the first page.

There we have it: a small chunk of JS that you can add to any multi-page Inkscape file to make the additional pages available via a web browser. What more could you possibly want?

Quite a bit as it happens. This code is good, but it suffers from a significant limitation: in order for the browser to execute it, the SVG file has to be loaded directly (or within an <object> tag). Most common ways of including SVG files in a web page – via an tag or a CSS url() function – are deliberately prevented from executing JavaScript. This significantly limits the usefulness of this code – at least as it stands.

Another issue is that we've just given each page a rather generic ID. Within Inkscape, it's possible to name each page – wouldn't it be nicer if we could use those names when referring to each page, rather

```
const inkNS = 'http://www.inkscape.org/namespaces/  
inkscape';  
const svgNS = 'http://www.w3.org/2000/svg';  
const pages = document.getElementsByTagNameNS(inkNS,  
'page');
```

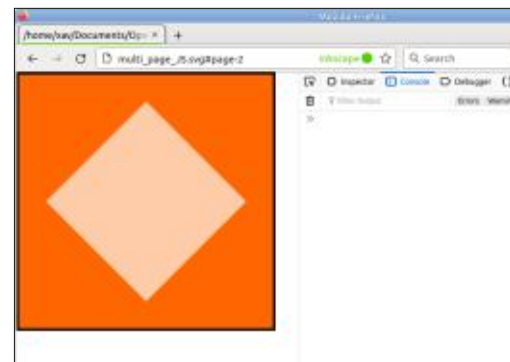
```
Array.from(pages).forEach((page, idx) => {  
  console.log(idx, page);
```

```
  const x = page.getAttribute('x');  
  const y = page.getAttribute('y');  
  const w = page.getAttribute('width');  
  const h = page.getAttribute('height');  
  const viewBox = `${x} ${y} ${w} ${h}`;  
  console.log(viewBox);
```

```
  const view = document.createElementNS(svgNS, 'view');  
  view.setAttribute('id', `page-${idx + 1}`);  
  view.setAttribute('viewBox', viewBox);  
  page.appendChild(view);
```

```
});
```

than just page-1 and page-2?



Next month, I'll extend this code a little further to address both these issues. It still won't give the simple, seamless experience that we could have had if Inkscape

natively created named views, but it's better than being stuck with multi-page files that can display only the first page!



Mark uses Inkscape to create comics for the web (www.peppertop.com/) as well as for print. You can follow him on Twitter for more comic and Inkscape content: [@PeppertopComics](https://twitter.com/PeppertopComics)



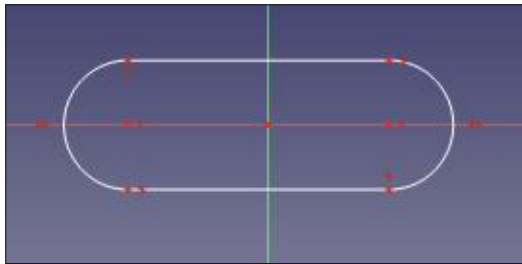
HOW-TO

Written by Ronnie Tucker

FreeCAD - Part 6

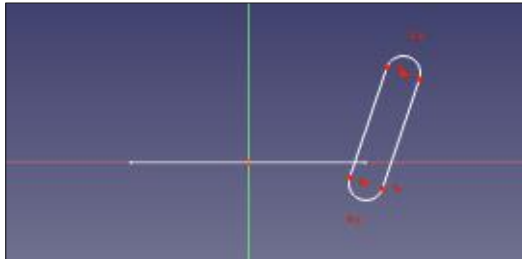
So, let's start a new project. It is, of course, File > New, then create a new body, and a new sketch, in the XY (top-down) plane.

This is where we want to make the path that our shape will follow.



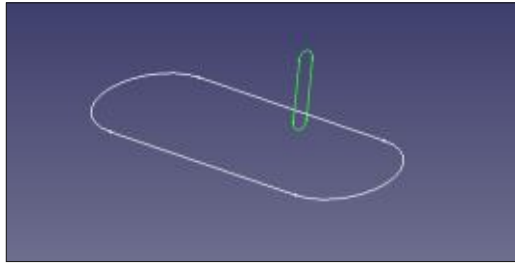
For this I used the Create Slot tool.

Now close that sketch and create a new sketch in the YZ (right) plane. That white line you see is our last sketch, the path. So I used the slot tool again to draw a quick shape.



Now for the magic.

Close that sketch.

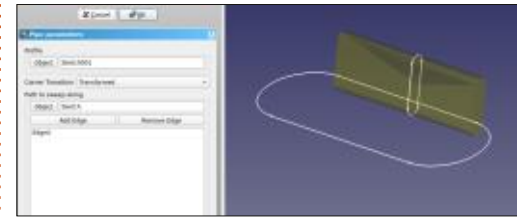


What we're seeing here is the path (in white) and the shape that'll follow that path (in green). Make sure the shape is selected (in green). If not, select it from the left panel which shows both sketches.

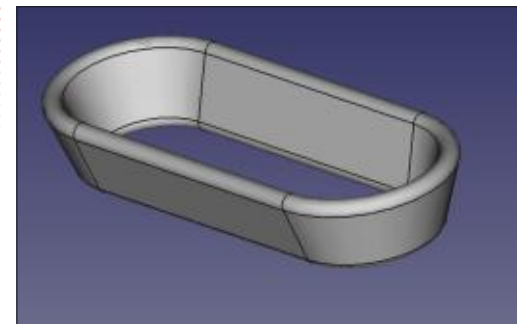
With the shape selected, click the Additive Pipe tool.



In the left panel, click Add Edge and select a piece of the path.



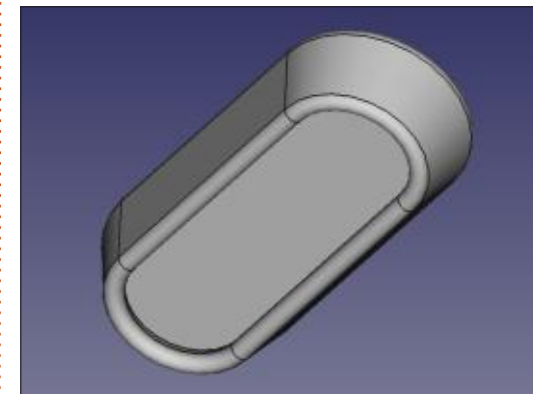
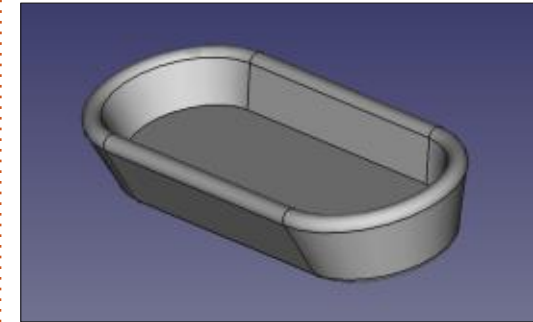
Don't worry about it looking semi-transparent or the color of it. This is just a preview. Click Add Edge and add the remaining edges. When all is done, click OK.



Nice! Now we have an irregular shape as a path. Now, be careful with the paths you make, as a wonky angle or something similar will cause your shape to also go a bit wonky. So, if your shape doesn't close properly, it's probably down to a wonky path.

Let's finish up by adding a crude bottom to our makeshift bath shape.

Click the sketch for the path (you may need to click the arrow beside AdditivePipe in the left panel to see the sketches), and click the Pad tool. As ever, edit the thickness of the padding.



You can, of course, double-click the sketch (in the left panel) to edit the path to then edit the object. Give it a try.

THE DAILY WADDLE

WENT FISHING WITHOUT A
LINE ... NOT EVERYTHING
IS BETTER WIRELESS





BACK NEXT MONTH



MICRO THIS MICRO THAT

Written by Greg Walters

Before we get started on this month's subject, I want to give you a little bit of an update on MicroPython 1.20.

As of the first of June 2023, the I2C issue still somewhat exists. You CAN use I2C, but, at this time, you can't use the old defaults. I say old defaults because it looks like the people at RPi asked that the defaults be changed to what they are now. So, to be sure that your I2C projects will work under 1.20, use the table below to set your I2C pins.

As to the NeoPixel driver, you CAN use the one you used previously, but should rename it, just to avoid any confusion between you and MicroPython. I've tested this using the Pico-W and the Pico with three sets of 1-meter strings of 30 devices each.

I2C(0)				I2C(1)			
SDA		SCL		SDA		SCL	
GPIO	Physical	GPIO	Physical	GPIO	Physical	GPIO	Physical
4	6	5	7	6	9	7	10

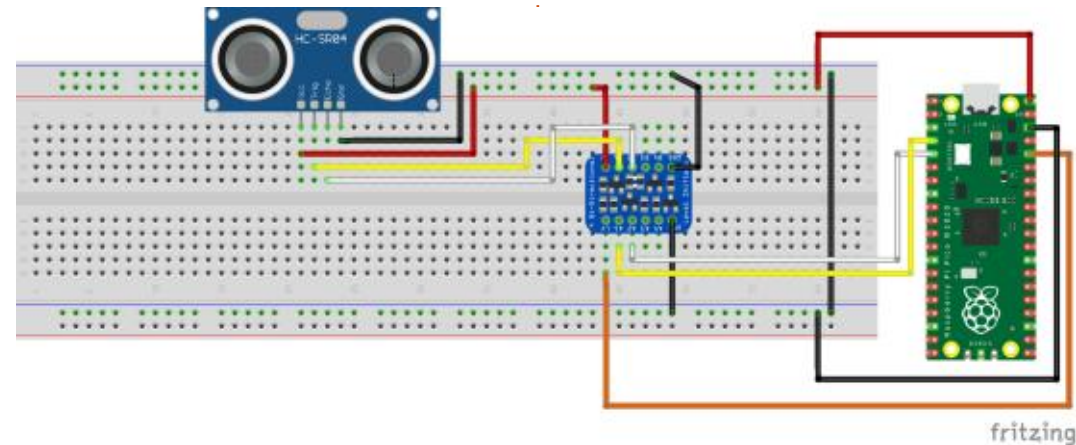
Everything works fine.

Also, as of June 2023, the BLE bluetooth drivers were still being worked on, but were showing great progress. When done, you should be able to use most code from ESP-32 boards.

I'll give you a hint about the subject of the next MTMT, which should be controlling motors using the Pico-W. This is a great way to break into robotics.

Now onto the subject of the month, the HC-SR04 ultrasonic obstacle avoidance sensor.

As far as I know, this sensor comes only in a 5Vdc version. However, not all is lost. I was able to get a small 4-channel I2C-safe Bi-directional Logic Level Converter that costs about \$4.00 USD. It



works well.

You can see it in the wiring image above.

The pinout is pretty straightforward, even if you are using the logic level converter. I've created a small table here (below) to help you along.

I've tested the project with and

without the logic level converter, and I haven't noticed any major difference between the two setups, so if you don't have a converter, you should be able to use it without any problems.

HOW IT WORKS

There is a lot of science behind this little sensor and I would be

	VCC	Trigger	Echo	Ground
Pico/Pico-W GPIO		2	3	
Pico/Pico-W Physical	40	4	5	38

MICRO THIS MICRO THAT

remiss if I didn't address it here. So I'm going to put on my science geek hat before we get into the code.

Looking at the datasheet, whenever the Pico/MicroController wants to see if there is anything within its range, it sends out a pulse on the Trigger pin of the sensor. That causes the sensor to send out a burst of 8 pulses of ultrasonic sound at 40 kHz through the emitter and then it starts listening for an echo. The ultrasonic sound will bounce off most objects within the 4-meter maximum range and hopefully come back to the receiver. The amount of time between the sending and receiving is measured (in microseconds), and can be converted into a distance measurement using a formula $\text{EchoTime}(\mu\text{s}) / 58 = \text{centimeters}$.

This is then sent back to the driver which, in turn, sends the distance back to our program.

Our program can then take the information from the driver and simply display it, or use it to determine how much closer to the detected object the application wants to get.

The sensor has a range somewhere between about 2 cm and 4 meters, but the max distance could be anywhere between 2 and 4 meters.

All of that having been said, we have to remember to consider what kind of material we are going to be encountering as we try to use this project.

Because we are relying on an echo coming back to the sensor, we

have to take into account what type of object the ultrasonic signal is bouncing off of. If it's a hard surface like wood or drywall, there won't be much absorption. However, if the object is like a curtain, a fair amount of the signal will be absorbed and might give you false data.

Another thing you want to take into consideration – if you are going to mount the sensor on, let's say, a rover or small robot – make sure that the sensor is mounted as straight as possible at the front of the device. This is due to the signal's reflection not returning. The law of physics says that the angle of incidence is equal to the angle of reflection. This gets into some pretty deep science, so I'll avoid a discussion here. If you are curious, you can look at <http://www.fast.u-psud.fr/~martin/acoustique/support/r%C3%A9flexion-r%C3%A9fraction.pdf> or [https://en.wikipedia.org/wiki/Angle_of_incidence_\(optics\)](https://en.wikipedia.org/wiki/Angle_of_incidence_(optics)). You could also do an Internet search for “ultrasonic reflection” to find other sources of information.

There are a few drivers out there and you can certainly “roll your own”. However, I'm going to suggest you start with one that works pretty well for me. You can find it, as well as their code to run the driver on the RPi Pico/Pico-W, at <https://microcontrollerslab.com/hc-sr04-ultrasonic-sensor-raspberry-pi-pico-micropython-tutorial/>

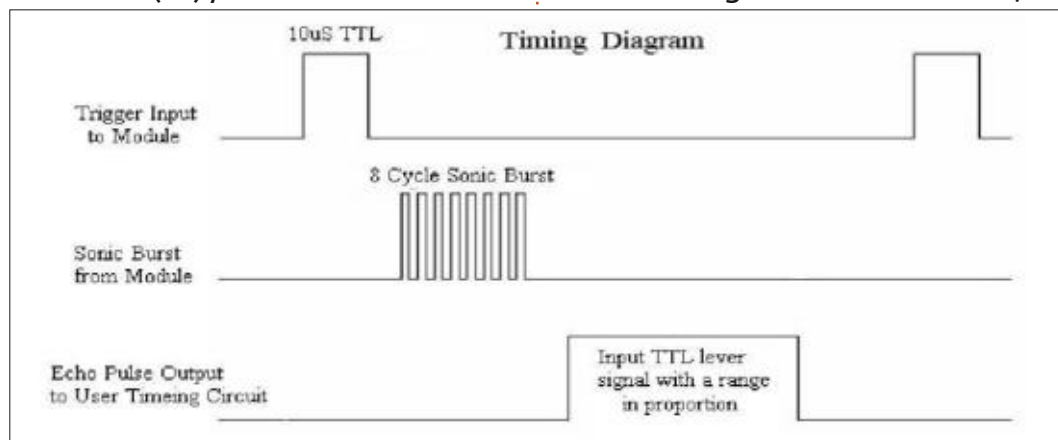
They have a much deeper explanation of how the whole thing works, so you might want to take a look at their article as well as getting the code. I'll also put it in a repository for your convenience.

First the driver. I'll show only the important parts here.

```
import machine, time
from machine import Pin
```

First, we have the imports. The driver simply uses machine, machine.Pin, and time.

Next, we have the actual class. I'm going to omit most of the comments.



THE CODE

```
class HCSR04:
    # echo_timeout_us is based in chip range limit (400cm)

    def __init__(self, trigger_pin, echo_pin, echo_timeout_us=500*2*30):
        self.echo_timeout_us = echo_timeout_us
        self.trigger = Pin(trigger_pin, mode=Pin.OUT, pull=None)
        self.trigger.value(0)
        self.echo = Pin(echo_pin, mode=Pin.IN, pull=None)
```

The `__init__` function (above) sets up the trigger and echo pins and the echo timeout value (in microseconds).

The `_send_pulse_and_wait()` function sends a pulse of 10 microseconds to the trigger pin,

then starts looking for a return pulse on the echo pin, checking the elapsed time (bottom left).

The `distance_mm()` function is not actually called from the test program, but is there if you want to get return data that is more granular than the `distance_cm()`

function (below).

The `distance_cm()` function

```
def distance_cm(self):
    pulse_time = self._send_pulse_and_wait()
    cms = (pulse_time / 2) / 29.1
    return cms
```

(below) is the actual function that gets called from the test program. It first calls the `_send_pulse_and_wait()` function, and gets back the pulse time. Then that value is returned as the number of centimeters that represents the time between the trigger and the echo return.

```
def distance_mm(self):
    """
    Get the distance in millimeters without floating point operations.
    """
    pulse_time = self._send_pulse_and_wait()

    mm = pulse_time * 100 // 582
    return mm
```

If you want to see the reasoning behind the formulas, you can look at the comments in the actual driver file.

As for the test program, it's really much simpler than the actual driver.

```
def _send_pulse_and_wait(self):
    self.trigger.value(0) # Stabilize the sensor
    time.sleep_us(5)
    self.trigger.value(1)
    time.sleep_us(10)
    self.trigger.value(0)
    try:
        pulse_time = machine.time_pulse_us(self.echo, 1, self.echo_timeout_us)
        return pulse_time
    except OSError as ex:
        if ex.args[0] == 110: # 110 = ETIMEDOUT
            raise OSError('Out of range')
        raise ex
```

We import the driver class and `time.sleep`.

```
from hcsr04 import HCSR04
```

```
from time import sleep
```

Then, we instantiate the class from the driver, assigning the trigger and echo pins, and providing a timeout value in

microseconds.

I added an optional line of code to assign a button on GPIO 20 as an input. This is so we can poll pin 20 for a low to let the program know that the button has been pressed. If the button has been pressed, it will allow the program to exit the forever loop that we will create. There is no issue if you don't include a button in your setup. The code will still look for the falling signal, but it really won't make any difference if it doesn't see it. You can always just use the stop button in Thonny to end the program.

We then enter the 'forever' loop (right). We poll the sensor and get a distance value. Then I include a calculation to convert centimeters to inches, display them both on the REPL terminal, check to see if the button has been pressed (if there is one), break the loop if we get a 0 from the button pin, and finally sleep for 1 second before starting the loop all over again.

TESTING THE PROGRAM

I'm going to assume that you are using a breadboard to hold your project. Place it on a table and

place a ruler between the sensor and something fairly big like a box of tissues. This will give you a known reference. When you run the test program, you should see, about every second, something like this...

```
Distance 28.95189 cm -  
11.39838 in
```

If it works, and the distance returned is somewhat reasonable, you can test the sensor by pointing at something else. If not, you will most likely be getting a return that looks something like this...

```
Distance 225.0 cm - 88.58268  
in
```

This is the "default" return that means that the sensor didn't receive a clear echo back before the timeout value. Given the sensor has a range between about 2 cm and 4 meters, as long as your target is within about 3 meters, this could mean that you have the trigger and echo pins swapped. The other possibility could be that the breadboard you are using has a bad spot or two. Try reseating the Pico, the sensor, and your jumper wires, and try again. I've seen it happen that even a brand new breadboard might have a bad spot, or 5, and

```
sensor = HCSR04(trigger_pin=2, echo_pin=3,  
echo_timeout_us=10000)  
# Optional - Button to exit loop  
button = machine.Pin(20,machine.Pin.IN,machine.Pin.PULL_UP)  
# set pin 20 as INPUT with PULL_UP
```

```
while True:  
    distance = sensor.distance_cm()  
    #print('Distance:', distance, 'cm')  
    cm2in=distance/2.54  
    print(f"Distance {distance} cm - {cm2in} in")  
    if button.value() == 0:  
        print("Button Pressed")  
        break  
    sleep(1)
```

the normal "tech support" answer is if you are using a breadboard, try different positions for your components and/or check your wires.

Well, that's about it for this month. As I said, I've put the code for both the driver and the demo program in my repository at https://github.com/gregwa1953/FCM194_MTMT. Be sure to save your code for the project and the sensor, because we will be revisiting this in a future article when we get into the subject of robotics.

Until then, as always; stay safe, healthy, positive and creative!



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.thedesignedgeek.xyz.



Linux on Your iPad

For as low as \$4.95, you can have your own personal Linux cloud computer in minutes on any device.





UBPORTS DEVICES

Written by UBports Team



THE DAILY WADDLE

WE HAVE NO SPAM FILTERS IN HELL





MY STORY

Written by Abdul-Jabbar Bozdar

That CD

This all started back in high school. We had a computer lab there, but we used to learn more theory rather than practical. My computer enthusiasm didn't let me sit idly in the school lab. I made my way to a private computer training center to learn all about word processing, spreadsheets, image editing, and a secretive method to communicate with people abroad. Even my teachers didn't know what the world wide web was back in 2003.

I WANT GENUINE SOFTWARE ONLY

My first home computer arrived in 2007 when I had already headed to college. It was my first semester, and I was having my meal at the dormitory canteen; my father made a phone call and told me that he bought a computer for the family to use. I was not only delighted to hear that, but also rocking and rolling to have a computer at home.

After mid-term exams, I returned home to see my first-ever

personal computer. I used a laptop before but I won't call it the first PC for various reasons. I had it for only 2 months, and due to weak hardware, it was unable to run pre-installed Windows 98. This time too, I didn't have time to play with the new PC. My father was telling me the wonders of the Windows XP. I couldn't do anything except

listen and grin somewhere in my heart. My father vacated the seat and said out loud, "There you go computer champ".

After a while, I pulled a CD out of my collection and inserted it into the CD ROM. My father said, "it must be a new game and you ain't gonna be a rock-solid engineer".

Well, I am still not a rock-solid one though. I could only smile that time. I turned off the computer and restarted it. I booted from the CD and voila. There was a live operating system running.

My father claimed that the operating system looked nice, but not so intuitive. He asked me how



MY STORY

much money I wasted on this CD. I told him, it is free and open-source, friends from abroad sent it to me last year. My father said Windows is also free. Free for us at least. I replied, "It's pirated; I want genuine software only". Everything in this disc, each piece of software, and this disk too, is free of cost for human beings. Pa, try it.

FOR THE FIRST TIME, NOTHING WORKS

That day, everybody at home tried Ubuntu 6.06.1 LTS for the first time. I did not install it as it was not the option that day. The pirated Windows won. The story did not end there. When I finished my first semester, I returned home having great ambitions. At college, I learned about dual-booting. A teacher did not teach me that. I learned how to dual-boot Linux alongside manuals, online, and books at the library. In the evening, I used to go to visit the university's digital library. It helped me a lot. I dual-booted Ubuntu and played a lot. I saw the OGG audio format for the very first time. The disk already had a few images and a video of an African person. I don't remember him anymore. I got my first laptop

in 2010 with Windows XP pre-installed. That day, I killed Windows forever for personal use. We had to run Windows in the lab during college. There was no option as Ubuntu Linux software was limited. Since then, neither could I get back to Windows nor have I ever convinced my father to use Linux. The family computer runs Linux Mint these days. My younger brother's laptop also runs Linux Mint. I am accustomed to Ubuntu and currently using the 22.04 LTS version. I did some distro-hopping as well, but nothing compared to Ubuntu. My second priority is always an Ubuntu or Debian based distribution for testing and professional purposes.

Few distros work the same way Ubuntu has taken the Linux approach. Ubuntu Linux is an operating system for the masses. When I run a different distro and stumble upon a new problem, I find a solution on the Ubuntu forums already published instead of that distribution's forum. Sometimes I wonder what kind of fate Linux would have if there was no Ubuntu at all. Ubuntu worked, and it nourished my career in systems and engineering.

HOW DID I GET THE CD?

That's something I should have told you earlier, shouldn't I? In 2003, my course at the private computer training center did not include web and email or even programming. As the use of the Internet was tied to charges, I had to collect my pocket money for four days to buy one hour of the Internet. It used to cost me about \$0.30 USD an hour. This opened a new world for me. I began to chat over IRC networks, and click on advertisements to fill out dozens of forms for free stuff. This continued for many years to come. I don't remember when I reached the Ubuntu website and ordered my CD in 2006. It arrived in late 2006 as far as I can recall. It was a brown envelope, stamped from abroad, but redistributed by a local agency.

I tried to find a computer to run Ubuntu, but I got no opportunity. My college exams were closer at the time, so no friend wanted to get along on a new adventure that could destroy their running Windows PC. The private computer training center had closed already. I could only keep the CD in my collection, but I was sure of one

thing. This disk had genuine software, and I wanted genuine software only to run on my computer because \$1 USD Windows OS CDs never looked genuine to me, even if they had keys included.



HOW-TO

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.



REVIEW

Written by Adam Hunt

Kubuntu 23.04

Released on 20 April, 2023, Kubuntu 23.04 is the second of three planned interim releases on the way to the next Long Term Support (LTS) version. That release, Kubuntu 24.04 LTS, is due out in April, 2024.

This is Kubuntu's 37th overall release and the 16th one with the Qt-based Plasma 5 desktop. This means it is a very mature project with a solid history and a solid fanbase too.

As an interim release, Kubuntu 23.04 will be supported for nine months, until January, 2024.

INSTALLATION

I downloaded the rather hefty 5.0 GB ISO file for Kubuntu 23.04 from the official source via BitTorrent. The file is not only slightly bigger than Ubuntu 23.04's 4.9 GB, but it is also 16% bigger than the last Kubuntu release. Both Kubuntu and Ubuntu seem to be quickly growing in size, for reasons that have not been well explained.

I did an SHA256 sum check on the ISO from the command-line, and it tested as correct.

I dropped the Kubuntu 23.04 ISO file onto a USB drive equipped with Ventoy 1.10.91 and it quickly booted up to the new Kubuntu 23.04 desktop.

SYSTEM REQUIREMENTS

The recommended minimum system requirements for Kubuntu 23.04 are the same as for Ubuntu

and have not changed for this release:

2 GHz dual-core processor

4 GB RAM

25 GB of hard-drive, USB stick, memory card or external drive space

Screen capable of 1024x768 pixel screen resolution

Either a CD/DVD drive or a USB port for the installation media
Internet access is useful but not essential

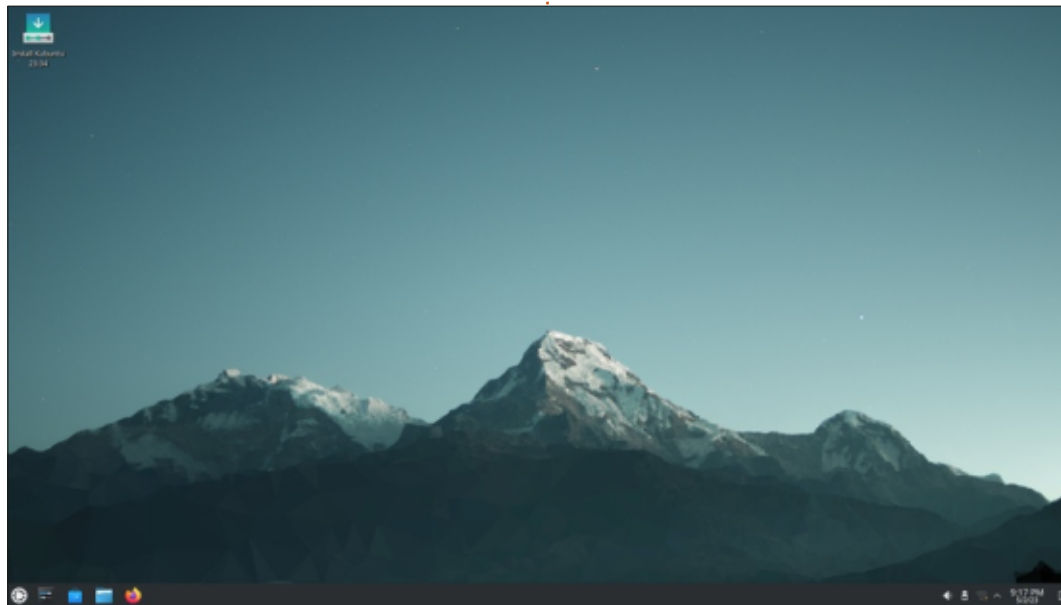
The recommended 4 GB of RAM is probably getting a bit light these days given how much is eaten up by

web browsing, so 8 GB is probably a more realistic amount for decent performance.

NEW

This Kubuntu release has moved up to the Qt 5.15.8 toolkit and the updated KDE Plasma 5.27 desktop. It has KDE Frameworks 5.104 and applications from KDE Gear 22.12 which brings a number of minor refinements. The Plasma 5.27 desktop features a new "Konqi-powered" wizard to guide users through their desktop setup, a window tiling system and new application themes, as well as improvements to the included tools and widgets.

While mainstream Ubuntu has been using Wayland by default since Ubuntu 21.10, Kubuntu still isn't there yet. Like every release since Kubuntu 21.04, Kubuntu 23.04 continues to offer an optional Plasma Wayland session available at boot-up, but it carries a warning that it "is available for testing, but is not supported."



SETTINGS

Every release of Kubuntu gets a fresh wallpaper design and 23.04 is no different. While some flavors, like Ubuntu, Ubuntu Unity, and Lubuntu, have new “lobster-themed” wallpaper after the codename for this release, Lunar Lobster, Kubuntu’s new wallpaper is entitled Mountain by Andy Betts. Not only has Kubuntu eschewed the lobster cliché, this wallpaper differs from other recent Kubuntu ones in not being a geometric design but a photo-like mountain range. It even automatically switches to a toned down version when a dark theme is selected. If

this isn’t to your taste then there are 35 other wallpapers provided, including many from older releases, although none with lobsters.

Kubuntu has always had the widest variety of user settings of any Ubuntu-based distribution which allows for lots of customization. I have always thought it is the degree to which Kubuntu can be personalized that accounts for much of its user appeal. Kubuntu is easy to make it feel like it is “yours”. In contrast, Ubuntu has very little in the way of user customization which makes it feel more antiseptic.

As in the other recent releases, Kubuntu 23.04 has four global themes, four application styles, six

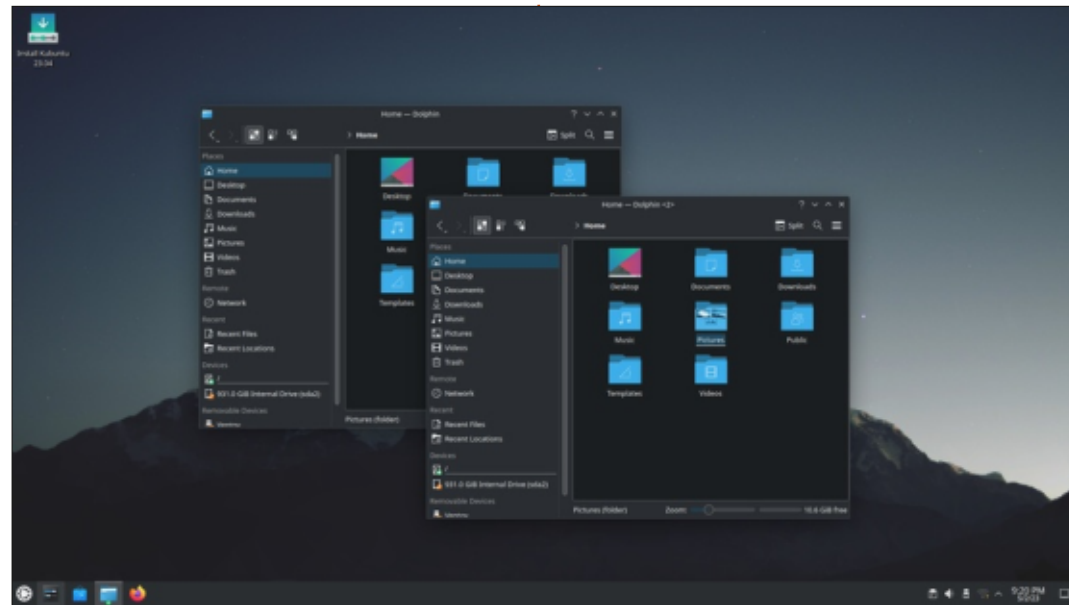
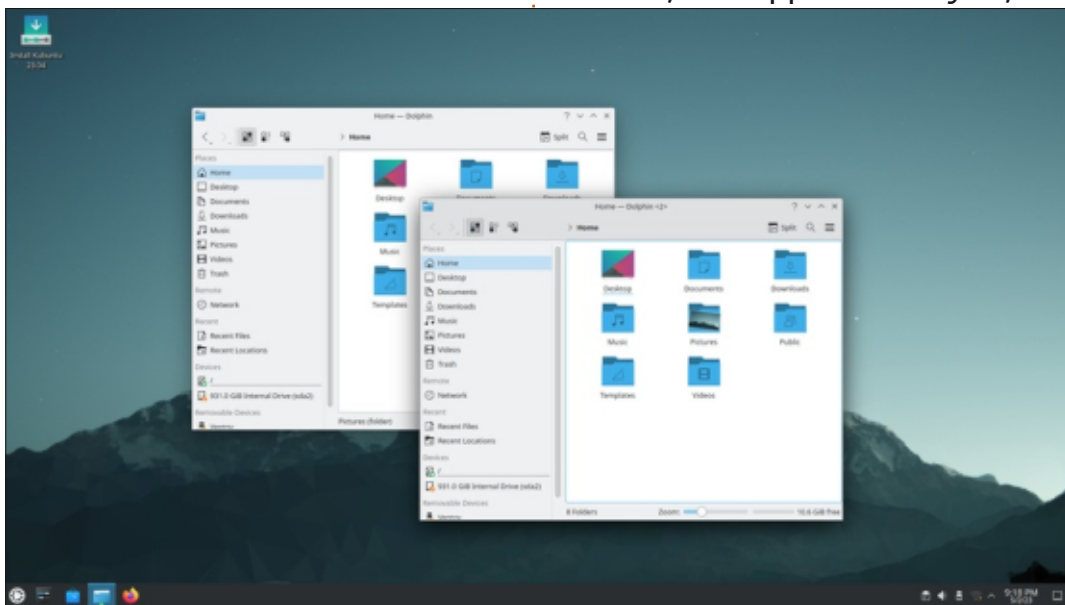
Plasma styles, five window colors, two window decoration styles, eight icon sets, and eight cursor styles. Of course, these are just the options that are preinstalled, as most of the individual setting pages have one-button downloads at the bottom for many more.

As in past versions, the KDE desktop panel can be relocated to anywhere on the screen including left, right, top, or bottom, or even in the middle, if you like. By default, it is found on the bottom. When it is moved, it reorients its icons on the fly. The panel can also be made wider or narrower and the icons will resize themselves automatically to fit.

As in the previous release, this one has 68 widgets available, plus many more that can be downloaded for use. These are small programs that add functionality to the desktop with things like clocks, CPU monitors, and calendars. You’re limited only by your screen size and your personal tolerance for clutter.

APPLICATIONS

Some of the applications included with Kubuntu 23.04 are:
Ark 22.12.3 archive manager
Discover 2.1.2 software store*
Dolphin 22.12.3 file manager
Elisa 22.12.3 music player
Firefox 111.0.1 web browser**



REVIEW

Gwenview 22.12.3 image viewer
Haruna 0.10.3 video player
Kate 22.12.3 text editor
Kcalc 22.12.3 calculator
KDE Partition Manager 22.12.3
partition editor
Konsole 22.12.3 terminal emulator
Kmahjongg 22.12.3 game
Kmines 22.12.3 game
Konversation 22.12.3 IRC client
Kpatience 22.12.3 game
Ksudoku 22.12.3 game
Ktorrent 22.12.3 BitTorrent client
LibreOffice 7.5.2 office suite, less
only LibreOffice Base database
Muon 5.8.0 package manager*
Okular 22.12.3 PDF viewer
PipeWire 0.3.65 audio controller
Plasma System Monitor 5.27.4
system monitor
Skanlite 22.12.3 scanning utility

Spectacle 22.12.3 screenshot tool
Startup Disk Creator 0.3.16 (usb-
creator-kde) USB ISO writer
Thunderbird 102.10.0 email client

* indicates same application
version as used in Kubuntu 22.10
** supplied as a snap so version
depends on the upstream package
manager

As can be seen by the lack of
asterisks, most of the applications
included are updated application
versions from KDE Gear 22.12.

There is actually one
unadvertised change to the list of
default applications in this release,
the Haruna video player has
replaced the VLC media player.

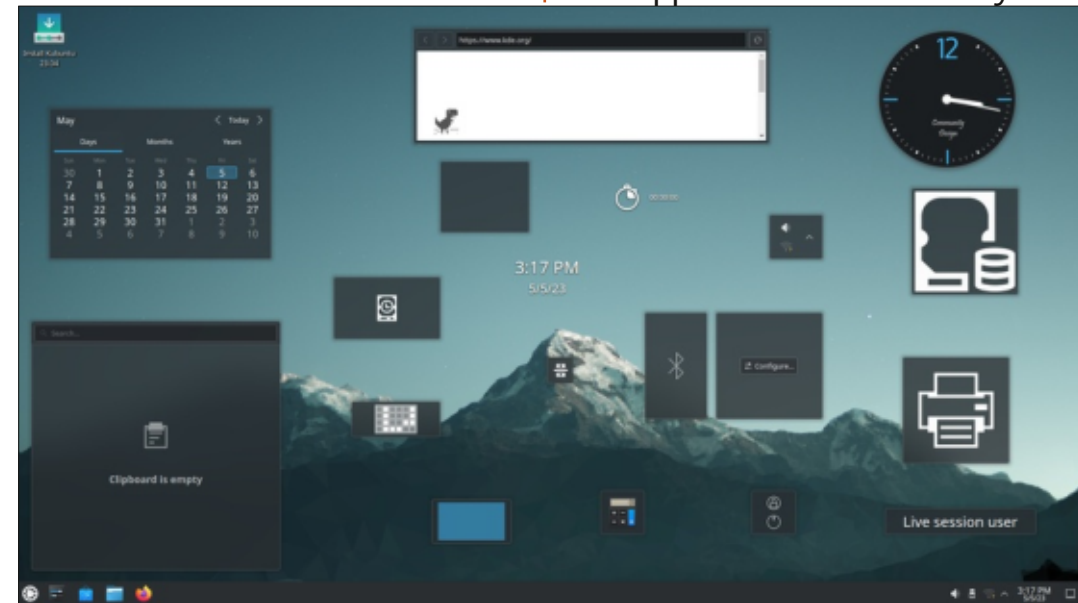
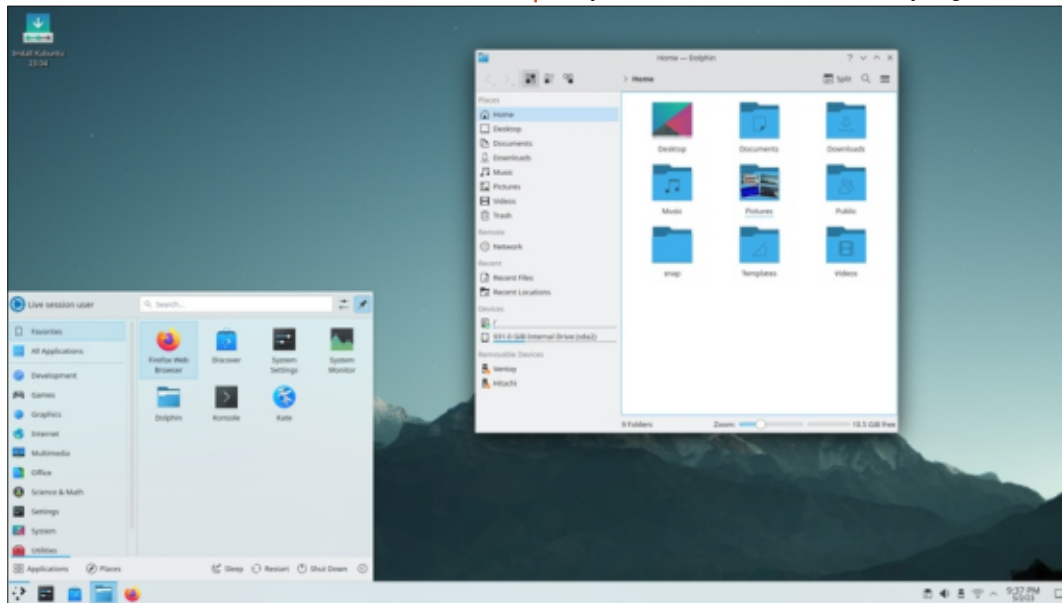
Neither the release announcement,
nor notes, explain why these were
swapped, but I do note that Haruna
is part of KDE Gear, and therefore
more integrated into the KDE
desktop, whereas VLC is an
independent project.

Like Ubuntu 23.04 and all its
flavors, starting with this release,
Kubuntu no longer offers the
ability to install applications from
Flatpak by default. The effort now
is on providing debs and snaps, and
all the behind-the-scenes support
for those formats, as per the new
Ubuntu policy. That said, the
Kubuntu 23.04 release notes have
detailed instructions on how to
install Flatpak, if applications from
there are desired.

As in past releases, Kubuntu
23.04 does not include a webcam
application, an image editor, CD/
DVD burner, or video editor, by
default, although there are many
options in the repositories, if
needed. KDE's Qt-based Kdenlive
remains probably the best choice in
a video editor.

LibreOffice 7.5.2 is once again
supplied complete, except for the
database application, LibreOffice
Base. Base is probably the least
used component of LibreOffice,
but it can be found in the
repositories, if required.

The Gear 22.12 versions of the
KDE applications include many



REVIEW

small improvements. These include the addition of remote permissions management of Samba shares from the Dolphin 22.12.3 file manager, as well as a new file "Selection Mode" opened by hitting the spacebar, allowing clicking-to-select files. The Gwenview 22.12.3 image viewer now has photo preview adjustment for brightness, contrast, and gamma, plus the Kate 22.12.3 text editor includes a new welcome screen and keyboard macro tool.

CONCLUSIONS

Kubuntu 23.04 is a highly refined and pretty much flawless release, with only a few small

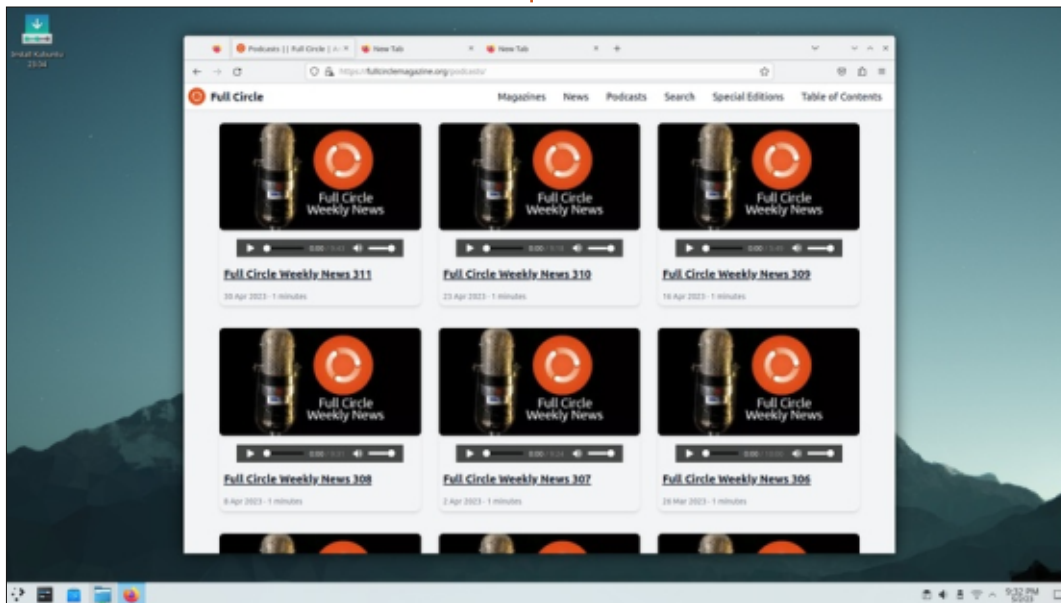
changes over the last Kubuntu version. All of this suggests that only incremental changes will be included in the next release, Kubuntu 23.10, due out on 12 October, 2023. Similarly, I expect the next LTS, Kubuntu 24.04 LTS, will not be dramatically different either, when it comes out in April, 2024. This is a good thing as, with a well-polished operating system like Kubuntu, only small advancements are needed.

EXTERNAL LINKS

Official website:
<https://kubuntu.org/>



Adam Hunt started using Ubuntu in 2007 and has used Lubuntu since 2010. He lives in Ottawa, Ontario, Canada, in a house with no Windows.

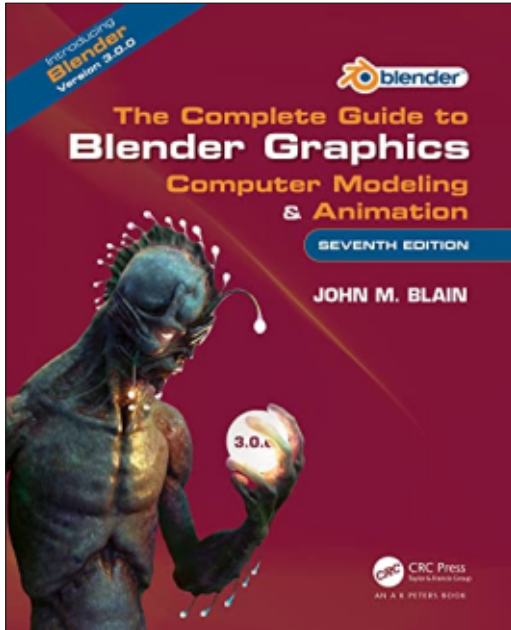




BOOK REVIEW

Written by Erik

Complete Guide to Blender



The Complete Guide to Blender Graphics Computer Modeling & Animation SEVENTH EDITION

Author: John M. Blain

Website: <https://www.taylorfrancis.com/books/mono/10.1201/9781003226420/complete-guide-blender-graphics-john-blain>

Price: \$62.25 USD (Amazon)

Blurb: "The Complete Guide to Blender Graphics: Computer Modeling and Animation, Seventh Edition, is a unified manual describing the operation of the program with reference to the Graphical User Interface for Blender Version 3.0.0, including nearly 100 pages of completely new content.

The book introduces the program's Graphical User Interface, and shows how to implement tools for modeling and animating characters and creating scenes with the application of color, texture, and special lighting effects."

First of all, thanks to Gary for the copy, it was another story getting a 500MB epub-file to load!! I had to install FBReader to open it.

When they say the book introduces the GUI, they don't kid around – they explain what is meant by words like buttons and sliders, moving on to using the window buttons, like close... "The

book provides instructions for New Users starting at the very beginning." I thought this was a bit much as in the beginning they have an assumption that you would know how to operate a computer. The explanation of the terms used in blender-speak, or should that be graphic design-speak, was top notch. Should any new user feel lost, they can always come back to the beginning of the book and look up a term.

Things are not only displayed, but highlighted in the accompanying screenshots to rule out any confusion. To understand the author's thoroughness, the first three chapters of the book are simply explaining what you see on the screen when you open blender for the first time.

Then, chapter four takes a bit of a jump explaining things like linear measurements and angular measurements. This thorough explanation continues in chapter five where things like vertices, and edges, and faces, are explained. In none of these chapters do you

make anything, but I would recommend this book to anyone that has no idea what it all means and how things slot together (me, I mean!). I picked up blender from a text document and messed around, as, at that time, I had shoddy 3G internet at like 20kb/s, so things like Youtube were out of the question. Well, the size of this e-book alone would also have been out of the question, but it is one of those things you wish you had when you started.

This book is chock full of illustrations, almost for every half a page of writing, there is an image. This is by no means a bad thing and really makes this something you need to have around. I wish I could lay my hands on a hard copy, but no bookshop has it and the ones offering to import it charge well over \$100 USD, which is ridiculous. Imagine having to pay 2000 EUR for a book, well, guess I'm not buying one. Man, I really miss those pop-up book stores that would sell books by weight!

If you are at all interested in

REVIEW

Blender, and you need a dummies guide (this out-dummies the dummies book by far!), this is your book. Trust me, now I also know what a boolean modifier is and it is not what I thought it was. Once my shift work is over, I will definitely transfer some of this knowledge when I continue our “make something with blender” tutorials

and I am sure it will speed up my workflow when I know what I am looking for , instead of me trying to find the thing I don't know the name of in: <https://docs.blender.org/manual/en/3.4/index.html>

Do you disagree: misc@fullcirclemagazine.org

There are numerous Editors for selection depending on the particular feature of Blender you wish to use. The different Editors will be introduced as features of the program are encountered.

Editors interrelate since what occurs in one will affect what occurs in another. This interaction between Editors is particularly evident when considering the relationship between the 3D Viewport Editor and the Properties Editor. Editing data (entering or modifying) in the Properties Editor affects what is displayed in the 3D Viewport Editor.

Data is entered or modified by activating (clicking) controls in an Editor.

Controls - Buttons, Icons and Sliders

Each Editor in the GUI is a separate panel comprising a Header at the top of the panel and sub-panels which display within the Editor. The Header and sub-panels contain buttons which you click to activate functions or display menus for selecting functions. The buttons are displayed as text annotation, icons and panels. Each of these, relay data to the program to perform an action.

Example 1 : The 3D Viewport Editor (the default Screen display – Upper LH Side)



Note: The buttons shown in the diagram can be seen in the panel at the upper left hand side of the default Blender Screen arrangement. A detailed description of the Screen Arrangement with its Editors and panels constituting Blender's GUI (Graphical User Interface) is presented in Chapter 1.

Note: In giving instructions, **Default** means, that which is displayed on the computer Screen before any action is taken.



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.



Join us on:

-  facebook.com/fullcirclemagazine
-  twitter.com/#!/fullcirclemag
-  linkedin.com/company/full-circle-magazine
-  ubuntuforums.org/forumdisplay.php?f=270

JUST SOME STUFF

I have been reading your magazine on and off since 2011. I never got a chance to write to you, and I really feel bad about it.

While reading the May issue, I realised it was time to send you a letter because I used to enjoy writing letters to magazines in high school, and that was a wonderful time. At first, I noticed you made a mistake on the front page. The front page was supposed to say issue #193 - May 2023, but instead of May, you printed April.

In the last several issues, the number of news items has grown so much that they cover almost one-third of the magazine's pages. Who likes so much news?

What I like the most in FCM is the Q&A section, but I wonder why you don't publish the names of readers who ask questions. It would be great if you publish their names and locations. I really appreciate Erik's answers to all those questions.

This time I had the opportunity to read Micro This and Micro That one more time, and I would ask Greg to never stop writing, please. I had a lot of trouble with I2C when I was finishing my MSc back in 2012. His article in this issue [FCM#193] really motivated me to tinker with my RaspberryPi 3B. I wish him good health and happiness.

I also loved to read two sections that are usually empty these days. One is My Opinion and the other is My Story. How about I write my story and share my opinion of software for electronics engineers? I won't call myself a great writer, but I'll try my best to write well.

In FCM#193, a fellow FCM reader, Jossy, wanted to know about good PCB development software. It looks like he does not know about KiCad. How about I introduce him to my favourite PCB Design Software, KiCad, through FCM?

This was all for now. I'll write you a letter next month. I'm still

reading the current issue. I wish everybody involved with FCM great health and utmost happiness.

Abdul-Jabbar Bozdar

Ronnie says: *thank you for writing in, and for the kind words! Let me address a couple of your points: yes, guilty as charged with the cover error, but I did catch it (thanks to reader Dale who emailed me), and uploaded an updated PDF. Regarding the Q&A section: sadly, the vast majority of our readers never write in. Erik uses a pool of questions from various sources which sometimes requires him to remove names, etc.. Again, with readers not writing in, I've no articles left to fill My Story or My Opinion sections. I keep asking, but never get any submissions.*

Speaking of submissions: yes, feel free to send over your story (see My Story this issue) and/or opinions, and an article or two on KiCAD would also be most welcome.

Thanks again, and I look forward to your next letter.

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.



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, UUIDs, 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 recently overheard someone singing to a song by coldplay - 'vida la vida' with misheard lyrics, "be my river"... (not the only mistake, but anyway). I corrected her, not being nasty, just pointing out that it is misheard lyrics. She looked me straight in the eyes, saying that she knew, but she preferred it her way. This started me thinking about one of the Arch Linux distributions by Eric du Bois that imitated a bunch of others with a button press, as the changes

were all cosmetic. This, in turn, sent me down a rabbit-hole of how Windows and MacOS don't really allow you to make your PERSONAL computer personal. (Yes there are ways, but you need to patch files, etc.) At the end of the day, we all prefer something our own way, and as much as 'they' (manufacturers) want to cookie cutter us, we still find a way - remember beige box PC's then black box PC's? Ever spray paint yours? On Linux, you can still have your own fashion statement, even if it is made up of other parts. By this I mean, you can grab icons and window manager themes and wallpapers and fonts and buttons and positions to make your desktop or laptop personal. This is also going away. I ran into two distro's that did not really allow this, and then the rabbit-hole went deeper - everyone is aiming for static base systems now, Ubuntu included. When you have an immutable base system, will Linux be relegated to Windows or MacOS sadness of allowing you to change only the wallpaper? Hell, I remember making icons for a distribution and really enjoying

myself. Is this pleasure soon to be denied too? Let us know your thoughts on: misc@fullcirclemagazine.org

Q : I inherited a Supermicro server from an uncle and I wanted to dual-boot with Ubuntu desktop and Windows 10. I can install Ubuntu, no problem, but when it comes to Windows, it seems to hang, then reboot. I have tried different Windows images and jump drives, and even having Ubuntu on a separate drive. I made a mistake in the beginning installing Ubuntu first, but I know now it has to be second, but now Windows is bent.

A : Yes, the issue is squarely on Windows' shoulders. Windows 10 does not load on server hardware, it has nothing to do with Ubuntu.

Q : I'm doing a course on Udemy that wants me to have a Lubuntu VM. I load it in hyperV and

all goes well until I boot the VM, then it stops responding. What could I be doing wrong? My settings are - <removed>

A : Load the Lubuntu VM in Virtualbox. If it works there, the issue is hyperV and you need to ask the question in their forums.

Q : I am using Ubuntu server and I keep seeing: "A fatal error has been detected by the Java Runtime Environment:" I am using the Ubuntu version - 17.0.7+7-Ubuntu-0ubuntu122.04.2. Should I use an official Java version instead?

A : There was a bug in that version, when you update to 18, your problem should resolve itself.

Q : I have a very long password that I need to see. My keyboard is not the best, sometimes typing, sometimes not. Problem is that, in MATE terminal, the password does not show up. How can I enable it please?

A: You need to add "pwfeedback" to your sudoers file. Here is a nice tutorial: <https://mfcallahan.blog/2022/01/04/enable-pwfeedback-and-show-asterisks-for-password-input-when-using-sudo/>

Q: When I open a meeting on google meet, it will not detect my camera. It used to work, but now it does not. I don't know how to troubleshoot it. I am using Xubuntu 22.04. <removed>

A: This is browser related (I had the same issue). I cannot tell you precisely what, only that it is. I suspect some addon or privacy setting on my side, but I just grabbed a fresh copy of Brave and pasted the link and it worked for me. This is the only reason I know what you are talking about. Google likes to fingerprint you, so if you run fingerprint blockers and canvas blockers, etc, you may have to uninstall them one-by-one until it works.

Q: I installed Proxmox VE on an old server, running 8 1TB disks

in hardware RAID. Everything goes well and I install Ubuntu 22.04 and set it up the way I like it. Then, tomorrow I find that it somehow crashed my raid array and everything is junk. I'm not sure what happens when I leave it on overnight, but if I do, it bombs out. I'm using the free version of Proxmox and the flagship version of Ubuntu. I also tried Kubuntu, but the same results.

A: Proxmox uses ZFS if I recall correctly, ZFS wants to manage all aspects of your disks, and does not play well with certain hardware RAID cards. It has nothing to do with Ubuntu, It is simply the act of loading some machines on it (Data). Have your data drives not* in hardware raid and you should be good.

Q: I am running a Nvidia 3050ti and I would like to see the usage. The built-in system monitor seems not to have that function. I can see it when I boot to windows in the task manager, what do I need to install on Ubuntu side to see something similar? Dual booting Ubuntu 22.04.

A: The nice thing about Ubuntu is that you have a choice. You can use a different system monitor, I would suggest: <https://github.com/hakandundar34coding/system-monitoring-center>

Q: My Ubuntu was taking forever to start, so I ran systemd-analyze blame and the issue seems to be the 44 loop devices that each take like 10s to start. How do I know what they are?

A: Snaps probably. Uninstall as many as you can and see what happens?

Q: Ubuntu is so confusing, I never know which package I need to download to work, so I just grab them all and install until one works. I just don't understand what to use. How do you tell? Like this; <removed>

A: Rule of thumb, if you have a 64-bit machine, install x64; if you have a 32-bit machine, install x86. If you have a PC or laptop, AMD64 is what you are going to look out for. To be 100% sure, open a terminal and type: `dpkg --print-architecture`

See: <https://wiki.debian.org/SupportedArchitectures>

Q: There's this scenario, where I need to use a jump host to ssh into some other machines that may or may not be containers. How can I see if they have an external IP?

A: You can try and run:

```
ip a sh | grep inet
```

from that machine / container.

Q: Why does my browser in Ubuntu say "XDG OPEN" when I try to download a torrent file for an Ubuntu flavour? Seems fishy to me.

A: It means Ubuntu is smart enough to realize you want to torrent something and will open your torrent client for you. There is nothing sinister about it, you just need to choose the correct client, like say Vuze or whatever you are using.

Q: I am using Ubuntu with a Dell WD19TPS dock on my Dell

Q&A

7400. If I start my computer with my dock attached, and my USB headphones, and the keyboard's dongle in the dock, it works fine. If I disconnect the dock and connect it again, it does not see the aforementioned dongle or headphones. I look in dmesg and nothing? Only thing that solves it, is a reboot.

A: This could be a number of things, but I would start by updating the dock firmware (I checked and there is a recent update) as well as the laptop firmware. If that does not work, try that same routine with a live CD and see if UBUNTU is to blame? (You may need a Windows computer to update the dock's firmware).

Q: I have an HP color laserjet pro M479 that will not print in Ubuntu. It is a replacement model for the model I have, which worked without issues or drivers. Now I do not seem to get any joy. I have heard that cups is now unsupported?

A: Simply install HP lip. See here for an explanation: <https://ubuntuhandbook.org/index.php/2023/06/quick-fix-hp-printer-ubuntu-2204/>



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.



TABLETOP UBUNTU

Written by Josh Hertel



BACK NEXT MONTH



Josh Hertel is a husband, father, mathematics educator, tabletop gamer, techie, and geek.

<https://twitter.com/hertelj1>



Price: £16

Steam: <https://store.steampowered.com/app/1390190/APICO/>

From the Steam page: "APICO is a laid-back beekeeping sim game about breeding, collecting, & conserving bees! Set in a series of lush environments, APICO uniquely combines resource gathering, biology, and beekeeping minigames, taking ideas from a mix of real-life and fantasy apiculture & floriculture."

The best way I can describe APICO is to think of Stardew Valley – but with bees. You start the game with a brief story about visiting your grandma on an island. You arrive on the island, and upon visiting gran, you're given your grandpa's beekeeping book. This is, effectively, your game tutorial.

TUTORIAL

Your grandpa's book guides you through the basics of crafting (cutting down trees, getting planks, etc), and building (floor tiles, walls, etc). Once you've got

the tools of the trade, you can start collecting bees and building hives. Adding bees to the hive gets you a queen who will make you some honey and output a couple of new bees (bay-bees?). You'll also need to make sure you have suitable flowers surrounding your hives to get that precious honey that you can also sell.

From there, you can start delving into storing resin, waterproofing your wood (to create better tools), building apiaries (to crossbreed bees), and, of course, buying and selling stuff

from various characters in the game.

There's even butterflies to turn your hand to!

CONCLUSION

I love this game. There are no time limits. No brick walls (things you must do before you can continue). No complex tasks. It's chill. Chill all the way. You can easily load it up and tinker for 10 or 15 minutes. Hours even. There's always something to do. I end up spending a good hour or two going



UBUNTU GAMES

round my hives, gathering the honey, restocking the hives with bees (to get more honey), collecting the racks from my apiaries, restocking with bees (checking their stats to try and up them), then getting the goods from the racks, putting new racks into the apiaries... you get the idea.

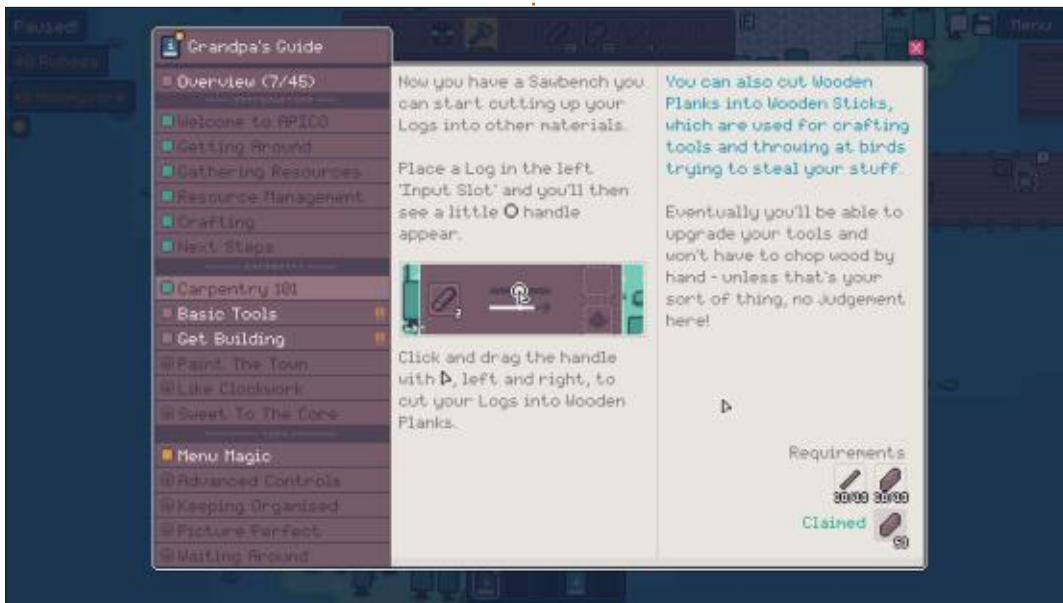
There's a lot to do in this seemingly simple game. It's not just breeding bees. It's also about getting them levelled up to get more goods from them. Then there's flowers. Same idea. You need the right flowers for the bees. You can crossbreed flowers too.

And I've not even touched on butterflies yet!

I have to drop a star from it due to some of the tasks being a bit badly worded or badly explained. At the moment, I'll be damned if I can get some bees to crossbreed. I'm not sure if I'm breeding them wrongly (going from A>B rather than B>A)? But it's a minor minor nitpick in this relaxing game.



Ronnie is the founder of Full Circle and, somehow, still editing this thing. He also paints, draws and does woodcarving in his spare time.





PATRONS

MONTHLY PATRONS

Alex Crabtree
 Alex Popescu
 Andy Garay
 Bill Berninghausen
 Bob C
 Brian Bogdan
 CBinMV
 Darren
 Dennis Mack
 Devin McPherson
 Doug Bruce
 Elizabeth K. Joseph
 Eric Meddleton
 Gary Campbell
 George Smith
 Henry D Mills
 Hugo Sutherland
 Jack
 Jack Hamm
 Jason D. Moss
 Joao Cantinho Lopes
 John Andrews
 John Malon
 John Prigge
 Jonathan Pienaar
 Joseph Gulizia
 JT
 Katrina
 Kevin O'Brien
 Lee Allen

Leo Paesen
 Linda P
 Mark Shuttleworth
 Moss Bliss
 Norman Phillips
 Oscar Rivera
 Paul Anderson
 Paul Readovin
 Rino Ragucci
 Rob Fitzgerald
 Robin Woodburn
 Roy Milner
 Scott Mack
 Sony Varghese
 Taylor Conroy
 Tom Bell
 Tony
 Vincent Jobard
 Volker Bradley
 William von Hagen

SINGLE DONATIONS

2023:
 Floyd Smith
 Richard Almeida
 Gavin Thompson
 Raymond Mccarthy
 Michael Grugel
 Linda Prinsen
 Thomas A Lawell
 Ronald Le Blanc

Ronald Eike
 Kenneth Martin
 Lance Jacob
 Roberto Machorro Mejia

The current site was created thanks to **Lucas Westermann** (ex-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.



<https://www.patreon.com/fullcirclemagazine>



<https://paypal.me/ronnietucker>



<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#195

Deadline:

Sunday 09th July 2023.

Release:

Friday 28th July 2023.



Full Circle Team



Editor - Ronnie Tucker

ronnie@fullcirclemagazine.org

Webmaster -

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.

Getting Full Circle Magazine:

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/>



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 - From Feb. 2023 Issuu are removing all free items with 50+ pages. So most (if not all) issues of FCM will vanish.

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