

Open Source

EVERYWHERE



Official website : www.ossec.tn

Software is just the beginning.
Open source has spread to
other disciplines, from the
hard sciences to the liberal
arts.



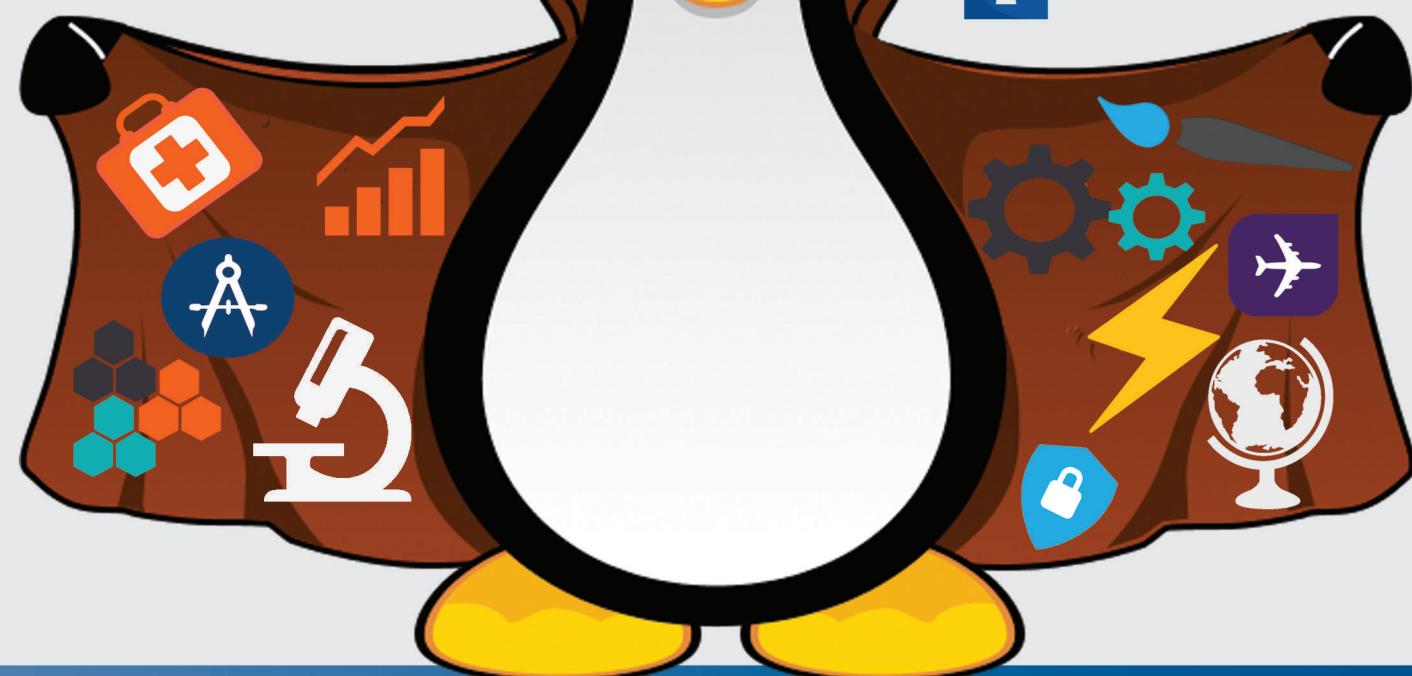
OPEN SOURCE IN
ELECTRICAL ENGINEERING
P 03



OPEN SOURCE IN THE
MEDICAL FIELD
P 08



OPEN SOURCE AND SECURITY
P 14



April - 2017



LINUS TORVALDS once said :

“

In open source, we feel strongly that to really do something well, you have to get a lot of people involved.

”

Within this spirit, we are thrilled to announce the birth of our sixth essence magazine making it a celebration of open source in every possible existent shade.

In this number, as it's entitled "[open source everywhere](#)", we would like to emphasize that the Software is just the beginning and open source is democratizing every aspect of our daily life.

We would be pleased to receive your suggestions opinions or contributions.

Your feedback is greatly appreciated.





summary

- P 03 : Open source in electrical engineering
- P 08 : Open source in the medical field
- P 13 : FOSSDAY (Event)
- P 14 : Open source and security
- P 15 : Ruby on Rails
- P 17 : Open source In administrations
- P 20 : Chris Dibona
- P 21 : Blockchain
- P 25 : Open source in telecommunication
- P 27 : Pisi Linux
- P 30 : News
- P 31 : MakerPlane
- P 33 : Open source architecture
- P 38 : Racket Lang
- P 41 : Open source hardware
- P 44 : Rubber Ducky
- P 47 : Blender
- P 51 : HACK4TN (Event)



Open source in ELECTRICAL engineering



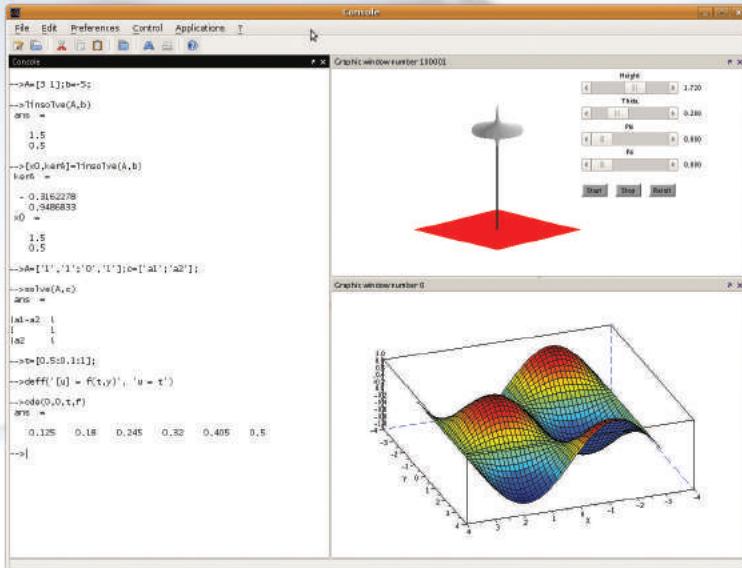
Being an Electrical engineer by trade (and heart) and a firm believer in the power of Open Source, I have often searched for alternatives to closed proprietary software right from the very early stages of my education and career.

When I first learned about open source software, and open source operating systems like Linux, I was pleasantly surprised and quite astonished by the power of collaborative development. At first there were just simple needs like what operating system to use, what software for writing documents and creating presentations, and what browser to use. But as I started specializing in my engineering department, I started searching for open solutions to proprietary software and systems - specific to various tasks in Electrical engineering.

I am happy to say that there is a lot of fruitful work going on in creating valuable and feature rich open source rivals to closed source software. This article will be a small attempt to highlight some of the most amazing open source software I have found for Electrical engineering. I personally use most of them quite frequently. Obviously, it goes without saying, this article encompasses software that I myself have found interesting and is thus by no means an exhaustive list of all the open source software available for Electrical engineers.



1. Scilab



Taken directly from the About page of Scilab:

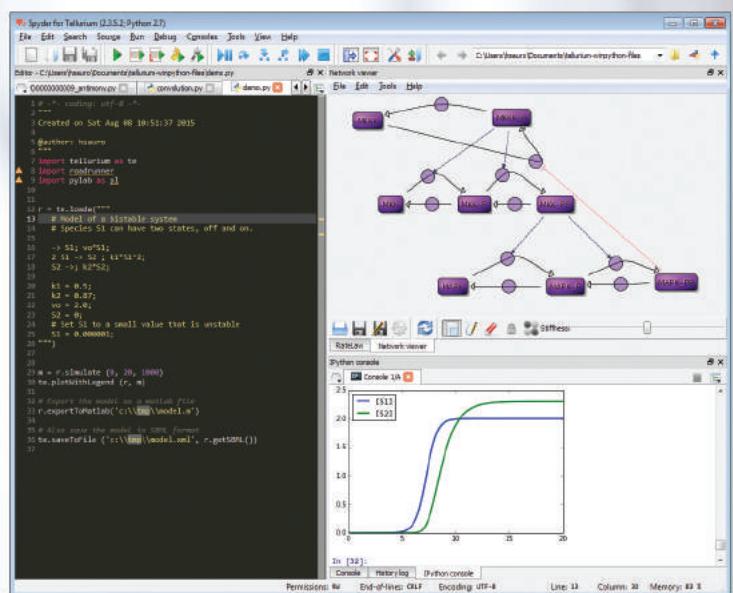
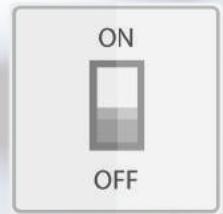
Scilab is free and open source software for numerical computation providing a powerful computing environment for engineering and scientific applications." An alternative to the commercially popular Matlab, Scilab is widely used for advanced mathematical calculations and simulation, 2- D and 3-D visualizations

(such as plotting graphs), algorithm performance optimization, statistics, control system design and analysis, signal processing, and even application development (using Scilab's inbuilt tools). It is basically the holy grail for all sorts of theoretical testing of engineering designs before trying out even a prototype of the design.

I use this fairly regularly and it is so powerful and feature rich that I keep finding new functions and features. It is backed by Scilab Enterprises which fund its development and use it to license certain software, related services and support to industries. Scilab is growing fairly well and has recently introduced a new website (scilab.io) and Scilab Cloud.

2. The WinPython Suite

The WinPython Suite is a complete open source suite for scientific computation. As its name probably implies, it is meant for Windows only. It contains the entire Python Package for Windows along with many useful libraries and bundled software (which use Python) for scientific computation such as Spyder (in picture), Jupyter Notebook and a small subset of Qt tools for creating applications if required.

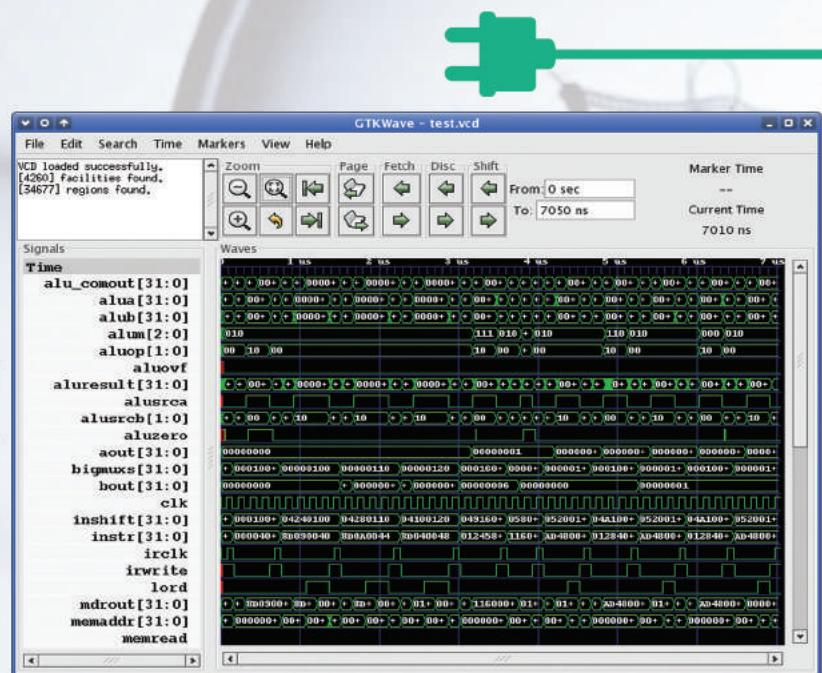


It was originally on SourceForge (which is what the first Google search might show) but has been since moved to Github ([winpython.github.io](https://github.com/winpython/winpython)). One of its biggest advantages is that it uses Python which is being increasingly used in the scientific community for its simplicity, easiness for beginners, excellent libraries and extreme flexibility. Another big advantage is that it has a portable version available which means installing on each and every computer is unnecessary. Its website even has a winpython-creator kit available to create our own versions.

For other platforms, similar suites are available (for example Anaconda available at continuum.io)

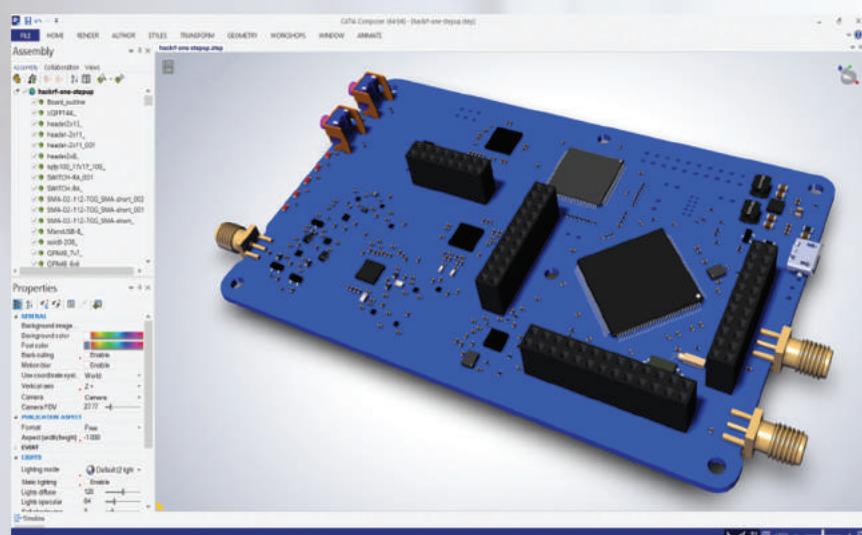
3. GTKWave

It is a full featured GTK+ based wave viewer for most popular platforms. It is quite a mature waveform viewer and supports both analog and digital signals together. However, recently I have noticed that its development has gotten a bit slow – but the current stable packages work well enough for general purposes. It is available at gtkwave.sourceforge.net.



4. KiCad

KiCad is an excellent open source PCB (Printed Circuit Board) design suite. Taken directly from their About Page on kicadpcb.org :

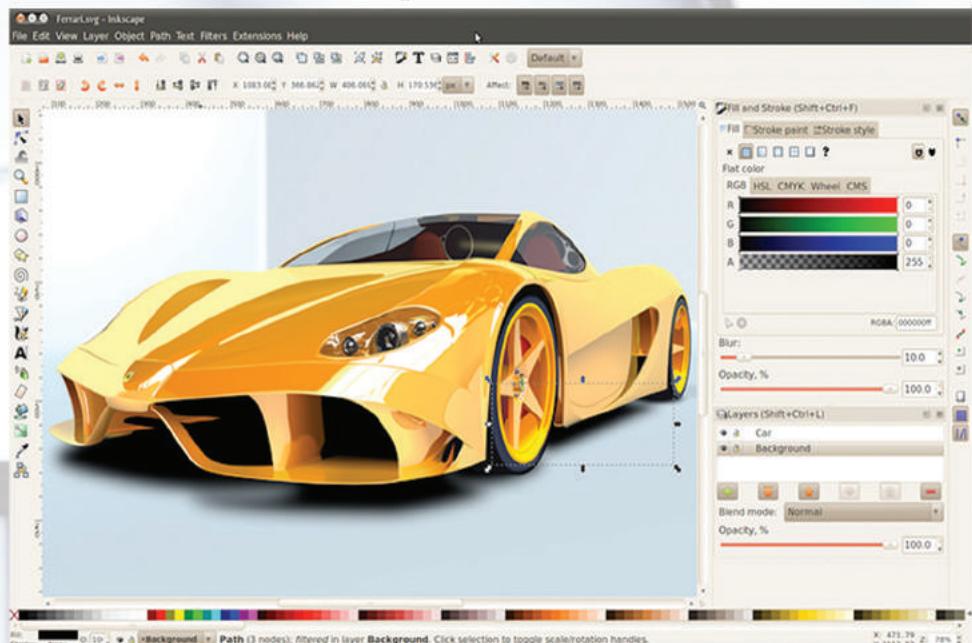


"KiCad is an open source software suite for Electronic Design Automation (EDA). The programs handle Schematic Capture, and PCB Layout with Gerber output. The suite runs on Windows, Linux and OSX and is licensed under GNU GPL v3." An interesting thing about KiCad is that it is sponsored by some very big names such as the CERN which

has contributed immensely to its development – both financially and feature-wise. I have even heard that CERN uses it for its own purposes – which is a good indicator of its reliability, industrial competence and practicality. It also means that unlike many open source projects, its development is less likely to be abandoned in the foreseeable future.

They recently had a website design overhaul as well. I have just started dabbling in it and I am liking it already. The documentation is crisp and the interface is rather easy to use. Although, personally I am new to EDA and software used for it, I do not regret dipping my toes via KiCad. Absolutely recommended.

5. Inkscape and GIMP

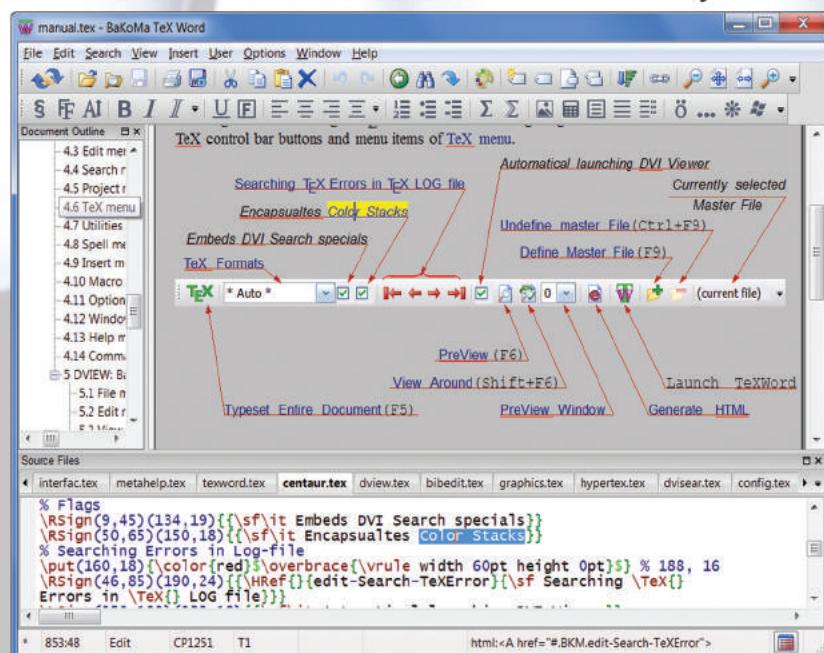


Inkscape and GIMP are excellent drawing and photo manipulation (respectively) programs for the general graphics drawing and manipulation. Whether it's drawing a block diagram, circuit schematics, or annotate plots generated with other tools, they will be more than suitable for the job and you will be delighted with the results.

Inkscape even features a flexible plugin engine and tons of plugins are available for it to provide a massive range of functionality.

6. LaTex

LaTex often needs no introduction for the usual scientific paper author but nevertheless, the introduction given on their website (latexproject.org) sums it up the best: "LaTex is document preparation system for high-quality typesetting. It is most often used for medium-to-large technical or scientific documents but it can be used for almost any form of publishing."



Personally, I think it is incredible. It renders documents and equations just the way I want without any fuss. In my opinion, it is like writing a document like a code – it may not exactly be the definition of intuitive, but if done correctly, you get exactly what you want (none of that dragging and dropping, and bad equations in Microsoft Word). For Windows, the easiest way to install LaTex is via the MikTex Suite which installs all required programs and binaries.

The above were some notable open source software in Electrical Engineering. And this list is just barely scratching the surface – the collection of open source in Electrical Engineering is, frankly, huge and it is improving and growing in leaps and bounds. These are glorious times for open source, and, in my own little way, as an Electrical Engineer, I am proud to be a part of the movement.

Swaraj Mohapatra



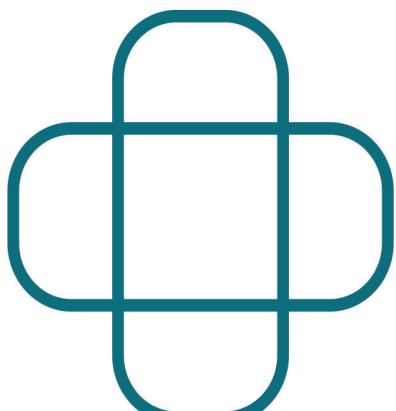
An electrical engineer with an interest in open source, technology, electronics and other things.
Graduated from the National Institute of Technology, Silchar-India.



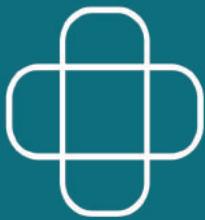
swaraj.innovate@gmail.com

BOUZIDOC

OPEN SOURCE IN THE MEDICAL FIELD



Open Source In The Medical Field

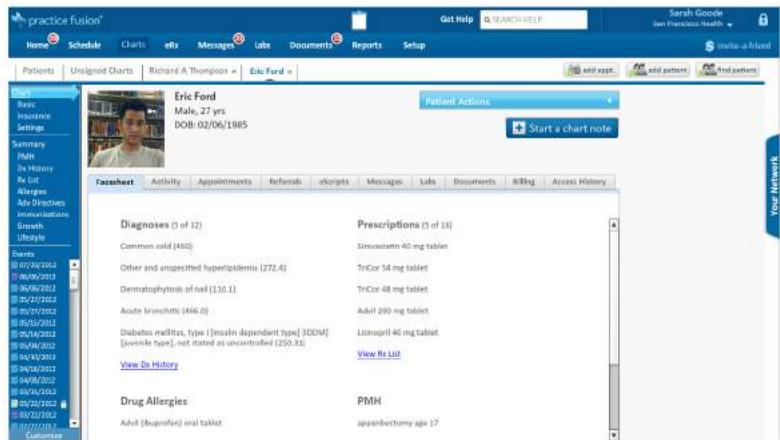


Open source is powering a revolution in medicine and health care in multiple ways. Open source software and methods make large-scale collaborative research projects feasible, multiply the brainpower applied to a project, expand the data pool, and create transparency and accountability.

This is a huge win for the Open source healthcare is forging forward quickly on the Internet. Although fast developments often produce many failures, a lot of medicinal open source projects have gained success development. This success shows that open source alone is not the solitary factor in development. Management, public relations, advancement of new treatments and cures, and cutting the costs of research. Open source practices and records software cut the costs of running medical practices, and puts practitioners in charge instead of software vendors.



TO LIMIT THIS LIST TO 5 PROJECTS MEANS THAT WE'VE TAPPED ONLY THE TIP OF THE MOUNTAIN OF OPEN SOURCE



The screenshot shows a patient profile for Eric Ford. Key details include:

- Demographics:** Male, 27 years, DOB: 02/06/1985.
- Diagnoses:** Common cold (440), Other and suspected hyperlipidemia (272.4), Dermatophytosis of nail (130.1), Acute bronchitis (446.0).
- Prescriptions:** Simvastatin 40-mg tablet, Tricor 50 mg tablet, Tricor 40 mg tablet, Adal 200 mg tablet, Lisinopril 40 mg tablet.
- PMH:** Appendectomy age 17.
- Events:** A list of recent events from 2012 to 2013.

Practice Fusion is a free web-based electronic health record (EHR) company founded in 2005, operated and privately owned by Practice Fusion, Inc. in San Francisco, California. The SaaS startup provides physicians and medical professionals with free, advertising-supported EHR and medical

practice management technology that include charting, scheduling, e-prescribing (eRx), medical billing, lab and imaging center integrations, referral letters, Meaningful Use certification training, support and a personal health record for patients.



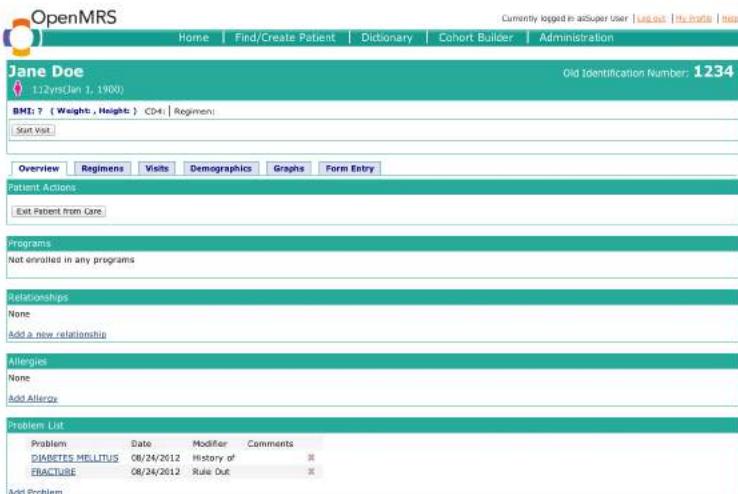
OpenMRS is a collaborative open source project to develop software to support the delivery of health care in developing countries.

OpenMRS is founded on the principles of openness and sharing of ideas, software and strategies for deployment and use.

The system is designed to be usable in very resource poor environments and can be modified with the addition of new data items, forms and reports without programming. It is intended as a platform that many organizations can adopt and modify avoiding the need to develop a system from scratch.

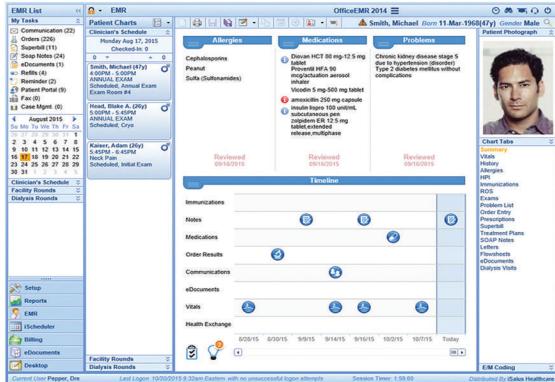
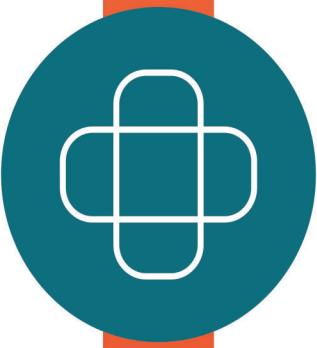
OpenMRS, Inc. is a registered non-profit that is the owner of all OpenMRS materials and the maintainer of the software's Public License. This entity will represent the OpenMRS project in legal and financial matters.

The software is licensed under the Mozilla Public License.



The screenshot shows a patient profile for Jane Doe, aged 12 years old. Key details include:

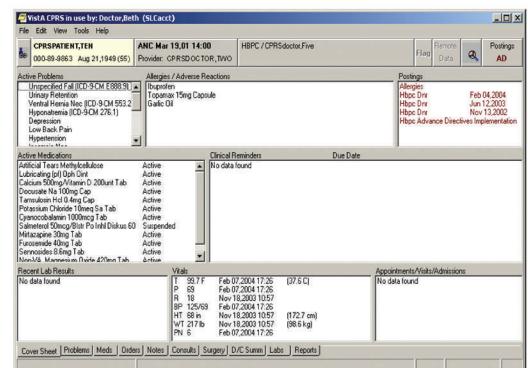
- Demographics:** Jane Doe, 12 yrs old, born Jan 1, 1999.
- Programs:** Not enrolled in any programs.
- Relationships:** None.
- Allergies:** None.
- Problems:** Diabetes Mellitus, Fracture.



iSALUS Healthcare offers a fully-integrated electronic health records (EHR), practice management and billing services solution with a host of features to support small to mid-sized practices. This system is flexible and securely accessible via the internet. iSALUS' EHR reports a first-submission claim rate of 98%, which helps decrease the time and effort spent correcting claim mistakes while also ensuring that physicians get paid for services rendered.



The Veterans Information Systems and Technology Architecture (VISTA) is a nationwide information system and Electronic Health Record (EHR) developed by the U.S. Department of Veterans Affairs (VA) throughout the U.S. to all 1200+ healthcare sites of the Veterans Health Administration (VHA).



The VHA manages the largest integrated healthcare network in the United States, providing care to over 8 million veterans, employing 180,000 medical personnel and operating 163 hospitals, over 800 clinics, and 135 nursing homes throughout the continental U.S., Alaska, and Hawaii on a single electronic healthcare information network.

VISTA consists of nearly 180 applications for clinical, financial, administrative, and infrastructure needs in VA integrated into a single, common database, permitting all VA applications to share one single, authoritative data source for all veteran-related care and services. The U.S. Congress has mandated to keep all veterans health information in one single authoritative system, and has mandated VISTA as this role.



A large, stylized illustration of a stethoscope is positioned at the top of the slide. The stethoscope has a light grey chest piece and two black tubes extending downwards. It is set against a background of horizontal red and white stripes, similar to the American flag. The brand name "BOUZIDOC" is printed in a white rectangular box on the white stripe area between the two red stripes.

BOUZIDOC

Email : contact@bouzidoc.tn

Telephone : +216 56 160 188

Site Web : www.bouzidoc.tn



FOSS day est la journée nationale des logiciels libres, organisée par G²FOSS ENIT.

Cet événement avait pour thème " l'apport de l'Open Source à l'innovation " et pour objectif de mettre l'accent sur l'importance des technologies open source à travers des activités aussi variées que fécondantes: Des conférences enrichissantes, des stands, des pauses musicales et notamment la finale du CTF "**Capture the FOSS**".

PRÉSENTATION DES CONFÉRENCES :

Trois conférences se sont tenues le jour de l'événement.

La première portait qui sur "OpenGov" était animée par monsieur Majed Khalfallah, secrétaire général de l'association des professionnels de l'Open Source, directeur central du système d'information au Sonède et secrétaire général du centre tunisien pour la démocratie et le développement.

La seconde était animée par monsieur Elyes Chater, concepteur et développeur de jeux vidéos et fondateur de K'Art Studio et avait comme sujet "La réalité augmentée".

La dernière par monsieur Wajdi Gharbi spécialiste en développement Web chez la Drupal. Les trois conférences se sont tenues dans une ambiance magnifique à peine interrompues par de brèves pauses-café.

PRÉSENTATION DE LA COMPÉTITION CTF :

G²Foss ENIT a organisé un tour préliminaire de la compétition CTF dénommée "Capture the Foss" qui s'est déclenché en ligne le 31 Janvier 2016 et a duré un mois. Cette compétition cible les étudiants intéressés par le domaine de la sécurité informatique. Plus que 100 équipes y ont participé. Les équipes participantes sont formées de 2 à 5 membres venants des écoles suivantes : ENIT, Ensi, Enicarthaghe, Insat, supcom, ensit, fst Tunis, IPEIT, ESPRIT, ISI, Iset com, ISAMM, ESSECT et Iset Rades. Les huit premières se sont qualifiées pour la finale qui s'est tenue le jour de l'événement. Cinq d'entre eux étaient présents parmi nous et l'équipe "FSOCIETY" de l'enicarthaghe a remporté le premier prix.

Open source And Security



Normally, when you think about open source and security, by reflex, the nomenclatura thinking brings questions to your mind such as : is it more or less secure to opensource my application ?

In this article, I won't be debating that. I am more a consumer or advisor on open source solutions, rather than a contributor or a developer, so I might speak about it from that prism. In fact, in my daily work, it frequently happens, that I advice activists who come to us looking for help to choose an operating system that won't spy on them or simply a calendar application that won't record all the meetings details and events of their lives in multinational datacenters.

It also happens that I direct them to the open source alternative of Skype or DropBox. Of course, in that case, you will need to anticipate scepticism about that reputation of lack of user friendliness. But, it would not be much of a game, when you convince about the respect of privacy gained when using a transparent, open, auditable and tweakable tool.

Last June, was celebrated the 3rd anniversary of Edward Snowden's revelations about the vastest spying facility established by the human being. The shockwave not only put internet governance in question but also the tools we use everyday. Some of the most famous proprietary softwares (closed boxes) were revealed "untrustworthy".

Resistance started to organize and initiatives such as Prism-Break [0] emerged to tell internet users that there are open source weapons we can build and use together to fight back.

It is not enough to just be aware that you live in a post-snowden, pre-quantum [1] era, but, everyone must act accordingly by giving a sense to his/her Homo consumericus [2] existence, especially if you are tech savvy.

Imagine if you are a nutritionist, would you eat anything ? of course not ! you would want to know what ingredients you are swallowing. Same here, for instance, when you choose to install UBlock-Origin [3] over AdBlock, you are not only getting rid of naughty ads but also promoting open source and your digital security at the same time. Same thing when you choose to communicate with your friends and family through Signal [4] over using WhatsApp or Viber.

Nowadays, many claim to be hackers : in my point of view being a Hacker doesn't necessarily mean you hack websites, or win CTFs : it is more than this, it's a state of mind, a set of intellectual reflexes that constantly push you to understand how the tools (and the hardware) works. as a result you can control it and you no longer are a simple product [5]. This is the path for you to be secure, and open source opens that door.



Mohamed Chennoufi

Mohamed is a security incident shift lead at Access Now. He is a computer science engineer. He graduated from the Institut Supérieur d'Informatique of Tunis el Manar (ISI) in Tunisia.  mohamed@accessnow.org

[0] Prism is the name of mass suvelience project, Prim Break tries to break it : <https://www.prism-break.org>

[1] Pre-quantum era:

<https://www.technologyreview.com/s/600715/nsa-says-it-must-act-now-against-the-quantum-computing-threat/>

[2] The man who is consuming : https://en.wikipedia.org/wiki/Homo_consumericus

[3] <https://github.com/gorhill/uBlock/>

[4] Signal is a peer to peer encryption partly opensource tool for mobile : <https://whispersystems.org/>

[5] You Are the Product: The Price of Free in the Growing Privacy Industry :

<https://www.goldenfrog.com/blog/price-of-free-in-online-privacy-industry>



Ruby on Rails
Sustainable productivity for web-application development

Have you already heard about web development, its techniques and its most-used technologies? Have you ever heard about ROR? Strange name, right ?

Ruby on Rails, also known as RoR and Rails is a free web framework written in Ruby. It follows the design pattern Model-View-Controller (MVC). As a framework, it provides the programmer with a structure that allows him to develop faster and more intuitively. It also adds a great level of abstraction in programming the application of a set of high-level functions and offers him the economy of writing itself most compulsory routines of a web application.

You've probably already used many of the applications that were built with Ruby on Rails such as: **Basecamp** , **GitHub** , **Shopify** , **Airbnb** , **Twitch** , **SoundCloud** , **Hulu** , **Zendesk** , **Square**, **Highrise**. Those are just some of the big names, but there are literally hundreds of thousands of applications built with the framework since its release in 2004.

Ruby on Rails is open source software, so not only is it free to use, you can also help make it better. **More than 4,200 people** already have contributed code to Rails. It's easier than you think to become one of them.

Learning to build a modern web application is daunting. Ruby on Rails makes it much easier and more fun. It includes **everything you need** to build fantastic applications, and **you can learn it** with the support of **our large, friendly community.**

Rails is based on two fundamental principles :

- Do not repeat: try as much as possible to have the elements of the application in one place. The MVC architecture and metaprogramming in Ruby can facilitate this.
- Convention over configuration: not specify configuration items if they do not meet the established conventions. Rails exploits this by providing default behavior for most of its features. This vision will be retained even after the merger with Merb and future modularization framework.

When starting a Rails project, regardless of configuration items are present. It usually contains the identifier and the access password to the database, essential in MySQL for example. However, one can start a project with nothing set up, with the default using SQLite.

Configuration files are kind of YAML.

Rails can work with WEBrick, the HTTP server included with Ruby. This method is often used for development and testing. In production, Apache, lighttpd or nginx with Mongrel or thin are recommended. It is also possible to use Unicorn, high performance in a dedicated environment. The use of FastCGI is not recommended from the 2.3 release. Indeed the initial library mod_ruby Apache has been reworked, and the interface now uses Apache Passenger10. Passenger is also available for nginx.

If you want to get started with RubyOnRails, Its really recommended to start with Michael Hartl.



Mahdi Ben Hmida



Software Engineering Student At The
National School Of Computer Science



OPEN SOURCE IN ADMINISTRATIONS

Since companies archives got filled with paper-based documents that finding a certain document became a hard and tiring job to be fulfilled computer scientists began to develop software systems to track, manage and store information and documents in easier and safer way. And so, nowadays, we have the Document Management System (DMS), named also Electronic Document Management (EDM) to facilitate and organize companies documents and files storage .Based on the concept of Open Source these software systems have various common component to get the job well done. These components are: Metadata which is stored for every document involving, for example, the date of storing.Integration that helps integrate the document Management into other applications. And so the user can retrieve documents, edit and save them back directly from and in the document management system repository. Most of the time this integration uses open standards such as ODMA, LDAP, WebDAV and SOAP. Capture which is making images of paper documents accepted and processed from scanners or multifunction printers added to other electronic documents and other computer-based files.



Validation; the Visual validation registration system and important data. Indexing that is tracking electronic documents and document identifiers and classifying document contents, which helps most retrieval. Storage, the act of storing and often managing documents linked to the place, the duration and the migration of stored documents. Retrieval of stored documents with much more flexibility. Distribution, to publish a document, the file needs to be in a format not easy to be changed. Besides, the equipment tasking the job must be endorsed and validated.

Plus the original master copy of the document is never to be published but to be archived. Security, some systems allow administrators to give access to documents to certain people. Workflow, that is a complex process that is diversified. Collaboration, saves the different markups made in a document by each individual user that took part in editing the document during the collaboration session. Versioning, is used for documents that need updating or change over time to check in or out of the document management system, and so users are allowed to retrieve previous versions and to continue work from a specific point. Searching documents using different attributes and document content. Federated search, refers to the ability of drawing results from multiple sources, or from multiple DMSes within an enterprise.

Publishing, includes different procedures of proofreading, peer or public reviewing, authorizing, printing and approving etc to guarantee prudence and logical thinking protecting the documents from random editing or theft. Reproduction, is necessary when using a system to get the needed information.



Besides there is another category of management software ERP, based on Open Source, that helps business holders to well manage various business activities of a firm like finance, product planning, marketing and sales etc. Being a box that contains all applications, ERP Open Source is a system that operates in real time providing a consistent look across many functional areas, called also modules including Financial accounting, management accounting, human resources, manufacturing, order processing, supply chain management, project management, customer relationship management, data services. Storing and organizing almost all the company's data, information and history through the ERP system can lead to better output as it can improve data security, opportunities for collaboration. ERP Open Source facilitates the business running of a society in a secure well organized space.



Wassim Chaabouni



Consultant ERP (Odoo) at Tamkeen

Technologies Saudi Arabia

Marwa Kouki



English student at faculty of
letters, arts and humanities Manouba



Chris DiBona

Open source has many champions: people who are spending their lives supporting and defending this great concept based on shared and collaborative work.

We are going to present a spokesman for open source, a person who might not be as famous as Richard Stallman and Linus Torvalds, but he is definitely a world leader in the use, creation and release of open source and free software. Our champion is Chris Dibona, the director of open source at Google. Let's take a closer look at this person's life.

Chris Dibona was born in October 1971 in Las Vegas, Nevada, US. He has always been a friend of computers and electronic devices. In fact, he wrote his first program on an Atari 800 with a basic cartridge at the age of 12. However, his love story with Open source began when he installed Linux on an old computer and has been hooked ever since.

Before joining Google Mr Dibona held different positions, where he progressed as a developer and remained a strong advocate for the open source and its future.

In 1998 he joined VA Linux as a marketing executive. VA Linux, currently known as Geeknet, is the online network for the global geek community. It aimed at integrating services of free software and open source development. During that period, he did way more than marketing. In fact, he wrote a book about open source named: *Open Sources: Voices from the Open Source Revolution*, (O'Reilly and Associates, 1999). He was also instrumental in creating the open source development network (OSDN) which is an online media network for open source developers and users.

In 2002, he left VA Linux to co-found Damage studios, a gaming development company which used open source in the development of a next generation massively multiplayer game called rekonstruction.

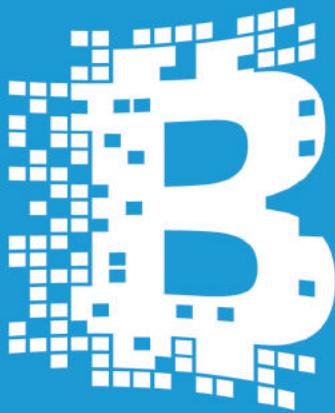
In august 2004, he joined Google, the famous American multinational technology company specialized in Internet related services and products.

Always advocate of open source software and related methodologies, he handled open source matters and policies for Google. His team supervises license compliance and supports open source developers through different programs such as the famous Google summer of code. Google summer of code is an international program that offers students aged more than 18 the opportunity to prove their coding skills in the open source community . It gives them stipends and encourages them to fund innovative projects. It actually exposes them to real-world software development scenarios and introduces them to the huge open source community where knowledge is shared and more source codes are created and released for the use and benefit of all.

Now, at the age of 45, Chris Dibona is still contributing enormously to the open source community by motivating youngsters and giving them the opportunity to contribute and to be part of such a great community.



Hanin Slimen
hanin.slimen@ensi-uma.tn



BLOCKCHAIN

Features & Platforms

Overview

Bitcoin is in its eighth year. This period seems like an eternity in IT standards but the Blockchain, the Bitcoin underlying technology, is only at its nascent stage. There is no month passing without an announcement from a startup or a some innovation Lab claiming creating the "Blockchain Platform" that will overcome some or all the Bitcoin limitations.

Each company that want to build the next Blockchain-based application must pick the right Blockchain Platform to build upon unless it is also interested in some low level cryptographic stuff.

This page focuses on scoping some existing platforms specifically for developing new Blockchain-based applications.

Blockchain Features

In assessing opportunities to evaluate the different platforms that can be suitable for potential use cases,it is helpful to have an understanding of the core Blockchain features that make it so unique.

Five key features that when combined, can create innovative platforms that we will talk about later in this paper :

- **Public vs Private Blockchain**

A **public blockchain** is a blockchain, in which there are no restrictions on reading blockchain data (which still may be encrypted) and submitting transactions for inclusion into the blockchain.

A **private blockchain** is a blockchain, in which direct access to blockchain data and submitting transactions is limited to a predefined list of entities.

● Permissionless vs Permissioned

A **permissionless blockchain** is a blockchain, in which there are no restrictions on identities of transaction processors (validators that are eligible to create blocks of transactions).

A **permissioned blockchain** is a blockchain, in which transaction processing is performed by a predefined list of entities with known identities (example Banks).

Permissioned blockchains are attractive in cases, where transaction-processing nodes need to be known to comply with regulations, as in the case of financial institutions.

Table 1: Categories of blockchains based on access to transaction processing (permissioned vs. permissionless) and access to data (public vs. private)

By access to transactions	By access to transaction processing	
	Permissioned	Permissionless
Public	Proprietary colored coins protocols	Existing cryptocurrencies (e.g., Bitcoin)
Regulated	Direct read / transaction creation access for clients (limited; leveraged by client-friendly devices and applications) and regulators	Some colored coins protocols (e.g., Colored Coins Protocol), where ability to create transactions can be regulated
Private	Access limited to transaction processors (i.e., opaque for clients); benefits of blockchain technology are diminished	Not applicable

Source : <http://bitfury.com/content/5-white-papers-research/public-vs-private-pt1-1.pdf>

● Blockchain Native Token

A unit of value that can be earned and used to create transactions on the system. If this token serves as currency in the system, we talk about cryptocurrency (like Bitcoin).

These Tokens don't exist as string literals but rather they exist conceptually as entries on a ledger (a blockchain). You can make transactions with the tokens that you own, only if you have a key (Private Key) that lets you create a new entry on the Blockchain.

In most Blockchain Platforms, tokens are used as transaction spam prevention mechanism (if all transactions cost some token, they limit the ability to spam); they are used also to incentivize Validator nodes (nodes that validate the state of the blockchain).

Some private/permissioned Blockchains are tokenless as the underlying peer to peer network is deployed in a controlled manner.

● Consensus protocol

Due to the distributed nature of the Blockchain structure, the underlying peer-to-peer network needs a consensus protocol to validate and agree on a new version of the ledger allowing the settling of transactions by incorporating them on the ever growing chain of blocks.

Bitcoin came up with a pretty genius protocol (Proof-Of-Work) which works perfectly for Bitcoin, but showed limits when applied to other domains. Therefore, proof-of-xx protocol suite appeared to accommodate different use cases (proof-of-stake, proof-of-burn, proof-of-activity ...)

● Mining

Mining is the process by which some special nodes (validators) on the peer-to-peer network get incentivized when validating blocks on the Blockchain. The incentive is the creation of a limited amount of the blockchain Native Token.

Blockchain Platforms

Giving the huge number of Blockchain platforms available, we picked a representative sample covering most of the possible use cases.

All studied platforms fall into five categories :

● Bitcoin blockchain/Non-bitcoin

These platforms leverage on the network effect of the Bitcoin Blockchain to build customized services on top of it. Services are built by adding metadata to Bitcoin transactions.

● Financial Services oriented Blockchain

Platforms that target specific FS use cases (Payment, Exchange, Remittances ...) generally geared toward transaction-based operating model.

● General purpose Blockchain Platforms

Platforms geared toward more complex business-logic schemes. Supports Smart contract executions on the Blockchain.

Smart contracts are contracts that do not require human interpretation or intervention to complete. Their settlement is done entirely by running a computer program.

Smart contracts are code-based contracts, as opposed to law-based contracts:

"Code is Law"

Possible applications of Smart contracts are:

- ✓ Escrow
- ✓ Multi-sig transactions
- ✓ Deposit
- ✓ Micropayment Channels
- ✓ Oracles services
- ✓ Decentralized applications

● Enterprise-class Blockchain

Target businesses that wish to leverage blockchain technology, but in a more controlled way. Some building blocks of the Blockchain architecture can be modified to fit business needs.

● Off-chain Blockchain Platforms

Public/Permissionless Blockchains suffer from limitations related to:

- ✓ High latency transactions
- ✓ Lack of Privacy/Anonymity
- ✓ Limited Scalability

Therefore, some platforms provide the possibility to do off-chain transactions on a Private Blockchain that are connected via a two-way peg to a main Public Blockchain (example Bitcoin). This approach helps leverage the best of the two worlds.

The Following table lists the characteristics of the different platforms that we studied :



Sami Belhadj

Project Manager at FIS GLOBAL



OPEN SOURCE TELECOMMUNICATION

"Open Source Is Eating The Software World" claimed Michael Skok, which is expected since Open Source softwares have always proved to be the best choice when compared to typical commercial products in terms of flexibility , reliability and visibility of the code.

Nowadays, the spread of Open Source philosophy has brought to light a new concept: The Open Source Telecommunication (OST) which is the use and the exploitation of Open Source solutions to create better products, more innovative, and faster to market that can achieve the degree of agility, interoperability and cost-effectiveness needed by IT manager in the communications sector, to respond to market fluctuations.

The OST companies and products can be basically organized into three categories: IP telephony, router equipment and network monitoring.

The first category is the IP telephony companies which develop and market products that deliver VoIP functionality to customers and include Digium, Fonality, Pingtel, and VoiceTronix.

The second category is the Router companies which develop and market devices or softwares that determine the next point to which a packet should be forwarded toward its destination.

Many Open Source software solutions are provided, for example, Quagga which is a routing software suite that provides implementations of different routing protocols for Unix platforms, particularly FreeBSD, Linux, Solaris and NetBSD.

furthermore, many Open Source routing platforms distributions were developed like pfSense (Router/Firewall) , Vyatta and Vyos.

Besides, other implementations of soft-switches have seen the light, which is a software that could switch IP calls in the same fashion central office equipment did for traditional phone calls.

FreeSwitch, Kamailio, OpenSIPS are a few examples of open source packages that emerged to enable the offering of telephony services over IP networks.

Plus, the Network monitoring companies providing products that either monitor a customer's network and report abnormal situations or provide a firewall to protect his/her network.



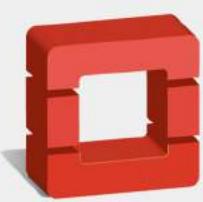
Looking into the future we can see increasing convergence and integration of the Internet and telecom infrastructures with communication services being offered in the cloud and decoupled from the underlying physical infrastructure, where new delivery and consumption models are being created not only with enterprises but telecommunication service providers as well.

Open Source has become the fulcrum for software development, and cloud services are allowing enterprises to deploy third-party solutions without having to manage or maintain them.

A famous powerful Open Source software platform for Cloud is the [OpenStack*](#).

In addition we can mention Telestax's Restcomm which is a development platform that has the promise to bridge the gap between open source communications functionality with cloud enabled services.

In the end, we can't neglect the WebRTC which is a project supported by Google, Mozilla and Opera. It's a free, open project that provides browsers and mobile applications with Real-Time Communications (RTC) capabilities via simple APIs.



openstack™

*[OpenStack](#) : is a set of software tools for building and managing cloud computing platforms for public and private clouds.

Med Amin Ben Tmim



Student at Higher School Of
Telecommunication (sup'com)



Pisi Linux

What is Pisi Linux ?

Pisi Linux is a GNU/Linux distribution forked from the old Pardus, a distribution that was developed by the Turkish National Research Institute of Electronics and Cryptology (UEKAE), an arm of the Scientific and Technological Research Council of Turkey (TÜBİTAK). PiSi (Pisi is Turkish for kitty).

The old Pardus was an original distribution because, unlike most distributions, it was not based on another distribution. Examples of original distributions are Debian, Red Hat, Gentoo and Arch Linux.

Pisi Linux is an operating system for desktop computer with software for listening to music, browsing the Internet and creating documents, etc... . Pisi Linux is built from scratch on a stable base, but many core user applications, such as the Firefox web browser Chromium or the VLC media player, are kept constantly up to date. And to increase the distribution's user friendliness, Flash player and other multimedia codecs are installed and pre-configured for immediate use.

Features :

Pisi Linux focuses on ease-of-use and stability. It is a 'Rolling-release' distribution where 32bit packages for applications such as Skype and Steam are included. There are more than 6000 packages in the Pisi Linux repositories, and more are being added by the time. Besides, Codecs for playing audio and video files are pre-installed

Unique Technologies:

- ÇOMAR /chowmar/ (COMAR (Configuration MAnageR)) : Configures and integrates installed software.

- PiSi (Packages Installed Successfully, as Intended), was built from scratch to address some shortcomings of traditional package management systems. It features a modern, simple, compact and highly efficient design. Compared to other systems, it is easy to learn building packages for yourself using Pisi. Written in Python the software uses XML for source files and incorporates both low and high-level operations (a separate tool such as APT, YUM, URPMI, is not required). Moreover it includes comprehensive CLI and GUI package management. LZMA compression and delta updates (saving time, bandwidth and space) are available too while the software utilizes Berkeley DB for fast database access.

- YALI (Yet Another Linux Installer) was designed with ease-of-use, and should not pose a problem to the novice Linux user.

The Install Routine Stages:

-Preliminaries:

Language, Licensing, Validation of installation media and Time-zone settings.

-User accounts :

Creating regular user/s and an Administrator user.

-Disk preparation

Partitioning Filesystem type, Mount-points and Swap-space.

Boot-loader.

Defaults or custom.

Actual installation.

Summary of intended installation (prior to accept and commit install to disk).

Note: as the installation proceeds, a slide-show will give you an opportunity to preview some aspects of Pisi Linux.

- Kaptan (Turkish for Captain) is a 'desktop greeter', which is automatically started post-install to assist with setting basic desktop preferences. A number of options are made conveniently available in Kaptan, including:

Mouse settings (L/R handed, single/double-click).

Desktop background or theme.

Icon theme Müdür: Designed to speed up the boot process.

PANDA

A video driver management tool (developed in-house by the Anka-Team)

Why should you use Pisi Linux?

Pisi Linux is highly recommended because it is:

* Free Software

The license does not restrict you from using, distributing or developing Pisi Linux. It is designed to guarantee your freedom.

* Multilingual

You can install and use Pisi Linux in much languages (Turkish, English, German, Dutch and more) without a problem.

* No Viruses

Surf the Internet is no longer risky and nightmarish. You won't be infected by a virus and won't be threatened to lose your work.

* Fast Installation

You'll be ready to use and work on your system in only 30 minutes of installation when you will have all needed software applications (office suite, browser, IM, and so) ready and configured for your use

* Easy to Use

It's easy to use Pisi Linux with its intuitive graphical user interfaces, menus and icons. You don't need to be a computer expert or to take a course to use Pisi Linux.

* Complete

You'll have all the needed software by default with the Pisi Linux DVD and after the Installation from the Repository.

* Easy to Personalize

You can personalize your system freely as you like. The limit is your imagination.

* Open

You can get and change the Pisi Linux source code to develop new distributions. Besides you can see all the internals by checking the source code.

* Multi-User

Every one using Pisi Linux can choose to run the system according to his or her needs.

* Fun

Kaptan for Desktop Settings, PiSi and ÇOMAR and many more keeping our company while using Pisi Linux, which makes it really funny to use and develop.

The stable Pisi Linux version is Pisi Linux 1.2, the latest version is Pisi Linux 2.0-Beta.

You can download all Pisi Linux versions here :

<https://sourceforge.net/projects/pisilinux/>

To check the Work of Pisi Linux you can take a view on the GitHub where you will find all Repositories from Pisi Linux.

<https://github.com/pisilinux>

For further information you can visit our Website:

<http://www.pisilinux.org/en>

and when you use help visit our Forum or the Webpages from groni

<http://forum.pisilinux.org/index.php>

<https://pisilinuxglobal.wordpress.com/>

<https://pisilinux2.wordpress.com/> (This is the Webpage for Pisi Linux 2.0)

If you have any suggestion to foster our system you are welcome to share your ideas and to contribute to Pisi Linux. Any kind of help is needed for our work to support Package Building, Translating, Artworks, Writings for the Wikis and Documentations and Tests for the coming Releases of Pisi Linux. Below is a link of our YouTube Channel from Pisi Linux Global to watch a short Promotion video: <https://youtu.be/yKkanZw4OTc>

Do not hesitate to contact and E-Mail us, if you have any ideas and wishe to join our team, with a short Introduction of yourself and what you have in mind for us. Each new Member is more than welcome
admin@pisilinux.org



Stefan Gronewold aka Groni

Project Manager/Coordinator at PisiLinux
,graduated from IGS Aurich West Oberstufe.
Many thanks for your interest in Pisi Linux enjoy it
to testing Pisi Linux and we hope we see you soon
 groni@pisilinux.org

news

Google centralise sur site ses projets Open Source

Google n'existerait pas sans l'Open Source et produit beaucoup de solutions dans ce domaine. En 2017, on peut recenser déjà plusieurs projets dont le code a été livré à la communauté. Il y a eu Chrome pour iOS, le partage de fichiers Upspin, le chiffrement de bout en bout E2EMail et l'encoder JPEG Gueztli.

Mais derrière ce foisonnement, il y a un hic. Trouver ces projets relève du parcours du combattant. Google s'est penché sur la question et vient de rendre sa copie en dévoilant Google Open Source Projects.

Dans un billet de blog, Will Norris, développeur au sein de la division des programmes Open Source de Google explique que « les logiciels libres et Open Source font partie de nos bases techniques et organisationnelles depuis les débuts de Google. En retour, nous avons publié des millions de lignes de code en Open Source, créé des programmes Google Summer of Code et Code-in, ainsi que le parrainage d'organisations comme la Software Freedom Conservancy ou l'Apache Software Foundation ».

Ubuntu abandonne Unity, pour revenir à Gnome

Changement de cap pour Ubuntu, qui lâche son interface Unity pour revenir à Gnome. Mark Shuttleworth exprime sa frustration, mais pense déjà à l'avenir : Cloud et IoT.

Mark Shuttleworth, le fondateur d'Ubuntu et de Canonical, vient de poster un billet qui promet de faire du bruit dans la communauté des utilisateurs de cet OS Linux. « J'écris pour vous informer que allons stopper notre investissement dans Unity8. Nous allons basculer l'environnement desktop par défaut d'Ubuntu vers Gnome pour Ubuntu 18.04 LTS. »

Fin de parcours donc pour Unity, l'interface utilisateur proposée aujourd'hui sur Ubuntu. Retour donc à Gnome 3. C'est aussi probablement la fin des velléités de Canonical dans le secteur des smartphones et tablettes. Tout ça pour ça, remarqueront les détracteurs d'Unity, qui n'a pas fait que des heureux. Chose d'autant plus vraie que Gnome était précédemment utilisé par l'OS Linux.



Quand les avions deviennent open source

L'open source touche de nos jours tous les domaines et tous les secteurs industriels et informatiques. Mais quand il s'agit du domaine aéronautique, ceci devient un peu plus impressionnant...



Le modèle du MakerPlane

MakerPlane est le premier projet d'avions open source, il est mené par John Nicol, pilote et ingénieur aéronautique et par une équipe d'ingénieurs venant des quatre coins du monde. Ce projet permettra aux gens de construire et de piloter des avions moins chers en utilisant pour les fabriquer des outils personnels et moins sophistiqués, ce qui doit créer une révolution au niveau de l'aviation puisqu'il va changer l'idée acquise par les gens : « les avions privés ne sont accessibles que par les riches ». En effet, ce projet permettra de télécharger des plans et des schémas gratuitement et de construire les pièces de l'avion à l'aide des fraiseuses à commandes numériques et des imprimantes 3D. Une démarche assez simple et assez pratique. Le coût estimé de la fabrication de cet avion ne devrait pas dépasser les 15.000\$.

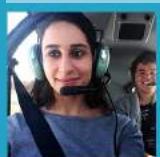
Ce qui impressionne le plus, c'est que ce genre de projet sera toujours en cours de développement vu qu'il ne dépend ni d'entreprises ni de la présence de personnes particulières. Il se développe tant qu'il y a des personnes motivées et intéressées qui prennent le relais et qui cherchent de nouvelles pistes de développement.

Ce type de projet a suscité plusieurs questions sur la sécurité et les lois régulant le décollage de ces avions. En effet, comme tous les avions, le MakerPlane serait normalement inspecté par les autorités aériennes du pays dans lequel il serait fabriqué et son utilisateur doit avoir un brevet de pilote pour cette catégorie d'avions.

Les lanceurs du projet espèrent donc que ce dernier trouve sa place aujourd'hui et que les gens du monde entier peuvent en profiter. En revanche, ce projet rencontre quelques défis notamment financiers comme la plupart des projets open source surtout avec la quantité énorme et le coût des équipements que ce projet nécessite, ce qui rend sa progression un peu lente.



Hala Sayehi



Student at ISAE Supaero Toulouse

Open Source Architecture

- What is "open source architecture"?
- What does the future hold for its embellishment across the globe and especially in Tunisia?
- Have you ever heard of open source architecture before? No?



The open source software was originated in the context of software development to designate a specific approach to creating computer programs. Over the years, the concept has been pushed way too far from its primitive form to shift through revolutionizing initiatives, products and open source projects in all domains. Years ago, architecture had made it to open source design. Today, it appears mainly on websites and smartphones as applications.

The open source architecture movement works on making architectural designs, documentations and 3D rendering accessible and available for other projects under open source licenses. Also, the concept is supposed to democratize design and architecture in a subtle way and to bring it to the people in a technical but digitally handy form. I vividly remember perceiving the idea as the new enemy of the profession. The idea of having a final project processed by several architects – who have never met before, neither were there on site – did not make sense to me back then. The interior enthusiasm I constantly carry for the uniqueness of every piece of architecture has misled to reject such an idea. But then, is architecture supposed, as a profession, to keep its embryonic practice the way it has always been – all the way through time? Technology has brought to architects new possibilities and optional freshness to the various features of the profession but the stake is real and the challenges are numerous.



To study open source architecture in an engaging understanding, I had to feature the core of the article with some concrete open source projects. There are surprisingly several OS architecture projects, but I had to put together two of today's most important projects: on an international level and one Tunisian project which is going to thrill you all.

Wikihouse: "Cities made by and for everyone."



Wikihouse is an open source construction system that proceeds using CNC machine-cut wood panels. It was founded by a UK-based architectural group. It aims to bring about digital revolution in the way we make houses and other structures.

The system has already « produced » various projects, like houses, micro-architecture, tea-pavilions and more. Wikihouse provides sketchup files along with filed instructions and documentations. There is an online **community forum** that frames the user to get additional common resources through a Google Drive shared folder. Today, the project is going international.

AN OPEN COMMUNITY CONSTRUCTION SET

www.wikihouse.cc

WikiHouse is a mass-collaborative design project. Its aim is to make it possible for almost anyone, regardless of their formal training or technical experience, to build structures which are affordable and suited to their needs. There is no single designer or architect behind the project and components are designed by an open community of designers and users for everyone's benefit.

CC-BY-SA license. All Wikihouse houses, drawings, plans and components are released under a Creative Commons Attribution-ShareAlike license. This means that you can use them, copy them, adapt them and share them freely, provided the original authors are acknowledged and a link is made back to the original source. The attribution pages go on after the license text. [View License](#). CC-BY-SA license. You are free to use it in any way you want.

Attribution

1
Download houses and components which are created and shared by an open community of users around the world. Individual components can be combined or adapted to form a structure which responds to an individual site or set of needs.

2
Click 'Make this house'. From the model, WikiHouse generates a complete set of milling drawings which can be used by a CNC cutter to fabricate the house parts.

3
The parts are cut by a CNC mill using locally-bought material. This is 18mm plywood, in the standard sheet size of 2440mm x 1220mm (8' x 4').

4
Set out the parts for each section onto the ground, assembling it like a jigsaw.

5
Wedge together the two layers to form a single section.

6
Stand the sections up vertically, positioning them approximately at 600mm intervals.

7
Fit the connector pieces into the slots in the sections. These should be staggered alternately. Use the mallet provided on the milling sheets to hammer these tightly into place.

8
Fit the internal and external cladding panels onto the structure. Internally these usually need to be screwed into place.

9
The structure is ready for insulation, cladding and services.

The philosophy behind Wikihouse is the idea of a digital 'Lego' - as it is mentioned in the brochure – which is open to be shared and written as a code. Lego, as a metaphorical formula for a handy and fun design for normal people to "craft" but within pre-set rules – which reduces the complexity that design on a professional level is about.

"Parametric design uses open data to instantly calculate cost, time, performance and impact and to produce manufacturing information." _ Wikihouse Brochure

www.wikihouse.cc/WikiHouse_Partners_2016_v1.7.1.pdf



"Reinventing our housing system" - Today, wikihouse aims to set digital tools to share new sustainable, scalable and resistant volume housing industry. The governments are increasingly familiar with this system and consider it as the next engine of sustainability and democratic development for the mass-housebuilding industry.

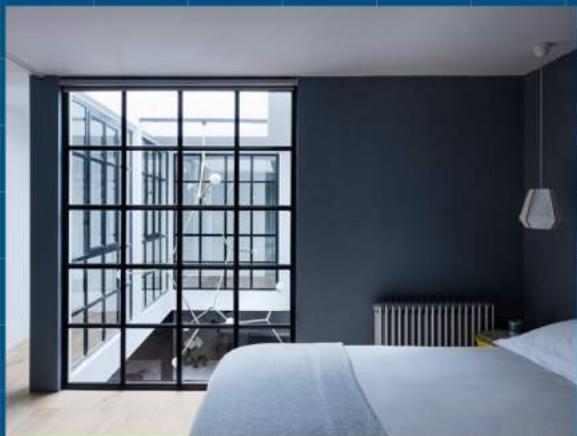
Paperhouse : "I am a cultural alloy."

www.paperhouses.co

Paperhouse issues open source home project is designed by architects from every nook and cranny around the world. The idea is to work on popularizing high-quality home designs in the suburban areas. The designs are not site-specific and can be easily customized through the use of a flexible construction system.

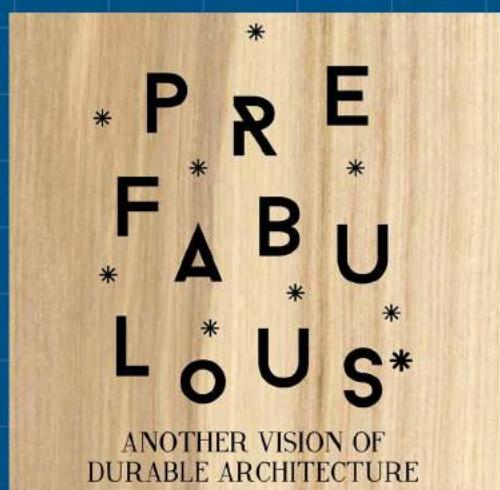
Check the showcase for more projects :

www.paperhouseproject.co.uk/showcase



Set up by Architectural Association trained Joana Pacheco, the real deal behind the idea – she says – is to put architecture within reach and engage people in their project. "The very Process" is what she believes in the most, to involve architects from different backgrounds in one project – "I contemplated many possible partners, but in the end I simply asked myself: If I were to build my house, who would I hire? I made a wish list and called everyone on it." She hopes that Paperhouse gets to the hearts of people, and then, they will process and build upon the idea, develop the plans, construe and accustom the models according to their lifestyles and territories.

Prefabulous : "A new way to live your daily life, but also your future"



Prefabulous is an altogether Tunisian start-up project, founded by Malik Nouira, a young architect, holding a master degree in architecture, who worked with renowned architecture firms and ran several important projects in different countries.

The idea behind Prefabulous is to provide modular prefabricated wooden pattern, in a way it makes it possible to build your house according to your necessities in terms of functional spaces and according to your lifestyle.

How? It's a system based on prefabricated wooden components or "modules", assigned each with a function, according to your needs : bedroom, bathroom, kitchen, living room, commercial, etc. You simply have to select the desired modules and join them the way you perceived it more practical. You can visualize the result on a 3D layout, save it and order it using a free application or on website.



The exciting part is that you can add as many modules as you need later on, you can also move it out to another city or country. Here lays the philosophy behind the idea. Our need for functional spaces mutates with time, family members' number gets bigger and the need for more room increases too.

Prefabulous offers an adaptable solution to our continuously changing lifestyle, hence our needs for space. Of course, this brave concept comes with an eco-friendly, recyclable and energy efficient characteristic.

Open source architecture is a promising field, it is emerging today faster than it was expected. Somehow, architecture found its way through this software expression, not only in what it offers for the technical designing itself but also in the very process of it and on a social scale. Henceforth, more is yet to come, people could possibly design high-quality architecture projects by means of OS architecture components.

Nevertheless, there is an ample controversy that floats around the current and future projects. Many architects find architecture ambiguous enough as a profession, and the users' tasks should be limited and should not intrude the architect's designing process and the philosophical study of the projects, since it's one of the most complex tasks of an architect. The generous "policies" that come to engage people and abridge the use are considered threatening, and can expand to damage the profession.

Many other OS projects deserve to be talked about, come what may the diversity of services or utilities the different open source fields offer, the generosity they all come up with for the people is revolutionizing the way we perceive open source software and to start reconsidering less hostility for all the forms of it.

Bibliography:

<http://www.wikihouse.cc/>

<http://paperhouses.co/about/what-we-do/>

<http://www.archdaily.com/785023/elemental-releases-plans-of-4-housing-projects-for-open-source-use>

<http://archinect.com/forum/thread/15223/open-source-architecture>

<http://paperhouses.co/open-source-heroines-joana-pacheco-founder-of-paperhouses/>

Nidhal Kmira



Architecture student at the Superior National Architecture School of Marseille (France) and the National School of Architecture and Urbanism (Tunisia)

Racket Lang



Introduction

This article is about a new programming language called Racket-lang which is gaining a lot of attention of programmers these days. You've certainly heard of JavaScript, Perl, Python, and Ruby. But Racket? Probably not. Just because it's not the most mainstream programming language doesn't mean you should discount its capabilities. Racket allows functional programming and other different paradigms that even hard-core programmers have never seen before.

What is Racket-lang

Racket (formerly named PLT Scheme) is a general purpose, multi-paradigm programming language in the Lisp/Scheme family. One of its design goals is to serve as a platform for language creation, design, and implementation. The language is used in a variety of contexts such as scripting, general-purpose programming, computer science education and research. It is a programming language renowned for its elegance, power, and challenging learning curve. But while Racket retains the functional goodness of Lisp, it was designed with beginning programmers in mind. It goes beyond Lisp and Scheme with dialects that support objects, types, laziness, and more.

Racket enables programmers to link components written in different dialects, and it empowers programmers to create new, project-specific dialects. Racket's libraries support applications from web servers and databases to GUIs and charts.

The platform provides an implementation of the Racket language (including a sophisticated run-time system, various libraries, JIT compiler, and more) along with a development environment called DrRacket (formerly named DrScheme) written in Racket itself. The IDE and an accompanying programming curriculum are used in the ProgramByDesign outreach program, an attempt to turn computing and programming into "an indispensable part of the liberal arts curriculum". The core language is known for its extensive macro system which enables the creation of embedded and domain-specific languages, language constructs such as classes or modules, and separate dialects of Racket with different semantics. The platform distribution is free and open source software distributed under the LGPL license. Extensions and packages written by the community are uploaded to Racket's centralized package catalog.

Applications and practical use

Apart from having a basis in programming language theory, Racket was designed to be used as a general-purpose language in production systems. Thus, the Racket distribution features an extensive library that covers systems and network programming, web development, a uniform interface to the underlying operating system, a dynamic foreign function interface, several flavours of regular expressions, lexer/parser generators, logic programming, and a complete GUI framework. Racket has several features useful for a commercial language, among them an ability to generate standalone executables under Windows, Mac OS X and Unix, a profiler and a debugger included in the integrated development environment (IDE), and a unit testing framework. Racket has been used for commercial projects and web applications. A notable example is the Hacker News website, which runs on Arc, which is developed in Racket. Naughty Dog has used it as a scripting language in several of their video games.

A glimpse of Racket-lang

Getting started with Racket-lang is pretty easy unlike lisp which requires a substantial amount of knowledge before even writing your first hello world program. Below is a small example to get you started with Racket-lang .

This program, taken from the Racket website, draws a Sierpinski triangle, nested to depth 8.

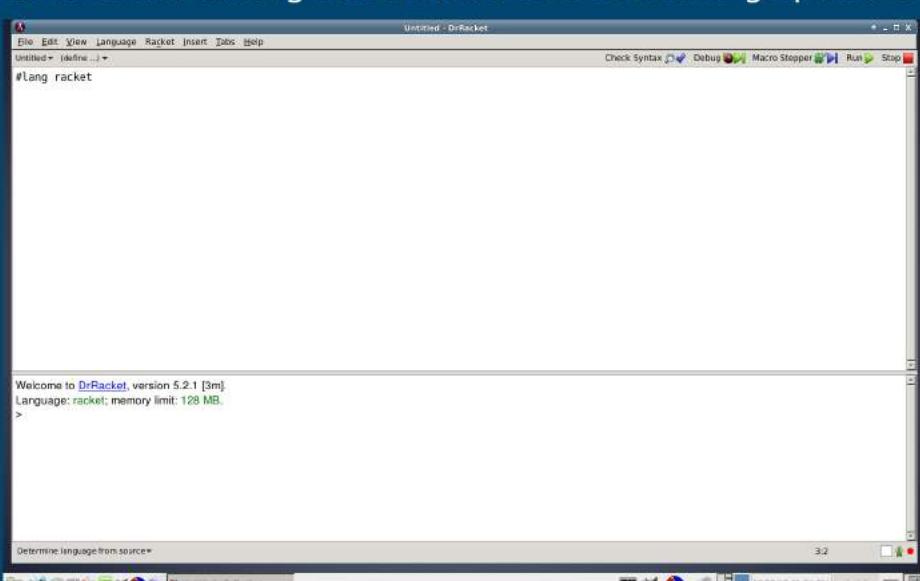
```
#lang racket
(require 2htdp/image)

(let sierpinski ([n 8])
  (if (zero? n)
      (triangle 2 'solid 'red)
      (let ([t (sierpinski (- n 1))])
        (freeze (above t (beside t t))))))
```



IDÉ for Racket-lang

Racket-lang has a plethora of tools and among those DrRacket is a Rackets graphical environment for developing programs.



Other useful tools are :

- [Futures Visualizer](#)
- [Optimization Coach](#)
- [PLaneT: Automatic Package Distribution](#)
- [Racklog: Prolog-Style Logic Programming](#)
- [RackUnit: Unit Testing](#)
- [Redex: Practical Semantics Engineering](#)
- [Scribble as Preprocessor](#)
- [Slideshow: Figure and Presentation Tools](#)

Go farther

Grow your Program:

Racket's interactive mode encourages experimentation, and quick scripts easily compose into larger systems. Small scripts and large systems both benefit from native-code JIT compilation. When a system gets too big to keep in your head, you can add static types.

Grow your Language:

Extend Racket whenever you need to. Mold it to better suit your tasks without sacrificing interoperability with existing libraries and without having to modify the tool chain. When less is more, you can remove parts of a language or start over and build a new one.

Grow your Skills:

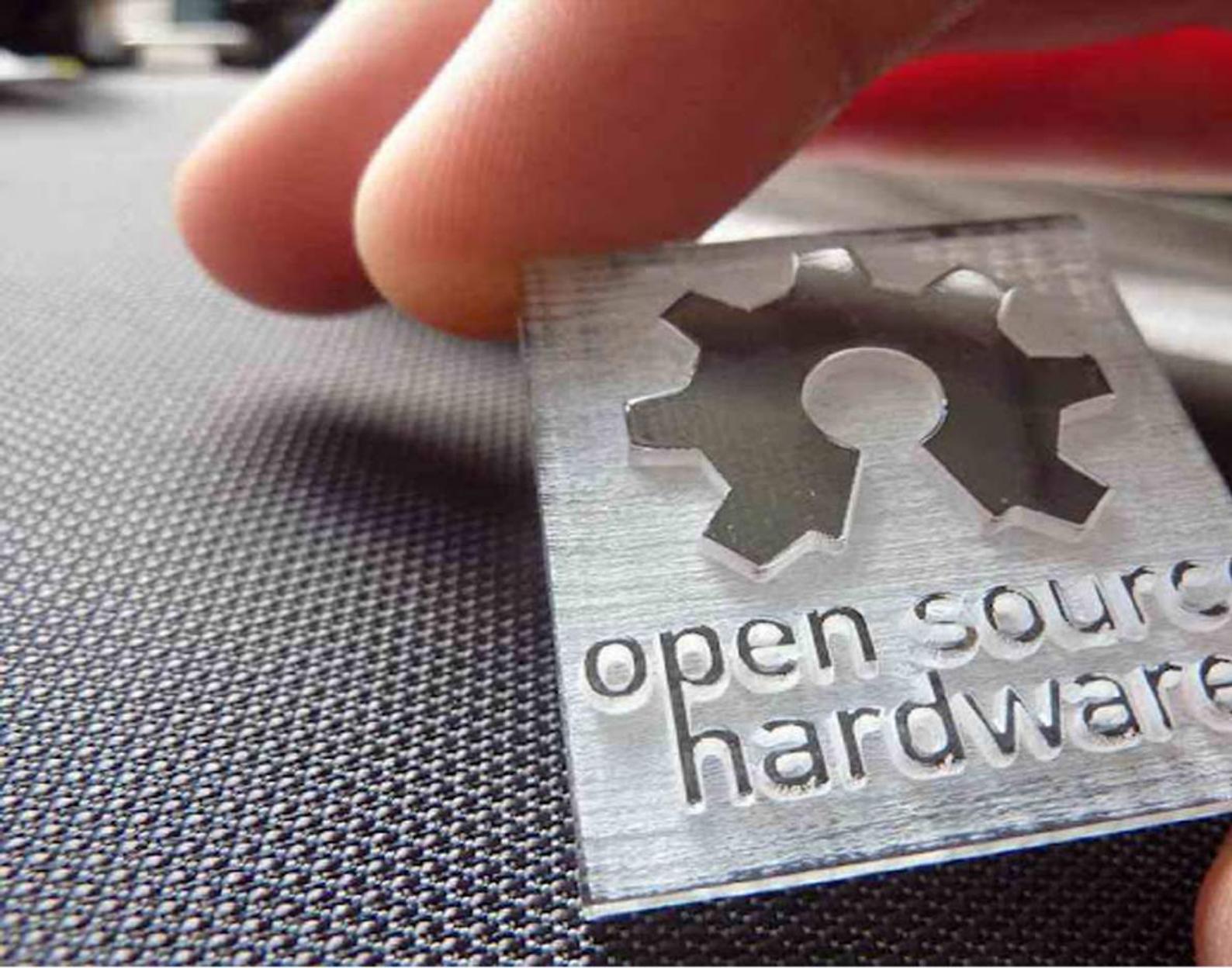
Whether you're just starting out, willing to know more about programming language applications or models, looking to expand your horizons, or ready to dive into research, Racket can help you become a better programmer and system builder.

Learning Racket is easy and the community is very active so you get your doubts clear easily. Also, the documentation of Racket is vast and complete and helps you understand every bit of language.

Ahmed Zammel



Software Engineer



One of the striking developments of the past decade is indeed the emergence of open source technologies and practices in areas such as hardware manufacturing and synthetic biology. This extension of the technical principles and ethos of the Free was followed (/accompanied) by a slogan that evokes the punk culture: "Do It Yourself".

The general spirit of this movement is that the technical creation whether making objects or hacking electronics systems is now within the reach of everyone.

This conviction feeds on the advent of new machines, affordable for individuals: 3D printers, laser cutters for producing low-cost objects and gadgets.

It enrolled also explicitly in the tradition of free software, which is underlined by the open source nature of many devices used nowadays, the emphasis on collaboration as a method, and especially the desire to promote a creative and playful relation to the technical in the spirit of hacking.



We can consider that 2005 is the real year of birth of the DIY movement. From a technological point of view, that's when a team from the University of Bath (UK) directed by Adrian Bowyer starts to develop a 3D printer, capable of producing various small plastic objects. Named RepRap, this machine does not deposit ink on paper but plastic layers on top of each other, thus making "stage by stage" objects in volume from digital files created on a computer-aided design software (CAD). In the same way we can print a text infinity with the same machine, the RepRap can potentially produce an infinite number of objects. It therefore lends itself to all fantasies.

The first model of the 3D printer, finished in 2006, is, nevertheless, rudimentary since only make objects in plastic and of small size. The technique used is not more revolutionary. The RepRap uses a process of "rapid prototyping" invented in the 1980s, already put to use for several years by designers and architects to quickly visualize their works during the design phase.

Its name is short for "Replicating Rapid-prototype" because its creator wanted it can self-replicated, or at least produce itself many different plastic parts of which it is composed, the remaining parts such as motors can easily be acquired.

Each owner of a RepRap is thus likely to make another one, easily and at a cost nearly one hundred times lower than existing rapid prototyping machines.

Adrian Bowyer immediately publish the plans of the machine under General Public License and encourages the formation of an innovative-user community on the model of open source projects. He registered his invention in the orbit of the free, with ambitions to disseminate as widely as possible to promote the formation of spin-off projects, and enable the rapid development of the technology.

Soon after, Cornell University in the United States launches a similar 3D printer project, Fab@Home, whose plans are also published under a free license. Other open source technical objects are created at the same time. The most emblematic germ in 2005 in the town of Ivrea in Italy, Bar di re Arduino. Named in honor of it, the Arduino board has been known for a huge success. Sold very cheap (about 30 euros), it connects to numerous devices and allows for many and diverse tasks: controlling a 3D printer, direct a robot, control a thermocycler, manage lighting or heating systems, etc.



open source hardware

An important part of the Arduino success comes from the DIY spirit that led to its design. Its creators originally destined to an audience of design students. They wanted to create a tool plug and play, easy to learn and to do things quickly. David Cuartielles, member of the core development team, described the "philosophy" of Arduino as follows: "If you want to learn electronics, you should be able to learn by doing from day one, instead of starting by learning algebra."

In line with the celebration of pedagogical virtues of the hack, the creators of Arduino published under free licenses hardware design of the card, its software components and related documentation. RepRap Arduino and have become emblems of "free hardware" or open hardware. These objects, devices or machines, whose technical characteristics, the timeline and software components are available, and which therefore apply the four freedoms of free software use, copy, modify, distribution.

If the term open hardware is new, the concept is long time aged. Technical history studies have shown that in the eighteenth and nineteenth centuries, several technologies were developed through open processes of collective invention.

There is little doubt, however, that this type of approach has experienced a real boom for ten years, working closely with the success of free software.

Thus, open hardware projects now exist in many areas, including consumer electronics (phones, cameras, game consoles, etc.), robotics or biology. They appear to outline the contours of a new technology package, underpinned by the will inherited from hacking to promote an experimental compared to the technique.

Mohammed Chedli Ben Yaghlen



- Computer science student at ISTIC Borj-Cédria.

- CO-founder of DevSpace.

- Member at OpenFab Tunisia.



Avez vous déjà entendu parler du "Rubber Ducky"?

Non, ce n'est pas l'innocent petit canard avec lequel joue un enfant. Certainement, la plupart d'entre vous ignore son existence, mais si je vous dis que c'est une sorte de clavier dans l'apparence d'une clé USB ? Un peu bizarre, n'est-ce pas ? Ma découverte de ce petit objet était le jour où j'ai trouvé un Flash Disque sur mon bureau de stage au sein de l'entreprise SIFARIS (entreprise française dans le domaine de la Sécurité), avec une note où est écrit "Rubber Ducky". C'était notre mission pour 2 jours : découvrir et implémenter un script pour obtenir tous les fichiers « texte » d'un PC victime.

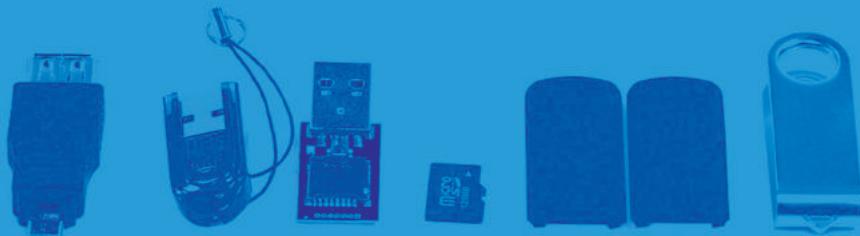
Je résume dans cet article ce qu'a mené notre recherche.



"If it quacks like a keyboard and types like a Keyboard, it must be a Keyboard", c'est sur ce principe que le Rubby Ducker est basé, en effet, l'équipe Hack5 l'a conçu de telle sorte qu'il se comporte exactement comme un clavier, une fois branché à un PC il se charge de taper à la place de l'utilisateur. La bonne nouvelle c'est que notre outil ne ressemble pas seulement à un clavier dans ses fonctionnalités mais il a hérité presque toutes ses faveurs :

- il est multiplateforme et peut s'exécuter sur les systèmes Linux, MacOs, Windows, et même les systèmes mobiles comme Android. Alors, équipez-vous d'un adaptateur USB et essayez-le
- Il ne demande pas de droits ou de pilotes spéciaux pour fonctionner : une confiance absolue de la part des systèmes d'exploitations est accordée.

Équipé d'un processeur 60 MHz 32-bit, il est tellement rapide qu'il peut écrire au-delà de 1000 mots par minute. Le Rubber Ducky, avec son apparence non sceptique, sa mémoire extensible via microSD, et sa communauté active, constitue le centre d'intérêt de plusieurs professionnels de la sécurité depuis 2010.



Alors qu'est ce qu'on peut faire avec un Ducky ?

Évidemment tout ce qu'on peut faire à l'aide d'un clavier, en effet en utilisant les raccourcis disponibles pour lancer des tâches système ou pour lancer des applications on finit par exécuter des commandes, voler des données et installer des programmes.

Prenons un exemple concret : Alan veux planter le pc de son ami qui utilise Linux, il peut appuyer sur "Alt + f2", taper "xterm", puis sur "entrée", taper " :(){ :|:& };" (ForkBomb) et finalement "entrée", Facile non ? Maintenant , c'est à votre tour de laisser votre imagination intervenir avec des idées innovantes . 😊

Rubber Ducky utilise un langage de script assez simple qu'on va détailler avec un petit tutoriel. Pour tester quelques scripts, jetez un coup d'œil sur ce lien : <https://github.com/hak5darren/USB-Rubber-Ducky/wiki/Payloads>, vous l'adorerez !



Avant de commencer , il faut signaler que toutes les commandes du script doivent être en majuscule. Voici une liste de commandes qu'on peut utiliser :

REM : pour écrire des commentaires.

DELAY : pour faire un temps d'attente. C'est important au début du programme pour donner une période suffisante pour que le pc victime reconnaise le périphérique.

STRING : pour écrire.

ENTER: pour appuyer sur la touche "entrée"

CONTROL, ALT, SHIFT, MENU, DOWNARROW, ...

Dans notre tutoriel on va aider Alan à planter le pc de son ami, c'est simple pour débuter :

```
DELAY 3000
ALT F2
DELAY 500
STRING xterm
ENTER
DELAY 500
STRING :(){ :|:& };;
ENTER
```



Maintenant, notre script est prêt, il ne reste que le compiler en binaire, et le mettre à la place de "inject.bin" qui se trouve par défaut sur la carte SSD de notre Ducky.

Il existe deux manières pour le faire , la plus facile est d'aller sur la page : <http://www.ducktoolkit.com/Encoder.jsp> qui est une interface graphique pour encoder votre code. N'oubliez pas de choisir la configuration du clavier relatif à votre pays, puis générer le script et enfin télécharger votre fichier binaire.

Si vous êtes un Geek et vous aimerez bien lancer des commandes, téléchargez l'encoder v2.1+ à partir du lien suivant : <https://code.google.com/p/ducky-decode/> puis utilisez la commande suivante :

```
java -jar duckencoder -l ressources\UK.properties -i notre_code.txt -o inject.bin
```

Bien sûr, remplacez UK par votre code de pays pour configurer la langue.

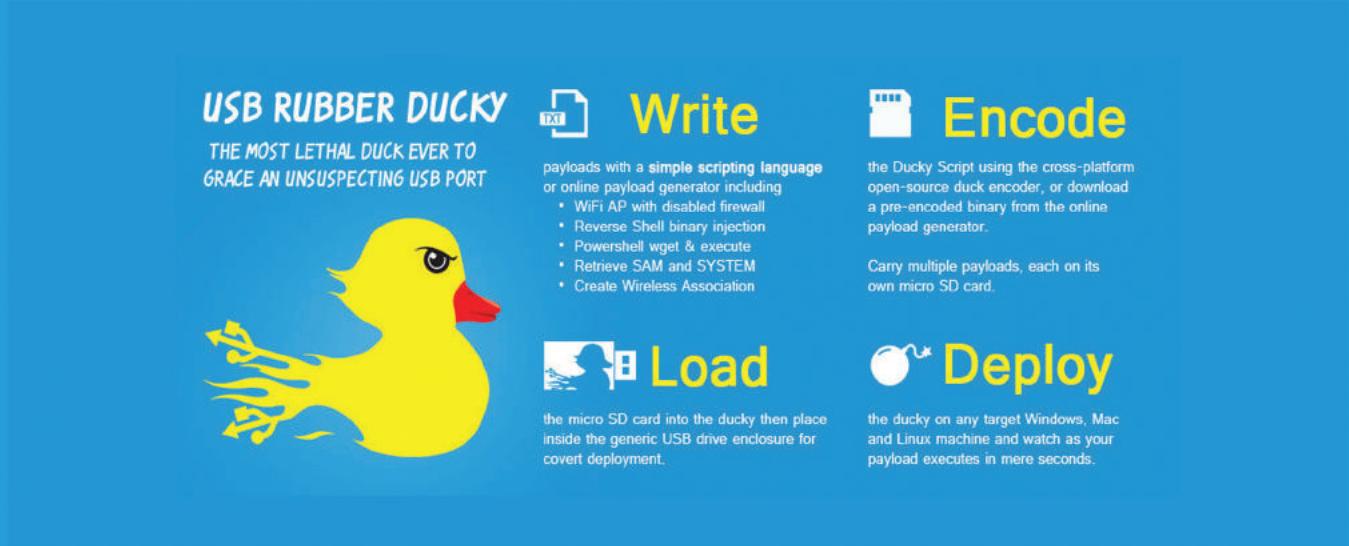
Pour les geeks, il est possible d'aller plus loin, le firmware écrit en "c" est disponible en source ouverte sur Github : <https://github.com/hak5darren/USB-Rubber-Ducky>, n'hésitez pas à contribuer.

Enfin il faut signaler un point important, nous avons remarqué que parfois l'installation du périphérique ainsi que le temps d'exécution des commandes varient d'une machine à une autre selon les caractéristique de celle-ci, vous devez alors prévoir un délai d'attente suffisant.

Vous pouvez acheter votre Rubber Ducky pour un prix de \$42.99 à partir de la page suivante :

<http://hakshop.myshopify.com/products/usb-rubber-ducky-deluxe?variant=353378649>

Amusez-vous bien !



References

- <https://github.com/hak5darren/USB-Rubber-Ducky/wiki/Duckyscript>
- <http://hakshop.myshopify.com/products/usb-rubber-ducky-deluxe?variant=353378649>
- <http://korben.info/usb-rubber-ducky.html>

Haithem Abdelli	
	<p>Software Engineer Ancient Chairman -OSSEC</p>



blender

The Freedom Of Your Creativity

The History of Blender starts with Ton Roosendaal, chairman of the Blender Foundation.

But let's go back to the year 1988, when Ton co-founded the Neo-Geo animation studio in the Netherlands. Back then the studio had a quick and growing success and was one of the most important in the country, but after some time, Ton , who was the art director and also responsible for software development, decided to rewrite the current in-house 3D toolset of Neo-Geo because it was too old and hard keeping it updated.

In 1995, the project he was rewriting, started to be the 3D software we know as Blender and a few years later Neo Geo was acquired by another company, causing some problem with the development of the code for Blender.

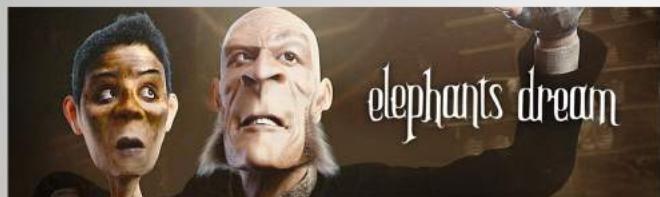
This is why, to keep Blender alive and develop it further, Ton and Fran Van Beek, founded a new company called Not a Number (NaN), starting to distribute Blender as Shareware.

In January of 1998 a free version of Blender was released.

The bad economy and the low sales of the NaN company, caused its closure in 2002, but in March of the same year Ton founded the Blender Foundation, which was created to continue the development of Blender.

In July 2002 the foundation reached the 100 000 EUR goal and in October they released the open source version of Blender.

The first movie produced exclusively with Blender was Elephant Dream, where seven artists from around the world collaborated.



Blender also has a powerful baking tool which can bake variety types of maps that can be used in other game engines.

There is a great support for a lot of different file formats. Be it for single frames, videos or 3D-models.

It can export rendered images into standard formats like jpg or png but also into high quality formats like exr and animations into every format you may desire for uploading to YouTube or further processing.

Blender is one of the most powerful 3D programs available out there. It's free but it can beat many other paid softwares which are usually used by big studios. The two most interesting things about Blender are its shortcut system and the ability to do most of CG related work like modeling, sculpting, texture painting or video editing and it even got an own game engine. All in a single software.

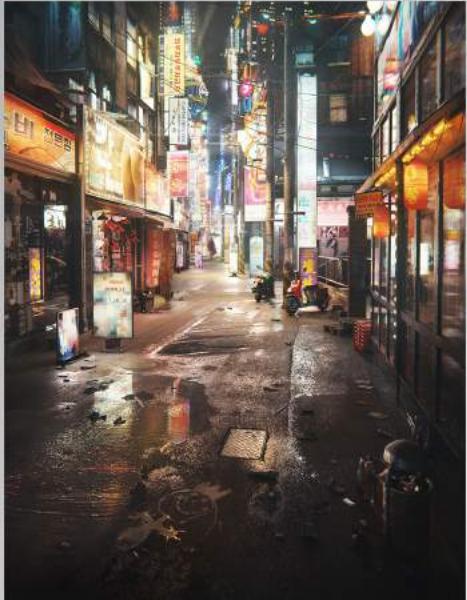
Other important features of the program includes UV unwrapping, raster graphics editing, rigging and skinning, different simulations like fluids, smoke/fire, soft body and cloth, particle/hair systems, rigid bodies, fracturing, animating, match moving, camera tracking, rendering and compositing.



Blender also brings five different render engines with it :

Cycles is an unbiased, physically based backwards path-tracing engine (similar to V-Ray and Arnold) making it a very powerful tool inside Blender, that allows to render very high detailed and photorealistic images without a compromise to the usability and speed (which with every new version increases). Another important feature of Cycles is the node system, which is a user-friendly and intuitive way to build powerful and realistic shaders.

Another big advantage of Cycles is that it can render with the GPU (optimally with Nvidia cards but AMD is always getting better support).



Seoul by Gleb Alexandrov

The so called Internal engine, which was the original engine of Blender is a biased rasterization engine. It is very suitable for comic-style renders but with a bit more experience one can reach a good level of photorealism too and even real-time rendering of nearly physically accurate materials. For a lot of simpler animation it got a speed advantage over Cycles. But it can only utilize the CPU for rendering.



Ms. Evelin by
Gerardus Mayella Aryo Timur

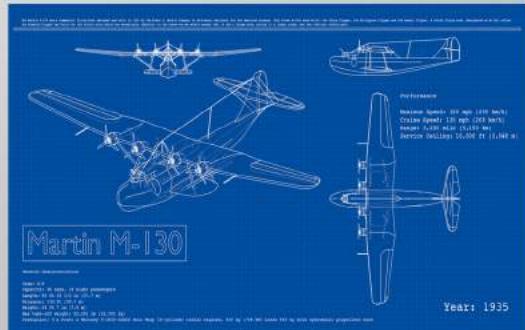


The Game Engine uses a similar approach as the Internal Engine. It is able to approach the level which you get with PBR really good and is also really fast (as it is for games obviously).



Vy Robin Flodin

Blender provides a NPR (non-photorealistic) render engine too: Freestyle. This enables you to just create outlines of your models and further use them (e.g. as an input for a laser-system or as a blueprint) or in combination with renders as an approach to cartoon renders.



Martin M-130 Blueprint by LightBWK

Last but not least is the Engine used in the viewport: OpenGL. It is often overlooked but really powerful as you can apply different MatCaps and just render a preview of your animation incredibly fast.



Space Nerd Room by bastART

Other famous and very powerful engines which are used or even developed by big studios can be used alongside Blender.

Examples are RenderMan (as developed by Pixar), V-Ray, Arnold or other open source software like LuxRender.

As open source software, Blender supports the python language which is fairly easy to learn. With the help of the python documentation any beginner can make little handy add-ons to automate processes, discover new ways of making something, share the knowledge and help the Blender community to grow. Blender brings some Python templates with it (like different menus) so that it is even easier to customize your workflow. Additionally you can adjust all the shortcuts as you like or even use preferences from other softwares like Maya (the same as with Blender Themes).



The software currently counts a lot of enthusiastic users that back the project and spread their art made with it, surely Blender has the largest community of all other softwares out there which is great because if you are stuck on a problem you can get help very fast and easy. 3D students usually look for constructive criticism and you can help others by doing that. This will develop your skills, not to mention the inspiration you can get by looking at a vast amount of artworks posted daily.

Everything from the official manual over video tutorials to more specific resources can be found online. (blender.org/manual, cgcookie.com, blenderguru.com)

Blender is a free software, but the most important thing is the freedom of use, rather than the fact that is free of charge. This is why we Blender users, as a community, do our best to keep Blender alive and updated.

BCGIU Staff :

- **Souvik Karmakar (India)**
- **Mary Shan Fazzolari (Italy)**
- **Tilmann LePoer (Germany)**

HACK4TN

EVENT

VIVITEK
Your Color Vivid Life



It is commonly known that students ought to start clubs in their universities to break up their studies and establish a kind of "personal brand". Here at our school ENSI we believe that your field of studies can also be your hobby ,it's like kicking two birds with one stone.

One of the most active clubs at our school and who has been doing great job gathering students and taking their studies outside of the classroom is GDG ENSI.

Google Developers Group at ENSI is a community for everyone interested in Google products or technologies. Our activities take many forms: Small gatherings where we discuss the latest technological advancements (mainly developed by Google), workshops where anyone can learn how to use Google products and even events where programmers participate in hackathons and coding sprints.

They say "**WHEN THERE IS A WILL THERE IS WAY**" , we say **WHEN THERE IS GDGERS THERE SHALL BE A WAY** , the GDG team, started working on our new and promoting event "HACK4TN" (Hack For Tunisia) in which we try with our modest resources to spread awareness and peace in Tunisia.

The event was held on the 9th and 10th of April 2016 at the National School of Computer Sciences. "All united for a better Tunisia" was the theme of the first edition of HACK4TN which was in partnership with the Ministry of Communication Technology and Digital Economy.

The journey started with the presentation of the hackathon's theme and rules to the participants.

Students and programmers from all universities dispersed into the nooks and crannies of our school for 24 straight action-packed hours.

While the participants were passionately and collaboratively building their own application, a discussion panel about "National security: A state matter or a collective duty" just started.



It was animated by Mr. Hamza Kaabar, Regional Programs Manager at the International Debate Education Association. Mr Sami Ghazeli Digital Economy & Investment Head at ICT Ministry, Mr Bechir Raddaoui Head of the Department of Information Systems and Decision Making and Mss Rahma Gharssalaoui Principal Engineer in Radio Communication at the National Agency of Frequencies took part in this interactive debate as coordinators.

Conferences were also in the rendez-vous and thus focused on various topics.

For instance, Mr Lotfi Saibi, President and Director at 21st Century Services & 4D-Leadership House, talked about Students and patriotism. Mr Nizar chaari, CEO of Media Visions Editing, talked about Journalism's role in the creation of a young patriot. Mr Anouar Romdhani, Large opportunity manager at Microsoft, was also present to talk about the Entrepreneurship life. And that's not all , cyber security was also among our topics . Mr Moez Chakchouk, CEO of La Poste,talked about the status of cyber security in Tunisia and how can young programmers/hackers contribute in improving the security of Tunisian systems.

Late at night, the event went on by etching in us unforgettable memories: A Fantastic musical show presented by the ENSI Musical Club, a CTF : Capture THe Flag (it's like pokemon go but you are going to catch a flag rather than a pokemon) and tournaments of various games filled us with so much joy and kept us awake and entertained .

The 10th of April was the big day for the hackathon's team.It began with workshops about web development and exhibition stands of our sponsors. And after 24 hours of coding,the moment everyone waited for was finally there , the participants started presenting their applications on the HACK4TN stage to a panel of renowned judges.

The hackathon's winners:

iRobots team (AbdelMonaem Ghzayel, Talel Krimi, Morsi malek, Mohamed ghorbel)

Leaders team (Abdennaceur Nebhen, Ichrak zarati, asma turki)

We thank everyone who made this event more significant with their presence: participants, technical staff, students, judges, speakers...and especially the GDG community that worked extremely hard to bring this event to life :"This is not the end this is just the beginning"



OSSEC (Open Source Software Ensi Club) est un club créé à l'école nationale des sciences de l'informatique (ENSI) en 2011 ..

Notre but est de promouvoir la culture de l'open source entre les étudiants de l'ENSI , qui se base essentiellement sur le partage et l'échange .

Nous visons ainsi à assurer un environnement de travail et de développement qui prépare nos adhérents à la vie professionnelle ..





Director :

Souha Znina

Editors :

Wafa Yahyaoui

Soumia Zammit

Oumaima Manari

Abyr Barhoumi

Lamia Ben Nticha

Marwa Kouki

Tewfik Gheriani

Radhouen Ayyed

Designer & International Affairs :

Rayen Messaoudi

Writers :

Swaraj Mohapatra (India)

Sami Belhaj (Tunisia)

Mohamed Chennoufi (Tunisia)

Souvik Karmakar (India)

Mary Shan Fazzolari (Italy)

Tilmann LePoer (Germany)

Hanin Slimen (Tunisia)

Ahmed Zammel (Tunisia)

Marwa Kouki (Tunisia)

Wassim Chaabouni (Tunisia)

Hala Sayehi (Tunisia)

BOUZIDOC (Tunisia)

Mehdi Ben Hmdia (Tunisia)

Haithem Abdelli (Tunisia)

G2FOSS (Tunisia)

Stefan Gronewold (Germany)

Nidhal Kmira (Tunisia)

Med Amin Ben Tmim (Tunisia)

Openfabs Tunisia (Tunisia)

GDG ENSI (Tunisia)

Ossec

Sharing knowledge is the most fundamental act of friendship.

Because it is a way you can give something without loosing something.

- Richard Stallman -



National School Of Computer Science
Open Source Software ENSI Club



ÉCOLE NATIONALE DES SCIENCES
DE L'INFORMATIQUE