

Devenez un super héros du JavaScript

**Choisir son
framework
JavaScript**

**JavaScript
pour les Jedis**

Faites carrière dans les **jeux vidéo !**

Geek

Des étudiants
révolutionnent
la cabine d'essayage

**Accessibilité
des applications**
Il y a urgence !

Android

- Les (meilleurs) outils pour développer vos apps
- La nouvelle interface Material Design

**Choisir son école
d'informatique**

**Késako le
Machine Learning ?**

SDK Azure : les nouveautés

1001100 1001100 1001100
1001100 1001100 1001100
1001100 1001100 1001100
1001100 1001100 1001100
1001100 1001100 1001100

1001100 1001100 1001100
1001100 1001100 1001100
1001100 1001100 1001100
1001100 1001100 1001100
1001100 1001100 1001100

1001 1001100
1001100 10
1001100
01100 1100

1001100 1001100
1001100 1001100 1001100
1001100 1001100
1001100 1001100 1001100
1001100 1001100 1001100



TOUR DE FRANCE

WINDEV®

TDF TECH 2015

VOUS ÊTES INVITÉ !

11 villes

du 10 mars au 7 avril

10.000 places

inscrivez-vous vite !

(gratuit)

www.pcsoft.fr



Inscrivez-vous vite !

Montpellier	mardi 10 mars
Toulouse	mardi 17 mars
Bordeaux	mercredi 18 mars
Nantes	jeudi 19 mars
Bruxelles	mardi 24 mars
Lille	mercredi 25 mars
Paris	jeudi 26 mars
Strasbourg	mardi 31 mars
Genève	mercredi 1er avril
Lyon	jeudi 2 avril
Marseille	mardi 7 avril

de 13h45 à 17h45

38 sujets techniques
traités.

Sur place, vous
recevrez le **DVD** conte-
nant le code source des
applications présentées.

**Inscrivez-vous vite
sur www.pcsoft.fr**

*(inscription gratuite, mais
seulement 10.000 places
disponibles)*



Tél province: **04.67.032.032**

Tél Paris: **01.48.01.48.88**

Fournisseur Officiel de la Préparation Olympique

www.pcsoft.fr

Des centaines de témoignages sur le site

100% TECHNIQUE



L'accessibilité, un vrai sujet sérieux

Parlons d'un sujet sérieux mais bien trop peu abordé dans le monde informatique et particulièrement dans le développement : l'accessibilité.

Régulièrement, les éditeurs de systèmes d'exploitation évoquent les fonctionnalités d'accessibilité pour les personnes ayant des déficiences, bref des handicaps plus ou moins lourds.

A Programmez !, nous avons abordé ce sujet il y a quelques années. Il nous paraissait important d'appuyer là où ça fait mal, et de refaire un point complet sur ce que l'on appelle l'accessibilité, les différents handicaps, les spécifications actuelles (particulièrement pour les applications et sites Web) et comment le développeur peut la prendre en compte dans ses apps et sites Web.

L'accessibilité concerne plusieurs millions de personnes. Et l'accès au numérique constitue, plus que jamais, un enjeu pour la société et les développeurs. Il ne faut pas voir cela comme une contrainte supplémentaire ou une fonction supplémentaire qui a vocation à devenir « par défaut », mais plutôt comme l'opportunité de s'adresser à de « nouveaux » utilisateurs. Nous espérons que notre dossier vous sensibilisera.

Ce mois-ci, nous envoyons du lourd : du bon gros JavaScript qui tâche bien (mais en mode Jedi, excusez du peu), de l'Android avec l'excellent Material Design (écrit par les experts de Genymobile, une super boîte française), le machine learning (1ère partie du dossier), comment choisir son école informatique et son cursus (vaste débat) ou encore les métiers dans les jeux vidéos !

Nous revenons aussi sur l'annonce du HoloLens de Microsoft, en attendant les détails dans les prochains mois (notamment à la conférence technique BUILD 2015) Et le time machine du numéro, revient sur MS DOS et Windows...

Enjoy !

ftonic@programmez.com,
redaction@programmez.com

10
Hacking



24
Machine Learning

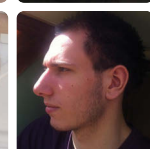
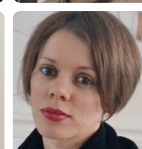
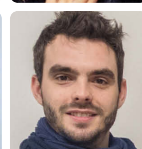
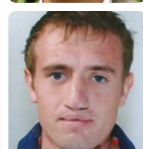
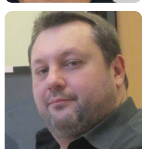
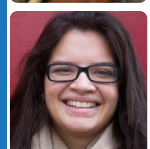
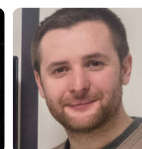
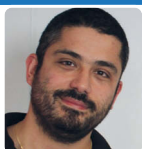
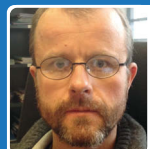
5
Agenda



16
Faire carrière dans les jeux vidéos



20
Choisir son école d'informatique



8
HoloLens

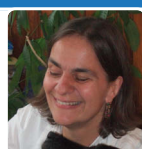
70
Coding4Fun

65
Android : Material Design

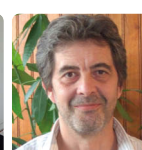
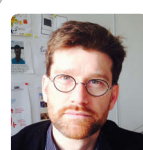
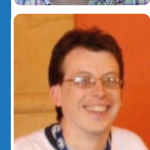
45
Les meilleurs outils pour Android

80
Time Machine

4
Tableau de bord



30
Les défis et l'importance de l'accessibilité



82
CommitStrip

76
Les nouveautés du SDK Azure



73
Cloud : Amazon Web Services

à lire dans le prochain
numéro n°184 en kiosque le 28 Mars 2015

JAVA 8

Java 8 est sortie il y a un an. Quel bilan ? Quels outils ? Les +, les -.

NOSQL

L'explosion des données de tout format et de toute nature a révolutionné la base de données. Qu'est-ce que le NoSQL ? Focus sur Cassandra.

XAMARIN

Retour sur Xamarin et les fonctions avancées de la plateforme de développement mobile.

Internet Explorer

Bonjour, Spartan

Internet des Objets :

les développeurs seraient déjà conquis (25 % selon une étude Evans Data). Hum, pas convaincu !

Windows 10

version entreprise payante, version serveur en sortie décalée

Le langage Go

arrive sur Amazon Web Services !

1,6 % pour Android 5.x...

Google fournit des chiffres intéressants sur les diverses versions d'Android actuellement actives (via un dashboard).

Version	Nom	API	Parc (en %)
2.2	Froyo	8	0.4 %
2.3.3 — 2.3.7	Gingerbread	10	7.4 %
4.0.3 — 4.0.4	Ice Cream Sandwich	15	6.4 %
4.1.x	Jelly Bean	16	18.4 %
4.2.x		17	19.8 %
4.3		18	6.3 %
4.4	KitKat	19	39.7 %
5.0	Lollipop	21	1.6 %

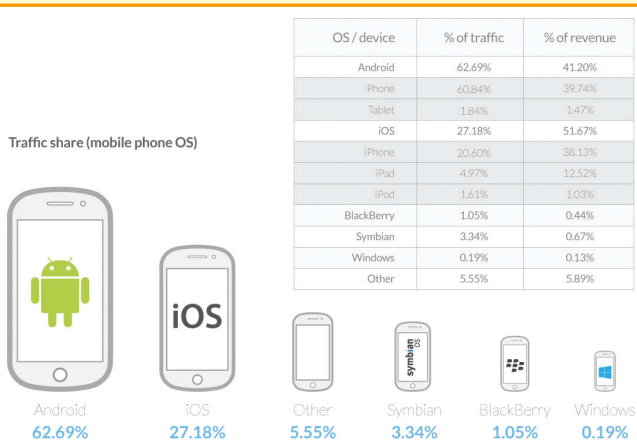
KitKat est toujours le premier Android avec presque 40 %. Par contre, la lignée 5.x (Lollipop) ne décolle pas. Disponible depuis octobre 2014, Android 5.x affiche un ridicule 1,6 % !

Tous les dashboards Google :

<http://developer.android.com/about/dashboards/index.html>

Android + iOS : 89, 87 % du trafic Web mobile, 98 % des revenus pubs mobiles !

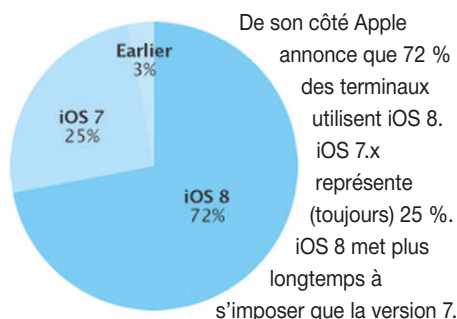
Plus que jamais, il y a Android et iOS sur les terminaux mobiles. Le dernier rapport d'operamediaworks.com est ahurissant. Android est la 1ère plate-forme pour le trafic Web mobile, mais iOS est toujours 1er sur les revenus publicitaires et le store.



YouTube

HTML 5 par défaut. Flash out.

... 72 %
pour iOS 8

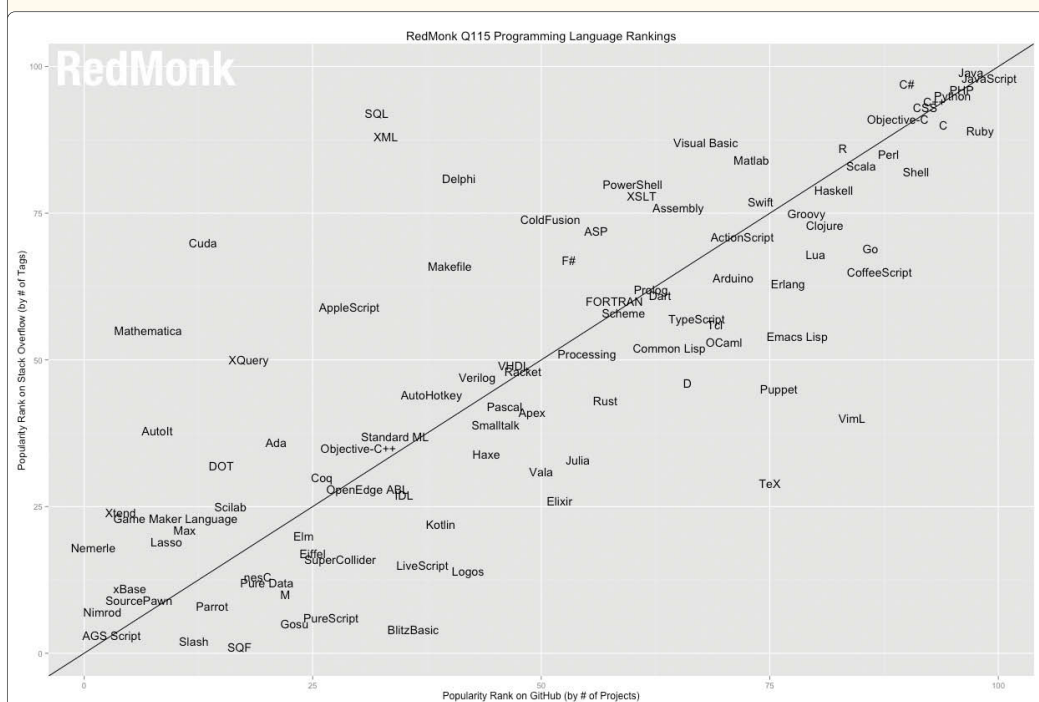


La prochaine étape sera la version 8.2 prévue d'ici fin mars avec le support de l'Apple Watch et sans doute de nouveautés autour d'Apple Pay et de HomeKit. Site :

<https://developer.apple.com/support/appstore/>

QUELS SONT LES LANGAGES LES PLUS UTILISÉS ?

Ce mois-ci, pas d'index Tiobo, mais une belle ligne des tendances de RedMonk qui utilise la popularité des langages sur GitHub et Stack Overflow.



■ mars

Windows 10 DevCamp 2

La communauté Windows 10 se retrouvera le 13 mars à Zaventem pour parler Windows 10, les nouveaux outils, les nouvelles fonctions du système. Un autre DevCamp sera organisé après la conférence BUILD.

A l'école 42 en mars...

Le 12 mars 2015 aura lieu à 42, le plus grand concours de développement de France. Plus de 700 développeurs en équipe entreprise, en agence ou en solo, s'affronteront sur des épreuves individuelles pour tenter de décrocher le titre. Une fois le code validé par le jury, le gagnant se verra attribuer la récompense de 10 000 €, et de nombreux lots récompenseront les suivants. C'est également l'une des plus grandes manifestations de l'hexagone autour du numérique avec de nombreuses conférences et la découverte d'entreprises qui façonnent le monde de demain."

Plus d'informations sur :

www.lemeilleurdevdefrance.com



■ avril

Devoxx France 2015 : le rendez-vous Java !

Attention, c'est l'événement des communautés Java de l'année. Devoxx France revient en force pour une édition qui promet d'être explosive : nouvelles sessions, nouveaux stands et surtout nouveau lieu. L'événement se déroulera au palais des congrès de Paris ! Pour cette 4e édition, plus de 2000

personnes sont attendues pour 3 jours de conférences et d'ateliers.

Plusieurs keynotes se joueront les jeudi et vendredi. Devoxx4kids sera de retour le samedi 11 avril.

Un hackergarden sera disponible durant l'événement. Le Hackergarten est l'endroit idéal pour rencontrer de nouvelles personnes, de s'amuser et d'améliorer la vie des développeurs du monde entier « un patch à la

fois » ; à condition d'être prêt à apporter votre ordinateur portable et de vous salir les mains en codant. Développer un plugin, rédiger de la documentation, rajouter une fonctionnalité ou apporter la correction d'un bug sur un projet open source qui vous intéresse... et ce, aux côtés de plusieurs experts open source prêts à partager leurs connaissances.

Retenez dès maintenant votre agenda :

du 8 au 10 avril !

Site officiel : <http://www.devoxx.fr>

Journée française des tests logiciels 7e édition

Le 14 avril se déroulera la nouvelle édition de la journée des tests logiciels. Cette année, l'événement change de lieu. Comme à chaque édition, la journée sera ponctuée de sessions générales et plus « techniques ». Une journée sera consacrée aux tutoriels (13 avril), la journée ouverte étant celle du 14 avril. De nombreux éditeurs et SSII seront présents.

Pour en savoir plus : <http://www.jftl.org>



zenika

03/03

Conférence Ha Proxy/Varnish :

<http://www.zenika.com/conference-e-haproxy-varnish.html>

Au cours de cette conférence, nous verrons comment intégrer HAProxy et Varnish, les points communs et les différences de ces solutions ainsi que leurs avantages et leurs inconvénients. Nous simulerons ensuite une infrastructure Varnish basé sur des conteneurs Docker et nous verrons comment HTTPS est tout aussi simple à mettre en œuvre avec HAProxy.

04/03

Matinale Zenika Lyon DevOps : Pourquoi et comment ?

<http://www.zenika.com/matinale-zenika-lyon-devops.html>

Zenika vous propose de découvrir les fondamentaux

26 et 27 mars

NoSQL matters Paris 2015

Dédié aux développeurs, cet événement est une occasion unique d'échanger avec les experts du NoSQL et Big Data et d'approfondir ses connaissances en participant aux formations et conférences proposés pendant ces deux jours.

<http://www.zenika.com/conference-nosql-matters-paris.html>



de la mise en place de cette démarche. Nous vous présenterons un décryptage des pratiques DevOps puis nous illustrerons cela par un retour d'expérience concret de déploiement continu sur plusieurs serveurs dans le cloud en un seul clic.

05/03

NightClazz Docker Avancé

12/03

Cloud Foundry France - Spring dans Cloud Foundry, Docker et Cloud Foundry

<http://www.zenika.com/cloud-foundry-france-spring-docker.html>

Cloud Foundry France est

une communauté qui encourage les discussions, les sessions techniques et le networking entre ses membres, constitués de développeurs, programmeurs, architectes, ingénieurs, consultants et chefs de projets partageant un intérêt commun pour le PaaS open source Cloud Foundry et la fondation Cloud Foundry.

19/03

Dev4Fun

Venez développer dans une ambiance convivial avec le langage que vous désirez et ce quel que soit votre niveau. Le but est d'apprendre et d'échanger sur nos méthodes de développement tout en s'amusant.

AGENDA DES JUG

Le Paris JUG annonce :

9 mars : thème ouvert

Site : <http://www.parisjug.org>

BreizhJug <http://www.breizhjug.org>

19 mars : soirée JOOQ

Lyon JUG <http://www.lyonjug.org/evenements>

10 mars : soirée human talks Lyon

16 & 17 avril : événement Mix-IT

19 mai : Akka sur cluster Raspberry Pi

16 juin : Cassandra et Spark

Global Azure Bootcamp

La communauté Azure se retrouvera le 25 avril dans plusieurs villes dans le monde entier. En France, il y aura des conférences à Paris et à Lyon. Sur Lyon, l'événement promet d'être animé dans la prestigieuse Manufacture des Tabacs. La journée s'articulera autour de conférences techniques et notamment sur les technologies Open Source.

Hololens, un casque de réalité virtuelle mais pas que..

A l'occasion de la conférence du 21 janvier dernier qui s'est articulée autour des nouveautés de Windows 10, Microsoft a surpris en présentant son nouveau produit : Hololens, un casque de réalité augmentée très prometteur et pas comme les autres.



Chouguiat Aymen
Supinfo International University
(1^{er} année) - Email: Aero@vf.vc

Qu'est-ce que c'est ?

Hololens est un casque de réalité augmentée c'est-à-dire un système informatique capable de projeter des objets virtuels dans la vision d'un espace réel. Durant la conférence, le casque a été présenté comme un système holographique ce qui est sans doute un terme un peu abusif comme nous le verrons plus loin. Celui-ci prend la forme de grosses lunettes, d'un encombrement similaire, ou à peu près, à celui des casques de réalité virtuelle tels que l'Oculus Rift, les Google Glass (qui sont plus fines) ou le Morpheus de Sony. Mais contrairement à ces derniers, l'utilisateur peut voir à travers, puisqu'il s'agit ici d'une technologie de réalité augmentée. Microsoft présente son casque comme étant le premier « Ordinateur Holographique Sans Fil » totalement autonome. C'est à dire qu'il



embarque un véritable ordinateur dans le casque contrairement aux autres casques et lunettes nécessitant un PC ou un smartphone pour fonctionner. Microsoft a précisé qu'il embarquait deux processeurs, dont un dédié aux « hologrammes ».

Le casque de Microsoft embarque un équipement qui permet de visualiser des Images en haute définition, d'avoir un son spatialisé, mais le plus bluffant qui a été montré lors de la démonstration de ce casque est la possibilité d'interagir avec les éléments virtuels comme si c'étaient des objets réels.

Réalité Augmentée vs Réalité Virtuelle

Microsoft a annoncé Hololens bien après les Google glass et l'Oculus Rift, mais qu'en est-il réellement maintenant que les trois produits phares ont été dévoilés au public, sachant que Google abandonne peu à peu son effort initial dans le domaine et tandis que l'Oculus rift continue de se perfectionner; prenons le temps de les comparer.

Oculus Rift : la réalité virtuelle au coude à coude

La dernière version de l'Oculus Rift (racheté par Facebook) connue sous le nom de Crescent



*Oculus Rift 3eme génération « Crescent Bay »

Bay est pour le moment l'appareil le plus avancé techniquement, mais il n'est toujours pas abouti notamment au niveau du design, là où le gabarit du Rift peut paraître « Brut » celui du Hololens arbore un design beaucoup plus pratique, plus « conventionnel » avec une visière transparente qui permet d'avoir une vision périphérique et c'est là où réside tout l'intérêt de la réalité augmentée proposée par Microsoft.

Le concept de base de ces deux appareils est similaire, placer l'utilisateur dans un environnement en 3D immersif grâce à une projection d'images (ou d'hologrammes dans le cas du Hololens) juste en face de vos yeux. Les deux casques s'adaptent aux mouvements de la tête et retransmettent ces mouvements dans l'environnement simulé.

L'Oculus Rift utilise une caméra externe pour aider au procédé d'adaptation des mouvements, pendant que vous interagissez avec votre environnement grâce à un contrôleur de jeu dédié. Quant au Hololens, il va prendre une entrée virtuelle et interagir directement avec des périphériques traditionnels (souris, clavier, joysticks...) Dans l'Oculus Rift la vision telle qu'on la perçoit est ignorée et un nouvel environnement est construit pour l'utilisateur; dans cet espace simulé, le seul élément physique qui importe est l'orientation et les mouvements de votre tête.

Avec l'Hololens la vision réelle comme on la perçoit au quotidien est un point d'ancrage essentiel. Des écrans virtuels peuvent être liés





WINDEV® DÉVELOPPEZ 10 FOIS PLUS VITE

TOUR DE FRANCE

TDF TECH 2015
VOUS ÊTES INVITÉ !

11 villes du 10 mars au 7 avril
inscrivez-vous vite !
(gratuit, voir page 2)



Élu
«Langage
le plus productif
du marché»



*Développez une seule fois,
et recompilez pour chaque cible.
Vos applications sont natives.*

Tél province: **04.67.032.032**
Tél Paris: **01.48.01.48.88**


Fournisseur Officiel de la Préparation Olympique

www.pcsoft.fr

Des centaines de témoignages sur le site

à un mur pour une téléconférence, des annotations peuvent être ajoutées aux objets réels devant vos yeux, et des hologrammes 3D peuvent être modélisés sur votre table ou sur le sol en face de vous.

D'un point de vue général les deux casques exécutent cette vision (virtuelle pour le rift et réelle pour le Hololens) de manières très distinctes, la principale différence entre les deux concepts étant la façon dont le monde réel est « Montré ».

Cette différence fondamentale n'est pas un hasard, en effet le marché ciblé par les deux appareils est diamétralement opposé.

L'Oculus Rift est en grande partie orienté vers le gaming, on a pu voir tout le potentiel qu'offrait le casque dans des jeux comme Minecraft et Heli island.

Le casque de Microsoft est, quant à lui, plus accentué sur des applications pratiques et l'usage professionnel, dont la plupart n'ont aucun lien avec les jeux; il peut être utilisé pour des réunions de staff, des projets de constructions (bâtiments, conceptions architecturales), de l'assistanat à distance, toutes les applications qui, à première vue, ne seraient pas adaptés à l'Oculus Rift pour des raisons de praticité.

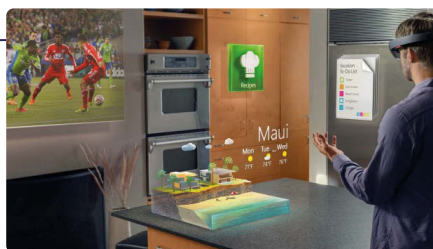
Google Glass : bataille autour de Spécifications intelligentes

Les Google Glass ont été au départ conçues pour être une paire de lunettes discrète, qu'on peut porter au quotidien, incluant des applications intelligentes et apportant une expérience utilisateur novatrice.

Au niveau du design c'est tout le contraire du Hololens, ce dernier étant considéré comme un ordinateur à part entière, totalement autonome certes mais qui a été conçu pour une utilisation fixe contrairement aux Google Glass.

D'un point de vue esthétique, si les Google Glass ont le même encombrement qu'une paire de lunettes classique, le Hololens est quant à lui un outil complet au form factor plus imposant, plus volumineux qui le destinent à un usage intérieur et professionnel, pour le moment.

A la différence de l'Oculus Rift, les Google Glass ont le même principe de base que l'Hololens, un casque de réalité augmentée qui ne vous isole pas du monde réel; à la place on trouve des couches d'interfaces qui sont superposées à travers le monde tel que vous le voyez. Au niveau software, les Google Glass embarquent un logiciel de reconnaissance vocale qui vous permet de donner des instructions. Les lunettes de Google disposent



aussi d'un pavé tactile pour défiler manuellement sur les interfaces affichées.

Microsoft va-t-il réussir là où les autres n'ont pas apporté de réponses ?

Hololens apporte une innovation majeure dans le domaine, le toucher virtuel avec le doigt; concrètement vous pourrez toucher, taper, retourner vos hologrammes dans tous les sens et les examiner sous tous les angles, le tout avec une précision accrue grâce au suivi des mouvements de la tête accordant un contrôle plus fin et plus détaillé des instructions. Pour conclure on peut dire que la conception des Google Glass est plutôt faite pour des tâches quotidiennes, les notifications, les post-

it / Agenda, les fonctions que vous utilisez en temps normal sur votre smartphone. A l'inverse, le Hololens de Microsoft est fait pour des tâches beaucoup plus complexes et immersives comme la mise en place de tutoriels en direct (comment réparer un évier ou une prise électrique), explorer un environnement virtuel à des fins de simulations et de tests, entre autres, ce qui ne l'empêche pas d'exécuter des tâches plus simples. Google a arrêté la vente de la version Explorer. Les Glass n'ont jamais su trouver un usage concret et les développeurs se sont peu à peu retirés. Google n'a jamais su trouver des usages et les multiples limitations et interdictions ont beaucoup nui à ce matériel, mais Google promet de revenir avec une nouvelle itération.

Le développement du Hololens en est encore au stade embryonnaire, mais le casque de Microsoft deviendra sans nul doute un must dans un futur proche, si toutefois les fonctionnalités présentées sont au point et que les applications répondent aux attentes des utilisateurs. A cet effet Microsoft mettra en place un kit de développement « HoloStudio » et des API pour créer des applications holographiques à vocation professionnelle ou grand public. Les applications universelles Windows 10 peuvent également se transformer en hologrammes.

Pour ce qui est de la date de sortie, Microsoft a annoncé le lancement de son produit dans le courant de l'année en parallèle avec la sortie de Windows 10.

Microsoft a aussi profité de sa conférence pour lever le voile sur une déclinaison de Windows 10 appelée Windows holographic, un système d'exploitation qui prendrait en charge les principales fonctions du Hololens. Durant la démonstration lors de l'évènement Windows 10, Microsoft a souligné que Windows holographic effectue un suivi en temps réel de la vision de l'utilisateur; en pratique cela veut dire que l'OS arrive à suivre votre regard en tout temps, vous permettant de pointer du doigt les objets affichés dans l'espace réel afin de les sélectionner.

Un dernier point sur lequel aucune information n'a été communiquée pour le moment, le prix du casque, prix qui devrait alléger la plupart des portefeuilles vu l'équipement embarqué, Satya Nadella, le grand patron de Microsoft, a néanmoins annoncé que le tarif conviendrait à la fois aux particuliers et aux entreprises. Reste à connaître l'ensemble des spécifications techniques, dont l'autonomie et les outils pour le développeur.

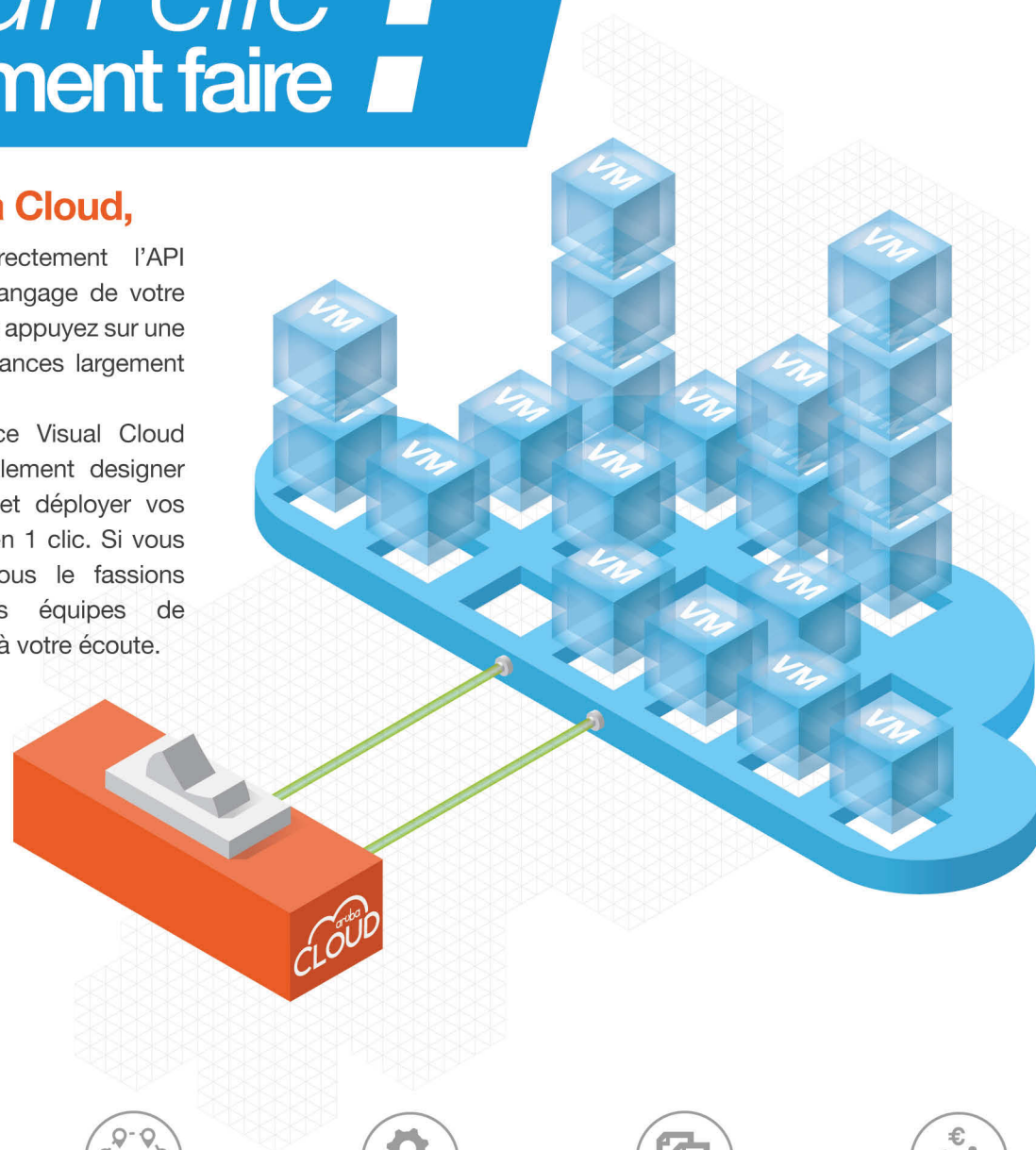


Je veux créer 20 cloud serveurs en un clic Comment faire ?

Avec Aruba Cloud,

vous utilisez directement l'API Aruba depuis le langage de votre choix et vous vous appuyez sur une base de connaissances largement documentée.

Grâce à l'interface Visual Cloud vous pouvez également designer votre datacenter et déployer vos serveurs virtuels en 1 clic. Si vous souhaitez que nous le fassions pour vous, nos équipes de Consultance sont à votre écoute.



3
hyperviseurs



6 datacenters
en Europe



APIs et
connecteurs



70+
templates



Contrôle
des coûts

1

Quitte à choisir une infrastructure IaaS, autant prendre la plus performante!
Aruba Cloud est de nouveau **N°1 du classement des Cloud**
JDN / CloudScreener / Cedexis (octobre 2014)

Contactez-nous!

0810 710 300

www.arubacloud.fr

aruba
CLOUD

Cloud Public

Cloud Privé

Cloud Hybride

Cloud Storage

Infogérance

MY COUNTRY. MY CLOUD.*

Anonymous : Internet est à nous !



Véronique Loquet
Spécialiste de l'Open Source et
de la sécurité des SI.
Fondatrice de l'agence RP AL'X
Communication.
sur Twitter @vloquet

Eux, ... C'est toi, c'est moi

Anonymous, c'est une mosaïque d'individus censés exercer leur libre choix dans une société où se délite le concept d'Etat-nation.

Hacktivistes trolleurs sous le masque de Guy Fawkes et dont l'adresse commune est Internet. Le mouvement agit sans leader, unis sous un même slogan : *"Nous sommes Anonymes. Nous sommes Légion. Nous ne pardonnons pas. Nous n'oublions pas. Redoutez-nous"*.

Caractérisé par l'anonymat de ses membres, le collectif échange sur les plates-formes de partage dont la plus connue est 4chan.org. Actif sur le canal IRC, diffusant ses vidéos via Pastebin, il maîtrise un arsenal de canaux, pour fédérer ses membres et amplifier ses actions, mêlant le viral aux modèles de diffusion traditionnels.

Lanceurs d'alertes, justiciers, idéologues, altermondialistes, crypto-arnarchistes...

Empreintes de subculture à la mode lulz, les énergies rassemblées se muent le plus souvent en actions de déni de service et autre défacement (NDLR : anglicisme pour désigner une modification pirate de la page d'accueil d'un site Web). Les cibles d'attaques varient selon les campagnes de revendications, Etats, dictatures, entreprises ou administrations. Depuis 2008 la notoriété des Anonymous n'a cessé de croître. Un palmarès d'exploits jalonné de faits d'armes décisifs, comme le projet Chanology, contre l'église de scientologie, HBGary Federal, expert en sécurité informatique, qui a retrouvé le contenu de son serveur de messagerie sur le Net alors qu'il menait des investigations sur les Anonymous pour le compte de l'administration américaine... Moultes opérations ont suivi : #opsony, lancée en représailles aux mesures anti-piratage de la firme nipponne, #Blackout en opposition aux projets de lois liberticides PIPA et SOPA, l'opération MegaUpload suite à la décision du FBI de fermer la célèbre plate-forme, le mouvement social international #occupy, l'apport d'un soutien technique aux indignés, ou plus récemment #OpGPII (Grands Projets Inutiles et Imposés) contre le ministère de la défense afin de venger la mort de Rémi

Fraïsse, tué par une grenade offensive militaire lors des affrontements entre opposants au projet du barrage de Sivens.

Quand il s'agit de se retrouver IRL, chaque individu incarne le groupe. On trouve une charte de 22 règles, destinées au code de conduite des Anons exfiltrés dans les manifestations publiques. Ces règles indiquent aux militants les basiques utiles à garantir leur succès sans enfreindre les lois.

La liberté d'expression est l'une des valeurs fondamentales du mouvement. Face à la censure, les Anonymous ont mené d'intenses ripostes contre des gouvernements, en Algérie, Arabie Saoudite, Égypte, Iran, Lybie, Tunisie, Yemen ou Venezuela.

L'Internet post-Charlie est-il celui de la guerre cybernétique ? Quel équilibre entre sécurité et liberté ?

Le 7 janvier dernier l'assassinat des caricaturistes de Charlie Hebdo fait réagir le mouvement. Une première vidéo est postée, rappelant que la liberté d'expression est un principe non négociable. L'annonce promet des attaques massives et frontales.

L'#opCharlieHebdo est lancée. Une liste de près d'un millier de profils Facebook et de comptes Twitter en lien avec le djihadisme radical est diffusée. Inondés de requêtes, des sites de propagande fondamentaliste sont tombés, défacés ou redirigés. Si plus que jamais nous devons revendiquer notre liberté d'expression, l'angélisme n'est pas de mise. Or, si elles sont louables les initiatives citoyennes des Anonymous peuvent parfois être contre-productives. Le risque de ces attaques est souvent sous-estimé, quels sont les dommages réels ? Quid de la désinformation ? Quel est le risque d'infiltration par des groupes criminels, ou de manipulation par des services de renseignement ? L'#opCharlieHebdo modifie considérablement les conditions de travail des enquêteurs et ne diminue pas pour autant la présence des radicaux sur Internet. Plus néfaste encore, elle a engagé des réponses offensives sous tous azimuts et des menaces nourries via twitter et le hashtag #Op-France. Quelques heures ont suffi pour faire tomber plus de 25000 sites français. Des sites étatiques, ceux des secteurs éducation et santé. Touchés, BNP, CNRS, Restos du cœurs, ou encore des pages Facebook, comme celle de la Cathédrale de Vannes, ou le compte Twitter du quotidien Le Monde.

Ces offensives ont été revendiquées par une

trentaine de groupes de pirates en informatique comme AnonGhost, Felaga team, Makers Hacker Team, MECA, Prodigy TN, Virus003, Votr3x... localisés en Algérie, au Maroc, en Mauritanie, Indonésie... d'autres n'ont pas revendiqué et certains seraient des militants d'extrême droite.

Pas simple d'y voir clair.

Au lendemain des attentats le gouvernement passe son nouveau décret d'application à Bruxelles. Le blocage administratif des sites est voté, il donne désormais la capacité au ministère de l'intérieur de décider, sans l'intervention du juge, quels sites seront censurés. Ces mesures devraient logiquement renforcer l'utilisation de postures en mode furtif. Pour contourner la censure, les tutoriels dédiés aux militants radicaux fleurissent. Tor vient tout juste d'annoncer qu'il continuait ses efforts d'amélioration pour permettre à de plus en plus de gens de l'utiliser avec succès, malgré les tentatives de certains gouvernements à le bloquer.

Ces dispositifs d'anonymisation ou de chiffrement révélés par le réflexe de censure ne sont-ils pas eux aussi contre-productifs ? La surveillance de masse attentatoire aux valeurs fondamentales semble aller dans le sens d'une généralisation. Confier à Facebook ou Twitter la responsabilité de détecter les

« contrevenants », en leur donnant des pouvoirs policiers est-elle une saine solution ? Dans le même temps on apprend la lourde condamnation d'un journaliste américain, Barrett Brown.

En lien avec Anonymous, il est condamné à une peine de 5 ans de prison et 890000 dollars d'amende pour publication de données volées et entrave à la justice.

La violence des réactions de toutes parts, et celle de la répression menace l'équilibre de nos sociétés. Les actions menées peuvent à la fois servir la démocratie et la pire tyrannie. Le philosophe et sociologue polono-britannique, Zygmunt Bauman, avait un jour déclaré : « La sécurité sans la liberté, c'est la prison ou l'esclavage. La liberté sans la sécurité, c'est le chaos et la jungle. »

Au moment où nous bouclons le journal, une seconde vidéo d'Anonymous est postée #opISIS affirme l'éradication de plusieurs centaines de comptes Twitter et Facebook de recruteurs djihadistes.

Le message à l'attention des cibles est clair : *« Nous vous traiterons comme un virus, dont nous sommes le remède. Internet est à nous ».*

Suite page 14.....



LE CLOUD GAULOIS, UNE RÉALITÉ ! VENEZ TESTER SA PUISSANCE

EXPRESS HOSTING

Cloud Public
Serveur Virtuel
Serveur Dédié
Nom de domaine
Hébergement Web

✉ sales@ikoula.com
☎ **01 84 01 02 66**
🌐 express.ikoula.com

ENTERPRISE SERVICES

Cloud Privé
Infogérance
PRA/PCA
Haute disponibilité
Datacenter

✉ sales-ies@ikoula.com
☎ **01 78 76 35 58**
🌐 ies.ikoula.com

EX10

Cloud Hybride
Exchange
Lync
Sharepoint
Plateforme Collaborative

✉ sales@ex10.biz
☎ **01 84 01 02 53**
🌐 www.ex10.biz

1&1 SERVEUR CLOUD

NOUVELLE GÉNÉRATION

Easy to use – ready to cloud*

Le nouveau serveur Cloud 1&1 procure une combinaison parfaite entre la performance d'un hardware dédié et la flexibilité du Cloud !

FLEXIBLE & ABORDABLE

Configuration individuelle

- CPU, RAM et stockage SSD sont configurables indépendamment et en toute flexibilité afin de s'adapter au mieux à vos besoins

Transparence des coûts

- **NOUVEAU** : facturation à la minute
- **NOUVEAU** : facturation détaillée, claire et structurée pour une totale maîtrise de votre budget

SIMPLE & SÛR

1&1 Cloud Panel

- **NOUVEAU** : l'interface innovante et conviviale facilite la gestion de votre serveur

Sécurité

- Les data centers haute performance de 1&1 comptent parmi les plus sûrs en Europe
- Les sauvegardes quotidiennes et snapshots vous protègent de la perte de données
- Le firewall intégré bloque les attaques contre votre serveur

TOUT INCLUS

Haute performance

- **NOUVEAU** : votre serveur Cloud livré en moins d'1 minute
- **NOUVEAU** : technologie SSD pour une performance maximale
- **NOUVEAU** : réseau privé VLAN, API, load balancing, firewall et de nombreuses autres fonctionnalités facilement configurables
- **NOUVEAU** : virtualisation reposant sur la technologie leader VMware
- **NOUVEAU** : applications incluses, prêtes à l'emploi : WordPress, Drupal™, Magento®
- Parallels® Plesk 12
- Trafic illimité



DOMAINES | MAIL | HÉBERGEMENT | E-COMMERCE | SERVEURS



Saisissez votre email et testez immédiatement



1 MOIS D'ESSAI GRATUIT !*

☎ 0970 808 911
(appel non surtaxé)

*Facile à utiliser – prêt pour le Cloud.
1&1 Serveur Cloud : 1 mois d'essai gratuit, sans demande de vos coordonnées bancaires, puis à partir de 15,84 € HT/mois (19,01 € TTC) sur la base de la configuration minimum. Pas de frais de mise en service. Conditions détaillées sur 1and1.fr.

1&1

1and1.fr

.... suite de la page 10

DDoS, l'attaque informatique privilégiée des Anonymous

L'attaque distribuée par déni de service (DDoS - Distributed denial-of-service) n'est pas nouvelle mais elle est en forte augmentation et son niveau de risque plus élevé. Elle a pour effet d'anéantir les serveurs d'une paralysie éphémère. Ses objectifs ne sont pas le vol de données ou leurs altération, le but est plutôt de nuire à la réputation ou de déclencher une perte financière en provoquant un ralentissement, une saturation, voire un crash de système.

L'étude annuelle de Arbor networks sur la sécurité des infrastructures IT mondiales, indique que les signalements d'attaques DDoS sont passés de 8 Gbit/s à 400 Gbit/s. La disponibilité d'une bande passante plus large est l'une des raisons de la multiplication par 50

du nombre de ces attaques en 10 ans, au même titre que l'exploitation de techniques dévastatrices comme l'amplification par réflexion.

Ce type d'attaque nécessite peu de ressource, mais une phase d'analyse de la cible pour y recenser les vulnérabilités exploitables, la compromission de nombreuses machines, l'hébergement des zombies et le lancement des ordres... Bref le plus simple, script-kiddies ou pas, reste l'utilisation des nombreux outils prêts à l'emploi comme HOIC (High Orbit Ion Cannon), déclenchable directement par un navigateur Web.

L'autre technique privilégiée est le défacement, là encore l'exploitation d'un site vulnérable ne requiert pas de compétence poussée en hacking. La plupart des CMS comme Drupal, Magento, Joomla ou Wordpress, ne sont pas tous très à jour des correctifs de vulnérabilités.

Bien sûr, d'autres techniques plus complexes sont utilisées par des membres chevronnés. Chacun joue avec ses compétences, c'est la do-ocratie qui prime.

Aucune mesure ne peut absolument prémunir d'une offensive, en revanche il est fortement recommandé d'observer une bonne hygiène informatique pour la prévention. Sécuriser les machines et les réseaux présentant des failles connues et dont les correctifs sont disponibles. L'application des patches permettra déjà de combler un grand nombre de failles pouvant être exploitées. Cependant nombres de systèmes sont compromis sans que les vulnérabilités puissent être détectées.



Rendez-vous pour le second volet « Contre-mesures » dans le prochain numéro de Programmez!

Ghost : une grave faille dans la glibc de Linux



Frédéric Mazué
Programmez!

Une nouvelle faille dans une librairie Linux a fait faire des cauchemars à nombreux administrateurs. La faille Ghost a été découverte dans la glibc de Linux. Cette librairie est présente dans tous les systèmes Linux. Selon l'éditeur de solutions de sécurité Qualys qui a publié des informations concernant cette faille, celle-ci est présente depuis la glibc-2.2 qui a été publiée le 10 novembre 2000. Depuis 2013, d'autres correctifs ont été apportés à cette librairie qui peuvent atténuer la gravité de la faille. La vulnérabilité se situe dans les fonctions `gethostbyname` et `gethostbyaddr`. Elle est exploitable par une classique attaque en débordement de tampon et permet potentiellement la prise de contrôle total, à distance, de la machine attaquée. Si l'exploit n'est pas très facile à réaliser, en revanche, il ne nécessite aucun droit dans le système ni aucune connexion au système préalables. Il suffit de provoquer une résolution DNS sur un nom de domaine mal formé. Ceci peut, selon la configuration du serveur attaqué, être obtenu par exemple par le simple envoi d'un mail. Les administrateurs système doivent vérifier d'urgence si leurs systèmes sont vulnérables et de toutes façons appliquer d'urgence le correctif, que la plupart des distributions Linux ont déjà mis à disposition.

Un code concret

Le code C ci-dessous, `test-ghost.c`, permet de tester la vulnérabilité des systèmes et l'efficacité des correctifs appliqués.

Compiler ainsi :

```
gcc test-ghost.c -o test-ghost
```

Puis lancer le test ainsi :

```
./test-ghost.
```

Le message `Not vulnerable` doit s'afficher après application du correctif de sécurité.

```
#include <netdb.h>
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <errno.h>
#define CANARY "in_the_coal_mine"
struct {
    char buffer[1024];
    char canary[sizeof(CANARY)];
} temp = { "buffer", CANARY };
int main(void) {
    struct hostent resbuf;
    struct hostent *result;
    int hermo;
    int retval;
    /** strlen (name) = size_needed - sizeof (*host_addr) - sizeof (*h_addr_ptrs) - 1; */
    size_t len = sizeof(temp.buffer) - 16 * sizeof(unsigned char) - 2 * sizeof(char *) - 1;
    char name[sizeof(temp.buffer)];
    memset(name, '0', len);
    name[len] = '\0';
    retval = gethostbyname_r(name, &resbuf, temp.buffer, sizeof(temp.buffer), &result, &hermo);
    if (strcmp(temp.canary, CANARY) != 0) {
        puts("vulnerable");
        exit(EXIT_SUCCESS);
    }
    if (retval == ERANGE) {
        puts("not vulnerable");
        exit(EXIT_SUCCESS);
    }
    puts("should not happen");
    exit(EXIT_FAILURE);
}
```

Qualys a été le premier à signaler la faille. Les codes sont extraits de <http://www.openwall.com/lists/oss-security/2015/01/27/9>



Offre spéciale abonnement !

et aussi...

PROGRAMMEZ!

le magazine du développeur

devolo
The Network Innovation



Pour un abonnement
2 ans, Devolo
et Programmez!
vous offrent un kit
complet CPL
dLAN 550
d'une valeur de 79,90 €

**Pour
Seulement 94,90€**

(au lieu de : 158,90 €, abonnement
2 ans / 22 numéros : 79 € + 15,90 €
de frais logistiques et postaux)

Attention :

- cette offre est strictement limitée à la France Métropolitaine et à la Corse.
- quantité limitée, jusqu'à épuisement des stocks. cette offre est susceptible de s'arrêter à tout moment.

2 ans 22 numéros
94,90€

1 an 11 numéros
49€
seulement (*)

Spécial étudiant
39€
1 an 11 numéros

2 ans 22 numéros
79€
seulement (*)

(*) Tarifs France métropolitaine

ABONNEZ-VOUS !

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 € (au lieu de 65,45 €, prix au numéro)

☐ Abonnement 2 ans au magazine : 94,90 € (au lieu de 158,90 €, kit CPL dLAN offert)

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

Offre abonnement + clé USB Programmez!

☐ 1 an (11 numéros) + clé USB : 60 €

Clé USB contenant tous les numéros de Programmez! depuis le n°100, valeur : 29,90 €

Tarifs France métropolitaine

☐ M. ☐ Mme ☐ Mlle 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

email indispensable pour l'envoi d'informations relatives à votre abonnement

La place des développeurs de jeux vidéo aujourd'hui

Il y a différents types de développeurs dans le monde du jeu vidéo. Vous pouvez assez facilement les deviner en regardant les jeux de ces dernières années. On peut séparer ces développeurs en trois grosses catégories : moteur, middleware et gameplay.



Benjamin Baldacci
Responsable Pédagogique
Game Programming
Isart Digital

Moteur

Le moteur de jeu est un ensemble de briques, qui, une fois assemblées, permettent de créer un monde virtuel plus ou moins réaliste. Il permet aux équipes de game designers/développeurs de se concentrer sur le contenu et le déroulement du jeu, et non sur la technique de rendu 3D, ou la façon de faire transiter les données via le réseau, etc.

Il y a bien sûr plusieurs types de développeurs moteur : 3D, physique, son, réseau, etc.

Les moteurs d'aujourd'hui doivent être multiplateformes, ils doivent aussi bien tourner sur PC, Xbox360, PS3, XboxOne, PS4, WiiU, PSVita ou 3DS. Pour cela les développeurs moteur travaillent sur du code spécifique à chaque plateforme et utilisent le langage de programmation adéquat à chacune d'entre elles (C/C++ pour la plupart).

Middleware (outils/logiciels)

Le middleware est un ensemble d'outils, mis à disposition des game designers, développeurs gameplay, artistes (graphistes, sound designers). Il permet la création des niveaux, le développement du gameplay, l'intégration de modèles 3D, de musiques et de sons, etc.

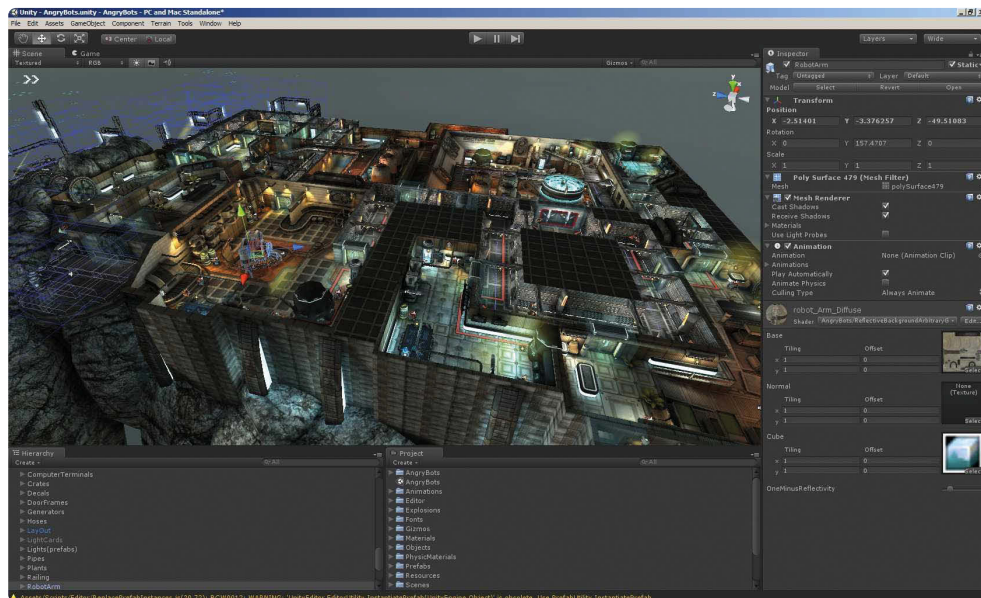
Le développeur middleware travaille donc sur l'amélioration de ce dernier, que ce soit sur la partie éditeur du monde pour les game designers, ou sur la partie compréhension des règles développées par les développeurs gameplay. Les principaux langages de programmations utilisés pour les middlewares sont le C++ et le C#.

Gameplay

Le gameplay est la perception qu'a un joueur lorsqu'il est en train de jouer.

Pour la programmation gameplay, on parle généralement des 3C : camera, character, control (caméra, personnage, contrôle).

La caméra peut être différente suivant le type de



jeu. Les trois types de caméras les plus utilisées sont : la 1ère personne, la 3ème personne et la top view (vue de dessus). La caméra à la 1ère et 3ème personne sont le plus souvent utilisées pour des jeux de tir ou d'aventure, quant à la caméra en top view, elle est utilisée pour les RTS (Real-Time Strategy).

Les contrôles sont bien évidemment les manettes, clavier, souris et autres pads. Leur maniabilité doit être adaptée au jeu, mais aussi au niveau du joueur (certains jeux proposent des contrôles différents suivant le niveau de jeu sélectionné).

Le personnage quant à lui a des actions possibles ; la marche, la course, le saut, le fait de pouvoir s'accrocher au rebord d'une falaise, de tenir une hache dans chaque main, etc.

Le gameplay doit être le même, peu importe la plateforme, c'est pour cela que le développeur gameplay utilise un middleware (qu'il soit interne à la société ou externe tel qu'Unreal Engine, Unity3D ou CryEngine). Son langage de développement dépend donc du middleware choisi (Unreal Engine : C++ ; Unity : C#, Javascript, Boo ; CryEngine : LUA).

Les compétences

Un métier est avant tout basé sur des compétences. Celles qui sont demandées aux développeurs

dans le jeu vidéo varient bien évidemment suivant le type de développement.

Le développeur moteur doit être très bon en C et C++. Il doit savoir lire et comprendre une documentation technique (en anglais). Les optimisations spécifiques aux plateformes sont importantes pour qu'un moteur de jeu soit puissant, il doit donc connaître les spécificités des plateformes sur lesquelles il travaille.

Le développeur middleware doit quant à lui pouvoir s'adapter, dialoguer, comprendre les besoins. Il doit s'adapter et comprendre le moteur de jeu afin de pouvoir construire les outils qui se reposent dessus. Mais le dialogue et la compréhension des besoins des game designers et des développeurs gameplay sont très importants pour permettre de faire les outils les plus pertinents.

Les développeurs gameplay sont un peu dans le même cas que les développeurs middleware, ils doivent constamment échanger avec les game designers afin de répondre à leurs besoins, ainsi qu'avec les testeurs, afin de prendre en compte leurs retours.

La technique est omniprésente pour ces trois métiers, mais ce n'est pas la seule compétence majeure. Un développeur de jeux vidéo – qu'il soit moteur, middleware ou gameplay – doit être créatif. Ce sont les programmeurs "moteur" qui font preuve de créativité lorsqu'ils mettent au point un nouveau système lumière permettant de faire de la réflexion réaliste. Quant aux développeurs



loppeurs middleware, ils doivent de façon constante faciliter la vie des autres corps de métiers en créant des outils utiles et efficaces. Pour finir, la créativité des développeurs gameplay se mêle à celle des game designers afin de rendre le jeu le plus plaisant possible.

LES MÉTIERS AUJOURD'HUI ?

Comprendre hier avant de voir aujourd'hui

Le jeu vidéo n'a pas de date d'apparition précise. Le jeu Pong (créé en 1967) n'est pas le premier, mais c'est lui qui lança (dans une moindre mesure) l'industrie vidéo-ludique.

À cette époque pas de framework de développement et encore moins d'outils comme les moteurs de jeu.

Au fur et à mesure des années des sociétés se sont fondées autour de ce secteur naissant, comme Nintendo, Ubi-Soft, etc.

Chacune d'entre elles développait uniquement en interne ses propres outils, encore loin de ceux d'aujourd'hui, mais surtout sans avoir de séparation entre moteur de jeu et le jeu en lui-même.

C'est dans le milieu des années 90 que le terme de « moteur de jeu » est apparu.

Il fût introduit principalement par idSoftware (Doom) et Epic Games (Unreal). Cela permit à ces deux sociétés de réutiliser une partie de leur code pour leurs prochains jeux.

Continuant dans leur lancée, Epic Games commença à commercialiser son moteur de jeu « Unreal Engine » dans la fin des années 90. Le coût d'achat pour un tel moteur pouvait varier entre dix mille et plusieurs millions de dollars.



Ce qui a changé la donne Un nouvel arrivant

De la fin des années 90 au milieu des années 2000 les seules sociétés développant des moteurs de jeu étaient les studios eux-mêmes.

En 2005 une société nommée « Unity Technologies » annonce le lancement du moteur de jeu « Unity3D » à la Worldwide Developer Conference d'Apple. Cette société ne développait aucun jeu, uniquement un moteur.

En 2009 Unity Technologies annonce la mise à disposition d'une version gratuite de son moteur à sa conférence annuelle « Unite ». Unity3D est alors en version 2.6.

Le prix

Unity Technologies n'est pas arrivé qu'avec une version gratuite. Alors qu'Epic Games vendait son moteur Unreal Engine 3 plusieurs centaines de milliers de dollars, Unity Technologies lança le sien au prix de 1500 dollars. Ce prix a évidemment une contrepartie, là où avec Unreal Engine 3 le code source du moteur est modifiable, Unity ne donne aucun accès au code source (laissant tout de même aux développeurs des points d'entrée au travers de leur API de plugins).

Le prix a donc fait évoluer les mœurs; le coût d'achat d'un outil comme Unity3D devenant moins élevé que le coût de développement d'outils en interne, les studios se sont mis au fur et à mesure à privilégier Unity3D à leurs propres outils. L'année dernière Epic Games réagit en annonçant la sortie d'Unreal Engine 4 pour seulement 20 dollars par mois (code source et mises à jour compris).

L'évolution des métiers de développeur

Les moteurs de jeu devenant de plus en plus abordables, les équipes de développements

évoluent. Dans les années 2000, les équipes étaient constituées de développeurs moteur, outils et gameplay.

Aujourd'hui les studios indépendants (ou petits studios) manquent de temps pour développer leurs jeux.

Le temps de développement d'un moteur de jeu est considérable, cela se compte en années. C'est pour cette raison qu'ils se sont tournés vers l'utilisation de moteurs comme Unity et Unreal Engine.

Les développeurs moteur se font donc de plus en plus rares dans les petites structures.

Toute la partie de développement outil qui était liée au développement moteur n'existe-t-elle donc plus dans ces studios ? Oui et non.

En effet, n'ayant plus de moteur interne, les outils liés à celui-ci n'ont plus lieu d'être. Unity et Unreal Engine sont très efficaces, il n'y a aucun doute là-dessus, mais ne seront jamais assez complet pour tout le monde.

C'est ici que le développeur outil intervient, il développe des outils (ou améliore l'existant) directement pour Unity ou Unreal Engine.

Le développeur gameplay quant à lui ne change pas ses habitudes, mis à part le fait que les outils mis à sa disposition sont de plus en plus performants et simples à prendre en main.

Les développeurs sont-ils amenés à disparaître ?

Avec les évolutions des moteurs comme Unity ou Unreal Engine, on pourrait être amené à se dire que le métier de développeur de jeux vidéo semble en danger.

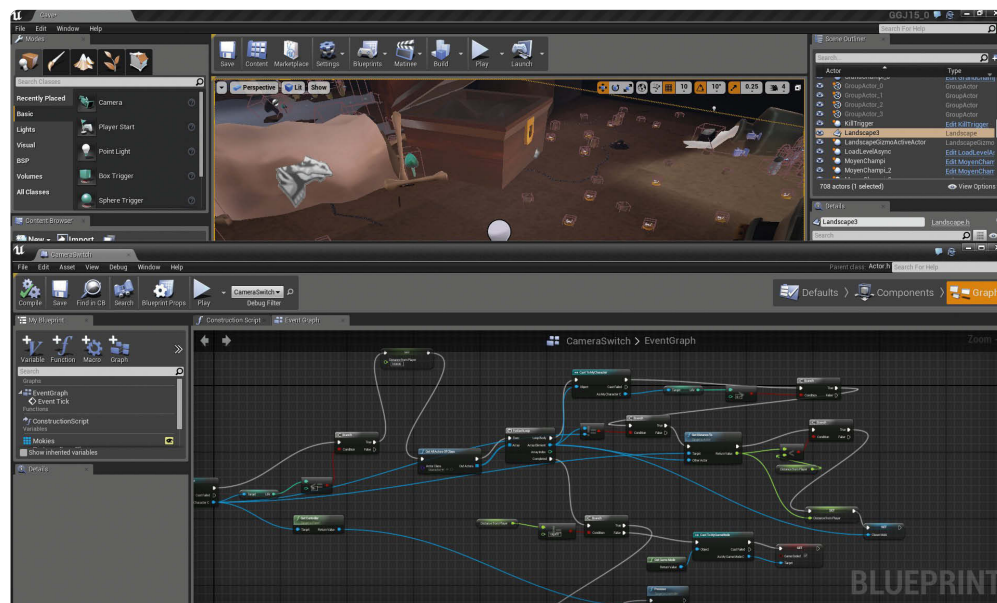
Une restructuration serait plus juste. Le métier de développeur moteur se raréfie dans l'ensemble des studios, pour mieux se retrouver dans des studios spécialisés (comme Epic Games, Unity, ...).

Le métier de développeur outils s'étend au développement d'outils pour des moteurs existants.

Mais cette « restructuration » ne date pas d'aujourd'hui.

Elle est présente depuis que le premier jeu a été créé. Aucun métier ne prédestinait à devenir développeur de jeux vidéo à cette époque. Il n'y avait donc que des « développeurs de jeux ».

Les technologies évoluant, les équipes de productions grossissant, les spécialités de développement moteur, développement outil et développement gameplay sont apparues et ne sont pas prêtes de disparaître.



Devenir développeur en jeux vidéo ?



Le jeu vidéo attire beaucoup les développeurs et les jeunes diplômés. Est-ce pour autant un monde tout beau et tout gentil ? La compétition est rude entre développeurs. Les conditions de travail et les salaires ne sont pas forcément à la hauteur des attentes. Questions – réponses avec David Bilemdjian (responsable pédagogique game design & programming à ISART Digital).

Le jeu est un domaine où la demande reste forte dans le développement, est-ce que cela a augmenté les demandes d'entrée dans l'école ?

David Bilemdjian : le développement informatique dans le jeu vidéo attire en effet d'année en année davantage de jeunes. La population dans son ensemble a pris conscience collectivement et individuellement des opportunités offertes par l'activité de développement informatique en général : opportunités économiques, salariales, de carrière, intellectuelles, etc. Il n'y a qu'à voir l'engouement suscité au niveau politique pour cette activité, nouveau héraut de la croissance, au point de vouloir l'enseigner dès le plus jeune âge. Il y a encore vingt ans l'informaticien avait une image d'ours atrabilaire, associable, empêtré dans ses problématiques technico-techniques totalement abscones. Ce n'est plus le cas aujourd'hui et le développeur informatique a gagné un véritable statut, presque désirable. Courtisé par les entreprises, il est de plus en plus en position de force et recourt même parfois aux services d'un agent spécialisé, à l'instar des vedettes de cinéma ou des sportifs de haut niveau, pour gérer sa carrière. Les réussites flamboyantes des entreprises de la Silicon Valley (Google, Apple, Facebook) ont transfiguré leurs créateurs en héros, voire en demi-dieux auxquels notre époque voue un culte démesuré : le geek est devenu cool et puissant à la fois, détenteur d'un pouvoir quasi surhumain. Le jeu vidéo a également contribué au dépoussiérage de l'image vétuste de l'informaticien : comment des développeurs de produits cools (les jeux vidéo) pourraient-ils être vus autrement que comme des êtres cools ? La demande de formation en développement informatique croît donc mécaniquement avec l'attrait du métier de développeur. Les besoins des industries secondaire et tertiaire pour ces profils sont également en hausse constante : l'informatique n'a jamais été aussi présente dans tous les domaines de l'activité humaine, et cette tendance n'est pas près de s'inverser, les avancées technologiques récurrentes assurant un renouvellement constant de la

demande en cerveaux frais.

Pour les nouvelles générations et leurs parents, devenir développeur informatique, c'est donc l'assurance de décrocher un « travail sympa, intéressant et rémunérateur ». Imaginez alors ce que peut représenter le développement informatique pour le jeu vidéo. Il s'agit pour beaucoup de jeunes du Graal, la combinaison parfaite d'un métier et d'une passion ludique.

Les jeunes entrants savent-ils réellement ce qu'il se passe derrière un jeu vidéo (desktop, console, mobile) ?

David : les jeunes ont les moyens aujourd'hui, grâce à Internet, de connaître l'envers du décor pour à peu près tout ce qui touche à l'informatique. L'information est disponible, ce n'est qu'une question de volonté. Les jeunes curieux s'informent donc et certains amorcent même en autodidacte une activité de développement logiciel, que ce soit du développement Web, applicatif ou de jeu vidéo. Les entrants à ISART DIGITAL sont donc bien souvent affranchis sur de nombreux points, même si leur pratique, les outils et processus de production suivis sont amateurs. La présentation d'un book de développement est un atout indéniable et un critère positif de sélection des candidats désireux de suivre une formation à l'école. Un étudiant qui entre à ISART DIGITAL a déjà donc amorcé un tant soit peu sa mue du « simple » joueur au développeur.

Est-ce que le jeu, comme d'autres secteurs numériques, s'est-il segmenté et spécialisé (dans les profils, les postes, les compétences) ?

David : les profils de développeurs sur un projet de jeu vidéo varient considérablement selon le type de produit et sa complexité. Une grosse production d'un jeu AAA menée par un studio de taille conséquente intègrera davantage de profils spécialistes que de profils généralistes : développeur gameplay, développeur moteur (IA, rendu, réseau, audio, etc), développeur outil, développeur serveur, développeur back-office, développeur de plateforme sociale, développeur pour portage d'une plateforme à

une autre, ... à chacun sa spécialité. Les compétences nécessaires varient bien évidemment d'une spécialité à l'autre mais, bien souvent, si le développement gameplay peut être assuré par des professionnels issus de formations courtes (type Bac+2), les développements moteur & outils nécessitent des connaissances théoriques solides en sciences et en informatique qui ne peuvent être acquises que lors de formations longues (type Bac+4, Bac+5, ingénieur). Une hiérarchie pyramidale somme toute classique mais peu profonde est mise en place pour assurer la structure de telles productions : les développeurs sont encadrés par un lead-développeur, lui-même encadré par un chef de projet, un directeur technique, et/ou un directeur de production. Un développeur évoluera assez naturellement vers le poste de lead-développeur, puis de directeur technique. Celui qui souhaite, après de longues années d'activité, prendre du recul par rapport au développement pur et s'intéresser davantage aux activités managériales et stratégiques, pourra évoluer vers les postes de « Chef de projet » et « Directeur de Production ».

Les grosses entreprises se font rares et la grande majorité des entreprises en France demeurent de petite taille (<10 personnes) ou de taille moyenne (<50 personnes). Les productions de ces studios sont moins ambitieuses et moins complexes que celles des gros studios, et les budgets surtout très serrés. Un développeur est un luxe incontournable qui doit être rentabilisé au maximum. Dans ce contexte, le développeur doit se montrer polyvalent et prendre en charge bon nombre des spécialités de développement énumérées plus haut. Il n'est pas rare dans les toutes petites structures que le chef d'entreprise soit également lead-développeur.

Vu de l'extérieur, le jeu attire toujours beaucoup de développeurs, la réalité est-elle aussi excitante que cela ?

David : l'activité de développement informatique au quotidien consiste en la répétition perpétuelle d'un cycle de résolution

de problématiques plus ou moins techniques. Un bug chasse un autre bug et lorsqu'on se croit sorti d'affaires, de nouveaux obstacles surviennent qui, s'ils semblent insurmontables au premier abord, sont finalement résolus au prix d'une grande persévérance et d'une grande concentration. Si l'expérience acquise permet de minimiser les risques d'occurrence des problèmes, ceux-ci émergent inévitablement à un moment ou un autre, et surtout au moment où on les attend le moins. Le développeur, sous son air impassible, voit son état psychologique évoluer sur des montagnes russes, passant du désarroi total à l'euphorie extatique plusieurs fois dans une même journée : cela est en effet très excitant pour certains (ceux qui recherchent le challenge permanent et la complexité), usant pour les autres. Non vraiment développer n'est pas une sinécure, mais grande est la satisfaction intellectuelle procurée à chaque franchissement d'obstacle et particulièrement en fin de projet!

Pour vous, quels sont les profils ou les compétences les plus demandés ou les plus intéressants ?

David : les développeurs moteur spécialisés et compétents sont toujours très prisés par les entreprises, tant leur nombre fait défaut en France! Beaucoup partent tenter l'aventure à

l'étranger où les conditions salariales et l'environnement économique sont plus favorables au développement vidéo-ludique. Recruter de tels profils en France est un challenge en soit. Les développeurs polyvalents et expérimentés seront tout aussi appréciés par les petites structures. Les développeurs qui sauront également être force de proposition, ou tout au moins avoir un esprit critique sur des domaines du jeu vidéo qui à priori ne font pas partie de leurs compétences, comme le Game Design ou le Level Design, auront la préférence des recruteurs. Les contraintes imposées par le jeu vidéo temps réel étant très fortes (de 30 à 60 fps), le développement informatique de jeu vidéo peut être perçu comme un artisanat d'art, au bon sens du terme. Il est au développement informatique ce que l'ébénisterie est à la menuiserie : les scripts logiciels développés pour assoir le gameplay sont souvent ciselés, optimisés, ils regorgent de trouvailles techniques, ils sont de véritables bijoux d'orfèvrerie. On retrouve cette même philosophie dans le développement dit embarqué, au sein de systèmes aux ressources restreintes. Cela est moins prégnant dans le développement de Systèmes d'Information car la puissance des machines serveurs mobilisables laisse une marge de manœuvre

confortable aux développeurs moins soucieux d'optimisation. Le développeur de jeu vidéo qui adopte cette philosophie de l'artisan d'art, sans perdre de vue les contraintes budgétaires et temporelles de la production, est une ressource de grande valeur.

Quels conseils donner à des développeurs, étudiants qui veulent faire du jeu ?

David : développer avant tout :

- Ses compétences en informatique,
 - Ses aptitudes analytiques et son sens critique,
 - Ses connaissances en mathématiques
- Coder, coder, coder :
- Développer soi-même un jeu, aussi petit soit-il. S'il présente un niveau de finitions suffisantes, il sera un sésame d'une grande efficacité pour séduire des recruteurs (écoles & entreprises).
 - Pousser la compréhension des technologies et des techniques. Cela nécessite une veille importante. Lire beaucoup de documentation, écouter les conférences données par les spécialistes en place dans les entreprises leader. Décortiquer le code de projets existants et accessibles.
 - Partager et confronter son savoir avec le savoir des autres (il y a toujours quelqu'un qui aura trouvé une solution aussi voire plus élégante que la vôtre)



Tout **Programmez!** sur une clé USB

Tous les numéros de *Programmez!* depuis le n° 100.



Clé USB 2 Go. Photo non contractuelle. Testé sur Linux, OS X, Windows. Les magazines sont au format PDF.



29,90 €*



* tarif pour l'Europe uniquement. Pour les autres pays, voir la boutique en ligne

Commandez directement sur notre site internet : www.programmez.com

Les filières pour devenir développeur

De nombreux cursus en écoles d'informatique, écoles d'ingénieurs, universités, BTS et IUT permettent d'accéder au métier de développeur avec un diplôme en poche. Ils apprennent aux jeunes à apprendre, indispensable face à l'évolution des technologies et des métiers.



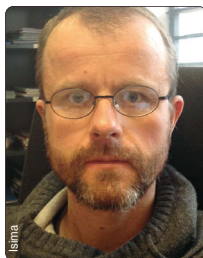
Christine Calais

« On peut tous devenir développeur, souligne Emmanuel Carli, directeur général d'Epitech. C'est comme la littérature, il faut du temps et des essais pour développer son style ! » Un autodidacte peut intégrer le marché du travail en tant que développeur, mais à certaines conditions : salaire moindre, poste en PME, évolution plus faible. Le diplôme est une garantie de qualité. Il est particulièrement valorisé dans la mentalité française. Pour accéder à un cursus de développement, le baccalauréat S est conseillé par plusieurs de nos interlocuteurs, notamment pour se garder toutes les portes ouvertes, en particulier avec l'option informatique et sciences du numérique. Mais les filières pour devenir développeur sont ouvertes à d'autres bacs : général ES ou L, technologique STI2D (Sciences et Technologies de l'Industrie et du Développement Durable) ou STMG - Sciences et Technologies du Management et de la Gestion, spécialité Systèmes d'Information de Gestion (SIG), bac pro Systèmes Electroniques Numériques (SEN). En tout cas, un esprit logique permettant d'aborder l'algorithmique de façon sereine est nécessaire ; de bonnes bases mathématiques sont toujours un plus. Les filières pour devenir développeur vont de bac+2 à bac+5 : BTS, cursus universitaires



Cours à l'Isima

général ou professionnels (DUT et licence pro), écoles d'informatique privées et écoles d'ingénieurs. « BTS et DUT forment des techniciens et des développeurs ; les écoles d'ingénieurs se proposent de former des cadres



Vincent Barra, directeur de l'Isima

capables à la fois de développer des logiciels et des outils métiers, d'encadrer des projets et des équipes de développement, et d'architecturer des solutions logicielles, » fait remarquer Vincent Barra, directeur de l'Isima, école d'ingénieur publique en informatique. Nicolas Sicard, responsable du cycle ingénieur de l'école d'ingénieurs EFREI, indique : « notre cursus permet de ne pas s'enfermer dans le métier de développeur mais de se projeter plus



Fête de promotion à l'Isima

loin, jusqu'à la fonction de DSI. 30% des cours sont orientés sur l'entreprise : communication, management, entrepreneuriat... »

Toutes les formations permettent d'acquérir les bases de la programmation, dans les principaux langages. Elles prônent des têtes bien faites et le développement des compétences humaines. Eric Lalitte, directeur d'In'Tech Info, explique : « 50% du temps est consacré aux compétences humaines et de gestion, notamment par un projet non technologique (gestion de droits, de brevets, de budget, création d'association ou d'entreprise...). » Bruno Beauvils, responsable de la licence professionnelle DA2I à l'IUT A de l'Université de Lille 1, ajoute : « La moitié du



Nicolas Sicard, responsable du cycle ingénieur de l'école d'ingénieurs EFREI.

Créer son entreprise de dev'

Les différentes filières soutiennent la création d'entreprise par leurs étudiants et diplômés. 6% des étudiants de l'EFREI créent leur entreprise à la sortie de l'école. L'Isima incube de jeunes entrepreneurs, parfois issus

de l'école, à l'intérieur de ses murs, et soutient les démarches d'entrepreneuriat (modules de formation, statut d'étudiant entrepreneur). In'Tech Info, en lien avec des partenaires financiers et le Numa aide les créateurs, notamment à valoriser leur idée au niveau marketing et financier. « A L'ESGI nous aidons les étudiants à avoir la fibre entrepreneuriale et les accompagnons par des conférences, un concours "Engrainages" et une structure d'accompagnement. » En



Samir Rinaz, directeur de l'ETNA.

février 2015, Epitech a lancé l'anti-incubateur, espace de conseils ouverts à toutes les start-up quelle que soit leur origine.

Samir Rinaz, directeur de l'ETNA, précise : « Ceux qui créent le font souvent au bout de deux ou trois ans. Je veux développer la création d'entreprises. Nous avons ainsi lancé l'incubateur Creative Valley. » Parmi les diplômés universitaires, beaucoup franchissent le pas après 4 ou 5 ans d'expérience, remarque Bertrand Mazure, de l'université d'Artois.



Bertrand Mazure, responsable du Master informatique - Parcours ingénierie logicielle pour l'Internet à l'université d'Artois



Cours à l'université d'Artois.

cursus a pour but de développer des compétences autres : gestion, droit de la propriété intellectuelle, projet de communication... Notre message aux jeunes, difficile à faire passer, est que nous développons la capacité à comprendre, pas uniquement les compétences techniques. Pour évoluer, le salarié ne peut rester purement technique, il devient manager. »

L'essor de l'alternance

Les recruteurs sont de plus en plus exigeants et apprécient davantage un jeune diplômé avec de l'expérience professionnelle, rapidement opérationnel. Aussi les formations en alternance se multiplient. A l'ESGI, les offres d'alternance fournies par 800 entreprises partenaires de l'école. L'alternance est obligatoire à l'ETNA, dont le directeur, Samir Rinaz, indique : « Nos alternants sont en blended learning, avec des cours vidéo et des séminaires certains vendredis. Si nos jeunes sont des passionnés du beau code, on leur apprend aussi à aller vite « quick and dirty », en étant fonctionnel. Nous formons des intrapreneurs, adaptables, qui font avancer l'entreprise où ils sont salariés ou qu'ils ont créée. »

L'université se professionnalise

Les cursus universitaires gommant leur image théoricienne. Bertrand Mazure, responsable du Master Informatique - Parcours Ingénierie Logicielle pour l'Internet à l'université d'Artois, met en avant : « Notre formation est construite en partenariat et en large partie dispensée par les nombreux acteurs en développement informatique de la région Nord Pas-de-Calais (Atos Worldline, Capgemini, GFI, Sopra, IBM, Alten, Nextoo, Norsys, ...). Cette proximité avec les attentes des entreprises qui embauchent nos étudiants dès leur fin de stage nous oblige à mettre à jour nos contenus régulièrement. » Bruno Beaufils, IUT A, université de Lille 1, ajoute : « la moitié de nos intervenants sont extérieurs. Si les écoles privées mettent beaucoup d'argent dans leur communication, dans une offre pléthorique, la licence professionnelle est un cursus exigeant à un rythme soutenu qui garantit de trouver du travail. Le DUT et la licence professionnelle s'adressent aux jeunes qui ont besoin d'être encadrés. »

La méthode projet reine des écoles d'informatique

La pédagogie par projets développée à l'origine à Epitech a essaimé dans les écoles d'ingénierie en informatique. Emmanuel Carli,

Où sont les femmes ?

Les jeunes femmes constituent 10% des effectifs en moyenne, de 5 à 20% selon les cursus, le taux étant un peu plus élevé en école d'ingénieur ou à l'ETNA (17%) car « l'alternance correspond à leur maturité plus forte », explique son directeur Samir Rinaz. Un chiffre stable malgré les efforts fournis par certains. Selon Eric Lalitte, « le travail doit être fait au lycée pour expliquer les métiers. Les entreprises ont leur rôle à jouer pour montrer des femmes qui ont de très belles carrières informatiques. » Isima organise des conférences avec des collégiens et des lycéens sur les métiers de l'informatique : des anciennes de l'école, des partenaires industrielles féminines, viennent présenter leur expérience et échanger. L'EFREI et l'ESIEA sont membres de l'association *Elles bougent*, qui vise à susciter des vocations d'ingénieures et de techniciennes.

Emmanuel Carli, DG d'Epitech, affirme : « Il faut leur donner des perspectives qui les intéressent, hors de l'angle technologique pur. Le numérique adresse les problèmes sociétaux et sociaux. Il permet de mieux vivre, de résoudre des problèmes du quotidien. L'analyse féminine est essentielle pour créer un environnement plus responsable. Il ne suffit pas d'inventer des nouvelles technologies, il faut également inventer les métiers de demain. »



Promotion de l'IT Paris Eiffel en présence de Max Agueh, responsable pédagogique.

DG d'Epitech, explique : « l'équipe pédagogique est là pour assister les étudiants dans leurs projets, les amener à se poser les bonnes questions et trouver les bonnes réponses. Les entreprises ont besoin de profils agiles, adaptables. L'essentiel n'est pas le problème mais l'attitude face au problème, l'esprit d'analyse. Nos axes sont autant le savoir-faire technologique que le savoir-être et l'innovation avec la création du hub d'innovation en avril 2014 : 200 projets innovants sont en cours depuis lors. L'ADN des étudiants d'Epitech, c'est d'être curieux. Nous leur fournissons l'environnement pour réussir. » A In'Tech Info, les projets longs sont privilégiés. A l'ESGI, les étudiants réalisent 10 à 15 projets par an, seul ou en groupe. Pour cela, ils ont accès à des laboratoires pédagogiques, dédiés à la veille, l'expérimentation et la recherche appliquée, avec du matériel de pointe.

Les débouchés

Après un bac+2 ou bac+3, l'étudiant aura accès à des postes de techniciens à 25/30 K€ de

salaire annuel brut, tels que développeur d'applications, pour mobile, le Web... Après un bac+5, des postes de cadres sont offerts, entre 30 et 40 K€ : architecte en systèmes d'information ou logiciel, ingénieur études et développement ou logiciel, expert technique, responsable d'applications, chef de projet logiciel.

Les développeurs continuent à se former quand ils travaillent. Pour Max Agueh, responsable pédagogique de l'IT Paris Eiffel, « il faut être très curieux pour exercer une veille technologique permanente. » La communauté des informaticiens est aujourd'hui très active sur Internet, permettant de faire de la veille pendant ou après ses études. Les cours en ligne ouverts et massifs (MOOC) sont aussi un bon outil d'autoformation. In'Tech Info prépare des SPOC (Small Private Online Classes), courts en ligne où les internautes étudiants sont suivis à distance.

IT Eiffel

Ce que veulent les jeunes développeurs

Diplômés ou étudiants, femmes et hommes, ils ont choisi des voies différentes après leur baccalauréat pour devenir développeur. Leur point commun : le goût de la programmation.



Christine Calais

Nous avons dressé le portrait d'une nouvelle génération de développeurs, entre 20 et 30 ans, qui a grandi avec les nouvelles technologies. Aujourd'hui, ils étudient ou sont passés par un BTS, une université ou une école d'informatique.

BTS Informatique, un bon point de départ

Les diplômés de baccalauréats technologiques sont de plus en plus nombreux à poursuivre leurs études après un BTS Informatique.

Keerthana Marie est aujourd'hui en 1ère année

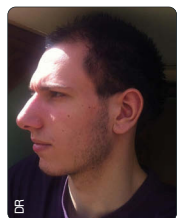


Keerthana Marie en Master 1 en développement chez Supinfo.

de Master en développement chez Supinfo, en contrat professionnel chez Capgemini pour le client ERDF pour qui elle fait du développement en Java EE en méthode agile (scrum). Après un baccalauréat

technologique, Gestion des Systèmes d'Informations (GSI) en 2011, elle voulait étudier en alternance ; « *Beaucoup de membres de ma famille, hommes comme femmes, travaillent dans l'informatique. J'ai développé en html seule au lycée.* » Aussi elle passe un BTS Services Informatiques aux Organisations, Solutions Logicielles et Applications Métiers (SIO SLAM) en travaillant chez BNP Paribas. Elle obtient en 2014 le Bachelor Informatique, spécialité Développement Web et Mobilité à l'IT Paris Eiffel, l'école d'informatique du groupe ECE. Elle apprécie le côté pratique de ses études, qui permet de mieux se débrouiller en entreprise : « *A Sup Info, le Master est moins technique que le BTS et le Bachelor, avec plus de management de projet. Je suis dans une classe avec un rythme spécial, où tous les étudiants travaillent pour Cap Gemini. J'aime diriger les projets.*

Après mes études, j'aimerais travailler à Londres



Christopher Maugis, étudiant en Bachelor à l'IT Paris Eiffel.

dans le développement, puis dans la gestion et la conception de projet. » Christopher Maugis a un parcours similaire. Après les mêmes bac et BTS que Keerthana Marie, il suit également le Bachelor de l'IT Paris Eiffel, en

alternance chez 3 octets, société de création de sites Web : « *Nous alternons trois jours par semaine en entreprise et deux jours à l'école, avec un projet de programmation en fil rouge. Mon entreprise prend en charge les frais de scolarité.* »

Les étudiants à bonne école... à l'université !

Après un bac général, certains s'orientent directement ou non vers l'université, filière de qualité et peu onéreuse, via une licence professionnelle ou une licence générale.



Edouard Vyncke, en licence professionnelle Systèmes Informatiques et Logiciels Développement et Administration Internet et Intranet (DA2I), à l'IUT A de l'Université de Lille.

Edouard Vyncke, après un bac S en 2011, s'inscrit en DUT Informatique à l'IUT A de l'Université de Lille. « *L'IUT est très encadré. Il repose sur le travail en binôme et la pratique de différents langages informatiques. Je suis aujourd'hui en licence professionnelle Systèmes Informatiques et Logiciels Développement et*



Deborah Gilbert, ingénier études et développement chez Nextoo, diplômée du Master 2 ILL de l'Université d'Artois.

Administration Internet et Intranet (DA2I), dans la lignée du DUT. Nous avons cette année des cours de communication et un projet événementiel à réaliser, un concert pour ma part. » Dans le cadre de son stage de licence chez Norsys, Edouard Vyncke va travailler sur la nouvelle version d'un outil de gestion de la formation interne de la SSII. Son stage de DUT était analyste développeur au Québec chez NetCommunications. « *La licence*

professionnelle est reconnue, et les stages sont souvent de la pré-embauche. »

Deborah Gilbert a décroché son premier CDI en octobre 2014 en tant qu'ingénieur études et développement chez Nextoo, où elle fait de la tierce maintenance applicative pour un client dans la vente. Après un bac S obtenu en 2008 et une 1ère année en DUT physique-chimie, elle se réoriente en licence d'informatique à l'université d'Artois, puis obtient son Master 2 en Informatique - Ingénierie du Logiciel en 2014. Elle a effectué ses stages en Master chez Worldline, groupe

Atos, le premier en développant une application mobile pour la santé, le second en développant sous Java. Elle est la seule femme de la promotion 2014 : « *Je regrette que l'informatique soit perçue comme un métier de garçons et qu'on ne soit pas assez informé au lycée.*

L'informatique n'est pas geek ! Si j'aime les maths, il n'est pas forcé d'être bon en maths. Il faut être logique et être bon en algorithmique pour réussir. » Elle apprécie l'autonomie que donne l'université : « *Quelle que soit la filière choisie, l'informatique nécessite de chercher par soi-même et de s'auto-former pendant et après*

ses études. »



Damien Deruy, ingénieur d'études et développement, Worldline, diplômé du Master 2 ILL de l'Université d'Artois.

Damien Deruy, de la même promotion du Master 2 que Deborah Gilbert, fait des études d'informatique par passion. Son oncle ingénieur en informatique lui a transmis le virus. Il a commencé adolescent à développer des sites Internet en html. Entré en 3^e année de licence à

l'université d'Artois après avoir échoué en 3^e année de l'école d'ingénieurs IG2I à Lens, qui dépend de Centrale Paris, il apprécie la faible taille de la promotion (une vingtaine d'étudiants) qui facilite l'entraide entre étudiants et la proximité avec les professeurs. « *Chaque semaine, une entreprise venait se présenter en Master.* » Ses stages effectués à Worldline, groupe Atos, en développement front et middle, il y est embauché en CDI comme ingénieur d'études et développement une fois diplômé en 2014 : « *La fac m'a beaucoup aidé, à relever la tête. C'est une très bonne école.* »

Les écoles d'informatique ou la pédagogie par projet

D'autres bacheliers s'orientent vers une école privée, comme ces diplômés et étudiants d'In'Tech Info. Adeline Martin est en 3^e année à In'Tech Info après un bac ES obtenu en 2012. Elle débute un stage de six mois chez CGI pour développer pour le compte de Sephora Europe (site Internet, appli mobile, appli magasin). Très connectée, elle a choisi la filière Ingénierie du Logiciel (IL). Elle met en garde contre l'image du geek qui rebute les jeunes femmes : « *Nous avons des cours de gestion et de communication. Les entreprises demandent des compétences transverses, de savoir*

communiquer sur notre travail. Après quelques années dans le développement, je pense devenir chef de projet, et ne pas rester en SSII. » Adeline



Adeline Martin, étudiante en 3^e année à In'Tech Info.

Martin a trouvé son stage via le forum d'entreprises de l'école. « Nous sommes vingt en 3^e année filière IL, dont trois filles. C'est une petite famille, avec une ambiance bon enfant. Dans un milieu masculin, il faut avoir du répondant face aux vannes, et s'affirmer. J'ai

été chef de projet sur la plupart des projets de groupe à l'école. » Côté projet de développement, sa meilleure expérience est le développement du programme « Your Meals » pour le site Saveurs d'enfants qui propose des menus pour les enfants. A partir du scan des code-barres des courses mises au frigo, de la date de péremption, des menus sont suggérés à l'internaute. Côté projet, son équipe a donné des cours à des immigrés qui apprennent le français dans une association d'Ivry-sur-Seine, pour apprendre à utiliser un ordinateur. « Nous avons dû adapter notre langage, être à l'écoute. On les a aidés à naviguer sur les sites des impôts, de Pôle Emploi, à rédiger un CV. »

Du développement à la création d'entreprise

Sylvain Paget et Isaac Duplan sont de la promotion 2010 d'In'Tech Info et ont créé



Isaac Duplan et Sylvain Paget, cofondateurs de Fivetouch, diplômés 2010 d'In'Tech Info.



Christine Calais

ensemble leur entreprise, Fivetouch, en août 2011. Ils ont apprécié la méthode de l'école : chacun des cinq premiers semestres débute par un mois de cours effectués pour 20% par des professeurs, 80% par des intervenants travaillant en entreprise qui apportent une vision concrète, puis par cinq mois de projets par petit groupe autonome de trois à cinq personnes, suivi par un tuteur. « Nous avons intégré un énorme socle technologique, notre cœur de métier. Et nous avons appris à apprendre grâce à la méthode projet. Les projets sont proposés par des clients, entreprises ou associations, avec qui nous étions en relation directe. Dotés de bases, nous devons trouver la solution à une problématique, donc nous adapter, comme en entreprise. » Ainsi, ils ont travaillé à la refonte du système de prise de notes du Samu Social pendant les maraudes. Ils ont développé une application cartographique en fonction des règles de calcul des démographes en Guyane pour définir les futurs itinéraires de bus en fonction du développement des agglomérations. Le projet a continué de façon rémunérée dans le cadre de la Junior-Entreprise de l'école. Leur entreprise

Fivetouch a d'abord été une agence de développement mobile pour les grands comptes, comme Gras Savoye ou Nokia. Début 2015, elle lance Evencity, une application mobile « pour les feignants » : une sélection personnalisée d'événements parisiens est proposée aux urbains pressés et aux touristes. Le business model repose sur une commission de 5 à 7% par achat de place de spectacle, d'exposition... Big Data et machine learning, qui permet de deviner les informations manquantes, permettent de proposer des événements pertinents en fonction des goûts repérés du mobinaute. Ils cherchent à lever 0,4 à 1 M€ auprès des business angels pour développer l'application parisienne, et répliquer l'application dans d'autres villes européennes. Ils recommandent In'Tech Info : « les jeunes qui sont autonomes et appréhendent la logique projet s'y épanouissent. »

Pour conclure, la détermination des étudiantes et diplômées nous a paru plus affirmée que celle des étudiants masculins, au moins autant que celle des deux créateurs d'entreprise. Mais elles restent malheureusement une faible minorité (5 à 15%) sur les bancs des écoles d'informatique. Nous sommes persuadés de la nécessité de les attirer plus nombreuses dans la filière du développement, comme Edouard Vyncke qui souligne : « Je conseille aux jeunes femmes de venir, elles trouveront toujours un poste dans les entreprises qui souhaitent des équipes de développement mixtes. »

Des filières pour devenir développeur (non exhaustif)

Filière	Niveau du diplôme	Spécialisations	Possibilité d'alternance	Frais de scolarité annuels (en euros)	Salaire moyen d'embauche du jeune diplômé en milliers d'euros	Site Internet
Ecoles d'ingénieur publiques et privées						
Efrei	bac+5	Ingénierie logicielle, Imagerie et réalité virtuelle	oui	7080 à 8480 selon l'année	38	www.efrei.fr
ESIEA	bac +5	Architecture et Ingénierie du Logiciel (AIL)	oui	6990 à 8160	38,1	www.esiea.fr
ISIMA	bac +5	Ingénieur en Informatique et Modélisation	oui, par contrat de professionnalisation en troisième année	610	30,7	www.isima.fr
Ecoles d'ingénierie informatique						
Epitech	bac+5	Expert en technologie	oui à partir de la 3 ^e année	5990 à 7800 selon l'année	38,3	www.epitech.eu
ESGI	bac+3, bac+5	Architecture des logiciels, Développement Web, Développement de jeux vidéo, Ingénierie des applications mobiles	oui	5700 à 6900 selon l'année	bac+5 35,2	www.esgi.fr
ETNA	bac+5	Architecte logiciel-développeur d'applications	obligatoire	Prep'ETNA: gratuit Bachelor et Master 1: 1600	37,5	www.etna-alternance.net
In'Tech Info	bac+3, bac+5	Ingénierie du Logiciel (IL)	obligatoire en 4 ^e et 5 ^e année	8900	bac + 3 29,4, bac + 5 36,8	www.intechinfo.fr
IT Paris Eiffel	bac+2, bac+3	BTS SIO SLAM, Bachelor Développement Web & mobilité	oui	7400 HT formation initiale, 7700 HT en alternance	27,5	www.it-pariseiffel.fr
Universités						
Université d'Artois	bac+5	Ingénierie Logicielle pour l'Internet	Non, mais en prévision dans les deux ans	474	31	www.cril.univ-artois.fr/master/ii/m2proil-home.html
Université de Lille	bac+3	développement internet et intranet (DA2I)	non	402	27,9	http://da2i.univ-lille1.fr/



Le « Machine Learning »

Quand les données remplacent les algorithmes



Pirmin Lemberger
SQLi

LES ALGORITHMES BIENTÔT OBSOLETES ?

D'ici quelques mois, le centre de cancérologie Memorial Sloan Kettering à New York va recruter un jeune spécialiste en diagnostic médical d'un type un peu particulier : non pas un brillant docteur en génétique, mais une intelligence artificielle (IA) développée par IBM et qui répond au nom de **Watson**. Durant ses « études » Watson a ingurgité 2 millions de pages de journaux spécialisés, et presque autant de rapports cliniques. Cette voracité de connaissances et ses capacités d'analyse hors pair le mettent d'ores et déjà en position de surpasser ses collègues humains, c'est du moins ce que prétendent ses tuteurs. De plus, contrairement à ses collègues humains, Watson possède le don d'ubiquité ; à terme il sera disponible sous forme de service Cloud pour tous les hôpitaux de la planète. Peu exigeant

sur sa rémunération, Watson pourrait contribuer par ailleurs à alléger la note pharamineuse des coûts de la santé dans de nombreux pays. L'exemple précédent, qui fait entrer la médecine dans une ère nouvelle, est une application d'un domaine de recherche aujourd'hui en plein essor : le « **machine learning** » (ML ou « apprentissage automatique »). L'engouement récent pour cette discipline tient en grande partie à une observation qui a surpris les spécialistes en IA eux-mêmes :

« L'utilisation conjointe de quantités massives d'informations et d'algorithmes d'apprentissage relativement simples rend possible la solution à des problèmes considérés il y a peu comme inaccessibles. »

Les capacités de traitements massivement parallèles ont de ce point de vue changé la donne depuis quelques années. Parmi les exemples spectaculaires, citons encore la réalisation par Google d'une voiture en **conduite autonome**.

Hormis ces exemples qui frappent l'imagination, il existe des applications plus

prosaïques du ML dans le contexte du e-business. L'analyse prédictive d'un **panier d'achat** d'un consommateur est sans doute l'usage le plus commun. La **détection de fraude** dans les transactions bancaires ou l'**estimation du risque** de non-remboursement d'un prêt en fonction du passé financier d'un demandeur de crédit sont d'autres exemples. Les technologies de **reconnaissance de forme** au sens large, comme la reconnaissance optique de caractères (**OCR**) ou celle des visages, la reconnaissance du langage naturel ou la traduction automatique relèvent toutes du ML. Dans le champ des études sociales, citons l'**analyse de sentiments** qui cherche à classer des documents selon leur tonalité émotionnelle dominante.

Toutes ces applications ont en commun de n'utiliser aucun algorithme explicite pour parvenir à leur fin. Dans le cadre de la reconnaissance des visages, il n'existe évidemment aucun algorithme répertorié capable de reconnaître M. Untel. Dès lors, plutôt que de chercher à peaufiner des algorithmes sophistiqués, l'approche par le ML fait le pari

qu'il est possible de construire des systèmes autonomes que l'on pourra « éduquer » sur la base d'un corpus d'exemples pertinents. Entendons-nous bien, il ne s'agit nullement de reconstituer les mécanismes cognitifs à l'œuvre dans le cerveau humain - sur ce plan-là le ML ne dit rien - mais plutôt de construire des mécanismes prédictifs.

LES STRATÉGIES DU « MACHINE LEARNING »

La possibilité de prédire un phénomène à partir d'observations passées présuppose l'existence d'un mécanisme causal. Pour prédire un phénomène, deux possibilités se présentent. La première, celle qu'utilise un chercheur, consistera à comprendre ce mécanisme causal en élaborant un **modèle explicatif**. La seconde, utilisée dans le ML et en statistiques, sans être aussi ambitieuse, reste très utile. Elle cherche simplement à découvrir des corrélations significatives dans un jeu d'observations, c'est-à-dire trouver un **modèle prédictif**. Les deux principales stratégies pour y parvenir sont les suivantes.

L'Apprentissage supervisé

Pour fixer les idées, imaginons qu'au sein d'une population nous observons une **corrélation entre le poids et la taille des individus**. Pour « apprendre » cette relation, et être en mesure de faire ultérieurement des prédictions du poids d'un individu, on procède en trois étapes.

- (1) Il nous faut d'abord une **représentation** de nos prédictions. Un exemple d'une telle représentation est l'ensemble des droites du plan défini par l'axe « taille » (horizontal) et l'axe « poids » (vertical). Chaque droite correspond évidemment à une prédiction.
- (2) Parmi ces prédictions, certaines sont meilleures que d'autres, il nous faut donc disposer d'une méthode d'**évaluation**. Dans notre exemple, une évaluation pourrait consister à additionner les « erreurs » commises sur chacun des individus, chacun d'elle étant interprétée comme la distance entre la droite et la valeur observée.
- (3) Il nous faut enfin une stratégie d'**optimisation** qui nous permette de trouver la meilleure prédiction possible. Ce triptyque « représentation, évaluation, optimisation » caractérise en réalité toutes les démarches d'**apprentissages supervisés (AS)** dans lesquelles on cherche à **généraliser** des associations connues, en l'occurrence l'ensemble des couples (taille, poids), à des cas non encore observés. Les valeurs observées de l'effet (le poids) constituent ce qu'on appelle le **feedback** de l'AS.



L'exemple précédent, dans lequel on cherche une relation optimale entre plusieurs variables numériques, est ce que l'on appelle en statistique un problème de **régression**. Son principal mérite pour nous est de servir d'illustration de quelques concepts du ML dans un contexte élémentaire. Il n'est cependant pas représentatif de la puissance des techniques de ML dans la mesure où le processus d'optimisation s'avère pour le coup trivial puisqu'il se réduit au calcul d'une **formule** qui donne explicitement la prédiction optimale en fonction des valeurs observées.

Un exemple plus significatif de ML consiste par exemple à chercher à savoir si un emprunteur potentiel, dont on connaît les revenus, l'âge et le taux d'épargne, fait partie ou non des clients à risque. On cherche donc à prédire l'appartenance d'un individu à une catégorie plutôt qu'une caractéristique numérique, mais le principe reste le même. Nulle formule explicite cette fois-ci, le ML prend alors tout son sens au prix sans doute d'une représentation sous forme d'un jeu de règles à optimiser (a).

L'un des problèmes fondamentaux d'un apprentissage supervisé est d'éviter l'effet dit d'**overfitting** (ou de **surapprentissage**) qui se manifeste lorsqu'à force d'entraînement, un jeu de règles se révèle parfaitement adapté aux données déjà observées mais ne conduit cependant à aucune généralisation utile. Des parades ont été élaborées par les **data scientists** pour se prémunir contre cet écueil. Pour partie, elles consistent à répartir astucieusement les données expérimentales en **données d'apprentissage** d'une part, et en **données de validation** d'autre part.

L'Apprentissage Non-supervisé

Imaginons maintenant que, dans le cadre d'une application CRM, l'on cherche à répartir un ensemble de prospects en niches dans un marché dont nous ignorons complètement la

segmentation. Il s'agit cette fois de découvrir des structures cachées dans les données plutôt que de généraliser des exemples à partir d'un feedback connu. C'est précisément l'objectif des techniques d'**apprentissage non supervisé (ANS)** du ML proches de celles du **data mining**. Contrairement à l'AS qui cherche à optimiser les paramètres d'un modèle, restreignant ainsi la recherche d'une solution optimale, les techniques de l'ANS ne présupposent rien, si ce n'est que des observations similaires ont des significations voisines. Dans les situations où aucun modèle n'est disponible, a priori l'ANS sera donc une approche à privilégier.

APPRENDRE LE « MACHINE LEARNING »

L'art de tirer le meilleur parti de données recueillies sur un échantillon d'observations est ce qu'on appelle l'**inférence statistique**. C'est le fondement théorique d'une grande partie du ML. Quiconque souhaite aborder le ML ne pourra en conséquence faire l'économie de se rafraîchir la mémoire sur ce qu'est une probabilité conditionnelle, une loi normale ou le degré de signification d'un test. Que les allergiques aux mathématiques (et même les traumatisés) toutefois se rassurent, car le ML est une discipline où l'intuition, l'astuce et l'expérience sont ici reines, bien plus que les beautés intemporelles des théorèmes. Le cœur du ML reste l'élaboration et l'optimisation d'algorithmes spécialisés propres à certaines tâches d'apprentissage.

Dans un contexte Big Data la question de savoir si un algorithme est aisément **parallélisable**, au moyen d'un schéma **MapReduce** par exemple, revêt une grande importance pratique. Un data scientist devra en tenir compte pour évaluer la pertinence d'un algorithme de ML. Quelques conseils enfin pour ceux qui souhaiteraient approfondir le sujet. Ce **MOOC** (<http://goo.gl/80JoFz>) de l'université de Washington propose un survol du ML en une vingtaine de vidéos dont le principal mérite est d'offrir un bon panorama du ML. Pour aller plus loin, on pourra consulter **cet ouvrage** (<http://goo.gl/tt0Jfv>) qui aborde en particulier les techniques récentes du ML. Enfin, pour les plus motivés, une approche pratique au ML reste la plus efficace. On pourra tester par exemple l'outil **Data Science Studio** de la société Dataiku qui, propose une large palette d'outils dédiés à l'apprentissage supervisé, au data mining et à l'analyse prédictive.

(a) Parmi les représentations complexes du ML citons les collections de règles, les arbres de décision ou encore les réseaux de neurones.



Azure Machine Learning

Une des valeurs importantes des données vient de la prédiction de l'avenir proche qu'on peut faire à partir de données passées. Les données peuvent aussi servir pour de la programmation par l'exemple ; c'est ainsi qu'après avoir exposé la machine à des milliers d'images avec ou sans chat, elle pourra ensuite nous dire si une image contient un chat, ou pas.



Benjamin Guinebertière
J'accompagne techniquement des startups et autres entreprises de tailles différentes qui vont vers le Cloud Microsoft Azure, avec ou sans Big Data ou Machine Learning. Egalement, je parle (conférences), écris (blogs, ...), et prends les feedbacks des utilisateurs. Vous pouvez me retrouver sur <http://3-4.fr>.

Tout cela est du domaine du machine learning. Les domaines d'application sont nombreux et augmentent. En voici quelques-uns. Dans le domaine du retail ou des ressources humaines, le « churn analysis » permet de prédire quand un client ou un employé va partir. Dans le domaine du numérique, la prédiction de l'intérêt pour un internaute pour telle publicité plutôt que telle autre, permet de viser juste. Il y a bien sûr aussi tout ce qui consiste à recommander à des utilisateurs des produits, en fonction de ce qu'ils ont choisi précédemment, et de ce que l'ensemble des utilisateurs choisit. On peut aussi prévoir des pannes dans le domaine de l'industrie et faire de la maintenance prédictive (voir exemple à <http://3-4.fr/ThyssenKruppML>). Pour citer un dernier exemple, on peut aussi prédire si un e-mail est du spam ou non; ce genre de filtre existe depuis des années. La discipline n'est donc pas nouvelle, mais elle se démocratise aujourd'hui car le traitement de beaucoup de données nécessaires à un apprentissage efficace est devenu beaucoup plus simple et abordable.

Mais nous sommes dans un magazine sur le développement, donc je vous propose de nous intéresser à cet aspect des choses. Il existe des bibliothèques et outils Open Source qui implémentent des algorithmes classiques de machine learning.

Comme cela fait appel à beaucoup de notions de statistiques, R est un de ces outils. R est disponible à <http://www.r-project.org/>.

Un des langages de développement importants dans le traitement de la donnée est Python car il est facilement lisible, et peut être utilisé aussi par des scientifiques dont le métier n'est pas l'informatique. Un certain nombre de bibliothèques de traitement de matrices ou plus généralement à destination des scientifiques sont disponibles dans ce langage comme NumPy et SciPy ou encore matplotlib pour la visualisation de données (dataviz). Au-dessus de ces bibliothèques a été construite la bibliothèque de machine learning SciKit-Learn disponible à <http://scikit-learn.org>.

Enfin, pour citer une dernière bibliothèque populaire, ML Lib est disponible pour tourner dans l'environnement distribué Spark. Elle est disponible à <http://spark.apache.org/mllib/>.

Machine Learning sur Azure

La première chose à laquelle on pense en termes de machine learning sur Azure, est naturellement Azure Machine Learning, et nous allons y revenir plus en détails dans la suite de cet article. Mais je voulais aussi évoquer des alternatives.

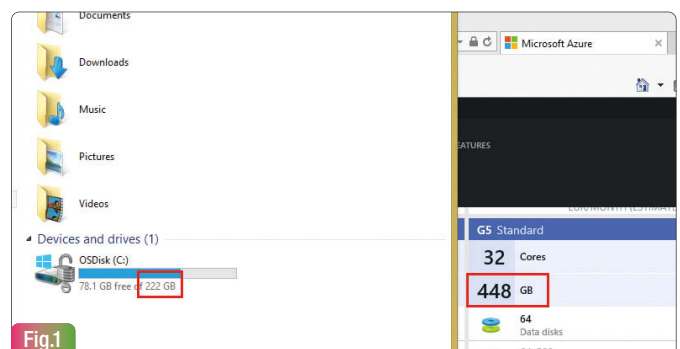
Pour ML Lib, un moyen simple pour avoir Spark sur Azure est de s'appuyer sur HDInsight et son exemple de script de personnalisation expliqué sur <http://azure.microsoft.com/en-us/documentation/articles/hdinsight-hadoop-spark-install/>.

Pour SciKit-Learn, les notebooks IPython sont une interface de développement couramment utilisée, et cela peut se mettre en œuvre dans une VM comme expliqué dans cet article d'azure.com : <http://azure.microsoft.com/en-us/documentation/articles/virtual-machines-python-ipython-notebook/>. Dans ce scénario et d'autres en ML, disposer de beaucoup de RAM peut être fondamental et Azure propose des VM jusqu'à 448 Go de RAM (2 fois la taille du disque dur de l'ordinateur avec lequel j'écris cet article !) **Fig.1**.

Azure Machine Learning

Mais revenons au service Azure Machine Learning qui propose à la fois un environnement de développement / expérimentation, de test et de production pour le machine learning. Je vous propose pour cela de rentrer dans le détail d'une démo que j'ai eu l'occasion de montrer en session plénière le mardi 10 février dernier aux TechDays (<http://mstechdays.fr>). On se place dans la peau d'une société de véhicules de tourisme avec chauffeurs (VTC) et on cherche à placer au mieux les véhicules en fonction des demandes anticipées. D'un point de vue opérationnel, un chauffeur demande depuis une application mobile les prévisions autour d'une zone et l'application lui indique où il y aura le plus de demande. Plus concrètement, un GET sur <http://{hostname}/api/prediction?z=75015&y=2015&m=2&d=10&h=18> qui peut se lire « donne-moi les prédictions pour la zone 75015 et autour, le 10 février 2015 à 18h », on obtient une réponse comme

```
{
  "geoZones": [
    { "geoZoneId": "75015", "remainingRides": 2 },
    { "geoZoneId": "92130p", "remainingRides": 8 },
    { "geoZoneId": "75016", "remainingRides": 2 },
    { "geoZoneId": "75116", "remainingRides": 2 },
    { "geoZoneId": "92100", "remainingRides": 2 },
    { "geoZoneId": "92170", "remainingRides": 2 },
    { "geoZoneId": "75014", "remainingRides": 2 },
    { "geoZoneId": "75006", "remainingRides": 2 },
    { "geoZoneId": "75007", "remainingRides": 2 }
  ],
  "electedZone": "92130p"
}
```



Si un deuxième chauffeur demande pour la même zone, il aura la même réponse mais avec remainingRides à 7 au lieu de 8 pour 92130p puisque cette zone est celle où se rend le premier chauffeur, et ainsi de suite jusqu'à avoir des changements de zone élue ; ci-dessous c'est 75016 qui est proposée au lieu de 92130p car beaucoup de chauffeurs se rendent déjà en 92130p :

```
{
  "geoZones": [
    { "geoZoneId": "75015", "remainingRides": 1 },
    { "geoZoneId": "92130p", "remainingRides": 1 },
    { "geoZoneId": "75016", "remainingRides": 2 },
    { "geoZoneId": "75116", "remainingRides": 2 },
    { "geoZoneId": "92100", "remainingRides": 2 },
    { "geoZoneId": "92170", "remainingRides": 2 },
    { "geoZoneId": "75014", "remainingRides": 2 },
    { "geoZoneId": "75006", "remainingRides": 2 },
    { "geoZoneId": "75007", "remainingRides": 2 }
  ],
  "electedZone": "75016"
}
```

Pour faire cela, il nous faut un algorithme qui soit capable de prédire le nombre de courses qui seront faites dans telle ou telle zone, en fonction des paramètres envoyés à l'API (date, heure, zone) et d'informations de contexte que l'API peut elle-même récupérer par ailleurs (météo, indice de circulation dans la zone, indice événementiel représentant le nombre de salons et autres manifestations qui se déroulent à cet endroit). La récupération de ces informations de contexte n'est pas couverte dans cet article. Concentrons-nous plutôt sur la partie Machine Learning de l'algorithme. On part de données passées que sont les courses déjà effectuées. Le jeu de données dont on dispose contient les colonnes suivantes : PickupDateRounded, PickupMonth, PickupHour, PickupWeekday,

DropOffDateRounded, NbRides, PickupGeoZoneId, PickupZipCode, PickupWeather, PickupEventsIndex, PickupTrafficDensity, DropOffGeoZoneId, DropOffZipCode, DropOffWeather, DropOffEventsIndex, DropOffTrafficDensity. NbRides est la colonne qu'on cherche à prédire en fonction des autres. Ces quelques données exemples vous donneront une idée intuitive du jeu de données : Fig.2. Nous allons utiliser un algorithme d'apprentissage supervisé (on dispose de données d'apprentissage qui contiennent les valeurs de la colonne que l'on cherche à prédire). Comme on cherche à prédire une valeur numérique (par opposition à une catégorie), nous allons utiliser un algorithme de régression.

Cet article n'entre pas dans le fonctionnement d'une régression linéaire ou de la façon dont on peut commencer à utiliser Azure Machine Learning. Vous pourrez trouver cela dans le cours en ligne sur Microsoft Virtual Academy, à <http://3-4.fr/mva-ml-101>, que j'ai eu le plaisir de réaliser. Il existe dans Azure Machine Learning plusieurs algorithmes de régression. Au-delà de la connaissance des caractéristiques principales de tel ou tel algorithme par rapport à tel autre, il peut être utile de tester plusieurs façons de faire et de comparer. C'est ici ce qui est illustré avec à gauche une régression linéaire et à droite une régression à base de réseaux neuronaux Fig.3.

On reconnaîtra dans le diagramme les étapes suivantes :

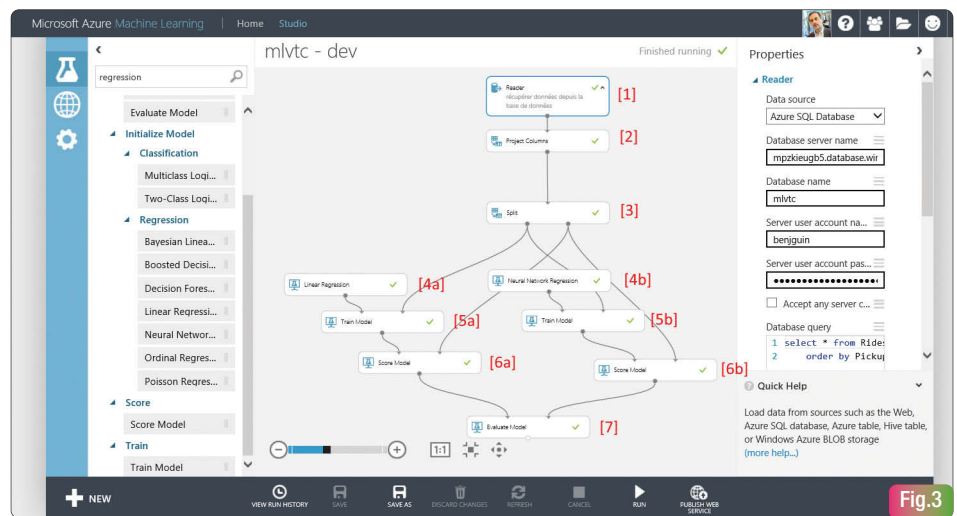


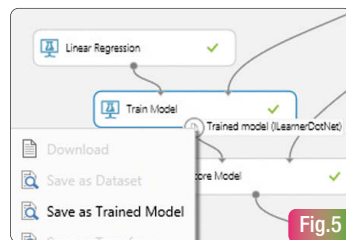
Fig.3

	NbRides	PickupGeoZoneId	PickupZipCode	PickupWeather	PickupEventsIndex	PickupTrafficDensity	DropOffGeoZoneId	DropOffZipCode	DropOffWeather	DropOffEventsIndex	DropOffTrafficDensity
22	4	75008	75008	clear	3	4	75008	75008	clear	3	4
23	8	75017	75017	clear	3	1	XXX	75000	clear	5	3
24	5	75008	75008	clear	3	4	75017	75017	clear	3	1
25	0	75015	75015	clear	0	2	75011	75011	clear	3	5
26	1	75015	75015	clear	0	2	75017	75017	clear	3	1
27	5	75012	75012	clear	2	3	75006	75006	clear	5	2
28	4	75011	75011	clear	3	5	75008	75008	cloudy	5	4
29	10	75006	75006	cloudy	5	3	75012	75012	cloudy	3	0
30	3	75014	75014	cloudy	2	3	75007	75007	cloudy	2	4
31	9	75017	75017	cloudy	4	2	75008	75008	cloudy	5	4

	PickupDateRounded	PickupMonth	PickupHour	PickupWeekday	DropOffDateRounded	NbRides	PickupGeoZoneId	PickupZipCode	PickupWeather	PickupEventsIndex	PickupTrafficDensity	DropOffGeoZoneId	DropOffZipCode	DropOffWeather	DropOffEventsIndex	DropOffTrafficDensity
22	2014-09-01 11:00:00.000	9	11	2	2014-09-01 12:00:00.000	4	75008	75008	clear	3	4	75008	75008	clear	3	4
23	2014-09-01 11:00:00.000	9	11	2	2014-09-01 12:00:00.000	8	75017	75017	clear	3	1	XXX	75000	clear	5	3
24	2014-09-01 12:00:00.000	9	12	2	2014-09-01 12:00:00.000	5	75008	75008	clear	3	4	75017	75017	clear	3	1
25	2014-09-01 12:00:00.000	9	12	2	2014-09-01 12:00:00.000	0	75015	75015	clear	0	2	75011	75011	clear	3	5
26	2014-09-01 12:00:00.000	9	12	2	2014-09-01 12:00:00.000	1	75015	75015	clear	0	2	75017	75017	clear	3	1
27	2014-09-01 13:00:00.000	9	13	2	2014-09-01 13:00:00.000	5	75012	75012	clear	2	3	75006	75006	clear	5	2
28	2014-09-01 13:00:00.000	9	13	2	2014-09-01 14:00:00.000	4	75011	75011	clear	3	5	75008	75008	cloudy	5	4
29	2014-09-01 14:00:00.000	9	14	2	2014-09-01 14:00:00.000	10	75006	75006	cloudy	5	3	75012	75012	cloudy	3	0
30	2014-09-01 14:00:00.000	9	14	2	2014-09-01 14:00:00.000	3	75014	75014	cloudy	2	3	75007	75007	cloudy	2	4
31	2014-09-01 14:00:00.000	9	14	2	2014-09-01 14:00:00.000	9	75017	75017	cloudy	4	2	75008	75008	cloudy	5	4

Fig.2

- [1] La lecture des données depuis une base de données externe.
- [2] La projection des colonnes. On ne sélectionne que les colonnes à partir desquelles on fera la prédiction. Certaines caractéristiques (« features ») nous semblent a priori participer à la valeur à prédire, d'autres pas. Par ailleurs on sera capable de fournir en entrée de l'algorithme certaines caractéristiques, d'autres pas. Par exemple ici, d'une part la date exacte n'est pas intéressante, contrairement au jour de la semaine et à l'heure ; d'autre part on cherche à connaître le nombre de courses au départ d'une zone, et pas le nombre de courses allant d'un endroit à un autre. On supprime donc toutes les colonnes DropOff*. Cette étape [2] consiste donc à rayer certaines colonnes de notre jeu de données : ~~PickupDateRounded~~, ~~PickupMonth~~, ~~PickupHour~~, ~~PickupWeekday~~, ~~DropOffDateRounded~~, ~~NbRides~~, ~~PickupGeoZoneId~~, ~~PickupZipCode~~, ~~PickupWeather~~, ~~PickupEventsIndex~~, ~~PickupTrafficDensity~~, ~~DropOffGeoZoneId~~, ~~DropOffZipCode~~, ~~DropOffWeather~~, ~~DropOffEventsIndex~~, ~~DropOffTrafficDensity~~. Cette phase, appelée « feature engineering » est très importante ; elle peut conditionner la pertinence d'un modèle. Elle inclut la définition de nouvelles caractéristiques ; ici par exemple, même si cela a été fait en amont en base de données, la colonne PickupWeekday est déduite de PickupDateRounded. On pourrait aussi imaginer une colonne Jour/Nuit. Ce sont des caractéristiques qu'on est capable de déduire en amont mais que l'algorithme de machine learning ne trouvera pas seul.
- [3] De façon à savoir si le modèle défini est pertinent, on met de côté une partie (30%) des données historiques qui nous permettront de tester le modèle (étape [6]). A gauche, les 70% des données serviront à l'apprentissage (étape [5]).
- [4a] On choisit ensuite l'algorithme à utiliser. Parmi les algorithmes disponibles dans la barre d'outils sur la gauche, on en choisit ici deux [4a] et [4b]. Cette étape [4] consiste aussi à fixer les paramètres propres à l'algorithme. Voici pour illustration ceux en [4b] : Fig.4.
- [5a] Cet algorithme doit apprendre à partir des données d'entraînement qu'on lui fournit. On a donc en entrée de cette étape les données d'apprentissage, l'algorithme non entraîné, en paramètres la colonne « label » que l'on cherche à prédire (ici : NbRides). En sortie, on a un algorithme entraîné Fig.5.
- [6a] « Score Model » porte ce nom parce que c'est là qu'on sait si le modèle est efficace ou pas. Par exemple, en visualisant les résultats sur quelques



Neural Network Regression

Hidden layer specification
Fully-connected case

Number of hidden nodes
10

The initial learning weight
0.1

The learning rate
0.005

The momentum
0

The type of normalizer
Do not normalize

Number of learning iterations
10

☒ Shuffle examples

Random number seed

☒ Allow unknown categories

Fig.4

- exemples, on voit en [6a] que cela n'est pas mal : Fig.6.
- [7] pas mal, mais encore ? Pour une régression, le coefficient de détermination (http://en.wikipedia.org/wiki/Coefficient_of_determination) permet de mesurer cette pertinence. 1 est la valeur idéale. Ici, l'étape [7] prend en entrée les résultats de [6a] et [6b] ce qui va aussi nous permettre de comparer la pertinence des deux modèles essayés : Fig.7.

Ici, on choisit donc la branche de gauche [Xa] puisque 0,95 est plus proche de 1 que 0,93.

Maintenant qu'on a un modèle qui fonctionne, il est simple (et c'est un des grands avantages d'Azure Machine Learning) de l'exposer sous forme d'API REST.

Il suffit de sauvegarder le modèle entraîné comme montré plus haut ([5x]), copier le modèle (Save As), Fig.8 remplacer l'entraînement du modèle par l'appel au modèle entraîné [10], sélectionner les éléments intéressants, définir les paramètres d'entrée [11] et de sortie [12], re-tester [13] et publier en tant que Web Service [14].

On dispose alors d'une documentation de ce Web service [20], Fig.9 Web service qu'on peut tester [23], appeler unitairement [21], ou en batch [22] (pour un grand nombre de lignes pour lesquelles on veut prédire). Le « endpoint » ainsi créé permet de faire quelques appels en développement/test sur le service. Les pages d'aide [21] et [22] donnent des exemples de code en C#, R et Python pour appeler le service, ce qui évite de lire la documentation pour écrire l'appel REST. Enfin, on peut créer un endpoint de production, scalable, en allant sur le portail Azure [24]. Cela permet des appels en parallèle au même modèle.

Pour revenir à notre société de VTC, on a une API qui implémente des

mlvtc - dev > Score Model > Scored dataset

rows
1228

columns
8






	NbRides	PickupGeoZoneId	PickupWeather	PickupEventsIndex	PickupTrafficDensity	Scored Labels
						
10	XXX	sleet	4	2	9.564658	
5	75008	wind	2	1	5.489035	
0	75017	wind	1	5	-0.175508	
4	75013	wind	2	3	3.486945	
3	XXX	fog	2	3	3.639493	
2	75007	fog	2	5	1.669728	
3	75003	clear	1	1	3.614482	
7	XXX	sleet	4	4	7.72098	
8	75010	sleet	2	1	6.436156	
3	75116	wind	2	3	3.562373	
4	92220	fog	2	3	2.9544	
1	XXX	fog	0	2	0.833049	
7	75006	fog	3	1	7.431064	

Fig.6

mlvtc - dev > Evaluate Model > Evaluation results

rows	columns
2	5

	Mean Absolute Error	Root Mean Squared Error	Relative Absolute Error	Relative Squared Error	Coefficient of Determination
view as					
[Xa]	0.61268	0.735306	0.217828	0.048286	0.951714
[Xb]	0.675095	0.858216	0.240019	0.065777	0.934223

Fig.7

règles métier (sauvegarder en base et décrémenter les prédictions données par le modèle, choisir la bonne zone, ...). Voici le code qui fait l'appel en parallèle (sur les différentes zones) au modèle :

```
public class PredictionController : ApiController
{
    public async Task<object> Get(string z, int y, int m, int d, int h)
    {
        return await MlvtcBusinessRules.VehiclePositionning.GetPredictionsAroundGeoZoneAsync(
            z, new DateTime(y, m, d, h, 0, 0));
    }
}

namespace MlvtcBusinessRules
{
    public static class VehiclePositionning
    {
        public static async Task<object> GetPredictionsAroundGeoZoneAsync(string geoZone,
            DateTime pickupDateTimeRounded)
        {
            (...)

            // lancer en parallèle sur les zones géographiques voisines
            foreach(string z in geoZones)
            {
                tasks[z] = GetPredictionsForOneZoneAsync(z, pickupDateTimeRounded);
            }

            // récupérer les résultats pour consolidation
            foreach (string z in geoZones)
            {
                (...)

                await Helpers.DecrementRidesPerHourPredictAsync(electedZone, pickupDateTimeRounded, maxRides);

                return result;
            }

            public static async Task<object> GetPredictionsForOneZoneAsync(string geoZone, Date
```

```
Time pickupDateTimeRounded)
{
    (...)

    var remains = await Helpers.GetRidesPerHourPredictFromSqlAsync(geoZone, pickupDateTimeRounded);
    if (remains == null)
    {
        // no prediction was cached in database. Call Azure ML

        // get additional data from context
        int eventsIndex = Helpers.GetEventsIndex(geoZone, pickupDateTimeRounded);
        int trafficDensity = Helpers.GetTrafficDensity(geoZone, pickupDateTimeRounded);
        string weather = Helpers.GetWeather(geoZone, pickupDateTimeRounded);
        int weekDay = 1 + (int)pickupDateTimeRounded.DayOfWeek;
        int hour = pickupDateTimeRounded.Hour;

        // call Azure ML Web API
        string predictionResult = await MachineLearningPrediction.InvokeRequest
            ResponseService(
                geoZone, eventsIndex, trafficDensity, weather, weekDay, hour);

        (...)

        return result;
    }
}
```

Le futur d'Azure ML

Nous avons vu quelques aspects d'Azure Machine Learning; il est également possible d'écrire du code R dans une expérimentation et d'exposer cela. Cela risque de prendre de l'ampleur avec le rachat annoncé de Revolution Analytics par Microsoft (<http://3-4.fr/Microsoft-to-acquire-RevolutionAnalytics>), sans pour autant remettre en cause l'intégration de Python telle qu'expliquée dans ce billet sur la feuille de route du service : <http://3-4.fr/azureml-python-blogpost>.



La suite du dossier "Machine Learning" dans Programmez! 184

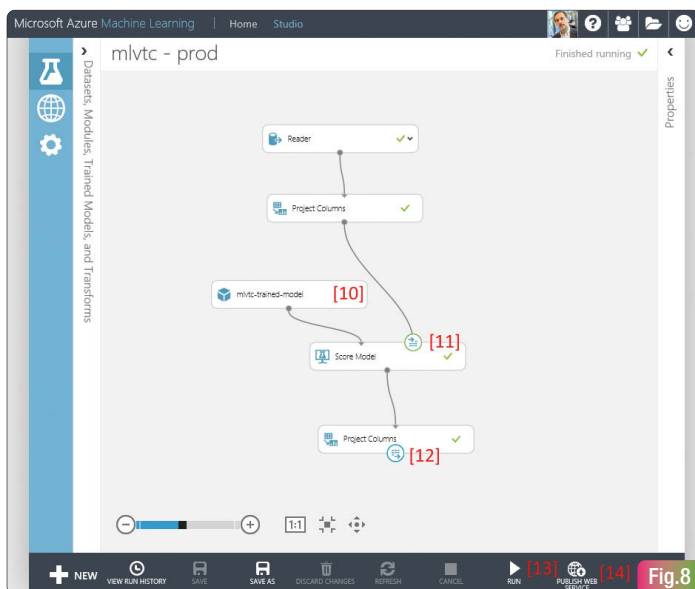


Fig.8

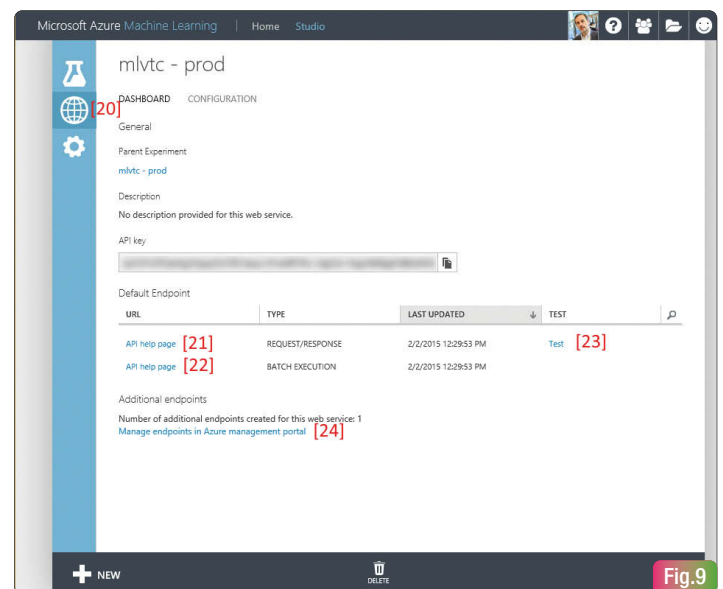


Fig.9

Développeurs : les défis de l'accessibilité

La France rajeunit son référentiel d'accessibilité et l'enrichit d'un label



La DSI de l'État a remis à jour le RGAA, le référentiel d'accessibilité numérique pour les sites publics. Mieux : elle l'a assorti d'une série de mesures d'accompagnement qui manquaient cruellement à la version précédente. La plus emblématique concerne un label qui traduit le niveau de conformité au RGAA mais qui valorise également une démarche organisationnelle déployée autour de l'accessibilité numérique.



Vincent
Berdot



Philippe
Bron

respectivement chargé de la communication et architecte à la Direction interministérielle des systèmes d'information et de communication (Disic)

« L'accessibilité c'est aussi le numérique. Et le numérique ne doit pas être une barrière de plus, un interdit de plus, mais un levier de plus, une chance de plus. C'est pourquoi la mise en accessibilité des sites de l'État est le premier impératif ». En clôture de la Conférence Nationale sur le Handicap (CNH) de décembre 2014, la tonalité du Président de la République résume bien la situation. Elle résonne comme un aveu tout en laissant entrevoir l'espoir d'une amélioration. L'aveu ? C'est celui du manque criant d'accessibilité des sites de l'Administration. Le chiffre commence à être connu : selon une étude Brailletnet(*), seuls 4% d'entre eux auraient affiché une déclaration de conformité. À savoir un document réglementaire autoproduit par l'administration, censé refléter son niveau d'accessibilité numérique. L'espoir ? À l'image des tables rondes préparatoires à la CNH, cette absence de considération des personnes handicapées (qui se traduit d'ailleurs par un manquement à la loi, nous y reviendrons), fait l'objet d'une prise de conscience. Mieux, celle-ci se double d'un plan d'actions ambitieux pour changer la donne : l'État, par le biais de la DSI de l'État (la Direction interministérielle des systèmes d'information et de communication), vient en effet de remettre à jour le RGAA (Référentiel Général d'Accessibilité pour les Administrations), le référentiel sur lequel sont tenues de s'appuyer les administrations françaises pour évaluer la conformité de leur site aux règles internationales. Sa nouvelle version, la v3, s'adapte enfin aux nouvelles technologies du Web (HTML5 / Aria). Mais tout aussi important que la mise à jour, cette

mouture est désormais assortie d'un volet accompagnement. Précisément celui qui a tant manqué à la version précédente du RGAA. Avant de zoomer sur les mesures qui devraient enfin mettre les administrations sur les rails de l'accessibilité numérique, revenons sur les raisons de la désaffection subie par le RGAA.

Les raisons d'un échec

Quelques rappels historiques : le RGAA est la transposition d'une directive européenne imposant aux états membres de respecter les WCAG (Web Content Accessibility Guidelines) pour les sites de la fonction publique. Sa création en 2009, est assortie d'un décret (n°2009-546) le rendant obligatoire. En détail, l'administration est tenue de respecter les critères de niveau A et AA du RGAA (le référentiel compte 3 familles de critères : A, AA et AAA). L'État avait ainsi 2 ans (2011) pour mettre son Web à la norme. L'administration territoriale disposait d'une année supplémentaire (jusqu'en 2012). Le décret prévoyait de publier sur une liste noire les sites « hors-la-loi ». Problème : il est vite apparu que le RGAA était peu opérationnel, voire très complexe à mettre en application à l'image de ses tests de conformité. Mais la principale limite était ailleurs : aucun accompagnement sur le terrain n'avait été prévu pour aider à sa mise en œuvre. Aucune formation non plus. Fait symptomatique,

la structure au sein du Ministère en charge des personnes handicapées censée mettre en place la « liste noire » n'a jamais été déployée. Sans compter qu'avec le temps, est venue se rajouter une composante d'obsolescence. Le RGAA n'avait pas été mis à jour depuis 2009, contrairement aux WCAG qui ont évolué avec les techniques et les usages numériques. Le Web de 2009 n'est pas celui de 2014. Le référentiel ne prenait pas en compte HTML 5 et interdisait le recours à JavaScript. C'était là un obstacle à la production de contenus Web modernes et dynamiques. Devant ces difficultés, le RGAA a donc été freiné dans son envol. La place qu'il devait prendre a rapidement été occupée par un autre référentiel d'accessibilité numérique basé lui aussi sur les WCAG mais plus facile à prendre en main : AccessiWeb. Édité sous l'impulsion de l'association BrailleNet, il est aujourd'hui principalement utilisé dans le monde privé même s'il touche une minorité de sites publics (dans ce cas, AccessiWeb sert à évaluer la conformité RGAA avec lequel il est compatible).

Une révision du RGAA en mode collaboratif

Le constat de cet échec par les autorités publiques ne date pas d'aujourd'hui. Il remonte à au moins 3 ans. À l'issue de la CNH (Commission Nationale du Handicap) de juin 2011, les pouvoirs publics ont en effet souhaité confier au FIPHP (le Fonds pour l'insertion des personnes handicapées dans la fonction publique), en lien avec la DISIC et le Service d'information du gouvernement, le soin de mettre en œuvre un plan d'accessibilité numérique. Ce plan, qui concernait en particulier les agents mal voyants ou non-voyants, avait comme vocation de rendre accessibles les sites Internet et intranet du gouvernement. Il visait également à promouvoir l'usage de logiciels accessibles au sein des administrations pour les agents. Pour la période 2011-2013, la DISIC a ainsi hérité d'une enveloppe de 4,5M€ sur une durée de trois ans. Son mandat : adapter le RGAA, favoriser les formations et réaliser une trentaine d'audits

UN DÉCOUPAGE DU RGAA QUI FACILITE SA MISE À JOUR CONTINUE

La nouvelle présentation du RGAA (entièrement Web) favorise son alignement continu sur les futures évolutions technologiques. En effet, il est désormais découpé autour de critères techniques (conditions à respecter pour garantir l'accessibilité numérique) qui pourront s'analyser unitairement. Sur le site, ces critères feront l'objet de commentaires et d'échanges des internautes, visant à les faire évoluer en fonction des WCAG ou des technologies.

d'éligibilité et de labélisation pour des employeurs des trois fonctions publiques. Dans ce contexte, au terme d'un marché, la DISIC a retenu un groupement (emmené par la société Smile) qui compte parmi les meilleurs experts en accessibilité numérique. Les travaux du programme Accessibilité Numérique de la DISIC ont débuté en avril 2014. Ils se sont articulés autour de 2 axes : d'une part la mise à jour technique du RGAA, d'autre part le déploiement de mesures d'accompagnement, destinées à assurer sa pérennité et à susciter la mobilisation des administrations. Bilan des travaux ? Si la nouvelle mouture du RGAA doit encore être examinée par la Commission Nationale d'Évaluation des Normes et le Conseil National des Personnes Handicapées, avant d'être officiellement reconnue, c'est à dire mentionnée par un arrêté, elle est depuis décembre 2014 techniquement finalisée. Cette révision résulte d'un processus collaboratif de plusieurs mois : entre juillet et septembre 2014, un groupe de travail centré sur les experts d'Accessiweb (une centaine de personnes) a donné le jour à une première proposition de référentiel. Pour cela, il s'est en grande partie inspiré de la dernière version d'Accessiweb, celle tenant compte de HTML5 / Aria. Dans un deuxième temps, courant octobre, ce premier livrable a été ouvert au public par le biais d'un appel à commentaires. Plus d'une centaine de contributions ont ainsi été déposées sur le site hébergeant le RGAA ainsi que les autres grands référentiels informatiques de l'État (<http://references.modernisation.gouv.fr>). Enfin, de septembre à décembre, le groupement a pris en compte et a intégré une partie des remarques et des réflexions du public.

Un label pour valoriser les démarches d'accessibilité

Le nouveau référentiel est donc fonctionnellement prêt. Seulement on l'a vu, aussi complet et clair soit-il, jamais les administrations ne s'en empareront sans y avoir été incitées. Or quelle autre mesure d'incitation que la valorisation pour susciter ce mouvement ? C'est le parti retenu par la DISIC : plutôt que de poursuivre une démarche répressive, incarnée par la liste noire prévue dans le décret, la DISIC a inversé la tendance en créant un « label » récompensant les employeurs publics s'engagent vers l'accessibilité. Baptisé « e accessible » il présente deux avantages notoires. D'une part il est un instrument de communication mettant en lumière une démarche aussi technologiquement avancée que vertueuse. D'autre part, il a beau être adossé au RGAA, il n'exige pas des sites qu'ils respectent le niveau légal, par ailleurs toujours aussi complexe à respecter (ensemble des critères A et AA). « e accessible » valorise tout autant une

LABEL : EXIGENCES ATTENDUES POUR CHACUN DES NIVEAUX

1er niveau : 50 critères précis de niveau simple A, attribuable une seule fois, non renouvelable.

2ème niveau : tous les critères applicables de niveau simple A

3ème niveau : tous les critères applicables de niveau simple A + 50 % des critères applicables de niveau double A (AA)

4ème niveau : tous les critères applicables de niveau simple A et double A (AA). A noter que ce niveau correspond au niveau légal attendu.

5ème niveau : tous les critères applicables du niveau simple A et double A (AA) + des critères du niveau triple A (AAA)

L'exigence du premier niveau été définie pour que les sites aujourd'hui conformes à AccessiWeb ou à l'actuel RGAAv2, puissent au moins être labélisés. Faute de quoi, découragés, ils risqueraient de ne pas s'engager dans ce processus de labélisation (alors qu'ils ont déjà entrepris des démarches avancées en matière d'accessibilité).

démarche et une progression vers l'accessibilité qu'une mise en conformité légale. Explications.

Un label à deux volets

Comme s'obtient le label ? Il impose comme prérequis la mise en place d'une organisation spécifiquement adaptée aux enjeux de l'accessibilité numérique. Il exige ainsi de :


- Responsabiliser la direction de l'administration concernée au travers d'une lettre d'engagement,
- Développer un système qualité interne tourné vers l'accessibilité numérique, avec notamment la désignation d'un référent accessibilité,
- Surveiller et améliorer en continu l'accessibilité des services en ligne. Et ce par deux biais. D'une part un audit interne réalisé chaque année, d'autre part la mise en place d'un « canal accessibilité » permettant aux utilisateurs d'un site de signaler la non conformité de certains de ses éléments.

Ce volet organisationnel ayant été validé lors d'une inspection réalisée par un organisme indépendant, le label exige ensuite du site qu'il affiche un minimum de conformité vis à vis du RGAA. En l'occurrence qu'il respecte le premier niveau du label qui en compte 5 en tout. (voir encadré).

Les conditions du succès

Parallèlement au label, la DISIC envisage un volet formation, parent pauvre de l'accessibilité numérique. Aujourd'hui celle-ci n'est pas prise en compte dans les formations (initiales ou continues) dispensées par l'Administration à ses agents. Elle est d'ailleurs absente dans la plupart des cursus généraux. Or de cette formation dépend en grande partie sa prise en compte au sein des DSI. Dans ce contexte, le groupement fournira des supports de formation clefs en main à destination des administrations. La DISIC ne s'interdira pas à terme de nouer des contacts avec le monde de la formation en dehors de la seule sphère publique.

Enfin, elle mettra en accès des ressources

spécifiques pour les développeurs et les chefs de projets. Les premiers pourront s'appuyer sur une liste de composants nativement accessibles. Des composants tels qu'une barre de navigation, un slider ou un carrousel, issus par exemple de librairies Javascript comme JQuery ou Bootstrap. Et pour ceux qui ne seraient pas accessibles, la DISIC (par le biais du groupement) proposera des correctifs ainsi qu'une documentation permettant aux codeurs d'effectuer eux-mêmes leurs modifications. Quant aux seconds, les chefs de projet, le groupement leur fournira des modèles prêts à l'emploi pour produire par exemple la page d'aide à la navigation (imposée par le RGAA), la déclaration de conformité du site ou encore le cahier des charges de ce dernier. Ils disposeront enfin de guides pour auditer leur site, et de références vers des contenus pédagogiques extérieurs (vidéo, tutoriaux, etc.). Déploiement d'un label, aide à la formation, guides, ressources techniques et pédagogiques, mais également co-financement pour la mise en accessibilité des sites (via le FIPHFP)... Tous ces éléments viendront donc accompagner la révision du RGAA et faciliter sa mise en application sur le terrain. Mais ces mesures ne sont pas les seuls éléments différenciants par rapport à 2009, et la v1 du RGAA. La gouvernance du programme accessibilité numérique a elle aussi changé. Disons, qu'elle existe désormais... Elle a donc été confiée à la DISIC, chargée de coordonner les travaux des administrations centrales sur l'accessibilité numérique. Cette direction interministérielle pilote déjà le système d'information (SI) de l'État lui-même placé sous la responsabilité du Premier Ministre depuis le décret du 1er août. Ce nouveau cadre réglementaire renforce ses prérogatives. Il lui confère un poids supplémentaire pour affirmer auprès des administrations la nécessité, voire l'obligation, de s'aligner sur les standards de l'accessibilité numérique. 

(*) Étude de BrailleNet et de l'Université UPMC en mai 2014, au cours de laquelle 600 services de communication en ligne de l'État, ministères, conseils régionaux, conseils généraux, préfectures, communautés d'agglomérations, mairies ont été examinés.

L'impact de l'accessibilité numérique pour les utilisateurs

Il y a 20 ans environ, le Web faisait son apparition. Même si au début il était surtout destiné aux chercheurs, universitaires ou informaticiens, il a depuis investi nos vies et on peut de moins en moins s'en passer. Utilisateurs de tous âges, toutes professions, de toutes origines et de toutes cultures se servent aujourd'hui du Web et de l'information numérique qu'on y trouve. Grâce au courrier électronique, une personne qui ne pouvait pas écrire sur un morceau de papier peut communiquer avec n'importe qui sans que son correspondant sache qu'il ne voit pas, n'entend pas ou qu'il ne peut pas se servir de ses mains.



Sylvie Duchateau, Consultante en accessibilité numérique chez Access42 et vice présidente de l'association nationale des maîtres de chiens-guides d'aveugles (ANMCGA)

@sylduch - access42.net - anmcga.fr

Chaque collectivité, journal, magasin, a aujourd'hui son site et y affiche les contenus le concernant. On peut maintenant, pour économiser du papier, faire des demandes de factures dématérialisées, remplir des formulaires en ligne ou lire de l'information sur le net. Si l'information numérique est aujourd'hui variée, ceux qui l'utilisent le sont tout autant et n'ont pas tous la même façon de naviguer.

La diversité des modes d'accès à l'informatique

Beaucoup d'internautes se servent d'un ordinateur avec un écran assez grand, mais certains consultent l'information en ne se servant que du clavier, de systèmes de

pointage spécifiques, ne peuvent pas afficher de trop grosses quantités d'informations parce que leurs outils ne permettent pas d'afficher beaucoup de contenu à la fois. D'autres ont besoin de grossir l'affichage des caractères, de modifier la police ou les couleurs, ne perçoivent pas les images, sont sensibles aux contenus qui bougent, n'ont pas les outils ou modules nécessaires pour lire certaines informations telles que des contenus multimédia. Ces diversités sont liées aux outils utilisés par les internautes : différents systèmes d'exploitation, divers navigateurs, mais aussi à leurs possibilités de consultation qui peuvent être différentes car ces utilisateurs ont un handicap ou font partie de la population la plus âgée.

Afin de permettre à tous d'accéder aux divers contenus numériques, des recommandations ont été publiées par le Consortium International W3C (World Wide Web Consortium) dès 1999 et mises à jour en 2008. Ces recommandations ont été adoptées par un grand nombre de pays et sont mentionnées par la Convention des Nations Unies sur les droits des personnes handicapées. Elles sont devenues une norme ISO en octobre 2012 et leur respect est ancré dans des dispositions législatives tout autour de la planète.

Mais pourquoi le respect de ces recommandations est-il si important pour les utilisateurs ?

Prenons un cas concret : il y a quelques temps, je souhaitais effectuer un virement. Or, je suis aveugle. Ne pouvant pas remplir un bordereau de virement classique, j'ai voulu me servir de ma banque en ligne. Je me suis connectée au service, j'ai trouvé le lien me permettant de passer un nouveau virement et d'ajouter un compte. J'ai pu remplir tous les champs obligatoires mais au moment de valider, je ne pouvais pas trouver le bouton de confirmation qui était une image. De plus, ce même bouton ne pouvait être activé qu'à l'aide d'une souris. Or je me sers exclusivement du clavier pour interagir avec l'ordinateur. J'ai tenté plusieurs fois l'expérience, mais sans succès. J'ai donc perdu beaucoup de temps et n'ai pas pu finaliser mon virement, d'où une grande frustration.

Il y a un ou deux ans, je devais former des personnes à l'accessibilité. On m'avait envoyé les questionnaires que mes stagiaires avaient

remplis. Certains avaient complété le questionnaire dans leur traitement de texte, puis l'avaient imprimé et de nouveau numérisé pour qu'il me soit communiqué au format PDF non modifiable. Malheureusement, ce document étant une image, je ne pouvais pas en prendre connaissance à l'aide de mon terminal braille ou le lire avec ma synthèse vocale. J'ai donc encore perdu du temps car j'ai dû me le faire lire par une personne qui voyait le document. Si le questionnaire avait été communiqué au format électronique classique, j'aurais pu le lire moi-même, en faire un copier/coller pour analyser les différentes réponses, retrouver son contenu ultérieurement en faisant une recherche de texte inclus dans le document.

Le plus gros problème en accessibilité numérique est l'absence d'équivalents textuels aux images. L'utilisateur auquel on pense spontanément est la personne aveugle qui ne voit pas les images et ne peut accéder à leur contenu si l'équivalent textuel est absent. Afin de parer à ce problème, le logiciel de lecture d'écran utilisé par l'internaute aveugle va lui communiquer l'adresse de l'image ou le nom du fichier qui est souvent un texte incompréhensible. Cette absence d'informations est particulièrement gênante lorsque l'image est un lien, la personne ne saura pas ce que permet de faire ce lien. Si l'image contient une information, telle qu'un numéro de téléphone, l'absence d'équivalent textuel empêchera le lecteur aveugle de prendre connaissance du numéro de téléphone écrit dans l'image.

Une personne avec des difficultés motrices, qui se sert de la dictée vocale pour naviguer sur internet, n'accèdera pas non plus au lien image, car son logiciel se base sur ces mêmes équivalents textuels pour trouver le lien à valider. Internet offre de nos jours de plus en plus de contenus vidéos dont la bande son est inaccessible à une personne sourde si cette vidéo ne comporte pas de sous-titres ou une traduction en langues des signes. Si l'internaute ne dispose pas de la technologie permettant de lire la vidéo, il ne pourra pas non plus prendre connaissance de son contenu si le site ne propose pas une transcription textuelle de la vidéo. Ce texte doit reprendre la totalité des dialogues, les bruits, les sons, et la description de tout visuel ou texte apparaissant à l'écran et non vocalisé. Cette transcription permettra là encore aux personnes sourdes ou

malentendantes, aux personnes aveugles ou malvoyantes, ou à celles qui n'ont pas de lecteur vidéo, de prendre connaissance de ce contenu multimédia.

Bénéfices concrets des contenus accessibles

Les équivalents textuels

Un équivalent textuel aide les personnes qui ont des difficultés à percevoir le contenu visuel. Les technologies d'assistance permettent à ces personnes de lire ce contenu à haute voix, de changer leur présentation visuelle ou de l'afficher en braille. Ces équivalents textuels permettent d'identifier la signification d'images telles que des photos, dessins, peintures, graphiques, diagrammes ou autres animations. Les personnes sourdes, malentendantes ou qui ont de la difficulté à comprendre l'information audio peuvent lire la présentation textuelle de ce contenu audio. Des recherches sont également en cours concernant la possibilité de traduire ce texte de façon automatique en langue des signes. Les personnes sourdes-aveugles pourront lire l'équivalent textuel en braille. De plus, les équivalents textuels donnent la possibilité de rechercher le contenu non textuel et de réutiliser ce contenu de diverses façons.

Permettre l'interaction avec la souris et le clavier

Lorsqu'une action peut être effectuée non seulement via la souris mais aussi au clavier,

cela est utile aux personnes aveugles qui se servent difficilement d'une souris, car elle nécessite une coordination de l'œil et de la main). Cela est également utile aux personnes mal voyantes, qui peuvent avoir du mal à trouver ou à suivre le symbole du pointeur sur l'écran.

Enfin, cela concerne les personnes ayant des tremblements de la main qui préfèrent l'utilisation d'un clavier à celle de la souris.

L'accessibilité, nécessaire pour certain(e)s, utile à tou(te)s

Même si de nos jours beaucoup d'informations sont disponibles sur Internet, encore faut-il pouvoir y accéder. Si les recommandations ne sont pas respectées, beaucoup d'utilisateurs n'accéderont pas à toute l'information, ne pourront pas accomplir une tâche jusqu'au bout. Les personnes débutantes n'auront pas envie d'utiliser le numérique pour lire l'information.

Si, au contraire, un contenu numérique est accessible, n'importe qui pourra le consulter, il sera possible de le personnaliser en l'affichant de différentes façons : gros caractères, dans une autre couleur, avec une police qui permettra une lecture plus facile pour une personne dyslexique par exemple, de la lire en braille ou de l'écouter grâce à une voix de synthèse. Un internaute qui ne peut pas se servir de ses mains pourra, à l'aide de ses outils, interagir avec le contenu, le faire défiler,

valider des informations ou demander à son outil d'activer un lien image, par exemple.

L'accessibilité numérique est l'effort minimum qui permettra au plus grand nombre d'internautes d'exploiter l'information. Un contenu accessible est en outre plus facile à gérer, modifier par les équipes de maintenance des contenus car il est en général mieux balisé et structuré et son aspect peut être facilement et rapidement modifié, sans avoir à retaper des centaines de lignes de codes.

L'accessibilité numérique facilitera l'intégration de tous les utilisateurs qui pourront appréhender un document sans aide extérieure, de façon indépendante. Un document accessible, avec des contrastes suffisants, pourra être plus aisément imprimé et reconnu par les outils de numérisation. S'il contient des mots clés, une structure bien identifiée, il pourra être plus facilement répertorié et retrouvé dans une multitude de documents.

Même si beaucoup pensent que faire accessible est coûteux au début, cela permet ensuite d'économiser de l'énergie car le contenu peut être rapidement modifié, mis à jour et indexé. Si certains utilisateurs ne peuvent pas accéder à une information qui ne respecte pas les normes, n'importe quel internaute profitera d'une information accessible car elle sera plus facile à lire, les interactions plus rapides et la personnalisation plus variée.



Dyslexie : de multiples impacts dans l'univers numérique

La dyslexie est un mal méconnu qui touche 6 à 10 % de la population. Pourtant des leaders de haut niveau dans le secteur numérique sont dyslexiques comme John Chambers de Cisco, Bill Gates de Microsoft et comme l'était Steve Jobs, et certainement aussi beaucoup de vos collègues ou vous-même, parfois sans même en avoir conscience comme nous le constatons souvent.



Philippe Rosado
Président de Numeridys
Liberté 0,
Accessibilité des
logiciels libres aux
personnes handicapées

Car nous essayons d'oublier certaines difficultés scolaires, parfois accompagnées de séances d'orthophonie alors que la dyslexie est un trouble durable même s'il peut être compensé.

Elle touche également de nombreux utilisateurs de vos services ou applications et il est important d'en tenir compte.

Qu'est-ce que la dyslexie ? Est-ce un handicap ?

Littéralement la dyslexie est une difficulté à lire souvent accompagnée d'une mauvaise orthographe, alors même que la personne est intelligente et a été scolarisée normalement.

Il y a plusieurs types de dyslexie, et leur niveau est plus ou moins important.

Elles sont souvent combinées entre elles et à d'autres troubles tels que la dysorthographe, la dysgraphie, des troubles de mémoire ou de l'attention, etc...

De plus ces situations sont souvent vécues sans que la personne ne le sache, ou ne veuille en accepter la réalité et choisisse le déni de ses difficultés de peur d'être exclue. Ainsi le patron de Virgin, Richard Branson explique qu'il

a du mal à faire une note à ses collaborateurs, c'est pourtant un manager hors norme ! Il n'est donc pas simple de donner une recette universelle pour apporter des réponses efficaces à des situations multiples.

Enfin, le Français vient compliquer encore plus les choses car c'est une langue opaque. C'est-à-dire qu'à un son ne correspond pas toujours la même lettre. Par exemple en Français on trouve de nombreuses transcriptions du son « O » : « o » bien sûr mais aussi « au » « aux » « eau » « haut » « ot », etc... à la différence des langues directes, telles que l'Espagnol ou l'Allemand où le son « O » s'écrit toujours « O ».

Cela crée des difficultés qui viennent tant compliquer la lecture que l'orthographe. On comprend alors aisément que face à ces difficultés les personnes vont choisir des carrières où elles y échapperont le plus possible. Ainsi on retrouve de nombreux programmeurs qui valorisent leurs qualités de logique et de synthèse et de créativité dans les métiers du numérique. Si c'est votre cas sachez que d'après *The Economist* on assiste à une nouvelle tendance : l'attrance de plus en plus grande pour les personnalités hors normes. Les créatifs, les rebelles et les excentriques sont maintenant recherchés dans toutes sortes de domaines.

Beaucoup de chefs d'entreprises présentent aussi des singularités et on y retrouve de nombreux dyslexiques. Lors d'une enquête menée au sein d'un groupe de chefs d'entreprises, Julie Login de la Cass Business School avait trouvé que 35% d'entre eux étaient dyslexiques, alors que les dyslexiques ne représentent que 10% de la population normale, et seulement 1% des gestionnaires. Les fondateurs de Ford, General Electric, IBM et Ikea sont dyslexiques. Les dyslexiques, qui ont des difficultés à la lecture, apprennent vite à déléguer, ce qui les prépare à la direction d'entreprise.

Plus de 50 % des employés de la NASA sont dyslexiques. On les recherche tout spécialement en raison de leurs aptitudes hors du commun à résoudre les problèmes et de leur excellente perception spatiale et tridimensionnelle.

(cf : http://fr.scips.eu/subjects_and_disabilities.html/informatique/info_dyslexie.html)

Les dyslexiques qui ont réussi ont su transformer leur handicap invisible en tremplin. Mais le parcours chaotique et en souffrance des élèves dyslexiques les conduit le plus souvent à l'échec. Il peut également s'installer des incompréhensions avec les collègues qui vont attribuer des fautes d'orthographe ou des comptes rendus de faible qualité à un manque de motivation ou pire à de la négligence. Ainsi si vous demandez à un candidat lors d'un recrutement d'écrire à la main un document il se retrouvera en difficulté, il est recommandé de permettre l'usage des correcteurs d'orthographe et une certaine tolérance ainsi d'éviter de juger un candidat ou un collègue uniquement sur des écrits. Privilégier un

« Les dyslexiques sont surreprésentés parmi les personnes exceptionnellement perspicaces, qui apportent une nouvelle perspective et qui pensent en dehors du cadre. »

entretien téléphonique ou présentiel chaque fois que possible. Enfin une part importante des utilisateurs d'applications, de sites Internet et de contenus numériques en général qui sont dyslexiques se trouvera en difficulté si certaines règles ne sont pas observées.

Programmeurs, comment pouvez-vous aider les dyslexiques à utiliser vos services ?

Nous pouvons les aider en adaptant les sites Internet. De plus ces adaptations, souvent très simples, vont faciliter la lecture du site aux dyslexiques mais aussi aux malvoyants, aux personnes âgées et à la population en général. Lire nécessite la coordination d'un grand nombre de capacités sensorielles et cognitives. Ces activités sont très complexes pour les « dys » et nécessitent beaucoup d'efforts. Pour cette raison rendre lisible son site est important.

Le choix de la police de caractères est important, Serif ou Sans Serif ?

Les polices d'écriture Serif à « empattements », présentent des marques décoratives qui forment la terminaison des lettres et vont en brouiller l'image. Les polices Sans Serif sans empattement plus simples et nettes seront plus rapidement perçues. La taille ne doit pas être trop réduite. Par exemple une police Arial et une taille de caractères minimum 12 sont une bonne base, en évitant les faibles contrastes sur de mêmes tons de couleurs.

Éviter de multiplier les pop-up, les phrases qui déroulent, les mots clignotants, bref tout ce qui rend une page trop fouillis.

Pour faciliter la lecture des sites il faut bien structurer la page, aérer vos pages et faire que vos articles puissent être lus par des robots, ce qui permettra l'usage d'un oralisateur.

Vous pouvez également enrichir un texte d'un schéma, d'une image ou d'une vidéo qui en complèteront la compréhension et la mémorisation. De plus une grande majorité de « Dys » sont aussi dysorthographiques, lors de l'utilisation d'un moteur de recherche il est important que celui-ci prenne en compte des variations d'orthographe.

Enfin si vous êtes un programmeur « dys » ou sensible à ce sujet, nous souhaiterions connaître votre expérience, votre parcours, vos astuces afin d'en faire bénéficier les projets que nous portons pour les dyslexiques. Nous sommes également très désireux de contacts dans l'univers de l'offre pédagogique car des transformations importantes se produisent dans cet univers.

Enfin si vous êtes parents soyez attentif aux éventuelles difficultés de lecture ou de mémorisation de vos enfants. Si un diagnostic confirme une dyslexie sachez que vous avez un avantage avec votre métier pour aider votre enfant et n'hésitez pas à faire appel à nous car le lien entre un outil numérique et son usage en compensation nécessite une certaine expérience. Le numérique est une formidable opportunité pour de nombreuses personnes, à nous de transformer le rêve en réalité.

N'hésitez pas à nous en parler ou nous poser vos questions : contact@numeridys.org



Votre Abonnement **PDF**
pour seulement **30 € par an**
(soit **2,73 € le numéro**)
www.programmez.com

Tester l'accessibilité de son code

L'acte le plus élémentaire, lorsque l'on veut rendre une page Web accessible, est de vérifier que son code respecte bien les recommandations des référentiels. C'est la toute première étape, indispensable, avant un audit complet et les tests utilisateurs.



Olivier Nourry,
Consultant et formateur en
accessibilité, auteur de
accessiblog.fr

L'application complète et précise des référentiels ne s'apprend pas en un article... Cependant plusieurs techniques de base peuvent être appliquées par tout développeur équipé d'un navigateur, des bonnes extensions, et de son bon sens. Cet article en présente quelques-unes.

Quels tests ?

On va aborder ici des tests dits "de présence", c'est-à-dire que l'on va vérifier la présence d'un dispositif permettant l'accessibilité d'un contenu. Par dispositif, on entend un élément de code HTML, un composant, ou parfois un simple texte permettant de renseigner efficacement l'utilisateur. L'autre grande famille de tests se compose des tests "de pertinence". Ils ne seront pas abordés ici, car on pourrait écrire tout un livre sur le sujet... et ils sont plutôt du ressort des contributeurs qui vont utiliser le code mis à leur disposition par les développeurs.

Quels outils ?

Des outils de tests automatiques, dédiés à l'accessibilité, permettent de tester du code "usine", des pages complètes, voire des scénarios de navigation. Vous pouvez par exemple soumettre vos pages à des outils comme Tanaguru ou Opquast Reporting, et scripter des tests de code avec un outil comme Tenon. Ils détecteront et vous signaleront de façon précise et exhaustive tout ce qui relève d'un code mal formé, les attributs manquants ou incorrects, les problèmes de contraste, etc. Mais on estime qu'environ 20% seulement des tests de conformité sont réellement automatisables. Pour la grande majorité des tests, il va donc falloir passer par des procédures manuelles. Commencez donc par faire tourner un ou plusieurs de ces outils sur votre code, et pour le reste, suivez le guide... Le test d'accessibilité impose de se placer du point de vue de l'utilisateur final. Le principal outil de test manuel est donc un navigateur. Ma préférence pour cet exercice va à Firefox complété par deux extensions, la barre d'outils "Web Developer" (WDT) et le plug-in Firebug,

déjà bien connus des développeurs Web. Navigateur, prêt ? Extensions, installées ? Page Web, ouverte ? C'est parti...

Le fonctionnement au clavier

La souris n'est pas le seul outil de pointage qu'utilisent les internautes, en particulier ceux en situation de handicap. Trackballs, claviers classiques ou adaptés, périphériques tactiles, commande vocale, eye-tracking, tiges fixées sur le front ou tenues en bouche, contacteurs actionnés par la bouche, un doigt, un pied, ou le souffle... la variété des solutions est impressionnante. De plus, l'usage de la souris est problématique si l'on ne voit pas, ou pas bien, le curseur. C'est donc l'un des aspects clés de l'accessibilité, et il importe de tester ses pages Web sous cet angle, surtout si elles comportent des composants d'interface riches. En HTML standard, les liens hypertexte et les éléments de formulaire (on parle "d'éléments focusables") sont nativement accessibles à la souris et au clavier. Mais parfois on va ajouter des comportements non standards, en JavaScript, comme par exemple l'affichage d'une bulle d'aide customisée au survol souris. Il va falloir également afficher cette bulle d'aide à la prise de focus au clavier. Bonne nouvelle : il suffit de tester l'accès au clavier et à la souris pour quasiment couvrir tous les cas d'usage. Et comme la prise en charge de la souris est quasiment systématique, on va surtout s'intéresser à l'accès au clavier.

Les tests

Ouvrez la page testée, et appuyez sur la touche [Tab] (tabulation). Sur une page accessible, le premier élément focusable prendra le focus, de façon visible. Appuyez de façon répétée sur [Tab], jusqu'à atteindre le dernier élément. Vérifiez que vous passez bien sur tous les éléments cliquables, selon un ordre logique (c'est-à-dire prévisible), et que le focus (ou le curseur de saisie) est toujours visible. Refaites le même parcours à l'envers, en appuyant sur [Tab] tout en maintenant la touche [Maj] (majuscule) enfoncée. Vérifiez que :

- Tous les éléments focusables sont atteignables au clavier,
- Les éléments interactifs peuvent être activés au clavier, par la touche Entrée en général (ou Espace pour une case à cocher ou un bouton radio). Méfiez-vous en particulier des

menus déroulants: assurez-vous qu'ils sont déroulés à la prise de focus comme ils le sont au survol souris,

- Il n'y a pas de piège au clavier (vous n'êtes jamais bloqué pour continuer la tabulation). Cela arrive parfois avec des composants comme les carrousels ou certains widgets qui provoquent un changement de page à la prise de focus,
- Le parcours du focus est intuitif : vous devez pouvoir deviner facilement le prochain élément atteint,
- Le focus rente dans la zone de visualisation, que ce soit la page, ou une pop-in,
- Le focus est visible, au moins autant qu'avec le réglage navigateur par défaut.

La visibilité du focus n'est pas toujours évidente sur Firefox, le liseré (appelé aussi caret) étant assez discret. En cas de doute, essayez sur Chrome ou Safari, où il est plus apparent. Si le doute persiste, vérifiez la propriété CSS outline de l'élément concerné, grâce à Firebug. Elle ne doit pas être dégradée par rapport au style par défaut de Firefox, qui est "#000 1px dotted".

Les cas particuliers

Sur un groupe de boutons radio, le focus se positionne sur le premier bouton dans l'ordre de tabulation. Tabuler à nouveau doit vous faire sortir du groupe de boutons radio, les flèches de direction vous permettant de naviguer dans le groupe. Certains composants d'interface riches devront bénéficier d'une gestion particulière du clavier. Il s'agit des composants codés en respectant le standard WAI-ARIA, ou ARIA (Accessible Rich Interface Applications). Reportez-vous aux règles d'implémentation définies par le W3C(1) pour traiter ces cas particuliers.

La notion d'alternative

Parfois, on intègre des composants provenant d'une bibliothèque que l'on ne peut pas modifier, et qui ne sont pas accessibles au clavier. Les référentiels prévoient alors la possibilité de recourir à un dispositif alternatif, qui, lui, sera accessible. Imaginons un menu déroulant ne fonctionnant qu'à la souris : si on retrouve l'arborescence de ce menu sous forme de listes de liens en pied de page, l'accès au clavier est possible, l'utilisateur n'est pas bloqué.

(1) http://www.w3.org/WAI/PF/aria-practices/#aria_ex

Aller plus loin : la gestion du focus

Les éléments d'interface riches tels que fenêtres modales, carrousels ou onglets peuvent justifier un parcours du focus non linéaire. On va alors "manipuler" le focus, c'est-à-dire forcer son déplacement selon un ordre adapté, ou dans des zones limitées. Par exemple, si une fenêtre modale de type *pop-in* est ouverte, le focus doit rester "prisonnier" de cette fenêtre, et ne pas circuler dans la page en arrière-plan. Et lorsque l'on ferme la fenêtre, le focus doit être repositionné sur le lien ou le bouton qui a permis de l'ouvrir.

La gestion du focus est particulièrement importante pour les utilisateurs qui n'ont pas une perception d'ensemble de la page.

Les images

Autre grande thématique, très emblématique, de l'accessibilité : les images. Comment résoudre le paradoxe consistant à rendre accessibles ces éléments purement visuels à des utilisateurs qui ne peuvent pas les voir ? En fournissant avec ces images leur alternative textuelle, qui pourra être restituée par technologie d'assistance telle qu'un lecteur d'écran par exemple.

L'alternative textuelle dépend totalement du contenu de l'image et de son contexte d'utilisation. Mais ce sont là des considérations de pertinence. Nous allons décrire ici quelques techniques permettant de détecter et visualiser facilement le cas le plus fréquent : les alternatives textuelles des images implémentées avec la balise ``, via son attribut `alt`. Pour ces tests, affichez la page avec CSS désactivé (avec la WDT : CSS > Désactiver les styles CSS > Tous les styles).

Détecter les images sans alt

Quel que soit le rôle de l'image, la balise `` doit comporter un attribut `alt`, même s'il est vide. En son absence, une technologie d'assistance restituera le contenu de l'attribut `src`, ce qui est rarement exploitable pour l'utilisateur. Sélectionnez l'option de la WDT : Images > Entourer les images > Images sans attribut `alt`. Toutes les images entourées en rouge ont un attribut `alt` manquant.

Afficher les alt des images

Grâce à la WDT, vous pouvez soit remplacer les images par leur alternative (Images > Remplacer les images par l'attribut `alt`), soit afficher l'alternative avec l'image (Images > Afficher les attributs `alt`). Les deux méthodes ont leurs avantages, à vous de trouver celle qui convient le mieux à la situation.

Ce que doit être cette alternative est un vaste

sujet, comme en témoigne le document du W3C : "*HTML5: Techniques for providing useful text alternatives(2)*"... Et on déborde largement du cadre de cet article. Il y a toutefois des cas simples à traiter :

- Une image de pure décoration, ou servant à la mise en page, doit avoir une alternative vide,
- Une image qui est le seul élément d'un lien ne doit pas avoir d'alternative vide,
- Une image légendée, ou nécessitant une description longue (comprendre : texte long de plusieurs phrases, ou structuré), ne doit pas avoir d'alternative vide,

Il est généralement inutile de préciser "image de..." ou "logo...", l'utilisateur étant informé par sa technologie d'assistance qu'il s'agit d'une image.

Les titres de section

Les titres de sections, *headings* en anglais, sont implémentés en HTML via les balises `<h1>` à `<h6>`. Ils permettent de structurer la page de façon hiérarchisée. Les niveaux doivent se succéder dans un ordre logique, sans "saut" de niveau. Et si une section comporte un titre de niveau N, les titres de niveau N+1 qui suivent vont découper cette section en autant de sous-sections, et ainsi de suite. Cette hiérarchie est très importante pour les utilisateurs de lecteurs d'écran, qui ont à leur disposition des fonctions permettant de naviguer rapidement dans la page via cette hiérarchie, un peu comme dans une table des matières interactive.

Les tests

Avec la WDT, activez l'option Infos > Plan du document. Le navigateur ouvrira une page où les différents titres présents dans la page sont affichés sous forme d'arbre hiérarchisé, avec leur niveau. Vérifiez que :

- Le premier titre est de niveau 1 (il en faut au moins un par page),
- Il n'y a pas de "trou" dans la hiérarchie (la WDT affiche "Titre manquant"),
- Il n'y a pas de titre vide,
- Le contenu et le niveau des titres sont cohérents avec ce que l'on perçoit visuellement.

Rappelez-vous que, comme souvent, il faut rester mesuré dans la fourniture de titres : une hiérarchie trop lourde (avec trop de titres, et trop de niveaux) aurait un effet négatif sur l'utilisabilité du contenu.

Aller plus loin : les titres en HTML5

Le HTML5 introduit le concept d' "élément sectionnant", associé à un algorithme de hiérarchisation "relative" des titres, très

séduisant dans son principe. Selon ce concept, un élément sectionnant comporte une hiérarchie de titres qui lui est propre, et donc indépendante de la hiérarchie d'ensemble de la page. Ainsi, un bloc autonome de contenu pourra être inséré dans une page, et son titrage s'adaptera à la hiérarchie des titres déjà en place. Seulement, c'est d'une telle complexité de mise en œuvre que les navigateurs et les technologies d'assistance ne semblent pas prêts de l'adopter... En attendant, malgré ses limitations, tenez-vous en à la méthode traditionnelle de titrage, c'est la seule qui soit fiable dans tous les contextes d'utilisation.

Les titres de pages

Le titre d'une page HTML est défini via la balise `<title>` que l'on trouve dans le `<head>`. Ce titre est la première information restituée par un lecteur d'écran. Sa présence et sa pertinence sont donc très importants pour se retrouver facilement dans les différentes fenêtres ouvertes, et savoir où l'on se trouve actuellement. Si la détection de cette information ne pose pas de problème (il suffit d'ouvrir le source), certains cas particuliers nécessitent plus d'attention.

Les contenus paginés

Les contenus paginés sont répartis sur une succession de pages, dans le cas par exemple d'un catalogue, d'un formulaire en plusieurs étapes, ou d'un grand nombre de résultats de recherche. Assurez-vous alors que le titre indique bien le numéro de la page courante, ainsi que le nombre total de pages. Exemple : "[titre de page] - page 2 sur 6".

Les résultats de recherche

Outre la pagination, rappeler les mots-clés utilisés permet à l'utilisateur de faire la distinction entre les différentes pages de recherche qu'il a ouvertes.

Les formulaires

Si vous réaffichez la page pour correction, suite à l'envoi d'un formulaire, il faut signaler à l'utilisateur la présence d'une erreur. Sinon, il pourra penser que tout s'est bien déroulé, et qu'il peut passer à autre chose.

Ce n'est qu'un début...

On n'a couvert ici qu'une petite partie des tests d'accessibilité prévus par les référentiels. Cependant, si vous les appliquez à votre code, vous détecterez déjà une partie importante des problèmes que rencontrent trop souvent les utilisateurs - et croyez-moi, ils vous en seront reconnaissants !

(2) <http://www.w3.org/TR/html-alt-techniques/>

Accessible Rich Internet Applications

La généralisation des applications Web développées via Javascript a rendu nécessaire une modification majeure dans la manière de gérer l'accessibilité des contenus Web. Commencé très tôt (dès 2005), ce travail a abouti à la publication d'une API importante dans l'écosystème HTML5. Son petit nom est ARIA (Accessible Rich Internet Application). Elle est destinée à rendre les composants riches, développés avec Javascript, plus accessibles dans deux modes de consultation plus précisément : via un lecteur d'écran, pour les utilisateurs aveugles ou malvoyants, et via l'utilisation exclusive du clavier, pour les utilisateurs qui ne peuvent pas bénéficier d'un système de pointage tel que la souris par exemple.



Jean-Pierre Villain
Directeur technique
Access42
<http://access42.net>

Avant de rentrer plus précisément dans le détail, il convient de rappeler quelques notions de bases. Un lecteur d'écran, dont la mission est de vocaliser les contenus et de permettre aux utilisateurs aveugles ou malvoyants de les utiliser, est un logiciel particulièrement complexe. L'une des choses la plus importante à comprendre est qu'un lecteur d'écran n'est pas un navigateur : il travaille essentiellement avec les informations transmises par le navigateur via le "DOM accessible" et l'API d'accessibilité système. Le DOM accessible est une copie dynamique et instantanée du DOM, enrichie, pour chaque élément qui le nécessite, des informations essentielles à l'accessibilité. L'API d'accessibilité stocke toutes les informations disponibles, fournies par le navigateur, sur les éléments du contenu sous la forme de couples propriétés/valeurs statiques ou dynamiques. L'ensemble de ces trois acteurs, le développeur qui produit le contenu, le navigateur qui l'interprète et le lecteur d'écran qui le restitue, constitue un système complexe et délicat surtout si l'un de ces acteurs ne joue pas correctement son rôle.

Case à cocher, non-cochée, oui

Prenons par exemple le cas d'une case à cocher et regardons rapidement quelles informations sont fournies à l'API d'accessibilité système par

le navigateur **Fig.1**. Nous voyons ici plusieurs informations essentielles que nous aurons l'occasion de retrouver par la suite, comme le rôle : "case à cocher", le nom : "oui" (ce qui correspond à l'étiquette LABEL), l'état : "défilant, pouvant être actif", ou encore la valeur : "" (cette propriété est nulle, une case cocher n'a pas de valeur, juste un état), enfin nous remarquons que, lorsque la case à cocher est cochée, des informations sont ajoutées à la propriété "état" : "actif, coché", dont on comprend bien le sens. Le lecteur d'écran va puiser dans ces deux sources d'information (le DOM accessible et l'API d'accessibilité) pour retranscrire, vocalement, l'élément en cours de consultation. NVDA avec Firefox, par exemple, annoncera, lorsque l'utilisateur activera l'élément via la prise de focus : « case à cocher, non cochée, oui » puis « oui, case à cocher, cochée » lorsque l'utilisateur aura coché la case. Cela établit une espèce de dialogue permanent, à chaque modification du contenu et actions de l'utilisateur, entre le lecteur d'écran et le navigateur. Reste à expliquer d'où viennent, *in fine*, ces informations sans lesquelles il n'y aurait tout simplement pas d'accessibilité possible.

La spécification, source de toute chose.

Là en revanche, les choses vont être beaucoup plus simples. En HTML toutes ces informations viennent du langage lui-même : le rôle d'un élément, c'est la sémantique de son balisage (le type "checkbox" pour la case à cocher), certaines

de ses propriétés sont intrinsèques, c'est-à-dire définies par la spécification, et gérées directement par le navigateur (comme l'état coché ou non-coché) mais d'autres dépendent de l'intégration c'est-à-dire de vous-même.

Bienvenue dans le monde réel

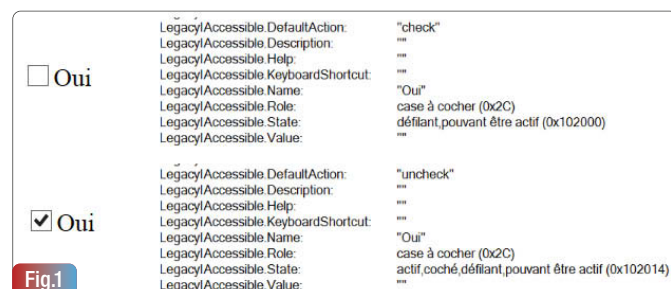
J'ai choisi cette citation célèbre de Morpheus dans Matrix pour donner une idée du changement d'univers dès lors que vos contenus vont s'enrichir de composants développés par Javascript.

Voici le code HTML d'une case à cocher, maquillée comme une voiture volée via CSS et gérée intégralement via Javascript, comme vous avez pu en faire ou comme on peut en rencontrer :

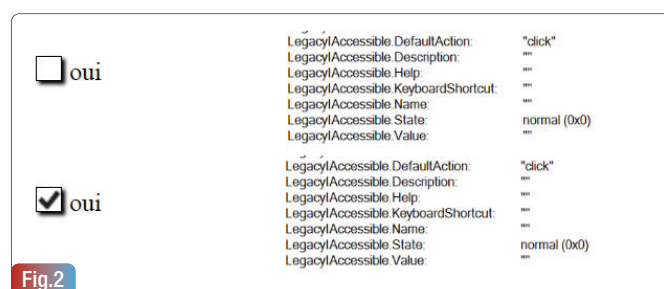
```
<span></span> oui
```

Si vous avez lu attentivement jusque-là vous comprendrez aisément que dans cette situation, du point de vue de l'accessibilité, c'est la loose : **Fig.2**.

Plus de rôle, plus de nom, un état "normal" (ce qui signifie que l'API d'accessibilité n'a rien à en dire) et pas de valeur (ce qui pour le coup est la seule propriété normale). Coté utilisateur c'est la catastrophe : NVDA va dire "clicable oui", il va dire cliquable car l'événement Javascript qui va vous permettre de simuler l'action "cocher" est la seule information que vous communiquez, l'utilisation de ce machin pour l'utilisateur est en réalité mission impossible. Si vous voulez rendre ça accessible, il va falloir vous insérer dans le système complexe décrit



Aperçu des informations disponibles via MSAA/Windows pour les deux états, non-cochée et cochée d'une case à cocher



Aperçu des informations disponibles via MSAA/Windows pour les deux états, non-cochée et cochée d'une fausse case à cocher développée via Javascript

plus haut car le navigateur, bien incapable d'identifier le composant ne peut rien transmettre d'utile au lecteur d'écran qui, par voie de conséquence, laissera l'utilisateur totalement désarmé face à un composant incompréhensible.

Il y a encore pire : l'utilisateur qui ne peut pas utiliser la souris et qui navigue avec la tabulation va très bien comprendre qu'il s'agit d'une case à cocher sans pouvoir l'utiliser : encore plus frustrant, non ?

Percevoir, identifier, comprendre et utiliser

C'est là qu'intervient l'API ARIA dont la baseline est "role, state and properties" et qui n'est qu'une longue liste de rôles et de propriétés associés à des descriptions des comportements, des structures et des interactions au clavier à respecter (ce qu'on appelle des "design patterns", motifs de conception en français). Grâce à de simples attributs, rendons votre case à cocher accessible :

```
<span role="checkbox" aria-checked="false" tabindex="0" aria-labelledby="foo"></span><span id="foo">oui</span>
```

Quand la case à cocher est "cochée", changez la valeur de la propriété aria-checked et passez-la à "true" via Javascript. C'est terminé, votre job est fait et bien fait, c'est accessible. Regardons ce que cela donne du point de vue de l'API d'accessibilité : **Fig.3.**

Nous obtenons quelque chose de très similaire à une vraie case à cocher HTML que nous pouvons à présent tester. Par exemple NVDA va vocaliser ce "nouveau composant" : "case à cocher, non cochée, oui" et "case à cocher, cochée, oui".

Le tour de passe-passe est presque parfait ! Arrêtons-nous un instant sur certaines caractéristiques de cette implémentation comme la propriété aria-labelledby qui vous permet de lier deux contenus ensemble et d'indiquer que le contenu qui est référencé via sa valeur d'ID peut être utilisé comme "nom" du composant. Il s'agit d'un mécanisme similaire à

celui d'un élément label HTML qui va être très utilisé pour les composants ARIA.

Enfin l'attribut tabindex est utilisé par ARIA pour donner (valeur 0) ou retirer (valeur -1) à un élément HTML la possibilité d'être utilisé au clavier. Enfin dans le même registre vous devrez, pour être complet, donner à l'utilisateur la possibilité de cocher la case avec la barre d'espace, comme c'est le cas pour l'élément natif HTML.

Dans certains cas le niveau de complexité de l'utilisation au clavier peut devenir important et nécessiter de gérer les flèches de direction, la touche escape, la touche entrée, la barre d'espace ainsi qu'un certain nombre de raccourcis au clavier destinés à en améliorer le support. La gestion du clavier est sans doute ce qui vous posera le plus de difficultés.

Vers l'infini et au-delà

ARIA va bien au-delà de cette mission essentielle d'informer et de permettre l'utilisation, via le navigateur et le lecteur d'écran, des composants personnalisés et en Javascript. Certains rôles et certaines propriétés jouent des rôles particuliers destinés à améliorer considérablement l'utilisation de vos applications.

Les landmark rôles

Prenons par exemple une catégorie de rôle appelée "landmark" (région) que vous rencontrerez de plus en plus souvent car ils sont particulièrement profitables à l'utilisateur. Ces rôles vont vous permettre d'insérer des points de navigation dans la structure du document qui pourront être parcourus via un raccourci clavier (par exemple le raccourci "D" dans NVDA).

Les plus importants sont : "banner", pour indiquer la zone d'en-tête de votre page, "navigation" pour indiquer le menu de navigation principal, "search" pour indiquer le moteur de recherche, "main" pour indiquer la zone de contenu principale et "contentinfo" pour indiquer le "pied de page". Ils s'implémentent aussi simplement que ça : <div role="banner"> en HTML4 ou <header

role="banner"> en HTML5 qui dispose d'un élément spécifique aux en-têtes de contenu.

Les live régions

Cette propriété a pour fonction de résoudre un des problèmes les plus complexes dans les applications Web et qui se résume à cette problématique : comment informer l'utilisateur qu'une partie de la page vient d'être mise à jour (par exemple un système de panier mis à jour via AJAX) ?

Lorsque vous avez besoin de signaler à l'utilisateur qu'une partie du contenu vient d'être mise à jour en résultat d'un événement qu'il vient de déclencher, vous devriez utiliser la propriété aria-live. Celle-ci accepte deux valeurs diablement utiles : "polite" aura pour effet de signaler au lecteur d'écran de vocaliser la zone mise à jour dès que l'utilisateur sera disponible (il n'effectue plus d'action au clavier par exemple), "assertive", à utiliser avec circonspection, signalera au lecteur d'écran de vocaliser la zone mise à jour immédiatement. Aria-live peut être accompagnée de propriétés complémentaires permettant d'affiner la restitution, par exemple ne vocaliser que les ajouts ou les suppressions, que le texte ou tout le contenu. Le fonctionnement, sous le capot, est simple à comprendre : toute zone possédant une propriété aria-live sera vocalisée dès que le navigateur détectera un changement dans le DOM, par exemple l'ajout ou la suppression d'un node texte, voire d'un simple attribut. Vous en aurez probablement besoin très souvent, et faute de la maîtriser, tout ce que vous pourriez faire par ailleurs ne servirait pas à grand-chose.

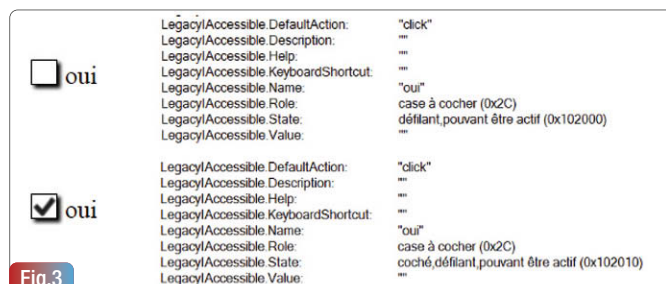
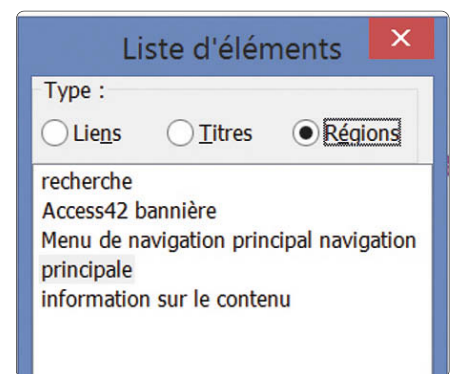
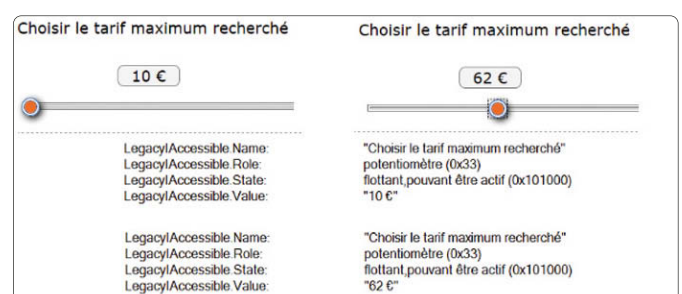


Fig.3

Aperçu des informations disponibles via MSAA/Windows pour les deux états, non-cochée et cochée d'une fausse case à cocher développée via Javascript et rendue accessible via ARIA



Ne soyez pas dupes de vos nouveaux pouvoirs, restez humbles !

À bien des égards, une fois apprivoisées ses subtilités, et surtout si vous avez un profil de développeur d'applications, vous finirez par considérer que l'utilisation d'ARIA est la réponse à tous vos problèmes. Plus besoin de négocier avec des éléments HTML qui peuvent se montrer rétifs au design CSS, à l'interopérabilité, à l'industrialisation... J'utilise ce que je veux car je peux tout modifier et tout adapter. Cela a conduit le groupe de travail, en charge de l'édition et de la maintenance de l'API, à proposer des règles de bonne conduite, décrites dans la note technique "Using WAI-ARIA in HTML" dont vous trouverez l'URL dans la partie ressources de cet article. Lisez cette note, relisez-la, ayez-la sous le coude, et suivez autant qu'il vous est possible de le faire ses recommandations.

Je ne citerai ici que la première règle qui sera votre ligne de vie dans l'élaboration d'applications Web réellement accessibles : "Si vous pouvez utiliser un élément HTML natif ou un attribut dont la sémantique et le

comportement sont prédéfinis, au lieu de redévelopper un élément en y ajoutant un rôle, un état ou une propriété ARIA pour le rendre accessible, faites-le !" Sachez enfin qu'ARIA est implémentée par tous les navigateurs récents et les principaux lecteurs d'écrans, que les bibliothèques Javascript ou les frameworks comme JQuery ou bootstrap, par exemple, tentent de proposer, par défaut, des implémentations ARIA correctes, ce qui vous facilitera la vie, même si vous devrez sans doute vous coltiner des adaptations.

Par où débiter ?

Les documents de référence

- Accessible Rich Internet Applications (WAI-ARIA) 1.0 (<http://www.w3.org/TR/wai-aria/>)
Il s'agit de l'API elle-même.
- WAI-ARIA 1.0 Authoring Practices (<http://www.w3.org/WAI/PF/aria-practices/>)
Le complément indispensable, ce document contient les design patterns à respecter notamment.
- Using WAI-ARIA in HTML (<http://w3c.github.io/aria-in-html/>)
Votre boussole.

Les sites de démonstrations et d'exemples

- OpenAjax Examples, (<http://oaa-accessibility.org/examples/>)
Incontournable même si certains exemples mériteraient une mise à jour.
- Accessible jQuery-ui Components Demonstration, (<http://hanshillen.github.io/jqtest/>)
Ce fork de JQuery propose 12 composants rendus accessibles, dans certaines situations il peut même être utilisé en production.
- Bootstrap accessibility plugin, (<http://paypal.github.io/bootstrap-accessibility-plugin/>)
Pour les utilisateurs de bootstrap, excellente implémentation par l'équipe de paypal, très active sur le domaine.
- iCITA (Illinois Center for Information Technology and Web Accessibility), (<http://test.cita.uiuc.edu/aria/>)
Une autre plateforme de démonstration de référence



L'accessibilité partout ! Retour d'expérience sous Debian

On connaît bien la grande frustration d'un enfant à qui l'on dit devant un manège : « tu ne peux pas y monter, tu es trop petit ». On peut imaginer la même frustration d'un adulte qui ne peut pas utiliser un ordinateur car il ne peut pas voir son écran, ou bien le texte est écrit trop petit, ou bien la souris est trop sensible,...



Samuel Thibault
Enseignant-Chercheur et
Développeur Debian

L'accessibilité du poste de travail est un problème

relativement connu, mais ne considérer que cette situation serait réducteur. Ces dernières années, la distribution GNU/Linux Debian, dont on parlera notamment ici, a pris en compte de plus en plus largement le souci d'accessibilité de l'ensemble du système qu'il propose, et ce, quel que soit l'usage ciblé de l'ordinateur sur lequel il tourne. Cela a bien sûr consisté à ajouter le support technique de l'accessibilité, mais a également nécessité de trouver des compromis, pour permettre l'usage d'un ordinateur par tous sans pour autant gêner les usages actuels.

Différentes situations, un même besoin d'accessibilité

En simplifiant grossièrement, on peut distinguer 4 situations différentes pour un ordinateur :

- Un poste public, un ordinateur en libre-service dans une bibliothèque ou un hôtel par exemple,
- Un poste privé, typiquement l'ordinateur familial,
- Un poste installé pour être utilisé par une personne handicapée, son poste de travail notamment,
- Un poste installé par une personne handicapée, son ordinateur personnel par exemple.

Dans le cas du poste privé, on en connaît a priori l'ensemble des usagers potentiels, et il y a peu de chances qu'il soit amené à être utilisé par un usager ayant un handicap. Ainsi, on ne pense pas forcément à y installer d'outils pour son accessibilité. Avec l'âge, les usagers sont cependant amenés à avoir au minimum besoin que le texte soit écrit plus gros.

Pour le poste public, par définition on ne connaît pas l'ensemble de ses usagers potentiels, et il faudrait donc prévoir qu'il puisse être utilisé par des personnes handicapées. Idéalement tous les supports

utiles pour compenser différents handicaps devraient être préinstallés et facilement activables. Il n'est pas envisageable, dans une bibliothèque, avant de pouvoir utiliser le poste pour faire une recherche bibliographique, d'avoir à attendre que l'administrateur système (qui n'est probablement pas présent en permanence) soit passé pour installer ou activer le support ! En ce qui concerne le poste de travail, il n'y a a priori qu'un seul usager, et l'on envisage donc naturellement de lui adapter (ou non) une bonne fois pour toutes ce poste. Il se peut cependant que cet usager veuille montrer un document à un collègue, ou le faire travailler dessus, et donc que le poste soit utilisé par une autre personne, qui est potentiellement n'importe qui dans l'entreprise ! Dans ces trois situations, on a donc à chaque fois un besoin potentiel plus ou moins probable de support d'accessibilité, avec plus ou moins de présence d'un administrateur système pour installer ou activer ce support ; et cet administrateur a plus ou moins de

compétences pour le faire... On comprend donc l'utilité que l'installation du support soit en fait systématique - la même dans les trois situations - et que son activation soit facile selon le besoin (à l'aide de raccourcis claviers par exemple). Le quatrième cas va plus loin : devoir réinstaller son ordinateur, cela arrive à tout le monde. Pour ceux qui en ont les compétences techniques, il est frustrant de ne pas pouvoir le faire soi-même simplement parce que le programme d'installation du système d'exploitation n'est pas accessible. Peu de systèmes ont un tel support.

Disponibilité du support de l'accessibilité

La première étape de support de l'accessibilité est bien sûr l'existence même et la disponibilité des logiciels fournissant ce support. On peut en trouver de nombreux sur Internet, pour les différents systèmes Windows, MacOS ou GNU/Linux. Leur disponibilité immédiate parmi les logiciels fournis avec le système fait cependant une grande différence : sans cela, un administrateur système devra fouiner sur Internet pour trouver les noms des logiciels à installer, sans forcément savoir les bons mots-clés (lecteur d'écran, plage braille, loupe, ...). On trouve ainsi typiquement des sections « accessibilité » dans les listes de composants installables intégrés à Windows, MacOS et les différentes distributions GNU/Linux. Le support intégré à Windows est cependant très pauvre, on doit par exemple installer à la main un lecteur d'écran comme NVDA. MacOS contient à l'inverse systématiquement un large support d'accessibilité, avec notamment le lecteur d'écran VoiceOver. En ce qui concerne GNU/Linux, les distributions proposent en général un large support d'accessibilité, avec notamment le lecteur d'écran Orca. Ce support est de plus en plus installé par défaut, notamment sur Ubuntu et Debian. On atteint ainsi l'objectif d'accessibilité des trois premières situations mentionnées plus haut : il suffit d'activer le support déjà disponible, ce qui va de la configuration de la taille du texte de toute l'interface, à une lecture vocale, en passant par des jeux de couleurs adaptés à différents handicaps visuels.

Installer soi-même le système

Pour un utilisateur un peu aguerri, installer un ordinateur est une opération assez banale, à condition de pouvoir la faire, bien sûr. Microsoft ne fournit malheureusement aucun support d'accessibilité pendant l'installation de Windows. Celle de MacOS, par contre, propose notamment depuis quelques années le lecteur d'écran VoiceOver, qu'il suffit d'activer pour

pouvoir réinstaller soi-même son Mac ! Pour différentes distributions GNU/Linux, grâce à leurs licences libres qui permettent de les modifier et de redistribuer les modifications, on trouve depuis des années des versions « accessibles » de leurs programmes d'installation, qui embarquent ainsi un lecteur d'écran et permettent donc d'installer soi-même GNU/Linux. Maintenir ces versions modifiées peut être cependant fastidieux, et elles ne sont en général pas disponibles directement depuis les sites principaux des distributions, il faut alors connaître le nom du site fournissant la version accessible. Depuis plusieurs années, les distributions Ubuntu et Debian ont donc intégré ce support de l'accessibilité dans leur programme d'installation officiel. Ainsi, si l'on branche un afficheur braille USB et que l'on boote le CD d'installation standard de Debian, celui-ci reconnaît automatiquement l'afficheur braille et démarre le lecteur d'écran, sans intervention de l'utilisateur !

L'accessibilité du système, une affaire de compromis

Comme on l'a vu dans tout ce qui précède, pour que le support de l'accessibilité soit effectif, il doit en quelque sorte s'imposer : être préinstallé, prêt à être activé, etc. Cela n'est pas forcément vu d'un bon œil, car cela peut risquer d'interférer avec un usage « standard ». Des compromis doivent donc être établis. L'espace disque nécessaire est en général raisonnable : pour le système Debian, cela représente moins de 100 Mo, et donc de l'ordre de 5% du système. Pour le programme d'installation, le support de la synthèse vocale, qui représente environ 8 Mo dans le programme d'installation, n'a pas été intégré aux images utilisées pour les systèmes embarqués; ceci parce que la question de place devient cruciale, et parce qu'on utilise en fait en général de toutes façons une console série plutôt qu'un écran. L'activation du support n'est pas non plus une question anodine. En effet, il n'est pas question que tous les systèmes Debian se mettent à jacasser dès leur installation ! Ainsi, c'est un raccourci clavier (documenté) qui permet par exemple d'activer la synthèse vocale. Cela ne fait cependant qu'introduire un nouveau problème : lors du démarrage du CD d'installation, à quel moment ce raccourci peut-il être frappé ? Devoir coller l'oreille au lecteur de CD pour déterminer s'il a terminé de charger le programme d'installation n'est pas vraiment raisonnable ! Le compromis qui a été trouvé est qu'un bip est émis lorsqu'il est temps de frapper le raccourci, gênant ainsi juste légèrement les installations « standard ».

Ce bip n'est par contre pas activé pour les versions de développement du programme d'installation, pour éviter à leurs développeurs des migraines à force d'entendre le bip à chaque test

L'accessibilité, un problème de pérennité

Comme évoqué précédemment, devoir maintenir une version « accessible » n'est pas une méthode pérenne. Au sein du projet Debian, nous avons donc intégré le support de l'accessibilité dans le processus même du développement du programme d'installation, et ce support est testé par différents développeurs avant de livrer une nouvelle version. La procédure est en effet documentée sur le wiki du projet (<https://wiki.debian.org/DebianInstaller/Accessibility>), et elle est réalisable sans avoir besoin de disposer de matériel spécifique, le logiciel de virtualisation QEMU/KVM proposant notamment un afficheur braille USB virtuel ! L'intégration réelle au sein du processus de développement apparaît également dans le choix des compromis. Dans l'exemple du bip au démarrage du CD d'installation de Debian, ce compromis a été discuté pendant quelques mois (!) avant d'être intégré. Cette intégration est cependant notablement pérenne : à chaque fois qu'il a été suggéré par un utilisateur de le supprimer, l'équipe de développement s'est souvenue de la discussion et a aussitôt rejeté elle-même la suggestion, de manière argumentée.

Encore plus loin

Dans les différentes situations énumérées au début de cet article, même lorsque le support d'accessibilité est disponible, il faut encore l'activer et surtout le configurer, car les handicaps étant très divers, leur compensation est la plus effective lorsque des réglages fins sont effectués (couleurs, délais, raccourcis, ...). C'est une étape qui peut être fastidieuse, surtout s'il faut ramener plus tard les réglages à leur valeurs utilisées par l'utilisateur régulier du poste ! Le projet GPII (<http://gpil.net/>) vise à automatiser cela. L'idée est de disposer d'un objet personnel que l'on branche au poste pour lui signifier que l'on va l'utiliser. Cela peut être par exemple une clé USB contenant les paramètres désirés. Le poste va alors automatiquement, lorsque la clé est branchée, activer les supports demandés, et amener temporairement les réglages aux valeurs voulues. Lorsque l'on débranche la clé, le système revient dans sa configuration par défaut. Une fois ce système disponible sur tout ordinateur, il ne restera plus qu'à résoudre l'accessibilité du port USB !



Pourquoi les lecteurs d'écran ont-ils besoin d'applications accessibles ?

L'accès au numérique est une véritable chance pour les personnes handicapées visuelles. Il leur donne la possibilité de lire des documents au format électronique plutôt que sur papier, sans compter le courrier électronique et la messagerie instantanée ou la vidéoconférence. Les démarches sur papier, impossibles à faire sans assistance, deviennent faisables, si elles se dématérialisent, en toute autonomie. Mais depuis que l'informatique est devenue graphique, elle comporte également de multiples défis.



Jean-Philippe MENGUAL
Association **accelibreinfo**
<http://accelibreinfo.eu>

Les défis posés par les interfaces graphiques pour les personnes aveugles

En effet, l'information à l'écran s'adresse à des personnes ayant une vision globale de l'environnement, chose impossible pour les personnes déficientes visuelles. Une page imprimée, tout comme un écran affichant une partie plus ou moins importante d'un document, communique d'abord un message global. D'un seul coup d'œil le lecteur peut comprendre l'organisation du contenu et aller directement à l'information essentielle. Faute de vision globale il faut la reconstruire par une lecture de l'ensemble du contenu en tenant compte des informations pertinentes sur la forme du document. C'est un processus beaucoup plus long, si bien qu'à la première lecture, la personne aveugle peut être moins efficace.

L'enjeu pour un utilisateur souhaitant accéder à un ordinateur s'exprime donc aussi bien en entrée qu'en sortie, c'est-à-dire qu'il doit pouvoir savoir où il se situe dans l'espace de travail, décider comment interagir avec celui-ci et obtenir le résultat rapidement. C'est pourquoi un logiciel qui se contenterait de numériser et de reconnaître les caractères affichés à l'écran ne serait pas assez opérationnel, notamment à cause de cet impératif d'interaction instantanée et de se situer dans l'espace.

Ces deux contraintes sont surmontées par la capacité d'une application à être utilisable au clavier, mais aussi par un outil qui est le lecteur d'écran.

Les solutions apportées par le lecteur d'écran

Il s'agit d'un logiciel qui transmet en braille ou vocalement le contenu et la nature de l'objet sur lequel se trouve le curseur d'écriture, ou

focus. Ces logiciels sont également capables de décomposer certains objets (par exemple, lire un texte ligne par ligne, mot à mot, caractère par caractère), au sein d'un objet fenêtre de document, case à cocher ou barre de menus. Ils ne savent en revanche pas interpréter des images même si elles contiennent du texte.

Pour ce faire, en particulier quand il s'agit d'objets, le lecteur d'écran lit l'étiquette associée à l'objet puis sa nature. Par exemple, dans une boîte de dialogue, quand la touche tabulation amène le curseur sur le bouton OK, le lecteur d'écran dira OK bouton, explicitant à l'utilisateur ce qui est écrit à l'écran, mais aussi de quel objet il s'agit et, par conséquent, comment il peut interagir avec.

Pour autant le lecteur d'écran n'est pas une intelligence artificielle et ne devine pas la nature et le contenu des objets où se trouve le focus. Il n'est qu'un cerveau captant de l'information, l'analysant et l'envoyant vers une synthèse vocale ou un protocole de communication d'une plage braille. Pourquoi n'arrive-t-il pas toujours à analyser les choses correctement ? Comment gère-t-il les situations où il est rigoureusement impossible d'interagir avec un environnement intégralement à l'aide du focus ?

Après avoir présenté le circuit conduisant un objet à être décrit par un lecteur d'écran, nous évoquerons comment il gère sa connaissance des objets pour offrir à l'utilisateur final la meilleure expérience.

De l'objet au lecteur d'écran : le circuit de l'information

Le lecteur d'écran s'inscrit dans une chaîne de circulation de l'information et dépend beaucoup des modes de communication et des éléments transmis au sein de ce circuit.

La plupart des programmes à interface élaborée s'appuient aujourd'hui sur des bases d'objets préexistantes, développées par une communauté ou par les concepteurs mêmes du logiciel : c'est le **toolkit**. Il contient un ensemble d'objets, les **widgets**, que le

programmeur peut paramétrer à sa convenance pour atteindre le résultat désiré. Puis, une autre partie du toolkit envoie le résultat vers une sortie et capte une entrée, elle gère les événements.

La sortie est le plus souvent une bibliothèque chargée de mettre en image l'information à l'écran. Mais certains toolkits comme GTK (GNU/Linux) ou GDI32 (Windows), ont été prévus pour envoyer de l'information permettant à une technologie d'assistance de disposer des éléments propres à aider l'utilisateur. On parle de bus d'accessibilité, AT-SPI sous GNU/Linux ou anciennement MSAA sous Windows.

Ce bus envoie l'information ainsi récoltée vers la technologie d'assistance, comme un lecteur d'écran (Orca sous GNU/Linux, NVDA ou Jaws sous Windows).

Ce dernier, par défaut, la transmet en l'état aux outils de sortie sonore, braille ou d'autres dispositifs d'assistance.

La technologie d'assistance est donc une plaque tournante dépendante de beaucoup d'éléments. D'abord, de la qualité du code qui la relie au bus d'accessibilité. Ensuite, de la qualité de la connexion entre le toolkit et le bus, ainsi que de la capacité du bus à bien interpréter l'événement qu'il doit gérer du toolkit. Enfin, de la qualité du code du widget qui, s'il lui transmet une information inexploitable, ne pourra qu'envoyer ce résultat défectueux à la technologie d'assistance.

Dans la pratique, se posent plusieurs difficultés dans ces différents niveaux de communication : il existe de nombreux toolkits (Qt, Java, Wxwidgets, etc), lesquels n'ont pas souvent prévu la possibilité d'envoyer de l'information au bus d'accessibilité. Ainsi, il arrive qu'aucune transmission au bus n'ait été prévue. Quand elle existe, encore faut-il qu'il puisse transmettre correctement les événements au lecteur d'écran. C'est par exemple pourquoi GTK n'est pas accessible sous Windows. Même dans les toolkits intégrant l'accessibilité, encore faut-il que l'objet utilisé soit standard. S'il ne l'est pas, il est indispensable de le

reconnecter à la bibliothèque générale de transmission au bus d'accessibilité. Ainsi, la barre d'outils surplombant la liste des messages dans Thunderbird est illisible au lecteur d'écran faute d'être un objet standard et connecté aux bus d'accessibilité des plateformes.

Enfin, même quand le circuit existe, le programmeur n'est pas exempté de tout effort, car, s'il veut un logiciel accessible, il a intérêt à montrer une certaine rigueur de programmation.

Programmer accessible

Par exemple, le développeur est invité à paramétrer ses widgets de telle sorte que l'étiquette et la nature du widget soient toujours transmises ensemble au bus d'accessibilité. Sans cela, le bus reçoit des informations incohérentes, par exemple une liste des étiquettes suivie d'une liste d'objets. C'est ce que connaissent un grand nombre d'applications, dont n'est pas exclue la suite Office par exemple. Pour que l'application soit accessible, JAWS ou NVDA ont dû prévoir des programmes, ou scripts, chargés de détecter l'arrivée d'informations issues des événements d'une application, et de les ordonnancer afin qu'avant leur traitement par le lecteur d'écran, elles soient intelligibles et permettent l'interaction avec les programmes. Si un tel effort est imaginable pour des applications populaires ou sous l'impulsion de communautés, il ne saurait être acceptable pour toutes les applications. Cette trajectoire a par exemple entraîné une réécriture complète d'Orca, lecteur d'écran sous GNU/Linux en 2012. Désormais, les développeurs tentent de sensibiliser les autres communautés de sorte qu'elles produisent des widgets exploitables par Orca plutôt qu'obliger le lecteur d'écran à les comprendre par une intervention humaine. Celle-ci serait d'autant plus fragile qu'à chaque mise à jour, elle pourrait être remise en cause, induisant des coûts de développement importants.

C'est donc souvent un environnement défavorable qui rend le lecteur d'écran inopérant ou lui complique la tâche dans beaucoup d'applications, cette réalité se retrouvant sur le Web comme sur l'ordinateur même. Mauvaise gestion de la circulation dans le bus d'accessibilité, absence de lien entre le toolkit et lui, paramétrage peu rigoureux des widgets, tels sont les éléments sur lesquels chacun peut agir pour améliorer l'accessibilité du numérique. S'appuyer sur des toolkits ou objets standards, paramétrer les widgets avec soin, penser à relier tout objet ou toolkit non

standard au bus d'accessibilité des systèmes visés, telles sont les pistes de réflexion élémentaires que peut suivre l'acteur du numérique; ceci autant en tant qu'auteur d'un contenant que d'un contenu s'il souhaite le rendre accessible à tous quelles que soient ses conditions d'accès à la plateforme. Ces pistes exigent moins une expertise de l'accessibilité qu'une démarche de programmation ouverte et responsable: ceci prémunissant chacun de coûteux développements d'interfaces dédiées à l'accessibilité, rompant avec l'universal design souhaitable, et risquant nécessairement d'être inadaptée à une partie des utilisateurs à cause de la diversité des modes d'accès actuels à l'information.

Ce simple effort peut rendre une application utilisable même si elle ne donne pas au curseur (focus) toute la latitude pour y naviguer et y activer des commandes.

Le lecteur d'écran a en effet de puissants moyens pour traiter ces situations.

Les modes de navigation du lecteur d'écran

Prenons l'exemple d'un terminal sous Windows ou GNU/Linux, ouvert depuis le bureau. Dans cet environnement, sauf à utiliser un outil tel qu'Emacs et ses buffers et son mode shell, le curseur (focus) est limité à la ligne de commande et son historique. Il ne peut à aucun moment parcourir la sortie d'une commande. Pour autant, l'objet concerné étant correctement transmis au lecteur d'écran, Orca comme JAWS permettent de le lire. Ils utilisent respectivement l'examen global et le curseur JAWS pour permettre à la personne de lire tout l'écran, au-delà des objets où se cantonne le curseur. Mieux encore, ce mode peut entraîner le déplacement du pointeur de souris, qui devient alors déplaçable au clavier et non plus seulement à l'aide de la souris. Le curseur JAWS, déplacé au clavier, parcourt ainsi ligne à ligne tout l'écran; l'examen global s'axant sur les objets qu'il contient, sauf dans une fenêtre de document qu'il parcourt ligne à ligne. Sur un objet, une touche du clavier permet de simuler un clic gauche ou droit, bref ou maintenu. De cette façon, si les objets sont inaccessibles au curseur, ils peuvent demeurer utilisables sous réserve que le programmeur en transmette les détails au lecteur d'écran de façon exploitable. VoiceOver et NVDA maximisent cet usage.

En effet, dans un environnement Apple, tout est objet. Le lecteur d'écran ne fait que se déplacer entre objets, les rangeant dans une logique qui lui est propre et n'offrant aucun autre moyen d'accès à l'environnement. Cette conception,

puissante dans les applications conçues de manière accessible, fragilise beaucoup l'efficacité de VoiceOver dans nombre d'applications où, précisément, l'accessibilité n'a pas été pensée pour Mac. C'est le cas des boîtes de dialogue de LibreOffice, de Firefox ou d'applications comme Alloresto auxquelles il est difficile d'accéder.

NVDA a repris cette logique mais uniquement sur demande de l'utilisateur. Tentant de combiner les techniques du curseur virtuel et de la navigation par objets, ce projet permet à l'utilisateur d'accéder à son écran avec le focus, la revue à plat ou par les objets. Dans le dernier cas, il crée une hiérarchie des objets basés sur la hiérarchie technique des éléments du système (bureau, barres sur le bureau, boutons sur les barres, applications correspondantes, objets dans l'application, contenu de chaque objet, etc). Cette solution est particulièrement commode sous Windows quand il s'agit d'amener le pointeur de souris à un endroit inaccessible.

NVDA exploite aussi les objets HTML, tout comme Orca et JAWS, en offrant aux utilisateurs la possibilité de les regrouper dans des boîtes de dialogue sous forme de listes. Cette pratique décomplexifie une page Web et regroupe les objets (liens, titres, cases à cocher, etc), avec leurs caractéristiques, facilitant la navigation en leur sein.

Le lecteur d'écran a donc des moyens importants pour assurer son rôle d'intermédiaire avec l'utilisateur. Mais sa qualité dépend beaucoup des informations qu'il réussit à récupérer. Si les concepteurs de ces logiciels s'efforcent de l'aider à comprendre ce qu'il reçoit, ils ne peuvent agir pour l'ensemble des applications informatiques existantes. La distribution Debian par exemple ne contient pas moins de 30 000 paquets. Il est difficile à un lecteur d'écran de s'adapter à la diversité et à l'évolution permanente des applications. Il semble plus optimal que chaque concepteur d'application soit sensibilisé à ces réalités techniques et qu'il recherche l'universalisme du circuit de circulation de l'information. De même il pourrait s'intégrer à des circuits existants par un code utilisant les standards et respectant les règles de design; à défaut, il pourrait créer un circuit qui leur est spécifique mais qui respecte ces passages obligés pour que l'information demeure toujours accessible. C'est sans doute dans ce sens que s'oriente la logique récente des référentiels d'accessibilité du Web ou les règles imposées aux applications candidates à l'entrée dans l'Apple Store.



Les outils pour être "accessible"

Le projet est terminé et il est temps de livrer, mais on vous pose la question qui peut tout stopper : « Le projet respecte-t-il les normes d'accessibilité ? »



Christophe Villeneuve
Consultant IT pour Neuros,
auteur du livre « **Drupal
avancé** » aux éditions
Eyrolles et auteur aux
Éditions ENI, Rédacteur
pour WebRIVER, membre

des Teams DrupalFR, AFUP, LeMug.fr
(MySQL/MariaDB User Group FR),
Drupagora, PHPTV...

Vous connaissez déjà la réponse, car cette tâche est passée à la trappe. Cependant, il existe de nombreuses solutions pour aider à vérifier si tout a été respecté.

LES ENVIRONNEMENTS

Le desktop

Les ordinateurs de bureau ou portables proposent différents logiciels ou extensions pour rendre « accessible » votre projet quel que soit le système d'exploitation que vous utilisez. Nous vous proposons une petite liste d'outils qui fonctionnent sur l'ensemble des systèmes (Windows, Mac, Linux) :

- Aide à la saisie et à la communication verbale : Chewing Word
- Aide à la communication : DonnerLaParole
- Système de pointage par reconnaissance de mouvement : eViacam
- Les navigateurs : Firefox (natif), Google Chrome avec Microsurf
- Extension vers OpenOffice / LibreOffice : odt2braille
- Aide au développement : aDesigner pour Eclipse, a11y pour Netbeans

Ces outils permettent de couvrir les points principaux d'un projet au niveau du développement, de la documentation, et de l'utilisation.

Pour utiliser les extensions ou logiciels listés, vous pouvez vous rendre sur les liens suivants pour consulter les documentations :

<http://scenari.etc.fr/capa/DOCS/SP4/Tuto/01/>
<http://www.technologie-handicap-accessibilite.net/feed/>

Le Web

Le Web est en constante mutation, tant au niveau du graphisme, que du langage. Toutefois pour vérifier si votre projet Web respecte bien les normes d'accessibilité pour être utilisé par un maximum d'internautes, vous pouvez utiliser les outils disponibles sous licence gratuite, commerciale ou en ligne. Ainsi, vous pouvez par vous-même évaluer,

réparer et vérifier si la conformité de votre projet existe bien. Aucun de ces outils ne peut toutefois tester l'intégralité des niveaux des référentiels, car ils ne remplaceront jamais l'intervention humaine. Ils peuvent cependant être très précieux et vont vous permettre d'effectuer des tests précis à tout moment. Tout d'abord, à toutes les étapes de validation vous lancez votre navigateur pour vérifier si les normes d'accessibilité sont bien respectées. Ces tests s'effectueront au niveau du langage HTML, CSS, des liens... Pour cela, vous utilisez une ou plusieurs extensions (barre d'outils) de votre navigateur favori :

- Firefox : Juicy Studio Accessibility Toolbar, Web Developer, total Validator
- Google Chrome : total Validator, Accessibility
- Internet Explorer : Web Accessibility Toolbar (WAT)

Bien entendu, il existe des extensions spécifiques comme la navigation au clavier, par le texte, les contrastes de couleurs, l'audio, etc. Ensuite, vous pouvez valider le niveau de l'accessibilité qui vous est demandée directement en ligne, c'est à dire que les tests d'accessibilité s'effectuent à distance tout en respectant les référentiels :

- WCAG avec Achecker ou le validator de W3C
- AccessiWeb avec Tanaguru AXS

Ces différentes extensions vont vous fournir des informations pour vous mettre aux normes.

Exemple

Si votre image se décompose de 3 rectangles et si vous proposez le code suivant :

```

```

Les outils vont vous répondre : "An img element must have an alt attribute, except under certain conditions. For details, consult guidance on providing text alternatives for images."

C'est-à-dire que le tag ALT est absent dans la balise image. Pour cela, la solution se présente de la manière suivante

```

```



Bien entendu, afin de vous aider chaque balise est vérifiée et contrôlée avec un maximum d'informations.

Pour aller plus loin :

<http://www.webaccessibilite.fr/liens-utilites-accessibilite-outils-de-test-et-de-verification.php>
<http://achecker.ca/checker/index.php>
<http://www.w3.org/WAI/>

Le mobile

Depuis l'apparition des téléphones mobiles et jusqu'à la prise de conscience de la nécessité de l'accessibilité par les opérateurs (il y a 10 ans), le portage d'un projet logiciel/Web accessible sur smartphones n'était pas évident, car il consistait à produire des logiciels et des appareils spécifiques (par exemple spécifiquement à destination des utilisateurs déficients visuels). Cela se traduisait par de grosses touches pour le clavier, un graphisme complètement refait pour laisser apparaître de très gros contrastes et des menus adaptés pour l'appareil. Cependant depuis ces dernières années, les éditeurs (Apple, Android, Firefox OS, Windows Phone, etc.) ont effectué un gros effort pour fournir des applications externes à leurs appareils afin de répondre aux différentes attentes de l'accessibilité, quelle que soit la licence (propriétaire, open source, libre). Ces solutions d'assistance proposent trois fonctions :

- L'agrandissement des caractères sur le principe de la loupe,
- La vocalisation des caractères à partir de lecteurs d'écrans,
- La conversion des caractères en braille.

Ainsi, l'utilisateur peut zoomer, modifier les couleurs d'affichage et la mise en évidence des zones d'information.

Grâce à tout cela, la communication entre son smartphone et un afficheur Braille a été rendue possible par Bluetooth.

Attention, moins de 20% des critères d'accessibilité sont automatisables. Les outils sont utiles, mais insuffisants. Et si on vous pose la question de l'accessibilité en fin de projet, sachez qu'il vous sera sans doute difficile de réellement prendre en compte tout ce à quoi il aurait fallu penser. Nous vous donnons des astuces, mais plus l'accessibilité est prise en amont, plus vous avez de chances de réussir votre projet pour toutes et tous, facilement.

MISE EN AVANT Tanaguru

Tanaguru est un logiciel libre d'automatisation d'accessibilité, dont le code source est disponible sur Github, avec la possibilité de le tester directement en ligne.

Il va pour vous effectuer 167 tests d'accessibilité par rapport aux différents référentiels en vigueur (RGAA, AccessiWeb, WCAG). Ainsi, il est possible d'utiliser ce logiciel de différentes manières :

- Vérifier les fichiers en mode hors connexion,
- Auditer une page web ou un site complet,
- Vérifier les scénarios.

Suivant votre choix d'utilisation, vous saisissez l'adresse du site en local ou sur Internet et vous obtenez un résultat sous la forme d'un histogramme et d'un camembert **Fig.1 et 2**.

Le résultat fournit le détail des différentes

catégories de référentiels (images, cadres, couleurs, multimédia, script, formulaires, etc.).

Chaque point est détaillé et vous aide à corriger le problème **Fig.3**.

URL :

<http://www.tanaguru.com/fr/>

Tanaguru contrast-finder

Il s'agit d'un autre outil indispensable de la même

famille, mais il va s'intéresser exclusivement au contraste des couleurs, qui est indispensable pour une utilisation sur smartphones, tablettes et Web.

Il est disponible en ligne ou sous la forme d'extension pour les navigateurs comme Firefox

Son but est de permettre de gagner du temps dans le choix des couleurs, tout en respectant les critères d'accessibilité au niveau de contraste minimum entre le fond et le texte. L'utilisation est facile, car vous saisissez la couleur en hexadécimal de la couleur de fond et de la couleur du texte et son niveau d'accessibilité :

AA : ratio 4.5

AAA : ratio 7 **Fig.4**.

L'application proposera un résultat d'un choix au niveau de la palette de couleurs pour être cohérente avec les différentes normes. De plus le code source est proposé au format hsl, rgb et hexadécimal **Fig.5**.

URL : <http://contrast-finder.tanaguru.com/?lang=fr>

WCAG Contrast Checker

Il s'agit d'une extension pour Firefox pour le référentiel WCAG, car il va s'intéresser aux contrastes des différentes zones de votre page web et plus précisément au niveau des blocs et des mises en avant de contenus.

Pour l'utiliser, vous activez l'extension (clic droit) et vous obtenez le résultat de suite avec les bons et mauvais résultats **Fig.6**.

Chaque position est cliquable et vous donne l'emplacement exact correspondant à la page **Fig.7**.

Le point que vous avez sélectionné est entouré par un trait de bordure, vous permettant de voir de suite la correspondance.

HeadingsMap

Pour mesurer la hiérarchie des titres et une navigation par le clavier (avec la touche TAB), vous pouvez utiliser l'extension HeadingsMap disponible sous Firefox.

L'intérêt d'avoir cet outil est une simulation humaine, c'est-à-dire que vous naviguez par le clavier sans toucher à votre souris.

Le résultat de l'extension se présente de la manière suivante : **Fig.8**.

Les erreurs sont mises en avant en pointant exactement l'emplacement de votre page Web.

En résumé

Comme vous pouvez le voir, il existe un large choix d'outils quels que soient votre navigateur et votre système d'exploitation. C'est pourquoi il est indispensable de ne pas oublier cette étape dans la réalisation de votre projet.

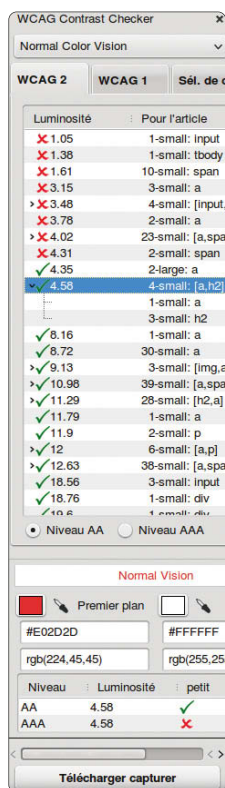


Fig.6

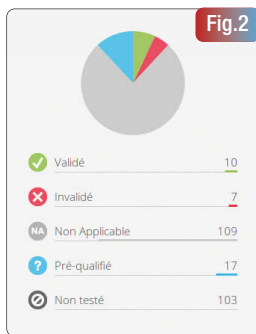


Fig.2



Fig.3

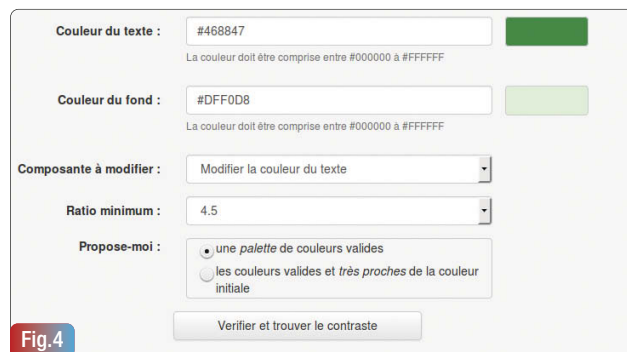


Fig.4



Fig.8

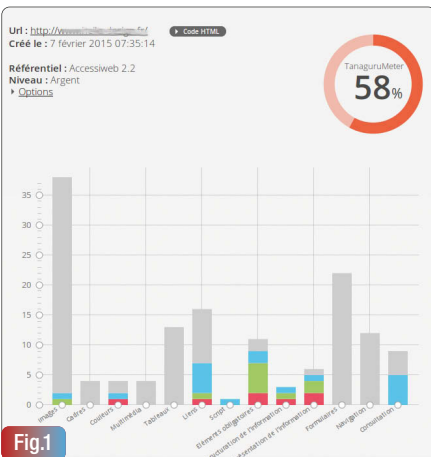


Fig.1



Fig.5

La boîte à outils idéale du développeur Android

Depuis son lancement il y a 8 ans, la plateforme Android a connu d'importants changements visant à améliorer l'expérience utilisateur et à proposer un système au design épuré. Parallèlement à ces évolutions, le développement d'applications Android n'a lui cessé de se complexifier, confrontant les développeurs à des challenges toujours plus nombreux. Dans ce contexte, il est devenu indispensable de posséder une boîte à outils performante afin de réaliser des applications optimisées s'exécutant sur un maximum de terminaux. Passage en revue des outils indispensables au développeur Android.



Sylvain SAUREL – Ingénieur d'Etudes Java / Android
sylvain.saurel@gmail.com – www.all4android.net

Comparé à juste titre à un artisan, le développeur informatique ne peut réussir sans une bonne trousse à outils à sa disposition. Constituée au fil du temps, elle doit lui permettre de gagner en productivité lors de ses développements. Dans cet article, nous nous intéressons plus particulièrement à la boîte à outils idéale pour le développeur Android. Pour passer en revue ces outils, nous allons utiliser une approche basée sur les différentes phases rencontrées lors du développement d'une application Android. Nous partirons ainsi de la phase de conception pour terminer par la publication sur un store, en listant pour chacune des phases les outils de référence. Bien entendu, cette liste est évolutive et il conviendra à chaque développeur de la mettre au jour au fil des évolutions de la plateforme via un travail de veille technologique régulier.

Conception

Au commencement d'une application Android, il y a avant tout une idée ou un concept. La première étape consiste à transformer cette idée en une vision applicative. Pour ce faire, rien de mieux que de prototyper les différents écrans envisagés. Parmi les outils existants sur le marché, Pencil est une solution open source légère et performante. Disponible à l'adresse suivante : <http://pencil.evolus.vn/>, Pencil est extensible via un système de templates ce qui le rend également utilisable pour du prototypage iOS par exemple. Un exemple d'écran prototypé avec Pencil est présenté à la figure 1.

Au niveau commercial, des solutions comme Balsamiq ou Fluid UI peuvent également se révéler pertinentes suivant le budget que l'on souhaite y consacrer. Une fois les écrans prototypés, il est temps de valider le concept et d'aller plus loin dans la réalisation, en passant à la phase de conception détaillée dans laquelle seront décrites les interactions sur chaque écran.



Fig.2 Preview rendu écrans multiples

Réalisation IDE

Sorti en version 1.0 fin 2014, Android Studio est désormais l'IDE recommandé pour le développement sous Android. Google a ainsi mis fin aux évolutions du plugin ADT pour Eclipse finalement abandonné au profit d'une solution pleinement orientée Android. Basé sur la version community d'IntelliJ IDEA, Android Studio est disponible gratuitement à l'adresse suivante : <http://developer.android.com/sdk/index.html>.

L'IDE s'appuie sur une nouvelle structure pour les projets tout en se basant nativement sur le nouveau système de build de la plateforme. Un éditeur IHM plus évolué est proposé permettant la prévisualisation du rendu des layouts sur des écrans multiples (figure 2). L'éditeur de traduction proposé en standard est également un must apportant un gain de productivité pour la création des différentes traductions d'une application. Les équipes de Google ont également eu la bonne idée de proposer un wizard facilitant l'import de projets Eclipse sous Android Studio avec une détection des dépendances externes. Celles-ci seront automatiquement ajoutées au sein du build Gradle du projet nouvellement importé. Les aficionados d'Eclipse devront s'y faire, l'avenir appartient désormais à Android Studio.

Il reste néanmoins possible d'utiliser Eclipse ADT pour le développement sous Android tout en profitant du nouveau système de build grâce au plugin Gradle pour Eclipse. Néanmoins, ADT n'étant plus maintenu, cela pourra rapidement s'avérer pénalisant.

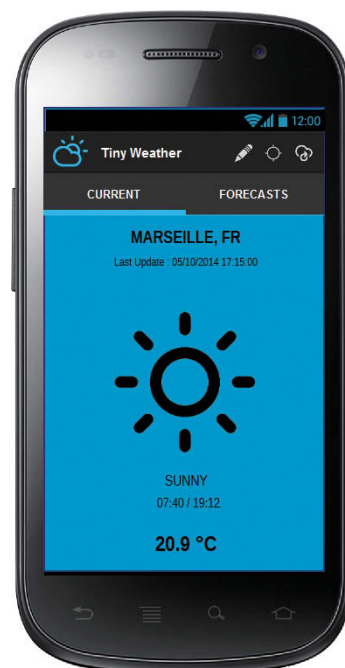


Fig.1 Ecran prototypé avec Pencil

Emulateur

En standard, le SDK propose un émulateur basé sur le logiciel libre QEMU. Il permet aux développeurs de créer des terminaux virtuels en spécifiant finement leurs caractéristiques matérielles (taille d'écran, résolution, densité de pixels, présence d'un GPS, ...) et logicielles (version d'Android). Parfaitement fonctionnel, cet émulateur présente de gros soucis de performances tant au démarrage qu'à l'utilisation. Impossible dans ces conditions de l'utiliser pour tester une application en conditions quasi réelles. Une solution pour accélérer cet émulateur existe pour les ordinateurs basés sur des processeurs Intel supportant la virtualisation. Intel fournit ainsi un outil nommé HAXM récupérable via le SDK Manager d'Android. Une fois l'outil installé, il permet de booster les performances de l'émulateur à condition d'utiliser une image Android x86 fournie par Intel. Une autre solution plus pérenne vers laquelle se

tourment de plus en plus de développeurs est de recourir à l'émulateur Android proposé par l'éditeur Genymotion. Gratuit pour des cas d'utilisation basiques dans un cadre personnel, il supporte les versions Android 2.3 à 5.0. Genymotion propose également une version professionnelle plus complète, offrant un support du multi-touch, de l'accéléromètre, du screencasting ou encore une API Java permettant son contrôle. Les performances sont réellement étourdissantes comme le montrent la comparaison des temps de démarrage et de déploiement entre Genymotion et un terminal physique (figure 3).

Frameworks

La plateforme Android propose un cadre de développement relativement libre. De fait, de nombreux frameworks ont vu le jour pour aider les développeurs à mettre en place de bonnes pratiques d'architecture structurantes pour leurs projets Android. Injection de dépendances, gestion simplifiée des appels réseaux, bus événementiel ou gestion de bases de données SQLite via du mapping objet / relationnel, les frameworks libres sont légion; il devient rapidement difficile de faire le bon choix devant tant de possibilités. Le tableau qui suit vient lister un certain nombre des frameworks les plus populaires pour adresser ces différentes problématiques.

Framework	Description
Retrofit	Client REST type-safe pour Android en particulier et Java en général. Basé sur des annotations, il facilite le requêtage de services Web.
Dagger	Solution d'injection de dépendances complète pour Android, Dagger se base sur la JSR-330 et permet de créer facilement des modules interchangeables et réutilisables.
Butter Knife	Buffer Knife est une solution d'injection dédiée à Android au périmètre plus restreint que Dagger puisque ciblant uniquement les vues. Les annotations sont interprétées à la compilation afin de produire le code Boilerplate directement, ce qui évite de recourir à de la réflexion à l'exécution.
RoboGuice	Implémentation de Guice pour Android, RoboGuice est une solution d'injection de dépendances complète recourant également massivement à des annotations, ce qui simplifie grandement le code. Néanmoins, les annotations sont traitées à l'exécution ce qui peut s'avérer pénalisant sur les anciennes versions d'Android.
Event Bus	Bus événementiel simplifiant la communication entre activités, fragments, threads ou encore services. Il permet de mettre en place une architecture de qualité pour une application Android tout en écrivant moins de code, bien que n'étant pas basé sur des annotations.
Otto	Bus événementiel conçu spécifiquement pour découpler les différentes parties d'une application tout en facilitant la mise en place d'une communication efficace entre celles-ci. Otto est un fork de Guava auquel des fonctionnalités ont été ajoutées pour le spécialiser pour la plateforme Android.
Green DAO	ORM conçu spécialement pour interagir avec la base de données embarquée SQLite sous Android. Léger et performant, il est bien adapté au monde Android avec une consommation mémoire minime.
OrmLite	ORM supportant un grand nombre de bases de données relationnelles, OrmLite propose un support de SQLite sur Android. Néanmoins, il souffre en termes de performance de n'être pas dédié uniquement à Android.
Volley	Bibliothèque HTTP facilitant la gestion des appels réseaux, Volley est intégré au sein du projet open source Android.
Picasso	Bibliothèque puissante pour le téléchargement et la mise en cache automatique d'images via des appels réseaux.
libGDX	Moteur de jeu 2D / 3D léger, facile à prendre en main et particulièrement bien documenté.

Ces différents frameworks peuvent être utilisés conjointement afin de se créer une stack complète sous Android. On peut ainsi imaginer utiliser Dagger pour l'injection de dépendances avec la solution de bus événementielle Otto pour faciliter un couplage faible. Retrofit étant ajouté en cas de besoin d'un client REST. De même, Green DAO serait

intéressant au cas où il serait nécessaire de requêter une base SQLite. Le gain apporté par ces frameworks étant double : faciliter le développement Android, tout en proposant des solutions performantes pour répondre à des problématiques courantes.

Composants

Moins structurants que les frameworks, les composants dédiés à Android sont également légion. L'émergence de plateformes telles que GitHub y est pour beaucoup. Prêts à l'emploi, ils permettent de réaliser des applications Android en tenant compte des bonnes pratiques tout en évitant de réinventer la roue. Si le développeur ne devait retenir qu'un seul outil, il s'agirait de l'Android Compatibility Library de Google. Elle permet une rétro compatibilité de composants Android standards actuels avec des versions antérieures du SDK. Ainsi, la V7 donne accès au développeur à des composants créés spécifiquement pour Android 5.0 tels que la RecyclerView, le composant CardView, la Palette ou encore le GridLayout. La V4 quant à elle, donnait notamment accès aux Fragments et Loaders. Bref, cette bibliothèque est indispensable à chaque projet Android désirant obtenir une compatibilité maximum.

Le tableau suivant présente les composants les plus populaires du moment :

Composant	Description
Jackson	Bibliothèque de manipulation JSON légère et ultra performante qui se trouve particulièrement bien adaptée à l'univers Android.
GSON	Pendant de Jackson poussé par Google, GSON offre les mêmes avantages en termes de concision de code, mais souffre quelque peu en termes de performances face à son concurrent.
Admob	Monétisation d'une application via le réseau publicitaire Admob. La gestion de la médiation en standard, de même que l'intégration avec Google Analytics, sont des plus indéniables.
Flurry	Solution donnant la possibilité au développeur d'obtenir des métriques poussées sur l'utilisation d'une application.
ACRA	Composant permettant d'obtenir un reporting précis des plantages et bugs d'une application et supportant l'interfaçage avec différents backends (Google Spreadsheet, Solution côté serveur, ...).
Pull To Refresh	Composant réutilisable de Pull To Refresh adaptable à différents types de vues : ListView, GridView, WebView, ViewPager, HorizontalScrollView, ...
SlidingMenu	Mise en place d'un menu glissant depuis un côté de l'écran à l'image de ce qui se fait sur l'application Google+.
ViewPagerIndicator	Vient compléter le ViewPager standard du SDK Android en fournissant un indicateur de position à l'image de ce qui existe sur l'application du Google Play Store.
Zxing	Composant de génération et de lecture de code-barres 1D et 2D.
Crouton	Améliore grandement les notifications standards en permettant de les adapter au contexte d'affichage.
MPAndroidChart	Les solutions pour réaliser des graphiques sont nombreuses mais MPAndroidChart l'emporte à nos yeux par sa simplicité d'emploi et la qualité des graphiques produits. Tous les types de graphiques sont supportés (lignes, barres, régions, camemberts, radars, ...) et ils sont également animables.
Material Dialog	Composant simple d'utilisation et grandement configurable permettant l'emploi de dialogs avec le thème Material Design quelle que soit la version d'Android utilisée.
Material Design Lib	Bibliothèque offrant l'accès au développeur à tous les widgets graphiques introduits avec Android 5.0 sur les versions antérieures à partir de la 2.2 d'Android.

Loin d'être exhaustive, cette liste de composants sera à compléter au fil du temps. Certains sites facilitent le travail de veille en proposant un regroupement par catégories des meilleurs composants libres pour Android. Le plus complet d'entre eux à l'heure actuelle étant The Android Arsenal disponible à l'adresse suivante : <https://android-arsenal.com> (figure 4).

Build

Avec la sortie d'Android Studio 1.0, un nouveau système de build dédié à Android est désormais proposé. L'Android Build System consiste en un plugin Android pour Gradle offrant flexibilité et extensibilité pour la gestion d'un projet Android. Les dépendances sont définies directement au sein d'un fichier de build Gradle. En sus, ce système offre l'avantage de découpler le build d'un projet de l'IDE, le rendant ainsi facilement reproductible. Android Build System va faciliter l'industrialisation des développements Android, et ce d'autant plus que Google l'a doté de quelques fonctionnalités supplémentaires telles que : le support des Builds Types pour générer différents APKs entre les version debug / release, le support des Product Flavor pour générer des APKs différenciant une version gratuite d'une pro, la gestion des APKs multiples, ou encore le support multi-dex. Le build est décrit via un DSL Groovy spécifique facilitant sa configuration. Du fait de l'extensibilité de Gradle, un grand nombre de plugins ont vu le jour pour répondre à des problématiques récurrentes lors du développement d'applications Android. Parmi les plugins essentiels on citera les 2 suivants :

- Android SDK Manager Plugin permettant de gérer les versions du SDK et de ses dépendances installées avant de lancer le build Gradle.
- Gradle Play Publisher qui s'appuie sur les API proposées par Google pour interagir avec la console Admin du Google Play Store. Le plugin permet d'uploader automatiquement un APK généré ainsi que les descriptions liées à l'application.

Test

La mise en place de tests est une étape cruciale dans le cycle de développement d'une application Android. De type unitaire ou fonctionnel, les tests vont permettre de s'assurer du bon comportement de l'application au fur et à mesure des mises à jour. Pour les tests unitaires, le SDK propose des classes d'instrumentalisation qui nécessitent un émulateur Android pour s'exécuter, ce qui restreint fortement toute pratique de type TDD. Le framework Robolectric vient corriger ce problème en permettant l'exécution de tests directement au sein d'une JVM ouvrant du même coup la voie au TDD pour le développement Android. Au niveau des tests fonctionnels, Robotium permet la mise en place de tests automatisés s'exécutant sur un terminal Android virtuel ou physique. L'API est simple d'accès et les possibilités sont sans limite. La société derrière Robotium propose d'ailleurs un outil d'enregistrement de tests automatisés s'interfaçant avec Android Studio. Bien que payant, l'outil peut être intéressant dans le cadre de gros projets nécessitant de grosses campagnes de tests. Toujours au rayon des tests automatisés, il est bon de signaler Espresso développé par Google qui est un projet prometteur facilitant les tests fonctionnels de type boîte noire via une API simple et élégante d'instrumentalisation. Avec le nouveau système de build Android, il existe désormais des plugins Gradle pour exécuter les tests associés à un projet. Ils seront

	Genymotion	Physical device
Deploy a 30 MB App	7s	21s
Start a device	15s	40s

Fig.3

Comparaison
Genymotion /
Terminal
physique

lancés à chaque build et vont parfaitement s'intégrer dans une démarche d'intégration continue efficace.

Publication

Étape cruciale pour une application Android, la publication représente bien souvent un travail fastidieux. L'upload de l'APK, des descriptions, des ressources graphiques ou encore des screenshots représentent des tâches chronophages qu'il est bon d'automatiser au maximum. En ce sens, le plugin de publication Gradle Plugin Publisher décrit précédemment, s'avère intéressant. Certaines démarches permettant d'associer tests automatisés et prise de screenshots automatisée sont également en train d'apparaître. Dans ce contexte, une approche intéressante a récemment été dévoilée par un développeur français sur son blog : http://flavienlaurent.com/blog/2014/12/05/screenshot_automation/. Concrètement, il propose de s'appuyer sur la puissance du framework UI Automator et de son API fournie par Google dans le SDK Android pour instrumentaliser des actions sur un périphérique Android avant de prendre ensuite les screenshots désirés à la volée, via la méthode takeScreenshot de l'API. Associée au build d'une application, cette démarche permet d'obtenir des screenshots mis à jour à chaque lancement de build et se révèle parfaite dans un processus d'intégration continue. On peut même facilement aller plus loin en automatisant l'upload de ces screenshots sur la console Admin du Google Play Store via l'API fournie par Google.

Conclusion

On le constate avec cette liste des outils indispensables pour le développeur Android, la tendance est clairement à l'industrialisation des développements. Mise en place d'un système de build performant, automatisation des tests fonctionnels, et publication automatique des informations associées à la console Admin du Google Play Store sont autant de pratiques qui vont permettre d'augmenter grandement la productivité des développeurs. Le nouvel IDE Android Studio est également un signal fort envoyé par Google aux développeurs Android, puisqu'il met à leur disposition un outil qui, à terme, rivalisera avec les IDE des plateformes mobiles rivales. Enfin, grâce à l'ouverture d'Android et à l'importante communauté de développeurs, les composants ou frameworks libres permettant de réaliser des applications de qualité n'ont jamais été aussi nombreux. Charge aux développeurs de faire les bons choix.



Android Arsenal

Feb 2, 2015

Android Wizard
Wizards

New Free

Wizard encapsulates navigation operations between fragments of an activity using **BackStack**.

visible scrollbar. The two most common modes are popup and indicator. Works seamlessly with **ListView** and **ExpandableListView**.

Feb 2, 2015

gradle-proguard-plugin
Gradle Plugins

New Free

The **gradle** plugin to add **proguard** snippets to your **proguard** setting file quickly.

This plugin pulls the proguard information from the **krschultz/android-proguard-snippets**.

Fig.4

Liste de
composants
d'Android
Arsenal

My JavaScript is beautiful !

JS

Vous allez aimer JavaScript ! Et vous n'allez pas avoir le choix. JavaScript (JS) a longtemps été le mal aimé des langages Web et à juste raison : complexe, lourd, pas toujours performant. Depuis quelques années, de gros efforts ont été réalisés sur les outils et les nombreux frameworks « simplifient » le développement JS.

Fondamentalement, JS reste un langage hardcore et pas toujours facile, surtout, pour les développeurs débutants. Mais la force de JS vient de sa communauté et des nombreux outils. D'autre part, l'arrivée de serveurs JS (Node.JS, Wakanda) a bouleversé la situation, et la possibilité de faire du Node.JS sur des objets connectés va sans doute ouvrir d'autres perspectives... Nous vous proposons un mini-dossier autour de JS : comment choisir un framework JS, JS en mode Jedi, et, enfin, nous terminerons notre voyage dans le merveilleux monde du fonctionnel...

```
{template.Redaction}
La redaction!
{/template}
```

Bien choisir son Framework JavaScript : AngularJS, Knockout, Ember, Backbone.js

Choisir un Framework JavaScript n'est jamais chose facile, ce choix impactera le temps de chargement, la vitesse de développement et la maintenabilité du code source de votre site Web. Il en existe des dizaines, dans cet article nous allons nous restreindre aux 4 plus connus.



Feltz Ludovic, Consultant **SoftFluent**
ludovic.feltz@softfluent.com

Plusieurs critères sont à prendre en compte dans le choix d'un framework :

- Le poids définira le temps de chargement des pages. Critère important pour les applications mobiles,
- La clarté de la documentation permet de réduire le temps d'apprentissage (lors de l'arrivée d'une nouvelle personne dans l'équipe par exemple),
- La facilité d'utilisation simplifie et accélère le développement,
- Une communauté active garantit l'évolution du Framework et une meilleure aide en cas de problèmes.

Nous allons commencer par un bref historique, puis nous allons plonger dans le vif du sujet avec un exemple simple que nous allons reproduire en utilisant chacun de ces Frameworks. Nous n'aborderons pas la communication avec le serveur et la sauvegarde dans une base de données.

POURQUOI UTILISER UN FRAMEWORK JAVASCRIPT ?

Les Frameworks JavaScript offrent des fonctionnalités prédéfinies permettant aux développeurs de ne pas avoir à manipuler le DOM

(Document Object Model) directement depuis leur code. Ils offrent donc un niveau d'abstraction et facilitent le développement d'applications Web. La plupart des Frameworks étudiés offrent la possibilité de développer des applications sur une seule page (Single Page Application) ce qui permet d'offrir une expérience plus fluide à l'utilisateur en évitant de recharger la page à chaque action. Le fonctionnement se rapproche alors d'une application de bureau. Les ressources sont chargées dynamiquement et ajoutées à la page quand c'est nécessaire, généralement en réponse à une action de l'utilisateur **Fig.1**. L'utilisation d'un Framework offre aussi un aspect structurant à l'application en proposant l'utilisation d'un pattern MVC, MVVM ou MVP. Un développeur arrivant sur le projet n'aura donc pas à se demander pourquoi et comment vous avez conçu telle ou telle partie de l'application.

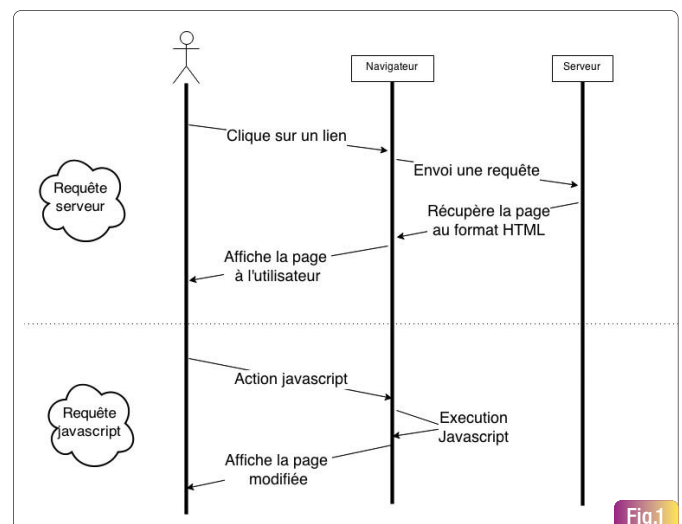


Fig.1

UN PEU D'HISTOIRE POUR COMMENCER

AngularJS: Apparu en 2009 sous le nom de GetAngular, il est utilisé par Misko Hevery, un des ingénieurs qui l'a développé pour recréer une application Web qui représentait plus de 17 000 lignes de code. En 3 semaines il est parvenu à réduire ce nombre à seulement 1 000 lignes, ce qui a convaincu Google de sponsoriser ce projet et ainsi créer sa renommée. Son but est de simplifier le développement et les tests de sites Web. Il utilise le pattern MVC en ajoutant du vocabulaire au code HTML de votre application.

Knockout : Né en 2010 et maintenu comme un projet Open Source par Steve Sanderson, employé chez Microsoft. Il simplifie l'utilisation du JavaScript et utilise le pattern MVVM.

Ember : Développé en 2007 par Sproutlt puis par Apple, il est finalement forké en 2007 par Yehuda Katz, un des principaux contributeurs de JQuery et Ruby. Il est utilisé par des grands noms tels que Yahoo, Groupon et ZenDesk. Il est basé sur le pattern MVC. Son avantage est de permettre le développement d'applications SPA (Single Page Application) grâce à son système de route.

Backbone : Créé en 2010 par Jeremy Ashkenas qui a aussi participé au développement de Coffee Script, il est très léger, ce qui lui a permis de se faire un nom parmi les autres Frameworks JavaScript. Il repose principalement sur le pattern MVP.

MISE EN PRATIQUE

Dans le but de comparer les différents Frameworks, nous allons créer une application simple en utilisant chacun des Frameworks présentés précédemment **Fig.2**. Cette application permet de faire du CRUD sur une liste de clients. La colonne Output affiche la valeur du modèle côté client (JavaScript). Pour peupler nos modèles nous allons à chaque fois réutiliser le même jeu de données. Comme nous n'allons pas aborder la connectivité avec une base de données nous allons utiliser de simples tableaux JavaScript :

```
var Genders = [
  {
    id: 1,
    key: "Mr",
    value: "Monsieur"
  },
  {
    id: 2,
    key: "Mme",
    value: "Madame"
  }
]
```

```
var ClientsList = [
  {
    id: 1,
```

Gender	Name	Age	Output
Mme	Jean	25	Madame: Jean (25 ans)
Mr	Juliette	30	Madame: Juliette (30 ans)
Mr	François	60	Monsieur: François (60 ans)
Mme	Anne	80	Madame: Anne (80 ans)

Ajouter un client

Fig.2

```
gender: Genders[0],
name: "Jean",
age: 25,
},
{
  id: 2,
  gender: Genders[1],
  name: "Juliette",
  age: 30,
}
]
```

AngularJS

AngularJS utilise le pattern MVC. Nous allons rapidement décrire le fonctionnement de ce pattern puis nous allons créer notre application.

Model : le modèle décrit la logique métier de notre application. Il contient les données ainsi que les méthodes de manipulation de celles-ci.

View : la vue est la façon d'afficher les données de notre application, c'est le code HTML qui est en charge de formater les données du modèle et d'envoyer les événements, tels qu'un clic sur un bouton, au Controller.

Controller : il reçoit les informations de l'utilisateur provenant de la vue, les traite et les envoie au modèle. Il renvoie ensuite les informations à la vue pour les afficher. Il agit comme le coordinateur entre le modèle et la vue.

Initialisation

Commençons par créer une application AngularJS très simple. Voici le contenu de la page HTML (index.html) :

```
<html ng-app="myApp">
<head>
  <title>AngularJS</title>

  <!-- Angular 1.3.0 (aucune dépendance) -->
  <script src="/script/angular.js"></script>

  <!-- Notre controller -->
  <script src="/script/controllers.js"></script>

</head>
<body>
  <div class="container" ng-controller="AppController as controller">
    <!--
                                Corps de la vue
    -->
  </div>
</body>
```

Et le code JavaScript (script/controllers.js) :

```
var app = angular.module('myApp', []);
app.controller('AppController', ['$scope', function ($scope) {
  /*
   * Corps du controller
   */
}]);
```

AngularJS définit ses propres attributs HTML, commençant par « ng- ». Angular permet également de définir des balises comme nous le verrons par la suite. Le premier attribut se trouve directement dans la balise HTML ouvrante (ng-app), il permet de définir la racine de notre application

donnant ainsi la possibilité au développeur d'utiliser toute la page ou seulement une portion de celle-ci.

Le deuxième attribut que l'on voit apparaître est **ng-controller**. Il fournit le contexte du **controller** et permet de lier notre vue avec notre modèle.

Affichage des données

Nous allons peupler notre modèle avec le jeu de données décrit précédemment et nous ajouterons une directive afin d'avoir un formatage du texte. Pour cela ajoutons simplement les variables **clients**, **genders** et une **directive** à notre Controller de la façon suivante:

```
app.controller("AppController", function ($scope) {
    $scope.clients = ClientsList;
    $scope.genders = Genders;
})
.directive('clientFormatted', function ()
    return {
        template: '{{client.gender.value}}: {{client.name}} ({{client.age}} ans)'
    };
});
```

Enfin modifions notre vue afin d'afficher tous les clients dans un tableau, le corps de notre page devient donc :

>> voir code complet (lien à la fin de l'article).

On voit apparaître ici 4 nouvelles balises appartenant à AngularJS. **ng-repeat** s'apparente à un « **foreach** ». Pour chaque client de notre **controller**, AngularJS crée une balise **<tr>** dont le titre sera l'identifiant de ce client. Les accolades : **{{expression}}** permettent à AngularJS de savoir quels blocs de codes interpréter, et d'effectuer une liaison avec le modèle.

ng-model permet d'effectuer une liaison dans les deux sens (two-way binding) avec notre modèle. C'est-à-dire qu'il permet de récupérer la valeur du modèle et que dès lors qu'une valeur est modifiée, les données du modèle sont automatiquement mises à jour !

ng-options permet de peupler notre liste déroulante et de définir ce qui sera affiché. Ici on affichera la clé de chaque **gender**

Enfin **ng-click** permet de spécifier le nom de la fonction qui sera appelée lors d'un clic sur le bouton.

Mise à jour des données

Il ne nous reste maintenant plus qu'à ajouter nos fonctions **removeClient()** et **addClient(client)** à notre **controller** :

```
$scope.addClient = function () {
    $scope.clients.push({ gender: Genders[0], name: "Inconnu", age: 0 });
};

$scope.removeClient = function (client) {
    index = $scope.clients.indexOf(client);
    $scope.clients.splice(index, 1);
};
```

addClient() ajoute seulement un nouveau client à notre collection grâce à la fonction JavaScript **push()** et **removeClient(client)** supprime un client avec la fonction **splice()**. Simple n'est-ce pas ? Passons maintenant à Knockout.

Knockout

Knockout utilise le pattern MVVM. Ce pattern permet de garder une organisation simple d'une application graphique en séparant le code en trois parties:

- **Model**: il décrit les données manipulées par l'application. Il offre des méthodes permettant de récupérer et de modifier ces données.
- **View Model**: est une abstraction de la vue qui sert de médiateur entre le modèle et la vue. Il fournit les fonctions permettant d'ajouter ou supprimer des éléments, etc...
- **View**: affiche les informations du **View Model**, envoie des commandes telles qu'un clic sur un bouton, et est automatiquement mis à jour dès qu'il y a un changement sur le **View Model**.

Initialisation

Avec Knockout il faut lier manuellement le **Model** au **View Model**.

Commençons donc par modifier notre modèle, **Genders** n'étant pas modifiable, sa structure ne change pas :

```
var Genders = [
    {
        id: 0,
        key: "Mr",
        value: "Monsieur"
    },
    {
        id: 1,
        key: "Mme",
        value: "Madame"
    }
]
```

Nous modifions maintenant le modèle de **Client**, on crée donc un constructeur avec la structure suivante :

```
var Client = function (id, gender, name, age) {
    this.id = id;
    this.gender = ko.observable(gender);
    this.name = ko.observable(name);
    this.age = ko.observable(age);

    this.clientFormatted = ko.computed(function () {
        return this.gender().value + ": " + this.name() + " (" + this.age() + " ans)";
    }, this);
}
```

Knockout a besoin de savoir quels champs du **View Model** observer afin de notifier la ou les **Views** des changements, pour cela on utilise **ko.observable(value)** qui prend en paramètre la valeur par défaut. Comme avec AngularJS, on veut avoir une sortie formatée qui affiche les informations de notre **Client**. Pour cela on utilise **ko.computed(function () {...})** qui permet à knockout de savoir que cette propriété dépend d'un ou plusieurs **observables**, et donc qu'il doit la mettre à jour lorsqu'une des dépendances est modifiée. Passons maintenant à notre **View Model**. On crée donc un objet contenant notre liste de clients :

```
var ViewModel = function () {
    var self = this;

    self.clients = ko.observableArray([
        new Client(0, Genders[0], "Jean", 25),
        new Client(1, Genders[1], "Juliette", 30),
    ]);
}
```

Notre tableau est un **observableArray** qui permet de suivre l'état de notre collection, ce qui sera utile lorsque nous implémenterons l'ajout et la

suppression de nouveaux clients dans notre collection. Enfin il faut activer Knockout afin d'associer notre **View** à notre **View Model**. Il suffit pour cela d'ajouter la ligne suivante à la fin de la page html contenant notre vue :

```
ko.applyBindings(new ViewModel());
```

Affichage des données

Maintenant que notre **View Model** est correctement construit passons à la vue. On commence par inclure knockout dans le header de notre document :

>> voir code complet (lien à la fin de l'article).

Le début du tableau est identique à AngularJS, on a ensuite la balise `tbody` avec un attribut `data-bind`. L'attribut `data-bind` n'est pas natif en html mais ne provoque pas d'erreur de validation et n'est pas interprété par le navigateur. Knockout va interpréter ces balises au moment de l'association entre la **view** et le **view model** (`ko.applyBindings(view)`). Le premier `data-bind` que l'on retrouve (`data-bind="foreach: clients"`) est simplement une boucle qui va parcourir notre collection de clients. Vient ensuite la liste déroulante de **Gender** (`data-bind="options:Genders, optionsText:'key', optionscaption:'Choose gender', value:gender"`). `options` représente la collection depuis laquelle on va afficher les données, `optionsText` est le champ que l'on souhaite afficher dans la liste, `optionscaption` est la valeur par défaut à afficher, et, enfin, `value` est l'attribut value que vont prendre les éléments de notre liste déroulante. On affiche ensuite nos différents `input` permettant à l'utilisateur de modifier les informations d'un client. Pour cela on ajoute l'attribut : `data-bind="value: name"` afin de lier notre vue au champ `name` de notre **view model**. Dès que la valeur est modifiée le **view model** est automatiquement mis à jour avec les nouvelles valeurs et affiché dans notre sortie formatée. Notre sortie formatée est simplement affichée de la même façon qu'avec les `input` sauf que l'on spécifie que la valeur n'est pas modifiable : `data-bind="text: clientFormatted"`

Mise à jour des données

Vous aurez remarqué l'ajout des boutons ajouter et supprimer qui appelle respectivement les fonctions `addClient` et `removeClient` grâce au tag `data-bind="click: addClient"`. Nous allons maintenant ajouter ces deux méthodes à notre **view model**. Dans notre classe **ViewModel** on ajoute donc :

```
self.addClient = function () {
    self.clients.push(new Client(null, Genders[0], "Inconnu", 0));
};

self.removeClient = function (client) {
    index = self.clients.indexOf(client);
    self.clients.splice(index, 1);
};
```

De la même façon qu'avec AngularJS, lorsque l'on ajoute ou on enlève un **Client** de notre collection, les modifications seront automatiquement reportées dans la vue.

Désormais l'application Knockout fonctionne. Comme on a pu le voir, il y a beaucoup de ressemblances entre AngularJS et Knockout. Bien qu'il y ait de nombreuses similitudes entre AngularJS et Knockout, on note toutefois plusieurs différences dont les principales sont :

- AngularJS définit des balises particulières pour chaque action (`ng-*`) au sein desquelles on peut mettre du code JavaScript. Tandis que Knockout utilise des balises `data-bind` à l'intérieur desquelles il définit des mots clefs indiquant l'action à effectuer.
- La seconde grosse différence est qu'avec Knockout il faut spécifier explicitement quels champs du modèle devront être observés afin de notifier la vue. Alors qu'avec AngularJS tout est implicite.

Ember

Ember utilise le pattern MVC comme AngularJS.

Premièrement nous allons avoir besoin d'Ember-data qui est une bibliothèque permettant de gérer les données et de faciliter les interactions avec le serveur, nous allons en avoir besoin pour créer nos modèles. Tous les appels de méthodes commençant par « **DS.** » utiliseront cette bibliothèque.

Initialisation

Commençons par créer une application Ember de la façon suivante :

```
window.App = Ember.Application.create();
```

Cette ligne rendra la variable `App` disponible dans toute notre application. C'est cette `App` qui contiendra toute la logique d'Ember. Créons maintenant un **Adapter** qui permet de communiquer avec la source de données. Dans notre cas comme nous n'utilisons pas de base de données, **Ember-data** nous fournit un système de **Fixture** qui permet de charger les données depuis notre tableau JavaScript :

```
App.ApplicationAdapter = DS.FixtureAdapter.extend();
```

Les données sont les mêmes que précédemment, à la différence que le **Gender** d'un client est maintenant donné par son id et non plus directement par l'objet :

```
var Genders = [
  {
    id: 0,
    key: "Mr",
    value: "Monsieur"
  },
  {
    id: 1,
    key: "Mme",
    value: "Madame"
  }
];

var ClientsList = [
  {
    id: 0,
    gender: 0, //Note: on donne l'ID et non l'objet !
    name: "Jean",
    age: 25,
  },
  {
    id: 1,
    gender: 1,
    name: "Juliette",
    age: 30,
  }
];
```

Déclarons le modèle qui accueillera nos données :

```
App.Gender = DS.Model.extend({
  key: DS.attr('string'),
  value: DS.attr('string'),
});
```



```
App.Client = DS.Model.extend({
  gender: DS.belongsTo('gender', { async: false }), //relation 1 à 1
  name: DS.attr('string'),
  age: DS.attr('boolean'),

  clientFormatted: function () {
    if (this.get('gender'))
      var gender = this.get('gender').get('value');
    else
      var gender = "Inconnu"

    return gender + ' : ' + this.get('name') + ' (' + this.get('age') + ' ans)';
  }.property('gender', 'name', 'age'),
});
```

A la différence des autres Frameworks, Ember contrôle le type de chaque champ de la même façon qu'une base de données. On remarque aussi la relation avec **Gender** qui est définie explicitement. Le champ **clientFormatted** est calculé à partir des autres champs, comme dans les exemples précédents. Il faut aussi vérifier que le champ **gender** est bien défini pour éviter une erreur. Il ne reste plus qu'à charger les données depuis nos collections **ClientsList** et **Genders**; on fait cela grâce aux **Fixtures** :

```
App.Gender.FIXTURES = Genders;
App.Client.FIXTURES = ClientsList;
```

Pour rendre nos deux modèles accessibles dans l'application, nous définissons la route principale :

```
App.ApplicationRoute = Ember.Route.extend({
  model: function () {
    return Ember.RSVP.hash({
      clients: this.store.find('client'),
      genders: this.store.find('gender')
    })
  },
});
```

Ensuite on associe cette route à une URL :

```
App.Router.map(function () {
  this.resource('clients', { path: '/' });
});
```

Cette route permet de demander à Ember de détecter lorsque l'URL de la page est « / » et d'afficher le Template **clients**, que nous verrons. Un Template est défini avec la syntaxe de scripts **handlebars** de la façon suivante :

```
<script type="text/x-handlebars" data-template-name="clients">
  <!--
    Corps du template
  -->
</script>
```

Affichage des données

A l'intérieur de ce bloc de script, tout ce qui est entre deux accolades ouvrantes et deux fermantes : `{{...}}` sera interprété par Ember. Voici donc le code de la page

```
<script type="text/x-handlebars" data-template-name="clients">
  <div class="container">
    <table class="table">
      <caption>Liste de clients</caption>
      <tr>
        <th>Gender</th>
        <th>Name</th>
        <th>Age</th>
        <th>Output</th>
        <th></th>
      </tr>
      {{#each client in model.clients}}
        <tr>
          <td>
            {{view "select" content=model.genders optionValuePath="content.id" optionLabelPath="content.key" prompt="Choose gender" selectionBinding="client.gender"}}
          </td>
          <td>{{input type="text" value=client.name}}</td>
          <td>{{input type="text" value=client.age}}</td>
          <td>{{client.clientFormatted}}</td>
          <td><button type="button" {{action 'deleteClient' client}}>Supprimer</button>
        </td>
      </tr>
    </table>
    <button type="button" {{action 'addClient' }}>Ajouter un client</button>
  </div>
</script>
```

Beaucoup de ressemblances avec les précédents exemples. Le « **foreach** » devient: `{{#each client in model.clients}}`.

La liste déroulante `{{view "select" ...}}` prend en paramètres **content** qui est le modèle qui va être listé, **optionValuePath** est la valeur que va prendre l'attribut value de chaque **option**, **optionLabelPath** est la valeur que va prendre chaque **option**, **prompt** est la valeur par défaut et enfin **selectionBinding** est la valeur qui va être sélectionnée (dans notre cas le **gender** de notre **client** courant).

Les champs de saisie et d'affichage du texte formaté ressemblent aux précédents exemples.

Mise à jour des données

On voit ensuite les deux boutons permettant d'ajouter et de supprimer des clients. Ces boutons appellent la fonction désignée par la balise `{{action 'actionName' }}`. On peut donner un paramètre, comme avec l'exemple du bouton supprimer qui prend en paramètre le client que l'on souhaite supprimer.

Nous allons maintenant écrire ces fonctions d'ajout et de suppression de clients, pour cela ajoutons un **Controller** à notre code JavaScript :

```
App.ClientsController = Ember.ObjectController.extend({
  actions: {
    addClient: function () {
      var todo = this.store.createRecord('client', {
        gender: 0,
        name: "Inconnu",
        age: 0
      });
    },
  },
});
```

```
deleteClient: function (client) {
  client.deleteRecord();
  client.save();
}
};
```

Dans ce Contrôleur sont définies les deux fonctions. Pour ajouter un **Client** on appelle `store.createRecord()` qui prend en paramètre le nom du modèle et les valeurs par défaut. Pour supprimer un client on appelle simplement `deleteRecord()` sur le client donné en paramètre. Voilà pour Ember, passons maintenant à Backbone.

Backbone

Finissons par Backbone, ce Framework utilise le pattern MVP qui est légèrement différent du pattern MVC présenté précédemment.

Modèle : comme pour les autres patterns, il décrit la logique métier, les données et les fonctions de manipulation de données.

View : affiche les données du modèle. Dans ce pattern, la vue n'a aucune connaissance du modèle, tout passe par le **Presenter**.

Presenter : gère les événements de l'interface graphique. A la différence d'un **Contrôleur**, le **Presenter** est totalement découplé de la vue et communique avec elle à travers une interface. Il n'y a pas de liaison entre la vue et le modèle, tout passe par le **Presenter**.

Initialisation

Nous allons voir que le fonctionnement de ce Framework est très différent des deux autres. Toute la logique se trouve déportée dans le code JavaScript (le **Presenter**), c'est pourquoi on se retrouve avec un code HTML très minimaliste :

```
<div id="backBoneView"> <!-- View -->
</div>
```

Voilà le seul code HTML dont nous avons besoin ! Regardons maintenant le code JavaScript. On commence toujours avec le même jeu de données :

```
var Genders = [
  {
    id: 0,
    key: "Mr",
    value: "Monsieur"
  },
  {
    id: 1,
    key: "Mme",
    value: "Madame"
  }
];

var ClientsList = [
  {
    id: 0,
    gender: Genders[0],
    name: "Jean",
    age: 25,
  },
  {
```

```
id: 1,
gender: Genders[1],
name: "Juliette",
age: 30,
}
];
```

Créons donc notre modèle **Client** :

```
var Client = Backbone.Model.extend({
  defaults: {
    id: -1,
    gender: Genders[0],
    name: "Inconnu",
    age: 0,
  },

  clientFormatted: function () {
    return this.get('gender')['value'] + ": " + this.get('name') + " (" + this.get('age') + " ans)";
  }
});
```

Le modèle prend donc une valeur par défaut qui sera utilisée lors de la création d'un nouveau **Client** et une fonction d'affichage formatée.

On ne va pas créer de modèle pour le **Gender** afin d'éviter d'alourdir le code pour un modèle qui n'est de toute façon pas modifiable.

Lions maintenant notre modèle à une **Collection**. Une collection est simplement une liste de modèles :

```
var List = Backbone.Collection.extend({
  model: Client
});
```

Créons ensuite la vue principale qui affichera la liste de nos clients et gèrera l'événement d'ajout d'un nouveau client :

```
var ListView = Backbone.View.extend({
```

Toutes les vues de Backbone contiennent une propriété `el` qui référence le DOM. Si cette propriété n'est pas définie, Backbone va en construire une automatiquement avec un élément `div` vide. Dans notre cas nous référençons l'élément HTML défini précédemment :

```
el: $('#backBoneView'),
```

Cette vue prend une fonction d'initialisation qui est automatiquement appelée au moment de l'instanciation de notre vue. Cette fonction instancie notre collection de clients.

```
initialize: function () {
  __.bindAll(this, 'render', 'addClient', 'appendClient');

  this.collection = new List(ClientsList);
  this.collection.bind('add', this.appendClient);

  this.counter = -1;
  this.render();
},
```

`__.bindAll(this, 'functionName', ...)` permet aux fonctions données en paramètre d'avoir accès au contexte « `this` ».

Les lignes suivantes permettent de créer la collection **List** avec le jeu de donnée **ClientsList** et de lier l'événement **add** avec la fonction

appendClient. La variable **counter** nous permettra de changer l'ID d'un client au moment de l'ajout.

Affichage des données

Enfin nous appelons la méthode **render** qui permet d'afficher notre liste. Cette fonction va générer le code HTML de notre tableau :

```
render: function () {
    var self = this;

    $(this.el).append("<table class='table'><caption>Liste de clients</caption><tr>
<th>Gender</th><th>Name</th><th>Age</th><th>Output</th><th></th>
</tr></table>");

    _(this.collection.models).each(function (item) {
        self.appendClient(item);
    }, this);

    $(this.el).append("<button type='button' id='add'>Ajouter un client</button>");
},
```

Backbone préconise de garder au maximum la logique dans le code JavaScript (dans le **Presenter**), c'est pourquoi on ne crée pas le tableau directement dans le code HTML de notre page, mais on préfère le générer dans la fonction de rendu.

On parcourt notre collection, et pour chaque client on appelle la fonction **appendClient()** suivante :

```
appendClient: function (item) {
    var client = new ClientView({
        model: item
    });

    $('table', this.el).append(client.render().el); //Automatically add <tr></tr>
}
```

Mise à jour des données

Cette fonction crée une vue client dont nous allons voir la syntaxe plus tard, et l'ajoute à notre tableau. Enfin, on ajoute un évènement **click** à notre bouton qui appellera la fonction **addClient()**:

```
events: {
    'click button#add': 'addClient',
},

addClient: function () {
    var client = new Client();
    client.set({
        id: this.counter
    });
    this.counter--;
    this.collection.add(client);
},

});
```

La fonction **addClient()** crée un nouveau **Client**, change son **id** et finalement l'ajoute à notre collection.

Voilà pour notre vue principale qui s'occupe d'afficher la liste de tous les clients.

Passons maintenant à la vue d'un seul client. Cette vue est

appelée dans la fonction **appendClient()** vue précédemment et va nous permettre d'afficher chaque client individuellement.

```
var ClientView = Backbone.View.extend({
```

On ajoute un attribut **tagName** qui est le tag de l'élément qui sera créé pour chaque client.

```
    tagName: 'tr',
```

La méthode d'initialisation fonctionne de la même façon que la vue générale à la différence qu'ici on ne travaille plus sur une collection mais sur un modèle:

```
initialize: function () {
    _bindAll(this, 'render', 'updateName', 'updateAge', 'unrender', 'remove');
    this.model.bind('remove', this.unrender);
},
```

Passons maintenant à la fonction d'affichage de notre vue **Client**:

```
render: function () {
    var self = this;
```

On commence par la fonction qui construit la liste de **Genders**. On aurait pu passer par un autre modèle avec sa vue, mais pour garder le code concis nous allons seulement parcourir notre jeu de données et afficher ses valeurs.

```
    var select = "<select id='gender'>";
    Genders.forEach(function (entry) {

        var selected = "";
        if (entry.id == self.model.get('gender')['id'])
            selected = 'selected';

        select += "<option " + selected + " value='" + entry.value + "'" + entry.key + "></option>";

    });
    select += "</select>";
```

Ensuite on affiche le modèle du **Client** courant :

```
    $(this.el).html("<td>" + select + "</td><td><input type='text' id='name' value="
    + this.model.get('name') + "'></td><td><input type='text' id='age' value='" +
    this.model.get('age') + "'></td><td>" + this.model.toString() + "</td><td><button
    type='button' class='delete'>Supprimer</button></td>");
    return this;
},
```

Maintenant que notre affichage est terminé, on ajoute les évènements correspondants au changement de sélection dans notre liste déroulante, au changement dans les textes boxes du formulaire, ainsi qu'au clic sur le bouton supprimer.

```
events: {
    'change select#gender': 'genderChange',
    'change input#name': 'updateName',
    'change input#age': 'updateAge',
    'click button.delete': 'remove'
},

genderChange: function(e){
    var index = $(e.currentTarget)[0].selectedIndex;
    this.model.set({ 'gender': Genders[index] });
    this.render();
},
```



```
updateName: function (e) {
    var val = $(e.currentTarget).val();
    this.model.set({ 'name': val });
    this.render();
},
updateAge: function (e) {
    var val = $(e.currentTarget).val();
    this.model.set({ 'age': val });
    this.render();
},
remove: function () {
    this.model.destroy();
},
unrender: function () {
    $(this.el).remove();
}
});
```

Lors d'un évènement on récupère la nouvelle valeur, on change la valeur correspondante dans le modèle, puis on appelle la fonction de rendu. La fonction unrender () est automatiquement appelée lors de la suppression du modèle. On l'a liée dans la méthode initialize() précédente. Finalement il ne reste plus qu'à créer notre vue principale afin que le rendu soit fait :

```
var listView = new ListView();
```

Voilà! Nous avons réalisé le même exemple d'application avec les 4 Frameworks. On a aussi remarqué un certain nombre de ressemblances et de différences.

En résumé

	AngularJS	Knockout	Ember	Backbone
Lignes de code	170 000	10 000	50 000	10 000
Poids avec dépendances (gzipped)	56.9KB	19.8KB	129 + 37.1* + 28.7** + 28.6*** = 223.4KB	6.8 + 37.1* + 5.5**** = 49.4KB
Contributeurs sur GitHub	1078	48	452	242
Projets annexes	400	100	500	200
Questions sur StackOverflow	66 444	12 374	12 846	16 863
Patterns de conception supportés	MVC	MVVM	MVC	MVP

*jquery, **ember-data, ***handlebar, ****underscore

Le temps de chargement d'une page Web est crucial pour sa réussite. Les utilisateurs ne montrent pas beaucoup de patience quant au temps de chargement d'une page Web, c'est pourquoi il est essentiel de prendre en compte le temps de chargement et d'initialisation d'une librairie. Ce temps aura aussi un impact si un Framework est lourd, car si son utilisation nécessite peu de lignes de code, il pourra tout de même être avantageux.

Pas de vainqueurs

AngularJS se démarque de ses concurrents grâce à sa simplicité d'utilisation, sa documentation très bien faite et sa communauté active sur StackOverflow. De plus sa méthode pour enrichir le code HTML avec ses balises personnalisées le rendent très lisible. Ce sont les raisons qui font qu'AngularJS est rapide à prendre en main. De plus il est développé et maintenu par Google ce qui est gage d'une qualité certaine. On pourrait lui reprocher un manque de modularité. Malgré tout, il convient parfaitement pour la majorité des projets.

Knockout ressemble beaucoup à AngularJS, sa documentation et notamment son système de tutorial interactif le rendent très simple à prendre en main. Malheureusement sa syntaxe qui consiste à tout mettre dans un seul attribut HTML (data-bind) fait perdre la coloration syntaxique. Le code de la page Web devient donc difficilement lisible sur de gros projets.

Ember se détache de ses concurrents par son système de route plus avancé qui l'oriente vers des sites SPA (Single Page Application). Son système qui lui permet de spécifier le type des attributs et son système de store simplifie les connexions à des bases de données. Malheureusement son apprentissage est un peu plus compliqué et sera réservé à des projets de plus grosse envergure.

Backbone Tout le code HTML étant généré par le code JavaScript, il est donc plutôt adapté à des développeurs maîtrisant bien ce langage. Toute la construction du code HTML et la gestion des événements doivent être gérées par le développeur, ce qui alourdit fortement le code final. Sa documentation ne propose aucun tutoriel pas à pas comme le font ses concurrents. Ainsi l'apprentissage de ce Framework peut se révéler compliqué pour un développeur débutant. Malgré tout, il offre une plus grande modularité par rapport à ses concurrents. Ainsi, il n'y a donc pas de réponse catégorique quant au choix d'un Framework JavaScript. Il dépendra avant tout du besoin, du temps et de l'équipe qui participera au projet. Cet article traite seulement de 4 des Frameworks les plus connus, mais il en existe des dizaines offrant des fonctionnalités différentes.

Développer son propre Framework ?

Utiliser un Framework connu permet aux développeurs d'être plus rapidement opérationnels sur un projet, et de faciliter le transfert de la maintenance du code source à un client ou à une équipe tierce. Mais alors pourquoi développer son propre Framework et réinventer la roue ? Tout d'abord cela permet d'avoir plus de souplesse, on peut le modifier, l'adapter et le faire évoluer selon les besoins, tout en restant plus léger qu'un Framework existant dont on n'utilisera probablement pas toutes les fonctionnalités. On décide de l'architecture à adopter et on n'est pas restreint à celle imposée par le Framework que l'on utilise. Malgré ces avantages, réinventer la roue n'est pas toujours la bonne solution; même si les composants développés seront réutilisables au fil des projets, un nouveau membre dans l'équipe de développement aura du mal à en comprendre les subtilités et mettra donc du temps à s'adapter. Il ne pourra pas forcément s'aider d'Internet ou de documentations pour avancer. De plus Il faudra gérer vous-même la maintenance, les évolutions et les adaptations aux nouveautés des langages sur lesquels il repose. La compatibilité avec les différents navigateurs par exemple, est un problème qui devra être pris en compte dès le début du développement. Cette solution est à envisager dans de rares cas seulement où des exigences bien précises sont requises, car elle demande beaucoup de travail supplémentaire, alors que de nombreux Frameworks existent déjà.

Sources :

Codes complets des exemples :

<https://github.com/ludo6577/JavascriptFrameworks>

Sites officiels :

AngularJS : <https://angularjs.org/>

Knockout : <http://knockoutjs.com/>

Ember : <http://emberjs.com/>

Backbone : <http://backbonejs.org/>

JavaScript pour les Jedis, épisode I : au cœur des fonctions

2015 est l'année de JavaScript. C'est pour cette raison que j'ai décidé de me consacrer à la rédaction d'une série d'articles concernant les fondamentaux de JavaScript, un langage très populaire mais en même temps encore méconnu. Ces articles auront pour but de vous expliquer en détail le fonctionnement de ce langage afin de faire de vous des JavaScript Jedis.



Wassim Chegham,
JavaScript ninja et consultant en technologies Web chez
SII Ouest.

Grâce à cette série d'articles, nous allons apprendre et comprendre JavaScript afin de tirer le meilleur de ce langage. A vrai dire, j'en ai un peu marre de voir beaucoup de développeurs traiter ce langage de tous les noms car ils n'arrivent pas à faire ce qu'ils ont l'habitude de produire avec leurs langages favoris. Mais en réalité, cela vient du fait que ces mêmes personnes ne prennent même pas la peine d'apprendre le langage, probablement à cause de sa syntaxe très ressemblante à Java.

« JavaScript, tu l'apprends
ou tu le quittes »

Vous vous demandez sûrement pourquoi nous commençons cette série par les fonctions, et non pas par les objets. Eh bien ! Étant donné la nature fonctionnelle de JavaScript, pour moi, maîtriser cet aspect du langage c'est ce qui fera de vous des Jedis. En effet, la plupart des développeurs venant d'autres langages orientés objet, essayent plus de reproduire des paradigmes propres à ces langages et font l'impasse sur les fonctions et les fermetures (closures). JavaScript est composé de ces trois concepts : les objets, les fonctions et les closures. Maîtriser JavaScript passe par l'apprentissage de cet aspect fonctionnel du langage ; Et croyez-moi, le niveau de sophistication du code que vous écrirez dépend de cet apprentissage.

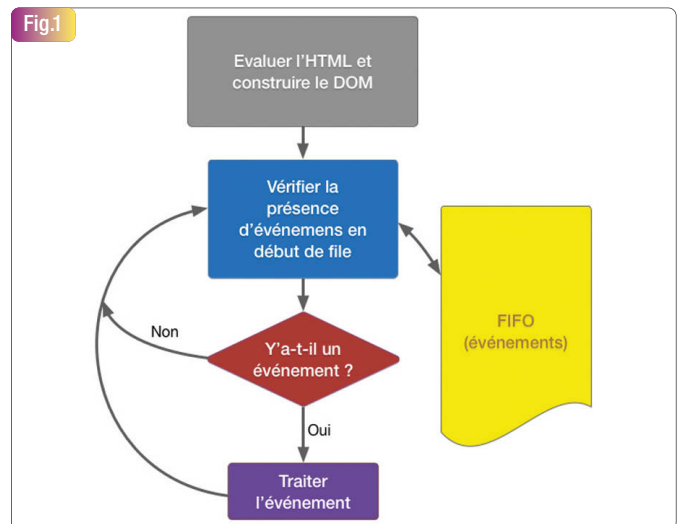
JavaScript est un langage fonctionnel

L'une des raisons qui fait que les fonctions et la nature fonctionnelle de JavaScript sont des concepts très importants, vient du fait que « la fonction » est le premier module d'exécution en JavaScript. C'est-à-dire que lorsque votre code est exécuté, il l'est au sein d'une fonction ; à l'exception des scripts qui sont interprétés lorsque le code HTML est en train d'être évalué par le navigateur. Je fais évidemment allusion à l'emblématique « document.write() » qui permettait de créer du DOM au runtime, obligeant le navigateur à bloquer l'interprétation du code HTML. Revenons à nos moutons, une chose très importante à connaître sur les fonctions en JavaScript, ce sont des objets de premier ordre (first-class objects). Ceci veut dire que les fonctions sont traitées comme des objets, elles peuvent donc être :

- Créées via des littérales,
- Assignées à des variables ou propriétés d'un objet,
- Passées en paramètre,
- Retournées comme résultat,
- Elles peuvent posséder des objets ou méthodes.

En plus d'être traitées comme des objets, les fonctions ont également une capacité spéciale : elles peuvent être invoquées (on en reparlera plus en détails dans la suite de cet article). Cette invocation est souvent effectuée de manière asynchrone. Voici pourquoi...

Fig.1



La boucle d'événements

Si vous avez déjà codé une interface graphique auparavant, vous avez sûrement procédé comme ceci :

- Mettre en place l'interface graphique,
- Se mettre dans une boucle d'événements en attente qu'un événement se produise,
- Invoquer les actions à exécuter pour chaque événement.

La programmation en JavaScript — dans un navigateur — est quasiment similaire, sauf que la gestion des événements est totalement prise en charge par le navigateur. Nous avons juste besoin de spécifier les actions (listeners) pour les différents événements qui peuvent être déclenchés — toujours au sein du navigateur. Ces événements sont placés dans une file au fur et à mesure qu'ils se produisent, et le navigateur traite ces événements en invoquant les actions associées. A noter que ce principe reste identique pour d'autres environnements JavaScript, par exemple NodeJS. Puisque ces événements peuvent se produire à n'importe quel moment et dans n'importe quel ordre, la gestion de ces événements, et donc l'invocation de leurs actions, se fait de manière asynchrone. Pourquoi me direz-vous ? Une chose à savoir, la boucle d'événements du navigateur est « mono-threadée », ce qui veut dire que chaque événement se trouvant dans la file d'attente, est traité dans l'ordre d'arrivée (FIFO). Les événements sont donc traités un par un, à tour de rôle. Voici un schéma très simplifié de ce processus : Fig.1.

Un exemple typique du fonctionnement de cette boucle : lorsque vous bougez le curseur de votre souris dans votre page Web, le navigateur détecte ces déplacements de souris et place cette série d'événements (mousemove) dans la FIFO. Ensuite, la boucle d'événements s'occupera du traitement de ces événements, en exécutant les actions associées. Ce principe d'actions — associer une fonction qui sera exécutée plus tard, lorsque l'événement se produira — illustre un mécanisme que l'on appelle les fonctions de callback.

Le principe de callback

Les fonctions de callback sont une partie très essentielle dans la bonne compréhension de JavaScript. Explorons ce point... Dans la partie précédente, nous avons dit que l'on pouvait associer des actions aux différents événements se produisant au sein du navigateur. Voici un exemple d'une action :

```
function executerAuChargement() { /* ... */ };
window.onload = executerAuChargement;
```

Ici, nous avons attaché une action (listener) qui sera exécutée lorsque la page est complètement chargée par le navigateur. Parce que les fonctions sont des « first-class object », nous pouvons affecter cette fonction à la propriété onload de l'objet window (représentant le navigateur). Aussi, nous avons dit que les fonctions peuvent être passées en paramètre. Ce qui veut dire que l'on peut écrire ceci :

```
function dire(callback) {
  alert(callback());
}

function bonjour() {
  return "Bonjour, JavaScript!";
}

function executerAuChargement() {
  return dire(bonjour)
}

window.onload = executerAuChargement;
```

Voilà un exemple intéressant ! Nous pouvons aussi appeler les fonctions de callback dans nos propres fonctions, pas besoin qu'elles soient associées à des événements. Dans cet exemple, nous avons attaché une fonction de callback à l'évènement onload. Lorsque la boucle d'évènement exécutera ce listener, à son tour, il exécutera la fonction « dire » à laquelle nous avons passé la référence vers la fonction « bonjour ». Une fois exécutée, la fonction « dire » invoquera la fonction « bonjour » ce qui provoquera l'affichage du message « Bonjour, JavaScript! ». La nature fonctionnelle de JavaScript nous permet de créer des fonctions en tant qu'entité à part entière, comme n'importe quel autre type, et les passer directement en paramètre. Voyons un exemple :

```
function dire(callback) {
  alert(callback());
}

function executerAuChargement() {
  return dire(() => "Bonjour, JavaScript!");
}

window.onload = executerAuChargement;
```

Nous avons modifié l'exemple précédent en supprimant la déclaration de la fonction « bonjour ». Nous l'avons remplacée par la déclaration d'une fonction sans nom, directement dans les paramètres de la fonction « dire ». En JavaScript, nous appelons ce genre de fonctions, les fonctions anonymes.

Les fonctions anonymes

Étant donné la nature fonctionnelle du langage, il est possible de créer des fonctions n'importe où dans le code — là où une expression est attendue. Cela a pour avantage de rendre le code plus clair, plus compact. En plus,

cela permet d'éviter de polluer l'espace de nom globale avec des noms de fonctions inutiles. En effet, une fonction anonyme est une fonction — comme les autres — déclarée en « in-line », qui n'a pas de nom pour la référencer. Ces fonctions anonymes sont généralement déclarées au moment où elles sont référencées. Généralement en JavaScript, lorsque nous estimons qu'une fonction va être référencée dans plusieurs endroits du code, nous lui donnons un nom. Sinon, si elle est destinée à servir juste à un seul endroit, alors pas besoin qu'elle ait un nom. Ce qui nous amène aux déclarations et invocations des fonctions.

Déclaration de fonctions

En JavaScript, les fonctions sont déclarées à l'aide du mot-clé « function » qui crée une valeur de type fonction. Rappelez-vous que les fonctions sont des objets de premier ordre, elles peuvent être manipulées dans le langage comme n'importe quel autre type. De ce fait, il nous est possible d'écrire ceci (en reprenant l'exemple précédent) :

```
var dire = function(callback) {
  alert(callback());
};

// ou en notation ES6
var dire = (callback) => alert(callback());
```

La valeur de la variable « dire » est une fonction ! Nous avons donc déclaré et créé une fonction anonyme référencée par la variable « dire ». Toutes les fonctions déclarées ont une propriété « name » qui contient le nom de la fonction (de type String). Pour les fonctions anonymes, cette propriété est présente mais elle est vide. Lorsqu'une fonction a été déclarée avec un nom, ce nom reste valide dans tout le contexte dans lequel cette fonction a été déclarée. Aussi, si cette fonction a été déclarée dans le contexte globale, une nouvelle propriété portant le nom de la fonction est ajoutée dans le « window » ; ceci est également vrai pour les variables. Ceci nous amène donc à la suite de cet épisode concernant le contexte et la portée des fonctions.

Portée et contextes d'une fonction

Lorsque nous déclarons une fonction, nous devons avoir à l'esprit, non seulement le contexte dans lequel cette fonction existe, mais également quels sont le ou les contextes que cette fonction crée. Nous devons également faire très attention aux déclarations faites au sein de ces contextes. La gestion des portées en JavaScript est très différentes de la plupart des autres langages dont la syntaxe a été influencée par le langage C. Plus précisément, ceux utilisant les accolades pour délimiter la portée des variables dans les blocs. Dans ces langages, chaque bloc crée son propre contexte. Ce n'est pas le cas de JavaScript. La portée d'une variable créée au sein d'un bloc continue d'exister en dehors de ce dernier. Par exemple :

```
if (true) {
  var foo = 'bar';
}
console.log(foo); // => 'bar'
```

Ce simple exemple prouve bien que la portée de la variable « foo » est globale, elle n'est pas liée au bloc « if ». Regardons un exemple un peu plus intéressant :

```
function fonction1() {
  var a = 1;
```



```
function fonction2() { /* ... */ }
var b = 2;
if (a === 1) {
  var c = 3;
}
fonction1();
```

Dans cet exemple, nous avons déclaré deux fonctions, « fonction1 » et « fonction2 », ainsi que trois variables, a, b et c. Pour compléter cet exemple, voici un schéma plus parlant qui explique les différents contextes créés : **Fig.2**.

Ce schéma résume les règles suivantes :

- La portée d'une variable existe depuis sa déclaration et ce jusqu'à la fin de la fonction dans laquelle elle a été déclarée, peu importe l'imbrication des blocs (variables declaration hoisting),
- La portée d'une fonction (non-anonyme) est globale à toute la fonction dans laquelle elle a été déclarée (functions declaration hoisting).

Cette deuxième règle nous montre que, contrairement aux variables, les fonctions peuvent être référencées avant leurs déclarations ! Oui, exactement, vous pouvez invoquer « fonction1 » avant de la déclarer. Par contre, ce n'est pas une bonne pratique. Ce n'est pas parce que c'est autorisé par le langage qu'il faut le faire. Maintenant que nous avons vu comment les fonctions sont déclarées, essayons de voir comment elles peuvent être invoquées. Attention, cela risque d'être très amusant...

Invocation des fonctions

Nous avons tous invoqué des fonctions en JavaScript. Mais avez-vous déjà essayé de comprendre ce qui se passe lorsque vous invoquez une fonction ? Il existe quatre façons d'invoquer une fonction en JavaScript. Chaque type d'invocation a un effet direct sur le contexte d'exécution de cette fonction. Plus précisément, le type d'invocation agit sur le paramètre « this » passé à la fonction lors de son invocation. Ecrire du code digne d'un Jedi, repose sur la compréhension et la bonne maîtrise de ces mécanismes d'invocations.

Voici les quatre types d'invocation en JavaScript :

- En tant que fonction (c'est le cas le plus répandu),
- En tant que méthode, ce qui permet de lier l'invocation à un objet (POO),
- En tant que constructeur, ce qui permet de créer un objet (instance),
- Via « apply » et « call » (expliqué plus loin).

Avant d'expliquer chaque type d'invocation, prenons quelques minutes et essayons de comprendre ce qui se passe lors d'une invocation de fonction (cela est vrai pour les cas 1, 2, et 3).

Arguments et paramètres

Lorsqu'une liste d'arguments est fournie lors d'une invocation de fonction, ces arguments sont affectés aux paramètres (spécifiés lors de la déclaration) de la fonction, dans le même ordre. Jusque-là rien de bien sorcier.

Maintenant, si le nombre d'arguments est différent de celui des paramètres, aucune erreur n'est déclenchée : JavaScript sait comment traiter ce cas :

- S'il y a plus d'arguments que de paramètres, alors l'excédent est ignoré,
- S'il y a moins d'arguments que de paramètres, alors les paramètres qui n'ont pas d'arguments seront mis à « undefined ».

De plus, au moment de l'invocation, deux paramètres additionnels sont passés implicitement : « this » et « arguments ». Autrement-dit, ces deux paramètres sont passés à la fonction qui est invoquée, même s'ils ne sont pas spécifiés en tant que paramètres dans la déclaration. Ils sont par la suite disponibles dans le corps de la fonction et peuvent être référencés comme n'importe quels autres vrais paramètres.

L'argument « arguments »

Cet objet est une sorte de collection de tous les arguments passés à la fonction.

```
function foo() {
  console.log(arguments.length); // => 3
  console.log(arguments); // => [1,2,'bar']
  console.log(arguments[2]); // => 'bar'
  console.log(arguments.filter()); // => undefined
}
```

Mais attention, même si cet objet possède un attribut « length », et permet d'accéder aux arguments via une notation comme pour un tableau « arguments[i] », ce n'est pas un tableau à proprement parler. Vous ne pouvez pas utiliser les méthodes spécifiques aux tableaux, comme map() ou filter() par exemple ; autrement dit, il n'est pas possible d'itérer sur l'objet « arguments ». Cependant, il existe des astuces permettant de convertir ce type de collection en un tableau.

Parmi ces astuces, nous pouvons citer :

```
// astuce #1
var args = [].slice.call(arguments, 0);
// astuce #2
[].forEach.call(arguments, (x => /* expression */));
```

L'argument « this »

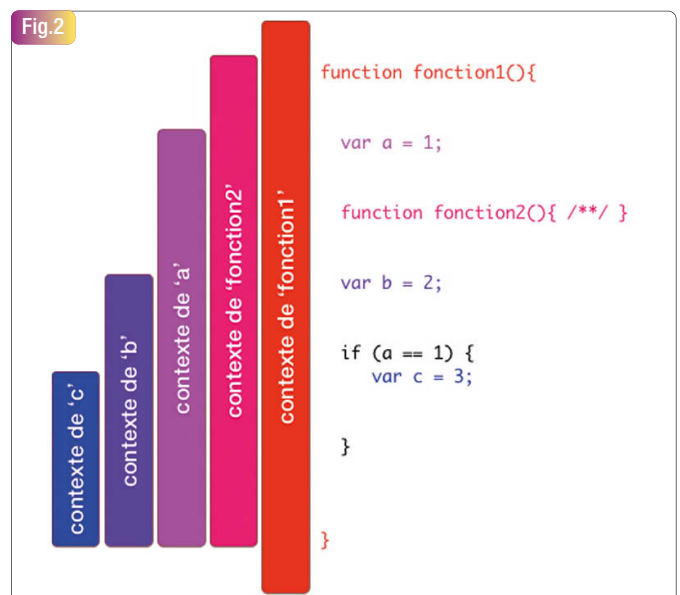
Cet argument est un peu particulier. Il représente l'objet qui est associé — de façon implicite — à l'invocation de la fonction.

Il est aussi connu sous le terme de contexte. Le terme contexte est bien connu des développeurs habitués au développement orienté objet.

Cependant, dans la majorité des cas, ces mêmes personnes pensent qu'en JavaScript, le mot clé « this » représente l'instance de la classe dans laquelle la méthode est définie. Ce n'est pas vrai !

En JavaScript, il se trouve que le contexte représenté par le paramètre « this » est conditionné par la façon dont la fonction a été invoquée, et non pas par la façon dont elle a été définie. C'est pour cela que ce paramètre est appelé le contexte d'invocation. Maintenant que nous avons bien compris ce que c'est « this » et ce qu'il représente.

Passons aux différents types d'invocations.



Invocation en tant que fonction

Ce type d'invocation est sûrement le plus utilisé. En effet, si une fonction n'est pas invoquée en tant que méthode, constructeur ou via « apply » ou « call », alors nous sommes en présence d'invocation d'une fonction. Cette invocation se produit lorsque la fonction est invoquée en utilisant les parenthèses et lorsque cette fonction n'est pas une propriété d'un objet.

Invocation en tant que méthode

Lorsqu'une fonction est une propriété d'un objet et que l'invocation se fait en appelant cette fonction, alors la fonction est invoquée en tant que méthode de cet objet. Pour illustrer cela, étudions cet exemple :

```
function foo() {
  return this;
}

// invocation en tant que fonction
foo(); // => this === window

var bar = {
  'foo': foo
};

// invocation en tant que méthode de 'bar'
bar.foo(); // => this === bar
```

Nous avons déclaré une fonction « foo » que nous invoquons de deux manières : en tant que fonction, puis en tant que méthode de l'objet « bar ». Nous constatons qu'en effet, le contexte d'invocation retourné, représenté par « this », est différent dans les deux cas. Quand invoquée en tant que fonction, « this » représente le contexte globale « window ». Lorsque la fonction est invoquée en tant que méthode, « this » représente l'objet hôte de la méthode au moment de l'invocation. Cette particularité prend tout son sens en programmation orientée objet en JavaScript.

Invocation en tant que constructeur

Un constructeur est une simple fonction, il n'a vraiment rien de spécial. La seule différence réside dans la façon dont elle va être invoquée. Pour invoquer une fonction en tant que constructeur, nous le précédon par le mot-clé « new ». Invoquer une fonction en tant que constructeur est un aspect très intéressant de JavaScript, car lorsqu'un constructeur est invoqué, plusieurs choses se produisent :

- Un nouvel objet vide est créé, le fameux paramètre « this »,
- Cet objet est passé au constructeur. Il devient donc le contexte d'invocation.

En l'absence d'une instruction « return » explicite dans le constructeur, cet objet est retourné par le constructeur. Prenons cet exemple :

```
function Foo() {
  this.bar = function() {
    return this;
  }
}

// exemple 1
var foo = new Foo();
console.log(foo); // => Foo
console.log(foo.bar() instanceof Foo); // => true

// exemple 2
var foo1 = Foo();
```

```
console.log(typeof foo1); // => undefined
console.log(typeof window.bar); // => function
```

Nous avons déclaré un constructeur « Foo » (notez la première lettre en majuscule), celui-ci déclare une propriété « bar », qui est une méthode retournant le contexte « this » (pour l'exemple). Dans un premier temps, nous invoquons ce constructeur avec le mot-clé « new » et nous avons bien un objet de type « Foo » qui a été créé, et qui possède bien une méthode « bar ». Un petit test sur le contexte « this » dans « foo », nous prouve que c'est bien une instance de « Foo ». Dans l'exemple 2, nous avons essayé de démontrer qu'invoquer un constructeur en tant que fonction, sans le mot-clé « new », ne donne pas du tout le résultat attendu. En effet, invoquer « Foo » en tant que fonction, exécute simplement le code de cette fonction, ce qui provoque la création de la propriété « bar » dans l'objet hôte — dans lequel « Foo » a été invoquée. Cet objet représenté par « this » dans « Foo » est bien évidemment « window », puisque la fonction « Foo » a été déclarée dans le contexte global de JavaScript. Jusque-là, nous avons vu que le type d'invocation d'une fonction agit directement sur le contexte d'invocation, représenté par le paramètre implicite « this ». Pour les méthodes c'est l'objet hôte ; pour les fonctions déclarées dans le contexte global c'est le « window » ; et pour les constructeurs c'est l'instance du nouvel objet créé. Maintenant, comment pouvons-nous faire si nous voulons forcer un contexte d'invocation particulier ? C'est bien grâce à « apply » et « call ».

Invocation via « apply » ou « call »

JavaScript offre une méthode simple, pour invoquer une fonction et lui spécifier explicitement un contexte d'invocation. Nous pouvons réaliser cela grâce à deux méthodes proposées par toutes les fonctions : « apply » et « call ». Pourquoi voudrions-nous faire cela ? Prenons l'exemple d'une situation que nous rencontrons régulièrement en JavaScript : les actions — ou callbacks — des événements déjà abordés précédemment. Lorsque nous déclarons une fonction de callback sur un événement, celle-ci est invoquée lorsque l'événement en question est traité par le navigateur. Cette fonction de callback est invoquée avec le contexte de l'événement, c'est un comportement par défaut du navigateur.

```
function foo() {
  console.log(this); // => this === window
  console.log(arguments); // => 'a', 'b', 'c'
}

var element = document.querySelector('#foo');
element.addEventListener('click', (e) => {
  console.log(this); // => element
  console.log(arguments); // => e === Event

  foo('a', 'b', 'c');
});
```

Comme l'illustre le code ci-dessus, le contexte d'invocation de la fonction de callback — représenté par « this » — est celui de « window » ; la callback « foo » est une fonction déclarée dans le contexte globale. Elle a été invoquée en tant que fonction. Voyons maintenant le même code mais cette fois-ci en utilisant la méthode « call ».

```
function foo() {
  console.log(this); // => this === element
  console.log(arguments); // => 'a', 'b', 'c'
}

var element = document.querySelector('#foo');
element.addEventListener('click', (e) => {
```

```
console.log(this); //=> element
console.log(arguments); //=> e === Event

foo.call(element, 'a', 'b', 'c');
});
```

En invoquant une fonction — ou méthode — avec « call », nous passons un premier paramètre qui représente le contexte d’invocation. Dans notre exemple, ce contexte est l’élément DOM qui nous intéresse. En plus du contexte d’invocation, nous avons aussi passé une liste de paramètre à la fonction « foo ».

```
function foo() {
  console.log(this); //=> this === element
  console.log(arguments); //=> a, b, c
}

var element = document.querySelector('#foo');
element.addEventListener('click', (e) => {

  console.log(this); //=> element
  console.log(arguments); //=> e === Event

  foo.apply(element, ['a', 'b', 'c']);
});
```

Avec « apply », c’est quasiment la même chose : nous passons également le contexte d’invocation en premier paramètre. Mais contrairement à « call », la méthode « apply » prend en second paramètre un tableau représentant la liste des arguments. Voici un autre exemple plus intéressant :

```
function forevery(list, callback) {
  for (var i = 0; i < list.length; i += 1) {
    callback.call(list, i);
  };
}

forevery(['a', 42, false, {}], (index) => {
  console.log('1', this); //=> ['a', 42, false, {}]
  console.log('2', this[index]); //=> 'a', 42, false, {}
  console.log('3', this[index + 1]); //=> 42, false, {}, undefined
  console.log('4', this[index - 1]); //=> undefined, 'a', 42, false
});
```

Dans cet exemple, nous avons déclaré une fonction « forevery » qui permet d’itérer sur une liste d’éléments. Cette fonction prend en second paramètre une fonction de callback qui sera invoquée pour chaque élément parcouru, avec le contexte de la liste (nous aurions pu spécifier un autre contexte), et nous donne en paramètre l’index de l’élément courant.

Quelle méthode utiliser ?

Cela dépend de votre cas d’usage, et plus précisément cela dépend de comment vous devez gérer vos paramètres. Si vous avez un ensemble de variables que vous devez passer en tant que paramètres, « call » serait idéale. Mais si vous avez déjà votre liste de variables sous forme de tableau (par exemple, arguments), « apply » est plus appropriée. Prenons par exemple la méthode Math.max(n1, n2, n3, ...) de JavaScript : cette méthode calcule le maximum d’une liste de valeur qu’elle prend en paramètre.

```
Math.max(0, 1, 2, 3, 4, 5, 6, 7, 8, 9); //=> 9
```

Supposons que vous voudriez calculer le maximum (ou le minimum) d’une suite de valeurs dont vous ne connaissez pas la longueur :

```
function max() {
```

```
  return (arguments.length === 0) ? 0 : Math.max.apply(Math, arguments);
}

max(0, 1, 2, 3, 4, 5, 6, 7, 8, 9); //=> 9
max(3, 8, 32, 2, 98); //=> 98
max(42, 0); //=> 42
max(); //=> 0
```

Dans cet exemple, je viens de déclarer une fonction nommée « max » qui calcule le maximum d’une suite de nombre passés en paramètre. S’il n’y a aucun paramètre, le résultat est zéro. Je me repose donc sur le paramètre implicite "arguments" pour réaliser cette vérification. Vous comprenez tout de suite qu’utiliser « apply » a tout son sens dans cet exemple. Veuillez noter également que j’ai passé « Math » en tant que contexte d’invocation à la méthode « Math.max() » : je n’étais pas obligé d’en spécifier un, mais c’est plus sûr comme ça !

Un mot sur « bind »

En JavaScript, il existe une autre façon de contrôler le contexte d’invocation d’une fonction. Contrairement à « call » et « apply », ce contrôle est fait au moment de la déclaration de la fonction, et non à l’invocation. Ce contrôle est rendu possible grâce à la méthode « bind(arg) » permettant de transformer le contexte d’invocation d’une fonction. Jetons un œil à l’exemple suivant :

```
var Foo = function() {
  this.counter = 0;
  this.inc = function() {
    this.counter += 1;
    console.log(this);
  };
};

var foo = new Foo();
var element = document.querySelector('#foo');

// cas #1
element.addEventListener('click', foo.inc); //=> this === element

// cas #2
element.addEventListener('click', () => foo.inc()); //=> this === element

// cas #3
element.addEventListener('click', foo.inc.bind(foo)); //=> this === foo
```

Nous avons déclaré un constructeur « Foo ». Ce constructeur possède une méthode « inc », qui permet d’incrémenter un compteur « counter ». Ensuite, nous attachons un événement sur un élément DOM (disons, un bouton par exemple). A chaque clic nous souhaitons incrémenter le compteur de « Foo ». Dans un premier temps (cas #1), nous attachons la méthode « foo.inc » en tant qu’action sur le clic. Mais là, problème ! En effet, l’attribut « counter » n’est pas résolu ; tout simplement puisque le contexte d’invocation de « foo.inc » n’est pas l’instance « foo » mais « element » (rappelez-vous l’exemple cité plus haut avec le « apply »). Pour résoudre ce problème, nous aurions pu passer par une fonction anonyme (cas #2) dans laquelle nous aurions invoqué la méthode « foo.inc() ». Mais pourquoi déclarer une fonction juste pour en invoquer une autre ? Il y a plus simple, nous utilisons la méthode « bind » en précisant « foo » comme contexte d’invocation (cas #3). Mais rappelez-vous que « bind » n’invoque pas la fonction en question, elle modifie simplement son contexte d’invocation.

Que la force soit avec vous...



Programmation fonctionnelle en JS : (Ré)Inventons ensemble la Monade !

3^e partie



Sebastien Nichele
SQLi

To have or not to have a value, that is ...?

Maintenant que nous avons vu (Programmez! 182) comment améliorer notre fonction à effet de bord, complexifions l'affaire.

Au secours, une valeur 'null' veut s'échapper de mon contexte !

Et si `imprimeRapport` était une fonction qui retournait une valeur à propager ? Revenons un peu en arrière, avant l'introduction du `ifIsNotNull`

```
function genererRapportSiNecessaire() {
  var donneesDuJour = recupererDonneesDuJourSiExisteSinonNull();

  if ( isNotNull(donneesDuJour) ) {
    return imprimeRapport(donneesDuJour);
  }
}
```

Avant de songer à l'améliorer, n'y a-t-il pas un premier problème à régler ? J'espère que ce code vous ennuie un peu ; car personnellement, j'ai un sacré souci de sémantique. Il manque un `return` dans ma fonction englobante `genererRapportSiNecessaire`, qui vaudra parfois `undefined` : or `undefined` = `void` = fonction générant un effet de bord et ne retournant JAMAIS rien. Ce qui est faux, puisque je peux parfois avoir un retour ! Un compilateur Java par exemple nous aurait immédiatement alerté sur l'absence de mot clé `return` hors de la condition. Réglons ce premier problème.

```
function genererRapportSiNecessaire() {
  var donneesDuJour = recupererDonneesDuJourSiExisteSinonNull();

  if ( isNotNull(donneesDuJour) ) {
    return imprimeRapport(donneesDuJour);
  } else {
    return null;
  }
}
```

C'est *légèrement* mieux. Je dis légèrement, car 'null', c'est très faible sémantiquement et ça force à propager beaucoup de null-checks dans les fonctions appelantes... Pour illustrer, créons une fonction `genererRapportEtDiffuser` qui va appeler `genererRapportSiNecessaire` :

```
function genererRapportEtDiffuser() {
  var resultat = genererRapportSiNecessaire();
  if (isNotNull(resultat)) {
    diffuser(resultat);
  }
}
```

```
function genererRapportSiNecessaire() {
  var donneesDuJour = recupererDonneesDuJourSiExisteSinonNull();
```

```
  if ( isNotNull(donneesDuJour) ) {
    return imprimeRapport(donneesDuJour);
  } else {
    return null;
  }
}
```

Personnellement, propager des `isNotNull()` ça ne m'enchant pas beaucoup. De plus, les `if/else` commencent à s'empiler... c'est beaucoup trop vertical ! Réintroduisons `ifIsNotNull` !

Comment combiner les fonctions pour minimiser les null-checks ?

Si je réintroduis `ifIsNotNull`, je peux combiner les conditions et les `returns` :

```
// - Avant
function genererRapportEtDiffuser() {
  var resultat = genererRapportSiNecessaire();
  if (isNotNull(resultat)) {
    diffuser(resultat);
  }
}

function genererRapportSiNecessaire() {
  var donneesDuJour = recupererDonneesDuJourSiExisteSinonNull();

  if ( isNotNull(donneesDuJour) ) {
    return imprimeRapport(donneesDuJour);
  } else {
    return null;
  }
}

// - Après
function genererRapportEtDiffuser() {
  ifIsNotNull(genererRapportSiNecessaire(), diffuser );
}
```

```
function genererRapportSiNecessaire() {
  return ifIsNotNull(recupererDonneesDuJourSiExisteSinonNull(), imprimeRapport );
}
```

```
function ifIsNotNull(x,doSomething){
  if (isNotNull(x)) {
    return doSomething(x);
  } else {
    return null;
  }
}
```

C'est mieux. Mais je vous vois déjà froncer les sourcils : "Mais il est toujours là, le 'return null' !!!" et on le propage toujours ! Patience ... Pourquoi ne pas faire évoluer notre fonction `ifIsNotNull` pour prendre **deux fonctions** en paramètres ? Faisons redescendre `diffuser` jusqu'en bas de la chaîne d'appel !

```
function genererRapportEtDiffuser() {
  genererRapportSiNecessairePuis(diffuser);
}

function genererRapportSiNecessairePuis(faireCa) {
  ifIsNotNull(recupererDonneesDuJourSiExisteSinonNull(), imprimeRapport, faireCa);
}

function ifIsNotNull(x, doSomething, thenDo){
  var intermediary = null;
  if (isNotNull(x)) {
    intermediary = doSomething(x);
    if (isNotNull(intermediary)) {
      return thenDo(intermediary);
    } else {
      return null;
    }
  } else {
    return null;
  }
}
```

WOW ! Quelqu'un suit ? Donnez-vous une tape dans le dos et offrez-vous un café. Ceci... n'est pas ... la solution.. à notre problème !

Ce que nous venons de faire porte un nom : **'continuation passing style'**. On fait disparaître les 'return' pour ne rien propager 'vers le haut' et on construit la chaîne d'appels en passant des fonctions, vers le bas. Certes, cela permet d'éviter d'avoir à propager des null-checks et permet de résoudre certaines problématiques précises en favorisant l'horizontalité du code; mais dans notre cas, cela va nous forcer à modifier continuellement la fonction en bout de chaîne (ici ifIsNotNull) pour nous adapter aux appelants. Ce qui n'est pas très en accord avec l'Open-Closed principe...

Essayons un peu différemment, avec les null-check à mi-chemin ?

```
function genererRapportEtDiffuser() {
  genererRapportSiNecessairePuis(diffuser);
}

function genererRapportSiNecessairePuis(faireCa) {
  ifIsNotNull(
    ifIsNotNull(
      recupererDonneesDuJourSiExisteSinonNull(),
      imprimeRapport
    ),
    faireCa
  );
}

function ifIsNotNull(x, doSomething){
  if (isNotNull(x)) {
    return doSomething(x);
  } else {
    return null;
  }
}
```

En remontant le null-check dans la fonction intermédiaire, on ne touche plus à ifIsNotNull... mais on empile au niveau au-dessus, ce qui n'est pas

forcément beaucoup mieux, même si le code est 'horizontal' et que l'on évite les variables intermédiaires. De plus, on est toujours obligé de passer la fonction diffuser à genererRapportSiNecessairePuis, ce qui n'est pas, à l'origine, de sa responsabilité. Ordre supérieur, soit, mais pas à tous les niveaux d'abstractions et en dépit de la séparation des responsabilités. Notre cause est-elle perdue ? Doit-on se résigner à utiliser des if ?

Vers une solution ?

Quelle relation les fonctions recupererDonneesDuJourSiExisteSinonNull, imprimeRapport et diffuser ont-elles ? Comme tout à l'heure : c'est idéalement de la composition de fonction

```
diffuser(imprimeRapport(recupererDonneesDuJourSiExisteSinonNull()))
```

Comparez avec notre code (rappelé ci-dessous) et remarquez comment nous avons inséré isIsNotNull dans la composition :

```
ifIsNotNull(
  ifIsNotNull(
    recupererDonneesDuJourSiExisteSinonNull(),
    imprimeRapport
  ),
  diffuser
);
```

C'est de la composition, mais 'renversée' ! Finalement, on ne compose plus directement nos fonctions, mais on compose (sous certaines conditions) la (valeur contenue dans les 'contexte de calculs') de ces fonctions – c'est le rôle de 'ifIsNotNull'. Et si pour améliorer ça, mon contexte d'exécution conditionnel, au lieu de me retourner une valeur brute, me renvoyait un autre contexte d'exécution conditionnel ? Et si la fonction ifIsNotNull me renvoyait une nouvelle fonction ifIsNotNull plutôt que la dernière valeur calculée ? Je n'aurais plus besoin de l'appeler explicitement ! Voyons ce que ça donne :

```
function genererRapportEtDiffuser() {
  genererRapportSiNecessaire() (diffuser); //attention, appel 'curryfié' !
}

function genererRapportSiNecessaire() {
  return ifIsNotNull(recupererDonneesDuJourSiExisteSinonNull(), imprimeRapport);
}

function ifIsNotNull(x, doSomething) {
  var localResult = null;
  if (isNotNull(x)) {
    localResult = doSomething(x);
  }

  return ifIsNotNull ...; // ?? wrong, je souhaite conserver la valeur de localResult dans la chaîne !
}
```

Aïe ! Les choses semblent bien parties, mais au moment de renvoyer ifIsNotNull, comment conserver localResult qui doit servir à alimenter justement cette fonction avec le résultat précédent ? Celui qui aura répondu 'Curryfication' et 'closure' aura raison : il faut effectuer une application partielle de la fonction pour capturer localResult et renvoyer la fonction partiellement appliquée. Avant d'appliquer partiellement, il faut donc curryfier, et pour ça, lodash nous fournit tout ce qu'il faut :

```
_.curry(mafonction);
```

Avec ça, au lieu de faire :

```
ifIsNotNull(x, uneFunction);
```

Je peux faire :

```
_.curry(ifIsNotNull)(x)(uneFunction)
```

Ou

```
var partiallyApplied = _.curry(ifIsNotNull)(x);
partiallyApplied(uneFunction);
```

Ce qui va nous donner au final :

```
function genererRapportEtDiffuser() {
  genererRapportSiNecessaire()(diffuser);
}

function genererRapportSiNecessaire() {
  return ifIsNotNull(
    recupererDonneesDuJourSiExisteSinonNull(),
    imprimeRapport
  );
}

function ifIsNotNull(x, doSomething) {
  var localResult = null;
  if (isNotNull(x)) {
    localResult = doSomething(x);
  }
  return _.curry(ifIsNotNull)(localResult); // renvoie une fonction
  // ifIsNotNull partiellement
  // appliquée
}
```

Ça a l'air très très bien ! Grâce à la curryfication et aux fonctions d'ordres supérieurs, je peux jouer sur la chaînabilité des appels de fonctions pour propager un contexte conditionnel d'évaluation de fonction - `ifIsNotNull` - afin d'obtenir un code concis, expressif et ouvert aux extensions.

Mais ... (car il y a toujours un 'mais'...)

Cette solution semble très bien, certes, mais elle n'est pas exempte de défauts... Désolé si vous aviez cru en avoir terminé ! Actuellement, `genererRapportSiNecessaire` ne retourne rien. Mais que se passerait-il avec le code suivant ?

```
function genererRapportEtDiffuser() {
  var contexte = genererRapportSiNecessaire()(diffuser);
}

function genererRapportSiNecessaire() {
  return ifIsNotNull(
    recupererDonneesDuJourSiExisteSinonNull(),
    imprimeRapport
  );
}
```

La variable `contexte` serait une fonction `ifNotNull` partiellement appliquée, soit un contexte d'évaluation de fonction ayant capturé la valeur de la dernière opération effectuée. Je pourrais facilement faire exécuter une autre fonction par ce contexte, fonction qui utiliserait la dernière valeur calculée. Mais que faire si je souhaitais simplement récupérer cette valeur ? Ho, je pourrais tricher, comme ceci :

```
function genererRapportEtDiffuser() {
  var fromContexte = null;

  genererRapportSiNecessaire()(diffuser)(capture);

  // je peux utiliser la valeur de fromContexte maintenant

  function capture(lastValueInContext){
    fromContexte = lastValueInContext;
  }
}
```

En faisant exécuter la fonction `capture`, je récupère la dernière valeur calculée dans le contexte et la stocke dans une variable capturée par la closure pour la récupérer.

Soit : un horrible hack, nécessitant d'utiliser une variable et une fonction dont le seul rôle est d'aller 'pêcher' une valeur dans un contexte autre. Très inélégant et pas du tout fonctionnel...

Si seulement la fonction `ifIsNotNull` me proposait une fonction pour 'extraire' la dernière valeur dans le contexte !

Oui, je parle bien d'une fonction d'un objet fonction... les fonctions étant des objets comme les autres, elles possèdent des méthodes.

Malheureusement, la consultation de la norme EcmaScript concernant le type 'Function' ne me propose pas de fonction 'getLastValueComputedInClosure'...

Les fonctions, bien que capables :

- De porter la notion de contexte de calcul isolé,
- D'être appelées de différentes façons grâce à la curryfication,
- De se combiner grâce aux mécaniques d'ordres supérieurs.

Présentent tout de même des limites ; elles ne sont pas suffisantes pour traiter élégamment de tous les problèmes de programmation dans le paradigme fonctionnel.

Nous aurait-on menti ?

Soudain, un cousin du monde objet s'invite sans prévenir...

Nous avons donc créé notre propre contexte d'évaluation conditionnelle en la fonction `ifIsNotNull`. Cette abstraction est bonne. Elle nous a permis d'horizontaliser des contextes d'évaluation temporaires - les blocs `if` -et d'en composer les contenus dynamiquement sans avoir recours à des variables intermédiaires. Elle semble cependant manquer de flexibilité et de capacité à transcender son statut de 'fonction en tant que contexte d'exécution'. Il est temps de faire appel à un concept plus étendu, celui de **Type Algébrique de Données** (ADT, en anglais). Pour faire très très simple : c'est en gros, la notion de classe en langage objet, mais à la sauce fonctionnelle. 'Créons' le type 'Function', pour illustrer (attention, pseudo-code objet qui ne correspond pas à une syntaxe JS) :

```
/* A 'Function' ADT */
Class Function {
  /* can be executed with a variable number of arguments */
  apply(args ...)
}
```


Il faut comprendre : « une instance de fonction peut être exécutée (apply), avec un nombre variable d'arguments ».

```
var func = new Function();
func.apply(1, 'toto');
```

Créons maintenant un 'type' `IfIsNotNull`. Il faut que celui-ci me permette d'exécuter une fonction sur une valeur que je lui passe, me renvoie un objet de type `IfIsNotNull` pour que je puisse bénéficier du **chaînage de fonction** à exécution conditionnelle, et que je puisse à tout moment **recupérer la dernière valeur calculée** :

```
Class IfIsNotNull {
  PartiallyAppliedIfIsNotNull : apply(x, doSomething);
}

Class PartiallyAppliedIfIsNotNull {
  PartiallyAppliedIfIsNotNull : apply(doSomething);
  getValueInside();
}
```

Je ne fais que transcrire en pseudo-objet ce que nous avons mis en œuvre précédemment, la récupération de dernière valeur en plus ! Bien. Notre 'type' `IfIsNotNull` permet d'appliquer une fonction à `x`, renvoie un `PartiallyAppliedIfIsNotNull` qui me permet de récupérer la dernière valeur calculée précédemment ou d'appliquer une autre fonction sur son contenu.

Je vous laisse retrouver par vous-même pourquoi j'ai créé deux types et non un seul ;) Je n'aime pas trop avoir deux types, justement... Qu'est-ce qui distingue ces deux types, d'ailleurs ?

Le fait que l'un contient une valeur et l'autre pas. On peut unifier les deux en ajoutant simplement un **constructeur** et en simplifiant `apply` !

```
Class IfIsNotNull {
  IfIsNotNull(x); // constructeur, injecte la valeur initiale
  IfIsNotNull : apply(doSomething); // travaille sur x !
  getValue();
}
```

Ma foi, cela m'a l'air très bien. Essayons en Javascript ?

```
function IfIsNotNull(x){
  this.x = x;
  this.apply = function(f){
    return new IfIsNotNull(
      (this.x !== null) ? f(x) : null
    );
  };
  this.getValue = function(){
    return this.x;
  };
}
```

Ce pseudo code correspond théoriquement parlant à la façon dont nous avons utilisé la fonction `IfIsNotNull` ; elle prenait deux paramètres, le premier pour injecter une valeur dans le contexte, le deuxième pour indiquer quelle fonction exécuter sur la valeur. Nous avons dans notre type ici présent scindé cette opération en deux.

Mise en œuvre en Javascript dans notre exemple fil rouge :

```
function genererRapportEtDiffuser() {
  var finalValue = genererRapportSiNecessaire().apply(diffuser).getValue();
}

function genererRapportSiNecessaire() {
  return new IfIsNotNull(recupererDonneesDuJourSiExisteSinonNull()).apply(imprimeRapport);
}
```

Cette fois-ci, c'est la bonne. Plus de if apparent. Un chaînage de fonctions qui n'explosera pas à la mode `NullPointerException`... C'est certes un tout petit peu plus verbeux de par le fait que Javascript nous force à écrire l'appel à `apply` ; dans d'autres langages, 'apply' serait compris comme un alias nommé d'invocation de fonction, et il ne serait pas utile de l'écrire.

"What ? Douzes pages de fumisteries théoriques pour me ressortir le concept de classe et me servir, au final, un *builder-like* ? Au bûcher !" Attendez ! Rangez les fourches et les torches !

Si vous avez 'compris' l'article, et suivi le cheminement mental qui nous a amené à 're' découvrir notre type `IfIsNotNull`, alors vous venez avec moi de 'réinventer' ce que l'on appelle pompeusement un 'type Monadique' (ou 'Monad' par abus de langage).

Un type monadique n'est autre - pour nous autres programmeurs - qu'une 'sorte de classe', qui une fois instanciée offre la possibilité de faire exécuter une chaîne indéterminée de fonctions sur une valeur 'isolée' du monde (le fameux contexte de calcul !), avec la possibilité de récupérer la valeur contenue dans le contexte à n'importe quel moment. Mais pas seulement ! Le concept de 'boîte' / 'contexte' contenant une valeur sur laquelle agir peut se décliner en de très nombreuses variantes, et les types monadiques sont légions.

On trouve des types monadiques dans tous les langages fonctionnels, inclus dans l'API de base du langage lui-même (haskell, Scala...) ou dans certaines bibliothèques complémentaires (Guava, FunctionalJ...).

Quelques exemples de types monadiques 'basiques', tous langages confondus :

- Peut contenir zéro ou une valeur : 'Maybe' / 'Option' => l'équivalent officiel de notre 'IfIsNotNull' ;)
- Peut contenir plusieurs valeurs : 'List' (oui oui, une collection est un type monadique...)
- Pourra contenir une valeur, plus tard : 'Future'
- Contient 'des trucs du DOM' : 'jQuery' (si si !)

Tout comme la fonction est l'élément de base pour écrire ses programmes, le type monadique est l'élément de base pour construire des abstractions de plus haut niveau et des 'designs patterns' fonctionnels.

Conclusion

J'espère que vous aurez tenu bon. J'espère aussi avoir réussi ce challenge : écrire un tutorial 'éclairant' permettant de (commencer à) comprendre pas à pas, ce qu'est une monade (sans un doctorat en mathématique) à travers une problématique de code concrète, en passant par l'intuition de "contexte d'exécution".

J'aurais sûrement pris des raccourcis et effectué quelques approximations ; mais je ne pense pas avoir trahi l'essence de la bête. Après tout, *un monade n'est qu'un monoïde sur la catégorie des endofoncteur* !

Dans un futur article, nous reviendrons sur la puissance des fonctions combinatoires et nous essaierons, là encore, de nous approprier le concept de façon la plus pragmatique possible.



Material Design :

Comment intégrer le nouveau design d'Android à vos applications ?

Avec l'arrivée récente de la nouvelle version d'Android (Lollipop, 5.0), Google marque un tournant en présentant les principes du Material Design.



Alexandre Masciulli
Design and development engineer, Genymobile
Christophe Vilet
Graphic designer, Genymobile

Approche plus minimaliste permettant d'améliorer nos interfaces et d'optimiser l'expérience utilisateur sous différents aspects comme la création d'animations, la profondeur de champ, l'utilisation d'une palette de couleurs vives aux nombreuses nuances, etc.

Plus qu'un "style", le Material Design est un langage visuel. Il se présente comme une métaphore de la feuille de papier faisant écho à la manière dont l'homme conçoit l'univers qui l'entoure. La lumière, la profondeur, la perspective et le mouvement, sont autant de notions de perceptions que le Material Design souhaite restituer visuellement. Le but étant de donner du sens à une action afin de guider l'utilisateur de manière intuitive dans son usage quotidien d'une application.

Au-delà de son aspect visuel, le Material Design a bien d'autres enjeux. Dans un souci d'harmonie et afin de proposer une expérience utilisateur novatrice, Google propose pour ses applications un système graphique adaptable aux différents supports (Web, mobiles, tablettes), mais également aux différentes plateformes (Android, iOS, Windows Phone, etc.). Le Material Design relève le défi d'uniformiser l'expérience utilisateur et croise ainsi des problématiques Cross Platform liées à la fois au Web et à l'appli-catif. Il s'agit ici de voir comment intégrer techniquement les guidelines du Material Design dans la conception d'une application.

1 - POINT DE DÉPART : LE THÈME MATERIAL

"L'intégration du thème à votre application"

Du côté du développement, le point de départ pour intégrer le Material Design à votre application est le thème système Theme.Material (ou Theme.Material.Light), fourni par le SDK Android depuis la version 5.0. Pour cela, faites hériter le thème de votre application de ce thème système grâce à l'attribut XML "parent".

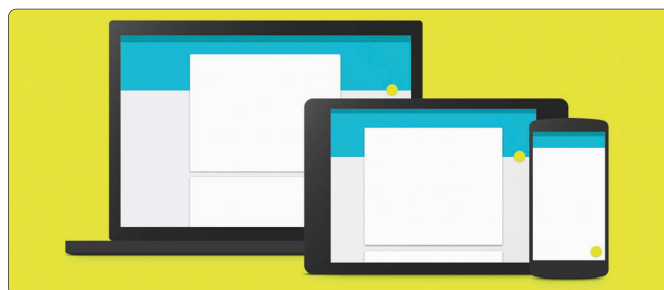
Si vous souhaitez rendre votre application compatible avec les versions plus anciennes d'Android, vous pouvez intégrer ce thème grâce à la librairie de compatibilité AppCompat en version 21, en utilisant le thème Theme.AppCompat (ou Theme.AppCompat.Light).

```
<resources>
<style name="AppTheme" parent="android:Theme.Material"/>
</resources>
```

2 - LA PALETTE DE COULEURS

"Plus de diversité pour une meilleure perception"

S'il fallait décrire la palette de couleurs, les mots qui viendraient à l'esprit seraient "vive et acidulée". Les tons pastel ou les effets de transparence n'ont pas leur place dans la conception du Material Design. La palette est riche et variée, offrant aux utilisateurs plus de 500 nuances.



<http://www.google.com/design/spec/material-design>

Depuis l'introduction du thème Material Design, l'utilisation des couleurs ne se fait plus forcément par l'application directe d'une ressource à une vue (comme l'ActionBar par exemple). Il est en effet possible de définir, dans votre thème qui hérite du thème Material, les couleurs principales de votre palette que le système se chargera d'appliquer à divers endroits de votre application. Par exemple, la couleur spécifiée par l'attribut "colorPrimary" sera la couleur principale de votre application et sera appliquée, entre autres, à votre ActionBar. L'attribut "colorPrimaryDark" représentera une variante plus sombre de votre couleur principale et sera par exemple appliquée à votre barre de statut. L'attribut "colorAccent" quant à lui, sera utilisé pour teinter les widgets comme les boutons ou les cases à cocher Fig.1. Les guidelines de Google sont très précises à ce sujet et le site (www.materialpalette.com) propose de tester l'association des couleurs primaires et secondaires à travers une quinzaine de nuances. Il existe d'autres attributs comme "colorControlNormal" ou "colorControlActivated" qui permettent de pousser encore plus loin la personnalisation de votre application Fig.2.

```
<style name="AppTheme" parent="Theme.Material">
<!-- couleur principale de l'app, utilisée pour l'ActionBar -->
<item name="colorPrimary">@color/main_color</item>

<!-- version sombre de la couleur principale, pour la barre de statut -->
<item name="colorPrimaryDark">@color/main_color_dark</item>

<!-- couleur par défaut des widgets (boutons, switches, etc.) -->
<item name="colorAccent">@color/accnt</item>
```



<http://www.google.com/design/spec/material-design>

```
<!-- Vous pouvez pousser plus loin la personnalisation en utilisant les attributs colorControlNormal,
colorControlActivated, colorControlHighlight et colorSwitchThumbNormal -->
</style>
```

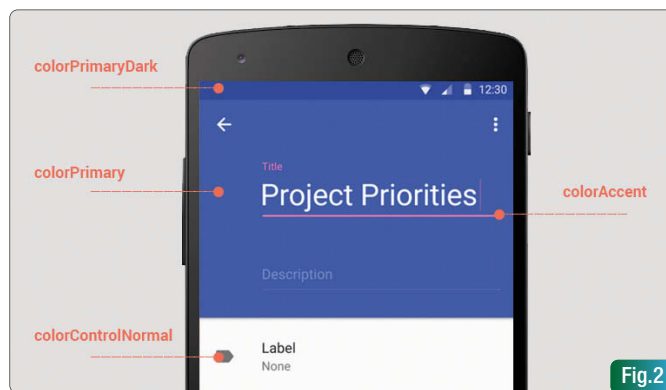
3- LES OMBRES

“Hiérarchiser l’information intelligemment”

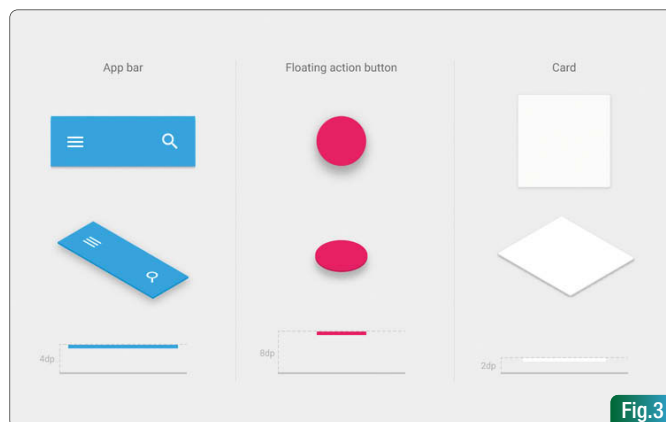
Toujours dans l’intention de guider au mieux l’utilisateur dans son parcours, Google souligne le rôle primordial des ombres dans l’organisation de l’interface. La perception que nous avons des objets qui nous entourent passe en partie par les ombres. Loin d’être un effet de style, elles délimitent de façon précise une section par rapport à une autre. Apportant du relief aux éléments, l’ombre indique intuitivement ce qui est proche et ce qui est plus lointain, ce qui doit être mis en exergue et ce qui ne doit pas l’être **Fig.3**. La présence d’ombres légères s’avère être très utile pour limiter les erreurs de compréhension lors de la navigation, et permet donc de mieux se repérer lors d’un effet de transition (que l’on abordera un peu plus tard). L’intensité de ces ombres joue un rôle important dans la définition hiérarchique des éléments entre eux **Fig.4**.

Pour implémenter les ombres de manière rapide et efficace, Google met à notre disposition deux nouveautés : l’API CardView et les ombres sur les vues. Une CardView est un composant de vue héritant de FrameLayout et permettant de réaliser des cards sans vous soucier de gérer l’aspect de celles-ci :

```
<android.support.v7.widget.CardView
xmlns:card_view="http://schemas.android.com/apk/res-auto"
android:layout_width="200dp"
android:layout_height="200dp"
<!-- cardCornerRadius vous permet de gérer le rayon des coins arrondis -->
card_view:cardCornerRadius="4dp">
```



<http://www.google.com/design/spec/material-design>



<http://www.google.com/design/spec/material-design>

```
<!-- insérez ici les vues contenues dans la card -->
</android.support.v7.widget.CardView>
```

En ce qui concerne les ombres elles-mêmes, il est maintenant possible d’en générer facilement, en renseignant l’attribut d’élévation des vues. Des ombres apparaîtront sur les contours de la vue avec un déport qui dépendra de la hauteur renseignée :

```
<TextView
android:elevation="2dp" />
```

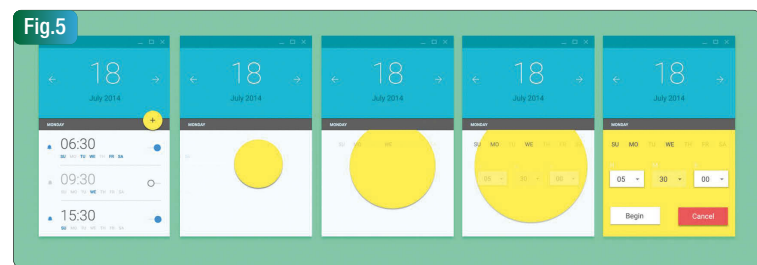
4 - ANIMATIONS ET TRANSITIONS

“Le mouvement donne du sens à l’action”

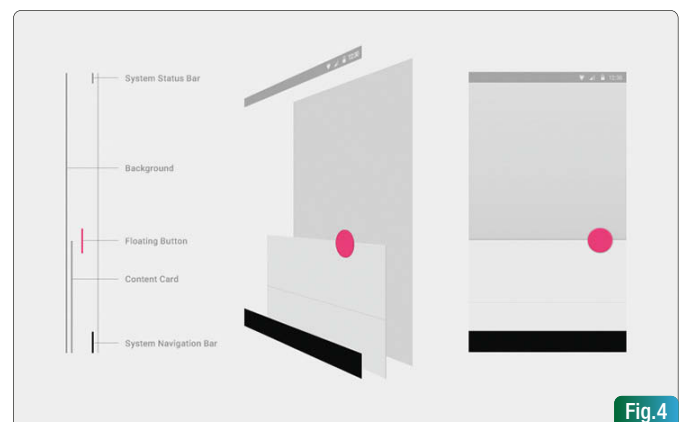
Véritable allégorie de la feuille de papier, il est amusant d’observer comment réagit cette *matière numérique* au toucher. L’ambition est alors de **matérialiser** les actions de l’utilisateur afin de les rendre les plus réalistes possible. L’action trouve donc son sens dans l’animation qui en découle. C’est ainsi qu’intervient la notion de “**mouvement**” évoquée plus tôt. Une action effectuée au toucher par l’utilisateur déclenche une animation, de laquelle émergent toutes sortes d’effets de transition entre les éléments de l’interface. Les transitions (*agrandissement, étirements, positionnement, etc.*) s’exécutent de manière fluide et cohérente. Le Material Design ajoute ce concept interactif à l’expérience utilisateur, offrant ainsi de nouvelles perspectives dans la conception d’une application **Fig.5**.

Pour implémenter les nouveaux patterns d’animations préconisés par le Material Design, le SDK met à notre disposition plusieurs nouveautés : de nouvelles API pour animer vos vues, ainsi que les Activity transitions. L’animation des vues peut désormais se faire facilement à une multitude de niveaux : touch feedback (réponse au toucher), apparition/disparition, changement d’état (sélection/désélection d’une checkbox par exemple), listes d’éléments, etc.

Par exemple, il est maintenant possible d’utiliser une animation prédéfinie pour faire apparaître la vue suivant un cercle qui révélera la vue en s’agrandissant. Cette animation est utilisée de la manière suivante :



Concept d’une animation de Ehsan Rahimi.



<http://www.greenmellenmedia.com/googles-new-material-design/>


```
View view = findViewById(R.id.my_view);

// on récupère les coordonnées du centre de la vue
int cx = (view.getLeft() + view.getRight()) / 2;
int cy = (view.getTop() + view.getBottom()) / 2;

// on récupère le rayon final de la vue
int finalRadius = Math.max(view.getWidth(), view.getHeight());

// on crée l'Animator avec un rayon initial de 0
Animator anim =
    ViewAnimationUtils.createCircularReveal(view, cx, cy, 0, finalRadius);

// on fait apparaître la vue et on démarre l'animation
view.setVisibility(View.VISIBLE);
anim.start();
```

Pour animer l'état d'une vue, on peut utiliser un selector comme ce qui existait auparavant pour changer l'aspect de la vue. La différence réside dans la possibilité d'animer le changement d'état. Ici, on choisit d'animer la propriété "translationZ" d'une vue lorsque celle-ci est pressée. Elle revient à son état initial lorsqu'on la relâche :

```
<!-- on anime ici l'attribut translationZ de la vue -->
<selector xmlns:android="http://schemas.android.com/apk/res/android">
  <item android:state_pressed="true">
    <set>
      <objectAnimator android:propertyName="translationZ"
        android:duration="200"
        android:valueTo="2dp"
        android:valueType="floatType"/>
    </set>
  </item>
  <item android:state_enabled="true"
    android:state_pressed="false"
    android:state_focused="true">
    <set>
      <objectAnimator android:propertyName="translationZ"
        android:duration="100"
        android:valueTo="0"
        android:valueType="floatType"/>
    </set>
  </item>
</selector>
```

Les nouvelles animations ne concernent pas seulement les vues, mais aussi les transitions à l'ouverture et à la fermeture d'une Activity. Ces transitions n'étaient jusqu'à maintenant pas personnalisables, on pouvait alors simplement choisir parmi quelques transitions basiques du SDK. De nouvelles transitions basiques ont été introduites : "explode" (explosion), "fade" (fendu) et "slide" (glissement).

À partir d'Android Lollipop (5.0), il est surtout possible d'utiliser des transitions personnalisées et des transitions à éléments partagés. Les transitions personnalisées vous permettent de créer une transition vous-même, en étendant la classe Transition.

Les transitions à éléments partagés sont quant à elles un nouveau moyen fourni par le SDK pour effectuer une transition entre deux Activity ayant une vue commune.

Pour créer une telle transition, il faut au préalable définir une ressource XML la décrivant. Pour cela, plusieurs animations sont mises à disposition par le système : changeBounds, changeClipBounds, changeTransform et changeImageTransform : Fig.6.

```
<!-- res/transition/image_transform.xml -->
<transitionSet xmlns:android="http://schemas.android.com/apk/res/android">
  <changeImageTransform/>
</transitionSet>
```

Afin que le système anime la vue initiale pour la faire évoluer vers la vue finale, utilisez l'attribut android:transitionName avec la valeur de votre choix sur la vue qui sera animée dans les deux Activity.

```
<ImageView
  android:id="@+id/image"
  android:layout_width="wrap_content"
  android:layout_height="wrap_content"
  android:transitionName="imagetransform"/>
```

Il vous faut également activer les transitions personnalisées dans votre thème et spécifier les ressources XML des transitions à appliquer en ouverture et fermeture d'Activity :

```
<style name="AppTheme" parent="android:Theme.Material">
  <!-- activation des transitions personnalisées -->
  <item name="android:windowContentTransitions">true</item>

  <!-- spécification des transitions à éléments partagés -->
  <item name="android:windowSharedElementEnterTransition">
```

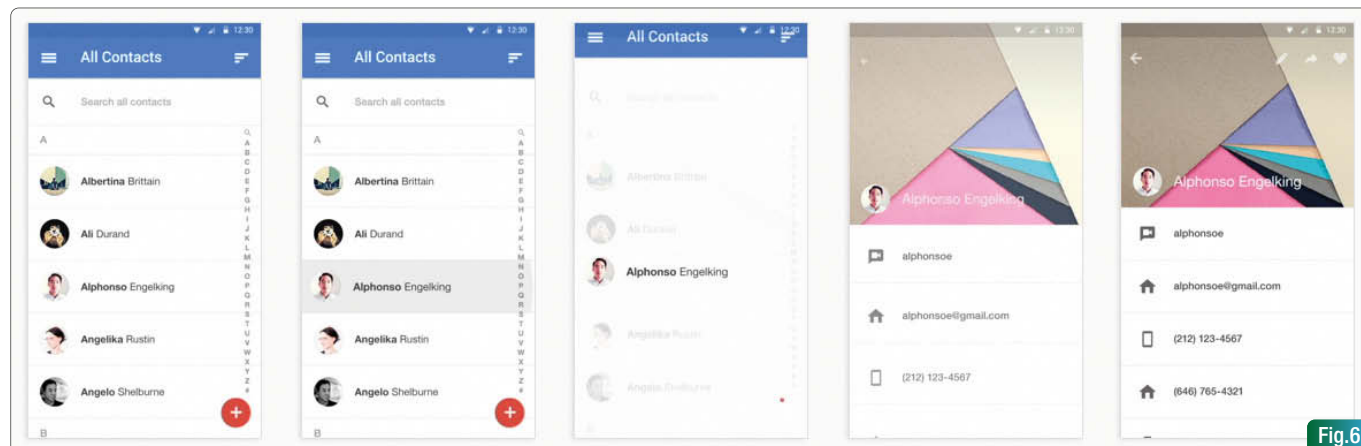


Fig.6

```
@transition/image_transform</item>
<item name="android:windowSharedElementExitTransition">
@transition/image_transform</item>
</style>
```

Enfin, lorsque vous lancez votre Activity, vous devez spécifier la vue à animer ainsi que le nom de la transition que vous avez renseigné avec l'attribut `android:transitionName` :

```
View view = findViewById(R.id.image);
Intent intent = new Intent(this, Activity2.class);
ActivityOptions options = ActivityOptions
    .makeSceneTransitionAnimation(this, view, "imagetransform");
startActivity(intent, options.toBundle());
```

Vous devriez maintenant pouvoir ouvrir la nouvelle Activity en observant votre vue se déplacer et se redimensionner pour prendre sa place finale. Pour fermer l'Activity, notez que l'effet inverse est paramétrable pour un coût minime : appeler `finishAfterTransition()` au lieu de `finish()`.

5 - LES FLOATING ACTION BUTTONS

“Le tout-en-un à portée de main”

La nouveauté dans la navigation au sein d'une application concerne un bouton en particulier. Il s'agit du bouton flottant appelé *floating action button* représenté par une icône ronde rattachée à une zone précise de l'interface (*pied de page, barre d'application, etc.*). Il est rapidement identifiable par le choix de sa couleur, qui diffère généralement de la couleur primaire de l'application, et par son ombre portée donnant la sensation qu'il se détache du reste de l'interface graphique. Avec des caractéristiques visuelles telle que l'animation, le relief, la couleur, le *floating action button* est le bouton de *référence* pour accéder aux éléments clés d'une application. Par son positionnement facile d'accès, il devient un raccourci que l'utilisateur a toujours à portée de main.

L'existence de ce bouton amène à repenser l'application afin d'identifier les actions à mettre en avant et la manière de les représenter [Fig.7](#).

Après avoir défini l'emplacement, la taille, la couleur de votre bouton, ainsi que l'animation qui en découle, vous pouvez passer à l'implémentation. Le mieux reste d'utiliser un `ImageButton` classique, qu'il vous faudra modifier de manière à ce qu'il apparaisse sous la forme d'un disque. L'image que vous attribuerez à votre bouton sera l'icône de votre *floating action button*. Superposez-le ensuite à votre `FrameLayout` ou `RelativeLayout`.

Il existe une alternative à l'utilisation directe de `ImageButton` : la librairie `FloatingActionButton` (<https://github.com/makovkastar/FloatingActionButton>).

Très simple d'utilisation, elle permet entre autres de créer directement un

tel bouton en XML, ainsi que de spécifier les couleurs (états relâché, pressé et ripple) :

```
<com.melnykov.fab.FloatingActionButton
    android:id="@+id/fab"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_gravity="bottom|right"
    android:layout_margin="16dp"
    android:src="@drawable/ic_action_content_new"
    fab:fab_colorNormal="@color/primary"
    fab:fab_colorPressed="@color/primary_pressed"
    fab:fab_colorRipple="@color/ripple" />
```

6 - LES GRILLES DÉCALÉES

“Vers un système modulaire”

Plus flexibles, plus malléables, les grilles proposées par le Material Design offrent des possibilités de mise en page multiples permettant une plus grande souplesse et une plus grande créativité. Les grilles permettent d'agencer des éléments au sein de l'interface, chacun d'entre eux devenant indépendant mais réagissant aux mouvements des autres afin de se positionner correctement. La diversité des supports, leur résolution, les animations, modifient l'espace. La grille doit alors être la plus modulaire et adaptable possible [Fig.8](#).

Néanmoins, le véritable changement dans la conception de ces grilles concerne surtout leur faisabilité technique, nouvellement améliorée.

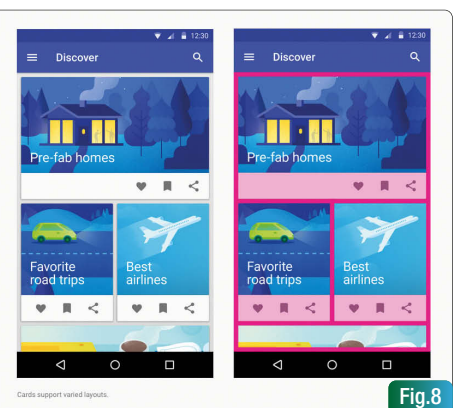
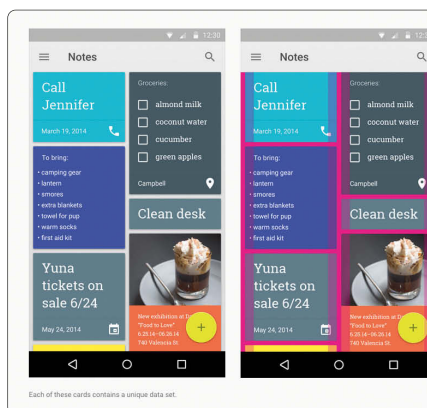
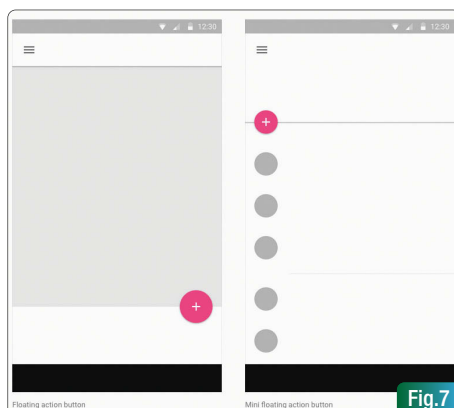
Avant Lollipop, le pattern des grilles décalées représentait un certain challenge si vous souhaitiez en intégrer un dans votre application. Désormais, il est possible de créer facilement des grilles décalées et d'autres types d'affichage de vues multiples grâce à `RecyclerView`. `RecyclerView` est une sorte de généralisation de `ListView` et `GridView`, qui intègre déjà le recyclage des vues et le pattern `View Holder`. Elle permet surtout d'utiliser un `LayoutManager`, une classe chargée de gérer la mesure et le positionnement des vues dans votre `RecyclerView`.

Vous pouvez ainsi utiliser un des `LayoutManager` disponibles par défaut dans l'API, ou encore créer votre propre `LayoutManager` en héritant de la classe. Ainsi, pour créer notre grille décalée, il suffit d'utiliser un des `LayoutManager` par défaut : `StaggeredGridLayoutManager`. Pour le reste, l'API de `RecyclerView` reste proche de celle de feu `ListView`.

```
//On récupère le RecyclerView
```

```
mRecyclerView = (RecyclerView) findViewById(R.id.my_recycler_view);
```

```
// création de le LayoutManager
```



```
mLayoutManager = new StaggeredGridLayoutManager(this);
mRecyclerView.setLayoutManager(mLayoutManager);
```

```
// création de l'Adapter traditionnel
mAdapter = new CustomAdapter(myDataset);
mRecyclerView.setAdapter(mAdapter);
```

7 - TOOLBAR ET NAVIGATION DRAWER

“Un kit modulable pour mieux naviguer”

Android Lollipop (5.0) introduit un nouveau widget, appelé Toolbar, qui peut être vu comme une généralisation de l'ActionBar utilisée massivement depuis l'introduction du thème Holo dans Android Honeycomb (3.0).

Une Toolbar est donc semblable à une ActionBar, à la différence qu'elle peut être partie intégrante de votre layout XML ou peut être créée en Java. Le comportement classique de l'ActionBar, comme la création des menus, peut ensuite être lié à votre Toolbar par un appel à la nouvelle méthode `setActionBar(Toolbar toolbar)` de la classe Activity. L'utilisation d'une Toolbar vous permet plus de libertés que l'ActionBar traditionnelle, puisque vous pouvez librement y ajouter des vues, la dimensionner comme vous le souhaitez et en utiliser plusieurs [Fig.9](#). Pour utiliser une Toolbar, déclarez-la simplement dans votre layout de la même manière que vous le feriez pour une autre vue :

```
<Toolbar
    android:id="@+id/toolbar"
    android:layout_height="wrap_content"
    android:layout_width="match_parent"
    android:minHeight="?attr/actionBarSize"
    android:background="?attr/colorPrimary" />
```

Ensuite, vous pourrez récupérer cette barre de manière classique dans votre Activity, et la définir comme ActionBar :

```
@Override
public void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_main);
    Toolbar toolbar = (Toolbar) findViewById(R.id.toolbar);
    setActionBar(toolbar);
}
```

ToolBar permet aussi d'utiliser de nouveaux composants visuels, comme le Navigation Drawer amélioré. Le Navigation Drawer est un pattern déjà utilisé dans certaines applications avant Lollipop. Utilisé en complément d'une Toolbar, il permet de faire figurer une animation au niveau de la touche de retour de la barre, en haut à gauche de l'écran. Lorsque le drawer s'ouvre, l'icône de menu à trois traits se transforme en flèche de retour [Fig.10](#).

Notez enfin que depuis Android 5.0, l'utilisation d'onglets tels qu'on les connaissait jusqu'à maintenant (en utilisant `setNavigationMode(NAVIGATION_MODE_TABS)`) est dépréciée. Pour un effet similaire, vous pouvez utiliser le pattern `SlidingTabs`, disponible dans les samples fournis par Google.

Le Material Design est un projet ambitieux d'un point de vue graphique, ergonomique et technologique. Il propose des réponses tangibles aux nombreuses problématiques ergonomiques des interfaces auxquelles sont confrontés les utilisateurs. Avec ses outils “clés en main”, il permet de concevoir des applications plus intuitives et adaptables.

Même si les langages visuels évoluent, les outils mis à la disposition des développeurs pour implanter le Material Design dans les applications Android permettent une prise en main rapide des nouveaux composants et une transition simplifiée.

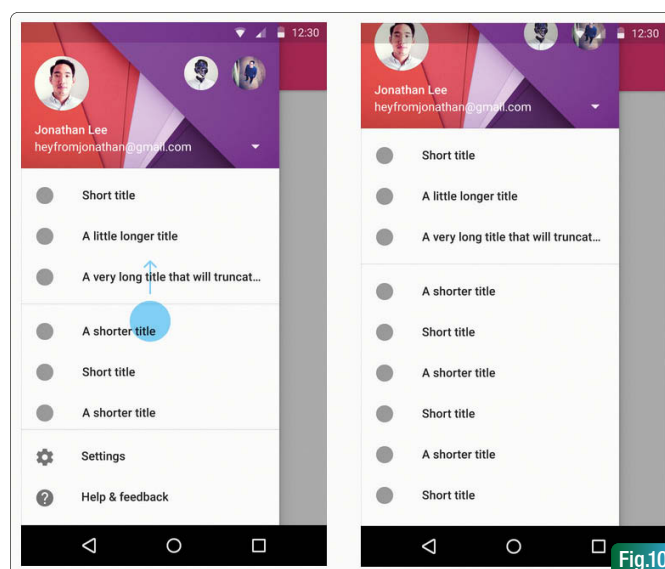


Fig.10

<http://www.google.com/design/spec/material-design>

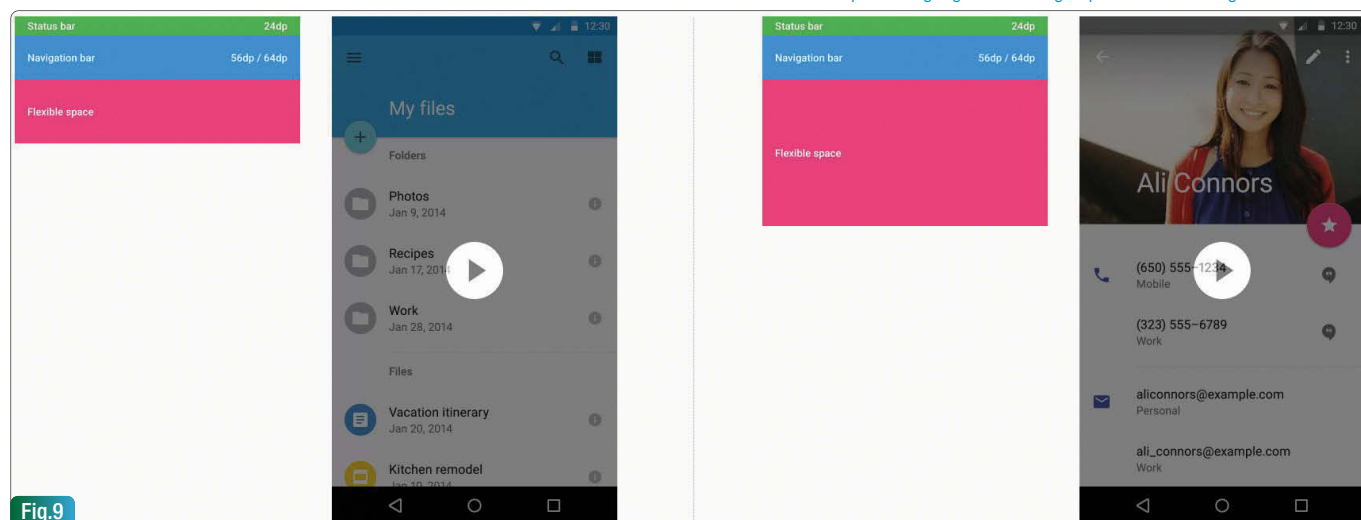


Fig.9

<http://www.google.com/design/spec/material-design>

Digital Wardrobe : Numérisez vos propres vêtements

Digital Wardrobe, comme son nom le laisse deviner, est une cabine d'essayage virtuelle. L'idée m'est venue d'un « problème » quotidien : je perdais du temps à choisir une tenue en vidant mon armoire avant de sortir. J'ai donc cherché une solution simple, à utiliser chez soi : numériser ses vêtements sur son ordinateur, et les essayer à l'aide d'une caméra.



Caroline Bagnost,
étudiant en 5^e année à Epitech

Cette idée a pu se concrétiser grâce aux 5 personnes qui ont travaillé avec moi durant ces deux années : Sébastien Béquignon, Benjamin Crochard, Camille Fabreguettes, Marina Joffrineau et Quentin Mangeon.

Digital Wardrobe a été réalisé dans le cadre de nos études de notre 3^e à notre 5^e année à Epitech. Ceci en parallèle de nos différents projets et, pour certains, de leur année d'étude à l'étranger. Ce qui sera présenté ici est encore en cours d'implémentation, la version finale sera disponible courant février.

Entrons à présent dans le vif du sujet : l'application permet donc de numériser, sauvegarder et d'essayer ses propres vêtements. Pour ce faire nous avons choisi d'utiliser le capteur Kinect, non seulement pour sa caméra RGB et sa détection infrarouge, mais surtout pour son SDK(1) très fourni, les nombreux samples et sa communauté très active.

Digital Wardrobe se découpe en 4 modules, qui seront détaillés par la suite, tous articulés autour de Kinect :

- **Interface**, où les interactions Kinect permettent à l'utilisateur de naviguer entre les modes,
- **Numérisation**, où l'on acquiert le vêtement à l'aide de Kinect Fusion,
- **Modélisation**, où l'on construit le modèle 3D,
- **Projection**, où le modèle est projeté suivant les mouvements traqués par Kinect.

Pour des questions de performances et de facilité (nous sommes partis de d'exemples existants), le projet Digital Wardrobe a été principalement implémenté en C#/.Net, sauf pour la gestion des modèles 3D, puisque nous utilisons le format .fbx dont le SDK(2) n'est disponible qu'en C++.

Interface

Le projet étant centré sur Kinect, l'Interface doit donc être totalement utilisable par gestes. La première problématique a été matérielle : nous avions cinq Kinect 360 et une For Windows. Le fonctionnement basique est le même pour les deux caméras, la For Windows étant plus précise, le SDK permet une détection de gestes contrairement à la 360 qui ne voit que la position de la main. Nous avons donc créé une API qui gère les interactions en fonction du type de Kinect branchée.

L'Interface permet d'accéder aux deux modes d'utilisation de l'application, la numérisation et l'essayage (correspondant au module Projection).

Fig.1



L'essayage nécessitant de choisir un ou des vêtements, l'Interface propose une navigation visuelle dans la garde-robe virtuelle de l'utilisateur. Les modèles numérisés n'étant pas encore disponibles à l'essayage, la navigation est en cours d'implémentation Fig.1.

Les vêtements sont ordonnés en trois possibilités : haut, bas ou tenue. Il suffit de sélectionner le vêtement choisi par geste pour passer à l'essayage. Dans le cas du capteur 360, le défilement s'effectuera en sélectionnant des flèches pour aller vers la gauche ou la droite. Kinect For Windows étant capable de détecter un poing fermé, il suffira de fermer le poing et de tirer le vêtement souhaité vers nous.

Numérisation

Le module de Numérisation est le cœur du projet, la phase où il faut récupérer un maximum d'informations sur le vêtement pour obtenir un modèle précis.

Nous sommes partis d'une application créée par Joshua Blake, Face Fusion(3) qui permet de scanner sa tête en 3D en tournant sur soi-même. L'application répondait à plusieurs problématiques :

- Modéliser un objet en l'isolant de son environnement,
- Obtenir facilement toutes les faces de l'objet : tourner sur soi,
- Interagir avec l'application en étant éloigné et immobile,
- Commandes vocales et son d'état de l'acquisition (harmonie si tout va bien, dissonance en cas de problème).

A partir de cette application, on obtient un modèle 3D sans couleur, au format .obj ou stl. Nous avons travaillé à ajouter dans un premier temps la couleur, puis élargi la zone de numérisation au corps entier avant de la recentrer sur un vêtement choisi.

L'utilisateur en lançant la numérisation, va déterminer s'il s'agit d'un haut, d'un bas ou d'une tenue complète. Une fois cette information sélectionnée, l'interface va lui permettre d'isoler le vêtement plus précisément, en déplaçant des barres pour les limites hautes et basses(4).

L'acquisition proprement dite démarre dès que les barres sont placées, l'utilisateur doit alors tourner sur lui-même, lentement, en suivant les indications sonores mentionnées précédemment. Une fois l'acquisition faite, l'application va sauvegarder le modèle .fbx du vêtement.

Côté technique, le module récupère donc les DepthStream, ColorStream et SkeletonStream du capteur Kinect, à l'aide de l'événement Kinect AllFramesReady. Il les stocke dans des instances de la classe

(1) Kinect SDK : <http://msdn.microsoft.com/en-us/library/dn799271.aspx>

(2) FBX SDK : <http://docs.autodesk.com/FBX/2014/ENU/FBX-SDK-Documentation/index.html>

(3) <http://facefusion.codeplex.com/>

(4) Des barres obliques pour les manches sont en cours de développement

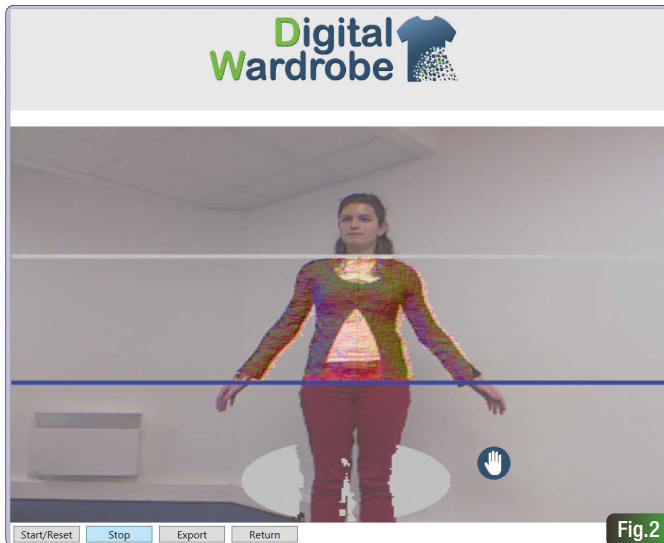


Fig.2



Fig.3

KinectFrameworkItem, applique des algorithmes de lissage à la DepthStream, puis mappe les DepthStream et ColorStream à l'aide de la fonction MapColorStreamToDepthStream Fig.2 et 3.

Des appels au SDK Kinect Fusion sont enfin faits pour intégrer la frame Kinect courante à un mesh 3D : la frame est d'abord alignée avec le mesh courant (AlignDepthFloatToReconstruction) puis intégrée si possible (IntegrateFrame).

```
bool trackingSucceeded = this.volume.AlignDepthFloatToReconstruction(
    depthFloatBuffer,
    12,
    residualFloatBuffer,
    out _alignmentEnergy,
    world_to_camera);

if (!trackingSucceeded || _alignmentEnergy > 0.1)
{
    //forget about the frame we just set
    volume.SetAlignDepthFloatToReconstructionReferenceFrame(_oldDepthFloat
Buffer);
    if (!trackingSucceeded)
    {
        this.trackingErrorCount++;
    }
    FusionStatusMessage = Numerisation.Properties.Resources.CameraTracking
Failed;
    _audioManager.State = AudioState.Error;
}
else
{
    FusionDepthProcessor.DepthToDepthFloatFrame(
    depthPixels,
    (int)depthSize.Width,
    (int)depthSize.Height,
    _oldDepthFloatBuffer,
    FusionDepthProcessor.DefaultMinimumDepth,
    FusionDepthProcessor.DefaultMaximumDepth,
    false);
}
```

Modélisation

Ce module doit fournir un modèle 3D utilisable par le module de Projection, d'après les informations envoyées par la Numérisation. Les modèles fournis par la Numérisation étaient en .obj (ou .stl) qui ne permettaient pas d'intégrer la couleur, nous sommes passés au .ply puis au .fbx. Ceci afin de pouvoir intégrer un squelette et les pondérations nécessaires aux suivis de mouvements et à l'animation des vêtements. Le SDK FBX d'Autodesk n'étant disponible qu'en C++, nous avons créé un wrapper C++/C# pour pouvoir l'utiliser sous la forme d'une DLL (bibliothèque dynamique) dans notre projet.

```
//Init FbxManager
_sdkManager = FbxManager::Create();

// Create a new scene so that it can be populated by the imported file.
_scene = FbxScene::Create(_sdkManager, "FbxScene");
_mesh = FbxMesh::Create(_sdkManager, "myMesh");

//prepare normals (depth vector for light sources and curves)
_normalsLayer = _mesh->CreateElementNormal();
_normalsLayer->SetMappingMode(FbxLayerElement::eByControlPoint);
_normalsLayer->SetReferenceMode(FbxLayerElement::eDirect);

//prepare color vertex layer
_colorLayer = _mesh->CreateElementVertexColor();
_colorLayer->SetMappingMode(FbxLayerElement::eByControlPoint);
_colorLayer->SetReferenceMode(FbxLayerElement::eDirect);

_uvLayer = _mesh->CreateElementUV("DiffuseUV");
_uvLayer->SetMappingMode(FbxLayerElement::eByPolygonVertex);
_uvLayer->SetReferenceMode(FbxLayerElement::eIndexToDirect);
```

Le module va donc construire le modèle .fbx en commençant par sa forme (mesh), puis sa couleur (un pixel par vertice) et enfin le squelette. La gestion des mesh comprend toutes les informations brutes envoyées par Kinect grâce au module de Numérisation : vertices, normales, polygones, sommets.

La couleur quant à elle est tout d'abord traitée sous la forme d'un Vertex color, un tableau liant chaque vertice à une lumière qui se diffuse sur le modèle. La déformation du modèle est gérée par la partie skin, chaque

articulation (jointe) a un cluster contenant une liste de vertices à déformer en fonction d'un poids donné.

A l'heure actuelle nous travaillons sur un moyen d'obtenir une couleur exploitable pour l'essayage. Le module de Projection fonctionne avec XNA(5), donc une couleur sous forme de texture en .jpg, alors que la Modélisation a une couleur pour chaque vertex. Un mapping UV (déplier la texture d'une image pré-découpée) a été envisagé mais le passage à Unity 3D(6) pour la Projection devrait changer les perspectives.

Projection

Le module de Projection doit afficher le modèle 3D sélectionné en fonction de la morphologie et des mouvements de l'utilisateur. Nous nous sommes inspirés du sample Kinect Avateering(7) où un avatar 3D suit les mouvements de l'utilisateur. Ce sample utilise la technologie XNA pour afficher une scène 3D complète au format .fbx.

Le modèle .fbx généré par le module Modélisation est chargé et parsé grâce à une DLL C++ afin de récupérer les informations qui nous permettront de le redimensionner. Le redimensionnement s'effectue en fonction de la morphologie de l'utilisateur mais aussi en fonction de la distance à laquelle il se trouve du capteur.

```
/// <param name="pNode">Node in the armature.</param>
/// <param name="toFind">Name of the bone.</param>

float FbxParser::PrintNode(FbxNode* pNode, const std::string & toFind)
{
    FbxSkeleton* ISkeleton = (FbxSkeleton*) pNode->GetNodeAttribute();
    std::string nodeName = pNode->GetName();

    // Print the node's attributes.
    if (nodeName.compare(toFind) == 0)
    {
        // In the .FBX file, contains the size of the bone in centimeters
        found = ISkeleton->Size.Get();
        return found;
    }

    // Recursively look for the children the children.
    for(int j = 0; j < pNode->GetChildCount(); j++)
        PrintNode(pNode->GetChild(j), toFind);
    return found;
}
```

Il faut ensuite lier le squelette du modèle 3D au squelette détecté par Kinect afin de suivre les mouvements en temps réel.

```
protected void BuildJointHierarchy()
{
    if (null == this.nuiJointToAvatarBoneIndex)
    {
        this.nuiJointToAvatarBoneIndex = new Dictionary<JointType, int>();
    }

    // Add(Kinect joint, Model Joint)
    this.nuiJointToAvatarBoneIndex.Add(JointType.HipCenter, 1);
}
```

(5) XNA : <http://msdn.microsoft.com/en-us/library/bb203940.aspx>

(6) Unity 3D : <http://unity3d.com/>

(7) Avateering : <http://msdn.microsoft.com/en-us/library/jj131041.aspx>



```
[...]
this.nuiJointToAvatarBoneIndex.Add(JointType.FootRight, 56);
}
```

L'utilisation de Game Studio XNA pose cependant quelques problèmes. L'affichage se fait dans une fenêtre de type Game XNA, il a fallu créer un projet de transition pour pouvoir créer un bitmap de cette fenêtre et l'intégrer dans un WPF.

```
namespace WPFGame
{
    class GameViewModel
    {
        private KinectSensor sensor;
        private APIKinect.APIKinectBase api;
        private System.Collections.Generic.Dictionary<string, Database.Session.strucOutfit>
dictionaryClothes;

        public GameViewModel(KinectSensor sensor, APIKinect.APIKinectBase api, System.
Collections.
Generic.Dictionary<string, Database.Session.strucOutfit> dictionaryClothes)
        {
            Game = new Microsoft.Samples.Kinect.Projection.Projection(sensor, api, dictionary
Clothes);
        }

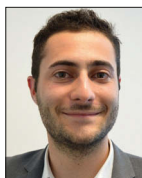
        public Microsoft.Samples.Kinect.Projection.Projection Game { get; private set; }
    }
}
```

XNA pré-compile également les modèles 3D ce qui handicape le chargement dynamique des modèles pour l'essayage. Pour ces raisons, en plus du fait que le rendu graphique est limité et que la technologie n'est plus supportée, nous avons commencé à travailler avec Unity Fig.4.

La release du projet sera disponible courant février et mise à disposition sur un github. N'hésitez pas à visiter notre site digitalwardrobe.fr pour plus d'informations.

Facilitez l'authentification des utilisateurs mobiles avec Amazon Cognito !

Nous allons exposer dans cet article comment utiliser le service Cognito afin de répondre aux besoins des utilisateurs d'applications mobiles. En effet, Cognito est un service simple de synchronisation de données et d'identités d'utilisateurs; il permet de gérer et de synchroniser de manière sécurisée les données d'applications de vos utilisateurs sur tous leurs périphériques mobiles.



Michael Garcia,
Solutions Architect AWS

Pourquoi ce service est-il essentiel pour les développeurs aujourd'hui ? Ces dernières années, le nombre de terminaux mobiles par utilisateur (smartphones, tablettes) a évolué de manière très importante. Cette multiplication des terminaux dédiés à un usage professionnel ou personnel nous amène à trois constats majeurs :

- Les utilisateurs souhaitent pouvoir avoir accès à leurs applications quel que soit le terminal utilisé. Le premier constat est donc qu'il est essentiel de développer une version mobile de son application dans le but de faciliter l'expérience des utilisateurs finaux sur des appareils de type smartphones et tablettes.
- Le second étant que la fédération d'identité est devenue un prérequis pour toutes les applications. En effet, les utilisateurs attendent un processus d'authentification simplifié, car cela n'est pas toujours facile pour eux de renseigner un nom d'utilisateur et un mot de passe sur les appareils mobiles. C'est pour cela qu'il est aujourd'hui nécessaire de proposer une connexion rapide à une application en utilisant son compte Facebook, Amazon ou Google.
- Enfin, les utilisateurs veulent pouvoir retrouver leurs données quel que soit le terminal utilisé; la synchronisation des données doit donc être automatique pour permettre aux utilisateurs une transition fluide.

L'implémentation de ces fonctionnalités représente un challenge pour les développeurs en raison du nombre des différents fournisseurs d'identité à intégrer et de la charge que représente le support des différents types de plateformes.

Cognito permet notamment d'apporter une solution à cette problématique de manière sécurisée, tout en minimisant l'effort de développement. En effet ce dernier permet d'unifier la couche d'authentification et intègre la synchronisation de données sur les différentes plateformes mobiles (iOS, Android), quel que soit l'appareil mobile utilisé. Plusieurs

types de fournisseur d'identité sont aujourd'hui supportés : les fournisseurs d'identités publics (Google, Facebook, Amazon), les fournisseurs d'identités privés (votre propre service d'authentification d'utilisateurs) ou encore tout fournisseur d'identité de type OpenId Connect (par exemple Salesforce.com).

Pour simplifier la prise en main, nous allons nous concentrer dans cet article sur les étapes qui vous permettront de démarrer facilement avec Cognito en vous intégrant avec un fournisseur d'identité public.

Création d'un pool d'identités Amazon Cognito

Identifiez-vous dans la Console Web AWS (<https://console.aws.amazon.com/>) et choisissez le service Amazon Cognito. Sur la page d'accueil du service, choisissez de créer un nouveau pool d'identité Cognito.

Pour proposer à vos utilisateurs l'authentification Google, Amazon ou Facebook dans vos applications, il est nécessaire de remplir la section dédiée aux fournisseurs d'identité avec l'identifiant adéquat (Fig.1).

Vous pouvez ensuite autoriser ou interdire l'accès aux utilisateurs non-authentifiés (Fig.2).

Les utilisateurs authentifiés sont ceux qui utilisent au moins un fournisseur d'identité pour s'authentifier auprès d'Amazon Cognito, tous les autres seront considérés comme non-authentifiés.

Dernière étape, la création de rôles vous permet de définir des règles de sécurité et de contrôler l'accès aux différentes ressources AWS que vos utilisateurs effectueront au travers de votre application. Avec Cognito, comme expliqué auparavant, il est possible de prendre en charge des utilisateurs qui se connectent via l'un des fournisseurs d'identités pris en charge (Amazon, Facebook ou Google), vous pouvez aussi prendre en charge des utilisateurs non authentifiés tout en utilisant le Cloud AWS pour enregistrer et synchroniser les données des utilisateurs de leurs applications mobiles. Vous remarquerez donc qu'il existe deux types de rôles, un pour les utilisateurs non-authentifiés et un pour les utilisateurs authentifiés (Fig.3).

En vous basant sur ces mécanismes, il vous est donc possible d'implémenter différents modèles d'authentification pour votre application. Vous pourrez par exemple interdire l'accès aux utilisateurs non-authentifiés, ou, au contraire, l'autoriser et laisser le loisir à l'utilisateur de s'authentifier lorsqu'il le souhaite. Dès lors ce dernier conservera son identité ainsi que ses données et deviendra alors un utilisateur authentifié.

Step 1: New Identity Pool

Identity pools are used to store end user identities. To declare a new identity pool, enter a unique name.

Identity Pool Name MyAppName
Example: My App Name

Public Identity Providers

Amazon Cognito recognizes tokens from these public identity providers. You need to specify the application provider identifier you plan to support in your application. You can always change your selection of providers after your identity pool is created. Learn more: [Android](#), [iOS](#).

Amazon App ID amzn1.application.188a56d827a7d6555a8b67a5d
Example: amzn1.application.188a56d827a7d6555a8b67a5d

Facebook App ID 7346241598935555
Example: 7346241598935555

Google Client ID 123456789012.apps.googleusercontent.com
Example: 123456789012.apps.googleusercontent.com

Unauthenticated Identities

Amazon Cognito can support unauthenticated identities by providing a unique identifier and AWS credentials for users who do not authenticate with an identity provider. If your application allows users who do not log in, you can enable access for unauthenticated identities. Learn more: [Android](#), [iOS](#).

☒ Enable Access to Unauthenticated Identities

Fig.1

Fig.2

Assign Role to Authenticated Identities

This role is used for end users who have logged in through a public identity provider.

IAM Role Create a new IAM role

Role Name Cognito_MyAppNameAuth_DefaultRole (preview IAM role policy)

Assign Role to Unauthenticated Identities

Unauthenticated end users automatically assume this role.

IAM Role Create a new IAM role

Role Name Cognito_MyAppNameUnauth_DefaultRole (preview IAM role policy)

Fig.3

Utilisateurs non-authentifiés

Passons maintenant à la partie développement mobile, votre première action sera d'intégrer le SDK mobile AWS à votre projet, pour cela rendez-vous sur <http://aws.amazon.com/mobile/sdk/>.

Nous allons maintenant initialiser le **credential provider**: il s'agit d'un objet qui a pour rôle de récupérer l'identifiant unique et spécifique à chacun de vos utilisateurs. Ce dernier aura également la charge de délivrer un accès temporaire aux ressources AWS en accord avec les politiques de sécurité que vous avez mises en place lors de la création du pool d'identités.

iOS

```
//initialize the Cognito credentials provider with the values from
//your identity pool.
AWSCognitoCredentialsProvider *credentialsProvider = [AWSCognitoCredentialsProvider
credentialsWithRegionType:AWSRegionUSEast1
accountId:@"1234567890", // your AWS Account id
identityPoolId:@"us-east-1:XXXXXXXX-XXXX-XXXX-XXXX-XXXXXXXX", //your identity
pool id
unauthRoleArn:@"arn:aws:iam::XXXXXXXX:role/YourRoleName", //an unauthenticated
role ARN
authRoleArn:@"arn:aws:iam::XXXXXXXX:role/YourRoleName" //an authenticated
role ARN
];
```

Android

```
//initialize a credentials provider object with your Activity's context and
//the values from your identity pool
CognitoCachingCredentialsProvider credentialsProvider = new CognitoCachingCredentials
Provider(
getContext(), // get the context for the current activity
"1234567890", // your AWS Account id
"us-east-1:XXXXXXXX-XXXX-XXXX-XXXX-XXXXXXXX", //your identity pool id
"arn:aws:iam::XXXXXXXX:role/YourRoleName", //an authenticated role ARN
"arn:aws:iam::XXXXXXXX:role/YourRoleName", //an unauthenticated role ARN
Regions.US_EAST_1 //Region
);
```

Lors de la première utilisation, le **credential provider** demandera un identifiant unique auprès du service Amazon Cognito et utilisera un mécanisme de cache pour que ce dernier persiste sur l'appareil utilisé. Pour les connexions suivantes, un appel vers le service Cognito sera également effectué par le **credential provider** mais avec l'identité précédemment mise en cache. La mise en cache est transparente, elle est entièrement gérée par le SDK mobile.

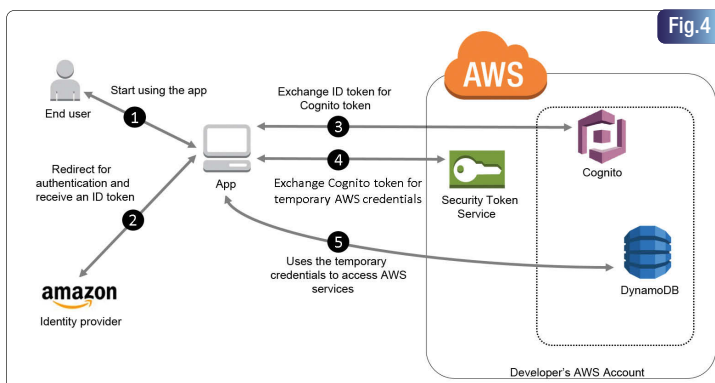


Fig.4

Utilisateurs authentifiés

Détaillons maintenant la gestion des utilisateurs authentifiés avec un fournisseur d'identité public (Google, Amazon ou Facebook) grâce au schéma suivant (Fig.4). Dans cet exemple nous utiliserons Amazon comme fournisseur d'identité.

Un utilisateur commence à utiliser l'application (Etape 1), il souhaite ensuite s'authentifier et sera donc redirigé vers le fournisseur d'identité choisi. Voici le type d'écran présenté par le fournisseur d'identité (Fig.5). Une fois son authentification effectuée le fournisseur d'identité lui donne un jeton d'authentification, ce jeton sera ensuite délivré à votre application et sera la preuve que votre utilisateur s'est bien authentifié auprès du fournisseur d'identité (Etape 2). Il faudra ensuite transmettre ce jeton au service Amazon Cognito pour que le service puisse associer l'identité de l'utilisateur au compte provenant du fournisseur d'identité (Etape 3). Une fois cela effectué, la dernière étape consiste à échanger le jeton Amazon Cognito contre un jeton temporaire d'accès aux services AWS en utilisant STS (Security Token Service) (Etape 4) ce qui permettra d'utiliser les autres services AWS, comme Amazon DynamoDB dans cet exemple (Etape 5).

Le SDK Mobile AWS a aussi pour rôle de simplifier le développement en masquant une partie de cette cinématique. En effet, une fois le jeton provenant du fournisseur d'identité récupéré (Etape 2), quelques lignes de code suffisent à récupérer une identité Cognito et effectuer des appels aux autres services AWS. Il suffit d'utiliser la propriété **login** de l'objet **credential provider**, soit à l'initialisation de l'objet soit au cours de son cycle de vie comme montré dans l'exemple qui suit.

iOS

```
// Retrieve the token from your identity provider.
NSString *token = "Token_From_Identity_Provider";
// For Facebook
credentialsProvider.logins = @{ AWS_CognitoLoginProviderKeyFacebook: token };
// For Google
credentialsProvider.logins = @{ AWS_CognitoLoginProviderKeyGoogle: token };
// For Amazon
credentialsProvider.logins = @{ AWS_CognitoLoginProviderKeyLoginWithAmazon: token };
```

Android

```
String token = "Token_From_Identity_Provider";
Map logins = new HashMap();
//Set the Facebook login token
logins.put("graph.facebook.com", token);
//Set the Google login token
```

Fig.5

```
logins.put("accounts.google.com", token);
//Set the Amazon login token
logins.put("www.amazon.com", token);

credentialsProvider.withLogins(logins);
```

Utilisation des services AWS

Vous possédez dorénavant un accès temporaire et sécurisé qui répond aux permissions définies précédemment dans votre pool d'identités que cela soit pour un utilisateur authentifié ou non-authentifié. Il est donc maintenant très facile d'utiliser le **credential provider** pour configurer le SDK Mobile, facilitant ainsi l'accès aux différents services AWS.

iOS

```
// The first step is to create a configuration object for the AWS SDK
// and initialize it with the Cognito credentials provider
AWSServiceConfiguration *configuration = [AWSServiceConfiguration
configurationWithRegion:AWSRegionUSEast1
provider:credentialsProvider];

// We can then set this as the default configuration for all AWS SDKs
[AWSServiceManager defaultManager].defaultServiceConfiguration = configuration;
```

Android

```
// The Android credentials provider object can be passed directly to the
// constructor for other AWS SDKs.
AmazonDynamoDB client = new AmazonDynamoDBClient(credentialsProvider);
```

Synchronisation des données

Cognito met également à disposition des développeurs un **Datastore** pouvant être synchronisé très simplement entre différents appareils. Ce **Datastore** permet de stocker des données selon un modèle simple de type clé-valeur, ce qui convient très bien pour stocker par exemple les préférences utilisateur d'une application. Vous pouvez créer autant d'identités que vous le souhaitez dans vos pools d'identités et votre espace de synchronisation. Chaque identité Cognito incluse dans l'espace de synchronisation dispose de son propre stockage d'informations utilisateur.

Voyons ensemble comment créer un jeu de données, le sauvegarder dans le **Datastore** Amazon Cognito et récupérer ce jeu de données sur un autre appareil mobile.

Ce cas de figure correspond au cas où un utilisateur se connecte à l'application sur version mobile puis plus tard sur la version Web (ou inversement), les données doivent être synchronisées automatiquement pour permettre une bonne expérience. Dans cet exemple, le jeu de données représentera le nom du joueur, son meilleur score ainsi que son niveau actuel dans un jeu.

Tout d'abord il faut initialiser le client de synchronisation Cognito.

iOS

```
AWSCognito *syncClient = [AWSCognito defaultCognito];
```

Android

```
//COGNITO_POOL_ID is a constant set to your Cognito Pool Id
//provider is the provider you just initialized
CognitoSyncManager syncClient = new CognitoSyncManager(myActivity.getContext(),
COGNITO_POOL_ID, Regions.US_EAST_1, cognitoProvider);
```

La création d'un jeu de données ainsi que la synchronisation avec le **Datastore** Cognito se fait facilement:

iOS

```
AWSCognitoDataset *dataset = [syncClient openOrCreateDataset:@"GameInfo"];

//call this once when you open the dataset to get the latest data from the cloud
[dataset synchronize];

[dataset setString:@"David" forKey:@"playerName"];
[dataset setString:@"29" forKey:@"currentLevel"];
[dataset setString:@"120345" forKey:@"highScore"];

//now synchronize this data to make it available to other devices
[dataset synchronize];
```

Android

```
Dataset dataset = client.openOrCreateDataset("GameInfo");

//call this once when you open the dataset to get the latest data from the cloud
dataset.synchronize(this, syncCallback);

dataset.put("playerName","David");
dataset.put("currentLevel","29");
dataset.put("highScore","120345");

//now synchronize this data to make it available to other devices
dataset.synchronize(this, syncCallback);
```

Nos données sont donc maintenant sauvegardées par le service grâce à l'opération de synchronisation. Nous sommes alors en mesure de récupérer ces données très facilement sur le même appareil ou sur un appareil différent appartenant à une autre plateforme.

iOS

```
NSString *playerName = [dataset stringForKey:@"playerName"];
NSString *currentLevel = [dataset stringForKey:@"currentLevel"];
NSString *highScore = [dataset stringForKey:@"highScore"];
```

Android

```
String playerName = dataset.get("playerName");
String currentLevel = dataset.get("currentLevel");
String highScore = dataset.get("highScore");
```

Voici un exemple d'un retour d'API au format JSON contenant les données montrées en exemple, on retrouve bien les 3 informations « playerName », « currentLevel » et « highScore » (**Fig.5**).

Conclusion

Cognito vous permettra d'améliorer l'expérience d'authentification de vos utilisateurs. Ceci en leur offrant la possibilité de s'authentifier facilement et rapidement en utilisant le fournisseur d'identité de leur choix si vous en intégrez plusieurs dans votre application. Bien sûr tout cela leur assurant une transition fluide s'ils décident d'utiliser votre application sur un autre appareil mobile ou une autre plateforme.

L'effort de développement de ce type de fonctionnalité sera fortement réduit en utilisant Cognito, ce qui vous permettra de passer plus de temps à développer votre application plutôt qu'à gérer sa maintenance. Enfin, ce service vous permettra également d'avoir facilement accès de manière sécurisée aux autres services AWS pour construire de nouvelles fonctionnalités.



Microsoft Azure SDK 2.5 : quoi de neuf ?

La plateforme Cloud de Microsoft est en constante évolution, et bien qu'il existe des portails et des APIs pour accéder à ses services, les développeurs apprécient le fait que l'accès à ses services soit disponible au sein de leur environnement de développement favori que ce soit Eclipse, Netbeans et bien entendu Visual Studio.



Wilfried Woivré
MVP Windows Azure - Soat

Aujourd'hui nous allons nous intéresser à l'environnement de développement de Microsoft, et notamment la version 2015, puisque c'est la première version du SDK qui supporte cette version en preview de Visual Studio. Cependant cette version de SDK n'oublie pas les utilisateurs des versions précédentes puisqu'elle est disponible sur les versions 2012, 2013 et 2015, elle est même disponible pour les versions de Visual Studio Express, et, bien entendu, la version « community ». Cette version apporte par ailleurs de nombreuses améliorations que nous allons détailler par la suite, et, bien entendu, corrige un certain nombre d'anomalies des versions précédentes.

Une opportunité de démarrer avec Azure

Depuis que la plateforme a démarré, il existe pléthore de documentation et d'exemples de code sur diverses versions des SDKs, et notamment des bibliothèques autour du Storage Azure. Si vous êtes développeur C#, ce SDK peut vous apporter de l'aide pour démarrer avec la plateforme, puisqu'il est désormais possible de créer des projets « QuickStarts » afin de comprendre comment certaines briques Azure fonctionnent comme on peut le voir ci-dessous : Fig.1.

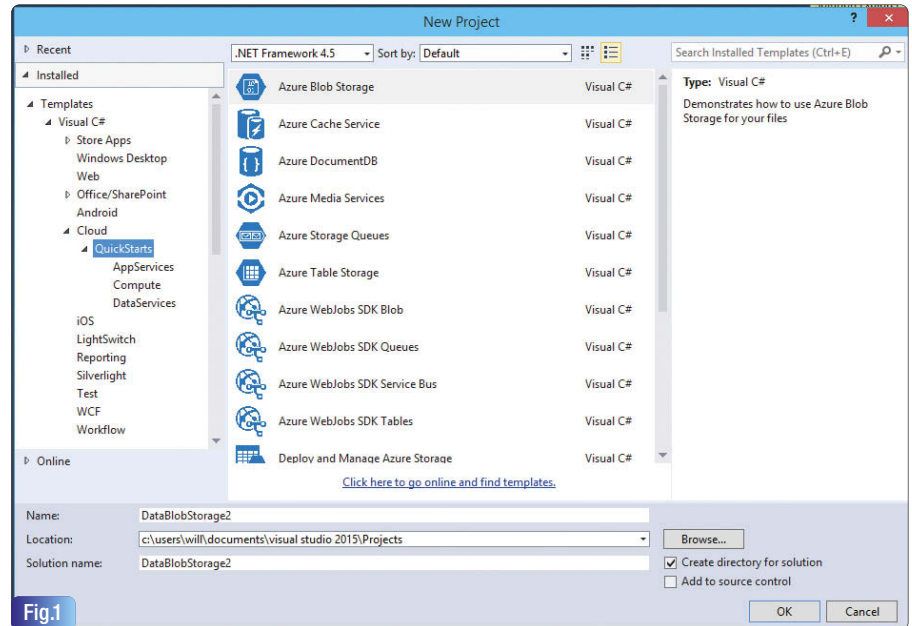


Fig.1

A noter que ces différents projets contiennent à la fois du code fonctionnel, simple à utiliser et à comprendre, et notamment beaucoup de commentaires, dont des liens utiles vers la documentation Azure, comme on peut le voir avec le template de projets pour DocumentDB Fig.2. Il est donc possible que cette galerie s'agrandisse avec le temps et prenne plus de cas d'usages, et bien entendu qu'elle se généralise à toutes les technologies supportées par la plateforme Azure.

Un nouveau type de projet

Mis à part les différents « QuickStarts » introduits avec cette version 2.5 du SDK, il y a un nouveau type de projet qui a fait son apparition et qui concerne cette fois-ci la fonctionnalité Azure Resource Manager: il s'agit du projet « Cloud Deployment Project », et il existe à la fois pour les développeurs C#, mais aussi VB.Net Fig.3.

Par la suite, un nouveau menu apparaît avec différents packages qui vont vous permettre de créer et de configurer plusieurs services dans Azure qui seront regroupés au sein d'un même Resource Group. A ce jour les packages qui existent s'articulent autour de Websites en ajoutant par la suite Application Insights, SQL Database et Redis Cache. Outre la création du site Web qui est somme toute assez classique au sein de Visual Studio, vous allez avoir un nouveau projet qui s'appelle WebApplication.Deployment. Ce dernier contient différentes données dont un fichier PowerShell qui servira lors de la publication sur Azure. Il y a aussi un fichier JSON pour les différents paramétrages de votre application en y incluant SQL Database et Redis si vous les avez sélectionnés dans votre template, et, pour finir, l'utilitaire AzCopy qui permet, entre autres, de faire des copies de blobs via des lignes de commandes.

Cependant, ne vous inquiétez pas, on ne vous demande pas de faire du Powershell: il est pos-

```
// In order to run this sample for Azure DocumentDB you need to have the following pre-requisites;
//
// 1) An Azure subscription
//    If you don't have an Azure subscription you can get a free trial. To create a free trial see:
//    http://azure.microsoft.com/en-us/pricing/free-trial/
//
// 2) An Azure DocumentDB Account
//    For instructions on creating an Azure DocumentDB account see:
//    http://azure.microsoft.com/en-us/documentation/articles/documentdb-create-account/
//
// NOTE: Never store credentials in source code. In this example placeholders in App.config are used.
// For information on how to store credentials, see:
// Azure Websites: How Application Strings and Connection Strings Work
// http://azure.microsoft.com/blog/2013/07/17/windows-azure-web-sites-how-application-strings-and-connection-strings-work/
//
// Additional Azure DocumentDB Code Samples can be found on http://go.microsoft.com/fwlink/?LinkID=512740
//
static readonly string endpoint = ConfigurationManager.AppSettings["endpoint"];
static readonly string authKey = ConfigurationManager.AppSettings["authKey"];
static DocumentClient client;

Preferences
static void Main(string[] args)
{
    try
    {
        // 1. Get a reference to your Azure DocumentDB.
        // It is recommended to create an instance of DocumentClient and reuse the same instance
        // as opposed to creating, using and destroying the instance time and time again
        // For this sample we are using the Defaults. There are optional parameters for things like ConnectionPolicy
        // that allows you to change from Gateway to Direct or from HTTPS to TCP.
        // For more information on this, please consult the Azure DocumentDB Documentation
        Console.WriteLine("1. Create an instance of DocumentClient");
        using (client = new DocumentClient(new Uri(endpoint), authKey))
        {
            // 2. The first thing you need is a Database.
            Console.WriteLine("2. Getting reference to Database");
            Database database = ReadOrCreateDatabase("QuickStarts");
        }
    }
}
```

Fig.2

sible de déployer votre application en faisant un simple clic droit sur votre projet puis Deployment, et vous avez une interface graphique qui vous demandera de saisir les différentes informations nécessaires pour le bon fonctionnement de la manipulation. Ce nouveau projet nous permet donc de facilement créer un ensemble de ressources dans Azure et de les regrouper au sein d'un même Resource Group; le but étant de gagner beaucoup de temps afin de gérer les différents environnements où vous souhaitez déployer sans avoir besoin d'une documentation qui détaille comment créer et configurer les différentes briques. Il n'aura jamais été aussi facile de gérer ces environnements de développement, de recette et de démonstration grâce à Azure.

Des diagnostics pour les machines virtuelles

Avoir des services sur Azure, c'est bien. Et comme souvent, c'est uniquement quand cela tombe en panne qu'on se dit qu'on aurait pu prévoir quelque chose. Pour cela, Microsoft a depuis le début, avec les Cloud Services, mis en place des outils pour gérer les diagnostics de nos machines, il en existe aussi sur les Websites. Cependant il existait un manque sur les machines virtuelles, et c'est maintenant résolu avec l'arrivée d'une nouvelle extension. Pour cela dans Visual Studio, il faut aller dans la fenêtre « Server Explorer », puis faire un clic droit sur une de vos machines virtuelles et installer l'extension «Microsoft Monitoring Agent

Diagnostics », puis il ne reste qu'à mettre en place les logs grâce à l'interface graphique, ou via le code de votre application [Fig.4](#).

L'avantage de cette extension, c'est qu'elle permet facilement de stocker nos diagnostics au sein du Storage Azure. Il est donc possible au sein de notre application de conserver la même expérience en termes de log, qu'elle soit hébergée au sein d'un Cloud Service, ou d'une machine virtuelle. Ou alors de pouvoir mixer nos différents diagnostics au sein du même stockage afin de faciliter la lecture de ceux-ci avec un outil tel qu'Azure Management Studio.

Amélioration des diagnostics avec « Event Tracing For Windows »

Les logs sont le nerf de la guerre lorsqu'on a des problèmes au sein d'une application, et bien entendu un applicatif hébergé sur Microsoft Azure n'échappe pas à la règle. Et je dirais même plus, qu'un applicatif hébergé sur une plateforme Cloud nécessite plus de log surtout en mode « Platform As A Service » vu que nous n'avons pas un accès total à la plateforme, ni aux machines déployées.

Pour activer les diagnostics ETW au sein de votre Cloud Services, il faut bien entendu avoir le dernier SDK installé, et développer sa solution d'Event Source : [Fig.5](#).

Il faut donc dériver de la classe EventSource, et implémenter nos différentes méthodes afin d'avoir des traces applicatives personnalisées que l'on va pouvoir stocker dans différentes

tables du storage Azure. Il faut ensuite activer ces logs dans l'outil de configuration de diagnostics de vos rôles. Il est de plus possible de choisir la table dans laquelle on veut stocker notre log en indiquant un id en attribut de votre méthode, puis de configurer la table de stockage pour cet id. Par ailleurs, l'avantage de passer par cette méthode plutôt qu'un log custom, c'est que l'on stocke nos traces facilement sans pour autant avoir à gérer la base de données, et notamment une base NoSQL comme c'est le cas ici, où la gestion des index est importante afin d'obtenir d'excellentes performances lors de la récupération des données.

Ajout de la gestion des blobs storage

Pour gérer vos comptes Azure Storage, vous devez certainement comme moi utiliser des outils tierces tel qu'Azure Management Studio, puisque le SDK Azure n'est pas le meilleur choix pour utiliser et monitorer cette brique. Cependant Microsoft continue de palier le problème de version en version, car dans cette release, la gestion des blobs storage a été améliorée. On peut notamment gérer les différents dossiers du container sans avoir une liste en mode fichier plat [Fig.6](#).

Ajout des diagnostics de « Crash Dump »

En plus des ETW, il est maintenant possible de configurer les Crash Dumps liés aux processus des machines. Par défaut, il est possible de

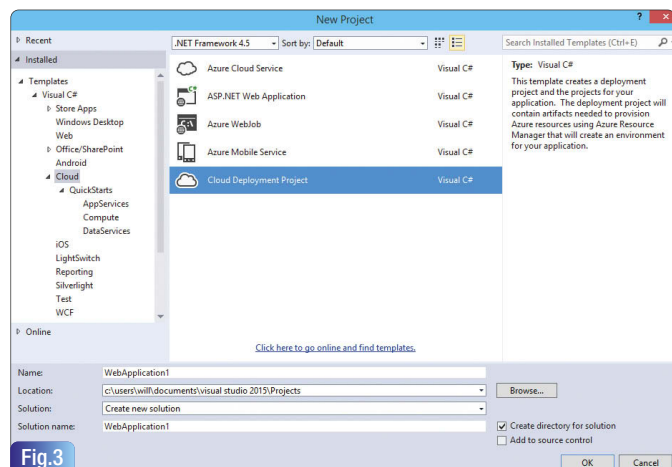


Fig.3

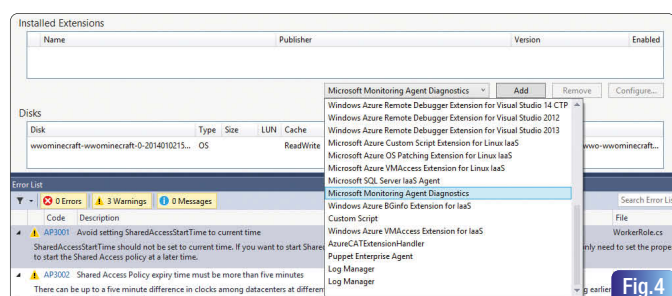


Fig.4

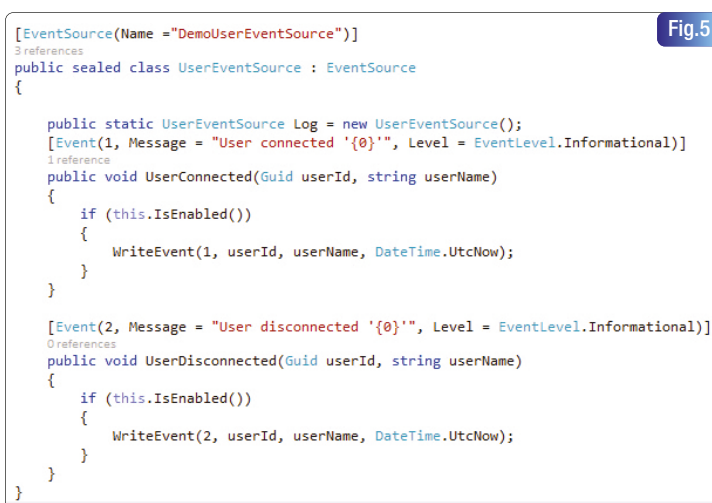


Fig.5

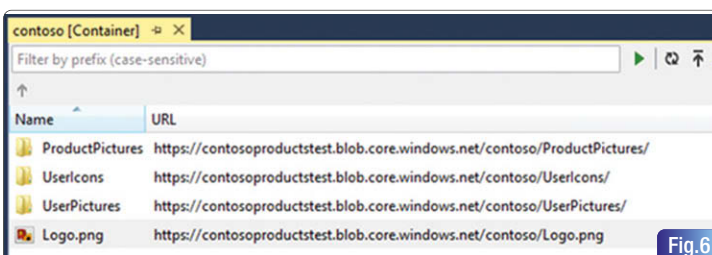


Fig.6

récupérer les crashs dumps des processus d'un Cloud Services. Lorsqu'on active ceux-ci, Visual Studio nous sélectionne par défaut les processus que l'on utilise dans nos Cloud Services à savoir `WaiSHost.exe` pour vos Web Roles, `WaWorkerHost` pour vos Worker Role et `w3wp.exe` pour votre pool IIS. Il est bien entendu possible d'en ajouter d'autres, ce qui peut être utile si vous configurez des « StartupTasks » sur vos Cloud Services [Fig.7](#).

Des nouveautés pour les Webjobs

Outre les logs qui ont été grandement améliorés dans cette version, il y a aussi le support de Webjobs qui s'installe avec ce nouveau SDK. Cela ne vous aura sans doute pas échappé que les Webjobs au sein d'Azure Websites sont maintenant disponibles en « GA ».

Cette nouvelle mise à jour de SDK va donc en profiter pour apporter énormément de fonctionnalités à Visual Studio pour cette brique d'Azure; notamment un template de projet de type console qui contient le SDK Webjobs au préalable, et ainsi la possibilité de le lier facilement à un Azure Website comme on pouvait le faire avec l'extension Azure Webjobs de Visual Studio [Fig.8](#).

On a donc suite à la disponibilité générale de WebJobs (20 octobre 2014), une intégration complète dans Visual Studio avec toutes les fonctionnalités qu'offre la brique Azure Websites. Ce qui est une bonne nouvelle pour

tous ceux qui utilisent cette fonctionnalité qui était jusqu'à présent très peu outillée à mon avis.

HDInsight arrive dans Visual Studio

Les aficionados du Big Data et de Visual Studio vont être ravis d'apprendre cette nouvelle puisqu'ils pourront utiliser HDInsight au sein de leur environnement de développement. On peut de plus créer de nouveaux projets de type Hive afin de gérer nos processus plus complexes avec HDInsight [Fig.9](#). Par ailleurs, le serveur explorer nous permet de naviguer au travers de nos différentes ressources HDInsight, mais pas d'en créer actuellement.

On va cependant pouvoir consulter nos différentes tables, et récupérer les 100 premières lignes de celles-ci, ce qui suffit amplement pour avoir par exemple le nom des colonnes associées à chacune d'elles, et ainsi faire des requêtes sur un cluster que l'on ne maîtrise pas, où alors pour débiter avec la technologie.

Visual Studio 2015 : votre nouvel IDE favori

A mon avis, les nouveautés les plus intéressantes pour les personnes qui utilisent couramment la plateforme Azure sont arrivées avec la nouvelle version de Visual Studio. En effet, on va retrouver divers ajouts qui vont nous simplifier la vie.

Connections avec plusieurs comptes Azure

Si vous aussi, vous avez plusieurs comptes Azure, qui sont éparpillés sur plusieurs adresses emails. Vous avez dû voir que Visual Studio n'était pas le meilleur outil pour vous accompagner sur Azure. Et bien c'est dorénavant fini, puisque vous pouvez vous connecter avec plusieurs comptes sur Azure depuis Visual Studio 2015. Vous n'aurez donc plus à switcher entre vos différents comptes, ou vous n'aurez plus à avoir un compte administrateur sur toutes vos souscriptions [Fig.10](#).

On garde toujours la possibilité de filtrer nos différents abonnements, mais aussi par région. De plus, lorsque vous êtes connecté à vos souscriptions Azure, il est maintenant possible d'accéder à nos serveurs SQL Database depuis la fenêtre Server Explorer.

Gestion des services connectés

Une des nouveautés de cette nouvelle version est de pouvoir ajouter des services connectés à notre application de n'importe quel type qu'elle soit. Il est possible d'ajouter différents types de services qu'ils soient proposés par Microsoft ou par d'autres éditeurs. En effet, dans cette première version il est possible d'ajouter Application Insights, Azure Mobile Services, Azure Storage, Office 365 APIs et même Salesforce pour .Net comme on peut le voir ci-

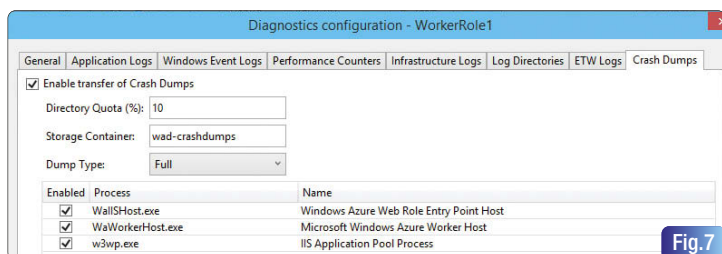


Fig.7

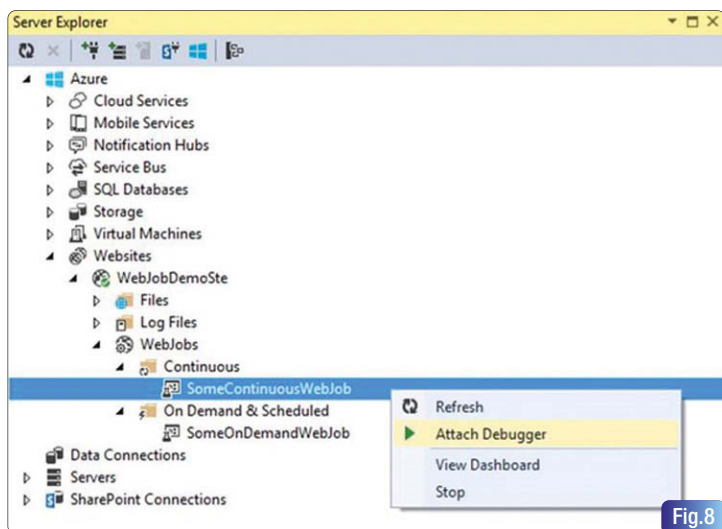


Fig.8

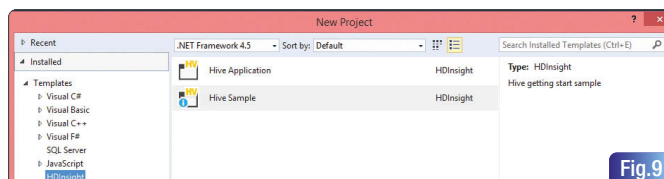


Fig.9

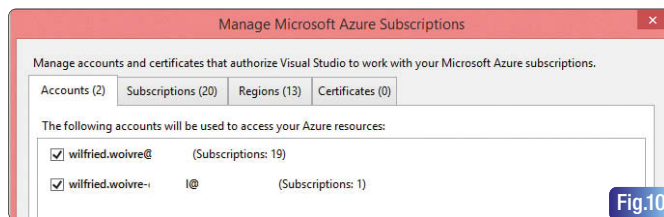


Fig.10

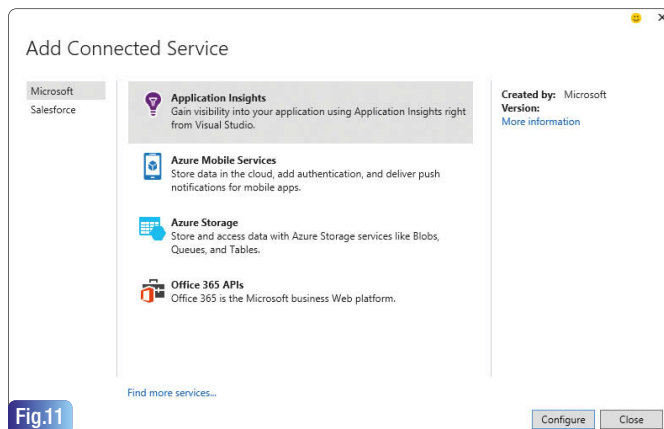


Fig.11

dessous : Fig.11. Selon moi l'intérêt d'ajouter ce type de service est qu'il devient très simple de référencer les différents services, ainsi on aura la dernière version du package NuGet pour le storage si on choisit celui-ci.

Single Sign On Entreprise

Pour diverses applications business, il est courant de devoir se connecter à un réseau Active Directory d'entreprise, bien entendu avec Azure AD. Il y a dans cette nouvelle version, une nouvelle fenêtre de configuration de l'Azure AD qui est plus « User-friendly » que celle qu'on pouvait retrouver sur l'AD avant Fig.11.

Code Analysis pour Azure

Etant pour ma part impressionné par les opportunités qu'offre Code Analysis dans Visual Studio 2015, il est tout à fait normal que cette annonce m'intéresse au plus haut point. Pour résumer la fonctionnalité en dehors d'Azure, il s'agit de nouvelle possibilité au Code Analysis de .Net afin de détecter les différents problèmes que l'on peut avoir dans notre code. Si on prend celui d'Azure, il permet entre autres d'avertir le développeur qu'il n'applique pas les bonnes pratiques de développement sur Azure.

Pour activer ceux-ci, il vous suffit d'aller sur NuGet, et d'installer le package « Microsoft.VisualStudio.Azure.CodeAnalysis » dans vos différents projets, et cela suffit pour qu'il soit activé et disponible au sein de notre application.

Grâce à ce package, on peut éviter énormément d'erreurs notamment avant les différents déploiements sur Azure. Par exemple, on peut voir que dans un Role, il ne faut pas utiliser le mot clé async au sein de celui-ci comme on peut le voir : Fig.12.

Mais aussi des astuces plus mineures, par exemple pour la gestion des token SAS pour le Blob Storage, il est conseillé de démarrer notre délai d'expiration 5 min avant l'heure fixée pour les problèmes de réglage des serveurs.

Support d'ASP.Net 5 Preview

Cette nouvelle version d'ASP.Net est pleine de promesses, et grâce à cette nouvelle version, il est possible de déployer nos applicatifs ASP.Net 5 sur Azure via un Publish Fig.13.

Pour résumer

Cette nouvelle version du SDK apporte beaucoup de fonctionnalités pour démarrer avec Azure, notamment avec les QuickStarts et le Code Analysis sur VS 2015. Cependant, il y a des améliorations sur toutes les briques d'Azure afin de simplifier les tâches des développeurs.

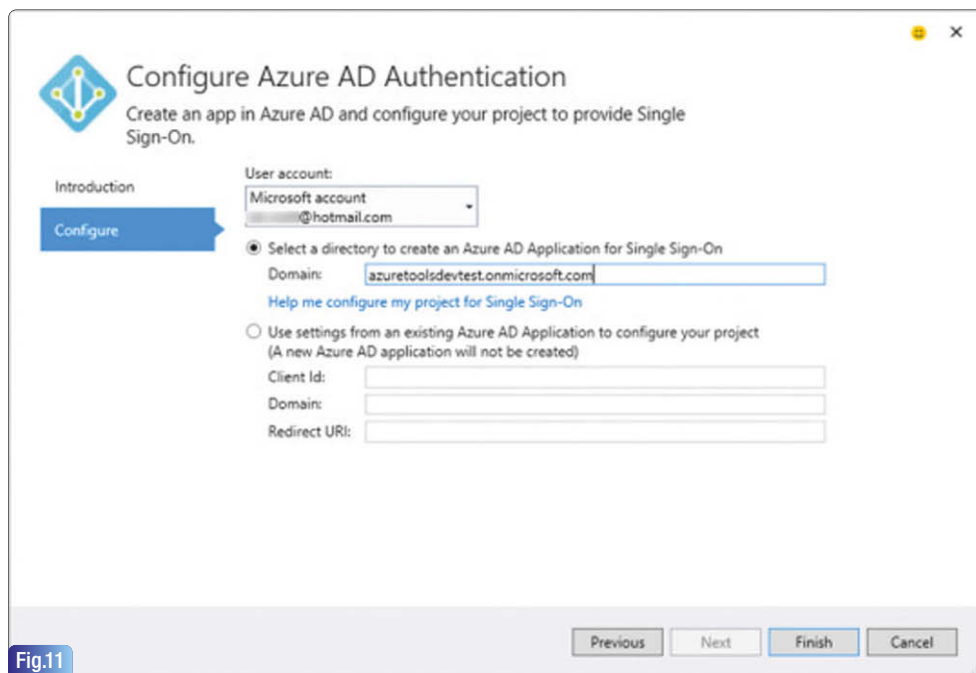


Fig.11

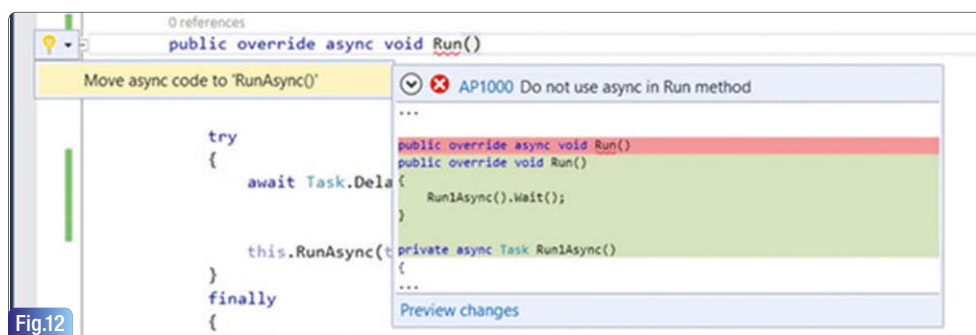


Fig.12

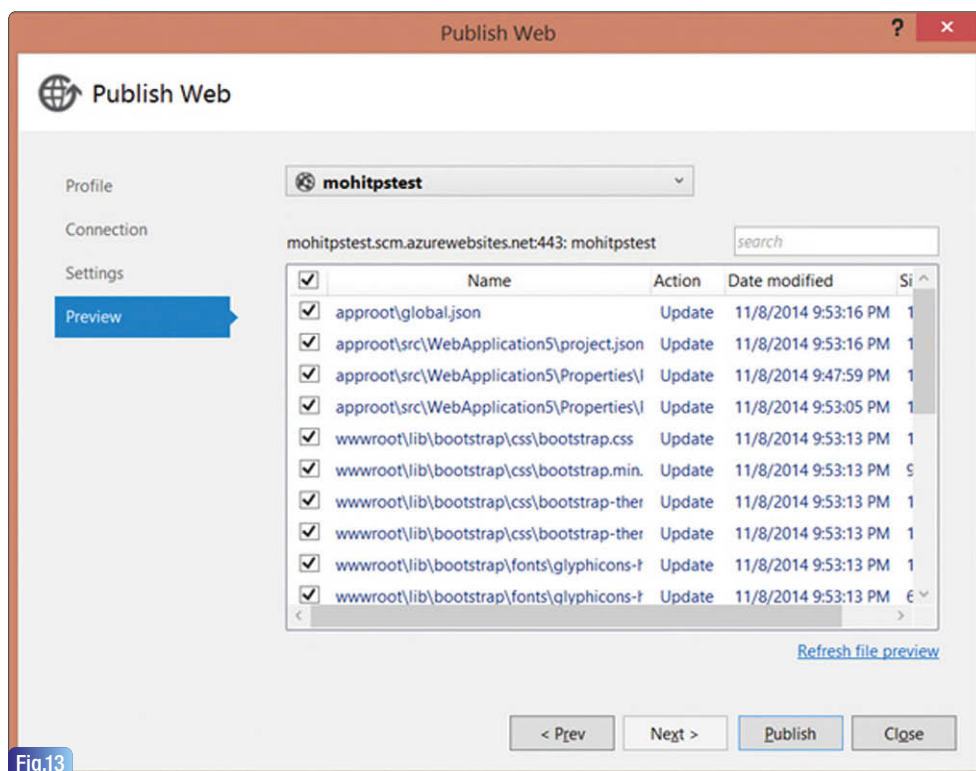


Fig.13

Timeline : 1995

objet : Windows devient un système d'exploitation

Après le couple DOS/Windows 3 en 1990, Microsoft rêve de passer à ce qui sera Windows XP. Il mettra dix ans pour y parvenir. Mais cela ne l'empêchera pas de devenir No 1 des systèmes d'exploitation avec des produits intermédiaires.

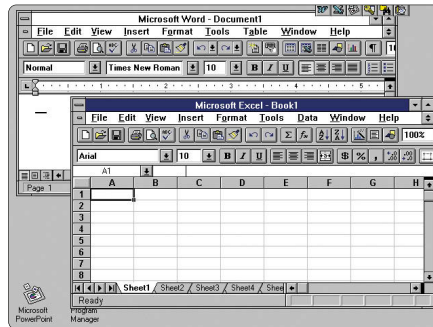


Yann Serra

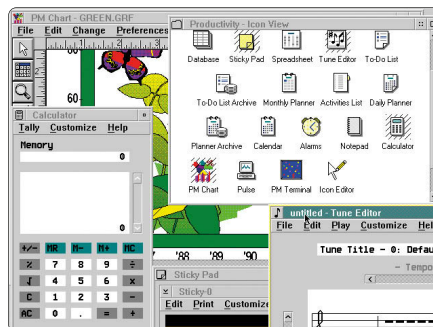
En 1990, Microsoft fait finalement régner son système d'exploitation MS-DOS dans le monde du compatible PC. Vendu à un tarif dérisoire aux fabricants d'ordinateurs s'ils le préinstallent en usine, MS-DOS 5.0 rend inutile l'achat supplémentaire de DR-DOS (Digital Research), de PC-DOS ou d'OS/2 (IBM). Le coup de grâce, c'est Windows 3. Depuis 5 ans, cet environnement graphique n'était qu'un gadget pour copier des fichiers à la souris. Mais cette version 3.0 permet désormais d'utiliser Word et Excel sur PC comme on le fait sur Macintosh, avec des menus et des icônes ! Dans la tête de Bill Gates, le patron de Microsoft, la suite est toute tracée : d'ici à 1992, Windows 3.1 aura, en plus, des polices de caractères de taille variable (TrueType) et des imprimantes qui imprimeront exactement ce que l'on verra à l'écran (WYSIWYG). Ensuite, il s'agira de proposer des fonctions que le Macintosh n'a pas. C'est le projet Cairo, qu'on échafaudera à partir du code 32 bits que Microsoft vient de récupérer en claquant la porte du projet OS/2. Dans ce code, il y a du fonctionnement en réseau, du stockage objet et de la protection mémoire. En revanche, vu les ambitions de ce système et vu qu'on ne sait encore faire que du 16 bits dans l'univers PC, il ne faudra pas compter sur Cairo avant 1996. Au moins.

OS/2 2.0 précipite le développement d'un Windows 32 bits

Oui, mais non. Bill Gates a crié victoire trop vite. Contre toute attente, IBM revient en avril 1992 avec un OS/2 2.0 qu'il a non seulement réussi à écrire sans Microsoft mais qui, en plus, fonctionne déjà en 32 bits. Et qui dispose d'un nouvel environnement graphique, le Workplace Shell, bien plus proche des fonctions du Macintosh que ne l'est Windows 3.1. Et qui, pour finir, est multitâche jusque dans ses fondements. Branle-bas de combat. Durant tout l'été, les ingénieurs de Microsoft planchent sur le design d'un système équivalent. Le but : lancer d'ici à la fin 1993 un OS « Chicago », reposant sur un noyau DOS multitâche 32 bits et dont toutes les fonctions



En 1990, Windows 3.0 pour MS-DOS 5 exécute Word et Excel sur PC comme s'il s'agissait d'un Macintosh. L'idée germe d'en faire un OS autonome.



En 1992, le très abouti OS/2 2.0 d'IBM incite Microsoft à avancer ses travaux sur un nouvel OS multitâche, graphique et 32 bits.

seraient directement accessibles à l'environnement graphique, sans passer par des commandes DOS, mais via des API appelées Win32. En juillet 93, le code 32 bits de Cairo permet surtout de produire un système qui n'a rien à voir avec Chicago : Windows NT (New Technology). Radicalement incompatible avec les applications pour DOS et Windows 3 déjà existantes, Windows NT est relégué à transformer les PC en serveurs, typiquement pour partager des fichiers en réseau. Et ce n'est pas inintéressant. Tricoté par David Cutler, l'un des architectes pour DEC de l'OS VMS, Windows NT se montre suffisamment performant pour, petit à petit, grignoter les parts de marché de Novell NetWare, alors leader des systèmes pour serveurs basés sur PC. Les avantages de Windows NT reposent sur sa rapidité à commuter les paquets en réseau et sur un système de fichiers, NTFS, plus riche. Problème, DEC ne tardera pas à découvrir que ces avantages sont en fait issus de bouts de codes que Cutler est allé chercher dans VMS. Microsoft enterrera l'action en justice avec un chèque de



En 1993, David Cutler, transfuge de DEC, assemble les codes d'OS/2 et les routines de VMS pour créer Windows NT.

En 1995, Bob Herbold est débauché de Procter & Gamble pour orchestrer une campagne marketing à 200 millions de dollars autour de Windows 95.

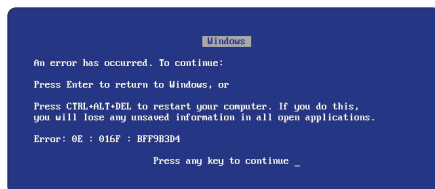


Windows 95 devient en un an le plus grand succès des systèmes d'exploitation de tous les temps.

150 millions de dollars en promettant à DEC de décliner Windows NT sur ses gammes de machines Alpha.

200 M\$ de marketing pour oublier que Windows 95 a deux ans de retard

En ce qui le concerne, Chicago, le système censé assurer la compatibilité ascendante de DOS/Windows, n'est pas du tout prêt en 1993. Des beta-versions plus ou moins laborieuses se succèdent encore pendant deux ans. Elles aboutissent finalement au système que l'on connaîtra sous le nom de Windows 95. Pour rattraper l'af-front personnel d'un tel retard, Bill Gates veut frapper un grand coup. Il veut faire de Windows 95 le produit de consommation que tout le monde rêve d'avoir, comme Apple en 1984 avec le Macintosh. Mais c'est mal parti : selon une étude menée par un cabinet externe, seuls 3 à 4% des consommateurs connaissent la marque Microsoft, alors qu'ils sont plus de 90% à savoir qui sont IBM et Apple. Bill Gates ne se décourage pas. Il embauche un champion du marketing, Bob Herbold, jusqu'ici patron de la pub chez Procter & Gamble et, sur ses conseils, signe un chèque de 200 millions de dollars pour financer



Les écrans bleus récurrents de Windows 95 traduisent un système aux fondations bancal.



Le rêve technologique d'un Windows grand public basé sur les bonnes idées d'OS/2 se concrétisera finalement en 2001 avec Windows XP, décliné en de multiples versions.

une campagne sans précédent dans l'histoire de l'informatique. Summum du caprice : s'offrir pour 6 mois la chanson Start me up des Rolling Stones en guise de jingle publicitaire. Juste parce que l'interface de Windows 95 se caractérise par un bouton Démarrer (« Start »). Mick Jagger demande plusieurs millions de dollars - 10 millions paraît-il - juste pour envoyer paître cet éditeur de logiciels qui croit jouer dans la cour des grands. Mais Bill Gates fait le chèque. En un an, Microsoft écoule 40 millions de copies de Windows 95, lequel devient alors le plus gros succès de l'histoire des systèmes d'exploitation. A noter que l'interface de Windows 95, son argument le plus plébiscité, n'a rien de révolutionnaire. Mais elle a le mérite de populariser des concepts inventés des années auparavant par

Sun (bouton droit), NeXT (icônes 3D), Acorn (barre des tâches) et, bien entendu, Apple (menu Pomme, corbeille).

Entre arrogance et écrans bleus

En 1996, IBM arrête le développement d'OS/2. En 1997, Apple est au bord de la faillite. Le tapage autour de Windows 95 est tel qu'il a séduit le grand public (pour la première fois Windows devient une plateforme de jeux vidéo), ainsi que les professionnels hors du carcan de la simple bureautique. Le succès est si démesuré que le gouvernement américain lui-même lance une action en justice pour concurrence déloyale contre Windows 95 qui inclut gratuitement un navigateur Web alors que Netscape le vend comme une option. Mais à ce stade de gloire, Microsoft s'en moque complètement et surpasse les meilleurs avocats pour envoyer balader l'administration US.

Le vrai problème de Microsoft à ce moment est plutôt qu'il doit maintenir deux gammes de systèmes d'exploitation qui sont, pourtant, complémentaires. D'un côté Windows NT se montre robuste et rapide mais il est incompatible avec l'existant (en particulier il n'a que très peu de pilotes pour supporter l'immense jungle des extensions - cartes graphiques, réseau, etc. - des PC). De l'autre, Windows 95 fonctionne sur tous les PC mais au prix d'un salmigondis de raccords entre des codes 16 et 32 bits qui ne cessent de s'effacer l'un l'autre dans la mémoire et provoquent des plantages quotidiens. Ce sont les fameux « écrans bleus de la mort », des gels de l'ordinateur avec perte brutale des données et affichage d'un code hexadécimal totalement abscons pour soi-disant expliquer l'erreur produite. La situation ridiculise à ce point l'image de marque de Windows qu'elle finit par favoriser l'émergence de concurrents (Linux sur les serveurs, OS X sur les Mac d'Apple), lesquels n'ont pas beaucoup d'efforts à faire pour convaincre que leurs produits ne laissent pas tomber l'utilisateur en plein travail, eux. Arrivé en 1999, Bill Gates ne peut que constater un double échec : il vient de rater durablement le marché émergent

des hébergeurs au profit de Linux et Apple lui a ravi l'image de technologie informatique la plus « cool ».

Windows XP, la consécration

Petit à petit, les applications et les pilotes reposent sur l'API Win32 commune à Windows NT et Windows. Si bien que la compatibilité ascendante avec l'ère DOS/Windows 3 perd de son importance. Mieux, les PC deviennent suffisamment puissants pour cloisonner cette compatibilité dans un émulateur interne, ce qui empêche les plantages de contaminer tout le système. Ainsi, en 2000, Windows NT 5.0, renommé Windows 2000, intègre la couche DirectX pour exécuter les jeux vidéo et Microsoft le vend même comme un système pour un PC de bureau. En 2001, dix ans après que Bill Gates ait imaginé Cairo, les deux systèmes fusionnent enfin en un seul : Windows XP (pour eXPerience).

Considéré comme le plus stable et le plus rapide des Windows grands publics jusqu'ici parus, XP sera décliné en de nombreuses versions : grand public, professionnelle, embarquée (pour les appareils industriels), Media Center (télévisions intelligentes, un bide), TabletPC (précurseurs de l'iPad, un bide aussi). On trouvera même une version dite « Fundamentals » capable de fonctionner sur les vieux PC et les clients légers avec très peu de ressources matérielles. Et puis, bien sûr, Windows XP servira aussi à continuer la lignée des Windows NT sur serveurs, sous le nouveau nom de Windows Server. XP se vendra à plus d'un milliard de copies jusqu'en 2014, année où Microsoft cessera définitivement de le supporter. L'extrême longévité de ce système - il équipait encore un tiers des PC en 2012 - s'explique par le fait que Microsoft a ensuite laborieusement tâtonné pour le faire évoluer. Windows Vista (2006) ne reconnaît plus les périphériques et aura des verrous de sécurité qui contraindront l'ergonomie. Windows 7 (2009) résoudra ces problèmes, mais les entreprises auront à peine eu le temps de l'envisager que Microsoft annoncera son remplacement par un Windows 8 (2012) conçu pour écrans tactiles... sur des PC non tactiles.

Abonnement : Service Abonnements PRO-GRAMMEZ, 4 Rue de Mouchy, 60438 Noailles Cedex - Tél. : 01 55 56 70 55 - 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



Directeur de la publication & rédacteur en chef : François Tonic

Ont collaboré à ce numéro :
S. Saurail, C. Calais, Y. Serra

Secrétaire de rédaction : Olivier Pavier
Experts : V. Loquet, Chouguat Aymen, B. Baldacci, P. Lemberger, B. Guinebertière, V. Berdot, P. Bron, S. Duchateau, P. Rosado, O. Nourry, J-P Villain, S. Thibaut, J-P Mengual, C. Villeneuve, F. Ludovic, V. Chegham, S. Nichele, A. Masciulli, C. Vilet, C. Bagnost, M. Garcia, W. Woivré,

Une publication Nefer-IT
7 avenue Roger Chambonnet
91220 Brétigny sur Orge
redaction@programmez.com
Tél. : 01 60 85 39 96

Crédits couverture : 09-24-13 © VasiaKoman / iStock

Maquette : Pierre Sandré

Publicité : PC Presse,
Tél. : 01 74 70 16 30, Fax : 01 41 38 29 75
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
Responsable titre : Terry MATTARD
Téléphone : 09 67 32 09 34

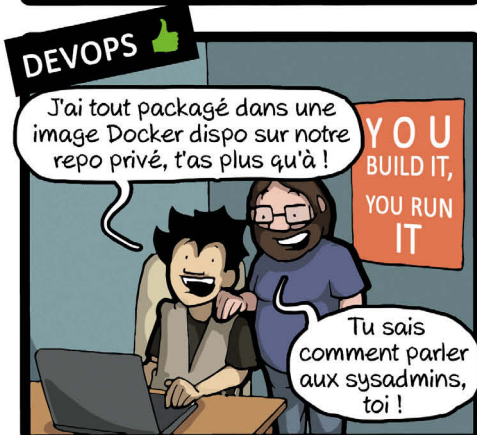
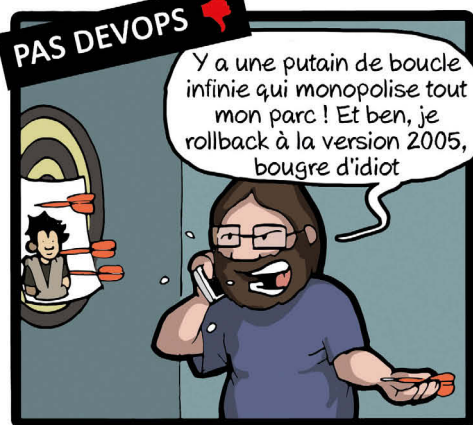
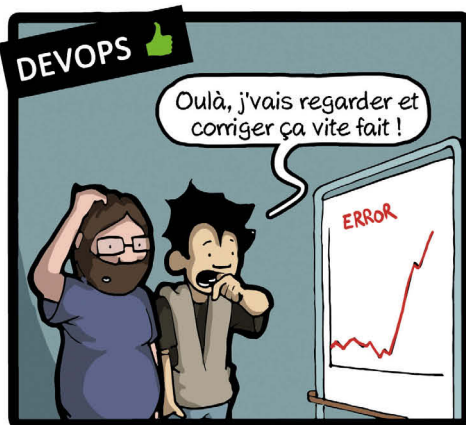
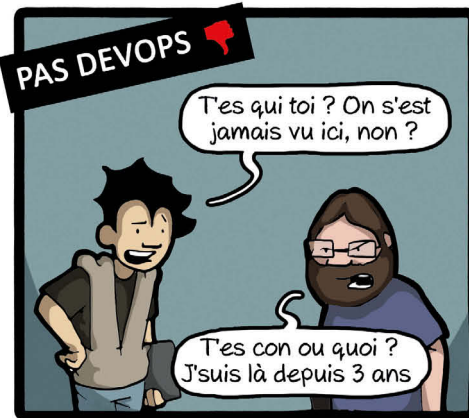
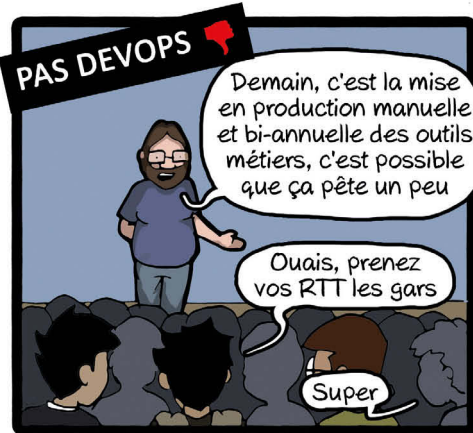
Contacts

Rédacteur en chef :
ftonic@programmez.com
Rédaction : redaction@programmez.com
Webmaster : webmaster@programmez.com
Publicité : pub@programmez.com
Evenements / agenda :
redaction@programmez.com

Dépôt légal : à parution - Commission paritaire : 1215 K 78366 - ISSN : 1627-0908

© NEFER-IT / Programmez, février 2015
Toute reproduction intégrale ou partielle est interdite sans accord des auteurs et du directeur de la publication.

Comment savoir si votre entreprise est DevOps ?



CommitStrip.com

Chaque semaine, de nouvelles aventures !

www.commitstrip.com



Sur abonnement ou en kiosque

Le magazine des pros de l'IT

Mais aussi sur le web



Ou encore sur votre tablette

L'INFORMATICIEN



DEVOXX FRANCE

La Conférence des Développeurs Passionnés

4^{ème} édition



Du 8 au 10 avril 2015
Paris, Palais des Congrès

3 jours de conférences et d'ateliers, avec plus de 140 sujets, 1800 personnes et un salon permanent de 30 exposants. Rendez-vous sur www.devoxx.fr pour plus d'informations.

Java, Web, Mobile, Agilité, Sécurité, Architecture, Future, Cloud et Devops.

Programme complet, informations et inscriptions : www.devoxx.fr

Twitter : @DevoxxFR



Événement organisé par Quantixx SARL - 75 rue du Faubourg Saint-Antoine, 75011 Paris - RCS Paris 80817989900014 - Société au capital de 3000 EUR - NAF 8230Z - Devoxx est une marque déposée par «The Java Community» - Tous droits réservés

