

NOUVELLE
FORMULE

Android | iOS 7 Windows Phone 8

*Tout sur iOS 7, gagner de l'argent avec Android,
Imaging SDK de Nokia*

Coding4fun

*Transformer
son Raspberry Pi
en serveur web*

C#

*Conseils
pour booster
son code*

Les salaires

*des développeurs
PHP*

Magento

*Gérer les frais
de port*

3D

*Utiliser Unity
avec PHP*

Nostalgie

*BeOS : celui
qui rêvait
d'être Apple*

CARRIÈRE

Développeur Cobol :

vraie-fausse bonne idée ?

Hacking

L'avenir du chiffrement web



Mensuel n°169 - Décembre 2013

M 04319 - 189 - F: 5,95 € - RD



Printed in EU - Imprimé en UE - BELGIQUE 6,45 €
SUISSE 12 FS - LUXEMBOURG 6,45 € DOM Surf 6,90 €
Canada 6,95 \$ CAN - TOM 940 XPF - MAROC 80 DH

WINDEV®

COMMANDEZ WINDEV® 19

ou WINDEV Mobile 19 ou WEBDEV 19

CHEZ PC SOFT ET RECEVEZ VOTRE MATÉRIEL AU CHOIX POUR «1 EURO DE PLUS»

Choisissez votre matériel :

(x2) Nouveau smartphone Samsung Galaxy S4
Ou choisissez la version «S4 mini» ou «S4 Active»
Configurations sur www.pcsoft.fr

(x2) Nouvelle tablette Samsung Galaxy Tab 3 10p
Ou choisissez la version 7 ou 8 pouces
Configurations sur www.pcsoft.fr

(x2) Nouvelle tablette Samsung Galaxy Tab 3 10p
Ou choisissez la version 7 ou 8 pouces
Configurations sur www.pcsoft.fr

WINDEV® 19
PLATEFORME INTÉGRÉE DE DÉVELOPPEMENT
DEVELOPPEMENT PLUS VITE

24" E3
intel XEON

DELL : PC Portable, PC Intégré ou Station de Travail
4 modèles au choix
Configurations sur www.pcsoft.fr

Samsung Galaxy Gear + Smartphone Samsung Galaxy Note 3
Configurations sur www.pcsoft.fr

Pour bénéficier de cette offre exceptionnelle, il suffit de commander WINDEV Mobile 19 (ou WINDEV 19, ou WEBDEV 19) chez PC SOFT au tarif catalogue avant le 20 décembre 2013: pour 1 Euro de plus, vous recevrez alors le ou les magnifiques matériels que vous aurez choisis. Offre réservée aux sociétés, administrations, mairies, GIE et professions libérales, en France métropolitaine. **L'offre s'applique sur le tarif catalogue uniquement.** Voir tous les détails et des vidéos sur : www.pcsoft.fr ou appelez-nous.
Le Logiciel et le matériel peuvent être acquis séparément. Tarif du logiciel au prix catalogue de 1.650 Euros HT (1.973,40 TTC). Merci de vous connecter au site www.pcsoft.fr pour consulter la liste des prix des matériels et les dates de disponibilité. Tarifs modifiables sans préavis.

Descriptif technique complet des matériels sur www.pcsoft.fr

OPÉRATION
1 POUR 1 EURO DE PLUS
JUSQU'AU 20 DÉCEMBRE

138 CM



Télévision SAMSUNG 138 cm LED 3D Full HD Réf UE55F6400
1.920 x 1.080. 200Hz DLNA Wifi 3xUSB
Gesture 4xHDMI 2xLunettes 3D

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

Fournisseur Officiel de la Préparation Olympique

www.pcsoft.fr

Programmez! 2.0

Bêta



« **Enorme !** » Nouvelle maquette, nouveau logo, nouveaux contenus, nouvelles rubriques, nouvel éditeur. Programmez ! change tout, ou presque. Mais notre philosophie ne change pas : le code, les outils, la technologie, votre avenir, votre devenir.

« **Le magazine du développeur** » Qui dit programmation et développement, dit développeur. Plus que jamais, nous vous remettons au cœur du magazine : parler de vous, de votre métier, de votre formation, de votre quotidien.

« **Mon ADN** » Depuis plus de 25 ans, je touche du code. Je me souviens de mes toutes premières fonctions programmées sur la vénérable Casio PB100, de mes premiers jeux sur Atari 520 ST. J'ai découvert Programmez ! au numéro 2, nous étions en 1998. Quelques semaines plus tard, j'écrivais ma première page dans ce magazine. J'anime Programmez ! en qualité de rédacteur en chef depuis 11 ans. Désormais, je le dirige en tant que directeur de la publication et actionnaire principal. Mois après mois, je suis toujours aussi excité par la technologie, étonné par la programmation et les prouesses d'un développeur ou d'un projet étudiant.

« **Toi aussi donne ton avis** » Nous avons voulu pour ce nouveau Programmez !, remonter aux sources du code, revenir aux fondamentaux. Pour y parvenir, nous avons besoin de vous, de votre soutien. Sans lectrices et lecteurs, Programmez ! n'existerait plus. Pour continuer à améliorer la revue, corriger les bugs, produire de meilleures itérations, vous faire vivre votre métier tous les mois, donnez votre avis. N'hésitez pas à proposer vos articles !

« **Bienvenue dans la matrice** » Et ce n'est qu'un début...

François Tonic
Directeur de la publication
& rédacteur en chef
ftonic@programmez.com

sommaire

74	Optimiser son code C#	64	Unity 3D et PHP	4	Les chiffres clés
16	Agenda	18	SQL Server		
28	Développer pour les terminaux mobiles	10	Une vie à la Silicon Valley #2		
58	Osez ou pas le Cobol ?	24	AR.Drone 2.0	82	Time machine
60	Coding4fun	22	Les développeurs du mois		
6	Actus du mois Google, Qt...	72	Magento	67	Guava 3e partie
		26	Sécurité & Hacking	14	Journal d'un dév #1
		77	Asynchronisme et C#		

À LIRE DANS LE PROCHAIN NUMÉRO
n° 170 en kiosque le 28 décembre

Le Cloud et le développeur
Comment le cloud computing influence le développement et le développeur ?

Tizen / Firefox OS
Les nouvelles plateformes pour smartphones et tablettes

Ecoles informatiques
Les cursus, les formations, les débouchés, le coût des études

900 millions \$

le coût de la gratuité de OS X et des logiciels pour Apple. Mais combien de milliards en + sur le matériel ?

+ 300 000

le nombre de lignes de code en Python dans OpenStack. Qui a dit que Python était un langage exotique ?

20 millions \$

c'est la levée de fonds de SkySQL. De quoi développer un joli SGBD !

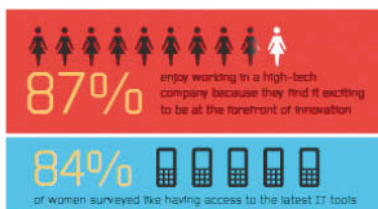
+ 30 %

c'est le taux de retour après achat de la montre connectée de Samsung. Pas assez d'usage, autonomie trop faible, applications inutilisables... En attendant une version 2...

1^{ère} amende

pour conduite avec les lunettes Google Glass. Légal ou illégal de conduire avec des lunettes connectées et de réalité augmentée ? Pas très clair...

Les femmes et l'informatique



Lenovo, constructeur de PC, a mené l'enquête : comment les femmes font bouger l'informatique.

L'innovation et les changements viennent souvent des femmes selon l'étude, mais le challenge est de trouver sa place dans un monde technique largement masculin.

► 87 % sont contentes de travailler dans une société de haute technologie.

► 84 % des femmes interrogées dans l'étude aiment utiliser les derniers logiciels et matériels informatique...

Quel est le plus beau CMS ?

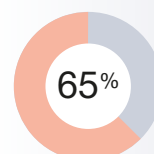
Smile, société de services informatique spécialisée dans l'open source, a publié son enquête sur les CMS open source. Et ils sont nombreux, il n'y a pas que Drupal dans la vie ! Les entreprises privées constituent toujours le gros du marché pour les CMS (+67 %) et étonnement, les entreprises de - 20 salariés représentent presque 55 % du marché...

Mais les utilisateurs connaissent-ils les CMS au-delà du nom ?

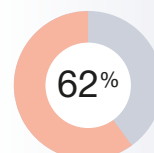
Drupal, Joomla et WordPress se détachent du lot avec 70 % des interrogés disant connaître un peu ou très bien un de ces CMS. Puis suivent SPIP, Typo3 et eZ Publish. Mais selon le pays cela peut changer.

85 % ont une opinion positive sur le CMS open source

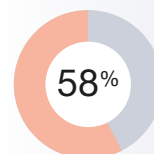
Quelles sont les fonctions attendues dans un CMS ?



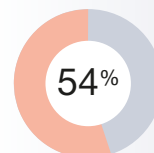
Structuration du contenu



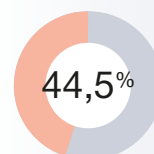
L'ergonomie du back-office



La qualité de l'architecture



Le processus de publication



L'expérience utilisateur en front-office

Le dur combat des données ouvertes en France

L'Open Knowledge Foundation a publié son classement sur les open data dans le monde. La France n'est pas très bien classée : 16e place ! Deux blocages : des données jugées comme fondamentales pas toujours disponibles, des redevances parfois trop élevées. Chaque pays possède un score selon les différentes données et leurs ouvertures. La France indique 510, contre 940 pour le Royaume Uni... Bref, nous avons encore beaucoup de progrès à faire !

Données	Score	Détails
Transport Timetables ⓘ	60%	
Government Budget ⓘ	90%	
Government Spending ⓘ	10%	
Election Results ⓘ	90%	
Company Register ⓘ	35%	
National Map ⓘ	35%	
National Statistics ⓘ	75%	
Legislation ⓘ	50%	
Postcodes / Zipcodes ⓘ	0%	
Emissions of pollutants ⓘ	65%	

Oui
 Non
 Incertain
 Pas de données

Miroir, miroir, quel est le langage le plus utilisé ?

L'index TIOBE donne chaque mois les langages les plus utilisés par les développeurs. Il s'agit d'un indicateur basé sur les recherches web. Des % à manipuler avec précaution.

Octobre 2013	Octobre 2012		Langage	en %	
1	1	=	C	17.246%	-2.58%
2	2	=	Java	16.107%	-1.09%
3	3	=	Objective-C	8.992%	-0.49%
4	4	=	C++	8.664%	-0.60%
5	6	↑	PHP	6.094%	+0.43%
6	5	↓	C#	5.718%	-0.81%
7	7	=	(Visual) Basic	4.819%	-0.30%
8	8	=	Python	3.107%	-0.79%
9	23	↑↑↑↑↑	Transact-SQL	2.621%	+2.13%
10	11	↑↑↑	JavaScript	2.038%	+0.78%

Pas de changements dans le peloton de tête : C, Java, Objective-C et C++ même s'ils connaissent des baisses. PHP remonte un peu. Au-delà, Groovy pointe à la 18e place (53e il y a un an) ! Et Cobol grimpe à la 20e place.

GREENIT ET ÉCO-CONCEPTION LOGICIELLE

8 GWh / an

14 700 Tb téléchargés par an, soit 3,6 millions de DVD

= consommation et le volume téléchargé par les utilisateurs pour les 100 sites web les plus visités en France. (Source Green Code Lab via webenergyarchive.com et audience mediamétrie)

1 milliard de \$

la somme investie dans BlackBerry qui n'est plus à vendre.

Presque 10 %

la part de marché de Windows Phone 8 en Europe

Google Glass 2

déjà prévue pour 2014, ou avant !

+ 4 milliards de \$

c'est le budget publicitaire de Samsung

+ 2 millions

de Raspberry Pi vendus

L'innovation vue par le gouvernement

Et vous, qu'en pensez-vous ?



LES PRINCIPALES MESURES DE LA

NOUVELLE DONNE POUR L'INNOVATION

4 objectifs



L'innovation par tous

- ▶ Des activités autour de la création d'entreprise à l'école pour développer le goût de la prise de risque et la culture de l'entrepreneuriat
- ▶ Le soutien à l'innovation dans toute sa diversité : de l'innovation technologique au marketing, au design, à l'innovation de procédés
- ▶ Le lancement d'un programme "Nouveaux argonautes" pour attirer les talents et faire venir en France les jeunes entrepreneurs de tous les pays
- ▶ La création d'une "bourse des nouveaux entrepreneurs" pour que chacun puisse créer son entreprise

L'innovation ouverte

- ▶ La création de laboratoires communs entre recherche publique et PME
- ▶ Une gouvernance régionale de l'innovation, au plus près des entreprises dans les territoires



L'innovation pour la croissance

- ▶ Une "industrie du financement de l'innovation" performante, disposant d'incitations en faveur du capital-investissement et de l'action des *business-angels*
- ▶ Un fonds souverain de propriété intellectuelle industrielle pour soutenir les savoir-faire nés en France
- ▶ Un nouvel interlocuteur de référence pour les créateurs de notre pays : le pôle Innovation de bpiFrance

L'innovation publique

- ▶ Evaluer les politiques d'innovation pour mieux identifier les voies de progrès.
- ▶ Une commission d'évaluation des politiques d'innovation sera créée au sein du Commissariat général à la stratégie et à la prospective
- ▶ Confier à la médiation interentreprises un rôle de médiateur de l'innovation
- ▶ Faire de l'innovation un des leviers de la modernisation de l'action publique



compte Twitter: @fil_gouv



gouvernement-fr.tumblr.com

FLEXIBILITÉ MAXIMALE POUR VOS PROJETS WEB



TOUT INCLUS

- Nom de domaine (.fr, .com, .info, .net, .org, .eu)
- Ressources illimitées : espace Web, trafic, comptes email et bases de données MySQL
- Système d'exploitation Windows ou Linux

APPS PERFORMANTES

- Plus de 140 Apps réputées (Drupal™, WordPress, Joomla!™, TYPO3, Magento®...)
- Support : des experts répondent à vos questions

OUTILS DE RÉFÉRENCE

- Software inclus : Adobe® Dreamweaver® CS5.5 et NetObjects Fusion® 2013
- 1&1 Mobile Website Builder
- PHP 5.4, Perl, Python, Ruby

MARKETING EFFICACE

- 1&1 Référencement Pro
- 1&1 Newsletter 2.0
- 1&1 WebStat
- Crédits Facebook®

INFRASTRUCTURE HIGH-TECH

- Disponibilité maximale grâce à la géo-redondance
- Connectivité > 300 Gbits/s
- RAM garantie : jusqu'à 2 Go
- 1&1 CDN powered by CloudFlare



1and1.fr

NVIDIA relève le défi de la programmation parallèle avec CUDA 6

Cela faisait longtemps que nous n'avions pas parlé de programmation parallèle et de l'utilisation des GPU dans son code. NVIDIA veut aider le développeur à utiliser plus souvent et mieux sa technologie CUDA, avec CUDA 6. Cette version a été annoncée mi-novembre.

Le fondateur met en avant trois grandes fonctions :

- **une Mémoire Unifiée** : elle simplifie le processus de programmation en permettant aux applications d'accéder à la mémoire des CPU et GPU sans qu'il soit nécessaire de copier manuellement les données

de l'une vers l'autre. La prise en charge d'une accélération du GPU avec un plus large éventail de langages de programmation est également facilitée.

■ la fonctionnalité

Bibliothèques « Drop-in » :

elle permet d'accélérer automatiquement jusqu'à 8 fois le temps de calcul des applications BLAS et FFTW en remplaçant simplement les bibliothèques de CPU existantes par les bibliothèques accélérées par le GPU.

■ **Optimisation multi-GPU :**

les bibliothèques BLAS et FFT GPU entièrement revues

permettent d'améliorer automatiquement les performances; elles peuvent concerner jusqu'à huit GPU pour un seul nœud, ce qui délivre une puissance atteignant jusqu'à neuf térafllops, et des performances deux fois plus précises par nœud, permettant ainsi de gérer des charges de travail encore plus importantes (jusqu'à 512 Go). L'optimisation multi-GPU peut également être utilisée avec la nouvelle bibliothèque BLAS.

Attention : CUDA 6 sera disponible début 2014.

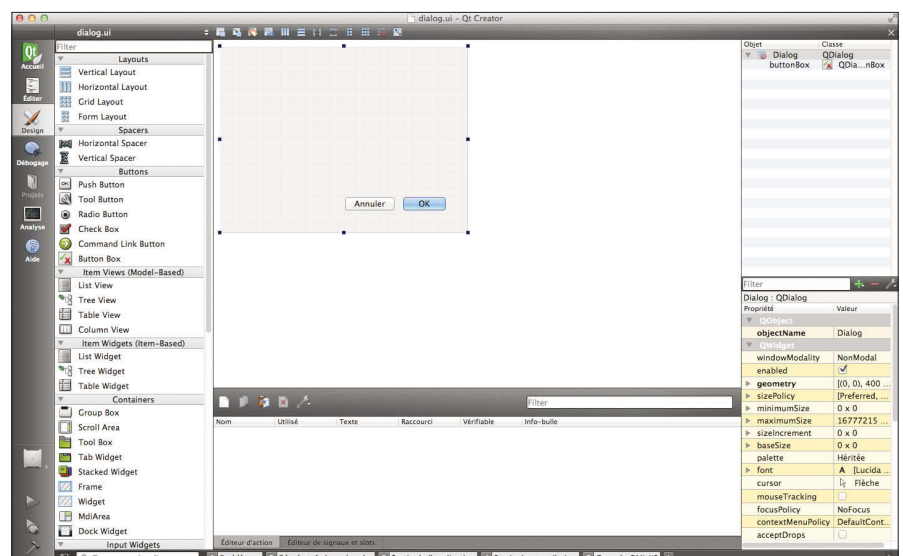
DIGIA PRÉPARE Qt 5.2

Qt, la librairie C++ pour créer des interfaces multiplateformes, va bientôt arriver en version 5.2. Cette version introduit d'importantes nouveautés, avec notamment un gros focus sur Android et iOS. L'éditeur annonce le support total de sa librairie sur ces plateformes. C'est un argument important pour les développeurs. Qt 5.2 introduira un nouveau rendu pour le Scene Graph : plus performant, moins consommateur de CPU. Autre grande nouveauté, le moteur JavaScript va être remplacé par un moteur spécifique Qt, en lieu et place au moteur V8. Un interpréteur JavaScript sera inclus pour répondre aux spécifications langages et aux règles du AppStore iOS. Cette version sonne aussi l'arrivée de Qt Creator 3.0. Creator est un IDE dédié Qt. Là

encore, la partie mobile en constitue la principale avancée (Android et iOS).

Ce positionnement mobile correspond totalement aux ambitions de Digia sur ce marché avec notamment l'annonce de Qt Mobile Edition, environnement complet de développement pour les terminaux mobiles. Cette édition s'appuie sur Qt 5.2. Elle cible iOS et Android, en attendant de nouvelles plateformes. Elle permet d'accéder à un service de stockage Cloud (Enginio Qt Cloud), d'accéder au support technique Digia, et de bénéficier de la licence commerciale. Qt Mobile Edition sera vendue à partir de 149 \$ / mois / utilisateur. Une version open source sera disponible.

Site : digia.com



NEW HOSTING

Backups quotidiens

Connectivité > 300 Gbits/s

Géo-redondance

Statistiques Web

NetObjects Fusion® 2013

PHP 5.4 **CDN**

Free Mode ou Safe Mode

MySQL **SEO**

Outil newsletter

Mobile Website Builder

Plus de 140 Apps

Drupal™, WordPress, Joomla!™, TYPO3, Magento®...

Performance garantie

Adobe® Dreamweaver® CS5.5 inclus



**PACKS TOUT INCLUS
POUR LES PROS**

À partir de

2,99
€ HT/mois
3,58 € TTC*



0970 808 911
(appel non surtaxé)

**OFFRE
SPÉCIALE**

.com
powered by VERISIGN

5,99
€ HT la 1^{ère} année
7,16 € TTC*



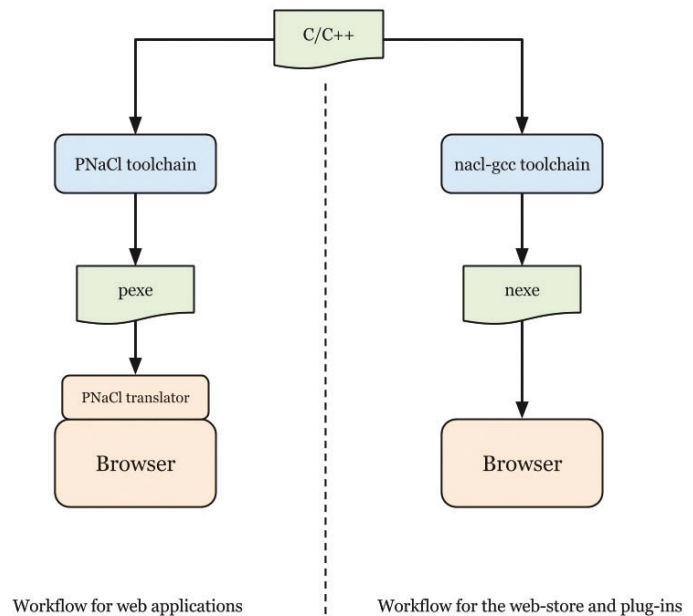
1and1.fr

* Les packs hébergement 1&1 bénéficient de la garantie 30 jours « satisfait ou 100 % remboursé ». En prépayant votre pack pour une période de 12 mois, vous bénéficiez d'un prix réduit. Ex. : 1&1 Basic à 2,99 € HT/mois (3,58 € TTC) au lieu de 4,99 € HT/mois (5,97 € TTC). Pas de frais de mise en service. Offre spéciale domaine : à l'issue de la 1^{ère} année, le .com est à son prix habituel de 6,99 € HT/an (8,36 € TTC). La garantie 30 jours « satisfait ou 100 % remboursé » ne s'applique pas aux noms de domaine. Conditions détaillées sur 1and1.fr.

Google veut du natif dans le navigateur

Le projet PNaCl de Google est l'abréviation de Portable Native Client. Objectif : exécuter du code/application natif dans Chrome, et ce, quelle que soit la plateforme matérielle (ARM, x86...). Le Native Client, selon les équipes chromium, permet d'utiliser la puissance CPU et GPU dans les applications Web. Idéal pour les applications gourmandes (photo, vidéo, 3D, etc.). PNaCl compile le code C et C++ dans une représentation intermédiaire et non par rapport à une architecture précise. Le bytecode (de type LLVM) est wrappé pour pouvoir devenir un exécutable portable et envoyé sur un serveur web comme un « vulgaire » site web ! C'est seulement quand Chrome récupère ce bytecode que ce code est optimisé selon l'architecture de la machine. Pour le moment, PNaCl est disponible uniquement sur Chrome. L'ambition est que le projet soit disponible sur d'autres navigateurs (via pepper.js). Un SDK est disponible. Décidément, le code natif fait le buzz du web !

Site : <https://developers.google.com/native-client/dev/>



JavaScript tout neuf pour Amazon

Amazon Web Services a dévoilé un nouveau SDK JavaScript qui permet d'accéder à de nombreux services Cloud d'Amazon (S3, SQS, SNS, DynamoDB). De plus, vous pouvez choisir JavaScript dans le navigateur et JavaScript dans Node.JS. C'est une grande nouveauté de ce SDK. Attention : accès en préversion. Pour en savoir plus :

<https://aws.amazon.com/fr/sdkforbrowser/>

Quel est le framework le plus rapide ?

Il existe de nombreux frameworks, pour tous les langages. Mais jusqu'à présent, il était parfois difficile de savoir lequel était le plus performant. TechEmpower tente de répondre à cette problématique. Il propose plusieurs catégories de tests : sérialisation JSON, requête simple et multiple, données, textes. Dans chaque catégorie, les perfor-

mances sont classées selon différents critères, par exemple la latence, les tables de données.

Plusieurs frameworks se démarquent : Grizzly, Falcore, Gemini... site : <http://www.techempower.com>

Compiler du C# en natif

En toute discrétion, Microsoft a présenté une nouvelle technologie lors du lancement de Visual Studio. Ce projet pourrait être le « Projet N » selon Mary-Jo Foley. L'objectif est très simple : booster les performances des applications C# grâce à une optimisation poussée lors de la compilation. Bref, être très proche du C++ ! Le compilateur reprendrait les mêmes caractéristiques qu'un compilateur natif C++.

Apple livre les fondations open source d'OS X 10.9

Peu après la sortie officielle d'OS X 10.9, Apple a livré sur son portail dédié les briques open source du système. Le système Unix de la Pomme utilise de nombreux composants ouverts. Par contre, iOS 7 n'était pas encore disponible sur le portail à la mi-novembre, ni les éléments de XCode 5.x. site :

<http://www.opensource.apple.com>

Vous connaissez WebGL et OpenCL, mais connaissez-vous WebCL ?

Le consortium Khronos travaille sur une nouvelle spécification : WebCL. L'objectif est de donner la possibilité aux codes JavaScript d'utiliser et de s'appuyer sur OpenCL pour faire de la parallélisation directement dans un navigateur web, en utilisant au mieux les capacités GPU et multi-cœurs des machines. Cette approche pourrait améliorer les performances de traitement pour l'audio-vidéo, la 3D, etc.

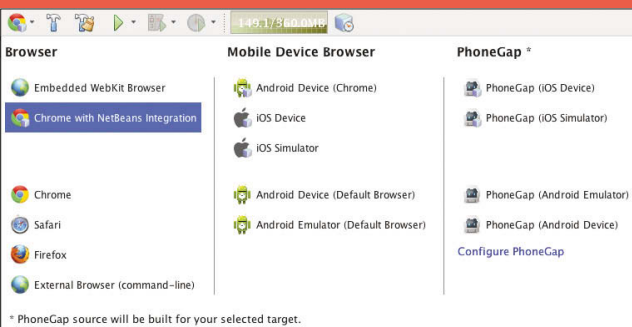
WebCL pourrait alors devenir une extension de HTML 5. L'initiative a été lancée en 2011. Samsung et Nokia avaient travaillé sur WebCL dès 2011.

Mozilla propose un prototype dans son moteur Gecko.

Site : <http://www.khronos.org/webcl/>

NETBEANS 7.4 : HTML5

Oracle lance la version 7.4 de son IDE Java/web, NetBeans. Un des focus importants de cette version est l'amélioration de l'éditeur HTML 5, notamment vers les terminaux Android et iOS. Parmi les améliorations : support de Cordova, support des navigateurs mobiles Android et iOS, moniteur réseau. La partie JavaScript s'améliore aussi avec le support de nouveaux frameworks (en édition) : AngularJS, ExtJS. Sur la partie Java, NetBeans commence à prendre en compte Java 8 (Profiles et Lambdas). Site : <https://netbeans.org>



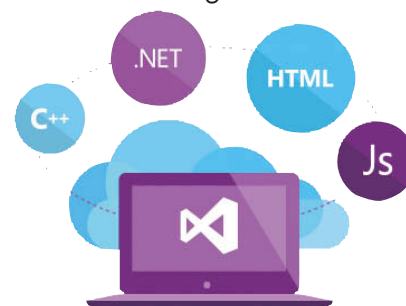
Visual Studio 2013 : services Cloud, cartographie du code, approche DevOps

Le 13 novembre dernier, Microsoft lançait mondialement Visual Studio 2013. Nous avons déjà publié plusieurs articles sur cette nouvelle version, notamment dans le n° 168 de Programmez !.

Visual Studio Online : non, ce n'est pas Visual Studio 2013 en ligne

Une des nouveautés a été l'annonce de Visual Studio Online (VSO), le nouveau nom de Team Foundation Services (bref VSO n'est pas Visual Studio en mode Cloud. Il s'agit de pouvoir gérer les projets et les codes directement sur le Cloud et de déployer les applications. VSO se compose de plusieurs modules. Ce qui a fait le plus de bruit est l'environnement de développement Web, Monaco. Au démarrage, Visual Studio sait si vous disposez ou non d'un profil VSO (attention : VS 2013 indique encore l'ancien nom de VSO). La connexion à son référentiel en ligne est très rapide. Vous gérez alors vos projets, les builds et les sources en équipe avec un référentiel en ligne. VSO est une extension naturelle de VS 2013. Le service est gratuit jusqu'à 5 utilisateurs. Il fonctionne aussi avec Visual Studio Express. VSO est inclus avec les abonnements MSDN. VSO comprend actuellement :

- Un contrôle de source hébergé
 - Un planning agile et les workitems (fonctions ALM)
 - Un service de build hébergé : attention, vous disposez de 60 minutes de build gratuites par mois
 - Un service de test de charge
 - Applications Insights (pré-version limitée) : comment savoir comment se comportent les applications ? Application Insights recueille les métriques, les données liées à votre application et permet d'avoir une vision 360° de l'application, de sa bonne santé (ou pas). Pour le moment, ce service supporte les applications .net et Java s'exécutant sur Windows Server, Windows Azure, le web et Windows Phone 8.
- Monaco est un éditeur de code accessible directement depuis son navigateur. C'est un outil idéal pour voir, modifier, créer du code dédié à un site web, par exemple, hébergé sur le service site web de Windows Azure (Monaco est limité à cette fonction pour le moment). Les



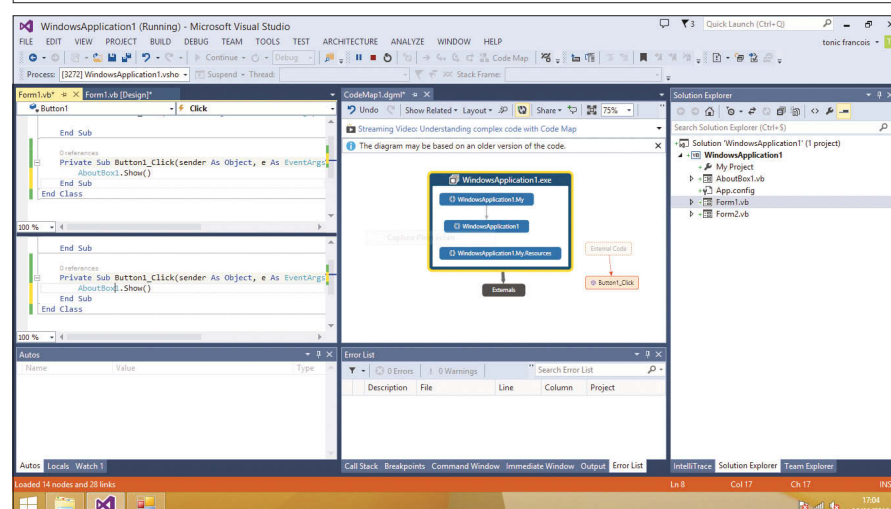
changements sont récupérés à la volée. Pour utiliser Monaco, il suffit d'activer « edit in visual studio online » (panneau configuration du site web). VSO est disponible en 3 éditions : Basic, Professional, Advanced. La tarification varie de 20 à 60 \$ / utilisateur / mois (tarif de lancement - 50 %). La gamme Expression est toujours disponible : TFS Express, Express pour Windows 8, Express pour web.

32 bits only

Visual Studio reste en 32 bits, ce qui limite les performances et l'utilisation des ressources mémoires des machines récentes. Mais peu à peu, Microsoft étend le support du 64 bits. Ainsi, avec le framework .Net 4.5.1, VS 2013 supporte la fonction « Edit and Continue » (permettant de debugger et de modifier le code quand celui-ci s'exécute) en 64 bits.

Site : <http://www.visualstudio.com>

Basic	<ul style="list-style-type: none"> ✓ Unlimited projects/repositories ✓ Agile planning ✓ Work item tracking ✓ Code commenting ✓ Work item chart viewing 	
Advanced	<p>= Basic capabilities +</p> <ul style="list-style-type: none"> ✓ Agile portfolio management ✓ Team rooms ✓ Work item chart authoring ✓ Request & manage feedback 	
Exclusive	<p>= Advanced capabilities +</p> <ul style="list-style-type: none"> ✓ Web test case management 	



Deux fonctions indispensables

CodeLens : de nouvelles informations s'affichent à la volée sur votre code.

Indication des références de classes et des méthodes, statuts des tests unitaires, de l'exécution des tests... Pratique et pas intrusif.

CodeMap : une fonction très pratique que le développeur devrait rapidement adopter. CodeMap permet de visualiser et de comprendre les relations entre les différents éléments de votre code. Cette cartographie sera vite indispensable dans les gros projets. Il peut aussi vous servir à comprendre un code ancien ou un code que vous ne connaissez pas. Il peut aussi vous aider à comprendre un bug. La création de la carte peut toutefois s'avérer assez longue.

Chapitre 2 : des révolutions en marche !

Le premier chapitre nous a permis de découvrir les origines, les étoiles ainsi que les domaines clés de la Valley ! Ceci nous ayant mis l'eau à la bouche, nous allons tenter dans ce second chapitre de mieux comprendre quelques domaines clés susceptibles d'engendrer de grandes révolutions sociétales à moyen terme, peut-être même à court terme !

Nous nous limiterons à 3 domaines : l'automobile, l'impression 3D et l'avènement de l'Internet Intelligent. Ce dernier est parfois considéré comme la mise en œuvre d'un « global brain ».

La Révolution automobile

Cette révolution, déjà en marche depuis de nombreuses années par différents acteurs à l'échelle mondiale, a tendance à prendre une sérieuse accélération au cœur de la Silicon Valley.

Ce domaine doit être scindé en deux parties :

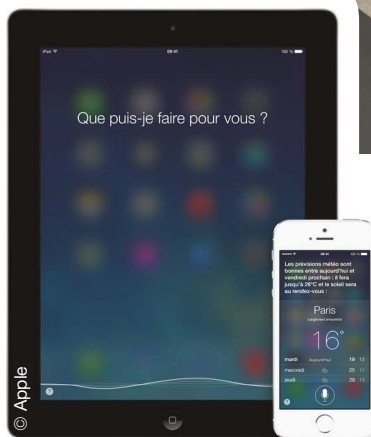
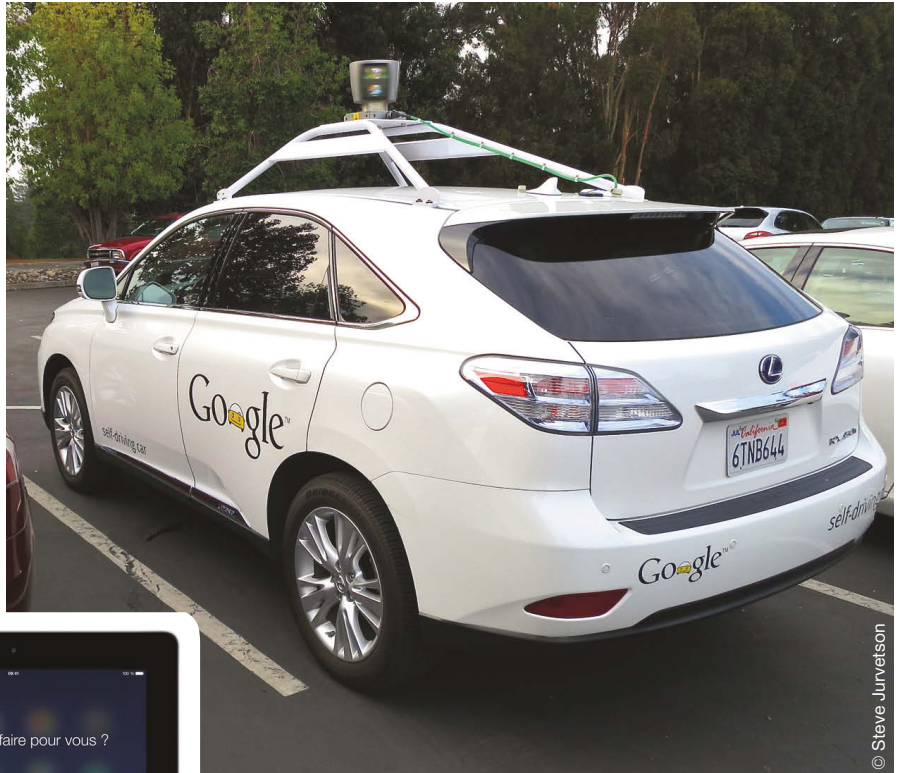
« Le 1er marché », entendez les véhicules neufs, et le « second marché », entendez les véhicules d'occasion, qui ont chacun leur propre cycle de vie et acteurs respectifs. Les ruptures en cours de ces deux domaines se situent autant sur le véhicule en tant qu'objet que sur le conducteur.

La voiture n'est plus une simple voiture

En ce qui concerne le véhicule, comme nous avons eu l'occasion de le comprendre rapidement lors du premier chapitre, Elon Musk, CEO de Tesla, est en passe de réussir le challenge que beaucoup d'autres marques automobiles n'ont pas réussi à franchir préalablement. Faire le saut énergétique : de l'énergie fossile à l'énergie électrique.

La Tesla Modèle S en est la preuve et l'écosystème déployé autour de celle-ci ne fait que rassurer les consommateurs. Vous pouvez ainsi effectuer sans trop de soucis des trajets de 400 km sans avoir à refaire le plein en énergie. Au cas où, vous auriez de toute façon la possibilité de côtoyer l'une de ces nouvelles stations électriques vous permettant, soit de faire le plein en moins de 20 minutes avec un système de charge rapide, soit encore de changer votre batterie de véhicule en un temps record, ce qui est l'équivalent de moins de la moitié du temps d'un plein d'essence traditionnel !

Elon Musk a clairement décidé de ne pas s'arrêter là et d'aller encore plus loin... au-delà de la technologie ! Effectivement, autour des stations électriques implantées aux USA, vous pouvez maintenant faire un San Francisco – Seattle en Tesla, Elon Musk a décidé d'implanter des



centres commerciaux de standing au regard de la population cliente de ces nouvelles voitures. Son tarif est à la hauteur de sa technologie : en moyenne 80 000 \$ (modèle Tesla S).

Non content d'accélérer la rupture technologique de l'automobile, Elon Musk poursuit en revendant sa technologie et son savoir-faire à des constructeurs et équipementiers automobiles mondiaux : il n'est pas rare de retrouver, derrière des grandes marques connues, le logo Tesla embarqué au plus profond des véhicules, sur le moteur électrique.

Et ce n'est pas tout, vous n'aurez certainement pas manqué les travaux de Sébastien Thrun à qui nous devons la Google Car ou comprenez la "Google Driverless Car" !

Après avoir accéléré la révolution électrique des véhicules, la Valley ne s'arrêtera pas en si bon chemin et poursuit à marche forcée une révolution d'un domaine vieux de plus de 100 ans et qui en définitive n'a vécu que très peu de changement... Il est donc venu le temps pour ce

domaine de passer du « Plus » au « Mieux » !!! Il suffit de regarder la valorisation boursière de Tesla pour comprendre l'appétit du marché.

Revenons à la Google Car et plus particulièrement aux travaux de Sébastien Thrun, considéré comme l'un des plus grands experts en Intelligence Artificielle avec Peter Norvig dont vous pouvez retrouver les cours maintenant gratuitement en ligne sur Udacity ! Cours que je ne peux que vous recommander ayant eu la chance de faire partie de leur cycle à Palo Alto en 2011.

Sébastien Thrun envisage non pas l'avenir automobile avec des véhicules volants mais bien avec des véhicules plus sécurisants; pour les sécuriser, des véhicules avec moins d'interventions humaines et plus d'interventions technologiques... Bien que la démarche puisse sembler au premier abord assez troublante, force est de constater que la technologie a une capacité de traitement et de fiabilité plus forte que celle de l'être humain, pour autant que nous ne nous trouvions pas face à un « écran de la mort » (ah les mythiques bluescreens) !

Et c'est sur ce point que les équipes de Google travaillent. A ce jour, la Californie étant le premier état au monde ayant autorisé les voitures

WINDEV®

ATELIER DE GÉNIE LOGICIEL
PROFESSIONNEL

DÉVELOPPEZ 10 FOIS PLUS VITE



NOUVELLE
VERSION

Opération «1 Euro»

Jusqu'au 20 décembre, commandez WINDEV 19 chez PC SOFT et recevez un superbe matériel pour 1 Euro de plus: télévision SAMSUNG 138 cm, ordinateur DELL, Galaxy S4, Galaxy Tab 3...



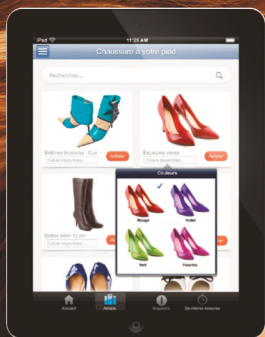
Applications natives

Linux, Mac, Internet, Intranet,
Windows 8, 7, Vista, XP...,
Cloud, Android, iPhone, iPad :
vos applications sont compatibles

Environnement de développement professionnel, intégralement en français (logiciel, documentations, exemples). Développez facilement vos applications et vos sites. La facilité de développement avec WINDEV est devenue légendaire: vos équipes développent plus vite des applications puissantes, la qualité de vos logiciels est par essence élevée, le nombre de fonctionnalités automatiques est impressionnant.

Vous délivrez plus vite vos logiciels, plus rapides et plus robustes.

VERSION
EXPRESS
GRATUITE
Téléchargez-la !



DEMANDEZ VOTRE DOSSIER GRATUIT

260 pages - 100 témoignages - DVD Tél: 04.67.032.032 info@pcsoft.fr

www.pcsoft.fr



Fournisseur Officiel de la Préparation Olympique

919
NOUVEAUTÉS

Elu
«Langage
le plus productif
du marché»



sans conducteur à circuler sur la voie publique, il est très fréquent de croiser des DriverLess Car sur les axes majeurs de la Silicon Valley (la 101 ou 280) permettant de valider dans la vraie vie les outils et systèmes mis en œuvre lors de la victoire de l'équipe de Sébastien Thrun au DARPA Grand Challenge en 2005.

Bientôt, nous pourrions ainsi rencontrer encore plus de véhicules sans conducteur ou à conduite assistée en période de fête par exemple... Ceci pourra préserver plus de vies au regard des chiffres d'accidents restant encore et toujours trop élevés ! Ce point étant à l'origine des travaux de Sébastien Thrun ayant perdu son meilleur ami dans un stupide accident de voiture et s'étant juré de consacrer sa vie à réduire sans cesse le nombre de tués !

Cette seconde révolution du véhicule, l'avènement de l'électronique intelligente au sein de la voiture, ne fait qu'ouvrir un nouveau monde pour la seconde rupture de l'automobile après le véhicule... la rupture du conducteur et de ses usages.

Pour savoir comment cette révolution va faire évoluer le conducteur, flashez le QRCode !



La révolution 3D : le pouvoir au peuple !

Attaquons maintenant brièvement les deux autres sujets de cette chronique sur lesquels nous reviendrons plus en profondeur prochainement. L'impression 3D, ou tridimensionnelle, trouve ses origines dans les années 60 avec Arthur Charles Clarke,

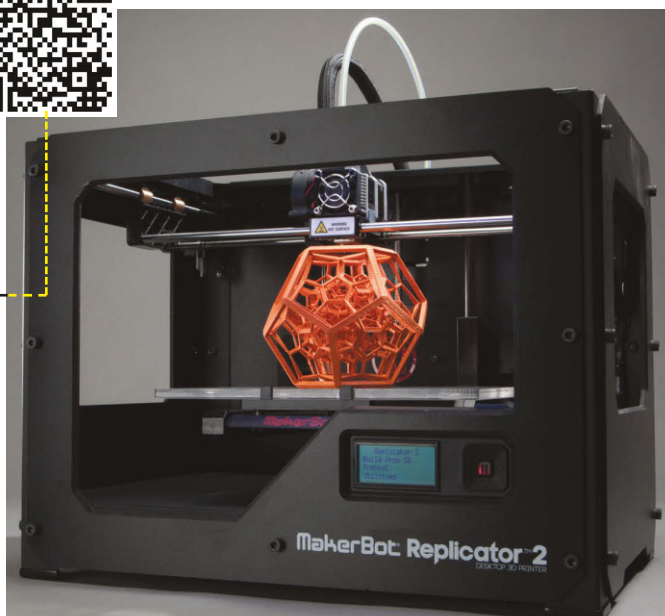
inventeur et écrivain de science-fiction britannique, mentionnant la machine à répliquer qui aurait pour effet de profondément révolutionner notre civilisation ! Et je pense que l'expression ou le terme révolution est faible ! L'impression 3D popularisée par MakerBot avec son Replicator I et surtout le II. Cette révolution est « énorme » et impactera en profondeur nos sociétés encrées dans un modèle arrivé au terme de son apogée !

Cette capacité de prototypage et de production alors réservée aux plus grands est maintenant accessible à tout un chacun !

L'impression 3D se diversifie ou se diversifiera autant que nous avons de domaines de produits consommables ou non, de l'objet quotidien (ustensiles de cuisine, jeux d'enfants, chaussures, objets décoratifs, gadgets, pièces de moteur, ...) mais aussi à des domaines moins attendus et plus en attente encore de rupture comme l'impression d'habitats, l'impression d'œuvres d'art, l'impression d'exosquelettes

afin de créer la rupture du fauteuil roulant, l'impression de Robot Humanoïdes, l'impression d'organes et bien d'autres encore ... Je vous invite à parcourir le site dédié à la centralisation des ruptures par domaines engendrées par l'impression 3D : <http://3dprintingindustry.com/>

La première force de l'impression 3D se trouvant ainsi dans « l'information » et non plus dans l'objet en tant que tel ! Notre société devient alors une société d'abondance au travers de la dématérialisation... Elle se préoccupe de moins en moins des notions de temps (temps de production, temps de livraison, ...), de coût (les productions en petites quantités), de rareté (quid d'une pièce où la production est stoppée) ou d'espace (comment faire livrer une prothèse au sein d'un village isolé dans une contrée éloignée, ...). La seconde force est la créativité



rendue aux mains de tous, aux mains du peuple ! Terminé le temps où vous ne pouviez imaginer démarrer la création de votre propre gamme d'objets, d'outils, d'œuvres en 3D... Tous ces domaines vont considérablement se voir impactés par le prix diminuant des impressions 3D, et, mieux encore par l'ouverture de boutiques d'impression 3D dans les villes, ce qui fut le cas il y a quelques semaines à Oakland avec l'ouverture du premier magasin où vous pouvez aller imprimer votre création 3D !

Et ceci sans imaginer ce qui se passera au terme de la campagne de Makerbot qui pousse fort, voire très fort afin de mettre à disposition une imprimante 3D dans chaque classe d'école américaine !

La révolution Global (big) Brain : ne surfons plus idiot

La révolution du Global Brain (ou l'avènement du Web Intelligent) est un des sujets forts de la

Valley. Nous disposons déjà de plusieurs produits et outils dédiés à ces nouveaux usages : Siri, Google Now, Watson, ainsi que les sociétés de big data, de l'ontologie, de la sémiotique... Le Web Intelligent se définit comme la « zone » où s'appliquent l'intelligence artificielle, le big data et les technologies de l'information au domaine du Web afin d'en produire les nouvelles générations de produits, services et Frameworks basés sur les protocoles Internet. Ce terme trouve ses origines dans une publication nommée « Web Intelligent » par Ning Zhong, Jiming Liu, Yao, Y.Y., Ohsuga, S. et présenté pour la première fois lors de la Computer Software and Applications Conference en 2000.

Cette révolution encore peu compréhensible est certainement l'une des plus fortes pour nos sociétés. Des acteurs majeurs comme Google,

Apple ou Facebook ont bien compris les enjeux et s'empressent d'acquiescer les startups ou visionnaires du domaine comme Ray Kurzweil chez Google au regard des enjeux business qui en seront touchés puisque nous parlons clairement d'une rupture de l'industrie de la consommation dans son ensemble !

Comme vous l'aurez compris, la place du développeur y est plus que centrale ! Nous reviendrons ainsi sur ce thème majeur, ses implications technologiques et migrations de compétences ou connaissances pour résoudre les challenges lors de notre prochain chapitre.

Et après ?

Tout comme après notre premier chapitre, nous pouvons constater que celui-ci est riche en ruptures en tous types et ce ne sont pas les seuls ! L'automobile, l'impression 3D ou encore le Web Intelligent ou Global Brain ne sont que des planètes dans une galaxie très dense que forme la Valley, voire notre civilisation.

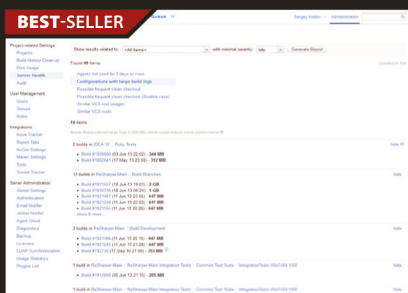
Nous poursuivons notre exploration de ces domaines dans notre prochain épisode ainsi que sur les implications du Web Intelligent, du Big Data pour le développeur, et dans différents domaines comme celui de la santé en pleine révolution à Palo Alto !

Gregory

xBrainSoft Inside

xBrainSoft, startup basée à Euratechnologies et au sein de la Silicon Valley, met à disposition une plate-forme permettant d'intégrer et configurer au sein de tout système la compréhension vocale et les assistants personnels.



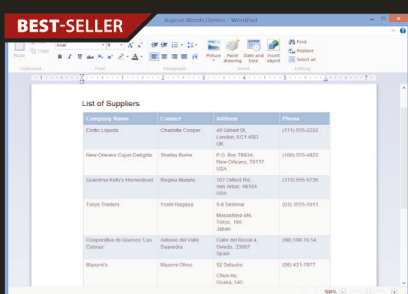


TeamCity à partir de € 1 763



Serveur d'intégration continue prêt à l'emploi, extensible et conçu pour les développeurs.

- Configuration rapide et intuitive des projets de builds
- Gestion intelligente de serveurs (usage des disques/état serveurs)
- Analyses et couvertures de code intégrées
- Extensibilité simple via REST API, messages des scripts de builds et API ouverte
- Support intégré pour une vaste gamme de technologies incluant Java, .NET, Ruby, Objective-C et Android



Aspose.Words for .NET à partir de € 731



Lisez, modifiez et écrivez des documents Word sans Microsoft Word.

- Création de documents, manipulation du contenu/formatage, puissante capacité de fusion de courrier et exportation en DOC/HTML
- Accès détaillé à tous les éléments d'un document par programmation
- Support les formats de fichiers: DOC, DOCX, WordprocessingML, RTF, HTML, OOXML, OpenDocument, PDF, XPS, EMF et EPUB

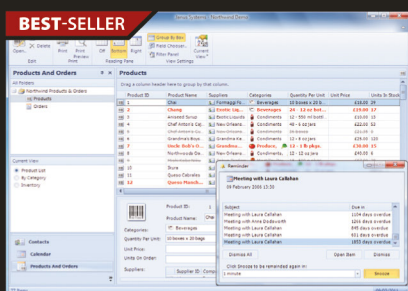


DevExpress Universal Suite à partir de € 1 611



400+ outils et contrôles WinForms, ASP.NET, WPF, Silverlight et Windows 8.

- Exploitez votre base de codes pour développer des applications tactiles multiplateformes
- Inclut un tableau de bord de visualisation interplateformes et un serveur de rapports
- Nouveaux contrôles DevExpress incluant Tableur et Carte
- Codage, débogage et refactorisation avec CodeRush pour Visual Studio
- Inclut la nouvelle galerie unifiée de modèles applicatifs DevExpress



Janus WinForms Controls Suite V4.0 à partir de € 678



Ajoutez des interfaces de style Outlook à vos applications .NET.

- Vues ruban, grille, calendrier, et barres chronologique/raccourcis
- Nouveau – Style visuel Office 2010 pour tous les contrôles
- Nouveau – Support des profils client Visual Studio 2010 et .NET Framework
- Janus Ribbon ajoute Backstage Menus et la fonctionnalité onglet comme dans Office 2010
- Prend désormais en charge la sélection de cellules multiples

#1 : dur matin

Je me réveille ce lundi matin, il est 10h et il pleut. Saleté de temps. De toute façon, dès octobre, c'est toujours la même chose : de la pluie ! De plus je suis en retard. Quoi qu'« officiellement » pas vraiment, mais je sais que j'ai un travail à finir pour vendredi. J'enfile le tee-shirt que je vois. Ma casquette, mon sac à dos. Je suis prêt pour aller au travail.

Ne pas oublier son badge !

Je sais bien que ça fait geek de s'habiller comme ça ; ou nerd. En fait je ne sais plus le terme exact à utiliser. J'oublie de prendre mon badge et me jette à toute allure sur mon vélo. Je vois des écureuils sur la route, je me dis que ce ne sont que des rats avec une queue velue et je me sens chanceux de ne pas être tombé sur les renards/biches, voire même un ours que j'ai déjà pu voir sur le campus Microsoft !

Je traverse en quelques minutes cette micro-ville qu'est l'immense campus Microsoft (Seattle). J'essaie de faire le malin sur un rebord de trottoir, mais ce vélo ne saute pas. J'avais pourtant vu un mec avant moi qui y arrivait très bien, sauf que le mien s'écrase complètement sur le petit rebord et je me tape la honte. Mais personne ne se moque. Remarquez, les gens ne se retournent pas non plus sur la Ferrari qui vient de passer à petite vitesse. Elle appartient à un des principaux développeurs de mon équipe. Il ne manque qu'un beau sticker Microsoft...

J'arrive à mon bureau, trempé, après avoir tenté une négociation infructueuse pour ne pas avoir le « badge of shame » qui est une sorte d'auto-collant qu'on vous colle sur le tee-shirt. Je prends une canette (gratuite) comme petit dej' (excellent régime alimentaire) et je me jette sur

mon canapé de mon bureau.

Ce bureau je l'aime bien. Sur les murs, j'y ai accroché tellement de choses censées représenter les bonnes pratiques que je ne les lis même plus. N'étant pas senior, je n'ai pas le droit à un bureau avec fenêtre, mais j'ai une ventilation (ouf !). D'ailleurs, je suis tellement fainéant qu'au lieu de demander de baisser la ventilation, je l'ai juste scotchée. Maintenant elle fait un bruit d'avion de chasse dans les bureaux d'à côté. Ça me fait marrer.

Avant le code, la réflexion...

Je commence à réfléchir à la journée qui démarre. Il est 11h. J'ai devant moi mon tableau blanc. Je dois architecturer une nouvelle fonctionnalité. Je sais que je dois faire attention à ce que je produis, c'est pourquoi je suis là. Je dois faire face à l'ambiguïté et mener à bien la mission qui m'a été confiée. Etant responsable de la tâche qui m'a été confiée, j'en suis le responsable. Faire mieux que bien !

Je dois faire au-delà de ce que l'on attend de moi. Je dois délivrer du code de production. Il doit être performant, robuste et maintenable. Si

mon code ne respecte pas ces principes, il sera mal vu, je serai mal vu pour ma revue de fin d'année (bilan annuel) et au-delà. Surtout, ce code ne passera pas l'étape du « code review ».

Cette fameuse étape où deux seniors, avec 12 ans d'expérience, relisent mon code, ainsi que mon testeur. Si tout se passe bien, le code est « tamponné », attestant que j'ai



le droit de l'envoyer dans le système de production.

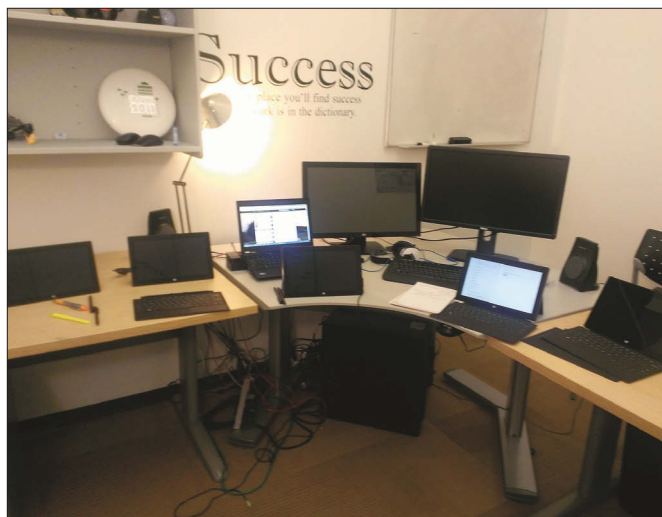
J'ai bien essayé de mettre mon nom dans un commentaire une fois. Histoire d'avoir ma trace dans Windows 8. Ils l'ont vu... Ils voient tout.

J'ai parfois l'impression qu'ils connaissent mieux mon code que moi-même. Ils sont précis, efficaces. Ils frappent là où il faut, quand il le faut. Je les admire. Ils sont ingénieurs, un mélange entre un scientifique et un artiste qui sait faire les bons jugements.

Alors je dessine, je dessine ce que je dois concevoir. Je me torture l'esprit. Est-ce assez encapsulé ? Ai-je privilégié la simplicité plutôt que la complexité ? Ma solution est-elle totalement idéale ? Ai-je appliqué tous les principes d'architecture que l'on m'a enseignés ici ? L'algo que je devrais écrire aura-t-il une complexité optimale ? Après quelques heures, je pense tenir quelque chose. Mais il est déjà 19h. Je coderai demain...

Chaque mois, je vous raconterai dans cette rubrique mes histoires sur le code, les USA, l'immigration, bref tout ce qui m'entoure. Je m'appelle Julien, je suis développeur pour Windows, XBOX et Surface, chez Microsoft. Je travaille dur pour un jour avoir un grand impact sur ce monde. C'est ma seule motivation : que l'on ne m'oublie pas (un peu trop mégalomanie ?). Mais comme il n'y a pas de titan à combattre, que les monstres n'existent pas et que je n'ai pas encore la capacité d'être l'élément déterminant pour remporter une des batailles de l'humanité, je fais du développement. Et j'aime ça !

Julien Dollon - <http://julien.dollon.net>



OFFREZ-VOUS L'UNIVERS DU DÉVELOPPEMENT !

1 an 11 numéros

50€

seulement (*)

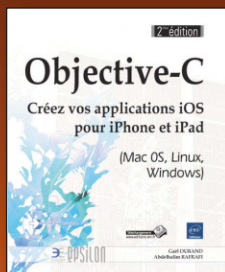
2 ans 22 numéros

79€

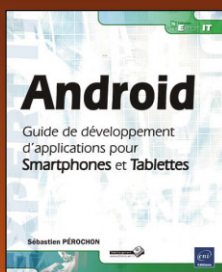
seulement (*)



Version numérique : 29,26 €



Version numérique : 40,50 €



Version numérique : 29,26 €

+ Un «livre numérique» au choix des Editions Eni

réservée à la France
Métropolitaine

Valeur des 11 numéros du magazine : 65,50€ + Valeur du livre : 29,26 € à 40,50 € = Valeur totale : 94,76 € à 106 €

(*) Tarifs France métropolitaine

Toutes nos offres sur www.programmez.com

Oui, je m'abonne

à retourner avec votre règlement à
Programmez, 17 route des Boulangers 78926 Yvelines cedex 9

Attention : l'offre avec livre Eni est limitée et réservée à la France métropolitaine

☐ **Abonnement 1 an au magazine + 1 « Livre Numérique ENI » :** 50 € (au lieu de 65,45 €, prix au numéro + valeur du « livre numérique ») ☐ Abonnement seul : 49 €

☐ **Abonnement 2 ans au magazine + 1 « Livre Numérique ENI » :** 79 € (au lieu de 130,9 €, prix au numéro + valeur du « livre numérique ») ☐ Abonnement 2 ans seul : 78 €

Livre à choisir : ☐ Html 5 CSS 3 et JavaScript ☐ Objective-C ☐ Android

Détails sur www.programmez.com/abonnement.php

Tarifs France métropolitaine

☐ M. ☐ Mme ☐ Mlle Entreprise : _____ Fonction : _____

Prénom : _____ Nom : _____

Adresse : _____

Code postal : _____ Ville : _____

Tél : _____

E-mail : _____ @ _____

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

☐ Je souhaite régler à réception de facture

(Attention, e-mail indispensable pour les archives sur internet et offre Eni)



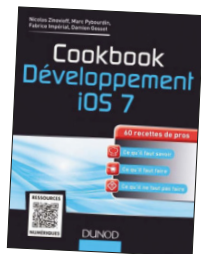
Le coin lecture

Internal Hacking et contre-mesures en environnement Windows (Editions Eni)

La sécurité demeure un thème central de l'informatique moderne. Si l'auteur revient sur les classiques du genre, il propose aussi d'aller plus loin : fabriquer ses propres outils de hacking. Même le responsable sécurité a le droit de s'amuser un peu. L'ouvrage aborde aussi la prise de contrôle, les méthodes pour effacer la moindre trace. A lire !



Cookbook développement iOS 7 (Dunod)



iOS 7 est disponible depuis quelques semaines. Pour le développeur, les changements sont parfois très importants. Cet ouvrage est conçu pour faire gagner du temps aux développeurs :

stockage, navigation, géolocalisation, gestion des photos, des comptes e-mails, du calendrier, des notifications et autres applications, interaction avec les services à distance... Il livre tous les secrets pour concevoir facilement une application avec iOS 7, le tout dernier système d'exploitation d'Apple. La version Android est également disponible !



décembre

LES JOURNÉES SQL SERVER

2 & 3 décembre à Issy les Moulineaux

Le groupe utilisateur de SQL Server organise deux journées SQL Server très riches : une quarantaine de sessions autour de l'administration et du développement SQL d'un côté, et de la Business Intelligence de l'autre, à travers un parcours « Nouveautés, Infrastructures et Cloud » la première journée, et un parcours « Perfectionnement, Usages et Outils » la seconde ! site : <http://guss.pro>

Les Journées SQL Server 2013

Le 2, 3 Décembre 2013, Paris - @GUSS_FRANCE - www.GUSS.pro

APIDAYS

4 & 5 décembre à Paris

Le monde des API évolue tout le temps. Les API sont désormais au cœur des entreprises et dans la vie quotidienne. APIdays revient en France avec plus de 50 speakers internationaux pour l'édition globale de la conférence dédiée aux API. Cette année, deux axes seront mis en avant : l'expérience utilisateur et les API dans les processus internes des entreprises. La dimension développeur n'est pas pour autant oubliée. De nombreuses conférences techniques et stratégiques auront lieu durant les deux jours.

Site : <http://paris.apidays.io/2013/>

HACKATHON PERL

6 & 8 décembre à Paris

Vous voulez faire du Perl ? Un hackathon Perl aura lieu début décembre à Paris. Même si

vous ne connaissez pas Perl, c'est l'occasion de découvrir ce langage historique.

Site : <http://www.agendadulibre.org/showevent.php?id=8048>

ANDROID LIVECODE PARIS

9 décembre

Du code sur scène ! C'est tout le défi de Android Livecode, une soirée 100 % code avec plusieurs experts mobiles. Cette soirée tournera autour de la 3D, d'Android et de Java.

Site : <http://goo.gl/9GMuVB>

LEWEB '13

10-12 décembre

Evènement du monde web à Paris du 10 au 12 décembre. LeWeb s'est imposé comme la conférence web française. Cette 10e édition annonce la couleur : les prochaines 10 années ! Futur, prospective, visions...

Site : <http://paris.leweb.co>

JUG PARIS

12 décembre à Paris

Le JUG Paris organise sa dernière soirée de l'année, le 12 décembre avec classloaders et optimisation JVM ! La soirée promet du très lourd avec Tomcat, TomEE, Glassfish, Jetty, WebLogic, etc. Comme toujours, les intervenants sont parmi les meilleurs du monde Java. La soirée suivante aura lieu le 14 janvier 2014 ! Site : <http://www.parisjug.org>



Agenda

Botconf'13 : Entièrement dédiée à la lutte contre les Botnets, elle est organisée par un comité scientifique indépendant et international. En Anglais. Lieu : Nantes, les 5 et 6 décembre. Site : <https://www.botconf.eu>



janvier 2014

CHALLENGE FIC 2014 - 21 & 22 janvier

Dans le cadre du salon FIC, le challenge FIC est ouvert à toutes celles et à tous ceux qui disposent de compétences techniques en matière de sécurité informatique. Organisé en partenariat avec l'association ACISSI, avec les étudiants de la licence CDAISI de l'IUT de Valenciennes et l'Ecole d'ingénieur EPITA, et avec le parrainage du Réseau Cyberdéfense de la Réserve Citoyenne, le Challenge FIC consiste en deux séries d'épreuves informatiques de 4 heures. Date limite des inscriptions : 15 décembre 2013. Site : <http://www.forum-fic.com>

DANS LA JUNGLE DU CLOUD, MIEUX VAUT CHOISIR LE BON PARTENAIRE.

b i z ☒ L O V Crédits photos : © Gettyimages / John Lund - John Lund / Blend Images / GraphicObsession



ET SI VOUS PASSIEZ À LA PUISSANCE CLOUD ?

VAR

Distinguez-vous de vos concurrents en valorisant votre offre.

SSII

Profitez d'infrastructures IaaS déployées en un clin d'œil.

ÉDITEURS

Passez au SaaS en vous appuyant sur notre savoir-faire.

PROGRAMME PARTENAIRE ARUBA CLOUD

- 2 niveaux de marque blanche disponibles.
- Gestion simple, souple et performante de l'infrastructure (publique, privée ou hybride).
- Modélisation et activation immédiate de votre datacenter virtuel dans le pays de votre choix.
- Interface client totalement personnalisable.
- Facturation en "Pay as you go".
- Finesse dans la gestion des droits utilisateurs.

Contactez-nous

Aruba, le bon partenaire pour bénéficier de la puissance d'un acteur majeur qui considère que chaque client, dans chaque pays, est unique. **MY COUNTRY. MY CLOUD.**



arubacloud.fr | TÉL : 0810 710 300
(COUT D'UN APPEL LOCAL)

SQL Server 2014 : la nouvelle base de données de Microsoft

L'édition 2014 de SQL Server se concentre sur les grandes tendances technologiques : base en mémoire, performance et montée en charge, fonctionnement hybride (serveur et Cloud), Hadoop (avec la distribution HDInsight, propre à Microsoft) et les services de Business Intelligence de Windows Azure.

Microsoft se place en concurrence directe avec Oracle et IBM mais aussi avec les bases open source telles que MariaDB et PostgreSQL. Aujourd'hui, pour accélérer les traitements et booster les performances, on met du « in-memory » partout. Il s'agit de mettre le moteur de la base et les données directement en mémoire. Avantage : des performances décuplées, mais il est nécessaire de disposer de machines capables d'accueillir une grande quantité de mémoire vive ou de travailler uniquement en SSD. Ce qui ne va pas sans conséquences sur le coût d'exploitation et les problèmes de performances de la mémoire (RAM ou SSD) sur la durée. Les performances d'un SSD se dégradent dans le temps.

Cette approche permet de faire les mises à jour de données non plus en synchronisation asynchrone (pour ne pas charger le réseau et bloquer éventuellement les bases) mais quasiment en temps réel quand cela est nécessaire. Cependant, tout le monde n'a pas besoin de

faire du quasi-temps réel. SQL Server 2014 met le tout mémoire sur plusieurs fonctions : PowerPivot (gros consommateur de puissance), utilisation d'un buffer pour les SSD et les traitements de requêtes.

Autre axe important, la montée en charge. Aujourd'hui, SQL Server 2014 est capable de fonctionner sur 640 processeurs, 64 processeurs virtuels (par instance), 1 To de mémoire vive par instance virtuelle. L'éditeur introduit la gouvernance des Entrées/Sorties (dans le moteur de gouvernance des ressources). Ainsi, vous pouvez utiliser au mieux les ressources machines et réseaux, et répartir vos différentes bases selon les besoins. Une base nécessitant une performance immédiate recevra la priorité des ressources (CPU, mémoire, E/S), alors qu'une base de données plus froide fonctionnera en ressources « minimales ». Dans le même esprit, la nouvelle base renforce la haute disponibilité avec des fonctions AlwaysOn améliorées, et une réplication « simplifiée ».

QUOI DE NEUF CHEZ MARIADB ?

MariaDB, un des forks de MySQL, continue d'évoluer; les équipes travaillent activement aux évolutions de la version 10. La 10.0.5 est actuellement en bêta. Cette version inclura des corrections de bugs et quelques nouveautés (réplication parallèle, kill query id). Elle complètera aussi certaines compatibilités et fonctions provenant de MySQL 5.6.10 (MariaDB 10 se base sur la branche MySQL 5.6).

Pour en savoir plus :

<https://mariadb.com/kb/en/plans-for-10x/>

Cloud par essence

SQL Server 2014 ouvre assez largement la porte au Cloud Computing, notamment dans le contexte hybride (serveurs locaux + services Cloud). Ainsi, vous pouvez faire du backup et du snapshot de base vers Windows Azure. En cas de problème avec votre SQL Server, vous pourrez restaurer via les services Azure et retrouver vos données. Autre utilisation, utiliser les services de Cloud en bursting, c'est-à-dire étendre les ressources sur le Cloud au lieu d'acheter de nouveaux serveurs.

Documentations et ressources techniques pour le développeur : <http://msdn.microsoft.com/en-us/sqlserver/>

Hekaton : In-Memory OLTP

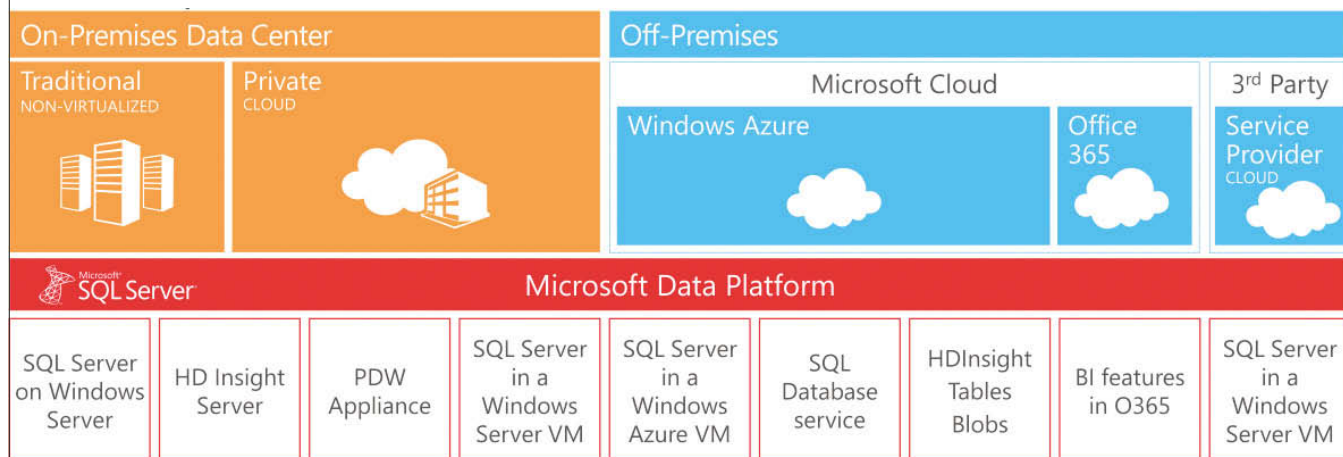
Hekaton est un nouveau moteur de base de données inclus dans SQL Server 2014. Il se dédie aux traitements OLTP. Il est désigné pour être concurrent dans le fonctionnement et les accès. OLTP signifie Online Transaction Processing ou

traitement transactionnel en ligne. Il est particulièrement intéressant quand il faut modifier les données en temps réel en exigeant une disponibilité critique des informations. Son accès se fait en lecture/écriture. Il s'oppose à l'approche OLAP

(online analytical processing) qui est en principe accessible uniquement en lecture. OLTP n'est pas une nouveauté dans SQL Server et les bases de données, mais Hekaton le transpose en mémoire (In-Memory). Ce moteur est

directement intégré à SQL Server et à son moteur relationnel. Hekaton supporte les tables en mémoire et les tables (stockées) sur disque, améliore l'exécution et les performances des procédures stockées T-SQL.

Ce moteur fonctionne uniquement en mode 64 bits.



Les outils des Décideurs Informatiques

*Vous avez besoin d'info
sur des sujets
d'administration,
de sécurité, de progiciel,
de projets ?
Accédez directement
à l'information ciblée.*

Cas clients

Actu triée par secteur | Avis d'Experts



Actus / Evénements | Newsletter | Vidéos

LE MAGAZINE DES DÉCIDEURS INFORMATIQUES

SOLUTIONS & LOGICIELS

NOVEMBRE 2013 N°43
www.solutions-logiciels.com

SECURITE
Gérer les **mobiles** comme des PC p.10
SIEM : gestion de vulnérabilité & contre-mesures p.14

IT des COLLECTIVITÉS LOCALES p.42

Data centers
Comment juger leur qualité ? p.20

Dématiser les FACTURES p.36

Comprendre le contrat Amazon Web Services p.28

RH et Mobilité p.30

Telety Group
Stéphane Duproz, Directeur Général de Telety Group France
Numéro 1 européen*
"La qualité de nos data centers est notre principal différentiel compte tenu des impératifs de performance et de criticité actuels" p.26

www.solutions-logiciels.com

☐ **OUI, je m'abonne** (écrire en lettres capitales)

Envoyer par la poste à : Solutions Logiciels, service Diffusion, GLIE - 17 chemin des Boulangers 78926 Yvelines cedex 9 - ou par fax : 01 55 56 70 20
1 an : 50€ au lieu de 60€, prix au numéro (Tarif France métropolitaine) - Autres destinations : CEE et Suisse : 60€ - Algérie, Maroc, Tunisie : 65€ , Canada : 80€ - Dom : 75€ Tom : 100€
10 numéros par an.

☐ M. ☐ Mme ☐ Mlle Société

Titre : Fonction : ☐ Directeur informatique ☐ Responsable informatique ☐ Chef de projet ☐ Admin ☐ Autre

NOM Prénom

N° rue

Complément

Code postal : Ville

Adresse mail

☐ Je joins mon règlement par chèque à l'ordre de SOLUTIONS LOGICIELS ☐ Je souhaite régler à réception de facture

Gérer la qualité des données avec Data Quality Services

SQL Server est la plateforme de gestion et d'analyse de l'information de Microsoft. La version 2012 intègre plusieurs grandes nouveautés, notamment Data Quality Services (la version 2014 va introduire d'importantes nouveautés, NDLR). Une solution permettant de valider et d'améliorer la qualité des données, reposant sur la mise en place de règles et sur l'alimentation d'une base de connaissances.

Qu'est-ce que Data Quality Services

Data Quality Services (DQS) est une solution qui améliore la qualité des données, notamment à travers la mise en place de règles. Lorsque l'on parle de règles, le premier exemple venant généralement à l'esprit est celui du format des adresses e-mail ou des numéros de téléphone que l'on vérifie à l'aide d'une expression régulière. Mais c'est bien plus qu'un simple vérificateur de format de chaîne. Data Quality Services est une solution reposant sur des connaissances. Il est possible de créer des processus de qualité des données qui améliorent continuellement les connaissances, et, par conséquent, la qualité des données. De plus, c'est une solution communautaire pouvant être utilisée par une multitude d'intervenants, qu'ils soient responsables des données, de l'IT, ou des équipes métiers. DQS a pour objectif de nettoyer une source de données manuellement depuis son outil client, ou de manière automatique dans un flux de données de SQL Server Integration Services (SSIS) [Fig.1](#). DQS peut également interagir avec Master Data Services (MDS) à travers un add-in pour Microsoft Excel.

Data Quality Client

Cet outil a pour but de créer et alimenter les bases de connaissances qui seront ensuite utilisées pour analyser les données. L'interface d'accueil, relativement simple et intuitive, est composée de trois parties : la gestion des bases de connaissances, la gestion des projets de qualité des données et les options d'administration [Fig.2](#).

Création d'une base de connaissances

La création d'une base de connaissances, à proprement dite, se limite à renseigner un nom de base et une description. Son alimentation quant à elle s'effectue au travers de trois activités [Fig.3](#) :

- La « **gestion de l'arborescence du domaine** » définit les domaines de qualité et y associe des règles. Ce dernier correspond généralement à un type métier. Par exemple, une adresse mail et une adresse IP sont deux entités différentes. Un domaine contient un ensemble de règles qui permettent la validation ou non des données.
- La « **découverte des connaissances** » permet d'explorer un jeu de données (un échantillon Excel par exemple) pour alimenter la base de connaissances avec des valeurs pouvant être corrigées. Une phase d'analyse permet également de vérifier que la base de connaissances de qualité fait bien ce pour quoi elle a été conçue.
- La « **Stratégie de correspondance** » a pour objectif d'identifier des doublons sémantiques sur la base de règles de correspondance.

Gestion de l'arborescence du domaine

La première étape pour alimenter une base de connaissances est de créer les domaines de qualité de données, ainsi que les règles qui leur sont associées. Pour chaque domaine, il existe cinq onglets de paramétrage.

- L'onglet « **Propriétés du domaine** » définit les informations de base du domaine (nom, description, type de données). Il est également possible de normaliser automatiquement les

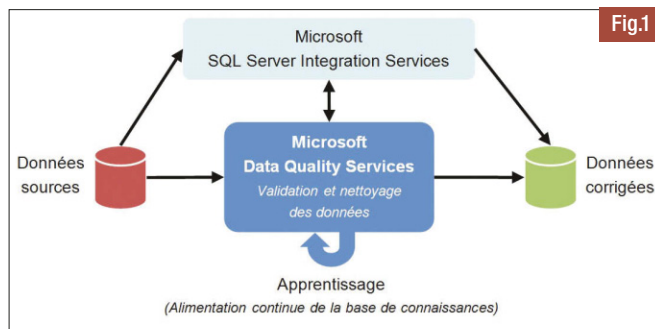


Fig.1

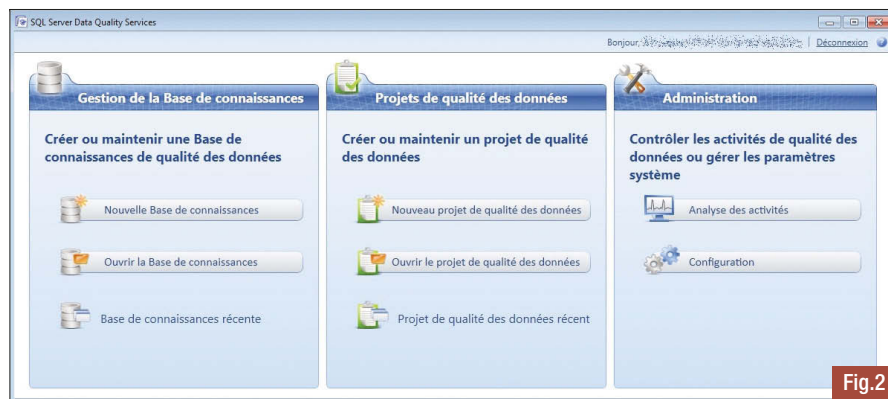


Fig.2



Fig.3

données et d'activer un vérificateur d'orthographe. Mais pour ce dernier, il ne s'agit que d'une fonctionnalité autonome de l'outil client n'existant pas côté serveur.

- L'onglet « **Données de référence** » permet de se connecter à un DataMarket et d'utiliser les services d'un fournisseur de données de référence pour vérifier la qualité des données du domaine.
- L'onglet « **Valeurs du domaine** » liste les valeurs connues de la base de connaissances. En d'autres termes, il est possible de générer un dictionnaire de valeurs valides et d'associer des corrections à des valeurs non valides. Ce dictionnaire peut être alimenté de manière automatique à travers l'activité de découverte des connaissances qui sera détaillée par la suite.
- L'onglet « **Relations à base de termes** » est utilisé pour gérer des relations entre des termes connus. Cette option peut être utilisée pour corriger les raccourcis couramment utilisés lors de la saisie des données. Par exemple, il est possible de créer une relation qui modifie le terme « Inc. » en « Incorporated ». Une

valeur « Société, Inc. » sera alors modifiée en « Société Incorporated ».

► L'onglet « Règles du domaine » permet de créer les règles de validation des données Fig.4. Dans le cas d'une adresse email par exemple, on peut faire correspondre la donnée à une expression régulière pour la restreindre à un format standard. Une option permet de tester la règle sur un jeu de tests créé à la volée Fig.5, ce qui simplifie les cas complexes nécessitant plusieurs essais, notamment pour les expressions régulières.

Une fois les règles mises en place, la base de connaissances est automatiquement publiée sur le serveur et devient accessible aux applications clientes SSIS et MDS.

Découverte des connaissances

L'activité de découverte des connaissances consiste à analyser un échantillon de données pour tester le bon fonctionnement des règles mises en place et à alimenter la base de connaissances. Pour cela, il faut d'abord définir la source de l'échantillon de données (une base de données SQL Server ou un fichier Excel) et associer les colonnes de données à des domaines de qualité. L'analyse de cet échantillon recense les nouvelles valeurs et les cas

non valides de chaque domaine Fig.6. A ce niveau, des corrections peuvent être proposées. Par exemple, un libellé de pays « FR » peut être corrigé en « France » Fig.7. La correction de ces données met à jour la base de connaissances.

Stratégie de correspondance

Il est possible de mettre en place une stratégie de correspondance pour rechercher les doublons sémantiques dans une source de données, ce qui peut s'avérer particulièrement utile lors d'une correction ou d'une fusion de référentiels. Son fonctionnement est basé sur des règles de correspondance sur lesquelles sont associés un ou plusieurs domaines. DQS évalue le degré de correspondance d'un enregistrement avec un autre à l'aide d'un poids associé à chaque domaine, ainsi qu'un critère de similarité qui définit, si on le souhaite, une correspondance exacte ou juste semblable Fig.8. Une fois l'analyse terminée, toutes les correspondances ayant un score supérieur ou égal à 80% sont affichées Fig.9. Les scores inférieurs à 80% ne sont pas pris en compte par DQS, ce qui pourrait dans certains cas être une limitation gênante. Mais les développeurs de Microsoft ont mis à disposition une exploration de chaque correspondance trouvée Fig.10. Ce qui permet d'avoir une meilleure visibilité de la règle mise en place et de pouvoir l'adapter selon le besoin. Par exemple, pour rechercher des personnes ayant une similarité sur leur nom et prénom. On pourrait associer les deux domaines de qualité correspondants avec un poids de 50% chacun. La similarité serait importante pour deux personnes ayant le même nom, mais avec un prénom qui diffère d'un seul caractère. Un petit exemple du mode de calcul : une similarité à 100% sur le nom et à 63% sur le prénom donnerait un score total de 81,5% ($1,00 * 0,5 + 0,63 * 0,5$) Fig.10. Pour approfondir un peu la recherche, il existe un critère « Condition préalable » qui restreint la

recherche de correspondance aux enregistrements ayant une correspondance totale. De plus, DQS permet de normaliser les chaînes de données pour un domaine, ce qui peut augmenter la précision de la correspondance.

Gestion d'un projet de qualité

Data Quality Services gère également des projets de qualité des données. L'objectif d'un projet est de nettoyer une source de données réelles à l'aide d'une base de connaissances. Il se découpe en quatre étapes :

- « Mapper », pour créer un lien entre les colonnes de la source et les domaines de qualité ;
- « Nettoyer », pour lancer le nettoyage des données à partir de la base de connaissances choisie ;
- « Afficher et gérer les résultats », pour explorer les valeurs nettoyées, et, approuver, refuser ou ajouter des corrections ;
- « Exporter », pour enregistrer le résultat du nettoyage dans une table SQL ou dans un fichier (CSV, Excel).

L'atout majeur est de pouvoir approuver ou refuser la correction pour chaque valeur rencontrée, et pour chaque enregistrement contenant cette valeur. Ce qui s'avère être très utile lorsqu'un utilisateur hésite sur une correction.

Pour conclure

La solution Data Quality Services s'avère être simple d'utilisation et propose des fonctionnalités claires qui pourront être utiles aussi bien sur les architectures importantes que sur les petites bases de données. De plus, son intégration dans SSIS est cohérente. Le mode de fonctionnement du composant de nettoyage DQS permet une réelle liberté dans les actions pouvant être réalisées en aval, même si la manipulation des codes états n'est pas très intuitive à la première utilisation. La solution DQS est proposée dans toutes les versions de SQL Server 2012, même la version la moins onéreuse, ce qui est toujours appréciable pour les PME.

Adrien Guet
Consultant, Sogeti BU EST
adrien.guet@sogeti.com - <http://www.azone.fr/Blog/>

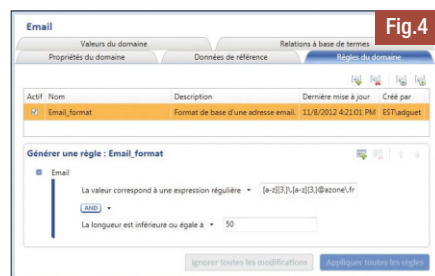


Fig.4

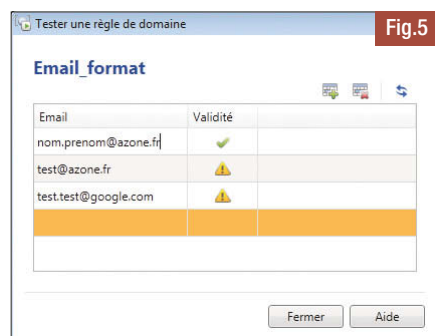


Fig.5

Champ	Domaine	Nouveau	Unique	Valide dans le domaine	Exhaustivité
Email	Email	27 (100 %)	27 (100 %)	26 (96 %)	<div></div>
IPAddress	IPAddress	27 (100 %)	27 (100 %)	26 (96 %)	<div></div>
Country	Country	26 (96 %)	6 (22 %)	24 (89 %)	<div></div>

Fig.6

Valeur	Fréquence	Type	Corriger vers
Canada	4	✓	
France	14	✓	
FR	3	⚠	France
France	1	⚠	France
Pays-bas	4	✓	

Fig.7

Domaine	Similarité	Poids	Condition préalable
Name	Similaire	50 %	<input type="checkbox"/>
Firstname	Similaire	50 %	<input type="checkbox"/>

Fig.8

ID d'enregistrement	Cluster	Score	Name	FirstName	Email	IPAddress	Country
1000000	1000000		GUET	Adrien	adrien.guet@azone.fr	192.168.0.1	France
1000001	1000000	81%	GUET	Adrian	adrien.guet@azone.fr	192.168.0.1	France
1000002	1000000	81%	GUETS	Adrien	adrien.guet@azone.fr	192.168.0.1	France

Fig.9

Exploration de correspondance d'enregistrement					Score
Contributions aux scores des champs					
Champ	Poids	Termes d'enregistrement à correspondance	Termes d'enregistrements pivots	Contribution au score	
Name	0.5	GUET	GUET	100%	81%
Firstname	0.5	Adrian	Adrien	63%	

Fig.10

Maxime Tricoire et Stéphane Jacquemain : la passion du mobile

Maxime et Stéphane se connaissent depuis l'école. Après des études en électronique puis en informatique, ils rejoignent des équipes Java. Ils se positionnent sur la technologie Android. En parallèle, ils créent quelques apps qui rencontrent un beau succès. Cela les incite à lancer leur propre société d'édition, nommée Progimax, et à s'y consacrer à plein temps. Deux ans plus tard, c'est une affaire qui roule. Voici nos deux développeurs du mois !

Comment êtes-vous tombés dans l'informatique et plus spécialement dans le développement ?

Déjà jeunes, nous étions deux passionnés d'informatique, d'électronique et de robotique. Comme il n'existait pas de Bac d'info, nous nous sommes inscrits en section STI électronique. Nous écrivions des scripts, des petits programmes pour PIC (microcontrôleur) ou calculateurs. À l'époque, c'est ce qui se rapprochait le plus de notre passion et nous permettait d'accéder à un BTS d'informatique industrielle.

Pour vous, qu'est-ce qui fait que l'on aime toujours et encore le développement, la technique ?

Nous adorons notre métier. Programmer, c'est créer. Chaque projet est unique. Chaque réalisation apporte son lot de contraintes et de problèmes. Il faut se réinventer en permanence, découvrir et apprendre de nouvelles techniques. Tout cela renforce notre passion. Nous avons parfois l'occasion d'intervenir sur des projets ultras motivants. Par exemple, nous avons développé une app d'aide aux entraînements sportifs à l'aide d'échantillons sonores. De manière générale, plus le sujet est complexe, et plus le défi nous intéresse. Nous cultivons la difficulté dans notre travail. Même si nos apps semblent simples en apparence, les techniques utilisées sont complexes. Ainsi, notre briquet virtuel se résume à une flamme vacillante qui s'éteint lorsque l'on souffle sur le micro. Cela met en œuvre des algorithmes et des mathématiques avancés, et nécessite souvent des heures et des heures de recherche. Il ne s'agit pourtant « que » de détecter le souffle de l'utilisateur dans le micro ou d'animer



Maxime Tricoire et Stéphane Jacquemain, de nombreux terminaux Android sur la table et quelques piranhas dans l'aquarium au fond.

une flamme à l'aide d'un moteur physique... Pour ce projet, nous avons découvert OpenGL. Dans la foulée, nous avons développé Falldown, un petit jeu 2D ayant rencontré un joli succès. Cela nous a permis de réaliser que l'on n'entre pas si facilement dans le Top 10 du Play Store.

Vous avez gardé un regard très geek : gadget, veille techno, c'est important pour votre job et votre passion ?

Nous avons acheté des robots (très chers) et ils ne sont toujours pas montés. Heureusement pour nos piranhas, le sous-marin de l'aquarium était vendu monté. Nous avons une véritable passion pour tout ce qui tourne autour de l'informatique et de la robotique, de la dernière montre de Samsung en passant par les Google Glass. Nous consacrons une partie de la journée à nous documenter, à faire de la veille technique, à suivre des présentations sur YouTube, en podcast ou dans des conférences.

Nous sommes d'ailleurs prêts à parcourir de nombreux kilomètres pour assister à une présen-



App Bougie sur Google Play

tation qui nous intéresse. Nous en profitons généralement pour discuter avec les autres passionnés et échanger sur nos expériences respectives. La conférence où nous allons le plus est le Paris JUG. Nous sommes pourtant à deux heures en voiture, mais nous adorons l'ambiance.

Être développeur n'est pas toujours facile : pression, évolution constante, frustration des projets et des « chefs », c'est quoi pour vous être développeur aujourd'hui ? Le job a-t-il changé depuis vos débuts ?



Être indépendants, avec nos apps, nous permet de limiter la pression des clients. Nous ne sommes pas frustrés par nos projets, puisque nous les choisissons. Nous aimons nous lancer des défis. Par exemple, crois-tu qu'une poignée de moto virtuelle, ça marcherait ? Non ? On creuse. On se demande s'il y a du code intéressant à faire. On essaie de programmer un son d'accélération. Et là, on réalise que ce n'est pas si évident. Mais on persévère, et on a raison, puisque l'application totalise déjà plus de trois millions de téléchargements. Notre truc, c'est d'inventer des apps étranges avec de vraies difficultés techniques. Généralement, ça touche à la physique du téléphone : micro, capteur de mouvements, caméra, etc.

Comme nouveau défi, nous prévoyons un logiciel de reconnaissance du « tic-tac » des montres pour aider les horlogers à les régler. Faire décoller un projet tout en se faisant plaisir n'est pas évident. Android a bouleversé nos habitudes. Avant, les clients nous demandaient de suivre un cahier des charges. Désormais, nous décidons nous-mêmes des spécifications et, bizarrement, nous tenons mieux les délais. Nous sommes beaucoup plus décontractés depuis que nous sommes nos propres patrons.

Et en dehors du boulot, qu'est-ce que vous aimez faire ? Comment trouvez-vous l'équilibre entre travail, vie privée, passion, famille ?

Ce n'est pas facile de concilier passion et vie de famille. Nous essayons de décrocher le week-end. L'avantage (mais aussi le piège) de l'informatique, c'est que l'on peut s'installer devant la télé à côté de sa femme, tout en codant discrètement sur son portable.

Nous nous rejoignons souvent pour faire du sport : course à pied, vélo, squash, escrime, muscu, etc. Cela permet de se vider l'esprit. Stéphane joue de la batterie électronique. Maxime s'adonne à l'aquariophilie et à la moto. Et nous aimons tous les deux la photo. Cela dit, quand on fait quelque chose en dehors du boulot, ça touche généralement à l'informatique.

Pouvez-vous nous présenter votre quotidien en quelques mots ?

Nous arrivons au bureau vers 8 h. Nous avons un coin détente (fauteuils et cafetière) où nous lisons les news, puis nous rejoignons notre poste de travail qui est dans la même pièce, mais séparé par un aquarium de mille litres.

Nous utilisons des portables sous Ubuntu, avec des grands écrans additionnels. Nous développons sous Netbeans. Nous nous efforçons de travailler avec les mêmes environnements/logiciels, ce qui permet d'être plus productifs. Nos projets sont stockés sur des serveurs Git hébergés chez OVH. Et bien sûr, nous avons quelques utilitaires sur App Engine. En fin de matinée, nous rejoignons notre cuisine pour y manger rapidement. De nouveau, nous nous accordons un café, le temps de lire quelques news. Et puis, nous avons un petit rituel ; nous nourrissons nos piranhas. Enfin, nous programmons jusqu'à 19 h. Nous nous réservons aussi des moments pour discuter de nos prochains projets. Nous notons tout sur un grand tableau.

Comment voyez-vous votre job évoluer ?

Aujourd'hui, nous travaillons sur Android. Demain, nous travaillerons peut-être dans un milieu complètement différent. Nous n'avons pas de vision à long terme. Nous voudrions développer pour iPhone, mais se former sur la



App Moto
sur Google Play

techno est trop long. Nous préférons nous concentrer sur Android, car nous maîtrisons tous les aspects (Gestion de projet, tests, environnements). En outre, nous avons développé des bibliothèques Java qui seraient très longues à porter. Notre but n'est pas de grossir uniquement avec le business des apps sur mobile. En effet, pour gagner des revenus convenables, il faut être très bien classé. Or cette place est dure à garder. Même si notre chiffre d'affaires, réalisé avec Android, nous satisfait, nous préférons nous lancer sur l'édition d'un produit qui pourra nous faire grossir. En gros, Android est notre propre capital-risqueur. Une de nos apps explosera peut-être en termes de revenus ; nous reverrons alors notre position.

Des conseils aux étudiants et développeurs qui nous lisent ?

Lancer sa société c'est facile. Le plus dur, c'est de faire rentrer de l'argent et d'en vivre. La plupart des développeurs, qui se lancent, développent pour des clients. Ça permet de bien gagner sa vie. Petit à petit, l'envie nous vient de travailler pour soi-même : un framework, un logiciel, un site révolutionnaire, ou même une app iPhone. Dans la vie d'un développeur, il n'y a pas plus grande satisfaction que de sortir un produit de son cru. Nous sommes fiers de nos 83 millions de téléchargements. Une grande satisfaction, c'est aussi d'être libres ; libres de choisir son client, libres sur nos horaires, libres pour nos choix...

Faire tourner une société, c'est aussi prendre des décisions. Nous avons compris l'importance des applications mobiles et avons pris le risque de moins développer pour des clients. Cela a été plus que positif pour nous. Progimax a été créée dans le sous-sol d'une maison individuelle, avec assez peu de moyens. L'informatique est l'un des seuls secteurs offrant ce type d'opportunités. Il faut savoir les saisir. Et surtout, suivre ses passions et envies, se remettre en questions constamment.



Progimax :
<http://www.progimax.com>

Thierry
Leriche-Dessier
Twitter : @thierryleriche

ERRATA DU MOIS DERNIER

Développeur du mois du n°168, une erreur dans le nom s'est glissée : **Graziano** et non Graziono

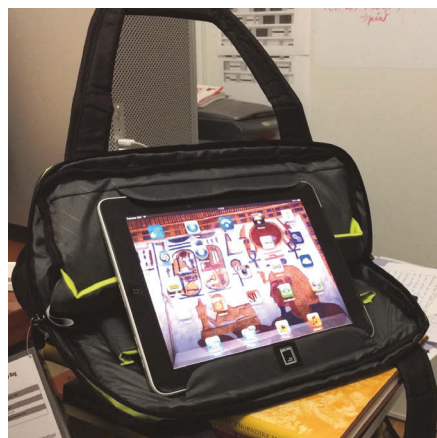
Kensington équipe vos tablettes et smartphones !

L'accessoiriste Kensington propose de nombreuses nouveautés pour l'automne et l'hiver 2013-2014. Voici un petit échantillon :

- ▶ le kit EVAP Rescue Pouch : ce kit permet de sécher votre mobile quand celui-ci tombe dans l'eau ou dans la neige. Il ne garantit pas une réussite à 100 % surtout quand l'appareil reste trop longtemps humide ou dans l'eau. Prix conseillé : 19,99 €



- ▶ Virtuoso Touch : grand classique, le stylo – stylet pour écran tactile. Simple et efficace. Si le toucher ne plaît pas, utilisez un stylet. Prix conseillé : 19,99 €
- ▶ Triple Trek : voilà un petit sac de transport très pratique, cadeau idéal pour les geekettes. Léger et résistant, il possède des poches dédiées aux ultrabook, tablettes et téléphone. Le + : les supports internes pour utiliser sa tablette. Prix conseillé : 44,90 €



- ▶ PowerBolt : gamme de chargeurs pour terminaux iOS. Plusieurs modèles « chargeurs voiture » sont disponibles dont une version avec deux connecteurs USB. Avec connecteur Lightning.
- ▶ Comercio : gamme d'étuis souples avec support, pour tablettes. Ces étuis proposent une bonne résistance. On aime beaucoup les petits emplacements pour les cartes de visite et les documents et le rajout magnétique.

Pour en savoir plus :

<http://www.kensington.com/kensington/fr/fr/home.aspx>

SANDISK CONNECT WIRELESS FLASH DRIVE : UN STOCKAGE PARTAGÉ

SanDisk Connect Wireless Flash Drive est un petit objet rectangulaire, facile à transporter, permettant de partager du contenu entre 8 appareils. L'objet propose un stockage de 16 ou 32 Go sur lequel vous mettez ce que vous voulez. Il est compatible avec les tablettes et smartphones. Si votre téléphone n'a plus de place, vous pouvez utiliser le SanDisk Connect.
Prix : 49,99 €/59,99 € selon la version.

Smartbot : pour les fans de robotique

Overdrive Robotics propose désormais SmartBot. SmartBot est un robot programmable qui utilise un Smartphone comme cerveau et capteurs. Les utilisateurs peuvent développer leurs propres programmes ou profiter d'applications prêtes à l'emploi.

Les utilisations peuvent aller de la programmation au jeu, en passant par le contrôle à distance avec retour vidéo ou la prise automatique de photos grâce aux technologies de détection de visage.

Le design du robot a été modifié et de nouvelles fonctions ont été ajoutées. Une gamme complète d'outils est d'ores et déjà disponible en accès libre pour permettre à tous de se plonger dans la programmation avancée avec

SmartBot. Le débutant, comme le programmeur confirmé, y trouvera tout ce dont il a besoin, comme un kit de développement gratuit, une documentation complète en ligne, ou encore des exemples de code. Le SDK est compatible iOS, Android et Windows Phone, en utilisant à chaque fois le langage natif de chaque plate-forme. Attention : le SDK est accessible si vous possédez un SmartBot (n° de série demandé). Un HDK (kit de développement matériel) est disponible pour utiliser les interfaces, créer des accessoires.

SmartBot permet aussi d'utiliser des plates-formes matérielles : Arduino et. NetMF (micro-framework .net). Vous pouvez directement



PNY dévoile la GeForce GTX 780 Ti

Le constructeur PNY a dévoilé sa carte GeForce GTX 780 Ti. PNY annonce une nette amélioration des cœurs CUDA (2 880 !). Elle embarque 3 Go de mémoire. La connectique est assez complète : HDMI, DVI, Display Port. Elle s'installe sur un slot PCI Express 3.0. Elle embarque la technologie GPU Boost 2.0. Cette nouvelle technologie permet aux joueurs de booster au maximum les performances de leur PC avec des options de contrôle précis. GPU Boost 2.0 surveille la charge du GPU avec des options de contrôle avancées comme la température maximale, l'overclocking et la surtension afin de garantir des performances ultimes. Cette bête de course est vendue environ 649 € !



utiliser une carte Arduino (ou. NetMF) pour étendre et piloter le SmartBot.

Prix : 180 € TTC. Plusieurs accessoires sont disponibles dont un kit d'extension (connecteur) à 2,5 €.. Tout est disponible sur la boutique en ligne.

Site : <http://www.overdriverobotics.com/SmartBot/>

AR Drone 2.0 : plus puissant, plus maniable

Parrot est un des leaders technologiques français. Il a récemment annoncé Parrot Flower, pour que l'on puisse surveiller la bonne santé de ses plantes grâce à plusieurs capteurs mesurant différents paramètres (humidité, lumière, température, taux d'engrais). ARDrone reste sans doute le « gadget » le plus connu du constructeur.



C'est quoi un drone ?

Le drone est un objet volant sans pilote, que l'on pilote à distance. Il existe des drones militaires et civils. Le principe de AR Drone est d'être un appareil équipé de 4 moteurs, capable de se mouvoir dans toutes les directions, et même de faire un looping. Il peut effectuer un vol stationnaire comme un hélicoptère.

Cela signifie que l'engin analyse en temps réel les vents et sa situation pour compenser de lui-même les poussées.

ARDrone est aussi très léger, facile à monter et démonter. La fragilité de certains éléments a été une des critiques de la première version; l'avantage est qu'il est très facile de changer des pièces. Au pilote d'avoir le doigté pour le mouvoir en douceur et éviter les actions trop brusques.

AR.Drone 2.0 possède une caméra HD frontale. Il peut streamer les images en vol. La résolution

a été améliorée par rapport à la première version. L'appareil dispose d'un port USB maître.

Un SDK pour s'amuser un peu ou être très sérieux

ARDrone est une véritable plate-forme de développement. Le SDK permet d'accéder aux fonctions du drone, au système, au firmware, aux éléments matériels (audio et vidéo, moteurs). Basiquement, le SDK comprend une librairie (ARDroneLib), la suite d'outils (ARDrone Tool) et une librairie de contrôle. Le développeur doit tout d'abord comprendre le fonctionnement des rotors et comment les coder et les utiliser dans les applications. Car selon la rotation des hélices, le drone fera telle ou telle action. ARDrone embarque un microcontrôleur qui contrôle l'engin. Pour communiquer, il embarque un système wifi (réseau utilisé par les

applications compatibles). L'appareil possède une adresse IP. Mais un des soucis que les ingénieurs ont dû résoudre a été le temps de latence entre l'application de contrôle et le drone. Car une latence trop longue risque de rendre l'engin incontrôlable.

Le SDK n'autorise pas l'accès aux couches les plus basses du drone, ni aux capteurs en eux-mêmes. Le SDK permet de coder pour iOS, Android, un PC Linux et Windows Phone 8 (non officiel).

Au-delà des SDK officiels, il est possible d'utiliser et de connecter son drone à divers périphériques de contrôles comme Kinect et Leap Motion. Certains développeurs ont même utilisé JavaScript pour créer un contrôleur AR.Drone ! En fouillant un peu sur le net, vous trouverez de nombreuses librairies non officielles. A utiliser en connaissance de cause !

Où acheter ?

Parrot propose deux versions : AR.Drone 2.0 et AR.Drone 2.0 Power Edition. Cette dernière inclut une batterie plus puissante capable de tenir 36 minutes contre 12 en version standard. Les pièces de rechanges et accessoires sont disponibles (moteurs, hélices, boîte à outils, etc.). Prix : à partir de 299 €.

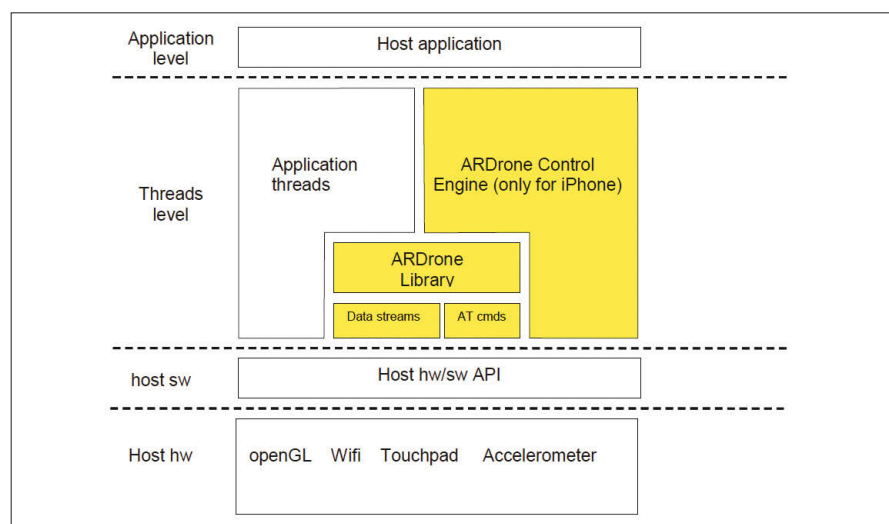
Quelques liens utiles

<https://projects.ardrone.org>

<https://github.com/ARDrone2Windows>

<http://dronehacks.com/category/ar-drone-sdk/>

<http://ardrone2.parrot.com>



Décryptage de l'avenir du chiffrement Web

En 2012, j'ai analysé quelque 90 millions de sites Web (tous les noms de domaine .com, .net et .org qui existaient alors) pour vérifier s'ils prenaient en charge les communications chiffrées. J'ai pu ainsi constater que seulement environ 0,5% de ces sites avaient les moyens de protéger le transfert de leurs données. La grande majorité d'entre eux ne chiffrait rien du tout. Et pour les sites Web qui ont tenté d'adopter le chiffrement, ce fut un échec pour beaucoup d'entre eux. Évidemment, pour les sites Web populaires, la situation est un peu meilleure, mais seulement environ 10% de ces sites prennent en charge le chiffrement.

Globalement, les sites Web se répartissent en trois catégories. Dans la première figurent les sites soucieux de la sécurité et qui ont des compétences et des moyens pour l'appliquer. La seconde catégorie regroupe des sites éventuellement conscients de certains problèmes de sécurité, mais qui n'y consacrent qu'un temps limité. Enfin, dans la troisième se trouvent les sites Web qui ne se soucient pas de la sécurité et qui ne sont pas conscients des problèmes de sécurité; en conséquence, ils n'utilisent aucun chiffrement.

Notre technologie continuant d'évoluer rapidement, la communauté de la sécurité investit traditionnellement l'essentiel de ses efforts dans la première catégorie. En effet, les professionnels de la sécurité tentent de s'entraider. Près de 20 ans après la première version publique de SSL, le protocole qui sécurise l'essentiel de l'Internet, cette technologie est finalement presque au point. Deux ressources critiques qui manquaient particulièrement sont en cours de déploiement pour traiter les derniers points faibles de SSL.

HTTPS et HSTS

L'un des gros problèmes du chiffrement sur le Web tel qu'il est actuellement déployé est qu'il n'est pas obligatoire, si bien que les déploiements sont peu efficaces. Par exemple, si vous tapez un nom de domaine dans la barre de navigation, votre navigateur Web tentera d'accéder au site Web sans chiffrement. Cela suffit pour qu'un pirate actif de type « man-in-the middle », c'est-à-dire une quelconque personne utilisant le même réseau WiFi que vous, puisse détourner toutes vos communications, même si le site Web utilise le protocole HTTPS chiffré. Dans la pratique, ce problème majeur peut être évité si personne n'oublie de taper « HTTPS » chaque fois que nécessaire ou bien de mettre en signet ses sites sécurisés. Mais, bien entendu, la plupart des internautes ne le font pas.

Heureusement, une nouvelle norme appelée *HTTP Strict Transport Security* (HSTS) publiée en 2012 résout ce problème en permettant aux sites Web de « forcer » le chiffrement de toutes les communications échangées avec eux. Actuellement, le support de HSTS est assez limité et seuls les navigateurs Chrome, Firefox et Opera prennent cette norme en charge. Nous attendons toujours qu'Internet Explorer et Safari fassent de même. Tout ceci n'est pas idéal, mais c'est mieux que rien.

Le problème des Autorités de Certification pirates

Le deuxième gros problème lié au Web sécurisé est qu'il repose sur des tiers de confiance, à savoir les *Autorités de Certification* (AC) chargées de garantir aux utilisateurs qu'ils se trouvent bien sur le bon site Web.

Et même si ces Autorités vous demandent de démontrer la propriété du nom de domaine avant de vous délivrer un certificat, elles n'ont actuellement pas besoin de votre consentement : en effet, une quelconque CA peut émettre un certificat pour n'importe quel site Web dans le monde. Ceci est fondamentalement mauvais. Dans la mesure où il existe certainement plus de 100 CA dans le monde entier, non seulement la sécurité de l'ensemble du système est extrêmement faible, mais il n'existe aucune protection contre les Autorités de Certification pirates.

Ce problème peut être résolu à l'aide du « *public key pinning* », une technique d'« épingle » des certificats qui permet aux sites Web

de reprendre le contrôle sur les certificats pouvant être utilisés pour eux. C'est à l'aide de cette stratégie que Google, qui gère son propre navigateur Web, est parvenu à se protéger. C'est en effet via son pinning intégré que nous avons appris la compromission de l'Autorité de Certification DigiNotar et, plus tard, l'émission par Turktrust d'un certificat provenant d'une Autorité de Certification subordonnée pirate. Google s'est montré généreux en autorisant d'autres sites Web de grande valeur à utiliser cette même technique, mais cette approche manuelle et centralisée n'est pas appropriée. C'est pourquoi deux nouvelles normes concurrentes sont en cours de développement.

L'une, baptisée *HTTP Public Key Pinning* (HPKP), configure le pinning via les en-têtes de réponse HTTP et s'applique donc seulement aux sites Web. L'autre, appelée *TACK*, fonctionne sur la couche protocolaire TLS et s'utilise avec n'importe quel protocole basé sur TLS pour la sécurité.

Le support du « public key pinning » par le navigateur Web restera limité pendant encore un certain temps. Cependant, nous voyons finalement le bout du tunnel car il sera bientôt vraiment possible de déployer le protocole TLS pour traiter avec efficacité un large éventail de menaces.

En fin de compte, si la sécurité vous importe, vous pouvez entreprendre dès à présent deux actions majeures, à savoir : commencer par déployer HSTS, puis vous préparer au déploiement de la technique de « public key pinning ». Ce qu'il est possible de faire pour les deux autres catégories de sites Web, ceux qui n'ont pas le temps de s'intéresser à ou qui se moquent de la sécurité, fera l'objet d'un prochain article entièrement consacré au sujet.



Ivan Ristic

Directeur de recherche sur la sécurité des applications chez Qualys, Ivan Ristic étudie depuis plusieurs années déjà les technologies SSL/TLS et PKI et se concentre sur

l'évaluation à grande échelle et la formation des utilisateurs. Il élabore les tableaux de bord SSL Labs et SSL Pulse sur lesquels il publie ses recherches et outils. Il travaille actuellement sur un nouvel ouvrage intitulé Bulletproof SSL/TLS and PKI.

Appât du gain, désaccord et mécontentement, vols de données : gare aux Botnets !

Une interruption des services en ligne entraîne de graves conséquences sur le métier et le business d'une entreprise. C'est l'objectif des attaques par Déni de Services, et, a fortiori, des attaques par Déni de Services Distribués (DDOS) lorsque plusieurs machines sont impliquées.

Les machines à l'origine des attaques DDOS font le plus souvent partie de réseaux botnets. Ces derniers sont composés de machines infectées au préalable par un malware et sont ainsi contrôlables à distance depuis le serveur de commande et de contrôle (C&C). A titre d'exemple le botnet «Mariposa» démantelé en 2010 comptait plus de 15,5 millions de machines infectées.

Pour utiliser ces botnets, un pirate n'a pas besoin d'avoir infecté en amont des milliers ou des millions de machines. En effet, Internet a permis l'émergence «d'agences de location de botnets» qui peuvent ainsi être acquises pour une somme allant de 5 à 200 \$ par heure selon la taille et la durée de l'attaque souhaitée.

Même si l'objectif final est identique, à savoir la perturbation de la présence d'une entreprise ou d'un organisme sur Internet, les attaques DDOS sont motivées par des raisons diverses :

- L'appât du gain qui résulte de malveillances entre concurrents ou encore d'extorsions.



07-17-11 © -MG-

Ces dernières font généralement suite à une attaque préliminaire visant à intimider la cible.

- Les «hacktivistes» qui désirent manifester leur désaccord suite à une décision politique ou économique. Les cas les plus célèbres font état des actions des «Anonymous» auprès d'organismes tels que le FBI, l'Élysée, l'Express ou encore Paypal et Sony.

- Faire diversion et cacher le véritable objectif du pirate qui vise à extraire des informations sensibles à l'aide d'une attaque ciblée, dite APT (Advanced Persistent Threat). L'équipe informatique interne étant en alerte à cause de

la saturation des ressources (site Web, lien Internet...), la fuite de données peut être effectuée de manière furtive.

Concrètement, une attaque DDOS vise à rendre indisponible un ou plusieurs services, ou à perturber son fonctionnement nominal. Pour cela, elle utilisera une combinaison des attaques suivantes :

- Attaques volumétriques (ou flood) qui visent à saturer la cible de sorte qu'elle ne puisse plus traiter de flux légitimes ;

- Attaques appelées «low and slow» consistant à envoyer du trafic en apparence légitime, avec un faible débit afin d'être le plus invisible possible, mais dont l'impact est similaire aux attaques volumétriques ;

- Attaques applicatives basées sur l'exploitation de vulnérabilités de la cible.

La comparaison du chiffre d'affaires généré par le marché de la protection DDOS et celui du firewall (275 Millions \$ contre 7 Milliards \$ en 2012) nous confirme que la mise en place de protection DDOS ne s'est pas démocratisée de la même manière. Toutefois, celui-ci est en pleine expansion avec une augmentation de 30 % par rapport à 2011 et la dernière étude du cabinet d'étude Infonetics Research prévoit un chiffre d'affaires de 530 millions \$ d'ici 2017.

Depuis un changement de stratégie des éditeurs de protection DDOS en 2012 qui a consisté à cibler davantage les entreprises, la répartition des revenus montre un équilibre entre ce secteur et celui des opérateurs.

◉ Arnold Tissier,
Manager Intégration, Nomios

Les failles de sécurité de HTML5

Si HTML 5 est partout, il y a un élément que l'on évoque finalement assez rarement : la sécurité et les risques potentiels. Nous avons publié il y a quelques mois de cela, un dossier sur le sujet. Et contrairement à une idée reçue, HTML 5 n'est pas aussi vertueux que cela. La communauté Open Web Application Security Project propose plusieurs documents essentiels pour comprendre les failles de sécurité et leur fonctionnement. Parfois, le développeur web ne s'embête pas à respecter la moindre règle. Il va coder en dur ce qu'il ne devrait jamais faire ! Les deux sources à lire sans aucune modération :

HTML et la sécurité, un point d'étape :
<http://fr.slideshare.net/Eagle42/2013-0404html5securityv2>

HTML5 Security Cheat Sheet :
https://www.owasp.org/index.php/HTML5_Security_Cheat_Sheet

Le top des vulnérabilités en 2013 ?

Sur quel système ou technologie le plus grand nombre de vulnérabilité a été découvert ? CVE détaille les résultats. Le top 5 est le suivant :

JRE :	180
JDK :	180
Noyau Linux :	155
Chrome :	154
Firefox :	135

Le top 5 des éditeurs (en nombre de vulnérabilité) :

Oracle :	495
Cisco :	364
IBM :	313
Microsoft :	298
Apple :	179

Mais attention, vulnérabilité ne signifie pas forcément faille ou attaque.

Site : <http://www.cvedetails.com/index.php>

YESWEHACK

Voici un nouveau site web pour promouvoir les carrières et les emplois dans la sécurité informatique. Il n'existait pas jusqu'à pas présent un site réellement dédié à ce domaine. Et pour les entreprises, il est parfois difficile de trouver des profils. Yeswehack ambitionne d'être le site d'emplois de référence sur ce domaine.

Site web : <https://yeswehack.com/>

Développer pour les terminaux mobiles



Dans Programmez ! nous parlons régulièrement du développement pour les terminaux mobiles. Et il y a de bonnes raisons à cela : explosion du marché avec 2 leaders, Android et iOS, et une 3e plate-forme en puissance, Windows Phone 8/Windows 8 sur tablette. Sans oublier que plusieurs systèmes rêvent aussi de trouver une petite place, comme c'est le cas de Tizen et de Firefox OS.

Le sujet étant monstrueusement vaste, nous nous sommes limités, pour les 3 grandes plateformes, à des points précis : les nouveautés introduites par iOS 7 (en 2 parties), historique, prise en main et utilisation

du in-app dans Android, l'incroyable Imaging SDK de Nokia pour les Lumia sous Windows

Phone 8, ou encore la détection du terminal Windows Phone dans son code...

Nous aborderons aussi un sujet brûlant en ce moment : faut-il développer nativement ou au contraire créer des applications multiplateformes ? La question est loin d'être simple. La réponse dépendra de votre budget, des contraintes de temps et des fonctionnalités et services que vous souhaitez voir dans l'application mobile...

Bonne mobilité !

François Tonic

Les solutions mobiles multiplateformes aujourd'hui

La diversité des systèmes d'exploitation mobiles, le choix laissé au consommateur, poussent de plus en plus les entreprises et les développeurs indépendants à s'intéresser à des solutions leur permettant d'être présents sur toutes les plateformes existantes (ou alors au moins les principales). D'ailleurs, ces solutions font toujours état de vrais débats sur la conception des applications pour smartphone, à savoir s'il faut ou pas créer des applications natives, hybrides ou web.

Quel type d'applications ?

Les principaux types d'application connus sont natif et web. Si on devait en dresser un comparatif succinct, ce serait celui-ci : Fig.1.

Une question de choix (et aussi de coût)

Avant de se lancer dans la conception de son application mobile, il y a des questions auxquelles il est judicieux de répondre :

- ▶ Pour quelle plateforme cible ? iOS ? Android ? Windows (Phone) ?
- ▶ Si je choisis de cibler une seule plateforme, vais-je envisager de porter par la suite mon application sur les autres OS ?
- ▶ Quels sont mes moyens ? Ai-je du temps ?

Une question de compétences

Quand les premières décisions vous poussent au développement d'une application multiplateforme, d'autres questions viendront vous guider sur les outils à privilégier pour la création de votre produit, ou sur les solutions à mettre en place pour vous aider à y parvenir. Et ce sont vos compétences (ou celles que vous souhaitez acquérir), guidées par vos moyens, qui vont vous orienter.

En effet, que vous soyez plutôt développeur Web ou pas, il y aura encore là une question de choix. La tendance est que, en fonction de vos capacités techniques, vous choisissiez les solutions adéquates. Si vous êtes donc développeurs Web, vous aurez par exemple des dispositions pour des outils vous permettant de construire des applications mobiles avec HTML5/JavaScript.

Faut-il vraiment favoriser les solutions multiplateformes ?

Microsoft, Apple et Google font en sorte de pouvoir attirer le regard des consommateurs sur leurs produits et les mobinautes y sont sensibles. Si la tendance des parts de marché s'est inversée au fil des années au profit de Google face à Apple, Microsoft voit aussi ses chiffres de vente nettement progresser. Cibler donc le maximum de plateformes vous

Web mobile	Natif
<ul style="list-style-type: none"> • Standard cross-platform • 1 version du site • Coûts faibles • Référencement via moteur de recherche • Expérience limitée • Online 	<ul style="list-style-type: none"> • Spécifique à la plateforme • Performances • Référencement via Store • Coûteux • Maintenance • Respect de l'expérience utilisateur • Online ou offline

Fig.1

Type	Description	Exemples	Cible	Langage
Framework Javascript	<ul style="list-style-type: none"> • Site web responsive • Aide au développement avec composants réutilisables 	jQuery Mobile, Sencha Touch, Zepto JS, ImpactJS, LimeJS, iUI, Wink...	Développeurs web	Javascript
Web vers natif	<ul style="list-style-type: none"> • Wrapper d'app HTML/JS • Single-page app • Web app avec fonctionnalités natives 	Adobe PhoneGap Build, Uxebu, Sencha Touch v2, MoSync...	Développeurs web	HTML/CSS/Javascript
Cross-compilation	<ul style="list-style-type: none"> • 1 langage compilé dans le langage natif cible 	Xamarin, Titanium...	Développeurs logiciel	Langage supporté (C#, Javascript, Java...)
Runtime	<ul style="list-style-type: none"> • Machine virtuelle qui fait tourner du code intermédiaire 	Adobe Air	Développeurs logiciel	Langage supporté (ActionScript, JS...)
Bas niveau	<ul style="list-style-type: none"> • Composants graphiques bas niveau • Utilisé pour les jeux 	Unity, MonoGame, DeltaEngine, Corona ...	Développeurs jeux	Langage supporté (C++, OpenGL...)

Fig.2

Quelles solutions et quelles méthodes aujourd'hui ?

permettra de cibler le maximum d'utilisateurs sans forcément bloquer votre choix sur un OS spécifique Fig.2..

Quel outil choisir ?

Xamarin Fig3.

Xamarin est un outil de la société du même nom déjà bien connue des développeurs utilisant Mono; c'est son fondateur qui en est également à l'origine.

Cet outil vous permet de pouvoir utiliser vos compétences en C# pour construire des applications natives à destination des téléphones iOS, Android ou Windows Phone. Son principal avantage est bien entendu que si vous êtes développeur C#, son utilisation pourrait s'inscrire dans une logique.

Développement

Langages : C#

IDE : Vous pourrez utiliser soit la suite Xamarin Studio mais également un plugin qui s'intègre dans Visual Studio. (Des templates sont disponibles pour iPad, iPhone et Android).

Build : Le build se fait en local mais il est également possible de lancer des tests de l'application dans le cloud.

Extensibilité : il est possible de développer ou de récupérer des modules dans leur « store » de composants. Beaucoup existent pour iOS ou Android, mais peu pour Windows Phone. Il est également possible d'importer des bibliothèques JNI pour les intégrer à une application Android.

Intégration continue : Jenkins, TFS ou Team City. Et pour les tests, il est possible



Fig.3

ensuite de lancer des tests automatisés sur différentes plateformes en utilisant leur service <http://xamarin.com/test-cloud>

Question sur le portage : Il est possible de scanner du code .net pour savoir quel est son pourcentage de compatibilité avec les projets Xamarin.iOS et Xamarin.Android (via <http://scan.xamarin.com/>)

Support

Exemples de code : iOS, Android, windows 8, Wp7, des samples d'applications et de modules de Xamarin sont sur Git : <https://github.com/xamarin/>

Communauté : Active, par forum, 10 boards 25k messages

Support : Le support via mail nécessite un package plus cher, gratuit par forum

Fréquence des mises à jour : ils sont à jour et font tout pour correspondre aux dernières versions des OS

Outils de formation : Pas à pas détaillés et vidéos sont là pour nous aider

Interface graphique

C'est un peu ici la principale difficulté que vous risquez de rencontrer en utilisant cet outil. En effet, la gestion de l'interface est spécifique à chaque plateforme, et il faudra s'adapter à chaque OS.

Contrôles graphiques : Contrôles graphiques spécifiques à chaque plateforme

Editeur graphique : Visual Studio / Blend pour Windows Phone, Xamarin Studio/Visual Studio pour Android, XCode iOS

Intégration graphique : A adapter sur chaque OS

Animations : Dépend des plateformes

Intégration

API Native : oui mais malheureusement il y a un développement spécifique par OS, cela passe par des conditions de compilations

GPS : Oui pour Android et iOS

Stockage de données en local : Oui au travers des APIs

Caméra : oui au travers des APIs

Contacts : oui pour les deux

Calendrier : oui au travers des APIs

Media : oui au travers des APIs

Autres spécificités OS : tout dépend de l'API mise à disposition, mais oui !

In-app purchase : Oui pour iOS, il faut passer par une librairie MonoDroid pour le faire fonctionner sur Android

Publication

Packaging : Il y a des étapes intermédiaires

pour certaines plateformes.

Pour Android : Une fois le package obtenu en mode release, il faut le signer en utilisant un assistant et le publier à la main (Tout est clairement expliqué lors de ce processus). De même pour iOS, un grand tutorial présente les différentes étapes à effectuer jusqu'à la soumission de l'application

Bilan

Avantages

- Debug depuis VS
- Automatisation de tests
- Beaucoup d'exemples, suivis et mises à jour
- Sources des modules Xamarin disponibles
- Mis à jour en même temps que les OS
- Designer intégré pour chaque plateforme
- Communication avec l'OS à 2 niveaux au choix (bas ou haut)
- Documentation très complète
- Store de composants
- Intégration réseaux sociaux

Inconvénients

- Minimum de natif à gérer
- Design à adapter à chaque OS, non mutualisable
- InApp Purchase Android géré via un module externe (gratuit)
- Pas d'aide à la publication (mais aide complète)
- A minima, 1 Mac distant si on cible du iOS

PhoneGap & Sencha Touch 2

PhoneGap est géré par Adobe (avec la partie Cordova qui est compatible Open Source) et Sencha Touch, quant à lui, par Sencha. Pour utiliser ces outils, vous devrez disposer essentiellement de compétences HTML/Javascript /CSS. Si PhoneGap est un déploiement packagé (HTML/JS/CSS embarqué dans une WebView native), Sencha est plutôt un Framework Javascript pour faire du Web Mobile Fig.4.

Développement

Langages

PhoneGap : JS + HTML5 + CSS3 + Langage de la plateforme

Sencha : JS + HTML5 + CSS3

IDE

PhoneGap : Pas d'IDE dédié ou spécifique mais quand PhoneGap s'installe, il intègre des plugins pour chaque IDE couramment utilisé pour chaque plateforme (Eclipse pour Android, XCode pour iOS, Visual Studio pour Windows Phone...). Sencha : N'importe quel IDE/éditeur web

Build

PhoneGap : Build en local ou dans le cloud (possible pour toutes les plateformes sauf pour WP8).

Sencha : Minification et optimisations optionnelles (à ajouter soi-même via NodeJS)

Extensibilité

PhoneGap : Accepte des plugins pour

étendre les fonctionnalités natives mises à disposition du code JS, et pour bien s'interfacer avec les frameworks JS.

Sencha : Oui ajout de librairies JavaScripts

Support

Exemples de code

PhoneGap : Très nombreux par la communauté donc du bon et du moins bon, mais globalement on trouve rapidement son bonheur. PhoneGap bénéficie de sa renommée.

Sencha : Plutôt complet de base.

Documentation

PhoneGap : Documentation complète pour utiliser les interfaces existantes (natif <=> JS) ou pour créer ses propres plugins.

Sencha : Riche, surtout pour démarrer, sur le site de l'éditeur.

Communauté

PhoneGap : Communauté active et nombreuse (newsgroups, forums, blogs...)

Sencha : Un peu moins que PhoneGap

Support

PhoneGap : Support payant par PhoneGap via 5 types de profils : Basic, Starter, Pro, Corporate, Enterprise. Et support gratuit via les forums etc.

Sencha : via des forums

Fréquence des mises à jour

PhoneGap : Fréquemment mis à jour (une dizaine de mise à jour en 2013)

Sencha : Pour l'instant une mise à jour majeure par an donc un projet vivant.

Interface graphique

Contrôles graphiques

PhoneGap : Non. Possibilité de packager avec certains JS/CSS distincts pour chaque plateforme.

Sencha : JS/CSS

Animations :

PhoneGap : JS + CSS

Sencha : JS + CSS

Notons que les performances des animations dépendront beaucoup du hardware sur lequel elles seront exécutées.

Intégration

PhoneGap

API Native : Oui, complètement

Support du touch : Support du touch de l'HTML/JS



Fig.4

GPS : Oui via un pont JS <=> natif

Stockage de données en local : Oui via un pont JS <=> natif ou en utilisant le localStorage du browser

Caméra : Oui via un pont JS <=> natif

Contacts : Oui via un pont JS <=> natif

Calendrier : Oui via un pont JS <=> natif

Media : Oui via un pont JS <=> natif

Autres spécificités OS : Oui via un pont JS <=> natif

In-app purchase : Oui via un pont JS <=> natif
Sencha : Tout est géré en Javascript, donc pas encore d'intégration complète (GPS, Photo...) mais à venir avec les versions prochaines des navigateurs (standardisation).

Publication

Packaging

PhoneGap : Packaging automatique et dans le cloud, donc permet de s'abstraire des contraintes de chaque plateforme (avoir un Mac pour packager pour iOS, etc).
Sencha : Génération du package en ligne de commande.

Bilan

PhoneGap

Avantages

- PhoneGap très populaire = grande communauté
- Gratuit
- Build et publication dans le cloud (plus besoin d'un Mac / outils installés)
- Outils de dev compatibles Windows/Mac/ Linux
- Support très complet des OS mobiles
- Plugins existants pour les IDE courants
- Branchable sur du natif
- Applis web existantes récupérables et adaptables à "faible" coût

Inconvénients

- PhoneGap Build dans le cloud payant
- Installation pouvant être laborieuse / beaucoup de prérequis non documentés
- Génération d'un projet mode boîte noire
- Documentation simple pas souvent détaillée.
- Peu d'exemples officiels
- Pas de compatibilité ascendante (ou peu) = support multi-version compliqué
- Performances pauvres par instant, induites par le rendu HTML/JS
- Debug de Javascript
- Interface HTML à adapter à chaque OS
- Pas possible de réutiliser des applis natives existantes pour les porter vers d'autres plateformes

Sencha

Avantages

- Thèmes / OS
- Installation simple
- Bien intégré à PhoneGap
- Beaucoup de librairies de base

Inconvénients

- Ajoute de nouveaux prérequis par rapport à PhoneGap (Ruby...)

Titanium Mobile



Titanium Mobile édité par AppAccelerator permettra tout comme les outils précédents de créer des applications natives, hybrides ou HTML5 à partir d'un SDK Javascript.

Développement

Langages : Javascript

IDE : basé sur Eclipse.

Templates : Il en existe qui ouvrent un tableau de bord Web mais cela nécessite un compte sur leur site.

Build : les outils de conversions « JS vers du code » sont dans le cloud.

Extensibilité : Il est possible de créer des modules en langage natif (pour l'instant Objective-C ou Java) afin d'interagir avec les OS spécifiques. Pour Windows, il faut encore attendre un peu.

Support

Exemples de code : Il existe pas mal de samples dans Titanium Studio.

Documentation : quelques « rares » articles mais des vidéos sur Youtube

Communauté : Chat, email, ou phone mais il y a différents packages : Standard ou Enhance. Pour avoir les prix, il faut les contacter via le support.

Support : Via la section support services de leur site.

Fréquence des mises à jour : Les documentations ne sont pas toutes à jour. Mais ils se mettent assez rapidement à jour par rapport aux versions d'OS

Outils de formation : Il existe des sessions de formations pour devenir développeur titanium certifié

Interface graphique

Il est possible, soit de créer une vue au format XML, soit en générant toute l'interface en JS

Contrôles graphiques : En utilisant "Alloy" (framework MVC utilisé par Titanium), il y a possibilité de surcharger les thèmes de base

Editeur graphique : Il n'y a pas d'éditeur graphique, tout se fait en XML (pour la partie MVC) ou autrement en JavaScript

Animations : Pour iOS il existe des animations préconçues

Intégration

API Native : Il n'y en a pas vraiment de base. Certains modules peuvent être faits en langage natif pour des parties spécifiques. Cela minimise de ce fait la part de code

réutilisable pour une autre plateforme.

Support du touch : Support du touch de l'HTML/JS

GPS : Titanium a une API pour gérer la localisation selon les devices

Stockage de données en local : ils ont une base de données spécifique (Titanium Database). Possibilité d'utiliser SQLite ou du LocalStorage selon le type d'application qui est créé (HTML ou « Native »). Il existe des APIs pour gérer les fichiers du système.

Caméra : Oui depuis leurs APIs

Contacts : Oui depuis leurs APIs

Calendrier : Il y a des solutions spécifiques aux OS.

Media : Géré par « Titanium.Media »

In-app purchase : Il y a des modules mais qui fonctionnent selon les plateformes.

Publication

Internationalisation : Il y a un namespace pour gérer les messages. "Ti.Locale"

Packaging : Possibilité de publier l'application en utilisant directement Titanium Studio. Les applications peuvent être publiées sur les plateformes OpenShift, Heroku, Capistrano ou sur un FTP

Store d'entreprise : Avec OpenShift il est possible de publier sur un réseau privé.

Bilan

Avantages

- Titanium Studio s'installe facilement
 - Plusieurs exemples disponibles
 - Build dans le cloud, pas besoin d'un Mac
 - Modules natifs (mais peu conseillés)
 - Internationalisation simple
 - Gestion de store privé
 - Framework MVC Alloy, structurant
- ### Inconvénients
- Pas vraiment de support pour Windows Phone (à venir fin 2013 !)
 - Configuration du Studio et SDK à installer à la main
 - Exemples pas tous à jour
 - Besoin d'une connexion internet
 - Pas de designer de base (outils payants uniquement)

Conclusion

Les technologies évoluent très rapidement, et il est probable qu'à la lecture de cet article, le support du multiplateforme se soit amélioré, élargi, que ce soit avec les outils présentés ou avec de nouveaux disponibles sur le marché. Par ailleurs, bien qu'exhaustif, chaque outil présenté peut être vu encore en long, en large et en travers au cours d'un dossier qui lui soit spécifique. Quoi qu'il en soit, la balle est maintenant dans votre camp, il ne vous reste plus qu'à faire ... le bon choix.

 David Poulin, Expert .NET chez Soat



iOS 7 de A à Z !

1^{ère} partie

iOS 7 est sorti depuis quelques semaines. Trois experts iOS détaillent ce cocktail survitaminé à la pomme : SDK, librairies, jeux, API !

SDK iOS 7 : QUOI DE NEUF ?

Les nouveautés du framework CoreMotion

Une des nouveautés particulières de l'iPhone 5S est le coprocesseur M7. Celui-ci est destiné à libérer le processeur principal du téléphone de certaines tâches liées aux différents capteurs sensoriels de l'appareil : gyromètre, accéléromètre, etc. La plupart de ces informations se récupèrent via le framework CoreMotion.

Celui-ci a été mis à jour pour être compatible avec le coprocesseur M7. Ce processeur, pour rappel, fonctionne en tâche de fond.

Auparavant, les informations ne pouvaient être récupérées que si l'application était en activité dans le système.

Dorénavant, le coprocesseur M7 gère un journal d'activité et permet de sauvegarder un historique de l'activité de l'utilisateur.

Des méthodes nous permettent de récupérer ces informations au lancement de l'application ou à sa remise en activité.

Le framework contient 3 nouvelles classes :

- ▶ CMMotionActivityManager ;
- ▶ CMMotionActivity ;
- ▶ CMStepCounter.

Les deux premières classes sont liées. Elles permettent d'obtenir l'état de l'utilisateur en termes de déplacement : stationnaire, en train de marcher, en voiture, etc.

Pour l'utiliser, vous devez préalablement ajouter le framework CoreMotion à votre projet et inclure le framework dans votre code :

```
@import CoreMotion;
```

Vous noterez la nouvelle écriture des imports dans Xcode5, celle-ci permet de ne plus gérer les headers des frameworks, mais de gérer des modules, qui seront intégrés à la compilation par LLVM/Clang.

Pour récupérer l'état en cours de l'utilisateur, il faut demander à la classe CMMotionActivityManager de nous informer du statut de l'utilisateur en cas de changement.

LES NOUVEAUX IPHONES, EN QUELQUES MOTS

iPhone 5C

L'iPhone 5C est à peu de chose près un iPhone 5 relooké. Son originalité vient de son design coloré, Apple le décline en vert, bleu, jaune, rouge et blanc. Afin de mettre en avant ces couleurs, le constructeur a préféré utiliser du plastique pour la coque et plus particulièrement du polycarbonate. Une des principales nouveautés, surtout pour nous français, est que l'iPhone 5C est compatible avec les réseaux 4G de France.

iPhone 5S

Lors de la conférence du 10 septembre, le haut de gamme d'Apple accueille le tout nouvel iPhone 5S. Il apporte son lot de nouveautés matérielles comme le capteur d'empreinte TouchID ainsi que le nouveau processeur A7. Nous allons évoquer les 3 grandes nouveautés technologiques.

Les A7 et M7

Le A7 est une puce 64 bits. La Pomme promet des performances doublées par rapport à la puce A6. Ce processeur utilise un nouveau jeu d'instructions, ARMv8, qui succède à l'ARMv7 utilisé jusqu'à maintenant. Bien que le passage au 64 bits permette une augmentation de la mémoire RAM prise en charge, cet iPhone reste à 1 Go de mémoire vive. Les capacités de la puce graphique intégrée à l'A7 ont aussi été doublées. Une des nouveautés de l'ARMv8 est la prise en charge des instructions pour la virtualisation. Un pas de plus pour le remplacement des ordinateurs par les tablettes qui ne peuvent à ce jour pas faire de virtualisation.

Apple a intégré un coprocesseur nommé M7. Sa mission est de capter toutes les informations liées au mouvement (accéléromètre, gyroscope, boussole, GPS...) afin de soulager la puce A7. Il est capable de savoir si la personne se déplace à pied, en voiture, en train, etc...

Capteur d'empreinte TouchID

TouchID permet d'enregistrer jusqu'à cinq empreintes qui serviront pour l'instant, soit à déverrouiller le smartphone, soit à payer du contenu sur iTunes (Apple agrandira certainement cette liste). Ce capteur



s'accompagne d'un nouveau bouton « home » entouré d'un anneau métallique permettant la détection du doigt. L'utilisation est des plus basiques; vous devrez positionner votre doigt sous différentes positions afin que le capteur enregistre le maximum d'informations. Une fois votre empreinte configurée, il suffira de poser votre doigt sur le capteur afin de déverrouiller l'iPhone. Pas besoin de le faire glisser, poser le doigt suffit. Les empreintes sont stockées exclusivement dans la puce A7 et ne sont jamais envoyées via le réseau Internet. Bien que cette protection ne soit pas infaillible, elle permet de déverrouiller son iPhone plus rapidement qu'avec un mot de passe.





```
if([CMMotionActivityManager isActivityAvailable]) {
    CMMotionActivityManager* mam = [[CMMotionActivityManager
alloc] init];
    [mam startActivityUpdatesToQueue:[NSOperationQueue current
Queue] withHandler:^(CMMotionActivity *activity) {
        if(activity.stationary) {
            NSLog(@"%s", @"à l'arrêt");
        } else if(activity.walking) {
            NSLog(@"%s", @"en balade");
        } else if(activity.running) {
            NSLog(@"%s", @"en course");
        } else if(activity.automotive) {
            NSLog(@"%s", @"en véhicule");
        } else {
            NSLog(@"%s", @"... ?");
        }
    }];
}
```

Dans un premier temps, nous vérifions si la fonctionnalité d'analyse est disponible sur cet iPhone (seulement l'iPhone 5S et plus).

Si c'est le cas, nous instancions un nouveau gestionnaire d'activité et demandons à être informés en cas de changement de statut.

Si vous ne souhaitez pas analyser l'activité en temps réel, vous pouvez utiliser le journal d'activité via la méthode suivante.

```
- (void)queryActivityStartingFromDate:(NSDate *)start toDate:
(NSDate *)end toQueue:(NSOperationQueue *)queue withHandler:
(CMMotionActivityQueryHandler) handler
```

Cette méthode permet de récupérer l'activité de l'utilisateur à une période donnée. Attention, le journal d'activité enregistré ne contiendra que les sept derniers jours.

La dernière classe CMStepCounter permet d'obtenir le nombre de pas effectués par l'utilisateur.

Son fonctionnement est similaire à la classe vue précédemment :

```
if([CMStepCounter isStepCountingAvailable]) {
    CMStepCounter* sc = [[CMStepCounter alloc] init];
    [sc startStepCountingUpdatesToQueue:[NSOperationQueue currentQueue]
        updateOn:1
        withHandler:
        ^ (NSInteger numberOfSteps, NSDate *timestamp, NSError *error) {
            NSLog(@"%d pas au compteur", numberOfSteps);
        }];
}
```

Nous n'oublions pas de vérifier que la capacité d'analyse de pas est disponible, puis nous demandons à être informés lorsqu'un certain nombre de pas a été réalisé (ici 1).

Les comportements dynamiques dans UIKit

Parmi les nouveautés apportées dans iOS7, il y a **UIKit Dynamics**. Derrière ce nom se cachent en fait des fonctionnalités, dites dynamiques, pour nos différents composants visuels. Cela s'explique par l'ajout de deux nouvelles fonctionnalités dans UIKit :

► Les comportements dynamiques. UIKit intégrant désormais un moteur *Physics*, nous pouvons gérer le comportement d'un

iOS 7 : UN SYSTÈME REVU ET CORRIGÉ

Flat Design (design plat)

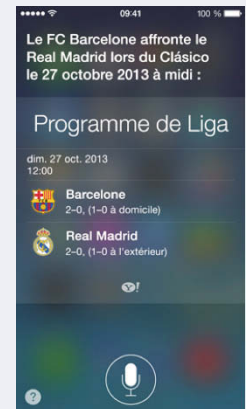
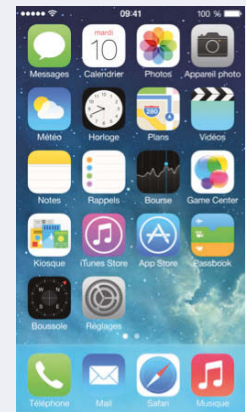
iOS 7 est un véritable lifting, qui n'avait pas subi de grands changements depuis sa sortie en 2007. Apple utilise la tendance actuelle du design : le design plat. Fini l'effet « glossy et le skeuomorphisme.

Quelques nouveautés

► le multitâche : Le multitâche n'a pas subi qu'un relooking, il utilise maintenant la mise à jour intelligente des données. Le système enregistre la fréquence et les heures auxquelles vous allez utiliser une application, par exemple votre application de réseau social favorite. Il va ensuite programmer une récupération des données, juste avant votre plage horaire d'utilisation, afin que toutes les informations soient déjà chargées au moment où vous ouvrez votre application; ceci vous permettant d'obtenir une navigation plus fluide, sans avoir à attendre le chargement des données.

► AirDrop : système de partage de fichier Apple qui utilise le WiFi ou le Bluetooth. Il est présent sur Mac depuis Mac OS X Lion (10.7) et est maintenant intégré au nouvel iOS. Cette nouvelle fonctionnalité va permettre de partager simplement des photos, des vidéos, des liens web (via Safari), des applications (via l'App Store), des contacts ou encore des lieux (via Plans) avec des personnes situées à proximité ayant eux-mêmes iOS 7 sur un iDevice compatible (iPhone 5, 5S, 5C, iPad 4 et iPad Mini). Bémol sur AirDrop iOS, il n'est actuellement pas compatible avec AirDrop Mac et il n'est pas encore possible de partager avec des utilisateurs Mac.

► iOS en voiture : Cette fonctionnalité permet de connecter son iPhone (au minimum iPhone 5) à sa voiture qui intègre alors iOS avec son système embarqué. Il est ainsi possible d'accéder à Plan, aux contacts ou encore à la musique directement en utilisant Siri avec le kit main libre de la voiture ou en utilisant l'écran de contrôle de celle-ci. « iOS en voiture » n'est pas encore disponible, mais devrait arriver courant 2014.



composant en suivant certaines propriétés : force, résistance, élasticité, etc. ;

► Des effets de mouvement, qui permettent d'ajouter des comportements particuliers entre différents composants : gravité, collision, poussée, etc.

Voyons à quoi cela il peut bien servir.

Créez un nouveau projet *Single Page View* dans Xcode, puis ajoutez deux composants dans votre vue : un bouton et un label. Ajouter les `IBOutlet`s correspondants dans le header de votre vue-contrôleur :

```
@property (nonatomic, strong) IBOutlet UIButton* aButton;
@property (nonatomic, strong) IBOutlet UILabel* aLabel;
```

Puis associez ces propriétés avec vos composants définis dans Interface Builder. Ajoutez ensuite trois variables privées dans votre vue-contrôleur :

```
@interface ITViewController : UIViewController {
    UIDynamicAnimator* _animator;
    UIGravityBehavior* _gravity;
    UICollisionBehavior* _collision;
}
```

Ces trois objets vont nous permettre de gérer le comportement dynamique de nos deux composants ainsi que deux effets de mouvement : la collision et la gravité.

Allez ensuite dans la méthode `viewDidLoad` de votre vue-contrôleur, puis ajoutez ceci :

```
- (void)viewDidLoad
{
    [super viewDidLoad];

    _animator = [[UIDynamicAnimator alloc] initWithReference
View:self.view];

    _gravity = [[UIGravityBehavior alloc] initWithItems:@[self.
aButton]];

    _collision = [[UICollisionBehavior alloc] initWithItems:@[self.
aButton, self.aLabel]];
    _collision.translatesReferenceBoundsIntoBoundary = YES;

    [_animator addBehavior:_collision];
    [_animator addBehavior:_gravity];
}
```

Dans un premier temps, nous initialisons notre contexte dynamique en prenant comme référence la vue principale de notre vue-contrôleur. Ensuite, nous ajoutons la gravité dans notre vue et définissons les composants impactés par celle-ci. Puis nous ajoutons un autre comportement pour gérer les collisions en

associant les composants impactés là aussi. Nous définissons la propriété `translatesReferenceBoundsIntoBoundary` à YES pour que notre moteur de collision prenne la vue de référence comme délimitation pour notre environnement dynamique.

Puis enfin nous ajoutons les différents comportements dynamiques (collision et gravité) et notre contexte d'animation.

Pour récapituler, la gravité ici s'appliquera uniquement au bouton, alors que la collision se fera entre le bouton et le label.

Exécutez votre application et vous devriez obtenir quelque chose comme ceci : [Fig.1](#).

Vous voyez que le bouton tombe, car attiré par la gravité, et percute le label, qui se déplace suite à cet impact, mais ne tombe pas. Ici la collision est définie pour prendre en compte les bornes de notre moteur Physics et les composants ajoutés au gestionnaire de collision. Il existe une propriété dans la classe `UICollisionBehavior` qui permet de modifier le mode de collision pour ne seulement prendre en compte que les composants visuels ou les bornes de notre contexte.

Par défaut, la collision s'applique sur les deux. Choisissons par exemple de ne définir les collisions que pour les bornes :

```
_collision.collisionMode = UICollisionBehaviorModeBoundaries;
```

Exécutez votre projet. Vous verrez que le bouton, lorsqu'il tombe, ne va plus rentrer en collision avec le label, mais va tout de même se confronter aux bornes de notre vue.

Puisque nous voyons une des propriétés de la classe `UIKitCollisionBehavior`, prenons par exemple le cas où nous souhaiterions exécuter une action particulière en cas de collision. Premièrement, ajoutez un délégué à votre moteur de collision :

```
_collision.collisionDelegate = self;
```

Puis, ailleurs dans votre implémentation de votre vue-contrôleur, ajoutez une des méthodes déléguées permettant de capter cet évènement :

```
-(void) collisionBehavior:(UICollisionBehavior*)behavior began
ContactForItem:(id<UIDynamicItem>)item1 withItem:(id<UIDynamic
Item>)item2 atPoint:(CGPoint)p {
    NSLog(@">A collision was detected on item %@ and item %@ at
point %@", item1, item2, NSStringFromCGPoint(p));
}
```

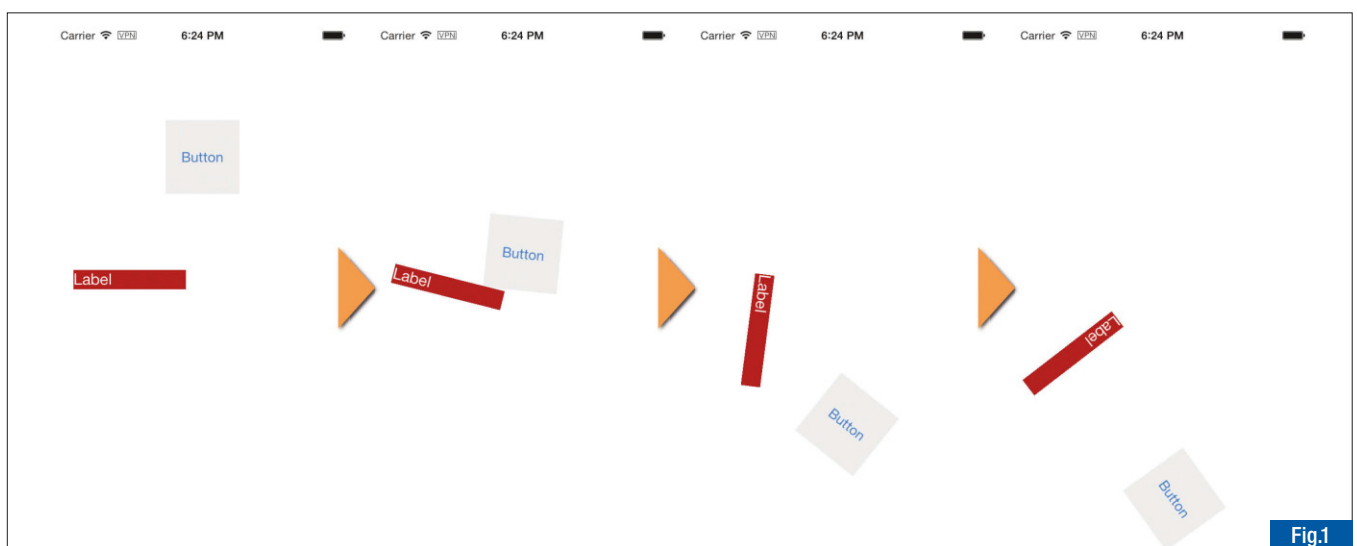


Fig.1

La collision et la gravité ne sont pas les seuls comportements dynamiques proposés :

- ▶ UISnapBehavior ;
- ▶ UIPushBehavior ;
- ▶ UIAttachmentBehavior.

Vous pouvez aussi créer vos propres comportements dynamiques en héritant de la classe `UIDynamicBehavior`.

`UISnapBehavior` est un comportement dynamique permettant le déplacement d'un composant vers un point précis. Ce déplacement se fait sous forme d'animation avec un effet d'élasticité **Fig.2**.

`UIPushBehavior` permet de simuler une poussée sur un objet. Une poussée est représentée par une force qui s'applique sur le composant. Elle peut être instantanée ou continue. Cela permet par exemple d'effectuer une poussée sur un composant lors d'une action de l'utilisateur **Fig.3**.

`UIAttachmentBehavior` permet de lier deux composants entre eux ou de lier un composant à un point particulier. Cela permet d'impacter l'action effectuée sur un composant à un autre composant. Un composant lié à un point donnera l'effet d'être attaché à une corde. Si une gravité est appliquée, cela donne un effet de pendu **Fig.4**. (Pour mieux vous faire comprendre l'animation, nous avons ajouté une ligne pointillée pour situer le point d'ancrage de notre liaison par rapport au composant).

Nous n'avons pas encore parlé d'une classe, pourtant importante, héritant aussi de `UIDynamicBehavior` :

`UIDynamicItemBehavior`.

Cette classe est la base pour la gestion des effets sur un ou plusieurs objets. Dans nos précédents essais, la collision entre le bouton et le label a entraîné la rotation de celui-ci.

Si nous avions utilisé cette classe, nous aurions pu définir que notre label ne ferait pas de rotation en cas de collision. Nous pouvons définir la résistance, l'élasticité et d'autres propriétés liées à l'environnement physique pour nos composants visuels.

TextKit

TextKit n'est pas un framework, mais un ensemble de classes incluses dans le framework UIKit.

Avant iOS 6, la solution la plus simple pour gérer du texte riche était de passer par un `UIWebView`.

iOS 6 a considérablement amélioré cela en ajoutant la possibilité de gérer des chaînes de caractères avec attributs de styles directement depuis UIKit.

Cependant, certaines fonctionnalités intéressantes nécessitaient encore le passage par l'API `CoreText`, qui malgré, sa puissance,

est une API de bas niveau. Certains composants graphiques, tels que les labels, n'étaient pas non plus directement basés sur `CoreText`.

iOS 7 a repensé cette philosophie en ajoutant **TextKit**. Cet ensemble de classes permet de faire abstraction de l'API bas niveau de `CoreText` tout en encapsulant l'ensemble de ses fonctionnalités.

De plus, les composants visuels manipulant les chaînes de caractères sont à présent tous basés sur `CoreText`.

Voyons quelques fonctionnalités simplifiées par l'utilisation de **TextKit**.

«Qui n'a jamais voulu écrire son propre éditeur de code avec coloration syntaxique ?»

Bon peut-être pas grand monde, mais avec **TextKit**, si l'idée vous venait, sachez que cela serait simple !

Depuis iOS 7, la classe `UITextView` est composée de trois nouveaux objets :

- ▶ `textStorage`, de type `NSTextStorage`, gérant la sauvegarde du texte et de ses attributs ;
- ▶ `layoutManager`, de type `NSLayoutManager`, qui gère la disposition du texte récupéré depuis un `NSTextStorage` puis adapte le texte pour respecter les contraintes imposées par l'objet de type `NSTextContainer` ;
- ▶ `textContainer`, de type `NSTextContainer`, est chargé de décrire, pour chaque ligne, le rectangle disponible pour écrire le texte.

Voici un schéma récapitulant la tâche de chacun des objets encapsulés par `UITextView` : **Fig.5**.

La classe `NSTextStorage` nous sera particulièrement utile si nous souhaitons modifier le texte écrit à la volée.

Tandis que la classe `NSTextContainer` pourra être utilisée pour modifier l'espace de saisie de texte.

Modifier le texte à la saisie

Une coloration syntaxique doit se faire instantanément lorsque l'utilisateur saisit du texte. Peu de gens acceptent que le texte soit mis à jour après un certain délai ou après avoir appuyé sur un bouton de rafraîchissement.

La classe `NSTextStorage` nous propose plusieurs méthodes à hériter pour gérer la modification *on the fly*.

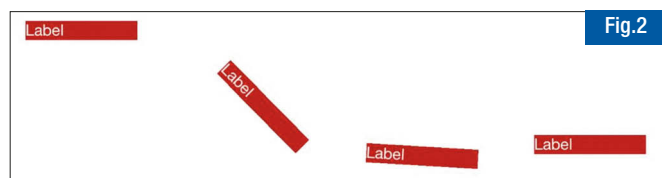


Fig.2

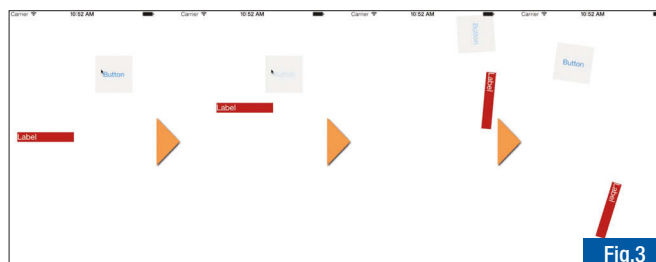


Fig.3

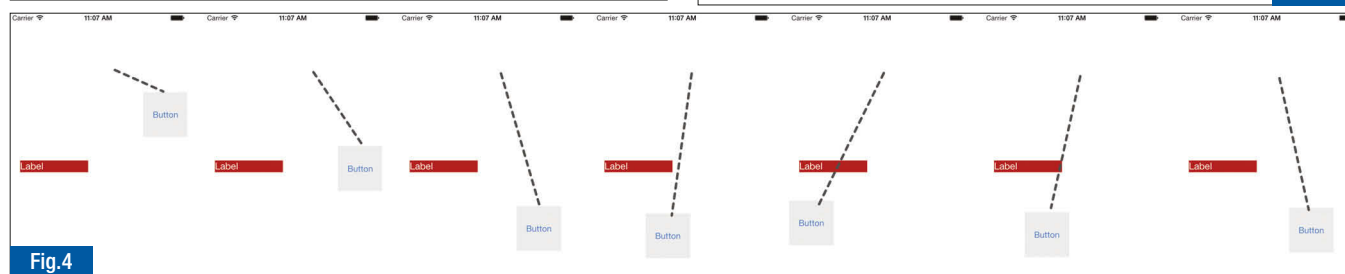


Fig.4

Premièrement, il vous faudra créer une sous-classe de `NSTextStorage`, par exemple `ITSyntaxHighlightTextStorage`. Pour simplifier la modification du texte, nous allons ajouter une variable d'instance qui servira pour sauvegarder le texte réel. Nous la nommerons `_persistence` et sera de type `NSMutableAttributedString`. `NSTextStorage` hérite de la classe `NSMutableAttributedString` donc nous devons surcharger certaines méthodes de notre classe pour utiliser notre variable d'instance plutôt que l'objet lui-même :

```
-(NSString*) string {
    return [_persistence string];
}

-(NSDictionary*) attributesAtIndex:(NSUInteger)location effective
Range:(NSRangePointer)range {
    return [_persistence attributesAtIndex:location effective
Range:range];
}

-(void) replaceCharactersInRange:(NSRange)range withString:
(NSString *)str {
    [self beginEditing];
    [_persistence replaceCharactersInRange:range withString:str];
    [self endEditing];
}

-(void) setAttributes:(NSDictionary *)attrs range:(NSRange)range {
    [self beginEditing];
    [_persistence setAttributes:attrs range:range];
    [self endEditing];
}
```

Comme précisée plus haut, la classe `NSTextStorage` fournit un ensemble de méthodes pour pouvoir analyser la modification de texte et donc modifier celui-ci à la volée. Nous devons donc signaler à notre classe que la chaîne de caractères a été mise à jour. Nous allons modifier les deux dernières méthodes, `replaceCharactersInRange:withString` et `setAttributes:range` pour informer l'objet qu'un changement s'est produit.

Pour cela, avant d'appeler `[self endEditing]`, dans la première méthode, rajoutez ceci :

```
[self edited:(NSTextStorageEditedCharacters|NSTextStorageEdite
dAttributes) range:range changeInLength:str.length-range.length];
```

Ici nous indiquons que la chaîne de caractères a été modifiée. La modification peut porter sur les attributs et la chaîne elle-même. Dans la seconde méthode, rajouter ceci, toujours avant `[self endEditing]` :

```
[self edited:NSTextStorageEditedAttributes range:range change
InLength:0];
```

Ici seuls les attributs changent, donc nous l'indiquons à notre objet. Ces deux méthodes, comme indiqué dans la documentation d'Apple, vont appeler une méthode de la classe `NSTextStorage` :

```
-(void) processEditing.
```

C'est dans cette méthode que nous allons demander la modification des attributs de notre chaîne de caractères.

Pour faciliter cela, nous allons créer deux méthodes :

- `(void) performReplacement:(NSRange)range`, chargée de récupérer l'intervalle complet de la ligne en cours d'édition par rapport à l'intervalle du texte modifié, puis qui appelle la méthode qui effectuera la modification des attributs dans cette ligne ;

- `(void) applyStyle:(NSRange)range`, chargée d'analyser le texte dans l'intervalle donné et de modifier les attributs de celui-ci.

Commençons par la deuxième méthode, pour l'exemple, nous allons colorer les chiffres 1, 2 et 3 par une couleur différente.

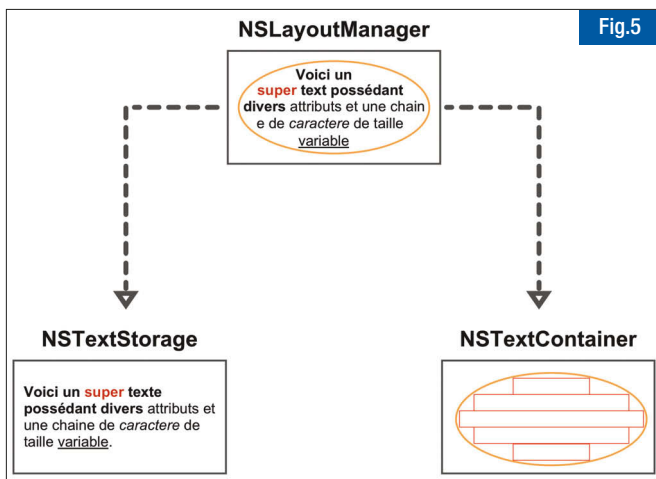
```
-(void) applyStyles:(NSRange)searchRange {

    NSArray* colors = @[UIColor redColor], [UIColor blueColor],
[UIColor orangeColor]];
    NSArray* regexes = @[@"1", @"2", @"3"];

    for (NSString* aRegex in regexes) {
        NSRegularExpression* regex = [NSRegularExpression regular
ExpressionWithPattern:aRegex options:9 error:nil];
        UIColor* color = (UIColor*) colors[[regexes indexOfObject:
aRegex]];

        [regex enumerateMatchesInString:[_persistence string] options:0
range:searchRange usingBlock:^(NSTextCheckingResult *result,
NSMatchingFlags flags, BOOL *stop) {
            NSRange matchRange = [result rangeAtIndex:0];
            [self addAttributes:@{NSForegroundColorAttributeName :
color} range:matchRange];

            if (NSMaxRange(matchRange)+1 < self.length) {
                [self addAttributes:@{ NSForegroundColorAttribute
Name : [UIColor blackColor] } range:NSMakeRange(NSMaxRange
(matchRange)+1, 1)];
            }
        }];
    }
}
```



Dans un premier temps, nous définissons nos couleurs (rouge, bleu et orange) pour les trois chiffres, respectivement 1, 2 et 3. Ensuite, pour chaque chiffre, nous créons une expression régulière et nous faisons une analyse de la chaîne de caractères pour énumérer les sous-chaînes correspondantes. Sur chacune de ces chaînes, nous appliquons les attributs nécessaires, sans oublier de rétablir les attributs par défaut après la chaîne, pour que seule la chaîne identifiée soit colorée. Nous allons maintenant mettre en place la première méthode chargée de récupérer la ligne entière, depuis la modification courante :

```
-(void) performReplacement:(NSRange)changedRange {
    NSRange realStartLineRange = [[_persistence string] line
    RangeForRange:NSMakeRange(changedRange.location, 0)];
    NSRange extendedRange = NSUnionRange(changedRange, realStart
    LineRange);

    NSRange realEndLineRange = [[_persistence string] lineRange
    ForRange:NSMakeRange(NSMaxRange(changedRange), 0)];

    extendedRange = NSUnionRange(changedRange, realEndLineRange);

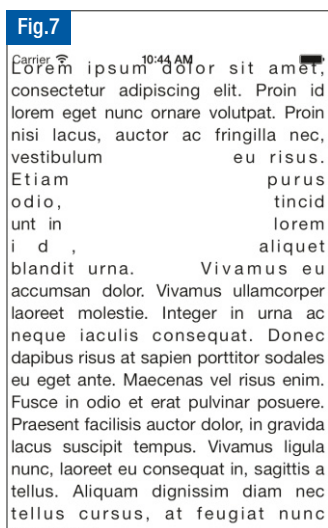
    [self applyStyles:extendedRange];
}
```

Nous utilisons la méthode `lineRangeForRange` pour récupérer le début et la fin de la ligne en cours de modification, puis nous appelons la deuxième méthode, créée précédemment pour appliquer les nouveaux styles.

Maintenant que ces deux méthodes sont créées, il faut que notre sous-classe de `NSTextStorage` les inclue dans son processus de traitement de texte. Pour cela, comme indiqué plus haut, nous allons modifier le comportement de la méthode `processEditing` :

```
-(void) processEditing {
    [self performReplacement:[self editedRange]];
    [super processEditing];
}
```

Il ne reste plus qu'à mettre en place un objet de type `UITextView` et d'utiliser notre nouvelle classe en tant que



`textStorage`. Vous devriez obtenir une coloration syntaxique assez simpliste, mais fonctionnelle Fig.6.

Modification de l'espace de saisie

Pour faire cela, nous allons utiliser la classe `NSTextContainer`. Celle-ci contient une propriété appelée `exclusionPaths` de type `NSArray`. Elle permet de définir des formes d'exclusion dans l'espace de saisie. Ces formes d'exclusion permettent d'indiquer au conteneur que le texte ne doit surtout pas pénétrer dans cet espace et doit s'adapter autour.

Reprenez le code précédent et ajoutez un `UIBezierPath` dans l'attribut `exclusionPaths`, de la propriété `textContainer`, de votre `UITextView`.

```
container.exclusionPaths = @[UIBezierPath bezierPathWithOval
    InRect:CGRectMake(60, 100, 200, 100)];
```

Relancez votre application et écrivez du texte. Vous verrez qu'au milieu de celui-ci, une forme ovale sera complètement vide et le texte s'adaptera pour être autour de celle-ci Fig.7.

Un espace vide, c'est bien, mais c'est plus utile si cet espace est rempli par quelque chose en lui donnant une utilité quelconque. Cela pourrait très bien servir pour intégrer des images, des tableaux ou tout autre média dans notre texte. Cela serait très utile si nous souhaitions développer un éditeur de texte, type *Microsoft Word*.

Pour notre démonstration, nous allons simplement utiliser cet espace pour mettre un bouton :

```
UIButton* button = [[UIButton alloc] initWithFrame:CGRectMake
    Make(60, 115, 200, 90)];
[button setTitle:@"Je suis un bouton placé en plein milieu
    du texte !" forState:UIControlStateNormal];
[button setBackgroundColor:[UIColor orangeColor]];
[button setTitleColor:[UIColor whiteColor] forState:UIControl
    StateNormal];
button.titleLabel.font = [UIFont boldSystemFontOfSize:20];
button.titleLabel.numberOfLines = 0;
button.titleLabel.adjustsFontSizeToFitWidth = YES;
button.titleLabel.textAlignment = NSTextAlignmentCenter;
button.layer.cornerRadius = 45;
[self.view addSubview:button];
```



Nous avons un peu customisé le bouton pour s'adapter à la forme ovale donnée comme forme d'exclusion Fig.8.

Le 64 bits

Grande nouveauté de l'iPhone 5S, le processeur A7 est en 64 bits, ce qui est une première pour un périphérique mobile.

La première chose à savoir quand on se pose la question de la migration vers une application compatible 64 bits sous iOS, c'est qu'actuellement Xcode 5 ne nous permet pas

de créer une application 64 bits si on ne met pas iOS 7 en tant que plateforme cible minimum. La possibilité de générer un bundle avec une application 32 bits pour iOS 6 et une application 64 bits pour iOS 7 est prévue par Apple.

Il est à noter aussi qu'iOS 7 inclut l'ensemble des frameworks 32 et 64 bits. Tant qu'aucune application 32 bits n'est installée sur le système, les frameworks 32 bits ne sont pas chargés en mémoire. Cela implique que dès qu'une application non compatible 64 bits est installée, le système va charger les frameworks des deux environnements.

Ce fonctionnement, bien que transparent pour l'utilisateur, a un impact : si une application 32 bits est installée, le périphérique sera théoriquement plus lent et disposera de moins de mémoire pour fonctionner.

La taille et l'alignement des types

Tous ceux qui ont fait du C/C++ plutôt bas niveau savent cela, mais un petit rappel ne fait pas de mal :

La taille d'une variable et son alignement en mémoire ne sont pas dépendant du langage, mais de la plateforme d'exécution.

Cette notion est à prendre en compte lors de vos développements. Prenons l'exemple de code suivant :

```
long anInteger = 21474836477523;
```

Si vous compilez votre application en 64 bits, pas de problème. Si vous compilez votre application en 32 bits, cela ne fonctionnera pas, en effet le LONG_MAX en 32 bits est de **2147483647**. De plus, le compilateur ne détectera pas forcément l'erreur, et c'est à l'exécution que vous trouverez une incohérence.

Voici un tableau récapitulatif des différents changements intervenant sur les types entiers dans la migration 32/64 bits [Fig.9](#).

Voici le même tableau pour les types flottants [Fig.10](#).

Ces changements entre les deux architectures nécessitent de prendre en compte certains points :

- ▮ **L'empreinte mémoire de votre application sera plus grande.** En effet, la plupart des types régulièrement utilisés dans

Floating-point type	ILP32 size	LP64 size
float	4 bytes	4 bytes
double	8 bytes	8 bytes
CGFloat	4 bytes	8 bytes

Fig.10

Integer data type	ILP32 size	ILP32 alignment	LP64 size	LP64 alignment
char	1 byte	1 byte	1 byte	1 byte
BOOL, bool	1 byte	1 byte	1 byte	1 byte
short	2 bytes	2 bytes	2 bytes	2 bytes
int	4 bytes	4 bytes	4 bytes	4 bytes
long	4 bytes	4 bytes	8 bytes	8 bytes
long long	8 bytes	4 bytes	8 bytes	8 bytes
pointer	4 bytes	4 bytes	8 bytes	8 bytes
size_t	4 bytes	4 bytes	8 bytes	8 bytes
NSInteger	4 bytes	4 bytes	8 bytes	8 bytes
CFIndex	4 bytes	4 bytes	8 bytes	8 bytes
fpos_t	8 bytes	4 bytes	8 bytes	8 bytes
off_t	8 bytes	4 bytes	8 bytes	8 bytes

Fig.9

une application (long, integer, etc.) n'ont plus la même taille. Dans la majeure partie des cas, cette taille a augmenté, ce qui veut dire que votre mémoire sera plus sollicitée si une application fonctionne en 64 bits qu'en 32 bits ;

- ▮ **L'échange d'information est plus risqué.** Cela ne veut pas dire que ce n'est plus possible, mais si vous proposez une version 32 et 64 bits d'une même application partageant des informations sur le Cloud, par exemple, il est nécessaire de faire attention aux informations. Si l'application sauvegarde un NSInteger avec la valeur maximale de ce type pour une application 64 bits, l'application 32 bits essaiera de sauvegarder cet entier dans une variable qui aura un espace mémoire plus faible, ce qui introduira une troncature ;
- ▮ **Les calculs peuvent avoir des résultats différents.** Un calcul dépassant la valeur maximale d'un type donné sur un processeur 32 bits ne produira pas le même résultat que sur un processeur 64 bits, si ce même type a une taille différente en mémoire. Ce dernier permettra certainement de contenir cette valeur alors qu'en mode 32 bits, le dépassement entraînera un comportement imprévu (troncature) ;
- ▮ **Le passage de 64 bits vers 32 bits entraîne des risques.** Une variable 64 bits transférée dans une variable 32 bits entraînera une troncature de celle-ci. La valeur comparée sera donc différente, et entraînera une différence qui n'en sera pas réellement une au sens fonctionnel.

L'appel de méthode

Lors du passage en 64 bits, le fonctionnement d'appel de méthode peut être différent.

Avant d'expliquer ce qui peut être différent, il est nécessaire de comprendre ce qui se passe dans le bas niveau de votre système.

Celui-ci est représenté par l'ABI. Ce terme, voulant dire *Application Binary Interface*, définit une interface de communication pour l'ensemble des programmes s'exécutant sur une plateforme.

Pour résumer, c'est l'ABI qui définit comment fonctionne le processus d'appel de méthode sur telle ou telle plateforme. Par exemple, si l'émetteur d'une méthode doit placer les paramètres de celle-ci dans des registres mémoire, ou dans la mémoire vive. Dans la plupart des cas, les développeurs haut niveau que nous sommes, n'ont pas besoin de connaître cette spécificité, car c'est le framework haut niveau qui gère les appels à l'ABI.

Cependant, si vous souhaitez développer en assembleur, vous devez connaître comment fonctionne l'ABI de la plateforme de développement.

Le développeur utilisant le moteur d'exécution 64 bits sur iOS doit être au courant de ce fonctionnement, car, contrairement au fonctionnement 32 bits où l'ABI décrivait le même fonctionnement pour l'ensemble des types de méthode, le fonctionnement diffère dans le cas où la fonction accepte un nombre variable d'arguments ("*variadic method*").

Encore une fois, ici il est question d'ABI, donc cela est transparent pour le développeur. Mais le développeur doit tout même connaître cette spécificité, car cela implique certaines règles lorsque l'on travaille sur du 64 bits avec iOS :

- ▮ **Chaque fonction utilisée dans notre code doit avoir un prototype.** En 32 bits, on pouvait se permettre de pouvoir faire sans. En 64 bits il faudra absolument éviter cela ;
- ▮ **Éviter au maximum de faire du casting de fonction.** Dans le



cas où vous ne prototypez pas exactement chaque méthode, le compilateur ne rejettera pas à la compilation. Mais si vous l'avez prototypé, celui-ci s'assurera que la méthode appelée correspond au prototype. Cependant, même si vous prototypez correctement vos méthodes, cela n'empêchera pas de faire du casting de méthode. Or le processeur 64 bits imposant un fonctionnement différent au niveau de l'API en fonction du type de méthode, ce casting peut être dangereux au niveau de la gestion mémoire et, plus généralement, de la gestion bas niveau de l'application (pointeur, allocation mémoire, etc.).

L'exécution d'assembleur embarqué

Voici le dernier point nécessitant une attention particulière : Si votre application embarque du code assembleur, principalement pour effectuer des optimisations, il faudra retravailler ce code. Les instructions ASM 32 bits diffèrent des instructions 64 bits.

Et surtout, faites attention, car la convention adoptée pour iOS est différente du standard proposé par ARM, donc soyez sûr de bien respecter les conventions adoptées par Apple pour iOS.

Synthèse

Pour récapituler, voici les différents points à prendre en compte pour éviter tout problème :

- ▮ Les variables de type `long` et les pointeurs n'ont plus la même taille et le même alignement en mémoire selon l'environnement 32 ou 64 bits ;
- ▮ Attention au troncage lors des calculs pouvant entraîner des valeurs supérieures à l'espace mémoire alloué en 32 bits sur un 64 bits (et vice versa) ;
- ▮ Évitez au maximum l'utilisation de variables ayant un alignement différent si vous prévoyez une application 32 et 64 bits ;
- ▮ Attention lors de l'utilisation de variables ayant des tailles et des alignements différents dans des structures complexes ;
- ▮ Réécrivez l'ensemble de vos codes assembleur embarqués pour être compatible 64 bits ;
- ▮ Évitez le `cast` de fonction.

Les nouveaux modes de gestion du multitâche

iOS est un système qui, au départ, ne laissait aucune application tourner en tâche de fond, ou presque... En effet, même si le développeur n'avait pas accès à l'API pour avoir une application tournant en tâche de fond, Apple avait quelques applications qui fonctionnaient comme cela.

Avec l'arrivée d'iOS 4, Apple a ouvert cette API et a autorisé les développeurs à écrire des applications capables de fonctionner en tâche de fond. Cependant, Apple fidèle à ses habitudes a restreint les possibilités pour le développeur de faire fonctionner son application en tâche de fond.

En effet, seul un certain panel d'applications peut prétendre à l'utilisation du mode tâche de fond :

- ▮ Les applications permettant la lecture de son (media player) ;
- ▮ Les applications d'enregistrement de son ;
- ▮ Les applications de VoIP ;
- ▮ Les applications de géolocalisation ;
- ▮ Les applications fonctionnant avec des accessoires externes ;
- ▮ Les applications récupérant du contenu depuis l'application Newsstand.

Avec l'arrivée de CoreBluetooth et du Bluetooth Low Energy

(4.0) dans iOS, Apple a rajouté deux nouveaux modes :

- ▮ Les applications agissant comme des périphériques Bluetooth 4.0 de type *peripheral* ;
 - ▮ Les applications ayant établi une communication Bluetooth 4.0 avec des appareils de type *peripheral (central)*.
- (*Central* et *Peripheral* sont les noms désignés dans la norme Bluetooth Low Energy pour parler respectivement d'un client et d'un serveur de données)

Vous noterez cependant qu'aucun mode ne permet au développeur d'effectuer des tâches régulières pour récupérer du nouveau contenu par exemple. Auparavant, celui-ci devait toujours s'appuyer sur le système de notifications Push fourni par Apple, nécessitant un serveur capable de gérer un parc de périphériques par applications.

Apple a donc décidé d'introduire deux nouveaux modes d'exécution en arrière-plan :

- ▮ `fetch` - permet à l'application d'effectuer régulièrement une requête vers le web pour télécharger du nouveau contenu ou `uploader` du contenu vers celui-ci ;
- ▮ `remote-notification` - permet à l'application de recevoir des notifications Push (disponible auparavant) et d'exécuter des requêtes vers le web pour télécharger du contenu associé à cette notification.

Vous l'avez sûrement compris, le deuxième mode est une adaptation d'un mécanisme fonctionnant déjà, mais nécessitant aujourd'hui de le définir explicitement dans la configuration de votre projet.

En effet, comme indiqué plus haut, les notifications Push existaient déjà dans les applications iOS puisqu'elles permettaient justement d'être averti des nouveautés. Cependant, celles-ci étaient limitées au contenu autorisé par Apple dans les notifications : une image, un son, un titre et un texte ne dépassant pas une certaine taille.

Dorénavant, le système de notification reste inchangé, mais il se voit ajouter la possibilité d'exécuter du code permettant le téléchargement de contenu additionnel.

Globalement ces deux modes se ressemblent, ils permettent d'avoir accès au contexte de l'application, de manière non prévisible, car géré par un serveur ou par le système, et permettent de mettre à jour les informations clients/serveurs durant ce laps de temps.

Attention cependant, ce temps est géré par le système (environ 20 à 30 secondes) et ne peut en aucun cas être dépassé. Si votre requête prend trop de temps, votre application va tout simplement lever une exception et annuler l'ensemble du *polling* tant que celle-ci ne sera pas relancée.

Les deux fonctionnements étant assez similaires.

Dans un premier temps, nous allons mettre en place un serveur web pour générer du contenu. Celui-ci sera développé avec NodeJS et générera un UUID à chaque requête pour simuler le nouveau contenu :

```
function s4() {
    return Math.floor((1 + Math.random()) * 0x10000).toString(16).substring(1);
};

function guid() {
    return s4() + s4() + '-' + s4() + '-' + s4() + '-' + s4() + '-' + s4() + s4() + s4() + s4();
}
```

```
};

var http = require('http');
http.createServer(function (req, res) {
    res.writeHead(200, {'Content-Type': 'text/plain'});
    res.end(guid());
}).listen(1337);
```

Prenez un navigateur web et accédez à la page web de votre serveur, vous devriez obtenir un UUID en *plain-text*. Passons ensuite au code réellement intéressant, l'Objective-C. La première chose à faire est d'ajouter le mode fetch à votre application. Allez dans l'onglet *Capabilités* de votre projet et cocher l'option *Background fetch*.

Ensuite, nous devons faire deux choses :

- 1 Indiquons à notre application que nous souhaitons régulièrement exécuter une requête vers le web pour récupérer du contenu ;
- 2 Fournissons une méthode déléguée de notre application qui sera régulièrement exécutée pour effectuer la requête web afin de récupérer le contenu en ligne.

Pour indiquer à notre application que nous souhaitons mettre en place un *polling* régulier, il faut utiliser la méthode `setMinimumBackgroundFetchInterval:` :

```
- (BOOL)application:(UIApplication *)application didFinishLaunchingWithOptions:(NSDictionary *)launchOptions
{
    // Override point for customization after application launch.
    [application setMinimumBackgroundFetchInterval:UIApplicationBackgroundFetchIntervalMinimum];
    return YES;
}
```

Nous avons mis la méthode directement après le lancement de l'application, mais cela aurait pu se faire plus tard dans l'application. L'important c'est que celle-ci soit correctement lancée avant d'initialiser le *polling*.

Passons ensuite au code pour récupérer le contenu en ligne. Pour cela, il faut fournir le code de la méthode `application:performFetchWithCompletionHandler:` :

```
-(void) application:(UIApplication *)application performFetchWithCompletionHandler:(void (^)(UIBackgroundFetchResult))completionHandler {
    NSLog(@"Demarrage du Background Fetch»);


    NSURLRequest* request = [NSURLRequest requestWithURL:[NSURL URLWithString:@"http://localhost:1337"]];

    [NSURLConnection sendAsynchronousRequest:request queue:[NSOperationQueue mainQueue] completionHandler:^(NSURLResponse* response, NSData *data, NSError *connectionError) {
        NSString* str = [[NSString alloc] initWithData:data encoding:NSUTF8StringEncoding];
        UINotification* notif = [[UINotification alloc] init];
        notif.alertBody = str;
        notif.alertAction = @"New UUID !»;
        notif.fireDate = [[NSDate date] dateByAddingTimeInterval:3];
        [application scheduleLocalNotification:notif];
        NSLog(@"Fin du Background Fetch»);
        completionHandler(UIBackgroundFetchResultNewData);
    }];
}
```

Ici nous créons une requête vers notre machine (nous utilisons le simulateur dans notre exemple) puis nous demandons son exécution de manière asynchrone en utilisant les blocks Objective-C. Une fois la requête effectuée, nous affichons une notification locale avec le contenu récupéré via la requête HTTP.

Dans la 2^e partie du dossier iOS 7, nous aborderons les moteurs de jeux, le GameController, AirDrop, les évolutions des principaux frameworks d'iOS !

 **Vincent Saluzzo**
Ingénieur Développement Mobile – Expert iOS - Itelios
<http://www.vincentsaluzzo.com>

 **Marc Duvignaud**
Ingénieur Développement Mobile - Itelios

 **Fabien Ducret**
Ingénieur Développement web & Mobile - Itelios

itelios

Restez connecté(e) à l'actualité !

- L'**actu** de Programmez.com : le fil d'info **quotidien**
- La **newsletter hebdo** : la synthèse des informations indispensables.
- **Agenda** : Tous les salons, barcamp et conférences.

Abonnez-vous, c'est gratuit !

www.programmez.com



The screenshot shows the Programmez.com website. At the top, there's a navigation bar with links like 'Accueil', 'Actualités', 'Livres', 'Agenda', 'Solutions', and 'Plus iOS'. Below the navigation bar, there are several featured articles and advertisements. On the left, there's a section for 'Abonnement' (Subscription) and 'Actualités' (News). In the center, there are articles about 'Concours Azure Raytracer', 'Nouveaux livres sur Programmez', and 'Le Groupe KPI, SSI spécialisée sur SAP'. On the right, there are advertisements for 'Microsoft' (Windows Azure), 'IRIScan Book 2', and 'Programmez.com' (newsletter). At the bottom, there's a section for 'Tribune Libre' (Free Forum) and 'UN BUG + VISSEUX'.



Android, quelques rappels

Ce dossier d'introduction vous permettra de découvrir comment le système de Google a réussi à s'imposer dans un domaine ultra concurrentiel en un temps record. Les fondamentaux nécessaires au développement Android vous seront aussi détaillés. Ainsi, vous serez à même d'entreprendre vos propres créations !

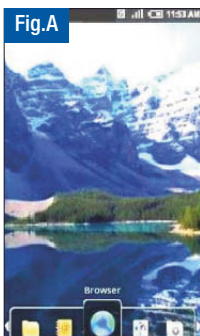
Historique : les versions d'Android

Le commencement

Android est un système d'exploitation mis au point par une startup du même nom. Prévu pour les plates-formes mobiles, le système est racheté par Google en 2005.

Android est ensuite enrichi par de nombreuses sociétés via l'Open Handset Alliance. Dès la première version du système, les fonctionnalités essentielles sont présentes. On retrouve un marché d'applications : Android Market (devenu Google Play en 2012), un navigateur web ainsi que certains services de Google sous forme mobile (Maps, Street View, Gmail) **Fig.A**.

Le premier téléphone à sortir sous Android est le HTC G1. L'appareil sort aux États-



Unis en partenariat avec T-Mobile en 2008. Il est face aux concurrents, le Nokia N95 et l'iPhone premier du nom. Il propose un écran tactile ainsi qu'un clavier physique. Android arrive en France via l'opérateur téléphonique Orange qui propose le HTC Dream, adaptation du G1. Son petit frère, le HTC Magic, arrive chez SFR quelques mois plus tard. Les deux terminaux sont assez semblables, la différence majeure étant l'absence de clavier physique sur le HTC Magic, qui le mettait en concurrence avec l'iPhone.

En avril 2009, une mise à jour importante, la version **1.5 (Cupcake)** du système ajouta notamment le support des widgets sur le bureau, l'enregistrement de vidéos au format MPEG-4, le copier-coller, ainsi qu'une refonte de l'apparence du système.

La version Cupcake fut la première à être utilisée par un nombre non négligeable d'utilisateurs, c'est pourquoi celle-ci est considérée par beaucoup comme la version initiale d'Android.

La version **1.6 (Donut)** voit le jour en septembre 2009. Elle ajoute une meilleure inté-



Fig.B

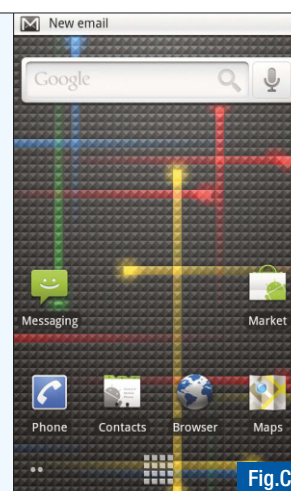


Fig.C

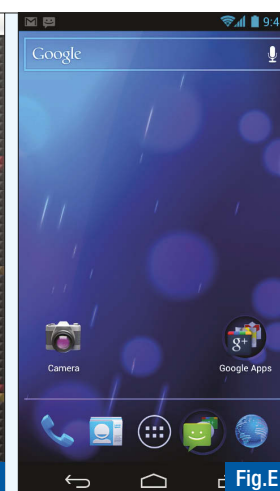


Fig.E

gration de la recherche vocale, une amélioration de l'interface du système et des applications standard.

Cette version 1.6 remplace la version 1.5 sur de nombreux téléphones : la mise à jour est en effet proposée par la plupart des constructeurs **Fig.B**.

Démocratisation

Les HTC Dream et Magic, considérés alors comme les premiers « Google Phone » voient arriver un nouveau représentant du système, le Motorola Droid.

Tournant sous la nouvelle version du système (**Eclair**), Google met les bouchées doubles face à son principal concurrent : l'iPhone 3GS.

Aux États-Unis les campagnes de pub sont agressives, la campagne « iDon't » met en avant les différences entre le système d'Apple et celui de Google.

La version **Eclair** apporte une optimisation du système, ainsi qu'une toute nouvelle interface utilisateur qui marque une grande évolution vis-à-vis de la version **Donut**.

Le multi-touch arrive sur Android bien que le zoom avec deux doigts ne soit pas disponible dans tous les pays pour éviter des procès. Des fonds d'écrans animés sont disponibles pour affirmer la nouvelle identité du système. De nombreux téléphones ne bénéficient pas de la mise à jour vers la

version **2.0** d'Android. Les constructeurs ne sont en effet nullement obligés d'effectuer un suivi de leurs appareils. On parle alors de fragmentation : on remarque que le parc des téléphones Android se scinde entre les différentes versions.

Android **2.2 (Froyo)** et **2.3 (Gingerbread)** accentuent la nouvelle identité d'Android en proposant des mises à jour de l'interface utilisateur, ainsi que des optimisations qui rendent le système plus réactif. Le système ajoute de nouvelles fonctionnalités : support du NFC, point d'accès mobile, chat vidéo, prise en charge des écrans haute résolution.

Les Nexus One et Nexus S sont lancés par Google pour mettre en avant les nouveautés hardware et software.

C'est à cette époque que le nombre de téléphones disposant du système Android décolle.

Chaque constructeur possède plusieurs modèles de smartphones. HTC, Sony, Samsung et bien d'autres proposent plusieurs gammes de mobiles pour satisfaire le plus grand nombre d'utilisateurs. La version qui restera longtemps majoritaire sur les téléphones Android est la **2.3.3**. **Fig.C**.

Support des Tablettes

Android arrive sur les tablettes avec la version **3.0 (Honeycomb)**. Le système sait

maintenant s'adapter en fonction des tailles d'écrans. Les applications tablettes et téléphones sont communes, mais celles-ci s'affichent différemment en fonction du périphérique, pour tirer parti de la taille de l'écran.

Le niveau qualitatif du système est à la hausse : l'ergonomie est repensée et les performances améliorées.

Avec la mise à jour **3.0** qui n'était pas à destination des téléphones portables, il n'a pas eu une visibilité aussi grande que la version précédente, mais a bien préparé l'arrivée de la nouvelle génération de téléphones avec le support des processeurs multi-cœur et des grandes tailles d'écrans **Fig.D**.



Fig.D

Aujourd'hui

La version **4.0** d'Android (**Ice Cream Sandwich**) est disponible sur tablettes, téléphones et même certains ordinateurs portables.

Cette version apporte une totale refonte graphique pour les téléphones et s'aligne avec la version **3.0** du système.

Le Galaxy Nexus et le Nexus 4 sont les téléphones mis en avant par Google pour représenter son système.

Cette version **4.0** ne fut pas disponible sur tous les appareils, par souci technique (manque de mémoire vive sur certains modèles), mais aussi par l'envie des constructeurs de sortir de nouveaux modèles et de renouveler les gammes.

La fragmentation augmente, on remarque une diminution du nombre d'appareils en **1.x**.

Plus de la moitié des terminaux font partie de la branche **4.x**.

Android **4.1**, **4.2** et **4.3** (**Jelly Bean**) apportent de nouvelles applications au sein du système.

Reconnaissance vocale sans internet, assistant personnel Google Now, nouveau clavier tactile, font partie des modifications de cette version.

Ces mises à jour sont principalement qualitatives **Fig.E**.

L'adoption par le grand public est massive : plus d'un milliard d'appareils Android sont activés dans le monde.

POSSIBILITÉS OFFERTES PAR ANDROID

Diversité des appareils

Android étant un OS open source, les différents constructeurs du marché peuvent l'incorporer dans tous types d'appareils (tablettes, smartphones, téléviseurs, netbooks...).

Il existe de nombreux constructeurs : Samsung, HTC, Sony, Asus, Archos et bien d'autres.

Cette diversité permet d'offrir une large gamme de mobiles. On peut trouver des téléphones disposant d'écrans totalement différents. (Samsung Galaxy Y 3 pouces, Samsung Galaxy Mega 6,3 pouces). Le prix va également varier (Sony Xperia Z Ultra à 800 €, Samsung Chat 357 à 100 €).

Pouvoir choisir son appareil

Choisir son téléphone ce n'est pas seulement une histoire de prix et de taille d'écran. Étant donné le grand nombre d'appareils mobiles tournant sous Android et la concurrence entre constructeurs, vous trouverez forcément un modèle qui vous correspondra. Vous aimez prendre en photo tout ce qui bouge ?

Optez pour un téléphone Android muni d'une très bonne optique. Vous avez tendance à vous perdre à chaque fois que vous bougez ? Choisissez un mobile qui possède la fonction GPS.

Il existe encore beaucoup d'autres fonctionnalités sur les téléphones Android telles que la radio FM, le capteur d'altitude, la boussole, le NFC, l'infrarouge, le Bluetooth... À vous de faire le bon choix !

Personnalisation des constructeurs

Outre le matériel embarqué dans vos appareils, les constructeurs peuvent personnaliser le système. Cette personnalisation peut ajouter de nouvelles applications (assistant vocal, compagnon sportif). Elle peut aussi se remarquer par une modification visuelle du système **Fig.F**.

UN MONDE COLLABORATIF

Android est fondé sur un monde collaboratif qui permet aux développeurs de s'entraider.

Java

Java est un langage connu et reconnu. Sa fiabilité et son approche orientée objet permettent de mettre en place de manière simple et puissante vos applications.

Le code source

Le code source d'Android est librement téléchargeable via l'AOSP : Android Open Source Project. Vous pourrez trouver le code source sur <http://source.android.com>.

Ce code source est édité par l'Open Handset Alliance. L'OHA est un consortium de plusieurs entreprises (dont principalement Google) qui visent à créer et faire évoluer la plate-forme ouverte qu'est Android.

Les bibliothèques et tutoriaux

Les bibliothèques sont des projets Android pouvant être liés à des applications Android. Elles apportent la plupart du temps une nouveauté que l'on ne peut pas utiliser nativement dans notre application ou des routines de codes génériques.

Voici quelques sites répertoriant des librairies ou tutoriaux :

- <http://android.developpez.com>

- <http://www.androidviews.net>



Fig.F



Les roms

Une rom est de manière globale le système d'exploitation du téléphone. Il existe des roms dites « officielles » : ce sont les roms qui sont à l'origine créées par les constructeurs pour les téléphones Android.

Mais du fait de la possibilité laissée à tout utilisateur de modifier les roms existantes, on peut désormais trouver un grand nombre de roms non officielles, dites « custom ». Parmi les plus connues, on trouvera : MIUI, CyanogenMod ou bien Paranoid Android.

Les stores applicatifs

Google fournit un marché d'applications pour télécharger de nouveaux programmes : le **Play Store**.

Android permet de télécharger des applications depuis d'autres endroits, c'est pourquoi de nouveaux marchés sont apparus comme **Aptoide**, **AppBrain** et bien d'autres...

OUVERTURE VERS LE FUTUR DE LA PLATEFORME

Android tend de plus en plus à se déclinier pour de nouveaux appareils, on le voit déjà avec l'explosion des ventes des tablettes sous Android, mais également depuis que l'OS s'étend aux ordinateurs personnels chez des constructeurs comme Lenovo, HP ou encore Acer.

On retrouve également de l'Android embarqué dans des objets « connectés », vous avez difficilement pu passer à côté des Google Glass ou des « smartwatches ». La plateforme est en train de redécouvrir ses frontières et nous espérons être surpris !

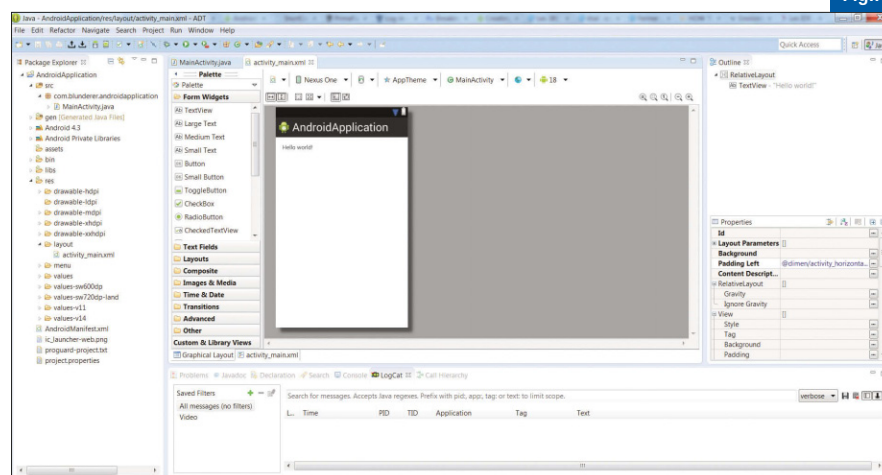
Michaël Ohayon (Epitech Promo 2016),
Denis Mondon (Epitech Promo 2016),
Jonathan Picques (Epitech Promo 2016),
Victor Leduc (Epitech Promo 2016),
Vincent Nêel (Epitech Promo 2016),
Jérémy Amsellem (Epitech Promo 2017)



IDE : votre meilleur ami

Pour développer un programme Android, il existe plusieurs IDE. Nous n'en retiendrons ici que trois mais sachez qu'il en existe d'autres. Les trois présentés ci-dessous fonctionnent sur Windows, OS X et Gnu/Linux.

Fig.1



► **Eclipse** est l'IDE recommandé pour le développement Android. Google a mis en place un pack pré-configuré : l'ADT-Bundle. Ce bundle contient tous les éléments requis pour créer vos applications **Fig.1**. Téléchargement : <http://developer.android.com/sdk/>

► **IntelliJ IDEA** est un IDE créé par JetBrains, il permet de développer en plusieurs langages dont Java pour Android. Il est disponible en version gratuite pour le développement mobile Android **Fig.2**. Téléchargement : <http://www.jetbrains.com/idea/download/>

► **Android Studio** est un IDE en développement présenté lors de la Google I/O 2013 en mai dernier qui est basé sur IntelliJ IDEA. Il n'est toujours pas sorti en version stable **Fig.3**. Téléchargement : <http://developer.android.com/sdk/installing/studio.html>

STRUCTURE D'UN PROJET CLASSIQUE ANDROID

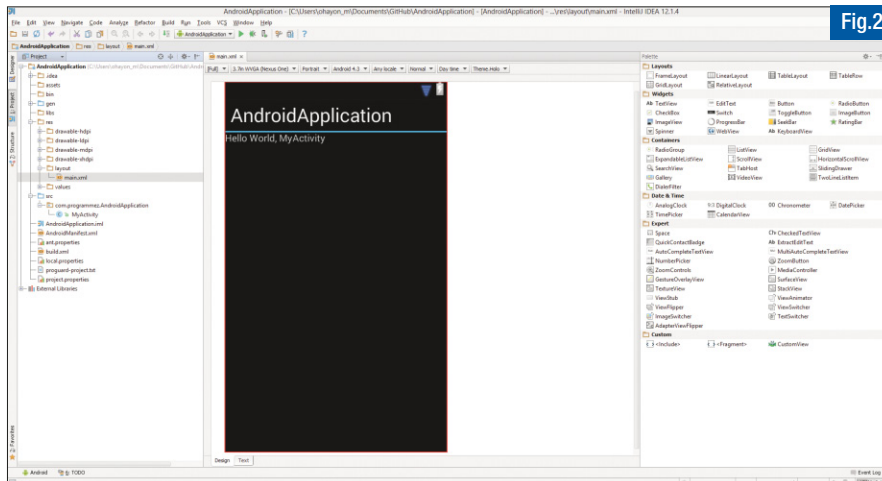
Le fichier « AndroidManifest » (Description de l'application)

Abordons maintenant la structure des fichiers nécessaires à la réalisation d'une application : si vous avez auparavant rencontré du code Android, vous avez certainement dû remarquer la présence de ce fichier : **AndroidManifest.xml**.

Il est indispensable que vous ayez ce fichier à la racine du code de votre application, sans quoi vous ne pourrez compiler votre travail.

Le manifeste contient des informations sur votre application ; comme notamment son nom complet (le package name), les différentes permissions que votre programme demandera à l'utilisateur lors de son installation (par

Fig.2



Raccourcis utiles pour Eclipse

Ctrl + Shift + F : Indenter le code du fichier ouvert

Shift + Alt + R : Renommer intelligent (renomme une méthode dans tous les fichiers)

Ctrl + Shift + O : Organiser les imports automatiquement

Ctrl + Clic Gauche : Aller à la déclaration / source de la classe en question

Ctrl + F : Faire une recherche / un remplacement dans le fichier ouvert

Ctrl + H : Faire une recherche dans tous les fichiers du projet (Onglet File Search)

Raccourcis utiles pour Android Studio

Ctrl + Alt + L : Indenter le code du fichier ouvert

Ctrl + E : Voir les derniers fichiers utilisés

Ctrl + F : Faire une recherche dans le fichier courant

Ctrl + R : Remplacer une chaîne de caractères par une autre

F4 : Ouvrir les propriétés du module

exemple : utiliser le Wi-Fi, la mémoire de stockage ou le GPS de votre appareil), le numéro de version de votre projet ou encore les versions d'Android qui seront supportées par votre application **Fig.4**.

Dossier « src » (Code source de l'application)

Votre projet comportera inévitablement un dossier « **src** », celui-ci contiendra le code source de votre application.

Package name

Le « **Package name** » est un identifiant unique que vous attribuerez à votre application.

Il existe un package name par projet et contrairement au nom de votre application, on ne peut pas le changer de version en version.

Cet identifiant sera toujours similaire à celui-ci : **com.mon_nom.mon_application**.

La première partie du package name (**com.mon_nom**) est une information sur l'entité qui a créé l'application : cela peut être le nom d'une entreprise ou du développeur par exemple. La seconde partie (**mon_application**) sert à identifier l'application concernée.

On peut également utiliser des points dans la seconde partie du package name : ce qui peut donner des noms de package comme « **com.mon_nom.mon_application.sous_package** ». Notez que le nom de votre application peut différer du nom donné dans votre package name : par exemple le package name de l'application de VoIP : « Google Hangout » est « **com.google.android.talk** » car Hangout est en fait une nouvelle version du projet Google Talk.

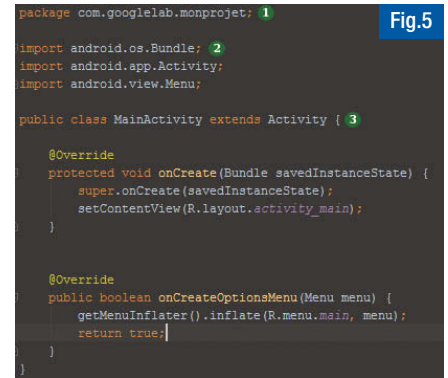
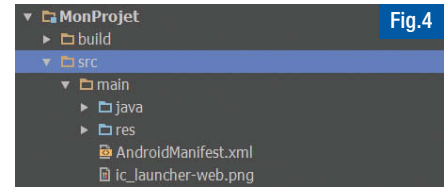
Classes java

Comme vous le savez peut-être, une application Android est principalement codée en Java. Le Java, comme la majorité des langages les plus utilisés actuellement est un Langage Orienté Objet, ce qui implique l'utilisation de « classes ». Une classe dans une application Android est un ensemble de propriétés et de méthodes. Vous aurez en général un fichier par classe pour des raisons de lisibilité du code. Dans votre classe, la structure de votre code doit suivre un schéma très précis, sans quoi vous ne pourrez pas compiler votre application : **Fig.5**.

► La première ligne indique dans quel package se trouve votre classe (autrement dit le chemin relatif par rapport au dossier source). Si votre classe se trouve dans le dossier source lui-même, cette ligne restera vide.

► Si vous souhaitez utiliser une des fonctionnalités provenant d'une classe décrite dans un autre package, vous devrez alors « l'importer » en utilisant la syntaxe : **import package.class** ; (exemple : **import android.app.Activity** ;).

► Finalement, votre classe est déclarée. Ici par exemple, la classe est nommée **MainActivity** et étend la classe **Activity**.



Dossier « res » (Ressources applicatives : écrans, images, textes, ...)

Le fichier R.Java

Il s'agit du fichier qui fera le lien entre tous les fichiers XML présents dans le dossier « **res** » et le code applicatif Java, il sera automatiquement généré si vous utilisez un IDE Android.

Layout

Un layout, qu'on pourrait traduire par « disposition » en français est un fichier XML de mise en page pour Android. Dans ce fichier, vous ajouterez les différents composants graphiques de votre activité, et pourrez ensuite les appeler dans votre code en Java.

Langues

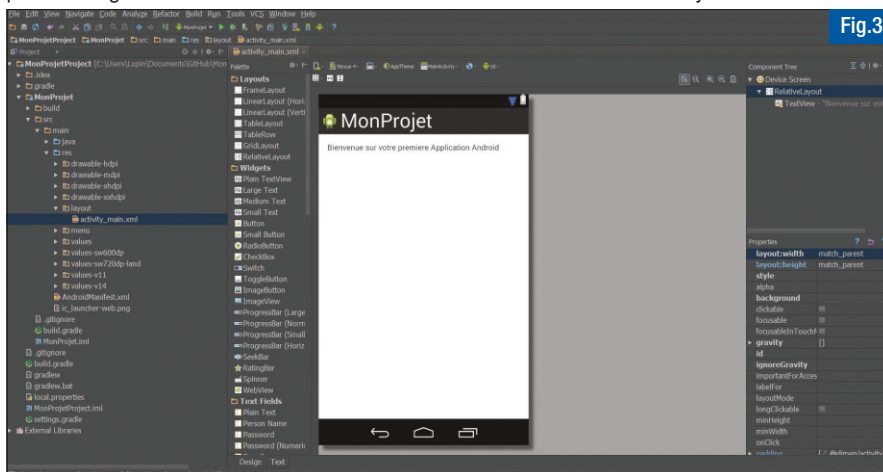
La gestion des langues sur Android se fait d'une manière assez simple, chaque dossier de ressources est localisable en ajoutant une extension relative à la langue de son contenu après son nom. Par exemple, on pourra créer un dossier « **values-fr** » pour proposer les textes traduits aux utilisateurs utilisant le système en français (dans un fichier **strings.xml** par convention).

Dossier « assets » (Ressources « brutes »)

Le dossier « **assets** » sert à inclure dans votre projet des ressources de tous types que vous pourrez utiliser dans votre code. Bien que les ressources telles que les images ou le texte aient leur propre répertoire, celui-ci peut servir dans des cas particuliers : l'utilisation d'un fichier **Json** pour stocker certaines données, par exemple.

Dossier « gen » (Code généré par l'IDE)

Le dossier « **gen** » est le dossier où l'IDE va compiler automatiquement les ressources de votre





application. Le contenu de ce dossier ne doit en aucun cas être modifié à la main.

LES COMPOSANTS GRAPHIQUES ANDROID

Les composants graphiques, aussi appelés Widgets (à ne pas confondre avec les Widgets que vous pouvez placer sur votre écran d'accueil) sont les éléments qui constituent la partie graphique de votre Application.

Les Layouts

Les **Layouts** sont des composants (graphiquement invisibles) qui peuvent contenir d'autres composants graphique. Les deux Layouts les plus courants sont :

► **RelativeLayout** : il permet de contenir des composants de façon relative : ceux-ci sont donc positionnés les uns par rapport aux autres. Un élément peut également être aligné sur un coté du RelativeLayout ou au centre par exemple. Ce Layout permet une grande liberté du positionnement des composants, mais attention, une mauvaise utilisation de ce Layout peut entraîner une superposition des composants, notamment sur les plus petits périphériques. Il est donc dans certains cas préférable d'utiliser des LinearLayout, quitte à les imbriquer Fig.6. [relativelayout.png]

► **LinearLayout** : ce Layout permet de placer des composants l'un à côté de l'autre ou bien l'un en dessous de l'autre, suivant l'orientation choisie : horizontale ou verticale Fig.7.

Astuce : Sachez qu'un Layout peut contenir un autre Layout et ainsi de suite. Il faut tout de

même éviter de les imbriquer de façon exagérée et inutile pour des raisons de performances.

Les composants graphiques simples

L'élément que vous utiliserez sûrement le plus est le **TextView**. Comme son nom l'indique, c'est un composant graphique vous permettant d'afficher du texte dans votre application.

Vous pouvez modifier de nombreux paramètres dont la taille ou la couleur du texte.

Un autre composant essentiel est l'**ImageView**, qui permet d'afficher une image. De nombreux paramètres sont disponibles dont la largeur et la hauteur.

Le **Button** est également un composant que vous utiliserez souvent, il vous permettra d'afficher un bouton à l'écran et d'apporter du comportement à vos applications Fig.8.

Des composants plus complexes

Android propose également des composants graphiques plus complexes comme les **List-View**. Une **ListView** permet, comme son nom l'indique, de créer une liste. Cette liste peut être composée d'un TextView (du texte) ou bien de plusieurs composants (des Layouts, des ImageView...) Fig.9.

Le **GridView** est un composant qui affiche les Views qu'il contient de manière bi-dimensionnelle Fig.10.

AVD MANAGER

Android Virtual Devices Manager, ou AVD Manager, est un utilitaire intégré dans le SDK Android permettant de créer et lancer des machines virtuelles, émulant des périphériques Android. Cet outil permet de tester son application sous de

nombreuses configurations, que ce soit au niveau de la taille de l'écran, de sa résolution, de la version de l'API Android, etc...

Gestion des API Android

Lors de la création d'une machine virtuelle Android, que l'on appellera un **AVD**, il est possible de choisir la version de l'API Android. Pour cela, il faut tout d'abord télécharger les images Android qui vous intéressent via le SDK Manager Fig.11. Par exemple, si vous souhaitez utiliser l'API 18 (qui correspond à la version 4.3 d'Android), lancez le SDK Manager puis cochez la case « Android 4.3 (API 18) ».

Une fois que vous avez coché tout ce dont vous aviez besoin, cliquez sur « Install packages » en bas à droite, et suivez l'assistant d'installation.

Création d'un AVD pour lancer nos applications

Pour créer un AVD, lancez l'AVD Manager. Cliquez ensuite sur « New » en haut à droite de la fenêtre ; vous devez ensuite choisir un nom, un périphérique à émuler, la version de l'API Android, le type de processeur (ARM ou x86 si vous l'avez téléchargé).

Vous pouvez aussi choisir de cocher « Hardware keyboard present » afin de pouvoir écrire avec votre clavier physique dans la machine virtuelle, ce qui est le plus pratique contrairement au clavier proposé par la machine virtuelle.

Si vous cochez « Display a skin with hardware controls », les boutons physiques du périphérique seront émulés sous la forme de boutons cliquables : cette option est conseillée.

Il est également possible d'émuler une caméra avec votre webcam. Vous pouvez ensuite choisir la quantité de RAM et de stockage interne et externe, si vous ne savez pas quoi choisir, laissez les valeurs par défaut.

Enfin, vous pouvez cocher soit « Snapshot » (mettre en veille la machine virtuelle au lieu de l'éteindre, ce qui entraînera un prochain démarrage plus rapide) ou « Use Host GPU » (permet à la machine virtuelle d'utiliser directement votre carte graphique, les animations sont alors beaucoup plus fluides).



Fig.6



Fig.7

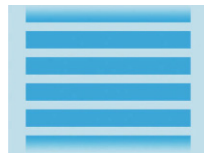


Fig.9

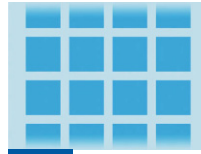
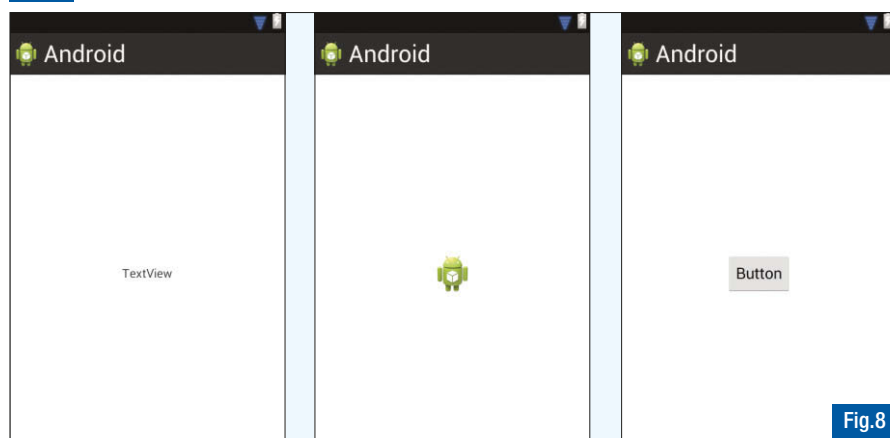


Fig.10



Un TextView

Un ImageView

Un Button

Fig.8

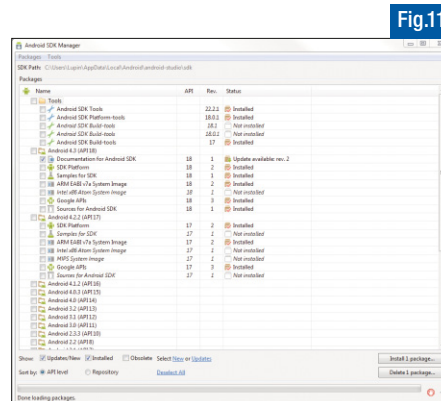


Fig.11

Une fois tout rempli, cliquez sur « OK ». Pour utiliser votre AVD, sélectionnez-le dans la liste, puis cliquez sur « Start... » à droite de l'AVD Manager. Une fois votre AVD correctement démarré (quand vous avez déverrouillé l'écran et que vous serez sur l'écran d'accueil Android), vous pouvez lancer votre Application grâce à votre IDE en choisissant l'AVD comme périphérique, votre application sera alors lancée sur la machine virtuelle.

MA PREMIÈRE APPLICATION « MON ASSISTANT »

Cycle de vie d'une Activité

Une Activité, dans Android, représente tout ce que l'on voit à l'écran dans une application. Android gère lui-même le cycle de vie d'une Activité. Voici le cycle complet : Fig.14.

La méthode à retenir est **onCreate()** : elle est appelée la première fois lorsque l'Activité est créée, c'est ici qu'il faut assigner chaque donnée statique (View, List...). Un appel à la méthode **setContentView()** permet de définir l'affichage à partir d'un fichier XML.

Mon application !

Nous allons maintenant rentrer dans le vif du sujet : la création de notre propre application Android. Nous allons réaliser une simple application permettant, à l'aide d'un ImageButton, d'allumer le flash du téléphone. Un Button sera utilisé afin de créer une nouvelle Activité qui listera toutes les applications qui sont installées sur votre appareil. Pour commencer, il faut créer un nouveau projet. Sur Eclipse, faites : File/New/New Android Application Project.

Indiquez le nom de votre application, un nom de projet et un package correct. Cliquez sur « Suivant ». Cliquez encore sur « Suivant », nous ne voulons pas créer une bibliothèque mais bien une application.

Sélectionnez une icône pour votre application (sachant que vous ne voudriez sûrement pas garder les icônes de base) et cliquez sur « Suivant ». Sélectionnez enfin une *Blank Activity*, donnez le nom que vous souhaitez à l'Activité principale et à son fichier xml correspondant.

Ma lampe torche

Commençons par implémenter la lampe torche. Le principe est le suivant : lorsque j'appuie sur un **ImageButton**, je veux que le flash de l'appareil photo de mon téléphone s'allume.

C'est pourquoi on va rajouter un **ImageButton** dans notre *activity_main.xml* : voir code source sur le site web

Au niveau du code Java, il faut récupérer l'instance de l'**ImageButton** et lui dire, lorsque l'on appuie dessus, d'allumer le flash du téléphone : voir code source sur le site web.

Petites explications : Pour répondre aux événements en tous genres (Touch, Click, LongClick, Drag...) sur l'**ImageButton**, il faut assigner à celui-ci l'interface correspondant à cet événement. Ici, on veut récupérer l'événement **Click** : on va donc assigner l'interface **onClickListener** à l'**ImageButton**.

Dans la méthode **onClick(View v)**, on écrit le code qui sera exécuté lorsque l'on appuiera sur ce composant. Dans notre cas, nous vérifierons si le flash est allumé. Si c'est le cas, il s'éteindra et sinon, il s'allumera tout simplement. Si on veut rajouter un peu de debug, la classe **Log** est notre amie ! Pour s'en servir, utilisez la méthode **Log.d** (debug simple), **Log.i** (debug à titre informatif) ou **Log.e** (debug à titre d'erreur).

Mes applications installées

Pour visualiser nos applications, nous allons utiliser une **ListView**. C'est un composant graphique plus compliqué que l'**ImageButton** car il nécessite de lui attribuer un **Adapter**. L'Adapter est l'outil qui permettra de créer chaque ligne de notre **ListView**. Il est possible de créer des Adapters totalement personnalisés. Cependant, dans notre cas, nous utiliserons la classe **ArrayAdapter** qui permet de créer des Adapters déjà configurés.

Tout d'abord, rajoutons un Button dans notre fichier *activity_main.xml* qui servira de pont entre notre activité principale et la nouvelle qui contiendra notre **ListView** : voir code source sur le site web.

Ensuite, dans le Java, il faut que ce nou-

veau Button puisse lancer l'Activité des applications installées.

Pour ce faire, nous utiliserons un **Intent**. Un Intent représente l'intention de faire quelque chose, par exemple créer de nouvelles Activités, Services... Notre nouvelle Activité s'appellera **ListViewActivity** ; il faut donc créer une nouvelle classe de ce nom. Voir code source sur le site web.

Dans notre nouvelle classe, il faut récupérer l'instance de notre **ListView** et lui assigner un **ArrayAdapter**. Les **ArrayAdapters** ont besoin de plusieurs choses : l'id du layout correspondant au Layout que l'on va utiliser (il ne doit contenir qu'un seul et unique **TextView**), le Context courant, ici l'Activité même, et une **List** de Strings qui composera notre **ListView** (le **TextView** de chaque ligne prendra le texte de sa String correspondant).

Nous voulons récupérer la liste des applications installées sur notre téléphone. Afin d'y arriver, nous utilisons une classe qui le fait pour vous : **PackageManager**. À partir de la **List** d'applications que nous a retourné le **PackageManager**, il suffit de parcourir celle-ci afin d'y récupérer le nom de chaque application et de l'ajouter à notre **List** de Strings finale. voir code source sur le site web.

Et voilà votre première application ! Elle pourra même vous servir au quotidien ! Il ne tient qu'à vous maintenant de l'améliorer en ajoutant quelques améliorations !

CONSEILS

Votre développement Android se fait en Java, essayez donc de tirer parti du langage !

Oracle propose des tutoriels si vous avez des hésitations :

<http://docs.oracle.com/javase/tutorial/java/>

La charte graphique est importante : c'est elle qui va différencier votre application des autres. Une application fonctionnelle mais laide ne restera pas installée longtemps sur un téléphone.

Google propose donc un guide pour designer vos applications :

<http://developer.android.com/design/>

La documentation officielle contient des informations sur la majorité des classes et des méthodes existantes dans le code d'Android. Elle deviendra vite un outil indispensable pour les développeurs en herbe :

<http://developer.android.com/guide/>

Michael Ohayon (Epitech Promo 2016),
Denis Mondon (Epitech Promo 2016),
Jonathan Picques (Epitech Promo 2016),
Victor Leduc (Epitech Promo 2016),
Vincent Nèel (Epitech Promo 2016),
Jérémie Amsellem (Epitech Promo 2017)

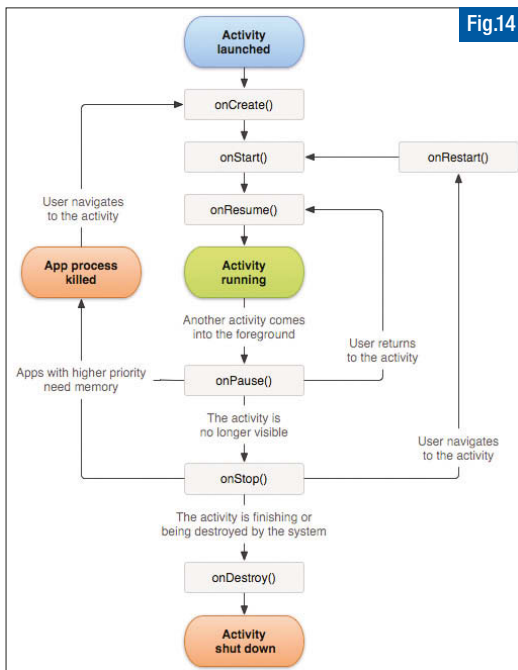


Fig.14



Monétiser son application Android avec les achats in-app

La monétisation des applications mobiles est une problématique centrale impactant tout un chacun, des studios de développement aux développeurs indépendants. Les développeurs Android n'échappent pas à la règle. Parmi les différents modèles de monétisation des applications, le modèle freemium qui propose aux utilisateurs l'achat de contenu in-app se révèle de loin le plus performant sur le Google Play Store à l'heure actuelle. Dans cet article, nous nous proposons de détailler la mise en oeuvre des achats in-app au sein des applications Android.

Problématique centrale du développement mobile, la question de la monétisation doit être posée dès la conception. En effet, le choix d'un modèle freemium ou payant comporte des impacts qu'il est essentiel de considérer au plus tôt. Reconnu comme la solution de monétisation la plus efficace sur l'App Store d'Apple, l'achat in-app a fait son apparition sur le Google Play Store début 2011. Limité aux seuls achats de contenus dans un premier temps, il s'est étendu mi-2012 pour proposer le concept de souscriptions permettant aux éditeurs de presse notamment, de proposer un abonnement à une édition numérique avec un prélèvement automatique du compte utilisateur. Dès lors, ce modèle de monétisation s'est rapidement répandu au point de devenir en 2013 le modèle choisi par plus de 80% des applications les plus rentables du Play Store ! Bien que ce modèle ait fait ses preuves, son implémentation sous Android laissait à désirer du fait d'une certaine complexité, tant au niveau du développeur, que de l'utilisateur, avec un workflow d'achats loin d'être optimal. Partant de ce constat, les équipes de Google ont proposé en cours d'année 2013 la version 3 de l'API d'achats in-app. Cette dernière simplifie la mise en oeuvre des achats in-app pour le développeur et améliore le workflow utilisateur. Dans ce qui suit, nous allons nous appuyer sur cette version pour réaliser une application proposant l'achat de contenus in-app. Volontairement simple, l'application proposera l'achat de clicks. Une fois un click acheté, l'utilisateur pourra l'utiliser en cliquant sur un bouton prévu à cet effet. Le click réalisé, l'achat sera de nouveau autorisé. Cet exemple se révélera suffisant pour montrer les problématiques essentielles rencontrées lors de la mise en oeuvre des achats intégrés.

Installation

Avant toute chose, il est nécessaire d'installer la dernière version du SDK Android disponible ici : <http://developer.android.com/sdk/>. Ensuite, il faut lancer le SDK Manager pour récupérer la bibliothèque «Google Play Billing Library» se trouvant dans les extras. Une fois téléchargée, la bibliothèque est située à l'emplacement suivant : `<sdk path>/extras/google/play_billing/`. Pour la réalisation de l'application, nous nous appuyerons sur Eclipse muni du plugin ADT de Google dont l'installation se fait via l'update site suivant : <https://dl-ssl.google.com/android/eclipse/>. Enfin, un compte développeur Google est nécessaire pour tester le paiement in-app. Les frais d'ouverture se montent à 25 dollars pour une utilisation illimitée dans le temps avec la possibilité de publier autant d'applications que souhaitées.

Initialisation du projet

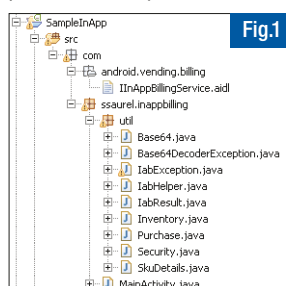
L'environnement de travail en place, il est temps de créer le projet de l'application. Pour cela, il suffit d'utiliser la fonction de création d'un nouveau projet de type Application Android sous Eclipse. L'application ayant vocation à être uploadée sur le Play Store pour les tests, il est

important de renseigner un nom de package unique et donc propre à chaque développeur. Dans mon cas, le package choisi est donc : `com.ssaurel.inapppbilling`. Ensuite, il est important de choisir la dernière version du SDK comme cible pour l'application. En outre, nous faisons générer l'unique activité de l'application ainsi que son layout `main.xml`. L'application utilisant le paiement in-app, elle doit déclarer la permission BILLING au sein du manifest de la sorte :

```
<uses-permission android:name="com.android.vending.BILLING" />
```

Afin de faciliter le développement d'applications utilisant l'API d'achats in-app, Google propose un projet d'exemple complet au sein du dossier d'extras récupéré précédemment. Ce projet contient notamment un ensemble de classes utilitaires facilitant les interactions avec la partie serveur de l'API Google Play Billing. Le contenu de ce projet est disponible via le chemin suivant : `<sdk path>/extras/google/play_billing/samples/TrivialDrive`. La première étape consiste à récupérer au sein de ce projet l'ensemble des classes du package `com.example.android.trivialdrivesample.util` et de les ajouter dans un package utile au sein de notre projet `SampleInApp`.

Le projet est alors quasiment initialisé, si ce n'est qu'il demeure une erreur se rapportant à la classe `labHelper` et à l'import de l'objet `com.android.vending.billing.IInAppBillingService`. Cet objet correspond au fichier `IInAppBillingService.aidl`; il représente le contrat d'interface défini pour communiquer avec la version 3 du service Google Play In-app



Structure du projet

Billing. Au format AIDL (Android Interface Definition Language), cette interface est utilisée par les classes utilitaires pour réaliser des appels de méthodes via IPC. Le fichier `IInAppBillingService.aidl` se récupère au sein du package `com.android.vending.billing` dans le projet `TrivialDrive`, et se copie au sein d'un package de même nom dans notre projet **Fig.1**.

Interface graphique

Défini au sein du fichier `main.xml`, le layout de l'activité principale de l'application affiche 2 boutons. Le premier correspond à un click qui sera proposé à l'achat, et le second servira à acheter un nouveau click via l'API d'achats in-app de Google. Enfin, un contrôle de type texte indique à l'utilisateur s'il doit acheter un nouveau click. Le contenu du layout est le suivant :

```
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent">
```

```

android:layout_height=>match_parent>
android:background="#EEEEEE">

<Button
    android:id=>@+id/buyBtn>
    android:layout_width=>wrap_content>
    android:layout_height=>wrap_content>
    android:layout_centerHorizontal=>true>
    android:layout_centerVertical=>true>
    android:onClick=>buyClick>
    android:text=>@string/buy>/>

<TextView
    android:id=>@+id/buyTxt>
    android:layout_width=>wrap_content>
    android:layout_height=>wrap_content>
    android:layout_above=>@+id/buyBtn>
    android:layout_centerHorizontal=>true>
    android:layout_marginBottom=>65dp>
    android:text=>@string/buyclick>
    android:visibility=>gone>/>

<Button
    android:id=>@+id/clickBtn>
    android:layout_width=>wrap_content>
    android:layout_height=>wrap_content>
    android:layout_above=>@+id/buyTxt>
    android:layout_centerHorizontal=>true>
    android:layout_marginBottom=>130dp>
    android:onClick=>buttonClick>
    android:text=>@string/clickme>/>

</RelativeLayout>

```

On note également la définition des handlers d'évènements click sur les 2 boutons de l'interface graphique via l'utilisation de l'attribut `android:onClick`.

Activité principale

L'activité de l'application doit gérer l'achat de clicks, l'utilisation des clicks achetés, et, enfin, elle doit afficher un message indiquant à l'utilisateur s'il peut acheter un click ou non :

```

public class MainActivity extends Activity {
    private Button buyBtn;
    private Button clickBtn;
    private TextView buyTxt;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.main);
        buyBtn = (Button) findViewById(R.id.buyBtn);
        clickBtn = (Button) findViewById(R.id.clickBtn);
        clickBtn.setEnabled(false);
        buyTxt = (TextView) findViewById(R.id.buyTxt);
        buyTxt.setVisibility(View.VISIBLE);

        public void buttonClick(View v) {

```

```

        clickBtn.setEnabled(false);
        buyTxt.setVisibility(View.VISIBLE);
        buyBtn.setEnabled(true);
    }
    // ...
}

```

L'application précise à l'utilisateur qu'il peut acheter un click à son lancement. La méthode `buttonClick` permet d'utiliser un click et de rendre à nouveau possible l'achat in-app d'un click. D'autre part, le handler `buyClick` n'est pas encore entièrement implémenté à ce stade de l'application. Cependant, une première exécution de l'application donne déjà un aperçu visuel **Fig.2**.

Configuration du Google Play Billing

Les fondements de l'application posés, il est temps de s'intéresser aux échanges avec le serveur Google Play Billing. Chaque développeur d'application échangeant avec ce serveur, doit s'identifier via une clé unique. La seule possibilité pour obtenir une telle clé étant de s'enregistrer en tant que développeur à l'adresse suivante : <http://play.google.com/apps/publish>.

Une fois les frais réglés, il faut accéder à la console développeur du Play Store et aller dans la partie Settings accessible depuis le menu de gauche. Il devient alors possible de créer le compte Google Waller Merchant nécessaire à l'utilisation des achats in-app. L'étape suivante est l'obtention de la clé de licence unique. Pour ce faire, une nouvelle application doit être créée dans la console développeur. A ce stade, il n'est pas nécessaire d'uploader l'APK de l'application. Nous pouvons donc aller directement dans la partie «Services & APIs» du menu de gauche. Cette partie affiche la clé publique RSA en base 64 pour l'application qui sera à utiliser lors des échanges avec le serveur **Fig.3**.

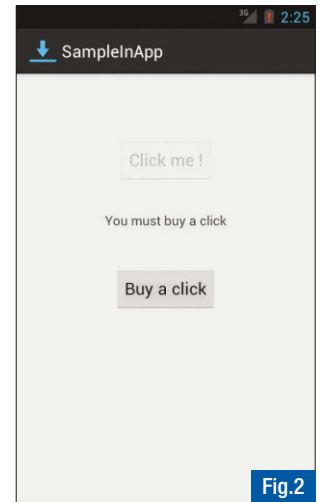


Fig.2

Ecran principal de l'application

Première connexion

Munis de cette clé de licence unique, nous allons pouvoir réaliser une première connexion entre notre application et le serveur Google Play Billing. Il faut ainsi ajouter un appel à la méthode `startSetup` de l'objet utilitaire `labHelper` lors de la création de l'activité :

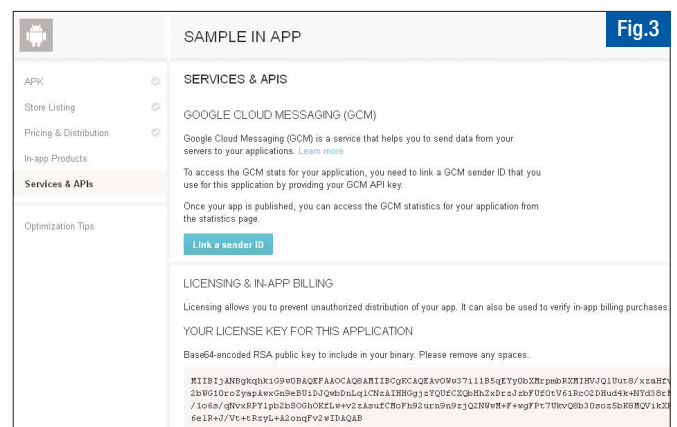


Fig.3

Affichage de la clé publique en base 64



```
iabHelper = new IabHelper(this, PUBLIC_KEY);
iabHelper.startSetup(new IabHelper.OnIabSetupFinishedListener() {
    @Override
    public void onIabSetupFinished(IabResult result) {
        if (!result.isSuccess()) {
            Log.i(TAG, «IAP setup failed : « + result);
        } else {
            Log.i(TAG, «IAP setup OK»);
        }
    }
});
```

Pour tester cette première connexion, il est impératif de lancer l'application sur un terminal physique puisque les échanges avec le Google Play Billing ne peuvent être testés sur un émulateur. Si tout a été configuré convenablement, le message de log indiquant que le setup a réussi devrait être affiché dans la console.

Réaliser un achat

La connexion avec le serveur Google Play Billing effective, nous pouvons lancer l'achat in-app d'un produit. Cet achat est lancé depuis la méthode `buyClick` de l'activité principale. Outre l'appel à la méthode lançant le processus d'achat in-app, il est également nécessaire d'implémenter la méthode `onActivityResult` au sein de l'activité. Celle-ci sert de listener permettant de savoir quand le processus d'achat est terminé. Ce dernier étant lancé via la méthode `launchPurchaseFlow` de l'objet `IabHelper` qui attend notamment l'identifiant SKU de l'item que l'utilisateur souhaite acheter. Pour les phases de test, Google fournit l'identifiant SKU suivant : *android.test.purchased*. Cet item résulte toujours en un achat réalisé avec succès. Le code à rajouter à l'activité est le suivant :

```
public static final String ITEM_SKU = «android.test.purchased»;
public static final int REQUEST_CODE = 10001;

public void buyClick(View v) {
    iabHelper.launchPurchaseFlow(this, ITEM_SKU, REQUEST_CODE,
        mPurchaseFinishedListener, «»);
}
```

On notera également le `REQUEST_CODE` identifiant l'achat en cours. Le listener `mPurchaseFinishedListener`, détaillé un peu plus tard, est appelé lorsque l'achat est complété. Enfin, le dernier paramètre est un payload pouvant être utilisé comme une sécurité supplémentaire par le développeur mais qui reste vide ici.

Le processus d'achat complété se termine par un appel à `onActivityResult` dans l'activité appelante. Dans cette méthode, il faut vérifier si le retour provient d'un achat in-app ou non. Dans le cas où il s'agit du retour d'un achat, on ne fait rien puisque l'objet de type `OnIabPurchaseFinishedListener`, passé en paramètre de `launchPurchaseFlow`, gère le résultat de l'achat. S'il ne s'agit pas du retour d'un achat, on laisse le travail normal de la méthode s'effectuer :

```
@Override
protected void onActivityResult(int requestCode, int resultCode, Intent data) {
    if (!iabHelper.handleActivityResult(requestCode, resultCode, data)) {
        super.onActivityResult(requestCode, resultCode, data);
    }
}
```

Gestion du retour d'un achat

La gestion du retour de l'achat est implémentée dans l'objet `OnIabPurchaseFinishedListener` défini précédemment. En son sein, on teste si la tentative d'achat a échoué. Si oui, on traite l'erreur. Dans le cas où un achat a bien été effectué pour l'item considéré, on consomme ce dernier et on effectue les mises à jour nécessaires au niveau de l'IHM. Ainsi, on désactive le bouton d'achat et on rend invisible le texte associé :

```
IabHelper.OnIabPurchaseFinishedListener mPurchaseFinishedListener = new IabHelper.OnIabPurchaseFinishedListener() {
    public void onIabPurchaseFinished(IabResult result, Purchase purchase) {
        if (result.isFailure()) {
            // Gérer erreur
            return;
        } else if (purchase.getSku().equals(ITEM_SKU)) {
            consumeItem();
            buyBtn.setEnabled(false);
            buyTxt.setVisibility(View.GONE);
        }
    }
};
```

Dans ce code, on note la présence de la méthode `consumeItem`, non définie encore, qui permet de consommer l'item. Concrètement, il s'agit de spécifier au serveur Google Play Billing que l'item acheté a bien été consommé par l'utilisateur et qu'il faut donc le retirer de la liste des items en attente côté serveur. Afin d'éviter des problèmes de sécurité, Google conseille de consommer l'item avant même la mise à disposition à l'utilisateur. C'est ce que nous faisons ici dans la méthode `consumeItem` en invoquant la méthode `queryInventoryAsync` sur l'objet utilitaire `IabHelper`.

Réalisé de manière asynchrone au sein d'un thread séparé pour ne pas bloquer l'UI Thread, cet appel liste les items déjà achetés par l'utilisateur. Ici, un seul item pouvant être acheté, il suffit de consommer l'item si le retour de l'appel n'est pas en erreur :

```
public void consumeItem() {
    iabHelper.queryInventoryAsync(mReceivedInventoryListener);
}

IabHelper.QueryInventoryFinishedListener mReceivedInventoryListener = new IabHelper.QueryInventoryFinishedListener() {
    public void onQueryInventoryFinished(IabResult result, Inventory inventory) {
        if (result.isFailure()) {
            // Gérer erreur
        } else {
            iabHelper.consumeAsync(inventory.getPurchase(ITEM_SKU), mConsumeFinishedListener);
        }
    }
};
```

Pour terminer la gestion du retour de l'achat, il faut implémenter le listener `OnConsumeFinishedListener` permettant de vérifier si l'item `ITEM_SKU` a bien été consommé :

```
IabHelper.OnConsumeFinishedListener mConsumeFinishedListener = new IabHelper.OnConsumeFinishedListener() {
```

```
public void onConsumeFinished(Purchase purchase, IabResult
result) {
    if (result.isSuccess()) {
        clickBtn.setEnabled(true);
    } else {
        // Gérer erreur
    }
}
};
```

Lorsque la consommation de l'item a réussi, nous activons le bouton de click offrant la possibilité à l'utilisateur d'accéder à son achat. Comme nous avons pu le voir, le processus d'achat in-app est facilité par le recours à l'objet `IabHelper` proposé par Google. Lors de la destruction d'une activité, il faut donc bien penser à libérer l'objet.

Test statique

Nous réalisons en premier lieu un test statique en laissant à l'achat le produit de test `ITEM_SKU`. Une fois l'application lancée sur un terminal physique, un click sur le bouton d'achat produit bien le résultat escompté avec le démarrage du processus d'achat [Fig.4](#). Une fois l'achat confirmé, le résultat du paiement apparaît [Fig.5](#).

L'interface graphique est bien mise à jour et le click qui vient d'être acheté est désormais accessible à l'utilisateur [Fig.6](#).

Test dynamique

Le test statique ayant fonctionné, nous passons aux tests en conditions réelles, en ajoutant un produit à l'application via la console développeur. Avant d'aller plus loin, il est temps d'uploader sur le Play Store l'APK de l'application généré via l'outil d'export d'Eclipse. L'APK uploadé, nous ajoutons un nouveau produit à l'application [Fig.7](#) via le menu «In-App Products».

De type managé, l'item click bouton a un identifiant unique qui sera utilisé lors de la requête d'achat depuis l'application. L'assistant de création d'item impose également de définir un nom, une description, ainsi qu'un prix d'achat. La console développeur permet enfin de lister l'ensemble des items disponibles [Fig.8](#).

Dans l'application, il suffit de remplacer la valeur de la variable `ITEM_SKU` par l'id de l'item nouvellement créé :


```
public static final String ITEM_SKU = «com.ssaurel.items.
btnclick»;
```

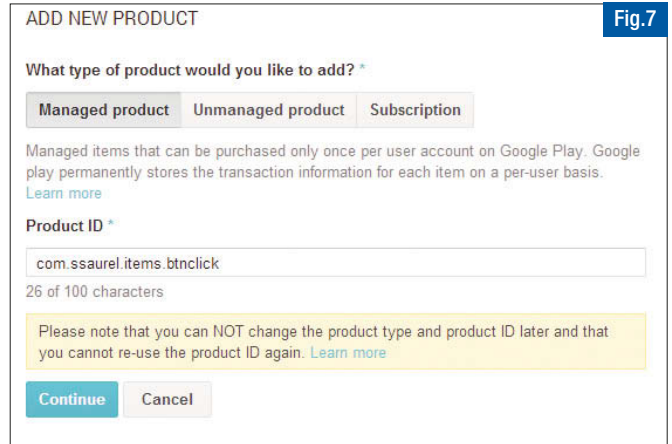
Enfin, au niveau du Play Store, il faut définir des comptes de tests permettant à des utilisateurs de tester les achats in-app de l'application

avant sa publication. Il est bon de noter que le compte du développeur de l'application ne peut être ajouté à la liste. L'ajout des utilisateurs de test se fait via le menu «Settings» de la console développeur dans la zone «Gmail accounts with testing access». Il ne reste plus désormais qu'à uploader l'APK sur les téléphones associés aux comptes de test. Le click sur le bouton d'achat valide alors le fonctionnement global de l'achat in-app mis en place dans l'application [Fig.9](#).

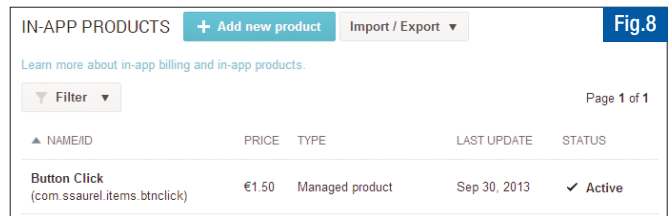
Conclusion

Associé au modèle freemium, l'achat de contenus in-app se révèle être le modèle de monétisation le plus rentable du Google Play Store. Tout l'enjeu du succès résidant dans la capacité des développeurs à trouver des produits suffisamment attrayants à vendre au sein de leurs applications. Le service d'achat in-app Google Play Billing mis à disposition par Google se révèle d'une grande puissance tout en restant simple d'accès comme nous avons pu le voir en mettant en oeuvre la version 3 de l'API. En outre, les classes utilitaires proposées par Google aident grandement cette facilité d'utilisation. Légèrement abordée dans cet article, la sécurité des achats in-app est garantie par l'architecture mise en place par Google à condition de prendre en compte l'ensemble des préconisations de sécurité du service côté application et côté serveur.

 Sylvain Saurel – Ingénieur d'Etudes Java / Java EE
sylvain.saurel@gmail.com

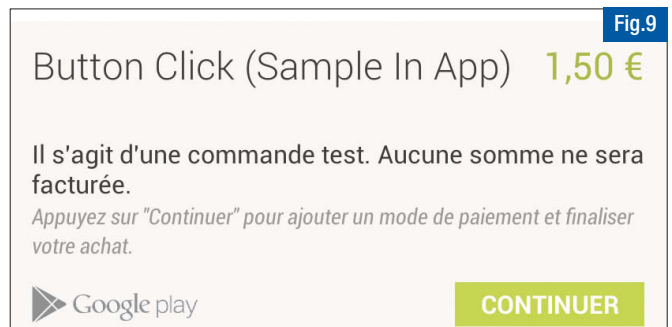


Item click bouton

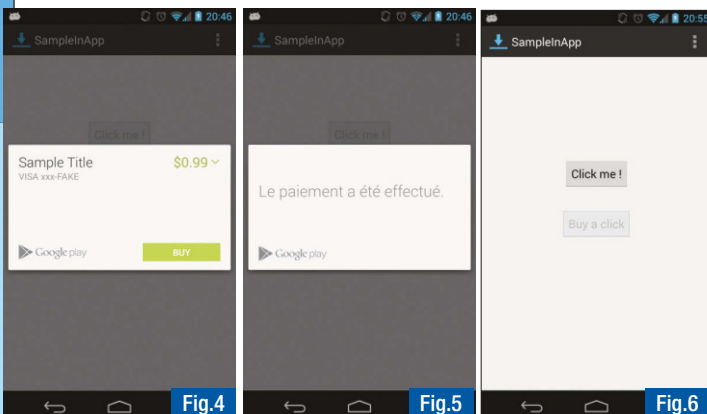


NAME/ID	PRICE	TYPE	LAST UPDATE	STATUS
Button Click (com.ssaurel.items.btnclick)	€1.50	Managed product	Sep 30, 2013	✓ Active

Liste des items



Test dynamique d'achat



Démarrage processus achat

Achat terminé

Click acheté

Souriez avec Nokia Imaging SDK for Windows Phone

Si ces derniers temps vous suivez un petit peu l'actualité des smartphones, vous avez sûrement remarqué que Nokia, constructeur phare de Windows Phone, mise beaucoup sur la photo. Avec la sortie du Lumia 1020, il devient difficile de dire s'il s'agit d'un smartphone avec un très bon appareil photo ou d'un appareil photo avec lequel il est possible de téléphoner. La photographie devient alors une des expériences primordiales de votre téléphone.

Afin de faciliter la création d'applications manipulant des images, Nokia propose depuis quelque mois un kit de développement nommé « Nokia Imaging SDK ». Celui-ci propose de nombreuses fonctionnalités intéressantes :

- ▮ Décodage partiel d'image : plutôt que de décoder une image entière, il peut ne décoder qu'une partie d'un JPEG. Cela s'avère très pratique lorsque l'on manipule des images en haute résolution (41 MPixels pour le Lumia 1020)
- ▮ Une galerie d'effets : avec la popularité d'Instagram, il est très à la mode d'appliquer un filtre sur une photo. Nokia vous propose une collection de 50 filtres à appliquer ou combiner.
- ▮ Manipulation d'images : la rotation, le redimensionnement, la découpe sont facilités avec une gestion de l'historique des opérations appliquées.

Ce SDK est disponible sous la forme d'une librairie WinPRT : vous pouvez donc l'utiliser depuis du code managé (C#/VB) ou natif (C++) dans n'importe quelle application Windows Phone 8. Il n'est pas disponible pour Windows Phone 7 à l'heure actuelle.

Ajoutons qu'il est bon de souligner que vous pouvez utiliser ce kit de Développement gratuitement, mais, surtout, sur n'importe quel Windows Phone et non pas uniquement les Nokia.

Comment l'utiliser dans un projet ?

Nokia propose plusieurs façons d'intégrer ce SDK dans votre projet :

- ▮ En téléchargeant un fichier MSI qui l'installera sur votre machine de développement. Pour cela, il suffit de vous rendre sur le site « <http://developer.nokia.com> » et de suivre le lien de téléchargement sur SDK.
 - ▮ En utilisant le gestionnaire de paquet Nuget depuis votre solution Visual Studio.
- C'est cette dernière solution que nous préférons car elle nous permettra de migrer nos applications une à une dès que Nokia proposera une nouvelle version de son SDK plutôt que de tout devoir faire à la fois. Voici les étapes à réaliser sur votre projet :
- ▮ Faire un clic-droit sur votre projet Windows Phone,
 - ▮ Choisir « Manage Nuget Package »,
 - ▮ Choisissez « online » ainsi que « include pre-release » car il s'agit d'une version beta,
 - ▮ Recherchez « Nokia »,
 - ▮ Choisissez « Nokia Imaging SDK » et cliquez sur « Install »,
 - ▮ C'est tout !

Fig.1

Une première manipulation d'image

Faisons maintenant notre premier essai avec ce SDK : générons une vignette de 100x100 pixels que nous allons retourner et sauvegarder pour l'utilisateur. Voici le code :

```
var task = new Microsoft.Phone.Tasks.CameraCaptureTask();
task.Show();
task.Completed += async (sender, e) =>
{
    if (e.TaskResult != TaskResult.OK) return;
    var photoBuffer = new byte[e.ChosenPhoto.Length];
    await e.ChosenPhoto.ReadAsync(photoBuffer, 0, photoBuffer.Length);
    using (var session = new EditingSession(photoBuffer.AsBuffer()))
    {
        //Création du filtre de rotation
        var rotateFilter = FilterFactory
            .CreateFreeRotationFilter(180, RotationResizeMode.Fit
            Inside);
        session.AddFilter(rotateFilter);

        //Encodage du jpeg
        var monJpeg = await session.RenderToJpegAsync(
            new Size(100, 100), OutputOption.PreserveAspectRatio);

        //Sauvegarde de l'image
        using (var ml = new MediaLibrary())
        {
            ml.SavePictureToCameraRoll(
                «mon image retournée.jpgg», monJpeg.ToArray());
        }
    }
};
```

Et voilà ! Ce qui aurait autrefois demandé de décoder une image, d'interpréter le format de ses bytes, de les manipuler et de les ré-encoder ne nécessite plus que quelques lignes de code facile à comprendre pour les néophytes de l'encodage d'image. Le code pour demander une image utilise une CameraCaptureTask classique, nous fournissons ensuite les bytes à un objet nommé « EditingSession » auquel nous ajoutons un filtre de rotation. Nous encodons alors l'image en Jpeg de taille 100x100 px et la sauvegardons dans la « pellicule » de l'utilisateur. Avant d'aller plus loin, vous remarquerez que le tableau de bytes « .NET » est converti en tableau de bytes « WinPRT » à l'aide de la méthode « AsBuffer ». Il s'agit d'une méthode d'extension du framework et il faut penser à ajouter cette ligne d'import de namespace :

```
using System.Runtime.InteropServices.WindowsRuntime;
```

Editing Session

Penchons-nous maintenant plus en détail sur cet objet central : EditingSession. Il s'agit souvent, si ce n'est toujours, du premier pas dans le SDK Nokia. Il existe plusieurs façons d'en créer un :

- ▮ En lui fournissant directement les bytes non décodés de notre image :

via une « CameraCaptureTask », en lisant un fichier, etc.

- En lui fournissant un objet Bitmap, ce qui revient sensiblement au même, avec des informations supplémentaires.
- En lui fournissant un objet de type IStorageFile (via la classe EditingSessionFactory).

Cet objet contient toutes les informations concernant les modifications apportées à notre image. Les différents effets appliqués sont stockés dans un historique et il est possible de revenir en arrière en les annulant. Toutes vos actions ne seront alors appliquées qu'au dernier moment sur l'image et non pas « au fur et à mesure ». Cela a l'avantage de n'effectuer les opérations coûteuses d'un point de vue CPU que lorsque l'on dispose du maximum d'informations sur ce qui va être effectué. Si, par exemple, vous effectuez une découpe sur votre image en ne gardant que la zone centrale, le SDK ne va plus décoder que cette partie et lui appliquer les effets dessus.

Finalement, comme nous avons pu le voir dans l'exemple précédent, cet objet permet de sauvegarder l'image finale sous la forme d'un tableau de bytes ou encore dans un bitmap.

Vous pouvez spécifier à ce moment la taille souhaitée ainsi que quelques options spécifiques à l'encodage (conservation du ratio ou non, espace de couleur utilisé, etc.).

L'image générée peut même être utilisée directement dans votre interface pour l'afficher à l'utilisateur. En voici un exemple :

```
using (var session = new EditingSession(photoBuffer.AsBuffer()))
{
    /* ajout de filtres */

    //Encodage du jpeg
    var wbmp = new WriteableBitmap(
        (int)session.Dimensions.Width,
        (int)session.Dimensions.Height);
    await session.RenderToWriteableBitmapAsync(wbmp);

    /* image est un contrôle «Image» de mon interface XAML */
    image.Source = wbmp;
}
```

LE SYSTÈME DE FILTRES

Qu'est-ce qu'un filtre ?

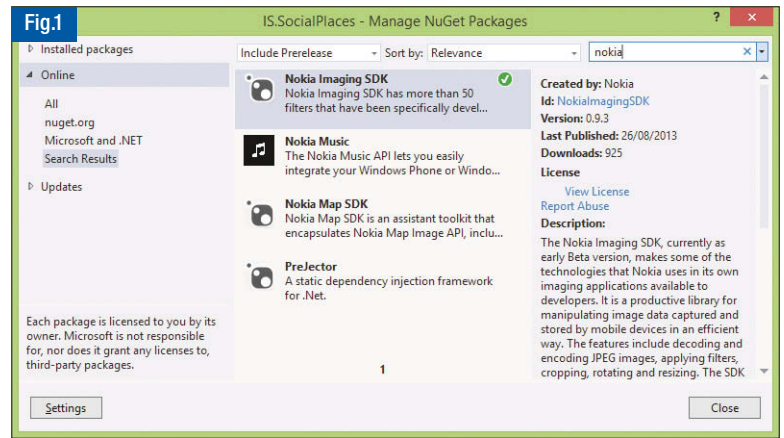
Toutes les manipulations et transformations qu'offre le SDK Imaging de Nokia sont appelées « filtre ». On compte à ce jour une cinquantaine de filtres dans le SDK. Un filtre, du point de vue du SDK, permet aussi bien de couper une image (crop), que d'effectuer une rotation (free rotation) ou bien d'obtenir une image en noir et blanc. Tous les effets sont listés sur la page suivante :

<http://developer.nokia.com/Resources/Library/Lumia/#!nokia-imaging-sdk.html>.

D'un point de vue code, le filtre est un objet WinPRT implémenté en C++. L'intérêt ici réside bien entendu dans la bien meilleure performance du code natif si on la compare à celle du code managé.

En fin de compte, dans le code, nous n'avons pas besoin de connaître précisément le type d'un filtre pour l'utiliser. Tout ce que nous savons, c'est qu'un filtre implémente l'interface `IFilter`. C'est ce type d'objet que vont utiliser les `EditingSession`. L'interface `IFilter` expose uniquement deux méthodes :

- `Apply(IEditingSession)` : Applique le filtre courant sur la session passée en paramètre



- `Undo(EditingSession)` : Annule le filtre courant sur la session.

Toutefois, actuellement, la méthode annule le dernier filtre appliqué à la session, et non pas le filtre courant (voir la partie Limitations)

Evidemment, l'unique paramètre de chacune de ces méthodes n'est autre que la session sur laquelle appliquer ou annuler l'effet correspond au filtre choisi. Mais justement, comment choisir cet effet ?

Comment créer un filtre ?

Le SDK nous permet d'instancier nos filtres en utilisant le design pattern « Factory » au travers de la classe `FilterFactory`. Facile n'est-ce pas ? C'est ainsi que l'on peut créer au choix un des 50 filtres proposés à ce jour par Nokia. Comptons, pour le plaisir, le nombre exact de filtre du SDK avec ce petit bout de code :

```
// récupération de l'objet Type de notre factory
var type = typeof(FilterFactory);
// récupération de toutes les définitions de toutes les
méthodes static et public
var methods = type.GetMethods(BindingFlags.Public | BindingFlags.Static);
// distinction des surcharges
var allNames = methods.Select(m => m.Name).Distinct().ToList();
var count = allNames.Count; // 53 !
```

Voilà ! 53 filtres ! Néanmoins, rien ne dit que si vous exécutez cette portion de code, vous obtiendrez le même résultat, puisque le SDK est en bêta, et que des filtres peuvent apparaître ou disparaître entre temps selon les mises à jour.

Concrètement, la factory expose toute ces méthodes en respectant la nomenclature suivante : `Create[Nom]Filter` où « [Nom] » est le nom du filtre voulu. Pour donner un effet vieilli à une photo par exemple, il est possible d'utiliser la méthode : `FilterFactory.CreateAntiqueFilter()`. Bien entendu, toutes les méthodes de la factory ne sont pas aussi simples; il faudra parfois fournir quelques paramètres en plus à la factory afin de préciser les options du filtre. Pour découper une image, il faut par exemple appeler la méthode `FilterFactory.CreateCropFilter(cropArea)`, où « cropArea » sera un objet de type `Windows.Foundation.Rect` (et non pas de type `System.Windows.Rect`) correspondant à la zone à conserver. Ainsi pour obtenir le carré central de 10 pixels de l'image, le filtre suivant est adapté :

```
var cropFilter = FilterFactory.CreateCropFilter(new Windows.Foundation.Rect(image.Width - 5, image.Height - 5, 10, 10));
```

Après avoir vu ce qu'est un filtre, l'étape suivante est son utilisation !

Appliquer un filtre

Pour résumer, l'objet en charge de manipuler les images est, comme vu précédemment, la session d'édition. En l'occurrence, réaliser une manipulation revient à ajouter un filtre à la session, quel qu'il soit, pourvu qu'il implémente l'interface `IFilter`.

Un exemple simple consiste donc à créer un projet, une nouvelle page, avec deux contrôles `Image`. Le premier contiendra l'image originale, le second l'image transformée afin de visualiser les changements apportés par les filtres. Il faut dans un premier temps ouvrir une session d'édition qui effectuera tout le traitement nécessaire à l'application des filtres. A noter que les objets de session sont de type `EditingSession`, implémentant les interfaces `IEditingSession` et `IDisposable`. Il est donc impératif d'appeler la méthode `Dispose` afin de libérer les ressources (WinPRT, natives, etc.) utilisées par la session. Dans notre cas cela est fait avec le mot clef `using` :

```
StorageFile imageFile = await StorageFile.GetFileFromApplicationUriAsync(new Uri("ms-appx:///Assets/got.jpg"));
using (EditingSession session = await EditingSessionFactory.CreateEditingSessionAsync(imageFile))
{
}
```

La session contient la méthode `AddFilter`, qui servira à appliquer successivement un filtre de vieillissement (antique), puis de retournement (flip) et enfin de balance des ombres (autoLevels) :

```
var antique = FilterFactory.CreateAntiqueFilter();
var flip = FilterFactory.CreateFlipFilter(FlipMode.Horizontal);
var autoLevels = FilterFactory.CreateAutoLevelsFilter();
session.AddFilter(antique);
session.AddFilter(flip);
session.AddFilter(autoLevels);
// le contrôle Image est complètement vide, il est donc ignoré pendant l'étape de rendu de la page. Pour remédier à cela, il faut donner une taille au contrôle.
TransformedImage.Width = OriginalImage.ActualWidth;
TransformedImage.Height = OriginalImage.ActualHeight;
// si le contrôle n'a pas une taille supérieure à 0, alors l'appel de la méthode suivante provoque une exception dans le SDK
await session.RenderToImageAsync(TransformedImage);
```



Le résultat : **Fig.2**

Les méthodes de rendu du SDK sont asynchrones. C'est pourquoi il est nécessaire de placer un `await` juste avant la fin du `using`, car dans le cas contraire, la session est libérée avant d'avoir pu faire le rendu. L'utilisation de la session et des filtres, telle que présentée ici, est très simple. Cela peut légèrement se compliquer lorsqu'on cherche à intégrer des actions utilisateurs.

Création d'un mini studio d'édition de photos :

L'exemple suivant consistera en la création d'une page de type panorama. La première zone permettra de visualiser l'image modifiée dans un contrôle `Image`, la seconde d'appliquer différents filtres, successivement. Bien entendu, il faudra permettre à l'utilisateur de choisir la photo qu'il souhaite éditer. Pour cela, la tâche du Framework Windows Phone `PhotoChooserTask` fera l'affaire :

```
var photoPicker = new PhotoChooserTask { ShowCamera = true };
photoPicker.Completed += async (sender, result) =>
{
    if (_session != null)
        _session.Dispose();
    _session = await EditingSessionFactory.CreateEditingSessionAsync(result.ChosenPhoto);
    await _session.RenderToImageAsync(Image);
};
photoPicker.Show();
```

Ici, une petite astuce pour que cela fonctionne, consiste à spécifier une largeur et une hauteur au contrôle `Image`. En effet, comme celui-ci est vide au départ, sa taille réelle est de 0.

Mais cela pose problème au moment d'inscrire l'image via la session, car celle-ci se base sur ces valeurs.

Afin de remédier à ce problème, il suffit d'ajouter le code suivant dans le handler de l'événement `Loaded` de la page :

```
Image.Width = ImageContainer.ActualWidth;
Image.Height = ImageContainer.ActualHeight;
```

Où `ImageContainer` est le panel contenant le contrôle `Image` (en l'occurrence, une `Grid`, en alignement horizontal et vertical `Stretch`). La deuxième zone, celle des filtres, comportera deux `ListBox`, l'une permettant d'ajouter un nouveau filtre, l'autre permettant de visualiser les filtres ajoutés et de les retirer.

La première sera bindée sur une collection d'objet de type `SimpleEffect`, la seconde sur une collection de `SimpleFilter`. Ces deux classes contiendront le code métier nécessaire au fonctionnement de la page (à noter qu'il ne s'agit en aucun cas d'une guideline officielle, ce sont simplement deux classes créées pour l'occasion, à but de démonstration) :

```
public class SimpleEffect
{
    /// delegate retournant un IFilter du type correspondant à l'effet
    public Func<IFilter> FilterProvider { get; set; }
    /// nom commun de l'effet
    public string Name { get; set; }

    public SimpleEffect(string name, Func<IFilter> filterProvider)
    {
        Name = name;
        FilterProvider = filterProvider;
    }

    /// Ajout d'un filtre correspondant sur la session
    public void ApplyOne(EditingSession session, ICollection<SimpleFilter> history)
    {
        var filter = new SimpleFilter(FilterProvider(), Name, history);
```

```

        filter.Apply(session);
    }
}

public class SimpleFilter : IFilter
{
    /// nom commun de l'effet
    public string Name { get; private set; }
    private readonly IFilter _baseFilter;
    private readonly ICollection<SimpleFilter> _history;

    public SimpleFilter(IFilter baseFilter, string name, ICollection<SimpleFilter> history)
    {
        Name = name;
        _baseFilter = baseFilter;
        _history = history;
    }

    public void Apply(IEditingSession editingSession)
    {
        if (!_history.Contains(this))
            _history.Add(this);
        editingSession.AddFilter(_baseFilter);
    }

    public void Undo(IEditingSession editingSession)
    {
        editingSession.UndoAll();
        _history.Remove(this);
        foreach (var filter in _history)
            filter.Apply(editingSession);
    }
}

```

Ici les objets de type `SimpleFilter` s'utilisent comme des `IFilter` du SDK, mais présentent l'avantage de s'ajouter eux même à un historique et de s'en retirer. Il est utile de conserver un historique, car le SDK permet à ce jour de ne retirer que le dernier filtre ajouté (et pas le deuxième sur cinq par exemple). C'est pourquoi dans ce cas, lorsque nous retirons un filtre, nous les retirons tous (`UndoAll`) avant de rajouter un par un ceux qui restent (pour plus de précision, voir la partie limitations plus bas). Une façon simple de remplir les deux collections avant de binder les `ListBox` dessus :

```

var type = typeof(FilterFactory);
var methods = type.GetMethods(BindingFlags.Public | BindingFlags.Static);

var effects = methods.Where(m => !m.GetParameters().Any()).Select(m =>
    new SimpleEffect(m.Name.Replace(«Create», «»).Replace(«Filter», «»), () => m.Invoke(null, null) as IFilter));
Effects = new ObservableCollection<SimpleEffect>(effects);

AppliedFilters = new ObservableCollection<SimpleFilter>();

```

Ici, toutes les méthodes sans paramètre de la factory sont récupérées, le nom du filtre est extrait de la méthode et un `delegate` est construit à partir de l'objet obtenu par réflexion, le tout passé à un objet `SimpleEffect`.

Ensuite, les handlers de nos `ListBox` sont définis comme suit :

```

// Ajout d'un effet sur l'image
private async void EffectListOnSelectionChanged(object sender,
    SelectionChangedEventArgs e)
{
    if (_session == null || e.AddedItems.Count <= 0) return;

    // On désactive la listBox temporairement pour éviter de
    // faire des traitements d'encodage simultanés sur la session
    EffectListBox.IsEnabled = FilterListBox.IsEnabled = false;

    foreach (var effect in e.AddedItems.OfType<SimpleEffect>())
        effect.ApplyOne(_session, AppliedFilters); // création d'un
    // nouvel IFilter et application sur la session

    EffectListBox.SelectedItems.Clear();
    await _session.RenderToImageAsync(Image);

    // Le traitement est terminé, on peut réactiver la listBox
    // car il n'y a plus de risque d'exception
    EffectListBox.IsEnabled = FilterListBox.IsEnabled = true;
}

// Retrait d'un filtre sur l'image
private async void FilterListOnSelectionChanged(object sender,
    SelectionChangedEventArgs e)
{
    if (_session == null || e.AddedItems.Count <= 0) return;
    EffectListBox.IsEnabled = FilterListBox.IsEnabled = false;
    foreach (var filter in e.AddedItems.OfType<SimpleFilter>())
        filter.Undo(_session);
    await _session.RenderToImageAsync(Image);
    EffectListBox.IsEnabled = FilterListBox.IsEnabled = true;
}

```

Et nous y voilà ! Nous désactivons les `ListBox` au moment des traitements afin d'empêcher un utilisateur de surcharger la session. Reste ensuite à sauvegarder l'image ainsi créée :

```

var jpeg = await _session.RenderToJpegAsync();
using (var ml = new MediaLibrary())
    ml.SavePictureToCameraRoll(«new image from mini studio.jpgg»,
        jpeg.ToArray());

```

Ainsi, pour finir, il peut par exemple être intéressant de combiner des filtres entre eux, afin de rendre plus accessible un effet complexe aux utilisateurs. Dans le cas précédent, il est envisageable d'ajouter un nouvel objet à la collection de `SimpleEffect`:

```

Effects.Insert(0, new SimpleEffect(«Red Cartoon Effect», () =>
    new FilterGroup(new[]
    {
        FilterFactory.CreateAutoLevelsFilter(),
        FilterFactory.CreateCartoonFilter(true),
        FilterFactory.CreateColorAdjustFilter(1.0, 0, 0)
    })));

```

La nouveauté réside ici dans l'emploi du `FilterGroup`, contenant lui-même un tableau d' `IFilter`. Mais la puissance du `FilterGroup` est



qu'il implémente lui-même l'interface `IFilter`. Une fois instancié, il s'utilise donc comme un filtre retourné par la factory.

LIMITATIONS À PRENDRE EN COMPTE

ARM Only

Le SDK ne fonctionne qu'en ARM, il faut donc impérativement utiliser un device physique pour tester le programme.

Encodage séquentiel

A noter qu'une session ne peut encoder en parallèle qu'une image. Plus précisément, si l'on reprend notre exemple de la première page et qu'on y rajoute un troisième contrôle image, que l'on nommera `TransformedImage2`, le code suivant lèvera une exception :

```
var t1 = session.RenderToImageAsync(TransformedImage);
var t2 = session.RenderToImageAsync(TransformedImage2);
await Task.WhenAll(t1, t2);
```

Ici, t1 arrivera certainement à terme, mais t2 lèvera une exception. Pour réaliser cela, il convient donc de le faire séquentiellement :

```
await session.RenderToImageAsync(TransformedImage);
await session.RenderToImageAsync(TransformedImage2);
```

Pensez donc à ce cas, notamment lorsque votre écran permet à l'utilisateur d'ajouter et de retirer successivement plusieurs filtres, comme dans notre deuxième exemple (où le problème est contourné en désactivant l'interface pendant l'encodage).

IFilter.Undo, attention piège

La méthode `Undo` d'un filtre ne retire pas le filtre courant de la session, il n'en retire que le dernier. Ainsi, si l'on considère le code suivant :

```
var imgWritableBitmap = new WriteableBitmap(Image, new ScaleTransform());
var nokiaBitmap = imgWritableBitmap.AsBitmap();
_session = new EditingSession(nokiaBitmap);
var antique = FilterFactory.CreateAntiqueFilter();
var grayscale = FilterFactory.CreateGrayscaleFilter();
antique.Apply(_session);
// ici on a le filtre antique
```

```
grayscale.Apply(_session);
// ici on a les filtres antique et grayscale
antique.Undo(_session);
// bug : ici on aura seulement le filtre antique et pas grayscale
RenderSession();
// l'image apparaît orange vieillis (effet antique) et pas
noire et blanche (effet grayscale)
```

Le comportement attendu n'est pas forcément le bon.

Et la licence ?

Le SDK peut être utilisé dans n'importe quel cas, sauf dans celui d'une application préinstallée. Impossible donc d'utiliser le SDK dans une application qu'un opérateur installerait par défaut sur ses téléphones avant de les vendre.

CONCLUSION

Le SDK Nokia est clairement la solution idéale pour mettre en place des effets graphiques sur des images. Il est léger et performant, et évite des heures fastidieuses de développement pour effectuer des transformations.

Néanmoins, la version testée présente quelques défauts : tout d'abord les plantages (car il y en a) ne sont pas assez bien documentés. Parfois des exceptions se lèvent sans que l'on comprenne vraiment pourquoi; les messages ne nous aident pas forcément, comme par exemple lorsqu'on essaie d'écrire une image dans un contrôle image qui n'a pas de largeur et de hauteur explicites. Certainement que la release corrigera ces défauts.

On aurait aimé pouvoir créer nos propres filtres en complément des existants, quitte à le faire en code managé. Le SDK offre tout de même une alternative tout à fait honorable : les `FilterGroup`, dont nous avons vu un exemple d'utilisation précédemment.

En bref, nous n'avons pas accès à toutes les fonctions de Photoshop, mais pour permettre à nos utilisateurs de faire quelques retouches sur leur photo, c'est tout à fait suffisant ! Et tellement simple qu'on aurait tort de s'en priver.

Thomas Ouvre – Infinite Square
touvre@infinitesquare.com

Jonathan Antoine – Infinite Square
jantoine@infinitesquare.com



ABONNEMENT PDF

30 € par an soit 2,73 € le numéro

www.programmez.com

Détecter le matériel mobile sous Windows Phone 8

Il peut arriver que l'on ait besoin de savoir sur quel type de smartphone Windows Phone est utilisée notre application, de connaître si elle tourne sur un appareil d'entrée de gamme n'ayant par exemple que 256 Mo de mémoire, ceci afin de désactiver la tuile dynamique ou autre chose... Cela peut impacter son utilisation.

Une class va nous sauver la vie !

La class `DeviceExtendedProperties` nous permet de récupérer différentes informations de l'appareil. Ainsi ce petit bout de code nous permettra d'afficher la marque du téléphone :

```
object _property;
if (DeviceExtendedProperties.TryGetValue («DeviceManufacturer»,
out _property))
{
    if (_property != null)
    {
        MessageBox.Show («Marque du smartphone = « + _property.ToString());
    }
}
```

Aussi, en remplaçant la chaîne de caractères «`DeviceManufacturer`» par «`OriginalMobileOperatorName`», on récupérera par exemple l'opérateur d'un téléphone lié à un opérateur. Sur un téléphone acheté indépendamment d'un opérateur elle retournera «`000-FR`». Une autre propriété intéressante est d'identifier anonymement l'utilisateur, très pratique lorsque vous faites un jeu en récupérant le **WindowsLiveAnonymousID**. Notez que sur Windows Phone 8 on utilisera `ANID2` à la place d'`ANID`. En effet `ANID2` est comme `ANID` sauf qu'il est unique par publisher en plus d'être unique par user.

```
public static string GetWindowsLiveAnonymousID()
{
    object _anid;
    string value = String.Empty;
    if (DeviceExtendedProperties.TryGetValue («ANID», out _anid))
    {
        if (_anid != null)
        {
            value = _anid.ToString().Substring(2, 32);
        }
    }

    return value;
}
```

On retrouvera tout le détail de cette classe sur MSDN :

[http://msdn.microsoft.com/en-us/library/windowsphone/develop/microsoft.phone.info.deviceextendedproperties\(v=vs.105\).aspx](http://msdn.microsoft.com/en-us/library/windowsphone/develop/microsoft.phone.info.deviceextendedproperties(v=vs.105).aspx)

Les codes précédents avec `DeviceExtendedProperties` fonctionnent sur toutes les versions des Windows Phone, c'est-à-dire 7.0, 7.1 et 8.0. Toutefois, on ne trouve plus vraiment beaucoup de version 7.0; la class `DeviceStatus`, qui, elle, ne fonctionne pas pour les versions 7.0, devient donc intéressante par sa simplicité et apporte des petites nouveautés

telle l'interrogation sur le type d'alimentation en cours et même sur l'utilisation de clavier externe. Mais il faut l'avouer, il est très rare de rencontrer ce genre de smartphone (ex : Le Dell Venu Pro sorti fin 2010) Pour connaître le fabriquant avec cette classe :

```
DeviceStatus.DeviceManufacturer.ToString();
```

Codons « vert » !

Ce qui est également intéressant, c'est que l'on peut notifier à notre application que l'utilisateur change de système d'alimentation grâce à l'évènement `PowerSourceChanged` :

```
public MainPage()
{
    InitializeComponent();

    this.Loaded += MainPage_Loaded;
}

void MainPage_Loaded(object sender, RoutedEventArgs e)
{
    DeviceStatus.PowerSourceChanged += DeviceStatus_PowerSourceChanged;
}

private void DeviceStatus_PowerSourceChanged(object sender, EventArgs e)
{
    Dispatcher.BeginInvoke(() =>
    {
        if (DeviceStatus.PowerSource == PowerSource.Battery)
        {
            MessageBox.Show («Utilisation de la batterie»);
        }
        else
        {
            MessageBox.Show («Source USB»);
        }
    });
}
```

Tout le détail de cette class sur MSDN :

[http://msdn.microsoft.com/en-us/library/windowsphone/develop/microsoft.phone.info.devicestatus\(v=vs.105\).aspx](http://msdn.microsoft.com/en-us/library/windowsphone/develop/microsoft.phone.info.devicestatus(v=vs.105).aspx)

Dans le même registre Windows Phone 8 nous permet d'aller plus loin et de savoir, par exemple, lorsque le terminal passe en mode d'économie d'énergie grâce à `PowerManager`.

 **Christophe Peugnet**

Gérant et développeur des logiciels SodeaSoft de la société EBLM

<http://www.sodeasoft.com>

Blog : <http://www.peug.net> - Twitter : @tossnet1

Les salaires des développeurs PHP

L'AFUP et le cabinet de recrutement Agence-e ont publié le baromètre des salaires de l'écosystème PHP en France. Ce baromètre se base sur un panel de 1300 répondants, avec une majorité vivant en Île de France.

Des salaires pas très différents du reste de la profession

Les profils « junior », avec 0 à 2 ans d'expérience, se situent dans une fourchette haute des salaires annuels : 31 000 €.

Dans le développement, il n'est pas rare que l'on débute à 28-29 000 € (Île de France). Pour une expérience de 3 à 5 ans, le salaire moyen passe à 35 000 €. Les profils « senior++ » (10 ans d'expérience et +) s'affichent à 45 000 €. Une légère majorité des répondants dit avoir une part de variable dans le salaire, cette part est faible : 5 à 6 %.

Le développeur PHP s'estime-t-il bien payé ? Visiblement non ! 53 % disent être mal ou très mal payés ! Avec l'expérience, ce sentiment s'estompe. Mais le sentiment d'être bien payé ne dépasse jamais 40 %.

Un salaire moyen très variable

Si on regarde par secteur d'activité, on constate que le salaire moyen varie considérablement, comme le montre le tableau.

	0-2 ans	3-5 ans	6-10 ans	10 ans/+
agence de communication	25	33,3	36	40
éditeur	32,3	35,5	41	48
société de services	32	38	38	40
startup	29	36	42,5	50

Année d'expérience. Salaires annuels bruts en milliers €

Modérons tout de même ces chiffres, car cela varie aussi selon le type d'entreprise : PME, grand compte ou indépendant. Et là encore, c'est le grand écart. Élément intéressant, l'indépendant gagne plus que le développeur salarié (à partir de 35 k€). Les très petites entreprises proposent souvent (selon le baromètre), un salaire moyen très bas (22,6 k€) contre un salaire moyen de 28 à 33 k€ dans la plupart des entreprises. Pour les développeurs ayant une compétence de 10 ans et +, la fourchette haute varie de 43 à 47 k€.

Faut-il être développeur, architecte ou consultant ?

Sans surprise, le développeur reste à des salaires moyens (très) inférieurs par rapport aux postes de responsables. Avec de l'expérience, si le développeur le peut, et s'il le souhaite, il pourra évoluer vers des postes de chef de projets ou d'architecte, la rémunération grimpe rapidement. Mais les élus à ces postes sont peu nombreux.

	3-5 ans expérience	6-10 ans expérience
développeur	33	36,8
lead développeur / architecture	40,7	42,8
chef de projet / consultant	39	40
cadre dirigeant / chef d'équipe	38	45

Salaires moyens en K€

Développeur polyvalent ou spécialisé ?

« En début de carrière, les écarts de rémunération entre les différentes spécialisations semblent largement négligeables. Au cours de la prise d'expérience, et au fur et à mesure que la spécialisation est poussée, les résultats de l'étude montrent un écart grandissant en faveur des utilisateurs du framework Symfony ou du Zend Framework » analyse le baromètre.

(voir tableau bas de page, en k€/an)

Cependant, la spécialisation peut être à double tranchant. Si le salaire est souvent supérieur, le développeur doit être capable d'évoluer et de s'adapter à de nouvelles technologies. « La formation est également un levier permettant la revalorisation salariale. Un quart des employés ayant récemment suivi une formation l'ont directement et immédiatement constaté. » constate l'étude. Que motive le développeur PHP ? L'intérêt du projet -et sa technicité-est l'élément qui revient le plus souvent (45 % en moyenne). L'ambiance dans l'entreprise est un autre élément non négligeable (un quart des répondants). La rémunération est très loin derrière.

François Tonic

Data Scientist : le nouveau profil

Depuis quelques mois le Data Scientist est le nouveau profil recherché par certaines entreprises ayant d'importants besoins en analyses de données, big data et business intelligence. Il est un spécialiste de la donnée, de son analyse et de ce que l'on peut en tirer en informations, interprétations et résultats.

Il est à la confluence de la technologie, des outils, de la donnée, du marketing. Pour la Harvard Business School (une des écoles de commerce les plus réputées au monde), le data scientist est LE métier du 21^e siècle...

Avec l'explosion du volume de données, la complexité des environnements de traitements et des données hétérogènes et non structurées, certaines entreprises ont besoin de ce profil hybride.

Ce n'est pas réellement un nouveau métier mais l'évolution de profils existants, comme l'analyste de données, même s'il ne s'agit pas d'être le nouveau statisticien.

Dans une étude récente d'Urban Linker, la grille salariale est la suivante :

Expérience	Salaires
-2 ans	50-60 k€
2-3 ans	65-75 k€
5 ans	+75 k€

Source : <http://www.urbanlinker.com/etude-des-salaires-developpeurs-2013-recrutement/>

Par comparaison, à Londres, un profil identique peut atteindre 60 000 £.

Aux États-Unis, le salaire sera souvent compris entre 75 et 100 000 \$ (mais un profil exceptionnel sera minimum au double).

Le Data Scientist est un profil haut de gamme et stratégique pour une entreprise ayant besoin de ce poste. Ces profils sont rares.

Autre compétence du big data, le data analyst qui est plus orienté technique. L'étape suivante est le Data Scientist.

Les profils « big data » vont se multiplier durant les prochaines années.

Les cursus universitaires commencent à ouvrir ici et là, parfois avec l'aide d'éditeurs. Des compétences seront indispensables pour ce genre de poste : statistiques, algorithmes, langages spécifiques (ex. : langage R).

En K€/an	0 - 2 ans	3 - 5 ans	6 - 10 ans	10 ans et +
Zend Framework	32,5	36	41,1	45,8
Symfony	31	37,2	42	45
Framework propriétaire	33	35	33,5	45
Drupal	28,8	32,6	39,3	47,3
Wordpress	30	31,8	38,4	39,7
Autre Framework ou CMS du marché	32	29,8	37,8	40,1
Solution E-Commerce	30	31,8	38,4	39,7
Multi-spécialiste	32	34,5	40,5	43,6
Autre	30,3	34	40	48,5

Osez (ou pas) le Cobol ?

En informatique, il ne faut jamais enterrer une technologie. Elle peut revenir sous une forme ou sous une autre. Le langage Cobol est apparu en 1959. Plus de 50 ans après, il demeure un des langages les plus anciens dans le monde (administrations, banques, assurances, grandes entreprises...). Au minimum 200 milliards de lignes de code Cobol fonctionnent toujours. Faut-il pour autant oser une carrière Cobol ou tout le moins, avoir une compétence ?

Au printemps dernier, Micro Focus, éditeur spécialisé dans les outils Cobol, avait dévoilé son Observatoire Cobol. Les résultats confirment la vivacité de ce « vieux langage » : 81 % des professionnels IT interrogés utilisaient Cobol pour les applications critiques, près de 46 % des répondants « Entreprises » comptabilisent plus de 2 millions de lignes de code Cobol dans leur entreprise, 16 % dépassent même les 10 millions de lignes. Les résultats de l'Observatoire 2013 montrent une prévalence d'applications de moindre ampleur dans les SSII. Les applications Cobol continuent donc de représenter de grosses volumétries. Une majorité des entreprises continue à utiliser les logiciels Cobol, mais 35 % souhaitaient une modernisation, partielle ou totale, de ce patrimoine applicatif. Mais attention, modernisation ne signifie pas réécriture totale

(moins de 20 % des entreprises interrogées dans le cadre de l'observatoire). En clair, Cobol reste un langage clé pour toute une partie de l'économie. Ce patrimoine logiciel a besoin de maintenance, de développements et d'évolution.

Les profils et compétences Cobol : un problème réel ?

Si on regarde l'observatoire cité ci-dessus, aujourd'hui, les 2/3 des développeurs/experts Cobol ont au moins 40 ans (dont 1/3 a plus de 50 ans). Dans les années à venir, de nombreuses compétences vont disparaître (chômage, retraite, changement de carrière), or, la compétence Cobol reste importante dans plusieurs secteurs économiques : banques, assurances, grande distribution, administration, bourse...

Il faut distinguer 2 types de profils :

le coboliste et le développeur avec une compétence cobol. Pour Stéphane Croce de Cobol-IT, c'est un

marché actif, mais qui ne cherche pas forcément des cobolistes purs et durs, mais plutôt des cobolistes ++ ayant la faculté de maîtriser le passé (Cobol) et l'avenir (nouvelles technologies, Web, langages modernes). Ces profils peuvent à fois maintenir le code actuel, tout en participant aux projets de modernisation. « Il doit avoir une compréhension de l'architecture existante » avertit M. Croce. Patrick Rataud (Micro Focus) ne dit pas autre chose : il faut des développeurs polyvalents et non des cobolistes. Il faut sortir du vieux débat Cobol contre Java, .net, etc. Mais la France forme peu de compétences Cobol ce qui peut obliger les entreprises à les

“ il faut des développeurs polyvalents et non des cobolistes ”

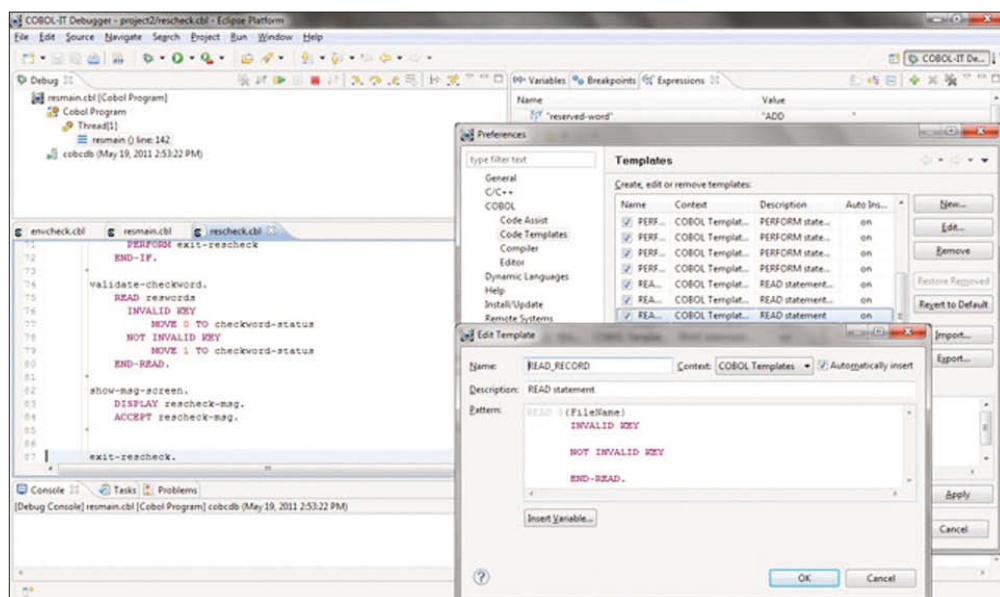
chercher ailleurs. Ensuite, une entreprise peut aussi avoir du mal à attirer des compétences à cause de la géographie : certaines régions sont plus attractives que d'autres.

Cette compréhension de l'existant est vitale pour maintenir et moderniser. L'erreur serait de confier ces chantiers sensibles à des développeurs n'ayant aucune expérience avec le patrimoine à remodeler.

Pour les entreprises concernées, il s'agit d'utiliser des plates-formes modernes, Java, .Net, langages Web, des bases de données « standards », un outillage plus standard (ex. : Eclipse).



© IBM



Un concours de jeux vidéo en cobol !

L'éditeur Micro Focus a lancé en début octobre un concours de programmation : créer un jeu vidéo en Cobol ! Une des conditions est que le code soit écrit à 50 % (minimum) avec Visual Cobol.

Résultat : fin janvier 2014.

Site : <http://academic.microfocus.com/code/index.asp>

À la date du 4 novembre, on pouvait consulter :

- 27 offres d'emploi Cobol sur Lesjeudis.com : analystes, analystes-programmeurs, ingénieur, développeur
 - 150 offres sur ANPE.fr (toute nature) : le ciblage n'est pas très précis, mais on peut tout de même trouver plusieurs dizaines d'offres d'emplois avec une dominante Cobol.
 - 19 offres Cobol sur monster.fr
- Côté salaire, les écarts sont importants, selon le profil, le poste et la société concernée (la plupart du temps des SSII et des grandes entreprises). Pour un développeur, la grille salariale ne sera pas très différente d'un développeur « standard », de 26 à 35 000 € par an. Pour certains profils pointus, on arrive à 50 000 € (minimum).

Faut-il réellement oser le Cobol ?

Depuis le début de cet article, vous pourriez croire que le Cobol est une carrière possible, mais en réalité, il faut se montrer très prudent. Oui, il y a une réelle demande Cobol, mais, sauf à être déjà coboliste, ne choisissez pas cette carrière sans avoir au minimum une double compétence sur d'autres outils et plates-formes. Car vous pouvez être développeur Java, C++ et avoir une compétence

Cobol. Cette approche est sans doute la meilleure pour votre carrière. Et avec la multiplication des chantiers de modernisation, on ne cherche pas tant des cobolistes que des développeurs ouverts et multi-compétences.

Cursus et formations

Des éditeurs et constructeurs du monde Cobol (Cobol-IT, Micro Focus, IBM...) participent activement à la formation Cobol et soutiennent les cursus et les modules universitaires. Mais attention, parfois, les universités ne proposent que des formations de quelques jours comme à l'université de Nantes ; le Cobol reste largement absent de l'enseignement supérieur. Néanmoins, vous pouvez vous tourner vers des formateurs comme CGI Learning Solutions.

« Oui le Cobol est toujours demandé.

Oui, il existe une réelle demande pour ces compétences.

Non, ne soyez pas un pur coboliste ».

François Tonic

COBOL EN QUELQUES CHIFFRES

90 % des entreprises du Fortune 500 utilisent chaque jour du Cobol

70 % des logiciels critiques sont écrits en Cobol

500 millions de téléphones portables se connectent grâce à ce langage

85 % des transactions business sont passées en Cobol

95 % des transactions des CAB utilisent le Cobol

5 milliards de lignes de code sont développées en Cobol chaque année

+ 2 millions de personnes travaillent directement ou indirectement en Cobol (dans le monde, estimation)

Sources : Aberdeen Group ; Giga Information Group ; Database & Network Journal ; The COBOL Report ; SearchEngineWatch.com ; Tactical Strategy Group ; The Future of COBOL Report.



metrixware

Cobos : Un environnement de développement COBOL Open source sous eclipse

Les développements Mainframe constituent encore aujourd'hui le principal socle technologique de la majorité des entreprises. Au cœur des Systèmes d'Information, ces usines logicielles sont et resteront stratégiques pour l'activité de bon nombre de banques, assurances et grandes industries mondiales.

Le langage COBOL, à l'origine des développements Mainframe, manque cruellement d'attractivité pour les développeurs. L'enjeu pour les DSI est plus que jamais de pérenniser et d'attirer les compétences COBOL.

Metrixware répond avec Cobos, l'atelier de développement COBOL Open source sous eclipse.

Construit en collaboration avec des développeurs, Cobos propose une interface intuitive et simple particulièrement bien adaptée à l'environnement mainframe.

Cet atelier s'intègre naturellement avec les éditeurs eclipse XML, HTML, Java,... ainsi qu'avec les outils open source de gestion de projets, modélisation, gestion de sources, etc...

Metrixware est un spécialiste de la modernisation des applications mainframe et propose, outre Cobos, la suite "System Code" pour la Cartographie applicative, la suite "Mainframe Integrator" pour la webisation des applications "green screen" permettant d'amener les applications mainframe jusque sur les tablettes sans modifier l'existant.

Les technologies et l'expertise de Metrixware aident ainsi les DSI à faire de leurs applications un avantage concurrentiel immédiat et à moindre coût.

Cobos est une solution techniquement et financièrement avantageuse, face aux solutions commerciales existantes. Cobos est le plus évolué des IDE COBOL mainframe Open source grâce à l'enrichissement des fonctionnalités avancées d'Eclipse associées aux meilleures pratiques issues de l'expérience Mainframe.

Plus d'info sur <http://cobos.metrixware.org/>



Raspberry Pi n°99 : mettre en place un serveur web personnel avec Nginx, PHP et MySQL

Vous ne me croyez pas ? Nous allons vous démontrer que tout est possible avec un Raspberry Pi...

Nginx

Nginx est un serveur web HTTP ainsi qu'un "reverse proxy". Ce serveur HTTP a tendance à se démultiplier sur la toile car il fonctionne de manière différente : plutôt que d'utiliser les threads pour gérer les différentes requêtes reçues, il utilise un mécanisme asynchrone poussé par une gestion des événements. Il est aujourd'hui utilisé par un grand nombre de sites: GitHub, Netflix, Hulu, Wordpress.com, etc.

Le fonctionnement de Nginx permet d'avoir une empreinte mémoire largement plus faible qu'avec un serveur synchrone type Apache. De plus, dans des situations nécessitant de nombreux accès simultanés, Nginx s'avère être plus performant.

Un serveur synchrone nécessite l'utilisation d'un load-balancing très coûteux, là où Nginx n'en a pas besoin.

PHP + MySQL

PHP n'est plus à présenter, c'est le langage de scripts server-side le plus utilisé au monde pour la réalisation de sites Web. MySQL quant à lui, a tendance à être de moins en moins utilisé. Originellement développé par Sun Microsystems, ce SGBDR a été largement démocratisé via sa version Community permettant d'héberger gratuitement son propre serveur sans avoir à payer de licence.

Aujourd'hui maintenu par la société Oracle suite au rachat de Sun Microsystems, certains développeurs originels du projet ont décidé de créer un fork de MySQL, car la société Oracle n'avait pas la même roadmap pour le SGBDR qu'eux; c'est notamment pour cela qu'est né MariaDB.

Raspberry Pi

Le Raspberry Pi est un microsystème basé sur une architecture ARM permettant d'installer un système d'exploitation Linux (mais pas seulement). Basé sur des composants peu coûteux, et suffisant pour une utilisation embarquée (carte SD, Processeur ARM, USB, etc.), la communauté a grossi de manière exponentielle.

Il est aujourd'hui très facile de trouver des articles ou des tutoriels sur comment faire certaines choses avec votre Raspberry Pi. Programmez ! en a déjà publié plusieurs.

Raspbian

Raspbian est une distribution Linux basée sur Debian adaptée au Raspberry Pi. C'est la première distribution directement maintenue par la fondation Raspberry. Grâce à la forte communauté derrière l'appareil, Raspbian dispose d'une grande quantité de paquets précompilés prêts à être installés sur votre appareil sans avoir à maîtriser la compilation sous Linux.

INSTALLATION DU RASPBERRY PI

Avant toute chose, pour mener à bien l'expérience, il est nécessaire d'avoir plusieurs composants et matériels :

- Un Raspberry Pi
- Une carte SD
- Un ordinateur (PC ou MAC)
- Une alimentation Micro-USB

Téléchargement et extraction de la distribution

Pour notre article, nous allons utiliser la distribution Raspbian. Nous aurions pu utiliser ArchLinux ou Pidora, mais Raspbian étant la distribution principale, nous n'avons pas vu l'intérêt d'aborder les deux autres.

Allez sur la page de téléchargement de Raspberry

<http://www.raspberrypi.org/downloads>.

Puis téléchargez la dernière distribution Raspbian. À l'écriture de cet article, la distribution date du 10 septembre 2013.

Une fois téléchargée, extrayez directement l'image de la distribution depuis le fichier .zip.

Installation de la distribution sur la carte SD

Nous allons maintenant installer la distribution précédemment téléchargée sur la carte SD.

Il y a plusieurs façons de le faire : sous Linux, sous Windows, sous OSX, en ligne de commande, etc.

Nous allons voir ici comment le faire en ligne de commande sous OSX.

Si vous souhaitez voir comment faire sur d'autres plateformes ou d'une manière plus simple, je vous invite à consulter le Wiki en question : http://elinux.org/RPi_Easy_SD_Card_Setup.

Premièrement, lancez votre application **Terminal**. Une fois le terminal ouvert, branchez la carte SD à votre ordinateur. Nous allons effectuer plusieurs étapes :

- Identifier le disque associé à votre carte SD ;
- Démonter ce disque dans le système ;
- Copier l'image de la distribution sur la carte SD.

Voyons cela en ligne de commandes :

- Dans un premier temps, exécutez la commande `diskutil list`.

Une liste de disques apparaîtra, par exemple :

/dev/disk0			
#:	TYPE	NAME	IDENTIFIANT
0:	GUID_partition_scheme		disk0
1:	EFI		disk0s1
2:	Apple_HFS	Macintosh HD	disk0s2
3:	Apple_Boot	Recovery HD	disk0s3

Dans mon exemple, il n'y a qu'un seul disque, il n'y a donc pas de carte SD. Il vous faudra être capable ici d'identifier votre carte SD. Pour vous simplifier la tâche, je vous invite à débrancher tous vos disques externes et clés USB, et à ne brancher que la carte SD à ce moment-là.

Normalement l'identifiant de votre carte SD devrait être `/dev/disk1`, car c'est le deuxième disque après le disque principal de votre ordinateur.

Vous pouvez aussi vous référer à la taille de votre disque affichée par la commande `diskutil list`.

Si vous avez une carte SD de 8Go, il est peu probable de vous tromper si la seule chose branchée sur votre ordinateur est un disque dur externe.

- Une fois votre disque identifié, nous allons le démonter du système avec la commande `diskutil unmountDisk /dev/disk1`.

Bien entendu vous adaptez l'identifiant de la commande avec celui que vous venez de chercher dans l'étape précédente.

Nous démontons ici le disque, car ne l'oublions pas, OSX est basé sur UNIX qui propose une séparation entre hardware et software. Tant qu'un disque n'est pas monté, il n'est pas utilisable dans le système en tant que système de fichier. Cependant rien ne vous empêchera d'écrire « sauvegarde » des octets sur celui-ci.

- Maintenant que votre disque est démonté, nous allons copier l'image de la distribution avec la commande `dd bs=1m if=votreImageDeRaspbian.img of=/dev/disk1`

Faites très attention lors de l'exécution de cette commande; OSX ne tient plus compte de ce disque en gérant le système de fichiers.

Cela veut dire que le système vous laissera écrire comme bon vous semble sur ce disque. Si par malheur vous vous trompez de disque en ne choisissant pas la carte SD, vous pouvez être à peu près sûr que votre système de fichiers en place sera corrompu et que certaines données seront écrasées.

Voyons les paramètres de cette commande :

- `bs=1m` : indique la taille de bloc que doit utiliser `dd` pour faire les copies. Dans notre cas, `dd` va copier des blocs de 1 mega-octet par mega-octet sur notre carte SD ;
- `if=votreImageDeRaspbian.img` : indique l'image à utiliser pour copier notre distribution sur la carte SD. Vous devez ici spécifier le chemin vers votre fichier `.img` précédemment extrait ;
- `of=/dev/disk1` : indique l'emplacement où les octets de l'image d'entrée seront copiés.

`dd` est une commande très utilisée; nous l'utilisons ici pour copier les données d'un fichier sur un disque, mais nous pourrions très bien faire de la copie de disque à disque, ou même graver un CD directement en utilisant cette commande.

Premier lancement

Votre carte SD est maintenant prête à être utilisée, nous vous invitons à la brancher directement à votre Raspberry Pi et à brancher celui-ci à l'alimentation (sans oublier de le raccorder à un câble Ethernet provenant de votre box).

Par défaut, le Raspberry Pi ne démarre pas avec une interface graphique, nous devons donc nous y connecter en SSH.

Par défaut, Raspbian a un utilisateur configuré avec un mot de passe :

- Login : `raspberry`
- Mot de passe : `pi`

Mais avant de vous connecter en SSH, vous devez connaître l'adresse IP de votre appareil. Si vous êtes sur OSX, je vous invite à utiliser un utilitaire nommé LanScan disponible sur l'AppStore Fig.1.

Il vous suffit de scanner votre réseau et de prendre l'adresse IP correspon-

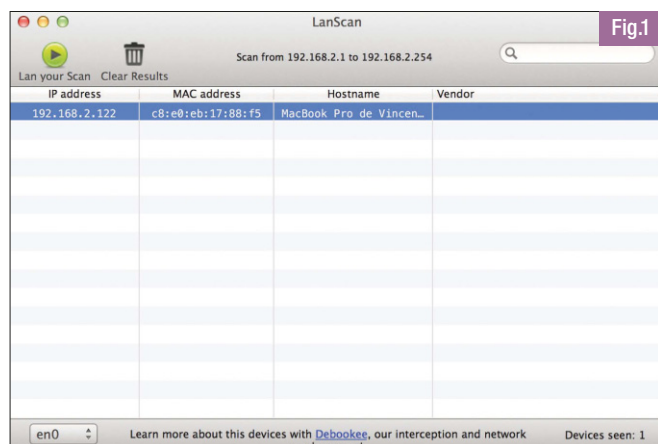


Fig.2

192.168.2.126

b8:27:eb:43:bf:95

piT*** Go Pro! ***

Raspberry Pi Foundation

dant à une interface avec comme Vendor : **Raspberry Pi Foundation** Fig.2. Ouvrez un terminal, puis connectez-vous en SSH sur celui-ci :

```
ssh pi@192.168.2.126
```

Puis entrez le mot de passe par défaut.

Vous voilà maintenant connecté sur votre Raspberry Pi, nous allons maintenant le configurer un peu.

Mais avant cela, je vous invite à faire une mise à jour du système : `sudo apt-get update && sudo apt-get upgrade`.

Configuration du système

La fondation Raspberry Pi a développé plusieurs outils pour son appareil. Le principal étant **raspi-update**.

Habituez-vous à voir écrit `raspi` et pas `Raspberry Pi` : c'est le nom raccourci régulièrement utilisé sur la toile.

Je vous invite donc à taper la commande `sudo raspi-config` Fig.3.

Cette interface est découpée en 9 parties :

1. **Expand Filesystem** - permet au système de prendre toute la place disponible sur la carte SD en agrandissant la partition ;
2. **Change User Password** - permet de changer le mot de passe de l'utilisateur **raspberry** ;
3. **Enable Boot to Desktop/Scratch** - permet de changer le mode de démarrage (mode console ou graphique) ;
4. **Internationalisation Options** - permet de modifier les paramètres de localisation de votre Raspberry Pi (Langue, configuration clavier, etc.)
5. **Enable Camera** - permet d'activer le module caméra fabriqué par la fondation Raspberry Pi et de télécharger les outils développés par la fondation à ce sujet ;
6. **Add to Rastrack** - permet d'ajouter le Raspberry Pi sur le site internet **Rastrack**. Ce site permet d'avoir un suivi de tous les Raspberry Pi disponibles dans le monde ;
7. **Overclock** - permet de surcadencer le processeur de votre appareil (attention, c'est plutôt risqué !) ;
8. **Advanced Options** - permet de configurer certaines autres options: la répartition mémoire, l'activation du SSH, le nom de l'appareil, etc. ;
9. **About raspi-config** - affiche quelques informations sur l'outil.

Dans notre cas, nous allons demander le redimensionnement de la partition sur toute la carte SD, nous allons changer le mot de passe par défaut et modifier quelques options avancées.

Dans un premier temps, choisissez donc l'option numéro 1 (**Expand Filesystem**) pour automatiquement redimensionner la partition. Ici tout est automatisé, vous n'avez rien à faire.

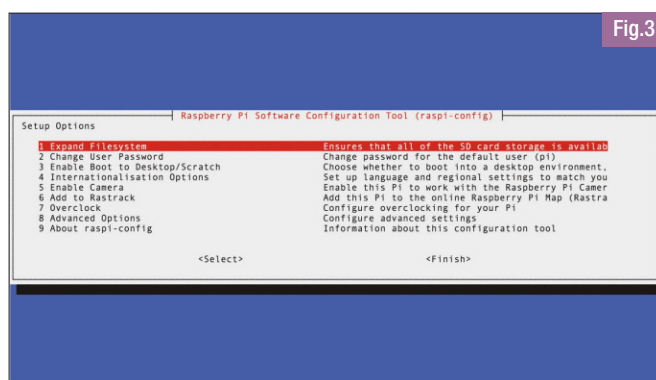


Fig.3

Sélectionnez ensuite l'option numéro 2 *Change User Password*, l'outil va disparaître pour laisser apparaître un message dans le prompt :

```
Entrez le nouveau mot de passe UNIX :
```

Vous devrez taper votre nouveau mot de passe deux fois. Ne vous inquiétez pas si le mot de passe ne s'affiche pas quand vous le tapez: la philosophie UNIX veut que le mot de passe ne doit surtout pas être affiché ou remplacé par d'autres caractères, car cela permettrait de déterminer la longueur du mot de passe.

Nous allons aller dans le sous-menu de l'option numéro 9 *Advanced Options*. Nous allons modifier deux choses : le nom d'hôte de votre Raspberry Pi et la répartition de la mémoire vidéo.

Sélectionnez l'option A2 *Hostname* dans le menu et saisissez le nouveau nom de votre appareil (par exemple piWeb) puis validez. Il faudra redémarrer complètement l'appareil pour que cela soit pris en compte, mais avant cela, nous allons modifier la répartition mémoire.

Sélectionnez maintenant l'option A3 *Memory Split*. Par défaut, le Raspberry Pi alloue 32 Mo de RAM pour la vidéo, dans notre cas, la vidéo ne sera qu'en mode console via le SSH, nous allons donc réduire cette mémoire à 16 Mo.

Une fois le choix validé, je vous invite à redémarrer votre appareil avec la commande `sudo reboot`.

MISE EN PLACE DU SERVEUR WEB

Maintenant que nous avons une installation complète et fonctionnelle sur notre Raspberry Pi, nous allons enfin faire l'installation de notre serveur Web.

Pour cela, nous avons deux méthodes possibles : la méthode la plus simple via les paquets précompilés et celle (un peu) plus compliquée, en compilant nous-mêmes les sources.

Installation facile

Dans un premier temps, nous allons installer quelques paquets nécessaires pour la configuration de Nginx :

```
sudo apt-get install php5-fpm libgd2-xpm libpcrecpp0 libxpm4
```

Une fois que cela est fait, nous allons lancer l'installation de nginx de la même manière :

```
sudo apt-get install nginx
```

Nginx est maintenant installé sur votre appareil. Comme vous pouvez le voir, c'est plutôt simple.

Nous allons tout de même configurer un peu plus notre serveur Web pour pouvoir le tester. La première chose à faire est de créer un répertoire pour héberger nos fichiers :

```
sudo mkdir /var/www
sudo chown -R www-data:www-data /var/www
```

Ici nous créons le répertoire `/var/www`, répertoire classique sur un système UNIX pour l'hébergement des fichiers Web, puis nous donnons les droits nécessaires sur le répertoire à l'utilisateur `www-data`, utilisateur utilisé pour le **daemon** de Nginx.

Pour valider le fonctionnement de PHP, nous allons créer un fichier très simple à la racine de ce dossier contenant :

```
<?PHP phpinfo(); ?>
```

(Vous pouvez faire cela simplement avec la commande `sudo touch /var/www/index.php && sudo sh -c 'echo "<?php phpinfo(); ?>" >> /var/www/index.php'`)

Nous allons maintenant modifier le fichier de configuration de Nginx pour utiliser PHP et notre nouveau répertoire de fichier Web. Modifiez le fichier `/etc/nginx/sites-available/default` avec votre éditeur préféré (nano, vim, etc.) pour ressembler à ceci :

```
server {
    listen 80;
    root /var/www;
    index index.php index.html index.htm;
    location ~ .PHP$ {
        fastcgi_pass unix:/var/run/php5-fpm.sock;
        fastcgi_index index.php;
        include fastcgi_params;
    }
}
```

Redémarrez votre serveur Nginx :



```
sudo service nginx restart
```

Si vous avez correctement suivi les étapes d'installation et de configuration, vous pouvez maintenant vous connecter avec un navigateur Web sur votre Raspberry Pi et vous obtiendrez une page d'information sur PHP **Fig.4**.

Installation complexe

Si vous avez suivi le processus d'installation simple utilisé précédemment, je vous invite à désinstaller Nginx :

Fig.4

PHP Version 5.4.4-14+deb7u4	
	
System	Linux piTELIOS 3.6.11+ #538 PREEMPT Fri Aug 30 20:42:08 BST 2013 armv6l
Build Date	Aug 27 2013 02:09:53
Server API	FPM/FastCGI
Virtual Directory Support	disabled
Configuration File (php.ini) Path	/etc/php5/fpm
Loaded Configuration File	(none)
Scan this dir for additional .ini files	/etc/php5/fpm/conf.d
Additional .ini files parsed	/etc/php5/fpm/conf.d/10-pdo.ini
PHP API	20100412
PHP Extension	20100525
Zend Extension	220100525
Zend Extension Build	API220100525,NTS
PHP Extension Build	API20100525,NTS
Debug Build	no
Thread Safety	disabled
Zend Signal Handling	disabled
Zend Memory Manager	enabled
Zend Multibyte Support	provided by mbstring
IPv6 Support	enabled
DTrace Support	disabled
Registered PHP Streams	https, ftps, compress.zlib, compress.bzip2, php, file, glob, data, http, ftp, phar, zip
Registered Stream Socket Transports	tcp, udp, unix, udg, ssl, sslv3, tls
Registered Stream Filters	zlib.*, bzip2.*, convert.iconv.*, string.rot13, string.toupper, string.tolower, string.strip_tags, convert.*, consumed, dechunk
This program makes use of the Zend Scripting Language Engine: Zend Engine v2.4.0, Copyright (c) 1998-2012 Zend Technologies	
Powered By 	

```
sudo apt-get -y install nginx
sudo apt-get -y remove nginx
sudo apt-get clean
sudo apt-get autoremove
```

Redémarrez votre Raspberry Pi pour être sûr que toutes les modifications ont été prises en compte (sudo reboot).

Commençons par installer quelques dépendances :

```
sudo apt-get -y install curl build-essential libpcre3-dev
libpcre++-dev zlib1g-dev libcurl4-openssl-dev libssl-dev
```

Une fois installé, il nous rapatrie le code source de Nginx. Pour l'article, nous allons utiliser la version 1.2.2 du serveur :

```
curl http://nginx.org/download/nginx-1.2.2.tar.gz | tar zxvf -
```

Cette commande va télécharger l'archive contenant le code source et va l'extraire. Allez dans le dossier (cd nginx-1.2.2) et effectuez une configuration du code source :

```
./configure --sbin-path=/usr/sbin/nginx \
--conf-path=/etc/nginx/nginx.conf \
--pid-path=/var/run/nginx.pid \
--error-log-path=/var/log/nginx/error.log \
--http-log-path=/var/log/nginx/access.log \
--with-http_ssl_module \
--without-http_proxy_module
```

Les 5 premières lignes de notre commande permettent de configurer Nginx pour correctement fonctionner sur notre distribution à base de debian. Les deux dernières indiquent que nous souhaitons activer le SSL (https) sur notre serveur, mais nous désactivons le mode proxy de Nginx. Une fois cela fait, il ne reste plus qu'à compiler et installer :

```
make && sudo make install
```

Votre serveur Nginx est correctement installé. Pour la configuration et le lancement, utilisez l'explication fournie dans la section précédente.

CONFIGURATION DU SERVEUR NGINX

Paramétrer un Firewall

Il existe plusieurs moyens de configurer le firewall sous Linux. Du plus simple au plus compliqué en passant par des outils graphiques. Nous allons voir ici comment le configurer en ligne de commandes avec la commande iptables. La première commande que vous devez essayer est

```
sudo iptables -L
```

Cette ligne de commande va afficher toutes les règles définies dans votre firewall

```
Chain INPUT (policy ACCEPT)
target    prot opt source               destination

Chain FORWARD (policy ACCEPT)
target    prot opt source               destination
```

```
Chain OUTPUT (policy ACCEPT)
target    prot opt source               destination
```

► **INPUT** correspond aux règles de filtrage à l'entrée de votre appareil (e.g. ce qui provient de l'extérieur: d'internet ou de votre réseau local) ;

► **OUTPUT** correspond aux règles de filtrage à la sortie de votre appareil (e.g. ce qui sort vers l'extérieur de votre appareil) ;

► **FORWARD** correspond aux règles de filtrage associées aux différentes connexions à l'intérieur de votre système (typiquement un message provenant de la carte Wi-Fi qui sera redirigé vers la carte Ethernet)

FORWARD est très utile lorsque nous voulons nous servir de notre appareil comme d'un routeur. Dans notre cas, nous allons principalement utiliser **INPUT/OUTPUT**.

La première chose à faire pour sécuriser notre appareil est d'interdire toute connexion entrante et sortante :

```
sudo iptables -t filter -P INPUT DROP
sudo iptables -t filter -P FORWARD DROP
sudo iptables -t filter -P OUTPUT DROP
```

Nous allons ensuite vider notre table de règles (au cas où vous en auriez) :

```
sudo iptables -t filter -F && sudo iptables -t filter -X
```

Afin de pouvoir continuer à nous connecter en SSH, nous allons autoriser le SSH, et, bien entendu, afin de préserver les connexions déjà établies, nous allons aussi ajouter une règle :

```
sudo iptables -t filter -A INPUT -p tcp --dport 22 -j ACCEPT
sudo iptables -A INPUT -m state --state RELATED,ESTABLISHED
-j ACCEPT
sudo iptables -A OUTPUT -m state --state RELATED,ESTABLISHED
-j ACCEPT
```

Nous avons dès à présent un firewall de base qui fonctionne. Nous devons cependant autoriser les requêtes entrantes vers notre serveur Web, pour cela :

```
sudo iptables -t filter -A INPUT -p TCP --dport 80 -j ACCEPT
```

Voilà, votre Raspberry Pi est maintenant prêt à être ouvert sur le Web et sera totalement sécurisé aux connexions entrantes/sortantes non désirables.

Notez que si vous souhaitez pouvoir effectuer quelques requêtes classiques depuis votre appareil, il va être nécessaire de rajouter quelques lignes pour autoriser la résolution de nom, la connexion FTP, etc. depuis le Raspberry Pi :

```
sudo iptables -t filter -A OUTPUT -p tcp --dport 21 -j ACCEPT
sudo iptables -t filter -A OUTPUT -p tcp --dport 80 -j ACCEPT
sudo iptables -t filter -A OUTPUT -p tcp --dport 53 -j ACCEPT
sudo iptables -t filter -A OUTPUT -p udp --dport 53 -j ACCEPT
sudo iptables -t filter -A OUTPUT -p udp --dport 123 -j ACCEPT
```



◉ Vincent Saluzzo

Ingénieur informatique - Expert technique iOS

<http://www.vincentsaluzzo.com>

itelios

Unity 3D & PHP : c'est possible !

Le moteur de jeux Unity permet de créer des jeux en réseaux, des animations en temps réel... Et comme son format est ouvert, vous pouvez l'associer avec les langages du web pour le rendre plus contrôlable et manipulable ! La preuve par PHP... Unity 3D & PHP : c'est possible !

Avant de se lancer dans un projet de développement de jeu, il est important de connaître les possibilités qu'offre Unity 3D. Ce moteur peut être utilisé sur différentes plateformes, comme les réseaux sociaux ou les plateformes dédiées de jeux. Bien entendu, les langages disponibles sont principalement les langages compatibles avec les langages web comme PHP, Python, Flash... Grâce à son système ouvert, Unity 3D offre une gestion de plugins permettant de recevoir les scripts que vous aurez générés. Ainsi, vous améliorez votre jeu en offrant des informations et un intérêt supplémentaire via le web. Ces extensions vont permettre de mémoriser des informations d'archivage, de stockage d'informations concernant les joueurs, leurs identifiants, les différents scores, etc...

Environnement

Lorsque vous voulez communiquer en utilisant les facilités offertes par le web, vous devez prévoir différents environnements pour développer les scripts de votre projet. Un environnement se compose d'un serveur AMP, c'est à dire Apache, MySQL, PHP, ou, tout simplement utiliser un des environnements embarqués disponibles dans votre système d'exploitation : EasyPHP, Wampserver, Xamp, Mamp.... Et bien entendu, vous devez installer votre logiciel.

L'API

L'application (API) propose de nombreuses fonctionnalités disponibles pour la plateforme Unity. L'article se penchera plus précisément sur quelques fonctions de scripts PHP pour le web comme :

- ▮ Identifier les utilisateurs
- ▮ Afficher un journal des scores

Identifier les utilisateurs

Quel que soit le mode de recherche ou d'interrogations de la base de données, vous devez identifier les utilisateurs.

Pour cela, la connexion avec un jeu doit suivre quelques règles importantes pour retourner des informations provenant du web. Cette procédure s'effectue avec l'API du moteur Unity ; avec cela, vous avez à disposition de nombreuses informations sans être obligé de rester connecter...

Nous allons voir :

- ▮ Création d'une base de données
- ▮ Vérification d'authentification
- ▮ Les différentes étapes de procédure

Création d'une base de données

Pour afficher la liste des utilisateurs, vous devez posséder une base de données qui peut se composer de la façon suivante : créez une table 'user' avec les noms et les mots de passes de vos joueurs.

```
CREATE TABLE `user` (
  `id` INT( 11 ) NOT NULL AUTO_INCREMENT PRIMARY KEY ,
  `name` VARCHAR( 30 ) NOT NULL ,
  `password` VARCHAR( 50 ) NOT NULL
) ENGINE = innodb;
```

Pour notre exemple, nous allons insérer un utilisateur exemple comme 'programmez' avec comme mot de passe 'MD5'.

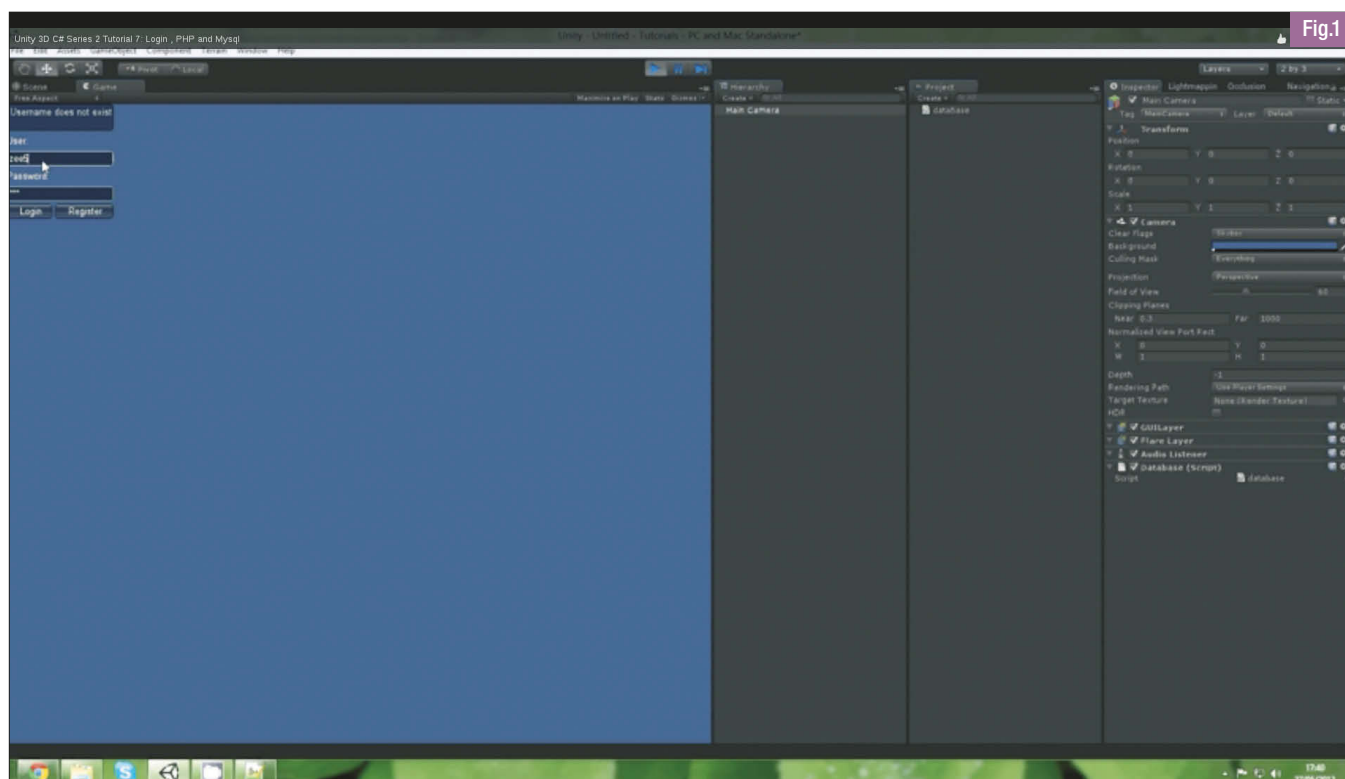


Fig.1

```
INSERT INTO `user` ( `id` , `name` , `password` )
VALUES ( NULL , 'programmez', MD5( 'programmez' ) );
```

Bien entendu, vous devez protéger le mot de passe par un système plus évolué.

Vérification d'authentification

Pour vérifier l'authentification de la personne lors du jeu, et afficher le score via le web, l'étape s'effectue en contrôlant la sécurité du login et du mot de passe que nous effectuons à travers différentes fonctions. Effectuez dans un premier temps des tests sur le login. Pour ce faire, l'utilisation des expressions régulières permet de régler les soucis de saisie. Ensuite, la suppression de valeur NULL se passe par la fonction `strip_tags`.

Enfin, si les valeurs sont acceptées, la fonction retourne la valeur de la variable avec une protection supplémentaire au niveau de l'injection SQL :

```
<?php
function securite_login($sql, $formUse = true)
{
    $sql = preg_replace("/(from|select|insert|delete|where|drop
table|show tables|'|#|\\*|--|\\\\\\\\)/i","", $sql);
    $sql = trim($sql);
    $sql = strip_tags($sql);
    if(!$formUse || !get_magic_quotes_gpc())
        $sql = addslashes($sql);
    return $sql;
}
?>
```

Ensuite, pour sécuriser le mot de passe, vous pouvez effectuer la même opération que précédemment avec la particularité d'ajouter un cryptage MD5 au mot de passe :

```
<?php
function securite_password($sql, $formUse = true)
{
    $sql = preg_replace("/(from|select|insert|delete|where|drop
table|show tables|'|#|\\*|--|\\\\\\\\)/i","", $sql);
    $sql = trim($sql);
    $sql = strip_tags($sql);
    if(!$formUse || !get_magic_quotes_gpc())
        $sql = addslashes($sql);
    $sql = md5(trim($sql));
    return $sql;
}
?>
```

Connexion

La connexion vers le serveur de la base de données que vous avez installée précédemment est indispensable :

```
<?php

$host = "votreServeur";
$user = "login";
$password = "password";
$dbname = "nomBaseDeDonnees";
```

```
$cnx=mysqli_connect($host, $user, $password,$dbname) or die
("Echec de la connexion : ". mysqli_connect_error());

?>
```

La fonction montre l'utilisation des fonctions MySQLi

Les différentes étapes

Lorsque vous souhaitez afficher les informations venant de votre base de données, certaines opérations sont nécessaires pour respecter la procédure conseillée par l'éditeur :

```
<?php

$phpHash = "hashcode"; // code du jeu

$login = securite_login($_POST['login']);
$pass = securite_password($_POST['pass']);
$unityHash = securite_login($_POST['hashForm']);

?>
```

Lorsque les données (login et password) ont été contrôlées et sécurisées, vous devez interroger la base de données pour vérifier si les données saisies sont exactes. Pour effectuer l'opération, suivez une suite logique de tests :

- Si le login et le mot de passe ne sont pas vides
- Si la clef de sécurité correspond bien au jeu en cours d'utilisation
- Vous devez récupérer les informations de l'utilisateur
- Vérifier si le mot de passe est bon par rapport à l'utilisateur

L'ensemble de ces étapes est nécessaire, elles permettent de garantir que le joueur est la bonne personne. Après, on peut passer à la suite.

```
<?php
if(!$login || !$pass)
{
    echo "login ou password ne peuvent être vides";
} else {
    if ($unityHash != $phpHash)
    {
        echo "Le code de HASH est différent de celui qui est défini";
    }
    else
    {
        $sql = "SELECT * FROM user WHERE name = ' " . $login . "'";
        $result = mysqli_query($cnx, $sql);

        if (!$result) die ("Problème : " . mysqli_error($cnx));

        if($result->num_rows != 0)
        {
            $datas = mysqli_fetch_array($result);
            if(!strcmp($pass, $datas["password"]))
            {
                echo "connexion réussie";
            }
            else
            {
                echo "Erreur dans le Login et/ou le mot de passe";
```

```

    }
}
else
{
    echo "vous ne possédez pas de compte";
}

}

}

// Close mySQL Connection
mysqli_close();
?>

```

Une petite particularité concerne la récupération des valeurs, car il faut utiliser les fonctions \$_POST

Afficher un journal des scores

A partir de l'identification, vous pouvez afficher un journal des scores des joueurs, ce qui permet d'avoir un plus à votre jeu. Pour cela, nous nous appuierons sur le chapitre précédent.

Base de données

Vous devez posséder une base de données et quelques données, comme :

```

CREATE TABLE IF NOT EXISTS `highscores` (
  `jeu` varchar(20) NOT NULL,
  `joueur` varchar(20) NOT NULL,
  `score` int(11) NOT NULL
) ENGINE=InnoDB DEFAULT CHARSET=latin1;

```

et avec quelques données pour chaque ligne :

```

INSERT INTO `highscores` (`jeu`, `joueur`, `score`) VALUES
('game', 'player1', '150000');

```

Le code

Pour que votre page web puisse communiquer avec votre jeu, vous devez réaliser un script PHP de récupération des scores et les enregistrer dans votre base de données. Pour réaliser ce script, vous récupérez les valeurs courantes en les échappant pour procéder de la manière suivante :

```

<?php
$jeu = $_POST['game'];
$player=$_POST['playerjoueur'];
$score=$_POST['score'];

$jeu      = mysqli_real_escape_string($cnx,$jeu);
$player   = mysqli_real_escape_string($cnx,$player);

```

```

localhost/unity/programmez/highscore/scoreboard.php

game-player1-150000
game-player2-240000
game-player3-150000
game-player4-50000
game-player1-10000
game-player6-100
--0
--0
--0

$score      = mysqli_real_escape_string($cnx,$score);

?>

```

L'insertion de ces valeurs s'effectue ainsi :

```

<?php

require_once ('config.inc.php');
$sql = "INSERT INTO `highscores` (jeu, joueur, score) VALUES
('$jeu', '$player', '$score')";

mysqli_query($cnx,$sql );

?>

```

Cette fonction va ajouter le nom du jeu, le nom du joueur et son score. La méthode suivante permet de sauvegarder tous les scores pour en afficher un classement dans une page HTML :

```

<?php
$result = mysqli_query($cnx,$select);
while ($row = mysqli_fetch_array($result) )
{
    echo $row['jeu'] . " - " . $row['joueur'] . " - " . $row
['score'] . "<br>";
}

?>

```

Le résultat sera affiché dans votre page internet.

Conclusion

Avec Unity 3D, grâce à la possibilité de mettre en place des scripts PHP, vous obtenez un jeu ouvert vers le web : les joueurs peuvent profiter d'un univers de jeu tourné vers le monde via Internet.

Christophe Villeneuve

Consultant IT pour Neuros, auteur du livre « PHP & MySQL-MySQLi-PDO, Construisez votre application », aux Éditions ENI. Rédacteur pour WebRIVER, membre des Teams DrupalFR, AFUP, LeMug.fr, Drupagora, PHPTV..



100%
cloud
computing

Durant les deux derniers mois, nous avons fait connaissance avec Guava. Nous avons vu en quoi la bibliothèque permet d'améliorer le code de nos programmes Java. Nous avons découvert les immutables, et nous nous sommes initiés à la programmation fonctionnelle. Dans la foulée, nous avons utilisé plusieurs nouveaux types de collections très pratiques au quotidien. À ce stade, vous devriez déjà être convaincu de l'intérêt de cette fantastique librairie. Si ce n'est pas le cas, ce troisième et dernier volet, placé sous le signe des performances, devrait finir de vous séduire.

CharMatcher

Commençons avec l'un des objets que l'on manipule le plus en Java : le String. On doit en changer la case, supprimer les espaces, remplacer des caractères isolés ou des patterns, extraire des sous-chaînes, etc. Or, toutes ces opérations ne sont pas évidentes à programmer proprement. De manière générale, on peut presque toujours les décomposer en deux temps : 1) trouver les caractères puis 2) les traiter. C'est d'ailleurs la première étape qui est la plus difficile, bien que cela ne se voit pas grâce à Guava :

```
String sansChiffre = CharMatcher.DIGIT.removeFrom("AB12cd34");
// ABcd
String sansLettre = CharMatcher.JAVA_LETTER.removeFrom("AB12cd34");
// 1234
String sansAscii = CharMatcher.ASCII.removeFrom("AB12cd34");
// vide
```

Dans l'exemple, on supprime les caractères qui correspondent. Il est bien entendu possible de cumuler plusieurs « matchers » pour sélectionner, par exemple, les chiffres et les majuscules :

```
String sansChiffreMaj = DIGIT.or(JAVA_UPPER_CASE).removeFrom(
    "AB12cd34"); // cd
```

Une fois qu'on sait détecter la position d'un pattern, on peut employer une des opérations offertes par la librairie. Précisons qu'elles sont loin d'être triviales :

```
int nb = DIGIT.countIn("AB12cd34"); // 4
String collapse = DIGIT.collapseFrom("AB12cd34", '*'); //
AB*cd*
String replace = DIGIT.replaceFrom("AB12cd34", "-+"); // AB-
+-+cd-+-+
int index = DIGIT.indexIn("AB12cd34"); // 2
```

“ Détecter un pattern est bien plus complexe que le traiter ”

Mais le point le plus difficile, à propos duquel de nombreux développeurs (pour ne pas dire tous) s'arrachent les cheveux, est l'encodage des caractères (charset). Java oblige d'ailleurs à traiter d'éventuelles exceptions. C'est dire combien cela peut être pénible :

```
String s = "AB12cd34";
try {
    byte[] bytes = s.getBytes("UTF-8"); // UnsupportedEncodingException: UTF-8
} catch (Exception e) {
    // traitement
}
```

Il est simple de se tromper de format (ici « UFT » au lieu de « UTF »), même si les dernières versions du JDK sont plus compréhensibles que par le passé, en particulier lorsque la valeur est injectée depuis un référentiel de propriétés. Avec Guava, vous allez aimer l'emploi de constantes :

```
byte[] bytes = s.getBytes(Charsets.UTF_16BE); // [B@882c01f
```

Avec les I/O

En Java, lorsqu'on travaille avec des fichiers, et plus généralement avec des « streams », on doit faire très attention à la fermeture des ressources. La bonne pratique consiste à la réaliser dans un bloc « finally ». Prenons la copie de fichier à titre d'illustration. Attention, ça pique les yeux :

```
public static boolean copie(String from, String to) {
    boolean ok = false;
    File f1 = new File(from);
    File f2 = new File(to);

    FileInputStream fis = null;
    FileOutputStream fos = null;

    try {
        fis = new FileInputStream(f1);
        fos = new FileOutputStream(f2);

        byte buffer[] = new byte[512 * 1024];
        int nb;
        while ((nb = fis.read(buffer)) != -1) {
            fos.write(buffer, 0, nb);
        }
        ok = true;
    } catch (FileNotFoundException e) {
        ...
    } catch (IOException e) {
        ...
    } finally {
        try {
            fis.close();
        } catch (Exception e) {
            ...
        }
        try {
            fos.close();
        } catch (Exception e) {
            ...
        }
    }
    return ok;
}
```

Une très grosse partie de ce code est purement technique. Le fonctionnel est donc perdu dans la masse. Guava dispose heureusement de méthodes qui changent la donne, épargnant notamment aux développeurs de coder un « buffer » :

```
try {
    ...
    ByteStreams.copy(fis, fos);
    ok = true;
} catch (FileNotFoundException e) {
    ...
} catch (IOException e) {
    ...
} finally {
    try {
        Closeables.close(fis, !ok);
    } catch (IOException e) {
        ...
    }
    try {
        Closeables.close(fos, !ok);
    } catch (IOException e) {
        ...
    }
}
```

L'utilisation de « `Closeables.close` », couplée avec le second paramètre (qui indique si la copie a été validée) permet d'éviter de polluer le code avec un test de nullité supplémentaire. Notez qu'on peut alors préciser une exception IO.

“ Les fournisseurs font tout le travail technique ”

Si on veut encore se simplifier la tâche, on pourra même utiliser des « fournisseurs » qui feront tout le travail à notre place :

```
try {
    ByteStreams.copy(Files.newInputStreamSupplier(f1), Files.newOutputStreamSupplier(f2));
    ok = true;
} catch (IOException e) {
    ...
}
```

Ces « fournisseurs » simplifient également la lecture des fichiers, dont je vous épargne la version en Java standard :

```
public static String lire(String from) {
    File f = new File(from);
    String s = null;
    try {
        s = CharStreams.toString(Files.newReaderSupplier(f, Charsets.ISO_8859_1));
    } catch (IOException e) {
        ...
    }
    return s;
}
```

Et on peut faire encore mieux à l'aide de la classe utilitaire « Files » :

```
s = Files.toString(f, ISO_8859_1);
```

Vous pensez qu'on ne peut plus faire (encore) mieux ? Détrompez-vous, car on va pouvoir réaliser des traitements directement pendant la lecture

du fichier. Voici par exemple comment filtrer les lignes de commentaires (commençant par dièse) d'un fichier de configuration :

```
public static List<String> lireConfiguration(String from) {
    File f = new File(from);

    final LineProcessor<List<String>> lp = new LineProcessor<List<String>>() {
        ImmutableList.Builder<String> builder = ImmutableList.builder();

        @Override
        public List<String> getResult() {
            return builder.build();
        }

        @Override
        public boolean processLine(String ligne) throws IOException {
            if (!ligne.startsWith("#")) {
                builder.add(ligne);
            }
            return true;
        }
    };

    List<String> lignes = null;
    try {
        lignes = Files.readLines(f, ISO_8859_1, lp);
    } catch (IOException e) {
        ...
    }
    return lignes;
}
```

Primitifs

Dans le premier épisode de cette série, nous avons vu qu'il est simple de créer des listes d'objets préremplies grâce à Guava. Il est possible de faire de même avec les primitifs :

```
List<Double> list = Doubles.asList(1.41, 1.73, 2, 2.23, 2.44, 2.64);
double[] tab1 = Doubles.toArray(list);
```

Comme d'habitude avec Guava, on pourra réaliser un ensemble d'opérations dont voici une courte sélection :

```
double[] tab2 = { 1.618, 3.1416 };
double[] tab3 = Doubles.concat(tab2, tab1); // 1.618, 3.1416, 1.41, 1.73, 2.0, 2.23, 2.44, 2.64
Double PI = 3.1416;
boolean isPiIn = Doubles.contains(tab3, PI); // true
int position = Doubles.indexOf(tab3, PI); // 1
```

On pourra aussi extraire les éléments les plus petits et les plus grands de la liste, très facilement :

```
double min = min(tab3); // 1.41
double max = max(tab3); // 3.1416
```

Et ce n'est pas plus difficile d'avoir une représentation de la liste sous forme de String, en spécifiant son propre séparateur :

```
String s = join("-", tab3); // 1.618-3.1416-1.41-1.73-2.0-2.23-2.44-2.64
```

Calculs

Lorsque vous réalisez des opérations mathématiques, il vous arrive de dépasser les capacités des types que vous utilisez sans vous en rendre compte. En effet, la JVM retombe toute seule sur ses pieds, en silence :

```
int n1 = Integer.MAX_VALUE; // 2147483647
int n2 = n1 + 10; // -2147483639
```

Dans l'exemple ci-dessus, la JVM a simplement bouclé sur les valeurs négatives. Autant dire que vos calculs financiers risquent de vous attirer des ennuis... Il n'y a pas grand-chose à faire pour rattraper le tir, mais on voudrait, au moins, avoir une notification pour savoir que quelque chose s'est mal passé. C'est ce que vont nous permettre les objets « IntMath », « LongMath » et « BigIntegerMath », dont les noms indiquent leurs types cibles :

```
int n3 = IntMath.checkedAdd(n1, 10); // ArithmeticException: overflow
```

Dans vos programmes scientifiques, vous devez aussi (souvent) calculer des racines carrées, des exponentiels, des logarithmes variés, etc. Je vous mets au défi de programmer de telles fonctions sans l'aide d'une bonne librairie :

```
int puis = pow(2, 5); // 32
int log10_5 = log10(5, HALF_DOWN); // 1
int log2_5 = log2(5, HALF_EVEN); // 2
double log2_7 = DoubleMath.log2(7); // 2.807354922057604
int racine5 = sqrt(5, UNNECESSARY); // ArithmeticException: mode
// was UNNECESSARY, but rounding was necessary
int racine5b = sqrt(5, FLOOR); // 2
boolean isInt = isMathematicalInteger(racine5b); // true
```

Notez que le dernier appel renvoie « true », car le second calcul de la racine carrée de 5 a été arrondi à l'entier inférieur. Guava propose en effet d'employer différents types d'arrondis dont vous avez une sélection dans l'exemple ci-dessus. La bibliothèque permet également certains calculs algébriques, comme le PGCD (Plus Grand Commun Diviseur) ou le coefficient binomial (dans le triangle de Pascal), qui vous rappelleront vos cauchemars post Bac, mais qui, ne vous en déplaise, sont ultra utilisés dans l'industrie :

```
int pgcd = gcd(6, 15); // 3
int factoriel = factorial(5); // 120
int coefBinomial = binomial(5, 3); // 10
```

Optional

Comme nous l'avons vu le mois dernier, la gestion de la nullité est au cœur des préoccupations des créateurs de Guava. Or, pour diverses raisons, nos programmes Java doivent régulièrement travailler avec des valeurs nulles et « blinder » les algorithmes en conséquence, ce qui peut devenir inutilement lourd. À titre d'illustration, imaginons une méthode qui cherche une personne en base de données, par son nom :

```
public Personne findByName(String name) {
    String query = "select * from personne where name = ?";
    return DAO.execute(query, name);
}
```

Il est parfaitement admissible que cette recherche ne remonte aucune donnée. Le bloc de code appelant devra donc vérifier la valeur de retour :

```
Personne sophie = findByName("Sophie");
if(sophie != null) {
    traiter(sophie);
}
```

Ce test, purement technique, devra d'ailleurs se répéter dans une grande partie de la « stack » d'appel et la polluer. Guava propose d'encapsuler la valeur dans un « wrapper » qui, lui, par définition, ne sera jamais nul :

```
Optional<Personne> wrapper = Optional.of(sophie);
transmettre(wrapper);
```

“des valeurs non nulles permettent d'activer des optimisations”

Ce « static factory » lève une exception lorsqu'on lui passe une valeur nulle. Or, à ce point, on préférera aller au bout des choses en s'épargnant de faire le moindre test, y compris à l'instanciation :

```
Optional<Personne> wrapper = Optional.fromNullable(sophie);
```

Dans certains cas particuliers, on pourra d'ailleurs créer un wrapper vide. À quoi peut bien servir un wrapper vide ? Tout simplement à indiquer l'absence de donnée, mais en manipulant un objet non nul :

```
Optional<Personne> vide = Optional.absent();
```

Ce n'est finalement que lorsque l'on aura besoin de connaître la valeur que l'on utilisera les méthodes « isPresent » et « get » qui permettent respectivement de savoir si le wrapper est vide et, sinon, de récupérer le contenu :

```
if(wrapper.isPresent()) {
    Personne p = wrapper.get();
    traiter(p);
}
```

Il ne faut jamais faire appel à « get » si le wrapper est vide, sous peine de s'en tirer avec une exception à traiter. Mais on voit bien qu'un appel systématique à « isPresent » ne répond pas au besoin d'un code épuré. On préférera alors employer « orNull » qui renvoie tout simplement une valeur nulle lorsque le wrapper est vide :

```
Personne p = wrapper.orNull();
```

Et tant qu'on y est, on lui préférera même la méthode « or » à laquelle on peut indiquer une valeur par défaut, de remplacement, qui sera renvoyée en cas d'optional vide :

```
Personne p = wrapper.or(new Personne("John", "Doe"));
```

Manipuler des variables non nulles a un impact fort sur la lisibilité du code, mais également sur le « plan d'exécution » de la JVM car des optimisations pourront être activées. Ce point est très important et fait même partie des concepts essentiels d'autres langages, comme Scala qui possède son « Option ». La version 8 de Java, prévue pour mars 2014, devrait également introduire un Objet « Optional » ayant un comportement très proche de celui de Guava.

StopWatch

Quitte à parler de performances, il peut être utile de mesurer celles d'un bloc de code. Pour cela, le mode opératoire classique est de noter la date, en millisecondes, avant et après l'exécution du bloc. Il suffit de faire la différence pour avoir la durée :

```
final long date1 = System.currentTimeMillis();

// traitement a mesurer...
```



```
final long date2 = System.currentTimeMillis();
final long duree = date2 - date1;
```

S'il y a plusieurs blocs à chronométrer, il faut reproduire l'opération autant de fois que nécessaire. Guava propose une version sensiblement plus élégante :

```
final Stopwatch sw = new Stopwatch();
sw.start();

// traitement a mesure...

final long duree = sw.elapsed(MILLISECONDS);
```

On peut spécifier la précision souhaitée à la méthode « `elapsed` », de la journée à la nanoseconde, bien qu'une précision si fine soit douteuse en Java :

```
final long dureeJ = sw.elapsed(DAYS);
final long dureeNs = sw.elapsed(NANOSECONDS);
```

Caches

Les problématiques de caches devraient également vous intéresser tant c'est un point d'optimisation incontournable. Vous devez mettre des données en cache à chaque fois que vous les chargez de manière régulière et que vous savez qu'elles ne vont pas changer pendant un laps de temps déterminé.

Par exemple, vous pouvez affirmer que l'âge de Sophie ne changera pas avant demain. Ceci est d'autant plus important si la demande est coûteuse, comme c'est le cas avec une recherche en base de données ou à l'aide d'un Web Service sur l'Internet.

Classiquement, en Java, il suffit de stocker l'âge de Sophie dans une variable. Si la valeur est nulle, ça signifie que c'est la première fois qu'on interroge le service et on doit lancer la recherche. Sinon on renvoie la valeur trouvée :

```
private Integer ageDeSophie;

public Integer getAgeDeSophie() {
    if (ageDeSophie == null) {
        ageDeSophie = webService.rechercheLongueAgeSurInternet("Sophie");
    }
    return ageDeSophie;
}
```

Cette technique, nommée « *mémoïsation* », est celle à laquelle les développeurs pensent généralement en premier. Elle n'est pourtant pas exempte de défauts, à commencer par le fait que les données, une fois chargées, ne seront plus jamais rafraîchies tant que le programme ne sera pas relancé. À l'aide de Guava, on va employer une approche plus efficace :

```
private Supplier<Integer> ageDeSophieCache = Suppliers.memoize(
    new Supplier<Integer>() {
        public Integer get() {
            return webService.rechercheLongueAgeSurInternet("Sophie");
        }
    });

public Integer getAgeDeSophie() {
    return ageDeSophieCache.get();
}
```

On pourra indiquer une période de validité à l'aide de la méthode « *memoizeWithExpiration* » en précisant une unité de durée dont la précision va du jour à la nanoseconde :

```
private Supplier<Integer> ageDeSophieCache = Suppliers.memoizeWithExpiration(
    new Supplier<Integer>() {
        ...
    }, 12, TimeUnit.HOURS);
```

La plupart du temps, toutefois, on ne s'intéresse pas qu'à une seule valeur, mais à toute une série. Là encore, la technique classique se base sur la nullité éventuelle des données, ces dernières étant stockées dans des maps :

```
private Map<String, Integer> ages;

public Integer getAge(String prenom) {
    Integer age = ages.get(prenom);
    if (age == null) {
        age = webService.rechercheLongueAgeSurInternet(prenom);
    }
    return age;
}
```

De nouveau, Guava va être plus performant, tout en gérant la durée de validité des données :

```
private LoadingCache<String, Integer> ageCache = CacheBuilder.
    newBuilder()
        .expireAfterWrite(12, TimeUnit.HOURS)
        .build(new CacheLoader<String, Integer>() {
            public Integer load(String prenom) {
                return webService.rechercheLongueAgeSurInternet(prenom);
            }
        });

public Integer getAge(String prenom) {
    try {
        return ageCache.get(prenom);
    } catch (ExecutionException e) {
        // traitement
    }
}
```

Dans l'exemple, vous remarquez qu'on a précisé que la durée d'expiration commence après l'insertion (`write`) dans le cache. On pouvait aussi demander que le chronomètre démarre à partir de la dernière lecture, ce qui est très différent. Et il sera bien entendu possible de forcer le rechargement à la demande.

“*un cache en mémoire, non sérialisé*”

Il est également possible de préciser le nombre maximum d'éléments qu'on souhaite conserver dans le cache. En effet, celui-ci risque de grossir démesurément, faisant exploser l'occupation mémoire par la même occasion :

```
private LoadingCache<String, Integer> ageCache = CacheBuilder.
    newBuilder()
        .maximumSize(2000)
        .expireAfterWrite(12, HOURS)
        .build(new CacheLoader<String, Integer>() {
```

```
...
});
```

Il existe d'autres bibliothèques de cache, à commencer par la plus utilisée : Ehcache. La particularité de la solution Guava est qu'elle travaille exclusivement en mémoire, contrairement à Ehcache qui sérialise les données. Le cache de Guava sera donc plus rapide, mais il ne sera pas persisté. Et ce n'est pas non plus un cache distribué. Pour ce dernier point, vous pouvez jeter un œil du côté de Terracotta.

Hash et Filtres probabilistes

En octobre, nous avons vu que Guava simplifie l'écriture de la méthode « hashCode ». La librairie dispose, plus simplement, d'utilitaires pour générer des « hash » personnalisés et connaît les principaux algorithmes du marché : MD5, Murmur3 sur 32 ou 128 bits, Sha1, Sha256, Sha512, etc.

```
HashFunction hf = Hashing.md5();
HashCode hc = hf.newHasher()
    .putDouble(92.5)
    ...
    .putString("Thierry")
    .putString("Leriche")
    .hash();
byte[] bytes = hc.asBytes();
```

Guava propose également un algorithme, nommé « Good Fast Hash », dont le principe repose sur un constat simple : les développeurs utilisent « MD5 » ou « Murmur3 » parce qu'ils ont besoin d'un algorithme. Or n'importe quel algorithme ferait l'affaire tant qu'il est rapide, ce qui est justement l'objectif de GFH. Il est aussi possible de spécifier la décomposition générique d'un objet à l'aide d'un « funnel » :

```
Funnel<Personne> funnel = new Funnel<Personne>() {
    @Override
    public void funnel(Personne personne, PrimitiveSink sink) {
        sink.putDouble(personne.getPoids())
        ...
        .putString(personne.getNom()) //
        .putString(personne.getPrenom());
    }
};
```

Pour finir, voici une dernière petite optimisation pour vous aider à chercher des éléments dans vos collections, surtout si elles sont volumineuses. Imaginons que nous ayons une liste contenant les habitants de toute la France, soit un peu plus de 65 millions d'items (65 585 857 au 1er janvier 2013) :

```
int NB = 65585857;
List<Personne> habitants = Lists.newArrayListWithCapacity(NB);
```

Lorsque l'on veut savoir si une personne, par exemple Sophie, est présente dans la liste, on utilise la méthode « contains ». Cette dernière doit parcourir la liste jusqu'à trouver Sophie. Elle renvoie alors une réponse positive. Mais elle doit parcourir l'intégralité de la liste lorsque Sophie est absente avant de pouvoir répondre par la négative, ce qui peut être extrêmement long :

```
Personne sophie = new Personne("Sophie", "Dupont", Genre.FEMME);
boolean isIn = habitants.contains(sophie);
```

Guava permet de contourner cet inconvénient à l'aide d'un filtre probabiliste qu'on va alimenter en même temps que la liste des habitants. Le surcoût en mémoire ne sera pas très important, car c'est seulement le hash de « chaque » élément que l'on utilise :

```
BloomFilter<Personne> bloom = BloomFilter.create(funnel, NB, 0.01);
Personne p = ...
habitants.add(p);
bloom.put(p);
```

On va se servir du « bloom » pour déterminer la présence de Sophie dans la liste, ce qui sera quasi instantané puisque l'algorithme se contente de tester les « hash » enregistrés, à l'aide d'un mécanisme binaire :

```
boolean maybeIn = bloom.mightContain(sophie);
```

Il est important de noter que « mightContain » renvoie parfois des faux positifs. Dans l'exemple, cela se produit avec la probabilité de « 0,01 » (ie. 1%) spécifiée à l'instanciation. En revanche, les réponses négatives sont toujours exactes, ce qui tombe bien puisque c'est généralement l'absence qu'on souhaite vérifier et non la présence.

Notez également que l'exemple présenté est un peu biaisé. Aucun programme ne va essayer de charger 65 millions d'éléments complexes en mémoire ; ça exploserait. Les habitants vont donc, en réalité, être sérialisés sur le disque ou en base de données. Seul le « bloom » restera en mémoire. La différence de performance entre la recherche de Sophie dans un fichier (jusqu'à plusieurs minutes) et dans le filtre probabiliste (quelques millisecondes) est d'autant plus conséquente. D'ailleurs, un peu pour les mêmes raisons, vous retrouverez un mécanisme relativement similaire dans les bases NoSql distribuées.

Conclusion

De nombreux développeurs à travers le monde considèrent que Guava est devenu une dépendance indispensable. Cette librairie est principalement utilisée dans le monde JEE. Il existe une déclinaison pour GWT (Google Web Toolkit) et même un rétroportage vers Java 5, permettant aux projets plus anciens d'en profiter.

À travers ces trois numéros consacrés à Guava, nous avons fait le tour des fonctionnalités clés de la librairie. Toutefois, nous n'avons fait qu'effleurer la surface de ce qu'elle propose. La plupart de ces fonctionnalités sont bien plus riches ; elles proposent des options complémentaires inédites.

En outre, Guava possède de nombreux autres utilitaires que nous n'avons tout simplement pas la place de présenter dans ces pages : tris, concurrence, réseaux, bus, réflexion, etc.

Maintenant que vous avez des bases solides pour comprendre le fonctionnement et la philosophie de Guava, je vous encourage à continuer votre découverte de cette fabuleuse bibliothèque sur le Web. Prenez le Wiki officiel comme point de départ.

Consultez les sources sur « GitHub ». Elles sont gratuites, bien écrites, claires et intelligentes. Suivez les nombreuses discussions sur « Stack Overflow », car Guava continue d'évoluer, de s'enrichir et de s'améliorer. Enfin, vous trouverez quelques pointeurs et compléments sur « icauda.com ». J'espère que ce sera, pour vous, une révélation comme ce fut le cas pour moi lorsque j'ai découvert Guava il y a quelques années.

Quelques liens

Guava : <https://code.google.com/p/guava-libraries/>

Eh cache : <http://ehcache.org/>

Terracotta : <http://terracotta.org/>

Java 8 : <https://jdk8.java.net/>

Icauda : <http://www.icauda.com>

🔴 Thierry Leriche-Dessirier

Formateur Java indépendant et Professeur à l'ESIEA

Twitter : @thierryleriche



Magento : comment calculer les frais de port ?

Magento propose par défaut à l'internaute de calculer les frais de port depuis le panier en renseignant le pays, la région et le code postal de l'adresse de livraison. Ce système n'est pas forcément judicieux, car on observe souvent que le prix des frais de port pour un pays reste identique, quels que soient la région ou le code postal de livraison Fig.1.

L'objectif est de donner au client un aperçu du montant total de sa commande – frais de port inclus – sans qu'aucune action ne soit nécessaire de sa part. Dès le panier, l'internaute aura ainsi connaissance du montant total de sa commande. Il pourra par la suite, s'il le souhaite, modifier ses frais de port en choisissant une autre option (une livraison express par exemple). Magento a besoin au minimum du pays de livraison afin d'obtenir la liste des frais de port à proposer. Si le site ne permet la livraison que pour un seul pays, celui-ci sera alors associé automatiquement à l'adresse de livraison. Dans le cas d'un site e-commerce multi-pays, le pays peut être récupéré par un système de géolocalisation, par exemple celui proposé par le Webservice MaxMind.

Une fois le pays défini, il suffira d'appliquer des frais de port au choix, généralement l'option la moins onéreuse Fig.2.

Architecture du module

Un simple observer nous permet d'associer automatiquement les frais de port à la commande.

- app/etc/modules/Altima_Autoquote.xml
- app/local/Altima/Autoquote/etc/config.xml
- app/local/Altima/Autoquote/Model/Observer.php

Développement du module

Le fichier de configuration présente 2 observers, le premier **checkout_cart_product_add_after** permet l'association des frais de port à la commande après ajout d'un produit au panier. Le deuxième, **checkout_cart_update_items_after** exécutera la même action lors d'une mise à jour du panier. Code complet sur le site www.programmez.com. La méthode **addDefaultShipping** de l'observer doit se charger d'associer le pays et les frais de port par défaut. Dans notre exemple, le pays par défaut sera la France (FR) uniquement si le pays du client n'est pas connu (par le biais d'une commande précédente par exemple).

Pour définir le pays et le mode de livraison associés à l'adresse, il nous faut récupérer l'objet quote. Dans notre exemple, le code des frais de port sera « `tablerate_tablerate` ». Le code suivant permet l'association du pays et des frais de port :

```
// Country Id
$country = 'fr';
$shipping = 'tablerate_tablerate';

// Retrieve Quote
$quote = Mage::getSingleton('checkout/session')->getQuote();

// Set Default Country for Billing
$quote->getBillingAddress()->setCountryId($country);

// Set Default Country and shipping Method for Shipping
$quote->getShippingAddress()
->setCountryId($country);
```

Fig.1

Fig.2

```
->setShippingMethod($shipping)
->setCollectShippingRates(true)
->collectTotals();

// Save Quote
$quote->save();

// Reset Checkout
Mage::getSingleton('checkout/session')->resetCheckout();
```

Si le client est connecté il nous est possible de récupérer le code du pays de livraison de l'adresse de livraison par défaut. Ainsi, si le site propose une expédition multi-pays, il est par conséquent facile de proposer des frais de port par défaut :

```
var $_country = 'FR';

protected function getCountry() {
    $country = $this->_country;

    if (Mage::helper('customer')->isLoggedIn()) {
        $address = Mage::helper('customer')->getCustomer()->get
DefaultShippingAddress();
        if ($address) {
            $country = $address->getCountryId();
        }
    }

    return $country;
}
```

La classe « `Altima_Autoquote_Model_Observer` » du module ressemble pour finir à :

```
class Altima_Autoquote_Model_Observer
{

    /**
     * Quote
     *
     * @var null
     */
    var $_quote = null;
    /**
```



```

* Default Country Id
*
* @var string
*/
var $_country = 'FR';

/**
* Default Shipping
*
* @var string
*/
var $_shipping = 'tablerate_tablerate';

/**
* Set Default Country and Shipping method
*
* @param Varien_Event_Observer $observer
* @return Mage_Autoquote_Model_Observer
*/
public function addDefaultShipping(Varien_Event_Observer $observer)
{
    try {

        // Retrieve Country Id
        $country = $this->getCountry();

        // Set Default Country fot Billing
        $this->getQuote()->getBillingAddress()->setCountryId(
($country);

        // Set Default Country and shipping Method fot Shipping
        $this->getQuote()->getShippingAddress()
            ->setCountryId($country)
            ->setShippingMethod($this->_shipping)
            ->setCollectShippingRates(true)
            ->collectTotals();

        // Save Quote
        $this->getQuote()->save();

        // Reset Checkout
        $this->getCheckout()->resetCheckout();

    } catch (Mage_Core_Exception $e) {
        $this->getCheckout()->addError($e->getMessage());
    } catch (Exception $e) {
        $this->getCheckout()->addException(
            $e,
            Mage::helper('checkout')->__('Load customer quote error')
        );
    }

    return $this;
}

/**
* Retrieve Default Country
*

```

```

* @return string
*/
protected function getCountry() {
    $country = $this->_country;

    if(Mage::helper('customer')->isLoggedIn()) {
        $address = Mage::helper('customer')->getCustomer()->get
DefaultShippingAddress();
        if($address) {
            $country = $address->getCountryId();
        }
    }

    return $country;
}

/**
* Retrieve Checkout Session
*
* @return Mage_Checkout_Model_Session
*/
protected function getCheckout()
{
    return Mage::getSingleton('checkout/session');
}

/**
* Retrieve Quote
*
* @return Mage_Sales_Model_Quote
*/
protected function getQuote()
{
    if (empty($this->_quote)) {
        $this->_quote = $this->getCheckout()->getQuote();
    }
    return $this->_quote;
}
}

```

Pour finir, il nous reste à déclarer le module à Magento :

```

<?xml version="1.0"?>
<config>
    <modules>
        <Altima_Autoquote>
            <active>true</active>
            <codePool>local</codePool>
            <depends>
                <Mage_Checkout />
                <Mage_Shipping />
            </depends>
        </Altima_Autoquote>
    </modules>
</config>

```

Code complet sur www.programmez.com.

À vous de jouer !

 **Matthieu Vion** - *Leader Technique*
Magento

Trucs & astuces pour optimiser son **code c#**

Malgré les performances croissantes de nos ordinateurs, il est important d'offrir à nos projets du code performant pour ne pas nuire à l'expérience utilisateur. Je vous propose quelques astuces en C#.

À force de développer sur des projets X ou Y plus ou moins importants, j'ai souvent constaté 3 étapes majeures dans la réalisation d'un algorithme :

1. Faire en sorte que l'algorithme fonctionne.
2. Refactoriser le code.
3. Optimiser en vitesse et en consommation.

Pourquoi bien optimiser en vitesse?

N'êtes-vous pas frustré quand vous jouez à un jeu qui est réputé pour avoir quelques problèmes de saut à l'écran? Ce n'est pas forcément à cause de la puissance de votre ordinateur, mais simplement à cause d'une mauvaise optimisation du code. La machine n'arrive pas à effectuer tous les calculs « tellement il y en a », et par conséquent l'affichage du frame suivant est long à venir. Nous allons donc voir ici qu'il existe diverses manières de réduire le nombre de calculs de votre machine, aussi bien en algorithmique qu'en .net.

Tout d'abord, gardez en tête que la plupart du temps, moins il y a d'itérations, plus le code est rapide. Bien entendu, il y a quelques exceptions.

Les opérateurs

Le code suivant :

```
i += 1;
```

Est plus rapide que :

```
i = i + 1;
```

Utilisation de select case

Le code suivant :

```
switch (variable)
{
    case "Thomas":
        // itérations
        break;
    case "Matthieu":
        // itérations
        break;
    .....
    case "Paul":
        // itérations
        break;
    case "Richard":
        // itérations
        break;
}
```

Est plus rapide que :

```
if (variable == "Thomas") {
    // itérations
}
else if (variable == "Matthieu") {
    // itérations
}
```

```
}
.....
else if (variable == "Paul") {
    // itérations
}
else if (variable == "Richard") {
    // itérations
}
```

Utilisation de With (VB.Net seulement)

Afin de réduire le nombre d'appels à un objet, vous pouvez utiliser le mot clé With. Ainsi, le code suivant :

```
With TextBox1
    .Text = "Hello World"
    .IsReadOnly = True
End With
```

Est plus rapide que :

```
TextBox1.Text = "Hello World"
TextBox1.IsReadOnly = True
```

Car on fait appel à TextBox1 une seule fois.

Optimiser les boucles

En .net, les boucles Do While, While, For et Foreach ont la même vitesse d'exécution. A vous de bien choisir laquelle utiliser. Par exemple, si vous n'êtes pas obligé de parcourir tout un tableau, vous pouvez utiliser While ou Do While. Et au contraire, si vous devez appliquer un traitement à tous les éléments d'une liste, il sera préférable d'utiliser For ou Foreach.

Comme dit précédemment, il faut essayer de réduire le nombre d'itérations. Par exemple, le code suivant :

```
var result = new List<string>();
foreach (var s in array)
{
    if (s.Contains("Thomas"))
        result.Add(s);
}
```

Est plus rapide que :

```
var result = new List<string>();
for (var i = 0; i < array.Count; i++)
{
    var s = array[i];
    if (s.Contains("Thomas"))
        result.Add(s);
}
```

Car il y a une variable et une itération en moins. Nous verrons un peu plus loin qu'il existe encore plus rapide.

Évitez également les actions constantes récurrentes dans vos boucles. Par exemple, le code suivant :

```
var x = 6 * 7;
for (var i = 0; i < 100; i++)
    y = i + x;
```

Est plus rapide que :

```
for (var i = 0; i < 100; i++)
    y = i + (6 * 7);
```

Vous pouvez également gagner de la vitesse dans l'imbrication des boucles. Par exemple, le code suivant :

```
for (var i = 0; i < 10; i++) // 10 itérations
for (var j = 0; j < 1000; j++) { // 1000x10 itérations
// itérations
} // Total : 10010 itérations
```

Est plus rapide que :

```
for (var i = 0; i < 1000; i++) // 1000 itérations
for (var j = 0; j < 10; j++) { // 1000x10 itérations
// itérations
} // Total : 11000 itérations
```

Utilisation de Linq

Linq est une extension que je juge indispensable. Non seulement vos routines sont plus performantes, mais en plus ça prend beaucoup moins de lignes de codes. Par l'exemple, ci-dessus nous avons démontré que le code suivant, avec foreach, était plus rapide que le même code utilisant un for :

```
var result = new List<string>();
foreach (var s in array)
{
    if (s.Contains("Thomas"))
        result.Add(s);
}
```

En utilisant Linq, le code est plus petit, et encore plus rapide :

```
var result = array.Where(s => s.Contains("Thomas")).ToList();
// Je récupère chaque élément de la liste qui contient le
terme "Thomas", le convertit en "List<string>" et l'assigne
à "result".
```

Vous trouverez à la fin de l'article un petit benchmark qui compare les performances.

Bien typer ses variables

Il est important de bien choisir le type de ses variables (que ça soit dans un langage orienté objet ou non), pour des soucis de mémoire et de vitesse. Le premier conseil que je donnerais est d'éviter la résolution du type à l'exécution, autrement dit, éviter d'utiliser le type « object ». Ainsi, le code suivant :

```
var array = new List<int>();
for (var i = 0; i < 100; i++)
    array.Add(i);
```

Est plus rapide que :

```
var array = new List<object>();
for (var i = 0; i < 100; i++)
    array.Add(i);
```

Car, dans le second code, le .net Framework effectue une résolution afin d'assurer la compatibilité entre le type object et int, ce qui prend du temps. Je vous conseille également de privilégier, selon vos besoins, les types les moins « boxés », c'est-à-dire, les types ayant subi le moins d'héritage. Voici un exemple avec deux Class :

```
class A {
    public int X { get; set; }

    public A(int x) {
        X = x;
    }

    public string ToString() {
        return X.ToString();
    }
}

class B : A {
    public void Increment(int i) {
        X += i;
    }
}
```

Par définition, le code suivant :

```
var variable = new A(12);
var result = variable.ToString();
```

Est plus rapide que :

```
var variable = new B(12);
var result = variable.ToString();
```

Car dans le second code, on doit résoudre l'encapsulation du type B avec le type A. Dans cet exemple, comme nous n'avons pas besoin de la méthode « Increment », il est préférable d'instancier directement le type A plutôt que le type B (qui hérite de A).

Bien utiliser les listes et collections

Quand vous devez faire appel à des tableaux de valeurs, faites attention à bien choisir le type de tableau que vous allez utiliser : List? Collection? Array? Dictionary?

Ces différents tableaux font tous la même chose : stocker une liste de données en mémoire. Mais ils proposent des méthodes et des fonctions différentes. Il faut donc bien les choisir selon vos besoins. Regardons ici les cas où l'on souhaite juste stocker des données en mémoire :

Le code suivant :

```
var array = new List<KeyValuePair<string, int>>();
for (var i = 0; i < maxValues; i++)
    array.Add(new KeyValuePair<string, int>(i.ToString(CultureInfo.
InvariantCulture), i * i));
```

Est plus rapide que :

```
var array = new Dictionary<string, int>();
for (var i = 0; i < maxValues; i++)
    array.Add(i.ToString(CultureInfo.InvariantCulture), i * i);
```

On peut l'expliquer par le fait que KeyValuePair est une simple structure, tandis que Dictionary est une Class. Le traitement appliqué lors du stockage de la donnée est sûrement moins lourd. On peut également en déduire que le type List est plus rapide qu'un Dictionary.

Après plusieurs tests en Benchmark, suite à une idée de mon confrère Aloïs de Gouvello, j'ai également pu constater les résultats suivants :

1. Une `List<string>` est plus rapide qu'un `StringCollection` qui est plus rapide qu'une `Collection<string>`.
2. En revanche, une `List<double>` est plus rapide qu'une `Collection<double>` qui est plus rapide qu'un `DoubleCollection`.

Vous trouverez à la fin de l'article un petit Benchmark qui compare les performances.

Performances de l'interface

L'une des actions les plus longues à réaliser par un ordinateur est l'affichage à l'écran. Il faut donc essayer de limiter le nombre de rafraîchissements de notre interface à l'écran.

Prenons en exemple une `ListBox`. Si j'utilise le code suivant :

```
for (var i = 0; i < 100; i++)
    ListBox1.Items.Add(i.ToString());
```

Je vais rafraîchir l'interface de mon application 100 fois. On peut très vite perdre de précieuses millisecondes de temps d'exécution (surtout en WinForm). La solution est donc d'interdire le rafraîchissement le temps de remplir notre `ListBox`. Ainsi, le code suivant est bien plus rapide, car le rafraîchissement de l'interface est fait une seule fois, à la fin :

```
ListBox.BeginInit() // suspend le rafraîchissement automatique
for (var i = 0; i < 100; i++)
    ListBox1.Items.Add(i.ToString());
ListBox.EndInit() // reprend le rafraîchissement
```

On aurait également pu utiliser une variable annexe et la méthode « `AddRange` » qui nous permet d'ajouter directement une liste de données à la `ListBox`. Idem, le code suivant :

```
TextBox1.Text = "ThomasMatthieu";
```

Est plus rapide que :

```
TextBox1.Text = "Thomas";
TextBox1.Text += "Matthieu";
```

Car on rafraîchit l'interface une seule fois. Par conséquent, si vous êtes amené à ajouter du contenu à un `TextBox` dans une boucle par exemple, il sera préférable d'utiliser une variable annexe.

Utilisation des Task (développement parallèle)

Grâce aux `Task`, vous pouvez gagner plusieurs secondes d'exécution sur de longs processus. Une `Task`, ça consiste à exécuter un bout de code parallèlement au reste de votre programme (qui exécute les codes ligne par ligne). De ce fait, avec une `Task`, on peut exécuter une autre ligne de code « en même temps » que celle du programme principal. Deux lignes de code en une, donc une exécution théoriquement deux fois plus rapide. Vous trouverez à la fin de l'article un petit Benchmark qui contient ces fameux `Task`. Vous pourrez constater que les tests sont effectués, grâce aux `Task`, de manière asynchrone (c'est à dire, en même temps, et non l'un après l'autre).

Utilisation du mot clé Await

Nombreux sont ceux qui utilisent le mot clé « `await` » à tort. `Await` sert à « attendre » qu'une fonction qui s'exécute de manière asynchrone ait terminé son exécution. Utilisez le mot clé `await` uniquement si vous avez besoin de la valeur de retour de cette fonction, sinon, vous pouvez la laisser s'exécuter de manière asynchrone (donc sans `await`) et poursuivre l'exécution, ce qui vous fera gagner souvent beaucoup de temps d'exécution.

Limiter l'accès au disque

Vous aurez beau avoir un PC avec 4GHz, 32Go de RAM, si vous avez un disque dur qui fait du 5200tr/min, vous ne pourrez jamais profiter de toutes les performances de votre ordinateur. Alors, comment optimiser son programme en ce qui concerne les ressources physiques?

Je vous conseillerais deux principes :

1. Si vous avez une donnée dans un fichier qui est rarement utilisé par votre programme, il vaut mieux charger cette donnée (et donc avoir accès au disque dur) uniquement au moment où votre programme en a besoin.
2. Si vous avez une donnée dans un fichier qui est souvent, voire tout le temps, utilisé par votre programme, il vaut mieux charger cette donnée au démarrage de votre application, et de préférence dans un `Task` si c'est possible. Ainsi, votre programme est prêt à être utilisé, et en arrière-plan, il continue à charger des données du disque dur.

Utilisation des Try ... Catch

L'une des plus grosses innovations apportées par Bjarne Stroustrup, inventeur du C++, est la gestion des exceptions. On retrouve bien évidemment cette gestion dans tous les langages orientés objet. Mais les `Try Catch` sont de gros consommateurs de ressources. Les performances de votre programme sont donc réduites.

Il est néanmoins très bien d'en mettre ! Il faut donc faire attention à leur utilisation, par exemple, évitez de mettre des `Try Catch` dans des boucles. Cela a pour effet de ralentir chaque itération de votre boucle, car à chaque passage de celle-ci, une nouvelle gestion d'exception est initialisée. Ainsi, le code suivant :

```
var result = new List<int>();
try {
    for (var i = 0; i < 100; i++)
        result.Add(i);
} catch (Exception ex) {}
```

Est plus rapide que :

```
var result = new List<int>();
for (var i = 0; i < 100; i++) {
    try {
        result.Add(i);
    } catch (Exception ex) {}
}
```

Conclusion

Pour bien optimiser un algorithme, il ne faut pas hésiter à chronométrer le temps d'exécution de vos codes et à regarder pas à pas les routines les plus consommatrices en ressources pour se remettre en question et placer ces routines à des points stratégiques de vos fonctions pour que celles-ci s'exécutent le plus rapidement possible. Une fois une vitesse d'exécution acceptable atteinte, il ne reste plus qu'à optimiser la consommation mémoire de vos programmes. Pour toutes questions, commentaires ou discussion à propos de l'optimisation de code, n'hésitez pas à me contacter directement : <http://www.velersoftware.com>

Télécharger les codes sources du Benchmark :

<http://blog.velersoftware.com/wp-content/uploads/2013/09/Benchmark.zip>



Etienne Baudoux

Etudiant à SUPINFO en 2e année - Microsoft Student Partner
Ambassadeur Microsoft pour SUPINFO
ebaudoux@velersoftware.com
<http://www.velersoftware.com/>
twitter : @VelerSoftware

Maîtriser l'asynchronisme de C# 5.0 2^e partie

L'asynchronisme Microsoft avant C# 5.0



L'objectif est d'analyser les éléments les plus saillants de chaque solution, afin de comprendre pourquoi Microsoft a décliné une nouvelle offre avec Visual Studio 2012. Enfin, je pense que cette rétrospective n'est pas négligeable pour comprendre en profondeur la proposition asynchrone de C# 5.0 que nous détaillerons dans le prochain article.

Lecture d'un fichier en mode synchrone

Pour mieux comprendre les prochains exemples asynchrones, je vous propose un exemple reposant sur la classe `FileStream`. Vous pouvez utiliser cette classe pour lire/écrire dans un fichier en mode synchrone, via les méthodes `Read` et `Write`. Voici l'illustration de la méthode `Read`.

```
readonly byte[] _buffer = new byte[100];

public void ReadSync(string filename)
{
    using (var fs = new FileStream(filename, FileMode.Open,
        FileAccess.Read,
        FileShare.Read, 1024))
    {
        try
        {
            Int32 bytesRead = fs.Read(_buffer, 0, _buffer.Length);
            Console.WriteLine(«Number of bytes read={0}», bytesRead);
            Console.WriteLine(BitConverter.ToString(_buffer, 0,
                bytesRead));
        }
        catch (IOException exception)
        {
            Console.WriteLine(«Error : {0}», exception.Message);
            throw;
        }
    }
}
```

La méthode `ReadSync`, instancie un objet de type `FileStream`, en passant en paramètres : le nom du fichier à ouvrir, le mode d'accès `FileAccess.Read`, le mode de partage en lecture via `FileShare.Read`, et la taille du buffer où les octets lus seront stockés. Puis nous appelons la méthode `Read` sur l'objet `FileStream`, où nous passons le buffer d'accueil des octets, l'offset de départ de la lecture à zéro, puis enfin, le nombre d'octets que nous souhaitons lire.

Comprendre l'implémentation

Reprenons l'exemple précédant à partir de l'appel à la méthode `Read`. Lorsque vous appelez cette méthode, votre code effectue une transition depuis votre thread managed vers du code natif Win32 via la méthode `ReadFile`. `ReadFile` alloue une petite structure appelée I/O Request Packet (IRP). Cette structure est initialisée afin de contenir principalement : le descripteur du fichier, l'offset du début de lecture, l'adresse mémoire du buffer de bytes où la lecture débutera et qui devront être remplis par les octets lus, et le nombre octet à transférer [Fig.1](#).

L'implémentation de fonction Win32 `ReadFile` repose sur la méthode `NtReadFile`, qui se trouve dans la partie noyau de Windows. L'appel à la fonction noyau `NtReadFile` prend en paramètre la structure IRP qui

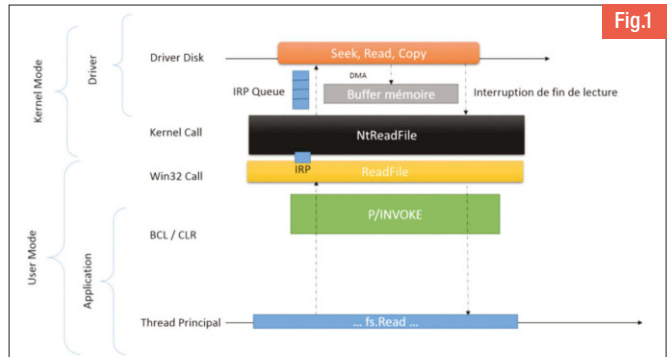


Fig.1

marque une transition en mode noyau. À partir du descripteur de fichier contenu dans la structure IRP, le noyau détermine le driver sous-jacent. Ainsi, Windows peut poster la structure IRP dans la queue de type IRP du driver NTFS sous-jacent. En fait, chaque driver maintient une queue IRP afin de répondre aux besoins de tous les processus en cours d'exécution sous Windows. Lorsque notre structure IRP est dépilée, elle est appliquée au matériel respectif, dans notre cas, le matériel est un disque dur.

C'est alors que le disque dur va réaliser l'opération de type Entrée/Sortie (E/S) réclamée : lecture de 100 octets (la taille choisie est complètement arbitraire). Pendant que le périphérique exécute son opération de type E/S, votre thread qui est à l'origine de l'appel n'a rien à faire. C'est pour cette raison que Windows place en sommeil votre thread afin de ne pas gaspiller du temps CPU pour rien.

Cependant, de nombreuses structures mémoires ont été allouées. Par exemple, la pile en mode utilisateur, la pile en mode noyau et d'autres structures mémoires qui sont présentes et totalement inaccessibles.

De plus, si l'application possède une interface graphique, la couche graphique ne peut plus répondre à l'utilisateur tant que le thread est bloqué. Lorsque le périphérique aura terminé son opération de type E/S, alors Windows réveillera le thread appelant afin de l'exécuter et de retourner depuis le mode noyau vers le mode utilisateur. Depuis le mode utilisateur, le code va retourner dans le thread managed initial, au niveau du retour de la méthode `Read`. Cette méthode peut alors retourner le nombre d'octets lus : `bytesRead`. À ce stade, le code peut consulter les octets du buffer dans la limite de la variable `bytesRead`.

Conclusion

Comme nous l'avons expliqué dans la première partie, les appels synchrones ne sont pas gênants, s'ils ne nuisent pas aux utilisateurs ou à la disponibilité des services. Notre exemple à l'avantage d'être à la fois simple et facile à comprendre. Cependant si le fichier se trouve sur un média particulièrement lent avec un nombre d'octets beaucoup plus conséquents, nous pourrions nous retrouver avec des latences insupportables pour un utilisateur. Pour éviter ces désagréments, nous pourrions rendre ce code asynchrone. Dans les exemples suivants, nous étudierons les techniques asynchrones relatives à la méthode `Read` à travers différentes API afin de juger à la fois le code engendré et l'implémentation interne.

Asynchronisme en C# 1.0

Vous avez peut-être déjà utilisé les API asynchrones disponibles depuis C# 1.0. Ce sont les versions asynchrones de méthodes synchrones relatives à des entrées/sorties (Système de fichier, Réseau, Web ...). Sur le plan de la syntaxe, le principe est simple, une méthode synchrone nommée XXX() se décline sous la forme d'un couple de méthodes de type BeginXXX/ EndXXX. La communication entre les deux méthodes est assurée par l'interface `IAsyncResult`. On appelle aussi ce modèle APM, pour **Asynchronous Programming Model**.

```
readonly byte[] _buffer = new byte[100];

public void ReadFileWithApm(string filename)
{
    var fs = new FileStream(filename, FileMode.Open, FileAccess.Read,
        FileShare.Read, 1024, FileOptions.Asynchronous);
    fs.BeginRead(_buffer, 0, _buffer.Length, ReadIsDone, fs);
}

private void ReadIsDone(IAsyncResult ar)
{
    var fs = (FileStream)ar.AsyncState;

    try
    {
        Int32 bytesRead = fs.EndRead(ar);
        Console.WriteLine(«Number of bytes read={0}», bytesRead);
        Console.WriteLine(BitConverter.ToString(_buffer, 0, bytesRead));
    }
    finally
    {
        fs.Close();
    }
}
```

La lecture est déferée au niveau du driver NTFS qui réceptionne cette demande de lecture d'octets dont la taille est `Buffer.Length`. Noter que l'option `FileOptions` doit utiliser le drapeau `FileOptions.Asynchronous` pour que l'exécution se déroule vraiment en mode asynchrone. Dans notre cas nous souhaitons lire quelques octets. Dans ce modèle, il n'y a pas de création de thread de travail supplémentaire. Sur le plan de l'exécution, ce modèle est parfait, nous sommes ici dans une implémentation 100% asynchrone.

Communication asynchrone

Pour bien comprendre chacune des offres, nous étudierons la structure de communication asynchrone. Comme nous l'avons déjà dit, c'est l'interface `IAsyncResult` qui assure la communication entre les méthodes `BeginRead` et `EndRead`:

```
public interface IAsyncResult
{
    bool IsCompleted { get; }
    WaitHandle AsyncWaitHandle { get; }
    object AsyncState { get; }
    bool CompletedSynchronously { get; }
}
```

La propriété `IsCompleted` permet de savoir si le traitement asynchrone

est terminé. Naturellement, cette propriété pourrait être utilisée au sein d'une boucle de manière à vérifier si le traitement est terminé. Mais cette philosophie est totalement opposée avec la volonté de rendre votre code le plus disponible, car dans les faits cette boucle ne fait que gâcher du temps CPU. La propriété `AsyncWaitHandle`, permet de récupérer une instance d'un type `WaitHandle` permettant d'attendre la fin du traitement de manière synchrone. Encore une fois, cette technique annule complètement le bénéfice attendu vis-à-vis de l'utilisateur. Pour information, la méthode `WaitOne` permet aussi de fournir un délai d'attente maximum. Le fragment de code ci-dessous illustre l'utilisation de la méthode `WaitOne` avec un délai d'attente.

```
if (ar.AsyncWaitHandle.WaitOne(2000, true))
{
    Int32 bytesRead = fs.EndRead(ar);
    fs.Close();
    Console.WriteLine(«Number of bytes read={0}», bytesRead);
    Console.WriteLine(BitConverter.ToString(Buffer, 0, bytesRead));
}
```

Si le résultat arrive avant 2 secondes le traitement se terminera correctement, sinon vous considérez que le traitement a échoué. Cependant, ce type de code ne devrait jamais être utilisé en production dans le cadre où la latence doit être la plus brève possible.

La propriété `AsyncState` permet de véhiculer un type personnalisé depuis la méthode appelante. Dans notre exemple, nous avons passé une instance de type `FileStream`:

```
fs.BeginRead(Buffer, 0, Buffer.Length, ReadIsDone, fs);
Que nous récupérons dans la méthode ReadIsDone.
private static void ReadIsDone(IAsyncResult ar)
{
    var fs = (FileStream)ar.AsyncState;
```

La propriété `CompletedSynchronously` permet de savoir si l'opération en mode asynchrone a été terminée de manière synchrone. C'est encore une information que nous pourrions exploiter dans le cadre d'une boucle, mais comme nous l'avons déjà dit dans le cadre de la propriété `IsCompleted`, cette propriété n'offre que peu d'intérêt vis-à-vis d'une exécution asynchrone.

Dans ce premier exemple, nous avons négligé la gestion des exceptions. Voici une version modernisée contenant un traitement des exceptions et un remplacement de la méthode `ReadIsDone` par une expression lambda, afin d'obtenir un code plus concis.

```
readonly byte[] _buffer = new byte[100];

void ReadFileWithApmAndLambda(string filename)
{
    var fs = new FileStream(filename, FileMode.Open, FileAccess.Read,
        FileShare.Read, 1024, FileOptions.Asynchronous);

    fs.BeginRead(_buffer, 0, _buffer.Length, ar =>
    {
        try
        {
            Int32 bytesRead = fs.EndRead(ar);
            Console.WriteLine(«Number of bytes read={0}», bytes
```



```

Read);
    Console.WriteLine(BitConverter.ToString(_buffer, 0, bytes
Read));
}
catch (IOException exception)
{
    Console.WriteLine(«Error : {0}», exception.Message);
    throw;
}
finally
{
    fs.Close();
}
}, null);
}

```

Cette version est à la fois plus complète, mais reste améliorable. Par exemple si notre code devait tourner dans une application graphique, nous aurions dû placer un code de synchronisation pour mettre à jour les contrôles graphiques depuis l'expression lambda qui s'exécute dans un thread du pool.

Comprendre l'exécution

Pour lire notre fichier, nous avons utilisé successivement les méthodes `ReadFileWithApm` et `ReadFileWithApmAndLambda`. Sur le plan de l'exécution, les deux programmes sont très similaires. Dans les deux cas, nous construisons un objet `FileStream`, mais cette fois nous passons le drapeau `FileOptions.Asynchronous`, afin de dire à Windows que nous souhaitons lire et écrire via des opérations asynchrones.

Pour lire le fichier, nous utilisons cette fois l'appel `BeginRead` à la place de `Read`. Comme `Read`, `BeginRead` appelle la fonction Win32 `ReadFile`. `ReadFile` alloue et initialise une structure `IRP` et passe celle-ci à la méthode noyau `NtReadFile` qui la poste dans la queue `IRP` du driver déduit du descripteur de fichier.

Contrairement au scénario synchrone où le thread appelant était bloqué, il retourne immédiatement vers l'appel `BeginRead`. Mais cette fois nous avons passé une continuation : la méthode de type callback, `ReadIsDone` dans le premier programme et l'expression lambda dans le second programme. Un délégué dans la structure `IRP` pointe sur la continuation de manière à l'exécuter lorsque le driver aura effectué la totalité de l'opération. Donc, quelque part dans le futur, le pool de threads va extraire `IRP` et invoquer votre continuation. Le prototype de la continuation ne retourne rien et prend un paramètre de type `IAsyncResult`. Cette interface permet d'informer la continuation sur l'état de l'opération asynchrone comme nous l'avons décrit dans la partie « Communication asynchrone ».

Conclusion

Pourquoi l'offre APM n'a-t-elle pas rencontré un franc succès auprès des développeurs (alors que celle-ci repose sur une implémentation 100 % asynchrone) ? Le problème principal est facile à comprendre, il suffit d'observer le code. La méthode synchrone `Read` est séparée en deux méthodes `BeginRead` et `EndRead`, ce qui ne permet pas lire et de maintenir le code facilement. À chaque appel de la méthode `BeginRead`, une référence sur l'interface `IAsyncResult` est retournée. Celle-ci doit être lue une seule fois par la méthode `EndRead`.

Dans le cadre d'un code plus complexe, avec plusieurs appels à `BeginRead`, le développeur devra s'assurer d'une parfaite relation avec les appels `EndRead` respectifs. Ceci peut être considéré comme une difficulté de programmation pour de nombreux développeurs. L'interface de communication `IAsyncResult` est rudimentaire et contient quelques propriétés peu pertinentes au regard de l'asynchronisme. Enfin, la méthode

`EndRead` s'exécute dans un thread du pool de threads ce qui peut occasionner des problèmes de concurrences vis-à-vis de la mise à jour de contrôles graphiques. Cependant, les développeurs d'applications graphiques comme **Windows Forms** et **WPF**, connaissent généralement bien la classe **`SynchronizationContext`** permettant d'envoyer ou de poster dans le thread graphique la portion de code permettant une mise à jour de contrôles graphiques.

Event-Based Asynchronous Pattern (EAP)

En C# 2.0, une nouvelle offre asynchrone orientée événement fut introduite sous le nom de EAP pour Event-Based Asynchronous Pattern. L'objectif de cette offre est de signaler la fin du traitement asynchrone sur la base d'un événement. Cette solution est particulièrement adaptée aux applications graphiques qui par nature sont orientées événements. Cette offre est parfois utilisée via la surface de design avec certains contrôles **Windows Forms**. Pour illustrer ce modèle, nous étudierons la classe `WebClient`. Nous utiliserons la méthode `DownloadStringAsync` qui permet d'initialiser un téléchargement asynchrone (la version synchrone se nomme `DownloadString`). Dans le cas d'un téléchargement asynchrone, la `WebClient` expose un événement `DownloadStringCompleted` qui est levé lorsque le chargement de la page WEB est terminé. Le code suivant illustre l'usage du pattern avec la classe `WebClient` :

```

var wc = new WebClient();
wc.DownloadStringCompleted += (o, e) =>
{
    var html = e.Result;
    listView.Items.Add(new PageLength(uri, uri.Length));
    txtResult.Text = «LENGTH: « + html.Length.ToString(«N0»);
};
wc.DownloadStringAsync(new Uri(«http://blogs.msdn.com/b/native
concurrency/»));

```

Une surcharge à la méthode `DownloadStringAsync` permet de passer un jeton permettant d'identifier l'appelant au niveau de l'événement, permettant des scénarios où de multiples appels simultanés sont nécessaires :

```

public void DownloadStringAsync( Uri address, object userToken);

```

Le code est relativement simple à écrire en comparaison avec le modèle APM. À l'instar de l'offre APM l'exécution interne reste quasi identique. On requête une URL Web qui a pour conséquence d'enregistrer cette demande du côté du driver HTTP, puis de retourner vers l'appelant en ayant pris soin d'édifier un lien entre la méthode de type callback via le pool de thread E/S et le driver HTTP.

Lorsque le driver recevra la totalité de la page Web, le contenu sera copié dans la propriété `Result` de la classe `DownloadStringCompletedEventArgs` que nous détaillons plus loin. L'exécution de l'événement `DownloadStringCompletedEventHandler`, a lieu dans le thread de l'appelant, ce qui signifie qu'un changement de contexte a été fait. En d'autres mots, l'offre EAP synchronise le code de l'événement avec le thread appelant, souvent le thread graphique.

Communication asynchrone

Contrairement au modèle APM où la communication est assurée par une structure rudimentaire `IAsyncResult`, l'événement EAP possède un argument bien plus riche fonctionnellement, `DownloadStringCompletedEventArgs`. Cette classe dérive de la classe `AsyncCompletedEventArgs` pour y ajouter la propriété `Result`, que nous avons déjà commentée.

L'essentiel des propriétés se trouve dans la classe mère `AsyncCompletedEventArgs`.

La propriété **Cancelled** permet de savoir si le traitement asynchrone a été abandonné (la méthode `wc.CancelAsync` a été appelée).

La propriété **Error** permet de récupérer potentiellement une exception levée durant le traitement.

La propriété **UserState** permet de véhiculer un type personnalisé.

Dans ce premier exemple, nous avons négligé la gestion des exceptions. Voici une version modernisée contenant l'utilisation d'une expression lambda, un traitement des exceptions, et l'illustration du traitement d'annulation.

```
var wc = new WebClient();

wc.DownloadStringCompleted += (o, e) =>
{
    if (e.Error != null)
    {
        try
        {
            // restore structured exception handling
            throw e.Error;
        }
        catch (WebException ex)
        {
            // Handle an exception
        }
    }

    if (e.Cancelled) return;

    var html = e.Result;
    totalLength += html.Length;
    listView.Items.Add(new PageLength(uri, uri.Length));
    txtResult.Text = «TOTAL LENGTH: » + totalLength.ToString(«N0»);
};

wc.DownloadStringAsync(new Uri(uri));
wc.CancelAsync();
```

Conclusion

Dans le cadre d'un développement graphique, l'offre EAP est supérieure au modèle APM à la fois sur le plan de l'API, et sur le plan de la structure de communication asynchrone. La structure de communication est plus pertinente vis-à-vis d'un usage asynchrone.

La capacité d'annuler le traitement est prise en compte, ce qui n'est pas le cas du modèle APM (même si vous pouvez écrire votre propre gestion d'annulation de traitement). Cette offre est conçue pour exécuter son événement de retour d'exécution dans le contexte de l'interface graphique, donc contrairement au modèle APM, l'expression lambda s'exécute dans le thread de l'appelant. Vous avez sans doute remarqué que la définition de l'événement consacré au traitement du résultat asynchrone est spécifiée avant d'initialiser le traitement lui-même, ce qui est sur le plan de l'usage un peu déroutant.

Task-Based Asynchronous Pattern (TAP)

Le type `Task` a été introduit avec la version 4.0 du Framework .NET afin d'offrir une forme de remplacement à l'API Thread jugée complexe et coûteuse en ressource (<http://blogs.msdn.com/b/devpara/archive/2010/04/06/programmation-parall-le-avec-c-4-0-offre-parall-le-orient-e-t-ches-part-1.aspx>). Le type `Task` se décline aussi sous la forme d'un type générique symbolisant un traitement dont le résultat sera disponible dans le futur.

Communication asynchrone

Si vous revenez un instant sur les modèles APM et EAP, nous avons dans les deux modèles une forme de résultat spécifique. Dans le premier cas, le résultat est incarné par l'interface `IAAsyncResult` et dans le second cas, le résultat est représenté par la classe `DownloadStringCompletedEventArgs`.

La classe `Task<TResult>` résout l'essentiel des complexités rencontrées dans les modèles précédents. Dans ce cadre, les éléments les plus intéressants sont :

- ▀ La propriété **Result** permet de récupérer le résultat du traitement. Notons que la propriété `Result` est synchrone et donc bloquante, tant que le traitement asynchrone n'est pas terminé.
- ▀ La propriété **Exception** permet de récupérer potentiellement une exception levée durant le traitement.
- ▀ La méthode **ContinueWith** permet de lancer une nouvelle tâche dès que la première sera terminée, c'est une continuation de la première tâche.

Avec le Framework 4.5, nous disposons d'une nouvelle déclinaison de méthodes asynchrones. Par convention ces nouvelles méthodes sont suffixées par le mot **Async**. Dans notre exemple, la classe `WebClient` contenait déjà une méthode dont la signature se terminait par **Async** : `DownloadStringAsync`. Nous savons que cette méthode joue le rôle de démarreur du traitement asynchrone dans le modèle EAP.

La nouvelle version du Framework 4.5 expose une nouvelle méthode, nommée `DownloadStringTaskAsync` et qui retourne un type `Task<string>`. Ces nouvelles méthodes asynchrones représentent un nouveau modèle appelé Task-Based Asynchronous Pattern (TAP).

Ce nouveau modèle repose sur l'utilisation du type `Task<TResult>` pour toutes les méthodes post fixées par **Async**. Ainsi l'appelant peut poursuivre l'exécution à travers une continuation de l'API `Task`. Voici l'illustration de la version TAP de notre exemple avec la classe `WebClient`.

```
Task<string> task = new WebClient().DownloadStringTaskAsync(
    uri);

task.ContinueWith(previous =>
{
    if (previous.Exception == null)
    {
        _totalLength += previous.Result.Length;
        listView.Items.Add(new PageLength(uri, uri.Length));
    }
},
    CancellationToken.None,
    TaskContinuationOptions.None,
    TaskScheduler.FromCurrentSynchronizationContext());
```

Dans cet exemple, nous utilisons la nouvelle méthode `DownloadStringTaskAsync` pour démarrer le téléchargement d'une page Web. Dans la continuation, `ContinueWith`, nous utilisons une tâche afin de récupérer le résultat et d'ajouter au contrôle graphique les informations requises. Pour satisfaire la contrainte de mise à jour d'un contrôle graphique depuis un thread différent du thread graphique, nous exécutons toute la tâche de continuation dans le thread graphique en le plaçant dans le paramètre dédié au Scheduler via `TaskScheduler.FromCurrentSynchronizationContext()`. Le paramètre `CancellationToken.None` signifie que nous souhaitons ne pas gérer un jeton d'annulation et enfin le paramètre `TaskContinuationOptions.None` signifie que nous n'imposons pas de contraintes sur l'exécution de cette continuation.

Sur le plan de l'exécution, le modèle TAP est très proche du modèle APM, lorsque les méthodes appelées sont relatives à des opérations de type

E/S. L'appel de la méthode **DownloadStringTaskAsync** engendre une suite d'appels qui aboutira à poster une structure **IRP** au sein du driver **HTTP**. Mais à la différence du modèle **APM**, le retour immédiat vers l'appelant va retourner une instance du type **Task<string>**.

Cet objet permet d'appeler la méthode **ContinueWith** pour enregistrer une méthode de type callback qui sera appelée lorsque l'opération de type E/S sous-jacente à **DownloadStringTaskAsync** sera terminée. Lorsque le téléchargement de la page Web sera terminé, une structure **IRP** initialisée avec les informations relatives au téléchargement sera placée dans l'un des threads du pool de thread E/S. Si une tâche de continuation a été enregistrée, celle-ci sera initialisée et exécutée par le thread du pool sélectionné.

Implémenter une méthode TAP

Dans les paragraphes précédents, nous avons constaté que l'utilisation des méthodes TAP est relativement simple. Cependant, nous n'avons pas abordé l'implémentation. Pour illustrer ce sujet, je vous propose la réécriture d'une nouvelle méthode du type **Task** du Framework 4.5 :

```
public static Task Delay(int millisecondsDelay)
```

Cette méthode permet de créer une tâche qui se terminera après un délai fourni par l'utilisateur.

Pour implémenter cette méthode, nous allons faire appel à un type déjà présent dans le Framework 4.0 :

```
TaskCompletionSource<TResult>
```

Le type de **TaskCompletionSource<TResult>** sert à deux objectifs connexes, les deux font allusion à son nom: il est une source de création d'une tâche, et la source de l'achèvement de cette tâche. Généralement, vous créez une instance de **TaskCompletionSource<TResult>** puis vous gérez à la main ce que sera la tâche sous-jacente. Cette tâche est accessible via sa propriété **Task**. Contrairement aux tâches créées par **Task.Factory.StartNew**, une instance de **TaskCompletionSource<TResult>** n'est pas animée par un scheduler.

Au contraire, le type **TaskCompletionSource<TResult>** fournit des propriétés et des méthodes qui vous permettent de contrôler la vie et l'achèvement de la tâche associée en fonction de vos besoins. Le type expose les méthodes suivantes : **SetResult**, **SetException**, **SetCanceled**, ainsi que **TrySet***, permettant de gérer complètement à la main la vie de la tâche sous-jacente. Cependant, certaines situations peuvent donner lieu à des exécutions concurrentes où plusieurs threads peuvent tenter d'accéder à la source d'achèvement simultanément. Dans ce cas les déclinaisons des méthodes **TrySet*** retournent des booléens indiquant le succès plutôt que de jeter une exception. Voici une implémentation simpliste, mais opérationnelle de la méthode **Delay**, illustrant l'usage du type de **TaskCompletionSource<TResult>**.

```
Task Delay(int millisecondsDelay)
{
    var tcs = new TaskCompletionSource<object>();

    var timer = new System.Timers.Timer(millisecondsDelay)
    { AutoReset = false };

    timer.Elapsed += delegate
    {
        timer.Dispose();
        tcs.SetResult(null);
    };
}
```

```
};

timer.Start();

return tcs.Task;
}
```

Dans un premier temps, nous créons une instance de type **TaskCompletionSource<object>**, car la méthode **Delay** retourne un type **Task** non générique. Puis nous créons une instance de type **Timer** en passant en paramètre le nombre de millisecondes à attendre, tout en indiquant que ce timer ne s'exécutera qu'une seule fois via la propriété **AutoReset** placée à **false**. Puis nous nous abonnons à l'évènement **Elapsed**, où nous fermons les ressources du timer, puis au niveau de l'instance **TaskCompletionSource** nous indiquons que la tâche ne retournera pas de valeur. Puis nous lançons le timer. Enfin nous retournons la valeur de la propriété **Task** de l'instance **TaskCompletionSource**.

L'objectif de ce paragraphe est de vous sensibiliser sur l'implémentation d'une méthode asynchrone. L'utilisation type **TaskCompletionSource** est largement utilisée dans l'implémentation de méthodes asynchrones.

Conclusion

Le modèle TAP s'appuie pleinement sur un jeu de nouvelles méthodes asynchrones disponibles à la fois en .NET 4.5 et sur WinRT. Le modèle s'appuie sur le type **Task** générique pour fournir une solution bien plus simple à utiliser que les modèles **APM** et **EAP**. Sur le plan de l'exécution, les nouvelles méthodes asynchrones E/S du Framework 4.5 reposent sur un modèle interne 100 % asynchrone comme pour les modèles précédents. Sur le plan de l'utilisation, le modèle TAP est bien plus simple que les modèles **EAP** et **APM**. À la fois plus naturel dans l'expression et plus souple dans l'usage, le modèle TAP n'est pas satisfaisant lorsque nous l'utilisons à travers du code fortement impératif comme à travers de boucles par exemple, comme nous l'avons démontré dans la première partie.

Utiliser TPL pour appeler le pattern APM

Depuis le Framework .NET 4.0, la librairie TPL simplifie grandement les appels asynchrones pour des méthodes de longue durée. Cependant, il est généralement préférable d'utiliser les méthodes **APM** que d'utiliser directement TPL pour implémenter la version asynchrone d'un code orienté E/S. Heureusement, la classe statique **TaskFactory** de la classe **Task**, a été pensée pour invoquer le couple de méthodes **APM**.

Le type **Task** implémente l'interface **IAAsyncResult**, ainsi les tâches peuvent supporter le modèle **APM** et parfois les étendre. Par exemple, la propriété **AsyncState** du type **Task** étend la propriété **AsyncState** de l'interface **IAAsyncResult**, ce qui vous permet de transmettre un type personnalisé en interne du traitement asynchrone. Car vous l'aurez compris, la méthode **EndRead** est appelée en interne, afin d'obtenir un code plus simple, orienté TAP. Un code orienté TAP est facilement transformable pour être utilisé avec l'offre **async/await**.

CONCLUSION

Dans cette seconde partie, nous avons montré à travers une rétrospective des technologies asynchrones Microsoft, les principaux défauts de ces offres. Naturellement, c'est grâce à cette expérience que l'éditeur a imaginé une nouvelle approche que nous détaillerons dans la prochaine partie.

 Bruno Boucard

Architecte senior | MVP C++ chez Finaxys

Son Blog : <http://blogs.msdn.com/devpara/>

Timeline : 1991

Objet : BeOS

Qui se souvient encore de la compagnie Be Inc. ? Nous sommes au début des années 1990, Jean-Louis Gassée, ancien d'Apple, a l'ambition de révolutionner le système d'exploitation et l'utilisation de l'ordinateur.

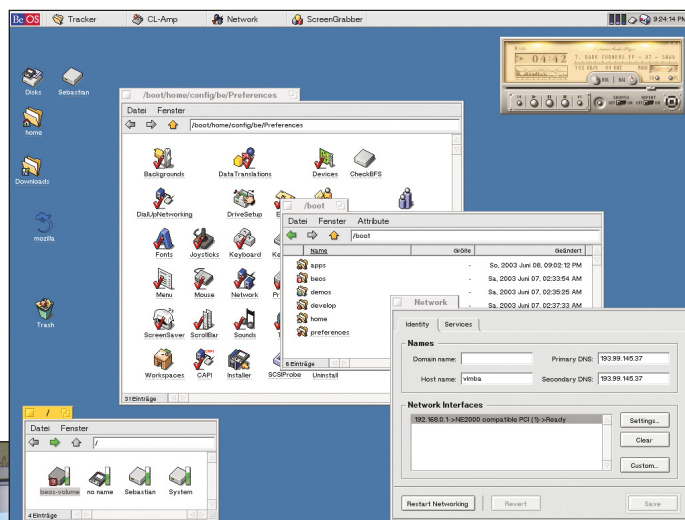
Une vie mouvementée

BeOS se voulait à la pointe technologique de l'époque : multitâche préemptif, gestion du 64 bits, système de fichiers journalisé, multiprocesseur. En 2013, cela pourrait être assez banal mais pas dans les années 1990 ! Dès le départ, le point fort de BeOS était une gestion native du multimédia, bien plus efficace que sur Mac ou Windows. Par certains aspects, on a souvent comparé BeOS avec NextStep, le système créé par NeXT (créé par Steve Jobs quand il fut éjecté d'Apple). Be avait un avantage certain : le prix. Les machines BeBox étaient plus abordables que les NeXTCube.

BeOS n'avait pas à se préoccuper d'un existant. Ainsi l'interface était une des plus fluides et des plus riches de l'époque, Windows se cherchait encore et Apple était bloqué avec Système (ver-

sions 1 à 9) et le futur Copland. Les fondations étaient solides : C++, Posix (compatibilité partielle), Unicode...

Les premières versions exploitables sortirent à partir de 1995 tout d'abord sur plate-forme Hobbit (ATT) puis dès janvier 1996 sur PowerPC. Un an plus tard, BeOS arriva sur processeur Intel.



Jusqu'en 2000, Be supporta les deux processeurs, mais finit par supporter uniquement Intel (R5.1), avant d'arrêter le développement. Mais le chemin sur Intel fut difficile. La première mouture fut particulièrement instable et incomplète.

Lorsque BeOS sortit dans une version exploitable, Apple cherchait un système capable de remplacer Mac OS dont l'évolution était bloquée par

les échecs du projet Copland (dont certaines briques techniques servirent dans Mac OS 8). Be Inc. était en négociation avec la Pomme pour un rachat. Mais le projet échoua, un des blocages était le prix exigé : 200 millions. Or, BeOS n'avait pas de marché et était encore immature. NeXT fut alors choisi... NeXTStep - OpenStep donnèrent Mac OS X.

En 2001, Be fut racheté par Palm. Des morceaux de BeOS se retrouvèrent par la suite dans Palm OS Colbat (2004)... Depuis la mort officielle du système, plusieurs projets open source tentent de faire vivre BeOS avec des systèmes compatibles, le plus connu est Haiku. La dernière version remonte à novembre 2012.

Cette rubrique va nous transporter à travers l'espace-temps de l'informatique. Pour découvrir ou redécouvrir des technologies, logiciels, langages oubliés ! Time machine est aussi votre rubrique. Envoyez-nous vos chroniques temporelles : redaction@programmez.com

Pour les développeurs

« Les développeurs et amateurs de nouvelles technologies ne manqueront pas d'apprécier. Debugger, sources, exemples par dizaines avec en prime un BeIDE, un CodeWarrior pour BeOS concocté par Metrowerks. Les développeurs sous CodeWarrior retrouveront leurs habitudes avec une interface très proche de son grand frère. L'environnement est une chose, mais le plus attrayant dans BeOS c'est un système entièrement objet !!! Une seule chose manque à l'appel, Java. Pour le moment, Java n'est pas implémenté au cœur de BeOS. Étant disponible sur Intel et PowerPC, il suffira de compiler sur chaque plate-forme le code de son application. Toutefois, il ne faudra pas oublier de consulter les notes techniques pour respecter les structures spécifiques aux plates-formes. » (Programmez ! n° 9)

Abonnement : Programmez, 17, Route des Boulangers, 78926 Yvelines Cedex 9 - 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 exclusivement sur www.programmez.com



Une publication Nefer-IT
7 avenue Roger Chambonnet
91220 Brétigny sur Orge
redaction@programmez.com
Tél. : 01 60 85 39 96

Directeur de la publication

& rédacteur en chef : François Tonic

Ont collaboré à ce numéro : S. Sauré

Secrétaire de rédaction : Olivier Pavie

Experts : Gregory, J. Dollon, A. Guet, T. Leriche-Dessirier, I. Ristic, A. Tissier, D. Poulin, V. Saluzzo, M. Duignaud, F. Ducret, M. Ohayon, D. Mondon, J. Piques, V. Leduc, V. Néel, J. Amsellem, T. Ouvre, J. Antoine, C. Peugnet, C. Villeneuve, M. Vion, E. Baudoux, B. Boucard.

Crédits couverture : 09-09-13 © miakiev / istock

Maquette : Pierre Sandré

Publicité : Régie publicitaire, K-Now sarl. Pour la publicité : Tél. : 01 41 77 16 03 - diff@programmez.com.

Imprimeur : S.A. Corelio Nevada Printing, 30 allée de la recherche, 1070 Bruxelles, Belgique.

Ce numéro comprend : un encart jeté sur une partie du tirage.

Marketing et promotion des ventes :

Agence BOCONSEIL - Analyse Media Etude

Directeur : Otto BORSCHA oborscha@boconseilame.fr

Responsable titre : Terry MATTARD

Téléphone : 0967320934

Contacts

Rédacteur en chef : ftonic@programmez.com

Rédaction : redaction@programmez.com

Webmaster : webmaster@programmez.com

Publicité : diff@programmez.com

Evenements / agenda :

redaction@programmez.com

Dépôt légal : à parution - Commission paritaire : en cours - ISSN : 1627-0908

© NEFER-IT / Programmez, novembre 2013

Toute reproduction intégrale ou partielle est interdite sans accord des auteurs et du directeur de la publication.

ASPOSE.TOTAL POUR JAVA



pour JAVA

Convertir, Imprimer, Créer
Combiner & Modifier

DOC PPT
PDF MSG
XLS VSD

formats d'images et beaucoup PLUS

ASPOSE.TOTAL POUR PRODUITS JAVA

PROGRAMMATION RAPIDE ET SIMPLE

Aspose.Words pour Java: Créer, modifier et convertir les documents Word et les autres fichiers de traitement de texte. Contrôle du contenu, effectuer le publipostage et convertir des documents aux formats de fichier natifs.

Aspose.Pdf pour Java: Créer, modifier et convertir les fichiers Adobe Acrobat PDF et les pages d'image et d'autres formats de fichiers. Travailler avec des formulaires, signatures et les documents sécurisés.

Aspose.Slides pour Java: Créer, modifier et convertir les présentations Microsoft PowerPoint. Créer les diapositives de scratch, enregistrer les diapositives, recourir à des sources de données externes et de protéger les présentations.

Aspose.BarCode pour Java: Lire et générer les codes à barres dans une gamme populaire 1D & 2D.

Aspose.Email pour Java: Travailler avec Microsoft Outlook et les fichiers d'archivage, de réception et d'envoi. Synchroniser les e-mails.

Aspose.OCR pour Java: Exécuter la reconnaissance de caractères dans différentes langues. Travailler avec un OMR pour construire des applications puissantes et efficaces.

Aspose.Imaging pour Java: Créer, modifier et convertir des images rastées et vectorisées. Travailler avec Adobe PhotoShop au format PSD pour dessiner des image et les préparer pour un traitement externe.

SCANNEZ POUR
REMISE DE -20%



Ventes Europe:
+44 141 416 1112
sales.europe@aspose.com

 **ASPOSE**
Votre Expert des formats de fichiers

HTML5+jQuery

Toute App, Tout Navigateur, Toute Plateforme, Tout Appareil



IGNITEUITM
INFRAGISTICS JQUERY CONTROLS



Téléchargez Votre Version d'Evaluation Gratuite
www.infragistics.com/telechargez



Infragistics Ventes France  0800 667 307 • Infragistics Ventes Europe +44 (0) 800 298 9055

Copyright 1996-2013 Infragistics, Inc. All rights reserved. Infragistics and NetAdvantage are registered trademarks of Infragistics, Inc. The Infragistics logo is a trademark of Infragistics, Inc.