

PROGRAMMEZ !

www.programmez.com

Mensuel n°161 - mars 2013

La boîte à outils du développeur

© istockphoto/Jamie Farrant

Windows, iOS, Android, Open Source, Java, web : le choix des développeurs, les outils, les bons conseils



Firefox OS

L'autre système pour smartphone !



Carrière

Devenez développeur de jeux vidéo !

Communauté Java

Les nouveautés de Devoxx France 2013

Au cœur des outils et des technologies

- Ant
- Intel HTML5 Porter Tool,
- APISpark, Windows 8 MVVM

- SharePoint 2013
- TFS TeamBuild...
- Heroku, Lua

Printed in EU - Imprimé en UE - BELGIQUE 6,45 €
SUISSE 12 FS - LUXEMBOURG 6,45 € DOM Surf 6,90 €
Canada 8,95 \$ CAN - TOM 940 XPF - MAROC 50 DH

M 04319 - 161 - F: 5,95 €



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



b i z & l o v Crédits photos : © Gettyimages / John Lund / Blend Images / GraphicObsession

Aruba lance Aruba Cloud avec pour ambition de devenir l'un des leaders européens des services Cloud.

Comment ? **En adoptant une démarche GLOBALE** par sa présence paneuropéenne et ses investissements technologiques dans les services IaaS, et **LOCALE** grâce à des datacenters et des équipes implantés dans chaque pays pour être au plus proche de ses clients.

- **4 datacenters** hautement sécurisés en France, en Italie, en République Tchèque et bientôt en Grande-Bretagne, Allemagne et Espagne.
- Une interface, un accompagnement et des interlocuteurs français.
- **Un large choix de technologies de virtualisation** (Hyper-V, Hyper-V Low Cost et VMware) et plus de **60 templates disponibles**.
- Des outils de création et de gestion d'infrastructure virtuelle intelligents comme **VisualCloud** ou des **API publiques**.
- Un système de projection mensuelle pour éviter les surprises et maîtriser ses dépenses.

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
(CÔTÉ D'UN APPEL LOCAL)

Le smartphone explose

Le marché du smartphone demeure l'un des plus actifs ou tout du moins l'un des plus stratégiques pour Apple, BlackBerry (ex-Rim), Google et Microsoft, côté système et les constructeurs.

Les moindres problèmes deviennent des drames : iOS 6.1 apporte des soucis d'autonomie liés à un problème Exchange, BlackBerry Z10 est bon mais sera-t-il le sauveur de BlackBerry ? Nokia pourrait aussi aller voir du côté d'Android, problèmes et bugs avec Windows Phone 7.8 et sur la Surface Pro, etc.

> Un marché de plus en plus concentré malgré la concurrence

Quand on regarde les derniers chiffres Gartner sur les ventes mondiales des téléphones portables, le doute n'est pas permis : au 4e trimestre 2012, Samsung est devenu le premier vendeur de téléphones (tous modèles confondus, du plus basique au smartphone) avec 106 millions d'appareils contre 85 pour Nokia et un « petit » 43 pour Apple. Si on considère toute l'année 2012, Samsung dépasse Nokia (384 millions contre 333), Apple arrivant 3e avec 130 millions d'unités. La chute de Nokia est impressionnante. Le smartphone étant encore un marché naissant pour ce constructeur. La multiplication des modèles par Samsung a fini par payer.

Si nous considérons uniquement le smartphone, qui prend de plus en plus d'importance, Gartner pense qu'ils représenteront 1 milliard d'unités fin 2013 sur un marché global de 1,9 milliard de téléphones. Et si nous regardons uniquement ce critère, la situation change radicalement : Android arrive 1er avec 144 millions d'unités, iOS est plus que jamais 2e avec 43M. d'unités et très loin derrière, BlackBerry, avec 7,3 millions (sur un an, le constructeur subit une très sévère chute). A noter que le système Symbian va disparaître dès 2013, Bada est insignifiant, et Windows Phone affiche 6,1 millions d'unités. Chiffres pour le 4e trimestre 2012.

Ces chiffres sont intéressants pour comprendre le marché smartphone et donc la logique développeur d'aujourd'hui. Soyons clair, pour un



éditeur ou un développeur qui souhaite se faire connaître, vendre une application, le choix est très simple : Android et iOS.

Le reste de la concurrence est trop loin derrière.

> BlackBerry 10 et Windows Phone 8 : même combat

Pour Microsoft, si le lancement de Windows Phone 8 est finalement relativement bon (intérêt des utilisateurs et un App Store qui

se remplit), le système souffre de deux problèmes cruciaux : manque de notoriété, trop peu de modèles de téléphones disponibles. Microsoft doit absolument élargir l'offre. Et parallèlement, il doit aussi draguer les développeurs pour qu'ils développent sur la plateforme : prouver qu'ils peuvent gagner de l'argent. Il ne suffit pas d'avoir x applications mais d'avoir un modèle économique réel et actif. Les dysfonctionnements de Windows Phone 7.8 doivent être un avertissement pour Microsoft : attention à rester vigilant... BlackBerry 10 est pour BlackBerry (ex-RIM) l'opération de la dernière chance. Avec deux modèles, dont le tout tactile Z10 (qui arrive un peu au compte-gouttes sur les différents marchés), le constructeur attend tout d'abord une stabilisation avant d'espérer une relance de sa part de marché. Si sur certains pays, BlackBerry a réussi jusqu'à présent à se préserver, aux Etats-Unis, la chute a été abyssale.

L'offre BlackBerry 10 est cependant séduisante : un modèle Z10 de qualité et bien fini (un modèle avec clavier existe aussi), un système performant avec de belles fonctions (comme le Hub, la gestion perso/pro) apparaissant comme un mixte entre iOS et Android sur l'interface. Côté développement, BlackBerry mise sur plusieurs modèles : natif / Cascade, HTML5, Adobe Air et Java. Le côté Java facilitant le portage des applications Android. Le but de BlackBerry est que le développeur puisse repackager très rapidement une application Android pour BlackBerry 10. Aujourd'hui, Android 2.3.3 est supporté mais l'objectif est de proposer Android 4.x courant 2013. L'enjeu est crucial pour le constructeur car si le développeur ne développe pas spécifiquement pour BlackBerry, il faut lui faciliter la tâche.

Pour 2013, clairement, la bataille Microsoft et BlackBerry est de savoir qui sera la 3e plateforme mobile après Android et iOS. Mais à eux de convaincre les développeurs et les utilisateurs... #



numéro 161 - mars 2013

Sources des articles sur
www.programmez.com



BUZZWORD

Le smartphone explose 3

ACTUS

En bref 6

RESSOURCES

Tout savoir sur Devoxx France 2013 10



MATERIEL

Une pluie de nouveaux SSD 12

DÉVELOPPEUR DU MOIS

Antonio Goncalves :
du JUG Paris à Devoxx 16

DOSSIER

La boîte à outils du développeur

L'outillage d'un développeur iOS – Windows 31
 Boîte à outils Windows :
 les bonnes pratiques d'un développeur 32
 Le développement iOS au quotidien :
 quels outils utiliser ? 34
 IDE Java : il y a du monde ! 37
 Passage en revue de la boîte
 à outils idéale du développeur Android 42
 Les outils complémentaires et utiles 44



VEILLE TECHNO

Firefox OS : l'autre système pour smartphones 18

Débuter avec Firefox OS 20

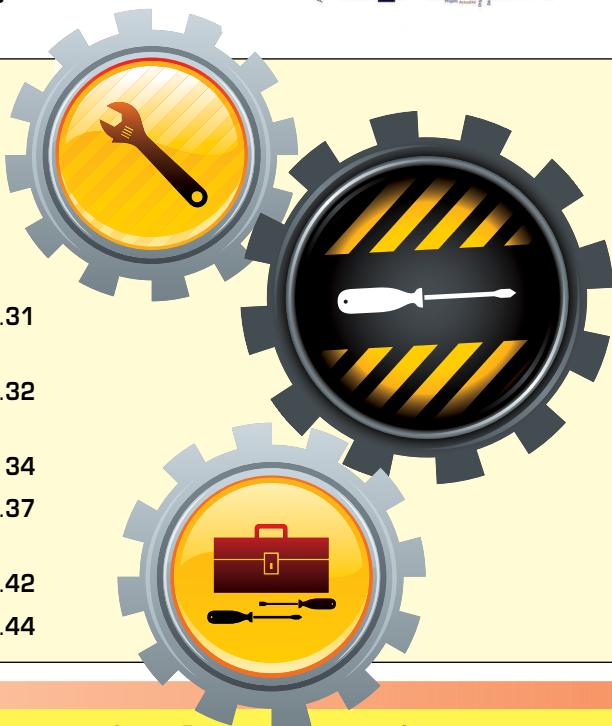
OUTILS

Nouveautés autour du développement

ActionScript avec Flash Builder 4.7 22

Agenda 23

Les nouveautés sociales de SharePoint 2013 24



A lire dans le prochain numéro

N°162-Avril 2013, à paraître le 30 mars 2013

- **Responsive Interface - Responsive Design :**

comment et pourquoi créer des interfaces capables de s'adapter automatiquement aux contextes utilisateurs ?

- **Métier Webmaster :**

Etre webmaster en 2013 : profils, nouvelles compétences, salaires.

- **Méthodes agiles : Scrum - Scrummaster**



48



© F. Tonic

WebKit, pourquoi es-tu si méchant ?

// mode troll

Sans moteur de rendu, le navigateur web serait incapable d'afficher la page internet. Depuis plus d'un an, WebKit est le troll par excellence. Ce mois-ci, le sujet revient une fois de plus en Une avec le passage du navigateur Opera à WebKit, tout en jetant son propre moteur...

Rappelons les événements : Opera, navigateur « alternatif », possédait son propre moteur de rendu, Presto. Il avait beau être léger, performant et efficace, Opera n'a jamais pu s'imposer auprès des utilisateurs. Mais aujourd'hui, les performances et les fonctionnalités de ces moteurs sont vitales pour le navigateur et mieux vaut une large communauté d'utilisateurs et de développeurs pour assurer l'évolution du moteur.

Opera a donc tranché : migrer purement et simplement sur WebKit. Aujourd'hui, ce moteur est la référence aussi bien sur l'ordinateur que sur les terminaux mobiles : Apple, Google, Nokia, Qt, Adobe, Amazon, Oracle, BlackBerry l'utilisent. A l'origine du projet était KHTML, WebKit étant un fork initié par Apple pour être le cœur de Safari. Le projet fut rendu open source en 2005. La popularité de ce moteur n'a jamais cessé de croître.

WebKit déjà omniprésent a cannibalisé un nouvel acteur, et de son plein gré ! Oui, et alors ? Certains ont voulu y voir un nouvel IE6, une monoculture pour le web, des standards mal supportés, etc. Bref, tout sauf WebKit ! Mais Haavard (employé d'Opera) a longuement décrit le pourquoi du comment de ce changement : « *oui, la monoculture est mauvaise, mais Opera n'a jamais été en position de prétendre à la première place...* ». Et ce changement est là justement pour accélérer la croissance d'Opera et devenir un contributeur important du moteur (projet open source et collaboratif rappelons-le). Opera soutient les standards ouverts et veut peser sur les fonctionnalités du moteur.

Il y a un an, Daniel Glazman (co-chairman du groupe de travail CSS du W3C) dénonçait en WebKit un nouvel IE6 du web dont les développeurs ne devraient pas utiliser les préfixes. Microsoft faisait la même mise en garde, tout en privilégiant ses propres préfixes. Nous avions analysé le problème dans le n°150 de Programmez !

Si WebKit est populaire, est-ce la faute d'Apple ?

Non. Si les navigateurs ont adopté WebKit, c'est parce que le moteur a prouvé ses capacités à évoluer, sa maturité, ses performances. Mozilla possède aussi un excellent moteur, Gecko, mais il ne s'est jamais imposé à l'extérieur de Firefox et le virage mobile a été tardif. Microsoft met en avant Trident d'IE10, mais il reste dans le périmètre de l'éditeur.

Ne faudrait-il pas jeter la première pierre à Mozilla et à Microsoft pour ne pas avoir voulu ou su promouvoir leur moteur de rendu ? Microsoft n'aurait-il pas intérêt à sortir Trident (idem pour Gecko) en projet open source et faire comme WebKit, des Nightly build pour les utilisateurs avancés et développeurs et IE pour la stabilité, avec une version majeure annuelle ?

Pour le moment, je vous propose : [shutdown -r now](http://shutupnow.com)

[ftonic@programmez.com] François Tonic

Blog Opera : <http://my.opera.com/haavard/blog/2013/02/13/webkit>

Post de Daniel Glazman : <http://goo.gl/6UNoE>

// fin de commentaires

CARRIÈRE

Devenir développeur de jeux vidéo 46

Un secteur attrayant mais très concurrentiel et sous pression	47
De nombreux métiers et des développeurs au cœur des jeux.....	48
Un bon étudiant ou développeur ne fait pas forcément un bon développeur de jeux vidéo !	51

JE DEBUTE avec...

Démystifier le développement des plug-ins d'Eclipse (Partie 4).....	52
Déployer une application Java EE dans le Cloud avec Heroku	56
Lua, vous avez dit Lua ?	60

CAS D'USAGE

Création d'activités pour TeamBuild	64
---	----

PRATIQUE

Développer facilement vos API web avec APISpark	68
Développer une application Windows 8 avec MVVM	72

WEBMASTER

Découvrir l'outil de construction de projets ANT	77
--	----

BUGTRACK

Android Developer Tool sur Linux 64 : bugs et installation	81
--	----

Microsoft Tech Days 2013 : les développeurs à l'honneur

Avec les nouveaux systèmes Windows, le développeur reste plus que jamais à convaincre et à courtiser. Microsoft a profité des derniers TechDays pour réaffirmer son attachement aux programmeurs. Il faut dire que la communication parfois un peu hasardeuse autour de XNA, Silverlight, la mise en avant systématique de HTML 5 / CSS/ JavaScript et le secret du SDK de Windows Phone 8 avaient agacé une partie de la communauté.

« Nous aimons les développeurs », voilà en substance ce qu'il fallait comprendre. Avec les nouveaux outils, les nouveaux systèmes, Microsoft veut repartir à l'assaut mais en étant plus juste dans les discours. Il a rappelé que Windows n'était pas seulement HTML 5 et JavaScript, mais aussi (et surtout ?) du C++ et C# / XAML.

L'éditeur doit convaincre le développeur de [re]venir. Aujourd'hui, sans applications, une plateforme ne vaut pas grand chose. Or il faut remplir les App Store Windows 8, pour les tablettes et désormais Office.

Microsoft a voulu mettre en avant le développeur/designer qui prend une place importante dans les projets mobiles et Windows 8, car l'ergonomie est primordiale ainsi que les usages. « En collaborant avec des développeurs, en travaillant à des applications innovantes, nous avons une position privilégiée pour observer ce qui se joue aujourd'hui sur la scène du numérique : le design est au cœur du métier de développeur, et inversement, le numérique est au cœur des préoccupations des designers, qui sont de plus en plus nombreux à intégrer « nativement » le digital dans leurs projets. C'est en tout état de cause l'avènement d'un duo « développeur/designer » qui voit le jour. C'est encore plus vrai depuis le lancement de Windows 8 et son emblématique interface « Modern UI » : tous les pro-



D.R.

duits Microsoft, toutes les applications développées sur Windows répondent désormais à ce 'dresscode' design. Ce n'est pas une mode, c'est le signe d'une transformation profonde de la nature même d'une application logicielle : ce n'est plus une somme de fonctionnalités, c'est un système intelligent et immersif qui accompagne un usage », a commenté Jean Ferré.

Microsoft propose Modern.IE dédié au développement web et à la réalisation d'un bon site pour IE (et les autres navigateurs). Site : <http://www.modern.ie/fr-fr>

> Beaucoup de matériel !

L'édition 2013 des TechDays a été marquée par une présence plus importante de constructeurs et particulièrement pour la partie portable hybride et tablette, deux marchés en pleine explosion. Mais la partie smartphone restait finalement assez discrète. HP, Samsung, Dell étaient très présents. Intel démontre une technologie maison prometteuse : Perceptual Computing. Une sorte de Kinect mais plus discrète et très efficace. Actuellement, seul le kit développeur est disponible. Parmi les communautés

sur place, notons la présence de Kinect Genius Bar, la communauté française pour Kinect Windows ! A découvrir absolument : <http://kinectgeniusbar.com>

> Deux sessions énormes !

Désormais incontournables : Coding4fun et Geek in the house ! Salle pleine à craquer, en direct live, ambiance électrique, intervenants déjantés, voilà la recette de ces sessions hors normes. Coding4Fun a été à la hauteur du défi. Mitsu Furuta a codé en live un Z-Order d'objet graphique qui a laissé sur place toute la salle. Encore plus fort : le Sandwich Locator. Après minuit, on prend tout sandwich qui se présente. Malheureusement Coding4fun ne se raconte pas, il faut le voir pour se mettre dans l'ambiance. Autre session, Geek in the house. Avez-vous déjà imaginé un casque de moto reprenant celui d'Iron Man avec vision augmentée, multi-écran virtuel, casque neuronal, pilotage de l'ouverture du casque par smartphone ? Salah Amer (geek++ qui alimente un serveur avec un panneau solaire) l'a fait ! Enfin presque, la Police n'a pas trop apprécié l'expérience.

> Quelques liens

Site officiel (avec replay des sessions) : <http://www.microsoft.com/france/mstechdays>

Session « L'histoire de HTML5 pour les développeurs Windows Phone 8 » : <http://fr.slideshare.net/davrous/lhistoire-dhtml5-pour-les-developpeurs-windows-phone-8>

Session « développez des applications mobiles avec un cloud » : http://fr.slideshare.net/Microsoft_France/dveloppez-des-applications-mobiles-ios-android-windows-phone-et-windows-8-avec-un-cloud-super-simple

Session « Mécanismes internes de la sécurité de Windows 8 » : http://fr.slideshare.net/Microsoft_France/mcanismes-internes-de-la-scurit-de-windows-8

Session « Amadeus Hotel Call Center » (WPF) : http://fr.slideshare.net/Microsoft_France/concevoir-et-valider-larchitecture-dune-application-wpf-modulaire-avec-visual-studio-2012

HTML5+jQUERY

Pour tout Navigateur - Plateforme - Application - Appareil



IGNITEUITM
INFRASTICS JQUERY CONTROLS



Telecharger gratuitement la version d'essai
www.infragistics.com/igniteui-trial

INFRASTICSTM
DESIGN / DEVELOP / EXPERIENCE

Infragistics Sales France +33 (0) 800 667 307 • US 800 231 8588 • Europe +44 (0) 800 298 9055 • India +91 80 4151 8042 • APAC +61 3 9982 4545

Copyright 1996-2013 Infragistics Inc. Tous droits reserves. Infragistics et NetAdvantage sont enregistres comme etant marque deposee. Le logo est marque deposee Infragistics Inc. Toute autres marque deposee ou enregistree est deposee dans le respect de la propriete de celui qui la detient.

Intel HTML5 App Porter Tool : une vraie-fausse bonne idée de conversion Objective-C ?

Mi-février, Intel proposait une bêta d'un nouvel outil de conversion de code Objective-C en HTML5 / JavaScript : HTML5 App Porter Tool.

Vous l'aurez compris, le but de l'outil est de convertir un projet iOS (donc un projet Objective-C provenant de Xcode) en projet HTML / CSS / JavaScript, en utilisant notamment JSON. Intel précise bien que cet outil ne fera pas de conversion à 100 % des projets iOS mais il est là pour faciliter le travail et aller plus rapidement, notamment sur le code « basic » : navigation, interfaces « simples », etc. Et nous l'avons constaté durant nos tests : l'outil évite de tout réécrire mais il demeure un important travail de refactoring, de réécriture partielle et surtout de mapping manuel des appels API non convertis, ce qui peut rapidement devenir fastidieux.

> On parse, on mappe, on génère

Fondamentalement, HTML5 App Porter Tool permet de :

- faire une conversion du code Objective-C en code JavaScript
- de transcrire les appels et les typages API d'iOS en objets et appels JavaScript / HTML 5
- les couches des vues provenant d'Interface Builder migrées en fichiers HTML et CSS
- les fichiers projets Xcode en projet Visual Studio 2012.

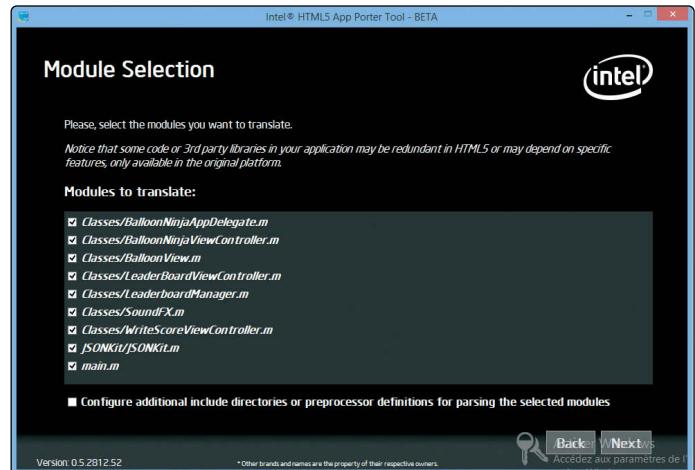
Précisons d'emblée que l'outil d'Intel fonctionne uniquement (pour le moment ?) avec Windows 8 et Visual Studio 2012. Un point négatif. Pourquoi ne pas proposer un support d'Eclipse par exemple ? Le code Objective-C vers JavaScript passe par un format intermédiaire d'abstraction ainsi, le code est parsé dans un arbre syntaxique intermédiaire

re avant de générer en sortie les fichiers de code JavaScript : le code converti et les API non supportées. Les appels API d'iOS sont mappés en appels JavaScript. Les appels API non supportés sont stockés dans un espace réservé. Puis, l'outil génère les fichiers finaux HTML5 / CSS et JavaScript.

> Quid des frameworks supportés ?

La prudence d'Intel est de mise : tout n'est pas possible avec la version bêta et tout ne sera pas possible. iOS s'appuie sur des milliers d'API réparties entre de nombreux frameworks allant de la librairie de base, comme le Foundation framework, au framework d'animation ou de cartographie. Actuellement, deux frameworks sont supportés (pas totalement) : UIKit et Foundation. Les autres frameworks ne sont pas supportés, ou alors limités à quelques classes.

Un projet iOS comprend, principalement : des fichiers projets, codes et d'interfaces. La partie interface est liée à Interface Builder et aux fichiers XIB. Ces fichiers sont convertis par l'outil en fichiers HTML et CSS. Les connexions, événements, contrôleurs de ces fichiers sont mis dans des fichiers .js. Attention, Intel liste les widgets et propriétés supportés. Les interfaces trop complexes risquent d'être mal ou pas converties, méfiance aussi sur les interactions trop nombreuses.

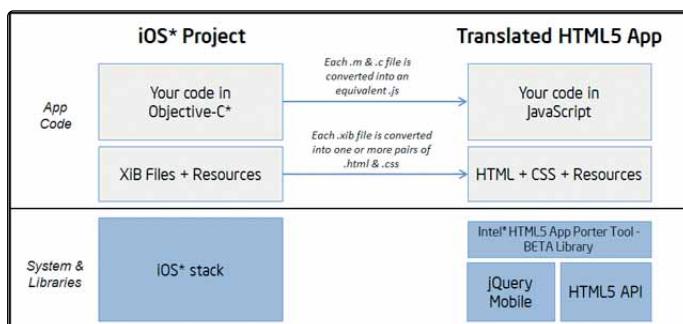


> App Porter Tool, puis VS 2012

Après avoir récupéré le projet Xcode, vous devez ouvrir HTML5 App Porter Tool puis indiquer le dossier contenant les fichiers projets (.pbxproj). L'outil demande les modules à convertir et le processus de conversion s'occupe du reste. Selon la taille du projet, la génération sera plus ou moins longue. À la fin du processus, vous aurez accès à deux rapports : translation report et ToDo Report. Le premier fournit le détail de mapping entre les .m et .js. Le rapport ToDo donnera les tâches à faire pour finir le portage. L'outil donnera une indication du taux de conversion. Nous avons effectué quelques tests sur l'exemple fourni par Intel et avec des exemples officiels d'Apple : aucune conversion ne s'est exécutée immédiatement. Il faut modifier, rajouter, supprimer du code. Du refactoring sera à prévoir. Parfois, les lignes de code JavaScript pourront être mal formées. D'autre part, l'outil génère beaucoup de fichiers JS / HTML / CSS, ce qui n'est pas sans inconvénient pour la qualité du code, sa structure et tout simplement pour s'y retrouver. La licence accordée est valable jusqu'au 8 août 2013. A voir comment évoluera l'outil.

Site : <http://software.intel.com/fr-fr/html5>

Documentation : <http://software.intel.com/fr-fr/articles/technical-reference-intel-html5-app-porter-tool-beta>



Translation results

85% of the Objective-C* API calls in your project have been translated successfully.

Translated Objective-C* files into JavaScript

All JavaScript files have been located at C:/Users/tonic/Downloads/PhotoScroller/PhotoScroller/HTML5_Conversion,

Source file	Translated file
PhotoScroller/Classes/ImageScrollView.m	ImageScrollView.js
PhotoScroller/Classes/PhotoViewController.m	PhotoViewController.js
PhotoScroller/Classes/TilingView.m	TilingView.js
PhotoScroller/main.m	main.js

» Zend met à jour le Server et le Studio.

Zend Studio 10 se focalise tout naturellement sur le développement PHP avec le support complet de PHP 5.4. Le développeur bénéficie d'une contextuelle, d'un mode debug plus rapide (PHP et JavaScript), de l'intégration avec le serveur, livré avec Zend Framework 1.12 et 2.1. Zend Studio simplifie la création d'applications mobiles avec un back-end serveur, afin que même les développeurs sans expérience préalable puissent réussir rapidement. Le développeur crée des applications tournant sur Apache Cordova (PhoneGap) et qui fonctionnent sur de nombreux systèmes mobiles. L'outil est disponible dès maintenant, prix normal : 299 €. Zend Server 6 doit permettre de contrôler et d'assurer une qualité de service (SLA). Et le serveur est disponible sur les Cloud IBM, Amazon, VMware, Red Hat. Une version gratuite de Zend Server 6 est également annoncée. Elle est destinée aux développeurs qui veulent tirer parti de ses avantages en termes de débogage et mesure des performances durant le développement.

Tout comme les versions payantes, elle dispose d'une stack PHP certifiée, du profilage et du débogage PHP.

» Spring prépare l'arrivée de la v4 de son framework.

L'éditeur, filiale de VMware, a dévoilé les principales nouveautés de la prochaine version de sa librairie. La version 4, qui doit arriver courant 2013, supporte Java 8, Java EE 7, le WebSocket. Ces nouveautés permettront d'être au plus près des évolutions de la plateforme Java. Détails sur : <http://blog.springsource.org/2013/01/16/next-stop-spring-framework-4-0/>

» Selon le baromètre TIOBE, Java redevient le langage le plus demandé (18,387 %)

après avoir été supplanté par C durant quelques mois. En 2e place, nous retrouvons C (17,080 %) et enfin en 3e position, Objective C qui fait un nouveau bond important (+ 2 places) et frôle les 10 %. C++ arrive 4e. Les trois premiers langages monopolisent 45 % des demandes ! Ce classement indique seulement la popularité des langages dans les requêtes sur les moteurs de recherche et non l'utilisation effective des langages.

Facebook complète les outils pour les développeurs.

Après avoir mis en place un agenda des changements et des nouveautés API, Facebook rajoute un élément intéressant pour tout développeur : les alertes d'erreurs API. Cet outil doit prévenir le développeur de tout problème d'API par une notification et par mail. Pour en savoir plus : <https://developers.facebook.com/docs/alerts/>

Apps > CastLine

[Edit App](#) [+ Create New App](#)

[See All](#)

Settings [Edit Settings](#) **Developer Alerts**

App ID/API Key 221899141232550
App Secret 0f4d86aae0ef7bdd3665f351bdd
App Namespace castline
Sandbox Mode **Off**
Platform Integrations Website with Facebook Login

App Center Review
Your App Details page has been submitted and is pending review.
4 seconds ago

Privacy Policy Compliance
Please update your privacy policy to be compliant with our Platform Policies.
[fix now](#) 55 seconds ago

APP CAMPUS

BOOST YOUR APP IDEAS

LEARN MORE WWW.APPCAMPUS.FI

© 2012 APPCAMPUS. ALL RIGHTS RESERVED AND APPLIED FIRMLY

ACCÉLÈRE TON SUCCÈS
reçoit de **20 000 à 70 000€**

→ une seule adresse : www.appcampus.fi

Et pour suivre l'actualité des développeurs Nokia en France :

www.facebook.com/NokiaDevFrance

www.nokiadevblog.fr

@NokiaDev_France

Tout savoir sur Devoxx France 2013

Devoxx France est une conférence organisée à Paris par l'équipe du Paris JUG, depuis 2012. Durant 3 jours, cette deuxième édition proposera des présentations et événements, principalement en français, autour de notre métier de développeur. C'est donc l'occasion de réunir pendant ces 3 journées (du 27 au 29 mars 2013) les développeurs, les architectes et les décideurs techniques autour de cinq grandes thématiques. L'équipe de Devoxx France vous présente les tendances et les nouveautés, en avant-première.



D.R.

> Le quotidien du développeur

La thématique «Développeur au quotidien» a pour ambition de couvrir l'ensemble de ce qui fait la vie d'un développeur. Tous les jours nous interagissons avec un écosystème riche et complexe aussi bien sur le plan technique qu'humain. Et ces dernières années les changements sont nombreux ! Entre l'utilisation des pratiques agiles, l'impressionnant boum des plateformes de e-learning, en passant par l'émergence du Cloud, notre façon de travailler évolue quasi quotidiennement. Nous allons donc vous donner la possibilité de sortir de votre zone de confort, peut-être même de remettre en cause quelques-unes de vos certitudes pour vous emmener encore un peu plus près de l'excellence et de ce que sera notre métier demain.

Nous vous proposons cette année de découvrir des outils et pratiques, qui feront de vous un développeur plus pragmatique et plus efficace dès votre retour de Devoxx France le lundi matin !

> Parlons un peu de DevOps !

La thématique «Infrastructure, hardware d'hier et de demain» a pour ambition de discuter un peu de ce qu'il y a sous le capot avec un aspect prospectif. Nous souhaitons aborder les aspects infrastructure, sécurité, des problèmes très bas niveaux de serveurs comme la consommation des data centers, et de tout ce qui nous permet de faire fonctionner notre quotidien de développeur. Cela s'ouvre bien entendu aux réponses DevOps, mais aussi aux sujets plus spécifiques que

sont l'embarqué et la domotique. Ce track est novateur. C'est une première cette année, car il est un peu en marge des sujets classiques de la conférence qui se focalise traditionnellement sur les langages tournant sur la JVM, les technologies du web et l'environnement de développement.

Il nous a semblé qu'au crépuscule du minitel et du X25 chez France Télécom, et à l'aube du tout internet, nous arrivions à une étape, et qu'il pourrait être agréable de faire l'écart entre hier et demain dans notre Hardware. Bien sûr nous serons dans les thèmes d'aujourd'hui, mais nous ouvrirons la porte à une certaine histoire ou préhistoire de l'informatique.

> Java, toujours au cœur de l'événement

La thématique Java est le cœur de la conférence Devoxx France. La JVM, les patterns, les principes de développements, les librairies, les tests, autant de sujets sur lesquels les intervenants exposeront leur points de vue, partageront leurs expériences, ou présenteront un outillage novateur. Ces dernières années nous observons une effervescence croissante de la communauté et l'exploration de concepts nouveaux. Avec les avancées réalisées ou les connaissances acquises par la communauté en France ou dans le monde, Devoxx est une opportunité efficace pour en profiter. Nous aurons par exemple quelques personnes qui travaillent sur le cœur de la JVM, d'autres qui établissent le standard Java EE, avec pour rappel une sortie de Java EE 7 prévue pour mai. Nous aurons aussi l'honneur de recevoir des personnes ayant achevé un travail remar-

quable sur les performances de leur application Java. Chaque année sera différente et offrira de nouvelles perspectives, affaire à suivre donc.

La thématique « Jeux, Web et Mobile » vous emmènera au cœur des derniers frameworks ou outils pour offrir une expérience riche et avancée aux internautes desktop ou mobile sans faire aucun compromis vis-à-vis de l'industrialisation. Sir Tim Berners Lee a inventé le Web en 1989. Il est depuis en révolution permanente. La frénésie de l'arrivée des possibilités offertes par HTML5 n'est pas finie, et pourtant la tendance est à l'industrialisation des technologies standards du web. Cela rapproche les développeurs Java des technologies du navigateur. Par ailleurs, la tendance est à une séparation de plus en plus stricte entre les serveurs qui délivrent des services REST et des clients web ou mobiles riches qui gèrent l'ergonomie, la présentation et le design.

Nous parlerons de frameworks de SPA, de performances front-end, des API mal connues d'Android ou encore d'API HTML5 permettant de réaliser des Jeux « lourds » dans un navigateur. Evidemment, tout cela se fera avec la présence de quelques stars, comme Devoxx France sait si bien le faire... Avec la thématique « Langages alternatifs » vous irez à la rencontre de peuplades mythologiques et pourtant bien réelles comme les Haskelliens, les Scalaïstes ou ces clojurians ("non rien de rien...") qui vivent dans des forêts sauvages reculées, loin de toute civilisation. Peut-être même croiserez-vous des développeurs qui ne sont pas sur la JVM ! Vous découvrirez l'usage des dialectes cabalistiques et aurez peut-être la chance de

boire avec de vrais locaux, le verre de l'amitié que certains appellent la li-monade.

Vous apprendrez qu'avec un éditeur de texte (sorte d'IDE de ces contrées primitives) on peut tronçonner en rondelles d'immenses arbres de données (appelés baobabs "BigData") ou invoquer des centaines de démons en criant : "Go".

Enfin, vous découvrirez comment ont été rajoutés dans le dernier JDK8 certains ingrédients exotiques, pour permettre à tous les habitants de Java de bénéficier des bienfaits de la programmation fonctionnelle.

> Une track pour les DSi

Grande nouveauté cette année à Devoxx France, la thématique « après-midi des DSi ». Cet espace d'échange dédié aux décideurs a pour but de savoir comment une technologie émergente s'intègre dans les SI. Le thème choisi cette année sera le Cloud. Huit exemples concrets pour des entreprises allant de la start-up à l'institutionnel vont être abordés par des DSi.

Les cas d'utilisations vous éclaireront sur les avantages mais aussi les écueils du Cloud. Ils décriront les impacts techniques et organisationnels de ce service dans leur entreprise. A l'issue de cet après-midi, vous aurez en main des éléments clés pour vos arbitrages. Vous voulez en savoir plus sur le Cloud, réservez vite le jeudi 28 mars 2013 entre 14h et 18h.

Comme vous pouvez le constater, Devoxx France c'est avant tout la voix du développeur. Nous souhaitons défendre une vision où le développeur est un passionné, un artisan et un maillon indispensable de la réalisation de vos projets informatiques.

Nous comptons réunir entre 1200 et 1400 personnes du 27 au 29 mars 2013. La journée commence par un petit déjeuner d'accueil, puis un déjeuner et une pause-café. Nous avons réservé 6 salles d'une capacité d'environ 250 personnes, dans lesquelles auront lieu les conférences.

Il y a plusieurs styles de présentations allant de 15 minutes (Quicky) à 3 heures (Université) ou de la plénière du matin à la conférence plus classique d'une heure.

Nous avons aussi 3 salles de plus petite taille qui accueilleront des ateliers et des conférences informelles. Ces salles donnent sur un hall d'exposition de 660 m², où nos partenaires tiennent leurs stands et animent les pauses. Lors de l'édition 2012, des événements inattendus ont eu lieu : tournois de babyfoot, danses de robots, fabrications de

sushis, etc... L'ambiance est studieuse, mais aussi festive ! Au total, il y aura environ 140 présentations et une vingtaine de sponsors avec leur stand.

Pour ce qui est des tarifs, vous pouvez venir uniquement le mercredi pour les universités à 150€, pour les conférences, le jeudi et vendredi à 350€ HT pour les deux journées, ou les 3 jours à 450€ HT. Et grâce à notre partenaire Oxiane vous pouvez utiliser votre droit individuel à la formation (DIF) pour vous inscrire.



> Information pratique

Lieu : Hotel Marriot Rive Gauche (Paris 14e)

Alors n'hésitez plus et venez rencontrer vos pairs à Devoxx France.

Pour en savoir plus : <http://www.devoxx.com/>

L'ECOLE DE LA GENERATION NUMERIQUE

C'EST UN PEU PRETENTIEUX ET POURTANT C'EST VRAI !



Iconoclaste ou surprenante, d'un dynamisme insoupçonné quand il faut lancer un projet ou une entreprise, telle est la génération numérique et son école, Epitech. Depuis 15 ans, par une pédagogie qui assume sa différence, par une confiance dans nos étudiants et dans leurs projets audacieux, nous apportons aux entreprises ce nouveau vent de passion indispensable. La génération numérique transforme chaque jour le monde. Son école a un nom, Epitech, l'école de l'innovation informatique.

www.epitech.eu



Établissement d'enseignement supérieur privé

Une pluie de nouveaux SSD !

Le disque SSD se retrouve de plus en plus en standard sur les ordinateurs, particulièrement sur les portables. L'hybride de disque dur - SSD commence à s'implanter pour mixer les performances et les capacités de stockage, comme la (très chère) option d'Apple.

PNY sort une nouvelle famille de SSD dédiés aux entreprises et serveurs : SSD Prevail et Prevail Elite. Pour PNY, ces disques offrent des performances et une robustesse suffisante pour être utilisés avec des applications gourmandes en ressources et performances disques comme la CAO. Le constructeur met en avant les performances systèmes et les logiciels comme le temps d'ouverture. Ces SSD montent jusqu'à 85 000 IOPS et les capacités vont de 120 à 480 Go. Le contrôleur est un SanDisk SF 2281 (pour les deux modèles). Il repose sur du SATA III / 6G. Par contre, la mémoire utilisée varie entre le Prevail et le Prevail Elite. A partir de 119 €.

De son côté, Intel lance un nouveau SSD mSata dédié aux ultrabooks (et NUC ?) : SS Intel 525 Series 6 go/s. Côté performance, il affiche un IOPS de 50 000 (maximum), et une écriture

séquentielle de 520 Mo/s. Intel travaille beaucoup sur les traitements antichoc pour éviter corruption et pertes de données. « *L'Intel SSD 525 ouvre la voie vers un nombre virtuellement illimité de solutions embarquées créatives, dans des domaines tels que le divertissement en vol, les stations de travail mobiles, les micro-serveurs ou les téléphones sur IP* », indique le constructeur. La capacité varie de 30 à 240 Go. Il possède une fonction de cryptage AES 128 bits et une garantie de 5 ans. Intel propose aussi la SSD Toolbox, une suite d'outils de diagnostic et de gestion des SSD, malheureusement réservée à Windows !

Toshiba est aussi un important constructeur de SSD. Il vient d'annoncer de nouveaux mécanismes de cryptage et d'effacement sécurisé pour ses disques de la gamme PX (SSD à multiniveaux de cellule, architecture standard). L'un des principaux intérêts de ces disques est l'effacement sécurisé avec Sanitize. Toshiba propose aussi aux constructeurs de portables d'inclure ce type de fonction par défaut en cas de vol ou d'intrusion malveillante. Certains modèles ne seront disponibles qu'au 1er trimestre.



Le Kinect like en approche !

Leap Motion, le principal concurrent de Microsoft Kinect, arrive doucement mais sûrement. Leap est un petit boîtier que l'on connecte à son ordinateur pour pouvoir contrôler, utiliser des logiciels ou jouer avec les gestes de la main et des doigts, comme on peut le faire avec Kinect. Il s'agit d'une nouvelle approche de l'interface naturelle, le tout tenant dans un boîtier de la taille d'un iPod. Les principaux usages sont : le jeu, l'interactivité dans des mondes et objets 3D, créer des contenus directement sur l'écran même s'il n'est pas tactile. Surtout il fonctionne aussi bien sur PC que Mac. Autre argument le prix : 69,99 \$!

Actuellement, Leap est toujours en pré-commande (il devrait être disponible courant février) mais les développeurs peuvent d'ores et déjà obtenir le kit de développement (le constructeur sélectionne les développeurs). Plusieurs milliers ont été envoyés depuis décembre. Le développeur est au centre de la problématique Leap pour créer des applications optimisées et compatibles et surtout créer les usages. Le SDK comprend le téléchargement du SDK proprement dit, l'accès aux documentations techniques, les outils de développement et un prototype du Leap Motion. Mais vous pouvez accéder au SDK sans avoir le matériel. Le constructeur s'est associé à Asus pour proposer le boîtier par défaut sur certains modèles...



Le constructeur français Archos continue à miser sur le marché des tablettes, que ce soit en sortant des tablettes Android ou des accessoires. Archos a dévoilé fin janvier un **clavier Bluetooth pour iPad**. Il permet de caler l'iPad et d'avoir une sorte de station complète. Mais il ne possède aucun connecteur pour brancher la tablette. Prix : 69 € TTC. Autre nouveauté,

l'Archos 97 Titanium HD. Il intègre un écran IPS, Android 4.1, l'accès à Google Play, Archos Media Center et un processeur 4 cœurs 1,6 Ghz. Sur la connectique, nous retrouvons une mémoire flash de 8 Go, une sortie mini-HDMI, un slot microSD. Prix : 169 € TTC. Début janvier, le constructeur avait présenté son Archos TV, une sorte de Google TV qui devrait être disponible courant février au prix de 129 €.



Une infinité d'applications une seule plate-forme

avec l'environnement de conception
de systèmes le plus complet qui soit



NI LabVIEW est le seul environnement de développement qui offre autant de possibilités d'intégration matérielle ; au point de vous permettre de relever n'importe quel défi en matière d'applications de mesure et de contrôle. LabVIEW est au cœur de ce que l'on appelle la conception graphique de systèmes : une approche qui combine les avantages d'une plate-forme logicielle ouverte et d'un matériel reconfigurable à l'infini, pour que vous dévelopez votre système en un minimum de temps.

Le logiciel de conception de systèmes LabVIEW offre une intégration matérielle inégalée et vous permet de programmer de la façon dont vous pensez : graphiquement.



>> Concevez beaucoup plus rapidement vos systèmes en visitant ni.com/labview-platform/

01 57 66 24 24

NATIONAL INSTRUMENTS France • 2 rue Hennape - 92735 Nanterre Cedex, France • Tél. : (0)1 57 66 24 24 Fax : (0)1 57 66 24 14 • Société de droit américain - capital social 1000,00 dollars • US - 11500 N Mopac Expwy, Austin-Texas USA - 10056236 - 344 497 649 RCS Nanterre - SIRET B 344 497 649 00022 - APE 516J - N.I.I. FR 57344497649

©2012 National Instruments. Tous droits réservés. National Instruments, NI, LabVIEW, et ni.com sont des marques de National Instruments. Les autres noms de produits et de sociétés mentionnés sont les marques ou les noms de leurs propriétaires respectifs. Pour plus d'informations concernant les marques de National Instruments, veuillez vous référer à la partie Terms of Use sur le site ni.com/legal/. 08371

 **NATIONAL
INSTRUMENTS™**

Inria sort la nouvelle version de son projet OpenVIBE

OpenVIBE est une technologie pour pouvoir piloter un ordinateur, des logiciels par la pensée. OpenVIBE 2.0 s'ouvre largement aux jeux vidéo. Pour rappel, OpenVIBE est un projet d'interface cerveau-ordinateur de l'INRIA, avec le soutien de plusieurs partenaires. Des éditeurs de jeux ont activement participé à cette évolution (Ubisoft, Kyotonn Games...). Il aura fallu presque 3 ans pour sortir cette nouvelle plateforme. Une interface cerveau-ordinateur permet à ses utilisateurs d'envoyer des commandes à un ordinateur en utilisant uniquement leur activité cérébrale. L'activité électrique émise par le cerveau est analysée en temps-réel et traduite en commande pour un ordinateur, ou tout autre système automatisé, permettant ainsi de piloter un curseur vers la droite ou la gauche, simplement en imaginant les mouvements de sa main. Le projet OpenViBE2 fait suite à ce premier projet, et vise à explorer plus en avant ces technologies en les étendant dans



un contexte applicatif particulièrement complexe : celui des jeux vidéo. L'objectif est d'améliorer les interfaces cerveau-ordinateur. Pour les chercheurs et les éditeurs, la commande mentale est une nouvelle voie prometteuse dans les jeux mais les chercheurs n'oublient pas l'utilisation de cette technologie pour les handicaps lourds. Un des problèmes est de pouvoir traiter efficacement les signaux du cerveau : filtrer les signaux, les interpréter. Cela passe notamment par l'élimination des bruits parasites tels que les mouvements musculaires, mouvements de mâchoire,

etc. Bref, utiliser uniquement l'activité cérébrale utile. De ce projet, les chercheurs ont pu découvrir de nouveaux concepts d'interactions : interface multi joueur, adaptation automatique du monde virtuel à l'état mental du joueur. Les industriels du jeu vidéo ont travaillé directement avec des laboratoires académiques de l'Inria, de l'Inserm, du CEA et GIPSA-Lab, pour mettre au point des jeux vidéo basés sur l'activité cérébrale. Cette collaboration a permis de mieux spécifier les actions de

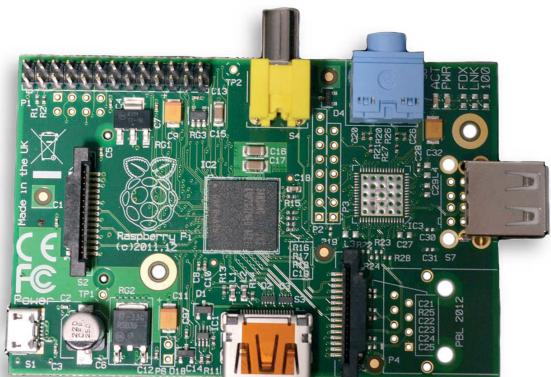
recherche scientifique tout au long du projet et de converger vers des solutions plus adaptées aux contraintes technologiques du domaine d'application. Un projet de standardisation a par ailleurs été piloté par UBISOFT pour définir le casque « idéal » afin de mieux correspondre au marché du jeu vidéo. Des jeux ont été développés conjointement avec l'INRIA pour tester et affiner les architectures logicielles et matérielles. Si le projet a bien évolué et permet d'ouvrir une nouvelle perspective, il faut maintenant passer au stade industriel et proposer des solutions concrètes. Le premier projet n'avait pas pu atteindre ce stade. Site : <http://openvibe.inria.fr>

HP lance son Chromebook

Très discret depuis quelques mois, le système connecté de Google, ChromeOS, a encore du mal à trouver son marché et à multiplier les machines le supportant. HP a dévoilé début février, un portable spécifique : le HP Pavilion 14. Il possède un écran 14", pèse tout de même 1,8 kg (ce qui est assez lourd), 2 Go de mémoire vive, un disque SSD de 16 Go (taille ridicule). L'autonomie serait de 4h15 et le tarif d'environ 240 euros. Reste à voir l'intérêt réel de cette machine.

Raspberry Pi Modèle A est disponible

Le Raspberry Pi Modèle A contient 256 Mo de mémoire vive (RAM), un port USB et, contrairement au Modèle B existant, ne possède pas de connexion Ethernet. Par conséquent, en plus d'être vendu à un prix inférieur, le nouveau Modèle A prend moins de place et consomme beaucoup moins d'énergie que le Modèle B. « L'arrivée du Modèle A est une bonne nouvelle pour les personnes qui souhaitent développer des applications à faible consommation électrique avec leur Raspberry Pi », se réjouit Glenn Jarrett, Global Head of Product Marketing chez RS. « Le prix proposé le

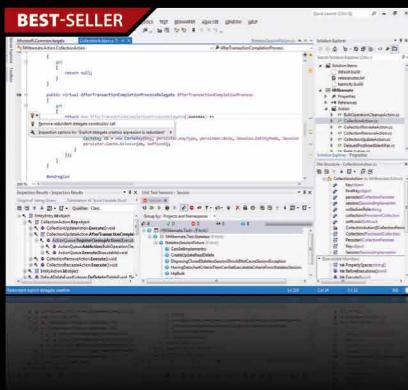


rend encore plus intéressant pour les intégrateurs de systèmes et plus accessible dans les domaines professionnels où le budget est limité, comme par exemple le monde de l'éducation ».



L'éditeur Canonical a montré les premiers prototypes de smartphones utilisant son système mobile basé sur Ubuntu

Le système devrait être disponible courant 2013 et les premiers terminaux en d'année. Mais reste à trouver des constructeurs, à convaincre les développeurs et éditeurs. Il pourrait utiliser des applications natives et HTML5. Ubuntu for Android permet d'installer et d'utiliser Ubuntu depuis un terminal mobile Android...



ReSharper

à partir de € 308



Outil de développement Microsoft Visual Studio 2005-2012 pour améliorer la productivité.

- Analyse de qualité du code à la volée dans C#, VB.NET, XAML, ASP.NET, MVC, JS, CSS, HTML, XML
- Accès ultrarapide à tout fichier, type ou membre dans votre base de code
- Signale les problèmes de code et propose des solutions rapides de correction
- Assistants de codage incluant IntelliSense et transformations instantanées du code
- Refactorisations de la solution pour changer la base de code en toute sécurité



Aspose.Words for .NET

à partir de € 747



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



GdPicture.NET

à partir de € 2 992



Outils complets d'imagerie documentaire et de gestion pour les développeurs.

- Capturez, traitez, créez, affichez, modifiez, annotez, composez, fractionnez, fusionnez et imprimez des documents depuis vos applications Windows et Web
- Lisez, écrivez et convertissez les images vectorielles et raster en plus de 90 formats, dont PDF, PDF/A, TIFF, GIF, JPEG, PNG, JBIG2, WMF, BMP, WBMP, ICO, PCX, PNM, XPM, JPEG 2000, HDR, PSD, TGA, PICT, EXR, DDS, PPM, SGI, PBM, PGM, PFM, XBM, IFF et le format graphique RAW



DXperience DXv2

à partir de € 1 112



Tous les outils DevExpress ASP.NET, WinForms, Silverlight, WPF et IDE Productivity en un.

- Abonnement de 12 mois pour tous les produits et mises à jour DevExpress et accès aux versions bêta en développement actif
- Modèles et thèmes d'application intégrés exceptionnels
- Support de nouvelle vue Windows 8 UI et panneaux ancrables tactiles
- Support interface codée pour test environnement utilisateur



Antonio, du JUG Paris à Devoxx

Antonio Goncalves est développeur senior Java EE. Indépendant depuis 7 ans, il a travaillé auparavant en SSII et chez BEA Systems. Ingénieur informatique des Arts et Métiers, son parcours a été sinueux dans des universités à l'autre bout du monde (Brésil) ou plus près de nous (Espagne et Grande Bretagne). Il est un des créateurs du Paris JUG, de Devoxx France, du podcast Les Cast Codeurs, et l'auteur de trois livres sur Java EE. Il a reçu le titre de « Java Champion » et plus récemment, a fait partie du palmarès des 100 personnes les plus influentes en informatique pour 2013.

Comment es-tu tombé dans l'informatique et plus spécialement dans le développement ?

Antonio : Grâce aux jeux. A 13 ans j'ai eu un Commodore 64 à cassettes et j'ai commencé à développer mes propres jeux en Basic. Puis j'ai eu l'extension Power Cartridge qui m'a permis de développer en assembleur. Je me souviens avoir développé un jeu de casse-briques avec effets 3D... sur écran monochrome bien sûr. Je suis un des rares à avoir fait un bac informatique (le Bac H à l'époque). Puis un jour, au salon Sicob, je suis tombé nez à nez sur une station Next. J'y ai vu l'avenir et j'ai voulu participer à cet avenir. J'ai ensuite développé en C, en Ada, Smalltalk, j'ai fait mon mémoire de recherche en Prolog, j'ai même mis les mains dans du Cobol, et découvert Java en 1998. Il ne se passait rien en France autour de ce langage à cette époque là, alors j'ai tout plaqué pour aller faire du Java à Londres durant 2 ans. Aujourd'hui, je continue à développer, même si mes missions s'enrichissent de conseils, de mentoring ou de formations.

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

Antonio : Je déteste penser que c'est de l'iné... mais j'ai l'impression qu'il y a quelque chose de cet ordre. Etant gamin je jouais aux

LEGO, aux mécanos, je désosais des radios pour récupérer les transistors, les leds, et les souder sur des circuits électroniques pour voir ce que cela faisait. J'ai retrouvé le côté "mécano" lorsque je développais en assembleur étant adolescent, car j'étais très proche du système. Aujourd'hui, je m'amuse à désosser des serveurs d'applications pour voir ce qu'ils ont sous le capot. J'ai donc l'impression d'exercer toujours la même passion depuis plus de 30 ans.

Tu as gardé un regard très geek : gadget, veille techno, c'est important pour ton job et ta passion ?

Antonio : La veille techno est primordiale : toujours à l'affût de la dernière librairie en Alpha, toujours en quête de l'API qui va me simplifier la vie, toujours besoin de tester de nouvelles fonctionnalités du langage. Par contre, pour ce qui est des gadgets électroniques, ce n'est pas trop mon truc. J'ai un téléphone Android... et c'est à peu près tout (je n'ai même pas de télé à la maison).

Etre développeur n'est pas toujours facile : pression, évolution constante, frustration des projets et des "chefs", c'est quoi pour toi d'être développeur aujourd'hui ? Le job a-t-il changé depuis tes débuts ?

Antonio : On ne va pas non plus trop se plaindre, il y a pire comme boulot. Comme je le disais, dans le fond, mon job n'a pas vrai-



ment changé : il y a 30 ans j'imbriquais des mécanos, aujourd'hui j'empile des API. En revanche, au quotidien il y a eu des révolutions. J'ai commencé l'informatique à une époque où il n'y avait ni internet (du moins, pas pour le grand public ni pour les sociétés) ni adresse e-mail. Les compilations étaient coûteuses, il fallait donc relire son code à deux fois. Aujourd'hui, je passe ma vie sur internet, je télécharge des librairies open source, je compile en continu sur mon IDE, je communique sur des forums, je partage mon code sur GitHub... à ce niveau, je peux dire que mon travail aujourd'hui n'a plus rien à voir avec celui des débuts.

Mais cette évolution constante, bien que génératrice d'une forme de stress (nous sommes des éternels bêta-testeurs), fait partie intégrante de notre travail, et c'est pour cela que j'aime faire ce métier : j'apprends tous les jours de nouvelles choses (même si elles ressemblent souvent aux mécanos de mon enfance).

Pour ce qui est des projets.... malheureusement, de ce côté là non plus ça ne change pas. Le cerveau d'un être humain ne sait naturellement pas gérer les projets de grande taille (disons, un projet qui implique plus d'une personne). Nous devons les découper en sous-projets, en tâches, en sous-tâches, y affecter des êtres humains, diviser, multiplier puis cumuler le nombre de jours/hommes pour avoir une estimation du

coût, pour ensuite dériver, revoir les estimations et les budgets à la hausse... pour bien souvent arrêter le projet. Pas mal de fiasco dans notre métier. Il suffirait simplement d'avouer que les humains ne savent pas planifier les gros projets informatiques et qu'il faudrait juste monter des équipes de développeurs passionnés (juniors, seniors), motivés, et les laisser faire de leur mieux en se basant sur des itérations courtes. Et arrêter le projet rapidement si l'on voit qu'on a pris la mauvaise voie. Mais nos chefs utilisent des techniques de gestion de projets issues du BTP, non adaptées à notre métier. Les ponts et les cathédrales sont faits pour durer des siècles, les projets informatiques dépassent rarement la décennie. Pourquoi appliquer les mêmes techniques de gestion de projet ? Nous continuons donc à errer de projets coûteux en projets en retard. Point optimiste quand même, je travaille de plus en plus sur des projets agiles où développeurs, utilisateurs et opérationnels vont dans le même sens, ce qui change un peu la donne.

Et en dehors du boulot, qu'est-ce que tu aimes faire ? Comment trouves-tu l'équilibre entre travail, vie privée, passion, famille ?

Antonio : En dehors de l'informatique, qui reste une vraie passion pour moi, j'ai deux autres passions : ma fille et la musique. Ma fille grandit, j'arrive donc à mêler les deux et l'emmène voir des concerts. A 8 ans elle aime Michel Petrucciani et chante déjà les Pink Floyd. Ayant fait de l'histoire de l'art, j'écume aussi les expositions et galeries parisiennes... et puis, retour à mes mécanos, j'adore faire la cuisine. C'est une manière de continuer dans la geekitude, c'est-à-dire empiler des ingrédients, fouiller dans des épices, essayer de nouveaux mélanges, découvrir de nouveaux goûts.

Peux-tu nous présenter ton quotidien en quelques mots ?

Antonio : Si par quotidien tu penses à "activité qui se répète tous les jours", alors je ne peux rien présenter. Ce qui me passionne dans mon métier, et mon statut d'indépendant, c'est que les semaines ne se ressemblent pas. Par exemple, je suis en ce moment enfermé chez moi à écrire un livre sur Java EE 7. Dans quelques semaines je commencerai une mission de conseil chez un client. Puis je m'arrêterai pour préparer et participer à Devoxx France. Je dois ensuite développer une application, pour enfin

“ne devenez pas chef de projets”

retourner chez un nouveau client. Le reste de l'année va être cadencé par la publication de mon livre, donc, la tournée des conférences européennes pour en parler, et d'autres clients qui auront d'autres besoins.

Comment vois-tu ton job évoluer ?

Antonio : Je ne peux là encore répondre à cette question. Elle me rappelle trop les RH

Antonio et son bureau

« Comme je n'ai plus le temps d'assembler les mécanos car j'utilise ce temps à empiler des API, je n'ai plus non plus le temps de bidouiller mes ordinateurs. J'ai arrêté d'assembler des pièces détachées pour avoir des serveurs Windows boostés (censés être puissants), arrêter de compiler les drivers wifi pour accéder à internet sous Ubuntu. Je suis donc passé de l'autre côté de la force, je suis sous Mac. J'ai un Mac Book Pro, deux Mac Mini, un écran 27 pouces et un switch pour passer de l'un à l'autre. Je n'utilise plus de souris mais le track pad. Super important, je n'ai presque plus de papier ou de stylos sur mon bureau. J'ai un scanner extra rapide, je scanne donc tous les documents papier que je reçois puis les jette. J'ai aussi une Dymo pour imprimer des petites étiquettes à coller sur les enveloppes pour ne plus avoir à utiliser de stylo. Et bien sûr, vous l'aurez compris, je travaille toujours toujours en musique. Essentiellement du jazz instrumental pour pouvoir me concentrer. »

des SSII qui me posaient tous la même question "et comment voyez-vous votre carrière dans 5 ans ?". Pour être accepté dans le job, je ne pouvais que répondre "dans 5 ans je me vois chef de projet", mon dieu, chef de projet, quelle horreur ! Je ne peux donc répondre à cette question car il y a 5 ans je n'avais pas encore créé le Paris JUG, je n'avais pas encore écrit mon livre Java EE 6, je n'avais pas encore créé Devoxx. Aujourd'hui, en plus de mon travail de développeur chez mes clients, ces activités font partie intégrante de ma vie et me prennent beaucoup de temps.

Si tu m'avais demandé il y a 5 ans "comment vois-tu ton job évoluer", jamais je n'aurais répondu "dans 5 ans je me vois monter la plus grosse conférence indépendante pour développeurs". C'est bon de se laisser des inconnues (vive les maths), c'est important de ne pas tout prévoir à l'avance, c'est sain de se laisser surprendre.

Des conseils aux étudiants et dévs qui nous lisent ?

Antonio : Oui, ne devenez pas chef de projets. Votre passion, vos études, vos valeurs, c'est le développement. La gestion d'un projet c'est une passion différente de la vôtre, des études différentes de celles que vous avez faites, des valeurs différentes des vôtres. Chacun son métier.

Faites le vôtre le mieux que vous pouvez, persévérez, et aidez-nous à changer la vision du développeur : un développeur peut avoir plus de 40 ans, il peut gagner plus que son chef de projet... et il peut même avoir les cheveux longs et ne pas porter de cravate, et ça, ça vaut de l'or.



Firefox OS : l'autre système pour smartphones

Mozilla était jusqu'à présent éloigné des problématiques de la mobilité, même si la fondation cherche à trouver sa place sur le marché des navigateurs mobiles largement dominé par les navigateurs basés sur WebKit. Au-delà, Mozilla veut être une alternative à Android, iOS, Windows Phone, BlackBerry. Mais proposer une nouvelle plateforme ne va pas de soi. Canonical cherche aussi une place avec Ubuntu pour les terminaux mobiles. Mozilla a dévoilé deux terminaux Firefox OS (Keon / Peak) mais sans date de sortie, sans tarification. L'équation est à la fois simple et compliquée : intéresser les constructeurs et opérateurs, motiver les développeurs à aller sur cette plateforme. Si l'alternative est bonne pour la concurrence et l'innovation, est-elle réellement crédible.

A vous de juger !

François Tonic



A l'occasion du « FirefoxOS applications Days » organisé par la fondation Mozilla les 25 et 26 janvier dernier à l'Epita Paris, et en simultané dans 25 autres villes du monde entier, nous avons pu connaître l'avancement du projet « Firefox OS » et impliquer les développeurs web dans ce nouveau système d'exploitation... Présentation.

> Les enjeux

Mozilla fête en 2013 ses 15 ans d'existence. Pour rappel, il s'agit d'une fondation à but non lucratif dont l'objectif premier est d'être communautaire et libre. Mozilla connaît très bien l'univers du web depuis les origines. Elle regarde la bataille actuelle autour de la mobilité entre iOS, Android, Windows Phone.

Mais le marché se doit être ouvert, et Mozilla le rappelle au travers d'un Hackaton. Cet événement était très attendu par l'ensemble des visiteurs pour connaître les défis qui seront lancés par la Team Mozilla et surtout les principales valeurs qu'il ne faut pas perdre, comme le rappelle Tristan Nitot (fondateur de Mozilla Europe) :

- **Learn** autrement dit, apprendre, se remettre en cause et surtout donner l'envie de découvrir et de faire des choses ensemble.
- Le **Hack** doit vous permettre de bidouiller, et principalement vous donner l'envie de créer quelque chose à travers le code pour vous pousser vers l'innovation. Bien sûr, ce petit truc en plus, va vous permettre de vous approprier de nouvelles technologies et de ne pas être seulement un simple utilisateur de la solution.

- La **fête**, car c'est très bien de se réunir pour faire avancer la technologie, mais il faut célébrer les innovations et les résultats obtenus.

> La vision Mozilla

Le fait de proposer un nouvel OS pour les terminaux mobiles doit répondre aux attentes de liberté que vous connaissez déjà au travers du web. Il y a 20 ans, les premiers réseaux apparaissaient (AOL / Compuserve / Minitel) qui étaient des réseaux innovants pour l'époque, propriétaires et fermés car ils ne pouvaient pas communiquer entre eux. Le phénomène a été renversé avec l'arrivée d'internet qui proposait la liberté d'échanges et de partage de la connaissance sur des principes simples, sur des standards ouverts, sans l'obligation d'avoir des kits de développements spécifiques et une documentation abondante. De nos jours, la comparaison est identique. La réalité montre que le secteur de la téléphonie et des tablettes propose la même approche que nous avons pu connaître, c'est-à-dire un modèle qui semble ouvert et libre alors que vous êtes dépendant d'une plateforme de distribution (market), avec l'obligation d'utiliser l'outil de développement associé (SDK) au système d'exploitation.

tion, d'apprendre un nouveau langage (pas toujours, NDLR), dont le propriétaire peut du jour au lendemain supprimer votre réalisation de son catalogue sans préavis (soumission des applications, NDLR). Le résultat peut alors être un problème de liberté, mais aussi économique par rapport au temps que vous avez passé. Ce qui pourrait être perçu comme une vision allant à l'encontre de la philosophie du web.

> FIREFOX OS, c'est quoi ?

Firefox OS est une plateforme complète pour les terminaux mobiles, s'appuyant sur les technologies standards du web : HTML 5 / CSS 3 / Javascript... Les applications s'exécuteront dans le navigateur. Ce système se structure autour d'une architecture système classique : services, noyau, applications. La plateforme de distribution s'appellera « Marketplace » (étonnant non ?) et offrira de nombreux avantages car vous pourrez utiliser la plateforme de Mozilla ou le placer ailleurs (par exemple chez vous). Ainsi, vous pourrez monétiser les applications, sur le principe de vente directe, sans besoin de passer par un intermédiaire (donc sans soumission, mais se posera alors de facto la notion de qualité et de contenu, NDLR). Mozilla s'entoure de partenaires pour l'aider dans ce projet. Car la fondation ne peut développer un tel système seule. Il faut impliquer des opérateurs, constructeurs, développeurs, etc.

Architecture

FirefoxOS s'appuie sur une architecture à 3 niveaux :

- « Gonk » est la distribution Linux servant de « core system ». A cela se rajoutent les services fondamentaux de téléphonie.
- « Gaia » est l'interface comprenant une application système et une suite d'applications comme la gestion des SMS, la possibilité d'utiliser le téléphone et un navigateur.
- « Gecko » est le moteur de rendu HTML. Il est utilisé par Firefox. C'est lui qui permet d'afficher et d'exécuter les applications.

Avec cette architecture, il n'est pas nécessaire d'avoir plus, car cela répond à ce qu'on attend d'un téléphone. Bien entendu, lorsque vous allumez votre téléphone, FirefoxOS utilisera un mode « lazy-loader » pour charger cette architecture, qui est un mode très rapide, par rapport à un mode classique de chargement de données sur disque. Ce mode permet en quelques secondes de pouvoir utiliser votre téléphone (comme les autres systèmes, NDLR).

Les Applications Web : le modèle de développement

Comment utiliser ? Comment déployer les applications ? Le développeur passe par la diffusion et le partage. Plusieurs modes de déploiement sont possibles :

- Un hébergement sur appCache : directement sur la plateforme Mozilla
- En s'appuyant sur le memcache
- A partir d'un package en zip externe

Par ailleurs, suivant le choix du type d'application que vous allez réaliser, vous utiliserez des niveaux d'accès différents. Trois modes d'applications sont proposées :

- Le mode « application certifiée »
- Le mode « privilège » est un certificat, qui accède à la mémoire du téléphone. Inclut l'accès aux services matériels.
- Le mode « Installé » est un mode ouvert, c'est-à-dire que vous pourrez installer sur votre terminal n'importe quelles applications.

Les permissions

Suivant le choix du type d'application, celle-ci disposera d'un certain niveau d'accès au système suivant les besoins du projet. Par exemple une application exploitant le mode GPS ne disposera pas des mêmes droits qu'un projet qui utilisera un accès aux photos. Par conséquent, le projet utilisera le niveau d'accès du système suivant la fonctionnalité et la finalité de l'application.

Il existe différents types de permissions et d'accès au contenu. Par exemple le système XHR, web application manager...



```

Fichier Edition Affichage Rechercher Terminal Onglets Aide
webdash x locales x bedrock x phpb x langchecker x Firefox OS x Terminal x
[origin: "https://marketplace.allizom.org/telefonica/", "installOrigin": "https://marketplace.allizom.org/telefonica/", "receipt": null, "installTime": 132333986000, "manifestURL": "https://marketplace.allizom.org/telefonica/manifest/42", "localId": 42, "marketplace": { "origin": "https://marketplace.firefox.com/telefonica/", "installOrigin": "https://marketplace.firefox.com/telefonica/", "receipt": null, "installTime": 132333986000, "manifestURL": "https://marketplace.firefox.com/telefonica/manifest/43", "localId": 43 } } Done (if [ -d ./git ]; then \ git log -1 --format="%%H%%at" HEAD > apps/settings/resources/gaia_commit.txt; else echo 'Unknown Git commit; build date shown here.' > apps/settings/resources/gaia_commit.txt; \ fi) Packaged webapps run-js-command webapp-zlp Done Generating extensions... Done Profile Ready: please run [b2g@firefox] -profile /home/pascalc/FirefoxOS/gaia[master] $ ~/FirefoxOS/build/dist/bin/b2g -profile [AccessFu] INFO attach forms.js loaded creating 11 [TabChild] SHOW (w,h)=(0,0) forms.js loaded loading app:/honescreen.gamobile.org/index.html#root, 1 creating 11

```

L'utilisation

Le multitâche a été incorporé dans la version de base car le comportement de navigation et d'utilisation d'une application est le même que sur un ordinateur (c'est le parti pris de Mozilla).

Au niveau de l'applicatif, vous naviguez d'une application à une autre sans difficulté, car celle que vous n'utilisez pas passera en fond de tâche (background), sous certaines conditions.

Ces conditions sont multiples, car lorsqu'une application passe en arrière-plan, celle-ci se met en sommeil. Si le système d'exploitation détecte qu'il lui manque de la place mémoire, il fermera les applications que vous n'utilisez pas pour libérer les ressources. Mais soyez vigilant sur le multitâche.

La publication

La publication d'une application sur FirefoxOS peut s'effectuer de 2 manières, par le marketplace ou l'externalisation.

Le Marketplace est un webmarket classique, géré par Mozilla. Il s'agit d'une plateforme de partage sur laquelle vous pouvez lister, classer et rechercher des applications, identique aux autres systèmes du marché.

Il se décompose en plusieurs zones :

- Le nom de l'application
- Une description
- Une capture d'écran
- La possibilité de télécharger l'application
- De tester directement en ligne avant de la télécharger

Chaque fiche applicative possède une zone ouverte pour noter, commenter.

L'externalisation est le 2e mode de publication. Vous ne passez pas directement par le marketplace de Mozilla, mais directement chez vous. Vous pouvez héberger les applica-



tions sur vos serveurs ou sur un service d'hébergement comme Git. Toutefois, vous avez toujours la possibilité de placer le 'manifest' ou le lien dans le marketplace pour être référencé dans le catalogue.

> Conclusion

La version 1.0 de FirefoxOS est disponible depuis quelques semaines, et une nouvelle version est en cours de développement. Ainsi, vous bénéficiez de la mise à jour automatiquement sans besoin de devoir changer de téléphone ou de faire une mise à jour manuelle. Concernant le matériel, vous pouvez déjà installer et utiliser Firefox OS sur les téléphones compatibles Android 4.x.

Ressources

Page officielle de Firefox OS :

<http://www.mozilla.org/fr/firefoxos/>

MarketPlace officiel de Mozilla :

<http://www.mozilla.org/fr/apps/>

Simulateur Firefox OS :

<http://people.mozilla.org/~myk/r2d2b2g/>

FirefoxAppDays : https://wiki.mozilla.org/Engagement/Developer_Engagement/FirefoxAppDays



Christophe Villeneuve
Consultant 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, PHP7V.

Débuter avec Firefox OS

Comme nous venons de le voir, **Firefox OS** est un nouveau système pour terminaux mobiles. Le succès de cette plateforme viendra de la confiance des constructeurs et des développeurs pour garnir l'App Store d'applications. Nous allons maintenant voir comment créer une application et la déployer.

> Rappel

L'architecture de Firefox OS repose donc sur 3 piliers :

- « Gonk » est la distribution Linux, la base du système
- « Gaia » est l'interface
- « Gecko » est le moteur de rendu HTML

Nous allons nous intéresser à Gecko. C'est par lui que l'on va pouvoir coder et optimiser son application HTML 5 / CSS 3.

Nous vous invitons à consulter la documentation officielle :

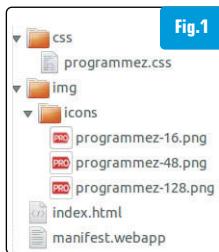
https://developer.mozilla.org/fr/docs/Mozilla/Firefox_OS

Toutes les fonctions HTML / CSS / JavaScript disponibles y sont listées et expliquées.

> Le développement

Structure d'une application

La structure d'une application Firefox OS se compose de différents fichiers qui peuvent être regroupés dans des sous-dossiers : [Fig.1]. Vous devez bien garder cette structure en tête et en comprendre le fonctionnement.



Manifest.webapp

Le fichier 'Manifest.webapp' contient les informations d'interaction entre le navigateur (et donc aussi avec le système) et l'application. Il est obligatoire. Ce manifeste est structuré au format JSON avec des informations obligatoires comme : un nom, une description, les icônes, le nom du ou des développeurs et les permissions.

En voici un exemple :

```
{
  "version": "1.0",
  "name": "Programmez",
  "description": "Programmez sur le mobile",
  "launch_path": "/index.html",
  "icons": {
    "16": "/img/icons/programmez-16.png",
    "48": "/img/icons/programmez-48.png",
    "128": "/img/icons/programmez-128.png"
  },
  "developer": {
    "name": "Christophe Villeneuve",
    "url": "http://www.programmez.com"
  },
  "installs_allowed_from": ["*"],
  "appcache_path": "/cache.manifest",
  "locales": {
    "fr": {
      "description": "Programmez sur Firefox OS",
      "developer": {
        "url": "http://www.programmez.com"
      }
    }
  }
},
```

```
"default_locale": "en"
```

```
}
```

Ce fichier est modulable. Il est possible d'ajouter des critères supplémentaires qui sont disponibles sur la page du manifeste de Mozilla, par exemple le FullScreen :

<https://marketplace.firefox.com/developers/docs/manifests>

Notre exemple montre 2 sous-fonctionnalités. Le support de plusieurs langues est possible, avec l'anglais par défaut et le français en langue alternative. Ainsi la description et l'affichage varient selon la langue du terminal. Il est possible d'ajouter d'autres langues.

L'autre particularité est de proposer plusieurs dimensions d'icônes de l'application, ce qui est utile car le téléphone prendra l'icône définie par défaut suivant le matériel et la dimension de l'écran, le système affichera l'icône prévue avec la taille associée [Fig.2]. Un grand classique sur les smartphones.

> L'application

Notre application affichera un texte centré horizontalement et verticalement. Comme notre exemple est réalisé en HTML5/CSS3, il est 100% compatible avec votre navigateur desktop. Pour effectuer ce développement, nous allons avoir besoin d'un fichier « index.html » et d'un fichier CSS. Vous pouvez voir le résultat [Fig.3].

Le fichier index.html

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8">
    <title>Programmez sur Firefox OS</title>
    <meta name="viewport" content="width=device-width">
    <link rel="stylesheet" href="css/programmez.css">
  </head>
  <body>
    <div id="container">
      <p>P R O G R A M M E Z</p>
      Le magazine du développement
    </div>
  </body>
</html>
```

Le source de ce fichier « index.html » ci-dessus, utilise la base « viewport », qui donne la surface de la fenêtre du navigateur disponible. Cette valeur est importante car la zone de contenu varie suivant les modèles de téléphones.

Le fichier programmez.css

```
p {
  background: #dddddd;
  font-weight:bold;
  color:#FF0000;
  height:200px;
```

```

line-height:200px;
text-align:center;
}

#container{
position:absolute;
text-align:center;
line-height: 200px;
position:relative;
border:2px solid #000000;
vertical-align:center;
}

```

Le fichier « programmez.css » contient la feuille de style. Nous appliquons une bordure avec un texte qui sera centré pour la balise « container ». Par ailleurs, le texte défini par la balise « <cp> » proposera un effet d'affichage différent.

> Validation

Lorsque vous avez réalisé votre application et effectué les premiers tests, passez l'application sur l'émulateur Firefox OS. Il est nécessaire que l'application passe ce test pour pouvoir déployer le projet.

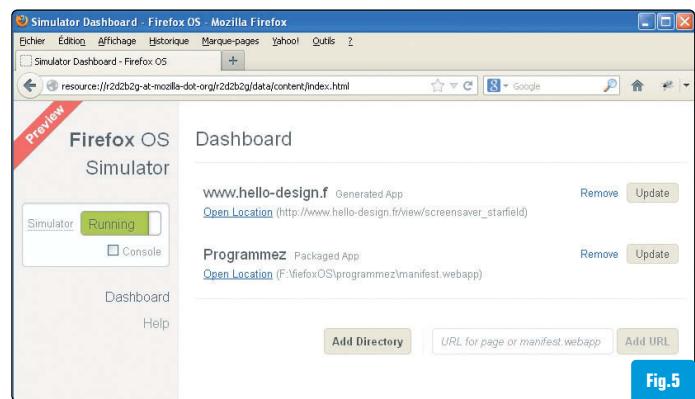
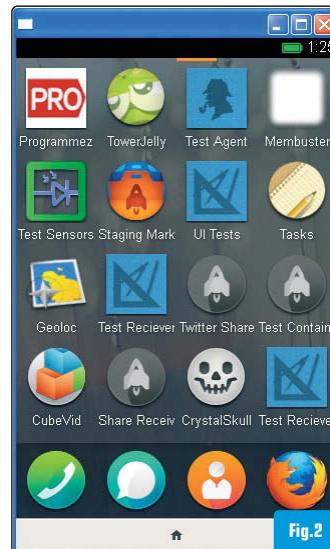
Les tests

L'environnement de test est une étape incontournable avant le déploiement de votre application sur mobile. Cette étape s'effectue par l'intermédiaire de votre navigateur desktop : vous devez installer l'extension du simulateur de Firefox OS, disponible à partir des extensions ou de l'adresse suivante [Fig.4] : <http://people.mozilla.org/~myk/r2d2b2g/>

Pour lancer l'émulateur, vous effectuez l'opération à travers le navigateur Firefox par le menu : outils \ developper web \ firefox OS Simulator. L'écran [Fig.5] propose une liste d'applications que vous devez télécharger sur votre disque dur ou à travers un lien URL. Ainsi, vous pouvez lancer votre simulateur de téléphone avec Firefox OS. Comme sur votre ordinateur, la fonction de glisser-déposer n'existe pas, vous devrez effectuer le geste avec votre souris.

Le mode debug

Lors d'un développement, les tests visuels ne suffisent pas pour résoudre les problèmes éventuels. Firefox OS Simulator propose « Firefox Remot Debugger » pour vous aider à corriger rapidement les bugs de vos applications. Vous pouvez aussi combiner l'émula-



teur « Firefox OS Simulator » et « Firefox Nightly ». Vous vous retrouvez avec une boîte à outils très puissante.

Si vous lancez le mode debug, vous pouvez effectuer du débogage en mode « pas à pas », ou encore mettre des points d'arrêts. Ces opérations sont très utiles pour connaître les valeurs des variables utilisées ou savoir si la fonction est bien prise en compte. Bon debug !

> Déploiement

Il existe plusieurs méthodes pour partager une application. Si vous choisissez une publication directement sur l'« App Store » de Mozilla, l'opération s'effectue en 4 étapes à partir de l'adresse suivante : <http://marketplace.firefox.com/developer>

Bien entendu, la validation du Manifest est de nouveau effectuée et le code HTML est automatiquement re-contrôlé et vérifié.

> Conclusion

Notre exemple est très simple mais montre les principales étapes. Firefox OS respecte les spécifications web du W3C, ce qui facilite le travail du développeur. N'oubliez pas que vous pouvez faire des applications pour Firefox OS, accessibles via le navigateur ou directement sur le terminal (applications web natives, mais pas natives au sens C++, NDLR).

Ressources

Doc Firefox OS : https://developer.mozilla.org/fr/docs/Mozilla/Firefox_OS

Market place : <https://marketplace.firefox.com/developers/>

DOC de débogage : <https://developer.mozilla.org/ms/docs/Tools/Debugger>

Hacking Firefox OS : <https://hacks.mozilla.org/2012/11/hacking-firefox-os/>

Christophe Villeneuve

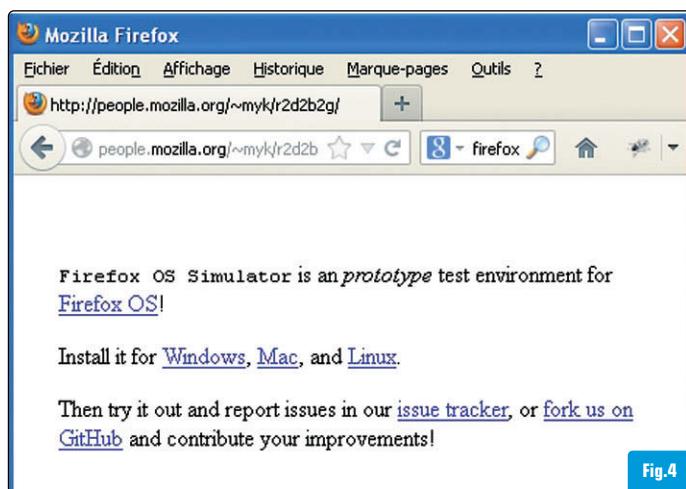


Fig.4

Nouveautés autour du développement ActionScript avec Flash Builder 4.7

En novembre 2012, Adobe dévoilait la version 4.7 de Flash Builder, accompagné d'outils tels Adobe Scout pour le profiling, et un changement de cap vers la création de jeux multi-plateformes (Gaming SDK). Petit tour de l'outil.

> Des nouveaux outils

L'apparition d'Adobe Scout (nom de code Monocle) pour analyser les performances et les goulets d'étranglement est une avancée majeure. Pour avoir le maximum d'informations, il est nécessaire d'activer la télémétrie détaillée au niveau du projet [Fig.1]. Dans le cas où vous n'avez pas accès au source d'une application ou si vous utilisez une version antérieure de Flash Builder ou Flash Professional, vous pouvez toujours utiliser Adobe Scout pour analyser vos fichiers SWF ou AIR, mais avec moins de compteurs de performance. Il existe néanmoins des scripts Python pour activer ces compteurs, avec l'aide du logiciel TelemetryEasy de Joseph Labrecque (<http://inflagrantedelicto.memoryspiral.com/2012/12/telemetryeasy-advanced-telemetry-utility-for-adobe-scout/>).

Voici un aperçu d'une session de débogage sur une tablette Android [Fig.2]. Dans le cas d'une session mobile/tablette, il est nécessaire d'installer préalablement Adobe Scout pour ces périphériques, disponibles sur leurs 'store' respectifs.

Pour le développement mobile via WIFI, je me suis préparé un fichier debugtablet.cmd (pour Windows uniquement, facilement portable sous iOS) :

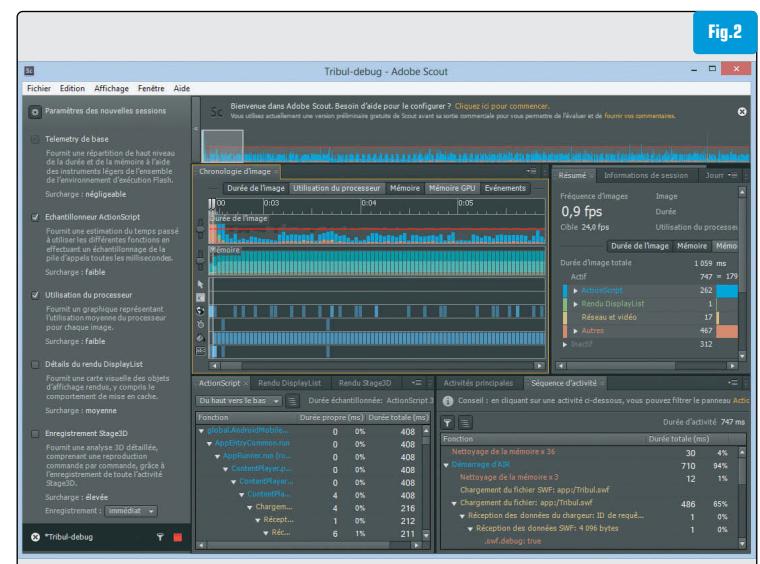
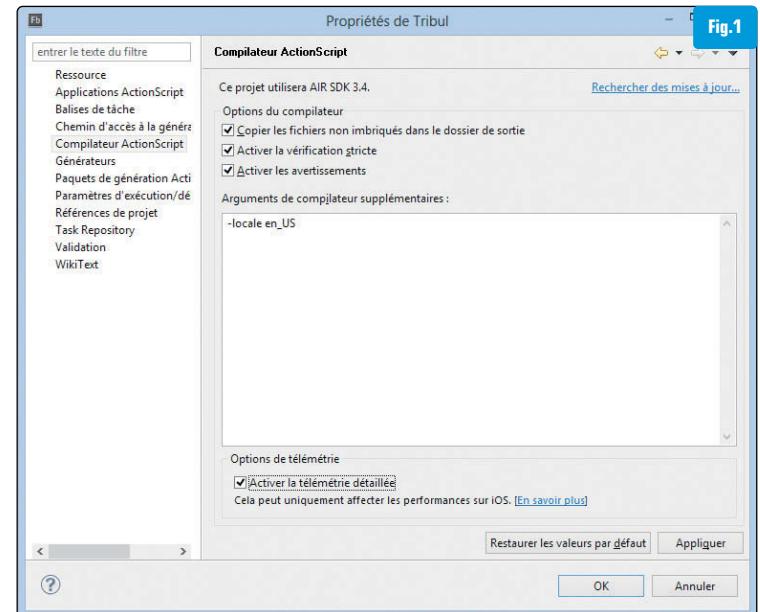
```
ipconfig | find "192.168"
c:
cd "C:\Program Files\Adobe\Adobe Flash Builder 4.7 (64 Bit)\sdks\4.6.0\lib\android\bin"
adb devices
adb tcpip 5555
adb connect 192.168.xx.yy
adb devices
pause
```

Cela me permet de lancer l'application en mode sans fil.

> Nouveautés de Flash Builder 4.7

Adobe Flash Builder 4.7 est désormais disponible en 64 bits et intègre le nouveau compilateur ASC 2.0, beaucoup plus rapide.

A noter, la séparation dans le SDK de Flex et de AIR : il est désormais possible de changer le SDK AIR indépendamment du SDK Flex, afin par exemple d'utiliser le SDK open source Apache Flex, donné à la fondation Apache (<http://incubator.apache.org/flex/>). La première sortie du SDK Apache Flex 4.8 est iso fonctionnelle avec la version Adobe Flex 4.6, mais au moment où vous lirez ces lignes, la version 4.9



d'Apache Flex sera sortie depuis quelques semaines, avec son lot de nouveautés. Un installateur pour Apache Flex est disponible à la même adresse. Pour le développement d'AIR Native Extensions, j'ai remarqué que Flash Builder 4.6 m'enlevait à chaque lancement d'une application la ligne suivante dans application.xml :

```
<extensions>
<extensionID>com.magicalhobo.utils.PenTablet</extensionID>
</extensions>
```

Pour éviter ce problème, je me suis créé un fichier de commandes Windows :

```
SET AIR_SDK=%USERPROFILE%\Documents\flex_sdk_4.6.0.23201B
SET ANE_PATH=..\..\PenTabletLib\bin
"%AIR_SDK%\bin\adtpause.bat" -package -XnoAneValidate -storetype
pkcs12 -keystore test.p12 -storepass test -target native OnyxPen
Tablet OnyxPenTablet-app.xml OnyxPenTablet.swf -extdir "%ANE_PATH%"
```

acenda

► MARS

- Le 26 mars 2013, Paris, Quai de Grenelle,
Journée Française des Tests Logiciels 2013

Un état de lieu du marché des tests logiciels en France sera présenté, ainsi que les résultats du premier « Observatoire des Pratiques du Test Logiciel », lancé en décembre dernier par le CFTI. <http://www.cfti.fr/>

• Du 26 mars au 28 mars 2013, Porte de Versailles - Pavillon 5 - Paris. Les **Salons INTRANET & Collaboratif et RSE 2013** tiendront leur 8e édition. Vous y découvrirez les outils, conseils, services, applications intranet, solutions d'entreprises collaboratives, réseaux sociaux d'entreprise.
<http://www.salon-intranet.com/>

AVRIL

- Du 02 avril au 06 avril 2013, Espace Cap 15, au pied de la Tour Eiffel, **Symfony Live Paris 2013**. Deux jours de formations, deux jours de conférences et une session de certification Symfony. Un Symfony Live Paris 100 % francophone !
<http://paris2013.live.symfony.com/>

Du 03 avril au 04 avril 2013, CNIT Paris la Défense, **Big Data Paris 2013**. 1500 participants, 60 exposants et 100 speakers sont attendus pour deux journées de conférences, d'exposition et retours d'expérience BigData. <http://www.biddataparis.com>

- 11 avril : Scrum Day

Organisée par le French Scrum User Group, cette journée sera entièrement consacrée à Scrum. Le but est d'expliquer l'agilité, de présenter des cas concrets.

Où : IBM France à Bois Coloré
Site : <http://www.scrumday.fr>

> Développement de jeux en ActionScript

Flash Builder s'oriente désormais vers le développement de jeux. Adobe a mis à disposition Adobe Game Developer Tools contenant :

- Adobe Scout
 - Adobe Gaming SDK : frameworks Starling, Feathers, Away3D et des AIR Native Extensions(ANE)
 - Flash C++ Compiler : compilateur C++ pour réutiliser le code C/C++ existant

Starling est un framework simplifiant l'accès aux primitives Stage3D, lequel permet l'utilisation du GPU pour une performance accrue. Associé à Feathers(<http://feathersui.com/>), de la société Gamua, l'interface utilisateur est stockée sur le GPU avec des thèmes personnalisables : boutons, barres de défilement, texte, listes,...

Starling ressemble fortement aux classes actionscript standard, nommées de la même façon. Par exemple, `starling.display.Sprite` remplace `flash.display.Sprite`, il faut donc porter une attention particulière aux 'imports' en tête de vos classes, avec l'autocomplétion, il faut choisir la bonne bibliothèque [Fig.3]. Ceci est déroutant au début.

Bruce Lane

Développeur du logiciel Onyx-VJ et musicien (<http://www.batchass.fr>)_Administrateur système Sophia-Antipolis (<http://www.casa-infos.fr>)

L'INFO permanente

- **L'actu** : le fil d'info quotidien de la rédaction
 - **La newsletter hebdo** : abonnez-vous, comme 38 000 professionnels. C'est **gratuit** !

C'est PRATIQUE !

- Le forum : modéré par la rédaction et les auteurs de Programmez!, rejoignez les forums techniques de programmez.com

- ## ● Les tutoriels : une solution en quelques clics

- ## Le téléchargement récupérez les nouveautés.



www.programmez.com

Les nouveautés sociales de SharePoint 2013

Les Réseaux Sociaux d'Entreprise (ou RSE) commencent à apparaître dans le paysage des Systèmes d'Information français. Selon l'Observatoire de l'intranet, 14% des entreprises sondées utilisent régulièrement un Réseau Social d'Entreprise, le reste des entreprises étant plus attentif aux résultats obtenus par ces pionniers. Cependant, ils sont très favorables à en mettre un en place dans les mois qui viennent.

Dans un premier temps, commençons par expliquer ce qu'est un Réseau Social d'Entreprise, ses objectifs, ses usages et les impacts occasionnés par sa mise en place. Un Réseau Social d'Entreprise est un espace centralisé qui permet de favoriser la mise en relation et le partage des connaissances entre les individus à travers un outil simple de communication, d'échange et de collaboration. Cet espace se caractérise par une mise en avant des expertises et connaissances des membres du réseau à travers leurs activités, pour permettre à l'entreprise d'être plus productive et efficace.

Maintenant, interrogeons-nous sur l'intérêt d'un réseau social interne en entreprise. La principale raison au jour d'aujourd'hui se situe dans la différence entre l'expérience rencontrée dans la vie personnelle et les réseaux sociaux dits publics tels que Facebook, Twitter ou

Viaideo/Linkedin et celle offerte en entreprise avec des portails vieillissants de communication et de collaboration ne répondant plus aux besoins utilisateurs, et plus particulièrement à la Génération Y ou Digital Natives.

Dans ces conditions, pour redonner l'image d'une entreprise innovante, le RSE importe les pratiques du grand public pour améliorer les usages déjà mis en place dans le monde de l'entreprise. L'utilisateur retrouve donc les habitudes de consultation d'internet, tout en étant orienté business, et non seulement communication et partage.

On pourrait citer par exemple, que la mise en place d'un Réseau Social d'Entreprise a permis de réduire efficacement le nombre de mails envoyés et reçus ou encore que la culture collaborative (co-création, gestion électronique de documents, communautés de travail, gestion d'appel à idées d'innovation,...) commence à être ancrée dans les

habitudes de travail de chacun. Par ailleurs, la gestion des connaissances en interne a grandement été améliorée avec notamment une mise en avant des meilleures réponses et des experts présents sur le réseau. Microsoft ne proposait pas nativement de plateforme Réseau Social d'Entreprise à son offre SharePoint 2010, même si des fonctionnalités sociales existaient avec la brique Mysite, comme une fiche profil et un mur d'activités non interactif. Conscient de ce point faible, Microsoft a réagi en proposant, avec sa nouvelle version SharePoint 2013, une nouvelle expérience à ses utilisateurs à travers une refonte complète de son offre sociale avec, par exemple, la mise en place d'un mur d'activités et d'un outil de micro-blogging ou la création à la demande de communautés. Par ailleurs, la

rial de Microsoft concernant la suite Office et son prochain système d'exploitation Windows 8. Autre bouleversement important, l'installation en natif d'une seule et même barre de navigation globale à tous les sites qui permet à l'utilisateur de naviguer entre ses différents sites d'équipes, ses communautés, ses documents et son profil. En effet, sous SharePoint 2010, lorsqu'on est sur un site d'équipe, le seul moyen d'accéder à son profil, ses sites, ses documents et à son mur d'activités était de cliquer sur son nom en haut à droite. Désormais, ces quatre fonctionnalités sont toujours présentes sur la nouvelle barre de navigation.

Même si sous SharePoint 2010, il était possible avec quelques modifications du code HTML et ses feuilles de styles d'avoir la même barre de navigation [Fig.1]. Enfin, comme sur SharePoint 2010, une barre de recherche sera présente sur la quasi-totalité des pages.



stratégie Réseau Social d'Entreprise de Microsoft ne se limite pas qu'à SharePoint mais continue avec l'acquisition de Yammer en Juin 2012 pour un montant de 1,2 Milliard, ce qui lui permettra de proposer une offre Réseau Social d'Entreprise dans le Cloud en mode freemium. Microsoft a désormais toutes les armes en main pour devenir l'un des leaders sur le marché. Nous allons maintenant détailler toutes les nouveautés apportées par SharePoint 2013 concernant la brique sociale.

> Navigation & Ergonomie

Premier changement visible, la nouvelle interface Windows 8 qui suit la ligne édito-

> NewsFeed

Aussi appelé Mur d'activités, cette fonctionnalité permet à l'utilisateur de visualiser les activités de ses collègues

(ajoutés ou synchronisés depuis l'annuaire d'entreprise) ou des site d'équipes/communautés auxquels il appartient et d'interagir comme par exemple en laissant des commentaires. Élément essentiel d'un Réseau Social d'Entreprise, le mur d'activités offert nativement par SharePoint 2010 ne répondait pas aux critères en termes de fonctionnalités et d'expérience utilisateur : pas de commentaires possibles, pas d'interaction ni de conversations avec un contact, pas de remontées d'activités des sites, un mur de discussion (note board) mal placé, etc... L'ajout d'une surcouche sociale telle que NewsGator ou Calinda Software, pour ne citer qu'eux, était donc obligatoire si l'on souhaitait avoir une expérience utilisateur proche de celle rencontrée sur les outils

grand public. Lucide sur ce problème, SharePoint 2013 se devait d'être au niveau de ce que proposent les réseaux sociaux grand public mais également d'entreprises comme BlueKiwi ou Jive si l'on parle de solutions concurrentes à SharePoint, ou bien de NewsGator ou Calinda Software pour les extensions sociales de SharePoint, pour ne citer qu'eux. Pari réussi car comme le montre cette capture d'écran [Fig.2], ce mur d'activités permet de retrouver les activités et conversations que l'utilisateur a souhaité suivre. En suivant des collègues, des tags, des sites et des documents, l'utilisateur verra remonter sur son mur tous les éléments liés à ses suivis. En effet, l'utilisateur verra toutes les activités de ses collègues, leurs mises à jours de profils, toutes les conversations avec ses tags, les conversations qui lui sont directement adressées (avec le symbole @username), les activités à l'intérieur de ses communautés et sites d'équipe, ainsi que les modifications des documents suivis. L'utilisateur aura donc une vision globale des échanges qui se déroulent sur le Réseau Social d'Entreprise et pourra également y participer, même s'il n'était pas convié au départ dans la conversation. Les échanges au sein de l'entreprise seront donc publics, acces-

sibles à tous les collaborateurs et non plus stockés dans des boîtes mails entre quelques destinataires privilégiés. De plus, grâce à la recherche sur les conversations, n'importe quel utilisateur pourra retrouver toute discussion qui lui serait utile dans son projet. Ainsi ce mur d'activités permet d'améliorer la communication interne entre collaborateurs mais également de chambouler la communication à sens unique en passant par une réduction du temps de recherche de l'information avec un moteur de recherche global qui ira indexer toutes les conversations, les questions et leurs réponses associées, tout en réduisant les mails et conversations privées.

Autre nouveauté sur ce mur, l'apparition d'un outil de type micro-blogging qui permet de s'exprimer très rapidement et de partager des documents, des images, des liens ou des vidéos. Ainsi, comme pour tout outil de réseautage, les fonctionnalités classiques sont présentes :

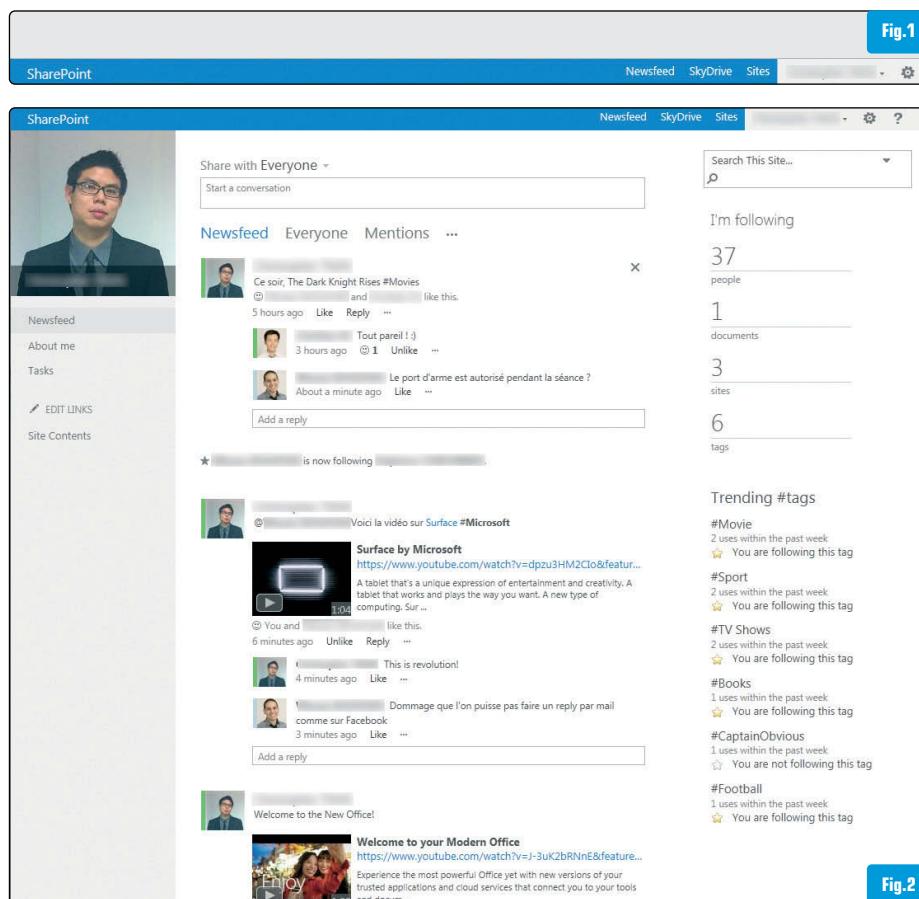
- Poster un statut ou un message pour ses collègues.
- Poster un message directement à ses communautés membres (ou les sites d'équipe comme on le verra plus loin) depuis son mur
- Partager des images et en afficher une miniature

- Partager des documents et les visualiser directement depuis le navigateur, sans avoir besoin du client lourd Office installé sur son poste de travail, grâce à Office WebApps
- Partager des liens avec un affichage plus convivial en cachant l'adresse url par un texte. Pour les liens, il est à noter que lorsqu'on insère un lien d'une vidéo provenant de Youtube, la description et l'image de prévisualisation de cette vidéo apparaissent. Les utilisateurs pourront donc regarder la vidéo directement depuis le mur
- Aimer un post, le commenter, le suivre ou encore le verrouiller si besoin
- Utiliser des tags avec le hashtag représenté par le symbole # pour donner un thème à son post, ce qui donne la possibilité de donner un sens au message et de pouvoir visualiser toutes les conversations liées à ce tag
- De la même manière que Twitter et ses Trending Topics, ces hashtags permettront aussi de connaître les sujets populaires sur le réseau avec l'apparition d'un Trending Tags. Ce dernier permettra d'afficher les trois tags les plus utilisés sur le Réseau Social d'Entreprise à l'instant T (et par extension les plus populaires) et proposera à l'utilisateur de suivre ces tags si ce n'est pas déjà fait
- Voir les messages qui lui sont directement mentionnés.

Autant de fonctionnalités permettant de rapprocher les collaborateurs et de favoriser les échanges pour améliorer la communication et la collaboration au sein de l'entreprise. Une des particularités propre à SharePoint 2013 est la possibilité pour l'utilisateur de visionner l'ensemble des conversations et activités de tous les membres du réseau grâce au filtre Everyone (cf la capture précédente) du NewsFeed, même si ces derniers ne font pas partie de ses collègues. En effet, cette fonctionnalité permet donc de lever les limites de la sphère personnelle (mes collègues et moi), de décloisonner réellement l'entreprise en permettant à toutes les personnes dans l'entreprise, qui n'ont aucun lien, mais qui partagent des compétences ou centres d'intérêts communs, de se connecter pour en discuter.

> SkyDrive / SkyDrive Pro

A première vue, on pourrait penser qu'il s'agit d'une nouvelle fonctionnalité de SharePoint 2013. Cependant, ce n'est qu'un changement de nom de l'onglet MyContent, même s'il apporte quelques changements et



The screenshot shows the SharePoint 2013 Newsfeed interface. At the top, there's a navigation bar with 'SharePoint' on the left and 'Newsfeed' as the active tab. Below the navigation bar, the main content area is divided into several sections:

- Share with Everyone**: A search bar with the placeholder 'Start a conversation'.
- Newsfeed**: A list of posts from various users. One post from 'Ce soir, The Dark Knight Rises #Movies' is highlighted, showing interactions like 'Like' and 'Reply'. Another post from 'Surface by Microsoft' includes a video thumbnail and a link to the video on YouTube.
- Everyone**: A section for posts from everyone, including a post from 'Welcome to the New Office!'
- Mentions**: A section for posts mentioning the user, with a post from 'Welcome to your Modern Office'.
- Following**: A summary of the user's following activity, showing 37 people followed, 1 document, 3 sites, and 6 tags.
- Trending #tags**: A list of trending hashtags with counts and status (e.g., 'Movie' is used twice in the past week, 'Football' is used once in the past week).

Fig.2

nouveautés très intéressantes. Pour rappel, MyContent liste l'ensemble des documents auxquels l'utilisateur a accès ainsi que ses documents personnels qu'il peut partager avec ses collègues s'il le désire [Fig.3].

Cet onglet, divisé en deux catégories, permet de lister dans un premier temps l'ensemble des documents que l'utilisateur a en sa possession et les documents auxquels on lui a donné un accès en lecture ou en écriture et dans un deuxième temps, les documents suivis par l'utilisateur. Pour chaque document, l'utilisateur a la possibilité de l'éditer, de le partager avec ses collègues, de le suivre ou bien de le lire en prévisualisation, si Office Web Apps est installé comme le montre la capture d'écran. Notons qu'Office Web Apps, permet également de créer de nouveaux documents ou d'édition les documents Office (Word, Excel, Powerpoint, OneNote) directement, sans avoir installé le client lourd sur sa machine. Une des particularités de SkyDrive est la possibilité offerte à l'utilisateur de synchroniser les documents présents dans cette bibliothèque sur son poste de travail à travers un explorateur Windows, et inversement si la suite Office 2013 est installée sur le poste utilisateur, avec notamment SkyDrive Pro. En effet, ce dernier installera un dossier spécial dans l'explorateur Windows sur le poste de travail de l'utilisateur, ce qui lui permettra de posséder en quelque sorte un disque dur virtuel, accessible à n'importe quel moment, qu'il ait internet ou non, pour travailler en ligne ou hors ligne. Ce fonctionnement est le même que l'on peut retrouver sous SharePoint 2010 et SharePoint Workspace mais ce dernier ne fait plus partie de la suite Office. On peut également noter qu'une des nouveautés que l'on retrouvera sur quasiment toutes les listes sous SharePoint 2013 est également présente : l'apparition d'un moteur de recherche localisé et dans notre cas, uniquement indexé sur cette bibliothèque de documents. Enfin, l'utilisateur a également la possibilité de visionner l'ensemble des documents qu'il suit pour voir l'activité de ces documents remontée sur le mur d'activités ainsi qu'une suggestion de documents qui seraient intéressants à suivre, en fonction de ses collègues, de ses expertises, de ses centres d'intérêts ou de ses activités sur le Réseau Social d'Entreprise.

> Sites

De la même manière que l'onglet SkyDrive permettait de recenser l'ensemble des documents suivis, l'onglet Sites offre également

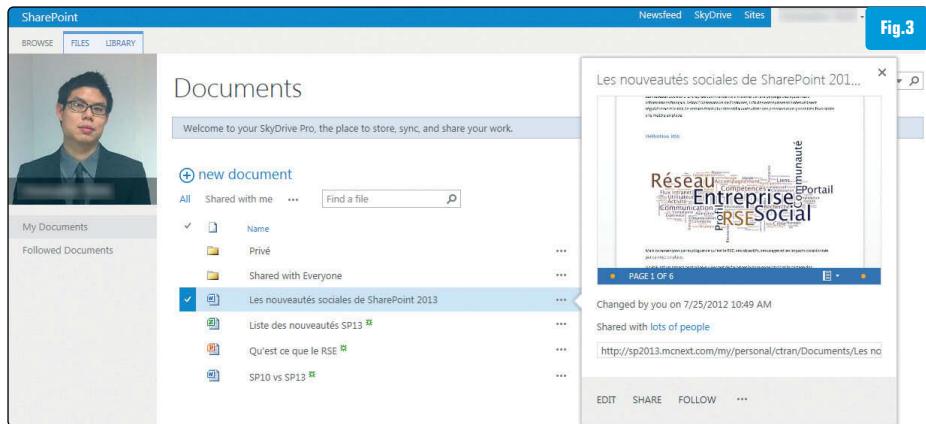


Fig.3

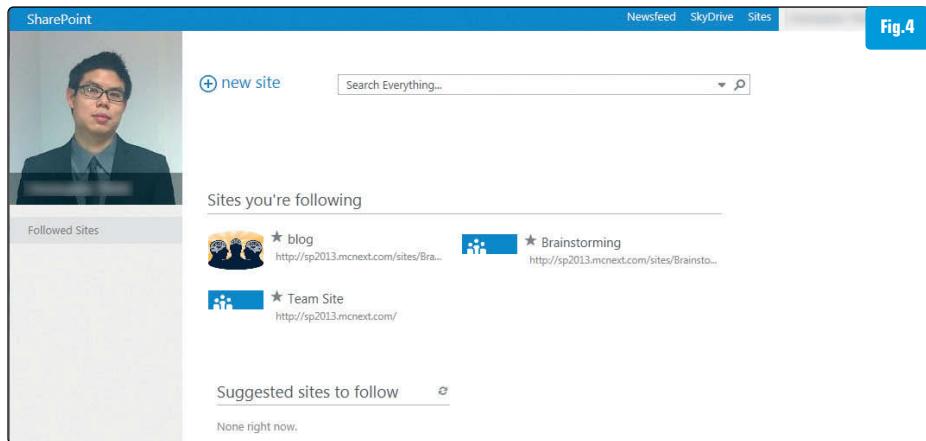


Fig.4

la même possibilité de lister l'ensemble des sites que l'utilisateur suit afin que les activités de ces derniers remontent sur son mur d'activités [Fig.4].

On retrouve également un module de suggestion de sites à suivre, qui se base, selon ses activités, sur le Réseau Social d'Entreprise, de ses collègues, de ses expertises ou de ses centres d'intérêts.

Enfin, une des nouveautés attendues par les utilisateurs concerne la possibilité pour chacun de se créer des sites d'équipe, très rapidement, grâce au bouton +New Site. Le processus de création de sites d'équipe devient donc plus facile et plus simplifié aux utilisateurs, tout en allégeant la charge de travail des Community Managers ou des Administrateurs qui n'auront plus à se préoccuper des processus de création. Néanmoins, il est à noter que cette fonctionnalité ne serait possible que si l'option d'autocréation de sites est activée dans l'Administration Centrale de la ferme SharePoint.

Nous allons maintenant détailler les deux principaux types de sites que l'on rencontrera très souvent sur le Réseau Social d'Entreprise : les communautés et les sites d'équipe. Le site de type communauté s'adresse plutôt aux utilisateurs qui désirent

avant tout échanger et partager des idées, des informations (axé donc sur la conversation et le forum de discussion) alors que le site d'équipe est avant tout un site pour collaborer et travailler (axé sur les bibliothèques de documents et les listes de tâches).

> Site de type Communauté

SharePoint 2013 introduit désormais la notion de communauté qui manquait sous SharePoint 2010, même s'il est vrai que sous cette version, un site d'équipe pouvait également être utilisé comme une communauté. Une communauté a pour but premier de rassembler des collaborateurs autour d'un thème donné, qu'il soit professionnel ou personnel, pour échanger et converser en toute tranquillité [Fig.5]. En effet, la communauté repose essentiellement sur la brique Forum de discussion qui permet à chaque membre de discuter, de poser des questions et d'en attendre une réponse de la part des membres. Nous allons maintenant détailler d'autres fonctionnalités propres à une communauté :

- Tout d'abord, il est à noter que toutes les communautés sont privées par défaut. Si le Community Manager (la personne en charge de modérer et d'animer la communauté) souhaite l'ouvrir au public, il doit ajouter le

- groupe « Everyone » comme membre
- Ce dernier a également la possibilité d'activer l'auto-approbation lorsqu'un visiteur souhaite devenir membre de la communauté. Ce processus permet au Community Manager de filtrer les arrivées des membres et de n'accepter que sur profil
- Enfin, dans le cas où la communauté est ouverte et sans restriction de lecture ou d'écriture (comprendre que le visiteur a les droits de contribution), ce dernier deviendra automatiquement membre de la communauté dès la publication de son premier post
- Qui dit communauté dit membres. Une liste des membres de la communauté est donc affichée pour permettre de voir rapidement tous les collaborateurs qui l'ont rejointe. Pour chaque membre, un minitableau d'activités (nombre de questions posées et de réponses données) est affiché ainsi que leur date d'adhésion, sans oublier leur badge obtenu ou leur nombre de points. Qui plus est, le membre peut quitter à tout moment sa communauté s'il le désire sans aucune action de la part du Community Manager
- Les sujets de discussion peuvent être classés selon les catégories voulues, sorte de thème à créer pour permettre une meilleure visibilité des thèmes abordés dans la communauté
- Le Community Manager peut épingle un sujet de discussion s'il considère que ce dernier est utile ou important à la communauté pour que les membres puissent le voir très rapidement, sans épucher tous les autres sujets
- De la même manière que Yahoo Answers, le CM (pour Community Manager) ou l'auteur de la question peut également désigner la meilleure réponse possible pour l'afficher directement sous la question, et permettre ainsi aux membres de la voir directement, sans parcourir toutes les

réponses du fil de discussion. Autre indicateur possible, si la communauté estime qu'une réponse est la meilleure, elle peut aussi l'aimer. Ainsi chacun pourra voir la réponse avec le plus de « like » et donc comprendre que la communauté approuve cette réponse, sans pour autant qu'elle soit validée par le CM.

- L'introduction de la ludification (traduction française de gamification) au sein du Réseau Social d'Entreprise en permettant à chaque membre de gagner des points au fur à mesure de sa contribution (que ce soit en posant une question, en donnant des réponses ou lorsqu'une de ses réponses est désignée comme étant « la meilleure ») entraîne plus de contribution et d'esprit compétitif de la part des employés. Plus le membre gagne des points, plus il est reconnu par la communauté comme étant une personne identifiée comme expert dans un domaine particulier.
- De la même manière, il est possible au CM d'attribuer des badges à certains membres pour les mettre en avant, comme par exemple, un badge Expert attribué à un membre qui est considéré comme expert dans un domaine particulier, un badge Modo pour les modérateurs, etc.

Le forum de discussion sous SharePoint 2013 retrouve donc à peu près les grandes fonctionnalités que l'on retrouve sur un forum grand public, même s'il en manque, comme par exemple la possibilité de verrouiller un sujet, qui existe pourtant sur le micro-blogging ou d'insérer une signature. Par ailleurs, il existe désormais un site servant d'annuaire de communautés qui permet de centraliser toutes les communautés existantes et aux membres du Réseau Social d'Entreprise de découvrir les communautés les plus populaires ainsi que de rechercher celles qu'ils désirent rejoindre ou bien de

vérifier qu'aucune autre communauté n'existe sur le même thème avant de faire une demande de création.

Il est à noter qu'il est possible d'ajouter l'application NewsFeed pour le micro-blogging dans une communauté pour favoriser la communication informelle ou non entre les membres et qui remonteront sur le mur de l'utilisateur. En effet, les conversations du forum ne remontent pas sur le mur d'activités, on peut juste savoir qu'une personne a posté une question sur le forum mais on ne peut pas connaître le contenu du sujet, ni les réponses qui suivent. Ainsi le forum servira uniquement de questions/réponses alors que le micro-blogging servira à toute autre communication et ne polluera pas le forum de discussions inutiles.

> Team Site

Modèle de site particulièrement utilisé depuis SharePoint 2003 pour la collaboration, le Team Site ou site d'équipe version 2013 s'enrichit du module de micro-blogging NewsFeed par défaut, en plus des listes classiques comme le calendrier, les tâches, les bibliothèques de documents ou les annonces, ce qui permet d'améliorer encore plus la collaboration existante et de favoriser la communication [Fig.6].

Toutes les conversations de ce mur local remonteront sur le mur personnel de l'utilisateur, ce qui lui permettra d'être toujours au courant des conversations du site d'équipe sans pour autant y accéder.

Autre nouveauté qui sera très utile pour la conduite de changement, SharePoint 2013 intègre désormais un guide de démarrage permettant d'accompagner interactivement le nouvel utilisateur pour qu'il puisse maîtriser toutes les fonctionnalités du site.

Concernant les autres listes (ou nouvellement appelé Apps), on constate que peu de nouveautés sociales ont été intégrées sauf bien sûr les documents que l'on peut suivre ou aimer. On peut cependant noter une amélioration des remontées des tâches affectées à l'utilisateur dans un site d'équipe.

En effet, toutes ses tâches apparaîtront sur son profil et plus particulièrement dans My Tasks qui regroupera toutes ses tâches affectées de tous les types de sites membres. Enfin, de la même manière qu'une communauté peut intégrer des Apps (les listes) pour collaborer, le site d'équipe peut également activer le module communautaire afin d'intégrer les fonctionnalités propres aux communautés, à savoir le

Fig.5

forum de discussion, la modération, la liste des membres et le système de valorisation des contributeurs [Fig.7]. Fonctionnellement, le profil utilisateur a peu évolué. L'ergonomie et la navigation ont cependant été améliorées pour permettre de le rendre plus clair et convivial par rapport au profil sous SharePoint2010. On remarque donc une meilleure présentation du profil, plus espacé et plus lisible, divisé en trois colonnes :

- A gauche, une photo de l'utilisateur plus grande et un menu de navigation vertical : la photo restera toujours présente du moment qu'on reste sur le profil.
- Une partie centrale qui affiche la description de l'utilisateur et ses expertises (sur lesquelles chaque visiteur peut cliquer pour poser une question concernant l'expertise concernée), puis ses activités sur le Réseau Social d'Entreprise
- A droite, des informations supplémentaires personnelles ou professionnelles, ainsi qu'un organigramme si on visite le profil d'un collègue, semblable à celui de SharePoint 2010.

Le profil utilisateur est donc mieux organisé et plus lisible par rapport à sa version 2010, plus proche des fiches profil des réseaux sociaux grand public. Comme expliqué dans la partie Site d'équipe, l'utilisateur peut désormais voir toutes les tâches qui lui sont assignées, et ce, depuis tous les sites d'équipe et communautés desquels il est membre. Outre la liste des tâches, classées par sites d'équipes et communautés, une frise de temps est également disponible pour permettre une visualisation « temporelle » et l'aider à définir ses priorités.

> Compatibilité Mobile

La possibilité de consulter son intranet est devenue au cours de ces derniers mois une fonctionnalité essentielle de tout produit ou logiciel informatique tant la mobilité est devenue commune et vitale dans le monde de l'entreprise. Cependant, SharePoint 2013 ne répond qu'en partie à ce besoin en offrant une version mobile du Réseau Social d'Entreprise toujours aussi archaïque, qui consiste à afficher des listes sans mise en page. De plus, la version « mobile » ne permet pas d'afficher des sites de publication, uniquement des sites d'équipe, ce qui handicape très fortement les sites internet basés sous SharePoint.

Il est cependant possible d'activer la vue PC pour profiter pleinement de toutes les fonctionnalités et mises en page du Réseau

Site Feed

Start a conversation

Brainstorming Newsfeed

Pour faire des screens, pourriez-vous participer à cette cté svp (sujets intéressants of course) ? :p
Communauté Sociale
Merci :)

16 minutes ago Like Reply ...

Enfin, juste recopier vos messages :p
10 minutes ago Like ...

Add a reply

Quelqu'un a déjà testé la fonctionnalité Site Mailbox? ça requiert une fonctionnalité de ferme et je ne peux donc pas tester
Yesterday at 6:19 PM Like Reply ...

▼ Show all 4 replies

Si c'est le cas je ne l'avais jamais vu et j'irai mettre fin à mes jours
dignement pour sauver mon honneur :)
About an hour ago Like ...

Non ça n'existe pas .. Le Team Mailbox te crée une liste de distrib Exchange
pour les membres d'un site d'équipe :)-
25 minutes ago Like ...

Add a reply

Hello World :)
You and ' like this.
About an hour ago Unlike Reply ...

▼ Show all 3 replies

veut coder son premier Hello World :) #Développeur
26 minutes ago Like ...

Déjà fait :p
25 minutes ago Like ...

Add a reply

SharePoint

About Christopher Tran

Newsfeed

About me

Tasks

EDIT LINKS

Site Contents

edit your profile

Je suis consultant SharePoint. J'ai travaillé sur SharePoint depuis sa version 2003. Je suis passionné d'internet et des réseaux sociaux, notamment des réseaux sociaux d'entreprise. Je travaille sur SharePoint 2013 depuis plusieurs mois.

Feel free to mention me in a post or ask me about any of the following topics:

- Calinda
- SharePoint 2013
- Yammer
- SharePoint 2010
- NewsGator
- Social Networking

l's Activities

Voici la vidéo sur Surface #Microsoft

Surface by Microsoft
https://www.youtube.com/watch?v=dpu3HM2Clo&feature...

A tablet that's a unique expression of entertainment and creativity. A tablet that works and plays the way you want. A new type of computing. Sur ...

You and ' like this.
3 minutes ago Unlike Reply ...

This is revolution!
About a minute ago Like ...

Add a reply

Consultant SharePoint
Pôle SharePoint
EMAIL @mcnext.com
PAST
PROJECTS
SKILLS Writing
SCHOOLS UMLV
BIRTHDAY November 24
INTERESTS Football, Sport, Movie, TV Shows, Books, test

SHOW LESS

Tags and Notes
SEE ALL

Social d'Entreprise. Par ailleurs, un système de push a été instauré afin d'envoyer des notifications d'événements (comme par exemple l'ajout d'un élément dans une liste ou sa modification par un utilisateur) sous forme de tuiles si on prend le cas d'un Windows Phone.

> Conclusion

En conclusion, SharePoint 2013 reprend les fonctionnalités proposées sous SharePoint 2010 et les enrichit pour enfin proposer un Réseau Social d'Entreprise, avec une expérience utilisateur proche de ce qu'on trouve dans le grand public. Outre la possibilité de

visualiser les activités de ses collègues et d'y interagir, cette nouvelle version améliore la collaboration en entreprise grâce à la suggestion de documents utiles, à la création simplifiée de sites d'équipe, ou bien sûr, la gestion de la connaissance interne avec la sélection des meilleures réponses.

On peut donc clairement dire que, SharePoint 2013 a enfin abordé le virage du social et l'a clairement réussi, même si on peut toujours regretter que Microsoft ait encore raté celui de la mobilité.

Christopher Tran

Consultant fonctionnel SharePoint – MCNEXT
www.mcnext.com

OFFREZ-VOUS UN ABONNEMENT jusqu'à **-50%**

Code, gestion de projets, développement web, mobilité, Programmez ! est à la fois votre outil pratique, des articles de code par les meilleurs experts et votre veille technologique.

1

Abonnement 1 an au magazine

49 €* (au lieu de 65,45 €, prix au numéro)

2

Abonnement Intégral : 1 an au magazine + Archives Internet et PDF 59 €*

3

Abonnement 2 ans au magazine

78 €* (au lieu de 130,90 €, prix au numéro)

4

Abonnement intégral 2 ans au magazine + Archives Internet et PDF 88 €*

5

Abonnement Etudiant 1 an au magazine + Archives Internet et PDF 39 €* 49 €*



(*) Tarifs France métropolitaine

Toutes les offres en ligne : www.programmez.com

Abonnez-vous à partir de 3,80 € seulement par mois



Oui, je m'abonne

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

- Abonnement 1 an au magazine** : 49 €* (au lieu de 65,45 €, prix au numéro)
- Abonnement Intégral : 1 an au magazine + archives Internet et PDF** : 59 €* (au lieu de 65,45 €, prix au numéro)
- Abonnement 2 ans au magazine** : 78 €* (au lieu de 130,90 €, prix au numéro)
- Abonnement intégral 2 ans au magazine + archives Internet et PDF** : 88 €*
- Abonnement Etudiant : 1 an au magazine** : 39 €* (joindre copie carte étudiant)
- Abonnement Etudiant : 1 an au magazine + archives Internet** : 49 €* (joindre copie carte étudiant)

(*) Tarifs France métropolitaine

M. Mme Mlle Entreprise : _____ Fonction : _____

Prénom : _____ Nom : _____

Adresse : _____

Code postal : _____ Ville : _____

Tél : _____ (Attention, e-mail indispensable pour les archives sur internet)

E-mail : _____ @ _____

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

Je souhaite régler à réception de facture



La boîte à outils du développeur

© istockphoto/Jamie Farrant

Pour bien développer, il faut avoir les bons outils au bon moment pour le bon projet. Cela peut paraître une évidence mais c'est une réalité. Un environnement de développement peut faire gagner ou perdre beaucoup de temps. Le développement implique l'utilisation d'outils de plus en plus nombreux dont la spécialisation augmente.

Si l'IDE sera l'environnement de référence, on doit lui adjoindre de multiples compléments pour les tests, les bonnes pratiques, la sécurité, le debug, voire, pour l'intégration continue, la gestion des builds, etc. La notion d'outil est très large, cela va de l'IDE ultra complet à la

bibliothèque de composants, aux API. Un environnement de développement résulte de plusieurs facteurs :

- il peut être imposé
- certaines technologies, certains langages et contextes techniques ne laissent pas beaucoup de choix
- l'affinité du développeur avec l'outil
- trouver un bon compromis entre l'efficacité, la productivité et les contraintes

Comment choisir ? Comment définir son environnement de développement ? Quelles bonnes pratiques adopter ? Des développeurs vous donnent leurs recettes, leurs conseils.

François Tonic



L'outilage d'un développeur iOS - Windows

Je vous présente aujourd'hui les différents outils que j'utilise lors de mes développements web. Je ne vais pas vous parler d'environnements de travail pour le développement iOS et Windows 8 pour la simple raison qu'on est un peu forcé dans le choix de l'outil, à savoir Xcode pour iOS et Visual Studio pour Windows 8. Je vous présente en revanche une palette d'outils, que l'on aime ou que l'on n'aime pas, pour développer dans ce monde qu'est "internet".

Petite précision avant de commencer, je travaille sur Mac, pour les non Mac-addict, je voulais vous trouver des équivalents mais ne les ayant pas testés, ce ne serait pas très utile...

> L'éditeur de texte

Le nerf névralgique de la guerre ! C'est sur lui que vous allez passer 85% de votre temps, il vous en faut donc un qui vous plaise et qui soit utile ! Au placard le bloc-notes de base et bienvenue à l'éditeur intelligent ! Il en existe un certain nombre, je vais vous présenter les 3 avec lesquels j'ai eu l'occasion de travailler.

> Coda :

<http://panic.com/coda/>

Mon premier éditeur sous Mac. Design sympa, fonctionnalités intéressantes, il sera utile pour vous connecter directement aux FTP et modifier les fichiers en ligne. Un système de gestion de projet vous permet de retrouver votre dernier site rapidement. Il a une coloration syntaxique intéressante et vous permet de le configurer encore plus via un système de plugin téléchargeable.

En revanche, j'ai trouvé qu'il était un petit peu trop « usine à gaz », j'ai donc changé pour passer à TextMate.

> TextMate :

<http://macromates.com/>

Beaucoup plus light, beaucoup plus sobre, il conviendra à tous ceux qui aiment le minimalistique. Il a pourtant les fonctionnalités essentielles : la gestion des projets, la coloration syntaxique en fonction des langages et l'auto-complétion (même si elle reste encore à améliorer).

Autre point fort, il possède une ligne de commande qui permet d'ouvrir n'importe quel fichier text avec TextMate au lieu de VI ! Et ca, c'est tout simplement génial !

Il en existe encore beaucoup d'autres

comme Sublime Text, Smultron ou encore Espresso. Chacun à son point fort et son point faible. Un seul conseil : testez-les, travaillez avec quelque temps, et vous trouverez vite celui qu'il vous faut.

> Le logiciel FTP

Outre Filezilla et son design préhistorique, j'ai testé Transmit (<http://panic.com/transmit/>), de Panic Inc., la même société qui a développé Coda, et je peux dire que je ne le lâche plus. Il permet de gérer ses favoris, de modifier les fichiers en ligne, de gérer les droits sur les dossiers, bref, toutes les fonctionnalités classiques d'un FTP avec un petit plus : on peut monter virtuellement un disque réseau lié à un ftp et le gérer comme un dossier local. Mis à part cela, le design est très sympa.

> L'outil de gestion de base de données

Vous gérez encore vos bases de données via PhpMyAdmin ? Stop ! Fermez immédiatement cet onglet de votre navigateur et allez récupérer Sequel Pro (<http://www.sequelpro.com/>), qui est gratuit en plus, pour pouvoir vous connecter à vos DB. Avec une interface sympa, vous aller pouvoir vous connecter à vos bases normalement, ou via Socket ou encore SSH. Plus besoin de demander à l'admin sys d'installer PhpMyAdmin ou de perdre votre temps avec la console... Vous allez pouvoir gérer vos favoris, créer des nouvelles bases, tables et autres, gérer les utilisateurs, exécuter des requêtes, gérer les liaisons... bref, une petite révolution dans la gestion de bases de données.

Pour les utilisateurs Windows, je n'ai pu trouver d'équivalent aussi poussé que Sequel, mais ne désespérez pas, ça viendra un jour !

En complément de Sequel, je vous propose également l'outil MySQL Workbench (<http://www.mysql.fr/products/workbench/>), utile

pour gérer les structures de vos bases et avoir un mapping de vos tables.

> L'outil de versionning

Travaillant de plus en plus sur des projets à plusieurs développeurs, nous utilisons Git en guise de dépôt de travail. Il me permet également d'avoir une sauvegarde de mes projets lorsque je travaille seul. En guise d'outil, je ne me sers pas de la ligne de commande, ni même du logiciel Github mais de Tower (<http://www.git-tower.com/>). Il est un peu lourd, mais va vous être utile pour gérer vos dépôts Git (Github et autres). Je ne vais pas en parler davantage, car il reste un outil Git classique.

Pour le SVN, je me suis penché sur Versions (<http://versionsapp.com/>) qui est un équivalent à Tower mais pour SVN.

> Autres outils

Voici quelques outils qui sont en rapport plus ou moins direct avec le développement :

- 1Password : trousseau de mot de passe dont le fichier principal peut être partagé avec d'autres collaborateurs via DropBox
- Adium : clients de messagerie multi canal
- Sparrow : client mail un peu lourd mais utile
- Dropbox : sérieusement, j'ai besoin de vous dire ce que c'est ?!

Voilà donc une liste non exhaustive des différents outils avec lesquels je travaille. Il y en a d'autres, bien entendu, plus ou moins intéressants. Si je peux vous donner un conseil, fouinez, fouillez, cherchez sur le web les derniers logiciels à la mode, discutez avec d'autres devs qui pourront également vous en apporter des nouveaux...



Laurent Menu

Développeur passionné de "l'internet"

<http://www.lmenu.fr>



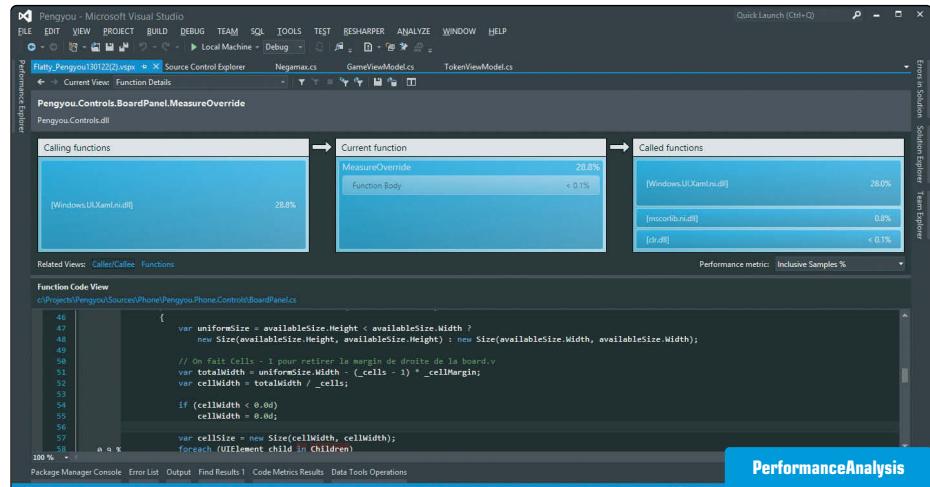
Boîte à outils Windows : les bonnes pratiques d'un développeur

Tout bon bricoleur se doit d'avoir une boîte à outils bien fournie. Et quoi de plus bricoleur qu'un développeur ? La seule différence c'est qu'on remplace les clefs Allen par quelque chose de quand même bien plus fun.

> L'environnement de développement

Ce n'est pas un secret, je suis un développeur .NET. Je développe essentiellement des applications à destination de deux plateformes, Windows Phone et Windows 8. J'ai aussi parfois à travailler sur des applications Silverlight, WPF ou même ASP.NET MVC. Mon environnement de développement est constitué d'un pc portable sous Windows 8 sur lequel j'ai installé **Visual Studio 2012** et le SDK Windows Phone 8. Afin de pouvoir tester correctement mes développements j'ai à ma disposition un téléphone Windows Phone 7, un autre sous Windows Phone 8 et enfin une tablette Surface.

Visual Studio supporte l'ajout de greffons dont certains sont tellement utiles qu'ils en deviennent indispensables. Parmi ces extensions indispensables se trouve le gestionnaire de paquet NuGet. Il permet d'accéder à des dépôts et d'installer des bibliothèques dans votre projet, automatiquement au bon endroit et avec toutes leurs dépendances. Certains projets font une utilisation encore



plus poussée de NuGet en proposant des lignes de commandes permettant de générer du code. Viennent ensuite une série d'extensions que j'utilise pour le développement plutôt orienté web : Microsoft Web Developer Tools, Visual Studio Extensions for Windows Library for Javascript, Web Essentials 2012 et enfin la petite dernière, TypeScript For Microsoft Visual Studio 2012. Ces extensions me permettent d'ajouter le support de LESS, de TypeScript, d'améliorer le support de WinJS, de l'auto-complétion et plein d'autres petites choses qui améliorent le quotidien du développeur et surtout sa productivité.

> Les bibliothèques

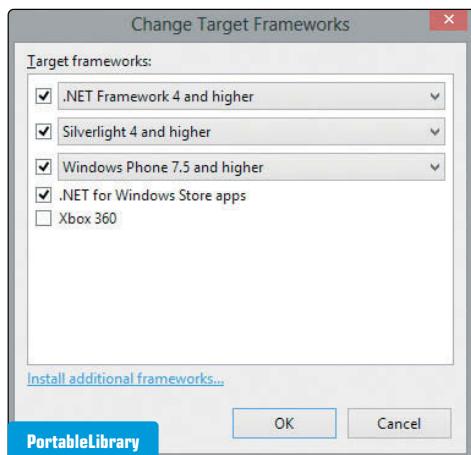
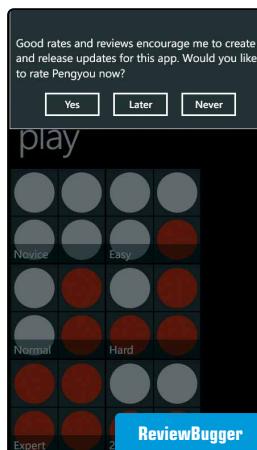
Outre les greffons il y a aussi un certain nombre de bibliothèques que j'ajoute presque à chaque fois. La première est MvvmLight. Elle apporte un support simple et robuste de Mvvm, du Messaging et surtout le petit bonus, un moteur d'inversion de contrôle (IOC) simple. Ce dernier fait toute la différence dans une application Windows Phone ou Windows 8 où on cherche en général à faire léger. Plus spécifique à Windows Phone on va retrouver le Silverlight Toolkit ainsi que le Coding4Fun Toolkit. Le premier apporte des contrôles graphiques plus classiques ou même la gestion de tests unitaires alors que le second apporte des

bonus ergonomiques et des contrôles un peu plus fun à vos applications ou des outils de surveillance de l'utilisation mémoire etc. J'adore vraiment le Coding4Fun Toolkit et je pense que vous aimerez aussi ses ToastPrompt, ses popups non bloquantes etc... J'ajoute aussi de manière quasi systématique YLAD (Your Last About Dialog) et ReviewBugger. YLAD est magique, elle génère pour vous cette page « A propos » qu'on oublie souvent de faire avant de publier l'application sur le Store. En plus, elle supporte la localisation et permet donc de traduire cette page dans toutes les langues déclarées par l'application. ReviewBugger fait partie des choses simples qui changent tout, il permet de compter le nombre de fois où a été lancée l'application et de demander à l'utilisateur de noter l'application au bout de n fois.

Voici pour la liste des bibliothèques que j'utilise le plus, il va sans dire que ces dernières sont toutes installables via NuGet.

> Contrôle de code sources

Puisqu'il faut bien entendu archiver ses sources j'utilise Team Foundation Server (TFS). Dans le cadre des projets d'entreprise c'est un TFS 2010 mais pour mes projets personnels, mon choix s'est porté sur TFS Online. Les deux s'intègrent parfaitement dans Visual Studio et c'est ce qui a arrêté mon choix. Bien que n'étant pas la personne





qui les a créés je profite des Builds automatisés et de l'intégration continue sur certains projets. C'est vraiment très pratique et permet de remonter les erreurs très rapidement pourvu que les tests unitaires soient faits. Et justement, puisqu'on parle de tests unitaires, j'utilise les tests intégrés dans Visual Studio c'est-à-dire les MsTests. De manière générale, en couplant tests unitaires, intégration continue et builds automatisés on gagne beaucoup en qualité et productivité. TFS offre de plus une gestion de la méthodologie agile en natif, ce qui est un plus lorsque les projets le nécessitent.

> Organisation du code

Afin d'améliorer la testabilité, l'isolation et le partage du code j'utilise aussi beaucoup de Portable Library. Les projets de type Portable Library sont intégrés dans Visual Studio 2012 et permettent de créer des bibliothèques que l'on peut référencer dans différents types de projets. A la différence d'une bibliothèque dont le code serait compatible avec plusieurs plateformes mais qui serait compilé pour chaque plateforme, les Portable Library ne nécessitent pas d'être compilés pour une plateforme spécifique. On configure la Portable Library pour cibler les plateformes intéressantes pour l'application et ensuite, seules les API communes aux plateformes sélectionnées deviennent disponibles. Par exemple, si on sélectionne Windows Phone 8 et Windows 8 on aura accès aux Task, mais si on sélectionne Windows Phone 7 et Windows 8 alors on n'aura pas accès. Il est donc important de bien choisir ses plateformes.

A titre d'exemple, j'ai développé mon application personnelle en utilisant une Portable Library qui cible Windows Phone 7, WPF 4.5, Silverlight 5 et Windows 8 ce qui, bien

que n'ayant pas été aisés, m'a forcé à avoir un code très simple et maintenable.

J'organise aussi généralement mes applications en plusieurs projets. Le projet principal contient les vues et les view models. Le modèle est contenu dans un autre projet de type Portable Library, de façon à pouvoir être utilisé dans une autre application ou bien même côté serveur. Un autre contient les services de l'application (accès aux données, appels réseaux, persistance) et un dernier contient les interfaces de ces services. Etant un fervent défenseur des principes DRY, KISS et SOLID j'utilise presque systématiquement de l'inversion de contrôle et de l'injection de dépendances sur ces services. Cela me permet de tester le comportement des view models qui utilisent ces services en injectant des bouchons.

> Quelques conseils

Au fil du temps j'ai acquis certains réflexes que je considère comme étant des bonnes pratiques mais qui n'engagent que moi.

La première d'entre elle est de toujours partir du principe qu'une application est multilingue et donc doit être développée comme telle. Tous les textes doivent être stockés dans des fichiers ressources et aucun texte en dur dans le code ou dans le XAML/HTML, ni aucun formatage de date ou de chiffres en dur ne doit être présent. C'est quelque chose d'assez contraignant au début mais qui rend de grands services assez rapidement.

La seconde est de loguer toutes les erreurs avec, si possible, un petit texte explicatif ou une exception personnalisée pour les erreurs auxquelles on s'attend le plus.

Quand bien même ces erreurs ne seraient pas stockées dans un fichier, rien que le fait de les afficher dans la console d'erreur peut

permettre de se sauver de quelques bugs difficilement détectables.

La troisième est de toujours faire en sorte de rafraîchir l'interface le plus rapidement possible. Par exemple, pour une application de news on va d'abord chercher les données en cache et les afficher ; pendant ce temps-là on peut télécharger les mises à jour et dès qu'une série d'articles est chargée, rafraîchir l'interface. Bref, tout ce qui améliore la sensation de vitesse et de fluidité que va ressentir l'utilisateur est bonne à prendre. J'avais d'ailleurs rédigé un article dans un précédent numéro de Programmez qui montrait comment améliorer le temps de chargement d'une application Windows 8 et qui expliquait toute cette problématique en détail. Le dernier point, et non des moindres, consiste à prendre régulièrement des mesures sur la performance de l'application. Visual Studio fournit un certain nombre d'outils pour cela. Certains, comme le Performance Analysis sont plutôt orientés utilisation CPU et énergie, tandis que d'autres, comme le Memory Analysis, se chargent de diagnostiquer l'utilisation mémoire, voir quand elle augmente, si elle diminue et donc de trouver les fuites ou arbres visuels trop lourds pour ce qui est requis par l'application. Plus tôt et plus régulièrement ces analyses sont faites, plus faciles sont les corrections. C'est le même principe qu'avec les tests unitaires, plus le temps avance et plus on accumule de la dette technique.

Et c'est ainsi que se finit le tour de ma boîte à outils. Puissiez-vous y avoir trouvé quelques matériels utiles.

John Thiriet

Consultant/Formateur à MCNEXT

<http://www.mcnext.com>

```

1138     private void GameTokensCollectionChanged(object sender,
1139     {
1140         OnGameTokensCollectionChanged();
1141     }
1142 }
1143 private void OnGameTokensCollectionChanged(TokenEventArgs e)
1144 {
1145     Token addedToken = null;
1146     if (e.AddedItems != null)
1147     {
1148         lock (SyncRoot)
1149         {
1150             foreach (var t in e.AddedItems.ToArray())
1151             {
1152                 if (addedToken == null)
1153                 addedToken = t;
1154             }
1155         }
1156     }
1157     int maxDistance = 0;
1158     var dict = new List<TokenViewModel>[8];
1159     if (e.UpdatedItems != null)
1160     {
1161         foreach (var t in e.UpdatedItems.ToArray())
1162         {
1163             if (dict[t.X] == null)
1164                 dict[t.X] = t;
1165             else
1166                 dict[t.X].Add(t);
1167             if (dict[t.X].Count > maxDistance)
1168                 maxDistance = dict[t.X].Count;
1169             if (dict[t.X].Count > 1)
1170                 dict[t.X].Sort((x, y) => x.Y - y.Y);
1171             if (dict[t.X].Count > 1)
1172                 dict[t.X].RemoveAt(1);
1173             if (dict[t.X].Count > 1)
1174                 dict[t.X].Sort((x, y) => x.Y - y.Y);
1175             if (dict[t.X].Count > 1)
1176                 dict[t.X].RemoveAt(1);
1177             if (dict[t.X].Count > 1)
1178                 dict[t.X].Sort((x, y) => x.Y - y.Y);
1179             if (dict[t.X].Count > 1)
1180                 dict[t.X].RemoveAt(1);
1181             if (dict[t.X].Count > 1)
1182                 dict[t.X].Sort((x, y) => x.Y - y.Y);
1183             if (dict[t.X].Count > 1)
1184                 dict[t.X].RemoveAt(1);
1185             if (dict[t.X].Count > 1)
1186                 dict[t.X].Sort((x, y) => x.Y - y.Y);
1187             if (dict[t.X].Count > 1)
1188                 dict[t.X].RemoveAt(1);
1189             if (dict[t.X].Count > 1)
1190                 dict[t.X].Sort((x, y) => x.Y - y.Y);
1191             if (dict[t.X].Count > 1)
1192                 dict[t.X].RemoveAt(1);
1193             if (dict[t.X].Count > 1)
1194                 dict[t.X].Sort((x, y) => x.Y - y.Y);
1195             if (dict[t.X].Count > 1)
1196                 dict[t.X].RemoveAt(1);
1197             if (dict[t.X].Count > 1)
1198                 dict[t.X].Sort((x, y) => x.Y - y.Y);
1199             if (dict[t.X].Count > 1)
1200                 dict[t.X].RemoveAt(1);
1201             if (dict[t.X].Count > 1)
1202                 dict[t.X].Sort((x, y) => x.Y - y.Y);
1203             if (dict[t.X].Count > 1)
1204                 dict[t.X].RemoveAt(1);
1205             if (dict[t.X].Count > 1)
1206                 dict[t.X].Sort((x, y) => x.Y - y.Y);
1207             if (dict[t.X].Count > 1)
1208                 dict[t.X].RemoveAt(1);
1209             if (dict[t.X].Count > 1)
1210                 dict[t.X].Sort((x, y) => x.Y - y.Y);
1211             if (dict[t.X].Count > 1)
1212                 dict[t.X].RemoveAt(1);
1213             if (dict[t.X].Count > 1)
1214                 dict[t.X].Sort((x, y) => x.Y - y.Y);
1215             if (dict[t.X].Count > 1)
1216                 dict[t.X].RemoveAt(1);
1217             if (dict[t.X].Count > 1)
1218                 dict[t.X].Sort((x, y) => x.Y - y.Y);
1219             if (dict[t.X].Count > 1)
1220                 dict[t.X].RemoveAt(1);
1221             if (dict[t.X].Count > 1)
1222                 dict[t.X].Sort((x, y) => x.Y - y.Y);
1223             if (dict[t.X].Count > 1)
1224                 dict[t.X].RemoveAt(1);
1225             if (dict[t.X].Count > 1)
1226                 dict[t.X].Sort((x, y) => x.Y - y.Y);
1227             if (dict[t.X].Count > 1)
1228                 dict[t.X].RemoveAt(1);
1229             if (dict[t.X].Count > 1)
1230                 dict[t.X].Sort((x, y) => x.Y - y.Y);
1231             if (dict[t.X].Count > 1)
1232                 dict[t.X].RemoveAt(1);
1233             if (dict[t.X].Count > 1)
1234                 dict[t.X].Sort((x, y) => x.Y - y.Y);
1235             if (dict[t.X].Count > 1)
1236                 dict[t.X].RemoveAt(1);
1237             if (dict[t.X].Count > 1)
1238                 dict[t.X].Sort((x, y) => x.Y - y.Y);
1239             if (dict[t.X].Count > 1)
1240                 dict[t.X].RemoveAt(1);
1241             if (dict[t.X].Count > 1)
1242                 dict[t.X].Sort((x, y) => x.Y - y.Y);
1243             if (dict[t.X].Count > 1)
1244                 dict[t.X].RemoveAt(1);
1245             if (dict[t.X].Count > 1)
1246                 dict[t.X].Sort((x, y) => x.Y - y.Y);
1247             if (dict[t.X].Count > 1)
1248                 dict[t.X].RemoveAt(1);
1249             if (dict[t.X].Count > 1)
1250                 dict[t.X].Sort((x, y) => x.Y - y.Y);
1251             if (dict[t.X].Count > 1)
1252                 dict[t.X].RemoveAt(1);
1253             if (dict[t.X].Count > 1)
1254                 dict[t.X].Sort((x, y) => x.Y - y.Y);
1255             if (dict[t.X].Count > 1)
1256                 dict[t.X].RemoveAt(1);
1257             if (dict[t.X].Count > 1)
1258                 dict[t.X].Sort((x, y) => x.Y - y.Y);
1259             if (dict[t.X].Count > 1)
1260                 dict[t.X].RemoveAt(1);
1261             if (dict[t.X].Count > 1)
1262                 dict[t.X].Sort((x, y) => x.Y - y.Y);
1263             if (dict[t.X].Count > 1)
1264                 dict[t.X].RemoveAt(1);
1265             if (dict[t.X].Count > 1)
1266                 dict[t.X].Sort((x, y) => x.Y - y.Y);
1267             if (dict[t.X].Count > 1)
1268                 dict[t.X].RemoveAt(1);
1269             if (dict[t.X].Count > 1)
1270                 dict[t.X].Sort((x, y) => x.Y - y.Y);
1271             if (dict[t.X].Count > 1)
1272                 dict[t.X].RemoveAt(1);
1273             if (dict[t.X].Count > 1)
1274                 dict[t.X].Sort((x, y) => x.Y - y.Y);
1275             if (dict[t.X].Count > 1)
1276                 dict[t.X].RemoveAt(1);
1277             if (dict[t.X].Count > 1)
1278                 dict[t.X].Sort((x, y) => x.Y - y.Y);
1279             if (dict[t.X].Count > 1)
1280                 dict[t.X].RemoveAt(1);
1281             if (dict[t.X].Count > 1)
1282                 dict[t.X].Sort((x, y) => x.Y - y.Y);
1283             if (dict[t.X].Count > 1)
1284                 dict[t.X].RemoveAt(1);
1285             if (dict[t.X].Count > 1)
1286                 dict[t.X].Sort((x, y) => x.Y - y.Y);
1287             if (dict[t.X].Count > 1)
1288                 dict[t.X].RemoveAt(1);
1289             if (dict[t.X].Count > 1)
1290                 dict[t.X].Sort((x, y) => x.Y - y.Y);
1291             if (dict[t.X].Count > 1)
1292                 dict[t.X].RemoveAt(1);
1293             if (dict[t.X].Count > 1)
1294                 dict[t.X].Sort((x, y) => x.Y - y.Y);
1295             if (dict[t.X].Count > 1)
1296                 dict[t.X].RemoveAt(1);
1297             if (dict[t.X].Count > 1)
1298                 dict[t.X].Sort((x, y) => x.Y - y.Y);
1299             if (dict[t.X].Count > 1)
1300                 dict[t.X].RemoveAt(1);
1301             if (dict[t.X].Count > 1)
1302                 dict[t.X].Sort((x, y) => x.Y - y.Y);
1303             if (dict[t.X].Count > 1)
1304                 dict[t.X].RemoveAt(1);
1305             if (dict[t.X].Count > 1)
1306                 dict[t.X].Sort((x, y) => x.Y - y.Y);
1307             if (dict[t.X].Count > 1)
1308                 dict[t.X].RemoveAt(1);
1309             if (dict[t.X].Count > 1)
1310                 dict[t.X].Sort((x, y) => x.Y - y.Y);
1311             if (dict[t.X].Count > 1)
1312                 dict[t.X].RemoveAt(1);
1313             if (dict[t.X].Count > 1)
1314                 dict[t.X].Sort((x, y) => x.Y - y.Y);
1315             if (dict[t.X].Count > 1)
1316                 dict[t.X].RemoveAt(1);
1317             if (dict[t.X].Count > 1)
1318                 dict[t.X].Sort((x, y) => x.Y - y.Y);
1319             if (dict[t.X].Count > 1)
1320                 dict[t.X].RemoveAt(1);
1321             if (dict[t.X].Count > 1)
1322                 dict[t.X].Sort((x, y) => x.Y - y.Y);
1323             if (dict[t.X].Count > 1)
1324                 dict[t.X].RemoveAt(1);
1325             if (dict[t.X].Count > 1)
1326                 dict[t.X].Sort((x, y) => x.Y - y.Y);
1327             if (dict[t.X].Count > 1)
1328                 dict[t.X].RemoveAt(1);
1329             if (dict[t.X].Count > 1)
1330                 dict[t.X].Sort((x, y) => x.Y - y.Y);
1331             if (dict[t.X].Count > 1)
1332                 dict[t.X].RemoveAt(1);
1333             if (dict[t.X].Count > 1)
1334                 dict[t.X].Sort((x, y) => x.Y - y.Y);
1335             if (dict[t.X].Count > 1)
1336                 dict[t.X].RemoveAt(1);
1337             if (dict[t.X].Count > 1)
1338                 dict[t.X].Sort((x, y) => x.Y - y.Y);
1339             if (dict[t.X].Count > 1)
1340                 dict[t.X].RemoveAt(1);
1341             if (dict[t.X].Count > 1)
1342                 dict[t.X].Sort((x, y) => x.Y - y.Y);
1343             if (dict[t.X].Count > 1)
1344                 dict[t.X].RemoveAt(1);
1345             if (dict[t.X].Count > 1)
1346                 dict[t.X].Sort((x, y) => x.Y - y.Y);
1347             if (dict[t.X].Count > 1)
1348                 dict[t.X].RemoveAt(1);
1349             if (dict[t.X].Count > 1)
1350                 dict[t.X].Sort((x, y) => x.Y - y.Y);
1351             if (dict[t.X].Count > 1)
1352                 dict[t.X].RemoveAt(1);
1353             if (dict[t.X].Count > 1)
1354                 dict[t.X].Sort((x, y) => x.Y - y.Y);
1355             if (dict[t.X].Count > 1)
1356                 dict[t.X].RemoveAt(1);
1357             if (dict[t.X].Count > 1)
1358                 dict[t.X].Sort((x, y) => x.Y - y.Y);
1359             if (dict[t.X].Count > 1)
1360                 dict[t.X].RemoveAt(1);
1361             if (dict[t.X].Count > 1)
1362                 dict[t.X].Sort((x, y) => x.Y - y.Y);
1363             if (dict[t.X].Count > 1)
1364                 dict[t.X].RemoveAt(1);
1365             if (dict[t.X].Count > 1)
1366                 dict[t.X].Sort((x, y) => x.Y - y.Y);
1367             if (dict[t.X].Count > 1)
1368                 dict[t.X].RemoveAt(1);
1369             if (dict[t.X].Count > 1)
1370                 dict[t.X].Sort((x, y) => x.Y - y.Y);
1371             if (dict[t.X].Count > 1)
1372                 dict[t.X].RemoveAt(1);
1373             if (dict[t.X].Count > 1)
1374                 dict[t.X].Sort((x, y) => x.Y - y.Y);
1375             if (dict[t.X].Count > 1)
1376                 dict[t.X].RemoveAt(1);
1377             if (dict[t.X].Count > 1)
1378                 dict[t.X].Sort((x, y) => x.Y - y.Y);
1379             if (dict[t.X].Count > 1)
1380                 dict[t.X].RemoveAt(1);
1381             if (dict[t.X].Count > 1)
1382                 dict[t.X].Sort((x, y) => x.Y - y.Y);
1383             if (dict[t.X].Count > 1)
1384                 dict[t.X].RemoveAt(1);
1385             if (dict[t.X].Count > 1)
1386                 dict[t.X].Sort((x, y) => x.Y - y.Y);
1387             if (dict[t.X].Count > 1)
1388                 dict[t.X].RemoveAt(1);
1389             if (dict[t.X].Count > 1)
1390                 dict[t.X].Sort((x, y) => x.Y - y.Y);
1391             if (dict[t.X].Count > 1)
1392                 dict[t.X].RemoveAt(1);
1393             if (dict[t.X].Count > 1)
1394                 dict[t.X].Sort((x, y) => x.Y - y.Y);
1395             if (dict[t.X].Count > 1)
1396                 dict[t.X].RemoveAt(1);
1397             if (dict[t.X].Count > 1)
1398                 dict[t.X].Sort((x, y) => x.Y - y.Y);
1399             if (dict[t.X].Count > 1)
1400                 dict[t.X].RemoveAt(1);
1401             if (dict[t.X].Count > 1)
1402                 dict[t.X].Sort((x, y) => x.Y - y.Y);
1403             if (dict[t.X].Count > 1)
1404                 dict[t.X].RemoveAt(1);
1405             if (dict[t.X].Count > 1)
1406                 dict[t.X].Sort((x, y) => x.Y - y.Y);
1407             if (dict[t.X].Count > 1)
1408                 dict[t.X].RemoveAt(1);
1409             if (dict[t.X].Count > 1)
1410                 dict[t.X].Sort((x, y) => x.Y - y.Y);
1411             if (dict[t.X].Count > 1)
1412                 dict[t.X].RemoveAt(1);
1413             if (dict[t.X].Count > 1)
1414                 dict[t.X].Sort((x, y) => x.Y - y.Y);
1415             if (dict[t.X].Count > 1)
1416                 dict[t.X].RemoveAt(1);
1417             if (dict[t.X].Count > 1)
1418                 dict[t.X].Sort((x, y) => x.Y - y.Y);
1419             if (dict[t.X].Count > 1)
1420                 dict[t.X].RemoveAt(1);
1421             if (dict[t.X].Count > 1)
1422                 dict[t.X].Sort((x, y) => x.Y - y.Y);
1423             if (dict[t.X].Count > 1)
1424                 dict[t.X].RemoveAt(1);
1425             if (dict[t.X].Count > 1)
1426                 dict[t.X].Sort((x, y) => x.Y - y.Y);
1427             if (dict[t.X].Count > 1)
1428                 dict[t.X].RemoveAt(1);
1429             if (dict[t.X].Count > 1)
1430                 dict[t.X].Sort((x, y) => x.Y - y.Y);
1431             if (dict[t.X].Count > 1)
1432                 dict[t.X].RemoveAt(1);
1433             if (dict[t.X].Count > 1)
1434                 dict[t.X].Sort((x, y) => x.Y - y.Y);
1435             if (dict[t.X].Count > 1)
1436                 dict[t.X].RemoveAt(1);
1437             if (dict[t.X].Count > 1)
1438                 dict[t.X].Sort((x, y) => x.Y - y.Y);
1439             if (dict[t.X].Count > 1)
1440                 dict[t.X].RemoveAt(1);
1441             if (dict[t.X].Count > 1)
1442                 dict[t.X].Sort((x, y) => x.Y - y.Y);
1443             if (dict[t.X].Count > 1)
1444                 dict[t.X].RemoveAt(1);
1445             if (dict[t.X].Count > 1)
1446                 dict[t.X].Sort((x, y) => x.Y - y.Y);
1447             if (dict[t.X].Count > 1)
1448                 dict[t.X].RemoveAt(1);
1449             if (dict[t.X].Count > 1)
1450                 dict[t.X].Sort((x, y) => x.Y - y.Y);
1451             if (dict[t.X].Count > 1)
1452                 dict[t.X].RemoveAt(1);
1453             if (dict[t.X].Count > 1)
1454                 dict[t.X].Sort((x, y) => x.Y - y.Y);
1455             if (dict[t.X].Count > 1)
1456                 dict[t.X].RemoveAt(1);
1457             if (dict[t.X].Count > 1)
1458                 dict[t.X].Sort((x, y) => x.Y - y.Y);
1459             if (dict[t.X].Count > 1)
1460                 dict[t.X].RemoveAt(1);
1461             if (dict[t.X].Count > 1)
1462                 dict[t.X].Sort((x, y) => x.Y - y.Y);
1463             if (dict[t.X].Count > 1)
1464                 dict[t.X].RemoveAt(1);
1465             if (dict[t.X].Count > 1)
1466                 dict[t.X].Sort((x, y) => x.Y - y.Y);
1467             if (dict[t.X].Count > 1)
1468                 dict[t.X].RemoveAt(1);
1469             if (dict[t.X].Count > 1)
1470                 dict[t.X].Sort((x, y) => x.Y - y.Y);
1471             if (dict[t.X].Count > 1)
1472                 dict[t.X].RemoveAt(1);
1473             if (dict[t.X].Count > 1)
1474                 dict[t.X].Sort((x, y) => x.Y - y.Y);
1475             if (dict[t.X].Count > 1)
1476                 dict[t.X].RemoveAt(1);
1477             if (dict[t.X].Count > 1)
1478                 dict[t.X].Sort((x, y) => x.Y - y.Y);
1479             if (dict[t.X].Count > 1)
1480                 dict[t.X].RemoveAt(1);
1481             if (dict[t.X].Count > 1)
1482                 dict[t.X].Sort((x, y) => x.Y - y.Y);
1483             if (dict[t.X].Count > 1)
1484                 dict[t.X].RemoveAt(1);
1485             if (dict[t.X].Count > 1)
1486                 dict[t.X].Sort((x, y) => x.Y - y.Y);
1487             if (dict[t.X].Count > 1)
1488                 dict[t.X].RemoveAt(1);
1489             if (dict[t.X].Count > 1)
1490                 dict[t.X].Sort((x, y) => x.Y - y.Y);
1491             if (dict[t.X].Count > 1)
1492                 dict[t.X].RemoveAt(1);
1493             if (dict[t.X].Count > 1)
1494                 dict[t.X].Sort((x, y) => x.Y - y.Y);
1495             if (dict[t.X].Count > 1)
1496                 dict[t.X].RemoveAt(1);
1497             if (dict[t.X].Count > 1)
1498                 dict[t.X].Sort((x, y) => x.Y - y.Y);
1499             if (dict[t.X].Count > 1)
1500                 dict[t.X].RemoveAt(1);
1501             if (dict[t.X].Count > 1)
1502                 dict[t.X].Sort((x, y) => x.Y - y.Y);
1503             if (dict[t.X].Count > 1)
1504                 dict[t.X].RemoveAt(1);
1505             if (dict[t.X].Count > 1)
1506                 dict[t.X].Sort((x, y) => x.Y - y.Y);
1507             if (dict[t.X].Count > 1)
1508                 dict[t.X].RemoveAt(1);
1509             if (dict[t.X].Count > 1)
1510                 dict[t.X].Sort((x, y) => x.Y - y.Y);
1511             if (dict[t.X].Count > 1)
1512                 dict[t.X].RemoveAt(1);
1513             if (dict[t.X].Count > 1)
1514                 dict[t.X].Sort((x, y) => x.Y - y.Y);
1515             if (dict[t.X].Count > 1)
1516                 dict[t.X].RemoveAt(1);
1517             if (dict[t.X].Count > 1)
1518                 dict[t.X].Sort((x, y) => x.Y - y.Y);
1519             if (dict[t.X].Count > 1)
1520                 dict[t.X].RemoveAt(1);
1521             if (dict[t.X].Count > 1)
1522                 dict[t.X].Sort((x, y) => x.Y - y.Y);
1523             if (dict[t.X].Count > 1)
1524                 dict[t.X].RemoveAt(1);
1525             if (dict[t.X].Count > 1)
1526                 dict[t.X].Sort((x, y) => x.Y - y.Y);
1527             if (dict[t.X].Count > 1)
1528                 dict[t.X].RemoveAt(1);
1529             if (dict[t.X].Count > 1)
1530                 dict[t.X].Sort((x, y) => x.Y - y.Y);
1531             if (dict[t.X].Count > 1)
1532                 dict[t.X].RemoveAt(1);
1533             if (dict[t.X].Count > 1)
1534                 dict[t.X].Sort((x, y) => x.Y - y.Y);
1535             if (dict[t.X].Count > 1)
1536                 dict[t.X].RemoveAt(1);
1537             if (dict[t.X].Count > 1)
1538                 dict[t.X].Sort((x, y) => x.Y - y.Y);
1539             if (dict[t.X].Count > 1)
1540                 dict[t.X].RemoveAt(1);
1541             if (dict[t.X].Count > 1)
1542                 dict[t.X].Sort((x, y) => x.Y - y.Y);
1543             if (dict[t.X].Count > 1)
1544                 dict[t.X].RemoveAt(1);
1545             if (dict[t.X].Count > 1)
1546                 dict[t.X].Sort((x, y) => x.Y - y.Y);
1547             if (dict[t.X].Count > 1)
1548                 dict[t.X].RemoveAt(1);
1549             if (dict[t.X].Count > 1)
1550                 dict[t.X].Sort((x, y) => x.Y - y.Y);
1551             if (dict[t.X].Count > 1)
1552                 dict[t.X].RemoveAt(1);
1553             if (dict[t.X].Count > 1)
1554                 dict[t.X].Sort((x, y) => x.Y - y.Y);
1555             if (dict[t.X].Count > 1)
1556                 dict[t.X].RemoveAt(1);
1557             if (dict[t.X].Count > 1)
1558                 dict[t.X].Sort((x, y) => x.Y - y.Y);
1559             if (dict[t.X].Count > 1)
1560                 dict[t.X].RemoveAt(1);
1561             if (dict[t.X].Count > 1)
1562                 dict[t.X].Sort((x, y) => x.Y - y.Y);
1563             if (dict[t.X].Count > 1)
1564                 dict[t.X].RemoveAt(1);
1565             if (dict[t.X].Count > 1)
1566                 dict[t.X].Sort((x, y) => x.Y - y.Y);
1567             if (dict[t.X].Count > 1)
1568                 dict[t.X].RemoveAt(1);
1569             if (dict[t.X].Count > 1)
1570                 dict[t.X].Sort((x, y) => x.Y - y.Y);
1571             if (dict[t.X].Count > 1)
1572                 dict[t.X].RemoveAt(1);
1573             if (dict[t.X].Count > 1)
1574                 dict[t.X].Sort((x, y) => x.Y - y.Y);
1575             if (dict[t.X].Count > 1)
1576                 dict[t.X].RemoveAt(1);
1577             if (dict[t.X].Count > 1)
1578                 dict[t.X].Sort((x, y) => x.Y - y.Y);
1579             if (dict[t.X].Count > 1)
1580                 dict[t.X].RemoveAt(1);
1581             if (dict[t.X].Count > 1)
1582                 dict[t.X].Sort((x, y) => x.Y - y.Y);
1583             if (dict[t.X].Count > 1)
1584                 dict[t.X].RemoveAt(1);
1585             if (dict[t.X].Count > 1)
1586                 dict[t.X].Sort((x, y) => x.Y - y.Y);
1587             if (dict[t.X].Count > 1)
1588                 dict[t.X].RemoveAt(1);
1589             if (dict[t.X].Count > 1)
1590                 dict[t.X].Sort((x, y) => x.Y - y.Y);
1591             if (dict[t.X].Count > 1)
1592                 dict[t.X].RemoveAt(1);
1593             if (dict[t.X].Count > 1)
1594                 dict[t.X].Sort((x, y) => x.Y - y.Y);
1595             if (dict[t.X].Count > 1)
1596                 dict[t.X].RemoveAt(1);
1597             if (dict[t.X].Count > 1)
1598                 dict[t.X].Sort((x, y) => x.Y - y.Y);
1599             if (dict[t.X].Count > 1)
1600                 dict[t.X].RemoveAt(1);
1601             if (dict[t.X].Count > 1)
1602                 dict[t.X].Sort((x, y) => x.Y - y.Y);
1603             if (dict[t.X].Count > 1)
1604                 dict[t.X].RemoveAt(1);
1605             if (dict[t.X].Count > 1)
1606                 dict[t.X].Sort((x, y) => x.Y - y.Y);
1607             if (dict[t.X].Count > 1)
1608                 dict[t.X].RemoveAt(1);
1609             if (dict[t.X].Count > 1)
1610                 dict[t.X].Sort((x, y) => x.Y - y.Y);
1611             if (dict[t.X].Count > 1)
1612                 dict[t.X].RemoveAt(1);
1613             if (dict[t.X].Count > 1)
1614                 dict[t.X].Sort((x, y) => x.Y - y.Y);
1615             if (dict[t.X].Count > 1)
1616                 dict[t.X].RemoveAt(1);
1617             if (dict[t.X].Count > 1)
1618                 dict[t.X].Sort((x, y) => x.Y - y.Y);
1619             if (dict[t.X].Count > 1)
1620                 dict[t.X].RemoveAt(1);
1621             if (dict[t.X].Count > 1)
1622                 dict[t.X].Sort((x, y) => x.Y - y.Y);
1623             if (dict[t.X].Count > 1)
1624                 dict[t.X].RemoveAt(1);
1625             if (dict[t.X].Count > 1)
1626                 dict[t.X].Sort((x, y) => x.Y - y.Y);
1627             if (dict[t.X].Count > 1)
1628                 dict[t.X].RemoveAt(1);
1629             if (dict[t.X].Count > 1)
1630                 dict[t.X].Sort((x, y) => x.Y - y.Y);
1631             if (dict[t.X].Count > 1)
1632                 dict[t.X].RemoveAt(1);
1633             if (dict[t.X].Count > 1)
1634                 dict[t.X].Sort((x, y) => x.Y - y.Y);
1635             if (dict[t.X].Count > 1)
1636                 dict[t.X].RemoveAt(1);
1637             if (dict[t.X].Count > 1)
1638                 dict[t.X].Sort((x, y) => x.Y - y.Y);
1639             if (dict[t.X].Count > 1)
1640                 dict[t.X].RemoveAt(1);
1641             if (dict[t.X].Count > 1)
1642                 dict[t.X].Sort((x, y) => x.Y - y.Y);
1643             if (dict[t.X].Count > 1)
1644                 dict[t.X].RemoveAt(1);
1645             if (dict[t.X].Count > 1)
1646                 dict[t.X].Sort((x, y) => x.Y - y.Y);
1647             if (dict[t.X].Count > 1)
1648                 dict[t.X].RemoveAt(1);
1649             if (dict[t.X].Count > 1)
1650                 dict[t.X].Sort((x, y) => x.Y - y.Y);
1651             if (dict[t.X].Count > 1)
1652                 dict[t.X].RemoveAt(1);
1653             if (dict[t.X].Count > 1)
1654                 dict[t.X].Sort((x, y) => x.Y - y.Y);
1655             if (dict[t.X].Count > 1)
1656                 dict[t.X].RemoveAt(1);
1657             if (dict[t.X].Count > 1)
1658                 dict[t.X].Sort((x, y) => x.Y - y.Y);
1659             if (dict[t.X].Count &gt
```



Le développement iOS au quotidien : quels outils utiliser ?

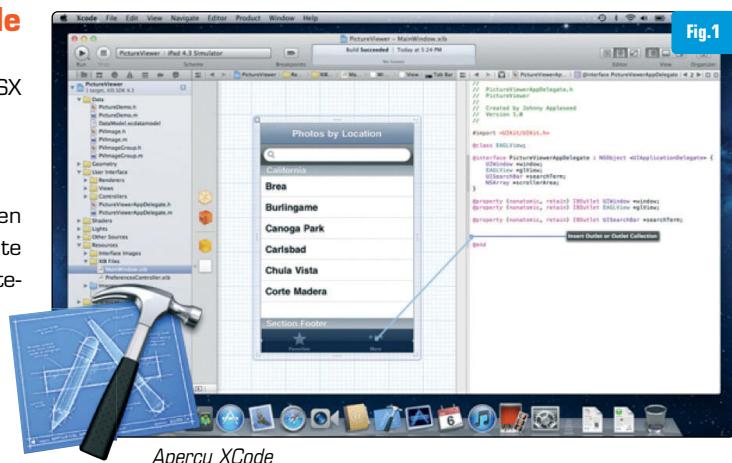
Lorsque l'on commence un développement, la première question que l'on se pose est : "Quel environnement de développement vais-je utiliser ?". Les habitués des langages populaires tels que Java ou C++ auront l'habitude de devoir choisir parmi une multitude d'EDI. Les puristes préféreront sûrement l'utilisation de 'Vi'. Cependant, Apple a choisi de proposer directement le sien, XCode. Celui-ci est spécialement adapté au développement pour les plateformes Apple et remplit parfaitement son rôle.

> L'environnement de développement : XCode

XCode fournit un ensemble d'outils :

- Interface Builder - générateur d'interfaces graphiques pour iOS et OSX
- Instruments - outil d'analyse des performances
- Apple LLVM - compilateur C/Objective-C
- iOS Simulator - simulateur d'iOS [Fig.1].

De plus, XCode 4 a ajouté le versionning via l'utilisation de Git en natif, de quoi faciliter le travail en équipe. A noter que si l'on souhaite développer des applications pour l'App Store d'Apple, il est fortement recommandé de développer son application avec XCode. Celui-ci intègre directement les outils pour pouvoir envoyer son application sur le store, à condition bien sûr que vous ayez souscrit au programme de développeur Apple (~ 79€)



On aime

- Interface Builder
- Intégration de Git native
- Intégration dans OS X

On aime moins

Peu de personnalisation de l'interface

NimbusKit

<http://nimbuskit.info/>

Ce framework est une réécriture d'un ancien framework : Three20. Les développeurs ont choisi de réécrire totalement Three20 car celui-ci n'était pas documenté correctement. Cela a conduit à un nouveau framework, NimbusKit, avec pour objectif : « Toute fonctionnalité non documentée correctement ne sera pas publiée ».

NimbusKit propose une multitude de composants permettant de combler certaines lacunes du framework d'Apple : une simplification de certains composants natifs (tels que les labels, les tables, etc.), l'implémentation de composants (une grille, les badges, etc.) et des outils permettant de faciliter le développement quotidien tels qu'un outil de débogage visuellement intégré à une application.

On aime

- La richesse des composants
- La documentation associée

On aime moins

Rien

SSToolkit

<http://sstoolkit.it/>

Ce framework est dans le même esprit que NimbusKit : bien documenté, il met à disposition des composants facilitant le quotidien. SSToolkit n'est pas meilleur que NimbusKit, ils proposent des concepts similaires et fournissent chacun leurs propres composants. Le choix entre les deux sera établi selon les fonctionnalités que l'un ou l'autre propose.

On aime

- La richesse des composants
- La documentation associée

On aime moins

Rien

> La documentation Apple

Tout développeur a déjà été confronté à un problème majeur : une documentation bien faite. Apple a énormément misé sur cet aspect. A l'instar de MSDN de Microsoft, la documentation d'Apple sur le développement en Objective-C est très complète, aussi bien en informations qu'en exemples. Il en est de même pour l'ensemble de ses frameworks. Voici la documentation iOS en ligne :

<https://developer.apple.com/library/ios/navigation/>

Celle-ci est organisée en plusieurs domaines :

- Type de ressource (Article, Tutoriel, Exemple de code, etc.)
- Sujet (XCode, Sécurité, Audio, Data, etc.)
- Framework (Cocoa Touch, Core Services, etc.)

De plus, un formulaire de recherche permet de trouver facilement une documentation sur un sujet, ou encore le document de référence d'une classe native de Cocoa.

On aime

- L'intégration de la documentation dans XCode
- La recherche avancée
- Les exemples fournis par Apple

On aime moins

L'impossibilité pour les utilisateurs de contribuer à la documentation

> Les frameworks utiles à connaître

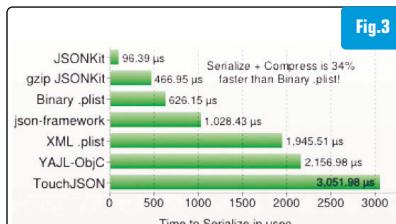
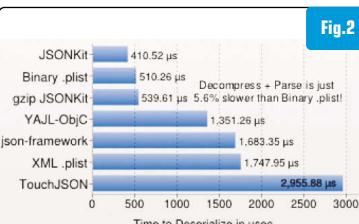
Ni le langage Objective-C, ni le framework CocoaTouch ne sont compliqués à utiliser. Cependant, comme dans bien des langages, il existe des bibliothèques (ou frameworks) qui permettent de simplifier leur utilisation. Voici une liste très utile. Bien sûr elle n'est pas exhaustive car de nombreuses bibliothèques de qualité existent.



JSONKit

<https://github.com/johnnezang/JSONKit>

JSONKit permet de simplifier la manipulation du format JSON (Fichier, chaîne de caractères, etc.). Il en existe plusieurs du même type (SBJson, TouchJSON, etc.), cependant d'après John Engelhart, le développeur de JSONKit, celui-ci est le plus rapide.



fonctionnalité qu'AFNetworking, ils peuvent être totalement complémentaires. Site Web : <https://github.com/square/SocketRocket>

On aime

Le respect de la norme RFC6455

La simplicité d'utilisation

On aime moins

L'absence d'implémentation serveur

RestKit

Ce framework, basé sur AFNetworking, permet de mapper directement les retours d'un web service à l'objet relationnel d'Apple (basé généralement sur l'utilisation d'une base de données) : CoreData. RestKit permet la sérialisation / désérialisation de requêtes au format JSON et YAJL.

Sites Web : <https://github.com/RestKit/RestKit> / <http://restkit.org>

On aime

La réutilisation du framework AFNetworking

On aime moins

Le peu de documentation

L'intégration avec CoreData

> Les frameworks 2D/3D

Lorsque l'on pense au développement mobile, on pense forcément un jour au jeu. Ayant pris une place conséquente sur ces plateformes, voici quelques frameworks qui vous permettront d'aborder le sujet plus facilement.

Cocos2D

Le plus connu, et sûrement le plus utilisé est sans aucun doute Cocos2D. Il propose une API simple et complète permettant de développer un jeu en 2D de A à Z. Une réécriture de Cocos2D, appelée Cocos2D-X, permet le développement multiplateforme en s'appuyant sur le langage C++ (ou JavaScript). La liste des plateformes supportées s'étend régulièrement : Android, iOS, Windows Phone, etc.

Site Web Cocos2D pour iPhone : <http://www.cocos2d-iphone.org>

Site Web Cocos2D-X : <http://www.cocos2d-x.org>

On aime

Le portage vers un environnement multiplateforme

La simplicité d'utilisation

On aime moins

L'absence de développement 3D

CoronaSDK

CoronaSDK est un framework permettant le développement d'applications multiplateformes unifiées sous un seul et même langage : Lua. Bien que CoronaSDK ne soit pas uniquement dédié au développement 2D/3D, il est souvent utilisé pour ça. Et pour cause, les développeurs ont totalement réécrit leur propre moteur de rendu OpenGL-ES, entièrement utilisable avec le langage Lua.

Site Web : <http://www.coronalabs.com/products/corona-sdk/>

On aime

L'aspect multiplateforme

La réécriture du moteur de rendu OpenGL-ES

On aime moins

L'utilisation d'un langage de script peu connu (on aurait préféré du JavaScript)

Unity3D

De plus en plus utilisé pour les jeux AAA, Unity3D est le framework 3D à utiliser pour un développement multiplateforme ultra-complet. Unity3D fournit un environnement de développement spécifique, composé de MonoDevelop et d'un environnement de conception 3D simple d'utilisation. Il permet le développement dans 3 langages différents : C# (Mono), JavaScript et Boo.

Il est toutefois nécessaire d'avoir une licence pour pouvoir développer des jeux sur mobile et les publier sur les stores.

À noter : son aspect multiplateforme dépasse de loin celui de CoronaSDK ou de Cocos2D-X. Unity permet le développement pour iOS, Android, Windows, PS3, Xbox360, Wii U et Flash Media Player.

Site Web : <http://www.unity3d.com>

On aime

La simplicité d'utilisation
La comparaison avec les autres frameworks

On aime moins

Rien

BButton, GMGridView, MBProgressHUD, etc.

Encore une fois, il existe une multitude de frameworks avantageux. Ne pouvant tous les citer, voici une liste de trois composants intéressants :

- BButton - Bouton mimant le style des bouton de Twitter Bootstrap (<https://github.com/mattlawer/BButton>)
- GMGridView - Grille de contenu (inexistant avant iOS6) (<https://github.com/gmoledina/GMGridView>)
- MBProgressHUD - Pop-up de progression personnalisable (<https://github.com/jdg/MBProgressHUD>)

> Les frameworks réseau

Apple fournit un ensemble de composants dédiés au développement réseau (NSURLConnection, CFNetwork.framework, etc.), cependant l'utilisation d'un framework tiers permet de combler quelques lacunes ou de simplifier une implémentation.

AFNetworking

Ce framework est l'un des plus utilisés actuellement par les développeurs iOS. Sa simplicité d'utilisation et le large soutien qu'il reçoit de la communauté open source sont des critères de choix essentiels pour tous les développements nécessitant l'utilisation de connexions réseaux.

Parmi ses fonctionnalités, voici celles qu'il faut absolument retenir :

- Entièrement asynchrone et distribué (via l'utilisation de NSOperation et NSOperationQueue)
- Codage/Décodage automatique de plusieurs formats : JSON, XML, pList, Image, etc.
- Utilisation des blocks Objective-C

Sites Web : <https://github.com/AFNetworking/AFNetworking>

<http://afnetworking.com>

On aime

La documentation du framework
La programmation par « block »

On aime moins

L'absence de logger intégré

SocketRocket

Ce framework permet d'implémenter un client WebSocket depuis votre application iOS. Totalement conforme à la norme RFC6455 (norme du WebSocket), ce framework vous permettra de communiquer avec tous vos serveurs WebSocket.

De la même manière que AFNetworking, ce framework est entièrement asynchrone et utilise une file d'attente afin de ne pas être bloquant dans nos applications. Cependant, n'ayant pas la même



On aime

La diversité des plateformes accessibles
(Console, PC, Mac, Web, Mobile)
L'utilisation du C# via Mono

On aime moins

Le prix de la licence

> Où trouver d'autres frameworks ?

Le web regorge d'espaces où trouver des frameworks intéressants. Les deux principaux sites web qu'il vous faut connaître pour cela sont :

- GitHub - Plateforme de source-control, basée sur Git et orientée réseau social, qui est majoritairement utilisée par les développeurs pour mettre en ligne leurs frameworks.
- CocoaControls - Bibliothèque de frameworks dédiée à Cocoa. Elle est principalement contribuée par les développeurs eux-mêmes, leur permettant ainsi de donner plus de visibilité à leurs travaux (<http://www.cocoacontrols.com>)

> Le débogage au quotidien

Toute personne ayant déjà développé une application a été confrontée à une phase de débogage. Certains langages, tels que Java, permettent de déboguer très facilement. Cela est principalement dû au fait que le compilateur Java transforme le code en byte code, qui lui-même est interprété par une machine virtuelle dédiée. Ceci permet d'avoir des traces complètes et facilement interprétables. Objective-C étant un langage héritant complètement du C, il n'a pas de mécanisme similaire, ne simplifiant pas toujours le débogage d'une application. Nous allons voir quelques outils intéressants pour faciliter cette étape.

> Un BugTracker automatisé

Appelé aussi Crash Recover, ce genre d'outil, directement intégré dans le code, permet de récupérer un maximum d'informations sur un crash survenu dans l'application lorsque celle-ci est utilisée dans un environnement de production. Plusieurs services permettent la centralisation sur internet : BugSense, TestFlight, Crittercism, HockeyApp. Chacun de ces services a ses points forts et ses points faibles. Le critère de choix dépendra principalement du goût du développeur et du prix que celui-ci est prêt à mettre. Si un seul devait être retenu, ce serait TestFlight qui, en plus de gérer les crashes, permet la livraison de version bêta. Il n'est cependant compatible qu'avec iOS.

> Un logger complémentaire

Apple fournit un outil permettant d'afficher des logs lors d'une utilisation en mode debug. Il est parfois intéressant de pouvoir logger une application lorsque celle-ci est publiée en release, notamment dans les phases de recette où il est compliqué de faire l'ensemble des tests avec un client. Voici une liste non exhaustive permettant cela :

- TestFlight - fournit un logger distant en plus d'un analyseur de crash
- LumberJack - réécriture d'une API de log plus flexible, simple et puissante

La plupart des sociétés redéveloppent leurs propres méthodes de log basées sur la méthode fournie par Apple (NSLog) sans savoir qu'il existe des frameworks plus performants pour cela.

> Quelques outils intéressants

Apple fournit un outil avec XCode, principalement utilisable pour analyser les performances d'une application, appelé « Instruments ». Celui-ci peut aussi être utilisable pour obtenir plus d'informations sur l'application lorsque cette dernière subit un crash (analyse des threads, dump mémoire, etc.). La toile permet aussi de trouver des petites merveilles. C'est par exemple le cas de DCIntrospect



Fig.4

DCIntrospect

(<https://github.com/domesticcatsoftware/DCIntrospect>) qui permet d'avoir un débogage visuel des composants graphiques [Fig.4].

> Un gestionnaire de dépendance : CocoaPods

Certains d'entre vous auront déjà utilisé ce qu'on appelle des « gestionnaires de dépendances ». En voici certains :

- Maven pour Java
- NPM pour NodeJS
- Aptitude pour GNU/Linux Ubuntu
- Gems pour Ruby
- NuGET pour .Net

Depuis Août 2011, date de la création du dépôt sur GitHub, il existe un projet open source dédié à la mise en place d'un gestionnaire de dépendances adapté au langage Objective-C : CocoaPods.

CocoaPods a deux objectifs principaux :

- Rendre la gestion des Frameworks plus simple sur XCode
- Faciliter la découverte et encourager la maintenabilité des frameworks en fournissant un écosystème adapté.

CocoaPods permet donc à tous les développeurs de mettre en ligne, et de distribuer facilement, l'ensemble des projets qu'ils font. Pour cela, CocoaPods a défini un ensemble de spécifications, facilement intégrables à un projet sous XCode. Ces spécifications définissent un certain nombre de règles ainsi que le format d'un fichier nécessaire à tout pod [1] : le podfile. Ce fichier contient des informations essentielles sur la nature d'un framework : nom, description, version, auteur, etc., ainsi que la liste des frameworks nécessaires au bon fonctionnement du pod.

On aime

La simplification d'intégration des frameworks
L'aspect contributif accessible à tous

On aime moins

rien

L'ensemble des outils présentés dans cet article vous permettra d'aborder un projet iOS sereinement. Le développement mobile étant en vogue, certains des frameworks présentés ici peuvent devenir obsolètes, ou bien encore un framework plus simple, plus flexible ou plus robuste détrônera celui le plus utilisé à ce jour. Il est donc primordial d'effectuer une veille régulière sur les technologies mobiles et sur les frameworks existants. N'hésitez donc pas à consulter la toile très souvent : de nombreux outils apparaissent tous les jours et de nouveaux frameworks sont développés régulièrement.

[1] : Un 'pod' représente un projet, un framework ou tout autre code source mis en ligne sur cocoapods

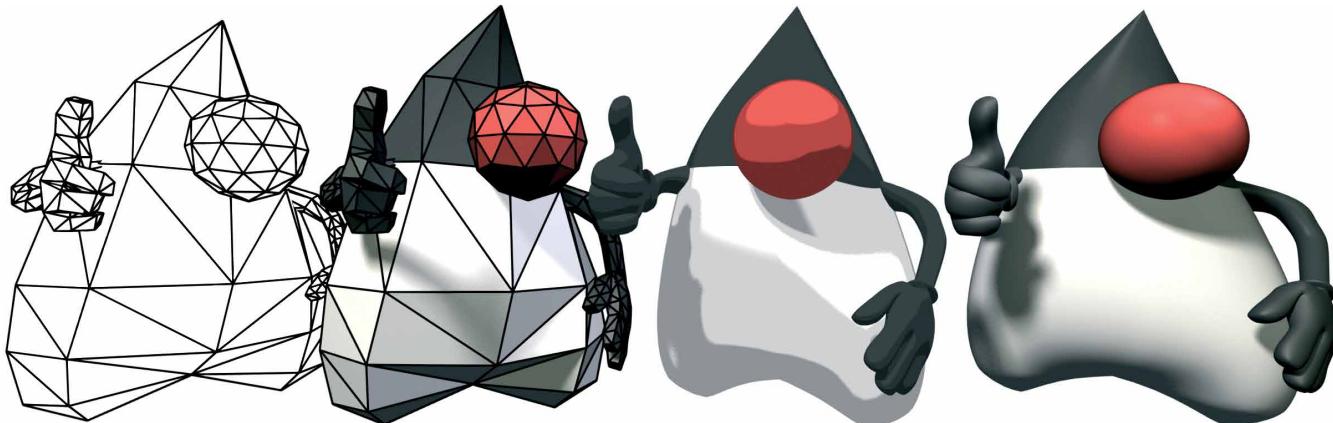
Vincent Saluzzo

Ingénieur développement Logiciel
« Expert en développement iOS »



IDE Java : il y a du monde !

IDE ou EDI en français, est un environnement de développement intégré : un logiciel, qui contient un ensemble d'outils pour développer des logiciels dans différents langages informatiques, comme Java, C++ ou Visual Basic. Un IDE permet aux développeurs de travailler plusieurs aspects de logiciels, sans quitter cet environnement.



Un IDE est souvent graphique (notamment avec des designers d'interface) et « interactif ». Il est loin le temps où les développeurs utilisaient un éditeur de texte pour écrire le code et un compilateur pour le transformer en programme exécutable. L'édition de programmes dans un IDE, est souvent assistée (coloration syntaxique, complétion automatique, accès aux documentations, indentation automatique, correction d'erreurs, templates...). La compilation passe d'une manière transparente pour le développeur. Un IDE contient également un débogueur, un interpréteur pour les langages interprétés, souvent un gestionnaire de versions. Il est à noter qu'un IDE peut accepter des modules additionnels (plug-in) pour étendre ses fonctionnalités initiales. Les plug-ins de tests unitaires, loguer des erreurs, la modélisation UML et la génération automatique de codes, sont parmi les exemples de plug-ins si l'IDE n'est pas doté out-of-the-box de ces outils.

Il existe des IDE pour la majorité des langages de programmation. Le domaine étant très vaste, cet article se concentre sur le langage Java et les trois IDE les plus utilisés dans deux domaines le Web (Google App Engine pour Java, JSF2) et les mobiles sous Android : Eclipse, IntelliJ IDEA, et NetBeans. Ils sont disponibles sur Linux, OS X et Windows.

Configuration utilisée

Système	Fedora 17 64 bits
SDK	Android SDK "android-sdk_r21.0.1" Google App Engine SDK for Java 1.7.4
Autres librairies	Java SE 7u13 (JDK) MyFaces Core 2.1.8
Versions des IDE	Eclipse Juno SR1 IntelliJ IDEA 12 NetBeans IDE 7.3 rc1

JAVA

> Installation de JDK

1. Téléchargez le fichier archive sur le site d'Oracle

<http://www.oracle.com/technetwork/java/index.html>

2. Décompressez

3. Ajoutez une variable JAVA_HOME et ajoutez son sous-dossier "bin" au PATH dans le fichier .bachrc.

GOOGLE APP ENGINE POUR JAVA

Google App Engine est une plateforme Cloud pour des applications Web développées en Java, Python et Go. Pour l'utiliser inscrivez-vous sur la plateforme Google. Il faut disposer d'un plug-in compatible et le SDK.

> Installation du SDK

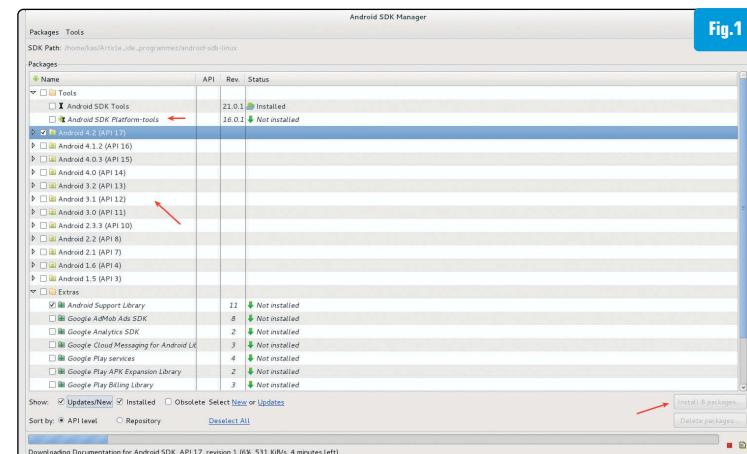
- Téléchargez sur <https://developers.google.com/appengine/downloads>
- Décompressez

ANDROID

Nous prenons Android comme exemple pour le développement Mobile. On doit avoir un plug-in et le SDK d'Android.

> Installation du SDK :

- Téléchargez à l'URL suivante <http://developer.android.com/sdk/index.html> le fichier ou l'installateur correspondant à votre OS.
- Pour Linux décompresser le fichier .tgz donne le dossier "android-sdk-linux"





Note : Le SDK d'Android est disponible uniquement pour Linux 32 bits. Il faut pour cette raison installer quelques bibliothèques 32 bits sur des distributions Linux (64 bits)

Ajoutez des plateformes

Le SDK installé (Android SDK Tools) a besoin des plateformes et de "platform tools". Pour cela, exécutez le fichier android-sdk-linux/tools/android

Cochez le/les plateformes désirées et Android SDK Platform-tools. Un clic sur le bouton "Install xxx packages".

Configuration des AVD

Exécutez le fichier (avec l'action avd) android-sdk-linux/tools/android avd pour créer un ou plusieurs périphériques virtuels [Fig.1].

ECLIPSE

Considéré par un grand nombre de développeurs comme l'IDE de référence pour Java/Java EE. Composé d'un noyau autour duquel viennent se greffer des plug-ins pour lui donner des fonctionnalités supplémentaires (plusieurs langages et toutes sortes de technologies). Le SDK d'Eclipse contient Eclipse Java development tools (JDT), offrant à Eclipse son compilateur incrémental. Eclipse est un programme FOSS diffusé par Eclipse Foundation.

Eclipse est un IDE complet prenant en charge la gestion des projets complexes. L'édition de fichiers sources est assistée par de nombreux outils puissants. L'exécution en local, le debugging et le remote debugging sont assurés, ainsi que le déploiement vers la majorité des serveurs. Scrapbook constitue un avantage majeur et permet de tester les codes à la volée. Eclipse est distribué sous la forme de bundles, chacun est composé d'un ensemble d'outils pour développer en un langage donné ou pour des tâches précises, par exemple Eclipse IDE for Java EE Developers.

> Installation :

- Téléchargez à l'adresse <http://www.eclipse.org/downloads/> une version pour votre OS
- Décompressez
- Exécutez le fichier eclipse dans le dossier de l'installation [Fig.2].

> Intégration avec Google App Engine

Google Plugin for Eclipse contient tous les outils pour développer, tester, et déployer une application vers la plateforme depuis l'IDE.

Installer Google Plugin for Eclipse

le menu Help > Install New Software... > un clic sur le bouton ADD > donnez un nom et une URL

<https://dl.google.com/eclipse/plugin/4.2> > OK > sélectionnez parmi les com-

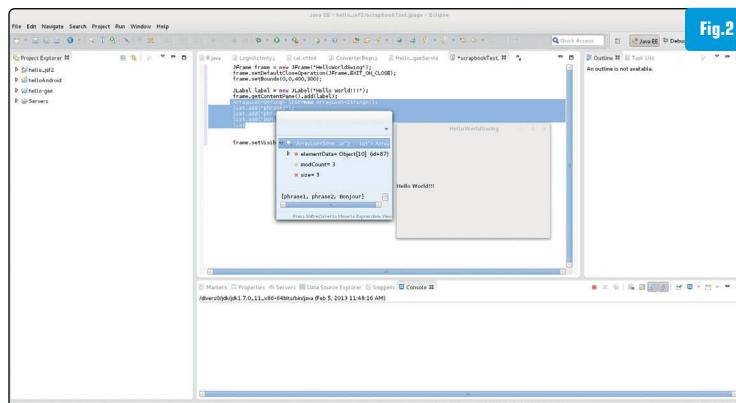


Fig.2

posants affichés > vérifiez et validez la licence > Redémarrez Eclipse.

Créer un projet GAEJ

Une flèche à côté de l'icône de Google dans Eclipse ouvre une liste d'actions sur Google App Engine > New Web Application Project > donnez un nom au projet et le package pour les classes > Dé-sélectionnez Use Google Web Toolkit > Finish pour créer le projet.

Pour exécuter le projet en local

Un clic droit sur le projet > Run As > Web Application.

Pour déboguer

Un clic droit sur le projet > Debug As > Web Application.

Pour déployer

Un clic sur la flèche à côté de l'icône de Google dans Eclipse > Deploy to App Engine [Fig.3].

> Intégration des frameworks Web

Les plug-ins d'Eclipse rendent les frameworks Web et autres disponibles sur cette plateforme. Le bundle Eclipse IDE for Java EE Developers, dispose de tous les outils nécessaires pour la gestion des projets Web, mais pour développer et tester des projets JSF2 vous avez besoin d'une implémentation JSF2 (ajoutée comme fonctionnalité ou nouveau projet) et d'un serveur (Tomcat par exemple).

Implémentation JSF2

On va utiliser l'implémentation de MyFaces2 (Apache), pour créer des projets JSF2

Le menu Window > Preferences > Java > Build Path > User Libraries > New > donnez un nom à la library > Add External JAR's > allez dans le dossier de décompression MyFaces2/lib > sélectionnez tout sauf myfaces-bundle.x.y.z.jar.

Les fichiers de configuration sont créés automatiquement. l'IDE se charge de compléter, et générer le code. On regrette cependant l'absence de la détection et la correction automatique pour les fichiers de facelets (*.jsf).

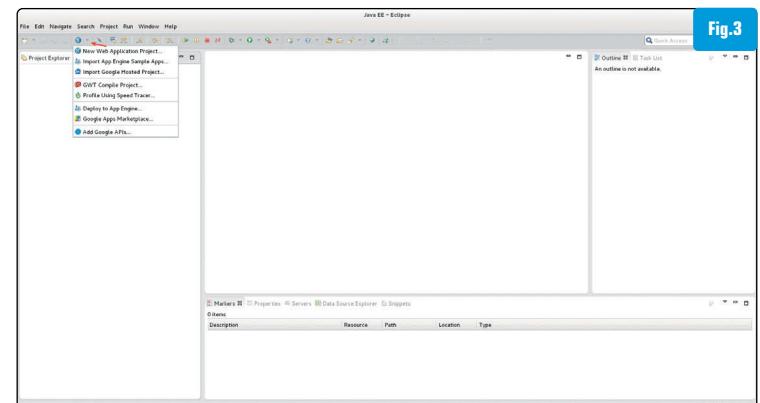
> Développement pour Android

ADT (Android Development Tools) est un plug-in pour Eclipse. Il permet de créer des projets Android complexes. Doté d'un UI Builder très intuitif pour créer les composants graphiques. Le plug-in permet de tester et déboguer des applications Android. La fenêtre Outline est fort utile. Malheureusement les fonctions de détection, correction d'erreurs de l'éditeur XML ne sont pas suffisantes.

Installer ADT

Le menu Help > Install New Software... > un clic sur le bouton ADD > donnez un nom et une URL

<https://dl-ssl.google.com/android/eclipse/> > OK > sélectionnez Developer



Outils Google App Engine



Tools > vérifiez et validez la licence > Redémarrez Eclipse. Normalement au démarrage un écran vous demande l'emplacement du SDK d'Android.

Créer un projet Android

File > New > Other > Android > Android Application Project > remplissez le nom, package et sélectionnez le SDK à utiliser > cochez create activity > configuez le launcher > sélectionnez le style d'activity > donnez un nom à l'activity et au layout > Finish.

Tester l'application

Si ce n'est pas déjà fait, commencez par configurer le AVD d'Android (Android Virtual Device), par aller dans le menu Run > Run Configurations. Ceci appelle le gestionnaire de AVD (dans le SDK d'Android) Clic droit sur le projet > Run As > Android Application [Fig.4].

INTELLIJ IDEA 12

Il a (spécialement Ultimate Edition) tous les outils pour développer pour le Web, les entreprises et les mobiles. Il dispose des fonctions très avancées pour l'édition de codes source. Des outils de production pour les bases de données, UML, contrôle de versions, de déploiement vers le Cloud. Des frameworks supportant le développement des entreprises et le Web. Enfin le UI Designer pour Android est disponible aussi dans la Community Edition. L'IDE est doté d'un système de plug-ins.

On regrette que dans la Community Edition, il manque beaucoup de fonctions intéressantes pour développer en Java, par exemple le support de JSP, JSF et Servlets. La gestion des projets est nettement plus difficile que sur Eclipse et NetBeans. Il n'est pas logique d'ouvrir un projet pour le supprimer ou d'aller supprimer les dossiers de projets !!! Deux versions sont disponibles Community Edition et Ultimate Edition.

	Community Edition	Ultimate Edition
Licence	Gratuite Open Source	Payante
Langages	Principalement Java SE, Groovy, XML, Android	En plus de ceux de la Community Edition, JEE, Python, PHP, Ruby, HTML, CSS, Javascript, etc.
Frameworks	Non. Quelques frameworks sont installables comme plug-ins	La plupart des frameworks des langages supportés et notamment Spring, Hibernate et JSF2
JUnit, TestNG	Oui	Oui
Serveurs	Non.	Tomcat, GlassFish, JBoss, WebLogic, WebSphere, Geronimo, Resin, Jetty.
VCS	Moins que L'Ultimate	Complet
UI Designer pour Android	Oui	Oui

> Installation :

L'installation est facile, voici comment l'installer sur Linux

1. Téléchargez une des versions disponibles

<http://www.jetbrains.com/idea/download/index.html>

2. Décompressez

3. Exécutez "idea.sh" dans le dossier "bin"

> Intégration avec Google App Engine

IntelliJ IDEA (Ultimate Edition) dispose de son propre plug-in installé pour développer pour le Cloud, sur Google App Engine en Java. Il suffit d'installer le SDK de Google (téléchargez et décompressez), à l'URL suivante <https://developers.google.com/appengine/downloads>.

l'IDE permet de créer des projets, avoir de l'assistance pendant le codage, tester et déboguer une application sur le serveur de développement de Google et le déploiement via l'interface graphique.

Parmi les assistances de codage, un analyseur à la volée puissant. Il détecte même les violations de codage imposées par Google App Engine. Un analyseur à la demande permet d'avoir des analyses très détaillées. Malgré des assistants performants, au niveau de l'intégration ainsi que le codage, quelques ajustements ont été nécessaires pour que tout fonctionne normalement.

Créer un projet GAEJ

Create New Project (ou File > New Project) > JavaEE Web Module > Google App Engine > Sélectionnez le dossier d'installation du SDK Google App Engine et la persistance désirée > Finish.

Pour exécuter le projet en local

Le menu Run > Edit Configurations (pour configurer le serveur). Puis Run 'nom_serveur'

Pour déboguer

Le menu Run > Debug

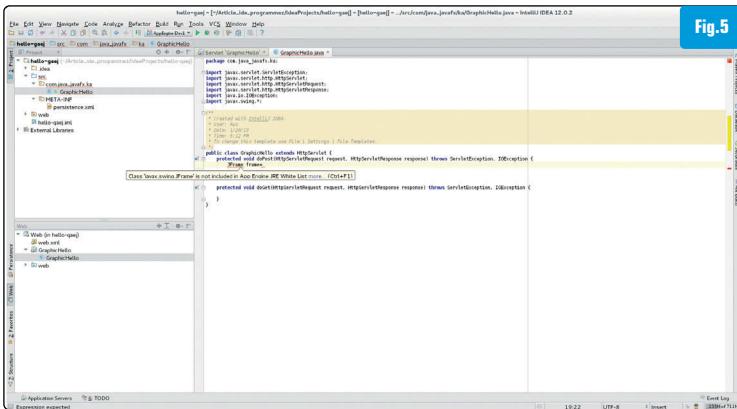


Fig.5

Analyseur à la volée

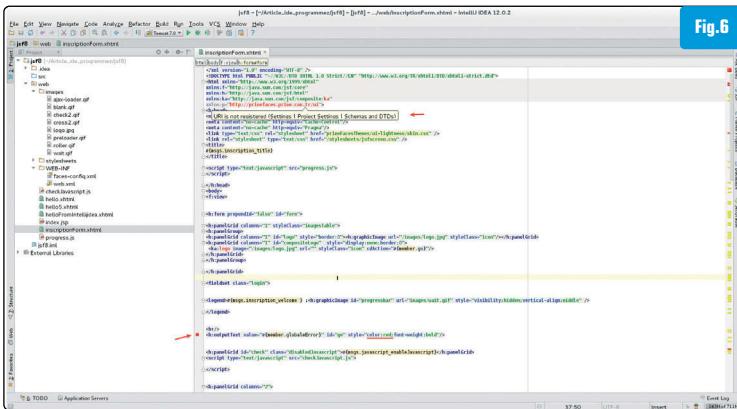
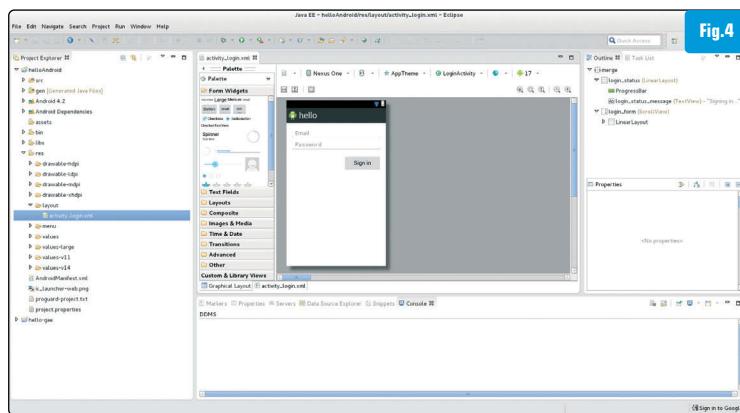


Fig.6



Détectio automatique d'erreurs dans une facelets



Pour déployer

Tools > Upload App Engine Application [Fig.5].

> Intégration des frameworks Web

L'Ultimate Edition de l'IDE dispose d'assistant pour développer sans difficulté en utilisant des frameworks tels que Struts 2, Spring ou JSF2. Il est possible d'ajouter des fonctionnalités JSF2 à un projet Web existant ou de créer un nouveau projet Java EE, avec le support JSF2. Il suffit d'indiquer au moment de la création d'un projet, l'implantation JSF2 (dossier de décompression MyFaces2/lib > sélectionnez tout sauf myfaces-bundle.x.y.z.jar) et le serveur à utiliser. Les fichiers de configuration sont créés automatiquement.

Au moment du codage, l'IDE se charge de compléter, rectifier, et générer le code [Fig.6].

> Développement pour les mobiles

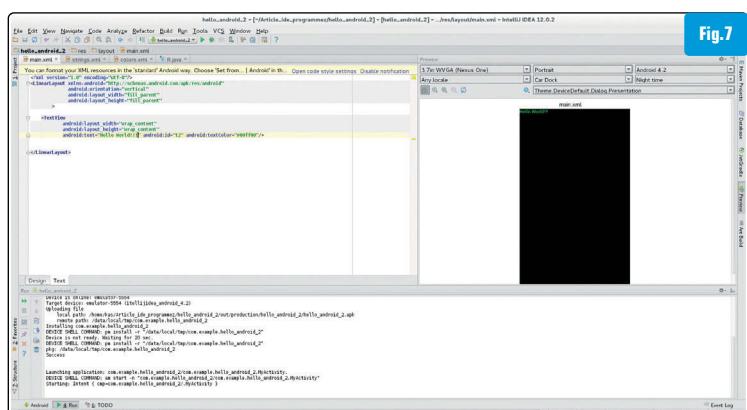
Les outils d'IntelliJ IDEA pour développer des applications Android sont très performants, y compris les assistants de codage, la possibilité de tester l'application sur un émulateur, le déploiement et un designer graphique, très intuitif pour les composants graphiques (UI Designer). La fenêtre de Preview reflète en temps réel le changement dans le code (même sans sauvegarder).

Créer un projet Android

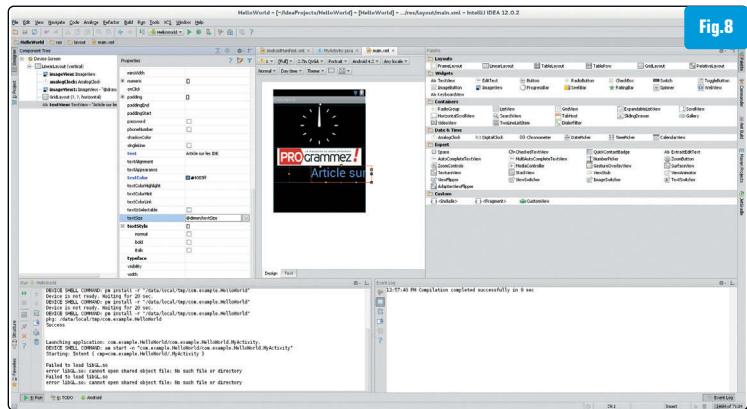
Create New Project > Application Module (Android) > Sélectionnez le dossier d'installation du SDK d'Android > sélectionnez la plateforme désirée et quelques options et le tour est joué.

Tester l'application

Si ce n'est pas déjà fait, commencer par configurer le AVD d'Android (Android Virtual Device), par aller dans le menu Run > Edit Configurations. Ceci appelle le AVD Manger [Fig.7 et 8].



Live demo Android (la fenêtre Preview)



UI Designer pour Android

NETBEANS

Est un programme FOSS. Il permet de développer des applications Desktop, Web et Mobile. La version 7.2 intègre les dernières technologies Web ; HTML5, JavaScript, et CSS. Ses nouveaux éditeur et débogueur Javascript sont simplement magnifiques. NetBeans est multi-langage Java, C/C++, PHP. Ses plug-ins permettent d'ajouter énormément de fonctionnalités. Il est à noter que NetBeans est supérieur aux Eclipse et IntelliJ IDEA au niveau de détection et correction automatique pour les pages facelets, Javascript et CSS. Eclipse est plus fort pour afficher la structure CSS. Plusieurs bundles sont disponibles, optez pour "All" ou Java EE bundle.

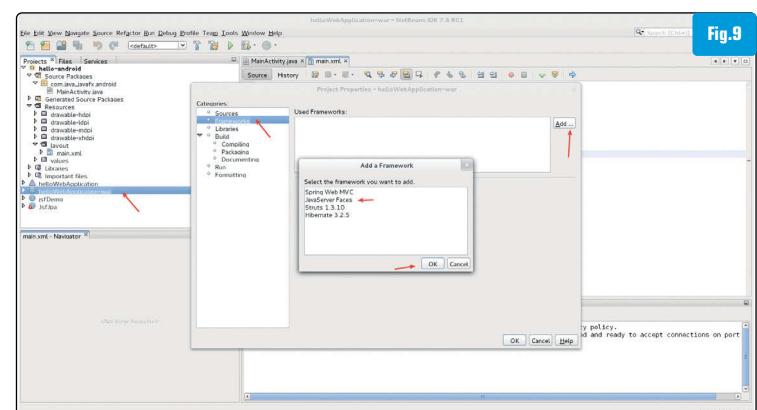
Installer NetBeans. Téléchargez un seul fichier sur le site <http://netbeans.org/downloads/>. Sur Linux, ouvrez un terminal. Tapez les commandes suivies de ENTER :

1. rendez le fichier exécutable
chmod u+x netbeans-x.y.z-ml-linux.sh
2. Lancez l'installation
. /netbeans-x.y.z-ml-linux.sh
3. L'installation ajoute une icône sur le bureau, pour démarrer NetBeans.
4. Pour supprimer NetBeans exécutez uninstall.sh dans le dossier de l'installation.

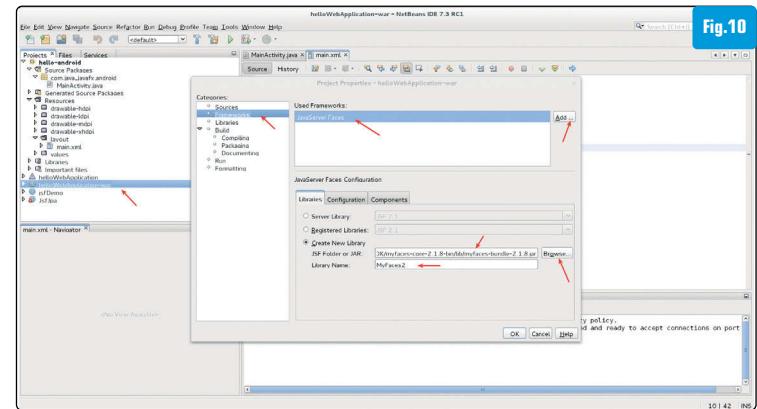
> Intégration avec Google App Engine

Un plug-in pour NetBeans est disponible sur <http://kenai.com/projects/nbappengine/pages/Home>

Son problème majeur, un décalage important avec les versions de NetBeans. Il y a des plug-ins officiellement pour NetBeans 6.8 et 6.9, mais sûrement pas pour la version 7.2 et supérieure. Il est



Frameworks pour le Web



Possibilité d'utiliser MyFaces2 ou Mojarra 2



aussi possible de bricoler pour faire tourner Google App Engine SDK sur NetBeans !!!

> Implémentation JSF2

Le Java EE bundle de NetBeans contient une implémentation JSF2 (Mojarra 2.0/2.1) et le serveur GlassFish 3. Donc rien à ajouter, mais on peut si on le souhaite utiliser l'implémentation JSF2 de MyFaces2.

Dans tous les cas le descripteur du projet est créé et configuré automatiquement selon l'implémentation JSF2 utilisée.

Les assistants pour générer le code des beans, facelets sont excellents. Les tags namespaces JSF, sont ajoutés d'une manière intelligente dès qu'on entre une balise.

On regrette ici que modifier la configuration ou supprimer un framework déjà ajouté, ne soit pas une tâche facile.

Ajouter le support JSF2 à un projet Web application existant

Clic droit sur un projet > Properties > Frameworks > cliquez sur le bouton Add > sélectionnez un des frameworks > Java Server Faces > OK > laissez le choix Server Library > un clic sur l'onglet Configuration > remplacez "/faces/*" par "*.jsf" > laissez sélectionner "facelets"

Note : On peut aussi créer un nouveau projet JSF2.

Pour utiliser MyFaces2 :

Au lieu de sélectionner "Server Library" > sélectionnez Create New Library > un clic sur Browse pour sélectionner myfaces-bundle.x.y.z.jar dans MyFaces2/lib > donnez un nom, comme par exemple MyFaces2 > OK [Fig.9, 10 et 11]

Développement Android

Des plug-in pour Android sont sur le site <http://www.nbandroid.org>. Il est possible d'installer le plug-in en utilisant l'auto update center ou via un fichier .zip. Peu d'outils sont disponibles pour développer des applications Android, l'assistant d'édition n'a pas de correction automatique. Pas de designer graphique pour les UI.

Installation du plug-in (auto update center) :

Le menu Tools > Plugins > Settings onglet > le bouton Add > donnez un nom à l'emplacement et entrez l'URL suivante : <http://nbandroid.org/release72/updates/updates.xml> > un clic sur l'onglet Available Plugins > tapez android dans la zone de recherche > cochez le nom du plug-in désiré. > le bouton Install > validez la licence > redémarrez NetBeans

Le chemin vers le SDK d'Android

Le menu Tools > Options > Miscellaneous. Onglet Android > sélectionnez l'emplacement de l'Android SDK

Créer un projet Android

Le menu File > New Project > Android > Android

Dans la fenêtre suivante, donnez un nom, package et chemin au projet. Nommez l'activity > Sélectionnez la plateforme Android cible > Finish.

Gestion des AVD

Pour éditer les plateformes le menu Tools > AVD Manager.

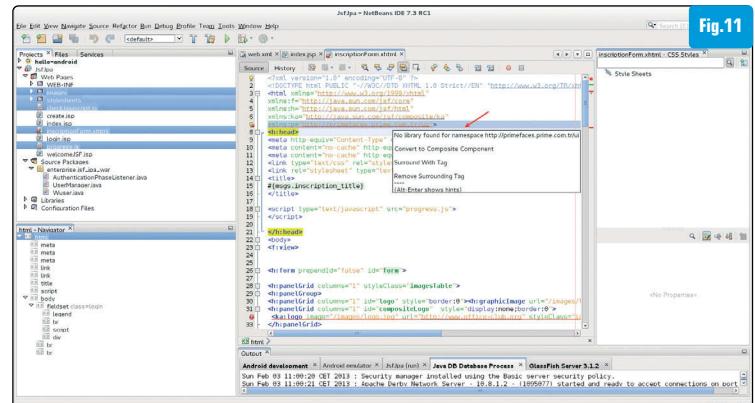
Tester l'application

Un clic sur l'icône play ou le menu Run > Run Main Project ou F6

Kaesar Alnijres

Développeur Java - Leader jug-Cergy

<http://www.java-javafx.com>



Détection automatique d'erreur dans une facelet

CONCLUSION

Et maintenant quel IDE choisir ? Si vous développez pour de multiples domaines (Desktop, Web, et mobile), Eclipse s'avère le meilleur compromis. Il présente un IDE complet, mais présente en revanche des faiblesses pour le développement Web. Domaine où s'imposent NetBeans et IntelliJ IDEA Ultimate. Optez pour IntelliJ

IDEA si vous êtes une entreprise, vous aurez en plus le support technique. NetBeans est déconseillé pour le développement avec Google App Engine et Android. Eclipse est le meilleur choix pour Google App Engine. Le plug-in est développé sous la direction de Google.

IDE	Pour	Contre
Eclipse	<ul style="list-style-type: none"> IDE de référence dans plusieurs domaines Soutenu par les ténors du marché de l'informatique FOSS, léger, interface native Une communauté très large 	<ul style="list-style-type: none"> Les assistants pour le développement Web ne sont pas à la hauteur, spécialement au moment du codage
IntelliJ IDEA	<ul style="list-style-type: none"> Multiples langages Support de la plupart des frameworks Fonctions très avancées pour l'édition 	<ul style="list-style-type: none"> La Community Edition est limitée, spécialement au niveau Web et frameworks La gestion de projets est un vrai casse-tête.
NetBeans	<ul style="list-style-type: none"> Editeur et débugger JavaScript Intègre les dernières technologies Web FOSS 	<ul style="list-style-type: none"> Le plug-in pour Google App Engine est obsolète Peu d'assistants pour les mobiles Pas de designer pour Android



Passage en revue de la boîte à outils idéale du développeur Android

La qualité d'une technologie ou d'une plateforme n'est pas un gage suffisant de réussite. En effet, le succès d'une plateforme passe également par la qualité des outils mis à disposition des développeurs facilitant leur travail d'appropriation. Le but étant dans un système comme Android de proposer une offre de contenu toujours plus grande aux utilisateurs et d'assoir ainsi la technologie. Du fait de l'ouverture d'Android, le nombre d'outils augmentant la productivité des développements est pléthorique et il devient même difficile de se constituer sa boîte à outils idéale devant un tel choix. Afin de faciliter le travail des développeurs Android, cet article propose un inventaire de la boîte à outils idéale.

> Compilateur

Au niveau du compilateur, il n'y a pas réellement de choix à faire puisque les sources d'applications Android doivent obligatoirement être compilées avec le JDK officiel. La version 6 de ce dernier est recommandée mais la dernière mouture est également supportée à condition de positionner le niveau de compliance à Java 5 ou 6 lors de la compilation. En revanche, le compilateur libre GCJ (GNU Compiler for Java) n'est pas supporté et ne peut donc être utilisé.

> IDE

Le choix en matière d'environnement de développement est fourni sur Android. Solution officielle développée par Google, les outils ADT (Android Development Tools) sont proposés sous la forme de plugins Eclipse. En perpétuelle amélioration, ils permettent de gérer un projet de sa création à son packaging sous forme d'APK en passant par le débogage lors du test d'application sur émulateur ou terminal physique. Actuellement en version 21.0.1, ADT s'utilise avec un Eclipse 3.6.2 au minimum. En outre, il propose au développeur un grand nombre d'outils du SDK Android, packagés sous forme de vues Eclipse avec la perspective DDMS (Dalvik Debug Monitor Service) regroupant plusieurs de ces vues. On soulignera également la possibilité de gérer des terminaux virtuels pour tester le rendu d'une application sur différents supports, la vue affichant les logs d'un terminal, les vues permettant d'inspecter les threads et les allocations mémoires ainsi qu'un explorateur de fichiers d'un terminal. Bénéficiant des dernières avancées du SDK, la solution

Eclipse poussée par Google est la référence en dépit de critiques sur les lenteurs de l'IDE. La principale lacune d'ADT concerne l'éditeur de création d'IHM, loin d'être à la hauteur de ce que propose Apple pour l'iPhone par exemple. En complément à ADT, le plugin Eclipse MotoDev est proposé par Motorola. Il facilite la gestion des bases des données SQLite d'un terminal et propose des snippets de code accélérant l'écriture de codes récurrents. Alternative numéro 1 d'Eclipse et ADT, IntelliJ IDEA 12 propose une version Community gratuite avec un plugin Android complet. A l'image d'ADT, ce plugin propose un ensemble d'outils pour gérer un projet avec un éditeur graphique un peu plus abouti. Néanmoins, les outils de monitoring proposés restent moins nombreux qu'avec ADT. Enfin, un mot concernant NetBeans pour lequel un plugin indépendant est proposé. Ce dernier est plutôt famélique, se contentant du support des actions essentielles réalisées sur un projet (création, debug, build et packaging).

> Emulateur

Le SDK fournit en standard un émulateur de terminaux basé sur le logiciel libre QEMU. Il donne la possibilité au développeur de créer des terminaux virtuels en spécifiant finement leurs caractéristiques matérielles (taille d'écran, résolution, densité de pixels, présence d'un GPS, ...) et logicielles (version d'Android). Cependant, cet émulateur souffre de certains problèmes de performances empêchant dans certains cas son utilisation pour tester des tablettes 10 pouces par exemple. Pour adresser cette problématique, il est possible de télécharger

directement une image ISO d'Android et de l'installer sur un logiciel de virtualisation tel que VirtualBox afin d'émuler un terminal de manière plus puissante et réactive. En ce sens, une solution comme AndroidVM s'avère intéressante car elle propose des images VirtualBox préconfigurées pour certaines versions d'Android.

> Composants

Du fait de son ouverture, la plateforme bénéficie d'un nombre important de composants prêts à l'emploi. S'il ne fallait avoir qu'une seule bibliothèque de composants dans sa boîte à outils, il s'agirait de l'Android Compatibility Library de Google. Celle-ci permet la compatibilité d'une application utilisant des concepts apparus dans la version 4 du SDK avec les versions antérieures de l'OS. Le support pour les fragments et les loaders se révélant essentiel. Pour le reste, les composants peuvent être trouvés via de multiples sources sur internet que ce soit via GitHub ou Google Code ou bien en allant sur des sites les référençant comme <http://www.theultimatemobilelibrary.com/> et <http://www.androidviews.net/>. Le tableau (voir tableau A) présente les composants les plus populaires du moment et permettant de gagner un temps précieux lors du développement d'applications.

> Frameworks

Plus structurants que les composants, les frameworks facilitent le travail du développeur dans un domaine précis. De par son architecture, Android nécessite l'écriture d'un grand nombre de lignes de code d'infrastructure, par exemple, et ne permet pas de réaliser simplement des tâches récurrentes



telles que l'injection de dépendances ou la gestion de bases de données SQLite via un mapping de type objet / relationnel. Les frameworks listés dans le tableau (voir **tableau B**) adressent ces différentes problématiques augmentant grandement la productivité. La plupart de ces frameworks peuvent être mis en œuvre conjointement permettant de se construire une stack complète, similaire à ce que peut connaître le développeur Java côté serveur. Ainsi, le couplage d'Android Annotations avec Dagger pour l'injection de dépendances et Otto pour la gestion d'un bus d'évènements permet d'obtenir une application avec un code boilerplate quasiment inexistant et un découplage de ses différentes parties.

> Tests

Phase essentielle du cycle de vie d'une application, le test n'est pas oublié sur Android avec, au sein du SDK, un framework entier

prévu à cet effet s'appuyant sur la référence Java en la matière, JUnit. ADT permet de créer directement au sein d'Eclipse un projet de test pour une application donnée et le SDK fournit ensuite des classes de tests telles que `AndroidTestCase` et `ActivityUnitTestCase` facilitant l'écriture des tests pour Android. Ces derniers étant instrumentalisés par Android, ils sont de fait très lents. Pour des tests unitaires qui par nature doivent s'exécuter aussi rapidement que possible, il est donc préférable d'utiliser la bibliothèque Roboelectric qui permet de lancer les tests sans avoir besoin de lancer l'application. Ce framework se charge de mockier les appels aux classes contenues dans le SDK Android. L'utilisation de Roboelectric conjointe au classique Mockito du monde Java permettant même d'envisager la mise en place du TDD pour le développement d'applications Android ! Enfin, pour des tests d'intégration avec la possibilité de

jouer des scénarios d'utilisation poussés niveau IHM, cela pourra être réalisé à l'aide de Robotium. Ce framework complète les outils officiels fournis par le SDK et oblige à coder les scénarios directement au sein de classes de tests. L'exécution pouvant ensuite se faire sur un émulateur ou un terminal physique.

> Divers

Au rayon divers, on citera l'outil d'analyse de code statique `lint` fourni avec le SDK, permettant à la manière d'un Checkstyle d'obtenir des warnings sur le code d'applications Android, via la définition d'un ensemble de règles et de vérifications à effectuer. Se présentant sous la forme d'un simple JAR, le petit utilitaire `Droid@Screen` est incontournable pour réaliser simplement et gratuitement des vidéos d'utilisation d'une application la mettant en valeur avant sa publication sur le Google Play Store. Destiné à l'exploration de bases de données SQLite, `SQLite Browser` est un complément intéressant aux outils d'ADT puisqu'après export de la base d'une application, il est possible de visualiser les données contenues dans le fichier obtenu. Enfin, un dernier mot sur la suite d'outils en ligne `Android Asset Studio`, fourni par Google, qui fournit des outils de génération graphique pour les applications Android. Elle permet la génération des différents types d'icônes d'une application en respectant les formats cibles des différentes densités d'écran mais également la génération de styles pour les barres d'action, par exemple.

Composant	Description	Tableau A
AChartEngine	Permet de réaliser un grand nombre de graphiques de différents types (lignes, barres, régions, camembert...).	
ActionBarSherlock	Composant étendant la bibliothèque de compatibilité Android facilitant l'utilisation du design pattern UI ActionBar à travers toutes les versions d'Android.	
ACRA	Composant permettant d'obtenir un reporting précis des plantages et bugs d'une application et s'interfaisant avec différents backends (Google SpreadSheet, ...).	
Admob / AdWhirl	Monétisation d'une application via le réseau publicitaire Admob ou en utilisant la médiation via AdWhirl.	
Flurry	Solution aidant le développeur à obtenir des données statistiques précises sur l'utilisation d'une application.	
Pull To Refresh	Fournit un composant réutilisable de Pull to Refresh comparable à ce qui se fait sur iOS. Utilisable sur plusieurs types de vue : ListView, GridView, WebView, ViewPager, HorizontalScrollView, ...	
SlidingMenu	Mise en place d'un menu glissant depuis un côté de l'écran, à l'image de ce qui existe sur l'application Google+.	
ViewPagerIndicator	Composant complétant le ViewPager Android standard en fournissant un indicateur de positions à l'image de ce qui se fait sur l'application Google Play Store.	
Zxing	Composant de génération et de lecture de code-barres 1D et 2D.	

Framework	Description	Tableau B
Android Annotations	Augmente la productivité des développements en générant le code boilerplate à la compilation via l'APT.	
Dagger	Framework d'injection de dépendances rapide et moderne fonctionnant aussi bien en Java que sur Android et sa JVM Dalvik spécifique au mobile.	
GreenDroid	Facilite le développement d'interfaces en proposant un ensemble de classes utilitaires.	
Green DAO	ORM facilitant la manipulation de bases de données SQLite. Basé sur la génération de code réalisée séparément du processus de compilation.	
Jackson	Bibliothèque de manipulation JSON légère et ultra performante qui se trouve particulièrement adaptée à l'univers Android.	
libGDX	Moteur de jeu 2D / 3D léger, facile à prendre en main et particulièrement bien documenté.	
OrmLite	Framework ORM Java léger proposant un support spécifique pour les appels natifs à la base de données SQLite présente sur les appareils Android.	
Otto	Bus d'évènements permettant de découpler les différentes parties d'une application Android. Fork du bus d'évènements de Guava, spécialement adapté pour les contraintes de l'univers mobile.	
Polaris	Framework facilitant l'utilisation de Google Maps au sein d'une application.	
Retrofit	Client REST basé sur des annotations comme le RestTemplate de Spring Android mais beaucoup plus léger (à peine plus de 50 Ko).	
RoboGuice	Framework d'injection de dépendances basé sur le célèbre Google Guice en version 3.0 et adapté pour l'univers constraint d'Android.	



Les outils complémentaires et utiles

Dans un développement, il n'y a pas que l'IDE qui soit important. Vous avez à votre disposition de nombreux outils et plugins qui peuvent s'associer à votre projet pour vous faciliter la vie et surtout vous aider à le maintenir.

Ces outils sont très variés et il n'est pas possible de tous les référencier. Nous avons donc volontairement ciblé notre propos sur :

- Manipulation des données
- Débogage
- Contrôle des standards
- Les tests
- Optimisation

> Manipuler les données

Quel que soit votre projet et le langage utilisé, il est nécessaire de modéliser vos données, bref de réaliser une représentation graphique de celles-ci, pour assurer une meilleure qualité de votre projet (et des données).

SQL Power Architect

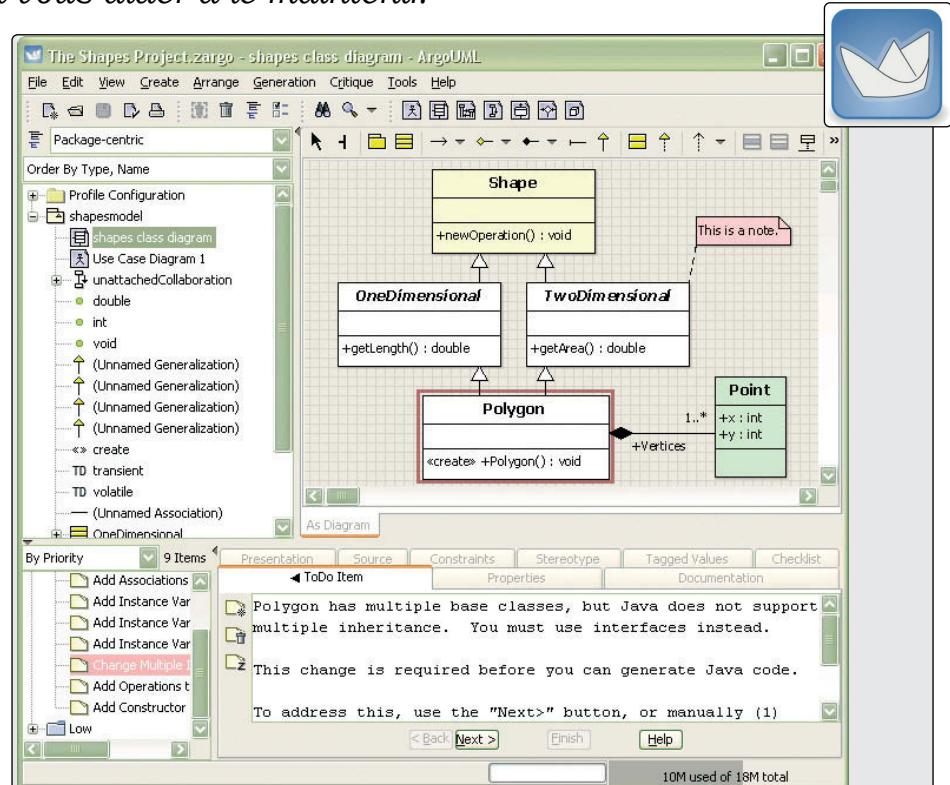
Power Architect est un outil de modélisation de données, c'est à dire un ETL, sous licence GPL 3. Il permet d'améliorer vos bases de données en augmentant l'utilisation des ressources. Ce logiciel est compatible avec l'ensemble des bases de données du marché : MySQL, SQL Server, Oracle, PostgreSQL, etc. Il va vous permettre d'exécuter des tâches complexes en un minimum de temps. Par ailleurs, Power Architect permet de capturer directement la structure de votre base de données et de la mettre à jour en temps réel. Bien sûr, vous pouvez dessiner un modèle de données en drag and drop, qui sera utilisable ultérieurement avec un mode hors connexion sous la forme de clichés des structures de la



base de données. Par ailleurs, vous pouvez vous connecter à plusieurs bases de données en simultané et exporter différents formats de vues (XML, PDF, JPEG...)

MySQL Workbench

MySQL Workbench est un outil graphique destiné aux architectes de bases de données. Il est optimisé pour les bases de données.



ées MySQL et MariaDB (et tout fork MySQL)... Son but principal est de vous aider dans la modélisation de données et le développement de code SQL. Par ailleurs, il propose des outils d'administration complets pour la configuration des serveurs, la gestion des utilisateurs

ArgoUML

ArgoUML est un outil de modélisation UML (Unified Modeling Language), sous licence BSD est open source. Son intérêt principal est de vous aider à utiliser les phrases de spécification et de conception, c'est-à-dire la possibilité de décrire des objets et leurs relations. Il est compatible avec l'ensemble des diagrammes UML standards. Il supporte les diagrammes de classe, les diagrammes d'état, les diagrammes « Use Case », de séquence, etc. Par ailleurs, ArgoUML propose de nombreuses extensions permettant de supporter de nombreux langages : Java, C++, PHP, C#, SQL, etc.

Enfin, il permet d'exporter vos réalisations de diagramme en XML, GIF, PNG, PostScript, EPS, PGML et SVG

> Débogage

Le débogage ne doit pas être oublié dans la phase de votre développement, même si les éditeurs de code (IDE) proposent la colorisation syntaxique, cela ne suffit pas pour afficher le contenu de vos pages correctement.

Avec votre IDE

Quel que soit l'IDE que vous utilisez, vous disposez dans les extensions ou plugins d'un débogueur lié à votre langage de développement. Mais vous pouvez l'étendre.

XDebug

Xdebug est une extension du langage PHP, qui est assez méconnu du grand public ou mal utilisé. Il permet d'accélérer et de simplifier efficacement les cycles de débogage d'une application. Vous pouvez l'intégrer dans votre IDE à travers les fichiers de configuration. Ce projet est toujours en activité et continue à évoluer, suivant les différentes versions de PHP. Lorsque vous activez cette extension, vous pouvez l'utiliser comme un débogueur classique que vous trouvez dans n'importe quel logiciel, autrement dit, vous



pouvez exécuter du code en mode pas-à-pas, placer des points d'arrêts.

Bien sûr, à tout moment, vous avez la possibilité de consulter la console qui propose de nombreuses informations sur les valeurs de vos variables.

Avec votre navigateur

L'autre moyen de déboguer vos applications consiste à corriger le contenu d'une page à travers votre navigateur. Pour cela, l'ensemble du secteur propose différentes extensions avec des rendus différents mais des informations identiques.

> FirePHP

FirePHP est une extension de la console FireBug pour le navigateur Mozilla. La communication entre le langage et le navigateur, passe par un simple appel d'une méthode du langage PHP. Les informations que vous obtenez dans la console, sont envoyées dans les headers de réponse. Bien entendu, elles ne casseront pas le rendu de vos pages.

Les autres

Ces deux extensions, qui ont été expliquées plus haut, sont disponibles sous un autre nom associé à votre langage de développement. Ces deux méthodes de débogage permettent de maintenir les pages web.

> Contrôle des standards

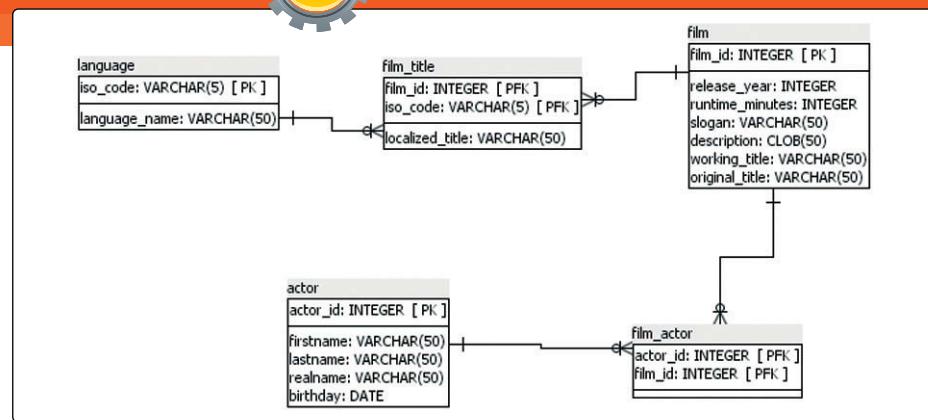
Le contrôle des standards dans la programmation est un point crucial, car si vous reprenez un ancien projet ou un projet existant auquel vous n'avez pas participé, il vous faudra passer par une phase d'études pour vous imprégner de celui-ci.

PHP CodeSniffer

La qualité d'un projet est importante car il est très facile de s'éloigner des spécifications d'origine si le projet dure très longtemps ou lorsqu'il est mal codé. PHP CodeSniffer va vérifier le respect des conventions de code mises en place autour de vos sources (=règles de programmation). Après son lancement, il va détecter pour vous les violations d'un standard de code défini. Le résultat que vous obtenez, permet d'identifier les lignes et les pages qui ne respectent pas la convention définie.

phpDocumentor

La documentation du code est aussi un autre point important à ne pas négliger. En PHP, il existe, entre autres, phpDocumentor qui



peut s'utiliser aussi bien en ligne de commande ou par l'intermédiaire d'une interface web.

PhpLangEditor

PhpLangEditor est une extension pour Firefox. Il va traduire à votre place des fichiers de langues et variables dans votre script. Comme ceci, si vous gérez un fichier de langue par fichiers, vous gagnerez énormément de temps au niveau de cette fonctionnalité.

PHP Lookup

Lookup PHP est une barre de recherche qui s'intègre à Firefox. Il vous aide dans la recherche rapide des références à syntaxe PHP. Bien entendu, il existe ce même type d'extensions pour les autres navigateurs.

> Les tests

Si vous souhaitez effectuer des opérations répétitives pour vérifier que votre site web fonctionne toujours, comme la navigation dans vos pages ou tester les différentes fonctionnalités, etc., vous vous lasserez très rapidement d'effectuer les mêmes opérations régulièrement, c'est pour cela qu'il existe de nombreux outils qui le feront à votre place. Et bien mieux !

PHPUnit et Atoum

Il s'agit de deux outils de la même famille, car ils vont effectuer des tests unitaires en automatique pour PHP. Chacun d'eux va vous aider à tester la stabilité de votre application web. Ainsi, au fil des semaines, vous pourrez répéter ces tests à volonté sans besoin de refaire ces opérations manuellement. Vous trouvez la même chose en Java avec JUnit, par exemple.

Selenium

Selenium (une référence à l'heure actuelle) est aussi un outil de tests, qui permet d'écrire des tests automatisés d'applications web d'interface utilisateurs, c'est-à-dire, qu'il va effectuer des tests à l'intérieur de votre navi-

gateur. Vous pouvez l'associer à un outil de test unitaire lié à votre langage de développement comme ci-dessus.

> Optimisations

eAccelerator

Il s'agit d'un accélérateur PHP, qui permet d'optimiser et de cacher le contenu dynamique. Il améliore la performance générale des scripts PHP en les mettant en cache. Par ailleurs, il optimise aussi les scripts pour accélérer leur exécution

APC

Il s'agit d'une librairie PHP permettant des économies de traitements divers dans le cœur de PHP. Ainsi, il va augmenter la vitesse générale de traitement d'une requête pour ce langage. Il fait partie des outils indispensables dans l'optimisation car il est très facile de le mettre en place dans son application.

Minify

Minify est un outil PHP qui compresse vos fichiers CSS et JS pour réduire le poids de chacun d'eux. Il existe différents modes de compression, depuis la suppression des espaces et des commentaires, jusqu'à la compression des fichiers au format gZIP ou ZIP. Pour ces options poussées, vous devez avoir installé l'extension ZIP en PHP pour effectuer cette opération.

> Conclusion

Nous vous avons montré quelques points importants que les développeurs ne pensent pas toujours à implémenter, et qui sont très facile à mettre en place dans vos développements. Bien sûr, il existe d'autres outils génériques pour tous les langages pour effectuer ces opérations de maintenance et d'aide dans le développement. A vous de jouer !

Christophe Villeneuve

Consultant 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.

Devenir développeur de jeux vidéo

Le jeu vidéo conjugue de nombreux métiers : développeur, graphiste, animateurs 3D, testeurs, etc. Si les jeux mobiles nécessitent « seulement » quelques jours ou quelques semaines, d'autres exigent de longs mois de conception.

Dossier réalisé par François Tonic



© Assassin's Creed 3

Vous rêvez de développer le prochain WoW ou Battlefield ? Développeur de jeux vidéo, voilà un job qui donne envie. Si la France est un pays avec de fortes compétences et des éditeurs très actifs, il ne suffit pas d'être bon développeur pour réussir. La liste des éditeurs et des indépendants français est longue. Citons : Ubisoft, Quantic Dream, Cyanide, Don't Nod, Asobo, Ankama, Pastagames, Arkedo. N'oublions pas de citer le très beau succès de Magma Mobile, éditeur de jeux mobiles, de notre ami Nicolas Sorel (Codes-Sources pour les connaisseurs). L'explosion du marché mobile a bouleversé le marché du jeu vidéo : prix en baisse, dématérialisation du support, et la crise économique est aussi passée par là. Le marché actuel du jeu n'a pas grand chose à voir avec le marché d'il y a 5 ans. La Wii puis la Kinect ont révolutionné l'interaction entre le joueur et le jeu.

> Plus de 250 sociétés mais souvent petites

Le syndicat SNJV, dans son livre blanc sur les jeux vidéo en France en 2012, donne quelques chiffres intéressants :

- l'industrie du jeu représente un chiffre d'affaires de 3 milliards €
- premier bien culturel : 50 millions de produits tout confondu

- 48 % des foyers français ont une console
- les jeux sur smartphones explosent : 25 à 30 % des applications téléchargées
- les plateformes mobiles sont des canaux de distribution très importants
- les plateformes de distribution telles que Steam ou Origin fédèrent plus de 80 millions d'utilisateurs

Le SNJV recense plus de 250 entreprises spécialisées sur le jeu dont un tiers ont moins de 2 ans d'existence. Le chiffre d'affaires varie énormément. L'export est un marché vital pour ces sociétés et éditeurs spécialisés. La France pèse à peine 5-7 % du marché mondial. Côté emploi, le marché fait travailler directement plus de 5 000 personnes (sans compter tous les emplois indirects), dont 3 000 dans la production. La moitié de ces entreprises compte moins de 10 salariés. Le SNJV estime qu'il y a 150 nouvelles offres d'emplois chaque mois. Ce chiffre est une moyenne.

> Les tendances selon le SNJV

- Une des difficultés du marché du jeu est de s'adapter aux évolutions : dans le type de jeu, le mode de consommation (distribution, terminaux). Le marché avec les jeux mobiles a exploité et de nombreux éditeurs pour smartphones se sont créés. Le syndicat a déterminé différentes tendances :
- transition technologique et économique : jeux

en ligne, jeux sur mobiles, mais le marché croît toujours

- la dématérialisation des jeux s'accélère, même si le support traditionnel continue à être profitable, il est très lié aux nouvelles consoles
- paiement à l'acte comme futur modèle de référence. Le modèle du free to play constitue à cet égard les prémisses d'un nouveau modèle de revenu universel pour l'industrie, quel que soit le type de jeu et le mode de distribution.
- Un financement plus diversifié et collaboratif : un jeu cela coûte cher !
- Un renouveau des productions indépendantes : c'est une chance pour un développeur indépendant de pouvoir se faire remarquer ou pour de petits studios d'émerger.

D'autre part, le syndicat met aussi en avant le succès de studios français qui ont produit de solides succès : Raymon Origins, From Dust, Wargame European Escalation, My Brute, Game of Throne, etc. Dans ce dossier, nous allons aborder de nombreux thèmes pour mieux comprendre comment devenir un développeur de jeu vidéo. Si ce métier fait rêver étudiants et développeurs, il est très exigeant. L'expérience et la parfaite maîtrise technique et technologique seront deux atouts cruciaux pour y faire carrière. Les meilleurs iront à l'étranger ou dans les studios les plus prestigieux. A vous de jouer !

Un secteur attrayant mais très concurrentiel et sous pression

Notre enquête auprès d'éditeurs et des recruteurs pose deux éléments : les développeurs sont attirés par le secteur du jeu vidéo mais les salaires ne sont pas plus élevés qu'ailleurs. Pis, il y a plus de demandes que d'offres, ce qui a tendance à peser sur les salaires. Explications.

Comme le rappelle Alexandre Pham (président Lynx RH, cabinet de recrutement), le jeu vidéo est un segment du marché informatique et il connaît une assez forte croissance. Des éditeurs et studios se créent régulièrement. « Il y a un côté glamour. Ce sont des passionnés. Mais il y a des contrastes entre les grosses machines très structurées avec leur ambiance de travail et les petits studios », précise Alexandre Pham.

Attention à ne pas avoir une fausse idée du monde du jeu vidéo, il s'agit d'un marché très concurrentiel, particulièrement sur les plateformes mobiles où la pression sur les développeurs est forte et plus l'éditeur sera petit, plus sa pérennité sera fragile (bien entendu, ce n'est pas une généralité), car son succès repose sur quelques jeux qui doivent être utilisés par des millions de joueurs. « Beaucoup de gens pensent que c'est tout rose ! Le développeur de jeu ne passe pas son temps à jouer. Il y a beaucoup de travail derrière », prévient Pierre Sulpice (GameDev Lab, Epitech). L'étudiant qui rêve d'être développeur de jeu doit avoir conscience de cette réalité.

> Salaire : développeur du monde...

Un développeur de jeux vidéo va gagner l'équivalent d'un développeur « classique ». Sur la région parisienne, nous serons vers 30 000 € / annuels, voire, même plutôt 30 – 32 k€. En province, nous serons dans la fourchette basse, voire, sous les 30 k€ comme nous l'a précisé Frédérique Doumic (PDG Ouat entertainment). « Ensuite, la fourchette est très large, pour les profils seniors / pointus nous serons entre 40 et 60 000 € », précise Alexandre Pham. Comme dans le marché du développement classique, les profils très pointus (ex. : outils 3D, animation, frameworks systèmes), les salaires peuvent rapidement monter, surtout pour les grands studios.

Ensuite, il y a la délocalisation du développement qu'il ne faut pas négliger. « C'est un métier très concurrentiel. Nous essayons de sous-traiter à l'étranger (des développements) où les coûts seront parfois 4 à 5 inférieurs aux coûts français », recadre Frédérique Doumic.

Cela sera vrai pour du core code sans valeur ajoutée, ou sur des technologies très spécifiques ou sur les besoins d'internationalisation des jeux. Là encore le problème vient du marché très concurrentiel du jeu notamment sur plateforme mobile et les budgets serrés des petits éditeurs.

D'autre part, l'attrait du jeu vidéo pousse les jeunes diplômés à envoyer plus facilement un C.V. La demande est souvent plus importante que l'offre réelle, ne favorisant pas la valorisation salariale.

> Quelles compétences ? Quels métiers ?

Le jeu vidéo fait intervenir de nombreux talents et technologies (selon le type de jeu) : ambiance sonore / musique, développeur natif, développeur multiplateforme, animateurs 2D / 3D, développeur gameplay, chef de projets, développeur d'outils / core code / moteurs, graphistes et créateurs, intelligence artificielle, R&D, veille technologique. Des studios tels que Ouat Entertainment

Les 3 G

Game Art : orienté graphisme, ambiance graphique et sonore,

Game Design : partie logique du jeu,

Gameplay : tout ce qui se rapporte au modèle de jeu, l'interaction jeu - joueur



recherchent plutôt des développeurs « passe-partout » très polyvalents, capables de s'adapter très rapidement à de nouveaux contextes, une nouvelle technologie ou plateforme. « Il y a un an, on faisait du jeu Flash, aujourd'hui on revient à du C++ et C », commente Frédérique Doumic. « Il n'y a pas un profil type. Certains vont faire plus de R&D avec moins de code, d'autres feront de la programmation « qui tache », d'autres plus de la 3D », renchérit Pierre Sulpice.

Un développeur gameplay sera souvent un ingénieur (bac + 5). Mais cela va dépendre du studio et du type de jeu : un jeu en ligne exige une maîtrise des technologies web, voire réseau, où même un modèle multi joueur ; un jeu mobile exige souvent la maîtrise des langages et outils adaptés à chaque plateforme (si le jeu est natif). Pour la 3D (graphisme, modélisation, animation, texture, etc.), le développeur devra maîtriser les outils, le C++ le plus souvent. Le but n'est pas de tout faire à la main mais d'utiliser des bibliothèques et des moteurs dédiés (Unity, Direct3D, OpenGL, etc.). Ce qui exige tout de même une solide expertise et une constante veille technologique.

L'autre profil est le développeur créant et maintenant les outils de création. Car souvent, les studios utilisent leurs propres outils, ce qui nécessite des développements très spécifiques.

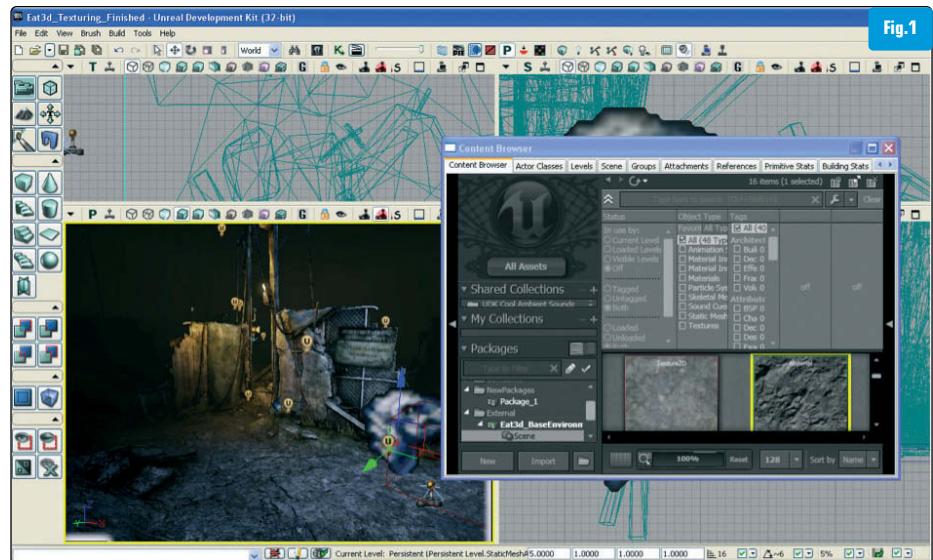
De nombreux métiers et des développeurs au cœur des jeux

L'univers du jeu vidéo attire énormément de monde, que ce soit les développeurs, mais également les artistes, scénaristes, etc. Il a dépassé, au niveau mondial, l'industrie du cinéma et de la musique. Mais en France, le secteur du jeu vidéo reste une niche, puisqu'il concerne moins de 300 entreprises en France (environ 5000 emplois), malgré le fait que la France soit le second pays producteur de jeux vidéo d'Europe.

> Les principaux métiers du jeu vidéo

Les grosses productions de jeu ayant atteint celles des productions de films, les génériques de fin sont devenus comparables à ceux des films hollywoodiens. Dans ces génériques on peut voir une multitude de profils différents comme un producteur, des Game Designers*, des Sound Designers, des artistes, des développeurs, et bien d'autres. Ces profils englobent bien évidemment chacun plusieurs métiers.

Le producteur de jeu supervise l'ensemble de la production, le délai, le budget, la communication externe, etc. Les Game Designers ont un métier très jeune, leur rôle était avant assuré par le Lead programmeur (responsable du développement). Ils s'occupent de la conception du jeu dans sa globalité, de l'idée au résultat final. Un Game Designer peut aussi bien être sur le gameplay*, le level design ou diriger l'équipe créative. Les Sound Designers s'occupent, comme on peut le deviner, du son. De l'acquisition aux musiques et bruitages finaux,



en passant par la phase de mixage, afin d'obtenir le bruit d'impact de balle voulu (et autres sons). Pour les artistes, cela va de l'infographiste 3D (création d'images et modèles 3D) au directeur artistique, en passant par le Character Designer (modélisa-

tion de personnages) et l'animateur. Les développeurs, quant à eux, peuvent être découpés en trois profils que nous allons détailler.

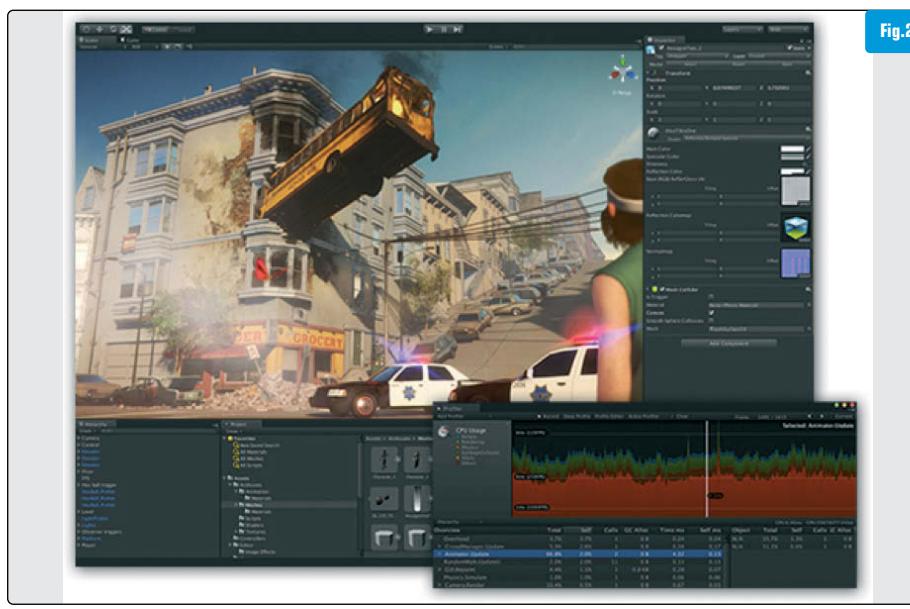
> Les développeurs

Il y a différents types de développeurs dans le monde du jeu vidéo. Vous pouvez assez facilement le deviner en regardant les jeux de ces dernières années.

On peut séparer ces développeurs en trois grosses catégories : moteur, middleware et gameplay.

Moteur

Le moteur de jeu est un ensemble de briques, qui une fois assemblées permettent de créer un monde virtuel plus ou moins réaliste. Il permet aux équipes de Game designer/développeurs de se concentrer sur le contenu et le déroulement du jeu, et non sur la technique de rendu 3D, ou la façon de faire transiter les données via le réseau, etc. Il y a bien sûr plusieurs types de développeurs moteur : 3D, physique, son, réseau, etc. Les moteurs d'aujourd'hui doivent être





Interview de Boris Briss, programmeur chez Ubisoft à Montréal

Afin de mieux comprendre ce qu'est le métier de développeur de jeux vidéo, je me suis rapproché de Boris pour lui poser quelques questions.

Tu travailles actuellement à Ubisoft Montréal, dans quelle équipe es-tu ?

Boris : je suis programmeur dans l'équipe qui a produit Far Cry 3 et je travaille plus précisément dans la cellule Gameplay.

Quel est le rôle de cette équipe ?

B. L'équipe gameplay est responsable de toutes les interactions entre le joueur et le jeu. Les contrôles, déplacements du héros, dégâts des armes, les mouvements de caméra... Je m'occupe en particulier du maniement et des interactions avec les véhicules.

As-tu à interagir avec d'autres types de profils ?

B. Bien sûr ! La production d'un jeu implique beaucoup de corps de métiers et j'ai à interagir avec beaucoup d'entre eux. En premier, avec mon Game Designer qui émet des volontés par rapport aux véhicules et qui va s'assurer qu'ils s'intègrent bien dans le reste du jeu. Les testeurs sont là pour valider ce que je produis, détecter les régressions ou simplement les problèmes de design. J'interagis avec d'autres programmeurs, spécialisés notamment en Intelligence Artificielle (les ennemis qui utilisent les véhicules) et Physique (collisions, accélération...). Aussi

avec des artistes, pour les effets spéciaux et pour les véhicules eux-mêmes. Et enfin, avec des ingénieurs son qui vont donner vie au jeu.

Pourrais-tu nous donner le déroulement d'une journée type ?

B. Ma journée commence par un thé, puis je rattrape les mails reçus durant la nuit (nous interagissons avec des gens partout dans le monde). Ensuite, je récupère la dernière version du jeu et vérifie si des bugs majeurs me concernant ont été trouvés par nos testeurs. Ensuite, réunion avec toute la cellule gameplay histoire de voir où tout le monde en est et ce qu'il y a à faire durant la journée. Généralement, je reçois des volontés de la part des designers. Nous en discutons et dès que nous arrivons à une solution qui nous convient, je passe au code. Une fois mon code validé par un de mes collègues, je soumets au reste de l'équipe !

Quelles sont les compétences primordiales pour travailler dans ton équipe ?

B. La volonté constante de révolutionner l'industrie du jeu vidéo ! Sincèrement, il faut avant tout être passionné de jeux. Ensuite viennent : une bonne communication (anglais et français), une ouverture

sur les autres domaines et savoir programmer, bien sûr.

Sur quels jeux as-tu travaillé ?

B. R.U.S.E, Child of Eden et Far Cry 3. Pour moi, c'est un peu particulier puisque j'ai aussi fait un certain nombre de courtes missions sur d'autres jeux tels qu'Assassin's Creed et Ghost Recon.

Quel est ton parcours, de ta formation à aujourd'hui ?

B. J'ai travaillé presque 3 ans dans l'équipe éditoriale d'Ubisoft à Paris avant de venir ici à Montréal ou je travaille depuis presque 2 ans.

Quelle a été ta formation post-bac ?

B. J'ai été à l'EPITA, une école d'ingénieur en informatique située près de Paris.

Où te vois-tu dans 5 ans ?

B. Je n'en sais rien, j'aime beaucoup ce que je fais pour le moment donc je vais continuer. Mais je pense m'orienter plus tard vers un rôle créatif ou alors quelque chose qui n'a vraiment rien à voir.

Quels conseils donnerais-tu à des étudiants ou jeunes diplômés passionnés de jeux vidéo ?

B. Ce qui m'a probablement le plus aidé lorsque j'ai cherché mon premier stage c'était de m'être constitué un portfolio contenant 3 projets que j'avais faits durant mes études et qui montrait ma motivation. Sinon, toujours rester à l'affût des nouveautés, lire et s'intéresser au monde !

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

Middleware

Le middleware est un ensemble d'outils, mis à disposition des Game Designers et développeurs gameplay. Il permet la création des niveaux, et le développement du gameplay. Le développeur middleware travaille donc sur l'amélioration de ce dernier, que ce soit sur la partie éditeur du monde pour les game designers, ou sur la partie compréhension des règles développées par les développeurs gameplay. Les principaux lan-

gages de programmations utilisés pour les middlewares sont le C++ et le C# [Fig.1 et 2].

Gameplay

Le gameplay est la perception du joueur lorsqu'il est en train de jouer. Pour la programmation gameplay on parle généralement de 3C : camera, control, character (caméra, contrôle, personnage). La caméra peut être différente suivant le type de jeu. Les trois types de caméras les plus utilisées sont : la 1re personne, la 3e personne et la vue de dessus. La caméra à la 1re et 3e personne sont le plus souvent utilisées pour des jeux de tir ou d'aventure, quant à la caméra en vue de dessus, elle est utilisée pour les RTS (Real-Time Strategy). Les contrôles sont bien évidemment les manettes, clavier, souris et autres pads. Leur maniabilité doit être

Définitions

- Game Designer** : Il n'y a pas directement de traduction pour Game Designer en français, on pourrait dire : concepteur de jeu.
- jouabilité** : ressenti d'un joueur lors d'une partie de jeu vidéo
- Epita** : Ecole Pour l'Informatique et les Technologies Avancées (www.epita.fr)
- Epitech** : l'Ecole de l'Innovation et de l'Expertise Informatique (www.epitech.eu)
- ENJMIN** : Ecole Nationale du Jeu et des Médias Interactifs Numériques (www.enjmin.fr)
- ISART Digital** : l'Ecole du Jeu Vidéo et du Film 3D (www.isartdigital.com)

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

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

> Compétences

Les compétences demandées en tant que développeur dans le jeu vidéo varient bien évidemment suivant le type de développement.

Le développeur moteur doit être très bon en C et C++. Il doit savoir lire et comprendre une documentation technique (en anglais).

Les optimisations spécifiques aux plateformes sont importantes pour qu'un moteur de jeu soit puissant, il doit donc connaître les spécificités des plateformes sur lesquelles il travaille.

Le développeur middleware doit quant à lui pouvoir s'adapter, dialoguer, comprendre les besoins. Il doit s'adapter et comprendre le moteur de jeu afin de pouvoir construire les outils qui se reposent dessus.

Mais le dialogue et la compréhension des besoins des game designers et des développeurs gameplay sont très importants pour permettre de faire les outils les plus pertinents. Les développeurs gameplay sont un peu dans le même cas que les développeurs middleware, ils doivent constamment échanger avec les Game Designers afin de répondre à leurs besoins, ainsi qu'avec les testeurs, afin de prendre en compte leurs retours.



> Les formations

Les formations pour devenir développeur, que ce soit moteur, middleware ou gameplay, sont vraiment variées. Attention tout de même, quelle que soit la formation choisie, il est indispensable d'être polyvalent, créatif, de savoir travailler en équipe et parler anglais afin de réussir dans ce milieu (la maîtrise de l'Anglais, surtout à l'oral, peut être un facteur d'échec à l'embauche, NDLR).

Généralistes

Les écoles de développement généralistes (comme Epita* et Epitech*) permettent d'apprendre le développement, et la plupart contiennent dans leur cursus soit une « option » jeu vidéo, soit un/des projet(s) dit libre(s) qui permettent aux étudiants de s'essayer.

Mais attention, ce n'est qu'une très petite partie des étudiants de ces écoles qui s'orientent vers le jeu vidéo.

Ce type de formation est généralement accessible directement en post-bac mais il y a également la possibilité d'intégrer ces écoles en 2e et 3e année.

Spécifiques

Les écoles de jeux vidéo ont, elles, un cursus complètement orienté vers le jeu (comme ENJMIN* et ISART Digital*). Elles ont également un atout par rapport aux écoles généralistes, qui est la mixité des profils. En effet ces écoles ne forment pas que des développeurs, elles forment également des artistes, des Game Designer, des Sound Designer et des producteurs. Il y a donc énormément d'échanges entre les étudiants des différentes formations, mais également des projets communs qui leur permettent de travailler en équipe avec les profils qu'ils rencontreront dans les studios. Suivant les écoles, l'entrée et le nombre d'années différent. Pour l'ENJMIN, la formation est accessible à partir du BAC+3, et a une durée de 2 ans. Pour ISART Digital, la formation est accessible directement après le BAC et dure 4 ans. Il est également possible d'intégrer le cursus en 2e et 3e année.



Benjamin Baldacci
Directeur Pédagogique Game Programming chez ISART Digital
www.isartdigital.com

L'information permanente

- L'actu de Programmez.com : le fil d'info quotidien
- La newsletter hebdo : la synthèse des informations indispensables.

Abonnez-vous, c'est gratuit !

www.programmez.com

PROgrammez !
Le portail des développeurs

Abonnement | Mon compte

Actualités | Magazine | Tutoriels | Livres blancs | Forum | Télécharger | Vidéo | Livres | Avis d'experts | Emploi

Recherche : RSS | NEWSLETTER | CONTACT

Abonnement

Supplément : Initial Parallel Studio

Actualités

Téléchargements

Livres blancs

Emploi

Livres

Agenda

Abonnement

Solutions

Flux RSS

Magazine

Sommaire

Minimag

Boutique

Abonnement

Archives

Interactivité

Enquête lecteurs

Tribune libre

Forum

Tutoriels

Tips

Amusante

Emploi

Candidats

Recruteurs

Mon espace

Inscription

Programmez ! est sur Twitter !

Désormais toutes actualités du site Programmez ! sont relayées sur Twitter. Si vous avez un compte Twitter, restez en permanence en contact avec nous et vous abonnez à nos tweets.

Actualités | Toute l'actualité | Flux RSS

Les AET – NI vues, ni connues

Les Techniques d'Évasion Avancées (AET) constituent un nouveau défi pour les équipes de sécurité. Découvrez comment elles peuvent être vaincues par les moyens de contournement...

Des stars du web et des TIC sur le salon La Mèlée Numérique

Plus de 2 000 visiteurs et 150 exposants se retrouvent à La Mèlée Numérique, événement annuel qui se déroule les 22 et 23 avril 2012 au Centre des Congrès Diagono à Lille (Tourcoing).

AFNOR recherche des experts en sécurité informatique

Oracle vs Google : Ellison enflame le clou, Page esquivé

Silk Central 12.0 et Silk Performer 6.0 : générer des tests logiciels

Les Tortues Flexeurs en 3D

ESET lance l'opération ma ville sans virus

SkySQL finalise un tour de table de séries A et lève 4 millions de dollars

La Nuit du Hack fête ses 10 ans !

Les actualités précédentes

Agenda

Cette semaine

Aucun événement

Séminaire prochaine

25/04 La Mèlée Numérique

27/04 Rencontre des entrepreneurs du numérique

Les jours suivants

Programmez ! n° 151

Le numéro 151 de Programmez ! est disponible en kiosque.

Consulter le sommaire | Lire le minimag | Acheter en PDF

Microsoft

Testez gratuitement Windows Azure pendant 90 JOURS. ▶

La plateforme pour héberger, exécuter et gérer vos applications dans le Cloud

Windows Azure

lomega

Powered by EMC

Bénéficiez de produits haut de gamme fonctionnaliés pour vos besoins

> Obtenir plus d'informations

Un bon étudiant ou développeur ne fait pas forcément un bon développeur de jeux vidéo !

Isart Digital est une école privée d'enseignement supérieur, spécialisée dans le jeu vidéo et les films 3D. Elle a été créée en 2001 et délivre un titre homologué de niveau II. Preuve de sa qualité, en 8 ans, elle a raflé 50 prix. Elle a tissé un partenariat avec le NCC, école de jeux vidéo japonaise. Et depuis janvier 2013, un établissement a été ouvert au Canada. Cyril Geordin, responsable des relations entreprises et institutionnelles, nous parle du métier, des cursus et de l'emploi.

Quel est aujourd'hui le marché du jeu vidéo pour un diplômé, quelqu'un qui souhaite devenir développeur de jeux ?

Cyril : Depuis plusieurs années, le marché français peine à trouver des programmeurs à embaucher ! Dans ce contexte très particulier de quasi plein emploi pour les programmeurs, ceux qui se spécialisent dans le jeu vidéo sont très recherchés. Qu'il code, assemble, sécurise, teste ou débogue, le programmeur devra toujours garder à l'esprit que l'utilisateur final est un joueur.

Développeur de jeux vidéo, qu'est-ce que cela signifie ? C'est quoi ? Quelles compétences, quelles motivations cela demande-t-il ?

Cyril : Pendant de longs mois, les acteurs du jeu vidéo se sont réunis et penchés sur cette question afin de définir très précisément les métiers du jeu vidéo. Le référentiel du SNJV (Syndicat National du Jeu Vidéo, NDLR), auquel l'école a participé, a listé les différents profils de « développeur de jeux vidéo ». Il y a en effet plusieurs métiers derrière ces termes : Lead Programmeur, Programmeur Moteur, Programmeur Gameplay et les Programmeurs spécialisés (IA, Outils, Physique, etc...). Tous ces métiers peuvent mener, avec de l'expérience et du talent, au poste de Directeur Technique. Des compétences sont communes aux programmeurs. Ensuite, chacun viendra ajouter son expertise correspondant à sa spécialité. En plus de l'apprentissage constant des outils, langages et particularités des moteurs de jeux vidéo, le programmeur devra être rigoureux, logique et réactif, et bien évidemment passionné de jeux vidéo.

La formation est importante mais en quoi consiste-t-elle ? Faut-il avoir un bon "feeling" technique et artistique à la base ?

Cyril : La formation est essentielle pour acquérir les différents langages et méthodes de programmation, apprendre à faire une veille technologique et maîtriser la logique de

programmation orientée jeux vidéo, et bien d'autres choses encore. Programmer pour le jeu vidéo implique également d'intégrer une équipe de production de jeu, et de travailler avec des game designers, des game artists, des sound designers... Il faut composer avec ces profils différents et les sensibilités artistiques, et comprendre les contraintes de chacun. Le travail en équipe s'expérimente. C'est pour cela que les étudiants en programmation jeu vidéo participent à la création d'un jeu vidéo en collaboration avec les autres formations de l'école en tant que programmeur (3e année) et lead programmeur (4e année).

Comment se déroule un cursus chez vous ?

Cyril : Le cursus de Game Programming dure 4 ans. Les deux premières années sont à temps plein et consacrées à la « pure » programmation. Un atelier d'un mois « sandbox » est intégré à la première année et permet de s'essayer à la programmation. La spécialisation en programmation jeu vidéo se fait en 3e et 4e année (en alternance). La spécialisation est accessible aux candidats extérieurs sur concours.

Quels débouchés en France et à l'étranger ?

Cyril : Les débouchés restent nombreux pour les programmeurs jeux vidéo. Il y a un milliard de joueurs dans le monde, dont 28 millions en France, le marché est dans une courbe de croissance très forte et ce pour encore plusieurs années. Certains observateurs considèrent que 50% des entreprises utiliseront la Gamification d'ici à 2015. Les compétences du programmeur jeux vidéo permettent à un profil junior de choisir la France ou l'un des pays qui recherchent également des bons programmeurs (USA, Canada, Japon, Chine, ...). Nous accompagnons d'ailleurs nos étudiants sur l'apprentissage des langues (Anglais, japonais) afin de leur donner ce choix, de s'expatrier, une fois diplômés.

Comment se situe la grille salariale dans ce métier ?

Cyril : D'après le référentiel métier du SNJV (Syndicat National du Jeu Vidéo), un programmeur spécialisé (IA, outils, physique,...), un programmeur Moteur et un programmeur Gameplay évolueront entre 27K€ et 45K€, un Lead Programmeur entre 32K€ et 45K€ et un Directeur technique entre 44K€ et 55K€. Tous ces chiffres sont des moyennes et dépendent de la taille du studio et du projet de jeu vidéo.

Quel avenir ?

Cyril : Au regard des différentes études institutionnelles, le marché du jeu vidéo bénéficie devant lui de plusieurs années avec une très forte courbe de croissance, qui va profiter à l'emploi des programmeurs jeux vidéo. A eux de continuer à se former tout au long de leur carrière, être en veille et alerte permanentes pour suivre les évolutions des technologies et langages de programmation. #



Viens créer ton jeu 2013

Les 6 et 7 avril prochains se dérouleront les finales du concours « Viens créer ton jeu » organisé par Epitech et Game One. Les inscriptions sont closes depuis fin février. Ce défi est réservé à tous les passionnés de 16 à 20 ans ! Aucune notion de développement n'est nécessaire : un logiciel spécialement personnalisé pour l'occasion te sera fourni. Après une séance de présentation de l'outil et du déroulement du week-end, tu auras le champ libre pour laisser cours à ton imagination.

Site : <http://www.vienscreertonjeu.fr>

Un joli défi !

Démystifier le développement des plug-ins d'Eclipse

4^e partie

Implémenter l'évaluation des expressions XPath

Nous disposons déjà d'un plug-in doté de templates pour l'édition XML en Java, des cheat sheets, dont certains utilisent des images personnelles. Notre objectif dans cet article est de développer une classe personnelle pour évaluer les expressions XPath, passées par l'utilisateur pour pouvoir les marquer sur le source XML. La classe utilisée dans le plug-in est nommée **XPathEval**. Cette classe va être utilisée par la suite à la fois pour l'exemple interactif et pour l'affichage d'une fenêtre dédiée au visualiseur XPath «**V.XPath**».

L'idée utilisée dans cette classe est simple, une expression XPath est évaluée. Selon son résultat le nom ou la valeur d'un nœud XML est modifié. Cette modification sert à marquer les parties concernées dans le fichier XML.

L'exemple interactif est parfait pour examiner l'évaluation XPath. Un ensemble d'expressions est fourni, permettant à la sélection de l'une d'elles d'afficher le résultat dans le source, avec une petite explication de l'expression.

> Créer un DOM et obtenir un objet XPath

```
public class XPathEval {
    private XPath path;
    private Document document;

    //constructeur
    public XPathEval(File file)
    {

        try {
            //On commence par obtenir un objet DocumentBuilder object,
            //en utilisant un objet DocumentBuilderFactory
            DocumentBuilderFactory factory=DocumentBuilderFactory.newInstance();
            DocumentBuilder builder=factory.newDocumentBuilder();

            //Une référence à un objet Document dans le DOM
            document=builder.parse(file);

            //Obtenir un objet XPath
            XPathFactory xpfactory = XPathFactory.newInstance();
            path = xpfactory.newXPath();

        } catch (Exception e)
        {
            e.printStackTrace();
        }
    }
}
```

Note : **DOM** est la représentation en mémoire de l'arborescence d'un document XML

XPath permet de localiser une partie(s) dans un document XML, sans être obligé de parcourir tout le document.

> Évaluer une expression XPath et trouver le type du résultat

```
//L'évaluation d'une expression peut retourner NodeList, Boolean
ou un objet Double.

//Il y a aussi une méthode retournant après l'évaluation d'une
expression une chaîne de caractères
//comme représentant du résultat de cette expression. Nous allons
commencer par utiliser cette méthode.

//Rechercher dans le résultat de cette méthode (String), on peut
trouver un indice sur le type susceptible
//d'être retourné, par l'expression XPath et par conséquent
utiliser une méthode d'évaluation plus
//appropriée

Object eval(String exp)
{
    Object object=null;
    if(exp==null || exp.trim().equals("")){
        return null;
    }

    try {

        //obtient le résultat d'une expression sous forme d'une chaîne
        //de caractères
        String result=path.evaluate(exp,document);
        result=result.trim();

        //Si l'expression est Boolean, la chaîne contiendra une valeur
        //true ou false
        if(result.equals("true") || result.equals("false"))
        {
            //nous savons maintenant que le résultat est un Boolean
        }
    }
}
```



```

//nous demandons le retour d'un Boolean, comme résultat
//avec XPathConstants.BOOLEAN
object=path.evaluate(exp,
document,XPathConstants.BOOLEAN);

//La chaîne doubleOrBooleanMsg est utilisée pour l'affichage,
du résultat si Boolean ou Number
//la chaîne ici contient true ou false
this.doubleOrBooleanMsg=object.toString();

//l'objet retourné va être analysé plus tard
return object;
}

//arrivé ici le résultat n'est pas Boolean. Second test, trouver
si un Number
try {

//on essaye d'obtenir le résultat comme un objet Double. Si l'exception
//NumberFormatException est lancée, ce n'est pas un Double
Double.valueOf(result);

//pas d'exception, donc XPathConstants.NUMBER (Double en Java)
object = path.evaluate(exp,
document,XPathConstants.NUMBER);

//ajouter la valeur du Double dans le message
this.doubleOrBooleanMsg=object.toString();
return object;

} catch (NumberFormatException en) {
//une exception est lancée. Peut-être le type est NodeList
}

//Arrivé ici on peut supposer que le type est
//NodeList

object = path.evaluate(exp,
document,XPathConstants.NODESET);

}

catch(Exception e)
{
e.printStackTrace();
return null;
}

return object;
}

```

> Doit-on modifier le DOM ?

Si la méthode d'évaluation d'expression XPath, qui retourne une chaîne **String**, retourne un résultat de type Boolean, Double ou null, on affiche simplement ce résultat à la fin de l'affichage XML. Par contre un autre type de résultat implique la modification de l'arbre

DOM, pour marquer les parties concernées (L'affichage se fera sous forme HTML dans le navigateur par défaut d'Eclipse). La méthode d'évaluation **String evalAll(String)** est appelée pour effectuer l'ensemble des opérations nécessaires pour aboutir à un stream HTML.

```

//cette méthode appelle la méthode eval(String). Si le résultat
est une NodeList
//on modifie le DOM, on transforme en HTML et on retourne une
chaîne de caractères en HTML
String evalAll(String exp)
{
Object object = eval(exp);

//object n'est pas null, Boolean ou Double. La méthode eval
(String), retourne
//donc un objet NodeList.
//Appeler la méthode modifyDOMTree() pour modifier le DOM
if(object != null
&&!object.getClass().getSimpleName().equals("Boolean"))
&&!object.getClass().getSimpleName().equals("Double"))

modifyDOMTree((NodeList)object);

//retourne le DOM comme une chaîne de caractères formatée en HTML
return transform2Html();

}

```

> Modifier le DOM

La méthode **ModifyDOMTree(NodeList)** Se charge de modifier le DOM. Les nœuds modifiés seront affichés avec une couleur différente en utilisant des balises HTML.

```

//obtenir un objet NodeList après l'évaluation de l'expression
XPath. Modifier les nœuds trouvés //dans le DOM
private boolean modifyDOMTree(NodeList nodeList)
{
boolean modified=false;
if(nodeList ==null)
return false;

//boucler sur les nœuds retournés
for(int i=0;i<nodeList.getLength();i++)
{

Node node=nodeList.item(i);

//pour les nœuds éléments et attributs on modifie le nom, pour
les textes et commentaires
//on modifie la valeur du nœud
switch (node.getNodeType())
{
case Node.ELEMENT_NODE:
case Node.ATTRIBUTE_NODE:

```

```

modified=modifyNodeName(node);
break;
case Node.TEXT_NODE:
case Node.COMMENT_NODE:
modified=modifyNodeValue(node);
break;

} //~switch
} //~for nodelist

return modified;
}

```

> Modifier le nom d'un nœud

Dans notre méthode `ModifyNodeName(Node)`, on utilise la méthode `renameNode(Node n, String namespaceURI, String qualifiedName)` de l'objet `Document`, en mettant une sorte de balise personnalisée avant et après le nom du nœud. Ces balises vont être remplacées par la suite par des balises ou tags de présentation HTML

```

private boolean modifyNodeName(Node node)
{
Node newNode=this.document.renameNode(node, node.getNamespaceURI(),
<b_span_>+node.getNodeName()+<_e_span_>);

return true;
}

```

> Modifier la valeur d'un nœud

Les nœuds de type texte ou commentaire n'ont pas de noms. Pour cette raison, on change leur valeur. Comme pour les noms, cette valeur va être affichée avec une couleur différente dans HTML, en entourant cette valeur par une balise personnalisée. La méthode `ModifyNodeValue(Node)` utilise la méthode `setNodeValue(String nodeValue)` de l'objet nœud `Node`.

```

private boolean modifyNodeValue(Node node)
{
node.setNodeValue(<b_span_>+node.getNodeValue()+<_e_span_>);

return true;
}

```

> Préparer l'affichage HTML

Notre méthode `String evalAll[String exp]` retourne une chaîne de caractères, formatée en HTML. La méthode `transform2Html()` se charge de formater tout, correctement, en HTML. Les nœuds modifiés sont affichés en couleur différente reflétant le résultat retourné par l'évaluation d'une expression XPath.

```

private String transform2Html()
{
ByteArrayOutputStream memBuffer=null;
String sBuffer=null;
try
{
//construire un buffer dans la mémoire en utilisant
ByteArrayOutputStream

```

```

//ceci peut être utilisé pour écrire/lire le DOM, comme dans
un fichier, il est utilisé pour modifier
//des tags HTML

```

```

memBuffer=new ByteArrayOutputStream();

//construire un objet Transformer. Cet objet sert à écrire en XML
//vers le buffer
Transformer t = TransformerFactory.newInstance().newTransformer();

//la sortie doit être en XML
t.setOutputProperty(OutputKeys.METHOD, "xml");

```

```

//écrire le DOM modifié en mémoire
t.transform(new DOMSource(this.document), new StreamResult
(memBuffer));

```

```

//obtenir le buffer en mémoire dans une chaîne de caractères
sBuffer=memBuffer.toString();

//remplacer les tags (personnalisés) ajoutés au nom ou la
valeur d'un nœud
//les résultats sont des tags <span> utilisés avec un style CSS
sBuffer=sBuffer.replaceAll("<b_span_>","<span>");
sBuffer=sBuffer.replaceAll("<_e_span_>","</span>");


```

```

//après l'étape précédente un nœud avec un nom modifié devient
//<span>node_name</span>
//...
//</span> node_name</span>

```

```

//pour les tags ouvrants d'un nœud (premier cas mentionné juste
avant),
//déplacer < après [span] et [/span] après >. Le second cas
est valable pour
//la fin d'une balise de fermeture
sBuffer=sBuffer.replaceAll("<\\<\\[span\\]>","<span>"); 
sBuffer=sBuffer.replaceAll("<\\<\\[span\\]\\>",">[/span]>");


```

```

//pour les tags fermants (l'ouverture d'un tag fermant), déplacer
</, après [/span]
sBuffer=sBuffer.replaceAll("<\\<\\<\\[span\\]>","<span></>");


```

```

//remplacer les < et > par des caractère d'échappement pour
l'affichage en HTML
sBuffer=sBuffer.replaceAll("<","&lt;"); 
sBuffer=sBuffer.replaceAll(">","&gt;<br/>");


```

```

//remplacer [span] par <span>
sBuffer=sBuffer.replaceAll("<\\[<,>>"); 
sBuffer=sBuffer.replaceAll("<\\]>",">");


```

```

//si message concernant le type Boolean ou Double, ajouter le

```

```

message ici

if(this.doubleOrBooleanMsg != null && !this.doubleOrBooleanMsg
.trim().equals(""))

{
sBuffer+=><br/><br/><span>Result:>+this.doubleOrBooleanMsg+<
/&span>;

}

//ajouter un en-tête et une fermeture pour l'HTML (ces deux
sont déclarés au niveau de la classe
//avec un style pour le tag <span>
sBuffer=this.beginHtml+sBuffer+this.endHtml;
}

catch(Exception e)
{
e.printStackTrace();
}
return sBuffer;
}

```

> Afficher le résultat d'une expression XPath

Nous avons déjà vu, comment afficher des images dans les cheat sheets. Ceci est aussi valable pour d'autres types de ressources. On

utilise la classe personnelle définie dans notre plug-in **AbstractItemExtensionElement1** et sa méthode **createControl(Composite)**, en ajoutant un attribut d'item dans une cheat sheet à la place désirée d'affichage. Pour l'affichage en HTML, on utilise le navigateur fourni avec Eclipse.

Note :

La classe **AbstractItemExtensionElement1** étend (extends) la classe **AbstractItemExtensionElement**. Dans cette classe, nous ajoutons des méthodes personnalisées (override)

Note :

Nous allons réutiliser le même attribut **myAtt** défini précédemment dans le plug-in. Seule la valeur de l'attribut change, ce qui permet de personnaliser l'affichage selon la valeur utilisée. Par exemple si la valeur passée à la méthode, **handleAttribute(String)** est **browser**, nous appelons la méthode **XPathEval.evalAll(String)** et affichons le fichier XML en HTML, avec si besoin des parties marquées, dans le navigateur d'Eclipse. Si sa valeur est **copy**, nous affichons un message de copyright, etc.

Voir le code complet sur le site www.programmez.com

Kaesar Alnijres

Développeur Java - Leader jug-cergy

<http://www.java-javafx.com>

Formez et fidélisez vos équipes techniques en leur offrant un Abonnement à Programmez !

Une formation et une veille technologique, à prix réduit



1 an d'abonnement = 39 € seulement par abonné (à partir de 10 abonnés)

Pour 3,25 € par mois par développeur,

+Offre de lancement : 1 page de Publicité offerte pour votre entreprise !

À partir de 100 abonnés, offre de lancement

Adressez un mail à diff@programmez.com, en indiquant le nombre d'abonnés souhaité, pour recevoir le devis.

www.programmez.com

Déployer une application Java EE dans le Cloud avec Heroku

Thématique ne cessant de faire les gros titres depuis de nombreux mois déjà, le Cloud est omniprésent dans nos vies quelle que soit sa forme d'utilisation. Concept connu depuis longtemps, le Cloud a réellement émergé au cœur des années 2000 sous l'impulsion de géants du Web comme Google ou Amazon. Malgré quelques réticences initiales, les entreprises n'auront d'autre solution à terme que de s'y intéresser sous peine de rater le train en marche.

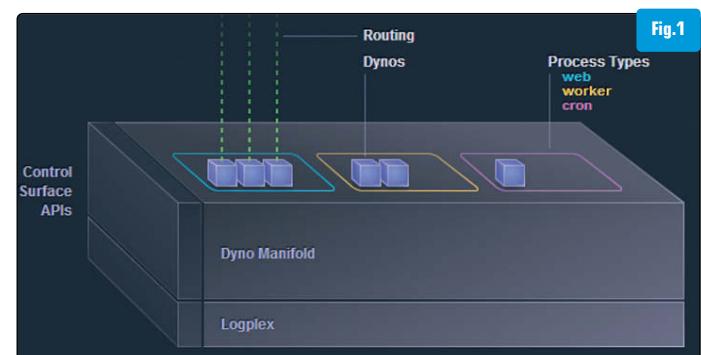
Dans ce contexte, les services Cloud de type PaaS comme Heroku présentent un intérêt évident pour déployer rapidement des applications web dans le Cloud en bénéficiant de toute la puissance de ces infrastructures. Dans cet article nous nous proposons, après une présentation d'Heroku, de voir comment déployer une application Java EE dans le Cloud grâce à la plateforme. Proposant un grand nombre de composants additionnels, Heroku permet ainsi un choix intéressant de bases de données qu'elles soient relationnelles (MySQL ou PostgreSQL) ou bien NoSQL avec MongoDB et Redis par exemple. Les nombreux add-ons disponibles couvrent la plupart des besoins classiques des applications d'entreprise à savoir l'envoi de mails, la génération de documents PDF ou Excel, les besoins en systèmes de messages asynchrones avec RabbitMQ, la gestion du cache ou bien encore le monitoring. Bien entendu, cette liste est loin d'être exhaustive et les possibilités offertes sont nombreuses.

> Architecture

Reposant sur AWS (Amazon Web Services), l'infrastructure d'Heroku met à profit plusieurs de ces services au sein de son architecture parmi lesquels on peut citer EC2 pour l'exécution du code et S3 pour le stockage. Le but étant de mettre à disposition de l'utilisateur final un environnement de déploiement et d'exécution collant au plus près à ses besoins. Répondant au doux nom de Celadon Cedar, la stack d'exécution par défaut d'Heroku est basée sur un Ubuntu en version 10.01 et met en pratique les 12 facteurs permettant de construire, déployer et gérer des applications web modernes (<http://www.12factor.net/>). Cette stack constitue un environnement d'exécution flexible, résistant sur la durée, polyglotte car n'étant basé sur aucun langage de programmation en particulier et offrant tous les outils nécessaires à son introspection.

Au sein de l'architecture d'Heroku [Fig.1], le point d'entrée du développeur se situe au niveau des API de contrôle. Ainsi, le développeur va gérer la configuration de son projet via GIT et un dépôt distant situé sur la plateforme, tout en utilisant un ensemble d'outils spécifiquement développés pour l'occasion : le Heroku Toolbelt. Constitué d'un outil en ligne de commande, il permet de créer et de générer des applications sur Heroku et propose en sus, l'outil Foreman pour exécuter et débugguer une application localement.

Une fois une application précédemment créée sur Heroku poussée vers le dépôt Git distant, la plateforme se charge instantanément de la compiler en un slug qui est une copie compressée et pré-packagée de l'application. Il sert à optimiser sa distribution à travers le



Architecture générale d'Heroku

Dyno Manifold dont nous détaillerons le travail un peu plus loin. Un slug constitue ainsi une unité matérielle logique adressant son propre système de fichiers transitoire et possédant une capacité en RAM exclusive ainsi que d'autres paramètres qui lui sont propres. Durant la compilation en slug, la plateforme analyse le code déployé afin d'en déduire le type d'application à faire tourner et de déterminer le Buildpack nécessaire à son bon fonctionnement. La plateforme n'ayant pas de support pour un langage de programmation spécifique par défaut, c'est en s'appuyant sur les Buildpacks qu'elle peut offrir un support pour un langage en particulier. Elle est de fait bel et bien polyglotte. Il est bon de noter sur ce point que la plateforme est facilement extensible puisqu'il est possible de créer son propre Buildpack qui n'est au fond qu'un ensemble de scripts.

> Plateforme centrée processus

L'architecture Heroku étant centrée sur les processus et non sur des instances virtuelles ou des serveurs, une application peut définir les processus nécessaires à son exécution. C'est durant la compilation en slug que la plateforme détecte les processus définis pour une application et son bon fonctionnement, en se basant sur le contenu du fichier Procfile d'un projet. Si c'est par défaut, ce fichier n'est pas nécessaire puisque Heroku crée un processus de type web pour une application, il est recommandé de le définir pour une plus grande flexibilité et un meilleur contrôle. En ce qui concerne les processus, la plateforme adopte le modèle Unix qui est une abstraction puissante pour exécuter des programmes côté serveur. Appliquée au monde du web, cela nous donne une solution unique facilitant la division des tâches et la scalabilité horizontale. Concrètement, un type de processus est un prototype de processus facilitant son instanciation. Le parallèle avec le monde objet se fait très facilement en considérant

la définition d'une classe et l'objet représentant son instantiation. Typiquement, un ou plusieurs processus de type web seront utilisés au sein d'une application. Il s'agit en fait d'instances de serveurs web permettant l'exécution de requêtes routées vers l'application. En sus, des processus de type worker pourront être adjoints afin de traiter en parallèle les calculs tiers. Enfin, d'autres processus systèmes sont accessibles avec notamment la possibilité de scheduler des traitements à un instant donné.

> Dynos

Afin de faire tourner ces différents processus, des unités d'exécution appelées dynos sont utilisées. Un dyno constitue un conteneur pleinement isolé et résistant dans le temps, chargé d'exécuter une commande unique spécifiée par l'utilisateur. La scalabilité de la plateforme se situe au niveau de ces dynos puisqu'il est possible de spécifier le nombre de dynos alloués pour un processus donné d'une application à n'importe quel moment. Si la scalabilité horizontale d'un slug est pratiquement infinie pour chacun de ses dynos, il est bon de garder à l'esprit que pour des applications web une stratégie trop agressive peut ne pas s'avérer payante. En effet, dans ce type d'applications les performances peuvent être vite limitées par les accès base de données. Attention donc à bien gérer la scalabilité d'une application de manière cohérente. Tout au long de son travail, le dyno va produire des logs détaillés en s'appuyant sur le composant Logplex de la plateforme, ce qui permet à l'utilisateur de savoir à tout moment ce qui est en train de se produire.

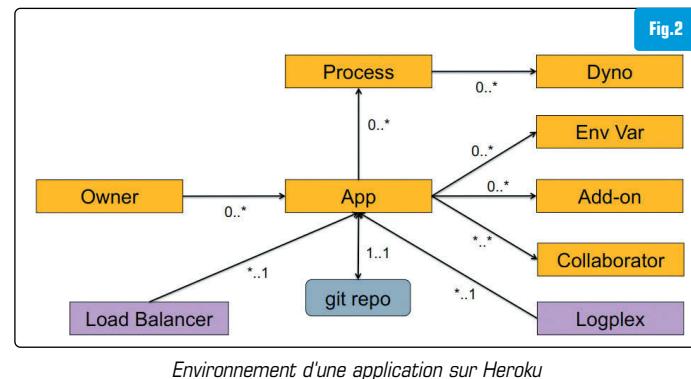
> Dyno Manifold

La gestion des dynos est réalisée par le Dyno Manifold qui constitue un environnement d'exécution scalable et tolérant aux erreurs pour les dynos d'une application. A partir d'une requête HTTP, il permet de sélectionner le ou les dynos à utiliser pour y répondre convenablement. Les différents dynos affectés à un même processus étant exécutés dans des environnements physiques distincts, la plateforme offre bien une tolérance aux erreurs de haut niveau. Le Dyno Manifold affranchit l'utilisateur du travail de gestion des processus d'une application dans le temps, en prenant à sa charge le démarrage et l'arrêt des dynos lorsque cela est nécessaire. Par exemple, lorsqu'un de ces derniers connaît une consommation mémoire supérieure à ce qui lui est permis. Lorsqu'un dyno dépasse son quota de 512Mb de mémoire, des messages d'erreur spécifiques sont produits au sein des logs et le dyno incriminé est instantanément arrêté et remplacé grâce au travail de gestion du Dyno Manifold. Il en va de même lors du crash d'un dyno, ce qui permet à l'utilisateur de la plateforme de s'affranchir d'une part de travail non négligeable.

> Routage HTTP

En termes de gestion des requêtes HTTP, Heroku load-balance automatiquement les requêtes reçues en ajoutant à cela le routage depuis le hostname d'une application vers ses dynos de type web. Le composant de routage se base sur un algorithme aléatoire pour le dispatching des requêtes vers les processus web. Il est bon de signaler qu'avec un langage comme Java fournissant un support multithread, il est bien entendu possible de gérer des connexions multiples au sein d'un seul et unique processus.

L'environnement d'une application [Fig.2] peut sembler complexe à première vue avec la nécessité de se familiariser avec la terminologie Heroku en se frayant un chemin au travers des slugs, dynos ou



autres profils. Néanmoins, le processus de déploiement reste simple et la maîtrise des différents concepts exposés jusqu'à présent n'est pas obligatoire pour débuter sur Heroku. Il est cependant toujours intéressant de mieux comprendre ce qu'il se passe sur une plateforme, ce qui s'avérera utile au fur et à mesure lors de cas d'utilisation plus poussés.

> Prix

Une fois un compte utilisateur créé, Heroku propose gratuitement l'exécution d'une application avec un seul dyno faisant tourner un processus de type web et la mise à disposition d'une base de données de 5Mb. Après les premiers tests de la plateforme, les besoins augmentent et avec eux la nécessité de passer aux formules payantes de la plateforme. Le modèle économique de Heroku repose sur un adressage fin du temps d'utilisation des dynos utilisés par un slug. Ainsi, l'allocation de dynos supplémentaires, de type web ou worker, sera facturée 0.05 \$ de l'heure mais permettra en contrepartie d'améliorer les performances d'une application, que ce soit en nombre de requêtes HTTP traitables ou en termes de puissance de calcul. A contrario, certains processus systèmes sont monitorés directement à la seconde près pour la facturation ! Heroku est donc un PaaS automatisé uniquement en exécution puisqu'il est à la charge de l'utilisateur de gérer au mieux les besoins en scalabilité horizontale d'une application. Cette gestion ayant d'ailleurs un impact direct sur la facturation d'une application par Heroku. Il est bon de noter que diverses possibilités existent pour automatiser cette gestion du scaling, que ce soit via des services tiers ou via l'offre grands comptes de la plateforme. Enfin, l'extension de la mémoire allouée à la base de données d'une application aura également un coût.

> En pratique

Pour les besoins de l'article, nous considérons une application Java EE 6 utilisant les principales technologies définies par la spécification. Basée sur JSF 2, l'application utilise ainsi également JPA 2 pour la persistance et CDI pour l'injection de dépendances. Pour fonctionner, elle nécessite de fait la présence d'une base de données, ce qui permettra de tester la connexion à une base de données PostgreSQL sur Heroku. Une fois la création d'un compte gratuit réalisée sur la plateforme, il faut télécharger la suite Heroku Toolbelt à l'adresse suivante : <https://toolbelt.heroku.com/>. Une fois celle-ci installée, votre poste pourra accéder à Git et à Foreman, le client ligne de commande Heroku permettant de travailler et d'échanger avec la plateforme. Dans un terminal, la première étape consiste à s'identifier sur la plateforme à l'aide de la commande heroku login en fournissant son e-mail ainsi que son mot de passe d'inscription. Si vous ne possédez pas de clé publique SSH, la ligne

de commande Heroku vous proposera d'en créer une. Celle-ci étant notamment nécessaire pour gérer votre projet avec Git.

> Initialisation du dépôt Git

Une fois placé dans le dossier de l'application web à déployer sur Heroku, nous allons pouvoir initialiser le projet sous Git. Pour ce faire, nous précisons à Git que le répertoire target est à ignorer et nous réalisons le premier commit de la sorte :

```
echo target > .gitignore
git init
git add .
git commit -m "Init"
```

> Configuration de la base de données

Par défaut, Heroku provisionne pour certaines applications une base de données PostgreSQL proposée en tant que add-on sur la plateforme. Afin de vérifier la présence de cette base, il est possible de saisir la commande heroku info au sein d'un terminal. Au cas où la base ne serait pas déjà présente, la ligne de commande suivante est à exécuter pour ajouter le add-on PostgreSQL :

```
heroku addons:add heroku-postgresql
```

D'une taille de 5Mb, cette base suffit durant la phase de développement pour différents tests mais pour une application en production, il est nécessaire de passer à des offres supérieures et payantes d'Heroku. Les informations de configuration de la connexion à la base de données sont rendues disponibles au travers de la variable d'environnement DATABASE_URL.

Si la valeur de la variable peut être obtenue suite à un appel à la commande heroku config, il n'est pas recommandé d'utiliser cette valeur au sein d'un fichier statique puisqu'elle peut changer à tout moment en fonction de l'environnement.

De fait, il est nécessaire de récupérer sa valeur à l'exécution au sein du code d'une application. La chaîne de caractères retournée par DATABASE_URL est formatée de la sorte :

```
[database type]://[username]:[password]@[host]:[port]/[database name]
```

Pour une application Java EE utilisant JPA 2 comme celle que nous allons déployer sur le Cloud, il est donc nécessaire de modifier la configuration d'accès à la base au démarrage de l'application. Pour ce faire, il est possible d'employer le code suivant :

```
Map addedOrOverriddenProperties = new HashMap();
URI dbUri = new URI(System.getenv("DATABASE_URL"));
String username = dbUri.getUserInfo().split(":")[0];
String password = dbUri.getUserInfo().split(":")[1];
String dbUrl = "jdbc:postgresql://" + dbUri.getHost() + ':' +
dbUri.getPort() + dbUri.getPath();
// Surcharge des propriétés
addedOrOverriddenProperties.put("hibernate.connection.url", dbUrl);
addedOrOverriddenProperties.put("hibernate.connection.username",
username);
addedOrOverriddenProperties.put("hibernate.connection.password",
password);
// Création avec surcharge des propriétés
EntityManagerFactory factory = Persistence.createEntityManagerFactory("pu",
addedOrOverriddenProperties);
```

> Ajout du Jetty Runner

Jetty Runner va nous permettre d'exécuter l'application web Java EE comme une application Java standard sans avoir à la déployer au sein d'un conteneur. Il s'agit en fait d'un simple jar, nommé jetty-runner.jar, qu'il va falloir copier au moment du packaging du projet au sein du dossier target de l'application et qui sera utilisé pour exécuter le war du projet.

Ce dernier étant basé sur Maven, il suffit de mettre à profit le plugin maven-dependency-plugin en ajoutant le code suivant dans la partie plugins du fichier pom.xml :

```
<plugin>
<groupId>org.apache.maven.plugins</groupId>
<artifactId>maven-dependency-plugin</artifactId>
<version>2.3</version>
<executions>
<execution>
<phase>package</phase>
<goals>
<goal>copy</goal>
</goals>
<configuration>
<artifactItems>
<artifactItem>
<groupId>org.mortbay.jetty</groupId>
<artifactId>jetty-runner</artifactId>
<version>8.1.8.v20121106</version>
<destFileName>jetty-runner.jar</destFileName>
</artifactItem>
</artifactItems>
</configuration>
</execution>
</executions>
</plugin>
```

> Procfile

Comme expliqué précédemment, Heroku met gratuitement à disposition un dyno avec un processus de type web que nous employons pour notre application Java EE. Cette configuration s'effectue au sein d'un fichier Procfile positionné à la racine du projet. En son sein, il faut préciser que nous lancerons le dyno web pour le Jetty Runner avec en entrée le WAR généré pour l'application. Cette opération est détaillée de la sorte :

```
web: java $JAVA_OPTS -jar target/dependency/jetty-runner.jar -
-port $PORT target/*.war
```

Le processus étant de type web, les requêtes HTTP reçues en entrée seront transmises au Jetty Runner et par ce biais à notre application.

> Test local

Avant de déployer l'application sur la plateforme Heroku, nous la testons en local pour vérifier son bon fonctionnement et débugguer d'éventuels problèmes. Avant toute chose, nous packageons l'application grâce à Maven via mvn package. Ensuite, le test peut être réalisé de 2 manières différentes. La première solution consiste simplement à exécuter au sein d'un terminal la ligne de commande suivante :



```
java -jar target/dependency/jetty-runner.jar target/*.war
```

L'autre solution se base sur l'outil Foreman. Avec la commande foreman start, l'utilitaire s'appuie sur le fichier Procfile à la racine pour lancer d'éventuels processus de type web ou worker. Il est bon de noter que par défaut, Foreman utilise le port 5000.

> Déploiement Heroku

Les dernières modifications du projet poussées vers notre dépôt local Git, nous pouvons créer l'application sur Heroku avec heroku create. Il en résulte la création d'une application Heroku basée sur la stack Cedar ainsi que du dépôt Git distant correspondant dont l'URL d'accès est affichée à l'écran [Fig.3].

Le déploiement du code se fait ensuite simplement en poussant les modifications du projet vers le dépôt distant Git sur Heroku :

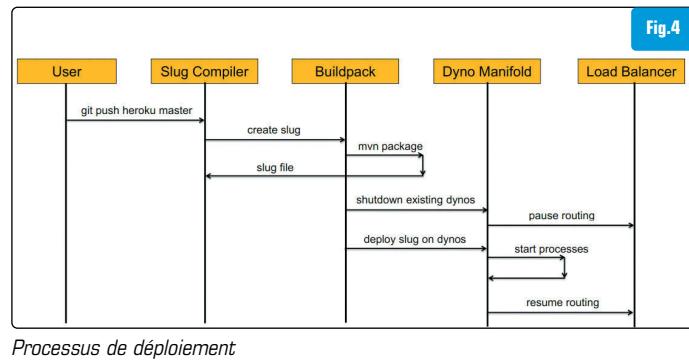
```
git push heroku master
```

Du côté de la plateforme, l'arrivée de ce code lance le processus de déploiement [Fig.4]. Ainsi, l'application est packagée en slug avant de détecter le type de Buildpack à utiliser pour son packaging. Dans notre cas, il s'agit bien entendu du Buildpack Java. Heroku lance donc un mvn package de l'application. Les différents dynos en cours d'exécution sur Heroku sont stoppés par le Dyno Manifold de même que le routing HTTP est mis en pause. Le nouveau slug est alors déployé, et c'est alors que le Dyno Manifold va démarrer les dynos désormais nécessaires à l'exécution de la nouvelle version packagée de l'application. Le routing HTTP peut alors de nouveau reprendre. Si le déploiement est réalisé avec succès par la plateforme, le terminal affiche l'URL à laquelle l'application est accessible et quelques informations la concernant telles que la taille de son slug ou bien ses processus associés. La [Fig.5] présente une copie d'écran de l'application Java EE déployée sur le Cloud avec Heroku.

> Scaling Horizontal

L'application déployée, il est désormais possible d'accéder aux processus qu'elle utilise via la commande heroku ps. Cette même commande permet de jouer avec son scaling en lui ajoutant des dynos à la volée. Ainsi, nous pouvons définir que l'application utilisera 3 dynos web comme suit : heroku ps:scale web=3. Il en va bien entendu de même pour les autres types de processus proposés par Heroku, que ce soit ceux de type worker ou les processus système. Enfin, les logs de la plateforme sont accessibles via la commande heroku logs pour laquelle il est possible de filtrer les logs uniquement par un type

Fig.3
Invite de commandes
C:\Users\sylsau\workspace\javaee6-sample>heroku create
Creating rocky-woodland-6468... done, stack is cedar
http://rocky-woodland-6468.herokuapp.com/ | git@heroku.com:rocky-woodland-6468.git
Git remote heroku added
Création de l'application Heroku



de processus. Pour le processus de type web, cela donnerait donc : heroku logs -ps web.

> Outilage

Au cours de l'article, nous avons principalement travaillé avec la ligne de commande pour gérer notre projet et interagir avec la plateforme Heroku. Ce mode de travail aisément peut ne pas convenir à certains qui préféreront gérer les échanges avec la plateforme directement au sein de leur IDE. En ce sens, Heroku met à disposition un plugin pour Eclipse permettant de s'affranchir complètement de la ligne de commande à condition d'avoir au préalable installé le plugin EGit. Le plugin Heroku permet de créer un projet Eclipse pour Heroku à partir de templates mais également d'importer des projets Heroku existants directement depuis la plateforme. En outre, les changements d'un projet peuvent directement être appliqués sur le dépôt Git distant. Une vue affichant les différentes informations de ses projets Heroku est également disponible. Enfin, le monitoring des applications Heroku est rendu possible directement dans l'IDE. La liste de fonctionnalités présentée ici est loin d'être exhaustive et la mise en place de ce plugin s'avère être une réelle plus-value dans l'optique d'accélérer les développements pour Heroku.

> Conclusion

Plateforme conçue pour développer et déployer des applications rapidement et simplement dans le Cloud, Heroku apporte quelque chose de réellement nouveau dans le paysage des applications en général et du monde Java en particulier. Son approche résolument agile permet de mettre en œuvre le déploiement d'applications continu dans le Cloud grâce à une architecture exploitant la puissance de Git. En outre, le processus de développement traditionnel est mis aux oubliettes avec Heroku qui se base sur 3 actions essentielles : développer, déployer et gérer une application. En bref, il s'agit d'un PaaS automatisé en exécution facilitant grandement la vie des développeurs et dont l'aspect polyglotte au niveau des langages de programmation est un avantage indéniable. Le modèle économique proposé permettant de gérer finement la scalabilité horizontale d'une application est intéressant et les nombreux add-ons déjà disponibles pour la plateforme sont autant d'atouts solides qui font de Heroku une plateforme des plus prometteuses dans la perspective de déployer des applications dans le Cloud.

Sylvain Saurel - Ingénieur d'Etudes Java / JEE
sylvain.saurel@gmail.com

Application Java EE sur Heroku

Lua, vous avez dit Lua ?

Lua ? Vous ne connaissez pas ? Pourtant, il fait partie des 20 langages référencés par l'index TIOBE (1). Nous allons le découvrir dans cet article !

> YASL ? Bof...

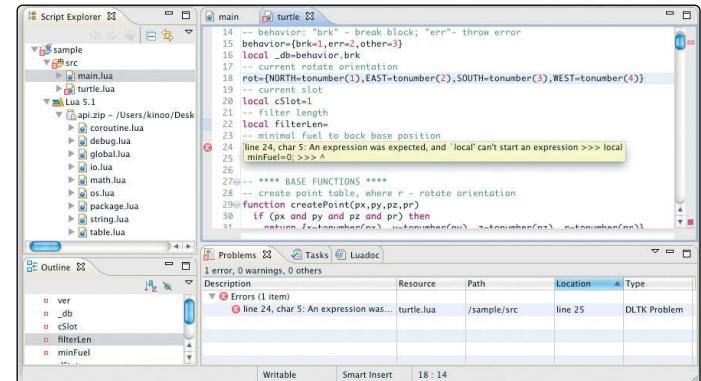
YASL (Yet Another Scripting Language) ? Bof, encore un langage de script me direz-vous, qui vient allonger la longue liste des ténors du domaine : Perl, PHP, Python, Ruby, Lisp, etc. Certes, on est souvent sceptique – voire paresseux – à l'idée d'explorer un nouveau langage lorsque celui qu'on utilise nous suffit, bon an mal an. Toutefois, Lua se place sur une niche peu courante au sein des langages de scripts : il est extrêmement minimalistes dans sa conception, puissant, rapide et largement interfaçable. Minimaliste, car la taille de l'interpréteur une fois compilé est d'environ 200 Ko et que toute l'architecture interne est basée sur des piles et des tables. Tiens, aurais-je attiré votre attention ? Si j'ajoute que l'exécution d'un script Lua est seulement 20 fois plus lente que le C/C++ compilé, là où Perl et PHP sont 100 fois plus lents, peut-être souhaitez-vous lire la suite ? Enfin, si je vous affirme que beaucoup de non-développeurs peuvent l'utiliser réellement rapidement et qu'il est très lié au C, permettant ainsi de réaliser des interfaçages avec des bibliothèques existantes de manière très simple, sentez-vous une légère vibration d'intérêt ? Si c'est le cas, sachez que vous n'êtes pas le seul : Lua est ainsi utilisé au sein de nombreux outils et logiciels. 40 % d'Adobe Photoshop Lightroom sont écrits en Lua, VLC en a fait son langage de script interne, ainsi que Wireshark, pour ne nommer que des logiciels connus.

Lexmark l'utilise au sein de ses drivers d'imprimante, et le chip d'encodage Mobilygen MG3500, présent dans de nombreux caméscopes, se pilote entièrement grâce à une API en Lua. Ah, j'oubliais naturellement le jeu en ligne World of Warcraft, dont on peut paramétriser l'interface grâce à des scripts Lua. La suite de l'article vous donnera, je l'espère, l'envie de l'intégrer dans vos projets.

> Un langage portable, rapide, compact, facile à intégrer

Depuis le début de son existence, les auteurs de Lua se sont efforcés à ce que le langage soit :

- **Portable.** Lua est écrit en C en respectant scrupuleusement le standard ANSI C89. Il suffit d'avoir un compilateur C sur une plate-forme donnée pour pouvoir y utiliser Lua.
- **Rapide.** L'interpréteur standard est nettement plus rapide que des langages analogues tels que Ruby ou Python. Si la vitesse est importante, on peut même se tourner vers LuaJIT, encore bien plus rapide.
- **Compact.** La version actuelle Lua 5.2.1 complète occupe environ 200 Ko de code exécutable, y compris la bibliothèque standard.
- **Facile à intégrer.** Lua est conçu pour être intégré dans une application (par exemple un jeu) ou dans un micrologiciel. L'API C est simple et complète.
- **Puissant.** Malgré sa petite taille, Lua reste très puissant. Les tables, les *userdata* et les *métatables* associées sont particulièrem-



Extension Lua sous Eclipse

ment versatiles. Une *métatable* est un mécanisme particulier qui permet de redéfinir des opérateurs sur un objet.

- **Libre.** Non seulement le code source est disponible gratuitement, mais sa licence MIT autorise expressément son intégration dans tout type de projet y compris commercial.

Lua est modulaire dans son architecture. Son cœur contient l'interpréteur et normalement le compilateur. Mais il n'a aucune fonction prédéfinie ! Pour que Lua puisse servir à quelque chose, il est nécessaire de charger des *modules* dans l'interpréteur. Le chargement de chaque module est optionnel et facilement modifiable. L'écriture d'un *sandbox* de sécurité est dès lors assez facile en Lua, bien que non triviale. La distribution standard Lua 5.2 comporte dix modules : base (contenant des fonctions d'usage général), bit32, coroutine, debug, io, math, os, package, string et table. On les trouve préchargés dans la plupart des utilisations de Lua. Généralement, d'autres modules sont nécessaires dans un projet, qui peuvent provenir de la communauté Lua (comme LuaSocket ou LuaFileSystem), ou être propres à l'application ou au jeu vidéo.

> Déjà 20 ans...

Lua fêtera cette année ses 20 ans ! Il a été créé en 1993 à l'Université Catholique Pontificale de Rio de Janeiro pour répondre initialement aux besoins de la compagnie pétrolière brésilienne Petrobras. À cette époque-là, deux équipes de développement du département d'informatique travaillaient indépendamment sur l'écriture de deux petits langages de scripts DEL (Data Entry Language) et SOL (Simple Object Language). Les deux équipes se sont réunies et ont décidé de développer un langage commun, fusion de DEL et de SOL, nommé naturellement *Lua*. En effet, SOL signifiant *Soleil* en portugais, son successeur se devait d'être la Lune, *Lua* en portugais ! Notez que pour cette raison Lua se prononce *loua*, et s'écrit avec seulement la première lettre en majuscule ; ce n'est pas un acronyme.

Lua a un modèle de développement original : bien qu'il soit open source depuis 1994 (et la licence est MIT depuis la version 5.0), c'est toujours la même équipe de trois personnes qui décide des

[1] Au 18e rang en décembre 2012

[http://www.tiobe.com/index.php/content/paperinfo/tpci/index.html]

[2] Lua 5.2 Reference Manual, disponible sur le site officiel de Lua [http://www.lua.org].

fonctionnalités du langage : Roberto Ierusalimschy, Luiz Henrique de Figueiredo et Waldemar Celes. Elle écrit la version de référence de Lua en pur C et publie en même temps un *manuel de référence* [2], qui est à la fois une documentation et une spécification du langage. Les versions de Lua sortent à un rythme lent comparé à d'autres logiciels libres. La version actuelle, 5.2, date de 2011 alors que Lua 5.1 est sorti en 2006 et Lua 5.0 en 2003. Il s'agit de sorties majeures, stables, mais qui ne sont pas entièrement compatibles avec les versions précédentes.

> Quelle utilisation de Lua ?

Initialement destiné à des ingénieurs non-informaticiens, Lua a été conçu pour être très intuitif et simple à apprendre. Vous pouvez souhaiter utiliser Lua parce que vous travaillez sur une application qui l'intègre déjà (c'est par exemple le cas de nombreux intervenants dans le jeu vidéo) ou parce que vous voulez tirer parti de ses nombreux avantages et l'intégrer dans une application. À titre d'exemple, la société InPixel l'utilise comme moteur de traitement d'images : des algorithmes élémentaires écrits en C/C++ et compilés sous forme de bibliothèques sont ensuite chaînés et configurés via des scripts Lua, offrant ainsi une grande souplesse aux algorithmiciens pour valider rapidement des solutions de traitement d'images, tout en se libérant des problèmes « informatiques » traditionnels (allocation et libération mémoire, écriture de structures, etc.). Mieux : avec l'implémentation LuajIT, il est même possible de valider un algorithme travaillant au niveau pixel en Lua tout en obtenant des performances tout à fait honorables (similaires au C compilé sans optimisations).

Autre cas : le développement de jeux vidéo, qui est un secteur par nature très exigeant car il nécessite toujours plus de performances malgré la contrainte de la multiplicité des métiers utilisés pour une production de jeu. Lua s'est naturellement imposé dans un grand nombre de studios pour sa légèreté et sa facilité d'apprentissage. Par exemple, il sera utilisé pour définir une base de règles d'une intelligence artificielle qui décrit les actions possibles des personnages dans un contexte précis. Les scripts Lua permettent d'associer les contextes aux actions à réaliser. Il est ainsi facile de multiplier la variété des comportements des personnages dans le jeu.

```
function BarbarianBehavior()
  if IsAttacked() then
    MeleeAttack()
  else
    Walk()
  end
end

function MagusBehavior()
  if IsAttacked() then
    FireBall()
  else
    ProtectionShield()
  end
end
```

> Petit aperçu du langage

Entrons maintenant dans le cœur du langage. Lua est un langage *dynamiquement typé*, ce qui signifie qu'une variable n'a pas de type dédié. C'est la valeur contenue dans la variable qui possède un type.

Lua connaît au total huit types de valeurs :

nil : c'est le type de la valeur nil qui représente l'absence de données. *boolean* : une valeur booléenne qui peut être true ou false. Lors d'un test conditionnel, seuls false et nil sont considérés comme faux, toute autre valeur, y compris 0, étant vraie.

number : un nombre, qui peut être entier ou à virgule flottante, et qui est toujours représenté par un double en interne.

string : représente une chaîne de caractères non modifiable. Lua internalise les chaînes pour éviter d'avoir plusieurs copies identiques en mémoire. Une chaîne peut contenir n'importe quelle suite d'octets entre 0 et 255 ; on peut donc l'utiliser pour stocker par exemple le contenu d'un fichier binaire.

function : le type des fonctions. C'est une valeur de première classe qui peut être manipulée comme les autres. De plus, Lua supporte les *closures*.

table : c'est le type le plus versatile. Il s'agit d'une collection composée de clés et de valeurs associées, lesquelles peuvent être de tout type sauf nil. C'est en Lua la seule forme de structure de données. Elle permet notamment de représenter des tableaux, des dictionnaires, des tables de hachage, des listes, des modules. Elle est aussi utilisée en interne pour stocker les variables globales, et les métatables sont elles aussi de simples tables Lua.

userdata : vu de Lua, il s'agit d'une valeur opaque représentant une zone mémoire. Mais via l'API en C, un programmeur peut enregistrer n'importe quelle donnée dans un *userdata*, le plus souvent un simple pointeur. Comme une métatable peut être associée à chaque *userdata*, ce type est souvent utilisé pour exporter vers Lua l'interface d'un objet C++.

thread : c'est le type des coroutines. Une coroutine est similaire à un thread d'un système d'exploitation, mais est *collaborative* et non préemptive. Le support des coroutines en Lua n'utilise aucune fonction du système d'exploitation et est donc 100 % portable.

Comme la plupart des langages interprétés, l'exécution d'un script Lua se fait en deux phases distinctes. Lua va d'abord compiler le code source en *bytecode*, c'est-à-dire en instructions binaires pour sa machine virtuelle, puis interpréter les instructions dans la machine virtuelle. Il est possible de charger dans Lua du code précompilé, ce qui est surtout utile pour les projets embarqués dans des microcontrôleurs. Cela permet de diminuer le temps de démarrage du script, ou alors de se passer du compilateur de Lua, économisant ainsi environ 40 Ko de code.

> Quelques exemples

Les commentaires commencent par deux tirets comme dans SQL :

```
– Ceci est un commentaire sur une ligne
–[[ Et ceci est un commentaire dit long
  pouvant tenir sur plusieurs lignes ]]
→ est une convention représentant une sortie à l'écran
```

Le fameux *Hello World* est fort simple en Lua :

```
print("Hello World!") → Hello World!
```

Comme Lua est un langage dynamiquement typé, une variable n'a pas de type *a priori* et son contenu peut changer de type. Les variables sont globales par défaut.

```
value = «Une chaîne» – variable globale
print(value) → Une chaîne
```

```
value = 42 — réaffectation avec un type différent
print(value) → 42
```

Les variables locales sont déclarées avec le mot clé local et leur portée s'étend jusqu'à la fin du bloc où elles sont déclarées. L'exemple suivant réalise l'équivalent de la commande Unix cat :

```
while true do
  local line = io.read() — lit une ligne du fichier
  if line == nil then break end — EOF termine la boucle
  print(line) — chaque ligne du fichier est affichée
end
— line n'est plus visible ici
```

Cet exemple peut être simplifié en utilisant une boucle for générique et l'itérateur io.lines :

```
for line in io.lines() do
  print(line)
end
— line n'est plus visible ici
```

La fameuse fonction factorielle peut être écrite ainsi en utilisant la récursion :

```
function factorial(n)
  if n <= 1 then return 1 end — condition d'arrêt
  return n * factorial(n-1) — récursion
end
print(factorial(4)) → 24
```

Ou alors ainsi, de manière itérative avec une boucle for numérique :

```
function factorialIter(n)
  local prod = 1
  for i=2,n do — i va de 2 à n par pas de 1
    prod = prod * i
  end
  return prod
end
```

Une table se construit avec des accolades, en fournissant optionnellement des valeurs pour l'initialisation. L'exemple suivant utilise un tableau tab contenant cinq nombres et une fonction printTable pour l'afficher.

On trie ensuite la table d'abord dans l'ordre numérique croissant (comportement par défaut de table.sort), puis dans l'ordre croissant de la partie fractionnelle des nombres (c'est-à-dire leur valeur modulo 1).

Pour cela, on fournit une fonction anonyme comme argument à table.sort. Notez que la fonction printTable est une closure car elle fait référence à une variable locale déclarée à l'extérieur de celle-ci. Le support des closures et des fonctions anonymes fait que Lua peut être utilisé pour de la *programmation fonctionnelle*.

```
local tab = {3.7, 5.2, 1.8, 0.5, 2.0}
function printTable()
  — tab est une variable locale extérieure!
```

```
for i=1,#tab do — #tab est la longueur de tab (5)
  print(tab[i])
end
printTable() → 3.7 5.2 1.8 0.5 2
table.sort(tab)
printTable() → 0.5 1.8 2 3.7 5.2
table.sort(tab, function(a, b) return a%1 < b%1 end)
printTable() → 2 5.2 0.5 3.7 1.8
```

Les tables peuvent accepter n'importe quel type comme clé et valeur. Il y a néanmoins des raccourcis syntaxiques quand les clés sont des chaînes de caractères assimilables à un identifiant. L'opérateur . permet d'avoir une syntaxe pratique pour des structures ou des objets. On peut aussi écrire et appeler des *méthodes* d'un objet, soit des fonctions qui reçoivent automatiquement une référence nommée self sur l'objet courant. Il faut pour cela utiliser l'opérateur : au lieu de . :

```
s = { prenom="Roberto", nom="Terusalimschy", oeuvre="Lua" }
s.annee = 1993 — équivalent à s["annee"] = 1993
function s:print() — écriture d'une méthode de la table s
  print(self.nom, self["prenom"], self.annee, self.oeuvre)
end
s:print() → Terusalimschy Roberto 1993 Lua
```

Le dernier exemple montre l'utilisation d'une métatable. Normalement, l'addition de deux tableaux provoque une erreur à l'exécution car le comportement n'est pas défini. Mais en placant une fonction nommée __add dans une table particulière associée à une autre table, on peut notamment effectuer une addition de deux vecteurs.

```
local meta = { — métatable simple pour un vecteur
__add = function(a,b) — redéfinition de l'addition
  local res = {}
  for i=1,#a do
    res[i] = a[i] + b[i]
  end
  return res
end }
function printVector(v) — fonction affichant un vecteur
  print(table.concat(v, ", "))
end
local v1 = setmetatable({4, 2, 1}, meta)
local v2 = {3, 8, 0}
printVector(v1) → 4, 2, 1
printVector(v2) → 3, 8, 0
local v3 = v1 + v2 — appelle __add de la métatable de v1
printVector(v3) → 7, 10, 1
```

> Un cas concret...

SciTE est un éditeur de texte (orienté code source) intégrant Lua, qui permet de scripter des comportements (transformation de texte), d'écrire des *lexers* (coloration syntaxique) ou de gérer des petits dialogues. On peut développer toute une librairie de fonctions qui peuvent être appelées par menu ou par une combinaison de touches. On peut ainsi écrire des fonctions pour corriger le formatage d'une sélection, modifier des valeurs d'un *path* SVG, transformer une ligne

de code de façon plus sophistiquée qu'avec des expressions régulières. Par exemple, si vous n'aimez pas les parenthèses collées aux mots clés (les faisant ressembler à des appels de fonction), voici une fonction qui rectifie cela au sein du document ouvert (pour Java ou des langages similaires) :

```

function FixKeywords()
  local lineNb = editor.LineCount
  local line, sn, substNb
  -- Permet d'annuler toutes les modifications d'un coup
  editor:BeginUndoAction()
  -- Parcourt les lignes du document dans l'ordre inverse,
  -- en sautant la dernière ligne (vide).
  for i = lineNb - 2, 0, -1 do
    line = editor:GetLine(i)
    if line ~= nil then -- Si la ligne n'est pas nil
      -- Élimine (remplace avec rien), à l'aide
      -- d'une expression régulière (ER), les séparateurs
      -- de ligne et espaces de fin de ligne.
      line = string.gsub(line, «[\r\n]*$», «»)
      -- Si la ligne commence avec un mot clé nécessitant
      -- une parenthèse ouvrante, met un espace entre eux.
      -- ER : espaces en début de ligne, des lettres,
      -- suivies immédiatement par une parenthèse ouvrante.
      -- Noter l'usage du % comme échappement dans les ER.
      -- Une fonction Lua peut retourner plusieurs valeurs,
      -- ici la seconde valeur est le nombre de
      -- substitutions effectuées.
      line, substNb = string.gsub(line, «^( +)(%l+)%(%«,
        -- Ici, gsub prend en second paramètre, non pas
        -- une chaîne, mais une fonction anonyme prenant
        -- comme paramètres les captures de l'ER.
        function (s, kw) -- Indentation et mot clé
          -- Si le mot clé est un de ceux attendus...
          if kw == «if» or kw == «for» or kw == «while»
            or kw == «switch» or kw == «catch» then
              -- ... retourne l'indentation, le mot clé,
              -- et la parenthèse avec l'espace requis.
              return s .. kw .. « («
            else
              -- Sinon on restitue la chaîne passée sans
              -- changement (appel de fonction / méthode).
              return s .. kw .. « («
            end
          end)
        -- Si on a eu au moins une substitution
        if substNb > 0 then
          -- Remplacer la ligne courante avec sa version
          -- transformée (avec une fonction maison).
          ReplaceLine(i, line, false)
        end
      end
      -- Fin de l'agrégation d'actions pour le undo.
      editor:EndUndoAction()
    end
  end

```

Ce n'est qu'un petit aperçu des possibilités de script de Lua. En

matière de transformations de texte, une librairie comme LPEG (écrite par l'un des créateurs de Lua) permet une analyse syntaxique beaucoup plus fine et complexe.

> Un langage, des implémentations

Lua est conçu de telle sorte que rien n'empêche d'en écrire une implémentation parallèle en se basant sur le manuel de référence. Il en existe ainsi de nombreuses variantes qui s'exécutent notamment dans des machines virtuelles Java, .NET ainsi qu'en JavaScript. Si l'on y ajoute des *bindings* de Lua vers d'autres langages, Lua peut vraiment tourner n'importe où.

La plus utilisée est sans doute LuaJIT (<http://luajit.org>), écrite par Mike Pall. Cette version de Lua possède un interpréteur écrit en assembleur et surtout un compilateur à la volée d'une efficacité redoutable. LuaJIT 2.0 supporte les processeurs x86, x64, ARM, PowerPC et MIPS. LuaJIT est tellement rapide qu'il s'approche des performances du C et peut être utilisé pour faire du traitement de signal ou d'image ! Il est notamment utilisé dans GSL Shell, un logiciel de calcul comparable à MATLAB.

En outre, il existe des frameworks dédiés à des utilisations plus spécifiques. On peut citer notamment eLua (<http://www.eluaproject.net/>), qui offre une solution de développement embarqué en Lua sur des platesformes à partir de l'ARM Cortex M3 !

Ou encore Löve (aussi souvent appelé Love2D), qui offre une solution complète pour créer des jeux en 2D entièrement écrits en Lua, la librairie C sous-jacente fournissant l'interface portable de bas niveau avec le système. Dans le domaine du jeu toujours, on citera également LuaPlayer, permettant de coder des jeux pour PSP (*homebrew*) directement en Lua ou MicroLua, son équivalent pour la DS. De nombreux moteurs de jeu, qu'ils soient gratuits (Moai) ou payants (Corona) incluent Lua pour scripter des niveaux ou de l'intelligence artificielle, voire pour écrire les jeux eux-mêmes, de façon portable (les deux moteurs cités visent aussi les téléphones portables, iOS et Android).

> Pour aller plus loin

Depuis 20 ans, une communauté s'est spontanément développée (notamment autour de la liste de diffusion *lua-l*, en anglais), avec pour point d'entrée le site principal <http://www.lua.org>. À partir de là, vous retrouverez le manuel officiel, ainsi qu'un wiki et une liste d'ouvrages. Outre le fameux «PIL» (*Programming In Lua* de Roberto Ierusalimschy), qui en est à sa troisième édition, il est aussi fait mention des autres ouvrages disponibles, dont le récent *Le guide de Lua et ses applications* écrit par vos serviteurs.



Le Guide de Lua et ses applications

aux Éditions D-BookeR (<http://www.d-booker.fr>).

Cyril Doillon

Ingénieur en informatique spécialisé en «images, interaction et réalité virtuelle»
cyril.doillon@gmail.com

Sylvain Fabre

Ingénieur logiciel et matériel, co-fondateur d'InPixel (<http://www.inpixel.com>)

Philippe Lhoste

Ingénieur logiciel orienté application web/desktop

Patrick Rapin

Développeur logiciel chez Xample Sàrl

Création d'activités pour TeamBuild

La Build de Team Foundation Server s'appuie sur une technologie qui peut paraître complexe mais qui est très puissante. Voyons comment s'en servir pour étendre les capacités de TeamBuild.

Dépuis TFS 2010, la Build est orchestrée par Workflow Foundation, qui a remplacé MSBuild. MSBuild est toujours utilisé, mais seulement là où il est le meilleur : la compilation des solutions. Le processus de Build lui-même est séparé entre plusieurs acteurs : les contrôleurs et les agents. Les premiers servent à distribuer la compilation sur les deuxièmes. Agents et contrôleurs peuvent être installés sur des machines différentes, ce qui rend cette architecture très puissante et très flexible.

Un Workflow est composé de deux parties : une petite partie qui s'exécute sur le contrôleur et une autre sur l'agent. C'est généralement cette partie qui fait l'objet de modifications. Une Build pouvant être lancée pour plusieurs raisons (une à la fois), le processus se doit de gérer tous les cas. Il faut tenir compte de cela lors de la personnalisation des scripts. Voici les différentes raisons qu'une Build a de se lancer. Ce sont les différentes valeurs que peut prendre l'énumération BuildReason [Fig.1].

Des activités supplémentaires sont déjà disponibles sur Codeplex (<http://tfsbuildextensions.codeplex.com>) mais nous allons voir qu'il n'est pas difficile d'en créer.

> Processus de Build

Les Workflow de Build se trouvent dans le dossier « BuildProcessTemplates » à la racine de chaque TeamProject. Si vous n'avez jamais ajouté d'activités venant de dll externe vous pouvez l'ouvrir directement. Comme je le disais plus tôt, c'est un Workflow. Veillez toutefois à faire attention d'ouvrir le Workflow avec la version de Visual Studio qui coïncide avec celle de TFS.

Le Workflow de Build par défaut (partie agent) est séparé en plusieurs parties. Les thèmes suivants se dégagent :

- Initialisation
- Compilation
- Tests
- Publication des symboles

Il est possible de modifier le Workflow en ajoutant de nouvelles activités avec la ToolBox comme pour un Workflow standard. L'ajout de

nouveaux paramètres se fait également via l'onglet « Arguments » du Workflow. Si on souhaite éditer ce paramètre depuis les propriétés de notre Build, il nous suffit de les ajouter dans les « metadata ». Cette structure de donnée permet d'identifier quels sont les paramètres de la définition de Build et ceux qui peuvent être configurés à chaque lancement (les deux comportements sont possibles simultanément). Il n'y pas d'aide à la saisie de nom, donc il faut bien faire attention à bien écrire le nom de l'argument [Fig.2].

> Ajout d'activités

L'ajout d'activités pour TeamBuild est aussi simple que d'ajouter des activités pour un Workflow classique... à l'exception près, que nous n'avons pas de support de projet .NET pour ajouter de nouvelles références. Nous verrons plus bas comment résoudre ce problème. Toute activité n'est pas utilisable dans TeamBuild. Une activité doit contenir l'attribut `Microsoft.TeamFoundation.Build.Client.BuildActivityAttribute`. Cet attribut permet à TeamBuild de savoir si l'activité peut être chargée côté contrôleur ou côté agent.

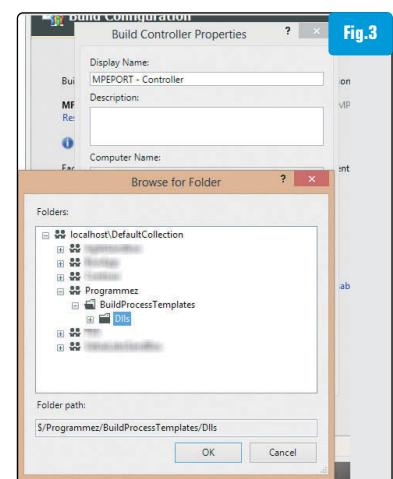
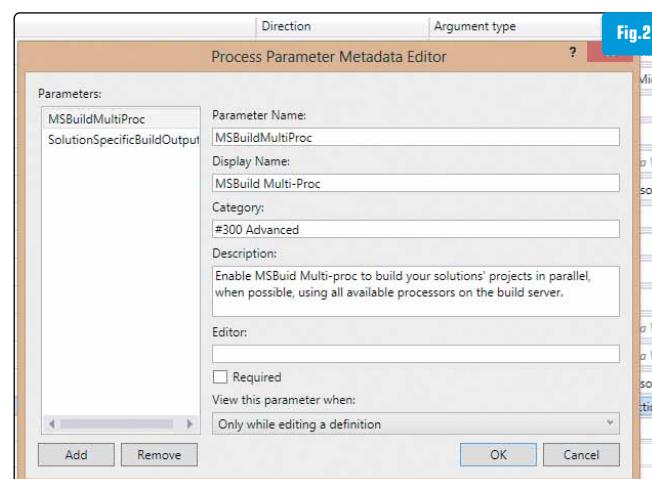
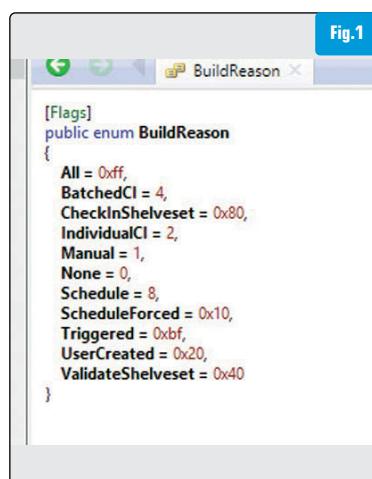
Comment référencer une nouvelle DLL d'activité dans TeamBuild ? C'est très simple, cela se joue du côté du contrôleur car le code lié à ces tâches personnalisées peut être exécuté à ce niveau-là [Fig.3]. Il suffit d'indiquer un répertoire, généralement un répertoire nommé « `CustomBuildActivities` » ou équivalent dans le dossier « `BuildProcessTemplates` ». Nous y stockerons les DLL de nos activités, par exemple celles de <http://tfsbuildextensions.codeplex.com>.

Créons maintenant un projet `ActivityEditor` de type `Workflow Library` (`Framework 4.5`) dans un sous-dossier « `Editor` » de `BuildProcessTemplates`.

Une fois le projet créé, il suffit de supprimer le ou les Workflows créés par défaut et ajouter via un lien celui de TeamBuild.

Nous pouvons maintenant ajouter les références aux assemblies de nos activités.

Dernière étape : ajouter les références aux assemblies de TFS (elles sont dans le GAC) et toutes celles que demande le Workflow. Maintenant le Workflow s'ouvre et il suffit d'ajouter des activités.



Attention à un détail : si vous ajoutez des variables dans des séquences, je vous conseille de bien nommer les séquences pour retrouver facilement la portée de vos variables.

Côté ajout de paramétrage, c'est très simple via l'onglet « Arguments ». Ceci est juste une procédure rapide pour éditer votre Workflow. Une procédure plus complète est disponible ici : http://tfsbuild_extensions.codeplex.com/wikipage?title=How%20to%20integrate%20the%20extensions%20into%20a%20build%20template&referringTitle=Documentation

> Ecriture d'activités simples

Sans rentrer dans les détails de Workflow Foundation, une activité à base de code est une classe qui hérite de `CodeActivity`. Ce genre d'activité doit être limité à des actions très courtes.

Voici une activité simple qui fait une addition et détaillons là ensuite :

```
[BuildActivity(HostEnvironmentOption.All)]
public sealed class AddActivity : CodeActivity
{
    [RequiredArgument]
    public InArgument<int> Value1 { get; set; }

    [RequiredArgument]
    public InArgument<int> Value2 { get; set; }

    [RequiredArgument]
    public OutArgument<int> Result { get; set; }

    protected override void Execute(CodeActivityContext context)
    {
        var value1 = context.GetValue(Value1);
        var value2 = context.GetValue(Value2);

        context.SetValue(Result, value1 + value2);
    }
}
```

> Ecriture d'activités asynchrones

Les activités asynchrones sont essentielles si vous avez des traitements longs. Car, contrairement aux activités synchrones, l'agent de Build a la possibilité d'annuler une activité asynchrone. Cela arrive par exemple si un utilisateur décide d'arrêter une Build. Dans ce cas, si l'activité est synchrone, l'agent part en timeout et une intervention sur la console d'administration de TFS est nécessaire: pas très user friendly!

L'écriture d'activités asynchrones nécessite 3 méthodes :

- `BeginExecute` : création de la tâche asynchrone
- `EndExecute` : récupération des données après l'exécution de la tâche
- `Cancel` : nettoyage optionnel du contexte

À vous donc de gérer le cas de l'annulation en envoyant le bon signal si nécessaire. Généralement l'écriture d'activités asynchrones s'appuie sur des API, elles-mêmes asynchrones, qui gèrent déjà l'annulation. Voici un exemple d'activité qui télécharge un fichier :

```
[BuildActivity(HostEnvironmentOption.All)]
public sealed class AsyncDownloadFile : AsyncCodeActivity
```

```
[RequiredArgument]
public InArgument<string> URL { get; set; }

[RequiredArgument]
public InArgument<string> FilePath { get; set; }

protected override IAsyncResult BeginExecute(AsyncCodeActivityContext context, AsyncCallback callback, object state)
{
    string url = context.GetValue(URL);
    var request = HttpWebRequest.Create(url) as HttpWebRequest;
    context.UserState = request;
    var result = request.BeginGetResponse(callback, state);

    return result;
}

protected override void EndExecute(AsyncCodeActivityContext context, IAsyncResult result)
{
    var request = context.UserState as HttpWebRequest;
    if (request != null)
    {
        var response = request.EndGetResponse(result) as HttpWebResponse;

        var lenght = response.ContentLength;
        if (lenght > 0)
        {
            var filePath = context.GetValue(FilePath);

            var filestream = File.OpenWrite(filePath);
            using (filestream)
            {
                var responseStream = response.GetResponseStream();

                responseStream.CopyTo(filestream);
                responseStream.Flush();
                responseStream.Close();
            }
        }
    }
}
```

> Ecriture d'activités composites

Une grande règle d'écriture de Workflow et donc en particulier de TeamBuild est la réutilisabilité. Même si l'interface a l'air simple, la technologie sous-jacente est complexe et l'on peut vite passer à côté d'erreurs qui peuvent ne pas être trouvées durant le développement. Je vais prendre exemple sur l'activité `InvokeProcess`. D'un coup d'œil rapide, l'activité pourrait ne contenir qu'un `Process.Start(...)`. Partons sur cette hypothèse et attachons nous à réaliser l'activité suivante qui consiste à exécuter NAnt (<http://nant.sourceforge.net>).

Nous avons plusieurs alternatives :

- Écrire une tâche synchrone : ce n'est pas une bonne idée, car le

processus peut prendre plusieurs minutes.

- Écrire une tâche asynchrone : au moment de l'annulation il suffira de tuer le processus de Nant : partons là-dessus.

Si NAnt est en train d'exécuter un autre processus et que nous annulons la Build, le processus de NAnt est détruit mais pas le processus qui a été lancé. La Build est certes arrêtée, mais en fait elle continue dans notre dos! Process.Start(...) n'est donc pas la solution. La solution est d'utiliser l'activité InvokeProcess. Plus que de lancer un processus, la tâche crée un Job dans lequel elle y met le processus qu'elle lance et les sous-processus qui pourraient apparaître ensuite. Lors de l'annulation, l'activité n'a plus qu'à détruire le job et les processus associés (ce qui se fait en une seule méthode de l'API des jobs).

InvokeProcess étant une activité, nous n'allons pas pouvoir l'utiliser comme une simple classe, il va falloir décrire en code le Workflow, au lieu de l'écrire en XAML. Cela peut paraître difficile, mais en fait avec de l'habitude, cela devient très facile à lire et l'on se rend compte surtout que c'est la manière la plus efficace d'écrire une activité. La plupart des activités de TeamBuild sont écrites de cette manière. Plus précisément : l'activité composite contient parfois de toutes petites CodeActivity qui sont utilisées dans son Workflow. Contrairement aux activités de code, nous allons implémenter une autre méthode. Cette méthode « CreateBody » va renvoyer le Workflow sous-jacent.

Voyons le code de cette activité qui se trouve sur Codeplex :

<http://tfsbuildextensions.codeplex.com> :

```
[BuildActivity(HostEnvironmentOption.All)]
[ActivityTracking(ActivityTrackingOption.ActivityOnly)]
public class InvokeNAnt : BaseActivity
{
    [Category("Parameters")]
    [RequiredArgument]
    public InArgument<ExecutionParameters> Parameters { get; set; }

    [Category("Configuration")]
    [DefaultValue(false)]
    public InArgument<bool> Verbose { get; set; }

    [Category("Configuration")]
    [DefaultValue(false)]
    public InArgument<bool> IgnoreExitCode { get; set; }

    [Category("Configuration")]
    [RequiredArgument]
    public InArgument<string> NAntDirectory { get; set; }

    [Category("Configuration")]
    public InArgument<string> WorkingDirectory { get; set; }

    [Category("Result")]
    public InArgument<string> LogFilePath { get; set; }

    protected override System.Activities.Activity CreateInternal
Body()
{
    var commandLine = new Variable<string>();
    var nantPath = new Variable<string>();
    var parameters = new Variable<ExecutionParameters>();
    ...
}
```

```
var exitCode = new Variable<int>();
var configError = new Variable<bool>();
var result = new Sequence
{
    Variables = { commandLine, parameters, nantPath, exitCode,
    configError }
};

result.Append(new List<Activity>
{
    new Assign<bool>
    {
        To = configError,
        Value = new InArgument<bool>(context => false)
    },
    new Assign<ExecutionParameters>
    {
        To = parameters,
        Value = new InArgument<ExecutionParameters>(context
=> context.GetValue(this.Parameters))
    },
    new If(context => parameters.Get(context) == null)
    {
        Then = new Sequence().Append(new List<Activity>
        {
            new Assign<bool>
            {
                To = configError,
                Value = new InArgument<bool>(context => true)
            },
            new WriteBuildError
            {
                Message = "parameters argument is null"
            }
        }),
        Else = new Sequence().Append(new List<Activity>
        {
            new If(context => !parameters.Get(context).BuildFile
PathExists)
            {
                Then = new Sequence().Append(new List<Activity>
                {
                    new Assign<bool>
                    {
                        To = configError,
                        Value = new InArgument<bool>(context => true)
                    },
                    new WriteBuildError
                    {
                        Message = "Build file does not exist"
                    }
                })
            }
        })
    },
    new Assign<string>
    {
        To = nantPath,
        Value = new InArgument<string>(context => Path.Combine
        ...
    }
});
```

```

(ProcessNantPath(context.GetValue(this.NAntDirectory)), «Nant.exe»))
},
new If(context => !File.Exists(nantPath.Get(context)))
{
    Then = new Sequence().Append(new List<Activity>
    {
        new Assign<bool>
        {
            To = configError,
            Value = new InArgument<bool>(context => true)
        },
        new WriteBuildError
        {
            Message = new InArgument<string>(context => string.
Format(`Nant not found: '{0}'`, nantPath.Get(context))),
        }
    })
},
new Assign<string>
{
    To = commandLine,
    Value = new InArgument<string>(context =>
        context
            .GetValue(this.Parameters)
            .CreateCommandLine(context.GetValue(this.LogFile
Path)))
},
new If(context => !configError.Get(context))
{
    Then = new Sequence().Append(new List<Activity>
    {
        new WriteBuildMessage
        {
            Message = new InArgument<string>(context =>
string.Format(`Command line: '{0}'`, commandLine.Get(context))),
            Importance = new InArgument<BuildMessageImportance>
(context => BuildMessageImportance.Normal)
        },
        new InvokeProcess
        {
            Arguments = commandLine,
            FileName = nantPath,
            Result = exitCode,
            WorkingDirectory = this.WorkingDirectory,
        },
        new If(context => !this.IgnoreExitCode.Get(context)
&& exitCode.Get(context) != 0)
        {
            Then = new Sequence().Append(new List<Activity>
            {
                new WriteBuildError
                {
                    Message = new InArgument<string>(context =>
string.Format(`Nant execution failed with exitcode '{0}'`, exit
Code.Get(context))),
                },
                new Throw
                {

```

```

                    Exception = new InArgument<Exception>(context =>
new NAntException())
                }
            })
        },
        Else = new Throw
        {
            Exception = new InArgument<Exception>(context =>
new NAntException())
        }
    });
    return result;
}

/// <summary>
/// Replaces %env% placeholders by environment variable values
/// </summary>
/// <param name=>input</param>string containing placeholders
to replace</param>
/// <returns>the input script with placeholder removed
</returns>
private static string ProcessNantPath(string input)
{
    if (string.IsNullOrWhiteSpace(input))
    {
        return string.Empty;
    }
    return Environment.ExpandEnvironmentVariables(input);
}
}

```

L'attribut ActivityTracking permet à TeamBuild de savoir s'il doit afficher les détails du Workflow généré par l'activité ou non. Généralement c'est non, car nous voulons voir l'activité comme une tâche atomique du point de vue du log. Pour envoyer des messages sur la sortie de Build, il suffit d'utiliser les activités de base de TeamBuild comme WriteBuildMessage et WriteBuildError.

> Conclusion

Ecrire des activités pour TeamBuild n'est pas en soi difficile. Ce qui est difficile est de bien comprendre la logique de TeamBuild et en particulier de WorkFlow Foundation pour savoir où injecter les activités et de bien les coder pour avoir les comportements voulus même en cas d'exception.

Chaque type d'activité a son utilisation, il suffit de bien choisir dès le départ. Se dire que l'on va faire une CodeActivity dans un premier temps et ensuite passer en asynchrone est une mauvaise idée : vous allez devoir tout refaire. Le compromis est dans l'activité composite : écriture de toutes petites activités synchrones utilisées dans un Workflow qui lui est, par nature asynchrone, ce qui vous évite de gérer tous ces problèmes liés à l'asynchronisme et l'annulation.



Michel Perfetti
 Manager - MVP Visual Studio ALM chez Cellenza
 Cellenza - Software Development Done Right
 Blog : <http://www.buildmeimfamous.net>
 Twitter : @miiitch

Développer facilement vos API web avec APISpark

Une API web est un ensemble de ressources accessibles sur le Web via des URI et exposant des opérations sous forme de méthodes HTTP standards, ainsi qu'un ensemble de représentations définissant la structure des messages échangés. Elle permet notamment de travailler sur les données, indépendamment de leur format, comme JSON, XML ou YAML, grâce à la négociation de contenu HTTP. Les API web sont intimement liées aux concepts de REST, le style d'architecture du Web.

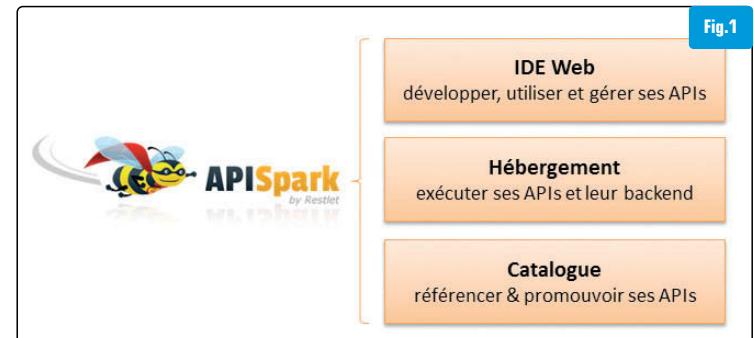
APISpark est une plateforme Cloud (PaaS) permettant de créer, d'héberger et de gérer rapidement et simplement des API web en mettant à disposition un environnement de développement intégré en ligne (IDE Web) ainsi qu'un service d'hébergement. Elle permet de définir des API web en se basant sur les concepts de REST. Son IHM guide le développeur afin d'exposer des ressources et des traitements derrière une URI. La puissance de l'outil réside dans le fait qu'aucune ligne de code n'est nécessaire puisque les interactions entre les API web et leur backend peuvent être configurées visuellement.

APISpark est actuellement disponible en version bêta à l'adresse <http://apispark.com>. La [Fig.1] fournit une vue générale d'APISpark.

Les apports d'APISpark sont tout d'abord son aptitude à mettre en oeuvre simplement et rapidement des API web pour des personnes n'ayant pas le temps ou les connaissances en programmation suffisantes pour obtenir le même résultat d'eux-mêmes en s'appuyant sur des briques open source. APISpark permet ainsi de développer une nouvelle génération de sites web dits cross-canaux, accessibles de façon homogène par tout type de machine (navigateur web, application native, programme informatique ou encore objet connecté).

APISpark s'appuie sur le concept de cellule, suivant la métaphore de la ruche. Une cellule est un module logiciel de base pouvant interagir avec d'autres cellules importées. Trois types de cellules principaux sont proposés comme l'illustre la [Fig.2].

Le type de cellule **Custom API** correspond à une API web métier accessible une fois déployée via le protocole HTTP et exposant des

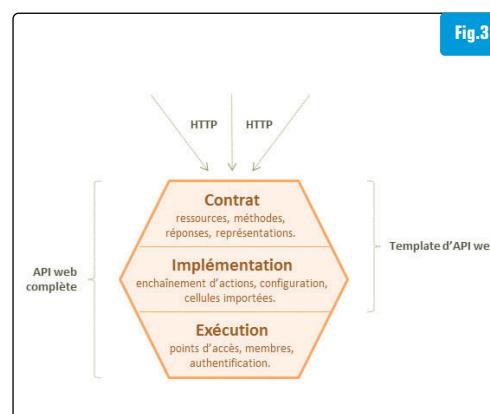
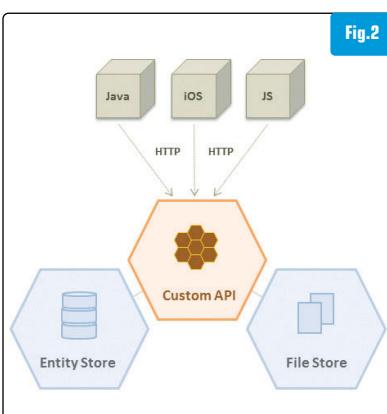


données. Elle peut pour cela s'appuyer sur les deux autres types de cellules: l'**Entity Store** pour les données structurées (équivalent d'une base de données) et le **File Store** pour les fichiers statiques (documents textuels, image, vidéos et autres feuilles de style). Maintenant que nous avons introduit APISpark ainsi que les possibilités qu'offre l'outil, abordons un exemple concret : l'implémentation puis l'utilisation d'une API web de gestion de contacts.

API WEB DE GESTION DES CONTACTS

Tout au long de cet article, nous allons illustrer comment développer une API web afin de gérer une base de contacts. Cette API offrira la possibilité de lister, créer, modifier et supprimer les contacts. Les API web dans APISpark suivent la structuration décrite dans la [Fig.3]. Tout d'abord, le contrat public correspond aux ressources mises à disposition par l'API aux utilisateurs. Pour notre API, nous allons définir plusieurs ressources pour gérer nos contacts ainsi que leurs éléments applicatifs associés (fichiers HTML, CSS, etc.) :

- Ressource de gestion des contacts, avec le chemin `/contacts/`. Elle permet de lister les contacts avec sa méthode GET et d'en ajouter avec sa méthode POST.
- Ressource de gestion d'un contact particulier, avec le chemin `/contacts/{contactid}`. `{contactid}` est une variable et correspond à l'identifiant du contact. Elle permet de récupérer un contact en particulier avec sa méthode GET et de le modifier ou supprimer avec respectivement ses méthodes PUT et DELETE.
- Ressource de gestion des fichiers du répertoire HTML avec le che-



min `/html/{file}`. Elle permet de retourner avec sa méthode GET le contenu du fichier, d'en ajouter un ou de le modifier avec sa méthode PUT, de le supprimer avec sa méthode DELETE.

- Ressource de gestion des fichiers du répertoire CSS avec le chemin `/css/{file}`. Elle permet de retourner avec sa méthode GET le contenu du fichier, d'en ajouter un ou de le modifier avec sa méthode PUT, de le supprimer avec sa méthode DELETE.

Les deux premières ressources s'appuient sur deux représentations de contacts, l'une pour un seul élément appelé Contact et l'autre pour les listes de contacts. La première contient les propriétés d'un contact, telles que son identifiant, ses nom et prénom ainsi que son âge. La seconde appelée ContactList correspond à une liste de contacts basée sur la représentation précédente. Ces deux représentations sont utilisées aussi bien en entrée qu'en sortie au niveau des méthodes des ressources.

D'autres représentations sont également utilisées pour les répertoires et fichiers.

Dans APISpark, une API ne met pas en oeuvre elle-même tous les traitements mais en délègue la plupart à d'autres types de cellules. Dans notre cas, il s'agit des data stores, un pour les données structurées (base de donnée) et un pour les fichiers statiques.

Commençons par la base de données.

CRÉATION DE LA BASE DE DONNÉES

Une API web permet souvent d'exposer et de modifier des informations structurées, stockées dans une base de données de type relationnel ou NoSQL. C'est également le cas pour notre API web de gestion de contacts. Dans APISpark, le type de cellule apportant cette fonctionnalité est appelé l'**Entity Store**. Il offre un entrepôt de données structurées dédié.

Une fois notre *entity store* créé (nommé **“Contacts DB”**), il suffit de lui définir une entité **Contact** regroupant les différentes informations utilisables pour un contact. Une entité est équivalente à une table dans une base relationnelle. Pour rester simple, nous définirons uniquement un identifiant, un prénom, un nom ainsi qu'un âge. La [Fig.4]. illustre cette définition dans l'outil.

Au niveau des propriétés de l'entité, il est possible de définir s'il s'agit d'une clé primaire, les cardinalités ainsi que le type, comme le montre la [Fig.5].

Le dernier aspect d'une entité à considérer

Properties

- id**
 - Name: id
 - Datatype: String
 - Min occurs: 1
 - Max occurs: 1
 - Primary key:
- nom**
- prenom**
- age**
 - Name: age
 - Datatype: Integer
 - Min occurs: 1
 - Max occurs: 1
 - Primary key:

Add a property

Fig.5

consiste dans les opérations utilisables. Bien qu'il soit possible de les définir manuellement, des méthodes par défaut sont ajoutées automatiquement lors de la création de l'entité. Ces méthodes permettent de gérer l'entité afin de réaliser notamment des opérations CRUD (Create Retrieve Update Delete) sur les données. La [Fig.6] décrit le détail d'une de ces opérations.

Une fois ces différents éléments définis, il suffit de déployer cet **Entity Store** afin de le rendre opérationnel et utilisable par notre API web de gestion des contacts. Cela se réalise simplement en cliquant sur le bouton *Deploy*. Il est à noter qu'un **Entity Store** n'est utilisable qu'au sein de la plateforme APISpark. Pour rendre accessible ses données de manière distante, il convient de passer par l'intermédiaire d'une API web (voire de plusieurs!).

Avant de nous attaquer à notre API web à proprement parler, nous allons créer un autre type d'entrepôt de données, le **File Store**, afin d'y stocker quelques fichiers statiques.

CRÉATION DE L'ENTREPÔT DE FICHIERS

Une API web peut également exposer des fichiers statiques. Dans le cas de notre API web de gestion de contacts, il peut s'agir de fichiers HTML, JavaScript et CSS afin d'afficher les données via un appel AJAX. Dans APISpark, le type de cellule prenant en charge cet aspect est appelé **File Store**. Il s'agit d'un entrepôt de fichiers organisé en répertoires, un peu comme les buckets S3 d'Amazon Web Services.

Une fois, notre *file store* créé (nommé **“Contacts Files”**), nous allons lui ajouter deux répertoires **HTML** et **CSS**, respectivement pour stocker les pages web/AJAX et les feuilles de style. La [Fig.7] illustre cette définition dans l'outil.

A l'instar des entités pour les **Entity Stores**, des méthodes sont

Entities

- Contact
 - Method: Overview
 - Conditions

Method: Overview Conditions

Name: SELECT

Description: Retourne la liste des contacts

Parameters

Add a parameter

Fig.6

Contact Files Data Store - File

Overview | Settings | Members | Messages

Actions Version 1 Draft Deploy

Folders

- HTML
 - Method: Overview Browser Security
 - Name: HTML
 - Description: Correspond au répertoire de stockage des fichiers HTML et JavaScript

Fig.7

Folders

- HTML
 - Method: Overview
 - Name: LIST
 - Description: Affiche la liste des fichiers du répertoire

Fig.8

automatiquement ajoutées aux répertoires afin de gérer leur contenu. Il s'agit dans ce cas d'opérations de listage du contenu et de lecture, modification et suppression d'éléments. La [Fig.8] décrit le détail d'une de ces opérations.

Une fois les différents répertoires définis, il suffit de déployer ce **File Store** afin de le rendre opérationnel et utilisable par notre API web de gestion des contacts. Cela se réalise simplement en cliquant sur le bouton *Deploy*. Comme pour l'**Entity Store**, il convient de l'utiliser par l'intermédiaire d'une API web (ou de plusieurs) pour rendre accessibles ses données.

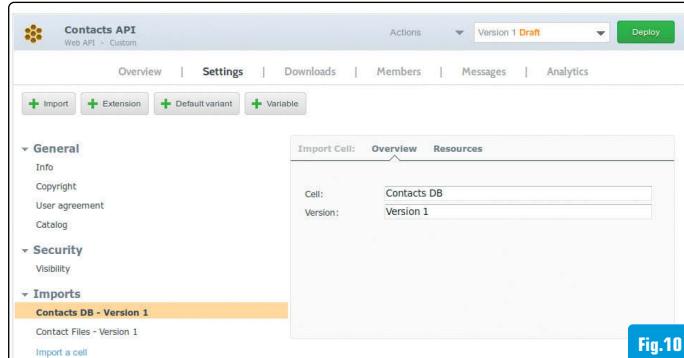
Lorsque le store est déployé, il est possible d'afficher directement la liste des fichiers contenus et d'en télécharger d'autres depuis la console graphique d'APISpark. Nous allons ainsi ajouter quelques fichiers statiques afin d'afficher nos contacts dans une page HTML via une requête AJAX avec jQuery. Le fichier suivant illustre le contenu de la page HTML. Il est également possible d'enrichir l'aspect de la page en ajoutant une feuille de style.

```
<html>
  <head>
    <link rel="stylesheet" type="text/css" media="screen" href="/v1/css/styles.css" />
    <script type="text/javascript" src="/v1/html/jquery-min.js"></script>
    <script type="text/javascript">
$(document).ready(function() {
  $.getJSON("/v1/contacts/", function(data) {
    var content = "";
    for (var i=0; i<data.list.length; i++) {
      var contact = data.list[i];
      content += "<li>" + contact.nom + " " + contact.prenom + " (" + contact.age + ")" + "</li>";
    }
    $("#contacts").html(content);
  });
});
  </head>
<body>
  <h2>Liste des contacts</h2>
  <ul id="contacts"></ul>
</body>
</html>
```

Nous allons maintenant utiliser l'onglet **Browser** détaillant les répertoires de notre store. Il permet de lister les fichiers présents et éventuellement d'en ajouter d'autres via des téléchargements. La [Fig.9] décrit comment télécharger notre fichier *contacts.html* précédent. Maintenant que nous avons défini le backend composé de deux **Data Stores**, nous pouvons nous attaquer à son exposition via l'API web à proprement parler.

CRÉATION DE L'API WEB

La mise en oeuvre de notre API web de gestion des contacts est relativement simple avec APISpark. En effet, nous allons pouvoir tirer



parti de son aptitude à créer automatiquement des ressources au sein d'une API web pour exposer des cellules dépendantes. Bien que cela puisse apparaître abstrait à ce niveau, nous allons nous efforcer de détailler l'utilisation de ces fonctionnalités tout au long de cette section.

Après avoir créé notre API web nommée **Contacts API**, nous allons importer en tant que dépendances les deux **Data Stores** précédemment créés via l'onglet **Settings**. Il est à noter que vous pouvez rattacher les cellules pour lesquelles vous êtes défini(e) en tant que membre et qui apparaissent donc dans votre tableau de bord. La [Fig.10] montre l'importation de ces deux stores dans l'API.

L'étape suivante consiste alors en la génération des ressources au sein de l'API afin de rendre accessible respectivement la gestion des données des contacts ainsi que des fichiers statiques. Pour ce faire, il suffit de se rendre dans l'onglet **Resources** des propriétés des dépendances et d'ajouter les ressources correspondantes grâce au bouton **Add resources**.

Les différentes ressources ajoutées ainsi que les représentations associées s'affichent alors dans la liste de l'onglet, ainsi que l'illustre la [Fig.11] relative aux propriétés de l'**Entity Store**.

Une fois cette opération également réalisée pour le File Store, les différentes ressources et représentations suivantes ont été obtenues dans l'API web.

Resources	Representations
▶ /contacts/	▶ Contact
▶ /contacts/{contactid}	▶ ContactList
▶ /html/	▶ FolderListRepresentation
▶ /html/{file}	▶ FolderRepresentation
▶ /css/	▶ HtmlFileRepresentation
▶ /css/{file}	▶ CssFileRepresentation

Toutes ces ressources et représentations sont correctement configurées pour utiliser les stores précédemment importés. Il est à noter que nous avons supprimé les ressources inutiles "/html/" et "/css/" servant à lister les fichiers des répertoires.

Comme vous pouvez le voir [Fig.12], les différentes ressources et représentations ont été ajoutées dans les sections **Resources** et

Import Cell: Overview Resources
Dependent resources
/contacts/
/contacts/{contactid}
Dependent representations
ContactList
Contact
Actions
Add resources
Update resources
Delete resources

Fig.11

Endpoints	Actions
contactsapi.apispark.net	Method: Overview Actions
Resources	
▶ /contacts/	Name: GET
▶ GET Retourne la liste des contacts	Description: Retourne la liste des contacts
▶ POST Ajoute un contact	
▶ Add a method	
▶ /contacts/{contactid}	
▶ /html/{file}	
▶ /css/{file}	
▶ Add a resource	
Representations	
▶ Contact	
▶ ContactList	
▶ FolderListRepresentation	
▶ FolderRepresentation	
▶ HtmlFileRepresentation	
▶ CssFileRepresentation	
▶ Add a representation	

Fig.12

Representations. Regardons un peu plus en détail les ressources avec l'URI /contacts. Deux méthodes sont présentes :

La première, la méthode GET, retourne la liste des contacts en se basant sur les représentations **Contact** et **ContactList**.

La seconde, la méthode POST, permet d'ajouter un contact en se basant sur la représentation **Contact**.

Comme vous pouvez le voir, les méthodes exposent un traitement via HTTP et configurent ce qui est accepté en entrée et retourné en réponse. Une ou plusieurs actions sont également configurables afin de définir quels traitements sont réalisés lors de l'appel. Dans le cas de la méthode GET, l'action correspond à un appel de l'entity store et plus particulièrement de la méthode SELECT de l'entité Contact afin de récupérer la liste des contacts. Cette liste sera formatée dans la représentation ContactList pour être retournée.

Les représentations définissent une structure de données échangées mais ne sont pas liées à un format particulier. En effet, APISpark supporte la négociation de contenu afin d'utiliser JSON, XML ou autres en fonction des besoins. Cet aspect se configure au niveau de l'API et des représentations en se basant sur les concepts d'extension et de variantes.

Il nous reste juste à déployer l'API web afin qu'elle devienne accessible depuis internet à travers le domaine apispark.net. L'adresse complète d'accès à l'API se base sur le nom de domaine spécifié lors de sa création. Si les méthodes des ressources sont sécurisées, il convient d'utiliser l'identifiant et le mot de passe accessibles dans les propriétés du point d'accès de l'API (endpoint), comme l'illustre la [Fig.13]. Nous avons désormais fini la mise en place de notre API web de gestion des contacts et allons maintenant décrire la manière d'accéder et de l'utiliser par différents moyens.

UTILISATION DE L'API CÔTÉ CLIENT

L'utilisation une API web n'impose en aucun cas APISpark ni même un langage de programmation particulier. Cela peut même se faire depuis un navigateur. Néanmoins pour tester votre API, nous vous recommandons l'utilisation d'outils tels que RESTClient

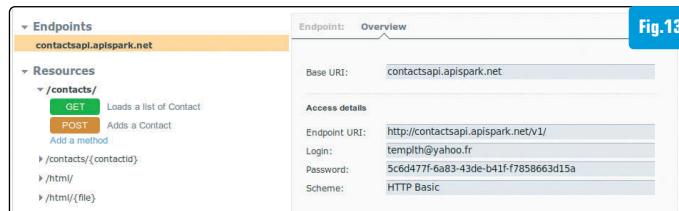


Fig.13

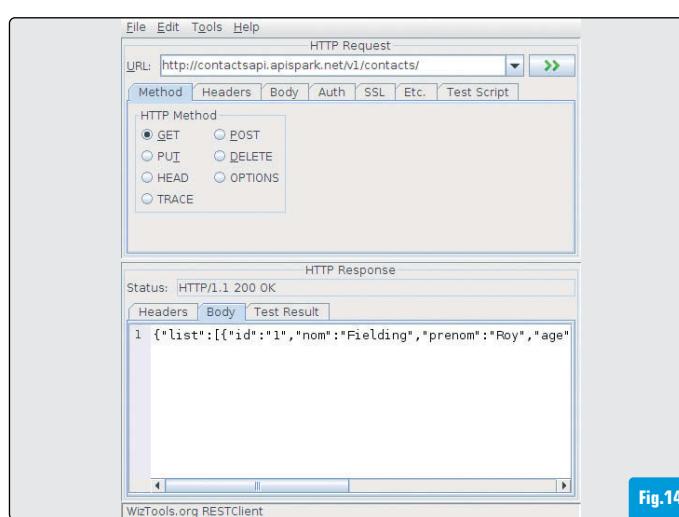


Fig.14

(<http://code.google.com/p/rest-client/>) offrant une interface graphique pour exécuter des appels HTTP. La [Fig.14] illustre l'appel de l'API de contacts avec cet outil. Il est également possible d'utiliser les fichiers HTML, JavaScript et CSS stockés précédemment dans le File Store. Il suffit pour cela d'utiliser l'URL <http://contactsapi.apispark.net/v1/html/contacts.html>. Il est à noter qu'APISpark fournit également des kits client pour différents cas d'utilisation de l'API tels qu'avec Java (JVM ou Android), JavaScript (AJAX ou Node.js) ou Objective-C (iOS). Enfin, le service d'hébergement collecte automatiquement des statistiques d'accès à vos API et propose des rapports visuels comme l'illustre la [Fig.15].

CONCLUSION

APISpark est une plateforme cloud (PaaS) mettant à disposition un IDE web productif afin de créer des API web et les backends associés, ainsi qu'un service d'hébergement intégré afin de les mettre à disposition sur Internet. Dans cet article, nous avons montré comment développer une API web de gestion de contacts avec APISpark, exposant des données structurées et des fichiers statiques.

Nous avons vu comment créer une API web à partir d'un modèle de données en laissant l'outil créer les ressources et représentations correspondantes. L'outil reste néanmoins ouvert pour des besoins plus avancés ou spécifiques en proposant une configuration fine au niveau des ressources (URI, représentations et implémentation). Nous avons également vu que le déploiement de l'API se réalise en un seul clic sur un bouton *Deploy*. Vous n'avez plus à vous soucier des problématiques de disponibilité et de scalabilité.

Nous n'avons pu vous décrire dans cet article que quelques fonctionnalités de la plateforme. Cette dernière met également à disposition un catalogue d'API pour la découverte et la promotion en leur ajoutant un niveau social. Il est possible de suivre les API ainsi que les relations entre elles. L'outil supporte également les templates afin de créer des API instantanément ou encore les contrats d'API faciles à partager entre plusieurs implémentations.

Thierry Templier (ttemplier@restlet.com)

Architecte R&D, Restlet SAS

Développeur, Restlet Framework

Co-auteur des ouvrages *Restlet in Action* (Manning), *Spring par la pratique et JavaScript pour le Web 2.0* (Eyrolles).

Jérôme Louvel (jlouvel@restlet.com)

Directeur Technique, Restlet SAS

Créateur et Leader Technique, Restlet Framework

Co-auteur de l'ouvrage *Restlet in Action* (Manning) et contributeur à l'ouvrage *RESTful Web Services* (O'Reilly)

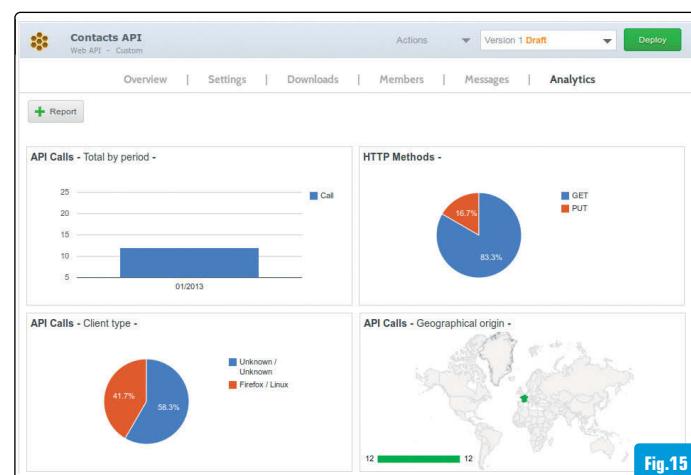


Fig.15

Développer une application avec MVVM

Vous souhaitez appliquer le pattern MVVM à vos applications Windows 8 ou tout simplement en apprendre plus ce fameux pattern d'architecture ? Voici un guide pratique pour développer vos applications Windows Store avec MVVM.

Utiliser le pattern MVVM, c'est avant tout séparer les couches de son application de façon à les rendre faiblement couplées. Pour les vues, cela se caractérise notamment par l'absence (ou, en tout cas, la diminution) du code behind.

View. Les vues correspondent aux pages et écrans de vos applications et contiennent la définition structurée de vos différents écrans, c'est-à-dire le XAML. La vue correspond à ce que l'utilisateur verra en lançant votre application.

Model. Les modèles correspondent aux classes représentant vos données du côté client. Le modèle peut se charger notamment de la validation des données.

ViewModel. Comme son nom l'indique, le ViewModel se situe entre la vue et le modèle. Le ViewModel regroupe les données consommées par la vue et contient la logique de manipulation du modèle. Généralement, on crée un ViewModel par vue.

RAPPEL SUR LE DATABINDING

Avant d'entrer dans le vif du sujet, voici un rapide rappel sur le binding en XAML. Il est indispensable de bien comprendre cette notion pour utiliser le pattern MVVM dans n'importe quelle application. Si vous n'avez aucune notion de binding, je vous invite à consulter un tutoriel avant de continuer cet article.

Le binding est défini dans le XAML et permet de lier dynamiquement la valeur d'un élément (le texte d'un bouton par exemple) à celle d'une propriété (une string qui contiendra le texte du bouton). Un binding peut être unidirectionnel ou bidirectionnel.

Dans le second cas, cela signifie qu'une modification de la valeur de l'élément bindé modifie la propriété source, et qu'une modification de la valeur de la propriété source modifie la valeur de l'élément sur lequel le binding est effectué. Pour qu'un binding soit effectif, il est nécessaire que l'élément bindé sache où trouver la propriété source. Cette propriété est soit récupérée dans la source du binding, si celle-ci est spécifiée, soit dans le DataContext de l'élément bindé ou de ses ancêtres. Chaque composant héritant de FrameworkElement contient une propriété nommée DataContext. Une fois défini sur un élément, le DataContext se répand dans l'ensemble de son visual tree. Ainsi, si un DataContext est défini pour une Grid, tous les éléments situés dans cette Grid utiliseront par défaut ce DataContext. En MVVM, on définit généralement le DataContext de chaque objet Page en lui associant son ViewModel correspondant. Ainsi, tous les binding effectués dans la vue ont comme source un ViewModel. Avec MVVM, le binding représente le système de communication entre une vue et un ViewModel. Cela signifie que les ViewModel définissent un ensemble de propriétés sur lesquels les vues vont se binder.

CRÉATION D'UN MODÈLE

Pour notre application de films, il nous faut avant tout un modèle permettant de représenter des films. Pour faire simple, nous allons partir du principe qu'un film est représenté par un ID, un titre et une image. Le modèle sera ici un simple POCO.

```
public class Movie
{
    public int ID { get; set; }
    public string Title { get; set; }
    public string Image { get; set; }
}
```

Pour la suite de l'article, nous admettrons qu'un repository nommé MovieRepository nous permet de récupérer une liste de films. Un lien disponible à la fin de l'article vous permettra de récupérer les sources de l'application.

CRÉATION D'UN PREMIER VIEWMODEL

Le premier ViewModel que nous allons créer sera celui que l'on associera par la suite à la page principale de l'application. Celle-ci permettra de lister les différents films. Le ViewModel doit donc notamment contenir une liste de films.

```
public class MainPageViewModel
{
    public MainPageViewModel()
    {
        this.Movies = MovieRepository.GetMovies();
    }

    private ObservableCollection<Movie> _movies;
    public ObservableCollection<Movie> Movies
    {
        get { return _movies; }
        set
        {
            if (_movies != value)
            {
                _movies = value;
            }
        }
    }
}
```

Lors de sa construction, le ViewModel récupère les films depuis le repository et les affecte à la propriété Movies.

CRÉATION D'UNE PREMIÈRE VUE

Cette première vue permet de présenter une liste de films. Cette liste sera affichée dans une GridView. La propriété Movies du ViewModel sera bindée sur la propriété ItemsSource de la GridView.

```
<Grid>
<Grid.ColumnDefinitions>
    <ColumnDefinition Width=>Auto</ColumnDefinition>
    <ColumnDefinition Width=>*</ColumnDefinition>
</Grid.ColumnDefinitions>
```



```

<Button x:Name=>backButton Click=>GoBack IsEnabled=>{Binding
Frame.CanGoBack, ElementName=pageRoot} Style=>{StaticResource
BackButtonStyle}>/>
<TextBlock x:Name=>pageTitle Grid.Column=>1 Text=>{Static
Resource AppName} Style=>{StaticResource PageHeaderTextStyle}>/>
</Grid>

<GridView ItemsSource=>{Binding Movies}>
    Padding=>120,0,0,0
    Grid.Row=>1>
    <GridView.ItemTemplate>
        <DataTemplate>
            <StackPanel Orientation=>Horizontal>
                <Image Source=>{Binding Image} Width=>120 Height=>
150 Stretch=>UniformToFill />
                <TextBlock Text=>{Binding Title}>
                    Style=>{StaticResource SubheaderTextStyle}
                    TextWrapping=>Wrap
                    Margin=>15,5
                    Width=>250 />
                </StackPanel>
            </DataTemplate>
        </GridView.ItemTemplate>
    </GridView>

```

Nous avons donc les trois éléments du pattern MVVM : la vue, le modèle et le ViewModel. Toutefois, pour l'instant, si vous exécutiez l'application, vous constaterez que la vue n'affiche rien. En effet, le ViewModel MainPageViewModel n'a pas été défini en tant que DataContext de la vue. Cela pourrait se faire simplement depuis le code behind, comme le montre l'exemple ci-dessous :

```
this.DataContext = new MainPageViewModel();
```

Cependant, nous n'allons pas procéder de la sorte. J'expliquerai cela en détail dans la section ViewModelLocator qui suit.

ET SI MES PROPRIÉTÉS CHANGENT, JE FAIS QUOI ?

Lorsque les valeurs des propriétés d'un ViewModel changent, il est nécessaire d'en avertir la vue. Cela se fait grâce à l'interface INotifyPropertyChanged. Cette interface définit un événement nommé PropertyChanged. Cet événement prend notamment en paramètre un objet de type PropertyChangedEventArgs permettant de représenter une propriété ayant été modifiée. Concrètement, dès qu'une propriété du ViewModel est modifiée, il est nécessaire de lever l'événement PropertyChanged afin d'en notifier la vue. Cette dernière se mettra alors à jour. Puisque tous les ViewModel ont besoin de ce système, nous allons créer une classe nommée ViewModelBase, dont hériteront tous les ViewModel.

```

public class ViewModelBase : INotifyPropertyChanged
{
    public event PropertyChangedEventHandler PropertyChanged;

    protected void RaisePropertyChanged([CallerMemberName] string
propertyName = null)
    {
        if (PropertyChanged != null)
    }

```

```

        PropertyChanged(this, new PropertyChangedEventArgs(propertyName));
    }
}
}
```

Dorénavant, il suffira d'appeler la méthode RaisePropertyChanged dans le setter de chaque propriété du ViewModel. Chaque propriété ... ou presque. Pour les collections cela est légèrement différent. Celles-ci ne doivent pas implémenter INotifyPropertyChanged, mais INotifyCollectionChanged. La classe ObservableCollection<T>, utilisée ci-dessus pour définir la liste de films, implémente déjà cette interface. Ainsi, dès qu'un élément est ajouté ou supprimé de cette liste, la vue est automatiquement mise à jour.

Gardez toujours à l'esprit que la vue doit être notifiée pour que les données qu'elle présente soient à jour.

VIEWMODELLOCATOR

Le ViewModelLocator, comme son nom l'indique, va permettre de localiser les ViewModel. Ce dernier est notamment utilisé pour éviter le couplage fort vue/ViewModel, comme le montrait l'initialisation du DataContext dans le code behind ci-dessus. Le ViewModelLocator va mettre à disposition des vues différents ViewModel sans que chacun ait pour autant une référence vers son ViewModel. Le ViewModelLocator a pour charge de construire les ViewModel qui seront affectés en tant que DataContext des vues. On distingue principalement deux façons de créer un ViewModelLocator. La première consiste à créer une classe contenant une propriété pour chaque ViewModel. Chaque vue peut ensuite binder son DataContext à son ViewModel en utilisant le nom de cette propriété. Voici un exemple de ViewModelLocator :

```

public class ViewModelLocator
{
    private static ViewModelBase _mainPageViewModel;
    public static ViewModelBase MainPageViewModel
    {
        get
        {
            if (_mainPageViewModel == null)
            {
                _mainPageViewModel = new MainPageViewModel();
            }
            return _mainPageViewModel;
        }
    }

    public static ViewModelBase AddMoviePageViewModel
    {
        get
        {
            return new AddMoviePageViewModel();
        }
    }
}
```

Remarquez que le premier ViewModel est unique, tandis que le second (que nous utiliserons plus tard) est instancié à chaque appel.

Le binding se fait ensuite de la façon suivante :

```

<common:LayoutAwarePage
  x:Name=>pageRoot
  x:Class=>MvvmW8.View.MainPage
  xmlns=>http://schemas.microsoft.com/winfx/2006/xaml/presentation
  xmlns:x=>http://schemas.microsoft.com/winfx/2006/xaml
  xmlns:local=>using:MvvmW8.Mvvm
  DataContext=>{Binding MainPageViewModel, Source={StaticResource
  VMLocator}}}
  mc:Ignorable=>d>

  <Page.Resources>
    <local:ViewModelLocator x:Key=>VMLocator</local:ViewModelLocator>
  </Page.Resources>

```

Cette première solution permet également de définir facilement un ViewModel à utiliser au runtime et un autre durant la conception de l'application. En fonction de la valeur de la propriété DesignMode.DesignModeEnabled, il est simple de créer un ViewModel contenant des données statiques à utiliser pendant la création de l'application, et donc affichable directement sur le designer Visual Studio / Blend. La deuxième solution pour créer un ViewModelLocator est d'utiliser une attached property. Dans ce cas, le nom du ViewModel sera récupéré dynamiquement en fonction du nom de la vue à laquelle il est associé. Il est ainsi nécessaire de définir une convention de nommage pour les ViewModel. Dans l'exemple suivant, le nom de chaque ViewModel correspond au nom de chaque vue auquel est préfixé « ViewModel ».

```

public class ViewModelLocator
{
  public static bool GetUseDefaultViewModel(DependencyObject obj)
  {
    return (bool) obj.GetValue(UseDefaultViewModelProperty);
  }

  public static void SetUseDefaultViewModel(DependencyObject obj,
    bool value)
  {
    obj.SetValue(UseDefaultViewModelProperty, value);
  }

  public static readonly DependencyProperty UseDefaultViewModel
  Property =
    DependencyProperty.RegisterAttached(<UseDefaultViewModel>,
    typeof(bool), typeof(ViewModelLocator), new PropertyMetadata
    (null, UseDefaultViewModelChanged));

  private static void UseDefaultViewModelChanged(DependencyObject
  d, DependencyPropertyChangedEventArgs e)
  {
    string viewName = d.GetType().Name;
    string viewModelName = String.Concat(viewName, <ViewModel>);

    Type viewModelType = Type.GetType(<MvvmW8.ViewModel> + viewModel
    Name);
    object newViewModel = Activator.CreateInstance(viewModelType);
  }
}

```

```

    FrameworkElement view = (FrameworkElement)d;
    view.DataContext = newViewModel;
  }
}

```

Il faut ensuite utiliser cette attached property de la façon suivante :

```

<common:LayoutAwarePage
  x:Name=>pageRoot
  x:Class=>MvvmW8.View.MainPage
  xmlns=>http://schemas.microsoft.com/winfx/2006/xaml/presentation
  xmlns:x=>http://schemas.microsoft.com/winfx/2006/xaml
  xmlns:local=>using:MvvmW8.Mvvm
  local:ViewModelLocator.UseDefaultViewModel=>True
  mc:Ignorable=>d>

```

COMMANDES

MVVM apporte son propre mécanisme d'interaction avec les contrôles grâce aux commandes. Désormais, lorsqu'il faudra répondre au clic d'un utilisateur sur un bouton, ce n'est pas un événement qui sera levé, mais une commande qui sera exécutée. Cette commande s'occupera alors d'exécuter un certain bout de code en réponse à l'action de l'utilisateur. Il n'existe pas nativement de classe permettant de représenter une commande ; il est nécessaire d'en créer une. Pour être utilisée en tant que commande, cette classe doit obligatoirement implémenter l'interface ICommand. Celle-ci définit une méthode nommée Execute ; cette méthode est appelée lorsque la commande est exécutée. ICommand contient également une propriété nommée CanExecute et un événement CanExecuteChanged. La propriété permet de savoir si la commande peut être exécutée, tandis que l'événement déclenche une notification indiquant le changement d'état de CanExecute. Voici une implémentation simplifiée d'une classe représentant une commande.

```

public class RelayCommand : ICommand
{
  private Action _action;

  public RelayCommand(Action action)
  {
    _action = handler;
  }

  public bool CanExecute(object parameter)
  {
    throw new NotImplementedException();
  }

  public event EventHandler CanExecuteChanged;

  public void Execute(object parameter)
  {
    if (_action != null)
      _action ();
  }
}

```

Le constructeur de cette classe prend en paramètre une action qui sera exécutée lors de l'exécution même de la commande. Cela per-

met d'avoir une classe « générique » et non une commande par action à réaliser. Nous allons maintenant créer une commande dans MainPageViewModel afin de naviguer vers une nouvelle page depuis MainPage lors du clic sur un bouton. Celle-ci se définit comme une simple propriété :

```
public MainPageViewModel()
{
    this.Movies = MovieRepository.GetMovies();

    _addMovieCommand = new RelayCommand(AddMovie);
}

private ICommand _addMovieCommand;
public ICommand AddMovieCommand
{
    get
    {
        return _addMovieCommand;
    }
}

private void AddMovie()
{
    //
}
```

Il reste maintenant à utiliser cette commande depuis la vue. Exemple avec un bouton présent dans une AppBar.

```
<Page.BottomAppBar>
    <AppBar>
        <StackPanel Orientation="Horizontal" HorizontalAlignment="Right">
            <Button Command="{Binding AddMovieCommand}" Style="{Static
Resource AddAppBarButtonStyle}"> />
        </StackPanel>
    </AppBar>
</Page.BottomAppBar>
```

En cliquant sur ce bouton, la méthode Execute de la classe RelayCommand sera appelée, ce qui provoquera l'appel de la méthode privée AddMovie.

NAVIGATION ENTRE LES PAGES

En MVVM se pose toujours le problème de la navigation entre les différentes pages de l'application et le passage des paramètres entre celles-ci. Pour rappel, avec Windows 8, la navigation se fait grâce aux frames. L'objet frame possède notamment une méthode Navigate prenant en paramètre le type de la page vers laquelle on souhaite naviguer. Toujours dans le but de découpler au maximum les différentes parties de l'application, nous allons créer un helper, nommé NavigationService, qui mettra à disposition des ViewModel différentes méthodes pour naviguer. Ce helper ne sera finalement ni plus ni moins qu'un wrapper pour la classe Frame.

En voici une version simplifiée :

```
public class NavigationService
{
    private static Frame _currentFrame;
```

```
static NavigationService()
{
    CreateOrGetCurrentFrame();
}

public static void Navigate(string pageName)
{
    Type pageType = Type.GetType(`MvvmW8.View.` + pageName);
    _currentFrame.Navigate(pageType);
}

public static void Navigate(string pageName, object parameter)
{
    Type pageType = Type.GetType(`MvvmW8.View.` + pageName);
    _currentFrame.Navigate(pageType, parameter);
}

private static void CreateOrGetCurrentFrame()
{
    if (_currentFrame == null)
    {
        if (Window.Current.Content != null && Window.Current.Content
is Frame)
        {
            _currentFrame = Window.Current.Content as Frame;
        }
        else
        {
            _currentFrame = new Frame();
            Window.Current.Content = _currentFrame;
            Window.Current.Activate();
        }
    }
}
```

L'utilisation est la suivante :

```
private void AddMovie()
{
    NavigationService.Navigate(`AddMoviePage`);
}
```

Notez que les méthodes de navigation ne prennent pas directement un type en paramètre, mais une chaîne de caractères. Une fois encore, cela nous permettra de ne pas coupler nos VM avec nos vues.

RÉCUPÉRER LES PARAMÈTRES

Admettons maintenant que nous souhaitons afficher les détails d'un film en cliquant dessus. Pour cela, nous allons rediriger l'utilisateur vers une nouvelle page en passant en paramètre le film sélectionné. Avant d'implémenter cela, il est important de se rappeler que les applications Windows Store possèdent généralement un bouton back (situé en haut à gauche des pages) permettant de remonter l'historique de navigation de l'utilisateur. Lorsque l'utilisateur clique sur le bouton back, les paramètres passés à chaque page sont resoumis et donc accessibles dans les méthodes OnNavigatedTo et LoadState (cette dernière méthode peut être surchargée si votre page hérite de LayoutAwarePage). Il s'agit là de la seule solution

pour récupérer ces paramètres. Nous allons donc être obligé d'écrire un peu de code behind afin de récupérer le paramètre. Une fois ce dernier récupéré, il est nécessaire de renvoyer la donnée en question vers le ViewModel correspondant à la vue en cours d'affichage. Pour simplifier cela, nous allons créer une méthode virtuelle nommée `LoadData` dans `ViewModelBase`. Cette méthode prendra un paramètre de type `object` et sera publique.

```
public virtual void LoadData(object parameter)
{
}
```

Admettons maintenant que lors de la sélection d'un film, la méthode suivante soit exécutée :

```
private void GoToMovieDetail()
{
    NavigationService.Navigate("MovieDetailPage", this.SelectedMovie);
}
```

Pour récupérer le film sélectionné en arrivant sur `MovieDetailPage`, nous pourrions utiliser le code suivant :

```
protected override void LoadState(Object navigationParameter,
Dictionary<String, Object> pageState)
{
    ViewModelBase vm = this.DataContext as ViewModelBase;
    vm.LoadData(navigationParameter);
}
```

Dans `MovieDetailPageViewModel`, le code suivant est présent :

```
public override void LoadData(object parameter)
{
    if (parameter is Movie)
    {
        //
    }
}
```

En faisant cela, la vue et son ViewModel restent faiblement couplés. Ici, l'idéal est de créer une classe permettant de gérer cela automatiquement pour toutes les vues.

BEHAVIORS

Dans une partie précédente, nous avons étudié la notion de commande. Cela fonctionne parfaitement avec les boutons, mais comment faire pour réagir à un autre évènement sans utiliser le code behind des vues ? Il faut passer par des behaviors.

Un behavior est une classe permettant d'ajouter des fonctionnalités à un contrôle existant. L'un des plus utilisés en MVVM est le behavior `EventToCommand` permettant d'exécuter une commande quand un certain évènement est levé. En WPF, un behavior Blend est présent pour réaliser ceci. Malheureusement, il n'existe aucun behavior Blend pour Winrt, il est donc nécessaire de les recoder. Pour nous simplifier le travail, un développeur a mis à disposition sur NuGet un package nommé `winrtbehaviors` permettant d'utiliser des behaviors dans les applis WinRT. Autant dire que ce package est rapidement indispensable pour tous ceux qui veulent créer une application Windows 8 MVVM. Sur Internet se trouve également un behavior `EventToCommand` pour WinRT (effectuez une recherche « A WinRt behavior to mimic EventToCommand »). Celui-ci utilise les Reactive

Extensions pour fonctionner. Il permet d'imiter le comportement du behavior Blend du même nom. Une fois les différents packages ajoutés à votre projet, ce behavior s'utilise de la façon suivante.

```
<common:LayoutAwarePage
    x:Name="pageRoot"
    x:Class="MvvmW8.View.MainPage"
    xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
    xmlns:winrtb="using:WinRtBehaviors"
    xmlns:behaviors="using:MvvmW8.Behaviors"
    DataContext="{Binding MainPageViewModel, Source={StaticResource VMLocator}}"
    mc:Ignorable="d">

    <GridView ItemsSource="{Binding Movies}">
        <winrtb:Interaction.Behaviors>
            <behaviors:EventToCommandBehavior Command="GoToMovieDetailCommand" Event="SelectionChanged" />
        </winrtb:Interaction.Behaviors>
    </GridView>

```

Dans cet exemple, dès qu'un film est sélectionné, la commande `GoToMovieDetailCommand` est exécutée.

MESSENGERS

Dernière notion fondamentale de MVVM, les messengers. Ces derniers permettent d'établir un canal de communication entre différents ViewModel (mais pas que) tout en gardant ces derniers découpés. Le principe est simple : le destinataire du message s'enregistre afin de recevoir un type de message donné et l'expéditeur envoie des messages d'un certain type. Les objets échangés peuvent être simples (string, entier), comme complexes.

Imaginons que, dans notre application, nous ajoutions une page pour créer un film. Une fois ce film créé, nous voulons notifier `MainPageViewModel` afin que ce film soit ajouté à la collection `Movies`.

Dans `AddMoviePageViewModel` :

```
private void AddMovie()
{
    Movie newMovie = new Movie();
    Messenger.Instance.Notify<Movie>("newMovie", newMovie);
}
```

Dans `MainPageViewModel` :

```
Messenger.Instance.Register<Movie>("newMovie", (m) => this.Movies.Insert(0, m));
```

Ici, `MainPageViewModel` s'enregistre pour recevoir des données de type `Movie` avec la clé « `newMovie` » et `AddMoviePageViewModel` notifie ce dernier.

CONCLUSION

MVVM n'est pas simple à appréhender. Toutefois, ce pattern d'architecture apporte un réel plus une fois qu'il est maîtrisé, notamment en termes de maintenabilité, testabilité et évolutivité de votre code. Pour conclure, je vous invite à vous rendre sur mon blog, dont l'adresse est disponible ci-dessous, afin de récupérer les sources de l'application m'ayant servi d'exemple tout au long de cet article.

Loïc Rebours

Consultant .NET, Avanade - Microsoft MVP C#
www.blog.loicrebours.fr

Découvrir l'outil de construction de projets ANT

ANT est un outil de construction de projets qui, s'il est au départ dédié aux projets Java, excelle pour tous types de projets et se fait remarquer par sa portabilité. Découvrons ensemble les principes de son maniement.

ANT est apparu au cours de l'an 2000. Il était alors intégré au serveur de Servlets Tomcat et servait uniquement à la construction de ce dernier. Très vite, les participants à de nombreux projets Open Source Java ont pris conscience du potentiel de ANT. S'appuyant sur XML, extensible au moyen de classes Java, ANT préserve en outre des désagréments rencontrés avec d'autres outils de construction, notamment au niveau de la portabilité. Les IDE Java modernes sont entièrement basés sur ANT, c'est le cas de NetBeans, ou l'intègrent, c'est le cas d'Eclipse. Si ANT est très utilisé dans l'univers de Java, il est intéressant de le considérer comme une solution valable pour tous types de projets, et même pour des projets très complexes ou très spécifiques, en raison de la facilité avec laquelle il est extensible.

1 INSTALLATION DE ANT

ANT en version binaire peut être téléchargé à <http://ant.apache.org/> et vient sous la forme d'une simple archive zip. L'installation de ANT consiste simplement à décompresser cette archive. Vous obtenez alors une arborescence de répertoires dans laquelle le répertoire bin contient les scripts de lancement de ANT. Il convient donc de faire pointer ce répertoire par la variable d'environnement PATH de votre système. En plus de PATH, vous devez définir deux autres variables d'environnement, soit finalement:

Variables d'environnement	Pointe sur	Exemple
PATH	Les scripts du répertoire bin	C:\apache-ant-1.8.4\bin
ANT_HOME	La localisation de ANT sur votre système	C:\apache-ant-1.8.4\
JAVA_HOME	La localisation de la JVM sur votre système	C:\jdk1.6.0_10

Pour vérifier votre installation, vous pouvez faire depuis une console :

```
ant -version
```

Pour construire un projet avec ANT il suffit de taper ant dans la console. L'outil cherchera alors à lire dans le répertoire courant le fichier build.xml contenant les déclarations pour la construction du projet. Si l'on souhaite utiliser un fichier XML d'un autre nom, on emploiera les commutateurs -f ou -file ou -buildfile.

2 ANATOMIE D'UN FICHIER DE CONSTRUCTION

Un fichier de construction ANT est un document XML. A la racine du projet se trouve fort logiquement une balise 'project'. Cette balise racine définit, via ses attributs, un nom de projet, une cible (target) par défaut, un répertoire de travail de base et une description. Les descriptions, que ce soit pour le projet ou d'autres balises, sont des commentaires facultatifs. ces commentaires peuvent être imprimés sur la console par ant -projecthelp.

La construction d'un projet se fait par accomplissement d'une suite d'étapes élémentaires appelées targets. Des cibles de construction autrement dit. L'ordre des étapes est défini par des dépendances entre cibles au moyen de l'attribut 'depends'. A l'intérieur des cibles sont effectuées, par ordre d'apparition, des tâches que l'on peut voir comme des appels de fonctions successifs. Les paramètres pour la tâche sont là encore définis grâce aux attributs de balises. Une cible peut concerner la compilation de sources, bien évidemment, mais n'est pas limitée à cela en raison de la richesse des tâches de ANT. Une tâche est une classe Java instanciée et exécutée par la JVM sur ordre de ANT. Il y a des tâches pour tout faire: compiler, émettre des messages, créer un répertoire, l'effacer, etc. Si la tâche dont vous avez besoin n'existe pas, vous pouvez écrire vous même une classe Java pour la créer. Nous verrons plus loin comment procéder. Ce concept de tâches, outre l'extensibilité, apporte la portabilité. Ainsi si les commandes mkdir existent sous UNIX, Linux, Windows, leurs syntaxes et arguments diffèrent. La tâche Java/ANT mkdir gomme cela. Pour en terminer avec l'anatomie d'un fichier de construction, signalons les propriétés (property) qui sont à ANT ce que sont les variables globales aux langages de programmation.

3 HELLO WORLD !

Faisons nous la main en sacrifiant à la tradition et demandons à ANT de nous dire coucou puisqu'il sait faire autre chose que compiler du code Java. Voici le fichier coucou.xml que l'on peut qualifier de Projet ANT minimal. Vous trouverez ce fichier ainsi que tous les autres de cet article sur notre site www.programmez.com.

```
<?xml version="1.0" encoding="ISO-8859-15" ?>
<project name="Coucou anté" default="coucou" basedir=".">
  <description>Programme ANT de type Hello World</description>
  <target name="coucou" description="Dire coucou">
    <echo message="Programmez! avec ANT" />
  </target>
</project>
```

La tâche echo incluse dans la cible "coucou" sera automatiquement exécutée, car notre cible coucou est automatiquement construite en tant que cible par défaut. Lancez la construction du projet comme ceci :

```
ant -f coucou.xml
```

Le projet se construit, avec pour seule action d'émettre un message sur votre console.

4 UN PROJET SIMPLE

Construisons maintenant un vrai projet avec cibles multiples et dépendances entre elles. Notre projet est constitué de deux fichiers Java dont le code, sans intérêt, n'est pas donné ici. Nous définis-

sons trois cibles de construction très classiques: compilation, nettoyage et reconstruction. Voici le fichier build.xml

```
<?xml version="1.0" encoding="ISO-8859-15"?>
<project name="Démonstration ANT N°1" default="compile" basedir=".">
<description>
    Un fichier build.xml de base
</description>

<target name="compile" description="Compiler tous les sources Java.">
    <mkdir dir="classes" />
    <javac srcdir="src" destdir="classes" />
</target>

<target name="clean" description="Effacer toutes les classes">
    <delete dir="classes" />
</target>

<target name="rebuild" depends="clean,compile"
       description="Effacer et recompiler" />
</project>
```

Analysons ce code. Par défaut, la compilation des sources Java est effectuée. La cible 'compile' s'acquitte de cela. Nous considérons que les sources résident dans le répertoire 'src' et les classes seront déposées dans le répertoire 'classes'. Ces deux répertoires étant relatifs aux répertoires définis par 'basedir' dans le projet. Le répertoire courant dans notre cas. Notons la présence, avant la tâche javac, de la tâche mkdir, car le répertoire 'classes' n'est pas créé par la tâche javac. Nous devons donc nous assurer de son existence. La cible clean efface tout le répertoire 'classes' et enfin la cible de reconstruction [rebuild] est déclarée comme dépendant des cibles clean et compile, dans cet ordre. Ainsi :

```
ant -f build.xml rebuild
```

par construction de cette cible rebuild, provoque d'un seul coup le nettoyage du projet puis sa recompilation complète.

5 UN PROJET PLUS COMPLET

Un projet plus complet va comporter au moins une cible pour lancer un test, en plus de la construction du projet [Fig.1]. Et aussi une cible pour construire une distribution de l'application. Enfin il est pratique de définir des variables au début du fichier, afin qu'une modification de celles-ci se répercute sur tout le projet :

```
<?xml version="1.0" encoding="ISO-8859-15"?>
<project name="Démonstration ANT N°2" default="compile" basedir=".">
<description>
    Un fichier build.xml plus évolué
</description>

<property name="classes.dir" location="classes" />
<property name="programmeur.name" value="Fred" />

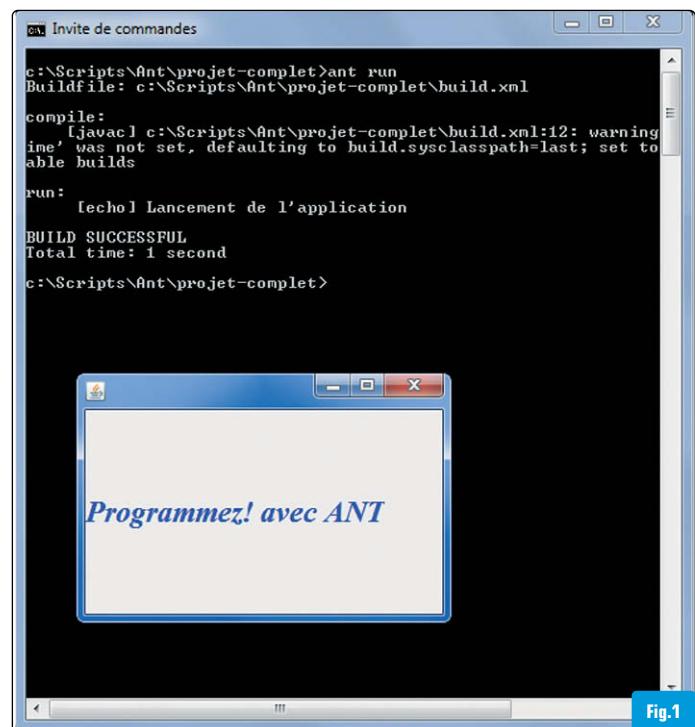
<target name="compile" description="Compiler tous les sources Java.">
    <mkdir dir="${classes.dir}" />
    <javac srcdir="src" destdir="${classes.dir}" />
</target>
```

```
<target name="clean"
       description="Effacer tous les fichiers .class">
    <delete dir="${classes.dir}" />
</target>

<target name="rebuild" depends="clean,compile"
       description="Effacer et recompiler" />

<target name="run" depends="compile" description="Exécuter le
programme">
    <echo message="Lancement de l'application"/>
    <java classname="programmez.fred.antdemo.MainClass"
          fork="true"
          spawn="true">
        <classpath location="classes"/>
        <arg value="Programmez!"/>
    </java>
</target>

<target name="makedist"
       depends="compile"
       description="Construire une distribution dans un Jar">
    <mkdir dir="dist" />
    <jar destfile="dist/antdemo.jar">
        <fileset dir="classes"/>
        <manifest>
            <attribute name="Created-By" value="${programmeur.name}" />
            <attribute name="Main-Class"
value="programmez.fred.antdemo.MainClass" />
        </manifest>
    </jar>
</target>
```



Une cible permet de lancer notre application dès que sa construction est terminée.

```
</project>
```

Pour les cibles de test et de création d'une distribution, nous utilisons des tâches ANT. Il n'y a aucune difficulté, nous renvoyons si besoin le lecteur à la documentation de ces tâches. Bien remarquer comment les "variables" sont définies de moyen de balises XML 'property' et comment les valeurs ainsi définies sont utilisées dans les cibles clean et compile.

6 CRÉER DES TÂCHES

Voyons maintenant comment étendre ANT avec de nouvelles tâches. A priori n'importe quelle classe Java peut devenir une tâche ANT, ce qui ouvre un champ de possibilités infinies. Voici une classe conforme à la vieille tradition du HelloWorld :

```
package programmez.fred.antdemo;

public class CoucouTask {
    public void execute() {
        System.out.println("Coucou à la tâche");
    }
}
```

Nous voyons que notre classe comporte une méthode baptisée conventionnellement 'execute'. ANT recherche toujours l'existence de cette méthode via le mécanisme de réflexion de Java. Parallèlement nous écrivons un fichier build.xml (fichier complet sur notre site) pour compiler tous les sources Java des exemples et les archiver dans un jar. Pour tester notre nouvelle tâche, il suffit d'ajouter une cible au fichier de construction. ANT n'étant pas devin, nous l'informons que notre classe est une tâche ANT au moyen d'une balise taskdef. Voici ce que cela donne :

```
<target name="coucouteest"
description="Test de CoucouTask"
depends="anttaskjar">
<taskdef name="coucou"
classname="programmez.fred.antdemo.CoucouTask"
classpath="${dist.dir}/anttask.jar" />
<coucou/>
</target>
```

Notre nouvelle tâche, pertinemment baptisée coucou est exécutée immédiatement après sa définition. Cette dernière comporte le nom qualifié de la classe et un classpath indiquant où la trouver, c'est-à-dire dans notre archive Jar.

7 UNE TÂCHE PLUS ÉVOLUÉE

La question qui se pose maintenant est de savoir comment procéder pour que notre tâche affiche non plus un message prédéfini mais plutôt un message défini dans un attribut de balise du fichier build.xml. Ou, formulé autrement, comment lui passer des données et des paramètres ? Nous voudrions aussi que notre tâche coopère mieux avec ANT. La librairie de ANT fournit la classe org.apache.tools.ant.Task pour cela et c'est d'elle que dériveront désormais nos nouvelles tâches. La tâche que nous créons maintenant est semblable à la tâche intégrée 'echo'. La tâche reçoit un

message défini soit dans un attribut de sa balise, soit dans le contenu de sa balise. Le message est imprimé sur la sortie standard via la méthode log de la classe parent Task. Passer par log est plus dans la philosophie ANT que System.out.println. En outre, un booléen peut être défini dans un attribut de la balise afin de définir le comportement de la tâche dans le cas où le message n'est pas défini. Soit on émet un message d'information, soit on lève une exception, ce qui aura pour effet d'interrompre la construction du projet. Voici notre nouvelle classe :

```
package programmez.fred.antdemo;

import org.apache.tools.ant.Task;
import org.apache.tools.ant.BuildException;

public class MyAntTask extends Task {

    public void setMessage(String message) {
        this.message = message;
    }

    public void addText(String message) {
        this.message = message;
    }

    public void setFailOnError(boolean failOnError) {
        this.failOnError = failOnError;
    }

    public void execute() {
        if(message != null) {
            String projet = getProject().getProperty("ant.project.name");
            log(projet + " : " + message);
        }
        else {
            if(failOnError)
                throw new BuildException(
                    "Exception levée: pas de message défini");
            else
                log("Pas de message défini");
        }
    }

    private String message;
    private boolean failOnError;
}
```

Et nous pouvons tester notre tâche comme ceci depuis le fichier build.xml :

```
<target name="myanttasktest" description="Test de MyAntTask"
depends="anttaskjar">
<taskdef name="myanttask"
classname="programmez.fred.antdemo.MyAntTask"
classpath="${dist.dir}/anttask.jar" />
<echo message="Exécution de la tâche myanttask" />
<myanttask/>
<myanttask message="Ne vous tuez pas à la tâche" />
```

```
<myanttask> Ceci est un message contenu </myanttask>
<myanttask failonerror="yes" />
</target>
```

Les valeurs d'attributs et les contenus de balises sont passés à la tâche via le mécanisme de réflexion de la classe. Lorsqu'il découvre un attribut de balise, ANT recherche un accesseur lui correspondant dans la classe de la tâche. Pour que cela fonctionne, une convention de nommage doit être respectée. L'accesseur doit commencer par `set`, suivi du nom de l'attribut de balise commençant par une majuscule. Ainsi pour un attribut 'message' l'accesseur sera '`setMessage`'. Il n'est pas obligatoire que le membre de la classe qui sera initialisé dans l'accesseur réponde à une convention de nommage, bien que cela soit une pratique répandue. Si l'accesseur n'existe pas, une exception est levée qui arrête le processus. Si l'accesseur est trouvé, il est invoqué et il reçoit en argument la valeur de l'attribut de balise. La valeur n'est pas brute mais transformée selon le contexte. Une chaîne sera passée telle quelle. Un attribut `yesno` sera converti en un booléen `truelfalse`, une propriété sera remplacée par sa valeur, etc.

Lorsque la balise elle-même contient quelque chose, le mécanisme est sensiblement différent. Dans ce cas ANT recherche une méthode de nom `addText` dans la classe. Si cette méthode n'existe pas, une exception est levée. Contrairement à ce qui se passe avec un attribut de balise, un texte contenu dans une balise n'est pas analysé et encore moins transformé. Un "yes" ne devient pas un booléen true mais reste la chaîne yes et une propriété telle que `${dist.dir}` devient la chaîne " `${dist.dir}`".

8 TRAVAILLER AVEC DES BALISES IMBRIQUÉES

XML permet de travailler avec des balises organisées de façon hiérarchiques. ANT le permet également. ANT procède alors avec un mécanisme semblable à celui que nous avons vu pour les attributs. Supposons que nous voulions que notre balise gère des balises imbriquées décrivant des membres de l'équipe de Programmez!.

```
<target name="anttaskprogrammeztest"
description="Test de AntTaskProgrammez"
depends="anttaskjar">
<taskdef name="anttaskprogrammez"
classname="programmez.fred.antdemo.AntTaskProgrammez"
classpath="${dist.dir}/anttask.jar" />
<echo message="Exécution de la tâche anttaskprogrammez" />
<anttaskprogrammez failOnError="false">
  <membre nom="Jean" />
  <membre nom="François" />
  <membre nom="Fred" />
</anttaskprogrammez>
</target>
```

Dans cet exemple, les balises imbriquées ont pour nom 'membre'. Nous écrivons une classe `Membre` qui encapsule cette notion de membre de l'équipe. La classe doit répondre aux règles que nous avons décrites pour les attributs de balise. Elle comportera donc des accesseurs correspondant aux attributs que peut supporter la balises 'membre'. Les nombres de balises imbriquées étant a priori inconnus, les instances de membres seront rangées dans un conteneur tel que `List` ou `Vector`. Enfin notre classe devra être instanciée.

Cela sera fait par une méthode qui tient lieu de fabrique d'objets. La méthode sera préfixée par `create` et se nommera donc `createMembre` pour notre exemple. La méthode pourra encore être préfixée par `add` ou `addConfigured`. Les nuances entre les trois possibilités sont expliquées dans la documentation de ANT.

```
package programmez.fred.antdemo;

import org.apache.tools.ant.Task;
import java.util.*;

public class AntTaskProgrammez extends Task {

    public void setMessage(String message) {
        this.message = message;
    }

    public void setFailOn Error(boolean failOn Error) {
        this.failOn Error = failOn Error;
    }

    public void execute() {
        if(message != null)
            log(message);
        for (Iterator it=membres.iterator(); it.hasNext(); ) {
            Membre membre = (Membre)it.next();
            log(membre.getNom());
        }
    }

    public Membre createMembre() {
        Membre membre = new Membre();
        membres.add(membre);
        return membre;
    }

    public class Membre {

        String nom;
        public void setNom(String nom){this.nom = nom;}
        public String getNom(){return nom;}
    }

    private String message;
    private boolean failOn Error;
    private Vector membres = new Vector();
}
```

9 CYCLE DE VIE D'UNE TÂCHE

Enfin sur notre site, le lecteur trouvera une classe, `AntTaskLifeCycle` qui est une classe de tâche qui vous permettra de visualiser un peu mieux comment vit une tâche lorsqu'elle est invoquée lors d'un processus de construction. Notamment, une méthode `init`, si elle existe, est invoquée avant toute autre méthode, au démarrage de la tâche.

Frédéric Mazué - fmauzé@programmez.com

Android Developer Tool sur Linux 64 : bugs et installation

L'installation des outils pour développer des applications Android, posait beaucoup de problèmes. Maintenant on peut télécharger l'ensemble des outils nécessaires pour le développement Android (y compris Eclipse) sur le site officiel. Merci à l'équipe d'Android !

Ceci ne va pas malheureusement sans quelques soucis sur des plateformes Linux 64-bits et notamment Fedora. La raison en est l'existence de programmes 32 bits dans l'archive destinée aux plateformes 64 (un bug!!!). Les tests ont été effectués sur Fedora 17.

Note : Toutes les instructions de ligne de commandes, concernant "adb" sont relatives au dossier `adt-bundle-linux-x86_64/sdk/platform-tools`. Le dossier `adt-bundle-linux-x86_64` est la racine de l'installation de ADT obtenu en décompressant son archive.

L'INSTALLATION DE ADT

- Connectez-vous à l'URL suivante : <http://developer.android.com/sdk/index.html>
- Téléchargez l'archive pour Linux 64 bits
- Décompressez
- Lancez Eclipse dans le dossier obtenu "adt-bundle-linux-x86_64" [Fig.1].

Normalement ceci devrait terminer l'installation. Mais non !

> Trouvez l'erreur

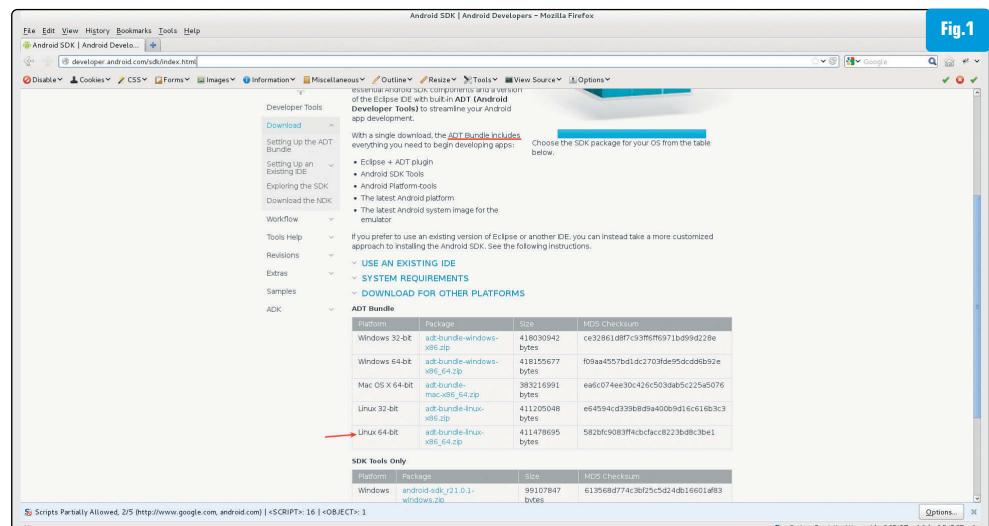
Quand on lance Eclipse, une erreur est affichée pour nous indiquer que le programme "adb" est introuvable !!! [Fig.2].

> Listez les fichiers

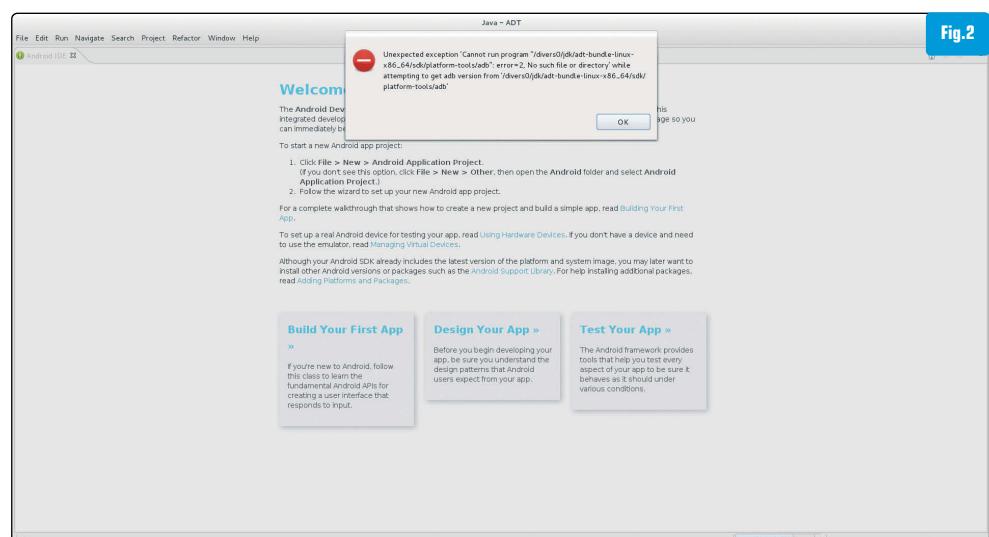
Dans un terminal tapez :

```
cd adt-bundle-linux-x86_64/sdk/platform-tools
```

Le fichier "adb" existe, ses attributs sont corrects [Fig.3].



Télécharger ADT



adb est introuvable

```
File Edit View Search Terminal Help
[kas@localhost platform-tools]$ ls -ld *
-rwxrwx---. 1 kas kas 1119899 Dec 6 04:43 aapt
-rwxrwx---. 1 kas kas 1226659 Dec 6 04:43 adb
-rwxrwx---. 1 kas kas 269818 Dec 6 04:43 aidl
drwxrwx---. 2 kas kas 80 Dec 6 04:44 api
-rwxrwx---. 1 kas kas 439993 Dec 6 04:43 dexdump
-rwxrwxr-x. 1 kas kas 2603 Dec 6 04:43 dx
-rwxrwx---. 1 kas kas 176262 Dec 6 04:43 fastboot
drwxrwx---. 2 kas kas 72 Dec 6 04:43 lib
-rwxrwx---. 1 kas kas 24867619 Dec 6 04:43 llvm-rs-cc
-rw-rw----. 1 kas kas 467531 Dec 6 04:44 NOTICE.txt
drwxrwx---. 4 kas kas 104 Dec 6 04:43 renderscript
-rw-rw----. 1 kas kas 37 Dec 6 04:43 source.properties
[kas@localhost platform-tools]$
```

Fig.3

```
File Edit View Search Terminal Help
[kas@localhost platform-tools]$ ls -ld *
-rwxrwx---. 1 kas kas 1119899 Dec 6 04:43 aapt
-rwxrwx---. 1 kas kas 1226659 Dec 6 04:43 adb
-rwxrwx---. 1 kas kas 269818 Dec 6 04:43 aidl
drwxrwx---. 2 kas kas 80 Dec 6 04:44 api
-rwxrwx---. 1 kas kas 439993 Dec 6 04:43 dexdump
-rwxrwxr-x. 1 kas kas 2603 Dec 6 04:43 dx
-rwxrwx---. 1 kas kas 176262 Dec 6 04:43 fastboot
drwxrwx---. 2 kas kas 72 Dec 6 04:43 lib
-rwxrwx---. 1 kas kas 24867619 Dec 6 04:43 llvm-rs-cc
-rw-rw----. 1 kas kas 467531 Dec 6 04:44 NOTICE.txt
drwxrwx---. 4 kas kas 104 Dec 6 04:43 renderscript
-rw-rw----. 1 kas kas 37 Dec 6 04:43 source.properties
[kas@localhost platform-tools]$ file adb
adb: ELF 32-bit LSB executable, Intel 80386, version 1 (SYSV), dynamically linked (uses shared libs), for GNU/Linux 2.6.8, not stripped
[kas@localhost platform-tools]$
```

adb est un exécutable 32 bits

> Le fichier "adb" est un exécutable 32 bits

Lancez l'utilitaire "file" dans le terminal pour trouver la nature du fichier. Surprise "adb" est un fichier exécutable 32 bits [Fig.4].

Quand on essaye d'exécuter "adb" manuellement, il affiche un message d'erreur concernant le fichier "/lib/ld-linux.so.2", car le chargeur pour les applications 32 bits est absent. Il est à noter que lancer la commande "ldd adb", ne sert à rien dans ce cas et affiche : programme non-exécutables. Pour trouver le programme à installer pour rendre "adb" exécutables et permettre au système d'avoir ld-linux.so.2, il suffit de taper dans un terminal yum provides ld-linux.so.2 [Fig.5].

> Installez glibc

Tapez su -c "yum install glibc-2.15-58.fc17.i686" dans un terminal, entrez le mot de passe root et validez [Fig.6].

> Trouvez les autres bibliothèques manquantes

Tapez `ldd adb`, puis `ENTER` dans un terminal [Fig.7]. Tous les fichiers indiqués comme non trouvés (`not found`), en exécutant `"ldd adb"`, doivent être installés, de la même manière que pour l'installation de `glibc` mentionnée plus haut, en utilisant `yum provides nomFichier`, puis `yum install`. Ajoutez à l'installation un autre fichier `"libz.so.1"`, nécessaire pour `"aapt"` [Fig.8, 9 et 10].

Bonne programmation !

Kaesar Alnijres

Développeur Java - Leader jug-Cergy
<http://www.java-javafx.com>

Une bibliothèque manque pour "adh"

Fig.6

```

kas@localhost:~/divers/0/jdk/adt-bundle-linux-x86_64/jdk/platform-tools
File Edit View Search Terminal Help
[kas@localhost platform-tools]$ su -c 'run install_glibc-2.15-58.fc17.1686'
Password:
Loaded plugin: langpacks, pretto, refresh-packagekit
Reading transaction dependencies...
--> Running transaction check
--> Package glibc-2.15-58.fc17 will be installed
--> Processing Dependency: libfreebl3.so(NSSWithASH_3.12.3) for package: glibc-2.15-58.fc17.1686
--> Processing Dependency: libfreebl3.so for package: glibc-2.15-58.fc17.1686
--> Package nss-softokn-freebl_1686@:3.14-5.fc17 will be installed
--> Finished Dependency Resolution

Dependencies Resolved

Package                               Arch      Version           Repository      Size
Installing:
glibc                               i686      2.15-58.fc17      updates          4.1 M
Resolving dependencies:
nss-softokn-freebl_1686@:3.14-5.fc17.1686      i686      3.14-5.fc17      updates          134 k

Transaction Summary
=====
Install 1 Package (+1 Dependent package)

Total download size: 4.2 M
Installed size: 14 M
Is this ok? [y/N]: y
Downloading Packages:
(1/2): glibc-2.15-58.fc17.1686.rpm | 4.1 MB  00:02
(2/2): nss-softokn-freebl-3.14-5.fc17.1686.rpm | 134 kB  00:00
-----
1.2 MB/s | 4.2 MB  00:03

Running Transaction Check
Running Transaction Test
Transaction Test Passed
Running Transaction
  Installing : nss-softokn-freebl-3.14-5.fc17.1686
  Installing : glibc-2.15-58.fc17.1686
  Verifying   : glibc-2.15-58.fc17.1686
  Verifying   : nss-softokn-freebl-3.14-5.fc17.1686

Installed:
  glibc.i686 0:2.15-58.fc17

Dependency Installed:
  nss-softokn-freebl.i686 0:3.14-5.fc17

Complete!
[kas@localhost platform-tools]$
```

Fig.6

Installer glibc

Fig.8

```

File Edit View Search Terminal Help
[kesl]localhost ~]# yum grouplist libnurses.so
Loaded plugins: langpacks, presto, refresh-packagekit
nurses:libs-5.9-4.20120204.fc17.1886 : Nurses libraries
Repo        : fedora
Matched from:
Provides : libnurses.so

[kesl]localhost ~]# yum install ncurses-libs
Password:
Loaded plugins: langpacks, presto, refresh-packagekit
ncurses-libs-libs-5.9-4.20120204.fc17.1886_0.0 already installed and latest version
Nothing to do
[kesl]localhost ~]# su -c < yum install ncurses-libs-5.9-4.20120204.fc17.1698>
Password:
Loaded plugins: langpacks, presto, refresh-packagekit
Resolving Dependencies
--> Running transaction check
--> Package ncurses-libs-1686-0.5.9-4.20120204.fc17 will be installed
--> Finished Dependency Resolution

Dependencies Resolved

+-----+-----+-----+-----+-----+
| Package          | Arch    | Version       | Repository | Size   |
+-----+-----+-----+-----+-----+
| ncurses-libs     | i686   | 5.9-4.20120204.fc17 | fedora    | 269 k |
+-----+-----+-----+-----+-----+
Transaction Summary
+-----+
| Install 1 Package |
+-----+
Total download size: 269 k
Installed size: 727 k

```

Fig. 9

Exemple d'installation de fichiers

The screenshot shows the Android Studio interface with the title bar "Java - hello/res/layout/activity_fullscreen.xml - ADT". The menu bar includes File, Edit, Run, Navigate, Search, Project, Refactor, Window, Help. The left sidebar shows the "Package Explorer" with a project structure for "hello" containing "activity_fullscreen.xml", "values", "values-v11", "values-v14", "AndroidManifest.xml", "Launcher.wkng", "pregrad-project.txt", and "project.properties". The "Android Dependencies" section lists "assets", "bin", "libs", and "res". The "res" folder contains "layout" with "activity_fullscreen.xml" and "values" with "values", "values-v11", and "values-v14". The "AndroidManifest.xml" file is also listed. The main workspace is the "Layout Editor" for "activity_fullscreen.xml", which displays a blue "DUMMY CONTENT" screen with a "Dummy Button". The "Palette" on the left lists "Form Widgets" like "Text Fields", "Layouts", "Composite", "Images & Media", "Time & Date", "Transitions", "Advanced", and "Other". The "Outline" panel on the right shows the XML structure of the layout. The "Welcome" panel on the right provides a quick start guide for new users. The bottom navigation bar includes "Problems", "JavaDoc", "Declaration", "Console", and "Progress". The bottom status bar shows "24M of 32M" and "80%".

ADT en acción

```
kas@localhost:~/divers0/jdk/adt-bundle-linux-x86_64/sdk/platform-tools]$ ldd adb
 linux-gate.so.1 => (0xf772b000)
 librt.so.1 => /lib/librt.so.1 (0xf7709000)
 libcurses.so.5 => not found
 libpthread.so.0 => /lib/libpthread.so.0 (0xf76ef000)
 libstdc++.so.6 => not found
 libm.so.6 => /lib/libm.so.6 (0xf76c3000)
 libgcc_s.so.1 => not found
 libc.so.6 => /lib/libc.so.6 (0xf7511000)
 libdl.so.2 => /lib/libdl.so.2 (0xf750c000)
 /lib/ld-linux.so.2 (0xf772c000)
[kas@localhost platform-tools]$
```

Fin 3

A screenshot of an Android application interface. The main screen has a blue background with the text "HelloWorld!!!" at the top left and "DUMMY CONTENT" in large white letters in the center. The status bar at the top shows the device ID "5554-ka_jellybean". A navigation bar at the bottom includes standard back, forward, and home buttons. A message at the bottom right of the screen reads "Exécution normale" and "Fin. 10". On the right side, there is a vertical toolbar with various icons for file operations like copy, paste, and search, and a message indicating "Hardware Buttons not enabled in ADB".

Fin-10

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](#) / [Événements](#) / [Newsletter](#) / [Vidéos](#)

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 Boulanger 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

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

NOM _____ **Prénom** _____

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

DÉVELOPPEZ 10 FOIS PLUS VITE

WINDEV®

ATELIER DE
GÉNIE LOGICIEL
PROFESSIONNEL

Document non contractuel. Le développement sur mobile est réalisé à l'aide de WINDEV Mobile. Toutes les fonctionnalités d'une application Windows peuvent ne pas être disponibles. Des fonctionnalités supplémentaires pour les mobiles sont proposées.



VERSION
EXPRESS
GRATUITE
Téléchargez-la !

DÉVELOPPEMENT D'APPLI POUR TABLETTE*: RECOMPILEZ VOTRE APPLICATION WINDOWS ACTUELLE, ET HOP, ÇA MARCHE**



* IPAD, ANDROID, WINDOWS 8 & RT

**: L'APPLICATION WINDOWS DOIT SIMPLEMENT AVOIR ÉTÉ
DÉVELOPPEE AVEC LE NUMÉRO 1 DES AGL EN FRANCE, WINDEV.
VOUS ÉCONOMISEZ DES MOIS DE DÉVELOPPEMENT

(il faudra peut-être retailler quelques fenêtres...)

Et ça marche également avec les applis Mac et Linux
développées avec WINDEV.

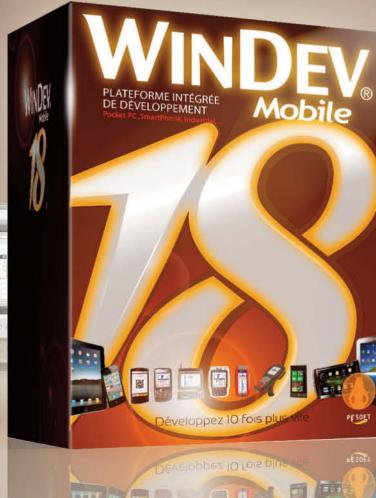


« WINDEV 18 »

Atelier de développement
multi-plateformes: Windows 8,
RT, 7, Vista, XP..., Linux, Internet,
Android, iPhone, iPad...

WINDEV Mobile 18 compile les applications mobiles.
WINDEV 18 compile les applications Windows, Mac et Linux.
WEBDEV 18 crée les sites Internet et Intranet.

Votre code source est unique



**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

