



Developpez

Magazine

Edition de Décembre-Janvier 2008/2009.
Numéro 19.
Magazine en ligne gratuit.
Diffusion de copies conformes à l'original autorisée.
Réalisation : Alexandre Pottiez
Rédaction : la rédaction de Developpez
Contact : magazine@redaction-developpez.com

Index

Java/Eclipse	Page 2
PHP	Page 10
(X)HTML/CSS	Page 17
C/C++/GTK/Qt	Page 21
JavaScript	Page 27
Conception	Page 32
Hardware	Page 41
Mac	Page 46
2D/3D/Jeux	Page 50
Liens	Page 59

Editorial

Fidèle au rendez-vous, nous vous proposons ce mois-ci encore une sélection de nos meilleurs articles, critiques de livres, et questions/réponses sur diverses technologies.

Profitez-en bien !

La rédaction

Article Hardware

Les batteries au Lithium-Ion : les comprendre et les utiliser

Comprendre le fonctionnement des batteries Li-Ion pour mieux les utiliser.

par **Anthony Nelzin**
Page 41



Article JavaScript

Introduction à la bibliothèque JavaScript jQuery

Apprenez à utiliser la bibliothèque JQuery

par **Dave Lizotte et Didier Mouronval**
Page 27



Compte rendu de la journée : Eclipse Now You Can 2008

Le 21 octobre 2008, s'est tenue à Paris la journée "Eclipse - Now you can" organisée par Geensys. Cette journée suivant l'EclipseCon, rassemble plusieurs grands acteurs autour de la plateforme Eclipse.

1. Préambule

Site officiel : <http://www.eclipse-nowyoucan/> ([Lien1](#))

Compte rendu 2006 : <http://dcabasson.developpez.com/articles/java/eclipse/eclipse-now-you-can/> ([Lien2](#))

Les slides des présentations 2008 sont mis à disposition sur le site officiel Eclipse Now You Can : <http://www.eclipse-nowyoucan.com/?Symposia/Program> ([Lien3](#))

Je tiens avant tout à remercier les organisateurs pour leur accueil et leur sympathie ainsi que les intervenants pour leur disponibilité.

2. Ralph Mueller - Eclipse Verticals: From Open Source Consumption to Value Creation



Le premier intervenant, après l'introduction par Geensys, est le directeur Eclipse Ecosystems Europe, Mr Ralph Mueller.

Insistant sur la nécessité de contribuer à Eclipse, Mr Ralph Mueller nous a également rappelé les bénéfices de l'Open Source :

- gain d'argent (gratuit)
- liberté d'utiliser le produit pour ce qu'on veut
- liberté de copier et redistribuer librement
- modifier le code et l'adapter
- vendre un produit modifié ou des services autour du produit
- principe communautaire
- écosystème gagnant/gagnant

Tout ceci est possible avec Eclipse Project.

Il y a actuellement plus de 186 membres dont 22 membres stratégiques (Intel, SAP, Borland, ...). Plus de 800 personnes participent à l'amélioration du produit ("committers") ce qui représente plus de 50 organisations. 46% des committers viennent d'IBM, principal développeur du produit, 25% d'individus non-membres et le reste réparti sur les membres.

De plus en plus d'organisations rejoignent la communauté en devenant membres.

3. Jochen Krause - Présentation de la nouvelle plateforme Eclipse e4

Jochen Krause, de la société allemande Innoopract, a présenté le futur de la plateforme Eclipse dans sa version 4. La particularité de cette version est que son architecture a été repensée afin de pouvoir réaliser des applications aussi bien en client lourd SWT comme

actuellement mais également en client léger à partir du même code !

Les principaux changements d'architecture ont dû être réalisés en raison du désir d'utiliser la plateforme en client léger ce qui ajoute des contraintes multi-utilisateurs :

- suppression des singletons (et variables statiques)
- plus d'uniformité
- plus de flexibilité
- injection de dépendances

De plus il sera possible d'ajouter des plugins réalisés dans d'autres langages que le Java.

La plateforme Eclipse n'étant pas vraiment "sexy", un mécanisme de thèmes, notamment CSS, sera ajouté.

La quasi totalité des fonctionnalités RCP ont été implémentées également dans la partie client léger (sélections, Drag'n Drop, listeners, ...).

La plateforme sera disponible courant 2010, avec mise en ligne à priori d'une pré-version en 2009. Par compte on ne devrait pas pouvoir compter sur une utilisation de l'environnement de développement Eclipse en client léger dans un premier temps même si cette fonctionnalité est proposée.

4. Eric Gaigner - Une plateforme OSGI embarquée ouvre la voie au transport intelligent.

Eric Gaigner, RDTL, a présenté l'utilisation d'Equinox dans le cadre de sa société de transport. Cette solution permet, via une application centrale et des PC dans chaque bus, de synchroniser et récupérer plusieurs informations ainsi que de gérer la billetterie (via un écran tactile). Ainsi il est possible de fournir de l'information en direct au conducteur, aux voyageurs et à la centrale (géo localisation, optimisation de carburant, aide à l'exploitation, ...).

Les échanges de données sont réalisés via des réseaux GPRS et Wifi et de façon ponctuelle afin de réduire les coûts de l'utilisation du réseau GPRS. Il est par ailleurs également possible de mettre à jour les programmes automatiquement.

Basé sur la plateforme OSGI d'Eclipse (Equinox), ce projet en cours de développement sera exploitable courant 2009.

5. Maître Cendrine Claviez, Maître Vincent Pollard - Propriété Intellectuelle et Licences de logiciels libres ou Open Source

Une intervention intéressante et atypique a été réalisée par ces deux avocats du cabinet TAJ. Leur principal rôle est de vérifier et mettre en place des licences pour logiciels sur demande des créateurs.

La tâche délicate est de vérifier la compatibilité des licences Open Source qui peuvent être classées en 3 grandes catégories :

- licences libres : GNU GPL, EUPL, Cecill A, ...
- licences semi-libres : Mozilla licence, Cecill D ...
- licences domaine public : BSD, ...

Notez la EUPL dans les licences libres. Cette licence a été élaborée au niveau européen (European Union Public Licence) suite à l'analyse entre autres de la LGPL et licences Cecill, afin, notamment, de résoudre des problèmes de droit des états européens et de compatibilité. Toute récente elle est déclinée en 27 langues qui, chacune, font foi.

Nous avons été rendus sensibles au fait qu'utiliser des briques logicielles sous licence libre de type GPL pouvait conduire à choisir le même type de licence pour son propre logiciel et donc publier le code source du produit. Ceci dû au principe "viral" de ces licences qui "contaminent" les logiciels qui les utilisent.

A noter que fournir un logiciel en Open Source peut permettre, en France, d'obtenir des subventions ou des traitements fiscaux de faveur.

6. Stands

Chaque société avait un stand où il était possible de dialoguer avec les intervenants. A noter la présence du stand de la société américaine Instanciations, présente mais n'intervenant pas, et présentant certains produits de développement intégrés Eclipse pour le GUI Testing entre autre.

7. Ateliers

J'ai suivi le Topic 1 : Eclipse pour augmenter et améliorer la productivité des équipes de développement. Le second Topic possible était : Plateforme outils (embedded development). Deux autres topics étaient à l'origine prévus mais étant donné l'engouement pour ces deux topics au détriment des deux autres, ils n'ont pas été maintenus.

7.1. ProxiAD - Industrialisation du packaging et des contrôles qualité logicielle dans Eclipse avec Maven

Arnaud Busine et Guillaume Lours nous ont expliqué ce qu'ils ont mis en place dans leur société afin d'uniformiser des déploiements et environnements de développement dans un but de productivité, de qualité et de maintenabilité des applications J2EE.

Points importants à retenir :

- leur étude a comparé les plugins Maven pour Eclipse : Maven eclipse:eclipse : ne répondait pas aux besoins, s'avère trop limité et oblige le développeur à exécuter des commandes Maven lourdes pour travailler. M2Eclipse : bon plugin ne permettant juste pas le filtrage Q4E/IAM : bon plugin ne permettant juste pas le filtrage
- Il ne faut pas qu'un développeur ait à exécuter de commandes Maven sur son poste
- WTP doit s'appuyer sur le résultat du build et non sur les sources pour pouvoir mettre en place le filtrage (à réaliser dans la configuration du POM)

7.2. Zend - PHP

Présentation des outils Zend pour le développement PHP, et donc notamment la plateforme basée sous Eclipse. En résumé la

présentation visait à montrer qu'il y a beaucoup de préjugés sur PHP (langage de bidouilleur, non sécurisé, lent, ...) mais qu'il est possible de fournir des produits de qualité grâce au Zend framework. Ceci avec facilité grâce à Zend Studio (Eclipse) fournissant les fonctionnalités avancées de développement avec des projets clé en main et de la complétion de code intelligente.

7.3. Anyware - Déployer Eclipse avec p2

Benjamin Cabé a présenté le successeur de l'update manager d'Eclipse : p2.

Cet outil, déjà intégré dans Eclipse Ganymède, change la philosophie derrière les mécanismes de mise à jour et déploiement d'Eclipse. Aussi bien en tant qu'environnement de développement, qu'en tant qu'application RCP.

P2 est un framework modulaire pilotage dans Eclipse (opérations de provisionning) mais pas forcément. Il peut installer des plugins, mais aussi des exe, ini, clés de registre, ...

Les update & install ont été optimisées (download multi-threadé, sélection du meilleur miroir, ...)

- " Bundle pooling "
- Bas niveau
- Mécanisme d'installateurs

Sur le principe on va installer des " Installable Units " (contrairement aux " features " des précédentes versions). Il y a 2 entités : les méta-données, contenant les installable units; les artefacts, contenant les données réelles : plugins, ...

p2 sera utilisable pour les gros déploiements à partir d'Eclipse 3.5, il peut cependant déjà être utilisé mais avec un risque de bugs qu'il faut dans ce cas corriger soi-même ou soumettre à la communauté pour résolution rapide.

p2 fait partie de la plateforme OSGI d'Eclipse, Equinox.

7.4. Blu Age - Faites passer Eclipse à la vitesse du Développement dirigé par les modèles avec BLU AGE, Agile Model Transformation

La présentation montrait la réalisation/génération d'une application J2EE complète à partir de schémas UML. La génération allant du modèle aux interfaces graphiques.

Assez impressionnant sur le principe.

7.5. Soyatec - IHM simplifiée

Mandatée par Microsoft, Soyatec développe un produit de génération d'IHM à partir de langage XAML. Une interface de développement est réalisée sous Eclipse sous le nom eclipse4SL (SL pour Silverlight). Le principe est de pouvoir réaliser avec le même code XAML, une application client lourd ou client léger.

8. Conclusion

Pour sa troisième édition, Eclipse Now You Can aura rassemblé trois fois plus de personnes qu'à sa première, signe de l'intérêt qui y est apporté et la qualité de l'événement. Beaucoup de thèmes sont abordés et permettent d'avoir une vue d'ensemble des possibilités offertes par la plateforme Eclipse.

Rappelons que l'événement est gratuit pour les participants, n'hésitez donc pas si vous avez l'occasion d'assister à une des prochaines éditions.

Retrouvez l'article de Nathanaël Roberts en ligne : [Lien4](#)

Débuter pas à pas avec Wicket et Eclipse

Wicket est un framework Java utilisé pour développer des applications web. Ce tutoriel s'adresse à tous ceux qui souhaitent découvrir le développement d'applications web à l'aide de ce framework sans connaissance préalable de JEE.

1. Préambule

La complexité et la multiplicité des modèles de programmation que l'on peut utiliser dans JEE créent une étape difficile à franchir quand on veut aborder ce domaine après un apprentissage du langage java. Wicket représente à mon avis une possibilité de découvrir les notions de base de la programmation d'application web en se servant quasi exclusivement de ses connaissances en java et en Html. Ce tutoriel s'applique à suivre cette démarche avec une approche pas à pas de la mise en oeuvre de wicket et de ses mécanismes de base .

2. Installation d'un projet Wicket

Une application web Java est en général contenue dans un fichier war (Web Application archive) et présente la structure suivante :

répertoire racine

```
|_+META-INF
|_+WEB-INF
|   |_+classes : les servlet (classes traitant les requêtes HTTP)
|   |               et autres classes
|   |_+lib : les librairies utilisées par l'application (fichiers .jar)
|   |_web.xml : le descripteur de déploiement de l'application
|_ressources (pages html, images, fichiers css...)
```

Nous allons nous servir d'Eclipse Ganymede version JEE pour créer notre projet d'application web utilisant wicket. Il va donc falloir créer cette structure, et référencer les librairies nécessitées par notre application. On peut le faire simplement directement à partir d'Eclipse selon la méthode décrite ici : création d'une application wicket avec eclipse ([Lien5](#))

Wicket a toutefois prévu une méthode de construction de projet appelée QuickStart que l'on peut trouver sur son site à l'adresse suivante : <http://wicket.apache.org/quickstart.html> ([Lien6](#)). Cette méthode utilise un outil de build ([Lien7](#)) appelée Maven2, pour construire ce projet. Il nous faudra donc tout d'abord télécharger et installer Maven 2.

Il va falloir ensuite utiliser la ligne de commande ([Lien8](#)) avec l'ordre généré sur le site par le quickstart. On peut le faire directement à partir d'eclipse si on a installé le plugin Wicked Shell qui permet d'utiliser la ligne de commande de Windows avec Eclipse.

Attention à ne pas confondre Wicked et Wicket !

2.1. Installation de Maven 2

On peut télécharger Maven 2 sur cette page : <http://maven.apache.org/download.html#Installation> ([Lien9](#)).

Il faut ensuite décompresser l'archive dans le répertoire de votre choix: ici ce sera : `E:\Programmation`. Maven se trouve alors dans le répertoire: `E:\Programmation\apache-maven-2.0.9`.

Il reste à positionner les variables d'environnement `M2_HOME` et `M2`.

Pour accéder aux variables d'environnement, sous Windows, il faut à partir du menu *démarrer*, cliquer sur *panneau de configuration*, puis sur *Système*, sur l'*onglet avancé*, puis enfin sur le bouton *Variables d'environnement*.

Dans les variables utilisateurs, il faut cliquer sur *nouveau*, et remplir les cases comme suit dans notre cas (à adapter selon votre

configuration):

- Nom de la variable : `M2_HOME` Valeur de la variable : `E:\Programmation\apache-maven-2.0.9`
- Nom de la variable : `M2` Valeur de la variable : `%M2_HOME%\bin`
- Nom de la variable : `JAVA_HOME` Valeur de la variable : `C:\Program Files\Java\jdk1.6.0_01` (si elle n'existe pas déjà)
- Mettre à jour la variable : `PATH` en rajoutant la valeur : `%M2%;%Path%`

Sous Linux, il faudra écrire dans la console : `# gedit .profile` sous *gnome*, ou `# kate .profile` sous *KDE*, puis ajouter les lignes suivantes :

Dans la console Linux

```
export M2_HOME=/home/user/apache-maven-2.0.9
export M2=$M2_HOME/bin
export JAVA_HOME=/usr/lib/java/jdk1.6.0_01/
PATH="$M2:$PATH"
```

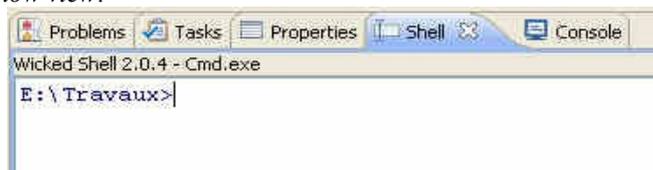
Il existe un plugin maven pour eclipse (m2eclipse) que l'on peut installer à partir de l'update d'eclipse à cette adresse : <http://m2eclipse.sonatype.org/update/> ([Lien10](#)).

2.2. Installation de Wicked Shell

On peut installer Wicked Shell via l'update d'eclipse en suivant les instructions de cette page : <http://www.wickedshell.net/download.html> ([Lien11](#))

Dans le menu *préférence* d'Eclipse, il suffit de cocher la case *Use workspace as startup directory* pour que la ligne de commande pointe sur le répertoire de travail d'Eclipse. Ici, c'est le dossier `E:\Travaux`.

On affiche ensuite la fenêtre de Wicked Shell à l'aide du menu *Show view*:



On peut alors tester l'installation de Maven avec la commande : `mvn --version`, qui si tout est bien installé devrait afficher le numéro de version de Maven.

2.3. Utilisation du QuickStart

On va remplir les 3 lignes suivantes de la page *Quickstart* du site Wicket:

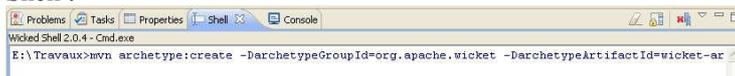
- *GroupId* qui correspondra au package java (ici on l'appellera : `com.tuto`)
- *ArtifactId* qui correspondra au dossier contenant le projet (ici : `project`)
- *Version* qui correspond à la version de Wicket que l'on veut utiliser. (ici : `1.4-SNAPSHOT`)

On obtient alors la ligne de commande suivante que l'on va copier:

```
mvn archetype:create
-DarchetypeGroupId=org.apache.wicket
-DarchetypeArtifactId=wicket-archetype-quickstart
```

```
-DarchetypeVersion=1.4-SNAPSHOT -DgroupId=tuto
-DartifactId=project
-DremoteRepositories=http://wicketstuff.org/maven/rep
ository/
```

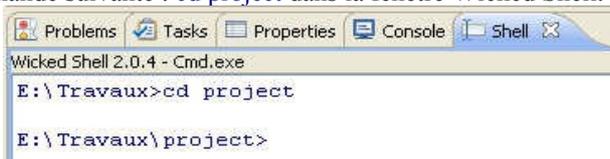
On va ensuite coller cette commande sur la fenêtre de wicked Shell :



En tapant sur *Entrée*, le projet se construit alors dans le dossier : `E:\Travaux\project`.

2.4. Ouverture du projet avec Eclipse

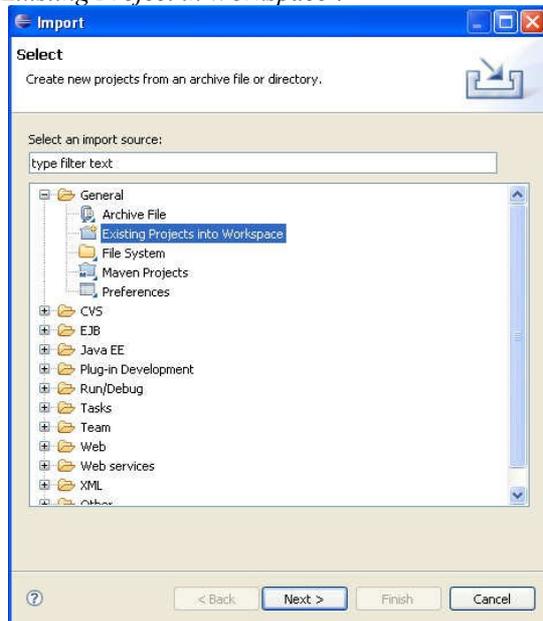
Il va maintenant falloir travailler avec ce projet dans Eclipse J2EE. Pour se placer dans le répertoire de notre projet, on tape la commande suivante : `cd project` dans la fenêtre Wicked Shell.



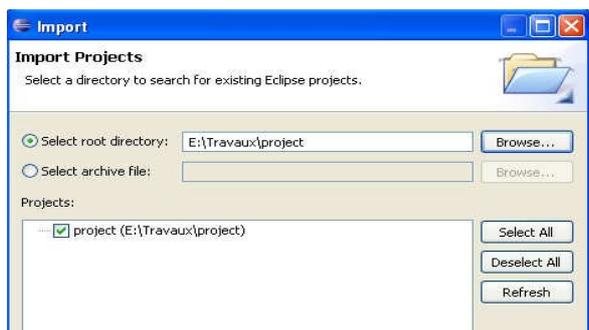
Puis on colle la commande suivante qui permettra l'utilisation du projet par Eclipse : `mvn eclipse:eclipse -DdownloadSources=true`



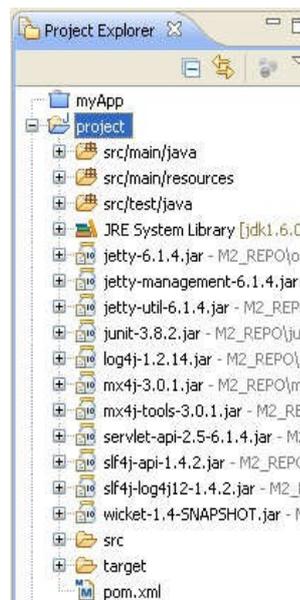
Il faut ensuite ouvrir le menu *File* d'Eclipse puis cliquer sur *Importer*. Dans la fenêtre *Import*, cliquer sur le noeud *General*, puis sur *Existing Project in Workspace* :



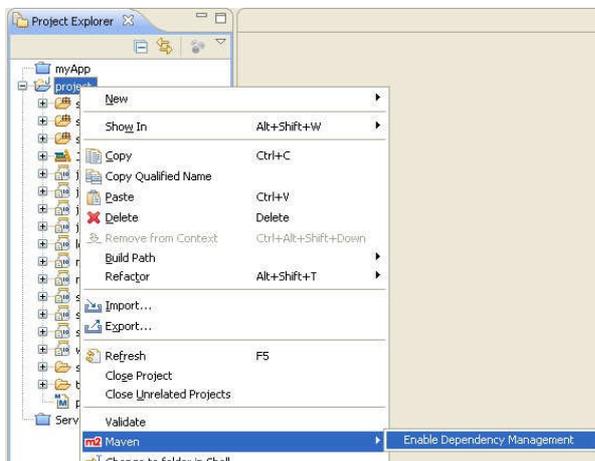
Puis sélectionner le répertoire du projet avant de fermer la fenêtre :



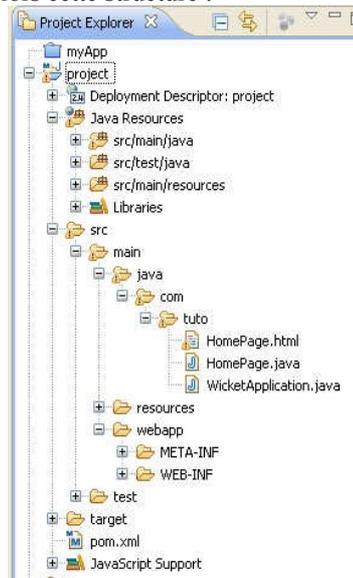
Notre projet est maintenant intégré à Eclipse, et en cliquant sur son nom dans l'onglet *Project Explorer*, on peut visualiser les différents répertoires et fichiers qui le compose.



Il reste ensuite à intégrer les dépendances en cliquant sur *Enable dependency management* comme suit :



Notre projet a alors cette structure :



Le projet est maintenant prêt à l'emploi sous Eclipse, mais pour déployer le projet Wicket sur internet, il nous faut un conteneur de servlet ([Lien12](#)), c'est à dire une application permettant d'exécuter des applications web java sur un serveur web.

Retrouvez la suite de l'article de Philippe Sénéchal en ligne : [Lien13](#)

Compte-rendu des Rencontres Spring 2008

Le 13 Novembre 2008 à Paris La Défense s'est déroulée la première édition des **Rencontres Spring**. Celles-ci sont nées d'un partenariat entre SFEIR et SpringSource. Ces conférences furent animées par de grands noms :

- Juergen Hoeller, principal développeur de Spring
- Mark Thomas, principal contributeur au projet Apache Tomcat
- Peter Cooper-Ellis, vice president of engineering chez SpringSource
- Guillaume Laforge, chef de projet de Groovy
- Julien Dubois, directeur régional de SpringSource France
- Didier Girard, directeur technique de SFEIR

1. Introduction

Le 13 Novembre a eu lieu à la Défense à Paris, la première édition des *Rencontres Spring*. Celles-ci sont le résultat d'un partenariat entre SFEIR et SpringSource.

Cet évènement fut l'occasion de rencontrer diverses personnalités Spring tels que Juergen Hoeller (le développeur de Spring), Mark Thomas (le principal contributeur de Tomcat), Peter Cooper-Ellis (vice-president of engineering de SpringSource) ou encore Guillaume Laforge (chef de projet de Groovy).

Concernant ce dernier, ce fût de plus l'occasion de discuter de l'acquisition de G2One (Groovy / Grails) par SpringSource.

2. Conférences

2.1. Keynote

2.1.1. Pourquoi Spring ?



Orateur : Didier Girard
Fonction : CTO SFEIR

Les conférences se sont ouvertes sur la keynote de Didier Girard qui a tenté de mettre en avant les raisons en faveur de l'utilisation de Spring.

Les premiers avantages de Spring sont bien connus : L'inversion de contrôle et l'AOP.

Si, historiquement, c'était sans conteste un gros avantage de Spring par rapport à J2EE, cette différence n'est plus aussi franche, JEE proposant maintenant lui aussi l'IoC et un support de la Programmation Orientée Aspect.

Mais alors pourquoi Spring ?

Pour répondre à cette question, Didier nous a fait un rappel du fonctionnement d'une application JEE.

Une application JEE se base sur les spécifications JEE qui sont un standard et censées permettre l'indépendance entre l'application et le serveur JEE.

Mais en pratique, une dépendance existe tout de même. Sans entrer dans le débat d'une dépendance technique, Didier nous prouve l'existence d'une autre dépendance par le biais de l'équipe de la production.

En effet, la production est par définition " conservatrice ". Une fois un système installé, testé et éprouvé par leur soin, elle va tout faire pour le garder tel quel le plus longtemps possible. Résultat: même si de nouvelles spécifications arrivent (JEE 5 par exemple), la production va tout faire pour bloquer le changement. La conséquence est que l'équipe de développement continuera à travailler sur l'ancien serveur et ne bénéficiera pas des apports de la

nouvelle spécification.

Et c'est là que Spring apporte une solution intéressante !

En effet, dans le cadre d'une application Spring, c'est l'application qui apporte le " conteneur ". Par conséquent, utiliser Spring 2.0 ou Spring 2.5 est totalement transparent pour la production. Celle-ci ne s'opposera donc pas à une migration vers une version de Spring qui permettrait à l'équipe de développement d'être plus productive.

En pratique ce changement devra tout de même être validé par l'équipe d'architecture, mais en général cette équipe est propice au changement aussi.

Téléchargez les slides de la présentation (pdf) ([Lien14](#))

2.1.2. Connaissez-vous SpringSource ?



Orateur : Julien Dubois
Fonction : Directeur régional SpringSource France, co-auteur de Spring par la pratique

Julien Dubois a ensuite pris la parole en tant que Directeur régional de SpringSource en France. Il a profité de la conférence pour annoncer officiellement la nouvelle version de son livre Spring par la pratique, qui paraîtra début 2009.

Concernant SpringSource, Julien a rappelé que SpringSource est la société distribuant les projets Spring (Spring Framework, Spring Batch, Spring Security, ..), et possède des employés à travers le monde : USA, UK, Pays Bas, ...

SpringSource SARL, la filiale française, étant quant à elle présente en France depuis avril 2008.

Mis à part les projets du portfolio, le business de SpringSource est fortement basé sur le support.

Tout d'abord, tous les clients du support accèdent automatiquement à divers produits commerciaux : monitoring, environnement de développement,...

De plus, en cas de problème, ils ont un contact direct avec les bonnes personnes (commiters sur le projet Spring posant problème) en cas de soucis.

Julien a profité de parler du support pour évoquer la nouvelle politique de maintenance qui a fait tant de polémiques, et il a résumé celle-ci par une phrase :

Accès à des versions packagées, certifiées et validées pour les Anciennes versions des projets

Julien a terminé sa keynote en parlant de leur dernière acquisition (G2One) avant de laisser la parole à Guillaume Laforge

Téléchargez les slides de la présentation (pdf) ([Lien15](#))

2.1.3. SpringSource rachète G2One



Orateur : Guillaume Laforge
Fonction : Chef de projet de Groovy, co-auteur de Groovy in Action, Spec Lead JSR-24

Avant toute chose, Guillaume s'est brièvement présenté. Il est le chef de projet de Groovy, le leader de la JSR-241 (la spécification de Groovy) et initiateur du framework Grails. Il est par ailleurs co-auteur du livre Groovy in Action.

Il a ensuite introduit Groovy. Celui-ci est un langage dynamique pour la JVM (au même titre que JRuby ou Scala). C'est un projet OpenSource sous licence Apache (et le restera !) qui est hébergé par la communauté Codehaus.

Ce langage est très proche de Java, et permet donc à n'importe quel développeur Java de faire le Switch facilement. Guillaume met d'ailleurs l'accent sur cette intégration transparente avec Java comme le principal intérêt d'utiliser Groovy et non un autre langage dynamique.

Ce fût ensuite à Grails d'être présenté. Celui-ci est un Framework de développement Web fortement inspiré par Ruby On Rails ou Django. Il se base sur le principe de " Convention over Configuration ", c'est-à-dire qu'un certain nombre de règles (manière d'organiser ou nommer des fichiers, utilisation d'une nomenclature des méthodes, ...) remplace une tonne de fichiers de configuration.

Pour permettre cela, Grails est fondé sur des produits OpenSource de grande renommée : Spring bien sûr, mais aussi Hibernate ou Quartz.

Malgré son coté jeune (la première release date de février 2008), plusieurs grands sites lui ont d'ores et déjà fait confiance : linkedin.com et sky.com

Guillaume a fini sa présentation sur les avantages pour Groovy et Grails d'être sous la bannière de SpringSource. Le premier d'entre eux est clairement la visibilité. Devenir un produit SpringSource va permettre à Groovy/Grails non seulement d'être connu, mais aussi de gagner la confiance des entreprises.

Pour la communauté, SpringSource possédant une équipe de développement spécialisée en Eclipse, il est fort probable de voir arriver un support officiel pour ces produits, l'offre actuelle étant particulièrement pauvre (à côté du support de JetBrains par exemple).

Concernant Spring, et même si Guillaume ne le promet pas, il est fort probable de voir plus d'intégration de Groovy dans les projets du portfolio.

Téléchargez les slides de la présentation (pdf) ([Lien16](#))

2.2. Roadmap des projets Spring



Orateur : Peter Cooper-Ellis
Fonction : SVP Engineering at SpringSource

Avant de rejoindre SpringSource, Peter Cooper-Ellis a travaillé chez Bea et c'est à lui qu'on doit la réussite du serveur d'application Weblogic. En tant que Vice-président of Engineering and Product Management, il s'occupe évidemment de la gestion des produits de SpringSource.

De ce fait, il est venu nous présenter ceux-ci qu'il classe en trois

catégories :

- Développement : portfolio, STS, modèle de programmation Spring, Groovy/Grails
- Déploiement : SpringSource Entreprise, SpringSource ERS, SpringSource Dm Server, Performance Pack
- Support : SpringSource AMF, abonnement au support, formation et consultance

A côté de la partie marketing, la roadmap non définitive de fin 2008 / 2009 nous a été présentée :

- Fin 2008
 - Spring .Net 1.2
 - Spring Integration 1.0
- 1er trimestre 2009
 - Spring 3.0 (REST, EL, Java 5)
 - Spring DM 1.2
 - Spring IDE 2.5
 - Groovy 1.6
 - Grails 1.1
- 2ème trimestre
 - Spring Dm 2.0 (RFC 124)
 - Spring Batch 2.0
 - Spring ROO
 - Spring Security 2.5
 - Spring Web (Flex)

Téléchargez les slides de la présentation (pdf) ([Lien17](#))

2.3. Tomcat en production



Orateur : Mark Thomas
Fonction : Un des principaux contributeurs du projet Apache Tomcat

Ce fût ensuite au tour de Mark Thomas de prendre la parole. Cette conférence fût des plus intéressantes, même si, une fois n'est pas coutume, elle était plus à destination des administrateurs système que des

développeurs.

Mark Thomas est l'un des principaux committers du projet Apache Tomcat. Il a résolu plus de 1500 bugs dans celui-ci sur une période de 5 ans.

Avant de rentrer dans des moyens concrets pour tuner Tomcat, Mark a tout d'abord voulu expliquer le processus à suivre pour détecter et corriger un goulot d'étranglement :

1. Bien comprendre l'architecture du système
Autrement dit, savoir ce que l'on teste, connaître les différentes ressources.
2. Stabiliser le système
Ne pas modifier le code durant la phase de tuning, car cela peut engendrer des effets de bords qui invalideront les résultats
3. Se donner des objectifs clairs
Il est primordial de savoir ce que l'on cherche à atteindre pour une application, afin de piloter l'optimisation dans le bon sens.
4. Mesurer les performances actuelles
5. Identifier le goulot d'étranglement courant
Il peut y avoir plusieurs goulots, mais il ne faut travailler que sur le plus important
6. Fixer la cause du problème
.. et pas les symptômes.
7. Répéter le processus jusqu'à atteindre les objectifs fixés

Mark a ensuite présenté différents moyens d'améliorer les performances de Tomcat.

Le premier point porta sur la gestion des fichiers de logs, la configuration par défaut n'étant pas optimale pour un environnement de production. La première optimisation consiste à élaguer la configuration afin d'écrire les logs uniquement dans des fichiers, exit donc la sortie sur la console, mais en ajoutant par ailleurs une rotation des fichiers afin d'éviter les fichiers de plusieurs Go.

Si cela est suffisant dans certains cas, dès qu'il y a beaucoup de traitements et donc beaucoup de traces écrites, un nouveau goulot d'étranglement apparaît : l'écriture sur disque. La solution, dans ce cas précis, est de passer sur du logging asynchrone. Autrement dit, les traces ne sont plus écrites directement sur disque mais stockées dans une pile mémoire. Celle-ci sera utilisée ensuite par un processus qui la videra en écrivant de manière optimisée sur le disque (écriture par bloc par exemple).

Il faudra cependant limiter la taille de cette pile afin de ne pas avoir de problème de surconsommation de mémoire. Si la capacité devait être atteinte, le logging redeviendrait temporairement synchrone.

Le gros point suivant que Mark présenta fût les différents connecteurs HTTP pour Tomcat :

- Java blocking IO
- Connecteur natif (APR)
- Java non-blocking IO

Le premier est le plus vieux de tous, et le plus stable. Cependant, il n'est pas toujours optimal. Il n'est pas des plus rapides pour les connexions SSL car il se base sur une implémentation Java (JSSE) de SSL.

Son autre point faible est le mode http Keep alive.

Pour rappel, sans ce mode, une connexion http fonctionne comme ceci :

- Ouverture connexion TCP
 - Envoi d'une requête HTTP
 - Réception réponse HTTP
- Fermeture connexion TCP

Autrement dit, pour chaque requête, une connexion TCP est créée et fermée, ce qui peut parfois être lent. Pour résoudre ce problème, un mode keep-alive permet de fonctionner autrement :

- Ouverture connexion TCP
 - Envoi d'une requête HTTP 1
 - Réception réponse HTTP 1
 - Envoi d'une requête HTTP 2
 - Réception réponse HTTP 2
 - ...
 - Envoi d'une requête HTTP N
 - Réception réponse HTTP N
- Fermeture connexion TCP

Dans ce mode, la connexion reste ouverte entre le client et le serveur pour N requêtes.

Mais le souci dans le connecteur IO bloquant, est qu'un thread est dédié à une connexion et qu'un nombre maximum de thread est fixé. Donc si ce nombre est X et que X clients se connectent en keep-alive, tous les threads sont utilisés et tout nouveau client se verra refusé.

Heureusement, le connecteur non bloquant résoud ce problème. Par contre il utilise toujours l'implémentation JSSE, et donc reste moyennement performant pour les connexions SSL.

D'où l'intérêt du connecteur natif. Celui-ci utilisant la librairie OpenSSL pour la gestion des connexions sécurisées, les performances sont bien meilleures. Il est tout aussi performant dans

la gestion des connexions keep-alive.

Voici un tableau qui résume assez bien les cas d'utilisation de l'un ou l'autre connecteur

Exigence	Préférence de connecteur
Stabilité	BIO ==> NIO ==> APR
SSL	APR ==> NIO ==> BIO
Faible accès concurrent	BIO ==> APR ==> NIO
Haut accès concurrent Sans keep-alive	BIO ==> APR ==> NIO
Haut accès concurrent Avec keep-alive	APR ==> NIO ==> BIO

Mark a ensuite expliqué deux autres moyens d'optimiser les performances :

- Utilisation d'un cache pour les ressources
- Tuning de la JVM (Garbage collector, taille du heap)

Toujours dans la problématique des performances, il est ensuite passé sur un autre sujet intéressant, le loadbalancing ainsi que le clustering de Tomcat.

Concernant le load-balancing, toute la configuration est totalement gérée du côté d'Apache, et particulièrement le mod_proxy_http. Il faudra configurer celui-ci correctement afin de rediriger les requêtes d'une même session http vers le bon Tomcat. Pour cela, une table de routage se base sur le JSessionId qui est l'identifiant unique de la session.

Cette configuration est déjà très intéressante, mais en cas de panne d'un des serveurs Tomcat, toutes les sessions de celui-ci seront perdues. Pour résoudre cela, il est nécessaire de configurer un cluster. La configuration d'un cluster par défaut se résume à une ligne dans le fichier de configuration de Tomcat.

Mark a conclu sa conférence par une série de petites astuces et conseils. C'est ainsi qu'il recommande l'utilisation de Tomcat clustérisés durant la phase de développement si l'on sait d'avance que cela sera le cas en production. De nombreux problèmes pourront ainsi être traités dès le départ.

Téléchargez les slides de la présentation (pdf) ([Lien18](#))

2.4. Spring 3.0



Orateur : Juergen Hoeller

Fonction : Leader du projet Spring

La conférence de Juergen Hoeller fût certainement celle la plus attendue de l'auditoire. Allions nous enfin avoir plus d'information sur la future nouvelle version de Spring ?

Et Juergen Hoeller à décider de faire durer le suspens !

Il a tout d'abord rappelé ce que la version 2.5 apporta :

- Configuration basée sur les annotations :
 - Propre à Spring : @Autowired, @Transactional, @Service, @Repository.
 - Mais aussi les standards : @PostConstruct, @PreDestroy, @Resource, ...
- Nouveau support tout les tests
- Nouvelle API pour Spring MVC, elle aussi basée sur les annotations
- Nouvelle API pour le support des portlets
- ..

Après ce rappel, il a tout de même enfin parlé de Spring 3.0.

Cette version nécessitera obligatoirement Java 5, mais reste compatible avec J2EE 1.4 et JEE 5.

Puis il présenta une des grosses nouveautés : les expressions langages.

Celles-ci permettront d'injecter dans un bean, une référence dynamique vers un autre bean, ou une propriété de celui-ci :

```
<bean id="monbeanA" class="test.monBeanA" init-method="initializeData"/>
```

```
<bean id="monBeanB" class="test.monBeanB">  
  <property name="maPropriete"  
    value="#{monBeanA.unePropriete}"></property>  
</bean>
```

Dans cet exemple, la propriété " maPropriete " de monBeanB sera toujours la valeur de la propriété " unePropriete " de monBeanA. Le mot " toujours " a son importance.

Il était déjà possible d'injecter une valeur provenant de la propriété d'un autre bean grâce à la classe PropertyPathFactoryBean. Mais la différence, c'est que la valeur injectée était statique. Autrement dit, si durant l'exécution la propriété monBeanA.unePropriete changeait, la propriété monBeanB .maPropriete elle ne l'était pas. Les deux valeurs n'étant ainsi plus identiques.

Avec les EL, ce n'est plus réellement la valeur de unePropriete qui sera injectée dans monBeanB mais un proxy qui à chaque utilisation fera l'équivalent de ceci :

```
Return  
applicationContext.getBean("monBeanA").getUnePropriete;
```

Au final, même si la valeur change dans le bean cible, les autres qui lui sont liés continueront de travailler sur la bonne valeur.

L'autre grosse nouveauté sera le support de REST dans Spring MVC. Le but de ce support est d'être intégré directement dans l'API de Spring MVC. Ce n'est donc pas un remplaçant de JAX-RS. Différents formats de représentation seront possibles : JSON, XML ou ATOM.

Le support de la validation du modèle sera aussi ajouté. Les annotations de Hibernate Validators seront utilisables ainsi que les futures annotations de la JSR 303.

Celles-ci seront à la fois utilisées pour la validation, mais aussi par Spring MVC pour faire du rendu. Par exemple, l'utilisation de @ShortDate sur un champ de type date aurait comme conséquence que ce champ s'afficherait sous forme de date, sans l'heure, automatiquement.

Spring 3.0 apportera aussi le support de Portlet 2.0.

Concernant la roadmap, Spring 3.0 M1 était annoncée pour fin novembre mais fût légèrement en retard pour être livrée dans le panier de Saint Nicolas.

Spring 3.0 RC1 quand à elle est prévue pour Mars 2009 et la version finale pour Avril 2009.

Téléchargez les slides de la présentation (pdf) ([Lien19](#))

2.5. Table ronde

Intervenant	Société
François	HSBC

Cherpion	
Alexandre Navarro	Société Générale Banque d'Investissement
Ben-Amar Kacimi	Voyages SNCF Technologies
David Duquenne	Improve
Guillaume Laforge	G2One / SpringSource

Les conférences se clôturèrent comme prévu par une table ronde avec des grands utilisateurs de Spring.

Durant celle-ci, nous avons pu voir que le framework est utilisé dans divers domaines et sous diverses formes.

Que ce soit comme fondation d'un framework 'maison', que ce soit dans un environnement temps réel ou encore dans des applications à très haute charge, Spring trouve toujours sa place. Son utilisation étant souvent synonyme de gain et de meilleure qualité de code.

Une question intéressante fût celle concernant Spring dm Server. Didier a en effet demandé aux utilisateurs présents si ce serveur serait potentiellement utilisé ou non.

Certains étudient clairement la solution (SGCIB) car la possibilité de recharger à chaud un module sans devoir stopper l'application est très intéressante.

Mais il en ressort qu'il va être assez difficile de faire comprendre aux 'décideurs' en quoi l'investissement est intéressant.

3. Après midi

Après les conférences, SFEIR et SpringSource ont organisé la première édition de *Speed-Consulting*.

A l'instar du speed-dating qui propose des rendez vous rapides, le speed-consulting propose de rencontrer des grands noms pendant un court moment pour avoir des informations ou des conseils de leur part.

En pratique, c'est rencontrer jusqu'à 5 personnes sur des sessions d'une vingtaine de minutes pendant lesquelles vous avez le droit de poser des questions sur n'importe quel sujet : Spring, JEE, OSGi ou encore l'acquisition de G2one.

Les personnes présentes furent :

- Didier Girard
- Julien Dubois
- Guillaume Laforge
- Peter Cooper-Ellis
- Juergen Hoeller

4. Conclusion

Pour une première édition, les Rencontres Spring furent un réel succès ! La salle fût pleine, les conférences très intéressantes. L'annonce de l'acquisition de G2One, l'arrivée de Spring 3.0, les premiers mois de Spring dm Server, tout était présent pour que cela réussisse.

Espérons que SFEIR et SpringSource récidiverons cet exploit l'année prochaine !

Retrouvez l'article de Gildas Cuisinier en ligne : [Lien20](#)

Les derniers tutoriels et articles

Débuter avec OpenID et PHP

Cet article est la traduction de **Getting Started with OpenID and PHP** (disponible ici ([Lien21](#))) et a pour but de vous faire découvrir OpenID et son utilisation.

1. Mémoire insuffisante

Quelque chose d'étrange est arrivé l'autre jour. Je surfais sur un site que je visite de façon irrégulière, je me suis authentifié avec mon login et password et cliqué sur le bouton submit. Après quelques secondes de clics et rectifications, la machine m'a informé que la vérification de mon password avait échoué. Perplexe, j'ai essayé encore plusieurs fois mais, ne rencontrant aucun succès, j'ai claqué des talons, prononcé quelques malédictions et suis retourné sur un travail plus productif. C'est seulement plus tard, après avoir répondu à quelques questions de sécurité et retrouvé mon password original, que j'ai réalisé ma gaffe, j'ai utilisé un password appartenant à un autre site.

Si cette histoire vous semble familière, c'est parce que c'est l'histoire du web d'aujourd'hui, trop de sites, trop de login et pas assez de capacité crânienne pour tous les classer précisément. Mais il y a de bonnes nouvelles à portée de la main : OpenID, un Framework libre et open source pour les "connexions singulières" à travers différents sites et applications. Les nouvelles encore meilleures ? Il existe déjà une liasse de widgets PHP qui permettent au développeur d'intégrer facilement OpenID dans les applications PHP et cet article va vous montrer comment les utiliser. Alors qu'est-ce que vous attendez ? Donnez un petit coup à la page et allons-y !

2. Moi numérique

Avant de plonger dans le code, passons quelques minutes à répondre à une question de base : Quelle est cette chose 'OpenID' et comment marche-t-il ? Selon le site Web officiel, OpenID ([Lien22](#)) est :

"Une façon simple d'utiliser une seule identité numérique à travers l'Internet". Fondamentalement, un OpenID est une URL personnalisée, choisi par vous comme votre identité en ligne et enregistré avec un prestataire de services OpenID. Chaque fois qu'un site externe doit vérifier votre identité pour des buts d'établissement de la connexion, vous fournissez cette URL au lieu de votre username; le site entre alors en contact avec votre prestataire de services OpenID pour l'authentification. Quel est le bénéfice ? Simple car votre OpenID est stocké chez votre fournisseur de services OpenID et n'importe quel site peut contacter ce fournisseur pour vous authentifier. Il n'y a pas besoin de créer de multiples comptes ou de vous rappeler une multitude de login et password pour des sites différents, tout ce que vous avez besoin c'est un simple OpenID. Cela veut dire, bien sûr, que le site externe soutient la structure OpenID; l'adoption de celui-ci augmente progressivement, et le site Web OpenID a quelques informations intéressantes sur les grandes organisations diverses qui ont commencé à utiliser la structure.

Typiquement il y a deux parties à une transaction OpenID : Consommateur et Fournisseur. Un Fournisseur ressemble à un conservateur : il permet aux utilisateurs de créer et enregistrer des

URL OpenID et gère leur compte OpenID, et il authentifie aussi l'utilisateur en tant que Consommateur sur demande. Un Consommateur (aussi parfois appelé un Parti d'Espoir) est un site Web OpenID permis.

La structure OpenID est complètement open source et n'importe quel site Web peut devenir un Consommateur ou un Fournisseur d'OpenID sans encourir n'importe quels coûts sur des honoraires de licence. En conséquence, il y a déjà un grand nombre de Fournisseurs OpenID sur le Web et un nombre croissant de sites Web ont commencé à permettre aux utilisateurs de souscrire à leurs services en utilisant OpenID.

Que se passe t'il lors d'une transaction OpenID ? Bien, quand un utilisateur essaye d'enregistrer dans un site Grand public avec un OpenID, le Consommateur entre en contact avec le Fournisseur pour vérifier les lettres de créance de l'utilisateur avant la permission de celui-ci ou son accès. L'utilisateur peut être redirigé vers le Fournisseur et prié de s'enregistrer en arrivant sur son compte avec le Fournisseur en utilisant un mot de passe; une fois que c'est fait avec succès, le Fournisseur redirige automatiquement l'utilisateur sur le site Grand public, qui maintenant traite l'utilisateur comme vérifié et lui donne les autorisations nécessaires. Une clef partagée, connue des deux partis et fortement cryptée est utilisée pour maintenir l'intégrité de la transaction et évite les "escroqueries".

Si vous êtes nouveau a OpenID, les informations ci-dessus devraient être suffisantes pour expliquer le concept de base et assurer que vous pourrez suivre la matière ce qui suit. Quoi qu'il en soit si vous désirez/avez besoin d'informations détaillées, je vous recommande de jeter un œil sur le site des développeurs de l'OpenID à l'adresse : <http://openid.net/developers/> ([Lien23](#)) ou les spécifications ([Lien24](#)) de l'OpenID.

3. Assembler les pièces

Maintenant que vous avez (espérons-le) compris les bases du fonctionnement du Framework OpenID, tournons-nous vers une question plus précise : où est-ce que PHP intervient ? Bien, il y a un bon nombre de bibliothèques écrites pour PHP et faites pour aider le développeur à ajouter rapidement le support OpenID sur son application, ce tutoriel en montre deux d'entre elles:

- La bibliothèque PHP d'OpenID ([Lien25](#)), maintenue par JanRain Inc (JanRain Inc opère aussi sur le site MyOpenID.com, un fournisseur populaire pour les identités OpenID). C'est une implémentation stable pour les clients et serveurs finaux d'une connexion OpenID, et est utilisée dans la plupart des exemples de ce tutoriel.
- Le package PEAR d'authentification OpenID ([Lien26](#)) proposé par Pádraic Brady. A noter que ce package est encore au stade de la proposition à l'heure où j'écris ces lignes et devrait bientôt être considéré au stade alpha. Elle est utilisée brièvement dans ce tutoriel pour montrer

une alternative à la librairie de JanRain Inc

Dans le cas où vous ne les avez pas encore, vous avez aussi besoin des packages PEAR suivants :

- Le package PEAR DB ([Lien27](#))
- Le package Crypt_HMAC2 ([Lien28](#))
- Le package Crypt_DiffieHellman ([Lien29](#))
- Le package Services_Yadis ([Lien30](#))

Vous pouvez installer ces packages à la main, ou en utilisant l'installateur PEAR comme suit :

```
shell> pear install Crypt_HMAC2
```

Dans le but d'essayer les exemples fournis dans le tutoriel, vous aurez aussi besoin de votre propre OpenID, allez en prendre un sur le site <http://www.myopenid.com/> ou chez n'importe quel autre fournisseur de service OpenID (et rappelez vous que vous pouvez l'utiliser sur n'importe quel site relié d'OpenID) Si vous utilisez le service MyOpenID, votre OpenID sera certainement de la forme <http://vousmeme.myopenid.com> et sera généré pour vous gratuitement.

Une fois que vous avez toutes les pièces ensemble, vous êtes prêts. Mais avant je dois vous donner un avertissement important : je ne suis pas un expert en OpenID et ce tutoriel ne prétend pas à être une référence exhaustive sur l'implémentation de OpenID (les spécifications et les bibliothèques clients changent trop rapidement pour attendre un but complet). Quoi qu'il en soit il prétend à être une introduction générale pour les développeurs PHP qui sont débutants en OpenID, pour leurs donner une idée générale de comment l'intégration PHP/OpenID fonctionne et monter leur niveau de confort avec ces technologies. Pour cette raison, j'ai gardé le code suivant très simple: souvenez-vous que vous pouvez trouver d'autres exemples plus complexes dans la documentation fournie avec les bibliothèques clientes proposées précédemment.

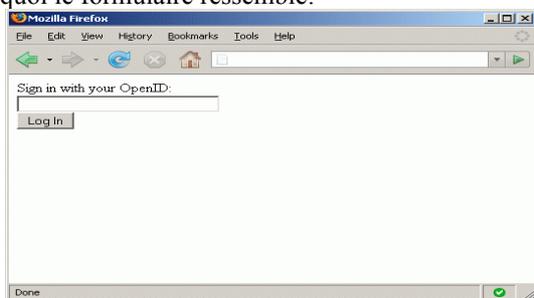
Cet avertissement donné, nous pouvons commencer.

4. Première étape

La première chose dont vous avez besoin, si vous voulez commencer à accepter OpenID sur votre site, c'est un formulaire d'authentification. Voici le code

```
<form method="post">
    Entrez votre OpenID: <br/>
    <input type="text" name="id" size="30" />
    <br />
    <input type="submit" name="submit" value="Log In"
/>
</form>
```

Voilà à quoi le formulaire ressemble:



Vous noterez que le formulaire d'authentification ne comporte pas de champs password. C'est parce que dans le Framework d'OpenID, l'authentification est gérée par le fournisseur OpenID: tout ce dont un utilisateur a besoin pour accéder à un site Consommateur est son OpenId.

Quand l'utilisateur valide ce formulaire avec son OpenID, le process du formulaire a besoin de localiser le fournisseur OpenID et de le rediriger vers celui-ci pour l'authentification. La bibliothèque PHP OpenID peut s'en charger pour vous. Regardez le code suivant, qui enveloppe le formulaire ci-dessus dans un test conditionnel et ajoute le code qui fonctionne sur la soumission du formulaire. (Je pars du principe que votre site - le site consommateur - est situé à l'adresse <http://consumer.example.com> mais soyez libre de le changer pour <http://localhost> pour vos essais.)

```
<?php
if (!isset($_POST['submit'])) {
?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
    "DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml"
xml:lang="en" lang="en">
    <head>
        <title></title>
    </head>
    <body>
        <form method="post" action="<?php echo
$_SERVER['PHP_SELF']; ?>">
            Entrez votre OpenID: <br/>
            <input type="text" name="id" size="30" />
            <br />
            <input type="submit" name="submit" value="Log In"
        />
    </form>
</body>
</html>
<?php
} else {
    // vérifie les valeurs du formulaire
    if (trim($_POST['id'] == '')) {
        die("ERROR: Entrez un OpenID valide svp.");
    }

    // fichiers inclus
    require_once "Auth/OpenID/Consumer.php";
    require_once "Auth/OpenID/FileStore.php";

    // démarrage de la session (requis pour YADIS)
    session_start();

    // crée une zone de stockage pour les données OpenID
    $store = new Auth_OpenID_FileStore('./oid_store');

    // crée un consommateur OpenID
    $consumer = new Auth_OpenID_Consumer($store);

    // commence le process d'authentification
    // crée une requête d'authentification pour le
fournisseur OpenID
    $auth = $consumer->begin($_POST['id']);
    if (!$auth) {
        die("ERROR: Entrez un OpenID valide svp.");
    }

    // redirige vers le fournisseur OpenID pour
l'authentification
    $url = $auth-
>redirectURL('http://consumer.example.com/',
'http://consumer.example.com/oid_return.php');
    header('Location: ' . $url);
}
?>
```

Comme expliqué précédemment, l'authentification avec OpenID est un process en deux étapes : premièrement le consommateur

contacte le fournisseur avec l'OpenID de l'utilisateur et ensuite, le fournisseur procède à l'authentification et redonne le contrôle au consommateur pour compléter l'authentification.

Le script ci-dessus procède à la première étape de ce processus. Une fois le formulaire soumis et validé, une nouvelle session PHP démarre et deux instances d'objets sont créées: `Auth_OpenID_FileStore` et `Auth_OpenID_Consumer`. `Auth_OpenID_FileStore` représente un emplacement sur le disque que la bibliothèque PHP OpenID utilisera pour stocker les données relatives à la tentative d'authentification courante. Le nom du répertoire à utiliser devrait être passé au constructeur de l'objet. (Cela tentera de créer le répertoire si il n'existe pas). L'objet `Auth_OpenID_Consumer` représente un consommateur OpenID et l'objet `Auth_OpenID_FileStore` génère précédemment doit être passé à son constructeur.

Pour commencer le processus d'authentification, le script appelle la méthode `begin()` de l'objet `Auth_OpenID_Consumer`, passé par l'OpenID de l'utilisateur. La valeur de retour de cette méthode est un objet `Auth_OpenID_AuthRequest`, qui représente une requête d'authentification. La méthode `redirectURL()` de cet objet est ensuite invoquée avec deux arguments : l'URL utilisée pour identifier votre site chez le fournisseur OpenID, et l'URL à laquelle le fournisseur OpenID doit rendre le contrôle après l'authentification. La valeur de retour de `redirectURL()` est une URL, utilisée pour rediriger le navigateur de l'utilisateur vers le fournisseur OpenID du site.

À ce point, le contrôle transfère vers le fournisseur OpenID, qui a besoin que l'utilisateur s'authentifie avec son password. Une fois ce processus complété, le fournisseur OpenID redirige le navigateur de l'utilisateur vers l'URL passée en second argument de la méthode `redirectURL()` - dans cet exemple le script `oid_return.php` du domaine consommateur. Typiquement, le fournisseur OpenID va aussi attacher quelques informations à la chaîne demandée en tant que paramètres GET, celles-ci sont utilisées par le consommateur pour terminer l'authentification.

Regardons maintenant ce que le script `oid_return.php` fait :

```
<?php
// Fichiers inclus
require_once "Auth/OpenID/Consumer.php";
require_once "Auth/OpenID/FileStore.php";

// démarre la session (requis pour YADIS)
session_start();

// crée une zone de stockage pour les données OpenID
$store = new Auth_OpenID_FileStore('./oid_store');

// crée un consommateur OpenID
// Lit la réponse du fournisseur OpenID
$consumer = new Auth_OpenID_Consumer($store);
$response = $consumer->complete('http://consumer.example.com/oid_return.php');

// renseigne les valeurs en fonction de celles de l'authentification
if ($response->status == Auth_OpenID_SUCCESS) {
    $_SESSION['OPENID_AUTH'] = true;
} else {
    $_SESSION['OPENID_AUTH'] = false;
}

// redirige vers la zone restreinte
header('Location: restricted.php');
?>
```

La première moitié de ce script est similaire à ce que nous avons déjà vu. Il initialise l'espace de stockage et crée l'objet `Auth_OpenID_Consumer`. Il appelle ensuite la méthode `complete()` de l'objet en lui passant l'URL de redirection du fournisseur d'accès. La valeur de retour de la méthode `complete()` de l'objet `Auth_OpenID_ConsumerResponse` qui représente la réponse du fournisseur OpenID à la requête d'authentification. Quatre codes de réponse sont possibles :

- `Auth_OpenID_SUCCESS` qui indique que l'authentification a été un succès
- `Auth_OpenID_FAILURE` qui indique que l'authentification a échoué
- `Auth_OpenID_CANCEL` qui indique que l'authentification a été abandonnée par l'utilisateur
- `Auth_OpenID_SETUP_NEEDED` qui apparaît seulement si le serveur OpenID devait authentifier de façon non interactive et en a été incapable.

Dans notre exemple ci-dessus, une fois l'authentification complète, une variable de session nommée `$_SESSION['OPENID_AUTH']` est initialisée avec un booléen indiquant si l'authentification a réussi ou non. Il est maintenant simple d'utiliser cette variable de session pour les pages d'authentification en surveillant celle-ci sur toutes les pages et seulement autoriser l'accès de l'utilisateur si celle-ci est à `True`. Voici un exemple simple sur la façon de l'implémenter sur une page restreinte (`restricted.php`)

```
<?php
// Vérifie le statut de l'authentification
session_start();
if (!isset($_SESSION['OPENID_AUTH']) ||
    $_SESSION['OPENID_AUTH'] !== true) {
    die('Vous n'avez pas le droit d'accéder à cette page! Loggez-vous svp.');
```

```
}
?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
    "DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml"
    xml:lang="en" lang="en">
    <head>
        <title></title>
    </head>
    <body>
        <h2>Page restreinte</h2>
        <p>Vous voyez cette page seulement car l'authentification OpenID a fonctionné.</p>
    </body>
</html>
```

Si un utilisateur essaie d'accéder à cette page sans avoir authentifié son OpenID, `$_SESSION['OPENID_AUTH']` devrait ne pas exister ou être à `False` et l'utilisateur verra simplement un message d'erreur. C'est seulement après une authentification réussie avec le fournisseur OpenID que `$_SESSION['OPENID_AUTH']` passe à `True` et l'utilisateur sera en mesure de voir la page restreinte.

5. Garder la simplicité

Quand vous créez pour un OpenID, il y a des chances pour qu'on vous demande pour des informations personnelles optionnelles, incluant votre nom, votre adresse email, votre langue et votre pays de résidence. Les spécifications OpenID incluent des dispositions pour le consommateur pour récupérer ces informations chez le fournisseur pendant le processus d'authentification. Cette extension appelée d'enregistrement simple est pleinement supportée dans la bibliothèque PHP d'OpenID, et la révision suivante du code précédant illustre ceci :

```

<?php
// fichiers requis
require_once "Auth/OpenID/Consumer.php";
require_once "Auth/OpenID/FileStore.php";
require_once "Auth/OpenID/SReg.php";

// démarra de session (requis pour YADIS)
session_start();

// crée une zone de stockage pour les données OpenID
$store = new Auth_OpenID_FileStore('./oid_store');

// crée un consommateur OpenID
// lit la reponse depuis le fournisseur OpenID
$consumer = new Auth_OpenID_Consumer($store);
$response = $consumer-
>complete('http://consumer.example.com/oid_return.php');
;

// crée une variable de session qui dépend de
l'authentification
if ($response->status == Auth_OpenID_SUCCESS) {
    $_SESSION['OPENID_AUTH'] = true;

    // récupère les informations d'enregistrement
    $sreg = new Auth_OpenID_SRegResponse();
    $obj = $sreg->fromSuccessResponse($response);
    $data = $obj->contents();

    // fais quelques choses avec lesdites informations
    // ...

} else {
    $_SESSION['OPENID_AUTH'] = false;
}

// redirige vers la page a accès restreint
header('Location: restricted.php');
?>

```

Dans une requête simple d'enregistrement, un consommateur peut demander pour n'importe laquelle des 8 informations : pseudo, adresse email, nom complet, date de naissance au format YYYY-MM-DD, genre, code postal, pays de résidence, langage, zone de temps. Chacune de ces informations est représentée par une clef. Par exemple 'dob' pour date de naissance ou 'email' pour l'adresse email. Pour créer une requête simple d'enregistrement en utilisant la bibliothèque PHP d'OpenID, appelez la méthode build() de la classe statique Auth_OpenID_SRegRequest avec deux tableaux en tant qu'arguments. Le premier tableau liste les clefs requises tandis que l'autre liste les clefs optionnelles. Dans l'exemple ci dessus, le nom complet de l'utilisateur, la date de naissance et le langage sont requis. Le pseudo de l'utilisateur est optionnel. Supposons l'authentification réussie, les attributs du profil demandé, si disponible chez le fournisseur OpenID, sont retournés au consommateur, avec les autres paramètres demandés. Elles peuvent être ensuite retirées en tant que tableau associatif en initialisant une instance de la classe Auth_OpenID_SRegResponse avec le paquet de réponse et en appelant la méthode contents() de l'instance, comme illustré ci dessous :

```

<?php
// fichiers requis
require_once "Auth/OpenID/Consumer.php";
require_once "Auth/OpenID/FileStore.php";
require_once "Auth/OpenID/SReg.php";

// démarra de session (requis pour YADIS)
session_start();

```

```

// crée une zone de stockage pour les données OpenID
$store = new Auth_OpenID_FileStore('./oid_store');

// crée un consommateur OpenID
// lit la reponse depuis le fournisseur OpenID
$consumer = new Auth_OpenID_Consumer($store);
$response = $consumer-
>complete('http://consumer.example.com/oid_return.php');
;

// crée une variable de session qui dépend de
l'authentification
if ($response->status == Auth_OpenID_SUCCESS) {
    $_SESSION['OPENID_AUTH'] = true;

    // récupère les informations d'enregistrement
    $sreg = new Auth_OpenID_SRegResponse();
    $obj = $sreg->fromSuccessResponse($response);
    $data = $obj->contents();

    // fais quelques choses avec lesdites informations
    // ...

} else {
    $_SESSION['OPENID_AUTH'] = false;
}

// redirige vers la page a accès restreint
header('Location: restricted.php');
?>

```

Il est maintenant facile d'utiliser ces informations de profil , dans le workflow du site consommateur - par exemple pour enregistrer automatiquement un utilisateur avec son nom ou pour envoyer un mail a son adresse. Pour illustrer considérons cette version améliorée du script précédent qui utilise l'adresse mail récupérée via le fournisseur OpenID pour vérifier si l'utilisateur a déjà un compte chez le fournisseur OpenID. Si la réponse est oui, un message d'accueil personnalisé est affiché avec l'adresse mail de l'utilisateur. Sinon un nouveau formulaire d'inscription est affiché dont les champs sont pré-remplis avec les informations récupérées depuis le profil de l'utilisateur.

Voici le code :

```

<?php
// fichiers inclus
require_once "Auth/OpenID/Consumer.php";
require_once "Auth/OpenID/FileStore.php";
require_once "Auth/OpenID/SReg.php";

// démarrage de session (requis pour YADIS)
session_start();

// crée une zone de stockage pour les données OpenID
$store = new Auth_OpenID_FileStore('./oid_store');

// crée un consommateur OpenID
// lit la réponse depuis le fournisseur OpenID
$consumer = new Auth_OpenID_Consumer($store);
$response = $consumer-
>complete('http://consumer.example.com/oid_return.php');
;

// crée une variable de session qui dépend de
l'authentification
if ($response->status == Auth_OpenID_SUCCESS) {
    $_SESSION['OPENID_AUTH'] = true;

    // récupère les informations d'enregistrement
    $sreg = new Auth_OpenID_SRegResponse();

```

```

$obj = $reg->fromSuccessResponse($response);
$data = $obj->contents();

if (isset($data['email'])) {
    // Si l'adresse mail est disponible
    // Vérifie si l'utilisateur a déjà un compte sur le
système

    // ouvre une connexion a la base
    $conn = mysql_connect('localhost', 'user', 'pass')
or die('ERROR: Connexion serveur impossible');
    mysql_select_db('test') or die('ERROR: Impossible
de sélectionner une base');

    // exécute la requête
    $result = mysql_query("SELECT DISTINCT COUNT(*)
FROM users WHERE email = '" . $data['email'] . "'") or
die('ERROR: La requête ne peut pas être exécutée');

    $row = mysql_fetch_array($result);
    if ($row[0] == 1) {
        // si oui affiche un message personnalisé
        $newUser = false;
        echo 'Bonjour et bienvenue, ' . $data['email'];
        exit();
    } else {
        // si non avertit que l'utilisateur est nouveau
        $newUser = true;
    }

    // ferme la connexion
    mysql_free_result($result);
    mysql_close($conn);
} else {
    // si l'adresse email n'est pas disponible
    // avertit que l'utilisateur est nouveau
    $newUser = true;
}
} else {
    $_SESSION['OPENID_AUTH'] = false;
    die ('Vous n'avez pas la permission d'accéder a cette
page! Re-loggez vous svp.');
```

```

}

// Si l'utilisateur n'a pas de compte,
// ou si l'adresse email n'est pas disponible
// avertit que c'est un nouvel enregistrement
// affiche un formulaire d'inscription avec les champs
pré-remplis
if ($newUser == true) {
?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0
Transitional//EN"
    "DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml"
xml:lang="en" lang="en">
    <head>
        <title></title>
    </head>
    <body>
        <h2>Créer un nouveau compte</h2>
        <form method="post" action="register.php">
            Name: <br/>
            <input type="text" name="name" value="<?php echo
@$data['fullname']; ?>" />
            <p />
            Adresse mail: <br/>
            <input type="text" name="email" value="<?php echo
@$data['email']; ?>" />
            <p />
            Date de naissance (YYYY-MM-DD): <br/>
            <input type="text" name="dob" value="<?php echo

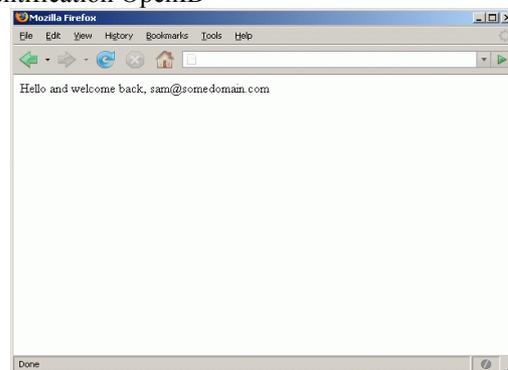
```

```

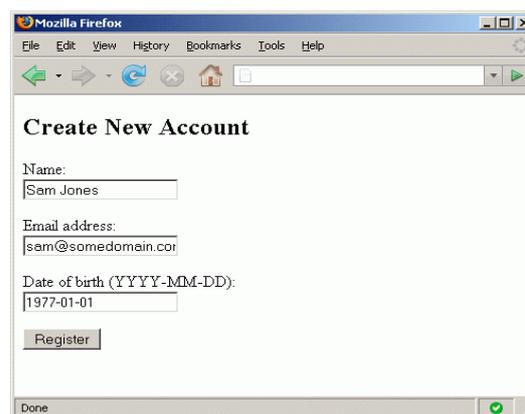
@$data['dob']; ?>" />
        <p />
        <input type="submit" name="submit"
value="Enregistrer" />
    </form>
</body>
</html>
<?php
}
?>

```

Voici ce qu'un utilisateur avec un compte sur le système doit voir après identification OpenID



Et voici ce qu'un nouvel utilisateur doit voir après identification OpenID



6. Une question de stockage

L'exemple précédent a utilisé entièrement la classe de stockage de la bibliothèque PHP d'OpenID pour le stockage des données OpenID. Si ce n'est pas à votre goût, vous pouvez aussi stocker vos données OpenID dans une base MySQL, PostgreSQL ou SQLite en remplaçant l'objet Auth_OpenID_FileStore par les objets respectifs Auth_OpenID_MySQLStore, Auth_OpenID_PostgreSQLStore ou Auth_OpenID_SQLiteStore.

L'exemple suivant illustre l'utilisation d'une base MySQL

```

<?php
<?php
if (!isset($_POST['submit'])) {
?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0
Transitional//EN"
    "DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml"
xml:lang="en" lang="en">
    <head>
        <title></title>
    </head>
    <body>
        <form method="post" action="<?php echo
$_SERVER['PHP_SELF']; ?>">
            Entrez votre OpenID: <br/>

```

```



```

Pour utiliser une base MySQL pour le stockage, initialisez un objet `Auth_OpenID_MySQLStore` et passez une connexion PEAR DB au constructeur de l'objet. Appeler la méthode `createTables()` de l'objet prend soin de créer les tables nécessaires et le reste du scénario continue comme auparavant. Quand vous utilisez les données retournées par le fournisseur OpenID dans le script de retour `oid_return.php`, n'oubliez pas de réutiliser l'objet `Auth_OpenID_MySQLStore` au lieu de `Auth_OpenID_FileStore`.

À propos, si vous voulez utiliser une bibliothèque d'abstraction quelconque au lieu de PEAR DB, vous pouvez le faire en surchargeant la classe `Auth_OpenID_DatabaseConnection` et en

l'utilisant avec votre toolkit d'abstraction. Similairement si vous voulez utiliser un mécanisme de stockage autre que par fichier ou base SQL comme base de vos sources personnalisées. Pour plus d'information là-dessus, regardez dans la documentation de la bibliothèque PHP d'OpenID de JanRain.

7. Une approche alternative

Une implémentation consommateur alternative d'OpenID est fournie par le package PEAR Authentication::OpenID_Consumer, à l'heure où j'écris il est en version alpha, mais il est intéressant de le voir en action malgré tout. (Notez que vous devez mettre le niveau de report d'erreur de PHP pour ignorer les avertissements et les notes renvoyées par le package. Ceci devrait être corrigé dans la version finale). Considérons le code suivant qui est équivalent au premier exemple du tutoriel.

```

<?php
error_reporting(E_ERROR);
if (!isset($_POST['submit'])) {
?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0
Transitional//EN"
"DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml"
xml:lang="en" lang="en">
<head>
<title></title>
</head>
<body>
<form method="post" action="<?php echo
$_SERVER['PHP_SELF']; ?>">
Entrez votre OpenID: <br/>
<input type="text" name="id" size="30" />
<br />
<input type="submit" name="submit" value="Log In"
/>
</form>
</body>
</html>
<?php
} else {
// vérifie les valeurs du formulaire
if (trim($_POST['id'] == '')) {
die("ERROR: Entrez un OpenId valide svp.");
}

// fichier requis
require_once 'OpenID/Consumer.php';
require_once 'OpenID/Store/File.php';

// démarrage de session
session_start();

// création du stockage
$store = new OpenID_Store_File('./oid_store');

// création du consommateur
$consumer = new OpenID_Consumer($store);

// débute le process d'authentification
// crée une requête d'authentification vers le
fournisseur OpenID
$auth = $consumer->start($_POST['id']);
if (!$auth)
die {Error: Entrez un OpenID valide svp.};
}

// redirige vers le fournisseur OpenID pour
authentification
$url = $auth-
>redirect('http://consumer.example.com/oid_return.php',

```

```
'http://consumer.example.com/');  
}  
?>
```

Mis à part les différences de notation, ceci est remarquablement similaire à ce que vous avez vu dans l'exemple précédent, le script crée un fichier de stockage, initialise un objet consommateur à partir du stockage, et appelle la méthode start() de l'objet Consommateur avec l'OpenID de l'utilisateur pour commencer le process d'authentification auprès du fournisseur OpenID. La méthode redirect() de l'objet Autorisation résultant qui accepte l'URL de retour et l'identifiant du site, prend soin ensuite de rediriger le navigateur de l'utilisateur jusqu'au site du fournisseur OpenID.

Une fois que l'utilisateur a complété son authentification, le fournisseur OpenID redonne le contrôle au script consommateur `oid_return.php`

```
<?php  
error_reporting(E_ERROR);  
// fichiers inclus  
require_once 'OpenID/Consumer.php';  
require_once 'OpenID/Store/File.php';  
  
// démarrage de la session  
session_start();  
  
// création stockage  
$store = new OpenID_Store_File('./oid_store');  
  
// création consommateur  
$consumer = new OpenID_Consumer($store);  
  
// crée la variable de session en fonction de  
l'authentification  
if (isset($_GET)) {  
    $response = $consumer->finish($_GET);  
    $result = $response->getResult();  
    if ($result == 'success') {  
        $_SESSION['OPENID_AUTH'] = true;  
    } else {  
        $_SESSION['OPENID_AUTH'] = false;  
    }  
}  
  
// redirige vers la page restreinte  
header('Location: restricted.php');  
?>
```

Ici, la méthode finish() de l'objet consommateur est utilisée pour compléter le process d'authentification. Le résultat de la méthode finish() est un objet réponse qui représente la réponse envoyée par le fournisseur OpenID vers la requête d'authentification du consommateur. La méthode getResult() de cet objet peut être utilisée pour tester le résultat du process d'authentification, et effectue les actions appropriées qui dépendent du résultat succès ou non.

8. Le service avec le sourire

Maintenant que vous avez vu deux implémentations différentes du consommateur OpenID, tournons notre attention vers la fin de connexion du fournisseur, d'abord nous devons noter que la bibliothèque OpenID PHP de JanRain utilisée dans le tutoriel intègre une version complète du serveur OpenID que vous pouvez utiliser pour créer votre code personnalisé de fournisseur OpenID. Un exemple de serveur est inclus dans le package pour vous aider à commencer.

Dans la plupart des cas, vous pouvez vous en tirer sans coder votre propre serveur OpenID spécialement si vos besoins sont simples. Il existe plusieurs packages open-source qui vous permettent d'installer et de gérer votre propre serveur OpenID, en voici une petite liste :

- phpMyID ([Lien31](#))
- SimpleID ([Lien32](#))
- Clamshell ([Lien33](#))

Et ce sera tout pour cet article. A travers ces quelques lignes je vous ai donné un aperçu d'OpenID expliquant ce que c'était et comment ça fonctionnait. Je vous ai aussi emmené dans l'installation de quelques bibliothèques PHP pour OpenID et montré comment vous pouviez implémenter OpenID dans vos applications. Je vous ai aussi montré l'extension d'enregistrement simple d'OpenID, vous permettant de récupérer des informations sur le profil d'un utilisateur et les réinjecter dans votre application. Finalement je vous ai montré une alternative basée sur l'implémentation PEAR de consommateur d'OpenID, et redirigé vers des packages faciles d'utilisation pour installer votre propre serveur OpenID

Si vous voulez en savoir plus sur PHP et OpenID, vous pouvez regarder ces ressources :

- An OpenID and PHP primer ([Lien34](#))
- The OpenID developer site ([Lien35](#))
- Plaxo's guide for OpenID-Enabling a Web site ([Lien36](#))
- OpenID for Non-Superusers ([Lien37](#))
- OpenID implementations for other programming languages ([Lien38](#))

Amusez-vous bien, et bon code !

Retrouvez l'article de Vikram Vaswani traduit par Joris Crozier en ligne : [Lien39](#)

Les derniers tutoriels et articles

Ajouter une icône de chargement pour vos grandes images

Ce tutoriel vous présentera deux méthodes distinctes pour réaliser un système de préchargement d'images en utilisant uniquement du CSS.

1. Introduction

Imaginons que vous ayez une grande image (ou plusieurs) sur une page en particulier et que vous vouliez que les visiteurs sachent que l'image est en cours de chargement. Vous pouvez utiliser un peu de JavaScript pour intégrer une image d'attente pour toutes les images qui n'ont pas encore été chargées. Mais, à moins que vous n'ayez une tonne sur la même page, nous avons une méthode beaucoup plus simple (et plus propre) pour réaliser la même chose.

2. Étape 1 : Trouvez une image de préchargement

Il existe beaucoup de merveilleux sites qui vous permettront de créer vos propres images de préchargement. Notre favori est ici ([Lien40](#)). Rappelez-vous simplement de ne pas choisir une icône qui serait trop grande (taille du fichier), autrement, elle ne se chargera seulement après que l'image "grand format" soit chargée. Voici celle que nous avons créée :



3. Étape 2 : Créez le style CSS

Il y a juste un peu de code à créer ici et il peut être collé directement dans votre feuille de style. Soyez certains de remplacer le texte "votreicone.gif" par le nom de votre image.

Code à coller dans votre feuille de style

```
.load
{
    background:url('images/votreicone.gif') no-repeat
    center;
}
```

4. Étape 3 : Appliquer le code

Il existe certainement plusieurs façons de faire, mais je vais vous présenter la meilleure solution, selon moi. Il vous suffit d'encapsuler l'image qui vous intéresse et qui doit être chargée

dans un *div* "load", et de lui appliquer la largeur et la hauteur de votre image.

Exemple

```
<div class="load" style="width:200px;height:200px">
    
</div>
```

Ici, cette méthode est appliquée à une image : exemple ([Lien41](#))

Comme cela a été mentionné par le contributeur *Lim Chee Aun* vous pouvez aussi contourner le *div* et appliquer la classe *.load* directement à l'image elle-même, et voici comment vous pouvez faire :

```

```

Nous n'avons pas mentionné cette idée au départ, car si l'image ne se charge pas, cela peut affecter la structure du design. Si vous êtes optimiste, que vous ne vous préoccupez pas des questions de chargement et que vous ne vous souciez pas du fait que l'image doit être centrée (selon la validation XHTML Strict) ou si vous appliquez la classe à plusieurs images, cela peut être une solution plus propre pour vous. Voir notre page d'exemples, mise à jour, ci-dessous pour voir les deux méthodes.

Voir le résultat final en action ! ([Lien41](#))

Si cette solution ne résout pas le problème pour votre utilisation, laissez-nous voir ce que vous faites différemment et nous nous ferons un plaisir d'aller plus en détail.

5. Compatibilité

Cette méthode a été testée et est compatible avec Internet Explorer 5.5, 6, 7, Firefox (PC et Mac), Netscape et Safari (PC et Mac).

Retrouvez l'article de DynamiX Web Design traduit par Alban Lelasseux en ligne : [Lien42](#)

Une maîtrise des titres : utilisation et mise en forme

Titres, je vous maîtrise. Ne me défiez pas, je suis le tout puissant développeur web. Je suis votre maître car je brandis une feuille de style avec laquelle je peux vous faire miens...

Wouaou ! D'où ça vient ? Ça semble tellement théâtral. Et, pourtant, ça définit si bien notre relation avec les éléments *titre*. Bien sûr tout le monde ne le ressent pas de cette façon. Pour ma part, je me souviens d'un temps où les titres étaient quelque chose que j'évitais d'utiliser. Je veux dire, franchement, ils sont si gros, si noirs et si laids. C'était au tout début de mon incursion dans le métier, aussi, quand les paragraphes, ou *pseudo paragraphes* comme j'aime les appeler maintenant, étaient créés en plaçant deux éléments de saut de ligne entre deux blocs de texte. Je ne savais pas faire mieux. Je ne me rendais pas compte du pouvoir que j'avais entre les mains avec ma feuille de style. J'ai appris.

1. Choisir les titres

Avant de rentrer en détail dans la magie des feuilles de style en cascade (CSS), nous devons tout d'abord savoir quels titres

utiliser. Après tout, il y en a six parmi lesquels faire son choix : *h1* à *h6*, respectivement du plus gros au plus petit, par conséquent du plus important au moins important. En règle générale, le *h1* ne sera utilisé qu'une fois (c'est ce que je recommande) et il définira

le site, le contenu de la page, ou les deux. Après le *h1*, le *h2* sera utilisé pour identifier les sections principales de la page. Ainsi, le premier titre du contenu sera un *h2*, qui pourra alors être suivi par un autre *h2*, à moins qu'il n'y ait avant une sous-partie nécessitant son propre titre. Dans un tel cas, un *h3* serait utilisé à la suite. Pensez parents, enfants, petits-enfants, et ainsi de suite, comme l'imposera l'organisation du contenu.

Voici un exemple de structure possible.

```
<h1> Animaux.com - chiens
  <h2> A propos des chiens
    <h3> Caractéristiques physiques
      <h4> Petits chiens
      <h4> Grands chiens
    <h3> Dresser des chiens
      <h4> Traits de caractère

  <h2> Navigation et liens
    <h3> Partenaires
      <h4> Amis
    <h3> Sponsors
```

Notez que je n'ai pas utilisé les sous-titres *h5* et *h6*. Je ne les ai pas mis parce que je trouve que j'en ai rarement besoin. Mais tout dépend de votre document web et de son contenu. Si ça a un sens de les utiliser, alors faites-le, mais ne sautez pas de taille. En d'autres termes, ne faites pas suivre un titre *h2* par un *h4*. Un *h3* devrait d'abord être nécessaire.

Prenez le temps de bien le faire. Les titres ne sont pas seulement importants pour l'organisation de la page et son indexation (par les moteurs de recherche), ils sont également extrêmement importants pour l'accessibilité du web. Certains utilisateurs seront capables de déterminer la sémantique d'une page à partir de la structure des titres, avant d'accéder au contenu lui-même. Certaines personnes seront en mesure d'utiliser vos titres pour naviguer dans la page courante.

Bien, maintenant que la voie est libre, passons au vif du sujet.

2. Mettre en forme les titres

Gros, noirs et laids ?! C'est tellement loin de la vérité. Les titres sont facilement mis en forme et plutôt coopératifs en offrant une cohérence de support décente multi-plateforme / multi-navigateur. Regardons ça plus en détail. Dans les exemples qui suivent je proposerai des textes avec des titres, tous *h4* parce que c'est approprié à ce cas, chacun traité différemment. Chaque exemple sera suivi par les styles utilisés pour un rendu correct sur ce site ([Lien43](#)). Cela peut quand même toujours s'avérer utile. Merci de noter que du fait des différentes polices proposées sur ce site, il peut y avoir quelques divergences de rendu. Les styles en-ligne ont été utilisés pour les exemples dans cet article.

Un titre flottant décoratif

Celui-ci est assez sympa. Il peut aussi flotter à gauche. Tout dépend de ce que vous recherchez. Pour le faire flotter à gauche, la marge doit être ajustée, en intervertissant les dimensions gauche et droite. Ça peut faire sympa avec un fond coloré ou une image. Une image de fond peut être positionnée au coin supérieur droit et un espacement ajouté à droite du titre pour faire de la place à l'image.

Voir l'exemple ([Lien44](#)) en ligne.

```
h4
{
  float : right;
  margin : 0 0 10px 30px; /* haut, droite, bas,
gauche (respectivement) */
  color : #cc6633;
  font : 150% georgia, palatino, serif;
  font-style : italic;
}
```

Un titre encadré avec fond coloré

Une couleur de fond (admirez ce contraste), une bordure, et un léger espacement : voilà tout ce qui est nécessaire pour l'obtenir. C'est un style idéal si vous voulez une forte démarcation visuelle de section. Un tel traitement convient probablement mieux à des titres supérieurs *h1* ou *h2*. Il peut être utilisé avec créativité sur une barre latérale tout aussi facilement. Il peut très bien rendre avec uniquement une bordure, ou en ajoutant une image de fond.

```
h4
{
  background-color : #f7f7f7;
  border : 1px solid #339999;
  color : #cc6633;
  padding : 3px 8px; /* haut/bas, droite/gauche
(respectivement) */
  width : 98.5%;
  margin-top : 25px;
}
```

Voir l'exemple ([Lien45](#)) en ligne.

Un titre en-ligne flottant à gauche.

Vous avez normalement vu ce type de traitement dans un livre, pas sur le web. Mais il n'y a pas de raison de ne pas l'utiliser. Ça pourrait être particulièrement utile si vous vous occupiez de nombreux titres et que vous vouliez garder de la place. Cet effet est aussi disponible à travers la propriété *display : run-in*, qui a été créée spécifiquement pour ce cas de figure (sans imbrication impropre comme cela serait nécessaire si la propriété *display : inline* était utilisée). Malheureusement, *display:run-in* n'est pas bien supportée ([Lien46](#)), pas même sous Firefox, donc, pour le moment, ceci est probablement la meilleure solution. Sinon, si une présentation au pixel près n'est pas cruciale, on peut opter pour une amélioration progressive permettant un rendu correct pour les navigateurs qui le supportent (Opera et Safari) et un rendu normal du titre pour ceux qui ne le supportent pas. Si vous utilisez des styles spécifiques par navigateur, alors vous pouvez étendre le niveau de support en proposant les deux méthodes.

```
h4
{
  float : left;
  margin : 0;
  padding : 5px 5px 0 0; /* haut, droite, bas, gauche
(respectivement) */
  color : #cc6633;
  font-size : 100%;
}
```

Voir l'exemple ([Lien47](#)) en ligne.

UN TITRE ÉLARGI EN PETITES MAJUSCULES

Je pense que ça semble plutôt sophistiqué, mais cela dépend fortement du type de police utilisé. Une police avec empattement pourrait bien rendre mais j'ai décidé de laisser votre choix de police prévaloir. Ça pourrait être un sujet d'expérimentation.

```
h4
{
  color : #cc6633;
  font-variant : small-caps;
  letter-spacing : 0.5em;
  font-size : 120%;
}
```

Voir l'exemple ([Lien48](#)) en ligne.

UN TITRE ÉLARGI ET CONVERTI

Convertissez-le en majuscules, vérifiez que c'est en gras, essayez de choisir une police grasse comme *Impact/Futura* comme j'ai utilisé ici, et étendez-le un peu. Blam. Ce traitement en fait une affirmation.

```
h4
{
  color : #cc6633;
  font : 150% impact, futura, sans-serif;
  text-transform : uppercase;
  letter-spacing : 0.3em;
  text-align : center;
}
```

Voir l'exemple ([Lien49](#)) en ligne.

Un titre avec image de fond

Utiliser des images de fond avec les titres offre presque des possibilités illimitées. Cet exemple est simplement l'une des myriades de possibilités. De plus petites images peuvent être répétées, ou vous pouvez être créatifs d'autres manières comme montré dans l'exemple qui suit.

```
h4
{
  background : url(headings-background.jpg) no-repeat;
  color : #cc6633;
  padding: 8px 0 18px 15px; /* top, right, bottom, left (respective) */
  margin : 25px 2px 0 2px; /* top, right, bottom, left (respective) */
}
```

Voir l'exemple ([Lien50](#)) en ligne.

UN TITRE USÉ

Voilà un effet sympa créé par khmerang.com ([Lien51](#)) qui utilise une balise *span* vide comme contenant positionné de façon absolue pour un arrière plan répété au dessus du texte auquel il s'applique, l'accessible méthode Gilder/Levin ([Lien52](#)) de remplacement d'image. Les tâches de couleur écaillent quelques pixels de texte provoquant cet effet. Le balisage ressemble à ça : `<h4>Texte</h4>`.

```
h4
{
  color : #cc6633;
  font : 200% impact, futura, sans-serif;
  text-transform : uppercase;
  float : left;
  overflow : hidden;
  position : relative;
  padding-left : 3px;
}
```

```
h4 span
{
  background : url(headings-worn.gif);
  position : absolute;
  width : 100%;
  height : 2em;
}
```

```
p
{
  clear : left;
}
```

Voir l'exemple ([Lien53](#)) en ligne.

Un titre avec image embarquée

Il ne s'agrandira pas comme les autres exemples de titre-image, mais c'est toujours une méthode viable pour mettre un titre sur une page. Et c'est toujours une méthode accessible. La mise en forme est extrêmement simple donc elle est directement suivie par le balisage.

```
h4
{
  color : #cc6633;
  text-align : center;
}
```

```
h4 img
{
  border : 0;
}
```

```
<h4>
  
</h4>
```

Voir l'exemple ([Lien54](#)) en ligne.

3. Devenir un chef

Les exemples ci-dessus ne sont que la proverbiale partie émergée de l'iceberg. Armé d'une feuille de style, de quelques images, et d'imagination, vous pouvez offrir le meilleur des mondes : accessibilité, lisibilité, organisation, et design, tout en offrant une page sémantique. Que vouloir de plus ?

Les titres sont-ils gros, noirs et laids ? Plus du tout.

Retrouvez l'article de Mike Cherim traduit par Sylvain Jorge Do Marco en ligne : [Lien55](#)

Introduction au CSS3 - Partie 5: Les multiples colonnes

Les multiples colonnes sont une façon importante de présenter le texte, les journaux les ont employées pendant des décennies. Cette mise en page est si courante qu'il est étonnant qu'actuellement ce genre de méthode soit l'une des techniques les plus complexes à comprendre pour un nouveau designer.

1. Introduction

Le CSS3 présente un nouveau module connu, à juste titre, en tant que mise en page en multi-colonnes. Il permet de spécifier en combien de colonnes le texte doit être divisé et comment celles-ci doivent apparaître. Comme d'habitude, des exemples peuvent être trouvés ci-dessous :

Voir la page d'exemple ([Lien56](#))

2. Multiples colonnes en utilisant le CSS3

Actuellement, cette propriété est disponible sur Firefox et Safari 3. Quand le module sera finalisé dans les spécifications du CSS3, il sera adopté par d'autres navigateurs et inclus dans leurs mises à jour.

Il y a quatre propriétés qui se rapportent à la mise en page de multiples colonnes en CSS3, permettant de définir le nombre de colonnes, la largeur, l'espace entre chaque colonne et une bordure entre chacune d'elles :

- *column-count*
- *column-width*
- *column-gap*
- *column-rule*

Actuellement, un préfixe propre au navigateur est également nécessaire pour définir comment Safari et Firefox doivent afficher la propriété. Le code à employer pour créer une mise en page de deux colonnes espacées de 1px serait :

```
.multiplecolumns
{
  -moz-column-width: 130px;
  -webkit-column-width: 130px;
  -moz-column-gap: 20px;
  -webkit-column-gap: 20px;
  -moz-column-rule: 1px solid #ddccb5;
  -webkit-column-rule: 1px solid #ddccb5;
}
```

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Aenean egestas blandit ipsum. Morbi nulla metus, luctus et, ullamcorper sit amet, commodo quis, nisl. Ut blandit lacus nec nibh. Phasellus eleifend enim et risus. Nam condimentum. Praesent euismod auctor dui.

Heading

Nunc ut leo vel magna adipiscing tempor. Donec pretium, ligula et hendrerit faucibus, sem velit accumsan tortor, sodales tempor est ligula non velit. Nulla sagittis, odio quis porta nonummy, mauris arcu gravida odio, quis aliquam lacus elit

non libero. Proin aliquam augue accumsan augue. Quisque ut eros at erat ultrices sodales. Nunc vitae ipsum. Mauris in elit in dolor imperdiet interdum. Vivamus egestas sagittis justo. Sed lorem. Sed vel neque in ipsum gravida nonummy. Nulla tempor blandit elit. Nullam a nibh. Nam quis diam non ligula pharetra sagittis. Maecenas rhoncus est vel tortor. Fusce in sem. Mauris in risus id lorem volutpat elementum. Pellentesque adipiscing laoreet ligula. Suspendisse erat. Donec porta auctor lacus. Vestibulum cursus, orci eget mollis ullamcorper, enim massa elementum dui, sed consequat nibh nisi eu tellus.

3. Enjambrer une colonne

Dans certain cas, vous voudriez peut-être qu'un élément enjambe

plus d'une colonne (un titre, une table ou une image par exemple). Ceci est facilité par l'utilisation de :

```
h2
{
  column-span: all
}
```

Vous pouvez également utiliser des chiffres pour permettre à l'élément d'enjambrer un certain nombre de colonnes. Actuellement cette propriété n'est utilisée sur aucun navigateur connu, mais devrait fournir une flexibilité appréciable pendant la conception d'un site. Il permettra de réaliser des effets comme :

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Aenean egestas blandit ipsum. Morbi nulla metus, luctus et, ullamcorper sit amet, commodo quis, nisl. Ut blandit lacus nec nibh. Phasellus eleifend enim et risus. Nam condimentum. Praesent euismod auctor dui.	non libero. Proin aliquam augue accumsan augue. Quisque ut eros at erat ultrices sodales. Nunc vitae ipsum. Mauris in elit in dolor imperdiet interdum. Vivamus egestas sagittis justo. Sed lorem. Sed vel neque in ipsum gravida nonummy. Nulla tempor blandit elit. Nullam a nibh. Nam quis
---	---

Heading

Nunc ut leo vel magna adipiscing tempor. Donec pretium, ligula et hendrerit faucibus, sem velit accumsan tortor, sodales tempor est ligula non velit. Nulla sagittis, odio quis porta nonummy, mauris arcu gravida odio, quis aliquam lacus elit

Fusce in sem. Mauris in risus id lorem volutpat elementum. Pellentesque adipiscing laoreet ligula. Suspendisse erat. Donec porta auctor lacus. Vestibulum cursus, orci eget mollis ullamcorper, enim massa elementum dui, sed consequat nibh nisi eu tellus.

4. En attendant

Il pourrait y avoir encore un moment avant que tout ça ne soit supporté sur assez de navigateurs pour que leur utilisation se généralise. *A List Apart* a publié en 2005 une implémentation JavaScript de cette fonctionnalité ([Lien57](#)) qui permet d'utiliser les différentes propriétés de la même manière sur tous les principaux navigateurs. Ces propriétés JavaScript pourront être abandonnées progressivement lorsque les propriétés CSS3 seront implémentées et utilisées plus largement. C'est une excellente solution temporaire si vous ne savez pas comment créer de multiples colonnes sur votre site.

5. En conclusion

La mise en page de multiples colonnes évitera un énorme mal de tête à beaucoup de designers, permettant un plus grand contrôle sur la façon dont le contenu sera affiché. Vous pouvez lire les spécifications complètes du W3 ([Lien58](#)) sur la mise en page avec une ou plusieurs colonnes au cas où vous voudriez en savoir plus.

Le prochain et dernier article de cette série montrera de nouvelles propriétés de fond disponibles dans le CSS3, incluant l'utilisation de plus d'un arrière-plan pour un élément et la possibilité de définir sa taille. Rappelez-vous, pour voir des exemples de ces nouveautés, jetez un coup d'oeil à notre page d'exemples CSS3 ([Lien56](#)).

Retrouvez l'article de Design Shack traduit par kalyparker en ligne : [Lien57](#)



Les derniers tutoriels et articles

Présentation du langage Vala

Vala est un nouveau langage de programmation. Enfin c'est ainsi qu'il est présenté sur le site officiel ([Lien58](#)). Dans ce tutoriel, nous allons donc découvrir ce qu'est vraiment Vala ainsi que les possibilités qu'il propose.

1. Public concerné

Je réserve ce tutoriel à un public averti, non pas que ce langage soit compliqué, mais il reste relativement jeune et il n'est pas rare de devoir mettre les mains dans le cambouis. Cependant cela devient de plus en plus rare et pour une application simple un débutant ne devrait pas rencontrer de problème.

2. Qu'est ce que Vala ?

Vala est un nouveau langage de programmation. Nouveau puisqu'il existe depuis 2006 (comparé au langage C qui existe depuis le début des années 70), et aussi nouveau par sa conception puisqu'il ne s'agit pas réellement d'un langage de programmation.

La première chose que l'on remarque lorsque l'on voit du code écrit en Vala c'est la forte similitude avec le C#, aussi bien au niveau syntaxe que fonctionnalités. La deuxième chose, plus étonnante, c'est que le compilateur Vala, va générer un code dans un langage intermédiaire, comme peuvent le faire le C# ou le Java.

A première vue, ça a l'odeur, la couleur et le goût du C#. S'il n'existait pas mono ([Lien59](#)) l'initiative serait intéressante. Mais alors où se trouve l'intérêt d'un tel langage ? Toute l'astuce de Vala réside dans le langage intermédiaire utilisé, comme je le laisse sous-entendre dans le titre de l'article, il s'agit bien sûr du C ! Du C à l'aide de la bibliothèque GObject.

Pour en finir avec cette présentation, le mariage entre le C#, pour l'écriture, et le C, pour la compilation, procure à Vala un certain nombre d'avantages, dont voici un résumé :

- Une facilité d'écriture : il est indéniable que le C# reste plus lisible que le C,
- Un langage réellement orienté objet : alors que le C/GObject accuse parfois d'une syntaxe bancal vis-à-vis de la POO (en particulier pour la définition d'une classe), Vala permet de corriger ceci,
- Une rapidité d'exécution proche du C pur : comme le code est traduit en C, on se retrouve avec des vitesses d'exécution proche de celle d'un programme écrit en C,
- La compatibilité binaire avec le C est conservée ce qui permet de ré-utiliser de nombreux outils et bibliothèques éprouvées,
- Ne dépend pas d'une machine binaire pour être exécuté.

Maintenant que les présentations sont faites, passons à la pratique.

3. Vala in a nuts

Nous nous contenterons ici d'une noix pour aborder la syntaxe du langage Vala, tout simplement parce qu'elle est proche du C# et les habitués du C++ ou de Java devraient s'y retrouver très rapidement.

Voici la liste des fonctionnalités proposées par le langage :

- Les classes
- Les interfaces
- Les classes abstraites
- Les propriétés
- Les génériques
- Foreach
- Les expressions lambda
- Les signaux
- Les types nullable
- La gestion assistée de la mémoire
- Les exceptions

Vous pourrez trouver l'ensemble des mots clés sur le wiki officiel : <http://live.gnome.org/Vala/Syntax> ([Lien60](#)).

Histoire de vous familiariser avec ce langage et en guise de pense bête, le vais détailler rapidement chaque fonctionnalité avec un exemple de quelques lignes.

3.1. Les classes

Il est possible de créer des classes, qu'elles soient publiques ou privées (portée limitée au fichier). Il est aussi possible d'utiliser l'héritage (simple uniquement).

Pour créer une classe, il faut qu'elle hérite (directement ou par l'intermédiaire de sa classe mère) de GLib.Object :

Pour finir, il est possible d'inclure une classe dans un espace de nom, soit en utilisant le mot clé **namespace** soit en préfixant le nom de la classe.

Voici quelques exemples pour illustrer ceci :

```
/* Une simple structure de données */
public MaClasse
{
}

/* Une vraie classe incluse dans l'espace de nom Dvp */
namespace Dvp
{
    public MaClasse : Object
    {
}

/* Une classe privée aussi dans l'espace de nom Dvp */
private Dvp.MaClasse : Object
{
}
```

La GLib étant la bibliothèque standard de Vala, il est inutile d'écrire GLib.Object, l'utilisation du namespace est implicite.

3.1.1. Constructeur

La construction d'un objet est quelque peu différente par rapport à ce qui existe dans d'autres langages.

Voici la séquence de création d'un objet :

- L'utilisateur demande création d'un nouvel objet grâce à l'opérateur **new**,
- Le système appelle le constructeur de classe pour l'ensemble de la hiérarchie en partant de la classe de base,
- Le système appelle le constructeur d'instance associé à notre classe.

Le constructeur de classe porte le nom **construct** et ne possède pas de portée, type de retour ou paramètre :

```
public class MaClasse
{
    construct
    {
```

Concernant le constructeur d'instance, il porte le nom de la classe sans type de retour :

```
public class MaClasse
{
    public MaClasse ()
    {
```

Il est possible de spécifier un ou plusieurs paramètres cependant il est uniquement possible de modifier les propriétés de l'objet dans cette partie (nous reviendrons plus tard sur les propriétés) :

```
public class MaClasse
{
    public string label { ... }

    public MaClasse (string label)
    {
        this.label = label;
```

Pour finir avec ce type de constructeur il est possible de le surcharger en utilisant un suffixe :

```
public class MaClasse
{
    public string label { ... }

    public MaClasse.with_label (string label)
    {
        this.label = label;
    }

    public MaClasse ()
    {
        this.label = "default";
    }
}
```

3.1.2. Destructeur

Comme très souvent, le destructeur porte le nom de la classe précédé d'un tilde :

```
public class MaClasse
{
    ~MaClasse ()
    {
    }
}
```

3.2. Interfaces

La déclaration d'une interface reste très classique, les champs à implémenter étant spécifiés comme abstraits.

```
interface MonInterface
{
    public abstract void foo ();
```

Il est possible de profiter de l'héritage entre interface et une classe peut implémenter une ou plusieurs interfaces de la même manière que pour l'héritage :

```
class Dvp.SuperClass : MaClass, MonInterface
{
    public override void foo ()
    {
```

3.3. Les classes abstraites

Si vous avez compris les interfaces, inutile de s'étendre sur les classes abstraites :

```
abstract class AbstractClass
{
    public abstract void foo ();

    public void bar ()
    {
```

3.4. Les propriétés

La syntaxe des propriétés est très riche, je vous renvoie au manuel de référence ([Lien61](#)) pour un aperçu complet des possibilités.

Voici tout de même un exemple simple de propriété :

```
class MaClass : Object
{
    private string _prop;

    public string prop
    {
        get
        {
            return this._prop;
        }
        set
        {
            this._prop = value;
        }
    }

    public static int main (string[] args)
    {
        var c = new MaClass ();
        c.prop = "test";
        print ("%s\n", c.prop);
        return 0;
    }
}
```

Pour pouvoir bénéficier des propriétés, la classe doit hériter de `GLib.Object`.

Pour simplifier l'écriture de propriétés "standard" (c'est-à-dire comme ci-dessus, qui ne font que manipuler une variable privée), Vala propose un mécanisme de propriétés automatiques qui cache cette partie du code :

```
class MaClass : Object
{
    public string prop
    {
        get;
        set;
    }
}
```

en plus de la visibilité de la propriété, il est possible d'affiner la visibilité des assesseurs :

```
class MaClass : Object
{
    public string prop
    {
        get;
        private set;
    }
}
```

Ce qui rendra la propriété lisible par tous, mais modifiable uniquement pas les instances de la classe.

Pour finir, si vous souhaitez qu'une propriété ne soit modifiée uniquement lors de la construction de l'objet, vous pouvez remplacer l'assesseur *set* par *construct* :

```
class MaClass : Object
{
    public string prop
    {
        get;
        construct;
    }

    public MaClass (string prop)
    {
        this.prop = prop;
    }
}
```

De plus ceci permet de mettre en place la valeur de la propriété avant l'appel au constructeur de classe *construct* (par défaut, la valeur n'est modifiée qu'ensuite).

Ceci n'étant pas très clair, voici un exemple qui devrait vous convaincre de l'utilité de ce mot clés :

```
class MaClass : Object
{
    public string prop
    {
        get;
        construct;
    }

    public string prop2
    {
        get;
        set;
    }

    public MaClass (string prop, string prop2)
    {
        this.prop = prop;
        this.prop2 = prop2;
    }

    construct
    {
        print ("MaClass.prop = %s\n", this.prop);
        print ("MaClass.prop2 = %s\n", this.prop2);
    }
}
```

```
public class Main
{
    public static int main (string[] args)
    {
```

```
        new MaClass ("1", "2");
        return 0;
    }
}
```

```
MaClass.prop = 1
MaClass.prop2 = (null)
```

3.5. Génériques

Vala supporte les génériques (ou template pour les adeptes du C++), de manière très classique :

```
class Wrapper<T> : Object { ... }
new Wrapper<Object> ();
```

Les différentes bibliothèques (en particulier la glib), ont été enrichies avec les génériques qui n'existent pas en C.

Leur utilisation n'est pas obligatoire (par exemple pour les listes) mais cela améliore la lisibilité du code et permet d'utiliser certaines fonctionnalités du langage, comme le foreach.

3.6. Foreach

Pour illustrer mes propos précédents, voici un exemple avec les listes :

```
List<string> files;
/* ... */
foreach (string file in files)
{
    print ("%s\n", file);
}
```

Le **foreach** est utilisable avec les tableaux, les listes (simple ou doublement chaînées), les tables de hachages et toutes les classes qui implémentent l'interface *Gee.Iterable*.

3.7. Délégué et expressions lambda

Les délégués (ou delegates, en anglais), sont utilisés à la place des pointeurs de fonctions et on les rencontre le plus fréquemment pour la gestion des événements (signaux). Pour associer une fonction callback à un signal, il suffit de l'ajouter :

```
var win = new Gtk.Window ();

win.destroy += Gtk.main_quit;
```

La même syntaxe est utilisée pour déconnecter la fonction :

```
win.destroy -= Gtk.main_quit;
```

Et plutôt que de créer une fonction pour les signaux demandant peu de traitement, il est préférable d'utiliser les expressions lambda qui permettent d'inclure le corps de la fonction directement :

```
win.destroy += (s) => {
    print ("Quit\n");
    Gtk.main_quit ();
};
```

3.8. Les signaux

Nous venons de voir comment intercepter un signal, nous allons donc nous attarder sur la création d'un signal pour notre objet.

C'est extrêmement simple, il suffit de déclarer que votre objet envoie un signal à l'aide du code suivant :

```
public signal void sig_1 (int a);
```

Ensuite, votre objet émet ce signal tout naturellement :

```
this.sig_1 (10);
```

Et pour finir, si vous souhaitez intercepter le signal, il suffit d'utiliser les *delegates* ou les fonctions lambda :

```
o.sig_1 += (s, a) => {  
    print ("sig_1 : %d\n", a);  
};
```

Le premier paramètre correspond à l'objet qui a lancé le signal (*s* pour *sender*), il est passé explicitement.

3.9. Les exceptions

Pour ce qui est de la gestion des exceptions, nous retrouvons une syntaxe traditionnelle :

```
class MaClass : Object  
{  
    /* Fonction pouvant lancer une exception de type  
    Error */  
    public void foo () throws Error { ... }  
  
    public static int main (string[] args)  
    {  
        try  
        {  
            foo ();  
        }  
        catch (Error e)  
        {  
            warning (e.message);  
        }  
        finally  
        {  
            /* ... */  
        }  
        return 0;  
    }  
}
```

Par contre pour la création et le lancement d'une exception, nous devons passer par les spécificités de la GLib.

Il faut commencer par créer un nouveau domaine d'erreur avec les différents types d'erreur possible :

```
public errordomain MyError  
{  
    ERROR_1,  
    ERROR_2  
}
```

Et ensuite, nous pouvons créer et lancer notre exception :

```
throw new MyError.ERROR_1 ("Error message");
```

4. Utilisez vos anciennes bibliothèques

Comme précisé en introduction, il est possible de réutiliser vos bibliothèques écrites en C pour vos nouveaux développements en Vala. Pour cela, il suffit de créer un fichier d'en-tête vapi. Ce fichier contient simplement la déclaration des classes. Il est bien sûr nécessaire que votre interface corresponde à une approche objet (même s'il est possible de corriger certaines imperfections grâce à l'annotation *CCode*).

S'il est possible d'écrire ces fichiers à la main pour de petites bibliothèques, cela n'est pas envisageable pour les interfaces plus conséquentes (par exemple pour GTK+). Il existe donc le programme *gobject-introspection* qui pourra vous aider dans cette tâche.

5. FAQ

Voici une mini-FAQ qui vous permettra de répondre à quelques questions qui ne se trouvent pas forcément sur le site officiel.

5.1. Comment fonctionnent les callback ?

Comme précisé ci-dessus pour intercepter un signal, il faut utiliser les *delegates*. La connexion est extrêmement simple, cependant le prototype de la fonction callback possède quelques subtilités non précisées dans la documentation. Premièrement, il faut utiliser l'annotation *Callback*, ceci permet au compilateur de créer un swapper afin d'assouplir la liste des arguments. Deuxièmement il est possible d'utiliser indifféremment une méthode de classe ou d'instance pour gérer le signal. Voici quelques exemples de *callback* :

```
public class TestCb : Object  
{  
    [Callback]  
    private void cb_1 ()  
    {  
    }  
  
    [Callback]  
    private void cb_2 (Gtk.Button sender)  
    {  
    }  
  
    [Callback]  
    private static void cb_3 ()  
    {  
    }  
  
    public static int main (string[] args)  
    {  
        /* ... */  
  
        Gtk.Button btn = new Gtk.Button.with_label  
        ("Click!");  
  
        // Méthode d'instance sans paramètre  
        btn.clicked += this.cb_1;  
  
        // Méthode d'instance avec un paramètre  
        btn.clicked += this.cb_2;  
  
        // Méthode de classe  
        btn.clicked += TestCb.cb_3;  
  
        /* ... */  
    }  
}
```

Le compilateur se chargera de passer la référence vers l'instance si nécessaire et de compléter la liste des arguments.

5.2. ref, out, weak et transfert de propriété, de quoi s'agit-il ?

Voici des mots clés peu connus des développeurs C.

Le mot clé **ref** est utilisé pour spécifier qu'un paramètre est passé par référence, donc peut être modifié par la fonction.

Le mot clé **out** précise que le paramètre est utilisé pour stocker une valeur de retour.

Le mot clé **weak** désigne une référence en la définissant comme faible, c'est à dire que si toutes les références sur un objet sont faibles, l'objet sera détruit.

Le transfert de propriété, indiqué par le caractère #, permet de transférer la propriété d'une référence d'une variable à une autre. Par exemple :

```
Foo foo = #bar;
```

Normalement, la référence de l'objet *bar* est copiée dans *foo* et le nombre de références augmentée d'un. Ici, *foo* va contenir la référence de *bar* puis *bar* sera mis à **null** sans que le compteur de référence soit incrémenté.

Ceci est particulièrement utilisé lorsqu'une fonction retourne une chaîne de caractère créée localement, afin que celle-ci continue d'exister après le retour de la fonction.

Ceci se retrouve en C# dans les propriétés avec le mot clés *new*.

5.3. L'avenir de vala

Vala est encore jeune et cela se ressent lorsqu'on développe une application importante : il n'est pas rare de devoir modifier les fichiers vapi à la main pour corriger un problème généralement dû à une ambiguïté dans le nommage des fonctions en C.

Au fils de mes développements j'ai constitué un patch pour le fichier `gtk+-2.0.vapi` que vous pouvez trouver ici ([Lien62](#)). Il contient les bugs qu'il est difficile de corriger.

5.4. Comment utiliser le fichier config.h ?

Les développeurs du monde Linux, en particulier ceux familiers

des autotools connaissent le fameux fichier *config.h*. Pour pouvoir l'utiliser, il suffit de créer un fichier d'interface vapi ressemblant à l'exemple suivant :

```
[CCode (cprefix = "", lower_case_cprefix = "",
header_filename = "config.h")]
namespace Config
{
    public const string PACKAGE;
    public const string PACKAGE_BUGREPORT;
    public const string PACKAGE_NAME;
    public const string PACKAGE_STRING;
}
```

5.5. Quel système de construction utiliser ?

Pour les adeptes des autotools, il est parfaitement possible de les utiliser avec vala. Le programme `vala-gen-project` vous génère tous les fichiers nécessaires pour débiter un projet avec les autotools.

Ou, si comme moi, vous trouvez les autotools repoussant, il existe `waf` ([Lien63](#)) qui prend aussi en compte le langage Vala.

6. Références

- The Vala Programming Language ([Lien58](#)), sur GNOME Live!
- Hackers' Guide to Vala ([Lien64](#))

Retrouvez la suite de l'article de Nicolas Joseph en ligne : [Lien65](#)

Intégrer des gadgets logiciels (ou widgets) avec QWebKit

Cet article est une traduction d'un article écrit par Kent Hansen sur le blog officiel de Qt. Il y explique comment intégrer facilement des gadgets logiciels (*widgets*) dans vos fenêtres Qt avec Qt WebKit. Vous pouvez trouver l'original ici ([Lien66](#)) et le code source sur le repository svn : `svn://labs.trolltech.com/svn`. Le code source étant basé sur QScript, voici une version c++ écrite par Mongaulois : C++ ([Lien67](#))

N'hésitez pas à laisser vos commentaires ou poser vos questions sur le forum ([Lien68](#))

1. Intégrer un gadget logiciel dans votre application Qt 4.4 avec QWebKit

L'interface de programmation d'intégration de Qt WebKit de la version 4.4 comporte une fonctionnalité très sympa qui vous permet d'intégrer des gadgets logiciels dans un QWebView. J'ai donc d'abord eu l'idée d'utiliser cette fonctionnalité pour intégrer des canevas Qt Designer à l'aide de QUiLoader pour charger le canevas et Qt Script pour saisir la logique du canevas. Après avoir passé environ une heure pour tout paramétrer, j'aimerais mettre en évidence cette partie comme étant une partie où "tout se met en place parfaitement" - où les toutes nouvelles options de la version 4.4 se combinent aux bonnes vieilles fonctionnalités de jadis pour produire un accomplissement synergique qui stimule le cerveau.

Côté HTML, j'ajoute un onglet comme ceci.

```
<object type="application/x-qtform" width="500"
height="400">
    <param name="form"
value="http://chaos.troll.no/~khansen/calculator.ui"/>
    <param name="script"
value="http://chaos.troll.no/~khansen/calculator.js"/>
</object>
```

L'onglet identifie un canevas et un script que le plug-in téléchargera et utilisera. Ensuite, je crée une sous-catégorie QWebPluginFactory qui sera invitée par WebKit pour créer mon plug-in. D'ordinaire, j'écrirais les classes liées au plug-in dans C++ , mais puisque le Qt Script Bindings Generator (générateur de

liaisons script) prend forme à présent (et parce que je reste un indécrottable mordu de scripts), j'ai tout rédigé en Qt Script. L'usine de plug-ins ressemble à ceci.

```
<pre>
function MyWebPluginFactory(parent)
{
    QWebPluginFactory.call(this, parent); // call base
class constructor }

MyWebPluginFactory.prototype = new QWebPluginFactory();

MyWebPluginFactory.prototype.create =
function(mimeType, url, argumentNames, argumentValues)
{
    if (mimeType != "application/x-qtform")
        return null;

    var formUrl = getArgumentValue("form",
argumentNames, argumentValues);
    var scriptUrl = getArgumentValue("script",
argumentNames, argumentValues);
    if (formUrl == undefined)
        return null;

    return new MyWebPlugin(new QUrl(formUrl), new
QUrl(scriptUrl));
}
</pre>
```

La fonction QWebPluginFactory::create() est réexécutée pour

gérer le type de mime *application/x-qform*. Les URLs du canevas et du script sont transférés vers la nouvelle instance My WebPlugin ; c'est le gadget logiciel qui est actuellement intégré dans l'affichage. Le MyWebPlugin agit pour télécharger le canevas et le script et s'initialise paresseusement à mesure que les données nécessaires deviennent disponibles. Voici l'exécution complète de MyWebPlugin.

```
<pre>
function MyWebPlugin(formUrl, scriptUrl, parent) {
    QWidget.call(this, parent); // call base class
    constructor

        this.initialized = false;
        this.formReply = this.downloadFile(formUrl,
this.formDownloaded);
        this.scriptReply = this.downloadFile(scriptUrl,
this.scriptDownloaded); }

MyWebPlugin.prototype = new QWidget();

MyWebPlugin.prototype.downloadFile = function(url,
callback) {
    if (this.accessManager == undefined)
        this.accessManager = new
QNetworkAccessManager();
    var reply = this.accessManager.get(new
QNetworkRequest(url));
    reply.finished.connect(this, callback);
    return reply;
}

MyWebPlugin.prototype.formDownloaded = function() {
    var loader = new QUiLoader();
    this.form = loader.load(this.formReply);
    var layout = new QVBoxLayout(this);
    layout.addWidget(this.form, 0, Qt.AlignCenter);
    this.initialize();
}

MyWebPlugin.prototype.scriptDownloaded = function() {
    var stream = new QTextStream(this.scriptReply);
    this.script = stream.readAll();
    this.initialize();
}

MyWebPlugin.prototype.initialize = function() {
    if (this.initialized)
        return;
    if ((this.form == undefined) || (this.script ==
undefined))
        return;
    var ctor = eval(this.script);
    if (typeof ctor != "function")
        return;
    this.instance = new ctor(this.form);
    this.initialized = true;
}
</pre>
```

La nouvelle interface de programmation de gestion de réseau présentée dans Qt 4.4 (QNetworkAccessManager et ses amis) sert

à télécharger les fichiers. (Avec un plug-in plus élaboré, vous voudriez certainement afficher les informations de progression pendant le téléchargement des données, en utilisant le signal QNetworkReply's downloadProgress().) A noter également que puisque QNetworkReply est un QIODevice, créer le canevas et préparer le script à la fin des téléchargements est d'une élégance infinie. Le "protocole" utilisé pour appliquer le script au canevas est le suivant : Le script est évalué et si le résultat est une fonction, cette fonction est appelée comme argument avec le canevas. La fonction accroche ensuite la fonctionnalité aux composants du canevas. Il est temps de créer un affichage et de l'essayer.

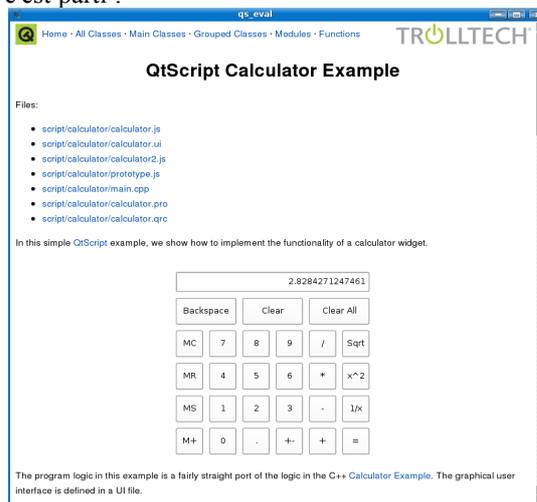
```
<pre>
var view = new QWebView();
view.settings().setAttribute(QWebSettings.PluginsEnable
d, true);

var factory = new MyWebPluginFactory();
view.page().setPluginFactory(factory);

view.load(new QUrl("script-calculator.html"));
view.show();

QCoreApplication.exec();
</pre>
```

Activez vos plug-ins, paramétrez l'usine à plug-ins, chargez votre page et c'est parti !



Exemple : Calculatrice

J'ai pris le HTML pour la documentation de Qt Script Calculator Example ([Lien69](#)), qui contient une image de l'application en action et je l'ai remplacé par l'onglet de l'image avec un ongles d'objet qui charge le canevas et le script. Alors, au lieu de regarder une capture d'écran, vous pouvez en fait utiliser la calculatrice qui se trouve sur la page de documentation ; le canevas et le script peuvent être les mêmes que ceux utilisés par l'application de la calculatrice seule. Neat-o.

Si vous voulez vous amuser avec, le code se trouve dans le dossier d'exemples du générateur de liaisons Qt (WebKitPlugins.qs).

Retrouvez l'article de Kent Hansen en ligne : [Lien70](#)

Les derniers tutoriels et articles

Introduction à la bibliothèque JavaScript jQuery

Nous allons, dans cet article, effectuer une introduction à la bibliothèque JavaScript : jQuery.

Il va sans dire que pour comprendre tant soit peu l'article qui suit, il vous faut un minimum de connaissances JavaScript et du modèle objet des documents (DOM). Nous verrons donc la base en vous donnant les détails nécessaires à votre apprentissage afin que vous soyez en mesure de progresser par vous-même facilement.

Pour démarrer, il vous faudra la bibliothèque jQuery. La version la plus récente se trouve ici : <http://jquery.com/src/> ([Lien71](#))

1. Hello world !

Comme presque tout ce que nous faisons avec jQuery, ce dernier lit ou manipule le modèle objet du document (DOM). Il faut donc être sûr que nous n'ajoutons des événements ou autres que lorsque le DOM (le document) est complètement chargé. Pour cela, nous enregistrons un événement « document chargé » ou « `$(document).ready(function(){})` ».

```
$(document).ready(function()
{
    // Traitement à effectuer lorsque le DOM est prêt
});
```

Mettre un message d'alerte dans cette fonction n'a pas beaucoup de sens, puisqu'un message n'a pas besoin que le DOM soit chargé. Alors poussons l'exemple un peu plus loin et essayons d'afficher un message lorsqu'un lien est cliqué.

```
$(document).ready(function()
{
    $("a").click( function()
    {
        alert("Hello world!");
    }
    );
});
```

Cela devrait afficher le message si vous cliquez sur un lien de votre page.

Un coup d'oeil sur notre script :

- `$("a")` est un sélecteur jQuery, ici, il sélectionne tous les éléments de type lien `<a>`.
- `$` est un alias pour la classe jQuery : `$()` fabrique un nouvel objet jQuery.
- La fonction `click()` appelée dans la suite est une méthode de l'objet jQuery. Elle associe un événement à tous les éléments sélectionnés et exécute la fonction fournie quand l'événement se déclenche.

Le code suivant génère le même message :

```
<a href="#" onclick="alert('Hello world');">Lien</a>
```

La différence est évidente :

- Nous n'avons pas besoin d'écrire un événement *onclick* pour chaque élément ;
- Nous obtenons une séparation claire entre la structure (HTML) et le comportement (JavaScript) de la même manière que votre feuille de style CSS est dissociée de votre code HTML.

2. Présentation des sélecteurs

2.1. Les sélecteurs

La première chose à apprendre de jQuery afin d'être en mesure de séparer vos codes JavaScript et (X)HTML sont les sélecteurs. Les sélecteurs permettent de sélectionner un ou plusieurs éléments du DOM, ces derniers pouvant être manipulés comme tout autre objet. Si vous êtes familier avec CSS, vous allez saisir rapidement les sélecteurs. Ceux-ci sont en réalité la même chose et ils utilisent presque la même syntaxe. jQuery permet la sélection des éléments grâce à une fonction spéciale nommée « `$` ».

Sélecteurs

```
$(document) // Extension de l'objet document à la classe jQuery.
```

```
$('.*) // Sélectionne tous les éléments.
```

```
$('#monDiv') // Sélectionne l'élément ayant l'ID "monDiv".
```

```
$('.p.first') // Sélectionne les éléments <p> ayant la classe "first".
```

```
$('.p[title]') //Sélectionne les éléments <p> ayant un attribut "title".
```

```
$('.p[title="Bonjour"]') // Sélectionne les éléments <p> dont l'attribut title est "Bonjour".
```

```
$('.p[title!="Bonjour"]') // Sélectionne les éléments <p> dont l'attribut title n'est pas "Bonjour".
```

```
$('.p[title^="H"]') // Sélectionne les éléments dont l'attribut title commence par "H".
```

```
$('.p[title$="H"]') // Sélectionne les éléments dont l'attribut title fini par "H".
```

```
$('.p[title*="H"]') // Sélectionne les éléments dont l'attribut title contient "H".
```

```
$('.ul, ol, dl') // Sélectionne les éléments <ul>, <ol> et <dl>
```

```
$('.div.desc') // Sélectionne les éléments ayant la classe "desc" descendants (au sens CSS) d'éléments <div>.
```

```
$('.div > .enfant') // Sélectionne les éléments ayant la classe "enfant" enfants d'éléments <div>.
```

```
$('#label + input') // Sélectionne les éléments <input /> dont l'élément précédent (dans le DOM) est <label>.
```

```
$('#debut ~ div') // Sélectionne les éléments <div> frères se situant après l'élément dont l'id est "debut".
```

2.2. La sélection par filtre

jQuery offre une très large possibilité de sélection d'éléments en fonction de filtres sur des collections d'éléments.

Le fonctionnement de la sélection par filtre est simple : on sélectionne d'abord un ensemble d'éléments (par défaut, tous) puis on affine cette sélection à partir de certains critères.

Pour bien comprendre les notions de filtre, il est important d'avoir des connaissances minimum sur la structure du DOM (Document Object Model). Pour en savoir plus sur le DOM ([Lien72](#)).

Filtres

```
// La notation [a|b] signifie que l'on applique le filtre a ou le filtre b.
```

```
$('#div:first') // Sélectionne le premier élément <div>.
```

```
$('#div:last') // Sélectionne le dernier élément <div>.
```

```
$('#div:not(.ok)') // Sélectionne les <div> n'ayant pas la classe "ok".
```

```
$('#div:[even|odd]') // Sélectionne les éléments <div> de rang [pair|impair] (le premier rang est 0).
```

```
$('#div:[eq|lt|gt](n)') // Sélectionne le ou les éléments <div> de rang [égal|inférieur|supérieur] à n.
```

```
$('.header') // Sélectionne les éléments <h1>.
```

```
$('.animated') // Sélectionne les éléments actuellement animés.
```

```
$('#div:contain('dvp')') // Sélectionne les éléments <div> contenant le texte "dvp" (sensible à la casse !)
```

```
$('#div:empty') // Sélectionne les éléments <div> vides.
```

```
$('#div:has(p)') // Sélectionne les éléments <div> ayant un descendant <p>.
```

```
$('#div:parent') // Sélectionne les éléments <div> ayant des enfants (y compris les noeuds texte).
```

```
$('#div:nth-child([n|even|odd|equation]')') // Les enfants de <div> [de rang n|pairs|impairs|résultat de].
```

```
$('#div:[first-child|last-child]') // Les éléments [premier|dernier] enfants d'un élément <div>.
```

```
$('#div:only-child') // Les éléments qui sont les seuls enfants d'un élément <div>.
```

2.3. Les éléments de formulaire

Pour terminer, jQuery possède quelques filtres spécifiques à la sélection d'éléments de formulaires en fonction de leur nature ou de leur état.

Eléments de formulaires

```
$('#:input') // Tous les éléments <input />, <textarea>, <select> et <button>.
```

```
$('#:[text|password|radio|checkbox|submit|image|reset|button|file|hidden]') // Les <input /> du type choisi.
```

```
$('#:[enabled|disabled|checked|selected]')
```

```
// Les <input /> possédant l'attribut indiqué.
```

3. Utilisation des sélecteurs et des événements

jQuery nous propose 2 approches afin de sélectionner des éléments.

La première utilise une combinaison de sélecteurs CSS et XPath passés comme chaîne de caractères au constructeur jQuery (comme par exemple `$("#div > ul a")`).

La seconde méthode, quant à elle, se sert de différentes fonctions de l'objet jQuery. Les deux approches peuvent être combinées.

Pour essayer certains de ces sélecteurs, sélectionnons et modifions, par exemple, une liste ordonnée `` insérée dans une page.

Tout d'abord, sélectionnons la liste elle-même. Cette liste a comme id « listeOrdonnee ». En JavaScript classique, vous pourriez la sélectionner avec `document.getElementById("listeOrdonnee")`. Avec jQuery, cela devient :

```
$(document).ready( function()
{
    $("#listeOrdonnee").addClass("rouge");
}
);
```

Admettons que notre feuille de style propose une classe « rouge » qui ajoute simplement un fond rouge. Donc, si vous ne rechargez pas la page dans votre navigateur, vous devriez voir que la première liste ordonnée obtient un fond rouge. Travaillons maintenant avec les enfants de cette liste.

```
$(document).ready( function()
{
    $("#listeOrdonnee > li").addClass("blue");
}
);
```

Ce dernier sélectionne tous les enfants `` de l'élément avec l'id « listeOrdonnee » et leur ajoute la classe « bleu ».

Quelque chose d'un peu plus complexe maintenant : nous voulons ajouter et enlever la classe quand l'utilisateur survole l'élément ``, mais seulement pour le dernier élément de la liste.

```
$(document).ready( function()
{
    $("#listeOrdonnee li:last").hover( function()
    {
        $(this).addClass("Rouge");
    },
    function()
    {
        $(this).removeClass("Rouge");
    }
    );
}
);
```

De nombreux autres sélecteurs à la syntaxe similaire à CSS ou XPath existent. Des exemples supplémentaires et une liste de toutes les expressions disponibles se trouvent ici ([Lien73](#)), sur le site web de jQuery.

Pour chaque événement existant, tels que `onclick`, `onchange`, `onsubmit`, etc., il existe un équivalent jQuery. D'autres événements comme `ready` et `hover` sont fournis par commodité pour certaines tâches afin de vous rendre la vie plus facile. Vous pouvez trouver une liste complète des événements sur Visual

jQuery ([Lien73](#)) dans la section Events du site web. Grâce à ces sélecteurs et ces événements, vous pouvez déjà faire plein de choses, mais il y a mieux soyez en sûrs.

```
$(document).ready( function()
{
    $("#listeOrdonnee").find("li").each( function(i)
    {
        $(this).html( $(this).html() + " WOW! " + i );
    }
    );
});
```

- **find()** vous permet des recherches supplémentaires parmi les descendants de l'élément déjà sélectionné. Ainsi, `$("#listeOrdonnee").find("li")` est tout à fait similaire à `$("#listeOrdonnee li")`.
- **each()** parcourt chacun des éléments et permet des traitements supplémentaires. La plupart des méthodes, telles que `addClass()`, utilisent `each()` elles-mêmes. Dans cet exemple, `html()` est utilisé pour récupérer le texte de chaque élément ``, lui ajouter du texte et le réaffecter comme texte de ces éléments.

Une autre tâche que vous avez fréquemment à effectuer est d'appeler des méthodes sur des éléments DOM non couverts par jQuery. Imaginez un formulaire que vous voudriez réinitialiser lorsqu'il a été traité correctement par soumission AJAX.

```
$(document).ready( function()
{
    // Réinitialiser un formulaire unique
    $("#reset").click( function()
    {
        $("#form")[0].reset();
    }
    );
});
```

Le code précédent sélectionne l'élément ayant l'id « form » et appelle la méthode `reset()` du premier élément sélectionné. Si vous aviez plusieurs formulaires, vous pourriez aussi faire ceci :

```
$(document).ready( function()
{
    // Réinitialiser plusieurs formulaires
    $("#reset").click( function()
    {
        $("form").each( function()
        {
            this.reset();
        }
        );
    }
    );
});
```

Cela sélectionne tous les formulaires au sein de votre document, les parcourt et appelle la méthode `reset()` pour chacun d'entre eux.

Un autre besoin que vous pouvez rencontrer est de ne pas sélectionner certains éléments. jQuery fournit `filter()` et `not()` pour cela. Alors que `filter()` réduit la sélection aux éléments qui

correspondent à l'expression du filtre, `not()` fait le contraire et enlève tous les éléments correspondant à l'expression. Imaginez une liste non ordonnée où vous voudriez sélectionner tous les éléments `` qui n'ont pas d'enfant ``.

```
$(document).ready( function()
{
    $("li").not("[ul]").css("border", "1px solid black");
});
```

Cela sélectionne tous les éléments `` puis retire de cette sélection tous ceux qui ont un `` comme enfant. Ainsi, tous les éléments `` récupèrent une bordure, hormis ceux qui ont un enfant ``. La syntaxe de `[expression]` est issue de XPath et peut être utilisée pour filtrer par éléments enfants et par attributs.

D'une autre part, il y a des situations où l'on a besoin de sélectionner les éléments précédents ou suivants, dénommés « siblings » (frères et soeurs : enfants du même parent). Imaginez une page de FAQ, où toutes les réponses sont cachées au départ et affichées lorsque la question est cliquée.

```
$(document).ready( function()
{
    $('#faq').find('dd').hide().end().find('dt').click(
function()
{
    var answer = $(this).next();
    if (answer.is(':visible'))
    {
        answer.slideUp();
    }
    else
    {
        answer.slideDown();
    }
}
);
});
```

Nous utilisons des enchaînements pour réduire la taille du code et obtenir de meilleures performances, puisque `#faq` ne sera sélectionné qu'une seule fois. En utilisant `end()`, le premier `find()` est annulé, ainsi nous pouvons nous mettre à chercher avec le prochain `find()` pour notre élément `#faq` plutôt que les enfants `<dd>`. À l'intérieur de l'événement « click », nous utilisons `$(this).next()` pour trouver le « sibling » suivant à partir du `<dt>` courant. Cela nous permet de sélectionner rapidement la réponse suivant la question cliquée.

4. En conclusion

La force de jQuery réside donc manifestement dans la richesse de sa fonction `$`, d'autant plus qu'en dehors de la richesse de ces sélecteurs, vous pouvez bien sûr les combiner entre eux pour atteindre n'importe quel élément du DOM très facilement.

Tous les sélecteurs CSS vont fonctionner avec jQuery et ce dernier permet encore plus de possibilités que CSS.

Retrouvez l'article de Dave Lizotte et Didier Mouronval en ligne : [Lien74](#)

À la découverte de script.aculo.us

Cette série d'articles a pour but de vous faire découvrir les différentes fonctions du framework ([Lien75](#)) script.aculo.us. Dans un premier temps, nous allons voir comment mettre en place le framework dans une page HTML.

1. Avant-propos

Avec l'avènement de ce que l'on appelle (avec du reste beaucoup de mal à le définir !) le Web 2.0, les développeurs Web se tourmentent de plus en plus vers les frameworks JavaScript pour concevoir et dynamiser leurs pages.

Parmi ces bibliothèques, Prototype ([Lien76](#)) est particulièrement appréciée. Cependant, Prototype est essentiellement axée vers la simplification de la syntaxe, la navigation dans le DOM, les appels AJAX et surtout la portabilité du code sur différents navigateurs.

Afin d'ajouter du dynamisme et de l'ergonomie aux pages, script.aculo.us ([Lien77](#)) a été créée, se basant sur la puissance de Prototype et en y ajoutant de nombreuses fonctionnalités que nous détaillerons par la suite.

Tous les exemples de cette série d'articles sont valides XHTML 1.0 transitional et CSS 2.1 et ont été testés sous : Internet Explorer 6 et 7, Firefox 2 et 3, Safari 3, Opera 9.6, Netscape 9 et Chrome.

2. Intégrer script.aculo.us à sa page HTML

La première des choses à faire est bien entendu de télécharger l'archive du framework disponible ici ([Lien78](#)) puis de la décompresser !

2.1. Pourquoi utiliser script.aculo.us ?

Tout d'abord, avant d'essayer d'aller plus loin, la première question à se poser est la suivante : "Ai-je réellement besoin de cette librairie pour mes pages ?"

En effet, différentes motivations peuvent inciter à utiliser un framework, des bonnes et des mauvaises. Or ces bibliothèques imposent de télécharger (parfois à chaque page en fonction des réglages du serveur et/ou de la configuration du cache du navigateur) jusqu'à 300ko de fichiers, ce qui peut ralentir la navigation sur votre site.

Parmi les bonnes raisons :

- Utiliser beaucoup de fonctionnalités complexes du framework.
- Mutualiser des fonctions génériques sur plusieurs pages.
- Être à l'aise avec JavaScript, mais vouloir optimiser son code dans un aspect "cross-browser".
- Savoir faire un ratio facilité de coder / difficulté de reproduire les fonctionnalités.

Parmi les mauvaises raisons :

- Apprécier un raccourci de code ou une fonctionnalité !
- Méconnaissance de JavaScript.
- Croire que le framework vous permettra de surmonter les difficultés à coder.

A titre d'exemple :

```
Event.observe(window, 'load', function(){
  $('une_image').setOpacity(0.5);
  $('une_image').observe('mouseover', function(){
    new Effect.Opacity('une_image', {from: 0.5, to:
1});
  });
  $('une_image').observe('mouseout', function(){
    new Effect.Opacity('une_image', {from: 1, to:
0.5});
  });
});
```

Voir l'exemple ([Lien79](#))

aura le même comportement que

```
var IsIE = !!document.all;
function fadePic(elt, debut, fin){
  if(debut == fin){ return false }
  IsIE ? elt.filters[0].opacity=debut :
elt.style.opacity=debut/100;
  debut > fin ? debut -= 2 : debut += 2;
  setTimeout(function(){ fadePic(elt, debut, fin); },
10);
}
window.onload=function(){
  var elt = document.getElementById('une_image');
  if(IsIE){
    elt.style.filter = 'alpha(opacity = 50)';
    elt.filters[0].opacity = 50;
  }
  else{ elt.style.opacity = '0.5'; }
  elt.onmouseover = function(){ fadePic(elt, 50, 100)
}
  elt.onmouseout = function(){ fadePic(elt, 100,
50) }
}
```

Voir l'exemple ([Lien80](#))

Nous voyons bien ici qu'intégrer une librairie comme script.aculo.us (presque 250ko et 8000 lignes de code) n'est pas forcément optimisée pour n'utiliser que quelques fonctions réalisables par soi-même pour peu que l'on maîtrise un minimum JavaScript et la compatibilité des navigateurs !

En revanche, dans de nombreux cas, les fonctionnalités vous aideront à concevoir et coder vos pages avec énormément d'efficacité.

2.2. Présentation de l'archive

L'archive fournie dans le lien est supposée contenir les fichiers et dossiers suivants :

Contenu de l'archive

- 3 fichiers (CHANGELOG MIT-LICENCE et README)
- Répertoire 'lib' ;
- Répertoire 'src' ;
- Répertoire 'test'.

Les fichiers, bien que sans extension, sont des fichiers texte dont je vous invite à prendre connaissance.

Le répertoire 'test' contient des pages HTML permettant de faire tourner des tests unitaires ou fonctionnels.

Le répertoire 'lib' contient la bibliothèque Prototype, elle est indispensable pour pouvoir utiliser script.aculo.us. Pour cet article, c'est la version 1.6.0.3 de Prototype et 1.8.2 de script.aculo.us qui ont été prises comme référence.

Le répertoire 'src' contient les fichiers source de script.aculo.us, organisés par modules.

2.3. Les différents modules

Le fonctionnement de script.aculo.us se fait par un système de modules. Chaque module regroupant un ensemble de fonctions d'un même type. L'intérêt de cette structure étant de pouvoir ne charger que les modules dont on a réellement besoin comme nous le verrons au chapitre suivant.

Les différents modules sont les suivants :

Modules script.aculo.us

- effects.js : gère les fonctions relatives aux effets ;
- controls.js : gère essentiellement les fonctions d'auto-complétion et de champs éditables ;
- dragdrop.js : gère les fonctions de glisser déplacer ;
- slider.js : gère les fonctions de barres de mesure ;
- builder.js : gère la création d'éléments du DOM ;
- sound.js : gère l'intégration de sons.

Le fichier unittest.js ne faisant pas à proprement parler partie du framework, il ne sera pas abordé dans cet article.

Le fichier de base du framework, 'scriptaculous.js' ne contient aucune fonction importante. Il n'est en fait qu'un loader utilisé pour vérifier la compatibilité avec la version de Prototype chargée et charger les modules désirés (par défaut, tous les modules). L'originalité de la librairie étant de pouvoir indiquer en paramètre du fichier les modules à installer.

2.4. Intégrer le script à la page

Il ne reste donc plus qu'à insérer les scripts dans la page HTML. Pour cela, nous allons insérer les balises <script> dans le head de notre page.

Contrairement à d'autres frameworks, les différents modules sont indépendants entre eux (à l'exception de *controls.js* et *dragdrop.js* qui utilisent des effets de *effects.js*), ainsi, aucune fonction de script.aculo.us ne nécessite la présence d'un module autre que celui qui la contient.

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
  <meta http-equiv="Content-Type" content="text/html;
charset=iso-8859-1" />
  <title>scripaculous</title>
  <script type="text/javascript"
src="js/prototype.js"></script>
  <script type="text/javascript"
src="js/scriptaculous.js"></script>
</head>
```

Nous avons évoqué la possibilité de ne charger que certains modules, dans ce cas, si vous souhaitez par exemple ne charger que les fichiers effects.js et dragdrop.js, il suffit de remplacer la seconde ligne par :

```
<script type="text/javascript"
src="js/scriptaculous.js?
load=effects,dragdrop"></script>
```

et le script se charge du reste !

Il est bien évidemment possible de charger directement les scripts, mais d'une part, pourquoi se priver d'une fonctionnalité créée spécialement pour cela et d'autre part, encore une fois, la vérification de la compatibilité Prototype / script.aculo.us est bien pensée pour les "étourdis" !

Il existe une archive très pratique, nommée protopack, qui contient beaucoup de choses utiles :

- Les dernières versions des fichiers ;
- Des versions avec différents niveaux de compression des deux framework (appelées protoculous) ;
- Différentes versions de protoculous intégrant uniquement le fichier effects ;
- Une version faiblement compressée (shrinkvar) de protoculous, facilement éditable pour se créer sa librairie personnalisée.

Cette archive est disponible ici ([Lien81](#)) mais ne comprend actuellement que la version 1.8.1 de script.aculo.us.

D'autres fichiers utiles peuvent être trouvés ici ([Lien82](#))

3. En conclusion

Maintenant que script.aculo.us est chargé dans votre page, il ne reste qu'à utiliser ses fonctions !

Nous partons ici du principe que vous connaissez les bases de Prototype, si ce n'est pas le cas, je vous invite à prendre connaissance de cet article ([Lien83](#))

Les prochains articles de cette série aborderont les différents modules en détail.

4. Articles de la série

A la découverte de script.auto.us ([Lien84](#))

Les effets visuels de script.aculo.us ([Lien85](#))

Retrouvez l'article de Didier Mouronval en ligne : [Lien84](#)

Les derniers tutoriels et articles

Compte rendu de l'Agile Tour 2008

Avec 7 étapes à travers la France et la Suisse, l'Agile Tour 2008 a parcouru le territoire pour parler Agilité avec les acteurs locaux pendant tout le mois d'octobre. Les organisateurs (Patrice Petit, Eric Lefevre, et Ramiro Sarmiento) ont fait preuve d'agilité en concrétisant leur volonté d'aller à la rencontre des régions plutôt que d'organiser un évènement central dans la capitale.

1. Introduction

1.1. Qu'est-ce que l'Agile Tour

L'Agile Tour est une suite d'événements Gratuits répartis sur plusieurs villes européennes pendant tout le mois d'octobre 2008. L'édition 2008 est organisée sur 7 conférences réparties en France et en Suisse. Ces 7 conférences se sont déroulées en plusieurs temps :

- 01 octobre : Besançon
- 03 octobre : Mulhouse
- 06 octobre : Genève
- 09 octobre : Grenoble
- 14 octobre : Lille
- 16 octobre : Toulouse
- 23 octobre : Valence

Son objectif étant de décentraliser les séminaires agiles de Paris vers les autres régions industrielles afin de promouvoir l'agilité, de la populariser, de la soutenir et d'inciter le développement de la culture de l'Agilité.

1.2. Le public

Le sondage mis en place par les organisateurs a eu un succès relatif avec (à l'heure de la publication) 167 votes (il est encore temps de participer ([Lien86](#))).

Voici un résumé des résultats ([Lien87](#)) :

A quel stade en êtes vous dans l'adoption de l'approche agile dans votre entreprise ?

- Pas connue : 20%
- Pas d'utilisation : 12%
- En cours d'analyse : 15%
- Analysée et Rejetée : 1%
- Projets Pilotes : 5%
- Mise en Oeuvre Partielle (Utilisation de certaines techniques agiles) : 25%
- Adoption Partielle (Certains projets utilisent ces méthodes) : 11%
- Adoption Totale (Tous les nouveaux projets utilisent ces méthodes) : 11%

Un public relativement hétérogène donc.

2. Etape Lilloise du 14 octobre

L'Agile Tour fait étape à Lille le 14 octobre, dans les locaux de Telecom Lille1, sur le campus de l'USTL de Villeneuve d'Ascq. Les 11 conférences & ateliers ont été répartis parmi les trois salles selon un vote préliminaire des participants : sur un grand tableau blanc, chaque participant a collé une gommette en face du nom des conférence qu'il souhaitait voir.

L'organisation n'était donc pas fixée à l'avance, et le planning s'est construit pour maximiser la satisfaction du public, pendant que nous assistions à la présentation de l'Agile Tour dans sa globalité. De cette présentation est ressortie la volonté de décentralisation (l'Agile Tour ne fait pas étape à Paris volontairement), l'envie de dynamiser les initiatives locales, et celle de proposer une organisation suivant les préceptes agiles.

Les conférences se sont ensuite enchaînées avec une ponctualité à toute épreuve.

Néanmoins, des désistements parmi les orateurs ont causé de nombreux remaniements dans liste des sessions proposées.

2.1. Intégration continue & offshore (Thomas Recloux), commenté par Hervé Delannoy

L'objectif : améliorer les projets utilisant l'offshore

2.1.1. Une pratique courante : la division

Une des organisations traditionnelles pour un projet en offshore est une équipe Analyse/Conception en local, et le développement est délocalisé. Cette organisation s'oppose au principe agile de favoriser les contacts humains. Une fracture se produit ainsi entre le client et les développeurs, et une seconde fracture s'installe entre le tandem Analyse/Conception et les développeurs. Elle n'est pas adaptée à des cycles incrémentaux et itératifs : la réalisation se déroule en cascade, avec en premier l'analyse, puis la conception et enfin le développement. D'autre part, cela entraîne généralement des débats longs et coûteux autour des dossiers.

Une autre organisation parfois rencontrée est une équipe complète en local, et une équipe de développeurs distants, en support. Cette manière de faire pose de gros problèmes à l'équipe distante : elle ne voit (presque) jamais le client, il est nécessaire de réexpliquer les dossiers, les contacts sont généralement limités. Cela génère au final des livrables de moins bonne qualité, parfois inadaptés au besoin, induit des délais plus longs, et peut donner un sentiment d'isolement et donc de désimplication à l'équipe distante.

2.1.2. Des contacts et du partage pour améliorer la qualité

Une des solutions possible serait, non pas de faire deux équipes (une locale et une distante), mais une unique dont certains membres sont éloignés.

Cela ressemble à un mauvais jeu sur les mots, mais les différences sont précises : les engagements de travail, de délais sont identiques pour tous les membres, les membres éloignés ne sont pas toujours les mêmes, les livrables et livraisons sont partagés.

2.1.3. Trois conseils et exemples pratiques

Tout d'abord, favoriser les rencontres physiques, dans les deux sens. Faire venir les membres distants durant la première itération permet d'établir un contact et améliorer les communications suivantes. Durant les itérations suivantes, instaurer un roulement afin que les membres éloignés ne soient pas toujours les mêmes (autant parmi les développeurs, que parmi les analystes, concepteurs, chefs, testeurs...) casse le sentiment d'isolement, permet de faciliter les compréhensions fonctionnelles et améliore les retours.

Investir dans la communication est essentiel pour garantir un bon feedback et une bonne transmission des informations. Rien n'est de trop : téléphone, visioconférence, messagerie instantanée, tableau blanc virtuel (pendant électronique du grand tableau couvert de notes, indispensable dans un projet agile). Des réunions en équipe complète sont elles aussi très importantes. Une par itération semble une bonne fréquence, lors des réunions charnières inter-itérations.

Instaurer l'intégration continue assure la propriété collective du code, et permet de toujours proposer une application testable en permanence. A terme, les coûts et les délais s'en trouvent réduits.

2.1.4. Quatre étapes pour mettre en place l'intégration continue

La mise en place d'un contrôle de code source unique pour tous les membres de l'équipe est la toute première chose à faire. Il ne faut pas le considérer comme un outil de sauvegarde, mais bien un outil de publication de son travail. Idéalement fonctionnant en mode déconnecté, il permet à tous de travailler en tout temps sur chaque fichier. On peut notamment citer SVN et TeamFoundationServer en mode connecté, et Mercurial en mode déconnecté.

Vient ensuite l'ajout d'un référentiel des demandes client. Pour garantir un suivi parfait, ce référentiel de demandes doit se lier au contrôle de code source afin de repérer quelle demande a occasionné quelle modification du code, et inversement. Trac est un très bon exemple d'outil permettant de définir des jalons, de poser des demandes, et de relier des tâches au code y répondant. Il s'intègre à SVN notamment.

Afin d'améliorer la productivité et de se focaliser sur la valeur ajoutée (au sens agile du terme, soit les fonctionnalités livrées et utiles), il est conseillé d'automatiser le plus possible tout ce qui a une faible valeur ajoutée. Par exemple, l'automatisation de la compilation et du déploiement n'est pas évidente dans tous les projets. Peut venir ensuite la génération du code technique et le lancement automatisé de batteries de tests. Maven est l'un des outils qui permet de faire tout cela en une opération.

Un automate d'intégration continue comme Continuum vient parfaire cette panoplie d'outils. Proche de Maven, il surveille le contrôleur de code source, lance des compilations / tests / déploiements chez le client de manière automatique et publie les résultats. Avec l'intégration continue s'installe une dynamique dans l'équipe, sous forme de jeu (un gage au premier qui empêchera la publication automatique, durer le plus de temps avant qu'elle n'échoue...)

2.2. Spécifications exécutables avec GreenPepper (Eric Mignot & Laurent Cobos), commenté par Hervé Delannoy

On parle souvent de tests unitaires, parfois automatisés. Mais quid des tests fonctionnels ? Généralement, ces spécifications sont

répertoriées dans un CDCF, une expression des besoins, un document pas toujours électronique et en langage naturel.

A l'autre bout de la chaîne, les testeurs établissent des scénarii de test, pas forcément en adéquation pour diverses raisons (erreurs, incompréhensions...).

Au milieu, les développeurs, qui doivent jongler entre les deux en comblant les lacunes potentielles dues au langage naturel. L'objectif des spécifications fonctionnelles exécutables est triple : être non ambiguës, parfaitement lisibles par un analyste et se vérifier très rapidement.

Ces spécifications s'intègrent parfaitement dans une optique TDD, en plus des tests unitaires. Elles s'écrivent en binôme voire trinôme (testeur/analyste/développeur). L'analyste exprime le besoin, les seuils ; le testeur construit le scénario ; et le développeur les traduit dans l'outil, de manière très visuelle et lisible (présentation sous forme de tableau ou de wiki dans GreenPepper).

Dans la foulée, le développeur code ce que l'on peut appeler Fixture. Cette fixture est un pont entre GreenPepper et le code de production. Ce sont les fonctions de la fixture qui seront appelées par GreenPepper, elles encapsulent les appels au modèle objet mis en production, et indiquent si tout c'est bien déroulé. Ainsi, l'outil est en mesure de lancer toute une batterie de tests afin de valider les diverses règles et scénarii spécifiés.

Une fois que tous les tests sont validés, on peut considérer que le coeur de l'application est terminé (il reste les tests ergonomiques à valider).

Tout cela peut intervenir dans une optique d'intégration continue et favorise les contacts analyste/développeur et analyste/testeur.

2.3. Les pratiques d'ingénierie incrémentale (Eric Mignot & Laurent Cobos), commenté par Ashgenesis

Les approches agiles nous proposent d'adopter une approche incrémentale dans la construction de nos logiciels ; de réfléchir à notre définition de "terminé" et d'étendre progressivement cette définition. Le besoin d'évolution de nos pratiques se fait alors vite sentir : que deviennent les longues phases de modélisation ? De recette ? Les livres nous demandent d'atteindre la qualité production dans chaque itération. Cela demande parfois beaucoup de changement. Quelles sont les pratiques rapidement adoptables ? Comment intégrer cette amélioration continue en cours de projet ?

Cette session s'est déroulée aussi de manière participative où tout le monde avait son mot à dire. Un partage d'expérience très fructueux sur les différents points de vues du Pair-Programming (Programmation à 2, ou appelé aussi binôme). Cette pratique a été débattue pendant toute la séance. Sur les avantages et inconvénients de la pratique comme la question de rentabilité, le fait de "bloquer" deux développeurs sur la même machine divise-t-on alors le rendement de production par deux. La qualité du code et la facilité de compréhension par le binôme qui finalement ne fait plus qu'une seule et même entité. La compatibilité entre les 2 développeurs composant ce binôme est-elle importante.

La question principale qui est revenue assez souvent était le fait que de bloquer deux développeurs sur une seule machine ne favorisait pas le rendement de l'entreprise pratiquant cette méthode. En fait la vision est toute autre le pair programming permet de développer plus rapidement car les bugs sont assez souvent et rapidement corrigés avant mêmes qu'ils ne se

produisent du à la complicité des développeurs. Ainsi la qualité du code étant améliorée, il y a moins de "pertes de temps" à corriger les bugs du programme. La facilité de compréhension d'un code nouveau est aussi évidente deux cerveaux en concert laissent moins facilement la place à des oublis et ainsi il y a vraiment une amélioration de compréhension à ce niveau là. Le code ainsi produit est un code de qualité car doublement réfléchi.

2.4. Le Dojo Randori, commenté par Ashgenesis

Je n'ai malheureusement pas pu participer à cet atelier mais je vais faire une brève introduction sur le dojo de programmation. Un dojo de programmation est identique à un dojo d'arts martiaux à la différence près qu'on pratique l'art de la programmation. Ce concept a été initié en France par, entre autres, Laurent Bossavit et Patrice Petit. Son but étant de progresser, d'apprendre, et de partager ses connaissances avec les différents participants, qui peuvent être de tous niveaux, à l'aide des principes et des méthodes de l'eXtreme Programming.

Le dojo se pratique de deux façons différentes en Randori ou en Kata. Le randori permet à tous les membres de pratiquer alors que le kata ressemble plus à un cours magistral. Le dojo commence par un tour de table où chacun propose un sujet en expliquant un peu le contenu du problème et la manière dont il souhaiterait le réaliser (en mode Kata ou Randori). Une fois les sujets évoqués un vote est fait et le problème intéressant la majorité est retenu.

- Le mode kata est assez simple une personne ou un binôme conçoit le programme sous le regard averti des autres participants qui peuvent interagir avec la personne effectuant le kata.
- Le mode randori oblige tous les intervenants à participer à la création du programme chacun prend à tour de rôle le clavier et la souris et endosse le rôle de développeur les autres continuant leurs réflexions et commentaires sur le code si nécessaire.

Le développement est piloté par Test Unitaire. C'est à dire, pour un randori, qu'au tour de chacun, le participant doit écrire la partie de code qui fait passer le Test Unitaire et écrire un test afin de passer le clavier à son voisin dans les mêmes conditions qu'il l'a reçu. Après 1h45 de développement, on passe à la phase de débriefing où l'on discute des difficultés rencontrées, de l'intérêt du sujet et des conclusions à en tirer. Je tiens à remercier Perrick Penet pour m'avoir fait découvrir cette pratique.

2.5. Les projets Agile - de la réponse aux appels d'offres à la maintenance - retour d'expérience (Oana Juncu), commenté par Ashgenesis

Les sujets liés à cette problématique sont :

- Réponse aux appels d'offre : Comment qualifier l'opportunité de réponse en Agile? Définir l'argumentaire
- Contractualisation : Mais quels sont les engagements à prendre ?
- L'initialisation du projet : le piège à éviter: Ca y est, on se lance dans le développement !
- Les jalons du sprint : (bilan, planning game, rétrospectives) toutes les raisons de les éluder sont mauvaises.
- L'amélioration continue : La régulation de la vitesse, le contrôle de la qualité du code, la dynamique de l'équipe, la rétrospective.
- La mise en production : des tâches incontournables à exécuter.
- La maintenance : quel sens dans un projet AGILE? Comment gérer les priorités de la productions sans contourner l'esprit de l'Agilité ?

Cette session était plutôt sur le type conférence qu'atelier

participatif. L'oratrice présentant un retour d'expérience sur les différentes étapes d'un projet du début à la fin de façon Agile. Pour tous projets les contraintes sont assez similaires :

- Respecter le budget
- Définir l'ensemble d'un périmètre
- Positionnement inattendu
- La pression d'un résultat dans un environnement externe
- Réputé complexe et mouvant
- Quelles garanties

Afin de respecter toutes ces contraintes, il faut éviter les pièges :

- Concevoir tout en même temps, car on s'adaptera en cours de route
- Ne pas mettre en place de modélisation/conception car c'est "development driver"
- On démarre avec ce que l'on a aujourd'hui, on est pressé, car c'est pour ça que l'on a choisi le développement agile
- Le niveau technique de l'équipe est bon, c'est le principal

Comment peut-on faire?

Bien définir ces objectifs, fiabiliser le backlog, constituer l'équipe en rapport avec le projet et la compatibilité des développeurs entre eux. La première itération (ou sprint suivant si on utilise l'eXtreme Programming ou Scrum), le sprint 0 donc, est incontournable et cela permet à l'équipe de se familiariser avec le projet et ainsi de réguler la vitesse (La vitesse n'est pas une mesure de productivité ([Lien88](#))) de l'équipe.

Le sprint 1 devient fiable suivant les résultats du premier sprint. Ce qu'il faut garder en tête c'est la constance des paramètres d'un sprint, sa durée (une durée assez courante est entre 1 et 2 semaines), pas d'échange standard de stories d'un sprint, l'abandon d'une story dans un sprint est toléré mais il ne doit pas y avoir de remplacement : l'équipe ayant effectué le planning game porte le sprint.

La souplesse de la maille via des maillons forts, la fin d'un sprint doit avoir ses stories finies, la vitesse se régule naturellement, on aperçoit aussi une amélioration continue au fur et à mesure des sprints, l'utilisation du Pair Programming aide aussi au développement du projet.

La mise en prod se fait après minimum 3 sprints, elle aura été définie dans le backlog.

Il faut adapter le rythme itératif avec la méthode de planification.

Lors d'un sprint, il est important de penser à la gestion du support de prod.

La maintenance est le moyen de stabiliser un outil.

3. Etape Toulousaine du 16 octobre

3.1. Impressions de Philippe Leménager

J'ai assisté à l'étape toulousaine en tant que novice complet : je ne connaissais pas du tout les méthodes agiles et j'y allais 'pour voir'. J'ai été intéressé par la présentation de Scrum et du logiciel open source développé à l'université de Toulouse : iceScrum.

En tant qu'ancien responsable qualité (pas dans l'informatique) j'ai retrouvé dans cette conférence des méthodes de management de la qualité :

- La roue de Deming de la qualité (Plan, Do, Check, Act) se déploie ici dans les sprints à coups de Develop, Wrap, Review, Adjust mais le principe est le même : l'amélioration continue
- Un intervenant a dit qu'il fallait prendre les modifications comme une chance. La maîtrise des modifications est un élément important des démarches qualité et de la norme ISO 9001
- Un intervenant venu du Canada (Philippe Kruchten, très applaudi) a également parlé des difficultés à la mise en

place des méthodes agiles, et notamment des 'excuses piscine' qui empêchent le lancement de la démarche : "Oui mais chez nous ce n'est pas possible parce que...". Le qualitatif se bat au quotidien contre ce genre de comportement

Au cours de cette conférence, l'accent a été porté également sur l'importance de tenir compte du contexte pour mettre en œuvre les méthodes Agile. Les pratiques seront différentes selon le nombre de personnes impliquées, les relations avec le client, le regroupement de l'équipe au sein d'un même lieu ou non...

Ainsi, le cœur de cible des méthodes Agiles est :

- Une équipe de 5 à 12 personnes ;
- Regroupée sur un seul site ;
- À temps plein ;
- Travaillant sur des systèmes interactifs ;
- À architecture logicielle définie ;
- Pour un projet à criticité faible ou modérée (si ça plante, ça tue des gens ou ça gêne seulement quelques personnes temporairement ?) ;
- Avec un management accommodant ;
- Pour des nouveaux développements.

Si l'on n'est pas dans le cœur de cible, les méthodes agiles restent applicables à condition de :

- connaître son environnement d'organisation ;
- analyser le contexte spécifique du projet ;
- sélectionner et adapter les pratiques selon ce contexte ;
- établir un plan d'adoption des pratiques; le suivre et l'ajuster au cours du projet.

Un détail pour finir : l'amphi mis à disposition était archi-plein !

3.2. Les présentations (en format PDF)

Liste des présentations disponibles (PDF)

- Introduction aux méthodes Agiles ([Lien89](#)), par Olivier Azeau, Claude Aubry et Thierry Cros
- Outils Agiles ([Lien90](#)), par Jean-Marie Damas et Stéphane Maldini
- Retour d'expérience XP chez Igeoss ([Lien91](#)), par David Desmarest
- L'Agilité en situation ([Lien92](#)), par Philippe Kruchten et Claude Aubry

4. Etape Grenobloise du 9 octobre

Ce sont Alexandre Boutin et Patrice Petit qui ont accueilli les participants dans l'amphi nord de l'université de physique, sur le campus de Saint Martin d'Hères, pour cette étape organisée par le Club Agile Rhône-Alpes (CARA) ([Lien93](#)).

Cette étape fût un vrai succès du fait de tous les retours positifs reçus des participants.

Les 201 personnes présentes (soit 2 fois plus que l'objectif initial) ont plébiscité l'invité d'honneur qui a vraiment lancé la conférence sur de très bons rails (excellente prestation de Jérôme Barrand) puis se sont naturellement répartis dans les autres salles pour écouter les différents orateurs qui ont été globalement très appréciés.

Les craintes des organisateurs sur la gestion de l'accueil, la répartition entre les salles et le timing de l'évènement ne se sont pas transformées en problèmes, et les buffets gratuits, pause café et dînatore, ont été appréciés par tous.

4.1. Introduction et Keynote

4.1.1. Commenté par Eric Lefevre

Après l'introduction par Alexandre (Boutin), nous avons eu droit à une bonne keynote par Jérôme Barrand que je ne connaissais pas, de l'Ecole de Management de Grenoble. Il semble avoir eu une réflexion parallèle à celle des Poppendieck ([Lien94](#)) et est tombé sur les mêmes conclusions, mais sans parler spécifiquement du domaine de l'informatique. En tout cas, il parle de Management Agile. Un bon orateur aussi.

Quelques extraits :

- l'agilité, c'est gérer les ruptures en les anticipant
- Tarzan est agile car c'est un roi de la jungle non imposé qui sait gérer les ruptures de lianes
- dans les 30 glorieuses on vendait des objets avec des techniques "toutes choses étant égales par ailleurs", c'est à dire qu'on ne se préoccupait pas du devenir de l'objet ou de la satisfaction profonde du client. Maintenant, on vend du relationnel.

Il a conclu avec 8 principes et 6 attitudes qui ne choqueront pas les connaisseurs de Lean.

4.1.2. Commenté par Aline Paponaud

D'après Jérôme Barrand, l'entreprise et son système d'information, grâce aux informaticiens, c'est devenu un gros truc innommable, avec tout le monde qui est en relation avec tout le monde, où tout va à fond la caisse et bouge tout le temps.

Et puis malgré ça, on reste (trop) souvent dans une structure d'entreprise hiérarchique, avec tout en haut le grand chef qui réfléchit et qui détient l'information et le pouvoir, et qui donne des choses à faire à des exécutants en bas. Et ça c'est historique, c'était comme ça au vieux temps...

C'est pour ça qu'il faut trouver un nouveau truc. Un nouveau mode de pensée et de management des entreprises. Eh oui, l'agilité ce n'est pas que des geeks dans un bureau qui ont une super machine de build qui s'allume en rose. C'est aussi tout un mode de "pensée au boulot", admis et recommandé, et même par des gens qui en ont vu, des choses, dans leur vie.

4.1.3. Commenté par Laurent Farcy

3 faits et dire marquants ont plus particulièrement retenu mon attention lors de cette introduction :

- Atos Origin, l'un des sponsors de la manifestation, réalise un projet stratégique sur 7 ans pour le compte d'ERDF (changement des 35 millions de compteurs électriques chez les particuliers) en mettant en oeuvre une démarche agile ;
- Pyxis, l'invité canadien, qui déclare "faire du logiciel et s'éclater", tout cela grâce aux pratiques agiles ;
- Tout le monde se (re)positionne sur l'agilité: c'est visiblement tendance et sans doute vendeur; ce qui est une bonne et une mauvaise chose à mon humble avis.

A suivi une 'performance' appréciée par le public de la part de Jérôme Barrand, "l'invité du 'dîner de con', toute chose égale par ailleurs...", sur l'agilité, un nouveau paradigme managérial. Mr Barrand ne s'est pas gêné pour accuser l'informatique et les technologies de l'information d'être la cause de tous les maux de notre société actuelle. Provocation qui a réussi à amuser et capter l'attention de l'assistance. Sans doute la séance la plus marquante de l'après-midi.

4.2. Introduction à Scrum (Eric Lefevre)

4.2.1. La session, par son auteur

Ma présentation **Introduction à Scrum**, en 1ère session, s'est bien déroulée. J'ai eu juste le temps d'introduire auprès d'une cinquantaine de personnes les concepts de base, montrer 3 retours d'expérience personnelle et donner quelques billes pour aller plus loin.

4.2.2. La session, commentée par Aline Paponaud

Bon ben voilà, c'est bon, je n'ai plus peur. En fait, on peut se faire le truc à notre sauce ! Pas besoin d'en faire trop d'un coup.

J'ai surtout noté des informations concernant les pratiques d'équipe plutôt que les techniques de gestion de projet avec sprints, backlogs et compagnie (parce qu'on avait vu ça en cours).

L'agilité pour une équipe c'est **Partage, Responsabilité, Confiance** (etc.). Effectivement, pour une équipe de basket ou d'improvisation, c'est une évidence. Mais pour du développement, on n'aurait pas imaginé... Et bien si, en fait c'est pareil ! Et on verra aussi dans la conférence sur les aspects psychologiques que c'est aussi un peu le même fonctionnement qu'en famille ...

- Ce qui passe en premier c'est pas les outils et les procédures compliquées, c'est les gens. Le socle, c'est les individus et l'esprit d'équipe ;
- Avoir **quelque chose qui marche**, c'est mieux qu'une doc super compliquée avec des 1.1.2.a.iv.B) alinéa 3 ;
- Le client c'est notre **partenaire** (parce qu'on est motivés en vrai pour lui, on cherche pas à l'avoir) ;
- On est capable de s'adapter au changement.

Et en fait tout ça, ça existe depuis longtemps, et les premiers à avoir imaginé quelque chose qui s'appelait "Lean Thinking", ce sont des américains. Lean Thinking a été mis en place par Toyota en 1950 pour la première fois. Et en 1987, il a été inventé un processus de développement "en spirale".

Dernière chose que j'ai notée, c'est que la réunion quotidienne préconisée par Scrum, c'est (moins de 15 minutes) :

- Qu'est ce qui s'est passé la veille ?
- Qu'est ce qui va se passer aujourd'hui ?
- Quelles sont mes difficultés ?

4.3. Retour d'expérience - implémentation Scrum / XP (Bruno Orsier)

4.3.1. La session, par son auteur

La présentation que j'ai faite lors de Agile Tour Grenoble est disponible sous forme d'un fichier pdf ([Lien95](#)) à télécharger.

En complément, voici quelques notes pour comprendre les diapos :

4.3.1.1. Introduction

Pratiquer le développement agile sensibilise aux bénéfices de la transparence et de la coopération. Cette présentation est donc proposée dans l'idée de vous permettre de prendre en compte notre expérience et de vous éviter certains écueils que nous avons rencontrés. Elle devrait aussi nous permettre de recueillir des suggestions pour progresser. De toute manière, au minimum, faire cette présentation nous a déjà permis de clarifier certaines de nos idées !

4.3.1.2. Sondage

Pour des raisons de confidentialité il n'est pas possible de donner trop de détails sur les projets en cours. Comme je souhaitais faire tout de même une présentation assez factuelle, j'ai organisé ce sondage interne afin de pouvoir présenter un état des lieux de notre degré d'adoption de Scrum/XP après 3 ans. Le sondage est inspiré de la 3ème enquête annuelle ([Lien96](#)) de la société VersionOne (nous utilisons leur produit pour gérer nos backlogs). 29 personnes (sur les 36 concernées) ont bien voulu répondre à ce sondage anonyme.

4.3.1.3. Question 1 (bénéfices des méthodes agiles)

La première question est directement calquée sur une question similaire de l'enquête V1. Il y a avait 5 réponses possibles ; pour synthétiser j'ai additionné les réponses "il y a une amélioration" et "il y a une amélioration significative". Puis j'ai trié les diverses propositions par ordre décroissant, et pour faire apparaître des zones, j'ai coupé arbitrairement à 75% et 50%. Enfin j'ai ajouté une colonne avec les résultats obtenus par VersionOne afin de comparer notre situation avec celle de la communauté (plus de 2300 personnes ont répondu à VersionOne).

Dans la zone verte, nous avons en général un score plus élevé que dans le reste de la communauté, et c'est vrai en particulier pour "les développements sont faits de manière plus disciplinée" (90% chez nous contre 59% dans la communauté). Il est également très satisfaisant de voir que la R&D et le marketing travaillent mieux ensemble, que l'amélioration de la qualité est largement constatée par nos participants à l'étude.

Dans la zone intermédiaire, les 62% concernant la facilité de maintenance et d'extension sont un peu décevants - mais ils restent du même ordre que les 56% de la communauté. Contrairement à ce que nous imaginions (naïvement sans doute), ces facilités ne sont pas forcément obtenues "gratuitement" en pratiquant Scrum/XP.

Dans la zone orange, le faible score sur l'amélioration du moral de l'équipe (48% contre 74% dans la communauté) est un signal d'alarme fort. C'est d'autant plus inquiétant que c'était l'une des promesses de Scrum (avoir plus de "fun" à travailler). Dans le même ordre d'idée, nos participants ne constatent pas vraiment d'amélioration du temps de mise sur le marché, alors que la communauté voit des améliorations. Ces deux points sont des premières indications de nos points faibles actuels.

Les deux derniers points (réduction des coûts de développement, gestion des équipes distribuées) pourraient également sembler inquiétants, mais nous avons à peu près les mêmes scores que le reste de la communauté. Donc ce n'est pas une indication que nous faisons quelque chose de travers (mais ces faibles valeurs ne sont bien sûr pas satisfaisantes).

4.3.1.4. Question 2 (intérêt des diverses pratiques)

Pour cette deuxième question il n'a pas été possible de comparer nos résultats avec ceux de VersionOne. En effet la question correspondante de VersionOne était plutôt "employez-vous telle pratique ?" ; dans notre contexte il m'a semblé plus pertinent de savoir ce que mes collègues pensaient réellement de ces pratiques qu'ils emploient assez souvent. J'ai donc proposé quatre réponses possibles, et j'ai additionné les réponses "intéressant" et "très intéressant". J'ai également trié et fait apparaître des zones de couleur.

Ce qui saute alors aux yeux c'est que les zones opposées semblent

correspondre exactement aux pratiques XP d'une part et Scrum d'autre part ! Grosso modo les pratiques XP sont jugées intéressantes, alors que les pratiques (ou "cérémonies") Scrum n'ont pas la faveur des participants. C'est bien sûr un autre signal d'alarme fort, et il va falloir comprendre rapidement ce qui cloche !

A ce stade il est encore difficile de donner une explication satisfaisante de cette faible adoption de Scrum. Il semblerait que l'aspect processus discipliné ne passe pas très bien, à moins que ce soit l'aspect amélioration continue (les rétrospectives).

4.3.1.5. Conclusion

En revenant aux 4 principes du manifeste agile, il semble maintenant clair que nous avons été faibles dans l'application du principe "favoriser les personnes et les interactions plutôt que les processus et les outils". En fait nous avons plutôt fait le contraire, en essayant de favoriser la mise en place d'un processus très discipliné. Nous l'avons fait pour de bonnes raisons (obtention de la certification ISO9001 version 2000, exigences de nos clients notamment dans le domaine pharmaceutique) mais ce faisant nous avons certainement négligé l'aspect humain (gestion du changement, passage d'un management directif à l'auto-organisation des équipes).

Remettre au premier plan ces aspects humains est donc l'un de nos principaux challenges actuels.

D'autre part il apparaît maintenant que les équipes de développement ont déjà fait beaucoup de chemin dans l'adoption de bonnes pratiques de développement. Les réserves de progrès sont maintenant peut-être situées plutôt au niveau de l'équipe de management, qui doit probablement adopter elle-même un mode de fonctionnement plus agile, avec plus de transparence, plus de communication, de coopération - tout cela afin de lever plus rapidement les obstacles rencontrés par les équipes. En effet il semblerait que les rétrospectives sont jugées peu intéressantes par les équipes car elles soulèvent régulièrement les mêmes obstacles qui ne sont pas levés suffisamment vite par le management.

Cette présentation illustre donc certaines des conséquences organisationnelles de la mise en place de Scrum/XP : l'équipe de management doit s'adapter !

Comme je fais partie de cette équipe... j'aurais certainement l'occasion d'écrire à nouveau sur la question !

4.3.2. La session commentée par Eric Lefevre

Bruno Orsier de la filiale de Grenoble (production de logiciels) de Varian a expliqué que Scrum & XP les ont aidé à améliorer la qualité, dont l'obtention de la certification ISO 9001. Par contre, ils n'ont pas pu encore améliorer le Time To Market.

Autre chose intéressante: les membres d'équipe apprécient les pratiques XP mais moins les pratiques Scrum. Contradictoire avec le fait que, parmi les inconvénients, ils citent des points d'ingénierie, alors que, parmi les avantages, ils citent des points de gestion de projet.

Une présentation 'candide' de qualité.

4.4. Aspects Psychologiques des méthodes Agiles (Ramiro Sarmiento), commenté par Aline Paponaud

Il y a des aspects psychologiques bien sûr, car les méthodes agiles, comme on l'a vu, sont tournées vers l'être humain. Et

l'objectif, c'est notre bien-être à tous !

Test-Driven Development = l'Apprentissage "Comme dans les réseaux Neuronaux" selon l'animateur.

Effectivement, commencer par le test, et s'améliorer pour réussir le test, c'est un entraînement ou un apprentissage. Cela permet de reconnaître notre objectif et de ne pas le lâcher, ne pas partir dans une autre direction non plus. "Amélioration Continue".

Binôme = Accepter de l'aide, échanger

"Vérifier" devient "S'entraider", "Concevoir" devient "Echanger avec l'autre" ... Que de belles choses !

Une alternative (vient de la méthode "Crystal"): le "Side By Side". On a chacun deux écrans, de manière à ce que le voisin puisse voir ce qui se passe chez l'autre et éventuellement intervenir.

Avantages du binôme :

- Concentration soutenue (peut être aussi un inconvénient parce que ça fatigue)
- Biorythmes psychologiques cumulés (en gros, si y'en a un qui est fatigué c'est l'autre qui gère. ça fait les vases communicants, comme dans un couple !)
- Transparence
- Assurance avec la relecture
- Consensus, capacité à trouver un terrain d'entente
- Apprentissage, partage de connaissance, erreurs

La rétrospective (le meeting quotidien de Scrum) = S'améliorer ensemble

A faciliter par un coach.

Equivalent à un conseil de famille où on propose des exercices à faire en famille pour s'améliorer (comprendre : le daily meeting de Scrum c'est comme Super Nanny !!!)

Impose le respect entre les membres de l'équipe... Ce que chacun dit est personnel et à prendre en compte.

Les artefacts... La machine de build qui s'allume en rose et la peluche sur l'ordi de ManUChenu ont un rôle !

Moi qui croyais que c'était juste pour le fun...

Nous les informaticiens on "fabrique" du vent (des opérations mathématiques basiques, finalement) Contrairement à un vendeur de légumes, un artisan ou un ouvrier. Les artefacts sont là pour matérialiser ce qu'on a fait (notamment les post-it avec les fonctionnalités à réaliser). Ce sont des symboles qui améliorent notre perception... Et donc on se sent mieux.

4.5. Mise en place d'outils pour industrialiser le développement (Emmanuel Hugonnet, Rémy Sanlaville)

4.5.1. La session, par ses auteurs

Notre présentation (slides au format pdf et ppt ([Lien97](#))) est un retour d'expérience sur la mise en place de différents outils d'ingénierie logicielle qui constituent l'infrastructure de développement à Orange Labs (anciennement France Télécom R&D) ainsi que son utilisation par différents projets industriels.

4.5.1.1. Introduction

Pourquoi des outils d'ingénierie logicielle à Orange Labs ? Afin de répondre aux besoins de ses clients et du marché, France Télécom opère depuis plusieurs années une transformation pour devenir un opérateur de services au dessus de son infrastructure réseau. Pour cela, il est nécessaire de professionnaliser le développement afin d'offrir des services reconnus et de qualité.

Outils d'ingénierie logicielle et agilité ? Le manifeste agile ([Lien98](#)) indique en effet : *Individuals and interactions over*

processes and tools. Nous sommes d'accord avec ce point et cela n'est pas contradictoire. Pour nous, l'humain reste au premier plan mais il a besoin d'outils pour réaliser au mieux ses tâches :

- Les outils ne sont qu'un moyen et pas un but ;
- Les outils pour une démarche d'amélioration continue.

4.5.1.2. Le Build

Après une rapide introduction de ce qu'est le build et sa problématique (reproductibilité dans le temps et dans l'espace), nous expliquons pourquoi nous avons choisi Apache Maven 2 ([Lien99](#)) comme outil de build. Nous présentons ensuite la plateforme Maven 2 que nous avons mis en place à Orange Labs, les difficultés rencontrées lors de cette mise en oeuvre et un premier bilan de l'apport d'une telle plateforme.

4.5.1.3. L'Intégration Continue

Après une introduction de ce qu'est l'intégration continue et sa problématique (détecter au plus tôt les problèmes pour les corriger au plus tôt), nous détaillons les différents concepts de l'intégration continue (triplet {événements de déclenchement, appel d'actions, résultats (notifications, rapports, artefacts)}) et son workflow dans le cycle de développement. Nous abordons ensuite les objectifs de la mise en place d'un serveur d'intégration continue à Orange Labs, l'étude de comparaison que nous avons menée pour en faire une recommandation et qui a mené au choix d'Hudson ([Lien100](#)) et enfin un ensemble de bonnes pratiques que nous avons pu identifier.

4.5.1.4. Retour Projets

Nous montrons ici plusieurs retours d'expérience qui permet d'illustrer l'amélioration de la qualité sur différents projets industriels d'Orange.

4.5.1.5. Bilan et Perspectives

Nous finissons par un bilan :

- au niveau de l'agilité : comment nous avons essayé de répondre au manifeste agile afin de passer de la compilation continue à l'exécution continue vers la production continue ;
- au niveau d'Orange Labs : quels sont nos retours d'expérience sur les axes d'améliorations et les difficultés rencontrées pour la mise en place d'outils d'ingénierie logicielle pour industrialiser le développement.

Sur ce que nous avons mis en place, nous avons maintenant un socle qui nous permet d'envisager plusieurs perspectives :

- mise en place d'un tracker pour gérer la traçabilité des exigences et pour structurer le développement (commit par issue...);
- mise en place de sondes/capteurs (prévenir plutôt que guérir) sur l'ensemble de notre chaîne de développement (build, gestion de configuration, intégration continue, tracker...);
- mise en place d'un tableau de bord afin d'améliorer la gestion de projet par le suivi quotidien de l'avancement et de la qualité du développement.

4.5.2. Commenté par Aline Paponaud

J'avais mal lu. Je croyais qu'on allait parler d'agile dans l'industrie... On arrive en retard... ça parle Java, Maven 2, ANT, Eclipse et NetBeans. Sympa, mais on a pas vu l'objectif du début parce qu'on était en retard... En gros :

- Ils ont une chouette infrastructure, avec une appli web, de la doc en ligne, tout ça.
- 5% des bugs (ceux découverts post-release) représentent

95% des coûts de production. Ah ben oui, là, on a peur.

- Les outils qu'ils ont mis en place permettent d'avoir des retours les plus rapides possibles, et c'est la fin du "ça marche chez moi ! Dommage pour toi !!!"

4.6. Agilité et avionique (Emmanuel Chenu)

Commentaire d'Emmanuel Hugonnet :

Depuis le temps que je voulais voir Emmanuel en vrai. C'est là qu'on voit le gouffre entre le monde de l'avionique et celui du SI. L'expression 'application critique' prend tout son sens. Une expérience intéressante car elle donne des clefs pour convaincre nos chers décideurs.

Commentaire d'Aline Paponaud :

Ben moi, XP ou pas XP, j'ai peur dans l'avion. Mais oui, malgré toute la complexité des normes imposées quand on fabrique un drone, un missile, un système de navigation GPS, ou toute machine de guerre destinée à être vendue pour ensuite aller tuer des millions d'innocents (ah oui pardon, on parlait *juste* des beaux systèmes pour les avions de ligne 1ère classe), une équipe d'irréductibles a réussi à appliquer de l'agilité, et ça marche mieux. Ce que j'en retiens en tout cas c'est encore une fois que ce qui est primordial c'est d'avoir une équipe ultra motivée, qui a envie de faire ça.

4.7. Senteurs Agiles (T. Lissajoux), commenté par Emmanuel Hugonnet

Un atelier vraiment enrichissant et un jeu de cartes tout simplement génial. J'ai hâte de pouvoir en obtenir une version :o)).

4.8. Architecture Agile chez Yahoo!/Kelkoo (Arnaud Delafosse), commenté par Eric Lefevre

Pas de grandes révélations ici. L'approche est de "sortir de la tour d'ivoire de l'architecture et d'en faire un travail collectif". Pour simplifier, ils délèguent autant d'autorité que possible aux équipes de développement, en exigeant tout de même un minimum. J'ai le sentiment qu'ils peuvent aller plus loin: bien qu'il explique que le point principal est d'avoir des logiciels modulaires et qui supportent bien la montée en charge, certaines exigences de l'architecture minimum comme les DAO me semblent aujourd'hui trop contraignantes (trop détaillées?). Par ailleurs, ils ne font pas encore d'itérations ou de réflexions régulières sur ce minimum.

Une citation à retenir: "L'architecture agile, c'est jouer au Tetris avec les exigences!" En gros, on prépare au mieux notre système à accepter des changements futurs, mais sans ajouter des fonctionnalités techniques précises.

4.9. eXtreme Programming - Retour d'expérience après 6 années de pratiques (J-M. Voisin), commenté par Emmanuel Hugonnet

Un retour d'expérience fort intéressant où j'ai pu m'apercevoir qu'Allianz était dans la même ligne qu'Orange Labs sur le choix des outils et des pratiques. Le chiffre du jour : un projet en XP entraîne un surcoût de 15% à la réalisation mais l'équipe qui passait de 50% de son temps en maintenance en est à moins de 10% (environ 1 bug par mois).

4.10. Développement Web Agile avec Ruby on Rails (G. Karékinian), commenté par Laurent Farcy

Il est toujours agréable de constater que d'autres font les mêmes paris que vous (en l'occurrence RoR) et ont une (courte) expérience assez similaire. Au sortir de cette présentation, on a pu

sentir la curiosité et l'intérêt suscité par Rails au niveau régional, sans toutefois pouvoir dire si l'adoption augmente significativement.

5. Etape Valentinoise du 23 octobre

Le niveau des sessions, la qualité des échanges ainsi que le buffet ont été au rendez-vous. Le tout a été orchestré d'une main de maître par les organisateurs !

L'amphithéâtre de l'ESISAR était plein : environ 120 participants sont venus partager cet événement.

Patrice Petit a ouvert les débats avec une introduction aux démarches agiles en 50 minutes.

5.1. Atelier « Artistes et Spécifieurs » (Gery Derbier)

Commentaire d'Emmanuel Chenu :

Géry Derbier fait un carton en animant son atelier "Spécifieurs et Artistes".

Il démontre en pratique que le développement est bien un jeu coopératif d'invention et de communication.

Commentaire d'Emmanuel Etasse :

Cet atelier m'a énormément plu. On y découvre de façon ludique les écueils possibles dans la communication entre client et développeurs, et surtout les moyens pour arranger les choses. Je ne regrette pas d'y être allé !

Commentaire d'Alexandre Boutin :

Gery Derbier m'a fait jouer à "Artistes et Spécifieurs" pour me faire réaliser, en autres, qu'il n'y a pas de solutions parfaites pour spécifier quelque chose. Ce jeu Agile m'a beaucoup plu et je vais le travailler pour mieux le comprendre et le reproduire en interne.

5.2. Agile Dojo : L'art de grandir en équipe (Ramiro Sarmiento), commenté par Bruno Orsier

Cela fait plusieurs fois que j'entends parler de ces dojos, donc j'étais très curieux d'en voir un en pratique (avec l'idée d'en animer un moi-même plus tard). Ramiro nous a expliqué que les dojos se pratiquaient sous deux formes :

1. le kata. C'est une séance préparée, durant laquelle quelqu'un programme en public - l'idée étant de montrer le geste parfait (terme évidemment emprunté aux arts martiaux)
2. le randori. C'est une séance moins préparée, durant laquelle le public participe. Un binôme programme sur un PC relié à un projecteur, et un membre du public remplace un membre du binôme toutes les cinq minutes. C'est donc très rythmé et intensif.

J'ai noté quelques autres points de l'introduction de Ramiro :

1. l'idée générale est de pouvoir (enfin !) programmer sans contrainte de type délai à respecter, et donc de pouvoir trouver la meilleure manière. Au minimum, le but est d'aimer ce qu'on a écrit !
2. on essaie de respecter le principe KISS - pour lui *Keep It Simple but not Stupid* - je connaissais plutôt *Keep It Simple, Stupid* mais bon chacun a l'air d'avoir sa propre formule, et wikipedia ([Lien101](#)) mentionne aussi « *Keep It Sweet & Simple* » et « *Keep It Short & Simple* ».
3. on peut passer par des phases appelées "spike", durant lesquelles on explore un sujet (par exemple pour trouver comment quelque chose fonctionne exactement - idéalement par des tests).
4. en principe on ne passe pas trop de temps à faire de la conception, on laisse parler le code et les tests.
5. durant ces séances, on explore aussi des voies d'amélioration comme la réduction de l'utilisation de la

souris (et je suis bien d'accord car je vois de temps en temps des collègues perdre plusieurs minutes à faire des opérations qui pourraient se faire en secondes - le point ennuyeux n'étant pas la perte de temps mais plutôt la perte du fil de pensée).

Suite à cette introduction, nous démarrons sur l'exercice proposé (réaliser un jeu de démineur). Le démarrage est évidemment un peu poussif, d'autant plus que certains (comme moi) ne connaissent pas le langage utilisé (Java).

A la pause je dois malheureusement m'éclipser car je souhaitais écouter la présentation de Alexandre Boutin - c'est l'inconvénient de ces conférences où il y a plusieurs sessions intéressantes en parallèle, et des choix difficiles à faire. Je n'ai pas donc pu voir le randori dans son aspect le plus intensif, mais cette introduction m'a conforté dans l'envie d'en organiser un moi-même. Il y a un ensemble de règles à respecter, voir par exemple ce site ([Lien102](#)) qui me permettra de préparer la future séance.

5.3. Retour d'expérience YAHOO International (Alexandre Boutin)

5.3.1. La session, par son auteur

L'amphi était plein pour ma présentation du retour d'expérience Yahoo sur le passage d'une méthode Waterfall à Agile entre 2005 et 2008. Les premiers feedbacks à chaud étaient très positifs. Par contre, en tant qu'orateur, j'ai bien réalisé la difficulté de tenir sur un format de 50 minutes - ce que nous avons imposé à Grenoble.

5.3.2. Commenté par Emmanuel Etasse

Le retour d'expérience d'Alexandre Boutin sur l'implémentation de Scrum chez Yahoo ! en Europe, Asie et Canada m'a beaucoup plu ! Il y avait question du choc des cultures. On relève au passage qu'Alexandre avait initialement tenté d'implémenter du cycle en V, mais a du retourner sa veste (du bon côté) en choisissant une démarche Agile. Je retiens aussi que la démarche initiale était un processus imposé visant à contrôler, alors que la forme Agile est un *framework* optionnel facilitant l'adoption de Scrum mais sans le rendre obligatoire : « *On peut forcer quelqu'un à suivre un processus, on ne peut le forcer à être Agile...* ». Enfin, une raison obligatoire au succès de la démarche est l'engagement du management.

5.3.3. Commenté par Emmanuel Chenu

Alexandre présente un retour d'expérience sur la transition vers l'agilité chez Yahoo!.

Pragmatisme et franc-parler sont au rendez-vous.

5.4. Le refactoring (Régis Medina)

Commentaire d'Emmanuel Chenu : Pendant 50 minutes, Régis Medina (Crossbow Labs) remanie du code en direct.

Cette session permet de recentrer l'attention des participants sur le (beau) code et les pratiques techniques.

Commentaire de Bruno Orsier : J'ai été très impressionné. En effet je suis familier depuis un moment avec le refactoring, mais je le pratique essentiellement à la main. Régis a fait une démonstration spectaculaire de remaniement de code Java grâce à IntelliJ IDEA. Il a aussi indiqué qu'il était un extrémiste du principe DRY - Don't Repeat Yourself - ca tombe bien, moi aussi (voir mon article ([Lien103](#)) qui essaie de promouvoir ce principe), mais j'ai trouvé mon maître.

Enfin, Régis a expliqué l'une des conséquences du refactoring constant (et impitoyable) sur un gros projet sur lequel il a travaillé

: un concept de plate-forme réutilisable (non prévu au départ) à fini par émerger. Bref, après cette belle présentation, le retour à la réalité - Delphi 7 et son manque d'outils de refactoring - est bien difficile ces derniers jours.

5.5. La méthode Crystal (Gery Derbier)

5.5.1. Commenté par Alexandre Boutin

Retour en Amphi pour écouter Gery Derbier nous parler des principes de Crystal dont les 3 essentiels que sont la livraison fréquente, la communication dans l'équipe et la réflexion pour s'améliorer. Gery a indiqué que les équipes Crystal avaient tendance à abandonner au bout d'un certain temps les itérations à durée fixe pour du KANBAN (Lean). Ce que je retiens surtout de Crystal c'est qu'il faut adapter ses pratiques en fonction du contexte et donc de l'avancement du projet car les besoins sont différents et début, milieu et fin de projet. Cette approche, qui me semble maintenant être une évidence, ne se retrouve pas clairement dans Scrum, sûrement quelque chose à réfléchir et mûrir.

5.5.2. Commenté par Emmanuel Etasse

Gery Derbier a présenté la méthode Crystal de Alistair Cockburn ([Lien104](#)) décrite dans les livres Agile Software Development ([Lien105](#)) et Crystal Clear ([Lien106](#)). J'y ai retenu que cette méthode pouvait s'apparenter à un *framework* où les pratiques concrètes doivent s'adapter en fonction du contexte et de la vie du projet. Ce *framework* s'appuie sur des points théoriques :

- Le développement logiciel est un jeu coopératif d'invention et de communication avec deux objectifs antagonistes : livrer un produit et se préparer pour la partie suivante.
- Le produit délivré par une équipe est à l'image de celle-ci

Crystal s'appuie 3 propriétés principales : livraison fréquente (au moins tous les 6 mois), communication de proximité entre tous les développeurs (moins de 30 sec pour obtenir une info), réflexion tactique pour s'améliorer (une fois minimum tous les 2 mois). Crystal me fait penser à une méta-méthode, relativement "universelle" en fait.

5.5.3. Commenté par Bruno Orsier

Cette présentation a été beaucoup appréciée par Emmanuel et d'autres participants.

Mais j'ai eu un peu de mal à voir les aspects concrets de cette méthode, et j'ai parfois perdu le fil de la présentation.

Crystal semble voir les choses de beaucoup plus haut que Scrum (Emmanuel en parle d'ailleurs comme d'une "méta-méthode"). Je suspecte que la grande réussite de Scrum vient de sa capacité à être rapidement comprise, et de sa mise en oeuvre assez directe. Crystal paraît moins "prescriptive".

En fin de séance, Emmanuel et moi avons interrogé Gery sur le rôle du manager dans un contexte agile. Gery nous a répondu que son **rôle de manager, c'était de produire des équipes** - et donc de travailler sur le recrutement, la formation, la communication. Encore un point à méditer, je ne pense pas que nous ayons atteint ce degré de maturité (comme je l'ai indiqué lors de ma présentation à Grenoble, pour nous le rôle de manager se superpose à celui de "pompier", expert technique etc.).

5.6. Forums ouverts

L'étape Valentinoise a été le lieu de plusieurs *forums ouverts* (Open Space), mais également des discussions de couloir ou autour du buffet.

Alexandre Boutin en a animé un pour les curieux et les sceptiques.

Une petite dizaine de personnes ont contribué à cette session. Les questions traitées concernaient la contractualisation Agile, le rôle et l'intervention des experts, et les difficultés rencontrés lors d'une itération.

Bruno Orsier dans les couloirs :

La pause a été l'occasion d'être présenté par Emmanuel à Laurent Bossavit, avec lequel j'avais été en contact par mail (il avait eu l'amabilité de relire mon tutoriel sur le TDD ([Lien107](#))).

Laurent nous indique que son sujet de prédilection depuis plusieurs années est la **rétrospective**. Ca tombe bien, c'est l'un des sujets qui nous semble délicat dans notre implémentation Scrum/XP. Laurent nous explique une des leçons qu'il a tirées : les actions de la rétrospective ne doivent concerner que les participants à la réunion de rétrospective. Conséquence : il faut parfois inviter des personnes en plus des membres de l'équipe (décideurs, managers, etc.) A méditer, car nos rétrospectives sont en général limitées aux seuls membres d'équipe.

6. Conclusion et remerciements

L'entrain autour des différentes étapes de l'Agile Tour témoigne du succès de cette initiative.

Il est fort à parier que l'édition 2009 sera très attendue, et nous ne pouvons qu'inviter les partenaires locaux potentiels à se manifester auprès des organisateurs, ceci afin de couvrir encore davantage de zones, et pourquoi pas un passage par la Belgique !

Un grand merci donc aux organisateurs globaux et locaux, sans qui la diffusion des courants de pensée agiles n'aurait été possible. Et bien entendu, merci à tous les contributeurs, directs ou indirects, à ce compte-rendu destiné à vous donner une couverture maximale de l'évènement.

Retrouvez l'article de Hervé Delannoy, Rémy Sanlaville, Bruno Orsier, Philippe Leménager, TheLeadingEdge, Ashgenesis, Emmanuel Hugonnet, Aline Paponaud, Alexandre Boutin, Eric Lefèvre, Emmanuel Etasse, Emmanuel Chenu et Laurent Farcy en ligne : [Lien108](#)

Les derniers tutoriels et articles

Les batteries au Lithium-Ion : les comprendre et les utiliser

Notre univers est rempli d'objets utilisant des batteries. Et de plus en plus, celles-ci utilisent la technologie Lithium-Ion. Alors que les technologies évoluent très vite, le fonctionnement de ces batteries est mal connu du grand public, et même de certains experts. Puisque nous avons nous-même usé quelques batteries à ne pas les utiliser correctement, nous avons décidé de mettre sur pied une petite synthèse afin de vous aider à comprendre le fonctionnement des batteries Li-Ion pour mieux les utiliser. En effet, la tendance est au mobile. En France, cette année, il devrait se vendre plus d'ordinateurs portables que d'ordinateurs de bureau. C'est déjà le cas depuis quelques temps pour un constructeur comme Apple, et nous sommes en train de vivre une frénésie ultra-portable, dont l'initiateur a été l'eeePC. Mais de plus en plus d'appareils utilisent des batteries soudées, non remplaçables (comme l'iPod ou l'iPhone), et il est donc impératif d'en prendre soin. Et alors que nous avons à notre disposition des technologies comme le Wi-Fi, le Bluetooth, et bientôt le wUSB ou le WiMax, il serait dommage de se priver de couper le cordon (d'alimentation).

1. Dedans et dehors...

1.1. Un peu d'histoire...

Ce sont les Américains, dès 1912, qui ont commencé à travailler sur les batteries au Lithium, en la personne du chimiste Gilbert Newton Lewis. Les premières piles utilisant cette technologie ont été commercialisées dans les années 1970, tandis que les premières batteries rechargeables (ou accumulateurs) sont apparues dans les années 1980. Seul problème : les batteries au Lithium sont instables, notamment à la charge, et peuvent exploser. Bell Labs a travaillé sur ce problème, tout comme le professeur John Goodenough de l'Université d'Oxford, permettant à Sony d'inonder le marché dès 1991.

1.2. A l'intérieur du dedans...

Une batterie est composée de plusieurs éléments appelés accumulateurs. Ils ont vaguement la forme d'une pile classique (un cylindre), et sont destinés à emmagasiner l'énergie. Peu chers à fabriquer, ils sont résistants et stables. Dans le vocabulaire technique, on parle de cellules. Une batterie à 6 cellules contient 6 accumulateurs, et ainsi de suite pour les batteries à 9, voire plus, cellules.

Une cellule est composée de trois éléments : l'anode (le pôle négatif), la cathode (le pôle positif) et des couches de séparation, en général en polyéthylène ou en polypropylène. Ces couches possèdent des pores qui ont la faculté de se fermer à 130°C, permettant de stopper toute réaction chimique en cas de surchauffe de la batterie. Divers éléments viennent envelopper le

cylindre, notamment des membranes qui peuvent casser pour réduire la pression en cas de choc important sur la batterie. Toujours dans le but de garantir la sécurité, les fabricants incluent, sous le pôle positif, un PTC (Positive Coefficient Temperature) qui réagit en cas d'élévation de la température en réduisant ou stoppant le flot de courant. Il est impossible de retirer le PTC sans détruire la batterie. Malgré tout, les batteries Li-Ion restent dangereuses, et peuvent gonfler ou exploser, comme l'a montré l'actualité technologique récente.

1.3. Le lithium

Le lithium trouve sa place dans les électrodes positives, négatives et dans l'électrolyte. Souvenez-vous de vos cours de chimie : le lithium est le métal le plus léger, et il possède des propriétés électrochimiques excellentes, lui permettant de disposer d'une densité énergétique deux fois supérieure à celle du Ni/Cd qui composait les batteries de votre jeunesse. Le lithium pur est très instable; les industriels ont donc choisi sa forme ionique, d'où le nom de Li-Ion, plus stable, mais aussi un peu moins efficace.

La cathode est composée de dioxyde de cobalt ou de manganèse, tandis que l'anode utilise des MCMB (Meso Carbon Micro Beards), en fait des sphères de graphite mises au point par Bell Labs. L'électrolyte, elle, est composé d'un sel conducteur.

Le tout est enveloppé dans une couche d'isolant; le pôle positif reçoit ainsi un manteau d'aluminium, tandis que le négatif est enrobé de cuivre.

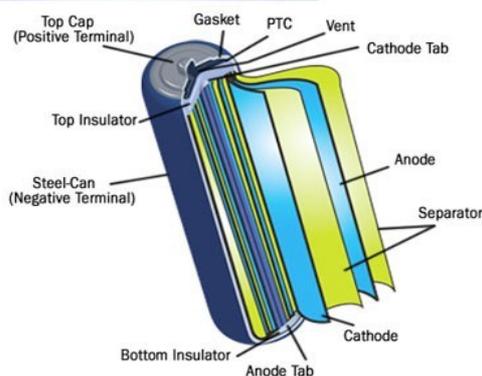
1.4. Le montage

Maintenant que vous savez tout sur les cellules, voyons comment elles sont disposées dans votre batterie.

Elles peuvent être placées en série ou en parallèle (des souvenirs ?). Dans les faits, on rencontre, dans nos batteries, un mélange des deux. Les batteries les plus courantes disposent de 8 cellules. Chaque cellule a les caractéristiques suivantes : 3,7V et 2200mAh. Quatre accus (ou cellules) ont montés en série afin d'obtenir une tension de 14,8V (4 x 3,7). On met ensuite les deux packs de quatre accus en parallèle afin d'accroître cette fois-ci la capacité, qui passe à 4400mAh.

Ceci n'est qu'un exemple d'une configuration de type 4S2P (4 en série, 2 parallèles). On trouve des batteries à 9 cellules en 3S3P (3

Cylindrical lithium-ion battery



©2006 HowStuffWorks

packs de 3 accus), ou même des batteries 12 cellules en 4S3P. Les accus sont différents d'un modèle à l'autre, avec des variations sur leur tension ou leur intensité.



1.5. Le circuit électronique

La dernière pièce de la batterie, à part habillage final, et un circuit électronique à la fonction vitale, puisqu'il est chargé du bon fonctionnement de la batterie, et est censé prévenir tout problème.

Il est composé d'un transistor à effet de champ chargé de couper l'alimentation si la tension de charge excède la valeur nominale de 0,1V. Divers fusibles sont présents, pouvant sauter si la température excède les 90°C. On trouve enfin un dispositif qui surveille la pression interne de la batterie, et qui coupe le courant si celle-ci dépasse 10 Bars.

Le circuit veille aussi à la tension de chaque cellule et à la bonne connexion entre les cellules. Si la surcharge est un problème, la sur-décharge en est aussi un : le circuit surveille donc aussi ce paramètre, et coupe tout si celle-ci est trop forte.

1.6. Le salaire de la peur

Malgré toutes ces précautions, vous l'aurez compris, il faut toujours se méfier d'une batterie.

L'électricité statique, un chargeur défectueux, des composants endommagés... sont tous des pièges qui peuvent causer la mort prématurée de votre batterie, voire un accident tel qu'un gonflement, une explosion, voire un incendie.

Les constructeurs d'ordinateurs portables sont les plus précautionneux. Non seulement leur batteries sont de très bonne qualité, avec une électronique irréprochable, mais en plus, il est quasiment impossible de les ouvrir sans les détruire. Et même ouvertes, ces batteries sont particulièrement dures à disséquer, et leurs secrets sont bien gardés. Les incidents peuvent néanmoins survenir. Ainsi, en 2005/6, chez Sony, du nickel avait été introduit dans les batteries à l'assemblage, ce qui provoquait l'instabilité de la batterie, et de possibles incidents. Heureusement, les batteries ont toutes été rappelées. Cette exemplarité n'est malheureusement pas présente dans tous les domaines, et certains fabricants, en tirant les coûts vers le bas, tirent aussi la qualité vers le bas. C'est notamment le cas chez certains fabricants de jouets, et l'on se souvient d'hélicoptères miniatures télécommandés qui étaient des copies du modèle de référence, dont la batterie, défectueuse, avait provoqué de nombreux accidents sur des enfants.

2. Charge et décharge...

2.1. De vieille habitudes...

Le Macintosh Portable utilisait une batterie au plomb, lui permettant d'afficher une autonomie record de 10h, mais un poids de 7kg... Vous avez aussi souvent entendu parler des batteries Ni/Mh ou Ni/Cd, et pris des réflexes datant de cette époque.

Le problème : avec les batteries Li-Ion, vos réflexes sont devenus mauvais, et il va falloir les combattre.

2.2. Ordinateur tout neuf

A la réception de votre ordinateur, vous constaterez que sa batterie est remplie à environ 40% de sa capacité totale. Nous verrons plus tard pourquoi.

Vous ne devez pas décharger votre batterie dès la réception, sous peine de perdre, en un cycle, 20 à 25% de sa capacité. Vous devez au contraire la charger une bonne fois, puis la décharger jusqu'à que votre système d'exploitation vous prévienne qu'il faut brancher votre ordinateur, et même jusqu'à que votre ordinateur s'éteigne.

En usine, les batteries sont en effet chargées très rapidement, et l'électronique interne a du mal à suivre. De même, à l'intérieur de la batterie, un peu de lithium s'oxyde et se dépose sur l'anode. En faisant cette opération, vous allez calibrer votre batterie, c'est-à-dire synchroniser l'état réel de votre batterie et l'électronique, ce qui permettra de vous donner une estimation plus précise de la capacité restante. Vous allez aussi consommer encore un peu de lithium, qui viendra se déposer, ce qui a pour conséquence de perdre (très) légèrement en capacité, mais de gagner en stabilité. Vous remarquerez que lors des premières charges, vous perdrez peut-être quelques pour-cent de capacité; c'est normal, vous consommez encore un peu de lithium, qui va former une véritable couche protectrice. Si par contre, vous observez une très forte perte, allez immédiatement au SAV faire changer votre batterie.

2.3. La décharge

La règle est simple : ne déchargez **jamais** complètement votre batterie.

Les accus sont reliés, nous l'avons vu, à un système électronique complexe. Celui-ci doit être constamment alimenté, et, de plus, une décharge totale entraîne la dégradation chimique des éléments contenus dans les accus. Une charge après une décharge totale peut être très dangereuse, car vous n'avez alors plus de circuit de protection, et les éléments sont instables.

Heureusement, votre système d'exploitation vous prévient quand vous approchez la zone dangereuse. Soyez prudent, n'allez pas plus loin, et trouvez-vous une prise, ou éteignez votre ordinateur (éteignez le, ne le mettez pas juste en veille, car cette veille consomme de l'énergie). Ce dispositif permet de ne pas décharger sa batterie à plus de 95%.

2.4. La charge

L'idéal est donc d'effectuer de courtes recharges.

La situation idyllique est celle du cadre habitant en banlieue, utilisant son ordinateur dans le RER, et le branchant en arrivant au bureau. Ou de l'étudiant qui utilise son ordinateur dans un amphithéâtre (un cours d'une heure), puis le branche en arrivant dans sa salle de travaux dirigés, en général équipées de prises.

Pour votre téléphone, n'hésitez pas à le brancher une fois votre journée finie, et faites de même pour votre baladeur.

2.5. Les cycles

Contrairement à une opinion répandue, un cycle n'équivaut pas à une opération de charge.

Votre batterie est chargée. Vous partez, et utilisez 50% de sa capacité. Vous la rechargez, puis re-belote. Vous avez chargé votre batterie deux fois, mais n'avez utilisé qu'un seul cycle. Deux charges de 50% font un cycle, comme quatre charges de 25%. Et comme il est conseillé de faire des charges partielles, vous aurez compris que sur ce plan là, vous pouvez y aller.

On estime la durée de vie d'une batterie à près de 1000 cycles, mais on constate plutôt une durée de vie à 300-500 cycles selon les conditions d'utilisation. Bref, de quoi voir venir (quelque chose comme 3 ans).

2.6. L'exception qui confirme la règle

A toute règle, son exception, et les batteries n'y coupent pas.

L'électronique interne de la batterie a tendance à mal percevoir l'état réel de la batterie, notamment quand nous faisons de courtes recharges, parce que le bouillon de culture des accus est un joyeux bazar réactionnel. Il faut donc, de temps à autres, calibrer la batterie.

L'intervalle moyen entre deux calibrages est de 30 cycles. Le plus simple est de vous y astreindre une fois par mois, un dimanche (comme ça, vous passerez au moins un dimanche par mois ailleurs que le nez collé sur votre écran).



Pour calibrer votre batterie, vous devez d'abord la décharger entièrement (il faut donc qu'elle soit pleine, à la base), même après le message d'avertissement, et laisser l'ordinateur s'éteindre. Votre batterie est donc complètement vide (il doit lui rester quelque chose comme 8% de capacité, juste assez pour alimenter le circuit interne). Laissez la reposer deux heures environ (nous verrons pourquoi juste en dessous). Branchez votre ordinateur, et, contrairement à ce que disent les fabricants, laissez-le éteint pendant cette charge. Une fois la charge complète, vous pouvez à nouveau vous amuser, mais veillez à ne pas le débrancher avant cinq bonnes heures. Vous comprenez maintenant pourquoi il faut calibrer le dimanche, ce n'est pas une histoire de chimie, mais de temps.

2.7. La température

Mais le temps est ici lié à la chimie. Pourquoi en effet charger sa batterie ordinateur éteint lors de la calibration ?

En fait, histoire de compliquer la chose, la température est un facteur critique. L'idéal serait que la batterie ne dépasse pas les 20°C. Malheureusement, ce ne sera jamais le cas, ce qui explique que malgré toutes vos précautions, irrémédiablement, votre batterie perdra de sa capacité.

En-dessous de 0°C, charger votre batterie entraînerait une migration du lithium contre les électrodes, et sa cristallisation. De quoi tuer une batterie en moins de deux. Et au-dessus de 45°C, vous jouez sur la pression interne de la batterie. D'ailleurs, souvent, un circuit interne va couper l'alimentation parce que votre batterie est trop chaude, vous faisant croire que la charge est terminée, alors qu'en fait, elle ne l'est pas...

Bref, sur ce terrain, malheureusement, il n'y a pas grand chose à faire, à part, si vous le pouvez, recharger votre batterie le plus souvent ordinateur éteint, ou au moins en veille. De quoi, encore une fois, vous permettre de dîner avec votre famille, et non pas devant votre écran.

2.8. L'effet mémoire ?

Une idée reçue veut qu'une charge partielle endommage les accus.

Quant la NASA a lancé ces premiers satellites, ceux-ci se chargeaient le jour, et se déchargeaient la nuit. La durée de charge/décharge était exactement la même, jour après jour. Mais au bout de quelques temps, les savants se sont aperçu que les batteries refusaient de délivrer toute l'énergie qu'elles avaient en réserve, et n'allaient en fait plus au-delà du précédent seuil de recharge. Voici ce qu'est l'effet mémoire, qu'il est impossible de reproduire dans la vie courante, puisque vos durées de charges et de décharges sont toujours différentes.

Il existe un faux effet mémoire, que les fabricants de batteries Ni/Cd ont lancé à la face de ceux de batteries Ni/Mh, et réciproquement. C'est en fait un phénomène de dépression de tension, qui apparaît lorsqu'on laisse branchée sa batterie Ni/Cd ou Ni/Mh alors qu'elle est pleine. Le courant va alors endommager la structure de l'accumulateur, qui va perdre en tension. On obtient alors des accus qui ne pourront délivrer la tension nominale requise pour le fonctionnement de l'appareil, et on pensera à l'effet mémoire. D'ailleurs, si vous utilisez des batteries Ni/Cd qui présentent ce phénomène, il suffit de les décharger jusqu'à leur seuil critique puis de les décharger pour les restaurer.

Les batteries au Li-Ion n'ont pas d'effet mémoire, y compris quand elles restent connectées au secteur (cas d'une batterie qui resterait dans un ordinateur portable). Au moment de la charge, le 'chargeur' alloue partie du courant au fonctionnement de l'ordinateur, et une partie pour la charge de la batterie. Ce qui explique que les chargeurs envoient environ 19V, alors que la batterie ne nécessite que 14V pour être chargée (pertes électriques comptées), et qu'une batterie est plus vite chargée ordinateur éteint. Une fois la batterie chargée, l'électronique interne coupe le courant envoyé aux accumulateurs, évitant ainsi toute surcharge.

Encore heureux, car la moindre surcharge causerait la perte immédiate de votre batterie, et peut-être de vos mains. La surcharge provoque en effet une production d'hydrogène, qui augmente la pression de la batterie, puis son explosion.

C'est pourquoi vous devez toujours utiliser le chargeur fourni d'origine avec votre matériel, qu'il s'agisse d'un ordinateur portable, d'un téléphone ou d'un baladeur. Et si vous devez vous procurer un chargeur de remplacement, vous pouvez utiliser un

autre chargeur, à la condition expresse que sa tension de sortie ne dépasse pas celle du chargeur d'origine.

2.9. La tension de charge

Pour finir ce chapitre sur la charge, intéressons-nous à cette fameuse tension de charge, pour vous les curieux...

La tension moyenne de charge d'un ordinateur portable est de 4,2V par cellule. Sur une batterie à 8 cellules, vous remarquerez que le chargeur indique 20V. En général, ces batteries sont constituées de deux packs de quatre cellules. La tension est donc divisée par quatre, mais, à cause des pertes électroniques, on est plus proche de 4,2V par cellule que des 5V.

Ainsi, évitez tout chargeur qui clame charger plus vite ou qui aurait un pouvoir de restauration des cellules (on ne restaure pas une cellule, mais on change une batterie entière). Ou sentez-vous libre de ne pas respecter ce conseil, et de vous faire très, très mal...



3. Usage sédentaire...

3.1. Du salon à la chambre...

De plus en plus de gens achètent un ordinateur portable pour être mobile... de la chambre au salon, et inversement. Et la plupart du temps, ces ordinateurs restent longtemps, très longtemps sur le secteur. Alors, faut-il enlever leur batterie ?

3.2. Consommer sans bouger

Même si votre batterie reste constamment sur secteur, elle perd légèrement de sa capacité, environ 10% par mois. Sur ces 10%, 8% sont imputables à une paresse des accumulateurs, tandis que les 2% restants sont dus au rôle de pseudo-onduleur que joue votre batterie (de même, si vous expérimentez une coupure de courant, votre ordinateur ne s'éteindra pas, car la batterie alimente

constamment celui-ci, et prendrait automatiquement le relais). Votre ordinateur va en fait alimenter votre batterie dès qu'elle aura perdu 5% de sa capacité. Ainsi, chaque mois, en usage sédentaire, votre ordinateur effectue deux recharges de 5%, sans vous en avertir.

3.3. Une histoire de température

Dès lors, comme les batteries Li-Ion n'ont pas peur des petites recharges, et ne craignent pas d'être branchées grâce à leur électronique interne, on peut se demander si l'idée reçue qu'il faut enlever la batterie de son ordinateur portable en usage sédentaire est bien valable.

Cela dépend en fait de l'utilisation que vous faites de votre ordinateur. Si vous vous contentez d'Internet et de la bureautique, et que votre ordinateur chauffe donc peu, vous pouvez laisser votre batterie à sa place. En plus, elle jouera son rôle de pseudo-onduleur.

Si votre ordinateur chauffe beaucoup, ou si vous l'utilisez de manière intensive, la température de votre batterie va dépasser certainement les 20°C recommandés. Dès lors, il est plus judicieux de la retirer.

3.4. Stockage prolongé

Si vous partez souvent en vadrouille avec votre ordinateur, vous laisserez certainement la batterie dans son logement.



Mais si vous n'utilisez jamais votre ordinateur en déplacement, vous allez certainement vouloir stocker votre batterie. Si c'est pour quelques jours, ou quelques semaines, vous la placerez dans un tiroir ou dans une armoire, où la température ne dépasse que rarement les 20°C, et vous veillerez à ce qu'elle ne soit chargée qu'à 40% (40% atteints en chargeant la batterie, et non pas en la déchargeant). En-dessous, compte tenu de la perte naturelle de capacité, vous pourriez endommager le circuit interne. Au-dessus, vous risqueriez une surdécharge, tout aussi néfaste. Enfin, si vous pensez stocker votre batterie plusieurs mois, l'idéal est de la stocker dans un endroit sec et froid (0°C), pour stabiliser l'électrolyte. Par exemple, vous pouvez la mettre dans un sachet étanche au frigo. Une batterie ainsi stockée ne perdra que 2% de sa capacité par an, contre 4% pour une batterie stockée à 25°C, et 20 à 25% pour une batterie stockée en plein soleil ou près d'un chauffage.

3.5. Calibration

En usage sédentaire aussi, la calibration est indispensable. Il en est de même pour les batteries que vous avez stocké. Voir ci-dessus.

4. Ce qu'il faut retenir...

Une batterie est un élément fragile et capricieux.

Vous ne devez jamais :

- ouvrir une batterie
- tenter de réparer une batterie
- utiliser un chargeur qui n'est pas celui d'origine
- utiliser des batteries non certifiées ou inadaptées

Si votre ordinateur est neuf (1^{ère} utilisation) :

- chargez complètement sa batterie
- déchargez-la jusqu'au message d'alerte
- chargez-la à nouveau totalement

Ensuite, pour toutes les autres utilisations :

- ne déchargez jamais complètement votre batterie
- n'ayez pas peur de faire des recharges partielles
- essayez de recharger ordinateur éteint ou en veille
- une fois par mois, calibrez votre batterie

En usage sédentaire prolongé :

- si votre ordinateur chauffe peu, vous pouvez y laisser la batterie
- s'il chauffe beaucoup, retirez la batterie

- le stockage se fait batterie chargée à 40%, dans un endroit frais et sec
- il faut aussi calibrer une batterie stockée

Durée de vie moyenne : 500 cycles

Pour plus d'informations : MacStyle.fr ([Lien109](#))

Retrouvez l'article d'Anthony Nelzin en ligne : [Lien110](#)

Les derniers tutoriels et articles

Interview de Patrick Geiller Créateur de JSCocoa

Interview de Patrick Geiller, Créateur de JSCocoa, langage permettant de créer des applications Cocoa avec du JavaScript.

1. Pouvez-vous vous présenter à nos lecteurs ?

Patrick, 30 ans, programmeur :) J'ai développé en C++, PHP, Javascript, et maintenant en Cocoa.

2. Comment avez-vous découvert le monde Mac et le développement sur Mac ? Et pensez-vous pouvoir vivre du développement sur Mac ?

Je suis venu au Mac par le Mini, en 2005. Je lorgnais depuis longtemps sur la plateforme car j'en entendais beaucoup de bien, même de la part des hackers de Slashdot. J'ai ensuite mis la main à la pâte avec XCode et Interface Builder qui m'ont tout de suite séduit par leur facilité : on 'dessine' les liens entre composants, on dépose les objets qu'on veut instancier, on utilise Cocoa Bindings pour lier le tout, facile et élégant ! Maintenant, j'ai un iMac de début d'année. Je ne sais pas si je pourrai vivre du développement Mac, je tente le coup pour voir.

3. Pouvez-vous nous dire rapidement ce qu'est JSCocoa ?

JSCocoa permet d'écrire des applications OSX en Javascript. Quiconque a déjà écrit un peu de Javascript se sentira tout de suite à l'aise :

```
* Navigateur (JavaScript):
document.getElementsByTagName('*') donne les nodes de
la page.
* JSCocoa :
NSWorkspace.sharedWorkspace.launchedApplications donne
la liste des applications lancées.
```

JSCocoa est un pont entre Javascript et Cocoa et donne accès à (quasi) toutes les fonctions disponibles pour le développeur qui utilise habituellement XCode.

4. Quels sont ses avantages et inconvénients ?

Des inconvénients ? il y en a toujours. ;-)

L'avantage majeur est le côté runtime : on peut charger du code à n'importe quel moment, exécuter une fonction quand on veut, tout ça quand l'application JSCocoa tourne.

On peut ainsi modifier ou déboguer son application quand on veut. On peut même modifier le code Objective-C existant et changer ainsi le comportement d'un bouton ou d'une fenêtre, toujours en runtime ! Alors que Objective-C impose de recompiler à chaque fois.

Cet avantage est également l'inconvénient majeur : c'est plus lent, et comme le code n'est pas compilé il faut s'assurer de son exactitude par le biais de tests. (Ce qui est alors plus un avantage :))

Ensuite on a tous les avantages de Javascript : syntaxe point (a.b.c.d au lieu de [[a b] c] d] en Objective-C), expressions régulières, utilisation simplifiée des tableaux (array[5] au lieu de [array objectAtIndex:5]) et des dictionnaires (hash['value'] ou

hash.value au lieu de [hash objectForKey:@"value"]).

5. Pourquoi passer du temps à apprendre ce nouveau langage ?

Précision ! :) JSCocoa n'est pas un nouveau langage, c'est une interface qui permet d'utiliser Cocoa (les bibliothèques développeurs d'Apple) en Javascript (le langage du web - Safari et JSCocoa utilisent tous deux JavaScriptCore.framework).

Si vous connaissez Cocoa, la syntaxe change de `[[NSWorkspace sharedWorkspace] activeApplication]` en `NSWorkspace.sharedWorkspace.activeApplication`.

Si vous connaissez Javascript, vous manipulez des objets Cocoa - `NSWorkspace.sharedWorkspace.activeApplication` - au lieu d'objets DOM - `document.getElementById('content').color = 'red'`

6. Pouvez-vous nous donner quelques exemples de codes et résultats en JSCocoa ?

Après avoir créé une vue Core Animation et récupéré un layer container `parentLayer`, voici comment rajouter un layer :



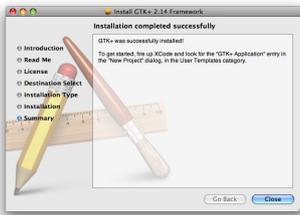
7. Que sera le projet JSCocoa dans l'avenir ? Evolution du langage ? Projets d'applications pour iPhone/iPod Touch ? évolution de l'équipe de développement ?

Les bugs et issues sont listés sur Google Code : <http://code.google.com/p/jscocoa/issues/list> ([Lien111](#))

Dans le futur proche :

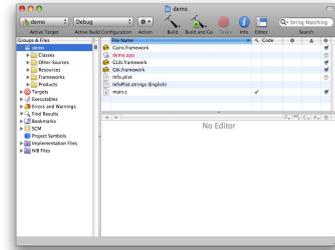
- faire tourner sur iPhone : marche déjà sur le simulateur; je cherche un volontaire pour tester sur un iPhone
- créer un type de fichier .jscocoa lancé et exécuté directement, sans passer par XCode
- intégrer JSCocoa à XCode : rajouter des templates pour créer des projets et fichiers .jscocoa

Pour l'évolution du langage et de l'équipe, toute idée ou membre est le bienvenu ! J'ai écrit JSCocoa pour répondre à mes attentes,

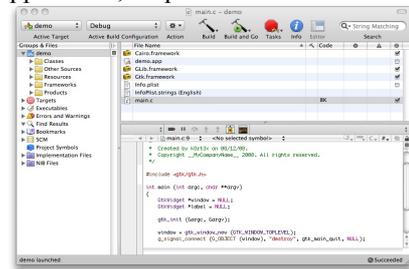


4.2. Ecriture du code

Voilà, le projet vient de s'ouvrir et nous pouvons maintenant commencer à coder sous Xcode.



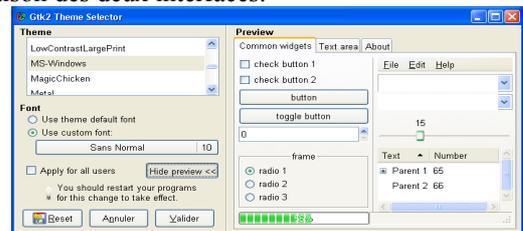
Il vous suffit de rentrer votre code dans le fichier main.c (ici un sélecteur de thème GTK comme exemple) Pour lancer l'application, cliquez sur le bouton "Build and Go".



Voilà, notre application se lance sans problème.



Voici une capture d'écran sous Windows pour faire la comparaison des deux interfaces.



Bon développement...

Retrouvez l'article d'Aurelien Gaymay en ligne : [Lien118](#)

Voilà, l'installation du Framework GTK+ est terminée.

4. Utilisation de GTK+ sous Xcode

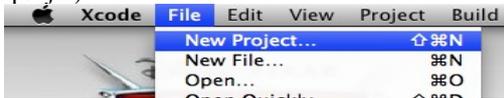
Nous allons maintenant lancer Xcode (Environnement de développement fourni par Apple). Si Xcode n'est pas encore installé sur votre machine, vous pouvez trouver toutes les informations ici ([Lien117](#)).

(Xcode se situe par défaut dans le dossier "/Developer/Applications" de votre disque)



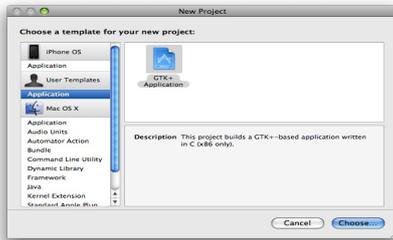
4.1. Création d'un projet sous Xcode

Une fois Xcode lancé, nous allons créer un nouveau projet. Pour cela, allez dans le menu "File" (Fichier), puis "New project" (Nouveau projet)

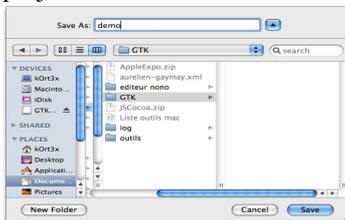


La fenêtre "New project" s'ouvre afin de sélectionner le type de projet que vous souhaitez créer.

Dans notre cas (GTK) nous allons choisir dans la partie de gauche, "User Templates" => "Application", choisissez dans le cadre centrale, "GTK+ Application" et cliquez sur le bouton "Choose..." (Choisir).



Là, on vous demande le nom et à quel endroit vous souhaitez enregistrer votre projet.



Les livres Mac

Optimisez vos tâches sous Mac OS X

Faites travailler votre Mac à votre place !

avec AppleScript, Automator, cron, curl, grep, make, shell, Tcl, etc



Cet ouvrage unique vous propose 164 recettes efficaces pour vous épargner les tâches répétitives et vous faire gagner du temps.

Organisé en cinq parties thématiques, cet ouvrage vous explique comment résoudre des problèmes classiques à l'aide d'un langage de script (AppleScript, Tcl, shell) ou d'un outil (Automator, cront, etc.) et les différents moyens d'automatiser les tâches courantes.

Qu'il s'agisse de manipulation de fichiers et de dossiers, de gestion de contenu, d'Internet, de contrôle et de maintenance de l'ordinateur et de ses applications, chaque problème trouve sa recette !

- Découvrez les ressources insoupçonnées de votre Mac, et mettez-les à profit avec quelques lignes de programmation (que de nombreuses annexes vous aideront à rédiger)
- Apprenez à faire exécuter par la machine des opérations répétitives ou des tâches fastidieuses qui requièrent un grand nombre de manipulations mettant en oeuvre différents programmes.
- Déclenchez des traitements de manières dynamique sur des éléments de données qui peuvent évoluer.
- Programmez vos actions pour qu'elles aient lieu à une date et à une heure précises.

Clair et résolument pratique, cet ouvrage ne requiert aucune connaissance préalable en programmation.

Critique du livre par Vincent Brabant

Ce livre est une mine d'or pour ce qui est de découvrir (ou mieux maîtriser) ce qu'on peut faire avec AppleScript, Automator, et un peu de commandes en ligne.

Les auteurs ont vraiment bien organisé ce livre, permettant aux personnes connaissant déjà AppleScript, Automator, les expressions régulières, ... de rentrer directement dans le vif du sujet.

Tout en ne délaissant pas les débutants, puisque les Annexes proposées sont vraiment d'une grande aide pour vite obtenir les bases, pour ceux qui ne l'ont pas, afin de pouvoir suivre ce livre.

Un conseil donc : si vous vous sentez largué à un moment donné, allez vite jeter un oeil dans les annexes. Il y a beaucoup de chance que vous y trouverez le petit truc pour raccrocher directement.

En tout cas, ce fut le cas pour moi.

Aussi, n'oubliez pas de télécharger les scripts du livre ([Lien119](#)) disponible sur le site ([Lien120](#)). Ça aide vraiment beaucoup pour la compréhension et la maîtrise puisque vous avez là le script complet, prêt à l'exécution.

Mac OS X Programmation

Cet ouvrage s'adresse aux développeurs UNIX et web qui souhaitent prendre en main et apprendre les notions fondamentales de la programmation Mac OS X. Les utilisateurs expérimentés et les développeurs Macintosh y trouveront une approche globale élargissant leur vision des possibilités de Mac OS X. Le but de cet ouvrage est de fournir une vue d'ensemble de la programmation Mac OS X. Vous y apprendrez des connaissances aussi diverses que programmer le WebKit qui anime Leopard ou l'iPhone, créer un widget pour Dashboard, implémenter le modèle MVC avec Cocoa, créer une interface Aqua pour vos commandes shell avec AppleScript. L'ambition de l'auteur est de transmettre au lecteur les fondations indispensables de la programmation Mac pour qu'il tire le meilleur parti des puissants outils de développement de ce système. Pour que ce livre soit un outil toujours à jour, il est prolongé par un site web qui actualisera en permanence les références citées : il suffira d'y saisir l'identifiant entre crochets de chaque référence du livre pour obtenir en ligne la référence actualisée.

Critique du livre par Aurelien Gaymay

Ce livre est conseillé pour les nouveaux utilisateurs Mac voulant se lancer dans la programmation Mac. Il aborde les différentes possibilités de programmation sous Mac OS X avec des exemples simples et utiles pour vous par la suite. Dans cet ouvrage, vous allez voir comment on peut créer un widget pour le DashBoard le plus simplement possible, réaliser des AppleScripts qui pourront vous être utile et permettra d'approfondir facilement l'AppleScript pour n'importe quelle application. Vous découvrirez aussi le Java, WebKit, Automator mais surtout Cocoa. Cocoa ne vous sera pas présenté dans les détails, mais un aperçu de ce que peut être le Cocoa et Objective-C.

Retrouvez ces critiques de livre sur la page livres Mac : [Lien121](#)

Les derniers tutoriels et articles

clGdiplus : Programmer un jeu. Première partie

Comment développer un jeu en VBA avec Gdi+.

Première partie : un jeu de tir.

1. Introduction

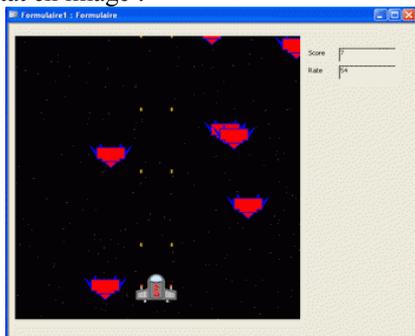
L'objectif de ce tutoriel est de développer un jeu de tir avec Access.

Pour gérer l'affichage on utilisera une classe spécifique **clGdiPlus** qui s'appuie sur la librairie Gdi+ de Microsoft.

Consultez la documentation et les tutoriels de la classe **clGdiplus** ([Lien122](#))

N'oubliez pas de télécharger la librairie **gdiplus.dll** si nécessaire.

Voici le résultat en image :



2. Spécifications fonctionnelles

Avant toute chose, nous allons écrire les spécifications fonctionnelles du jeu.

- Un écran de jeu fenêtré (pas en plein écran) pour obtenir une fluidité acceptable
- Le fond d'image est un ciel étoilé statique
- Un vaisseau est dirigé par le joueur sur les deux axes (horizontal et vertical) avec les flèches du clavier
- Le tir du vaisseau du joueur est déclenché avec la touche espace du clavier
- Le tir du vaisseau du joueur est vertical
- Les ennemis apparaissent en haut de l'image et se déplacent vers le bas à une vitesse aléatoire
- Le score est affiché à côté de l'image
- Lorsque le vaisseau du joueur entre en collision avec un ennemi, le jeu s'arrête

3. Spécifications techniques simplifiées

Ci-dessous et en bref, les techniques de base utilisées pour le développement.

Tout sera plus détaillé dans les chapitres suivants.

- Utilisation du timer du formulaire : l'image est mise à jour à chaque événement de ce timer.
- Utilisation de Gdi+ pour le graphisme, avec l'aide de la classe **clGdiplus**.

- Création d'un module de classe pour chaque famille d'objet (vaisseau, tir, ennemis, ...).
- Utilisation d'une collection pour les objets multiples (tir, ennemis).
- Le vaisseau du joueur et les ennemis sont des images stockées à l'extérieur de la base de données.
- Les tirs du vaisseau seront dessinés à base de rectangle.

4. Création du formulaire

On crée un formulaire dans lequel on place :

- Un contrôle image nommé **Img** pour l'affichage du jeu
- Une zone de texte **TxtScore** pour l'affichage du score

La zone de texte est utilisée en affichage uniquement, donc on définit ses propriétés :

- **Activé** => **Non**
- **Vérouillé** => **Oui**

Le formulaire n'est pas utilisé pour afficher des données, donc on peut supprimer les éléments inutiles dans ses propriétés :

- **Afficher sélecteur** => **Non**
- **Boutons de déplacement** => **Non**
- **Diviseur d'enregistrement** => **Non**

5. Création du module de classe clGdiplus

Téléchargez la classe au format texte ([Lien123](#))

Dans l'éditeur VBA :

- 1 - Créez un nouveau module de classe (dans le menu : **insertion** - > **module de classe**).
- 2 - Collez-y le contenu du fichier texte.
- 3 - Sauvegardez le module avec le nom **clGdiPlus**.
- 4 - Compiler (**Débogage** --> **Compiler**)

Attention : le nom sous lequel vous sauvegardez ce module est important.

6. Déclaration et initialisation de la classe clGdiplus

On va écrire notre code dans le module du formulaire. Cliquez sur **Affichage** --> **Code** pour ouvrir ce module.

Vérifiez que vous avez l'instruction **Option Explicit** en haut du module.

Si non rajoutez cette ligne, elle impose la déclaration de toutes les variables et évite ainsi les erreurs de saisie dans leur nom.

En-tête de module

```
Option Compare Database  
Option Explicit
```

Pour pouvoir utiliser la classe il est nécessaire de la **déclarer** et **initialiser**.

La classe est un objet dont le type est le nom sous lequel on a sauvegardé notre module de classe.

Elle se déclare comme n'importe quelle autre variable.

Ensuite on initialise la classe dans l'événement **Sur Ouverture** du formulaire.

Déclaration de la classe

```
Private clGdip As ClGdiPlus ' Classe pour utilisation de gdipplus
```

Initialisation de la classe

```
Private Sub Form_Open(Cancel As Integer)
' Création de la classe à l'ouverture du formulaire
Set clGdip = New ClGdiPlus
End Sub
```

Pensez à libérer la classe à la fermeture du formulaire.

(même si celle-ci est libérée automatiquement, je préfère la libérer explicitement).

Dans les propriétés du formulaire, définissez [**Procédure événementielle**] dans l'événement **Sur Fermeture**.

Cliquez sur les trois petits points [...] pour générer l'événement dans le code.

A l'intérieur de la procédure **Form_Close** on va libérer la classe : il suffit de lui donner la valeur **Nothing** si elle n'a pas déjà cette valeur.

Libération de la classe

```
Private Sub Form_Close()
' Libération de la classe à la fermeture du formulaire
If Not clGdip is Nothing Then Set clGdip = Nothing
End Sub
```

7. Dessin du fond étoilé

7.1. Création de l'image Gdi+

Pour pouvoir dessiner il faut d'abord créer un bitmap (=une image) avec la fonction **CreateBitmap**.

La taille de ce bitmap doit être exprimée en pixel, donc nous utilisons les fonctions **PointsToPixelsX** et **PointsToPixelsY** pour convertir la taille du contrôle **Img** en pixel.

Le bitmap obtenu est alors de la même taille que le contrôle **Img**.

Notez que nous conservons la taille de l'image dans des variables privés du module afin de les réutiliser plus tard.

Autre remarque : l'image est noire et transparente par défaut; nous remplissons donc l'image de noir opaque pour éviter des soucis de superposition d'image plus tard.

Déclaration en en-tête de module

```
Private gWidth As Long, gHeight As Long ' Taille de l'image en pixel
```

Création du bitmap dans Form_Load

```
' Taille de l'image = taille du contrôle
gWidth = clGdip.PointsToPixelsX(Me.Img.Width)
gHeight = clGdip.PointsToPixelsY(Me.Img.Height)
' Création d'un nouveau bitmap
clGdip.CreateBitmap gWidth, gHeight
' Rempli l'image de noir opaque
clGdip.FillColor vbBlack
```

7.2. Dessin des étoiles

Les étoiles sont des points dessinés à l'aide de la fonction **DrawPixel**.

La couleur et la position de ces points est aléatoire.

Déclaration à ajouter dans Form_Load

```
Dim lCpt As Long ' Compteur
Dim lCalc As Long ' Variable de calcul
```

Dessin des étoiles après création du bitmap

```
' Dessin des étoiles
Randomize Timer
For lCpt = 1 To 300
lCalc = 55 + Rnd * 200
clGdip.DrawPixel Rnd * gWidth, Rnd * gHeight,
RGB(lCalc, lCalc, lCalc)
Next
```

7.3. Affichage à l'écran

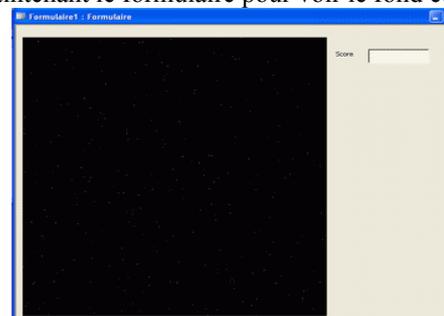
Pour l'instant, si on exécute le code, rien ne s'affiche. On a uniquement dessiné en mémoire.

Utilisons la fonction **RepaintControl** pour injecter l'image dans le contrôle **Img**.

Affichage à l'écran, à la suite du code précédent

```
' Dessin dans le contrôle
clGdip.RepaintControl Me.Img
```

Afficher maintenant le formulaire pour voir le fond étoilé.



8. La minuterie

Pour faire un jeu il faut rafraîchir la position des objets et l'affichage à intervalle régulier.

Pour cette opération nous utilisons le timer du formulaire.

8.1. Activation de l'événement "Sur Minuterie"

Celui-ci est accessible dans les propriétés du formulaire dans l'onglet événements:

Définissez [**Procédure événementielle**] dans l'événement **Sur minuterie**.

Puis cliquez sur les trois petits points [...] pour générer l'événement dans le code.

Définissez ensuite l'intervalle de minuterie.

8.2. Choix de l'intervalle de minuterie, la théorie.

L'intervalle de minuterie ne doit pas être choisi au hasard.

Il définit le nombre d'image qui sera dessiné par seconde.

Par exemple, en théorie, pour un intervalle de 20 millisecondes : $1000/20 = 50$ donc on obtient 50 images par secondes.

Cela peut paraître beaucoup, mais si vous souhaitez déplacer pixel par pixel un objet sur toute la hauteur d'un écran de 800 pixels, il faudra compter :

$800/50 = 16$ secondes.

8.3. Choix de l'intervalle de minuterie, la pratique.

Pour tester notre minuterie, nous allons afficher le nombre d'exécutions par seconde sur le formulaire.

Créer une zone de texte nommée **TxtRate**.

Dans l'événement **Form_Timer** on va compter le nombre d'exécutions par seconde :

Affichage du framerate

```
Private Sub Form_Timer()  
' Affiche le framerate  
Static sTimer As Double  
Static sNb As Long  
Dim lTimer As Double  
Dim lInterval As Long  
lTimer = Timer  
If lTimer - sTimer >= 1 Then  
    Me.TxtRate.Value = CLng((sNb + 1) / (lTimer -  
sTimer))  
    sNb = 0  
    sTimer = lTimer  
Else  
    sNb = sNb + 1  
End If  
End Sub
```

Définissez un intervalle de minuterie de 25 ms et exécutez le formulaire.

On s'attend à voir un framerate de $1000/25 = 40$.

Pourtant l'affichage nous indique environ 33 ce qui correspond à un intervalle de 30.

En fait la minuterie n'a pas une résolution de 1 ms.

C'est à dire qu'on ne peut pas définir notre intervalle à la milliseconde près.

On ne peut définir l'intervalle que par pas de 10, 15 ou 55 suivant l'environnement :

- 55 Pour Windows 95 et 98.
- 15 Pour Windows supérieur à 98 et Dual Core.
- 10 Pour Windows supérieur à 98 sans Dual Core.

Choisir un intervalle de 15 ms permet d'avoir 50 images/s sans Dual Core et 66 images/s avec un Dual Core.

C'est cet intervalle de minuterie que nous choisissons pour ce tutoriel.

9. Création, affichage et déplacement du vaisseau

Comme son titre l'indique, nous allons dans cette section gérer le vaisseau du joueur.

9.1. Création d'un module de classe

Notre vaisseau est un objet avec des propriétés telles que sa position, sa vitesse, ...

Commençons par créer un module de classe nommée **clShip**.

Inscrivons-y les propriétés évidentes :

- la **position** : X et Y
- la **vitesse** : Speed

Classe clShip

```
Option Compare Database  
Option Explicit  
Public X As Long, Y As Long ' Position  
Public Speed As Long ' Vitesse
```

9.2. Déclaration et initialisation de l'objet dans le formulaire

Comme il n'y a qu'un vaisseau, on lui crée une variable objet **gShip** qu'on initialise à l'ouverture du formulaire et qu'on libère à la fermeture du formulaire.

À l'initialisation, on définit la position du vaisseau et sa vitesse en pixel.

Déclaration en en-tête de formulaire

```
Private gShip As ClShip ' Objet Vaisseau
```

Initialisation de l'objet vaisseau Form_Load

```
' Initialisation de l'objet Vaisseau  
Set gShip = New ClShip  
gShip.X = gWidth / 2 ' Centré horizontalement  
gShip.Y = 0.9 * gHeight ' Verticalement à 90% du haut  
de l'image  
gShip.Speed = 5
```

Libération de l'objet vaisseau dans Form_Close

```
' Libération de l'objet vaisseau  
Set gShip = Nothing
```

9.3. Chargement et affichage de l'image du vaisseau

Préparez une image avec votre éditeur de dessin favoris.

Pour ce tutoriel, j'ai simplement utilisé **Paint** et **Microsoft photo editor**.



Évitez le format Jpeg qui compresse l'image, sinon vous ne pourrez plus travailler l'image pour rendre le tour transparent (il faut qu'il soit uni).

À la charge du formulaire, ajoutez l'image du vaisseau dans la classe **clGdiplus**.

Vous pouvez également redimensionner cette image.

Chargement et redimensionnement de l'image dans Form_Load

```
' Chargement des images  
clGdiplus.ImageListAdd "ship", CurrentProject.Path &  
"\img\ship.bmp" ' Chargement du fichier  
clGdiplus.ImageListResize "ship", 80 ' Redimensionne à 80  
pixels de largeur
```

L'affichage est géré dans la procédure **Form_Timer**.

Nous utilisons la fonction **DrawImage** pour dessiner le vaisseau sur l'image principale.

Le paramètre **GdiplusSizeModeAutoSize** indique que l'image est dessinée à sa taille réelle et centrée sur les coordonnées **gShip.X** et **gShip.Y**.

Puis la fonction **FastRepaint** redessine l'image du jeu sur le formulaire.

Noter que nous n'injectons pas l'image dans le contrôle avec la fonction **RepaintControl** car ce serait trop lent.

Dessin du vaisseau

```
' Dessine le vaisseau  
clGdiplus.DrawImage "Ship", gShip.X, gShip.Y, , , ,  
GdiplusSizeModeAutoSize  
' Dessin l'image sur le formulaire  
clGdiplus.FastRepaint Me.Img
```

9.4. Gestion de la transparence

Si l'image est dans un format qui ne supporte pas la transparence, le vaisseau est dessiné dans un rectangle plein.

Il faut rendre ce fond transparent :

1 - soit en remplaçant la couleur du fond par une couleur transparente

```
' Chargement des images  
clGdiplus.ImageListAdd "ship", CurrentProject.Path &  
"\img\ship.bmp" ' Chargement du fichier  
clGdiplus.ImageListResize "ship", 80 ' Redimensionne à 80
```

```

pixels de largeur
clGdip.ImageListReplaceColor "ship", vbWhite,
vbBlack, , 0 ' Remplace le blanc par du noir
transparent

```

2 - soit en définissant la couleur de transparence à l'affichage.
(remarque : l'image étant transformée lors du redimensionnement, cette méthode ne donne pas les meilleurs résultats).

```

' Dessine le vaisseau
clGdip.DrawImage "ship", gShip.X, gShip.Y, , , vbWhite,
GdipSizeModeAutoSize

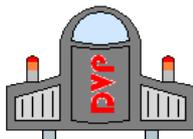
```

3 - soit en sauvegardant l'image dans un format qui supporte la transparence.

Le format PNG donne de bons résultats.

Il n'y a pas alors besoin de gérer la transparence dans le code.

Voici l'image du vaisseau au format PNG avec transparence.



9.5. Gestion des déplacements

Le déplacement du vaisseau se fait avec les touches fléchées.

Lorsqu'on appuie ou relâche une touche, un événement **KeyDown** puis **KeyUp** est déclenché.

Si on laisse une touche appuyée, des événements **KeyDown** successifs sont déclenchés.

Modifier la position du vaisseau directement sur cet événement rendrait un mouvement très saccadé et fonction de la configuration de la répétition des touches.

Pour avoir un mouvement fluide, on va définir 4 variables booléennes (une par touche) dont on modifiera l'état (Vrai ou Faux) lors des événements **KeyDown** et **KeyUp**.

Déclaration en en-tête de formulaire

```

Private gKeyLeft As Boolean ' Etat flèche gauche
Private gKeyRight As Boolean ' Etat flèche droite
Private gKeyUp As Boolean ' Etat flèche haut
Private gKeyDown As Boolean ' Etat flèche bas

```

Sur touche appuyée

```

Private Sub Form_KeyDown(KeyCode As Integer, Shift As Integer)
Select Case KeyCode
Case 37 ' Gauche
gKeyLeft = True
Case 38 ' Haut
gKeyUp = True
Case 39 ' Droite
gKeyRight = True
Case 40 ' Bas
gKeyDown = True
Case Else
End Select
End Sub

```

Sur touche relâchée

```

Private Sub Form_KeyUp(KeyCode As Integer, Shift As Integer)
Select Case KeyCode
Case 37 ' Gauche
gKeyLeft = False
Case 38 ' Haut
gKeyUp = False
Case 39 ' Droite

```

```

gKeyRight = False
Case 40 ' Bas
gKeyDown = False
Case Else
End Select
End Sub

```

Dans l'événement "Sur minuterie", on vérifie alors l'état des touches grâce à ces variables et on modifie la position du vaisseau en fonction de sa vitesse.

Dans Form Timer avant l'affichage du vaisseau

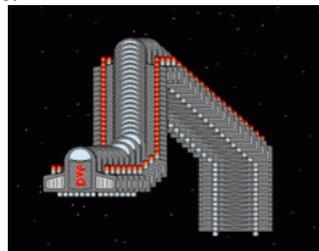
```

' Changement de la position du vaisseau
If gKeyLeft Then gShip.X = gShip.X - gShip.Speed
If gKeyRight Then gShip.X = gShip.X + gShip.Speed
If gKeyUp Then gShip.Y = gShip.Y - gShip.Speed
If gKeyDown Then gShip.Y = gShip.Y + gShip.Speed

```

9.6. Réinitialisation de l'image à chaque itération

Si vous exécutez le formulaire et déplacez le vaisseau, vous vous apercevez que le vaisseau est dessiné à chaque fois par dessus l'image précédente.



Il faut, avant chaque dessin de l'image dans **Form_Timer**, réinitialiser l'image de fond avant de dessiner les objets dessus.

Dans Form Load après le dessin des étoiles

```

' Conserve l'image de fond
clGdip.KeepImage

```

Dans Form Timer au début

```

' Rétabli l'image de fond
clGdip.ResetImage

```

Avoir rempli l'image de fond de noir opaque à sa création est utile ici sinon le fond serait transparent et ça ne marcherait pas.

Vous pouvez maintenant exécuter le formulaire et utilisez les flèche pour déplacer le vaisseau.

10. Création, affichage et déplacement des ennemis

Comme son titre l'indique, nous allons dans cette section gérer les vaisseaux ennemis.

10.1. Création d'un module de classe

Les ennemis sont, comme le vaisseau du joueur, des objets avec des propriétés telles que leur position, leur vitesse, ...

Commençons par créer un module de classe nommée **CIEnnemi**.

Inscrivons-y les propriétés évidentes :

- la **position** : X et Y

- la **vitesse** : Speed

Classe cIEnnemi

```

Option Compare Database
Option Explicit
Public X As Long, Y As Long ' Position
Public Speed As Long ' Vitesse

```

10.2. Déclaration et initialisation de l'objet dans le formulaire

Comme il peut y avoir plusieurs ennemis, on crée une collection qu'on initialise à l'ouverture du formulaire et qu'on libère à la fermeture du formulaire.

Les ennemis seront créés puis insérés dans cette collection.

Déclaration en en-tête de formulaire

```
Private gEnnemis As Collection ' Collection d'ennemis
```

Initialisation de la collection d'ennemis dans Form_Load

```
' Initialisation de la collection d'ennemis  
Set gEnnemis = New Collection
```

Libération de la collection d'ennemis dans Form_Close

```
' Libération de la collection d'ennemis  
Set gEnnemis = Nothing
```

10.3. Création des ennemis

Les ennemis sont créés en cours de jeu dans la procédure **Form_Timer** puis ajoutés à la collection.

Nous avons donc besoin d'une variable **lEnnemi** de type **ClEnnemi**.

Un ennemi apparaît par exemple toutes les demi-secondes. Il nous faut donc une variable pour conserver le timer lors de la dernière création d'ennemi.

Déclaration en début de procédure Form_Timer

```
Static sLastEnnemi As Double ' Timer lors de la  
création du dernier ennemis  
Dim lEnnemi As ClEnnemi ' Objet ennemi
```

Si on n'a pas créé d'ennemis depuis plus d'une demi-seconde, on crée un objet ennemi avec sa position et sa vitesse.

L'ennemi apparaît en haut de l'écran, avec une position horizontale et une vitesse aléatoire.

Création d'un ennemi en début de procédure Form_Timer

```
' Nouvel ennemi toute les demi-secondes  
If Timer - sLastEnnemi > 0.5 Then  
    sLastEnnemi = Timer ' Conserve le timer de la  
dernière création  
    Set lEnnemi = New ClEnnemi ' Nouvel objet  
    lEnnemi.X = Rnd * (gWidth) + 1 ' Position  
horizontale aléatoire (entre 1 et gwidth)  
    lEnnemi.Y = 0 ' Position vertical = en haut  
    lEnnemi.Speed = Rnd * 9 + 1 ' Vitesse aléatoire  
(entre 1 et 9 pixel)  
    gEnnemis.Add lEnnemi ' Ajoute l'ennemi à la  
collection  
    Set lEnnemi = Nothing ' Libère l'objet  
End If
```

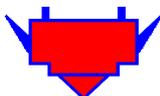
Remarque : l'objet est libéré après son ajout à la collection. Cela ne signifie pas que l'objet est détruit, car il est toujours contenu dans la collection.

C'est là tout l'avantage d'avoir créé une collection en en-tête de module.

10.4. Chargement de l'image des ennemis

Préparez un fichier image de la même manière que pour le vaisseau puis chargez cette image à l'ouverture du formulaire.

Image utilisée dans ce tutoriel :



Vous pouvez également redimensionner cette image.

Chargement et redimensionnement de l'image dans Form_Load

```
' Chargement des images  
clGdip.ImageListAdd "ennemi", CurrentProject.Path &  
"\img\ennemi.png" ' Chargement du fichier  
clGdip.ImageListResize "ennemi", 80 ' Redimensionne à  
80 pixels de largeur
```

10.5. Affichage et déplacement des ennemis

L'affichage et le déplacement des ennemis sont gérés dans la procédure **Form_Timer**.

Comme il y a plusieurs ennemis, il faut faire une boucle sur la collection d'ennemis.

On souhaite également retirer l'ennemi de la collection s'il dépasse le bas de l'écran.

C'est pour cela que le compteur **lCpt** est à rebours; on peut supprimer un élément et continuer la boucle vers les éléments d'indice moins élevé.

Déclaration en début de procédure Form_Timer

```
Dim lCpt As Long ' Compteur
```

Dessin et déplacement des ennemis dans Form_Timer

```
' Ennemis  
For lCpt = gEnnemis.Count To 1 Step -1  
    Set lEnnemi = gEnnemis.item(lCpt)  
    ' Dessine l'ennemi  
    clGdip.DrawImage "ennemi", lEnnemi.X,  
lEnnemi.Y, , , GdipSizeModeAutoSize  
    ' Déplace l'ennemi  
    lEnnemi.Y = lEnnemi.Y + lEnnemi.Speed  
    ' Si le missile arrive en bas de l'écran, on le  
supprime  
    If lEnnemi.Y > gHeight Then  
        ' Supprime l'objet  
        gEnnemis.Remove lCpt  
    End If  
Next
```

11. Création, affichage et déplacement des missiles

Nous allons dans cette section gérer les missiles tirés par le vaisseau du joueur.

11.1. Création d'un module de classe

Les missiles sont également des objets.

Commençons par créer un module de classe nommée **CIMissile**.

Inscrivons-y les propriétés évidentes :

- la **position** : X et Y

- la **vitesse** : Speed

- la **taille** : Size

Classe CIMissile

```
Option Compare Database  
Option Explicit  
Public X As Long, Y As Long  
Public Speed As Long  
Public Size As Long
```

11.2. Déclaration et initialisation de l'objet dans le formulaire

Comme il peut y avoir plusieurs missiles, on crée une collection qu'on initialise à l'ouverture du formulaire et qu'on libère à la fermeture du formulaire.

Les missiles seront créés puis insérés dans cette collection.

Déclaration en en-tête de formulaire

```
Private gMissiles As Collection ' Collection de  
missiles
```

Initialisation de la collection de missiles dans Form_Load

```
' Initialisation de la collection de missiles
Set gMissiles = New Collection
```

Libération de la collection de missiles dans Form_Close

```
' Libération de la collection de missiles
Set gMissiles = Nothing
```

11.3. Création des missiles

Les missiles apparaissent lorsqu'on appuie sur la barre d'espace. On procède de même que les touches fléchées, avec une variable qui contient l'état de la touche espace.

Déclaration en en-tête de formulaire

```
Private gKeySpace As Boolean ' Etat de la touche espace
```

Sur touche appuyée

```
Case 32 ' Espace
    gKeySpace = True
```

Sur touche relâchée

```
Case 32 ' Espace
    gKeySpace = False
```

Les missiles sont créés en cours de jeu dans la procédure **Form_Timer** puis ajoutés à la collection.

Nous avons donc besoin d'une variable **IMissile** de type **ClMissile**.

Pour éviter de créer des missiles trop fréquemment (un à chaque minuterie), on utilise une variable static pour conserver le timer lors de la dernière création de missile.

Déclaration en début de procédure Form_Timer

```
Static sLastMissile As Double ' Timer lors de la
création du dernier missile
Dim lMissile As ClMissile ' Objet missile
```

Si la touche espace est appuyée et qu'on n'a pas créé de missile depuis plus de 0.2 seconde, on crée un nouvel objet missile.

Le missile doit apparaître à un emplacement bien défini, en fonction de la position actuelle du vaisseau.

En fait on crée deux missiles car notre vaisseau a deux canons.

Création de missiles en début de procédure Form_Timer

```
' Nouveaux missiles
If gKeySpace Then ' Si espace appuyé
    If Timer - sLastMissile > 0.2 Then ' Si pas de
missile depuis plus de 0.2 secondes
        ' Premier missile
        Set lMissile = New ClMissile
        lMissile.X = gShip.X - 27
        lMissile.Y = gShip.Y - 10
        lMissile.Speed = 10
        lMissile.Size = 5
        gMissiles.Add lMissile
        Set lMissile = Nothing
        ' Deuxième missile
        Set lMissile = New ClMissile
        lMissile.X = gShip.X + 23
        lMissile.Y = gShip.Y - 10
        lMissile.Speed = 10
        lMissile.Size = 5
        gMissiles.Add lMissile
        Set lMissile = Nothing
        ' Conserve le timer de la dernière création
de missile
        sLastMissile = Timer
    End If
End If
```

11.4. Affichage et déplacement des missiles

L'affichage et le déplacement des missiles sont gérés dans la procédure **Form_Timer**.

Comme il y a plusieurs missiles, il faut faire une boucle sur la collection de missiles.

On souhaite également retirer le missile de la collection s'il dépasse le haut de l'écran.

C'est pour cela que le compteur **lCpt** est à rebours; on peut supprimer un élément et continuer la boucle vers les éléments d'indice moins élevé.

Pour le dessin du missile, nous n'utilisons pas d'image. Un simple rectangle jaune avec un fond rouge suffit.

Dessin et déplacement des missiles dans Form_Timer après le dessin du vaisseau

```
' Missile
For lCpt = gMissiles.Count To 1 Step -1
    Set lMissile = gMissiles.item(lCpt)
    ' Dessine le missile
    clGdip.DrawRectangle lMissile.X - 1, lMissile.Y -
lMissile.Size, lMissile.X + 1, lMissile.Y, vbRed,
vbYellow, 1
    ' Déplace le missile
    lMissile.Y = lMissile.Y - lMissile.Speed
    ' Si le missile arrive en haut de l'écran, on le
supprime
    If lMissile.Y <= 0 Then
        ' Supprime l'objet
        gMissiles.Remove lCpt
    End If
Next
```

Le projet est déjà bien avancé, on peut déplacer le vaisseau du joueur, des ennemis sont affichés, et on peut tirer des missiles.

Il reste un gros travail à faire : gérer les collisions entre les objets.

12. Gestion des collisions

Nous allons gérer les collisions entre :

- le vaisseau et les ennemis => si collision alors fin du jeu
- les missiles et les ennemis => si collision alors destruction de l'ennemi et du missile

12.1. Création d'une région pour le vaisseau du joueur

On pourrait utiliser les paramètres **pRegion*** de la fonction **DrawImage** pour créer automatiquement une région constituée des points non transparents de l'image.

Ce serait la solution la plus simple à développer, mais également la plus lente à l'exécution.

Nous allons donc, pour accélérer la vitesse d'exécution du programme, créer des régions polygones.

Dans la classe **clShip**, nous ajoutons une variable **Polygon** de type **Variant**.

Comme il n'est pas possible de définir un tableau directement dans la déclaration de la variable, nous définissons ce tableau à la création de la classe dans l'événement **Class_Initialize**.

Ajout à la classe clShip

```
Public Polygon As Variant

Private Sub Class_Initialize()
    Polygon = Array(9, 63, 23, 60, 23, 41, 31, 42, 31, 60,
46, 55, 46, 29, 59, 10, 74, 6, 93, 14, 102, 30, 102,
56, _
```

```
117, 61, 117, 41, 125, 42, 125, 60,
140, 64, 139, 96, 9, 96, 9, 63)
End Sub
```

Ce tableau est constitué de points en pixels (X1,Y1,X2,Y2, ...) pris sur le fichier image **ship.png**.

Ces points forment un polygone fermé qui détermine les limites du vaisseau.

Après avoir dessiné le vaisseau, nous créons une région à partir de ce polygone.

Pour tester visuellement notre région, la fonction **FillRegion** remplit la région de rouge en semi-transparence.

Dans Form Timer

```
' Dessine le vaisseau
clGdip.DrawImage "ship", gShip.X, gShip.Y, , , ,
GdipSizeModeAutoSize
' Crée une région polygonale
clGdip.CreateRegionPolygon "ship", gShip.Polygon
' Pour test, rempli la région
clGdip.FillRegion "ship", vbRed, , , 150
```

Affichez le formulaire : vous visualisez en rouge la région créée.

Cette région ne suit pas le déplacement du vaisseau et est plus grande que l'image du vaisseau.

Le problème de taille est dû au fait qu'on a redimensionné l'image après l'avoir chargée.

Les coordonnées prise sur l'image avant redimensionnement doivent être corrigée.

Le facteur de correction est, pour l'image de ce tutoriel : 80 / 156

- 80 taille avant redimensionnement

- 156 taille après redimensionnement

Correction de taille

```
' Crée une région polygonale
clGdip.CreateRegionPolygon "ship", gShip.Polygon
' Redimensionne la région
clGdip.ScaleRegion "ship", 80 / 156, 80 / 156
' Pour test, rempli la région
clGdip.FillRegion "ship", vbRed, , , 150
```

Pour repositionner la région, il faut la déplacer en fonction de la position et de la taille de l'image du vaisseau.

Correction de position

```
' Crée une région polygonale
clGdip.CreateRegionPolygon "ship", gShip.Polygon
' Redimensionne la région
clGdip.ScaleRegion "ship", 80 / 156, 80 / 156
' Déplace la région en fonction de la position du vaisseau
clGdip.TranslateRegion "ship", gShip.X -
clGdip.ImageListWidth("ship") / 2, gShip.Y -
clGdip.ImageListHeight("ship") / 2
' Pour test, rempli la région
clGdip.FillRegion "ship", vbRed, , , 150
```

Affichez le formulaire : vous voyez maintenant la région en rouge se déplacer avec le vaisseau.

Vous pouvez retirer maintenant la ligne de code de remplissage de la région.

12.2. Création d'une région pour chaque ennemi

Comme pour le vaisseau du joueur, on utilise des régions polygonales.

Pour les ennemis cependant, chacun doit pouvoir être identifié de

manière unique.

Chaque instance d'une classe possédant un pointeur unique, nous allons l'utiliser.

Dans la classe **CIennemi**, nous ajoutons une variable **Polygon** de type **Variant** et sa définition dans l'événement **Class Initialize**.

Ajout à la classe CIennemi

```
Public Polygon As Variant
Public Id as long

Private Sub Class_Initialize()
' Identifiant de l'objet
Id = ObjPtr(Me)
' Polygon pour région
Polygon = Array(4, 12, 23, 30, 24, 16, 34, 16, 34, 6,
42, 6, 42, 16, 98, 16, 98, 6, 105, 6, 105, 16, 116, 16,
-
116, 30, 136, 12, 117, 47, 117, 58, 101, 58,
100, 68, 91, 69, 70, 89, 48, 69, 40, 69, 39, 59, 23,
58, 24, 48, 4, 12)
End Sub
```

Ce tableau est constitué de points en pixels (X1,Y1,X2,Y2, ...) pris sur le fichier image **ennemi.png**.

Ces points forment un polygone fermé qui détermine les limites d'un ennemi.

Après avoir dessiné chaque ennemi, nous créons une région à partir de ce polygone.

Les corrections de position et de taille adoptent le même principe que pour le vaisseau.

Le nom utilisé pour la région est l'Id de l'objet; ainsi nous créons une région distincte par objet.

Pour tester visuellement notre région, la fonction **FillRegion** remplit la région de rouge en semi-transparence.

Dans Form Timer

```
' Dessine l'ennemi
clGdip.DrawImage "ennemi", lEnnemi.X,
lEnnemi.Y, , , , GdipSizeModeAutoSize
' Crée une région polygonale
clGdip.CreateRegionPolygon lEnnemi.Id,
lEnnemi.Polygon
' Redimensionne la région
clGdip.ScaleRegion lEnnemi.Id, 80 / 156, 80 / 156
' Déplace la région en fonction de la position du vaisseau
clGdip.TranslateRegion lEnnemi.Id, lEnnemi.X -
clGdip.ImageListWidth("ennemi") / 2, lEnnemi.Y -
clGdip.ImageListHeight("ennemi") / 2
' Pour test, rempli la région
clGdip.FillRegion lEnnemi.Id, vbRed, , , 150
```

Affichez le formulaire : vous voyez maintenant des régions en rouge se déplacer avec chaque ennemi.

Vous pouvez retirer maintenant la ligne de code de remplissage de la région.

Mais on n'en a pas encore terminé avec les régions des ennemis. Lorsqu'un ennemi disparaît en bas de l'écran, on supprime l'objet de la collection mais sa région n'est pas supprimée automatiquement.

Avant de retirer l'objet de la collection (et donc de le détruire) on va supprimer sa région.

Dans Form Timer

```
' Supprime la région
clGdip.DeleteRegion lEnnemi.Id
```

```
' Supprime l'objet  
gEnnemis.Remove lCpt
```

12.3. Création d'une région pour chaque missile

Pour les missiles les régions sont rectangulaires.

Il suffit d'ajouter le nom de la région à créer en paramètre de la fonction **DrawRectangle**.

On ajoute, comme pour les ennemis, un identifiant dans la classe **clMissile** utilisé pour distinguer les régions.

Et n'oublions pas de supprimer la région d'un missile lorsqu'on le détruit.

Ajout à la classe clMissile

```
Public Id As Long
```

```
Private Sub Class_Initialize()  
Id = ObjPtr(Me)  
End Sub
```

Modification dans Form Timer pour création de la région

```
' Dessine le missile et crée sa région  
clGdip.DrawRectangle lMissile.X - 1, lMissile.Y -  
lMissile.Size, lMissile.X + 1, lMissile.Y, vbRed,  
vbYellow, 1, , , lMissile.Id
```

Modification dans Form Timer pour suppression de la région

```
If lMissile.Y <= 0 Then  
' Supprime la région  
clGdip.DeleteRegion lMissile.Id  
' Supprime l'objet  
gMissiles.Remove lCpt  
End If
```

Toutes les régions sont créées. On peut maintenant tester les collisions.

12.4. Test de collision entre le vaisseau et les ennemis

Pour tester l'intersection entre deux régions, nous utilisons la fonction **RegionsIntersect**

Après chaque création de région d'un ennemi, on teste si la région de cet ennemi et la région du vaisseau ont une intersection.

Si le vaisseau entre en collision avec un ennemi, on positionne un flag à Vrai, puis en fin de procédure on teste ce flag pour afficher le texte de fin de jeu.

Ajout de déclaration dans Form Timer

```
Dim lGameOver As Boolean ' Flag pour fin de jeu
```

Test collision vaisseau avec ennemi

```
[...]  
' Déplace la région en fonction de la position du  
vaisseau  
clGdip.TranslateRegion lEnnemi.Id, lEnnemi.X -  
clGdip.ImageListWidth("ennemi") / 2, lEnnemi.Y -  
clGdip.ImageListHeight("ennemi") / 2  
' Test collision avec vaisseau  
If clGdip.RegionsIntersect("ship", lEnnemi.Id) Then  
lGameOver = True  
End If
```

Fin de jeu : en fin de procédure Form Timer

```
' Test game over  
If lGameOver Then  
' Stoppe la minuterie  
Me.TimerInterval = 0
```

```
' Affiche un texte du fin de jeu  
' On affiche d'abord le texte en noir et décalé  
pour un effet d'ombrage du texte  
clGdip.DrawText "GAME OVER", 65, "Arial", 0 + 3, 0  
+ 3, gWidth + 3, gHeight + 3, , , vbWhite, 150, , , , ,  
, , True  
clGdip.DrawText "GAME OVER", 65, "Arial", 0, 0,  
gWidth, gHeight, , , vbRed, , , , , , , True  
' Dessine l'image de manière permanente dans le  
contrôle  
clGdip.RepaintControl Me.Img  
End If
```

Affichez le formulaire : si le vaisseau touche un ennemi, le jeu s'arrête.

12.5. Test de collision entre les missiles et les ennemis

Pour tester l'intersection entre deux régions, nous utilisons la fonction **RegionsIntersect**

Après chaque création de région d'un missile, on teste si la région de ce missile a une intersection avec un ennemi.

Pour chaque missile, il nous faudra faire une boucle sur les ennemis; on déclare donc un deuxième compteur.

Si le missile entre en collision avec un ennemi, on supprime le missile et l'ennemi.

On gère également le score qui est incrémenté à chaque ennemi détruit. Il faut penser à initialiser le score.

Initialisation du score dans Form Load

```
' Initialisation du score  
TxtScore.Value = 0
```

Ajout de déclaration dans Form Timer

```
Dim lCpt2 As Long ' Compteur
```

Test collision missile avec ennemi dans Form Timer

```
[...]  
' Dessine le missile et crée sa région  
clGdip.DrawRectangle lMissile.X - 1, lMissile.Y -  
lMissile.Size, lMissile.X + 1, lMissile.Y, vbRed,  
vbYellow, 1, , , lMissile.Id  
' Collision missile avec ennemi  
For lCpt2 = gEnnemis.Count To 1 Step -1  
Set lEnnemi = gEnnemis.item(lCpt2)  
' Test de l'intersection entre le missile et  
l'ennemi  
If clGdip.RegionsIntersect(lMissile.Id,  
lEnnemi.Id) Then  
' Si intersection, on supprime l'ennemi  
et sa région  
clGdip.DeleteRegion lEnnemi.Id  
gEnnemis.Remove lCpt2  
' Puis on supprime le missile et sa  
région  
clGdip.DeleteRegion lMissile.Id  
gMissiles.Remove lCpt  
' Et on incrémente le score de 1  
TxtScore.Value = TxtScore.Value + 1  
' On ne peut détruire qu'un seul ennemi  
avec un missile donc on quitte la boucle  
Exit For  
End If  
Next
```

Attention : si on supprime le missile, on ne peut ensuite plus le déplacer.

Pour éviter une erreur, on teste si la région du missile existe avant de le déplacer.

Ajout test d'existence de missile avant déplacement

```
' Déplace le missile s'il n'a pas été supprimé
If clGdip.RegionExists(lMissile.Id) Then
    ' Déplace le missile
    lMissile.Y = lMissile.Y - lMissile.Speed
    ' Si le missile arrive en haut de l'écran, on
le supprime
    If lMissile.Y <= 0 Then
        ' Supprime la région
        clGdip.DeleteRegion lMissile.Id
        ' Supprime l'objet
        gMissiles.Remove lCpt
    End If
End If
```

Affichez le formulaire : les vaisseaux sont détruits par les missiles.

13. Le son

Et le son dans tout ça? Le jeu manque de bruitage.

Dans ce tutoriel, je ne détaille pas la possibilité d'intégrer des bruitages.

Sachez qu'il est tout à fait possible de le faire en VBA :

- soit avec l'API **PlaySound**, simple à utiliser mais ne gère qu'un seul son à la fois

- soit avec l'API **mcisendString**, plus complexe mais qui peut jouer plusieurs sons simultanément.

Ci-dessous un exemple d'utilisation de **PlaySound**.

Déclaration de PlaySound

```
Public Declare Function PlaySound Lib "winmm.dll" Alias
"sndPlaySoundA" _
    (ByVal lpszSoundName As Any, ByVal uFlags As
```

Long) As Long

Exemple d'utilisation de PlaySound

```
PlaySound CurrentProject.Path & "\\fx\MonSon.wav", 1
```

14. Conclusion

Ce tutoriel est une première approche de ce que l'on peut faire en VBA avec Gdi+.

On a également vu l'utilisation des modules de classe qui nous permettent de structurer proprement notre code.

On pourrait aller bien plus loin en développant des menus, en ajoutant d'autres objets et même animer les images.

La seule limitation est le temps d'exécution qui doit rester inférieur à l'intervalle de minuterie, sous peine de voir apparaître des ralentissements.

Il faut donc ne pas voir trop grand.

Consultez le deuxième tutoriel ([Lien124](#))

Téléchargez les jeux créés avec Gdi+ ([Lien125](#))

Merci à l'équipe Office de developpez.com pour ses relectures, commentaires et encouragements!

15. Téléchargements

Téléchargez la classe au format texte ([Lien126](#))

Téléchargez la base Access de ce tutoriel au format ACCESS 2000 ([Lien127](#))

Retrouvez l'article de Thierry Gasperment en ligne : [Lien128](#)

Blogs 2D/3D/Jeux

Ubisoft recherche des testeurs

Dernièrement Ubisoft a fait part de son désir de trouver des personnes volontaires pour tester des jeux dans leurs locaux. Concrètement cela se traduit aujourd'hui par un communiqué officiel que voici, donc si vous vous sentez l'âme d'un testeur, si pour vous un jeu est une bonne récompense alors tentez votre chance. Pensez surtout que c'est un moyen comme un autre de rentrer dans le milieu et peut-être, de se faire des contacts : chose essentielle ici-bas.

« Afin d'améliorer la qualité de nos jeux et pour mieux répondre aux attentes des joueurs, nous recherchons des joueurs motivés

pour tester nos produits en développement. Pour cela, les personnes intéressées peuvent s'inscrire à l'adresse suivante :

<http://testeurs.fr.ubi.com/> ([Lien129](#))

Cette inscription est réservée aux personnes résidant en région parisienne, nos tests se déroulant dans nos studios de Montreuil sous Bois. Un jeu est offert à chaque participant à la fin des sessions de test. Les informations fournies par les joueurs ne sont utilisées que pour nous permettre de les contacter par mail ou téléphone. »

Retrouvez ce billet sur le blog de la rubrique 2D/3D/Jeux : [Lien130](#)

Liens

Lien1 : <http://www.eclipse-nowyoucan.com/>
Lien2 : <http://dcabasson.developpez.com/articles/java/eclipse/eclipse-now-you-can/>
Lien3 : <http://www.eclipse-nowyoucan.com/?Symposia/Program>
Lien4 : <http://nathanael-roberts.developpez.com/articles/java/eclipse/eclipse-now-you-can-2008/>
Lien5 : http://blog.developpez.com/djo-mos?title=creation_d_une_application_wicket_avec_c
Lien6 : <http://wicket.apache.org/quickstart.html>
Lien7 : <http://dico.developpez.com/html/2251-Conception-build.php>
Lien8 : <http://dico.developpez.com/html/1236-Generalites-en-ligne-de-commande.php>
Lien9 : <http://maven.apache.org/download.html#Installation>
Lien10 : <http://m2eclipse.sonatype.org/update/>
Lien11 : <http://www.wickedshell.net/download.html>
Lien12 : <http://dico.developpez.com/html/903-Langages-servlet.php>
Lien13 : <http://psenechal.developpez.com/tutoriels/java/jee/debuter-avec-wicket-et-eclipse/>
Lien14 : http://www.rencontres-spring.com/public/01_keynote_didier_girard.pdf
Lien15 : http://www.rencontres-spring.com/public/02_keynote_julien_dubois.pdf
Lien16 : http://www.rencontres-spring.com/public/02_keynote_julien_dubois.pdf
Lien17 : http://www.rencontres-spring.com/public/04_roadmap_peter_cooper_ellis.pdf
Lien18 : http://www.rencontres-spring.com/public/05_tomcat_mark_thomas.pdf
Lien19 : http://www.rencontres-spring.com/public/06_spring_juergen_hoeller.pdf
Lien20 : <http://hikage.developpez.com/reportages/java/spring/2008/Rencontres-Spring/>
Lien21 : <http://devzone.zend.com/article/3581-Getting-Started-with-OpenID-and-PHP>
Lien22 : <http://www.openid.net/>
Lien23 : <http://openid.net/developers/>
Lien24 : http://openid.net/specs/openid-authentication-1_1.html
Lien25 : <http://www.openidenabled.com/>
Lien26 : <http://pear.php.net/pepr/pepr-proposal-show.php?id=500>
Lien27 : <http://pear.php.net/package/DB>
Lien28 : http://pear.php.net/package/Crypt_HMAC2
Lien29 : http://pear.php.net/package/Crypt_DiffieHellman
Lien30 : http://pear.php.net/package/Services_Yadis
Lien31 : <http://siege.org/projects/phpMyID/>
Lien32 : <http://sourceforge.net/projects/simpleid/>
Lien33 : <http://wiki.guruj.net/Clamshell!Home>
Lien34 : http://dev.aol.com/article/2007/openid_primer_for_php
Lien35 : <http://openid.net/developers/>
Lien36 : http://www.plaxo.com/api/openid_recipe
Lien37 : <http://www.intertwingly.net/blog/2007/01/03/OpenID-for-non-SuperUsers>
Lien38 : <http://wiki.openid.net/Libraries>
Lien39 : <http://jcrozier.developpez.com/tutoriels/web/php/openid/>
Lien40 : <http://www.ajaxload.info/>
Lien41 : <http://css.developpez.com/tutoriels/icone-chargement-pour-images/fichiers/exemple.html>
Lien42 : <http://css.developpez.com/tutoriels/icone-chargement-pour-images/>
Lien43 : <http://accessites.org/>
Lien44 : <http://css.developpez.com/tutoriels/mise-en-forme-titre/fichiers/exemple1.htm>
Lien45 : <http://css.developpez.com/tutoriels/mise-en-forme-titre/fichiers/exemple2.htm>
Lien46 : <http://www.quirksmode.org/css/display.html>
Lien47 : <http://css.developpez.com/tutoriels/mise-en-forme-titre/fichiers/exemple3.htm>
Lien48 : <http://css.developpez.com/tutoriels/mise-en-forme-titre/fichiers/exemple4.htm>
Lien49 : <http://css.developpez.com/tutoriels/mise-en-forme-titre/fichiers/exemple5.htm>
Lien50 : <http://css.developpez.com/tutoriels/mise-en-forme-titre/fichiers/exemple6.htm>
Lien51 : <http://www.khmerang.com/index.php?p=95>
Lien52 : <http://www.mezzoblue.com/tests/revised-image-replacement/#gilderlevin>
Lien53 : <http://css.developpez.com/tutoriels/mise-en-forme-titre/fichiers/exemple7.htm>
Lien54 : <http://css.developpez.com/tutoriels/mise-en-forme-titre/fichiers/exemple8.htm>
Lien55 : <http://css.developpez.com/tutoriels/mise-en-forme-titre/>
Lien56 : <http://designshack.developpez.com/tutoriels/css/introduction-css3-multiples-colonnes/fichiers/exemples.htm>
Lien57 : <http://www.alistapart.com/articles/css3multicolumn>
Lien58 : <http://www.w3.org/TR/css3-multicol/>
Lien57 : <http://designshack.developpez.com/tutoriels/css/introduction-css3-multiples-colonnes/>
Lien58 : <http://live.gnome.org/Vala>
Lien59 : <http://vincentlaine.developpez.com/tuto/dotnet/mono/>
Lien60 : <http://live.gnome.org/Vala/Syntax>
Lien61 : <http://www.vala-project.org/doc/vala-draft/classes.html#properties>
Lien62 : <http://svn.gna.org/svn/valide/trunk/gtk+-2.0.vapi.diff>
Lien63 : <http://code.google.com/p/waf/>
Lien64 : <http://rodney.id.au/dev/vala/hackers.html>
Lien65 : <http://nicolasj.developpez.com/gtk/vala/>
Lien66 : <http://labs.trolltech.com/blogs/2008/04/11/webkit-designer-forms-and-scripting/>
Lien67 : ftp://ftp.developpez.com/qt/gadget-webkit/fichiers/web_kit.rar
Lien68 : <http://www.developpez.net/forums/d647328/c-cpp/bibliotheques/qt/integrer-gadgets-logiciels-widgets-qwebkit/>
Lien69 : <http://doc.trolltech.com/latest/script-calculator.html>
Lien70 : <http://qt.developpez.com/gadget-webkit/>
Lien71 : <http://jquery.com/src/>
Lien72 : <http://javascript.developpez.com/faq/?page=DOM>
Lien73 : <http://docs.jquery.com/>
Lien74 : <http://pckult.developpez.com/tutoriels/javascript/frameworks/jquery/introduction/>
Lien75 : <http://dico.developpez.com/html/2456-Conception-framework.php>

Lien76 : <http://prototypejs.org/>
Lien77 : <http://script.aculo.us/>
Lien78 : <http://script.aculo.us/dist/scriptaculous-js-1.8.2.zip>
Lien79 : <http://dmouronval.developpez.com/tutoriels/web/frameworks/scriptaculous/introduction/exemples/fading2.html>
Lien80 : <http://dmouronval.developpez.com/tutoriels/web/frameworks/scriptaculous/introduction/exemples/fading1.html>
Lien81 : ftp://ftp-developpez.com/dmouronval/tutoriels/web/frameworks/scriptaculous/protopack_v2.20.zip
Lien82 : <http://groups.google.com/group/prototype-core/files>
Lien83 : <http://dcabasson.developpez.com/articles/javascript/ajax/documentation-prototype-1.4.0/>
Lien84 : <http://dmouronval.developpez.com/tutoriels/web/frameworks/scriptaculous/introduction/>
Lien85 : <http://dmouronval.developpez.com/tutoriels/web/frameworks/scriptaculous/effets/>
Lien86 : <http://www.agiletour.com/quelpublicAT2008.html>
Lien87 : <http://www.agiletour.com/node/45/results>
Lien88 : <http://www.aubryconseil.com/dotclear/index.php/2007/11/18/331-la-velocite-n-est-pas-une-mesure-de-productivite>
Lien89 : http://www.sigmat.fr/public/SigmaAgileTour/AT2008_prez_MethodesAgiles.pdf
Lien90 : http://www.sigmat.fr/public/SigmaAgileTour/AT2008_demo_OutilsAgiles.pdf
Lien91 : http://www.sigmat.fr/public/SigmaAgileTour/AT2008_rex_RetourSurXpIGE OSS.pdf
Lien92 : <http://www.sigmat.fr/public/SigmaAgileTour/AgiliteEnSituation.pdf>
Lien93 : <http://www.clubagile.org/>
Lien94 : <http://www.poppendieck.com/>
Lien95 : [ftp://ftp-developpez.com/bruno-orsier/fichiers/\(developpez\)_Retour_experience_AgileTour_2008.pdf](ftp://ftp-developpez.com/bruno-orsier/fichiers/(developpez)_Retour_experience_AgileTour_2008.pdf)
Lien96 : http://www.versionone.com/pdf/3rdAnnualStateOfAgile_FullDataReport.pdf
Lien97 : <http://www.ehsavoie.com/2008/10/agile-tour-grenoble-2008-les-slides.html>
Lien98 : <http://agilemanifesto.org/>
Lien99 : <http://maven.apache.org/>
Lien100 : <https://hudson.dev.java.net/>
Lien101 : <http://fr.wikipedia.org/wiki/KISS-principe>
Lien102 : <http://wiki.agilefinland.com/?CodingDojo>
Lien103 : <http://bruno-orsier.developpez.com/principes/dry/>
Lien104 : <http://alistair.cockburn.us/>
Lien105 : <http://alistair.cockburn.us/Agile+software+development+book>
Lien106 : <http://alistair.cockburn.us/Crystal+Clear+book>
Lien107 : <http://bruno-orsier.developpez.com/tutoriels/TDD/pentaminos/>
Lien108 : <http://conception.developpez.com/reportage/agilite/agile-tour-2008/>
Lien109 : <http://macstyle.fr/>
Lien110 : <http://anelzin.developpez.com/tutoriels/hardware/batteries/>
Lien111 : <http://code.google.com/p/jscooca/issues/list>
Lien112 : <http://parmanoir.com/>
Lien113 : <http://inexdo.com/JSCocoa>
Lien114 : <http://code.google.com/p/jscooca/>
Lien115 : <http://mac.developpez.com/interviews/patrick-geiller/>
Lien116 : <http://www.gtk-osx.org/>
Lien117 : <http://aurelien-gaymay.developpez.com/tutoriels/mac/xcode/ada/>
Lien118 : <http://aurelien-gaymay.developpez.com/tutoriels/mac/gtk/>
Lien119 : http://www.pearson.fr/resources/download.cfm?GCOI=27440100977420&thefile=2257_Scripts_OSX.zip
Lien120 : <http://www.pearson.fr/livre/?GCOI=27440100977420>
Lien121 : <http://mac.developpez.com/livres/>
Lien122 : <http://arkham46.developpez.com/articles/office/clgdiplus/>
Lien123 : <ftp://ftp-developpez.com/arkham46/articles/office/clgdiplus/doc/fichiers/clgdiplus.zip>
Lien124 : <http://arkham46.developpez.com/articles/office/clgdiplus/tuto/tutoclgdiplusgame2/>
Lien125 : <http://arkham46.developpez.com/articles/office/clgdiplus/#LIV>
Lien126 : <ftp://ftp-developpez.com/arkham46/articles/office/clgdiplus/doc/fichiers/clgdiplus.zip>
Lien127 : <ftp://ftp-developpez.com/arkham46/articles/office/clgdiplus/tuto/tutoclgdiplusgame/fichiers/tutogame.zip>
Lien128 : <http://arkham46.developpez.com/articles/office/clgdiplus/tuto/tutoclgdiplusgame/>
Lien129 : <http://testeurs.fr/ubi.com/>
Lien130 : http://blog.developpez.com/jeux?title=ubisoft_recherche_des_testeurs