

# Présentation de JAVA FX 8

# JAVAFX 8 EN QUELQUES MOTS

1, 2... et 8!

- Troisième itération du toolkit officiel d'Oracle pour la définition d'Interfaces Homme-Machine (IHM) en Java
- Il a pour but de permettre la définition d'interfaces riches et modernes sur toutes les Java Virtual Machines (JVM) disponibles
- JavaFX 8 supporte les dernières évolutions technologiques hardware
  - Optimisé pour l'accélération matérielle GPU (Direct3D et OpenGL)
  - Optimisé pour les architectures multi-cores
  - Déployable sur tablettes Android et iOS via JavaFX Android port et RoboVM
  - Support dans Java Embedded Virtual Machine pour architectures CPU ARM

# JAVAFX 1.0 LES ORIGINES

Une première version décevante

- Issu du framework F3 créé par SeeBeyond en 2005
- Sorti en 2007 après le rachat par Sun de F3
- Runtime propre à ajouter à la JVM : similaire à l'installation d'un plugin
- Support d'un langage déclaratif non XML (appelé aujourd'hui VISAGE)
- Performances graphiques insuffisantes
- Compatibilité difficile avec les autres toolkits graphiques Java : Swing, Eclipse RCP

**Adoption très marginale par la communauté**

# JAVAFX 2.\* BÂTI POUR DURER

On change (presque) tout

- Performances accrues
- Support natif dans la JVM (à partir de Java 7)
- API riches disponibles (médias) et documentées
- Nouveau langage déclaratif FXML basé sur XML
- Nouveaux modes de déploiement supportés
- Intégration HTML 5 via WebKit

**Oracle veut proposer JavaFX comme remplacement de plusieurs API Java historiques : Swing, AWT, Applet, API vidéo et audio**

# JAVAFX 8 L'OFFENSIVE D'ORACLE

A la conquête de nouveaux marchés

- Utilisable sur JVM ARM Java Embedded
- Porté sur Android / iOS
- Nouvelles API 3D
- Nouveaux widgets
- Open source!
- Meilleur support de HTML5
- Intégration avec Swing efficace

**Java 8 représente une évolution majeure pour Oracle, JavaFX 8 fait partie des innovations poussées**

# SUCCÉDER À SWING

Le monde a changé depuis 1997

- Modernisation de la conception d'IHM en Java
  - Gérer les animations / effets
  - Gérer des flux multimédias
  - Intégrer de nouvelles « Gestures » de type touch
- Faciliter le déploiement d'un Client lourd / riche
  - Options Web (WebStart, Embedded)
  - Options Desktop (avec JVM ou sans)
- Améliorer le système de databinding
  - Principe des Listeners
  - Bindings avec Property

# SWING ET AWT

## Petit retour sur les dinosaures Java

- AWT est le premier toolkit IHM disponible en Java
  - Composants "lourds", utilisant le système
  - Limité à l'intersection des capacités des différentes plate-formes
- Swing introduit en Java 1.2
  - Se repose en partie sur AWT (essentiellement pour raisons de compatibilité)
  - Composants "légers", entièrement gérés par Swing
  - Grande liberté dans la définition des interfaces
  - Outils WYSIWYG disponibles (What You See Is What You Get)
  - Adoption très large par la communauté

# PARTICULARITÉS DE SWING

L'abandonware le plus connu du JDK

- Description des composants et systèmes de layout entièrement en Java
- Notion de GUI Thread
  - obligation de gérer / comprendre plusieurs threads sous peine de performances dégradées
- Peu d'évolutions depuis plusieurs années

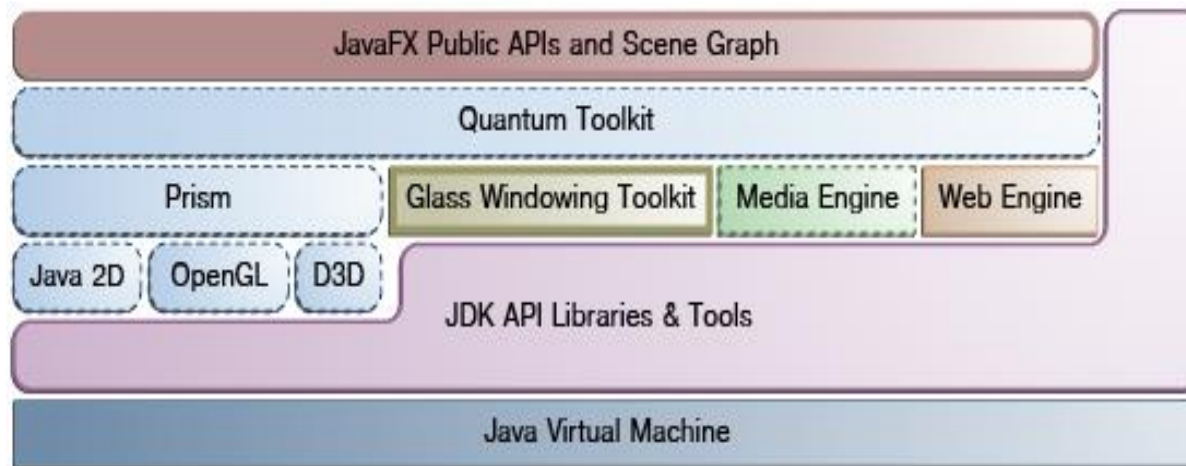
**De manière générale : des efforts assez importants à fournir pour maîtriser le rendu graphique (performance, style...) car Swing reste assez « bas niveau »**



# LE RUNTIME JAVA FX

## Un saut technologique

- Le moteur de rendu est sélectionné par rapport à la plateforme (DirectX, OpenGL, Software rendering au pire)

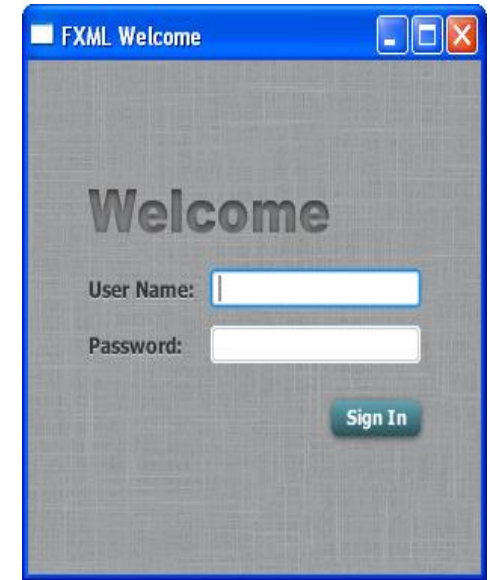


- API riche et de type « haut niveau »
- Nombreux composants disponibles
- Bibliothèque d'animations et effets fournie

# FXML + CSS + JAVA

## Un combo intéressant

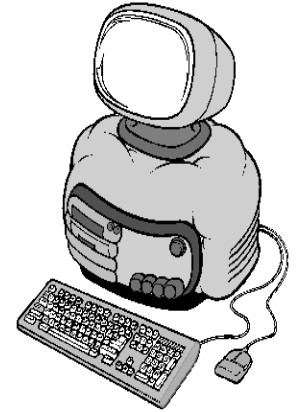
- FXML : XML descripteur de l'IHM et de sa mise en page (*layout*)
  - Semblable à XAML, XUL, MXML, XHTML...
  - Transfert de compétence facilité
  - Intégration facile à un IDE et à un outil WYSIWYG
  - Limite la verbosité de la déclaration d'IHM en Java
- CSS pour personnaliser l'apparence de l'application
  - Meilleure maîtrise du rendu global
  - Meilleure maintenabilité
- Java : la valeur sûre
  - Utilisé pour le comportement de l'application, la logique métier
  - Fiable, maintenable, scalable



# LE CLIENT LOURD AUJOURD'HUI

Pourquoi?

From Computer Desktop Encyclopedia  
© 1998 The Computer Language Co., Inc.



- Créer des applications avec mode déconnecté
- Définir des écrans complexes
- Enrichir d'effets graphiques, utiliser des visuels avancés
- Tirer partie des interactions avec le système et le hardware
- Meilleure durabilité et maintenabilité que les interfaces Web
- Java est plus maintenable que la plupart des langages pour le Web
- Non soumis à l'évolutivité et la diversité des navigateurs Web
- Adhère à Java et la JVM : support maîtrisé par Oracle
- Prochainement : multi-support (iOS et Android)
- Etc...

# LE MULTI-SUPPORTS

Indispensable aujourd'hui

- Intégré au JDK Oracle depuis la version 1.7
- Disponible sous Windows, MacOSX, Linux
- Version pour ARM avec Java Embedded (Raspberry Pi, ...)
- Gestion du multitouch
- Portages disponibles sur iOS et Android

