

Intro JavaFX

12 décembre 2020

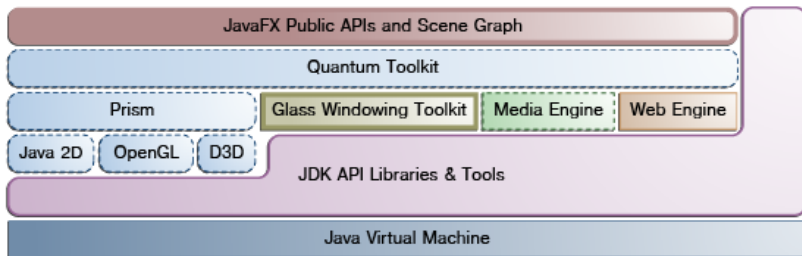
Les IHM en Java : d'AWT à JavaFX.

- AWT : Abstract Windows Toolkit. (Utilise composants natifs du système)
 - interactions classiques : fenêtres, souris, claviers.
 - **programmation évènementielle.**
- SWING : à partir de Java 2. (Composants purs java pour l'essentiel)
- JavaFX : (depuis 2011)
 - plateforme pour créer des applications plus riches :
 - animations, - pages web,
 - lecteur média, - 3D
 - simplifier le développement d'application :
 - properties and binding - SceneBuilder and fxml
 - avant version 2.0 langage de script (plus supporté)
bibliothèque native java (bibliothèque standard de Java8 → Java11)

Programmation évènementielle : principe général.

- Il y a des **évènements** (Event) : (on peut créer nos propres évènements). évènements de bas niveau/haut niveau (clic-souris/action sur un bouton).
Remarque : un même Event peut concerner plusieurs objets de l'interface.
 - JavaFX gère la propagation/distribution des évènements.
 - il doit être possible d'intervenir sur cette propagation.
- On écrit des **écouteurs** (EventListener) qui vont réagir à ces évènements.
 - Différents EventListener, spécialisé chacun pour une classe d'évènements.Exemple : On écrit un EventListener *el* réagissant aux *MouseEvent*.
- On **abonne** les composants de l'interface aux EventListener.
Exemple : on abonne le champ texte *c* à *el*.
⇒ JavaFX sait que *c* doit réagir aux mouvements de la souris.
- Le système gère les évènements pendant l'exécution :
Si on bouge la souris sur *c*, → on exécute le code de *el*.

Programmation événementielle: Architecture



- JavaFX Public API Scene Graph : Point de construction de l'application java FX.
 - Quantum Tool Quit : fait l'interface entre le niveau supérieur et
 - PRISM : moteur de rendu graphique (3D compris). Utilise moteurs hardware et software sinon (Java2D si pas d'acceleration matérielle).
 - Glass Windowing Toolking : gestion de fenêtres, timers, surfaces ; gestion de la file d'évènements.
- gère les différents threads entre rendu et **gestion d'évènements**.

Scene Graph fournit différents composants (**Node**) à placer :

- **Shape** purements graphiques : Circle, Line, Texte, ...
- **Control** spécialisés pour interagir
 - simples : Button, TextField, CheckBox, Label ...
 - plus évolués : Slider, DatePicker, ...
- **Pane** des panneaux/conteneurs :
 - de base : Pane :
 - avec gestionnaires de placement : FlowPane, BorderPane, HBox, ...
- Autres :
 - Canvas (Panneau de dessin).
 - WebView, MediaView, ...

Caractéristiques

- **AWT**, **SWING**, **JavaScript** et **HTML5** interopérabilité.
- **Property** et **binding** : gestion "simplifiée" des événements (pour certains cas).
- graphiques 2D et 3D,
- tire profit des avancées des architectures modernes mobiles et PC
- contrôles UI "open-source" avancés et **CSS** (Cascading Style Sheets)
- lit les contenus web multimédias **VP6** et **MP3**
- boîte à outil média basée sur *GStreamer*
- *WebView* : lecteur web intégré, rendu *HTML* avec *WebKit*.
- *FXML* (XML définissant l'interface de l'application)

Caractéristiques de JavaFX : Property et Binding

Simplifie la gestion des événements, et des traitements de ceux-ci.

- **Property** : objet encapsulant une valeur.

IntegerProperty, DoubleProperty,...

- événements lorsque la propriété est changée,
- possibilité récupérer et traiter ces événements.

⇒ propagation automatique de certaines modifications.

les caractéristiques des composant JavaFX sont des propriétés :
width d'un bouton, layoutX d'une étiquette, value d'un slider.

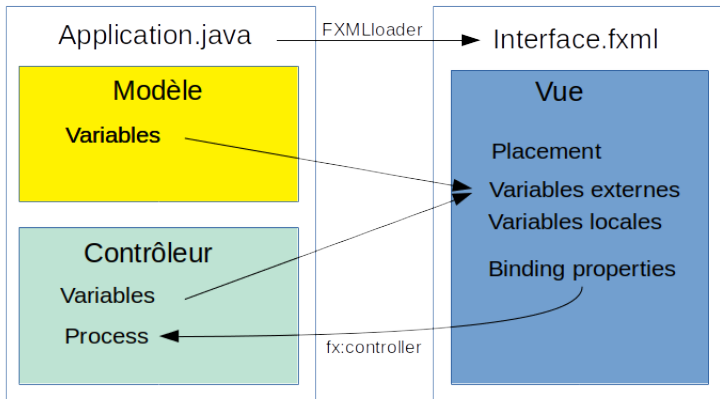
- **Binding** : liaison entre 2 Properties $P1$ et $P2$.

- Si $P1$ est modifiée, alors $P2$ est automatiquement modifiée. -
Binding unidirectionnel, ou bidirectionnel.

- et les outils autour (en particulier pour pouvoir faire des calculs sur les propriétés).

Caractéristiques de JavaFX : JavaFX et FXML

FXML : langage de script à *la XML* interprété par javaFX pour construire la vue.



SceneBuilder : outil graphique pour faire le fichier XML.

Caractéristiques de JavaFX : JavaFX et FXML

Programmation en utilisant :

Java API pour JavaFX

- Pur développement java
- toutes les possibilités de java (generics, annotations, multi-threading, etc)
- compatible IDE (Netbeans, Eclipse...)
- outils script JavaFX (bind()...) toujours utilisables
- langages alternatifs supportés par la JVM (Groovy, Scala...)

FXML et CSS

- FXML : langage de script (basé sur XML) pour définir l'interface
- apprentissage facile pour les développeurs web
- autorise l'inclusion de scripts écrits en langages supportés par la JVM.
- outil de création **SceneBuilder**

Caractéristiques de JavaFX : Animations

la classe **Animation** (et leurs filles) permettent de faire des animations : déplacement, changement de couleur, de taille, etc des noeuds)

- ① **Timeline** : permet de fixer l'évolution d'une (ou des) propriétés.
 - valeur de départ, valeur finale, durée d'évolution, etc...
 - methodes start, pause, reset,
 - évènements associés.

⇒ si je fixe un Timeline sur le layoutX d'un bouton, ce bouton va se déplacer.

Je peux faire évoluer l'épaisseur d'un trait, une couleur, etc...
- ② **Transition** : permet de fixer une trajectoire sur un composant.

même principe, mais plus de choix dans les trajectoires.

possibilité de paralléliser les transitions (`RotateTransition` et `TranslateTransition` par exemple)