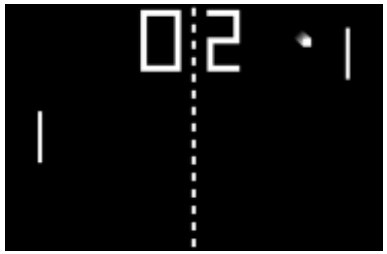
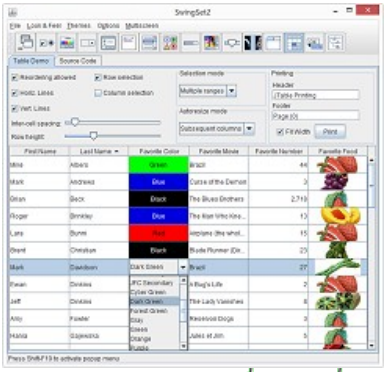
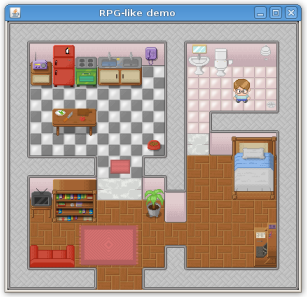
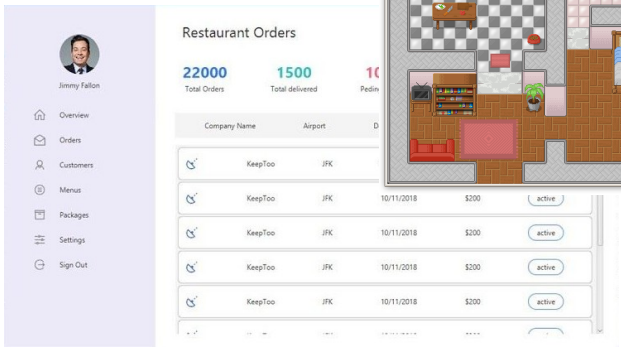
Interfaces Homme Machine

JAVA FX









Fabrice PELLEAU

L'importance de l'IHM

- Pour les utilisateurs
  - « Le produit c'est l'IHM »

    (*Jef Raskin*)
- Pour les développeurs
  - « 80% du code des systèmes interactifs est consacré à l'interface utilisateur »

    (*Laurence Nigay*)
- Pour ceux qui financent les produits
  - « 63% des gros projets informatiques connaissent des dépassements de coûts »

    (*Sally Greenberg*)

APIs Graphiques Java

- AWT
  - depuis Java 1 / natif
- Swing
  - depuis Java 2 / pure Java
  - Basée sur AWT, sans ressources
- Java FX
  - Java + XML
- SWT
  - Concurrent Swing, créée pour Eclipse

Environnement de développement

- Java
  - 8 : contient le SDK Java FX
  - 11 : on doit lui ajouter le SDK Java FX
- Eclipse
  - 4 ou plus
  - Plugin e(fx)clipse (Cf. Marketplace)
- Scene Builder
  - V11+/Java11 ou V8/Java8
  - Gluon fourni une version compilée
  - Oracle distribue le code source

JavaFX et Java 8

- Installer
  - JDK8
  - Eclipse+e(fx)clipse
  - Scene Builder 8
- Configurer Eclipse
  - Installed JRE => jdk8
  - Complier => 1.8
  - JavaFX => path de scenebuilder.exe
  - Encodage => utf-8

JavaFX et Java 11

- Installer
  - JDK11
  - Eclipse+e(fx)clipse
  - Scene Builder 11 ou plus
- Configurer Eclipse
  - Installed JRE => jdk11
  - Complier => 11
  - JavaFX => path de scenebuilder.exe
    => path du SDK JavaFX
  - Utiliser les modules (conseillé)
    - Ou (sans module) Run Configuration / VM arguments
  - `--module-path="C:\Program Files\java\javafx-sdk-11\lib"`
    `--add-modules=javafx.controls,javafx.fxml`

JavaFX et Java 17

- Installer
  - JDK17
  - Eclipse+e(fx)clipse
  - Scene Builder 17 ou plus
- Configurer Eclipse
  - Installed JRE => jdk17
  - Complier => 17
  - JavaFX => path de scenebuilder.exe
    => path du SDK JavaFX
  - Utiliser les modules (conseillé)
    - Ou (sans module) Run Configuration / VM arguments
    - `--module-path="C:\Program Files\java\javafx-sdk-17\lib" --add-modules=javafx.controls,javafx.fxml`

Application JavaFX

- Chaque application est dans un premier temps basée sur une classe dérivée de `javafx.application.Application`
- Cycle de vie :
  - Construit une instance de la classe Application spécifiée
  - Appelle la méthode `init()`
  - Appelle la méthode `start(javafx.stage.Stage)`
  - Attend l'achèvement de l'application,
  - Appelle la méthode `stop()`
- `start()` est abstraite mais `init()` et `stop()` ont une implémentation par défaut (vide)

javafx.scene.layout.**Pane**

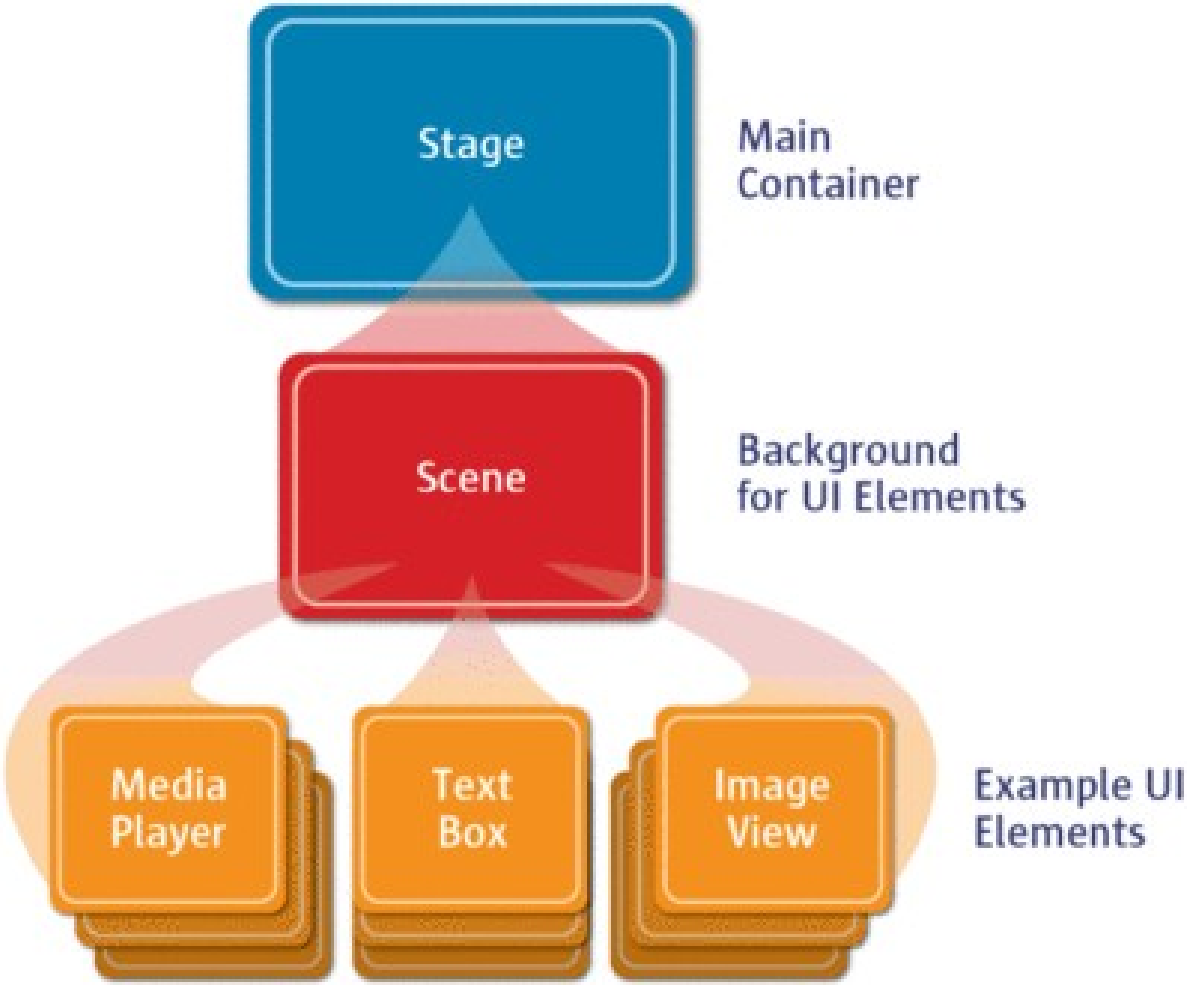
- Les composants Pane sont des nœuds graphiques chargés de gérer la disposition d'un ensemble de nœuds.
  - Contrairement aux Panels Swing auquels on applique un LayoutManager interchangeable, chaque Pane à sa propre façon de faire (et ses propres méthodes).

- Démonstration en ligne des principaux Pane :
  - java.developpez.com/faq/javafx/?page=Mise-en-page
  - (Google : JavaFX Mise en page)

javafx.scene.layout.**Pane**

- `BorderPane`
  - cinq parties : top, down, right, left, center.
- `Hbox / VBox`
  - aligner horizontalement / verticalement
- `StackPane`
  - Empilement (le dernier arrivé est au dessus)
- `GridPane`
  - grille organisée en lignes et colonnes
- `FlowPane, TilePane`
  - Comme des mots (même taille ou non)
- `AnchorPane`
  - Un élément par rapport aux bords : top, bottom, right et left.

Structure d'une application JavaFX



Composants JavaFX

- L'arborescence des composants JavaFX est très riche
- Ce sont tous les « `Node` » :
  - `Button, Label, TextField,` etc.
  - `TextArea, HTMLEditor,` etc.
  - `TableView, TreeView,` etc.
  - `ColorPicker, DatePicker,` etc.
  - `Alert, FileChooser,` etc.
- Sites démonstratifs :
  - Jakob Jenkov `tutorials.jenkov.com/javafx`
    - (google : JavaFX Jenkov)
  - devstory.net `devstory.net/11009/javafx`
    - (google : JavaFX devstory)

Comment créer une IHM ?

- C’est un assemblage de composants

FXML

- Le langage FXML (dialecte XML) permet la définition déclarative des interfaces graphiques
  - IHM = arborescence de composants
  - FXML = arborescence de balises
    - Une balise est un composant
- Il existe des éditeurs Graphiques
  - notamment Scene Builder
- *Une vue FXML sera associée à un objet Java (son « controller »). Il porte :*
  - *Les initialisations algorithmiques*
  - *Les méthodes associées aux Handlers*
  - *Les relations entres les vues*

Mise en forme CSS

- La mise en forme des applications JavaFX peut être paramétrée en utilisant les **`Cascading Style Sheets`** (`CSS`)
- Avantages : Souplesse, Standardisation
- Charte par défaut : modena.css
  - jfxrt.jar
  - com.sun.javafx.scene.control.skin.modena/modena.css
- Tutoriel Oracle
  - docs.oracle.com/javase/8/javafx/user-interface-tutorial/css_tutorial.htm
- Documentation de référence
  - docs.oracle.com/javase/8/javafx/api/javafx/scene/doc-files/cssref.html

CSS JavaFX

- La construction est la même que pour le langage CSS Web.
  - Les sélecteurs sont adaptés
    - .label, .menu-bar, etc.
  - Il existe des pseudos classes
    - :default, :selected, etc.
  - On peut créer ses propres classes (et les associer aux composants)
  - Les directives sont proches des directives web mais préfixées par `-fx-`