

Interaction Humain-Machine

TP : Gestion du placement de widgets

Objectifs

- configurer l'environnement de travail
- se familiariser avec les widgets, conteneurs et gestionnaires de placement.

Pré-requis

— Maîtriser les parties 1^1 et 2^2 du cours 1.

1 Hello JavaFX

D'une manière générale votre code pourra s'inspirer de l'exemple minimal suivant :

Astuce : pour éviter de devoir taper les imports à la main, vous pouvez utiliser la fonctionnalité d'auto-complétion de votre IDE en utilisant les touches Alt + Shift + O.

```
import javafx.application.Application;
import javafx.scene.Scene;
import javafx.scene.control.Label;
import javafx.scene.layout.VBox;
import javafx.stage.Stage;
public class SimpleScene extends Application {
  public void start(Stage stage) {
    VBox root = new VBox();
    Label msg = new Label("Hello JavaFX");
    root.getChildren().add(msg);
    Scene scene = new Scene(root, 300, 50);
    stage.setScene(scene);
    stage.setTitle("Hello JavaFX");
    stage.show();
  }
  public static void main(String[] args) {
    Application.launch(args);
  }
}
```

2 Configuration de Visual Studio Code ou VSCodium

(Non nécessaire en salle TP) Téléchargez le JavaFX SDK³ pour votre système (version 21.0.7) et décompressez l'archive à un endroit connu. Dans les salles TP, le répertoire se trouve à l'adresse suivante : /home/public/javafx-sdk-21.0.7/lib/

Les informations complémentaires sont disponibles à cette adresse : https://gitlab.univ-lille. fr/gery.casiez/javafxvscodeconfig

^{1.} https://www.iut-info.univ-lille.fr/~gery.casiez/R2.02/Cours/R2.02-Cours1partie1.pdf

^{2.} https://www.iut-info.univ-lille.fr/~gery.casiez/R2.02/Cours/R2.02-Cours1partie2.pdf

^{3.} https://gluonhq.com/products/javafx/

3 Stage

L'objet Stage⁴ est le conteneur principal qui est habituellement une fenêtre avec des bordures.

Q 1. Testez les différentes méthodes disponibles pour mettre la fenêtre en plein écran, toujours devant les autres fenêtres, régler son degré d'opacité (ne fonctionne pas sur les machines des salles de TP) et la rendre non redimensionnable. Testez également les différents styles de fenêtres (decorated, undecorated, transparent, utility).

Q 2. Ajoutez une seconde fenêtre modale à la première (notion vue en cours). Vous utiliserez pour cela les méthodes initOwner⁵ et initModality⁶.

Q 3. La fenêtre apparaît par défaut centrée à l'écran. Utilisez les méthodes set X⁷ et set Y pour positionner la fenêtre modale juxtaposée à droite de la fenêtre parente.

4 Gestionnaires de placement

Le gestionnaire de placement (layout) a pour rôle de gérer la position des composants d'une interface. Différents gestionnaires de placement existent, chacun représentant une stratégie de placement particulière.

4.1 FlowPane

Q 4. FlowPane⁸ permet de placer les noeuds les uns à côté des autres, de gauche à droite par défaut. Ajoutez 10 boutons dans un FlowPane de manière à obtenir le résultat visible Figure 1. Observez comment évolue la position des boutons quand vous redimensionnez la fenêtre. Pour modifier les marges, utilisez la méthode statique setMargin⁹ de FlowPane.

4.2 HBox et VBox

Q 5. Testez de la même façon Hbox¹⁰ et VBox¹¹.

4.3 BorderPane

Q 6. Ce gestionnaire de placement ¹² utilise 5 positions pour placer les composants. Ajoutez un bouton dans chaque emplacement. Notez comment les boutons sont positionnés ainsi que leurs dimensions. Pour obtenir le résultat représenté Figure 1, modifiez la taille maximale des boutons avec la méthode setMaxSize¹³ et en passant comme paramètres Double.MAX_VALUE. Pour modifier les marges, utilisez la méthode statique setMargin¹⁴ de BorderPane.

4.4 TilePane

Q 7. Ajoutez 16 boutons dans un TilePane¹⁵ de manière à obtenir le résultat visible Figure 2. Vous utiliserez à nouveau setMaxSize pour que les boutons aient tous la même taille.

https://docs.oracle.com/javase/8/javafx/api/javafx/stage/Stage.html#initOwner-javafx.stage.Window https://docs.oracle.com/javase/8/javafx/api/javafx/stage/Stage.html#initModality-javafx.stage.

Modality-

^{4.} https://docs.oracle.com/javase/8/javafx/api/javafx/stage/Stage.html

^{7.} https://docs.oracle.com/javase/8/javafx/api/javafx/stage/Window.html#setX-double-

^{8.} https://docs.oracle.com/javase/8/javafx/api/javafx/scene/layout/FlowPane.html

^{9.} https://docs.oracle.com/javase/8/javafx/api/javafx/scene/layout/FlowPane.html#setMargin-javafx.scene. Node-javafx.geometry.Insets-

^{10.} https://docs.oracle.com/javafx/2/api/javafx/scene/layout/HBox.html

^{11.} https://docs.oracle.com/javafx/2/api/javafx/scene/layout/VBox.html

^{12.} https://docs.oracle.com/javase/8/javafx/api/javafx/scene/layout/BorderPane.html

^{13.} https://docs.oracle.com/javase/8/javafx/api/javafx/scene/layout/Region.html#setMaxSize-double-double-

^{14.} https://docs.oracle.com/javase/8/javafx/api/javafx/scene/layout/BorderPane.html#setMargin-javafx.

scene.Node-javafx.geometry.Insets-

^{15.} https://docs.oracle.com/javase/8/javafx/api/javafx/scene/layout/TilePane.html

		BorderPane exemple	
		Nord	
FlowPane exemple Bouton 1 Bouton 2 Bouton 3 Bouton 4 Bouton 5 Bouton 6 Bouton 7 Bouton 8 Bouton 9 Bouton 10	West	Center	East
		South	

FIGURE 1 - Illustration de FlowPane et BorderPane

4.5 GridPane

Q 8. Ajoutez 10 boutons dans un GridPane¹⁶ de manière à obtenir le résultat visible Figure 2. Vous utiliserez à nouveau setMaxSize pour qu'un bouton puisse occuper plusieurs lignes ou colonnes.

						GridPane exemple	
•••	Ti	lePane exemp	le		Bouton 1	Bouton 2 Bouton 3	Bouton 4
Bouton 1	Bouton 2	Bouton 3	Bouton 4	Bouton 5		Bouton 5	
Bouton 6	Bouton 7	Bouton 8	Bouton 9	Bouton 10	 		
Bouton 11	Bouton 12	Bouton 13	Bouton 14	Bouton 15		Bouton 6	Bouton 7
Bouton 16					Deviter 0	Bouton 9	
					Bouton 8	Bouton 10	

FIGURE 2 – Illustration de TilePane et GridPane

4.6 Positionner manuellement les noeuds

Q 9. Utilisez Pane¹⁷ pour positionner manuellement 2 boutons en utilisant les méthodes setLayoutX et setLayoutY, de manière à obtenir un résultat similaire à la figure 3.

•••	Pane exemple
Bouton 1]
Bo	uton 2

FIGURE 3 – Illustration de Pane

^{16.} https://docs.oracle.com/javafx/2/api/javafx/scene/layout/GridPane.html

^{17.} https://docs.oracle.com/javase/8/javafx/api/javafx/scene/layout/Pane.html



FIGURE 4 – Un éditeur de texte.

5 Un éditeur de texte

Q 10. Proposez une hiérarchie de composants pour reproduire l'interface d'un éditeur de texte représenté Figure 4.

Q 11. Implémentez la proposition. Pour le menu, vous utiliserez MenuBar¹⁸ à laquelle vous ajouterez des MenuItems¹⁹. Les séparateurs dans les menus s'obtiennent en utilisant SeparatorMenuItem²⁰. Liens vers l'icône de la flèche de gauche²¹ et celle de droite²²

6 Pour les plus rapides

Q 12. Reproduisez aussi fidèlement que possible l'interface représentée Figure 5.

7 BONUS : Création d'applications natives avec JavaFX

Le déploiement permet de créer un installeur et une application sous la forme d'un .msi sous Windows ou d'un .app sous Mac. Le déploiement intègre le .jar et la machine virtuelle Java si bien qu'un utilisateur de votre application n'a rien de supplémentaire à installer pour l'utiliser. A ne pas tester sur les machines des salles TP.

Si vous voulez en savoir plus, vous pouvez consulter le projet maven-jpackage-template²³.

^{18.} https://docs.oracle.com/javase/8/javafx/api/javafx/scene/control/MenuBar.html

^{19.} http://docs.oracle.com/javase/8/javafx/api/javafx/scene/control/MenuItem.html

^{20.} https://docs.oracle.com/javafx/2/api/javafx/scene/control/SeparatorMenuItem.html

^{21.} previous_motif.gif

^{22.} next_motif.gif

^{23.} https://github.com/wiverson/maven-jpackage-template

type filter text HelloPaint SimpleScene Convertisseur	Filter Closed Projects Filter Deleted Projects	Main Arguments VM Arguments: module-path C:\U javafx.controls.javafx Working directory: Default: Other:	Isers\Pierre\IHM\javafx-sdk-18\libadd-modules x.fxml \${workspace_loc:TP3}
			Show command line Revert Apply

FIGURE 5 – Extrait de l'interface d'Eclipse.