

# Interfaces Graphiques

## TPs 4, 5, 6 et 7 - Architecture MVC - Twisk

### TP6

L'objectif de cette séance est de donner à l'utilisateur la possibilité de sélectionner une ou plusieurs activités, de les renommer, de les supprimer, de les déplacer, avec les arcs qui lui sont attachés.

N'oubliez pas de compléter le diagramme de classes, il vous sera demandé à la fin de ces quatre séances.

### 1- Activer la sélection d'une étape

Un clic sur une étape sélectionne l'étape ; un second clic sur la même étape supprime la sélection. Ajoutez un écouteur qui prévient le monde ; celui-ci garde la mémoire des étapes sélectionnées. Imaginez un style particulier pour identifier visuellement les étapes sélectionnées.

### 2- Ajouter un menu

Ajoutez une nouvelle vue **VueMenu** sous-classe de **MenuBar**. Dans le constructeur de **VueMenu**, ajoutez dans le composant un menu *Fichier* (instance de **Menu**) avec le seul item *Quitter* (instance de **MenuItem**), et un menu *Édition* avec le seul item *Supprimer*.

L'écouteur de l'item *Quitter* fait simplement un appel à **Platform.exit()**; ce qui a pour effet de fermer l'application.

L'écouteur de l'item *Supprimer* supprime toutes les activités sélectionnées, ainsi que les arcs qui lui sont attachés.

### 3- Renommer une activité

Dans le menu *Édition*, ajoutez avec l'item *Renommer la sélection*. La saisie du nouveau nom se fait par l'intermédiaire d'une boîte de dialogue (instance de **TextInputDialog**).

Pour que cet item ne soit actif que si la sélection est réduite à une seule activité, la classe qui gère le menu doit être une vue, rafraîchie à chaque transformation du modèle (donc elle doit implémenter **Observateur**).

## 4- Déplacer une activité

Le déplacement d'une activité est réalisé par le *drag'n drop*. Ce mécanisme met en jeu une source et une destination, ainsi qu'une information qui va transiter de l'une à l'autre. Cette information peut être une chaîne de caractères qui identifie la source et/ou une image qui sert à suivre visuellement le déplacement de la source.

Voilà deux pages qui vont vous aider dans la réalisation :

- <https://java.developpez.com/faq/javafx/?page=Drag-n-Drop>
- <https://docs.oracle.com/javase/8/javafx/events-tutorial/paper-doll.htm>

La première page détaille chaque point de ce mécanisme ; prenez le temps de la lire. La seconde (en anglais) détaille un autre exemple.

Pour gérer le *drag'n drop*, il faut écrire plusieurs écouteurs :

- un écouteur sur chaque source (donc chaque activité, instance de **VueEtapeIG**), activé au moment du *drag* (**setOnDragDetected**),
- un écouteur sur le panneau (instance de **VueMondeIG**) pour accepter le *drop* (**setOnDragOver**),
- un autre écouteur sur le panneau, activé au moment du drop (**setOnDragDropped**).

Pour déclencher le *drag'n drop* dans l'écouteur de la source, il faut appeler la fonction **startDragAndDrop**, avec en paramètre le type de *drag'n drop* (copie, transfert, lien) ; ici, il s'agit d'un transfert donc **TransferMode.MOVE**. Cette fonction retourne un objet de type **DragBoard** (un genre de presse-papier) qui doit être rempli avec l'information qui va passer de la source à la destination, en passant par l'intermédiaire de **ClipboardContent**. Par exemple, ici, on range dans le **DragBoard** l'identifiant de l'activité source et on attache cet identifiant au composant graphique correspondant (**setId**). On peut également y placer une image, comme la capture de l'objet, de sorte que le déplacement soit visualisé.

Pour réaliser cette nouvelle fonctionnalité, vous pouvez procéder par étape :

- Écrire les trois écouteurs sans oublier de placer un identifiant unique dans le **ClipboardContent** du **DragBoard** ; cet identifiant doit être également attribué à la source en dehors de l'écouteur. Par un affichage sur la sortie standard dans les écouteurs de la source et de la destination, vérifier l'identifiant qui a transité entre les deux.
- Compléter l'écouteur de la destination avec l'affichage des coordonnées de la souris, que l'on trouve dans l'événement. À noter que ces coordonnées sont celles du centre de la source.
- Dans l'écouteur de la destination, retrouver l'identifiant de la source dans le **DragBoard** et utiliser les coordonnées de la souris pour replacer l'activité (pour ce faire, ajouter des nouvelles fonctions dans le monde).
- En plus de l'identifiant, ajouter l'image du composant graphique dans le **DragBoard**. Cette image est utilisée par le système de gestion d'événements pour visualiser le déplacement de la source à l'écran. Cette image s'obtient par la fonction **snapshot** sur le composant.

**Attention** : il est possible que sur certaines machines, la version de JFX ne permette pas le *drag&drop*. Dans ce cas, il faut ajouter une nouvelle option à la VM (Merci à Kévin).

**-Djdk.gtk.version=2**

Et pour forcer l'utilisation du GPU pour éviter les warnings sur la sortie standard (Merci à Rémy) :

**-Dprism.forceGPU=true**

## 5- Activer la sélection d'un arc

Ajoutez la possibilité de sélectionner un arc. Un clic sur l'item de menu *Supprimer la sélection* supprime aussi tous les arcs sélectionnés.

## 6- Désactiver la sélection

Une fois traités les éléments d'une sélection (renommer, etc.), les éléments ne sont plus sélectionnés.

Ajoutez également l'item *Effacer la sélection* dans le **Menu Edition** permettant de désélectionner tous les éléments sélectionnés.

## 7- (option pour ceux avancent vite et bien) Une autre façon de dessiner un arc

Utilisez le mécanisme du **drag'n drop** entre points de contrôles pour permettre à l'utilisateur de tracer un arc entre deux points de contrôle.