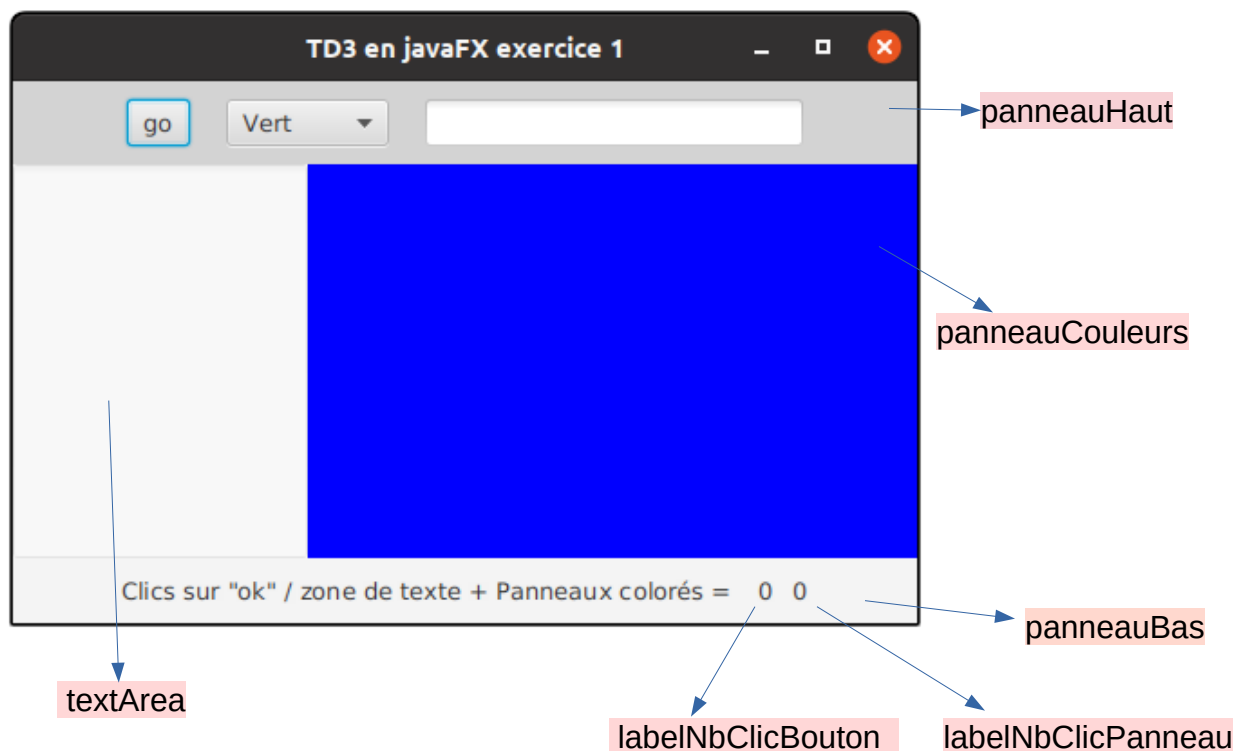


TD3: Gestion des événements

Nous allons dans cet exercice mettre en place des gestionnaires d'événements

Le code de la vue de l'application vous est fourni dans le fichier *Appli.kt*. Pour chaque question, il faudra développer des écouteurs dans des fichiers séparés (stockés dans le répertoire *ecouteurs* et ensuite abonner le nœud concerné de la vue (objet graphique ou conteneur) à cet écouteur.



1) Lorsque l'utilisateur clique sur le bouton *go*, il faut que le label nommé *labelNbClicBouton* reflète le nombre de clic déjà réalisé sur ce bouton.

Vous développerez l'écouteur dans le fichier *BoutonGoEcouteur.kt*. Il est déjà partiellement écrit.

Explications:

La classe *BoutonGoEcouteur* implémente l'interface *EventHandler<ActionEvent>*

Pour connaître le type de l'événement que devra gérer l'écouteur, il suffit de regarder dans le tableau à la fin du cours où sont répertoriés, les actions utilisateurs, les composants cibles et les événements associés.

Le clic sur un bouton génère un événement de type *ActionEvent*

La méthode *handle(...)* de cette classe, que vous devez redéfinir, est la méthode déclenchée lorsque l'événement a lieu.

Vous remarquerez que la classe *BoutonGoEcouteur* a un constructeur qui a comme paramètre, **la vue de l'application**. Dans cette vue, certains des nœuds de la scène sont en attributs. Suivant la visibilité autorisée, on pourra donc dans l'écouteur, via la vue, manipuler ses attributs (changer la valeur de la propriété d'un nœud liée à la couleur, le contenu d'un label ...)

Une fois la méthode *handle()* redéfinie, il ne reste plus qu'à associer l'écouteur au nœud concerné dans la vue.

A vous de jouer ! Développez le premier écouteur

2) Lorsque l'utilisateur choisit une couleur dans la **ComboBox** *choixCouleur*, le panneau *panneauCouleurs* se colore de la couleur choisie. Il faudra utiliser l'attribut *selectionModel* de la *comboBox* pour pouvoir accéder à l'item sélectionné

Ecouteur => ChoixCouleurEcouteur

3) Lorsque l'utilisateur clique sur le panneau *panneauCouleurs*, il faut que le label nommé *labelNbClicPanneau* reflète le nombre de clic déjà réalisé sur ce panneau

Ecouteur => ChoixPanneauEcouteur

4) On veut maintenant que le texte saisi dans le **TextField** *zoneTexte* soit recopié à la volée dans le **TextArea** *textarea*.

Écouteur : RecopieurTexteEcouteur

5) Lorsqu'on clique 2 fois sur le champs texte *zoneTexte* alors son contenu s'efface

Ecouteur : EffaceurTexteEcouteur

On voit ici un nœud peut être abonné à plusieurs écouteurs

6) On veut maintenant, que si dans le **TextField** *zoneTexte* un caractère '\$' est saisi, alors une fenêtre de dialogue de type ALERT s'ouvre en stipulant le problème et le caractère est effacé de la zone de saisie

7) On veut maintenant que lorsque le pointeur de souris entre dans *panneauHaut*, le panneau se colore en rose et lorsqu'il en sort, le panneau reprend sa couleur initiale.

Pour ceci, il faut regarder la classe *MouseEvent*

<https://docs.oracle.com/javase/8/javafx/api/javafx/scene/input/MouseEvent.html>

On peut accéder au " type " de l'événement *MouseEvent* via la méthode *getEventType()*. Elle renvoie un des types d'événement qui sont des variables "statiques" de la classe.

Par exemple : *MouseEvent.MOUSE_PRESSED*

Ecouteur : PanneauHautEcouteur

8) utilisation de lambda expression

Dans la vue, désabonnez le bouton *go* de son écouteur. Maintenant, obtenez le même comportement en utilisant une lambda expression (voir cours).

9) On veut maintenant que lorsque l'utilisateur appuie sur la touche "ctrl" de son clavier le *panneauBas* se colorie en rose. Lorsqu'il relâche la touche, le *panneauBas* doit redevenir blanc. Définir où placer l'écouteur, quel type d'événement intercepter et les traitements à effectuer. Implémentez ceci. Que constatez vous?

Ecouteur : `EcouteurToucheControl`

Explication:

Certains composants ont des gestionnaires d'événement intégrés qui les font réagir à la relâche des touches du clavier. Vous pouvez tester. Par exemple, si un bouton a le focus, la relâche de la touche "entrée" (après son appui bien sûr) l'actionne. Si une `ComboBox` a le focus, l'appui/relâche sur les touches "pageUp" et "pageDown" permet de modifier l'item sélectionné...

Dans notre cas, quand on appuie sur la touche "ctrl", l'événement est bien capturé par l'écouteur `EcouteurToucheControl` associé à la scène, par contre quand on relâche la touche, un composant de la vue le capture avant qu'il ne puisse remonter vers la scène (regarder le cours sur les 2 phases de propagation des événements)

Pour pallier à ceci, chercher dans le cours le moyen pour que la scène capture l'événement avant tous les autres composants. Testez le.