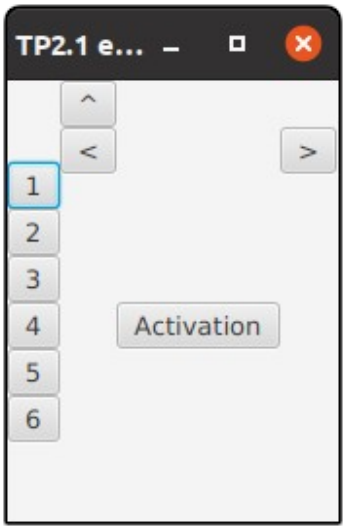
TD2 de développement d'applications avec IHM

Le but de ce TD est d'apprendre à réaliser une IHM en utilisant des composants graphiques (controls) et des conteneurs et d'utiliser des feuilles de style

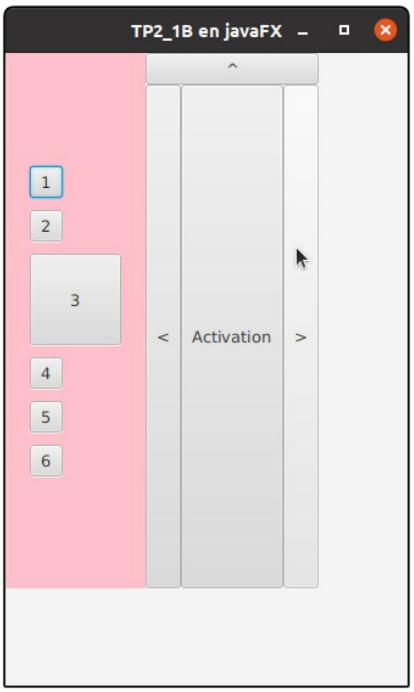
I) Détermination des composants graphiques et des conteneurs

1) Soit l'interface suivante dont deux vues vous sont présentées

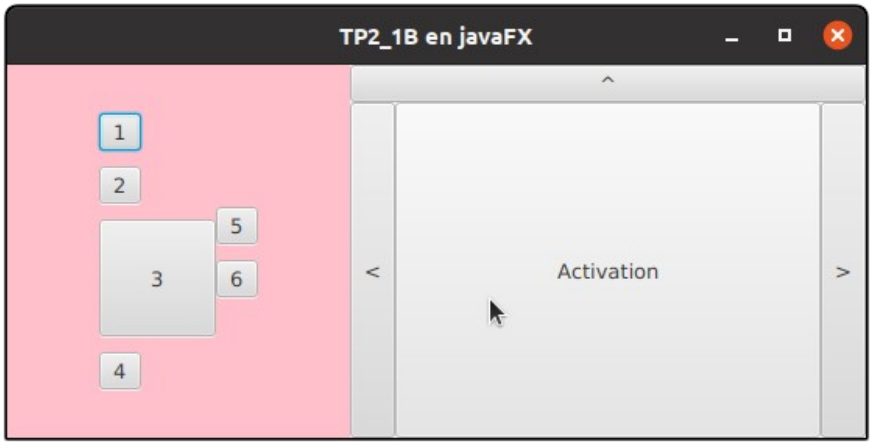


a) Dessiner sur une feuille de papier l'arbre de la scène de cette interface et donner si besoin les valeurs utilisées pour certaines propriétés de nœuds.

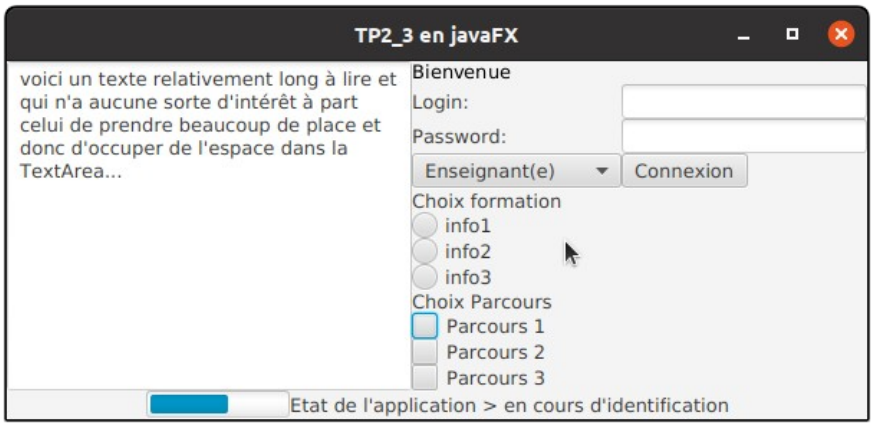
b) Déterminer les propriétés des nœuds qu'il faut modifier pour parvenir au visuel et au comportement suivant:

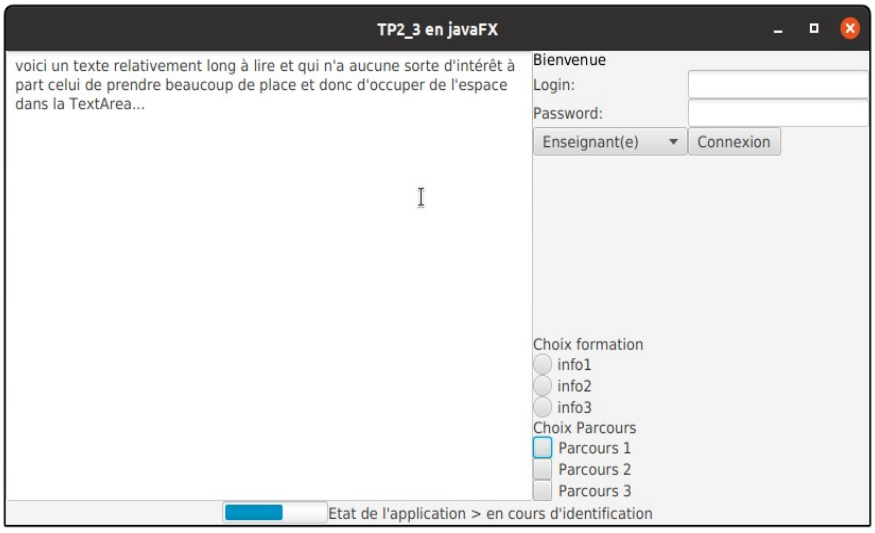


c) Comment obtenir maintenant ce comportement ?



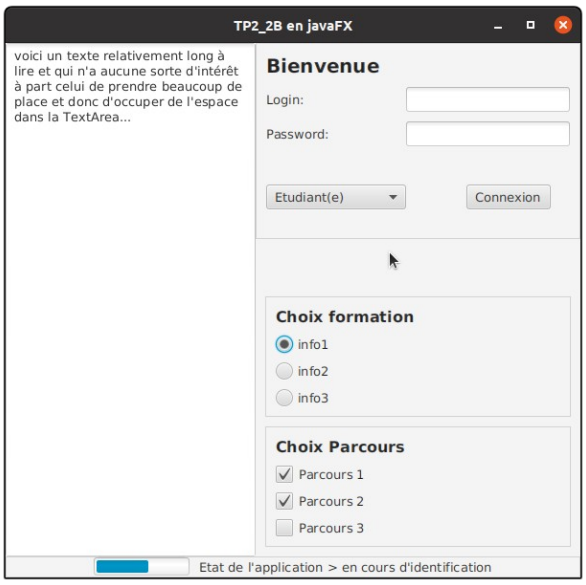
2) Soit l'interface suivante dont deux vues différentes vous sont présentées





a) Dessiner sur une feuille de papier l'arbre de la scène de cette interface et donner si besoin les valeurs utilisées pour certaines propriétés des nœuds.

b) Déterminer les modifications à apporter pour parvenir au visuel suivant (les propriétés des nœuds à manipuler et leurs valeurs)



II) Implémentation en javaFX

Lien git du projet IntelliJ:
https://gitlab.univ-nantes.fr/iut.info1.dev.ihm/dev.ihm.td2

Vous pouvez utiliser la documentation Oracle :
https://docs.oracle.com/javase/8/javafx/api/toc.htm

1) Développer les deux interfaces étudiées précédemment

2) Une fois la deuxième interface développée, nous allons maintenant utiliser des **feuilles de style** pour mettre en place son design plutôt que de modifier directement dans le code la valeur de certaines propriétés. Il faut d'abord que vous recopiez dans un autre fichier (TD2_2CssFx.kt) le code contenu dans *TD2_2Fx.kt* et vous commentez les lignes où vous avez valué des propriétés pour le design final. Nous allons maintenant utiliser des styles définies dans une feuille de style
Nous utilisons l'outil Maven, il faut stocker la feuille de style dans le dossier *src/main/ressources*. Vous en disposez d'une dans **ihm.td2.css** qui se nomme **style.css** que vous allez modifier.
Dans le code Kotlin pour y accéder ensuite, il faut charger la feuille de style avec l'instruction suivante :

```
scene.stylesheets.add(TP2_2CssFx::class.java.getResource("css/style.css").toExternalForm())
```

Si on veut appliquer un style sur un nœud, il suffit d'écrire :
`noeud.styleClass.add("my_style")` si *my_style* est une classe définie dans *style.css*

Documentation javaFX et CSS:
https://docs.oracle.com/javase/8/javafx/api/javafx/scene/doc-files/cssref.html

Un tutoriel : https://docs.oracle.com/javafx/2/css_tutorial/jfxpub-css_tutorial.htm

III) Implémentation en JavaFx d'une nouvelle interface

Vous commencerez par déterminer à l'écrit les différents composants graphiques et conteneurs utilisés et ensuite vous développerez l'interface en javaFX.

Un nouveau type de conteneur est utilisé ici : **TitledPane.** Il est constitué d'une zone avec un titre (par exemple : Consultation des livres) et on peut aussi lui associer un conteneur qui se placera en dessous du titre (dans l'exemple précédent : un conteneur qui contient les deux boutons pour se déplacer dans la liste de livres et la zone d'affichage.