



# Interfaces Graphiques

## TP1 - Se familiariser avec les composants de base de JavaFX

### Environnement de développement

Dans **IntelliJ**, créez un nouveau projet **Java** de nom **testIG**.  
Poussez régulièrement le code sur **gitlab** dans le projet créé à cet effet.

En Java 11, **JavaFX** n'est pas intégré dans la distribution. Il faut l'ajouter "à la main" dans les librairies du projet (**File/ProjectStructure/Libraries**) :

- si vous travaillez sur les machines atelis, vous le trouverez dans le répertoire dans **/usr/share/openjfx**
- si vous travaillez sur votre ordinateur personnel, il faut le télécharger à l'adresse <https://gluonhq.com/products/javafx/> ; l'archive doit être placée là où vous voulez dans votre arborescence (mais surtout pas directement dans le projet IntelliJ).

Il faut ensuite sélectionner le répertoire **lib** pour **File/ProjectStructure/Libraries**.

Pour que l'exécution se passe sans problème, il faut également préciser quels modules de **JavaFX** sont utilisés par la machine java virtuelle. Dans la configuration d'exécution du projet, ajouter les paramètres suivants à la machine virtuelle (VM) :

```
--module-path /usr/share/openjfx/lib/ (à adapter si vous travaillez sur votre ordinateur personnel)  
--add-modules javafx.controls, javafx.fxml
```

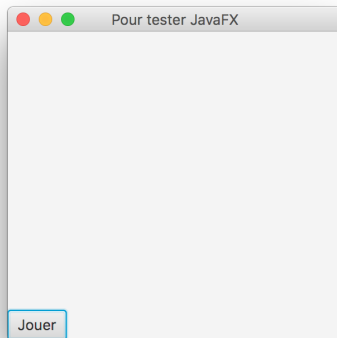
### La classe Main

Dans ce premier TP, nous allons utiliser **JavaFX** en mode procédural. Ecrivez la classe principale **Main**, en vous inspirant d'une version simple du cours, avec un titre de fenêtre et un **BorderPane** vide.

Assurez-vous que l'exécution se passe sans problème.

## Fonctionnalités à réaliser

Sans (trop) vous soucier de la structuration en classes, complétez la fonction **start** pour obtenir les fonctionnalités listées ci-dessous.



*Attention : Chaque question doit être complètement résolue avant de commencer la suivante. Ne cherchez pas à en faire plus que demandé ...*

1. Ajoutez un bouton (**Button**) en bas, étiqueté **Jouer**. Attention de ne pas vous tromper dans l'importation des classes : on utilise le package **javafx**.
2. Un clic sur le bouton provoque l'affichage d'un message de bienvenue sur la sortie standard.
3. Un clic sur le bouton provoque aussi l'affichage (toujours sur la sortie standard) du nombre de fois où le bouton a été cliqué.
4. La fenêtre principale affiche au centre un panneau **HBox** contenant trois labels (**Label**) étiquetés avec la valeur **1**. À noter que, comme il n'y a pas de composant placé dans la partie gauche du **BorderPane**, le panneau ne s'affiche pas au centre .... On y remédiera plus tard.
5. Un clic sur le bouton provoque, en plus du reste, le changement de la valeur affichée sur chaque label. Ces valeurs sont indépendantes et prises au hasard entre 1 et 6 inclus (on simule ainsi le lancé de 3 dés).
6. La fenêtre principale affiche un nouveau panneau en haut, indiquant le nombre de coups joués et le nombre de fois où la combinaison des trois nombres est 421.
7. La fenêtre principale affiche à gauche un nouveau bouton permettant de quitter le jeu (faire un appel à **Platform.exit()**). Placez une image **png** sur ce bouton, plutôt qu'un texte ; cette image est rangée dans le répertoire **src/ressources**.
8. Remplacer les nombres affichés sur les labels par des images des faces des dés montrant le nombre correspondant.
9. Les 3 labels (les faces des dés) sont à présent remplacés par des boutons. Avant de cliquer sur le bouton **Jouer**, le joueur sélectionne (par clic) les nombres qu'il veut conserver.
10. Utilisez les fonctionnalités des composants/panneaux pour améliorer le look de votre application.

Ce TP est à déposer sur Arche, sous forme d'une archive exécutable, ne contenant pas les sources. Pour vous aider dans la création d'une telle archive, retrouvez les documents sur arche, dans le cours de BPO. L'exécution de l'archive nécessite les mêmes arguments que ceux cités dans le cadre au-dessus.