

JAVA

Version 1.0.0

Niveau requis : 3/7



Première application JavaFX

version 17

*(OpenJDK 11 de Adoptium avec usage de
l'IDE IntelliJ Community et SceneBuilder)*

Sommaire

I.	PREAMBULE.....	3
I.I.	OBJET.....	3
I.II.	PRE-REQUIS	3
I.III.	VERSIONS DU DOCUMENT	3
I.IV.	DOCUMENTS DE REFERENCE	3
II.	JDK ET IDE.....	3
II.I.	JDK 11 x64 (ADOPTIUM).....	3
II.II.	INTELLIJ COMMUNITY DE JETBRAINS – IDE JAVA	5
III.	CREATION DE LA PREMIERE APPLICATION FRISTJAVAFX.....	7
III.I.	CREATION DU PROJET	7
III.II.	LANCEMENT DU PROJET DEMO	9
III.III.	LANCEMENT PAR MAVEN	10
IV.	STRUCTURATION DU PROJET.....	13
IV.I.	UTILISATION DES MODULES	13
IV.II.	REPARTITION VUE/CONTROLEUR.....	13
IV.II.1	<i>La vue hello-view.fxml</i>	13
IV.II.2	<i>Le contrôler</i>	14
V.	SIMPLIFICATION DE L'EDITION DE LA VUE AVEC LE PLUGIN SCENEUILDER	15
V.I.	SCENE BUILDER PAR INSTALLEUR MSI	15
V.II.	CONFIGURER INTELLIJ	20
V.III.	DANS LA VUE CLIQUER SUR DOWNLOAD JAVAFX	22
V.IV.	EDITION EN MODE GRAPHIQUE.....	22
VI.	CREATION DU LIVRABLE	23
VI.I.	CONFIGURATION DU FICHIER POM.XML	23
VI.II.	GENERATION DE LA CONSTRUCTION DU LIVRABLE	24
VII.	FIN DU DOCUMENT	26

I. Préambule

I.I. *Objet*

L'objet de ce document est de présenter la mise en place d'une application de type HelloWorld en client lourd JavaFX avec la version 11 d'openJDK de Adoptium et l'usage de l'IDE de JetBrains gratuit IntelliJ Community.

I.II. *Pré-requis*

Dans le cadre de ce tutorial, nous serons sur un environnement Windows 10.

I.III. *Versions du document*

Version	Date	Auteur	Description
1.0	19/12/2022	Péquignat.eu	Création du document

I.IV. *Documents de référence*

#	Document	Version	Auteur(s)
[R1]	https://www.jetbrains.com/help/idea/opening-fxml-files-in-javafx-scene-builder.html	17	jetbrains.com

II. JDK et IDE

II.I. *JDK 11 x64 (Adoptium)*

Télécharger le JDK11 correspondant à votre plateforme :

<https://adoptium.net/temurin/releases/?version=11>

Use the drop-down boxes below to filter the list of current releases.

Operating System	Architecture	Package Type	Version
Windows ▾	x64 ▾	JDK ▾	11 ▾

jdk-11.0.17+8 Temurin  October 25, 2022	Windows	x64	JDK - 174 MB	 <i>Installeur</i>
			Checksum	
			JDK - 197 MB	 <i>Archives</i>
			Checksum	

Dans le cadre de ce tutorial nous avons téléchargé et installé la version Installeur msi de windows:

```
OpenJDK11U-jdk_x64_windows_hotspot_11.0.17_8.msi
```

Le chemin d'installation a été laissé par défaut et se situe sur :

```
C:\Program Files\Eclipse Adoptium\jdk-11.0.17.8-hotspot
```

II.II. *IntelliJ Community de JetBrains – IDE JAVA*

Je recommande d'utiliser la version Communautaire de IntelliJ qui permet de créer des applications Java, même commerciale.

<https://www.jetbrains.com/fr-fr/idea/download/#section=windows>

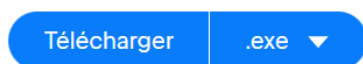
La version actuelle est la version : 2022.3

Télécharger IntelliJ IDEA

Windows macOS Linux

Ultimate

L'IDE leader pour Java et Kotlin



Essai gratuit de 30 jours disponible

Community Edition

L'IDE pour le développement Java et Kotlin pur



Gratuit, conçu à partir de code open source

IntelliJ IDEA
Ultimate

IntelliJ IDEA Community Edition ⓘ

Une fois téléchargé, lancer le programme et accepter le lancement.

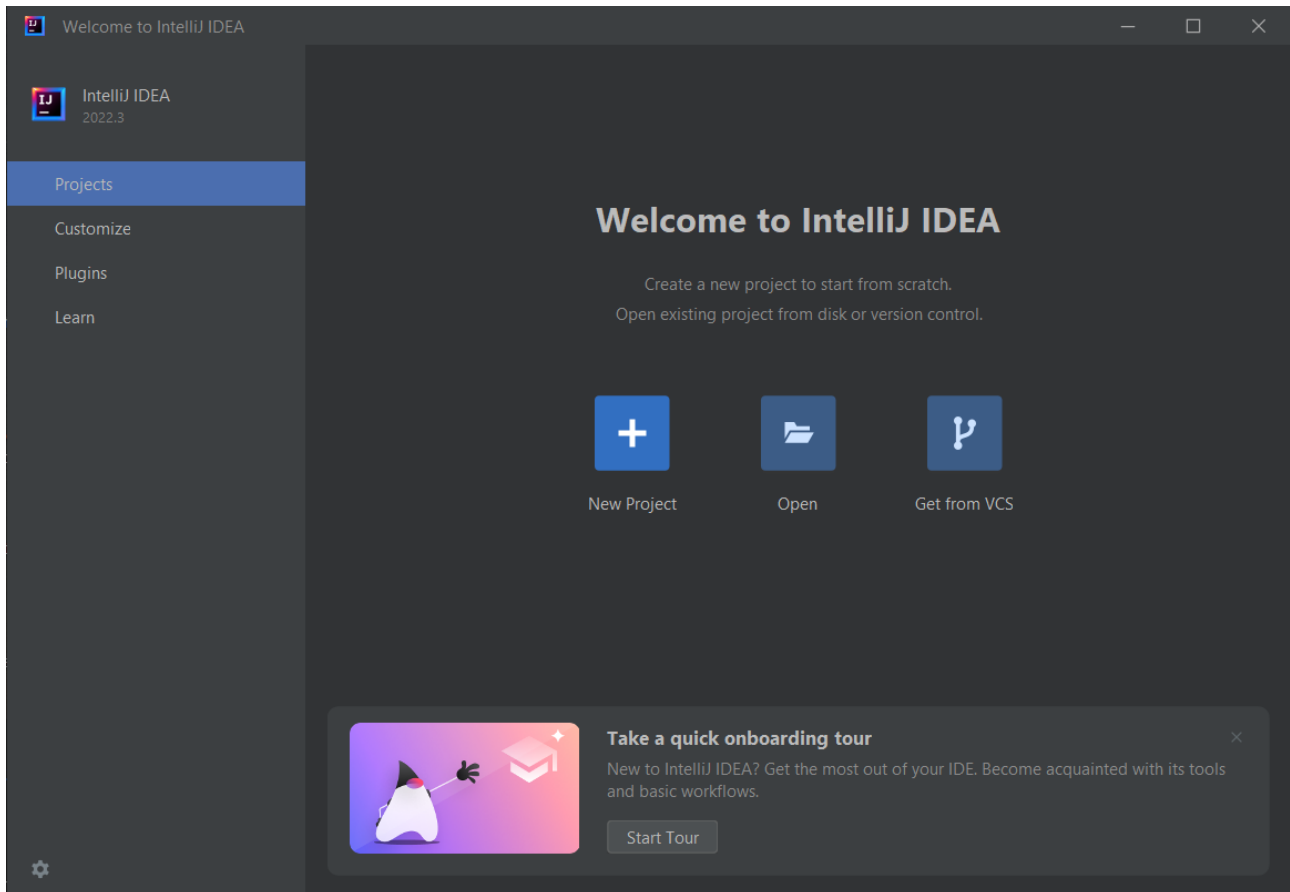
Si vous aviez déjà une version précédente de IntelliJ, Désinstallez là. Puis pour l'installation, cliquer sur le choix de ne pas importer la configuration précédente afin d'avoir une version blanche d'installation.

Laisser tout par défaut, c'est très suffisant.

Créer le raccourci sur le Bureau en cochant la case lors de la proposition.

Une fois installé IntelliJ Community se situe dans le répertoire : "C:\Program Files\JetBrains\IntelliJ IDEA Community Edition 2022.3\"

Double cliquer sur le raccourci :

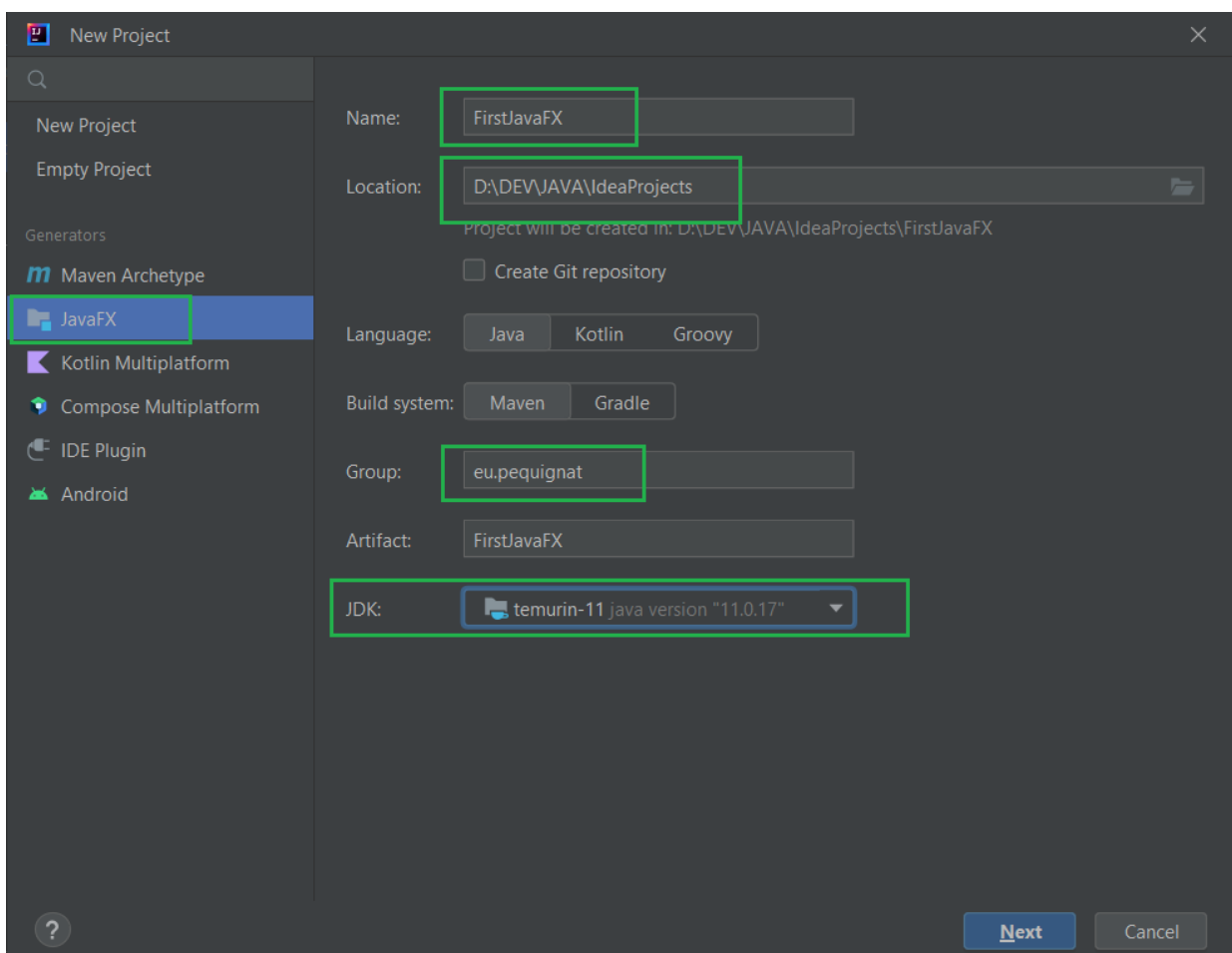
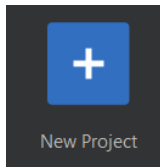


Vous pouvez faire une formation rapide avec le « Take a quick emboarding tour » pour voir des fonctions de base et des possibilités de refactoring.

III. Création de la première application FirstJavaFX

III.I. Création du projet

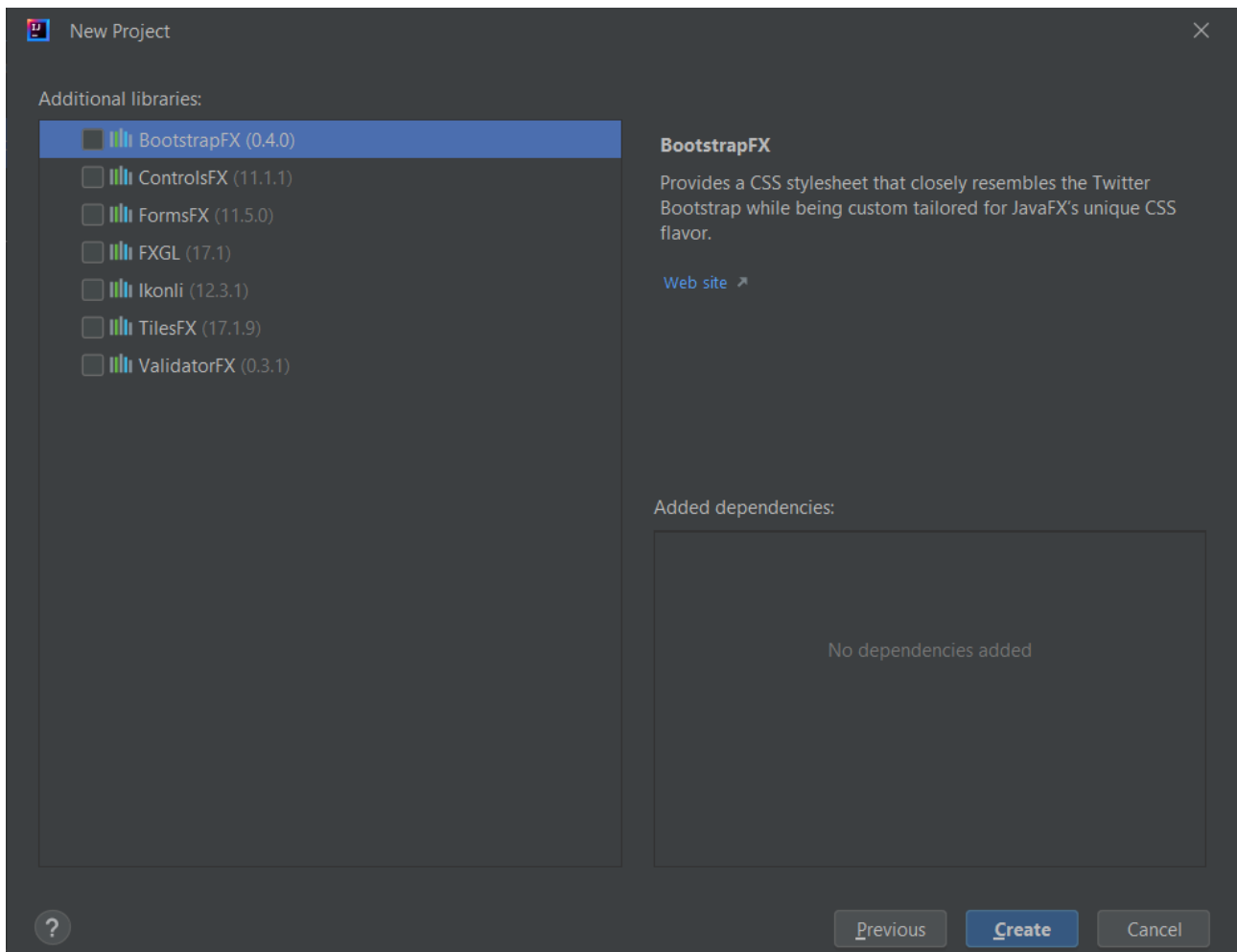
Cliquer dans IntelliJ Community sur « New Project »



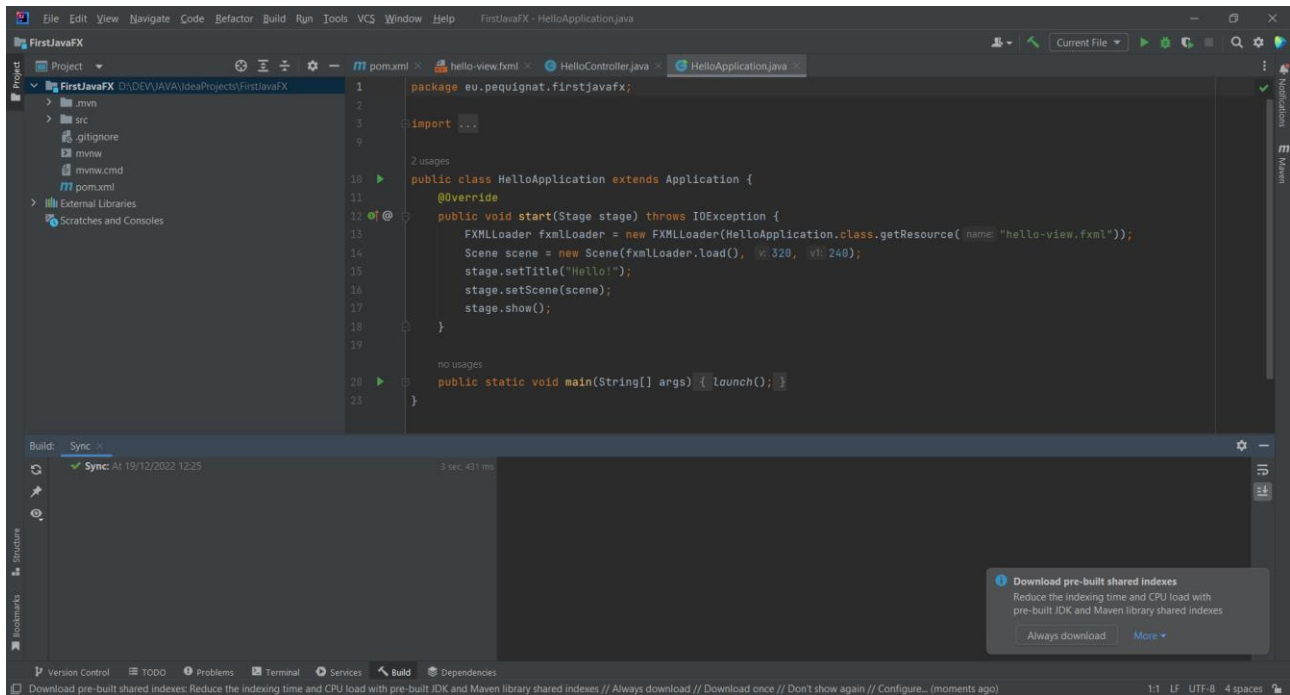
Dans la partie JDK, vous devrez cliquer sur « Add JDK... » et sélectionner le répertoire :

```
C:\Program Files\Eclipse Adoptium\jdk-11.0.17.8-hotspot
```

Cliquer sur Next



Cliquer sur « Create »

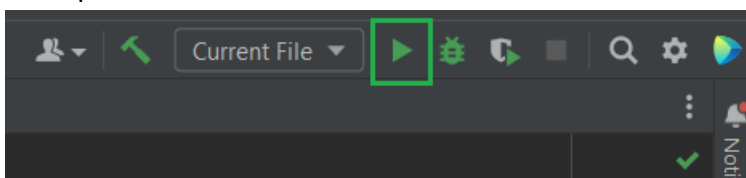


III.II. Lancement du projet demo

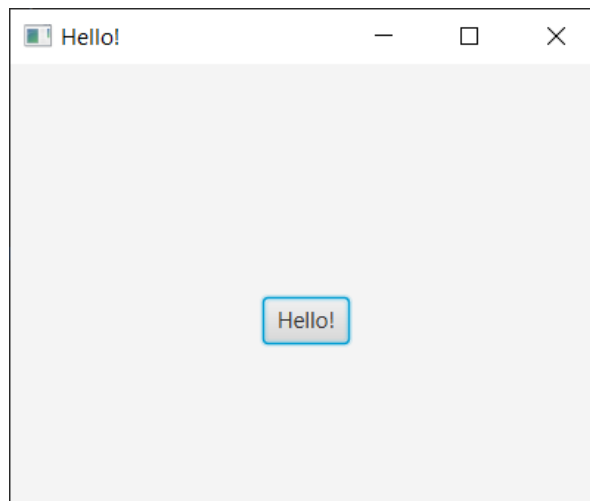
Attendre qu'automatiquement les dépendances maven soient récupérées.

Une fois vers, ouvrez le fichier « HelloApplication » présent dans le package « eu.pequignat.firstjavafx ».

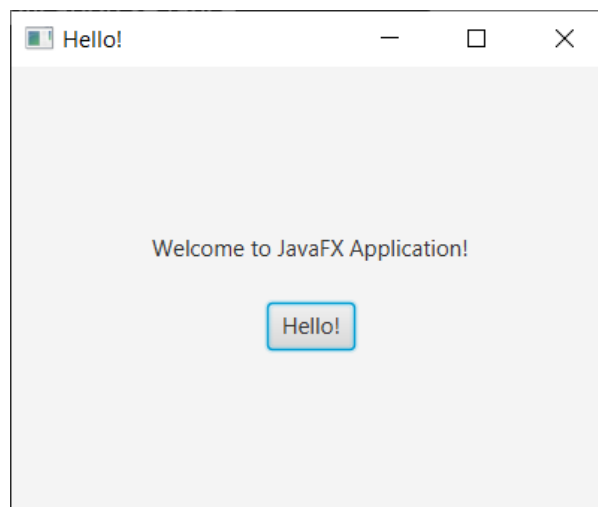
Et cliquer en haut à droite sur la flèche verte de l'IDE :



L'application se lance :



En cliquant sur le bouton « Hello ! »



III.III. *Lancement par maven*

Pour lancer par maven le projet javafx :

```
mvn clean javafx:run
```

Les traces dans la console de maven :

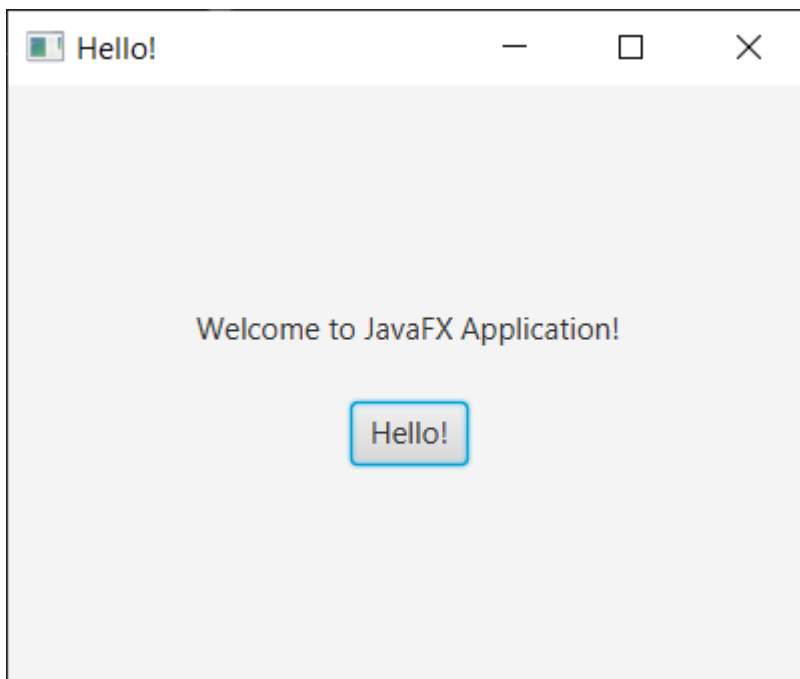
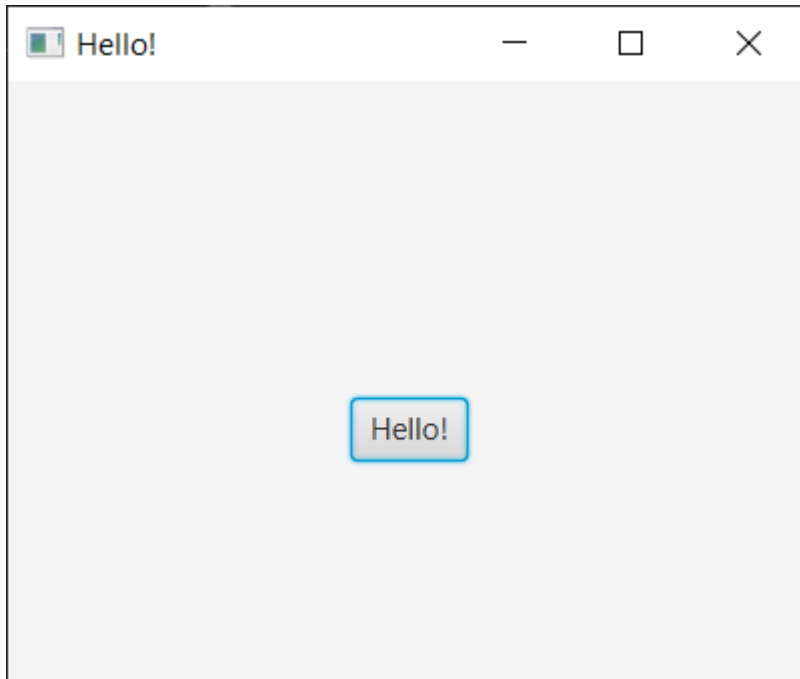
```
"C:\Program Files\Eclipse Adoptium\jdk-11.0.17.8-hotspot\bin\java.exe" -
Dmaven.multiModuleProjectDirectory=D:\DEV\JAVA\IdeaProjects\FirstJavaFX -
Djansi.passthrough=true "-Dmaven.home=C:\Program Files\JetBrains\IntelliJ IDEA
Community Edition 2022.3\plugins\maven\lib\maven3" "-
Dclassworlds.conf=C:\Program Files\JetBrains\IntelliJ IDEA Community Edition
2022.3\plugins\maven\lib\maven3\bin\m2.conf" "-Dmaven.ext.class.path=C:\Program
Files\JetBrains\IntelliJ IDEA Community Edition 2022.3\plugins\maven\lib\maven-
event-listener.jar" "-javaagent:C:\Program Files\JetBrains\IntelliJ IDEA
Community Edition 2022.3\lib\idea_rt.jar=50609:C:\Program
Files\JetBrains\IntelliJ IDEA Community Edition 2022.3\bin" -Dfile.encoding=UTF-
8 -classpath "C:\Program Files\JetBrains\IntelliJ IDEA Community Edition
2022.3\plugins\maven\lib\maven3\boot\plexus-classworlds-2.6.0.jar;C:\Program
Files\JetBrains\IntelliJ IDEA Community Edition
2022.3\plugins\maven\lib\maven3\boot\plexus-classworlds.license"
org.codehaus.classworlds.Launcher -Didea.version=2022.3 clean javafx:run
[INFO] Scanning for projects...
[INFO]
[INFO] -----< eu.pequignat:FirstJavaFX >-----
[INFO] Building FirstJavaFX 1.0-SNAPSHOT
[INFO] -----[ jar ]-----
[INFO]
[INFO] --- maven-clean-plugin:2.5:clean (default-clean) @ FirstJavaFX ---
[INFO] Deleting D:\DEV\JAVA\IdeaProjects\FirstJavaFX\target
[INFO]
[INFO] >>> javafx-maven-plugin:0.0.8:run (default-cli) > process-classes @
FirstJavaFX >>>
[INFO]
[INFO] --- maven-resources-plugin:2.6:resources (default-resources) @
FirstJavaFX ---
[INFO] Using 'UTF-8' encoding to copy filtered resources.
[INFO] Copying 1 resource
[INFO]
[INFO] --- maven-compiler-plugin:3.10.1:compile (default-compile) @ FirstJavaFX
---
[INFO] Changes detected - recompiling the module!
[INFO] Compiling 3 source files to
D:\DEV\JAVA\IdeaProjects\FirstJavaFX\target\classes
[INFO]
```

```
[INFO] <<< javafx-maven-plugin:0.0.8:run (default-cli) < process-classes @  
FirstJavaFX <<<
```

```
[INFO]
```

```
[INFO]
```

```
[INFO] --- javafx-maven-plugin:0.0.8:run (default-cli) @ FirstJavaFX ---
```

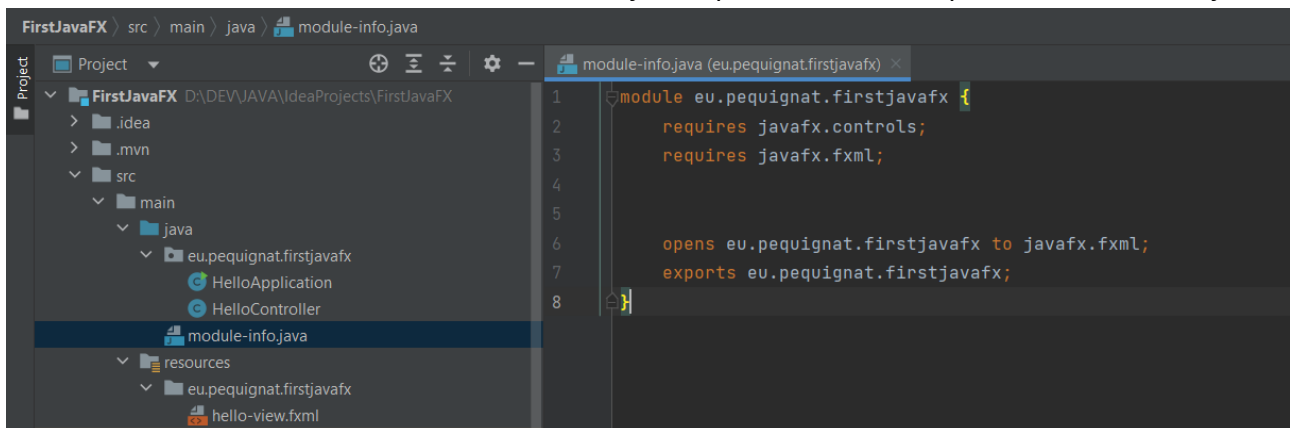


IV. Structuration du projet

IV.I. Utilisation des modules

Ce projet exploite l'usage des modules au sens Java.

Voir la déclaration dans le fichier « module-info.java » présent dans le répertoire « src/main/java »



IV.II. Répartition Vue/Contrôleur

IV.II.1 La vue hello-view.fxml

La vue correspondant à la répartition de la page des éléments graphique indépendant que leur comportement se retrouve dans le fichier « hello-view.fxml » présent dans le répertoire « src/main/resources/eu/pequignat/firstjavafx »

Ce fichier est au format fxml un fichier xml respectant la syntaxe de JavaFX :

```
<?xml version="1.0" encoding="UTF-8"?>

<?import javafx.geometry.Insets?>
<?import javafx.scene.control.Label?>
<?import javafx.scene.layout.VBox?>

<?import javafx.scene.control.Button?>
<VBox alignment="CENTER" spacing="20.0" xmlns:fx="http://javafx.com/fxml"
    fx:controller="eu.pequignat.firstjavafx.HelloController">
    <padding>
        <Insets bottom="20.0" left="20.0" right="20.0" top="20.0"/>
    </padding>

    <Label fx:id="welcomeText"/>
</VBox>
```

```
<Button text="Hello!" onAction="#onHelloButtonClick"/>
</VBox>
```

IV.II.2 Le contrôleur

On retrouve ici la déclaration du controller :

```
fx:controller="eu.pequignat.firstjavafx.HelloController"
```

qui permet de lier le contrôleur à cette vue.

On retrouve les composants

```
<Label fx:id="welcomeText"/>
```

Défini dans le controller « HelloController »

```
@FXML
private Label welcomeText;
```

Et son action avec la méthode :

```
<Button text="Hello!" onAction="#onHelloButtonClick"/>
```

```
@FXML
protected void onHelloButtonClick() {
    welcomeText.setText("Welcome to JavaFX Application!");
}
```

V. Simplification de l'édition de la Vue avec le plugin SceneBuilder

V.I. *Scene Builder par installateur msi*

<https://gluonhq.com/products/scene-builder/#download>

Download Scene Builder

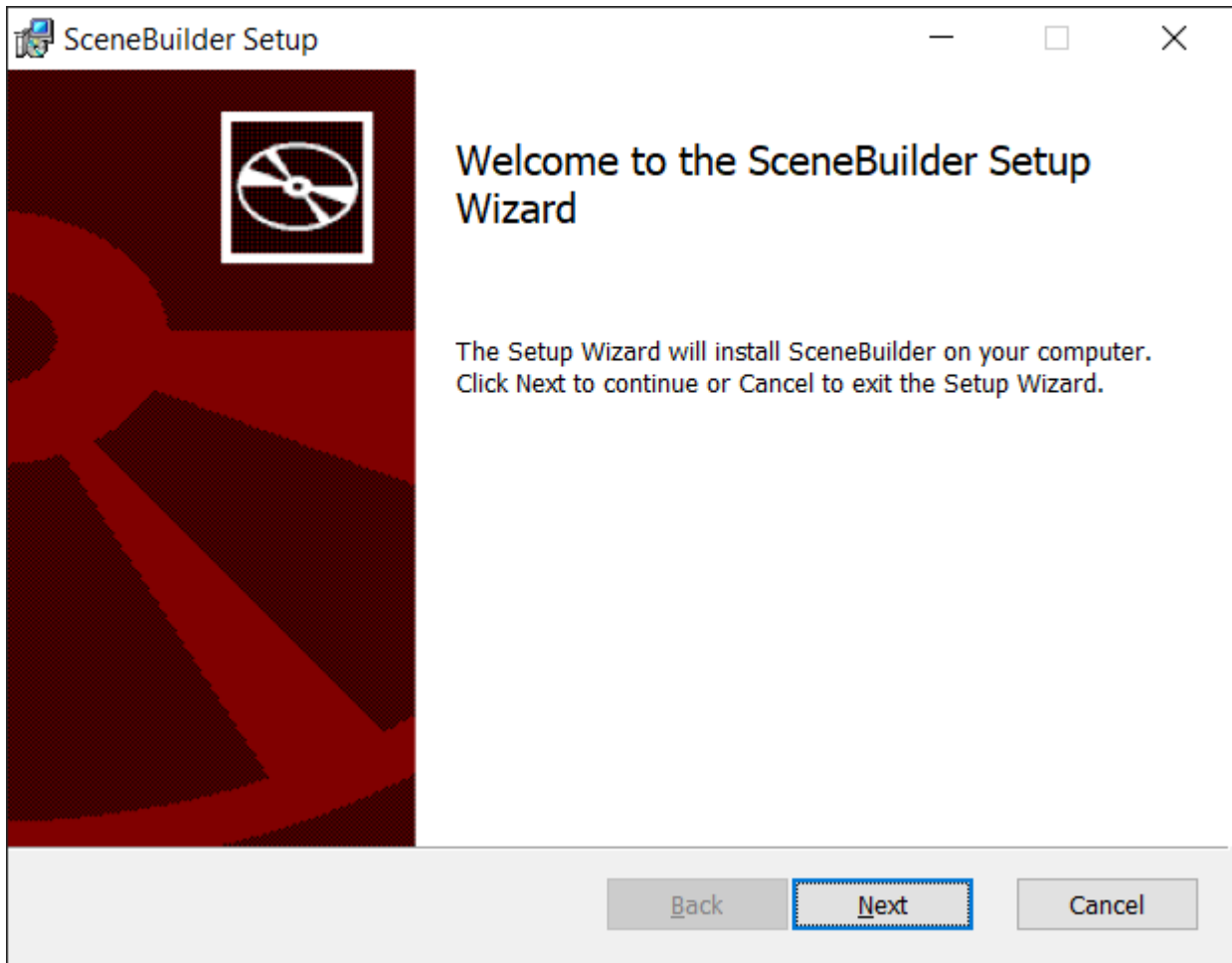
Scene Builder 19.0.0 was released on **October 7, 2022**.

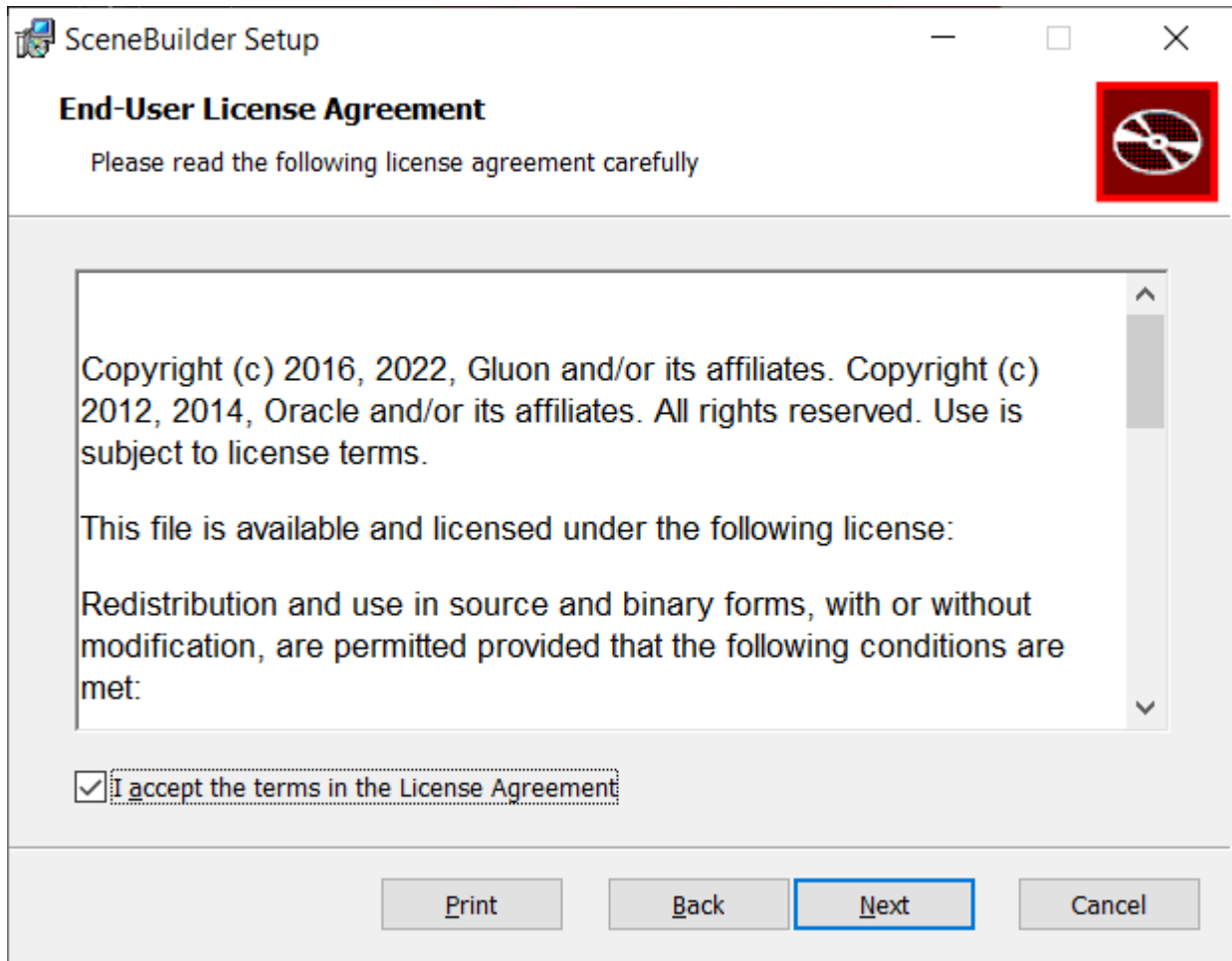
You can use this Scene Builder version together with **Java 11 and higher**.

Product	Platform	Download
Scene Builder	Windows Installer	Download
Scene Builder	Mac OS X dmg (Intel)	Download
Scene Builder	Mac OS X dmg (Apple Silicon)	Download
Scene Builder	Linux RPM	Download
Scene Builder	Linux Deb	Download
Scene Builder Kit info	Jar File	Download

Nous sommes sous Windows 10, donc on va utiliser le Windows Installer :

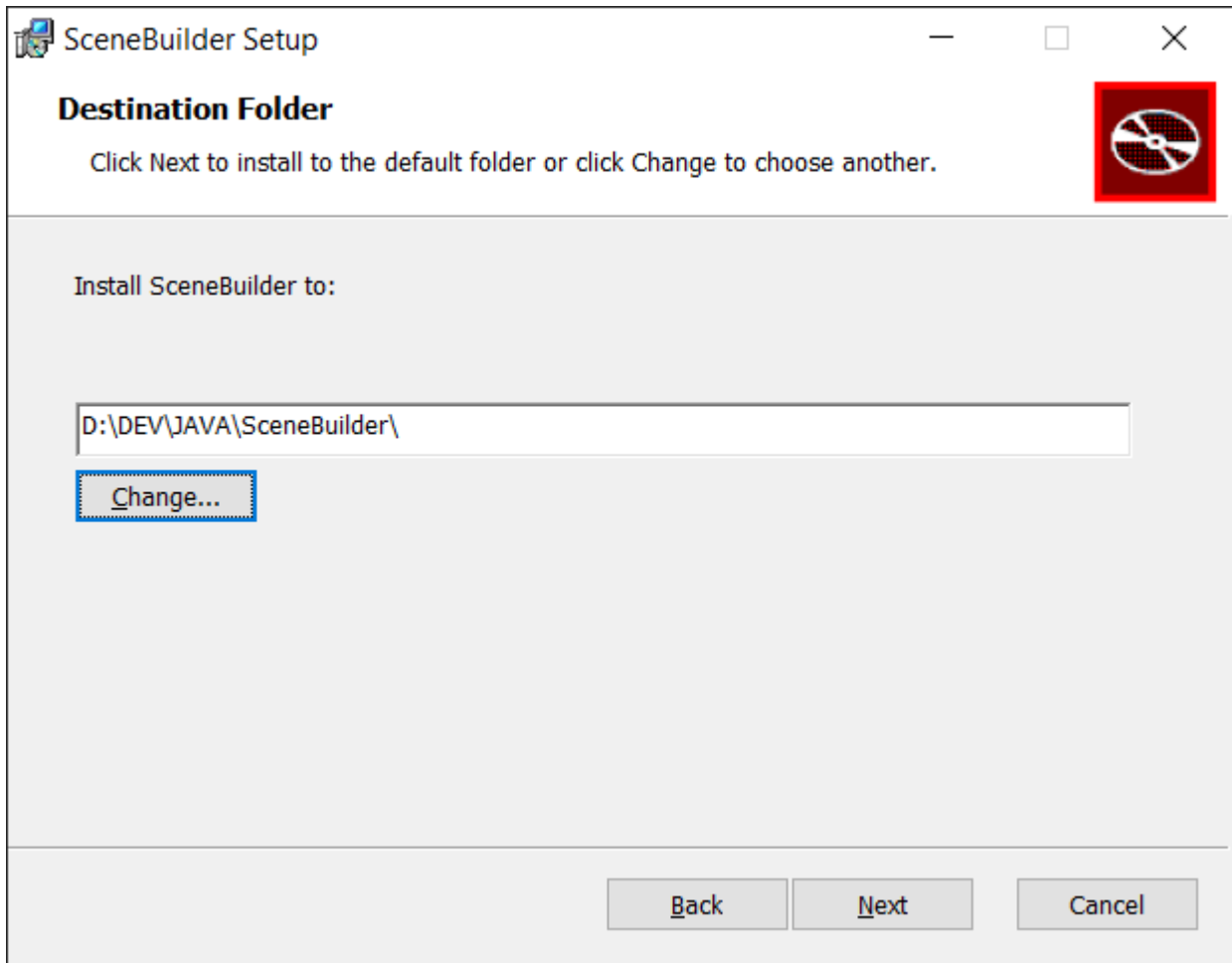
<https://gluonhq.com/products/scene-builder/thanks/?dl=https://download2.gluonhq.com/scenebuilder/19.0.0/install/win/SceneBuilder-19.0.0.msi>

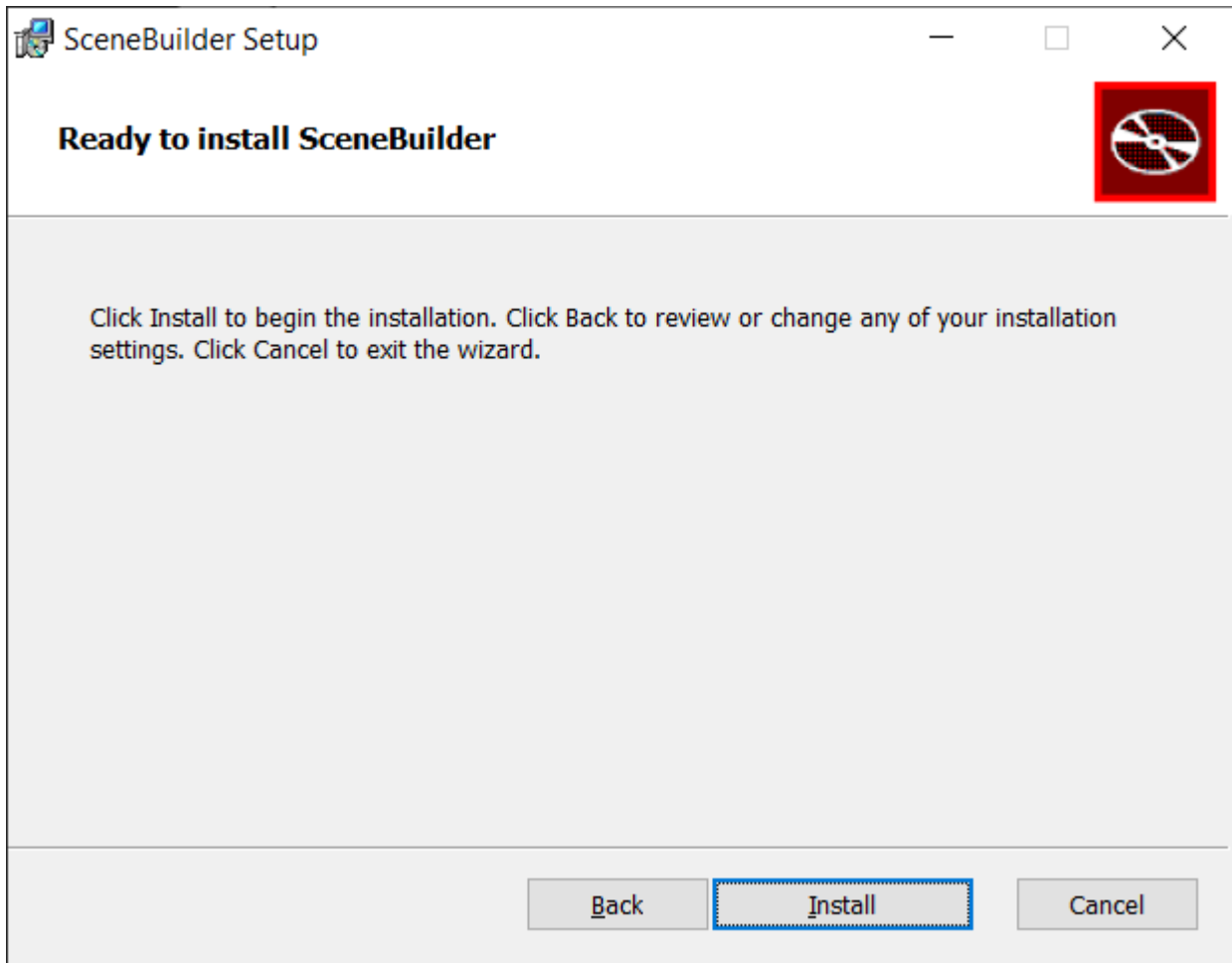


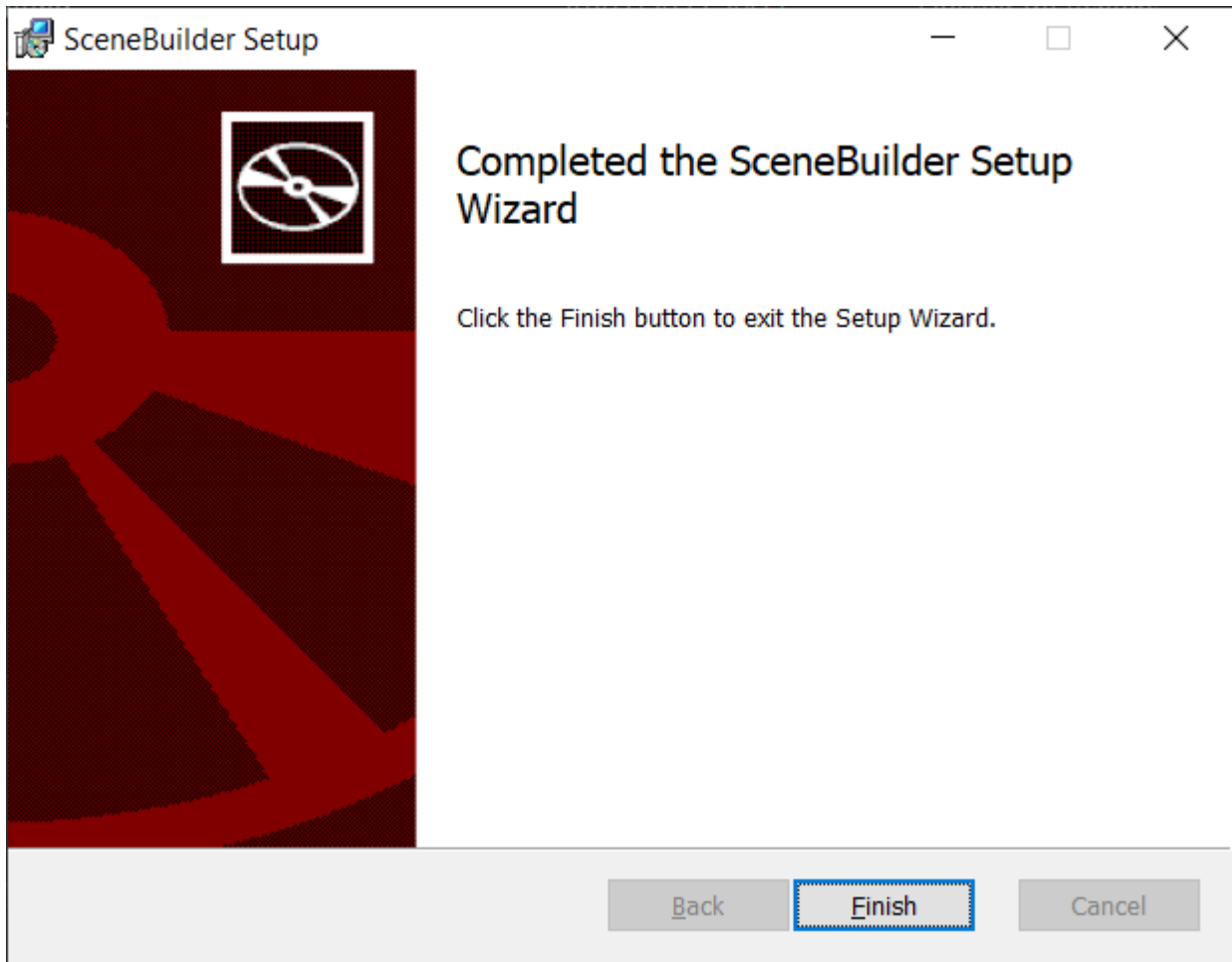


Choisissez l'endroit où vous souhaitez installer ce logiciel, exemple :

D:\DEV\JAVA\SceneBuilder\



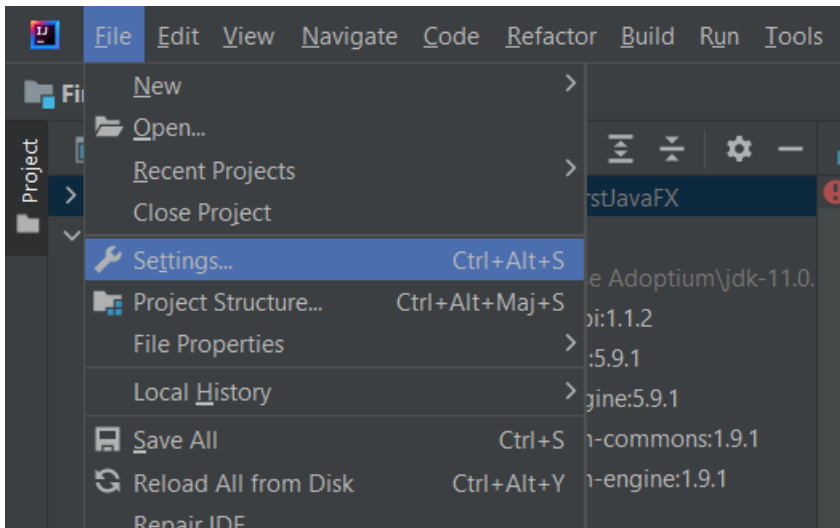




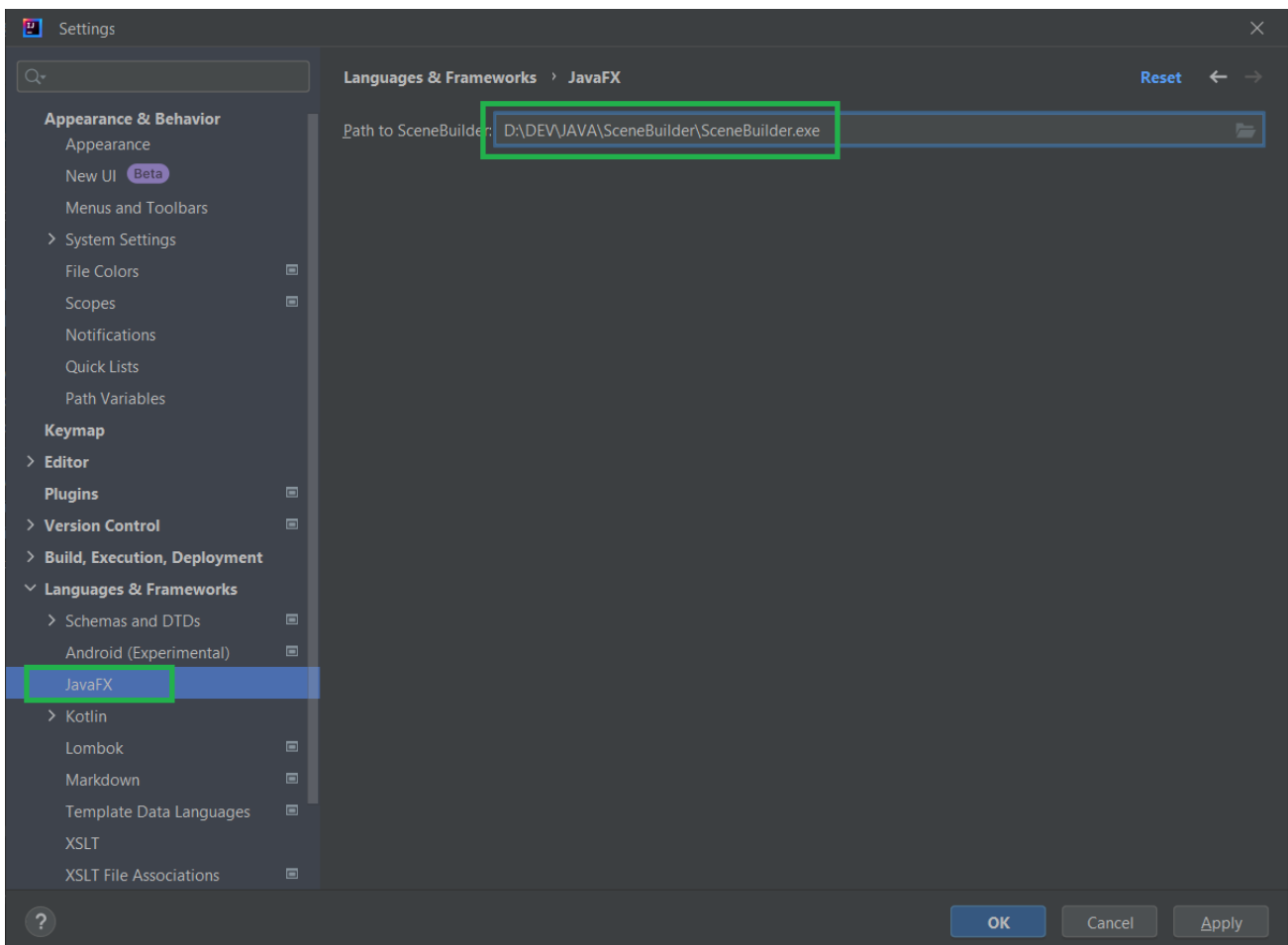
V.II. *Configurer intelliJ*

<https://www.jetbrains.com/help/idea/opening-fxml-files-in-javafx-scene-builder.html>

Aller dans les Settings de IntelliJ (File > Settings) :

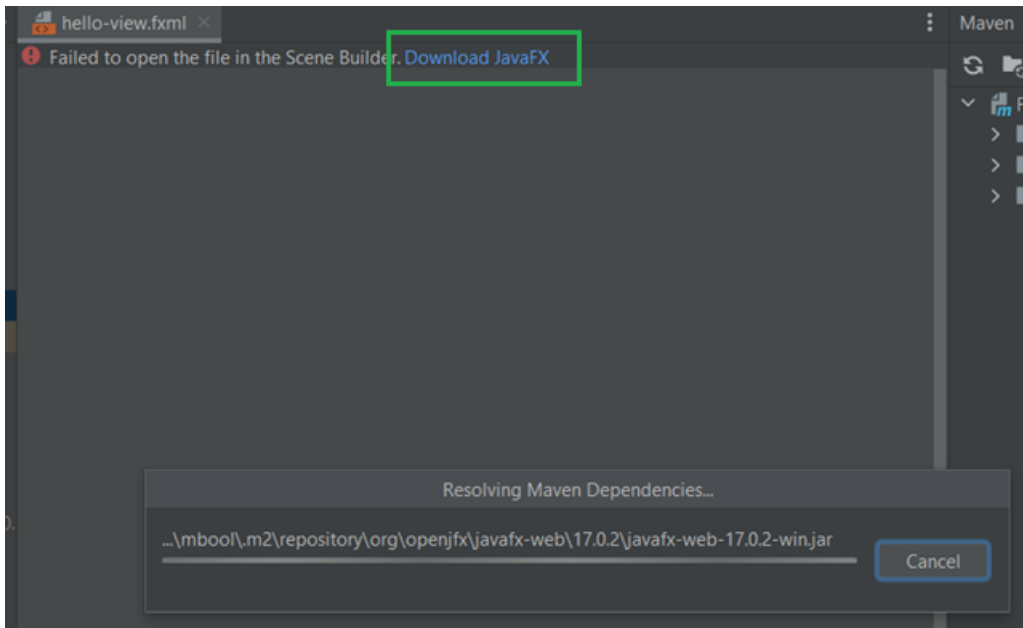


Dans le menu de Gauche, ouvrir « Languages & Frameworks », choisissez JavaFX et dans le Path to SceneBuilder mettre par exemple « D:\DEV\JAVA\SceneBuilder\SceneBuilder.exe »

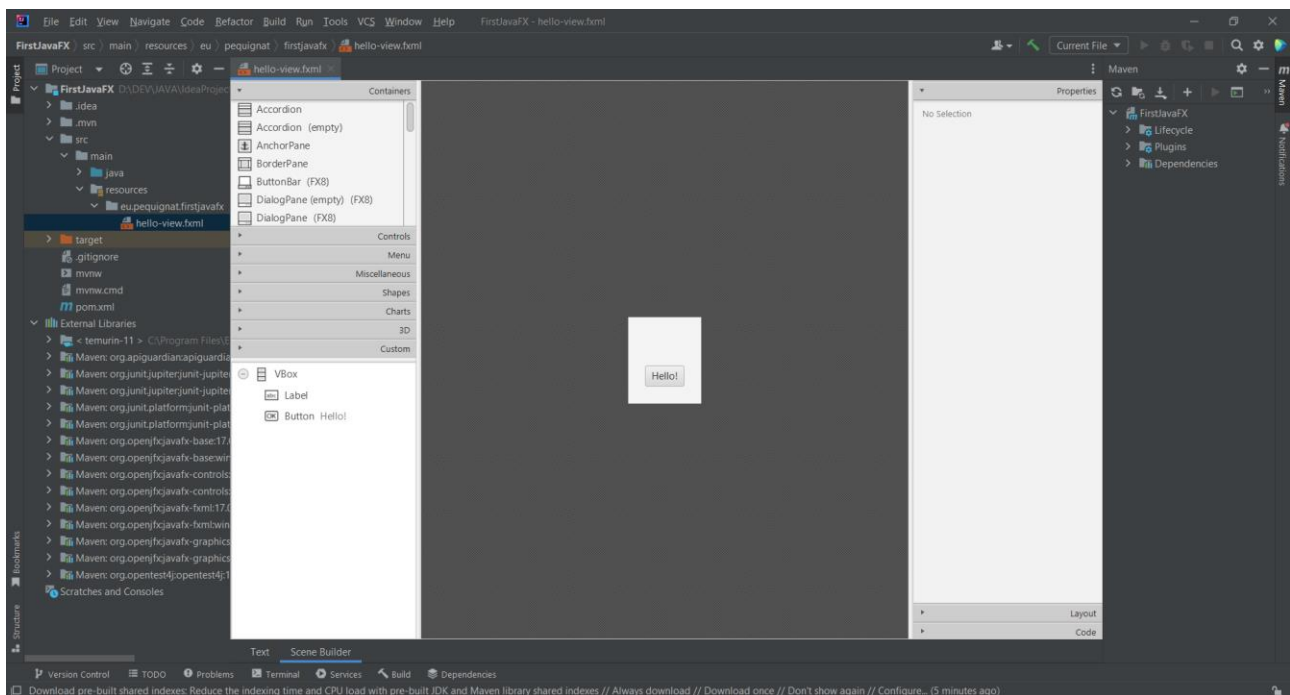


Cliquer sur « Apply » puis « OK ».

V.III. Dans la Vue cliquer sur Download JavaFX



V.IV. Edition en mode graphique



VI. Création du livrable

VI.I. Configuration du fichier pom.xml

Dans le fichier pom.xml à la racine du projet dans la partie <build>

Vous pouvez éditer les paramètres suivants :

```
<plugin>
  <groupId>org.openjfx</groupId>
  <artifactId>javafx-maven-plugin</artifactId>
  <version>0.0.8</version>
  <executions>
    <execution>
      <!-- Default configuration for running with: mvn clean javafx:run --
>
      <id>default-cli</id>
      <configuration>

<mainClass>eu.pequignat.firstjavafx/eu.pequignat.firstjavafx.HelloApplication</m
ainClass>
      <launcher>launcher</launcher>
      <jlinkZipName>FirstJavaFX</jlinkZipName>
      <jlinkImageName>FirstJavaFX</jlinkImageName>
      <noManPages>true</noManPages>
      <stripDebug>true</stripDebug>
      <noHeaderFiles>true</noHeaderFiles>
    </configuration>
  </execution>
</executions>
</plugin>
```

VI.II. Génération de la construction du livrable

La génération du livrable se fait avec tout le contexte d'exécution minimal nécessaire pour l'exécution de l'application par la ligne de commande maven :

```
mvn clean javafx:jlink
```

Dans la console maven :

```
"C:\Program Files\Eclipse Adoptium\jdk-11.0.17.8-hotspot\bin\java.exe" -
Dmaven.multiModuleProjectDirectory=D:\DEV\JAVA\IdeaProjects\FirstJavaFX -
Djansi.passthrough=true "-Dmaven.home=C:\Program Files\JetBrains\IntelliJ IDEA
Community Edition 2022.3\plugins\maven\lib\maven3" "-
Dclassworlds.conf=C:\Program Files\JetBrains\IntelliJ IDEA Community Edition
2022.3\plugins\maven\lib\maven3\bin\m2.conf" "-Dmaven.ext.class.path=C:\Program
Files\JetBrains\IntelliJ IDEA Community Edition 2022.3\plugins\maven\lib\maven-
event-listener.jar" "-javaagent:C:\Program Files\JetBrains\IntelliJ IDEA
Community Edition 2022.3\lib\idea_rt.jar=50915:C:\Program
Files\JetBrains\IntelliJ IDEA Community Edition 2022.3\bin" -Dfile.encoding=UTF-
8 -classpath "C:\Program Files\JetBrains\IntelliJ IDEA Community Edition
2022.3\plugins\maven\lib\maven3\boot\plexus-classworlds-2.6.0.jar;C:\Program
Files\JetBrains\IntelliJ IDEA Community Edition
2022.3\plugins\maven\lib\maven3\boot\plexus-classworlds.license"
org.codehaus.classworlds.Launcher -Didea.version=2022.3 clean javafx:jlink

[INFO] Scanning for projects...

[INFO]

[INFO] -----< eu.pequignat:FirstJavaFX >-----

[INFO] Building FirstJavaFX 1.0-SNAPSHOT

[INFO] -----[ jar ]-----

[INFO]

[INFO] --- maven-clean-plugin:2.5:clean (default-clean) @ FirstJavaFX ---

[INFO] Deleting D:\DEV\JAVA\IdeaProjects\FirstJavaFX\target

[INFO]

[INFO] >>> javafx-maven-plugin:0.0.8:jlink (default-cli) > process-classes @
FirstJavaFX >>>

[INFO]

[INFO] --- maven-resources-plugin:2.6:resources (default-resources) @
FirstJavaFX ---

[INFO] Using 'UTF-8' encoding to copy filtered resources.

[INFO] Copying 1 resource

[INFO]
```

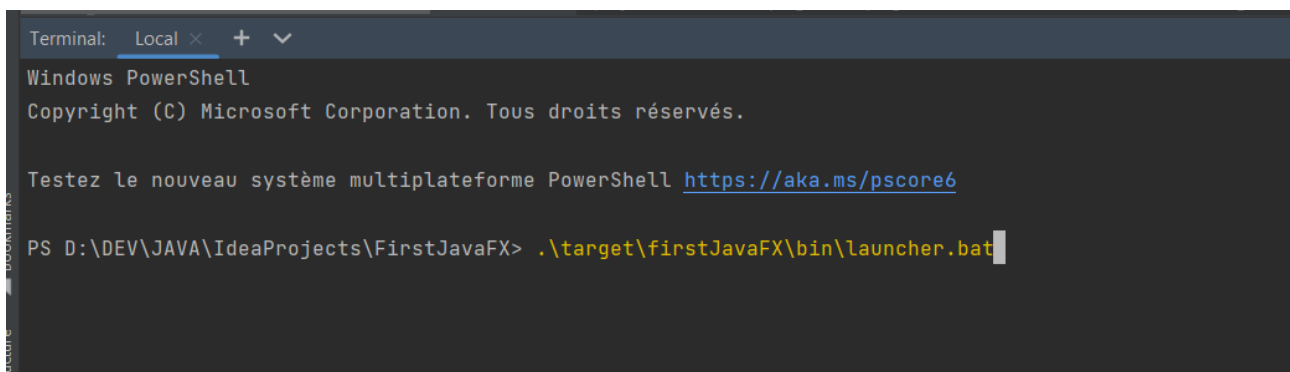


```
[INFO] --- maven-compiler-plugin:3.10.1:compile (default-compile) @ FirstJavaFX
---
[INFO] Changes detected - recompiling the module!
[INFO] Compiling 3 source files to
D:\DEV\JAVA\IdeaProjects\FirstJavaFX\target\classes
[INFO]
[INFO] <<< javafx-maven-plugin:0.0.8:jlink (default-cli) < process-classes @
FirstJavaFX <<<
[INFO]
[INFO]
[INFO] --- javafx-maven-plugin:0.0.8:jlink (default-cli) @ FirstJavaFX ---
[INFO] Building zip: D:\DEV\JAVA\IdeaProjects\FirstJavaFX\target\firstJavaFX.zip
[INFO] -----
[INFO] BUILD SUCCESS
[INFO] -----
[INFO] Total time: 11.883 s
[INFO] Finished at: 2022-12-19T16:28:05+01:00
[INFO] -----

Process finished with exit code 0
```

Cela a pour effet de créer un répertoire « firstJavaFX » et un zip firstJavaFX.zip dans le répertoire target.

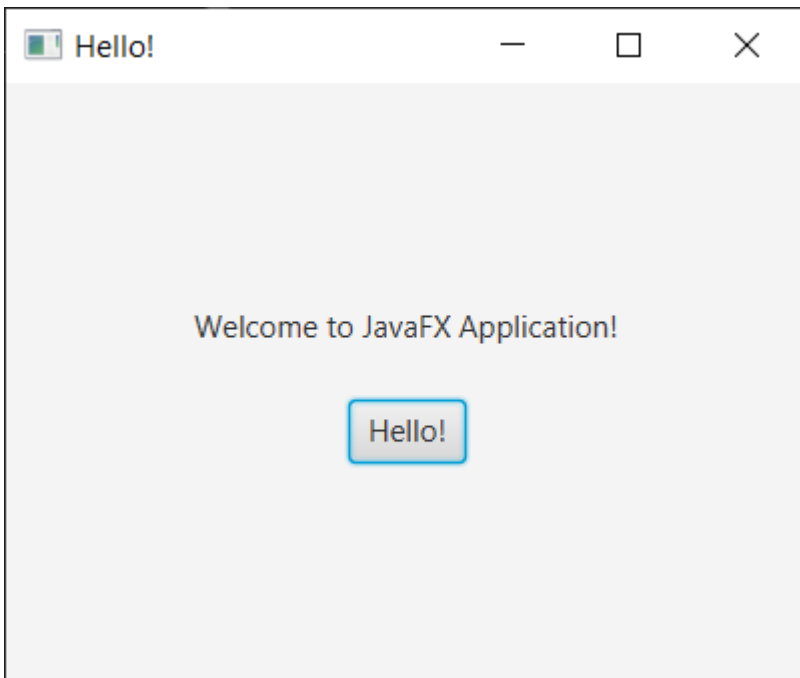
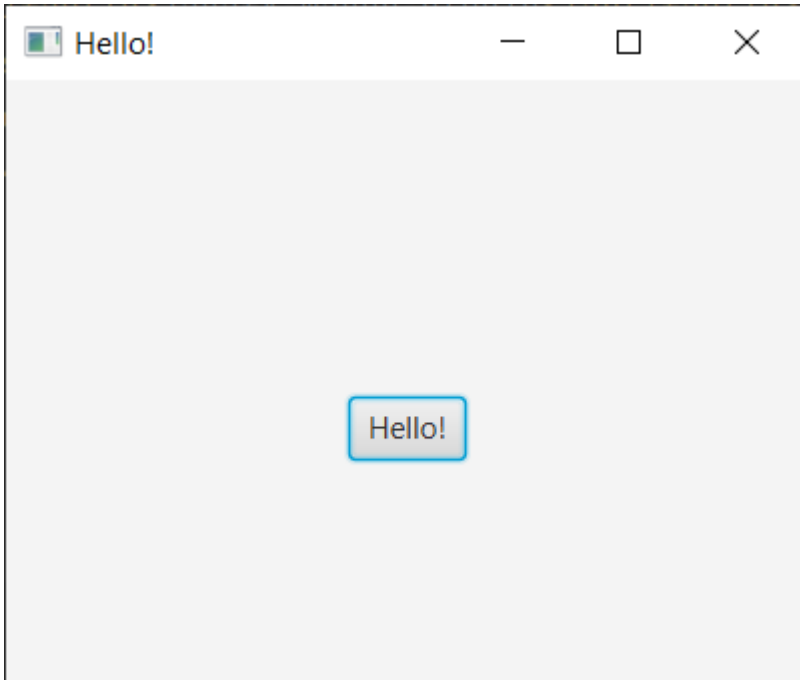
Pour exécuter l'application, il suffit de lancer le launcher.bat se trouvant dans « target\firstJavaFX\bin\launcher.bat »



```
Terminal: Local x + v
Windows PowerShell
Copyright (C) Microsoft Corporation. Tous droits réservés.

Testez le nouveau système multiplateforme PowerShell https://aka.ms/pSCORE6

PS D:\DEV\JAVA\IdeaProjects\FirstJavaFX> .\target\firstJavaFX\bin\launcher.bat
```



VII. Fin du document