

# LAB3 Tutorial

## 一、安装 FX 及 Scene Builder

在本文中，将学习如何在 **Eclipse IDE** 开发和执行 JavaFX 编程。

### e(fx)eclipse

**e(fx)eclipse** 是一组工具和必要的库，它们帮助您执行 JavaFX 编程，确保您已经作为插件将其安装在 eclipse 上了。如果未安装 **e(fx)eclipse**，可以在以下链接查看安装指南：

- <http://www.yiibai.com/javafx/install-efxclipse-into-eclipse.html>

### JavaFX Scene Builder

**JavaFX Scene Builder** 是一种可视化设计工具，它允许通过拖放快速创建应用程序界面。并且代码创建为 **XML** 格式的文件。这是一个选项，为了执行 **JavaFX** 编程，最好也应该安装它。

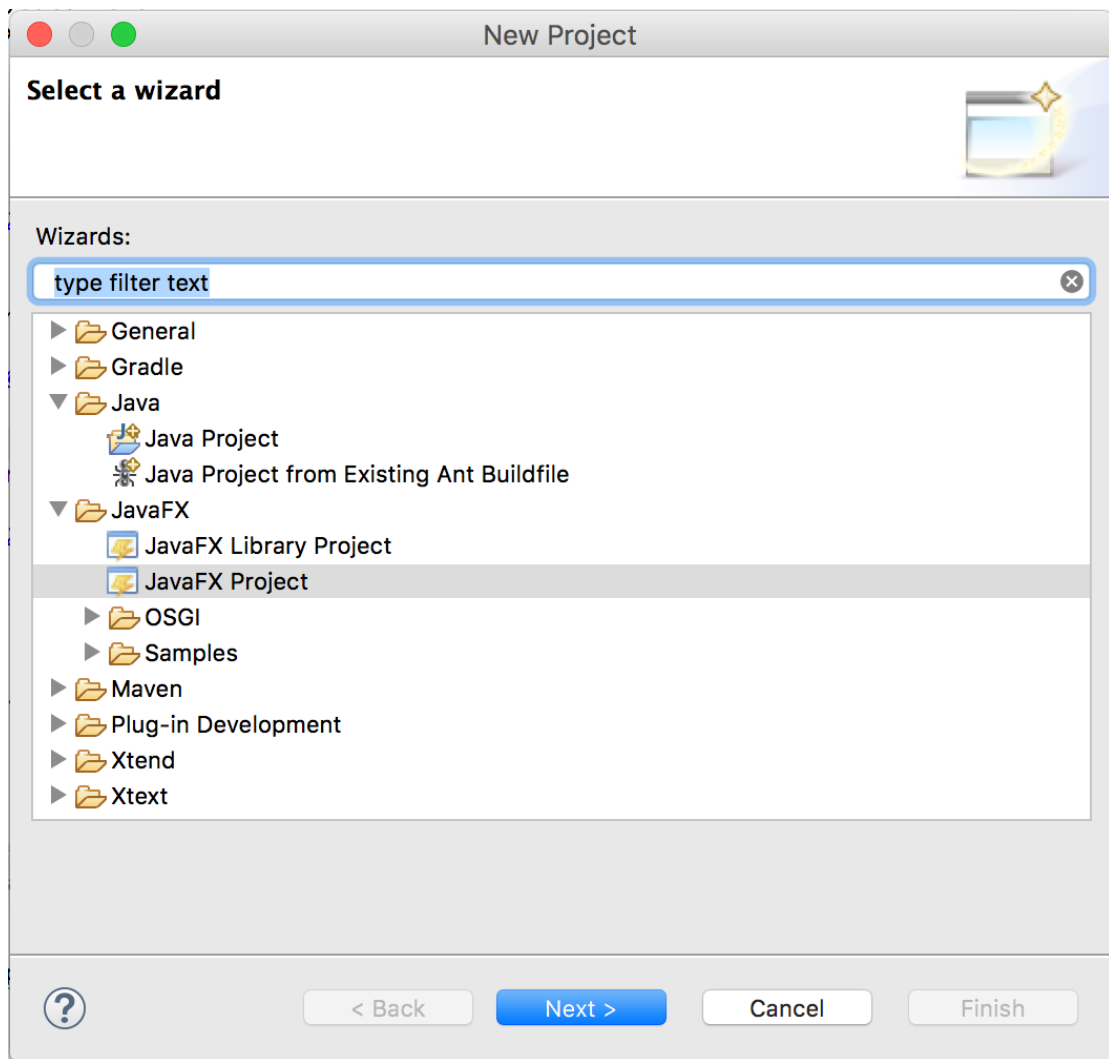
如果还未安装 **JavaFX Scene Builder**，可以在以下链接查看说明：

- <http://www.yiibai.com/javafx/install-javafx-scene-builder-into-eclipse.html>

## 二、创建项目

打开 **Eclipse**，并在 **Eclipse** 中选择：

- **File -> New -> Project**



选择 JavaFX Project。

创建一个项目名称为：JAVA2Lab3Demo（或别的什么），如下图中所示：

New Java Project

### Create a Java Project

Create a Java project in the workspace or in an external location.

Project name:

☒ Use default location

Location:

JRE

☒ Use an execution environment JRE:

☐ Use a project specific JRE:

☐ Use default JRE (currently 'Java SE 8 [1.8.0\_121]') [Configure JREs...](#)

Project layout

☐ Use project folder as root for sources and class files

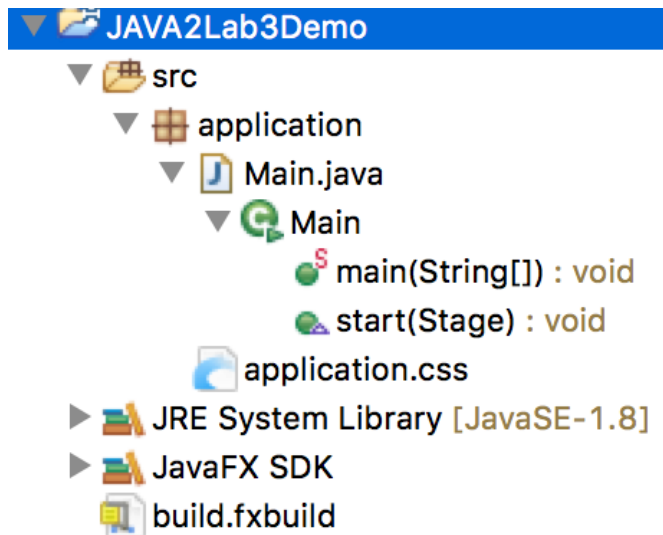
☒ Create separate folders for sources and class files [Configure default...](#)

Working sets

☐ Add project to working sets

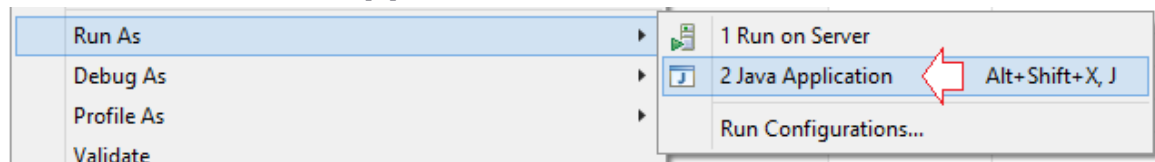
Working sets:

创建完的工程目录结构如下图所示：

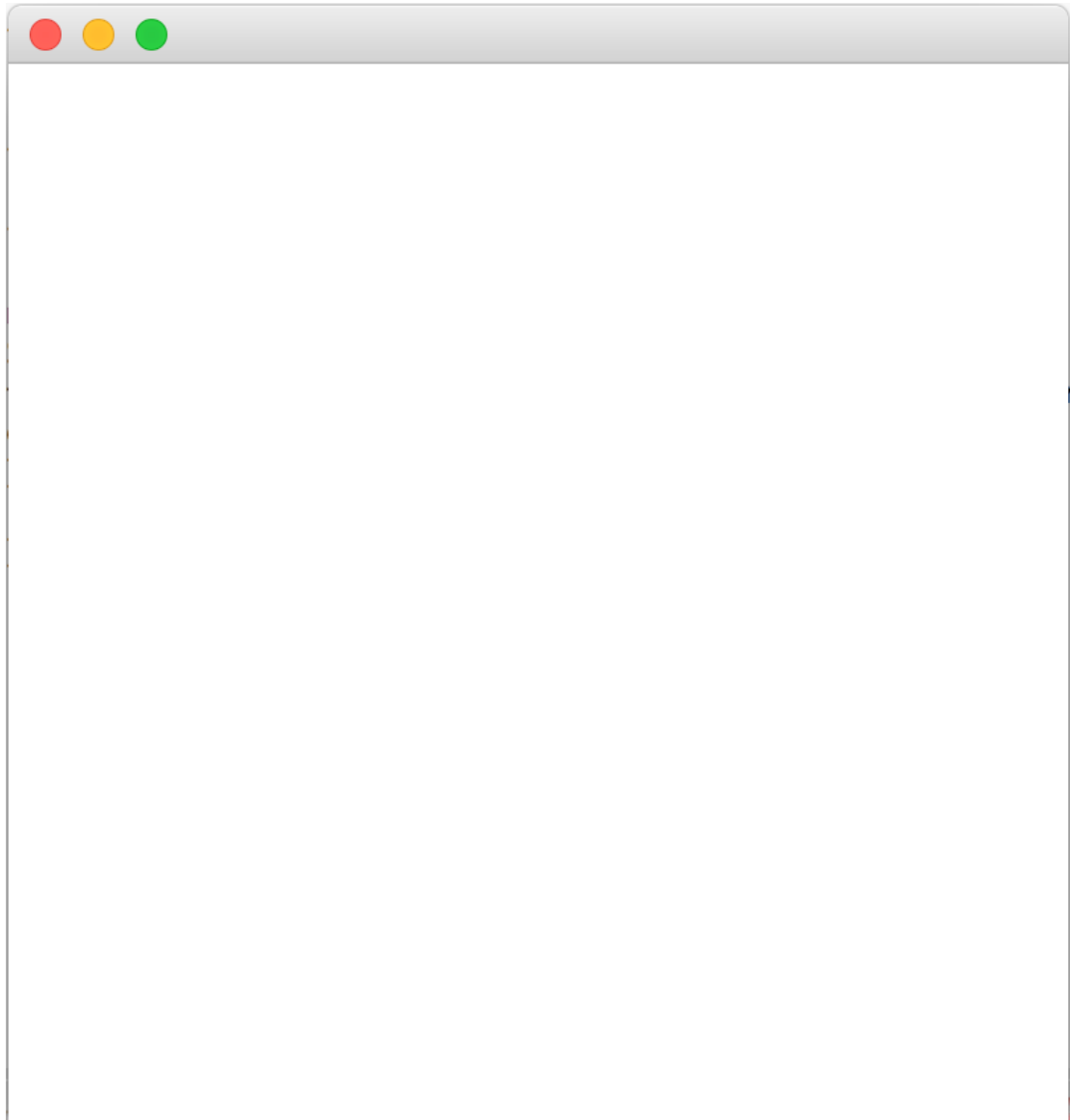


首先，让我们确保运行JAVA2Lab3Demo例子成功。 右键单击主类(Main)并选择：

- **Run As -> Java Application**



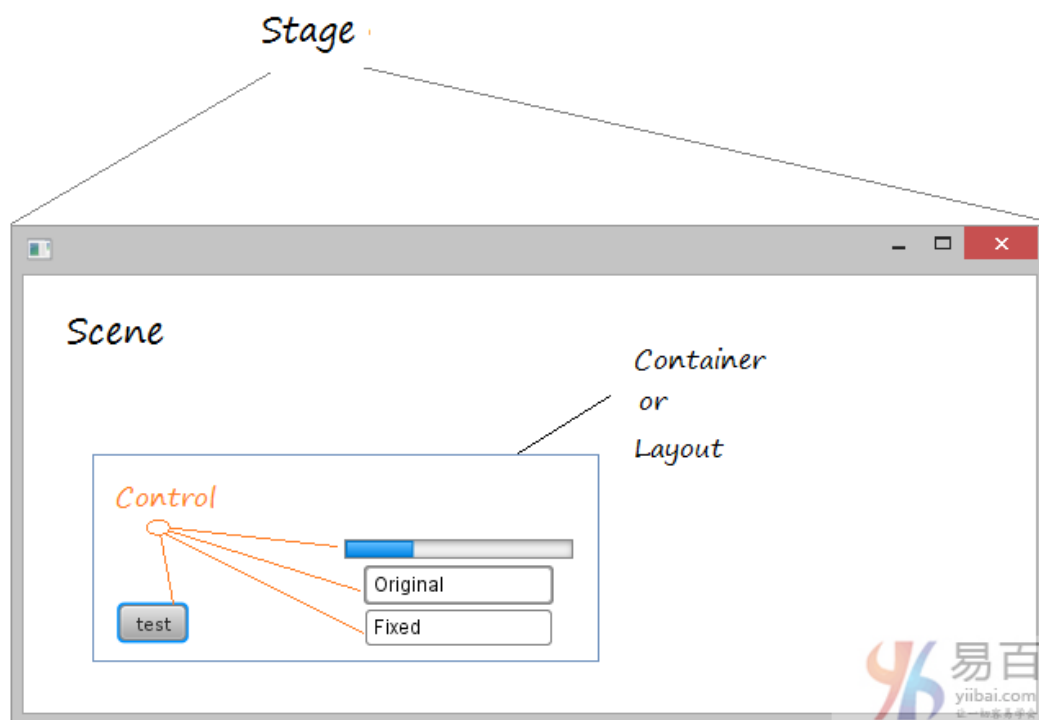
JavaFX的JAVA2Lab3Demo应用正在运行，结果是一个空白的界面，如下：



### 三、 JAVA2Lab3Demo 示例说明

在上面的步骤中，您已成功创建并运行 **JavaFX** 的 **JAVA2Lab3Demo** 应用。

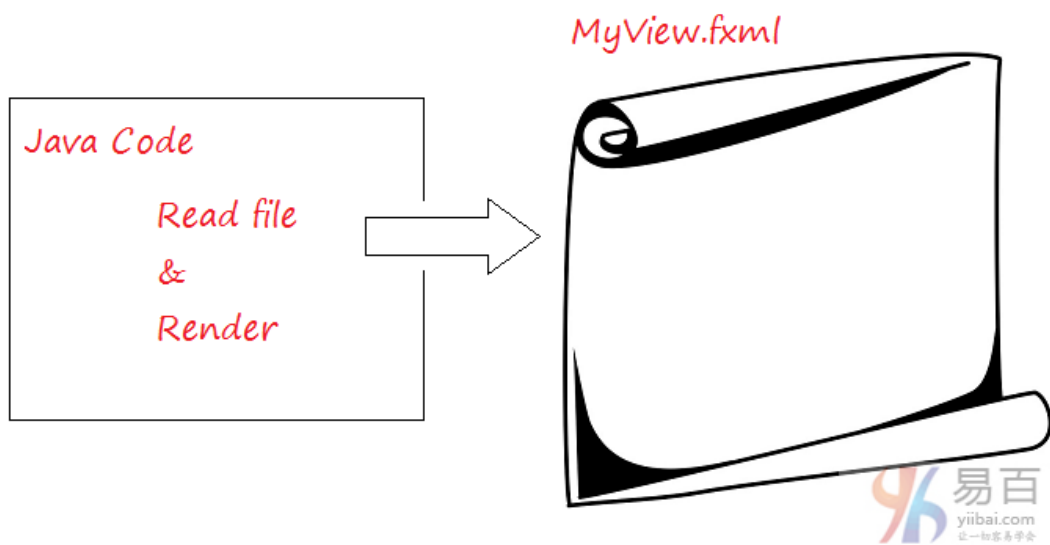
下图显示了舞台(Stage)，场景(Scene)，容器(Container)，布局(Layout )和控件(Controls)之间的关系：



在**JavaFX**中，**Stage**是应用程序窗口，其中包含称为**Scene**的空间。**Scene**包含界面的组件，如**Button**，**Text**，...或容器。

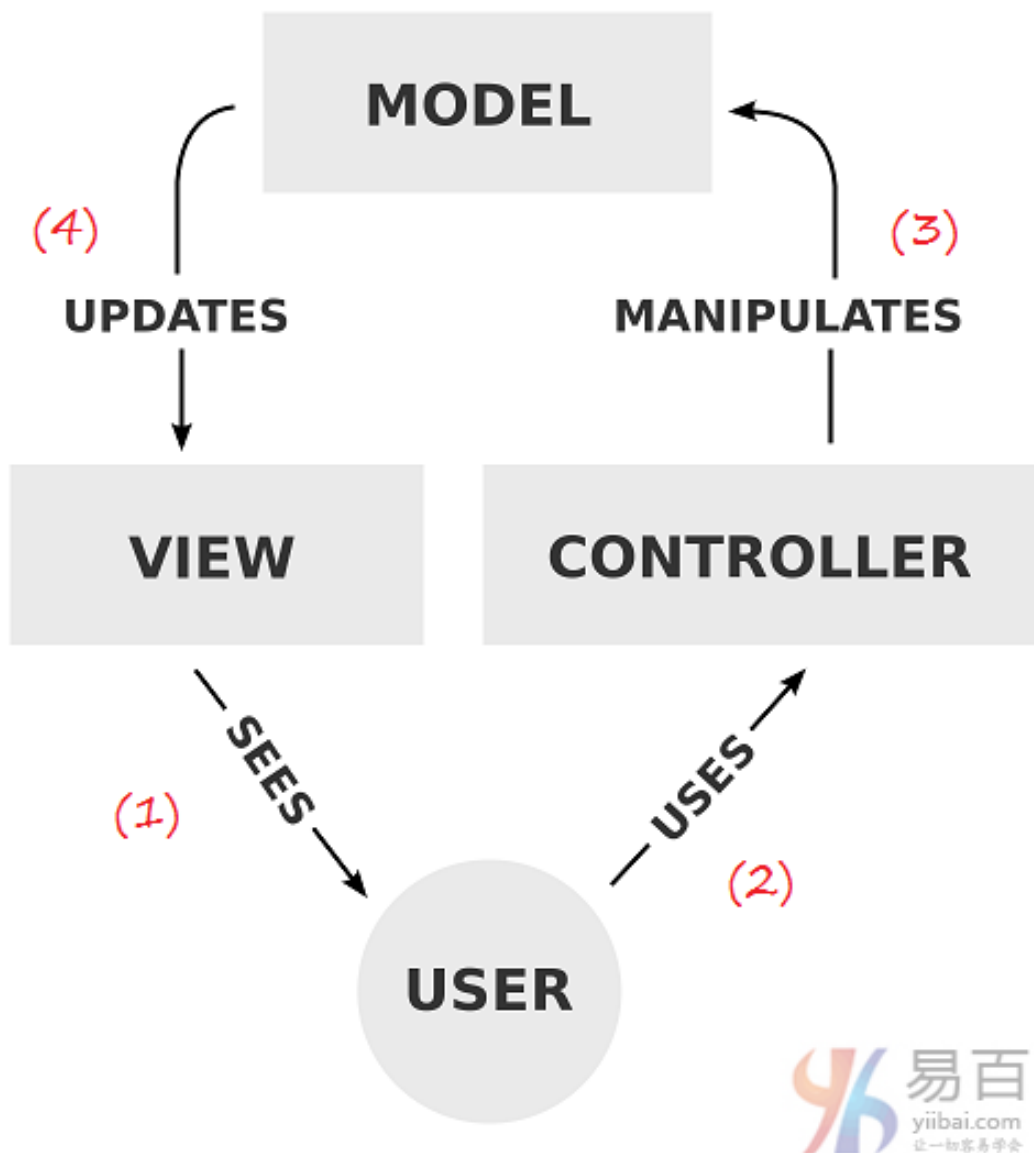
#### 四、JavaFX Scene Builder 简介

为了创建一个**JavaFX**应用程序界面，可以完全编写**Java**代码。但是，需要这么多时间来做到这一点，**JavaFX Scene Builder**是一个可视化工具，允许您设计**Scene**的界面。生成的代码是**XML**代码保存在**\*.fxml**文件中。



## 五、使用 JavaFX Scene Builder 的示例

这是一个小例子，使用**Scene Builder**来设计应用程序界面。应用于该示例的MVC的模型如下所示：

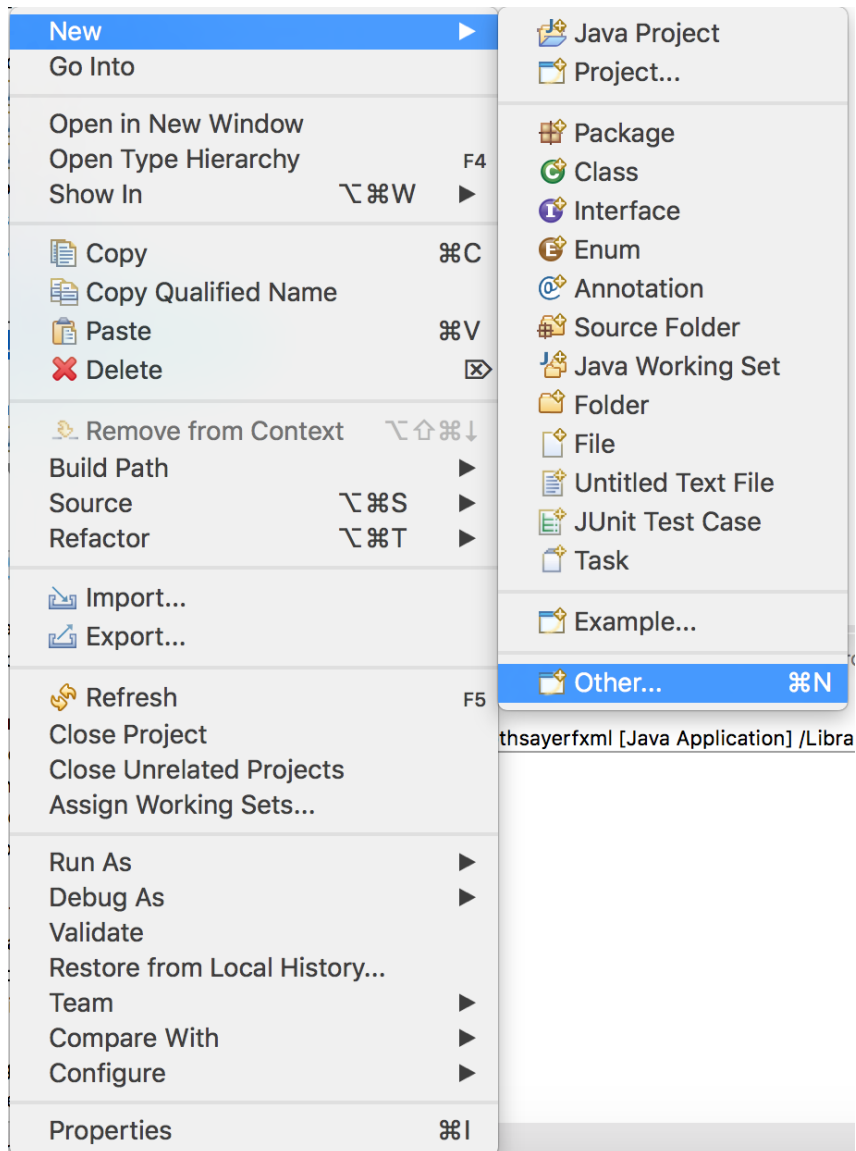


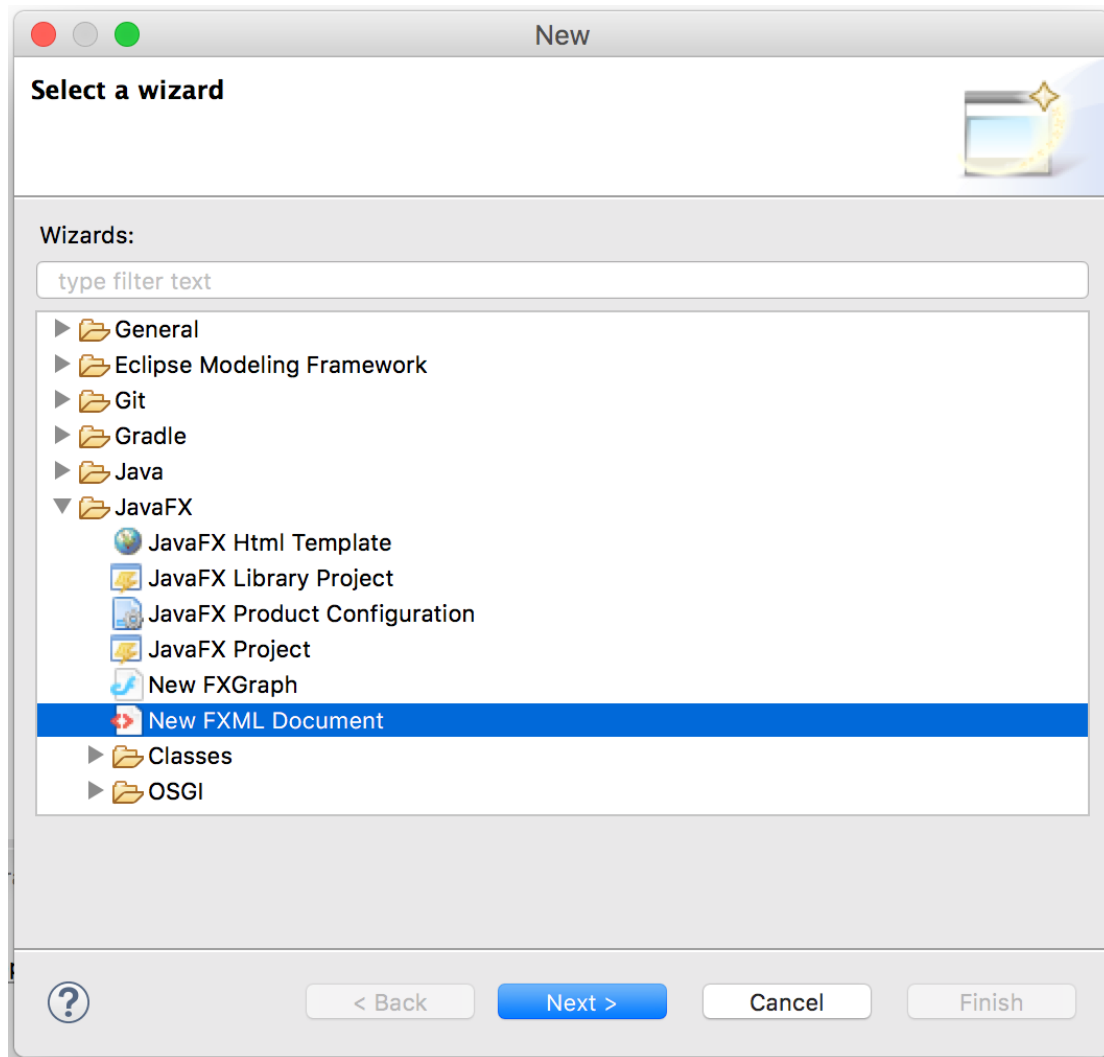
- 在**VIEW**上显示它
- 用户使用**CONTROLLER**
- 操作数据(更新, 修改, 删除, ..), **MODEL**上的数据已更改。
- 在**VIEW**上显示**MODEL**的数据。

下面我们来创建一个新的 **mainUI.xml** 文件。

**File -> New -> Other...**

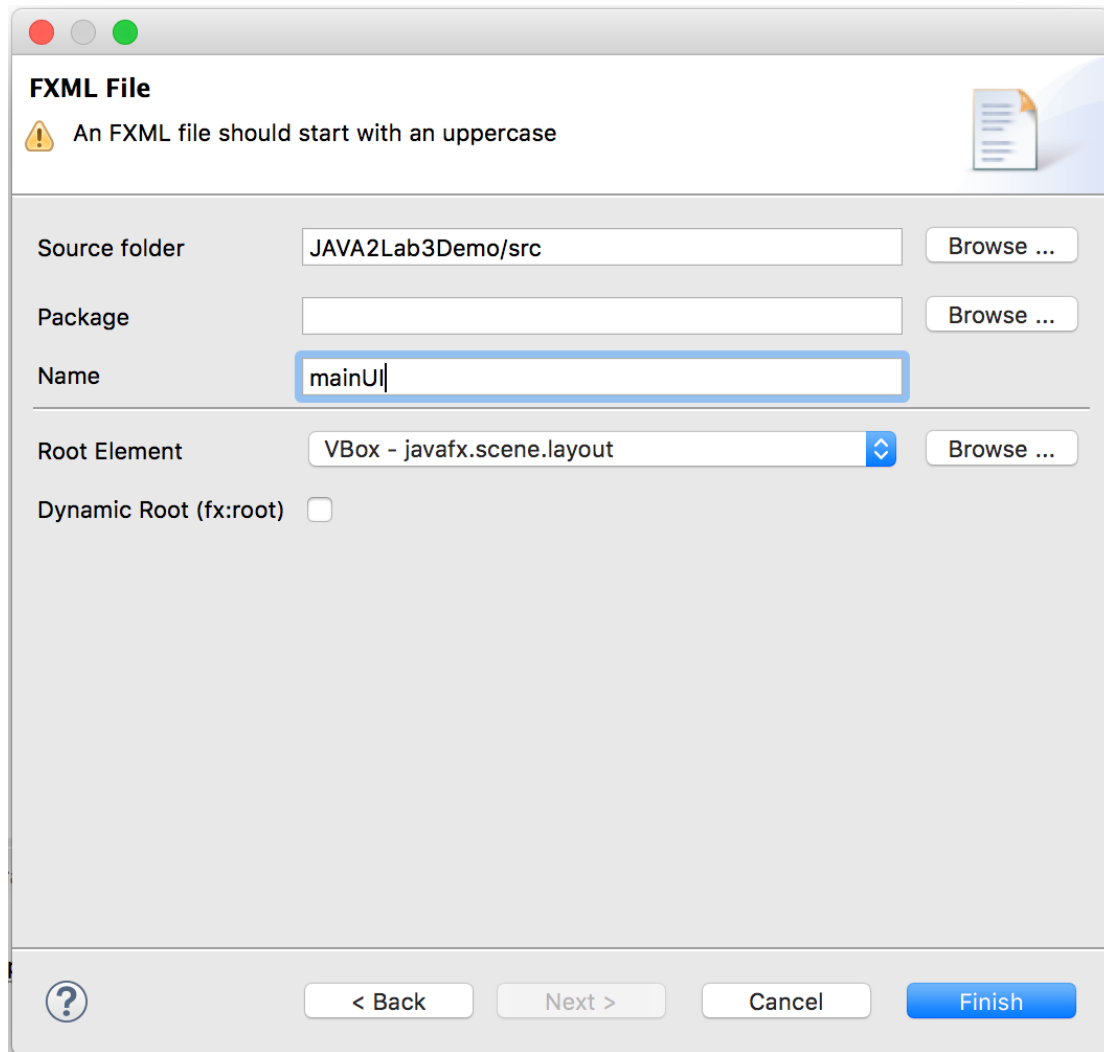






输入文件名称：例如“mianUI”或者自定义名字

Root element 选择 VBox



The image shows a dialog box titled "FXML File" with a warning icon and the message "An FXML file should start with an uppercase". The dialog contains several input fields and buttons:

- Source folder:** A text field containing "JAVA2Lab3Demo/src" and a "Browse ..." button.
- Package:** An empty text field and a "Browse ..." button.
- Name:** A text field containing "mainUI" with a blue selection border.
- Root Element:** A dropdown menu showing "VBox - javafx.scene.layout" and a "Browse ..." button.
- Dynamic Root (fx:root):** A checkbox that is currently unchecked.

At the bottom of the dialog, there is a help icon (question mark), and four buttons: "< Back", "Next >", "Cancel", and "Finish".

创建结果如下图所示 -

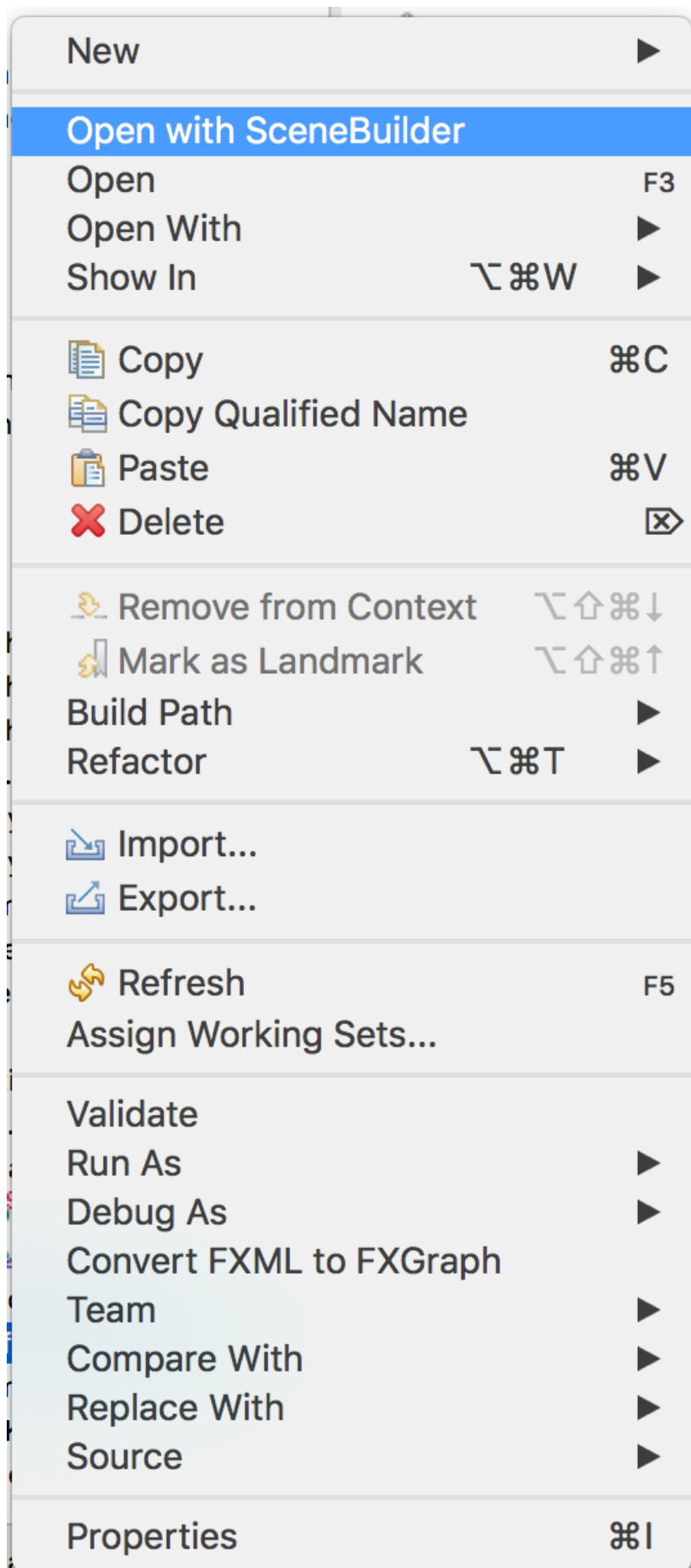


The image shows a code editor window titled "mainUI.fxml" with the following content:

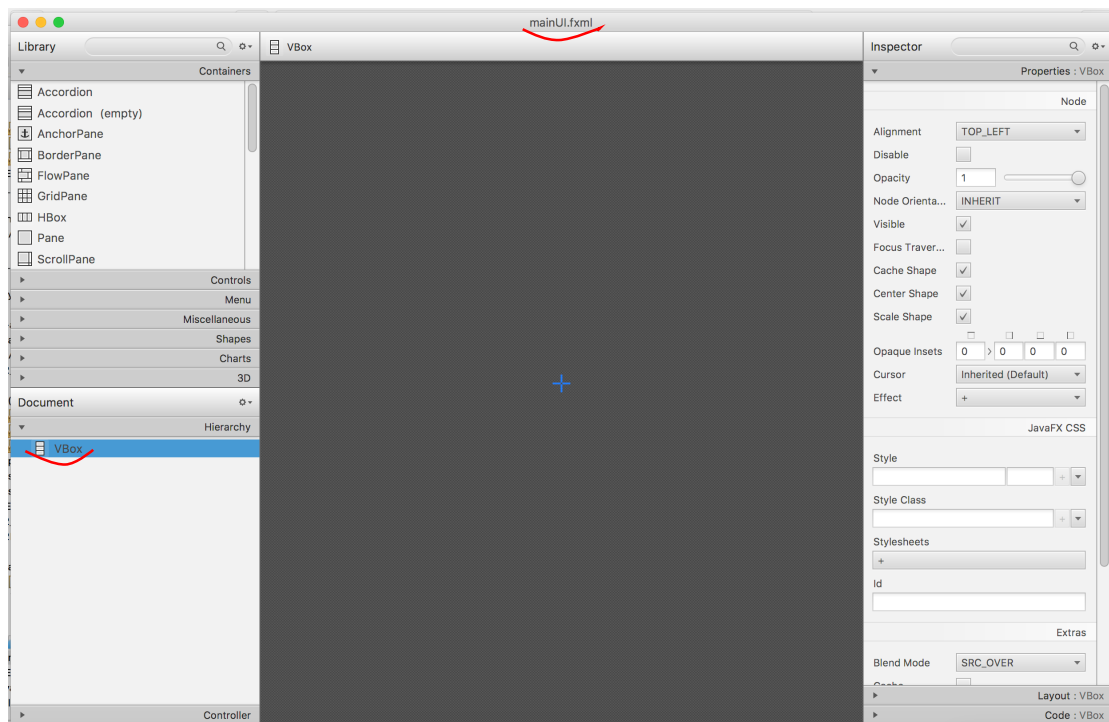
```
1 <?xml version="1.0" encoding="UTF-8"?>
2
3 <?import javafx.scene.layout.VBox?>
4
5 <VBox xmlns:fx="http://javafx.com/fxml/1">
6   <!-- TODO Add Nodes -->
7 </VBox>
8
9 |
```

可以使用 **JavaFX Scene Builder** 打开 **fxml** 文件。

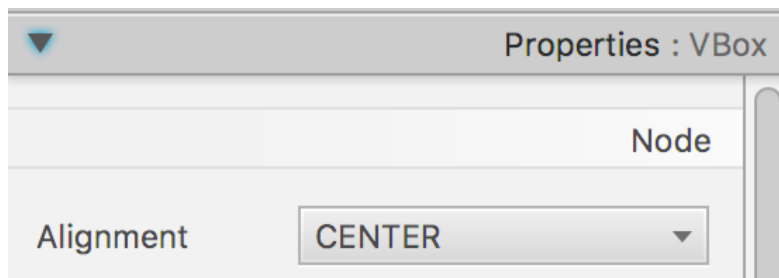
注意：确保已安装 **JavaFX Scene Builder**，并将其集成到 **Eclipse** 中。



mainUI.fxml 的界面设计屏幕如下：



设置 VBox 的属性-



		Size
Min Width	<input type="text" value="USE_PREF_SIZE"/>	▼
Min Height	<input type="text" value="USE_PREF_SIZE"/>	▼
Pref Width	<input type="text" value="600"/>	▼
Pref Height	<input type="text" value="400"/>	▼
Max Width	<input type="text" value="USE_PREF_SIZE"/>	▼
Max Height	<input type="text" value="USE_PREF_SIZE"/>	▼
Width	<input type="text" value="600"/>	
Height	<input type="text" value="400"/>	

► Properties : VBox

▼ Layout : VBox

Internal

Padding ☐ 10 > ☐ 10 ☐ 10 ☐ 10 ☐

Spacing 8

Specific

Fill Width ☒

Size

Min Width USE\_PREF\_SIZE ▼

Min Height USE\_PREF\_SIZE ▼

Pref Width 600 ▼

Pref Height 400 ▼

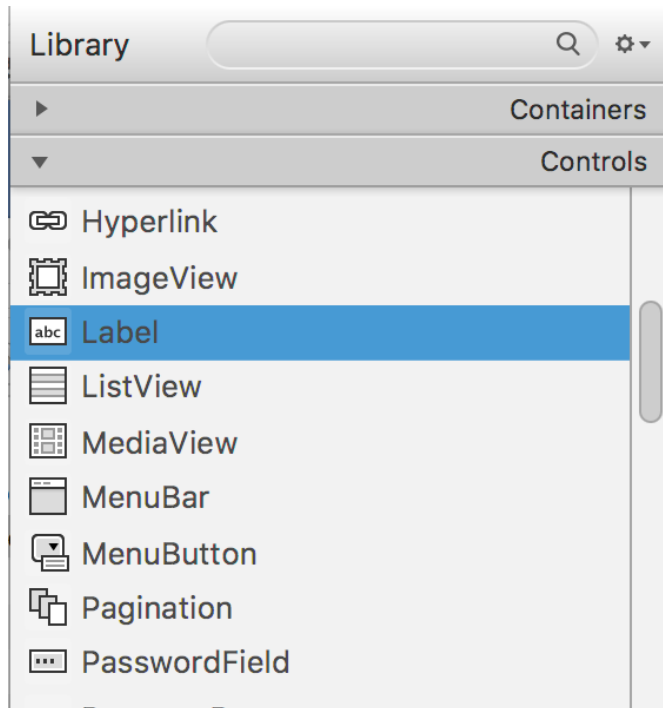
Max Width USE\_PREF\_SIZE ▼

Max Height USE\_PREF\_SIZE ▼

Width 600

Height 400

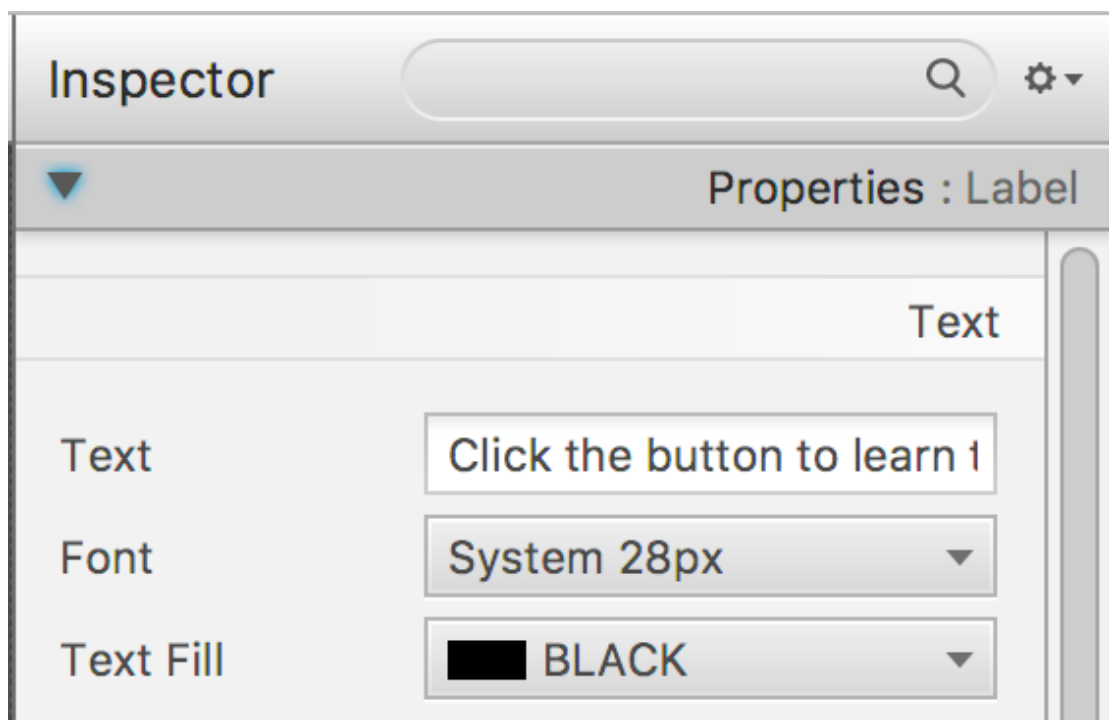
拖拽一个 label 到 VBox 中-



将其 Text 设置为要显示的文字-

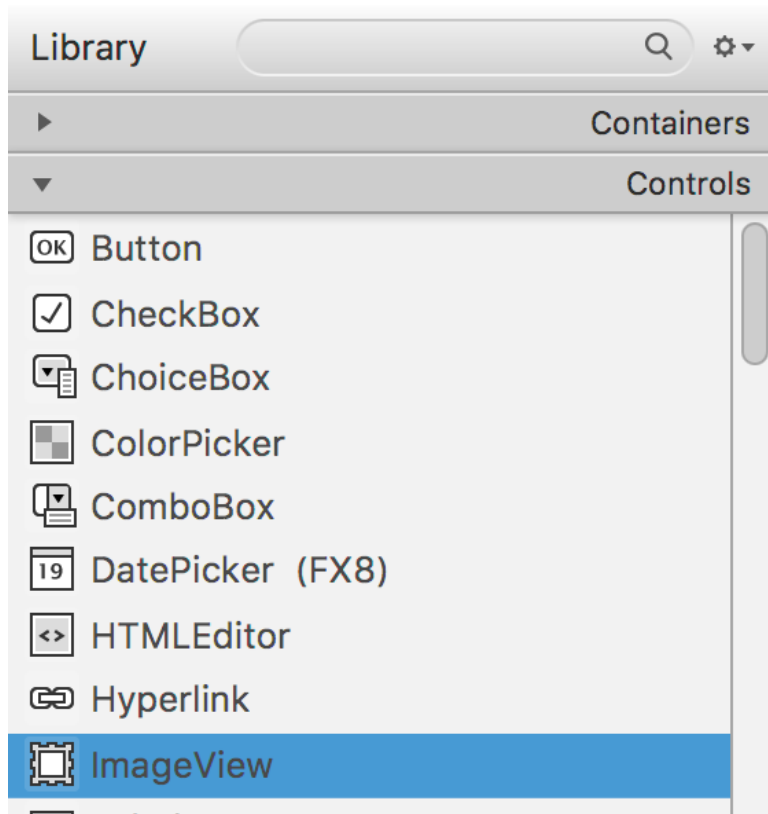
比如：“Click the button to learn the meaning of life”

继续设置字体及文字颜色-

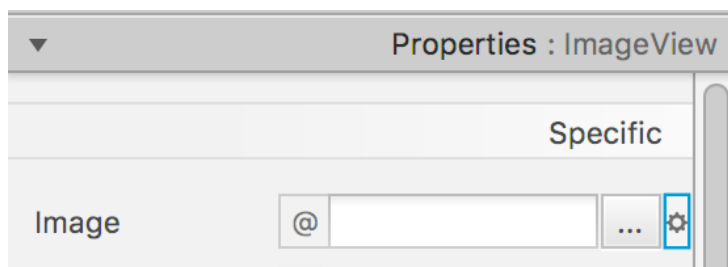


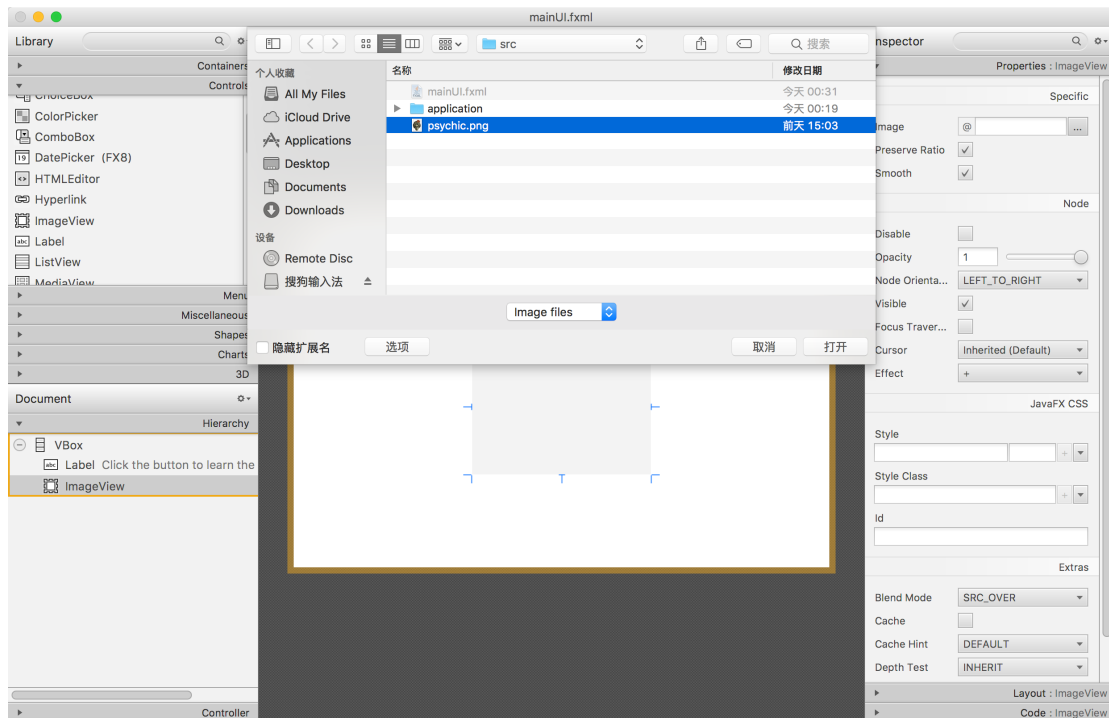
拖拽一个 ImageView 到 VBox 中-



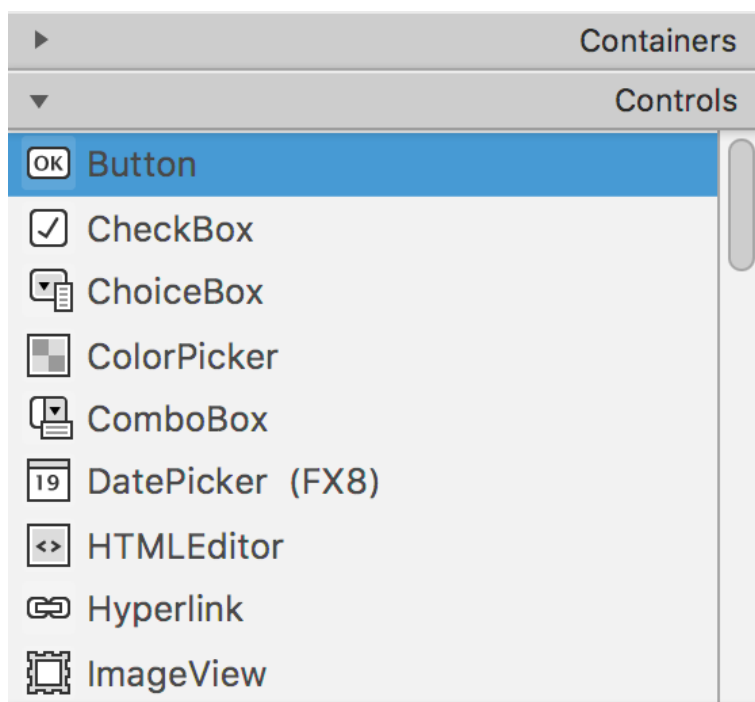


点击 “...” 选择图片-





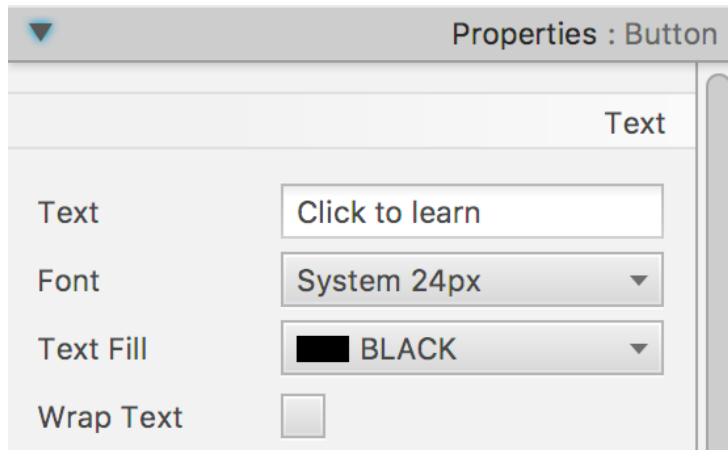
拖拽一个 Button 到 VBox 中-



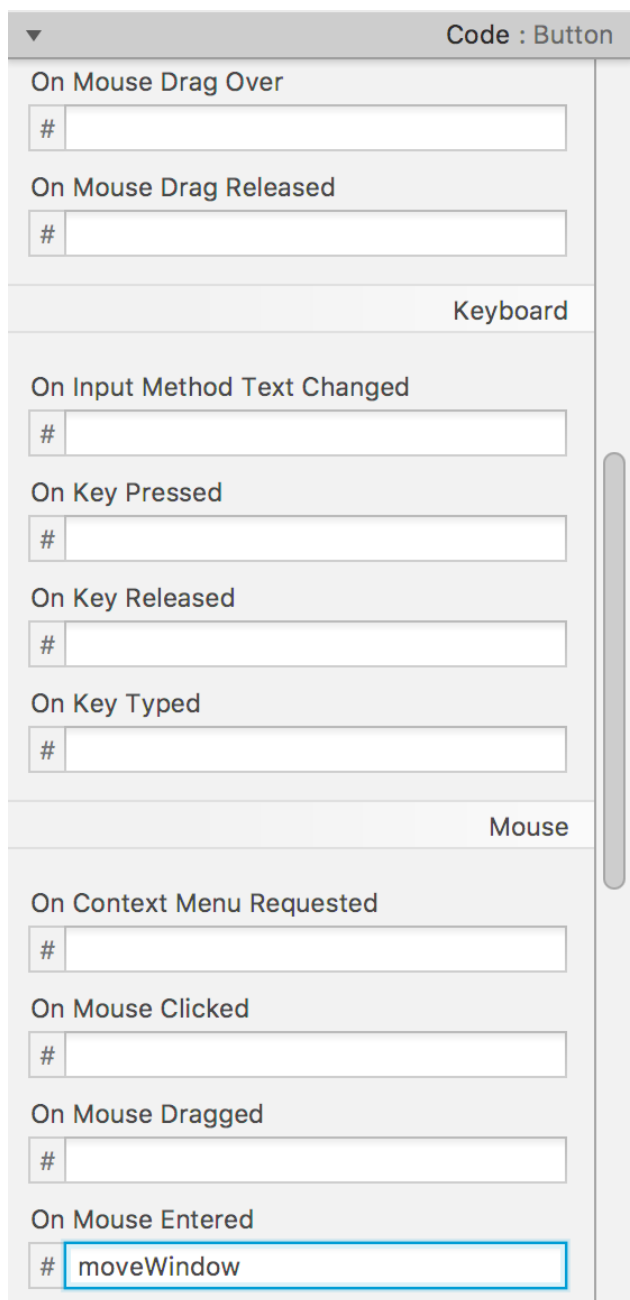
将其 Text 设置为要显示的文字-

比如：“Click to learn”

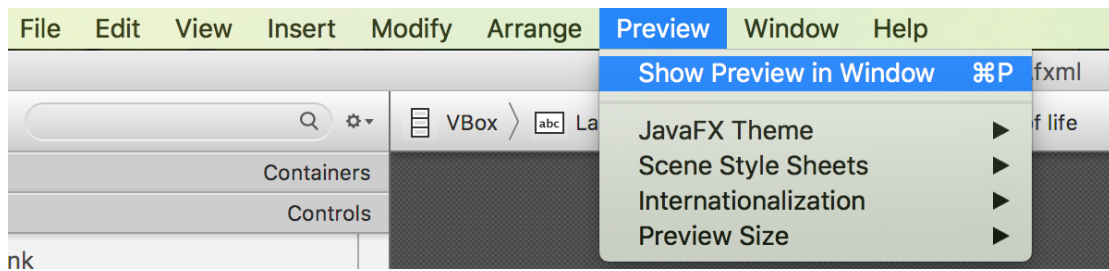
继续设置字体及文字颜色-



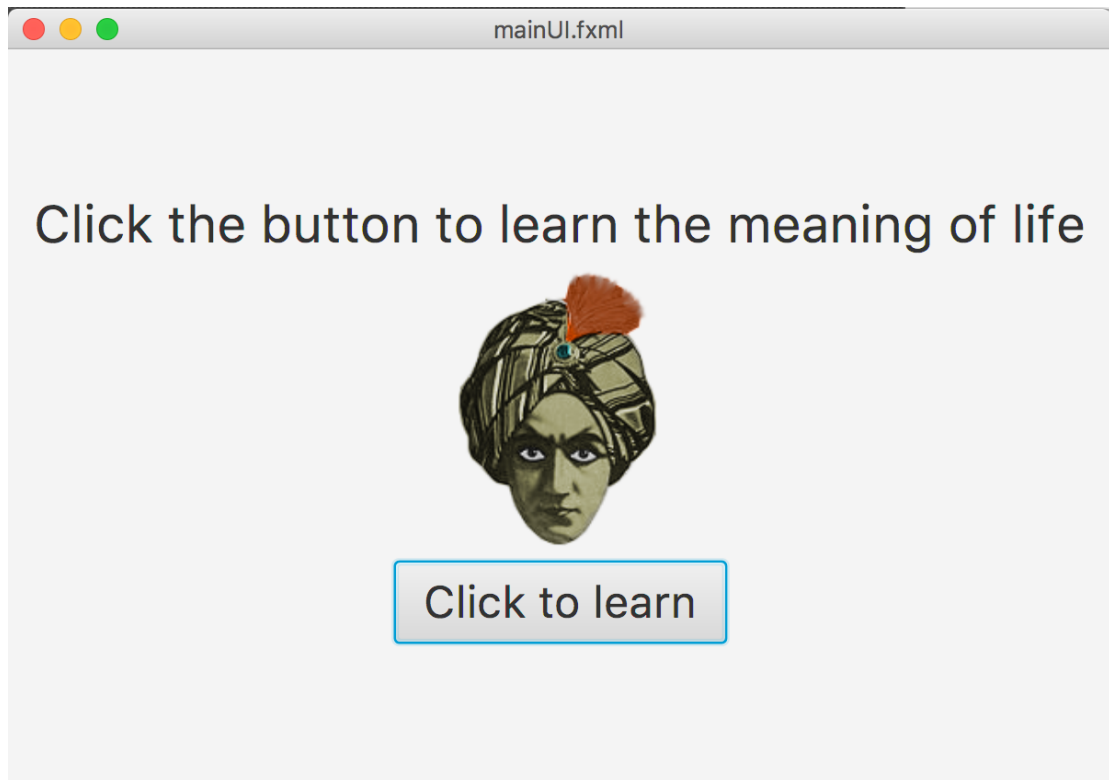
设置方法“moveWindow”，在鼠标进入按钮时调用。



选择文件/保存以保存更改。并在窗口中选择“预览/显示预览”以预览您的设计。



完成后的界面如下图所示：



关闭 **Scene Builder** 窗口并在 **Eclipse** 上刷新项目。您可以查看此时 mainUI.fxml 文件中生成的代码：

```

1 <?xml version="1.0" encoding="UTF-8"?>
2
3 <?import javafx.scene.image.*?>
4 <?import javafx.scene.text.*?>
5 <?import javafx.scene.control.*?>
6 <?import javafx.geometry.*?>
7 <?import java.lang.*?>
8 <?import javafx.scene.layout.*?>
9 <?import javafx.scene.layout.VBox?>
10
11 <VBox alignment="CENTER" maxHeight="-Infinity" maxWidth="-Infinity" minHeight="-Infinity" minWidth="-Infinity"
12   prefHeight="400.0" prefWidth="600.0" spacing="8.0" xmlns="http://javafx.com/javafx/8" xmlns:fx="http://javafx.com/fxml/1">
13   <padding>
14     <Insets bottom="10.0" left="10.0" right="10.0" top="10.0" />
15   </padding>
16   <children>
17     <Label text="Click the button to learn the meaning of life">
18       <font>
19         <font size="28.0" />
20       </font>
21     </Label>
22     <ImageView fitHeight="150.0" fitWidth="200.0" pickOnBounds="true" preserveRatio="true">
23       <image>
24         <Image url="@psychic.png" />
25       </image>
26     </ImageView>
27     <Button mnemonicParsing="false" onMouseEntered="#moveWindow" text="Click to learn">
28       <font>
29         <font size="24.0" />
30       </font>
31     </Button>
32   </children>
33 </VBox>
34

```

将属性 `fx:controller` 添加到 `<VBox>` 中，Controller 将对位于 `VBox` 内部的控件 (如 `myButton`) 有引用。

```

1 <?xml version="1.0" encoding="UTF-8"?>
2
3 <?import javafx.scene.image.*?>
4 <?import javafx.scene.text.*?>
5 <?import javafx.scene.control.*?>
6 <?import javafx.geometry.*?>
7 <?import java.lang.*?>
8 <?import javafx.scene.layout.*?>
9 <?import javafx.scene.layout.VBox?>
10
11 <VBox alignment="CENTER" maxHeight="-Infinity" maxWidth="-Infinity" minHeight="-Infinity" minWidth="-Infinity" prefHeight="400.0"
12   prefWidth="600.0" spacing="8.0" xmlns="http://javafx.com/javafx/8" xmlns:fx="http://javafx.com/fxml/1" fx:controller="application.MainUIxml">
13   <padding>
14     <Insets bottom="10.0" left="10.0" right="10.0" top="10.0" />
15   </padding>
16   <children>
17     <Label text="Click the button to learn the meaning of life">
18       <font>
19         <font size="28.0" />
20       </font>
21     </Label>
22     <ImageView fitHeight="150.0" fitWidth="200.0" pickOnBounds="true" preserveRatio="true">
23       <image>
24         <Image url="@psychic.png" />
25       </image>
26     </ImageView>
27     <Button fx:id="myButton" mnemonicParsing="false" onMouseEntered="#moveWindow" text="Click to learn">
28       <font>
29         <font size="24.0" />
30       </font>
31     </Button>
32   </children>
33 </VBox>
34

```

此时，我们在 `application` 包中新建 class，名为 `MainUIxml`，记住这个类的名字要跟 `fxml` 中设置的一致。

New Java Class

**Java Class**

✖ Type already exists.

Source folder: JAVA2Lab3Demo/src Browse...

Package: application Browse...

☐ Enclosing type: Browse...

Name: MainUixml

Modifiers: ☒ public ☐ package ☐ private ☐ protected  
☐ abstract ☐ final ☐ static

Superclass: java.lang.Object Browse...

Interfaces: Add... Remove

Which method stubs would you like to create?

☐ public static void main(String[] args)  
☐ Constructors from superclass  
☐ Inherited abstract methods

Do you want to add comments? (Configure templates and default value [here](#))

☐ Generate comments

? Cancel Finish

MainUixml.java 的代码如下图所示：

```
mainUI.fxml MainUIxml.java
1 package application;
2
3 import java.net.URL;
4
15 public class MainUIxml implements Initializable {
16     // Screen size
17     private Rectangle2D screenBounds
18         = Screen.getPrimary().getVisualBounds();
19     // The random generator MUST be static
20     private static Random rand_generator ;
21     // Keep track of the current location of the window
22     private double x;
23     private double y;
24
25
26
27 @Override
28 public void initialize(URL location, ResourceBundle resources) {
29     // TODO Auto-generated method stub
30     rand_generator = new Random();
31 }
32
33 public void moveWindow(MouseEvent me) {
34     Node node = (Node) me.getSource();
35     // Returns a reference to the button
36     Stage stage = (Stage) node.getScene().getWindow();
37     double height = screenBounds.getHeight();
38     double width = screenBounds.getWidth();
39
40     // Add a fixed value to make sure that the Window
41     // moves far enough
42     double x_move = width / 10 + rand_generator.nextDouble() * width / 2;
43     double y_move = height / 10 + rand_generator.nextDouble() * height / 2;
44     // As x and y represent the upper left corner, we
45     // take care of not having part of most of the
46     // window outside the screen
47     this.x = (double) ((long) (this.x + x_move) % (long) (width - stage.getWidth()));
48     this.y = (double) ((long) (this.y + y_move) % (long) (height - stage.getHeight()));
49     stage.setX(this.x);
50     stage.setY(this.y);
51 }
52 }
```

Main.java 的代码如下图所示：

```
mainUI.fxml MainUIxml.java Main.java
1 package application;
2
3 import javafx.application.Application;
4 import javafx.fxml.FXMLLoader;
5 import javafx.stage.Stage;
6 import javafx.scene.Parent;
7 import javafx.scene.Scene;
8
9 public class Main extends Application {
10     @Override
11     public void start(Stage primaryStage) {
12         try {
13             Parent root = FXMLLoader.load(getClass().getResource("../mainUI.fxml"));
14
15             primaryStage.setTitle("My Application");
16             primaryStage.setScene(new Scene(root));
17             primaryStage.show();
18         } catch (Exception e) {
19             e.printStackTrace();
20         }
21     }
22
23     public static void main(String[] args) {
24         launch(args);
25     }
26 }
27 }
```

## 六、注意事项

虽然本 Tutorial 中的代码保留了 application 的包，并且 fxml 和.java 文件也并没有在同一个目录下，但是同学们交作业的时候请注意，请删除包，不要这个系统默认的 application 的 package，提交的代码不需要有包，并将所有的代码写到一个.java 中去，将 fxml 和.java 文件也要放到同一个目录下，切记切记。

## 七、参考教程

<http://www.yiibai.com/javafx/>