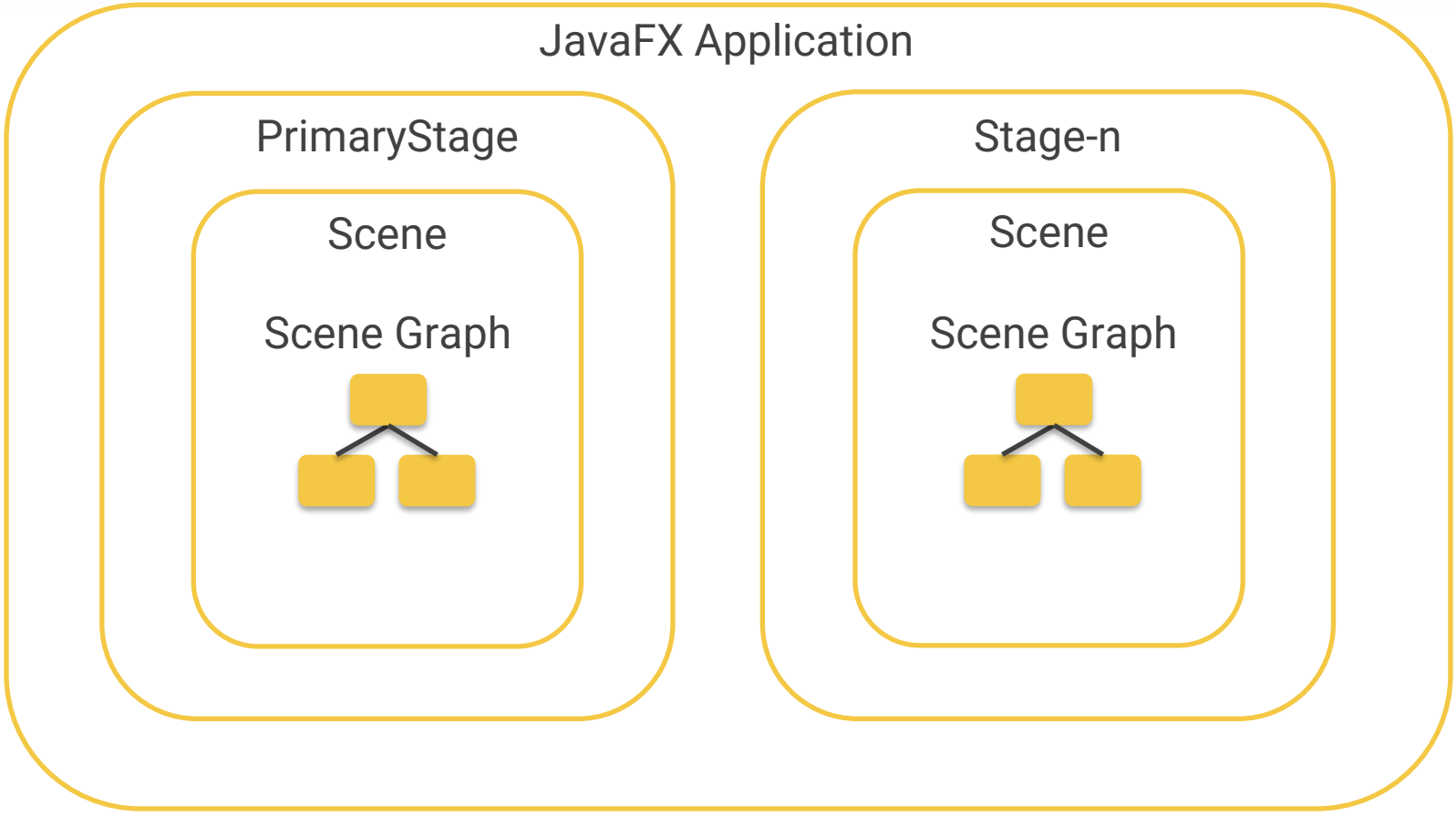
First GUI Project

La Salle Gràcia

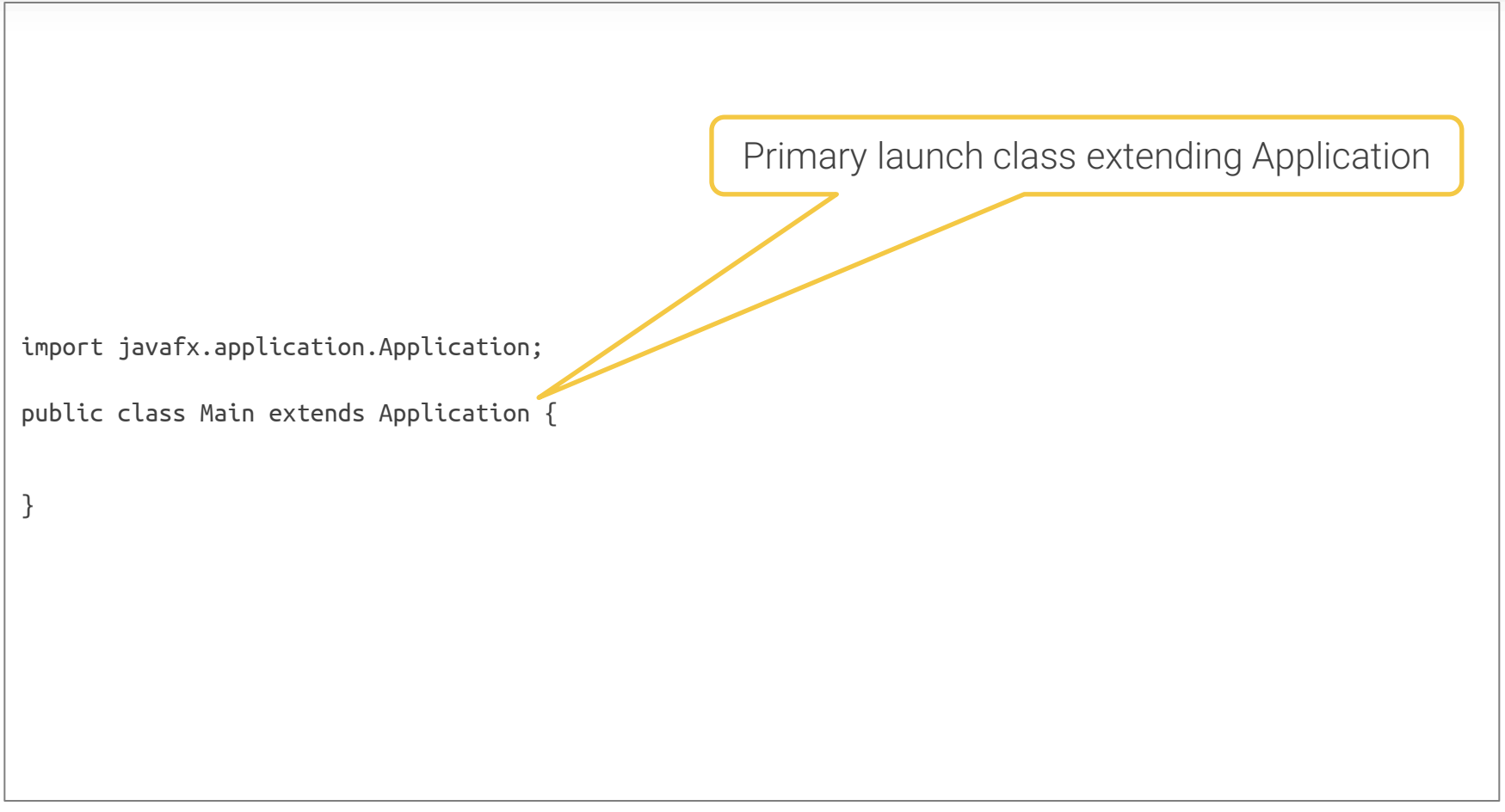
1. JavaFX Application
2. FXML

JavaFX Application

1. JavaFX Application



1. JavaFX Application



```
import javafx.application.Application;
import javafx.stage.Stage;

public class Main extends Application {

    @Override
    public void start(Stage primaryStage) throws Exception{
        primaryStage.setTitle("Hello World");
        primaryStage.show();
    }

}
```

Implement the abstract method start()

Call the method show to display the Stage

1. JavaFX Application

1. JavaFX Application

```
import javafx.application.Application;
import javafx.scene.Scene;
import javafx.scene.control.Label;
import javafx.stage.Stage;

public class Main extends Application {

    @Override
    public void start(Stage primaryStage) throws Exception{
        primaryStage.setTitle("Hello World");
        Label label = new Label("My first JavaFX Application!");
        Scene scene = new Scene(label, 400, 200);
        primaryStage.setScene(scene);
        primaryStage.show();
    }

}
```

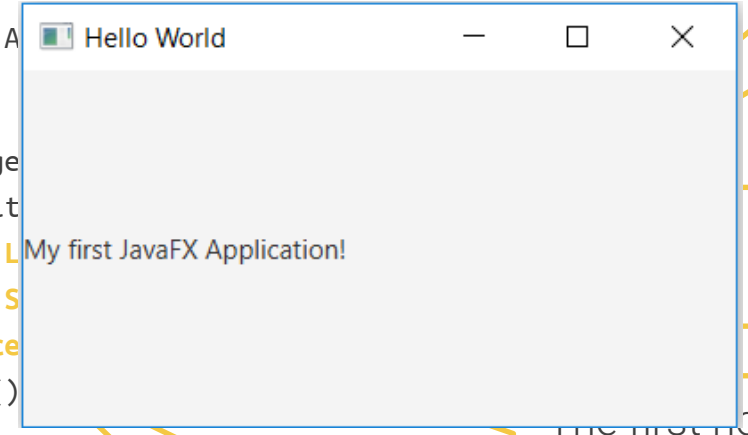
Create a Label control

Create a Scene and add the Label control to it

The first node added into the scene is the root of the scene graph

Add the Scene to the stage

```
import javafx.application.Application;
import javafx.scene.Scene;
import javafx.scene.control.Label;
import javafx.stage.Stage;

public class Main extends A

    @Override
    public void start(Stage
        primaryStage.setTit
        Label label = new L
        Scene scene = new S
        primaryStage.setSce
        primaryStage.show()
    }

}
```



Create a Label control

Create a Scene and add the Label control to it

ode added into the scene is the root of the scene graph

Add the Scene to the stage

1. JavaFX Application

```
import javafx.application.Application;
import javafx.scene.Scene;
import javafx.scene.control.Label;
import javafx.stage.Stage;

public class Main extends Application {

    @Override
    public void start(Stage primaryStage) throws Exception{
        primaryStage.setTitle("Hello World");
        Label label = new Label("My first JavaFX Application!");
        Scene scene = new Scene(label, 400, 200);
        primaryStage.setScene(scene);
        primaryStage.show();
    }

    public static void main(String[] args) {
        launch(args);
    }

}
```

Optionally implement main method

Must call Application.launch

FXML

FXML features

- User interface markup language created by Oracle for JavaFX
- XML format
- Allow separating the layout code from the rest of the JavaFX application
- Can be used both to define the whole app layout or just small parts
- Anything created or implemented in a FXML file can be expressed using JavaFX directly

```
<?import javafx.scene.control.Label?>
<Label text="Hello, World!"/>
```

FXML features

- User interface markup language created by Oracle for JavaFX
- XML format
- Allow separating the layout code from the rest of the JavaFX application
- Can be used both to define the whole app layout or just small parts
- Anything created or implemented in a FXML file can be expressed using JavaFX directly

```
<?import javafx.scene.control.Label?>
<Label text="Hello, World!"/>
```

JavaFX

```
BorderPane border = new BorderPane();
Label toppanetext = new Label("Page Title");
border.setTop(toppanetext);
Label centerpanetext = new Label ("Some data here");
border.setCenter(centerpanetext);
```

FXML

```
<BorderPane>
    <top>
        <Label text="Page Title"/>
    </top>
    <center>
        <Label text="Some data here"/>
    </center>
</BorderPane>
```

So,... when should use FXML and when should use JavaFX?

FXML peforms better than JavaFX and is easier to undestand.

- Implement the project GUIs using FXML
- Use JavaFX only to modify the scene in run time

Main.java

```
import javafx.application.Application;
import javafx.scene.Scene;
import javafx.scene.control.Label;
import javafx.stage.Stage;

public class Main extends Application {

    @Override
    public void start(Stage primaryStage) throws Exception{
        primaryStage.setTitle("Hello World");
        Label label = new Label("My first JavaFX Application!");
        Scene scene = new Scene(label, 400, 200);
        primaryStage.setScene(scene);
        primaryStage.show();
    }

    public static void main(String[] args) {
        launch(args);
    }

}
```

Main.java

```
import javafx.application.Application;
import javafx.scene.Scene;
import javafx.scene.control.Label;
import javafx.stage.Stage;

public class Main extends Application {

    @Override
    public void start(Stage primaryStage) throws Exception{
        primaryStage.setTitle("Hello World");
        Label label = new Label("My first JavaFX Application!");
        Scene scene = new Scene(label, 400, 200);
        primaryStage.setScene(scene);
        primaryStage.show();
    }

    public static void main(String[] args) {
        launch(args);
    }

}
```

Main.java

```
...
@Override
    public void start(Stage primaryStage) throws Exception{
        primaryStage.setTitle("Hello World");
        Label label = new Label("My first JavaFX Application!");
        Scene scene = new Scene(label, 400, 200);
        primaryStage.setScene(scene);
        primaryStage.show();
    }
...
```

myScene.fxml

```
<?import javafx.scene.control.Label?>
<Label text="My first JavaFX Application!"/>
```

Main.java

```
...
import javafx.fxml.FXMLLoader;
import javafx.scene.Parent;
...
@Override
    public void start(Stage primaryStage) throws Exception{
        primaryStage.setTitle("Hello World");
        Parent root = FXMLLoader.load(getClass().getResource("myScene.fxml"));
        Scene scene = new Scene(root, 400, 200);
        primaryStage.setScene(scene);
        primaryStage.show();
    }
...
```

myScene.fxml

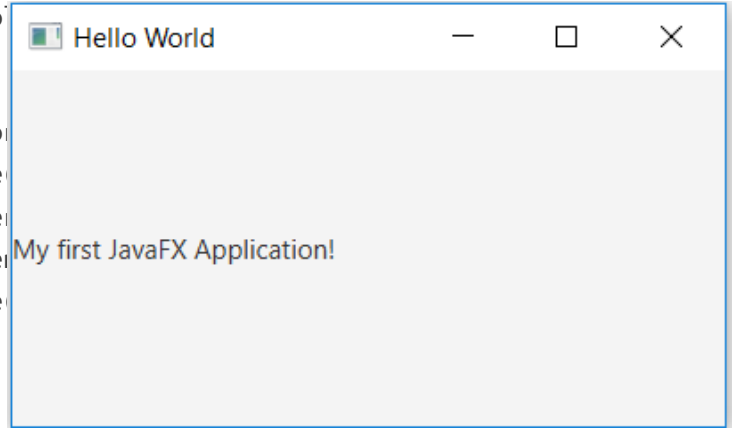
```
<?import javafx.scene.control.Label?>
<Label text="My first JavaFX Application!"/>
```

Main.java

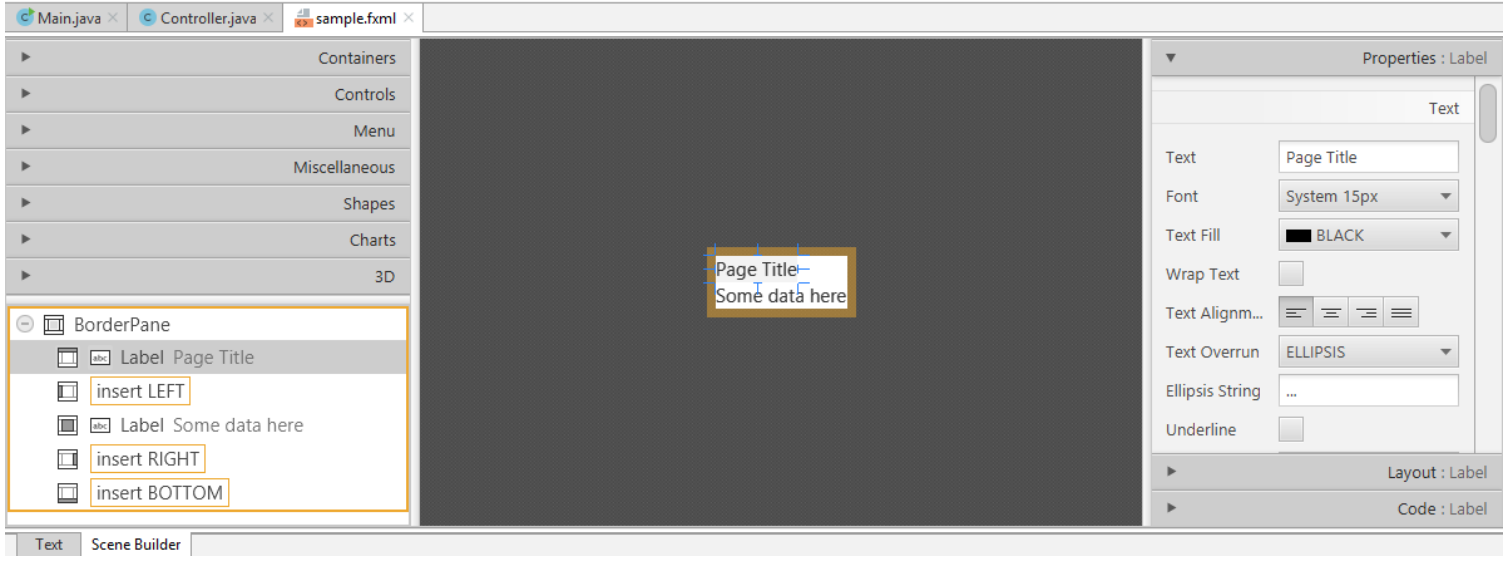
```
 import javafx.application.Application;
        import javafx.scene.Scene;
        import javafx.fxml.FXMLLoader;
        import javafx.scene.Node;
        import javafx.stage.Stage;

public class Main extends Application {

    @Override
    public void start(Stage primaryStage) throws Exception{
        primaryStage.setTitle("Hello World");
        Node root = FXMLLoader.load(getClass().getResource("sample.fxml"));
        Scene scene = new Scene(root, 400, 200);
        primaryStage.setScene(scene);
        primaryStage.show();
    }

    public static void main(String[] args) {
        launch(args);
    }

}
```

Main.java

```
 import javafx.application.Application;
        import javafx.scene.Scene;
        import javafx.fxml.FXMLLoader;
        import javafx.scene.Node;
        import javafx.stage.Stage;

public class Main extends App

    @Override
    public void start(Stage p
        primaryStage.setTitle
        Node root = FXMLLoade
        Scene scene = new Sce
        primaryStage.setScene
        primaryStage.show();
    }

    public static void main(String[] args) {
        launch(args);
    }

}
```



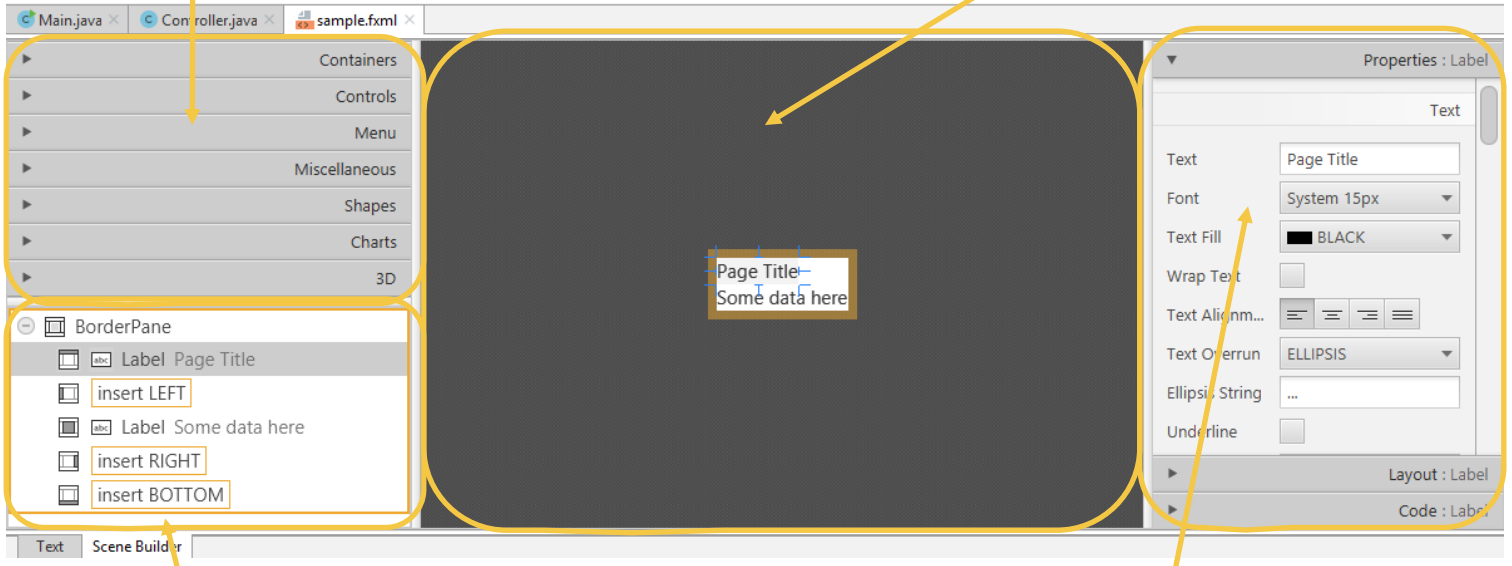
Scene Builder Tool



Available containers and controls

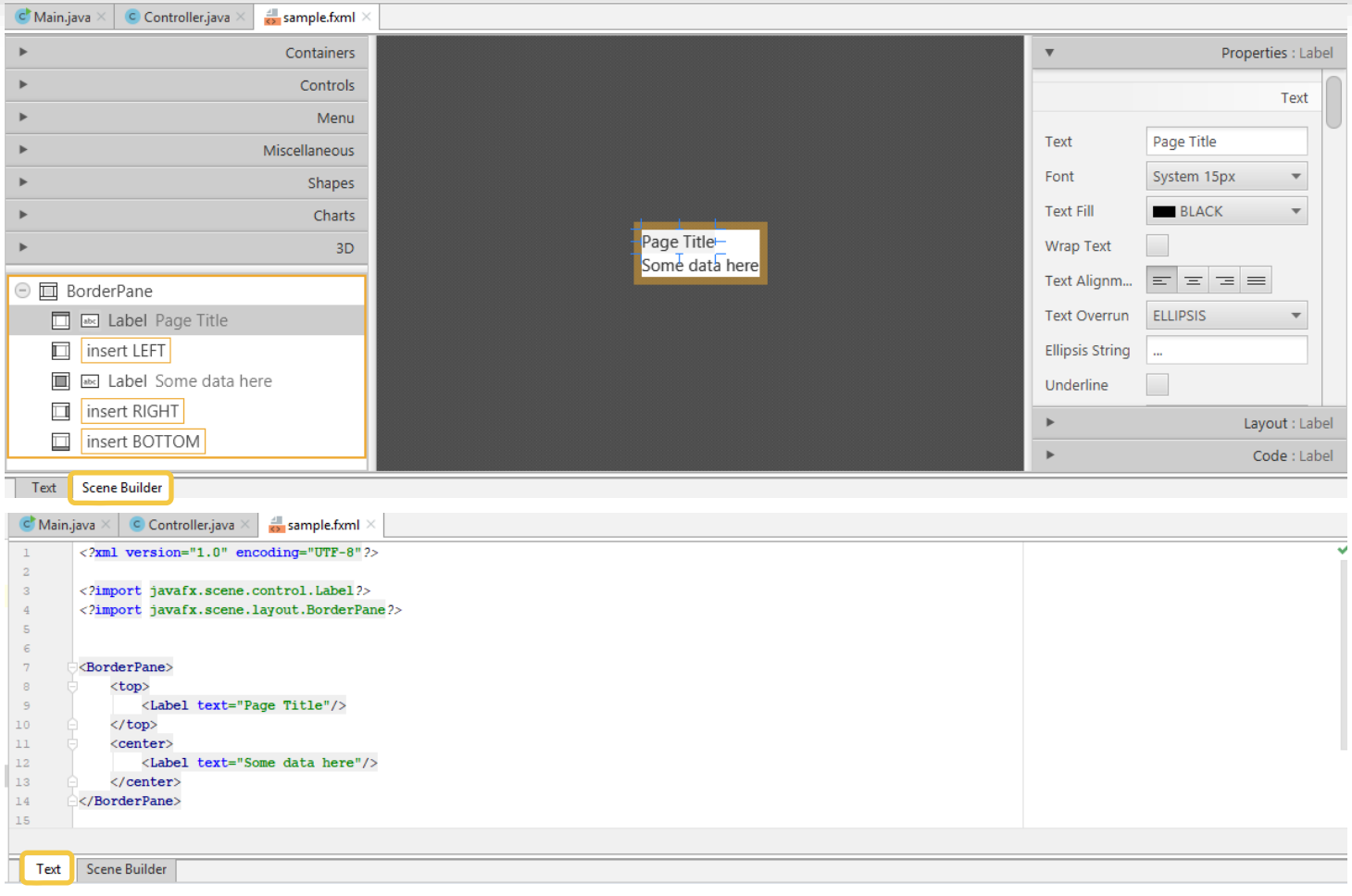
Scene Builder Tool

Scene protopype



Scene hierarchy

Control properties, layout and event handling


Main.java
Controller.java
sample.fxml
Containers
Controls
Menu
Miscellaneous
Shapes
Charts
3D
BorderPane
Label Page Title
insert LEFT
Label Some data here
insert RIGHT
insert BOTTOM
Page Title
Some data here
Properties : Label
Text
Text
Page Title
Font
System 15px
Text Fill
BLACK
Wrap Text
Text Alignm...
Text Overrun
ELLIPSIS
Ellipsis String
...
Underline
Layout : Label
Code : Label
Text
Scene Builder
Main.java
Controller.java
sample.fxml
1 <?xml version="1.0" encoding="UTF-8"?>
2
3 <?import javafx.scene.control.Label?>
4 <?import javafx.scene.layout.BorderPane?>
5
6
7 <BorderPane>
8 <top>
9 <Label text="Page Title"/>
10 </top>
11
12 <Label text="Some data here"/>
13
14 </BorderPane>
15
Text
Scene Builder

“Existen tres posibles respuestas a un diseño:
Si, No y WOW!.
WOW es a lo que hay que aspirar.“

Milton Glaser, diseñador gráfico

La Salle
Gràcia