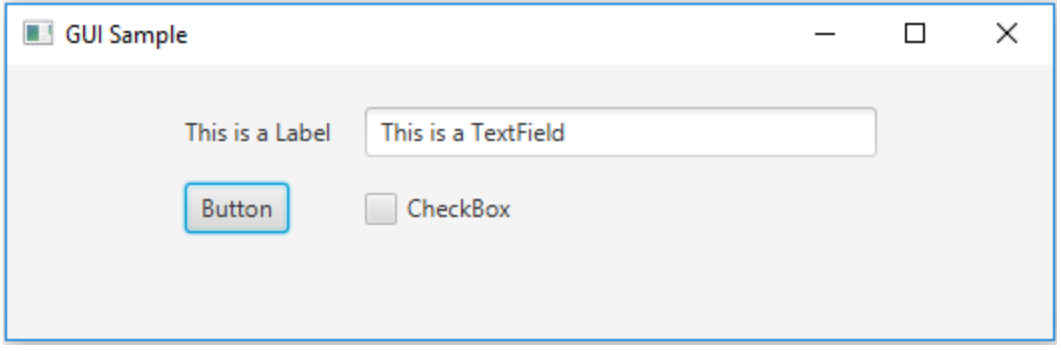
Graphical User Interface

1. Introduction to GUI
2. Java GUIs
3. JavaFX

Introduction to GUI

- GUI stands for Graphical User Interface.
- It presents an easy-to-use visual display to the user.
- It is made up of graphical components.
- Components are set into containers.
- Components are organized following a layout.

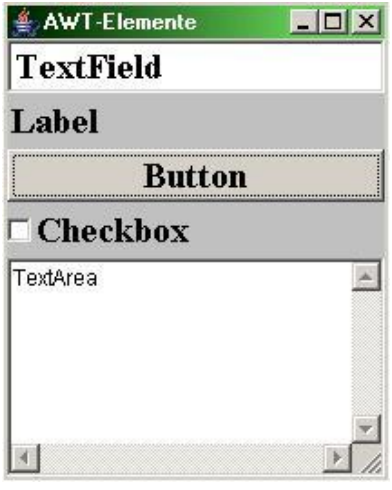


Java GUIs

JavaFX is an open source, next generation client application platform for desktop, mobile and embedded systems built on Java. It is a collaborative effort by many individuals and companies with the goal of producing a modern, efficient, and fully featured toolkit for developing rich client applications.

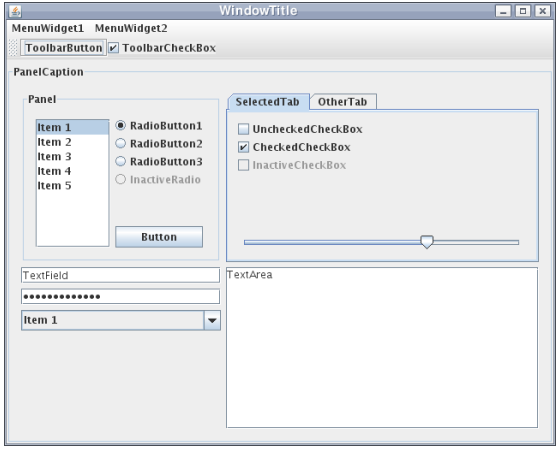
AWT

- The original set of classes, dating back to JDK 1.0.
- Very limited for nowadays user expectations.



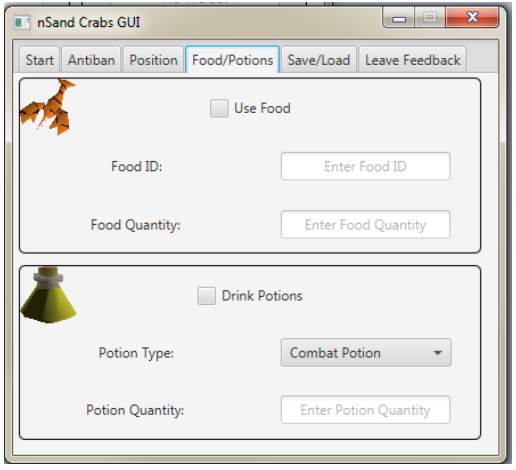
Swing

- Fixes some difficulties of AWT.
- Controls look equal in all OP but currently look old.
- Swing keeps AWT dependency in some classes.



JavaFX

- The newest set of GUI classes in Oracle standard Java.
- Faster, sophisticated and aesthetically pleasant UIs.
- Better sound, image, video or web content integration.



JavaFX

JavaFX main packages

Packages

Package	Description
javafx.application	Provides the application life-cycle classes.
javafx.animation	Provides the set of classes for ease of use transition based animations.
javafx.event	Provides basic framework for FX events, their delivery and handling.
javafx.css	Provides API for making properties styleable via CSS and for supporting pseudo-class state.
javafx.geometry	Provides the set of 2D classes for defining and performing operations on objects related to two-dimensional geometry.
javafx.stage	Provides the top-level container classes for JavaFX content.
javafx.scene	Provides the core set of base classes for the JavaFX Scene Graph API.
javafx.scene.canvas	Provides the set of classes for canvas, an immediate mode style of rendering API.
javafx.scene.chart	The JavaFX User Interface provides a set of chart components that are a very convenient way for data visualization.

https://docs.oracle.com/javase/8/javafx/api/toc.htm

JavaFX main packages

Packages

Package	Description
javafx.application	Provides the application life-cycle classes.
javafx.animation	Provides the set of classes for ease of use transition based animations.
javafx.event	Provides basic framework for FX events, their delivery and handling.
javafx.css	Provides API for making properties styleable via CSS and for supporting pseudo-class state.
javafx.geometry	Provides the set of 2D classes for defining and performing operations on objects related to two-dimensional geometry.
javafx.stage	Provides the top-level container classes for JavaFX content.
javafx.scene	Provides the core set of base classes for the JavaFX Scene Graph API.
javafx.scene.canvas	Provides the set of classes for canvas, an immediate mode style of rendering API.
javafx.scene.chart	The JavaFX User Interface provides a set of chart components that are a very convenient way for data visualization.

https://docs.oracle.com/javase/8/javafx/api/toc.htm

JavaFX main packages

Packages

Package	Description
javafx.application	Provides the application life-cycle classes.
javafx.animation	Provides the set of classes for ease of use transition based animations.
javafx.event	Provides basic framework for FX events, their delivery and handling.
javafx.css	Provides API for making properties styleable via CSS and for supporting pseudo-class state.
javafx.geometry	Provides the set of 2D classes for defining and performing operations on objects related to two-dimensional geometry.
javafx.stage	Provides the top-level container classes for JavaFX content.
javafx.scene	Provides the core set of base classes for the JavaFX Scene Graph API.
javafx.scene.canvas	Provides the set of classes for canvas, an immediate mode style of rendering API.
javafx.scene.chart	The JavaFX User Interface provides a set of chart components that are a very convenient way for data visualization.

https://docs.oracle.com/javase/8/javafx/api/toc.htm

JavaFX main packages

Packages

Package	Description
javafx.application	Provides the application life-cycle classes.
javafx.animation	Provides the set of classes for ease of use transition based animations.
javafx.event	Provides basic framework for FX events, their delivery and handling.
javafx.css	Provides API for making properties styleable via CSS and for supporting pseudo-class state.
javafx.geometry	Provides the set of 2D classes for defining and performing operations on objects related to two-dimensional geometry.
javafx.stage	Provides the top-level container classes for JavaFX content.
javafx.scene	Provides the core set of base classes for the JavaFX Scene Graph API.
javafx.scene.canvas	Provides the set of classes for canvas, an immediate mode style of rendering API.
javafx.scene.chart	The JavaFX User Interface provides a set of chart components that are a very convenient way for data visualization.

https://docs.oracle.com/javase/8/javafx/api/toc.htm

JavaFX main packages

Packages

Package	Description
javafx.application	Provides the application life-cycle classes.
javafx.animation	Provides the set of classes for ease of use transition based animations.
javafx.event	Provides basic framework for FX events, their delivery and handling.
javafx.css	Provides API for making properties styleable via CSS and for supporting pseudo-class state.
javafx.geometry	Provides the set of 2D classes for defining and performing operations on objects related to two-dimensional geometry.
javafx.stage	Provides the top-level container classes for JavaFX content.
javafx.scene	Provides the core set of base classes for the JavaFX Scene Graph API.
javafx.scene.canvas	Provides the set of classes for canvas, an immediate mode style of rendering API.
javafx.scene.chart	The JavaFX User Interface provides a set of chart components that are a very convenient way for data visualization.

https://docs.oracle.com/javase/8/javafx/api/toc.htm

JavaFX main packages

Packages

Package	Description
javafx.application	Provides the application life-cycle classes.
javafx.animation	Provides the set of classes for ease of use transition based animations.
javafx.event	Provides basic framework for FX events, their delivery and handling.
javafx.css	Provides API for making properties styleable via CSS and for supporting pseudo-class state.
javafx.geometry	Provides the set of 2D classes for defining and performing operations on objects related to two-dimensional geometry.
javafx.stage	Provides the top-level container classes for JavaFX content.
javafx.scene	Provides the core set of base classes for the JavaFX Scene Graph API.
javafx.scene.canvas	Provides the set of classes for canvas, an immediate mode style of rendering API.
javafx.scene.chart	The JavaFX User Interface provides a set of chart components that are a very convenient way for data visualization.

https://docs.oracle.com/javase/8/javafx/api/toc.htm

JavaFX main packages

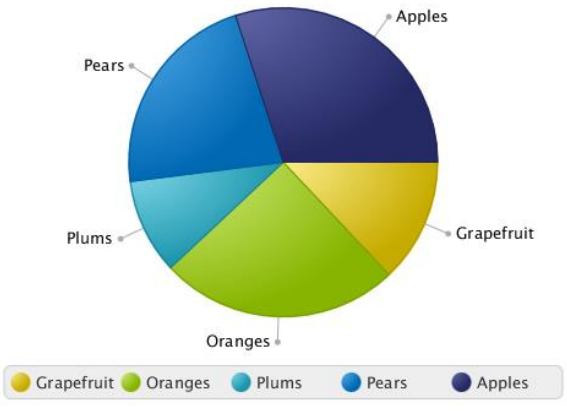
Packages

Package	Description
javafx.application	Provides the application life-cycle classes.
javafx.animation	Provides the set of classes for ease of use transition based animations.
javafx.event	Provides basic framework for FX events, their delivery and handling.
javafx.css	Provides API for making properties styleable via CSS and for supporting pseudo-class state.
javafx.geometry	Provides the set of 2D classes for defining and performing operations on objects related to two-dimensional geometry.
javafx.stage	Provides the top-level container classes for JavaFX content.
javafx.scene	Provides the core set of base classes for the JavaFX Scene Graph API.
javafx.scene.canvas	Provides the set of classes for canvas, an immediate mode style of rendering API.
javafx.scene.chart	The JavaFX User Interface provides a set of chart components that are a very convenient way for data visualization.

https://docs.oracle.com/javase/8/javafx/api/toc.htm

Nodes

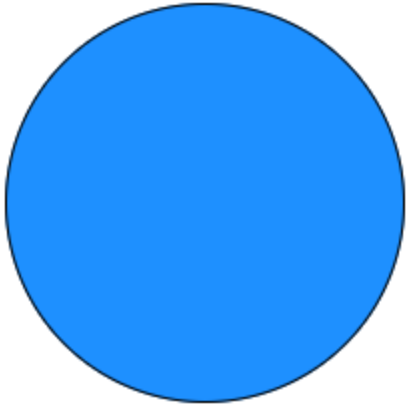
javafx.scene.Node



Node

Button

This is a sample toolpit

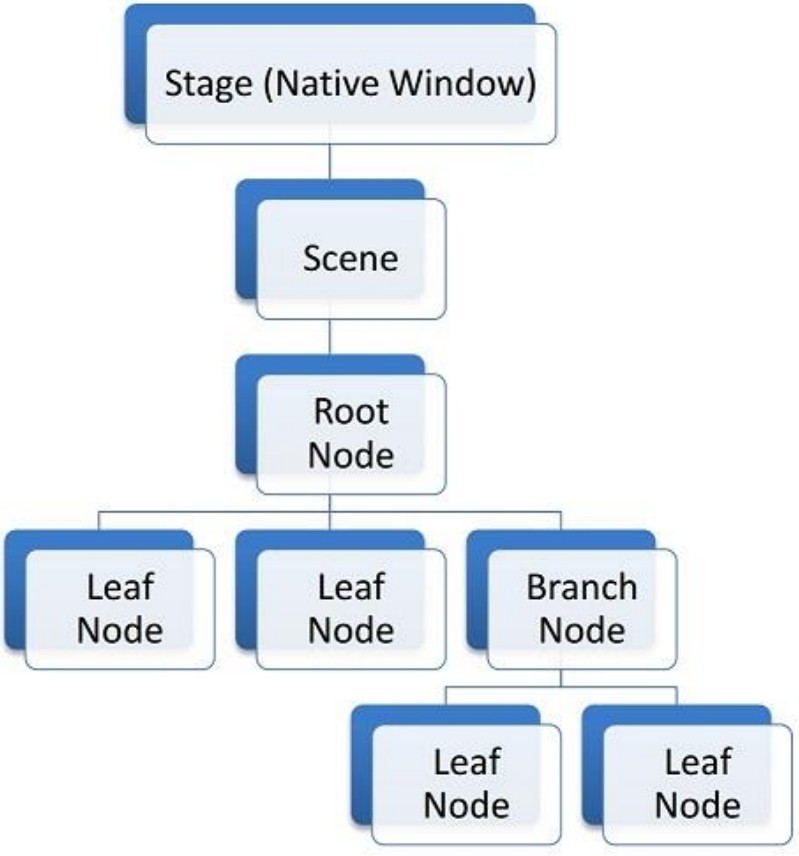


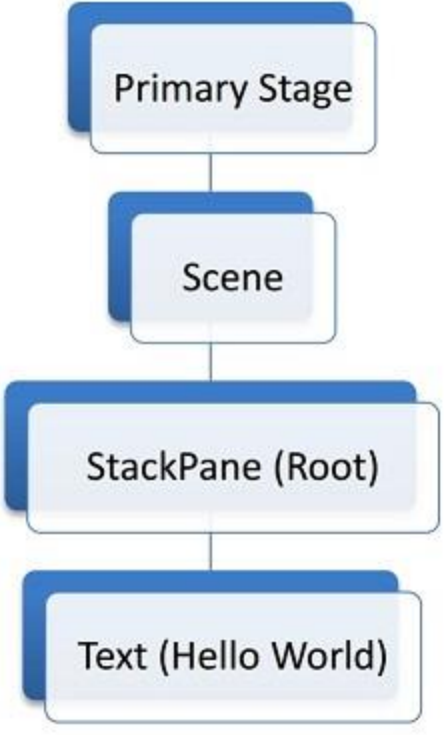
CheckBox

GridPane (0 x 0)

ButtonBar (0, 0)

JavaFX Application



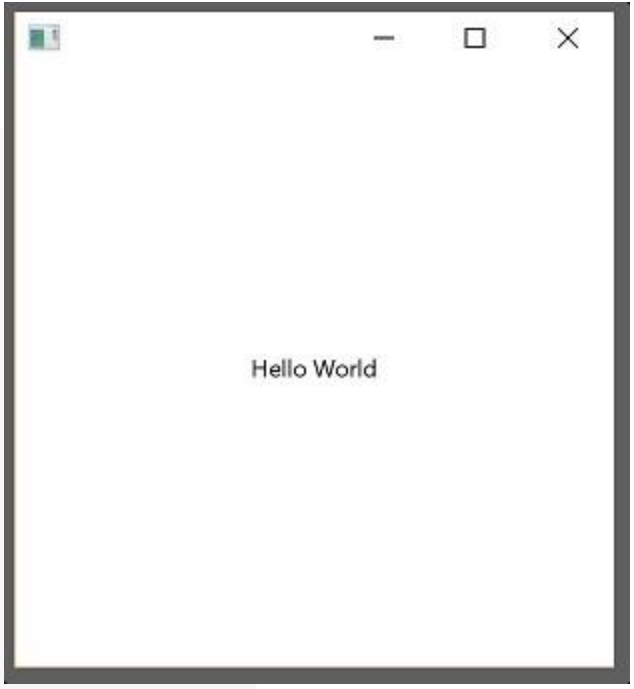


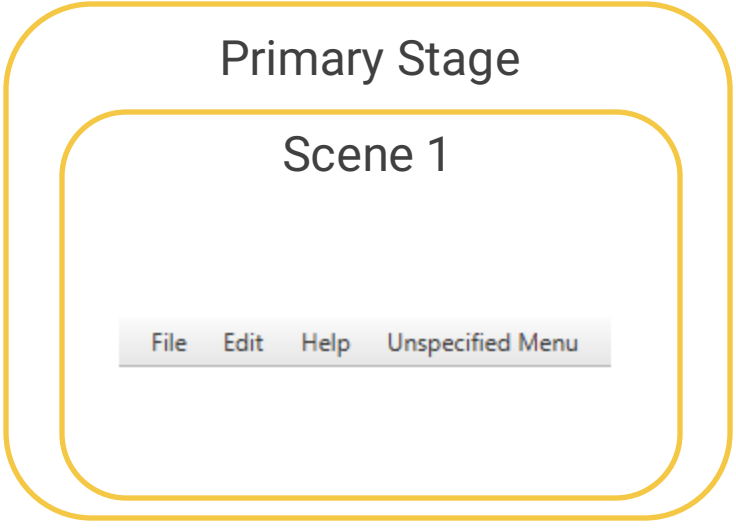
Exemple “Hello World” Demo

JavaFX Application “Hello World” Demo

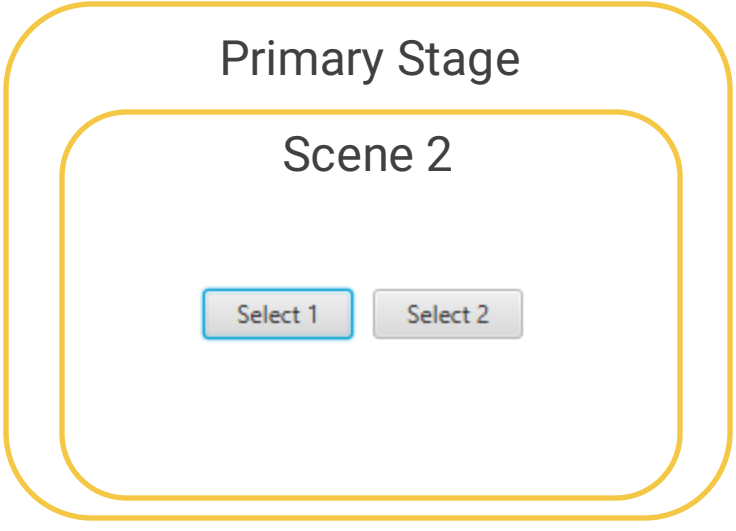
TransformApp.class

```
public class TransformApp extends Application {

    private Parent createContent() {
        Rectangle box = new Rectangle(100, 50, Color.BLUE);

        transform(box);

        return new Pane(box);
    }

    private void transform(Rectangle box) {
        // we will apply transformations here
    }

    @Override
    public void start(Stage stage) throws Exception {
        stage.setScene(new Scene(createContent(), 300, 300, Color.GRAY));
        stage.show();
    }

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

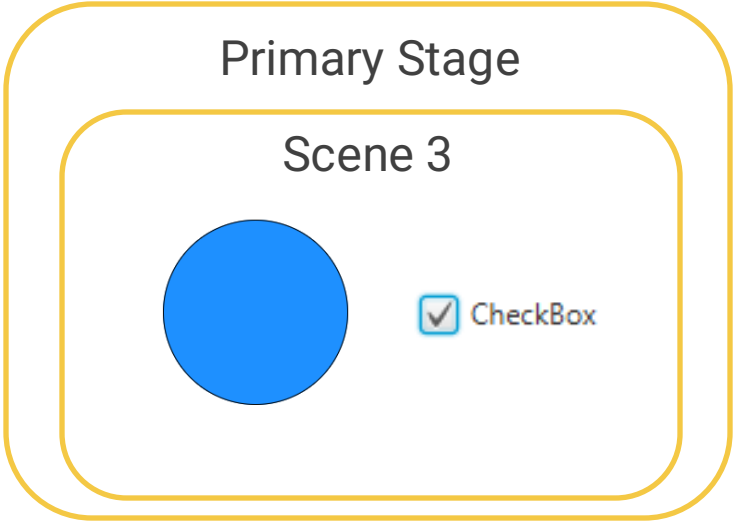


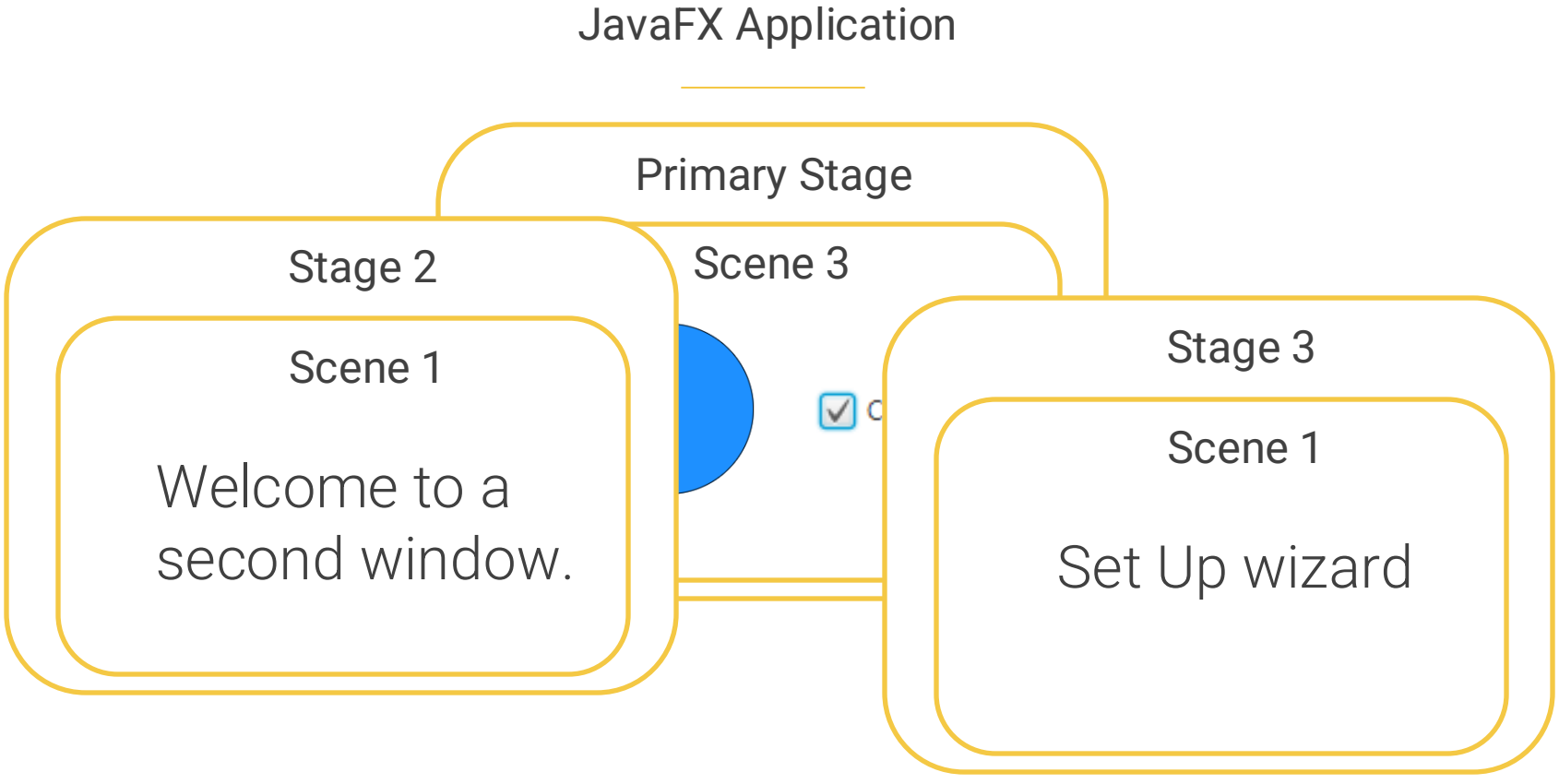

Primary Stage
Scene 1
File
Edit
Help
Unspecified Menu

JavaFX Application

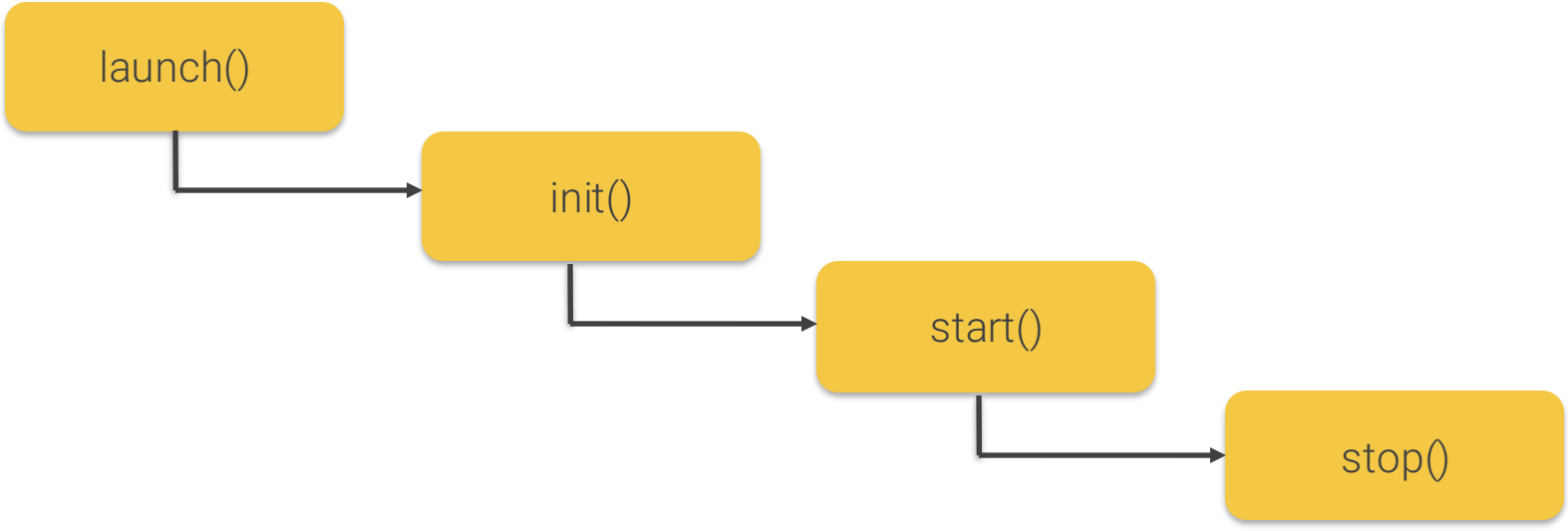


JavaFX Application



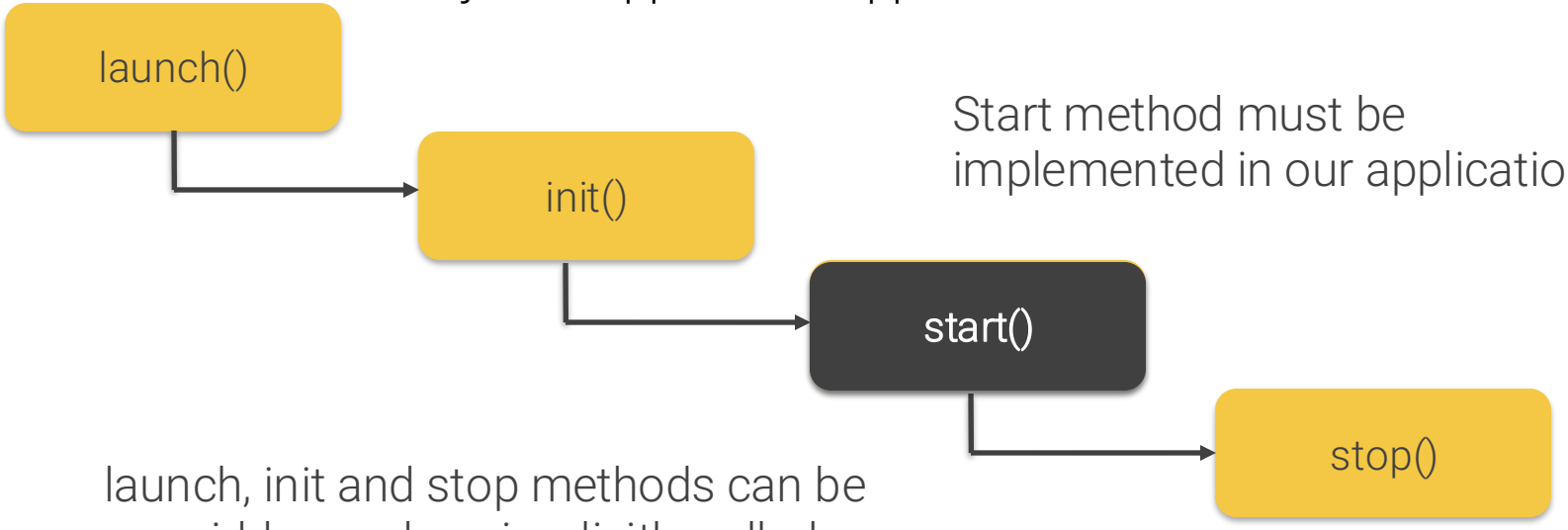

JavaFX Application
Primary Stage
Scene 3
Stage 2
Scene 1
Welcome to a second window.
Stage 3
Scene 1
Set Up wizard

JavaFX Application Lifecyle



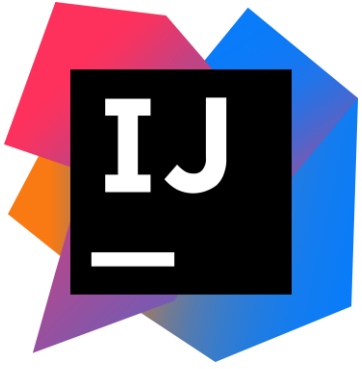
JavaFX Application Lifecyle

javafx.application.Application



Start method must be implemented in our application

launch, init and stop methods can be overridden and are implicitly called



Application DEMO

“Los símbolos sencillos y económicos funcionan mejor y de forma más universal que los cargados de detalles.“

Susan Kare, diseñadora gráfica

La Salle
Gràcia