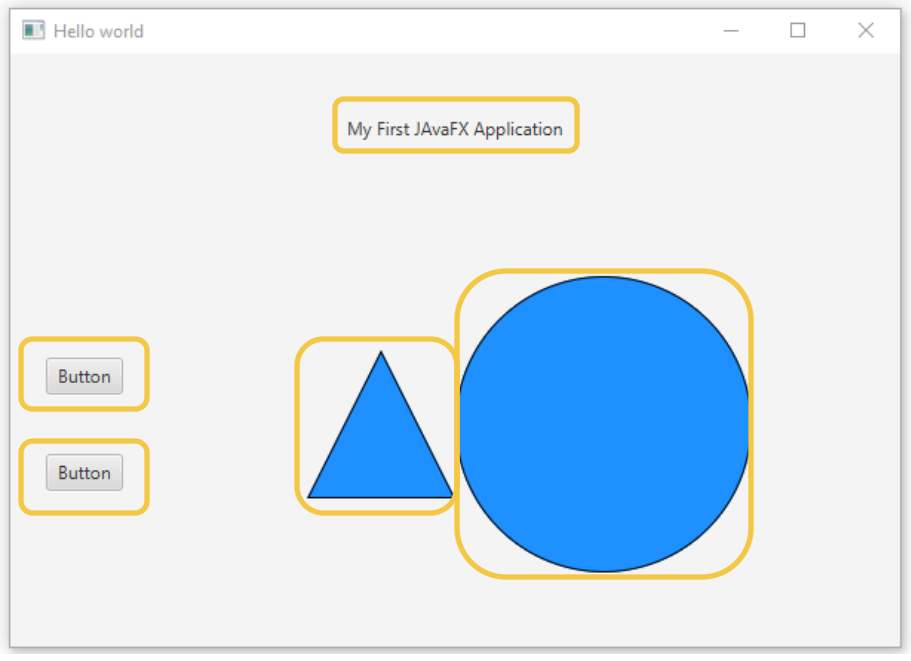
GUI Events

1. GUI events
2. Event handling
3. Demo

GUI events

Main aspects of GUI

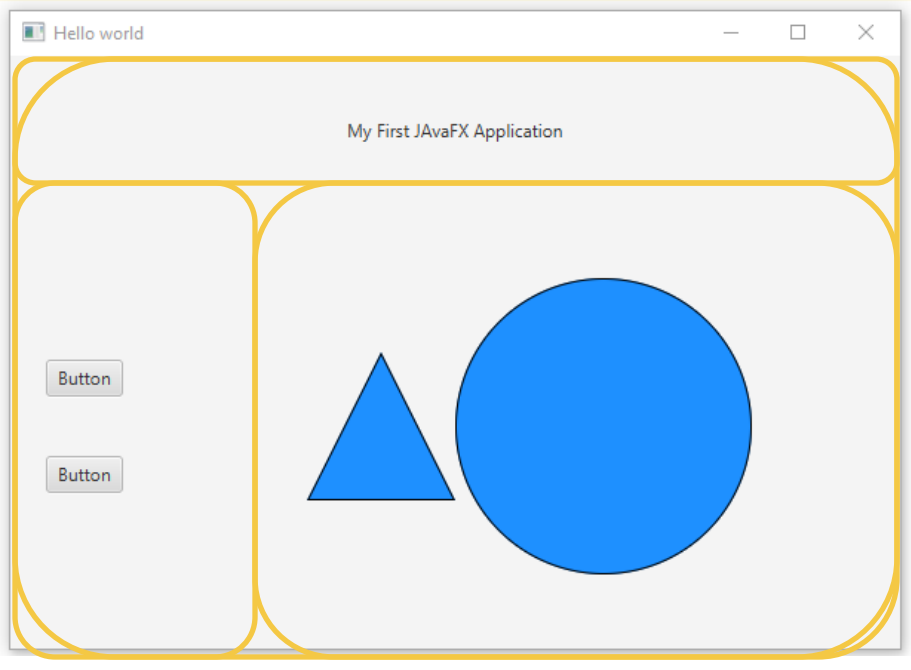
Controls



Main aspects of GUI

Controls

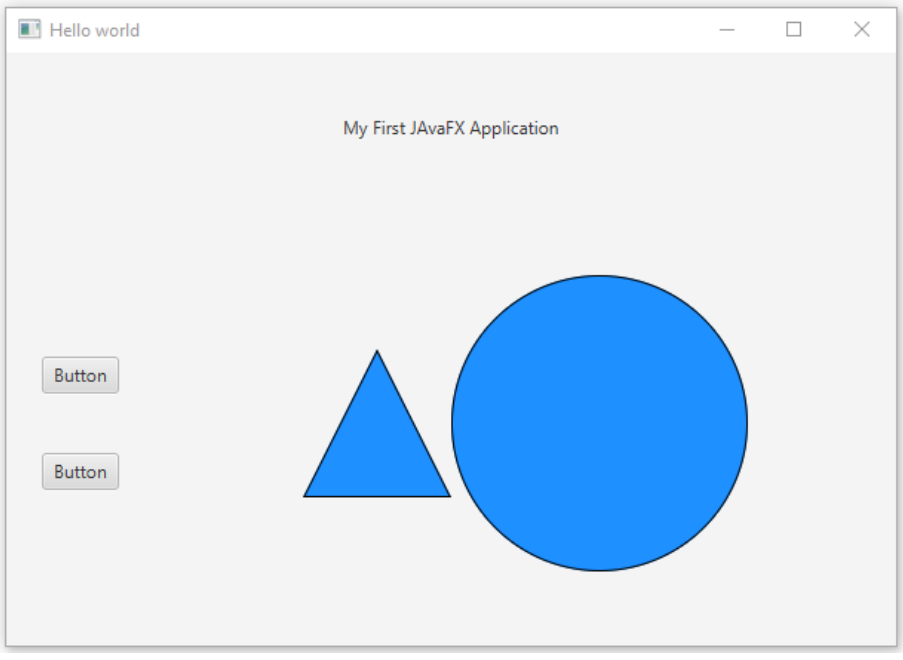
Layouts



Main aspects of GUI

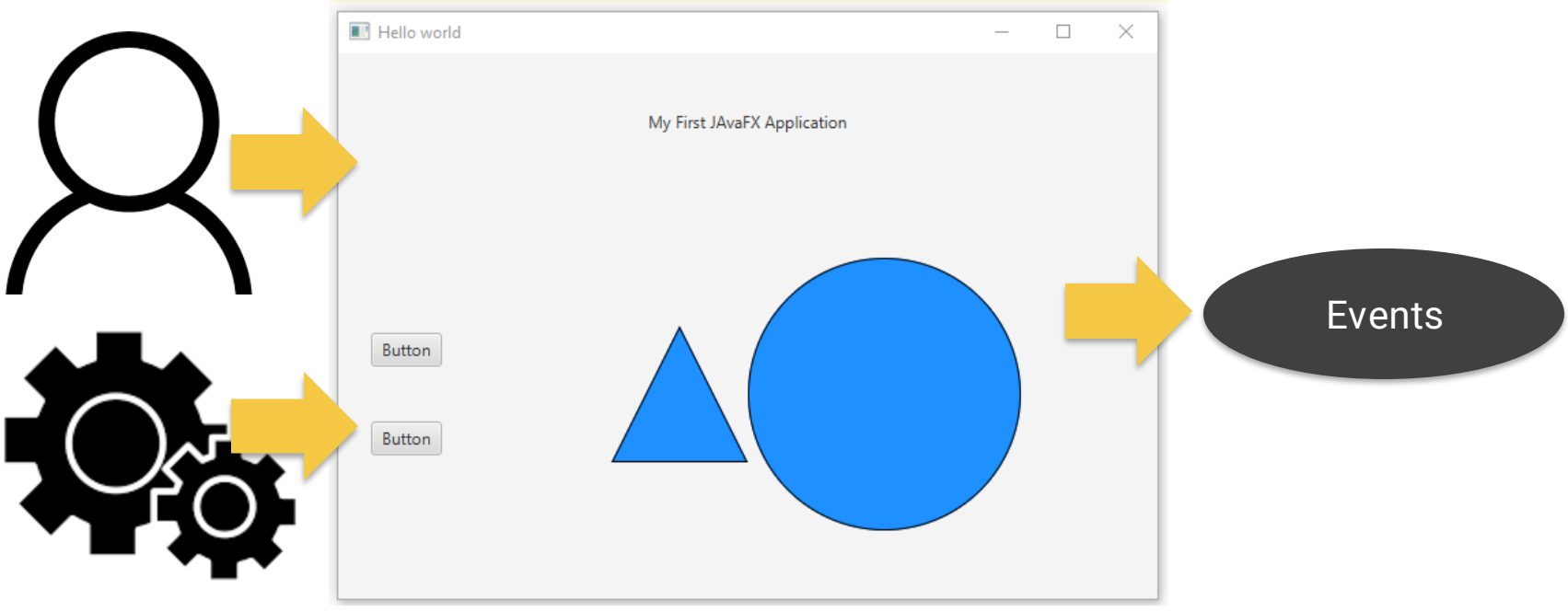
Controls

Layouts



Events

An event is an action or occurrence recognized by the application.



Event types

Foreground events

- Require the direct user interaction.
- Consequences of interplay with the graphical components.
- Clicking on a button, moving the mouse, key press,...

Background events

- Require the interacion of a backend user
- Not directly generated by user interplay with the graphical components.
- OS interruptions, timer expiry, operation completion,...

Event types

Foreground events

- Require the direct user interaction.
- Consequences of interplay with the graphical components.
- Clicking on a button, moving the mouse, key press,...

Background events

- Require the interacion of a backend user
- Not directly generated by user interplay with the graphical components.
- OS interruptions, timer expiry, operation completion,...

Events in JavaFX

javafx.event.Event

- MouseEvent
- WindowEvent
- KeyEvent
- DragEvent

Events in JavaFX

javafx.event.Event

`mouse clicked` `mouse pressed`

MouseEvent

`mouse entered target`

`window hiding` `window shown`

WindowEvent

`window hidden`

`key pressed` `key released`

KeyEvent

`key typed`

`drag entered` `drag dropped`

DragEvent

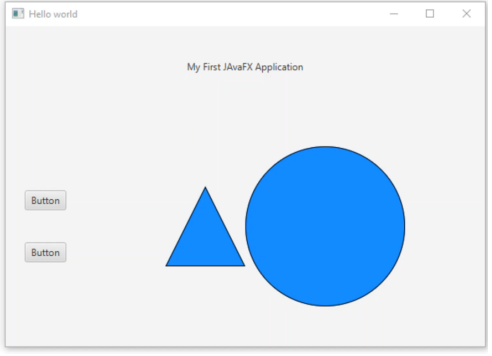
`drag entered target`

JavaFX event properties

Source – The source from which the event is generated.

Type – Type of the occurred event

Target – The node on which an event occurred



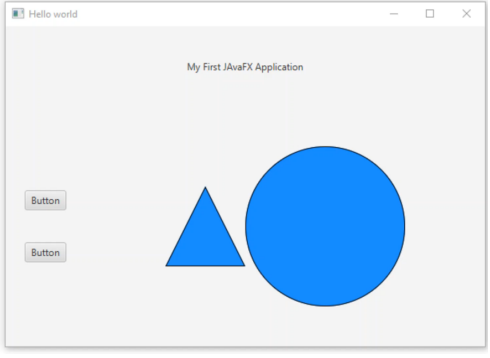
JavaFX event properties

Source – The source from which the event is generated.

Type – Type of the occurred event

Target – The node on which an event occurred

Source
mouse

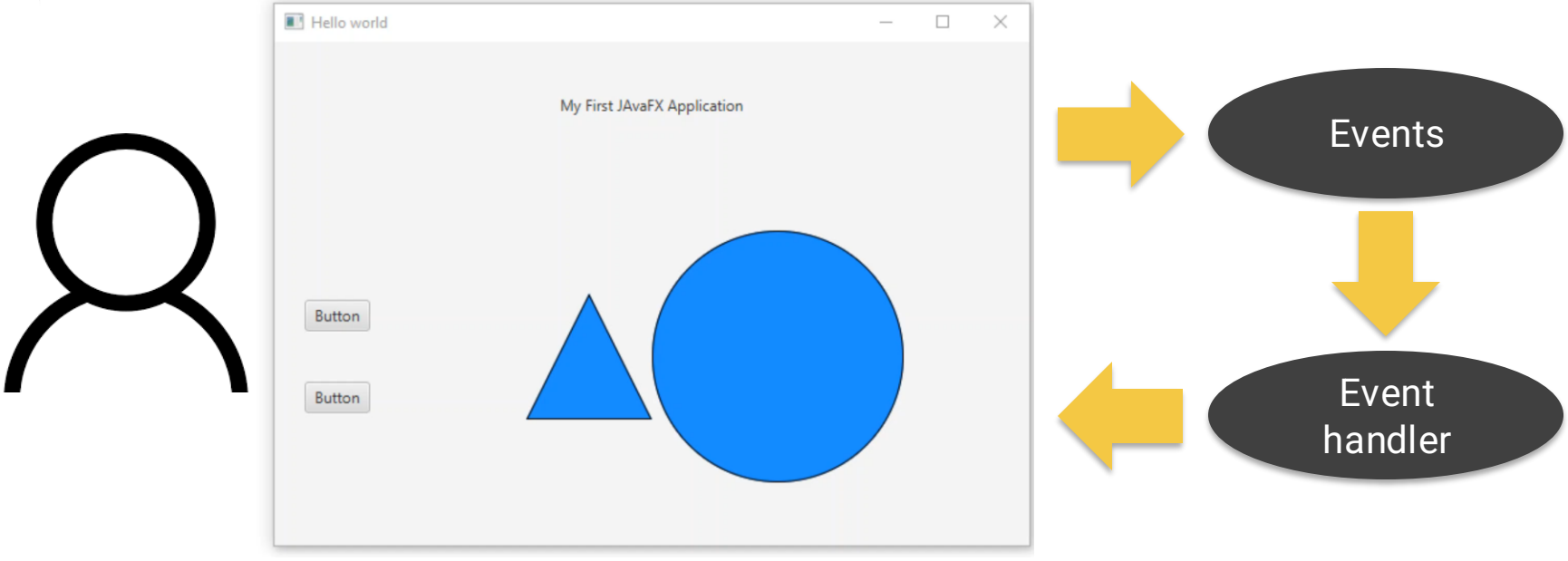


Type
mouse click

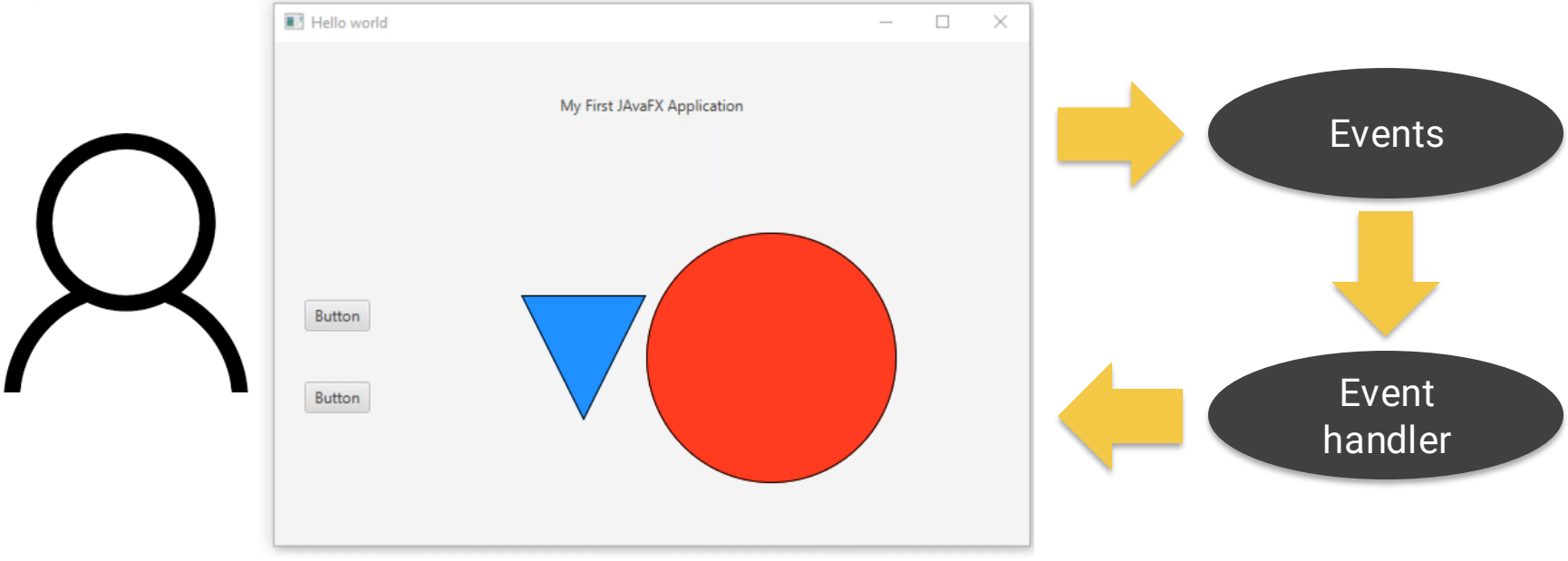
Target
the button

Event Handling

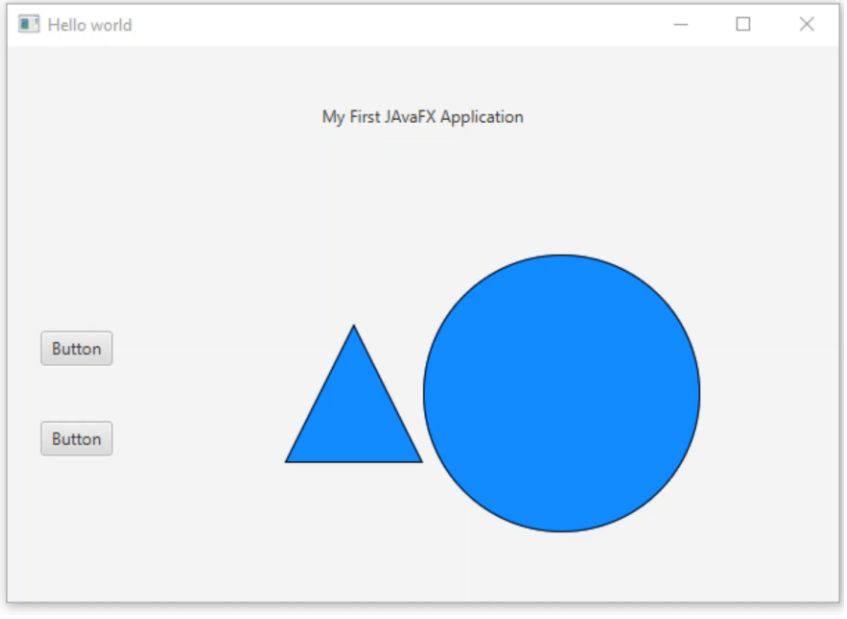
2. Event handling



2. Event handling



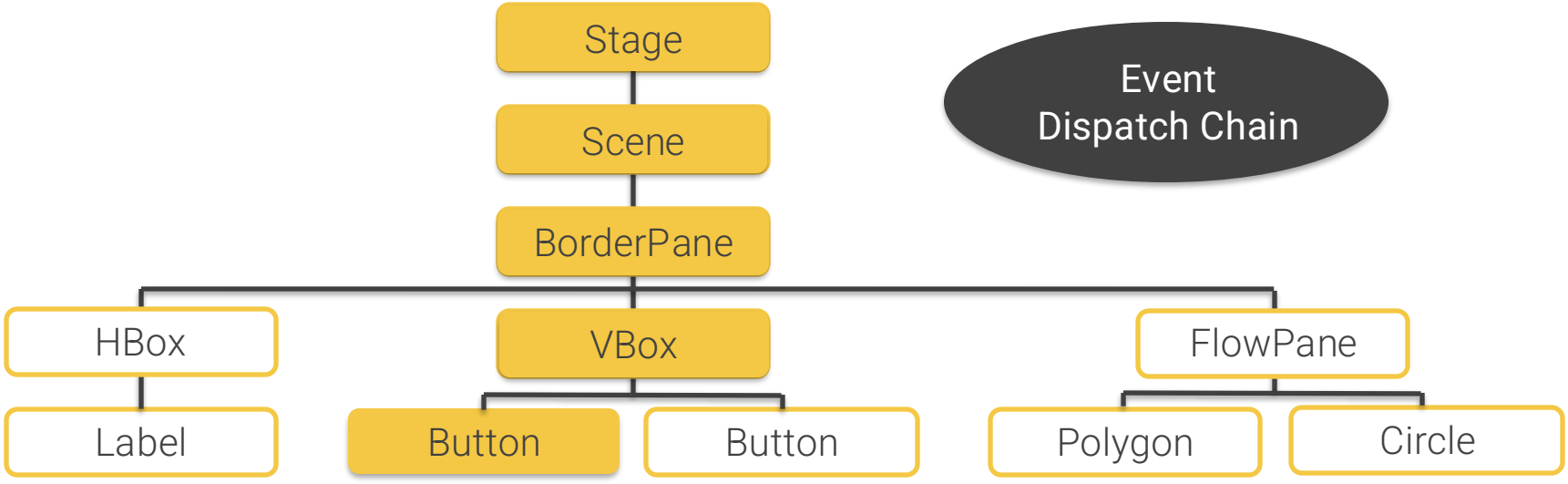
Event Delivery Process: Target selection



Button

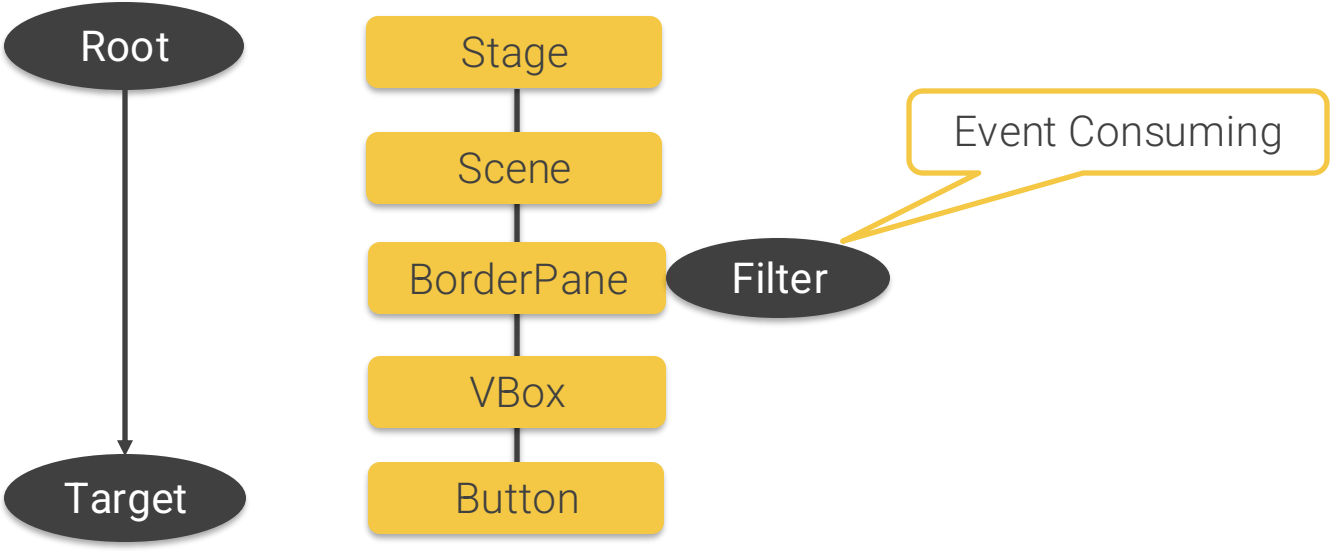
The system determines which node is the target based on internal rules

Event Delivery Process: Route construction



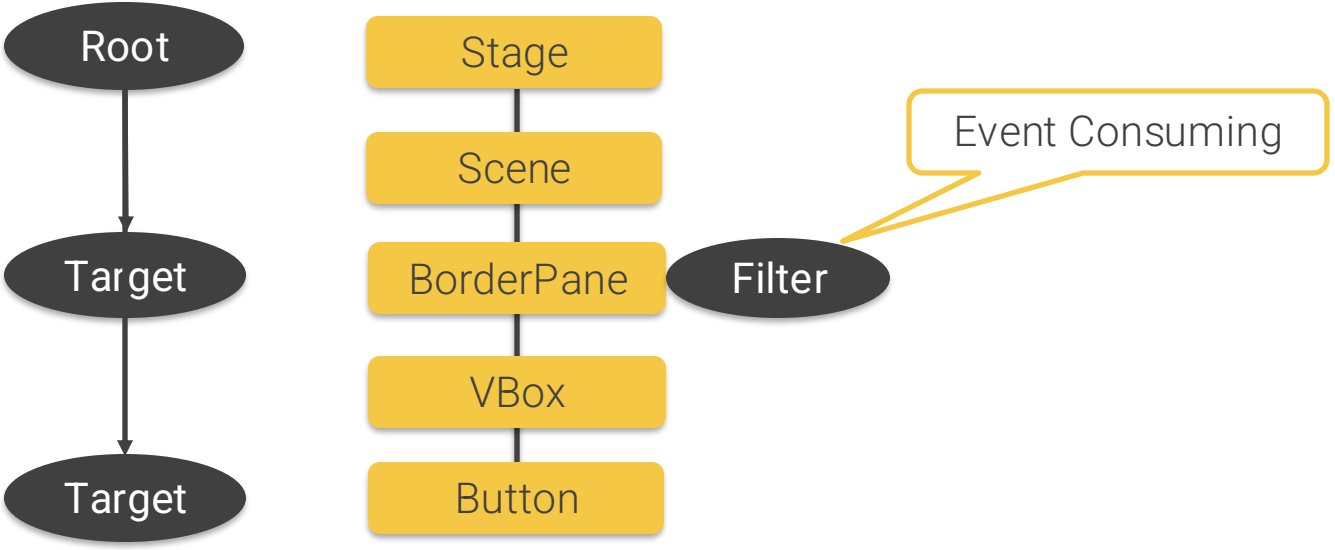
Construction of the event dispatch chain

Event Delivery Process: Event Capturing



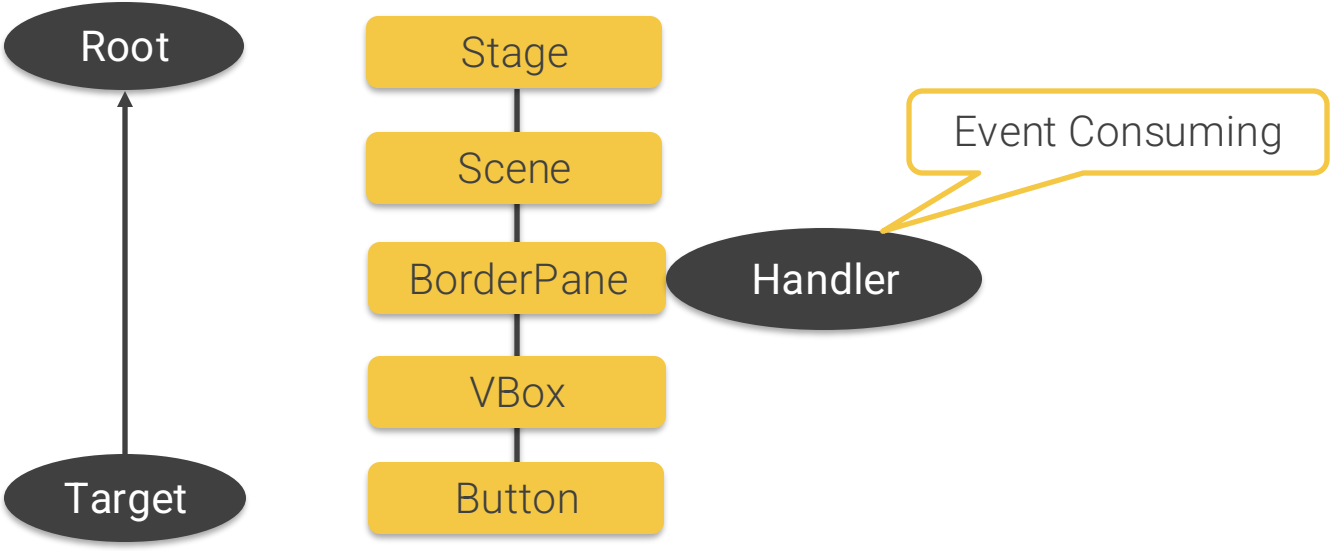
The event is dispatched from the Root to the Target

Event Delivery Process: Event Capturing



If a filter consumes the event the chain stops and that node becomes the target.

Event Delivery Process: Event Building

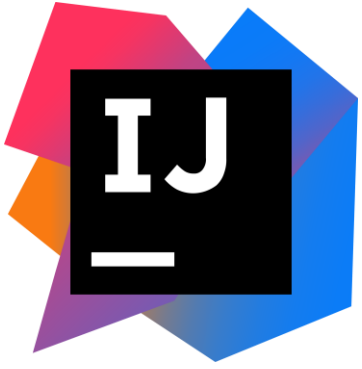


The event travels from the Target to the Root

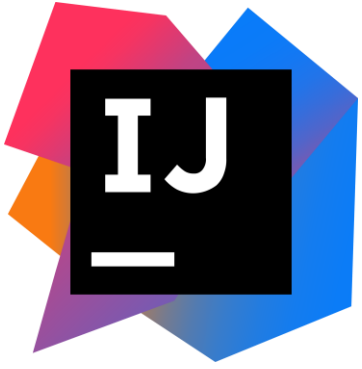
Event handling methods

- Event handling using filter
- Event handling using handler
- Event handling using convenience method
- Event handling using method reference

Demo



Handling events in a JavaFX Project with FXML DEMO



Handling events in a JavaFX Project without FXML DEMO

Event handling using filter

Main.java

```
                                ...
EventHandler<MouseEvent> ehChangeCircleColor = new EventHandler<MouseEvent>() {
            @Override
            public void handle(MouseEvent event) {
                circle.setFill(Color.RED);
            }
        };

 button1.addEventFilter(MouseEvent.MOUSE_CLICKED,ehChangeCircleColor);
                                ...
```

Event handling using handler

Main.java

```
                                ...
EventHandler<MouseEvent> ehChangeCircleColor = new EventHandler<MouseEvent>() {
            @Override
            public void handle(MouseEvent event) {
                circle.setFill(Color.RED);
            }
        };

 button1.addEventHandler(MouseEvent.MOUSE_CLICKED,ehChangeCircleColor);
                                ...
```

Event handling using convenience method

Main.java

```
                                ...
button1.setOnMouseClicked(new EventHandler<MouseEvent>() {
            @Override
            public void handle(MouseEvent event) {
                circle.setFill(Color.RED);
            }
        });
                                ...
```

Event handling using method reference

Controller.java

```
                                ...
@FXML
   public void changeCircleColor(Event e){
       circle.setFill(Color.RED);
   }
                                ...
```

myScene.fxml

```
                                ...
<Button mnemonicParsing="false" onMouseClicked="#changeCircleColor" text="Button" />
                                ...
```

“Vale más hacer y arrepentirse, que no hacer y arrepentirse.“

Nicolás Maquiavelo, filósofo