

Event Listeners

Week 6 Presentation 4



There are different rules for the system to find the target widget that has the focus for a particular events. EG cursor position for mouse events, button presses, etc. Consider issues such as if an element is hidden by another it's the one on top and it is considered to be the target.

Similar to
Exceptions:
“bubbles up”

Event
Handlers are
like “catch”



Events and Change Listeners

1. Determine the Event Target

EventTarget interface

Window

Scene

Node



Events and Change Listeners

.setOnSomeAction() methods for nodes:

“KeyPressed”

“KeyReleased”

“KeyTyped”

“MouseClicked”

“MouseExited”

many more...



Events and Change Listeners

.setOnAction() method for
Buttons
RadioButtons
CheckBoxes



Events and Change Listeners

Use Lambda Expressions:

```
Button btn = new Button();  
btn.setText("Say 'Hi'");  
btn.setOnAction(e -> System.out.println("Hi!"));
```



```
public class HelloWorldFX extends Application {
```

```
    public void start(Stage stage) {
```

```
        Label message = new Label("First FX Application!");  
        message.setFont( new Font(40) );
```

```
        → Button helloButton = new Button("Say Hello");  
          helloButton.setOnAction( e -> message.setText("Hello World!") ); ←  
        Button goodbyeButton = new Button("Say Goodbye");  
        goodbyeButton.setOnAction( e -> message.setText("Goodbye!!") );  
        → Button quitButton = new Button("Quit");  
          quitButton.setOnAction( e -> Platform.exit() );
```

```
        HBox buttonBar = new HBox( 20, helloButton, goodbyeButton, quitButton );  
        buttonBar.setAlignment(Pos.CENTER);  
        BorderPane root = new BorderPane();  
        root.setCenter(message);  
        root.setBottom(buttonBar);
```

```
        Scene scene = new Scene(root, 450, 200);  
        stage.setScene(scene);  
        stage.setTitle("JavaFX Test");  
        stage.show();
```

```
    } // end start();
```

```
    public static void main(String[] args) {  
        launch(args); // Run this Application.  
    }
```

```
} // end class HelloWorldFX
```



Events and Change Listeners

Use Lambda Expressions:

```
helloButton.setOnAction( e -> message.setText("Hello World!") );
```



Events and Change Listeners

Use Lambda Expressions:

```
Button helloButton = new Button("Say Hello");  
helloButton.setOnAction( e -> message.setText("Hello World!") );  
Button goodbyeButton = new Button("Say Goodbye");  
goodbyeButton.setOnAction( e -> message.setText("Goodbye!!") );  
Button quitButton = new Button("Quit");  
quitButton.setOnAction( e -> Platform.exit() );
```



Exercise

- Download the HelloWorldFX Application that was discussed in this presentation from blackboard resources section or the Java8 online text
- Run the program in your IDE
- Test some changes and familiarize yourself with the methods used (e.g. change the text in the action listener, remove the the quit button action, etc)





You can also define Event Filters to intercept (block) some events or override them

Event Filters are activated before event handlers.

When an event occurs it is first passed through the chain to the target (for example if you have a scene with a button, or even a shape) the event would normally bubble up from the target through a pane -> scene -> stage and be acted on by an event handler if there are any.

Before the event is passed to **any** event handlers, it is first passed to any event filters that exist in the pane, scene, stage or target would be applied first and could prevent the child nodes from receiving the event (block) and possibly do an alternative action (override).



Events and Change Listeners

Instead of `.setOnxxxx()` methods, you can use `addEventHandler()`:

```
button.addEventHandler(MouseEvent.MOUSE_ENTERED, e->(...) );
```

