

Reacting To User Operations In the UI



Buddha Jyothiprasad

AUTHOR

@prbuddha

<http://controls-space.info>



Outline



Event

- Input Events
- Event Filters
- Event Handlers
- Action Events

Implement add/update task

Alert Dialog



Event



Event

Represents an occurrence of something of interest to the application, such as a mouse being moved or a key being pressed.



Click
KeyEvent
Drag
Rotate
Zoom
Touch

Do-It!!!

Priority	Description	Progress
High	Complete Design Document	10%
Medium	Update Class Diagram	0%
Low	Fix Bug 245232	0%

High Complete Design Document Update

Progress 10 Completed Cancel

Close
Minimize
Drag

Click
KeyEvent

Click
KeyEvent

Click

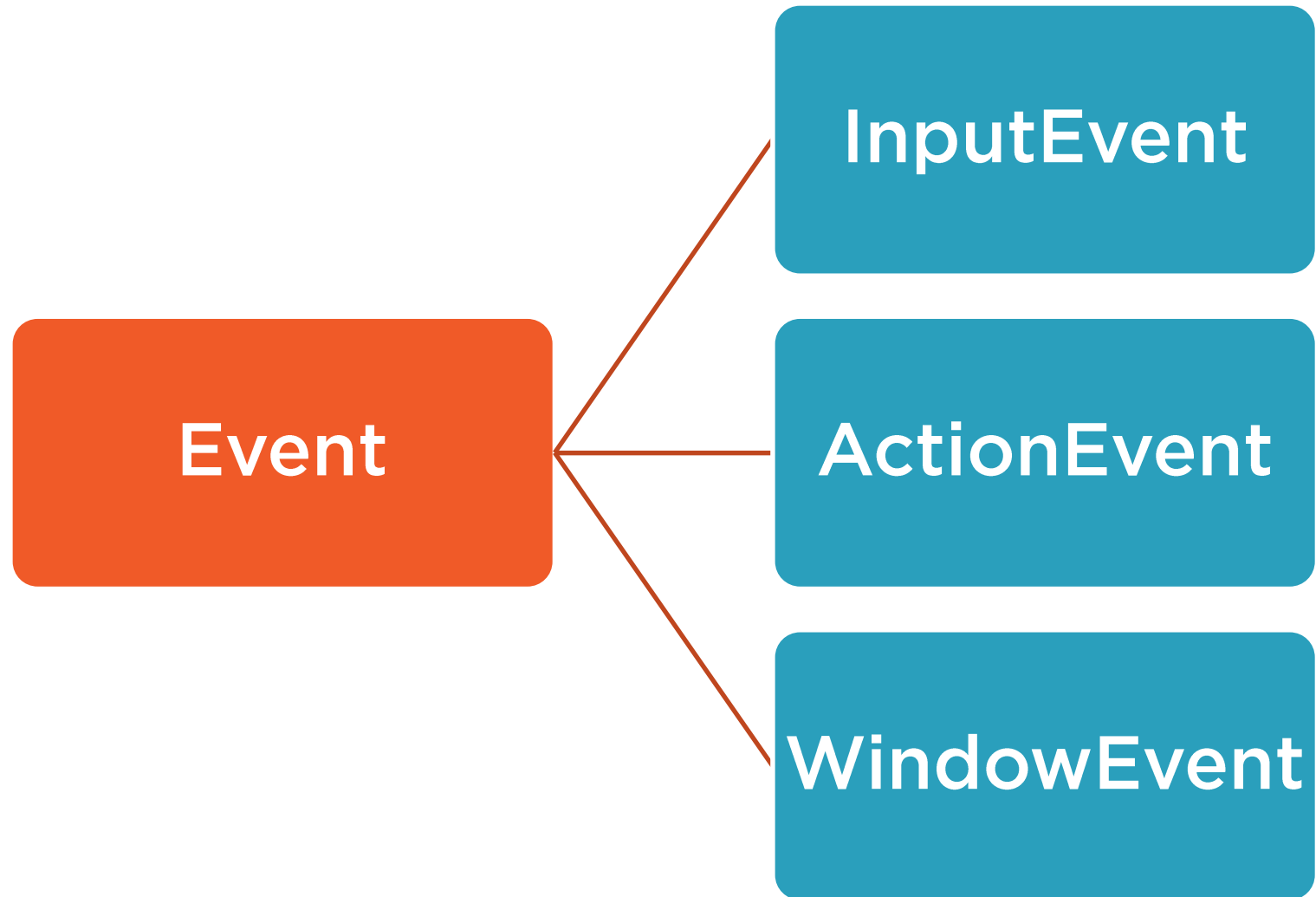
Click
KeyEvent

Click

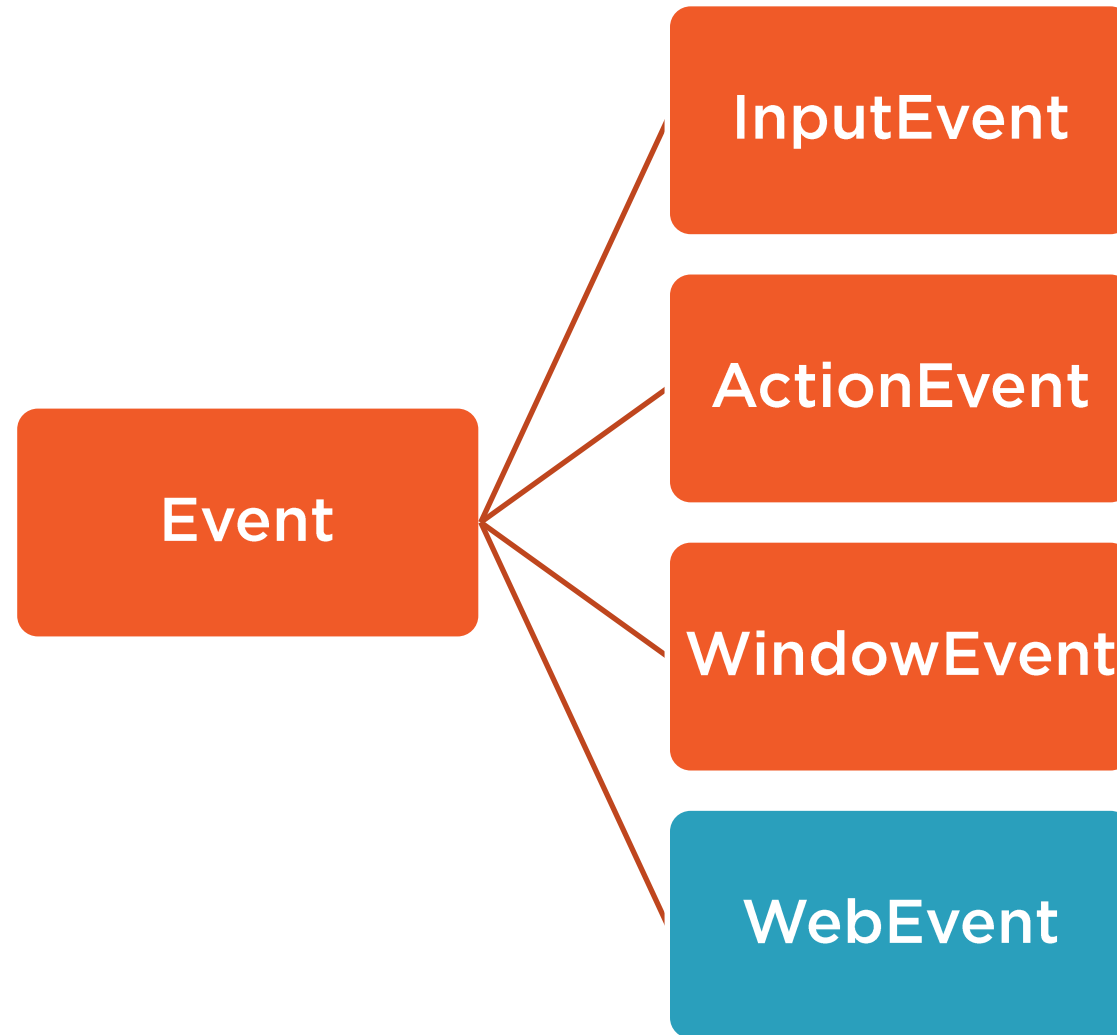
Click
KeyEvent



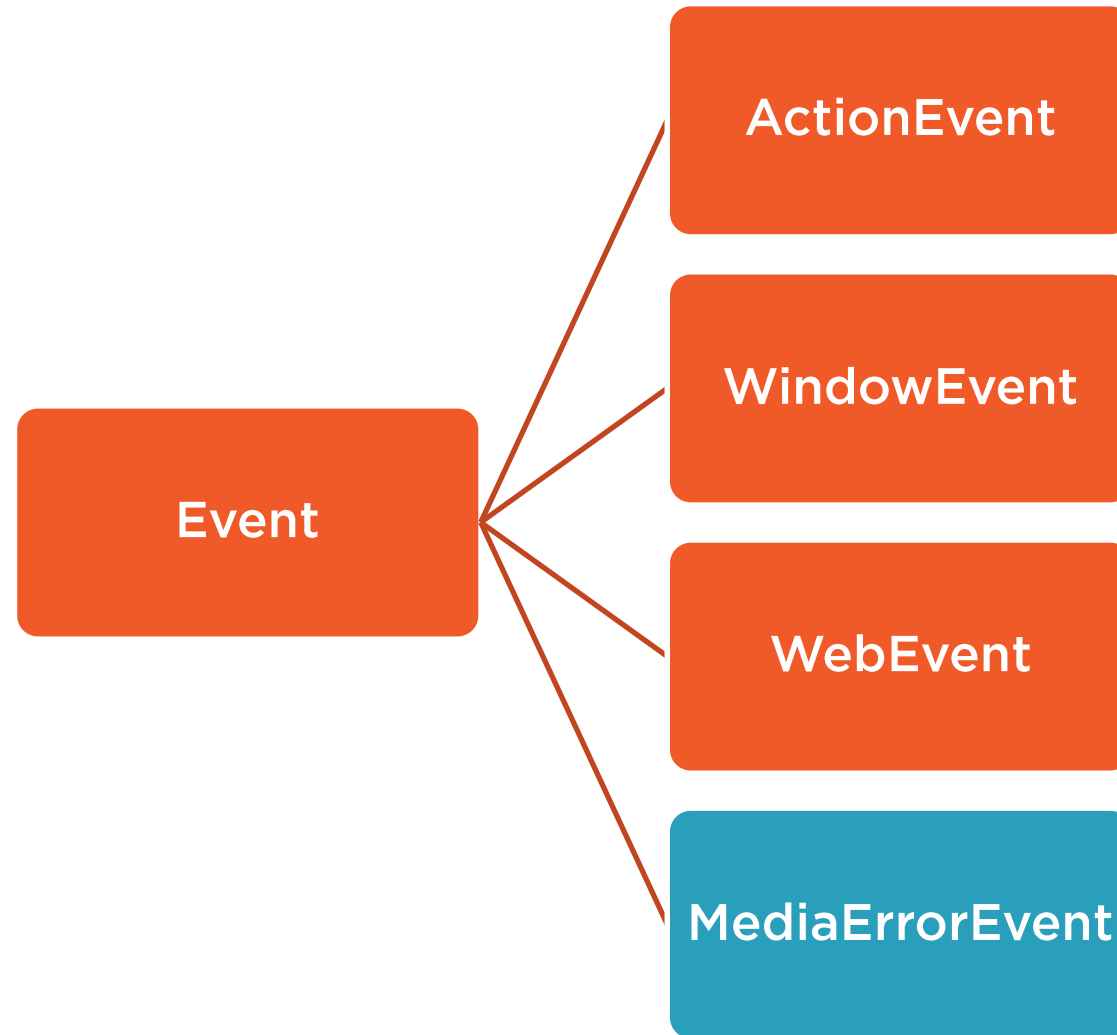
Event Type Hierarchy



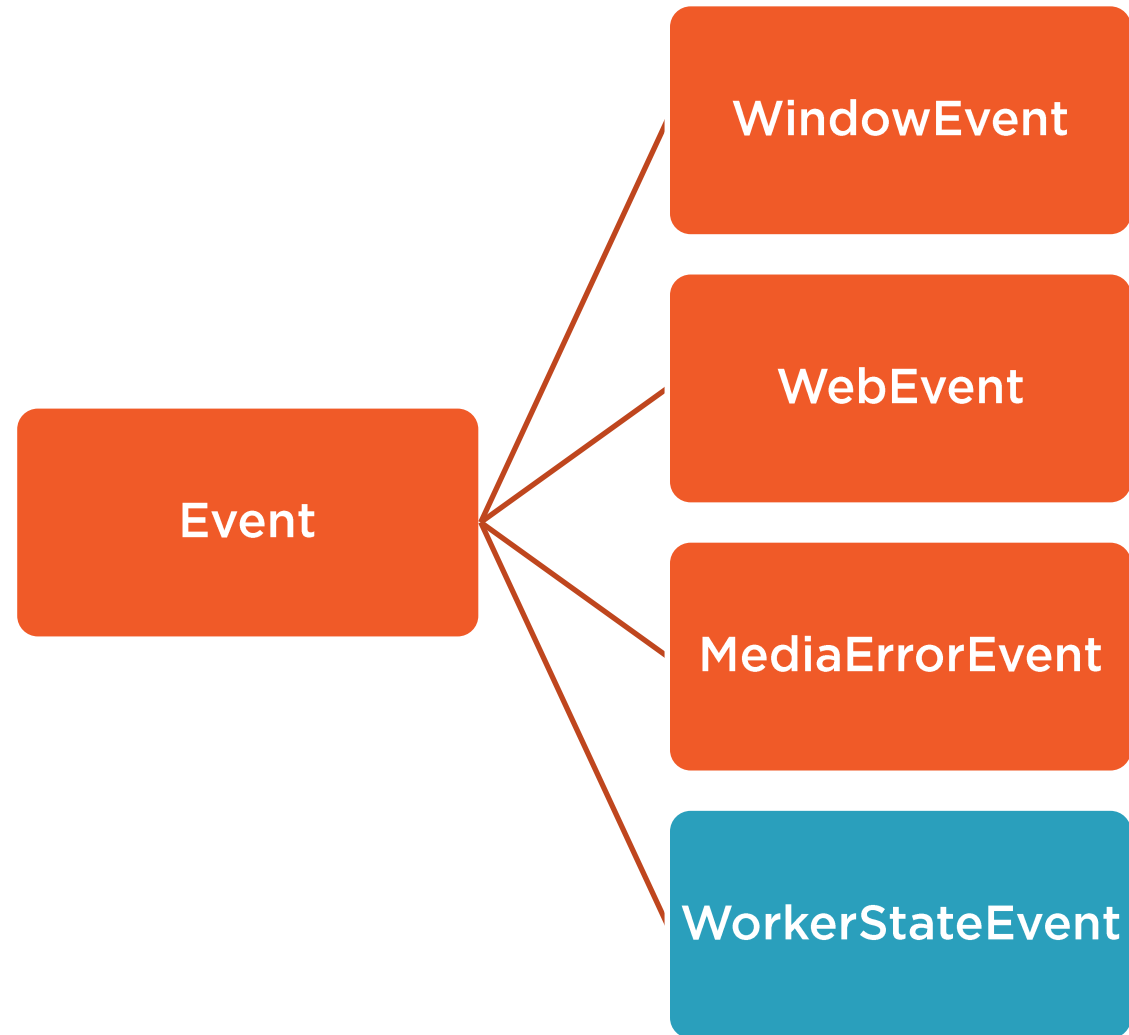
Event Type Hierarchy



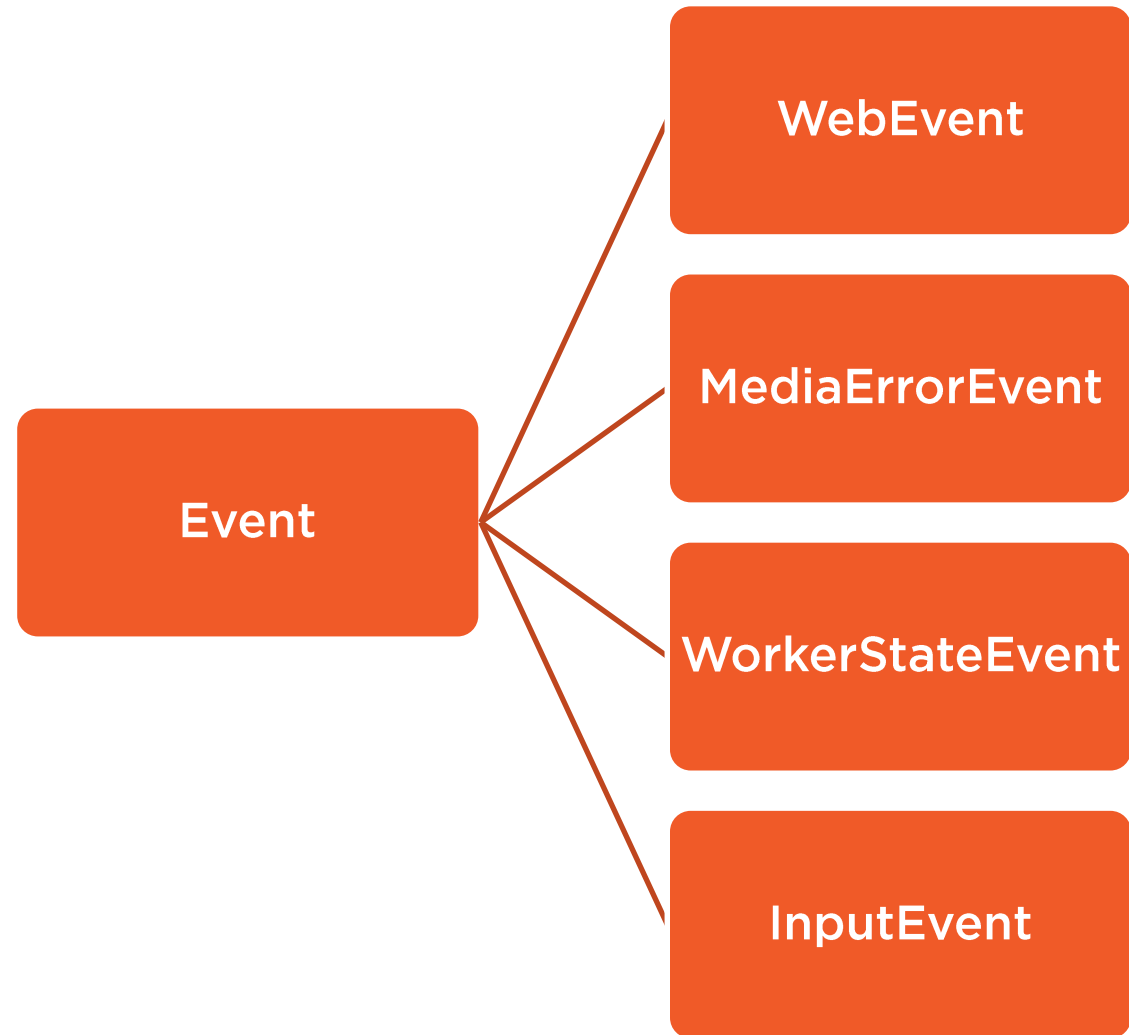
Event Type Hierarchy



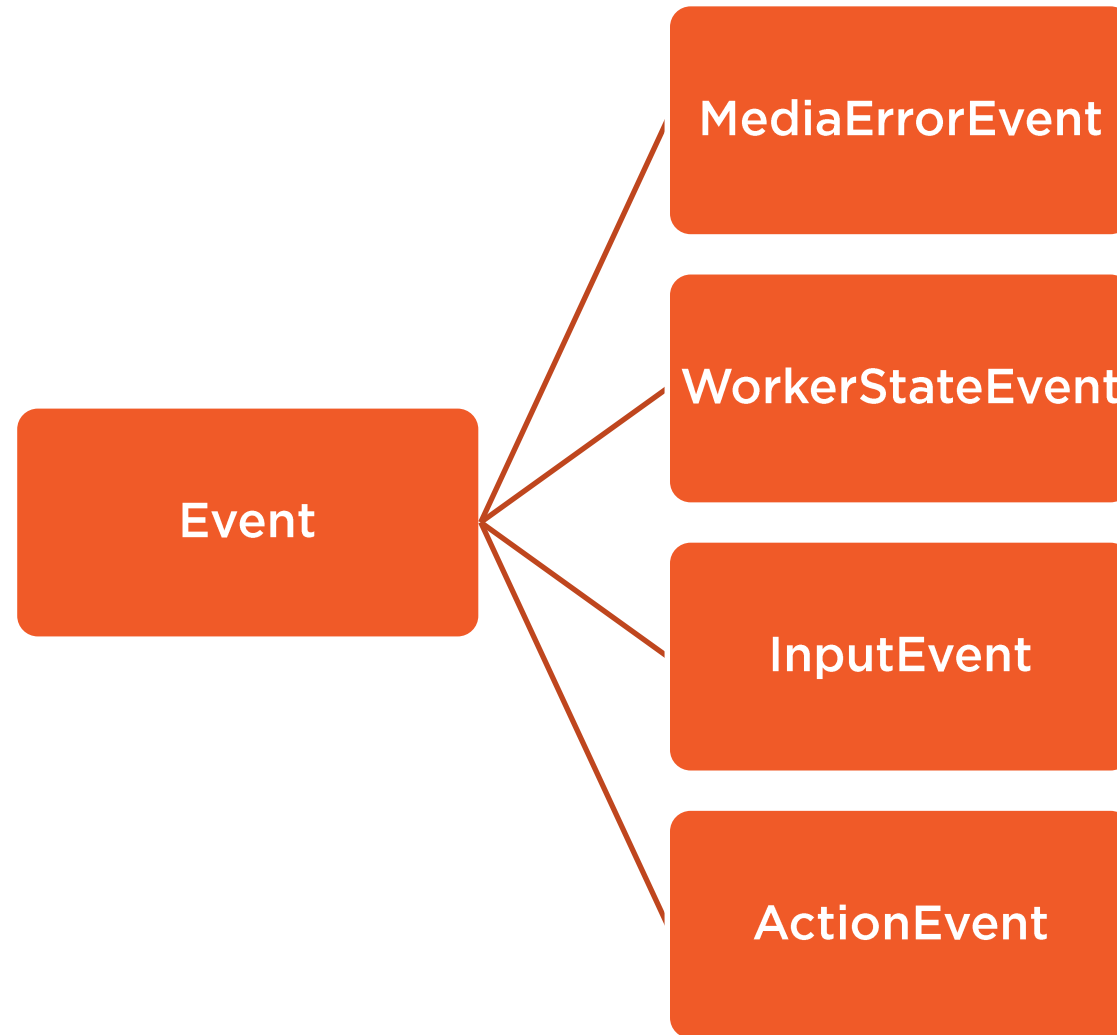
Event Type Hierarchy



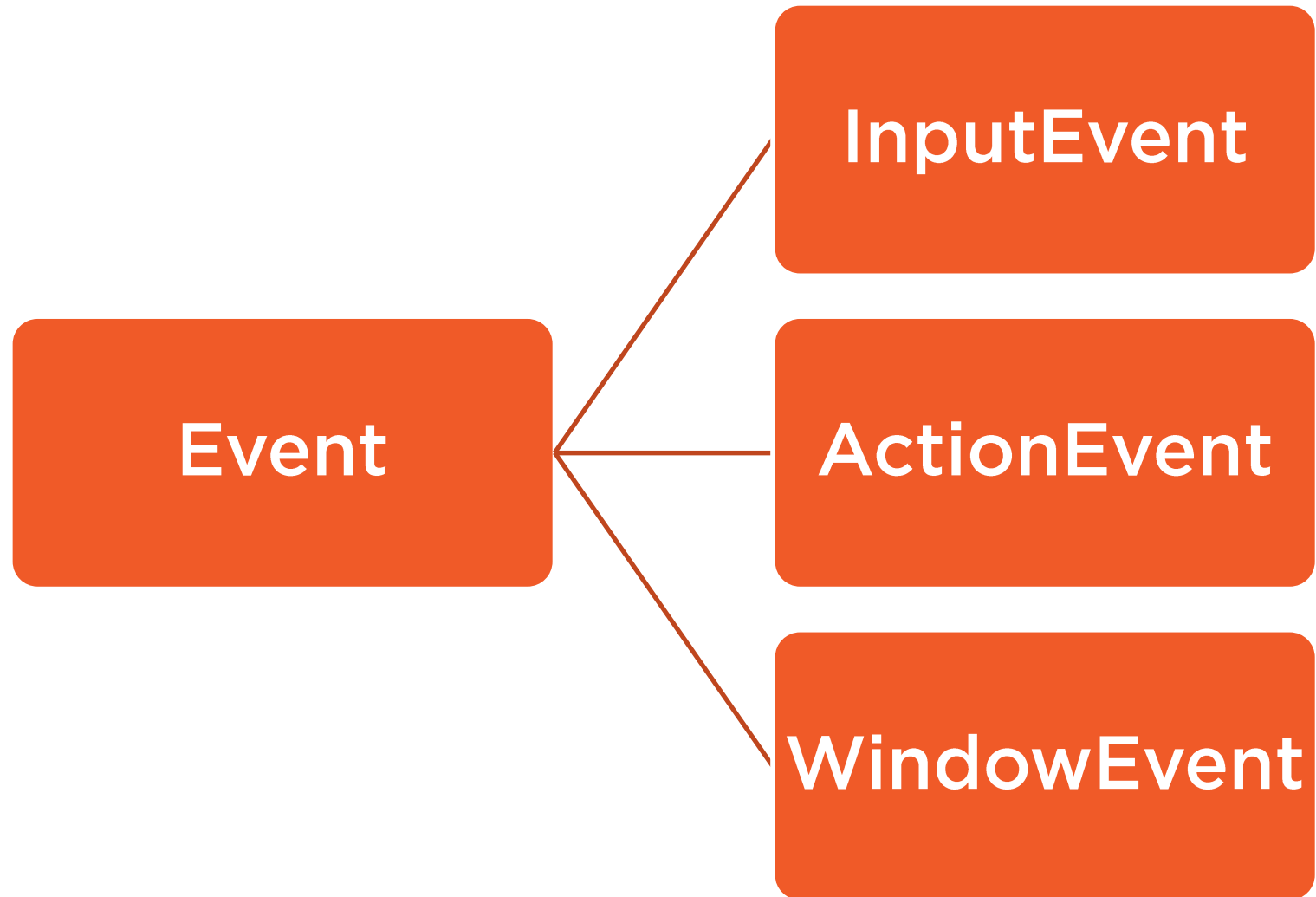
Event Type Hierarchy



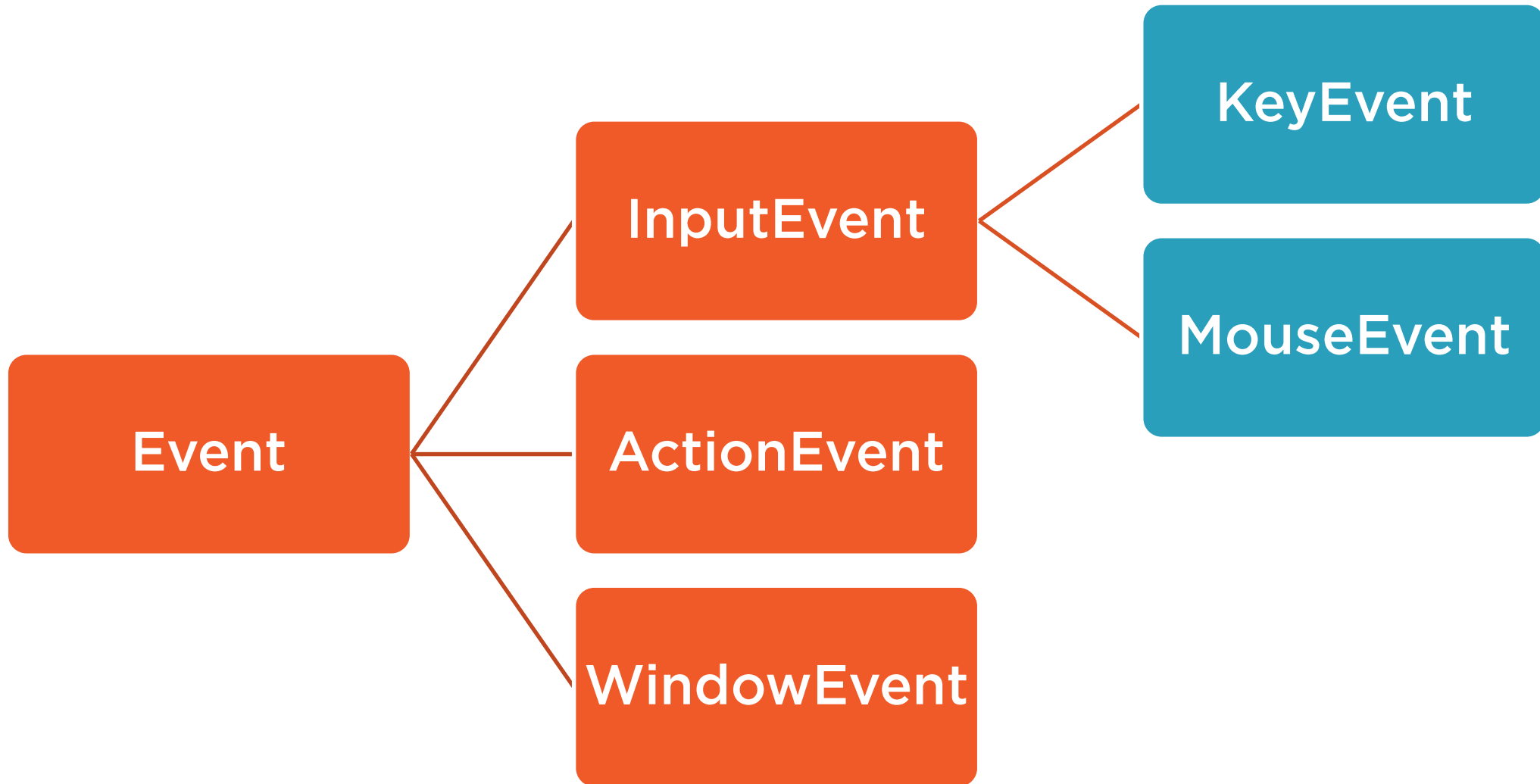
Event Type Hierarchy



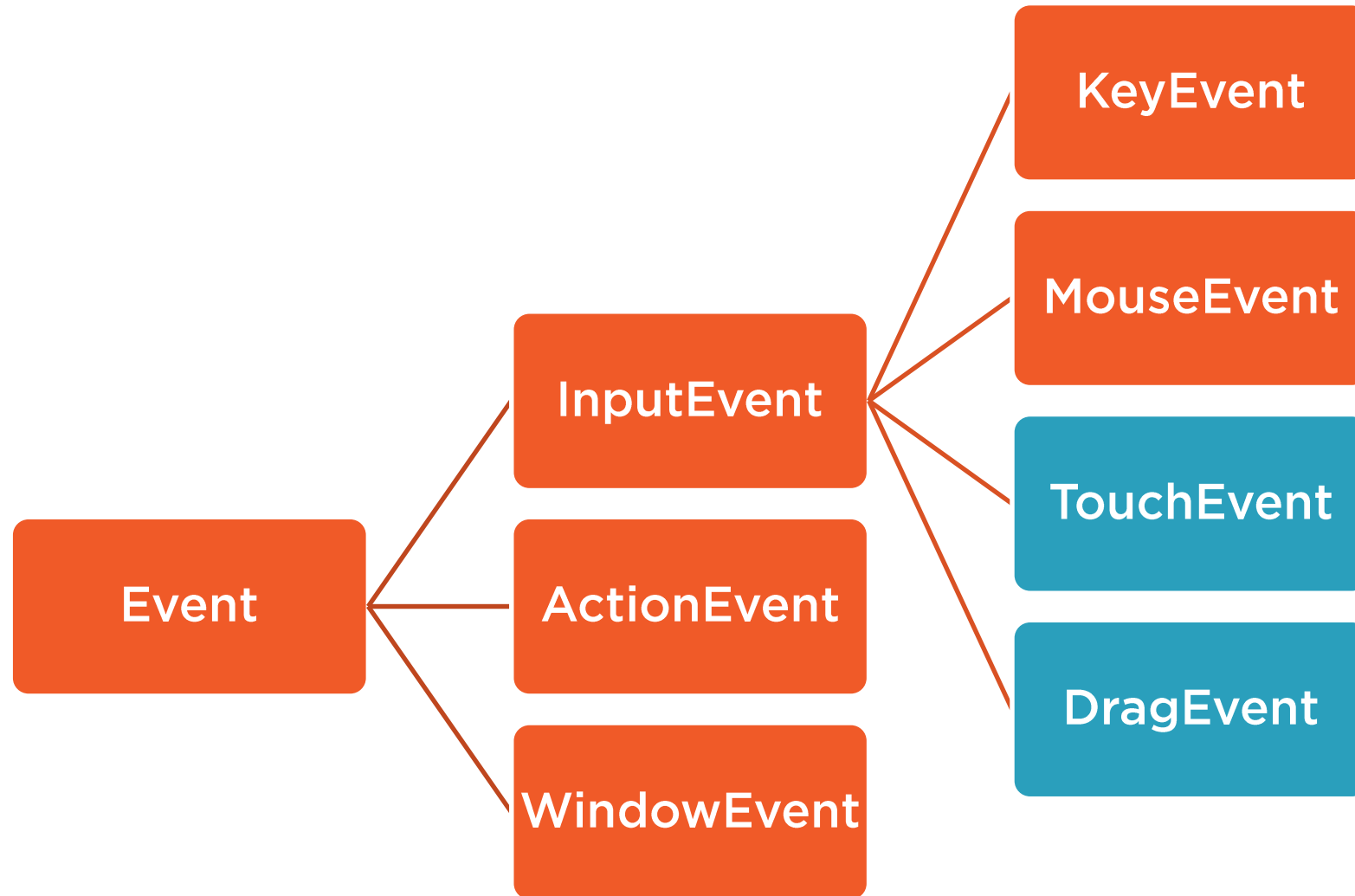
Event Type Hierarchy



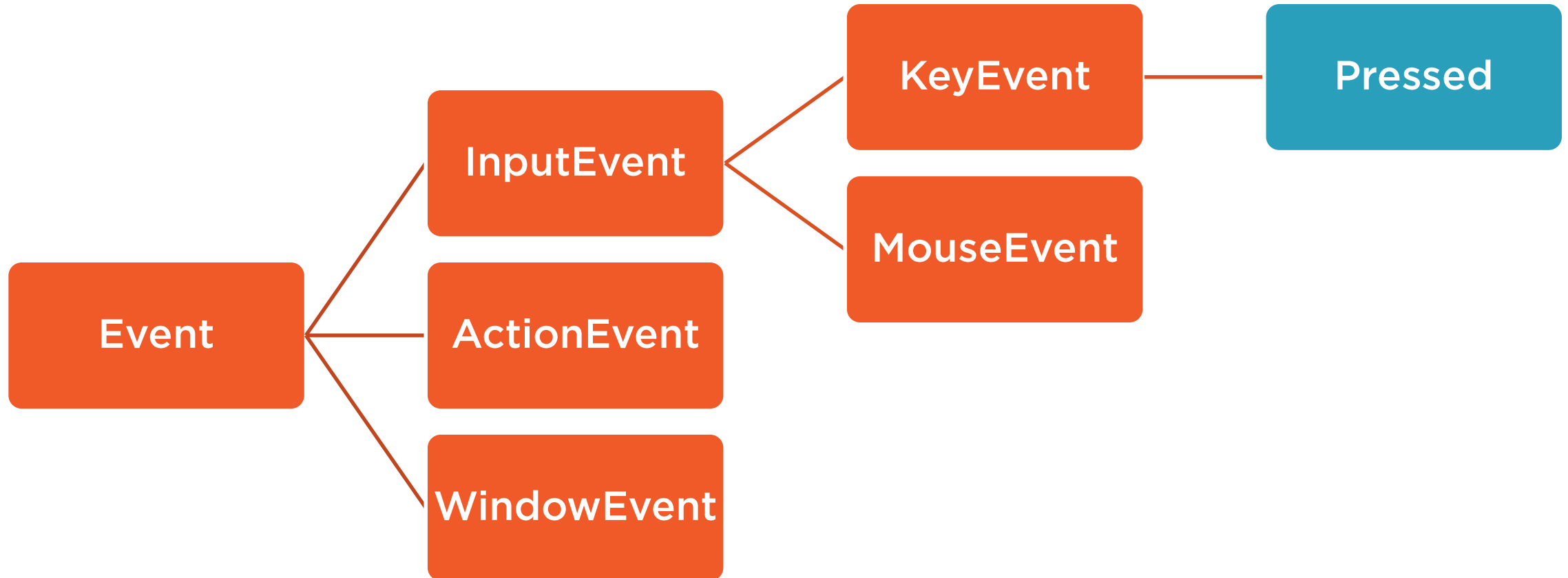
Event Type Hierarchy



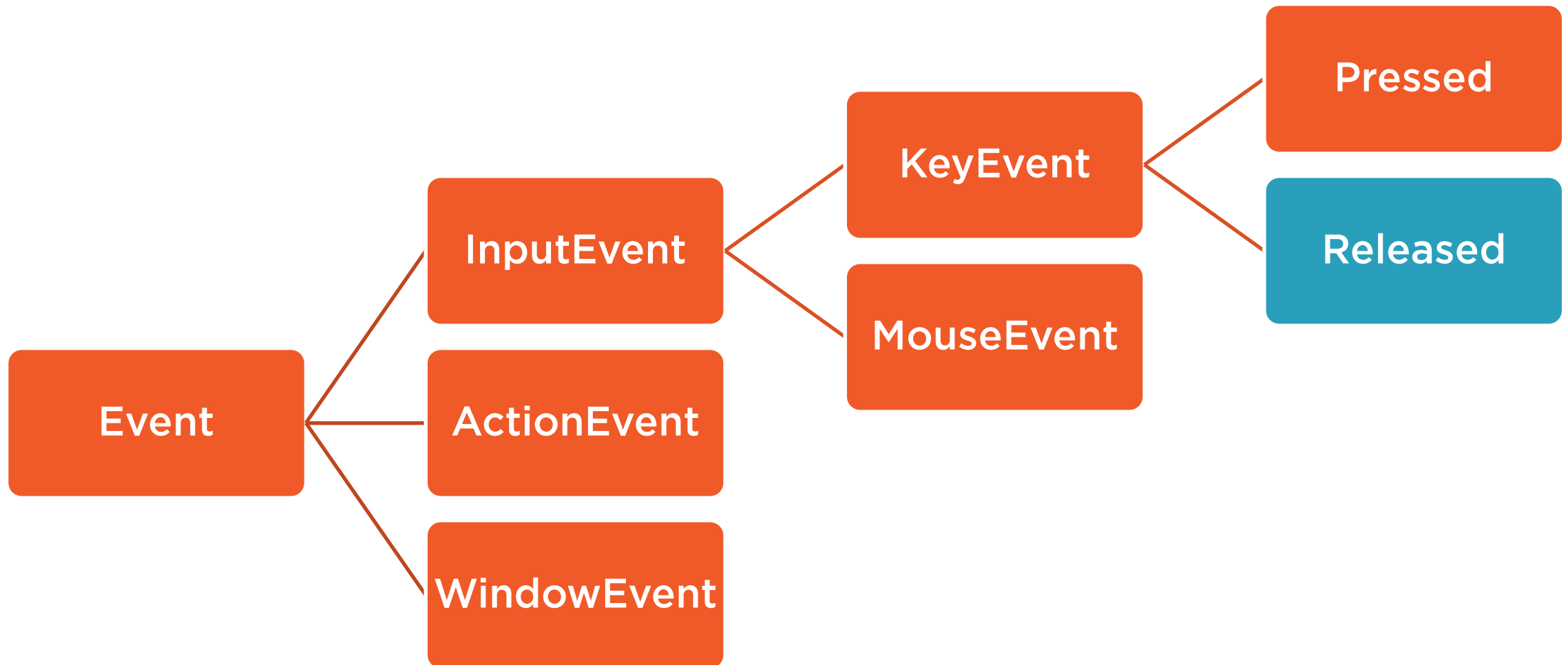
Event Type Hierarchy



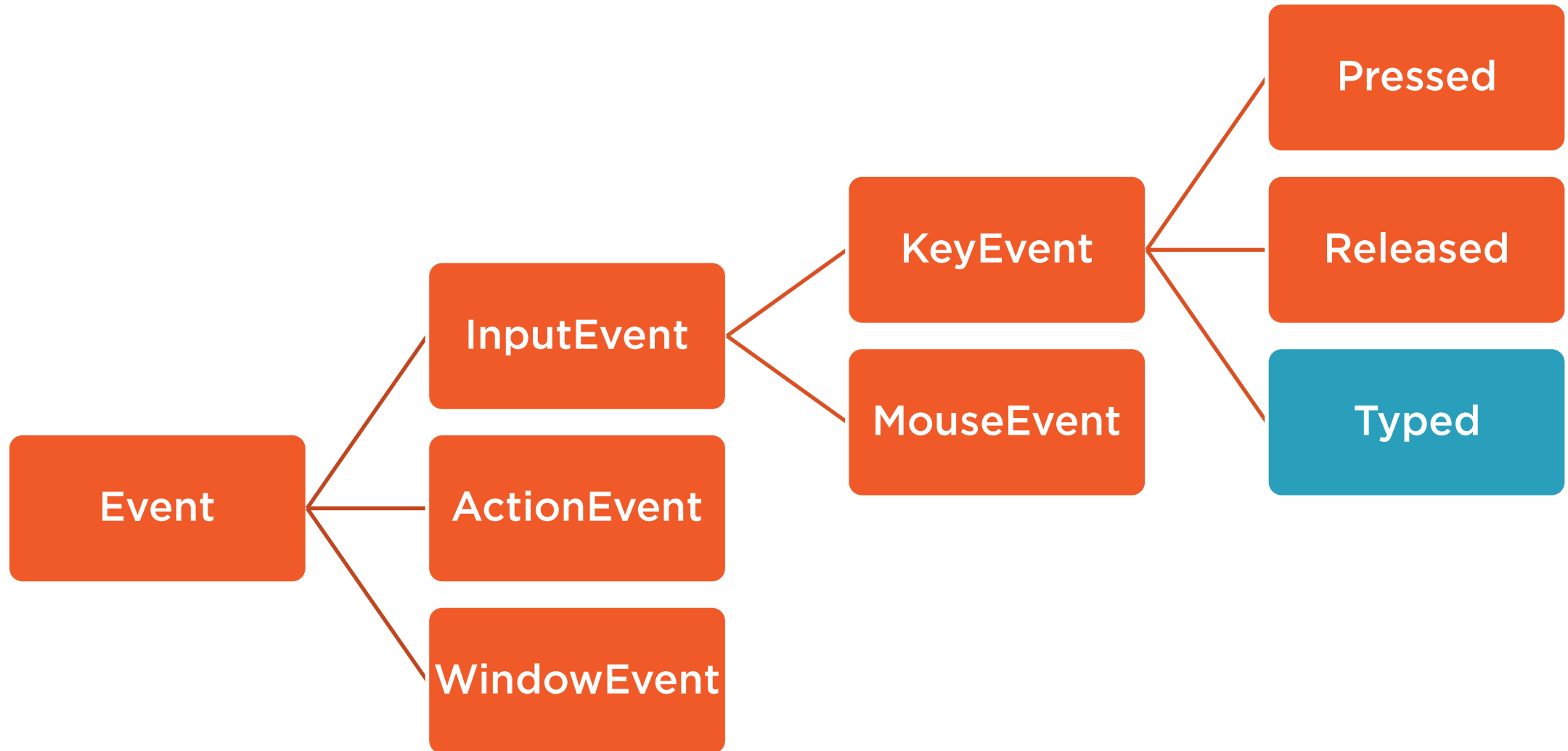
Event Type Hierarchy



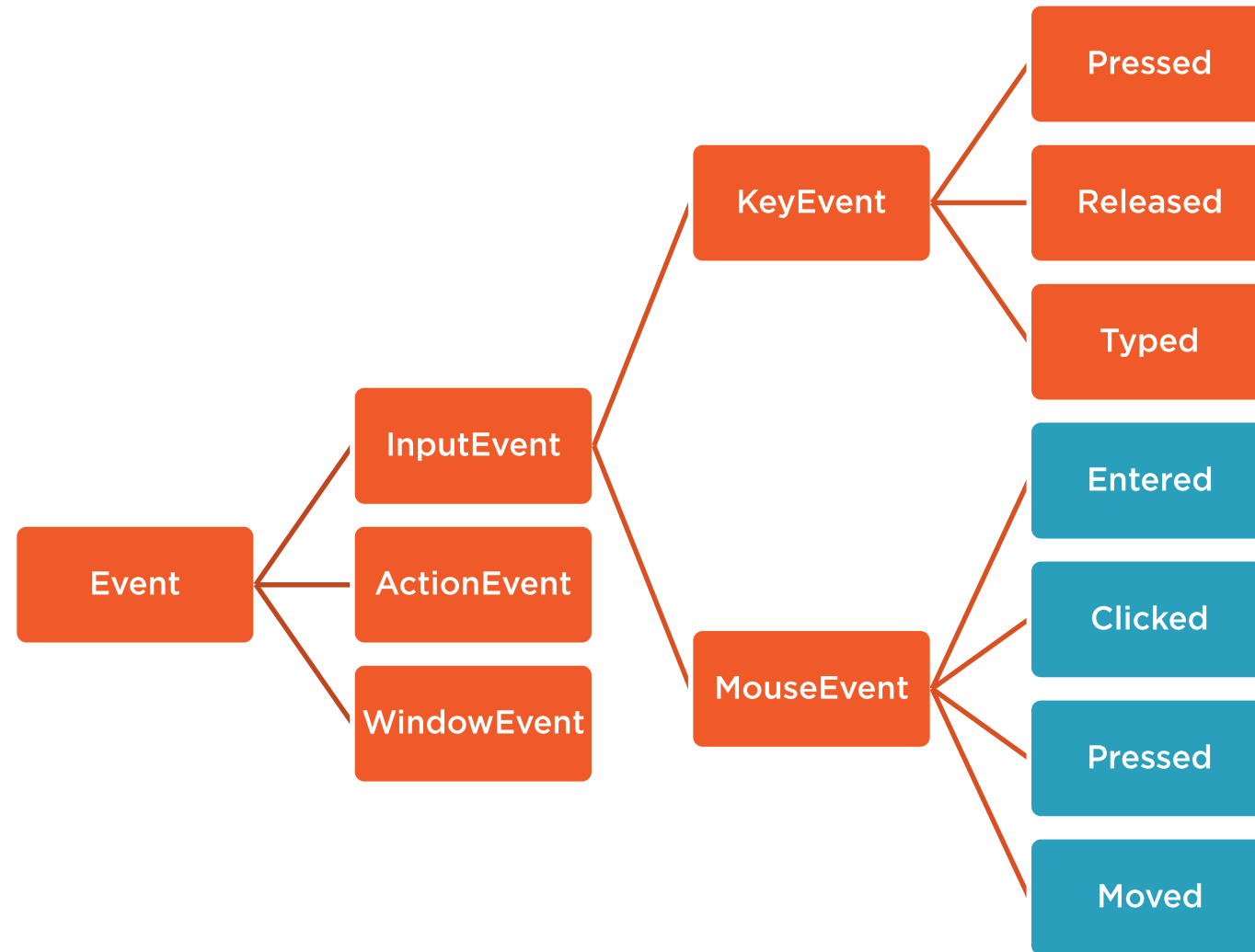
Event Type Hierarchy



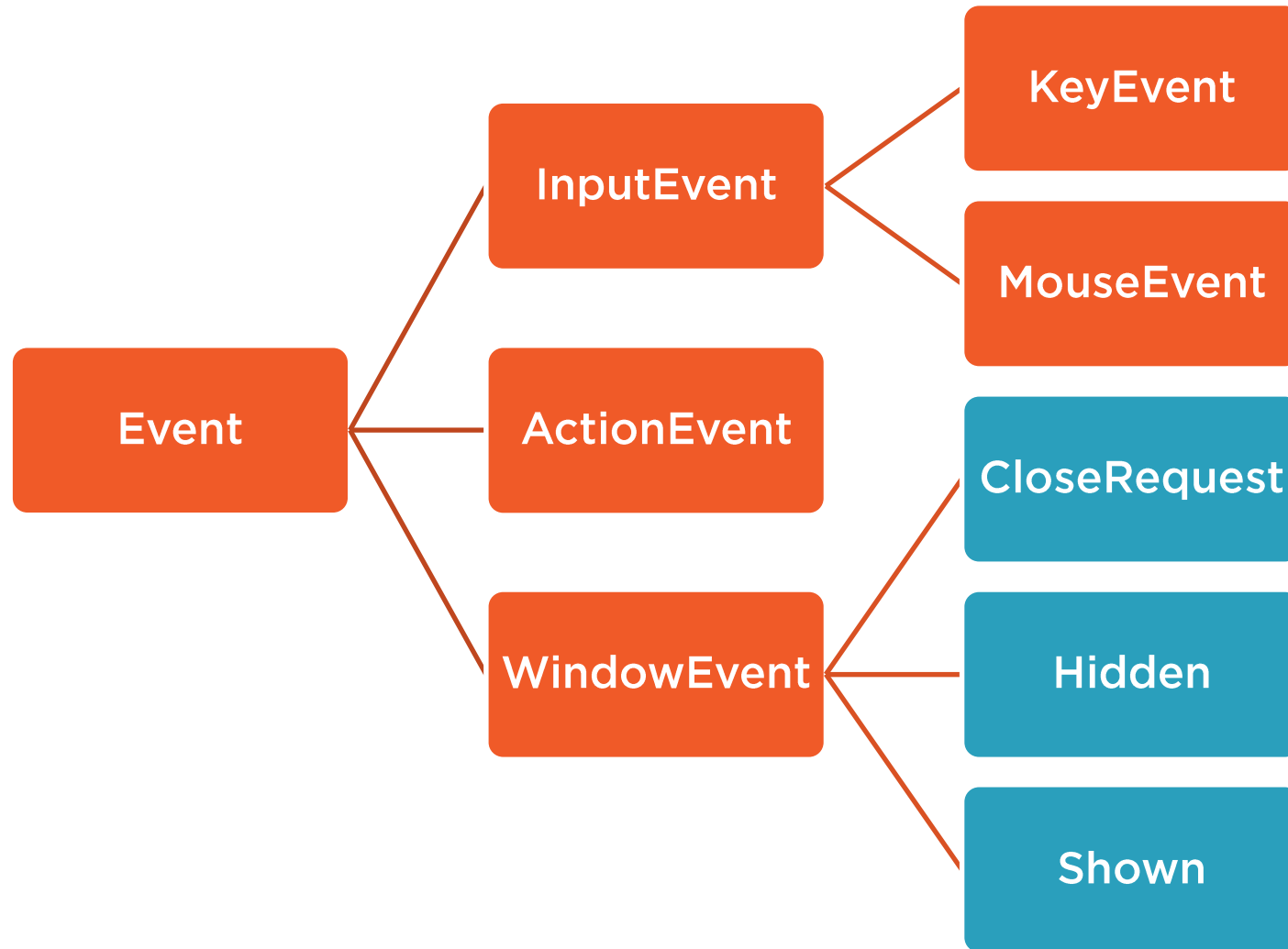
Event Type Hierarchy



Event Type Hierarchy



Event Type Hierarchy



Event Processing



Event Delivery Process

Target Selection



Route Construction



Event Capturing



Event Bubbling

The screenshot shows a window titled "Do-It!!!". It contains a table with three columns: Priority, Description, and Progress. The table has three rows of data: High priority for "Complete Design Document" (10% progress), Medium priority for "Update Class Diagram" (0% progress), and Low priority for "Fix Bug 245232" (0% progress). Below the table, there is a control area with a priority dropdown set to "High", a text input field containing "Complete Design Document", an "Update" button, a "Progress" field with the value "10", a "Completed" checkbox, and a "Cancel" button.

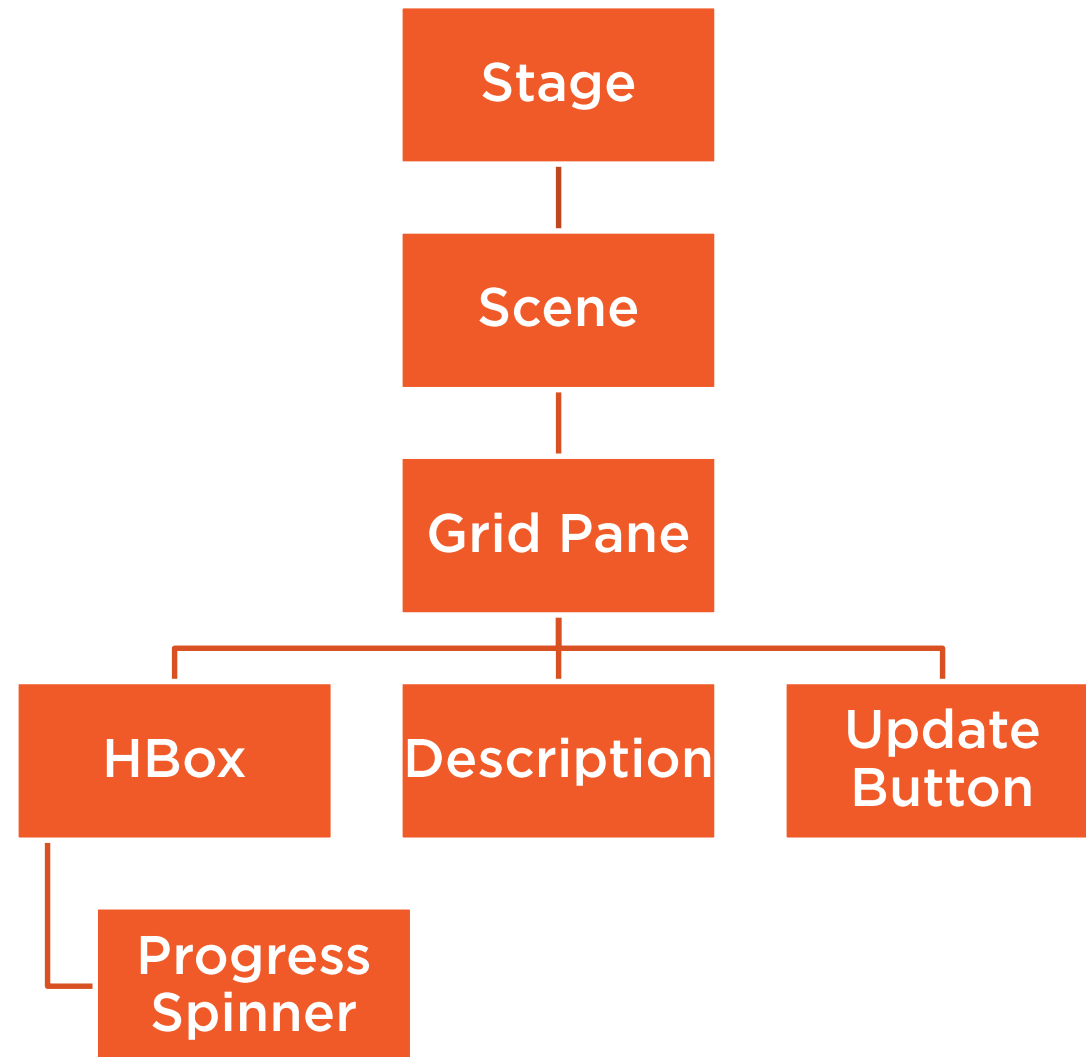
Priority	Description	Progress
High	Complete Design Document	10%
Medium	Update Class Diagram	0%
Low	Fix Bug 245232	0%

High

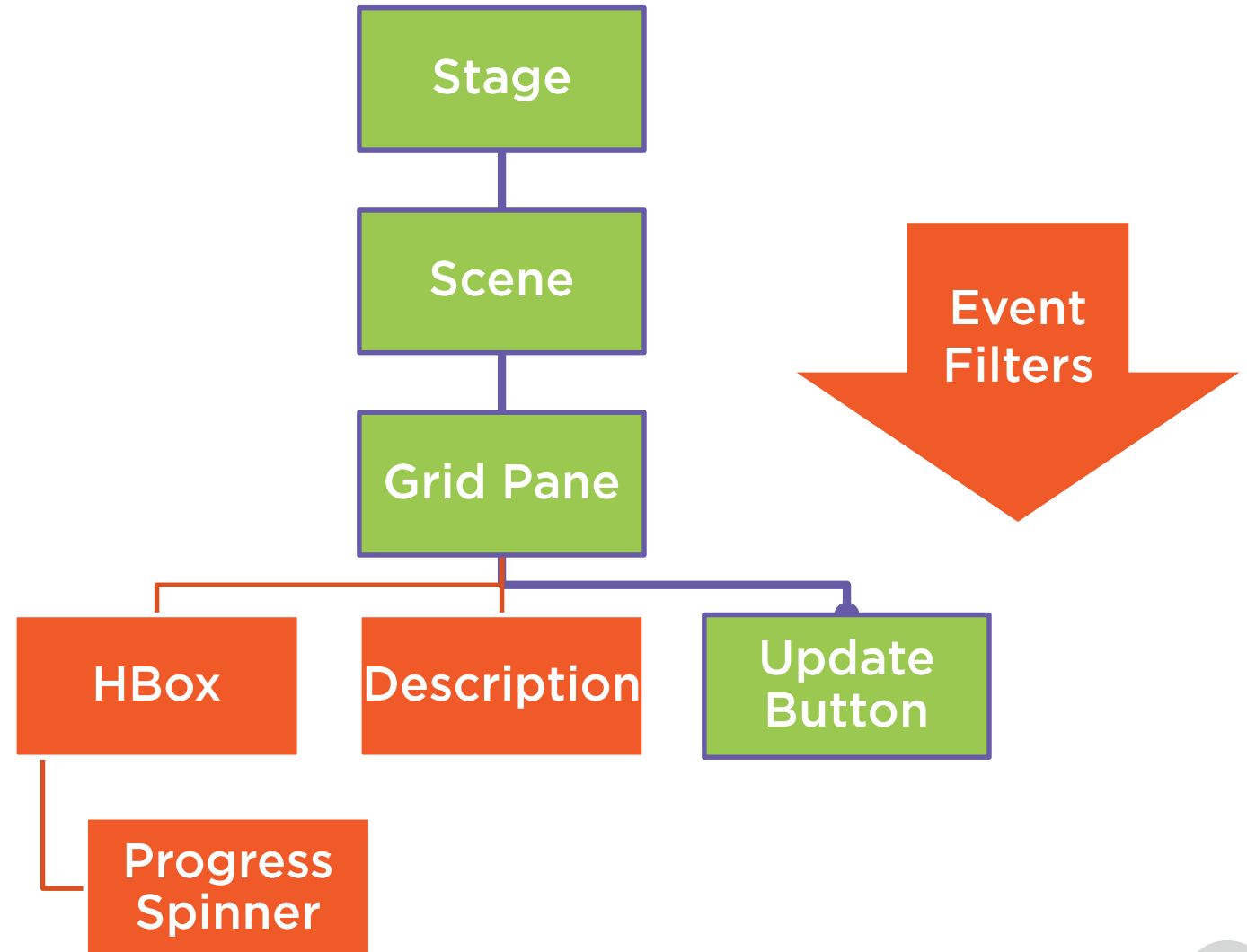
Progress ☐ Completed



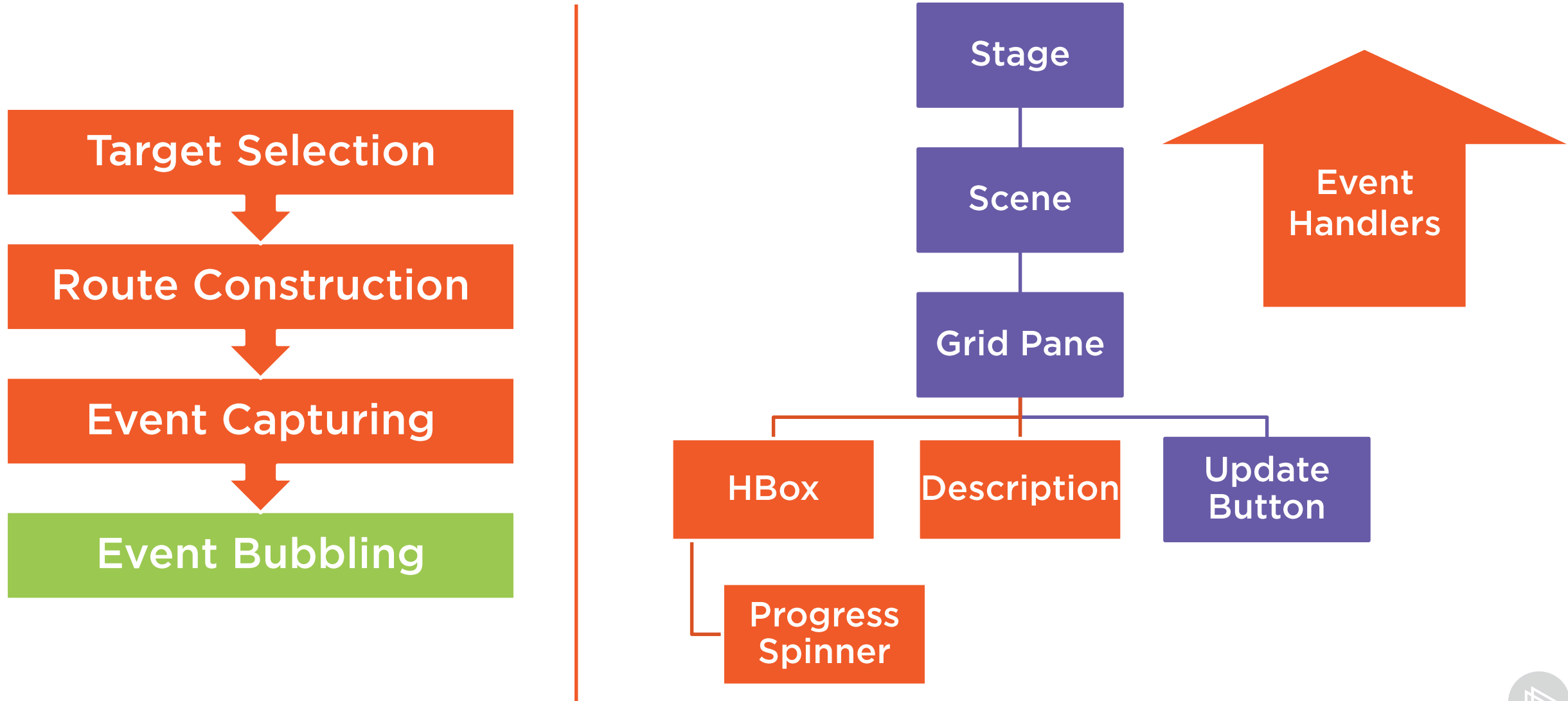
Event Delivery Process



Event Delivery Process



Event Delivery Process



Registering Handlers & Filters



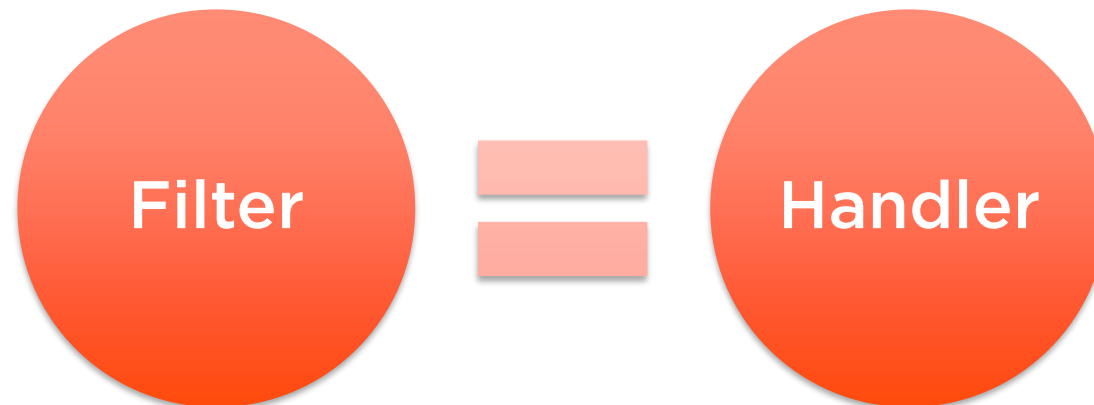
```
add.addEventHandler(MOUSE_RELEASED, new EventHandler<Event>(){  
    @Override  
    public void handle(Event event) {  
        System.out.println("Handler: Released");  
    }  
});
```

Event Handler



```
add.addEventFilter(MOUSE_RELEASED, new EventHandler<Event>(){  
    @Override  
    public void handle(Event event) {  
        System.out.println("Filter: Released");  
        event.consume();  
    }  
});
```

Event Filter



Demo



Event Handlers

Event Filters

Event Consumption



Action Event



Plain Text Button

My Text

Make a choice...

Option 1

Option 2

Option 3

Option 4

Option 5

Option 6

Longer ComboBox item

Option 7



```
completedCheckBox.setOnAction(event -> {  
    System.out.println(event.getEventType().getName());  
});  
  
completedCheckBox.addEventHandler(ActionEvent.ACTION,  
    event -> {  
        System.out.println(event.getEventType().getName());  
    });
```

Two ways to set Action Event Handlers



Demo



Demonstrate Action Event

Implement Add/Update Operation

Implement Cancel operation



Alerts & Dialogs



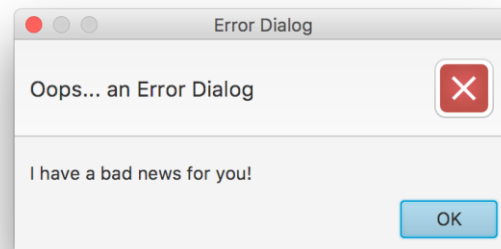
```
Alert alert = new Alert(AlertType.INFORMATION);  
alert.setTitle("Info Dialog");  
alert.setHeaderText("Some Information Dialog");  
alert.setContentText("I have some news for you!");  
alert.showAndWait();
```

Information Dialog



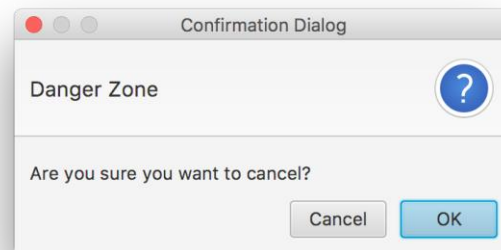
```
Alert alert = new Alert(AlertType.ERROR);  
alert.setTitle("Error Dialog");  
alert.setHeaderText("Oops... an Error Dialog");  
alert.setContentText("I have a bad news for you!");  
alert.showAndWait();
```

Error Alert



```
Alert alert = new Alert(AlertType.CONFIRMATION);  
alert.setTitle("Confirmation Dialog");  
alert.setHeaderText("Danger Zone");  
alert.setContentText("Are you sure you want to cancel?");  
Optional<ButtonType> confirmation = alert.showAndWait();  
if(confirmation.get() == ButtonType.CANCEL)  
    System.out.println("Cancelled");
```

Confirmation



Other Dialogs

TextInputDialog

For reading an text input

ChoiceDialog

For showing a list of values

Dialog

For creating custom dialogs



Demo



See some examples of Dialogs
Implement a cancel confirmation



Summary



Event Handling Mechanism

Implemented Add/Update Functionality

Implemented Cancel

Learnt about dialogs

Confirm before cancelling

