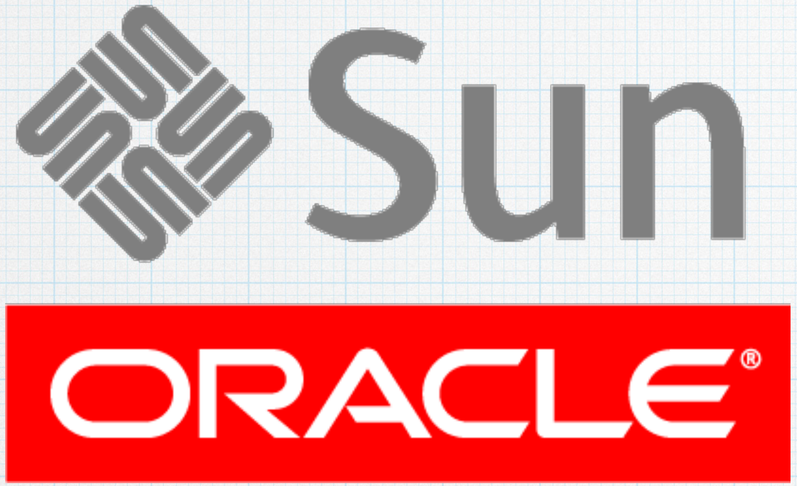
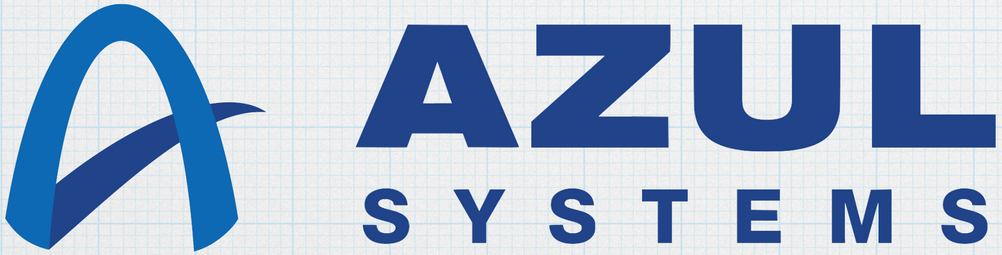
Testing JavaFX

Sergey Grinev
Azul Systems, Inc.

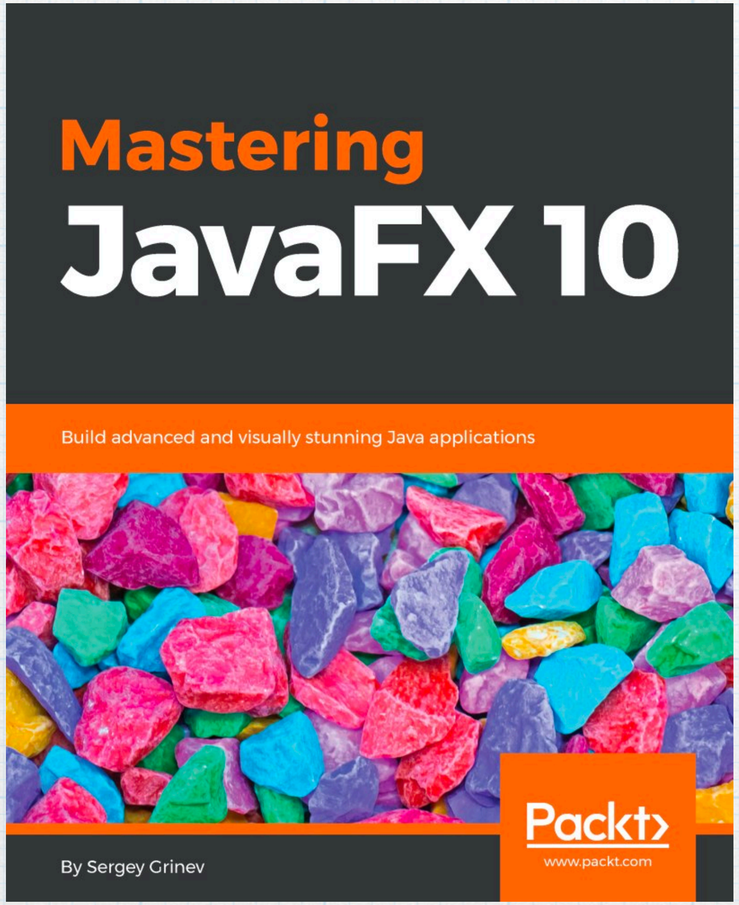
sergey@azul.com
@SergeyGrinev



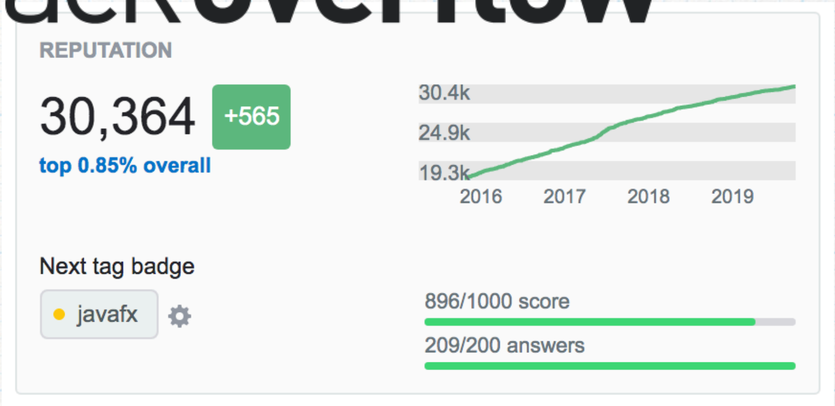


Sergey Grinev
Azul Systems, Inc.

sergey@azul.com
@SergeyGrinev



stackoverflow



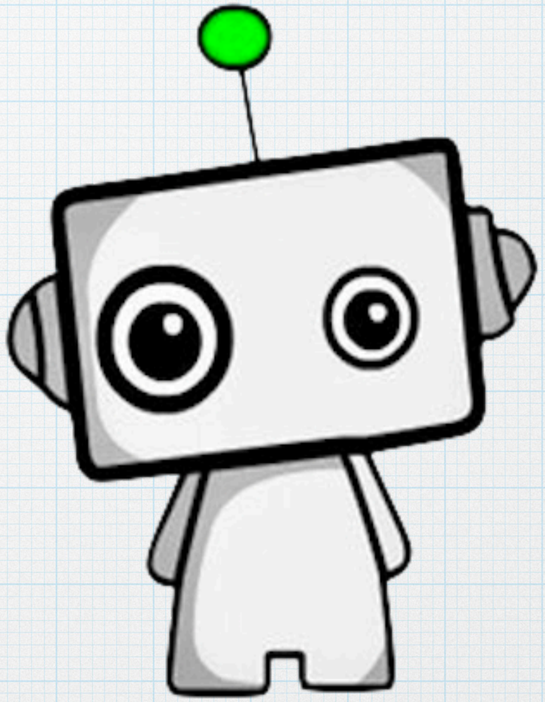
Sergey Grinev
Azul Systems, Inc.

sergey@azul.com
@SergeyGrinev

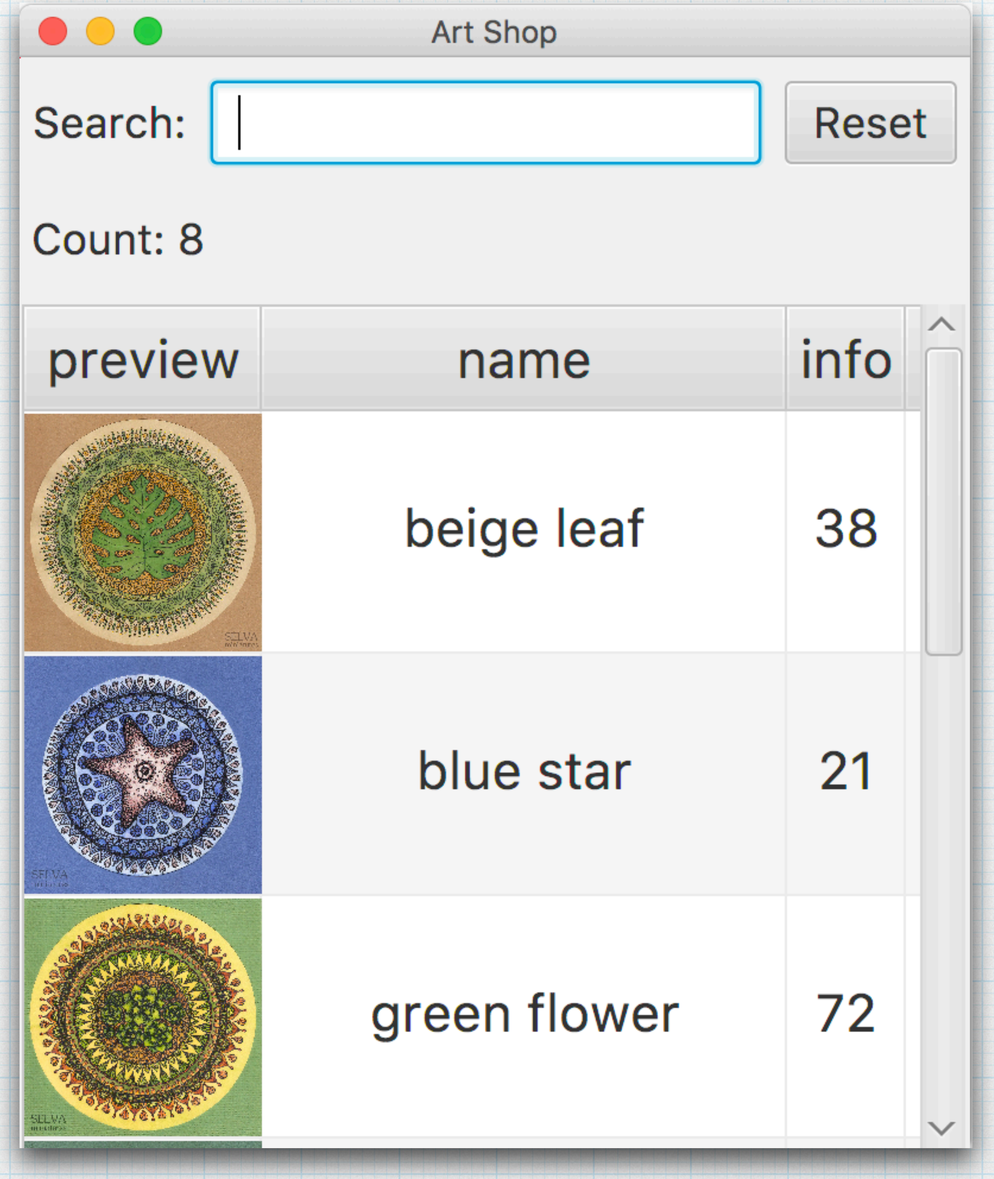
How can we test JavaFX?

javafx.scene.robot.Robot

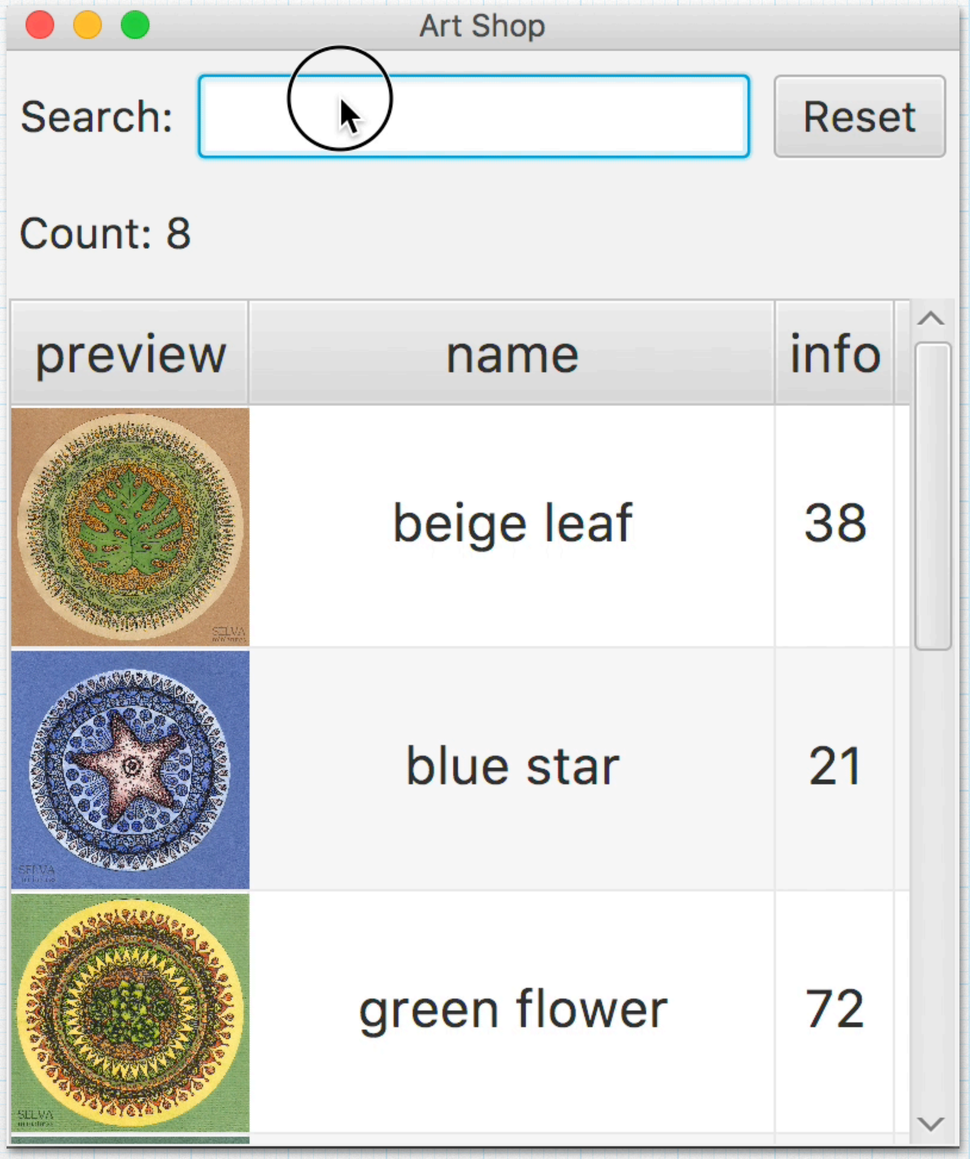
* moves mouse cursor
* makes clicks, presses keyboard buttons
* makes screenshots
* doesn't want any dependencies



Mini-robot by Krypton/Doragon - CC BY-SA 3.0

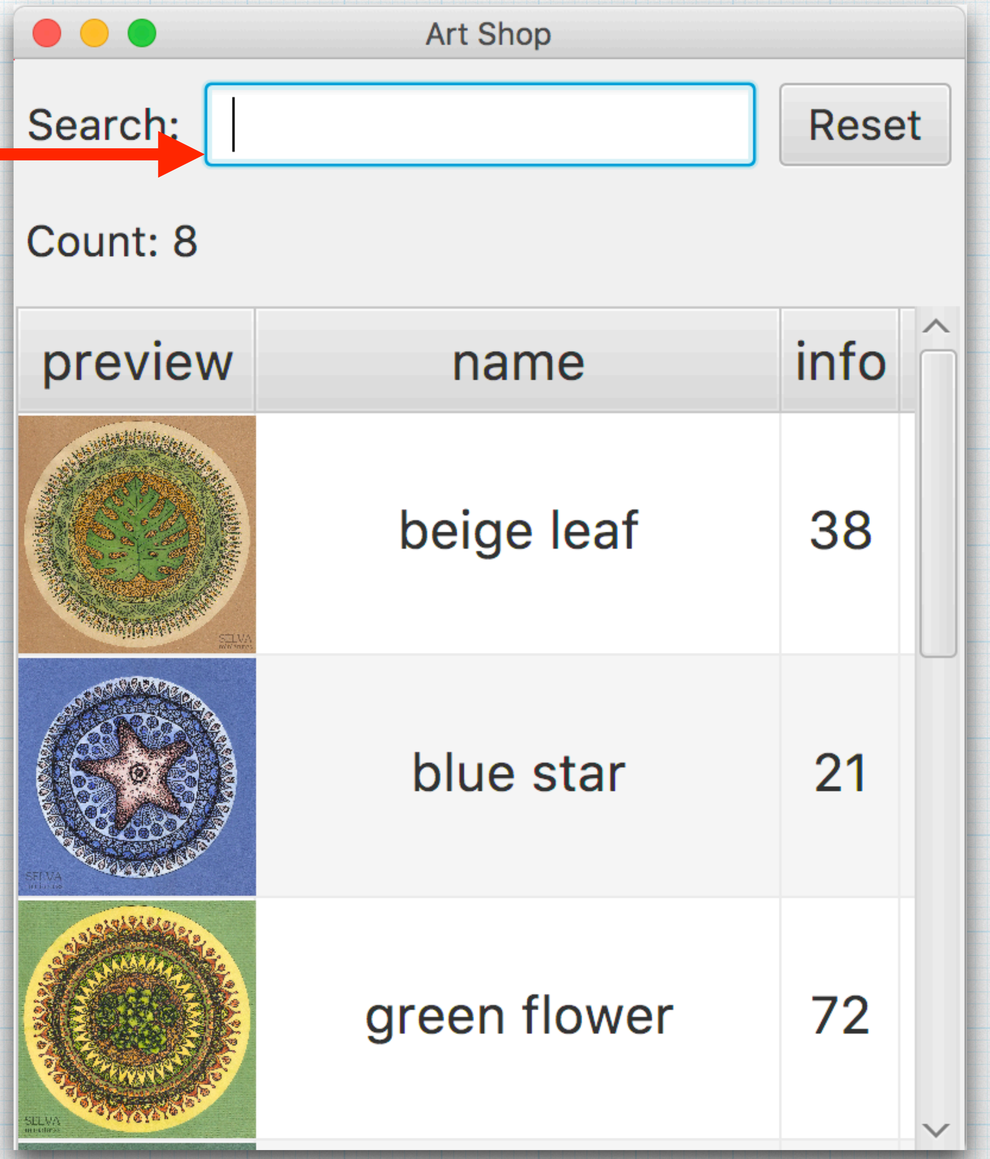


All images courtesy of @selvaminiatures


Art Shop
Search:
Reset
Count: 8
preview
name
info
beige leaf
38
blue star
21
green flower
72

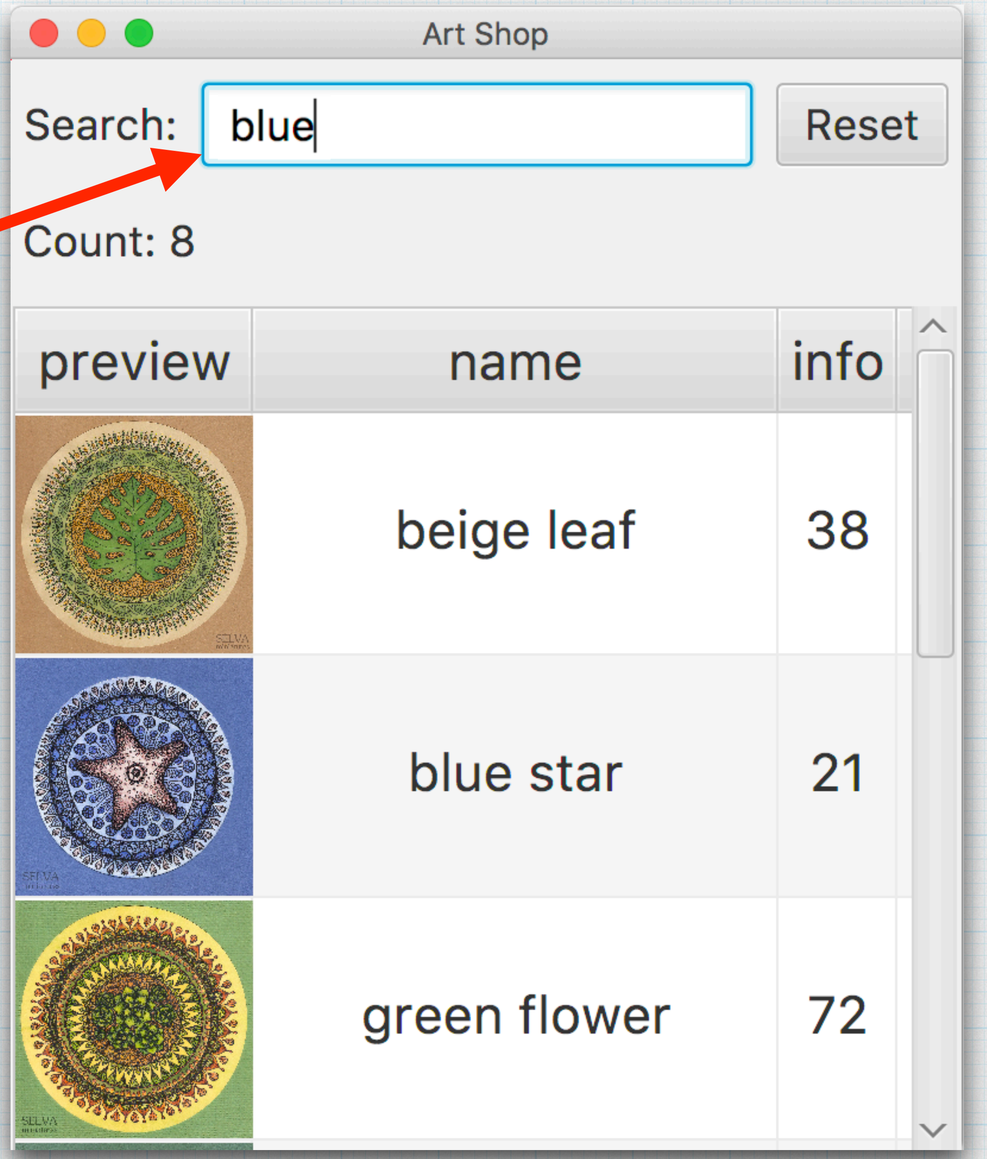
Basic test with JavaFX Robot

```
// create robot and move mouse to the text field
Platform.runLater(() -> {
    robot = new javafx.scene.robot.Robot();
    Node tfSearch = scene.lookup("#tfSearch");
    robot.mouseMove(getScreenCoordinates(tfSearch));
    robot.mouseClick(MouseButton.PRIMARY);
});
```



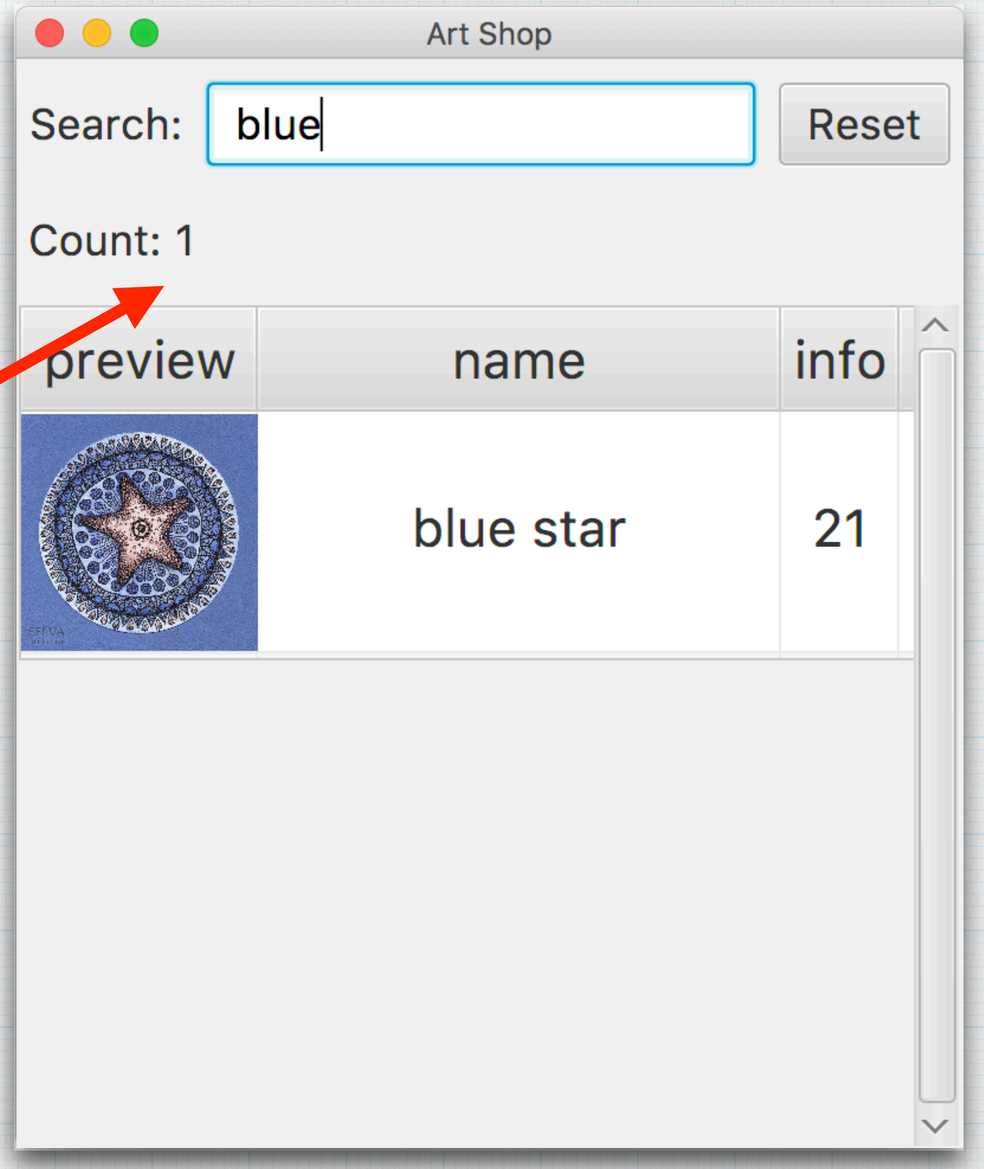
Basic test with JavaFX Robot

```
// create robot and move mouse to the text field
Platform.runLater(() -> {
    robot = new javafx.scene.robot.Robot();
    Node tfSearch = scene.lookup("#tfSearch");
    robot.mouseMove(getScreenCoordinates(tfSearch));
    robot.mouseClick(MouseButton.PRIMARY);
});

// press keyboard buttons to type "blue"
Platform.runLater(() -> {
    robot.keyType(KeyCode.B);
    robot.keyType(KeyCode.L);
    robot.keyType(KeyCode.U);
    robot.keyType(KeyCode.E);
});
```



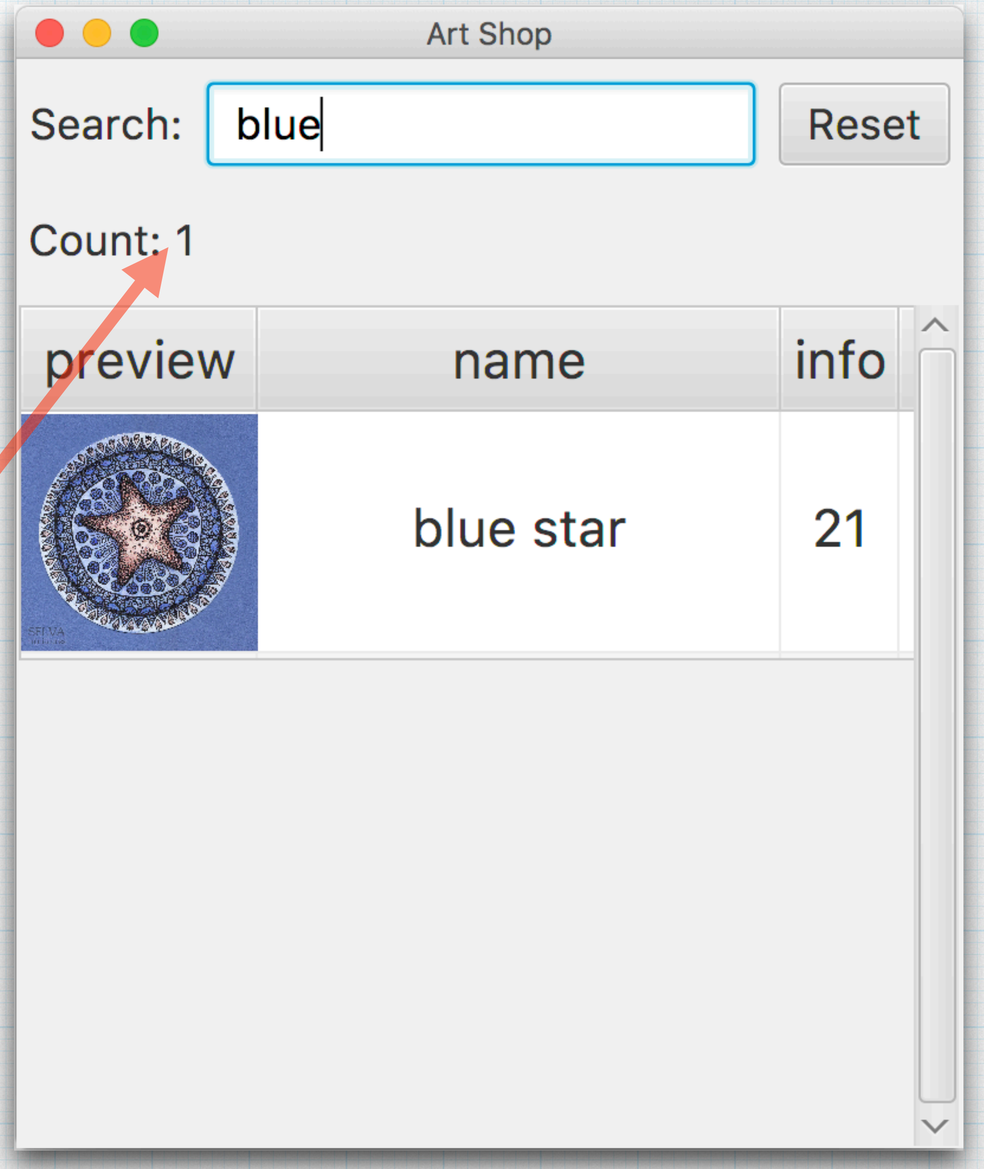
Basic test with JavaFX Robot

```java
// create robot and move mouse to the text field
Platform.runLater(() -> {
    robot = new javafx.scene.robot.Robot();
    Node tfSearch = scene.lookup("#tfSearch");
    robot.mouseMove(getScreenCoordinates(tfSearch));
    robot.mouseClick(MouseButton.PRIMARY);
});

// press keyboard buttons to type "blue"
Platform.runLater(() -> {
    robot.keyType(KeyCode.B);
    robot.keyType(KeyCode.L);
    robot.keyType(KeyCode.U);
    robot.keyType(KeyCode.E);
});

// allocate time for UI to response
try {
    Thread.sleep(500);
} catch (InterruptedException e) {}
```



Basic test with JavaFX Robot

```java
// create robot and move mouse to the text field
Platform.runLater(() -> {
    robot = new javafx.scene.robot.Robot();
    Node tfSearch = scene.lookup("#tfSearch");
    robot.mouseMove(getScreenCoordinates(tfSearch));
    robot.mouseClick(MouseButton.PRIMARY);
});

// press keyboard buttons to type "blue"
Platform.runLater(() -> {
    robot.keyType(KeyCode.B);
    robot.keyType(KeyCode.L);
    robot.keyType(KeyCode.U);
    robot.keyType(KeyCode.E);
});

// allocate time for UI to response
try {
    Thread.sleep(500);
} catch (InterruptedException e) {}

// verify the count value
Platform.runLater(() -> {
    Label labelCount = (Label) scene.lookup("#labelCount");
    // pseudo-assert
    System.out.println(
          labelCount.getText().equals("1") ? "PASS" : "FAIL");
});
```



JavaFX Robot

```java
// create robot and move mouse to the text field
Platform.runLater(() -> {
    robot = new javafx.scene.robot.Robot();
    Node tfSearch = scene.lookup("#tfSearch");
    robot.mouseMove(getScreenCoordinates(tfSearch));
    robot.mouseClick(MouseButton.PRIMARY);
});

// press keyboard buttons to type "blue"
Platform.runLater(() -> {
    robot.keyType(KeyCode.B);
    robot.keyType(KeyCode.L);
    robot.keyType(KeyCode.U);
    robot.keyType(KeyCode.E);
});

// allocate time for UI to response
try {
    Thread.sleep(500);
} catch (InterruptedException e) {}

// verify the uppercased value in the left text field
Platform.runLater(() -> {
    Label labelCount = (Label) scene.lookup("#labelCount");
    Assert.assertEquals(labelCount.getText(), "1");
});
```

TestFX

```java
clickOn("#tfSearch");

write("blue");

// WaitForAsyncUtils.waitForFxEvents()

Label label = (Label) lookup("#labelCount").query();
assertEquals(label.getText(), "1");
```

JemmyFX

```java
SceneDock scene = new SceneDock();

TextInputControlDock textFieldDock =
        new TextInputControlDock(scene.asParent(), "tfSearch");

textFieldDock.type("blue");

LabeledDock label =
        new LabeledDock(scene.asParent(), "labelCount");

assertEquals(label.getText(), "1");
```

TestFX

```java
clickOn("#tfSearch");

write("blue");

Label label = (Label) lookup("#labelCount").query();
assertEquals(label.getText(), "1");
```

JemmyFX

* used by for Swing, AWT, and JavaFX testing
* large test base at jfx-dev/tests repo
* great utility working with screenshots
* But:
  * intricate old-school API
  * still in Mercurial
  * being updated slowly and rarely

* http://hg.openjdk.java.net/openjfx/jfx-dev/tests/

TestFX

* being actively developed
* used by several opensource projects and libraries
* modern fluent API
* support many modern technologies
* headless execution
* But:
    * still in alpha status
    * no backward compatibility between major releases
    * limited screenshot capabilities

Commercial Frameworks

* Allows recording steps
* Provides IDE to work with them
* Can play those steps on multiple platforms
* Coding is minimal or not required
* Costs a lot of money

Test Phases

TestFX

```
public class AppTest extends ApplicationTest {

    public void start (Stage stage) throws Exception {
        Parent root = FXMLLoader.load(
                getClass().getResource("landing.fxml"));
        stage.setScene(new Scene(root, 400, 500));
        stage.show();
    }

    @Test
    public void testBlue() {
        clickOn("#tfSearch");
        write("blue");

        Label label = (Label) lookup("#labelCount").query();
        assertEquals(label.getText(), "1");
    }

    @After
    public void tearDown () throws Exception {
        FxToolkit.cleanupStages();
        release(new KeyCode[]{});
        release(new MouseButton[]{});
    }

}
```

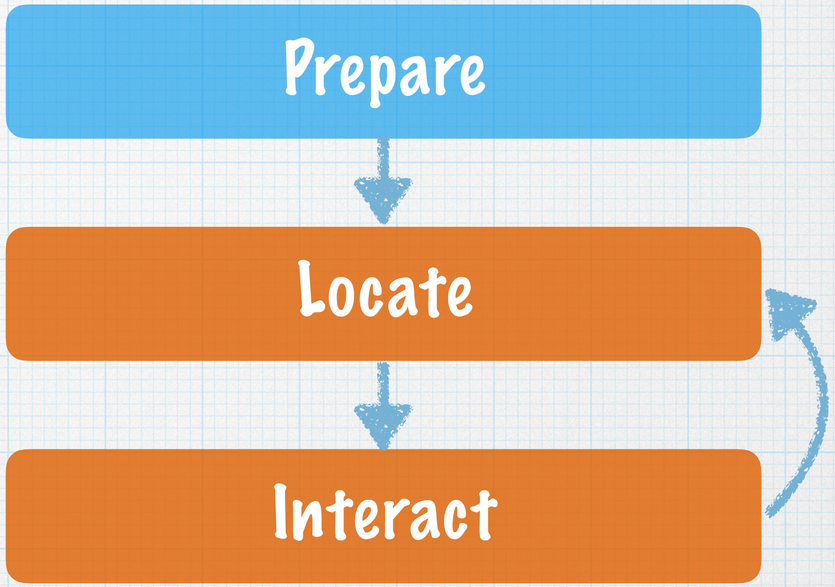
Test Phases: Prepare

```
public class AppTest extends ApplicationTest {

    public void start (Stage stage) throws Exception {
        Parent root = FXMLLoader.load(
                getClass().getResource("landing.fxml"));
        stage.setScene(new Scene(root, 400, 500));
        stage.show();
    }

    @Test
    public void testBlue() {
        clickOn("#tfSearch");
        write("blue");

        Label label = (Label) lookup("#labelCount").query();
        assertEquals(label.getText(), "1");
    }

    @After
    public void tearDown () throws Exception {
        FxToolkit.cleanupStages();
        release(new KeyCode[]{});
        release(new MouseButton[]{});
    }
}
```

Prepare

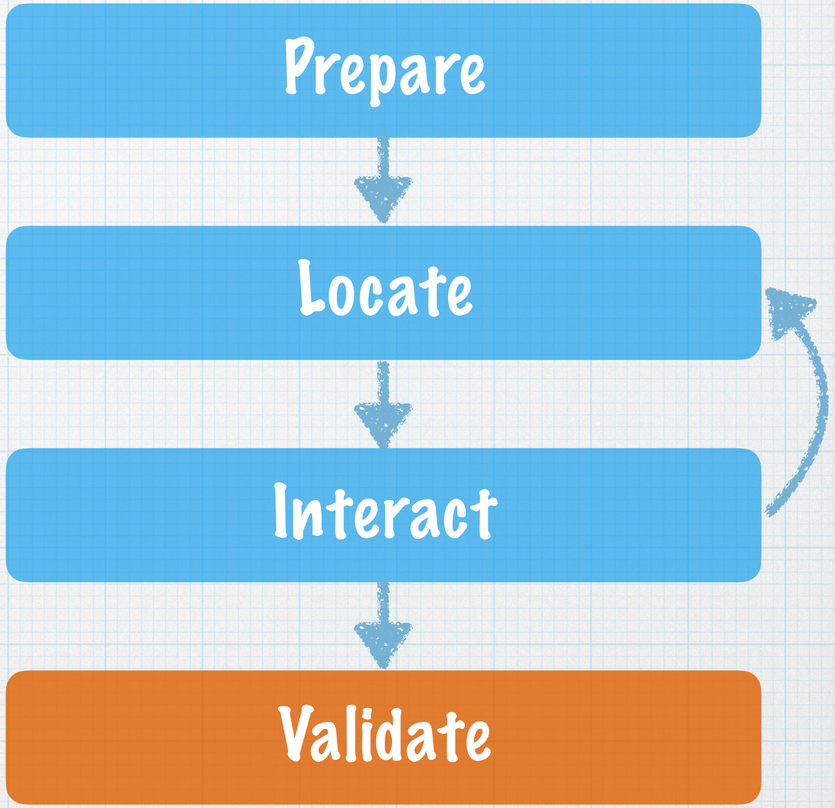
Test Phases: Locate and Interact

```
public class AppTest extends ApplicationTest {

    public void start (Stage stage) throws Exception {
        Parent root = FXMLLoader.load(
                getClass().getResource("landing.fxml"));
        stage.setScene(new Scene(root, 400, 500));
        stage.show();
    }

    @Test
    public void testBlue() {
        clickOn("#tfSearch");
        write("blue");

        Label label = (Label) lookup("#labelCount").query();
        assertEquals(label.getText(), "1");
    }

    @After
    public void tearDown () throws Exception {
        FxToolkit.cleanupStages();
        release(new KeyCode[]{});
        release(new MouseButton[]{});
    }
}
```



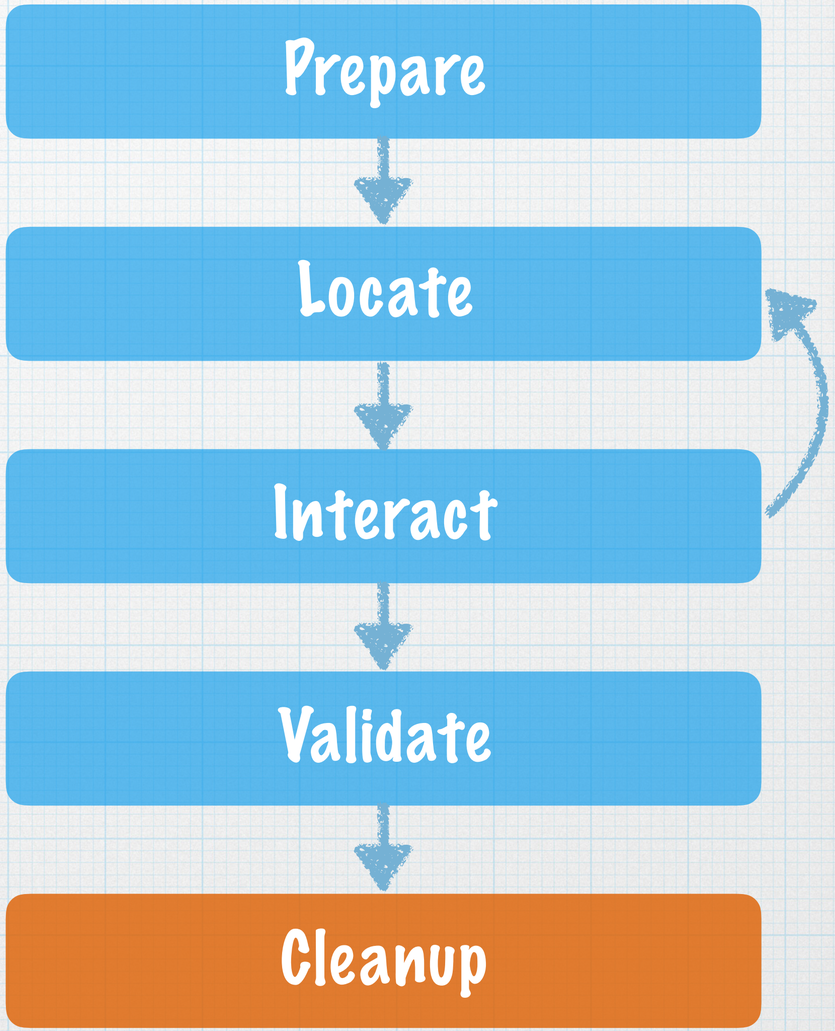
Test Phases: Validate

```
public class AppTest extends ApplicationTest {

    public void start (Stage stage) throws Exception {
        Parent root = FXMLLoader.load(
                getClass().getResource("landing.fxml"));
        stage.setScene(new Scene(root, 400, 500));
        stage.show();
    }

    @Test
    public void testBlue() {
        clickOn("#tfSearch");
        write("blue");

        Label label = (Label) lookup("#labelCount").query();
        assertEquals(label.getText(), "1");
    }

    @After
    public void tearDown () throws Exception {
        FxToolkit.cleanupStages();
        release(new KeyCode[]{});
        release(new MouseButton[]{});
    }
}
```



Test Phases: Cleanup

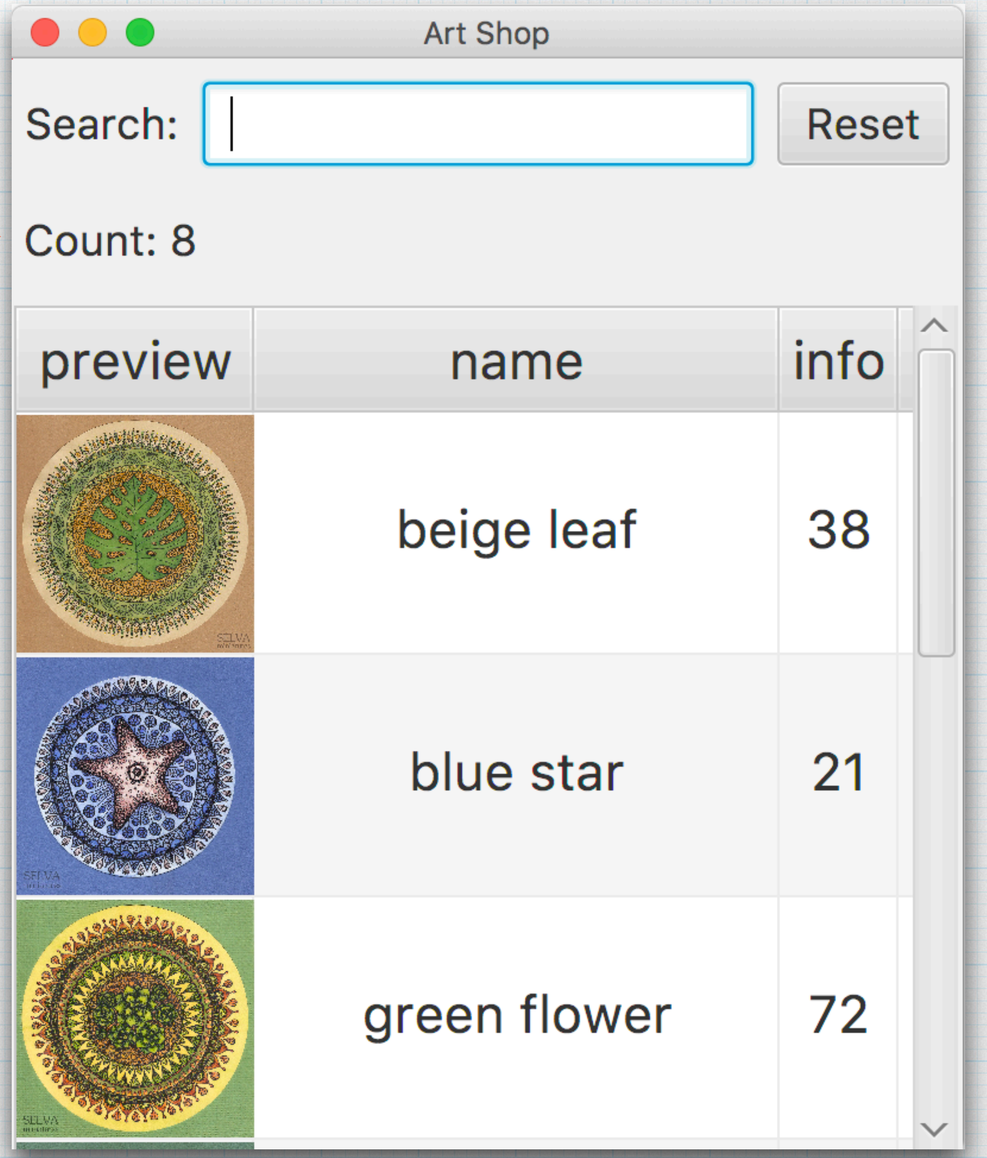
```java
public class AppTest extends ApplicationTest {

    public void start (Stage stage) throws Exception {
        Parent root = FXMLLoader.load(
                getClass().getResource("landing.fxml"));
        stage.setScene(new Scene(root, 400, 500));
        stage.show();
    }

    @Test
    public void testBlue() {
        clickOn("#tfSearch");
        write("blue");

        Label label = (Label) lookup("#labelCount").query();
        assertEquals(label.getText(), "1");
    }

    @After
    public void tearDown () throws Exception {
        FxToolkit.cleanupStages();
        release(new KeyCode[]{});
        release(new MouseButton[]{});
    }
}
```



Locate

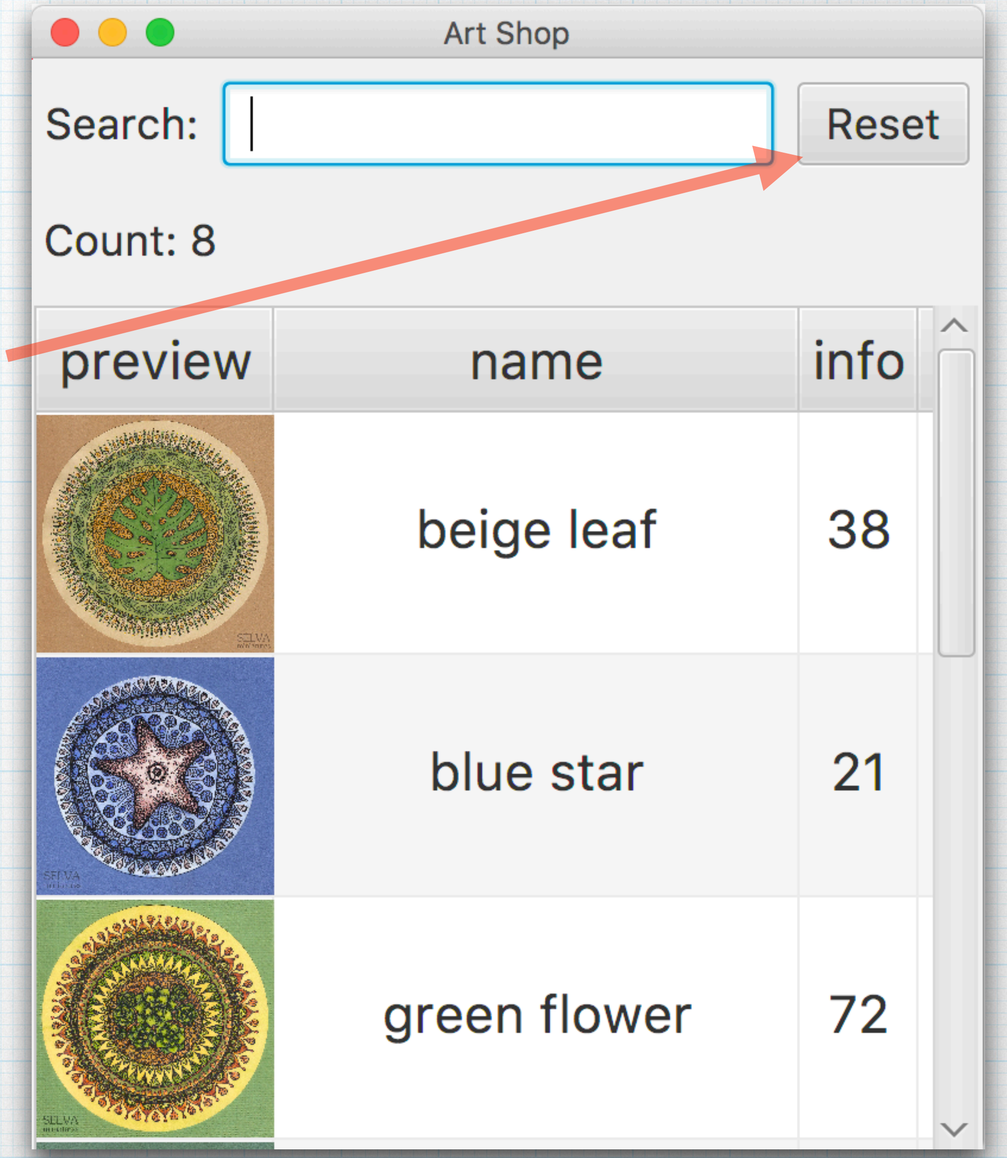
Locate by `org.testfx.api.FxRobotInterface`

```
// by fx id
lookup("#labelCount").query()
```



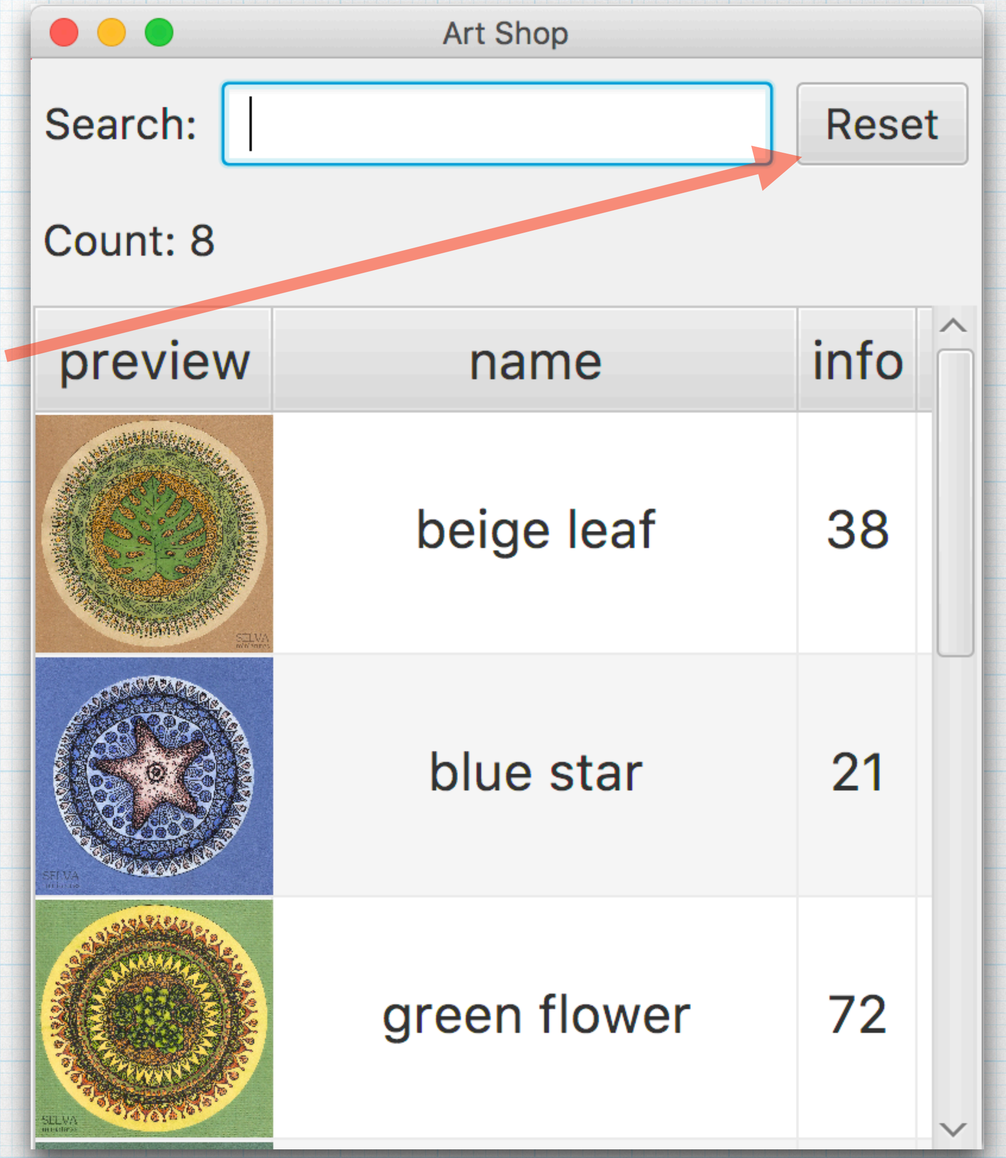
Locate by `org.testfx.api.FxRobotInterface`

```
// by fx id
lookup("#labelCount").query()

// find first thing with text "Reset"
lookup("Reset").query()
```



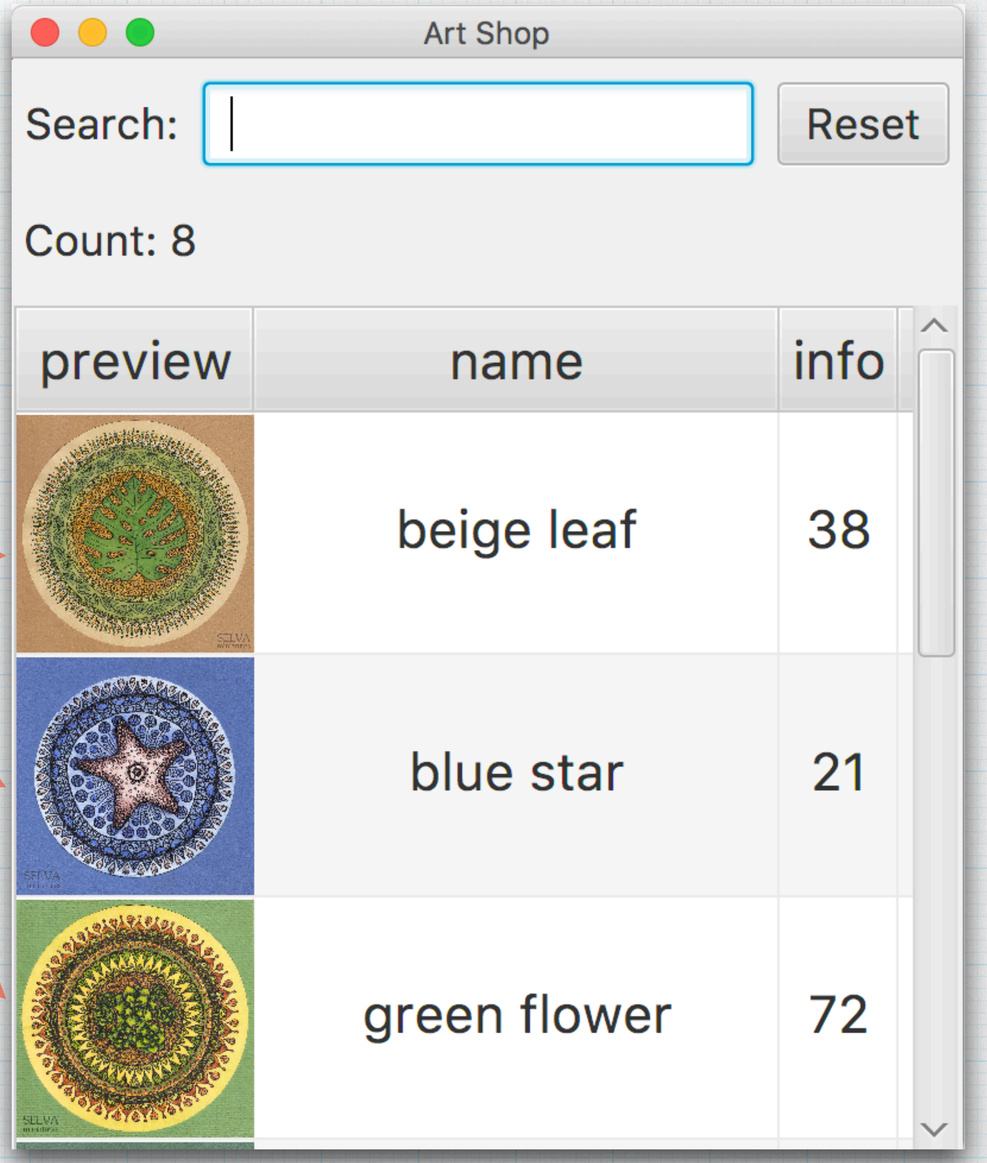
Locate by `org.testfx.api.FxRobotInterface`

```
// by fx id
lookup("#labelCount").query()

// find the first thing with the text "Reset"
lookup("Reset").query()

// lookup by Predicate
lookup((Button b) -> b.getParent() instanceof HBox).queryButton();
```



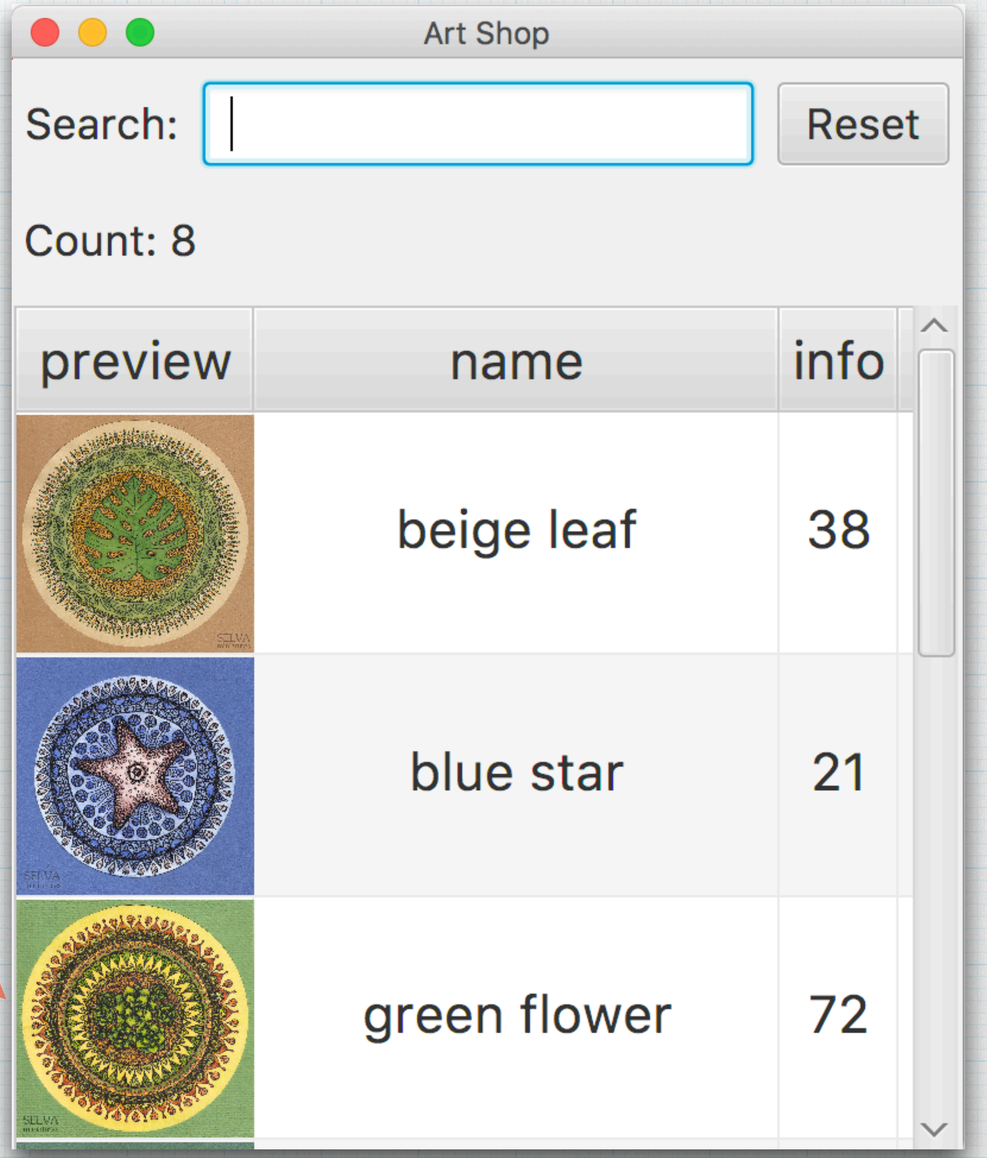
Locate by `org.testfx.api.FxRobotInterface`

```
// by fx id
lookup("#labelCount").query()

// find the first thing with the text "Reset"
lookup("Reset").query()

// lookup by Predicate
lookup((Button b) -> b.getParent() instanceof HBox).queryButton();

// by css class - returns a range of objects
lookup(".table-row-cell").query()
```



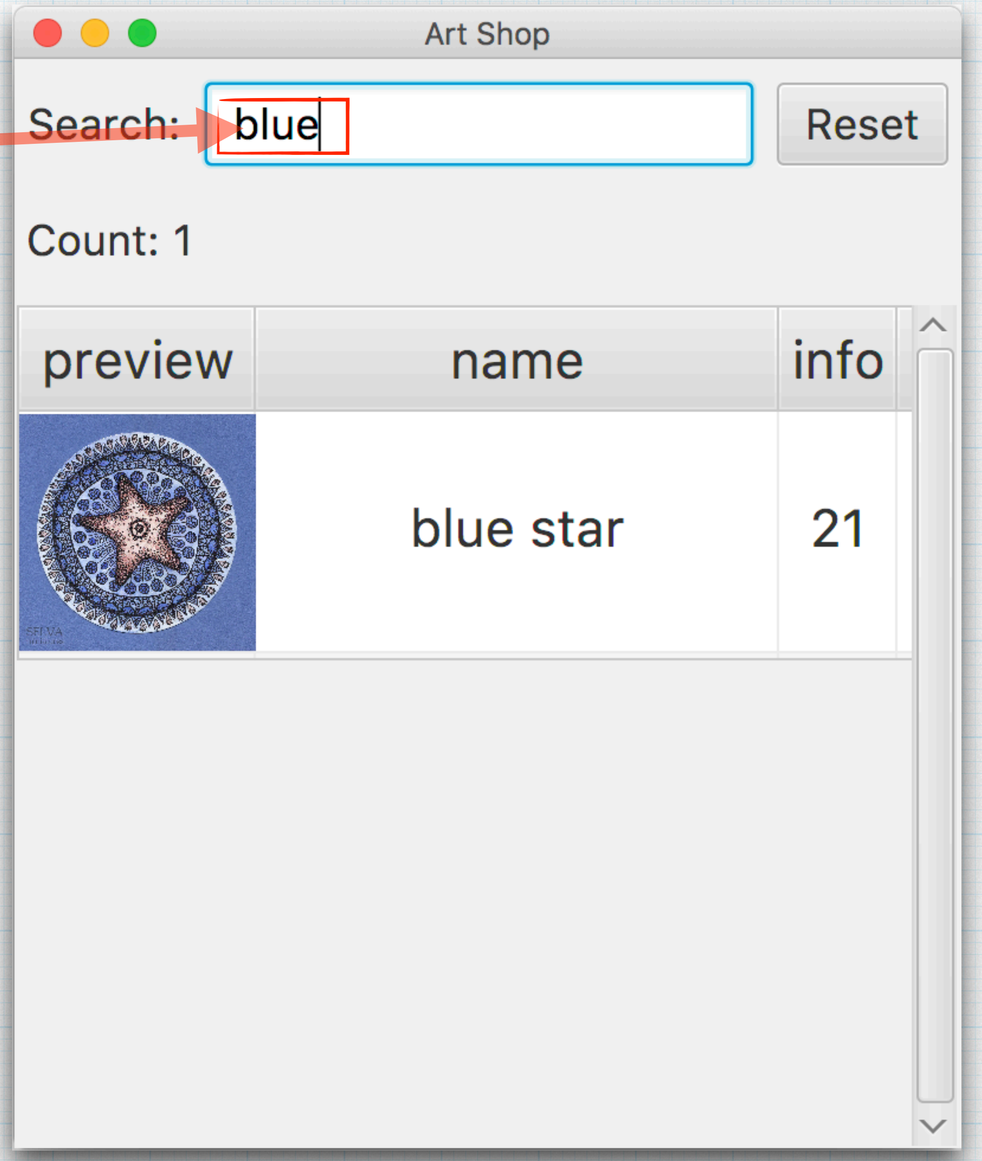
Locate by `org.testfx.api.FxRobotInterface`

```
// by fx id
lookup("#labelCount").query()

// // find the first thing with the text "Reset"
lookup("Reset").query()

// lookup by Predicate
lookup((Button b) -> b.getParent() instanceof HBox).queryButton();

// by css class - returns a range of objects
lookup(".table-row-cell").query()

// choose 3rd of this type
lookup(".table-row-cell").nth(2).query();
```



Locate by `org.testfx.api.FxRobotInterface`

```
// hierarchical lookups
lookup("#tfSearch").lookup(".text").query();
```



Locate — Validate

```
// throws EmptyNodeQueryException
lookup("#the-is-no-such-thing").query();


// return Optional
lookup("#the-is-no-such-thing").tryQuery();
```

Interact

Mouse Interactions

```
clickOn("Reset", MouseButton.SECONDARY);

doubleClickOn();

rightClickOn();

moveTo("#tableImages");

drag(100,100, MouseButton.PRIMARY).dropTo("#tableImages");

scroll(10, HorizontalDirection.LEFT);
```

Keyboard Interactions

```
write("Hello");


write("slowly", 500); // 500ms between each letter

// simultaneously
push(KeyCode.CONTROL, KeyCode.ALT, KeyCode.DELETE);

// one by one
type(KeyCode.CONTROL, KeyCode.ALT, KeyCode.DELETE);


// drag with a shift
press(KeyCode.SHIFT).drag().dropTo("tableImages").release(KeyCode.SHIFT);
```

Validation

Synchronization

* UI is asynchronous
* plain `Thread.sleep()` is not enough:
  * either your pause it too short and the test is flaky
  * or pause is long enough but your test runs for eternity

Synchronization in TestFX

Solution: check in a loop with small pauses

```
// waits for FX event queue to clear
// called after every operation
// currently default for 5 attempts with 10ms pauses
WaitForAsyncUtils.waitForFxEvents();


// custom wait, which checks in a loop
WaitForAsyncUtils.waitFor(10, TimeUnit.SECONDS, () -> {
    // this check will be called every 200ish ms
    final Optional<Node> opNode = lookup("#btn").tryQuery();

    return opNode.isPresent();
});
```


Todo: make two slides with a scheme

API validation

* every action in TestFX is validation
* you can call JavaFX API methods to verify UI state
  * combining with JUnit asserts
  * or hamcrest matchers
  * or AssertJ

Hamcrest matchers

```
// instead of
Label label = (Label) lookup("#labelCount").query();
assertEquals(label.getText(), "1");

// you can write
FxAssert.verifyThat("#labelCount", LabeledMatchers.hasText("1"));

// and some static-imports-fu

verifyThat("#labelCount", hasText("1"));
```

Hamcrest matchers

```
// instead of
Label label = (Label) lookup("#labelCount").query();
assertEquals(label.getText(), "1");

// you can write
FxAssert.verifyThat("#labelCount", LabeledMatchers.hasText("1"));

// and some static-imports-fu

verifyThat("#labelCount", hasText("1"));
```

Hamcrest matchers

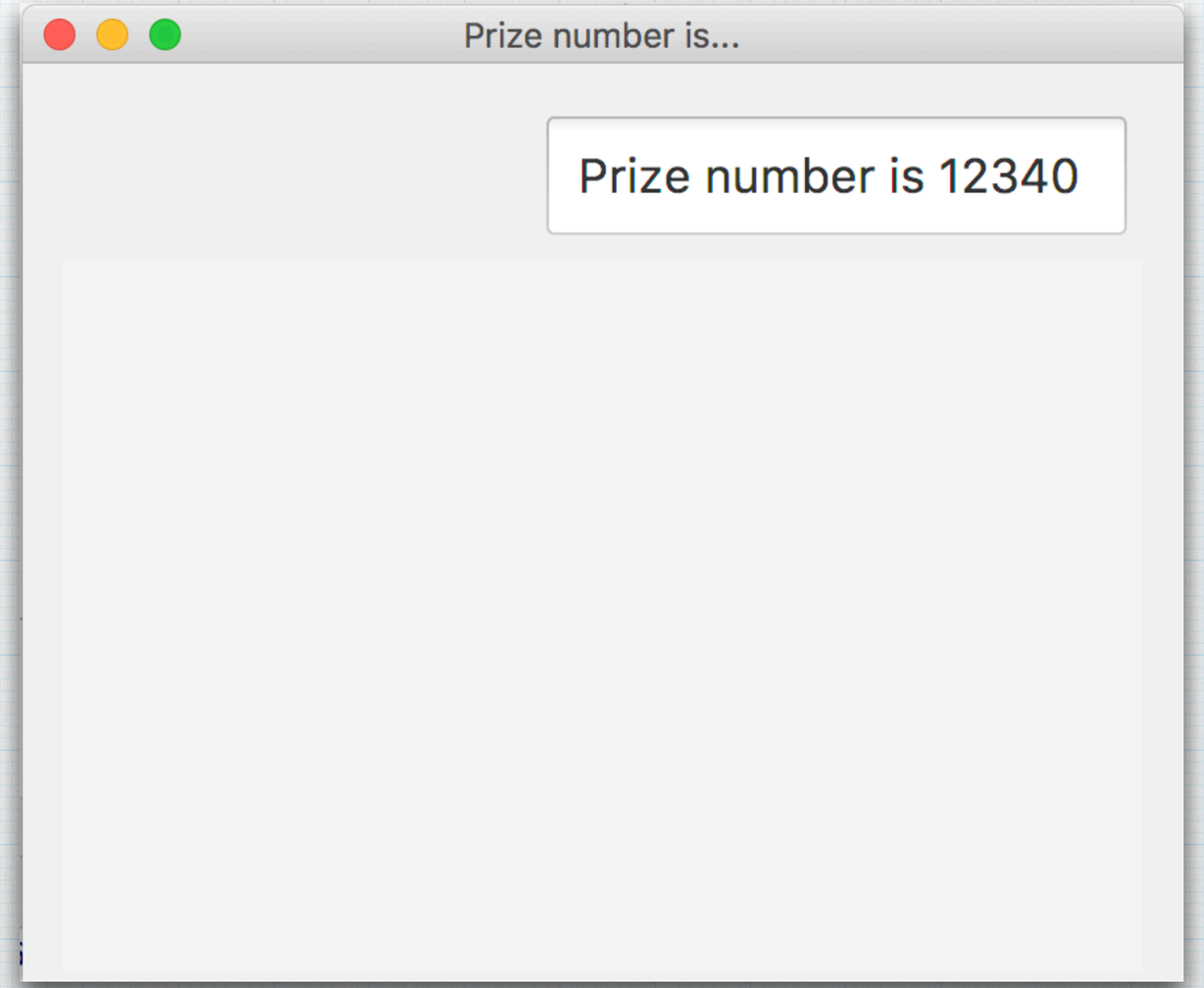
* 15 different classes at `org.testfx.matcher`
* 20 different options for `FXAssert .verifyThat()`

more examples?
see https://github.com/TestFX/

When API validation is not enough?

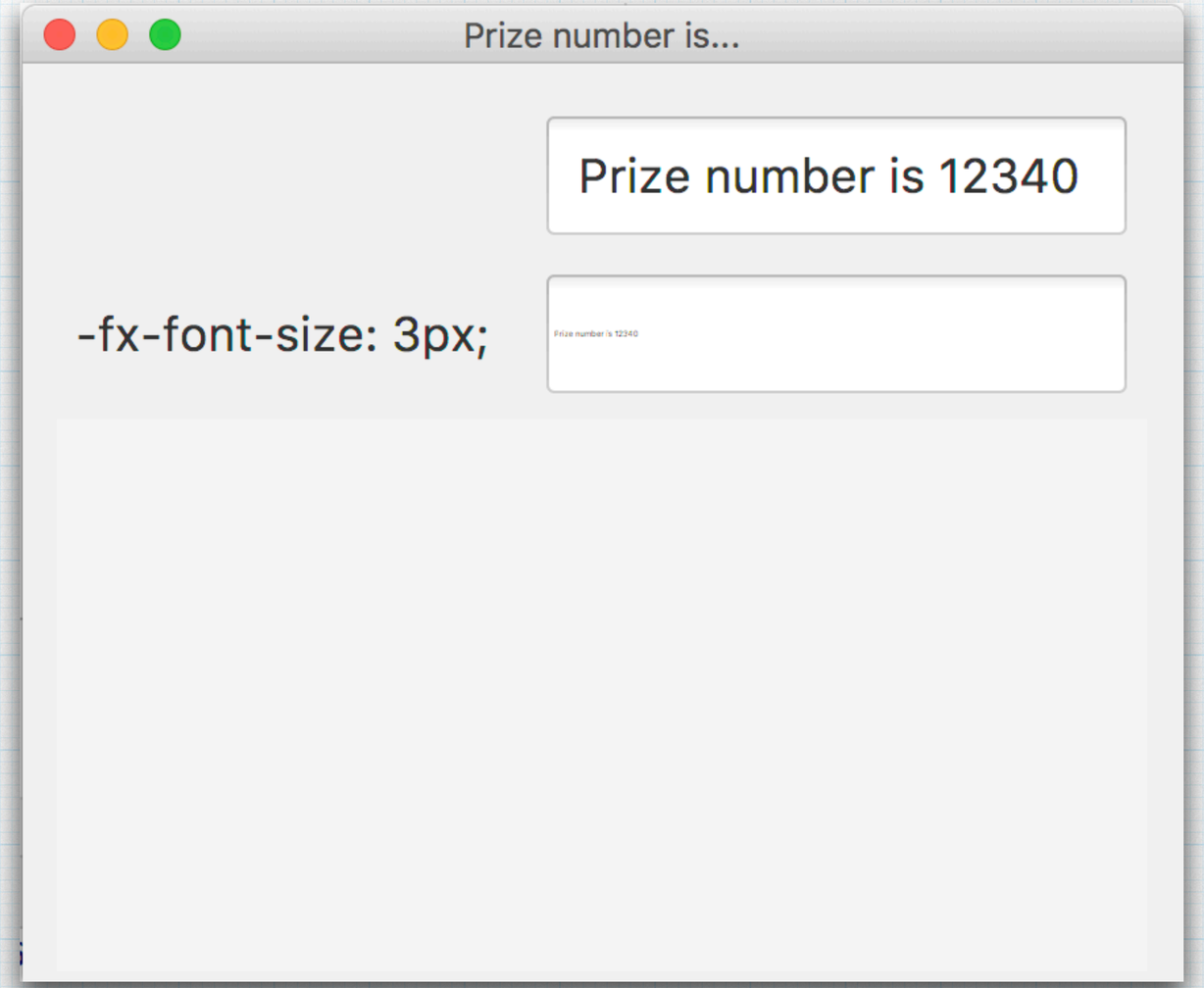
When API validation is not enough?

```
assertEquals(
  textField.getText(),
  resources.getString("prize")
                          + " 12340");
```



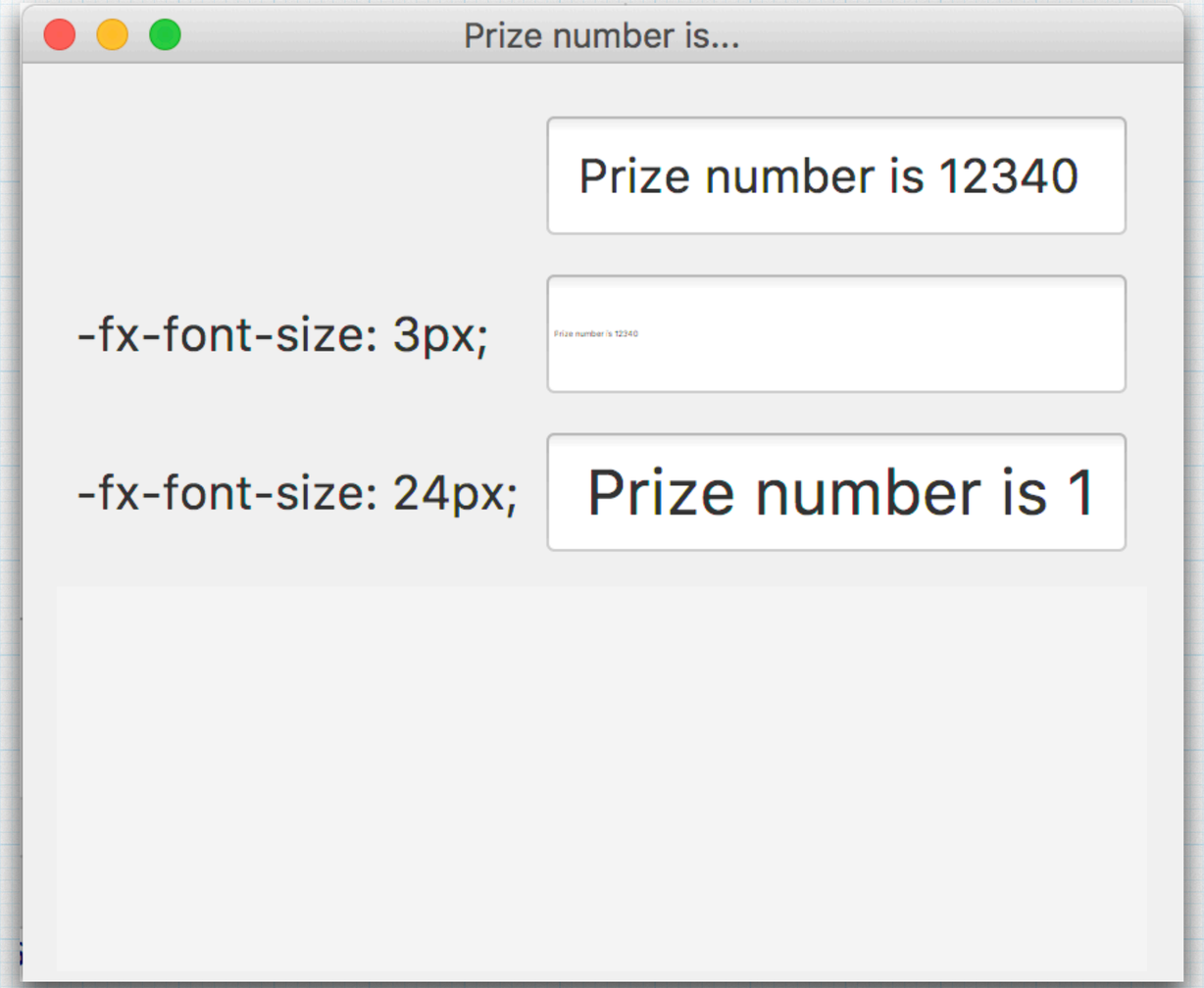
When API validation is not enough?

```
assertEquals(
  textField.getText(),
  resources.getString("prize")
                          + " 12340");
```



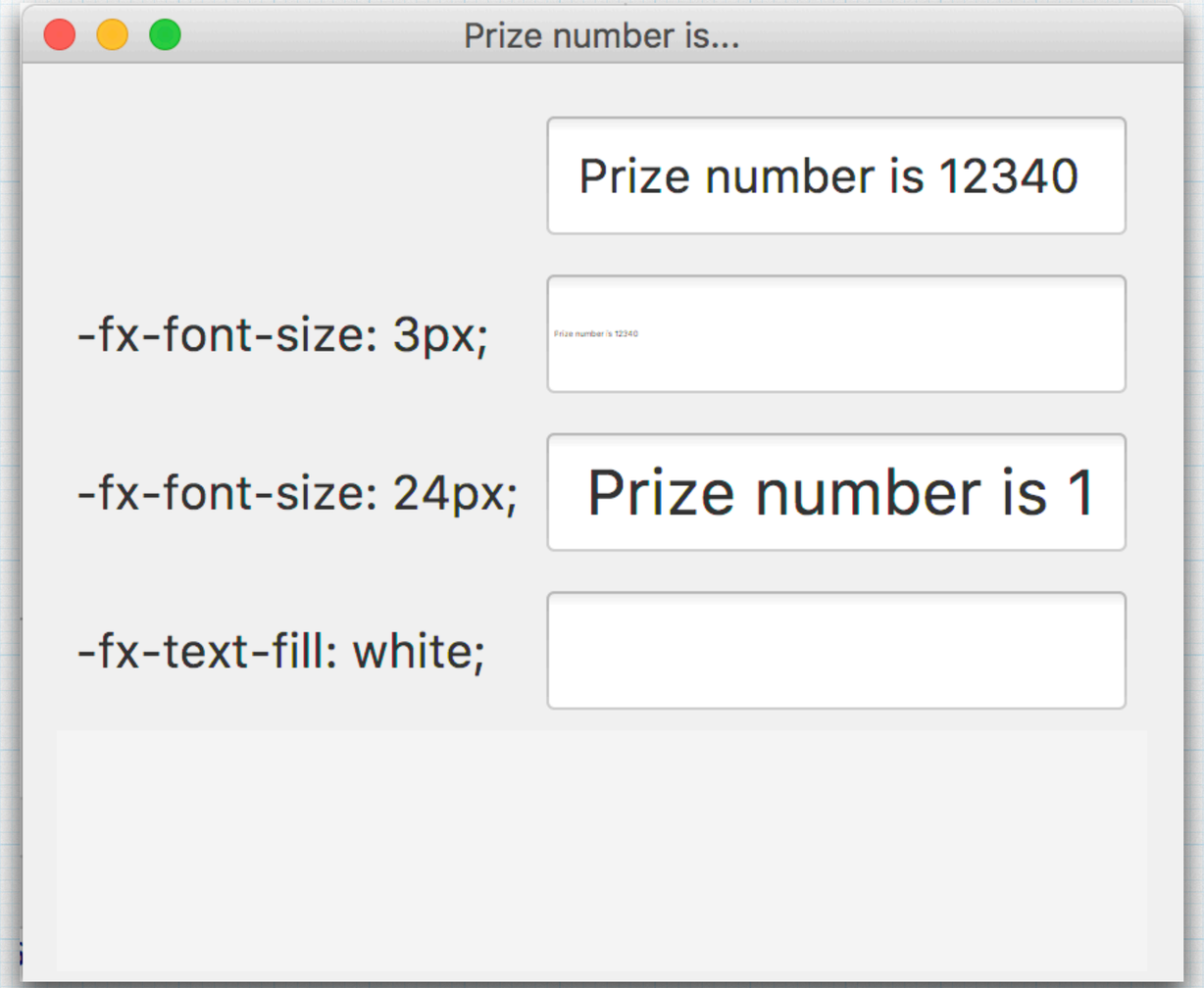
When API validation is not enough?

```
assertEquals(
  textField.getText(),
  resources.getString("prize")
                          + " 12340");
```



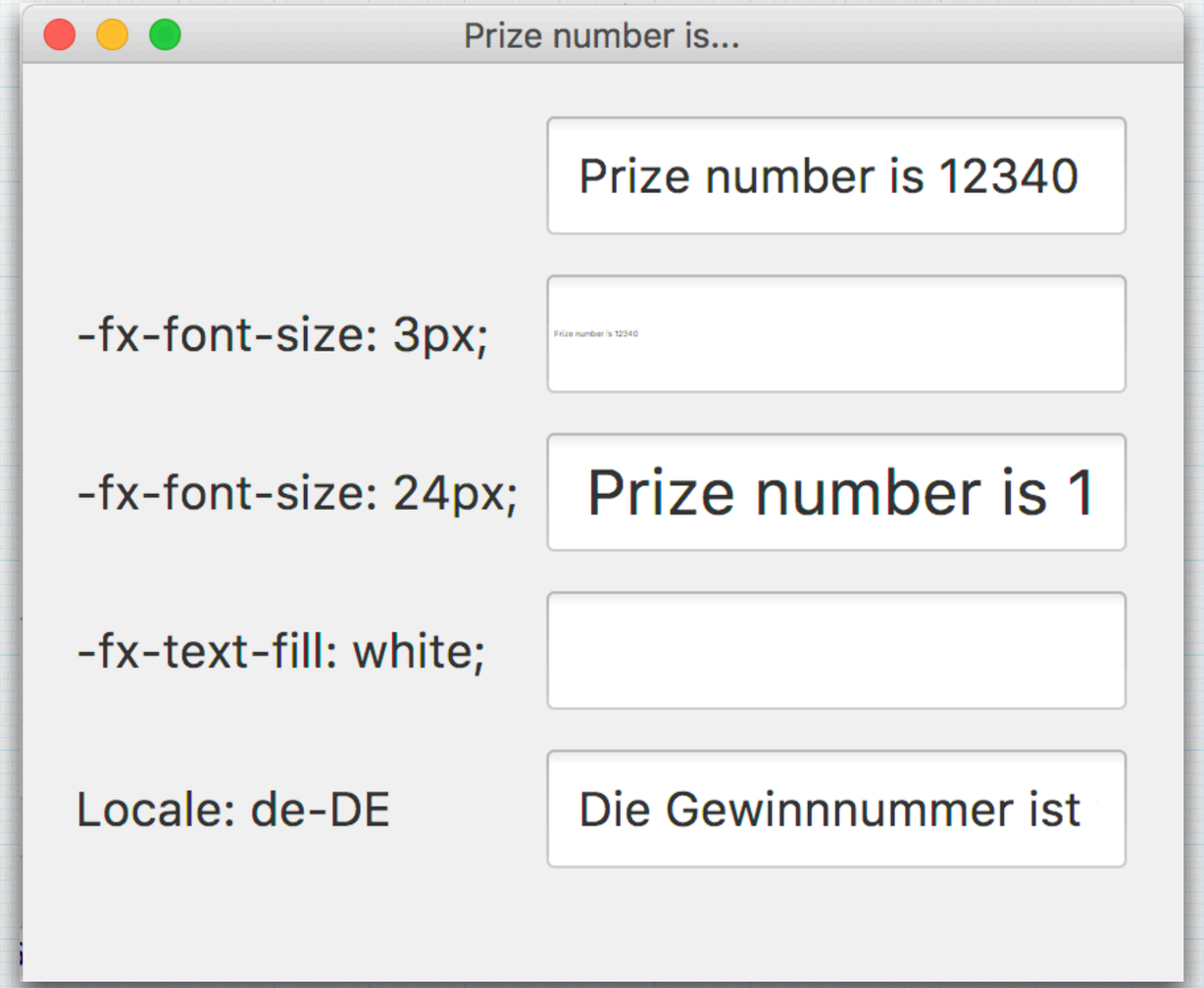
When API validation is not enough?

```
assertEquals(
  textField.getText(),
  resources.getString("prize")
                          + " 12340");
```

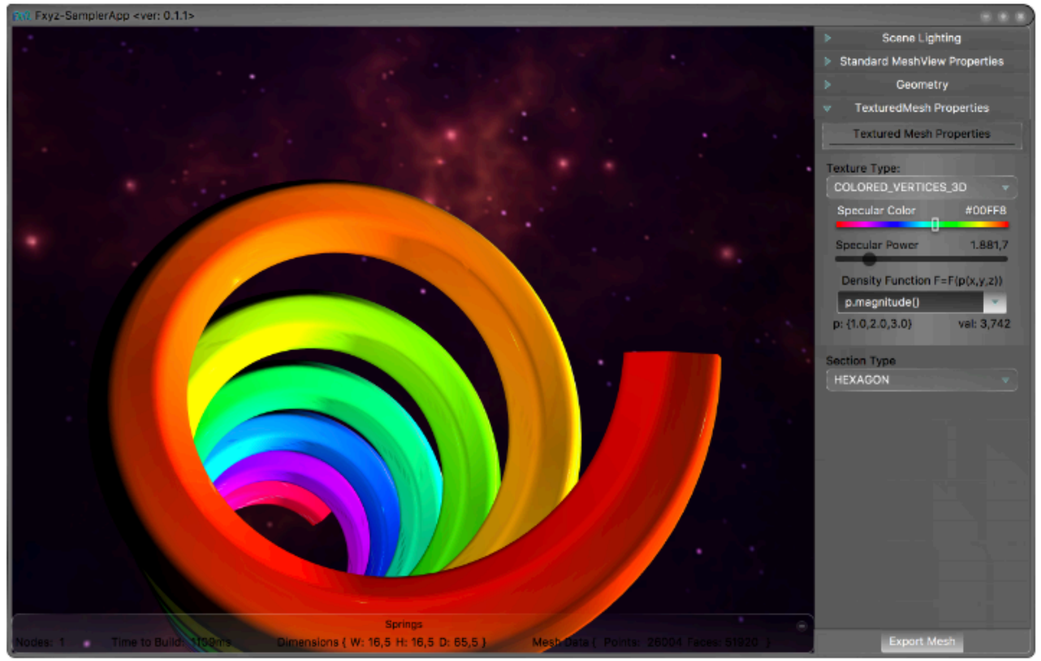


When API validation is not enough?

```
assertEquals(
  textField.getText(),
  resources.getString("prize")
                          + " 12340");
```

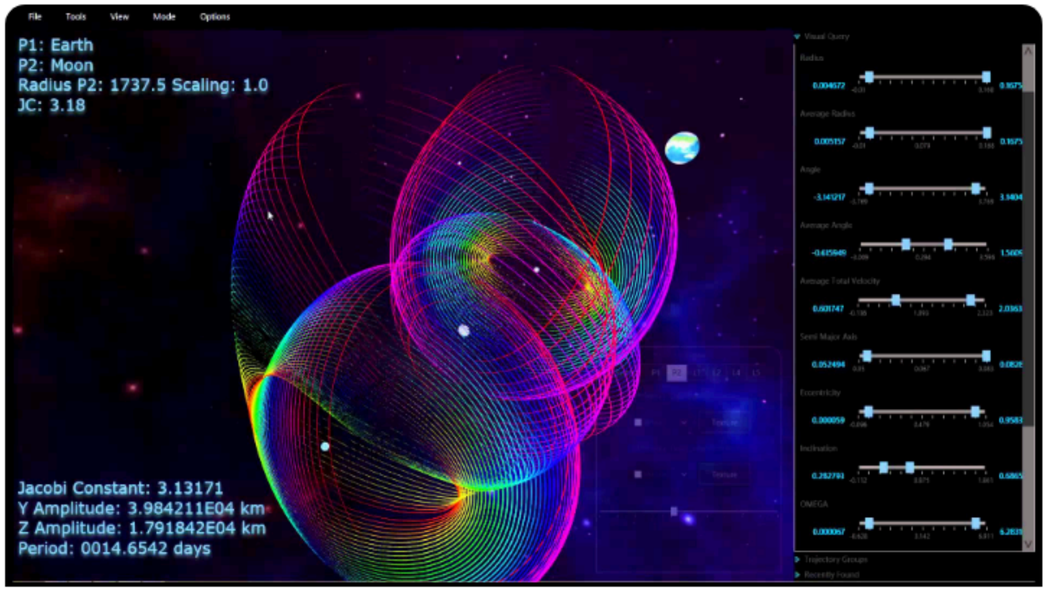


When API validation is not enough?



FXyz project

3D Visualization and Components library



DSTE

The Deep Space Trajectory Explorer

images from https://openjfx.io/

Screenshots

* Using golden screenshots is a big investment
* You may have to keep set for different platforms, rendering engines, Java version, look-and-feels, etc
* Screenshots suffer from the modern UI elements

Main enemies of a screenshot testing

* Designers!

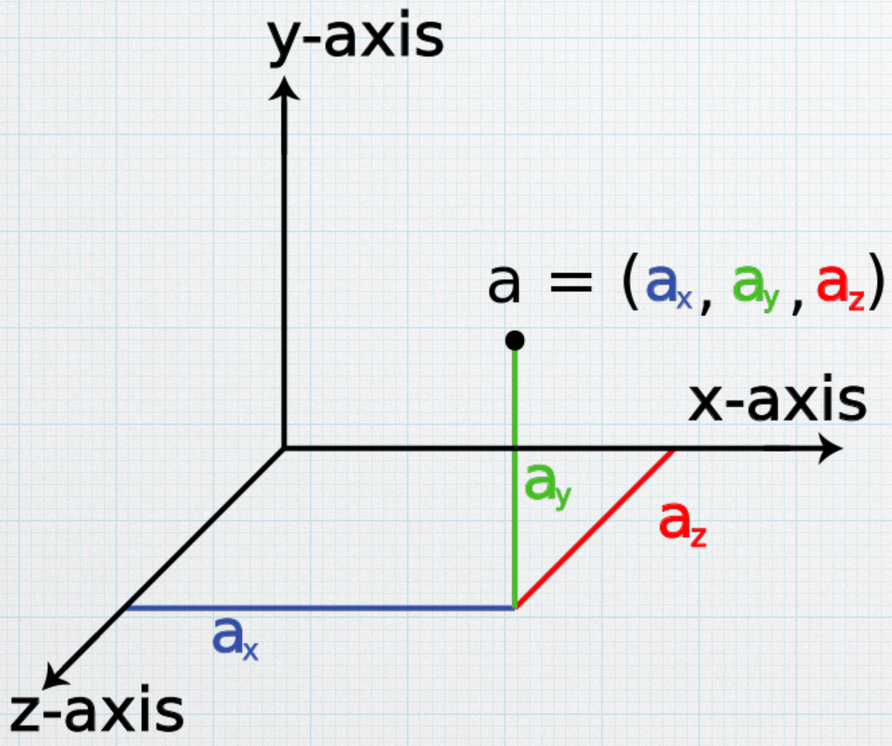
Main enemies of a screenshot testing

* Designers!
* UI changes
* Modern UI effects:
  * Shadows
  * Shaders
  * Gradients
  * Various fonts

Main enemies of a screenshot testing

* Text
* Subpixel rendering

Pixel distance



* you can calculate "distance" between matching pixes
* Average distance works pretty well as a threshold

image by Falcorian - CC BY-SA 2.5

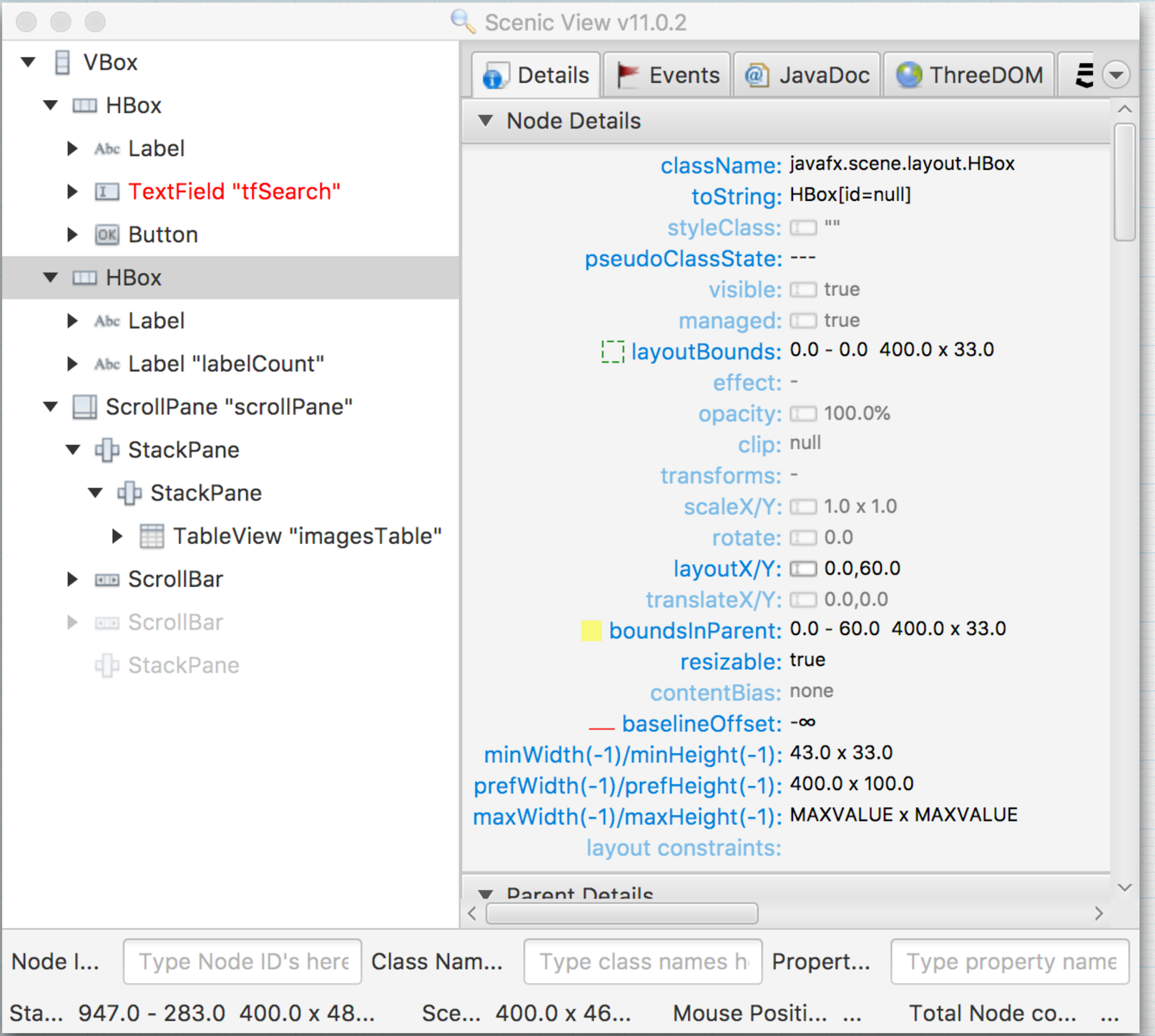
Pixel distance based diffs

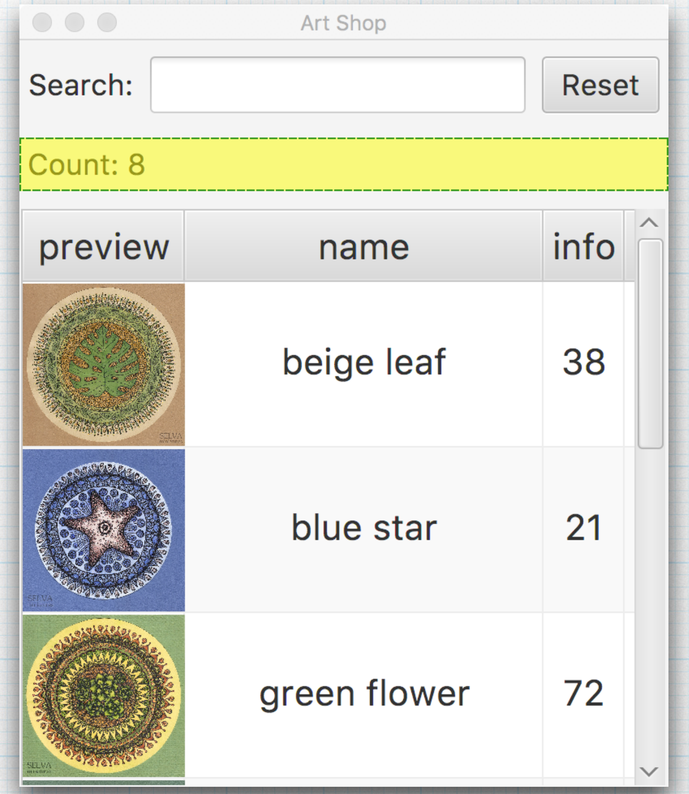
Tools

Monocle — headless test execution

* Doesn't block you during test run
* Ideal for CI, cloud or docker execution
* Allows parallel tests execution
* Great for unit tests
* Maven: org.testfx:openjfx-monocle

Scenic View


Scenic View v11.0.2
VBox
HBox
Label
TextField "tfSearch"
Button
HBox
Label
Label "labelCount"
ScrollPane "scrollPane"
StackPane
StackPane
TableView "imagesTable"
ScrollBar
ScrollBar
StackPane
Details
Events
JavaDoc
ThreeDOM
Node Details
className: javafx.scene.layout.HBox
toString: HBox[id=null]
styleClass: ""
pseudoClassState: ---
visible: true
managed: true
layoutBounds: 0.0 - 0.0 400.0 x 33.0
effect: -
opacity: 100.0%
clip: null
transforms: -
scaleX/Y: 1.0 x 1.0
rotate: 0.0
layoutX/Y: 0.0,60.0
translateX/Y: 0.0,0.0
boundsInParent: 0.0 - 60.0 400.0 x 33.0
resizable: true
contentBias: none
baselineOffset: -∞
minWidth(-1)/minHeight(-1): 43.0 x 33.0
prefWidth(-1)/prefHeight(-1): 400.0 x 100.0
maxWidth(-1)/maxHeight(-1): MAXVALUE x MAXVALUE
layout constraints:
Parent Details
Node I...
Type Node ID's here
Class Nam...
Type class names h
Propert...
Type property name
Sta... 947.0 - 283.0 400.0 x 48...
Sce... 400.0 x 46...
Mouse Positi... ...
Total Node co... ...


Art Shop
Search:
Reset
Count: 8
preview
name
info
beige leaf
38
blue star
21
green flower
72

Q&A

Sergey Grinev
Azul Systems, Inc.

sergey@azul.com
@SergeyGrinev