GroovyFX:

Making JavaFX Groovy

Jim Clarke
Oracle Corporation
jim.clarke@oracle.com
@JimClarke5 on Twitter
http://www.codetown.us/profiles/blog/list?tag=groovyfx

Dierk König
Canoo Engineering AG
dierk.koening@canoo.com
@mittie on Twitter
http://www.canoo.com/blog

First App


```
import static groovyx.javafx.GroovyFX.start

start {
  stage(title: "GroovyFX @ JavaONE", show: true ) {
    scene(fill: groovyblue, width: 420, height: 420 ) {
      text("Hello World!", layoutY: 50, font: ”bold 48pt serif")
    }
  }
}
```

First App (Java)


```
import javafx.application.Application;
import javafx.scene.Group;
import javafx.scene.Scene;
import javafx.scene.paint.Color;
import javafx.scene.text.Font;
import javafx.scene.text.FontWeight;
import javafx.scene.text.Text;
import javafx.stage.Stage;

public class HelloWorld extends Application {
    @Override
    public void start(Stage primaryStage) {
        Font font = Font.font("serif", FontWeight.BOLD, 48);

        Text text = new Text("Hello World!");
        text.setLayoutY(50);
        text.setFont(font);

        Group parent = new Group(text);
        Scene scene = new Scene(parent, 420, 420, Color.rgb(99, 152, 170));

        primaryStage.setScene(scene);
        primaryStage.setTitle("JavaFX @ JavaONE");
        primaryStage.show();
    }

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

```
import static groovyx.javafx.GroovyFX.start

GroovyFX.start {

  stage(title: "GroovyFX @ JavaONE", show: true ) {
    scene(fill: groovyblue, width: 420, height: 420 ) {
      text("Hello World!", layoutY: 50,
                    font: "bold 48pt serif")
    }
  }
}
```

The SceneGraph


```
stage(title: "GroovyFX BorderPane Demo", show: true) {
    scene(fill: groovyblue, width: 650, height:450) {
        borderPane {
            top(align: CENTER, margin: [10,0,10,0]) {
              button("Top Button")
            }
            right(align: CENTER, margin: [0,10,0,1]) {
              toggleButton("Right Toggle")
            }
            left(align: CENTER, margin: [0,10]) {
              checkBox("Left Check")
            }
            bottom(align: CENTER, margin: 10) {
              textField("Bottom TextField")
            }
            label("Center Label")
        }
    }
}
```

What you see


GroovyFX @ JavaOne
Top Button
Left Check
Center Label
Right Toggle
Bottom TextField

Colors - Gradients


```
stage(title: "GroovyFX @ JavaONE", show: true) {
  scene(width: 420, height:420) {
    fill linearGradient(
         start: [0, 0.4], end: [0.9, 0.9],
         stops: [groovyblue, rgb(153,255,255)])
```

==========

```
    fill radialGradient(radius: 0.6, center:[0.5, 0.5],
         focusDistance: 0.6, focusAngle: -65,
         stops: [groovyblue, '#99FFFF'])
```

What you see

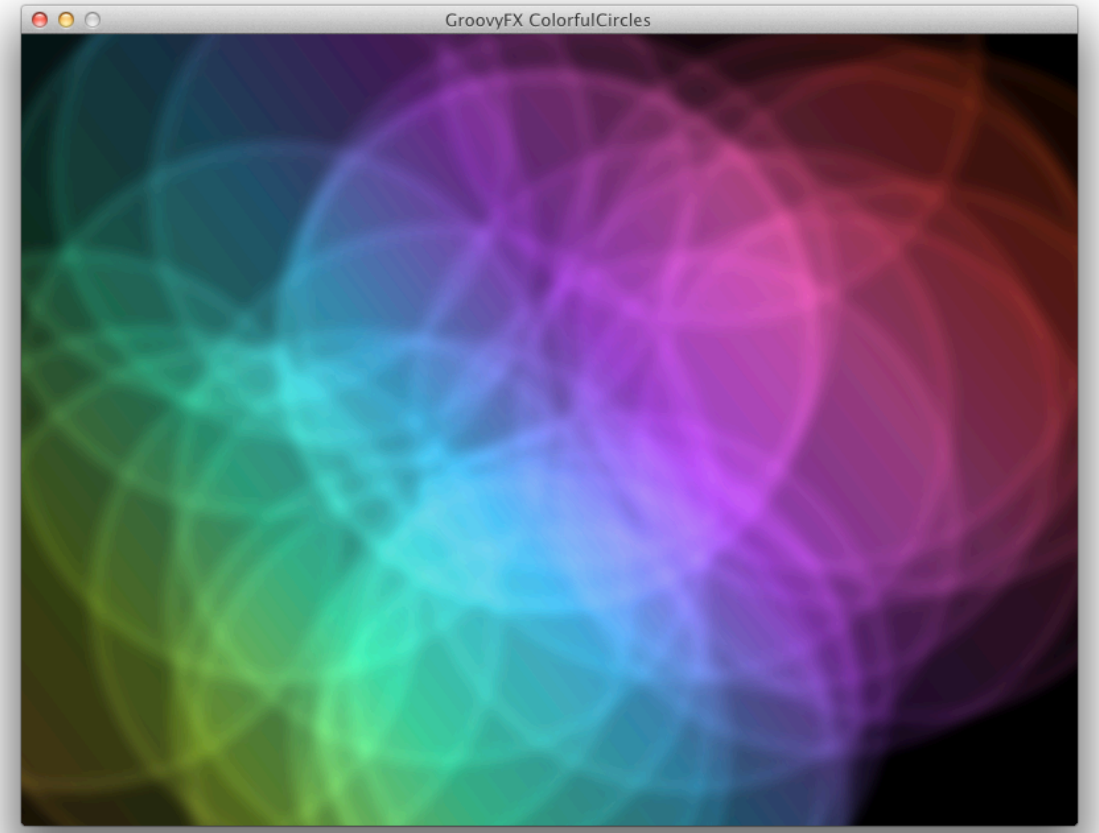

GroovyFX @ JavaOne
Top Button
Left Check
Center Label
Right Toggle
Bottom TextField


GroovyFX @ JavaOne
Top Button
Left Check
Center Label
Right Toggle
Bottom TextField

Graphics & Effects


```
stage(title: 'GroovyFX ColorfulCircles', resizable: false, show: true) {
  scene(width: 800, height: 600, fill: black) {
    group {
      group {
        30.times {
          circle(radius: 200, strokeWidth: 4, strokeType: 'outside',
                 fill: rgb(255, 255, 255, 0.05),
                 stroke: rgb(255, 255, 255, 0.16))
        }
        effect boxBlur(width: 10, height: 10, iterations: 3)
      }
      rectangle(width: 800, height: 600, blendMode: 'overlay') {
        def stops = ['#f8bd55', '#c0fe56', '#5dfbc1', '#64c2f8',
                     '#be4af7', '#ed5fc2', '#ef504c', '#f2660f']
        fill linearGradient(start: [0f, 1f], end: [1f, 0f], stops: stops)
      }
    }
  }
}
```

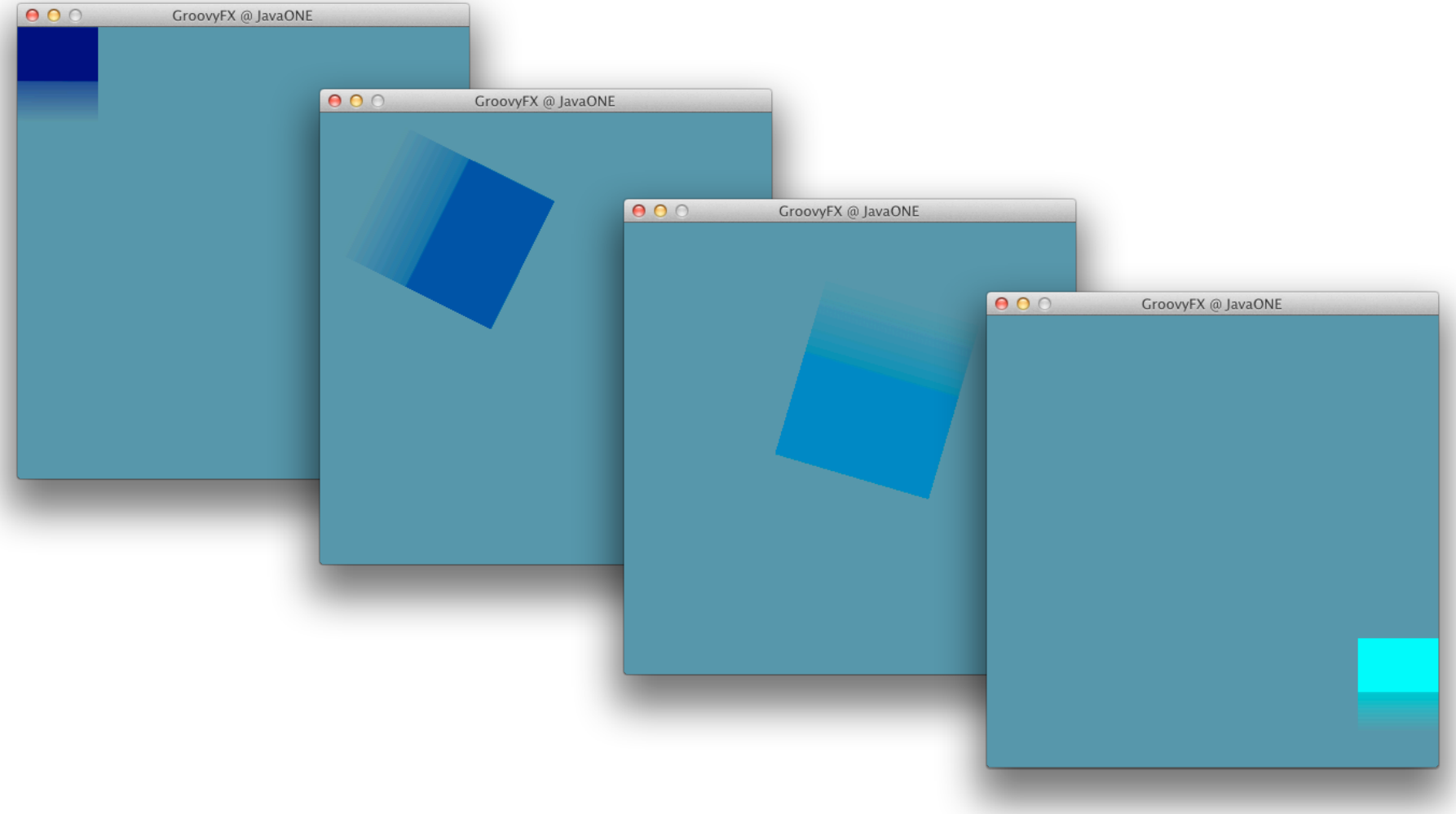
What you see



Animation


```
stage(title: "GroovyFX @ JavaONE", show: true) {
    scene(fill: groovyblue, width: 420, height:420) {
        rect = rectangle(width: 75, height: 50, fill: navy) {
            effect reflection()
            animation = parallelTransition(cycleCount: indefinite, autoReverse: true) {
                translateTransition(4.s, fromX: 0, fromY: 0, toX: 420-75, toY: 300)
                rotateTransition(4.s, toAngle: 360)
                fillTransition(4.s, from: navy, to: cyan)
                sequentialTransition() {
                    scaleTransition(2.s, from: 1.0, to: 2)
                    scaleTransition(2.s, from: 2, to: 1.0)
                }
            } animation.playFromStart();
        }
    }
}
```

What you see

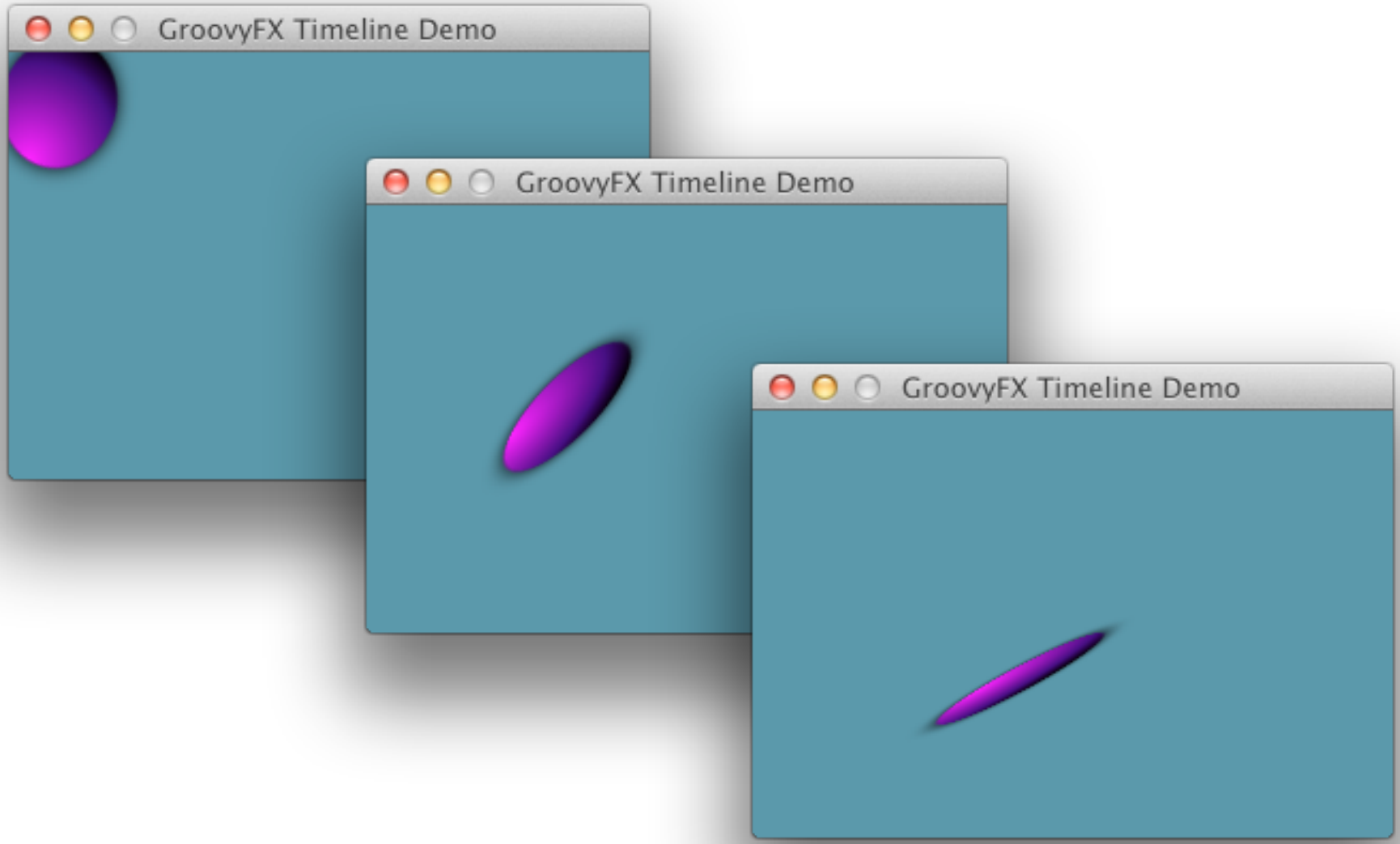

GroovyFX @ JavaONE
GroovyFX @ JavaONE
GroovyFX @ JavaONE
GroovyFX @ JavaONE

Animation - Timeline

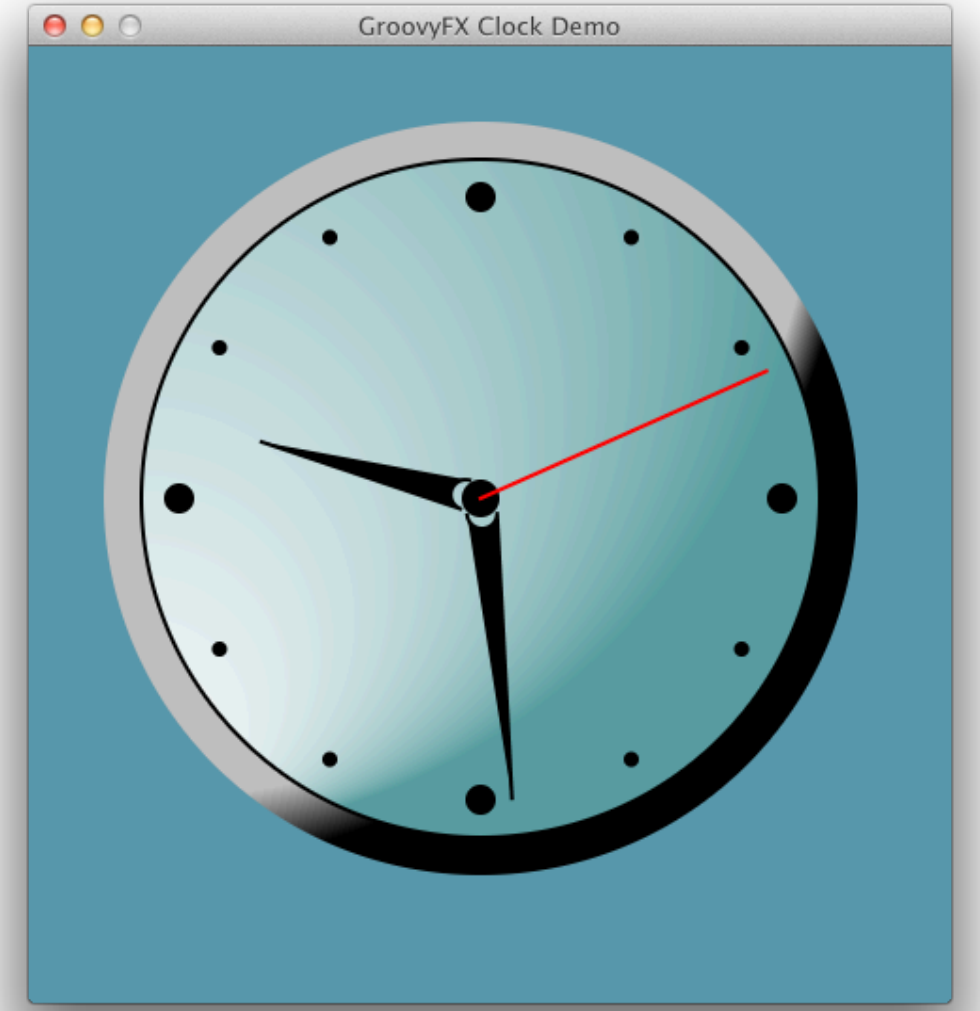
GroovyFX

```
start {
  stage(title: "GroovyFX Timeline Demo",  show: true) {
         scene(fill: groovyblue, width: 300, height:200) {
            circle(id: “circ”, radius: 25, rotationAxis: [0,1,1]) {
                fill radialGradient(radius: 1, center:[0.5, 1],
                    stops:[[0, magenta], [0.75, indigo], [1, black]])
                effect dropShadow()
            }
         }
         timeline (cycleCount: indefinite, autoReverse: true) {
            at(2.s) {
                change(circ.layoutXProperty()) { to 150; tween easeboth }
                change(circ.layoutYProperty()) { to 150; tween easeboth }
                change(circ.scaleXProperty()) { to 2 }
                change(circ.scaleYProperty()) { to 2 }
                change(circ.rotateProperty()) { to 180 }
            }
        }.playFromStart()
    }
}
```

What you see


GroovyFX Timeline Demo
GroovyFX Timeline Demo
GroovyFX Timeline Demo

GroovyFX Binding



Binding – Analog Clock


```
@FXBindable
class Time {
    Integer hours
    Integer minutes
    Integer seconds

    Double hourAngle
    Double minuteAngle
    Double secondAngle

    public Time() {
        // bind the angle properties to the clock time
        hourAngleProperty.bind((hours() * 30.0) + (minutes() * 0.5))
        minuteAngleProperty.bind(minutes() * 6.0)
        secondAngleProperty.bind(seconds() * 6.0)

        // Set the initial clock time
        def calendar = Calendar.instance
        hours = calendar.get(Calendar.HOUR)
        minutes = calendar.get(Calendar.MINUTE)
        seconds = calendar.get(Calendar.SECOND)
    }
    public void addOneSecond() {
        seconds = (seconds + 1) % 60
        if (seconds == 0) {
            minutes = (minutes + 1)  % 60
            if (minutes == 0)
                hours = (hours + 1) % 12
        }
    }
}
```

JavaFX Properties

GroovyFX
Binding DSL

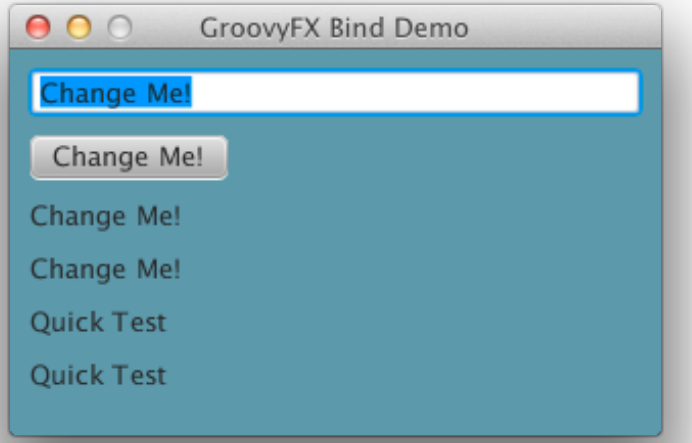
Binding – Analog Clock

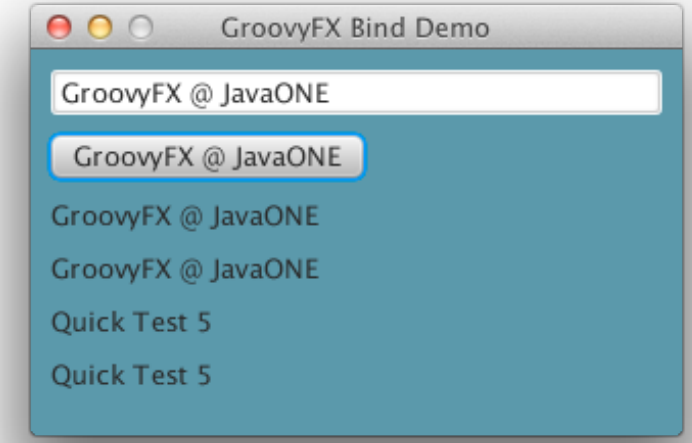

```
// hour hand
path(fill: black) {
    rotate(angle: bind(time.hourAngle()))
    moveTo(x: 4, y: -4)
    arcTo(radiusX: -1, radiusY: -1, x: -4, y: -4)
    lineTo(x: 0, y: -radius / 4 * 3)
}
// minute hand
path(fill: black) {
    rotate(angle: bind(time.minuteAngle()))
    moveTo(x: 4, y: -4)
    arcTo(radiusX: -1, radiusY: -1, x: -4, y: -4)
    lineTo(x: 0, y: -radius)
}
// second hand
line(endY: -radius - 3, strokeWidth: 2, stroke: red) {
    rotate(angle: bind(time.secondAngle()))
}
```

More Binding


```
class QuickTest {
    @FXBindable String qtText = "Quick Test”

    private int clickCount = 0
    def onClick = {
        qtText = "Quick Test ${++clickCount}"
    }
}
start {
    def qt = new QuickTest()

    stage(title: "GroovyFX Bind Demo", x: 100, y: 100, show: true) {
        scene(fill: groovyblue, width: 400, height: 400) {
            vbox(spacing: 10, padding: 10) {
                TextField tf = textField(text: 'Change Me!')
                button(text: bind(tf,'text'), onAction: {qt.onClick()})
                label(text: bind(tf.textProperty()))
                label(text: bind({tf.text}))
                label(text: bind(tf.text()))
                // Bind to POGO fields annotated with @FXBindable
                // These next two bindings are equivalent
                label(text: bind(qt, 'qtText'))
                label(text: bind(qt.qtText()))
                label(text: bind(qt.qtTextProperty()).using({"<<< ${it} >>>"}) )
            }
        }
    }
}
```

What you see


GroovyFX Bind Demo
Change Me!
Change Me!
Change Me!
Change Me!
Quick Test
Quick Test


GroovyFX Bind Demo
GroovyFX @ JavaONE
GroovyFX @ JavaONE
GroovyFX @ JavaONE
GroovyFX @ JavaONE
Quick Test 5
Quick Test 5

TableView Control

The Java Way:

```
public class Person {
  private StringProperty firstName;
  public void setFirstName(String val) { firstNameProperty().set(val); }
  public String getFirstName() { return firstNameProperty().get(); }
  public StringProperty firstNameProperty() {
    if (firstName == null)
      firstName = new SimpleStringProperty(this, "firstName");
    return firstName;
  }

  private StringProperty lastName;
  public void setLastName(String value) { lastNameProperty().set(value); }
  public String getLastName() { return lastNameProperty().get(); }
  public StringProperty lastNameProperty() {
    // etc.
  }
}
```

TableView Control

The Java Way:

```
ObservableList<Person> items = ...
TableView<Person> tableView = new TableView<Person>(items);

TableColumn<Person,String> firstNameCol =
        new TableColumn<Person,String>("First Name");

firstNameCol.setCellValueFactory(
        new Callback<CellDataFeatures<Person, String>,
                     ObservableValue<String>>() {
  public ObservableValue<String> call(CellDataFeatures<Person, String> p) {
    return p.getValue().firstNameProperty();
  }
});

tableView.getColumns().add(firstNameCol);
```

TableView Control

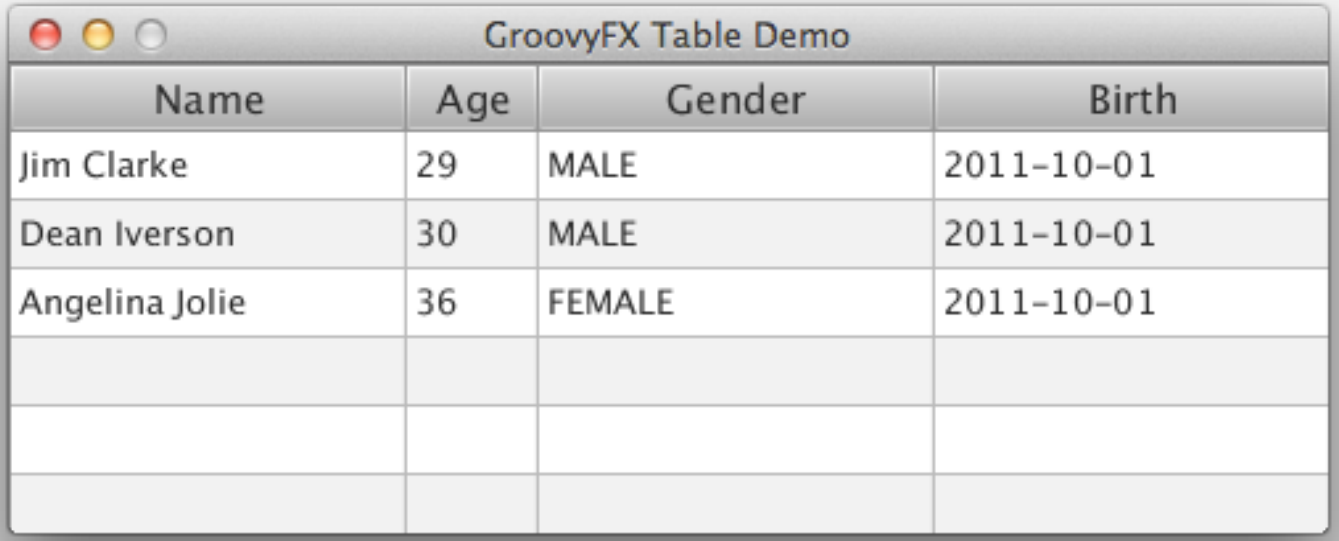

```
enum Gender { MALE, FEMALE }

@Canonical
Class Person {
  @FXBindable String name
  @FXBindable int age
  @FXBindable Gender gender
  @FXBindable Date dob
}

def persons = [
  new Person("Jim Clarke", 29, Gender.MALE, new Date()),
  new Person(”Dierk König", 30, Gender.MALE, new Date()),
  new Person("Angelina Jolie", 36, Gender.FEMALE, new Date())
]
```

TableView Control


```
def dateFormat = new SimpleDateFormat("yyyy-MM-dd");

start {
  stage(title: "GroovyFX Table Demo", show: true) {
    scene(fill: groovyblue, width: 500, height: 200) {
      tableView(items: persons) {
        tableColumn(property: "name",   text: "Name",   prefWidth: 150)
        tableColumn(property: "age",    text: "Age",    prefWidth: 50)
        tableColumn(property: "gender", text: "Gender", prefWidth: 150)
        tableColumn(property: "dob",    text: "Birth",  prefWidth: 150,
          type: Date,
          converter: { from ->
            return dateFormat.format(from)
          }
        )
      }
    }
  }
}
```

What you see


Fx
GroovyFX


GroovyFX Table Demo
Name
Age
Gender
Birth
Jim Clarke
29
MALE
2011-10-01
Dean Iverson
30
MALE
2011-10-01
Angelina Jolie
36
FEMALE
2011-10-01

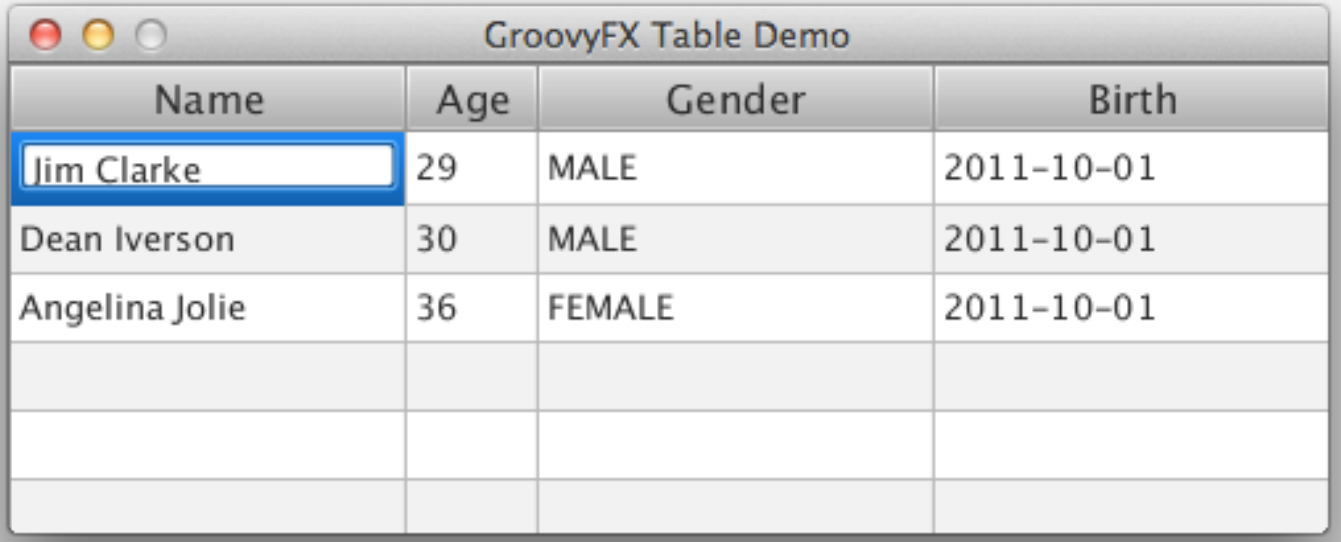
TableView Control

GroovyFX

```
tableView(items: persons, selectionMode: "single",
          cellSelectionEnabled: true, editable: true) {
  tableColumn(editable: true, property: "name", text: "Name",
              prefWidth: 150,
              onEditCommit: { event ->
                int row = event.tablePosition.row
                Person item = event.tableView.items.get(row)
                item.name = event.newValue
              })

  tableColumn(editable: true, property: "dob", text: "Birth",
              prefWidth: 150, type: Date,
              converter: { from ->
                return dateFormat.format(from)
              },
              onEditCommit: { event ->
                int row = event.tablePosition.row
                Person item = event.tableView.items.get(row)
                // convert TextField string to a date object.
                Date date = dateFormat.parse(event.newValue)
                item.dob = date
              })
}
```

What you see


Fx
GroovyFX


GroovyFX Table Demo
Name
Age
Gender
Birth
Jim Clarke
29
MALE
2011-10-01
Dean Iverson
30
MALE
2011-10-01
Angelina Jolie
36
FEMALE
2011-10-01

Layout

JavaFX uses static methods to set constraints:

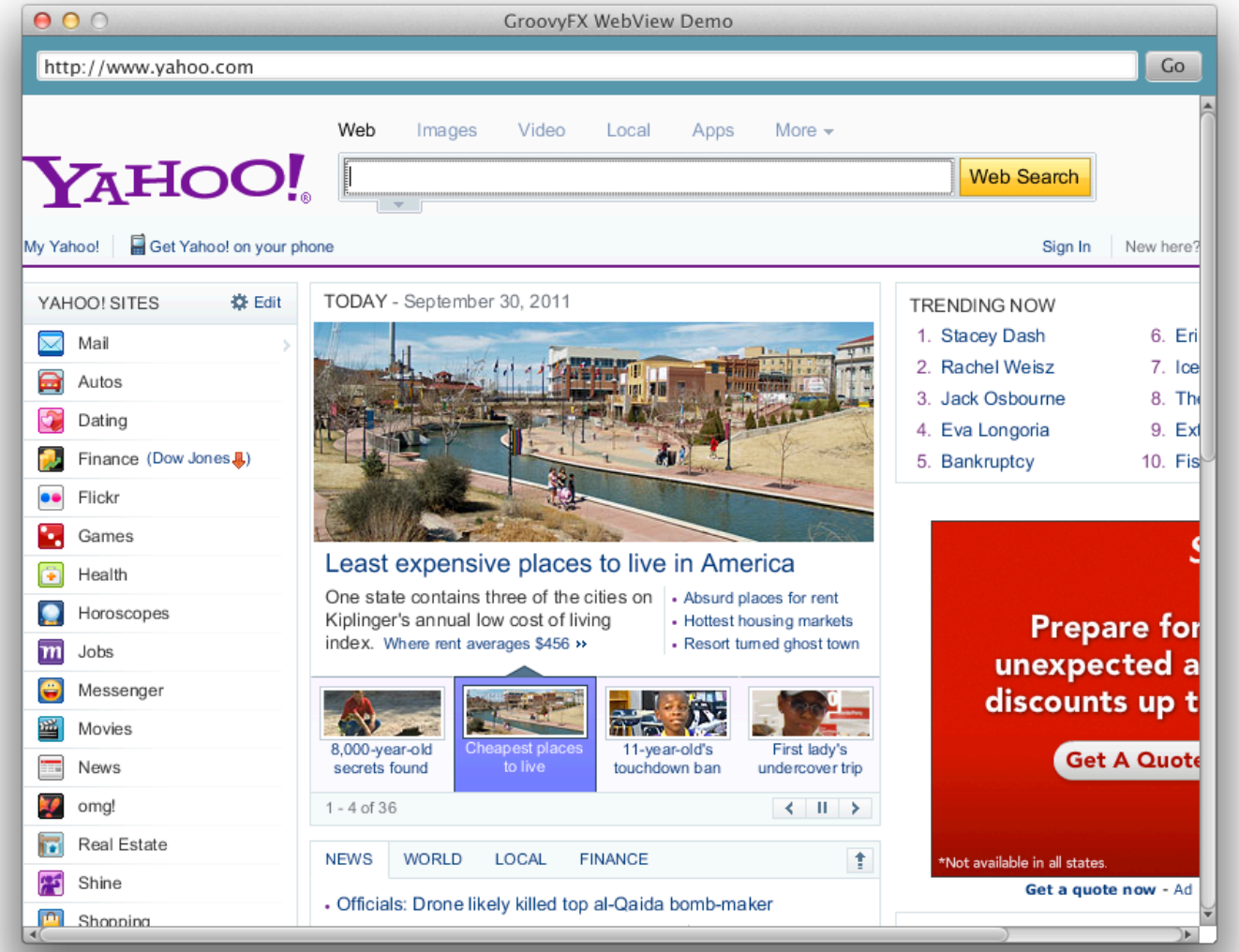
```
TextField urlField = new TextField("http://www.google.com");
HBox.setHgrow(urlField, Priority.ALWAYS);

HBox hbox = new HBox();
hbox.getChildren().add(urlField);

WebView webView = new WebView();
VBox.setVgrow(webView, Priority.ALWAYS);

VBox vbox = new VBox();
vbox.getChildren().addAll(hbox, webView);
```

Layout


```
stage(title: "GroovyFX WebView Demo", visible: true) {
   scene(fill: groovyblue, width: 1024, height: 800) {
      vbox {
         hbox(padding: 10, spacing: 5) {
            urlField = textField(text: homePage, onAction: goAction, hgrow: "always")
            button("Go", onAction: goAction)
         }
         webView(vgrow: "always")
      }
   }
}
```

What you see

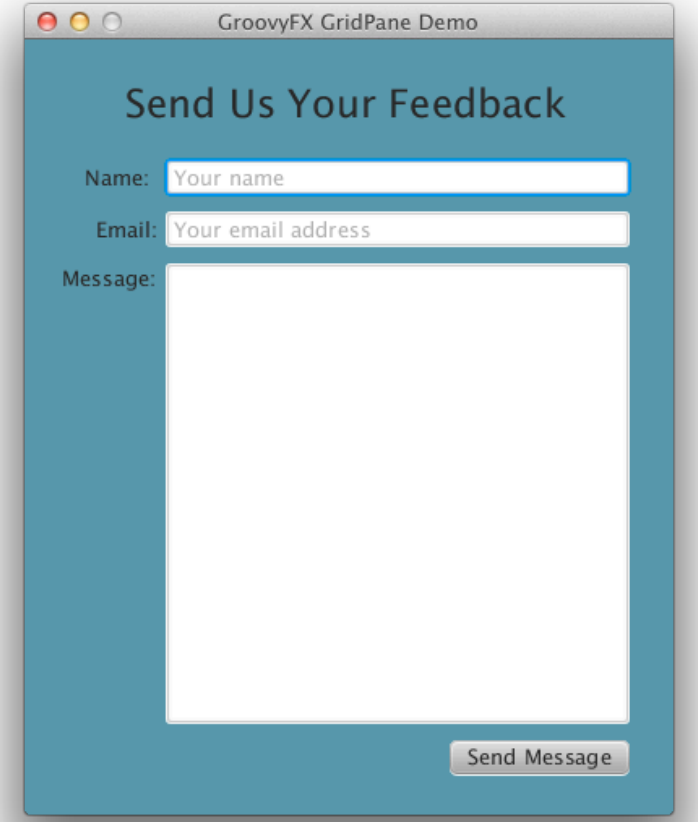
GroovyFX

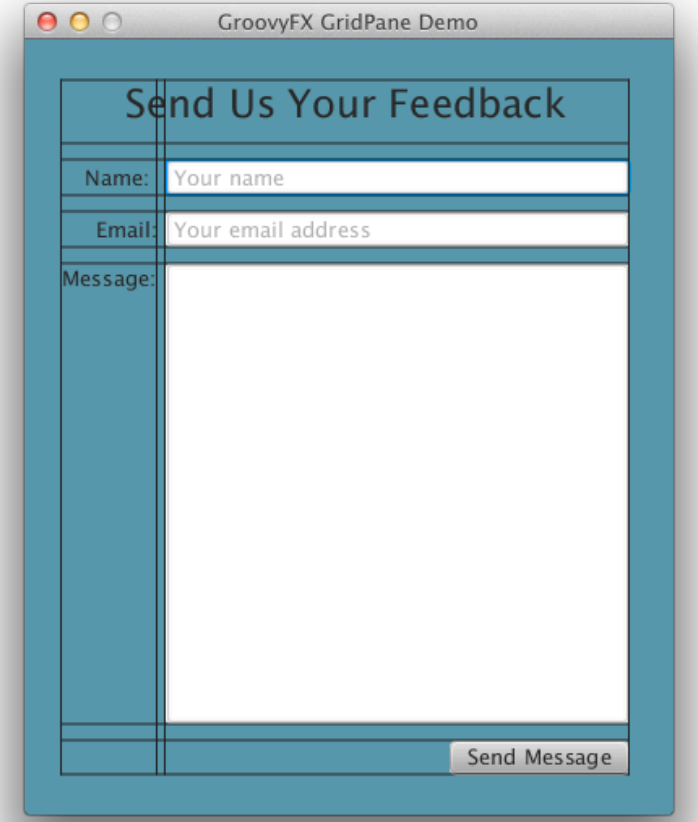


Layout

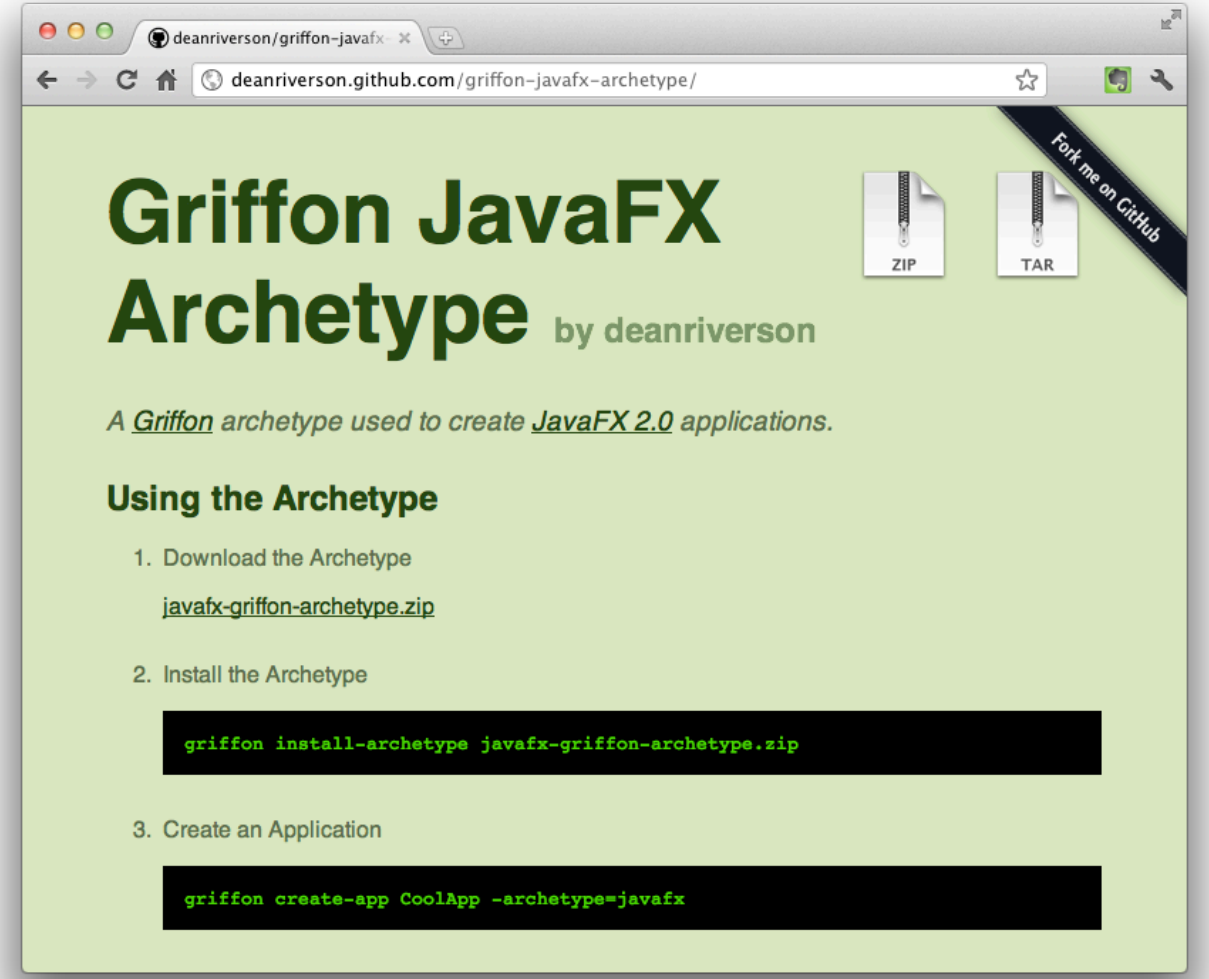

```
gridPane(hgap: 5, vgap: 10, padding: 25) {
  columnConstraints(minWidth: 50, halignment: "right")
  columnConstraints(prefWidth: 250)

  label("Send Us Your Feedback", font: "24pt sanserif",
        row: 0, columnSpan: GridPane.REMAINING, halignment: "center",
        margin: [0, 0, 10])

  label("Name: ", row: 1, column: 0)
  textField(promptText: "Your name", row: 1, column: 1, hgrow: 'always')

  label("Email:", row: 2, column: 0)
  textField(promptText: "Your email", row: 2, column: 1, hgrow: 'always')

  label("Message:", row: 3, column: 0, valignment: "baseline")
  textArea(row: 3, column: 1, hgrow: "always", vgrow: "always")

  button("Send Message", row: 4, column: 1, halignment: "right")
}
```

What you see


GroovyFX GridPane Demo
Send Us Your Feedback
Name:
Your name
Email:
Your email address
Message:
Send Message


GroovyFX GridPane Demo
Send Us Your Feedback
Name:
Your name
Email:
Your email address
Message:
Send Message

GroovyFX + Griffon

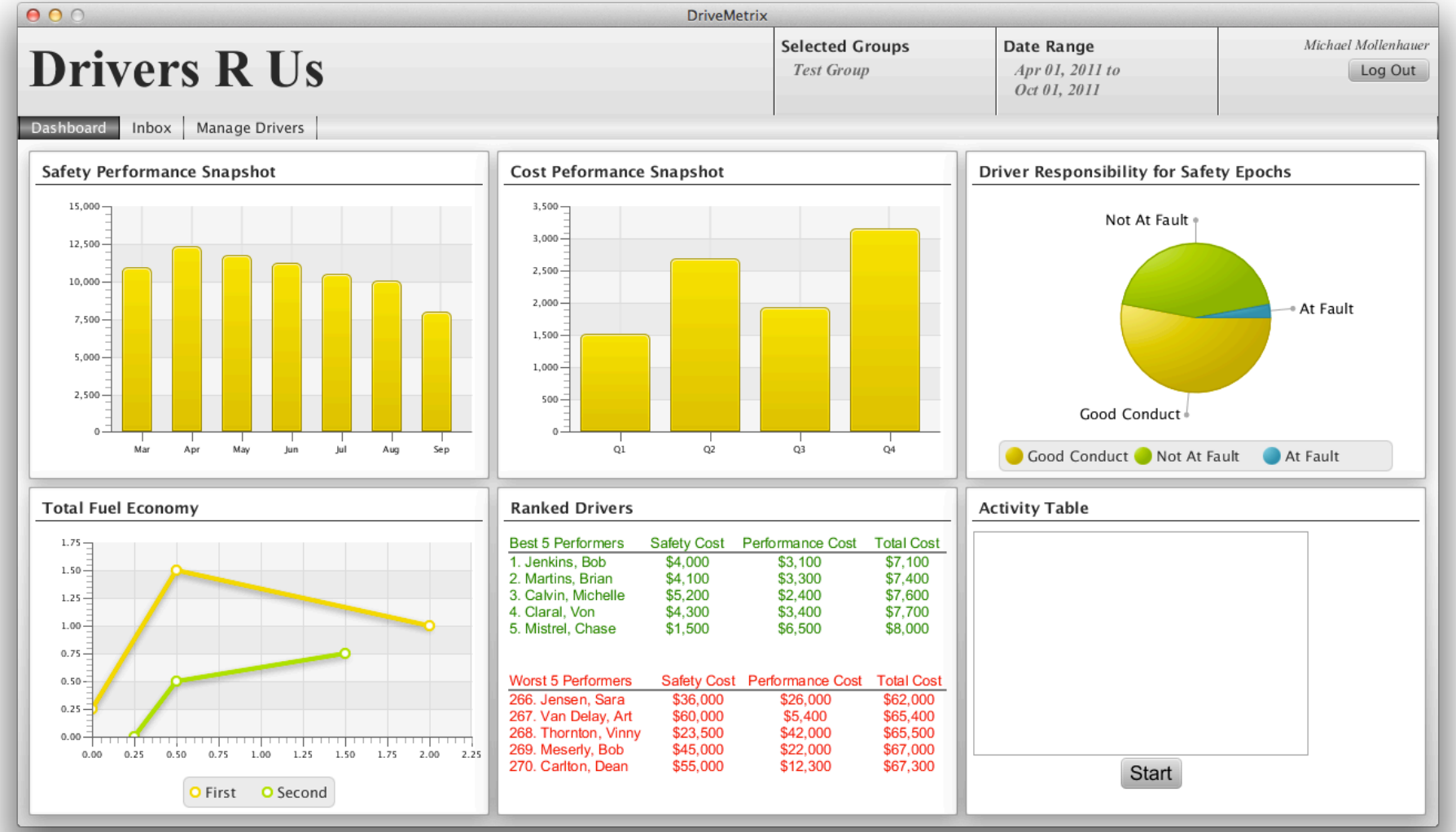


GroovyFX + Griffon

- Download JavaFX archetype from:
  - http://deanriverson.github.com/griffon-javafx-archetype
- griffon install-archetype javafx-griffon-archetype.zip
- griffon create-app CoolApp -archetype=javafx

GroovyFX + Griffon

GroovyFX + Griffon



Resources

- GroovyFX Website:
  - http://groovyfx.org
- Griffon Website:
  - http://griffon.codehaus.org