



JavaOneSM

JFXtras

JavaFX Controls, Layouts, Services, and More

Stephen Chin

GXS

steve@widgetfx.org

tweet: @steveonjava

Dean Iverson

Virginia Tech Transportation Institute

deanriverson@gmail.com

tweet: @deanriverson

Meet the Presenters...

Steve

Dean

Family Man



Motorcyclist

Family Man

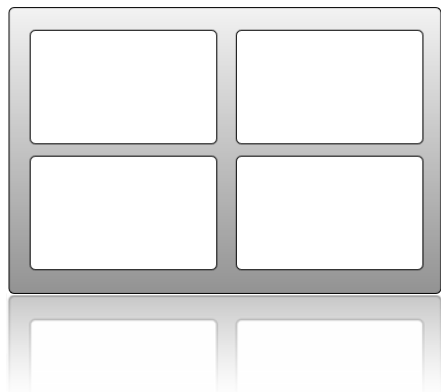


Robotics Coach

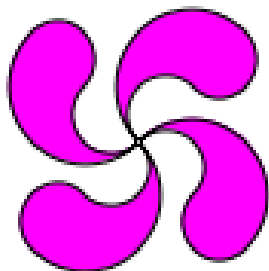
Agenda

- > JFXtras 0.7 Summary
- > JavaFX 2.0 Recap
- > The Future of JFXtras
- > A Little Announcement...

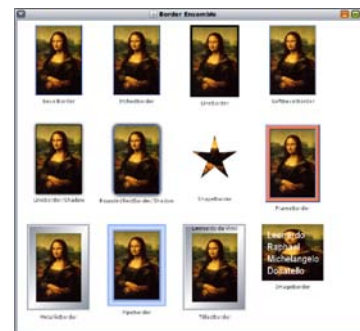
JFXtras 0.7 Summary



Layouts



Shapes

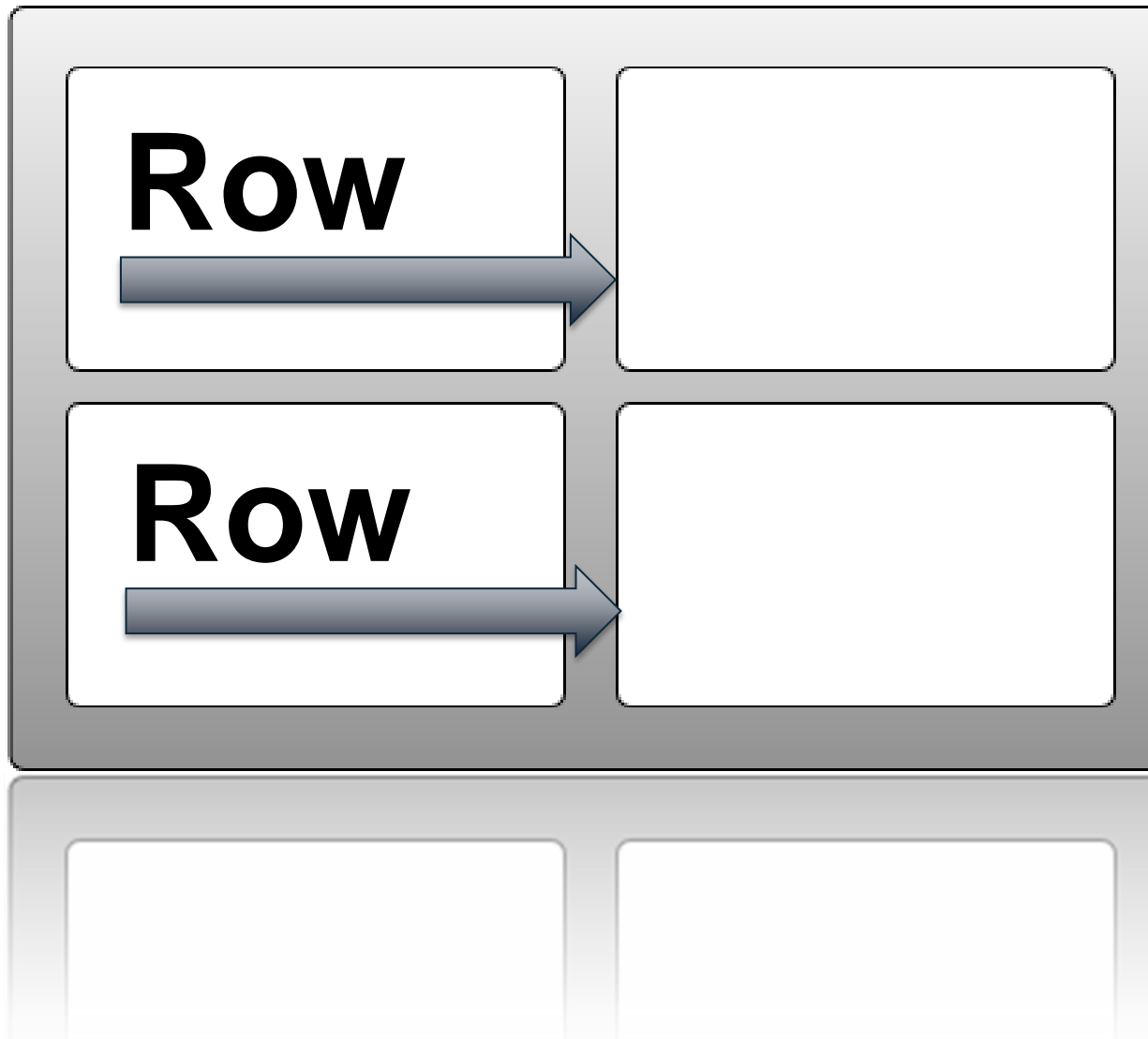


Borders



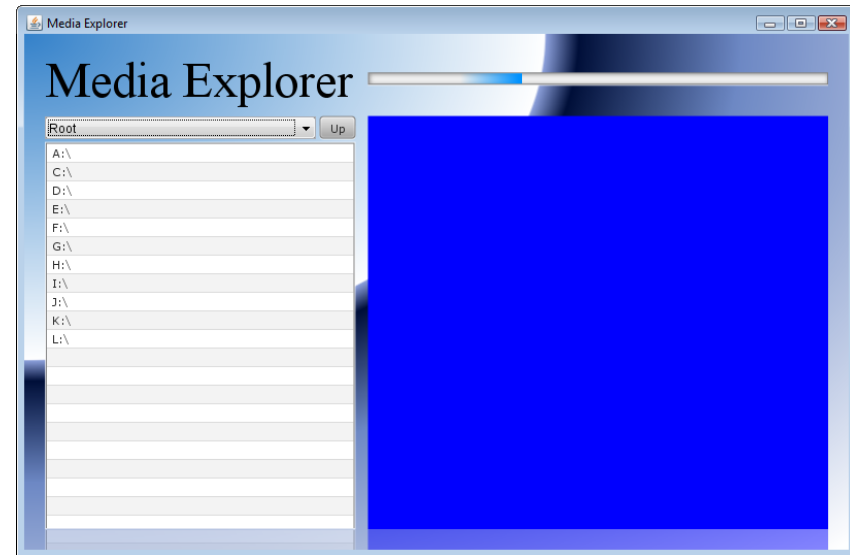
Controls

JFXtras Grid

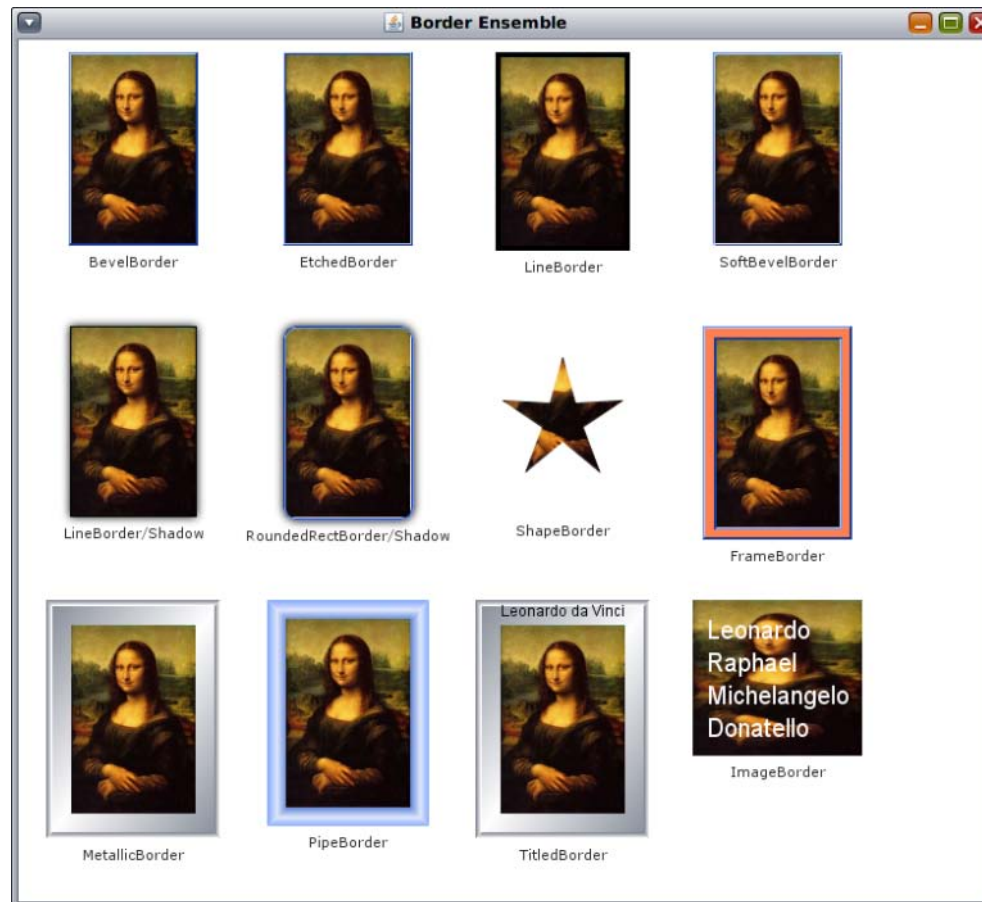


JFXtras Grid

```
Grid {  
    effect: Reflection {}  
    border: 20  
    vgap: 12  
    hgap: 12  
    rows: bind [  
        row([text, progressBar]),  
        row([navigator, mediaGrid])  
    ]  
}
```



JFXtras Borders



JFXtras Borders

Function:

```
override function create() {
```

```
  TitledBorder {
```

```
    id: "imageTitle"
```

```
    title: file.getName()
```

```
    content: FrameBorder {
```

```
      id: "imageFrame"
```

```
      node: ResizableImageView {
```

```
        preserveRatio: true
```

```
        smooth: true
```

```
        image: bind image
```

```
      }
```

```
    }
```

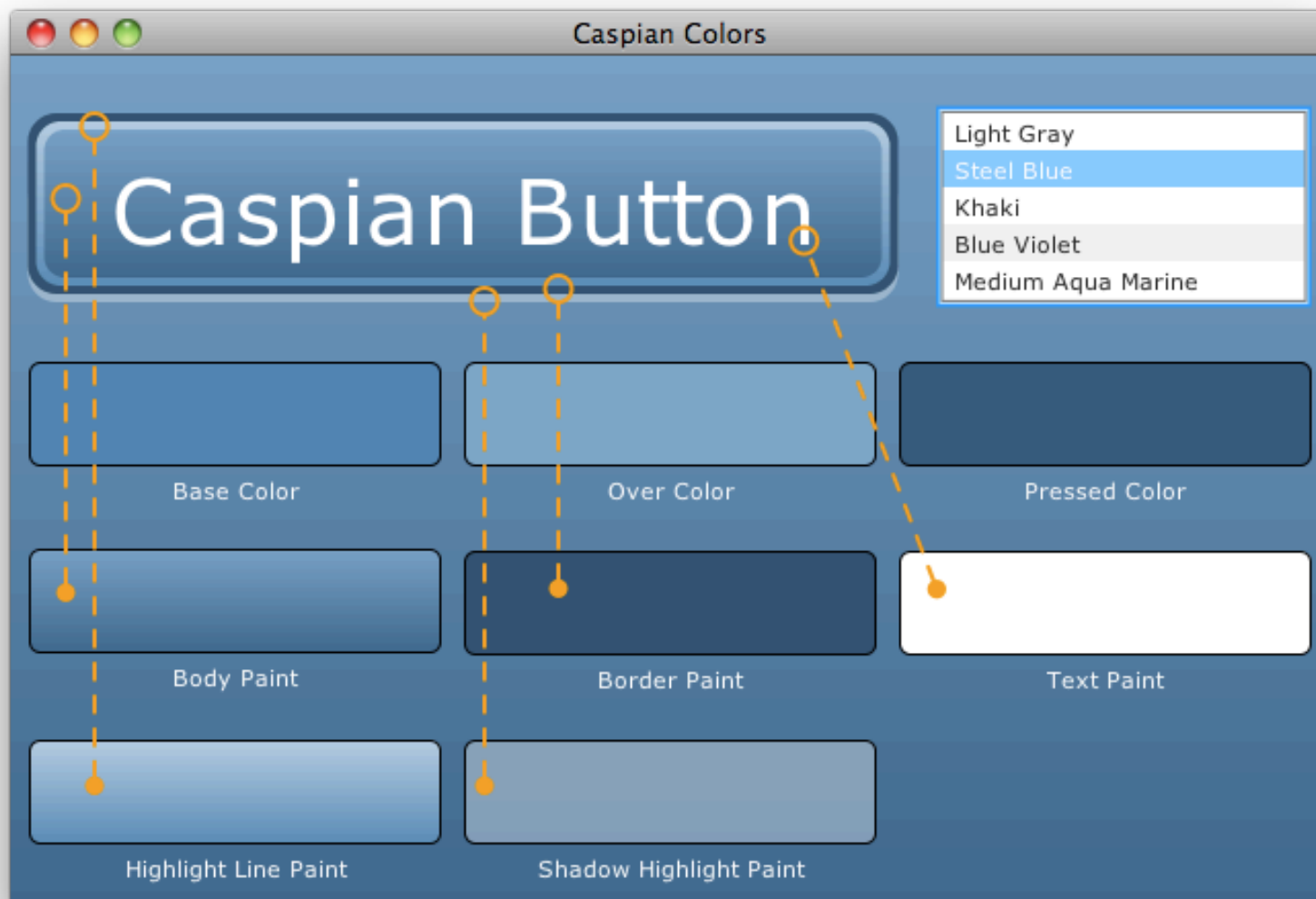
```
  }
```

```
}
```

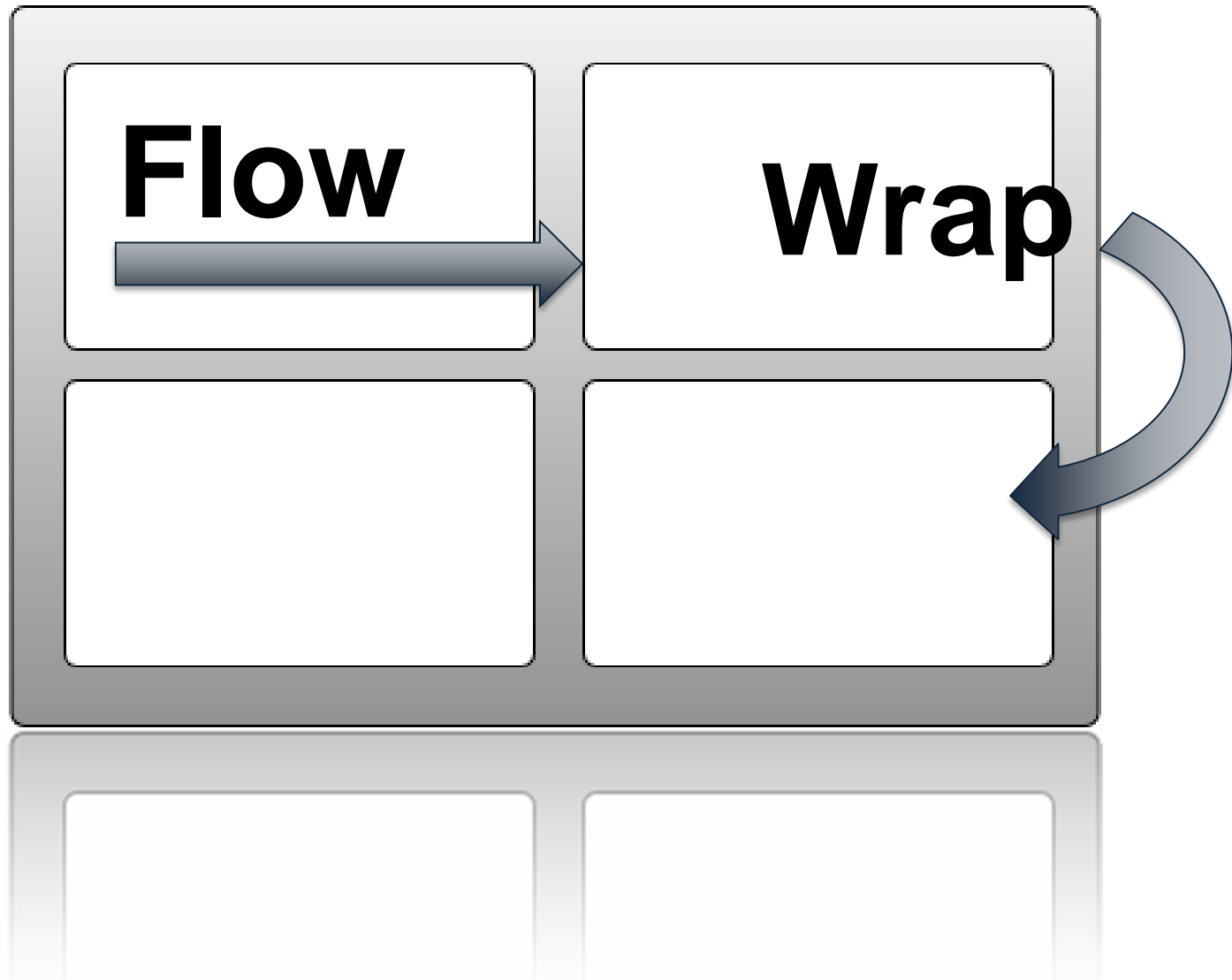
Border and Grid Example



MigLayout for JavaFX



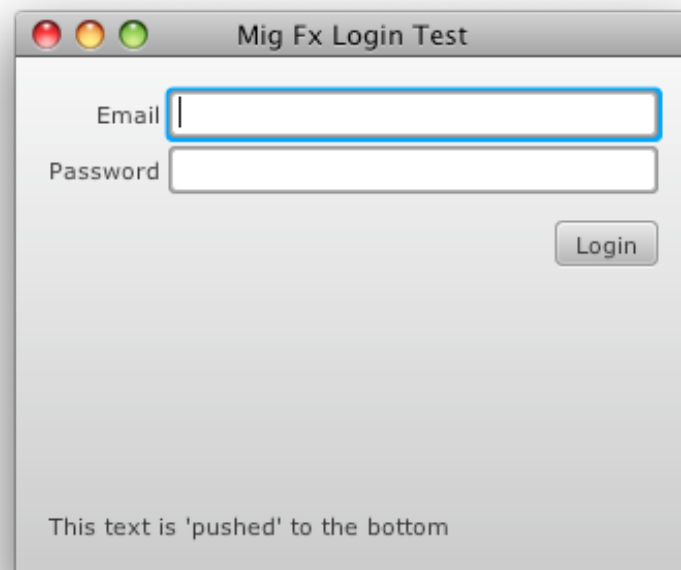
Flexible Grid Layout



```

XMigLayout {
  constraints: "fill, wrap"
  columns: "[]"
  rows: "[][]4mm[]push[]"
  content: [
    Label {
      text: "Email"
      layoutInfo: nodeConstraints( "ax right" )
    }
    TextBox {
      layoutInfo: nodeConstraints( "growx, pushx" )
    }
    Label {
      text: "Password"
      layoutInfo: nodeConstraints( "ax right" )
    }
    TextBox {
      layoutInfo: nodeConstraints( "growx, pushx" )
    }
    Button {
      text: "Login", layoutInfo: nodeConstraints( "skip, right" )
    }
    Label {
      text: "This text is 'pushed' to the bottom", layoutInfo: nodeConstraints( "span" )
    }
  ]
}

```



Flexible Border Layout

```
Stage {  
  title: "Mig Docking Test"  
  scene: XScene {  
    width: 400, height: 400  
    fill: Color.LEMONCHIFFON  
    content: XMigLayout {  
      constraints: "fill"  
      content: [  
        migNode( createLabel( Color.KHAKI, "North" ),    "north" ),  
        migNode( createLabel( Color.GOLDENROD, "South" ), "south" ),  
        migNode( createLabel( Color.GOLD, "East" ),      "east" ),  
        migNode( createLabel( Color.DARKKHAKI, "West" ), "west" ),  
      ]  
    }  
  }  
}
```


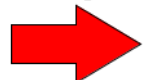








Inflexible Absolute Layout

```
function createLabel( color:Color, label:String ) {  
    XMigLayout {  
        constraints: "fill"  
        content: [  
            ResizableRectangle {  
                fill: color  
                layoutInfo: nodeConstraints( "pos 0 0 container.x2 container.y2" )  
            }  
            Text {  
                content: label  
                font: Font { size: 18 }  
                layoutInfo: nodeConstraints( "center, grow" )  
            }  
        ]  
    }  
}
```



JFXtras Shapes

	Almond	Intersection of two circles (Vesica Piscis)	<code>centerX, centerY, width</code>
	Arrow	Arrow shape	<code>x, y, width, height, depth, rise</code>
	Asterisk	Asterisk with rounded corners	<code>centerX, centerY, width, radius, beams, roundness</code>
	Astroid	Hypocloid with four cusps	<code>centerX, centerY, radius</code>
	Balloon	Rectangular shape with a tab	<code>x, y, width, height, arc, anglePosition, tabWidth, tabHeight, tabLocation, tabDisplacement</code>
	Cross	Symmetrical cross shape	<code>centerX, centerY, width, radius, roundness</code>
	Donut	Regular polygon with a hole	<code>centerX, centerY, innerRadius, outerRadius, sides</code>
	Lauburu	Four comma-shaped heads	<code>centerX, centerY, radius</code>

Continued...

JFXtras Shapes (continued)



MultiRoundRectangle Rectangle with configurable corners `x, y, width, height, topLeftWidth/Height, topRightWidth/Height, bottomLeftWidth/Height, bottomRightWidth/Height`



Rays Multiple rays extend from center `centerX, centerY, rays, radius, extent, rounded`



RegularPolygon Polygon with equal sides `centerX, centerY, sides, radius`



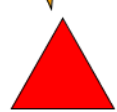
ReuleauxTriangle Curved triangle shape `centerX, centerY, radius`



RoundPin Cone with rounded top `centerX, centerY, height, radius`



Star2 Multipoint star `centerX, centerY, innerRadius, outerRadius, count`



ETriangle Equilateral triangle `x, y, width`

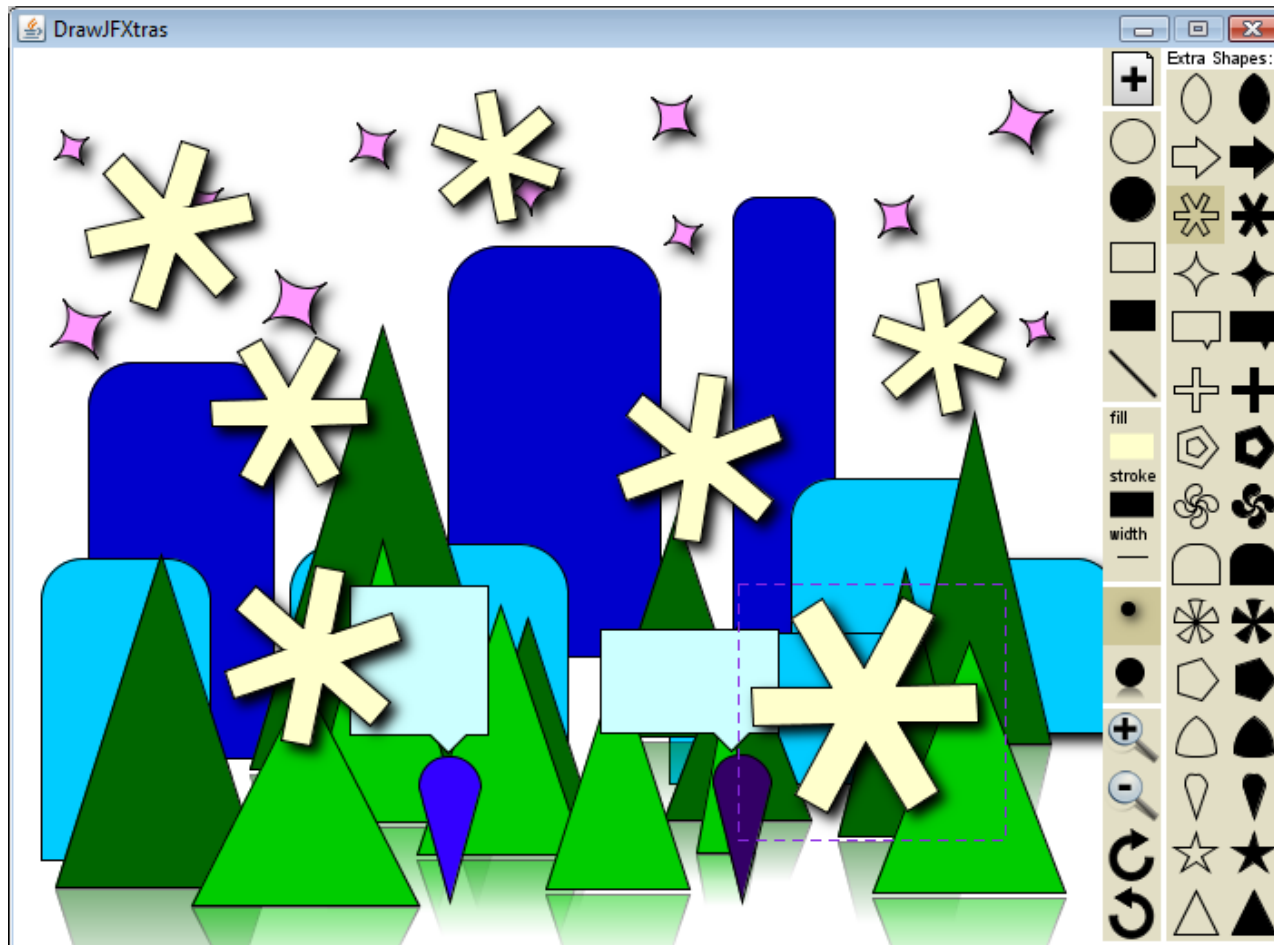


ITriangle Isosceles triangle `x, y, width, height`



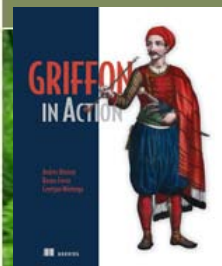
RTriangle Right triangle `x, y, width, height, anglePosition`

JFXtras Shapes



 Launch

Sphere Challenge



Andres Almiray's Weblog

http://www.jroller.com/aalmiray/entry/griffon_gfxbuilder_fxbuilder_side_by

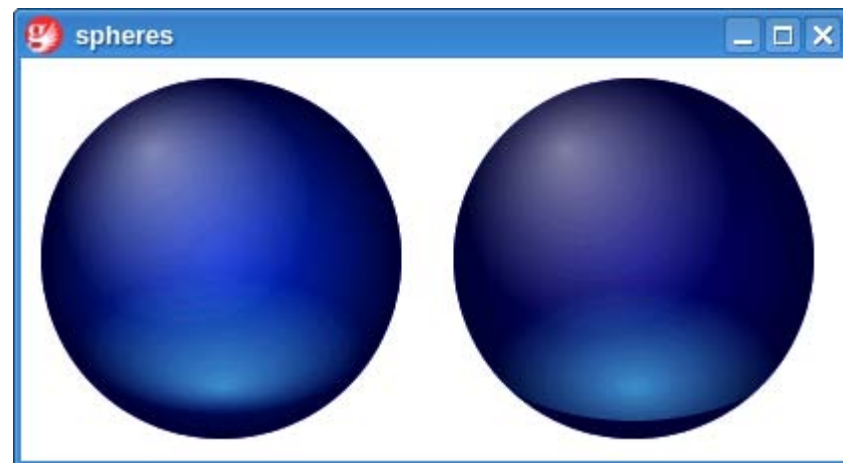
“The following snapshot shows a couple of spheres drawn with GfxBuilder and FxBuilder, can you guess which one is which?”

...

This is by no means a post to bash JavaFX rather to point out some of its deficiencies”

-- Andres Almiray

(taken completely out of context)



Sphere Challenge – JavaFX Response

- > Composition:
 - RadialGradient for the Sphere
 - Three additional RadialGradients for the light sources
 - A blurred shadow underneath
- > Features:
 - All Bound/Relative Coordinates
 - Configurable –
 - Base, Ambient, Specular, Shine Colors
 - Shadow Size and Height
 - Uses New JFXtras ColorUtil Library
 - JavaFX Caching for High Performance



JFXtras Controls

- > Simple Controls
 - XHyperlink
 - XEtchedButton
 - XPane
- > Complex Controls
 - XPicker
 - XCalendarPicker
 - XShelfView
 - XTableView

Hyperlinks and Etched Buttons

```
XHyperlink {
    text: "Oracle's Homepage",
    url: "http://oracle.com/"
}

var hposSeq = [ HPos.LEFT, HPos.CENTER, HPos.RIGHT ];
HBox {
    spacing: -1
    content: for (i in [0..2]) {
        XEtchedButton {
            graphic: ImageView { image: images[i] }
            buttonGroupHPos: hposSeq[i]
        }
    }
}
```

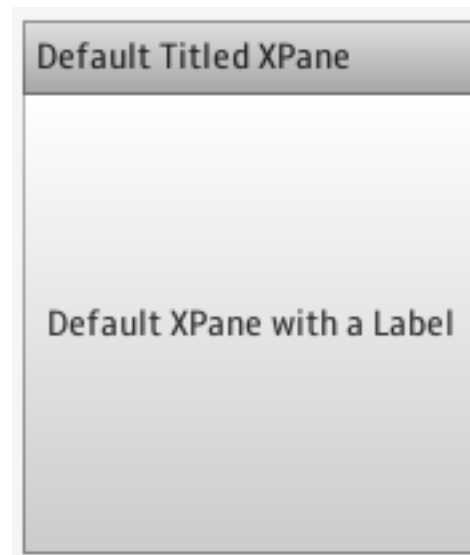


XPane



XPane

```
XPane {  
  title: "Default Titled XPane"  
  contentNode: Label { text: "Default XPane with a Label" }  
  layoutInfo: LayoutInfo { width: 200, height: 200 }  
}
```



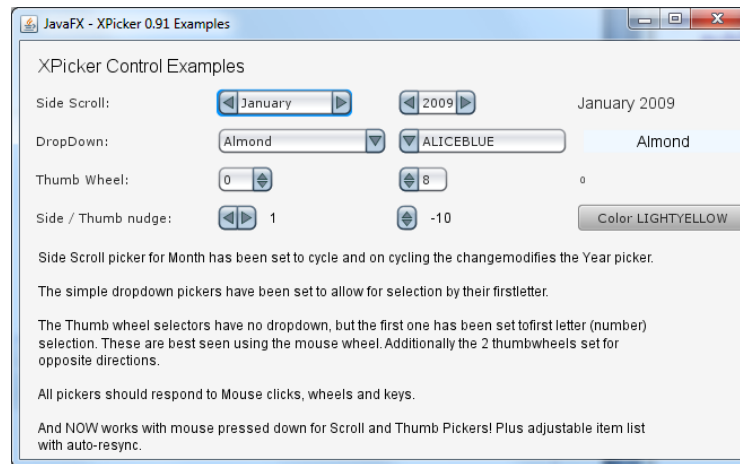
XPicker

> Multiple Picker Types

- ❑ Side Scroll
- ❑ Drop Down
- ❑ Thumb Wheel
- ❑ Side/Thumb Nudge

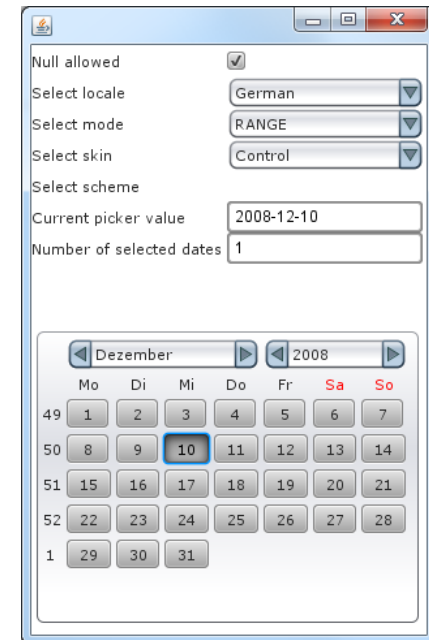
> Supports All Events

- ❑ Mouse Clicks
- ❑ Mouse Wheel
- ❑ Keyboard



XCalendarPicker

- > Configurable Locale
- > Multiple Selection Modes
 - Single
 - Multiple
 - Range
- > Completely Skinnable



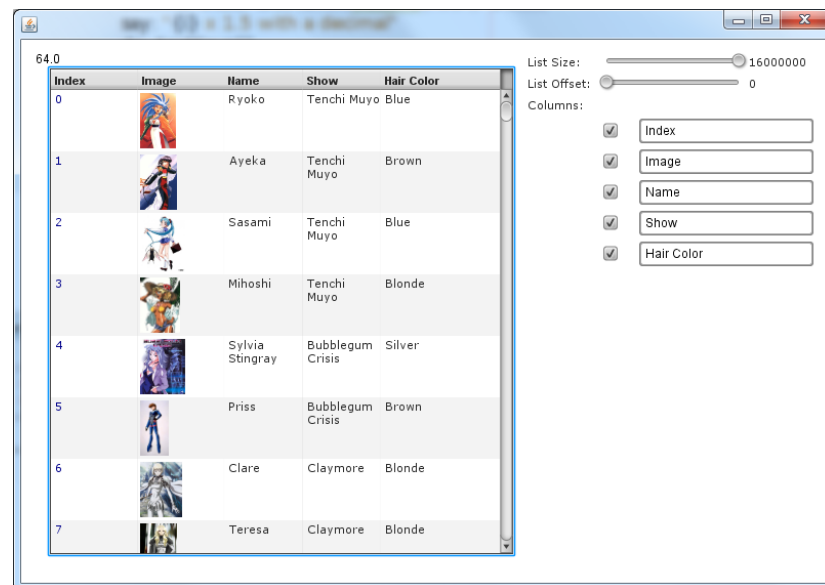
XShelfView

- > High Performance
- > Features:
 - Scrollbar
 - Image Title
 - Reflection Effect
 - Aspect Ratio
 - Infinite Repeat
- > Integrates With JFXtras Data Providers
- > Automatically Updates on Model Changes



XTableView

- > Insanely Scalable
 - Up to 16 million rows
- > Extreme Performance
 - Pools rendered nodes
 - Caches images
 - Optimized scene graph
- > Features:
 - Drag-and-Drop Column Reordering
 - Dynamic Updating from Model
 - Automatically Populates Column Headers



BandMatesFX - JSONHandler in 3 Steps

1

POJO

```
public class FreebaseResult {  
    public var code: String;  
    public var result: Result;  
    public var status: String;  
    public var transactionId: String;  
}
```



2

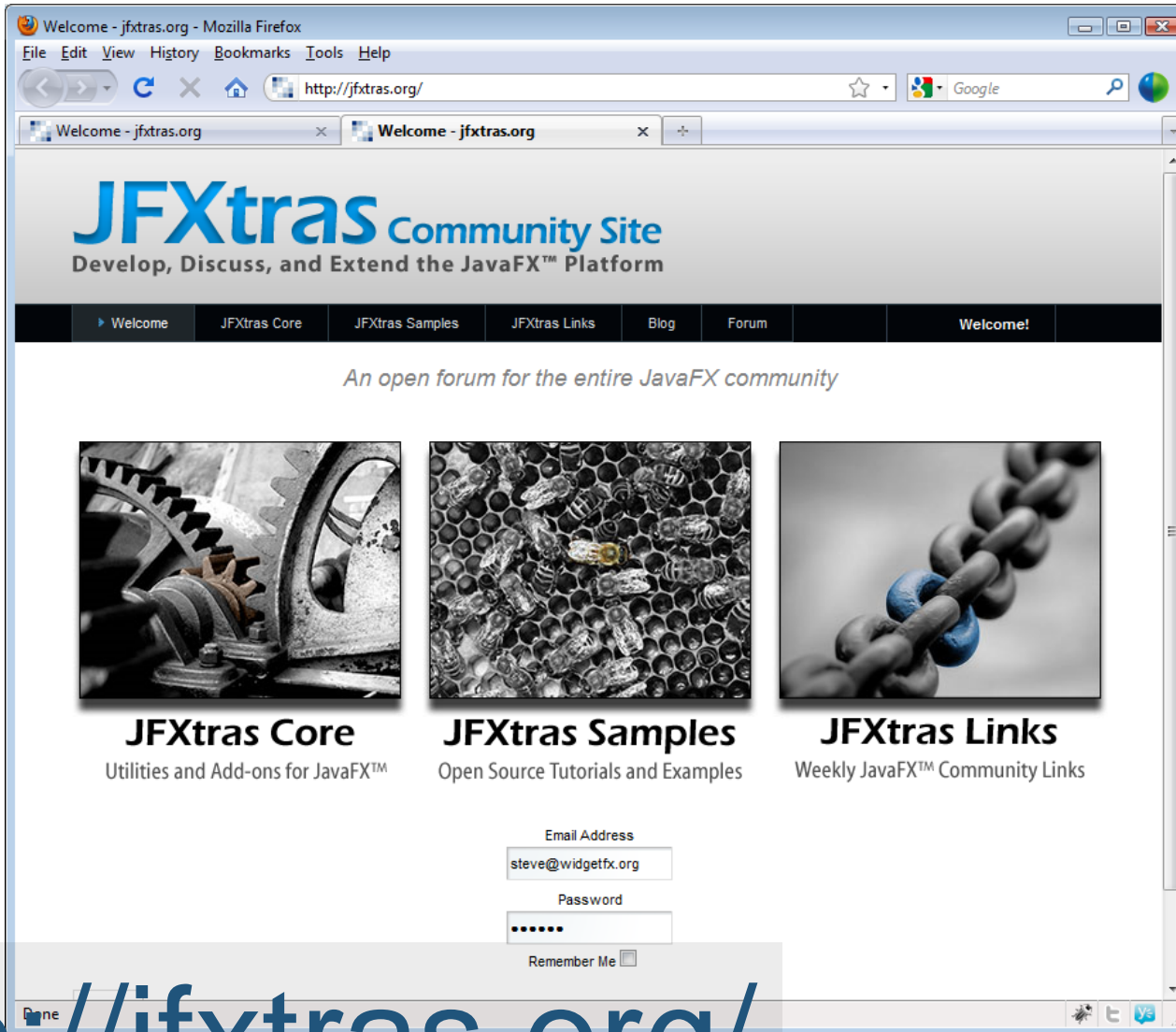
JSONHandler

```
var albumHandler:JSONHandler = JSONHandler {  
    rootClass: "javafxpert.FreebaseResult"  
    onDone: function(obj, isSequence): Void {  
        freebaseResult = obj as FreebaseResult;  
    }  
}}
```

3

HttpRequest

```
req = HttpRequest {  
    location: queryUrl  
    onInput: function(is: java.io.InputStream) {  
        albumHandler.parse(is);  
    }  
}}
```



<http://jfxtras.org/>

JavaFX 2.0 Recap

> The Good

- JavaFX is here to stay
- Controls will be open source
- Calling JavaFX code from different language is pretty cool!

> The Bad

- We have to wait a bit... (next summer)
- Mobile is no longer on the radar

> The Ugly

- JavaFX Script was a pretty nice language...

The Future of JFXtras

- > Mission:
 - Provide Quality JavaFX Add-ons
- > Going Forward:
 - Migrate Non-Overlapping Functionality
 - Support Java APIs For All Components
 - Continue to Fill the Gaps
- > Reality:
 - Migration Will Be Challenging
 - Release Dates Will Trail Oracle
 - (until they are fully open source)

How about JavaFX on... Java

```
public class HelloStage implements Runnable {  
  
    public void run() {  
        Stage stage = new Stage();  
        stage.setTitle("Hello Stage");  
        stage.setWidth(600);  
        stage.setHeight(450);  
  
        Scene scene = new Scene();  
        scene.setFill(Color.LIGHTGREEN);  
  
        stage.setScene(scene);  
        stage.setVisible(true);  
    }  
  
    public static void main(String[] args) {  
        FX.start(new HelloStage());  
    }  
}
```

How about JavaFX on... Clojure

```
(defn javafxapp []  
  (doto (Stage. "JavaFX Stage")  
    (.setWidth 600)  
    (.setHeight 450)  
    (.setScene (doto (Scene.)  
      (.setFill Color/LIGHTGREEN)  
      (.setContent (list (doto (Rectangle.)  
        (.setX 25)  
        (.setY 40)  
        (.setWidth 100)  
        (.setHeight 50)  
        (.setFill Color/RED))))))  
    (.setVisible true)))  
(javafxapp)
```

How about JavaFX on... Groovy

```
FxBuilder.build {
    stage = stage(
        title: "Hello Rectangle (Groovy FxBuilder 2)",
        width: 600,
        height: 450,
        scene: scene(fill: Color.LIGHTSKYBLUE) {
            rectangle(
                x: 25, y: 40,
                width: 100, height: 50,
                fill: Color.RED
            )
        }
    )
    stage.visible = true;
}
```

How about JavaFX on... Scala

```
object HelloJavaFX extends JavaFXApplication {  
  def stage = new Stage {  
    title = "Hello Stage"  
    width = 600  
    height = 450  
    scene = new Scene {  
      fill = Color.LIGHTGREEN  
      content = List(new Rectangle {  
        x = 25  
        y = 40  
        width = 100  
        height = 50  
        fill = Color.RED  
      })  
    }  
  }  
}
```

How about JavaFX on... Visage

```
Stage {  
  title: "Hello Stage"  
  width: 600  
  height: 450  
  scene: Scene {  
    fill: Color.LIGHTGREEN  
    content: Rectangle {  
      x: 25  
      y: 40  
      width: 100  
      height: 50  
      fill: Color.RED  
    }  
  }  
}
```

Announcing Project Visage

- > ***“Visage is a domain specific language (DSL) designed for the express purpose of writing user interfaces.”***

- > Visage project goals:
 - Compile to JavaFX Java APIs
 - Evolve the Language (Annotations, Maps, etc.)
 - Support Other Toolkits

- > Come join the team!

- > For more info: <http://visage-lang.org/>



JavaOneSM

Thank You

Stephen Chin
steve@widgetfx.org
tweet: @steveonjava

Dean Iverson
deanriverson@gmail.com
tweet: @deanriverson