

Swing looking good: Synth L&F

Arjan Schaaf
luminis®



Even voorstellen...

- Arjan Schaaf
- Senior Software Engineer bij luminis®
- Sinds 1997 bezig met Java en sinds 1999 met Swing user interfaces.
- Verder gespecialiseerd in OSGi en J2ME (mobiele telefoons)



- April 2002 opgericht
- Vanaf het begin betrokken bij de NL-JUG
- Kantoor in Arnhem
- 20 medewerkers
- Iedereen is van harte welkom op onze stand hier op de J-Fall





Synth?

- Weinig aandacht voor Synth
- Recente Swing boeken 'benoemen' Synth
- Weinig aandacht voor Synth vanuit SUN



Java Swing

- Lightweight alternatief voor heavy weight AWT
- JavaOne 1997
- Swing 1.0 februari 1998
- Swing 1.1 -> JDK 1.2
- Veel ontwikkelingen sinds Java 5.0 / Java 6.0



Swing Look & Feels

- javax.swing.plaf
- Basic Look & Feel (L&F)
- Platform L&F's: Windows, OSX, GTK
- Generic L&F's: Metal





3th party L&F

- Alloy L&F
- JGoodies Looks
- Overview at <http://www.javatoo.com>



Synth L&F

- Ontwikkelen Swing L&F is moeilijk
 - Starting from scratch (javax.swing.plaf)
 - Extending bestaande L&F (Basic L&F)
- SUN had een flexibele L&F voor GTK L&F nodig
- Swing L&F ontwikkelen zonder te programmeren



Synth L&F

- Sinds Java 5
- Daarvoor al artikelen over painting components with filler objects.
- XML gebaseerd
- Nieuwe UIDelegates op basis van Basic L&F
- Swing architect Scott Violet
- JavaOne demo's Romain Guy



Synth L&F

- GTK L&F
- Synthetica (www.javasoft.de)
- SUN example L&F was aangekondigd





Synth L&F toepassen

```
SynthLookAndFeel laf = new SynthLookAndFeel();  
laf.load(MyClass.class.getResource("synth.xml"));  
UIManager.setLookAndFeel(laf);
```



Synth XML

- JDK JavaDoc voor de DTD en uitleg
- Synth element
- Style element
- Andere elementen zoals:
 - State
 - Color
 - Font
 - Insets



Synth Style

- Beschrijft de 'style' van een component
- Bound per 'region' or by name
- Styles zijn te klonen als extensie mogelijkheid



Binding a style

```
<style id="button">
    <color value="RED" type="BACKGROUND" />
    <font name="Dialog" size="16" />
</style>
```

```
<bind style="button" type="region" key="button" />
```

OR

```
<bind style="button" type="name"
    key="MySpecialButton" />
```

```
JButton myButton = new JButton();
myButton.setName("MySpecialButton");
```





Clone style

```
< style id= "clonedButton" clone= "button">  
  < font name= "Dialog" size= "14"/>  
< /style>
```



Component State

- Koppel gedrag aan de component state
 - DEFAULT (impliciet)
 - ENABLED
 - MOUSE_OVER
 - PRESSED
 - DISABLED
 - FOCUSED
 - SELECTED

```
<state value="SELECTED and PRESSED" >  
  <color value="RED" type="BACKGROUND" />  
</state>
```



Default properties

- Synth XML element “defaultsProperty”
- Opgeslagen in UIDefaults table
- Bruikbaar via javax.swing.UIManager

```
<style id="example">  
  <defaultsProperty key="Tree.rowHeight"  
                    type="integer"  
                    value="20" />  
</style>
```



Synth properties

- Gebruik gelijk aan UIDefaults properties
- Gebruikt door de Synth UIDelegates
- Zie uitleg in de Synth JavaDoc

```
<style id="anotherExample">
  <object class="javax.swing.plaf.ColorUIResource"
    id="AnotherColor">
    <int>242</int>
    <int>243</int>
    <int>244</int>
  </object>
  <property key="Table.gridColor"
    type="idref"
    value="AnotherColor"/>
</style>
```

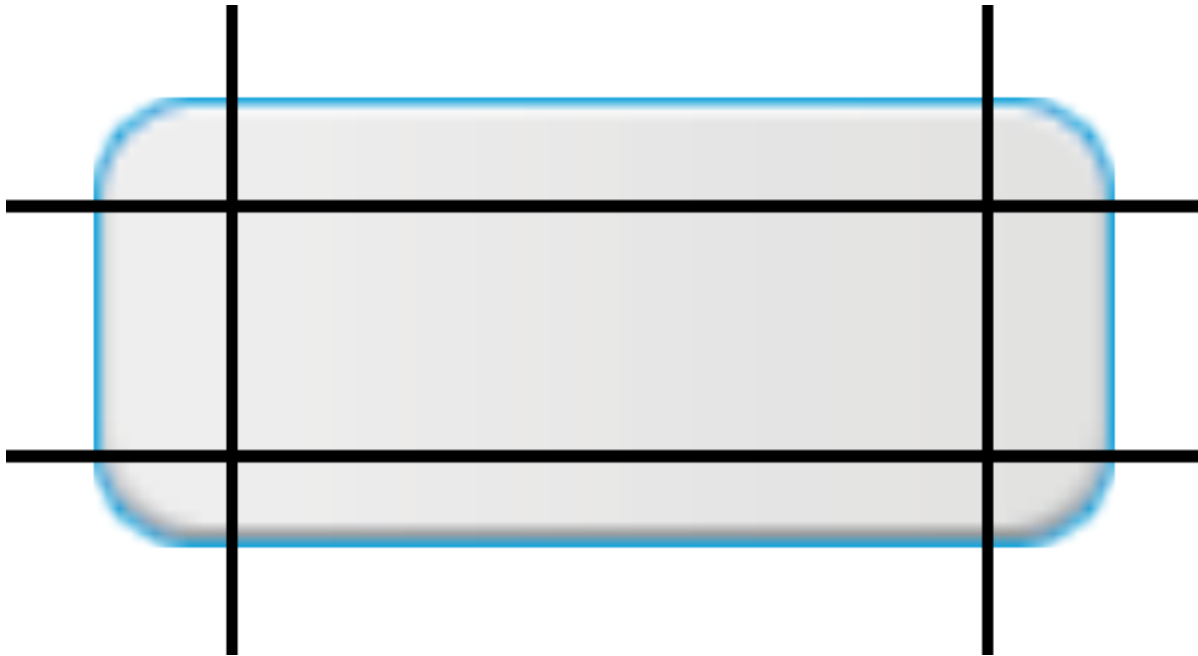


Meer dan alleen properties

- Vormgevers verwachten meer mogelijkheden:
 - gradients
 - shadows
 - andere vormen dan alleen rechthoeken
- Hoe gaat Synth hier mee om?







Synth Painters

- `paintBorder`
- `paintBackground`
- Combineer deze 2 methoden met verschillende plaatjes of gebruik alleen `paintBorder` met de `paintCenter=true` optie



Customizing Synth

- Custom painters
- Composite components
- Ouderwets programmeren: nieuwe UIDelegate's



Custom Painters

- Gradient painter
- SVG painter



Tooling

- Weinig ondersteunende tooling
- SynthArtist: Grafische editor voor de Synth XML
 - Preview mode
 - Niet alle Swing componenten
 - Maar het is een start!



Demo



Resultaat

```
< style id= "button">
  < state>
    < imagePainter path= "resources/button.png"
      sourceInsets= "5 5 5 5"
      paintCenter= "true" />
  < /state>
  < state value= "MOUSE_OVER">
    < imagePainter path= "resources/highlight_button.png"
      sourceInsets= "5 5 5 5"
      paintCenter= "true" />
  < /state>
< /style>
```



Wat kan er nu met Synth?

Filter

Version from/to ▼

from to

Details +

Name	Version	Vendor
bundle-1158929887957	1.0	Luminis B.V
bundle-1158929889002	1.0	Luminis B.V
bundle-1158929889366	1.0	Luminis B.V
bundle-1158929889492	1.0	Luminis B.V
bundle-1158929889834	1.0	Luminis B.V
bundle-1158929890812	1.0	Luminis B.V
bundle-1158929891192	1.0	Luminis B.V
bundle-1158929891344	1.0	Luminis B.V
bundle-1158929892134	1.0	Luminis B.V
bundle-1158929893506	1.0	Luminis B.V
bundle-1158929893671	1.0	Luminis B.V

15 2 1 3 ★ 2 ↻ 2 ✕ 1



Synth bugs

- Veel kleine bugs
- Toch een stabiele implementatie
- Maar veel onhandigheden als je iets net even anders wilt doen
- Package private classes
- Overerven gedrag van Basic L&F



Synth Tricks & Tips

- Pak de sourcecode erbij
- Doet een painter niet wat je wilt: Gebruik een eenvoudig plaatje om goed te zien wat er gebeurt
- Forums JavaDesktop.org



Synth voordelen

- Alternatieven voor het ontwikkelen van een eigen L&F kosten meer
- Flexibel
- Goede samenwerking met vormgevers



Synth nadelen

- Te weinig documentatie
- Geen goede voorbeelden
- Weinig mogelijkheden voor workarounds voor bugs





Links

- www.javadesktop.org
- www.l2fprod.com
- www.javootoo.com
- www.jgoodies.com
- <http://code.google.com/p/synthartist/>



Tot ziens!

