Groovy

David Pick, Mark Jenne, Patrick Nowicki

A little about Groovy

- Dynamically compiled to Java Virtual Machine byte code
- Object-Oriented
- Closures
- Operator Overloading
- Lists!

http://i136.photobucket.com/albums/q172/IMANDEA/GroovyBaby.jpg

Getting Started With Groovy

- Generally a .java file can be renamed to .groovy
- Groovy has lists and maps, no more importing ArrayList :)
- Closures, Java should have them, but it doesn't, so now Groovy does

http://i128.photobucket.com/albums/p177/nandysewton/groovy.gif

Differences from Java

- == means equals on all types
- alternate for-loop syntax
  for (i in 0..len-1)
  for (i in 0..<len)
- semicolons are optional
- Nested classes not currently supported
- throws clause not checked by
- dynamic and static typing is supported

Some Groovy Features

- multiple assignment and one-line variable swapping
- named parameter passing
- passing closures into methods

```
def closure = { param -> param + 1 }
def answer = [1, 2].collect(closure)
assert answer == [2, 3]
```

- safe navigation

```
def streetName = user?.address?.street
```

GUI Programming

- GUI design can directly use Java Swing elements or the Groovy SwingBuilder
- SwingBuilder simplifies the syntax of Swing elements
  - Builders - handle the busywork of Swing for you
  - Code is much more readable and maintainable
- "actionPerformed" attributes can be set on elements
  - these act as closures
- Reference elements by ID, like JavaScript

SwingBuilder Example

- Elements can be given an id instead of complete definitions
- actionPerformed attribute acts like a closure

```
import groovy.swing.SwingBuilder
import java.awt.BorderLayout as BL

def swing = new SwingBuilder()
count = 0

def frame = swing.frame(title:'Frame', size:[300,300])
{
   borderLayout()
   label(id:'clickCount', text:"Click the button!", constraints: BL.NORTH)
   button(text:'Click Me',
          actionPerformed: {count++;
                            clickCount.text = "Clicked ${count} time
(s).";},
          constraints:BL.SOUTH)
}

frame.show()
```