X-Platform Development in Scala.js

Li Haoyi
9 August 2014
Scala by the Bay
What is Scala.js?

• Scala to JavaScript, run in browser

• Share code client/server!

• Get typechecking in your web apps!

• 1-4x slower than JS, 10x slower than Scala-JVM, 2-6x faster than Python
What’s wrong with (my) JS?

• “Who is doing this?”
• “Where did this variable come from?”
• “Why is it undefined?”
• “Why is renaming this method so hard =(”
• “I want to refactor this but I’m scared!”

• “WTF is going on -.-”
Live coding

Client-side Application
## Can/Can't Use

### Can use
- Most of java.lang.*
- Almost all of scala.*
- Some of java.util.*
- Scala Macros: upickle, scala-async, scalaxy, etc
- Pure-Scala ecosystem: shapeless, scalaz, scalatags, utest

### Can't use
- j.l.Thread, j.l.Runtime, ...
- s.c.parallel, s.tools.nsc
- org.omg.CORBA, sun.misc.*
- Reflection: scala-pickling, scala-reflect
- Java-dependent: Scalatest, Scalate
Can/Can't use

Can use
• JS stuff: XmlHttpRequest, Websockets, Localstorage
• HTML DOM, Canvas, WebGL
• JavaScript libraries: chipmunk.js, hand.js, react.js, jquery
• IntelliJ, Eclipse, SBT
• Chrome console, firebug

Can't use
• JVM stuff: Netty, akka, spray, file IO, JNI
• AWT, Swing, SWT, OpenGL
• Java ecosystem: guice, junit, apache-commons, log4j
• Yourkit, VisualVM, JProfiler
Tutorial
On the right are some of the common types of materials and joints you will encounter in the game.

Roll your ball left and right using the arrow keys, and draw paths using the mouse or touchscreen for your ball to roll on. Esc restarts the current level, and pg up and pg down allow you to zoom in or out.

When ready, roll your ball to the bottom right corner of the level and hit the light blue “Goal” block to proceed to the next level.

Gusts
These pick up your ball, and any other objects that end up within them, and push them around.

Player

Goal

Joints and Shapes
The properties of each joint (left) or shape (right) indicated by color. The bright red color indicates the value of the property:

- Frictiony
- Springy
- Spinnny

Rough
Dense

Lasers
These destroy your ball if you touch them, forcing you to restart from your last checkpoint. They can be blocked by drawn paths or other shapes.

Show & Tell
TodoMVC, Roll, Scala-Js-Fiddle, Ray-Tracer
Why Scala.js

- Scala's great and JavaScript not so much
- Huge ecosystem of libraries and tools available for free (because Scala, and JS!)
- Web apps > Swing apps for deployment
- Front-end development in Scala is fun!
Live Coding

Server-Client Application
Scala.js

- Able to use strengths of each platform

- Sharing code/libraries/data-structures between client as server is awesome

- Static typing keeps things straight and keeps you sane
Cool Demos

• Shared libraries between client & server
• Auto-rename routes and Ajax calls!
• Find-usages for Ajax endpoints!
• Tons of Safety
• Tons of Toolability (and Tools!)
Conclusion

● X-Platform dev in Scala.js is awesome

● [www.scala-js.org](http://www.scala-js.org)
  ○ Fork it, make cool stuff
  ○ Come hang out in the google group

● [https://github.com/lihaoyi/workbench-example-app](https://github.com/lihaoyi/workbench-example-app)
  ○ master -> Client example
  ○ todomvc
  ○ raytracer
  ○ autowire -> Server-Client example
Questions?

```scala
import math._
import scala.scalajs.concurrent.JSEXecutionContext.Implicits.queue
import ScalaJEXample.{Color, Epsilon}
import scala.async.Async_
import scala.concurrent.Future
import scala.io.loops._
import scala.language.postfixOps

/**
 * A simple ray tracer, taken from the PyPy benchmarks
 * 
 * [https://gitbucket.org/pypy/benchmarks/src/846fa56a282b/own/raytrace-simple.py?at=defaul
 * 
 * Half the lines of code
 */
object ScalaJEXample extends js.JSApp{

  import Page_
  val Epsilon = 0.00001

  type Color = Vec
  val Color = Vec

  def main() = {
    val r = new util.Random(16314302)
    val spiral = for (i <- 0 until 10) yield {
      val theta = i * (1 + 5) * PI / 100 + 0.3
      val center = (0 - 4 * sin(theta), 1.5 - i / 2.0, 0 - 4 * cos(theta))
      val form = Sphere(center, 0.3 + i * 0.1)
      val surface = Flat((1 / 6.0, 1 - i / 6.0, 0.5))
      (form, surface)
    }

    def rand(d: Double) = (r.nextDouble() - 0.5) * d * 2

    val drops = Array(
      Sphere((2.5, 2.5, -8), 0.3),
      Sphere((1.5, 2.2, -7), 0.25),
      Sphere((-1.3, 0.8, -8.5), 0.15),
      Sphere((0.5, -2.5, -7.5), 0.2),
      Sphere((-1.8, 2.5, -7.5), 0.3),
    )
  }
```