Scales IDE: A web-based development environment for teaching functional reactive programming using media computation.

Benjamin J. Kruger (Northeastern State University) && *Namchi Do (Trinity University)* && *Brian Howard , Faculty Mentor (DePauw University)*

Background

Introducing a functional language early in a student's college experience provides an excellent setting in which to explore recursively-defined functions and data structures. To that end, the Scales IDE (SIDE) builds on previous student research in teaching functional programming through media computation.

Previous work has revealed two impediments to student learning:

- Unfamiliar or convoluted syntax.
- IDEs requiring complicated installation and configuration.

Objectives

The Scales IDE comprises two integrated projects:

- The Scales Library: a reactive media library for functional Scala targeting the HTML5 Canvas.
- A web-based IDE for developing Scales applications.
  - With file management, source code editor, and output canvas.

Goals for this year's effort included:

- Simplify the syntax.
  - By creating a simple reactive media library for Scala.
- Eliminate large downloads and complex installations.
  - By deploying the IDE as a web app.

IDE

- Toolbar
- Multiple file support
- Full-featured source code editor
- Resizable Panels
- Graphical output via HTML Canvas

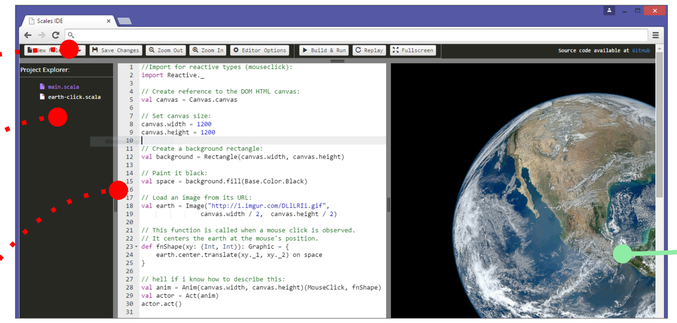


Figure 1. A look at the latest version of the Scales IDE and its features. The image on the right is loaded by URL and placed with a mouse click.

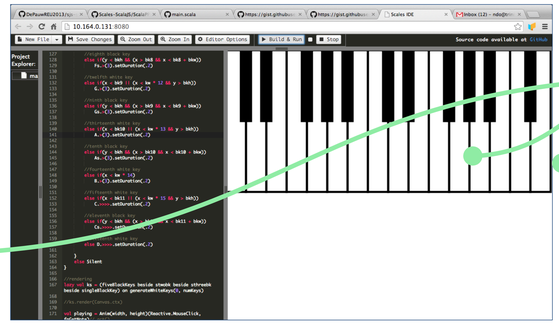


Figure 2. This piano was created using the Scales audio, animation, and graphics libraries. When a key is clicked, the corresponding note plays.

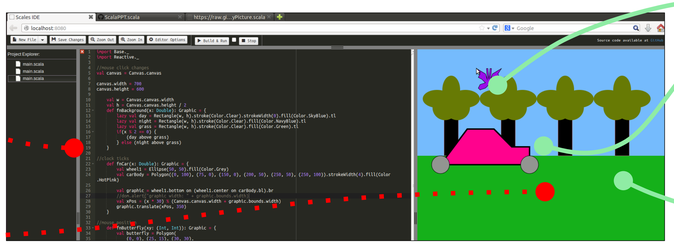
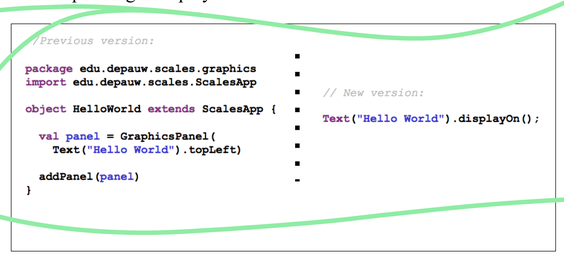


Figure 3. The scene depicted in the canvas above uses the Act library. The trees in the background are static graphics. There are three animations: the butterfly reacts to mouse position, the car moves across the screen in a loop for several seconds, and the sky changes from day to night on clicks. A sequence of notes plays after the car stops.



```
//Previous version:

package edu.depauw.scales.graphics
import edu.depauw.scales.ScalesApp

object HelloWorld extends ScalesApp {

  val panel = GraphicsPanel(
    Text("Hello World").topLeft)

  addPanel(panel)
}

// New version:

Text("Hello World").displayOn();
```

Figure 4. The current Scales library abstracts away elements such as the canvas from the novice user, resulting in simpler code. On the left is an example of rendering the graphical text "Hello World" on a canvas. On the right is the counterpart in this version of the Scales library.

Library

- Graphics API
  - *Composes graphics functionally.*
  - *Supports text art, shapes, customizable polygons, imported images, turtle-like paths, and rendered bitmaps.*
- Audio API
  - *Creates single notes of a given frequency and duration.*
  - *Uses Web Audio.*
- Animation API
  - *Creates animations by listening for certain events.*
  - *Currently supported events include:*
    - *Timer*
    - *Mouse click*
    - *Mouse position*
    - *Key presses*
  - *Builds upon Scala.rx, a functional reactive library for Scala.*
- Act API
  - *A wrapper library that allows users to create animations, music, and graphics on the same canvas.*
  - *Prior to the creation of the Act library, we discovered behavioral bugs that erased still graphics when animations were redrawn. Thus, the Act library was needed.*

Conclusion

- Scales IDE will be used in class at DePauw University in Spring 2015. The convenience of the cloud and rewarding nature of multimedia should deliver an engaging learning experience..
- The graphics API allows students to compose shapes functionally. The audio API allows students to create musical notes and arrange them both melodically (sequentially) and harmonically (in parallel). The animation library produces graphics or audio based on observable events such as clock ticks, mouse events, and keyboard events.
- By delivering the IDE in a web page and employing the Java-like syntax of Scala, we eliminate two major impediments faced by the novice student.

Future Work

Scales IDE would benefit from further refinements and extensions in these areas:

- Extending the Scales Audio API, making use of more of Web Audio's native features, such as support for audio synthesis and sound wave visualization.
- Implementing user interface for some file management actions currently only accessible through the browser's JavaScript console.
- Fixing browser inconsistencies and bugs in the IDE (e.g. full-screen playback that fails in edge cases).
- Support for client-side parsing was added, but no parsers were implemented.
- Extending the server with user authentication, file management, improved error reporting.

Acknowledgements

We would like to thank the following organizations for their funding and support in the successful completion of this project:

This work was supported by
National Science Foundation
Grant Number CNS-1156893

DEPAUW UNIVERSITY
Est. 1837