About this repo

Project Overview: Functional Programming and Data Visualization (DISCUSSION)

Compact summary of project

- goals
- non-goals
- what's here

Data Visualization for Functional Programmers

Functional Programming for Data Visualizers

Next steps: guide to other docs

- TUTORIALS
- HOW-TO
    - Spago example from scratch
- DISCUSSION software enginering and technology considerations
    - roles and responsabilities
    - finally tagless as a method / approach
    - alternatives / extensions / future-work
- REFERENCE library architecture

Finally Tagless Visualization (DISCUSSION)

What it is

Super simple examples from earlier repo

Why it's useful: extensibility and multiple-interpreters

In visualization: graphs, meta-trees and printers

Roles and responsibilities (DISCUSSION)

Visualization as "decoration" vs "core"

Maintainability, closed loops, longer lived code, life-cycles, interaction design

Role 1: data to data-structure

File handling, AJAX, conversion and validation of JSON or CSV, calculating derived data, accessors to provide rich API for data visualisation. Concerned with the semantic representation of the data.

Role 2: data-to-DOM

The part that is closest to the traditional D3.js script: declarative but highly sequential. Concerned with the visual representation of the data.

Role 3: web app development

Tying it all together. Treating the data visualization as a component, even if it is completely interactive. Example framework: Halogen.

Library architecture: three DSLs (REFERENCE)

The Selection and Simulation Monads

The FFI

Callbacks, attributes, delegation thru functions to allow out of monad calls (pros and cons)

Tutorials (TUTORIAL / HOW-TO)

Three Little Circles (TUTORIAL)

- just walk thru the process in a standalone version of the TLC example

General Update Pattern (TUTORIAL)

- motivation for considering enter-exit-update separately
- introduction of transitions (discussion)
- walk thru the code of GUP

Graph: Toy example, Les Mis (TUTORIAL)

- intro to the domain / problem
- the JSON (miserables.json)
- reading the JSON via AJAX and making a graph from it
- writing the "script" part to draw the graph
- adding simple interaction: drag, zoom (interaction contained within the visualization)
- adding more complex interaction: folding and/or near-neighbour highlighting (interaction raises action to the web app: subscriptions, callbacks)

So you want to build an app with strongly integrated visualization (TUTORIAL / HOW-TO)

Graph: Spago example (HOW-TO)

- intro to the domain / problemthe JSON (multiple sources, need to synthesize, generating trees from graphs etc)
- (reading the JSON, same as Les Mis)

- building accessors, hiding the unsafe coercions
- writing the script part to draw the graph
- managing a complex model

Swapping out, re-writing or augmenting D3.js (DISCUSSION)

Using graphviz or other layouts, replacing the Selection and/or simulation FFIs

Removing the unsafe coercions completely, advantages etc. Imposition of opaque type due to FFI (check that there is no way around this?)