

Genealogy of D3 Code

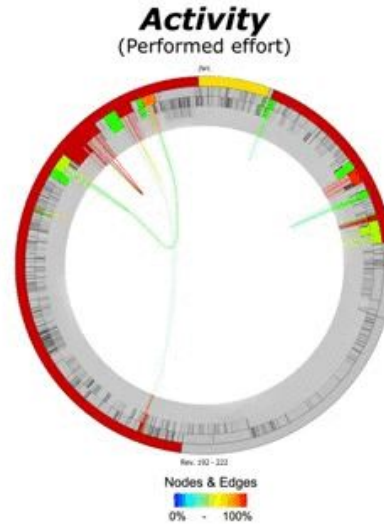
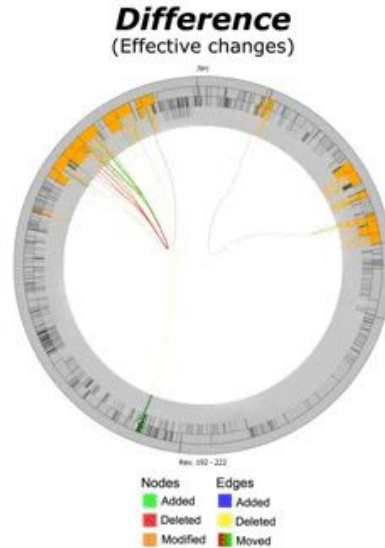
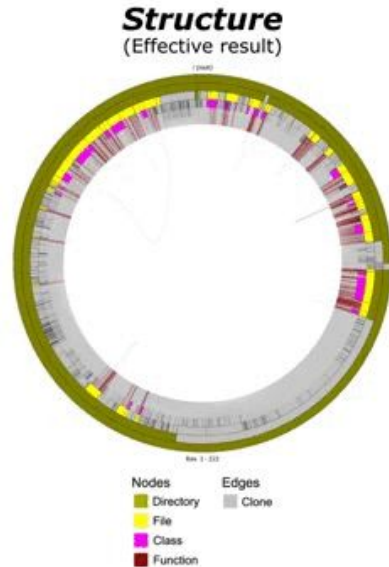
Cathy Zhu, Lucas Throckmorton

Genealogy of D3 Code

- Goal: identify the shared pieces of code amongst pieces of d3 code via MOSS and visualize the evolution of D3 code snippets.
- Scope: about 30k examples of D3 code from blocksplorer.org, starting with about 600 code snippets from the d3.pie.layout API call.
- Currently working with
 - 14 links from d3.force.layout (64 code snippets)
 - 135 links from most recent 3644 d3 gists
- Challenges
 - MOSS script doesn't take more than 4k arguments currently
 - Blocksplorer issues with downloading more than 6k code snippets

Related Work: ClonEvol

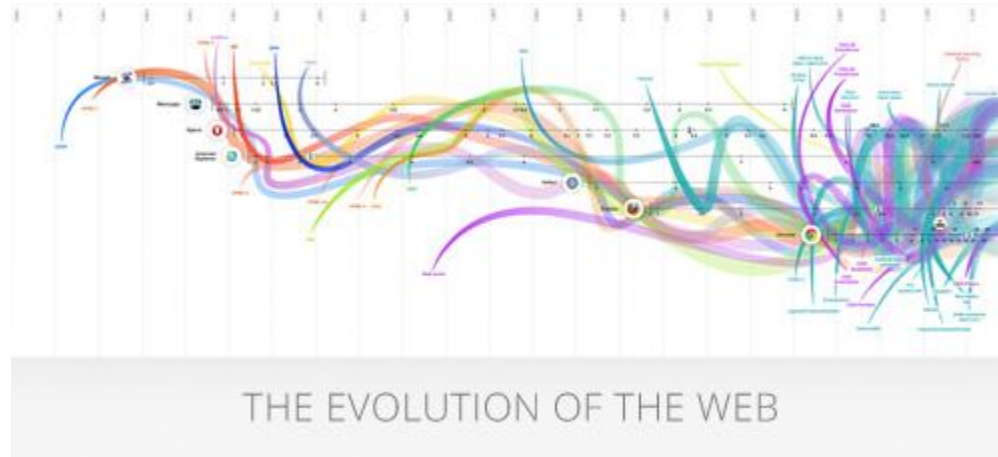
Evolution of FileZilla
(<https://svn.filezilla-project.org/svn/FileZilla3/trunk>)



Evolution of code within single repository over time, in terms of diffs, additions, deletions, modifications, etc.

<http://www.cs.rug.nl/svcg/SoftVis/ClonEvol>

Related Work: The Evolution of the Web



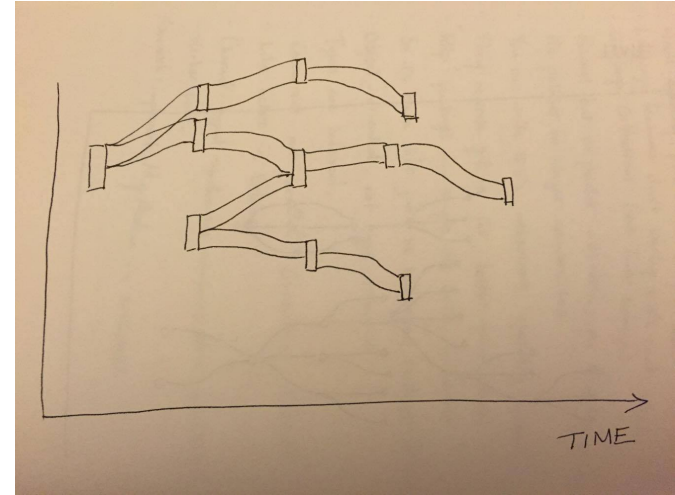
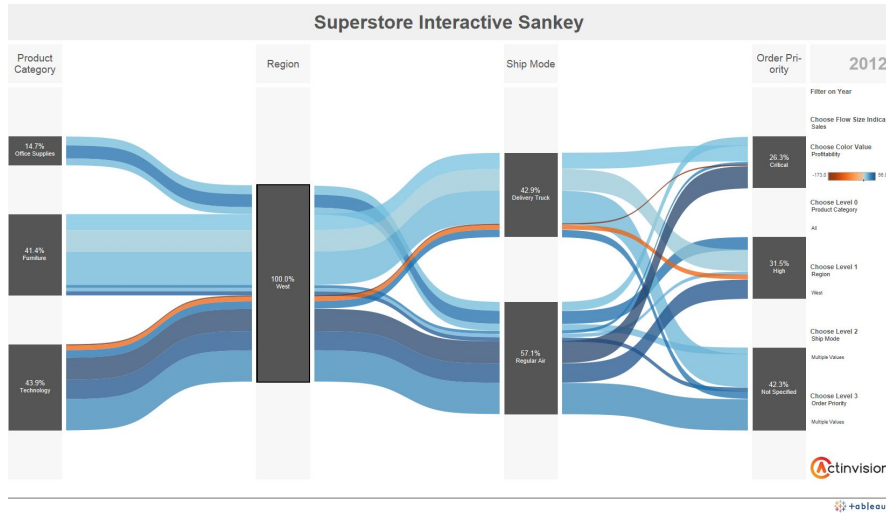
Timeline visualization of which browsers + versions supported which web technologies.

<http://www.evolutionoftheweb.com/>

Current Progress (Completed)

1. D3 code scraper (parses json from blocksplorer, fetches code snippets, outputs relevant metadata).
2. MOSS uploader + data extractor.
3. Data cleaner + formatter
4. Visualization sketches
5. Small scale hand-labeled data visualization

Current Design: Timeline Sankey



Emphasize tracing code snippets or percentage of shared code.

09/15

10/15

11/15

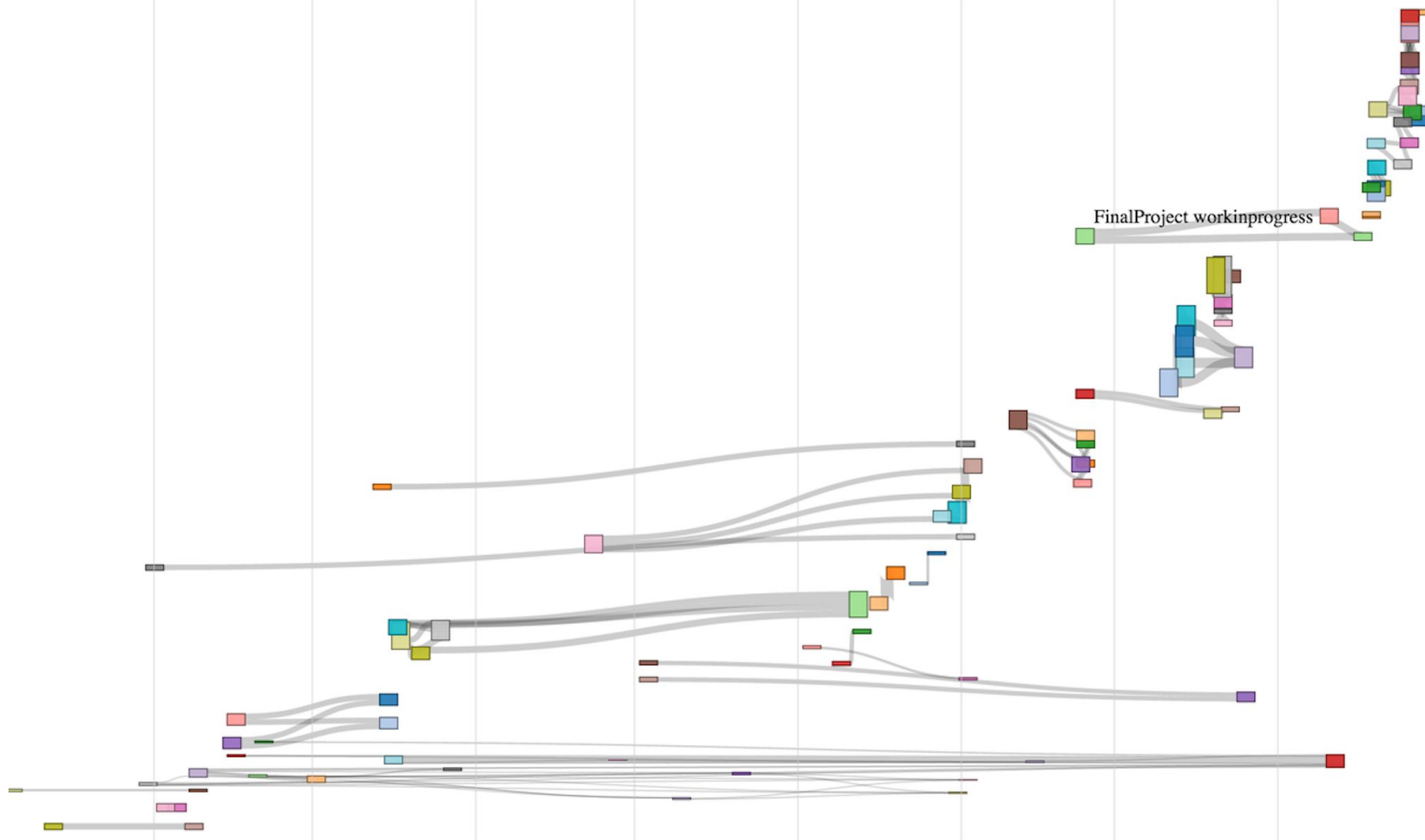
12/15

01/16

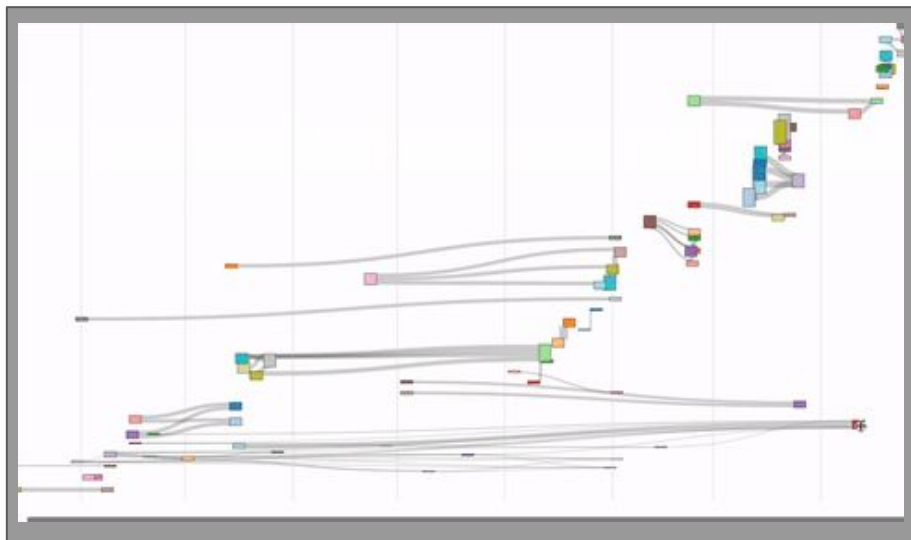
02/16

03/16

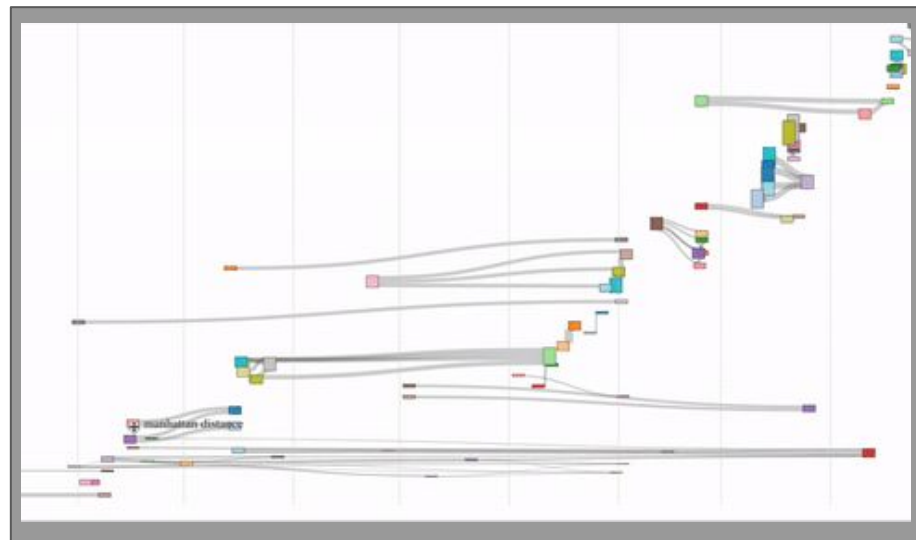
04/16



Current Interactions

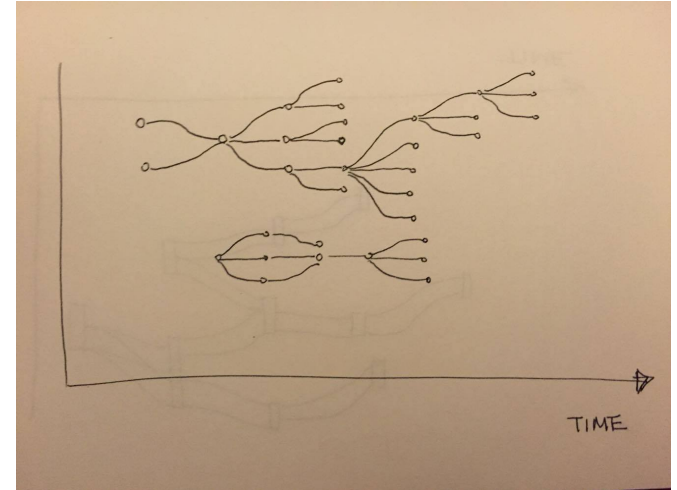
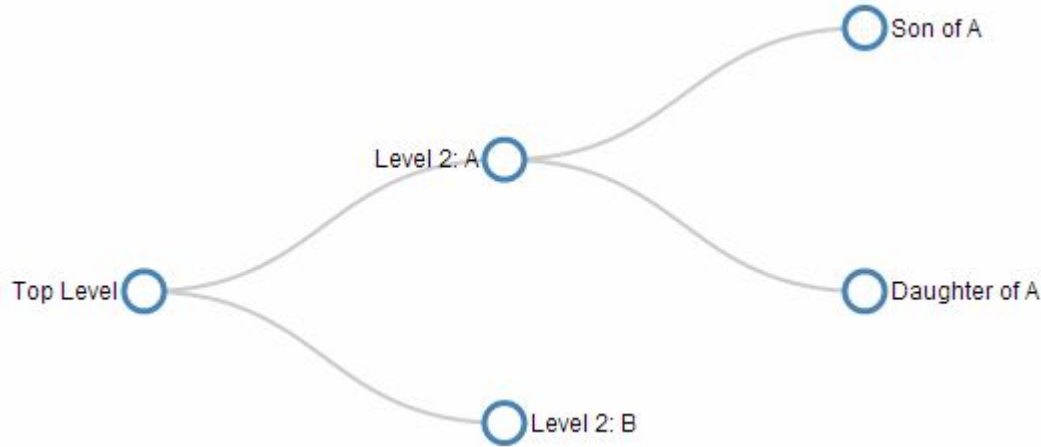


Vertical Drag



Display name on hover

Alternate Design: Timeline Tree



Use collapsibility to emphasize immediate parents or children.

Completion Plan (Next Steps)

1. Streamline data processing for scalability (i.e. refactor steps 1-3 to single script) and get large-scale data.
2. Implement large scale visualization.
3. Interactivity: scrolling, collapsing, drag, zoom.
4. Filters (e.g. by author, by api call, by dates etc.)
5. Metadata and details

Questions for Feedback

What ideas do you have for collapsing data or otherwise managing large visualizations?

What are some potential use cases for this data? What are the interesting relationships / attributes to drill down on?

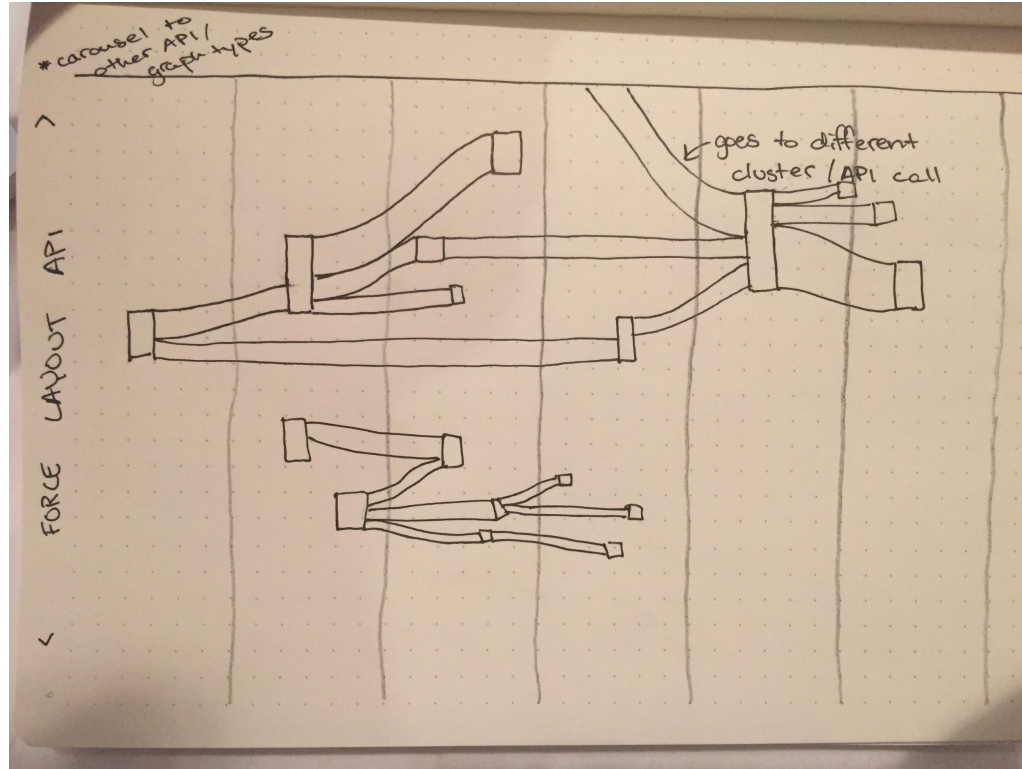
How to best represent grandchildren relationship? We currently identify “parent” by closest timestamp. Beyond direct parent, is ancestry information interesting, necessary, worthwhile?

While Sankeys provide an extra dimension for expressing data (edge width), they also tend to take up more space than a traditional tree diagram. Do you think one type of visualization would better fit our data?

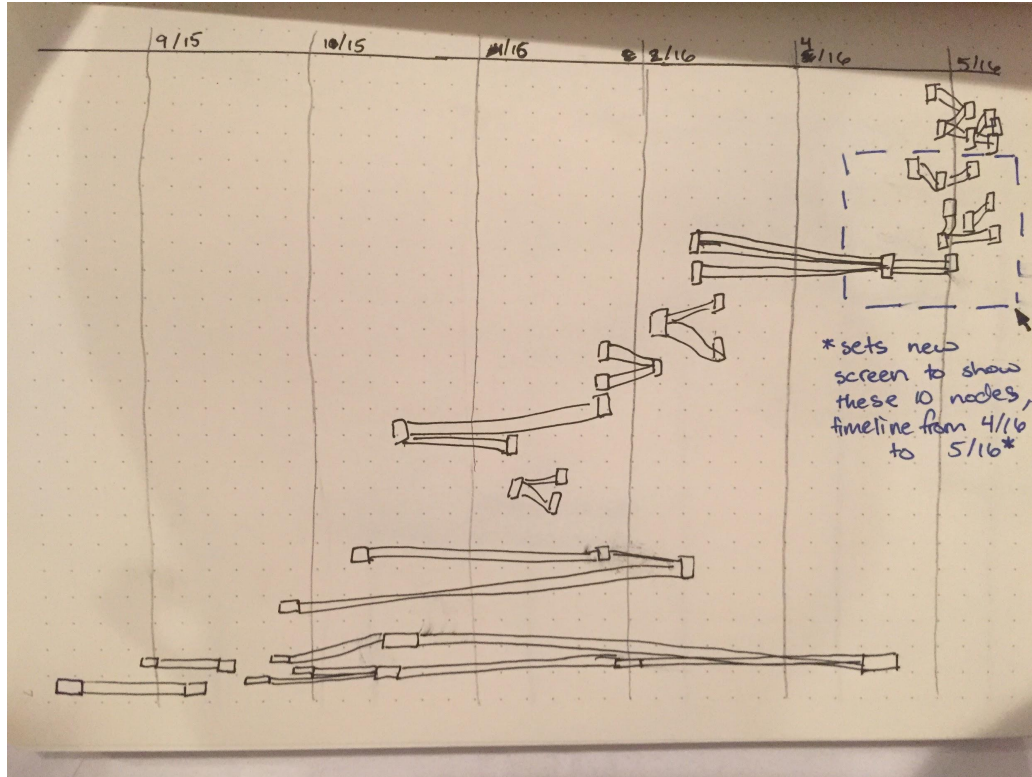
Current Challenges

1. Handling large scale data
 - a. vertical collapse
 - b. horizontal zoom
2. Designing interactivity
 - a. Filters
 - b. Manipulation

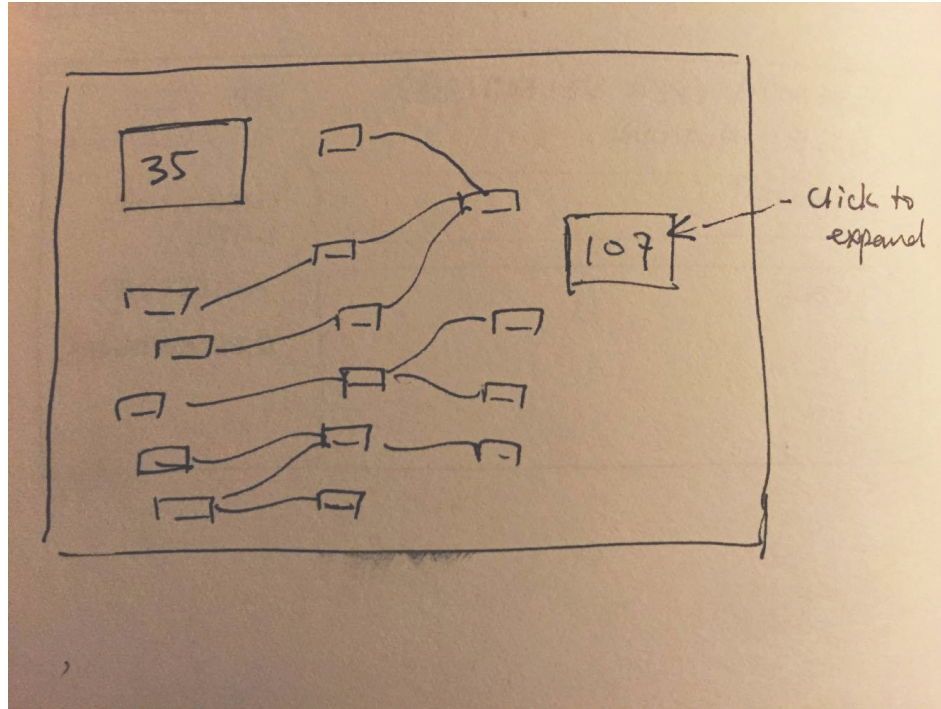
Zoom Carousel



Click+Drag Zooms to Selection



Auto-hide nodes / click to expand



General View

