

# GSoC 2016 Project Ideas - Brainstorming

Originally captured in a [Google Doc](#)

<https://summerofcode.withgoogle.com/organizations/5729245854695424/>

## Ideas based on 2015 project ideas

[Google Summer of Code 2015 with the Fluid Project](#)

- Build tool for generating icon fonts
- Matching content to preferences, related to metadata ( see Joanna's designs )
- webrtc: related to Vidyo infrastructure
- Improved build tools and dependency management system for Infusion

### Data Visualization and Sonification with Infusion

Building from Fluid's `fluid.model.transformWithRules()` and following its Model Relay system for connecting component endpoints, this project will build a method of connecting an Infusion app to an arbitrary data source and transforming this data in preparation to be rendered. Too commonly data pipelines bake in a representational schema that cannot be escaped by a further rendering engine. Otherwise, data is put into a representational framework (ie., a visualization library) that ties the data transformations to the specific rendering elements.

The goal of this project will be to build a functional I/O platform for data rendering so that common-type datasources can be transferred into an application model and transformed into a generic JSON schema that can further be given rules that transpose the data to a representation. Be it audio or visual representations, the platform will utilize the data in kind so as to develop a pattern for developing representational templates that are agnostic to data sources. This will lead to a friendlier, more accessible approach to representing data usefully to end-users.

## Ideas based on Floe Brainstorming

[Floe Brainstorming 2015](#)

- Internet of Things ( related to 'inclusive metrics', dashboard connection, quantified self, etc.)
  - mechanism for exporting data from the Internet of Things to be used by the self-assessment dashboard
- Game for learning-to-learn or first discovery
  - design mentor: Dana
  - [previous work on a first discovery game](#) (PGA Y2)
- Dynamic and/or accessible user states and contexts
- A mechanism for connecting learners with matching/convergent/contrasting learning styles together, to share OER with one another
- Sonification for more types of charts, tables, images, and etc.
  - design mentor: Sepideh
- Consideration for game / gamification of digital literacy (perhaps paired with first discovery tool for basic and early learning)
- Framework for stages of digital literacy development (e.g. speaking recently to someone who was sent as a volunteer to set up e-learning in another country only and ended up teaching keyboarding skills because the workers had never used a computer) – does this fit in what we do?
- Game for early digital literacy that teaches the basics of coding/design/etc. and allows the user to create their environment through customization (some sort of extension of first discovery?) – turning users into makers
- Creation of resources to help train the trainer to be someone who could help develop a user's problem solving skills

## Other Ideas

- Responsive UIO
- Extend metadata demo to accept more video parameters and alternatives
- Something related to authoring accessible content
- Sharing preferences sets, connecting users with common needs
- Automatic a11y tests using something like [deque's axe core](#) or the tool from Google.
  - A project to setup the infrastructure and write the tests
- Interactive and inclusive version of the accessible cards
  - would also allow for printing at different sizes and etc.
  - provide a way to add to the content.
- Vision Technology Service/SNOW – game for teaching kids/adults introductory keyboard shortcuts that are useful for navigating the computer (e.g. using TAB, arrow keys, ENTER; CTRL-TAB, copy cut paste save undo redo; text movement shortcuts like CTRL-arrow keys); it would be a game environment thought, not a Windows/Mac replica; e.g. to move a player along the map, you could use the arrow keys, but to make them jump from town to town (quicker), they could use CTRL with the arrow keys; each location would teach one or two keyboard shortcuts for a function and then perhaps give them a more real life comparison