Google Summer of Code 2018 with the Fluid Project

Update for April 23, 2018

Per the Google Summer of Code Timeline, accepted proposals will be notified today at 16:00 UTC.

The Fluid Project received nearly a hundred proposals for its projects this year, and unfortunately can't accept anything close to all of them.

We want to thank everyone who submitted a proposal, and would encourage all to continue applying to GSoC in future years if you didn't receive a spot with us or another open source organization this year - the program is extremely competitive.

Overview

Fluid is an open source community of designers and developers who help improve the usability and accessibility of the open web. We contribute to a variety of open source projects (e.g. jQuery UI, GPII, PhET), and work on our own projects including:

- Infusion: A JavaScript application framework for developing flexible user interfaces.
- FLOE: Provides the resources to personalize how we each learn and to address barriers to learning
- Social Justice Repair Kit: The goal is to support youth at risk who have learning differences to re-engage in education through an inclusively designed social justice platform that integrates authentic project-based learning.
- The Inclusive Design Guide

Fluid Infusion is built on top of jQuery, providing all the stuff you need to create user interfaces that are incredibly flexible, accessible, and easy-to-use. Infusion is an application framework and a suite of user interface components built with HTML, CSS, and JavaScript. In contrast to many other user interface toolkits, Infusion components aren't black boxes--they're built to be modified, adapted, and changed to suit your application or context. Taking a "one size fits one" approach, Infusion even lets end-users customize their experience with the UI Options component.

We're looking for students to collaborate with us on the Google Summer of Code 2018 program. Working with Fluid gives you a chance to learn more about accessibility, usability, and inclusivity while writing code with cutting-edge open web technologies. Create cool stuff and make a real impact on users at the same time!

For information about the various ways we communicate with each other, see our Get Involved wiki page.

Getting Started

Make sure to read through the Getting Started on GSoC page for information on joining the Fluid community and preparing to work on a GSoC project.

Projects

Inclusive Design Guide Digital-Print copy workflow prototypes

What we're looking for is to be able to provide small scale on-demand printing of the Inclusive Design Guide content. We regularly edit or add content to the Guide and we currently have two versions of the content, web and print. At the moment the print copy needs to be updated separately every time we make changes to the source content. We want to devleop a simple process that ensures that the print copy is always up to date with any changes to the source content. We also want to make it easy for people to print the cards and to be able to pick and choose a subset of the Guide to print to cards if desired. Also, the printable content is slightly different than the web content since we have to reduce it to fit on printable cards, and we need a way to label /tag the printable content.

Project Description:

- Come up with possible workflows and prototypes that facilitate the creation, editing, and printing of the Inclusive Design Guide from a web copy.
- Example Workflow 1: provide a way to mark-up / tag / add metadata on the web copy to identify what to generate the print copy from
- Example Workflow 2: keep two separate copies of the web and print, and maintain a "diff layer" (see desctiption below) to reveal the differences / changes between the web and print copies (in order that the differences can be reconciled).
- Example Workflow 3: a combination of the above
- The web copy will be static HTML (i.e. not using a CMS like Wordpress).
- The print copy will be produced using Adobe InDesign (or other).

    - this could also be done by creating an authoring/editing/templating environment into which the web content (or marked up subset of content) could be imported and then massaged into print copy. Normally the web copy has more graphics and text than the print copy because the printing is done on small pieces of paper (about 8.5 inches by 5.5 inches). So there will need to be a way to say what parts of the web content is to be printed, and what parts are for web only.
- Considerations should be given to ease of use, maintainability, accessibility, and standards compliance.
- A GSoC candidate will be responsible to researching workflows, documenting each approach, and implementing prototypes.

Tag: fluid

Difficulty: Medium.

Mentor: Dana Ayotte, Jonathan Hung

IRC: danayo, jhung

Skills: HTML, CSS, Print workflows

Is this a coding project? What skills are required?

This is not a traditional coding project. Much of this project is researching possible solutions and workflows that will satisfy the project goal. A prototype or implementation can be created to demonstrate the feasibility of a solution.

A good candidate for this project should be able to demonstrate strong ability in independent research, and technical skills to build a web-to-print workflow. The exact technical skills are open and will depend on the ideas the candidate discovers during research. Helpful skills will be knowledge of HTML, CSS, Javascript, and tools that assist code automation (i.e. node, grunt, Python etc.).

What is meant by a "diff layer":

- A "diff layer" is some sort of middleware that tracks changes on either the print side or the web side so that maintainers can ensure that changes are reflected in both online and print.

Additional Information:

- Github repository for source files for the Inclusive Design Guide website
- Inclusive Design Guide website
- Inclusive Design Guide PDFs used for printing

Import External Data to MyL3

Project Description: *'My Life Long Learning Lab'* (MyL3) allows learners to become experimental researchers in subject of their own learning. MyL3 not only provides custom tools for learners to track their personal data, but it also allows them to import data from external sources to the system. This would enable learners to track various factors in one place, find potential correlations that may impact their learning, and make adjustments accordingly.

This project focuses on importing data from the following external sources to the MyL3 tracker to get real time data into the system:

- Weather
- Location
- Personal Health App (TBD)

Tag: fluid

Difficulty: Medium

Development Mentor: TBD

Design Mentor: Sepideh Shahi

IRC: sepidehshahi

Skills required: JavaScript, HTTPS

Upgrade AChecker

Project Description: AChecker is an online accessibility validator. It currently supports PHP versions up to PHP5. Also, some libraries used by AChecker, such as PHP HTML DOM Parser and PclZip, are no longer supported by their development team, which results in the same issue of being stuck at old PHP versions. This project includes:

1. Upgrade AChecker to work with the latest PHP and MySQL;
2. Replace or patch non-supported libraries.

References:

- Migrating from PHP 5.6.x to PHP 7.0.x
- PHP libraries used by AChecker

Tag: achecker

Difficulty: Medium

Mentor: Cindy Li

IRC: cindyli

Skills required: PHP, Javascript, HTML, CSS.

How to get started: Interested students should start by cloning AChecker github repository, setting up a local installation, getting familiar with its design and the code base. When ready:

1. Pick some simple bugs from AChecker bug tracker to work on. (Note that when using the bug tracker, please first select "AChecker" project from the "Project:" drop down list box at the top right corner.)
2. Understand and create a list of parts that need to be upgraded, research on how to upgrade and present the research result in the proposal.
3. When upgrading libraries, the order of the preferred ways:
    a. Replace with PHP built-in features whenever possible;
    b. Replace non-supported libraries with other libraries that are under active development and support the latest PHP;
        i. Write technology evaluations when selecting replacement libraries. Some examples of technology evaluation can be found here.
    c. Patch the library to work with the latest PHP.

Implement dynamic descriptions for an interactive PhET simulation

PhET Simulations creates free interactive math and science simulations. The project has over 55 HTML5 simulations with graphical interfaces that foster learning and exploration. We are now adding dynamic descriptions and content to the simulations so that they are accessible with assistive technology. This project will involve adding screen reader accessible descriptions and real-time alerts.

PhET simulations are implemented with a custom scene graph called Scenery. The Scenery API now supports setting accessible content for objects in the display. The project will involve using this API to enhance existing simulations with accessibility. Students will work on implementing accessible descriptions and alerts from a design document.

Tag: phet

Difficulty: Medium

Mentor: Jesse Greenberg

IRC: jessegreenberg

Skills: Javascript, knowledge of object oriented design patterns and the scene graph data structure a big plus

How to get started: Interested students should start by familiarizing themselves with the structure of a PhET simulation. All code is open source, here is one example: Balancing Act. Interested students should also review the PhET Development Overview to get familiar with PhET's libraries, code style guidelines, how to set up a development environment.

Build a service for adaptive content and learning supports

Project Description: Many recognized supports for learning and comprehension of content (especially by students with learning differences or people with cognitive disabilities) such as dictionary look-up, translation, spell check, pronunciation and simplification are available via web-based APIs; others are available as open-source modules in various languages.

This project would explore building an application functioning as service to make it easier for different applications to make use of these kind of services, as each third-party service has different patterns for access, rate limiting, the open-source libraries have different APIs and levels of maturity, etc. The application would provide a consistent, simplified API via REST web services for functionality such as (these are examples only, many more possibilities!):

- Returning the three most common dictionary definitions of a word.
- Simplifying text.
- Translating text into other languages.

Some of this project would be working directly with third-party APIs, but a significant part of it would be working out a general design for a service that could grow in the future and add further features related to transforming content to support learning and comprehension. There would also be a research component to identify and categorize third-party services and modules of interest - some examples of online dictionary services and libraries are discussed here. The student would have the opportunity to establish a starting point for future work on an open middleware-type application that could be of significant use in areas such as education and accessibility.

Tag: fluid**,** scaffoldingapi

Difficulty: Medium

Mentor: Alan Harnum

IRC: alanharnum

Skills: Javascript (including Node), API design, working with third party APIs

How to get started:

- You should familiarize yourself with Infusion, the Javascript framework we use for our projects. If you like to learn via hands-on, the interactive Developer Introduction to basic concepts of Infusion may be helpful.
- Kettle is an Infusion-based server-side framework based on Express, which we'd likely use for this project.

- A practical starting point would be building a simple dictionary web service on top of one of the dictionary services and libraries discussed here. Defining the public REST-style API would be a good first step before proceeding to implementation.
- As an example only, a simple conceptual diagram is available here; URLs and JSON are examples only, and the diagram doesn't adhere to any formal style.

Create Self-Paced "Live Learning" Infusion Tutorials

Project Description: To grow our community and engage with both expert and novice developers, the Infusion documentation site currently provides a mix of documentation and tutorials. Our tutorials are static documents with code examples that can be modified in real time on CodePen. This project would improve upon this by creating self-paced tutorials that encourage the user to verify their understanding of the material using live coding exercises. Live coding exercises present concrete goals to be reached and provide feedback in real time as goals are reached.

This "live learning" harness would require the learner to:

1. Review prior art within and outside of the Fluid community.
2. Create or extend a simple live coding harness like CodePen or the Fluid Sandbox.
3. Define or adapt an existing structure that describes a "live" tutorial, including:
    a. The learning material to be read.
    b. Exercise instructions for each live coding scenario.
    c. The starting code, markup, and CSS for each live coding scenario.
    d. Clearly expressed goals to be reached in the live coding scenarios, and test conditions that can be used to verify when a goal has been reached.
4. Break down one or more existing (or new) tutorials to use this structure, and provide a document to assist others in creating their own "live learning" tutorials.
5. Create or extend a mechanism to record which tutorials and parts of tutorials have already been completed, so that learning can be broken up over multiple sessions.

The last point touches potentially on issues like user management and data privacy, for this project it's more likely the progress would be recorded locally, for example using the Web Storage API.

Tag: fluid

Difficulty: Medium

Mentor: Tony Atkins

IRC: the-t-in-rtf

Skills: Javascript and HTML, mostly front end, largely using existing Fluid components. Limited work with standard APIs like the Web Storage API.

How to get started: Go through one or more existing Infusion tutorials. Try out at least one of the live examples (usually linked via text like "Live Example of the code below on CodePen").

Accessible Music User Interface Toolkit with Nexus.js

Project Description: Currently, there are no web-based user interface libraries that are inclusive of the needs of diverse musicians working with different platforms, devices, and interaction modes. This project involves collaborating with the Nexus.js, Flocking, and Fluid communities to add accessibility features to Nexus.js or a similar musical user interface library. Nexus.js provides a useful collection of musical UI components, but in its present incarnation, it is not usable by people with disabilities (such as those who use screen reader) nor by those who prefer to use the keyboard as much as possible. This project will include ensuring that the user interface components are:

1. Responsive to different screen sizes and devices (e.g. phone, tablet, and desktop)
2. Functional using both touch and a mouse cursor
3. Navigable and controllable entirely with the keyboard
4. Usable with assistive technologies such as magnifiers and screen readers.

This project may also involve writing adapters, as needed, for Nexus.js that allow it to be used more idiomatically with Fluid Infusion-based components.

To get involved in this project, please see the instructions on the IRC Channel page, which describe how to set up an IRC client so that you can join the #fluid-work chat channel, where many contributors to the Fluid community are available.

Tag: fluid**,** flocking, nexus.js

Difficulty: Medium

Mentor: Colin Clark

IRC: colinclark

Skills: Javascript and HTML, mostly front end. Knowledge of electronic music production software and techniques.

How to get started: Learn about Nexus.js. Go through one or more existing Infusion tutorials. Read about ARIA and accessible web user interface development.

Inclusively Design & Build a Game for Kids

Project Description: Young children often learn through play, exploration, and discovery. Kids who use eyegaze or switch interfaces are often limited to using games that have simple interfaces such as 'choose an answer' style games. This does not give kids a chance to play, discover, and explore. How can you create a game with simple enough controls that still preserves the ability to explore and discover through play?

An additional use case involves very young children who are blind. While sighted toddlers are swiping and playing digital games at an early age (and thereby acquiring digital literacy), their peers who are blind are unable to do the same. With the current games, until they have language, children who are blind are unable to acquire digital literacy. How can a game be created that works for toddlers who are blind and helps them begin a journey toward digital literacy?

It would be cool to develop a game for these kids. (This project will likely be done in partnership and close collaboration with Beit Issie in Tel Aviv, Israel and Bloorview Children's Rehabilitation Hospital in Toronto).

Tag: play, kids, education, exploration, fun

Difficulty: medium

Mentor: Primary: Jess Mitchell; Technical: Alan Harnum

Skype: jesshmitchell on irc @jessm and @alanharnum

Skills: Javascript, HTML, CSS

How to get started: Be excited and curious and willing to learn new things.

Infusion Documentation

Most of the work we do here either uses or directly involves the Infusion Framework and Component Library. These links should get you started learning about Infusion, and should lead you to many more pages.

- Contributing Code To Infusion
- Infusion Documentation
- Tutorial - Getting started with Infusion
- Tutorial - Developer Introduction to the Infusion Framework
- An Infusion Pattern Language