

Google Summer of Code 2020 with the Fluid Project



Over the last week, many of you have submitted draft proposals and are eagerly awaiting feedback from the project mentors. We would like to assure you that we will be providing feedback in the next couple days, but you should know that some of the mentors have a lot on their plates at the moment, especially considering recent developments related to COVID-19.

We ask that you please bear with us, and we thank you for your patience and understanding.

To those of you who have yet to submit your draft proposals, please know that we cannot accept any proposals outside of the GSoC website, and that only students who submit a final proposal will be eligible for acceptance. If you have any questions about this part of the process, please feel free to reply to contact us using the methods detailed below.

Table of Contents:

- [Getting Started](#)
 - [Where and how to contact us](#)
 - [Read about Inclusion](#)
 - [Read our Contribution Guidelines](#)
- [Who are we?](#)
- [What's in it for you?](#)
- [Projects](#)
 - [Port UIO+ Chrome Extension to Firefox and Safari; add more adaptations](#)
 - [Migrate FLOE and Fluid project websites to a Modern Static Site Generator](#)
 - [Refactor and Automate Infusion Documentation](#)
 - [Using a Game Controller as a Navigation Aid](#)

Getting Started

Don't panic! Please read through this page first, it has many answers! :)

TL;DR:

- Reach us on [IRC](#) or the [mailing list](#)
- Read the "How to get started" for the project you're interested in
- Please be respectful of mentors and all others
- Please don't expect immediate responses, we'll do our best to reply quickly

Where and how to contact us

For feedback during North American business hours, please reach us on the [#fluid-work IRC channel](#) on Freenode. Please note that some mentors may appear online but may not be active or able to respond to messages immediately. Alternatively, you can reach us using the [fluid-work mailing list](#). We tend to use the mailing list for asynchronous discussions that may not work so well in a chat room setting.

- [#fluid-work IRC channel](#): <https://wiki.fluidproject.org/display/fluid/IRC+Channel>
- [fluid-work Mailing List](#): <https://lists.idrc.ocad.ca/mailman/listinfo/fluid-work>

We encourage open and transparent communication within our community. With that in mind, please contact us in the [#fluid-work](#) channel and on the mailing list. As mentors have a number of other responsibilities, it may take us some time to reply. Please allow mentors the opportunity to get back to you in their own time. Please also try to keep questions specific to a particular project. We expect that GSoC applicants communicate with respect for everyone.

Read about Inclusion

It is important to read and understand [Inclusion in the Fluid Community](#). We follow this set of values that promotes inclusion, cooperation, mutual respect, openness and the sharing of ideas. We expect students to be helpful and supportive and to openly communicate their work process in a manner that aligns with our values.

Read our Contribution Guidelines

To keep ourselves organized and consistent, we do our best to make sure our work follows [these Contribution Guidelines](#). Some projects may have their own specific guidelines.

Try not to focus on existing issues with these projects — it is more valuable to invest into research, understanding possible solutions, and participating in the community. This will help you with your GSoC proposals.

Who are we?

[Fluid](#) is an open source community of designers, developers and researchers who help improve the usability and accessibility of the open web.

What's in it for you?

Working with Fluid gives you a chance to learn more about accessibility, usability, and inclusion while writing code with cutting-edge open web technologies in an international open source community. These project descriptions are flexible and give you room to make your own mark and propose your own approach.

Projects

Port UIO+ Chrome Extension to Firefox and Safari; add more adaptations

User Interface Options Plus (UIO+) allows a user to customize websites to match their own personal needs and preferences. Settings for the adaptations can be set via the UIO+ adjuster panel or, if on a **Morphic**-enabled machine, from a keyed-in preference set. UIO+ currently supports a dozen or so adaptations, including contrast; line, word and character spacing; table of contents, text-to-speech and zoom.

The primary task of this project will be to port the extension to work in both the Firefox and Safari web browsers. Ideally the same codebase should be used to build the extensions for all browsers. Once the ports are successful, additional adaptations should be created to provide more user customization options. The specific adaptations should be decided on with the mentors.

Tag: uio+

Repository: <https://github.com/GPII/gpii-chrome-extension>

Issue tracker: <https://issues.gpii.net/issues/?q=project%20%3D%20GPII%20AND%20component%20%3D%20%22Web%20Personalization%20Browser%20Extension%22>

Mentors and their IRC usernames: Justin (*Justin_o*), Cindy (*cindyl*)

Skills needed: JavaScript

How to get started:

1. You will need to have access to a macOS machine for the Safari extension.
2. Familiarize yourself with how the extension works from a user's perspective and with the `code` — try building, installing, and running the unit tests.
3. Research the browser extension APIs and requirements for Firefox, Safari and Chrome.
4. The code is mostly written using **Infusion**, the Javascript framework we use for our projects. If you like to learn in a hands-on way, the interactive **Developer Introduction to basic concepts of Infusion** may be helpful. The porting will likely require less interaction with the Infusion code, but as you move to writing adaptations, you'll need to be prepared to make use of the framework. It will also be helpful for understanding the current structure of the code as you work through the porting process. You should familiarize yourself with how the extension works from the user's perspective and with the `code` (including building, installing, and running the unit tests).

Migrate FLOE and Fluid project websites to a Modern Static Site Generator

Migrate the FLOE Project and Fluid Project websites to use a modern static site generator (like Hugo, Jekyll, etc.). The FLOE Project site is currently using static, hand-coded HTML, while the Fluid Project site is currently built using DocPad.

It is important that the new sites are accessible, work reliably and have well-defined processes for updating and adding new content. Fixes to the User Experience (UX) are not a priority, as we may redesign the site in the future.

Tag: websites

Repositories:

- FLOE Project: <https://github.com/fluid-project/floeproject.org>
- Fluid Project: <https://github.com/fluid-project/fluidproject.org>

Mentors and IRC usernames: Jon (*jhung*), Gregor (*gmooss*), Colin (*colinclark*)

Skills needed: HTML, CSS, markup templating, JavaScript

How to get started:

1. Understand the broader goal and research possible solutions to creating, maintaining, and automating static websites.
2. Familiarize yourself with the existing websites and their github repositories.
3. Examine other automated website repositories at the IDRC for inspiration:
 - Social Justice Repair Kit: <https://github.com/fluid-project/sojustrepairit.org>
 - Coding to Learn and Create: <https://github.com/codelearncreate/c2lc-websit>

Refactor and Automate Infusion Documentation

Update the Infusion Documentation to be generated using Hugo or another actively-maintained site/documentation generator instead of DocPad, with the Infusion documentation content residing in a separate repository. The goal is to improve maintainability, functionality and flexibility (ability to move to a new site generator easily).

Any changes to documentation or creation of new tagged releases should trigger automation to build new documentation.

The project is required to support tagged versions of Infusion documentation — the technology will need to output and present these tagged versions. Currently there are 3 tags:

- Infusion 1.5,
- Infusion 2.0,
- and Master.

Tag: documentation automation

Repository: <https://github.com/fluid-project/infusion-docs/>

Mentors & IRC handles: Jon (*jhung*), Justin (*Justin_o*), Gregor (*gross*)

Skills needed: HTML, CSS, Markdown, markup templating, JavaScript

How to get started:

1. Checkout the current Infusion-docs repository and see how the documentation is currently built using Docpad.
2. Start thinking of ways to refactor the content from the rendering technology.
3. Familiarize yourself with web tools that may be used to generate documentation located in a different repository.
4. Review and evaluate other static / documentation site generators and workflows for managing content; it is possible that a solution other than Hugo may make more sense

Using a Game Controller as a Navigation Aid

Project Description:

Projects like [the Xbox Adaptive controller](#), the [Logitech G Adaptive Gaming Kit](#), and various [modified controllers](#) strive to include a wider range of people in traditional PC and console gaming. The goal of this project is to extend the usefulness of adaptive controllers by making it possible to navigate using a game controller as an alternative to a keyboard or mouse.

As a proof of concept, we should at least be able to demonstrate support for [keys commonly used for browser navigation](#), such as the arrow keys, tab, shift, space, enter, and escape. The ideal solution would also allow the user to select which actions to associate with which gamepad inputs, including key combinations, macros, and possibly simulated mouse input.

Tag: game controller

Mentors & IRC handles: Tony (*the-t-in-rtf*)

Skills: JavaScript, HTML

Is this a coding project? Yes

Additional Information:

- This project would likely use the [HTML5 Gamepad API](#), which [is well supported in a range of browsers](#), and provides the ability to use a huge range of inexpensive game controllers with a browser, including Xbox, Playstation, and PC joysticks.
- This project will involve developing a solution that can be used with external sites and applications that do not themselves provide gamepad support. It is likely that the candidate would work with something like [Electron](#) or [a custom Chrome extension](#).

How to get started:

1. Try browsing the web for a half hour using only your keyboard to navigate. Write up your experience and think about how you might improve this for others.
2. If possible, find a controller compatible with the HTML5 Gamepad API, such as a PS4 or Xbox controller. Try the controller out with an HTML5 gamepad tester. Look at how some inputs are digital (only sending 0 or 1) and how some inputs are analog (sending a range of values instead). Think about how you might use each type of control to improve the experience of navigating.
3. Read up on the underlying technical topics, particularly Electron and the Chrome Extension. Although you will have help, you will need to demonstrate that you will be comfortable building a solution in either of these well-documented but highly technical environments. Going through a tutorial or two on each topic might help you prepare.