Google Summer of Code 2020 with the Fluid Project

Overview

- Overview
- Getting Started
  - Infusion Documentation
- Projects
  - Port UIO+ Chrome Extension to Firefox and Safari; add more adaptations
  - Migrate FLOE and Fluid project websites to Hugo Static Site Generator
  - Refactor and Automate Infusion Documentation
  - Using a Game Controller as a Navigation Aid

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 2020 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.

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

Projects

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

**Project Description:**

Port UIO+ Chrome extension to Firefox and Safari, and 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 (https://morphic.world/) enabled machine, from a keyed in preference set. UIO+ currently supports the following adaptations:

- Captions (embedded YouTube videos)
- Character Space
- Contrast
- Enhance Inputs
- Line Space
- Reading Mode
- Right-Click to Select
- Selection Highlight
- Syllables
- Table of Contents
- Text-to-Speech (only for selected text)
- Word Space
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+

**Difficulty:** Medium

**Mentor:** Justin, Cindy

**IRC:** Justin_o, cindyli

**Skills:** JavaScript

**Is this a coding project?** Yes

**Additional Information:**

You will need to have access to a macOS machine for the Safari extension.

**How to get started:**

- 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).
- You should research the browser extension APIs and requirements for Firefox, Safari and Chrome.
- The code is largely written using Infusion, the Javascript framework we use for our projects. If you like to learn via hands-on, the interactive *Developing an 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.

**Useful Links:**

- Repository
- Issue Tracker

---

**Migrate FLOE and Fluid project websites to Hugo Static Site Generator**

**Project Description:**

This project task is to migrate the FLOE Project and Fluid Project websites to use Hugo static site generator.

**Tag:** websites

**Difficulty:** Low

**Mentor:** Jon Hung, Gregor Moss, Colin Clark

**IRC:** jhung, gmoss, colinclark

**Skills:** Hugo, HTML, CSS, markup templating, JavaScript

**Is this a coding project?** Yes

**Additional Information:**

- FLOE Project website github repository
- Fluid Project website github repository

**How to get started:**

- Familiarize yourself with the existing website github repositories.
- Examine other Hugo website repositories at the IDRC like Social Justice Repair Kit website and Coding to Learn and Create website.

---

**Refactor and Automate Infusion Documentation**

**Project Description:**

The project will automate the generation of tagged versions of Infusion documentation using Hugo, with the Infusion documentation content residing in a separate repository.

The project will need to address and support these processes:

- Documentation is required to support tagged versions of Infusion documentation - Hugo will need to output and present these tagged versions. Currently there are 3 tags: Infusion 1.5, Infusion 2.0, and Master.
- Any changes to documentation or creation of new tagged releases should trigger automation to build new documentation.
There are some issues to consider:

- Hugo will require platform specific front matter / metadata to render content. Where will this information exist while keeping the documentation content independent and clean of rendering technology metadata?
- How should the content be structured and organized in the content repository so it is logical and understandable?

**Tag:** websites, Hugo, documentation automation

**Difficulty:** Medium

**Mentor:** Jon Hung, Justin

**IRC:** jhung, Justin_o

**Skills:** Hugo, HTML, CSS, markdown, HTML templating, JavaScript

**Is this a coding project?** Yes

**Additional Information:**

Infusion documentation currently exists in this github repository which uses Docpad to render the content.

**How to get started:**

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

**Difficulty:** Medium

**Mentor:** Tony Atkins

**IRC:** 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:**

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.

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.

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.