Google Summer of Code 2016 with the Fluid Project

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 (such as jQuery UI), and we work on a few projects of our own: the Design Handbook, a guidebook of techniques for improving usability, and Infusion, a JavaScript application framework for developing flexible user interfaces.

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 2016 program. Working with Fluid gives you a chance to learn more about accessibility and usability 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.

Projects

Game for First Discovery of Preferences

The goal of this project is to build a game (or game-like tool) that allows for the first-time discovery of digital preferences.

Students and other users of this tool will engage in a process of "learning to learn" - that is, discovering and choosing the preferences that work best for them. This may be directly applicable to a learning environment (for example, preferences that might help someone learn math), or it may be more general (preferences that help someone fill out a form on the internet). In many cases, these preferences are the same (for example, high contrast helps me to see the screen better, so I can complete an on-line math course). Preferences may include things like high or low contrast, text-to-speech, or simplify-content. For a list of preferences, please refer to the following links: Cloud 4 All Common Terms, PGA Preference Categorisation.

The goal of this tool is to introduce the user/player/learner to the experience of setting digital preferences, and to provide a method for doing so in a playful and engaging way, aimed at user groups who may not have a lot of experience in using digital devices. Providing a fun way to discover preferences and try them out in a non-intimidating environment is an important part of the learning-to-learn process. The Discovery Cats design mockup shows one approach, where preferences are set upfront in a game-like interface; once this process is complete the user enters the game itself. Another approach would be to integrate preference discovery and selection into the game in itself in a creative way (e.g. how can setting preferences help a player reach a specific goal or fulfill a quest?).

This project will involve working with the design team to make any necessary refinements to the designs and to implement a fully functional game. It is expected that students will make use of Fluid Infusion and any other appropriate web technologies and frameworks to implement the web based game. The game should be implemented such that it is fully controllable through the mouse, keyboard, and touch interfaces and make use of ARIA attributes for Assistive Technologies.

Stretch goal: Consider also the addition of a "dashboard" or other interface that allows the user/player to keep track of their progress over time. This would enable users to measure and track data regarding their own performance in relation to the preferences they set.

See also: Preference Framework and First Discovery Tool for more information and examples on preference editing tools

For latest updates on the project: Google Summer of Code (GSoC) 2016 Project Progress Repository

Difficulty: medium

Mentor: Dana Ayotte (design), Justin Obara (dev)

IRC: danayo, Justin_o

Skills required: JavaScript, CSS, and HTML. Familiarity with canvas, SVG and/or javascript game engines a bonus.

Data Visualization and Sonification with Infusion

Building from Fluid's `fluid.model.transformWithRules` API 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 (i.e., 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.

JavaScript SoundFont 2 Parser and Synthesizer Engine

SoundFonts provide a means for packaging and distributing audio samples for use in wavetable synthesizers and samplers. They typically provide a variety of instrument sounds sampled at different pitches and octaves, making it easy to create realistic-sounding digital instruments. SoundFonts are particularly useful for data sonification, since they provide a simple and low-cost way to give users the ability to choose from a variety of instrumental sounds when creating their sound designs.

The Floe Project (Flexible Learning for Open Education) is developing new tools for sonification and data presentation using audio. These tools are based on Flocking, a framework for audio signal processing, synthesis, and music composition, which uses the Web Audio APIs now built into most modern web browsers.

However, neither Flocking nor the Web Audio API currently provides any support for parsing or playing back SoundFont-based instruments. While there are JavaScript libraries (such as MIDI.js and the soundfont-player library) that claim to provide SoundFont support, these are all based on an approach that extracts the sound files from a SoundFont with a fixed duration and envelope, significantly limiting the usefulness and expressiveness of the SoundFonts.

This project will entail the development of a robust, pure JavaScript SoundFont 2 parser as well as a Flocking unit generator that is capable of expressively playing a SoundFont. The student may choose to take the sf2-parser library written by Gree and modified by Colin Clark as the basis of the project. The parser should support all the major features of the SoundFont 2 specification, and should be able to parse standard .sf2 files and expose their contents in a data-oriented, JSON-style data structure. The playback unit generator should provide inputs that can control and modulate all of the essential parameters of a SoundFont. All code written should be accompanied by unit tests. Along the way, the student is encouraged to create an awesome musical demo of their work.

Difficulty: Medium

Mentor: Colin Clark

IRC: colinclark

Skills required: In-depth knowledge of JavaScript and web development. Knowledge of digital audio and MIDI.

Accessible, Responsive Music UI Controls

With the introduction of the Web Audio API and music frameworks such as Flocking, it's possible to make music and develop custom instruments entirely using Web technologies.

A variety of user interface component libraries, such as Nexus UI, jQuery Kontrol, Interface.js and G200K's Polymer controls, have been developed to assist in the creation of musical interfaces. However, the majority of them aren't very "web-like." Many are based on Canvas or bitmap images, and aren't compatible with responsive design techniques, can't be easily re-styled or customized using tools like CSS, and aren't accessible via the keyboard or with assistive technologies such as a screenreader.

This project will involve the creation of a small collection of high-quality, responsive, SVG or DOM-based musical user interface controls such as knobs, sliders, x/y pads, button matrices, envelope editors, or waveform viewers. The student is free to choose which components to build, but each component will support extensive customization via CSS, will support use on mobile, tablet, and desktop devices, will include ARIA markup for assistive technologies, and will be fully controllable with the keyboard. Where visual presentations convey real-world controls (such as rotary knobs), the mouse and touch interactions will be consistent with the metaphor (e.g. rotary knobs should support a circular gesture for increasing and decreasing the value, not just a linear up/down mapping). An interaction designer from the Fluid Project community will be available to help with visual and interaction questions from the student throughout the project. These controls should be compatible with Flocking and Fluid Infusion.

Difficulty: medium

Mentor: Simon Bates & Michelle D'Souza

IRC: simonjb michelled

Skills required: In-depth knowledge of JavaScript, CSS, and HTML or SVG. Familiarity with common music applications and user interfaces on desktop and mobile.

Implement User Interface / Learner Options Responsive Design

User Interface Options (UIO), also referred to as Learner Options, is a tool which allows a user to customize a web page or application to their own specific needs and preferences. The current implementation has been designed for a traditional desktop experience; however, as the trend to mobile devices continues it is necessary to provide a responsively designed implementation that will work equally well for those on any size device. An initial set of designs have been mocked up by the design team, providing a starting point for the work.This project will involve working with the design team to make any necessary refinements to the designs and to implement a fully functional responsive design that will allow UIO to work in a user friendly manner on screens /devices of all sizes (e.g mobile, tablet, desktop) and respond appropriately to user preference adjustments that a user makes. The student should expect to make use of CSS, in particular media queries, for implementing the responsive design. Ideally the CSS will be written using the Stylus CSS preprocessor. There may be additional JavaScript coding and HTML changes required to facilitate changes in the new designs.

Difficulty: low - medium

Mentor: Jonathan Hung

IRC: jhung

Skills required: JavaScript, CSS, Stylus and HTML. Familiarity with responsive layout techniques (e.g. fluid layouts, media queries).

WebRTC Echo/Sound Test Application

WebRTC is an open framework for the web that enables Real Time Communications in the browser. It includes the fundamental building blocks for high-quality communications on the web, such as network, audio and video components used in voice and video chat applications. These components can be accessed through a JavaScript API, enabling developers to easily implement their own RTC web application.

Vidyo is a videoconferencing solution that enables high definition, low-latency, error resilient, multi-point video communication to both desktop and room system end points across general purpose IP networks. It was the first industry solution to support the H.264 SVC (Scalable Video Coding) standard for video compression and was part of the initial design of Google Hangouts. Vidyo also offers a WebRTC server that allows web browsers to make calls and join conferences without any software installation. This means that participants joining through WebRTC can interoperate with clients in other platforms supported by Vidyo, like native Vidyo endpoints as well as third party H.323, SIP, and Microsoft Lync clients.

One common issue in video conferences is adjusting volume levels across participants. It's often the case that a participant will sound too quiet or too loud, even with automatic volume configuration being provided by some clients. Participants then have to blindly adjust their microphone's volume level and ask other participants if now they sound okay. This is a costly process that often delays web conferences and causes unnecessary distraction. It makes for inefficient, sometimes embarrassing experiences for remote users. We want everyone to feel welcome and heard.

The goal of this project will be to build an application (and accompanying HTML5 website) using the WebRTC API that allows participants to connect to a video conference and test their volume levels by having their voice echoed back. This could be done by asking the participant to say something for a pre-defined amount of time and echoing it back to her. Another solution is to make the echoing constant but with a small delay so the participant can keep saying words and hearing back how she sounds in almost real-time.

Difficulty: medium

Mentor: Giovanni Tirloni

IRC: gtirloni

Skills required: In-depth knowledge of JavaScript. Familiarity with web conferencing technology.

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
Infusion Framework Best Practices

Good First Bugs

T	Key	Summary	Assignee	Reporter	P	Status	Resolution	Created	Updated	Due
	FLUID-5135	Reset on grades demo also resets schema demo	Unassigned	Michelle D'Souza		OPEN	Unresolved	Sep 16, 2013	Mar 06, 2018	
	FLUID-4731	the grunt build should produce a file specifying the build settings and information	Unassigned	Justin Obara		REOPENED	Unresolved	Jul 19, 2012	Jan 29, 2019	
	FLUID-4042	Feature our cool little helper snippets on the demo (scrolling table, aria labeller, etc)	Unassigned	heidi valles		OPEN	Unresolved	Jan 21, 2011	Feb 27, 2017	

	FLUID-4019	Image Reorderer container should not be restricted to just Form elements	Unassigned	Jonathan Hung		OPEN	Unresolved	Jan 05, 2011	Aug 12, 2016
	FLUID-3697	Undo hard-codes selector classes instead of using user-configured values	Unassigned	Anastasia Cheetham		OPEN	Unresolved	Aug 30, 2010	Mar 01, 2016
	FLUID-3344	The jQuery selectbox plugin used by the dropdown Inline Edit doesn't support noConflicts() mode	Unassigned	Colin Clark		OPEN	Unresolved	Nov 01, 2009	Mar 10, 2017
	FLUID-2797	The prev/next link moves as user clicks previous	Unassigned	Daphne Ogle		REOPENED	Unresolved	May 28, 2009	Nov 29, 2018
	FLUID-2779	Cursor over current page number and disabled prev /next links in Pager(renderer version) should be an arrow, not the hand	Unassigned	Yura Zenevich		REOPENED	Unresolved	May 27, 2009	Mar 23, 2019
	FLUID-2514	Custom build package is incorrect when trying to include some things from a directory and exclude others from the same directory.	Unassigned	Michelle D'Souza		OPEN	Unresolved	Apr 04, 2009	Jan 16, 2017
	FLUID-2512	[Uploader] the currentBatch. totalBytesUploaded needs to be broken into two new model values	Eli Cochran	Eli Cochran		OPEN	Unresolved	Apr 03, 2009	Jan 16, 2017
	FLUID-2457	[Progress] Progress could use a few events for integrators to be able to use in integrations	Eli Cochran	Eli Cochran		OPEN	Unresolved	Apr 01, 2009	Nov 05, 2014
	FLUID-1972	Portlet avatar containers have the incorect size: using Safari	Unassigned	Justin Obara		OPEN	Unresolved	Dec 13, 2008	Jan 16, 2017
	FLUID-1868	More markup flexibility for buttons	Unassigned	Jacob Farber		OPEN	Unresolved	Nov 28, 2008	Jan 16, 2017
	FLOE-239	Add captions to the Types of Forces EPUB exemplar video	Jonathan Hung	Jonathan Hung		REOPENED	Unresolved	Sep 04, 2014	Feb 09, 2018

14 issues