ARTG5330 Visualization Technologies

Assignment 1
Due: 5:00PM Wednesday October 1

Description

Many of the data visualizations you will build during and after this course can be thought of conceptually as “data browsers”—devices that people use to navigate through, and discover insights from, large data sets.

The quote above, from the seminal essay on visual information-seeking by Ben Shneiderman, describes the way in which a skilled information seeker might engage with a large data set. It also broadly outlines the functions that a data browser should have.

Let’s look at an example in practice: http://www.nytimes.com/interactive/2013/10/02/us/uninsured-americans-map.html.

This visualization from the New York Times on the distribution of Americans without health insurance broadly fits the specifications outlined by Shneiderman.

With this assignment, you will use HTML, CSS and JavaScript to build the wireframe structure for a similar data browser, which will be your first step towards building a full-fledged interactive visualization later on in the term.

Objectives

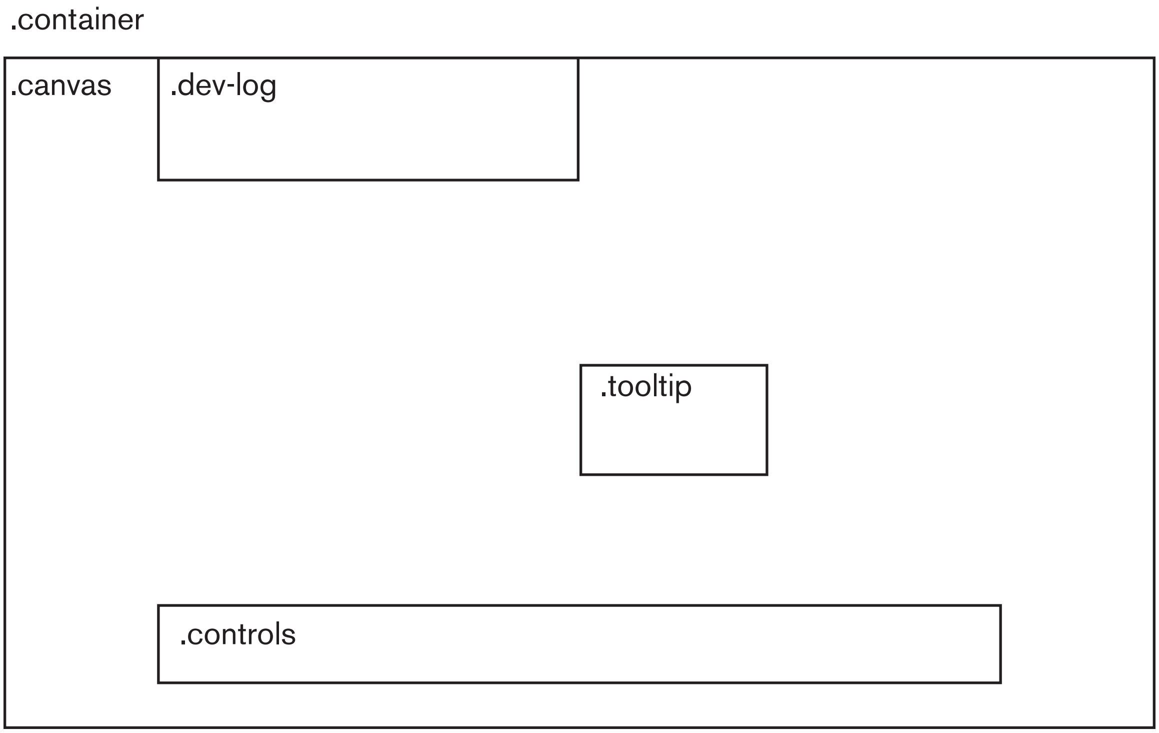
- Become proficient in writing HTML and CSS documents. In particular, become familiar with using CSS selectors in conjunction with HTML;
- Become proficient in using key CSS properties: position, size, font, background color;
- Familiarize yourself with the use of interactive HTML elements: anchor, button, form, text inputs;
- Learn to handle user interactions with jQuery, and become proficient with the “event-callback” pattern.

Task 1: Downloading Starting Code

Download the directory called “Assignment 1” from Blackboard, and observe the folder structure. You’ll see that a skeletal HTML file is included, as are a blank CSS file and a “script” folder.Rules and Regulation

Task 2: CSS Size and Position

A data browser can come in many shapes and sizes, but for the purpose of this assignment, we’ll develop a full-screen layout. See the diagram below.



The class names of the <div> tags will give you a hint as to their role within the data browser. For example, <div class=”canvas”> will be the part displaying the actual interactive visualization, while <div class=”controls”> will be the “control panel” where users can choose to filter or query the dataset.

During development, it’s also a good idea to include a development log that tracks key variable values and user events. This allows you to observe whether your program is behaving as expected. Often, this is taken care of by using console.log(), but for the purpose of this assignment, you should log everything to <div class=”dev-log”>, and position it somewhere visible on the page.

Task 3: Including Interactive HTML Elements

Now let’s turn to <div class=”controls”>, our control panel. This part of the data browser should capture user inputs to filter and query data. To do that, we can rely on interactive HTML elements: anchors, buttons, and forms.

Include the HTML for each of the following in <div class=”controls”>; you can assign any arbitrary value to them at this stage. Don’t worry about their behaviors for now:

- Buttons: use the anchor <a> tag;
- Form elements: include one instance each of 1) radio buttons 2) checkboxes 3) text input 4) select.

Don’t forget to include appropriate id’s and class names to these elements. They will help jQuery to access them later.

Task 4: Capturing and Logging User Inputs

Make sure you've included jQuery into your HTML document.

We need to make sure that user inputs, entered via the control panel, are accurately captured. To do that, we will use the .on() method to attach "event listeners" to each of our interactive elements. Please consult the documentation here (http://api.jquery.com/on/).

The hard part is to determine which events you should "listen for" that are appropriate to the element in question. For example, the user event on the anchor tag to listen for is "click", while for radio buttons, it is "change". To get a better sense of the range of user events one can listen for, consult this page:

http://api.jquery.com/category/events/form-events/

Finally, in the event callbacks, log all the values associated with the user inputs to <div class="dev-log">, using the .html() jQuery method.

Task 5: Submission

You can submit this assignment in one of two ways:

- Commit and upload your directory to Github, and send me the url to your repository;
- Send the entire project, as a zipped folder, to me by email.

Either way, the email subject line should be "**ARTG5330 Assignment 1" + your last name**.