CMSI 370-01

INTERACTION DESIGN

Fall 2012

Assignment 1204 Feedback

Carlos Agudo

1e — Your design choices show a very good understanding of affordances. The look and feel of your selectable items matches well, in my view, with their functionality. (+)

3a — You clearly demonstrate user interface construction knowledge from the perspective of standalone web pages. What is specifically missing in this assignment, with regard to user interface construction, is the reusability aspect of being a "widget." Your basic HTML structure is good (although not optimal—see below), but your JavaScript code is not reusable at all—i.e., it has to be *copied* from one page to another, for every page that needs to use your interface. This is what you need to restructure: first, your JavaScript code must be a separate *.js* file that will not need to be copied and modified from page to page. Thus, you need to *parameterize* the parts in that code that do change—i.e., the array of sites that get associated with the `li` elements in your HTML. The code that is specific to the page thus just turns into a function call with that array, like so:

```
convertToTabs($(".projectTab"), $("#proExp"), arrayOfSites);
```

...or something like that. Second thing: one of the goals of reusable code is to minimize the code that the developer has to generate as much as possible. In your case, the `selectable` and `projectTab` classes should not have to be placed in the HTML; your widget code should do that, because clearly anything inside your menu class should fit a particular pattern. Thus, the developer should only supply the `ol` and `li`s, but should not have to worry about what those classes should be.

Anyway, those characteristics are important aspects of being a reusable widget. Other examples you can spot include Bootstrap and jQuery UI. (|)

3b — Your event handling knowledge is evident here. (+)

3c — The very inlining of JavaScript in your HTML code detracts from demonstrating a solid understanding of MVC. You probably feel that there was no other way to do this, but believe me, there is. (|)

3d — The nature of your widget somewhat precludes low-level event handling. You should not be penalized for this, of course, but at the same time most of your demonstration of this proficiency rests in the direct manipulation assignment. (|)

4a — Your code is functional and does what is intended; its main flaw is its design, and that is addressed elsewhere. (+)

4b — Proper widget structuring and reusability was not achieved in this version as already mentioned, and that does detract from your separation of concerns and MVC. (|)

4c — Your code is generally readable and easy to understand. Your spacing is not perfectly consistent (e.g., you should place blank lines between CSS blocks; there are others that I have mentioned before, and which you should be able to glean from the sample code), and that is a bit of a distraction. (|)

4d — You made good use of resources and external documentation to do this assignment, except for how to do the widget aspect well. (+)

4e — Your commit phasing and messages (including when this code was in the *project* directory) are both appropriate for the work done. (+)

4f — Submitted on time, with some additional material after the due date. (+)