CMSI 370-01
Interaction Design
Fall 2015

Assignment 1029 (due 1103) Feedback

Note that, as a condition for the due date extension, you were still expected to commit something by 1029. This will factor into your *4f* proficiency.

Nicholas Soffa

SoffaKing88 / zaramath88@gmail.com

Notes while running (asterisks indicate major observations):

- Fun initial page state—but the color choice of dark gray over black isn't very readable. Why was this choice made? (*3a*)
- Because your page front loads a couple of network connections right away, putting some immediate, visible feedback that loading is in progress would be a good idea. Things are decently fast when I tested it, but it won't always be like that for everyone and so in general, loading feedback is a good idea. (*3a*, *3b*, *4d*)
- Actually that feedback idea applies across the board—they would be good for the searches too, in case some kind of unforeseen delay occurs. (*3a*, *3b*, *4d*)
- *** The click-on-image function is straightforward, but it causes a disconnect when I am on any other state than the two Top 10 searches—i.e., when I click on an image then go back, I lose the prior state. Better to just display the image in the same page (say a modal) rather than clicking out. (*3a*, *3b*, *4a*)
- Although you have five distinct searches going on, the overall flow of what each search *does* is actually the same: issue a request, navigate the results, then present them on the page. In terms of code, I expect this to be very compact. We will see what happens when I look at the source. (*4b*)
- The image display is certainly the most interesting part of using the web service, but upon looking at the data you get for each image, I think there is a lost opportunity here. You get *so* much more information about each image. I would say that an "image profile" display is fair to ask for, given that code-wise, your front end ultimately does just one thing, with the web service providing the differentiation. This would actually go along nicely with the earlier thought of having the single-image display occur within the page so as not to lose prior state: a combo image display + image "info card" would go well together. (*3a*, *3b*, *4a*, *4d*)
- For the relative simplicity of the search interfaces, I think their layouts could have been more refined. One seldom sees text fields with buttons hanging directly beneath them without vertical space. How do most search fields look? They have the search button on the right, spaced nicely. In fact Bootstrap has a set of classes especially meant for that, called `form-inline`. A quick scan through their documentation would have revealed that, and with little extra effort you would have had a layout that is closer to what the "pros" have online. (*3a*, *4d*)
- *** The general Imgur search does not properly report an issue if the user tries to search with an empty text field. For that matter, the Subreddit search, while it does work (defaulting to the top popular images), should probably still "notice" if the search field is empty then clarify to the user how it is handling such a search. (*3a*, *3b*, *4a*, *4d*)

Code review:

1. *** You indented with spaces on CSS, but not JavaScript and HTML. (*4c*)
2. *** You already have a custom *format.css* file. Why is this `style` element not there instead? (*4b*)
3. Bootstrap does not have a `col` class, nor do you have one in *format.css*, so I suspect this is a typo or a mistake. (*3a*, *4a*)

CMSI 370-01

Interaction Design

Fall 2015

Assignment 1029 (due 1103) Feedback

Note that, as a condition for the due date extension, you were still expected to commit something by 1029. This will factor into your *4f* proficiency.

4. Here is an example for why we indent with spaces and not tabs: the mix of spaces and tabs on this line makes the indentation look wrong on many editors. Maybe not yours, but for others, it will look like you indented this wrong. (*4c*)
5. *** Avoid inline `style` attributes—these should be placed in a separate CSS file to respect MVC separation of concerns. You may need to assign additional classes so the styles get assigned to just the right element(s). Or, scan the Bootstrap documentation for built-in classes that you may find useful. (*4b*)
6. The Bootstrap `form-group` class is not meant to accommodate both an `input` element and a `button` (as you can see from their examples). You should either place the `button` in a `form-group` of its own, or you can "attach" the `button` to the `input` field (see the Bootstrap docs; they support that directly). (*3a*, *4d*)
7. Now, *this* line is just flat-out indented wrong, tabs or not. (*4c*)
8. Unless adjacent to same-sided parentheses, have a space before and after braces. (*4c*)
9. You will probably use this value a lot…and thus you will not want to *repeat* it a lot. (*4b*)
10. Semicolons are indeed optional in JavaScript, but stay consistent. (*4c*)
11. Space after `for`, `if`, and most other reserved words please. (*4c*)
12. *** This is missing the `var` reserved word, and that makes this variable global. Not good. (*4b*)
13. Recommendation: add a blank line between major blocks of code. (*4c*)
14. For function definitions, place a space between `function` and the argument parenthetical. Think of it as a function statement, but without the name in between…there's still a space there, right? (*4c*)
15. Nice, you discovered `wrap`. Useful in many situations, like this one **:)** +(*4d*)
16. This is redundantly copied (as anticipated in note #9)—better to stick this in a variable. (*4b*)
17. *** Actually, beyond just sticking the client ID in a variable, this entire `done` function is nearly identical to the one for popular images, differing only by the selector used. This is *not* DRY code—you should have noticed that they are the same (and you can't say that you didn't notice, because you most likely copied this code from earlier), and taken steps to consolidate the repeated code. Parts that differ become parameters. Get into this habit sooner rather than later; it will save you a *lot* of pain as you start maintaining longer-lived code. (*4b*)
18. *** Also, because this code is virtually identical to the earlier version (lines 9–26), please observe that **all of the notes** from the prior code apply here also. It's just too time-consuming to reapply the comments to the copied version. But they're all there **:)** (all outcomes from applicable notes)
19. Note that in *this* copy, you have a call to `empty`. On the other hand, calling `empty` for the popular and meme image searches would have had no effect anyway, so it is fine to include this in the consolidated function. (*4b*)
20. Now, to give you a concrete example for why striving for DRY code *proactively* is not just a convenience but an actual *necessary practice* in software development, think back to one of the runtime bullets where I noted that you should change the "click-an-image" behavior to be an in-page modal, with an accompanying profile. If you decided to implement this function, *you would need to change your code in 5 different places* to get the feature to work with all of your searches. If you instead have a consolidated `displayImageResults` function (or something similar), you would only have to change the code *once*, inside that function,

CMSI 370-01

INTERACTION DESIGN

Fall 2015

Assignment 1029 (due 1103) Feedback

Note that, as a condition for the due date extension, you were still expected to commit something by 1029. This will factor into your *4f* proficiency.

> and automatically all of the searches will work with the new feature. *That* is one of the reasons why we pursue DRY code aggressively. (*4a*, *4b*)

3a — | …Image presentation is reasonably clean, but color choice is not the most readable and the search sections could be laid out much better (as simple as they already are).

3b — / …Events are generally handled well, except in the case of search errors and in the design choice of what happens when clicking on an image. Because one of the key points of effective interaction design is the minimization of errors, this weighs quite heavily. That, plus the loss of state (mental model disconnect) when clicking on an image then going back, takes this down.

4a — | …Although five distinct image *searches* are happening, the core functionality of how those searches are *handled* is largely the same. This should have left room for additional leveraging of the API, primary of which is the display of additional information for each image. The click-on-image behavior partially counts as a functionality issue as well.

4b — – …The biggest weight here is the not once- nor twice- nor thrice- but *five*-fold repetition of essentially the same code for displaying image search results. That alone would bring the proficiency to a /, but additional separation of concerns issues like the mixed-in `style` elements and attributes in HTML take it down further.

4c — / …Formatting with tabs is a key issue, taking things down a notch right away. Add to that the indentation issues and inconsistent spacing/choices, and we reach this proficiency.

4d — | …Generally good job with the API, but more exploration of Bootstrap would have yielded additional ideas in how to present/layout your front end.

4e — Decent number of commits here, with descriptive messages. If only you started earlier than after the initial due date… (an appropriate start-commit not only speaks to punctuality but your ability to pace your work over the duration of a project) (|)

4f — The express condition of the extension to 1103 was that you show non-trivial work being done by 1029. Unfortunately this did not happen (first commit 1101). Because it was a specific condition, it must weigh pretty significantly here. (/)