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.

Vic Frolov

VicFrolov / vicfrolov@gmail.com

Notes while running (asterisks indicate major observations):

- Nicely done, very cohesive and well-executed! +(*3a*, *3b*, *4a*, *4d*)
- Bootstrap is used well overall, and nicely integrated with your own custom styles. +(*3a*, *4a*, *4d*)
- Nothing much else to say, really…pretty much operates as advertised. +(*3a*, *3b*, *4a*, *4d*)
- One thing we have discussed before is the way results are presented—that can be spruced up. With the [almost] plain text display, it is hard to distinguish that your category search returns *users*, but the trending search is for *hashtags* or *topics*. (well, having the button for trending tweets labeled *Find Users* is also a cause for confusion!) (*3a*, *4d*)
- In the location-based tweets, an explicit location should be mutually exclusive with "Use Current Location," right? The front-end allows them both, leading to potential confusion on what will be used.
- The focus indicator used in the tweet search text fields is red…sends the wrong non-verbal signal to me because it makes me think there is an error. (*3a*)
- The location search results are presented better, looking more like actual tweets—more of that please **:)** However your app (understandably) loses state when the user clicks out to see a profile or a tweet, then navigates back. There's more than one way to address this issue; we can talk about it sometime if you like. (*3a*, *3b*, *4d*)
- The Google Map integration is fun and of course works well for tweet display, but it also makes me feel like I should be able to set a location using that map. See if there's a way to send the current location in the map out into your app. (*3a*, *3b*, *4d*)
- I think you've mentioned this to me before—some coordination between the map pins and the tweet list would also make sense. (*3a*, *3b*, *4d*)
- Some kind of "clear" button would probably be good, for easy starting-over. (*3a*, *4a*, *4d*)
- For the location field, I frequently make the mistake of *not* clicking an actual dropped-down location, and the result is that I sometimes get "no tweets." (e.g., "Culver City, CA, United States" works, but "Culver City" alone does not) Might there be a way to help the user so that partial results are still set to some kind of valid location? (*3a*, *3b*, *4a*, *4d*)
- Successive tweet searches appear to accumulate results. This can go either way for me…but maybe, at least, provide an option of clearing previous search pins or keeping them. (*3a*, *4a*)

Code review:

1. You indented with spaces on HTML, but not CSS. Some tabs also found their way into your JavaScript, in the *tweetDistrict.js* file. (*4c*)
2. At 2 spaces per indent, you can afford to indent `head` and `body`. (*4c*)
3. Don't allow lines to get excessively long. A good maximum line length these days is 120 characters. Get your editor to help you here; many of them let you set such a limit. (*4c*)
4. I don't see a need for multiple `containers` at this point. I think you can have one overall `container` with a bunch of `rows` in there. At least that is how I interpret the Bootstrap documentation. (*3a*, *4d*)

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.

5. Watch what you copy-paste—I think `basic-addon1` is a dangling reference. (*3a*, *4d*)
6. Instead of comments separating major sections, why not blank lines? At least that will not go obsolete, and they clutter your code less. (*4c*)
7. They have something for numbered lists—it's the `ol` element. (*3a*, *4d*)
8. One helpful rule with code spacing is to "space like you're being proofread"—i.e., unless syntactically meaningful, apply whitespace the way you would to natural language content. That means adding space with most punctuation. There are exceptions, but this is a nice starting point. (*4c*)
9. CSS is more readable if you add a space after the colons (to separate property names from their assigned values). (*4c*)
10. Commented-out code…does not belong in code. Use version control if you don't want to lose it. (*4c*)
11. Keep opening braces on the same line; add blank lines between CSS blocks. (*4c*)
12. *** These are all globals! Modularization is good, but you should encapsulate also. At a minimum, enclose a module within a single JavaScript object (the way jQuery and the Google JavaScript APIs do—notice `google.maps.Map`). For greater flexibility look up the *requires* JavaScript library. Or you can implement this as a jQuery plugin. (*4b*, *4d*)
13. The definition of functions as first-class objects that are then assigned to variables or attributes is preferred, because that expresses the semantics of JavaScript functions more faithfully. (*4b*)
14. Assigning directly to `onclick` is obsolete; use jQuery or `addEventListener`. (*4b*, *4d*)
15. Semicolons are indeed optional in JavaScript, but stay consistent. (*4c*)
16. 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*)
17. Concatenated HTML code is functional, but not always the easiest to maintain. Look into a combination of jQuery's `clone`, `find`, and `text` methods to give your code more structure. (*4b*, *4d*)
18. This function should have encompassed *all* of the code in this file. (*4b*)
19. You're serving both the site resources and the service relay from the same endpoint, so all you need here is an absolute URL to the same server, such as `/tw` or `/funnytw`. (*3a*, *4a*)
20. In JavaScript, triple equals (`===`) is the appropriate equality operator. Double-equals is not sufficiently strict and is ultimately a language design flaw. (*4a*, *4c*)
21. Make use of JavaScript truthiness/falsiness—avoid this comparison, and if this is somehow needed specifically, explain it in a comment. (*4a*)
22. You've seen the cooler way to do this. (*3a*, *4a*)
23. Your dropdown-driven searches shouldn't be enabled if a proper choice is not selected in the dropdown menu. This is basic menus-forms-dialog interaction style behavior.

3a — **+** …For a first go-round, this is nicely done, combining standard Bootstrap with your own visual ideas for your front-end.

3b — **+** …Event-handling is generally decent, with some cute coordination between some elements.

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.

4a — **+** …Functionally quite solid here

4b — **/** …The leaking globals alone drop this one pretty hard. Add up the smaller areas for improvement and you get this.

4c — **|** …First: tabs. Second: a variety of spacing, linebreaking, and other code presentation glitches.

4d — **+** …Good job figuring out multiple APIs and Bootstrap. Some more to learn, but for the scope of the assignment you got a lot.

4e — Nothing much to say; keep version controlling like you have been **:)** (**+**)

4f — Started before 1029, submitted on time. (**+**)