

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.

Josh Kuroda

jkealii / jkealii@gmail.com

Notes while running (asterisks indicate major observations):

- *** I'll have to see what your code tries to do, but, upon running, things seem kind of...bare. The requested five functionalities are somewhat there, but the labeling (you actually include "assignment" in the content?), information display ("Allowed scope: user_read,channel_read" looks more like a debug statement and not presented information) and layout (avatar display; logout button) can be better. (3a)
- Case in point: I used an account that doesn't have an avatar. Your front end can do better than display a broken image. (3a, 4a)

Code review:

1. Thank you for using spaces to indent all of your code! +(4c)
2. Your file naming scheme is not consistent: you diverge from *frendass* with the JavaScript filename, and that divergence is actually redundant: we don't need "*frendjsas*" when the file extension is already *.js*! (4c)
3. There is actually supposed to be just one container. What you have isn't necessarily wrong, but it uses a lot more code than necessary. (4c, 4d)
4. Oops missed indent. (4c)
5. Overall, except for the container excessiveness, you make good use of built-in Bootstrap classes, in particular some that were not covered in class. Your own custom classes look well-chosen, too. +(3a, 4d)
6. Oops, I take note #5 partially back (just a little bit)...you don't need a *hidden* class because Bootstrap already defines it for you! (4d)
7. Not your usual visual for a hovered-over button, but I appreciate the experimentation and the choice generally works for this specific user interface. +(3a)
8. This *might* be necessary based on what the Twitch API requires, but I will say that it raises a red flag for me. At the very least, it is probably best to explain this choice, because in general anything short of libraries or predetermined access points in the top-level scope is not a good practice. (4b)
9. 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)
10. Unless adjacent to parentheses, have a space before and after braces. (4c)
11. Display-wise, you can do better than a plain string concatenation. (3a, 4a)
12. Semicolons are indeed optional in JavaScript, but stay consistent. (4c)
13. *** I looked at what this API call returns...ack, so much more information to display! You could have really had a field day here with information presentation. Lost opportunity. (3a, 4a, 4d)
14. OK, now *this* is a bit of a cop-out—you're merely pulling out another small bit of information from the same API call as before! I'm not particularly opposed to the API call reuse—sometimes it is justified (see note #15)—but here, for the amount of information given by the API and the small *fraction* that is displayed by the front end...there is simply major unused potential here. (3a, 4a, 4d)
15. *** A few issues here: first, already observed at run time, this code does not accommodate the possibility that `user.logo` does not exist. Second, this code unconditionally appends a new `img` element whenever the

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.

button is clicked, resulting in multiple images. Not exactly a mental model match. (note that I don't have an issue with the reused user method here) (*3a, 3b, 4a*)

16. Surround binary operators with spaces for readability (compare this to line 8). (*4c*)

3a — | ...Good Bootstrap library use shown, with some visually appealing layout choices. But given the amount of data available, the display could have been much richer.

3b — | ...Event handling is generally well done here, but the loopholes in the user logo display put a damper on this proficiency.

4a — / ...The front end largely works here (except for user logo), but what it actually does is a somewhat low bar given the additional information provided by the API.

4b — + ...No real problems with separation of concerns or MVC here; benefit of the doubt provided on the one *4b*-related code issue, note #8.

4c — + ...Overall, the code is well-presented and highly readable. There are some points of improvement as seen in the code review, but overall these are relatively minor and are at least mostly consistent.

4d — | ...HTML/CSS/JavaScript/jQuery/Bootstrap learning beyond what was shown in class is pretty decent. The hiccup is in the general underutilization of the Twitch API.

4e — Nicely paced, spread-out commits, with descriptive (and candid!) messages. I think for a program like this, you could have been more fine-grained, but what you have is good as a starter. (+)

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