

CMSI 370-01

INTERACTION DESIGN

Fall 2014

Assignment I030 Feedback

This feedback is effectively a code review; to keep it concise, notes are provided as a numbered list. Inline comments in your code illustrate an example of the numbered item (there may be more). Asterisks (*) denote high-priority issues. If any code review item is unclear or if the resulting proficiencies are not self-explanatory, please ask me.

Rachel Rivera

rachelriv / rachelriv94@gmail.com

1. Text fields used for gender rather than radio buttons or dropdown menu (3a, 4a)
2. You can search—nice bonus! (3a, 3b, 4a)
3. Interesting choice to supply only what needs to be changed for edits (3a, 3b, 4a)
4. Fun help page, though standalone only (i.e., no tooltips, contextual help, etc.) (3a, 4a)
5. Dangling closing tag (as far as I can tell) (4a)
6. This is actually more for the next assignment, but since it's already there I might as well point it out—your d-pad is functionally a “third-party” library because of its generality, and so its code should be loaded before the app-specific items, i.e., *rpg.js* and *search.js* (4a)
7. I'm pretty sure you already know this, but this modal markup is (understandably) repeated in all files, and as such would be a target for templaticization or server-side inclusion if the backend were switched to a real web app server (4b)
8. For readability, maintain blank lines between distinct code blocks; let minifiers take care of tightening this up for deployment (4c)
9. The `keyup` event has been superseded by `input` for implementing type-as-you-go interaction (`input` handles things like mouse-based copy/paste, which `keyup` does not)—you can use jQuery's `bind` method to hook up to this (3b, 4a)
10. Keep your spacing consistent—the only time parenthesized expressions [in JavaScript] are *not* preceded by a space is in a function call; the only time they are *not succeeded* by a space is right before a semicolon or a right parenthesis (due to nesting) (4c)
11. Don't conflate expressions with statements—statements within expressions may obscure their side effects (4b)
12. Enclosing in `$` already sets the function for calling on load (4a)
13. When the event handler is a single argument-less function call, just make the *function itself* be the event handler (4a, 4c)
14. Something to look up—I've heard some debate that `Object.keys()` with an iterator function is preferred over `for (key in object)` these days...not sure what the latest opinion is (4a)
15. Missing `var`—this goes to top-level scope (4b)
16. One-time use variable used solely for a `return`...just return the expression (4c)
17. Very similar functions differing only by an array of selectors—can be consolidated further into a single function (4b)
18. Potential consolidation: collapse repetitive statements into an iteration (not necessarily a negative because the design might not always call for it; more a point for future consideration) (4b)

3a — +

3b — +

CMSI 370-01
INTERACTION DESIGN
Fall 2014

Assignment I030 Feedback

This feedback is effectively a code review; to keep it concise, notes are provided as a numbered list. Inline comments in your code illustrate an example of the numbered item (there may be more). Asterisks (*) denote high-priority issues. If any code review item is unclear or if the resulting proficiencies are not self-explanatory, please ask me.

4a — +

4b — +

4c — +

4d — +

4e — +

4f — +