CMSI 370-01

Interaction Design

Fall 2014

Assignment 1030 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.

Matthew Flickner

mwflickner / mwflickner@gmail.com

1. No published version seen (*4a*)
2. Item spawn shows no visible display (*3a*, *4a*)
3. Character creation, editing, and deletion require manual refresh (*3a*, *3b*, *4a*)
4. Edit character doesn't auto-fill data (*3a*, *4a*)
5. Odd choice—confirmation modal appears for character edit? (*4a*)
6. Rudimentary help tab (*3a*, *4a*)
7. Obsolete event handler assignment: do this completely in JavaScript (*3b*, *4b*)
8. Incorrect/inconsistent indentation (*4c*)
9. Tabs in the source code (*4c*)
10. Non-descriptive names; some are copy-paste holdovers (*4c*)
11. Unnecessary comment blocks (*4c*, *4e*)
12. Inadequate/inconsistent spacing: space before braces; space after most punctuation (commas, semicolons, colons, etc.); separate function arguments; etc. (*4c*)
13. Duplicate HTML element ID (*4a*, *4c*)
14. Similar markup that might be worth consolidating—might be better to use the same markup and have JavaScript set create-vs.-edit behavior (*3a*, *4b*)
15. Incorrect script-loading order—make sure that you load scripts in order of dependency (i.e., your code depends on Bootstrap with depends on jQuery, so you load jQuery first, then Bootstrap, then your code—doesn't always break things but is safer this way) (*4a*)
16. Separate distinct blocks of code with blank lines (*4c*)
17. Code in top-level scope—except for reusable libraries, you very rarely ever need this; certainly not in the case of what you declare here (*4b*)
18. Function statements rather than declaration as variables—declaring functions using `var` notation carries much clearer scoping rules and speaks to the first-class nature of functions in JavaScript (*4c*)
19. Multiple jQuery function calls on the same selection: chain them (*4a*, *4c*)
20. Start `else` clauses on the same line as the preceding closing `}` (*4c*)
21. Superfluous empty string concatenation (*4a*, *4c*)
22. Potential consolidation: collapse repetitive statements into an iteration (*4b*)
23. Editing and creating code is *very* similar—strongly consider unifying into a single function with parameters (*4a*, *4b*)

3a — |

3b — / …virtually all client-side event-handling is via inline JavaScript strings, which is not the way to do it these days—instead, use event listener functions (e.g., `addEventListener`, jQuery's `on` or `bind`)

CMSI 370-01

Interaction Design

Fall 2014

Assignment 1030 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 — +