CMSI 370-01
INTERACTION DESIGN
Fall 2014

Assignment 1204b 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.

Daniel Levine   *dannymlevine / dannymlevine1@gmail.com*

1. The idea of a switch with a threshold is fine, as we have discussed, but you really should still preserve the "switch-ness" of the control: beyond 50 should snap to the right, and less than that goes to the left. I think you missed that the direct manipulation part is just the interaction; the final value still remains as one of two possibilities (well, generalizable to more than that but let's not get ahead of ourselves here). (*2b*, *3a*)

2. Meanwhile, those filenames provide no notion at all of what your widget does! (*4c*)

3. …and upon looking at your code—ack, it isn't even a jQuery plugin! Plus the slider just uses the built-in `range input` element! This totally misses the point of the assignment. (*3a*, *3b*, *4a*)

4. The absence of the plugin architecture is pretty much the deal breaker—this is what prevents us from customizing behavior in a way that is safely encapsulated. This was a matter of following the procedure for defining a jQuery plugin (*http://learn.jquery.com/plugins/basic-plugin-creation/*) (it was also done live in class sometime in mid-November). (*3a*, *4a*, *4d*)

5. With the plugin, we could have even salvaged the use of a `range input` element. It still wouldn't have been as appropriate as a completely manual widget (note the "from scratch" part of this assignment's name), but the custom behaviors of making it act like a switch (i.e., snapping it from one end to another depending on how far the user drags it) can still be delivered from within the plugin. In fact the plugin could have created the element itself, so you aren't relying on the developer's markup (meaning there may be programming errors). (*3a*, *3b*)

6. Of course, the ideal scenario would have been to build everything yourself: generic elements with CSS and event handlers that reproduce the behavior of a sliding switch—analogous to how our boxes code turned generic `divs` into draggable, well, *boxes*. (*2b*, *3b*, *4a*)

7. On a separate note…even if it were right, that *widget-from-scratch.js* code retains a lot of the code presentation issues noted in prior feedback to prior assignments. There is an erroneous code formatting choice on virtually every line. Definitely a habit that you need to break. (*4c*)

2b — **/**

3a — **/**

3b — **/** …Some event coordination seen here, but without functionality we cannot give much.

4a — **/**

4b — **/**

4c — **–**

4d — **–** …This is ultimately the root of the issues—you were unable to find the key information that would have gotten you started.

4e — **|** …The unfortunate part is that you actually started out in the right direction, building it from scratch (knob and circle, we hardly knew ye **:(**).

4f — **/** …Still, the on-time commit remained incomplete, with more code coming in later days.