

**CMSI 370-01**  
**INTERACTION DESIGN**  
Fall 2012

## **Assignment I204 Feedback**

**Lisa Rosenbaum**

*1e* — Your letter-chooser is executed quite well visually, but your event-handling choice goes against the affordance communicated by these visuals. Your implementation uses a single click, allowing the knob to move in a single direction—but the knob presentation does not afford this. Instead, the knob “wants” to be turned/dragged, and bidirectionally also. This is quite contrary to the actual behavior of your widget. (/)

*3a* — You clearly demonstrate user interface construction knowledge from the perspective of standalone web pages. What is specifically missing in this assignment is the reusability aspect of being a “widget.” An easy way to think about reusability is to ask yourself: if you need *more than one* middle-initial knob selector on your web page, do you need to modify your code beyond just adding another HTML element to convert into a middle-initial knob selector? As written, the answer in your submission is yes; good componentization would answer no (see, for example, the various Bootstrap components—one JavaScript file, any number of dropdowns, search fields, menus, etc.). (|)

*3b* — You show an understanding of event-driven programming in your widget, but the choice of a click handler all by itself shows only a basic understanding. This widget, especially with the affordance that it communicates, demands more than that basic understanding. (|)

*3c* — Your cleanly-separated HTML, CSS, and JavaScript files show good understanding of MVC. (+)

*3d* — Your chosen widget behavior of responding only to click events somewhat precludes low-level event handling. Typically, this is not automatically penalized, but due to the mismatch of this behavior choice with the visual choice, it has a definite negative impact for this particular widget. Still, the handler itself shows a decent amount of low-level, detailed behavior (which could be coded better though—see *4a*), so that mitigates things somewhat. (|)

*4a* — Overall your code appears to work as you intended, although, as mentioned, it is not in a fully reusable widgetized form. In addition, the code itself shows noticeable room for “under the hood” improvements (see the inline comments for details). These keep the proficiency from going over the top. (|)

*4b* — Proper widget structuring and reusability was not achieved in this version as already mentioned, and that does detract from your separation of concerns outside of pure MVC. (|)

*4c* — Your code is decently readable, although some of your expressions are tighter than I would have liked, and the multiple transform properties could have used some inline comment explanation. (I understand why you did it that way, but not everyone who reads your code will) Overall, though, I will count these as somewhat minor, and feel OK with giving maximum proficiency here. (+)

*4d* — Implementation-wise, your code shows good use of available resources and documentation, especially with the thorough, cross-browser implementation of CSS transform properties. The non-widget structure of the code, plus its conflicting affordance-vs.-behavior issue, were not sufficiently figured out, however, despite material pertaining to this being available both in course materials and on the web at large. (|)

*4e* — Outside of your rename commit, I see a single commit for all of your widget code. Clearly, this work can be phased better, for the amount of code that was produced. (/)

*4f* — Submitted 10 days late. (−)