CMSI 370-01
INTERACTION DESIGN
Fall 2012

Assignment 1120 Feedback

Joe Crawley & Terran Moore

(This is based on *boxes.html*, *boxes.css*, and *boxes.js* from Terran's GitHub repository—that looked like the latest version.)

1e — The distinct look you gave to your resize handles is not typical (they are usually square and solid), but remains sufficiently recognizable as a likely resize handle. I take this as a sign of understanding of the affordances concept. However, you are really only handling lower-right corner resizes—but you have a resize handle in all four corners. This detracts from the case of demonstrating an understanding of affordances.

Finally, there is no immediate feedback as the user drags the box outside of the drawing area that letting go of the mouse may trigger box deletion (the way the Mac OS X dock displays a "cloud" when you drag an icon off it). Instead, you wait for the mouse to be released, after which you ask the user if the box should be removed. This interaction design decision does not work for me on two counts: (1) It "breaks" the direct manipulation interaction style by not giving immediate, rapid feedback, instead waiting until the user might have done something they don't want to do then turning to a dialog box for confirmation. (2) If the user *does* cancel, you are left with a box outside of the drawing area—which is kind of an odd state of affairs (original code notwithstanding).

So, all told, I would say you could have made better choices here along the lines of the messages that a user interface says to the user, which is the affordance concept is all about. (**/**)

3a — You show some good advancement in user interface construction here. (**+**)

3b — You also show some good advancement in event-driven programming with your code. (the interaction design decisions might not have been great, but the actual event handling that goes along with them looks just fine) (**+**)

3c — You show good separation of concerns here. (**+**)

3d — As mentioned, the design choice for deletion might not have been great, but the actual implementation at the event-handling level looks fine. For resizing, you almost got it right except for a glitch in the way you track your resize anchor point. Remember that some base values change when the lower-right corner crosses boundaries. That is the main code that you are missing. (**|**)

4a — Aside from the resize glitch, your code generally functions as you intended. In terms of general code quality, you have a couple of questionable hardcodes (noted in the code), but otherwise no other glaring faux pas. You should be beyond avoidable hardcodes at this point though, so that's big. (**|**)

4b — Your code shows good MVC separation. (**+**)

4c — Readability-wise, your code had a lot of distracting miscues—not fitting for junior-level computer science major code! Missed indents, bad spacing choices, overlong lines, commented-out code blocks… what would Dr. Toal say if he saw this source? This is an area for serious improvement. At this stage you shouldn't even be needing to reformat your code after things are done, because your code should be indented and spaced correctly *as you write it*. You are computer science upperclassmen now—your source code should read like it. (**/**)

4d — You solved a good number of event and state handling issues on your own here. (**+**)

4e — I find it concerning that, as Joe said in his commit log, "Github and my computer don't always get along." This should bother you; it should not be a problem that you let lie. Ask a senior; ask me; ask Masao. Imagine yourself on the job, and their version control system "doesn't get along" with your work computer. No excuse. Beyond that, phasing and messages are OK, but you seem to have started late. (**|**)

4f — Submitted on time. (**+**)

CMSI 370-01
INTERACTION DESIGN
Fall 2012

Assignment 1120 Feedback

Joe Crawley & Terran Moore

(updated feedback based on commits as of 12/5/2012; only re-evaluated proficiencies are included—also)

1e — The disconnect between having four resize handles but only one active resize corner (lower-right) still remains, but your design choice for dragging boxes outside of the drawing area is definitely much better. Still not perfect due to the resize display, but definitely an improvement. (**|**)

3d — Resize anchor point glitch is still there, so no change. (**|**)

4a — You have removed the drawing area hardcode, but not the resize box hardcode. Plus, the resize glitch is still there, as mentioned, so this proficiency remains the same. (**|**)

4c — For readability, you have two divergent versions of *boxes.js* (see below). Both still have indentation issues, though admittedly one version does show improvement. *But there shouldn't be more than one version to begin with* (again, see below). Because of this confusing (and infuriating) state of affairs, this proficiency does not change. (**/**)

4e — OK, so last time, you were called out for not getting a handle on *git*, despite numerous available resources for figuring things out. Unfortunately, this resubmit takes that down a notch—I mean, *come on*, you have *two separate and different* copies of the same resubmission? That's not how this is supposed to go down. You're supposed to *share* a repository, with different commits by different users *going to the same place*.

Here's one instance of why this hurts you: when re-evaluating indentation, *I had to look at both versions of boxes.js*, and I found that *they were indented differently!* If you got your version control act together, this would not have happened. (**/**)