

CMSI 370-01

INTERACTION DESIGN

Fall 2015

Assignment 1211 Feedback—Direct Manipulation Widget

Chris Franco

cf7 / spe861@gmail.com

Notes while running (asterisks indicate major observations):

- Demo plays out nicely :) +(2b, 4a)
- The drag “sticks” when the mouse leaves the window—you’ll want to consider that to be an end-drag event as well. (2b, 3b)
- And the plugin integrates very nicely into your front end. This use case will be a good place for refining feedback, and perhaps deciding what should be parameterized as options. But the core functionality (and usability) is there, and that is what we’re after. +(2b, 4a, 4b)

Code review (asterisks indicate major observations):

1. Yay, no tabs :) (+4c)
2. You’ll need to do more tests here. Ideally the user can target the “restructurable” portions at a finer grain, without losing intuitive direct manipulation. (4b)
3. Yes, so your plugin has page-wide scope. This needs some refinement. (4b)
4. No need to indent *this* far. One level at a time is sufficient. (4c)
5. Don’t go with direct CSS—use *classes* for better MVC separation (you’re already assigning them, so this is a matter of adding to your CSS file). (4b)
6. You should load your plugins *before* your main code. It happens to work because your main code is wrapped around \$(...), but ultimately you should still preserve the right order. (3a, 4b)

2b — + ...Still a ways to go for full polish, but for this assignment good enough.

3a — +

3b — +

4a — +

4b — | ...See notes above. Innocuous as-is, but implications are greater later. Better to address them now.

4c — +

4d — +

4e — +

4f — | ...A few tweaks and cleanup after the due date.