

**CMSI 370-01**  
**INTERACTION DESIGN**  
Fall 2015

## Assignment 1211 Feedback—Direct Manipulation Widget

Josh Kuroda

*jkealii / jkealii@gmail.com*

*Notes while running (asterisks indicate major observations):*

- Whoa, that's a watermelon alright...generally works, but can use better feedback. (2b)
- As for integration with the front end, it's functional but you need to deactivate the default browser behavior. (I'm going to guess that you are relying on the caller to set `draggable=false`—instead that should be your plugin) (4a)

*Code review (asterisks indicate major observations):*

1. Yay, no tabs :) (+4c)
2. Noted, but if you used *jQuery's* mouse event functions, that would have been cleaned up for you. (4d)
3. Commented-out code—no no. (4c, 4e)
4. The definition of functions as first-class objects that are then assigned to variables or attributes is preferred, because that expresses the semantics of JavaScript functions more faithfully. (4b)
5. \*\*\*\* Oh no no no no. Look up “eval is evil.” I never showed you this; you *never* should use it. Especially because *this whole class has been about showing you how to manipulate web elements*. It's as if you already forgot about the boxes code and jQuery. (4b)
6. \*\*\*\* These are *global* variables. Another huge no-no. (4b)
7. \*\*\*\* It seems there is also a misunderstanding about how the plugin mechanism works. No, this is not reusable code. Your plugin code is hardcoded to elements whose class is `image`. Instead, you should have used the `this` variable. Look up the jQuery plugin documentation. (3a, 4a, 4b, 4d)
8. \*\*\*\* Why is your plugin file *copied*? I told the class that I would run the web server *above* the directories so that you can use relative URLs, avoiding copied code. (4b)
9. Note that the selector here makes no difference, *because you hardcoded the selector inside the plugin*. This is a fundamental design issue that hampers your plugins reusability. (4b)

2b — | ...Functionality (utility) is there, but needs feedback for better usability.

3a — | ...Was looking OK, until I saw that hardcoded selector.

3b — + ...No problem with actual implementation.

4a — + ...Functionality is also good—plugin itself, demo, front end.

4b — - ...The global variables, copied code (file), and non-reusable plugin invocation by themselves would have taken this to /. The use of `eval` is what drops this further down.

4c — | ...At least the code is fairly clean. Except for the commented-out code. Which was almost as much as the non-commented-out code.

4d — | ...I will guess that you did some research for this code; additional jQuery lookup and learning would have been good too.

4e — + ...Borderline better granularity, but good enough for a bump.

4f — +