CMSI 370

INTERACTION DESIGN

Fall 2010

Assignment 1028

Midterm malaise? Nothing like some low-level event programming to get you out of it **:)**

Not for Submission

By October 21

1. Shore up the material from the last few weeks with the readings from Dr. Toal and my textbook draft.
2. Fork off your own copy of the sample code on GitHub so that you have online examples and some room to experiment.

By October 26

3. Be in the "nearly done except where stuck or have questions" state with the three widgets. This will allow you to ask any final, burning questions during class on that day.

For Submission

Implement, from first principles, web versions of the following user interface components:

- A push button
- A labeled check box
- A menu bar

Each web component must behave as similarly as possible to their desktop equivalents, with one exception. In particular:

- Your web button must allow the user to back out of clicking it prior to letting go of the mouse button, and provide feedback to that effect.
- The check box's "hot" (clickable) area must include its text label.
- Your menu items must provide feedback to show that they were indeed selected (for example, on Mac OS X, chosen menu items flash three times).
- (this is the exception) Your menu bar does *not* need to support more than one level of menus.

Structure your code so that:

- JavaScript functions serve as one-stop "converters" for the components, which are identified by its `id`. For example, you may have a function `MyUI.buttonize(id)` which, upon invocation, makes the web element with the given `id` behave like a standard push button.
- Your HTML does not have `class` or `style` attributes — these should be applied entirely by your JavaScript "converter" functions.
- Your HTML does not define any event handler attributes (e.g., `onclick`, `onmousedown`, etc.).

You may use jQuery, but *not* jQuery UI or any other JavaScript libraries.

How to Turn it In

In light of the continuing Keck lab file server outage, we will continue to use e-mail for submission. It is recommended, however, that you use local version control for your work: since you already have *git* working (right?), you can create a local repository for yourself.

1. Package your code as an HTML5-CSS3-JavaScript file trio, with the HTML containing all three user interface components.
2. Your HTML5 and CSS3 must validate (with the exception of browser-specific, but non-established, CSS3 property names).
3. Your JavaScript should pass JSLint, with *The Good Parts* turned on. *Assume a browser* and *Predefined* (e.g., **$**) may be used when appropriate.
4. This is how I will retrieve and run your work:
   - Open the HTML file and try out the user interface components
   - Have W3C's validators scan your HTML5 and CSS3 files
   - Have JSLint scan your JavaScript file
   - Make comments/mark up your submission