Assignments up to 1204

For this latter half of the semester, I figured it would be more instructive to package the remaining assignments as a single “suite,” so that you can plan out your time more effectively.

Yes, this is it. No other work for the rest of the semester (outside of refining your portfolios).

Outcomes

The programming assignments (1120, 1204) will affect your proficiency measures for outcomes 1e, 3a–3d, and 4a–4f.

The “dream” user interface design assignment will affect your proficiency measures for outcomes 1a, 1e–1f, 2a, 2c, and 4d–4f.

Background Reading

Textbook reading is centered on the direct manipulation interaction style, which would be Shneiderman/Plaisant Chapter 5.

For the programming assignments, the most helpful additional material from the JavaScript textbook will be the case studies in Chapter 9. These case studies demonstrate lower-level event handling with some direct manipulation elements, and includes an overview of the Sierpinski gasket program (one of the options for the direct manipulation exercise).

For Submission: 1120

Direct Manipulation Exercise

This exercise gives you some firsthand experience in implementing direct manipulation. You have three choices, each offering something different depending on your interests or inclination:

• Modify the boxes bazaar code so that it supports resizing and deletion (by dragging a box outside of the designated drawing area). This choice offers experience in traditional mouse-driven rubberbanding and drag-and-drop.

• Modify the boxes-touch bazaar code so that it supports creation and deletion, allowing for more than one action at a time (one per finger). Do this if you have ready access to a device with a multi-touch web browser, and are comfortable with repeated uploads to your my.cs.lmu.edu account.

• Modify the sierpinski-webgl bazaar code so that you can scale/zoom and translate/pan the displayed Sierpinski gasket. This choice gives you a little exposure of 3D graphics programming (a.k.a. CMSI 371).

How to Turn it In

Commit a copy of your modified code under direct-manipulation in your private 370 GitHub repository (i.e., don’t make the changes in place and issue a pull request), and upload your work to my.cs.lmu.edu so that it is available under the path ~username/cmsi370/direct-manipulation.

For Submission: 1129

A “Dream” Interface Design for Headmaster

This is your chance to cut loose—given the back-end functionality of the Headmaster web service, design your idea of a “dream” user interface for that system. Think outside of the box, be creative, mix and match interaction styles—it’s your call.

Some (minimal, I hope) ground rules:

• You may mix and match any existing shipping technology (e.g., multitouch, speech, audio/video, gesture, 3D, accelerometers, gyroscopes, GPS) regardless of current platform

• Prototype or speculative technologies are off-limits (e.g., brain control, holograms, see-through displays, human-like vision or comprehension)

That’s pretty much it. All else is fair game. Your design should include the following:

1. A top-level design or layout
2. At least two usage scenarios
3. Rationale for your design: relevant priorities, mental models, interaction design concepts, guidelines, principles, theories, etc.
4. Usability metric analysis of your design—what are its strong metrics? Weak metrics?
Illustrate things as needed, with diagrams, screen mockups, etc. Don’t forget to cite references.

**How to Turn it In**

Commit your work under *headmaster-dream-design*. As usual, LaTeX is recommended. You may also describe your design as one or more web pages. If you choose this route, commit your files to the repo and upload them to ~username/cmsi370/headmaster-dream-design on my.cs.lmu.edu.

**For Submission: 1204**

**A User Interface Widget from Scratch**

We end by going back to the basics: design and implement a reusable widget for use in web browsers. The point here is to see how low-level event handling (e.g., mouse or keyboard activity) translate into higher-level ones (e.g., selection or change events). You have seen a number of these in Bootstrap—now it’s your turn.

Some ideas, revolving around Headmaster to stay “in theme,” include:
- A GPA selection knob or slider
- A selector or switch for school year terms (i.e., spring, summer, fall)
- An entry field customized for 4-digit years
- A selector or switch for degrees (i.e., BS, BA, MS, MA, etc.)
- A middle-initial (i.e., single-letter) knob or dial
- A phone number entry widget that has actual number buttons in a grid
- A state selector that uses a USA map thumbnail
- A sex selector that uses the Mars/Venus gender symbols (recommendation: implement knob-like action where rotating the symbol switches from one to the other)
- A password field with built-in strength indicator (e.g., changing text or background color; built-in “temperature” bar)

Note that many of these widgets can be done with “native” HTML elements, but the point here is to implement one of them without these built-ins.

The only third-party library that you may use is jQuery. All other code should be yours. You may implement your widget as a jQuery plugin if you are able to figure out how to do that.

If you have a widget idea that is not in this list, check with me to see if it will work.

**How to Turn it In**

Commit your code under *widget-from-scratch*. Provide two distinct pieces:
1. The reusable code for the widget itself (typically CSS and JavaScript)
2. A “demonstration page” that shows more than one instance of your widget in action

Upload your work to ~username/cmsi370/widget-from-scratch on my.cs.lmu.edu.