

CS160 SP06, Interactive Prototype #1 (Group)

Due: Wednesday, March 15, 2006

Goals

The goal of this assignment is to learn how to build prototypes of user interface ideas using an interactive user interface builder. You will revise your user interface ideas and then use interactive tools to build a prototype of the design.

Interface Redesign

Use the results of your low-fi prototype tests to design a revised user interface. Develop new and/or revised scenarios for your tasks by storyboarding your ideas. The tasks that most of you used in the low-fi assignment should be sufficient for this, but some were simple or partial tasks that did not adequately cover your proposed functionality. Make sure to revise those tasks. If you are changing your tasks, make an appointment with us to present your new tasks, design ideas, and storyboards for discussion.

Prototyping

You will use a prototyping tool to create an interactive prototype of your application. Make sure you follow the directions under “Project Tips” on the course web site for installing Visual Studio 2005 and the Mobile SDK. This can be installed on a student laptop, or is available already installed on the machines in 330 Soda. Talk to us if there are other tools you would like to use.

Your prototype should implement the three scenarios that you developed for your tasks. In addition, the design of the prototype should now start to account for the size, resolution, colors, and other attributes of your target platform. For speech UI's, the wording of the scripts should be polished, the grammar should be fixed and your prototype should respond to exactly what the end-user says. You may wish to experiment with the Suede toolkit for developing your app: <http://guir.berkeley.edu/projects/suede/>

The underlying functionality does not have to be fully implemented. For example, applications requiring large databases of information can instead have a sufficient number of hard-coded data points for supporting the three tasks. The phones have wi-fi capability and can be used with Airbears for network access.

You have a short period of time to complete this prototype, so you should focus on showing only what is essential and try to avoid writing code where it is not necessary. You will likely have to make some difficult choices!

Deliverables

1. **Prototype**

Your prototype must be accessible and/or executable by everyone in the class from bspace. It must be accompanied by a README file that describes any installation requirements and operating instructions, including any limitations in the implementation. The prototype binary should be executable without installing any additional software. You should include your source code as well, and a (working) project build file.

2. **Report**

You will submit one copy of a printed report of no more than four pages of text in class. You must also put a copy of the report online on your website.

3. **Presentation**

Two members of your team will present your project in class, including a demo of your prototype – it is much easier to do this with the Emulator rather than trying to show the actual phone screen. Presentations will be very short (~ 10 mins) because of the class size. Practice in advance! You must use PowerPoint slides and make them available for download on your website.

Report

The report should follow this outline with separate sections for the top-level items.

1. **Problem and solution overview (1 paragraph)**

2. **Tasks (1/4 page)**

- 3 representative tasks to test your interface (easy, medium, hard)

3. **Revised interface design (1 page plus screenshots or scripts)**

- Changes as a result of low-fi testing and rationale behind the changes (refer to screenshots or scripts)
- Sketches or scripts for unimplemented portions of the interface
- Storyboards of tasks (annotated screenshots or scripts)

4. **Prototype overview (2 pages)**

- Tools used:
 - How the tools helped
 - How the tools did not help
- Overview of the UI implemented (reference figures or scripts from next section)
- What was left out and why
- Any wizard of oz techniques that are required to make it work

5. **Prototype screenshots or scripts (as many as needed)**

Grading

The report and prototype will be graded together, and the presentation will be graded separately. Here is the grading for the report and prototype (50 pts total):

Design (20 Points)

- Tasks
 - Do the tasks cover the interesting features of the project?
 - Do the tasks have an appropriate difficulty/complexity specified?
 - Do the tasks altogether form a compelling story for the project?
- Changes
 - Were appropriate changes made to address the important problems discovered?
 - Are these changes well illustrated with screenshots or scripts?
- Transition from low-fi to interactive prototype
 - Were the limitations of the low-fi addressed?
 - Were appropriate constraints from the final target platform considered?
 - Does the design adequately represent the final target platform?
 - Were any non-standard interactions described and justified?

Prototype (15 pts)

- Is the prototype accessible and working?
- Can users complete the three tasks with the prototype?
- Were appropriate tradeoffs made between functionality and completeness?
- Are the limitations and tradeoffs described and justified in the report?
- Does the README file summarize these limitations and any other details needed?

Report (15 pts)

- Writing
 - Does the report cover all the topics in the outline?
 - Does the organization follow the outline?
 - Are sub-sections used for easy scanning of important parts?
- Screenshots and Storyboards or Scripts
 - Are important figures referenced and placed inline with the text? *
 - Is there a complete set of screenshots or scripts in the appendix?
 - Are they clearly annotated?

* Use [Relevance-Enhanced Image Reduction](#) to create effective thumbnail images.

Presentation

The presentation grading will be broken into two components: the individual grades of the presenters and a group grade for the presentation of the prototype. The grades for each of these components are explained in more detail below.

Presenter grade

- Suggested Organization
 - Overview
 - Overall problem
 - Representative tasks
 - Overall UI idea including design changes from the previous iteration, rationale behind changes
 - Summary
- Presentation
 - Use slides. Ensure that the presentation shows appropriate preparation, and that visual aids are effective, properly prepared, and properly employed. Make sure that people at the back of the room can see your slides.
 - Cover the required scope within the 10+1 minute time period. Practice and time your presentation.
 - Ensure the presenter makes eye contact and projects well.
- Team work
 - The most common mistake in CS160 presentations is the trying to demo while speaking. One person in ten can do this effectively. Most lose the audience.
 - Have one person do the demo while the other speaks. Practice coordination.

Group grade

- Representative Tasks
 - Did they provide coverage of the functionality?
 - Where the tasks too easy or too hard?
- User Interface
 - Was the interface novel and creative?
 - Was it appropriate for the supported tasks?
 - Does it follow from the task analysis, low-fi prototype, and other sound reasoning?
- Presentation of Functionality
 - Was enough presented to illustrate the representative tasks?
 - Was enough presented to give a flavor of the interface?
 - Was there an appropriate amount given the difficulty of interface ideas?