CS160 Interactive Prototype #1

Problem and Solution Overview (1 paragraph)

Basically reuse from contextual inquiry assignment.

Tasks (¼ page)

3 representative tasks. Make sure that they cover the entire spectrum of easy, medium and difficult.

Do the tasks cover the interesting features of the project?
E.g. for MS Word, an interesting feature would be spell check.

Do the tasks have an appropriate difficulty/complexity specified?
Tasks themselves have to be difficult to perform (at least for the difficult task), and they have to involve features that are difficult (or at least somewhat challenging) to implement.

Do the tasks altogether form a compelling story for the project?
Would the 3 tasks be representative of the problem statement?
Eg. For MS Word, you need to have representative tasks like text entry, print preview/actual printing, retrieving and saving files.

Revised Interface Design (1 page plus screenshots or scripts)

Reuse some parts from lo-fi assignment. Specifically, from the Discussion section in which you highlighted which changes will be made, and why. Include some screenshots and/or scripts to illustrate these changes (don't count under 4-page limit), meaning that it should be clear what was before and after the changes. Focus on the more important changes, e.g. those critical incidents with higher severity ratings.

Previously, in lo-fi assignment, you explained how your lo-fi experiment/prototype was limited. Now, you reuse those limitations, but point out how the hi-fi prototype addresses these limitations. E.g. lo-fi prototype was done using pencil sketches and couldn't display critical info using color, which the interactive prototype addresses.

Discuss how you considered the various constraints of the target platform in your implementation. For e.g., wireless connectivity is available only in vicinity of Park Ranger's station, so to provide users with access to online data when they are elsewhere, we developed our prototype to cache relevant info.

If you have non-standard interaction techniques, be sure to describe them adequately so that we can understand what they are.

Scenarios for 3 tasks. Use storyboards to illustrate how a hypothetical user can perform these tasks using your interface. Storyboards can be actual screen captures, unlike lo-fi assignment.

Prototype Overview (2 pages, 15 points)

Tools used. Up to 1/3 of a page. Describe what tools (e.g. programming language, editors, compilers, emulators, etc.) you used, how the tools helped you to get the prototype implemented (e.g. emulator streamlines development time because we don't have to test code on actual hardware until it's debugged), and the limitations of certain tools (e.g. emulator does not expose undocumented bugs that were specific to hardware platforms).

Adapt overview of UI from lo-fi assignment. Give a description of high-level functionality (e.g. MS Word is an application that allows users to write documents electronically on a desktop or laptop computer…), and then narrow down to low-level description (e.g. MS Word has a menu bar with options for file saving… It comes with a Help feature for users to activate the help agent…). But focus more on low-level description.

And give screen shots for the important parts. E.g. an overall screen shot of MS Word showing the toolbar, document pane, etc. And individual screen shots that are representative of tasks user would need to perform. E.g. print dialog.

About functionality. Make sure that I can run your program and carry out the steps as described in your 3 scenarios, without causing your program to crash, etc. due to bugs.

There may be cases in which certain functionality cannot be implemented in time. Indicate what these features are, and explain why they weren't implemented for this assignment.

E.g. : for speech recognition, if you can't implement this on time, you need to tell the grader how he can Wizard of Oz his way thru this unimplemented feature to proceed to the rest of the scenario.

Appropriate tradeoffs between functionality and completeness. Ideally, you should aim for total functionality, as in, the features for the 3 tasks are all fully implemented. But, due to real world constraints, for the sake of completeness, you leave some buggy features aside and focus on other features to make sure that you have at least a partial implementation for all 3 tasks.

Prototype Screenshots or Scripts (as needed)

All illustrations won't be counted in the 4-page limit, so use them liberally.

Final advice: There are 15 points allocated for report and good writing/presentation, so please proof-read adequately.