HCIPlayer: Project Evolution

ECSE424 Winter 2010, McGill University

Updated: Tuesday, April 13th, 2010

Developing the Alpha System

During the time since the high-fidelity prototype, we have received feedback from user testing as well as from the TAs allowing us to evaluate our prototype's performance. It is necessary to discuss the important observations made by the evaluation team. These assessments also need to be compared to the observations made from our own testing, to gain a better understanding of the utility of the prototype interface.

Discussion on interface

To decide what gestures would be acceptable for controlling the device, it was necessary to try several implementations on multiple different people. This test process and subsequent discussion led to our implementation prior to the evaluation team's tests; the added feedback from the evaluation team has helped us make key decisions while building the alpha system. Additionally, discussion with Dalia, and our own reevaluation of the basic goals of the system have helped improve our design. We analyzed the strengths and weaknesses reported by the evaluation team and have adjusted our prototype to address the usability goals as outlined in our benchmark tasks document. Improvements were made in the following areas:

Stability

System crashes were reported as rampant in the evaluation, which was unfortunately detrimental to the ability of the group to test the usability of the system. This was clearly seen from the results of the Data Collection Survey. We thus decided to change the interface in order to give the user clearer and more descriptive information regarding a system crash. We believe that system crashes that the evaluation team experienced with the prototype were the result of three main issues, namely

1. a weak wireless signal that could have resulted in crashes or states that would be reasonably interpreted as crashes
2. combinations of user inputs that had not yet been accounted for

3. system bugs, often only inconsistently reproducible, which had not yet been removed

We have been working to remove all of the system crashes reported by the evaluation team in order to construct a more robust alpha system. It is difficult, however, to properly indicate to the user that a total system crash has occurred using the iPhone operating system. In the prototype used during the evaluation, a system bug caused a crash upon startup after each crash, leading to two consecutive crashes regardless of what the user did after startup the second time. This was not a well understood system failure at the time of the evaluation, but this issue has since been fixed.

Visual Feedback

We are also aware that the user needs to be able to better understand the difference between a system crash and sluggish normal operation. To allow this, we are increasing the amount of both visual and haptic feedback. It is important when adding visual feedback to ensure that we don't do so at the expense of the user experience of the fully blind. Developing a reasonably robust system of visual cues for the visually impaired required a bit of research. Our research into a proper approach eventually led us to this TED talk, which gave us a few ideas for how to go about the design.

This information resulted in us implementing a moving symbol as an easily recognizable visual cue to indicate that the system is processing and will need additional time to compute. The visual cue chosen is a circle moving in a circular path around the center of the screen. We intend to test additional moving visual cues to see which implementations of moving visual cues work best, possibly offering room for more information to be conveyed. These improvements utilize the iPhone screen and are able to provide useful feedback to a substantial population of our target users, hopefully alleviating some of the ambiguity present in the processing state without causing a detriment to its usability for blind individuals.

Additionally, we have decided to pay more attention to the text and images displayed on the screen during normal operation, which previously had only been for our own debugging purposes. For example, while we were demonstrating our prototype to Dalia, it was unclear if the device was paused or muted. While we have decided to get rid of the mute functionality (as explained later), we still think that displaying more status on the screen would be beneficial. Our alpha system now displays a large white “play” icon when playing, and a large red “stop” icon when the music is stopped. This will require more testing, especially with our assigned visually-impaired individual, to determine how easily these cues can be distinguished.

Our increased focus on visual feedback during this last round of testing have led us to consider some more interesting uses for the music player. We think it would be interesting to add a feature which supported music library navigation without the often clumsy voice commands followed by audio feedback. If time permits, we are considering implementing a quick prototype of this to see if it can improve the usability of the music player.

Redundant Feedback

We believe the player can be made more intuitive by giving the user additional feedback on many different states of operation. With a wide range of use cases which may vary across our range of target users, redundancy of feedback across a number of modes can be beneficial. Since visual feedback should remain exceptionally macroscopic in detail in order to remain accessible to a large gambit of the visually impaired, we are limited in the range of visual cues that can be implemented. Regardless, it is important to include many audio, tactile, and visual cues to assist the user.

Clutter

A lot of thought was put into what features actually add real value to the player, as opposed to just taking space in the user manual or even getting in the way of normal operation. It was determined that a gesture for "*mute*" fit nicely in latter category, providing an additional (not easily distinguishable) gesture for the system to mistake for something else. In only negligible realistic cases would a mute function be preferable to an easily accessible "*pause*" button. It was therefore removed.

The question is not so simple when it comes to speech commands that may or may not be necessary. As so many more commands can be used while retaining a high rate of recognition, it seems like the having e.g. "*previous*" and "*next*" commands available through speech would not hurt the system's usability. It is nearly always easier to use the gesture, as the system requires that the user put a finger on the screen for speech recognition anyways. Realistically, removing commands such as these from the voice recognition system would not significantly improve its ease of use, but could potentially severely reduce it if any user attempted to tell the player 'next' with no response. Like with a number of verbal commands in the system, we have opted for the safer road of redundancy, in hopes that it may, if only in a few cases, improve the usability of the system. As such, we have decided to keep these commands.

Tutorial

Team Ocelot's device featured a brief tutorial, an automated hands-on walkthrough of the device, which we found to be an especially easy way to learn the basic ins and outs of the device, especially for a visually impaired user who may struggle with a printed manual. As such, a tutorial system has been added to the Player to better instruct new users. This makes it simpler for the Player to accommodate a wide range of technical familiarity with portable music players. The evaluation team suggested that our documentation was unclear and a trial-and-error approach eventually prevailed. While we have improved our documentation to address this issue, we also see opportunity to cater even better to first time users.

Evolution from Alpha to Beta

With feedback from Dalia and team e-Mall's evaluation with a test subject we identified a number of improvements that would improve the usability of the beta system. We made advances in the following areas:

Errors

The alpha system suffered from stability issues, much of which was due to the transition from Python to the iPhone's native Objective-C. While it made the errors more predictable, it greatly increased the complexity of the code. The errors encountered in the evaluation were numerous and again detracted from the ability of the evaluation to test the interface usability rather than that of the whole system. The voice and gesture recognition performed very poorly and the tutorial did not function. All of these have been fixed and are fully functional and reliable in the beta system.

Browsing

With a more robust and functional system to work with, we decided to add a way to browse artists and albums. This makes it possible for the user to manage a much larger music collection by breaking that collection down so the user doesn't have to remember the name of every album and song by every artist. With the more complex states available in the device, the "what can I say?" command is now context-aware. This was originally planned in our low-fidelity prototype but was never implemented due to a lack of time. It is a system that would be very helpful to evaluate on users but it was regrettably not implemented in time.

Visual Feedback

Use of visual feedback was increased in the beta system with the addition of a high-contrast screen-width bar to indicate volume level. A question was raised surrounding the icon use in the alpha system and whether it was consistent with the expectations of the user. A play arrow displayed could be seen as either a button while in the pause state or a status indicator in the play state. We came to the same conclusion as team e-Mall, which was to use them as indicators, showing the play icon when playing and the pause icon while stopped.

Voice Commands

The accepted grammar and vocabulary was greatly increased for the alpha system, allowing an array of options, should the user forget a command. Rather than requiring the user to issue a series of separate commands to add multiple items to the queue, they can be listed in a single command using the same flexible grammar offered for other commands. The system ignores commands that deviate too much from expected options to reduce the number of misunderstood commands, which are extremely detrimental to the usability.

Complementary Commands

With the addition of more flexible voice commands, we tried to be less insistent on a single correct way to do most things. The user manual doesn't cover all of the details of different ways to say things since the goal is not to confuse the user. More of this flexibility is lightly touched upon in the tutorial and it is hoped that it will catch mistakes and help the user to feel less confined. Replay, which used to be strictly a voice command is now complemented by the action swiping gesture for previous. This follows a more common model in players where the previous button will restart the track unless it is performed in the first few seconds.