Improving teaching through more flexible presentation control with an iPhone

Joey van der Bie & Martijn Walraven
Media Technology MSc. Program, LIACS, Leiden University
Niels Bohrweg 1
Leiden, 2333 CA The Netherlands
+31-71-5277061/96
joey@vanderbie.net, martijn@martijnwalraven.com

ABSTRACT
Limitations of current methods of presentation control may affect the form and quality of teaching. The design of presentation software and traditional clickers emphasizes linear control over more flexible interaction. We've designed and implemented a prototype of a presentation control application for an iPhone that we believe will help teachers better adapt to their audience by supporting a more free-flowing presentation style.

Keywords
User Interaction and Presentation, Mobile and Personal Devices, Multitouch Devices, Knowledge Sharing.

INTRODUCTION
Slideshow presentation software is increasingly used in university teaching [1]. Where teachers previously used overhead sheet projectors and blackboards to teach, many now teach their classes with PowerPoint presentations. As with any change in teaching materials this has both its advantages and its disadvantages. Some of these disadvantages are not necessarily a result of the use of presentation software however, but of limitations of current methods of presentation control.

As our final project for an HCI course offered as part of the Media Technology program at Leiden University, we've designed and implemented a prototype of a presentation control application for the iPhone [2]. The application goes beyond the functionality of a traditional clicker to support a more flexible presentation style we believe to be especially beneficial for teaching, including the ability to go directly to a specific slide and to make use of additional presentation structure. The iPhone's relatively large screen and multi-touch interface allow the most common actions to be controlled through gestures that do not require focused attention.

The rest of this paper is structured as follows. In the next section we discuss the limitations of current methods of presentation control and its effect on teaching. Following this we introduce our iPhone presentation control application and describe its features, the results of our user evaluations and the changes we made to our design along the way. Next we summarize and discus our experiences. The paper concludes with a brief discussion of potential future work.

Proceedings 13th Computer-Human Interaction Netherlands Conference June 11 2009, Leiden (The Netherlands)

HOW PRESENTATION CONTROL AFFECTS TEACHING
A presenter can usually move forwards and backwards through slides one slide at a time by pressing the arrow keys on the presentation computer or by using the buttons on a clicker. These interactions can be performed relatively thoughtlessly, but anything beyond this requires more focused attention. Presenters often find themselves wanting to skip slides, to go directly to a particular slide, or to remind themselves of the slides that are coming up. This requires going through all the intermediate slides or going back and forth between slides, which disturbs the flow of the presentation and is distracting both for the presenter and for the audience. You'll often find presenters in a frantic search for that one specific slide while the audience is watching. In some cases, presenters may be forced to exit the full screen presentation mode completely in order to locate a specific slide visually or to browse for additional materials such as movies or websites.

Even if a clicker is used, anything beyond moving forwards and backwards one slide at a time requires physical interaction with the presentation computer. Moreover, this usually means exiting the full screen presentation mode and performing interactions in the standard interface, which is not designed for presentation control and was never meant to be visible to an audience.

Most current presentation software has the option of presenting on a separate screen. This allows the display of some additional information on the presentation computer without this being visible to the audience, such as the next upcoming slide or speaker notes. Although this can be helpful, it does little to alleviate the limitations of current methods of presentation control.

We feel these limitations are especially pressing for teachers, as good teaching requires being able to adapt to the audience and to the time available. Teachers have to be able to stray from a preset path, in responding to questions or if they are short on time. If the effort required is too high, teachers may opt not to show that slide that would have helped clarify a question or may spend precious time on a slide they should have skipped. In examples like these, limitations of presentation control have an immediate effect on the quality of teaching.

Our user analysis indicated that these limitations have led some teachers away from using presentation software. This is partly a result of having different teaching styles. Some like a relatively linear presentation so they can plan their lessons ahead in detail. Others prefer a more free-flowing presentation and feel constrained by current presentation software. The latter tend to rely more on the use of websites, the blackboard or other materials. All users agreed they had at times experienced a need for more flexible control of slides and other materials. In particular, they all recognized the limitations of going through slides one at a time. Additionally, there had been occasions where they would have liked to use slides from other presentations or made more use of additional materials such as movies or websites. These needs confirm previous research done on non-linear presentation software [4].

GOING WITH THE SLIDE FLOW

To avoid disrupting the flow of teaching, interactions with presentation software should not be visible to the audience and should distract the presenter as little as possible. Presenters should be able to perform the most common actions without the need for focused attention. Combining the benefits of presenting on a separate screen and using a clicker, we've implemented an application for presentation control on an iPhone. The application is written in Objective-C and uses the iPhone SDK [3] to be consistent with standard iPhone application behavior.

The iPhone is a modern mobile device with a relatively large, high-resolution screen and a multi-touch interface. It is controlled almost completely through gestures and contains an accelerometer to detect orientation changes. A gesture-based interface helps in implementing interactions that do not require focused attention, because we can design our interface so that the precise location of gestures used to control common actions doesn't matter.

Another useful feature of the iPhone is that it supports wireless connectivity over Wi-Fi. Our application uses this to connect to presentation software running on a computer. Although presentation software is not the focus of our project, we implemented a custom presentation application for the sake of prototyping. Manual configuration of IP addresses is not needed because presentation computers on the wireless network are discovered automatically through Bonjour. We recommend creating a private wireless network on the presentation computer however, to avoid disruption of the connection and to keep network quality constant as much as possible.

Our iPhone application operates entirely in landscape mode and uses the whole screen to show the current slide. As a result, slides are shown in relatively high detail, with text and figures at least somewhat readable. This means presenters may not have to look at the presentation screen as often and can glance at the iPhone screen instead. We also chose this approach over an alternative with both current and next slides visible at the same time because we want our application to be able to replace all uses of a traditional clicker. Most clickers include a laser pointer that is used to point to locations on the screen. Although not yet implemented in our prototype, our idea is to allow users to tap a location on the iPhone screen instead and have this indicated on the presentation screen with a semi transparent circle.

Moving forwards and backwards through slides is done through right and left sliding gestures. These kinds of gestures are not new and are used for several touch interfaces types [5]. In addition, the gesture-controlled interface elements on the iPhone obey a sort of virtual physics. Using a sliding gesture to scroll builds up momentum, with the scrolling movement first accelerating and then decelerating. This allows you to either move through slides fast or move from one slide to another slowly. Scrolling through slides is done in a discontinuous manner, a behavior known as paging. When the finger is released the scrolling movement continues until a whole slide is visible. Only when movement comes to a complete stop is the presentation screen changed to reflect the current slide. This enables peeking at the next or previous slide without this being visible to the audience, as the display only snaps to the slide that is more than halfway visible after the finger is released. When scrolling goes beyond the first or last slide the display bounces back, providing additional visual feedback. These behaviors give moving through slides a realistic feel, and have the advantage that users are already familiar with them from their experiences with physical objects.

During our user analysis we found that there are often a few slides to which a presenter returns. These slides can be designated keyslides and function as sort of chapter markers or crossroads because a user can navigate directly to the next or previous keyslide by performing a right or left sliding gesture with two fingers instead of one. The behavior is otherwise similar to that of a standard horizontal scroll, meaning peeking at the next or previous keyslide is also supported.

Additional to horizontal scrolling, up and down sliding gestures can be used to scroll vertically. Vertical scrolling is also discontinuous, and care has been taken not to have horizontal and vertical scrolling interfere. This is done by measuring whether a finger moves a greater distance horizontally or vertically at the start of the sliding gesture and then locking the scrolling into either direction until the finger is released. Unfortunately, this turned out not to be fully supported by standard iPhone interface elements, requiring a somewhat involved implementation.

Allowing for both horizontal and vertical scrolling should convey the impression of moving over a sort of virtual surface. Conceptually, upward scrolling zooms out to show the current element in its context while downward scrolling zooms in to get to elements on a somehow lower level. Our current thoughts on the latter are to use downward scrolling to get to materials related to the current slide, such as movies or websites, but this hasn't been implemented yet.

Scrolling upwards from a slide view gives a thumbnail view, where the current slide is shown in the context of its surrounding slides. Horizontal scrolling can be used to go to the previous or next thumbnail view. Tapping a slide thumbnail in any of these views will make that the current slide and take you back to the full screen slide view. This implementation has similarities with the zoomable user interface Good and Bederson propose [5], but we use a fixed grid instead of a flexible 2D space.

RESULTS AND EVALUATION

To help develop and validate our design, we conducted two rounds of user evaluations using a cooperative evaluation method. We combined user interviews with tests of prototypes of the iPhone presentation control application. Our user group consisted of four university teachers with different teaching styles, all associated with the Media Technology program. As our user group was highly educated and has some affiliation with human computer interaction, we were able to benefit from their well-articulated critique and alternative design proposals.

The use of a full screen slide view seems to work well. Users liked being able to see slides in relatively high detail and believed this would help them avoid looking at the presentation screen too much. The use of horizontal scrolling to get to the next or previous slide and the implementation of sliding gestures were well received. Users immediately felt comfortable with the design and observed it felt natural. This was true even for a user that hadn't been exposed to the iPhone scrolling implementation before.

A feature we hadn't anticipated was the peeking behavior. During our first user evaluation, users discovered that as a result of our implementation of slide scrolling they could peek at the next or previous slide without this being visible to the audience. Users were enthusiastic about this and asked us to consider ways to allow peeking at more than one slide in either direction. We thought up several design proposals and discussed these with some of the users. All designs seemed to suffer from too much complexity however, necessitating the use of multi-finger gestures or additional modes. Although the thumbnail view offers some opportunities for peeking, the slide thumbnails are too small to read and so do not offer true peeking behavior. We therefore feel this could be an area for further exploration.

Although we haven't been able yet to fully implement this in our prototype, users were also enthusiastic about the idea of using the full screen slide view as a pointer replacement. One design we considered was to have an indicator track touches continuously, but users convinced us tapping once at a location instead of navigating towards a location would be less distracting to the audience. It avoids movement of the pointer altogether, including the all too common constantly jiggling pointer. Again, we haven't been able to test any of this so we can't be sure these ideas will work in practice.

One of the main design issues we struggled with was how to get from the full screen slide view to other views, including the thumbnail view and related materials view. In our first prototype, we used changes in device orientation detected through the accelerometer for this. The full screen slide view would be displayed whenever the device is in landscape mode, while an alternative view would be displayed in portrait mode. This view included a tab bar to switch between thumbnail views and a related materials view. After some struggling with the implementation, we felt the design worked well. But users told us they were afraid turning the device would frequently be done accidentally, especially for those who move their hands while walking or talking. Unexpected switches to other modes are always a frustrating experience, but are even worse in speaking situations, where users are often nervous.

Several users suggested using vertical scrolling to implement additional views instead. One user suggested thinking of the area over which scrolling occurs as a virtual surface, with vertical scrolling leading to contextual or detail views, a design found in some digital cameras. We liked the idea and implemented a version of this for the next prototype, with upward vertical scrolling being used to get the thumbnail view. Although since corrected, the implementation had some flaws and as a result we've only been able to do limited testing of this design. Users did think it could be made to work well, and our own observations since confirm this. But further testing and improvement of the design and implementation are likely to be needed, including using downwards scrolling to implement a related materials view. Although holding a device used as a remote control in landscape mode is less natural than holding it in portrait mode, users did not show any real discomfort. Nevertheless, we believe our current design is an improvement in this respect, because sliding gestures for the most common actions are supported in either orientation.

Users liked the concept of keyslides, but we've only been able to perform limited testing of our current implementation. We discussed possible designs for the use of keyslides during the first user evaluation, and converged on a two finger sliding gesture. One user was not convinced the user of multiple fingers would be a natural gesture, but unfortunately we haven't been able to test this with him. Others thought it could be made to work, if implemented well. Our own observations are that it works relatively well in the current prototype. One issue is that it isn't always clear whether you're scrolling over all slides or just over the keyslides. There is definitely a need for better visual feedback here, and our current idea is to use an indication icon for this.

CONCLUSION AND DISCUSSION

As the available time for the project was limited, we haven't been able to implement and test all our ideas. Nevertheless, we are happy with the results we've gotten so far. Our user group has been deservedly critical of some of the design details but very supportive of the general concept. Users were enthusiastic about the prototype and seemed to feel the application had the potential for improving their teaching.

Especially when developing for a mobile device with a distinctive interaction style, there is no real alternative to testing designs on a working prototype. Too much of the user experience depends on using the actual device and on details of the interface implementation (e.g. scrolling behavior). As a result, a lot of our time was spent on developing the prototype application. The iPhone offers a relatively well-designed development environment, but getting to know and using Objective-C and the iPhone SDK still took time. Furthermore, although some features we thought would be difficult to implement turned out to be easy, other functionality turned out to take a disproportionally large part of our time. Some of our design proposals required adapting standard user interface elements, sometimes requiring extensive debugging (e.g. of threading issues) or hacks (e.g. accessing private variables through pointer casts to decompiled structs). Altogether though, we believe the iPhone does offer some of the best development experiences for any mobile device.

All in all, we believe we have at least validated the concept of a presentation control application running on an iPhone, as well as some of the features of our current design. In the process, we've gained a much better understanding of the requirements for an interface that is to be used for teaching. Some of our initial ideas, like the use of orientation changes as a way of switching modes, seemed nice enough in theory but didn't work in practice. Other features, like the peeking behavior, we had not thought about beforehand at all but turned out to be important to users. If anything, these experiences underline the importance of continuous user involvement and prototyping.

FUTURE WORK

Although remote controllers for portable touch devices have been around for several years [6], and new ones continue to appear, none seem to focus on performing a presentation for teaching purposes. Most merely try to port a subset of the functionality that is available on the computer to a portable device. At least five presentation control applications for the iPhone are currently available in the iPhone App Store, including the recent Keynote Remote [7] application developed by Apple. Although some applications are better designed than others, we are not aware of other presentation control applications that allow users to go directly to a specific slide or that support a concept like keyslides, let alone present and fully control additional materials like movies and websites.

We therefore feel continued development of our application is viable and useful. At a minimum, we plan on completing the implementation and testing of the already designed features. Additionally, we believe investigating ways to utilize more flexible presentation structures and to present and control additional materials has the potential for yielding real benefits to teaching.

REFERENCES

1. Craig, R. J., Amernic, J. H. PowerPoint presentation technology and the dynamics of teaching. Innovative Higher Education, 31, (2006), 147-160
2. Apple iPhone information at http://www.apple.com/iphone/specs.html.
3. Apple iPhone SDK, available at http://developer.apple.com/iphone/.
4. Chiu, P., Liu, Q., Boreczky, J., Foote, J., Fuse, T., Kimber, D., Lertsithichai, S., Liao, C. Manipulating and annotating slides in a multi-display environment. Proceedings of INTERACT '03 (2003), 583--590.
5. Good. L., Bederson. B. B., Zoomable user interfaces as a medium for slide show presentations. Information Visualization, 1, 1, (March 2002), 35-49
6. Myers, B. Using Handhelds for Wireless Remote Control of PCs and Appliances. Interacting with Computers 17, 3, (2005), 251-264.
7. Keynote Remote information at http://www.apple.com/iwork/keynote/#remote.