Studying Applications for Touch-Enabled Mobile Phone Keypads

Paul Holleis
University of Duisburg-Essen
Essen, Germany
paul@hcilab.org

Jussi Huhtala, Jonna Häkkilä
Nokia Research
Helsinki, Finland
{jussi.huhtala, jonna.häkkilä}@nokia.com

ABSTRACT
We present a platform to evaluate mobile phone applications that make use of an additional dimension for key presses. Using capacitive sensors on each key, merely touching buttons as well as the force of the press can be measured. A set of applications well known from current mobile phones has been extended with functionality exploiting those new possibilities. The results of a study undertaken with this prototype are presented and conclusions are drawn for the design and implementation of such applications.

Author Keywords
Capacitive touch sensing, mobile phone interaction, small device user interface, haptic input, user studies.

ACM Classification Keywords
H.5.2 User Interfaces: Haptic I/O, Prototyping; H.1.2 User/Machine Systems: Human factors; B.4.2 Input/Output Devices

INTRODUCTION
Mobile phones offer a challenge to input and output techniques due to their small size, which allows only rather limited UI functionality design. Whereas screen resolution has increased allowing more sophisticated GUIs, the input mechanics has remained rather unchanged through the short history of mobile phones. Current phones either use buttons (with tactile feedback) or a touch screen (without tactile feedback). Some projects try to combine these and add tactile feedback to touch screens (e.g. Poupyref and Maruyama [3] who use a small vibrating actuator placed behind small displays), but the affordance and feeling of a button can only be simulated this way.

To overcome the limitations of small keypads, several approaches have been suggested including speech input and auditory UIs [1], but e.g. social situations can limit their usage. Gesture input has also been suggested, but the method may suffer from problems in gesture recognition, as well as in social acceptability [6]. One approach that enhances the conventional keypad input is to add a touch sensitive layer on top of the buttons. In this way, it is possible to detect touches and light presses on the buttons as well as use the button in its normal, known way. This technology enables additional functionality and interaction types to be added to standard applications.

In [5], Rekimoto et al. first presented a working prototype of a keypad augmented with capacitive sensors and introduce the notion of "previewable user interfaces". Similar to tooltips on PCs, touching a button can give information about what will happen if this button is pressed. The authors provide several application ideas; however, no evaluations with users have been reported. Moreover, there is no working version with a real mobile phone. Clarkson et al. [2] add a layer of piezoresistive force sensors, one below each phone key. This enables them to continuously measure force that is exerted on each key. It therefore adds a continuous dimension to each key and enables applications such as smooth zooming into images proportional to the force on a button. The authors present among others an application for text input that uses a technique similar to that presented by Zeleznik et al. [7] simulating a tri-state button. A disadvantage of the approach with the force sensors is that below a certain amount of pressure on a button the force sensors cannot register the touch. The sensors also cannot distinguish between pressure, e.g., in a pocket from the touch of fingers.

In a way, Rekimoto et al.'s PreSenseII [4] further develops this concept by using a touch pad and adding force sensors below it. Thus, a continuous space in three dimensions can be achieved. However, this looses the affordance and feeling of buttons and tactile feedback has to be simulated with a piezo-actuator. Again, no formal studies have been undertaken to get feedback from users. Although the touchpad clearly gives more freedom and possibilities, we argue in favor of the affordance and clear separation of buttons and see an advantage in leaving the current user interface unchanged to ease the process for users of getting used to such a new way of interaction.

TECHNOLOGY AND PROTOTYPE

Our prototype uses proximity and touch sensors based on capacitive sensing. This technology has been employed in many research projects and can also be found in several appliances like modern stoves where controls can be activated by simply touching on a specific area. A sensor like the Analog Devices[1] QT110 is used to detect changes in the capacity of an electrode. Touching an electrode or coming in close vicinity of it, e.g. touching a thin glass layer placed on top of it, changes the measured capacity enough to trigger an event.

Figure 1: Nokia N73 with Microcontroller + Bluetooth board and AD7142 touch sensor board (far right) attached to the top.

An AD7142 sensor can detect touches on 12 electrodes (nearly) concurrently. It also features complex on-chip environmental calibration methods. Main advantages of this type of sensor - besides a very small footprint of roughly 6x6mm - include a fast update rate of 36ms and the support of several programmable parameters that allow custom calibration of the sensing mechanism for each electrode.

A Microchip[2] PIC2550 microcontroller communicates with the sensor using the I^2C protocol. After a power-up sequence, the configuration and initial calibration data is written. Sensor values can then be read at approximately 25Hz. The AD7142 sensor can be configured such that it provides an interrupt output which is set to high whenever a touch is detected. Upon detection, the microcontroller sends an event with the key that has or the keys that have been touched to the phone.

Since it is difficult to directly interface current phones with external hardware, we use a Bluetooth module (Linkmatik 2.0[3]). The microcontroller is connected to this module and can send information through a serial connection to the Bluetooth module. The module supports connections to any device and program that support the serial communication over Bluetooth protocol. Thus, it is possible to pass events from sensor readings to J2ME, Python for S60, etc. programs running on the phone with negligible delay.

In order to place the electrodes on the phone keypad, the N73 phone was disassembled and thin isolated metal wires were placed around the keys. Figure 2 shows how the wires have been routed in a way they do not touch and still cover a large part of each key. The wiring is nearly invisible and does not change feeling and functionality of the keys at all.

We decided to use the FlashLite environment to develop the application simulations. This has the advantage that graphics and animations can easily be integrated in programs. It also enables the straightforward initial testing and evaluation of the applications. Background themes and actions can also quickly be altered and replaced.

Figure 2: Left: Keypad of the N73. Position and layout of the not touching electrode wires is indicated with thick black lines. Right: The 3 main touch-sensitive areas used in the test applications. Keys '5' and '8' always trigger the same action which helps in the process of disambiguation.

A problem with FlashLite in its current version is that it does not support serial connections to Bluetooth devices. Inspired from the Flyer project[4] (which relies on Python for S60 and requires a full installation of Python and Python Script Shell), we developed a J2ME midlet that connects to a Bluetooth module and relays all data to a FlashLite application. To achieve this, it uses the XMLSocket connection available in FlashPlayer 3.0. Thus, touch events from the sensor platform received by the J2ME midlet are forwarded to the FlashLite applications where they trigger a specific action like showing a pop-up menu.

Controls

We deliberately did not use the hotkeys / joystick region of the phone to attach the touch sensors. First of all, the area is very small such that touch with fingers could often lead to wrong detections. Second, we did not want to restrict ourselves to ideas using menu navigation etc. but also plan to enhance text and number input. And lastly, there are more and more applications that use the keypad as main control or shortcuts since hotkeys are dedicated to menu and options navigation.

APPLICATIONS

We developed 3 distinct applications building upon the existing phone book view and the image gallery, simulated with FlashLite.

---

[1] Analog Devices, http://www.analog.com

[2] Microchip http://www.microchip.com/

[3] RF Solutions Ltd, http://www.rfsolutions.co.uk

[4] Flyer project page: http://www.flyerframework.org/

User interface

The applications use a graphical user interface resembling the existing Series 60 user interface found in Nokia phones. The applications presented a phonebook and image gallery application. The basic functionality was the same as in the standard applications except the number keys are used to scroll and select.

Info screen application

The first application was used to introduce the touch-press feature. It was possible to browse names available in the phone book by pressing up and down and to look at the respective phone number by pressing the middle button. When touching the middle button, it was possible to see recent calls to this contact in a pop-up screen, see Figure 3, left and center image. This info screen disappears after a moment when the finger is released from the button.

Figure 3: Phonebook application with two types of pop-up info screens that appear when touching the select key.

Pictures application

The second application also used the phone book, but the pop-up screen contained a collection of images related to the selected contact (Figure 3, right). During the time the pop-up was shown, all available images could be scrolled by touching the up and down keys. This pop-up also automatically disappears if the user does not touch any buttons for a short while.

Figure 4: Image gallery application. While clicking selects, touching zooms in and out of the currently selected image.

Gallery application

The third application featured a thumbnail gallery, where users could browse through images by pressing up and down buttons and enter a full screen browsing mode by pressing the middle button. In full screen mode it was possible to zoom into and out of the image by touching the up and down keys. The sequence is illustrated in Figure 4.

USER STUDY

The main focus of the study was to find out whether people understood the concept of touchable buttons, how they judged the user experience of this kind of interaction and whether they could find benefits from such an extension.

We recruited 10 people for the tests, 4 female, all of them Finnish, between 20 and 50 years old, most of them having a technical background. All of them were right-handed, familiar with the S60 look-and-feel, used the phonebook daily and image gallery at least weekly. Besides telling the users that the buttons were touch sensitive, we did not give any details in order to see how intuitive the prototype user interface was.

In the tests we had two kinds of use for this technique. One was using touch sensitive button as enabling pop-up windows which included additional information about menu items and another was using the technique as shortcut to command. The given tasks were:

1. Select “Lisa” in the two phonebook applications and touch the middle key.
2. Scroll the images in the (second) phonebook application and exit the view.
3. Select the image of the city in the gallery and find out how to zoom.

The user tests took place in a usability lab furnished like a standard living room. A small video camera mounted on the phone was continuously recording finger movements and screen content. After the tests, people filled in a questionnaire where they could judge and comment on different usability metrics.

Study Outcome

Only two of the ten participants had difficulties using the first application – mainly because there was a too long delay before the window popped up and users didn't keep their fingers long enough in the same position. We reduced the delay from 0.7 to 0.5 seconds for the other participants which clearly improved the interaction. Despite of the delay, people perceived a clear difference between touch and press actions, and they were not unsure whether the system recognized press or touch input.

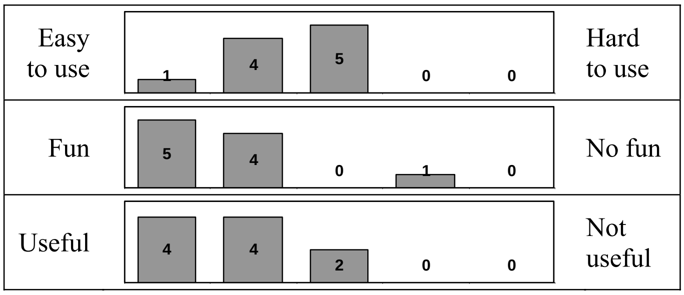

Figure 5: Collected feedback on 3 measures. Values ranged from ‘completely agree’ to ‘neutral’ to ‘completely disagree’.

The usefulness of the technique was rated high, figure 5, and the participants found both pop-ups and shortcuts practical. The participants felt that touching can increase input speed and in general liked it. Our application made zooming easier and faster than in existing s60 galleries, and the interaction techniques was seen as a way to reduce the use of menus. Improvements in usability were mentioned after trying out the zooming in the image gallery, because the use of complex menus could be reduced. One participant mentioned that the input technique could also improve possibilities to use phone with one hand only, as today for example the copy-paste function in S60 phones needs to be performed with two hands.

As application specific feedback, the users mostly liked the idea of pop-up screens, but typically wanted to see some other information than recent calls, e.g. a phone number, address, a picture, or application help. The opinion on the way the pop-up window was closed was split. One half liked that it disappeared automatically, the other half wanted to manually control when the screen would go back to standard mode. The image gallery application turned out to be the most popular one. Eight of ten users found out the zoom function without help, and after finding it, they all liked it. Here, dealing with the delay was somewhat problematic again.

Even though people were quite positive towards the interaction technique, they mentioned also potential difficulties related to it, as seen also from the feedback in figure 5. The concerns with the input method were related to the learnability and memorability. The participants were worried that they had more things to remember because they have to learn where touch is enabled and what functions the buttons had, and if it was too different to the conventional UI. Also, anxiety to learn the correct sensitivity for a proper use of touch-press was mentioned. Interestingly, when using the device, some people tried also if scrolling movements would also work.

DISCUSSION AND CONCLUSIONS

The last section is used to sum up the results from the studies of the touch enhanced mobile phone keypad.

To have such functionality in a deployed product, the sensing system must work extremely well, as people are very reluctant in accepting erroneous or slow device behavior. Here, achieving such a correct behavior is especially challenging since the sensor placement and sensitivity should be optimized for the mass market. Most people are also averse to having to do calibrations when they start using a new phone. Based on our observations, we recommend that it should also be possible to switch the feature off.

As can be seen in Figure 6, it must be taken caution to have enough spacing between the buttons. When pressing a button, the one below is often touched at the same time. This effect is especially strong if the keypad is used with the thumb and increases for buttons farther away from the center of the hand since then the finger is held more horizontally.

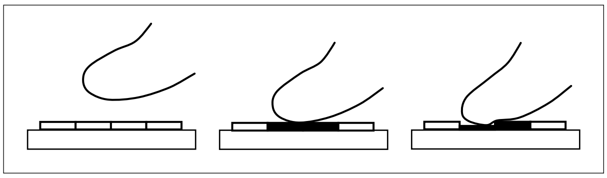

Figure 6: Illustration showing how a finger often touches a lower key, e.g. '8', at the same time when clicking on a key on top of that, e.g. '5'.

There is a necessity of a small delay (we found a value of less than 0.5 seconds to be sensible) between touch and triggered action since otherwise the action would always occur when pressing a button. Since people do not like to wait, such touching interactions should mainly be used to display additional, non-critical information or to enable functionality with a large benefit (e.g. shortcuts). Often, people are hesitating before they initiate an action, and offering help at such times seemed to be a potential type of application. However, we also observed that people often move or keep their fingers on a button without a particular intention or just to reflect about something, potentially resulting in erroneous actions. Also, based on our observations, wide and varying use of touch and press can lead to confusion and be hard to learn and remember. Thus, UI designers must be very careful in using touch for that since otherwise it might quickly be more annoying than helpful.

REFERENCES

1. Brewster, S. Overcoming the Lack of Screens Space on Mobile Computers. *Personal and Ubiquitous Computing*, 6(3):188-205, 2002.
2. Clarkson, E. C., Patel, S. N., Pierce, J. S., Abowd, G. D. Exploring Continuous Pressure Input for Mobile Phones. GVU Technical Report. GIT-GVU-06-20. http://hdl.handle.net/1853/13138. 2006
3. Poupyrev, I., Maruyama, S. Tactile Interfaces for Small Touch Screens. In *Proc. UIST'03*. 217-220. 2003
4. Rekimoto, J. Schwesig, C. PreSenseII: Bi-directional Touch and Pressure Sensing Interactions with Tactile Feedback. In *Extended Abstracts CHI'06*. 1253-1258. 2006
5. Rekimoto, J., Ishizawa, T., Schwesig, C., and Oba, H. PreSense: Interaction Techniques for Finger Sensing Input Devices. In *Proc UIST'03*. 203-212. 2003
6. Ronkainen, S., Häkkilä, J., Kaleva, S., Colley, A., Linjama, J. Tap Input as an Embedded Interaction Method for Mobile Devices. In *Proc. TEI'07,* 263-270.
7. Zeleznik, R., Miller, T., Forsberg, A. Pop Through Mouse Button Interactions. In *Proc. UIST'01*. 195-196. 2001