Keyboard with tactile feedback on smartphone touch screen

Emanuele Panizzi
Department of Computer Science, University of Rome 'La Sapienza'
Rome, Italy
panizzi@uniroma1.it

ABSTRACT

Pressing buttons on a smartphone touch screen is difficult if you are not looking at the screen. We developed a numerical keyboard that provides a tactile feedback using phone short vibrations. The feedback is provided both when the user swipes the keyboard and when he presses keys. We describe how we implemented it on iPhone7, using the iPhone 3Dtouch capability and the UIFeedbackGenerator.

CCS CONCEPTS

• **Human-centered computing → Haptic devices**; *Touch screens*; Interaction paradigms;

KEYWORDS

haptic feedback, tactile feedback, iPhone 3Dtouch, keyboard, touch screen

ACM Reference Format:
Emanuele Panizzi. 2018. Keyboard with tactile feedback on smartphone touch screen. In *AVI '18: 2018 International Conference on Advanced Visual Interfaces, AVI '18, May 29-June 1, 2018, Castiglione della Pescaia, Italy*. ACM, New York, NY, USA, 3 pages. https://doi.org/10.1145/3206505.3206563

1 INTRODUCTION

We are used to pressing buttons on the touch screen of our smartphone. We have no different haptic feedback when touching the button than when we touch any other point on the screen, so we must rely on our sight to be sure that we hit the button, especially when there are more of them on the display. We also need a visual feedback to reassure that the button has been pressed, i.e. that the system received our input. By contrast, on feature phones (e.g. older mobile phones) there are physical buttons and we are able to recognize them just touching, without looking at the keyboard, thanks to the tactile feedback they provide. Moreover, the button associated to number 5 has a raised dot or line that helps recognizing it specifically, and that helps locating the finger position with respect to the whole keyboard. Finally, tactile feedback received when pressing a physical key is generally very noticeable on feature phones.

This work aims at studying if it is possible and useful to replicate the haptic feedback of a physical keyboard on a smartphone, to allow locating and pressing buttons without the need to look at the screen.

AVI '18, May 29-June 1, 2018, Castiglione della Pescaia, Italy

ACM ISBN 978-1-4503-5616-9/18/05.
https://doi.org/10.1145/3206505.3206563

We designed and implemented a numerical keyboard to be displayed on a smartphone that provides a tactile feedback using the phone vibrations to allow recognition of keys and of key-press. We chose an Apple iPhone7 smartphone because it supports different types of vibrations and also detects the force applied by the finger thanks to a feature called 3DTouch. The advantage of using 3DTouch will be explained in section 3.

We developed our tactile keyboard and we ran a user test with 34 users. We describe here its implementation and use. The analysis of the user test we conducted is matter of another paper being written, but results are encouraging and can justify the applicability of the idea in mobile applications to be used in situations where the user has few possibilities of looking at the smartphone screen.

Our aim is to demonstrate the tactile keyboard at the conference to foster discussion and get feedback.

2 RELATED WORK

Tactile feedback in touch screens has been studied extensively in the last decades, and we focused on studies related to mobile devices, that explore different tactile feedbacks, and aimed at the evaluation of eyes-free interaction for normally-sighted users.

Interesting studies like [3], [4], and [5] evaluate tactile feedback in keyboards, and several researches approached techniques to add tactile feedback to mobile devices, like [6], [7], [8], [9], and [10]. Our work explores a space similar to theirs, but using a standard commodity device; our study confirms some of their results.

Some studies try to convey meaning by differentiating tactile feedbacks: Brewster and Brown proposed Tactons (tactile icons) [2]; Yatani [12] differentiates vibration patterns to indicate position and shape of buttons. Pakkanen et al [11] associated different haptic feedback to keys in a numeric keyboard. With respect to this aspect, our keyboard, just with two different vibrations patterns, is able to reduce input error rates.

3 TACTILE KEYBOARD

We designed and developed a single-view iPhone application with a numerical keyboard containing the keys from 1 to 9 (not 0) positioned as in a telephone keyboard (key 1 on the top line, thus opposite than a calculator keyboard), see figure 1. Our keyboard is deliberately non standard, in fact it misses the fourth row (buttons star, 0, and pound). In fact, we decided to start testing the feasibility of our idea with a simplified keyboard, and then, if successful, extend it to a standard keyboard or apply it to single buttons.

We designed the interaction so that the user can touch the keyboard, swiping his finger on it, and this does not produce any input. While swiping, he receives tactile feedback and can determine his finger position on the keyboard without looking at it. In fact, crossing the borders of key 5, a short vibration is produced, useful to feel

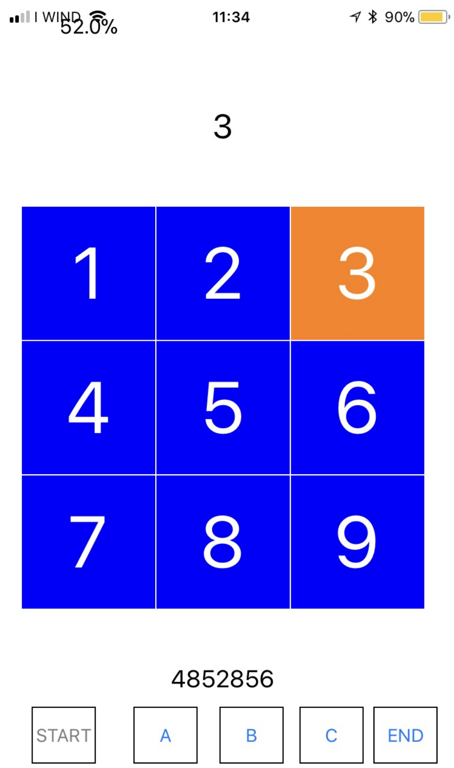

Figure 1: Tactile keyboard plus buttons and labels for experimental purposes. Key 3 has just been pressed (firmly).

when the finger enters or exits that key. We call this vibration *border feedback*. For example, swiping the finger from bottom to top from key 8 to key 5, a short vibration is provided when the low border of key 5 is crossed, and then, continuing the swipe towards key 2, the same feedback is provided again when border 5-2 is crossed. The same applies swiping from top to bottom on that line and on the horizontal line of keys 4-5-6 in both left and right directions.

Borders other than those of key 5 do not provide feedback, in order to avoid confusion. In fact it seemed to us very possible to point at the remaining keys (1,3,7,9) with hopefully low error, based on the position of key 5 and its vertical and horizontal neighbours.

The user, after positioning his finger on a button, must press firmly on it in order to input its number; so he gets a new feedback that we call *key-press feedback*, different than the border feedback.

Vibrations, especially those we used for border feedback, are very subtle, and are perceived better on the palm of the hand that keeps the iPhone than on the finger that is touching the screen.

A visual feedback is also provided. In fact, the button currently below the finger changes colour from blue to orange; when a button is pressed firmly, its input is reported on the label above the keyboard (figure 1).

We developed the iPhone application in Swift language, making five iterations which included tests to single users and refinement of the visual and haptic behaviour of the keyboard. We used the `UIFeedbackGenerator` class to define and trigger the different feedbacks.

We use the `UISelectionFeedbackGenerator` sublcass to handle the border feedback, carefully preparing feedbacks in advance and thus reducing latency when the feedback is triggered. We chose this feedback as Apple suggests to use it *for communicating movement through a series of discrete values* [1], for example in pickers. With this feedback, the user perceives light taps when he crosses the borders of key 5.

On the other hand, we use the `UIImpactFeedbackGenerator` for the key-press feedback, initializing it with the `.heavy` style to maximize the feedback perception. With this feedback, the user perceives a *thud* when he presses firmly a key. After some pilot tests, we opted for a threshold of 90% of maximum possible force, over which the key accepts the input. In fact, we noticed that a threshold as high as 80% was not enough to prevent unwanted key presses. In order to avoid errors, moreover, we do not accept a second key-press on the same key if the finger force doesn't drop below a low threshold and raises again. This low threshold has been set to 50%, with good results in pilot tests.

3.1 Test modes and keyboard appearance

The buttons at the bottom of the view and the numerical labels displayed in the view were needed for the user test.

The bottom buttons are designed so that the keyboard can be shown in three different appearances: START provides the standard appearance described above; both A and B (used for two different tasks in the experiment) remove the visual feedback, in fact the keyboard is rendered in white color over a white background (so it is not possible to see it), but preserve border and key-press feedbacks; C is like A and B but it also removes the border feedback. Note that the position of the keyboard remains unchanged in all three appearances.

When a user presses one of the bottom buttons (START, A, B, C), a new random 7 digit number is generated and displayed in the label below the keyboard. This number is used during the tests as input task for the user, and can be used to demonstrate the keyboard or to exercise with it. The label above the keyboard displays the user input, while the left topmost label in the view shows the current finger force applied, as percentage of the maximum detectable force. The END button is used to finish a user test and save results. In fact, collected data are sent to our server and saved in a database on a per-test basis.

4 CONCLUSIONS AND FUTURE WORK

We described the design and implementation of a haptic keyboard on iPhone7 which provides tactile feedbacks when crossing the borders of key 5 and when inputting any digit. We ran a test with 34 users, not described here, that showed how tactile feedback can help finding the position of keys and inputting numbers without looking at the smartphone screen. In the conference demo users will test the interface as above, and discuss their experiences.

Next work will include testing the keyboard after a longer training and testing the border and keypress feedback on single buttons in different GUIs and and in apps used in the car environment.

REFERENCES

[1] 2018. Feedback - User Interaction - iOS Human Interface Guidelines. (2018). Retrieved Jan 20, 2018 from https://developer.apple.com/ios/human-interface-guidelines/user-interaction/feedback/

[2] Stephen Brewster and Lorna M. Brown. 2004. Tactons: Structured Tactile Messages for Non-visual Information Display. In *Proceedings of the Fifth Conference on Australasian User Interface - Volume 28 (AUIC ’04)*. Australian Computer Society, Inc., Darlinghurst, Australia, Australia, 15–23. http://dl.acm.org/citation.cfm?id=976310.976313

[3] Stephen Brewster, Faraz Chohan, and Lorna Brown. 2007. Tactile Feedback for Mobile Interactions. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI ’07)*. ACM, New York, NY, USA, 159–162. https://doi.org/10.1145/1240624.1240649

[4] Malcolm Hall, Eve Hoggan, and Stephen Brewster. 2008. T-Bars: Towards Tactile User Interfaces for Mobile Touchscreens. In *Proceedings of the 10th International Conference on Human Computer Interaction with Mobile Devices and Services (MobileHCI ’08)*. ACM, New York, NY, USA, 411–414. https://doi.org/10.1145/1409240.1409301

[5] Eve Hoggan, Stephen A. Brewster, and Jody Johnston. 2008. Investigating the Effectiveness of Tactile Feedback for Mobile Touchscreens. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI ’08)*. ACM, New York, NY, USA, 1573–1582. https://doi.org/10.1145/1357054.1357300

[6] Seung-Chan Kim, Ali Israr, and Ivan Poupyrev. 2013. Tactile Rendering of 3D Features on Touch Surfaces. In *Proceedings of the 26th Annual ACM Symposium on User Interface Software and Technology (UIST ’13)*. ACM, New York, NY, USA, 531–538. https://doi.org/10.1145/2501988.2502020

[7] Emilia Koskinen, Topi Kaaresoja, and Pauli Laitinen. 2008. Feel-good Touch: Finding the Most Pleasant Tactile Feedback for a Mobile Touch Screen Button. In *Proceedings of the 10th International Conference on Multimodal Interfaces (ICMI ’08)*. ACM, New York, NY, USA, 297–304. https://doi.org/10.1145/1452392.1452453

[8] Pauli Laitinen and Jani Mawnpaa. 2006. Enabling mobile haptic design: Piezoelectric actuator technology properties in hand held devices. In *Haptic Audio Visual Environments and their Applications, 2006. HAVE 2006. IEEE International Workshop on*. IEEE, 40–43.

[9] Rock Leung, Karon MacLean, Martin Bue Bertelsen, and Mayukh Saubhasik. 2007. Evaluation of Haptically Augmented Touchscreen Gui Elements Under Cognitive Load. In *Proceedings of the 9th International Conference on Multimodal Interfaces (ICMI ’07)*. ACM, New York, NY, USA, 374–381. https://doi.org/10.1145/1322192.1322258

[10] Andrew Nashel and Sharif Razzaque. 2003. Tactile Virtual Buttons for Mobile Devices. In *CHI ’03 Extended Abstracts on Human Factors in Computing Systems (CHI EA ’03)*. ACM, New York, NY, USA, 854–855. https://doi.org/10.1145/765891.766032

[11] Toni Pakkanen, Roope Raisamo, Katri Salminen, and Veikko Surakka. 2010. Haptic Numbers: Three Haptic Representation Models for Numbers on a Touch Screen Phone. In *International Conference on Multimodal Interfaces and the Workshop on Machine Learning for Multimodal Interaction (ICMI-MLMI ’10)*. ACM, New York, NY, USA, Article 35, 4 pages. https://doi.org/10.1145/1891903.1891949

[12] Koji Yatani and Khai Nhut Truong. 2009. SemFeel: A User Interface with Semantic Tactile Feedback for Mobile Touch-screen Devices. In *Proceedings of the 22Nd Annual ACM Symposium on User Interface Software and Technology (UIST ’09)*. ACM, New York, NY, USA, 111–120. https://doi.org/10.1145/1622176.1622198