Gaming the Android OS for Improving the Design of Smartphone Launchers

Matthias Böhmer

DFKI GmbH

Saarbrücken, Germany

matthias.boehmer@dfki.de

Antonio Krüger

DFKI GmbH

Saarbrücken, Germany

Antonio.krueger@dfki.de

Abstract

With a growing number of mobile applications available on application stores and the improved capabilities of smartphones, people download more applications to their devices. With people having more applications installed, finding applications quickly when they need them can become a time consuming issue that impacts user experience. In this paper, we present our approach to improve future design of smartphone launcher menus by following a large-scale research method. We designed a game that takes advantage of the user's smartphone itself as a game court. By measuring the time of a simple visual search task for an icon, we aim to deduce how well a user knows where he can find his applications, thus how well he can build a mental model of his smartphone launcher menu. We present work in progress, preliminary results and discuss our approach and future work.

Author Keywords

Mobile applications; game; operating system; launcher menus; smartphones; research in the large.

ACM Classification Keywords

H.5.2. User Interfaces: Evaluation/methodology.

Introduction

The number of applications (apps) available on mobile app stores is steadily increasing, and so is the number of apps people install on to their smartphones. This also results in a growing number of apps people have installed on their devices. People put quite some effort into organizing their launcher menus [2]. However, with an increasing number of icons in a launcher menu, finding an app can also become a time critical task. Although such time spans might be very short, searching too long for an app might negatively impact user experience. Minimizing this search time is the goal of ongoing research (e.g. [2,5,6,8]).

The question of our research is: How well do people know their current launcher menus? Our goal is to quantify the time it takes to find a certain app on a smartphone, and to build a model for predicting this effort. This will help to improve menus of smartphone launchers, e.g., if a user would take exceedingly long to launch his next app (what can be predicted [5,6]), an adaptive menu could provide a shortcut to that app where he can easily find it when visually searching for that particular app [8]. Ziefle and Bay [9] studied how people build mental models of the previous phone generation's menu structures. In [2] we study how people organize icons in their launchers.

This paper makes three contributions: (i) We present a game as a study in the large to address our specific research question on visual search times in launcher menus; (ii) we present preliminary findings on how fast people can find icons; and (iii) we present our approach of gaming the operating system (OS), and discuss how it can be used for informing future design of smartphones conducting research through app stores.

Approach: Gaming the Android OS

The design and release of games has proven to be a vital research tool for running research studies through mobile app stores. Games are naturally good candidates for high degree of user interest and engagement, which is required to successfully collect enough data for meaningful results. Pioneering work by McMillan et al. [4] leverage a game to study the capability of research in the large as a method itself. Henze [3] was able to study several questions by releasing a simple game. However, so far such game-based studies investigate questions that were encapsulated and controlled within the released app itself. In contrast, in our approach we designed a game that becomes part of the users smartphone itself and makes it an essential part of the game play. The user's smartphone itself becomes the apparatus of our study.

We have built a simple game to study how well fast people can find icons in their launcher menus, called *rappidly*[1]. The objective of the game is to start apps as fast as possible. Therefore users can start new laps, and in each lap he has to find a new random app that he has installed, as Figure 1 shows. When a user decides to start a new lap he will see a countdown (Figure 1a-c), and finally the icon and name of an application with a started stop watch (Figure 1d). From this screen the user has to search for the shown app (here: *WhatsApp*) in own launcher menu and launch it from there (Figure 1e). As soon as he has clicked the app icon he will return to our game and see how long it took him to find the app icon (here: 6.334 seconds; Figure 1f). From here he can decide to start a new lap or post his lap stopping time to *Facebook*.

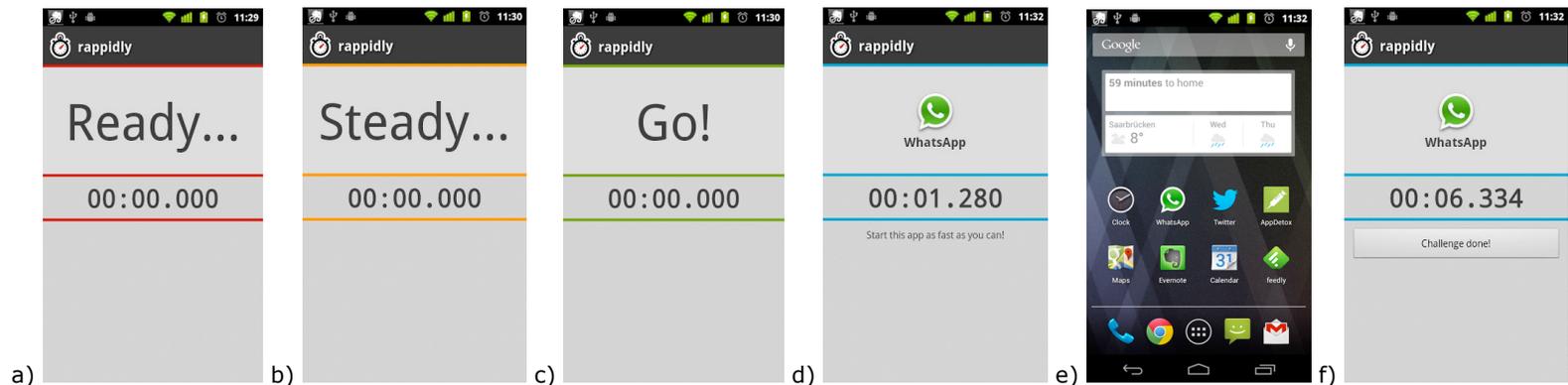[1] Available on Google Play: `http://bit.ly/rappidly`

Figure 2. Gameplay and the screens a user will see when starting a new lap in our game: (a-c) countdown for launching the app, (d) instructions which app to start, (e) searching for the app in smartphone launcher menu for launching it, (f) final scoring screen when app was launched.

We enriched the gameplay with elements of social networking to make the game itself more competitive for games, and to collect demographics on our players (when they connect to *Facebook*).

Preliminary Findings of Study

Within the game we implemented means for measuring the user's performance when playing the game. Essentially we kept track of a user id, a lap's app, and how long it took him to start the app. This data was collected and send to our servers when a user shared it publicly on Facebook (otherwise not). We released the first version in June 2012, and since then it was installed more than 590 times. So far, 41 users have contributed data from 304 laps played with 164 different apps. One user played 7.41 laps mean (SD 19.90, min 1, max 128). On average, in the laps they played people took 5.66 seconds to find an app (SD 4.46). On the lower end (min 1.20, below 2 sec) in particular we can find laps on note taking apps (e.g. *Evernote*), which one usually needs fast access to when noting down a quick thought. We found the maximum search time for somebody searching for *Facebook* (40.65 sec).

Discussion and Challenges

Gaming the OS to Inform Design

Our approach of making the operating system itself the game court of the game we designed allows us to study aspects, which are not inherent in our application, but are customized and created by the very end-user himself outside of our direct control through the app. We think that this approach can be extended to study other aspects of smartphone UI design leveraging the method of research through the app store. For example, by adopting a game on memorizing sequences of items (e.g., TouchMe [7]) to smartphones one could test for how well people can memorize the stack of recent apps. Speaking more general, the stimulus for a study task needs to be given and controlled by the released app, while the task is conducted on the users smartphone out of direct control, and the task needs to be observable either directly or through a proxy measure.

Information on Navigation Paths and Menu Structure
In our current implementation of the game we cannot retrieve any information on the path a user takes for navigating to an app icon after leaving our application. Such tracing would only be possible by applying more intrusive techniques (e.g. logging raw touch events), or getting a handle on the launcher menu itself. The latter would also allow querying how a user has arranged his icons [2]. However, so far we were not able to implement such functionality and by deploying a new launcher app with the game itself we would destroy the menu structures and mental models that users created, thus bias search performance.

Data Cleaning and Ground Truth
As a common shortcoming of studies conducted through the app store in a rather uncontrolled setting, we cannot know about some circumstances of the user when playing the game. Obviously, we cannot know if the user is playing the game with full attention, what might result in lower performance if not the case. Further, we do not know if using *rappidly* has an impact on how a user organizes his launcher menu; e.g., we cannot know if *rappidly* is such engaging for a user that he optimizes his icon arrangement for best performance in the game instead of common reasons [2]. For instance, it remains unclear why somebody took more than 40 sec to launch Facebook in our sample.

Conclusion and Future Work
In this paper we presented *rappidly*: a game designed to study how fast people can launch mobile apps. We described preliminary results for informing the design of future smartphone launcher menus. Further, we discussed our approach of gaming the OS, which allows implementing research questions on the general design of smartphone UIs with the advantages of conducting research through the app store. Future work is on enlarging the user base of *rappidly*. Besides collecting data on how fast people can launch apps we will trace general app usage using our *AppSensor* [1]. This will allow testing hypothesis like: the more often people launch apps, the faster they can find them; or: the more apps people have installed, the longer it takes them to find single apps.

References
[1] M. Böhmer, B. Hecht, J. Schöning, A. Krüger, and G. Bauer. Falling asleep with angry birds, facebook and kindle: a large scale study on mobile application usage. In: MobileHCI '11

[2] M. Böhmer and A. Krüger. A study on icon arrangement by smartphone users. In: Proc. of CHI 2013.

[3] N. Henze. Hit it!: an apparatus for upscaling mobile HCI studies. In: Proc. of CHI '12 Extended Abstracts.

[4] D. McMillan, A. Morrison, O. Brown, M. Hall, and M. Chalmers. Further into the wild: Running worldwide trials of mobile systems. In: Proc. of Pervasive 2010.

[5] A. Parate, M. Böhmer, D. Chu, D. Ganesan, and B. M. Marlin. Practical prediction, prefetch, and prelaunch for faster access to applications on mobile phones. In: Proc. of UbiComp 2013.

[6] C. Shin, J.-H. Hong, and A. K. Dey. Understanding and prediction of mobile application usage for smart phones. In: Proc. of UbiComp, 2012.

[7] Wikipedia: Touch Me (arcade game), http://goo.gl/J2UTL

[8] C. Zhang, X. Ding, G. Chen, K. Huang, X. Ma, and B. Yan. Nihao: A predictive smartphone application launcher. In: Proc. of MobiCASE 2012.

[9] M. Ziefle and S. Bay. Mental models of a cellular phone menu. comparing older and younger novice users. In: Proc. of MobileHCI 2004.