Usability Testing for Mobile Scenarios of Fragmented Attention

Grischa Schmiedl, Kerstin Blumenstein, Markus Seidl

University of Applied Science St. Pölten
Matthias Corvinus-Str. 15
3100 St. Pölten, Austria
+43/2742/313228-0
{grischa.schmiedl,kerstin.blumenstein,markus.seidl}@fhstp.ac.at

ABSTRACT

During the use of mobile applications on the move the user's cognitive resources are split into handling the application and conducting a primary task, i.e. walking or driving a car. Traditional lab-based usability tests do not take the influence of fragmented attention into account. In field tests on the other hand we cannot control environmental conditions.

In this paper we show how to use a car simulator to generate a controllable primary task. By comparing an undisturbed reference test with a subsequent simulator-based test we gain key figures, which reveal problems that only appear if the application is used in scenarios of fragmented attention.

We conducted a test series to evaluate our method and hardware setup. First results gained in these tests show that *state inspections* – frequent glances the user needs to observe the application's state during a system induced latency - are an avoidable cause for mobile usability problems.

INTRODUCTION

The mobile internet has developed from an early adopter's playground to an important part of today's communication landscape. Modern smartphones featuring GPS-navigation, camera, and high speed Internet access are designed towards being used on the move. However, the users cannot concentrate on handling the application without interruption in these situations. The interaction with the device is fragmented into tiny interaction units.

To test the usability of mobile applications being used on the move, we have to test these applications in realistic mobile situations. However, in field tests we must accept high technical effort to observe and evaluate usability. A further disadvantage of field tests is that we can neither control nor easily reproduce environmental conditions. To minimize the impact of random influencing factors we would have to conduct a high amount of redundant test iterations.

In this paper we show a novel approach as alternative to time-consuming and expensive field tests. Using a car simulator we generate a primary task that acts as a substitute for real mobile situations within a controlled laboratory environment. Thus we create the wanted distraction but avoid uncontrollable influences. We expect, that results gained in these tests are more exact and reproducible.

First, we briefly survey previous work. In the subsequent sections, we describe our physical and logical test setup and describe the data we measure and calculate to evaluate mobile usability. Subsequently, we report the results gathered in first tests using the simulator. We explain the typical problem of *state inspections* we recognized in the tested applications, and demonstrate basic approaches how to avoid them. Finally, we present our lessons learned and offer instructions for further tests.

PREVIOUS WORK

In 1999 Kristoffersen and Ljungberg realized that the main problems of mobile computing were not only technical limitations (small low resolution screen, slow internet connection, miniature keyboard,…) but also conceptual problems [7]. Applications and devices were simply not meant to be used on the move. Consequently users had to align the mobile computers with the work situation at hand. A typical example is a driver who has to stop the car to read and answer a message.

In 2000 Hoyoung et al. conducted a survey with 37 participants. They equipped all users with a brand new internet capable mobile phone and encouraged them to use it whenever they liked. They analyzed 1552 user sessions to find relevant contexts in which the mobile internet was used effectively [6]. The authors found out that the participants mainly used the mobile internet devices when they were in a joyful, calm and quiet environment.

Following these results there was little need testing applications in scenarios of fragmented attention in the year 2000. Mobile applications simply were not used that way.

However five years later Zhang et al. reported that testing the usability of mobile applications using traditional laboratory tests did not deliver convincing results. Yet when the authors conducted experiments in the field, lack of control over test persons and the test environment did not allow for reliable results. Zhang et al. concluded that usability test methods that were successful for desktop applications cannot necessarily be transferred to mobile application testing. The question left open in the paper is: How can usability of mobile multimedia applications be evaluated effectively? [15]

In 2007 Looije et al. tested mobile map applications. They realized as well a necessity to test the usability of mobile devices while the user is mobile. This mobility makes evaluation difficult. They agreed with Zhang that a method for realistic usability testing that avoids the problems of field experiments would be desirable. They resumed that there were still a lot of challenges for both design solutions and usability testing of mobile devices. [8]

In 2007 Coursaris and Kim prepared a research agenda for mobile usability. They went through published articles and papers and reported that 58% of all studies they found used laboratory tests to evaluate mobile applications. While 22% of the studies concentrated on field tests, 11% combined both methods to get better results. In 9% the methodology was not released. [1]

In [12] Perry and Hourcade describe their efforts to improve usability for mobile applications running on touch screen devices. They found that due to physical constraints not all positions on screen were equally accessible. In their tests the participants had to touch predefined regions on screen. They measured accuracy and speed and rendered a resulting map that shows preferred and less preferred regions to be used for interaction elements. However the authors have to admit that their tests were conducted in laboratory environment. They assume that when using a mobile application on the busy street, results could be different from their findings.

To fully understand the impact of environmental influence Hummel et al. introduced a framework to monitor environmental disturbances and demonstrated the effects of acceleration (moving), changing light conditions, sound, temperature and humidity on user performance. [4]

Eventually Oulasvirta et al. tried to learn more about real mobile scenarios. One of their most important findings was that applications that are used in scenarios of fragmented attention should limit the time a user needs for an interaction unit. [11] In their work they analyzed a range of different situations and found that even in situations like sitting in a public bus the average duration for the continuous span of attention to the mobile device is only six seconds.

Several works focus on the identification of suitable equipment for laboratory and / or field-based usability tests. In [13] Schusteritsch et al. present the infrastructure used by Google Laboratories for mobile usability testing. In the paper they concentrate on various options for recording the screen and observe interactions.

A professional hardware configuration used in mobile usability field observation is presented by Oulasvirta and Nyyssönen in [10]. The introduced equipment is transportable in two aluminum cases and costs about 10000 Euro.

An important insight must be awarded to Kallio and Kaikkonen. In comparing laboratory and field testing of mobile applications they found that the usability problems that were actually identified in the tested applications were essentially the same regardless what method was used. [5] Nevertheless the impact a given problem has in a mobile scenario is not necessarily the same as when the application is used in a calm and controlled lab environment. Thus lab testing is still meaningful but it should be complemented by additional methods.

In this paper we propose a mobile usability testing method that is one possible answer to Zhang's question on how to effectively evaluate usability of mobile multimedia applications. Following Oulasvirta et al.'s work we believe that the key difference between conventional laboratory tests and realistic field testing of mobile applications is that using the application in the field almost always means that the user is constantly distracted by a primary task while working with the application.

Research Questions

The primary goal of our efforts initially was to find an efficient approach to test the usability of mobile applications in scenarios of fragmented attention.

Thus we focused our research on these questions:

- Which mobile scenario can represent situations, when using the phone is not the only or even primary task?
- How can we effectively observe this scenario and measure the usability of applications used in it?
- How can we control and reproduce environmental conditions in this mobile scenario?
- Which data should be gathered and how can these data be interpreted to evaluate usability?

As an additional benefit we tried to find a method requiring only low budget equipment. Thus the method would be suitable for smaller companies and educational or non-profit organizations.

Test Setup

We discussed a wide range of potential scenarios in work, leisure and travel where users cannot concentrate solely on the mobile application they use. While many real situations would deliver the required distraction, there was no way to control the environmental conditions in these situations. To achieve reliable and reproducible measurements we would therefore need many test iterations thus making the method tedious and expensive.

Referring to ideas already proposed by Duh et al. [2], we finally decided to use a videogame-based car simulator to simulate a pseudo realistic field scenario of fragmented attention. The criteria for the choice where:

- Minor spatial requirements of the hardware setup and very low price for the simulator.
- The scenario can be observed and recorded easily.
- The environmental parameters that influence the primary task's complexity are adjustable and reproducible.
- The primary task needs nearly constant vigilance so only limited attention for handling the application remains. However by configuring different vehicles, routes, and driving conditions, we can adjust this point if required.

Of course some of the mobile applications we tested are quite unlikely to be used by any responsible driver. However the point in using the simulator is not to simulate a super-realistic driving experience. The simulator simply represents one possible task out of a wide range of primary tasks a user might have to concentrate on while using the mobile device.

Nevertheless it is the secondary task – working with the mobile application - we want to observe.

Hardware Setup

The components for the simulator we use in our experiments include a standard game console (Sony Playstation 3) and a car racing game (Need 4 Speed Pro Street). The game supports a practice mode, which doesn't exhibit random incidents, i.e. there are no other random cars on the road and a route is always in the same condition.

For easier and more intuitive feeling we connected a steering wheel and a break/accelerator pedal box (Logitech Driving Force GT) and fixed these on an adjustable car racing game table (Speedblack EVO).

The main objective was to find a setup, in which the testers could use their preferred hand to handle the mobile device at any time during the test.

We recorded the tests using two video cameras. Testers wore a robust wide-angle helmet camera (Camera 1) to record the driver's view. A second camera was positioned next to the TV set pointing at the driver (Camera 2). As the driver's ability to move was limited, we could adjust a close-up of chest and head that allowed us to observe the driver's eye movements. Thus it was easy for us to notice even the slightest glimpse to the phone.

To avoid reflections on the mobile phone's screen we used indirect lightening.

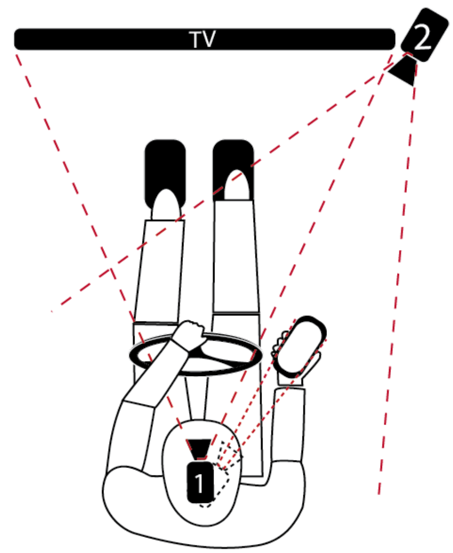

Figure 1. hardware setup for the simulator.

Test Procedure

As soon as the tester is familiar with the simulator, the device, and the test scenario, the measured test run can start.

In order to thoroughly test an application on usability we have to identify use cases first. There is no need to perform mobile usability tests on all use cases within an application. We concentrate only on those use cases that are relevant in scenarios of fragmented attention.

In order to compare resembling applications we identify similar use cases to compare their performance in scenarios of fragmented attention.

The test procedure for a single use case consists of two steps:

The Reference Step:

In the *reference step* the tester must perform the scenario undisturbed (without the simulator) as fast as possible. We record this step with a standard consumer camera. Another possibility is the use of an advanced mobile camera mounted to the mobile phone as suggested by Schusteritsch et al [13].

We use the recorded video to analyze the total time for completing the scenario ($t1_{total}$) and the summarized durations of all system induced delays ($t1_{delays}$) (i.e. user has to wait for response from the device before he can resume the task).

The effective time the user needs to finish the scenario can then be calculated as

$$t1_{eff} = t1_{total} - t1_{delays}$$

The Simulator Step:

In the *simulator step* the participant repeats the same scenario while driving in the simulator. The test manager should stress that neither driving speed nor the total time to accomplish the scenario is relevant. The second advice is necessary to avoid that users try to use the phone when driving situations typically do not allow distractions.

Assuming that camera 2 is positioned correctly, we can observe the participant's eye movements and measure how often ($c2$ … count) and how long ($t2_{eff}$) the tester pays attention to the evaluated application (i.e. looks at the phone).

Please note that we do not subtract delays caused by the system to calculate the effective time since users will typically shift their focus back to the road in these times.

Comparing results:
Using these values we can then calculate the average time for a single attention unit:

$$t2_{avg} = t2_{eff} \ / \ c2$$

and the *deceleration factor* to reveal the correlation between test 1 and test 2:

$$df_{12} = t2_{eff} \ / \ t1_{eff}$$

Deceleration factors above 1 are to be expected although it is well possible to achieve a factor beyond 1 if an application supports handling without looking at the device at all.

Test Run

To evaluate our test setup and procedure we conducted a series of usability tests using four different applications. Before the actual usability tests began, members of the research team evaluated the applications using classical heuristic evaluation as described by Jacob Nielsen in [9].

As a result we had a quite good understanding about the existing usability problems and therefore a basic idea what to expect from the usability tests. Of course we did not yet know the impact these problems would have on our test persons.

In our first series of tests we tested three different navigation software packages: Google Maps Navigation (Android), Navigon and Tom Tom Navigator. While Google Maps is a turn-by-turn GPS application that is currently available for Android OS only, the other two competitors are iPhone apps. The task for all three applications was to search a predefined destination and to start navigation.

Figure 2. Testing of navigational software.

While these three tests can be compared directly, the fourth test was a totally different scenario. Here the participants had to order a custom pizza using the website of a local pizza delivery service. Payment was excluded from the scenario.

Table 1 shows the average measurements we gathered in our tests. Every single test was performed by three different participants. Seven test persons participated in the complete test series. The spread in all test runs was between 6 and 15%.

Application	$t1_{eff}$	$t2_{eff}$	$c2$	$t2_{avg}$	df_{12}
G.Maps	17	29	30	0.97	1.71
Navigon	18	26	25	1.04	1.44
TomTom	20	25.5	24	1.06	1.28
pizzeria	24	41.3	54	0.76	1.72

Table 1. Test Results

Results

Interpretation of Gathered Test Data

By comparing the effective times in the columns $t1_{eff}$ *and* $t2_{eff}$, respectively for the navigation systems we found that Google Maps was the fastest software when used in an undisturbed environment. However in the simulator df_{12} = 1.71 indicates that using the software on the move is more complex than the usage of specialized car navigation software (df_{12} = 1.44 / 1.28).

We assume, that the input method of the destination address can be a cause of this difference. Google Maps accepts the complete address in a single input field in various formats. In Navigon and Tom Tom the user has to input town or zip code first, then street and finally the street number. The input procedure is separated into three screens and supported by showing a list of possible items (e.g. streets in the chosen town) that are filtered continuously while typing. Although the user types in nearly the same amount of data, splitting up the scenario into smaller parts facilitates the mobile scenario. In [11] Oulasvirta et al. suggest that any single interaction unit should take no longer than five seconds. In our tests we found out, that on the road the single 'attention unit' (the duration a user 'dares' to look at the phone while driving – we again want to stress that we do not recommend trying that in a real car!) was never longer than two seconds. Thus a 5 sec. interaction unit can be accomplished within three 'glances'.

Yet the average time for single attention ($t2_{avg}$) is only about one second for the navigation software tests. Especially the pizza ordering scenario exhibits a very low value ($t2_{avg}$ = 0.76). At this point we must add that based on the results of the preliminary heuristic evaluations we already expected challenges with the pizza ordering scenario. The website seems to be too complex to be used while driving. Thus we expected a high deceleration factor. Our experiments approved that fact.

Interpretation of the Intermediate Results and Additional Experiments

One question left is, why the average time of a single attention decreases in complex scenarios? To answer this question, we took a second look on the durations of single attention units and found that they fell roughly into two classes:

- numerous short attention units of 0.1 to 0.4 seconds
- units of 1.5 seconds or longer

By reviewing the recordings of camera 1 (the helmet cam) we found out that the observed durations correspond to two different *attention unit types* we termed *productive interactions* and *state inspections.*

Productive interactions refer to attention units where the work is done. The user reads the content on the screen, decides what to do next, and interacts with the application.

State inspections on the other hand are very short glances the user needs for mainly two reasons:

- While performing the task, system immanent latency occurs. That, in itself, might be unavoidable however the application neither indicates the expected duration nor gives any audible, tactile, or even clear visual feedback when it becomes available again. Thus the user has to check frequently for the system becoming responsive again. This is a typical problem of web applications when a new page is loaded.
- In a highly stressful scenario like driving a car, user minimize the time they spend in attention units as much as possible. They observe the screen to understand the next step (that is the productive interaction). The subsequent action (tapping the screen, 'clicking' a button or link) is often done while the user already focuses on the primary task again. The action itself (moving the finger, touching the screen) is accomplished 'blindly'. However if the hit-target on screen is too small the user unconsciously looks back at the device to double-check if s/he effectively triggered the action (that is the *state inspection*). The situation becomes even worse when triggering the action (e.g. 'clicking' a link in a web application) causes no immediate visual feedback.

It seems obvious that audible and/or tactile feedback on keypress decreases the amount of *state inspections*. However this only holds true if the user does in fact find the target blindly and if s/he feels certain that s/he got the correct interaction element – and not one nearby.

This raises the question how a user interface must be designed, so that users will blindly find touchable elements without problems. We examined that question by conducting an additional limited set of tests with and without using our simulator. This time five participants had to utilize a simple test application we developed for that purpose. (see Figure 3). The program shows a fullscreen table with configurable dimensions. On start the application lights up ten random cells in a row that must be touched by the user. The application does not only respond to correct hits but to any touch on the surface and counts the number of hits and misses. After each touch a new random cell is lit. The testers were not instructed to touch blindly – however to work expeditiously (but not necessarily in full speed).

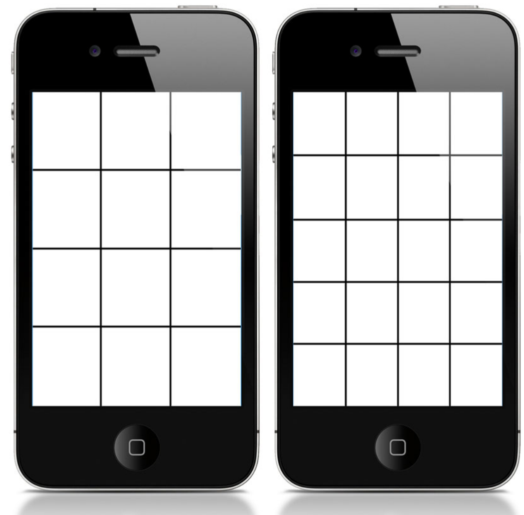

Figure 3. Finding the optimal grid for blind typing

Without being distracted by a primary task (i.e. not using the simulator) all five test persons scored without failure using a 5*4 grid (each test was done twice by all five testers). Indeed many touch screen devices do use this resolution (e.g. the iPhone's home screen shows a grid of 5*4 application icons).

Yet when we repeated the tests using the simulator we got a very different picture. This time only 68% hits were achieved and we could observe the users sometimes obviously inadvertently double-checking again – even though they knew that the application would not give any visual feedback to show if the target was hit. When we configured the software to show a 4*3 grid, the hit rate increased to 93%.

Consequently we suggest that applications used in scenarios of fragmented attention should divide the screen into cells that can easily and certainly be touched blindly. Only one interaction element may be placed into each cell. Based on the perceptions we got from our tests we suggest that an *interaction grid* of 4*3 cells should work well for most users.

In [12] Perry and Hourcade describe a comparable test. In contrast to our application, their targets were not equally spread on screen. They positioned targets in corners and next to borders closer to each other. Perry and Hourcade could demonstrate that the success rate to hit the desired target as well as the speed is not only dependent on the target's size but also on the position on screen.

In contrast to our study, in which we used an Apple iPhone 4, they used devices with palpable screen borders The iPhone features a glassy front plate where the screen edges are not tangible. Additionally, Perry and Hourcade's experiments differ from our tests in test design and object of study. Yet they both show that appropriate target arrangement is an important factor in designing usable mobile interfaces. Additional experiments will be necessary to fully understand and evaluate this topic for scenarios of fragmented attention.

Limitations of Tests and Method

All tests were conducted using devices featuring a touch screen only (Apple iPhone 4 and HTC Desire (Android)). We assume, that our findings will not be reproducible with other devices. Naturally devices with hardware keyboard do have an advantage. However we do believe that there is a strong trend to devices without hardware keyboard although some devices feature an additional slide-out keyboard.

Please keep in mind that driving a car simulator is not the same as driving a car. Testing an application successfully in the simulator does not guarantee that it is in fact appropriate for in-car use. Test persons will most likely 'dare' much more in a simulator than would be acceptable while driving.

Simulator-based tests will give you an indication of possible constraints the application will face in scenarios of fragmented attention. The key figures we introduced will tell you that something is wrong – but not exactly what. Therefore we must also analyze the recordings qualitatively and discuss them with the participants directly after the test to get a better understanding.

The hardware setup we introduced is not suitable for detailed screen capturing. We believe that simulator tests should be an addition to conventional lab tests and hence detailed screen recordings are not necessary. Should a problem require screen recordings, one of the various setups and devices used by Google's user experience research team in mobile device observation should work [13]. Since the user handles the mobile device within a limited area, an additional stationary high resolution camera pointing on the device from atop should be an appropriate solution.

Lessons Learned

Necessary Preparations for Participants:

Before starting the actual test runs it is important to make sure that participants are accustomed with the environment used in the tests. We eventually found the following sequence of tasks to be effective for successful tests:

- Test persons need time to learn how to use the mobile device. Ideally participants use their own device. We found out, that test results can be misleading if users are not familiar with exactly the device used in the test, i.e. owners of an Apple iPhone did not find the hardware-back-button on Android devices when browsing the web. As a result they were not able to cancel an incorrect 'click'.
- Test persons must be familiar with the car simulator, although we found out that test results are not influenced by the skill of the drivers. Experienced gamers might drive faster but performance/speed is not a measured criterion for evaluating the usability of the mobile application. We suggest that participants should even train on the track that will be used later in the tests to minimize the effect of practice during a series of tests.
- Finally the participant should understand and try out the mobile scenario s/he will perform in the subsequent tests. This is necessary because otherwise trying to understand the task would decrease speed in the reference step. There is no time limit and no rush in completing the scenario in the preparation.

Adapting the Environmental Parameters

In mobile scenarios users are confronted with a range of disturbing influences. In [4] Hummel et al. introduced a framework to monitor environmental disturbances and demonstrated the effects of acceleration, light conditions, sound, temperature and humidity on user performance. Simulating special environments (very loud / cold, … environment) might be beneficial if the application in question will be used in these contexts.

Timing:

Getting the count and durations of all attention units can be done by analyzing the recording of camera 1. However doing this frame by frame is a lengthy and tedious job. To speed up that task we developed a small iPhone application that will help you time the attention units in real time during watching the video (Figure 4). You can use the application to time and count two separate rows of events (Timer A & Timer B). The timing starts at the Touch-Down event and stops at each Touch-Up event. The open-source software is available for free at https://github.com/GrischaSchmiedl/Multitimer. Effectively using the software demands some practice to get accurate and reproducible timings.

Figure 4. MultiTimer Application

Simulating Less Demanding Tasks:
Using an application while driving is probably (and hopefully) a rare situation. Not all scenarios of fragmented attention will necessarily be that demanding. In [11] Oulasvirta et al. analyzed several mobile situations and environments and found that for a given task the average *duration of continuous attention* (aka an attention unit) ranged from four (walking in a busy street) to eight seconds (working in a metro car). In our simulator scenarios the average duration for an attention unit was only about one second. Still the simulator can be used to simulate the desired *stress level*. The challenge is to find a game that will meet the criteria:

- The user's preferred hand stays free for handling the mobile device
- The game demands continuous attention from the user (you cannot simply stop) although some level (the desired time span) of inattentiveness is compatible with playing. We believe that civilian (non action) flight simulator software might be an option.

Omit Faulty Test Runs:
Not all performed tests should be analyzed quantitatively. If the participant was unsatisfied with her/his driving performance, the car crashed in the middle of the scenario or the participant lost track in the use case (e.g. by following a wrong link in a web application) the measured data is not valid any more. The test run should then be omitted. In our tests about 40% of all test runs had to be repeated.

Conclusion

In this paper we demonstrated how to use a car simulator to test the usability of mobile applications in scenarios where the user's attention is divided in using the application in question and a primary task (e.g. driving a car). In comparison to traditional undisturbed laboratory tests our procedure poses the test person in a situation closer to real mobile situations. However, in difference to field tests our environmental conditions are controllable and test results are therefore reproducible. An additional benefit of our solution is the very reasonable price.

We introduced how to gather relevant data in our two step method. By comparing measured values from both test steps we calculated the *deceleration factor* as well as the *average time for a single attention unit* in the simulator step. These values combined will help you to identify possible usability problems that appear in scenarios of fragmented attention. Additionally our test method can be used to estimate the impact of know usability problems in mobile situations.

By interpreting the results of our first test series we found out, that some use cases and applications showed a small value for the average time for a single attention unit. We analyzed the recorded videos and observed, that frequent *state inspections* were responsible for many short glances. These *state inspections* however are avoidable. Using a reduced interaction grid of 4*3 cells with an iPhone, we got better results in the simulator than with a 5*4 grid as it is used on the iPhone's home screen. Clear visual, audible and / or tactile feedback is supportive as well.

We still recommend to perform traditional usability evaluation methods (lab-based usability tests, heuristic evaluation, comparing against guidelines, thinking-aloud tests) for mobile applications. However we believe that simulator-based usability tests are an effective method that will produce additional information about the usability of mobile applications in a mobile world.

Future Work

Using the simulator we are now ready to test for *real mobile usability*. However we need additional information on factors that make applications usable in scenarios of fragmented attention. Finding the optimal grid-size certainly is a starting point but many options are not yet fully explored.

The optimal design of the user interface depends on the usage situation of the application. Future applications might be context-aware and adapt the user interface based on the mobile scenario at hand.

In future work we plan to concentrate on suitable audible and tactile feedback to support mobile scenarios. In 2010 Yu et al. showed, that a sonically-enhanced menu interface will reduce task completion time and minimize the need of the user's cognitive resources – especially in mobile situations [14]. Auditory icons are widely used in GUIs of desktop operating systems (e.g. the sound that is played if a file is deleted) [3]. But still additional work will be necessary to find ideal auditory and *tactile icons* for mobile applications:

- We think that mobile applications should accompany the end of system induced delays by a self-evident sound.
- On-screen buttons should give tangible response that feels similar like pushing a physical button.

Coming back to the simulator we plan to further investigate in alternative setups (i.e. other games) to simulate less demanding environments. Thereby we plan to reconstruct a wider range of typical mobile situations as described in [11].

REFERENCES

1. Coursaris, C.K. and Kim, D. A research agenda for mobile usability. *Ext. Abstracts CHI 2007*, ACM Press (2007), 2345–2350.
2. Duh, H.B., Tan, G.C., and Chen, V.H. Usability evaluation for mobile device: a comparison of laboratory and field tests. *Proc. Mobile HCI 2006*, ACM Press (2006), 181–186.
3. Helander, M. *Handbook of human-computer interaction*. Elsevier, Amsterdam, New York, 1997.
4. Hummel, K.A., Hess, A., and Grill, T. Environmental context sensing for usability evaluation in mobile HCI by means of small wireless sensor networks. *Proc. iiWAS 2008*, ACM Press (2008), 302–306.
5. Kallio, T. and Kaikkonen, A. Usability testing of mobile applications: A comparison between laboratory and field testing. *Journal of Usability Studies 1*, (2005), 4–16.
6. Kim, H., Kim, J., Lee, Y., Chae, M., and Choi, Y. An Empirical Study of the Use Contexts and Usability Problems in Mobile Internet. *Proc. HICSS 2002,* IEEE Computer (2002).
7. Kristoffersen, S. and Ljungberg, F. "Making place" to make IT work: empirical explorations of HCI for mobile CSCW. *Proc. SIGGROUP 1999*, ACM Press (1999), 276-285.
8. Looije, R., te Brake, G.M., and Neerincx, M.A. Usability engineering for mobile maps. *Proc. Mobility '07,* ACM Press (2007), 532–539.
9. Nielsen, J. *Usability inspection methods*. Wiley, New York, 1994.
10. Oulasvirta, A. and Nyyssönen, T. Flexible Hardware Configurations for Studying Mobile Usability. *Journal of Usability Studies 4*, 2 (2009), 93-105.
11. Oulasvirta, A., Tamminen, S., Roto, V., and Kuorelahti, J. Interaction in 4-second bursts: the fragmented nature of attentional resources in mobile HCI. *Proc. SIGCHI 2005*, ACM Press (2005), 919–928.
12. Perry, K.B. and Hourcade, J.P. Evaluating one handed thumb tapping on mobile touchscreen devices. *Proc. GI 2008*, ACM Press (2008), 57-64.
13. Schusteritsch, R., Wei, C.Y., and LaRosa, M. Towards the perfect infrastructure for usability testing on mobile devices. *Ext. Abstracts CHI 2007*, ACM Press (2007), 1839–1844.
14. Yu, Y. and Liu, Z. Improving the performance and usability for visual menu interface on mobile computers. *Proc. AVI 2010*, ACM Press (2010), 369–372.
15. Zhang, D. and Adipat, B. Challenges, Methodologies, and Issues in the Usability Testing of Mobile Applications. *International Journal of Human Computer Interaction 18*, 3 (2005), 293-308.