1-1-2015

Using the Wii remote for mobile device application testing: a proof-of-concept

Mark Freeman
University of Wollongong, mfreeman@uow.edu.au

Using the Wii remote for mobile device application testing: a proof-of-concept

Abstract

There has been a dramatic shift in the interaction methods of mobile devices over the past decade. From devices simply being able to make phone calls to being able to handle complex tasks traditionally performed on personal computers PCs; this change has led to new interaction issues that need to be understood during the application development process, particularly as these devices now commonly incorporate a touch-screen as their primary source of input. Currently, the methods of conducting software user experience testing of these devices employs techniques that were developed for PCs, however mobile devices are used within different contexts of use. This research initially reviews the current methods for user experience testing of applications running on mobile devices and then presents, through a proof-of-concept approach, an innovative method for conducting user experience testing employing actual devices.

Disciplines

Engineering | Science and Technology Studies

Publication Details

Freeman, M. Bruce. (2015). Using the Wii Remote for mobile device application testing: a proof-of-concept. International Journal of Handheld Computing Research, 6 (1), 34-43.

Using the Wii Remote for Mobile Device Application Testing:
A Proof-of-Concept

Mark Bruce Freeman, University of Wollongong, Wollongong, Australia

ABSTRACT

There has been a dramatic shift in the interaction methods of mobile devices over the past decade. From devices simply being able to make phone calls to being able to handle complex tasks traditionally performed on personal computers (PCs); this change has led to new interaction issues that need to be understood during the application development process, particularly as these devices now commonly incorporate a touch-screen as their primary source of input. Currently, the methods of conducting software user experience testing of these devices employs techniques that were developed for PCs, however mobile devices are used within different contexts of use. This research initially reviews the current methods for user experience testing of applications running on mobile devices and then presents, through a proof-of-concept approach, an innovative method for conducting user experience testing employing actual devices.

Keywords: *Human-Computer Interaction, Mobile Devices, Proof-of-Concept, Usability, User Experience*

INTRODUCTION

With the dramatic increase in the adoption of mobile devices (including smartphones and tablets) methods for developers to conduct user experience testing of applications for these devices has been somewhat of an ad-hoc process. In 2007 Apple Ltd. released the iPhone, a touch-screen smartphone; this device introduced a new focus on small, lightweight and inexpensive portable computers and introduced an influx of touch-screen mobile devices to the market. Previous devices ran variations of Operating Systems (OSs) that were designed for traditional personal computers (PCs), however by Apple using an OS designed specifically for their phones, the industry adapted. Currently smartphones have two major competing OSs: Apple's iOS and Google's Android. These OSs introduced a new class of user interfaces that are not like those of traditional PCs, with support for touch-screens, movement from traditional Graphical User Interfaces (GUIs) to Natural User Interfaces (NUIs) has occurred. NUIs have been also popularized in the gaming industry with the introduction of the Nintendo Wii Remote, Sony Move and Microsoft Kinect. Both of these distinct types of devices (mobile and gaming) offer the user a unique experience beyond the traditional Windows, Icon, Menu, Pointer (WIMP) that they are use to since the creation of the GUI experience based on an office desk from the 1970s.

DOI: 10.4018/IJHCR.2015010103

There are a number of categories of touch-screen devices that could be focused on with this shift to NUIs and their associated new user experience paradigms during application development processes; Tilvawala et al. (2011) categorized these types of devices into three main groups: smartphones, pads (also referred to as tablets), and boards (custom display and interaction devices, typically used in education and workplace settings). The focus of this paper will be on smartphones as these are the most commonly used devices and have the added complexity of being mobile. These devices aim to bring the whole PC experience (and potentially even more) to users whilst they are on the move. Phone manufacturing companies believed that users would be happy doing what they can currently do on PCs on much smaller handheld-devices, because they can move with them, especially for in youth market (Sharp et al. 2006). The success and popularity of both smart phones and tablets demonstrates that users see the benefits in carrying these devices around; with the ability to access their data, utilities and entertainment from their device.

The motivations of the user are of utmost importance when understating their experience of an application should be conducted. These devices are not used in isolation and potentially are not used when the user has the ability to provide complete cognitive focus on the task at hand. When users are on the move, their movement has the potential to interrupt their interaction with the device. Within a changing environment there is the potential for distraction beyond the traditional device-user dichotomy that has been typically studied when conducting application testing (Sharp et al., 2006). Prior studies have shown that working and background noise decrease the recognition error rate of speech-based text entry on the mobile devices (Goldman et al., 2006) a common feature presented in the marketing materials of smart phones and tablets. Therefore, environmental factors cannot be ignored in the testing of applications designed for mobile devices. However, traditional methods of application testing are commonly carried out in lab-based environments with fixed conditions. Employing methods designed to test traditional PC based applications with GUI interfaces. For example, logging systems incorporating the recording of software interactions are suggested by some of the leading software developers in the industry. However, these applications focus on the logging of keystrokes and mouse movements that are not directly relevant to mobile touch-screen devices (TechSmith Corporation, 2009b). What needs to be of importance when testing mobile touch-screen devices is the focus on the entire device that the user interacts with beyond just the keyboard clicks and mouse movements. Currently there is no easy way to log this; software falls into two main categories software that can record the videos of users using the device (TechSmith Corporation, 2009a) from particular point of view through a camera or software directly recording the device screen changes. Recent advancements in smartphones have resulted in using the inbuilt mirroring feature within products, such as RefectorApp and AirParrot (for Apple iOS devices) and an inbuilt feature in the latest version of Android; that can capture changes in the screen. These products are ideal for creating demonstrations of application features however as they do not allow understanding of the user experience and their interactions with the interface (that is what they actually pressed on or meant to be pressed) or the user (through using the device part of the screen is covered) vital information is missing for user experience and usability testing purposes. These applications also only capture the screen of the device not the overall interaction with the device itself.

The majority of methods used for the evaluation mobile touch-screen devices by developers, being based on the analysis of recorded videos of the devices' screen. From an evaluator's perspective, it takes a large amount of time to watch all the videos. In addition, it is also hard to tell from the videos if it is the participant completed an unexpected operation, or if it was an unexpected behaviour of the tested subject. One option available to application developers is through the use of simulators. The problem is the environment for testing the application is not on the real device, this leads to different experiences for users when they actually use the application on a real device. Since the users are not able to interact with the screen and the device during their evaluation. These tacit interactions perform part of the overall experience of using a touch-screen device.

One Important distinction with current mobile devices is that there are a number of levels of interaction that can occur. Currently, users are not only are faced with the interface of the actual device itself, but with an increasing number of third party applications that are designed to be used on the devices (available through the OS online stores e.g. iTunes for iOS and the Google Market for An-droid) and also interfaces through web browsers on the devices. If focus on a smart phone the device can still be used for basic cell phone functions, such as making a phone call, storing a contact's details and sending a text message, however they can now be used to browse the web, read and edit documents and play games. The change in the way that these devices are being used has led to what Janlert and Stolterman (2010) refer to as complex interaction. This leads to the research question of: *how can the complexities of touch screen mobile devices be further understood when developing applications and conducting user experience and usability testing to see if they meet the user's needs?*

This research will initially present the current methods and tools used in evaluating mobile device applications, and propose, through proof-of-concept analysis, a new method along with the appropriate tool for the evaluation of small touch-screen devices to help developers collect a greater amount of data during the testing phase of software development and to increase user experience of the final application developed.

BACKGROUND LITERATURE

Potentially, developers and users have a different understanding on what they expect from the same product or piece of software. In traditional environments developers were the ones who attempted to solve the problem where users were the ones who used the product. There has been a shift in this world-view with the introduction of paradigms such as user-involved or user-centred development (Damodaran 1996; Gulliksen et al., 2003). However, for most users they just want know what are the key features of their applications and how can they use them to achieve their goals. Therefore the usability of applications has become a basic determinant of the acceptance of the system (Abran et al., 2003). The term "usability" is defined differently in different contexts (Heo et al., 2009), Abran et al. (2003) suggest that usability should refer to the effectiveness, efficiency, satisfaction, security and learnability of the product. Usability testing can been seen as a process for providing feedback on a developers' work (Polkosky, 2010), as such testing of applications on mobile devices during their development is vital for application success.

Traditional mobile phones, had one major limitation over current touch-screen technologies, a constrained input interface (Cockburn & Gutwin, 2010) consisting of the number-pad and a few other keys to move between functions on the device. Cockburn and Gutwin (2010) argued that "in many interaction contexts direct manipulation can be awkward, cumbersome, or impossible to use" (p. 132) this contradicts the current trend in mobile touchscreen device technologies with the touchscreen input as the primary method of interaction with the device. Mobile devices have wide variations in form factor, input modes, processing power, battery life and screen size (Adipat & Zhang, 2005; Gonsalves & Ryan, 2005). Unlike PCs, mobile devices are also capable of sensing their surrounding environment including gravity, acceleration, brightness, location and speed through the use of internal accelerators, sensors and Global Positioning Systems (GPS). These additional features found in mobile devices, although increasing their complexity, add means for an increased user experience in various situations. Gonsalves and Ryan (2005) define the activities of users' using mobile devices as interaction between users, applications (via the device), and the physical environment making the devices pervasive. Environmental factors play an important role in ensuring that these types of mobile devices are usable in all contexts, with the methods of interaction not fixed to one particular location. These considerations need to be understood when developing applications for these devices and should be evaluated accordingly during the development process.

Current Testing Methods and Tools

To identify what is needed in the process of analysing mobile touchscreen devices, traditional methods used in user experience and interaction testing for PCs were reviewed. The goal of the evaluation is to provide feedback of the tested subject to support the development process (Rosson & Carroll, 2002) or to find the problems of a prototype and to check if a target user group can use the product effectively or efficiently (Sharp et al., 2006). Typical user interaction methods are observations, interviews, and questionnaires (Sharp et al., 2006). In the testing phase of application development, a combination of methods can be used depending on whether the aim of the test to collect quantitative and qualitative data for further analysis and evaluation.

Qualitative data sources such as interviewers, user and observation notes and questionnaires have the potential to allow developers to gain understand of the how and what regarding user feelings of an application. Quantitative data allows consistent and detailed analysis across different tested users (Preece, 1994) and can be from the time taken to complete tasks and the number of errors. Interviews and questionnaires are highly dependent on the goals of the evaluation. These techniques are used to gather the information that cannot be collected by observation such as a user's background, feelings and opinions. With interviews and questionnaires, the evaluator gets mostly qualitative data. To ensure the validity and reliability of the result, data analysis and statistical methods are applied (Lazar et al., 2010).

Observation of users without disturbing their activity during the data gathering process needs to be a goal of conducting testing, to allow 'real-world' experiences. Computer systems can capture both quantitative and qualitative data, and analysis the information automatically (or semi-automatically). Current software such as TechSmith's Morae is able to record, and analyse users activities, including key presses and mouse movements, to generate reports to help the evaluator in making meaning of how a user interacted with the system (TechSmith Corporation 2009b). However, the qualitative data such as screenshots, video recording of users' facial expressions and audio recording of "think-aloud" needs to be reviewed and analysed by experts as well (a semi-autonomous process). While this is appropriate for PC based evaluations when it comes to mobile touch-screen devices they do not have the power to compute the analysis activities, thus expert involvement is currently required.

Context of Using Mobile Touch-Screen Devices

To understand the differences of testing mobile touchscreen devices, the context of using them should be identified to understand the interaction between the user and their mobile device. Uther (2002) stated that some of traditional usability guidelines relating to navigation, structure and error prevention can be applied to evaluate mobile devices, but contextual factors also need to be considered. Gonsalves and Ryan (2005) define the activities of users' using mobile devices as interaction between the users, the application (via the device), and the physical environment.

The majority of current mobile devices in the market use a touchscreen as the main input/output method. This allows the overall size of the device as small as possible, whilst having the largest display for the user to interact with. However, there are issues regarding touchscreens, users' cover part of the screen while they are using the device, so they are not be able to see the blocked part of the screen if changes occur to the presentation of information (Robbins, 2010). Secondly, the size and location of touchable elements such as keys becomes significant, as application developers have access to the entire screen. Previous studies suggested that 'touch' keys should be larger than 9.2 mm to be usable and for reasonably accurate information entry to occur (Parhi et al., 2006; Perry & Hourcade, 2008). Park and Han (2010) stated that users' tend to press small touch keys more carefully and slower than touching larger keys, and the will use different techniques or parts of fingers to do so. In addition, how a user holds the device needs be considered as a factor that affects the user experience.

With current mobile devices using touchscreens as the main input method; it is impossible to analyze the recording of operations directly on the device, therefore the best way is to build a system for this process outside of the device. A user's fingers are used to control the touch-screen, which also indicate the interaction with devices. The operations completed by the fingers can theoretically be treated as cursors on computers. The touches (on devices) are clicks (on PCs) and slides (on devices) are drags (on PCs). Therefore, as an initial proof-of-concept operation data can be recorded and analyzed through the use of current usability evaluation software, such as TechSmith's Morae.

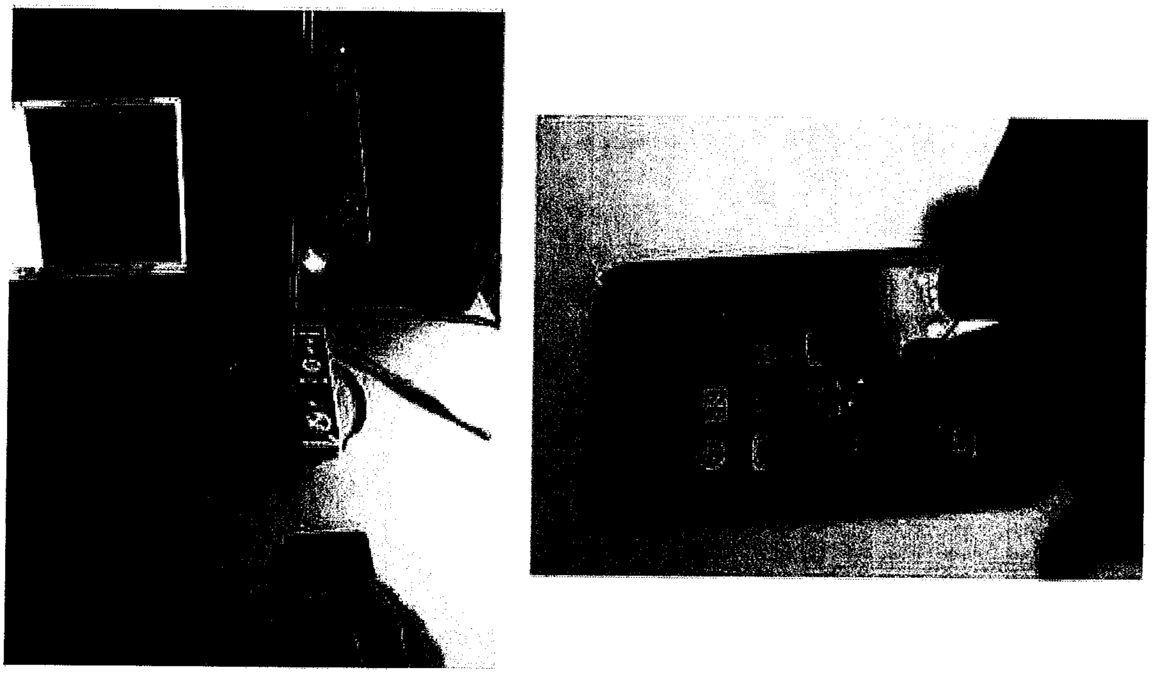
PROOF-OF-CONCEPT

To record the operations of the user of a mobile device, a finger tracking system needs to be able to recognize any movement in a small space. This should include three-dimensional (3D) movement of fingers above the screen; with the ability of a mouse-button style feature so the screen will not be 'clicked' (pressed) unless the finger actually touches the screen (or makes the screen react in some way). In addition, a finger tracking system needs to be as small as possible (in its final marketable state) so it can be used to evaluate mobile device interactions in real-world contexts.

In designing this proof-of-concept, a simple solution of tracking marked objects was created by using the camera of Wii Remote, which has a 128×96 monochrome cam-era with built-in image processing and an infrared (IR) filter, this device provides 1024×768 resolution for up to four tracked points using 8x sub-pixel analysis. Wii Remotes are also Bluetooth devices so they do not need to be physically connected to a computer to capture the images. Since the Wii Remote camera reports locations and sizes of tracked 'dots' rather than the original image, it can also referred to as an IR sensor. However, a major issue is that the IR sensor can only track an object in a two-dimensional (2D) plane. To track in a 3D space, an additional sensor is needed, through a process of combining the data from two sensors; 3D coordinates can be determined. For this to occur both IR sensors must see the area of the screen of the device. Another issue is that the IR sensor only responds to inferred rays, the tracked objects need to be marked by infrared sources or inferred-reflective materials (markers).

The structure designed for this proof-of-concept experiment, places the two IR sensors (Wii Remotes) vertically above each other, so a paired set of data from two sensors based on the horizontal location of the dots (see Figure 1 (a)) with sensor 1 providing an overview of the screen and sensor 2 identifying when the finger is touching the screen. In this proof-of-concept 5mm IR LEDs (using a small 1.5v watch battery directly attached) were attached to the finger, to prove that they could be tracked (see Figure 1 (b)). The issue with this setup if it was expanded beyond the proof-of-concept phase would be that this setup is potentially distracting for the tester due to the size and weight of the LED and battery.

Figure 1. (a) Proof-of-concept structure setup, (b) close up of the IR LED and battery



Feasibility Testing

For the feasibility test, the IR camera was monitored using Wiimote Whiteboard for Mac (Schmidt 2008). This application has the ability to connect to two Wii Remotes, and display the location of four tracked dots for each remote in a monitor window on an Apple Mac. This test was used to examine if the two sensors placed in the structure described above receive IR readings from the markers concurrently. The markers were evaluated by the stability of the tracked dots in the monitor window with the result demonstrating that the dots were clear and stable in the window.

For the feasibility test, the Wii Remotes were located vertically approximately 20cm apart. The sensors received the inferred from the LED as long as the finger was moving along the screen and the screen was located inside the overlapping view of both sensors. There was noise (unexpected tracked dots) shown in the monitor, however they disappeared when moving the touch-screen away from view, the noise could be a result of a small amount of reflected inferred from the devices screen. To avoid the noise, the sensitivity of sensors needed to be adjusted for the following experiments. The distances of the sensors to the screen between sensor 1 and sensor 2 were different (the sensor 2 is closer to the screen than sensor 1), which meant that the intensity of the inferred received by the sensors was be different, the sensitivity of the sensors needed to be modified in the tracking program developed.

Software Development

It is impossible to calculate exactly where a user touched the screen unless it was directly recorded from the location of the device. The location from the device is not the exact location as a user uses their finger to touch the touchscreen, and the contact surface of the screen and finger is not as small as the dot that is tracked by the sensor. Therefore, the calculated location is acceptable if it is within the area of touched key on the screen. Research has found that the average index finger is 19mm across (Parhi et al., 2006) and this entire surface has the potential to be registered on the touchscreen.

In this proof-of-concept, the marker, sensors, and the calculation of the 3D coordinate are mainly used to generate the deviation of the coordinate. The marker does not indicate the location of the finger. There is a small offset, which is the distance of IR source to the LED to the top of the finger. Since the marker is used during the calibration process (gathering information of what the 3D space is like to setup the 3D coordinate system) and tracking, the offset will be in the calculation displayed in the application. The location of markers in the coordinate system of the marker is calculated rather than the location of fingers and where they actually touch the screen.

The data produced from the Wii Remote cameras are two unsigned integers that represent the location of tracked dot in a 2D view; the upper-left corner of the view is the origin, (0, 0). With the 3D coordinate system developed in this proof-of-concept the unsigned integers actually represent different lengths, with the ratio of the units between coordinate systems needs to be used in the calculation, as different devices could be used with the developed system. Since the sensors are in different positions, the views of the sensors are different and they perceive different screen dimensions. Therefore, each time the system is used a calibrating process is required to analyse the space above the screen to indicate the 3D coordinates of the current device being evaluated and the exact location of the IR sensor on the users finger.

The solution developed allowed for the easiest way to calculate a random dot X in the 3D coordinate system based on the data from the sensors. In the solution, it is assumed that the images of the sensors are parallel to the top edge of the screen in which the shapes of screen in the image of sensor 1 are changed least. Sensor 2 is placed in the same level with the device, so it only sees only the width of the screen (as shown in Figure 1 (a)).

The proof-of-concept was developed to track a single finger, and the record the movements and presses of a finger on the touch-screen as a cursor on a PC to enable screen-capture and testing software such as Morae to record movement and presses, to reduce the time evaluating test data of mobile device interaction. The software was developed using an open source C/C++ library to access the data of the Wii Remotes connected to a computer. The software contains three main processes, connecting to Wii Remotes, calibration, and tracking the user's finger. Before running the software, the Wii Remotes should be connected to computer via Bluetooth by adding new Bluetooth devices in the control panel. After connecting to the remotes, the software will be able to communicate with the Wii remotes in a similar way to other Bluetooth devices.

During the calibration process, the resolution of the screen is typed into the system. The developed system then requires the user touch the corners of the screen with the marked finger in a clockwise direction and hold it still for at least 3 seconds at each corner, as the IR LED is always on, this can prevent the system from recording data when the finger is moving [corner A: ($xA1$, $yA1$), ($xA2$, $yA2$); corner B: ($xB1$, $yB1$), ($xB2$, $yB2$); corner C: ($xC1$, $yC1$), ($xC2$, $yC2$); corner D: ($xD1$, $yD1$), ($xD2$, $yD2$)] with a slant angle of $\alpha 1$ and $\alpha 2$]. Finally the resolution of the screen is entered into the system in pixels for the overall screen width and length. Based this data and the coordinate of dot A in sensor 1 and sensor 2, ($x1$, $y1$) and ($x2$, $y2$).

Values of dot A are used to calculate the changing value of the coordinate ($\Delta xs1$, $\Delta ys1$, and $\Delta ys2$), because the dot A is set as the origin of the 3D coordinate system. The angle α is the absolute difference of slant angle of two Wii Remotes, which avoids the possibility of the screen not being horizontal. The system declares the finger touching the screen when the z value of the dot is less than a constant, because the z value is changing between certain ranges while touching the screen by different parts of the finger. For the proof-of-concept the constant was set to 20 in the software with a screen resolution of 480×320, but this could be different in other situations.

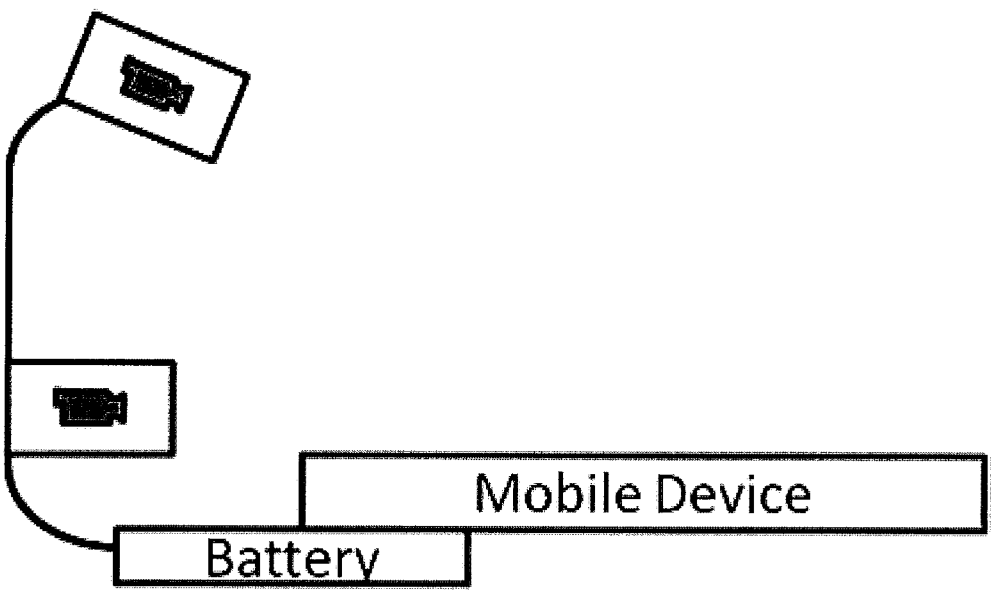
Proof-of-Concept Review

The single finger tracking system designed in this proof-of-concept can record the movement of finger into operations typically performed by a mouse that then can be logged. Thus, as a proof-of-concept the design of the system was successful. One identified issue is that current testing software does not support recording operations from more than one mouse, which is an issue as mobile devices on the market are multi-touch. Also, with only capturing one finger, the current system is too simple to detect gesture operations on multi-touch devices. Therefore, new methods of analysing the input by a user would need to be developed in a commercial version of the product.

The accuracy is quite low and may not high enough for professional usability testing, as the system can only report where the finger approximately interacts with the screen, this could be increased through the use of the screen capture software that has recently become available on mobile devices. There are some changes could be made to improve the accuracy of the proof-of-concept by using modified sensors that are suitable for close-up monitoring, changing the way of marking fingers that reduces the offset of fingers and markers and the development of another solution for the calculation, the currently calculation brings the largest inaccuracy. However, as a proof-of-concept this application demonstrates that there is potential for using IR tracking technologies (such as those found in a Wii Remote) for improving testing of mobile devices.

In the experiment, Wii Remotes were used for the proof-of-concept. However, the size of these remotes is too big to be employed in a commercial system. Since only the IR sensor/camera is used in the system, a device that is reasonably small, the system would be much smaller. Two sensors could be attached on an adjustable arm with method of attaching it to the mobile device (see Figure 2). Future versions of the system have the potential for adding a camera to record the facial expressions of the user.

Figure 2. Potential version of product designed from the proof-of-concept



CONCLUSION

Unlike traditional PCs, the context of using mobile touchscreen devices is varying with contextual factors playing a large role in affecting the devices usability. The lack of tools for testing and evaluation makes the analysis of applications developed for these devices long and ineffective. This paper has presented a proof-of-concept for a finger tracking system to help evaluators of applications designed for mobile touchscreen devices. This proves the possibility of using the IR sensors of Wii Remotes to record the operations on the touchscreen, however the current size and accuracy of the system is not suitable for evaluation in commercial settings, but is appropriate in research environments and for further development of the proof-of-concept. Further research needs to be conducted to solve the issues raised in the development of the current system. In its final state, the system could be portable and have the ability to record all needed data including video, audio, operations, and motions in or out of the lab. This automation process would reduce the testing time of applications developed for mobile touchscreen devices.

REFERENCES

Abran, A., Khelifi, A., Suryn, W., & Seffah, A. (2003). Usability Meanings and Interpretations in ISO Standards. *Software Quality Journal, 11*(4), 325–338. doi:10.1023/A:1025869312943

Adipat, B., & Zhang, D. S. (2005). Challenges, methodologies, and issues in the usability testing of mobile applications. *International Journal of Human-Computer Interaction, 18*(3), 293–308. doi:10.1207/s15327590ijhc1803_3

Cockburn, A., & Gutwin, C. (2010). A model of novice and expert navigation performance in con-strained-input interfaces. *ACM Transactions on Computer-Human Interaction, 17*(3), 13. doi:10.1145/1806923.1806927

Damodaran, L. (1996). User involvement in the systems design process—A practical guide for users. *Behaviour & Information Technology, 15*(6), 363–377. doi:10.1080/014492996120049

Goldman, R., Price, K. J., Jacko, J. A., Lin, M., Feng, J., & Sears, A. (2006). Motion does matter: An examination of speech-based text entry on the move. *Universal Access in the Information Society, 4*(3), 246–257. doi:10.1007/s10209-005-0006-8

Gonsalves, A., & Ryan, C. (2005, January). *The effect of context and application type on mobile usability: An empircal study.* Paper presented at the 28th Australasian conference on Computer Science - Volume 38, Newcastle, New South Wales.

Gulliksen, J., Göransson, B., Boivie, I., Blomkvist, S., Persson, J., & Cajander, Å. (2003). Key principles for user-centred systems design. *Behaviour & Information Technology, 22*(6), 397–409. doi:10.1080/01449290310001624329

Heo, J., Ham, D.-H., Park, S., Song, C., & Yoon, W. C. (2009). A framework for evaluating the usability of mobile phones based on multi-level, hierarchical model of usability factors. *Interacting with Computers, 21*(4), 263–275. doi:10.1016/j.intcom.2009.05.006

Janlert, L.-E., & Stolterman, E. (2010). Complex interaction. *ACM Transactions on Computer-Human Interaction, 17*(2), 1–32. doi:10.1145/1746259.1746262 PMID:20627676

Lazar, J., Hochheiser, H., & Feng, J. H. (2010). *Research Methdos in Human-computer interaction.* Chichester, UK: Wiley.

Parhi, P., Karlson, A. K., & Bederson, B. B. (2006, September). *Target size study for one-handed thumb use on small touchscreen devices.* Paper presented at the 8th conference on Human-computer interaction with mobile devices and services, Finland. doi:10.1145/1152215.1152260

Park, Y. S., & Han, S. H. (2010). One-handed thumb interaction of mobile devices from the input accuracy perspective. *International Journal of Industrial Ergonomics*, *40*(6), 746–756. doi:10.1016/j.ergon.2010.08.001

Perry, K. B., & Hourcade, J. P. (2008, May). *Evaluating one handed thumb tapping on mobile touchscreen devices*. Paper presented at the Proceeding of graphics interface 2008, Windsor, Ontario.

Polkosky, M. (2010). To Usability Test...Or Not? *Speech Technology*, *15*(5), 7.

Preece, J. (1994). *Human-computer Interaction*. Reading, MA: Addison-Wesley Pub.Co.

Robbins, B. (2010, January). Rethinking User Interface. *Game Developer*, *17*, 15.

Rosson, M. B., & Carroll, J. M. (2002). *Usability Engineering: Scenario-Based Development of Human-Computer Interaction*. Sydney, New South Wales: Morgan Kaufmann.

Schmidt, U. (2008). (2008) Wiimote whiteboard. Retrieved 17 September, 2013, from http://www.uwe-schmidt.org/wiimote-whiteboard

Sharp, H., Rogers, Y., & Preece, J. (2006). *Interaction Design: beyond human-computer interaction*. Hoboken, NJ: Wiley.

TechSmith Corporation. (2009a). Hardware Usability Testing with Morae, Retrieved 20 January, 2014, from http://www.techsmith.com/learn/morae/3/customer-research/Hardware-or-Mobile-Device-Usability-Testing.pdf

TechSmith Corporation. (2009b). Software Usability Testing with Morae, Retrieved 20 January, 2014, from http://www.techsmith.com/learn/morae/3/customer-research/Software-or-Web-Site-Usability-Testing.pdf

Tilvawala, K., Myers, M., & Sundaram, D. (2011). *Design of ubiquitous information systems for digital natives*. Paper presented at the Pacific Asia Conference on Information Systems, Brisbane. Retrieved from http://aisel.aisnet.org/pacis2011/192

Uther, M. (2002, August). *Mobile Internet usability: What can 'Mobile Learning' learn from the past?* Paper presented at the IEEE International Workshop on Wireless and Mobile Technologies in Education, Växjö University, Växjö.