Cross Platform Usability: Evaluating Computing Tasks Performed on Multiple Platforms

Mr. Brian Patrick, Brigham Young University

Brian Patrick is an upperclassman attending Brigham Young University in pursuit of a BS in Information Technology. Primary interested in studying cyber-physical systems and related applications. Hobbies include participating in the maker community and building enthusiast PCs.

Dr. Richard G. Helps, Brigham Young University

Richard Helps has degrees in EE and a PhD in Instructional Technology. He is primarily focused on Cyber-Physical systems in IT with related interests in HCI. He is also interested in instructional design. He is a member of ACM-SIGITE, IEEE-CS and ASEE.

A Proposed Methodology for Comparative Usability Analysis of Cross-platform Computing

Abstract

Evaluating the usability of software across various computing hardware, including desktop computers, laptops, tablets, and smartphones, is becoming increasingly important as our need to complete computing tasks on various computing platforms becomes more commonplace in our daily lives. This shift in computing expectations focuses on the ability to compute horizontally, performing the same tasks within different computing environments, as opposed to vertically, performing different tasks in the same computing environment. In this paper, we propose a method by which the usability of a given piece of software can be comparatively analyzed across all hardware platforms on which it is available. This methodology is based on the Majrashi and Hamilton’s twelve factors of usability and has been extended to include specific applications and metrics for the following computing platforms: desktop computers, laptops, tablets, smartphones.

These metrics focus on collecting quantitative data about the usability of each platform. This quantitative data contrasts with recent studies which focus extensively on qualitative data regarding the user experience. Combining these two types of studies will allow researchers to capture a more comprehensive picture of software usability across computing platforms.

The methodology has been validated by performing an initial study using the tasks and performance metrics across each of the defined computing platforms. The results from these initial tests have allowed us to make improvements to the testing methodology as well as some hypotheses for further testing. This proposed methodology, when used in conjunction with qualitative research, can provide a reliable approach for cross-platform usability analysis.

Some considerations for educational design of cross platform methodology are discussed.

1. Introduction

Advancement in technology has presented new opportunities for users to perform computing tasks on many new platforms and in many new environments. These new opportunities introduce new challenges for software developers as they seek to design a cohesive usability experience across all computing platforms [1,2,5,7]. With multiple computing platforms available, such as desktops, laptops, tablets, and smartphones, users can accomplish the same computing task in many different environments. As the variety of platforms made available to us continues to grow it will become increasingly important that we are able to evaluate the usability of these platforms and validate their ability to help us effectively complete our computing tasks.

It has become natural for people to switch between computing environments, from desktop computer to smartphone, as they themselves move between functioning areas, from home to commuter train. For people to continue their work and/or play uninterrupted, it is important that these users can seamlessly transition from one platform to the next. Additionally, it is important to note the current trend in computing is toward mobile devices and away from desktop computers. This trend is evidenced by the widely reported slowing of desktop computer sales throughout the world. Desktop computing remains important, but many users of desktop computing environments also use mobile platforms to supplement their work and help them accomplish their computing tasks [9]. This makes it important for designers to understand the differences in usability between computing platforms so they may best adapt to the computing expectations of their customers.

This modern shift in computing expectations focuses on the ability to compute ‘horizontally’. As defined by Majrashi and Hamilton, ‘horizontal computing’ is the ability to perform the same tasks within different computing environments, as opposed to vertical computing, performing different tasks within the same computing environment [3,10]. We propose a method by which the usability of cross-platform software can be comparatively analyzed horizontally between hardware platforms on which a software application is available.

The comparative analysis methodology outlined in this paper goes further than previous cross-platform analyses as it takes into consideration all the major computing platforms that are commonly used by both professionals and casual users. These platforms include desktop computers, laptop computers, tablet computers, and smartphones. To accurately grasp the uniqueness of each platform, the testing methodology outlined takes into consideration the hardware capabilities, limitations, and adaptations of each platforms. Additionally, we are considering the environment in which each of these platforms are commonly used as this may have significant impact on a user’s ability to complete a task on any given platform. By focusing on quantitative metrics for each computing platform we can more granularly analyze the performance differences between each.

Greater understanding into how tasks are performed effectively across different computing platforms can provide insight into the design methodology for future software development. There are many articles on cross-platform development which focus on rapidly developing the same application for multiple platforms [2,5,7]. However, this approach can result in a lackluster user experience for some computing environments for which the application was not the primary target. Understanding the strengths and weaknesses of the user experience for each platform allows developers to design their applications that maximize the strengths and minimize the weaknesses for each platform, resulting in an optimal user experience across every available device.

2. Usability Measurement

It is not uncommon for a user to begin writing an email on a desktop computer and complete the same email on his/her mobile phone later. Each platform has unique attributes which can add or detract from the user experience of performing a given task; for example, a desktop computer benefits from a full-size keyboard and large screen, but is restricted in mobility by its size and power requirements. In this study, we are focusing on how well a user can perform the same task on each of four defined platforms. We will use the following common definition of usability, “Usability is the extent to which a product can be used by specific users to achieve specified goals with effectiveness, efficiency, and satisfaction in a specified context of use.” [3] and develop usability tests to compare usability across platforms.

This study focuses on analyzing and comparing the usability of each platform, it does not evaluate the process of transition between platforms. We note that there are many services, such as Apple's "Handoff" or Android's "Pushbullet" application, which acknowledge the importance of horizontal computing by providing software which allows a user's task to be synchronized across multiple platforms.

Two common measurements of user experience are usability and user satisfaction. Usability is measurable by defined metrics such as time taken and number of errors while the computing task is performed. User satisfaction is measured by observation, surveys, diaries, interviews and similar instruments [4,10]. Waljas et al [10], Sahar et al [8] and others, in their studies of cross platform user experience have focused on the satisfaction of user experience by documenting diaries and interviews with participants. However, these research studies focus on user satisfaction and they neglect other aspects of usability, putting an emphasis on qualitative over quantitative data. Understanding both qualitative and quantitative data can significantly enhance understanding of the complete user experience [4].

In normal use, the operating environment is significantly different for each platform. We defined a set of operating conditions that represent typical usage for each platform. For example, a desktop is used at a desk sitting down, whereas a smartphone is used in public environments, standing or walking. These are described in Appendix C.

3. Methodology

Majrashi and Hamilton outline twelve factors, that can be used to accurately identify the usability of a computing platform [3]. Their factors are: efficiency, effectiveness, learnability, memorability, business, accessibility, understandability, satisfaction, universality, helpfulness, safety, and visibility. As previously mentioned, recently there have been many articles which focus directly on the user experience which identify factors such as accessibility, understandability, satisfaction, helpfulness, safety, and visibility. However, our aim is to accurately represent quantifiable usability. Therefore, we will emphasize the factors of efficiency and, to some extent, effectiveness in our methodology.

We initially created nine tasks from three application categories to span the range of user experience. Each task was tested on the four computing platforms defined. Later in the development of the methodology, we significantly simplified the testing process down to three tasks, one from each application category. This was done for practical testing reasons and to eliminate data redundancy. The application categories include business, entertainment, and games. The categories were selected after reviewing lists of most popular apps from Apple's App Store, Google's Play Store, and Microsoft's Windows Store. The most popular apps are dependent on application domains and vary by platform, and there is no common classification scheme, but the three selected categories appear as a major category in all listings. These represent a broad cross section of activities performed on all platforms.
In a separate study Majrashi, Hamilton and Uitdenbogred indicate that one of the main issues to be addressed before defining cross-platform usability is defining usability metrics [4]. Following this model, we have defined metrics of performance which can be used to measure user

performance for each task. Defined performance metrics include: time, number of clicks/taps, and number of errors; each of which are dependent on the nature of the task. For example, email tasks are considered successful when a user can accurately communicate a desired message efficiently and without error. Therefore, measuring this task on time to execute and number of errors provides appropriate analytical data. This data is then used to provide insight into the differences between the various platforms.

It is important to note that some of the software applications have been adapted to a specific platform. These adaptations may include a modified user interface or workflow for some, if not all, of the tested platforms. To provide consistency between all platforms, software applications were selected that have been natively developed, where available, for each platform. For the initial study application consistency was most effectively obtained within the Apple ecosystem as developmental guidelines for software on Apple computers and mobile devices encourage developmental consistency. Details regarding how the selected applications were adapted to each of the platforms is detailed in Appendix B.

The details of the tasks performed can be found in Appendix A at the end of the paper. These details include the computing category, task, and application; as well as an explanation of each task, the specific metrics of performance measured, and task outlines and specifications.

3.1 Randomization and Learning Bias

Originally we proposed that all users should perform all tasks on all platforms. Our initial testing showed that each user would be testing for extended periods of time—several hours of test time, which we determined to be unreasonable. Additionally, there is a learning bias as each task is repeated on a subsequent platform. We propose to control for this bias and also shorten the tests by allowing each user performs the tasks on only two platforms. Platform order will be assigned to users randomly. By reducing the platforms to two, the time per user will be greatly reduced, the learning effect will only be from one platform to a second platform, rather than across four platforms. By randomizing the platform selections, we will have as many subjects learning from platform A to B as we do from B to A. Statistically the learning effect will be removed. This approach requires significantly more test subjects but will require a much shorter test time for each.

3.2 Metrics of Performance

Quantifiable metrics permit quantitative analysis, as has been discussed. The metrics selected for the tasks as discussed above are time, errors and clicks/taps. In each case, lower numbers indicate better performance. Details of the metrics for each task are in Appendix A.

3.3 Development of Task Design

We initially defined nine tasks across four platforms (up to 36 tests/user). Upon realizing the time required for a user to complete all tasks was unreasonable, we reduced this to two platforms per user and three tasks per platform. These changes were made as we realized the types of user interactions and metrics of performance we wished to measure were redundantly tested between several of the tasks. The remaining tests consist of each test subject performing three tasks twice (two platforms) for a total of six tasks. This reduction helped to greatly reduce the amount of time each user spent testing.

3.4 Autocorrect

All four defined platforms have a form of autocorrect to help the user avoid common spelling and grammar errors. These autocorrect features correct user's mistakes but may occasionally introduce unintended mistakes when the autocorrect doesn't match the user's intended input. Since these software optimizations have been implemented natively as part of each platform we have chosen to accept these autocorrect features as characteristics of the platform. The 'text entry' test did not include autocorrect.

4. Validating the Methodology

We ran initial tests with very few users using eight of the original tasks. This gave us the data in Table 1. These tests were useful in analyzing the testing procedures and developing analytical approaches. Most of the data collected must be considered formative as it mostly resulted in modifications to the tests and procedures. Subsequently we ran tests with three tasks and two platforms. Some data was collected using the final form of the test procedure from a small group of eight students, acting firstly as test subjects then subsequently as testers. The data for the two-platform tests was summarized and averaged across the test subjects as shown in Table 2. This data shows testing in two sequences; from a PC (laptop/desktop) to a Mobile (phone/tablet) and then from a Mobile to a PC. This was done so that learning biases can be identified. Some details are omitted (clicks, errors) and some condensed (combining platforms) to simplify the presentation of this data.

5. Analysis

Testing was done in three phases. The first phase was done with the methodology developers and resulted in mostly changes to the measurement procedure and some understanding of the required analysis techniques. The second phase was with a small group of test subjects (students in a related class) and resulted in much greater simplification of the procedure. (see table 1) The third phase was with a larger group of users, and improved understanding of the testing and analysis procedures (see Table 2). All phases used small groups and so the data is not statistically significant, but provides indicators for future research.

The measurements that we are interested in are the differences between two computing platforms. The validation tests identify areas to be watched carefully as testing is extended to more users, as well as suggest the types of analysis that are relevant.

Table 1. Summary of the results for eight tasks (phase one) and some relationships. The ratios shown are relative to the desktop values. Three repetitions of each test.

Task	Type	unit	Desktop	Tablet	Ratio	Smartphone	Ratio
Text entry. (60 sec test)	business	wpm	97	44	*0.45*	41	*0.42*
		words	96	61	*0.64*	44	*0.46*
		err	2	19	*9.50*	4	*2.00*
	correct word %		98%	69%		91%	
Email	business	secs	188	309	*1.64*	388	*2.06*
		errs	0	1		1	
Browse	entertainment	secs	33	94	*2.85*	72	*2.18*
		clicks	5	19		24	
Music	entertainment	secs	88	56	*0.64*	99	*1.13*
Video	entertainment	secs	22	18	*0.82*	22	*1.00*
Cards	game	secs	320	322	*1.01*	382	*1.19*
First person	game	secs	318	230	*0.72*	185	*0.58*
Top-down	game	secs	367	413	*1.13*	356	*0.97*
		points	160	134	*0.84*	125	*0.78*
Time totals		secs	1336	1442	*1.08*	1504	*1.13*

Table 2. Summary of data for three tasks (phase two). The ratios shown are relative to the PC values. Only the time data (seconds) is shown in this table. n=8

Platform	PC to Mobile			Mobile to PC	
Task	Type	PC	Mobile	Mobile	PC
Email	Bus.	66.5	110.5	134.5	100.5
Music	Ent.	71.5	83.5	134.5	92.5
Cards	Game	305	307.5	399.5	362.5
Totals		443	501.5	668.5	555.5

The major use of these results so far has been in adapting the test procedures. For example, the initial tests required the user to watch a video or listening to music for several minutes after locating the video clip or song. This audience time only masked the significance of the usability data with an activity that did not reflect usability, but merely represented passive reception. All tasks were re-analyzed and certain procedures were modified to focus on activities that indicated differences in usability. The testing time overall has been reduced from the initial design of several hours per user (four platforms, nine tests) to about 20 minutes per platform or about 45 minutes for a complete single user experience (two platforms, three tests). It would have been difficult and expensive to recruit a statistically significant number of users to complete a set of tasks lasting several hours per user.

One of the difficulties of measuring cross-platform occurs when the app developers have optimized the tasks in different ways for each platform. This difference is part of what we are interested in but creates problems when quantifying user experience across multiple platforms. For example, the UI for Spotify, a music steaming application which is used during the music task, is very different, and more efficient, when used on the tablet than when it is used on the desktop. The Spotify UI on the smartphone application was not platform optimized to the same degree as the tablet application. This lack of consistency in optimization, along with environmental factors, resulted in decreased performance on the smartphone when compared to either the desktop or tablet platforms.

Responsive design and platform-specific designs are more efficient than standardized design, however they can be more expensive to develop. On the other hand, standardized interfaces across platforms are easier for users to move between platforms. Thus, the paradoxical usability goal for developers is to make the interface have a common look and feel across all platforms, while adapting it for each platform.

Even in this preliminary study some expected, and unexpected, results emerged. For example, referring to Table 1, the increased error rate of the text entry test on mobile platforms without autocorrect (two to ten times as many errors)—suggest that the 'autocorrect' features of mobile platforms are probably necessary. Other potentially interesting results emerge from the gaming category, where it appears that platform optimizations make mobile platforms more effective than desktop ones. This has been noted in earlier work (not published) where students modified a desktop version of the "Plants vs. Zombies" game to play on a large touch screen monitor. Scores and satisfaction improved radically as the touch interface allowed two-handed play on different parts of the screen almost simultaneously rather than using a mouse to move a single cursor around the screen. These examples suggest that platform change efficiencies are not all one-sided. These trends will be monitored with interest as more data is collected and analyzed.

Although our test samples are still too small for statistical reliability, the results are interesting enough to propose hypotheses for further research. Our modified procedure includes analyzing the order in which participants repeat tests on different platforms to counteract learning effects. The data in Table 2 showed clearer between-platform results. Users slowed down when going from PC to Mobile platforms, and sped up when going from Mobile to PC. The clear and unsurprising conclusion is that people work faster with a full screen and keyboard. An expected result is that the availability of a full keyboard and screen for initial testing (first platform) resulted in 30 to 100% faster performance on the subsequent mobile device (second platform). A less expected result is that, when tested as a second platform the mobile device is slightly faster than a pc as a second platform by about 10%. This appears to be evidence of a learning effect but it is not always clear in the results; more data is required.

One of the other concerns of this study is the wide range of available applications which could be used for testing as well as the rapid evolution of applications and platforms. However, such evolution is an insufficient justification for not testing. By designing the tests as a series of modules in different application genres it is much easier to execute a test on a set of platforms and repeat the testing procedure for new circumstances as they arise—and still compare the results across platforms. Furthermore, if a developer were only interested in one genre or application, then a targeted series of tests could provide useful quantitative information with a significantly reduced test procedure.

6. Implications for Computing Education

Computing educators, particularly those in the fields of human-computer interaction, website design, and mobile application design, need to be aware of the growing trend towards cross-platform mobility. Designers working on specific platforms are typically well-educated in designing to meet the needs of specific platforms. However, the issue of cross-platform usability is seldom directly addressed. “Responsive” websites have become popular and heavily discussed in recent years [11]. These and other multi-platform methods address the need to design across platforms, but do not inherently address the need to test usability across platforms. The methodologies described here provide a framework to address that need. This study did not specifically incorporate curriculum design for cross-platform usability but the test methods could be incorporated in human-computer interaction courses or any course that deals with multiple platforms.

7. Conclusions

Applying the principles of Majrashi and Hamilton [3] has allowed us to create a model by which future researchers can measure and compare the usability of cross-platform computing. The defined metrics of performance, proposed simulated environments, and task outlines provide a framework for evaluating quantifiable usability for current and future cross-platform software applications. The model includes adaptations for different computing platforms in typical working environments. It also includes applications that are optimized for use on specific platforms. When used in conjunction with other methods of analyzing user experience, such as those used by Wäljas et al. [10] and Sahar et al. [8] we may obtain a more complete picture of the overall usability of cross-platform computing applications.
Analysis of the tasks and creation of appropriate test environments provides a significant method to compare performance relevant to each platform.

One possible future direction for this research is to develop custom applications for each platform that would enable testing specific usability functions. This would provide good isolation of usability effects but it sacrifices authenticity. It also loses the good effects of professional app developers optimizing their apps across platforms.

Future research in automating the process is also possible. Having simplified and streamlined the testing procedure it is very possible that the process could be scripted and delivered as an on-line testing packing. The system could measure time to completion and record responses for later manual or automated analysis. This would greater simplify the problem of recruiting enough volunteers.
Future research in this area will include more complete testing with multiple test subjects. Extended research in this area will also include specialized measurement software for task optimization by platform and measuring the handoff experience as users transition from one computing platform to another.

8. References

[1] Arhippainen, Leena. 2013. “A Tutorial of Ten User Experience Heuristics.” *Proceedings of International Conference on Making Sense of Converging Media*, ACM, 2013, 336–37. doi:10.1145/2523429.2523491.

[2] Macik, Miroslav, Tomas Cerny, and Pavel Slavik. 2014. "Context-Sensitive, Cross-Platform User Interface Generation." *Journal on Multimodal User Interfaces, Springer 2014,* 8 (2): 217–29. doi:10.1007/s12193-013-0141-0.

[3] Majrashi, Khalid, and Margaret Hamilton. 2015. "A Cross-Platform Usability Measurement Model." *Lecture Notes on Software Engineering* 3 (2): 132–44. doi:10.7763/LNSE.2015.V3.179.

[4] Majrashi, Khalid, Margaret Hamilton, and Alexandra L. Uitdenbogerd. 2015. "Multiple User Interfaces and Crossplatform User Experience : Theoretical Foundations." *Computer Science & Information Technology (CS & IT),* no. JANUARY: 43–57. doi:10.5121/csit.2015.50205.

[5] Mesfin, Gebremariam, Gheorghita Ghinea, Dida Midekso, and Tor Morten Grønli. 2014. "Evaluating Usability of Cross-Platform Smartphone Applications." *Lecture Notes in Computer Science (including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)* 8640 LNCS: 248–60. doi:10.1007/978-3-319-10359-4_20.

[6] Molish, R., & Nielsen, J. 1990. "Heuristic Evaluation of User Experience - Case Nokia." *Proceedings of ACM CHI,* 249–56.

[7] Mullins, Cheri. 2015. "Responsive, Mobile App, Mobile First." *Proceedings of the 33rd Annual International Conference on the Design of Communication - SIGDOC '15*, 1–6. doi:10.1145/2775441.2775478.

[8] Sahar, Farrukh, Jari Varsaluoma, Sari Kujala, and Kaisa Väänänen-Vainio-Mattila. 2014. "Identifying the User Experience Factors of a Multi-Component Sports Product." *Proceedings of the 18th International Academic MindTrek Conference on Media Business, Management, Content & Services - AcademicMindTrek '14*, 85–92. doi:10.1145/2676467.2676499.

[9] US Census Bureau, "Computers and Internet Use: 2013" American Community Service Reports, Issued Nov 2104 accessed May 2016, https://www.census.gov/history/pdf/2013computeruse.pdf

[10] Wäljas, Minna, Katarina Segerståhl, Kaisa Väänänen-Vainio-Mattila, and Harri Oinas-Kukkonen. 2010. "Cross-Platform Service User Experience." *Proceedings of the 12th International Conference on Human Computer Interaction with Mobile Devices and Services - MobileHCI '10*, 219. doi:10.1145/1851600.1851637.

[11] Podjarny, Guy "RWD Ratio in Top 100,000 websites - refined", Blogpost, Jan 2014, accessed March 2017, http://www.guypo.com/rwd-ratio-in-top-100000-websites-refined/

9. Appendix A

9.1 Computing Platform Comparison

9.1.1 Desktop
Platform: Apple iMac
Operating System: OSX
Key Characteristics: Large screen (non-touch), full-sized keyboard, traditional mouse
Usage Factors: Desktop computers are most often found in a home or office setting. Spatial constraints are an issue as desktop computers traditionally require a desk large enough to manage the monitor, full-sized peripherals, and the computing tower.

9.1.2 Laptop
Platform: Apple MacBook
Operating System: OSX
Key Characteristics: Medium screen (non-touch), mid-sized keyboard, trackpad mouse
Usage Factors: Laptop computers are most often found in a home, office, or classroom setting. Laptops are the most versatile platform as they provide the flexibility of a full operating system in a mobile form-factor. While laptops do not have the same spatial constraints as a desktop they do require the user to be seated.

9.1.3 Tablet
Platform: Apple iPad
Operating System: iOS
Key Characteristics: Medium screen (touch), Mid-sized virtual keyboard
Usage Factors: Tablet computers are used in a home, office, classroom, or any public setting. They are traditionally very compact and lightweight allowing users to operate the device with both hands while standing up, sitting down, or lying down. The tablet has a different interface, using on-screen touch and data entry instead of a keyboard mouse.

9.1.4 Smartphone
Platform: Apple iPhone
Operating System: iOS
Key Characteristics: Small virtual keyboard, small screen (touch)
Usage Factors: Smartphones are found in almost any environment as they are traditionally with the user whether they are actively being used or not. The extremely small form-factor allows users to operate the device with either one or both hands while walking, standing, sitting, or lying down.

10. Appendix B

10.1 Categories, Applications, Activities, and Metrics of Performance

10.1.1 Email Task
Category: Business
Task: Email
Application: Gmail
Metrics of Performance: Time and Number of Errors

Explanation: The user will receive an email with three easy to answer questions about themselves. The reason they are answering questions about themselves is so they won't have to think too long and hard about the question before arriving at an answer. The user will be asked to respond by retyping the complete question and then answering each question in complete sentences. Users will be measured on the time it takes them to send a response to the email and the number of typographical errors in their response.
Cross-platform adaptation: Gmail was designed to be responsive to the size of the screen for every device. The larger screen size devices (desktop and laptop) allowed for the entire email as well as the typing area to be seen simultaneously. However, the smaller screen size devices (tablet and smartphone) meant that the questions were not visible while typing due to the lack of screen real-estate.

10.1.2 Music Task

Category: Entertainment
Task: Streaming Music
Application: Spotify
Metrics of Performance: Time
Explanation: The user will find three playlists in a specified genre and number of followers (the playlist should be different for each platform). They will be measured on how long it takes them to identify the three playlists and select one song from each playlist. The Spotify application allows users to continue searching while listening to a song in the background.
Cross-platform adaptation: The Spotify music streaming application on the tablet platform uses cascading menus/tables for the user interface. This means that when a musical selection (genre, artists or song) is chosen a new panel is added to the screen and all previous panels are still available as tabs on the side of the screen. This allows the user to jump back any number of steps with a single touch. This differs from the desktop, laptop, and smartphone applications as they use traditional forward and backward buttons to allow users to navigate between previous windows, which requires additional input from the user.

10.1.3 Game Task

Category: Games
Task: Card Game
Application: Hearthstone
Metrics of Performance: Time
Explanation: Hearthstone has a six-mission tutorial which players must play through before they can play the actual game. The user will be measured on how long it takes them to complete the first two missions of the tutorial on each platform.
Cross-platform adaptation: The card game Hearthstone is adapted on the smartphone platform to accommodate the lack of screen real-estate. All other platforms have the user's hand of cards at the bottom of the screen which can be directly selected, dragged, and dropped to perform in game actions. The smartphone platform places an icon off to the right side of the screen which, when selected, becomes larger and allows users to perform actions on the cards.

11. Appendix C

11.1 Computing Platform Test Environments

11.1.1 Desktop

Platform: Apple iMac

Operating System: OSX
Desktop Environment: Home Desk
Environment Simulation: Desktop environment tasks will be performed at a desk. Here the user will have access to a full-sized keyboard, mouse, and monitor. The user will also have ample space for sitting comfortably in an office chair.

11.1.2 Laptop

Platform: Apple MacBook
Operating System: OSX
Laptop Environment: Bench in public office-style environment
Environment Simulation: Laptop environment tasks will be performed on a bench in a relatively quiet but public environment. The user will only have access to the keyboard, trackpad, and 13-inch screen that are built into the laptop computer. The user will have adequate elbow room but will not have a desk on which to place the laptop.

11.1.3 Tablet

Platform: Apple iPad
Operating System: iOS
Tablet Environment: Airplane Seat
Environment Simulation: Tablet environment tasks will be performed in a simulated airplane seat. The simulation environment will take place in row of seats in a classroom. The user will sit in the small seat with restricted leg room while two other volunteers sit on either side of the user.

11.1.4 Smartphone

Platform: Apple iPhone
Operating System: iOS
Phone Environment: Standing/Walking
Environment Simulation: The phone environment should simulate an urban environment. The user will either walk or stand to simulate the times when the mobility of smartphones is of the most importance.