GESTURE BASED SOCIAL MEDIA AND NEWS INTERFACE

By

Steven T. King Jr.

A DISSERTATION

Submitted to

The University of Liverpool

in partial fulfillment of the requirements

for the degree of

MASTER OF SCIENCE

dd/mm/201x

{**Edited 2011-January, 9 --- Version 12.2:** Dissertation Template (nested numbering version) and Author Notes Look like this. This is a special style used to annotate the boilerplate and also used to leave instructions and notes. This style stands out as a reminder that comments must be replaced or removed before a polished document is submitted. It is also a way to exchange notes with an advisor or review without impacting the document directly. It is also possible to use Microsoft Word to find all paragraphs in this AuthorNote style.

We thank Denis Hamilton who provided the template from which this one was adopted}

ABSTRACT

GESTURE BASED SOCIAL MEDIA AND NEWS INTERFACE

By

Steven T. King Jr.

This document is designed to use Body Text styles for all content. This is a typical body text paragraph.

The abstract should be around 350 words and should not exceed more than two pages.

DECLARATION

I hereby certify that this dissertation constitutes my own product, that where the language of others is set forth, quotation marks so indicate, and that appropriate credit is given where I have used the language, ideas, expressions, or writings of another.

I declare that the dissertation describes original work that has not previously been presented for the award of any other degree of any institution.

Signed,

Steven T. King Jr.

Student, Supervisors and Classes:

|  |  |
| --- | --- |
| Student name: | Steven T. King Jr. |
| Student ID number: | 15480884 |
| GDI name: | Taly Sharon |
| RMT (GDI) class ID: | ComputingReserachMethodsTraining.2012.21.01.906 |
| DA name: | Taly Sharon |
| DST (DA) class ID: | ComputingAdvisorClass. 2012.05.03.203 |

ACKNOWLEDGEMENTS

The University of North Carolina at Chapel Hill School of Journalism and Mass Communication and Dean Susan King provided equipment, testing space and support to complete this project.

I would like to acknowledge the Massachusetts Institute of Technology Media Lab and the DepthJS development team for their original development and accepting my contributions to the source code.

My advisor Taly Sharon of University of Liverpool was instrumental in my development and completion of the project. Also, professor Laura Ruel of the University of North Carolina at Chapel Hill for her advisement and suggestions regarding user testing and my wife Amy King for her support and copy edits.

{CAUTION: There is a section change at the end of this page. Do not do anything that deletes it. The section change is responsible for switching to the use of Roman-numeral page numbers.

Captions for figures are created by going to Insert | Reference | Caption… . The cross-references to tables, figures, and headings are by created using Insert | Reference | Cross-reference… . You can make more places to cross-reference by using Insert | Bookmark, as in the bibliography, and you can refer to those by Insert | Hyperlink… , as when making citations in the text. All of these cross-references will be implemented by unobtrusive hyperlinks in the resulting electronic document and the PDF.}

TABLE OF CONTENTS

Page

LIST OF TABLES vi

LIST OF FIGURES vii

Chapter 1. Introduction 1

1.1 Scope 1

1.2 Problem Statement 1

1.3 Approach 2

1.3.1 Research Methodology 2

1.3.2 Project Conduct 2

1.4 Outcome 3

1.5 Document Structure 3

Chapter 2. Background and review of literature 4

2.1 Related Work 4

Chapter 3. Analysis and Design 6

3.1 Architecture 6

3.2 Hardware Integration 6

3.3 Software Development 7

3.4 Research and Testing 12

3.5 A 13

3.6 B 13

3.7 C 13

Chapter 4. Results and Evaluation 14

4.1 A 14

4.2 B 14

4.3 C 14

Chapter 5. Conclusions 15

5.1 Lessons Learned 15

5.2 Future Activity 15

5.3 Prospects for Further Work 15

REFRENCES CITED 16

LIST OF TABLES

Page

[Table 1 Principle Topics in Pattlets 12](#_Toc120440496)

[Table 2 Full Pattern-Description Organization 13](#_Toc120440497)

{To update the LIST OF TABLES, right click on the entries, chose “Update Field”, choose “Update entire table” and click on OK}

LIST OF FIGURES

Page

[Figure 1. Navigational Data-Entity Hierarchy (typical) 14](#_Toc120440498)

[Figure 2. Converse Matching of Association Types in Data-Entity Relationships 15](#_Toc120440499)

{To update the LIST OF FIGURESS, right click on the entries, chose “Update Field”, choose “Update entire table” and click on OK

}

# Introduction

New technologies for interacting with computers are changing the way we consume social information and news media. While touch devices such as the iPad and Kindle have changed media habits and website interaction by using gesture controls and touch interfaces, gaming systems such as the Xbox 360 have integrated motion and depth detection.

At the same time, HDTV flat screens are becoming more prevalent throughout homes and business and better way to interact with those devices is needed. Motion and depth sensing technology is rapidly developing and with the mass release of the Xbox Kinect in MONTH of 2011 by Microsoft, this technology can be found in millions of homes and businesses.

While the hardware is developing and price falls there are efforts to bring this technology to the personal computer and not have it limited to tablets or priority gaming systems. Microsoft enabled the ability to develop motion-based interfaces by releasing the Kinect Application Programing Interface (Kinect API). Open source projects such as DepthJS worked to integrate the Kinect API and the universal web browsers but have seen little adoption.

All of these developing technologies, both hardware and software are now converging into gesture controlled, large screen experience. Users in public places no longer have to use a touch screen kiosk but can not begin interacting with an application simply by stepping forward or swiping a hand. In homes, users a no longer limited to a remote control or a keyboard and mouse but can control the cursor with the wave and push of their hand.

This project will integrate the most common hardware configuration to test some current news and social websites as well as develop a custom-designed interface for gesture control and interaction with news and information.

## Scope

This dissertation will explore the necessary design and user experience concerns publishers and developers should consider when developing for the emerging technology. It implements the most common hardware configuration and integrates gesture enabling software design for the browser. This project will provide qualitative analysis of existing desktop and mobile interfaces as well as custom designed experiences. It will also provide lists of design and development best practices for future development of gesture-based interfaces and applications.

The research and physical user testing is limited to the use of Xbox Kinect and DepthJS though other hardware and software solution are emerging in the market. The interfaces tested include a single website in the social (Facebook) and news (New York Times) category.

Usage of the word “interface” in this work refers to a browser-based user interface such as a webpage.

## Problem Statement

Gesture based, non-touch user interfaces are becoming more common in console gaming and other specialized industries such as health care but has yet to become a common and useful way to navigate the Internet, specifically news and social media sites.

This project will seek to answer the question of, “Can gesture-based interfaces using motion tracking sensor technology be used to consume news and engage with social and current events content?” In order to answer this question the project will design, develop and test a software interface and a hardware installation that enables users to interact with news and social information using gestures without requiring touch or extra peripherals in order to understand the common usage.

## Approach

### Research Methodology

To discover if gesture-based interfaces using motion tracking sensor technology can easily be used to consume news and engage with current events content through a custom-designed user interface, an analysis of existing technologies was preformed, contribution to an open-source project was developed, two custom-designed gesture interfaces we developed and existing websites and the new interfaces were tested.

Qualitative analysis research methodology of the interfaces as well as technical observations was adopted. This approach provided input from typical users as well as important technical analysis of the current implementations of the technology and the issues associated with current interfaces.

The main body of research consists of qualitative user tests of news, social and custom interfaces using gesture-based interactions.

### Project Conduct

This project required the implementation of consumer available hardware with the development of new software and user interface in order to compare the effectives and usefulness of this new technology with existing interfaces. The project took on the three major phases of exploration, development and testing.

Exploration

The first task was to determine the most useful and most common hardware configuration for using gesture-based non-touch interfaces in a home environment. Screens, computers and sensors were compared to determine the most likely and most useful consumer implementation of gesture technology. The ideal implementation was tested documented and acquired.

Once a hardware solution was determined then the software options were explored. Various consumer products and open-source applications were compared and tested through informal developer experimentation. It became apparent the non of the products or applications available at the time of development would be sufficient to provide the ability to not only test existing web interfaces but to be able to innovate a new user experience.

The most viable and contributive solution to the industry and the project was to partner with developers in the Massachusetts Institute of Technology Media Lab and contribute to the open-source project DepthJS.

Development

The development stage required two major coding endeavors. The first was contributions to make the hardware work through the DepthJS browser plugin work properly with Google Chrome, a multi-platform browser. Once complete, development of two web-based interfaces were created to demonstrate and test various commonly seen user interface elements.

Testing

Once the interfaces were developed a five-person user test was conducted. The

task-driven user evaluation required users to complete a series of navigational tasks on six different interfaces of social and news websites. Each test was monitored by the researcher and recorded with multiple cameras so the actions could be reviewed after the test.

Five users or evaluators were recruited through a university-wide email list to find users who were familiar with the Internet.

Prior to the evaluation, users were instructed on how to use the system and be allowed time to interact a website not being tested. This allowed the user to become familiar with the new experience of non-touch gesture control as well as gave time for the “wow factor” to wear off.

The user evaluation script, Appendix I, details the steps of the user evaluation. The researcher administered the user test each time directly following the script and all users were tested within 24 hours to ensure consistency in the test.

During the tests, the order of the tested interfaces was random and each task was timed.

After completing the user test, the evaluator was asked a series of questions regarding the three different experiences through a web-based survey. This survey found in Appendix II, in addition to observations and timed tasks provided the data necessary to draw conclusion as to the effectiveness of non-touch gesture-based interfaces.

## Outcome

This project contributes to the computer science industry as well as the journalism and social media industries by providing multiple usable, public and open-source non-touch gesture interfaces, which can be used as a starting point for further development. Code has been contributed and merged with the open-source DepthJS project. All code is available to public through Github.

In addition to the code base, a series of best-practices and a summary of insights and observations will help future development of gesture applications. As gesture sensor technology progresses, these insights should enable a more rapid release of new gesture interfaces by social and news publishers.

The application and findings has already been presented to researchers, journalists and developers at the Journalism Interactive Conference hosted by the University of Florida and to faculty researchers of the School of Journalism and Mass Communications and the School of Information Sciences at the University of North Carolina at Chapel Hill.

## Document Structure

This document provides detailed analysis of background research and current literature in Chapter 2 – Background and Review of Literature. Chapter 3 – Analysis and Design provides information regarding the development and testing of the application. The researching and development methods are detailed in Chapter 4 – Methods and Realization. Chapter 5 – Results and Evaluation provides explanation of the results from the analysis and user testing. The final chapter, Chapter 6 – Conclusion provides a the findings and details the achievements and objectives of the project.

[ADD Appendix information here]

# Background and review of literature

## Related Work

Non-touch gesture based user interfaces are a fairly recent development in the personal computing world. Must of the research available refers to touch-based gestures which is valuable and comparable in many situations. Those writings regarding non-touch gesture interfaces have been used in specific fields of study such as medical sciences and geography.

It was not until Microsoft released Kinect for Xbox360 that the sophisticated motion sensors reached many households. Kinect still holds the record for the fast selling electronic product ever. Microsoft took the technology farther by making providing an open API which has allowed developers to create new uses for the motion sensor technology and gesture based interfaces are being used for many unforeseen purposes such as security, health care and consumption of information (Schramm, 2010).

Periodical writings regarding non-touch gesture based interfaces provide details into the technical sensor, hardware, integration and usage of the technology.

Kam Lai, in his presentation to IEEE Southwest Symposium (2012) provide necessary scientific documentation that the Microsoft Kinect product was able to detect eight common hand gestures with 99% accuracy in a typical home environment. Lai’s research scientifically established the fact that the Kinect product is dependable allowing for further research on the next layer of integration, the application programing interface.

Mark Doulos (2011) wrote a journal article about how to use Kinect in GIS along with Google Earth. The piece explained how to integrate the sensor with the existing Google Earth Application but did not critically analyze the usage and user experience. The must useful part of Doulos’s article is the systematic explanation of integrating the hardware and the application.

The dissertation by Marcel Aarts (2012) not only demonstrates an application that integrates the motion sensor technology but also provides a detailed analysis of existing Kinect integration services and APIs. For example, Aarts provides valuable insights into the problems associated with the popular KinectJS which was technology considered for this project. Aarts’ work is focused on his specific application and the issues faced in development of his specific usage.

In Long’s Human Factors presentation on gesture interfaces in 1999 foretold many of the concerns and issues facing the industry today such as standards and inconsistencies but despite being dated offers insights into important human interaction interfaces.

Dan Saffer’s book, *Designing Gestural Interfaces* (2009) is the most comprehensive and applicable resource on the subject. This complete technical volume explains the broad implications of using gestural interfaces and the physical and psychological considerations. This book while providing a valuable introduction is limited in the technical detail regarding these interfaces.

Each of these peer-reviewed sources and published book provide some insights and establish a basis for further research but the most helpful insights come from periodicals with are making new and recent articles as the technology progresses. This is especially true concerning development and design of gesture-based interfaces where there no academic writing was found concerning the usability of gesture-based interfaces.

For example, Izkovitch (2012) wrote about the various considerations to take into account when designing for this technology for Mashable.com and App-side provided a list of top five considerations when designing gesture user experiences.

An explanation of the workings of the Microsoft Kinect sensor array can be found in the article by Mike Schramm (2012), which provides both an insightful view of PrimeSense, the company that designed and built the technology licensed by Microsoft, as well as a look at the technology inside the case. This article is one of few that profile PrimeSense as most focus on Microsoft.

In addition to providing design and technical insights the periodicals also provide a view of the future of non-touch computing. For example, in an article about the future of mobile phones for *The Atlantic,* A. Madrigal predicts the use of non-touch gesture interfaces for uses currently services by mobile phones. And an article on GTV shows a patent application Microsoft filed that has direct implication to a gesture home environment.

Advancing the current body of research must be built on previous established technical implementation while pushing the interface into new uses and designs that can be tested. This previous work both peer-reviewed and periodical is the basis of the new design that will further researched in this project.

# Analysis and Design

{Documentation of the analysis and design containing full details of the design. The design documentation should also be supplied (possibly as an appendix).}

## Architecture

The project encompass, integrating existing hardware and developed interactive software to create a user evaluation test as seen in Diagram X. The project examines existing, interfaces as well as custom user experiences and looks at the different technology, applications and testing in this area.

In order to effectively test gesture-based interfaces the hardware must be integrated into an environment that can be controlled and monitored. This requires the physical install of hardware and software configuration in the testing location as well as the development environment.

## Hardware Integration

The central device in gesture interfaces is the sensor technology. There are several gesture-enabled motion tracking technologies available on the consumer market but by far the most popular is the Microsoft Kinect, which set a record for the “fastest selling consumer device” with over 10 million sold. Typically, the Microsoft Kinect is used with the Xbox360 gaming console but Microsoft released a publically available API that enabled desktop applications to interface with the sensor.

In addition to a large market penetration, Microsoft also publically released the Application Programing Interface (API) that enabled developers to create custom applications that use the Kinect sensor. The Kinect API is vital to innovation in this area and the large adoption and uage of this API made it an appealing option for the sensor technology.

The Kinect device consists of an infrared emitter and depth sensor, a color VGA video sensor and a four-microphone array that are all used for motion and depth tracking. The camera with a resolution of 620x480 pixels captures images and depth at 30 frames per second and used in conjunction with the infrared grid that is invisibly broadcasted while in use. This data is processed and made available through the Kinect API.



Many of the available consumer sensor technologies are similar to the Microsfot Kinect is resolution and tracking methods but it should be noted that a new and entirely different method of motion tracking has recently been be release. Leap Motion provides a more precise tracking matrix in a smaller space enabling the use of smaller movements and hand gestures rather than full body gestures. This product had not been release at testing and development but could provide a new way of interacting with websites. This technology was not chosen for testing due to the lack of consumer availability.

Selection of the ideal screen size required less detailed analysis. A recent consumer study from Morgan Stanly reported that the majority of consumer though about 45 inches was the ideal screen size for a “smart tv.” Though this project does not use or require a Smart TV which is a television with an Internet connection and interactive applications, this became the basis for the size selection as a smart TV is similar in usage from a consumer’s perspective. Also in consideration is the distance from the screen when interacting with gestures. The Microsoft Kinect requires the user to be at least five feet away and may applications suggest 6 to 10 feet and screen that allowed users to be able to read standard web text on a 1920x1080 resolution display from about six feet away had to be chosen. After considering each of the factors, a 44 inch 1920x1080 LCD flat screen monitor was used for both development and user testing.

It was important that the software be computer platform agnostic and both Apple Mac computers as well as Windows based PCs be able to be used for gesture-based interaction. A MacBook Pro was used for the user test do to personal availability but the applications will work on either major platform.

For the purposes of later study the user test were recorded using two Cisco Flip Cams and this device was chosen due to available and ease of use but became problematic during testing as the cameras would at times turn off and stop recording with notification.

Hardware Implementation

* 1920x1080 HDTV
* Mac Computer
* Microsoft XBOX Kinect (Motion Tracking Sensor)
* Multi-camera video recording for monitoring

For the user testing, the screen was mounted with the top edge 6 feet high on a 8’x10’ wall. The Kinect sensor array was placed on a small bookshelf three feed off the floor directly centered under the screen. The computer was placed on the shelf below the Kinect. One camera was placed facing the user just above the sensor. The second camera was placed behind the user’s right shoulder so the user and the screen could be seen.

## Software Development

Various options for integrating web sites with the Kinect hardware were considered. The Kinect API provides a lot of functionality but an intermediary application or utility is required to allow web browser to interface with the Kinect sensor data.

The first consideration was a browser plugin ZigFu that has a JavaScript API. ZigFu is designed primarily for use with custom generated Flash or Unity 3D content and most examples are online games. The available API events were limited and did now have a way show a traditional cursor. This plugin would work for custom development but would not allow for the testing of existing websites like NYTimes.com or Facebook.

JavaScript library Kinesis.io was a consideration for development and there are several working examples of non-touch gesture-based interfaces developed for the Kinect but this library was not multi-platform enabled and only worked on Windows machines. This solution also has the drawback that it would not allow for testing of sites not specifically developed on this platform.

KinectJS is a popular JavaScript library that also provides access to the Kinect API. This library is designed and used primarily for gaming and not used for traditional browser navigation.

It became apparent that not only would new interfaces have to be built but in order to compare and test existing interfaces some middleware or utility application would be required to bridge the gap between the Kinect API and the browser.

Developers in the Media Lab of MIT created such a project in the form of a browser plugin. This plugin was built for Google Chrome, which provided the cross platform availability as well as the conceivability that this technology could soon be widely adopted.

DepthJS is both a browser plugin with a JavaScript API that allows developers to create custom events tied to specific gestures detected by the Kinect but it also provides default functionality for existing websites through the Chrome Browser API.

When the DepthJS plugin is enabled in Google Chrome, websites using HTTP protocol, not HTTPS, can be controlled through specific default gestures.

Waving:

Holding up one hand and pressing or pushing the hand forward enables a cursor. A blue circle appears in the center of the screen and the user can move the circle around the screen. To select a link, the user presses or pushes forward and any links that are within the radius of the cursor circle are presented in a simple menu and the user can select a specific link.

Scrolling:

Users can scroll up and down the page using an open hand and moving it quickly up or down pushing towards the top or bottom of the browser window.

Swiping:

When a user swipes from left to right or right to left the browser will emulate the pressing of the forward or back browser buttons, respectively, allowing users to navigate multiple webpages.

Use of these default gestures allowed for the testing of existing news and social sites. This functionality also makes it feasible for a large user adoption of the technology because more than just custom sites can be navigated with gestures.

In addition to the default gestures, DepthJS also provides and a JavaScript API that enables developers to create interfaces that make the best use of gestures and overrides the default gestures.

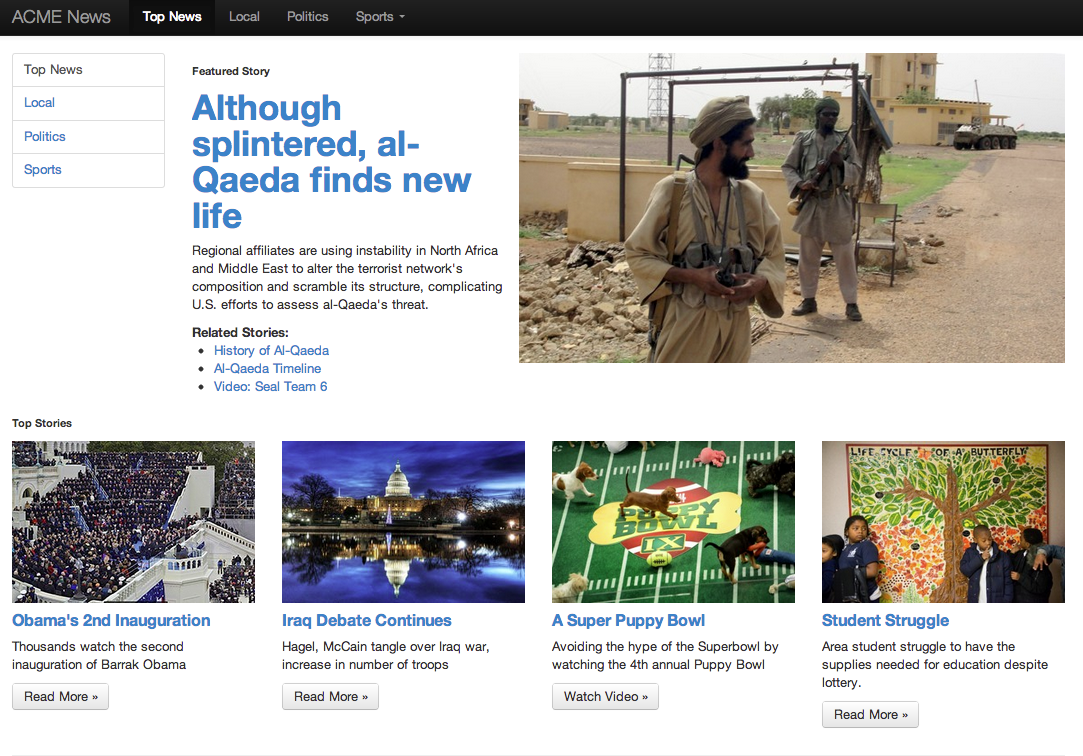
Though much development and effort was spent on DepthJS during a short development cycle, the codebase has not been updated since some major changes in Google Chrome that changed access to the browser API as well as how plugins interact with the browser. This became problematic and computers with the latest browser updates could not run the DepthJS plugin.

In order for the extension to work with the latest version of Chrome and be cross platform and enable testing of multiple sites several changes needed to be made to the existing code. The Manifest needed to be upgraded to the newest version and depreciated items removed as well as all inline links to scripts had to be replaced with absolute links to JavaScript files. The way that the extension or plugin sent and received messages from the browser API had to be changed to the new syntax. Also, new menu files had to be created. Everything was documented and the developers at MIT merged a pull request in October of 2012 and the plugin publically was distributed on Git Hub.

Once the browser extension was complete developer of the user interface began. Two interfaces were created that would allow for the testing of the more popular user interfaces elements.

A traditional but simplified news site was created using HTML5, CSS3, JavaScript, jQuery and the Twitter Bootstrap responsive framework. This industry-wide adopted framework includes easy implementation of user elements such as tabs, vertical menus, dropdown menus and buttons. This interface presented a single dominant element or main story and four secondary stories along with navigation tools at both the top and the left side to access other sections.

Diagram XX implements basic and traditional navigational items detailed above such as standard links and vertical tabs. Each navigational item can be accessed through the default gesture events

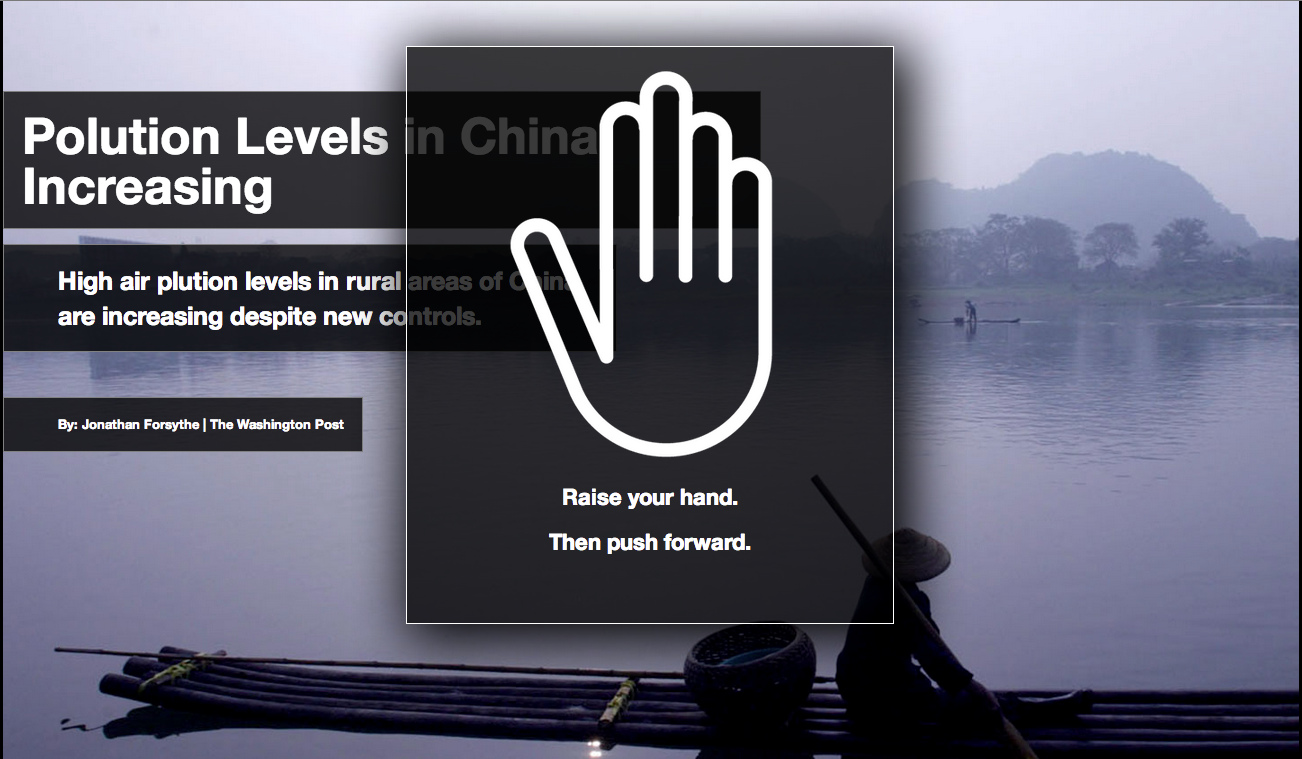


Users have a blue circle, (not shown) that represents a cursor that is controlled through gestures. User can click on links, button, and menus and on the photos.

The second interface was modeled after tablet photo viewer experiences where a single image consumes most of the screen and additional images are available through swiping. The same numbers of stories were presented with the most important being first and all the other stories followed but were not visible except individually after a swipe. Each story was represented with an image and a headline and subhead test to promote the story and the user could use the push gesture to access the text of the story.

This interface was developed with HTML5, CSS3, JavaScript, jQuery and the Twitter Bootstrap responsive framework as well along with implementation of a jQuery carousel plugin.

Diagram XX displays the slider interface where new stories/photos are displayed with swipes and text is displayed with a push gesture.



The final technological hurtle was to find a way to emulate a mobile device through a desktop browser so that the Facebook and NYTimes mobile sites could be tested using non-touch gesture interfaces. This was accomplished through a Chrome plugin that changes the request headers to resemble a mobile device.

Figure XX displays the software design and the integration with the Kinect API. Starting from the top the Kinect API interfaces directly with the DepthJS browser extension in Google Chrome. The Chrome browser displays the HTML pages and the associated languages or CSS and JavaScript.

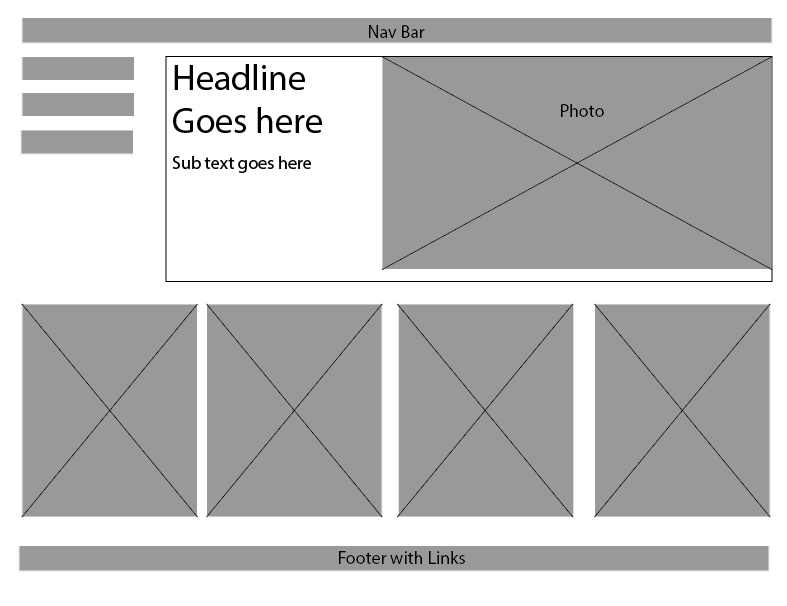
The overarching design methodology employed in development of the interfaces was an adaption of Website Design Method which is a user-centric method that has been employed widely in Kiosk development practices. Typically, WSDM analyzes the users and classifies users so that each group’s individual needs can be addressed. Then interface designs are made and the application architecture is designed. The design is then developed and implemented. Finally the different classified user groups test the product and changes are made as needed. It should be noted that WSDM was originally implemented and often used in oriented programing and though HTML and JavaScript is not object oriented but the methodology still works.

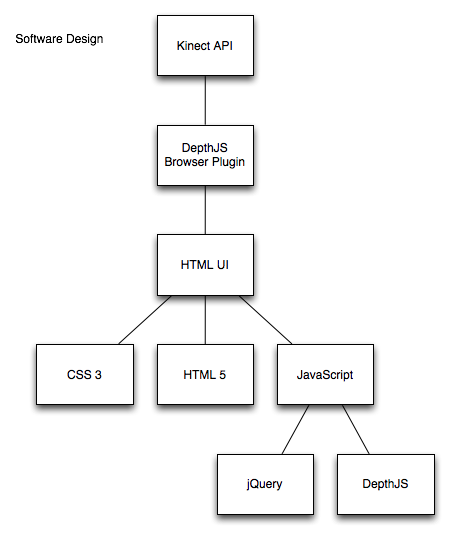
In the case of this project only one user group was classified in order to have an accurate qualitative user test.

The targeted group for both developed interfaces was 20 to 30 year olds who were Internet users and comfortable using technology but had never used non-touch gesture interfaces before. With this in mind it made it easier to develop an interface for this group because many of the user interface elements were modeled after sites this group would frequent.

Development of both interfaces also used an iterative design process where the designs were first wireframe. Then color comprehensive layouts were design and the code was then built based on the layout designs. These designs changed over time with several iterations based on developer testing experience and informal user testing. For example, the direction of large photo interface the direction of the sliding was changed based on ad-hoc feedback from individuals who were in the target user group that tested the interface during development and felt the concept was backwards because it was the opposite of the iPad. Another example includes the addition of the simple instructions screen based on feedback during development.

Diagram XX shows the original wireframe the where the basic design elements were determined. The wireframe and color comprehensive layouts were developed in Adobe Illustrator.





Once integration of the hardware, development and developer testing of the browser extension, two interfaces and mobile emulation was complete the user test was preformed.

# Research and Testing

Methods and Realization

{How the design was implemented? Changes made to the design in the course of the implementation. How was the data collected? How was the implementation tested? Typically code listings, screen shots and test runs will appear as appendices}

A task-driven user evaluation was designed to test and compare existing news and social media sites as well as the interfaced developed for this project.

The user testing approach was modeled after Jacob Nelson’s five user test. A group of five individuals were selected from respondents to email requests distributed on multiple email list services across the University of North Carolina. The makeup of the group was selected based on the WSDM user group categorization. The group consisted of two Men and three Women in their twenties or thirties who were comfortable with technology and used the Internet daily. All were right handed and ranged in height from 5’ 2” to 6’ 2”.

It is important to note that all users were right handed because though there should be no differences in the ways a left handed or right handed person interacts with the system, the sample size was not large enough to make comparisons on this factor so only right handed users were tested.

Prior to the evaluation, users were instructed on how to use the system and allowed time to interact with website that was not being test and was not a social or news website. This allowed the user to become familiar with the new experience as well as give time for the “wow factor” to wear off. Typically, the instruction time tool about one minute and the user took between one and two minutes before they alerted the researcher they were ready to continue.

The test required users to complete a series of navigational tasks on six different interfaces of social and news websites. Each test was monitored by the researcher and recorded with multiple cameras.

The user evaluation script, Appendix A, details the steps of the user evaluation. The researcher administered the user test each time directly following the script to ensure consistency in the test. User were asked to complete three different tasks for each website and each task was timed and recorded for comparison. If a task could not be completed after three minutes of trying the research asked the user if they would like to continue trying or move on to the next task. Appendix B provides the researcher’s worksheet used to record success, failure and time to complete each task.

The sites tested were, nytimes.com, mobile.nytimes.com, facebook.com, mobile.facebook.com, the gesture-enabled traditional news site and the large photo interface. The order of interfaces presented was random. The randomness of the order presented was important to eliminate the factor that users become more familiar with the gesture technology over time which could skew the results if the same interface was always tested first or last.

At the conclusion of all the tasks required, users were asked a series of questions regarding the six different experiences through a web-based survey. The survey provided the basis of the qualitative data to evaluate each interface. This survey can be found in Appendix C.

# Results and Evaluation

{Description of the results followed by their review. These may include, where appropriate, feedback from test groups, users and the project sponsor.}

## A

## B

## C

# Conclusions

{A summary of the project as a whole. Lesson learned. Possible applications and extensions of the work}

## Lessons Learned

## Future Activity

## Prospects for Further Work

REFRENCES CITED

{This section will demonstrate the Harvard-style bibliography for the entire dissertation. It must use the required format described <http://www.liv.ac.uk/library/ohecampus/ref.htm>. We have included some samples so that the use of cross-references from citations to the bibliography entries is illustrated.}

Aarts, M. (2012), *Using Kinect to interact with presentation software*. Uppsala: Malardalens hogskola. Available at:  <http://www.idt.mdh.se/utbildning/exjobb/files/TR1288.pdf>

[Accessed 17 September 2012].

App-side (2012), Top Five Things to take into account when considering the gesture controlled user experience. Available at: <http://www.app-side.com/2012/03/top-five-things-to-take-into-account-when-considering-the-gesture-controlled-user-experience/> [Accessed 17 September 2012].

Boulos, M., Blanchard, B. & Walker, C. (2011), Web GIS in practice X: a Microsoft Kinect natural user interface for Google Earth navigation. *International Journal of Health Geographics*, 10(45).

Gray, T. (2001), The challenges of multi-touch gesture interfaces | Embedded. Available at: <http://www.embedded.com/design/embedded/4231144/The-challenges-of-multi-touch-gesture-interfaces?pageNumber=0> [Accessed 17 September 2012].

Izkovitch, A. (2012), Designing for Context on Multiple Devices. Available at: <http://mashable.com/2012/07/07/designing-for-multiple-devices> / [Accessed September 17. (2012].

Lai, K. (2012), A gesture-driven computer interface using Kinect. In *Image Analysis and Interpretation (SSIAI),*. 2012 IEEE Southwest Symposium on Image Analysis. Santa Fe, NM: IEEE, pp. 185 –188.

Long, A., Landay, J., & Rowe, L. (1999), Implications for a gesture design tool. In *Proceedings of the SIGCHI conference on Human factors*. CHI 99. Pittsburgh, Pennsylvania, USA: ACM.

Madrigal, A. (2012), iPhone 5? Yawn. What Will the “Phone” of 2022 Look Like? - Alexis C. Madrigal - The Atlantic. *TheAtlantic.com*. Available at: http://www.theatlantic.com/technology/archive/12/09/what-will-the-phone-of-2022-look-like/262300/ [Accessed 17 September 2012].

Rosenberg, A. (2012) Microsoft Patent Takes Aim At Holodeck Gaming - G4tv.com. Available at: http://www.g4tv.com/thefeed/blog/post/727820/microsoft-patent-takes-aim-at-holodeck-gaming [Accessed 17 September 2012].

Saffer, D. (2009) Designing Gestural Interfaces: Touchscreens and Interactive Devices. 2nd ed., Sebastopol: O’Relly Media.

Schramm, M. (2010). Kinect: The company behind the tech explains how it works | Joystiq. Available at: http://www.joystiq.com/2010/06/19/kinect-how-it-works-from-the-company-behind-the-tech/ [Accessed 17 September 2012].

Kessler, S. (2013) Microsoft Kinect Sales Top 10 Million, Set New Guinness World Record. [ONLINE] Available at: <http://mashable.com/2011/03/09/kinect-10-million/>. [Accessed 04 March 2013].

APPENDICES

###### User Evaluation Script

Adapted from Rocket Surgery Made Easy by Steve Krug

<http://sensible.com/downloads-rsme.html>

Hi, \_\_\_\_\_\_\_\_\_\_\_. My name is \_\_\_\_\_\_\_\_\_\_\_, and I’m going to be walking you through this session today.

Before we begin, I have some information for you, and I’m going to read it to make sure that I cover everything.

You probably already have a good idea of why we asked you here, but let me go over it again briefly. We’re asking people to try using a new way of interacting with websites that we’re working on so we can see whether it works as intended. The session should take about an hour.

The first thing I want to make clear right away is that we’re testing the *technology*, not you. You can’t do anything wrong here. In fact, this is probably the one place today where you don’t have to worry about making mistakes.

As you go through the tasks, I will sometimes ask you to try to think out loud: to say what you’re looking at, what you’re trying to do, and what you’re thinking. This will be a big help to us.

Also, please don’t worry that you’re going to hurt our feelings. We’re doing this to improve the technology, so we need to hear your honest reactions.

If you have any questions as we go along, just ask them. I may not be able to answer them right away, since we’re interested in how people do when they don’t have someone sitting next to them to help. But if you still have any questions when we’re done I’ll try to answer them then. And if you need to take a break at any point, just let me know.

You may have noticed the microphone and the cameras. With your permission, we’re going to record what happens on the screen, your actions and our conversation. The recording will be used to help us figure out how to improve the technology and demonstrate the different actions. And it helps me, because I don’t have to take as many notes.

If you would, I’m going to ask you to sign a simple permission form for us. It just says that we have your permission to record you, and the recording could be published along with the written analysis.

|  |
| --- |
| Give them a recording permission form and a pen  While they sign it, START the SCREEN RECORDER |

First, I would like to show you how to use the gesture-based interface.

[Demonstrate: Wave, Click, Swipe Left, Swipe Right, Scroll Up and Scroll Down]

You can practice to get used to the motions.

[Give a few minutes to allow user to feel comfortable]

Now I’m going to ask you to try doing some specific tasks.

TASKS:

On New York Times

* Please find and view the main news story about [ENTER NEWS TOPIC OF TOP STORY FOR THE DAY]. Then return to Homepage.
* Go to the Sports Section and then return to Homepage.
* View a video at the bottom of the page.

Repeat Steps using mobile NYTimes

On Facebook,

* Please select any photo in the news feed to view. Return to the news feed.
* Like any item in your news feed.
* View your Facebook Messages (email-like page)

Repeat steps using Mobile Facebook

On Gesture News

* Please find and view the top news story about [ENTER NEWS TOPIC OF TOP STORY FOR THE DAY]. Then return to Homepage.
* Go to the Sports section. Return to Homepage.
* View the Top Headlines Video

On Photo Slider

* Please select view and read the first story.
* View other stories
* Go back to a previous story.

Thanks, that was very helpful. Now, I need you to please complete this survey. Thank you for your help and your time.

###### User Survey

The users tested were asked to complete a survey about the experience. The image below shows the form they were asked to fill out.

