SPEED-ACCURACY COMPARISON OF NAVIGATIONAL INTERFACES

by Daniel A. Afergan

B.A. in Cognitive Science, May 2005, University of Pennsylvania

A Thesis submitted to

The Faculty of
The School of Engineering and Applied Science
of The George Washington University
in partial fulfillment of the requirements
for the degree of Master of Science

May 17, 2009

Thesis directed by

John L. Sibert

Professor of Engineering and Applied Science

Dedication

This thesis is dedicated to my mother, Marjorie Afergan, for instilling her love for learning in me and always willing me to succeed.

Acknowledgements

The author would like to thank Professor John L. Sibert, members of the examination committee, all of the participants in the study, and his family— his father Barry, his grandfather Bert, his brother Michael, and his sister-in-law Suzanne. He would especially like to thank his wonderful fiancée Rebecca Bortnick for being there for him during his entire thesis preparation.

Abstract

Speed-Accuracy Comparison of Navigational Interfaces

The goal of this research is to test the effect of different computer interfaces on the amount of time it takes a user to move a cursor from a start point to a target, using Fitts' Law, a model that describes the performance of pointing of input devices. Participants in a study used a mouse, Xbox 360 controller, and Nintendo Wii remote to point at and select target regions. The goal is to see the effects of interface, distance to the target, and target width on movement time, information throughput, and hit rate. Additional path metrics and the speed-accuracy tradeoff will be covered.

Table of Contents

List of Figures

List of Tables

1. INTRODUCTION

1.1 Human-Computer Interaction

Human-computer interaction is the study of how people use computers. An important part of this field is the use of interfaces. Interfaces are the set of objects that receive input from, or give input to, the user. This includes the display, input devices, auditory and sound cues, and any other systems with which the user can control the application. They are an important part of any simulation because how the user controls the environment directly affects the user's experience. All interfaces must directly transmit messages to the computer, and interfaces can range from a simple keyboard and mouse to fully immersive systems. Software applications are crafted according to users' needs, and in turn interfaces are developed with the functionalities of the software package in mind. This interest has led to the development of the field called human-computer interaction, which focuses on the study of the interaction between humans and computers. Olson and Olson define human-computer interaction as "the study of how people interact with computing technology," which involves the cognitive, perceptual, and motor components of the encounters between a human and a computer (Olson and Olson 2003).

The purpose of this research is to test the effect of different computer interfaces on the amount of time it takes a user to move a cursor from a starting point to a target, using Fitts' Law, a model that describes the performance of pointing of input devices. Participants in this study used a mouse, Microsoft Xbox 360 controller, and Nintendo Wii remote (Wiimote) to point at and select target regions. The goal is to see the effects of interface, distance to the target, and target width on movement time, information throughput, and hit rate. Additional path metrics and the speed-accuracy tradeoff will also be discussed.

1.2 Interfaces

An interface is defined as any device that receives input or gives output. For most computers, a pointing device plays an integral role in interacting with the graphical user interface (GUI), a display that consists of windows, icons, menus, and pointers. Most operating systems rely on a pointer device for input. The user navigates the pointer to a target in order to perform a task, and repeats this process several times for each session. Because of the amount of time that a user spends manipulating the pointer, it is important to maximize pointing performance. While goal-directed aiming was first researched in 1899 by R.S. Woodworth, the kinematics of pointing developed into an important field after the Industrial Revolution and both World Wars, when the performance of a human operating a system gained importance (Göktürk 2000).

When considering cursor movement and on-screen navigation, the two most important metrics are the amount of time it takes for a user to reach his or her intended target, and the success rate of the user selecting the target area. By studying how different interfaces fare when selecting targets, we can determine which interfaces allow what functionality and how well users can perform using them.

As the current information age is in the process of expanding to ubiquitous computing, it is important to consider the interfaces involved in daily usage. Humans rely on interfaces to give input to computers, and moving a cursor plays a central role in daily computing. However, there are many input devices that can be used to manipulate a cursor, and based on the specific task, some may be more appropriate than others. Because computer and video game interfaces are now commonplace, it is important to study how various computer and video game interfaces are able to perform on-screen navigation to targets.

The video game industry is becoming an extremely influential realm, and video game consoles are now able to provide many functions beyond just entertainment. The latest version of consoles (Nintendo Wii, Microsoft Xbox 360, and Sony PlayStation 3) all offer wireless Internet capabilities and some form of Internet browser. Additionally, each has a complex menu and functions beyond the simple start/pause menus of early consoles. As a result, users must be able to navigate on-screen and easily select between different options. While the Xbox 360 and PlayStation 3 use an analog control stick, also called a thumbstick, to select different options, the Wii implements a cursor similar to a computer. Video game consoles are extremely popular as well— in 2008, the computer and video game industry grossed over $22 billion and as of the end of March 2009 (MarketWatch), Nintendo had sold roughly 50 million Wiis, and nearly 28.5 million Xbox 360s and 20 million PlayStation 3s had been sold worldwide (New York Times).

As computer video functions and quality continues to improve, television is becoming a central unit for all forms of computing, with multiple menus. Cable menus and digital video recorders necessitate the use of menu systems during television viewing, and television monitors allow one to switch between watching television, seeing a computer screen, and using a video game console with the push of a button.

As a result, the crossbreed of interfaces for multiple functions must be considered in order to seamlessly integrate the devices. As wireless information techniques become commonly accepted, these devices can communicate with multiple interfaces. However, just because this exchange is possible, it does not mean that it is ideal. Different interfaces are designed with different goals in mind, and each has its own limitations.

2. LITERATURE REVIEW

2.1 Fitt's Law

One of the fundamental laws of the human motion is Fitts' Law. Fitts' Law is a well-proven human-computer interaction (HCI) model that describes the performance of pointing of input devices. Fitts' Law measures pointing performance, which can be defined as "the measure for the speed and accuracy characteristics of pointing movement for a particular subject, input device, or task" (Göktürk). While it is used and supported by many prominent HCI scientists, Fitts' Law is still not universally accepted (Kong and Ren 2007). Fitts' Law is a psychological model of movement; it predicts the amount of time it takes to point to a target. While it has been adapted over time, it has always been used to describe the relationship between movement time and the variable target distance, width, and interface constants. As Kelso explains, "the relation between amplitude, movement time, and precision (or tolerance) has come to be known as Fitts's Law because of its wide applicability to different perceptual-motor tasks" (Kelso 1992). This research was pioneered by Air Force Lieutenant Colonel Paul Fitts, founder of the Air Force Research Laboratory. Lt. Col Fitts measured the time it took participants to perform simple movement tasks; alternating tapping metal plates, moving washers from one pin to another, and transferring pins from one set of holes to another (Fitts 54). The first task remains the most famous; Fitts asked participants to use a stylus to alternate tapping two metal plates on opposite sides of an apparatus, as pictured in Figure 1, as quickly and accurately as possible. Participants performed this task under two conditions: with a stylus weighing one ounce and with a stylus weighing one pound. Although he proved the validity of his claim using such simple mechanical procedures, this law has been applied to computer use and computer navigational input devices.

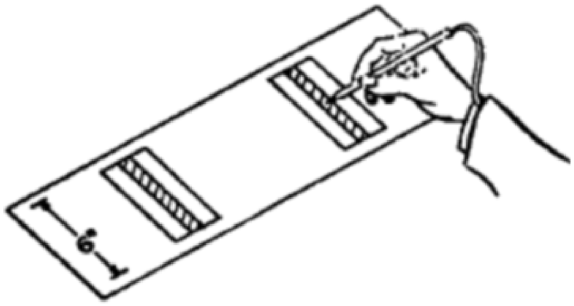

Figure 2-1: Fitts' tapping apparatus

Fitts based his findings off an information processing theory, Shannon's Theorem 17, which states that "the capacity of a channel of band W perturbed by white thermal noise of power N when the average transmitter poser is limited to P is given by

$$C = W\ log(\frac{P+N}{N}) \qquad (2.1)$$

" (Fitts). This was the basis for his index of difficulty (ID), designed to measure the complexity of a task, and movement time (MT) formulas; the latter, Equation 2.3, is now known as Fitts' Law.

$$ID = -log_2\left(\frac{W_S}{2A}\right) \tag{2.2}$$

$$MT = a + blog_2\left(\frac{2A}{W}\right) \tag{2.3}$$

In these formulae, W is the target width (tolerance range) in inches and A is the amplitude (distance) of the movement. *a* and *b* are interface constants, with *a* describing the start/stop time of the device and *b* being correlated to interface speed. ID is measured in terms of bits per *response*, while MT is measured in seconds. This general formula has been altered for many purposes, but the most stable and widely used version is MacKenzie's 1992 formula, describing MT as:

$$MT = a + b\ log_2\left(\frac{A}{W} + 1\right) \tag{2.4}$$

While the formula has been adapted over time, the fundamentals of it remain the same. However, now the measures are most widely accepted in the form:

$$ID = log_2\left(\frac{A}{W} + 1\right) \tag{2.5}$$

$$MT = a + b\ ID \tag{2.6}$$

Fitts' Law can be used as an evaluation method for novel interfaces, but can also predict the amount of time required to perform a task, as well. Instead of using Fitts' Law to predict MTs, if MTs are measured under a variety of conditions, we can extract the coefficients in Fitts' Law and use them to predict performance.

2.2 Effective Index of Difficulty

While Fitts' Law properly measures performance for a given task, it measures how the users are supposed to perform with the given distance and width, rather than how they actually utilize the target space. A widely accepted variation of the ID, called the "effective index of difficulty" or ID_e, takes the spread of selections points into account. It is defined as:

$$ID_e = log_2\left(\frac{A}{W_e} + 1\right) \tag{2.7}$$

The denominator W_e is the effective target width and is defined as 4.133 * SD_x, where SD_x is the standard deviation in selection coordinates along the target path. 96% of the target selections should fall within this range. While the classic Fitts' Law formula uses variables to determine difficulty according to the task presented to the subjects, this variation allows us to concentrate on the difficulty of what the participants actually achieved. Murata proposed this idea, stating that "This means that the ID in a two-dimensional pointing task cannot be determined properly using the displayed size W. Rather, it may be proper to determine the ID using the distribution of x and y coordinates of response points" (Murata 1999). We can also use the effective distance (A_e) which is the average distance between the start and target points, to calculate the ID_e which makes the equation:

$$ID_e = log_2\left(\frac{A_e}{W_e} + 1\right) \tag{2.8}$$

This enables us to find out the index of difficulty of the motion that the user actually performed rather than the task that they were given.

2.3 Throughput

Throughput is a measure that measures the combination of speed and accuracy across trials. There exists an ISO standard, ISO 9241-9 (ISO 2000), that proposes using throughput as the main metric of performance, defined as

$$TP = \frac{ID}{MT} \tag{2.9}$$

in bits per second. Since speed and accuracy are averaged over all of the ID conditions, they both play a role in the computation of throughput. This keeps the fundamental variables involved in Fitts' Law, but essentially ignores the coefficients. However, since it averages ID/MT from different ID points and depends on the ID target settings, it defeats the purpose of using Fitts' Law (Zhai 2004). As a result, MacKenzie and Isokoski came out with a revised model of throughput

$$TP_e = \frac{log_2(\frac{A_e}{4.133 * SD_x}+1)}{MT} \tag{2.10}$$

This is the essentially the second form of ID_e, using both the effective distance and effective width, divided by movement time.

While this is the most accurate form of throughput to date, using a different formula for large targets might be more appropriate since users tend to click near the closest edge of a large target rather than the center, thus the distribution tends to be unbalanced (Kong and Ren). TP remains constant despite changes in the speed-accuracy tradeoff. Although MT and hit rate change, the value for throughput remains constant despite if users aim to maximize speed or accuracy (MacKenzie and Isokoski 2008).

Because tasks and conditions vary so much, there are no accepted values for TP. Soukoreff and MacKenzie compiled 24 papers from the early 1990s to present day that analyze mouse performance and use a Fitts' Law model, and found that TP ranged from 2.55 bps to 12.5 bps. However, nine studies that all used the ISO9241-9 standard, performed an adjustment for accuracy, and calculated throughput using Equation 2.9 only had a variation of 1.2 bps, from 3.7 to 4.9 bps. The isometric joystick and touchpad also benefitted from a reduction of throughput variability by using this standard and procedures (Soukoreff and MacKenzie 2004). By establishing a standard formula for throughput that promotes accuracy, these authors hope to bring a global benchmark to the community.

2.4 Other Performance Metrics

While a simple linear regression will find the coefficients of Fitts' Law, there are a number of other metrics that can be used to measure pointing device performance. Although there has been a push to use TP as the main measure of performance, MacKenzie, Kaupinnen, and Silfverberg (2001) also suggest a series of pointer path metrics.

In their "Accuracy Measures for Evaluating Computer Pointing Devices," MacKenzie, Kauppinen, and Silfverberg propose a set of measures designed to characterize movement path beyond the amount of time it takes to reach the target. They propose seven different metrics of accuracy: (1) target re-entries, the number of times the pointer enters, leaves, and re-enters the target; (2) task axis crossings, the number of times the pointer moves across the task axis (the straight-line ideal movement path); (3)

movement direction changes, the number of times the pointer changes direction relative to the task axis; (4) orthogonal direction changes, the number of times the pointer changes direction relative to the axis orthogonal to the task axis; (5) movement variability, the standard deviation from the task axis; (6) movement error, the average deviation from the task axis regardless of direction; and (7) movement offset, the mean deviation from the task axis. These metrics analyze the entire journey, not solely the destination. The authors verify the validity of these metrics by finding differences in these metrics between interfaces even though the throughput of these devices is similar. However, only target re-entries and movement offset yield significant differences across interfaces. It should be noticed that the first four measures are discrete measures, while the last three are continuous.

Accuracy is best measured in terms of the hit or success rate, the percentage of time that a user clicks in the target area. Hit rate is important in evaluating any interface because it shows how much control one has with the device, and missing a target often results in additional time and steps to rectify the misclick. However, one must consider how they treat the target area and if they take advantage of the entire target area. The index of utilization (I_u) measures the spread of the endpoints and how well one uses the entire target area. I_u is formally defined as

$$I_u = log_2\left(\frac{W_e}{W}\right) \tag{2.11}$$

and thus measures the amount that the one utilizes the effective width of the target (W_e = 4.133 * SD) versus the actual size of the target. A positive value for I_u indicates that one over-utilizes the target area and misses the target area with more than 4% probability, while a zero value for I_u means that the user takes advantage of the target area perfectly, and a negative value means that the user under-utilizes the target and leaves a safety margin (Zhai, Kong, and Ren 2004). By analyzing the I_u across interfaces, we can see how well users take advantage of a target region.

2.5 Speed-Accuracy Tradeoff

Stemming from Fitts' Law is the speed-accuracy tradeoff. This is one of the earliest relationships studied in aimed movements, stemming from Woodworth in 1899. One sacrifices accuracy performance as speed increases, and conversely a greater focus on accuracy leads to a decrease in velocity. Fitts' ID is based on this ratio, as the target width (which requires greater accuracy) and distance (which requires greater velocity) maintain an inverse relationship-- assuming constant MT. Humans must try to find the equilibrium, where they can maximize speed without causing an unacceptable degradation in performance. While the underlying mechanisms behind the tradeoff remain universal, the decisions vary from person to person according to motor skills and personal preference. In actuality, "speed–accuracy relationship in pointing also contains another subjective layer which depends on how obediently the performer complies with the specified target width and what bias the performer takes (toward either accuracy or speed)" (Zhai, Kong, and Ren). The target size and distance affect movement in different ways, as "amplitude primarily determines acceleration duration and peak velocity, while target width has almost no effect on acceleration duration and peak velocity but affects the duration of the deceleration phase" (Guignon, Baraduc, and Desmurget 2008). This means changing the distance will lead to a change in how fast the user moves to the target, while changing the target size affects how the user slows down at the end of the task.

Studies show that participants utilize a mental model of Fitts' Law and demonstrate "an intimate knowledge of prospective motor actions" (Young et al. 2007) in order to consistently choose movements that require the least amount of time. The development of this inner mechanism begins at an early age and performance increases over time. Infants as young as seven months display kinematic pattern differences. For example, infants reach more slowly when reaching for small objects, and their precision improves between months seven and eleven (Zaal and Thelen 2005). Lambert and Bard's 2005 study found that visuomanual control increases throughout child development, while MT and trajectory path variability in a Fitts' task steadily decreases from ages six to ten.

Recent neuroimaging studies show that different areas of the brain are activated according to whether a task emphasizes speed or accuracy. However, the mechanisms behind the tradeoff are not well understood. It has been shown that emphasizing speed affects both the premotor decision process and the following motor activity decision processes (Rinkenauer et al. 2004). There is an increase in brain activity in areas related to response preparation and execution demonstrated when subjects were asked to emphasize speed over accuracy, most notably the "premotor areas of the frontal lobe, the basal ganglia, the thalamus, and the dorsolateral prefrontal and left parietal cortices" (Van Veen, Krug, and Carter 2008). This research validates the claim that speed-accuracy tradeoff originates during response planning, and not from when the subject sees the stimulus.

2.6 Fitts' Law Task Factors

There are many factors that must be taken into consideration when planning and running a Fitts' Law task. The selection method and phase play a significant role in performance. Selection method can be defined as the way to inform the computer to ensure that the current location is the intended location, and selection phase is the "portion of overall pointing movement after the positioning movement ends completely but before the computer is informed that the positioning is ended (by means of clicking, speaking, hitting, dwelling etc.)" (Göktürk). However, as long as both of these remain constant throughout the experiment, it should add a constant amount of time, which does not affect the ID.

While it is important to keep the task variables constant and controlled, it is equally important to consider the human usage factors. For Fitts' Law tasks, the environment plays a significant role in performance. Posture, display gain, and muscle groups involved in controlling the interface all can affect MT and TP.

This applies to tasks beyond just linear movement. While Fitts and MacKenzie described a formula for one-dimensional tasks, Fitts' Law has also been generalized to two dimensions, where users must move both horizontally and vertically (MacKenzie and Buxton 1992), and three dimensions, such as in driving tasks (Accot and Zhai 2001) and additionally it has been changed to add a gain parameter (Radix, Robinson, and Nurse 1999). Furthermore, the law applies for action perception, meaning that people can accurately judge MTs of arms they are imagining or watching, even if these arms are robotic, prompting Grosjean to declare it "the first motor principle that holds in imagery and the perception of biological and nonbiological agents" (Grosjean, Shiffrar, and

Knoblich 2007). For this paper, we will limit the scope of pointing to the hand and arm, including other joints within the limb.

The effects of posture on movement cannot be ignored during Fitts' Law tasks. Subjects adapt their behavior according to their posture and the settings around them. MT to reach small targets (using a pointing stylus) increases when the participant is standing (there is no effect for large targets), and there is a coordination between the user's trunk and arm (Berrigan et al. 2006). According to this research, balance control is affected when trying to make precise movements, which does not affect sitting users because they have a stable base. In addition, target orientation plays a role in hand/arm orientation and plays a significant role in the premotor planning of movement and response time (Fan, He, and Tillery 2006).

Of additional importance is the interplay of limbs. Kelso states that "the brain produces simultaneity of action not by controlling each limb independently, but by organizing functional groupings of muscles that are constrained to act as a single unit" (Kelso). Because of the fundamental neuromuscular differences and levels of control between different parts of the body, not all of the parts of the arm will perform the same on a Fitts' Law task. Balakrishnan and MacKenzie found that the wrist and forearm have similar throughputs (of 4.1 bps) and outperform the index finger (TP = 3.0); however, the thumb and index finger working together have a higher TP than any of these conditions, with a rate of 4.5 bps (Balakrishnan and MacKenzie 1997).

Along with the different muscles and joints used in interface control, the locomotion method between interfaces can also vary greatly. The most common navigational interfaces are the mouse and joystick. A number of studies involve both of these interfaces, and interface comparison charts consistently show better performance with the mouse than the joystick (Göktürk, MacKenzie and Isokoski, Sohn and Lee 2004). However, there is always a search for more effective interfaces, both by HCI scientists as well as inventors. Recent research shows that a two-handed trackball-mouse can outperform a mouse or one-handed trackball-mouse, but the researchers admit that this only holds true under certain conditions, and that it is not possible to integrate a two-handed trackball-mouse and keyboard operation at the same time (Isokoski et al. 2007). Isokoski et al. admit that the standard mouse is the most efficient of these interfaces.

2.7 Interfaces

While the mouse and joystick are the two most popular interfaces that have been studied in relation to MT and TP, one new type of interface that is rapidly gaining popularity is the user interface wand. The Gyromouse is a hybrid mouse (produced by Gyration) and UI pointing wand, and many devices such as the Nintendo Wii remote, Philips Research UI Wand, and PHANTOM haptic stylus allow the user to point at the screen and have a direct relationship between actual movement and on-screen movement. All of these wands function by recognizing infrared LEDs to detect position. However, this requires a line-of-sight with the display, and occlusion by an object or person can ruin tracking. These interfaces have a few advantages—they do not need a surface to operate on, can operate in a sterile environment (such as an operating room), and translate physical movement to on-screen movement in an intuitive fashion. Also, they provide an extra degree of freedom by allowing tilting motion. Also, the sensitivity of the wand can be adjusted by changing the distance between the interface and display because of the angular change.

While both the Gyromouse and UI Wand perform worse in terms of MT, ID, and hit rate than a mouse (with the Gyromouse outperforming the UI Wand), the qualitative performance is different (Stefels et al. 2007). Participants also rated that they enjoyed using the UI Wand more than the Gyromouse, and enjoyed both of them more than a standard mouse, and even believed they were quicker with the UI Wand (than the Gyromouse). Sohn and Lee evaluated performance of the SonarPen wand in Fitts' Law tasks compared to a touchpad and trackball using the ISO 9241-9 standard and found TP to be very similar. They evaluated the TP to be 2.87 bps for the SonarPen, compared to 2.70 bps for the joystick and 2.97 bps for the trackball (Sohn and Lee 2004). By comparison, mouse TP tends to be in the 3.7 to 4.9 bps range (Soukoreff and MacKenzie).

Fitts' Law tasks are designed to measure performance of targeting and pointing motions. By controlling the target amplitude and width, we can figure out how effective interfaces are for pointing tasks. This has implications in interface design, because designers must layout their icons accordingly, minimizing distance yet maximizing size. While many metrics can be used to evaluate performance, the latest TP models are providing the most normalization, and other metrics can be used to evaluate the user's path. These must be measured taken into account when designing future interfaces.

3. CONTRIBUTIONS

The purpose of this research is two-fold: to evaluate performance of navigational interfaces and to evaluate existing metrics. Very little research has been done regarding the Wiimote, especially assessing its cursor performance. In order to evaluate the Wiimote in comparison to other interfaces, one must measure all of them under the same conditions since performance varies according to the task and conditions under which the task is performed.

3.1 Performance

Movement time and hit rate will be used as the main metrics to compare performance across the interfaces. These measure speed and accuracy independently, so TP will be evaluated in order to compare the participants' overall speed-accuracy operation. In addition, the index of utilization will describe how well participants take advantage of the target size.

Path performance will also be measured. These seven measures will indicate how much the participants' movement patterns diverge from a perfect line from the start point to the target. Target re-entries, directional changes, and deviation from the ideal path will all be taken into account.

3.2 Metrics

One of the drawbacks of Fitts' Law research is that there is no standard for measuring and analysis of the data. There are multiple forms of the standard Fitts' Law equation and multiple ID and TP models. The results of this study will indicate which models capture performance most accurately. Also, since these movement metrics proposed by MacKenzie, Kauppinen, and Silfverberg have not been proven or disproven by any other researchers, this study aims to verify or disprove them.

3.3 Initial Hypotheses

The majority of published research indicates that the mouse has a higher throughput than joysticks/thumbsticks, with UI wands having an even lower throughput. It is expected that this relationship will remain in this study. In terms of path performance, the original paper shows that the mouse yields better results for all seven of the metrics than the joystick. However, the Xbox 360 analog stick can be pushed into place such that the cursor moves in a straight line, which would produce superior results, whereas the forearm cannot move in a perfectly straight line and must rotate around the elbow, causing the mouse to move in an arcing path. In terms of the Wiimote, the outstretched arm can be moved so that it points in a straight line, but it is difficult to keep the arm steady. Thus, it will be interesting to see which of the interfaces performs best for path performance.

4. METHODS

4.1 Experimental Setup

This experiment was set up to give all participants the same experience across three interfaces, while testing the effects of the dependent variables, distance and width. The goal was to test participants using a standard Fitts' Law experiment in order to measure performance within a well-understood paradigm.

4.2 Interfaces

The three interfaces used in the study were a computer mouse, Microsoft Xbox 360 controller, and Nintendo Wiimote, as pictured in Figure 4-1. The mouse is a pointing device that users drag across a flat surface, where forward or backward movement cause the cursor to move up or down, respectively, and moving the mouse left or right translates the cursor on screen in the same direction. The Xbox 360 controller is designed to be held with two hands, with an analog control stick, often called a thumbstick, for the left hand that allows a user to cradle his or her thumb on top of the stick and move the cursor with minimal thumb movement. This is a variation of a joystick but works the same way, with forward and backward movement being converted to vertical cursor movement, and horizontal stick movement corresponding to horizontal cursor movement. The Wiimote is an input wand, which users point at a screen while their physical movement in any direction is translated to cursor movement.

Figure 4-1: Mouse, Xbox 360 controller, and Wiimote

One main difference to consider across all three of these interfaces is the muscles involved in manipulating them. While the Xbox 360 controller needs only thumb movement, the mouse is controlled primarily by forearm movement (and hand movement for finer actions). The Wiimote is controlled primarily by the hand and wrist, but uses the forearm for larger movements. This gives the latter two much less stability than the controller, and a different speed-accuracy tradeoff.

4.3 Procedure

Twelve participants (9 male, 3 female) were recruited via e-mail. The mean age was 26.3 years (SD = 3.67). All participants were regular users of a GUI and mouse, reporting that they use a mouse 30+ hours/week, and almost all indicated they had rarely or occasionally used a Wiimote and an Xbox 360 controller-style joystick. Full demographic information can be found in Appendix C.

After reading instructions for the task, the participants were given one minute to practice navigating the cursor on-screen and become familiar with movement for each interface. The task environment consisted of two bars extending from the top to the bottom of the screen on opposite sides of the screen. Figure 3-2 below shows a screen capture of the simulation under the 600 pixel distance / 30 pixel width condition. The red crosshair in the center of the right bar indicates that it is the target.

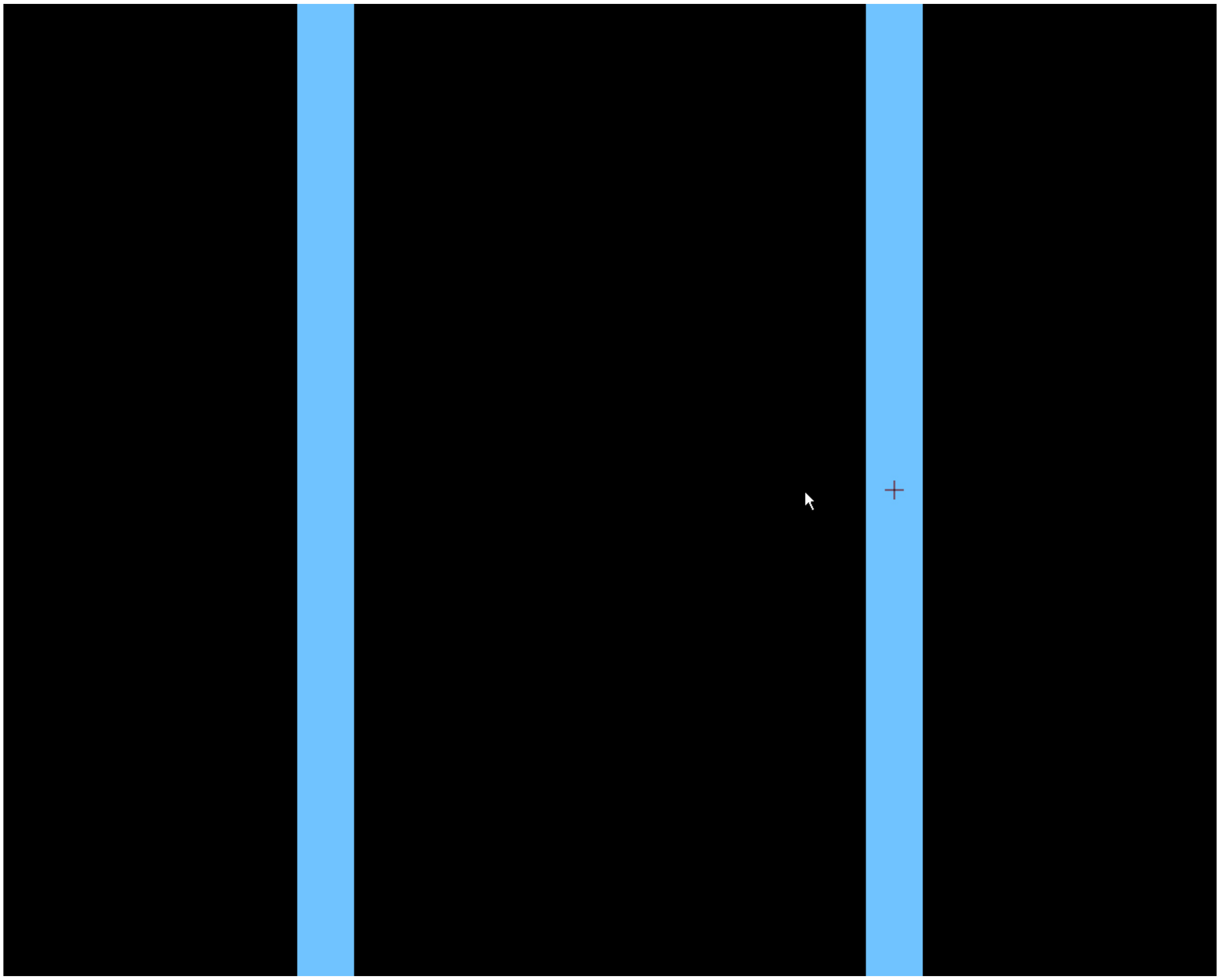

Figure 4-2: Screenshot of simulation with cursor.

For each block, participants moved the cursor horizontally between the two targets bars and clicked on them for 25 iterations, with a crosshair in the center of the target bar. The standard Windows cursor image was used for all conditions, with the tip of the arrow being used as the cursor position. The cursor always began in the center of the left bar, with the right bar being the target. Participants were directed to move as quickly and accurately as possible and to aim for the center of the target bar. Consistent with the literature, they were instructed that the center of the crosshairs was a perfect hit and to aim for the center, but that hitting anywhere inside the width of the target was considered a successful hit (MacKenzie and Isokoski 2008). Each participant used all three interfaces, but the order of using them was varied using a 3x1 Latin Squares method. The participants were first presented with a practice block at the 30 pixel width/600 pixel distance condition before completing nine blocks (one in each possible condition) in a random order. In all, the 12 participants used 3 interfaces, and ran 9 blocks of 25 trials for each.

4.4 Design

A trial consisted of the user moving from one bar to the other and pressing the input button, and was considered complete once the user released the input button. The width of the bars and distance between the bars varied between blocks, with widths of 15, 30, and 45 pixels and distances of 300, 600, and 900 pixels between the center points. The task was implemented using Microsoft XNA 3.0 Game Studio, a C# framework, and Microsoft Visual C# Express Edition. It ran off an Intel Core 2 Duo laptop running Microsoft Windows XP, which was plugged into a power source at all times during the testing, and used a Sun 24-inch monitor as a display. The simulation ran at an average of 60 Hz, and recorded data, including cursor position, timestamps, and other summary information at the end of each cycle.

Efforts were made to simulate the conditions under which participants would normally use the interfaces. One issue was that a mouse is typically used with computer monitors, while the Xbox 360 controller and Wiimote are often used on a larger television screen. While this should not affect mouse or Xbox 360 controller movement, the size of the screen directly impacts the ratio of physical movement to on-screen movement. Therefore, the 24-inch monitor allows the participants to feel as if they are using a computer screen, while still being the size of a small television screen. Participants were situated in front of a conventional workstation, and had space to move in order to feel as comfortable as possible.

Tethering was another issue that needed to be addressed. The mouse is typically tethered to a computer (although there are many wireless mice), while Wiimotes are wireless, and Xbox 360 controllers can be wireless or tethered to a system. However, since the experiment is designed to test how users perform in an actual environment, a wireless Xbox 360 controller was used since the Xbox 360 console supports wireless connectivity, only requiring the user to plug in the controller to charge it. Both the Xbox 360 controller and Wiimote used Nyko rechargeable battery packs and were fully charged between participants.

The Wiimote uses a three-axis accelerometer to detect how the Wiimote is rotating and an optical sensor to detect a sensor bar with ten LEDs to determine relative positioning and where the Wiimote is pointing. In order to keep tracking consistent with Wii console performance, a Wii was plugged in (behind the experiment setup) and the LED sensor bar was stationed directly below the monitor. Because of this, the tracking for the Wiimote condition was identical to that used for the Wii console.

4.5 Interface Implementation

While the simulation stayed the same for all of the interfaces, they all required a different setup in order to interact with the software.

4.5.1 Mouse Condition

A Logitech USB optical mouse with two buttons and a scroll wheel was used for the mouse condition. XNA has inherent support for the mouse, so no coding was necessary in order to move the cursor. Because of this, the mouse uses the same movement patterns as in the native Windows environment. Participants clicked the left mouse button once they reached the target.

4.5.2 Xbox 360 Controller Condition

An Xbox 360 controller with a Windows wireless adapter was purchased in order to communicate with the computer. Because they are both Microsoft products, XNA supports input from Microsoft Xbox 360 controllers. There were only 2 lines of code (plus a declaration statement) added in order to manipulate the cursor according to the Xbox 360 controller input:

```
MouseState mState;
mState = Mouse.GetState(); //used to determine the current position of
//the mouse
Mouse.SetPosition((int)(mState.X + 9 *
input.GamePads[0].ThumbSticks.Left.X), (int)(mState.Y - 9 *
input.GamePads[0].ThumbSticks.Left.Y));
//moves the cursor according to the current X/Y position of the
//thumbstick, up to 9 pixels/cycle on both the X and Y axes
```

Each cycle, the Xbox controller indicates the thumbstick positioning on X and Y scales from -1.0 to 1.0. This rate is then multiplied by 9 (and truncated), and the mouse moves that many pixels in the appropriate direction. Although the value is truncated, this is on the scale of one pixel, and since the refresh rate is so high, the motion of the cursor still looks natural and continuous. Also, since the participants moved the cursor in both directions, the movement gained and lost is canceled out over the course of the block. This allows a user to move at a maximum of 9 pixels in each direction per cycle, which means that the user can move a maximum of 540 pixels per second on each axis. The input button for the Xbox 360 controller was the A button, located on the right side of the controller. The difference between the Xbox 360 controller and the other two conditions is that there is a maximum velocity and regular rate for Xbox 360 controller.

4.5.3 Wiimote condition

In order for the simulation to read input from the Wiimote, it had to be connected to the computer. The Wiimote uses Bluetooth to communicate, so a Cables Unlimited USB to 100 Meter Class 1 Bluetooth Adapter dongle was attached to the computer in order to communicate with the Wiimote. The BlueSoleil 6.4.249.0 driver software was used to pair the laptop with the Wiimote and implement the Bluetooth stack. A full license was purchased in order to not be restricted by the 5 MB data limitation of the trial version.

Once the computer could accept the Wiimote input, it needed to be converted into mouse movement data. Microsoft endorses and publicizes a Wiimote .NET library, named WiimoteLib and written by Brian Peek (http://www.codeplex.com/WiimoteLib), which provides an application programming interface (API) that allows programmers to access Wiimote functions and data structures. Although it seems strange that Microsoft can implement one of its competitors' controllers, the Wiimote can only be connected to a PC and not the XBox 360, so this does not infringe on its video game console in any way. Since the API is written in C#, one must simply add the library to one's development project and then call the necessary functions. The Wiimote transmits both its accelerometer data and infrared sensor data each cycle, allowing for mouse repositioning at the same speed as the rest of the simulation. Participants used the A button, located on top of the Wiimote and used as the primary input button

While this functionality was implemented into the testing and the program detected the mouse movement, it was decided that an existing program, WiinRemote (http://onakasuita.org/wii/index-e.html) did a better job of translating the Wiimote movement into cursor movement. The WiinRemote application allows one to use the Wiimote to control the cursor. By using this program, the code for the Wiimote condition was identical to the mouse condition. However, there was the addition of a calibration screen, where the cursor was hidden and the participant pointed at a crosshair on an otherwise blank screen (corresponding to the center of the left bar, where the cursor would start) in order to match the participant's arm position with the mouse position.

The Wiimote is also a first-order input device, meaning that it is rate-based and the more it is rotated (via wrist movement) in the direction that the Wiimote is moved, the greater the velocity of cursor movement. Research suggests that using the Wiimote as a zero-order input device (only controlling cursor position) yields faster movement times than the ability to control position and velocity, with no change in error rate

(Campbell et al. 2008). However, the Wiimote is usually used as a first-order input device, so it would be useless to test its performance as a zero-order interface.

5. RESULTS

5.1 Data Collection

Upon analyzing the data, the first trial of each block displayed irregular data, because the mouse cursor moved on its own and the participants did not know where the starting and target bars would be. Due to these irregularities, the data from the first trial of each block was discarded. Since the twelve participants each used three interfaces for nine blocks of 24 trials, this gave a total 12x3x9x24 = 7776 data points. Additionally, any trials over 13 seconds or with a distance of zero pixels (indicating a double-click) were discarded. There were eight trials total over 13 seconds, all for the Wiimote condition, and they were the result of either the user losing tracking for the Wiimote or the Wiinremote application occasionally crashing. There was only one double-click, but the trial after this double-click also had to be discarded since the cursor was already next to the target. This yielded a total of 7766 trials, and a total of 324 block averages for each metric.

5.2 Statistical Analysis

In order to measure the variance, a general linear model with repeated measures ANOVA (analysis of variation) test was used. This allows us one find if interface, distance, and width have a significant effect on all of the metrics—including MT, hit rate, index of difficulty, TP, and the previously mentioned path metrics. Table 5-1 shows the averages across interfaces (averaging all of the subjects and conditions together), while Table 5-2 shows the significance levels of the independent variables across each of the metrics. Because of variable levels of sphericity, the Greenhouse-Geisser correction was made to all of the measures. This resulted in more conservative levels of significance, and thus higher *p* values. The significance was tested at a 95% confidence interval, meaning that *p* values less than .05 indicated significance.

5.3 Data Tables

Table 5-1: Means and Standard Deviations of Accuracy Measures

	Mouse		Wiimote		Xbox 360	
	Mean	SD	Mean	SD	Mean	SD
MT (sec.)	1.12	3.3	2.54	1.26	2.05	0.87
Hit rate (%)	.96	.07	.64	.26	.83	.26
ID_e (A/W_e) (bits)	3.84	1.74	5.37	1.83	4.13	1.25
ID_e (A_e/W_e) (bits)	3.94	1.74	5.03	1.43	4.01	1.33
TP (A/W) (bits)	4.16	1.11	2.09	.90	2.49	1.00
TP_e (A/W_e) (bits)	3.75	1.94	2.50	1.15	2.19	.60
TP_e (A_e/W_e) (bits)	3.87	1.07	2.36	1.07	2.12	.63
Task axis crossings	1.93	.52	1.52	.56	.64	.41
Target re-entries	.19	.28	.13	.13	.19	.23
Movement error (pixels)	14.30	3.53	123.31	40.56	17.22	75.61
Movement offset (pixels)	9.01	2.60	91.384	38.43	12.44	72.11
Movement variability (pixels)	8.96	9.56	103.05	25.30	10.30	19.91
Movement direction changes	1.92	.65	1.90	1.11	.12	.15
Orthogonal direction changes	.92	.36	1.86	1.24	.60	.42
Index of utilization	2.71	2.53	-.67	1.56	.30	1.32

Distance from center (pixels)	8.32	6.85	53.91	43.17	18.52	15.06

Table 5-2: P-values by Interface

	Interface	Distance	Width	Interface * Distance	Interface * Width	Distance * Width	Interface * Distance * Width
MT	.00**	.00**	.00**	.00**	.04*	.57	.59
Hit rate	.00**	.00**	.00**	.00**	.00**	.31	.46
ID_e (A/W_e)	.04*	.00**	.22	.73	.19	.34	.59
ID_e (A_e/W_e)	.11	.00*	.67	.40	.42	.45	.49
TP (A/W)	.00**	.01*	.00**	.00**	.00**	.12	.44
TP_e (A/W_e)	.03*	.02*	.01*	.03*	.06	.67	.11
TP_e (A_e/W_e)	.00**	.01*	.03*	.12	.38	.36	.30
Task axis crossings	.00**	.00**	.60	.26	.42	.52	.23
Target re-entries	.34	.00**	.00**	.01*	.03*	.17	.71
Movement error	.06	.18	.27	.35	.23	.59	.73
Movement offset	.12	.22	.28	.38	.25	.61	.74
Movement variability	.01*	.17	.40	.26	.35	.55	.55

Movement direction changes	.00**	.00**	.15	.00**	.31	.55	.56
Orthogonal direction changes	.00**	.00**	.16	.00**	.22	.77	.77
Index of utilization	.00**	.02*	.00**	.16	.27	.40	.35
Distance from center	.00**	.11	.15	.24	.09	.90	.85

*p < .05 ** p < .001

5.4 Significant Effects

These tables yield interesting results. They show a significant effect of interface for most of the variables tested.

5.4.1 Speed-Accuracy Measures

Interface, distance, and width all had a significant effect on MT, as Fitts' Law dictates, but there were also significant interactions between interface and distance and interface and width, meaning that there were additive effects of changing interface and one of the other independent variables. The mouse was the fastest of the interfaces, and had a significantly higher hit rate than either of the other interfaces. The Wiimote fared the inverse, taking the most time to reach the target and having the lowest hit rate. Figures 5-1 and 5-2 below illustrate sample data of MT and hit rate by interface. Both of these correspond to the data of one participant, and use the *a priori* ID according to the defined task. Notice that ID and MT have a direct relationship, while ID and hit rate are inversely correlated.

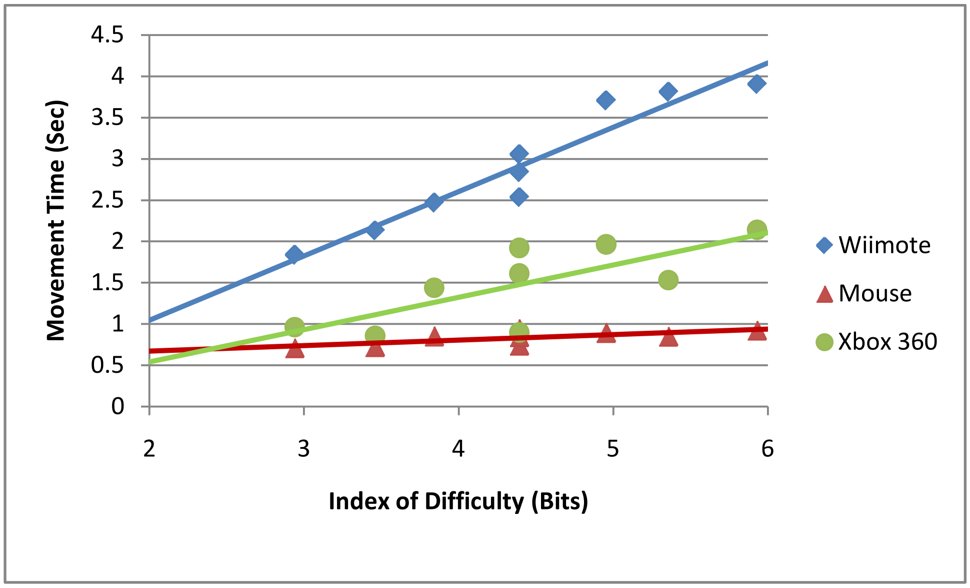

Figure 5-1: Movement Time by Interface

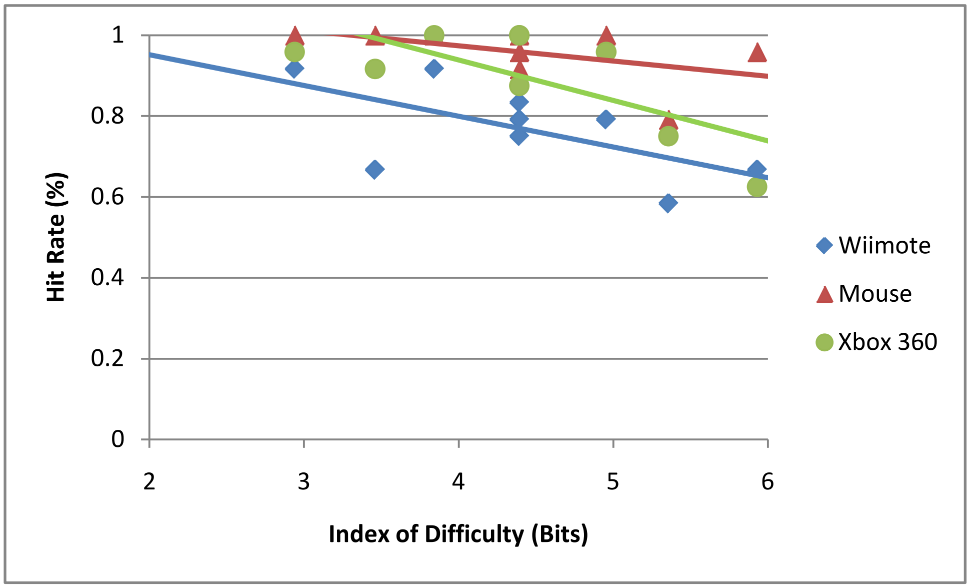

Figure 5-2: Hit Rate by Interface

These graphs show the mouse having the lowest average MT and highest hit rate, with the Xbox 360 controller outperforming the Wiimote.

The mouse also has the lowest effective indexes of difficulty. While ID_e using the effective distance and effective width was not a significant interaction across interfaces, dividing it by movement time (to create TP_e) yielded a significant interaction across interfaces. Throughput is designed to provide a metric that combines speed and accuracy. While the normal TP equation measures how well the participants performed the task they were given (moving the cursor from crosshair to crosshair), the effective throughput (TP_e) equations measure how well the participants performed the task they actually performed (moving the cursor from the actual point where they started to the point they selected). The mouse significantly outperformed the Wiimote and Xbox 360 controller for all three TP calculations. And while the Xbox 360 controller had a greater TP for the given task, the Wiimote had higher effective TP values. This means that the Xbox 360 controller is better in terms of speed and accuracy for the computer simulation and its given parameters, but the Wiimote actually outperformed the Xbox 360 controller in terms of where the participants actually selected with each interface. While the participants were told to move as quickly and accurately as possible, TP is independent of the speed-accuracy tradeoff, so these TP values should remain constant even if they were told to emphasize time or success rate.

All three measures of TP were significant across interface, distance, and width. However, it is particularly interesting that there is only a significant interaction between interface & distance and interface & width when the effective distance and widths are not used, respectively. Thus, the values for TP produce a significant interaction between interface & distance and interface & width, while TP_e using (A/W_e) only has a significant interaction between interface & distance, and TP_e using (A_e/W_e) does not have a significant interaction between any of the independent variables.

Distance and width also have a significant effect on MT, hit rate, and TP. This means that participants display different behavior models when they are moving different distances or to targets of different widths. Most likely, this is most likely due to cognitive planning mechanisms and the brain and limbs working together. This led to participants exploiting the target region to different levels. Participants had a large I_u for the mouse condition, which means that they favored the center of the target and did not utilize the entire target area. Since they were told to aim for the center of the target, this is not necessarily indicative of poor performance. Participants using the mouse consistently clicked close to the center of the crosshairs, averaging 8.3 pixels away from the center point. However, they utilized the entire target almost perfectly for the Xbox 360 controller, with an average I_u of .30, and overextended the target area while using the Wiimote, as indicated by the negative I_u (-.67) .

5.4.2 Path Performance Measures

In order to measure cursor path performance, the seven path accuracy metrics were tested. The researchers who proposed these metrics only found target re-entries and movement offset to be significant across interfaces (MacKenzie, Kauppinen, and Silfverberg); however, these two metrics were not significant across interfaces in this experiment. While movement offset was not significant across any variable, target re-entries had significant differences across distances and widths, and there were interactions between interface & distance and interface & width. Movement error was the only other metric that did not produce significant differences between interfaces. Task axis crossings, movement variability, movement direction changes, and orthogonal direction changes all had significant differences across interfaces, and all of these except for movement variability were also significant across distances. This is somewhat logical, because as distance increases the cursor path is more likely to change directions/cross the axis more times.

Across all metrics, it is clear that the Xbox 360 controller follows the most efficient path. Movement variability is the only significant metric in which the Xbox 360 controller does not produce the best results; the mouse has slightly less variability, but their totals are within one standard deviation of each other. However, the Wiimote fares considerably worse than either of the other two interfaces. The Xbox 360 controller outperforms the mouse and Wiimote with the rest of the path metrics: task axis crossings, movement direction changes, and orthogonal direction changes.

6. CONCLUSIONS

6.1 Discussion

6.1.1 Interface Performance

Based on the data collected in this study, it is quite apparent that the mouse is the most efficient interface in terms of both speed and accuracy. It was consistently the best in terms of MT, hit rate, ID, and TP. The Xbox 360 controller operated significantly worse than the mouse, but still did much better than the Wiimote in all of these metrics. Also, the slope of the lines in Figure 5-1 indicate that the mouse has the lowest interface speed constant (*b* in the Fitts' Law formula described by Equation 2-6), with Wiimote having the highest constant. This means that as ID gets higher, the MT of the Wiimote would continue to grow at a higher rate than the mouse, with the Xbox 360 controller growing at a rate in between the two.

The Xbox 360 controller fared significantly better than the other interfaces in terms of path metrics, with the mouse outperforming the Wiimote. This indicates that the Xbox 360 controller has a smoother movement target-selection trail, and that it may excel in terms of more precise tasks. The Wiimote metrics indicate that it is a poor choice of interface to use for accuracy tasks.

6.1.2 Metric Validity

This paper addressed multiple versions of throughput equations and addressed the validity of path performance metrics. The results indicate that all three equations describing throughput demonstrate significant differences across interfaces, distances, and widths. However, only when effective distance and width are used can the interactions between interface & distance and interface & width be disregarded, respectively. This indicates that the formula for TP_e described by Equation 2.10, and reproduced here as Equation 6.1, is the most appropriate in order to avoid an interaction. This formula also yields the lowest standard deviations within each interface, which is important because TP should remain constant within each interface regardless of task difficulty.

$$TP_e = \frac{log_2(\frac{A_e}{4.133*SD_x}+1)}{MT} \tag{6.1}$$

The index of utilization metric also provided an interesting piece of data, showing how users chose speed versus performance across interfaces by measuring how well they aimed.

It is inappropriate to either prove or disprove the legitimacy of path performance metrics since these results were so different from the original research. Contrary to what was expected, many of the metrics produced a significant difference across interfaces, and the measures that had previously been shown to be important actually did not demonstrate variance across interfaces. More research needs to be done in this realm in order to evaluate the legitimacy of these standards.

6.2 Implications

The results from this study indicate that the mouse is the fastest and most accurate interface, while the Xbox 360 controller excels in controlled path movement, and that users tend to over-utilize speed and the target area with the Wiimote. These implications shape how GUIs utilize each of these interfaces. Many applications designed for the mouse involve small buttons placed close together, such as pop-up dialog boxes, radio buttons, or drop-down menus. While the Xbox 360 controller is often used to move a cursor or aim in Xbox 360 video games, many Xbox 360 menu systems allow the user to scroll through menu options with the control stick. Since there is very little variation in straight-path performance with the thumbstick, users can reach their intended choice with ease.

In direct contrast to the mouse and computer GUIs, the Wii console uses large icons spaced far apart for Wiimote navigation, as pictured below in Figure 6-1.

Figure 6-1: Wii System Menu

This takes advantage of users' speed and allows them to aim for extremely large targets, thus reducing error rate.

6.3 Mitigations

There are also ways to increase the performance of interfaces. Balakrishnan found ways to "beat" Fitts' Law through virtual enhancements by decreasing target distance or increasing target width as users navigated towards it. The distance to targets can be decreased by laying targets in a linear or circular pie-menu, or dragging targets towards the cursor once the cursor begins navigating towards these targets. Empty space between targets also increases distance, and thus also MT, with no benefit to the user. Performance can also be enhanced by editing target width. By increasing the target size as the pointer gets close to it (similar to how the Mac OS X dock icons expand when the cursor gets close), target size increases, thus lowering the ID (Balakrishnan 2004). MT can be reduced by moderately increasing the control-display gain, or ratio of on-screen movement to physical movement, when the user is moving quickly (which indicates that the user is not yet close to the target).

Also, cues may be used in order to indicate that one has reached a target region. Auditory cues during a Fitts' task for when the cursor touched a virtual target significantly reduced MT for a PHANTOM stylus device similar to a Wiimote (Zahariev and Mackenzie 2008[1]). Similar results were also found using haptic force-feedback when a user was on a target by implementing a force-feedback mouse, which also reduced task discomfort and pain (Dennerlein and Yang 2000). The Wii implements both of these features in its menu system in order to facilitate user movement.

The Wiimote also has other control options that allow the user to navigate more precisely when required. A directional control pad on the Wiimote allows one to scroll between options in a menu system. There is also a separate optional attachment, pictured in Figure 6-2, called the Nunchuck that contains the same three-axis accelerometer

---

[1] This research was done by Christine L. Mackenzie, not the noted I. Scott MacKenzie.

(without the optical sensor) along with two additional buttons and an analog joystick. This joystick enables Wiimote users to have the same accuracy and path precision afforded by the Xbox 360 analog stick.

Figure 6-2: Wiimote with Nunchuck Attachment

6.4 Next Steps

6.4.1 Future Studies

There are many more studies which can explore other aspects of Fitts' Law and interface performance. Also, the Wiimote may perform better on tasks better suited to it, such as one that uses large targets or any of the width or feedback enhancers listed above. Although TP should remain constant, a task emphasizing speed over accuracy will aid

Wiimote performance. Path performance also needs to be studied in greater detail, as the results of this study contradict the previous research in the field.

6.4.2 Cross-Interface Simulation

One of the interesting aspects of this simulation is that since the XNA Framework supports all three interfaces and allows for networking, multiple users can interact within the same environment while operating different input devices. XNA allows networking through Internet Protocol (IP) or a Local Area Network (LAN). Either of these methods may be used for PC to PC networking and Xbox 360 to Xbox 360 networking. However, PC to Xbox 360 connectivity is only possible through LAN (via Ethernet cables). A Microsoft XNA Creators Club account is needed to transmit data between a Windows PC and the Xbox 360 console (or port games from the PC to the Xbox 360), and Xbox LIVE account is needed to communicate between Xbox 360 consoles over IP. As of the publication of this paper, there is no way to link PCs and XBox 360s over IP or to create a LAN game between the two with more than one of each system (XNA Team Blog). Possible combinations are featured below in Figure 6-3.

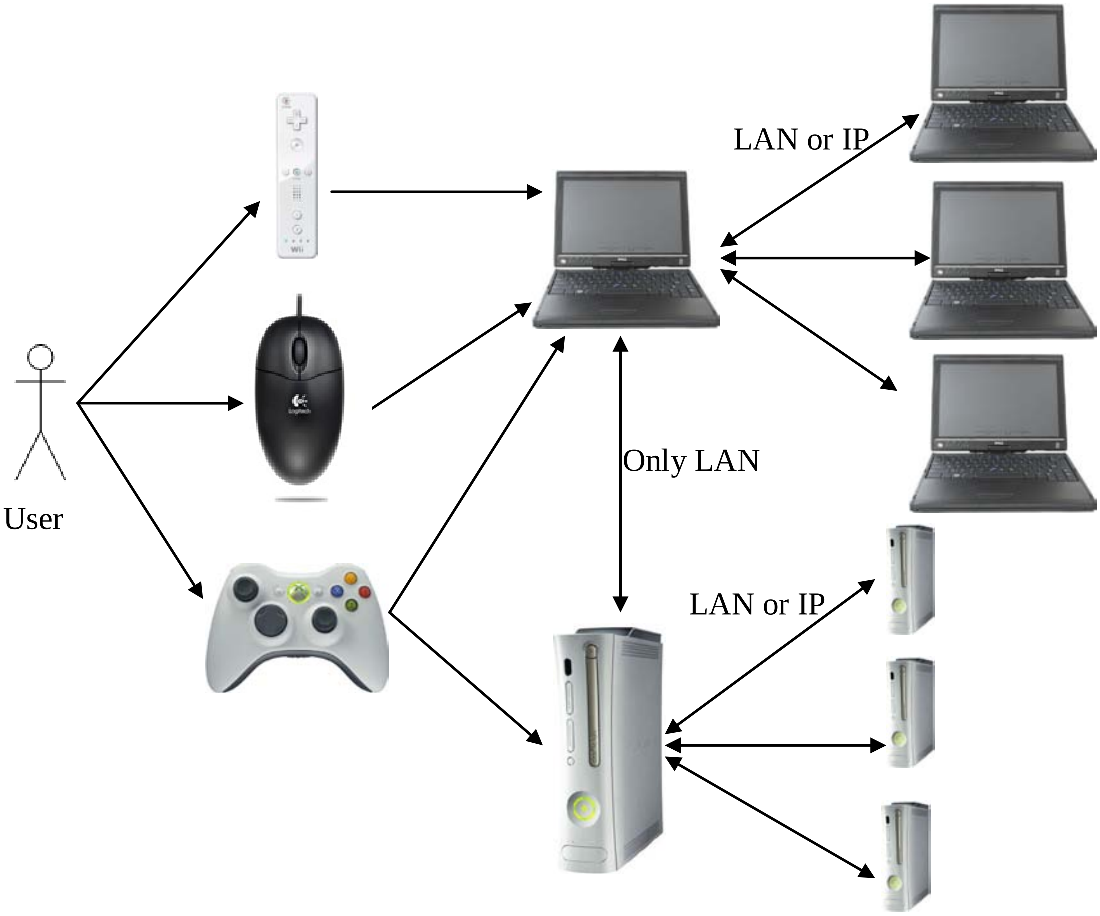

Figure 6-3: Possible Interface and Network Connectivity of an XNA simulation

References

Accot, Johnny and Shumin Zhai. "Scale effects in steering law tasks." Proceedings of ACM SIGCHI 2001: 1–8.

Balakrishnan, Ravin. "“Beating” Fitts' Law: virtual enhancements for pointing facilitation." International Journal of Human-Computer Studies 61.6 (2004): 857-874.

Balakrishnan, Ravin, and I. Scott MacKenzie. "Performance Differences in the Fingers, Wrist, and Forearm in Computer Input Control" Proceedings of ACM SIGCHI 1997: 303-310.

Berrigan, Félix, Martin Simoneau, Olivier Martin, and Normand Teasdale. "Coordination between posture and movement: interaction between postural and accuracy constraints." Experimental Brain Research 170.2 (2006): 255-264.

Coding4Fun: Managed Library for Nintendo's Wiimote. http://blogs.msdn.com/coding4fun/archive/2007/03/14/1879033.aspx. Accessed Dec. 2008.

Dennerlein, Jack T., and Maria-Helena J. DiMarino. "Forearm electromyographic changes with the use of a haptic force-feedback computer mouse." Human Factors 48.1 (2006): 130-141.

Fan, Jing, Jiping He, and Stephen Tillery. "Control of hand orientation and arm movement during reach and grasp." Experimental Brain Research 171.3 (2006): 283-296.

Fitts, Paul M. "The information capacity of the human motor system in controlling the amplitude of movement." Journal of Experimental Psychology 47 (1954): 381-391.

Göktürk, Mehmet, "The significance of selection method in computer pointing tasks." Doctoral Dissertation, The George Washington University, August 2000.

Grosjean, Marc, Maggie Shiffrar and Günther Knoblich. "Fitts's Law holds for action perception." Psychological Science 18.2 (2007): 95-99.

Guigon, Emmanuel, Pierre Baraduc, and Michel Desmurget. “Computational motor control: feedback and accuracy.” European Journal of Neuroscience 27 (2008): 1003–1016.

Isokoski, Poika, Roope Raisamo, Benoît Martin, and Grigori Evreinov. “User performance with trackball-mice.” Interacting with Computers 19 (2007): 407–427.

ISO 2002. Reference Number: ISO 9241-9:2000(E). Ergonomic requirements for office work with visual display terminals (VDTs)—Part 9—Requirements for non-keyboard input devices (ISO 9241-9) (2002): International Organisation for Standardisation.

Kelso, J. A. Scott. “Theoretical concepts and strategies for understanding perceptual-motor skill: from information capacity in closed systems to self-organization in open, nonequilibrium systems.” Journal of Experimental Psychology: General 121.3 (1992): 260-261.

Kelso, J.A. Scott, Dan L. Southard and David Goodman. “On the nature of human interlimb coordination.” Science 203 (1979): 1029-1031.

Kong, Jing, and Xiangshi Ren. “The optimal calculation method to determine the effective target width for the application of Fitts’ Law.” IEICE Transactions on Information and Systems 90.4 (1992): 753-758.

Lambert, Julie, and Chantal Bard. "Acquisition of visuomanual skills and improvement of information processing capacities in 6- to 10-year-old children performing a 2D pointing task." Neuroscience Letters 377.1 (2005): 1-6.

MacKenzie, I. Scott. "Fitts' Law as a research and design tool in human-computer interaction." Human-Computer Interaction 7.1 (1992): 91-139.

MacKenzie, I. Scott and Poika Isokoski. “Fitts' throughput and the speed-accuracy tradeoff.” Proceedings of ACM SIGCHI 2008: 1633-1636.

MacKenzie, I. Scott, Tatu Kaupinnen, and Miika Silfverberg. “Accuracy measures for evaluating computer pointing devices.” Proceedings of ACM SIGCHI 2001 3.1: 9-16.

MacKenzie. I. Scott, and William Buxton. “Extending Fitts’ Law to two-dimensional tasks.” Proceedings of ACM SIGCHI 1992: 219-226.

Marketwatch. Computer and Video Game Industry Tops $22 Billion in 2008. http://www.marketwatch.com/news/story/computer-video-game-industry-tops/story.aspx?guid={5129AAEA-E888-4E52-A4D7-348CF71D8C98}. Accessed Apr. 2009.

Murata, Atsuo. "Extending effective target width in Fitts' Law to a two-dimensional pointing task." International Journal of Human-Computer Interaction 11.2 (1999): 137.

New York Times. Next Wave of Media Threatens Video Game Makers. http://www.nytimes.com/2009/03/30/technology/30game.html?_r=1&hpw. Accessed Apr. 2009.

Olson, Gary M. & Judith S. Olson. Human-Computer Interaction: Psychological Aspects of the Human Use of Computing.” Annual Review of Psychology, 54 (2003): 491-516.

Radix, C.L., P. Robinson, and P. Nurse "Extension of Fitts' Law to modeling motion performance in man-machine interfaces." IEEE Transactions on Systems, Man & Cybernetics: Part A 29.2 (1999): 205-209.

Rinkenauer, Gerhard, Allen Osman, Rolf Ulrich, Hiltraut Müller-Gethmann, and Stefan Mattes. "On the locus of speed-accuracy trade-off in reaction time: inferences from the lateralized readiness potential." Journal of Experimental Psychology / General 133.2 (2004): 261-282.

Sohn, Misook and Geehyuk Lee. "SonarPen: An Ultrasonic Pointing Device for an Interactive TV." IEEE Transactions on Consumer Electronics 50.2 (2004): 413-419.

Soukoreff, R. William, and I. Scott MacKenzie. "Towards a standard for pointing device evaluation: perspectives on 27 years of Fitts' law research in HCI." International Journal of Human-Computer Studies 61.6 (2004): 751-789.

Stefels, C. N., J. Kneisser, R. G. Aarnink, P. Kaufholz, C. A. Grimbergen, and J. Dankleman. "Equipment control in a sterile environment using the Gyromouse and a new interface, the user interface (UI) wand." Minimally Invasive Therapy & Allied Technologies 16.3 (2007): 163-172.

van Veen, Vincent, Marie K. Krug, and Cameron S. Carter. "The neural and computational basis of controlled speed-accuracy tradeoff during task performance." Journal of Cognitive Neuroscience 20.11 (2008): 1952-1965.

Welford, Alan T. Fundamentals of Skill. London: Methuen, 1968.

XNA Team Blog: XNA Framework Networking and LIVE Requirements. http://blogs.msdn.com/xna/archive/2007/11/16/xna-framework-networking-and-live-requirements.aspx. Accessed Apr. 2009.

Young, Scott J., Jay Pratt , and Tom Chau. "Choosing the fastest movement: perceiving speed-accuracy tradeoffs." Experimental Brain Research 185 (2008):681–688.

Zaal, Frank T. J. M., and Esther Thelen. "The developmental roots of the speed-accuracy trade-off." Journal of Experimental Psychology / Human Perception & Performance 31.6 (2005): 1266-1273.

Zahariev, Mihaela A., and Christine L. Mackenzie. "Auditory contact cues improve performance when grasping augmented and virtual objects with a tool." Experimental Brain Research 186.4 (2008): 619-627.

Zhai, Shumin. "Characterizing computer input with Fitts' law parameters—the information and non-information aspects of pointing." International Journal of Human-Computer Studies 61.6 (2004): 791-809.

Zhai, Shumin, Jing Kong, and Xiangshi Ren. "Speed–accuracy tradeoff in Fitts' law tasks—on the equivalency of actual and nominal pointing precision." International Journal of Human-Computer Studies 61.6 (2004): 823-856.

Appendix A – Study Information Sheet

Study Information Sheet

Speed-Accuracy Comparison of Navigational Interfaces

GW IRB number: 110816

Principal Investigator: John Sibert	Telephone number: 202-994-4953
Sub-Investigator: Daniel Afergan	Telephone number: 508-930-6652

1) Introduction

You are invited to participate in a research study under the direction of Dr. Sibert of the Department of Computer Science, George Washington University (GWU). Taking part in this research is entirely voluntary. Your academic standing will not, in any way, be affected should you choose not to participate or if you decide to withdraw from the study at any time.

2) Why is this study being done?

The purpose of this study is to investigate the time needed to reach targets using different interfaces. The research will be conducted at the following location(s): George Washington University, Staughton Hall. A total of 12 participants will be asked to take part in this study.

3) What is involved in this study?

If you choose to take part in this study, you will be asked to navigate to targets on screen using 3 different interfaces: the Wiimote, a mouse, and an XBox 360 controller. You will be allowed to practice using each of the interfaces and practice navigating to targets before being tested.

The total amount of time you will spend in connection with this study is 1 to 1.5 hours.

4) What are the risks of participating in this study?

Participating in this study poses no risks that are not ordinarily encountered in daily life. You may refuse to answer any of the questions and you may take a break at any time during the study. You may stop your participation in this study at any time.

5) Are there benefits to taking part in this study?

You will not benefit directly from your participation in the study. The benefits to science and humankind that might result from this study are an increased understanding in human-computer interactions.

6) What are my options?

You do not have to participate in this study if you do not want to**.** Should you decide to participate and later change your mind, you can do so at anytime.

7) Will I receive payment for being in this study?

No.

8) Can I be taken off the study?

The investigator can decide to withdraw you from the study at any time. You could be taken off the study for reasons related solely to you (for example, not following study-related directions from the Investigator) or because the entire study is stopped.

9) How will my privacy be protected?

If results of this research study are reported in journals or at scientific meetings, the people who participated in this study will not be named or identified. GW will not release any information about your research involvement without your written permission, unless required by law.

10) Problems or Questions

The Office of Human Research of George Washington University, at telephone number (202) 994-2715, can provide further information about your rights as a research participant. If you think you have been harmed in this study, you report this to the Principal Investigator of this study.

*Please keep a copy of this document in case you want to read it again.

Appendix B – Experiment Directions

Experiment Directions

Thank you for agreeing to volunteer in our study. The purpose of this study is to see how well you can navigate on-screen to targets using three different interfaces. You will be using the Nintendo Wii Remote (Wiimote), a standard mouse, and an XBox 360 controller, not necessarily in that order. For each interface, you will first have a minute to move around on-screen and get familiar with using it. Your task will consist of you alternating pointing at bars on the left and right side of the screen. A red crosshair will always be in the center of your target bar. After that, you will do 1 practice trial and 9 trials, each consisting of 25 clicks. Your cursor will always begin in the middle of the left bar (with the right bar being the target) and timing will begin as soon as you see your targets. Please navigate **as quickly AND accurately** to the target as possible. Once the cursor is in the target region, press the controller button. A perfect click is exactly in the middle of the target bar and you should aim for this, but it is OK to hit anywhere in the bar. Please do this as quickly as possible once you are sure the cursor is in the target boundaries, and press the button only once. Once you press the button, immediately begin navigating to the other target. In the event of an error or misclick, please just continue.

If at any time you feel uncomfortable, sick, or wish to take a break or leave the study, please let the experimenter know. This study will take approximately 1 hour.

Appendix C – Participant Questionnaire

Response counts are in parentheses and Italics

1. What is your age?

___________ *(Mean = 26.3, SD = 3.67)*

2. What is your gender?

Male *(9)* Female *(3)*

3. What is your hand dominance?

Left-handed *(9)* Right-handed *(3)* Ambidextrous

4. How many hours a week do you use a computer (with a mouse)?

40+ *(10)* 30-40 *(2)* 20-30 10-20 0-10

5. How often have you used a Wii?

Never *(1)* Rarely *(7)* Occasionally *(4)* Frequently

6. How often have you used an XBox360 controller or joystick/thumbstick?

Never *(2)* Rarely *(7)* Occasionally *(3)* Frequently

7. How often do you play video games?

Never *(1)* Rarely *(5)* Occasionally *(5)* Frequently *(1)*

8. Do you have any cognitive or physical handicap that could handicap your input device use?

Yes No *(12)*