MAGICWAND:
A COMPARISON OF GESTURAL AFFORDANCES BETWEEN CYLINDRICAL AND FLAT DISPLAY FORM FACTORS

by

LAHIRU LAKMAL PRIYADARSHANA, MARAMBA KODIPPILIGE

A thesis submitted to the School of Computing
In conformity with the requirements for
the degree of Master of Science

Queen’s University
Kingston, Ontario, Canada
(February, 2017)

Copyright © Lahiru Lakmal Priyadarshana Maramba Kodippilige, 2017
Abstract

An affordance is the intrinsic ability of an object to allow an action. Affordances are generated by matching the fit of the body under action to the physical shape of the object. For example, a vertical door handle affords pulling, whereas a horizontal flat bar affords pushing to open. Modern smartphones with traditional flat screens offer poor affordances for gestural interactions. E.g., the flat form factor prevents a comfortable grip hindering wrist movements. Moreover, flat displays have a display area that can only be viewed from one side, and the visibility is reduced when the device is rotated.

In this thesis, we present MagicWand, a cylindrical display device consisting of two 5.5” Flexible Organic Light-Emitting Diode (FOLED) screens wrapped around a 3D printed body. MagicWand features a smartphone running the Android operating system. Gesture recognition allows the use of the wand movements as a form of input. We were interested in exploring how a cylindrical form factor will offer physical affordances for actions that are quite different from those of a traditional flat form factor of a smartphone.

To exhibit interactions with our prototype, an application scenario was developed where MagicWand was used as a game controller that can display a variety of 3D game elements. We believed that the physical affordance of this novel display device could affect the speed with which users learned specific gestures to operate the graphical elements on display. We conducted a user study to compare the guessability of gesture sets between flat and cylindrical device conditions. First, gestures were elicited for invoking common Graphical User Interface tasks in both device conditions, collecting over 100 unique gestures for 29 tasks. Second, a recognition engine was trained to recognize the top gestures for each task, for each device condition. Third, we measured the time it took for participants to discover these top gestures.

The results of the study suggest significantly faster discovery of input gestures in the cylindrical condition over the flat device with 20 percent of gestures, specifically those involving tilt.
Acknowledgements

The success of this work would not have been possible without the advice of my supervisor, Dr. Roel Vertegaal, whose feedback and guidance led me to explore new and creative ideas outside of the box. Furthermore, I would also like to thank all the extremely talented friends and colleagues of the Human Media Lab that I had the opportunity to work with. Thank you, Antonio Gomes, JP Carrascal, Sean Braley, Jesse Burstyn, Victoria Porter, Aaron Visser, Daniel Gotsch, Calvin Rubens, and Xujing Zhang. I am grateful to you for your support and for helping me make this work a success.

Furthermore, I would like to thank Dr. Alan Kay for taking his time to observe and provide feedback at a demonstration of MagicWand at the ACM Conference on Human Factors in Computing Systems (CHI) 2016 interactivity session.

I would like to thank my friends and family for their continuous support. A special note of appreciation is reserved for Kim McFarlane for her proof reading and editing of this work, ongoing encouragement, and being an exceptional friend. Finally, thank you to my parents for giving me the freedom to do what I love. I am grateful to my father for showing me the virtues of not giving up when the times are tough, and to my mother for motivating me to realise my dreams.
# Table of Contents

Abstract ................................................................................................................................. ii
Acknowledgements .............................................................................................................. iii
List of Figures ....................................................................................................................... vii
List of Tables ...................................................................................................................... ix
List of Abbreviations .......................................................................................................... x

Chapter 1 Introduction ......................................................................................................... 1
  1.1 Overview and Motivation ............................................................................................... 1
  1.2 Contributions ................................................................................................................ 5
  1.3 Outline of Thesis .......................................................................................................... 7
  1.4 Collaboration ................................................................................................................ 8

Chapter 2 Related Work ....................................................................................................... 9
  2.1 Wand-Like Input Devices ............................................................................................. 9
  2.2 Non-flat and Cylindrical Display Devices ................................................................... 12
  2.3 Mobile Gestural Interaction ......................................................................................... 15
  2.4 User-Defined Gestures .............................................................................................. 16
    2.4.1 Taxonomy of Motion Gestures .............................................................................. 16

Chapter 3 Design ................................................................................................................. 18
  3.1 Cylindrical Shape and Ergonomic Tool Fit ................................................................. 18
  3.2 Optimal Diameter and Length Tradeoff ..................................................................... 19
  3.3 Conformability of the FOLEDs .................................................................................. 20
  3.4 Conforming with Prior Gestural Game Controller Designs ....................................... 21
  3.5 Unconstrained Motion .............................................................................................. 21
  3.6 Optimal Design Space for Wrist Gestures ................................................................. 22
    3.6.1 Supporting Wrist Movements .............................................................................. 22

Chapter 4 Apparatus ........................................................................................................... 25
  4.1 Hardware ..................................................................................................................... 25
    4.1.1 Flexible OLED Displays .................................................................................... 26
    4.1.2 Assembling the Display ...................................................................................... 28
  4.2 Software ....................................................................................................................... 31
    4.2.1 Graphics ............................................................................................................. 31
    4.2.2 Recognizing Motion Gestures ............................................................................ 32

Chapter 5 Fantasy Game Application ................................................................................ 35
List of Figures

Figure 1. Hermione Granger is using her wand to cast the “Obliviate” charm to modify her parents' memories in *Harry Potter and the Deathly Hallows Part 1* (© Warner Home Video 2011, Fair use) [54].

Figure 2. A vertical bar on a door affords pulling; whereas a horizontal bar affords pushing (Creative Commons Image) [37].

Figure 3. The wand form factor ((a) PlayStation Move, (b) Wiimote) provides an ergonomic fit between the opposable thumb and fingers, as opposed to a (c) flat smartphone (Creative Commons and Public Domain Images) [34] [47] [13].

Figure 4. The MagicWand prototype with Google Maps.

Figure 5. A user performing pointing and steering tasks in Abaci et al.’s game environment using a magnetically tracked wand [2].

Figure 6. *DynaCan* (left) [3] and *Foldable Displays* by Lee et al. (right) [23] utilize projection mapping to render interfaces on non-flat displays.

Figure 7. *D20* [35], physical prototype (top-left) and virtual 3D model (bottom-left). *Digital Roll* [32], physical prototype (top-right) and virtual 3D model (bottom-right).

Figure 8. *Z-agon* [26] LED matrix prototype (left), pCubee [42] (center), and *Display Blocks* [33] (right).

Figure 9. Reflex [43] (left) and Holoflex [17] (right).

Figure 10. In a power grip, all the fingers are wrapped around the handle [24].

Figure 11. Three sets of distinct wrist movement (Creative Commons Image) [52].

Figure 12. MagicWand architecture. (1) 3D printed body (2) Android boards (3) Flexible OLED Displays (4) Cooling fan.

Figure 13. A FOLED screen demonstrated by LG Display (© LG Display, Fair use) [15].

Figure 14. LG display FOLED screen (left). Controller board running Android 4.4 (right).

Figure 15. The structure of a FOLED screen. Two layers of organic material are sandwiched between two conductors, and fabricated on a flexible substrate (Creative Commons Image) [15].

Figure 16. A FOLED damaged during the molding process.

Figure 17. Laminated FOLED screens with secured connectors before wrapping around the cylindrical body.

Figure 18. MagicWand system architecture.

Figure 19. Wekinator receives inputs, processes them using computational functions called models, and produce outputs.
Figure 20. A Dynamic Time Warping model interface on Wekinator. .................................................................34
Figure 21. Game characters transported to the magical island on the large display........................................36
Figure 22. Gestures: (a) Rotation, (b) Swirl, (c) Slash, (d) Tilt.................................................................37
Figure 23. A player interacting with MagicWand, (a) unlocking a dungeon door in Scenario 01, (b) destroying a spider web that obstructs the player’s path in Scenario 03. ...........................................40
Figure 24. Game characters and tools displayed on MagicWand........................................................................41
Figure 25. When the wand is rotated, the graphics circumnavigate according to IMU data giving the illusion of motion parallax......................................................................................................42
Figure 26. Flat displays have a display area that can only be viewed from one side, and the visibility is reduced when the device is rotated.........................................................................................43
Figure 27. Participant creating a gesture for “Menu” in the first experiment, with MagicWand (top) and a flat phone (bottom). ............................................................................................................46
Figure 28. A snippet from an IMU data file. First line shows the data type of each column..............................47
Figure 29. Graphical User Interface element corresponding to “Menu”. ..........................................................50
Figure 30. Agreement for each task sorted in descending order for the Cylindrical display device shown in blue. Gesture agreement for Flat display device is shown in orange..........................................................54
Figure 31. Top gestures used for discovery in the second experiment............................................................57
Figure 32. Navigating by moving the device relative to an external viewport (left) in contrast to moving the viewport inside the device (right). ....................................................................................63
List of Tables

Table 1. Comparison of affordances offered by controllers with cylindrical and flat displays..........44
Table 2. The list of tasks used in the participatory design study, grouped by category. ....................48
Table 3. The list of 20 tasks selected for the second experiment. ..................................................56
Table 4. Means, standard errors (s.e.) and t-values for actions with significantly faster gesture discovery time (seconds) on the Cylindrical display device (p < 0.05). .................................................................60
## List of Abbreviations

<table>
<thead>
<tr>
<th>Abbreviation</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ACM</td>
<td>Association for Computing Machinery</td>
</tr>
<tr>
<td>AIDL</td>
<td>Android Interface Definition Language</td>
</tr>
<tr>
<td>CHI</td>
<td>Conference for Human Factors in Computing Systems</td>
</tr>
<tr>
<td>DTW</td>
<td>Dynamic Time Warping</td>
</tr>
<tr>
<td>FOLED</td>
<td>Flexible Organic Light-Emitting Diode</td>
</tr>
<tr>
<td>GUI</td>
<td>Graphical User Interface</td>
</tr>
<tr>
<td>IMU</td>
<td>Inertial Measurement Unit</td>
</tr>
<tr>
<td>LCD</td>
<td>Liquid Crystal Display</td>
</tr>
<tr>
<td>LED</td>
<td>Light-Emitting Diode</td>
</tr>
<tr>
<td>OLED</td>
<td>Organic Light-Emitting Diode</td>
</tr>
<tr>
<td>OSC</td>
<td>Open Sound Control</td>
</tr>
<tr>
<td>OUI</td>
<td>Organic User Interface</td>
</tr>
<tr>
<td>VGA</td>
<td>Video Graphics Array</td>
</tr>
</tbody>
</table>
Chapter 1

Introduction

1.1 Overview and Motivation

The implementation of inertial sensors in mobile devices has paved the way for the use of motion-based gestural input in everyday mobile interactions. However, modern smartphones and tablets were primarily designed for touch input, and their flat, brick-like form factor make them less than suitable for gesture-based input. By contrast, game controllers, such as Nintendo’s Wiimote or Sony’s Playstation Move, offer cylindrical wand form factors that are primarily designed for gestural input.

Their design is based on a long history of using wands and staffs as a means of gestural expression. Staffs were a symbol of power among those in the Old Testament. For example, Moses, using his staff, parted the Red Sea in order for the Israelites to cross the water on dry ground ([11] Exodus 14:16). Monarchs traditionally use the sceptre as a means of regal and imperial power, and continue to do so.

Wands have been used in rituals and are described in fantasy literature and motion pictures as tools for casting spells [44]. The motion picture series based on the *Harry Potter* fantasy novels depict the use of wands by wizards to channel their magic (Figure 1). It is a crucial element for being a wizard and it is the weapon of choice when casting spells.
A baton, a specific type of wand, is used by conductors to enhance their expressiveness when directing musical ensembles. Conductors use musical batons to communicate the nuanced and phrased musicality to a group of musicians through gestures [49].

While prior work has explored the use of the wand form factor for motion-based gestural control in Human-Computer Interaction [22, 25], there has been little to no work on devices that offer both the high resolution display of a smartphone and the gestural affordances of a wand in one cylindrical form factor [9, 48].

The popularity of the wand form factor for gestural expression is due to its physical affordances. According to Norman [14], the affordances of an object are those properties that directly suggest how a user might interact with that object. Affordances are generated by the brain by matching the fit of the body under action to the physical shape of an object [14]. For example, a vertical door handle affords pulling because its physical shape affords grasping the handle with
fingers to pull towards. By contrast, a horizontal flat bar affords pushing by placing the hands on the bar as the fingers do not fit to grasp (Figure 2).

The wand form factor affords physical holding by the hand, as it provides an excellent ergonomic fit between the opposable thumb and fingers (Figure 3). Its connection to the wrist allows the object to be freely rotated and treated as an extension of the arm. By contrast, flat multitouch displays afford one handed holding for finger-based input by the dominant hand. Flat computer screens can offer visual affordances, e.g., through 3D graphics, but their physical shape does not naturally correspond to that of the visualization. According to Gaver [16], such conflict can create a false affordance that can potentially lead to error: the inability of a user to determine the correct action.

Figure 2. A vertical bar on a door affords pulling; whereas a horizontal bar affords pushing (Creative Commons Image) [37].
The concept of physical affordance is key to Organic User Interfaces (OUIs) [20]: designing interactions for devices that do not have flat or even rigid displays. OUIs offer unique physical affordances in addition to their visual display, a property that has been exploited in various novel interaction techniques [3, 35].

The recent availability of high-resolution DisplayObjects [3] with Flexible Organic LED (FOLED) displays now allows researchers to study how the affordances of non-flat display form factors can influence the way users interact with their devices in the future.

Figure 3. The wand form factor ((a) PlayStation Move, (b) Wiimote) provides an ergonomic fit between the opposable thumb and fingers, as opposed to a (c) flat smartphone (Creative Commons and Public Domain Images) [34] [47] [13].
1.2 Contributions

In this thesis, we contribute MagicWand, a wand-like DisplayObject with two 5.5” Flexible Organic Light-Emitting Diode (FOLED) displays wrapped around a cylinder (Figure 4). MagicWand provides motion-based gestural interactions with a high-resolution full colour Graphical User Interface (GUI) in a cylindrical form factor.

MagicWand uses gesture recognition through its Inertial Measurement Unit (IMU) as its primary input method. We contribute an application, a fantasy adventure game that takes advantage of MagicWand’s features. In this game, MagicWand works as a game controller that displays different tools depending on the game context. Users discover gestures that operate the tools through visual and physical affordances.

To evaluate the affordances of the cylindrical display form factor, we conducted a user-centered elicitation study [51] that compared gestural affordances of a flat smartphone with MagicWand (both without touch) to discover what gestures are naturally chosen to execute a number of common GUI operations. Over 100 unique gestures were collected for 29 tasks, after which a recognition engine was trained to recognize the most frequently performed gestures per task.

We contribute a second study in which we measured the time it took for participants to discover this top gesture set, per task, per device, analyzing 20 tasks in which gesture sets were identical between devices. Results suggest significantly faster discovery of top gestures in the MagicWand condition over the flat smartphone condition in 20 percent of cases, specifically those involving tilt.
Figure 4. The MagicWand prototype with Google Maps.
1.3 Outline of Thesis

This thesis is presented in nine chapters. The first chapter introduces the topic of motion gestural interactions with handheld mobile devices and discusses the motivation for creating a handheld cylindrical display device. Furthermore, this chapter also introduces the concept of physical affordances and how Organic User Interfaces (OUIs) could offer better user experiences.

The second chapter presents an overview of the related work that has been done in the areas of wand-like input devices, non-flat and cylindrical display devices, mobile gestural interaction, and user-defined gesture studies.

Chapter three provides details about our design rationale for making MagicWand. The fourth chapter describes the hardware and software components of the MagicWand apparatus.

The fifth chapter presents a fantasy game application that highlights some of MagicWand’s capabilities. This section of the work provides details about the game design, implementation, motion gestures, and interactions with MagicWand as a game controller.

Chapter six presents the first experiment, a gesture elicitation study where users created a gesture set for 29 tasks in both cylindrical and flat device conditions. This section includes descriptions of the experimental design, the participants, measurements, procedure and task. A discussion of the results is also presented.

Chapter seven focuses on the second experiment, where the gesture discoverability time was measured for the most frequent gestures for both cylindrical and flat device conditions. This includes descriptions of the experimental design, the participants, measurements, hypothesis, procedure and task. A discussion of the results is also presented.
The eighth and ninth chapters provide a detailed discussion of findings from the two experiments including user feedback. This section summarizes the main findings of the thesis and describes the limitations and future work in this area.

1.4 Collaboration

The fantasy game application presented in Chapter 5 was designed and implemented collaboratively with Victoria Porter. Technical solutions to secure the displays, as described in Chapter 4, were developed collaboratively with Aaron Visser. The author developed the hardware prototype and implemented the software that runs on MagicWand for the game application demonstrated in Chapter 5, and for the experiments described in Chapter 6 and Chapter 7. The experiments presented in Chapter 6 and Chapter 7 were designed collaboratively with JP Carrascal.
Chapter 2

Related Work

Both gestural interaction with mobile devices and the prototyping of wand-shaped input devices have received considerable attention in the past. MagicWand builds upon several areas and techniques of previous research: (1) wand-like input devices; (2) non-flat and cylindrical display devices; (3) mobile gestural interaction; and (4) user-defined gestures.

2.1 Wand-Like Input Devices

The physical form factor of a wand or a stick naturally affords grasping, pointing, and making motion gestures. Sticks have been the simplest and earliest tools used by humans. Stick-like tool use can be traced back to ancient civilizations to hunt, to conduct music, and for use as sports equipment or weaponry, in the form of spears, batons, hockey sticks, or swords. Previous work includes explorations of the wand form factor to interact with computing systems. In this section, we will discuss related prior work involving passive (without any embedded electronics), active (with embedded electronics), haptic-enabled, and commercially available wand-like devices.

Cao et al.’s VisionWand [9] was a passive wand tracked in 3D space for interacting with large displays. Made from a simple plastic rod with two coloured ends, it did not contain any embedded electronics. Both endpoints of the wand were tracked using two cameras in 3D space resulting in a 3D ray projection. A black cross was displayed by the system denoting the intersection of the 3D ray and the screen, indicating the screen position that the wand was pointed at.
VisionWand’s system supported basic interactions by changing the position of the wand in relation to the display (e.g., altering the distance of the wand from the display to scale objects). However, complicated tasks were performed using visual widgets, which were activated with simple gestures such as rotate and tilt. The creators of VisionWand observed that the gestures and postures of wand-based interaction were easily understood and performed by users.

XWand [48], an active wand augmented with inertial sensors, used pointing gestures for controlling devices in interactive environments. The design of XWand was partially motivated by the natural tendency of humans to look at, point at, and talk about objects that the user wants to control [7]. With XWand, users could point at the device they wish to control, then use simple gestures or voice commands to issue commands. The device also contained a button to directly communicate state information with a computer (e.g., to turn on or off a light).

Abaci et al. [2] presented a game-like virtual reality application with a magnetically tracked wand to perform pointing and steering tasks (Figure 5). The system was limited to simple gestures and utilized voice commands for complex interactions, which they denoted as “spell-
casting”. The authors suggested that adopting an intuitive device such as a wand reduces the learning curve as compared to a joystick or a 3D mouse.

In music performances, a conductor leads and guides an orchestra to make sure that the music piece is interpreted properly by the various members of an ensemble. In order to achieve this, conductors use a wand-like tool, known as a baton. Modern batons are generally made of wood or fiberglass and do not contain any electronics. The Radio-Baton [25] by Mathews was an example for an early expressive gesture-based control for music performance through radio tracking. The Radio-Baton consisted of two batons with radio transmitters at the end of each stick, and a receiving antenna. The system tracked the motion of the two batons in three-dimensional space by comparing the radio signal strength. This was an early example of utilizing motion gestural expressions with wand-like devices in electronic music performances.

Researchers have also explored the effect of haptic feedback in wand-like input devices to enhance the user experience. In Hapticast, Andrews et al. [4] used a Phantom Omni [31] to provide force-feedback to a virtual wand in a game environment. Users interacted with the game world using four different wands, by casting spells. Based on distinctive properties of each wand a Phantom Omni was used to provide haptic feedback to the user. Nakagaki et al. introduced Linked-Stick [29], a shape shifting stick-like device that mirrors the shape of another device. The system consisted of two devices: a master and a slave. Both devices had a wand-like form factor and contained an accelerometer to detect the motion. The slave device consisted of 2 servomotors in addition to the accelerometer. The authors presented a performer-audience configuration where the master device transmitted its movements to the slave device, mimicking its motion to provide haptic feedback to the user. The authors suggested that a system like Linked-Stick could convey
the gestural expressions of a performer to an audience in a way that is otherwise difficult to convey using a purely digital medium.

Commercial game controllers with a wand-like shape, such as the Nintendo’s Wiimote and the Sony PlayStation Move, also used motion sensing for gestural input. This type of game controllers were sometimes credited with introducing the use of physical exertion in video games. These game controllers, however, did not feature high resolution displays. One notable exception was the Wii U’s GamePad. Although it had embedded inertial gesture sensors, it mostly relied on a joystick and buttons for input and lacks the wand-like form factor.

2.2 Non-flat and Cylindrical Display Devices

The unavailability of actual deformable displays had been a major technical hurdle in past research, forcing researchers to rely on projection mapping [3, 23] and virtual 3D models [3, 32, 35]. Akaoka et al.’s DisplayObjects [3] was an example of using projection mapping to render interfaces on the surface of non-flat objects, including the cylindrical DynaCan (Figure 6).

Figure 6. DynaCan (left) [3] and Foldable Displays by Lee et al. (right) [23] utilize projection mapping to render interfaces on non-flat displays.
They observed that having the interactive graphics match the physical shape of the object allowed better use of the affordances of the physical shape. Similarly, Lee et al. [23] presented a series of foldable display designs using projection mapping. Unlike traditional flat displays, these foldable display designs allowed the user to quickly increase or decrease the size of the display surface.

Researchers have also employed virtual 3D models to explore interactions with non-flat display devices. For instance, Poupyrev et al. [35] presented D20, an icosahedral display device rendered as a 3D object and controlled by an external non-display device (Figure 7, left). The shape of D20 afforded rotation of the device to explore the display surface. Pillias et al. [32] used a similar approach when exploring a hand-held cylindrical form factor called Digital Roll (Figure 7, right). They studied how such a form factor could be used as a reading device. Their findings suggested that a rotating motion is a natural gesture prompt by the cylindrical form factor to scroll through text. Both D20 and Digital Roll were used to investigate the impact of the shape of a display on user experience and performance, however, the prototypes were developed using virtual 3D simulations.

Figure 7. D20 [35], physical prototype (top-left) and virtual 3D model (bottom-left). Digital Roll [32], physical prototype (top-right) and virtual 3D model (bottom-right).
As an alternative to projection mapping and virtual 3D rendering, researchers developed multi-faceted, non-flat display devices by stitching together multiple flat displays, however, only to create simple cubical shapes. This approach was explored in the work of Matsumoto et al. [26], Stavness et al. [42] and Pla et al. [33].

Z-agon by Matsumoto et al. [26] was one of the earliest examples for cubic displays. They explored application scenarios to demonstrate interactions with Z-agon using physical mock-ups. Their application scenario, Z-game, presented a conceptual game screen with a character that runs around the cube. Z-map was another application concept to use Z-agon as a portable map navigation system. Matsumoto et al. also presented a prototype built with four 16x16 LED matrix modules (Figure 8, left). Stavness et al. designed pCube [42], a multi-screen display device with VGA resolution LCD screens (Figure 8, center). The system included a headtracker to simulate motion parallax. Both Z-agon and pCube prototypes were handheld, but not self-contained. Display Blocks by Pla et al. [33] was another example of a handheld multi-faceted display device (Figure 8, right). Display Blocks were self-contained and each face contained an independently controlled Organic Light-Emitting Diode (OLED) screen.

Figure 8. Z-agon [26] LED matrix prototype (left), pCube [42] (center), and Display Blocks [33] (right).
Each prototype demonstrated that the cubic shape affords turning of the device by the fingers to explore the content displayed on different faces.

The above non-flat multi-faceted display prototypes featured a large bezel (Figure 8) that interrupts the continuity of the display. Recent advances on FOLED displays have made possible the development of truly conformable display devices [17, 43]. Reflex [43] and HoloFlex [17] are deformable mobile devices featuring FOLED displays (Figure 9). Users interacted with these mobile devices using bends as input, deforming the display itself. This technology allowed displays to be conformed to a 3D shape, making it possible to create bezel-less non-flat display devices.

2.3 Mobile Gestural Interaction

MagicWand builds upon previous work on gestural interactions with handheld mobile devices. Rekimoto [38] presented one of the earliest studies on gestural input for mobile devices, using a combination of button clicks and tilt sensing for one-handed interaction. He demonstrated how motion input can be used to perform interactions with menu items, scroll bars, map browsing and 3D object viewing. As mobile devices started to become ubiquitous, researchers subsequently
explored the use of tilt as input [6, 18, 41], and display orientation change [19] which eventually became a universal feature in handheld mobile devices.

2.4 User-Defined Gestures

Gestural interaction is more prevalent today, since most modern smartphones have inertial sensors. However, the unconstrained nature of gestures can make it difficult to develop gesture sets that minimize the user’s cognitive load when using a system. Wobbrock et al. [50] proposed a methodology for improving guessability of symbolic input that has been successfully applied in gesture elicitation studies. In this type of study, participants were asked to perform gestures with the intention to trigger a set of actions. By finding the gestures with the highest agreement in the resulting set, a common vocabulary can be constructed.

Ruiz et al. [39] used this methodology to propose a taxonomy of motion gestures for smartphone applications. They mapped gestures to both actions and navigation-based tasks. A similar methodology has been used for finding natural gesture sets for interactions with tabletops [51], drones [12] and TVs [45].

Although this is the subject of continued debate, findings from Morris et al.’s [27] comparison of user-defined gestures and HCI researcher-authored gestures suggested that participants prefer gestures created by user elicitation methodologies. They found that user-defined gestures were physically and conceptually simple, while designers tend to construct more complex movements.

2.4.1 Taxonomy of Motion Gestures

Based on the results of previous elicitation studies, researchers had classified gestures along apparent parameters utilized when performing motion gestures. Ruiz et al. [39], in their
work on user-defined motion gestures for mobile interaction, presented a taxonomy for motion gestures that contained two different classes of taxonomy dimensions: *Gesture Mapping* and *Physical Characteristics*.

*Gesture Mapping* involved how users map gestures to actions by considering their *Nature, Context* and *Temporal* dimensions. The *Nature* dimension defined the mapping of the gestures to physical objects, which the researchers further categorized as *Metaphor, Physical, Symbolic*, and *Abstract* mappings. Wobbrock et al. [51] constructed a similar categorization for surface gestures. A metaphorical gesture mapping was a *Metaphor* of acting on another physical object (e.g., picking up the device to answer a call as per the way of picking up an old-fashioned telephone handset). A *Physical* gesture mapping acted physically on the device itself (e.g., direct manipulation). A *Symbolic* gesture visually depicted a symbol (e.g., drawing the letter B with the device), while an *Abstract* gesture mapping was arbitrary. The second dimension in *Gesture Mapping* was the *Context* dimension, which defined whether the gesture is performed in a specific context. The third dimension, the *Temporal* dimension, described if the action occurred during or after completion of gesture.

*Physical Characteristics* were the second class of taxonomy dimensions. These involved the characteristics of a gesture such as *Kinematic Impulse* properties, *Dimension*, and *Complexity*. *Kinematic Impulse* described the rate of change of acceleration of the movements. The *Dimension* of a gesture described the number of axes involved in the movement, and the *Complexity* described whether a gesture consists of a simple single gesture or a compound gesture that can be decomposed into multiple single gestures.
Chapter 3

Design

Based on our literature review, we envisioned MagicWand as a cylindrical, handheld display device with gestural input that acts as a virtual container for 3D objects displayed on its surface. We ideated and developed a fantasy game application to demonstrate the potential of the device by using physical affordances mapped to motion gestures to interact with game objects. In the design process of MagicWand we considered the following design parameters.

3.1 Cylindrical Shape and Ergonomic Tool Fit

During the Stone Age (10,000 B.C. – 4,000 B.C.), humans used palm-sized stones with a semi-cylindrical shape as tools. Wooden cylindrical handles were introduced later to add an ergonomic grip when handling tools with larger action radii that are not of a cylindrical shape, such as a cutting knife or screwdriver [8]. MagicWand was based on such classic tool designs, given its primary use as a handheld tool for gestural interaction. We chose a cylindrical shape for these reasons: 1) A cylindrical shape naturally affords rotation and tilt motion by fitting to the opposable thumb and fingers of the hand such that it becomes an extension of the wrist. 2) A cylindrical shape provides a large area of contact, with no edges that can produce areas of high pressure on the hand that can prevent a comfortable grip. By contrast, shapes with edges will introduce uneven local pressure points inhibiting the user experience [30]. 3) Additionally, a cylindrical display provides a continuous display area when the device is rotated, whereas with a flat display the screen will be hidden when rotated away from the user.
3.2 Optimal Diameter and Length Tradeoff

The size of the device was to be sufficiently comfortable to hold with one hand, while at the same time fitting the available FOLED displays. In the design of MagicWand, we drew ideas from modern hand tools to determine the optimal diameter and length. Our opposable thumb allows the fingers to grasp and handle objects with precision (e.g., when turning a key) and power (e.g., when using a hammer) [40], but only provided the tool fits the hand. A power grip, as shown in Figure 10, provides maximum hand force to the tool, with all the fingers wrapped around the handle while the thumb applies pressure. In an ergonomic modern hand tool (e.g., screwdrivers, hammers), the preferred diameter of the handle is 30 mm to 50 mm to provide a comfortable grip [24].

Figure 10. In a power grip, all the fingers are wrapped around the handle [24].
When gripping the handle, it should allow the operator to maintain a natural wrist posture without the need of wrist flexion, extension, or ulnar or radial deviation. Additionally, the handle should extend across the entire breadth of the palm. If the handle is too short, this can cause unnecessary compression in the middle of the palm. The preferred handle length is 140 mm, with a recommended minimum of 102 mm [24].

When deciding on the size of MagicWand, we had to consider the technical constraints such as the controller board size and the size of available screens. The cylinder was to be large enough to fit the controller boards, and its diameter was to allow the available screens to wrap around to provide as close to a 360 degree display surface as possible. A final constraint was to not stress the screens beyond their maximum curvature. The board and display size produced a 56 mm diameter cylinder of 175 mm height when two displays were applied side by side. The board size required a diameter slightly too large to cover the entire cylinder with a seamless display: a strip of approximately 1 cm was used as a bezel. The final design was, as such, only slightly larger than the preferred ergonomic measurements of a handheld device.

### 3.3 Conformability of the FOLEDs

MagicWand consists of two FOLED displays wrapped around its lateral surface. Even though these screens are flexible enough to wrap around an object, it was imperative to choose a shape with no edges, making an even surface with no local pressure points. The cylindrical shape, a most prominent basic shape, exerts equal pressure on a conformed FOLED, around its surface that prevents separation of the layers of the FOLED due to uneven radii or pressure points on circuitry.
3.4 Conforming with Prior Gestural Game Controller Designs

In the design process, ideas were also drawn from motion-based game controllers such as the Wiimote and PlayStation Move. The body of the Wiimote measures 160 mm long, 36.2 mm wide, and 30.8 mm thick. Its shape resembles a standard TV remote with curved edges. PlayStation Move, on the other hand, resembles a cylindrical shape and measures 200 mm long and 46 mm in diameter. These controllers are designed to provide a comfortable grip and support natural hand and wrist movements for motion gestural interactions. Both Wiimote and PlayStation Move use Inertial Measurement Unit (IMU), a combination of accelerometers and gyroscopes, to provide gesture-based input, and this became a requirement for MagicWand as well.

3.5 Unconstrained Motion

It was crucial to create a self-contained and autonomous device to limit any impediments to gestural interaction with the device. As motion-gestural input was to be the primary method of interacting with MagicWand, it should be untethered to allow unhindered wrist and arm movements. Moreover, the device should have necessary computing power to detect motion gestures and to render 3D and 2D graphics standalone.

MagicWand combines the user experience of a game controller (e.g., Wiimote, PlayStation Move) with that of a smartphone. We used boards with wireless communication and integrated motion gesture recognition capabilities. The device was designed to be battery powered for approximately 45 minutes, providing uninterrupted user experience with a single charge. Finally, our design fits all the hardware inside the cylinder having most of its lateral
surface area covered by a display. To do so required the additional of a cooling circuit with fans on the extremities of the cylinder.

3.6 Optimal Design Space for Wrist Gestures

In the absence of a touch screen, MagicWand was designed to rely primarily on gestural input. The following are the design parameters we considered to maximize the available design space for motion gestures with the wrist.

3.6.1 Supporting Wrist Movements

We envisioned MagicWand as an extension of the user’s wrist, supporting the maximum number of degrees of freedom provided by its joints. The wrist is capable of three sets of distinct movements: flexion and extension, radial deviation and ulnar deviation, pronation and supination.

Flexion describes the movement of bending the palm inwards and extension is the movement of bending the palm outwards, raising the back of the hand. Radial deviation is when the wrist is bent towards the thumb or the radial bone while ulnar deviation is the movement of bending the wrist towards the little finger or the ulnar bone. Pronation and supination describes rotational movements of the wrist where the former is rotating the forearm into a palm down position and the latter is rotating the forearm with the palm facing up (Figure 11).

A neutral position is defined as the wrist in straight alignment with the forearm (no flexion, extension, radial or ulnar deviation) and at the mid-point between supination and pronation. Flexion, extension, and radial or ulnar deviation are used in gestures with tilt motions whereas pronation and supination movements are used in rotational gestures.
We based our initial set of gestures for MagicWand on the above capabilities of the wrist: providing gestures for rotation (pronation/supination), tilt (flexion/extension), as well as pointing (deviation). Motion gesture design was also inspired by everyday object handling. For example, actions such as turning a key to unlock a door, or turning a door knob to open a door, involve rotation through pronation/supination of the wrist. When handling cylindrical fluid containers such as bottles, the object is naturally tilted through flexion/extension of the wrist to drink from or pour liquid out. Pointing with a cylindrical object such as a remote control, laser pointer, or stick involves radial and ulnar deviations of the wrist. Deviations of the wrist becomes a motion gesture when cutting or slashing using tools such as knives and swords. We were similarly inspired by the swirl motion used when stirring food with a kitchen utensil such as a spoon, which involves a combination of deviation and flexion/extension actions by the wrist. Based on the
above, our initial gesture set involved *Rotation, Tilt, Point/Slash* and *Swirl* gestures, as illustrated and used in a gaming scenario in Chapter 5.
Chapter 4

Apparatus

4.1 Hardware

MagicWand consists of two 5.5” LG Display FOLED screens wrapped around a 3D printed cylindrical body. Each screen covers half of the lateral surface, creating a 155 mm x 135 mm display area. The total resolution of the display is 1440 by 1280 pixels. The cylindrical body has a radius of 28 mm and a height of 175 mm. Two cooling fans reside at the extremities of this cylindrical body to prevent MagicWand from overheating (Figure 12).

![Figure 12. MagicWand architecture. (1) 3D printed body (2) Android boards (3) Flexible OLED Displays (4) Cooling fan.](image)
The two displays are controlled independently using two Android boards running the Android 4.4 operating system. Each board is powered by a 3.7V 750mAh lithium polymer battery. Since touch input is not provided, MagicWand primarily relies on gesture control, which is enabled by the IMUs on the controller boards. The device also contains a microphone and a speaker for audio input and output.

4.1.1 Flexible OLED Displays

Organic Light-Emitting Diode (OLED) is an emerging display technology that enables thinner, lighter, and more efficient display panels. They are already used in many mobile devices and usually fabricated on a glass suitable for rigid screens. A Flexible OLED (Figure 13) is a type of Organic Light-Emitting Diode (OLED) fabricated on a flexible substrate, typically plastic [15].

Figure 13. A FOLED screen demonstrated by LG Display (© LG Display, Fair use) [15].
The flexible property of these screens allows their use in the production of rollable and deformable displays. MagicWand uses two rollable 5.5” LG Display FOLED screens that are controlled using controller boards running the Android 4.4 operating system (Figure 14).

Figure 14. LG display FOLED screen (left). Controller board running Android 4.4 (right).
4.1.2 Assembling the Display

A Flexible OLED consists of five layers (Figure 15). Two layers of organic material are sandwiched between two conductors, and fabricated on a flexible substrate. Even though the FOLEDs are flexible, sheer forces introduced when molding the screens to a shape can shift the layers, rendering the screens unusable. Additionally, the stress introduced on the connectors when bending the screens can damage communications with the display unit. The process of building the MagicWand prototype required approximately 15 screens. Figure 16 shows a screen that was damaged during the molding process.

Figure 15. The structure of a FOLED screen. Two layers of organic material are sandwiched between two conductors, and fabricated on a flexible substrate (Creative Commons Image) [15].
To overcome these problems, we initially applied heat on the screens before wrapping them around the cylinder, making the layers more flexible. We experimented using a preheating station hot plate and a rework solder station heat gun. Applying heat indeed made the screens more flexible; however, the layers continued to separate. Wrapping the screens without heat produced better results if a clamp was applied to prevent the connectors from separating. The clamp holds the connectors in place preventing them from bending with the screens, preventing separation. Both screens were laminated together with a cold lamination sheet before wrapping them around the cylinder. The connectors were then secured with tape for added protection.

Figure 16. A FOLED damaged during the molding process.
(Figure 17). Another challenge we faced when building the prototype was to hold the screens securely in place on the cylindrical body to prevent them from unwrapping and to provide an even force along the surface area of the screens. Gluing the displays introduced a force differential between inner and outer layers. This was addressed by encapsulating the screens inside a clear acrylic tube.

Figure 17. Laminated FOLED screens with secured connectors before wrapping around the cylindrical body.
4.2 Software

4.2.1 Graphics

We developed a Unity application that runs independently on the two Android boards to render 2D and 3D graphics on the wand displays. The two Android boards are wirelessly synchronized via Open Sound Control (OSC) messages to update graphics in real time, making the two screens perform as a single display. Figure 18 illustrates the high-level architecture of the system. The applications running on the device also communicate with a computer wirelessly via OSC messages.

Figure 18. MagicWand system architecture.
4.2.2 Recognizing Motion Gestures

MagicWand uses gestural input as its primary interaction method. During the implementation of software, we tested three different implementations of gesture detection modules. First, we developed a simple gesture recognition algorithm that observed the changes of acceleration of movements. These observations included the number of peaks within a time interval or detecting a sinusoidal pattern in acceleration. This simple gesture detection module could only detect a limited number of distinct motions. It was also not possible to train this algorithm for new gestures with this implementation.

In the fantasy game application presented in Chapter 5, we used the Android Gesture Recognition Tool [5] for gesture detection. It provided a service to recognize newly performed gestures and sent the recognition result to Android applications (activities) that were subscribed to its Android Interface Definition Language (AIDL) service interface. The same service interface could be used to train new gesture sets. This method had a higher rate of accuracy compared to our first implementation. However, when the gesture set became too large this method had difficulties in differentiating distinct gestures.

In our user studies, Wekinator [46], a real-time interactive machine learning engine, was used to train and recognize IMU data into interactive gestures. Wekinator receives inputs, processes them using computational functions called models, and produce outputs (Figure 19). Inputs can be sent in real-time as Open Sound Control (OSC) messages. Models in Wekinator are constructed by providing training examples of inputs and the correspondent outputs. Wekinator uses supervised machine learning algorithms to build the models.
Dynamic Time Warping (DTW) is a technique to find an optimal alignment between two given (time-dependent) sequences [28]. In Wekinator, a training example in Dynamic Time Warping, captures how the inputs have changed over a period of time. Wekinator needs only one DTW training example of any gesture class to be recorded before it can operate. Once the model is running, a blue bar displays the degree of similarity between the current gesture the user is performing and the closest example that is recorded for each gesture type. When the similarity of the closest match is above the set threshold, Wekinator sends out the OSC message corresponding to that gesture type. The threshold can be adjusted interactively using a slider in the application window. We use a Dynamic Time Warping (DTW) model in Wekinator to train and detect motion gestures in our user studies. MagicWand sends IMU data wirelessly to a host computer running the Wekinator engine. Once the machine-learning engine is trained with gesture data examples, it can be used to identify gestures in real-time (Figure 20).
Figure 20. A Dynamic Time Warping model interface on Wekinator.
Chapter 5

Fantasy Game Application

5.1 Introduction

In the realm of fantasy literature, a wand is a quasi-sentient instrument that helps a witch or wizard to concentrate their magical powers and centralise the effects for more complex results. Children are inspired by these stories in their imaginative play through the use of sticks that become wands casting spells to defeat imaginary magical beasts, or swords to fight imaginary fire breathing dragons.

To highlight some of MagicWand’s capabilities, we developed a first-person fantasy adventure game [36] for aspiring wizards. This game requires the player to use several tools that fit a cylindrical form factor — a candle, a wand, a key, a whirlwind, a sword, a magic potion, and a spoon — to interact with characters that fit a cylindrical form. The game’s goal is to collect all the characters and transport them from the wand display to a virtual magical island on a large external display (Figure 21), through the application of gestures.

5.2 Game Play

During the game, tools and characters are rendered on the MagicWand, with the player having to discover the gesture needed to put the tool or character into action. We believe that the visual representation of the game objects, supplemented with MagicWand’s unique physical affordances, make it easier for players to discover new gestures that operate the tool. Therefore, we designed the game with gestures and graphics that fit the cylindrical form factor of MagicWand. Informal and qualitative evaluations were also conducted on user interactions.
5.2.1 Motion Gestures

In the fantasy game application, the player must perform motion gestures to advance through different stages of the gameplay. We implemented four main gestures: Rotation, Swirl, Slash, and Tilt. As presented in Chapter 3, these gestures were designed based on motions of everyday object handling and interactions with gesture-based game controllers. The gestures are illustrated in Figure 22 and presented with game scenarios in section 5.2.2.
Figure 22. Gestures: (a) Rotation, (b) Swirl, (c) Slash, (d) Tilt.
5.2.2 Game Scenarios

The game scenarios of the fantasy game application are designed with interactions that fit the cylindrical shape. The game characters and tools should fit the cylindrical form factor of MagicWand providing visual clues for users to discover the gestures needed to put the game objects into action.

The game commences with a candle being displayed on the MagicWand. The player picks up the MagicWand and blows out the candle. This action activates the magical powers and presents the player with a virtual magic wand. Next, the user is presented with four game stages. Each stage begins with a virtual wand on the MagicWand’s display, with the player performing a spell-casting gesture to activate the large display to continue the game play. A spell-casting gesture is a compound motion of a Swirl followed by a Slash gesture.

Scenario 01:

In the first scenario of the game, a virtual wand is shown on the MagicWand’s display and the player performs a spell-casting gesture. This action activates magical powers and shows a guardian character on the MagicWand. The player then Rotates the device to explore the guardian and finds a key on the back of the character. Discovering the key presents the player with a dungeon door on the large screen. At this point of the game, a key is displayed on the MagicWand, allowing the player to perform a Rotate gesture to unlock a dungeon door. Unlocking the door completes the first game scenario and takes the player to the second level.

Scenario 02:

The second scenario of the game also starts with a virtual wand on the MagicWand’s display. The player performs a spell-casting gesture, which displays a character floating inside an
animated tornado-like whirlwind on the MagicWand. With a Swirl gesture, the player accelerates the whirlwind propelling the character to the large screen, as well as over obstacles to reach a goal. The game scenario completes upon reaching the goal and takes the player to the next scenario.

Scenario 03:

In the third game scenario, performing a spell-casting gesture shows a sword on the MagicWand and a spider web obstructing the path of a game character on the larger screen. The player performs a Slash gesture to use the sword to destroy the spider web. Once the spider web is destroyed and the path is clear, the game character on the large screen moves to the magical island allowing the player to proceed to the next scenario.

Scenario 04:

Similar to the previous scenarios, this game stage also starts with a virtual wand on the MagicWand’s display. The player performs a spell-casting gesture, which displays a game character floating inside a magic potion-like a liquid. A Tilt gesture allows the player to pour the liquid into a cauldron on the large screen to prepare a magic potion. This action presents the user with a spoon on the MagicWand. The player uses the spoon to stir the cauldron by performing a Swirl gesture. This completes the preparation of the magical potion that transfers all the game characters safely back to their magical island concluding the gameplay.

Figure 23 demonstrates a player performing the motion gestural interactions with MagicWand in scenario one and three to advance through the gameplay.
Figure 23. A player interacting with MagicWand, (a) unlocking a dungeon door in Scenario 01, (b) destroying a spider web that obstructs the player’s path in Scenario 03.
5.3 Application Implementation

5.3.1 Software

The game application was developed in Unity 3D, which runs on a server, while a second Unity application was developed to run independently on the two Android controller boards on the MagicWand. The server communicates with MagicWand over a local Wi-Fi network using Open Sound Control (OSC) messages. It receives and processes the incoming gestural data and updates graphics to be displayed on the wand.

In this system, MagicWand acts as a game controller. A large external screen is used as the primary game display. The server controls this display as well, running the game and coordinating all components. To recognize motion gestures performed by the player, we used the Android Gesture Recognition Tool, running on the MagicWand, as described in Chapter 4.

Figure 24. Game characters and tools displayed on MagicWand.
5.3.2 3D Graphics

We used free 3D models from Mixamo [1] for characters and free models from the Unity assets store for tools and other game elements. Figure 24 displays some of the game characters and tools rendered on the wand.

With two FOLED screens that cover almost all its lateral surface, MagicWand can provide the visual illusion that displayed objects are enclosed within the device. Visual perspective correction is used to strengthen this illusion of a 3D display. When the wand is rotated, the graphics are rotated per IMU data to give the illusion of motion parallax (Figure 25).

Figure 25. When the wand is rotated, the graphics circumnavigate according to IMU data giving the illusion of motion parallax.
5.4 Qualitative Findings and Observations

Through informal user testing at the Human Media Lab and interactivity demonstrations at the ACM Conference on Human Factors in Computing Systems (CHI) 2016 [36], we noticed that when provided with a flat smartphone users grasped the device with the fingers applying pressure along the edges of the phone. In contrast, the users grasped the MagicWand with all the fingers wrapped around the device with a comfortable grip making it ideal for wrist-based gestural interactions.

As flat displays have a discrete display area limited to one side of the device, we also noticed that the visibility is reduced when the device is rotated (Figure 26). This made it difficult for users to rotate the smartphone to explore 3D characters on display. In contrast, a cylindrical display offered a 360-degree continuous display area around the device (Figure 25).

Figure 26. Flat displays have a display area that can only be viewed from one side, and the visibility is reduced when the device is rotated.
<table>
<thead>
<tr>
<th>Form</th>
<th>Visibility</th>
<th>Grasp Affordance</th>
<th>Gestural Affordance</th>
</tr>
</thead>
<tbody>
<tr>
<td>Flat</td>
<td>Discrete</td>
<td>Fingers</td>
<td>Finger motion (touch)</td>
</tr>
<tr>
<td>Cylindrical</td>
<td>Continuous</td>
<td>Hand</td>
<td>Wrist motion (rotate)</td>
</tr>
</tbody>
</table>

Table 1. Comparison of affordances offered by controllers with cylindrical and flat displays.

From our qualitative findings, we observed that compared to a Flat handheld display, a Cylindrical form factor provides several affordances worth exploring in gameplay (Table 1). To test this hypothesis more quantitatively, we conducted a user based elicitation study to create a gesture set for both Cylindrical and Flat device scenarios and compared the gesture discovery time as a mean of measuring affordance of each display Shape. This work will be discussed in detail in chapters 6 and 7.
Chapter 6

Experiment 1: Elicitation Study

One important observation from our experimentation with gaming was that the physical affordances of MagicWand could affect the speed with which users learned gestures. To test this hypothesis more quantitatively, we conducted two experiments. The purpose of both experiments was to compare gestural affordances between two display form factors; a Cylindrical and a Flat smartphone. To do this, we first needed to elicit a user-defined gesture set for each device condition. The first experiment focused on developing a user-defined gesture set for both Cylindrical and Flat display form factors. In the first experiment, we asked participants to design gestures for a list of common Graphical User Interface tasks. Each gesture was created independently for each display condition. We specifically focused on what user-generated gestures map to commonly used mobile OS actions.

6.1 Participants

20 paid participants volunteered in this experiment (9 females). The mean age of participants was 24.6 years. 17 participants were right-handed. It was imperative for the experiment that the participants understood the tasks and grasped the concept of creating motion gestures that triggered computer actions. Therefore, we recruited participants with prior experience with gesture-based controllers, such as the Nintendo Wiimote, Microsoft Kinect, or Sony PlayStation Move.
6.2 Display Devices

The main factor was display device Shape, with two levels: Flat and Cylindrical, as shown in Figure 27. MagicWand was used as the Cylindrical display form factor and an off-the-shelf Android smartphone as the Flat display form factor. We implemented a Unity application that presented simple Graphical User Interface actions on each device.

Figure 27. Participant creating a gesture for “Menu” in the first experiment, with MagicWand (top) and a flat phone (bottom).
6.3 Measurements

The study was conducted independently for both the Flat and Cylindrical devices. An Android application running on each device streamed IMU data to a logging application running on the experiment computer that recorded motion data for each gesture. In addition, all the sessions were video recorded. Each frame of all the videos was timestamped and synchronized with gesture data.

6.3.1 Data recording

In our first study, we recorded gesture data from both devices by logging IMU data through Open Sound Control (OSC) protocol. This data was used to train a gesture recognition engine for the second experiment. Figure 28 illustrates a sample from a data file.

```
[user id, device id, action id, action name, accelerometer x, y, z, rotation, timestamp]

P2  D1  A1  Move SA1  0.2791243 -0.6085902 0.6746719 0.9698651 14:13:09:95
P2  D1  A1  Move SA1  0.2791243 -0.6085902 0.6746719 0.9698651 14:13:09:97
P2  D1  A1  Move SA1  0.2781471 -0.6113971 0.667597 0.9725931 14:13:09:99
P2  D1  A1  Move SA1  0.2909574 -0.6049306 0.6560066 0.9769287 14:13:10:01
P2  D1  A1  Move SA1  0.3129182 -0.6113971 0.6695482 0.9819818 14:13:10:02
P2  D1  A1  Move SA1  0.3240201 -0.5905348 0.6591792 0.98912 14:13:10:04
P2  D1  A1  Move SA1  0.3240201 -0.5905348 0.6591792 0.98912 14:13:10:05
P2  D1  A1  Move SA1  0.3226773 -0.5840682 0.646856 0.9978275 14:13:10:07
P2  D1  A1  Move SA1  0.3226773 -0.5840682 0.646856 0.9978275 14:13:10:09
P2  D1  A1  Move SA1  0.3138938 -0.5863866 0.6779675 1.006329 14:13:10:10
P2  D1  A1  Move SA1  0.3114541 -0.5849224 0.6895563 1.013807 14:13:10:12
P2  D1  A1  Move SA1  0.3114541 -0.5849224 0.6895563 1.013807 14:13:10:14
P2  D1  A1  Move SA1  0.3210917 -0.5996854 0.6889464 1.012501 14:13:10:15
```

Figure 28. A snippet from an IMU data file. First line shows the data type of each column.
<table>
<thead>
<tr>
<th>Category</th>
<th>Sub-Category</th>
<th>Task Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>Action</td>
<td>System Level</td>
<td>1. Answer Call</td>
</tr>
<tr>
<td></td>
<td></td>
<td>2. Hang-up Call</td>
</tr>
<tr>
<td></td>
<td></td>
<td>3. Ignore Call</td>
</tr>
<tr>
<td></td>
<td></td>
<td>4. Place Call</td>
</tr>
<tr>
<td></td>
<td></td>
<td>5. Voice Search</td>
</tr>
<tr>
<td></td>
<td>Application Specific</td>
<td>6. Act on Selection</td>
</tr>
<tr>
<td></td>
<td></td>
<td>7. Hold Selection</td>
</tr>
<tr>
<td></td>
<td></td>
<td>8. Menu (Show, Hide)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>9. Okay</td>
</tr>
<tr>
<td></td>
<td></td>
<td>10. Cancel</td>
</tr>
<tr>
<td></td>
<td></td>
<td>11. Undo</td>
</tr>
<tr>
<td></td>
<td></td>
<td>12. Redo</td>
</tr>
<tr>
<td></td>
<td></td>
<td>13. Cut</td>
</tr>
<tr>
<td></td>
<td></td>
<td>14. Paste</td>
</tr>
<tr>
<td></td>
<td></td>
<td>15. Delete</td>
</tr>
<tr>
<td></td>
<td></td>
<td>16. Duplicate</td>
</tr>
<tr>
<td>Navigation Based</td>
<td>System Level</td>
<td>17. Home</td>
</tr>
<tr>
<td></td>
<td></td>
<td>18. Next Application</td>
</tr>
<tr>
<td></td>
<td></td>
<td>19. Previous Application</td>
</tr>
<tr>
<td></td>
<td></td>
<td>20. Open</td>
</tr>
<tr>
<td></td>
<td></td>
<td>21. Close</td>
</tr>
<tr>
<td></td>
<td>Application Specific</td>
<td>22. Move (Up, Down, Left, Right)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>23. Next</td>
</tr>
<tr>
<td></td>
<td></td>
<td>24. Previous</td>
</tr>
<tr>
<td></td>
<td></td>
<td>25. Zoom In</td>
</tr>
<tr>
<td></td>
<td></td>
<td>26. Zoom Out</td>
</tr>
<tr>
<td></td>
<td></td>
<td>27. Rotate (CW, CCW)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>28. Scroll (Up, Down, Left, Right)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>29. Navigate List (Up, Down)</td>
</tr>
</tbody>
</table>

Table 2. The list of tasks used in the participatory design study, grouped by category.
6.4 Tasks

Based on prior guessability studies [39, 51], we first created a list of 29 tasks covering common tasks that users perform on mobile platforms. Tasks were classified into two main categories: actions and navigation-based tasks. The list of tasks was further classified into two sub-categories: system-level or application-specific tasks and is shown in Table 2.

To further simplify the list, actions pertaining to the same task that were differentiated only by directionality were merged together. For example, the actions Move Up, Move Down, Move Left, Move Right were grouped as the Move task. Duplicate tasks and tasks that were unavailable on the wand (e.g., minimize) were removed.
Procedure and Task

Participants were handed one of the two display conditions (Cylindrical and Flat) and presented with a task from the list, followed by a verbal explanation. They were then shown an animation of the result of the task on the device. For example: “Open a menu” resulted in a menu depicted on the device (see Figure 29). Next, participants were asked to create and perform a gesture that they thought was the most appropriate for this task.

Furthermore, participants were asked to hold the device with their dominant hand and use only one hand when performing gestures. They were encouraged to invent and perform various motion gestures by moving the device freely in the air making wrist and arm movements. Since touch input was not an option, instructions were given to only use motion gestures for all the
tasks. We also encouraged the participants to invent unique gestures for different tasks, however, the participants were allowed to repeat gestures if they considered it necessary.

The device conditions were presented in counterbalanced order, while tasks were presented using a Latin square. Subjects were instructed to perform gestures for all 29 tasks per device condition.

For each task on each device, participants performed a single-handed gesture. The participants were allowed to try more than one gesture and use the most preferred one as their final choice. In our elicitation study, we followed a similar procedure presented in Wobbrock et al.’s [51] guessability study on surface gestures. Following Wobbrock’s work, immediately after performing each gesture, participants were given two 7-point Likert scales on gesture goodness and ease. The experiment concluded with the experimenter asking the participants for comments, feedback and suggestion on their experience with the two display devices.

6.6 Experiment 1: Results

6.6.1 Gesture Elicitation

38 elicited gestures were collected per device for each of the 20 participants, resulting in a total of 1520 gestures. To identify and label the unique gestures, two researchers browsed the video recordings of all gestures and agreed on a coding scheme. A sample consisting of 100 randomly chosen gestures per device (a total of 200 random gestures) were then coded by both researchers. The inter-rater agreement was calculated using the Cohen’s Kappa test, obtaining an agreement of 0.83. The remaining gestures were then coded by one of the researchers.
A total of 122 unique gestures were identified for the *Cylindrical* display form factor and 102 unique gestures for the *Flat* display form factor. 54 gestures were common to both devices, 68 were exclusive for the *Cylindrical* device and 48 were exclusive for the *Flat* device. On average, a participant created 19 unique gestures for each device.

The *Tilt* gesture was the most frequently used with both display shapes, at 41% (312 out of 760 individual gestures) for the *Cylindrical* display device and at 41.8% (318 out of 760 individual gestures) for the *Flat* display device. On average, each task contained 3 unique gestures that involved a *Tilt* motion in both device conditions.

The *Cylindrical* shape had a higher percentage of gestures that involved rotational motions than the *Flat* shape. 10.8% of the gestures performed on the *Cylindrical* device had rotational motion along y-axis and 6% were swirl motions, whereas only 8.3% of the gestures performed on the *Flat* device involved rotational motion along y-axis and only 2.9% were swirl motions.

To evaluate the degree of consensus among our participants, we followed the procedure indicated by Wobbrock et al. [51] to calculate an *agreement score* for each task. The definition of an *agreement score*, \( A(t) \), for a given task \( t \) for which feedback has been elicited from multiple participants during a guessability study was introduced by Wobbrock et al. [51] as the following sum of square ratios:

\[
A(t) = \sum_{P: i \subseteq P} \left( \frac{|P_i|}{|P|} \right)^2
\]

where \( P \) is the set of proposed gestures for task \( t \), \(|P|\) the size of the set, and \( P_i \) is a subset of identical gestures from \( P \). The range for \( A(t) \) is \([0, 1]\). As an example of an *agreement score*
calculation, the task *answer call* had 5 groups with sizes of 14, 3, 1, 1, and 1. Therefore, the agreement score for *answer call* for the *Flat* device, we compute:

\[
A_{\text{answer call}} = \left(\frac{14}{20}\right)^2 + \left(\frac{3}{20}\right)^2 + \left(\frac{1}{20}\right)^2 + \left(\frac{1}{20}\right)^2 + \left(\frac{1}{20}\right)^2 = 0.52
\]

This score determines the degree of consensus in the creation of gestures among participants. An agreement score of 1 indicates that all participants performed the same motion gesture for a given task. In contrast, an agreement score of 0 indicates that all participants performed different gestures for that task. Figure 30 shows the agreement score for all the tasks for both display form factors.

The top two agreement scores were for *Answer Call* and *Rotate* in both devices (\(A_{\text{rotate}} = 0.73\), \(A_{\text{answer call}} = 0.51\) for *Cylindrical* and \(A_{\text{rotate}} = 0.81\), \(A_{\text{answer call}} = 0.52\) for *Flat*). Then were the *Move* tasks followed by *Next* and *Previous*. Agreement scores from our user-defined motion gestures are similar to those shown for Wobbrock et al.’s [51] gesture set for surface computing and Ruiz et al.’s [39] gesture set for mobile interaction. Simple tasks such as *Rotate, Move, Next,* and *Previous* had high agreement scores, whereas the complex tasks such as *Undo, Redo,* and *Delete* had low scores. We believe that the conceptual complexities of tasks correlated inversely with their agreement, as more complex tasks elicited lesser gestural agreement.
Each participant was given two 7-point Likert scales on gesture goodness and ease, immediately after performing each gesture. We used data collected from these Likert scales to calculate goodness and ease ratings for each gesture in both device conditions.

68.9% of the 29 tasks on the Cylindrical device trended higher goodness ratings than the Flat device. Ease ratings were more evenly split at 51.72% of tasks trending higher on the Cylindrical device. A discussion on the results of this experiment is presented in Chapter 8.

Figure 30. Agreement for each task sorted in descending order for the Cylindrical display device shown in blue. Gesture agreement for Flat display device is shown in orange.
Chapter 7

Experiment 2:

Effects of Display Form Factors on Gesture Discovery Time

In this chapter, we will discuss the second experiment that was designed to evaluate and compare gestural affordances between Cylindrical and Flat display form factors, by measuring the gesture discovery time for each device condition.

7.1 Categorizing Gestures and Measurement

This experiment used unique gestures identified in the first experiment. The top unique gestures for each task were selected using the following formula: gestures used by more than 15% of participants were marked as the top gestures until the totaled gesture sets equaled 50% of the participants. As an example, if a task had 6 groups of identical gestures each with 30%, 20%, 15%, 15%, 10%, 10%, the groups with 30% and 20% were selected as the top gestures because it equaled 50% of the study’s participants. We then used the top gestures for each task as a training example to create a Dynamic Time Warping (DTW) model in Wekinator, independently for each device. We focused the analysis only on the 20 tasks in which the gesture sets were identical between device conditions. This was done to keep the gestures as a constant between device conditions for each task. These unique tasks are presented in Table 3. Figure 31 illustrates an overview of the top gestures and the tasks they pertained to. These gestures were then used as guessability targets in the second study: Tilting the device along various axes, Moving or Shaking the device along various axes, and bringing the device to the ears and mouth. The time it took for participants to discover a gesture, per task, per device was measured.
<table>
<thead>
<tr>
<th>Category</th>
<th>Sub-Category</th>
<th>Task Name</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Action</strong></td>
<td>System Level</td>
<td>1. Answer Call</td>
</tr>
<tr>
<td></td>
<td></td>
<td>2. Ignore Call</td>
</tr>
<tr>
<td></td>
<td></td>
<td>3. Place Call</td>
</tr>
<tr>
<td></td>
<td></td>
<td>4. Voice Search</td>
</tr>
<tr>
<td></td>
<td>Application Specific</td>
<td>5. Act on Selection</td>
</tr>
<tr>
<td></td>
<td></td>
<td>6. Hold Selection</td>
</tr>
<tr>
<td></td>
<td></td>
<td>7. Menu (Show, Hide)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>8. Okay</td>
</tr>
<tr>
<td></td>
<td></td>
<td>9. Cancel</td>
</tr>
<tr>
<td></td>
<td></td>
<td>10. Undo</td>
</tr>
<tr>
<td></td>
<td></td>
<td>11. Redo</td>
</tr>
<tr>
<td></td>
<td></td>
<td>12. Paste</td>
</tr>
<tr>
<td></td>
<td></td>
<td>13. Duplicate</td>
</tr>
<tr>
<td><strong>Navigation Based</strong></td>
<td>System Level</td>
<td>14. Home</td>
</tr>
<tr>
<td></td>
<td></td>
<td>15. Open</td>
</tr>
<tr>
<td></td>
<td></td>
<td>16. Close</td>
</tr>
<tr>
<td></td>
<td>Application Specific</td>
<td>17. Zoom In</td>
</tr>
<tr>
<td></td>
<td></td>
<td>18. Zoom Out</td>
</tr>
<tr>
<td></td>
<td></td>
<td>19. Rotate (CW, CCW)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>20. Navigate List (Up, Down)</td>
</tr>
</tbody>
</table>

Table 3. The list of 20 tasks selected for the second experiment.
Figure 31. Top gestures used for discovery in the second experiment
7.2 Participants

12 paid participants volunteered for this study (8 females). The mean age of participants was 21.5 years and all participants were right handed. Similar to the first experiment, all participants had previous experience with motion gesture controllers such as Nintendo Wiimote, Microsoft Kinect, and Sony PlayStation Move. None of the participants were involved in the first experiment.

7.3 Apparatus

The second experiment was conducted using the same device conditions as the first experiment. We developed a Unity application, which ran on both MagicWand and the Android smartphone to present each task and their relevant animations. The application streamed IMU data to the experiment computer running Wekinator with our trained Dynamic Time Warping (DTW) model. When a gesture was recognized, this computer would register the completion time and send a signal to the display device to display the result from the action.

7.4 Hypothesis

We hypothesized that the Cylindrical device would have faster gesture discovery times, compared to the Flat device. This hypothesis was based on the physical affordances of the wand-like form factor facilitating movements of the wrist, resulting in faster gesture discovery and better performance.
7.5 Procedure and Task

Similar to the first experiment, participants were first handed a display device and presented with a task followed by a verbal explanation. At the end of the explanation, participants were shown a recorded animation on the device. Next, participants were asked to discover and perform a gesture for the given task. Participants were instructed to hold the device with their dominant hand and perform one-handed gestures. They were asked not to use touch input.

Subjects were given 2 minutes to discover the correct gesture for the task, after which they moved on to the next task. The device conditions were presented in counterbalanced order, while tasks were presented using a Latin square. Subjects were instructed to perform gestures for all 29 tasks per device condition. However, to keep the gestures and tasks as constants between the two device conditions, we focused the analysis on the 20 tasks for which the top gesture sets were identical between device conditions (see Table 3).

7.6 Experiment 2: Results

The second experiment measured gesture discovery time for the 20 tasks with identical gesture sets between devices. In the Flat condition, 2 participants took longer than 2 minutes to discover a gesture set. To not overly penalize this condition, we substituted these with the slowest discovery time for any user in that condition (117 seconds). Table 4 illustrates results from a two-tailed t-test for the 4 tasks that showed significant differences between conditions, with t statistics.
Out of 20 tasks shown in Table 3, the *Cylindrical* shape condition had significantly faster gesture discovery times in 20% of tasks. There were no cases where gesture discovery was faster for the *Flat* shape condition. Gesture discovery for the *Ok*, *Cancel*, *Duplicate* and *Rotate* tasks were 75%, 68%, 170% and 35% slower in the *Flat* condition than in the *Cylindrical* condition.

<table>
<thead>
<tr>
<th>Task</th>
<th><em>Cylindrical</em></th>
<th><em>Flat</em></th>
<th>t (11)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Okay</td>
<td>3.0 (0.6)</td>
<td>5.3 (1.2)</td>
<td>-3.109</td>
</tr>
<tr>
<td>Cancel</td>
<td>4.75 (1.0)</td>
<td>8.0 (2.0)</td>
<td>-1.815</td>
</tr>
<tr>
<td>Duplicate</td>
<td>18.32 (7.4)</td>
<td>49.39 (13.4)</td>
<td>-2.752</td>
</tr>
<tr>
<td>Rotate</td>
<td>1.83 (0.94)</td>
<td>2.47 (0.26)</td>
<td>-2.136</td>
</tr>
</tbody>
</table>

Table 4. Means, standard errors (s.e.) and t-values for actions with significantly faster gesture discovery time (seconds) on the *Cylindrical* display device (p < 0.05).
Chapter 8

Discussion

In this chapter, some of the key findings will be summarized. Furthermore, this chapter will provide a discussion of the observations in both experiments presented in Chapters 7 and 8.

8.1 Gesture Discovery Times

Results from the second experiment showed it is possible to measure differences between device shapes in terms of the gesture discovery time. Even when gesture recognizers are trained independently, using IMU data recorded only for the specific device, we observed that gestures for 20% of the actions were discovered significantly faster with MagicWand. In our first experiment, the Tilt gesture, involving flexion of the wrist, was the prevailing gesture type, at 41% of the total gestures performed for the Cylindrical display device. Similarly, all the gestures with faster discovery time involved some form of Tilt action. We believe this was facilitated by the physical affordances of the Cylindrical shape aiding rotations of the wrist joint.

8.2 Directionality

The first experiment also demonstrated that tasks that can be considered in pairs always resulted in a similar gesture that differentiated only in directionality. For example, tilting away from the body was the most frequently used gesture for Open, and participants most frequently used tilting towards body for Close.
8.3 Navigating

In navigational tasks (e.g., Scroll) using the Flat shape, participants mostly performed gestures involving translation motions on the XY plane involving wrist deviation (i.e., moving the device up, down along the y-axis and left, right along the x-axis). However, with the Cylindrical shape top gestures involved both translational and rotational motions involving pronation and supination of the wrist (move the device up, down along the y-axis and rotate left, right around the y-axis). We also observed that some users preferred to move the device relative to an external viewport, while others preferred to scroll the viewport inside the device [21] (Figure 32).

On the Flat device, the consensus was to move relative to an external viewport (i.e., move the device to the left to move to the left in a map) in all four directions. Interestingly, on the Cylindrical device, the consensus was also to move relative to an external viewport, but only along the y-axis. Users preferred to scroll the viewport (i.e., rotating the device to the right around the y-axis to scroll to the left) when translating the map along the x-axis. We believe that this behavior was due to the cylindrical and asymmetrical shape of the device. A similar behavior was seen in the top gestures for Next and Previous tasks.
Figure 32. Navigating by moving the device relative to an external viewport (left) in contrast to moving the viewport inside the device (right).

8.4 Selecting and Duplicating

When moving a selection between items, some users preferred the air bubble metaphor [35] on the Cylindrical device. This was clearly observed when moving the selection to the left or right where the participants rotated the device around its y-axis. The top gestures for the Duplicate task were always a simple gesture performed twice (e.g., tilt away from the body two times).

8.5 Phone Tasks

Participants were not instructed to consider MagicWand as a phone and experimenters did not mention the existence of a microphone or a speaker. However, participants performed gestures on an imaginary microphone and speaker on the Cylindrical device when performing
Answer Call, Place Call and Voice Search tasks. Furthermore, participants mentioned that it felt natural to perform those tasks with the Cylindrical device as the shape closely resembled a handset of an old-fashioned wired telephone:

“Phone calling feels a lot like a phone handle on older home phones and movements felt more natural” (P19).

8.6 Other User Comments

User comments seemed to indicate it was easier to perform gestures with the Cylindrical display device. Participants often mentioned that the Cylindrical device was “natural” to hold and had a firm grip, which better facilitated gestural interaction.

“[The] cylindrical device was easier to grip than the flat device” (P17).

Most participants commented that they felt like the Flat device could be “dropped more easily” when performing gestures.

“It is more natural to hold the cylinder because you can firmly grasp it and move gestures, whereas with a flat device, it could fly out of your hand because you only touch it on two points” (P9).

“[The] smartphone feels awkward in my hand, that it could fall out and break when making gestures. Cylindrical device felt comfortable” (P13).

Users suggested that the Cylindrical device is better for scrolling tasks, navigation tasks, reading documents, music applications, and gaming applications. They further added that the Flat device is better for applications where the entire display should be visible at one glance, and the Cylindrical device is better for scrolling actions.
Chapter 9

Conclusion, Limitations and Future Work

9.1 Conclusion

In this thesis, we presented MagicWand, a cylindrical display device consisting of two 5.5” FOLED displays wrapped around a 3D printed body. With motion gestural interaction as the primary input method, MagicWand provides a wand-like form factor that affords natural rotations of the wrist. MagicWand was designed as a self-contained, handheld, and a gesture-based mobile device with a near-continuous display on its lateral surface.

To demonstrate interactions methods, we implemented a fantasy game application where players used the MagicWand as a game controller. Our qualitative observations suggested that the cylindrical form factor of MagicWand affords easy movements of the wrist when performing gestures.

We quantitatively studied whether the physical affordance of the display shape could affect the speed with which users discovered motion gestures, comparing the guessability of gesture sets between a Flat smartphone and a Cylindrical display conditions. First, we elicited gestures for invoking common GUI tasks in both device conditions, collecting over 100 unique gestures for 29 tasks. Secondly, a recognition engine was trained to recognize the top gestures for each task, in each device condition. A second study measured the time it took for participants to discover this top gesture set, per task, per device. We only analyzed those 20 actions in which gesture sets were identical between conditions. Results suggest significantly faster guessing of motion gestures in the Cylindrical condition over the Flat smartphone in 20 percent of cases all
involving tilt. We attribute this to the different physical affordances provided by each device shape, leading to faster use of wrist flexion with the cylindrical device.

9.2 Future Work

MagicWand was designed with specific design goals and under particular technical constraints. Although the empirical findings support the design rationale, many of the details in the experimental results pertain only to the specific hardware implementation. This section discusses how the implementation could be improved in future work.

Our studies were limited in scope to specific device conditions. Both the Cylindrical and Flat devices primarily relied solely on gesture control. All tasks were triggered only using motion gestures, which can be more difficult to discover than touch interactions [53]. Based on these limitations, future work needs to be conducted to explore the interactions with both touch input and gestures combined.

In both experiments (Chapter 6 and Chapter 7), the user interfaces were created with simple 2D shapes. However, a future study could explore how different graphical representations influence gesture discovery. For example, a user might use different gestures for rotating a 2D square than they would for a 3D cube. Similarly, it would be interesting to explore how animated user interfaces influence direction and speed of gestures.

Even though the user feedback was in favour of the Cylindrical shape for gestural interaction, some users preferred the device to have a smaller diameter for comfortable grasp. The size of the MagicWand was constrained by the controller boards. With newer technology, such as flexible printed circuits, this constraint could be removed.
We designed MagicWand to have a fixed cylindrical form factor with permanently enfolded displays. However, the prototype could be extended to have rollable and unrollable displays with the advancement of FOLED display technologies in the future. This feature was also mentioned in user feedback during the elicitation studies.

In this thesis, MagicWand is presented as a single display device. In the future, it would be interesting to explore use cases with multiple MagicWands. Similar to Display Blocks [33], two or more MagicWands could potentially interact with each other to present novel interaction methods. In addition, a future implementation could make wands that snap together allowing a user to increase or decrease the display size. This feature could introduce new interactions in video games where MagicWand is the game controller. It is also worth exploring how haptic feedback could enhance the user experience of a cylindrical display device [10], which could be an essential improvement for a game controller.
References


Appendix A

Gesture Discovery Times from Experiment 2
<table>
<thead>
<tr>
<th>Task</th>
<th>Cylindrical</th>
<th>Flat</th>
<th>t (11)</th>
<th>p-value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Okay</td>
<td>3.0 (0.6)</td>
<td>5.3 (1.2)</td>
<td>-3.109</td>
<td>.005</td>
</tr>
<tr>
<td>Cancel</td>
<td>4.75 (1.0)</td>
<td>8.0 (2.0)</td>
<td>-1.815</td>
<td>.048</td>
</tr>
<tr>
<td>Duplicate</td>
<td>18.32 (7.4)</td>
<td>49.39 (13.4)</td>
<td>-2.752</td>
<td>.009</td>
</tr>
<tr>
<td>Rotate</td>
<td>1.83 (0.94)</td>
<td>2.47 (0.26)</td>
<td>-2.136</td>
<td>.028</td>
</tr>
<tr>
<td>Act on Selection</td>
<td>4.27 (1.5)</td>
<td>7.5 (2.47)</td>
<td>-1.096</td>
<td>.148</td>
</tr>
<tr>
<td>Zoom In</td>
<td>11.7 (5.0)</td>
<td>20.52 (8.39)</td>
<td>-.853</td>
<td>.206</td>
</tr>
<tr>
<td>Zoom Out</td>
<td>5.73 (3.2)</td>
<td>5.37 (1.55)</td>
<td>.098</td>
<td>.462</td>
</tr>
<tr>
<td>Home</td>
<td>32.11 (11.13)</td>
<td>17.70 (7.82)</td>
<td>1.432</td>
<td>.090</td>
</tr>
<tr>
<td>Menu</td>
<td>2.93 (0.44)</td>
<td>3.82 (0.39)</td>
<td>-1.335</td>
<td>.104</td>
</tr>
<tr>
<td>Open</td>
<td>3.57 (0.49)</td>
<td>4.24 (0.67)</td>
<td>-1.046</td>
<td>.159</td>
</tr>
<tr>
<td>Close</td>
<td>6.32 (1.47)</td>
<td>6.92 (2.62)</td>
<td>-.192</td>
<td>.425</td>
</tr>
</tbody>
</table>

Means, standard errors (s.e.) and t-values for 20 actions with gesture discovery times in seconds. Actions with significantly faster gesture discovery in the Cylindrical condition are shown in bold (p < 0.05).
<table>
<thead>
<tr>
<th>Task</th>
<th>Cylindrical</th>
<th>Flat</th>
<th>t (11)</th>
<th>p-value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Hold Selection</td>
<td>20.3 (6.89)</td>
<td>19.72 (9.2)</td>
<td>.059</td>
<td>.477</td>
</tr>
<tr>
<td>Navigate List</td>
<td>2.66 (0.55)</td>
<td>3.33 (0.63)</td>
<td>-.725</td>
<td>.241</td>
</tr>
<tr>
<td>Undo</td>
<td>12.37 (7.0)</td>
<td>9.12 (2.6)</td>
<td>.658</td>
<td>.262</td>
</tr>
<tr>
<td>Redo</td>
<td>2.72 (0.6)</td>
<td>3.07 (0.75)</td>
<td>-.881</td>
<td>.198</td>
</tr>
<tr>
<td>Paste</td>
<td>5.16 (1.9)</td>
<td>12.0 (4.15)</td>
<td>-1.427</td>
<td>.090</td>
</tr>
<tr>
<td>Voice Search</td>
<td>8.63 (3.9)</td>
<td>10.1 (4.9)</td>
<td>-.277</td>
<td>.393</td>
</tr>
<tr>
<td>Answer Call</td>
<td>5.6 (3.5)</td>
<td>5.2 (2.35)</td>
<td>.085</td>
<td>.467</td>
</tr>
<tr>
<td>Ignore Call</td>
<td>5.67 (2.1)</td>
<td>7.14 (2.44)</td>
<td>-.401</td>
<td>.348</td>
</tr>
<tr>
<td>Place Call</td>
<td>5.32 (1.8)</td>
<td>4.34 (0.64)</td>
<td>.505</td>
<td>.312</td>
</tr>
</tbody>
</table>

Means, standard errors (s.e.) and t-values for 20 actions with gesture discovery times in seconds. Actions with significantly faster gesture discovery in the *Cylindrical* condition are shown in bold (p < 0.05) (cont'd).
Appendix B

General Research Ethics Board Approval
July 14, 2016

Dr. Roel Vertegaal
Professor
School of Computing
Queen's University
Goodwin Hall
Kingston, ON, K7L 3N6

GREB Ref #: GCOMP-090-16; Romeo # 6018708
Title: "GCOMP-090-16 MagicWand: Evaluating Gestural Affordances for Operating a GUI on a Cylindrical DisplayObject by Measuring Gesture Discovery Time"

Dear Dr. Vertegaal:

The General Research Ethics Board (GREB), by means of a delegated board review, has cleared your proposal entitled "GCOMP-090-16 MagicWand: Evaluating Gestural Affordances for Operating a GUI on a Cylindrical DisplayObject by Measuring Gesture Discovery Time" for ethical compliance with the Tri-Council Guidelines (TCPS 2 (2014)) and Queen's ethics policies. In accordance with the Tri-Council Guidelines (Article 6.14) and Standard Operating Procedures (405.001), your project has been cleared for one year. You are reminded of your obligation to submit an annual renewal form prior to the annual renewal due date (access this form at http://www.queensu.ca/traq/signon.html; click on "Events"; under "Create New Event" click on "General Research Ethics Board Annual Renewal/Closure Form for Cleared Studies"). Please note that when your research project is completed, you need to submit an Annual Renewal/Closure Form in Romeo/traq indicating that the project is 'completed' so that the file can be closed. This should be submitted at the time of completion; there is no need to wait until the annual renewal due date.

You are reminded of your obligation to advise the GREB of any adverse event(s) that occur during this one year period (access this form at http://www.queensu.ca/traq/signon.html; click on "Events"; under "Create New Event" click on "General Research Ethics Board Adverse Event Form"). An adverse event includes, but is not limited to, a complaint, a change or unexpected event that alters the level of risk for the researcher or participants or situation that requires a substantial change in approach to a participant(s). You are also advised that all adverse events must be reported to the GREB within 48 hours.

You are also reminded that all changes that might affect human participants must be cleared by the GREB. For example, you must report changes to the level of risk, applicant characteristics, and implementation of new procedures. To submit an amendment form, access the application by at http://www.queensu.ca/traq/signon.html; click on "Events"; under "Create New Event" click on "General Research Ethics Board Request for the Amendment of Approved Studies". Once submitted, these changes will automatically be sent to the Ethics Coordinator, Ms. Gail Irving, at the Office of Research Services for further review and clearance by the GREB or GREB Chair.

On behalf of the General Research Ethics Board, I wish you continued success in your research.

Sincerely,

John Freeman, Ph.D.
Chair
General Research Ethics Board

c: Mr. Lahiru Priyadarshana, Co-investigator
Ms. Karilee Whiteway, Research Coordinator