COFFEE: CONTEXT OBSERVER FOR FAST ENTHRALLING ENTERTAINMENT

A Thesis

presented to

the Faculty of California Polytechnic State University

San Luis Obispo

In Partial Fulfillment

of the Requirements for the Degree

Master of Science in Electrical Engineering

by

Anthony Lenz

June 2014

COMMITTEE MEMBERSHIP

TITLE:	COFFEE: Context Observer For Fast Enthralling Entertainment
AUTHOR:	Anthony Lenz
DATE SUBMITTED:	June 2014
COMMITTEE CHAIR:	Lynne Slivovsky, Ph.D. Professor of Electrical Engineering
COMMITTEE MEMBER:	Franz Kurfess, Ph.D. Professor of Computer Science
COMMITTEE MEMBER:	Bridget Benson, Ph.D. Assistant Professor of Electrical Engineering

ABSTRACT

COFFEE: Context Observer For Fast Enthralling Entertainment

Anthony Lenz

Desktops, laptops, smartphones, tablets, and the Kinect, oh my! With so many devices available to the average consumer, the limitations and pitfalls of each interface are becoming more apparent. Swimming in devices, users often have to stop and think about how to interact with each device to accomplish the current tasks at hand. The goal of this thesis is to minimize user cognitive effort in handling multiple devices by creating a context aware hybrid interface. The context aware system will be explored through the hybridization of gesture and touch interfaces using a multi-touch coffee table and the next-generation Microsoft Kinect. Coupling gesture and touch interfaces creates a novel multimodal interface that can leverage the benefits of both gestures and touch. The hybrid interface is able to utilize the more intuitive and dynamic use of gestures, while maintaining the precision of a tactile touch interface. Joining these two interfaces in an intuitive and context aware way will open up a new avenue for design and innovation.

ACKNOWLEDGMENTS

I want to thank my thesis advisor Professor Lynne Slivovsky for dedicating so much time and effort into this project. Thank you so much for your continual instruction and collaboration throughout, as well as your willingness to put in the extra-time and energy when I put things off until the last minute. Thanks!

I would also like to extend thanks to my committee members Professor Franz Kurfess and Professor Bridget Benson. Thank you for your time and willingness to be on my committee and lending guidance when needed. Thank you Professor Kurfess for allowing me to use your class time and students to conduct my usability tests.

Thank you to my friends and collaborators:

- Andrew Weinfield, for taking the bull by the horns and creating an fantastic gesture recognition library custom tailored for this project. You did a fantastic job creating awesome gestures, despite using prototype hardware and having to optimize for sitting users that had a massive table in front of them.
- Troy Weber, for helping me design and build the coffee table, and being a sounding board during the whole process.
- Andrew Hughes, for your expert review and critique of the project. Thanks for all your personal and professional support throughout this project.

I also want to extend thanks to all the study participants who took time out of their day for testing and offering feedback.

Thank you to our contributors:

- Cal Poly College of Engineering, for the R-IDC funds that funded the entire project.

- Qualcomm, for their generous donation of a IFC6410 DragonBoard that powered the multi-touch coffee table.

- Microsoft, for giving us early access to the next-generation Kinect for Windows to use in this project.

I would also like to thank my lovely wife Becca Lenz. Without your constant encouragement and loving support, this project would have gotten nowhere. Thank you for your understanding of all those late nights and dealing with my continuous bouts of frustration. You are my best friend, and I love you for that!

Finally, I would like to thank my Lord and Savior Jesus Christ. Thank you for giving me the strength to get me through it all. Thank you for giving me insight into all the problems I faced, and giving me all those epiphanies while in the rabbit hole chasing bugs.

TABLE OF CONTENTS

APPENDICES

LIST OF TABLES

LIST OF FIGURES

1 INTRODUCTION

The proliferation of smartphones and other 'smart' devices has created a world that is ever increasingly connected. However, with all these devices there is a glaring issue of usability when transitioning between devices or trying to use two or more devices to accomplish a task. When using devices in combination, most interactions are superficial with devices acting as remotes or casting content to another device, e.g., Chromecast [2] or Apple TV [1], with very little true collaboration. Anytime a new technology becomes ubiquitous, that technology becomes ingrained into our everyday work-flow and requires careful consideration for Human-Computer Interaction (HCI) design. Poor HCI can lead to unnecessary interruptions and an increased cognitive load [16]. When the user's attention is drawn away from their current task to focus on how to interact with a system, productivity and/or enjoyment is lost. By using devices in combination, the user must often have to think about both interfaces and how they interact with them in order to accomplish the task at hand, creating a lackluster user experience.

This paper seeks to mitigate the troubles of using multiple devices by creating a framework for a hybrid interface that allows for easy collaboration between devices in an intelligible manner. The framework is built incorporating context into the interactions and designing interactions around cognitive models to allow for a more usable and natural user experience. This specific design focused on making a hybrid interface composed of an immersive/gesture interface in concert with a touch interface, though the underpinning concepts and techniques can be used with any type of interface. The implementation used a custom built multi-touch coffee table running Android for the touch interface and a multimedia PC equipped with the next generation Microsoft Kinect for the immersive/gesture interface. The system was created for a living room environment and focused on user interactions geared for entertainment.

The proposed system is important because it strives to create novel user interactions and simplify existing interactions. The main purpose is to streamline the use of multiple devices and establish a collaborative methodology of how content is presented to the user. Making the way people interact with computers more meaningful, by promoting the content and diminishing the underlying technology. In doing so, the hope is to contribute to the field of HCI and raise the general user's expectations of seamless and unobtrusive computer interfaces.

This paper is broken down into the following chapters: Chapter 2 gives a brief overview and the current trends of HCI, listing some related works. Chapter 3 presents the proposed system and describes how it allows for more natural user interactions. Chapter 4 walks through the implementation of the proposed system and lists the problems and concessions made in putting the concepts into practice. Chapter 5 lists the current limitations of the system. Chapter 6 presents the validation framework used to gauge the success of the system as a natural user interface. Chapter 7 presents the findings of a usability test to support whether the project was successful or not. Chapter 8 discusses the possibilities for expanding on the project to further extend and refine the prototype system to a level of practical use. Chapter 9 summarizes the outcomes of this paper and gives final thoughts on the viability of the system.

2 BACKGROUND

This chapter provides a brief overview and the current trends of HCI, highlighting some contemporary related systems.

2.1 HCI

Dix in *Human-computer interaction* defines Human-Computer Interaction (HCI) as, *"the study of the way in which computer technology influences human work and activities. The term computer technology now-a-days includes most technology from obvious computers with screens and keyboards to mobile phones, household appliances, in-car navigation systems and even embedded sensors and actuators such as automatic lighting."* [14] HCI also covers the interfaces in which humans physically interact with computer systems. The three main interfaces are WIMP (Windows, Icons, Menus, Pointer), touch, and immersive/gestures, with a proposed hybrid interface. With consideration to usability, each interface has advantages and drawbacks.

2.1.1 WIMP

The WIMP (Windows, Icons, Menus, and Pointer) interface is the most prevalent interface and can be found on almost every personal computer. Paired with the GUI (Graphical User Interface), the WIMP interface gained popularity in

1984 with the Macintosh computer and was later copied by Microsoft Windows on the PC [20]. Almost every desktop and laptop computer utilizes this type of interface which relies solely on a keyboard and mouse as the primary means of input. The reason for the dominance of the WIMP interface is based on being the first graphical interface that was not text based. It's this legacy that is the main advantage this interface has over the others, as tradition has rooted it as the de facto standard.

In addition to WIMP's legacy there are several other advantages that it offers as well. In Myers paper, *Past, Present and Future of User Interface Software Tools*, he makes the case that WIMP was well suited for the set of input tools available, and *"provided a path of least resistance"* for users and developers to have a consistent user interface with a low learning curve [17]. Dam, in *Post-WIMP User Interfaces*, also notes that the consistencies in look and feel from application to application in WIMP allow for a lower-learning curve and user friendliness. The interface also provides a lot a functionality through nesting operations in menus and logical separation using windowing. However having a copious amount of functionality usually creates a less usable environment as items are obfuscated into menus and sub-menus.

The main disadvantages of the WIMP interface have to do with input and navigating the interface. Using a keyboard and mouse is not always intuitive to users, as typing is a very laborious method of input and mice are relative positioning devices. In general, using a mouse to relatively control a pointer on a screen is not natural or intuitive. Usability is also sacrificed for functionality by packing as many features as possible into lists and menus, creating a complex work-flow. Myers notes that there is a certain level of stagnation that WIMP has caused, because WIMP has been relied on for so long, users and developers are hesitant to adopt new interfaces that are better suited for other input tools [17]. The biggest current problem of WIMP is its infectious nature into other interfaces, although it does provide some consistency. Since WIMP is so deeply ingrained into how we interact with computers, interaction designers often rely on old WIMP paradigms that do not really fit rather than innovating on more cohesive designs and interactions.

2.1.2 Touch

In the context of this paper, a touch interface is defined as an interface in which user input is a result of direct physical contact with a touch input device such as a touchscreen. The touchscreen has long existed in science fiction and has been a reality for a few decades now. The prevalence of smartphones and tablets in the consumer market has thrust the touch interface into the limelight. Ever since the iPhone took the commercial market by storm in 2007, the touch interface has become a driving force in user interface design. In Windows 8, Microsoft even redesigned Windows with a more touch-friendly user interface (UI). Since the original launch of the iPhone, other devices have joined in: the iPad, Android smartphones, Android tablets, Microsoft Windows Phone, Microsoft Surface, and BlackBerry touch-based phones. Touchscreen devices have become the standard user interface for mobile devices.

The touch interface allows for a compact system, as the display usually serves as the touch input, negating the necessity for peripherals like a mouse and keyboard [21]. The interface allows more precise user input than gesture interfaces, but can be less precise than their WIMP counterpart. Touch-gestures, like pinch-to-zoom, can also be leveraged to enhance the the user's experience but are not always intuitive. Another benefit is that most touch-based UIs follow a similar paradigm to WIMP interfaces, which allows for higher usability for users familiar with WIMP conventions.

The major disadvantage with most touch devices is that the touch screen is often covered in grease and grime from the user's fingers, causing a sub-optimal user experience [21]. The user's fingers also get in the way of the content on the screen and make it hard to accurately perform operations [21]. Touch-based interactions, such as pinch-to-zoom, are not always intuitive or easily discoverable. One of the most subtle problems with touch-interfaces is that designs tend to fall back on WIMP interactions rather than being innovative and creating novel interactions. This is one of the biggest problems with gesture interfaces as well, designs are often just retrofitted or evolved forms of WIMP, robbing these interactions of their potential.

2.1.3 Immersive/Gesture

In the context of this paper, an immersive/gesture interface is a system where user interaction is defined in an interaction space where the user uses in-air motion to provide input into a system [5]. Thus far, the mainstream market for gesture interface systems have mainly been in the video game industry with devices such as the Microsoft Kinect, the PlayStation Move [19], and the future Xtion PRO from ASUS [11]. There have been other devices to enter the foray of gesture interface devices, especially in the realm of personal computing. Devices such as the Leap Motion bring an in-air gesture based alternative to the primarily WIMP driven interactions found in personal computers [6].

The benefit of a gesture interface is that it tends to be more intuitive than other user interfaces as there is one less layer of abstraction [21]. The user does not need to manipulate an input device to interact with a system, instead they can just use their own body as the input. This gives the unique advantage of mimicking real world interactions. The user could now actually kick to virtually kick a soccer ball; rather than having to use a mouse and keyboard. This type of use case allows the interaction to be more intuitive to the user as the physical motion is easier to comprehend and utilize. Overall, the gesture interface allows for a wider and more dynamic range of input over traditional interfaces.

The gesture interface also has quite a few drawbacks. The main problem is that a majority of gesture based UIs try to fit interactions into a more traditional WIMP interface. Another drawback is most gesture interface systems currently lack precision, making the use of your body to navigate menus and settings an unpleasant user experience. In addition, users tend to become fatigued by the physical activity required by these systems [21].

2.1.4 Hybrid

In this paper, a hybrid interface is defined as a combination of two or more of the aforementioned interfaces. A hybrid interface maintains autonomy for each individual interface while also offering user interactions that utilize both interfaces in a complementary fashion. A few existing systems come close to being considered hybrid interfaces, but fall short to meet the aforementioned criteria because the interaction between interfaces is mainly uni-directional.

2.1.4.1 Related Systems

An example of a system that comes close to a hybrid interface is the Xbox 360/Xbox One and SmartGlass [10]. SmartGlass enables the active collaboration between a touch interface via a smartphone/tablet and an immersive interface via the Microsoft Kinect [10]. Microsofts system allows for unique user interactions, but falls short in creating a seamless user experience. Overall, the Xbox 360/Xbox One in concert with SmartGlass is the closest realization of the proposed hybrid system outlined in this paper, but does not provide true multi-directional interactions.

3 SYSTEM FRAMEWORK

This chapter details the overall system and a high-level overview of how it functions to create a natural user interface.

3.1 NUI

Steve Mann has been credited with the creation of the term Natural User Interface (NUI). Mann's describes a natural user interface in terms of what are the natural capabilities of the human body and intelligence when interacting with a system [15]. It is important to note that the natural property of a natural user interface is external to the interface itself, as *natural* refers to the ways the user interacts with the system and the general impression the user has when using it [21]. *Natural* relates directly to the user's feelings and experiences while using the interface, rather than any particular feature that interface has [21]. The goal in trying to make a natural user interface is to increase the usability and appeal of the interface, and decrease the learning curve needed to use the system, all while maintaining functionality. A natural user interface should feel intuitive and instinctive both physically and cognitively.

3.1.1 NUI Through Hybridization

The main goal of the framework is to achieve a NUI through the hybridization of multiple user interfaces when using multiple devices. As smartphones, tablets, and technology continue to be infused into our lives and become essential to everyday tasks, natural and intuitive interactions are crucial in decreasing cognitive load and increasing productivity and enjoyment of the user. The benefit of a hybrid interface is to take advantage of all the benefits and mitigate all the disadvantages of the coupled interfaces, enumerated in Chapter 2. The closer coupling of interfaces can be achieved by the systematic sharing of direct user input and a context aware framework for presenting relevant user interactions based on the user's current needs and wants. The context aware framework is described in further detail in the following section.

3.2 Context Aware

The purpose of creating a context aware framework is to lay a foundation to easily conceive and build unique and natural user interactions. The framework is designed to focus around the user and to be task oriented, building a context around what the user is currently doing and what they are trying to accomplish. A context is defined by a user's interaction spaces and the current state of the system. The context is then acted upon by a main executor, or arbiter, to decide how the current context influences how the user's input and the interfaces are directed.

3.2.1 Interaction Spaces

A context is defined by two interaction spaces, a physical interaction space and a mental interaction space, that are combined to create a pool of information to design interactions around.

3.2.1.1 Physical Interaction Space

Building off the idea of Microsoft's interaction space for the Kinect [5], a volumetric physical space where user interactions occur with the Kinect, a physical interaction space encompasses all physical properties of user interactions. Keeping the scope of physical interaction spaces incredibly broad forces designers to keep interactions very generic, allowing for easier future expansion and augmentations. The main physical properties can be broken up into three categories: environmental properties, direct user input, and indirect user input.

Environmental properties establish a setting in which interactions take place. Environmental properties include, but are not limited to: room topography, location, temperature, humidity, ambient lighting, and time. The environment helps establish a backdrop in which to interpret user input and frame interactions. Collecting and using these properties are not crucial in creating a physical context, but more information can lead to more fulfilling and applicable interactions. Knowing and applying these fine details can provide a richer and more immersive user experience.

Direct user input is the input the user intends to give to the system. One of the greatest problems when interacting with computers is interpreting a user's intentions, which usually causes user frustration in the form of false-positives or false-negatives. Mitigating frustrating user experiences such as bad voice to text, unrecognized gestures, or an accidental click on a touchpad, is paramount in creating an enjoyable user experience. Having sanitized direct user input is extremely important in creating a physical interaction space.

Indirect user input is input not intended to interact with the system, which can be gleaned from the user to enrich the user experience. The most important use of indirect input is to identify it and use it to improve proper recognition of direct user input. Examples of indirect input include, but are not limited to: number of users, user recognition, user body language, user attention, heart rate, and accidental input tendencies. Very similar to environmental properties, indirect input is not essential to a physical interaction space, but it can provide pivotal information to provide a truly natural user interface.

3.2.1.2 Mental Interaction Space

Just as a physical interaction space tries to interpret a user's physical interactions, the mental interaction space tries to interpret the users cognitive engagement. The hope is to wrangle some semblance of the user's cognitive processes, or train of thought, to offer predictive interactions that are natural and anticipatory. However, there is already enough difficulty interpreting explicit user physical interactions that interpreting a user's mental state seems improbable. In order to alleviate the complexity of trying to understand the human mind, the scope of a mental interaction space has been narrowed to a loose interpretation of a user's working memory.

In *Working models of working memory*, Barak defines working memory as "a system that maintains and manipulates information for several seconds during the planning and execution of many cognitive tasks" [13]. There are several competing models for how working memory functions, but one of the most prominent and widely cited models is Braddeley's model of working memory [12]. During the initial design of the framework, a very preliminary system was laid out for the context aware system that was later determined to have striking similarities to Braddeley's model. After more research, these commonalities were embraced and Braddeley's model was referenced in further designs.

Braddeley's model has an entity called the "central executive", which can be thought of as a person's cognitive director that guides attention and coordinates cognitive tasks [12]. The central/cognitive executive is essentially the voice in your head you use to dictate what you think and do. The concept of a central executive helps shape and mold the framework's own decision maker, designated the Arbiter, which will be discussed in more detail in the following section. Nomenclature was borrowed from the Norman and Shallice model for their cognitive scheduler for helping to queue cognitive tasks, the supervisory attentional system (SAS) [18]. The framework has a supervisory attentional sub-system (SASS), which acts not as a cognitive scheduler but as a task recommender. The arbiter and SASS work in combination to emulate the current working memory of the user based on the current task the user is trying to accomplish, in addition to any memory of previous user interactions. The mental interaction space is therefore just a collection of current and future tasks that can be used to derive unique and useful user interactions.

3.2.2 Arbiter

As discussed in the previous section, the framework has an arbiter that acts similar to Braddeley's central executive. The arbiter is in charge of receiving, scheduling, and general handling of tasks. Tasks being very similar to cognitive tasks that are a part of a person's working memory, these tasks are short-lived and only one of them can be executed at a given time. The arbiter is also in charge of distributing input to the correct interface based on the given context. The arbiter is the main component that drives the context aware framework.

In general, people are more susceptible to distraction and mistakes while multitasking [16]. To create a more natural and user friendly experience, the framework is designed to efficiently handle and tailor user interactions around a single task at time. A single task based framework does note overwhelm the user and allows for intuitive interactions. Unlike traditional schedulers, which are often used to give equal time to multiple processes to run, the arbiter prioritizes the single most important task. The SASS suggests a task from an interface and the arbiter prioritizes the tasks based on the current context, and launches reflexes (auxiliary functions) that the other interface may or may not help in fulfilling.

The other unique part of the framework is that the arbiter, based on the current context, directs user input to the appropriate interface. When a SASS presents a task, and it is accepted, the task will request certain types of input if available. Funnelling user input allows the arbiter to override an interface's default behavior and expand the types of user interactions that would otherwise not be possible.

4 IMPLEMENTATION

In this implementation, functionality will be limited to entertainment in the context of a traditional living room setting. The system consists of a multi-touch coffee table, providing the touch interface, paired with a Microsoft next-generation Kinect, providing an immersive/gesture interface. The multi-touch table resides in the Kinect's interaction space, contributing to the physical context. In this particular interaction context, input comes from both the table and the Kinect, while output comes from the table and a television. See Figure 4.1 for the general relationship between the arbiter and the interfaces.

Input comes from both a touch and gesture interface. Through mixing these modalities, passive input (observed user behavior) can be used to create a context in which active input can be interpreted in a more meaningful manner. In this system, the Kinect would observe what the user is doing and relay that information to the multi-touch table that could be used to augment the touch input. The 3-D data provided allows for a spatial natural user interface [21], giving all the user's 3-D movement meaning in this hybrid interface. This context aware system has a distinct advantage in that the user does not have to actively engage the system to perform a task, creating a more immersive user experience. The potential of such a system allows it to be a more anticipatory than reactive interface.

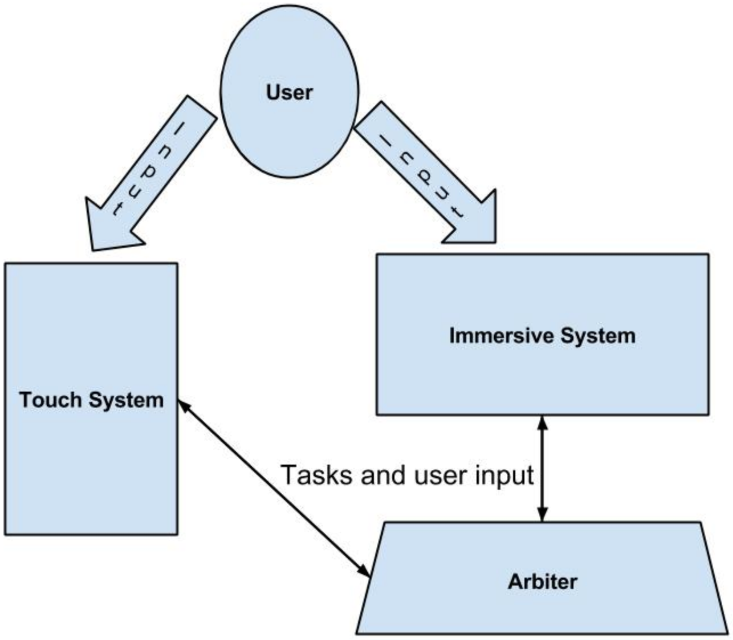

Figure 4.1: Framework overview.

To illustrate proactive interactions, when a user leaves the room, the system sees the user leaving, pauses their current activity, and signs them out. Conversely, the system could also see a user enter the room, perform some biometrics and user recognition, sign the user into the system, and resume their last activity. In both of these use cases the user does not actively engage the system, yet the system performs useful operations anticipating what the user would want. Another situation is that the user navigates through their movie collection on the multi-touch table, then does a throw gesture to put it up on the screen; where playback can be done with in-air gestures or on the table. While watching the movie, the table presents relevant information about the movie, anticipating the potential desire for that information. The system could even pause the movie if it sees the user has fallen asleep, or it could even try to wake the user. These examples are just a few of many potential applications of this interface which deviate from the traditional WIMP conventions.

The overarching goal of this hybrid interface, and any interface for that matter, is to provide a user interaction that is intuitive and makes the method of interaction as unnoticeable as possible to the user. The user should not have to think about how to perform an action within a system [21], the action should happen as if it was a continuation of their body. This is what a more natural user interface should do, and this hybrid interface can provide the means to do it.

4.1 Multi-Touch Table

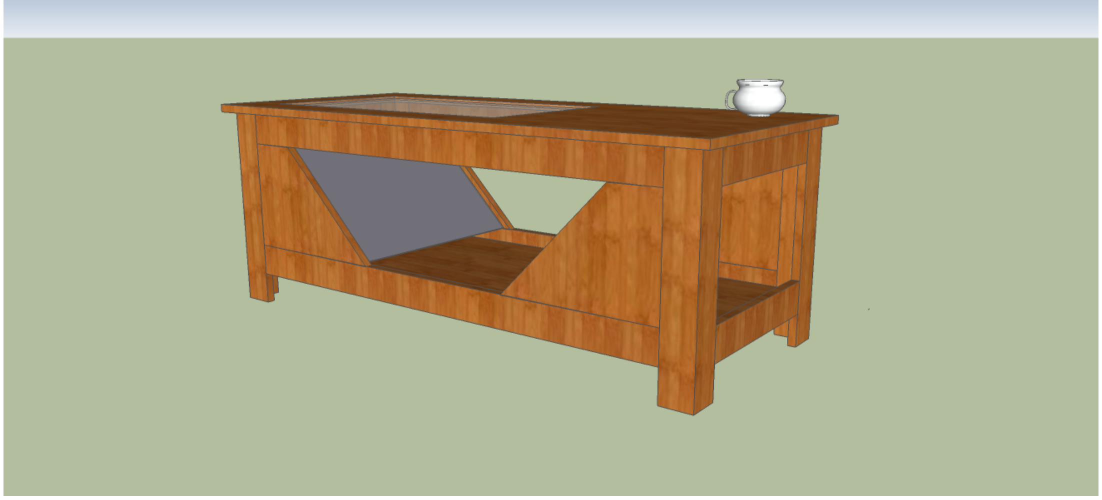

Figure 4.2: Table concept designed by Troy Weber.

A multi-touch coffee table was used as the touch interface because it provided a large surface to interact with, and stayed stationary in the interaction space. A do-it-yourself guide on instructables.com was used as a reference for acquiring the required components and provide the general concepts to create a multi-touch

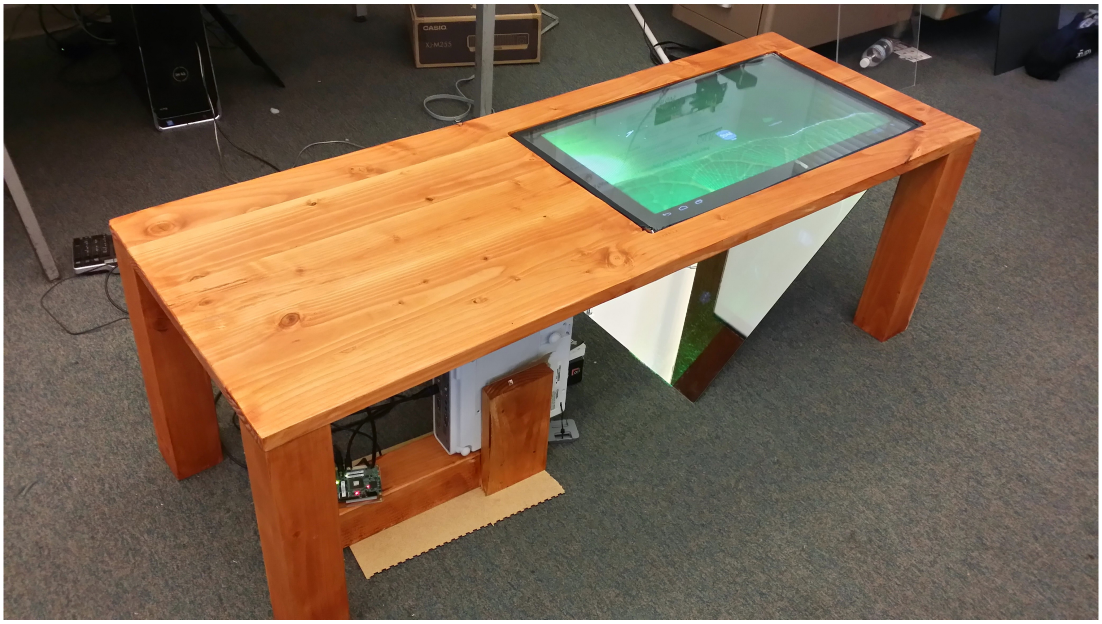

Figure 4.3: Constructed coffee table.

surface [3]. Figure 4.2 shows the initial concept of the coffee table and Figure 4.3 shows the constructed table. A simpler design was implemented based on time constraints.

The table is made of dogwood and coated with a golden pecan stain. The screen consists of a rear-projection piece of acrylic with an EndLighten piece of acrylic that has a string of 850nm infrared (IR) light-emitting diodes (LEDs) affixed to the border of it. The LEDs flood the EndLighten acrylic with IR and uses frustrated total internal reflection (FTIR) to highlight touches. A PlayStation Eye Camera [7] was fitted with an 850nm band-pass filter and was used to view the touches on the table as seen in Figure 4.4. The image was then processed on a server using the open source digital image processing (DIP) library reacTIVision [8]. The DIP library performed thresholding, blob detection, and finger tracking on the image, as seen in Figure 4.4, and provided raw finger coordinates which were passed to the arbiter on the server for processing.

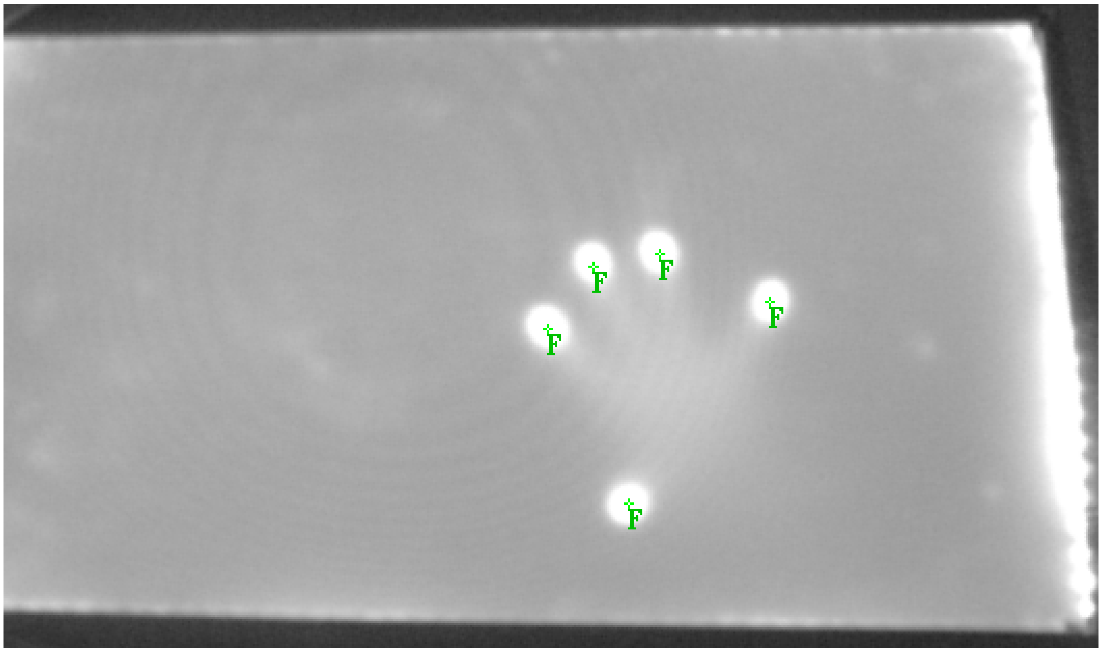

Figure 4.4: Visible FTIR touches with blob detection and finger tracking.

The coffee table runs a custom version of Android based on version 4.1.2. Qualcomm generously donated a DragonBoard, more specifically a Snapdragon 600 APQ8064-Based IFC6410 development board, for academic purposes. The DragonBoard has an HDMI out that is connected to a Casio XJ-M255 projector, which projects onto the rear-projection acrylic after being reflected off a mirror at 45 degrees in respect to the ground. No image manipulation was required. The mirror flipped the image so it was in the correct orientation for rear-projection, and the position of the projector allowed for a perfectly aligned projection. A background service runs on the coffee table to connect to the arbiter, allowing for tasks and user input to be exchanged between the arbiter and Android over a TCP network connection. When user input is received via touch input or gestures, the background service injects touch events into the Android framework using the Instrumentation test class [4].

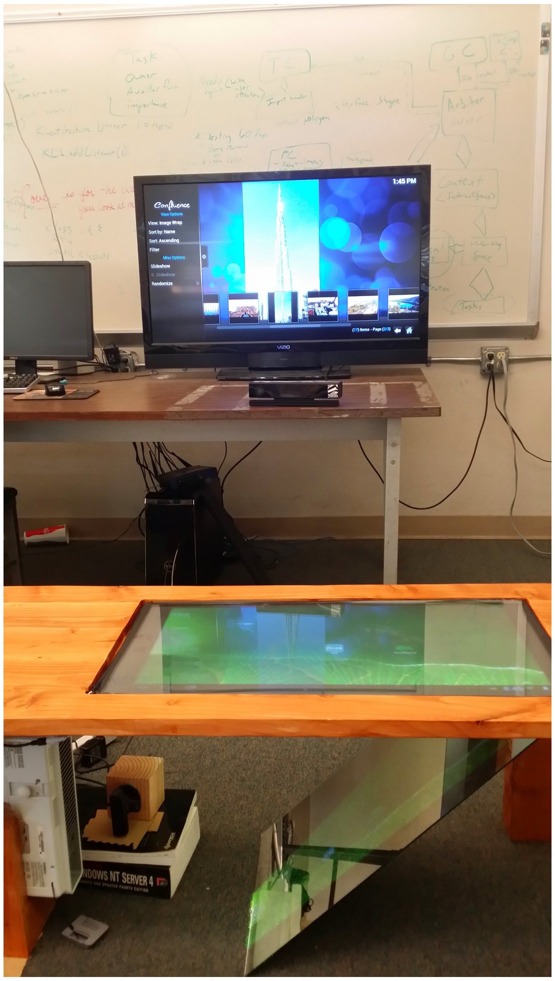

Figure 4.5: Finished system.

4.2 Multimedia Center

The multimedia center is the immersive/gesture interface in the implementation, running XBMC [9] (Figure 4.6) and equipped with the next-generation Kinect. XBMC is an open source multimedia center application that allows for easy browsing and viewing of movies, pictures, and music. As of this writing, the next-generation Kinect is only available in conjunction with an Xbox One and does not have support for the PC. Microsoft granted us early access for their "Kinect For Windows" product, which provided prototype hardware and access

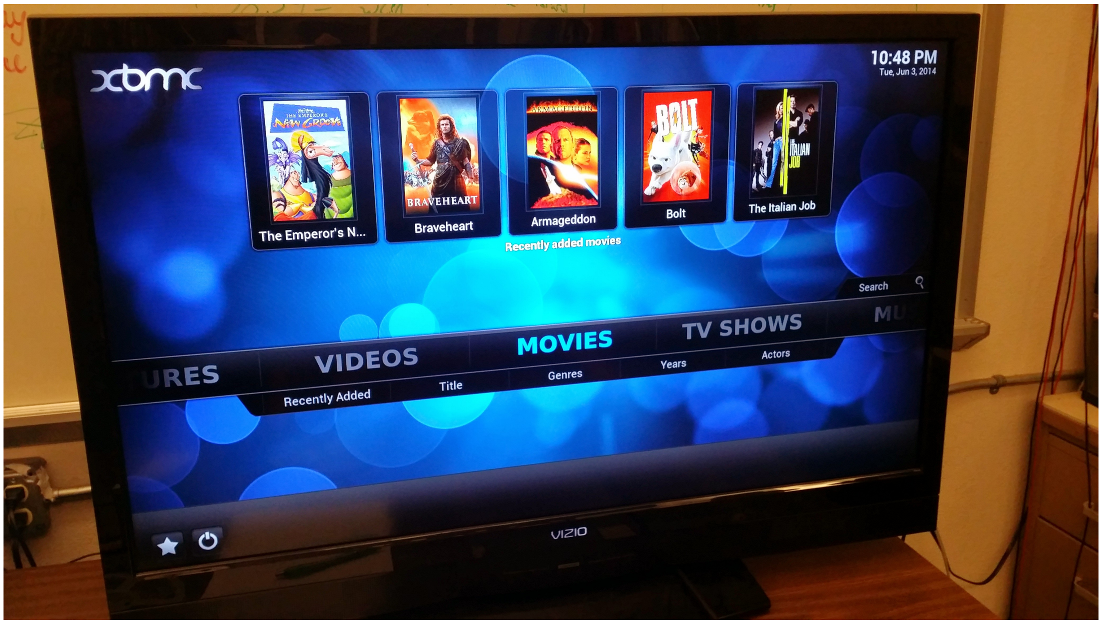

Figure 4.6: XBMC media center main menu.

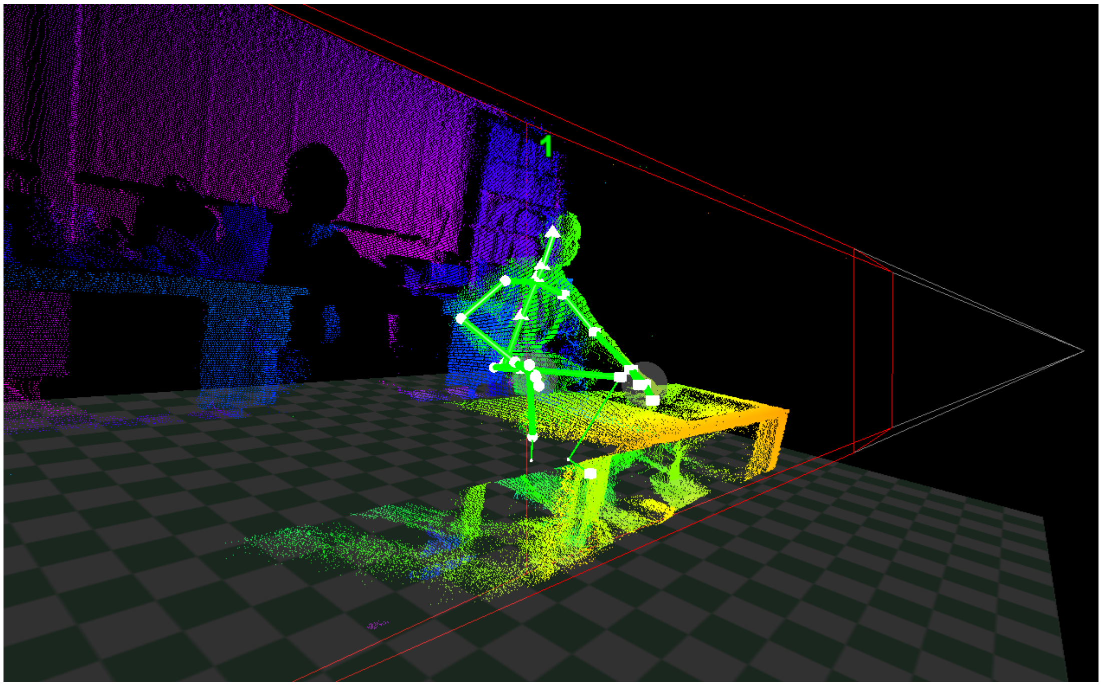

Figure 4.7: Kinect interaction space.

to their SDK. Andrew Weinfield implemented a gesture recognition library for this project, specifically optimized for a seated user. Weinfield leveraged the very limited alpha Kinect SDK to create a wide gesture set based on skeletal tracking. Figure 4.7 shows the Kinect's interaction space and the skeletal tracking it provides. The library supports some limited user tracking relative to the table. The current gestures supported are: left and right hand swipes and flings, attention tracking, some single handed finger gestures, and a grab and drag gesture. The gesture interface registers with the arbiter and feeds all user gestures to the arbiter for processing. The arbiter dispatches gestures and other input to the appropriate interface for use depending on the current context.

4.3 Arbiter

The current implementation of the framework has a server client model for communication between the interfaces and the arbiter. The arbiter is part of the main program that starts a TCP server which the interfaces connect to as clients, and is designed to allow for the dynamic adding and dropping of multiple interfaces, though its current state puts a hard-coded cap at two. The gesture and touch interfaces, XBMC and the Android background service respectively, connect to the server and register with the arbiter.

After the initial connection, each interface acts independently of the other until the user initiates a task on an interface. When the user tries to launch a task, the interface's SASS contacts the arbiter and asks for permission to run the task, which the arbiter can either grant or deny permission based on the current context. If the arbiter declines the SASS request, that task is immediately stopped. If the arbiter grants the request, any other current running tasks are stopped by the arbiter and the newly accepted task is started. The task contains a set of properties, such as: an id, a currently running flag, a reflex URI, and any other information the interface may provide for the context of that task.

Once the task is finally running, the arbiter redirects user input based on that task's current preferences and sends the task's reflex URI to the other interfaces. The reflex URI is basically a request to the other interfaces to present auxiliary content and functions to the user to support the content provided in the URI. The term reflex was used to denote the natural flow the secondary interface should emulate when presenting auxiliary content or functionality. The reflex system and the arbiter are the core pieces of the framework that allow for natural and novel user interactions.

5 LIMITATIONS

The main objective was to create a framework that allowed for a collaborative environment between interfaces. The framework was then tested by creating a fixed set of demonstrable user interactions on prototype hardware to test the viability of the framework. The first iteration of the framework and the implemented system have limited functionality.

5.1 Input

5.1.1 Touchscreen Limitations

In order to expedite implementing multi-touch image processing, a third-party library was used on the server, which forced touch input to by sent over the network which increased touch latency. Also, touch input enters the Android framework through Android's testing Instrumentation class [4], which also increases latency and prevents touch injections over system UI elements. The lack of touch on system UI prevents the user from touching anything in the notification area or the navigation bar (the home, back, and recents buttons). A keyboard was added to the current setup to provide the back and home functionality. Also, since the touchscreen uses frustrated total internal reflection (FTIR) and digital image processing, performance is already subpar when compared to modern touchscreen technology.

5.1.2 Gesture Limitations

The limitations of the gesture input are a direct reflection of the available tools. The current Kinect for Windows SDK is still in a pre-release state, which shows when considering available functionality and general robustness and reliability of the incoming data. The gesture library was able to somewhat accurately detect user attention, which was implemented using a workaround because certain functionality outlined in the SDK's documentation was missing altogether. The adolescent SDK was not the only problem. Being part of the early access program meant prototype devices were issued. The prototype Kinect was not always reliable and would crash occasionally, or enter a state where a restart or reinitialization was needed in order to function properly.

5.2 Arbiter

In the current implementation, the arbiter task scheduler always chooses the newest task as the task to run, which is not always the desired use case. There is also no pausing and resuming of tasks. Once a task is started, the user must restart that task to run it again. User input redirection is also limited since tasks can only request all input or none, and cannot request a specific list of input. Arbiter communication to the interfaces can also be sped up. The communication protocol uses TCP and is not optimized, leading to higher latency for user input which negatively impacts the user experience.

6 VALIDATION FRAMEWORK

Creating a validation framework was a challenge. The overall project had a very broad scope and vague success metrics, forcing the creation of narrower success criteria to judge the project. The main objective was to create a natural user interface (NUI) by combining multiple interfaces into a smart and seamless experience. The main points that drive the validation framework come from Wigdor and Wixon's *A Brave NUI World*, when describing the must haves in the design guidelines of a NUI. [21]

- *Create an experience that, for expert users, can feel like an extension of their body.*
- *Create an experience that feels just as natural to a novice as it does to an expert user.*
- *Avoid falling into the trap of copying existing user interface paradigms.*

6.1 User Experience Experiment

The novel nature of the system does not make direct comparisons between the proposed system and existing systems a viable validation path. Instead a usability experiment was conducted to compare the usability between novice and proficient users. The success on the project is contingent on the human computer interaction (HCI) aspect of the system. To validate the system an experiment was conducted around the tenants of Wigdor and Wixon's NUI design guidelines.

The experiment was arranged in the following manner:

- A group of volunteers were divided into novice and proficient user groups based on self-evaluations and perceived technical affinity. Novice and proficient users were given the same instructions and limited training with the system. See Appendix B for the surveys issued.
- Usability metrics were defined based on an initial and follow up survey to help determine the user's frustration and their perceived usability of the system.
- After the experiment, that data was analyzed to validate the system. First and foremost, the system was evaluated to determine if the system is natural and intuitive to the majority of the user base. If not, the system fails validation. Next, if the system passes the prior validation and the difference in usability between user groups is minimal, the system is considered a Natural User Interface, as defined by Wigdor and Wixon's criteria.

6.1.1 Usability Test

The usability test started by having the user sign an informed consent form and fill out an initial survey (see Appendix A for the informed consent form and Appendix B.1 for the initial survey). The user was then given information on how the system was designed and what problem it was trying to solve. They were also instructed on the gesture set available to them and associated operations. The user was then guided through a series of interactions to test the usability of the system.

The first interaction was to view a slide show on XBMC. The associated reflex was to have the Android coffee table present a map of where that picture was taken. The geographic information was read from the image's EXIF data and sent to Android, where Android was able to provide auxiliary content through the Google Maps application. The user would use swipe gestures to navigate through the slide show, and the Maps application would pan to the appropriate location. The next interaction was to launch 2048, a puzzle game, on the coffee table. The reflex started some background music, and a simple hand gesture allowed the user to skip tracks. Both gesture and touch input were used to interact with the game. The final interaction had the user navigate to a list of movies in XBMC and play one. The reflex for that task was to bring up information for that movie on the coffee table. The user was then tasked to stop the video using a grab and hold gesture to control a mouse on screen and select stop.

After all these interactions were completed a follow-up survey was given to the user to gauge the usability of the system and framework (see Appendix B.2 for the follow-up survey). Afterward, the user was also able to ask questions or provide any additional feedback they might have.

6.2 Novel Metric

The proposed system was also judged on originality. Existing user interface paradigms should be thrown out, and only considered when absolutely necessary. The system should not be based on old user interface paradigms, but have innovative user interactions that are inspired by the system. The idea is not to innovate for the sake of innovation, but try to create a pure natural user interface and weed out old paradigms that are continually used based on tradition.

7 FINDINGS

A usability test was given to two groups of users in order to validate the system as a viable NUI framework. The two groups consisted of proficient/technical users and novice/average users. The technical users were computer science students at Cal Poly San Luis Obispo, and the average users were non-technical related majors at Cal Poly. The user's self-evaluation of their technical savviness, see Table 7.1, closely matched the particular group they were placed in. The 21.4% (3 users) of the novice users that rated themselves as tech savvy, were evaluated to be competent but not enough to boost them into the proficient/technical user group.

	Proficient User	**Novice User**
Considered themselves tech savvy	100%	21.4%
Others consider them tech savvy	94.4%	21.4%

Table 7.1: Self evaluations.

Strongly Disagree - Strongly Agree (1-5)	**Proficient User**	**Novice User**
Usefulness (Category Average)	3.52	3.85
Ease of use (Category Average)	3.41	3.76
It is simple to use.	*4.00*	*4.21*
Ease of learning (Category Average)	4.21	4.30
Satisfaction (Category Average)	3.66	4.11
It is fun to use.	*4.33*	*4.50*

Table 7.2: Follow-up category results.

The averages of the two user group's follow-up surveys are shown in Table

7.3. A brief evaluation of each category in the survey allows for a quick view of how the usability test went (see Table 7.2). In general both groups were neutral to somewhat agreeable as to the usefulness of the system. This was expected since the demo was a fixed list of entertainment interactions and the users were not given any examples of how the system could increase productivity or meet a direct need. The "ease of use" category received the lowest average score of all categories. This was caused by the amount of effort and frustration the users had in getting gestures to work. However, the statement "It is simple to use" received above average agreement among both groups, which helps reaffirm that it was the gesture frustration that lead to lower marks in that category. The "ease of learning" category received above average agreement among both groups, which is indicative of a natural and intuitive user interface. The final category "Satisfaction" earned neutral to slightly favorable marks, but the general consensus during testing is that the system was fun to use, which bodes well for an interface and interactions designed for entertainment.

The surprising result from the data is that the novice user group almost always had a higher level of agreement for each statement than that of the proficient group. It is surprising because the technical nature of the interface would suggest that the more technical user would have a higher appreciation or would be more apt to see the potential value it has to offer. The fact that the opposite occurred suggests that the more technically exposed the user was, the more critical they were of the system. The general trend could also imply the less technically experienced users have a easier time adapting to new interfaces.

In order to validate the success of the proposed framework, the usability between these user groups was used to judge whether the system could be consider a natural user interface. In all the statements in the follow-up survey, only 7 of the 30 statements had an assessment difference greater than 10% between the groups. In general, there were very few extremes, and the data between the groups was similar enough to say that the groups were in agreement, which passes the validation framework for a NUI. The high marks for “ease of learning” are also indicative that the interface was natural and intuitive.

It is also important to note that the implemented system passed the “novel metric” of the validation framework. User interactions were designed to be unique and tailored to the user interactions. Existing user interface paradigms were avoided when they did not directly fit into the designed user interactions. The sample hybrid interface allowed for the creation of unique and novel user experiences through natural and intuitive interactions.

Strongly Disagree - Strongly Agree (1-5)	**Proficient User**	**Novice User**
Usefulness (Category Average)	**3.52**	**3.85**
It helps me be more effective	3.44	3.93
It helps me be more productive	3.17	3.71
It is useful	4.17	4.29
It gives me more control over the activities…	3.61	3.79
It makes the things I want to accomplish easier…	3.44	3.71
It saves me time when I use it.	3.00	3.36
It meets my needs.	3.61	3.93
It does everything I would expect it to do.	3.72	4.07
Ease of use (Category Average)	**3.41**	**3.76**
It is easy to use.	3.39	4.14
It is simple to use.	4.00	4.21
It is user friendly.	3.67	4.14
It requires the fewest steps possible…	3.72	3.86
It is flexible	3.33	3.50
Using it is effortless.	2.50	3.21
I can use it without written instructions.	2.94	2.93
I don't notice any inconsistencies as I use it.	3.39	3.43
Both occasional and regular users would like it.	3.89	4.00
I can recover from mistakes quickly and easily.	3.94	4.21
I can use it successfully every time.	2.72	3.71
Ease of learning (Category Average)	**4.21**	**4.30**
I learned to use it quickly.	4.33	4.43
I easily remember how to use it.	4.33	4.43
It is easy to learn to use it.	4.39	4.29
I quickly became skillful with it.	3.78	4.07
Satisfaction (Category Average)	**3.66**	**4.11**
I am satisfied with it.	3.83	4.21
I would recommend it to a friend.	3.72	4.36
It is fun to use.	4.33	4.50
It works the way I want it to work.	3.50	4.07
It is wonderful.	3.72	4.21
I feel I need to have it.	2.61	3.21
It is pleasant to use.	3.89	4.21

Table 7.3: Follow-up survey results.

8 FUTURE WORK

There is a lot of room to improve and expand on the current system. The first step would be in making the current implementation more robust physically and continue to improve the input framework. The coffee table can be further improved to make setup and calibration easier. The coffee touchscreen can also be improved by having the table's main processing unit do the DIP and finger tracking, which would minimize latency since not every touch would have to be sent over the network. Revamping the current gesture set would also be recommended once the Kinect SDK fully matures and a more stable commercial device is available. The overall network communication can also be restructured to minimize latency as well, and refine the current protocol to be more efficient.

The main framework can also be improved. Tasks can be handled more efficiently and effectively. Pausing and resuming tasks can also be added, to provide even more fulfilling user interactions by not forcing the user to always restart their last task if they were interrupted. The biggest area for possible expansion and further research is the capabilities of the arbiter. The arbiter's task management and scheduling can be improved by applying machine learning or artificial intelligence. The current implementation can easily be expanded to include other devices and interfaces, allowing for further exploration and research of hybrid interfaces in a context aware environment.

9 CONCLUDING THOUGHTS

As more and more devices become available to the average user, the interactions between those devices and the user become more important. Multi-device interactions that are more meaningful, and less cognitively taxing on the user, will be a driving force in HCI and user interaction design. The goal of this thesis was to provide a solution through a context aware hybrid interface, and build a framework to demonstrate the viability of such a concept. A validation framework was created in order to evaluate the usability and success of the system. The system created a context based on physical and cognitive models, allowing for the creation of novel user interactions. The implementation was then tested and found to be a feasible and applicable solution.

The framework also pushed for user interactions that are not dependent on existing user interface conventions. Interactions with interfaces should be natural and intuitive, and appropriately designed for specific applications. All interfaces should aspire to be natural user interfaces, and should be a driving force in HCI design. As the complexity of devices increases, so can the complexity of how we interact with them. Interaction design must continually evolve and adapt to the changes in technology, carving out new user experiences and break the mold of old conventions. The hope is that this project and research shows the possibilities offered through a context aware hybrid interface, and provides a solid foundation to build off of for future research and development.

REFERENCES

[1] Apple TV. www.apple.com/appletv/. Accessed: 2014-06-1.

[2] Google Chromecast. http://www.google.com/chromecast. Accessed: 2014-06-1.

[3] How to build a Multi-Touch surface. http://www.instructables.com/id/How-to-build-a-Multi-Touch-surface/. Accessed: 2014-06-3.

[4] Instrumentation. http://developer.android.com/reference/android/app/Instrumentation.html. Accessed: 2014-06-3.

[5] Interaction Space. http://msdn.microsoft.com/en-us/library/hh973071.aspx. Accessed: 2014-06-3.

[6] Leap Motion. https://www.leapmotion.com/. Accessed: 2014-06-3.

[7] PlayStation Eye Camera. http://us.playstation.com/ps3/accessories/playstation-eye-camera-ps3.html. Accessed: 2014-06-3.

[8] reacTIVision. http://reactivision.sourceforge.net/. Accessed: 2014-06-3.

[9] XBMC. http://xbmc.org/. Accessed: 2014-06-3.

[10] Xbox SmartGlass. `http://www.xbox.com/en-US/smartglass`. Accessed: 2014-06-3.

[11] Xtion PRO. `http://www.asus.com/Multimedia/Xtion_PRO/`. Accessed: 2014-06-3.

[12] A. Baddeley. Working memory. *Science*, 255(5044):556–559, 1992.

[13] O. Barak and M. Tsodyks. Working models of working memory. *Current Opinion in Neurobiology*, 25(0):20 – 24, 2014. Theoretical and computational neuroscience.

[14] A. Dix. *Human-computer interaction.* Springer, 2009.

[15] S. Mann. *Intelligent Image Processing.* John Wiley & Sons, Inc., New York, NY, USA, 2001.

[16] D. C. McFarlane and K. A. Latorella. The Scope and Importance of Human Interruption in Human-Computer Interaction Design. *Human-Computer Interaction*, 17(1):1 – 61, 2002.

[17] B. Myers, S. E. Hudson, and R. Pausch. Past, Present and Future of User Interface Software Tools. *ACM TRANSACTIONS ON COMPUTER-HUMAN INTERACTION*, 7:3–28, 2000.

[18] T. Shallice and P. Burgess. Higher-order cognitive impairments and frontal lobe lesions in man. *Frontal lobe function and dysfunction*, pages 125–138, 1991.

[19] L. F. Teófilo, P. A. Nogueira, and P. B. Silva. GEMINI: A Generic Multi-Modal Natural Interface Framework for Videogames. *CoRR*, abs/1302.3862, 2013.

[20] A. van Dam. Post-WIMP User Interfaces. *Commun. ACM*, 40(2):63–67, Feb. 1997.

[21] D. Wigdor and D. Wixon. *Brave NUI World: Designing Natural User Interfaces for Touch and Gesture.* Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 1st edition, 2011.

A INFORMED CONSENT FORM

INFORMED CONSENT TO PARTICIPATE IN A RESEARCH PROJECT, 'Usability and Human Computer Interaction'

A research project on the usability and human computer interaction of a novel system is being conducted by Tony Lenz and Andrew Weinfield, students in the Department of Electrical Engineering at Cal Poly, San Luis Obispo, under the guidance of Dr.Lynne Slivovsky. The novel system incorporates a touch screen coffee table in junction with a TV media system equipped with gesture recognition technology.

You are being asked to take part in a study by completing two questionnaires and by performing a set of tasks using the system while being recorded. Your participation will take approximately 30 minutes. Please be aware that you are not required to participate in this research and you may discontinue your participation at any time without penalty. You also do not have to answer any questions you choose not to answer. The possible risks associated with participation in this study include eye, joint, and arm fatigue, as you will be looking at two monitors and interacting with the system through touch and gestures. If you should experience any discomfort requiring medical attention, please be aware that you can contact or visit Cal Poly Health center in building 27, (805) 756-1211.

Your confidentiality will be protected. We will not cite your name in any reports of this research without your permission, though a 3-dimensional silhouette recording or image may be used in future publications and presentations. Any published image of you will have your face obfuscated. Potential benefits associated with the study include possible refreshments and contributing to the research and design of future interactive systems.

If you have questions regarding this study or would like to be informed of the results when the study is completed, please feel free to contact Tony Lenz (alenz@calpoly.edu). If you have concerns regarding the manner in which the study is conducted, you may contact Dr. Steve Davis, Chair of the Cal Poly Human Subjects Committed, at (805) 756-2754, sdavis@calpoly.edu, or Dr. Dean Wendt, Interim Dean of Research, at (805) 756-1508, dwendt@calpoly.edu.

If you agree to voluntarily participate in this research project as described, please indicated your agreement by signing below. Please keep one copy of this form for your reference, and thank you for your participation in this research.

Signature of Volunteer Date

Signature of Researcher Date

B SURVEY

B.1 Initial Survey

Background (Circle One)

I own a tablet (iPad/Android/Windows/etc.) Yes / No

I own a laptop Yes / No

I own a smart phone (iPhone/Android/Windows/etc.) Yes / No

I own an eReader (Kindle/Nook/etc.) Yes / No

I am in general very tech savvy Yes / No

Others consider me to be very tech savvy Yes / No

Experience (Rate 1-10, never used it up to complete expert)

XBMC:

Android:

Motion tracking (Microsoft Kinect/Leap Motion/etc.):

Windows operating system:

Touchscreen tables (Windows PixelSense/etc.):

B.2 Follow-up Survey

Please rate your agreement 1-5

Strongly Disagree to Strongly Agree

Usefulness

It helps me be more effective:

It helps me be more productive:

It is useful:

It gives me more control over the activities in my life:

It makes the things I want to accomplish easier to get done:

It saves me time when I use it:

It meets my needs:

It does everything I would expect it to do:

Ease of Use

It is easy to use:

It is simple to use:

It is user friendly:

It requires the fewest steps possible to accomplish what I want to do with it:

It is flexible:

Using it is effortless:

I can use it without written instructions:

I don’t notice any inconsistencies as I use it:

Both occasional and regular users would like it:

I can recover from mistakes quickly and easily:

I can use it successfully every time:

Ease of Learning

I learned to use it quickly:

I easily remember how to use it:

It is easy to learn to use it:

I quickly became skillful with it:

Satisfaction

I am satisfied with it:

I would recommend it to a friend:

It is fun to use:

It works the way I want it to work:

It is wonderful:

I feel I need to have it:

It is pleasant to use: