ABSTRACT

SIEGLER, BRADLEY MICHAEL. Supporting Electronic CRC Card Sessions with Natural Interaction. (Under the direction of Robert St. Amant.)

This paper is an exploration into the potential of a new collaborative tool called the DiamondTouch. An application was developed to support CRC Card Sessions and tested with different users. As part of this research, experiment participants created software designs, both using and not using the software, where metrics were gathered about their performance. The results are examined both in a qualitatively and quantitatively manner for insight into usability of the system. With these results, an evaluation is made on the hardware and the software. Suggestions are made about the future of this application and can provide guidance for developing other collaborative applications on the DiamondTouch.

Supporting Electronic CRC Card Sessions with Natural Interaction

By
Bradley Michael Siegler

A thesis submitted to the Graduate Faculty of
North Carolina State University
in partial fulfillment of the
requirements for the Degree of
Master of Science

Computer Science

Raleigh

2004

APPROVED BY:

________________________________ ________________________________

____
Chair of Advisory Committee

DEDICATION

I would like to dedicate this work to my Parents who have always been supportive and my loving girlfriend for putting up with me through this whole process.

BIOGRAPHY

Bradley Michael Siegler was born in Topeka Kansas to parents W. Michael Siegler and Carolyn Siegler on February 19, 1981. He attended elementary school in Topeka before moving to Fayetteville Arkansas. Brad completed his education in Fayetteville and graduated from Fayetteville High in 1999. After completing High School, He enrolled in the Computer Science program and North Carolina State University, located in Raleigh North Carolina. He received his undergraduate degree in Computer Science form North Carolina State in December of 2002. Brad continued his education at NCSU by entering into the graduate school to receive a Masters of Science for Computer Science. While in graduate school, he studied Human Computer Interactions. Upon graduation, he will enter into the workforce as a software engineer. He plans on returning to academics to receive an MBA and potentially a PhD.

ACKNOWLEDGEMENTS

I would like to sincere thank all those who helped me conduct this research and to create this body of work. I would specifically like to thank the following:

Dr. Robert St. Amant for his guidance in conducting the research and during the writing of this document.

Amy Poling for her work building a knowledge base of related research.

Colin Butler and Thomas Horton for providing feedback and taking part in the preliminary user test.

MERL for providing a DiamondTouch to the Knowledge Discovery Lab to be used for this and other research.

All the Experiment participants for their willingness to take part in this research.

Without the help of all these people, this research and experiment would not have been possible.

TABLE OF CONTENTS

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

1. INTRODUCTION

Hardware has played a pivotal role in the increase of computer functionality. These hardware improvements have made computers faster, with the ability to do more simultaneously. Most of these hardware improvements have been internal, but arguably the most important improvement has been external: the display. The display revolutionized the role of the computer from being a backend tool to a front-end tool. No longer do users have to be programmers or computer experts. The display opened the doors to mainstream use.

No matter how pivotal the monitor was to the evolution of the computer, it alone was not enough to change the landscape of computing. The display afforded new interactions, but could not deliver these new interaction techniques without a new framework and software that took full advantage of its new functionality. A framework is a conceptual idea about how users will interact with computers, i.e. inputting information and receiving feedback. The frameworks of the time provided feedback in the form text, but did so only by printed output. These frameworks offered to no room for the addition of a monitor.

The first real framework to emerge that utilized the display was still textual based. This framework was not actually new; it was a modification of an existing framework, just with a change in the output device. In this framework the display simply replaced the role of the printer. Though these frameworks did not take full advantage of all that a display could offer, it was an improvement. Now computer users could receive more instantaneous feedback about the current state of the system and the programs running.

This instantaneous feedback, combined with the improvements in hardware, such as the hard disk and system memory, allowed for computers to become an interactive tool as opposed to a machine whose main purpose was batch operations. This shift in purpose from batch operations caused a shift in the skill set needed to use and run a computer. Previously, the programmers were experts in writing the code, but were not experts in running the code on the machines. This required another technician to load the programs into the machine to be run and to gather the output from the printer, which would be returned to the programmer. There was wasted time for the programmer in this process because they would be forced to wait till the program finished executing. Often time was also spent waiting for the program to just be loaded into the system. In today's world, machine time is cheap and developer time is expensive. This was not always the case. As this shift in skill need use a computer progressed, programmers found themselves taking on more important roles while the technicians started fading away. Even though the programmers were increasingly loading and executing their own programs, computer were still very complicated and still required an expert.

Though the computers were still hard to use and extremely expensive, the groundwork was being laid computers to become more accessible to the actual users.

The next framework to utilize the monitor evolved out of the textual framework that preceded it, but was influenced by the saying "A picture is worth a thousand words." This framework was based on the idea of Shneiderman's Direct Manipulation. This framework was largely graphical in nature, but still had the same support for textual based applications embedded in it. This framework offered the opportunities for evolving interfaces than had been previously available.

Under the influence of this framework, the WIMP (Windows Icons Menus Pointer) interface was developed. This interface design was credited to Douglas Englebart, but Alan Kay and Adele Goldberg designed most of the other elements that we associate with WIMP interfaces: bit-mapped display with movable/resizable windows, buttons, pop-up menus, and the desktop metaphor. This interface was first implemented by Xerox on the STAR workstation, but manifested itself in Apple's Lisa computer and first commercially available on the Macintosh [8].

One of the biggest problems facing computers was their difficulty in use. As stated before, the programmers initially were not skilled enough to operate the machine. This changed with textual based user interfaces, but were still difficult to use, especially for someone who was not an experienced computer user. This problem can best be identified as the failure in communication between the computer and the user. This may seem like an odd statement, but the user could not understand what the computer was displaying, nor could the user direct the computer to perform some functions. The solution to this problem is establishing a language that both the computer and the user understood.

Rather than try to teach each user a common language, it was easier to program the computer to communicate in such a way that every user could understand and communicate back.

This standard communication is embodied in the idea of a metaphor. When a metaphor is applied to an interface, physical objects from the real world are represented as electronic objects in the computer, thus a metaphor. The real world items that are used in the metaphor should conceptually the same as the digital items they represent. The uses and operations of the item should line up fairly well between the physical and digital objects.

Though the WIMP interface does not explicitly call for a metaphor to be used, most interfaces use the standard metaphor of the desktop. Almost anyone's desktop lends itself well to represent the workspace in computer. On this desk, folders and files exists. These folders can contain files or even other folders. This matches conceptually the directory structure that is used to organize a computer's file systems. By using the desktop metaphor, every user who uses a desk can understand what is being represented in the computer. Now, a language exists that most users can understand and use to communicate with a computer.

As with the textual framework, the graphical interface made it possible for the computer to gain more widespread use acceptance. This was largely due to the WIMP interface design principles. The graphical framework, specifically the WIMP interface, was better not because it could do more than the textual framework, it rather offered a richer vocabulary that was more natural and easier to use. Because the WIMP interface was orders of magnitude easier to use than textual based applications, users no longer had to be experts. With improvements in hardware and refinements of interfaces, the computer has become a mainstay item in society.

The graphical framework is where most mainstream interface designer takes place. In addition, much research also takes place under this framework. A current trend in research continues to build on the motivation behind the WIMP interface's most common metaphor and well-known characteristic, the desktop. Just as WIMP interfaces bridged the gap between computer world and real world by establishing something common among them, current research seeks to bridge the same gap in similar ways.

The remaining portions of this document will discuss related research to collaborative environments and the hardware to support them. In particular the DiamondTouch hardware will be discussed and how it can be used to support collaborative CRC Card sessions. The software developed to conduct CRC Card sessions on the DiamondTouch will also be discussed. An experiment was conducted to test the usability of this system. The experiment process is documented and the results are analyzed. With this information, an evaluation is made on the software, the hardware, and the entire system

2. RELATED RESEARCH

2.1 Tangible User Interfaces

One such area of research is Tangible User Interfaces (TUI). TUIs have gone one step beyond WIMP interfaces by exploring the relationship between physical representation and digital information [9]. Rather than have some digital representation of digital information, a physical object is used to represent digital information. Users no longer manipulate information through a set of software tools; a real physical device is used, that a computer can interact with.

The main difference a TUI and WIMP interface, (a Graphical User Interface, or GUI), is the distinction between input and output devices. GUIs treat input devices and output devices as separate devices that serve different purposes. For example, GUIs rely a mouse as an input device and the monitor as the output device. The mouse in this example will never be an output device. Tangible interfaces seek to eliminate this distinction [9]. Consider a mouse that provides feedback to the user by increasing resistance to movement while the mouse is over an item on the desktop. The mouse in this example is still an input device, but it is also an output device. These devices output information in a form that humans directly perceive by their senses. Use of all human senses in parallel offers the potential to revolutionize how humans and computers interact. The goal of TUIs is to work towards new computationally mediated interfaces that permeate society and seamlessly become part of everyday life.

These new devices and interactions have certainly captured the attention of many, but wide spread acceptance will be difficult to achieve. This is mainly due the interaction frameworks that already exist and their associated constraints. GUIs dominate mainstream computer use and switching to TUIs requires hardware to be redesigned. This redesigning of hardware to support TUIs will raise the cost. This increase in cost can potentially overshadow the added functionality in the public's eye. A force feedback mouse is not needed when compared to a normal mouse, though it is novel, it is not clearly an improvement. This will be the same story for all new hardware; only time will overcome this.

In addition to the issues dealing solely with the hardware, the software also becomes an issue because of the lag time between the release of new hardware and the release of software that fully utilizes it. As with the monitor when it was created, no system could use it. Initially, the monitor did not replace the printer; rather it was used in addition to the printer. Over time, the monitor replaced the printer as the primary output device as more software became available that utilized the new functionality that a printer could not support. The same will occur for new hardware. Software will catch up with the hardware to allow seamless use, but that day is not here.

The last remaining issues are the properties of the physical devices themselves. As anyone who as worked on a small desk, space is a premium. These new input/output devices typically are not small when they first become available. The replacement of a single input device with a input/output device will simply take more room. This is only compounded by the fact that these new devices have very specific uses and do not lend themselves to general use. For example, computer joysticks have evolved to include force feedback. This qualifies as an input/output device for TUI. Evolving from these latest joysticks, are steering wheels with force feedback. The joystick could be used in a general task, however the steering wheel is very application specific and does not make sense outside of the application that uses it. Also, note that many of these devices are connected by wire to the computer. Though WIFI enabled devices are becoming more available and popular, but the technology is not available to eliminate the clutter of multiple wires for multiple devices. Desk space will suffer immensely till a input/output devices is created that can support general purpose tasks.

The drawbacks of such hardware and interaction are easy to see, but the benefits that can be obtained from this type of interface are not quite clear. The expectation from the research is the language of communication established between computer and user will become less important as it becomes more natural. This language initially was not completely natural because computers simulating real world objects forged it. Though this is language is based off natural interactions, it could be more natural if tools were actually used rather than just simulated. TUIs cultivate tool-like, rather than language-like, modalities of interaction [9]. Humans have distinguished themselves from other animals by use of tools and interact with their environment through all senses simultaneously. Current interfaces use only vision and to a lesser extent touch and hearing. If an interfaces can be built that take all senses, or at least more, into account, then a new level of usability, and thus productivity, can be potentially achieved. This will also accelerate the integration of computers into society.

2.2 Immersive Envirionments

Another area of research that is closely related to TUIs is an immersive environment. Immersive envirionments fall into the category of augmented reality. This research differs from TUIs in the environment that each uses. TUIs use the real world and try to fit their objects into this world. Immersive environments do not follow this rule; instead they complete remove the physical world in favor of being able to create a new one from scratch. Virtual Reality is the most well known immersive environment.

Immersive environments are similar to GUIs in that they seek to create an interface, which is not bound by the laws of the physical world. They differ however, on the scale in which they do this. GUIs, particularly WIMP interfaces, create an interface that is part of the environment. Immersive interfaces *are* the environment.

This type of research has taken concepts from other areas of research, changed the scale and applied them in new ways. The result is a new interface, which has characteristics of other interfaces yet is completely different.

With such a different approach, many problems that were not problems for the donating interfaces are now problems that must be solved. One problem is the based on the potential separation of the physical environment and the digital one that is created. If the governing laws of this digital environment are not carefully paid attention to, the user will not know how to function in this new world. They will essentially be infants in this new environment and will have to learn how to function. For example, keyboard layout illustrates this perfectly. Consider a computer user who is proficient with a QWERTY keyboard, but is now asked to use a Dvorak keyboard. The proficiency that the user once had is gone and they must relearn how to type. The Dvorak keyboard is considered better than the QWERTY keyboard because of the arrangement of keys, but it would still hinder user performance if the user were not familiar with it [8]. Changing the environment, though the resulting environment may still be similar, can yield the same effect. Usability, will initially suffer before it ever gets better.

Another issue that always affects new hardware is the availability and price. Because this technology is largely unperfected, it has not been made available to the masses. Once it is made available to the masses, it will be slowly adopted for the same reasons as TUI hardware and monitor. There will be little software support for the hardware. It will be an alternative to other interfaces, not a replacement.

The main benefit of using an immersive environment stems from one of its weaknesses, the decoupling of the environment with the physical world. This complete separation has opened the doors for innovative ideas by removing the constraints which designers have been forced to stay within for years. Research has always sought to push the barriers, but never to completely remove them. Innovators are now only limited by their imagination. The potential of such a freedom in design is unseen. With the correct vision, no one knows how revolutionary the idea may be nor does anyone know what will be left in its wake.

2.3 Evaluation of Related Research

The purpose of discussing these current areas is to draw attention to the fact that no silver bullet has been found for the future of interfaces. No replacements have been found, only alternatives. The various interfaces that have been found have benefits and drawbacks, but no one is clearly better than another. They are simply different. The characteristics of the next revolutionary interface are unknown. Pieces of this interface may already exist, just in different areas of research. This interface may very well take the form of various insights gained by distant research areas combined into a single user interface. Discovering this next interface will only be achieved by sharing insights across research area boundaries.

Sharing has occurred between research areas in the past and continues to happen today. However, this sharing is selectively occurring and in a restrictive manor. It seems that this sharing of information has primarily focused on single user systems with single user interfaces. Collaborative research in large has been ignored and left behind by many researchers. The current domination of single user interface research has focused on increasing the productivity of a single user, but has failed to look at leveraging what has already been achieved. This leveraging of what has been already achieved can be done by careful utilization of the most powerful and adaptive engine known, the human brain.

Applying these insights gained in single user interface research to collaborative interface research will seriously advance the state of research in this area. Though these new ideas may have to be adapted to fit multi-user systems, they will still change the landscape of collaborative research. Likewise, collaborative research can greatly impact single user interface designs. Collaborative research looks at how people work together in the presence of computers and through computers. This dialog that takes place is very natural and spontaneously occurs. If insights can be gained into this natural exchange between users, then it might be able to be applied to how humans interact with computers making it more refined. After all, making human computer interaction more natural is one the main goals of almost all interface research.

3. COLLABORATIVE HARDWARE

As new interface frameworks are created, they are often preceded by new hardware or new functionality for existing hardware. Collaborative environments are no different in this regard. Recently, two companies, Sony and Mitsubishi Electronics Research Lab (MERL), simultaneously developed two different synchronous multi-user touch pads respectively. Sony's system is called SmartSkin, while MERL's tablet is called the DiamondTouch(DT). Both are still in the prototype stages of development. These devices are revolutionary because they support multiple users simultaneously and have the potential to change collaborative interfaces indefinitely. In addition, this new hardware could extend to single user interfaces.

These devices do not fit any certain research area. Obviously the DiamondTouch and SmartSkin are collaborative tools, but they can also be categorized as a TUI or an immersive environment. These devices can create a digital representation of a physical object, similar to the way a WIMP interfaces does, but is manipulated as if it were a real tangible object. The interaction is similar to that of a TUI, just the representation is digital rather than physical. These devices are also pseudo immersive environments because as they create the digital representation of physical objects, they also build the environment that these object exist in. Immersive environments call for a removal of the physical environment in favor of a digital one. These devices instead provide a platform to create an environment that fits into the physical world while being able to control the rules of the environment and still have natural use.

The main difference between SmartSkin and DiamondTouch is the way that each device tracks individual touches. SmartSkin is able to unambiguously determine individual touches, but is unable to distinguish between users [6]. The DiamondTouch on the other hand is able to distinguish between users, but is unable to unambiguously determine the coordinates of multiple touches from a single user. Ideally a device that tracks multiple touches per user as well as multiple users is desired, but since this technology is so new, this is an unrealistic expectation.

Since this technology was developed to support collaboration, tracking individual users is more important than being able to fully support multiple touches. Though individuals may be limited by this constraint, the support for tracking individual users will be advantageous in researching collaboration.

3.1 DiamondTouch

The DiamondTouch looks like a typical table based touch pad. The DiamondTouch, unlike other touch pads, also serves as the display. The DiamondTouch is not a rear project device like a monitor; rather it must have an image projected onto its white surface. This is typically done via a projector mounted horizontally and a mirror to refract the image downward onto the DiamondTouch

The DiamondTouch works by transmitting a different electrical signal to each part of the table surface, which needs to be uniquely identified. When the user touches the surface of the DiamondTouch, these unique electrical signals are capacitively coupled from directly beneath the user's touch, through the user and received by a receiver associated to each user. These receivers are connected to the user by an electrostatic pad that the user sits on.

The DiamondTouch surface is embedded with antenna arrays arranged in a rectangular grid. These antennas can be of arbitrary shape and size and the arrangement can vary. A general purpose arrangement is optimal compared to an arrangement for a specific task. The current arrangement of these antenna arrays is called the "full matrix" pattern and is the most general of all arrangements. This arrangement is also the most difficult to manufacture. In reality the full matrix pattern may be unnecessary for many applications, but it offers the best support for new and innovative applications that may require the DiamondTouch.

The DiamondTouch has gone through several changes since it was first developed. It can be found in three forms: a tablet version, a serial table version, and the newest USB table version. Each version used the same antenna array. Each version improved upon previous version and with each change, the DiamondTouch comes closer to usable tool to the masses and less of a research lab prototype.

The tablet was the first version of the hardware to use the DiamondTouch technology. It was a 20cm by 20cm surface, which contained 80 imbedded antennas. These antennas were arranged in 40 rows and 40 columns. The tablet was scanned for user activity 75 times per second. This device passed raw data to a computer via a fast RS-232 serial connection. Though this device was small, it was a proof of concept and implementation [1].

The serial table was the next evolution of the DiamondTouch hardware. This device has 160 vertical antennas and 96 horizontal antennas for an aspect ration of 5:3. This new table, through weighted interpolation, increased resolution by a factor of 16 in both directions, resulting in a resolution of 2560x1536. Also, this version generates 15 frames/second/user and would deliver this raw data over USB cable connected to a computer. This USB connection is viewed by a computer as a series of serial connections, four of them to be exact. Embedded in the diamond touch is a Keyspan 4 port serial to USB adaptor. Only about 15 of these units were distributed outside of MERL. An interesting aside about this implementation was that the surface as made of white board material so dry erase markers could be used on it [1].

The latest version of the DiamondTouch Technology took the form of a true USB table. This version moved away from serial connections of any type and completely implemented a USB interface for the table. The DiamondTouch surface changed to contain 128 vertical antennas and 96 horizontal antennas for an aspect ratio of 4:3. This change in aspect ratio was done to better match the displays currently in use. The same interpolation methods were used as with the serial table to give a resolution of 2048x1536. This version also increased the amount of data collected about each user's actions by increasing the frames per second per user from 15 to 30. This doubling of the sampling rate allows the DiamondTouch to detect more details about the user's movement. This version is the best implementation yet not just because of the improvements listed above, but also because it appears to be less susceptible to RF noise than previous version [1].

The next generation of DiamondTouch hardware is said to be rear projected. This will make setting up the DiamondTouch easier. Other improvements are unknown at this time.

4. CATEGORIES OF TASKS

With the platform selected, the question now becomes what to do with it. Many tasks could be implemented using the DiamondTouch, but certain tasks with known characteristics will benefit more than others with this type of device. Though there are many characteristics that could be used to identify a task, relatively few are important. The questions that should be asked when evaluating a task are “is it group activity” and “does it have some set of physical objects that can be modeled on the DiamondTouch”. If yes can be answered to both questions, then the task is a candidate for the DiamondTouch.

4.1 Games

One of the first categories of tasks proposed was games. Games are multi-user tasks and typically have some physical objects. Games meet the requirements for being a candidate; in addition, games in general are enjoyable. The specific games themselves had to also meet the same selection criteria. The initial games considered were air hockey and memory.

Air hockey was the first to be implemented. This game consists of two users trying hitting a plastic disk into the other user’s goal. On a real air hockey table, the plastic disk rides on a cushion of air. The DiamondTouch simulates the plastic disk riding on the cushion of air and places the user’s paddles at the points where the users are touching.

This idea meets the selection criteria of multi user with physical objects. This idea was actually implemented on the DiamondTouch by Thomas Horton using C++ and open GL. This served as proof that the DiamondTouch could be used as part of an application.

The next game implemented was the elementary game of memory. This game consists of two or more users trying to match pairs cards which are face down. Each user takes a turn turning over two cards. If there is a match, they go again. If there is not match, the cards are turned back over and the next user goes. This meets the criteria of a multi-user task with physical objects being used. This game was also implemented on the DiamondTouch by a team of students, including myself, for a graduate course in Human Computer Interactions. This application was done in java rather than C++ to prove the Java could also used with the DiamondTouch.

The category of games was ultimately ruled out as a viable candidate for the DiamondTouch because of the adversary relationship that participants establish with each other. With this type of relationship between users, communication will certainly drop leaving less to be quantified and examined for meaning. The selection criteria should be refined to look for a multi-user task that is collaborative.

4.2 Software Design Tools

The next category to be examined was Design tools for computer scientists. Computer Scientists typically work alone most of the time under older development methodologies, but engage in highly collaborative design sessions during the design of object-oriented systems. This category is certainly a multi-user task with each participating in a collaborative manner.

4.3 Universal Modeling Language

The first idea examined in the category of Design tools was Universal Modeling Language (UML) design session. UML is basically a graphical representation of code at a high level. UML lends itself well to being implemented on the DiamondTouch due to the similarities between the DiamondTouch's surface and a large sheet of paper where UML is typically drawn. The requirement of having a physical object to model on the DiamondTouch is met. However, UML is typically not created in a collaborative manner. People may collaborate during the creation of UML, but collaboration typically occurs as part of a reviewing process, not during the actual creation. Participants would not use the DiamondTouch to collaborate, but would instead use it as a surface to project a design onto for evaluation. This is would be poor utilization of the hardware.

Collaborative activities seem to be highest during design stage of software. Since UML has already been ruled as a candidate, the only other design work that typically happens during software development is Class Level Designs. Class Level Designs involve software engineers deciding how a software system will be laid out based on a set of requirements or specs. A design session of this type will require the participants to work together to design a system that utilizes individual classes that each participant creates.

4.4 Class Responsibilities Collaborators

One type of class design is called Class Responsibilities Collaborators (CRC) cards. These cards are used to represent an object in the design of the system. A card is typically a 3-inch by 5-inch note card, which has been divided into 5 sections to contain different information [10]. The first of these sections is the class name. This name should give a reasonably well idea about the functionality that will be encapsulated in the class. The next section is the super class or parent. This section is used to establish a hierarchy to the objects. The third section is the responsibilities section. This section will include information about what the object is specifically responsible for doing. These responsibilities will come from the requirements and often for the basis for the methods in the class. The next section is the collaborators section. This section will include information about what other objects will be used to accomplish its responsibilities. The last section is the description section. This section can contain any information that is deemed important to the specific class. Most often, this section is used for a brief description of how a responsibility might be accomplished. With this information, framework code can be written. More importantly, a system design can be created [10].

CRC cards are created during a design session called a CRC Card Session. During this session, users will create objects around specific functionality. The responsibility for creating an object is equally shared among all participants. In addition to just creating objects, users are responsible for editing the information about the object. Again, this responsibility is equally shared. Often, a card is created by one participant and then edited by others. Participants fill in information on the cards, the actively engage in active role-play as they animate scenarios, or use cases, for the system [2]. When a design session is complete, the ideas of all users about a design have been considered and may potentially be incorporated in the design.

A CRC Card Design session is a perfect match for the selection criteria specified, as they are intrinsically group based [2]. The design session has a goal of creating a system design, participants must work together to accomplish this goal, and there are physical items that can be modeled on the DiamondTouch.

5. COMPUTER SUPPORTED COLLABORATIVE WORK

Software of this type, which is aimed at facilitating tasks of this nature, is actually an area of research unto itself. This area of research is Computer Supported Collaborative Work (CSCW), specifically GroupWare(GW). CSCW aims to understand how software can aid in the accomplishing of collaborative tasks. This includes understanding the social interactions of humans collaborating and how individuals interact with a computer while working on a collaborative task [2]. GroupWare is a much more narrow area of research. Its main focus is on the technology designated to support group activities.

5.1 GroupWare

GroupWare can further be divided into single or multiple displays and synchronous or asynchronous work. Each category has benefits for the type of collaboration that is must support.

Single display GroupWare focuses on a single display coupled with multiple input devices connected. The purpose of having multiple input devices is to overcome the limitations of current hardware for single user use [5]. For example, a touch screen only supports a single user at a time.

Multiple display GroupWare focuses on multiple displays and how users use them [5]. These displays may or may not be connected to a single computer. One interesting problem that is currently being examined is collaboration in separate locations through software [2]. Depending on the answer to this problem, the way businesses collaborate with each other could change so that travel is no longer needed for this type of collaboration.

Asynchronous GroupWare systems focus on collaboration when participants work on a common task at separate times [5]. This is a very realistic model of collaboration, as schedules often do not match up correctly between participants. This could also have an interesting effect on business.

The last group is Synchronous GroupWare Systems. This area focuses on a more "traditional" collaboration where participants are working at the same time on a common problem. It is important to note that participants do not have to be in the same location, all that matters is the participants working at the same time [5].

For the purposes of the Collaborative CRC Card application on the DiamondTouch, a single display will be used with synchronous activity. Due to the hardware restrictions, a single display is almost required. CSCW research has shown for synchronous activities in the same location, a single display is optimal [5]. Given this current set up, productivity should be enhanced over other implementations.

5.2 Alternative Implementation

Though the hardware is new, the idea of a collaborative CRC card application is not. This idea has implemented at the University of Canterbury in New Zealand as a synchronous multi-display device. This type of GroupWare was chosen to allow conference participants to be located anywhere. The problem with this implementation is the nonverbal communication is lost. While designing this system, the decision was made to not mirror the physical items used during a CRC card session. Rather, the abstract idea that the card represented was modeled which could be acted upon in a similar manner. Research shows that real cards increase the level of individual involvement, particularly during the enactment of scenarios [2].

The DiamondTouch, with digital representations of the physical items, can facilitate the collaborative tasks of a CRC Card Design session and perform at least as well as the traditional "by hand" approach. This does not make any prediction about the overall quality of the design, it simply means that a CRC Card Session can be conducted on the DiamondTouch while providing better support to the users during the design session. It is alleged that the users will conduct the design session more effectively with the DiamondTouch due to this support.

6. DIAMONDTOUCH CRC CARD SOFTWARE

The motivating factor for the development of the DT CRC was natural interaction. Natural interaction is not just defined as verbal communication; it includes all forms of communication between an individual and their environment. Supporting these other communication channels become pivotal since having an awareness of others is essential for coordinated activity [4]. At the same time, these other communication channels need to be managed as they could distract the user. A user's attention is finite and thus, should be treated as a resource [7]. If the participants are able to design a system and are not hindered by using the DiamondTouch, meaning necessary communication channels have not been severed, then that goal has been accomplished. To achieve this goal, the software went through three iterations. Specifically, the user interface went through three iterations.

6.1 DiamondTouch CRC Iteration 1

The first iteration of the DiamondTouch CRC card software was overloaded with restrictive procedures for users to follow and did not offer the full functionality that users needed. The main focus for this iteration was getting a working prototype of the software with a user interface that supports the activities. Screen shots of this interface can be seen in figures A.1, A.2, and A.3 located in Appendix A.

The user interface was designed and implemented with the idea that user's have their own personal space in which to work and ownership of the tools which they use. For collaborative tasks, this personal workspace must become a shared space for both users to work. From this the basic design, a large workspace is placed in the center of the user interface flanked on each side by a tool pallet for each user.

The workspace shared by both users is the area where cards are created and arranged to create the system design. Since this area will contain the CRC Cards, which will be a shared item between both users at times and a privately held at other times, rules had to be formed around the manipulation of the cards. When a card is being editing by one user, it is not acceptable for another user to take the card from the first user. This is the type of rule that the DiamondTouch CRC Card software had to enforce. In addition to this rule, the other user should not be allowed to manipulate the card in any way. If the other user needs to manipulate the card, they will have to communicate this desire to the owning user and have them release their ownership of the card. The state of ownership will be preserved till the user gives up ownership by taking ownership of another card or touching the background of the workspace. This is similar to the rules enforced by society when conducting a CRC card session except leaving a card for a long period of time will not remove ownership on the DiamondTouch CRC Card software as it might in a real world CRC Card session.

Since both users must share this area, the workspace cannot be divided to support the personal workspace that each user requires. To over come this issue, the separation must be rule based and selectively enforced. This rule is based upon the observation that when a user is editing information on a card, the card is positioned in front of the user. To mirror this observation, the user's card will be repositioned onto their side when the card is being edited. Since the card is considered "owned" while being edited, the other user would be unable to move the frozen card from their personal workspace if it was left in their workspace. This could negatively affect their performance on a task. In social terms, this would an inconsiderate action and would be frowned upon.

As expected by the sheer dominance of the shared workspace in the user interface, most of the touches, and thus activity, will occur in this area. An interesting issue arises from this observation: are all touches intended to be treated the same. The answer to this is no. In some cases the user wishes to take "ownership" of a card, which involves finding the visible card at the point that the user is specifying. In other situations, the user already owns a card and is touching an item on the card to select it. In this case, the touch needs to be validated as being within the boundary of the card and validated as being a selectable item. The problem that needs to be solved is how to decide what the context of the touch should be. The solution to this problem is “user modes.” These “user modes” are used to determine when context functionality should be enforced. The user mode is defaultly not specified to allow the user to freely move the cards around the workspace. The user mode changes as the user starts a process by pressing one of the buttons in the tool pallet. Once this user mode has been changed, the rules for the associated process are enforced. When the process has completed, the user mode is returned to the default value.

The tool pallets for each user can be viewed as part of a user’s personal space. For a real world CRC Card Design session, each user has a pen, which they use exclusively. The tool pallet inherited this exclusive property, meaning that the tool pallet is user specific. Just as a user cannot take other user’s card when it is owned, users cannot “impersonate” another user and use their tool pallet. Each user is given the same tools so there is no reason for one user to need the tools that are assigned to the other user. By both users having their own tools, they can work in parallel, thus eliminating blocking between users [5].

Another issue around the tool pallets was the layout of the buttons. The motivating factor behind the button layout was frequency of use and impact of use. The first requirement simply looks for the buttons that are going to be used the most frequently and seeks to minimize the time reaching for the button. According to Fitt’s Law, the movement time to the specific button is a function of the distance to the button. For this reason, the most commonly used buttons will be located closest to the user. The second requirement seeks to minimize damage to the design by accidentally selecting the wrong button. The first button to be placed was the exit button. This button will only be used once per session and is very destructive. It will shut the entire system down. This is why it is placed at the top of the tool pallet. The next button, delete card, is going to be used more than the exit button, but not as much as other buttons. It is also a destructive operation. It will delete a card from the session. Once the card is deleted, it cannot be recovered. Since the Exit button has a more severe consequence of misuse and will be used significantly less than the delete card button, a gap was placed between them to help guard against accidental pressing. The third button to be placed was the create card button. The general belief is that more time will be spent editing a card than creating new ones. This is not a destructive action since it creates a card. The card that is created can just be deleted. The order of all the Edit buttons was done based on order they appear on the card. The last button is the least destructive and most commonly used of all. It is the flip card button and is placed on the bottom of the tool pallet, closest to the user.

The text on each button was also changed to minimize the required width of the tool pallet. The text was truncated to allow the workspace to be maximized. The Edit buttons all lost the word "Edit" from their button. Though this created ambiguity as to the function of the button, the tool pallets were using too much space.

The last remaining issue addressed in the user interface was the input of text. The keyboard model was chosen as the input device since most computer users are familiar with how to use it. But adding multiple keyboards with the DiamondTouch hardware could possibly confuse the user about how to use the DiamondTouch. To overcome this possible problem, a soft keyboard was created to appear within the application. The reason for the keyboard appearing within in the application instead of another window was to keep the same context for all operations. Having different windows appearing and disappearing would only add an unneeded layer of complexity that could confuse the user.

When the keyboard is displayed within the application, it has the potential to block the card being edited and the buttons on the lower portion of the tool pallet. Since the keyboard could not be moved, the buttons and the card could never occupy the same space as the keyboard. This was easy for the buttons since they don't move. The buttons were simply arranged in the upper portion of the tool pallet. The cards however are more complicated than the buttons since they can move. This was address by shifting the selected card up to a position where the keyboard would not cover it when the keyboard appeared.

6.2 Iteration 1 Processes

When creating a card, the user would hit the create card button which would create a new card and pop a keyboard for the user to use. The user would be forced to enter a name for this new card and either hit enter on the keyboard or the create card button. If no value was entered for the name, no card would be created. If the name was specified, then the user would be forced to enter a parent for the card. This value was allowed to be blank. The user would then hit the Create Card button at this time and the keyboard would disappear. The user could now move the card around the work area. The user would not be allowed to add or edit information in the name or parents field on the card.

Editing information on a card has much of the same types of rules as creating a card. For editing information on a card, an edit button existed for each category of information. For example, to edit the description, the user would hit the "Description" button. If the section associated with the button was not showing at the time of the button press, then the card would be flipped. The color of the button would then change and the user would select the item in the section of the card that they wished to edit. If the item selected was of the type associated with the edit button, the background color of the item would change from white to yellow. If the selected item was not of the type associated with the button, the selection would be disallowed. The user would not receive a prompt to select an item on the card for the associated type, nor would they receive any information as to why the selection was disallowed if an invalid item was selected. The next step for the user is to hit the edit button again. If the user has made a valid selection from the associated section on the card, then the edit button would change color again and a keyboard would be displayed. If a valid selection was not made, the button would return to its default color and the user would be allowed to move the card around the workspace. Once the keyboard appears, the user can edit the selected entry on the card. When the user finishes they would hit the edit button again. The keyboard will disappear and the button will return to its default color. To edit another item on the card, the user will be forced to go through all the steps above again.

6.3 Iteration 1 Evaluation

CSCW Research has shown that prevention of turn taking, structuring of the idea space, and no process constraints are requirements for CSCW tools that support idea generation and structuring [5]. The first two requirements have been met, however the third has not. In all, there are different user modes for the various processes. These processes do not mirror the same process that the user would go through if they were doing the same task without the software and the DiamondTouch, nor is there any information that could be gleaned from the environment for the user to figure out these processes. The user's expectations for given actions do not match what is displayed. This leads to user frustration and ultimately the DiamondTouch not to be accepted as a viable tool for CRC Cards. The usability of this version of the software would negate any benefit of using a tool of this sort.

6.4 DiamondTouch CRC Iteration 2

The second iteration of the DiamondTouch CRC Software saw the implementation of additional functionality and process refinement to make them more natural. Screen shots of this interface can be seen in figures A.4 and A.5 located in Appendix A.

6.5 Iteration 2 Changes

The only real visible change, not related to a process change, was the addition of the Edit Parent button. This caused the layout of the buttons to be change. The button layout was again done with the idea of frequency and impact of use. As before, the Exit and Delete

Card buttons were located farthest from the user to avoid accidental contact. The Edit Parent button was then inserted between the Create Card Button and the Responsibilities Button. Users are more likely to edit information on a card than creating new cards. Users are also more likely to edit information other than editing the parents. This was the motivation behind the layout of the buttons. The Edit Parent process was similar to the editing processes from the first iteration, except the step where the user has to select the item to edit has been removed. When the user presses the Edit Parent button, the card is flipped if needed and a keyboard is displayed. The row on the card corresponding to the parent entry is selected by default since there is no ambiguity as to which parent item the user wishes to edit since there is only one.

The editing process also changed to be closer to the process described for editing the parent information. The process was changed so that the user did not have to select the item they wished to edit before the keyboard was displayed. In order for the system to display the keyboard something had to be selected for editing. This stipulation was overcome by selecting the first item in the desired section by default. The user's expectation that keyboard to be displayed when the edit button was pressed was meet. Another change in the editing process was the removal of the requirement that the user had to go through this process for every item they wished to edit. Now, users could select another item to edit, provided that it was in the same section. Users can change the selected item by either touching another item or by using the tab key to cycle through the items. In addition to the support for the tab key, support has also been added to the enter key. Most users use the enter key to indicate that they are finished with the task they are currently working on. In version two, the software would recognize this as indication that the user no longer wished to be in an editing mode.

6.6 Iteration 2 Evaluation

Version two was a complete version of the software with some progress made on its overall usability. The requirement for the context of user actions was still required, but the number of "user modes" decreased from nine in version one to six in version two. This is indicative of more flexible processes. Considering the addition of new features increased the number of user modes, this drop is fairly significant. The overall usability of this version is higher than the previous version.

6.7 DiamondTouch CRC Iteration 3

The third iteration of the DT CRC Software's main focus was to simplify and streamline the user interactions. No new features were added during this iteration of the software. The previous versions were dependent upon knowing the context of the user's actions and enforced rules based on that context. This version will move away from multiple contexts for similar activities. Screen shots of this interface can be seen in figures A.6 and A.7 located in Appendix A.

6.8 Iteration 3 Changes

The main refinement that was done was to editing process. Namely, it was generalized to apply to all items on a card. Rather than having a process for editing the description and another for the responsibilities, the new editing process is unaware of the type of item being edited. This generalized process removed the need to have multiple buttons to edit the various sections on the cards.

The user interface, as an effect, became less cluttered. The buttons that were removed were difficult to understand exactly what they did. The title “Responsibilities” does not describe exactly what the button does. An experienced user would know that it edits the responsibilities section, but new users would not be able to figure this out without using it. The button that replaced all the non-descriptive edit buttons was the “Edit Card” button. By pressing the “Edit Card” button, the keyboard would appear and the first item on side of the card showing would be selected. To end the editing of the card, the user would simply hit enter on the keyboard or press the Edit Card button again.

The create card process took on many of the characteristics of the edit card process; particularly it’s flexibility. Now users can enter the name of the card and then select other items on the card to edit. The rule still exists and is enforced to make sure the name for the card is specified. As before, if the name is not specified, the card is be deleted.

6.9 Iteration 3 Evaluation

The dependency on context was not eliminated, but it was reduced. The total number of user modes has been reduced to 3. Rather than having a user mode for each type of editing as before in version two, on a single user mode exits for all editing. This is by far the most natural of all the versions. The processes are no longer heavily laden with rules about when a user can perform certain actions. Rather, users can do pretty much anything they want. This flexibility of the system allows users to interact with each other and the software without the software forcing a process onto the collaboration.

7. EXPERIMENT DESIGN

7.1 Purpose

The purpose of this experiment is to prove that the DiamondTouch is no worse with respect to performance and collaboration when compared to doing the same task by hand. As stated before, no claims are being made about the quality of the design produced, this aspect is completely user controlled.

7.2 Tasks

This experiment will have groups of two users participate in CRC Card Design sessions. The participants will design solutions to two different problem descriptions. These problem descriptions will be different in the details, but abstractly they will be the same. By having the problems be abstractly similar, the difficulty of the design can be kept constant between experiments.

The First task is a parking lot management System for an airport and the second is a pay-at-the-pump management system for a gas station. For a description of each, see sections 10.2.1 and 10.2.2 in Appendix B. Both of these systems are essentially a vending machine of sorts. The parking lot management application dispenses parking spots and the gas pump application dispenses gallons of gas. If the design for one of the problem descriptions was generalized enough, it could be reused as the system design for the other problem description.

7.3 User Selection

Taking the task into consideration, it is fairly obvious that the experiment participants need to be computer programmers. All participants have some minimum knowledge of programming and design. Since this task deals with design, which is first taught in software engineering, all participants should be currently enrolled in undergraduate Software Engineering or have previously taken the course. Of the eight groups tested, three groups were currently enrolled in the software engineering course, two groups were seniors in software development programs, and the remaining three groups were professional developers.

Each participant was matched to another participant in terms of skills. This was done so no one user in a group would dominate group decisions based solely on having more knowledge of system design. Also, an effort was made to form groups were the participants had interacted and collaborated with their partner. All groups ended up having worked with their partner before the experiment.

7.4 Execution

This experiment will be conducted in the Knowledge Discovery Lab in Venture III on NC State's Centennial Campus. Each group of users will participate in the experiment separate from other groups so as to not influence the findings. Each group will be asked to design a system using the traditional CRC card technique to a specified problem, using actual note cards, and also design a system of similar complexity using the DiamondTouch. Both experiments will be conducted in same lab to keep the environment constant between experiments.

Though the design problems are similar in abstraction and different in specifics, some users may see the similarities between the two tasks lending them an advantage when conducting the second task. To overcome this, the order of the task will change for each group.

The platform used for each task will also change for each group. This will help equalize any advantage that a participant might gain during the second task and spread it equally between the results gathered from the experiments being done by hand the experiments being done on the DiamondTouch.

The process for conducting the experiment by hand will follow the following process:

- Video taping will start once both participants are seated at the design table
- Each participant will be given a print out over viewing what a CRC card is, how to fill out a card, and how to use them in a CRC card design session. It will not be necessary for the participants create the cards because a stack of CRC cards will be provided for their use. This may or may not be the first overview of CRC cards depending on the ordering in which experiments are conducted. The

proctor of the experiment will not answer questions, but questions can be asked of the other participant.

- Then users will receive a task for them to implement during the CRC card session, after the participants have finished reading the description of the task, a timer will start that will be used to record the duration of the design session. The participants will now be allowed to start on the task.
- Time will stop when both users agree the design complete.
- Videotaping will of this section of the experiment will stop after the timer has stopped.

The portion of the experiment dealing with the DiamondTouch will include all the procedures used for the CRC card design session conducted by hand as well as others that pertain only to the user of the DiamondTouch and it's software.

- Videotaping will start once both participants are seated at the DiamondTouch.
- An overview of how to use the DiamondTouch and how to use the software be offered to the participants. Participants will be allowed to ask questions about how use the hardware and software of the proctor. Once this overview session has concluded, participants may not ask the proctor any questions.
- Each participant will now be given an overview of what a CRC card is, how to fill out a card, and how to use them in a CRC card design session. Again, the CRC card will be formatted for the user, this time by the software. As before,

participants will not be able to ask questions of the proctor, but will be allowed to ask questions of the other participant.

- Participants will now receive a problem description for which they must design a system that will solve the problem using CRC cards. After both participants have finished reading the description, a timer will start that will be used to record the duration of the design session. The participants will now be allowed to start on the task.
- The timer will stop once both participants agree that the design is complete.
- Videotaping will of this section of the experiment will stop after the timer has stopped.

After both experiments have concluded, the participants will take part in a feed back session. This will be done at the same workspace as the last experiment conducted for each group. This stage of the experiment will be used to gather qualitative data about the two CRC card sessions respectively as well as user's feelings about their experience.

- Participants will fill out a written questionnaire independently of their partner. This form is located in section 10.2.3 in Appendix B. Note: Videotaping will not be conducted during this portion of the feedback session.
- Upon completion of the written questionnaire, videotaping will resume where the participants will take part in an open-ended feedback session. It is believed that the feelings of one participant will influence the other and vice versa.

Collectively the two users should collaborate to summarize their feelings of the experiment.

Then the open-ended feedback session has completed, the videotaping will stop and the participants will be excused.

7.5 Information Gathered

The items that will be collected from this experiment are videos of each group's user testing and their feedback session, the user questionnaires, and the actual design created during the CRC Card sessions.

The videos will be used to gather numerical data to analyze. The three sets of numbers that will be gathered from the videos are the total time of the task, the time spent talking by each user, and the time spent actively working on a task. The total time will be gathered as metric for user performance. This will give an idea of the overall performance of a group for the given task. The time spent talking by each user will be used as a metric to evaluate the impact of the DiamondTouch on collaboration. This metric is important since this task is collaborative in nature and the DiamondTouch aims at extending computer-based collaboration. The last metric, active time on a task, will be used to evaluate how much time each user spent actually being active. This last metric, when used with the total time, will show what percentage of the total time actively engaged in designing. This is important because in collaborative tasks, users have to work together as well as accomplish tasks independently.

The user questionnaires will be used to gain insight into the user perception of their designs and the DiamondTouch. This information will be largely opinionated and will be treated as such. The basis for the qualitative analysis for the system will come from information gleaned from the user's evaluation and the observations made from watching the user's work on each task.

The last artifact that will be gathered from this experiment is the design. The design will be evaluated for completeness to safeguard against overly simplified designs leading to misleading numerical data. The design will also be used to gather information about the design itself. The number of words will be used to gauge the amount of work done. Also, the total number of cards created will also be used to evaluate the complexity of the design. The total number of words will be used with the total number of cards to give a numerical representation of the amount of depth in the design.

7.6 Expected Results

It is expected that the DiamondTouch will be, at worst, par with doing the tasks by hand. Specifically, there will be weaknesses and strengths to using the DiamondTouch, however culmination of all of them should balance out.

The performance is expected to be slightly worse, not necessarily because of the interaction technique, but because of the hardware and it's perception. The interaction technique, though new, is modeled after the real world and is composed of real world items. Since this interaction technique similar to the real world, it is expected to be negligible to the user's performance.

However, something that cannot be assumed as negligible is the user's perception of the DiamondTouch. From initial user studies of other applications written for the DiamondTouch, user's viewed the device as fragile because of it's novelty. This view made the user's constantly aware of how they were using and interacting with the device. If the user where to use a mouse instead of the DiamondTouch, the user's would not even pay attention to the mouse because it is not novel. This is expected to hinder a user's performance.

The hindrance to the user's performance from the novelty of the device, we believe, will be balanced out by the familiarity of items used to create the interface. This is directly related to the similarity of the software to the real world. More importantly, it is believed that the use of keyboards will actually facilitate better performance from the users. This is attributed to the ability of most users, considering the participant selection, to type faster than they may be able to write the same text using a pen. Also, the text that is written is guaranteed to be legible to anyone who tries to read it, this same guarantee cannot be made for using a pen due to differences in handwriting.

The usability of the DiamondTouch is expected to be about par as well. As before, there are differences when using the DiamondTouch when compared to doing the task by hand, but this should be negligible. Using the DiamondTouch has two main limitations. The first one limitation deals with how the image appears on the DiamondTouch. Since the image is projected from above the DiamondTouch, the user's hand will actually be blocking the image from being properly displayed on the screen. This is viewed as insignificant because when a user takes part in an activity using direct manipulation via their hands, their hands actually block their view of the task. The other main limitation of the DiamondTouch is the size of it. Though it is not overly small, it is significantly smaller than a table is. However, it is believed that the DiamondTouch has adequate room for conducting the task of designing a system using CRC Cards.

The ability of the user's to collaborate should not be impacted and maybe positively impacted. As stated before, the DiamondTouch is novel, but the interaction technique is not. Since users still interact with artifacts in a similar way, though they are in a digital format now, it is expected that the users will interact with each other in similar ways as if they were not using the DiamondTouch. It is actually possible for the use of the DiamondTouch to foster collaboration since both users are going to be new users to the DiamondTouch and will have similar questions about the use of it. The users can help each other do different tasks, in essence teaching the other user about the device.

8. EVALUATION

With the results from the experiment, the system will be evaluated both qualitatively and quantitatively. But before any evaluation can be done on the system, the parts comprising the system must first be evaluated. Since the DiamondTouch is a new device and the software is also new, each should be evaluated individually so benefits and drawbacks can be correctly identified.

8.1 DiamondTouch Hardware Evaluation

The DiamondTouch is the half of the system and the piece that allows for research such as this. The DiamondTouch is a very unique system and affords many new interactions that were previously not available. The main benefit of the DiamondTouch hardware is the support of new interaction for computers. Despite being an innovative piece of hardware, there are still problems with the DiamondTouch.

The first category of problems related to the image. Because the DiamondTouch is not rear projected, a project has to be used to project an image down onto the DiamondTouch surface. The expected issue of the user's hand blocking the image was indeed an issue, but as expected not a significant problem. The mirror proved a more significant problem. The mirror used to project the image down onto the surface of the DiamondTouch was a glass-based mirror. Glass based mirrors have a sheet of glass over the mirror to protect it. In this application, the glass actually reflected some of the image, which caused two images to be reflected onto the surface of the DiamondTouch. This caused a blurry image to be projected. For short uses of the diamondtouch this would not be an issue, but for long tasks using the diamondtouch this would be a significant issue. This could lead to user fatigue. The correct mirror to use in this application is a "first surface mirror" which is not glass covered. Since this mirror is not glass covered, the reflected image will be in focus on the diamondtouch.

Another issue with the mirror is vibrations. Since the projector is mounted to the ceiling and the mirror mounted to the projector frame. Any vibrations in the ceiling would cause the image to move slightly. This was not a significant issue during testing, but could easily become with a project mount more susceptible to vibrations.

The biggest issue with the imaging being project down onto the DiamondTouch is the calibration of the image. If the image is off slightly, then the user is not touching what they think they are by looking at the image. Slight vibrations in the ceiling can cause the mirror to reposition, thus throwing the image out of calibration. For any task requiring even a moderate level of precision, this will be an issue.

A different problem dealing with fine motor skills is the selection instrument; the tip of a finger. The tip of a finger is a very blunt object that does not lend itself to finely detailed work. The diamondTouch is based off of a single point representing the user's touch. This point could be anywhere under the user's fingertip. This ambiguity of where the actual touch is will make use of the DiamondTouch more difficult for tasks requiring any precision. Changing the selection instrument would make the interaction not as elementary and natural.

Overall, these issues facing the DiamondTouch can be avoided, or at least minimized. The DiamondTouch technology is not perfected yet, as no one would expect it to be, but it is well on its way to becoming a viable input technology.

8.2 CRC Card Application Evaluation

The CRC Card Application had little issues left to be resolved. Through the three iterations of the software, any divergence in the software from the way CRC Cards are handled in the real world were identified and resolved. Despite the software meeting all the requirements of it, there is still one major flaw in the design.

This flaw is the mixing of interaction techniques. When creating CRC Cards by hand, a pen is used to add and edit information on them. The software uses a keyboard to add and edit information on the cards. Using a keyboard is a typical way to modify text within a computer system so it should not be difficult for the users to use. This is especially true considering the demographic of participants. The keyboard is an issue because it does not match the way text is modified by hand. The goal of this application was to model as closely as possible the real environment for conducting a CRC Card session and provide the tools to the user to actually conduct one.

The switching of the pen for the keyboard should not cause the user any trouble in understanding how to modify text on a CRC Card. Since this substitution has been made, there is room for improvement by finding a different input device that more closely models how users modify text with a pen.

8.3 SYSTEM EVALUATION

The overall consensus from the users for the system was it was helpful and easy to use, except for the keyboard. The DiamondTouch worked well with the CRC Card software and provided most of the functionality that the users were expecting. This But there were shortcomings.

The first shortcoming was the lack of support for gestures. Gestures are an important means on conveying intention [2]. In the case of the DiamondTouch, a user could be pointing to a card as a reference for a discussion, without actually wanting to take ownership or select an item on the card, but the software does not understand this nor does the hardware support this. This significantly limits the vocabulary for interaction with the DiamondTouch.

Another shortcoming of the system dealt with editing of text. Typically, computer users are given the tools of cut, copy, and paste. Despite this application being computer-based, it did not provide these tools. These tools were not required to complete the task, but they were viewed as nice to have tools that would make the tasks easier to complete. Most often users would like to have this functionality when an item was being deleted from a card so the blank spot that was created could be filled.

Not having these tools would not be as big of a deal if users could easily re-enter text onto the card. This was not possible due to the temperamental keyboards. All users consistently complained about the keyboards, though most felt it was the implementation of the keyboards that made them difficult to use, not the concept of using the keyboard.

The reasons for the difficulty in using the keyboard mostly stem from the issues discussed when dealing the DiamondTouch exclusively. The main issue with the keyboard is the lack of tactile feedback. The surface of the DiamondTouch is completely smooth so users have no idea what they are actually touching unless they look at where their fingertips are. This leads to another problem discussed earlier; the hands are actually blocking the image from being projected on to the DiamondTouch. This issue is only compounded when the image is not correctly calibrated for the DiamondTouch. Not only are users having difficulty determining what their fingers are positioned over, they are most likely wrong because the image does not correspond to what is actually located at the specified point on the DiamondTouch. This issue is not helped by the fact that using the keyboards is a task that requires precision and fingertips are not the best choice. This created a frustrating situation for users trying to create and edit text on the cards.

As expected, the novelty of the DiamondTouch did impact some of the participants. However, this actually turned out to be less than expected; borderline negligible. Some participants expressed feelings of pressure to perform when using the DiamondTouch, but also stated that the DiamondTouch didn't hinder ability to collaborate, nor did it play a significant role in creating a better design. Users were more aware of the DiamondTouch while conducting the tasks, but it did not impact the process of accomplishing the task. Many users felt the DiamondTouch was in fact better at organizing the system design than doing it by hand. This is significant because it shows that the DiamondTouch CRC Card application can be substituted for the manual method from a process standpoint without negatively impacting it. However, the benefits of such an action have yet to be discussed.

8.4 Quantitative Analysis

Three metrics will be examined when evaluating if using the DiamondTouch performed at least as well as the manual method of conducting a CRC Card Session. These metrics are total task time, total time talking of each user, and the total time of activity for each user. From these metrics, the benefit of the DiamondTouch can be quantified.

The first set of numbers to be examined is the total time numbers. The table of values is located in Appendix C as C.1. When comparing the times of the RDU task with the GAS task, it is easily seen that the RDU task was a harder task taking roughly 6:45 longer to complete the task. This is interesting since the problems are abstractly the same. This could be explained by the perceived complexity of a parking lot management application when compared to a gas pump. This can be seen graphically in figure D.1 located in Appendix D.

More important to this experiment is the comparison of times sorted by the platform used. When using the DiamondTouch, the tasks took nearly four minutes longer than a CRC card session conducted without the DiamondTouch. At first look, this indicates that the DiamondTouch actually negatively affected user performance. However, the range of times for the tasks executed on the DiamondTouch was less than the range of times for the tasks conducted without the DiamondTouch. This suggests that the time for conducting a CRC Card session on the DiamondTouch is more consistent and thus easier to predict. This can be seen graphically in figure D.2 located in Appendix D.

The next set of numbers to be examined is the amount of time spent by each user talking. The table of values is located in Appendix C as C.2. The raw times that each user spent talking offer little insight. However the percentage of the total time spent talking offers insight. During the task conducted on the DiamondTouch, user one spent 20.9% and user two spent 23.5% respectively of the total time talking. For the task conducted without the DiamondTouch, these numbers rose to 28.6% for user one and 32.4% for user two of the total time. This suggests that verbal communication was reduced when using the DiamondTouch. This raises an interesting question: if verbal communication was reduced, was nonverbal communication reduced as well or increased. A possible explanation for this can be created around the action of adding information to an existing card. When doing this action without the DiamondTouch, a dialog may take place as one user asks the other what information is being added to the card. As information is added to a card, the other user cannot read the information from the card as the hand position blocks their view. In the same scenario with the DiamondTouch, cards are not covered when information is added to them because the keyboard does not cover the card. The other user can gain the information they need from the environment without having to engage in verbal communication, which may be slower, with the user adding information to the card. A scenario such as this would say nonverbal communication increased, but this cannot be denied or confirmed, although some user feedback indicated more nonverbal communication while using the DiamondTouch. This can be seen graphically in figures D.3 and D.4 located in Appendix D.

Another interesting insight can be seen when comparing the percentage of total time talking for user one against the percentage of total time talking for user two. The average percentage of user one talking when compared to the average time user two spent talking for the DiamondTouch was 90.8%. This value increases to 99% for the non-DiamondTouch tasks. Though these numbers are separated by roughly eight percentage points, they are close enough to suggest that user one's level of communication when compared to user two's level of communication was relatively the same across platforms. This suggests the roles each user assumed remained relatively constant across platforms. This can be seen graphically in figure D.5 located in Appendix D.

The last set of data to be examined is the time of activity for each user. The table of values is located in Appendix C as C.3. As with the time for each user talking, the raw time for each user's activity does not offer much insight. Instead, the percentage of total time that each user is active offers insight into the collaboration. User one's average percentage of active time for the DiamondTouch tasks was 34.6% compared to 26.4% for the non-DiamondTouch task. On average, user one spent more time during the task actively engaged in the design. For user two, the average percentage of acive time for the DiamondTouch tasks was 35.5% compared to 36.9% for the non-DiamondTouch tasks. On average, user two spent slightly less time actively working on the design. However, the range of data points was tighter for the DiamondTouch tasks. User two was more consistent with the amount of work done when using the DiamondTouch. This can be seen graphically in figures D.6 and D.7 located in Appendix D.

An interesting insight is seen when comparing the percentage of total time of activity for user one against the percentage of total time of activity for user two. The average percentage of time for user one engaged in activity compared to the average percentage of time user two spent engaged in activity was 96.5%. This value drops to 94.6% for the non-DiamondTouch tasks. An ideal value for this metric would be 100% indicating that each user was active the same amount of time. The DiamondTouch appeared on average to share the work slightly better users. The real interesting piece of this data is the range of the data points. The data has a much narrower range for the DiamondTouch than does the non-DiamondTouch data. This points to more consistent load sharing between users. Basically, all users carried their own weight in the group when using the DiamondTouch. This can be seen graphically in figures D.8 located in Appendix D.

Most of the values gathered are not statistically significant. They suggest trends that should be verified with more tests. The only statistically significant result is the duration to conduct each task. Different tasks should be used to eliminate this result.

8.5 Evaluation

The question of is the DiamondTouch better is a difficult question with no clear answer. The question of did the DiamondTouch perform at least par with the non-DiamondTouch tasks is equally difficult. It is a judgment call to both of these questions because there are benefits and costs for using this application that need to be evaluated on a per case basis.

The user feedback indicated that the software was difficult to use in certain ways. Particularly when manipulating text with keyboard. This was backed up when the total time comparisons are made with the percentage of activity for each user. Overall, the users spent about the same percentage of total time being active for the DiamondTouch and manual tasks. However, the DiamondTouch tasks took longer. Users were doing the same amount of work, but were taking longer. Something slowed down user performance on the DiamondTouch and this could be the keyboard.

Even with this negative impact on user performance, it was not a severe one. The question that needs to be answered is whether or not it is beneficial to have the design in an electronic format for the cost of nearly four minutes. To input a design into a computer would certainly take more than four minutes, but whether this is worth the cost is still an individual judgment call.

9. FUTURE WORK

During the feedback sessions with the users, many suggestions were discussed about how to make the system better. Some of these were viable changes that could be done and potentially improve usability, while many were not viable.

One idea is to remove the DiamondTouch from the software in favor of multiple keyboards and multiple mice. The display would be project on the wall and each user would use their keyboard and mouse to interact with the cards. An interesting benefit of doing something like this would be users could be located at different locations and still be able to conduct a CRC Card design session. As the system got bigger to support more users, it would become more difficult to track all the different user's pointers and to keep track of who is doing what. At MERL, this was considered as an alternative to building the DiamondTouch, but was dismissed because of this concern [1].

Don Norman, in his book "Things That Make Us Smart", discusses an updated airplane cockpit where the traditional controls are replaced trackballs and monitors. As it turns out, this negatively impacted the collaboration because the nonverbal communication between pilot and co-pilot was reduced. Rather than the pilot moving his whole arm to pull a leaver, which the co-pilot could see in his peripheral vision [3], the pilot would just move their fingers to accomplish the same goal. The co-pilot could not tell what the pilot was doing because there was nothing distinguishing about his actions and movement.

Another possible improvement to the system is to replace the soft keyboard with a real keyboard. This would certainly overcome the limitations of using a soft keyboard, but would cause the user to switch from using the DiamondTouch to using the keyboard and back. The users arms could be come tired from doing this.

The most viable of all the improvements to the system is to replace the soft keyboard with a stylist and handwriting to text translation engine. This would be a more natural interaction, as it is almost identical to the way a CRC Card design session would be done without the DiamondTouch. The idea of user modes could also be removed because there is no ambiguity as to what the user is doing when they are using a pen. However, implementing this would be difficult to do as the stylist would have to be custom made and the translation engine would have to be either written on an off the shelf version used. These engines are complicated and not readily available in an open source format.

The above changes have identified the keyboard as the problem area and offer alternatives to its use. Rather than replace it, it could be fixed. It is not clear how to fix this, or even if it can be fixed to be more usable, but it could greatly change the results of the experiment if it was no longer a troublesome tool for the user to use.

10. LIST OF REFERENCES

1. P. Dietz and D. Leigh. “DiamondTouch: A Multi-User Touch Technology.” UIST ’01 Nov 2001: pages 219-226.
2. N. Churcher and C. Cerecke. “GroupCRC: Exploring CSCW Support for Software Engineering.” n.d.: n. pag.
3. D. Norman. Things That Make Us Smart. Reading, Massachusetts, 1993.
4. S. Pekkola, M. Robinson, M. Saarinen, J. Korhonen, S. Hujala, and T. Toivonen. “Collaborative Virtual Environments in the Year of the Dragon.” CVE 2000 n.d.: pages 11-18.
5. T. Prante, C. Magerkurth, and N. Streitz. “Developing CSCW Tools for Idea Finding – Empirical Results and Implications for Design.” CSCW ’02 Nov. 2002: pages 106-115.
6. J. Rekimoto. SmartSkin: “An Infrastructure for Freehand Manipulation on Interactive Surfaces.” CHI 2002 April 2002: n. pag.
7. J. Shell, T Selker, and R. Vertegaal. “Interacting with Groups of Computers.” Communications of the ACM March 2003: pages 40-46.
8. R. St. Amant. "CSC 554: Human-Computer Interactions." Textiles, North Carolina State University. Spring Semester 2003.
9. B. Ullmer and H. Ishii. “Emerging Frameworks for Tangible User Interfaces.” Human-Computer Interaction the New Millennium Aug. 2001: pages 579-601.
10. Tutorial. http://www.csc.calpoly.edu/~dbutler/tutorials/winger96/crc_b/tutorial.html

11. Appendices

APPENDIX A: Screen Shots

Figure A.1: DiamondTouch CRC Application Iteration 1 Interface

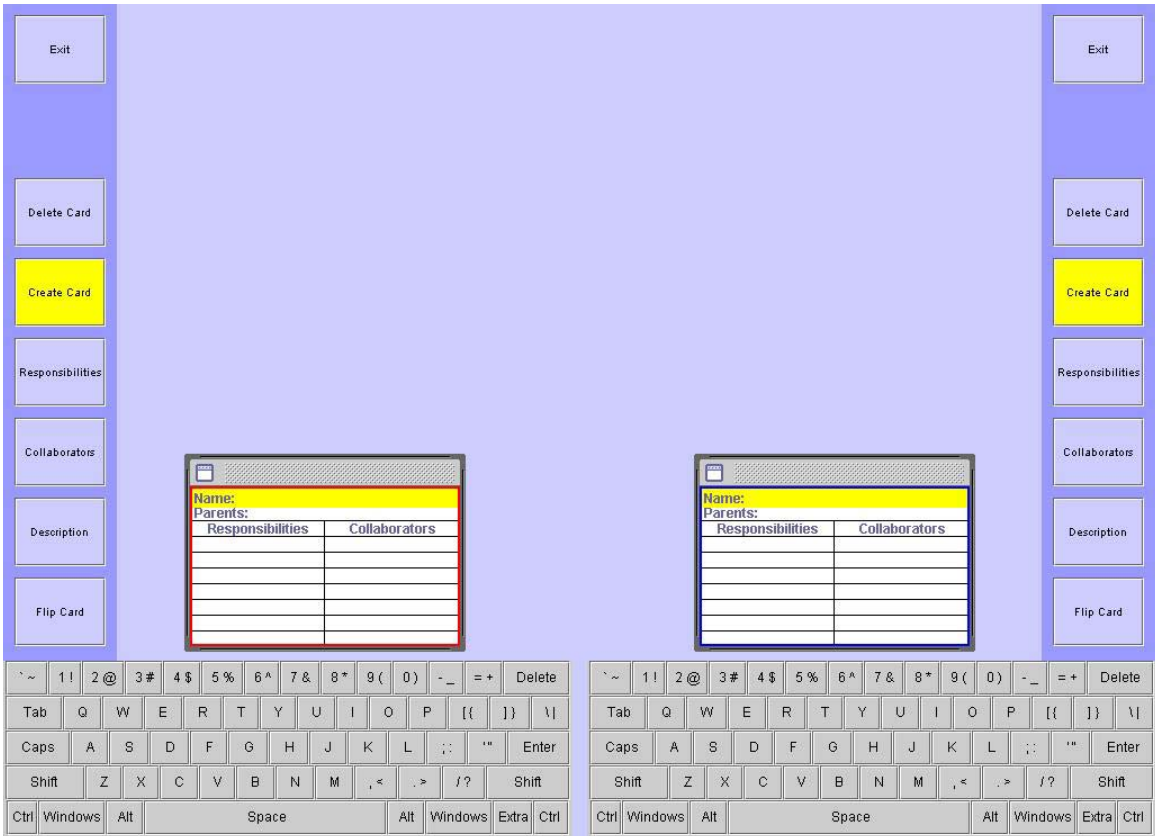


Figure A.2: DiamondTouch CRC Application Iteration 1 Interface:
first stage of creating card

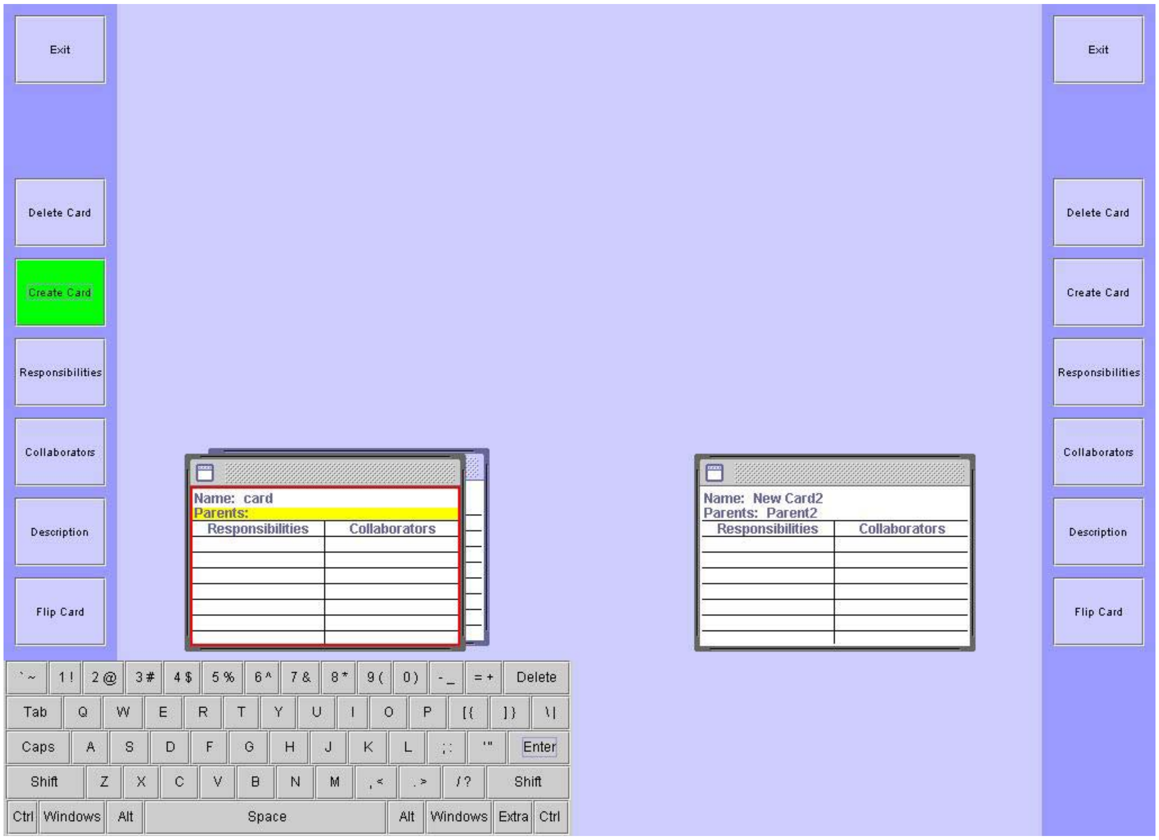


Figure A.3: DiamondTouch CRC Application Iteration 1 Interface: second stage of creating card

Figure A.4: DiamondTouch CRC Application Iteration 2 Interface

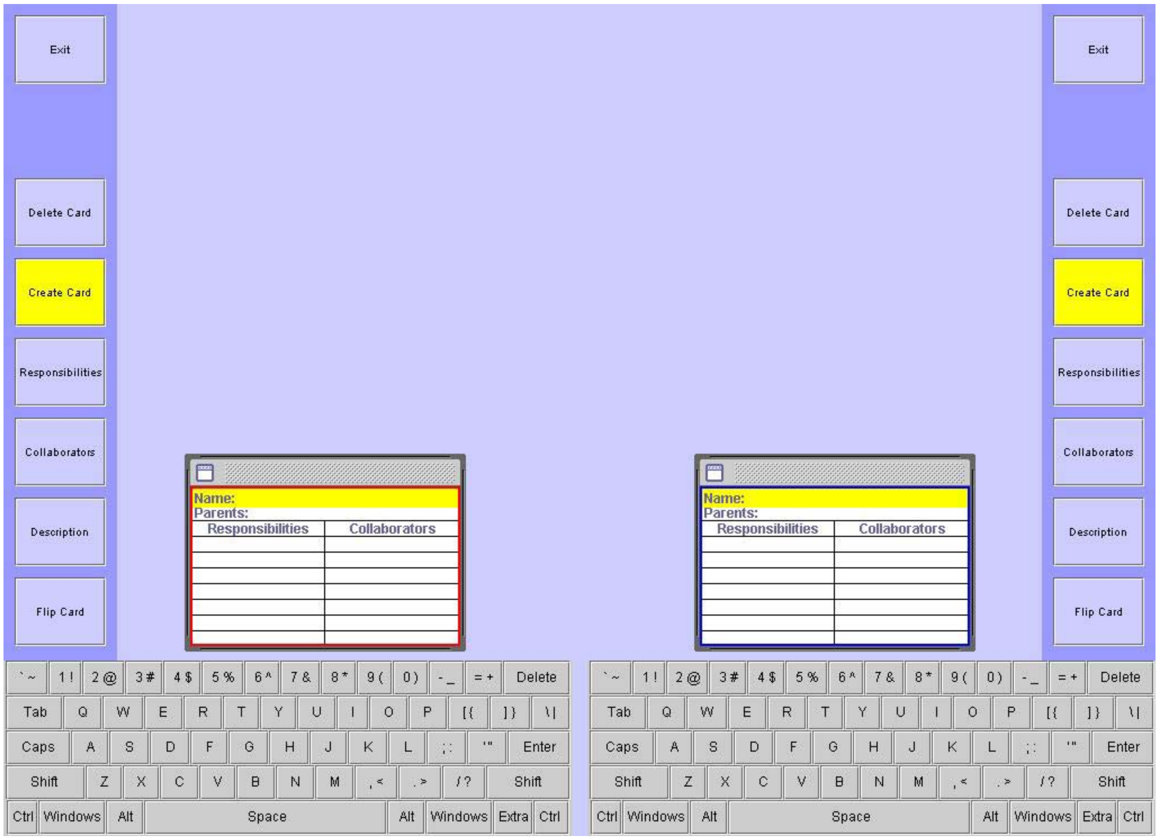


Figure A.5: DiamondTouch CRC Application Iteration 2 Interface with keyboards

Figure A.6: DiamondTouch CRC Application Iteration 3 Interface

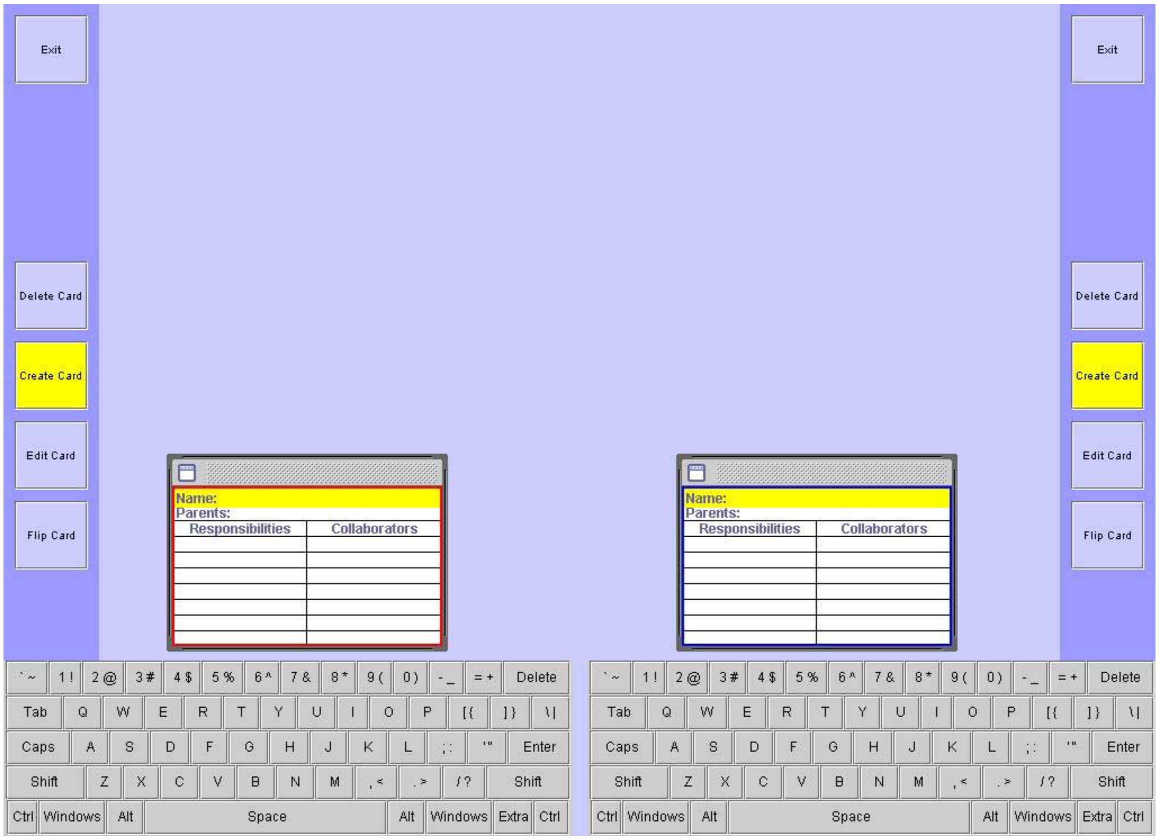


Figure A.7: DiamondTouch CRC Application Iteration 3 Interface with keyboards

Appendix B: Experimental Materials

10.2.1 Parking lot Management Task Description

RDU Parking Management Application

RDU airport, as part of their renovations is adding new parking lots and a new system for managing available spaces in each lot. The general contractor for the project has asked your company to provide the software that will manage these lots. The requirements have been provided to you:

- There is to be only one system that will manage all parking lots
- The system must keep track of the capacity and the number of available spots for each lot.
- Each lot is billed at a different rate based on proximity to the terminals. The rate of pay is billed in half-day increments. Time will be rounded up to the nearest half-day. If a vehicle is in a parking lot for 1.65 days, then this will be billed as 2 days.
- The drivers will specify which lot they would like to park in. If space is not available, the driver is to be informed that there is not enough room.
- In addition, drivers will be allowed to have trailers attached to their car. If the driver does have a trailer, then this will require 2 available parking spots, all other vehicles will only require 1 available parking spot.

    *do not be concerned with layout of parking spots. If 2 parking spots are available and a vehicle has a trailer, it is sufficient that the vehicle and trailer will fit leaving the lot full.
- When each vehicle enters a lot, the driver will be given a ticket specifying the lot being used, the time the vehicle entered the lot, if the vehicle had a trailer.
- Upon leaving the lot, the driver will return the ticked to a system kiosk, which will calculate the amount due and accept the payment. Upon full payment, the user will be allowed to leave the lot.

    *specifics of payment methods are not important.

Your task is to design a system using CRC cards that meets the requirements specified above. Implementation details are not needed, but details about what classes are to be responsible are.

10.2.2 Pay-at-the-pump Management Task Description

Pay-at-the-Pump Management

Veritas Gas, a startup automobile gas company, is designing a series of new stores to be built in the area. During the initial planning stages, the decision was made to have each gas pump operate without the need for employee intervention by being able to dispense fuel and accept payment for the fuel dispersed.

- Each pump must have the ability to disperse any of the fuels that the station has on hand.
- Each gas will cost a different amount, depending on quality and octane level
- The rate at which gas will be billed is per gallon.
- Fuel will only be dispersed as whole gallons. Partial gallons will not be sold.
- Each pump must make sure a full gallon exists for the specified fuel before dispensing. If the user has not started pumping, then the customer must be informed that the selected fuel is not available. If the customer has already started fueling and they used the remaining fuel, then the pump must stop.
- Each customer will select the type of fuel they wish to purchase.
- When the customer stops pumping, then the pump should calculate the amount due and accept payment. Upon full payment, a receipt should be printed for the customer.
- The store will have multiple pumps, but common gas tanks for the storage of the fuel. Each pump should be able to operate independently of the other pumps, but must share common storage tanks

  *Do not worry about synchronization issues that may arise

Your task is to design a system using CRC cards that meets the requirements specified above. Implementation details are not needed, but details about what classes are to be responsible are.

10.2.3 Participant Questionnaire

Are you male or female?

Have you ever work together/collaborated with your partner?

How many years have you been programming

Were you familiar with CRC Cards prior to this experiment?

Which did you like better, doing the experiment by hand or with the DiamondTouch?

Do you feel one system design was better than the other?

If so, which one and did the DiamondTouch play a role in making the design better or worse?

Was using the DiamondTouch easier or harder than doing it by hand, why?

Did using the DiamondTouch hinder your ability to collaborate with your partner?

Did the DiamondTouch draw your attention more to collaboration between you and your partner?

Appendix C: Tables

Table C.1: Task Time Analysis

	Group	Task	Platform	Task Order	Platform Order	Duration	Number of Cards	Total Number of Words
1	Group 1	Gas	DT	RDU/Gas	Hand/DT	945	5	90
2	Group 2	Gas	Hand	Gas/RDU	Hand/DT	678	3	72
3	Group 3	Gas	Hand	RDU/Gas	DT/Hand	594	4	48
4	Group 4	Gas	DT	Gas/RDU	DT/Hand	3131	8	109
5	Group 5	Gas	DT	RDU/Gas	Hand/DT	975	2	63
6	Group 6	Gas	Hand	Gas/RDU	Hand/DT	294	4	28
7	Group 7	Gas	Hand	RDU/Gas	DT/Hand	444	4	118
8	Group 8	Gas	DT	Gas/RDU	DT/Hand	1042	5	58
9	Group 1	RDU	Hand	RDU/Gas	Hand/DT	1077	6	173
10	Group 2	RDU	DT	Gas/RDU	Hand/DT	809	4	57
11	Group 3	RDU	DT	RDU/Gas	DT/Hand	1578	6	62
12	Group 4	RDU	Hand	Gas/RDU	DT/Hand	1198	6	132
13	Group 5	RDU	Hand	RDU/Gas	Hand/DT	1562	3	137
14	Group 6	RDU	DT	Gas/RDU	Hand/DT	1027	5	58
15	Group 7	RDU	DT	RDU/Gas	DT/Hand	981	4	81
16	Group 8	RDU	Hand	Gas/RDU	DT/Hand	693	5	60

Table C.2: User Talking Analysis

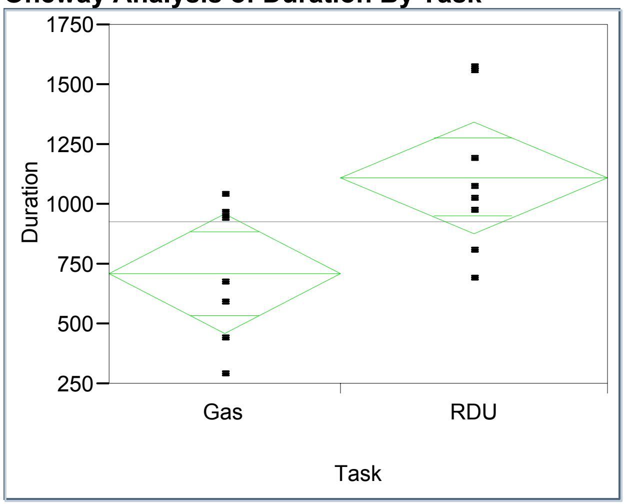
	Group	Task	Platform	User 1 Time Talking	User 2 time talking	User 1 Talking Percentage	User 2 talking Percentage	User 1 taking : User 2 talking
1	Group 1	Gas	DT	48	260	0.050794	0.275132	0.184615
2	Group 2	Gas	Hand	258	249	0.380531	0.367257	1.036145
3	Group 3	Gas	Hand	186	262	0.313131	0.441077	0.709924
4	Group 4	Gas	DT	1022	899	0.326413	0.287129	1.136819
5	Group 5	Gas	DT	66	148	0.067692	0.151795	0.445946
6	Group 6	Gas	Hand	121	84	0.411565	0.285714	1.440476
7	Group 7	Gas	Hand	92	60	0.207207	0.135135	1.533333
8	Group 8	Gas	DT	181	142	0.173704	0.136276	1.274648
9	Group 1	RDU	Hand	192	538	0.178273	0.499536	0.356877
10	Group 2	RDU	DT	303	225	0.374536	0.278121	1.346667
11	Group 3	RDU	DT	422	512	0.267427	0.324461	0.824219
12	Group 4	RDU	Hand	494	361	0.412354	0.301336	1.368421
13	Group 5	RDU	Hand	352	581	0.225352	0.371959	0.605852
14	Group 6	RDU	DT	235	283	0.228822	0.27556	0.830389
15	Group 7	RDU	DT	180	147	0.183486	0.149847	1.22449
16	Group 8	RDU	Hand	113	130	0.163059	0.18759	0.869231

Table C.3: User Activity Analysis

	Group	Task	Platform	User 1 activity Time	User 2 Activity Time	User 1 Activity Percentage	User 2 Activity Percentage	User 1 activity : User 2 activity
1	Group 1	Gas	DT	586	422	0.620106	0.446561	1.388626
2	Group 2	Gas	Hand	152	344	0.224189	0.507375	0.44186
3	Group 3	Gas	Hand	120	164	0.20202	0.276094	0.731707
4	Group 4	Gas	DT	968	813	0.309166	0.259661	1.190652
5	Group 5	Gas	DT	287	332	0.294359	0.340513	0.864458
6	Group 6	Gas	Hand	84	35	0.285714	0.119048	2.4
7	Group 7	Gas	Hand	169	251	0.380631	0.565315	0.673307
8	Group 8	Gas	DT	217	255	0.208253	0.244722	0.85098
9	Group 1	RDU	Hand	490	652	0.454968	0.605385	0.751534
10	Group 2	RDU	DT	369	392	0.456119	0.484549	0.941327
11	Group 3	RDU	DT	500	531	0.316857	0.336502	0.94162
12	Group 4	RDU	Hand	415	228	0.346411	0.190317	1.820175
13	Group 5	RDU	Hand	7	622	0.004481	0.398207	0.011254
14	Group 6	RDU	DT	239	309	0.232717	0.300876	0.773463
15	Group 7	RDU	DT	325	420	0.331295	0.428135	0.77381
16	Group 8	RDU	Hand	151	203	0.217893	0.292929	0.743842

Appendix D: Figures

Oneway Analysis of Duration By Task



Oneway Anova

Summary of Fit

Rsquare	0.335683
Adj Rsquare	0.284582
Root Mean Square Error	305.5755
Mean of Response	926.4667
Observations (or Sum Wgts)	15

t-Test

	Difference	t-Test	DF	Prob > |t|
Estimate	-405.34	-2.563	13	0.0236
Std Error	158.15			
Lower 95%	-747.00			
Upper 95%	-63.68			

Assuming equal variances

Analysis of Variance

Source	DF	Sum of Squares	Mean Square	F Ratio	Prob > F
Task	1	613386.4	613386	6.5690	0.0236
Error	13	1213893.3	93376		
C. Total	14	1827279.7			

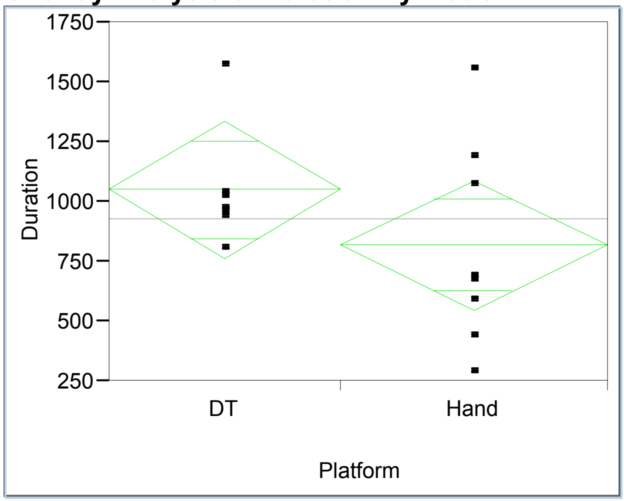
Means for Oneway Anova

Level	Number	Mean	Std Error	Lower 95%	Upper 95%
Gas	7	710.29	115.50	460.77	959.8
RDU	8	1115.63	108.04	882.22	1349.0

Std Error uses a pooled estimate of error variance

Figure D.1: Analysis of Time by Task

Oneway Analysis of Duration By Platform



Oneway Anova

Summary of Fit

Rsquare	0.111395
Adj Rsquare	0.043041
Root Mean Square Error	353.4152
Mean of Response	926.4667
Observations (or Sum Wgts)	15

t-Test

	Difference	t-Test	DF	Prob > |t|
Estimate	233.50	1.277	13	0.2241
Std Error	182.91			
Lower 95%	-161.65			
Upper 95%	628.65			

Assuming equal variances

Analysis of Variance

Source	DF	Sum of Squares	Mean Square	F Ratio	Prob > F
Platform	1	203549.7	203550	1.6297	0.2241
Error	13	1623730.0	124902		
C. Total	14	1827279.7			

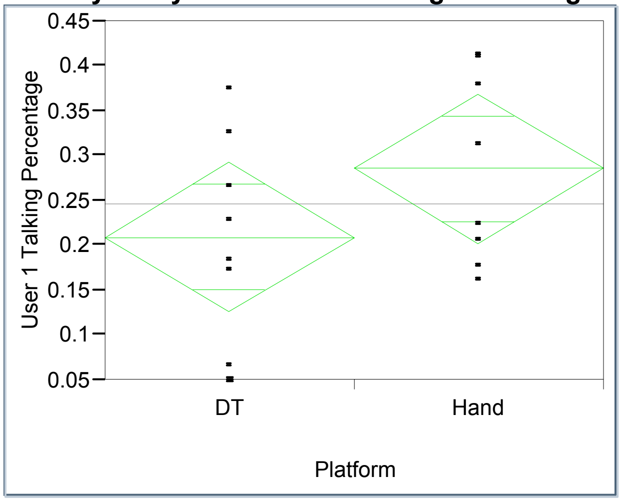
Means for Oneway Anova

Level	Number	Mean	Std Error	Lower 95%	Upper 95%
DT	7	1051.00	133.58	762.42	1339.6
Hand	8	817.50	124.95	547.56	1087.4

Std Error uses a pooled estimate of error variance

Figure D.2: Analysis of Time By Platform

Oneway Analysis of User 1 Talking Percentage By Platform



Oneway Anova

Summary of Fit

Rsquare	0.123399
Adj Rsquare	0.060785
Root Mean Square Error	0.110161
Mean of Response	0.247772
Observations (or Sum Wgts)	16

t-Test

	Difference	t-Test	DF	Prob > |t|
Estimate	-0.07732	-1.404	14	0.1822
Std Error	0.05508			
Lower 95%	-0.19546			
Upper 95%	0.04081			

Assuming equal variances

Analysis of Variance

Source	DF	Sum of Squares	Mean Square	F Ratio	Prob > F
Platform	1	0.02391639	0.023916	1.9708	0.1822
Error	14	0.16989711	0.012136		
C. Total	15	0.19381350			

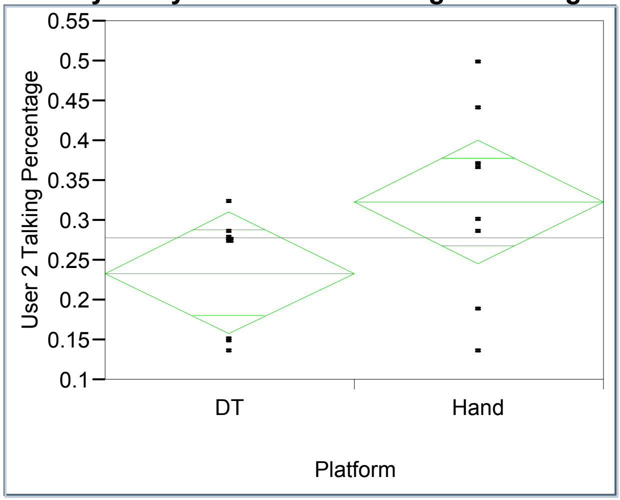
Means for Oneway Anova

Level	Number	Mean	Std Error	Lower 95%	Upper 95%
DT	8	0.209109	0.03895	0.12557	0.29264
Hand	8	0.286434	0.03895	0.20290	0.36997

Std Error uses a pooled estimate of error variance

Figure D.3: Analysis of Time User One Spent Talking by Platform

Oneway Analysis of User 2 Talking Percentage By Platform



Oneway Anova

Summary of Fit

Rsquare	0.17933
Adj Rsquare	0.120711
Root Mean Square Error	0.101666
Mean of Response	0.279245
Observations (or Sum Wgts)	16

t-Test

	Difference	t-Test	DF	Prob > |t|
Estimate	-0.08891	-1.749	14	0.1022
Std Error	0.05083			
Lower 95%	-0.19794			
Upper 95%	0.02012			

Assuming equal variances

Analysis of Variance

Source	DF	Sum of Squares	Mean Square	F Ratio	Prob > F
Platform	1	0.03162016	0.031620	3.0592	0.1022
Error	14	0.14470344	0.010336		
C. Total	15	0.17632360			

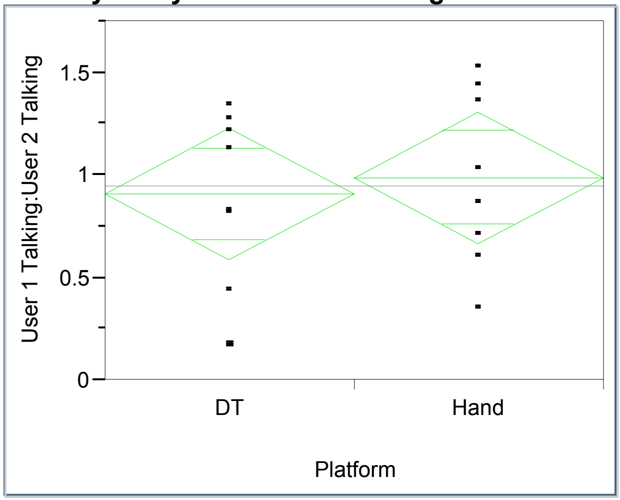
Means for Oneway Anova

Level	Number	Mean	Std Error	Lower 95%	Upper 95%
DT	8	0.234790	0.03594	0.15770	0.31188
Hand	8	0.323701	0.03594	0.24661	0.40079

Std Error uses a pooled estimate of error variance

Figure D.4: Analysis of Time User Two Spent Talking by Platform

Oneway Analysis of User 1 Talking:User 2 Talking By Platform



Oneway Anova

Summary of Fit

Rsquare	0.010472
Adj Rsquare	-0.06021
Root Mean Square Error	0.423769
Mean of Response	0.949253
Observations (or Sum Wgts)	16

t-Test

	Difference	t-Test	DF	Prob > |t|
Estimate	-0.08156	-0.385	14	0.7061
Std Error	0.21188			
Lower 95%	-0.53600			
Upper 95%	0.37289			

Assuming equal variances

Analysis of Variance

Source	DF	Sum of Squares	Mean Square	F Ratio	Prob > F
Platform	1	0.0266071	0.026607	0.1482	0.7061
Error	14	2.5141169	0.179580		
C. Total	15	2.5407239			

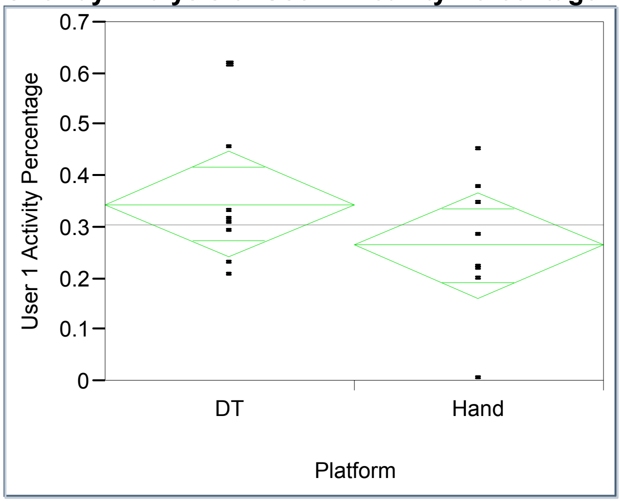
Means for Oneway Anova

Level	Number	Mean	Std Error	Lower 95%	Upper 95%
DT	8	0.908474	0.14982	0.58713	1.2298
Hand	8	0.990032	0.14982	0.66869	1.3114

Std Error uses a pooled estimate of error variance

Figure D.5: Analysis of Time User One Spent Talking Against Time User Two Spent Talking by Platform

Oneway Analysis of User 1 Activity Percentage By Platform



Oneway Anova

Summary of Fit

Rsquare	0.094044
Adj Rsquare	0.029333
Root Mean Square Error	0.135328
Mean of Response	0.305324
Observations (or Sum Wgts)	16

t-Test

	Difference	t-Test	DF	Prob > |t|
Estimate	0.08157	1.206	14	0.2480
Std Error	0.06766			
Lower 95%	-0.06355			
Upper 95%	0.22670			

Assuming equal variances

Analysis of Variance

Source	DF	Sum of Squares	Mean Square	F Ratio	Prob > F
Platform	1	0.02661503	0.026615	1.4533	0.2480
Error	14	0.25639052	0.018314		
C. Total	15	0.28300555			

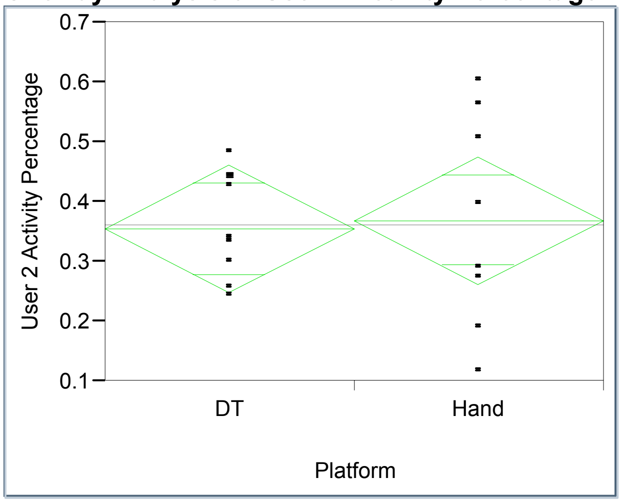
Means for Oneway Anova

Level	Number	Mean	Std Error	Lower 95%	Upper 95%
DT	8	0.346109	0.04785	0.24349	0.44873
Hand	8	0.264538	0.04785	0.16192	0.36716

Std Error uses a pooled estimate of error variance

Figure D.6: Analysis of Activity Time for User One by Platform

Oneway Analysis of User 2 Activity Percentage By Platform



Oneway Anova

Summary of Fit

Rsquare	0.002865
Adj Rsquare	-0.06836
Root Mean Square Error	0.141039
Mean of Response	0.362262
Observations (or Sum Wgts)	16

t-Test

	Difference	t-Test	DF	Prob > |t|
Estimate	-0.01414	-0.201	14	0.8439
Std Error	0.07052			
Lower 95%	-0.16539			
Upper 95%	0.13710			

Assuming equal variances

Analysis of Variance

Source	DF	Sum of Squares	Mean Square	F Ratio	Prob > F
Platform	1	0.00080022	0.000800	0.0402	0.8439
Error	14	0.27848625	0.019892		
C. Total	15	0.27928647			

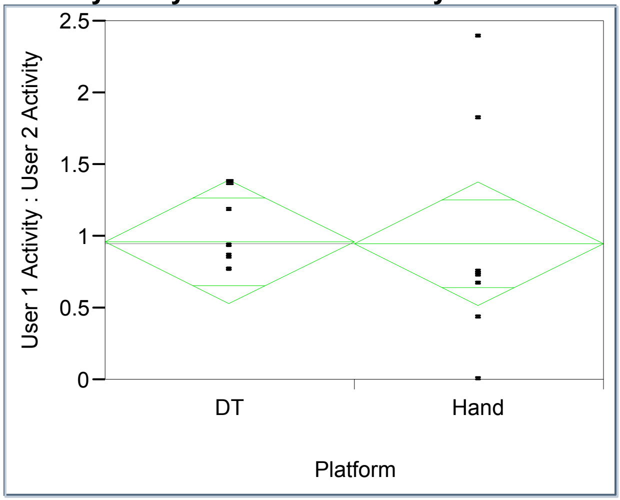
Means for Oneway Anova

Level	Number	Mean	Std Error	Lower 95%	Upper 95%
DT	8	0.355190	0.04986	0.24824	0.46214
Hand	8	0.369334	0.04986	0.26238	0.47628

Std Error uses a pooled estimate of error variance

Figure D.7: Analysis of Activity Time for User Two by Platform

Oneway Analysis of User 1 Activity : User 2 Activity By Platform



Oneway Anova

Summary of Fit

Rsquare	0.000315
Adj Rsquare	-0.07109
Root Mean Square Error	0.569022
Mean of Response	0.956163
Observations (or Sum Wgts)	16

t-Test

	Difference	t-Test	DF	Prob > |t|
Estimate	0.01891	0.066	14	0.9480
Std Error	0.28451			
Lower 95%	-0.59131			
Upper 95%	0.62912			

Assuming equal variances

Analysis of Variance

Source	DF	Sum of Squares	Mean Square	F Ratio	Prob > F
Platform	1	0.0014299	0.001430	0.0044	0.9480
Error	14	4.5330001	0.323786		
C. Total	15	4.5344300			

Means for Oneway Anova

Level	Number	Mean	Std Error	Lower 95%	Upper 95%
DT	8	0.965617	0.20118	0.53413	1.3971
Hand	8	0.946710	0.20118	0.51522	1.3782

Std Error uses a pooled estimate of error variance

Figure D.8: Analysis of Activity Time for User One Against Activity Time for User Two by Platform