THE UNIVERSITY OF NEW HAVEN

TAGLIATELA COLLEGE OF ENGINEERING

Department of Electrical & Computer Engineering and Computer Science

TANGIBLE USER INTERFACE FOR A VIRTUAL SCIENCE LABORATORY

A THESIS

Submitted in partial fulfillment

of the requirements for the degree of

MASTER OF SCIENCE IN ELECTRICAL ENGINEERING

BY

Frank Pellicano

University of New Haven

West Haven, Connecticut

July 2017

TANGIBLE USER INTERFACE FOR A VIRTUAL SCIENCE LABORATORY

BY

Frank Pellicano

A THESIS

Submitted in partial fulfillment

of the requirements for the degree of

MASTER OF SCIENCE IN ELECTRICAL ENGINEERING

APPROVED BY

Christopher Martinez, Ph.D
Thesis Advisor

Ali Golbazi, Ph.D
Department Chairperson

Bijan Karimi, Ph.D
Committee Member

Ronald Harichandarn, Ph.D
Dean of TCoE

Office of the Provost Administration
University of New Haven

ACKNOWLEDGEMENTS

I would like to thank my supervisor, Dr. Christopher Martinez, for offering his unending wisdom and encouragement throughout the entire thesis process. With his support and guidance, I was able to develop an effective time-management plan to ensure I did not become overwhelmed with the rigors of a thesis, working two jobs, attending and succeeding in my courses, and still have time to enjoy with family and friends. His mentorship will be invaluable to my long career in the engineering world.

I would like to thank Dr. Bijan Karimi, Dr. Ali Golbazi, and Dr. Bouzid Aliane for their suggestions and advice on the hardware portion of my thesis, as well as advising on the concept.

I would like to thank Dr. Tiffany Hesser and Mr. Richard Weber for their assistance on the chemistry portion of my thesis. Dr. Hesser's advice on improving a chemistry laboratory and Mr. Weber's assistance in executing the labs were invaluable.

I would like to thank Mr. Austin Matthews for his assistance with the mechanical portion of my thesis, as well as allowing me to use his personal 3D printer.

I would like Mr. Mark Morton for ensuring the lab equipment was available and open for me, as well as providing the components necessary in my hardware prototype stage.

Finally, I would like to thank my parents for their support, love, and guidance through my entire graduate studies, as well as all of my family members and friends who have supported me.

ABSTRACT

As society grows more and more technologically advanced, new methods of human-to-machine interaction are being developed. The 21st century has seen an exponential growth of Tangible User Interfaces (TUIs) as the primary mode of interaction between machines. TUIs appear in a variety of settings, from research labs, museums, video games, or even specific work environments. This thesis is an electronic alternative to the traditional chemistry laboratory, leveraging four TUI devices that are designed to mimic the look and feel of traditional chemistry equipment. These cubes interface with a virtual laboratory manual that guides the user through the various steps of completing a chemistry experiment. Advantages of having a virtual chemistry lab are plentiful. Traditional chemistry experiments require intimate knowledge of the equipment, while a virtual lab would not require that knowledge, making it easy to use. Additionally, the virtual world is not prone to the errors seen in the real world, such as equipment being poorly calibrated or element measurements being inaccurate. Virtual chemistry also removes danger such as toxic fumes or simultaneous combustion, as well as rapidly speeding up the time needed to complete a full experiment. The future of society is virtual reality, and this thesis is a stepping stone to a virtual chemistry lab where the concept of these cubes and portal will be pivotal in that future.

TABLE OF CONTENTS

Contents

LIST OF FIGURES

LIST OF TABLES

LIST OF ABBREVIATIONS

TUI – Tangible User Interface

RGB – Red, Green, Blue

LED – Light Emitting Diode

GUI – Graphical User Interface

GPIO – General Purpose Input / Output

LCD – Liquid Crystal Display

PCB – Printed Circuit Board

IR – Infrared

KT – Kepner Tregoe

1. CHAPTER I: Introduction to the Problem and its' Background

1.1. Overview and Purpose

This thesis will focus on virtualizing a chemistry lab to make the subject less difficult to learn. Across the United States, 49.0% of AP students failed to receive a satisfactory grade to qualify for AP Chemistry credit in 2016. In comparison, 19.4% did not qualify for AP Calculus BC credit and 36.2% did not qualify for AP Biology credit [1]. Modern chemistry laboratories utilize antiquated equipment. This could be a potential reason for the disproportionate quantity of students struggling to succeed in chemistry.

Current chemistry labs are filled with a variety of equipment such as Bunsen burners, beakers, pipettes, and chemical elements. In a virtual chemistry lab, of this equipment would be replaced by tangible objects that will map into virtual reality. For example, one tangible object could be a 500 mL beaker. The virtual beaker is an electronic cube with various electrical sensors such as a gyroscope and a color sensor. This electronic cube would then interface with a virtual reality world where the graphics are of a 500mL beaker. Based on how the user interacts with the electronic cube in the real world, the virtual beaker will experience the same effects in the virtual world. Not only will this modernize the classroom, it will allow students who are not chemistry majors to avoid the frustration and dangers associated with the difficulties of the lab environment.

1.2. Thesis Objective

The goal of this thesis was to develop the methodology for a modern and effective method of learning and interacting in a science and engineering classroom environment by leveraging a tangible user interface. This was done by developing interactive cubes that tie to an instructional

LCD display. The TUI could be used for an educational subject. In particular, the test subject chosen to demonstrate the TUI methodology was chemistry. Ideally, the technology created would be a low cost solution for wide adaptation in a variety of markets.

In particular, four TUI cubes were developed, each representative of a real component in a chemistry lab. Cube 1, titled the Dropper Cube, simulates equipment that transports elements from one location to another, such as a dropper or crucible tongs. Cube 2, titled the Burner Cube, simulates emitter equipment such as a Bunsen burner. Cube 3 and Cube 4 are mirrors of each other, both representing equipment that can contain elements, such as a beaker or Petri dish. Each cube is shaped to appear similar to their real life counterpart.

Parallel to the cubes development was the creation of a virtual lab manual on an LCD display to guide the user through a full chemistry lab. This virtual portal informs the user on the driving chemistry theory which makes the lab possible. The portal also shows how to manipulate the electronic cubes to achieve the lab's desired outcome. After each step of the lab is successfully completed, the steps' particular reaction is displayed on the screen from a pre-recorded video, tying the virtual chemistry to the real world.

Finally, a pre and post survey were created and administered to participants in testing the project. The goal of the pre survey was to explain what the participant is about to partake in and gather what their thoughts and expectations are on the subject of virtual education. The goal of the post survey was to acquire their feedback on how the experience of virtual chemistry felt, as well as any constructive criticism. If participants felt their experience with the device was positive and felt it was easier to learn chemistry virtually, then the percent of students failing chemistry would decrease.

1.3. Significance of Thesis

Education is imperative to spread knowledge within society and advance the technology that makes everyday life easier, enjoyable, and convenient. On a small, short term scale, the importance of this thesis is that it empirically studies, tests, and demonstrates that there are easier and safer alternatives to learning and experiencing chemistry. A virtual chemistry lab is a safe, convenient, and error free way to experiment with different elements.

On a grand scheme, the concept of a virtual reality education can apply to any field of study. Museums could adapt the technology as a way to interact with human anatomy. Astronomers could adapt the technology as a way to explore the universe. Additionally, the technology could be external to the traditional classroom environment and be sold in stores as an educational, virtual reality toy for any applicable subject. It would provide instant feedback, be completely safe, and be an effortless, error free, and fun way to learn.

1.4. Pre-Assumptions

Before embarking on the journey of virtual laboratory development, pre assumptions had to be laid out. First, there are no electronic chemistry labs that utilize a tangible user interface to perform the experiment and convey the information necessary to learn chemistry. That is to say modern chemistry labs do not conduct experiments by interacting with a virtual lab manual, but rather use traditional methods developed hundreds of years ago. Second, the user has any level of chemistry experience and can utilize basic motor skills to follow the on-screen instructions to perform the labs. Real chemistry equipment can have a difficult learning curve to use properly and safely which can cause unnecessary stress and challenges when performing the labs. Third, the user is willing to learn chemistry virtually. The target audience is not those who shy away from innovation or change, but rather embrace it. Fourth, the concept will remove the danger associated with traditional chemistry labs. For instance, glass equipment such as beakers are prone to break spontaneously, or certain elements result in explosive reactions that can be too dangerous for traditional chemistry environments. Fifth, the concept will remove the errors found when performing traditional chemistry. There will be no human error, such as misreading a measurement, no static error, such as a pipette constantly leaking, or random error, such as humidity changes due to the weather.

2. CHAPTER II: Literature Review

2.1. Tangible User Interfaces in Museums

Although the concept of using a tangible user interface scholastically is fairly new, there have been a variety of related projects. One example is *Using Hand Held Devices and Situated Displays for a Collaborative Planning of a Museum Visit* [2] . The concept was to create a tangible user interface within a museum to enhance group interaction. The individuals would use a handheld device and a museum visitor's guide to interact with a situated display located at the entrance of the museum. The user first uses the handheld device to rank their interests from most to least.

Figure 1: Image of the handheld device and application it shows.

Then, the user walks to the screen and on their handheld device a pre-planned tour would appear. A group of tour guests can link together wirelessly, allowing the pre planned tour to be the same for the entire group. Users can gain access to the large display to show the entire group their tour plan. They found the concept was easy to adapt to and enhanced visitor satisfaction for they were able to easily plan and re-plan their visit as they saw fit.

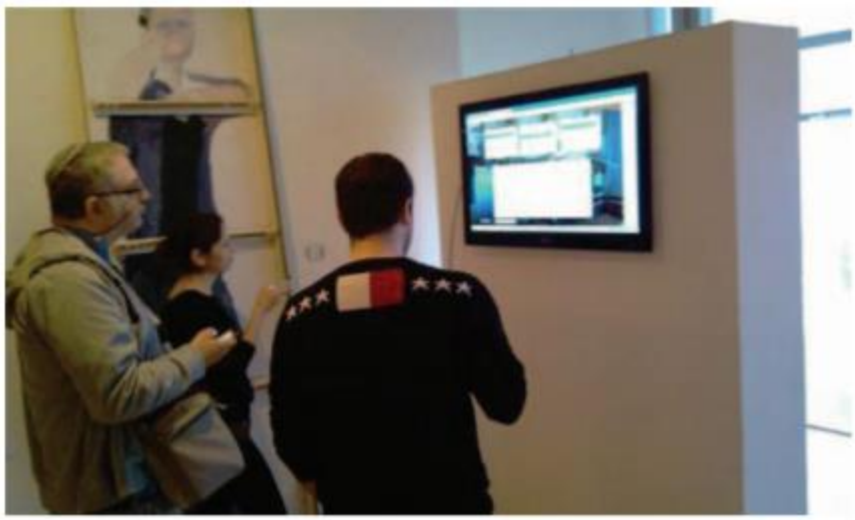


Figure 2: Image from the publication showing users holding the device by the screen.

Museums are constantly evolving as technology improves to enhance how knowledge is shared. Conveying historical lessons in chemistry or other related fields could benefit from using the TUI developed here. Curious minds could interact with the cubes shaped to emulate the equipment used by Louis Pasteur (1822–1895) or Robert Boyle (1627–1691). A level of engagement never before seen in museums could be realized to connect modern day society with those from the past, all by leveraging TUI's.

2.2. Machine interface with Tangible User Interfaces

Another example of previous work in this area was *Tangible Menus and Interaction Trays: Core Tangibles for Common Physical/ Digital Activities* [3]. The group that worked on this developed objects that could be interacted with physically that had multiple functions in order to allow interaction between a variety of tangible and embedded interfaces. In particular, the group focused on interactive menus and interaction trays. These trays connect to a device such as a computer. Each tray has various sensors which perform a particular action. If a user rotates a tray clockwise, it could perform a save action. If they rotate counter clockwise, it could perform a load action, etc. Early on, they found the experience was a positive step forward in menu interaction.

Figure 3: Two participants use the tangible interaction trays to load data, save data, manipulate parameters, etc.

Modern manufacturing requires one or more operators to run the machines that produce every day products. As automation replaces manual labor, more than ever operators need a method of human to machine interfacing that is safe and reliable. Using a TUI could allow an operator to stay out of harms' way by distancing themselves from the machine but still being able to interact with it.

2.3. Wii Gesture Remote

Similar to the previous example is *Gesture Recognition with a Wii Controller* [4]. This group recognized that user interaction was moving away from mouse and pens and instead was becoming pervasive and much more physical and tangible. Human machine interaction was becoming more intuitive, and this paper focused on developing gesture recognition with a Wii controller, as shown in Figure 4, to interact with any device.

Figure 4: Side, front, and rear view of the Wii controller.

In 2008, this was ahead of its time for gesture recognition and was not as commonplace as it has become today. The group used the Wii remotes accelerometer to develop a library of arbitrary gestures, titled Square, Circle, Roll, Z, and Tennis, as seen in Figure 5, for their devices to recognize. Each gesture would correspond to a command in the machine interface.

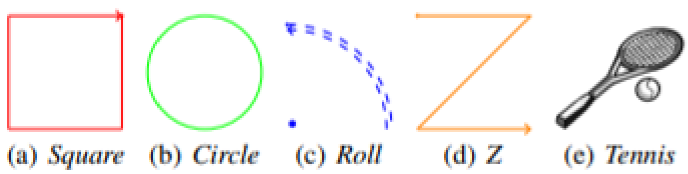


Figure 5: Gestures developed for the Wii Remote. The Tennis gesture requires the user to wave their hand back and forth.

Due to the large volume of data being read from the Wii remotes' accelerometer, the team elected to use a Baum-Welch algorithm for data interpretation in a Hidden Markov Model. This was required to ensure the gestures being performed were accurately detected. Upon development, they found the tangible user interface via Wii remote was a positive experience. In one instance, they were able to swipe their hands across a TV using the tennis action in order to view photos. This system founded the basis for their work on intuitive media browsing.

Building on this concept, using a TUI for chemistry would require gestures to be accurately measured and implemented to simulate performing chemistry. Taking from this research, the cubes could be similar in function to the remote. The remote has accelerometers and pushbuttons for user feedback. The accelerometer also requires filtering and data analysis techniques which will be required in the cubes simulating chemistry.

2.4. Tangible Music Composition

A United Kingdom publication titled *Audio D-Touch: A Tangible User Interface for Music Composition and Performance* [5] dives into using TUI's for music. The team utilized a camera and customizable blocks to provide users with an interactive tangible interface for time based musical tasks. Having movable objects allows for real time mapping and development of music.

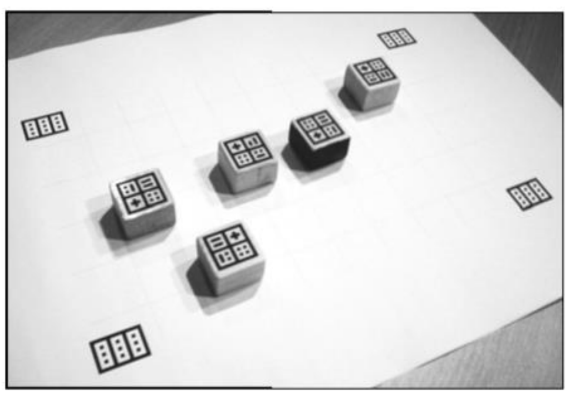


Figure 6: Example of the system with interactive blocks on top of a base board.

Each object has a pattern which is detected by the camera, which sends this data to be processed by a computer. Based on the quantity, orientation, and placement on the base board, a certain sound will be played. Moreover, the closer the objects are to one another, the less time there will be between sounds. This allows the user to create complex compositions or sounds, even if it would not be possible by human hands. Application wise, the team felt this could reach a wide variety of audiences. For example, children from an early age could create music and train their ears before they are even able to pick up a pen.

Although not utilized in this report, the concept of a vision system to detect the cubes is intriguing, for it solves many issues of spatial awareness. Without a camera, it is quite difficult to determine where in the real world objects are or where they have moved to. Unlike with the Wii remote, which used one tangible object, this project uses many objects. The chemistry lab also has many objects, which requires many devices to keep track of.

2.5. Cognitive Cubes

Cognitive Cubes: A Tangible User Interface for Cognitive Assessment [6] was a study to develop a better way to assess patients' cognitive abilities, such as problem solving or reading. Conducting cognitive research in clinical diagnosis requires assessing spatial and constructional ability in a patient. This is an inherently difficult task study due to a limited way of scoring and administering this research. The team developed cubes to create tangible method to improve flexibility, reliability, and control in cognitive testing.

Figure 7: Participant using the developed cognitive cubes.

Figure 7 shows a participant using developed cubes. The objective of this project was to develop a range of tests for the participant to perform in order to test their cognitive ability. They tested their three dimensional cubes against two dimensional tasks. Their findings were that being able to hold a device, such as a cube, and using it to perform various tasks, resulted in less error and higher overall performance. Taking from this study, building a virtual chemistry laboratory for educational purposes would significantly benefit with a tangible aspect, such as a set of cubes, to facilitate learning. The cognitive cube team felt the concept could be improved by having more polished tasks and more concise definition of task difficulty. Chemistry is a very specific field of study and lab difficultly can be measured by how difficult it is to comprehend and complete.

2.6. Holding a Tangible User Device

The next study, titled *HandSense – Discriminating Different Ways of Grasping and Holding a Tangible User Interface* [7], researched the most effective ways to use a tangible device, such as a cell phone or a cube. To do so, the team developed their own tangible device, called HandSense, which employed capacitive sensors to detect when and where the device was held and touched by a body part. The device could detect where it was being held, picked up, and adjusted in the users hands, up to 80% accuracy.

Figure 8: Concept drawing of different interactions the HandSense device can distinguish.

Figure 8 shows a concept drawing of three methods users implored. They were holding up, holding down, grasping right, grasping left, holding left, and holding right. Although holding the device with the entire hand was more common then grasping with fingers or holding strictly upright or down, the team found that users hold tangible devices in a variety of ways. They noted that detecting handedness of a user was important to enhance the interaction with the device to ensure maximum comfort and control. The device should also be capable of being grasped in a variety of ways without hindering function. These lessons are all taken into consideration with the design of the cubes to perform tangible chemistry. Cubes should be designed to be ambidextrous, such that any sensors are equally capable of being used by any hand. They should also be able to be held in a variety of ways.

2.7. Tangible Data Interaction Using Tokens

The next topic in the literature review is from the publication titled Tangible Interaction with Large Data Sets using Active Tokens [8]. This publication is about creating a tangible method of exploration of large data sets. The team used programmable objects, each with a display, actuators, and sensors, integrated into it, to explore large data sets. For example, the team explores a data set of a physical DNA model. Traditionally, button clicks and menu selections are used to explore this data type, however the team found this method ineffective. By using the tangible tokens to manipulate the data, the team found that by linking motor, perceptual, and cognitive skills together, the data was easier to traverse and interpret.

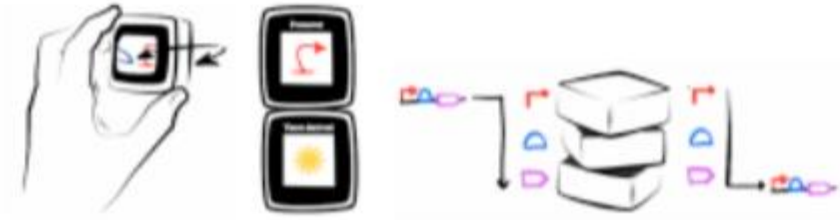


Figure 9: Conceptual design of the tangible token with various techniques for data browsing: (left) tilting search for traversing hierarchical databases, (right) stacking to collapse and expand.

As seen in Figure 9, the conceptual design allows the users to perform various actions to traverse a dataset. The sensors on the device detect tilting actions which map to scrolling through a hierarchical database. The tokens can also be stacked to expand or collapse data. The team sites that with technological advancements, such as smart watches and small robots, this technology will allow for tangible interaction techniques in various data intensive applications, allowing the user to be more connected to the machine and better understand and traverse large databases.

2.8. Chemist App

Finally, there is a cell phone application produced by THIX Development called *Chemist* [9]. The application is a game that is used for educational purposes of learning chemistry. The app acts as a virtual chemistry lab. Upon launch, the main portal is a virtual chemistry table with various instruments such as beakers, pipettes, ring stands, and more. Using their fingers, the user is able to manipulate these objects as needed. Once the equipment is set, the user is then prompted to begin mixing chemicals from a periodic table of elements. While mixing, the app will show the reaction in real time. The app will also show mistakes. For example, applying too much heat to a test tube will cause the glass to shatter. Figure 10 shows an example of the various equipment and elements set up in the app.

Figure 10: Example of various equipment and elements available in the Chemist App.

Unfortunately, the app suffers from having a counter-intuitive method of interacting with the various pieces of equipment. The user must place their finger on the screen and will often struggle to be accurate in their interactions. This requires precise finger dexterity that the casual user will not have. Additionally, since the user does not have anything physically in their hands, the experience feels one dimensional.

Despite its limitations, the interface is very similar to an ideal interface for the virtual chemistry portion of the tangible user interface for virtual chemistry. To improve upon this, the objects would have to be mapped to their own cubes. The user would then use these cubes to perform actions inside the app.

3. CHAPTER III: Methodology

3.1. Grand View

The theory of the design is to utilize devices that are tangible and electronic in order to interact with a virtual simulation of the real world. In the virtual simulation, elements of the real world would be present and controlled by those electronic devices. Each device can take on multiple roles and be mapped to different real world elements. The sensors on the devices allow the user to perform various actions in the real world, resulting in an identical action in the virtual world. The virtual world can be presented in a flat, two-dimensional environment or in an immersive three-dimensional environment.

Application wise, the tangible objects can be used to learn based on the virtual environment presented. For example, the environment presented could be an operating room, and each device would map to various surgical instruments [10]. One object could be a scalpel while another could be scissors. A medical student or curious persons could use this tangible experience to perform a virtual autopsy as a method of learning about the intricacies of the human body. It would start with the user sitting at a computer screen or wearing holographic projection glasses. In front of them would be the various devices, such as the scalpel or scissors. Instructions presented in the virtual world tell the user how to utilize the devices to achieve an objective, such as locating, identifying, and removing an organ. This will allow the user to see how a surgery is performed, how the tools are used, and learn the necessary steps that they could apply to real life.

3.2. Tangible Requirements

For this thesis, the application environment is a chemistry lab. To make this a reality, certain criteria of a chemistry lab must be met. That criterion is as follows.

1. Elements from the periodic table
2. Equipment to hold these elements
3. Equipment to transport these elements
4. Equipment to interact with these elements
5. Visible reactions based on the element interaction

All five of these criterions must be created as tangible elements to perform chemistry electronically. The user should feel that using the electronic alternative is natural and each element should connect to the real life counterpart. For instance, manipulating the equipment should feel natural virtually as in real life. In other cases, the experience should feel better than in reality. In a real lab, elements are held in containers, jugs, and bottles. They come in 3 different states; solid, liquid, or gas. This can be cumbersome for the user to both store and manipulate. Thus, a better solution should be presented. For this thesis, the focus shall be to develop these 5 elements of a chemistry lab into a practical and effective method to electronically perform chemistry, utilizing a tangible user interface.

4. CHAPTER IV: Prototype Design and Implementation

4.1. Proposed Experiments

With the methodology laid out and goal in place, chemistry experiments were chosen to design for. The first experiment was creating table salt, also called sodium chloride, by titrating sodium hydroxide with hydrochloric acid. Additionally, phenolphthalein is used as an indicator in the acid based titration to show the mixture is correct [11]. The chemical formula, as seen in equation (1), shows that the byproduct is sodium chloride and water.

$$HNaO + H_3ClO \rightarrow NaCl + 2H_2O \tag{1}$$

To remove the water and extract the salt, the water can be converted to gas through boiling. The melting point of salt is 801˚C, while the boiling point of water is 100˚C. As a result, the salt is unaffected by the exothermic reaction. This lab was chosen for its clear visual byproduct, as well as to demonstrate that a difficult procedure such as titration can be completed using TUI. Connecting to the five focus areas for this thesis, the elements required are hydrochloric acid, phenolphthalein, and sodium hydroxide. In the real world, the experiment uses a beaker and a burette to contain the elements. Transportation wise, one beaker receives two elements, while the burette receives another. Then, interaction between the containers is pouring the contents of the burette into the beaker. Finally, the solution is boiled, with the byproduct of the reaction being salt.

Another experiment is the reaction between potassium *(K)* and water *(H_20).* In a normal lab setting, this experiment is considered too dangerous for non-chemistry students to attempt. Chemistry students are advised to take extreme caution during this experiment. To do this experiment properly, a precise portion of potassium is placed into a Petri dish of water. The reaction shown in (2) results in a powerful explosion [12]. Simulating this virtually removes the element of danger from the experiment, allowing the user to focus on learning the chemistry behind the reaction.

$$2K + 2H_20 \rightarrow 2KOH + H_2 \tag{2}$$

The final experiment chosen was a classic chemistry trick; turning a copper penny into silver, then into gold. The copper penny is placed into a solution of zinc sulfate with zinc metal. Combining zinc and copper results in a disproportionate oxidization-reduction reaction (redox) [13], as seen in equation (3).

$$Zn + Cu \rightarrow Zn^{2+} + Cu^{2-} \tag{3}$$

The reduced copper then reacts with the zinc sulfate, resulting in a byproduct of silver zinc extracted from zinc sulfate, as seen in (4). The sulfate is also reduced.

$$Cu^{2-} + ZnSO_4 \rightarrow Cu + Zn + {SO_4}^{2-} \tag{4}$$

This reduced sulfate reacts with the oxidized zinc metal to form Zinc Oxide. This zinc is bonded to the copper temporarily. When heating the zinc oxide that is residing on the penny, it loses oxygen resulting in a golden yellow color, as seen in (5).

$$ZnO + heat \rightarrow Zn^{2+} + \frac{1}{2}O_2 \tag{5}$$

4.2. Conceptual Design

As seen in 4.1, each experiment requires a few pieces of equipment, as well as various elements. Figure 11 shows the block diagram of the conceptual design, leveraging five cubes and one control system. Initially, the five cubes would represent three different methods of containing elements, a transportation element, and an emitting element. The thought was that these cubes would contain various sensors to allow numerous modes of interaction between the user and the LCD display. Additionally, the cubes would have some type of indicator attached to them to allow the user to see their reactions without seeing the LCD display. The ‘brains’ behind the thesis would be the Raspberry Pi; chosen for its’ versatility and expansive libraries.

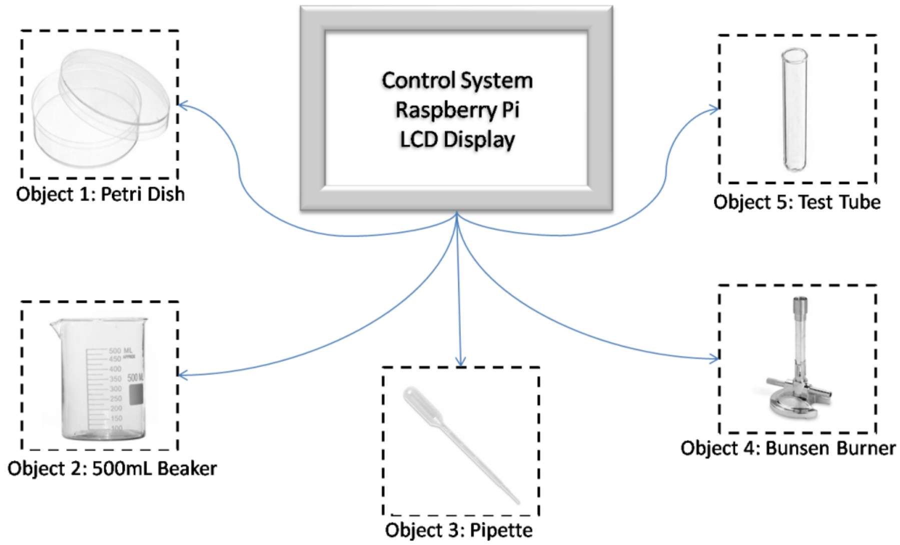


Figure 11: Block Diagram showing conceptual design of the chemistry set.

Numerous methods of communication between the cubes and the Raspberry Pi were explored. These methods were wireless communication through a Wi-Fi or Bluetooth module, using a USB data line, or point to point serial communication. Between the three, point to point serial communication was chosen for a variety of reasons. Keeping in mind that the objective was to develop a proof of concept, wireless communication would require more development time to realize. Using a USB data line would require each cube to have a microcontroller to manage any inputs and outputs from the cube. Figure 12 shows a focused block diagram with theoretical ideas for sensors each cube would have. The purpose and use of these sensors is detailed in section 4.3. Each cube would need to support approximately twelve inputs and outputs.

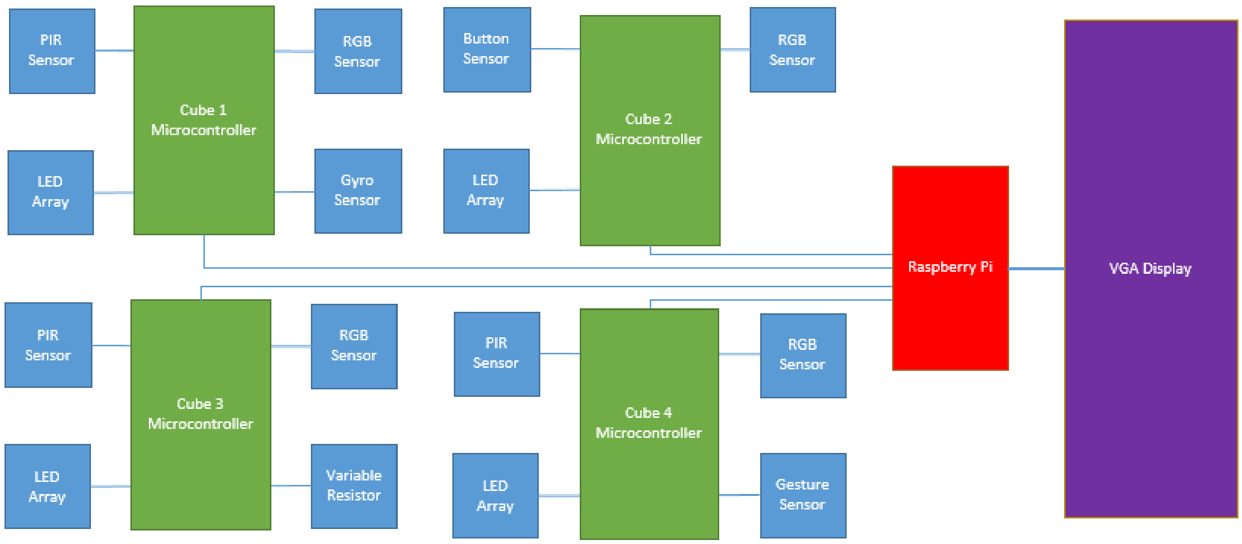


Figure 12: Focused block diagram of thesis system.

While using a USB data line was an interesting thought, having to include a microcontroller per cube would increase the cost per device, as well as increase development time for additional programming. Thus, the third option, point to point serial communication, was chosen. Unfortunately, the Raspberry Pi utilizes fourteen general purpose inputs and outputs, or GPIO [14], which is not enough. In this case, the Arduino MEGA microcontroller was chosen for it supports 54 GPIO [15].

4.3. Action Design

With the theory of connecting the user to the display completed, work could begin on designing each individual action for each cube. The actions to be covered are as follows:

i) Picking up and putting down chemical elements

ii) Storing and pouring chemical elements

iii) Interacting with stored elements external to the container

Considering the limited budget in place to develop the prototype, the minimum number of cubes to represent the maximum number of equipment and labs was designed. Four cubes were chosen, one to represent the action of transporting elements, one to the containment action, and one to represent the external interactive action. This allows the user to move elements to and from containers, interject or emit elements into the system, and contain and mix elements as seen fit. The first cube developed was the transporter, titled the ‘dropper’ cube. This cube simulates equipment that performs the first action, such as a tongs or a dropper. The schematic for the cube is seen in Figure 13.

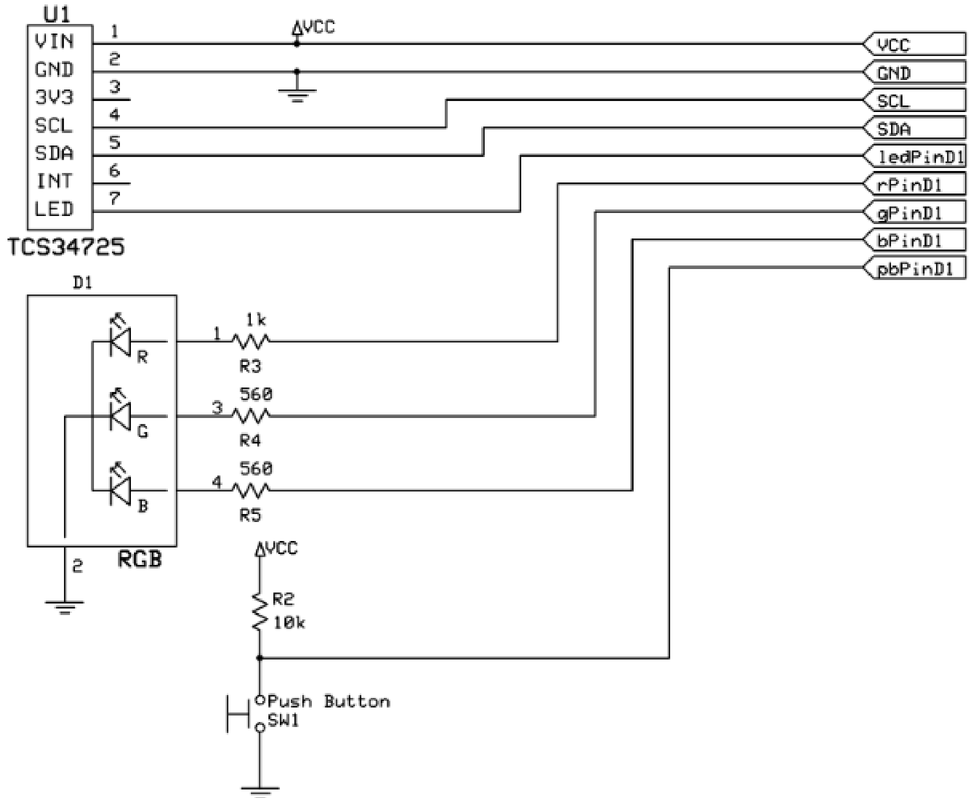


Figure 13: Schematic diagram of the push button Dropper cube.

Since the purpose of the dropper cube is to transport elements, a method of presenting and taking elements from the user to the display was necessary. In reality, there are 118 confirmed elements of the periodic table. Additionally, there are approximately 9 million named compounds, or combinations of those elements, in existence [11]. Although reaching the 9 million named compounds is out of scope for this thesis, a chart of the 118 can be achieved by color coding each element. Then, a color sensor could decipher each color. Figure 14 shows the color sensor chosen was the TCS32725 RGB Color Sensor made by AdaFruit. The process to perform the action is as follows.

1. User holds Dropper cube.
2. User clicks push button.

a. RGB LED is activated and color changes based on color read.

b. Color sensor LED is activated.

3. User hovers over a color which corresponds to a chemical element.

a. The RGB LED will match that color.

4. User clicks the push button again to lock that color in.

a. Color sensor LED is deactivated.

b. RGB LED color does not change.

5. User clicks the push button a third time to empty the dropper.

a. RGB LED is cleared.

Figure 14: Image of the TCS34725 RGB Color Sensor by AdaFruit.

This color sensor, priced at $7.95 USD a piece, is a cost effective solution to accurate color reading. Photodiodes generate electrical current by absorbing protons from light. When more current is produced, a more intense wavelength is read [16]. As seen in Figure 15, Based on the intensity of the wavelength produced, the response can be analyzed as color.

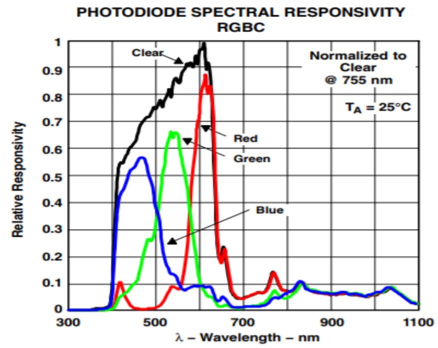


Figure 15: Graph showing Photodiode Spectral responsiveness to RGB colors.

To capture the color values being read by the device, the cube also utilizes an active-low momentary push button. The signal to the microcontroller is normally high, in this case by means of a 10k Ohm pull-up resistor. Active low implies this normally high signal suddenly goes low, and is used for its noise immunity and safety [6]. Also seen in Figure 13 is a common cathode RGB light emitting diode (LED). This LED allows the user to have a visual from the cube itself that the correct element color was selected.

Once an element has been selected, it must be placed in some type of container to do the actual experiment. In comes the next cube design, titled the 'beaker' cube, covering the second action. Figure 16 shows the developed schematic for the beaker. This cube leverages the same RGB LED, a linear gyroscope, and a magnetic switch, called a Reed switch.

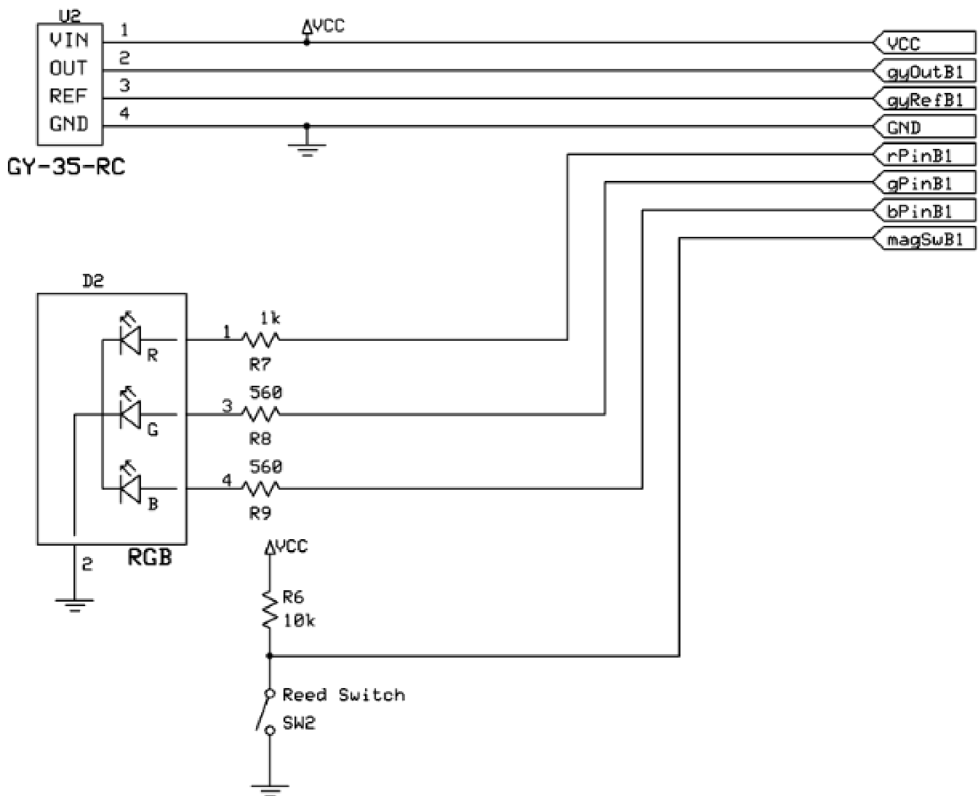


Figure 16: Schematic diagram for the Beaker cube.

Interaction between the dropper and beaker cubes starts with the magnetic Reed switch. Theoretically, a magnet inside the dropper cube would be placed over the Reed switch, signifying that the dropper is correctly positioned over the beaker. This removes the possibility of chemical spills and accidents. Like with the dropper, the beaker has an RGB LED to indicate that the correct element color has been placed. Finally, the beaker utilizes a single axis gyroscope, the GY-35-RC, as seen in Figure 17. The process to add an element to the Beaker Cube, or to empty it, is as follows.

1. Use the dropper to select and lock in an element.
2. Line the bottom of the dropper with the top of the beaker.
    a. The dropper's RGB LED will flash to show correct alignment.

3. While the dropper LED is flashing, press the push button.
    a. This empties the dropper contents into the beaker.
    b. The beaker LED will have the color from the dropper LED.
4. To empty, slowly lift the beaker vertically into the air
5. Quickly tilt the beaker towards the pointed stout.
    a. The RGB LED will now be cleared.
    b. Pouring away from the stout may also work, but it is not safe lab practice!

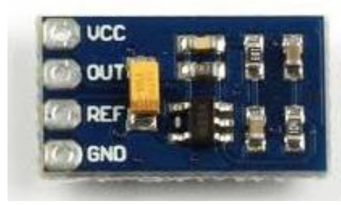


Figure 17: Single Axis Gyroscope GY-35-RC.

Beakers also have the ability to transfer contents similar to the dropper, so the gyroscope allows the user to physically move the beaker cube and move the element. If they move it over nothing, the beaker goes into an ‘empty’ state. If they move it over another beaker, the contents are emptied into one beaker to another.

For the third action, the last cube developed, the cube to emit elements, has the simplest design. Designated the ‘Bunsen Burner’ cube, it utilizes a single pole, single throw (SPST) switch to turn on and off the emitter, displayed with a red LED. An infer-red (IR) beam is used to detect the distance of an object placed above the burner. Finally, a second LED is used to show if the object is at the correct distance. Figure 18 shows the schematic diagram for the Burner.

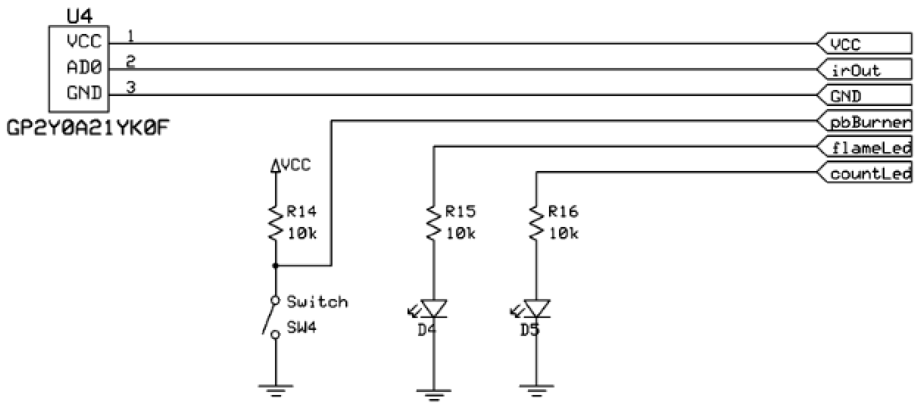


Figure 18: Bunsen Burner cube schematic diagram.

The GP2Y0A21YK0F is an IR sensor with a minimum distance of approximately 6 cm and maximum distance of 80 cm. As the distance of the object from the sensor changes, the voltage output changes as seen in Figure 19.

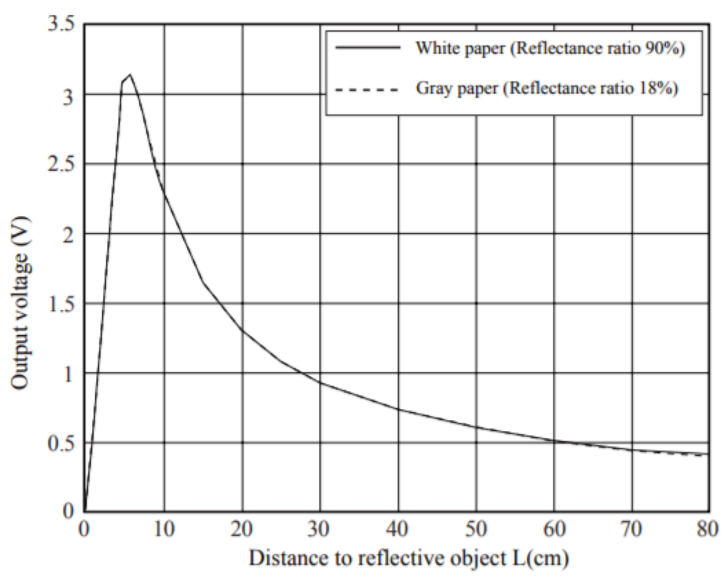


Figure 19: IR Distance sensor output voltage vs distance of reflection graph.

The process to use this cube is as follows.

1. Change the switch to the ON position.
    a. The red LED will turn on, indicating the burner is active.
2. Hold a beaker directly over the top of the top of the burner.
    a. The blue LED will turn on.
3. Wait the designated amount of time required for the lab.
    a. The blue LED will flash for a pre-determined amount of seconds.
    b. If the user moves away, the count starts over
    c. The blue LED will become constant when time is complete.
4. When complete, turn the switch to the OFF position.

Figure 20 shows the physical IR sensor with two 4-40 sized mounting holes. With the schematics for the 3 cubes designed and with the main components select, testing could begin on the practical application of using cubes to interact with an interface as a method of performing a chemistry experiment.

Figure 20: Physical case for the IR sensor with 4-40 mounting holes.

4.4. Pre-Prototype

With the concept completed, cubes designed, and the parts selected, work began on building a 'black box' version of the hardware. The goal of this pre-prototype was to test how the interaction between the cubes would feel. Since the concept of TUI for educational purposes is unique, it was important to ensure the components selected in the conceptual design would feel natural to use.

The first step was to test each individual component on a breadboard, as shown in Figure 21. This served a variety of purposes. Breadboarding a schematic allows the designer to test their circuit and make changes on the fly. Most important to this application, however, was that it required the code for each particular cube to be functional in order to properly test. Having the hardware code completed up front meant less code to develop down the road. The completed code can be found in APPENDIX I: ARDUINO C CODE FOR TUI CUBES.

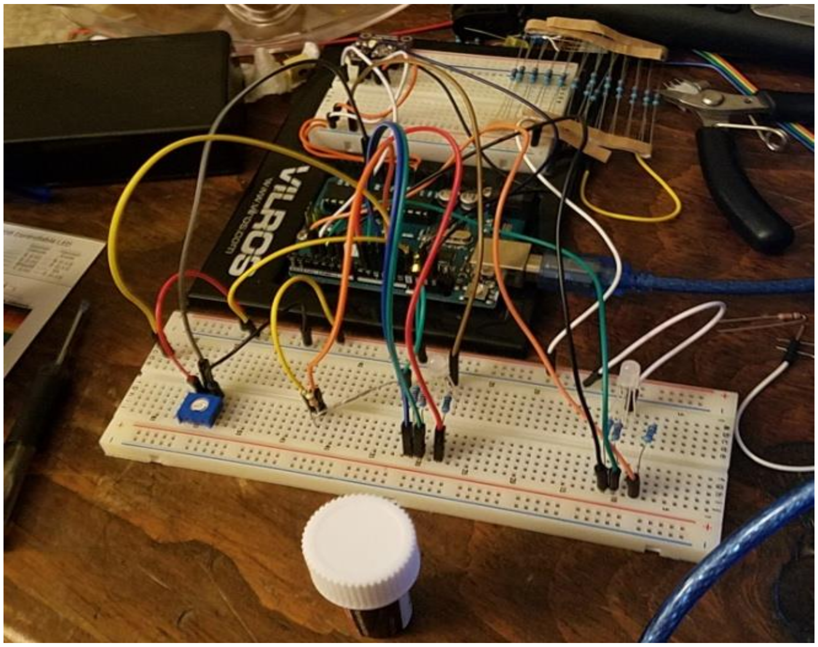


Figure 21: Cube schematics being tested on a breadboard and with an Arduino Uno.

Next was to enclose each circuit in a generic housing to test the interaction between each cube. For this preliminary design, three cubes were developed; the dropper, the beaker, and the Bunsen burner. The components in each cube were wired point to point to a ribbon cable which connected with the Arduino MEGA.

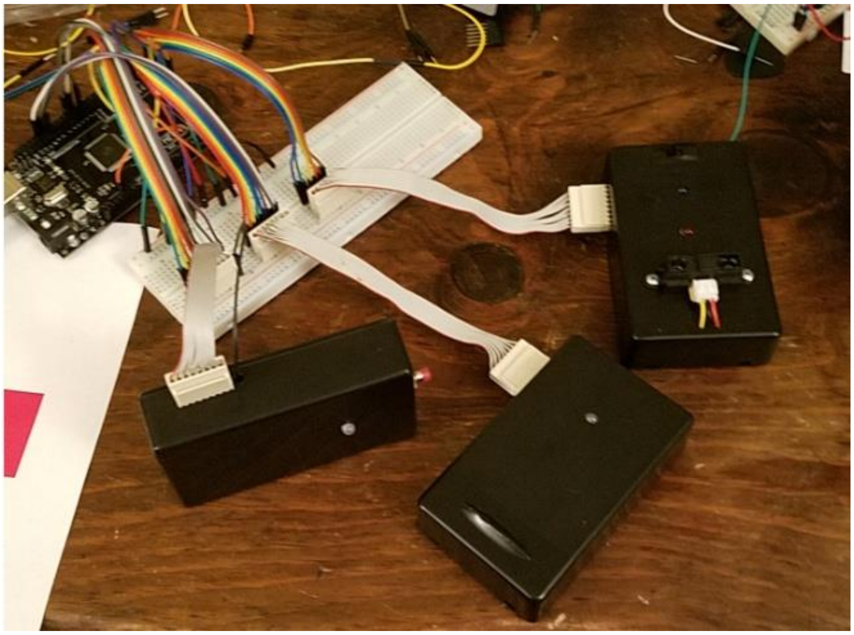


Figure 22: Black box version of the cubes.

Procedurally interacting with each of the three cubes was as follows. The user would press the button on the black box representing the dropper. This would cause the RGB sensor's LED to illuminate, as well as the RGB LED mounted on the front to project an approximate match to the color being read by the sensor. A second click would 'lock' that color into the black box representing the dropper, and would be held steady on the RGB LED. Additionally, the LED on the sensor would turn off. The user would then line up a magnet inside the dropper black box with the reed switch exterior to the beaker. A final third click on the dropper's push button would remove the color from the dropper black box and send it to the RGB LED on the beaker black box, representing the contents moving from the dropper to the beaker. For the final interaction, the user would turn the switch on the black box representing the burner to the ON position. This would activate the red LED, simulating the flame being activated. The user would then hold the beaker cube over the IR sensor on the burner cube. A blue LED would flash, indicating the correct height requirements were met.

Avoiding personal bias when finding criticism with the early design, these black box cubes were informally tested on family and close friends. This allowed a preliminary baseline of the concept to be developed, as well as a way to gather insight on what needs to be improved for the final prototype. Overall, the preliminary testing was successful, and met with much positivity. All criticism was towards the actual mechanics of making it feel more natural, and not towards the concept or method of learning. The first most common criticism was that it was difficult to align the magnet from the dropper and the beaker. The second most common criticism was that it was difficult to determine how high to hold the beaker. The final most common criticism was that the ribbon cable wires were quite short and felt very limiting with being able to move the cubes. This criticism was vital in the design of the actual cubes.

4.5. Tangible User Interface Mechanical Design

Orthographic drawings are used by mechanical engineers as a way to convey size and dimension to objects, thus three orthographic drawings were developed to shape and size each cube. To begin, the first cube developed was the dropper. The dropper is pivotal to the entire design, for it is the mode of interaction between the elements and the lab. In the design criteria, the dropper must be able to house the RGB sensor to read color, a magnet to activate the reed switch, have an LED indicator, have a push button, and have a general look and feel of a dropper. Figure 23 shows the developed orthographic drawing for the dropper. Not reflected in the drawing is the hole that was drilled on the bottom for the RGB color sensor to see through. The hole was omitted because the final RGB sensor had not been selected. The design is ambidextrous, and is best when held firmly in the users' hand.

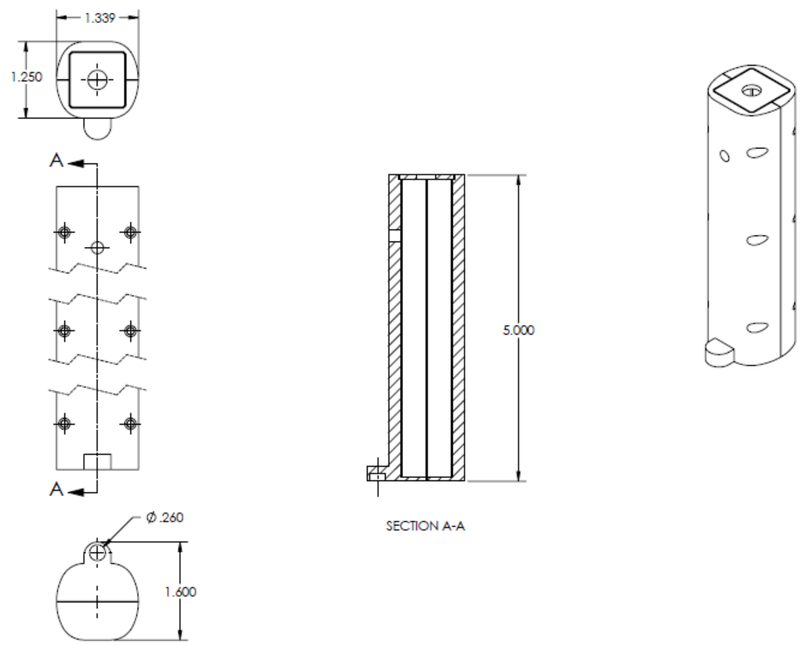


Figure 23: Orthographic Drawing of the dropper cube which was developed. Modifications were required after development in that a hole was added to the bottom for the RGB color sensor to scan through.

Following the dropper, the next cube developed was the beaker. Although there are two beakers, only one drawing and design was required since their function is identical. As discovered when experimenting with the black box version, it was important that the lid of the beaker had a slot to ensure the dropper's magnet always lined up with the reed switch. Learning from the alignment difficulty criticisms, as seen in Figure 24, the lid of the beaker has a slot that is an exact replica of the bottom of the dropper. There is a slight tolerance in size of 0.01" to ensure a smooth fit and transition so the magnet is always aligned.

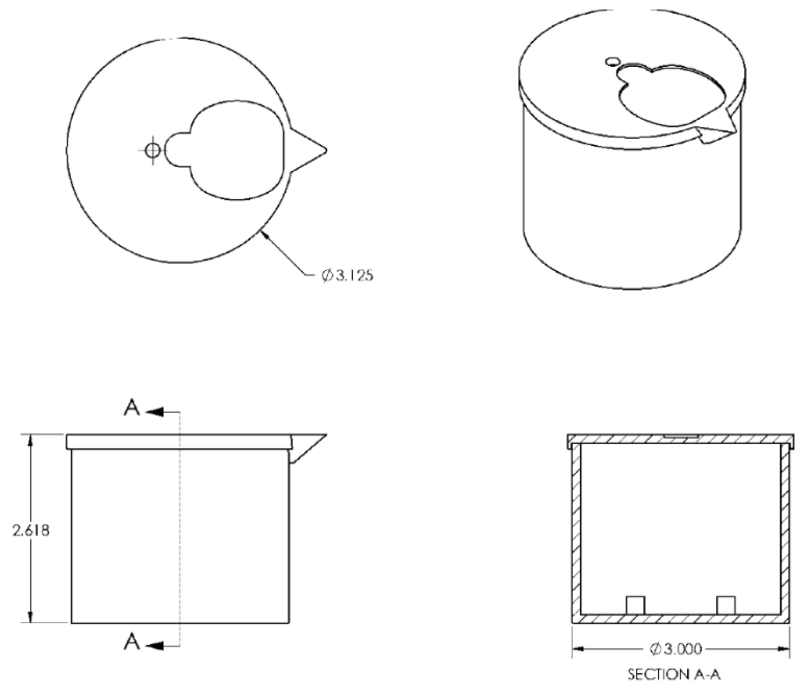


Figure 24: Orthographic drawing of the beaker cube.

Also featured here are four standoffs attached to the bottom of the beaker cube. The four standoffs are used to mount a printed circuit board (PCB) as a way to mount the electrical components to the beaker. Additionally, it was imperative that the gyroscopic sensor be mounted in a stable location due to its highly sensitive nature. Since it is a single axis gyroscope, the direction of the sensor must align with the lid of the beaker cube to allow the pouring motion to feel and look natural.

The third and final cube developed was the Bunsen burner cube. This cube features mounting holes on top for the IR distance sensor, a hole for an on-off switch, and two holes for LED indicators. Shape and design wise, the goal was to mimic that of a realistic Bunsen burner. Figure 25 shows the orthographic drawing of the Bunsen burner cube.

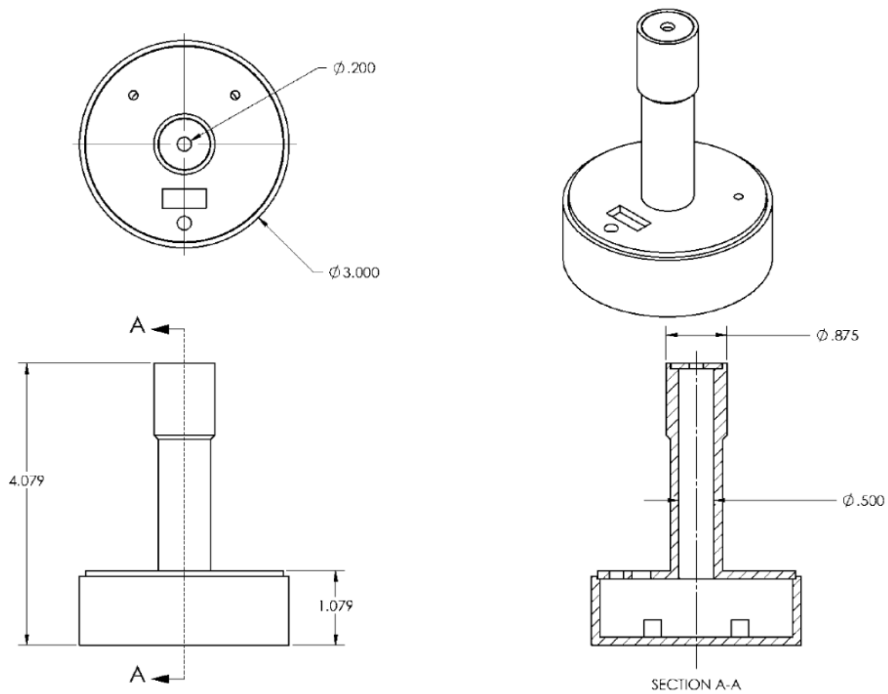


Figure 25: Orthographic drawing of the Bunsen Burner cube.

Careful consideration went into the height of the tower. As stated in section 4.3, the minimum distance of the IR sensor was approximately 6 cm, or 2.3”. The height of the housing was 0.5”. With these criteria in mind, the height had to be greater than 2.8” to ensure the closest the user could get to the top of the Bunsen burner was, in fact, the minimum distance the sensor could detect. With the schematic tested, electrical components selected, and cubes designed, a Bill of Materials (BOM) could be developed to optimize the costs of the entire project, as seen in APPENDIX IV: BILL OF MATERIALS.

To save time and cut costs, polyamide 3D printing was chosen as the material and build of the cubes. In order to 3D print, the orthographic drawing had to be ported to SolidWorks. Figure 26 shows the completed Bunsen Burner cube with components installed. The IR sensor is slightly angled to assist maximum reflection.

Figure 26: Bunsen burner cube 3D printed and with components installed.

Figure 27 shows the next cube developed; the dropper cube. While the burner and beaker cubes were designed for pressure fit, this cube was designed for a 4-40 screw seal.

Figure 27: Dropper cube 3D printed with components installed.

Lastly, the two beaker cubes were printed and assembled, as seen in Figure 28.

Figure 28: Beaker cubes 3D printed with components installed.

Each cube posed its' own set of challenges after completion. Initially, PCB's were designed on two layer boards with solder mask and silk screen layers for ease of component installation. Unfortunately, purchasing these PCB's significantly exceeded the budget, thus a hand soldered wire-wrap solution was chosen, as seen in Figure 29. This helped to cut costs at the expense of more development time required.

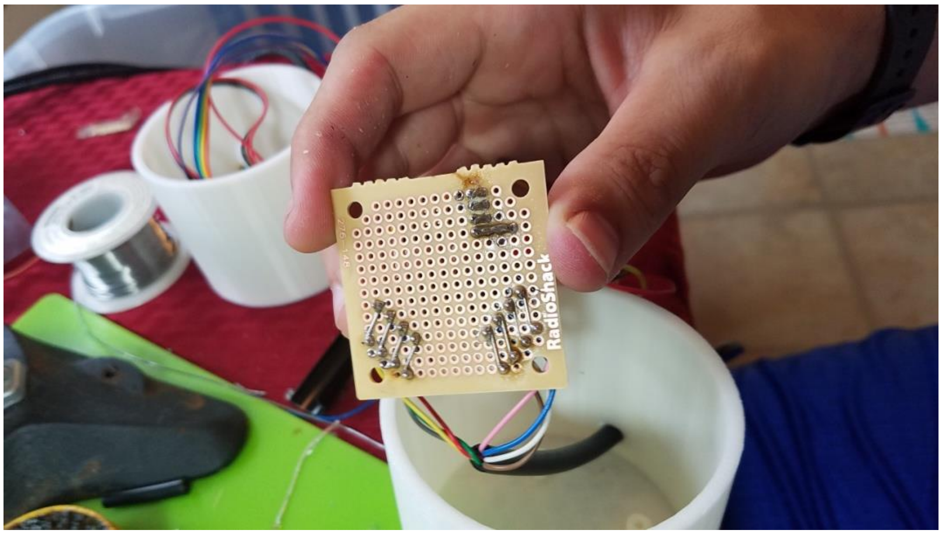


Figure 29: Wire-wrap PCB solution for component installation.

4.6. Software Development

Following the successful design and implementation of the four cubes, work could begin on the software development for the chemistry portal. Keeping in mind the goals designated at the start of the project, the portal had to allow the user to interface with it using the cubes. This portal had to guide and instruct the user on completing the lab, as well as provide relative feedback based on the users actions. Finally, the portal itself should be easy to read and navigate such that the widest range of user chemistry experience can benefit from the device. Figure 30 shows the flow diagram for the portal.

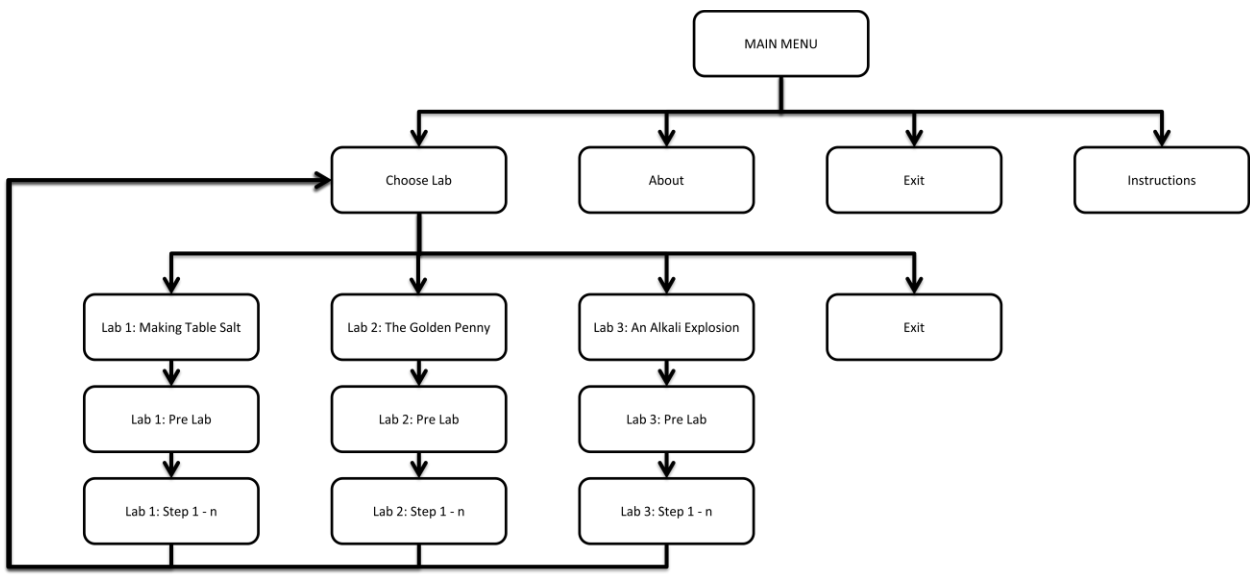


Figure 30: Flow Diagram of Portal

The basic flow of the program is that the user enters the main menu, moves to choose a lab, then, when they pick their lab, does the pre lab, then follows the step to completion. Once done, they can choose another lab. Currently, the labs are hard coded. Adding additional experiments would require adding the code and doing a software update on the program.

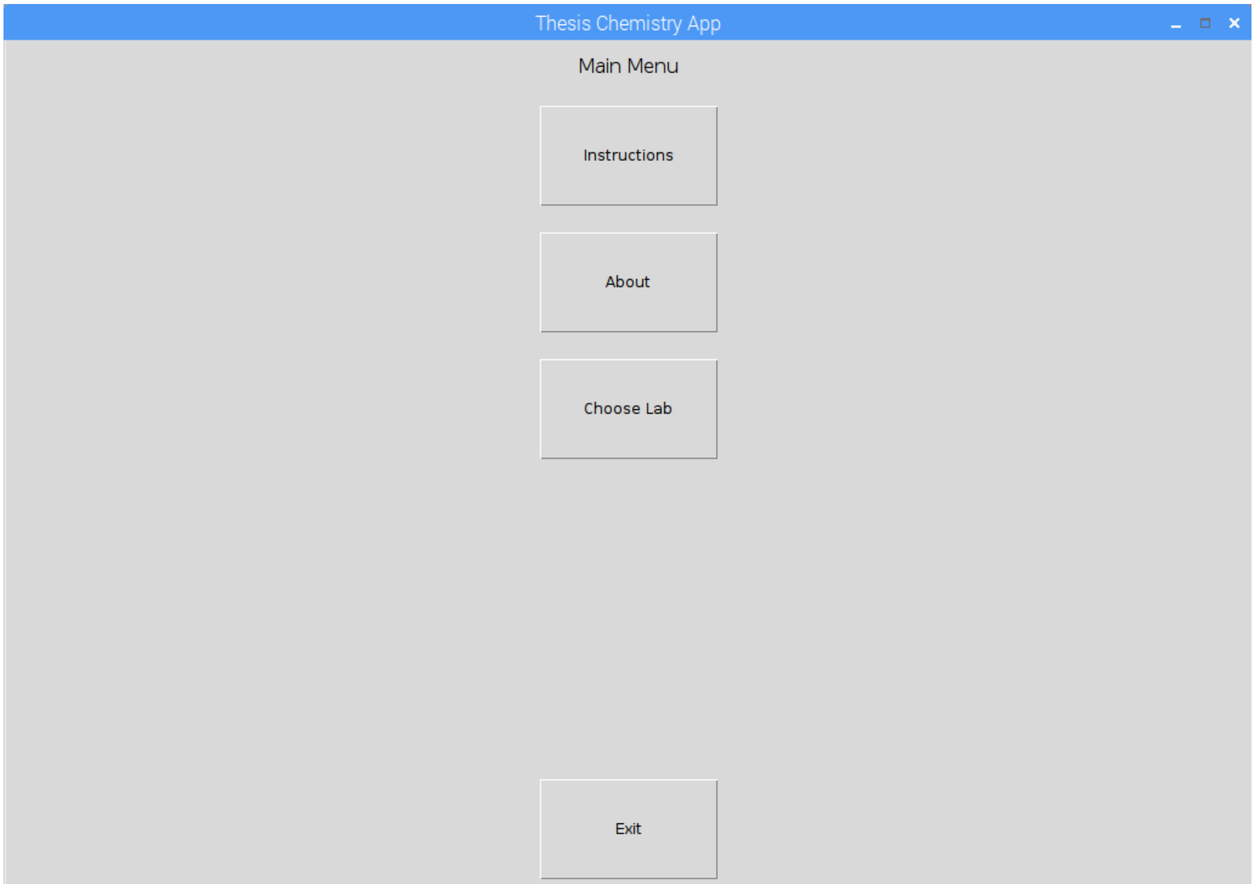


Figure 31: Main page of the portal with various options to select.

The portal opens to a main page as seen in Figure 31. This main page has multiple buttons for the user to interact with, each bringing up a separate page. The *Instructions* button opens up a page with details on how to manipulate and operate each cube, as well as how to navigate the portal. The *About* page gives background on the concept, the author, as well as thanks to those that assisted the author along the way. The *Choose Lab* button opens to the page where the user would choose which lab to do. Finally, the *Exit* button closes the program.

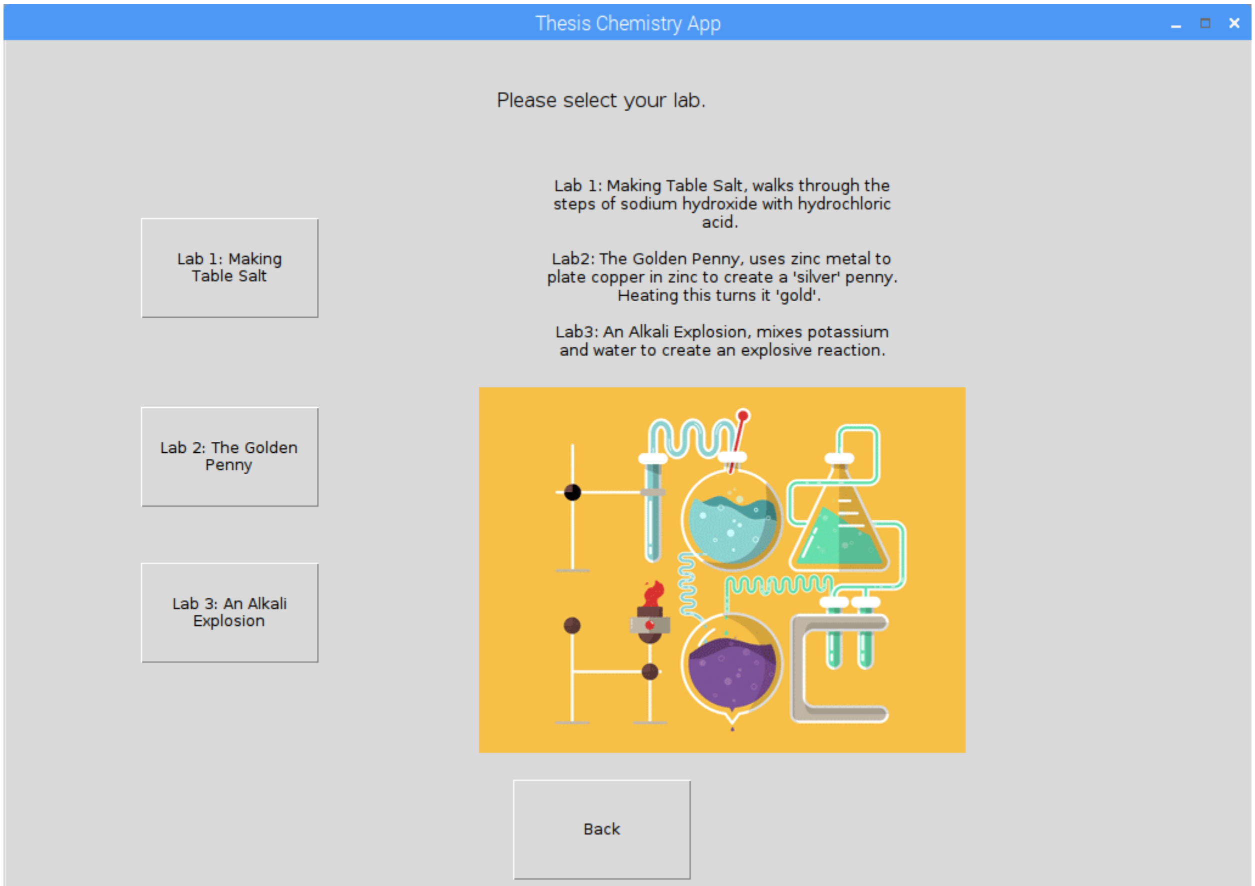


Figure 32: Choose Lab page with 3 playable options with short description.

As seen in Figure 32, the user has 3 options of labs to select. Each lab has a short description about what the lab entails. There is also a *Back* button to bring the user back to the main menu. Upon selecting a lab to try, the portal then opens up that particular labs' pre lab page, as seen in Figure 33. In this pre lab page, the user is given the necessary background information on the lab, as well as any associated chemical equations. There is also details on the procedure of manipulating the cubes to achieve the desired outcome.

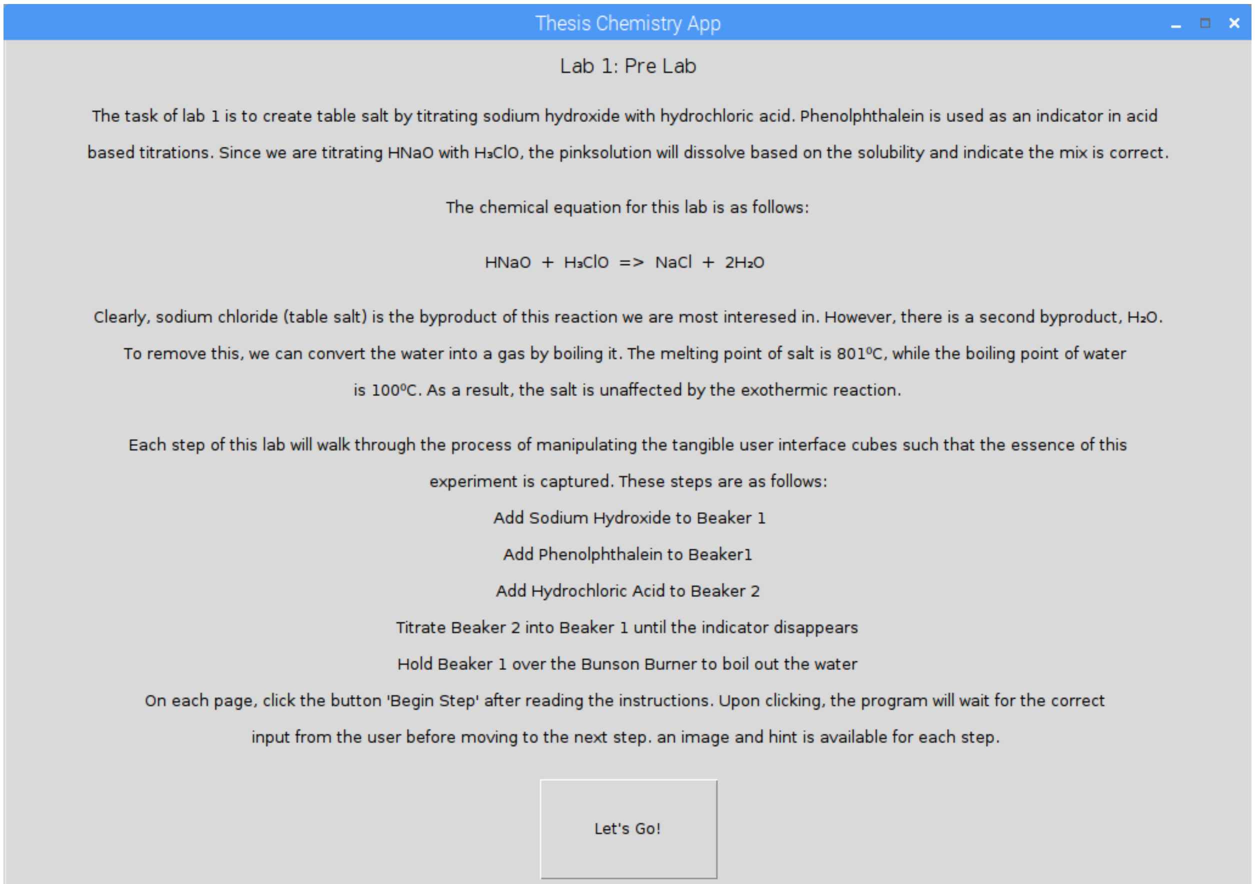


Figure 33: Pre lab page with details on how to do the lab as well as the chemistry to happen.

Upon pressing the *Let's Go!* button, the lab begins on Step 1 in the procedure. Each step has four pieces to it. The first piece is the actual instruction of what to do, such as pouring a beaker or grabbing an element with the dropper. The next piece is a graphical image, showing the user what action to complete. The third piece is a hint on manipulating the equipment to successfully complete the step. Finally, the fourth piece is a video that plays upon a successful action. Each video is unique and filmed by the author showing the actual chemistry in the physical world. An example of the step pages can be seen in Figure 34. The completed code for the portal can be found in APPENDIX II: PYTHON CODE FOR CHEMISTRY APP PORTAL.

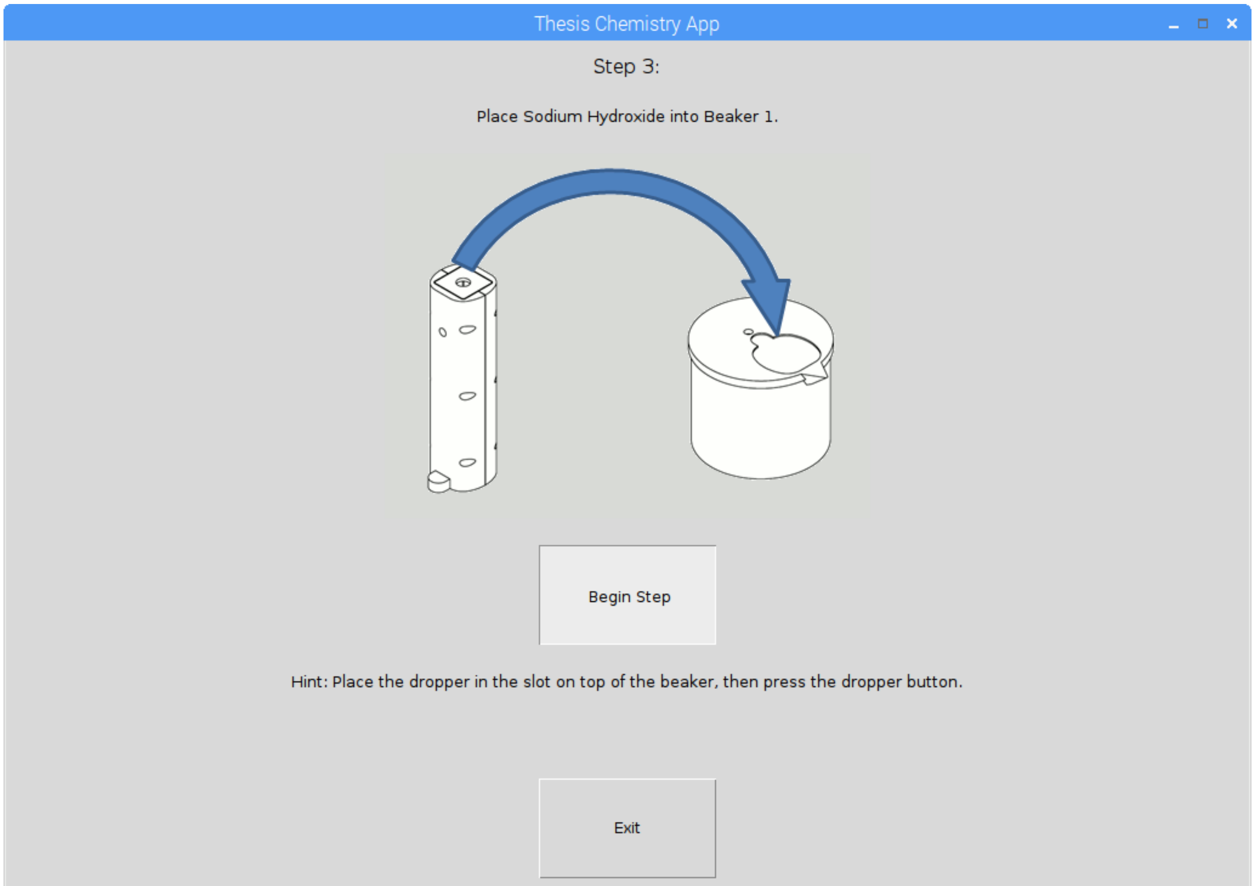


Figure 34: Example of a step in the procedure for completing the labs.

5. CHAPTER V: Testing Method and Procedure

5.1. Method of Research

To develop the proof of concept of a tangible user interface to learn chemistry, an experimental research method was chosen. There are thousands of various sensors that could have been chosen for the cubes. The best method to determine which to use came from a combination of researching the types of sensors required, the ease of development, and experimentally testing each device on a bench top.

For example, three different types of distance sensors were tested on the burner cube's distance detection. The first sensor was an IR sensor, the second was an ultrasonic sensor, and the third was a proximity sensor. As seen in Table 1, a Kepner Tregoe (KT) Analysis was used to determine which sensor to choose. The KT method is a decision making model that compares alternatives using weighted criteria [17]. Since testing with the IR sensor was successful, the need was not there to pursue tests with the proximity sensor.

Alternative		IR Sensor		Ultrasonic Sensor		Proximity Sensor	
MUSTS	1. Cost Effective	YES		YES		YES	
	2. Able to detect Distance under 10cm	YES		NO		YES	
	3. Have a small package size	YES		YES		YES	
Wants	Weight	Rating	Score	Rating	Score	Rating	Score
1. Minimum Size	2	1	2	X		2	4
2. Ease to Mount	4	3	12			2	8
3. Accuracy	6	2	12			2	12
		TOTAL	26	TOTAL	0	TOTAL	24

Table 1: KT Analysis for alternatives on distance detection for the burner cube.

5.2. Data Gathering via Survey Study

Adequately testing the proof of concept utilized human research through a survey study. In theory, participants would be presented a pre-survey, be presented the lab, then upon completion, a post survey would be conducted. The purpose of the pre-survey was to gather background information on the participants' level of chemistry experience, age, and assumptions about the lab to be presented. Following this survey, the lab would then be administered and the participant would use the various cubes to interact with the chemistry portal interface. Upon completion, the participant would then complete a post survey. The objective of this survey was to gather necessary information on the users' experience, if the proof of concept was successful, as well as any criticism to help improve the device. The pre and post surveys can be found in APPENDIX III: PRE AND POST SURVEYS.

6. CHAPTER VI: Results and Analysis

6.1. Completed Prototype

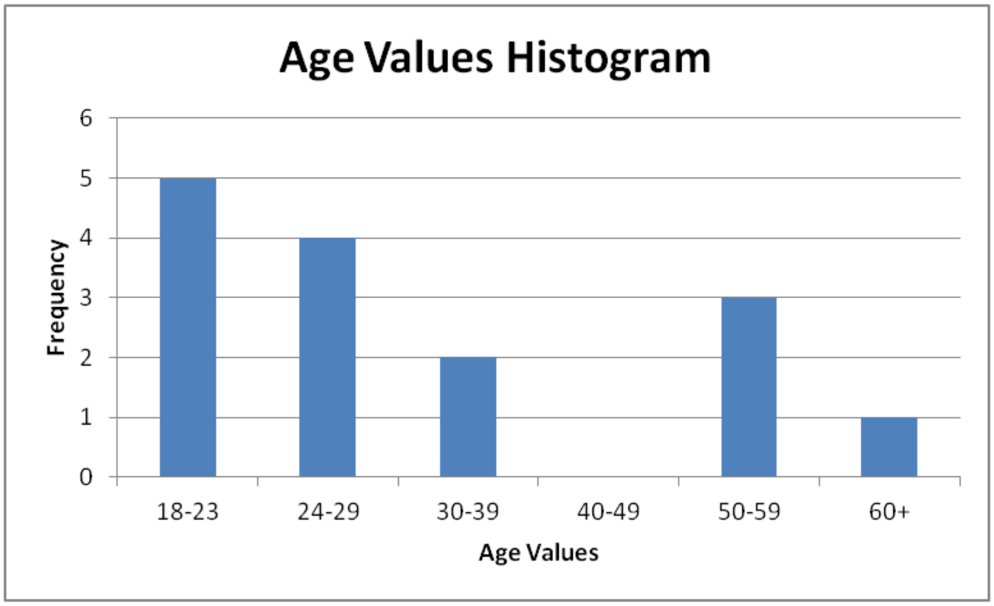
Included in the completed prototype was the following collection of items. First, there was a fully functional portal whose purpose was to guide any user through virtual chemistry experiments. Second, four cubes were available to interact with the user interface. These cubes were a dropper cube, a Bunsen burner cube, and two beaker cubes. Third, a pre and post survey was created to accompany the prototype. After building the four cubes, constructing the portal, and creating the surveys, the prototype was completed and ready to be administered to participants.

6.2. Questionnaire Data

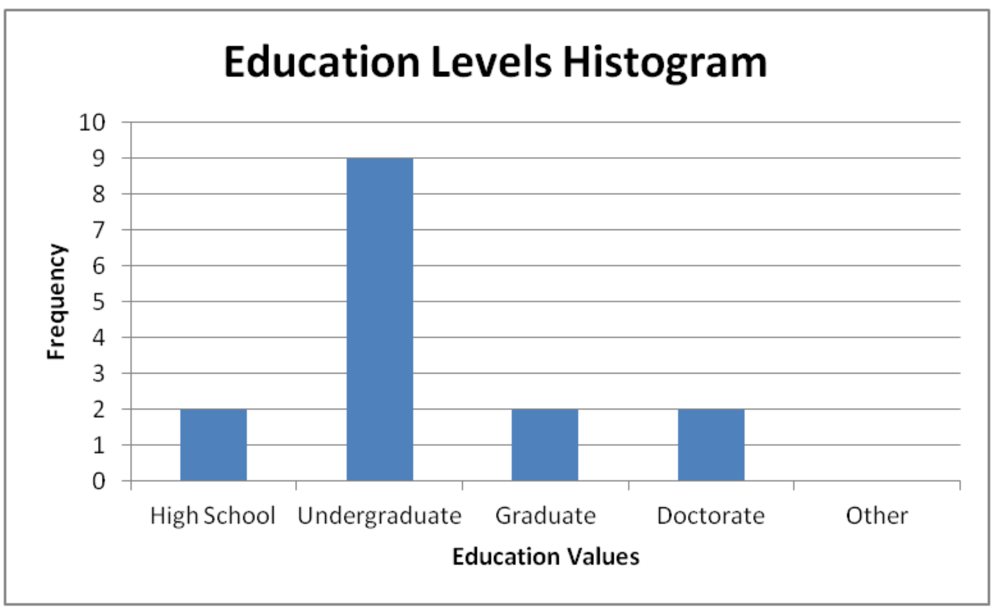
With the prototype completed, 15 anonymous participants voluntarily aided in the research by completing an experiment using the TUI. Each filled out the pre and post surveys. The goal of the surveys was to gather user input on the concept and implementation. All data gathered from the questionnaire was as follows.

Pre Survey

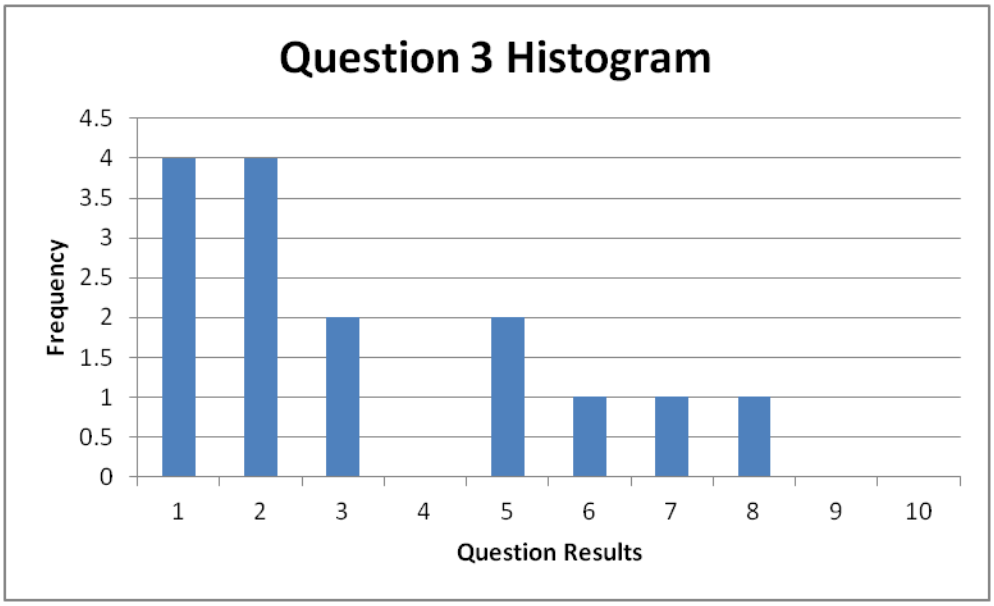
1. Age Values:



2. Education Levels:

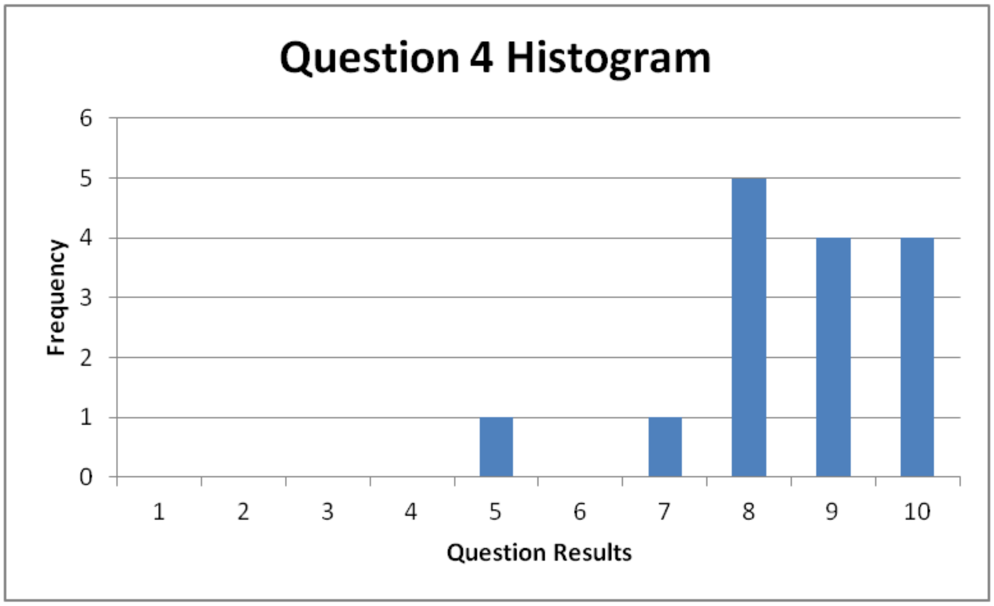


3. Average Level of Chemistry Experience:



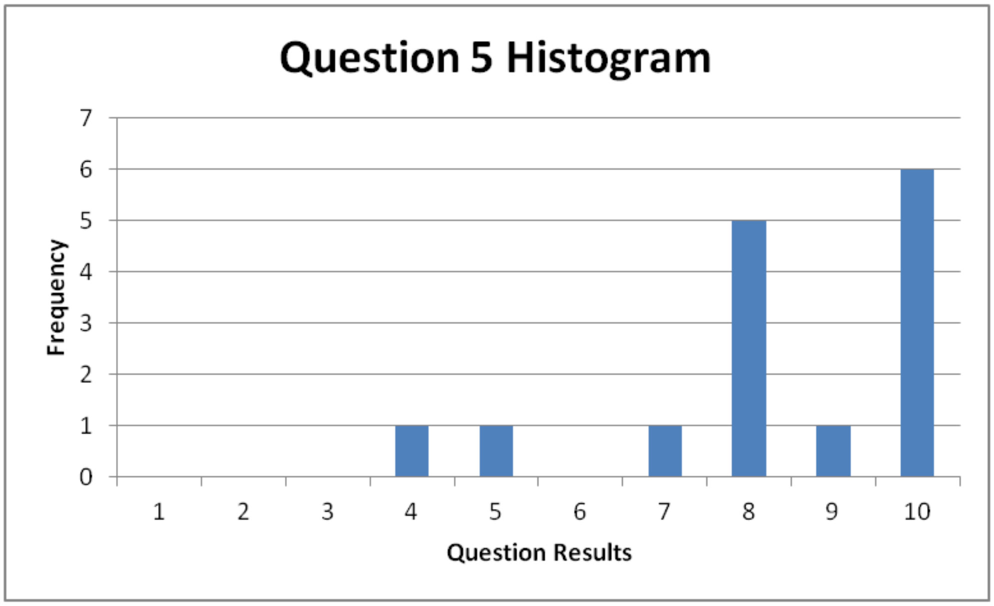
Average = $\frac{6+3+1+3+1+2+8+2+1+5+2+7+1+2+5}{15} = \frac{49}{15} = \mathbf{3.3}$

4. How practical does this concept sound?:



Average = $\frac{8+10+9+7+9+10+5+10+10+8+9+9+8+8+8}{15} = \frac{127}{15} = \mathbf{8.5}$

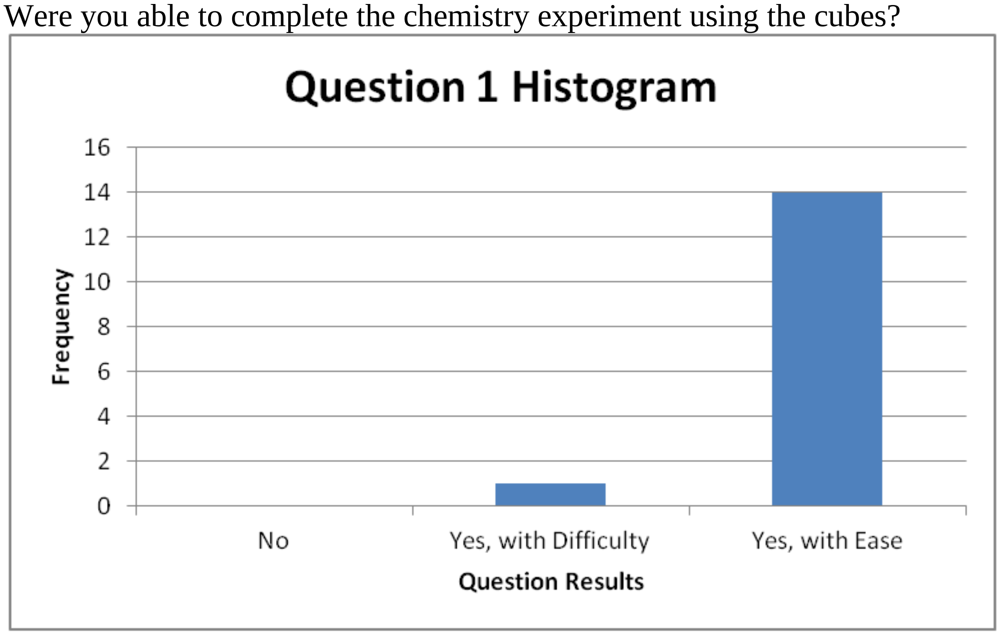
5. How likely would you be to choose this alternative over traditional chemistry?:



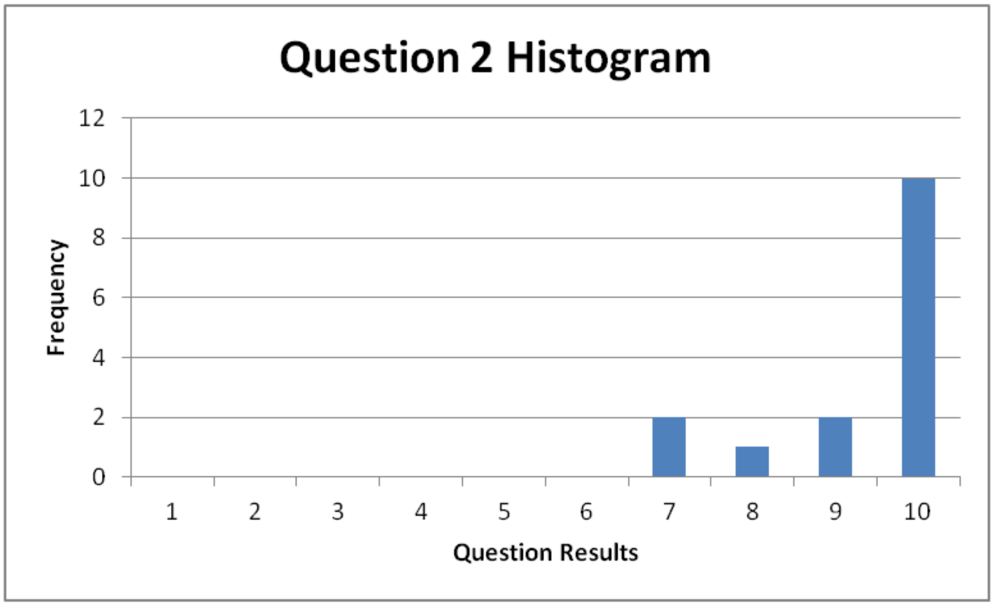
Average = $\frac{4+10+8+7+10+8+5+10+10+8+10+10+8+9+8}{15} = \frac{125}{15} = \mathbf{8.3}$

Post Survey

1. Were you able to complete the chemistry experiment using the cubes?

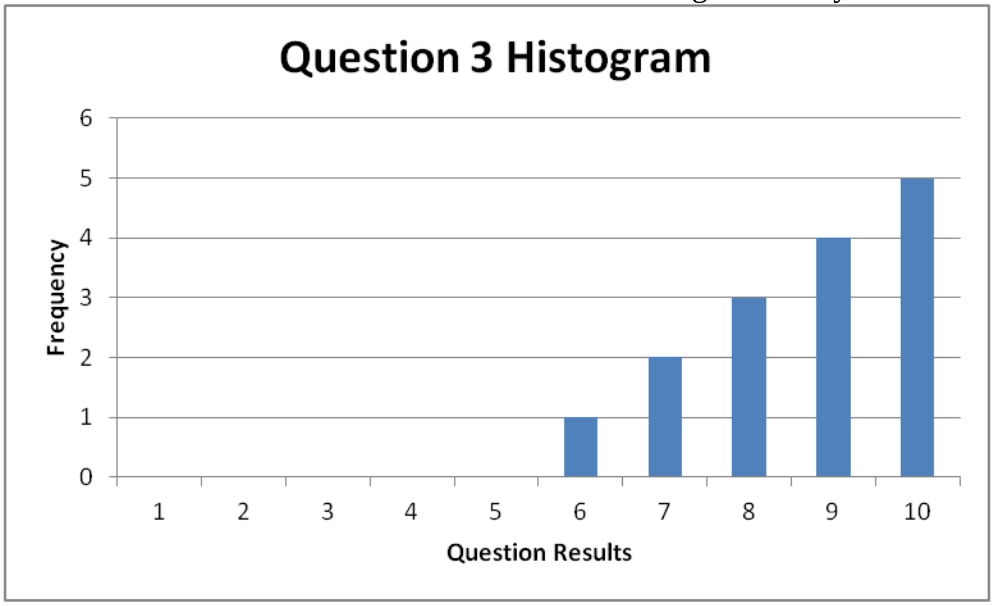


2. How positive would you rate your experience?



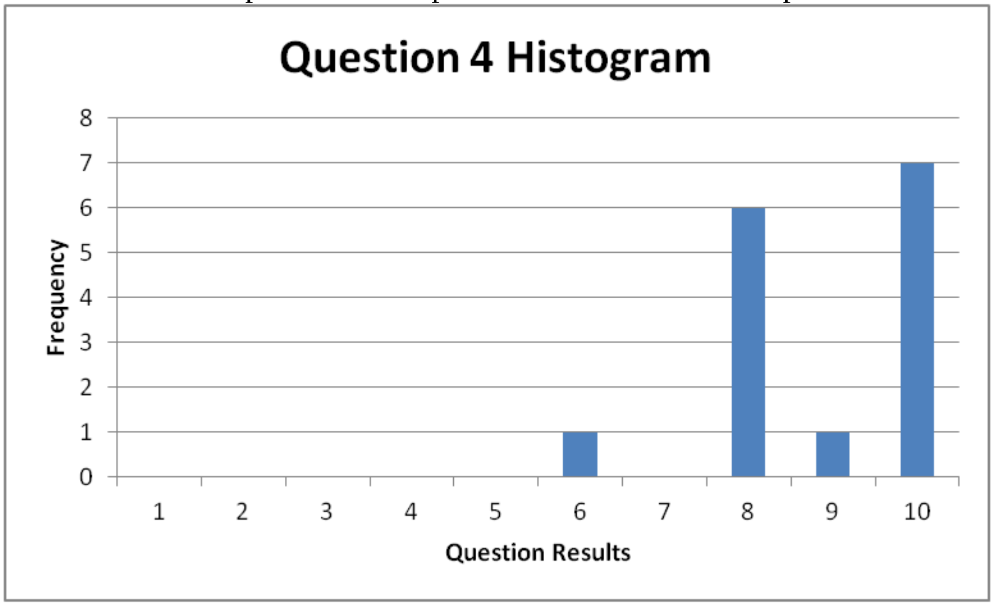
Average = $\frac{7+10+10+10+10+9+10+10+10+7+10+9+10+10+8}{15} = \frac{143}{15} = \mathbf{9.3}$

3. How effective does this method seem to be for learning Chemistry?



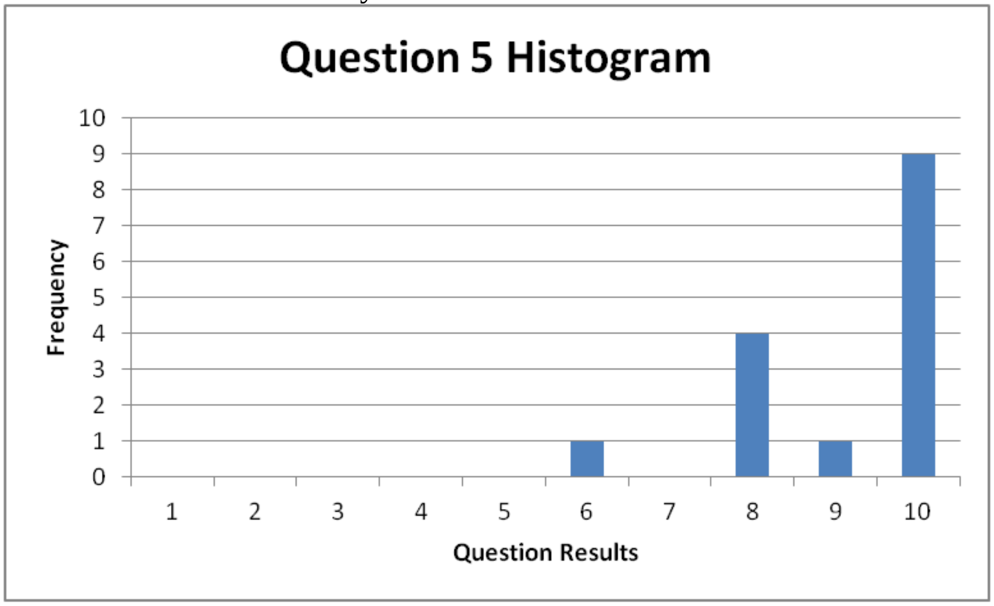
Average = $\frac{7+10+8+8+10+9+10+6+9+7+9+9+10+10+8}{15} = \frac{128}{15} = \mathbf{8.5}$

4. How well does the proof of concept relate to the actual concept?



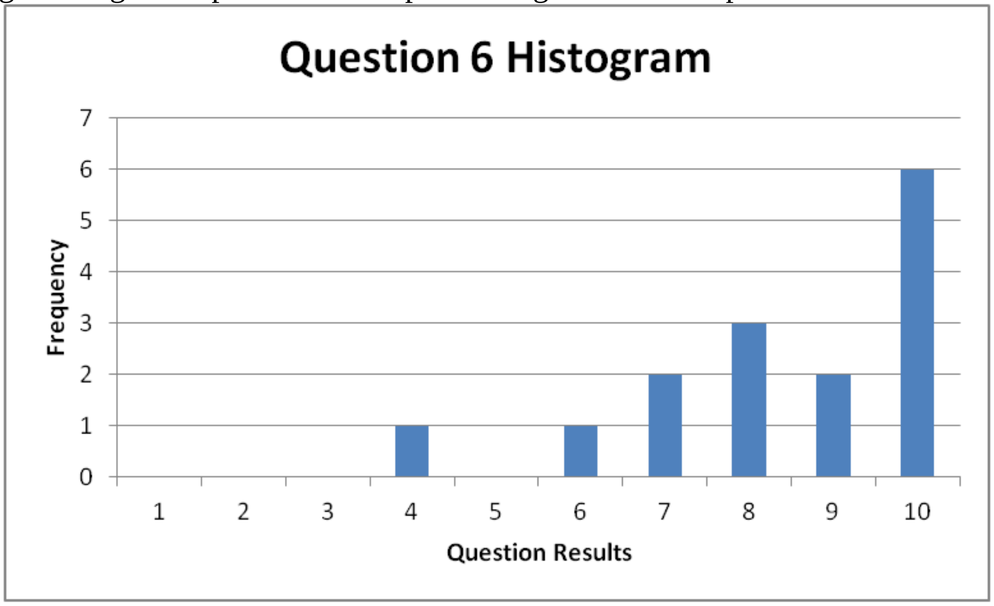
Average = $\frac{6+10+8+8+10+8+10+10+10+8+10+10+8+9+8}{15} = \frac{133}{15} = \mathbf{8.8}$

5. How likely would you be to choose a chemistry course that uses this method of teaching over the traditional chemistry labs?



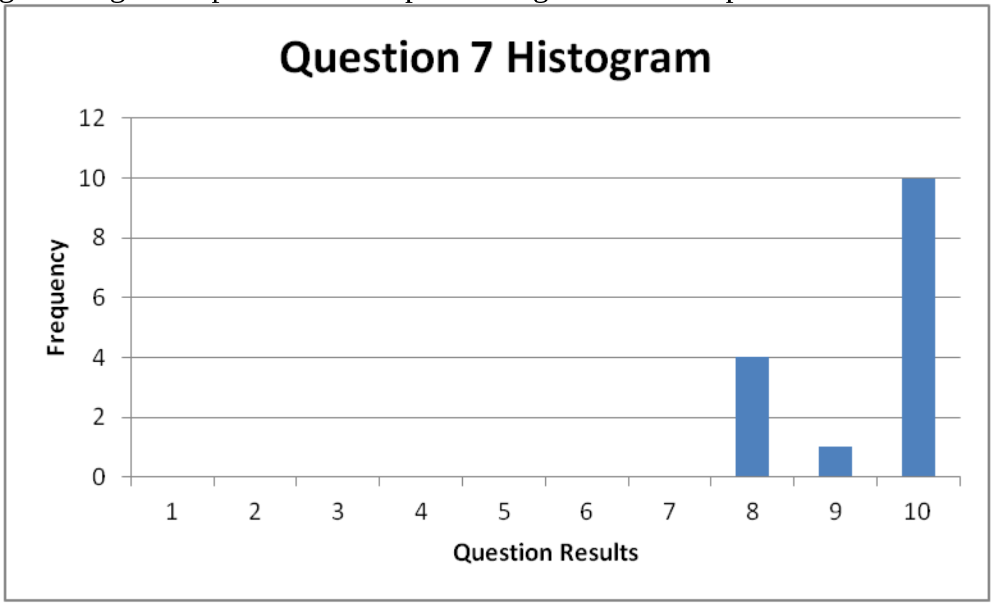
Average = $\frac{6+10+8+8+10+8+10+10+10+8+10+10+10+9+10}{15} = \frac{137}{15} = \mathbf{9.1}$

6. How accurate do you feel the dropper ‘cube’ represented a real life dropper to simulate a good tangible experience when performing the virtual experiment?



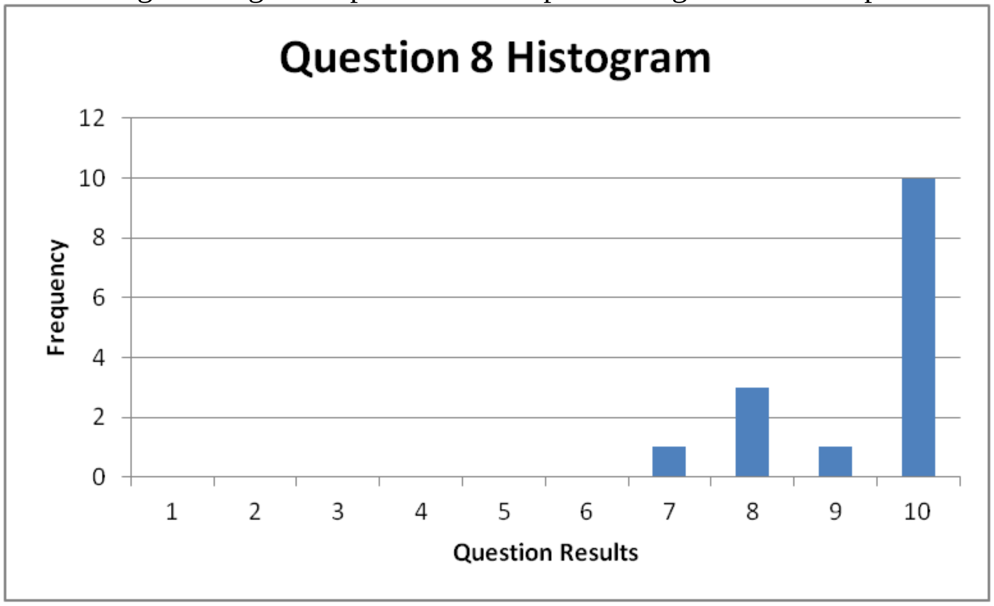
Average = $\frac{4+10+10+10+10+8+6+8+7+8+10+10+7+9+9}{15} = \frac{126}{15} = \mathbf{8.4}$

7. How accurate do you feel the beaker ‘cubes’ represented a real life beaker to simulate a good tangible experience when performing the virtual experiment?



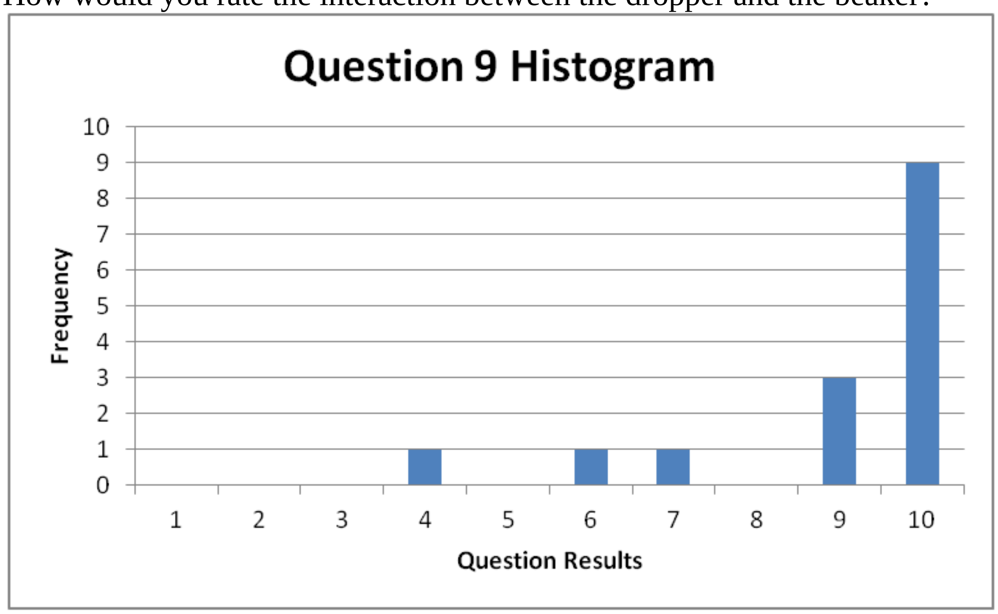
Average = $\frac{9+10+10+10+10+8+10+10+10+8+10+10+8+10+8}{15} = \frac{141}{15} = \mathbf{9.4}$

8. How accurate do you feel the burner ‘cube’ represented a real life Bunsen burner to simulate a good tangible experience when performing the virtual experiment?



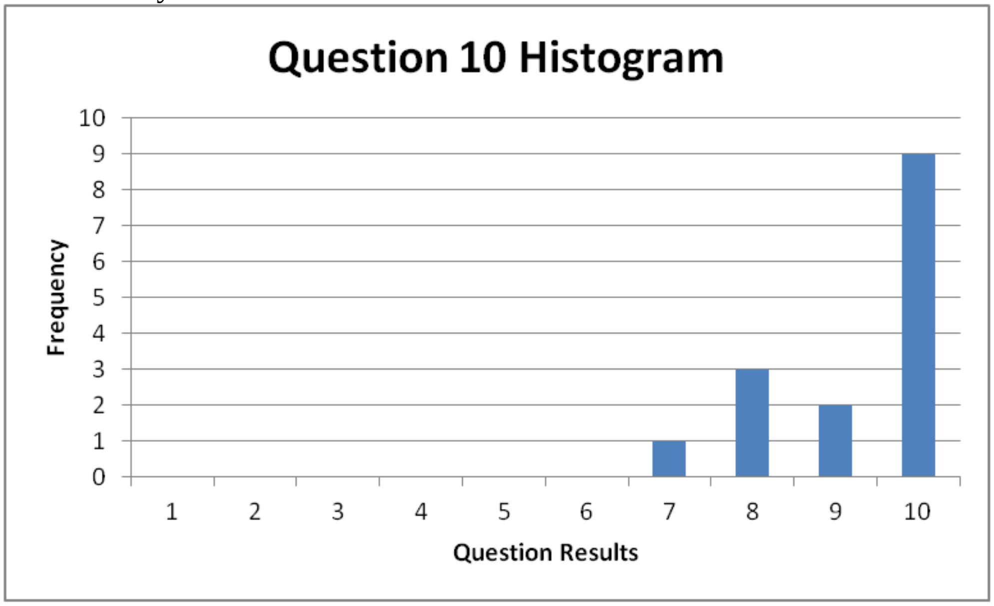
Average = $\frac{9+10+10+10+10+10+7+10+10+8+10+10+8+10+8}{15} = \frac{140}{15} = \mathbf{9.3}$

9. How would you rate the interaction between the dropper and the beaker?



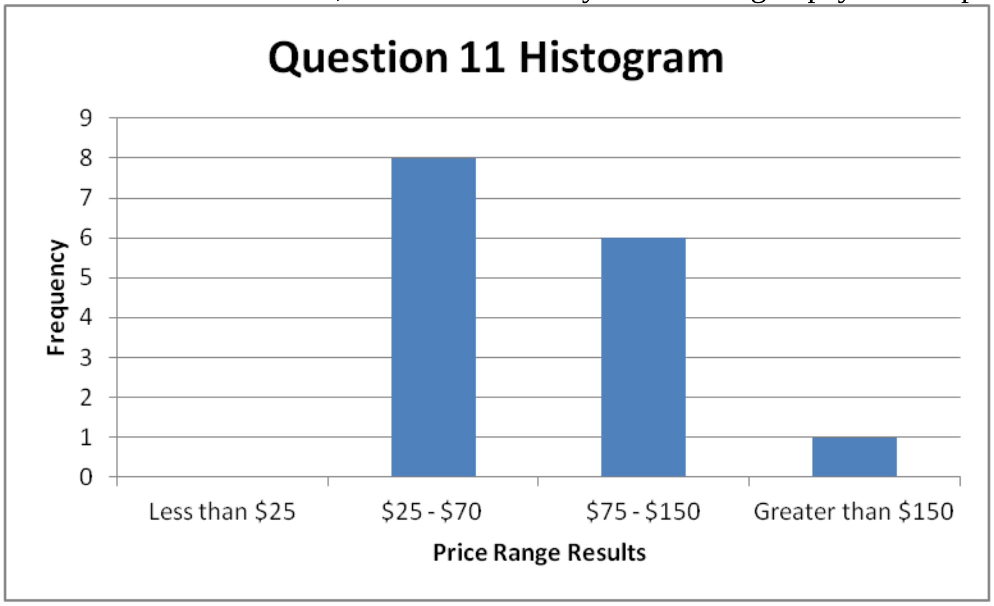
Average = $\frac{4+7+10+10+10+10+6+8+10+10+10+10+8+8+10}{15} = \frac{131}{15} = \mathbf{8.7}$

10. How would you rate the interaction between the beaker and the Bunsen burner?



$$\text{Average} = \frac{9+10+9+10+8+10+7+10+10+8+10+10+8+10+10}{15} = \frac{139}{15} = \mathbf{9.3}$$

11. If made available in stores, how much would you be willing to pay for this product?



6.3. **Data Analysis**

Analyzing the data from 6.2, the reception of the thesis was overwhelmingly positive. Fifteen participants tested the project. Of those fifteen, ten were considered to have less chemistry experience, while five were considered to have greater chemistry experience. Thirteen of the fifteen participants had a college degree of any level. As determined in the pre survey question 3, the average chemistry experience was moderately low. Question 4 and 5 showed an overall positive outlook and open-mindedness towards virtual chemistry.

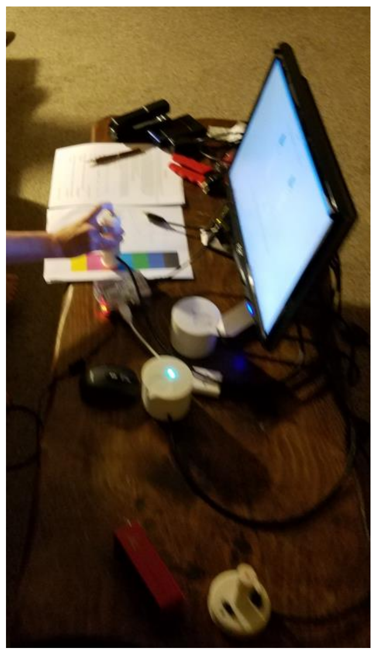


Figure 35: Participant in the action of testing the thesis while conducting the survey.

In the post survey, fourteen of fifteen participants completed the survey with ease. The one participant who had difficultly completing the experiment had issues with the dropper and beaker cube interaction. Unfortunately, from the wear and tear of testing, the reed switch in beaker 2 had dismounted, causing the magnet to struggle to make contact and move to the next step. Overall, the ratings show the majority of participants had a positive experience and would be receptive to taking a course utilizing a TUI to learn chemistry.

As for the cubes and their interactions, the dropper cube met the most criticism from those with chemistry experience. The beaker cube on average rated at 9.4 out of 10 for having a realistic feel and good experience, and the burner cube rated at 9.3 out of ten. The dropper cube, however, received 8.4 out of 10. Due to its versatile nature, those participants felt the design was limiting. The dropper cube has many rolls; for instance it acts as a pipette, crucible tongs, a dropper, or a burette. The beaker and burner cubes were identically rated. Both had fairly simplistic and straight forward applications. The beaker cube can accept elements, and elements can be poured out. The burner cube turns a flame on and off.

Two main interactions were measured between the cubes; the dropper and beaker interaction and the beaker and burner interaction. The interaction between the dropper and beaker was rated at 8.7 out of 10 on average, while the beaker and burner was rated at 9.3 on average. As described earlier, the dropper interacting with the beaker was more critically rated than between the beaker and burner. The dropper attaches to the beaker through a designed slot to ensure the user is lined up with the magnetic switch. Visual aid of the action being completed was the droppers LED would flash. The beaker interacts with the burner with a distance sensor. The participants quickly determined the height based on visual aid from the burners LED indicator, and would hold until complete.

6.4. Summary of Findings

In summary, two groups of participants tested the TUI chemistry labs. Group 1 consisted of 10 participants who did not possess significant chemistry experience. Group 2 consisted of 5 participants who were moderately to highly experienced in the field of chemistry. Averaging the findings between the two groups showed an overwhelmingly positive reception to the concept and actual implementation of the project. Conceptually, the participants felt overwhelmingly positive about the idea of a virtual chemistry laboratory. Practically, the implementation was successful with only a few minor changes needed moving forward. Most criticism was towards the dropper cube due to its versatile nature. Overall, testing was highly successful and helped validate moving forward with the project.

7. CHAPTER VII: Summary, Conclusions, and Next Steps

7.1. Conclusions

Dissecting this thesis, it can be broken down into four phases. The first phase was research on the concept and idea, as well as research on how the idea should be implemented. Phase two was building and testing the hardware portion, or the cubes. This required a multidisciplinary skill set to achieve; electrical schematics, mechanical drawings, and software coded in C. Phase three was the portal software and development. Aside from utilizing computer science concepts to program the portal, knowledge of chemistry was vital to ensure correct equations and procedures, as well as knowledge of video and photograph editing software to correctly size and modify graphics for the portal. Finally, phase four entailed testing the thesis on actual participants to gather feedback on the quality and reliability of both the idea and implementation. In this case, both the concept and implementation were viewed positively, thus both were successful.

Summarizing the research phase, it was apparent that tangible user interfaces are on the rise in a variety of applications. Museums are beginning to leverage TUI's as a mode of making visitors feel more engaged in the exhibits. Factories are always evolving to find better methods for human to machine interface. Finally, chemistry is a subject matter that would absolutely benefit from being taught virtually, such as in the Chemist App.

Researching and implementing the hardware was a different animal. Utilizing the real equipment in a chemistry lab, each element has a unique application that can be tricky to emulate with sensors. Additionally, there are hundreds of sensors to choose from, also making the research on each particular component that much more vital. Although bench top testing and breadboarding may be successful, there is always a chance the device will fail to meet expectations when practically implemented.

As for the software, using the Raspberry Pi provided challenges due to the limitations of the Linux operating system. One of these limitations was that the portal had to be coded in python. Although there is nothing inherently wrong with the python language, the amount of available libraries for creating a GUI is significantly smaller than with more developed languages such as Java or C++. Additionally, designing the optimal layout and transitional elements of the GUI from screen to screen were difficult hurdles in their own respect.

The process of testing this thesis on the participants was as follows. A demonstration video was created and posted online to attract attention to the concept, which in turn brought willing participants to the testing lab. On average, ten to twenty minutes was spent per participant to read and complete the surveys, as well as complete the lab. The participants would do the post survey, then with minimal guidance conduct the lab. Upon completion, they would fill the post survey. Data gathered will be vital in assessing what the next steps of this project should be.

7.2. **Recommendations**

Moving forward, the proof of concept should become a reality with a variety of improvements learned upon from the extensive testing and research put into the proof of concept. Mechanically, the cubes should be made out of a higher grade material than the 3D printing glue that was used. For example, polystyrene is a plastic that is significantly less brittle and commonly found in today's children toys. Additionally, pressure fitting pieces together was not successful, thus a better method of locking the cubes pieces together should be used. For instance, having screw holes or clip locks would be a better choice.

Electrically, the devices could be fitted with Wi-Fi chips to make each cube wireless. This would also require each cube has a battery, thus a charging port or removable battery option should be added. Lithium Ion batteries are commonly found in modern devices as a means of rechargeable energy. The PCB could also be designed using surface mount parts, ensuring the cost and space used is minimized. Finally, the portal could be designed to be sizeable to any screen resolution, and a cube could be developed to remove the need for navigating by touch screen or mouse click. This would open the doors for the device to be able to interface with a holographic virtual laboratory.

REFERENCES

[1] T. C. Board, "STUDENT SCORE DISTRIBUTIONS AP EXAMS 2016," 2017. [Online]. Available: https://apscore.collegeboard.org/scores/about-ap-scores/score-distributions//.

[2] Belinky, Lanir, and Kuflik, "Using Hand Held Devices and Situated Displays for a Collaborative Planning of a Museum Visit," 2012. [Online]. Available: http://dl.acm.org/citation.cfm?id=2307817.

[3] Authors, Numerous, "Tangible Menus and Interaction Trays: Core Tangibles for Common Physical/ Digital Activities," 2008. [Online]. Available: http://dl.acm.org/citation.cfm?id=1347436.

[4] Schlomer, Poppinga, Henze, and Boll, "Schlomer, Poppinga, Henze, Boll, Gesure Recognition with a Wii Controller," 2008. [Online]. Available: http://dl.acm.org/citation.cfm?id=134795.

[5] S. R. Costanza, "Audio D-Touch: A Tangible User Interface for Musical Composition and Performance," 2003. [Online]. Available: https://eprints.soton.ac.uk/270957/1/dafx.pdf.

[6] I. W. K. S. L. Sharlin, "Cognitive Cubes: A Tangible User Interface for Cognitive Assessment," 2002. [Online]. Available: http://dl.acm.org/citation.cfm?id=503438.

[7] B. Wimmer, "HandSense: Discriminating Different Ways of Grasping and Holding a Tangible User Interface," February 2009. [Online]. Available: http://dl.acm.org/citation.cfm?id=1517736.

[8] G. M. U. O. Shaer, "Tangible Interaction With Large Data Sets Using Active Tokens," 2015. [Online]. Available: http://cs.wellesley.edu/~hcilab/publication/physicalData.pdf.

[9] T. A. D. Group, "Chemist," [Online]. Available: https://itunes.apple.com/us/app/chemist-by-thix/id440666387?mt=8.

[10] R. Nimitz, Surgical Instrumentation: an Interactive Approach, 2nd Edition, Saunders, 2013.

[11] M. Silberberg, Chemistry: Molecular Nature of Matter and Change 6th Edition, McGraw-Hill, 2011.

[12] K. Lew, The Alkali Metals: Lithium, Sodium, Potassium, Rubidium, Cesium, Francium, Rosen Publishing Group Inc, 2010.

[13] R. H. Petrucci, General Chemistry: Principles & Modern Applications. 9th Edition, Pearson, 2007.

[14] E. Upton, Raspberry Pi User Guide 4th Edition, Wiley, 2016.

[15] Atmel, 8-bit Atmel Microcontroller with 16/32/64KB In-System Programmable Flash, Datasheet, 2014.

[16] Texas Advanced Optoelectronic Solutions Inc., TCS3472 Color Light-To-Digital Converter with IR Filter, Datasheet, 2012.

[17] R. N. Lussier, Management Fundamentals: Concepts, Applications, & Skill Development, Sixth Edition, SAGE Publications, Inc, 2015.

APPENDIX I: ARDUINO C CODE FOR TUI CUBES

Cube code:

```
#include <Wire.h>
#include "Adafruit_TCS34725.h"

// Variables
************************************************************************************
// Dropper Pins, PWM pins for RGB LED
#define rPinD1 3                    // 1k ohm resistor
#define gPinD1 4                    // 560 ohm resistor
#define bPinD1 5                      // 560 ohm resistor
#define ledPinD1 32                   // Color Sensor LED pin
#define pbPinD1 38                    // Dropper Push Button, active low

// Beaker 1 Pins, PWM pins for RGB LED
#define rPinB1 6                      // 1k ohm resistor
#define gPinB1 7                     // 560 ohm resistor
#define bPinB1 8                     // 560 ohm resistor
#define magSwB1 42                    // Magnetic Reed Switch, active low
#define gyroRefB1 A3                  // Analog pin for the gyroscope reference volts
#define gyroOutB1 A4                  // Analog pin for the gyroscope output volts


// Beaker 2 Pins, PWM pins for RGB LED
#define rPinB2 9                      // 1k ohm resistor
#define gPinB2 10                     // 560 ohm resistor
#define bPinB2 11                     // 560 ohm resistor
#define magSwB2 40                    // Magnetic Reed Switch, active low
#define gyroRefB2 A5                  // Analog pin for the gyroscope reference volts
#define gyroOutB2 A6                  // Analog pin for the gyroscope output volts


// Burner Pins
#define irPin A2                      // IR distance sensor
#define ledBurner 36                  // Red LED on burner
#define pbBurnerPin 28                // Burner on/off switch, active low
#define counterPin  34                 // LED showing counting

#define commonAnode false               // Ground for common cathode

// Variables for Gyroscope
#define gyroFactor 0.67               // Gyro conversion factor is 0.67
#define gyroMaxMove 300               // define max move at 300 degrees / second
float gyroMoveB1, gyroMoveB2;           // Detects move, degrees per second
float refB1, refB2;                 // Gyro reference voltage
float outB1, outB2;                 // Gyro output voltage

// Variables for Burner
#define burnerDistance 500              // Distance to trigger counter
#define counterMaxVal 30              // Count max, rate of 6 per second

// RGB -> eye-recognized gamma color
byte gammatable[256];

// Saves last color
```

```
int s1, s2, s3, colorOut;
int mag1, mag2;
int clearG1, clearG2;
int countDone;
int irDistance, flameOn;

// For PB state machine
int buttonState = 0;
int lastButtonState = 0;
int state = 0;

// Variables for timer
static long totalMillis, totalSec, totalMin, totalHour;
static long currentMillis, currentSec, currentMin, currentHour;
int counter = 0;

Adafruit_TCS34725 tcs = Adafruit_TCS34725(TCS34725_INTEGRATIONTIME_50MS,
TCS34725_GAIN_4X);

// Microcontroller setup (Arduino MEGA2560)
********************************************************
void setup() {
  Serial.begin(9600);
  Serial.println("Start");

  if (tcs.begin()) {
    Serial.println("Found sensor");
  }
  else {
    Serial.println("No TCS34725 found ... check your connections");
    //while (1); // If no color sensor, don't run
  }

  // Define all pins
  pinMode(rPinD1, OUTPUT);                // Pin  3
  pinMode(gPinD1, OUTPUT);                // Pin  5
  pinMode(bPinD1, OUTPUT);                // Pin  6
  pinMode(rPinB1, OUTPUT);                // Pin  9
  pinMode(gPinB1, OUTPUT);                // Pin 10
  pinMode(bPinB1, OUTPUT);                // Pin 11
  pinMode(rPinB2, OUTPUT);                // Pin  9
  pinMode(gPinB2, OUTPUT);                // Pin 10
  pinMode(bPinB2, OUTPUT);                // Pin 11
  pinMode(pbPinD1, INPUT);              // Pin 13
  pinMode(ledPinD1, OUTPUT);              // Pin  4
  digitalWrite(ledPinD1, LOW);            // Initialize sensor as off
  pinMode(magSwB1, INPUT);                // Pin 12
  pinMode(magSwB2, INPUT);                // Pin 12
  pinMode(ledBurner, OUTPUT);           // Pin
  digitalWrite(ledBurner, HIGH);          // Initialize as off (active low)
  pinMode(pbBurnerPin, INPUT);            // Pin
  pinMode(counterPin, OUTPUT);            // Pin
  digitalWrite(counterPin, LOW);          // Initialize as off

  // Gamma Table
  for (int i=0; i<256; i++) {           // 8 bit
    float x = i;
```

```
    x /= 255;
    x = pow(x, 2.5);
    x *= 255;

    if (commonAnode) {                // Changes based on the type of RGB LED used
      gammatable[i] = 255 - x;
    } else {
      gammatable[i] = x;
    }
    //Serial.println(gammatable[i]);
  }
}

// Main Loop ***********************************************************************************
void loop() {
  uint16_t clear, red, green, blue;

  tcs.setInterrupt(false);              // turn on LED
  delay(60);                      // takes 60ms to read
  tcs.getRawData(&red, &green, &blue, &clear);
  tcs.setInterrupt(true);               // turn off LED
/*
  Serial.print("C:\t"); Serial.print(clear);
  Serial.print("\tR:\t"); Serial.print(red);
  Serial.print("\tG:\t"); Serial.print(green);
  Serial.print("\tB:\t"); Serial.print(blue);
*/
  // Figure out some basic hex code for visualization
  uint32_t sum = clear;
  float r, g, b;
  r = red; r /= sum;
  g = green; g /= sum;
  b = blue; b /= sum;
  r *= 256; g *= 256; b *= 256;
  //Serial.print("\t");
  //Serial.print((int)r, HEX); Serial.print((int)g, HEX); Serial.print((int)b, HEX);
  //Serial.println();
/*
  Serial.print((int)r);
  Serial.print(" ");
  Serial.print((int)g);
  Serial.print(" ");
  Serial.println((int)b);
*/
// Dropper Logic ***********************************************************************************
  buttonState = digitalRead(pbPinD1);          // Read PB current state
  mag1 = digitalRead(magSwB1);
  mag2 = digitalRead(magSwB2);

  if(buttonState != lastButtonState ) {         // Compare state to previous state
    if(buttonState == LOW) {              // Active low
      state++;                      // Increment
      if(state > 3) state = 0;              // Reset after 3rd state
    }
    delay(50);                      // Debounce time
```

```
  }

  if(state == 1) {                        // State 1; Read sensor
    s1 = r;
    s2 = g;
    s3 = b;
    digitalWrite(ledPinD1, HIGH);
    LedColor(rPinD1, gPinD1, bPinD1, r, g, b);
  }
  else if(state == 2) {                    // State 2; Read save, send to Dropper LED
    digitalWrite(ledPinD1, LOW);
    LedColor(rPinD1, gPinD1, bPinD1, s1, s2, s3);
    if((mag1 == LOW || mag2 == LOW) & totalMillis % 10) {
      LedColor(rPinD1, gPinD1, bPinD1, 0, 0, 0);
    }
    else {
      LedColor(rPinD1, gPinD1, bPinD1, s1, s2, s3);
    }
  }
  else if(state == 3 & mag1 == LOW){ // State 3; Reset Dropper and send to Beaker 1
    LedColor(rPinB1, gPinB1, bPinB1, s1, s2, s3);
    state = 0;
  }
  else if(state == 3 & mag2 == LOW){ // State 3; Reset Dropper and send to Beaker 2
    LedColor(rPinB2, gPinB2, bPinB2, s1, s2, s3);
    state = 0;
  }
  else {                              // State 0; Reset Dropper LED
    digitalWrite(ledPinD1, LOW);
    LedColor(rPinD1, gPinD1, bPinD1, 0, 0, 0);
    state = 0;
  }
/*
  Serial.print(!buttonState);
  Serial.print("_");
  Serial.print(s1);
  Serial.print("_");
  Serial.print(s2);
  Serial.print("_");
  Serial.print(s3);
  Serial.print("_\n");
  */

// Gyroscope Logic *********************************************************************************
// Beaker 1 Gyroscope
  refB1 = analogRead(gyroRefB1);              // Read reference voltage
  outB1 = analogRead(gyroOutB1);              // Read output voltage
  gyroMoveB1 = abs(refB1 - outB1);            // Difference is move
  gyroMoveB1 /= gyroFactor;                   // Converts to degrees (from Datasheet)

  if(gyroMoveB1 < gyroMaxMove) {             // Don't detect smaller moves
    gyroMoveB1 = 0.0;
    clearG1 = 0;
  }
  else {                                 // Empty or pour beaker
    LedColor(rPinB1, gPinB1, bPinB1, 0, 0, 0);
```

```
    clearG1 = 1;
    //Serial.print(" Angle G1: ");
    //Serial.println(gyroMoveB1);
  }

// Beaker 2 Gyroscope
  refB2 = analogRead(gyroRefB2);                // Read reference voltage
  outB2 = analogRead(gyroOutB2);                // Read output voltage
  gyroMoveB2 = abs(refB2 - outB2);              // Difference is move
  gyroMoveB2 /= 0.67;                           // Converts to degrees (from Datasheet)

  if(gyroMoveB2 < gyroMaxMove) {                  // Don't detect smaller moves
    gyroMoveB2 = 0.0;
    clearG2 = 0;
  }
  else {                                    // Empty or pour beaker
    LedColor(rPinB2, gPinB2, bPinB2, 0, 0, 0);
    clearG2 = 1;
    //Serial.print(" Angle G2: ");
    //Serial.println(gyroMoveB2);
  }

  delay(50);                          // 50ms delay to help processing speed

// Burner Logic
*********************************************************************************
  irDistance = analogRead(irPin);
  flameOn = digitalRead(pbBurnerPin);
  if(flameOn == LOW) {        // If switch is triggered
    digitalWrite(ledBurner, HIGH);            // Turn on light
    if(irDistance > burnerDistance & currentSec % 2 == 0 & counter != counterMaxVal) {
      digitalWrite(counterPin, HIGH);    // If correct height, even time, counter low, count led
      counter++;                        // Increment counter (6 per second)
    }
    else {
      digitalWrite(counterPin, LOW);
    }
  }
  else {                          // Switch is off, so no lights, clear counter
    digitalWrite(ledBurner, LOW);
    digitalWrite(counterPin, LOW);
    counter = 0;
  }

  if(counter == counterMaxVal) {          // Counter triggered, react and keep light on
    digitalWrite(counterPin, HIGH);
    countDone = 1;
    //ColorReaction(rPinB1, gPinB1, bPinB1, s1, s2, s3);
  }
  else {
    countDone = 0;
  }
 // Serial.println(" \nIR ");
 // Serial.print(analogRead(irPin));
```

```
// Serial.println();
// Serial.println(digitalRead(magSwB2));

 Clock();                          // Run the clock
 lastButtonState = buttonState;             // Record new last state for dropper button

 if(s1 > 54 & s1 < 62 & s2 > 110 & s2 < 118 & s3 > 57 & s3 < 65) {
   colorOut = 1;   // Green
 }
 else if(s1 > 49 & s1 < 57 & s2 > 114 & s2 < 122 & s3 > 111 & s3 < 119) {
   colorOut = 2;   // Cyan
 }
 else if (s1 > 110 & s1 < 130 & s2 > 66 & s2 < 74 & s3 > 46 & s3 < 54) {
   colorOut = 3;   // Red
 }
 else {
   colorOut = 0;
 }

 Serial.print(!buttonState);
 Serial.print("_");
 Serial.print(colorOut);
 Serial.print("_");
 Serial.print(colorOut);
 Serial.print("_");
 Serial.print(colorOut);
 Serial.print("_");
 Serial.print(!mag1);
 Serial.print("_");
 Serial.print(clearG1);
 Serial.print("_");
 Serial.print(!mag2);
 Serial.print("_");
 Serial.print(clearG2);
 Serial.print("_");
 Serial.print(!flameOn);
 Serial.print("_");
 Serial.println(countDone);

}

// Sub Functions ***************************************************************************************
// Function to write to LED based on pin and gamma table color
void LedColor(int r, int g, int b, int x, int y, int z){
   analogWrite(r, gammatable[(int)x]);       // Red pin
   analogWrite(g, gammatable[(int)y]);       // Green pin
   analogWrite(b, gammatable[(int)z]);       // Blue pin
}

// Function using system clock to determine time
void Clock(){
 totalMillis = millis();                        // Uses time controller has been on
 totalSec = totalMillis / 1000;
 currentSec = totalSec % 60;
 totalMin = totalSec / 60;
```

```
  currentMin = totalMin % 60;
}

// Function to create color reaction when under bunsen burner for alloted time
void ColorReaction(int r, int g, int b, int x, int y, int z) {
  if(r > g & r > b){                // If red is predominant
    LedColor(r, g, b, 255, 0, 0);
  }
  else if(g > r & g > b) {            // If green is predominant
    LedColor(r, g, b, 0, 255, 0);
  }
  else if(b > r & b > g) {            // If blue is predominant
    LedColor(r, g, b, 0, 0, 255);
  }
  else {                          // Other flips colors
    LedColor(r, g, b, z, x, y);
  }
}

```

APPENDIX II: PYTHON CODE FOR CHEMISTRY APP PORTAL

Portal code:

```
# Author:   Frank Pellicano
# Title:    Chemistry Portal for Thesis
# Started:  April 9th, 2017
# Description : Interactive Chemistry Portal, for syncing with hardware cubes
#               User selects which chemistry lab they would like to do, then
#               manipulates hardware cubes according to on screen instructions.
#               Upon success or failure, video will play showing the real life
#               equivalent of the equipment and reaction. Upon completion, the
#               user is brought back to the main portal to start over.

import tkinter as tk
import serial
import re
import os

XLARGE_FONT = ("Verdana", 58)                   # Define standard large font
LARGE_FONT = ("Verdana", 12)                    # Define standard large font
NORM_FONT = ("Verdana", 10)                     # Define standard large font
SMALL_FONT = ("Verdana", 8)                     # Define standard large font
WINDOW = "1024x700"

ser = serial.Serial('/dev/ttyACM0', 9600)       # Arduino port, 9600 Baud rate


#****************************************************************************************
***********
def QueryArduino():                             # Read arduino data from serial port
    v = ser.readline()
    if(str(v).find("Begin") != -1):             # If data does not contain begin
        print("V:", v)                          # Print it
    else:
        v = re.sub("\D", "", str(v))            # Else delete it

    print(v)
    return v                                    # Reads one line at a time from serial buffer

def QueryState(a, b, val):                      # Start at a, to b, value to check for
    ser.flushInput()                            # Flush buffer before doing a new read
    v = QueryArduino()                          # Read serial data

    print("Initial state: ", v[a:b])            # Check the initial bit
    while(v[a:b] != val):
        v = QueryArduino()                      # Keep reading until bit is high
    print(v)
    print("New State: ",(v[a:b]))               # Check bit's after value
    ser.flushInput()
# x_x_x_x_x_x_x_x_x_x       01, 12, 23, 34, 45, 56, 67, 78, 89, 910
# Dropper _ R _ G _ B _ Mag1 _ Gyro1 _ Mag2 _ Gyro2 _ Burner _ IR

def PlayVid(y):
    os.system("omxplayer -o hdmi /home/pi/Desktop/ThesisGame/"+y)

```

```
def popupmsg(msg):                              # Create a popup message
    popup = tk.Tk()
    popup.wm_title("!")
    popup.wm_geometry("200x100")
    label = tk.Label(popup, text = msg, font = NORM_FONT)
    label.pack(side="top", fill="x", pady = 10)
    button1 = tk.Button(popup, text = "Okay", command = popup.destroy)
    button1.pack()
    popup.mainloop()                            # Popup is an independent loop from the main loop


#***************************************************************************************************
class ChemThesisApp(tk.Tk):     # Parent class

    def __init__(self, *args, **kwargs):        # initial method
        tk.Tk.__init__(self, *args, **kwargs)   # Initialize self
        self.wm_title("Thesis Chemistry App")   # Add title
        self.wm_geometry(WINDOW)              # Set the window size
        self.resizable(width=False, height=False)
        self.wm_iconbitmap(bitmap = "@chemicon.xbm")    # Set program icon

        container = tk.Frame(self)              # Creates the frame of the window
        container.pack(side = "top", fill = "both", expand = True)  # Fills entire container space
        container.grid_rowconfigure(0, weight = 1)                  # Configure the grid row
        container.grid_columnconfigure(0, weight = 1)               # Configure the grid column

        self.frames = {}                        # Empty dictionary for all of the program frames

        for F in (StartPage, Page001, Page002, Page003, Page004, Page005,
                  Page006, Page007, Page008, Page009, Page010,
                  Page011, Page012, Page013, Page014, Page015,
                  Page016, Page017, PreLab1):
            frame = F(container, self)          # The Opening page of the app
            self.frames[F] = frame              # Puts the Start Page in the dictionary
            frame.grid(row = 0, column = 0, sticky = "nsew")        # Align north south east west

        self.show_frame(StartPage)              # Show the start page

    def show_frame(self, cont):                 # Method to show frames
        print(cont)
        frame = self.frames[cont]               # Adds this frame to the dictionary
        frame.tkraise()                         # Raise this to the front
        run = 1
        print(run)


#***************************************************************************************************
```

```
class StartPage(tk.Frame):      # Start page, Inherets all the stuff from the frame

    def __init__(self, parent, controller):
        tk.Frame.__init__(self, parent)         # Initialize the start page

        label = tk.Label(self, text = "Chemistry Start Page!", font = LARGE_FONT)

        button1 = tk.Button(self, text = "Click here to enter the main menu!", font = XLARGE_FONT,
                            wraplength = 820, command = lambda: controller.show_frame(Page001))

        label.pack(padx = 10, pady = 10)        # Packs on the label
        button1.pack(padx = 50, pady = 50, fill =tk.BOTH, expand = True)

#****************************************************************************************************
class Page001(tk.Frame):        # Main Menu

    def __init__(self, parent, controller):
        tk.Frame.__init__(self, parent)         # Initialize Page 1

        label1 = tk.Label(self, text = "Main Menu", font = LARGE_FONT)

        button1 = tk.Button(self, text = "Instructions", height = 5, width = 15,
                            command = lambda: controller.show_frame(Page002))
        button2 = tk.Button(self, text = "About", height = 5, width = 15,
                            command = lambda: controller.show_frame(Page003))
        button3 = tk.Button(self, text = "Choose Lab", height = 5, width = 15,
                            command = lambda: controller.show_frame(Page004))
        button4 = tk.Button(self, text = "Exit",  height = 5, width = 15,
                            command = lambda: controller.show_frame(StartPage))

        label1.pack(padx = 10, pady = 10)        # Packs on the label
        button1.pack(pady = 10)
        button2.pack(pady = 10)
        button3.pack(pady = 10)
        button4.pack(pady = 10, side = "bottom")


#****************************************************************************************************
class Page002(tk.Frame):        # Instructions Page

    def __init__(self, parent, controller):
        tk.Frame.__init__(self, parent)         # Initialize Page 2

        label1 = tk.Label(self, text = "Chemistry Lab Instructions:", font = LARGE_FONT)
        label2 = tk.Label(self, text = instr_txt1, font = NORM_FONT)
        label3 = tk.Label(self, text = instr_txt2, font = SMALL_FONT)

        button1 = tk.Button(self, text = "Back", wraplength = 120,
                            height = 5, width = 15,
                            command = lambda: controller.show_frame(Page001))

        img = tk.PhotoImage(file="giphy.gif")
```

```
        imglb = tk.Label(self, image=img)
        imglb.image = img

        label1.pack(padx = 10, pady = 10)       # Packs on the label
        label2.pack(padx = 10, pady = 5)        # Packs on the label
        imglb.pack(padx = 10, pady = 10)
        label3.pack(padx = 10, pady = 5)        # Packs on the label
        button1.pack(pady = 10, side = "bottom")


#**************************************************************************************************
class Page003(tk.Frame):        # About Page

    def __init__(self, parent, controller):
        tk.Frame.__init__(self, parent)         # Initialize Page 3

        label1 = tk.Label(self, text = "About", font = LARGE_FONT)
        label2 = tk.Label(self, text = about_txt1, font = NORM_FONT)
        label3 = tk.Label(self, text = about_txt2, font = SMALL_FONT)

        button1 = tk.Button(self, text = "Back", wraplength = 120,
                            height = 5, width = 15,
                            command = lambda: controller.show_frame(Page001))

        img = tk.PhotoImage(file="mi.gif")
        imglb = tk.Label(self, image=img)
        imglb.image = img

        label1.pack(padx = 10, pady = 10)       # Packs on the label
        label2.pack(padx = 10, pady = 10)       # Packs on the label
        #imglb.pack(padx = 10, pady = 10, side = "right")
        label3.pack(padx = 10, pady = 10)       # Packs on the label
        button1.pack(pady = 10, side = "bottom")

#**************************************************************************************************
class Page004(tk.Frame):        # Choice Page

    def __init__(self, parent, controller):
        tk.Frame.__init__(self, parent)         # Initialize Page 4

        label1 = tk.Label(self, text = "Please select your lab.", font = LARGE_FONT)
        label2 = tk.Label(self, text = lab_descr, wraplength = 300, font = NORM_FONT)

        button1 = tk.Button(self, text = "Lab 1: Making Table Salt", wraplength = 120,
                            height = 5, width = 15,
                            command = lambda: controller.show_frame(PreLab1))
        button2 = tk.Button(self, text = "Lab 2: The Golden Penny", wraplength = 120,
                            height = 5, width = 15,
                            command = lambda: popupmsg("Not available"))
        button3 = tk.Button(self, text = "Lab 3: An Alkali Explosion", wraplength = 120,
                            height = 5, width = 15,
                            command = lambda: popupmsg("Not available"))
        button4 = tk.Button(self, text = "Back", wraplength = 120,
                            height = 5, width = 15,
```

```
                            command = lambda: controller.show_frame(Page001))

        img = tk.PhotoImage(file="giphy.gif")
        imglb = tk.Label(self, image=img)
        imglb.image = img

        self.grid_columnconfigure(0, minsize=100)
        self.grid_columnconfigure(2, minsize=100)
        self.grid_rowconfigure(0, minsize=100)

        label1.grid(row = 0, column = 3, columnspan = 1, padx = 10, pady = 10)
        label2.grid(row = 1, column = 3, columnspan = 5, padx = 10, pady = 10)
        button1.grid(row = 1, column = 1, padx = 10, pady = 10)
        button2.grid(row = 2, column = 1, padx = 10, pady = 10)
        button3.grid(row = 3, column = 1, padx = 10, pady = 1)
        button4.grid(row = 6, column = 3, columnspan = 1, padx = 10, pady = 10)
        imglb.grid(row = 2, column = 3, rowspan = 4, columnspan = 5, padx = 20, pady = 10)

#***************************************************************************************************
class Page005(tk.Frame):        # Lab 1: Step 1

    def __init__(self, parent, controller):
        tk.Frame.__init__(self, parent)         # Initialize Page 5

        def cmd():
            QueryState(0,1,'1')                 # 0:1 is dropper
            print("exit")
            controller.show_frame(Page006)

        label1 = tk.Label(self, text = "Step 1:", font = LARGE_FONT)
        label2 = tk.Label(self, text = "Activate the Dropper.", font = NORM_FONT)
        label3 = tk.Label(self, text = "Hint: Press the button on the dropper.", font = NORM_FONT)

        button1 = tk.Button(self, text = "Begin Step", wraplength = 120,
                            height = 5, width = 15,
                            command = cmd)
        button2 = tk.Button(self, text = "Exit", wraplength = 120,
                            height = 5, width = 15,
                            command = lambda: controller.show_frame(Page001))

        img = tk.PhotoImage(file="DropperOn.gif")
        imglb = tk.Label(self, image=img)
        imglb.image = img

        label1.pack(padx = 10, pady = 10)
        label2.pack(padx = 10, pady = 10)
        imglb.pack(padx = 10, pady = 10)
        button1.pack(pady = 10)
        label3.pack(padx = 10, pady = 10)
        button2.pack(side = "bottom", pady = 10)

#***************************************************************************************************
class Page006(tk.Frame):        # Lab 1: Step 2
```

```

    def __init__(self, parent, controller):
        tk.Frame.__init__(self, parent)         # Initialize Page 6

        def cmd():
            QueryState(1,2,'1')                 # 1:2 is R
            QueryState(2,3,'1')                 # 2:3 is G
            QueryState(3,4,'1')                 # 3:4 is B
            QueryState(0,1,'1')                 # 0:1 is dropper
            print("exit")
            controller.show_frame(Page007)

        label1 = tk.Label(self, text = "Step 2:", font = LARGE_FONT)
        label2 = tk.Label(self, text = "Select Sodium Hydroxide "
            "using the dropper.", font = NORM_FONT)
        label3 = tk.Label(self, text = "Hint: Check the chemicals sheet, then "
                          "press the button on the dropper.", font = NORM_FONT)

        button1 = tk.Button(self, text = "Begin Step", wraplength = 120,
                            height = 5, width = 15,
                            command = cmd)
        button2 = tk.Button(self, text = "Exit", wraplength = 120,
                            height = 5, width = 15,
                            command = lambda: controller.show_frame(Page001))

        img = tk.PhotoImage(file="DropperOn.gif")
        imglb = tk.Label(self, image=img)
        imglb.image = img

        label1.pack(padx = 10, pady = 10)
        label2.pack(padx = 10, pady = 10)
        imglb.pack(padx = 10, pady = 10)
        button1.pack(pady = 10)
        label3.pack(padx = 10, pady = 10)
        button2.pack(side = "bottom", pady = 10)


#****************************************************************************************************
class Page007(tk.Frame):        # Lab 1: Step 3

    def __init__(self, parent, controller):
        tk.Frame.__init__(self, parent)         # Initialize Page 7

        def cmd():
            QueryState(4,5,'1')                 # 4:5 is Mag1
            QueryState(0,1,'1')                 # 0:1 is dropper
            controller.show_frame(Page008)
            PlayVid("BeakerPour.mp4")

        label1 = tk.Label(self, text = "Step 3:", font = LARGE_FONT)
        label2 = tk.Label(self, text = "Place Sodium Hydroxide into Beaker 1.", font = NORM_FONT)
        label3 = tk.Label(self, text = "Hint: Place the dropper in the slot on "
                          "top of the beaker, then press the dropper button.", font = NORM_FONT)

```

```
        button1 = tk.Button(self, text = "Begin Step", wraplength = 120,
                            height = 5, width = 15,
                            command = cmd)
        button2 = tk.Button(self, text = "Exit", wraplength = 120,
                            height = 5, width = 15,
                            command = lambda: controller.show_frame(Page001))

        img = tk.PhotoImage(file="DropperBeaker.gif")
        imglb = tk.Label(self, image=img)
        imglb.image = img

        label1.pack(padx = 10, pady = 10)
        label2.pack(padx = 10, pady = 10)
        imglb.pack(padx = 10, pady = 10)
        button1.pack(pady = 10)
        label3.pack(padx = 10, pady = 10)
        button2.pack(side = "bottom", pady = 10)

#***************************************************************************************************
class Page008(tk.Frame):           # Lab 1: Step 4

    def __init__(self, parent, controller):
        tk.Frame.__init__(self, parent)            # Initialize Page 8

        def cmd():
            QueryState(0,1,'1')                    # 0:1 is dropper
            print("exit")
            controller.show_frame(Page009)

        label1 = tk.Label(self, text = "Step 4:", font = LARGE_FONT)
        label2 = tk.Label(self, text = "Activate the Dropper.", font = NORM_FONT)
        label3 = tk.Label(self, text = "Hint: Press the button on the dropper.", font = NORM_FONT)

        button1 = tk.Button(self, text = "Begin Step", wraplength = 120,
                            height = 5, width = 15,
                            command = cmd)
        button2 = tk.Button(self, text = "Exit", wraplength = 120,
                            height = 5, width = 15,
                            command = lambda: controller.show_frame(Page001))

        img = tk.PhotoImage(file="DropperOn.gif")
        imglb = tk.Label(self, image=img)
        imglb.image = img

        label1.pack(padx = 10, pady = 10)
        label2.pack(padx = 10, pady = 10)
        imglb.pack(padx = 10, pady = 10)
        button1.pack(pady = 10)
        label3.pack(padx = 10, pady = 10)
        button2.pack(side = "bottom", pady = 10)

#***************************************************************************************************
class Page009(tk.Frame):           # Lab 1: Step 5

```

```
    def __init__(self, parent, controller):
        tk.Frame.__init__(self, parent)         # Initialize Page 9

        def cmd():
            QueryState(1,2,'3')                 # 1:2 is R
            QueryState(2,3,'3')                 # 2:3 is G
            QueryState(3,4,'3')                 # 3:4 is B
            QueryState(0,1,'1')                 # 0:1 is dropper
            print("exit")
            controller.show_frame(Page010)

        label1 = tk.Label(self, text = "Step 5:", font = LARGE_FONT)
        label2 = tk.Label(self, text = "Select Phenolphthalein "
            "using the dropper.", font = NORM_FONT)
        label3 = tk.Label(self, text = "Hint: Check the chemicals sheet, then "
                          "press the button on the dropper.", font = NORM_FONT)

        button1 = tk.Button(self, text = "Begin Step", wraplength = 120,
                            height = 5, width = 15,
                            command = cmd)
        button2 = tk.Button(self, text = "Exit", wraplength = 120,
                            height = 5, width = 15,
                            command = lambda: controller.show_frame(Page001))

        img = tk.PhotoImage(file="DropperOn.gif")
        imglb = tk.Label(self, image=img)
        imglb.image = img

        label1.pack(padx = 10, pady = 10)
        label2.pack(padx = 10, pady = 10)
        imglb.pack(padx = 10, pady = 10)
        button1.pack(pady = 10)
        label3.pack(padx = 10, pady = 10)
        button2.pack(side = "bottom", pady = 10)


#***************************************************************************************************
class Page010(tk.Frame):        # Lab 1: Step 6

    def __init__(self, parent, controller):
        tk.Frame.__init__(self, parent)         # Initialize Page 10

        def cmd():
            QueryState(4,5,'1')                 # 4:5 is Mag1
            QueryState(0,1,'1')                 # 0:1 is dropper
            controller.show_frame(Page011)
            PlayVid("DropperVid.mp4")

        label1 = tk.Label(self, text = "Step 6:", font = LARGE_FONT)
        label2 = tk.Label(self, text = "Place Phenolphthalein into Beaker 1.",
            font = NORM_FONT)
        label3 = tk.Label(self, text = "Hint: Place the dropper in the slot on "
                          "top of the beaker, then press the dropper button.", font = NORM_FONT)

        button1 = tk.Button(self, text = "Begin Step", wraplength = 120,
```

```
                            height = 5, width = 15,
                            command = cmd)
        button2 = tk.Button(self, text = "Exit", wraplength = 120,
                            height = 5, width = 15,
                            command = lambda: controller.show_frame(Page001))

        img = tk.PhotoImage(file="DropperBeaker.gif")
        imglb = tk.Label(self, image=img)
        imglb.image = img

        label1.pack(padx = 10, pady = 10)
        label2.pack(padx = 10, pady = 10)
        imglb.pack(padx = 10, pady = 10)
        button1.pack(pady = 10)
        label3.pack(padx = 10, pady = 10)
        button2.pack(side = "bottom", pady = 10)

#****************************************************************************************************
class Page011(tk.Frame):        # Lab 1: Step 7

    def __init__(self, parent, controller):
        tk.Frame.__init__(self, parent)         # Initialize Page 11

        def cmd():
            QueryState(0,1,'1')                 # 0:1 is dropper
            controller.show_frame(Page012)

        label1 = tk.Label(self, text = "Step 7:", font = LARGE_FONT)
        label2 = tk.Label(self, text = "Activate Dropper. ",
            font = NORM_FONT)
        label3 = tk.Label(self, text = "Hint: Press the button on the dropper.", font = NORM_FONT)

        button1 = tk.Button(self, text = "Begin Step", wraplength = 120,
                            height = 5, width = 15,
                            command = cmd)
        button2 = tk.Button(self, text = "Exit", wraplength = 120,
                            height = 5, width = 15,
                            command = lambda: controller.show_frame(Page001))

        img = tk.PhotoImage(file="DropperOn.gif")
        imglb = tk.Label(self, image=img)
        imglb.image = img

        label1.pack(padx = 10, pady = 10)
        label2.pack(padx = 10, pady = 10)
        imglb.pack(padx = 10, pady = 10)
        button1.pack(pady = 10)
        label3.pack(padx = 10, pady = 10)
        button2.pack(side = "bottom", pady = 10)

#****************************************************************************************************
class Page012(tk.Frame):        # Lab 1: Step 8

    def __init__(self, parent, controller):
```

```
        tk.Frame.__init__(self, parent)         # Initialize Page 12

        def cmd():
            QueryState(1,2,'2')                 # 1:2 is R
            QueryState(2,3,'2')                 # 2:3 is G
            QueryState(3,4,'2')                 # 3:4 is B
            QueryState(0,1,'1')                 # 0:1 is dropper
            controller.show_frame(Page013)

        label1 = tk.Label(self, text = "Step 8:", font = LARGE_FONT)
        label2 = tk.Label(self, text = "Select Hydrochloric Acid. ",
            font = NORM_FONT)
        label3 = tk.Label(self, text = "Hint: Check the chemicals sheet, then "
                          "press the button on the dropper.", font = NORM_FONT)

        button1 = tk.Button(self, text = "Begin Step", wraplength = 120,
                            height = 5, width = 15,
                            command = cmd)
        button2 = tk.Button(self, text = "Exit", wraplength = 120,
                            height = 5, width = 15,
                            command = lambda: controller.show_frame(Page001))

        img = tk.PhotoImage(file="DropperOn.gif")
        imglb = tk.Label(self, image=img)
        imglb.image = img

        label1.pack(padx = 10, pady = 10)
        label2.pack(padx = 10, pady = 10)
        imglb.pack(padx = 10, pady = 10)
        button1.pack(pady = 10)
        label3.pack(padx = 10, pady = 10)
        button2.pack(side = "bottom", pady = 10)

#****************************************************************************************************
class Page013(tk.Frame):        # Lab 1: Step 9

    def __init__(self, parent, controller):
        tk.Frame.__init__(self, parent)         # Initialize Page 13

        def cmd():
            QueryState(6,7,'1')                 # 6:7 is Mag2
            QueryState(0,1,'1')                 # 0:1 is dropper
            controller.show_frame(Page014)
            PlayVid("BeakerPour.mp4")

        label1 = tk.Label(self, text = "Step 9:", font = LARGE_FONT)
        label2 = tk.Label(self, text = "Place Hydrochloric Acid into Beaker 2.  ",
            font = NORM_FONT)
        label3 = tk.Label(self, text = "Hint: Place the dropper in the slot on "
                          "top of the beaker, then press the dropper button.", font = NORM_FONT)

        button1 = tk.Button(self, text = "Begin Step", wraplength = 120,
                            height = 5, width = 15,
                            command = cmd)
        button2 = tk.Button(self, text = "Exit", wraplength = 120,
```

```
                            height = 5, width = 15,
                            command = lambda: controller.show_frame(Page001))

        img = tk.PhotoImage(file="DropperBeaker.gif")
        imglb = tk.Label(self, image=img)
        imglb.image = img

        label1.pack(padx = 10, pady = 10)
        label2.pack(padx = 10, pady = 10)
        imglb.pack(padx = 10, pady = 10)
        button1.pack(pady = 10)
        label3.pack(padx = 10, pady = 10)
        button2.pack(side = "bottom", pady = 10)


#*************************************************************************************
**************
class Page014(tk.Frame):        # Lab 1: Step 10

    def __init__(self, parent, controller):
        tk.Frame.__init__(self, parent)         # Initialize Page 14

        def cmd():
            QueryState(7,8,'1')                 # 7:8 is Gyro2
            controller.show_frame(Page015)
            PlayVid("BeakerTitrate.mp4")

        label1 = tk.Label(self, text = "Step 10:", font = LARGE_FONT)
        label2 = tk.Label(self, text = "Pour Beaker 2 into Beaker 1.  ",
            font = NORM_FONT)
        label3 = tk.Label(self, text = "Hint: Hold beaker 2 over beaker 1 "
                          "and tilt it quickly.", font = NORM_FONT)

        button1 = tk.Button(self, text = "Begin Step", wraplength = 120,
                            height = 5, width = 15,
                            command = cmd)
        button2 = tk.Button(self, text = "Exit", wraplength = 120,
                            height = 5, width = 15,
                            command = lambda: controller.show_frame(Page001))

        img = tk.PhotoImage(file="BeakerOn.gif")
        imglb = tk.Label(self, image=img)
        imglb.image = img

        label1.pack(padx = 10, pady = 10)
        label2.pack(padx = 10, pady = 10)
        imglb.pack(padx = 10, pady = 10)
        button1.pack(pady = 10)
        label3.pack(padx = 10, pady = 10)
        button2.pack(side = "bottom", pady = 10)

#*************************************************************************************
**************
class Page015(tk.Frame):        # Lab 1: Step 11

    def __init__(self, parent, controller):
        tk.Frame.__init__(self, parent)         # Initialize Page 15
```

```python

        def cmd():
            QueryState(8,9,'1')                  # 8:9 is Bunsen Burner
            controller.show_frame(Page016)
            PlayVid("BurnerOn.mp4")

        label1 = tk.Label(self, text = "Step 11:", font = LARGE_FONT)
        label2 = tk.Label(self, text = "Turn on the Bunson Burner. ",
            font = NORM_FONT)
        label3 = tk.Label(self, text = "Hint: Press the switch on the Burner. ", font = NORM_FONT)

        button1 = tk.Button(self, text = "Begin Step", wraplength = 120,
                            height = 5, width = 15,
                            command = cmd)
        button2 = tk.Button(self, text = "Exit", wraplength = 120,
                            height = 5, width = 15,
                            command = lambda: controller.show_frame(Page001))

        img = tk.PhotoImage(file="BurnerOn.gif")
        imglb = tk.Label(self, image=img)
        imglb.image = img

        label1.pack(padx = 10, pady = 10)
        label2.pack(padx = 10, pady = 10)
        imglb.pack(padx = 10, pady = 10)
        button1.pack(pady = 10)
        label3.pack(padx = 10, pady = 10)
        button2.pack(side = "bottom", pady = 10)

#****************************************************************************************************
class Page016(tk.Frame):        # Lab 1: Step 12

    def __init__(self, parent, controller):
        tk.Frame.__init__(self, parent)          # Initialize Page 16

        def cmd():
            QueryState(9,10,'1')                 # 9:10 is IR
            controller.show_frame(Page017)
            PlayVid("BurnerBoil.mp4")

        label1 = tk.Label(self, text = "Step 12:", font = LARGE_FONT)
        label2 = tk.Label(self, text = "Place Beaker 1 over Bunson Burner for 5 seconds. ",
            font = NORM_FONT)
        label3 = tk.Label(self, text = "Hint: the blue light on the burner "
                             "will start to blink when the height is correct.", font = NORM_FONT)

        button1 = tk.Button(self, text = "Begin Step", wraplength = 120,
                            height = 5, width = 15,
                            command = cmd)
        button2 = tk.Button(self, text = "Exit", wraplength = 120,
                            height = 5, width = 15,
                            command = lambda: controller.show_frame(Page001))

```

```
        img = tk.PhotoImage(file="BeakerBurner.gif")
        imglb = tk.Label(self, image=img)
        imglb.image = img

        label1.pack(padx = 10, pady = 10)
        label2.pack(padx = 10, pady = 10)
        imglb.pack(padx = 10, pady = 10)
        button1.pack(pady = 10)
        label3.pack(padx = 10, pady = 10)
        button2.pack(side = "bottom", pady = 10)

#*******************************************************************************************************
class Page017(tk.Frame):        # Lab 1: Step 13

    def __init__(self, parent, controller):
        tk.Frame.__init__(self, parent)         # Initialize Page 16

        label1 = tk.Label(self, text = "Step 13:", font = LARGE_FONT)
        label2 = tk.Label(self, text = "Congratulations! You've made table salt. ",
            font = NORM_FONT)

        button1 = tk.Button(self, text = "Finish", wraplength = 120,
                            height = 5, width = 15,
                            command = lambda: controller.show_frame(Page004))
        button2 = tk.Button(self, text = "Exit", wraplength = 120,
                            height = 5, width = 15,
                            command = lambda: controller.show_frame(Page001))

        img = tk.PhotoImage(file="Result.gif")
        imglb = tk.Label(self, image=img, height = 400)
        imglb.image = img

        label1.pack(padx = 10, pady = 10)
        label2.pack(padx = 10, pady = 10)
        imglb.pack(padx = 10, pady = 10)
        button1.pack(pady = 10)
        button2.pack(side = "bottom", pady = 10)
#*******************************************************************************************************
class PreLab1(tk.Frame):        # Lab 1: Pre Lab

    def __init__(self, parent, controller):
        tk.Frame.__init__(self, parent)         # Initialize Page 16

        label1 = tk.Label(self, text = "Lab 1: Pre Lab", font = LARGE_FONT)

        label2 = tk.Label(self, text =
                          "The task of lab 1 is to create table salt by titrating sodium hydroxide "
                          "with hydrochloric acid. Phenolphthalein is used as an indicator in acid "
                          "\n\nbased titrations. Since we are titrating HNaO with H\u2083ClO, the pink"
                          "solution will dissolve based on the solubility and indicate the mix is "
```

```python
                              "correct.\n\n\n The chemical equation for this lab is as follows: \n\n\n"
                              "HNaO  +  H\u2083ClO  =>  NaCl  +  2H\u2082O \n\n\n"
                              "Clearly, sodium chloride (table salt) is the byproduct of this reaction "
                              "we are most interesed in. However, there is a second byproduct, H\u2082O."
                              "\n\nTo remove this, we can convert the water into a gas by boiling it. The"
                              " melting point of salt is 801\u2070C, while the boiling point of water "
                              "\n\nis 100\u2070C. As a result, the salt is unaffected by the exothermic "
                              "reaction.\n\n\n"
                              "Each step of this lab will walk through the process of manipulating the "
                              "tangible user interface cubes such that the essence of this\n\n"
                              "experiment is captured. These steps are as follows:\n\n "
                              "Add Sodium Hydroxide to Beaker 1\n\n"
                              "Add Phenolphthalein to Beaker1\n\n"
                              "Add Hydrochloric Acid to Beaker 2\n\n"
                              "Titrate Beaker 2 into Beaker 1 until the indicator disappears\n\n"
                              "Hold Beaker 1 over the Bunson Burner to boil out the water\n\n"
                              "On each page, click the button 'Begin Step' after reading the "
                              "instructions. Upon clicking, the program will wait for the correct \n\n"
                              "input from the user before moving to the next step. an image and hint "
                              "is available for each step. ",
                              font = NORM_FONT)
        button1 = tk.Button(self, text = "Let's Go!", wraplength = 120,
                                height = 5, width = 15,
                                command = lambda: controller.show_frame(Page005))


        label1.pack(padx = 10, pady = 10)
        label2.pack(padx = 10, pady = 10)
        button1.pack(side = "bottom", pady = 10)

#*****************************************************************************************************
class PreLab2(tk.Frame):        # Lab 1: Pre Lab

    def __init__(self, parent, controller):
        tk.Frame.__init__(self, parent)         # Initialize Page 16

        label1 = tk.Label(self, text = "Lab 2: Pre Lab", font = LARGE_FONT)

        label2 = tk.Label(self, text =
                              "The task of lab 1 is to create table salt by titrating sodium hydroxide "
                              "with hydrochloric acid. Phenolphthalein is used as an indicator in acid "
```

```
                             "\n\nbased titrations. Since we are titrating HNaO with H\u2083ClO, the pink"
                             "solution will dissolve based on the solubility and indicate the mix is "
                             "correct.\n\n\n The chemical equation for this lab is as follows: \n\n\n"
                             "HNaO  +  H\u2083ClO  =>  NaCl  +  2H\u2082O \n\n\n"
                             "Clearly, sodium chloride (table salt) is the byproduct of this reaction "
                             "we are most interesed in. However, there is a second byproduct, H\u2082O."
                             "\n\nTo remove this, we can convert the water into a gas by boiling it. The"
                             " melting point of salt is 801\u2070C, while the boiling point of water "
                             "\n\nis 100\u2070C. As a result, the salt is unaffected by the exothermic "
                             "reaction.\n\n\n"
                             "Each step of this lab will walk through the process of manipulating the "
                             "tangible user interface cubes such that the essence of this\n\n"
                             "experiment is captured. These steps are as follows:\n\n "
                             "Add Sodium Hydroxide to Beaker 1\n\n"
                             "Add Phenolphthalein to Beaker1\n\n"
                             "Add Hydrochloric Acid to Beaker 2\n\n"
                             "Titrate Beaker 2 into Beaker 1 until the indicator disappears\n\n"
                             "Hold Beaker 1 over the Bunson Burner to boil out the water\n\n"
                             "On each page, click the button 'Begin Step' after reading the "
                             "instructions. Upon clicking, the program will wait for the correct \n\n"
                             "input from the user before moving to the next step. an image and hint "
                             "is available for each step. ",
                             font = NORM_FONT)
        button1 = tk.Button(self, text = "Let's Go!", wraplength = 120,
                              height = 5, width = 15,
                              command = lambda: controller.show_frame(Page005))


        label1.pack(padx = 10, pady = 10)
        label2.pack(padx = 10, pady = 10)
        button1.pack(side = "bottom", pady = 10)

#****************************************************************************************************
class PreLab3(tk.Frame):        # Lab 1: Pre Lab

    def __init__(self, parent, controller):
        tk.Frame.__init__(self, parent)         # Initialize Page 16

        label1 = tk.Label(self, text = "Lab 3: Pre Lab", font = LARGE_FONT)

        label2 = tk.Label(self, text =
```

```
                              "The task of lab 1 is to create table salt by titrating sodium hydroxide "
                              "with hydrochloric acid. Phenolphthalein is used as an indicator in acid "
                              "\n\nbased titrations. Since we are titrating HNaO with H\u2083ClO, the pink"
                              "solution will dissolve based on the solubility and indicate the mix is "
                              "correct.\n\n\n The chemical equation for this lab is as follows: \n\n\n"
                              "HNaO  +  H\u2083ClO  =>  NaCl  +  2H\u2082O \n\n\n"
                              "Clearly, sodium chloride (table salt) is the byproduct of this reaction "
                              "we are most interesed in. However, there is a second byproduct, H\u2082O."
                              "\n\nTo remove this, we can convert the water into a gas by boiling it. The"
                              " melting point of salt is 801\u2070C, while the boiling point of water "
                              "\n\nis 100\u2070C. As a result, the salt is unaffected by the exothermic "
                              "reaction.\n\n\n"
                              "Each step of this lab will walk through the process of manipulating the "
                              "tangible user interface cubes such that the essence of this\n\n"
                              "experiment is captured. These steps are as follows:\n\n "
                              "Add Sodium Hydroxide to Beaker 1\n\n"
                              "Add Phenolphthalein to Beaker1\n\n"
                              "Add Hydrochloric Acid to Beaker 2\n\n"
                              "Titrate Beaker 2 into Beaker 1 until the indicator disappears\n\n"
                              "Hold Beaker 1 over the Bunson Burner to boil out the water\n\n"
                              "On each page, click the button 'Begin Step' after reading the "
                              "instructions. Upon clicking, the program will wait for the correct \n\n"
                              "input from the user before moving to the next step. an image and hint "
                              "is available for each step. ",
                              font = NORM_FONT)
        button1 = tk.Button(self, text = "Let's Go!", wraplength = 120,
                              height = 5, width = 15,
                              command = lambda: controller.show_frame(Page005))


        label1.pack(padx = 10, pady = 10)
        label2.pack(padx = 10, pady = 10)
        button1.pack(side = "bottom", pady = 10)

#*****************************************************************************************
# Long Text Fields
instr_txt1 = ("Simulate pre-determined chemistry labs using a tangible user interface "
    "and watch the reaction on the screen!\n")
```

```

instr_txt2 = ("To play, click \"Choose Lab\" in the main menu. Then, select one of "
    "the labs to attempt. Read the on-screen instructions, click Begin Step and orient "
    "\n\nthe equipment such that it matches the instructions. Upon completion, watch the video "
    "reactions. ")

about_txt1 = ("Thesis made by Frank Pellicano spanning July 2016 to July 2017, per the "
    "\n\nrequirements for ELEC6697 and ELEC6698, Master's Thesis I and II.\n")

about_txt2 = ("The goal of this project was to develop a better, virtual, alternative to "
    "the traditional chemistry lab. The target audience is one whom "
    "\n\nchemistry is not a strong suit. A virtual lab removes all of the "
    "error in a traditional lab setting. No longer is there a need to "
    "\n\nensure precise measurements, be at risk of injury, or be stuck "
    "when certain instruments or elements are unavailable. \n\n"
    "\n\nThis lab has many components spanning a variety of subjects. "
    "Electrical engineering design, analysis, and implementation "
    "\n\nwent into the tangible user interface cubes. Computer engineering "
    "went into the software design of the brains behind the hardware, "
    "\n\nthe Arduino and Raspberry Pi. Computer science went into the actual "
    "coding of the C based arduino and Python based Linux GUI. Mechanical"
    "\n\nengineering went into the design, analysis, and implementation of "
    "the cubes themselves. Chemical engineering went into the development, "
    "\n\ntesting, and exceuting the labs provided. Finally, graphic design, "
    "particularly with video and image editing, was pivitol to the portal.\n\n"
    "\n\nMany thanks to Dr. Christopher Martinez (Computer Engineer, Advisor), "
    "Dr. Tiffany Hesser (Chemical Engineer), Richard Weber (Chemical Engineer), "
    "\n\nand Austin Mathews (Mechanical Engineer) for assistance in their respective"
    "fields as it related to this thesis.")

lab_descr = ("Lab 1: Making Table Salt, walks through the steps of "
    "sodium hydroxide with hydrochloric acid. \n\nLab2: The Golden Penny, uses zinc metal to "
    "plate copper in zinc to create a 'silver' penny. Heating this turns "
    "it 'gold'. \n\nLab3: An Alkali Explosion, mixes potassium and water to "
    "create an explosive reaction.")


app = ChemThesisApp()         # Creates actual app to run
app.mainloop()                # Main loop to run app code

```

APPENDIX III: PRE AND POST SURVEYS

Pre Survey

This is the Pre-Survey per the requirements of the University of New Haven Master's Thesis program. This survey is to be taken *before* attempting the virtual chemistry lab proof of concept.

The thesis, titled *Developing a Virtual Chemistry Laboratory using a Tangible User Interface*, is about modernizing and improving the way chemistry labs are implemented. The traditional method of performing chemistry in a laboratory with real, physical elements can be dangerous, difficult to follow, and prone to error. Instead, the equipment is replaced with electronic equipment, the elements are mixed and combined electronically, and the reactions all happen virtually. This removes the danger, removes the error, and makes the experience of the chemistry lab fun, modern, and effective.

To make this a reality, there are three parts to the concept.

1. The User (who is doing the experiment)
2. The Interface (virtual lab where the experiment happens)
3. The User Input (tangible 'cubes' to interact with the interface)

The end concept is one where the user interacts with electronic 'cubes' to cause a chemistry experiment to happen on a virtual screen. The 'cubes' will look and feel like real chemistry equipment, and the virtual component will accurately depict the chemical reactions that would happen in real life, as well as show information regarding the reaction. In theory, the user should be able to perform a near infinite number of experiments virtually.For all questions, please circle one answer.

1. What is your age?

18– 23 24 – 29 30 – 39 40 – 49 50 - 59 Greater than 59

2. Highest Education Level?

Middle School High School Undergraduate Graduate Doctorate Other

3. Level of Chemistry Experience?

Not experienced **Extremely experienced**
1 2 3 4 5 6 7 8 9 10

4. How practical does the concept of a virtual reality chemistry laboratory sound?

Not practical **Extremely practical**
1 2 3 4 5 6 7 8 9 10

5. How likely would you be to choose this alternative over performing chemistry traditionally?

Not likely **Extremely likely**
1 2 3 4 5 6 7 8 9 10

Post Survey

This is the Post-Survey per the requirements of the University of New Haven Master's Thesis program, course ELEC6698. This survey is to be taken *after* attempting the virtual chemistry lab proof of concept.

The proof of concept, for the thesis titled *Developing a Virtual Chemistry Laboratory using a Tangible User Interface*, is designed to emulate the ideas presented in the pre-survey description. It requires a user, features four electronic cubes that emulate real chemistry equipment, and has a virtual interface that demonstrates the chemistry based on the users input. The cubes are a dropper, Bunsen burner, and two beakers. The user navigates the interface and selects one of three pre-determined training experiments that walk the user through using the cubes. Upon a successful action, a video plays showing the action in real-time.

For all questions, please circle one answer.

12. Were you able to complete the chemistry experiment using the cubes?

No　　Yes, with difficultly　　Yes, with ease

13. How positive would you rate your experience?

Not positive									**Extremely positive**
1	2	3	4	5	6	7	8	9	10

14. How effective does this method seem to be for learning Chemistry?

Not effective									**Extremely effective**
1	2	3	4	5	6	7	8	9	10

15. How well does the proof of concept relate to the actual concept?

Not related									**Extremely related**
1	2	3	4	5	6	7	8	9	10

16. How likely would you be to choose a chemistry course that uses this method of teaching over the traditional chemistry labs?

Not likely									**Extremely likely**
1	2	3	4	5	6	7	8	9	10

17. **How accurate do you feel the dropper 'cube' represented a real life dropper to simulate a good tangible experience when performing the virtual experiment?**

Not Accurate									**Extremely Accurate**
1	2	3	4	5	6	7	8	9	10

18. **How accurate do you feel the beaker 'cubes' represented a real life beaker to simulate a good tangible experience when performing the virtual experiment?**

Not Accurate									**Extremely Accurate**
1	2	3	4	5	6	7	8	9	10

19. **How accurate do you feel the burner 'cube' represented a real life Bunsen burner to simulate a good tangible experience when performing the virtual experiment?**

Not Accurate									**Extremely Accurate**
1	2	3	4	5	6	7	8	9	10

20. **How would you rate the interaction between the dropper and the beaker?**

Poor									**Excellent**
1	2	3	4	5	6	7	8	9	10

21. **How would you rate the interaction between the beaker and the Bunsen burner?**

Poor									**Excellent**
1	2	3	4	5	6	7	8	9	10

22. **If made available in stores, how much would you be willing to pay for this product?**

Less than $25 $25 - $70 $75 - $150 Greater than $150

If you have any comments, suggestions, or ideas, please write them here:

APPENDIX IV: BILL OF MATERIALS

Table 1 shows the item, vendor, part number, quantity, cost, and schematic location.

Item Description	Schematic	Vendor	Part Number	Q	Cost ea	Total
Dropper						
1000 Ohm Resistor	R3	DigiKey	CF14JT1K00CT-ND	1	$0.04	$0.04
560 Ohm Resistor	R4, R5	DigiKey	CF14JT560RCT-ND	2	$0.04	$0.08
10000 Ohm Resistor	R2	DigiKey	CF14JT10K0CT-ND	1	$0.04	$0.04
N/O Momentary Push Button	SW1	DigiKey	EG2025-ND	1	$1.15	$1.15
Common Cathode RGB LED	D1	Amazon	Microtivity IL611	1	$0.46	$0.46
RGB Color Sensor	U1	Adafruit	TCS34725	1	$7.95	$7.95
Magnet	M1	DigiKey	469-1005-ND	1	$0.21	$0.21
Case and PCB				1	$2.00	$2.00
Beaker						
1000 Ohm Resistor	R7, R14	DigiKey	CF14JT1K00CT-ND	2	$0.04	$0.08
560 Ohm Resistor	R8, R9, R12, R13	DigiKey	CF14JT560RCT-ND	4	$0.04	$0.16
10000 Ohm Resistor	R6, R10	DigiKey	CF14JT10K0CT-ND	2	$0.04	$0.08
Reed Switch	SW2, SW3	DigiKey	374-1339-ND	2	$0.49	$0.98
Common Cathode RGB LED	D2, D3	Amazon	Microtivity IL611	2	$0.46	$0.92
Gyroscope Sensor	U2, U3	Amazon	GY-521	2	$6.99	$13.98
Magnet	M2, M3	DigiKey	469-1005-ND	2	$0.21	$0.42
Case and PCB				2	$2.00	$4.00
Burner						
10000 Ohm Resistor	R14	DigiKey	CF14JT10K0CT-ND	1	$0.04	$0.04
N/O Momentary Push Button	SW4	DigiKey	EG2025-ND	1	$1.15	$1.15
Red LED	D4	DigiKey	732-5016-ND	1	$0.25	$0.25
IR Sensor	U4	Adafruit	GP2Y0A21YK0F	1	$9.95	$9.95
Case and PCB				1	$2.00	$2.00
External						
Raspberry Pi	U6	Adafruit	Raspberry Pi Z	1	$10.00	$10.00
Arduino Mega	U5	Amazon	MEGA2560R3	1	$11.99	$11.99
Case and PCB				1	$2.00	$2.00
					Total:	$69.92

Table 2: Final Bill of Materials meeting budget requirements of under $100.