INCREASING USABILITY IN SIMULATED LABS THROUGH THE USE OF USER INTERFACE DESIGN PRINCIPLES

By

JEFFREY A. PHELAN

A Thesis Submitted to The Honors College

In Partial Fulfillment of the Bachelors degree
With Honors in

Computer Science

THE UNIVERSITY OF ARIZONA

AUGUST 2015

Approved by:

Dr. Richard Snodgrass
Department of Computer Science

The University of Arizona Electronic Theses and Dissertations
Reproduction and Distribution Rights Form

Name (Last, First, Middle)
Phelan, Jeffrey, A

Degree title (eg BA, BS, BSE, BSB, BFA):
BS

Honors area (eg Molecular and Cellular Biology, English, Studio Art):
Computer Science

Date thesis submitted to Honors College:
August, 17, 2015

Title of Honors thesis:
INCREASING USABILITY IN SUMULATED LABS THROUGH THE USE OF USER INTERFACE DESIGN PRINCIPLES

The University of Arizona Library Release Agreement

☑ Yes, make my thesis available in the UA Campus Repository!

Student signature: ____________________ Date: ____________________

Thesis advisor signature: ____________________ Date: August 16,2015

□ No, do not release my thesis to the UA Campus Repository.

Student signature: ____________________ Date: ____________________

Abstract

Educating students in the formal sciences is a difficult challenge. It can be especially difficult when attempting to attract new students to a field when the fundamental concepts are not intuitive nor easy to comprehend such as computer science. The Laboratory for Computer Science (LoCuS) software system was designed to address the issue of teaching students difficult theoretical concepts that usually require a deep technical background. LoCuS teaches students computational theory through the use of virtual laboratory experiments that immerse the student in a laboratory desktop environment. As proficient as LoCuS is at teaching students through laboratory experiments, it could be improved upon by increasing the usability of the system. The focus of this thesis is to improve upon LoCuS through the use of insights from user interface research. By thoughtfully using insights from user interface design principles to enhance a virtual laboratory, it becomes possible to make computer science labs more immersive, to better approximate a physical laboratory experiment and increase usability for students.

Contents

Chapter 1

Introduction

Educating young minds in the disciplines of science has been a continual challenge for institutions and educators. This is particularly true when the aim is to attract potential students into a field of formal scientific study that is not necessarily intuitive nor easily comprehended such as computer science. One of the major challenges facing the formal sciences is finding a way to present field-specific laws, theories, and principles to entry-level students without overwhelming them with technical aspects that might not be necessary to understand the topic of study. This circumstance holds particularly true for the field of computer science where the students must learn programming, a foreign concept to most, alongside abstract fundamental theories in order to understand them. In other scientific fields, theory is typically taught through the use of engaging demonstrations and experiments that are concrete. One common example is seen in chemistry classes where the students will participate in a experiment to shrink a balloon filled with air by submerging the balloon in liquid nitrogen and condensing the air into a liquid, thus shrinking the balloon. Through experiments such as this one, students receive a visual affirmation of the scientific laws or principles being demonstrated along with the subsequent educational material needed to understand what has happened in the experiment. Herein lies the difficulty with presenting a formal science such as computer science. How can computer science be presented to students in a similarly engaging way that not only has the potential to attract students, but also demonstrates fundamental concepts of the field? One potential solution to this dilemma is the LoCuS software system.

The Laboratory for Computer Science (LoCuS) is a University of Arizona project that was designed to tackle the challenges of teaching entry level students the fundamentals of computer science theory [7]. As mentioned, the difficulty in teaching formal science theories lies in presenting them without requiring a significant amount of prerequisite technical knowledge to understand them. To address this challenge, the LoCuS system was designed to emulate a laboratory environment in which the students are given all of the information they need to fully understand the computing theories presented. This laboratory environment separates the programming code and much of the technical prerequisites from the computing theory so that the students can more easily comprehended the core concepts of each lab. The labs were created to simulate a laboratory environment that a student would typically interact within a physical lab so as to present the student with a familiar experience. In each of the LoCuS labs the student is given an interactive notebook, chalkboard, and range of apparatuses to work with, that demonstrate computing theory concepts or manage data to that end. Through these tools LoCuS is capable of conveying the theoretical principles that would otherwise required some form of programming or technical knowledge. As it stands, LoCuS is a functional tool that can be used to educate a student on a computing theory; however, there is still room to improve the laboratory environment and make it more immersive to better increase the usability of the tool for students.

We posit that it is possible to increase the usability of an emulated laboratory environment by applying insights from traditional user interface design principles. Within a simulated environment, all of the components are digital and as such can be enhanced through insight from traditional user interface designs layered within the simulation. As an example, consider a virtual lab notebook with pages and sections. Using insights from user interface design principles, it is feasible to give the user a better means of document traversal while maintaining the continuity of the virtual notebook. One method of accomplishing document traversal can be found in Section 4.5. There is also a considerable body of user interface research on how to properly display virtual documents so that the document is usable while still allowing the user to remain immersed in the environment. By thoughtfully using insights from user interface design principles to enhance a virtual laboratory, it becomes possible to make computer science labs more immersive, to better approximate a physical laboratory experiment and increase usability for students.

The body of this thesis is divided into six chapters.

- **Approach** covers the current LoCuS environment, the procedure use to come up with the suggested enhancements, and the procedure use to determine what suggestions are implemented and how the implementations are evaluated.
- **Previous Work** covers the user interface literature and research that was used to determine that enhancing the LoCuS interface would increase its usability. This chapter also covers the user interface literature and research that was used to generate the suggested enhancements.
- **Suggested Enhancements** gives a detailed description of each suggested enhancement and explains any reasoning behind them that was not already covered in Section 3.
- **Implementation** describes how the chosen suggested enhancements from Section 4 were implemented.
- **Evaluation** evaluates the chosen enhancement to determine the extent to which they improved usability in LoCuS.
- **Future Work** describes possible future works based on the work done in this thesis.

Chapter 2

Approach

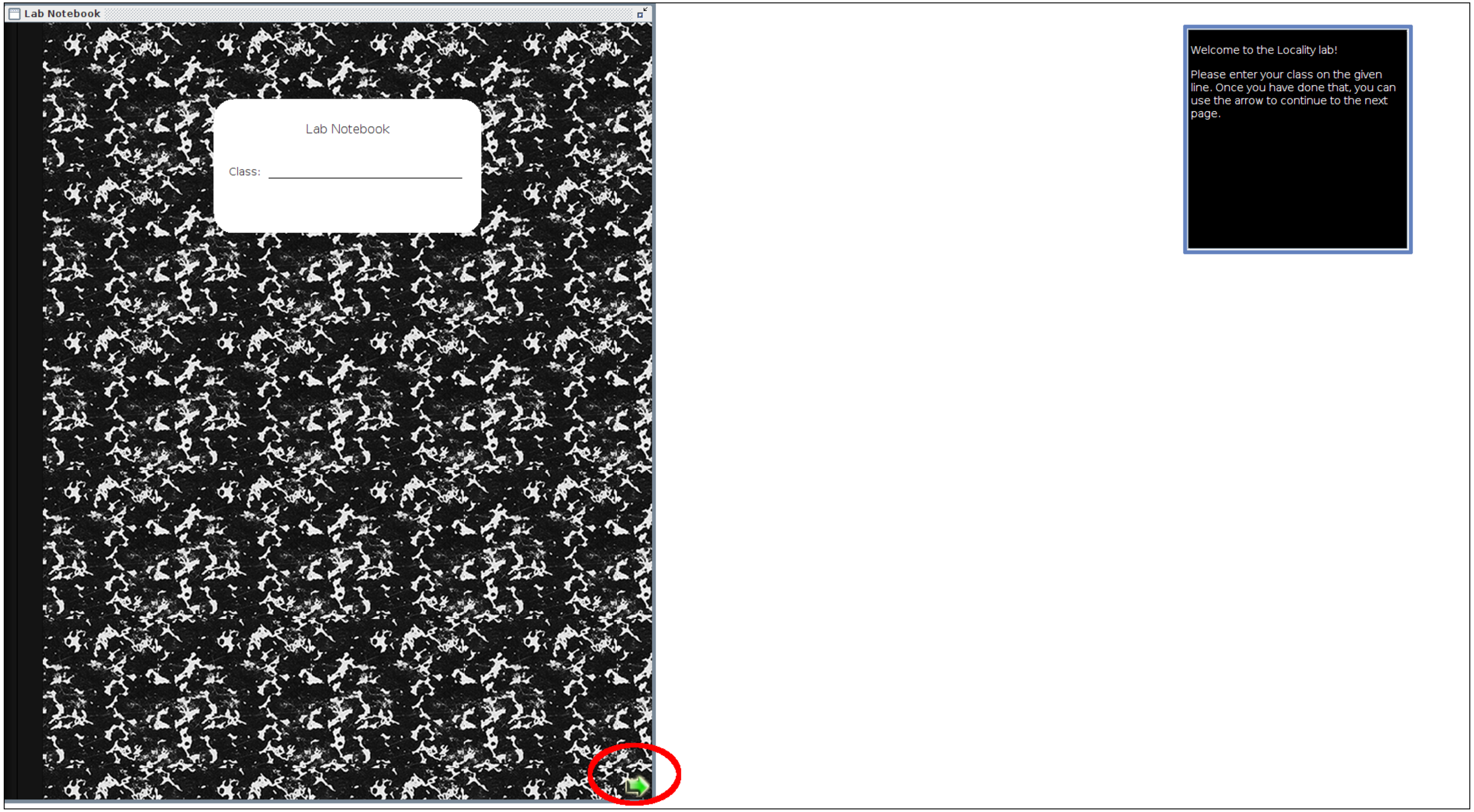
This chapter briefly describes the current state of the LoCuS software system, the steps taken to determine if any changes can be made to improve LoCuS's usability and aesthetics, and the implementation and evaluation of the suggestions. Section 2.1 summarizes the current LoCuS environment and its various parts. When describing LoCuS, some features, such as the student and instructor web portals, have not been included. These features were excluded since they are not affected by any of the suggestions given in Chapter 4. Section 2.2 describes the steps taken to derive the suggestions including why the suggested changes were chosen as apposed to other possible improvements. Implementation and evaluation are discussed in Section 2.3, which covers how implementation of the suggested enhancements are handled along with several ways that these implementations can be evaluated.

Throughout the rest of this document we refer to the different parts of LoCuS as components. Specifically, the notebook, chalkboard, background, and apparatuses are considered components. Although the background does not necessarily appear to be a component, it is the main container for all of the other components and is distinct at the software level. Each component has some attributes. For example, the notebook component has attributes size, location, boarder, texture, text, and navigation among others.

2.1 The Current LoCuS Environment

The LoCuS system is a large sophisticated software infrastructure designed to support laboratory experiments that use a collection of apparatuses along with common components that are present in every lab. The goal of the system is to teach students fundamental principles of computing theory. The laboratory experiments are written by lab authors and described in XML files that are processed by the LoCuS infrastructure. LoCuS will populate the common components, such as the notebook, with text, images, and launcher buttons that launch the apparatuses used to teach that specific lab. The entire infrastructure, including the apparatuses, labs, and build files, has a total of 97,807 lines of Java and XML code excluding white space. There is a total of thirteen labs and fifteen apparatuses within LoCuS. Shown in Figures 2.1, 2.2, and 2.3; all of the LoCuS components use the default Java theming and some apparatuses also include connector sockets to connect them to other apparatuses.

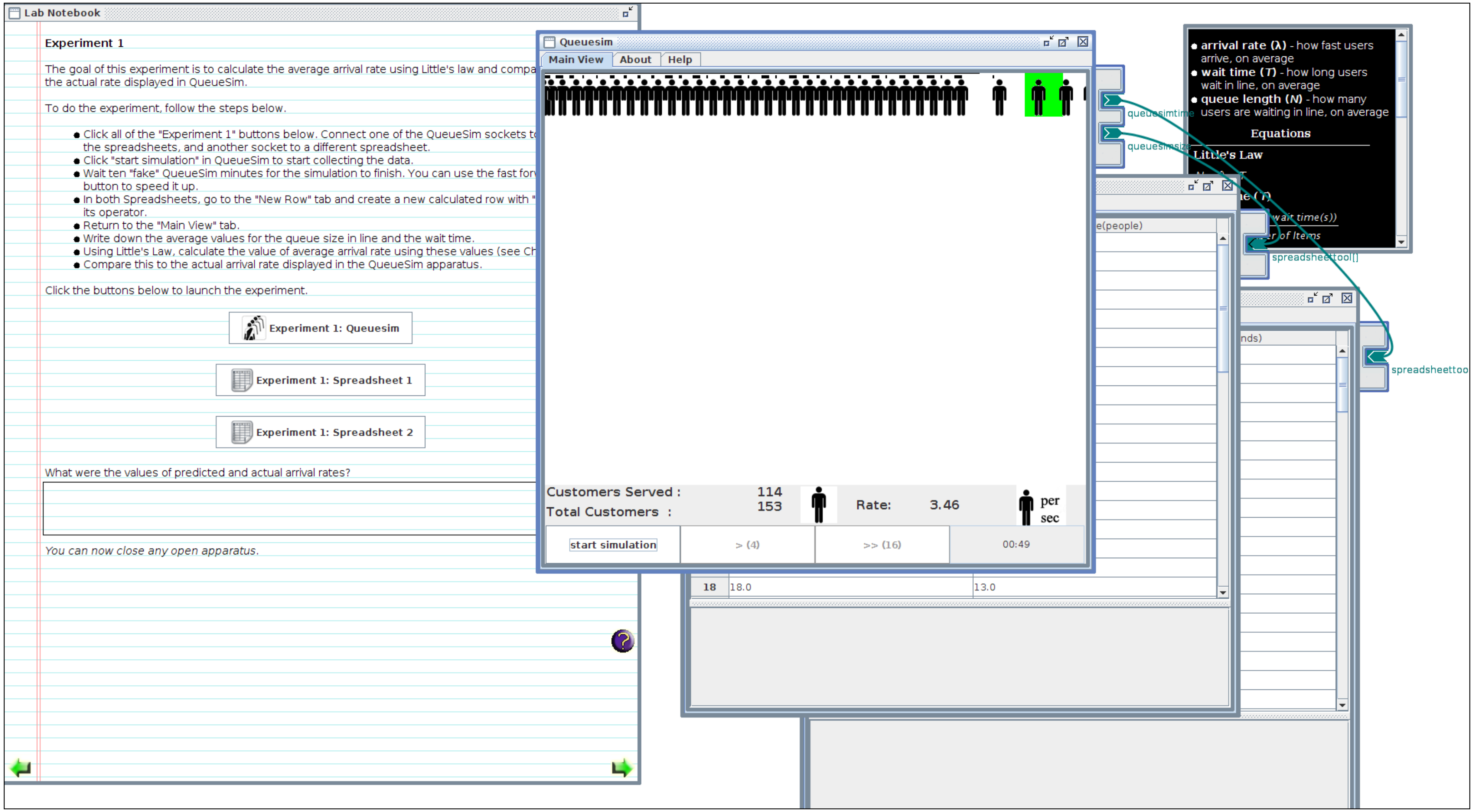
To complete a laboratory experiment in LoCuS the student must go through a multi-step process. When the student starts a lab he is presented with a full screen environment that has the notebook and the chalkboard as seen in Figure 2.1. The student enters their student ID and begins reading through the notebook. Every time the student moves to the next page the notebook is updated along with the chalkboard if necessary. When the student reaches a point in the lab where an apparatus is needed to explain or demonstrate a topic, there is a launcher button to click. Clicking the launcher launches an apparatus that appears over all of the other elements in LoCuS, shown in Figure 2.2. From there the student follows the instructions in the notebook and chalkboard regarding the apparatus. The student proceeds through the rest of the notebook in the same manner until he has reached the end and finish the lab. A sample of some of the apparatuses can be seen in Figure 2.3 for visual reference.



Welcome to the Locality lab!

Please enter your class on the given line. Once you have done that, you can use the arrow to continue to the next page.

Figure 2.1: This is how LoCuS appears when a student first starts the lab. Within the notebook cover a student can login to the lab and begin traversing the notebook with the forward button (a green arrow, circled in red). The black box in the upper right corner is the chalkboard which displays relevant information to the students as they progress through the lab. Notice that both the notebook and the chalkboard use the default Java windowing theme.



Experiment 1

The goal of this experiment is to calculate the average arrival rate using Little's law and compa
the actual rate displayed in QueueSim.

To do the experiment, follow the steps below.

- Click all of the "Experiment 1" buttons below. Connect one of the QueueSim sockets t
  the spreadsheets, and another socket to a different spreadsheet.
- Click "start simulation" in QueueSim to start collecting the data.
- Wait ten "fake" QueueSim minutes for the simulation to finish. You can use the fast for
  button to speed it up.
- In both Spreadsheets, go to the "New Row" tab and create a new calculated row with "
  its operator.
- Return to the "Main View" tab.
- Write down the average values for the queue size in line and the wait time.
- Using Little's Law, calculate the value of average arrival rate using these values (see Ch
- Compare this to the actual arrival rate displayed in the QueueSim apparatus.

Click the buttons below to launch the experiment.

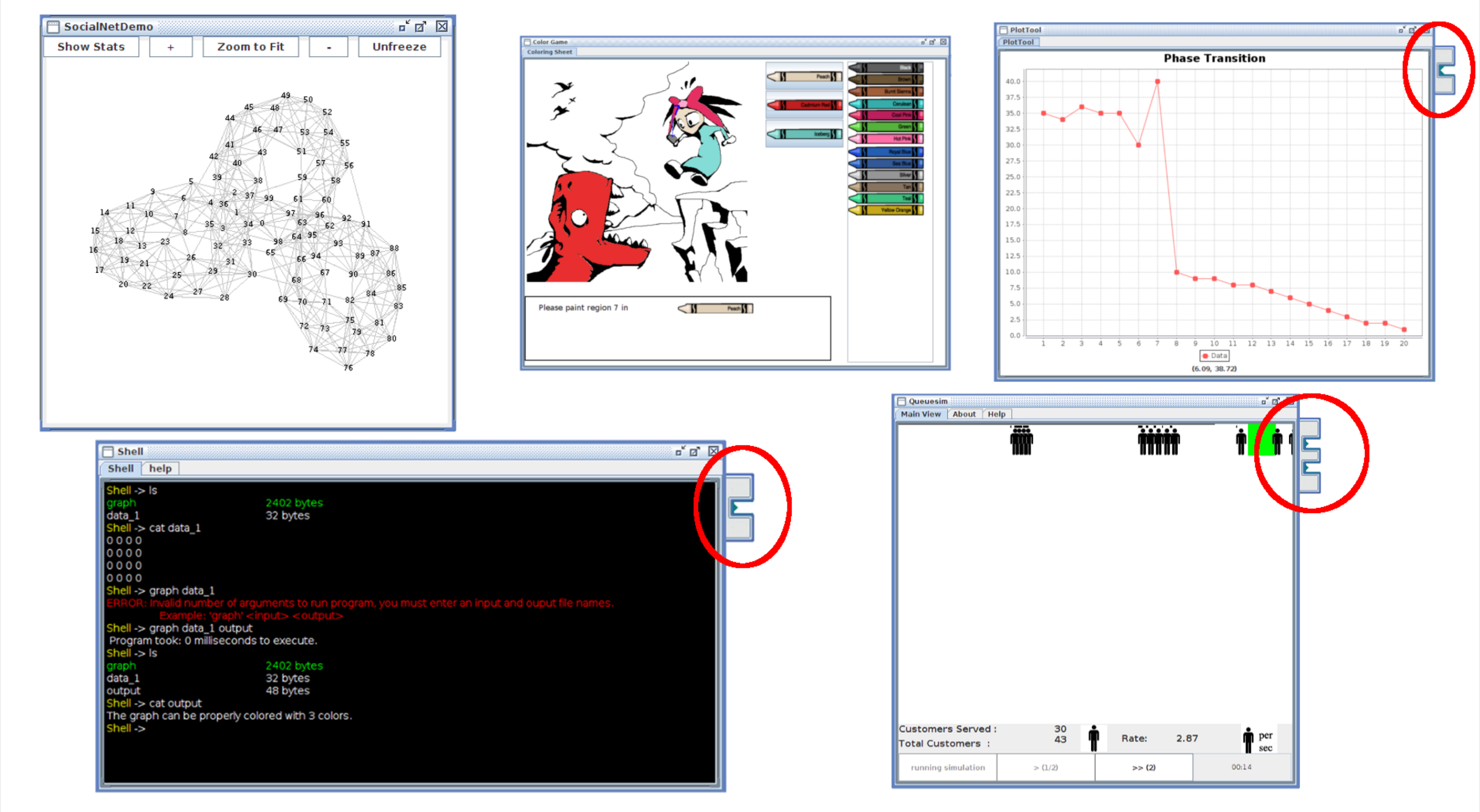
What were the values of predicted and actual arrival rates?

You can now close any open apparatus.

Customers Served : 114
Total Customers : 153
Rate: 3.46 per sec

- **arrival rate (λ)** - how fast users arrive, on average
- **wait time (T)** - how long users wait in line, on average
- **queue length (N)** - how many users are waiting in line, on average

Figure 2.2: This is how LoCuS appears to a student in the middle of a lab, with three apparatuses open and connected to each other, indicated with the teal lines.



```
Shell -> ls
graph                          2402 bytes
data_1                         32 bytes
Shell -> cat data_1
0 0 0 0
0 0 0 0
0 0 0 0
0 0 0 0
Shell -> graph data_1
ERROR: Invalid number of arguments to run program, you must enter an input and ouput file names.
          Example: 'graph' <input> <output>
Shell -> graph data_1 output
 Program took: 0 milliseconds to execute.
Shell -> ls
graph                          2402 bytes
data_1                         32 bytes
output                         48 bytes
Shell -> cat output
The graph can be properly colored with 3 colors.
Shell ->
```

Figure 2.3: Here are examples of current apparatuses. Notice that they all have the default Java windowing theme and several of them have socket connectors (circled in red) that can be used to connect multiple apparatuses.

2.2 Procedure

The objective of this thesis is to answer the question 'Is it possible that LoCuS could benefit from enhancing its user interface, and in what way?'. From this question we began to research what the effects were when the user interface of an application or software system was enhanced. We needed to determine whether there was any benefit besides greater visual appeal. To this end, we started to investigate research studies, books, and other literature on the benefits of enhancing a user interface. Through studying these sources we found research that states that increasing the aesthetic appeal of an application also increase its usability even though the general functionality of the application has not changed [9]. There was also literature describing how enhancing the user interface can help guide the user through the software environment and make it easier and more appealing to use, thus increasing usability [5]. With this knowledge we can answer the question and state that LoCuS would benefit, in terms of usability, from enhancing its user interface. The next step was to determine what enhancements to make so that we would see the benefits described in the research.

To increase the usability of LoCuS we needed to identify ways in which we could enhance its user interface. Amongst the research literature used to determine if there were any benefits to enhancing a user interface, there were several research studies that could apply to different components of the LoCuS environment. For example there was a research study on different methods of indicating a user's progress through a task and how to effectively convey progress [4]. This research directly related to how we could indicate a student's current progress through a LoCuS lab. Other useful literature included a book on designing interfaces using interface design patterns and papers on the effects of multimedia on learning [6, 8]. We used this literature to examine the different components and component attributes in LoCuS. When examining the components and attributes we were able to find design patterns that would correlate in a similar fashion to that of the user progress research and user progress in LoCuS. With multiple correlations between parts of LoCuS and the research articles and design patterns, we were able to start making suggestion on how to enhance LoCuS's user interface.

To enhance the LoCuS user interface we needed to determine how to apply the research literature to the components and attributes that we had previously identified. To accomplish this we examined each component and attribute to see how the literature applied to it and if there was a way to implement the suggestions or results of the research. In many cases, making a suggested enhancement was a straightforward process. Section 4.2 is an example of the straightforward application of an interface design pattern. Following the pattern of matching the literature to LoCuS components and attributes we made eight suggested enhancements that are described in Chapter 4.

2.3 Implementation and Evaluation

Implementing the suggested enhancements is determined by priority and available resources. When determining priority of a recommendation, time and impact were considered. Time was a measurement of how long the implementation was expected to take and impact was a assessment of how much the change effects the student. There was also a time constraint that was based on the allotted time available to implement the changes. Finally, all recommendations had to take overall LoCuS project plan into consideration. For example, the LoCuS project was developed in Java and any changes must consider which features are available in which version of Java. This is particularly important when considering whether a recommendation requires a new library such as JavaFX to function properly. After all of the recommendations were evaluated a course of action was chosen and the resulting implementations was recorded in Section 5. Once implemented, the enhancements were evaluated.

To evaluate the implementations we determined if they resulted in an increase in the usability of LoCuS. Answering this posed a difficult problem. How to properly evaluate such changes? There were two possible methods that could have been used in evaluating the result of these implementation. One would be to conduct a study on students and compare an existing lab with one that has the enhancements implemented in it. A more detail description of how a study can be conducted is in Section 7. This method would be time prohibitive and therefore was not used for this thesis. The method we chose to use for evaluating the results was for us to compare the implementations to the literature. The literature states that making these improvements increases the usability or aesthetic value, which also increases usability. If the implementations follow the literature's suggestions and results, then there should be an increase in usability in LoCuS. As an example consider the progress indicator mentioned in Section 2.2. Evaluating whether our implementation was effective would be a matter of comparing our results to the study's suggestions. Does our implementation indicate the user's progress through LoCuS in a manner that agrees with the study. If so then we have conformed to the research suggestions and increased usability, either directly or through enhancing aesthetics.

Chapter 3

Previous Work

The purpose of this chapter is to explain the literature behind the suggested enhancements. A number of books and research papers were used to determine the suggestions in this thesis. All of them are works that are related to user interface design in some way or another. The literature is grouped by their general concepts and a description of why they pertain to the LoCuS software system is included. Section 3.1 covers the literature that was used to determine if enhancing the LoCuS user interface would be beneficial to the students. Section 3.2 covers the literature that was used to make specific suggestions that could be applied to LoCuS.

3.1 Usability and Reasoning

We decided upon three sources when investigating the literature to determine whether enhancing the LoCuS user interface would be beneficial. This previous work includes (i) a study on how aesthetics relate to usability, (ii) a book about how the human mind processes user interface elements, and (iii) a study on how students react to various type of multimedia learning techniques. The literature provided evidence that enhancing a user interface is beneficial and increases usability.

The first source was a study that was conducted to determine the relationship between the perceived beauty of a system and the perceived usability of that system [9]. To test the correlation between beauty and usability the authors created an Automatic Teller Machine (ATM) application with several different visual layouts that participants would use and rate. The participants rated the different layouts by aesthetics, usability, and information conveyed. The study found that there is a high correlation between the user's perceptions of the aesthetics of the application and their perceptions of the usability of the application. These results suggest that enhancing the aesthetic appeal of the LoCuS user interface would increases the student's perception of the usability of LoCuS.

In the second source Johnson analyzed the psychological reasoning involved when a user interacts with a user interface [5]. The book explains how to plan a design around the user's visual and mental processes. A wide range of topics are covered including human perception, Gestalt principles, visual structures, color blindness, and vision, among others. The purpose of the book is not just to give designers tricks to apply to an interface, but also to give designers a deeper understanding of the user's mind so that they can guide the user through the interface in a meaningful way. With this knowledge we identified LoCuS components and component attributes that would benefit from user interface enhancements. By enhancing the components and attributes using the methods describe in the book, it is easier for the student to mentally and visually process the content of LoCuS which increases usability.

In the third source Mayer conducted a study done on the effects of teaching students through the use of various forms of multimedia [6]. The study covers multiple forms of multimedia learning techniques including the use of video, animation, images, and narration. When conducting the study students were given several lessons, each taught using multiple forms of multimedia learning, and then tested to determine how effective the learning method was. The results of the study listed the different methods of multimedia learning and how an instructor or teacher can use multimedia elements to better design lessons and increase learning. This study confirmed that teaching student though the use of a notebook with embedded images, as is done in LoCuS, was a valuable form of multimedia learning. The study also found positive result when teaching with a document that uses video, narration, and animation along with images. Based on these result we decided to suggest the addition of more multimedia components in LoCuS. Adding these extra capabilities to LoCuS makes learning the lab material simpler for the student making the LoCuS environment more usable.

3.2 Patterns and Application

This section discusses literature related to user interface design patterns and the application of specific research to the LoCuS environment. Many of the enhancements we recommend use design patterns from the literature to implement the results from the research in Section 3.1. For others, such as the virtual document, we use the specific research results to build our suggestions.

The design patterns came from Tidwell's book that guides readers through the process of designing user interfaces using interface design patterns [8]. The book explains general interface guidelines and techniques, including how to recognize interface elements that need to be redesigned. The design patterns are well defined with several examples of how to use them correctly and explanations of why they should be used. When we identified a potential component or attribute in LoCuS that could be enhanced we searched through Tidwell's design patterns to see if there were any similarities. If a similarity was found we would base our suggestion on the recommendations from Tidwell's book.

When we were making suggestion on how to enhance the notebook we referred to research done on the value of creating realistic virtual books. The authors showed that emulating the physical properties of a book or document increased the user's enjoyment [3]. In the study they converted digital books, that would typically be paginated, into a virtual book with the appearance of the physical book it represents. Some physical aspects that were chosen for emulation include page turning, book thickness, page age, and so on. By using these physical aspects the virtual book was able to relate important information to the user in a enjoyable manner. Through this study the authors came to the conclusion that emulating a physical book enhances the enjoyment for the user and increases the user's perceived value of the book. With this research we decided to suggest several changes to the LoCuS notebook. One major difference between the research and our suggestions is that the research used a three-dimensional book, whereas we are using a two-dimensional notebook. We propose that the difference is negligible since the primary reading view of the book and notebook will be from the same angle.

When considering how to indicate user progress we relied on the work of Harrison et al. who conducted a study to measure the usefulness of different types of progress bars when used with non-linear content [4]. In the study the authors found that the traditional progress bar was ineffective when used with content that can vary in difficulty or length between sections. To find these results they tested various different ways of presenting how much of the content the user had completed using a progress bar and then they gaged the user response. The results of the study suggested creating a more informative progress bar that does not describe the exact progress completed, but instead indicated progress in a non-linear fashion. From this research we decided to suggest other means of indicate progress in LoCuS. Our suggestions follow the researcher's idea that progress should not be indicated in a linear fashion, however the suggestions do not use a progress bar.

Finally we used Altaboli and Lin's study on the effects of screen layout to rearrange the notebook and chalkboard components of LoCuS [2]. The study was conducted on participants to tests what effects changing the aspects of screen elements had on the participants. The aspects that were manipulated in the study were balance, unity, and sequence. The combination of these three aspects account for the size, location, and similarity of elements on a screen. By changing the ratio of these aspects they were able to show a measurable difference in the aesthetic appeal of the application on the participants. In the study they found that by attempting to maximize these three aspects it is possible to increase the aesthetic appeal of an application. We suggested that some of the components in LoCuS be resized and moved to conform with the results of this research.

Chapter 4

Suggested Enhancements

We now list possible changes that could be made within the LoCuS environment to improve the usability and student enjoyment of the project. Each suggestion is contained in its own section and is applied individually to LoCuS as it is described in Section 2.1, unless noted otherwise. These suggestions are presented as an idea followed by the recommended application of the idea. The idea portion consists of a general concept derived from prior research on usability or relevant topics. The application portion presents a description of how the research could be applied within the LoCuS environment.

4.1 Screen Layout

Idea The idea to change the screen layout came from Altaboli and Lin's research on the effect of screen layout on screen elements [2]. In their research, Altaboli and Lin measured the appeal of eight different screen layout models as seen in Figure 4.1. These models were created by changing the size, location, and similarities of four screen elements. Each of these models was tested by participants for visual aesthetics. The most appealing model was model one and the least appealing model was model eight. As you can see, model one has all four screen elements the same size, and they are grouped so that there is a visual balance between all four elements. Model eight's elements are all random sizes and in random locations. This makes model eight unbalanced and visually less appealing. To apply this research to a software application we need to try and visually balance the elements. They do not need to be perfectly balanced as they are in model one. Any increase in the balance and similarities of the elements will increase the overall aesthetic appeal.

Application To apply Altaboli and Lin's research to the components of LoCuS we needed to identify the components that are stationary throughout a laboratory experiment. The notebook and the chalkboard remain in the same locations for the duration of a lab. These components are seen in Figure 4.2. The notebook was the first component we considered enhancing. Since the notebook is at the edge of both the top and the left of the screen we decided that it could be move closer to the center to make it a floating component similar to the chalkboard. The chalkboard is already a floating component so we decided that its width could be increased significantly and its height could be increased slightly. The chalkboard should also be aligned with the notebook so that the components are better balanced. Doing this would increase the aesthetic of LoCuS and, in the case of the chalkboard, make it more readable. We have been careful not to resize nor move the components too much or they could interfere with other components of LoCuS. LoCuS already imposes a minimum screen resolution of 1280 by 800 to help insure that there is enough screen space. The large empty space on the LoCuS desktop is required so that the students still have room for apparatuses. When an apparatus launches it should fit in this space so that the chalkboard and notebook are readable. Our suggestion takes this space into consideration and are seen in Figure 4.3. If there is an extra large apparatus or multiple apparatuses on the screen, which do not allow for a full view of the chalkboard or notebook, then the chalkboard and notebook are overlapped by the apparatus(es). In order for the students to view the chalkboard or notebook, they have to move the apparatuses.

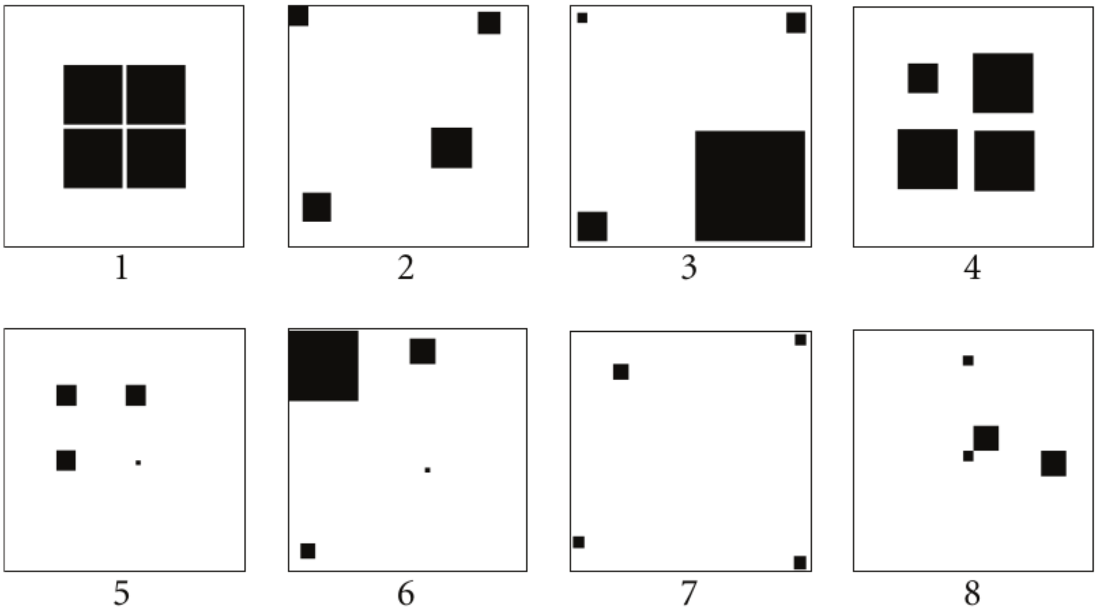
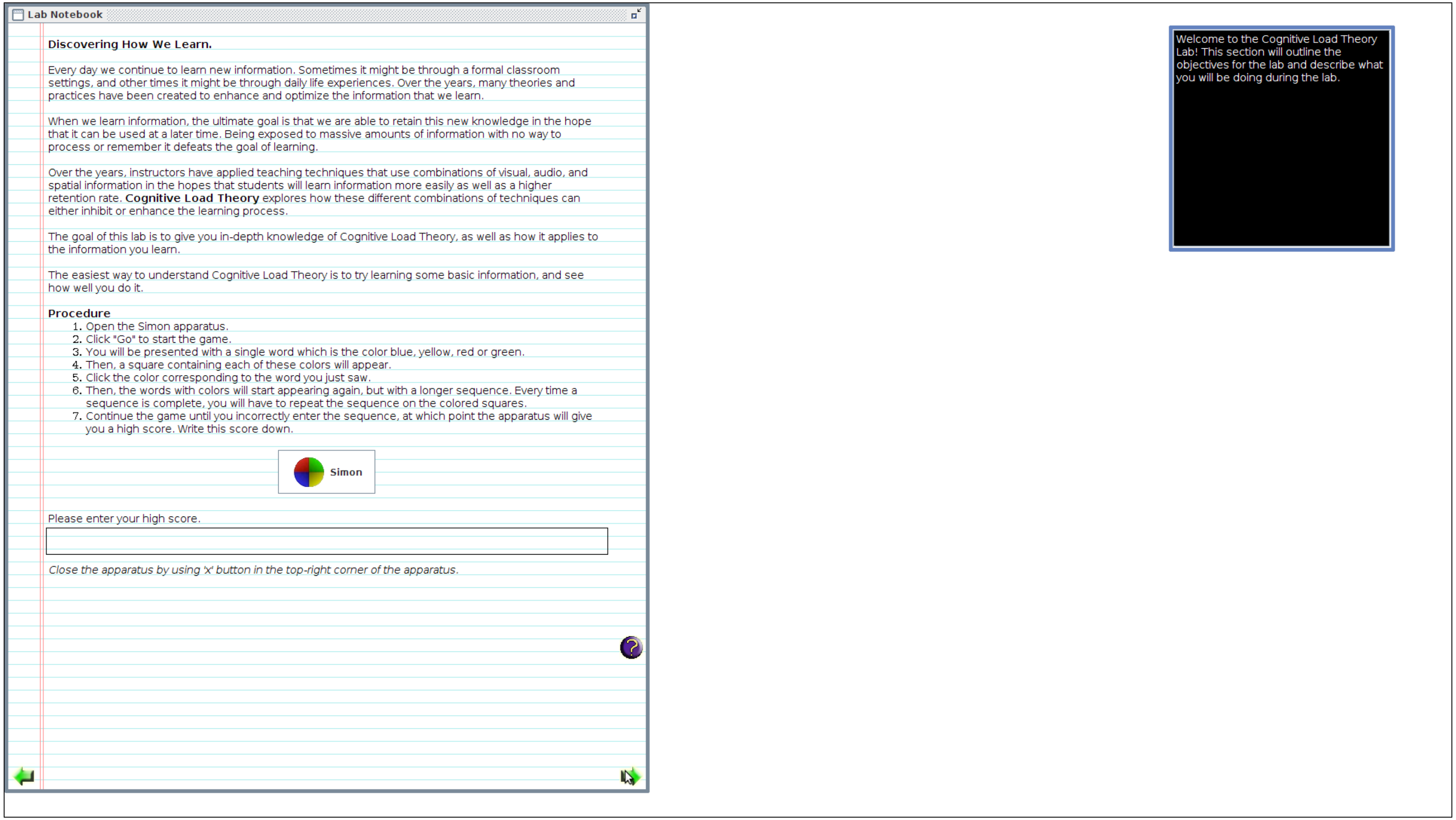


Figure 4.1: The eight models from Altaboli and Lin's research study on the layout of screen elements. Model one has the highest aesthetic appeal and model eight has the lowest. [2, Fig. 1]



Discovering How We Learn.

Every day we continue to learn new information. Sometimes it might be through a formal classroom settings, and other times it might be through daily life experiences. Over the years, many theories and practices have been created to enhance and optimize the information that we learn.

When we learn information, the ultimate goal is that we are able to retain this new knowledge in the hope that it can be used at a later time. Being exposed to massive amounts of information with no way to process or remember it defeats the goal of learning.

Over the years, instructors have applied teaching techniques that use combinations of visual, audio, and spatial information in the hopes that students will learn information more easily as well as a higher retention rate. **Cognitive Load Theory** explores how these different combinations of techniques can either inhibit or enhance the learning process.

The goal of this lab is to give you in-depth knowledge of Cognitive Load Theory, as well as how it applies to the information you learn.

The easiest way to understand Cognitive Load Theory is to try learning some basic information, and see how well you do it.

Procedure

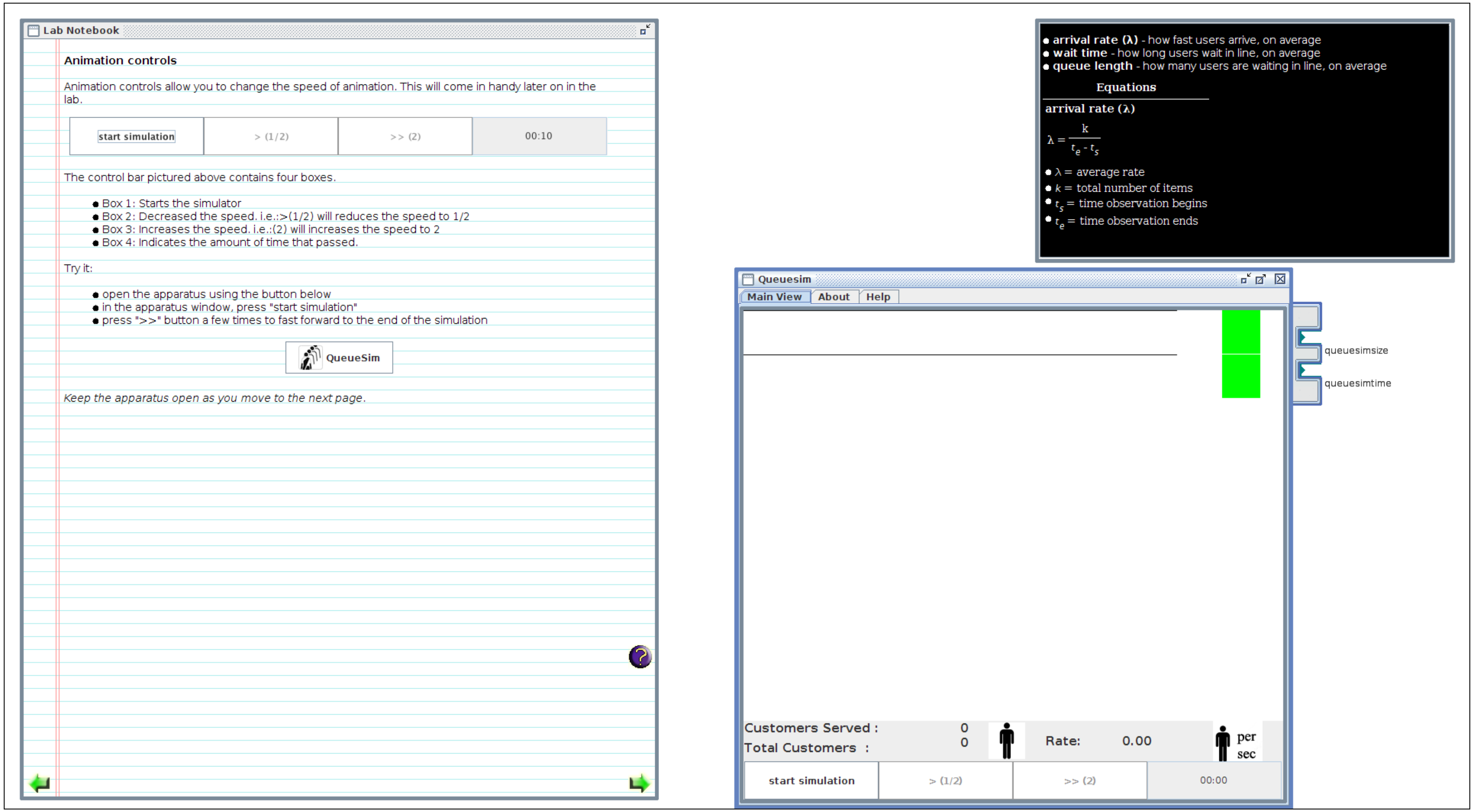
1. Open the Simon apparatus.
2. Click "Go" to start the game.
3. You will be presented with a single word which is the color blue, yellow, red or green.
4. Then, a square containing each of these colors will appear.
5. Click the color corresponding to the word you just saw.
6. Then, the words with colors will start appearing again, but with a longer sequence. Every time a sequence is complete, you will have to repeat the sequence on the colored squares.
7. Continue the game until you incorrectly enter the sequence, at which point the apparatus will give you a high score. Write this score down.

Please enter your high score.

Close the apparatus by using 'x' button in the top-right corner of the apparatus.

Welcome to the Cognitive Load Theory Lab! This section will outline the objectives for the lab and describe what you will be doing during the lab.

Figure 4.2: The current size and location of the notebook and chalkboard in LoCuS.



Animation controls

Animation controls allow you to change the speed of animation. This will come in handy later on in the lab.

The control bar pictured above contains four boxes.

- Box 1: Starts the simulator
- Box 2: Decreased the speed. i.e.:>(1/2) will reduces the speed to 1/2
- Box 3: Increases the speed. i.e.:(2) will increases the speed to 2
- Box 4: Indicates the amount of time that passed.

Try it:

- open the apparatus using the button below
- in the apparatus window, press "start simulation"
- press ">>" button a few times to fast forward to the end of the simulation

Keep the apparatus open as you move to the next page.

- **arrival rate (λ)** - how fast users arrive, on average
- **wait time** - how long users wait in line, on average
- **queue length** - how many users are waiting in line, on average

Equations

arrival rate (λ)

$$\lambda = \frac{k}{t_e - t_s}$$

- λ = average rate
- k = total number of items
- t_s = time observation begins
- t_e = time observation ends

Customers Served : 0
Total Customers : 0
Rate: 0.00 per sec

Figure 4.3: The suggested locations of the notebook and chalkboard. The chalkboard has been increased in size, however it is not so large as to be hidden by a large apparatus as demonstrated in the figure. Both the notebook and chalkboard are equal distances from the top and sides of the main framing window that encapsulates them.

4.2 Deep Background

Idea "Deep Backgound" is one of Tidwell's user interface design patterns [8, pp. 499–503]. This design pattern separates the foreground elements from the background elements to direct the user's focus to the foreground elements. The idea is to use a background that contrasts with the foreground and is slightly out of focus so that it creates depth between the foreground elements and the background elements. By using a background that creates depth between itself and foreground elements a pseudo-3D effect is produced that draws the user's focus to the foreground. Deep Background also conforms to the Gestalt principle of Figure and Ground, which explains how human visual perception separates foreground and background images based on visual context [5, pp. 11–24]. The Gestalt principles are a set of observation about how human visual perception registers information. The Gestalt observations found that the human visual system enforces shape recognition, seeing objects as whole instead of separated part, and grouping objects based on similarity and location. In the case of Figure and Ground the visual system typically chooses the smaller object as the foreground and any encapsulating object as the background. When considering the Deep Background design pattern it is clear that it reinforces the Figure and Ground principle by drawing all of the desired elements to the foreground. All that we need to apply the Deep Background design pattern is a background that contrasts with the foreground elements and separates them from the background.

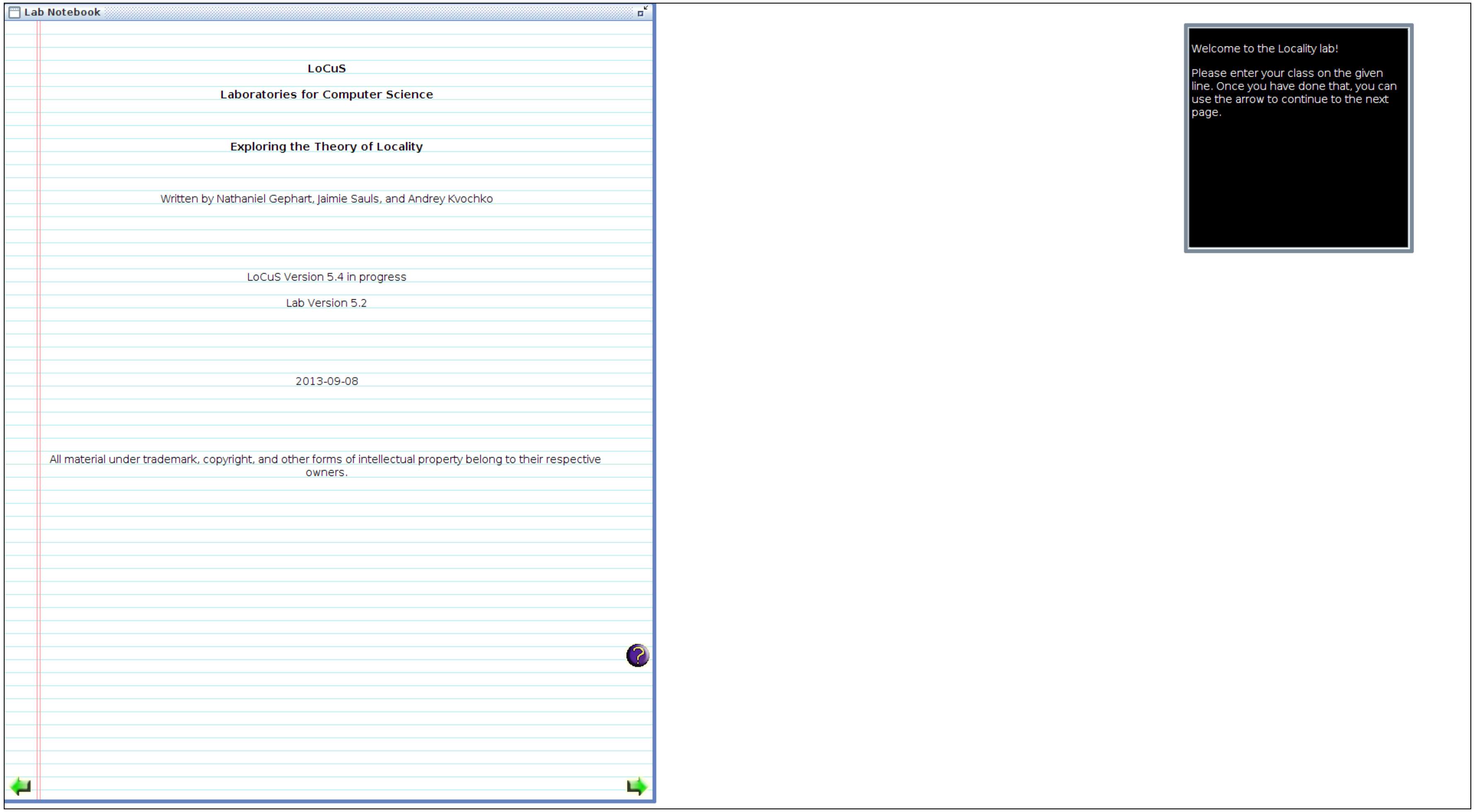
Application To apply the Deep Background design pattern to LoCuS we need to change the main frame's background component to one that contrasts with the foreground and slightly blurs the image. This will draw the student's focus to the notebook and chalkboard as the primary components.

There are at least two alternates options available that we suggest to achieve the desired results. Figure 4.4 is a base representation of the LoCuS environment which includes the open notebook and the chalkboard. In the figure the notebook is open since it is open for the duration of the lab except when the student initially starts a LoCuS lab. Figure 4.5 contains the same components with an alternate themed background to match the school environment. Figure 4.6 contains the same images with an alternate background that better conforms to the Deep Background design pattern.

Alternate 1: In this alternate, we suggest that the background image should be a wooden desktop similar to the desktops in a classroom or laboratory. An example is shown in Figure 4.5. By using a wooden desktop as a background we would conform to the Deep Background design pattern and theming, covered in Section in 4.6, which would provide the student with a visually consistent user experience.

Alternate 2: In this alternate, we suggest that a background is chosen that better conforms to the Deep Background design pattern. To achieve this a nondescript image would be chosen that contrasts with both the notebook and the chalkboard. An example is shown in Figure 4.6. Although this alternate would fit the design pattern better it does not conform to the laboratory theme in LoCuS and as such might have the effect of drawing the student out of the laboratory environment.

Supplemental Option: Include an instructor option that allows the instructor to choose a background as part of administering the lab. If this option is included then it would be important to make the instructor aware of the Deep Background design pattern so that they chosen background would not be counter productive. This option could also include a means of blurring the background image to better conform to the Deep Background principle.



LoCuS

Laboratories for Computer Science

Exploring the Theory of Locality

Written by Nathaniel Gephart, Jaimie Sauls, and Andrey Kvochko

LoCuS Version 5.4 in progress

Lab Version 5.2

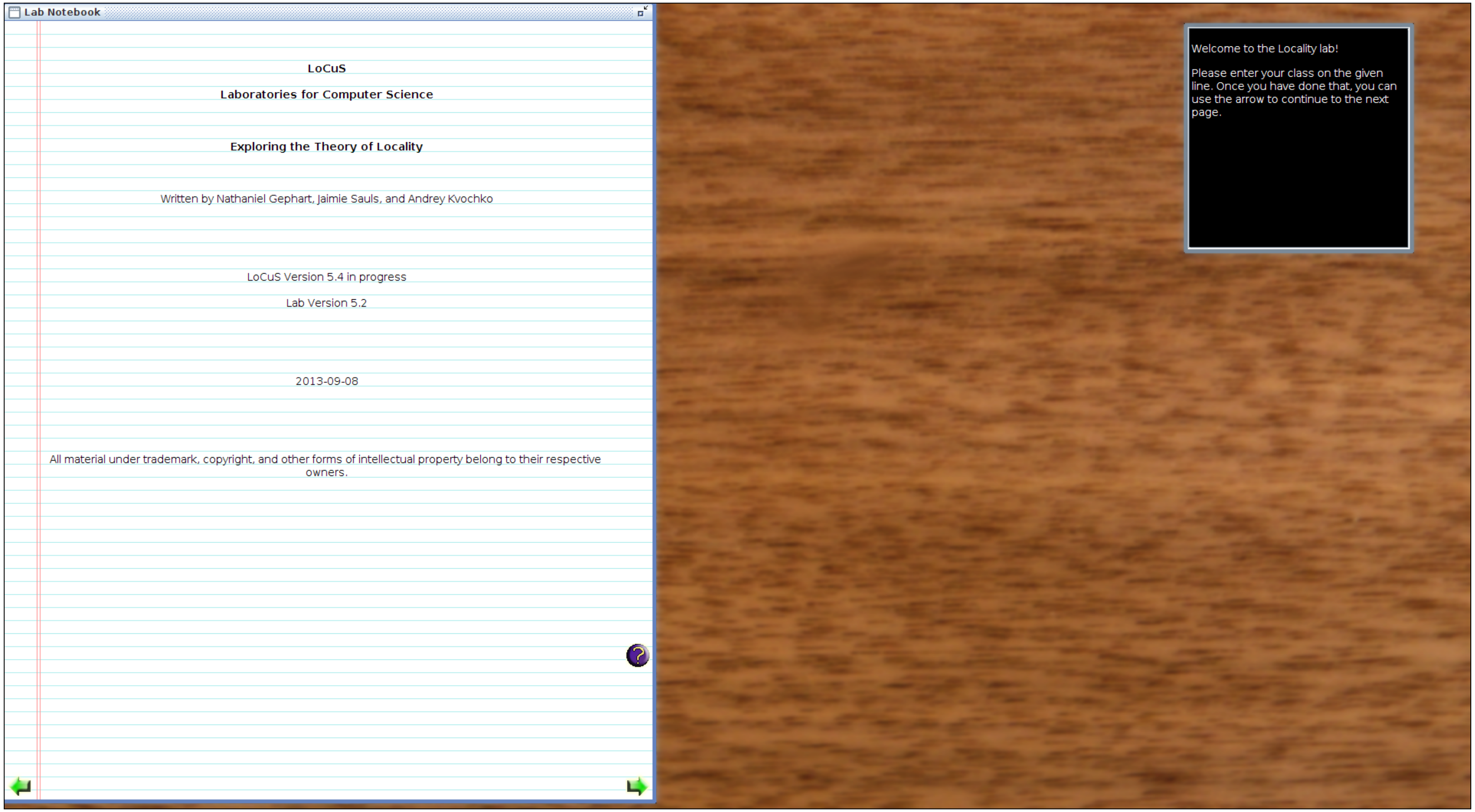
2013-09-08

All material under trademark, copyright, and other forms of intellectual property belong to their respective owners.

Welcome to the Locality lab!

Please enter your class on the given line. Once you have done that, you can use the arrow to continue to the next page.

Figure 4.4: LoCuS with the current background, the open notebook, and the chalkboard displayed.



LoCuS

Laboratories for Computer Science

Exploring the Theory of Locality

Written by Nathaniel Gephart, Jaimie Sauls, and Andrey Kvochko

LoCuS Version 5.4 in progress

Lab Version 5.2

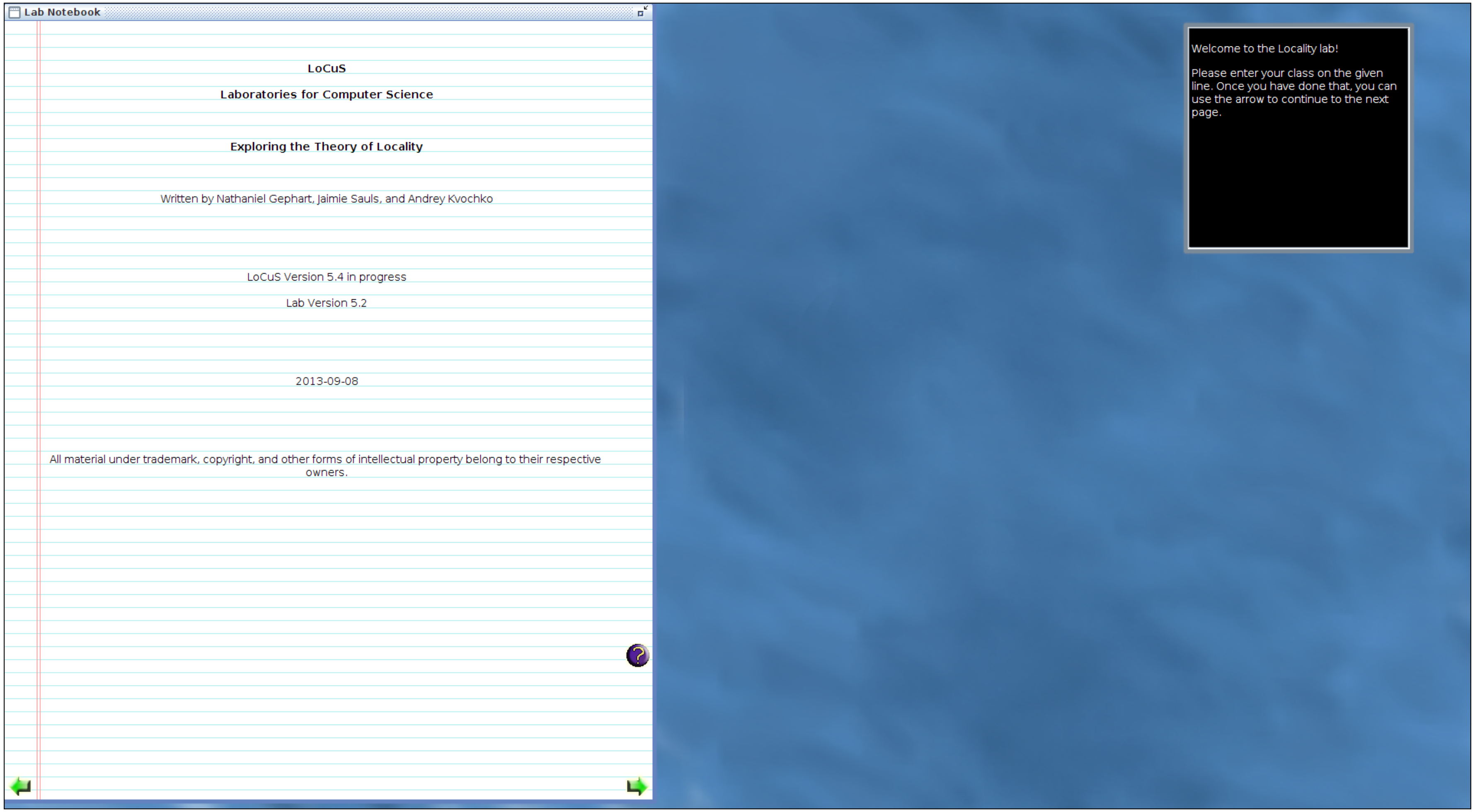
2013-09-08

All material under trademark, copyright, and other forms of intellectual property belong to their respective owners.

Welcome to the Locality lab!

Please enter your class on the given line. Once you have done that, you can use the arrow to continue to the next page.

Figure 4.5: LoCuS with a slightly blurred wooden desktop as the background, the open notebook, and the chalkboard.



LoCuS

Laboratories for Computer Science

Exploring the Theory of Locality

Written by Nathaniel Gephart, Jaimie Sauls, and Andrey Kvochko

LoCuS Version 5.4 in progress

Lab Version 5.2

2013-09-08

All material under trademark, copyright, and other forms of intellectual property belong to their respective owners.

Welcome to the Locality lab!

Please enter your class on the given line. Once you have done that, you can use the arrow to continue to the next page.

Figure 4.6: LoCuS with a nondescript background, the open notebook, and the chalkboard.

4.3 Virtual Document

Idea Emulate the physical properties of a document when using a virtual representation of it to increase the user's enjoyment and make the environment containing the virtual document more immersive [3]. By simulating features of a real document we can present the user with an experience that they are familiar with and understand how to use. This make the users more comfortable with the environment they are using and intern draws them into the virtual environment in which the document exist.

Creating virtual documents also conforms to the Gestalt principles of Continuity and Closure as described by Johnson [5, pp. 11–24]. The principle of Continuity states that people attempt to see forms, such as line segment, as connected rather than disconnected. Continuity explains why parts of a virtual document are perceived as a whole book rather than disconnected segments. The principle of Closure is similar to that of Continuity in that the visual system attempts to complete an object; however in the case of Closure it is attempting to close open spaces instead of connecting closed spaces. When viewing a virtual document, a user perceives its depth and spacial features by using Closure to close the empty spaces left by objects that are not present on the screen, but still perceived. An example of this would be a virtual books cover. Only the edge of the cover is printed on the screen, but the viewer would visually register the entire cover. With an understanding of the Gestalt principles of Continuity and Closure we can see that a virtual document is designed to work with the human visual perception system.

The combination of Chu et al. research along with the Gestalt principles gives a strong foundation to conclude that changing a paginated document into a virtual document would increase the usability of the document and the environment that contains it.

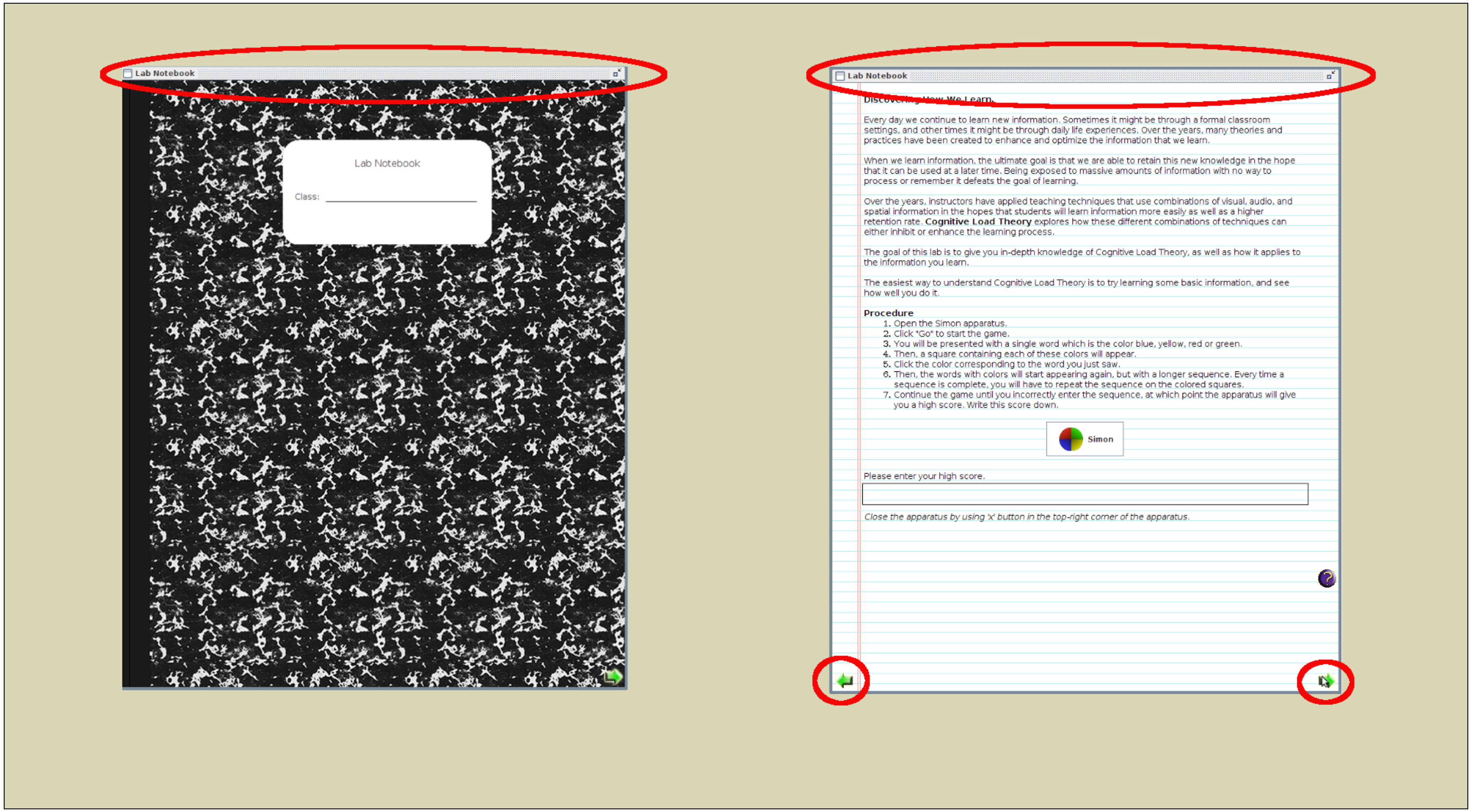
Application We suggest that the notebook in LoCuS be converted to resemble a physical spineless notebook in order to enhance visual aesthetics, conform to Chu et al. research, and make better use of the Gestalt principles.

In the current implementation the notebook is framed by the Java default windowing decorations. For the student to progress through the document, the pages are traversed using pagination along with forward and back buttons as shown in Figure 4.7. The navigation buttons can also be replaced with a more modern and easier to use form of navigation by using the entire side of the page instead of the forward and back buttons. To facilitate the recommended enhancements it is necessary to remove the Java window decorations and shift the notebook away from the extreme left edge of the LoCuS desktop.

Page Texturing: As part of creating a virtual document it is important to make sure the texture of the pages emulate that of the document that is being copied. As part of page texturing we suggest that the notebook paper be given a flowing three-dimensional look as seen in Figures 4.8, 4.9, and 4.10. Another part of page texturing is adding page lines and markers. This has already been implemented in the current version of LoCuS and can be seen in any of the figures containing the notebook.

Page Turning: We suggest that the notebook should emulate a page turning when the student progresses to the next page. The page turning should be visually similar to that of a spineless notebook when a page is turned. There are two alternate ways of handling the open spineless notebook. The pages can be open and flat, or the pages can be wrapped around the back of the notebook. Both of these methods are explained in alternate one and two.

- *Alternate 1:* Have the page flip to the side as if the notebook were open flat in a two page layout. It would require the backside of the previous page to fall primarily off screen in order to keep the present lab format as in Figure 4.8.
- *Alternate 2:* Have the page flip behind the current page(s). Using this format would not require any of the notebook to be off screen. The left side of the open notebook would be represented by a



Discovering How We Learn.

Every day we continue to learn new information. Sometimes it might be through a formal classroom settings, and other times it might be through daily life experiences. Over the years, many theories and practices have been created to enhance and optimize the information that we learn.

When we learn information, the ultimate goal is that we are able to retain this new knowledge in the hope that it can be used at a later time. Being exposed to massive amounts of information with no way to process or remember it defeats the goal of learning.

Over the years, instructors have applied teaching techniques that use combinations of visual, audio, and spatial information in the hopes that students will learn information more easily as well as a higher retention rate. **Cognitive Load Theory** explores how these different combinations of techniques can either inhibit or enhance the learning process.

The goal of this lab is to give you in-depth knowledge of Cognitive Load Theory, as well as how it applies to the information you learn.

The easiest way to understand Cognitive Load Theory is to try learning some basic information, and see how well you do it.

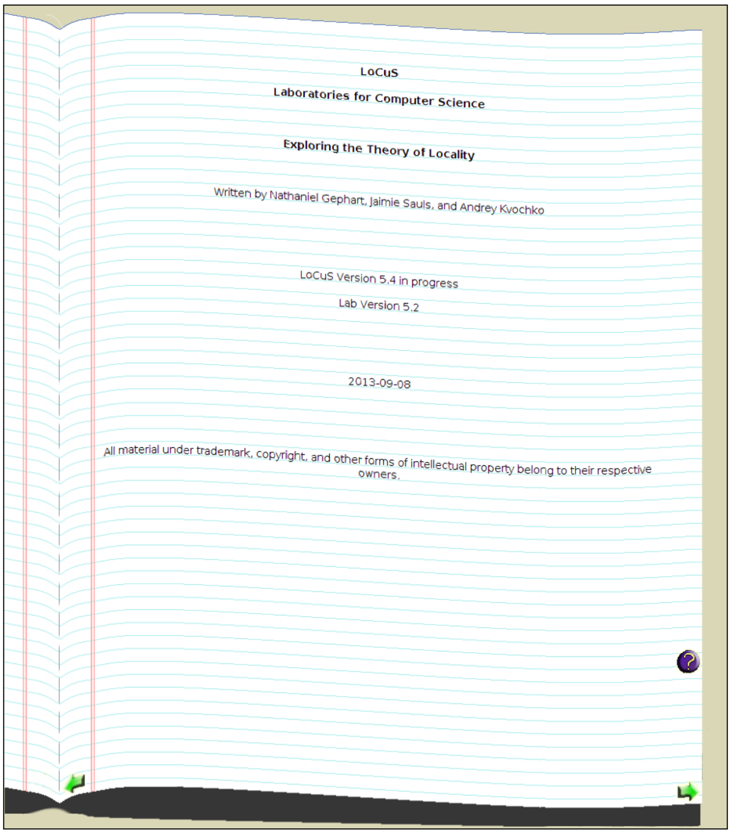
Procedure

1. Open the Simon apparatus.
2. Click "Go" to start the game.
3. You will be presented with a single word which is the color blue, yellow, red or green.
4. Then, a square containing each of these colors will appear.
5. Click the color corresponding to the word you just saw.
6. Then, the words with colors will start appearing again, but with a longer sequence. Every time a sequence is complete, you will have to repeat the sequence on the colored squares.
7. Continue the game until you incorrectly enter the sequence, at which point the apparatus will give you a high score. Write this score down.

Please enter your high score.

Close the apparatus by using 'x' button in the top-right corner of the apparatus.

Figure 4.7: The current LoCuS notebook with Java decorations and arrows to navigate pages. The Java title bar decoration and the navigation arrows have been circled in red. The figure represents both the closed and open notebook.



LoCuS

Laboratories for Computer Science

Exploring the Theory of Locality

Written by Nathaniel Gephart, Jaimie Sauls, and Andrey Kvochko

LoCuS Version 5.4 in progress

Lab Version 5.2

2013-09-08

All material under trademark, copyright, and other forms of intellectual property belong to their respective owners.

Figure 4.8: The notebook with previous pages open and off of the screen to the left. Also note the flowing three-dimensional page as apposed to the flat two-dimensional notebook in Figure 4.7

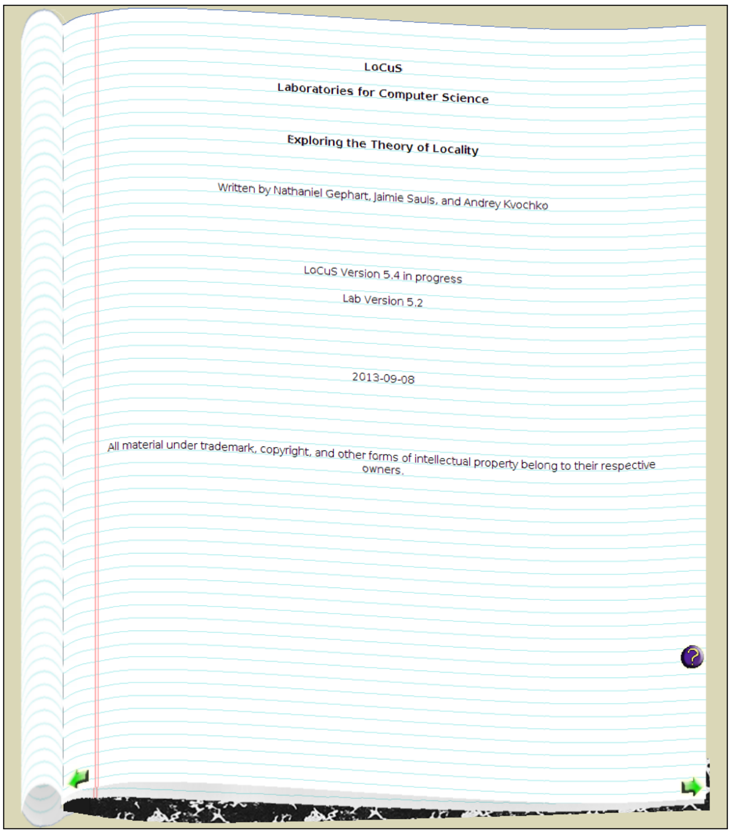
layered fold of the ’spine’ to mimic the visual effect of the pages wrapping around the backside of the notebook. Figure 4.9 is a mock image of how this would appear.

Page Traversal: We suggest that the way a student navigates to the next page should be change to increase the usability of the page forward and page backward navigation buttons. Whenever the mouse is anywhere except the page forward or back area there is only a hollow green arrow. When the student moves the mouse over the page forward or back area it will be darkened to indicate that it is the clickable area to change page. This is illustrated in Figure 4.10. Although it is not shown, backwards traversal would be done in the same manner using the overhanging or folded over page section.

Notebook Depth: We suggest that the amount of remaining pages in the notebook be indicated by displaying the notebook depth. Accomplishing this can be done by adding a dynamically changing border to the right side of the notebook to create the appearance of a stack of pages that changes height during progress forward and backward through the lab. Figure 4.11 show three different stage of progress in the notebook.

Text Layout: We suggest that the text on the notebook pages should be aligned to match the line spacing provided on the notebook pages. This is currently implemented in the notebook on regular text, but does not apply to subscript or components such as radio buttons.

Font: The font currently used in the notebook is Vera. Vera has the appearance of a typed digital font and as such distances the user from the notebook theme. To remedy this, we suggest that the font be replaced by one or more open source fonts that mimic handwriting. The difference between the current font and a possible suggested fonts is shown in Figure 4.12.



LoCuS

Laboratories for Computer Science

Exploring the Theory of Locality

Written by Nathaniel Gephart, Jaimie Sauls, and Andrey Kvochko

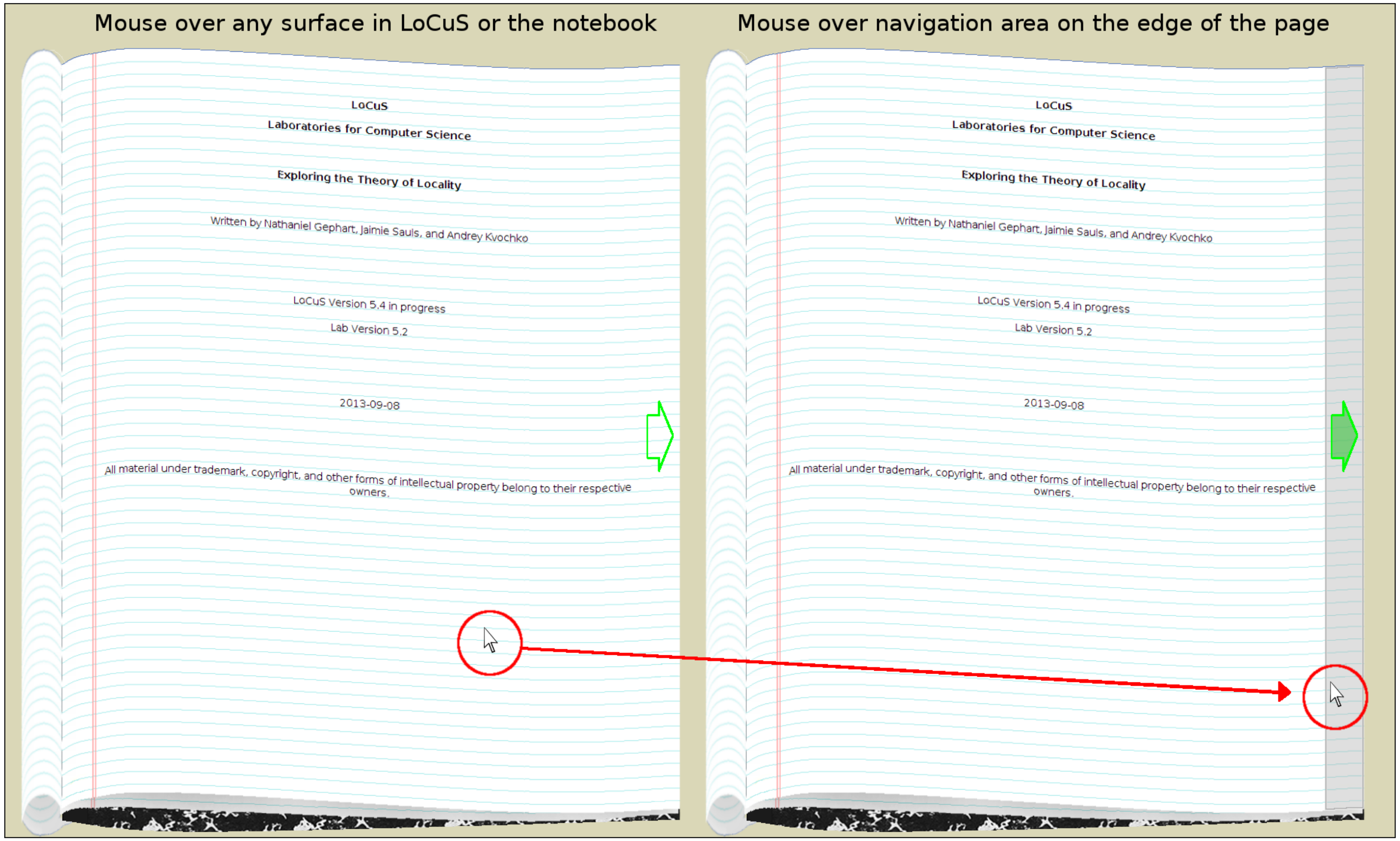
LoCuS Version 5.4 in progress

Lab Version 5.2

2013-09-08

All material under trademark, copyright, and other forms of intellectual property belong to their respective owners.

Figure 4.9: The notebook with previous pages flipped behind the current pages.


Mouse over any surface in LoCuS or the notebook

LoCuS

Laboratories for Computer Science

Exploring the Theory of Locality

Written by Nathaniel Gephart, Jaimie Sauls, and Andrey Kvochko

LoCuS Version 5.4 in progress

Lab Version 5.2

2013-09-08

All material under trademark, copyright, and other forms of intellectual property belong to their respective owners.

Mouse over navigation area on the edge of the page

LoCuS

Laboratories for Computer Science

Exploring the Theory of Locality

Written by Nathaniel Gephart, Jaimie Sauls, and Andrey Kvochko

LoCuS Version 5.4 in progress

Lab Version 5.2

2013-09-08

All material under trademark, copyright, and other forms of intellectual property belong to their respective owners.

Figure 4.10: Traversing the notebook by using the side of the notebook as a forward button instead of just a small arrow. This figure illustrates what would happen when the user move the mouse from the center of the page to the side of a page. When the arrow is over the side, the side will darken and the forward arrow will be highlighted. The entire side would then become a clickable surface.

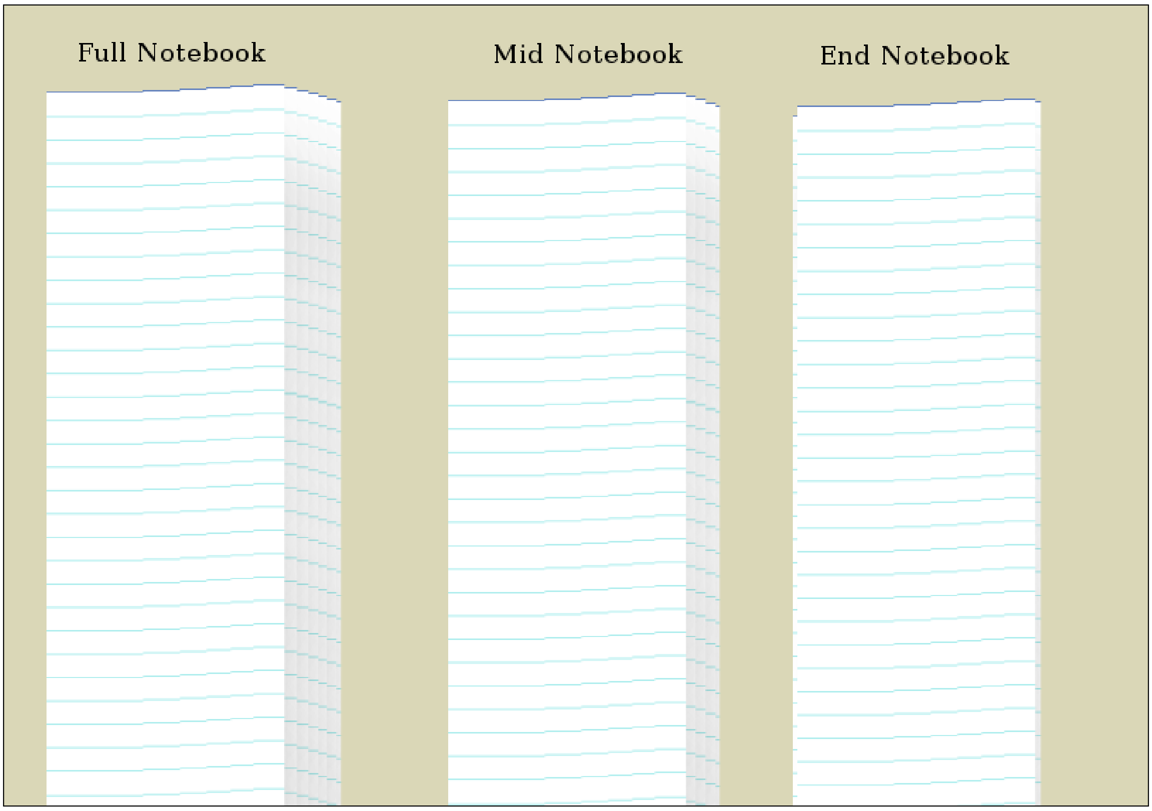
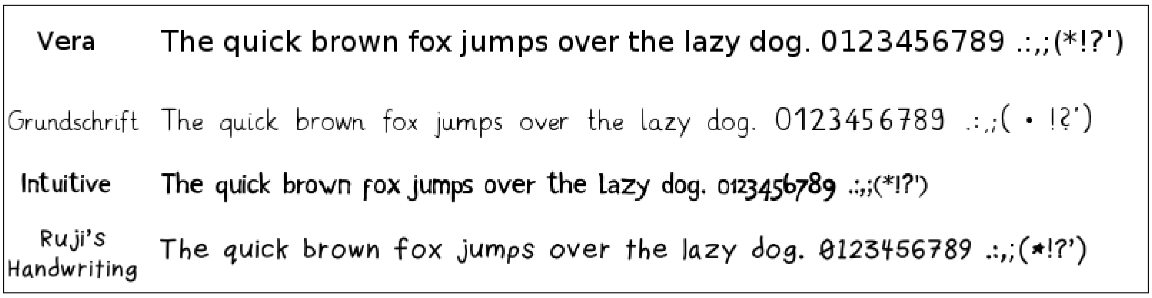


Figure 4.11: The number of pages remaining in the notebook change as the student traverses to the next page. This would give the student an indication that they are progressing towards completion of the lab.



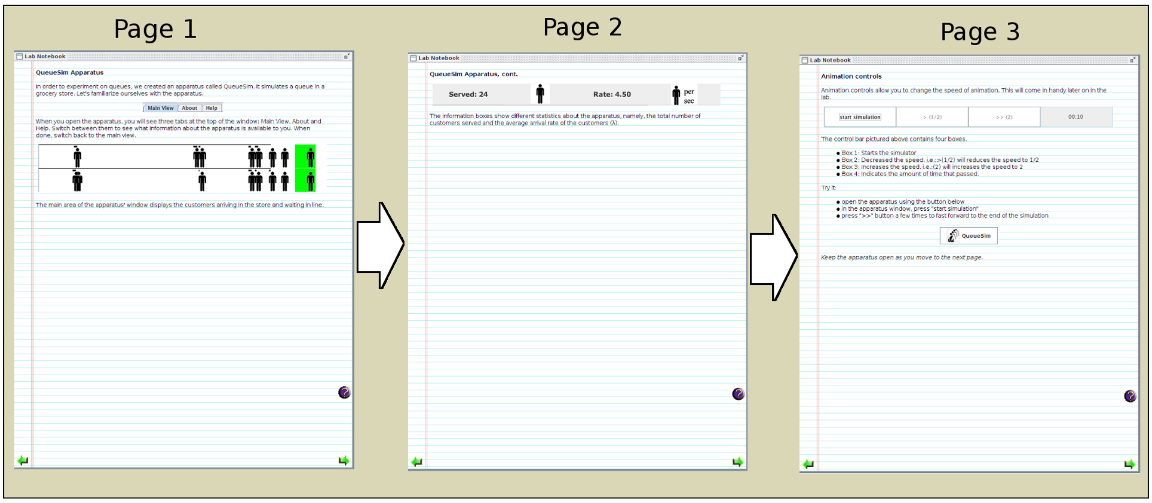
Vera	The quick brown fox jumps over the lazy dog. 0123456789 .:,;(*!?')
Grundschrift	The quick brown fox jumps over the lazy dog. 0123456789 .:,;(• !?')
Intuitive	The quick brown fox jumps over the lazy dog. 0123456789 .:,;(*!?')
Ruji's Handwriting	The quick brown fox jumps over the lazy dog. 0123456789 .:,;(*!?')

Figure 4.12: Here is a selection of fonts that could be use in the notebook. The first font, Vera, is the current font in the notebook and all of LoCuS. The rest are possible replacements that mimic handwriting.

4.4 Guided Walkthrough

Idea The idea for a guided walkthrough comes from one of Tidwell's design patterns called a Wizard [8, pp. 54–58]. The Wizard design pattern is used to guide a user through complex or confusing tasks in the correct order. It does this by breaking up the task into reasonable steps and ordering them for ease of understanding to present them in an intuitive manner. To apply this pattern we need to break up the chosen task into small steps and present the task in a way that guides the user through the task.

Application We suggest that guided walkthroughs be added to apparatuses to guide the student through the use of the apparatus. One of the main purposes of this suggestion is to remove the instruction from the notebook and create separate functionality for learning how to use the apparatuses. Currently the apparatuses are explained in the notebook which can take up a significant portion of a lab. As an example, the explanation of how to use the QueueSim apparatus in the Little's Law Lab takes three pages as shown in Figure 4.13. Even though three pages may seem trivial, the Little's Law lab, excluding front matter, has ten pages of apparatus instruction out of thirty-six pages in total. This account for 28% of the lab. A few of those pages include some course material, but the main focus of the pages are apparatus instruction. Finding a way to combine these instructions with the apparatus would go a long way towards making the LoCuS environment more usable and remove some of the burden from the lab authors.



QueueSim Apparatus

In order to experiment on queues, we created an apparatus called QueueSim. It simulates a queue in a grocery store. Let's familiarize ourselves with the apparatus.

When you open the apparatus, you will see three tabs at the top of the window: Main View, About and Help. Switch between them to see what information about the apparatus is available to you. When done, switch back to the main view.

The main area of the apparatus' window displays the customers arriving in the store and waiting in line.

QueueSim Apparatus, cont.

The information boxes show different statistics about the apparatus, namely, the total number of customers served and the average arrival rate of the customers (λ).

Animation controls

Animation controls allow you to change the speed of animation. This will come in handy later on in the lab.

The control bar pictured above contains four boxes.

- Box 1: Starts the simulator
- Box 2: Decreased the speed. i.e.:>(1/2) will reduces the speed to 1/2
- Box 3: Increases the speed. i.e.:(2) will increases the speed to 2
- Box 4: Indicates the amount of time that passed.

Try it:

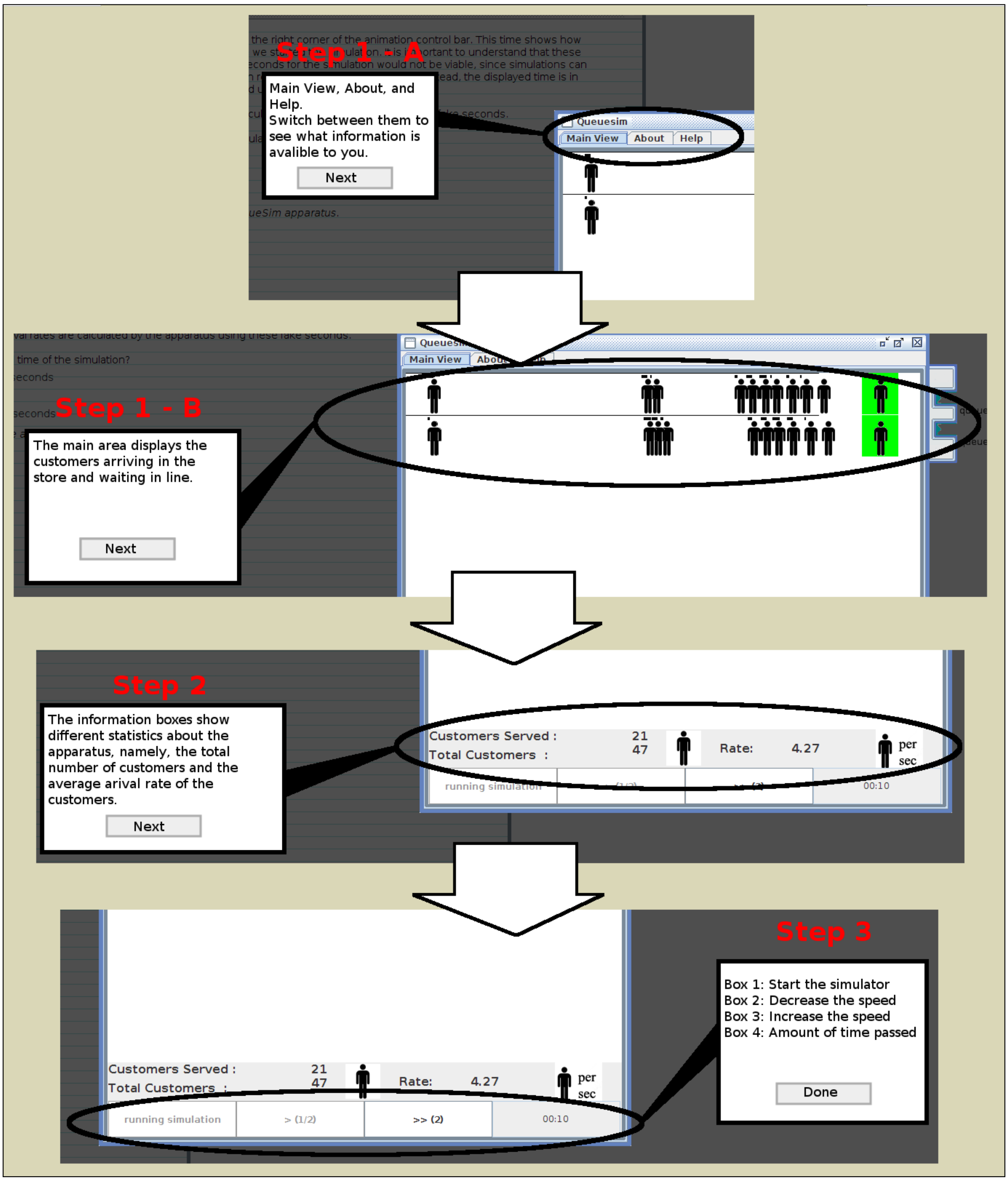
- open the apparatus using the button below
- in the apparatus window, press "start simulation"
- press ">>" button a few times to fast forward to the end of the simulation

Keep the apparatus open as you move to the next page.

Figure 4.13: Current method of explaining QueueSim apparatus. There are three pages explaining the functionality of the QueueSim apparatus.

We suggest that the explanation of an apparatus should be presented using the Wizard design pattern to guide the user though the apparatus step by step. This could be done with informational pop-ups explaining the purpose of the elements on the screen. Everything behind the apparatus could also be darkened to highlight the apparatus and popup instructions. Figure 4.14 shows a possible walkthrough for the QueueSim apparatus.

By demonstrating the functionality of an apparatus in this way it requires greater student engagement and allow the lab designer to direct more focus on the lesson by removing the requirement to teach the tools as well. An additional benefit of this approach would be adding a level of separation between the tools and the theory being presented to the student. Another advantage of the walkthrough technique would be giving the student the ability to choose whether the walkthrough is needed. If the student is already familiar with the apparatus they could have the option to skip the demonstration.



Main View, About, and Help.
Switch between them to see what information is avalible to you.

The main area displays the customers arriving in the store and waiting in line.

The information boxes show different statistics about the apparatus, namely, the total number of customers and the average arival rate of the customers.

Customers Served : 21
Total Customers : 47
Rate: 4.27 per sec

Box 1: Start the simulator
Box 2: Decrease the speed
Box 3: Increase the speed
Box 4: Amount of time passed

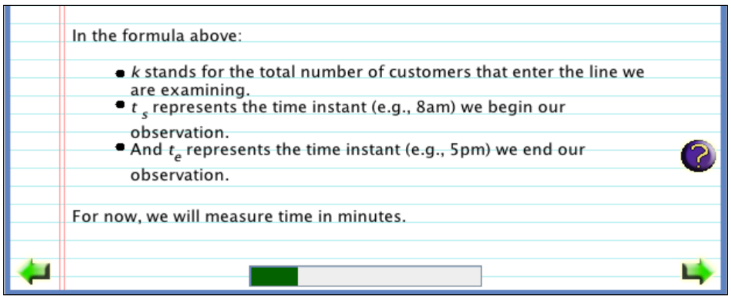
Figure 4.14: Explaining the functionality of QueueSim by using a guided walkthrough. The steps would be displayed in floating popups and everything behind the apparatus would be darkened. The arrows indicate the order in which the steps are executed and each step would have a next button to progress to the next step.

4.5 User Progress

Idea The idea is to use Tidwell's Sequence Map design pattern to indicate the progress of the current user and remove any vague information that leaves the user with little knowledge of the their progress [8, pp. 118–120].

The Sequence Map design pattern aligns with the suggestions by Harrison et al. in that progress for non-linear task should be indicated in a non-linear way [4]. Where the Sequence Map differs is instead of changing the pace of a standard progress bars, the pattern replaces it with a more informative, step based system. The pattern describes a way to inform the user of what steps they have taken, the current step the user is on, and what steps are left for the user to complete without indicating a specific percent completed.

Application We suggest the the current LoCuS notebook's progress bar be changed to better inform the student of their position and progress in the lab. The current LoCuS notebook contains a limited progress bar that informs the student that they have moved closer to the end of the lab. The current progress bar is shown in Figure 4.15. Each page in the notebook is separated by a XML `<page break \>` although the page splits if the physical screen size is smaller that the defined page. This results in the progress bar appearing to be non-functional on new pages that were not generated by a `<page break \>`. Another issue with the current implementation of the progress bar is that the amount of information between page breaks can vary greatly. Both of these issues render the current progress bar useless since it portrays no useful information to the student.



In the formula above:

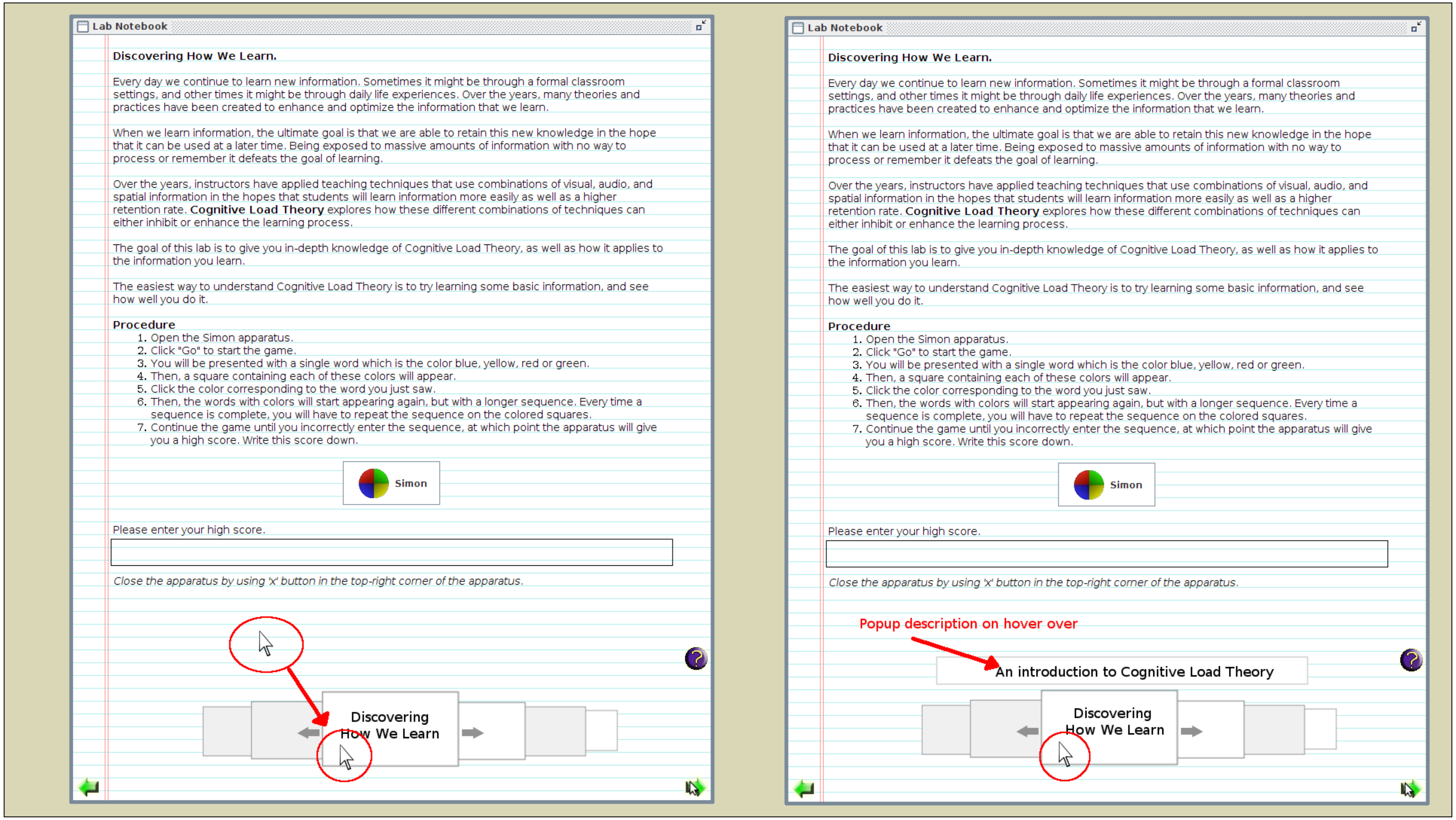
- *k* stands for the total number of customers that enter the line we are examining.
- t_s represents the time instant (e.g., 8am) we begin our observation.
- And t_e represents the time instant (e.g., 5pm) we end our observation.

For now, we will measure time in minutes.

Figure 4.15: Current progress bar

Our suggested course of action would be to create a new progress bar that displays useful information to the student. The new progress indicator can be placed in the same location as the former progress bar or on some unused portion of the screen, possibly in-between the notebook and chalkboard. The new progress indicator would be a list of the page titles with the pages that the student has already done dimmed, the current page highlighted and future pages presented in such a way that the student would know they haven't done them yet. This would give the student some indication of how much progress they have made and how much is left to accomplish. Additionally this would make it possible for the student to jump to a section that they might need to reread without traversing the notebook again. Naturally the presentation of list would have to be aesthetically pleasing and minimally intrusive as to not be a visual detraction. Two possible implementations of a progress are given in alternates one and two.

- *Alternate 1:* Create a progress bar similar to those used for website navigation and some applications. The student's current location in the lab is centered, the sections that student has already visited are darked to indicate completion, and the section the student still needs to complete are lighted to indicate in-completion. The student then uses the arrows to navigate the sections. When the mouse hovers over one of the sections a short popup description will be displayed above the navigation panels. Shown in Figure 4.16.



Discovering How We Learn.

Every day we continue to learn new information. Sometimes it might be through a formal classroom settings, and other times it might be through daily life experiences. Over the years, many theories and practices have been created to enhance and optimize the information that we learn.

When we learn information, the ultimate goal is that we are able to retain this new knowledge in the hope that it can be used at a later time. Being exposed to massive amounts of information with no way to process or remember it defeats the goal of learning.

Over the years, instructors have applied teaching techniques that use combinations of visual, audio, and spatial information in the hopes that students will learn information more easily as well as a higher retention rate. **Cognitive Load Theory** explores how these different combinations of techniques can either inhibit or enhance the learning process.

The goal of this lab is to give you in-depth knowledge of Cognitive Load Theory, as well as how it applies to the information you learn.

The easiest way to understand Cognitive Load Theory is to try learning some basic information, and see how well you do it.

Procedure

1. Open the Simon apparatus.
2. Click "Go" to start the game.
3. You will be presented with a single word which is the color blue, yellow, red or green.
4. Then, a square containing each of these colors will appear.
5. Click the color corresponding to the word you just saw.
6. Then, the words with colors will start appearing again, but with a longer sequence. Every time a sequence is complete, you will have to repeat the sequence on the colored squares.
7. Continue the game until you incorrectly enter the sequence, at which point the apparatus will give you a high score. Write this score down.

Please enter your high score.

Close the apparatus by using 'x' button in the top-right corner of the apparatus.

Lab Notebook

Discovering How We Learn.

Every day we continue to learn new information. Sometimes it might be through a formal classroom settings, and other times it might be through daily life experiences. Over the years, many theories and practices have been created to enhance and optimize the information that we learn.

When we learn information, the ultimate goal is that we are able to retain this new knowledge in the hope that it can be used at a later time. Being exposed to massive amounts of information with no way to process or remember it defeats the goal of learning.

Over the years, instructors have applied teaching techniques that use combinations of visual, audio, and spatial information in the hopes that students will learn information more easily as well as a higher retention rate. **Cognitive Load Theory** explores how these different combinations of techniques can either inhibit or enhance the learning process.

The goal of this lab is to give you in-depth knowledge of Cognitive Load Theory, as well as how it applies to the information you learn.

The easiest way to understand Cognitive Load Theory is to try learning some basic information, and see how well you do it.

Procedure

1. Open the Simon apparatus.
2. Click "Go" to start the game.
3. You will be presented with a single word which is the color blue, yellow, red or green.
4. Then, a square containing each of these colors will appear.
5. Click the color corresponding to the word you just saw.
6. Then, the words with colors will start appearing again, but with a longer sequence. Every time a sequence is complete, you will have to repeat the sequence on the colored squares.
7. Continue the game until you incorrectly enter the sequence, at which point the apparatus will give you a high score. Write this score down.

Simon

Please enter your high score.

Close the apparatus by using 'x' button in the top-right corner of the apparatus.

Popup description on hover over

An introduction to Cognitive Load Theory

Figure 4.16: An in-text progress indicator. The center page contains the current section name. When the mouse is moved over it a short description would popup above or below the indicator. The pages to the left would be pages before the current location and the pages to the right would be those that are after the current page. Pages that have been visited would be darker than pages that have not.

- *Alternate 2:* Create a progress bar that mimics the appearance of page tabs or section cut-outs. These tabs/cut-outs would be located to the side and would allow the student to traverse the notebook through clicking the tab. Hovering over the tab would make the page slide out and slightly darken the area around it. This would also produce a popup that indicates which section the tab is related to. As in Alternate 1, the color and saturation of the text would change depending on if the student has completed the section, is currently in the section, or has not visited the section yet. Adding tabs would also require the dimensions of the notebook to change slightly in order to accommodate the space needed to hold the tabs.

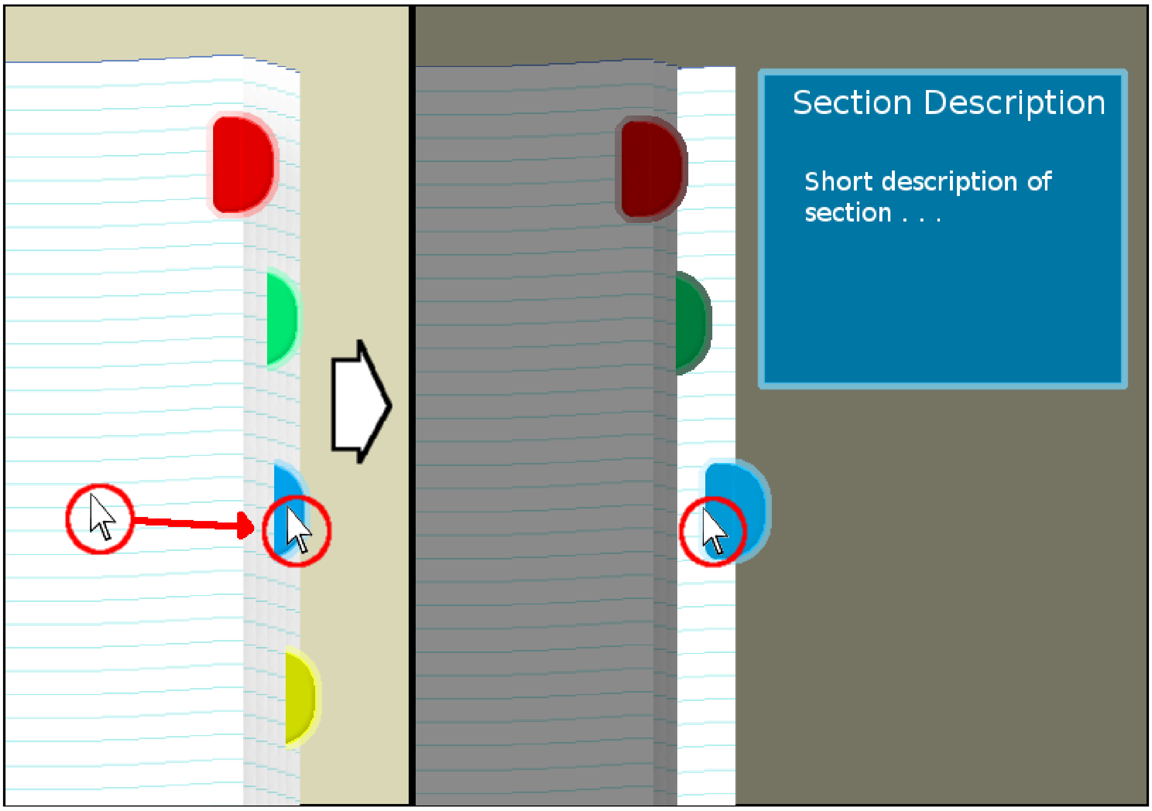


Figure 4.17: Using tabs as a progress indicator. When the mouse moves over the tab the page slides out and all of the area surrounding the page would darken. A popup with the section title and a short description would appear. If the student clicked on the tab or the protruding page they would jump to the section indicated by the tab.

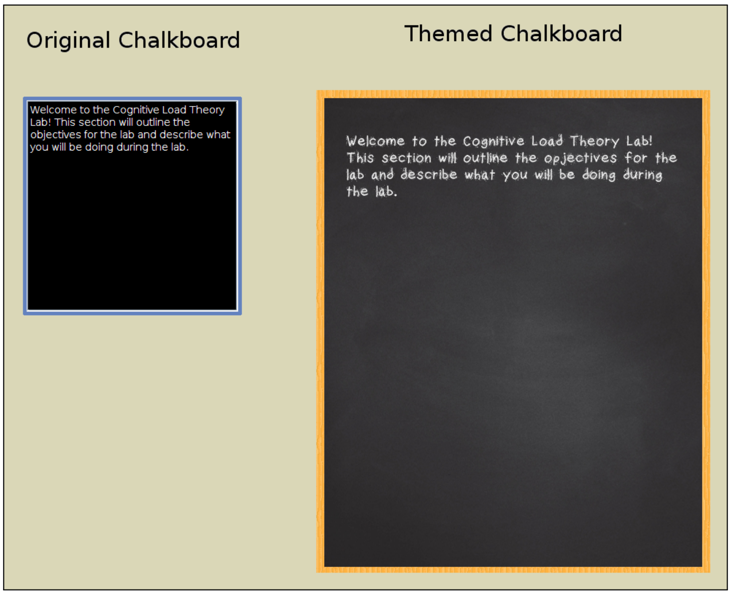
Supplemental Option: We also suggest adding a save current progress option to the user progress bar. Currently LoCuS automatically saves the students work, however the student isn't necessarily aware of this feature. By adding a save button the student can exit the application assured that their current work has been accounted for. This should increase the flexibility of the labs with regards to a student's schedule and possibly remove any apprehension that a student might have about closing the application.

4.6 Theming

Idea We suggest using *theming* to increase the continuity of a software environment. By using theming on elements that the user interacts with, we are creating a uniform environment that immerses the user in the application [8, pp. 477–498]. Theming is prevalent throughout software and web projects. Many times it is used to enhance the users experience by creating a uniform look and feel of the environment. Theming is also heavily used in game development to immerse and engage the user in the game environment.

Application We suggest theming the different components of LoCuS to match the laboratory theme that LoCuS is based on. We have suggested theming in as part of other sections of this chapter. For instance, converting the notebook to a virtual document would require theming the notebook and has been discussed in Section 4.3. Along with the suggestions in other sections, theming could be applied to the chalkboard and apparatuses.

Chalkboard: The chalkboard could be enhanced through theming by removing the windowing and changing the background and font of the chalkboard. This would better represent a real world chalkboard as shown in Figure 4.18.



Welcome to the Cognitive Load Theory Lab! This section will outline the objectives for the lab and describe what you will be doing during the lab.

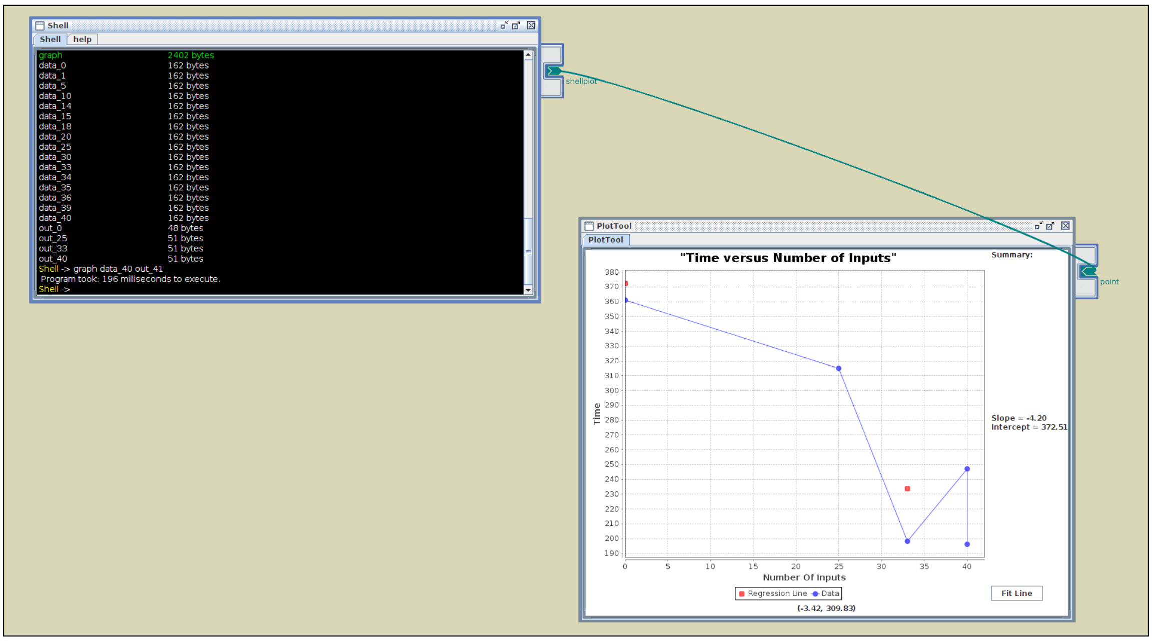
Welcome to the Cognitive Load Theory Lab! This section will outline the opjectives for the lab and describe what you will be doing during the lab.

Figure 4.18: The chalkboard before and after applying *theming*. The new updated chalkboard would have a textured background, a wooden frame, and a font that imitated chalk writing.

Apparatuses: Enhancement of apparatuses could be accomplished by replace the windowing with a style that matched the desktop theme. Figure 4.19 is the current implementation of the apparatus theme. We have made two alternate suggestions as to how to theme the apparatuses.

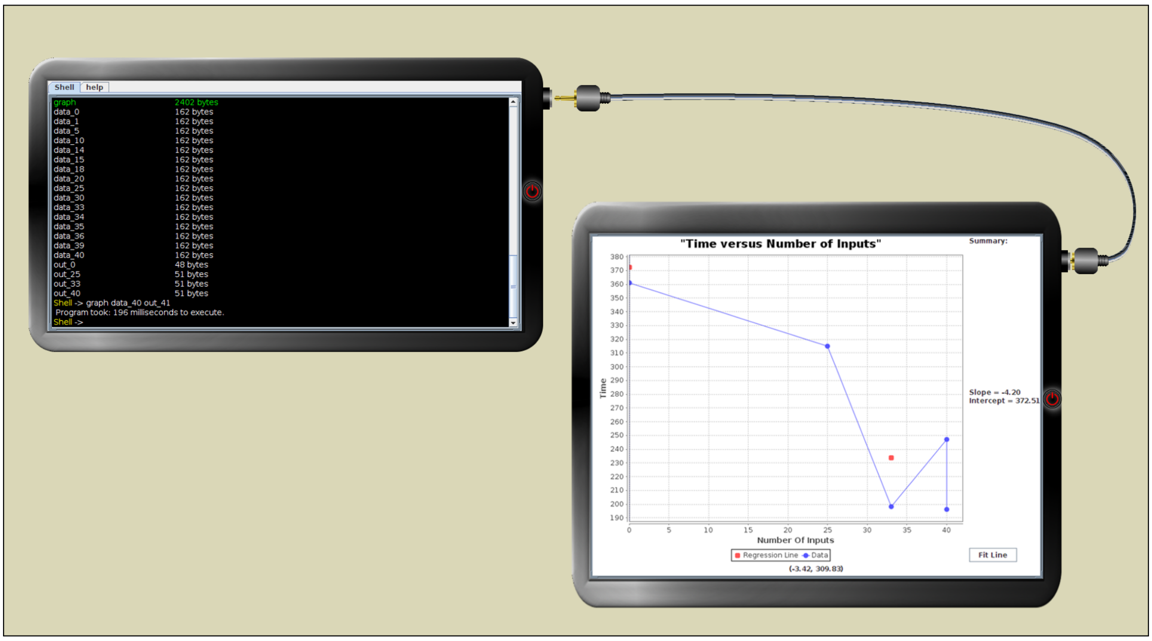
- *Alternate 1:* Change the windowing of apparatuses to simulate a tablet frame with the apparatus inlaid in the writing area. This would resemble a tablet that the student might use in class or a laboratory experiment. Shown in Figure 4.20.
- *Alternate 2:* Change the windowing of apparatuses to simulate an electronics testing workbench tool similar to a spectrum analyzer or an oscilloscope. This would resemble a tool they might interact with in a physical lab environment. Shown in Figure 4.21.

Supplemental Option: Another possibility would be allowing the lab authors or developers to chose between alternate one and two. By allowing them to choose a theme, we would be giving them greater flexibility in content design and increasing enjoyability for the student.



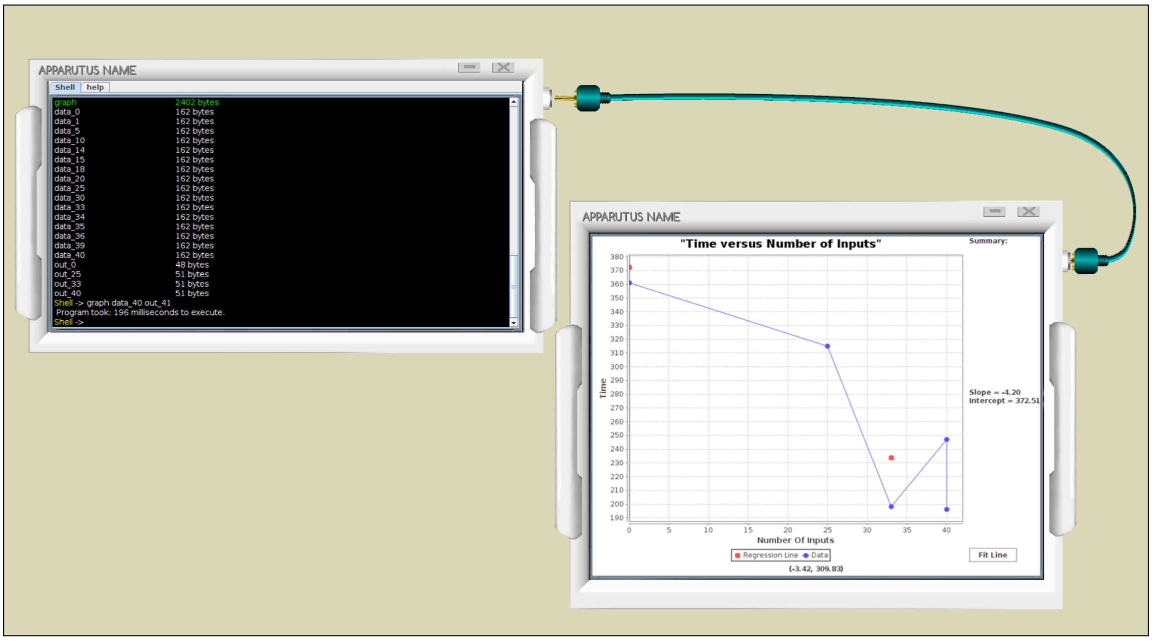
```
graph                 2402 bytes
data_0                162 bytes
data_1                162 bytes
data_5                162 bytes
data_10               162 bytes
data_14               162 bytes
data_15               162 bytes
data_18               162 bytes
data_20               162 bytes
data_25               162 bytes
data_30               162 bytes
data_33               162 bytes
data_34               162 bytes
data_35               162 bytes
data_36               162 bytes
data_39               162 bytes
data_40               162 bytes
out_0                 48 bytes
out_25                51 bytes
out_33                51 bytes
out_40                51 bytes
Shell -> graph data_40 out_41
 Program took: 196 milliseconds to execute.
Shell ->
```

Figure 4.19: The original Java theme for apparatuses. The line between them is a connector used to send data from one apparatus to another.



```
graph                 2402 bytes
data_0                162 bytes
data_1                162 bytes
data_5                162 bytes
data_10               162 bytes
data_14               162 bytes
data_15               162 bytes
data_18               162 bytes
data_20               162 bytes
data_25               162 bytes
data_30               162 bytes
data_33               162 bytes
data_34               162 bytes
data_35               162 bytes
data_36               162 bytes
data_39               162 bytes
data_40               162 bytes
out_0                 48 bytes
out_25                51 bytes
out_33                51 bytes
out_40                51 bytes
Shell -> graph data_40 out_41
 Program took: 196 milliseconds to execute.
Shell ->
```

Figure 4.20: Two apparatuses with tablet themes. The connector between them has been change to a data transfer cord and the close button has been move to the side to mimic the power button on a tablet.



```
graph                  2402 bytes
data_0                 162 bytes
data_1                 162 bytes
data_5                 162 bytes
data_10                162 bytes
data_14                162 bytes
data_15                162 bytes
data_18                162 bytes
data_20                162 bytes
data_25                162 bytes
data_30                162 bytes
data_33                162 bytes
data_34                162 bytes
data_35                162 bytes
data_36                162 bytes
data_39                162 bytes
data_40                162 bytes
out_0                  48 bytes
out_25                 51 bytes
out_33                 51 bytes
out_40                 51 bytes
Shell -> graph data_40 out_41
 Program took: 196 milliseconds to execute.
Shell ->
```

Figure 4.21: An electronics testing workbench tool theme for a set of apparatuses. The cord has been changed to match the tools and the close button is in the same location as it would be in a normal window.

4.7 Additional Multimedia

Idea Multimedia has been used in texts, websites, software, and classrooms to increase a student's ability to learn more deeply [6, pp. 111–122]. Several forms of multimedia include images, sounds, narration, video, and animation. By adding multimedia elements to educational software it is possible to increase a student's retention and understanding of the given material. A description of how Mayer tested these different forms of multimedia to find out how effective they are in education is given in Section 3.1.

Application Currently LoCuS is a multimedia environment designed to teach students scientific principles related to computer science. In LoCuS the notebook and apparatuses are interactive and the chalkboard content correlates with the notebooks content. We suggest adding video, animation, and narration to the LoCuS labs in order to make it possible for the lab author to enhance the usability and functionality of a lab.

Videos and Animation: When creating a labs content the author has the options to add static text, images, questions of varying form, and launchers for apparatuses. We suggest adding videos and animation to this list. By adding the ability to play videos and animations such as animated Gif files the lab author would have access to a new range of educational resources. Adding video with sound could cause problems in group classrooms or for the deaf. To account for situations where sound would be a problem we suggest that video be accompanied by a closed captioning option.

Narration and Animation: Narration has been shown to increase student retention and understanding when it is included with an animation illustrating the concept being conveyed [6]. We suggest adding functionality for a narration to be played with animated elements in LoCuS. This would give instructors the option to create animations with a verbal guide to help demonstrate difficult topics, or simplify descriptions that would normally require a significant written explanation. As with video, narration could cause problems in group classrooms or for the deaf. We suggest adding an option similar to closed captioning that would allow students to read along with the narration. This could be done with scrolling text that can be enabled or disabled.

4.8 Accessibility

Idea Accessibility for the color-blind should be increased by designing software and websites to accommodate their visual range. Color blindness effects around 8% of males and 0.5% of females [5, pp. 59]. Software and websites can be designed to accommodate the affected people by choosing the correct colors during the design process. In his book "Designing with the Mind in Mind", Johnson gives a color pallet that helps elements standout for the color blind [5, pp. 58–63]. He also explains that designers can approximate color-blindness by converting an image of their interface to gray-scale [5, pp. 59–60]. Any colors that appear to be the same in a gray-scale image will be challenging for a color-blind person to see. An example of this is presented in Figure 4.22.

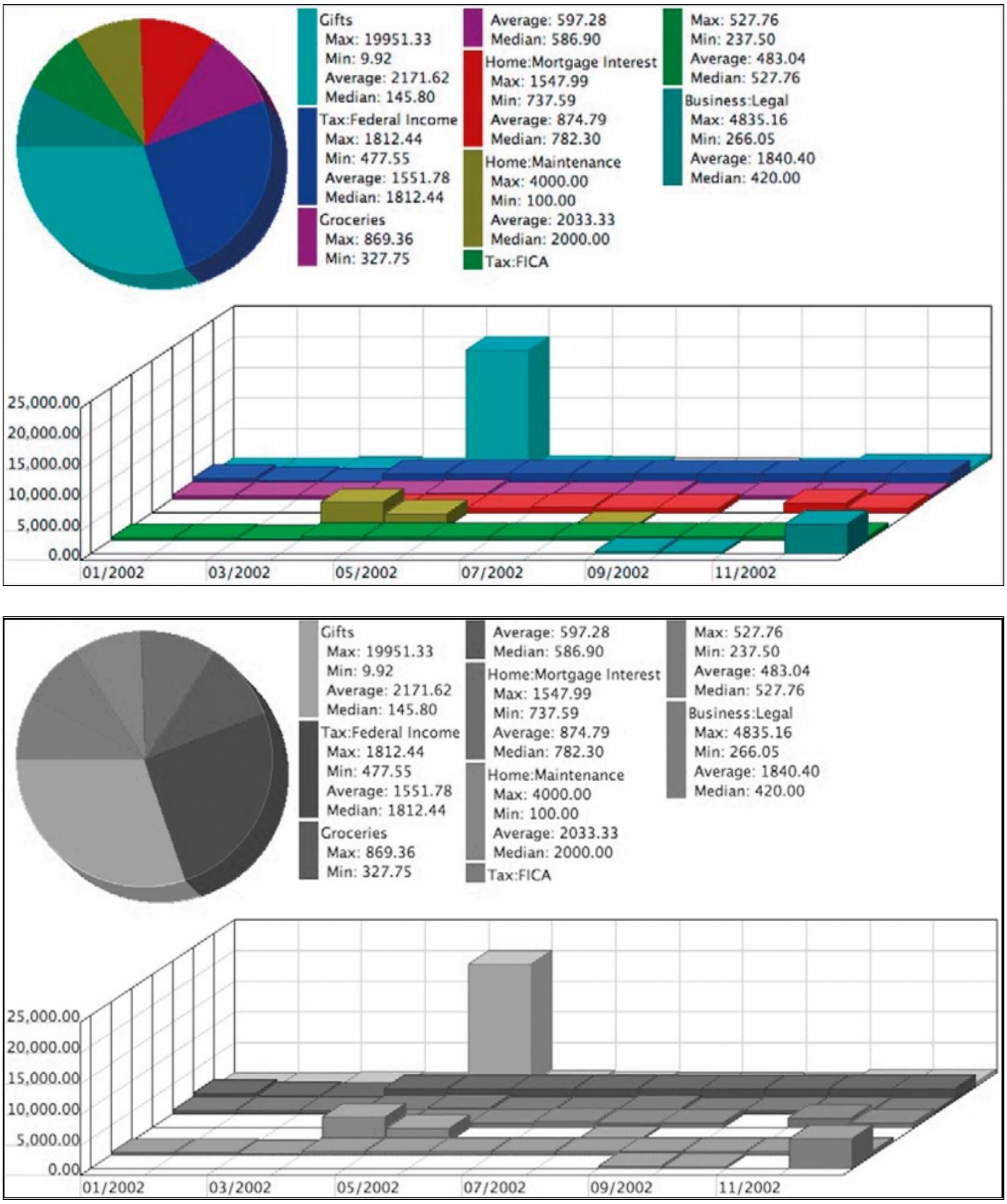


Figure 4.22: A colored image converted to gray-scale to approximate color-blindness [5, Fig 5.10, 5.11].

Application We suggest changing any part of LoCuS that is difficult for a color blind person to see. The best way to do this is by creating a short set of guidelines for LoCuS developers and authors regarding color blindness. The guidelines could direct the developers and authors to a website the would guide them through the process of choosing colors for the color-blind. The guidelines could also include set of instructions on how to convert the current screen to gray-scale so that a person who is not color-blind can approximate color-blindness when choosing colors. Additionally, an Eclipse pluggin could be developed that would convert a project to gray-scale to help the developers spot areas that are difficult for the color-blind. We did not consider blind people in when making suggestion for accessibility. LoCuS is a very visual environment and adding accessibility for the blind would be difficult to impossible.

Chapter 5

Implementation

This chapter discusses the suggested enhancements that were implemented within the LoCuS environment. We do not go into the specific code details, rather we describe the final results of the implemented enhancements in the LoCuS software system. An evaluation of each of the implemented suggestions is covered in Chapter 6.

The first suggested enhancement we implemented was in Section 4.1, changing the screen layout to increase the aesthetic appeal. We changed the location of the notebook and chalkboard components of LoCuS. The chalkboard was also resized to allow for more content. The new locations are the same as in the suggested enhancement so they appear as they do in Figure 4.3. We made sure that the chalkboard dynamically resizes depending on the screen resolution so that it does not appear off of the screen. The chalkboard is also sized so that it is not obscured by a running apparatus. Resizing according to the screen resolution was already a feature of the notebook. We relocated and resized these components to match the suggestions by Altaboli and Lin [2].

We implemented the suggested enhancement in Section 4.2 to match the recommendations from Tidwell [8]. Tidwell recommended using a contrasting backgrounds that was, at least, slightly out of focus, to differentiate between the foreground and background. We chose a wooden background, as suggested in Section 4.2. Alternate one was chosen because it matched the school laboratory desktop theme that is present in LoCuS. To achieve the wooden desktop effect we created a woodgrain image and modified it so that it was slightly out of focus. We chose not to include the supplementary option of allowing the instructor to choose a background. Even though allowing an instructor to choose the background would have been a nice feature, it opened up the possibility that they would choose a background that did not agree with the Deep Background recommendations. The final implementation is similar to Figure 4.5, however the resolution of the woodgrain is finer. No figure showing the implemented version is included in this document since the resolution is such that it would just appear brown. When the wooden background is viewed through a computer screen the details of the wood are visible.

We implemented suggestion 4.6 to include theming of the chalkboard and apparatuses. Theming is also present in suggestions 4.2, 4.3, and 4.5. Theming included outside of Section 4.6 will be implemented with its corresponding suggestions. Here, we only discuss the chalkboard and apparatuses. The chalkboard was given a frame and its background was changed to mimic a used chalkboard that has been erased. The apparatuses were given a tablet frame. The apparatus frames are smaller than they appear in Figure 4.20 so that they do not use so much space as to hinder the student's ability to see the notebook or chalkboard. The connectors were left the way they were already implemented, with only the color of the socket modified. We chose not the change the shape of the connectors since all of the labs include instructions for using them in their current state. The connectors and sockets can be change in the future when it is possible to change the lab instructions to match.

We implemented part of the Section 4.8 to help developers and lab authors create content that is accessible for the color-blind. We added a chapter to the LoCuS Getting Started Guide that explains how to design content for the color-blind according to Johnston's suggestions [5]. This chapter gives the developers and lab authors information on choosing colors that are easier for color-blind people to differentiate between. We also included suggestions such as using different shapes to help differentiate between objects that, by necessity, need to be colors that could be confusing for the color-blind. We did not change any of the existing LoCuS labs or apparatus to meet these guidelines or create a pluggin that could convert an Eclipse project to gray-scale as suggested in Section 4.8. These options are included in Chapter 7.

Chapter 6

Evaluation of Implementation

We now evaluate the implemented enhancements from Chapter 5. There were two possible ways that we considered evaluating the implemented enhancements. One way to evaluate the enhancements was through the use of a student survey which is described in Chapter 7. A survey would have been the best method of evaluation, however it was time, and resource, prohibitive at the time of evaluation. Instead we chose to evaluate the implemented enhancements by comparing them to the literature described in Chapter 3. If our enhancements agreed with the suggestions in the literature then we can reasonably conclude that they were successful to some measure and thus increased usability in LoCuS.

To evaluate the screen layout changes that were implemented from the suggestions in Section 4.1, we compared it to Altaboli and Lin's paper on screen layouts [2]. In their paper they showed that increasing the balance between the components size and location makes the environment more visually appealing. In our implementation we increased the balance between the chalkboard and notebook's size by increasing the chalkboard size. We also increased the location balance between these two components by locating them at equal distances from the upper and outer edge of the main frame of LoCuS. By changing the size and location of the notebook and chalkboard we have conformed to the research suggestion and increase the aesthetic appeal of LoCuS, thus increasing usability.

We evaluated the implementation of Section 4.2 by comparing it to Tidwell's user design pattern, Deep Background [8]. Her design pattern suggest that a contrasting background that is blurred be use to set the foreground components apart from the background of the application. Our implementation uses a slightly blurred woodgrain background that contrasts with the white notebook and black chalkboard. Our implementation conforms to Tidwell's Deep Background design pattern, and increases the usability of LoCuS.

We evaluated theming, covered in Section 4.6, by comparing it to Tidwell's description of visual motifs and overall visual design [8]. Theming is one of the more difficult suggestions to evaluate since there is no straightforward design patterns or guidelines to evaluate against. Tidwell suggested that a strong, consistent visual design is key to increasing the appeal of the application or website that you are creating. She mentions using texture, fonts, textures, and border designs, among other design aspects. Many of these theming ideas are already implemented in LoCuS. For our implementation we chose to increase the consistent visual motif of a laboratory desktop by adding borders and textures to the chalkboard and apparatuses. By doing this, we have followed Tidwell's advice to have a strong and consistent visual design to increase usability.

The evaluation of the color-blind accessibility implementation, Section 4.8, was straightforward. The portion of Section 4.8 that we chose to implement was adding a chapter to the LoCuS Getting Started Guide, instructing developers and lab authors on how to design for the color-blind. All of the design suggestions are summaries of material from Johnston's book [5]. Since our suggestions are derived directly from the literature they conform to it and increase usability for color-blind students.

Chapter 7

Future Work

In this chapter we discuss possible future work. We consider other research and literature on the topic of user interfaces and human computer interaction. We also touch on the three dimensional features of LoCuS, why they are not included in this thesis, and how they can be addressed. Finally, we examine a possible survey that can be done in the future to determine the effectiveness of the implemented enhancements.

Time constraints did not permit us to implement several part of this thesis. The parts that were not implemented are sections 4.3, 4.4, 4.5, 4.7, and the Eclipse color-blind pluggin along with addressing color-blind issues in the current LoCuS environment. Each of these sections requires a significant amount of work and has been left as projects for future undergraduate Honor students. Each section has been described in detail and included in the LoCuS MantisBT feature tracking database for future implementation [1].

There is a significant amount of research and literature on user interface design and human computer interaction that was not covered in this thesis. There could be many other possible enhancements that we never came across or considered. We could investigate other literature that was not covered in the current work or reviewed during the discovery phase, and search for possible enhancements.

The LoCuS software infrastructure includes a complex three-dimensional environment that was not discussed. Due to the differences between a two-dimensional and three-dimensional environment we did not include material on LoCuS's three dimensional-environment. Most of the material in this work only applies to a two-dimensional applications. To properly cover the three-dimensional aspects of LoCuS would required a separate paper to be written.

A user survey could be done comparing a functional version of LoCuS prior to this thesis and a version after the recommended modifications are completed. Running a survey of this type is possible since many of the currently existing labs are complete and the content will not be change, only the user interface. In this way it would be a direct comparison of a minimal user interface and one with user interface design principles applied. The LoCuS software system also includes a student surveying option that can be easily implemented in any lab removing many of the difficulty associated with software surveys.

The reason a survey of this type would be informative is that, to the knowledge of this author, no such study has been done on such a large scale. Most user interface studies are done on small applications that have very little practical real-world use. In many cases the application was created specifically for the study and ask users to play out a hypothetical situation. With this study the user would be responding to survey on an application that has a direct impact on their life in the form of a scholarly grade. The results of the study would be valuable in determining the effectiveness of the recommended user interface design principles in a complex real-world environment. It would also be useful in determining whether the recommendations increased perceived usability, aesthetic value, and immersion into the laboratory environment.

Chapter 8

Conclusion

Through the use of user interface design principles we were able to enhance a virtual laboratory experiment to make it more immersive, to better approximate a physical laboratory experiment and increase the usability for the students. We have examined user interface research and literature to ascertain design principles that would be implemented in LoCuS to increase immersion and usability. By increasing immersion and usability we have improved LoCuS's ability to convey theoretical concepts through laboratory experiments. The more capable LoCuS is of conveying theoretical concepts and drawing the students into the laboratory experiment the higher the likelihood that the student will enjoy learning about computer science theory and not be overwhelmed by the technical aspects of computer science. By teaching abstract theoretical concepts in a manner that removes the technical difficulties and increases students perceptions of computer science, we have a higher chance of increasing enrollment in computer science and possible other formal sciences [7].

Bibliography

[1] TAU Project's Mantis Bug Tracker, August 2015. `http://cgi.cs.arizona.edu/projects/tau/mantis/`.

[2] ALTABOLI, A., AND LIN, Y. Investigating Effects of Screen Layout Elements on Interface and Screen Design aesthetics. *Advances in Human-Computer Interaction* (2011), pp. 1–10.

[3] CHU, Y.-C., BAINBRIDGE, D., JONES, M., AND WITTEN, I. H. Realistic books: A bizarre homage to an obsolete medium? In *Proceedings of the Joint ACM/IEEE Conference on Digital Libraries* (2004), IEEE, pp. 78–86.

[4] HARRISON, C., AMENTO, B., KUZNETSOV, S., AND BELL, R. Rethinking the Progress Bar. In *Proceedings of the 20th Annual ACM Symposium on User Interface Software and Technology* (2007), ACM, pp. 115–118.

[5] JOHNSON, J. **Designing with the Mind in Mind: Simple Guide to Understanding User Interface Design Guidelines**, 2013. Elsevier.

[6] MAYER, R. E. Multimedia Learning. *The Psychology of Learning and Motivation 41* (2002), pp. 85–139.

[7] SAULS, J. E. Changing Perceptions of Computer Science, 2012. Honors Thesis, The University of Arizona.

[8] TIDWELL, J. **Designing Interfaces**, 2010. O'Reilly Media, Inc.

[9] TRACTINSKY, N., KATZ, A. S., AND IKAR, D. What is Beautiful is Usable. *Interacting with Computers 13*, 2 (2000), pp. 127–145.