

TEACHING AND VISUALIZING FITTS' LAW

By

GAVIN JOEL SIMONS gsimons@email.arizona.edu

---

A Thesis Submitted to the Honors College

In Partial Fulfillment of the Bachelors Degree With Honors in

Computer Science

THE UNIVERSITY OF ARIZONA

DECEMBER 2011

Approved by:

---

Professor Richard T. Snodgrass  
Department of Computer Science

## STATEMENT BY AUTHOR

This thesis has been submitted in partial fulfillment of requirements for a degree at The University of Arizona and is deposited in the University Library to be made available to borrowers under rules of the Library.

Signed: \_\_\_\_\_

## Abstract

The purpose of this thesis was to develop a laboratory activity and an associated program which would convey the concepts of Fitts' Law and what Fitts' Law is used for in the real world. The program is called QUIKFITT, which is designed to demonstrate and teach the computer science theory of Fitts' Law to people without a background in computer science and to present it in such a manner that people see it as genuine science. QUIKFITT is part of the larger LOCUS [1] project, which is composed of many programs like QUIKFITT which each demonstrate and explain a different concept or theory from computer science. The majority of this thesis is the QUIKFITT program, which I developed from start to finish and is currently being deployed by the LOCUS project. It was an interesting experience, since I had to research the theory and learn it well enough to teach it. Also, writing the code for everything was fairly difficult since I had to learn several new techniques in areas of computer science that I have relatively little experience with. Overall it was a good experience since I learned both how to handle a large project and work in a team.

# 1 Introduction

Out of the many fields of science, there is one where its members are not seen as scientists by the general public. I speak, of course, about the field of computer science. Despite having the word "science" in the title, computer scientists are often just seen as some sort of engineer, only making products and never doing anything equivalent to the physicist shooting molecules with a laser or the geologist analyzing rocks in a lab. However, this perception is false and there are many theories and concepts in the field of computer science that are on the same scientific level as Newton's Laws of Motion or the Theory of Relativity.

The program QUIKFITT is designed to demonstrate and teach the computer science theory of Fitts' Law to people without a background in computer science and to present it in such a manner that people see it as genuine science. QUIKFITT is part of the larger LOCUS [1] project, which is composed of many programs like QUIKFITT, each demonstrating a specific concept or theory from computer science. To teach Fitts' Law, QUIKFITT is used by the main LOCUS program which runs a lab. This lab is similar in format to a traditional science lab; there is a pre-lab, a procedure, hypothesis, and so on. This is meant to get the student to better connect the theories and concepts being presented with real science.

LOCUS labs in general are also intended to be similar to the computer-based labs which are often seen in high schools. Programs which allow a teacher to do things with their students which may be impossible to do otherwise, such as a virtual dissection of a human or reconstruction of a dinosaur skeleton. Within LOCUS labs programs like QUIKFITT are referred to as apparatuses, since each program is used like a tool in the labs to help teach a concept. Apparatuses are much like equipment found in real-life labs, such as microscopes or spectrometers. The idea is that a high school teacher can use a computer science lab from LOCUS to easily introduce their students to computer science theories and concepts, even if the teacher has little to no background in computer science. This document will focus mostly on background information of Fitts' Law, information on how the program works, the layout of the lab QUIKFITT will use when it is released, possible future improvements, and the experience I had in creating the program.

## 2 The Theory

Fitts' Law was first conceived of in 1954 by Pail Fitts', and has since developed to a formula which can predict the amount of time it will take a user to move a pointer between two specified points on a computer screen using any sort of input device, which could be a mouse, a keyboard, or even a stylus [4] [5]. A subject moves a pointer from a point A to a point B, and the time it takes the subject to do that is recorded. This is repeated several times, with the distance between A and B and the distance across B varying. Once all this data is collected, it is put into the Fitts' Law formula so that it can account for the user's unique reaction times. Then, we are then able to predict how long it will take the subject to move the pointer between any points A and B.

### 2.1 Fitts' Law Formula

The following goes into detail about the formula and the variables that make it up:

Fitts' Law:  $MT = r + s \cdot \log_2\left(\frac{D}{A}\right) + 1$  [3]

- *Task*: A task consists of moving a pointer from the start to the target as quickly and accurately as possible.
- *start*: The first point selected in the task by using an input device, meant to represent a random point on a computer screen where a user would begin before moving the pointer. It is from here that we begin measuring  $D$ .
- *target*: This is the second point that is selected in the task, meant to represent the point a user would move a pointer to. It is the center of target where we stop measuring  $D$ .
- $r$ : The y-intercept coefficient in the regression. A constant, representing the time needed to use the input device and their reaction time. An example would be depressing the mouse's left button once the target is reached. The user first needs to register that they have entered the target, and then they must push the button, a process which is not instantaneous. This varies from person to person and machine to machine.
- $s$ : The slope coefficient in the regression.  $s$  is representative of the user's human motor system's capacity to process information, essentially their reaction time.
- $MT$ : Movement Time, the total time it takes to get from the start to the target. In this lab, we measure  $MT$  in milliseconds.
- $D$ : the distance from start to the target's center  
 $A$ : the distance across the target. The line that  $S$  is measured on is a continuation of the line formed by the point where the cursor leaves *start* and enters *target*.
- *Index of Difficulty*: represented by the  $\log_2\left(\frac{D}{A}\right) + 1$  part of the equation, represents the relative difficulty of moving the pointer from the start to target using an input device, for that particular person.

### 3 The QUIKFITT Apparatus

#### 3.1 Introduction

An apparatus is the term used in the LOCUS project for the specific programs used to help demonstrate theories or concepts of computer science. These programs can be anything from QUIKFITT here, to a program that graphs data onto xy-coordinates. These apparatuses are then used by the labs in LOCUS whenever the student running the lab needs them. The QUIKFITT apparatus' basic structure and functionality are described here. Through activities that are abstract, analogies, or imitations of GUIs, Fitts' Law is demonstrated by gathering and using user generated data. These activities are simulated chemistry labs, abstract activities involving dots, and simulated GUIs.

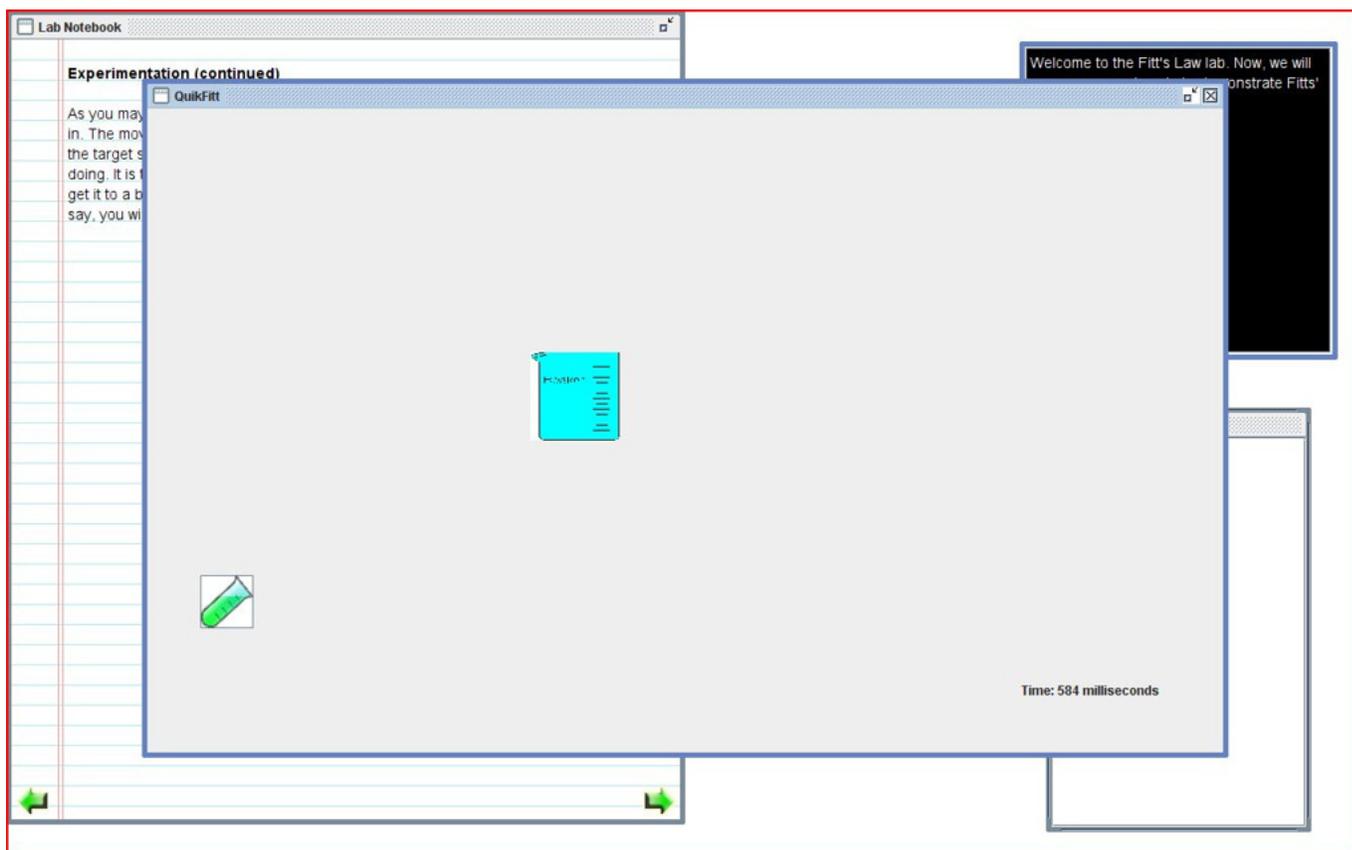


Figure 1. The QUIKFITT apparatus being run from a LOCUS lab

## 3.2 Functionality

There are several different activities which can be generated by this apparatus, each being distinct. These activities follow three main themes: a chemistry lab, a simulated desktop, and an abstract view. However, the themes are only there to help ease the user into the concepts of Fitts' Law, starting with the chemistry lab and desktop to help show how Fitts' Law functions in the real world. The abstract view is a simplified view with only dots, for when the user has a better understanding of Fitts' Law and is ready to see it without any real-world context.

## 3.3 User Interface

QUIKFITT has three tabs: Main, Help, and Instructions, all of which are located along the top and are part of the common layout. The first tab from left to right is the Instructions tab. The Instructions tab contains instructions on how to do the current activity and the instructions screen is the first thing the User sees when QUIKFITT comes up. The next tab is the Main tab. The Main tab of QUIKFITT displays one of the QUIKFITT activities. The last tab is the Help tab. The Help tab brings the user to the help menu for QUIKFITT. The help tab is divided into two main areas: general and activity-specific. The general help covers QUIKFITT in general, helping with things like error messages, what each tab is for, and so on. The activity-specific help covers things specific to the current activity. To exit the activity, the user just clicks on the x-button in the upper-right of the frame. After clicking it a window will pop up asking the user if they are sure they want to quite and warning them that all data from the activity will be lost. This will happen no matter which kind of activity the user is doing.

## 3.4 The Activities

The activity displayed is determined by the XML code from the Lab. The activities all revolve around clicking on a start area then clicking on an end target area. In the bottom left corner of the Activity a timer is displayed. This timer shows the user how much time it took to get from the start area to target area. The activity can have one of three main themes, a chemistry lab theme, a computer desktop theme, or an abstract theme. In the chemistry activities, the user clicks on various pieces of lab equipment, in the abstracts the user is just clicking on dots, and in the desktop activities the user clicks on desktop icons, task bars, and other desktop elements. The distances between the start and end points and the width of the end point are generated by reading in a data file containing pairs consisting of a distance and width. The format of the data file and what happens if there are errors in it are specified below in 3.8.

### 3.4.1 Common Activity Elements

Each activity has certain elements in common with all other activities which do not change due to theme. The distance between the start point and target point and width of the target point are determined by the data file that is read in when QUIKFITT is first activated. The data file is a text file where on every line there are two ints. The first int is the distance between the start area and target area, and the second int is the width of the target area. The location of the target point is always either fixed or randomly determined, and if it is randomly determined it will always be the required distance away from the start point. If there is no data file, then the distance between the start point and target point and the target point width will be randomly

determined. Whenever the user successfully goes from the start point to the target point, the time it took to do so is displayed in the upper left corner.

### **3.4.2 Chemistry 1D Horizontal**

This activity focuses on horizontal movement and uses a chemistry lab theme. The user sees a lab bench from the side and a test tube is generated on the left side. The user is to move the test tube to a beaker that is shown on the right side by first clicking on the test tube then on the beaker. When the user selects the test tube, it will move along with the mouse to simulate how a person would have to carry the test tube to the beaker in a real chemistry lab environment. In all of the Chemistry activities, when the chemicals are mixed, there is a short chemical reaction animation as the time is displayed. This animation is just a poof of smoke over the beaker and a change in color of the liquid in the beaker.

### **3.4.3 Chemistry 1D Vertical**

This activity focuses on vertical movement and uses a chemistry lab theme. There is a buret suspended over a beaker. What the user needs to do is tighten the clamp that holds the buret then release the liquid by turning the valve at the bottom. This is done by clicking on the clamp then on the valve. Tightening the clamp triggers an animation where the clamp turns to tighten, and releasing the liquid triggers an animation of the valve turning to release the liquid and of the liquid in the buret draining into the beaker. After the draining animation is done, the chemical reaction animation will trigger.

### **3.4.4 Chemistry 2D**

This activity focuses on movement in a 2D environment. Like in Chemistry 1D Horizontal, the user needs to get a test tube to a beaker. However, this time the straight line between the test tube and beaker is at an angle.

### **3.4.5 Abstract 1D Horizontal**

This activity is an abstract demonstration of Fitts Law. There are no analogies or metaphors involved, and it is the simplest way the concept can be shown in this program. There are two dots on the panel, one labeled start and the other labeled end. The "start" dot will have constant size, while the "end" dot will have a variable radius, changing after each set is completed. The dots will always be on the same horizontal line. It is timed how long it takes the student to click the end dot after clicking the start dot. After clicking on the start dot, it will be dragged along with the mouse until the end dot is clicked. This will be true for all Abstract activities.

### **3.4.6 Abstract 1D Vertical**

This activity is the same as Abstract 1D Horizontal, except the dots are always vertically aligned.

### **3.4.7 Abstract 2D**

This activity is similar to Abstract 1D, but has both vertical and horizontal movement instead of strictly horizontal or vertical.

### **3.4.8 Desktop Infinity**

This activity shows a simulated desktop, complete with desktop icons and a task bar. The user will click their mouse at the Start icon, and then go to the icon near the bottom of the frame. In this activity, the mouse is captured by the panel, causing it to stop at the edge of the panel. The user will be told by a pop-up frame that the mouse is captured by the activity's panel. This activity is meant to demonstrate how the edges of a computer screen technically have infinite width, making it very easy to get to buttons on the task bar. This "infinite width" is there because even if you move the mouse much farther than you need to when going towards the task bar, the mouse cursor will always stop at the edge of the screen. This "infinite" width effectively makes the target area very easy to get to, primarily because its impossible to overshoot the target area. This is caused by the way the mouse cursor stops when it hits the edge of the screen. It can't go any farther, no matter how far you moved the mouse, so it is a physical impossibility to overshoot the target area. So, according to Fitts' Law, it will be much more efficient to place things at the edge of a screen as opposed to elsewhere.

### **3.4.9 Desktop Wrap-Around**

This activity shows a simulated desktop, complete with desktop icons and a task bar. The user will start their mouse at the Start icon, and then go to the highlighted icon near the bottom. In this activity, the mouse wraps around from one side of the panel to the other. This activity is meant to contrast with the GUI Desktop Infinity activity, showing that it will take longer to click on the same targets in this activity than it did in the Infinity activity. This will be because there will be no barrier that will stop the mouse.

### **3.4.10 Desktop Dropdown**

This activity shows a simulated desktop, but there are no desktop icons and the task bar cannot be interacted with. The user will be instructed to right-click their mouse, causing a drop-down menu to appear with the mouse pointer in one of the top corners. One of the options in the drop-down menu will be highlighted when it appears, and the user will have to click on it.

### **3.4.11 Desktop Dropdown Distance**

This activity shows a simulated desktop, but there are no desktop icons and the task bar cannot be interacted with. The user will be instructed to right-click the mouse, causing a drop-down menu to appear a pre-determined vertical distance from the mouse pointer. One of the options in the drop-down menu will be highlighted when it appears, and the user will have to click on it. This activity is meant to contrast with the GUI Dropdown activity to show that drop-down menus appearing with the pointer in the corner is the most efficient layout.

### 3.4.12 Keyboard Activities

For all the previously defined activities, there is a Keyboard version. In a Keyboard activity, everything is exactly the same as the normal version, but instead of using the mouse the user uses the arrow keys. They will use the arrow keys to move a selector around and use the enter key to select the start and target areas.

## 3.5 XML Specification of Options

When putting QUIKFITT into a lab, the lab author must decide on which activity to display and which data file to read in for the start/end distance and end target width. These are shown in the sample XML node:

```
<options>
  <dataFile fileName = "data.txt"/>
  <activity type="Chemistry1DVertical"/>
  <deviceType device="mouse"/>
  <numberPoints number="20"/>
</options>
```

The `dataFile` element has only one attribute, `fileName`. This attribute is used to specify what data file QUIKFITT will read in to make the pairs. If the `dataFile` element is omitted, then QUIKFITT will randomly generate pairs.

The `activity` element has the `type` attribute, which tells QUIKFITT which activity to run. When specifying which activity you want to run in the XML code, you simply put in the activity's name but with no spaces, as shown in the example XML code above. The valid choices for an activity are: `Chemistry1DHorizontal`, `Chemistry1DVertical`, `Chemistry2D`, `Abstract1DHorizontal`, `Abstract1DVertical`, `Abstract2D`, `DesktopInfinity`, `DesktopWrapAround`, `DesktopDropdown`, and `DesktopDropdownDistance`. The `device` attribute from `deviceType` tells QUIKFITT which input device it will be receiving input from, a mouse or keyboard. The `numberPoints` attribute has `number`, which tells QUIKFITT how many of the data points read in from the file to use. If the pairs were randomly generated, `numberPoints` tells QUIKFITT how many pairs to randomly generate. If `numberPoints` is omitted but an input file is specified, then QUIKFITT will read in and use the entire input file.

## 3.6 Implementation Details

In the Chemistry activities, an animated GIF will be used for the chemical reaction animations.

## 3.7 Coupling with Other Apparatuses

The QUIKFITT apparatus sends data over to the spreadsheet and plotter apparatuses. The data will be saved in the format of a text file of three integers with distance first, the width of the target, and finally then the movement time. The data will be sent over to the PLOTTER in the form of an XML node. The data sent over will be generated by using the data gathered from the user during the activity. The integers in the files are in

the units of centimeters for distances and milliseconds for time. The distances from the input are just scalars, the actual distance saved in the output file will depend on the screen size of the computer running QUIKFITT. By using the PLOTTER, the user will be able to plot their data and do a linear regression on the data, giving them a complete Fitts' Law formula. Once the formula is complete, they will be able to make predictions on how long movement times will be if they know the distance and the target's width. The gathered data will be sent over to the plotter or spreadsheet all at once in the form of an input file, so we won't need to have the plotter or spreadsheet up while QUIKFITT is running. This will happen automatically, so the user will not need to worry about transferring the data themselves.

### 3.8 Input File

The input file that is read in by the QUIKFITT program needs to be in a specific format, otherwise the program will not function properly. The data file is just a text file, where on each line there are two integers separated by a space. The first int is the distance to the target, and the second int is the width of the target. These integers are only considered valid if they are within the range of 1-100 inclusive. These widths and distances are scaled to different widths and distances when the program runs, while keeping them all relative. If an invalid distance or width is read in, it is ignored and QUIKFITT continues reading in the file. Also, an error message is sent to the console for debugging and testing purposes. This is something the user will not see.

## 4 QUIKFITT in action

QUIKFITT can best be shown using the `Abstract1DHorizontal` activity. It showcases QUIKFITT's main functionality, that of having a user click a start and target area and time how long it takes. The following images illustrate how a user generates a data point.

First, the apparatus comes up. There is a button labeled "start" on the left and a button labeled "finish" on the right, as shown in Figure 2.

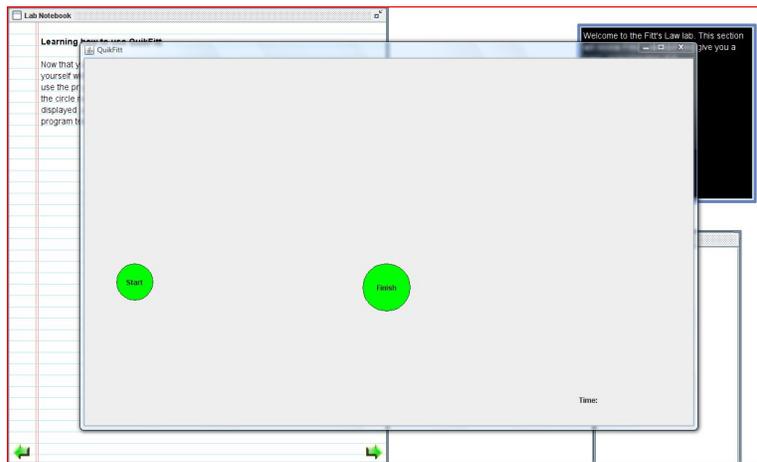


Figure 2. QUIKFITT when it first comes up, before the user has done anything

Figure 3 shows that once the user clicks on the start button, it changes color to indicate that it has been pressed. Figure 4 shows that after the user has clicked on the start button and then the finish button, the apparatus gives the user a new pair of buttons and displays the time from the last set.

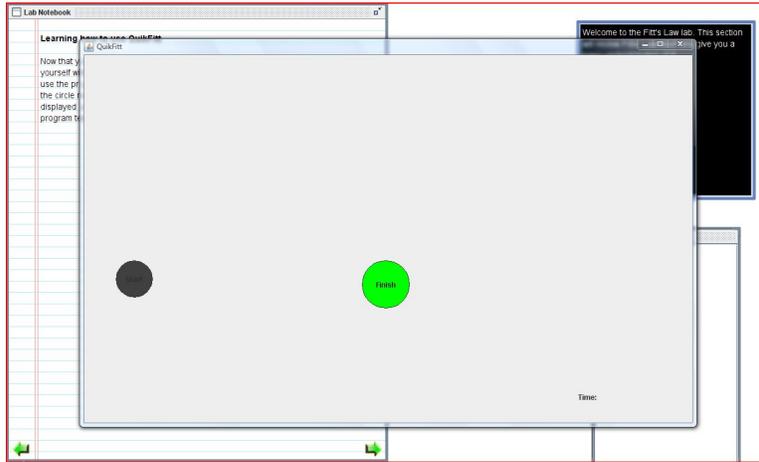


Figure 3. QUIKFITT just after the 'start' button has been clicked

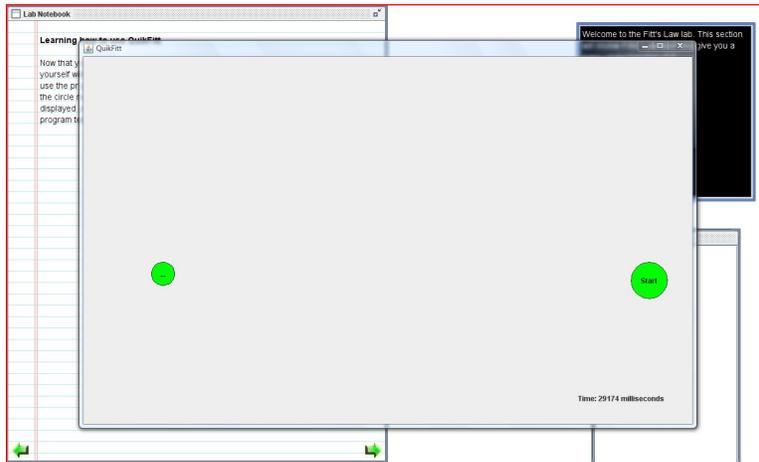


Figure 4. QUIKFITT just after the 'target' button has been clicked and the apparatus has reset itself for the next data point

## 5 The Fitts' Law Lab

Like all labs in LOCUS, the Fitts' Law lab is divided into introduction, pre-lab, lab, and post-lab sections. There are also the Chalkboard and Lab Bench, which are situated on the right of the screen (see Figure 5). The Chalkboard is used by the lab to show brief messages, and the Lab Bench is where the user can access apparatuses used in the lab. The introduction just consists of a question asking if the user would like to participate in a survey meant to help increase the quality of the LOCUS labs and a brief introductory paragraph about Fitts' Law. The lab then goes into the pre-lab section (see Figure 5). Here the user learns just what Fitts' Law is, and the formula is explained in detail. Both analogies and the actual definitions are used to help ease the user into these new concepts. The user is also introduced to the apparatus, running one of the more abstract activities so they can get used to how it works before they begin gathering data. The pre-lab section ends with a quiz, testing the user to make sure they understand Fitts' Law (see Figure 6). The next portion of the lab does the actual experimentation, and it is very straight-forward. The user states a hypothesis, a procedure is given, and the user then completes several chemistry-lab themed activities from the apparatus. The section ends with some questions asking the user about what conclusions they made during the lab. The post-lab section helps the user draw connections to real-world applications of Fitts' Law and asks a few more questions relating to what other applications of Fitts' Law the user can think of. After this, there is just the survey which the user was asked if they would like to do at the beginning of the lab.

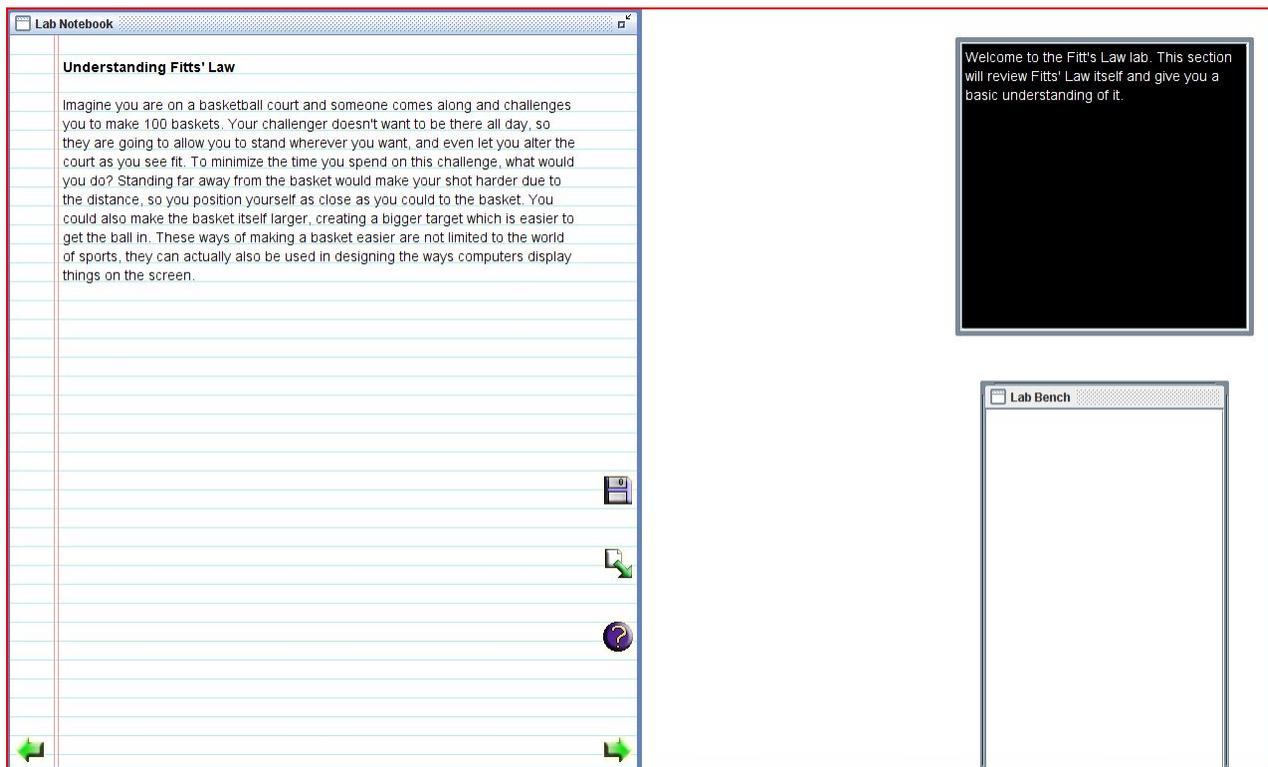


Figure 5. An image from the pre-lab section of the QUIKFITT lab

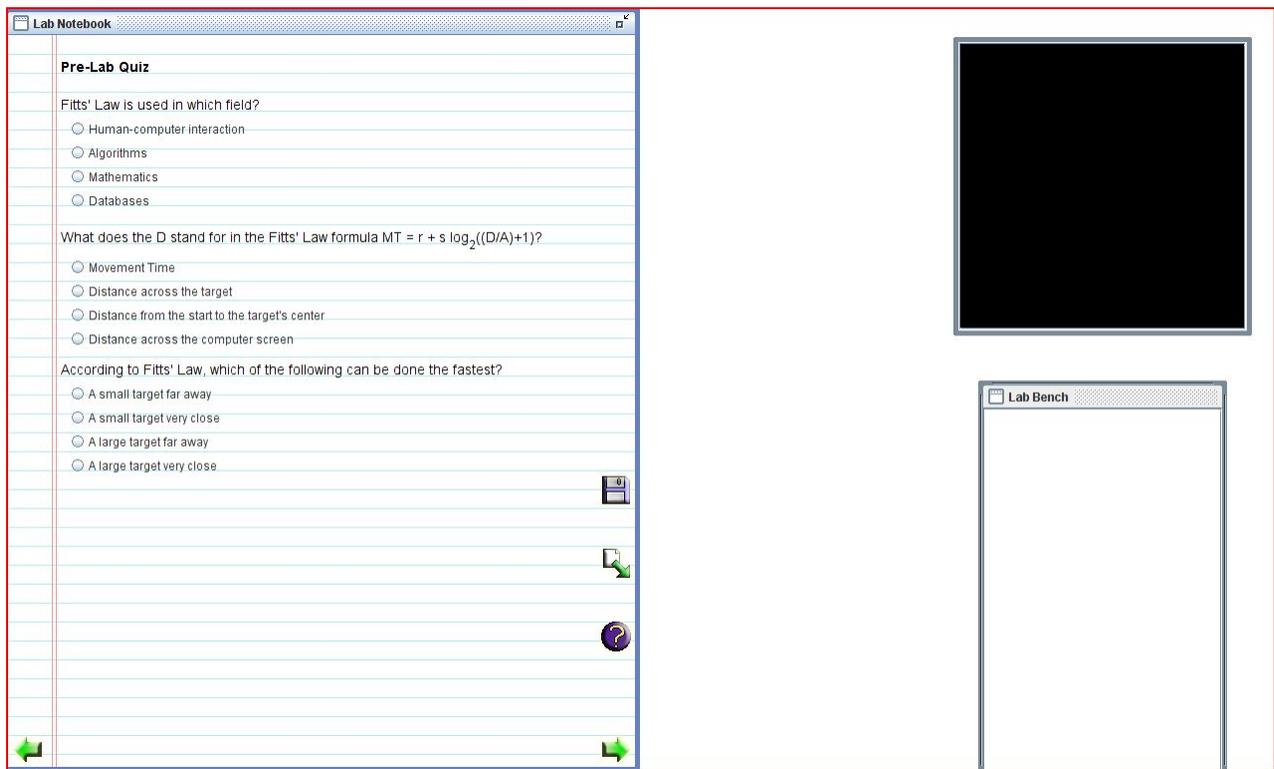


Figure 6. A set of quiz questions from the QUIKFITT lab, typical for LOCUS labs

## 6 Development Experience

Making something that would help teach people unfamiliar with computer science about Fitts' Law seemed simple enough, but the project turned out to be a bit more complicated than anticipated. While it was all an interesting learning experience, sometimes learning things was a painful and slow process. Learning and understanding the theory itself was my first experience in teaching myself a computer science concept, and it went smoother than I anticipated. There were plenty of papers and material which covered Fitts' Law, and the only real issue was that they were all written with a fair amount of complexity. This made it a little difficult for me to get a good enough grasp on the theory so that I could teach it to others, but after that everything fell into place.

The next major step of the project was making the QUIKFITT program itself. This took me the largest amount of time, and required learning more about a part of the Java programming language called Swing [2]. Unfortunately, Swing has many parts which are very tricky to use without a fair amount of experience and trial-and-error. Most of my time programming was spent learning how to use the new Swing concepts, testing and getting everything to work properly. The final part of this thesis was writing a lab. The labs for LOCUS are written in a language called XML, which I had to learn for this project, but it was not too difficult. Overall, it was simple and straightforward, a relaxing end to a complicated project.

## 7 Future Work

Due to time constraints (and the current limits of my skill), there are many things which I had envisioned QUIKFITT having but were not implemented. These are as follows:

- In the chemistry-themed activities, being able to drag the test tube and have it move with the mouse.
- In the chemistry-themed activities, having an animation occur when clicking on the beaker. Likely of the test tube pouring into it and some sort of animation representing a chemical reaction, such as a poof of smoke.
- In the desktop-themed activities, having the simulated toolbar at the bottom of the screen look like an actual toolbar.
- Thematically appropriate backgrounds for all the activities. This is more of an artistic problem than a programming problem, but still important. Good backgrounds would give QUIKFITT a more professional look and make it more appealing overall to the average user.

## References

- [1] "Ergalics: Field Guide." University of Arizona. <http://www.cs.arizona.edu/projects/focal/ergalics/fieldguide/> Viewed December 12, 2011
- [2] Loy, Marc, and Robert Eckstein. 2003. "Java Swing." Sebastopol, CA: O'Reilly.
- [3] MacKenzie, I.S., and Buxton, W. (1992). Extending Fitts' law to two dimensional tasks. Proceedings of the CHI '92 Conference on Human Factors in Computing Systems. New York: ACM.
- [4] Paul M. Fitts and James R. Peterson (1964). Information capacity of discrete motor responses. *Journal of Experimental Psychology*, 67(2):103-112, February 1964.
- [5] Paul M. Fitts (1954). The information capacity of the human motor system in controlling the amplitude of movement. *Journal of Experimental Psychology*, volume 47, number 6, June 1954, pp. 381-391. (Reprinted in *Journal of Experimental Psychology: General*, 121(3):262-269, 1992).