LoCuS (Laboratory fOr CompUter Science) System is instructional software for teaching computer science theories through experiments. Experiments within the context of LoCuS are laboratory excursions akin to those found in other sciences like biology, chemistry, and physics. A **locus** (Latin for “place”) is a “collection of points which share a property” or “the path through which a point moves to satisfy a given condition.” The LoCuS logo is a Boy’s surface, named after its founder, Werner Boy. Boy’s surface is a particularly intricate locus which is described by a system of complex equations. Just like the M"{o}bius strip, Boy’s surface has only a single side. Our goal is for LoCuS to define a new place to emphasize the science of computation as well as a new path for computer science, in the form of labs.
1 Introduction

The QuikFit apparatus will be used to teach Fitts Law, a fundamental theory in computer science, to high school, middle school, and college level students. When using a computer, the time it takes to click from one spot on the screen to another is determined by the distance between the two spots and the area of the second spot clicked on. This relationship is described by Fitts Law, and the equation is such:

\[ T = a + b \log_2(1 + \frac{D}{W}) \]

where \( T \) is time, \( D \) is the distance between the two areas being clicked, and \( W \) is the width of the second area being clicked. \( a \) and \( b \) are constants that vary with the user.

This apparatus will be provided values used within Fitts Law, namely the distance, width, and time variables. There will be three different views of QuikFit; all will have the same underlying theme of teaching Fitts Law, but each version will provide different distances and widths and be entertaining to the target audience.

2 Functionality

There are several different views which can be generated by this apparatus, each being distinct. Each view in Fitts Lab involves clicking on a source and then clicking on another object, measuring the distance between them and the time it takes to click both. The different views provide different scenarios for this clicking activity.

3 User Interface

There are multiple views of QuikFit, and we will examine each one in turn.

3.1 Circles

The first view involves two circles of varying radii that the student must click on. The objective of the activity will be to click within the Start circle followed by clicking within the End circle. This apparatus, taking up about half of the window in the user interface, will be able to send the variables of distance, width, and time to any other apparatus. The student must click on the Start circle first followed by the End circle for the activity to register the data. Clicking on the End circle first will be ignored. The timer will begin once the student has clicked on the Start circle so it will not matter where the mouse is at on the computer screen. After they have properly clicked Start and End, a new set of circles will appear. The student will click 10 sets of circles to get 10 sets of data points. The locations and widths of the circles will be generated randomly. Instructions for this activity will appear in the window if the student has clicked on the Help tab at the top left of the activity window. The About tab next to the Help tab will show the names of those that contributed to this project. The QuikFit tab will bring the user to the main activity window. The layout of the first view of QuikFit is shown in the drawing below.
3.2 Ice Cream

The second view of QuikFit will be presented as an Ice Cream Game. It will still record distances, widths, and times and will be able to send that data to other apparatuses. This game provides a different scenario and is used to compare to predicted data from the original option. The game layout is such:

- Different ice cream scoop flavors are on the left of the screen.
- At the bottom are kinds of toppings to put on the cones.
- On the left are the three different cone sizes, small, medium, and large (a bowl).
- At the top, ice cream orders will appear, having at least a cone and ice cream scoop, and sometimes containing a topping. They will have 30 seconds to complete each order, and the time remaining is shown.
- In the middle (on top of the store counter) are cone holders where the students will have to drag the ingredients to make their cone.
- In the bottom right corner the total points are displayed.
- In the bottom left corner the cones left to make are displayed.
- Before the game starts, a smaller screen will appear showing general instructions for the game.
- A Help tab at the top left of the game will show instructions when clicked on.
- An About tab at the top left of the game will show the people who contributed to the project.
- A QuikFit tab will bring the user to the main game window.

As the game starts, orders will start appearing at the top of the window detailing which type of ice cream cone to make. The student must make the cone in logical order, placing the cone first, then the ice cream scoop, then the toppings (this will be stated in the instructions). When the student has assembled the correct cone, the order is complete, and the cone disappears (the customer has to eat it, of course), and the proper points are awarded. The player gets 30 points if the cone is completed within 0-10 seconds, 20 points if it takes 11-20 seconds, and 10 points if it takes 21-30 seconds. Consequently, this view of QuikFit will have another connection that is able to send the total points scored. If the incorrect cone is made, they get 10 points deducted from their total, no matter how much time it took for them to create the cone. The student will create between 20 and 30 cones total.

When the user clicks a cone/ingredient, the cursor turns into that ingredient. Then, when they bring it over to the cone holders, the holder will be highlighted, so they know they are in the specified area to place the cone/ingredient. The area for the medium cone holder is bigger than the small cone holder, and the large cone holder is bigger than the medium one, which will change the width variable in Fitts.
Law. There will also be a myriad of cone/ingredient combinations so the user will get multiple distances for the distance variable.

3.3 Balloons

The final view of QuikFit is a Balloon popping game. It is similar to the second view in regards to having two connections: one that can send distances, widths, and times and one that can send the total score of the user. Once the Balloon Game has started, a window will appear with the layout shown in the drawing below. A green button will be present saying [Start] that will begin the game. This view also has the same tabs as the others but is not shown for redundancy reasons.

The game layout will be as follows:

• A 5x5 grid in the center of the screen to hold the different colored balloons.
• Different colored darts are in two columns, one to the left and one to the right of the grid.
• The darts will all be the same size.
• The balloons will vary in size, and in distance from their corresponding dart.
As the game starts, balloons will start appearing on the grid randomly, and the goal of the game is to pop balloons as they appear on the grid.

Balloons can only be popped by darts of the same color. So, the user will have to click a dart and then click a balloon of the same color in order to pop that balloon. When the balloon has been popped, it disappears and the proper points are awarded.

When the user clicks a dart, the cursor turns into that dart. Then, when they bring it over to the balloons, the balloon will be highlighted, so they know they are in the specified width to place the dart. The widths will vary by balloon, which will change the W variable in Fitts Law. The different position of each colored dart gives the user will get multiple distances for the D variable. The score will be tracked by total balloons popped, with one point being awarded per balloon. As the game progresses, balloons will appear faster and faster. The length of this game will vary depending on a parameter of time in minutes, which will be determined before the lab has started. When the grid is totally filled with balloons, the game is over.
4 Options

Options are specified in the lab and are sent to the apparatus. The following is an example of these options.

```
<options>
  <mode view = "IceCream" />  
  <gameLength seconds = "30" />  
  <action type = "Click" />
</options>
```

The lab chooses the view mode of the apparatus. This will be set by the `mode` element which has three view attributes: Circles, IceCream, and Balloons. The Circles, IceCream, and Balloons attributes represents the aforementioned Circles, Ice Cream, and Balloons respectively. The default view will be the Circles view.

The lab chooses how long the Ice Cream and Balloon game last. This will be set by the `gameLength` element which has one attribute seconds: it will be an integer representing the duration of the lab in seconds length. The Fitts mode will depend on how many clicks or drags are required, it will be completed after the last Click/drag in the time frame given. The Ice Cream game length will depend on how many orders are completed. The Balloon game length will depend solely on time; after that determined time balloons will appear at a rate where it is impossible to keep the board clear. The default time of the game is 30.

The lab states the action being performed, either clicking and dragging. This will be set by the `action` element which will have two type attributes: Click or Drag. The Click option requires the user to click the first position followed by clicking the second position. The Drag option requires the user to click and drag the first position to the seconded position. This would have to be uniform throughout all the views in order for the data sent to be consistent. The default will be Click.

5 Connections

Each view of this apparatus will be able to connect with other apparatuses used in the Fitts lab. One of those connections will send ClickData. It will have three parts to it to send three variables: distance, width, and time. Each time a student clicks on a circle, ice cream component, or balloon, the distance between the two objects, the width of the second object, and the time it took to click both objects will be sent.

The other connection will send the GameScore from the Ice Cream and Balloon games. As described previously, those two views keep track of a score to provide entertainment for the students. Their score for each game will be able to be sent to other apparatuses.
6 Networking

Each view of this apparatus will have the option of saving data while in use. Data must be sent to a server where it is then stored in a subdirectory of the server. This could be the machine the program is running on after the option starts a server. Later on, a small change to the system would allow data to be captured from all over on a central server. The data will be appended to a flat file of text where each line is the distance, width, and time.

7 Animation

Create animation and sound for the Ice Cream and Balloon games. There could be animation when a balloon is popped and a popping sound would play. Also, the ice cream scoops could fall into the cone and make a sound each time one is clicked. There will be an option to turn the sound on or off as well.

8 Extra Credit

For extra credit, these features can be implemented.

1. +5 Have the student explore what happens when the click over horizontal distances vs. diagonal distances, horizontal vs. vertical, etc. We would modify the layout of each view to only have locations of the two items clicked on be horizontal or vertical.

2. +3 Instead of randomly generating location in the views, we could have information be read from a data file that would have fixed locations and widths for the two objects being clicked on. When the views are started, all the locations that will be shown will already be known instead of being randomly generated.

3. +2 The borders of the windows where the views are shown could be made effectively infinite, so the mouse can’t escape. We would then place some of the objects that get clicked on the edges of the window, so the student can discover it is easier to click those objects.

4. +8 Another view that could be developed is one that is more realistic, i.e. in terms of computer GUIs. For example, the student would explore drop down menus, desktop icons, and the arrangement of buttons and see which designs are more beneficial than others.