CSCI-1302

Software Design
Spring 2014

Project 2

Fitts' Law

Learning Outcomes

1) Implement a graphical user interface using Swing
2) Implement event-handling in Java
3) Demonstrates comprehension of the tradeoffs involved in design choices when building a user interface design.

Points This project is worth 70 points

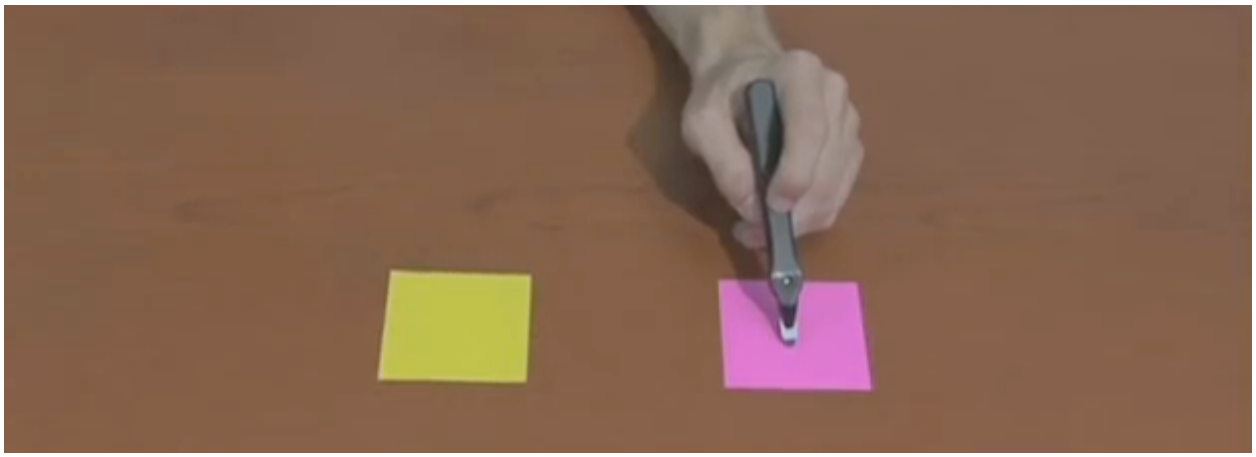
Due Date This project is due by 11 pm on Friday, March 21, 2014

Late Penalty Otherwise, 12% off the maximum original point value is deducted for each 24-hour period the assignment is late for up to two days late. Saturday/Sunday count as one day.

Collaboration Policy For this assignment, you may not discuss any code in person or in email with anyone other than the instructor or teaching assistants. You may, however, post questions on Piazza and help each other out as long as specific code isn't discussed.

The Java API, Java Tutorials, and the course textbooks are fair resources for this project. Any resources you used should be clearly documented in your readme.txt file.

The Science of User Interface Design



One of the main "laws" of modern graphical user interfaces is Fitts' Law. Fitts' law states that the amount of time it takes to move quickly to a target area is a function of the distance ot the target and the size of the target. Originally, Fitts' Law experiments were conducted using pencils where an experimenter would measure how quickly a person could touch the middle of targets of different sizes. Fitts' original work is available in the *Journal of Experimental Psychology,* volume 45, number 6, pages 381-391.

In particular: $MT = a + b \log_2(2A/W + c)$

Where MT = movement time, a and b are empirically determined constants, that are device dependent, c is a constant that's typically 0, 0.5, or 1 (see), and A is the distance of movement from start to the target's center, and W is the width of the target. Note, there are other formulations of this equation.

What does this mean for computer users? Well, think of how long it takes to move the pointer on a screen using a mouse or touchpad. The time it takes to move the pointer to a target depends on how big the target is and how far you are from it.

Your next project will involve designing a graphical user interface to create a Fitts' Law experiment! Think of this as creating a variation of the game "whack-a-mole" where moles are of various sizes.

Operation

- You will need to think about the flow and operation of your program. In particular, your writeup for this assignment will be a 2-page user manual explaining all the functionality of your system.
    - The operation of the experiment will be as follows: once your user completes any initial setup steps, you will present the user with a series of objects of different shapes. In particular, your system should operate in these steps:
        1. Display an initially blank field
        2. Give a countdown from 5 to 0
        3. Present the user with an on-screen target
        4. The user should move the pointer to click on the target. You should measure the amount of time it takes for the user to move the mouse and click to the display (see below)
        5. Once clicked, the target goes away and is immediately replaced by another target somewhere else of a different, random size
        6. Repeat steps 4 and 5 for as many trials you are measuring
- You need to decide how and when you should be starting the experiment. Think of different ways that you could prompt a user or various features you could put in the interface.
- You experiment should consist of at least 50 trials (but you may have more, if you desire).
- You should use the following Swing components in your UI where each component must accomplish something substantial. That is, you can't have one of these components in that doesn't have any real impact in how your program operates:
    - A slider
    - A button
    - A dialog box
- Your program needs to generate a "comma separated value" file (CSV) that contains information in the following format:

  Trial Number, Target Size (in pixels), Distance to Target (in pixels), Time To Click (in milliseconds)

  An example row might look like: 1,100,800,50
- Your writeup will also include a 1-page appendix in which you explain your design choices for the layout, placement, and operation of GUI components.
- Your writeup will also include a plot of the data that you gathered. Is there a visible relationship between time, distance to target, and size of target?

- Your writeup should be named lastname_writeup_p3.pdf and submitted along with your source code to Nike (where lastname is your lastname).
- You'll need to determine the threshold for sizes that you'll randomly present your targets. They should reasonably vary between smaller and larger, but not be as large that it takes up more than ½ of the field, or so small that you have someone trying to click on a few pixels.

Extra Credit

- Using a tabbed pane, create a second version of the program that's geared as a game for kids. This requires making the application that looks appealing for young children and "fun features" You'll need to submit another writeup called lastname_extra_credit_p3.pdf along with your source code that details your design decisions and features. Make sure you talk about how you are creating the "backstory" for the game play.

Input and Output Requirements

- Your driver class should be called `P3.java`
- You should also have a class for `FittsLaw.java`
- You may construct any helper classes you wish using appropriate naming conventions
- You should have an option File -> Save To where a user can indicate the directory in which they want to save the data collected (in csv format)
- Your program should also end/terminate only under the appropriate mechanism of someone choosing File -> Exit or using the keyboard accelerator
- There are no input arguments

Notes about GUI and Nike

If you are connecting remotely to Nike, you need to set up X11 forwarding using SSH in order to see the GUI program appear on your local machine. Here's a helpful link: http://tldp.org/HOWTO/XDMCP--HOWTO/ssh.html. If you are connecting via terminal on a Mac or Linux machine, simply use the –X flag when you are connecting to Nike. Alternately, you may opt to develop locally (i.e. using javac on your own Windows, Mac, or Linux computer).

Points

This project is worth 60 points towards your course grade. Grading of this programming project will use the following rubric:

Proper documentation	5 points
(pre & post statements, commenting conditionals, not excessive commenting)	
Writeup	10 points
As specified as above, with emphasis on your design justifications	
Passing Test Cases	20 points
Exceptional Condition Handling	15 points
Your program should never exit without being in control	
Design and Usability	20 points
	Total: 70 points

***Note**: This is our approximate grading distribution. Point values may vary.*

Submission Instructions

1. 1. Create a folder in an Nike account called **lastname_proj3** where lastname1and lastname2 is the last name of you.
   2. Copy all **thoroughly commented** Java source files in the folder created in step 1.
      3. Place a working makefile in the folder created in step 1 that has three directives:
         1. `compile:` compiles all of the source code
         2. `run:` runs an example of your program
         3. `clean:` removes all class files
      4. Add a `readme` file to the folder created in step 1 which has your name and clear instructions on how to compile and run your team's program.
      5. Remove all class files before submitting.
      6. Navigate to the parent directory of the folder created in step 1 on Nike, and issue the command below.

         ```
         submit  lastname_proj3  cs1302a
         ```

      7. If the submission was successful, then a file that begins with rec will be created in the submitted folder.