

## CS 3053

### Project – Window Design

*Due Tuesday 2018.03.06 at the beginning of class.*

#### **Overview**

In this assignment you will refine the layout of the main window of your application. The goal is to finalize the size and relative position of all components in the window. (In later assignments, you will refine the style and operation of all components to finalize the appearance and behavior of your application.) In class, we covered Fitt's Law and two frameworks for studying interaction. We had a group exercise to identify key user tasks, dissect a task into discrete interactions, and estimate how long the task might typically take overall. We used the result to speculate on ways to improve an existing window design. Out of class, you will use what you have learned to refine your window design in a systematic way.

As a group, you will identify the three most important tasks for your browser to support, prioritize them, predict their most likely interaction paths, sketch a updated widget layout to accommodate those paths, estimate total movement times using Fitt's Law, then analyze your wireframe layout to establish guidelines for your team's actual window layout. Individually, you will sketch another refined widget layout based on your team's guidelines, then implement a horizontal low-fidelity prototype that displays a structurally complete main window. *(Your will integrate your individual prototypes as a group as a part of the next assignment, so don't do that yet.)*

In this assignment, **all group tasks come before individual tasks**. Make sure to leave enough time for all of your team members to complete their individual tasks, especially implementation.

#### **Group Tasks**

Complete **all** group tasks before proceeding to individual tasks. Your entire team **must** participate.

**#1:** Review the slides from class on Fitt's Law and Analyzing Interactions. Discuss the general kinds of collection browsing and editing activities that you would like your application to support.

**#2:** Drawing from your personas and scenarios, brainstorm about the different tasks that you expect typical users to want to perform in the main window of your application. *Focus on tasks that involve: (1) navigating through the items in the collection; (2) filtering to see a subset of the collection; or (3) editing the characteristics of items in the collection.* Create a list of **two** tasks per team member, all clearly different. Prioritize them by utility and relevance to your theme.

**#3:** Briefly describe your top **three** tasks. Identify a sequence of appropriate interactive widgets for performing each task. In each description include a sentence that summarizes the sequence. *In other words, describe the purpose of the task followed by the mechanical steps to perform it.*

**#4:** Discuss how well your team's wireframe supports each task. Sketch a new design for the main window that includes all of the specific widgets needed to perform all three tasks. Your new sketch may include additional widgets and decoration as you determine appropriate. (Like before, only draw the interior of the window. Don't include a menubar. You may use lines and other decorations to visually group and label related sets of components. Annotate your sketch with arrows and text, including to indicate components referred to in your writeup in task #7.)

**#5:** Estimate the total movement time for the topmost task using Fitt's Law. Calculate the total as the sum of individual movement times between successive widgets in the interaction sequence. For movement to the first widget, choose a start point that is appropriate for the task.

**#6:** Using the individual and total movement times, analyze the efficiency of performing the task expected if your newly sketched layout were implemented. Develop and describe (as an outline) a set of flexible guidelines for sizing and positioning widgets in support of your top three tasks.

**#7:** Write up your work on each of the parts of tasks #1–#6. Compose this together. Be clear, objective, detailed, and thorough, yet succinct. *In grading we will be looking in particular for: thematically appropriate activities (#1); reasonably diverse tasks with sensible prioritization (#2); sufficiently thorough description (#3); substantial progress toward a final layout design (#4); correctness of Fitt's Law calculations including inputs (#5); useful, well-justified guidelines (#6).*

Your writeup should be between 1.5 and 2.0 single-spaced pages of writing, not including your Fitt's Law calculations. Use regular paragraphs and standard formatting (12 point font, 1 inch margins, etc.) Start the first page with a few lines stating your team number, name/logo, and list of member names. Attach a quality scan/photo of your sketch, followed by the details of your Fitt's Law calculations. Refer to the sketch and calculations in your writeup where appropriate.

To **turn in** your group work, go to the “Group - Window Design” assignment in Canvas to submit your results as a PDF. Only one team member needs to turn in the group component.

### ***Individual Tasks***

*All individual tasks must be completed entirely on your own.*

**#8:** Sketch a refined widget layout based on your team's guidelines from task #6. Draw widgets based on the **JComponent**s you've used in previous assignments. You may draw placeholders for any areas that display images of collection items. *Don't simply copy your team's sketch from task #4.* We'll look for clear evidence of thoughtful refinement in response to the guidelines. Turn your sketch into a PDF and put it in the **Results** subdirectory as **sketch.pdf**.

**#9:** Implement a new frame to display your structurally complete main window. Start from a copy of your team's integrated build from the previous assignment. Duplicate the `stage3 main()` class, call it **Stage4.java**, and uncomment the `createScript()` line for it in **build.gradle**. Whenever you build, the executable **stage4** should appear in **build/install/base/bin**.

You may *judiciously* create new classes and even packages; if you do, organize them inside the **edu.ou.cs.hci.stages** package. Document your code thoroughly and appropriately.

Reproduce your refined layout as a hierarchy of **JPanels** containing appropriate instances of the **JComponent** subclasses you've used in previous assignments. Make each widget active, populate it with appropriate default value(s), and implement basic event handlers that simply print the widget's name and current value(s) to the console.

When your app starts, show only your new frame. *Don't create or show any of the frames from earlier assignments.* When the user tries to exit by closing the frame, display a **JFileChooser** for saving to a **.txt** file. If they choose a file that doesn't already exist, create a **File**, write each widget's name and current value(s) to it in a readable format, close it, then exit the program. Use this feature to create a suitably representative **example.txt** file and put it in the **Results** subdirectory. Just before doing that, take the corresponding screenshot of your frame, trim it, turn it into a PDF, and put it in the **Results** subdirectory as **screenshot.pdf**.

**#10:** Document how your sketch reflects (or not) each of your team's guidelines. Do the same for the layout of your implemented prototype. One medium-length paragraph for each is enough. Write this documentation as a text file, and put it in the **Results** subdirectory as **design.txt**.

To **turn in** your individual work, *first test your project using Gradle on the command line to make sure it builds and runs as intended*. Run **gradle clean** to reduce the project size. Append your 4x4 to the **project** directory; mine would be **project-weav8417**. Zip the renamed directory. Submit your zip file to the “Individual - Window Design” assignment in Canvas.