

## CS 3053

### Project – Styled Application

*Due Friday 2018.04.06 at 11:00pm.*

#### **Overview**

In this assignment you will incorporate color, text, and precise layout into the design of your application. In class, we talked about the gestalt principles of design and the various ways that they inform the grouping and layout of widgets and their labels. We also talked about color physics, perception, cognition, and design factors. On Tuesday we will talk about designing with text. Out of class, you will use what you have learned about the gestalt principles to improve the grouping and labeling of the elements in your UI, and what you have learned about color to style your application to suit your project theme and improve how information is shown in it. You will specify an updated UI design, analyze your design choices, and refine your team's prototype.

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:** Decide which of your Menu Design implementations, or combination of them, to adopt for your entire team moving forward. Integrate the necessary code into your team's shared build. This will be the build that you all start from for the following group and individual tasks. Briefly summarize the progress on developing your integrated UI and who has contributed which parts.

**#2:** Review the slides from class on Gestalt Principles of Design, Color & Vision, and Text. (The final slides on Text will be available after class on Tuesday.) Discuss your application's browsing and editing functions in terms of: (1) the ways in which they are similar or different from each other, and how strongly; (2) how they can be organized into groups (and groups of groups, etc.); and (3) where each function falls along a spectrum from *general-purpose* to *theme-specific*.

**#3:** Discuss the theme of your application. Define an appropriate color scheme for the design of your main window. Your scheme must consist of between **two** and **five** colors. List them by thematic name and hexcode value; for example, OU crimson is #841617 ([www.ou.edu/content/brand/colors.html](http://www.ou.edu/content/brand/colors.html)). For each color, briefly explain what it represents and/or is meant to evoke thematically, and describe generally how and where you intend to use it in your design.

**#4:** Discuss how collection information is displayed in your application. Determine how you will use color—from your scheme **or otherwise**—to represent information about: (1) the collection as a whole, and (2) individual items in the collection. For each, identify which parts of the UI are involved, describe the information being shown, and explain how colors are used to convey that information or otherwise serve a functional purpose (see slides on “Designing with Color”).

**#5:** Explain how your chosen colors are not only relevant to the theme, but also reasonable to use in a UI for browsing and editing thematic items. Analyze how the colors—both individually and collectively—take into account the cultural, physiological, and environmental factors that you expect to pertain to your application in practice. Draw conclusions about how you expect usability—*predictability*, *visibility*, *consistency*, and *redundancy*—to be affected, first for the (1) and (2) aspects of the UI that you identified in #4, and then for the application as a whole.

**#6:** Discuss your target users and their typical circumstances of application use. Identify **three** ways to improve the text of your 10 custom menu items to make them more *familiar* and *efficient*

for **new** users. Your three changes must be clearly different in nature and apply to distinct menu items. Briefly describe each change in terms of how the text appears before and after. Explain how your changes improve the familiarity and efficiency of your menu design for new users, in terms of any major linguistic, cultural, physiological, and environmental factors that you expect to pertain to use of the application in practice.

**#7:** Discuss the layout of widgets and labels in the main window of your application, focusing on the following six gestalt principles of design: *proximity, similarity, closure, continuity, symmetry, and surround/space*. For each principle, analyze how well the widgets and labels in your layout follow the corresponding guideline (see slides). Describe a plan to improve the layout based on your analysis. Explain how your proposed changes follow the principles to more clearly group related components. For sake of easier implementation, limit changes to those you can make by reorganizing the components in your main `JFrame`, modifying `LayoutManagers` in `JPanels`, using `JLabels`, and putting `Borders` around `JPanels`. You **don't** need to redo everything!

**#8:** Create three images to attach to your writeup:

- from #1, a screenshot of your team's integrated UI;
- from #4, a well-annotated sketch of your UI with your proposed color choices; and
- from #7, a well-annotated sketch of your UI with your proposed layout and labeling choices.

**#9:** Write up your work on each of the parts of tasks #1–#7. Compose this together. Be clear, objective, detailed, and thorough, yet succinct. *In grading we will be looking in particular for: an honest report of progress and contribution (#1); thematically appropriate number and choices of colors (#3); well-reasoned justification of color uses (#4); constructive analysis and reasonable conclusions about usability (#5); reasonable text choices with helpful reflection about factors for new users (#6); useful layout analysis and reasonable plan of improvement (#7); readable, well-annotated, helpful sketches (#8).*

Your writeup should be between 2.0 and 3.0 single-spaced pages of writing. Start the first page with a few lines stating your team number, name/logo, and list of member names. Use regular paragraphs and standard formatting (12 point font, 1 inch margins, etc.) Attach scans/photos of your three images (one full page for each). Refer to the images in your writeup appropriately.

To **turn in** your group work, go to the “Group - Styled Application” 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.*

**#10:** Start from a copy of your team's integrated build from task #1. Duplicate the `stage5` `main()` class, call it `Stage6.java`, and modify `build.gradle` to have a new `createScript()` line. Whenever you build, the executable `stage6` should appear in `build/install/base/bin`.

**#11:** Read about using `LayoutManagers` and `Borders` to achieve precise, decorated layouts:

<https://docs.oracle.com/javase/tutorial/uiswing/layout/using.html>  
<https://docs.oracle.com/javase/tutorial/uiswing/components/border.html>

**#12:** Implement your team's proposed styling changes from #1–#9. When your app starts, show only your main frame. Have each component and menu item respond to interaction by printing out a simple message, just like in the previous two assignments. Have your Quit action simply exit the application (regardless of how it's triggered).

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.

Take a screenshot of your frame after performing some representative interactions. Trim it, turn it into a PDF, and put it in the **Results** subdirectory as **screenshot-with-style.pdf**.

**#13:** Document how well your version of the application implements your teams' proposed color, layout, and labeling choices. One short paragraph for each is enough. Write this documentation as a text file, and put it in the **Results** subdirectory as **design-with-style.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 - Styled Application" assignment in Canvas.