CS 3053
Homework: Prototype D
Due Wednesday 2019.04.24 at 11:59pm.

All homework assignments are individual efforts, and must be completed entirely on your own.

In this homework you will: (1) gain additional experience with prototyping complex UI layouts; (2) implement a horizontal prototype of your layout from the Design D assignment; (3) analyze the prototype in terms of the Gestalt Principles of Design, both on its own and as compared to your original design; and (4) analyze the prototype in terms of its how well it utilizes color, text, and icons to improve (or not!) the usability of the layout.

Reviewing Guidelines for Layout, Color, Text, and Icons

Review the slides from class on the following four topics. Focus on the following slides:

- Gestalt Principles of Design — all, especially the nine that list Principle+Corollary+Guideline
- Color & Vision — all, especially *Color: Guidelines* and *Designing with Color* (x3)
- Text & Reading — *Typography*, *Fonts*, and *Guidelines for Easier Reading in UIs*
- Icons & Semiotics — *Types of Icons* and *Using Icons in Design*

Choosing an Implementation Language/Library

Prototypes A, B, and C used Swing, JavaFX, and HTML, respectively. In this assignment, you can choose any one of the three to implement your prototype. I encourage you to choose the *least* familiar one, to broaden your experience and grow your development toolbox! (Keep in mind that you won't need very advanced features of whichever language/library you choose.) Regardless of your choice, *all code must be yours alone.* You may reuse code (yours or mine) from the earlier Prototype assignments. You may not use UI builder tools like the JavaFX Scene Builder or Adobe Dreamweaver. Organize your code cleanly and clearly, and document it well.

The Implementation Process

To make implementing easier, the download provides starter code for all three implementation choices. In the `edu.ou.cs.hci.assignment.prototyped` package, go into the `fx`, `swing`, or `web` directory. For JavaFX, focus on `Model.java`, `SettingsPane.java`, `View.java`, and `View.css`. For Swing, focus on `Model.java` and `View.java`. For HTML, focus on `page.html` and `style.css`. You may add files (including subdirectories) inside your chosen directory, so long as the additions are appropriate, clearly relevant, and neatly organized. *Take care to preserve all three directories. Don't rename the `Application.java` and `page.html` files in them. These files are the entry points we'll use to run your code when grading.*

Implementing the Settings Pane Specification as a Prototype

The Design D instructions included a specification of a panel for adjusting settings during text editing. In this assignment, you will implement your Design D mockup as a horizontal prototype. Write code to implement your design either as a single pane in a frame (in Swing or JavaFX) or a single page (in HTML). Use expected UI widgets wherever possible. All components should be responsive but not functional. For example, checkboxes should turn on and off and textfields should allow text entry, but not have any other effect. Organize your layout as a hierarchy of JPanels (in Swing), subclasses of Pane (in JavaFX), or <div>s (in HTML). Incorporate color, text, and icons to create a *potentially* finished appearance. *(Don't go to extremes on polishing!)*

Analyzing the Prototype

For this part, you will analyze the layout, color, text, and icons of your prototype. You will then write up your analysis as a brief, formal document. Your writeup should be about 500 words long (400 minimum, 750 maximum). Write regular prose paragraphs and use ordinary formatting (single spaced, 12 point font, 1 inch margins, etc.)

Layout: Consider the same two features that you chose to analyze in Design D. Analyze how well each feature in your prototype conveys the intended grouping of related settings to users in terms of relevant Gestalt Principles. Assess advantages and disadvantages, weigh tradeoffs, and compare the layout in your implemented prototype to the one in your mockup design.

Color: Consider the colors in your layout, including white/black/grays. Describe how the colors are intended to convey information or otherwise serve a functional purpose. Briefly discuss a common perceptual, cultural, or environmental factor that is likely to negatively affect usability in some significant way, and propose an improvement to the colors to mitigate that factor.

Text: Consider the text used throughout your layout. Discuss how the shape and meaning of one of the labels helps users to learn and remember its associated function. Discuss how one of the other labels could be improved in the same regard. Assess how well the layout as a whole applies specific typographic details to help convey intended groupings of settings to users.

Icons: Pick any setting that appears as a set of multiple buttons with icons. For any one of that set's icons, briefly discuss (1) how well it represents an *object*, *attribute*, *function*, or *state*; (2) how you intend users to interpret the icon to learn the button's purpose; and (3) a user who might have particular difficulty interpreting the icon. Briefly discuss how you could improve the icons **as a set** in terms of their *detail*, *color*, *shape*, *consistency*, *location*, and/or *juxtaposition*.

Turning It In

Turn in a complete, cleaned, renamed, zipped copy of your `PrototypeD` directory:

- Take a screenshot of your application window when it's in an interesting graphical state.
- Put the screenshot in the `Results` directory as `snapshot.png` or `snapshot.jpg`.
- Put the analysis writeup in the `Results` directory as `analysis.pdf`.
- Go into the `ou-cs-hci` directory.
  - Make sure it contains all of the code modifications and additions that you wish to submit.
  - Run `gradlew clean` to reduce the size of your build.
  - If you're using Eclipse, run `gradlew cleanEclipse` and delete the `bin` directory.
- Append your 4x4 to the `PrototypeD` directory; mine would be `PrototypeD-weav8417`.
- Zip your entire renamed `PrototypeD` directory.
- Submit your zip file to the Homework - Prototype D assignment in Canvas.

These steps will make your submissions smaller and neater, which speeds up grading a lot.

To score the assignment, we'll be looking at: how many elements in the UI specification appear as components in your prototype, how well the layout and styling of your prototype follow the design principles and guidelines covered in class, how clearly and constructively your analysis covers each of the discussion points, the overall cogency of analysis and quality of writing, and how clearly your code is organized and documented. The maximum score is 20 out of 20.