**CS 3053**
**Project – Evaluation & Wrap-Up**
*Due Monday 2018.04.23 at 11:00pm.*

## *Overview*

In this assignment you will wrap-up development of your app and evaluate its design. In class, we talked about usability testing, and held group exercises to start thinking about usability tests for your app designs. We also talked about how to use HTML and CSS as a means to structure and style more complex text documents including hypertext. Out of class, you will use what you have learned to add an about window to your individual app implementations. You will then plan and run a usability test together to assess your "final" product.

In this assignment, **all individual tasks come before group tasks.** Coordinate with your team to make sure you finish your individual tasks in time for your team to complete the group tasks.

## *Individual Tasks*

*All individual tasks except #2 must be completed entirely on your own.*

**#1:** Start with your individual build from the previous assignment. Duplicate the stage7 main() class, call it `Stage8.java`, and modify `build.gradle` to have a new `createScript()` line. Whenever you build, the executable `stage8` should appear in `build/install/base/bin`.

**#2:** Read about Hypertext Markup Language (HTML) and Cascading Style Sheets (CSS):
www.w3schools.com/html/ and www.w3schools.com/css
On HTML, read the topics from *Introduction* to *Images*, plus *Classes* and *File Paths*. On CSS, read from *Introduction* to *Links*. (Also read about tables/lists if you want to use them in task #4.)

I extended the starter code in the provided `Stage8.java` with an example of showing HTML in a `JEditorPane`. It uses the `Resources.getResource()` method to get the local URL of an example HTML file in the `about` subdirectory of the `resources` package. The file references a stylesheet (`style.css`) and a small image (`star.png`) in the same directory. Clicking a link in the page in `JEditorPane` triggers the `HelloPanel.hyperlinkUpdate()` method, which calls a new method in `Resources` that attempts to open the link in the user's web browser.

**#3:** Implement a new feature to display "About" information for your app. Create a `JWindow` that contains: (1) a `JEditorPane` to display an HTML file, and (2) a `JButton` to close the window. To open the window, add a suitable menu item somewhere prominent in your main menu bar. If reading the HTML fails for any reason, display a suitable message in a `JOptionPane` without showing the about window.

In your implementation, put any new classes inside the `edu.ou.cs.hci.stages` package. Document your new code thoroughly and appropriately. Include all needed HTML, CSS, and image files inside the `resources` package.

**#4:** Write a basic HTML file that displays the following (not necessarily in this order):

- the name and version of your application;
- a brief description of your app's purpose (a few sentences);
- a small image or icon that suitably conveys the theme of your application;
- a list of your team members with the roles they played;
- mentions of any other people, groups, or products you'd like to acknowledge;
- a statement that the app was created as a HCI project, with a link to the OU CS page; and
- links to any other web pages that users are likely to find particularly helpful and appropriate.

Apply a combination of inline, internal, and external CSS styling to layout and format your HTML in a manner that represents your app info in an appealing, readable, and useful way.

***Important:*** *Keep your HTML* ***simple****. Java supports only HTML 3.2 and a subset of CSS. See the* `javax.swing.text.html.CSS` *API documentation for a list of supported CSS properties.*

**#5:** Demonstrate that your app successfully displays "About" information. Open the new window and scroll (if necessary) to the end of your information page. Take a screenshot showing the about window, in front of your main frame if possible. Take a second screenshot showing how the same HTML appears in your web browser. Trim the screenshots, turn them into PDFs, and put them in `Results` as `about-appinfo.pdf` and `about-browser.pdf`, respectively.

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 - Evaluation" assignment in Canvas.

## *Group Tasks*

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

**#6:** Review the slides from class on Usability Testing. Discuss the components of usability test design as it relates to your project. To make this more fun and efficient, consider redoing the group exercises on the slides again, but with longer time limits and only your team members. Briefly summarize how you conducted discussion and the contributions each of you made to it.

**#7:** Based on your discussion, decide upon the following aspects of a usability test for your app:

- What is the **purpose** of the usability test?
- What are the **concerns and goals** that motivate the usability test?
- What THREE **tasks** will you have participants perform in the app?
- Within what **scenario** will you have participants perform those tasks?
- For each of the three tasks, what qualitative or quantitative **measurement** will you collect? *(Required: Two measurements must be quantitative, the other qualitative.)*
- Will your test be a **diagnostic**, a **comparison**, or a **validation**?
- What **method** will you use to collect the measurements?
- What kind of **location** will you use to conduct your usability test?
- Who will be your **participants**, **testers**, and **observers** (if any)?
- When will you **schedule** your test? How many sessions will you need for each participant? How long will each session take?
- What **scripts** and other **test materials** will you need to conduct your test?

Briefly describe and explain each of your choices. *(Note that the bullets in this list parallel the slides from class. The slides may be useful as a reference.)*

**#8:** Have each team member individually recruit **two** people to participate in your usability test. Participants can be anybody except your HCI classmates. Rotate through your individual app implementations from tasks #1–#5 above; that is, have one team member's recruits perform the test tasks on the ***next*** team member's implementation. Follow the guidelines and phases on the "Finally Ready" slide. It's fine to be informal so long as you respect the spirit of the process. Briefly describe your participants (without names) and report on how your tests actually went.

**#9:** Discuss your process and analyze your measurements to draw conclusions about the usability and utility of your app design for the test tasks, and overall. Cite specific usability and utility principles in your analysis. Draw conclusions about whether your test achieved its overall

purpose and specific goals. Based on your analysis, come up with three more tasks—two more quantitative and one more qualitative—to further investigate your app's design. Briefly explain why you think additional testing with those tasks would help you better understand your design.

**#10:** Discuss your individual implementations. Drawing from your usability test results, decide how to integrate your individual CSV and About Window implementations into your "final" project application. Make sure all code and resource files are well organized, appropriately named, and **thoroughly** documented. This will be the application you present to the class and describe in your project digest. Briefly summarize who contributed which code and resource components to your final build. ***Note:*** *Your final code will be submitted with the project digest.*

**#11:** Write up your work on each of the parts of tasks #6–#10. Compose this together. Be clear, objective, detailed, and thorough, yet succinct. *In grading we will be looking in particular for: evidence of good-faith participation in planning usability tests (#6); sufficient specification of a basic usability test with tasks appropriate to your application theme and its key functions (#7); sincere effort to complete at least a basic usability test, given very limited time (#8); objective analysis of test results and identification of promising follow-up tasks (#9); clear description of who contributed what (#10).*

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.) You may attach screenshots of your individual implementations and refer to them in your writeup as you think appropriate.

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