Heuristic Evaluation Lab Instructions

Each student will individually evaluate one prototype from another team. Teams will be matched such that all team members on one team will evaluate the prototype of another team.

Step 1: Teams: brief your evaluators by introducing your persona & system to them.

1. Describe your persona (especially goals & attributes)
2. Describe your functional + non-functional design requirements
3. Introduce your system to your evaluators, briefly overviewing its features
4. Evaluators: take detailed notes as you are listening to this briefing.

Step 2: Students individually conduct a heuristic evaluation of assigned prototype.

1. Use Schneiderman's 8 Golden Rules to evaluate the interface *(+ 1 of Nielsen's heuristics, see below)*
2. Create a heuristic evaluation report with the following:
    a. **Make a numbered list of at least 10 usability problems *and* successes** you find. For each problem and success:
        - Describe the problem or positive feature
        - Identify the relevant usability heuristics (from Schneiderman's 8 Golden Rules)
        - Discuss violation or conformance of the heuristic

    b. For problems:
        i. To estimate the severity of problems, ask:
            1. How common is problem: Does this issue happen in multiple aspects of the design?
            2. Will problem persist: Will users keep running into this issue?
            3. How easy for user to overcome: Is it a barrier to them doing what they need to do?
            4. How seriously will problem be perceived: A small annoyance or major disturbance?

        ii. Rate problems
            1. Cosmetic problem (only fix if extra time)
            2. Minor problem (low-priority fix)
            3. Major problem (important to fix, high-priority)
            4. Catastrophe (must fix)

        iii. Recommend solutions for the problems

You may use your notes and any course readings to assist you in your evaluation.

Step 3: Your final deliverable for this lab is your heuristic evaluation report. The report must be readable & easy to understand.

1. Don't bury the problems you found in reams of free-flowing prose: be concise, neat, organized
2. Where possible, include screenshots to illustrate your points
3. Write your name at the top of your report
4. At the end of class, email a PDF copy of your report to
    a. the appropriate team members and
    b. CC' Prof. Parker & Farnaz
    c. Subject must be [CS5340] Heuristic Evaluation

Source: https://www.cs.umd.edu/users/ben/goldenrules.html; http://www.nngroup.com/articles/ten-usability-heuristics/

Schneiderman's 8 Golden Rules

1. Strive for consistency. Consistent sequences of actions should be required in similar situations; identical terminology should be used in prompts, menus, and help screens; and consistent color, layout, capitalization, fonts, and so on should be employed throughout. Exceptions, such as required confirmation of the delete command or no echoing of passwords, should be comprehensible and limited in number.

2. Cater to universal usability. Recognize the needs of diverse users and design for plasticity, facilitating transformation of content. Novice to expert differences, age ranges, disabilities, and technological diversity each enrich the spectrum of requirements that guides design. Adding features for novices, such as explanations, and features for experts, such as shortcuts and faster pacing, can enrich the interface design and improve perceived system quality.

3. Offer informative feedback. For every user action, there should be system feedback. For frequent and minor actions, the response can be modest, whereas for infrequent and major actions, the response should be more substantial. Visual presentation of the objects of interest provides a convenient environment for showing changes explicitly.

4. Design dialogs to yield closure. Sequences of actions should be organized into groups with a beginning, middle, and end. Informative feedback at the completion of a group of actions gives operators the satisfaction of accomplishment, a sense of relief, a signal to drop contingency plans from their minds, and an indicator to prepare for the next group of actions. For example, e-commerce web sites move users from selecting products to the checkout, ending with a clear confirmation page that completes the transaction.

5. Prevent errors. As much as possible, design the system such that users cannot make serious errors; for example, gray out menu items that are not appropriate and do not allow alphabetic characters in numeric entry fields. If a user makes an error, the interface should detect the error and offer simple, constructive, and specific instructions for recovery. For example, users should not have to retype an entire name-address form if they enter an invalid zip code, but rather should be guided to repair only the faulty part. Erroneous actions should leave the system state unchanged, or the interface should give instructions about restoring the state.

6. Permit easy reversal of actions. As much as possible, actions should be reversible. This feature relieves anxiety, since the user knows that errors can be undone, and encourages exploration of unfamiliar options. The units of reversibility may be a single action, a data-entry task, or a complete group of actions, such as entry of a name-address block.

7. Support internal locus of control. Experienced users strongly desire the sense that they are in charge of the interface and that the interface responds to their actions. They don't want surprises or changes in familiar behavior, and they are annoyed by tedious data-entry sequences, difficulty in obtaining necessary information, and inability to produce their desired result.

8. Reduce short-term memory load. Humans' limited capacity for information processing in short-term memory (the rule of thumb is that we can remember "seven plus or minus two chunks" of information) requires that designers avoid interfaces in which users must remember information from one screen and then use that information on another screen. It means that cell phones should not require re-entry of phone numbers, web-site locations should remain visible, multiple-page displays should be consolidated, and sufficient training time should be allotted for complex sequences of actions.

**** And, one of Nielsen's Heuristics:** Aesthetics & Minimalist Design. Dialogues should not contain information which is irrelevant or rarely needed. Every extra unit of information in a dialogue competes with the relevant units of information and diminishes their relative visibility.

Source: https://www.cs.umd.edu/users/ben/goldenrules.html; http://www.nngroup.com/articles/ten-usability-heuristics/