Heuristic Evaluation (Individual)
Due: Start of Thu/Fri studio (Nov 9-10)—no credit if not turned in at start

Goals
Learn how to apply Nielsen’s adapted heuristics for evaluating and iterating on a user interface. Understand the tradeoffs compared to usability testing and other methods.

Assignment Overview
You have been hired as a design consultant to provide outside assistance to another team in your studio. Your CA will send you links to your assigned team’s relevant materials. You will:

1. Read through your assigned team’s Assignment 6 (medium-fi prototype) materials. This should help you get oriented on their project and prepare to evaluate their UI more critically. We recommend looking at their slides and README, and running through the prototype a couple times.

2. Conduct a heuristic evaluation of your assigned team’s user interface**. To do this, you will apply Nielsen’s heuristics and the 2 new heuristics we’ve added. Focus on giving feedback on what is currently implemented rather than pointing out missing features.

3. Produce an organized report of the problems you discovered in the interface. Organize the violations by task flow (e.g., all violations for task 1 grouped together). If the violation occurs across all tasks include these in an “All Tasks” section. If the violation occurs outside of a specific task, include this in a “Extra Violations” section. Please use the heuristics and numbering scheme from our lecture slides, also found at the end of this document. Further instructions below.

**If you’re evaluating a speech-based interface, these heuristics from this paper may be useful.

Report Instructions
Part I: Prototype Description
A one-sentence description of the project you are evaluating.

Part II: List of violations
Each violation in your list should be numbered sequentially and include the heuristic violated (number and title), the violation severity, the problem description, the rationale for why it violates that heuristic, and a recommendation to fix the problem. Use the ratings defined in lecture: 0 = not a problem, 1 = cosmetic, 2 = minor, 3 = major, 4 = UI catastrophe. Format the list of violations as follows:

[problem#]. [H#] [Heuristic Title] / Severity: [0-4]
Description:
Rationale:
Fix:

CS 147 Autumn 2023 website
https://hci.stanford.edu/courses/cs147/2023/au/
For example:

1. H4 Consistency & Standards / Severity: 3
   Task: Specify your dietary preferences
   Description: The interface used the string “Save” on the first screen for saving the user’s information, but used the string “Store” on the second screen.
   Rationale: Users may be confused by this inconsistent terminology for the same function.
   Fix: Use “Save” on all screens.

**Part III: Summary**

Give the total number of violations found using the table below. Double check your math.

<table>
<thead>
<tr>
<th>Category</th>
<th># Violations</th>
</tr>
</thead>
<tbody>
<tr>
<td>H1: Visibility of System Status</td>
<td></td>
</tr>
<tr>
<td>H2: Match b/w System &amp; World</td>
<td></td>
</tr>
<tr>
<td>H3: User Control &amp; Freedom</td>
<td></td>
</tr>
<tr>
<td>H4: Consistency &amp; Standards</td>
<td></td>
</tr>
<tr>
<td>H5: Error Prevention</td>
<td></td>
</tr>
<tr>
<td>H6: Recognition not Recall</td>
<td></td>
</tr>
<tr>
<td>H7: Flexibility &amp; Efficiency of Use</td>
<td></td>
</tr>
<tr>
<td>H8: Aesthetic &amp; Minimalist Design</td>
<td></td>
</tr>
<tr>
<td>H9: Help Users with Errors</td>
<td></td>
</tr>
<tr>
<td>H10: Help &amp; Documentation</td>
<td></td>
</tr>
<tr>
<td>H11: Accessible Design</td>
<td></td>
</tr>
<tr>
<td>H12. Value Alignment &amp; Inclusion</td>
<td></td>
</tr>
<tr>
<td><strong>Total Violations</strong></td>
<td></td>
</tr>
</tbody>
</table>

**Part IV: Overall Recommendations**

Close with 1-2 paragraphs covering general impressions, trends identified across the heuristic evaluation, and overall recommendations you have for improving the interface, including any problems you found that didn’t necessarily fit into the heuristics.

CS 147 Autumn 2023 website
https://hci.stanford.edu/courses/cs147/2023/au/
Deliverables
You will individually submit a link to your Google Doc report through this Google form. Make sure your CA can view the file and that the general access permissions are set so that anyone at Stanford University with the link can view the document.

1. **Individual Report**
   Please name your file [YourName]-[ProjectYouEvaluated]-HE.

Examples
**Note: this assignment spec has undergone some changes, but the quality of the work in this example still stands (we did not require severity ratings or organization by task last year).**

Example

Grading Criteria
You will be graded on coverage of the issues present in the current user interface design, clarity of your violation descriptions, and quality of your recommendations. Reports that focus excessively on features that are missing will be marked down.

Report (100 pts)
*Project Description (5)*
   ___ Accurately and succinctly describes the project

*Violations (65)*
   ___ Found a large percentage of the violations
   ___ Thorough coverage of violations present in each task
   ___ Heuristic variety in violations found; Non-repetitive violations
   ___ Found some less obvious violations in addition to the more obvious ones
   ___ Descriptions of violations clear and easy to understand
   ___ Rationale for the heuristic used is clear and valid
   ___ List is organized and violations are in the correct format (by task flow & numbered)

*Summary & Recommendations (30)*
   ___ Summary table is free from errors
   ___ General impressions and trends noticed across violations found
   ___ Includes feedback and recommendations that don't fit into the violations

CS 147 Autumn 2023 website
https://hci.stanford.edu/courses/cs147/2023/au/
12 Usability Heuristics

Adapted from Nielsen (2nd version). These are 12 general principles for user interface design. They are called “heuristics” because they are more in the nature of rules of thumb than specific usability guidelines.

H1. Visibility of system status
The system should always keep users informed about what is going on, through appropriate feedback within reasonable time.

H2. Match between system and the real world
The system should speak the users' language, with words, phrases and concepts familiar to the user, rather than system-oriented terms. Follow real-world conventions, making information appear in a natural and logical order.

H3. User control and freedom
Users often choose system functions by mistake and will need a clearly marked "emergency exit" to leave the unwanted state without having to go through an extended dialogue. Support undo and redo.

H4. Consistency and standards
Users should not have to wonder whether different words, situations, or actions mean the same thing. Follow platform conventions.

H5. Error prevention
Even better than good error messages is a careful design which prevents a problem from occurring in the first place. Either eliminate error-prone conditions or check for them and present users with a confirmation option before they commit to the action.

H6. Recognition rather than recall
Minimize the user's memory load by making objects, actions, and options visible. The user should not have to remember information from one part of the dialogue to another. Instructions for use of the system should be visible or easily retrievable when appropriate.

H7. Flexibility and efficiency of use
Accelerators -- unseen by the novice user -- may often speed up the interaction for the expert user such that the system can cater to both inexperienced and experienced users. Allow users to tailor frequent actions.

H8. Aesthetic and 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.

CS 147 Autumn 2023 website
https://hci.stanford.edu/courses/cs147/2023/au/
H9. Help users recognize, diagnose, and recover from errors
Error messages should be expressed in plain language (no codes), precisely indicate the problem, and constructively suggest a solution.

H10. Help and documentation
Even though it is better if the system can be used without documentation, it may be necessary to provide help and documentation. Any such information should be easy to search, focused on the user’s task, list concrete steps to be carried out, and not be too large.

H11. Accessible design
Users can interact with the system using alternative input methods. Content is legible with distinguishable contrast and text size. Key information is upfront and not nested for screen readers. Purely visual or auditory content has text-based alternatives for users with low vision and low hearing.

H12. Value alignment and inclusion
The design should encode values that users can understand and relate to. It should make a diverse group of users feel included and respected. The design should prevent the reproduction of preexisting inequities and not create additional burdens for disadvantaged populations.