HEURISTIC EVALUATION
Presentations!!!

• Great job! More on Thursday...
Today

• Introduction to evaluation
• Usability testing
• Heuristic evaluation
• Exercise
Today

• Introduction to evaluation
• Usability testing
• Heuristic evaluation
• Exercise

you will do this yourselves; one by Friday
Today

• Introduction to evaluation
• Usability testing \textit{you will do this yourselves; one by Friday}
• Heuristic evaluation \textit{we will do this today in class}
• Exercise
Evaluation - Goal
Evaluation - Goal

• Does it work?
Evaluation - Goal

• Does it work?
• Does it help the user?
Evaluation - Goal

• Does it work?
• Does it help the user?
• Is it easy to use?
Evaluation - Goal

• Does it work?
• Does it help the user?
• Is it easy to use?
• How does it fail?
Evaluation - Goal

• Does it work?
• Does it help the user?
• Is it easy to use?
• How does it fail?
• Where does the user get stuck?
Evaluation - Goal

- Does it work?
- Does it help the user?
- Is it easy to use?
- How does it fail?
- Where does the user get stuck?
- How can we fix it?
Evaluation - Goal

• Does it work?
• Does it help the user?
• Is it easy to use?
• How does it fail?
• Where does the user get stuck?
• How can we fix it?
• Does the user like it?
Evaluation - Goal

• Does it work?
• Does it help the user?
• Is it easy to use?
• How does it fail?
• Where does the user get stuck?
• How can we fix it?
• Does the user like it?
• **Will it sell?**
Evaluation - Goal

• Does it work?
• Does it help the user?
• Is it easy to use? (usability)
• How does it fail?
• Where does the user get stuck?
• How can we fix it?
• Does the user like it?
• Will it sell?
Evaluation - Goal

• Does it work?
• Does it help the user?
• Is it easy to use? \hspace{1cm} \textit{usability}
• How does it fail?
• Where does the user get stuck?
• How can we fix it?
• Does the user like it? \hspace{1cm} \textit{user experience}
• Will it sell?
Evaluation - Dimensions

• Formative
  – assess a system being designed (lo-fi prototype)
  – gather input to inform design

• Summative
  – assess an existing system (hi-fi prototype)
  – judge if it meets some criteria
Evaluation - Dimensions

• Observational
  – What works? What doesn’t work?

• Comparative
  – Which works better?
  – Between-groups, within-groups
Evaluation - Dimensions

• Types of data
  – process data
    • observations of what users are doing & thinking
  – summary, statistical, or bottom-line data
    • summary of what happened (time, errors, success)

  *might not tell you where the problems are*
Evaluation - Dimensions

• Quantitative
  – Indicate results with numbers

• Qualitative
  – Indicate results with words

form of data being obtained
Evaluation - Dimensions

• Objective
  – Information independent of person reporting it

• Subjective
  – Opinions that depend on person reporting it
Evaluation techniques

get information from the user

ask them

Interviews
Questionnaires

observe them

Ethnography
Passive observation
Think-aloud
Empirical user studies

make them observe themselves

Experience sampling
Diaries/logs
Inspection

get information from the user
get information from the user

Heuristic evaluation
Cognitive walkthrough
Action analysis
Evaluation - Dimensions

• End-user testing
  – Tester is a representative of the target user

• Expert evaluation
  – Tester is a UI/UX expert
Deciding on type of evaluation

• Depends...
  – what stage you are in the design
  – what your goal is
  – what resources you have
USABILITY TESTING
Usability testing w/ paper prototypes
Getting ready

• Make the prototype
• Recruit participants and schedule test
• Prepare the setup
• Prepare scenarios and tasks
• Prepare a script and checklist
• Decide what to measure
• Practice or pilot test
Make the prototype
Recruit participants

• Flyers
• Mailing lists
• Facebook
• Word-of-mouth

Human-Robot Interaction Study:
Help Simon ask good questions

We are conducting a research study that investigates how humans ask questions while learning tasks and skills. We aim to apply our findings on our humanoid robot Simon, so he can learn tasks/skills efficiently from humans by asking questions. The study involves watching task demonstrations, asking questions, and reproducing tasks. It takes about 30 minutes. Participants will receive $5 compensation in cash. You will also get the opportunity to have a picture taken with Simon. Sign up on our webpage.

Maya Cakmak & Andrea L Thomaz

http://www.simontherobot.com/ask
Place: CCB 2nd floor; RIM Center Undergrad Lab
Dates: August 24 - Sept 2, 2011
Prepare the setup

- **Computer**: "Computer" with components laid out in order, for quick access
- **Facilitator**: Facilitator, guiding user through tasks, prompting for user's thoughts
- **Observer**: Observer taking notes on index cards
- **User**: User, with lo-fi prototype in use
Prepare the setup

- Table/seats
- Materials
- Recording devices
- Note taking
Prepare tasks

• Setting up the context (scenario)
• What is the participant asked to do
Prepare tasks

• Bad:
  – Artificial subgoals
  – Artificial ordering
  – Giving the answers

• Good
  – Giving context
Prepare script, instructions, checklists

Preparation
• Make sure you know participant ID number
• Make sure you know participant’s condition
• Make sure you have all the instructional material (user manual, tutorial, video ready to play)
• Check that the microphone has batteries
• Check that all the props are there
• Prepare the video camera
• Move PR2’s arms to the neutral pose
• Make sure there is a printed consent form

Introduction
Thank you for agreeing to participate in our study. This is our robot PR2 (which stands for Personal Robot 2). The goal of our research is to allow end-users of robots like PR2 to be able to program it by demonstrating what they want. Today we will ask you to program several skills on PR2. This involves using speech commands and physically interacting with PR2 to move its arms. At the end, we will ask you to fill in a questionnaire regarding your interaction.

Consent form
Before moving onto the details, please take a look at these forms and sign when you are ready. Let me know if you have any questions.
• Make sure kill switch is ON
• Start the program

Explain the study
Today we are interested in evaluating the design of our instructional materials for using the robot. We are not evaluating you.

...
Decide what to measure

PARTICIPANT # ____

### PART 1

Please rate Simon's questions in terms of **informativeness** for the robot.

<table>
<thead>
<tr>
<th>Q1</th>
<th>Q2</th>
<th>Very informative</th>
<th>Not informative</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>2</td>
<td>3</td>
<td>4</td>
</tr>
</tbody>
</table>

Please rate Simon's questions in terms of **unexpectedness**.

<table>
<thead>
<tr>
<th>Q1</th>
<th>Q2</th>
<th>Unexpected</th>
<th>Predictable</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>2</td>
<td>3</td>
<td>4</td>
</tr>
</tbody>
</table>

Please rate Simon's questions in terms of **ease of answering**.

<table>
<thead>
<tr>
<th>Q1</th>
<th>Q2</th>
<th>Easy to answer</th>
<th>Difficult to answer</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>2</td>
<td>3</td>
<td>4</td>
</tr>
</tbody>
</table>

### PART 2

Please rate Simon's questions in terms of **informativeness** for the robot.

<table>
<thead>
<tr>
<th>Q1</th>
<th>Q2</th>
<th>Very informative</th>
<th>Not informative</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>2</td>
<td>3</td>
<td>4</td>
</tr>
</tbody>
</table>

Please rate Simon's questions in terms of **unexpectedness**.

<table>
<thead>
<tr>
<th>Q1</th>
<th>Q2</th>
<th>Unexpected</th>
<th>Predictable</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>2</td>
<td>3</td>
<td>4</td>
</tr>
</tbody>
</table>

Please rate Simon's questions in terms of **ease of answering**.

<table>
<thead>
<tr>
<th>Q1</th>
<th>Q2</th>
<th>Easy to answer</th>
<th>Difficult to answer</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>2</td>
<td>3</td>
<td>4</td>
</tr>
</tbody>
</table>
Practice!

• and fix:
  – the prototype
  – the setup
  – the tasks (scenarios)
  – the script and checklist
  – the measures
Ethical considerations

• Sometimes tests can be distressing
  – users have left in tears

• You have a responsibility to alleviate
  – make voluntary with informed consent
  – avoid pressure to participate
  – let them know they can stop at any time
  – stress that you are testing the system, not them
  – make collected data as anonymous as possible

• Often must get participant approval
HEURISTIC EVALUATION
Heuristic evaluation

• Goal
  – find **usability problems** in design
  – judge compliance with design principles/heuristics

• Process
  – small set of evaluators examine the interface
  – aggregate findings and summarize results

• Different evaluators will find different problems

• Can perform on working UI or paper prototype
Why multiple evaluators?

- Different evaluators will find different problems
Number of evaluators
Number of evaluators

- 3 to 5
Setup

• Two alternatives
  – **Evaluator alone**, evaluator writes a report at the end
  – **Evaluators and observer**, evaluator talks observer takes notes
Process
Process

• Pre-evaluation training
  – give evaluators domain knowledge & information on the scenario
Process

• Pre-evaluation training
  – give evaluators domain knowledge & information on the scenario

• Evaluation
  – individuals evaluates UI & makes list of problems
Process

• Pre-evaluation training
  – give evaluators domain knowledge & information on the scenario

• Evaluation
  – individuals evaluates UI & makes list of problems

• Severity rating
  – determine how severe each problem is
Process

• Pre-evaluation training
  – give evaluators domain knowledge & information on the scenario

• Evaluation
  – individuals evaluates UI & makes list of problems

• Severity rating
  – determine how severe each problem is

• Aggregation
  – group meets & aggregates problems (w/ ratings)
Process

• Pre-evaluation training
  – give evaluators domain knowledge & information on the scenario

• Evaluation
  – individuals evaluates UI & makes list of problems

• Severity rating
  – determine how severe each problem is

• Aggregation
  – group meets & aggregates problems (w/ ratings)

• Debriefing
  – discuss the outcome with design team
How to perform heuristic evaluation
How to perform heuristic evaluation

• Go through interface twice
  – First pass: get a feel for the flow and general scope
  – Second pass: focus on specific interface elements
How to perform heuristic evaluation

• Go through interface twice
  – First pass: get a feel for the flow and general scope
  – Second pass: focus on specific interface elements

• If system is walk-up-and-use or evaluators are domain experts, no assistance needed
  – otherwise might supply evaluators with scenarios
How to perform heuristic evaluation

• Go through interface twice
  – First pass: get a feel for the flow and general scope
  – Second pass: focus on specific interface elements

• If system is walk-up-and-use or evaluators are domain experts, no assistance needed
  – otherwise might supply evaluators with scenarios

• Explain each problem with reference to heuristics
  – Don’t simply say that you don’t like it
  – Same interface element can have multiple problems
Nielsen’s heuristics - Version 1

- H1-1: Simple & natural dialog
- H1-2: Speak the users’ language
- H1-3: Minimize users’ memory load
- H1-4: Consistency
- H1-5: Feedback
- H1-6: Clearly marked exits
- H1-7: Shortcuts
- H1-8: Precise & constructive error messages
- H1-9: Prevent errors
- H1-10: Help and documentation
Nielsen’s heuristics - Version 2

• H2-1: Visibility of system status
  – The system should always keep users informed about what is going on, through appropriate feedback within reasonable time.
Nielsen’s heuristics - Version 2

• H2-1: Visibility of system status

![Windows Defender scan interface](image_url)
Nielsen’s heuristics - Version 2

• H2-2: Match between system & 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.
Nielsen’s heuristics - Version 2

• H2-2: Match between system & real world
  – Bad example: Mac desktop
  • Dragging disk to trash should delete it, not eject it
Nielsen’s heuristics - Version 2

• H2-3: User control & 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.
Nielsen’s heuristics - Version 2

• H2-4: Consistency & standards
  – Users should not have to wonder whether different words, situations, or actions mean the same thing. Follow platform conventions.
Nielsen’s heuristics - Version 2

• H2-4: Consistency & standards
Nielsen’s heuristics - Version 2
Nielsen’s heuristics - Version 2

• H2-5: 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.
Nielsen’s heuristics - Version 2

• H2-5: Error prevention

```bash
% rm -rf *
%
```
Nielsen’s heuristics - Version 2

• H2-6: 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 whenever appropriate.
Nielsen’s heuristics - Version 2

• H2-7: 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 (e.g. macros).
Nielsen’s heuristics - Version 2

- H2-7: Flexibility and efficiency of use
Nielsen’s heuristics - Version 2

• H2-8: Aesthetic & 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.
Nielsen’s heuristics - Version 2

• H2-8: Aesthetic & minimalist design
Nielsen’s heuristics - Version 2

• H2-9: Help users recognize, diagnose, & recover from errors
  –Error messages should be expressed in plain language (no codes), precisely indicate the problem, and constructively suggest a solution.
Nielsen’s heuristics - Version 2

• H2-9: Help users recognize, diagnose, & recover from errors
Nielsen’s heuristics - Version 2

Good error messages:
– Clearly indicate something’s wrong
– Be human readable
– Be polite
– Describe the problem
– Explain how to fix it
– Be highly noticeable

Your PC ran into a problem and needs to restart. We’re just collecting some error info, and then we’ll restart for you. (0% complete)

If you’d like to know more, you can search online later for this error: HAL_INITIALIZATION_FAILED
Nielsen’s heuristics - Version 2

• H2-10: 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.
Additional heuristics

• General heuristics
  – Nielsen
  – Tog

• Category specific heuristics
  – Mobile heuristics

<table>
<thead>
<tr>
<th>Mobile Heuristic</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Heuristic 1</td>
<td>Visibility of system status and losability/findability of the mobile device</td>
</tr>
<tr>
<td>Heuristic 2</td>
<td>Match between system and the real world</td>
</tr>
<tr>
<td>Heuristic 3</td>
<td>Consistency and mapping</td>
</tr>
<tr>
<td>Heuristic 4</td>
<td>Good ergonomics and minimalist design</td>
</tr>
<tr>
<td>Heuristic 5</td>
<td>Ease of input, screen readability and glancability</td>
</tr>
<tr>
<td>Heuristic 6</td>
<td>Flexibility, efficiency of use and personalization</td>
</tr>
<tr>
<td>Heuristic 7</td>
<td>Aesthetic, privacy and social conventions</td>
</tr>
<tr>
<td>Heuristic 8</td>
<td>Realistic error management</td>
</tr>
</tbody>
</table>
Tog’s heuristics

- Anticipation
- Autonomy
- Color Blindness
- Consistency
- Defaults
- Efficiency of the User
- Explorable Interfaces
- Fitts' Law

- Human Interface Objects
- Latency Reduction
- Learnability
- Use of Metaphors
- Protect Users' Work
- Readability
- Track State
- Visible Navigation
Where to look for problems?

– single location in UI
– two or more locations that need to be compared
– problem with overall structure of UI
– something that is missing

• common problem with paper prototypes; it is okay if things have not yet been implemented; don’t focus on those.
Severity ratings

• Used to allocate resources to fix problems
• Estimates of need for more usability efforts
• Combination of
  – frequency
  – impact
  – persistence (one time or repeating)
• Should be calculated after all evaluations are done
• Should be done independently by all judges
Severity ratings

• 0 - don’t agree that this is a usability problem
• 1 - cosmetic problem
• 2 - minor usability problem
• 3 - major usability problem; important to fix
• 4 - usability catastrophe; imperative to fix
Example: How to report problems

• Can’t copy info from one window to another
  – violates “Minimize the users’ memory load” (H1-3)
  – fix: allow copying

• Typography uses different fonts in 3 dialog boxes
  – violates “Consistency and standards” (H2-4)
  – slows users down
  – probably wouldn’t be found by user testing
  – fix: pick a single format for entire interface
Example: How to report problems

• [H1-4 Consistency] [Severity 3] [Fix 0]
  – The interface used the string "Save" on the first screen for saving the user's file, but used the string "Write file" on the second screen. Users may be confused by this different terminology for the same function.
Class exercise

• Heuristic evaluation of paper prototypes