Usability
Based on material by Michael Ernst, University of Washington

Announcements

- CHECK YOUR GRADES
- Quiz 1-9, HW 1-6, Exam 1-2 in Rainbow Grades
- Feedback on Homework in Submitty
  - We’ll have HW7 and HW8 by the end of the week
- Check if excused late days are properly accounted for. Check if any notes on homework
- HW7 RESUBMIT & HW9 due December 12th
- A GUI Interface for your path finding algorithm

Outline of Today’s Class

- Usability
- Iterative Design
  - Design
    - Design principles
  - Implement
  - Low-fidelity prototypes
  - Evaluate
    - User testing

User Interface Hall of Shame

Fall 17 CSCI 2600, A Milanova

Better
You cannot directly edit the Date and Time fields from the keyboard. If you want to edit Time, you must click on “Set Time”, which triggers the “Clock” control, where you have to use the mouse to move the Minute and Hour hands.

- Mode
- Mode errors
- Unrecoverable errors
- Safety

- Designing User Interfaces Is Hard
  - You are not the user
  - Most software engineering is about communicating with programmers
    - Who are a lot like us
  - UI is about communicating with users
    - Users are NOT like us
  - The user is ALWAYS right
    - Usability problems are the design’s fault
    - Hard lesson to learn: if the user consistently gets stuck, this is not because the user is dumb, but because the interface is poorly designed

- Designing User Interfaces Is Hard
  - … Unfortunately, user is not always right
  - User cannot predict what they really want
  - 1950’s experiment with telephone handsets
    - Users thought weight was fine
    - Actually, they really wanted half the weight
  - # of results displayed for a Google search query
    - Users say they want 30
    - Actually, usage drops 20% relative to 10
Iterative Design

- UI development is an iterative process
- Iterations can be costly
- If the design turns out to be bad, you may have to throw away most of your code

Spiral Model

- Use throw-away prototypes and cheap evaluation for early iterations

Usability

- Usability: how well users can use the system’s functionality
- Dimensions of usability
  - Learnability: is it easy to learn?
  - Efficiency: once learned, is it fast to use?
  - Safety: are errors few and recoverable?
- Memorability: is it easy to remember what you learned?
- Satisfaction: is it enjoyable to use?

Dimensions of Usability

- Learnability
- Efficiency
- Safety
- Simplicity (not a usability dimension)
- Different dimensions vary in importance
  - Depends on the user
  - Depends on the task
- Usability is only one aspect of the system

Learnability

- IBM’s Real CD

- IBM’s Real CD, Source: Interface Hall of Shame
Learnability
LMS Create Weighted Column

Facts About Memory & Learning

- Working memory
  - Small: $7 \pm 2$ “chunks”
  - Short-lived: gone in $\sim 10$ seconds
  - Maintenance rehearsal is required to keep it from decaying but costs attention

- Long-term memory
  - Practically infinite in size and duration
  - Elaborative rehearsal transfer chunks to long-term memory

Design Principles for Learnability

- Consistency
  - Similar things look similar, different things different
  - Terminology, location, ...
  - Internal, external, metaphorical design

- Use common, simple words, not tech jargon!

- Recognition, not recall
  - Labeled buttons are better than commands
  - Combo boxes are better than text boxes

Source: Interface Hall of Shame

Tech Jargon
UHLS Catalog Advanced Search
Google Back in 2003

Visibility

Facts About Human Perception

Design Principles for Visibility

Facts About Motor Processing
Pointing Tasks: Fitts’s Law

- How long does it take to move your hand to a target of size $S$ at distance $D$ away?

  - E.g. moving mouse to target on screen

  \[ T = RT + MT = a + b \log \left( \frac{D}{S} \right) \]

  - $\log \left( \frac{D}{S} \right)$ is the index of difficulty of the pointing task

Derivation of Fitts’s Law

- Moving your hand is closed-loop control
- Each cycle covers remaining distance $d$ with error $\epsilon$
- After two cycles, within $\epsilon^2 D$ of target

Path Steering Tasks: Steering Law

- Fitts’s Law applies only if path to target is unconstrained
- But the task is much harder if path is constrained to a tunnel

  \[ \text{Steering Law: } T = RT + MT = a + b \left( \frac{D}{S} \right) \]

Design Principles for Efficiency

- Fitts’s Law and Steering Law
- Make important targets big, nearby, or at screen edges
- Avoid steering tasks!
- Provide shortcuts
  - Keyboard accelerators
  - Bookmarks
  - History

Usability Dimensions

- Learnability
- Efficiency
- Safety
- Simplicity
Mode Errors

- Modes: states in which actions have different meanings
  - E.g., vi’s insert mode vs. command mode
- Avoiding mode errors
  - Eliminate modes entirely
  - Visibility of mode
  - Disjoint action sets in different modes

Avoiding mode errors

- Eliminate modes entirely
- Visibility of mode
- Disjoint action sets in different modes

Confirmation Dialogs

Confirmation Dialogs: Deleting files in the LMS file system

I selected the files and clicked Delete

Not done! Confirmation dialog pops up. Clicked OK

Confirmation Dialogs: Deleting files in the LMS file system

Still not done! Clicked Submit and finally done! Another issue with file system: behavior inconsistent with other systems.
Design Principles for Error Handling (Safety)

- Use confirmation dialogs sparingly
- Prevent errors as much as possible
  - Selection is better than typing
  - Avoid mode errors
  - Disable illegal commands
  - Separate risky command from common ones
- Support Undo

Good error messages
- Precise
- Speak the user’s language
- Constructive help
- Be polite

Simplicity

Source: Interface Hall of Shame

Simplicity, Google Back in 2003

Source: Alex Papadimoulis

Simplicity, Google Now

Source: Google
Design Principles for Simplicity

- “Less is More!”
  - Omit extraneous information, graphics & features
- Good graphic design
  - Few well-chosen colors and fonts
  - Group with whitespace
- Use concise language
  - Choose labels carefully

Document Your System

- Write the user manual
  - Program and UI metaphors
  - Key functionality
  - Do not include: exhaustive list of all menus
- What is hard to do?
  - Who is your target audience?
    - Power users need a manual
    - Casual users might not
  - Piecemeal online help is no substitute

Outline of Today’s Class

- Usability
- Iterative Design
  - Design
    - Design principles
  - Implement
    - Low-fidelity prototypes
  - Evaluate
    - User testing

Low-fidelity Prototype

- Paper is a very fast and effective prototyping tool
  - Sketch windows, menus, dialogs, widgets
  - Crank out lots of designs and evaluate them
- Hand-sketching is OK --- even preferable
  - Focus on behavior & interactions, not fonts & colors
  - Similar to design of your ADTs and classes
- Paper prototypes can even be executed!
  - Use pieces to represent windows, dialogs, menus
  - Simulate computer’s responses by moving pieces around and writing on them

User Testing

- Start with a prototype
- Write up a few representative tasks
  - Short but non-trivial
    - E.g., “add this meeting to calendar”
    - E.g., “type this letter and print it”
- Find a few representative users
  - 3 is often enough to find obvious problems
- Watch them do tasks with the prototype

How to Watch Users

- Brief the user first
  - “I’m testing the system, not testing you”
  - “If you have trouble, it’s the system’s fault”
  - “Feel free to quit at any time”
  - Ethical issues: informed consent
- Ask user to think aloud
  - Be quiet!
    - Don’t help, don’t explain, don’t point out mistakes
    - Two exceptions: prod user to think aloud, and move on to the next task when stuck
- Take lots of notes
Watch for Critical Incidents

- Critical incidents: events that strongly affect task performance or satisfaction
  - Usually negative
    - Errors
    - Repeated attempts
    - Curses
  - Can also be positive
    - “Cool!”
    - “Oh, now I see.”

Summary

- You are not the user
- Keep human capabilities and design principles in mind
- Iterate over your design
- Write documentation
- Make cheap, throw-away prototypes
- Evaluate them with users