COGS 121
HCI Programming Studio

Week 02
Last Week

- Amy Fox: Intro to COGS 121
  - Class Logistics (lectures, discussion, web site, TopHat, Piazza)
  - Assignments and Projects (Assignment 0, DELPHI datasets)
  - Teams (CATME)
  - Project Management (Tools)
- Jasmine: Designing The Web
  - Network, Communication and the Internet
  - HTML, Javascript CSS, JSON, AJAX
- Yannis Katsis: DELPHI
  - Datasets
  - Access to Data
Logistics: Questions?

COGS 121

Cognitive Science 121 is the second course in the HCI sequence and naturally follows COGS 120 (CSE 170). It focuses on interaction architectures, programming techniques, and cognitive and computational principles for designing effective systems.

We begin by covering the fundamentals of user interface design for web-based systems and quickly move to larger scale design and implementation activities.

A major component of the class is completion of a substantial programming project in which students work together in small teams.

Phase 1

The first part of the class (weeks 1-5) will have a dual focus:

- On Tuesdays’ Design Lectures we will learn fundamental concepts, theories and principles for designing effective user interfaces.
- On Thursdays Tech Lectures we will create the basis for being able to implement some of these concepts as part of real web-based applications. In particular we will work with the MEAN stack (MongoDB, ExpressJS, AngularJS and NodeJS) and relational databases (MySQL/PostgreSQL) and we will apply concepts from data mining, social computing (exploiting the Facebook/Twitter Social APIs and SDKs), and interactive visualization (using the D3 Javascript visualization library). We will also learn how to use high-fidelity and low-fidelity wire framing tools (e.g. Axure and Balsamiq or others) to rapidly prototype interactive applications.

Phase 2

The second part of the class (weeks 6-10) will focus on a group project (4-5 students per group) that will leverage the technology learned in the first part.

We will partner with the DELPHI project at UCSD and exploit the population health data, environmental data, and other data they are collecting to drive the implementation of web-based interactive data analytics.
Interactive Systems
Usability, Interface Design, and Design Goals
Usability

• Synonyms for “user-friendly” are easy to use; accessible; comprehensible; intelligible; idiot proof; available; and ready

• But a “friend” also seeks to help and be valuable. A friend is not only understandable, but understands. A friend is reliable and doesn’t hurt. A friend is pleasant to be with.

• These measures are still subjective and vague, so a systematic process is necessary to develop usable systems for specific users in a specific context
Usability Requirements

• The U.S. Human Engineering Design Criteria for Military Systems already in 1999 stated these purposes:
  
  • Achieve required performance by operator, control, and maintenance personnel
  
  • Minimize skill and personnel requirements and training time
  
  • Achieve required reliability of personnel-equipment/software combinations
  
  • Foster design standardization within and among systems

• Should improving the user’s quality of life and the community also be objectives?

• Usability requires project management and careful attention to requirements analysis and testing for clearly defined objectives
Usability ISO 9241 definition

• The effectiveness, efficiency and satisfaction with which specified users achieve specified goals in particular environments.

  • **Effectiveness**: the accuracy and completeness with which specified users can achieve specified goals in particular environments

  • **Efficiency**: the resources expended in relation to the accuracy and completeness of goals achieved

  • **Satisfaction**: the comfort and acceptability of the work system to its users and other people affected by its use
Usability Measures

- Define the target user community and class of tasks associated with the interface

- 5 human factors central to community evaluation:
  - **Time to learn**
    How long does it take for typical members of the community to learn relevant task?
  - **Speed of performance**
    How long does it take to perform relevant benchmarks?
  - **Rate of errors by users**
    How many and what kinds of errors are made during benchmark tasks?
  - **Retention over time**
    Frequency of use and ease of learning help make for better user retention
  - **Subjective satisfaction**
    Allow for user feedback via interviews, free-form comments and satisfaction scales
Cognitive/Perceptual Abilities

• The human ability to interpret sensory input rapidly and to initiate complex actions makes modern computer systems possible
  • Long-term and semantic memory
  • Short-term and working memory
  • Problem solving and reasoning
  • Decision making and risk assessment
  • Language communication and comprehension
  • Search, imagery, and sensory memory
  • Learning, skill development, knowledge acquisition, and concept attainment
Cultural Diversity

- Characters, numerals, special characters, and diacriticals
- Left-to-right versus right-to-left versus vertical input and reading
- Date and time formats
- Numeric and currency formats
- Weights and measures
- Telephone numbers and addresses
- Names and titles (Mr., Ms., Mme.)
- Social-security, national identification, and passport numbers
- Capitalization and punctuation
- Sorting sequences
- Icons, buttons, colors
- Pluralization, grammar, spelling
- Etiquette, policies, tone, formality, metaphors
User’s Skill Level

• “Know thy user”

• Age, gender, physical and cognitive abilities, education, cultural or ethnic background, training, motivation, goals and personality

• Design goals based on skill level
  • Novice or first-time users
  • Knowledgeable intermittent users
  • Expert frequent users

• Multi-layer designs
8 Golden Rule
Shneidernam (1987, 1998, ... 2010!)

1. Strive for consistency
2. Cater to universal usability
3. Offer informative feedback
4. Design dialogs to yield closure
5. Prevent errors
6. Permit easy reversal of actions
7. Support internal locus of control
8. Reduce short term memory load
UI Design Guidelines
(Nielsen & Molich, 1993)

• Visibility of system status
• Match between system & real world
• User control & freedom
• Consistency & standards
• Error prevention
• Recognition rather than recall
• Flexibility & efficiency of use
• Aesthetic & minimalist design
• Help users recognize, diagnose, & recover from errors
• Provide online documentation & help
UI Design Guidelines
(Stone et al., 2005)

• **Visibility**: First step to goal should be clear

• **Affordance**: Control suggests how to use it

• **Feedback**: Should be clear what happened or is happening

• **Simplicity**: as simple as possible & task-focused

• **Structure**: content organized sensibly

• **Consistency**: similarity for predictability

• **Tolerance**: prevent errors, help recovery

• **Accessibility**: usable by all intended users, despite handicap, access device, or environmental conditions
Design Principles
TopHat:
Good/Bad Design?
TopHat: Design Disasters
Design principles
(not exhaustive list)

- Visibility – can I see it?
- Feedback – what is it doing now?
- Constraints – why can’t I do that?
- Mapping – where am I and where can I go?
- Consistency – I think I have seen this before?
- Affordance – how do I use it?
- Mental/conceptual models – I think I know how this operates?
Visibility

• Can see the state of a device and possible actions

• Systems are more usable when they clearly indicate
  • their status
  • the possible actions that can be performed
  • and the consequences of those actions
Visibility

- Car controls are positioned in a way that they can be easily found and used
Visibility

• Problems arise when we cannot see how to use a device
  • Sensor technology like auto faucets – not sure how to use – guess where to put hands
  • Visible knobs, dials and buttons have been replaced by invisible and ambiguous “active zones”
Visibility

- Elevator Control Panels: is it so difficult?

http://www2.isye.gatech.edu/~jjb/misc/elevators/elevators.html
Visibility

- Google makes it clear where to enter text
TopHat
Car Dashboard and Visibility
Solution? Nonsense?
Feedback

• What is it doing now?

• Sending information back to the user about what has been done

• Needs to be immediate and synchronized with user action

• Includes sound, highlighting, animation and combinations of these
  • Listen to your mouse when you click it
  • Can’t it be designed more silent? Should it be?
  • Glowing switch
  • Elevator buttons
Constraints

- Restricting the possible actions (the kind of interaction) that can be performed (can take the place)

- Helps prevent user from selecting incorrect options

- Why do we need three attempts to insert USB plugs?

  - Tip: Always keep seams down!
Constraints

- Only enable possible options
Mapping

- Turn a wheel, flip a switch, or push a button
  - What effect do you expect?

- Mapping is a relationship between controls and their movements and the results in the world (effects)

- **Good (natural) mapping** -> the effect corresponds to the expectation

- **Poor mapping** -> the effect does not correspond to the expectation

- Good mapping between controls and their effects results in greater ease of use
TopHat: Mapping
Mapping

• Good mapping primarily a function of similarity

• Meaning – e.g., shutdown button is colored red (people associate red with STOP)
Consistency

• Design interfaces to have similar operations and use similar elements for similar tasks

• For example:
  • Always use ctrl key plus first initial of the command for an operation – Ctrl(Command)+C, Ctrl+S, Ctrl+V ...

    • What does V stand for? Mapping?

• Main benefit is consistent interfaces are easier to learn and use
Consistency breakdowns

- What happens if there is more than one command starting with the same letter?
  - e.g. save, spelling, select, style

- Have to find other initials or combinations of keys, thereby breaking the consistency rule
  - E.g. Ctrl+S, Ctrl+Sp, Ctrl+shift+L

- Increases learning burden on user, making them more prone to errors
Inconsistencies
Affordance

- Refers to an attribute of an object that allows people to know how to use it

- How something looks indicates how it’s can be used
  - Chair for sitting
  - Table for placing things on
  - Knobs for turning
  - Slots for inserting things into
  - Buttons for pushing
Affordance

- Refers to an attribute of an object that allows people to know how to use it
Affordance

- Complex things may need explaining, but simple things should not
  - When simple things need pictures, labels, instructions, then design has failed
  - Their usage should be obvious based upon their appearance

Why do we have (pull) bars on both sides?
Norman’s Doors

https://www.youtube.com/watch?v=yY96hTb8WgI&noredirect=1
Mental model

- Representations of systems derived from experience

- People understand and interact with systems based on mental representations developed from experience
  - They compare the outcomes of their mental models with the real-world systems:
    - When the outcomes correspond, a mental model is accurate
    - When the outcomes do not correspond, the mental model is inaccurate or incomplete
Mental model

- A mental model allows the user to simulate the operation of the device
- A good mental model allows the user to predict the effects of their actions

Mental models built from:
- Affordances and constraints
- Mappings
- Transfer effects
- Population stereotypes/cultural standards
- Instructions
- Interactions
Mental model

- Models allow people to mentally simulate operation of design (device, software,...)

- Models may be wrong, particularly if the above attributes are misleading
Mental model

The case of ABS

- Anti-lock braking system (ABS) vs. conventional brakes
- Statistics show that ABS have not reduced the frequency or cost of accidents
  - In spite of the fact that they have measurable safety benefits in controlled tests
  - Why is this so?

- We can explain this by using the concept of wrong mental model of ABS
Mental model

The case of ABS

• Good interaction model for conventional brakes
  • On slick surfaces
    • Depress the brake pedal smoothly
    • Pump brakes to prevent them from locking up
    • Do not steer while braking, except to counter-steer
    • Noise and vibrations are signs that something is wrong

• Good interaction model for ABS
  • On slick surfaces
    • Depress the brake pedal fast and hard
    • Do not pump breaks
    • Steer while braking
    • Noise and vibrations are signs that the system is operating properly
Mental model
The case of ABS

• The likely cause that ABS have not made the difference
  • People are not using ABS properly (wrong metal model)
  • They transfer the knowledge from conventional breaks
  • All this simply means that ABS are not properly designed
  • It also suggests that designers gave little consideration to the interaction model of the target audience in the design process

• So, design with people’s interaction models in mind
Next Steps

• Today (until the end of class)
  Design Principles in action: Amy Fox

• Tech Lecture: Thu 4/7
  - Web technologies and Mining Social Media: Instagram and Facebook - Introduction to Assignment 1 (Social Media)
  - Project Teams and Assignment 1

• Discussion Studios (required), Fri 4/8
  Assignment 1
  Quiz on Week 1 and Week 2 material (design lecture, tech lecture, assignment 0 and readings)

• Readings (required)
  - Shneiderman (Designing the User Interface) - Chapter 5
  - Johnson (Designing With the Mind in Mind) - Chapter 1 + Appendix A

• Next Design Lecture: Tue 4/12
  Direct Manipulation and Perception
Design principles
...in action
Design and Emotions

Don Norman, 2003

http://www.ted.com/talks/don_norman_on_design_and_emotion
Thanks