CSC540, Fall 2003 - Introduction

We will cover development of user interfaces (to computer systems) from perspective of the field of human-computer interaction (HCI).

What is HCI?
HCI's roots are in Human Factors & Ergonomics: study of human interaction with machines.

**Human-computer interaction:** term adopted in 1980's to describe birth of new field

- Interested in more than just visual design of interface
  - Term "user-friendly" was becoming a cliché for better-looking graphics, but not necessarily better for the user

- HCI: a "set of processes, dialogues, and actions through which a human user employs and interacts with a computer" (Baecker & Buxton, 1987)

- HCI: "a discipline concerned with the design, evaluation and implementation of interactive computing systems for human use…" (ACM SIGCHI, 1992).
HCI is a Multidisciplinary field:
- **computer science and engineering**: graphics, artificial intelligence (AI), theory of programming languages (OOP), software architecture, new hardware
- **psychology (\& other fields that study human behavior)**: psychology, sociology, linguistics, anthropology
- **graphic arts**

Influence of psychology on HCI:
- emphasis on **designing for human**: based on knowledge about human capabilities
- empirical approach to **evaluation** of interfaces: healthy (polite) skepticism about claims until validated by experiments

Examples of **jobs in HCI**: (Note: often working on same team)
- **Computer programmer**: implements interactive systems with complex user interface (BS/MS in CS w/ HCI courses)
- **Usability tester**: Tests usability of proposed or finished interfaces (MS in HCI, or Psy w/HCI)
- **Graphic artist/designer** (MS in Design w/HCI)
- **HCI Researcher**: (Ph.D. in CS or Psy)
For more info on careers, see links on course web page to main professional organizations in USA:
• ACM SIGCHI
• etc. …

Goal of this course to give you broad understanding of issues in developing human-computer interfaces
• not just the popular style of interfaces today (web)
• not just what is possible using the tools of today

Examples of human-computer interfaces

• Command-line "dialogue": UNIX shell
  User:   >cd
  System: /afs/…/nlgreen

• GUIs (Graphical User Interfaces): many styles, e.g.
  • MS Windows & Mac OS
  • Web pages
  • Drawing tools (CAD, pedigree, etc.)

• Speech input/output: train schedule, movie, weather
**Multimodal input:**
- *Modality* = type of perception/action
- Example: system understands users' input by: speech, pointing to map, selecting region on map
  - **System:** [Displays map with cargo icons]
  - **User:** [Draws bounding box around a port.] "Move this [point] cargo to that port [in bounding box]."

**Interfaces of the future:** ex. VR, ubiquitous, wearable, affective (we can't even imagine all of them!)

Sometimes useful to study good/bad examples of **user interfaces to non-computer systems**, e.g:

- **People-to-people:** speech (with intonation), gesture (deictic/non-deictic), gaze
- **People-to-stove/oven:** temperature setting, temperature feedback, select which burner/oven, feedback to tell when burner/oven is on

**Number of controls** no excuse for bad design:
- Most people can learn to use a new car quickly
- Many people have trouble learning to use VCR
Two **key principles of good interface design** (whether computer or not) [Norman, 1988, 1992]:

- **Visibility**: controls are visible; relation between user goals, user actions, and effects are clear and make sense
  - Example of poor visibility: 4-burner stove where order of controls did not match position of burners

- **Perceived affordances**: what operations the user *thinks he* can do on interface objects
  - Example of bad design: door that you cannot tell if you are supposed to push/pull to open it

Ask class: other examples of non-computer interfaces that are bad because of poor visibility or poor use of affordances?

See links on course web page to examples of poor interfaces
- Bad human factors design web page
- Interface Hall of Shame
Important goal of user-interface design from HCI perspective is called **usability**

- How can usability be objectively measured?

**Usability Criteria:** (Shneiderman)

1. **Time to learn:** how long does it take for typical user to learn to perform typical task using the interface?

2. **Speed of performance:** how long does it take to perform a typical task using the interface?

3. **Error rate:** How many and what kinds of errors do typical users make using the interface?

4. **Retention:** How well do users remember how to use the interface?

5. **User satisfaction:** How much do the users enjoy using the interface?
Philosophy of system design in HCI different than traditional software engineering (i.e. as covered in CS340/640)

Phases in traditional Waterfall SW lifecycle model are strictly ordered:
1. **Requirements**: what does user need?
2. **Specification and design** of system: what should programmers build?
3. **Implementation** and unit testing
4. **Integration** and **system testing**

In contrast, **HCI philosophy** is

- **More user-centered**:
  - Requirements phase: expanded to analyze users and their tasks
  - Try to involve user in all phases
  - Use of prototypes

- **More iterative**: phases not in strict order and may be repeated as often as necessary to improve usability

- Evaluation (**usability testing**) performed throughout the lifecycle