HCI

- Human Computer Interaction

- HCI studies the design of effective interfaces between computer software and a human (or humans).

- An “effective” interface allows a system to deliver its functionality.

- A poor interface can prevent even the world’s greatest software from being effective.

- Ineffective interfaces cause software not to be used.

- Effective interfaces can save huge amounts of money.
  - e.g., faster, fewer errors, satisfied users.
The subject of Algorithms is concerned with the *design* of processes.

- {small scale structure and behavior}

Software Engineering is concerned with the *design* of programmed systems.

- {large scale structure and behavior}

HCI is concerned with the *design* of interfaces to such systems.

- {the delivery of functionality of the programmed systems}
What factors influence a design?

- General properties of the human user.
  - Memory, Perception, Motor skills.
  - Natural abilities and limitations.

- The characteristics of the user.
  - e.g., personality, education, etc.

- The user’s task.
  - e.g., stressful, repetitive, etc.

- The interaction hardware.
  - e.g., screen, mouse, etc.

- The interaction and display methods used.
  - menus, layout, colors, icons, etc.
  - strengths and weaknesses of methods.

- The implementation of the methods.
  - e.g., speed, tool selection and use, etc.

- Graphical design.
IT’S OBVIOUS!

Good HCI is very hard, and requires:
- a lot of knowledge of HCI research;
- knowledge of interaction techniques;
- design skill;
- experimental evaluation;
- knowledge of what can be programmed;
- experience;
- good taste.

Good HCI requires “trade-offs”:
- It’s rare that a particular method is ideal.
- It’s rare that one choice is independent of another.
- An interface is often more than the sum of the parts.
- The quality of an interface is in the details.

Experimental results show what techniques work under which circumstances.
- When is it bad to use a Mouse?
- When should you use red letters?
The majority of interactions with a computer are not for programming.

Your computer use is *not* typical.

You are *not* a typical computer user.

Designing interfaces that you like is *not* a good way to design interfaces in general.

Interfaces need to be developed for all User/Task combinations.
These are NOT *normal* people!!

Normal people do not use computers.

When you design interfaces you are not designing them for yourself. You design them for people doing a task.
Reverse Engineering...
... is concerned with figuring out why an artifact was designed to be that way.

You need a lot of knowledge to do it!

So... try to reverse engineer the WPI Home Page on the Web.

- Can you figure out what all the requirements were for the Home Page?
- Why does it look like that?
- Why is it arranged in that manner?
- Why does it include that text?
- Why are things in that order?
COURSE GOALS

Some of the goals of this course are to:

- Provide you with web page design and development skills.
- Have you read and analyze articles from the recent HCI literature.
- Have you do an HCI experiment.
- Have you design and build an interface using event driven programming.
- Have you go through a group-based creative design process for an interface.
- Provide you with the HCI knowledge that you’ll need for effective interface design.