

Usability

Prof. Rob Miller
MIT EECS

Spring 2006

6.170 Laboratory in Software Engineering

1

User Interface Hall of Shame



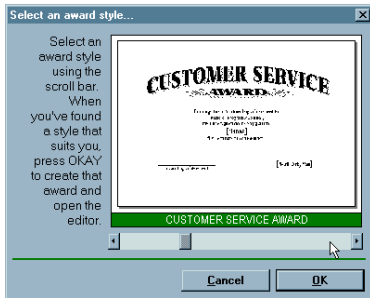
Source: Interface Hall of Shame

Spring 2006

6.170 Laboratory in Software Engineering

2

User Interface Hall of Shame



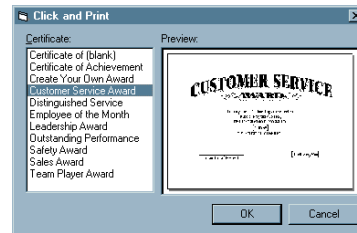
Source: Interface Hall of Shame

Spring 2006

6.170 Laboratory in Software Engineering

3

Redesigning the Interface



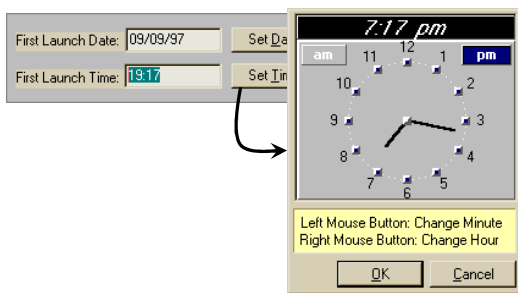
Source: Interface Hall of Shame

Spring 2006

6.170 Laboratory in Software Engineering

4

Another for the Hall of Shame



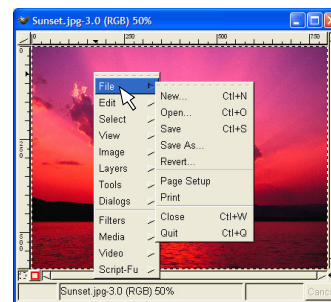
Source: Interface Hall of Shame

Spring 2006

6.170 Laboratory in Software Engineering

5

Hall of Fame or Hall of Shame?



Spring 2006

6.170 Laboratory in Software Engineering

6

Hall of Shame or Hall of Fame?



Spring 2006

6.170 Laboratory in Software Engineering

7

The User Interface Is Important

- User interface strongly affects perception of software
 - Usable software sells better
 - Unusable web sites are abandoned
- Perception is sometimes superficial
 - Users blame themselves for UI failings
 - People who make buying decisions are not always end-users

Spring 2006

6.170 Laboratory in Software Engineering

8

User Interfaces Are Hard to Design

- You are not the user
 - Most software engineering is about communicating with other programmers
 - UI is about communicating with users
- The user is always right
 - Consistent problems are the system's fault
- ...but the user is not always right
 - Users aren't designers

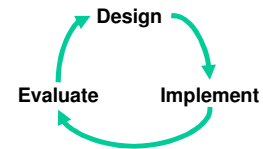
Spring 2006

6.170 Laboratory in Software Engineering

9

Iterative Design

- UI development is an iterative process



- Do you want the design cycle to be internal, or visible to your customers?

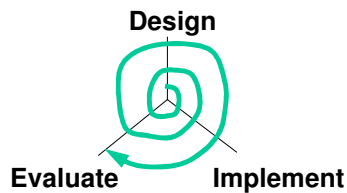
Spring 2006

6.170 Laboratory in Software Engineering

10

Spiral Model

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



Spring 2006

6.170 Laboratory in Software Engineering

11

Usability Defined

- 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?
 - Memorability: is it easy to remember what you learned?
 - Errors: are errors few and recoverable?
 - Satisfaction: is it enjoyable to use?

Spring 2006

6.170 Laboratory in Software Engineering

12

Outline

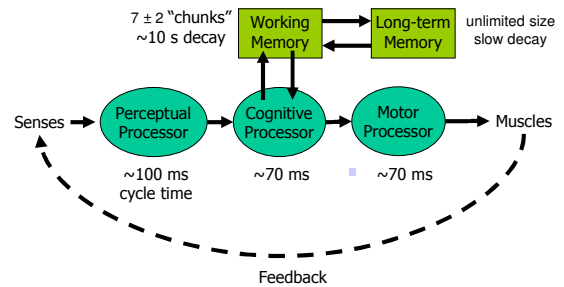
- Rest of today: human capabilities
 - Perception
 - Motor skills
 - Memory
 - Color vision
- Next time: usability engineering

Spring 2006

6.170 Laboratory in Software Engineering

13

Human Information Processing



Spring 2006

6.170 Laboratory in Software Engineering

14

Perceptual Fusion

- Two stimuli within the same cycle of the perceptual processor ($T_p \sim 100\text{ms}$) appear **fused**
- Consequences
 - 10 frames/sec is enough to perceive a moving picture
 - Computer response < 100 ms feels instantaneous
 - Causality is strongly influenced by fusion

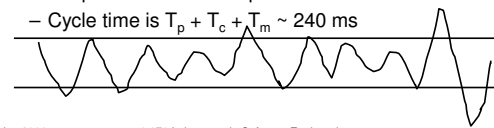
Spring 2006

6.170 Laboratory in Software Engineering

15

Motor Processing

- Open-loop control
 - Motor processor runs a program by itself
 - Cycle time is $T_m \sim 70\text{ ms}$
- Closed-loop control
 - Muscle movements (or their effect on the world) are perceived and compared with desired result
 - Cycle time is $T_p + T_c + T_m \sim 240\text{ ms}$



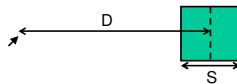
Spring 2006

6.170 Laboratory in Software Engineering

16

Pointing Tasks: Fitts's Law

- How long does it take to reach a target?



- Moving mouse to target on screen
- Moving finger to key on keyboard
- Moving hand between keyboard and mouse

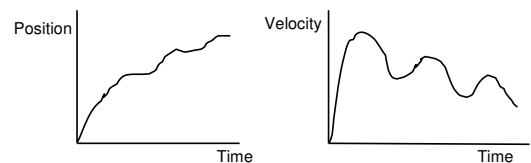
Spring 2006

6.170 Laboratory in Software Engineering

17

Analytical Derivation of Fitts's Law

- Moving your hand to a target is closed-loop control
- Each cycle covers remaining distance D with error ϵD



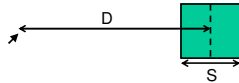
Spring 2006

6.170 Laboratory in Software Engineering

18

Fitts's Law

- $T = RT + MT = a + b \log(D/S)$



- $\log(D/S)$ is the *index of difficulty* of the pointing task

Spring 2006

6.170 Laboratory in Software Engineering

19

Path Steering Tasks

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



- This is why cascading menus are slow!

Spring 2006

6.170 Laboratory in Software Engineering

20

Memory

- Working memory
 - Small: 7 ± 2 "chunks"
 - Short-lived: ~10 sec
 - Maintenance rehearsal fends off decay (but costs attention)
- Long-term memory
 - Practically infinite in size and duration
 - Elaborative rehearsal transfers chunks to long-term memory



Spring 2006

6.170 Laboratory in Software Engineering

21

Chunking

- "Chunk" = unit of perception or memory
- Chunking depends on presentation and existing knowledge
 - M W R C A A O L I B M F B I B
 - MWR CAA OLI BMF BIB
 - BMW RCA AOL IBM FBI
- 3-4 digit chunking is ideal for encoding unrelated digits

Spring 2006

6.170 Laboratory in Software Engineering

22

Color Blindness

- 8% of males can't distinguish red-green



- Blue-yellow color blindness also exists, but is rarer

Spring 2006

6.170 Laboratory in Software Engineering

23

Chromatic Aberration

- Lens can't focus blue and red at same time
- So blue-on-red text looks fuzzy and hurts to read



Spring 2006

6.170 Laboratory in Software Engineering

24

Next Time: Usability Engineering

- Design heuristics
- Low-fidelity prototypes
- User testing