EAST WEST UNIVERSITY

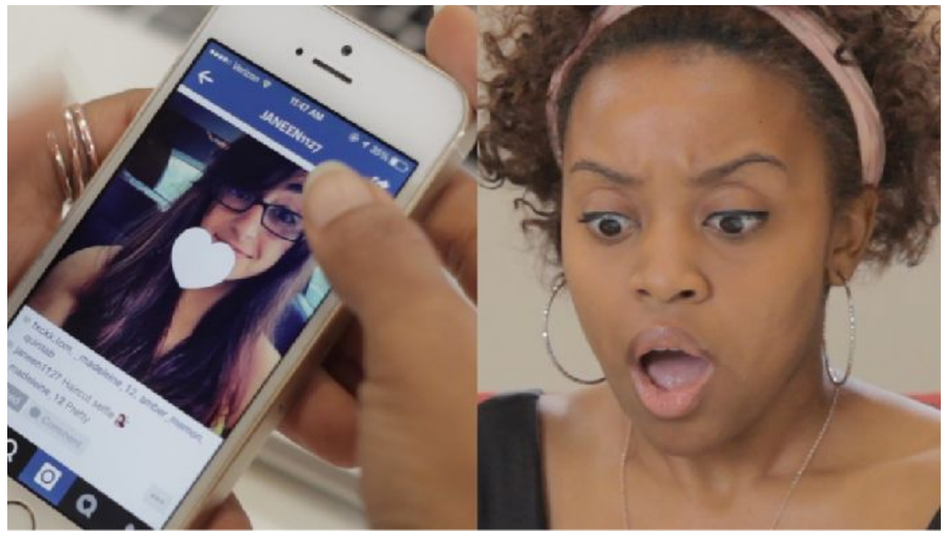
# CSE 428
# Human Computer Interaction

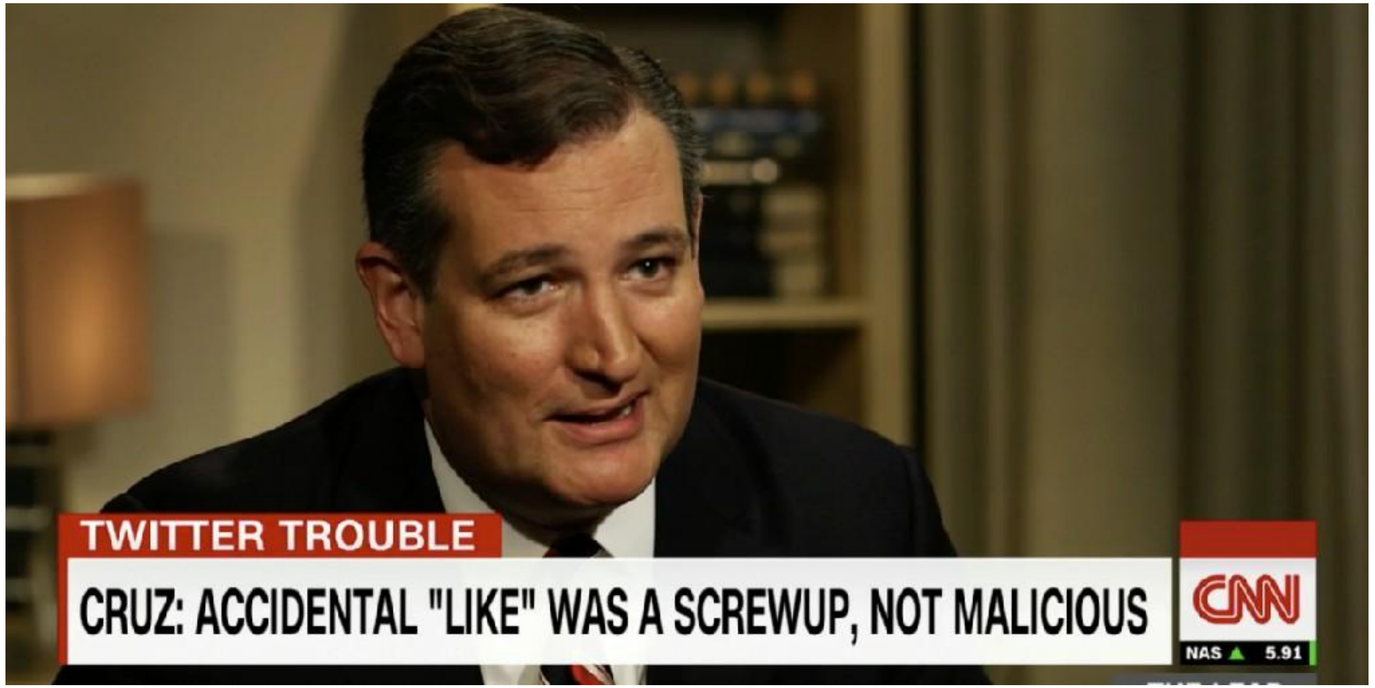
**Yasin Sazid**
Lecturer
Department of CSE
East West University

# Design Principles II

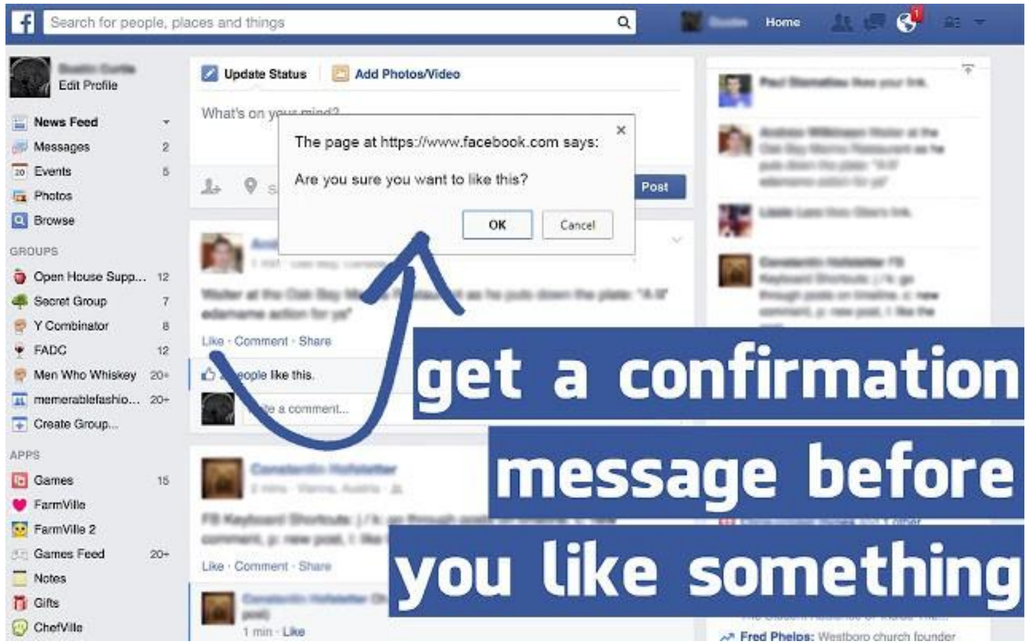
# Hall of Fame or Shame?

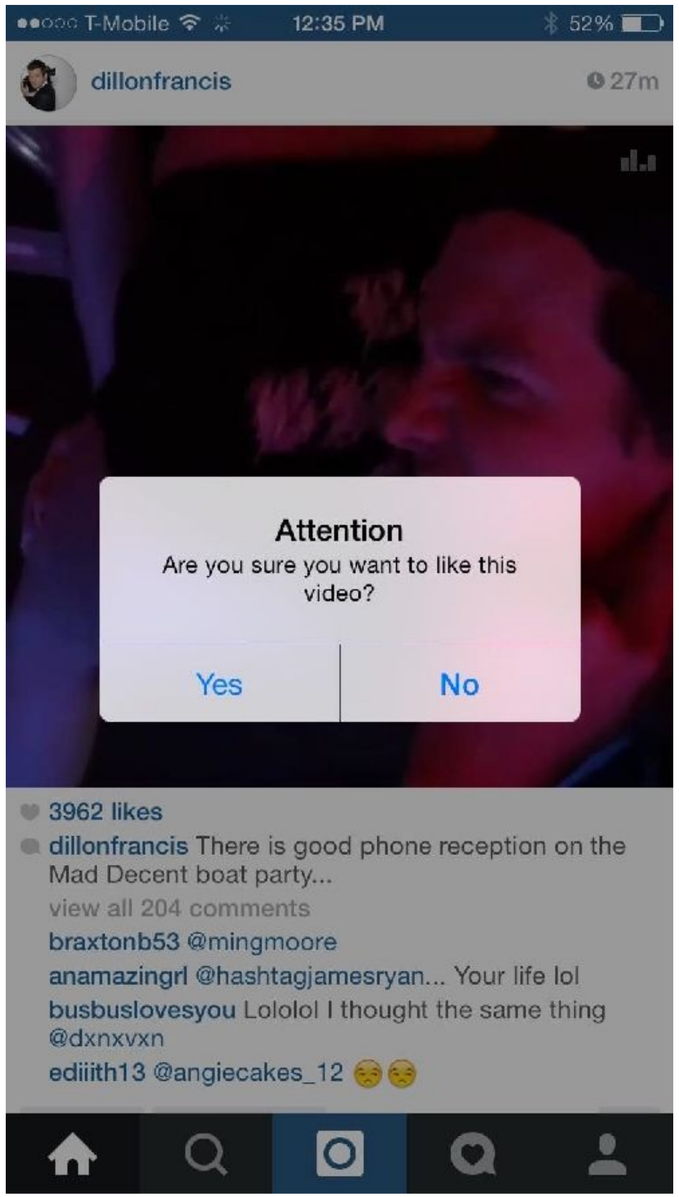

## The “Accidental Like”




TWITTER TROUBLE
CRUZ: ACCIDENTAL "LIKE" WAS A SCREWUP, NOT MALICIOUS
CNN
NAS 5.91


Tweets 19.4K
Following 7,740
Followers 3.01M
Likes 1,248
Follow
Ted Cruz
@tedcruz
Father of two, @heidiscruz's husband, fighter for liberty. Representing the great state of Texas in the U.S. Senate.
Houston, Texas
tedcruz.org
Joined March 2009
Likes
Sexuall Posts @SexuallPosts · Sep 11



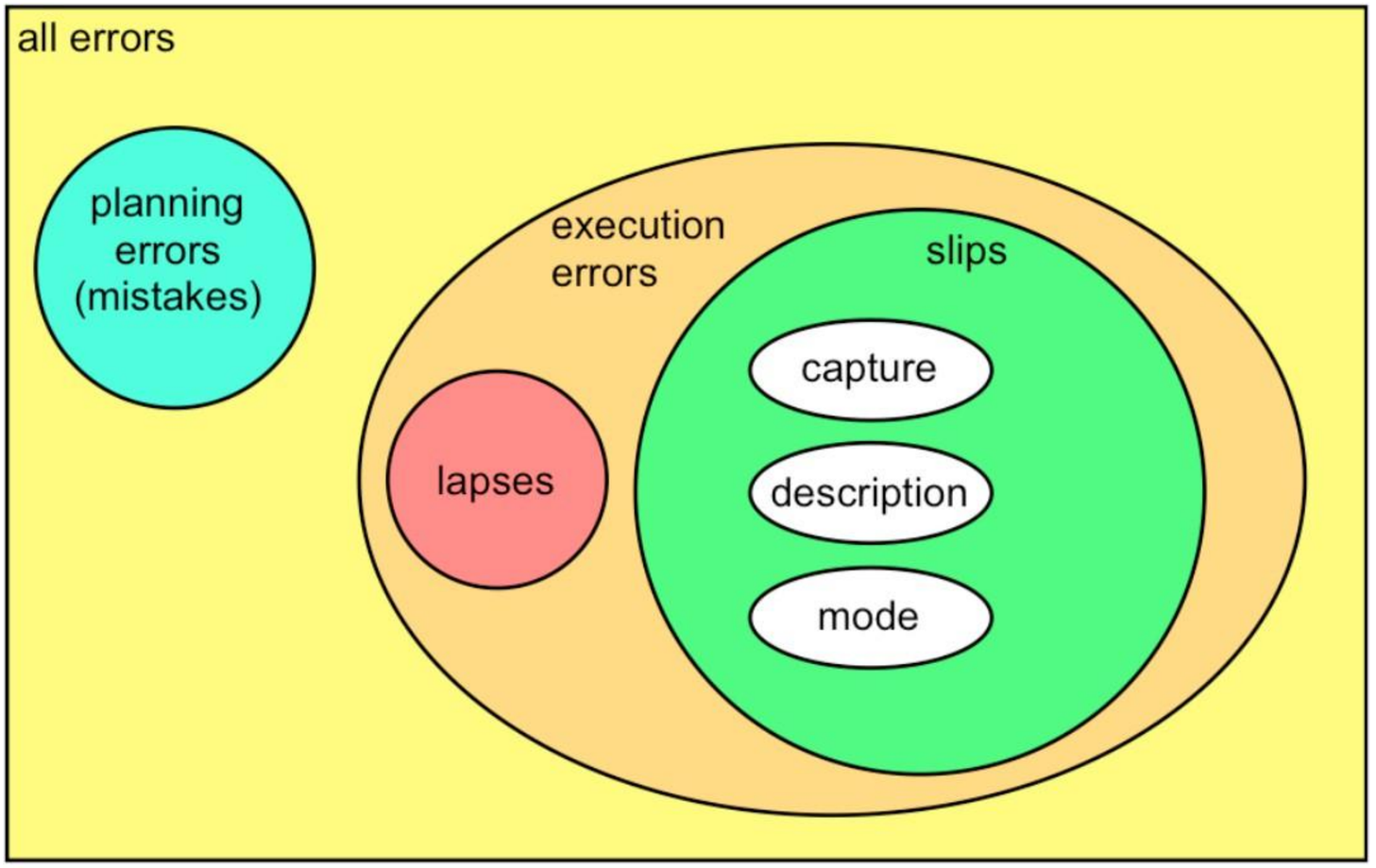

**How might you prevent this error?**

# Safety

# Kinds of Errors

# Kinds of Errors

- **Slips and Lapses**
  - Failure in successfully executing a skill that a user has already learned
  - **Slip**: Failure due to execution or control
    - Example: Missing the button on a click, Ctrl-V instead of Ctrl-C
  - **Lapse**: Failure due to memory
    - Example: Forgetting to add attachment to email
- **Mistakes**
  - Error made in planning or rule execution


all errors
planning errors (mistakes)
execution errors
slips
capture
description
mode
lapses

# Kinds of Slips

- **Capture Slip**
  - A person starts executing one sequence of actions, but then veers off into another (usually more familiar) sequence that happened to start the same way
  - Example: Leave your house and find yourself walking to school instead of where you meant to go
  - Example: In the text editor vi, it's common to quit the program by issuing the command ":wq", which saves the file (w) and quits (q). If a user intends just to save the file (:w) but accidentally quits as well (:wq), then they've committed a capture error.

# Kinds of Slips

- **Description Slip**
  - Two actions are very similar. The user intends to do one action, but accidentally substitutes the other.
  - Example: Reaching into the refrigerator for a carton of milk, but instead picking up a carton of orange juice and pouring it into your cereal.
  - Example: Mic Drop button looks like Send


# Kinds of Slips

- **Mode Error**
  - Modes are states in which the same action has different meanings. Slips happen when you forget which mode you are in.
  - Example: if the user means to type lowercase letters but doesn’t notice that Caps Lock is enabled, then a mode error occurs.

# Causes of Slips

- **Inattention!**
  - Involves execution of already learned behavior
  - Insufficient attention or distraction of attention at a key moment
- **“Strong but Wrong”**
  - Sometimes due to strong **similarity** to correct behavior (capture or description slips) or high **frequency** relative to correct behavior (capture slips)
- **Speed/Accuracy tradeoff**

# What kind of Error?



Slip

Lapse

Mistake

Description Slip

# What kind of Error?

meant to type

```
% rm *.class

        [select all files]
```

actually typed

```
% rm *>class

        [pipe output into file class]
```

Slip

Lapse

Mistake

Mode Error

Why? Look at where . and > are on your keyboard!

# Preventing Errors

# Remember Consistency

- Similar things should look and act similarly
  -> **Different things should look different**
- Keep dangerous commands away from common ones!

# Modes: yes or no?

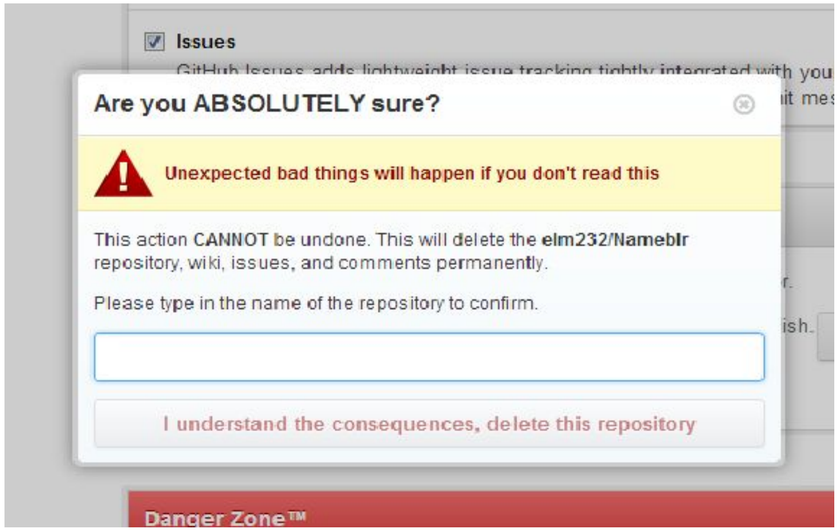
- Generally speaking, eliminate modes
- If you must use them:
  - Increase visibility of modes
  - Spring-loaded or temporary modes

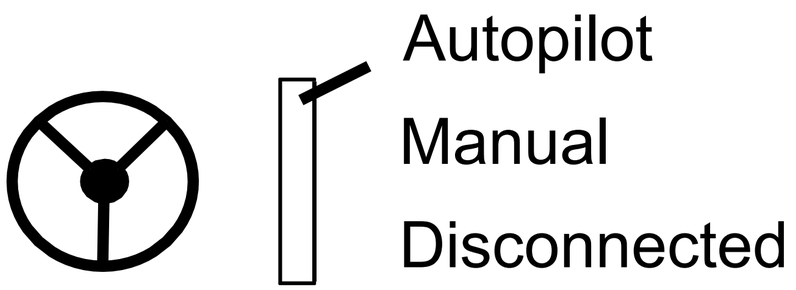
# Should you use confirmation dialogs?

- Reduces efficiency, requiring 2 actions now when it was 1.
- If frequently seen, then expert users will learn to expect it and habitually press OK without reading or noticing it! Now we're back to square one.
- In general, reversibility (i.e., **undo**) is a better solution.

# Should you use confirmation dialogs?

- You should use it for **rare, catastrophic** events.
- Make it look very different from everything else
- Draw attention to it: no OK button, forces you to think




https://en.wikipedia.org/wiki/Torrey_Canyon_oil_spill

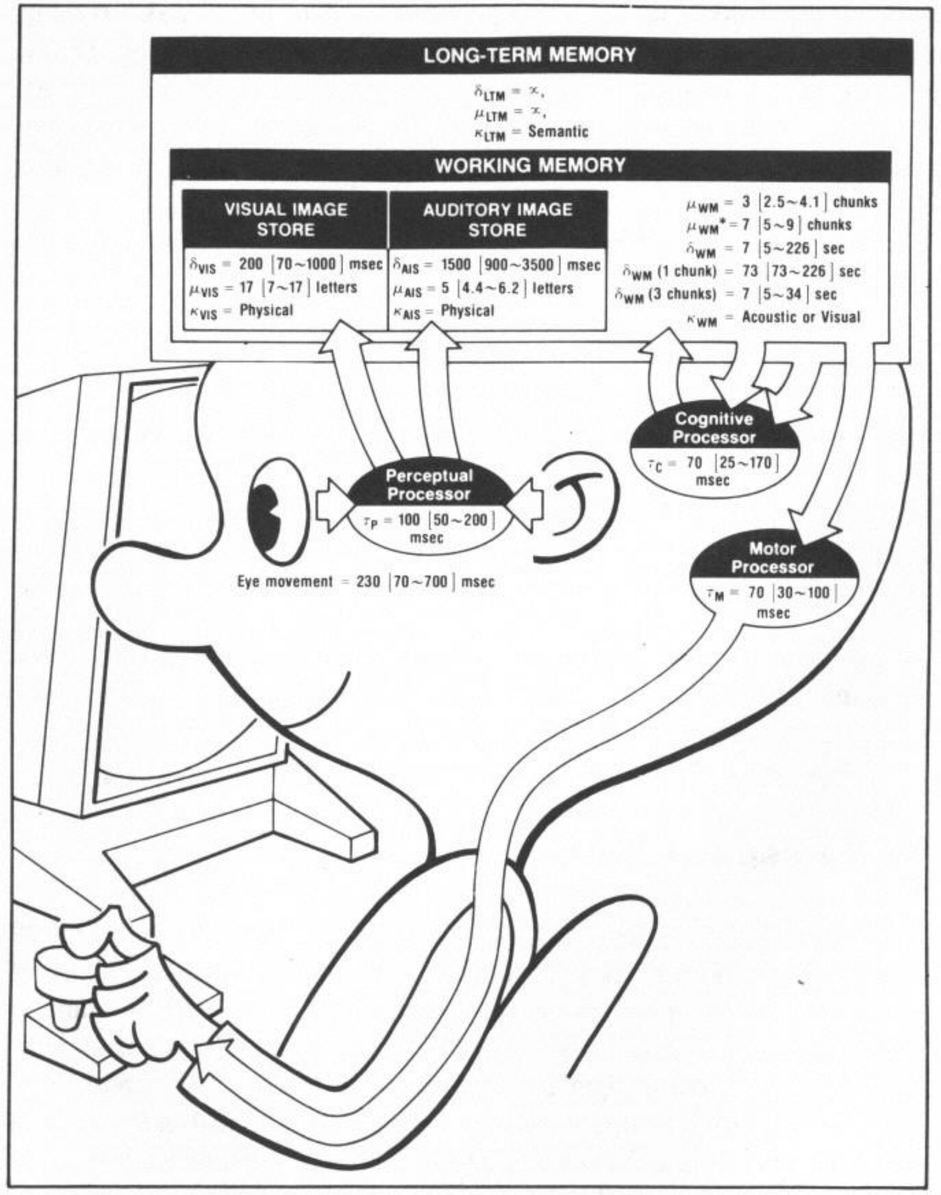
# Activity (6 min)

- Think about a time when you encountered problems due to a system’s lack of safety

- What did you do?
- What was the outcome?
- How might you change the system to prevent the problem?

# Efficiency

# Model Human Processor

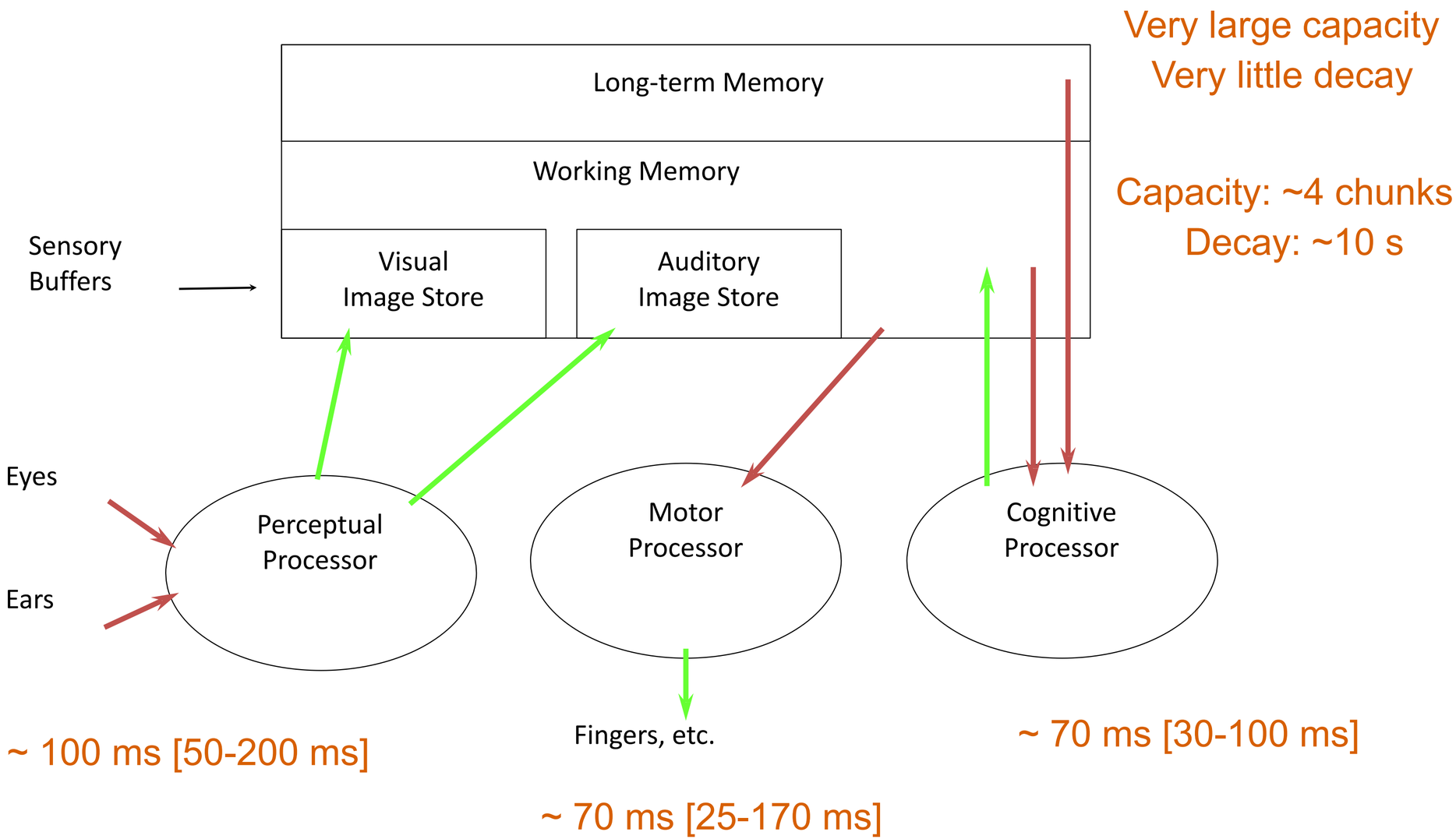



# The Model Human Processor

Developed by Card, Moran, & Newell (1983)

Based on empirical data, drawing an analogy between how humans remember things and how a computer accesses its memory

Same book that named “human computer interaction” for the first time


Very large capacity
Very little decay
Long-term Memory
Working Memory
Capacity: ~4 chunks
Decay: ~10 s
Sensory
Buffers
Visual
Image Store
Auditory
Image Store
Eyes
Ears
Perceptual
Processor
Motor
Processor
Cognitive
Processor
Fingers, etc.
~ 100 ms [50-200 ms]
~ 70 ms [25-170 ms]
~ 70 ms [30-100 ms]

# Chunking

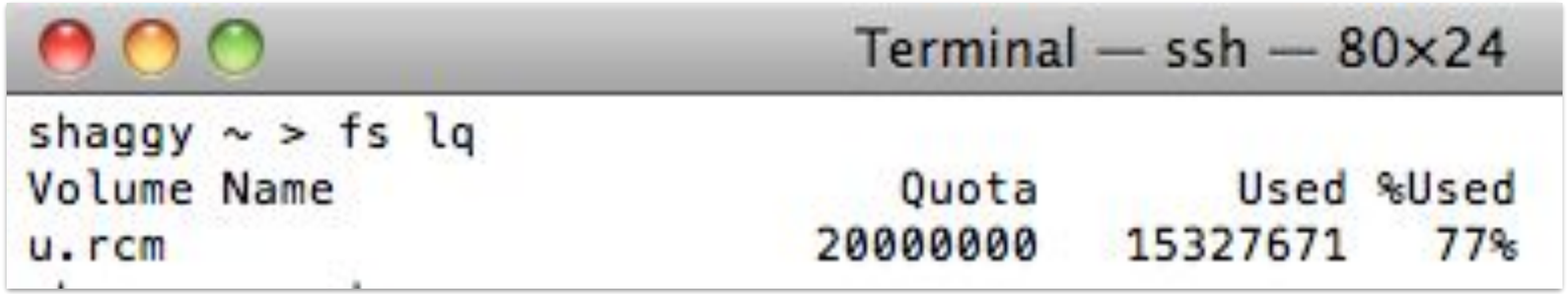
- A “chunk” is a unit of memory or perception
  - In one sense, chunks are defined symbols; in another sense, a chunk represents the activation of past experience.

# Memory

- Working memory:
  - Small! ~4 chunks.
  - Short-lived: ~10 seconds
- Our ability to form chunks in working memory depend on how the information is presented!
  - Grouping will improve efficiency of output to add to working memory
  - As will making the groups more familiar

Hard: M W B C R A L O A B I M B F I
Easier: MWB / CRA / LOA / BIM / BFI
Easiest: BMW / RCA / AOL / IBM / FBI

# Activity (10 min)




- Let's redesign this filesystem quota display for the command line so that it's more efficient. Remember to consider user goals.
- Sketch it on paper (but remember this is for the command line)!
- Come up with an idea on your own first but then discuss with a neighbor and improve it based on feedback

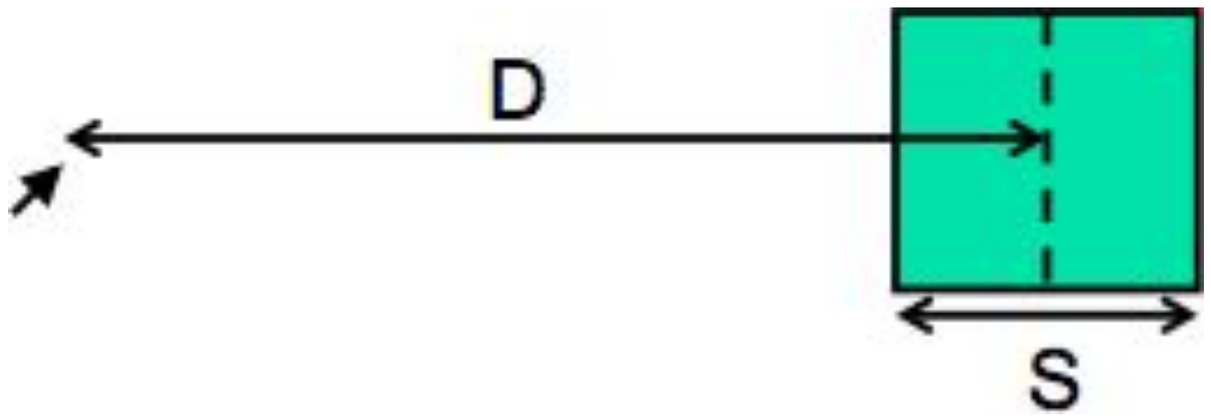
# Fitt’s Law

# Fitt's Law (1954)

- Models time to acquire targets in aimed movement
  - Reaching for a control in a cockpit
  - Moving across a dashboard
  - Pulling defective items from a conveyor belt
  - Clicking on icons using a mouse
- Very powerful, widely used
  - Holds for many circumstances (e.g., under water)
  - Allows for comparison among different experiments
  - Used both to measure and to predict

Time T to move your hand to a target of size S at distance D away is:

$$T = \text{Reaction Time} + \text{Movement Time} = a + b \log (D/S + 1)$$




As D goes up, time goes up.
As S goes up, time goes down.

important part of the equation is the

**Index of Difficulty (ID): log (D/S + 1)**

Fitts's Law claims that the time to acquire a target increases linearly with the log of the ratio of the movement distance (D) to target size (S)

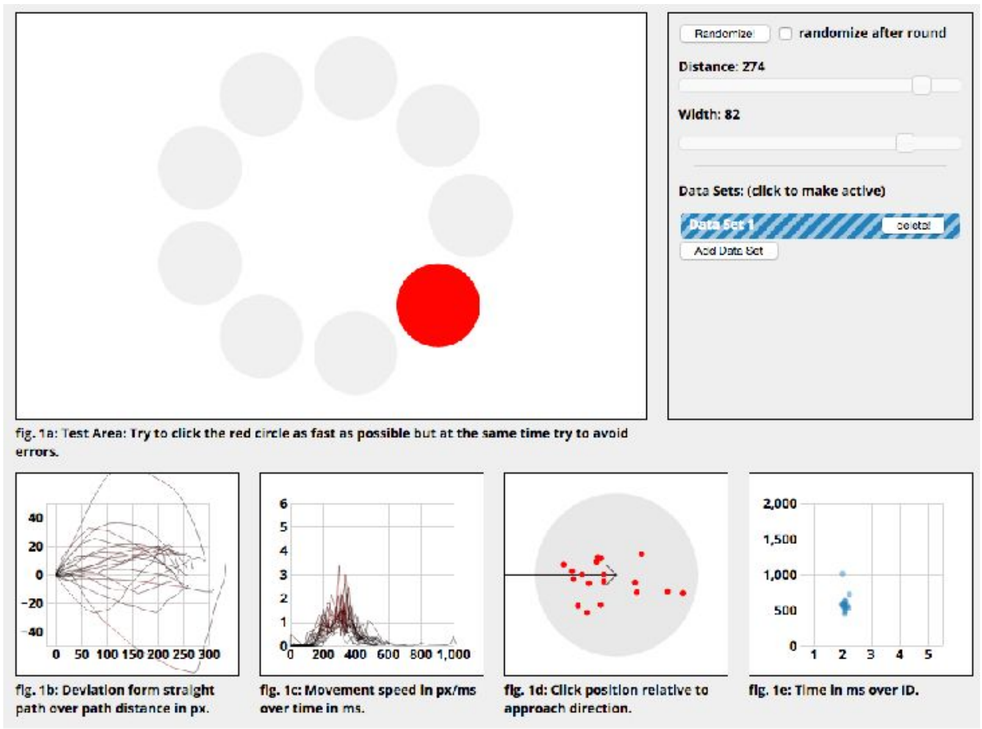
As D goes up, difficulty goes up.
As S goes up, difficulty goes down.

Because it's a ratio, units of D and S don't matter!
Allows comparison across experiments

# Fitts’s Law

interactive Fitts’ law test

http://www.simonwallner.at/ext/fitts/




- **Trial 1:** Easy: make targets large, put them close
- **Trial 2:** Harder: make targets small, space them out

- Click around the circle a couple times
- Look at the figures to compare & understand

$$T = \text{Reaction Time} + \text{Movement Time} = a + b \log (D/S + 1)$$

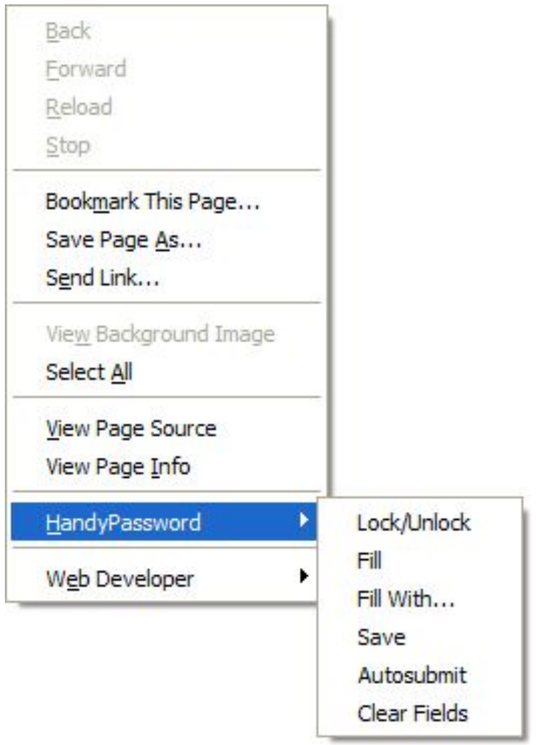
a = reaction time what is b?

b = **throughput**

- bandwidth of the communication channel from the human to the computer
- can be affected by anything from human (motor skills, fatigue), input device, display/feedback/perceptual skills

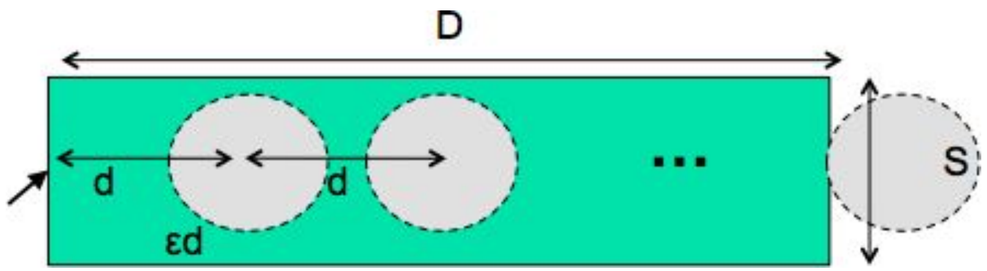
# Implications of Fitt's Law

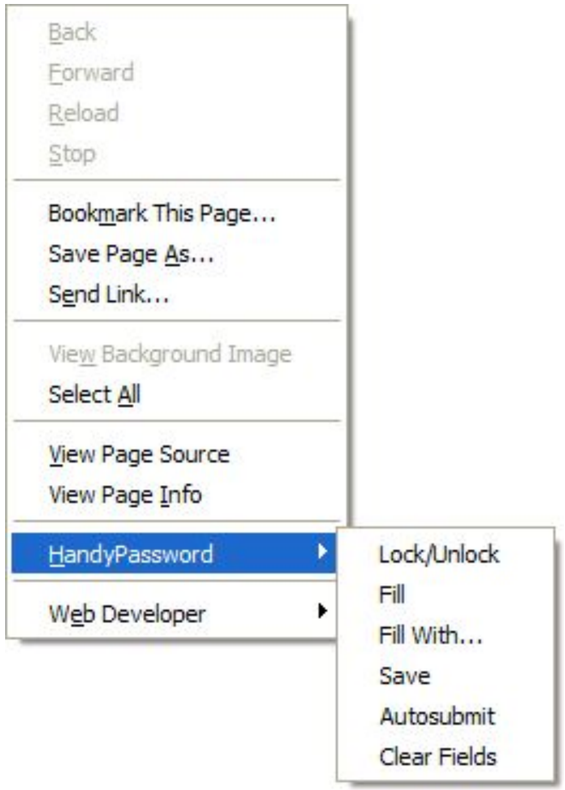
- Targets at the screen edge are easy to hit
  - Save them for frequent actions!
  - Unclickable margins are a bad idea
- Pie menus are actually easier than linear popup menus! (doesn't mean you should use them)



# Steering Tasks

- **Steering** is much harder than pointing because it constrains the size of the error you can make as you're moving towards a target.
- Thus, cascading submenus are hard to use



# Takeaways

- Make frequently-used targets big
- Put targets used together near each other
- Use screen corners and screen edges
- Avoid steering tasks

# THANK YOU