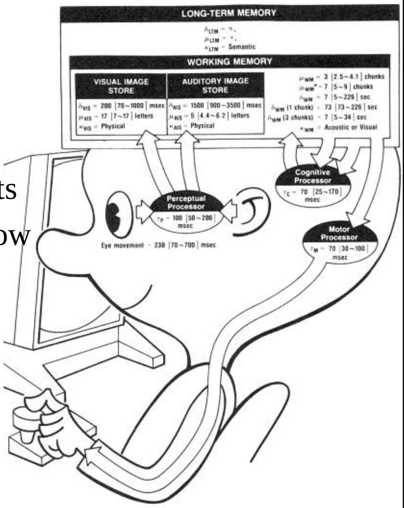
Questions?

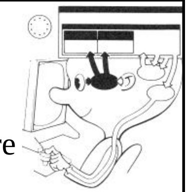
- Midterm
- In-class very short quizzes (out of class participation grade)

Human Information Processor (Card, Moran, Newell)



- Very influential
- Brought together several aspects of cognitive psychology
- Made accessible for computer scientists
- Practical model – **not** description of how things actually work

- What are the three components?
  - Perceptual Processor
  - Cognitive Processor
  - Motor Processor



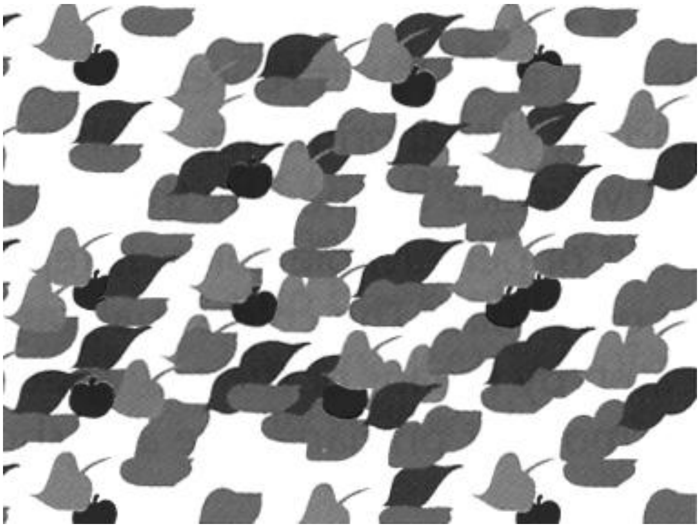
Perceptual Processor

- Physical sensing, here sight, into Visual Image Store
- Difference between **perceptual memory** and **working memory?**
- Decode for transfer to working memory
  - Progressive
    - *Example: 10ms/letter*
  - Selective
    - *Spatial*
    - *Pre-attentive: color, direction…*
    - *Not semantic (can't do "odd numbers")*

Perceptual Processor

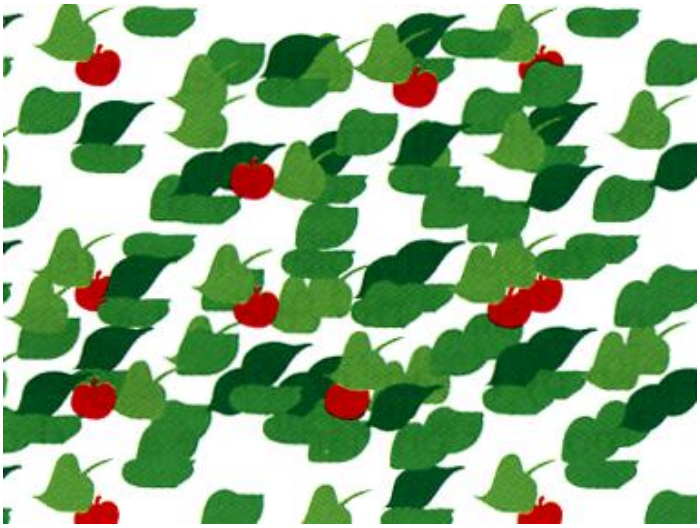
- Capacity
  - Example: 17 letters
- Visual image store fades
  - Half-life: 200 ms [90 – 1,000ms]
- Cycle time
  - Initial impression to being available in visual image store
  - τ = what? 100 ms [50 – 200ms]
- What is Slowman, Middleman, and Fastman?
- Why do events separated by < 0.1sec seem causally related?
- Why are films shown at 24 fps then instead of 10 fps?

Where are the cherries?



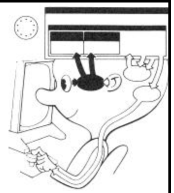
From Information Visualization, C. Ware

Where are the cherries?

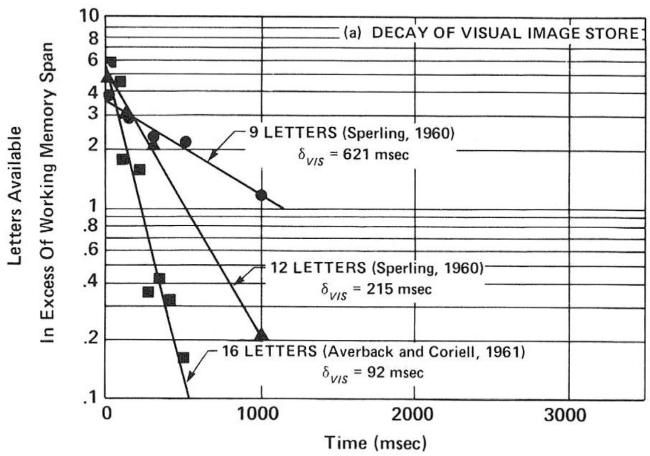


From Information Visualization, C. Ware

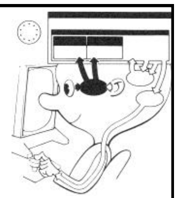
Perceptual Processor



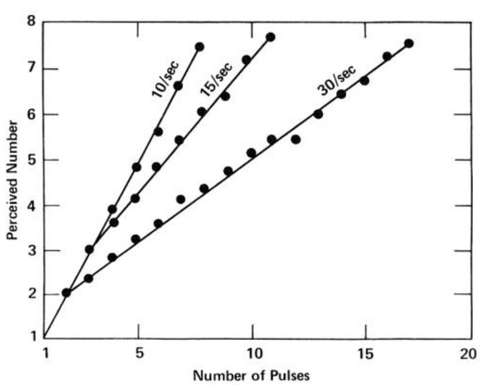
- Decay: 200ms

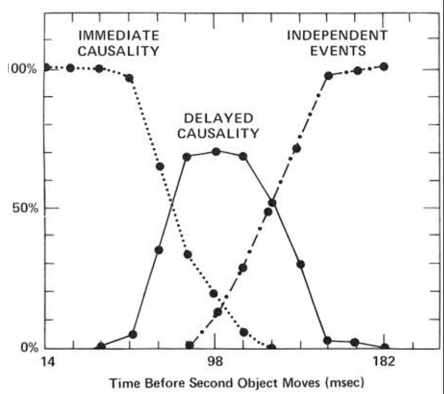


Perceptual Processor

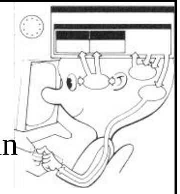


- Cycle time
  - Quantum experience: 100ms
    - *Percept fusion*
    - *Causality*

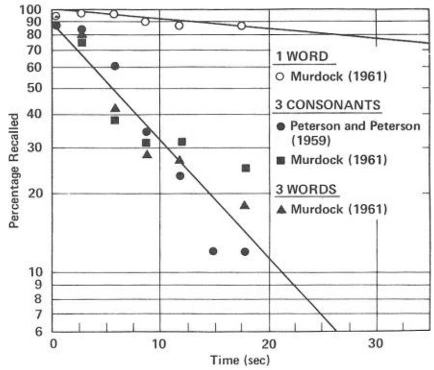




Working Memory



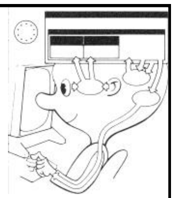
- Why is "UMDMSNBC" much easier to remember than "MCUSMBDN"?
- Access in chunks
  - Task dependent construct
  - 7 +/- 2 (Miller)
- Decay
  - Content dependant
  - Limit attention span
  - Half-life ~7 secs



⇒ Reliance on STM is bad

⇒ External cognition tool (like paper) is good.

Long term memory

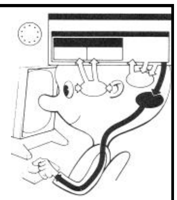


- Very large capacity
  - Semantic encoding
- Associative access
  - Fast read: 70ms
  - Expensive write: 10s
    - *Several rehearsals and/or recalls required,*
- Context at the time of acquisition key for retrieval
- Noisy
- What is half-life?
  - Infinity!
  - But retrieval might fail if:
    - *No associations can be found*
    - *Associations to other items interfere*

Cognitive Processor

- Cycle time: 70ms
  - Can be modulated
- Typical matching time
  - Digits: 33ms
  - Colors: 38ms
  - Geometry: 50ms…
- Fundamentally serial
  - One locus of attention at a time
    - *Eastern 401, December 1972*
      - Crew focused on checking the landing gear indicator bulb,
      - Meanwhile the aircraft is loosing altitude (horn, warning indicator…),
      - Aircraft crashed in the Everglades
      - see "The Humane Interface" by Raskin, p25
    - *Northwestern flight last month – pilots overshot airport by 150 miles*
    - *But what about driving and talking?*

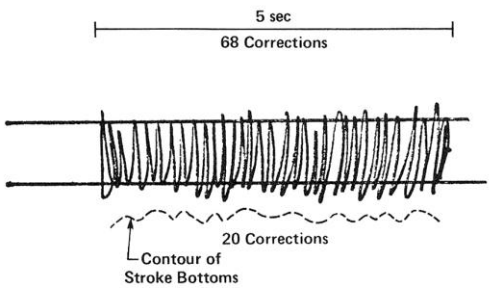
Motor Processor



- Receive input from the cognitive processor
- Execute motor programs
  - Pianist: up to 16 finger movements per second
  - Point of no-return for muscle action
- Draw up and down as rapidly as possible between two lines for 5 seconds so you can count individual strokes.

Motor Processor

- Receive input from the cognitive processor
- Execute motor programs
  - Pianist: up to 16 finger movements per second
  - Point of no-return for muscle action



Human Information Processor - Summary

Perceptual Processor

Cycle time: τ_P = 100 ms

Cognitive Processor

Cycle time: τ_C = 70 ms

Motor Processor

Cycle time: τ_M = 70 ms

It is a model – understandable by computer scientists

Simplistic, but predictive

Does not describe actual underlying mechanisms

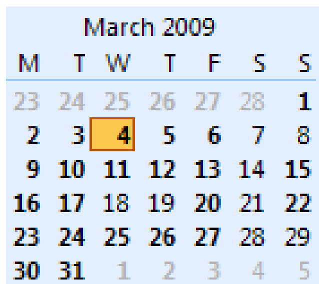
Put it together: Do two letters have same name?

a A

Perceive first letter
Start clock
Perceive second letter (τ_P)
Recognize letter (τ_C)
Match (τ_C)
Initiate response (τ_C)
Respond (τ_M)

=> $\tau_P + 3\tau_C + \tau_M = 100 + 3*70 + 70 = 380ms$

You try it



How many days are in this month:

Perceive calendar (τ_P)
Recognize structure (τ_C)
Decide to move eye to last day (τ_C)
Move eye to last number (τ_M)
Perceive number (τ_P)
Recognize number(τ_C)
Initiate response (τ_C)
Respond (τ_M)

=> $3\tau_P + 3\tau_C + 2\tau_M$ = *3*100 + 3*70 + 2*70 = 650ms*

Eye movement

- Eye movement time
  - 230 ms [70 – 700ms]
- If you move one saccade per phrase while reading, and one phrase is 2.5 words, then (60 sec/min x 2.5 words/saccade) / (.230 sec/saccade + .170 sec) = 375 word / minute

⇒ So, speed readers (2,500 words / minute) don't see all words

Putting it together: Reading Speed

How can you speed up reading?

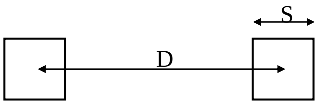
⇒ RSVP

$\tau p + \tau c$ = 100ms + 70ms

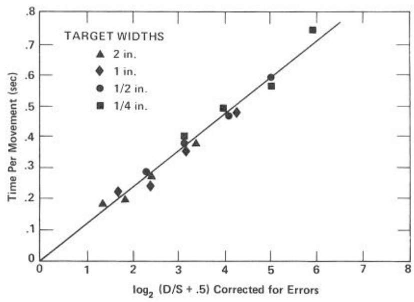
(2.5 word / phrase) x (60 sec / min) / (.170 sec / phrase)

= 882 word / minute

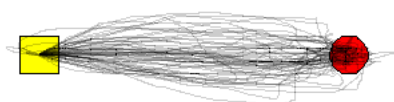
Put it together: Fitts' law (tapping task)



$$T = I_M \log_2(D/S + 0.5)$$



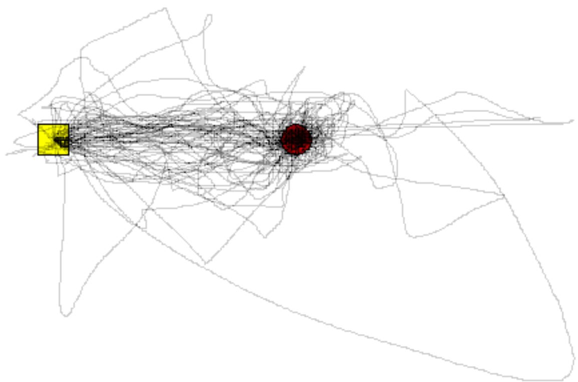
Implications: Fitts' Law



All paths taken by adult participants to click on a 32 pixel target at a distance of 256 pixels.

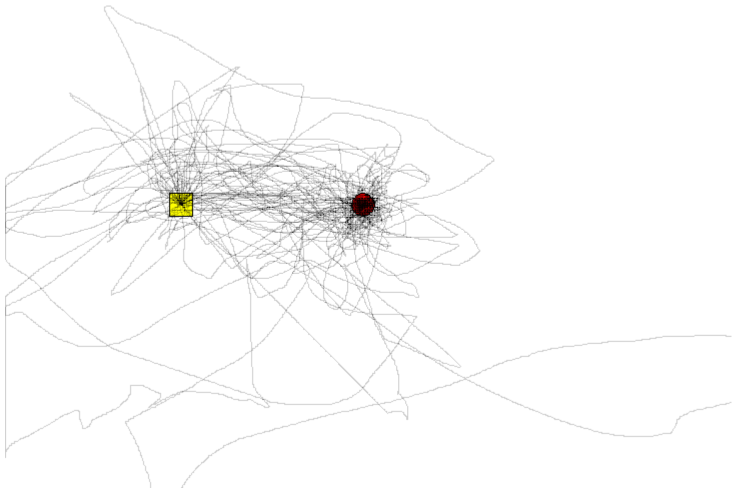
Hourcade, J. P., Bederson, B. B., Druin, A., & Guimbretière, F. (2004) **Accuracy, Target Reentry and Fitts' Law: Performance of Preschool Children Using Mice**, *Transactions on Computer-Human Interaction*, New York: ACM, 11 (4), pp. 357-386.

Implication: Fitts' Law



All paths taken by 5 year-old participants to click on a 32 pixel target at a distance of 256 pixels.

Implication: Fitts' Law



All paths taken by 4 year-old participants to click on a 32 pixel target at a distance of 256 pixels.

Implication: Fitts' Law

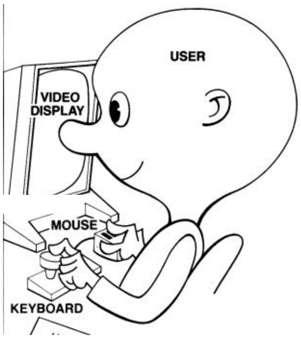
- Location of buttons on screen
  - Windows vs. Mac
  - Borders around buttons
  - Pull-down menus vs pie menus
  - What are the best locations on the screen?
- Size/distance trade-off. Which is faster?

Fitts Law Questions

- http://www.asktog.com/columns/022DesignedToGiveFitts.html

Keystroke Level Model (KLM)

- Fitts' Law focuses on movement only
- Next level of model is KLM
- Focused on expert user – no selection rules
- Describe the task using the following operators:
  - K: pressing a key or pressing (or releasing) a button
    $t_K = 0.2$ *sec*
  - P: pointing
    $t_P = 1.1$ *sec (without button press)*
  - H: Homing (switching device)
    $t_H = 0.4$ *sec*
  - M: Mentally prepare
    $t_M = 1.35$ *sec*
  - R(t): system response time
    $t_R = t$



How to use KLM

- Encode using all physical operator (K, P, H, D(n,l), R(t))
- Apply KLM rules [0-4] to add M's
- Transform R followed by an M
  - If $t \leq t_M$: R(t) → R(0)
  - If $t_M < t$: R(t) → R($t - t_M$)
- Compute the total time by adding all times
  - Will describe expert user behavior

KLM <u>Heuristics</u> (Raskin p. 77)

0: Insert M
- *In front of all K*
- *In front of all P's selecting a command*

1: Remove M between *fully anticipated* operators
- *PMK → PK*

2: if a string of MKs belong to *cognitive unit* delete all M but first
- *4564.23: MKMKMKMKMKMKMK → MKKKKKKK*

3: if K is a *redundant terminator* then delete M in front of it
- *↵/↵: MKMK → MKK*

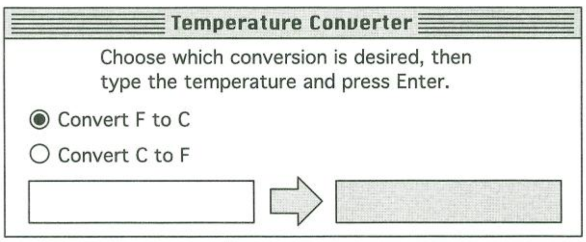
4a: if K terminates a constant string (command name) <u>delete</u> the M in front of it
- *cd↵: MKKMK → MKKK*

4b: if K terminates a variable string (parameter) <u>keep</u> the M in front of it
- *cd class↵: MKKKMKKKKKMK → MKKKMKKKKKMK*

Converting Temperature

("Humane Interface", Raskin)

- Convert 92.5F to Celsius

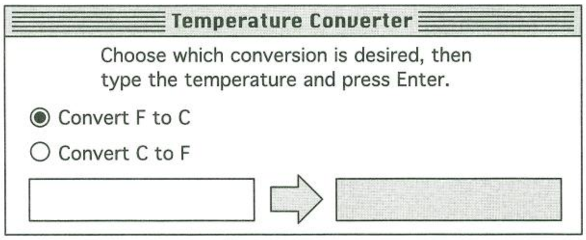


Assume :
- the focus is on the dialog box
- 4 characters for temperature

Converting Temperature

("Humane Interface", Raskin)

- Convert 92.5F to Celsius



Assume :
- the focus is on the dialog box
- 4 characters for temperature

- MKKKKMK (3.7s)
- HMPKHMKKKKMK (7.15s)

=> Average: 5.4s

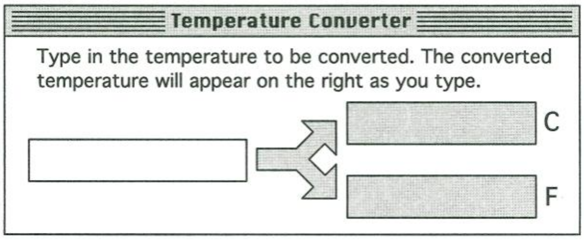
Converting Temperature

- Your design. Can you do better?
- Design interface in pairs
- Perform KLM analysis and report

Try Google:
- 92.5f in c
- 37c in f

Converting temperature: Raskin's Redesign

("Humane Interface", Raskin)

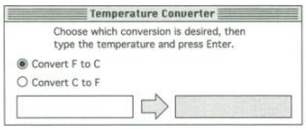


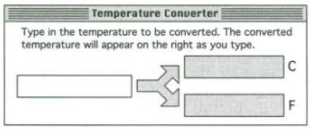
MKKKK => 2.15s

Pros and Cons

What are other factors?

- Learnability
- Accuracy
- Cognitive load





GOMS (Card et al.)

- Family of methods (KLM, CMN-GOMS, NGOMSL, CPM-GOMS)
- For skilled users only, but with multiple solution approaches (models cognitive tasks, not problem-solving)
- Describe the user behavior in term of
  - Goals
    - *A thing to do*
    - *Like a function name – independent of application (i.e., "print page" works in any word processor)*
  - Operators
    - *Elementary perceptual, motor or cognitive actions.*
  - Methods
    - *Sequences of sub-goals and operators that can accomplish a goal.*
    - *Like the body of a function – implements a goal.*
  - Selection rules
    - *Used if several methods are available for a given goal*
    - *Not subjective – GOMS should run like a program*

GOMS: Application and limitations

- Applications
  - CAD system
  - Telephone operator (CPM-GOMS)
  - Text editing with keyboard and mouse (KLM)
- Limitations
  - Skilled users
  - Does not deal with error
  - Does not deal with skill acquisition
  - Does not deal with high level issues (Functionality, workload, Fatigue)
  - Better for relative than absolute timing

Value of KLM/GOMS?

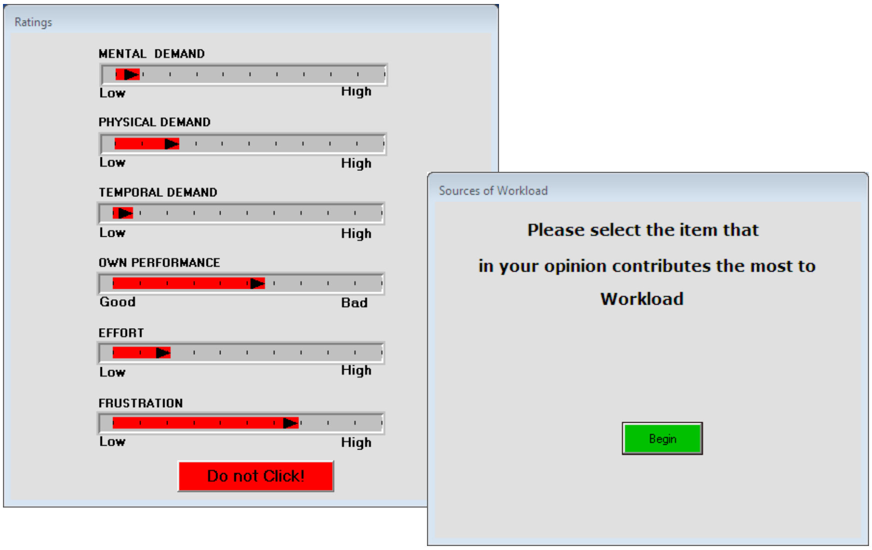
- Possibly good for high value decisions
- Possibly good for making strong argument
- Definitely good for helping designers develop an intuition about works and doesn't and the impact of design decisions on speed

- Look at www.hcibib.org for work on GOMS

HIP Implication: Cognitive Load

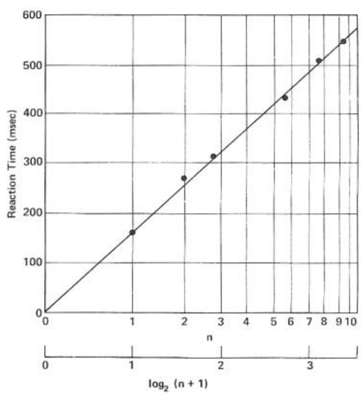
- Interfaces that are fast and accurate may still be bad
- Cognitive load: "amount of cognitive resources needed to perform a task"
- Need to also measure "cognitive load"
  - Subjective self-report measures
  - Performance-based measures such as secondary tasks
  - Physiological measures such as pupillary dilation
- Subject self-report measures most common
  - NASA Task Load Index (TLX)
  - Subjective Workload Assessment Technique (SWAT)
  - Paas scale

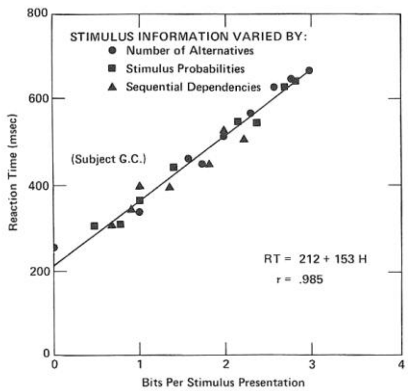
NASA TLX



Hick's law

- Cost of taking a decision: $H = \log_2(n + 1)$
- I.e., People do binary search (when possible)

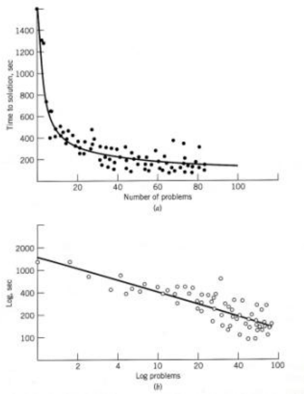




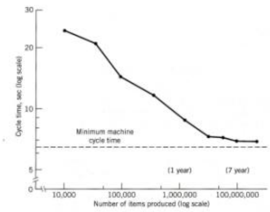
Learning

(“Learning and memory” Anderson)

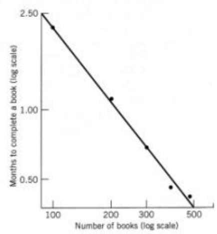
- The time T_n to perform a task on the nth trial follows a power law: $T_n = T_1 n^{-\alpha}$



Problem solving



Manual skill



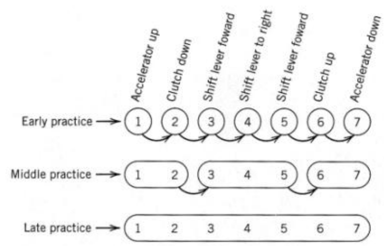
Writing books

Stages of skill acquisition

(“Learning and memory” Anderson)

Example: Using a manual transmission

- Cognitive
  - Verbal representation of knowledge
- Associative
  - Proceduralization
    - *From rehearsal to recognition*
- Autonomous
  - More and more automated
  - Faster and faster
  - No cognitive involvement
    - *Difficult to describe what to do*
  - **The importance of motor program**



Experts

=> Automatized Skills

- Doesn’t engage cognitive system
  - No working memory load
  - Not interruptible
  - Perception goes away as system goes open-loop

Closed Loop vs. Open Loop

Closed Loop

- Feedback from perception through cognitive to motor
- Examples?

Open Loop

- Control is planned in advance and motor executes without perception or cognitive
- Examples?

Implications: confirmation steps

- Pros and Cons?

- See also “The humane interface” Raskin, p23

Implications: dynamic menus

- Pros and Cons