

# Questions?

# Neilsen's heuristics

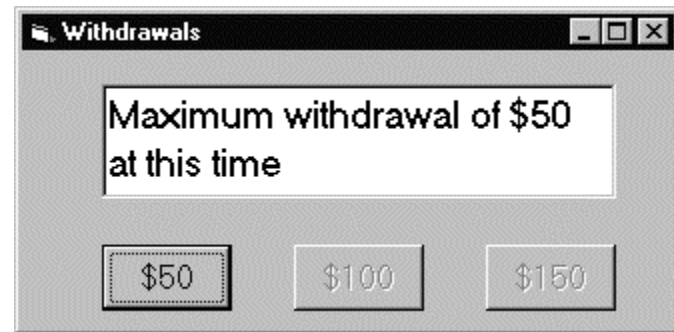
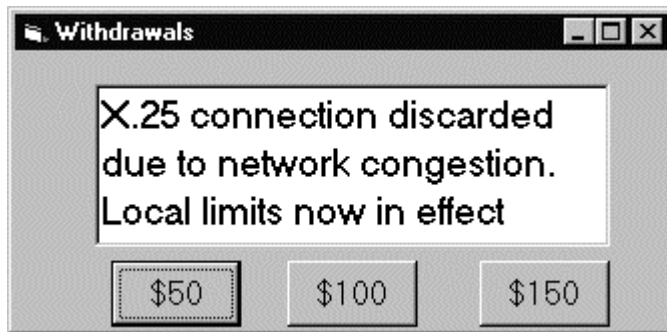
- Simple and natural dialog
- Speak the users' language
- Minimize user memory load
- Consistency
- Feedback
- Clearly marked exits
- Shortcuts
- Prevent errors
- Good error messages
- Provide help and documentation

# Simple and natural dialog

- Present information in natural order
  - See class on graphic design
- Use windows frugally
  - Avoid complex window management
- Remove or hide irrelevant or rarely needed information
  - They compete with important information on screen
    - *Pro: Palm Pilot*
    - *Against: Dynamic menus*
- Use Occam's razor
  - less to learn, to get wrong, to distract...

# Speak the users' language

- Use a language compatible with users' conceptual model
  - Example: withdrawing money at an ATM



- Use meaningful mnemonics, icons and abbreviations

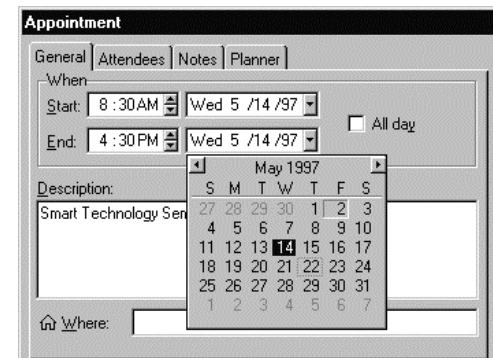


# Minimize user memory load

- Promote recognition over recall
  - Recognition is easier than recall



- Describe expected input clearly
  - Don't allow for incorrect input



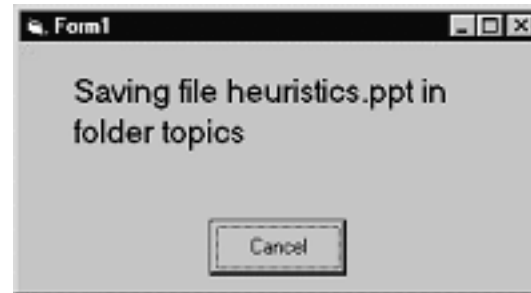
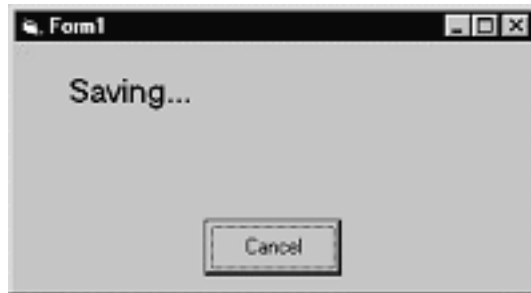
- Create orthogonal command systems
  - Using generic commands that can be applied to all interface objects

# Consistency

- Consistency promotes skills acquisition and/or transfer
- Be consistent in
  - Command design
    - *Same action, same effect in equivalent situations*
  - Graphic design
    - *Input format*
    - *Output format*
  - Flow design
    - *Similar tasks are handled in similar ways*

# Feedback (I)

- Users should always be aware of what is going on
  - So that they can make informed decision
  - Be specific:



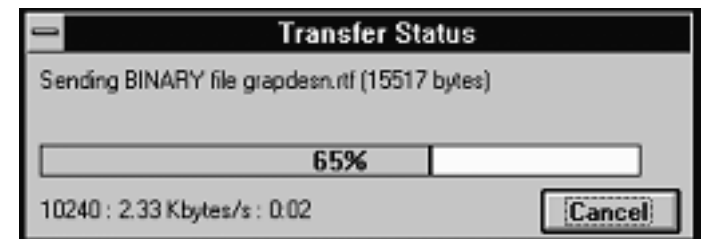
- Provide redundant information



Feedback: Toolbar, cursor, ink

# Feedback (II)

- Different feedback time scales
  - Shall I wait for that task to finished or go for coffee?
    - .1s Causality
    - 1s Delay but user's flow of thought is uninterrupted
    - 10s Difficult to stay focused
    - > 10s User will switch to another task while waiting
- Different techniques
  - Short transaction: hour glass cursor
  - Longer transaction: estimate of time left
    - *An overestimate is always better!*



# Clearly marked exits

- Users don't like to be trapped!

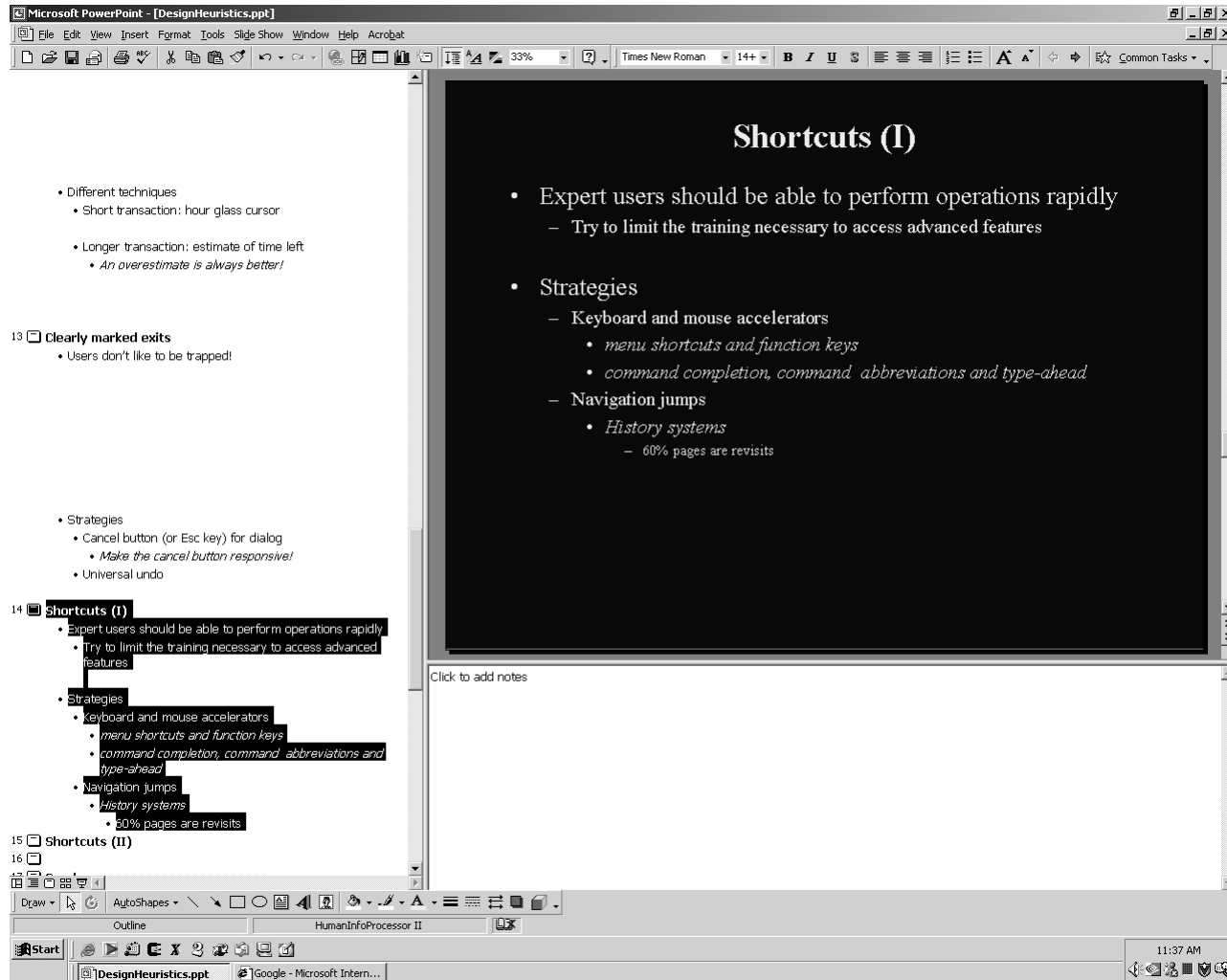


- Strategies
  - Cancel button (or Esc key) for dialog
    - *Make the cancel button responsive!*
  - Universal undo

# Shortcuts (I)

- Expert users should be able to perform operations rapidly
  - Try to limit the training necessary to access advanced features
- Strategies
  - Keyboard and mouse accelerators
    - *menu shortcuts and function keys*
    - *command completion, command abbreviations and type-ahead*
  - Toolbars and tool palettes
    - *Trade screen real estate for rapid access*
  - Navigation jumps
    - *History systems*
      - 60% pages are revisits

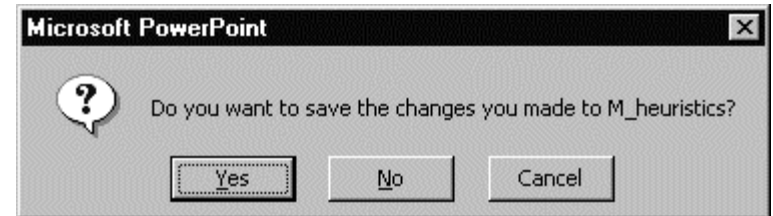
# Shortcuts (II)



Shortcuts: Keyboard accelerators, toolbars, page size scrolling, launch bar...

# Preventing errors

- Error types
  - Mistakes
    - *Conscious decision with unforeseen consequences*
  - Slips
    - *Automatic behaviors kicking in*
      - Drive to the store, end-up in the office
      - Press enter one time to many...
    - *Mode errors*
      - Forget the mode the application is in
    - *Loss of activation*
      - Forget what your goals were



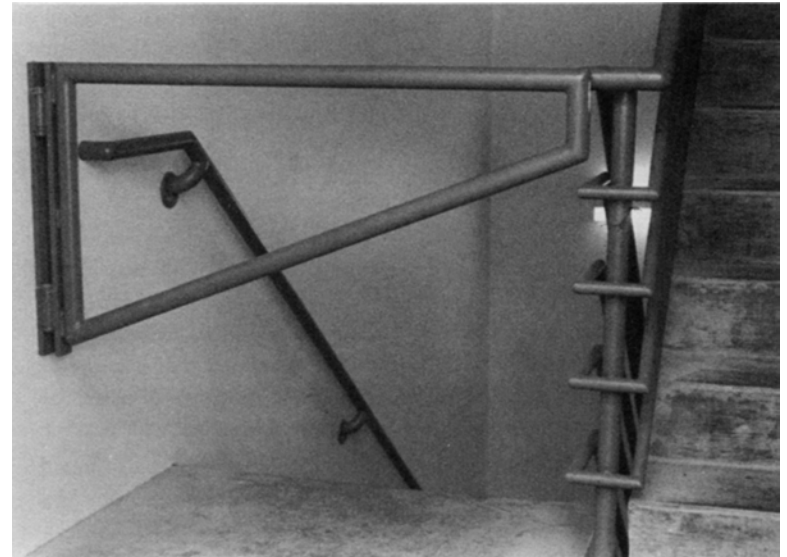
# Designing for slips

One ounce of prevention is worth more than a pound of cure!

- Examples
  - Design modeless interfaces
    - *Use kinesthetic feedback*
  - Instead of confirmations provide undo mechanisms
    - *Dismissing them will become an habit!*
  - Check for reasonable input
    - *Be prepared to handle several formats*
    - *Help user correct their mistakes*
    - *Make entering a incorrect format impossible*
      - See previous calendar example
  - Make the current goal clear
    - *Prevent lost of activations*

# Forcing functions

- Interlock mechanisms
  - Enforce proper sequence
    - *Switching from P to D in a car*
    - *Microwave oven door*
- Lockin mechanisms
  - Prevent unexpected exit
    - *Soft power switches*
- Lockout mechanisms
  - Prevent entering a dangerous situation
    - *Exit stairways*



# Dealing with errors

- People will make errors!
  - You can ignore them
    - *Generally very confusing*
  - You can correct them automatically
    - *Spelling corrector*
    - *But will I trust the system to be right 100%*
  - You can discuss about it
    - *But novice/expert tradeoff*
  - You can try to teach the user what to do
    - *Office assistant*

# Good error messages

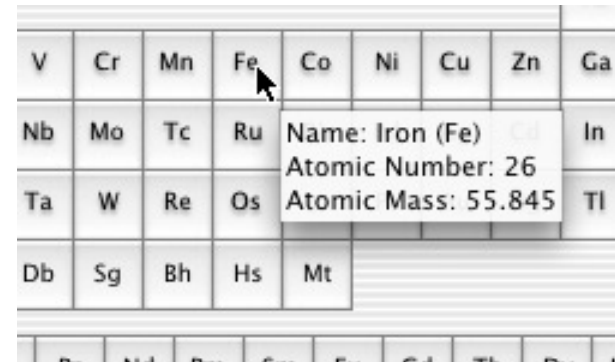
- Provide meaningful error messages
  - Explain the problem in term or user conceptual model
  - Don't make the user feel stupid
  - Offer a way to correct the problem
  - Compare
    - *Error 25: access denied*
    - *Cannot open "chapter 5" because "Microsoft Word" is not installed. Do you want to use Notepad instead?*

# Provide help and documentation

- Providing help is not an excuse for poor design!
  - Saving a couple of line of code or writing several pages of documentation?
  - Users don't like to read manuals
    - *They prefer to learn while making progress toward their goals*
- Most users will stay at the intermediate level
  - Need reminders and a clear learning path
  - Need a quick way to access critical information
    - *Online documentation and good search tool*

# Types of help (I)

- Tutorial and/or getting started manuals
  - Presents the system conceptual model
    - *Basis for successful explorations*
  - Provides on-line tours and demos
    - *Demonstrates basic features*
- Reference manuals
  - Designed with experts in mind
- Reminders
  - Short reference cards, keyboard templates, tooltips...



|    |    |    |    |                                                             |    |    |    |    |
|----|----|----|----|-------------------------------------------------------------|----|----|----|----|
| V  | Cr | Mn | Fe | Co                                                          | Ni | Cu | Zn | Ga |
| Nb | Mo | Tc | Ru | Name: Iron (Fe)<br>Atomic Number: 26<br>Atomic Mass: 55.845 |    |    |    | In |
| Ta | W  | Re | Os |                                                             |    |    |    | Tl |
| Db | Sg | Bh | Hs | Mt                                                          |    |    |    |    |
| Rf | Nd | Pm | Sm | Eu                                                          | Gd | Th | Pa |    |

# Types of help (II)

- Wizards

- Walks user through typical tasks
  - *Users feel they are losing control*
  - *What if I do not have the information requested?*

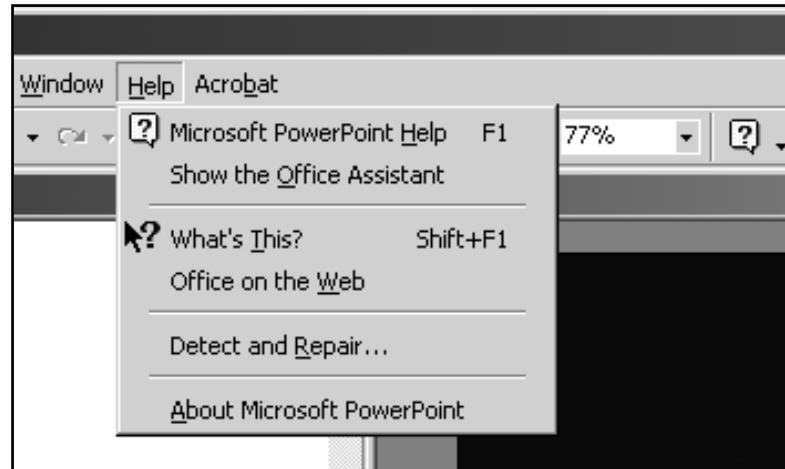


- Tips

- Migration path to learning new features
- Can become boring and tedious

# Types of help (II)

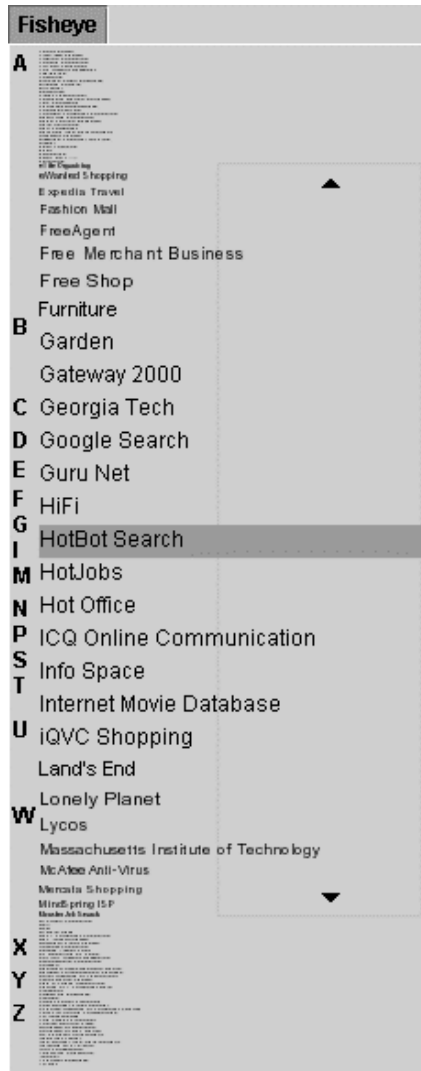
- Context sensitive help



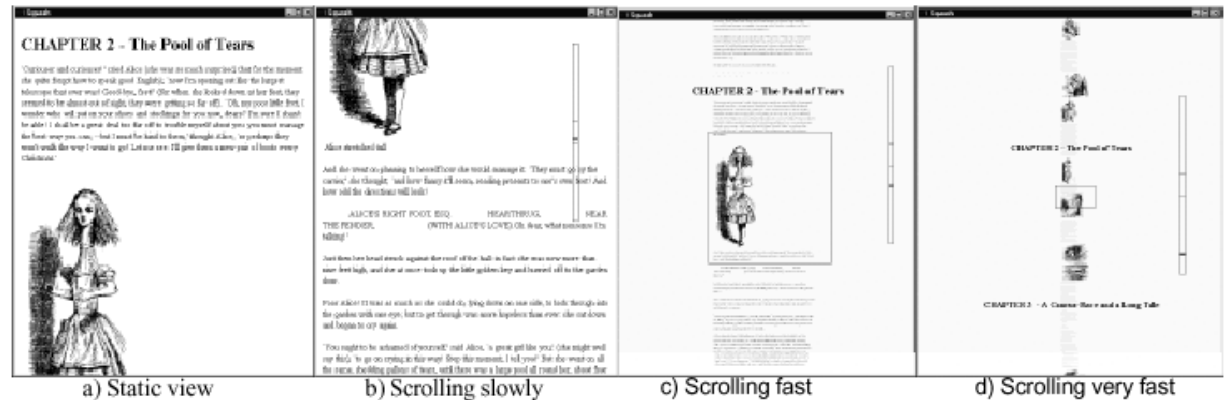
# How to select a movie? (I)

- Picking from a list
  - Standard list (Windows for example)
  - Fisheye Menus [Bederson 2000]
  - Speed dependant automatic zooming [Igarachi 2000]
  - Dynamic query system (like the SpotFire)
- Using text entry

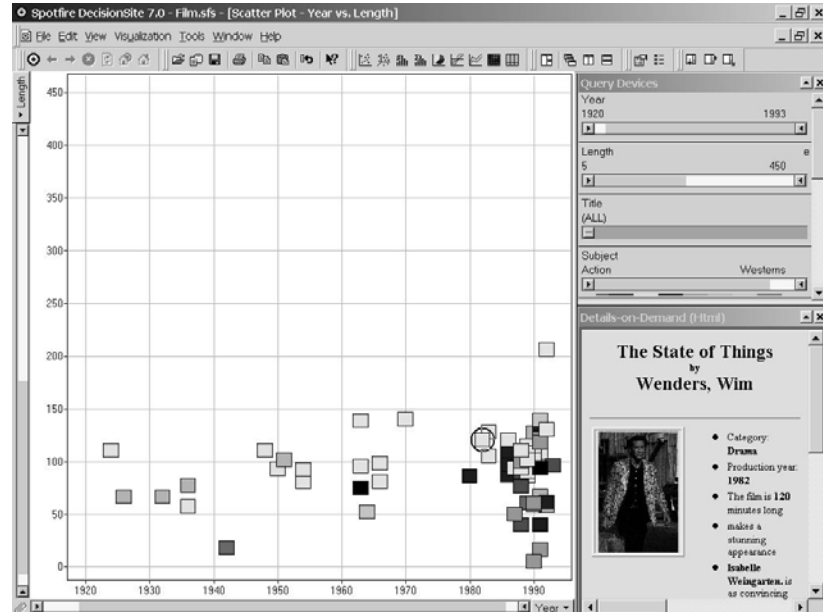
# Picking a movie from a list



Fisheye menu



Speed dependant automatic zooming



SpotFire

# How to select a movie? (II)

- Picking from a list
- Text entry
  - Keyboards
    - *Standard*,
    - *Miniatures*
      - Phones, Blackberry
    - *Virtual on screen*
      - Touch based, game like (selecting each letter with a joystick)
  - Gesture based
    - *UniStroke* [Goldberg 93] *and Graffiti*
    - *QuikWrite* [Perlin 98] *and T-Cube* [Venolia 94]
    - *Dasher* [Ward 2000]
    - *Handwriting recognition*
  - Smart completion and advanced search engines
    - *Google...*

# Keyboards



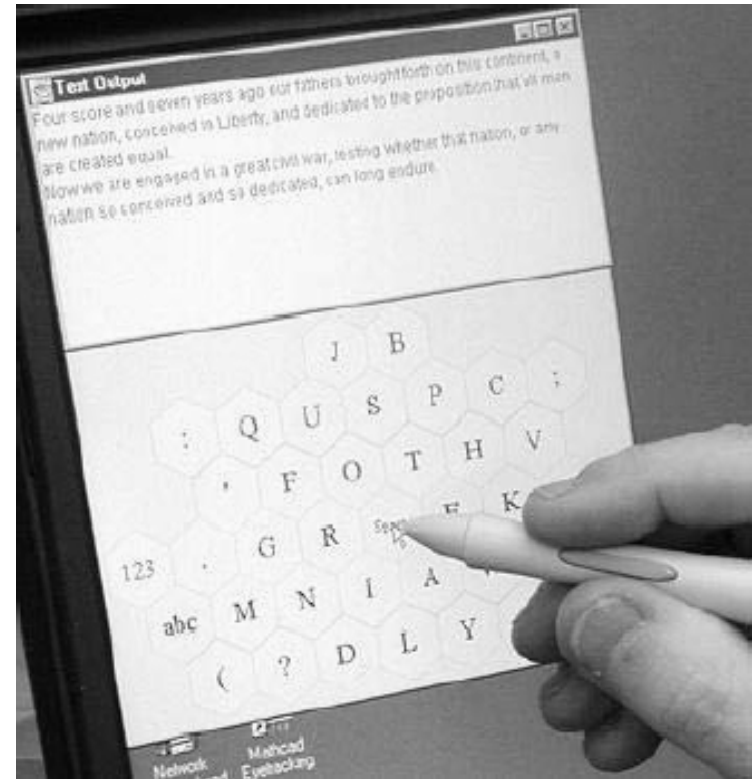
Office keyboard



Cell phone

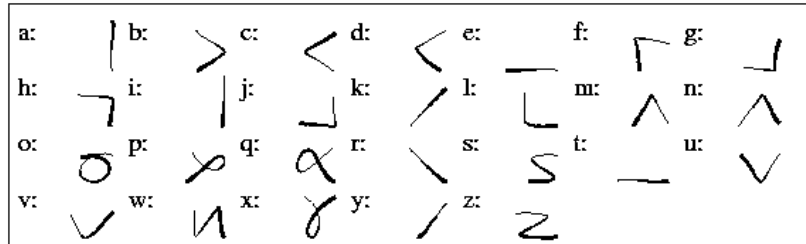


Blackberry

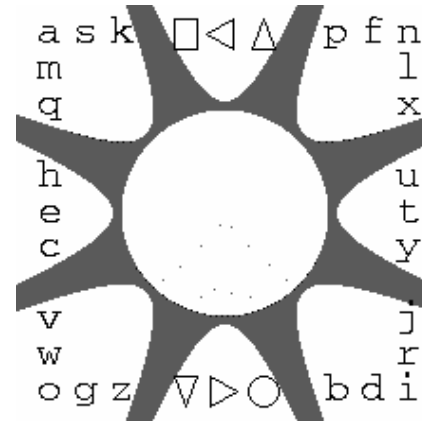


Metropolis keyboard (Zhai, IBM, 2000)

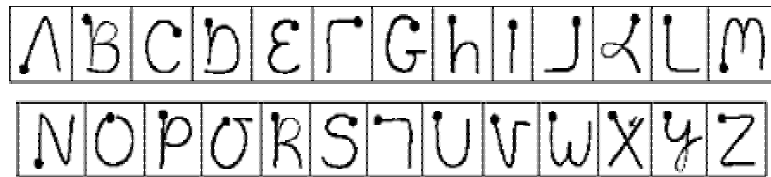
# Gestures based



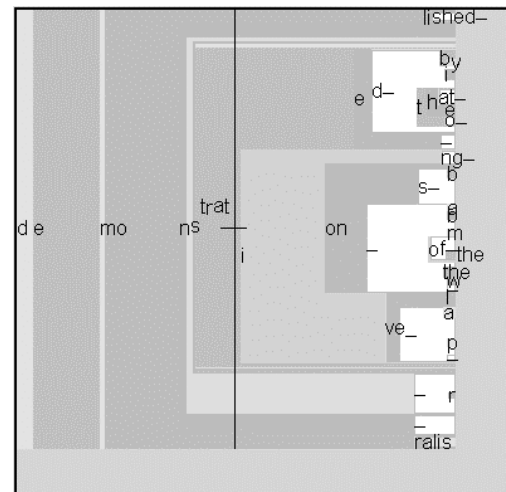
Unistroke alphabet [Goldberg 93]



QuikWriting system [Perlin 98]



Graffiti (Palm OS)



demonstrat

Dasher [Ward 2000]

# How to pick a movie? (III)

- Inside a Blockbuster?
- For my on demand video system?