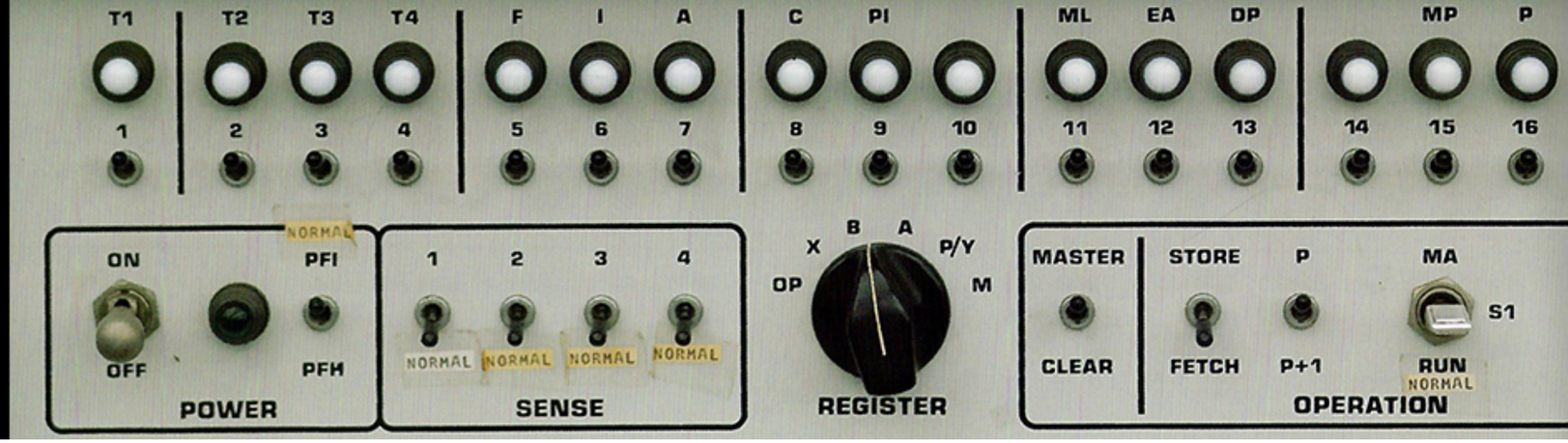
CS160

USER INTERFACE DESIGN

FALL 2018



MODES, METAPHORS, AND INPUT DEVICES

26 SEP 2018

ERIC PAULOS
www.paulos.net

UNIVERSITY OF CALIFORNIA Berkeley


GROUP 11
31
32
33


R

ANNOUNCEMENTS

DESIGN 02: Heuristic Evaluation (due before class TODAY)

PROG 2-B (Due Next Friday 5 Oct) 10 DAYS!!

PROG 2-B Status?

Teams?

Section this week: APIs ++

Midterm: 15 Oct 10:30-noon in Sibley Auditorium

Midterm Review: 12 Oct in Section

FINAL CRITIQUE

04 Dec Tue of RRR Week • FINAL CRITIQUE

Final Presentations in 310 Jacobs

Session 1 • 10am-12pm in 310 Jacobs

Session 2 • 3pm-5pm in 310 Jacobs

05 Dec Wed of RRR Week • PUBLIC SHOWCASE

Public Posters, Demos, and Showcase in 310 Jacobs

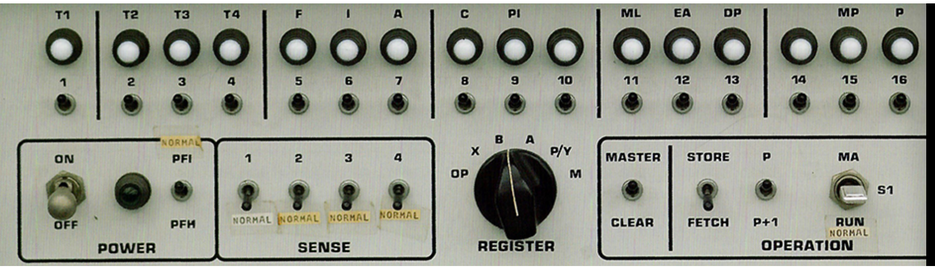
Session 01: 9:45am-11:30am in 310 Jacobs

Session 02: 11:45am-1:30pm in 310 Jacobs

07 Dec Fri of RRR Week • FINAL MATERIALS

Final materials due on 7 Dec (11:59pm)


9:42 AM
9
Text
Calendar
Photos
Camera
YouTube
Stocks
Maps
Weather
Clock
Calculator
Notes
Settings
iTunes

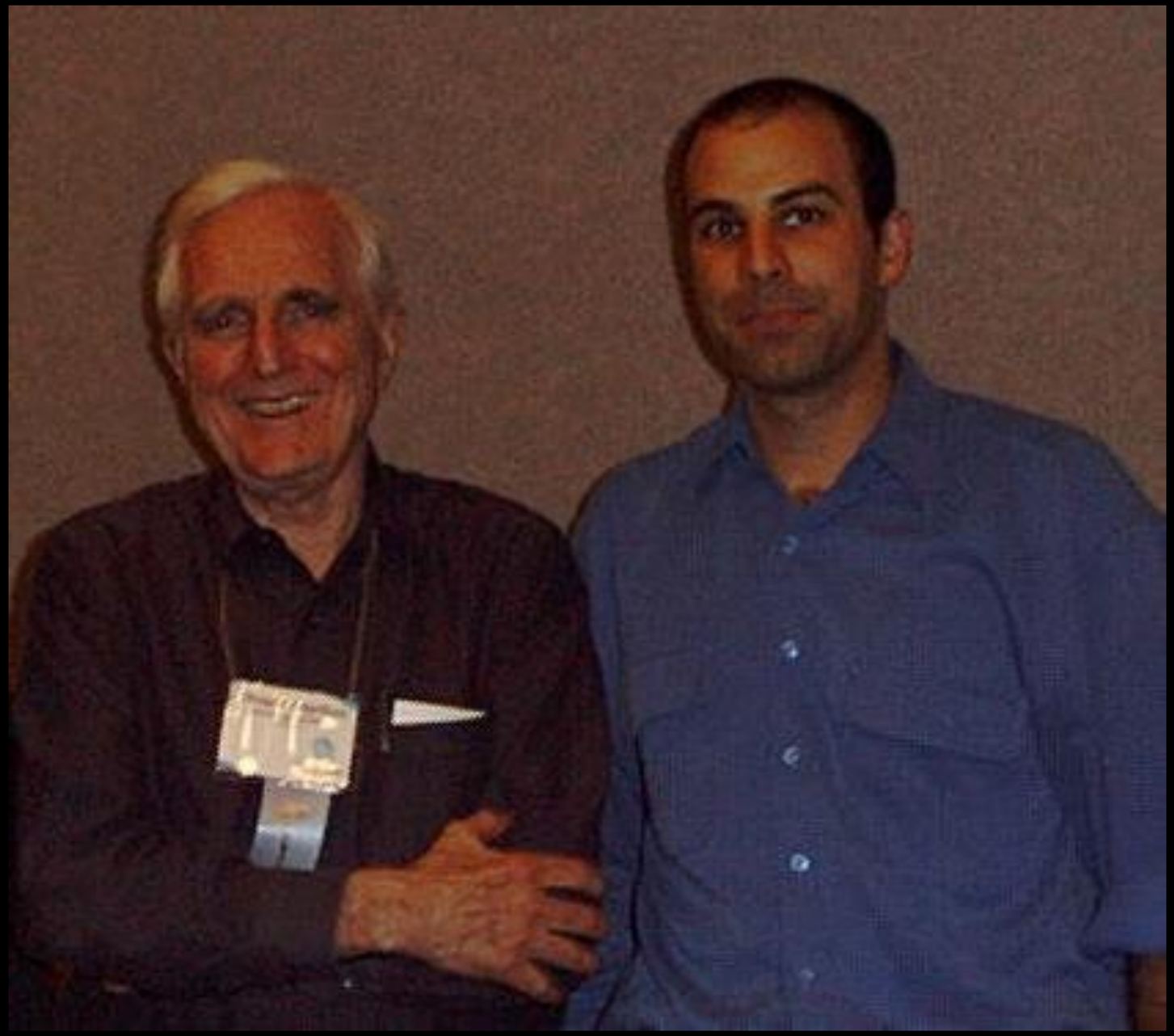


METAPHOR IN USER INTERFACES

THE MOTHER OF ALL DEMOS

Doug Engelbart's December 9, 1968, computer demonstration at the Fall Joint Computer Conference in San Francisco. The 90-minute presentation essentially demonstrated almost all the fundamental elements of modern personal computing

- windows
- hypertext
- graphics
- video conferencing
- the computer mouse
- word processing
- dynamic file linking
- revision control
- collaborative real-time editor



Your Professor (as a PhD student) and Doug Engelbart in 1998

December 9, 1968: The Demo

The Basics

Control Devices

Real-Time Collaboration

1968

nearly 50 years before...

windows
hypertext
video conferencing
the computer mouse
word processing
dynamic file linking
revision control
collaborative real-time editor

METAPHOR

Definition

The transference of the relation between one set of objects to another set for the purpose of brief explanation

Lakoff & Johnson

“...the way we think, what we experience, and what we do every day is very much a matter of metaphor."

in our language & thinking - “argument is war”

...he attacked every weak point
...criticisms right on target
...if you use that strategy

Metaphors can highlight some features, suppress others

INTERFACE METAPHORS

Purpose

Leverages knowledge of familiar, concrete objects/experiences

Transfer this knowledge to abstract tasks and concepts

Problem

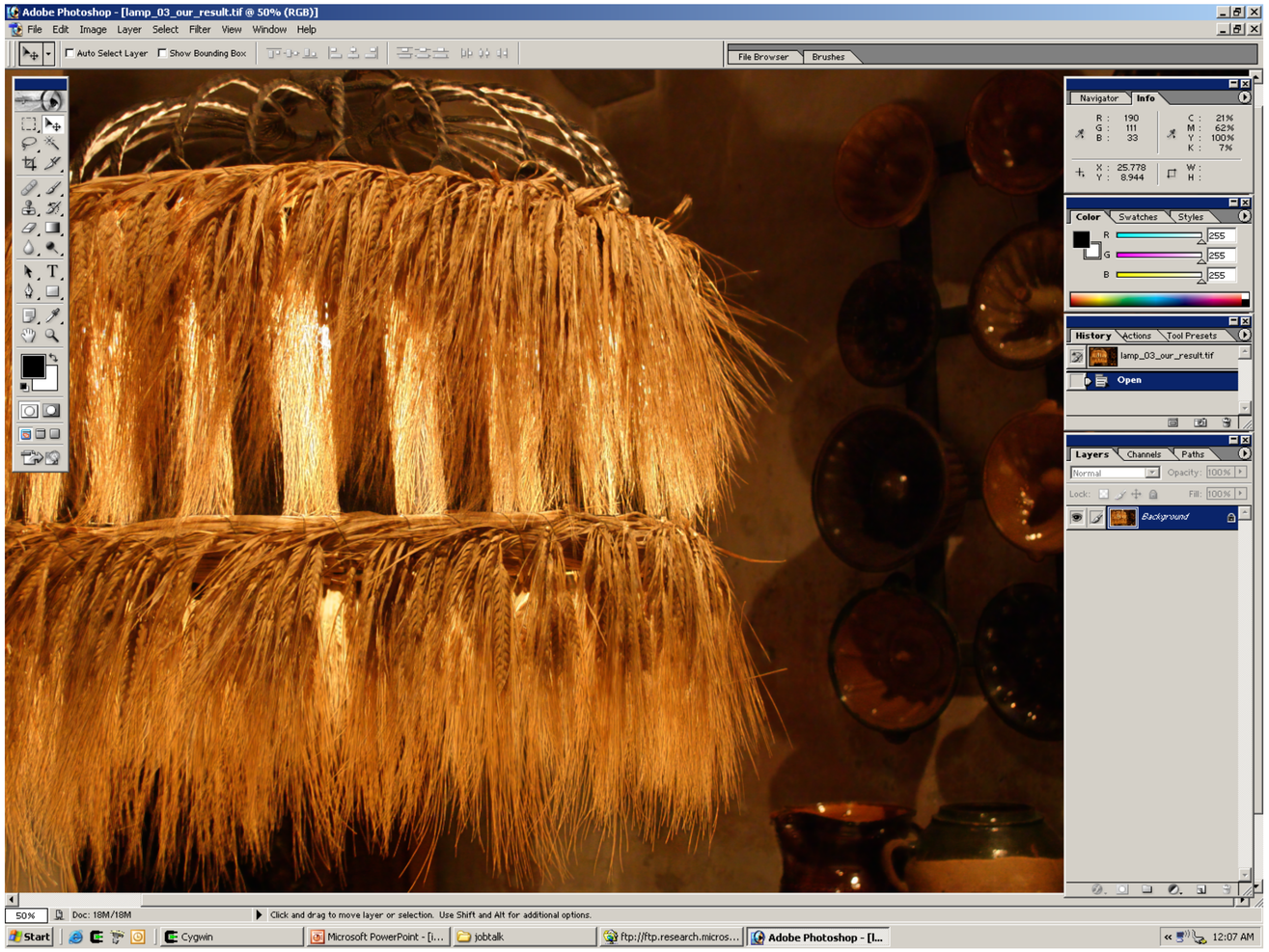
Inaccurate or naive conceptual model of the system

A presentation tool

is like

an slide projector

THE PAINTING METAPHOR



THE DESKTOP METAPHOR

Started at Xerox PARC

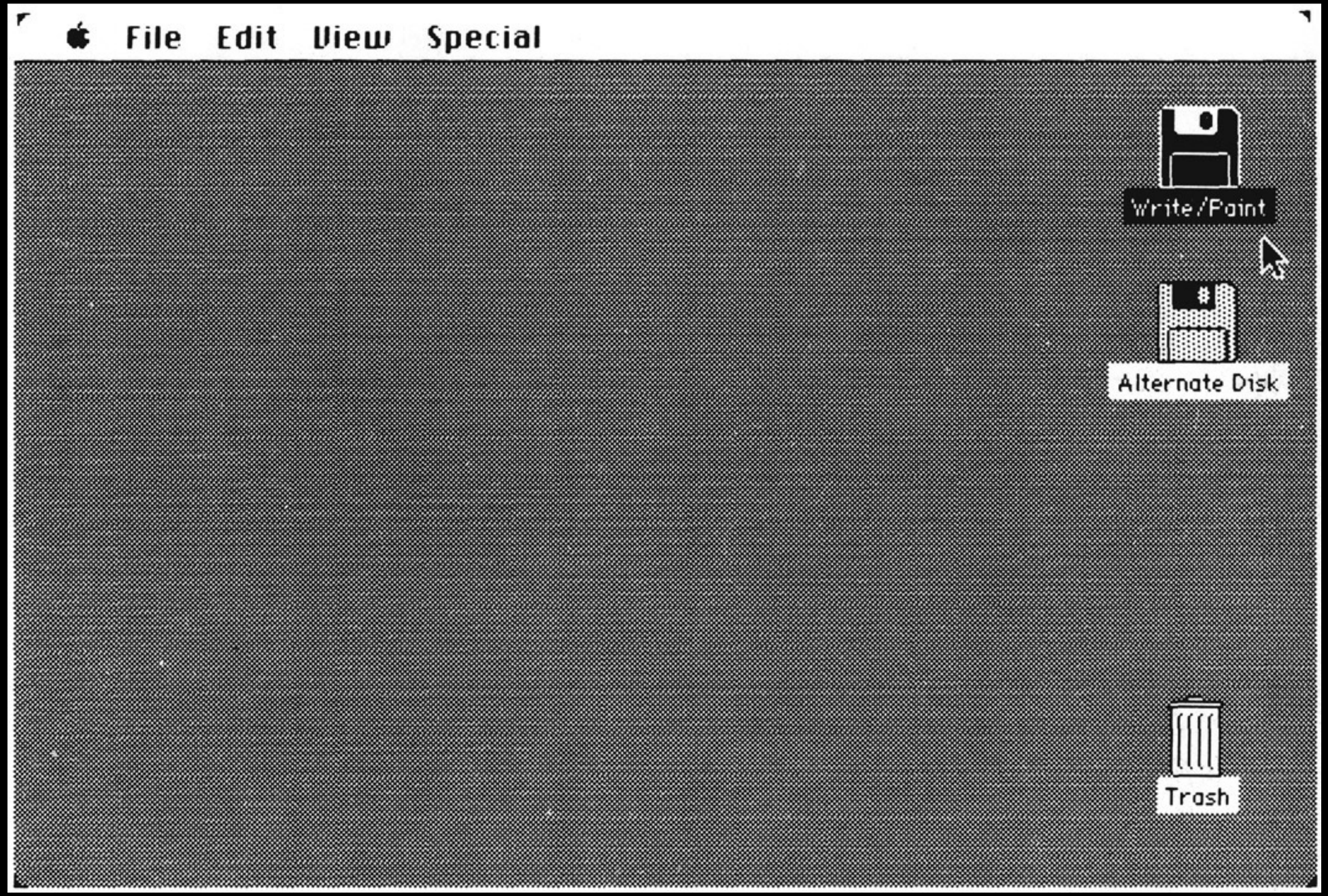
Xerox Star

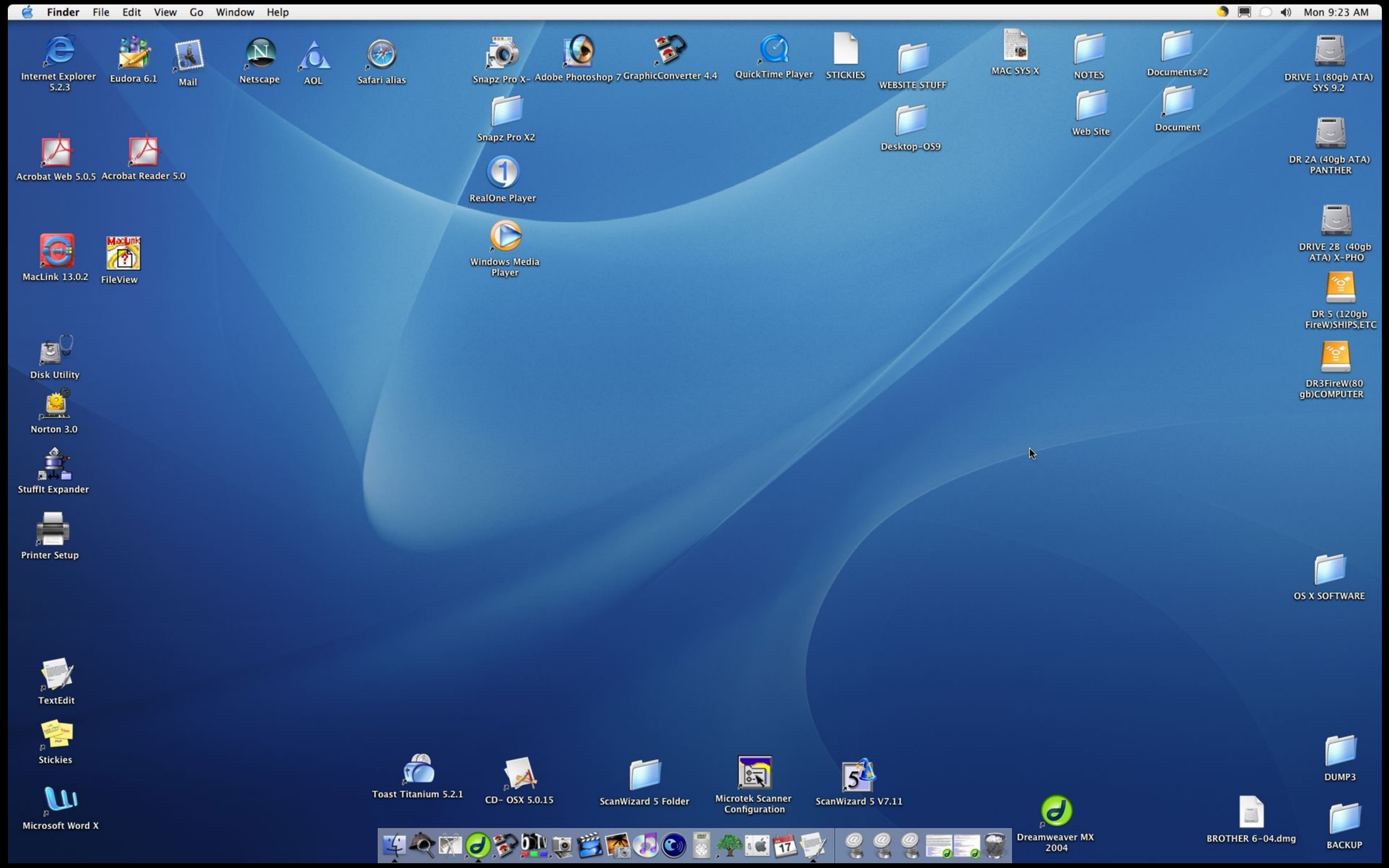
Bitmapped screens made it possible

Not meant to be a real desktop

Organize information the way people use information

Allow windows to overlap – make screen act as if ob


File Edit View Special
Write/Paint
Alternate Disk
Trash


Finder File Edit View Go Window Help
Mon 9:23 AM
Internet Explorer 5.2.3
Eudora 6.1
Mail
Netscape
AOL
Safari alias
Snapz Pro X-
Adobe Photoshop 7
GraphicConverter 4.4
QuickTime Player
STICKIES
WEBSITE STUFF
MAC SYS X
NOTES
Documents#2
DRIVE 1 (80gb ATA) SYS 9.2
Snapz Pro X2
Desktop-OS9
Web Site
Document
Acrobat Web 5.0.5
Acrobat Reader 5.0
RealOne Player
DR 2A (40gb ATA) PANTHER
MacLink 13.0.2
FileView
Windows Media Player
DRIVE 2B (40gb ATA) X-PHO
DR 5 (120gb FireW)SHIPS,ETC
DR3FireW(80 gb)COMPUTER
Disk Utility
Norton 3.0
StuffIt Expander
Printer Setup
OS X SOFTWARE
TextEdit
Stickies
Microsoft Word X
Toast Titanium 5.2.1
CD- OSX 5.0.15
ScanWizard 5 Folder
Microtek Scanner Configuration
ScanWizard 5 V7.11
Dreamweaver MX 2004
BROTHER 6-04.dmg
DUMP3
BACKUP

MICROSOFT BOB'S DESKTOP METAPHOR

BOB'S "LIVING ROOM" METAPHOR

3D DESKTOPS

Robertson 2000

Sun's Looking Glass


Navigate Floor Plan
N
5
L
GOOGLE ART PROJECT
Google
© 2010 Google
Report a problem

VIRTUAL ASSISTANT METAPHOR

METAPHOR CAVEATS

METAPHOR CAVEATS

Too limited

The metaphor restricts interface possibilities

Too powerful

The metaphor implies the system can do things it can’t

Too literal or cute

Makes it difficult to understand abstract concept

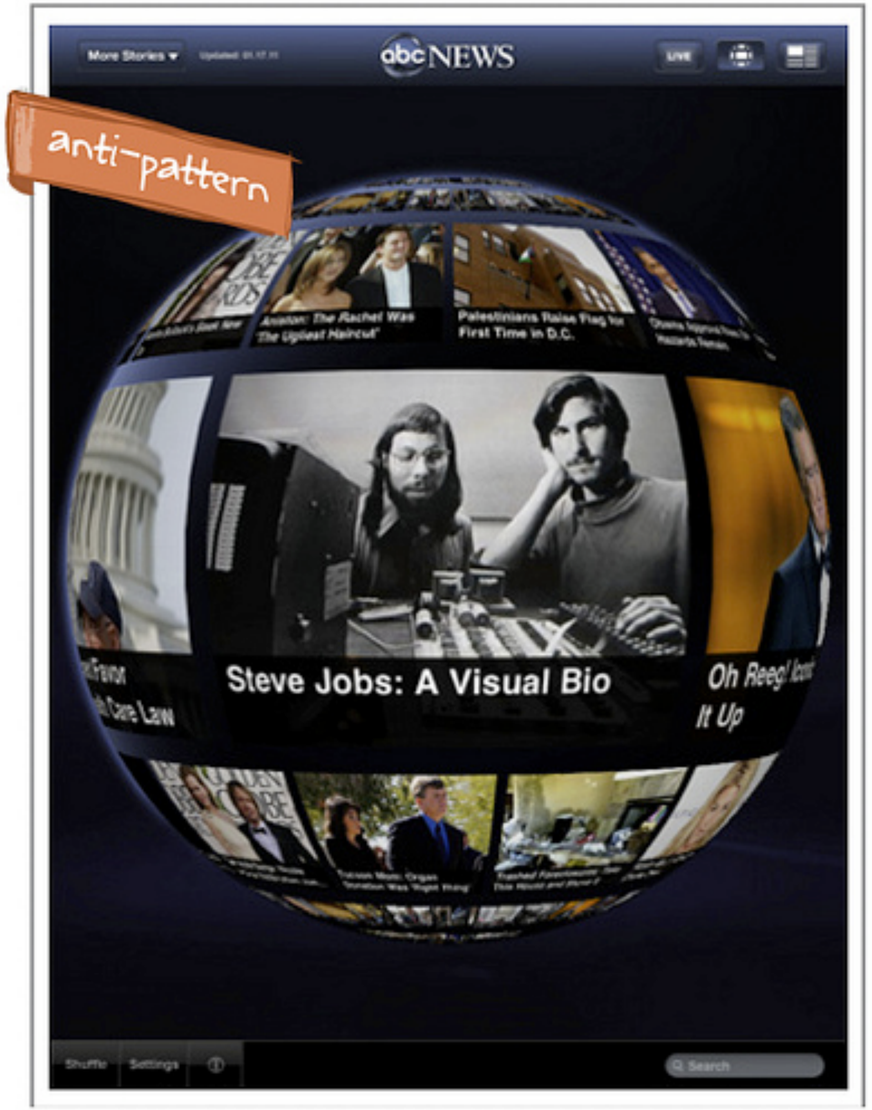
Mismatched

The metaphor conveys the wrong meaning

MISMATCHED METAPHORS

What is being controlled here?

MISMATCHED METAPHORS



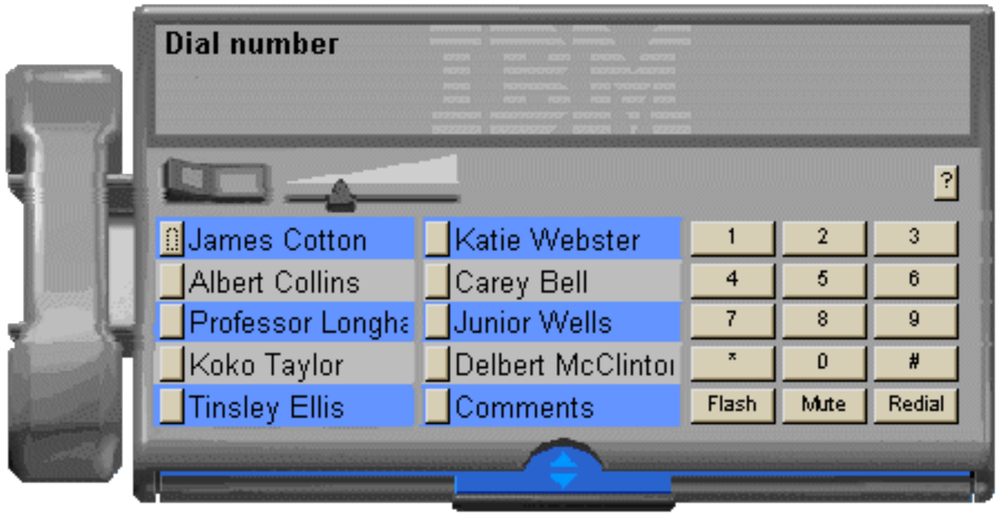
MISUSED METAPHORS

Direct translations

Software music player that requires turning volume knob with mouse

Software telephone that requires the user to dial a number by clicking on a simulated keypad

Airline web site that simulates a ticket counter!



Southwest Airlines Home Gate

The Home of Southwest Airlines on the World Wide Web

Southwest Airlines Home Gate

The Home of Southwest Airlines on the World Wide Web

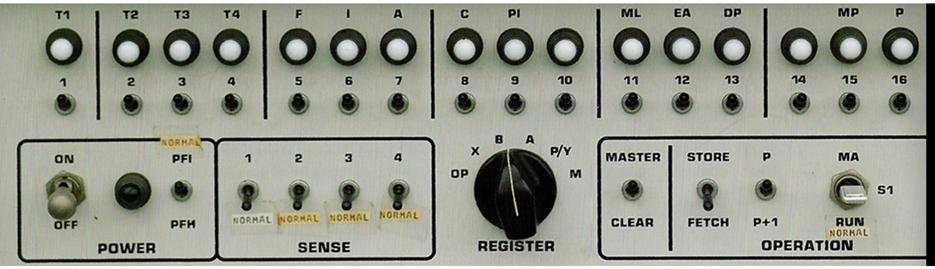
GUIDELINES FOR DESIGN

Good Metaphors

Capture essential elements of the event / world

Deliberately leave out / mute the irrelevant

Appropriate for user, task, and interpretation

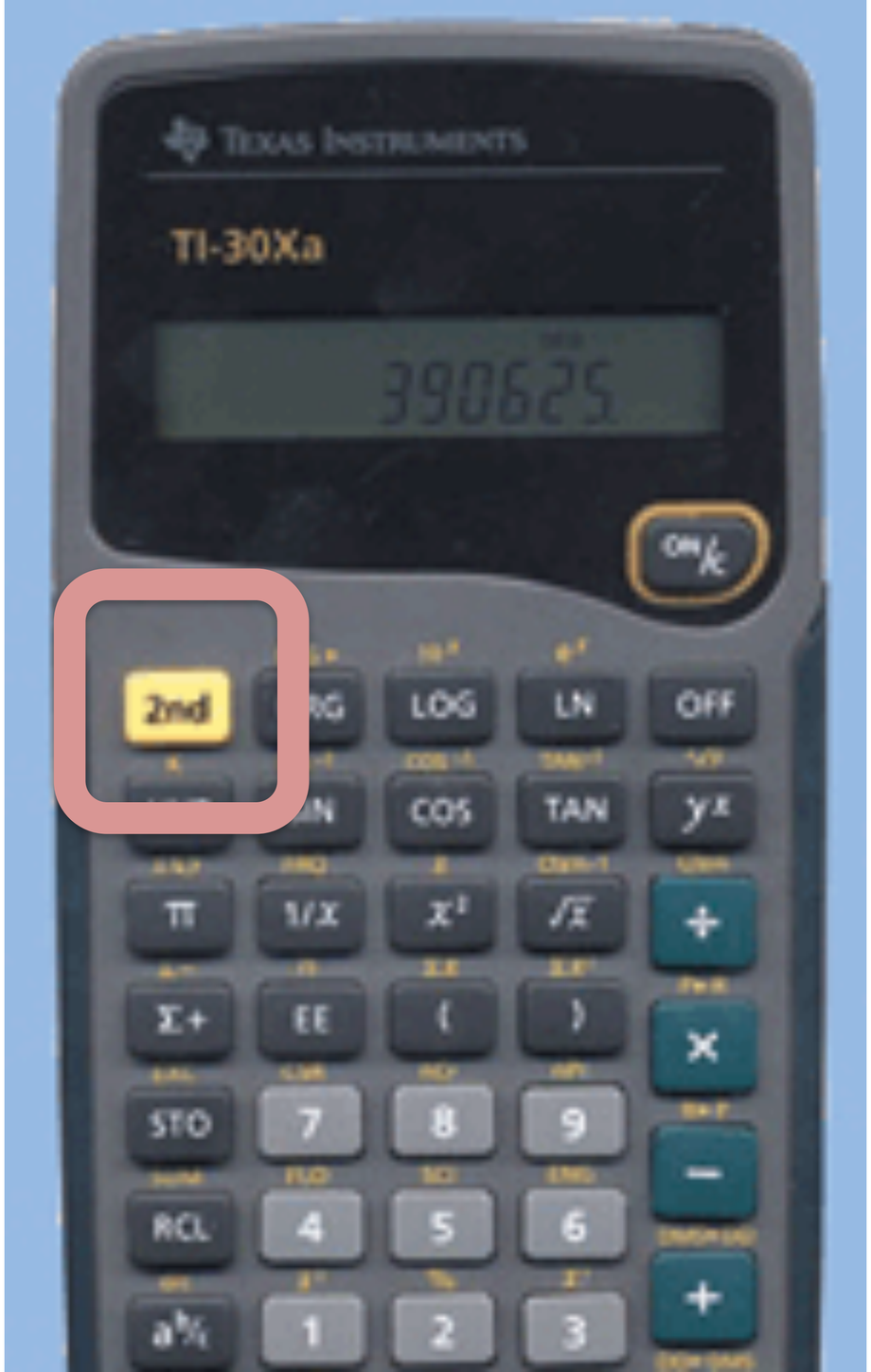

T1
T2
T3
T4
F
I
A
C
PI
ML
EA
DP
MP
P
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
NORMAL
ON
PFI
OFF
PFH
POWER
1
2
3
4
NORMAL
NORMAL
NORMAL
NORMAL
SENSE
X
B
A
P/Y
OP
M
REGISTER
MASTER
CLEAR
STORE
FETCH
P
P+1
MA
S1
RUN
NORMAL
OPERATION

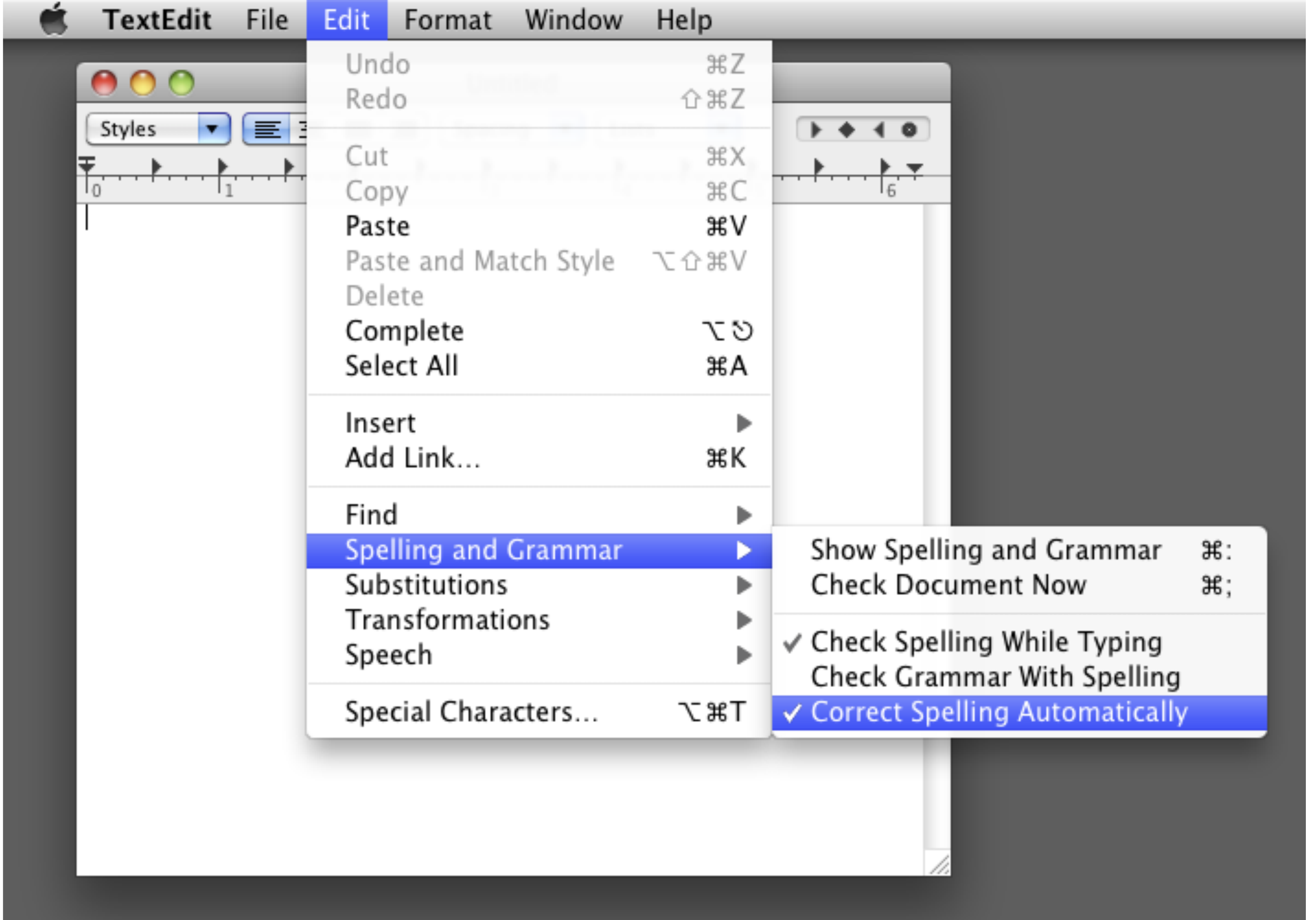
MODES

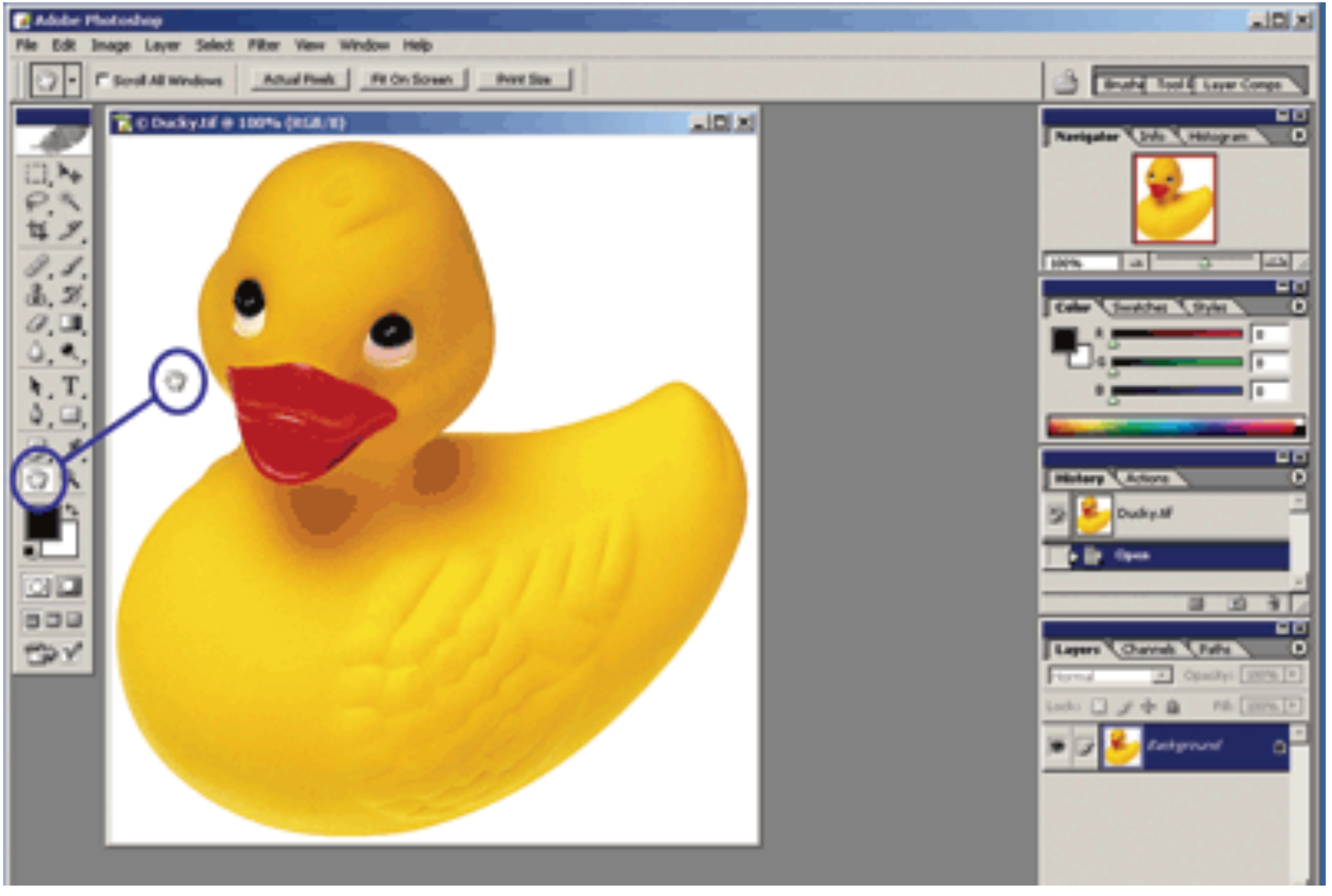
MODES: DEFINITION

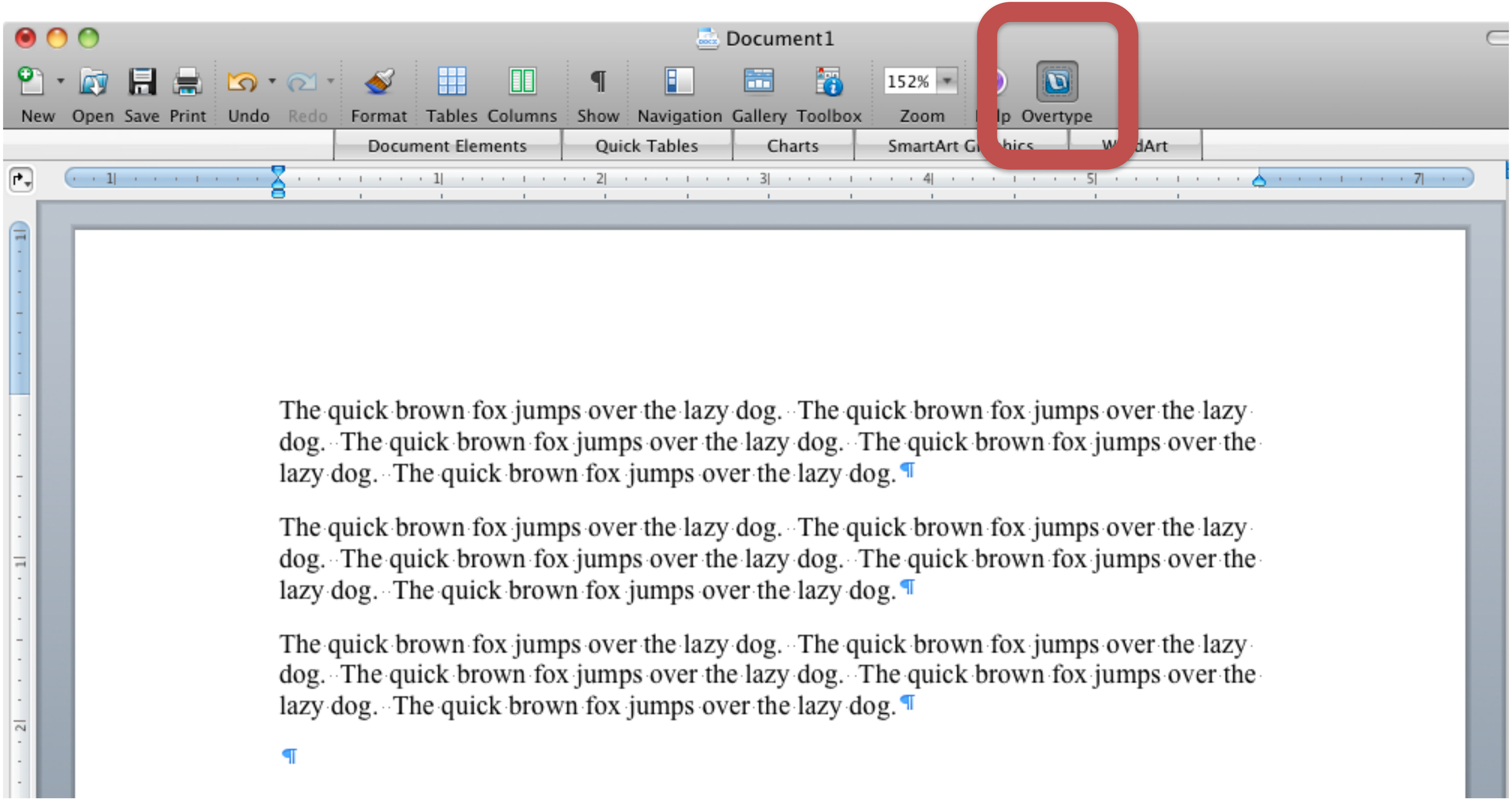
The same user actions have different effects in different situations.

MODES: EXAMPLES


Texas Instruments
TI-30Xa
390625
ON/C
2nd
LOG
LN
OFF
COS
TAN
y^x
π
1/x
x²
√x
÷
Σ+
EE
(
)
×
STO
7
8
9
−
RCL
4
5
6
+
a b/c
1
2
3


TextEdit
File
Edit
Format
Window
Help
Undo ⌘Z
Redo ⇧⌘Z
Cut ⌘X
Copy ⌘C
Paste ⌘V
Paste and Match Style ⌥⇧⌘V
Delete
Complete ⌥
Select All ⌘A
Insert
Add Link... ⌘K
Find
Spelling and Grammar
Substitutions
Transformations
Speech
Special Characters... ⌥⌘T
Show Spelling and Grammar ⌘:
Check Document Now ⌘;
Check Spelling While Typing
Check Grammar With Spelling
Correct Spelling Automatically
Styles


Adobe Photoshop
File Edit Image Layer Select Filter View Window Help
Scroll All Windows
Actual Pixels
Fit On Screen
Print Size
Navigator
Info
Histogram
Color
Swatches
Styles
History
Actions
Ducky.tif
Open
Layers
Channels
Paths
Background


Document1
New
Open
Save
Print
Undo
Redo
Format
Tables
Columns
Show
Navigation
Gallery
Toolbox
152%
Zoom
Overtype
Document Elements
Quick Tables
Charts
The quick brown fox jumps over the lazy dog. The quick brown fox jumps over the lazy dog. The quick brown fox jumps over the lazy dog. The quick brown fox jumps over the lazy dog. The quick brown fox jumps over the lazy dog.
The quick brown fox jumps over the lazy dog. The quick brown fox jumps over the lazy dog. The quick brown fox jumps over the lazy dog. The quick brown fox jumps over the lazy dog. The quick brown fox jumps over the lazy dog.
The quick brown fox jumps over the lazy dog. The quick brown fox jumps over the lazy dog. The quick brown fox jumps over the lazy dog. The quick brown fox jumps over the lazy dog. The quick brown fox jumps over the lazy dog.

USING MODES IN INTERFACES

When are they useful?

Temporarily restrict users' actions

When logical and clearly visible and easily switchable

Drawing with paintbrush vs. pencil

Why can they be problematic?

Big memory burden

Source of many serious errors

How can these problems be fixed?

Don't use modes – redesign system to be modeless

Redundantly visible

REDESIGNING TO AVOID MODES

Setting the time on a clock

Modal

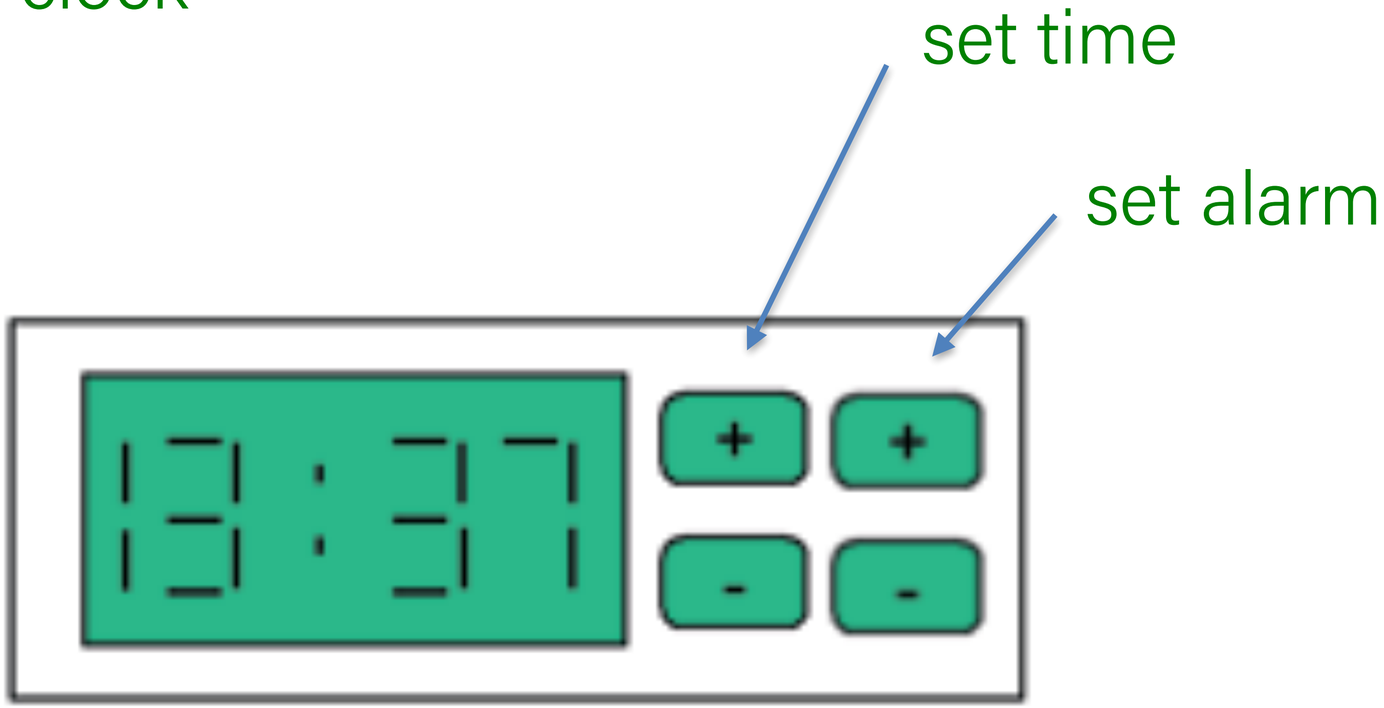
REDESIGNING TO AVOID MODES

Setting the time on a clock

Modeless

REDESIGNING TO AVOID MODES

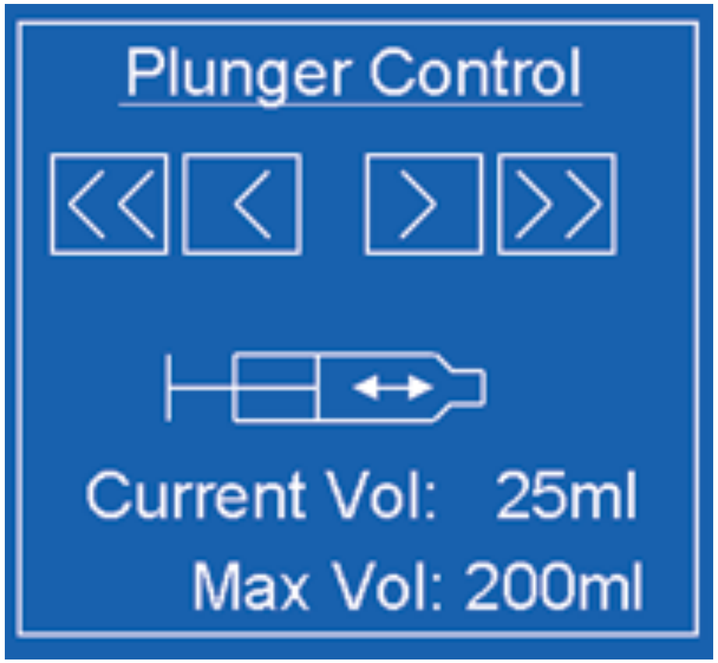
Setting the time on a clock



Modeless

MODES ARE SOMETIMES GOOD

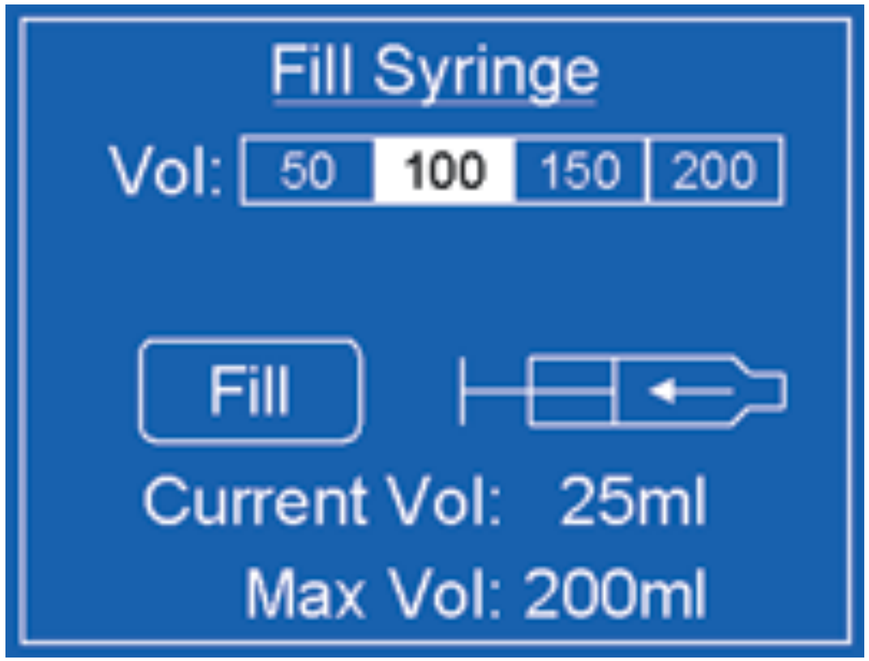
Fill and empty syringe



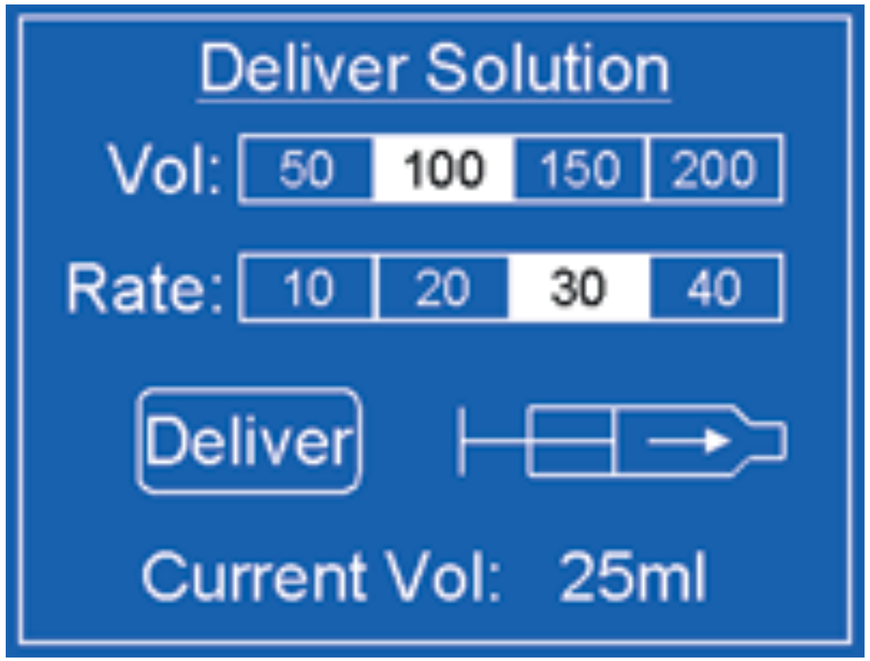
Modeless

MODES ARE SOMETIMES GOOD

If task requires modes, interface may also contain modes



Fill Mode



Deliver Mode

QUASIMODES

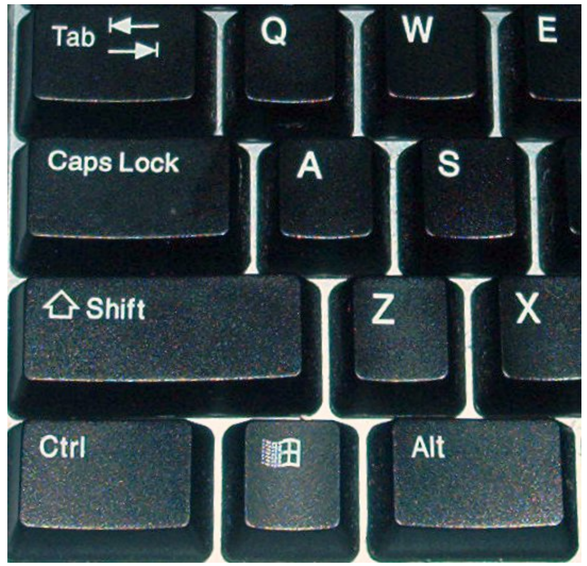
Set and hold a mode via conscious, continuous action

Shift key to capitalize (vs. Caps Lock)

Foot pedal that must remain pressed

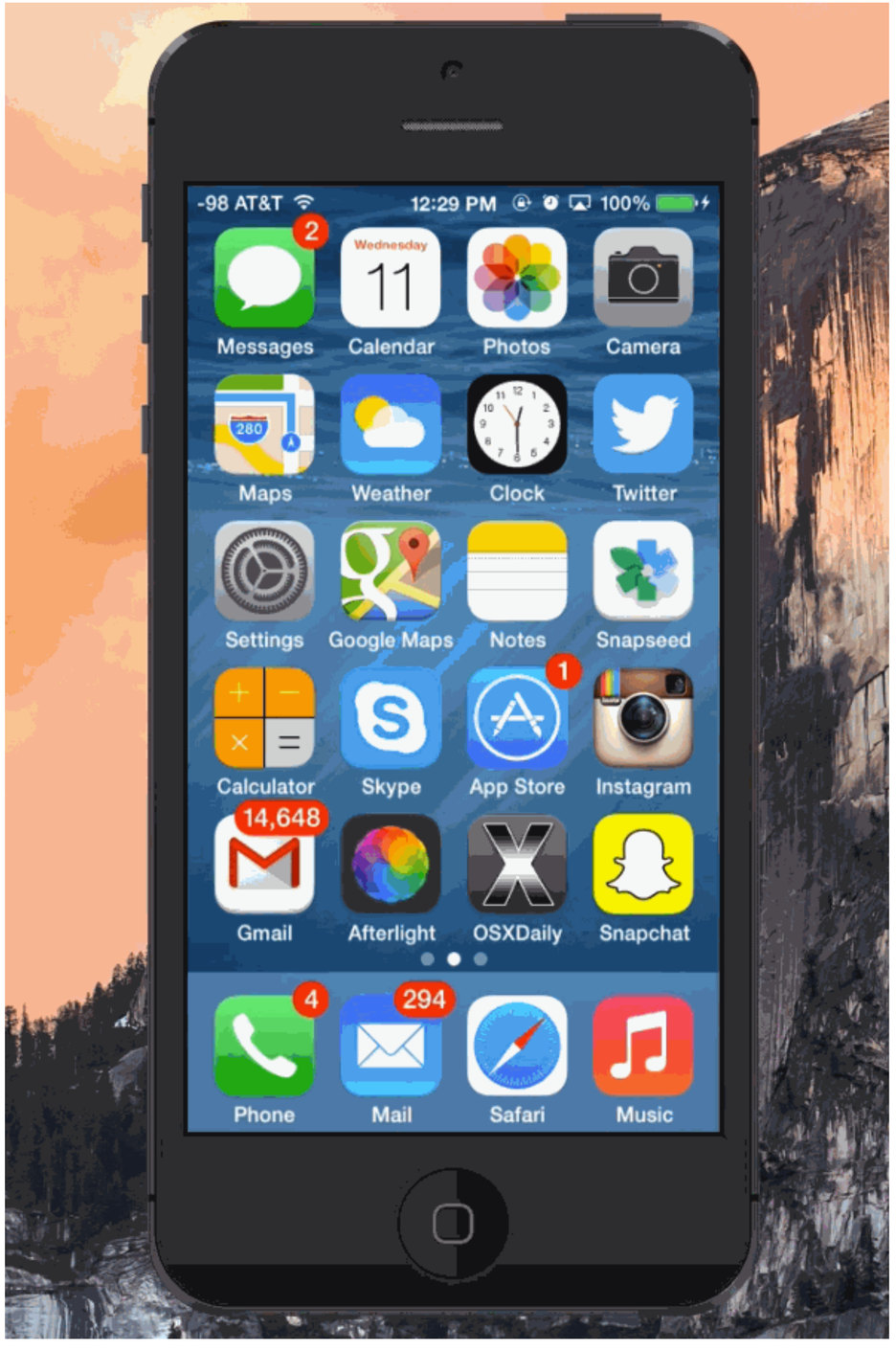
Pull down menus

Muscle tension reminds users they are holding a mode



Also known as “spring-loaded modes”


esc
F1
F2
F3
F4
F5
F6
F7
F8
F9
F10
tab
caps lock
shift
fn
control
alt
option
command
command
alt
option


-98 AT&T
12:29 PM
100%
Messages
Calendar
Wednesday
11
Photos
Camera
Maps
Weather
Clock
Twitter
Settings
Google Maps
Notes
Snapseed
Calculator
Skype
App Store
Instagram
Gmail
Afterlight
OSXDaily
Snapchat
Phone
Mail
Safari
Music

SUMMARY

Metaphor

Leverages knowledge of familiar objects & experiences

Transfer this knowledge to abstract tasks and concepts

Easily mismatched or misused so be careful!

Modes

Can create memory issues and cause serious errors

Avoid modes in your designs!

Reducing the number of modes will reduce the user's mental effort in using your interface

Design modes that match user tasks

3D TOUCH

Glimpse: a Novel Input Model for Multi-level Devices

Clifton Forlines, Chia Shen
Mitsubishi Electric Research Labs (MERL)
201 Broadway, Cambridge MA 02139 USA
{forlines, shen}@merl.com

Bill Buxton
Buxton Design
888 Queen St. East, Toronto
Ontario Canada, M4M 1J3
bill@billbuxton.com

CHI 2005

ABSTRACT
We describe a technique that supports the previewing of navigation, exploration, and editing operations by providing convenient *Undo* for unsuccessful and/or undesirable actions on multi-level input devices such as touch screens and pen-based computers. By adding a *Glimpse* state to traditional three-state pressure sensitive input devices, users are able to preview the effects of their editing without committing to them. From this *Glimpse* state, users can undo their action as easily as they can commit to it, making *Glimpse* most appropriate for systems in which the user is likely to try out many variations of an edit before finding the right one. Exploration is encouraged as the cumbersome returning to a menu or keyboard to issue an *Undo* command is eliminated. *Glimpse* has the added benefits that the negative effects of inconsistencies in the *Undo* feature within an application are reduced.

Author Keywords
Pressure Sensitive Input, Undo, Direct Manipulation, Three-State Input, Touch Screens, Stylus, Navigation

ACM Classification Keywords
H.5.2.h Information interfaces and presentation (e.g., HCI): User Interfaces - Input devices and strategies

H.5.2.i Information interfaces and presentation (e.g., HCI): User Interfaces – Interaction Styles

INTRODUCTION
Undo is a critical feature in many computer applications as it frees the user from the fear of experimenting with changes to the application's state. Creativity is enhanced when users are able to easily retract their changes if the result is unsatisfactory. People in creative industries will often state that the quality of their end product is a direct function of how many variations they *tried out and threw away* during the development of their product. Because of the frequency with which *Undo* is used, even small improvements to this feature can have a large positive effect. Additionally, many editing operations that take place

We propose a system-wide method of providing a *Glimpse* of the results of any operation completed with multi-level input devices. By multi-level, we mean that the input device is capable of sensing at least two levels of input (e.g. a stylus that senses light and heavy pressure or a mouse with a two-state button [5]) in addition to providing positional feedback. This positional feedback can be on-screen, in the case of an on-screen mouse pointer that tracks the movement of the mouse, or implicit, in the case of a finger or stylus with which the user operates directly on the display surface with graphical elements that are positioned directly underneath the input device. Our method has the added benefit that opting out of an action is as easy to perform as committing to it, making *Glimpse* most appropriate for systems in which the user is likely to try out many variations of an edit before deciding on any particular one.

BACKGROUND PART 1: PRESSURE SENSITIVE INPUT
Figure 1 shows Buxton's three-state model for stylus input [2]. In this model, light pressure input results in the

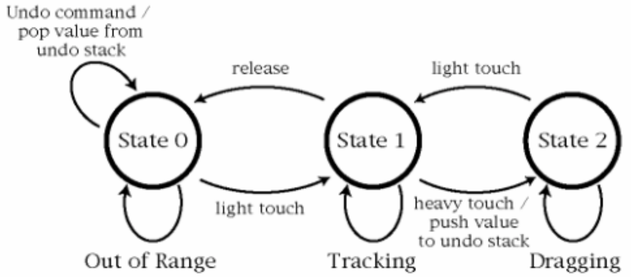


Figure 1. Buxton's three-state model for pressure sensitive input. Dragging an object results in its value being pushed onto the system undo stack. Restoring this saved value occurs after a separate command.

"tracking" of the input device (similar to moving the mouse while its button is up). While "tracking", a graphical pointer follows the movement of the input device on screen. Heavy pressure input results in "dragging" operations (similar to

Glimpse: a Novel Input Model for Multi-level Devices

Clifton Forlines, Chia Shen
Mitsubishi Electric Research Labs (MERL)
201 Broadway, Cambridge MA 02139 USA
{forlines, shen}@merl.com

Bill Buxton
Buxton Design
888 Queen St. East, Toronto
Ontario Canada, M4M 1J3
bill@billbuxton.com

CHI 2005

ABSTRACT
We describe a technique that supports the previewing of navigation, exploration, and editing operations by providing convenient *Undo* for unsuccessful and/or undesirable actions on multi-level input devices such as touch screens and pen-based computers. By adding a *Glimpse* state to traditional three-state pressure sensitive input devices, users are able to preview the effects of their editing without committing to them. From this *Glimpse* state, users can undo their action as easily as they can commit to it, making *Glimpse* most appropriate for systems in which the user is likely to try out many variations of an edit before finding the right one. Exploration is encouraged as the cumbersome returning to a menu or keyboard to issue an *Undo* command is eliminated. *Glimpse* has the added benefits that the negative effects of inconsistencies in the *Undo* feature within an application are reduced.

Author Keywords
Pressure Sensitive Input, Undo, Direct Manipulation, Three-State Input, Touch Screens, Stylus, Navigation

ACM Classification Keywords
H.5.2.h Information interfaces and presentation (e.g., HCI): User Interfaces - Input devices and strategies

H.5.2.i Information interfaces and presentation (e.g., HCI): User Interfaces – Interaction Styles

INTRODUCTION
Undo is a critical feature in many computer applications as it frees the user from the fear of experimenting with changes to the application's state. Creativity is enhanced when users are able to easily retract their changes if the result is unsatisfactory. People in creative industries will often state that the quality of their end product is a direct function of how many variations they *tried out and threw away* during the development of their product. Because of the frequency with which *Undo* is used, even small improvements to this feature can have a large positive effect. Additionally, many editing operations that take place

We propose a system-wide method of providing a *Glimpse* of the results of any operation completed with multi-level input devices. By multi-level, we mean that the input device is capable of sensing at least two levels of input (e.g. a stylus that senses light and heavy pressure or a mouse with a two-state button [5]) in addition to providing positional feedback. This positional feedback can be on-screen, in the case of an on-screen mouse pointer that tracks the movement of the mouse, or implicit, in the case of a finger or stylus with which the user operates directly on the display surface with graphical elements that are positioned directly underneath the input device. Our method has the added benefit that opting out of an action is as easy to perform as committing to it, making *Glimpse* most appropriate for systems in which the user is likely to try out many variations of an edit before deciding on any particular one.

BACKGROUND PART 1: PRESSURE SENSITIVE INPUT
Figure 1 shows Buxton's three-state model for stylus input [2]. In this model, light pressure input results in the

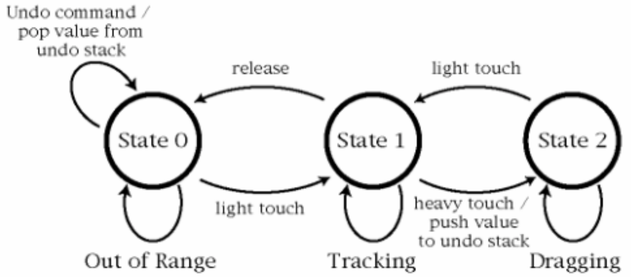


Figure 1. Buxton's three-state model for pressure sensitive input. Dragging an object results in its value being pushed onto the system undo stack. Restoring this saved value occurs after a separate command.

"tracking" of the input device (similar to moving the mouse while its button is up). While "tracking", a graphical pointer follows the movement of the input device on screen. Heavy pressure input results in "dragging" operations (similar to

CHI 2004


Pressure Marks
Gonzalo Ramos
Ravin Balakrishnan
Dynamic Graphics Project
University of Toronto

Pop Through Mouse Button Interactions

UIST 2001

Robert Zeleznik, Timothy Miller and Andrew Forsberg
Brown University
Department of Computer Science
Providence, RI 02912
(401) 863-7653; {bcz,tsm,asf}@cs.brown.edu

ABSTRACT

We present a range of novel interactions enabled by a simple modification in the design of a computer mouse. By converting each mouse button to *pop through* tactile pushbuttons, similar to the focus/shutter-release buttons used in many cameras, users can feel, and the computer can sense, two distinct "clicks" corresponding to pressing lightly and pressing firmly to pop through. Despite the prototypical status of our hardware and software implementations, our current pop through mouse interactions are compelling and warrant further investigation. In particular, we demonstrate that pop through buttons not only yield an additional button activation state that is composable with, or even preferable to, techniques such as double-clicking, but also can endow a qualitatively novel user experience when meaningfully and consistently applied. We propose a number of software guidelines that may provide a consistent, systemic benefit; for example, light pressure may invoke default interaction (short menu), and firm pressure may supply more detail (long menu).

KEYWORDS: mouse, double-action, gesture, interaction, click through, buttons, pop through, input devices, haptics.

INTRODUCTION

Inspired both by the fluid interactivity of the camera button mechanism and the observation that people often press elevator buttons harder as they grow impatient, we considered a number of approaches for integrating pressure controls into computer interfaces. The choice we present, replacing mouse buttons with pop through pushbuttons (see Figure 1), increases the information bandwidth of a mouse, but we hypothesize may well be easier for users to learn, remember, and control than other choices, such as adding spatially distinct or continuously pressure-sensitive buttons. Moreover, our device is easy to deploy because it does not require visible changes to a commercial mouse, and legacy applications can be compatible simply by ignoring the added button state.

After developing a collection of novel interaction techniques to demonstrate the potential of pop through mouse interac-

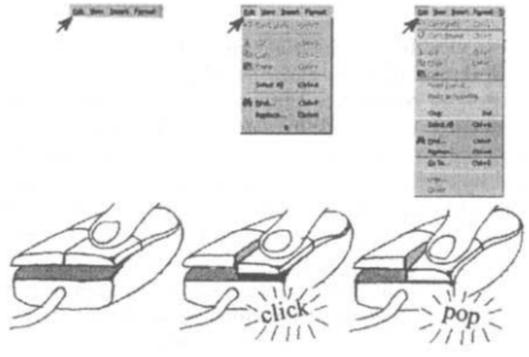


Figure 1: Pressing the button lightly clicks and in this example invokes a short menu; pressing harder pops through with a second click, in this example invoking a longer menu. (Button displacement exaggerated.)

tion, we performed a patent seach that uncovered a 1994 patent on a substantially similar device by Apple Computer [1]. Interestingly, the patent does not consider any of the interaction techniques that we present.

PROTOTYPE HARDWARE AND SOFTWARE

Hardware Design For the first of two different pop through mouse prototypes, we glued a small pushbutton on top of a mouse button and connected it to an RS232 port for software polling.[1] Because of the differing activation forces of the buttons, pressing down on the pushbutton first activates the button native to the mouse and then, with more pressure, pops-through to activate the additional pushbutton. In informal evaluation, users had no difficulty distinguishing or controlling the activations of the native or additional button.

For our second protoype, we replaced the internal button mechanism with an integrated double-action surface-mount pushbutton[2], similar to a camera's focus/shutter-release button. The activation forces for this pushbutton are quite subtle, with no perceptible click for the first contact and a very soft

Pop Through Mouse Button Interactions

UIST 2001

Robert Zeleznik, Timothy Miller and Andrew Forsberg
Brown University
Department of Computer Science
Providence, RI 02912
(401) 863-7653; {bcz,tsm,asf}@cs.brown.edu

ABSTRACT

We present a range of novel interactions enabled by a simple modification in the design of a computer mouse. By converting each mouse button to *pop through* tactile pushbuttons, similar to the focus/shutter-release buttons used in many cameras, users can feel, and the computer can sense, two distinct "clicks" corresponding to pressing lightly and pressing firmly to pop through. Despite the prototypical status of our hardware and software implementations, our current pop through mouse interactions are compelling and warrant further investigation. In particular, we demonstrate that pop through buttons not only yield an additional button activation state that is composable with, or even preferable to, techniques such as double-clicking, but also can endow a qualitatively novel user experience when meaningfully and consistently applied. We propose a number of software guidelines that may provide a consistent, systemic benefit; for example, light pressure may invoke default interaction (short menu), and firm pressure may supply more detail (long menu).

KEYWORDS: mouse, double-action, gesture, interaction, click through, buttons, pop through, input devices, haptics.

INTRODUCTION

Inspired both by the fluid interactivity of the camera button mechanism and the observation that people often press elevator buttons harder as they grow impatient, we considered a number of approaches for integrating pressure controls into computer interfaces. The choice we present, replacing mouse buttons with pop through pushbuttons (see Figure 1), increases the information bandwidth of a mouse, but we hypothesize may well be easier for users to learn, remember, and control than other choices, such as adding spatially distinct or continuously pressure-sensitive buttons. Moreover, our device is easy to deploy because it does not require visible changes to a commercial mouse, and legacy applications can be compatible simply by ignoring the added button state.

After developing a collection of novel interaction techniques to demonstrate the potential of pop through mouse interac-

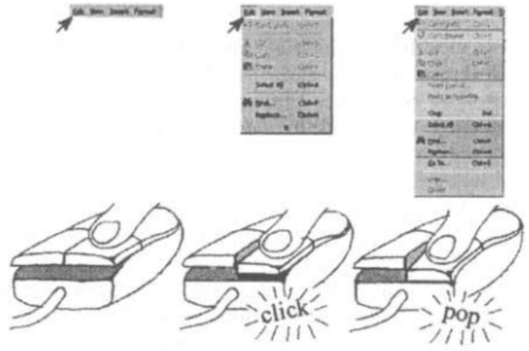


Figure 1: Pressing the button lightly clicks and in this example invokes a short menu; pressing harder pops through with a second click, in this example invoking a longer menu. (Button displacement exaggerated.)

tion, we performed a patent seach that uncovered a 1994 patent on a substantially similar device by Apple Computer [1]. Interestingly, the patent does not consider any of the interaction techniques that we present.

PROTOTYPE HARDWARE AND SOFTWARE

Hardware Design For the first of two different pop through mouse prototypes, we glued a small pushbutton on top of a mouse button and connected it to an RS232 port for software polling.[1] Because of the differing activation forces of the buttons, pressing down on the pushbutton first activates the button native to the mouse and then, with more pressure, pops-through to activate the additional pushbutton. In informal evaluation, users had no difficulty distinguishing or controlling the activations of the native or additional button.

For our second protoype, we replaced the internal button mechanism with an integrated double-action surface-mount pushbutton[2], similar to a camera's focus/shutter-release button. The activation forces for this pushbutton are quite subtle, with no perceptible click for the first contact and a very soft

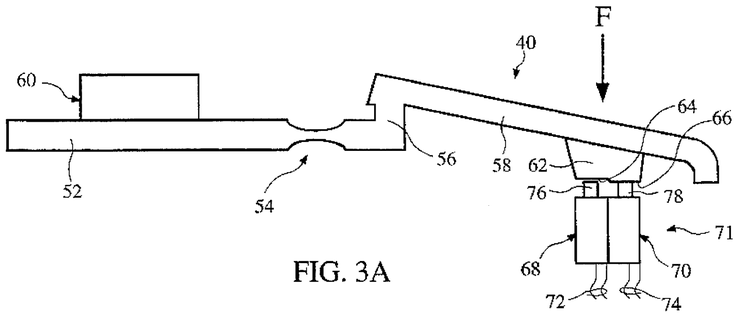


FIG. 3A

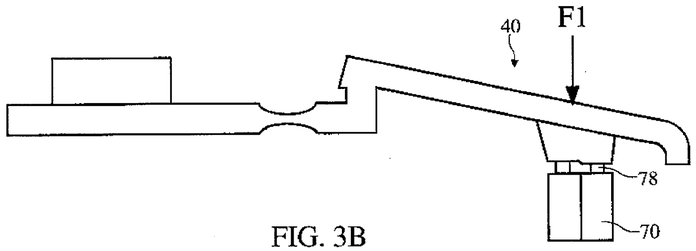


FIG. 3B

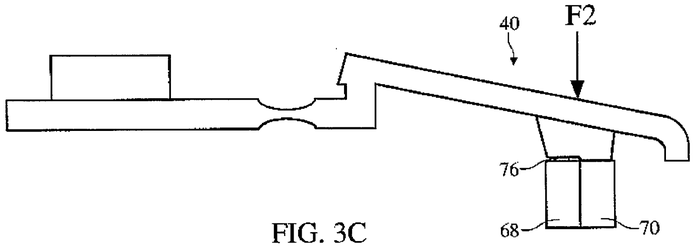


FIG. 3C

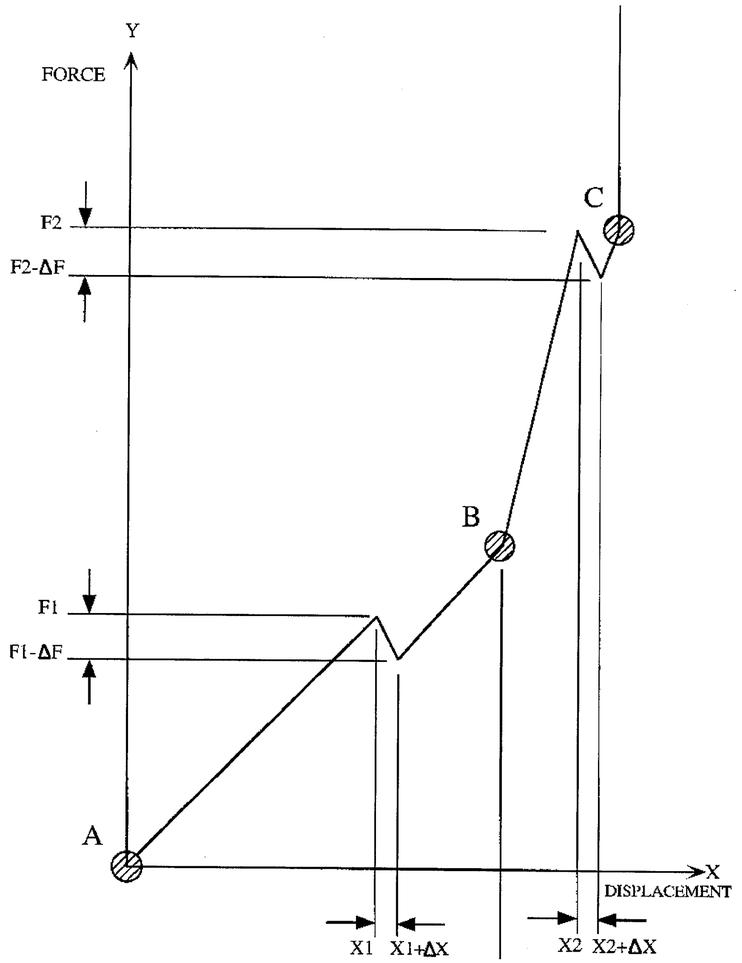


FIG. 4

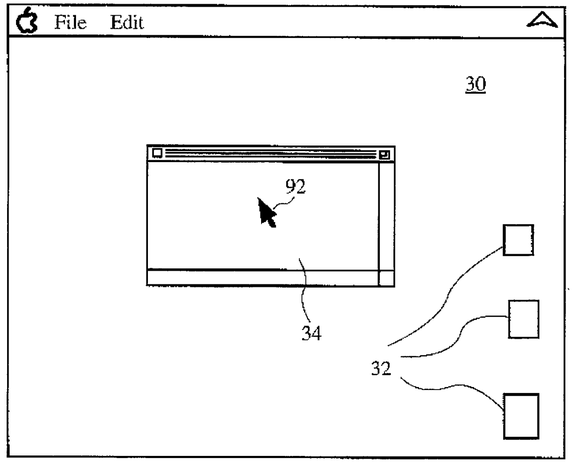


FIG. 6A

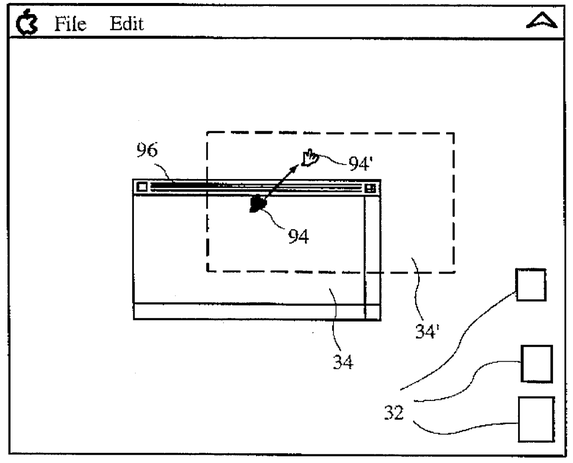
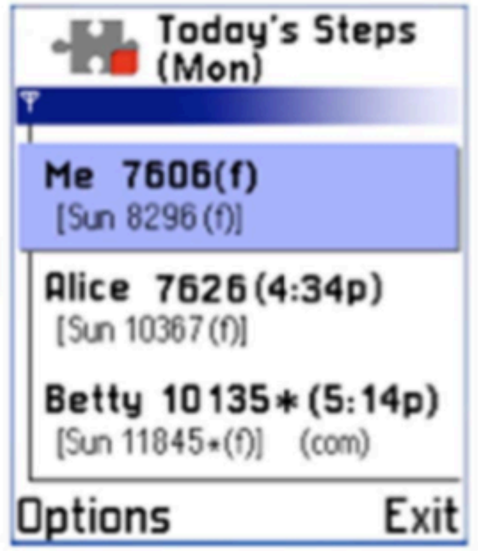


FIG. 6B


T-Mobile LTE
1:49 PM
55%
May
M T W T F S S
Today, May 9, 2018
Move
351/350 calories
36Cal
History
Workouts
Achievements
Sharing
Activity
3:44

Houston

- Sunny Consolvo et al (Intel, UW)



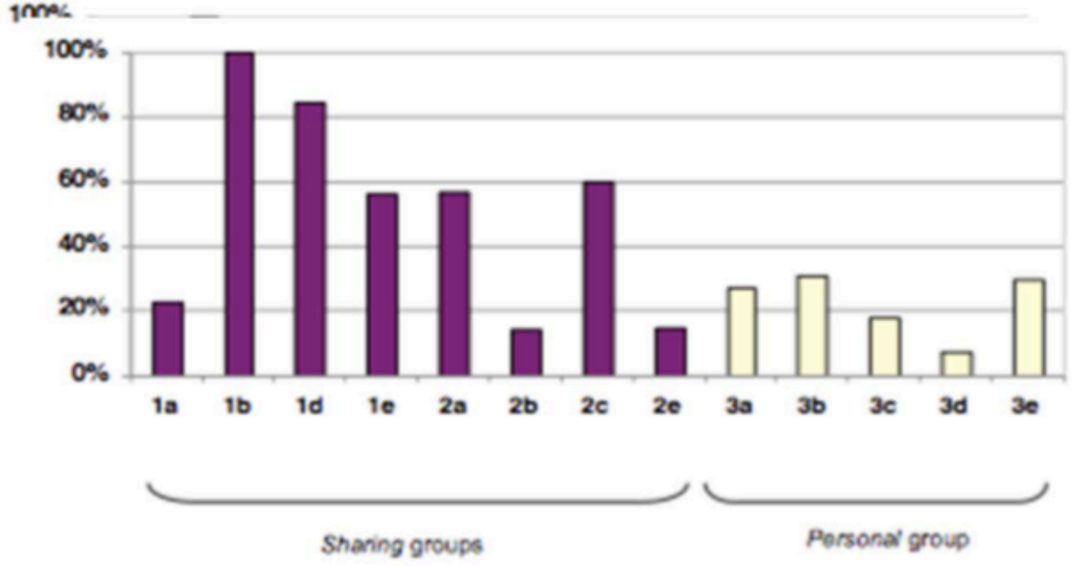


Figure 3. Percentage of days participants met their goals.

Figure 1. a) The Omron HJ-112 pedometer, b) the pedometer in use, and c) the Nokia 6600 mobile phone running Houston.

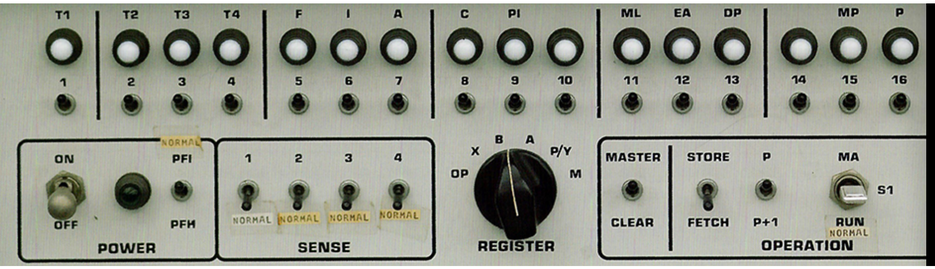
2006

a)

b)

c)

CHI 2007



INPUT DEVICES

QUESTIONS:

What (low-level) tasks are the users trying to accomplish with an input device?

How can we think about the space of possible input devices?

What interaction techniques are encouraged/discouraged by a particular device?

IMPORTANT TASKS

Text Entry

Pointing/Marking

- Target acquisition
- Steering / positioning
- Freehand drawing
- Drawing lines
- Tracing and digitizing
- ...

TEXT ENTRY: KEYSTROKE DEVICES

Array of Discrete Inputs

Many variants of form and key layout

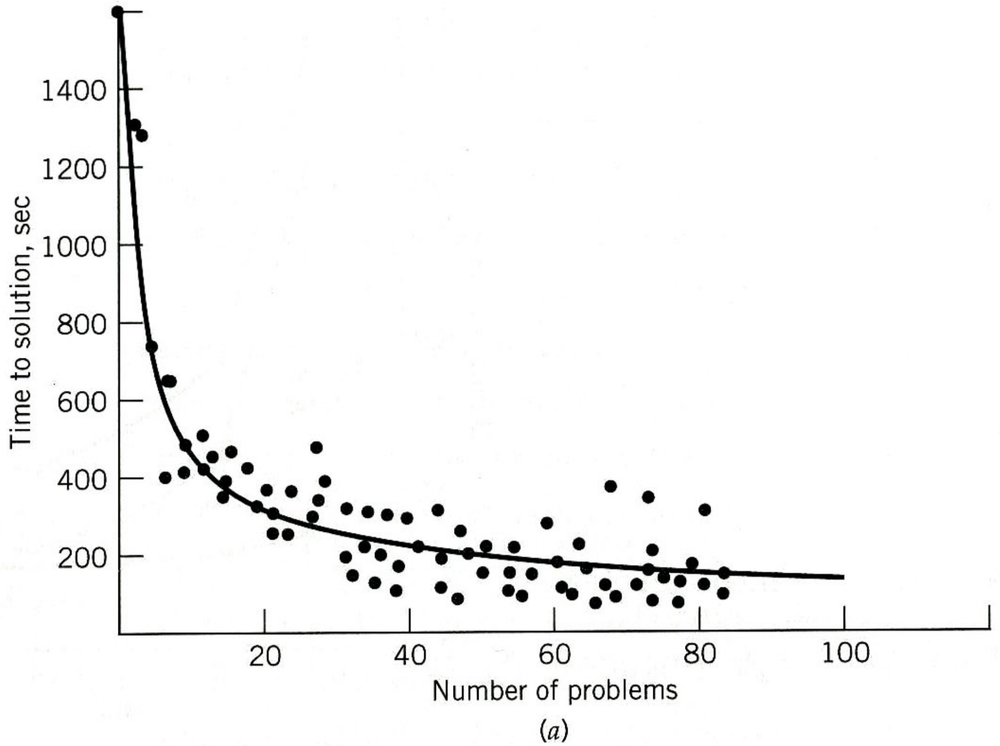
Can be one-handed or two

Wide range of sizes

Two-hand full keyboard is relatively standardized, Less standa
generic function keys, cursor movement, numeric keypad,...

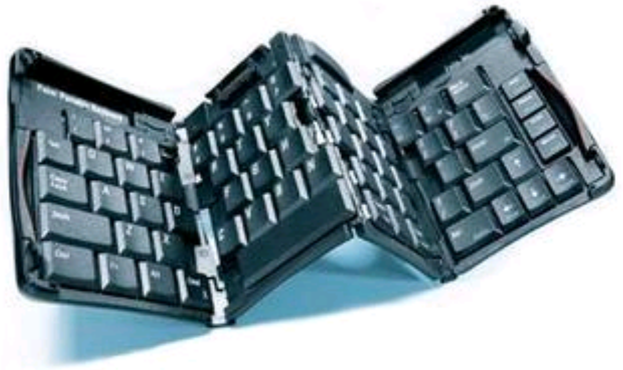
Take advantage of procedural memory

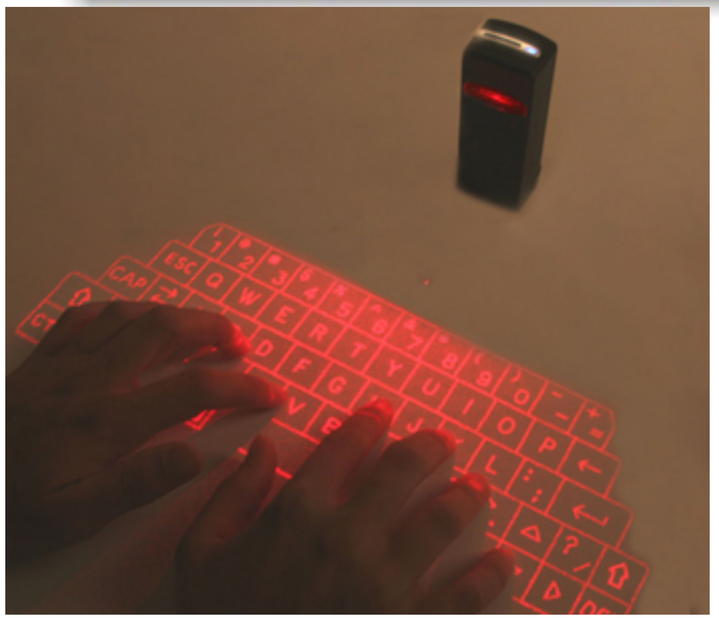
Power law of practice

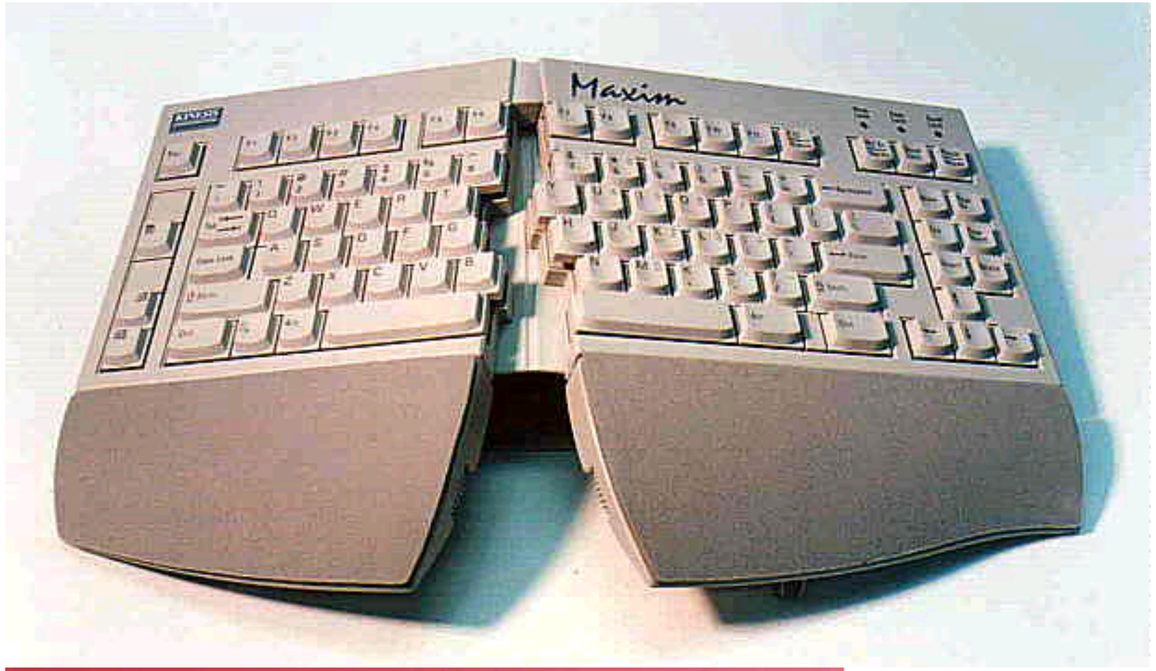


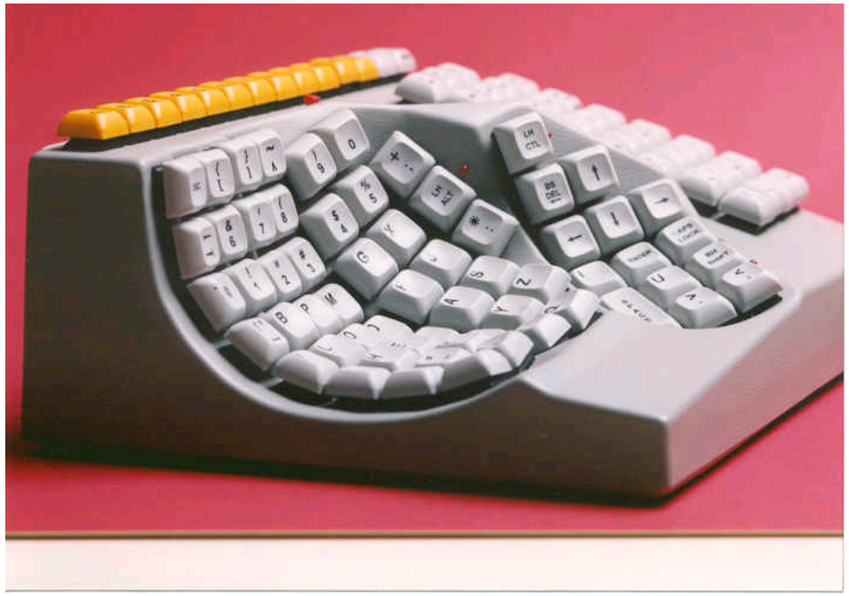
$$T_n = T_1 n^{-a} + c$$

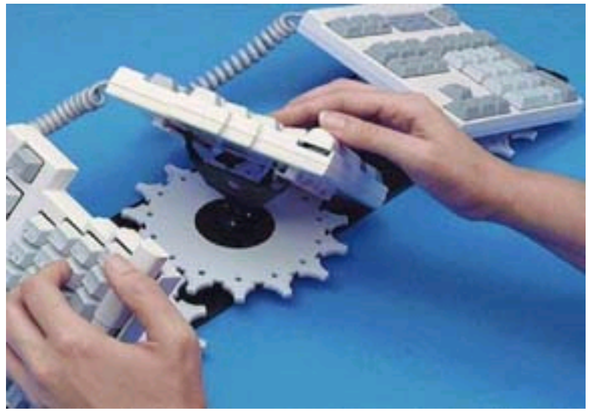
KEYBOARDS






Maxim





KEY LAYOUTS


QWERTY
Q W E R T Y U I O P
A S D F G H J K L
Z X C V B N M
DVORAK
P Y F G C R L
A O E U I D H T N S
Q J K X B M W V Z

DIFFICULTY: TEXT ENTRY

Still very hard on mobile devices

Keyboards (on-screen and thumb)

Full hand-writing recognition

Graffiti

EdgeWrite

ShapeWriter

MOBILE TEXT ENTRY: KEYPADS

Multi-tap mappings

Multiple presses per letter

Ambiguity resolution

One press per letter, dictionary lookup

MOBILE TEXT ENTRY: KEYPADS

Chording

Multiple keys pressed simultaneously

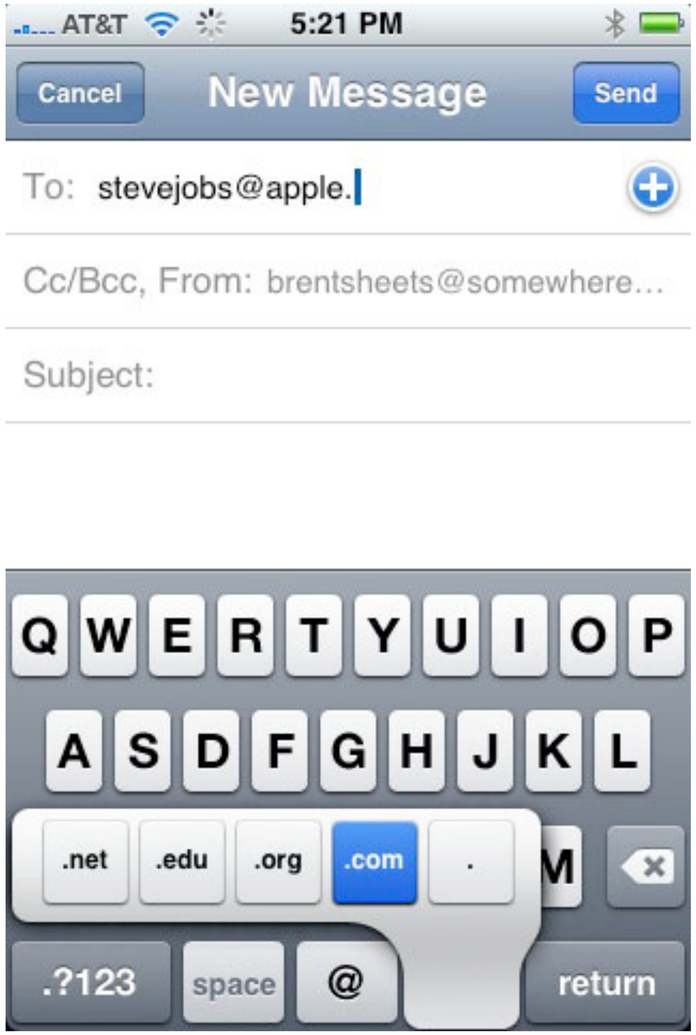
2^n combinations for n keys

Twiddler2, HandyKey

MOBILE TEXT ENTRY: SOFT KEYS

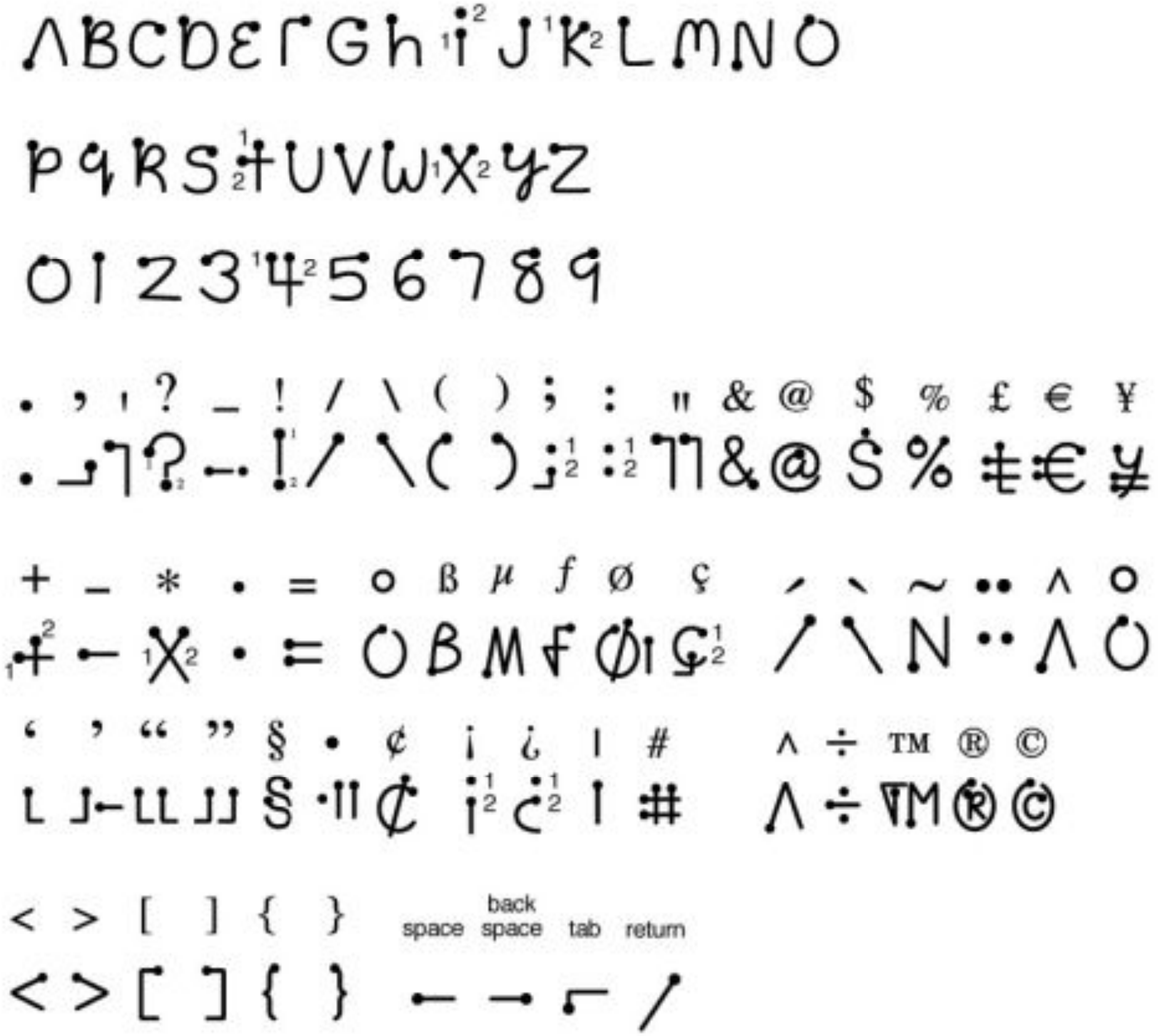
Soft Keyboards

Benefits? Drawbacks?



MOBILE TEXT ENTRY: HANDG RECOG

GRAFFITI – UNISTROKE TEXT ENTRY



EDGEWRITE

Corner-based text input technique

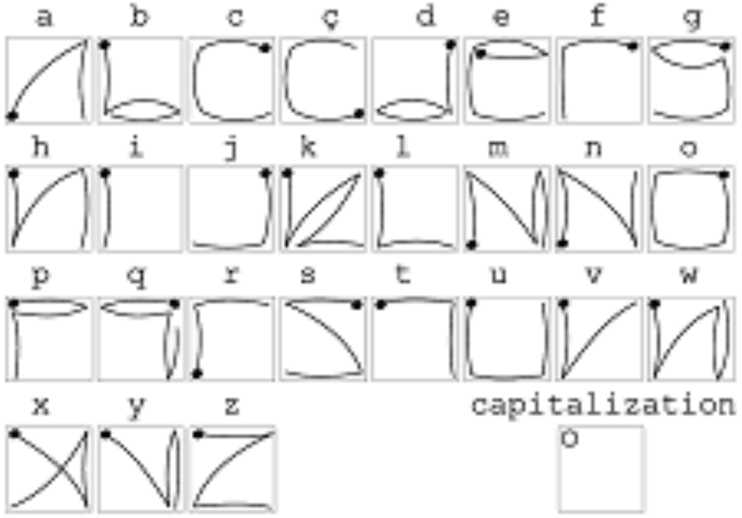
Makes use of physical edges and corners to improve input time

Particularly effective for users with motor impairments

Edges provide stability

Implementable in many different input modalities

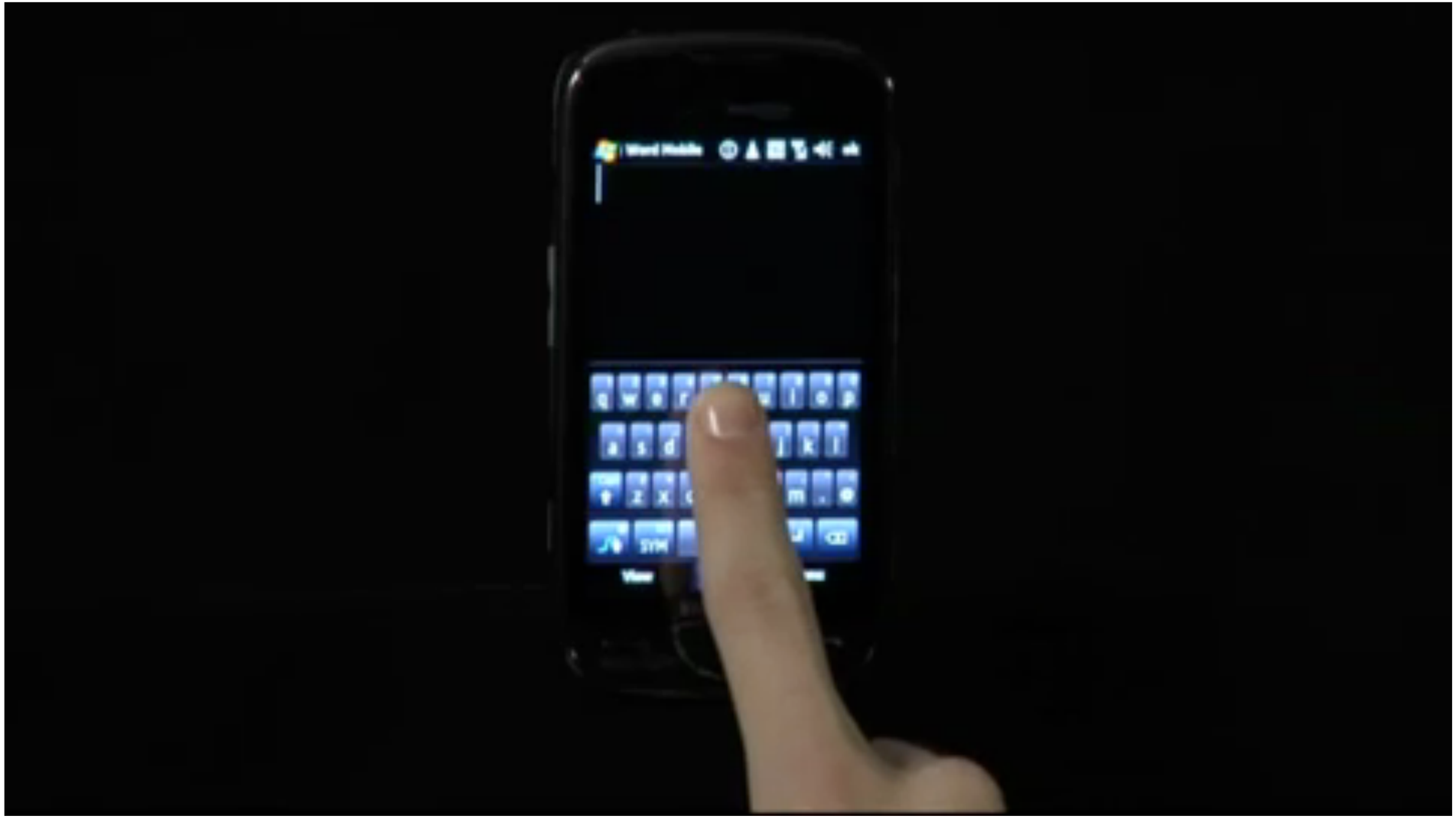
stylus, joysticks, trackball



Jacob Wobbrock, UIST 2003

MOBILE TEXT ENTRY: TOUCH / STYLUS

Stroke Entry Methods (e.g., Swype, ShapeWriter)

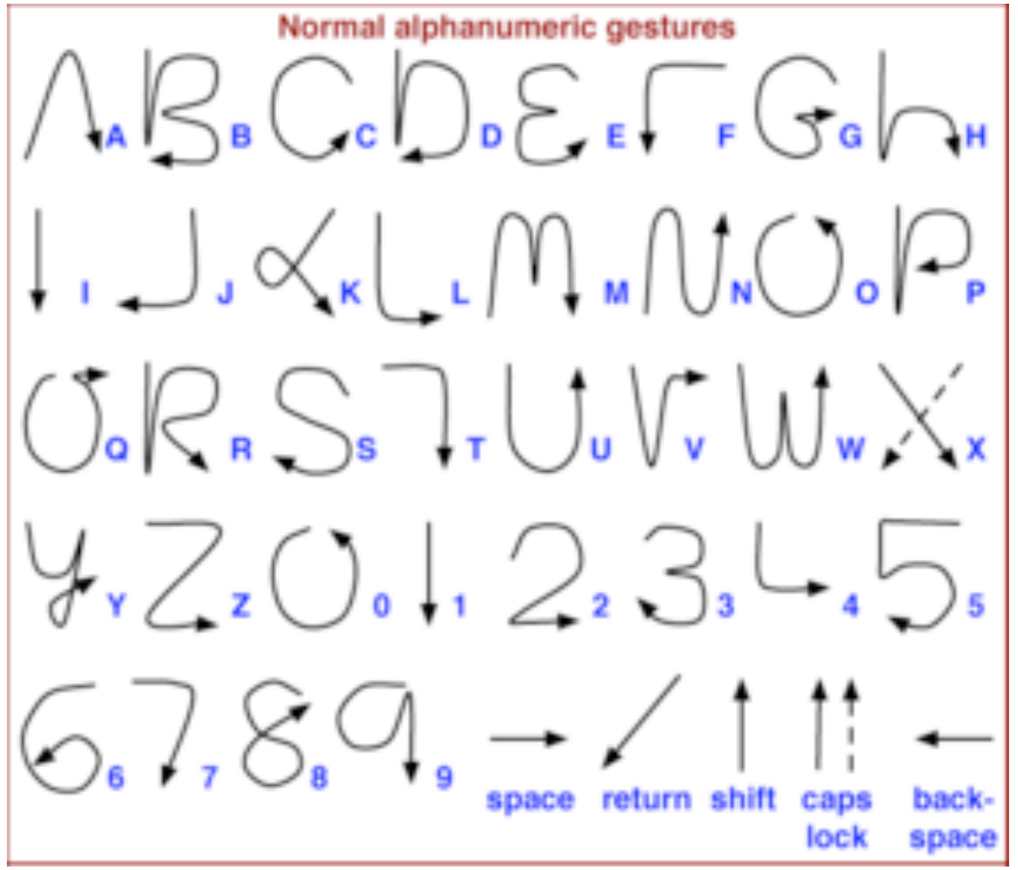


MOBILE TEXT ENTRY: TOUCH / STYLUS

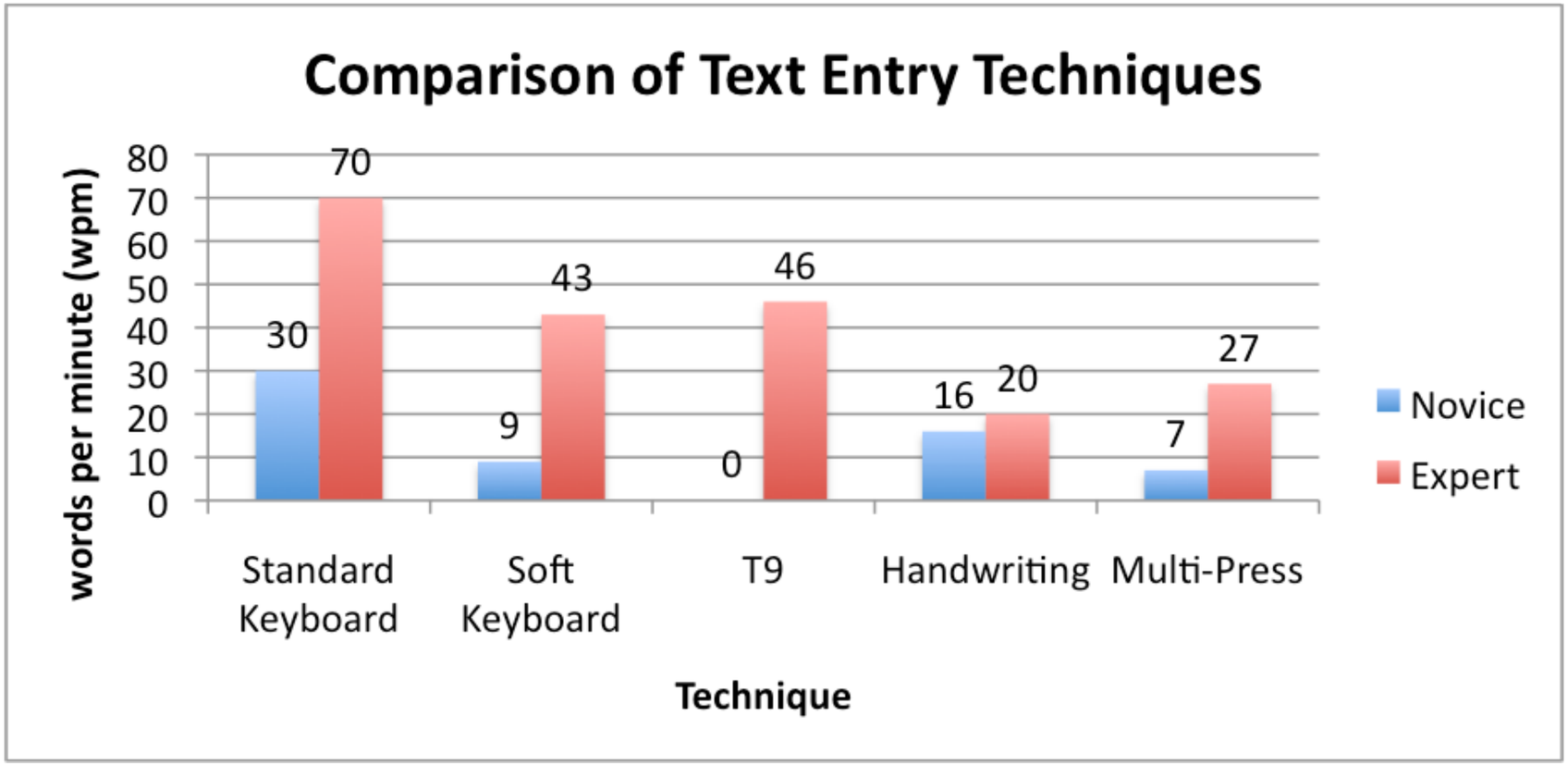
Custom symbol sets

improve recognition accuracy;

appropriate for indirect (eyes-free) input



WHICH IS FASTEST?



WHAT ABOUT SPEECH RECOGNITION?

Dictation is faster than typing (~100 wpm)

WHAT ABOUT SPEECH RECOGNITION?

Dictation is faster than typing (~100 wpm), BUT:

Speech is different from written language:
Speaking in well-formed, complete, print-ready sentences is cognitively challenging

High cost of correcting errors through speech channel alone

Social awkwardness?

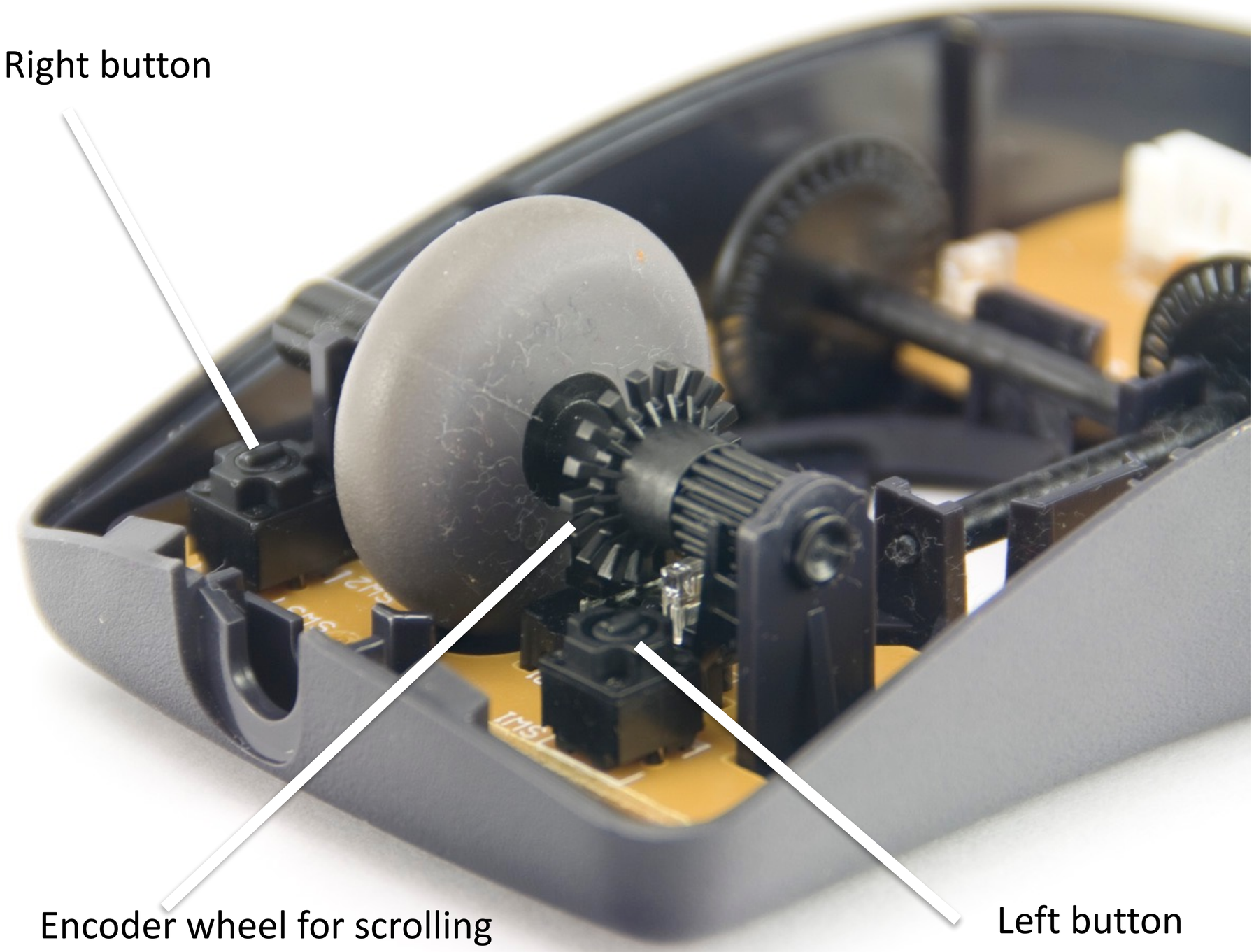
POINTING DEVICES

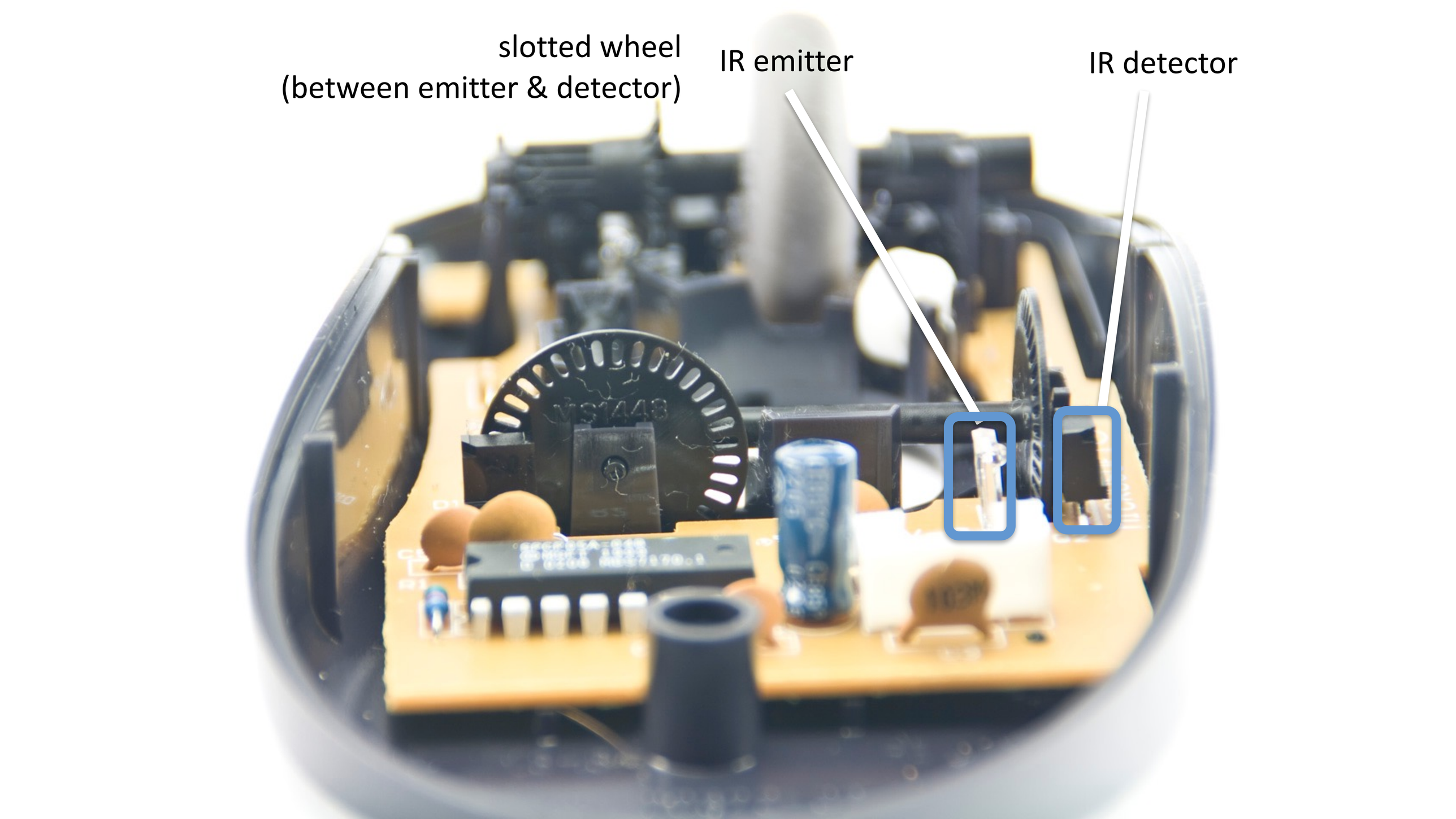
Mouse. Engelbart and English ~1964

Source: Card, Stu. Lecture on Human Information Interaction. Stanford, 2007.

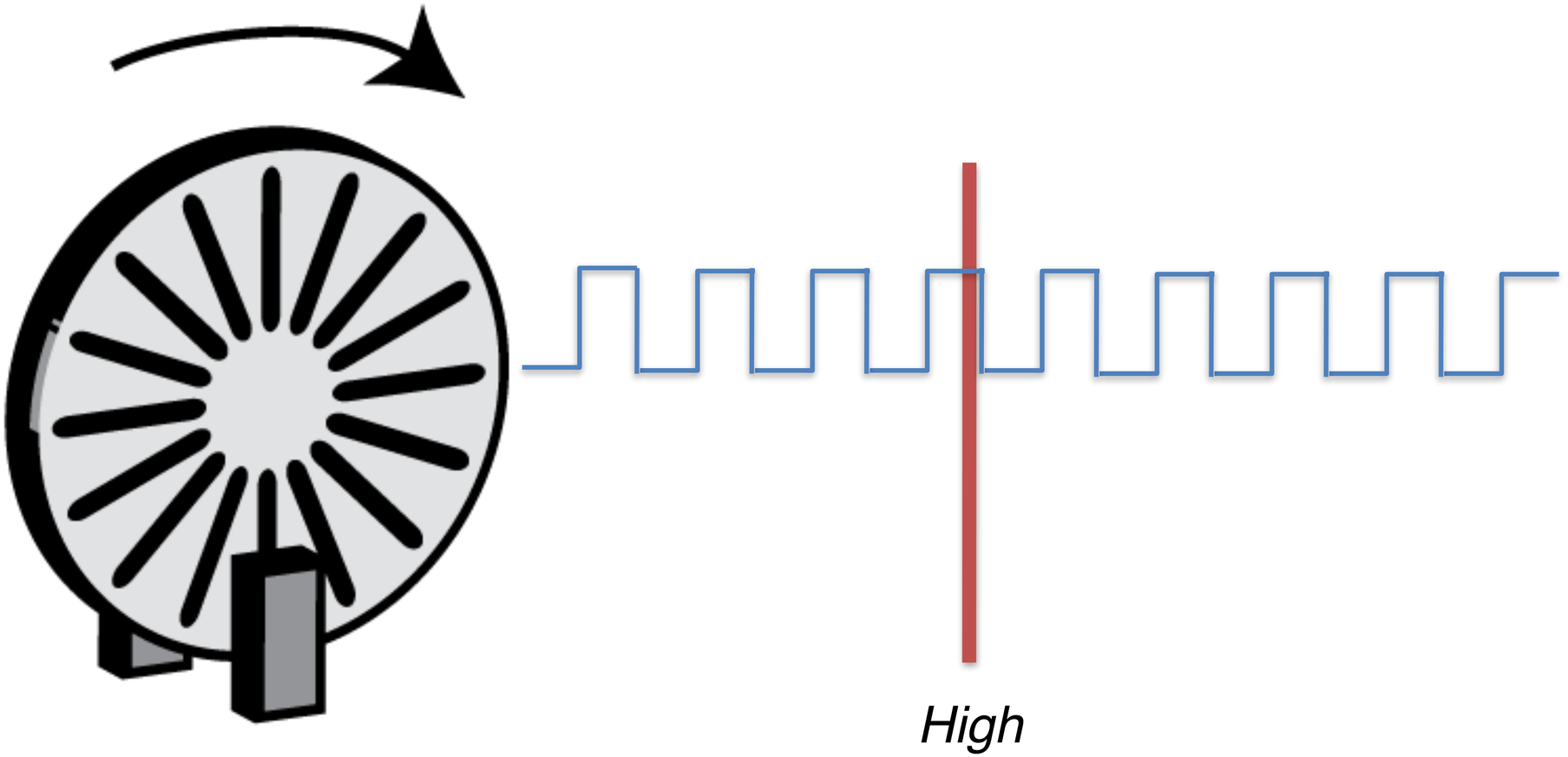

DELL
by Microsoft


R2
C6
C3
D2
D3
R1
U1


Right button
Encoder wheel for scrolling
Left button


slotted wheel
(between emitter & detector)
IR emitter
IR detector

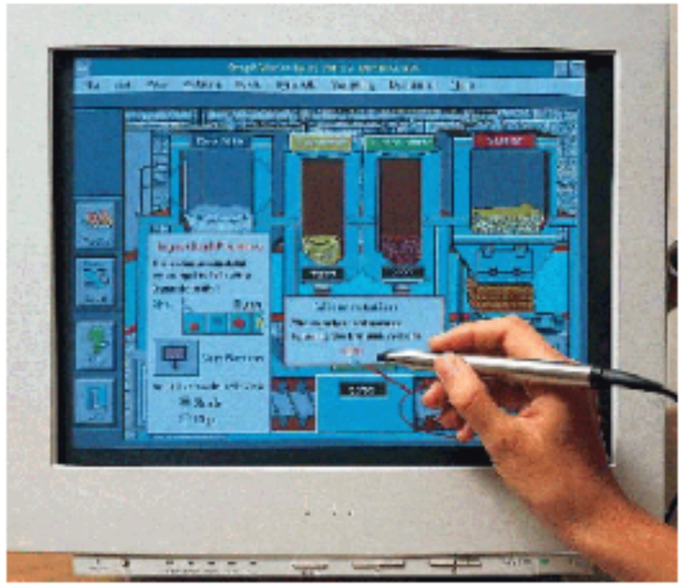
SENSING: ROTARY ENCODER



ABSOLUTE VS RELATIVE

Track Ball (relative, indirect locator)

Mice (relative, indirect locator)



Light Pen (absolute, direct locator)

Touch Screen (absolute, direct locator)

Absolute locators: have an origin location and locate in this frame of reference

Relative locators: report location relative to their previous location, rather than relative to a fixed origin

Direct locator: user points directly at the screen

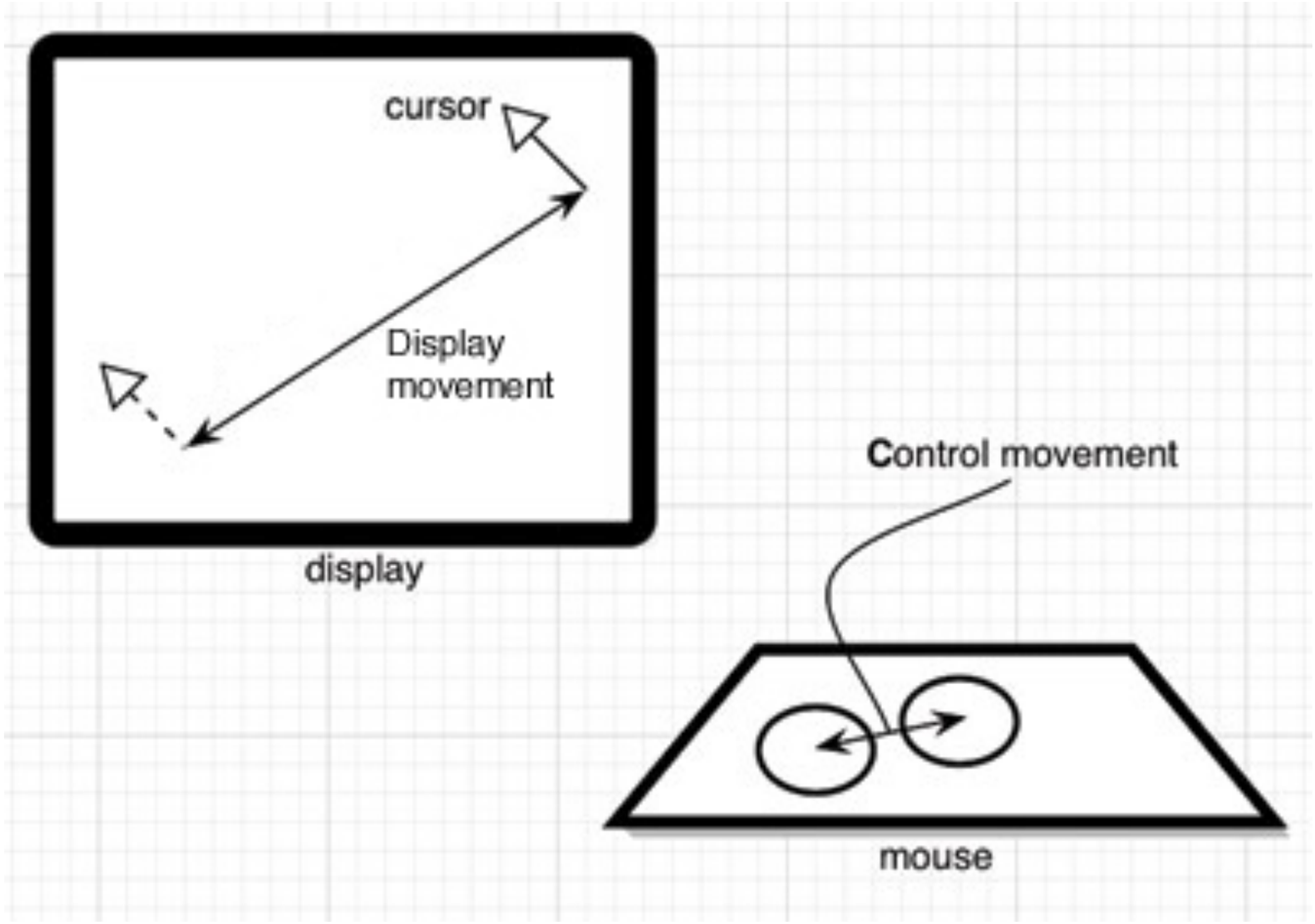
Indirect locator device user moves a cursor on the screen using a device separate from the screen

CONTROL TO DISPLAY RATIO (C:D RATIO)

Ratio of the speed of hand movement (**C**ontrol) to the speed/distance of cursor movement (**D**isplay) for a continuous locator device

Large ratio - large hand movement / small cursor movement (Good for accurate positioning, poor for long movements)

Small ratio - small hand movement / large cursor movement (Good for rapid movements across long distances, poor for accurate positioning)



OTHER DEVICE PROPERTIES:

Indirect vs. Direct

Direct: Input and output space are unified

C:D Ratio

For one unit of movement in physical space, how far does the cursor travel in display space?

Q: What is the C:D ratio for direct touch screen input?

Device Acquisition Time

MOBILE POINTING

D-Pad
(see: arrow keys)

Trackball

Direct touch
(see: Trackpad)

Stylus

Everything is best for something and worst for something else.

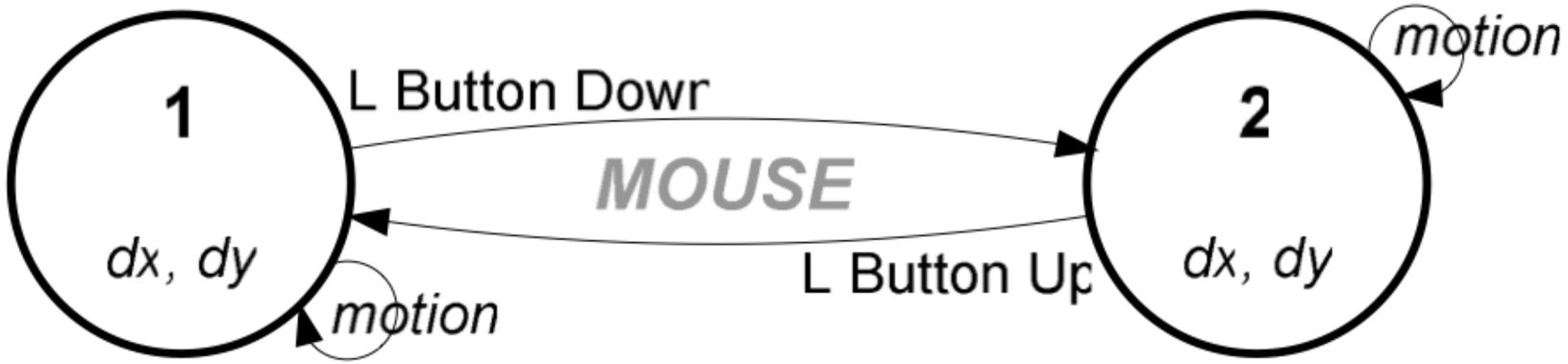
- Bill Buxton

3-STATE MODEL OF INPUT (BUXTON)

State	Description
0	*Out Of Range:* The device is not in its physical tracking range.
1	*Tracking:* Device motion moves only the cursor.
2	*Dragging:* Device motion moves objects on the screen.

(Table from Hinckley Reading)

MOUSE



(Figure from Hinckley Reading)

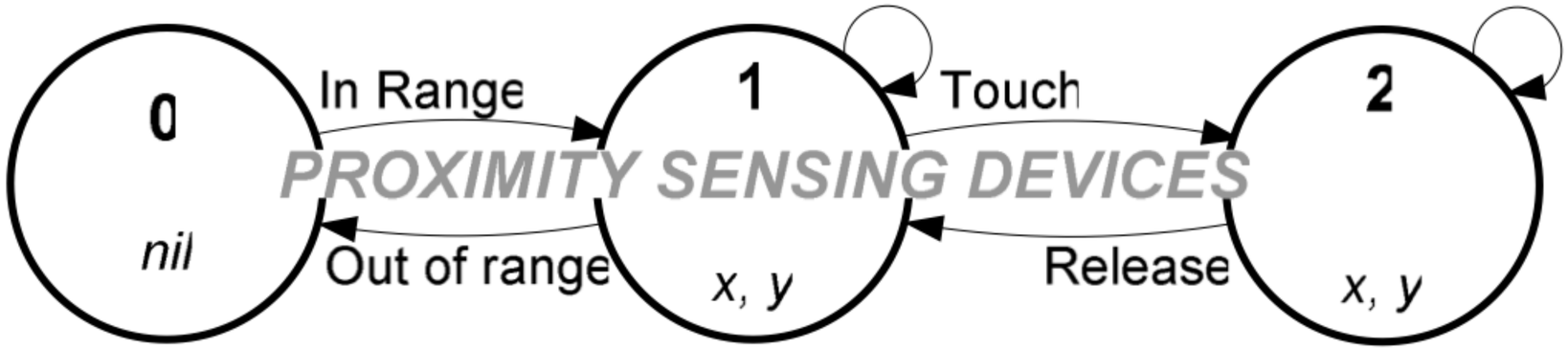
State	Description
0	*Out Of Range:* The device is not in its physical tracking range.
1	*Tracking:* Device motion moves only the cursor.
2	*Dragging:* Device motion moves objects on the screen.

TOUCH SCREEN

(Figure from Hinckley Reading)

State	Description
0	*Out Of Range:* The device is not in its physical tracking range.
1	*Tracking:* Device motion moves only the cursor.
2	*Dragging:* Device motion moves objects on the screen.

STYLUS ON TABLET



(Figure from Hinckley Reading)

State	Description
0	*Out Of Range:* The device is not in its physical tracking range.
1	*Tracking:* Device motion moves only the cursor.
2	*Dragging:* Device motion moves objects on the screen.