Input Devices

CS160: User Interfaces
John Canny
Assignments

• Contextual Inquiry and Task Analysis due 10/4
• Competitive Analysis due 9/29
• Individual programming assignment 3 due 9/29
XML Parsing

<?xml version="1.0" encoding="UTF-8" ?>
xmlns:gd="http://schemas.google.com/g/2005" gd:etag="W/"CxEEH8H7cSp7I2A9Wx5WEU4.""/>

  <atom:title>Faulkner's Birthplace</atom:title>
  <atom:content type="application/vnd.google-earth.kml+xml">
    <Placemark>
      <name>Faulkner's Birthplace</name>
      <Style />
      <Point>
        <coordinates>89.520753,34.360902,0.0</coordinates>
      </Point>
    </Placemark>
  </atom:content>
  <atom:link rel="self" type="application/atom+xml"
    href="http://maps.google.com/maps/feeds/features/211131567569304217895/000490ce5221bab68bb30/full/000490d241f76e7e396cb" />
  <atom:link rel="edit" type="application/atom+xml"
    href="http://maps.google.com/maps/feeds/features/211131567569304217895/000490ce5221bab68bb30/full/000490d241f76e7e396cb" />
</atom:entry>
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 standardization on others: Command keys, generic function keys, cursor movement, numeric keypad,...

- Use procedural memory
  - Power law of practice

\[ T_n = T_1 n^{-a} + c \]
Key Layouts

QWERTY

QWERTYUIOP{}

ASDFGHJKL;

ZXCVBNM,.?/

DVORAK

!@#$%^&*()_+
1234567890

.<!>/?'".:;

:;.,"PYFGLCIR:

AOEUHDSL:

QJKXBMNWVZ
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
Mobile Text Entry: Touch / Stylus

Soft Keyboards
• Benefits?
• Drawbacks?

Mactoids.com
Mobile Text Entry: Handwriting Recog.
Mobile Text Entry: Touch / Stylus

- Custom symbol sets
- Improve recognition accuracy; appropriate for indirect (eyes-free) input
Mobile Text Entry: Touch / Stylus

- Stroke Entry Methods (e.g., Swype, ShapeWriter)
Which is fastest?

Comparison of Text Entry Techniques

<table>
<thead>
<tr>
<th>Technique</th>
<th>Novice</th>
<th>Expert</th>
</tr>
</thead>
<tbody>
<tr>
<td>Standard Keyboard</td>
<td>30</td>
<td>70</td>
</tr>
<tr>
<td>Soft Keyboard</td>
<td>9</td>
<td>43</td>
</tr>
<tr>
<td>T9</td>
<td>0</td>
<td>46</td>
</tr>
<tr>
<td>Handwriting</td>
<td>16</td>
<td>20</td>
</tr>
<tr>
<td>Multi-Press</td>
<td>7</td>
<td>27</td>
</tr>
</tbody>
</table>
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

(cc) Flickr photo by Mike fj40
Mouse. Engelbart and English ~1964

Right button

Encoder wheel for scrolling

Left button
slotted wheel (between emitter & detector)

IR emitter

IR detector
Sensing: Rotary Encoder

High
Sensing: Fwd Rotation

Low
Sensing: Backwd Rotation

Low  Oops!
Solution: Use two out-of-phase detectors
Sensing: Rotary Encoder

Low
High
Sensing: Rotary Encoder

Coding:
HH→ LH: $dx = 1$
HH→ HL: $dx = -1$
Device Abstraction

- Click, DoubleClick, MouseUp, MouseDown,MouseMove …
What about optical mice?

Source: http://spritesmods.com/?art=mouseeye
What is sensed?

<table>
<thead>
<tr>
<th></th>
<th>Linear</th>
<th>Rotary</th>
</tr>
</thead>
<tbody>
<tr>
<td>Position</td>
<td>Position $P$</td>
<td>Rotation $R$</td>
</tr>
<tr>
<td>Absolute</td>
<td></td>
<td>Delta rotation $dR$</td>
</tr>
<tr>
<td>Relative</td>
<td>Movement $dP$</td>
<td></td>
</tr>
<tr>
<td>Force</td>
<td>Force $F$</td>
<td>Torque $T$</td>
</tr>
<tr>
<td>Absolute</td>
<td></td>
<td>Delta torque $dT$</td>
</tr>
<tr>
<td>Relative</td>
<td>Delta force $dF$</td>
<td></td>
</tr>
</tbody>
</table>

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
Trackball, Trackpad
Trackpoint

- Indirect, force sensing, velocity control
- Nonlinear transfer function

(credit image by flickr user tsaiid)
Mobile Pointing

- D-Pad  
  (see: arrow keys)

- Trackball

- Direct touch  
  (see: Trackpad)

- Stylus
Which is faster?

Which is faster?

Engelbart

Experiment: Mice are fastest!

Fitts’ Law

• Time $T_{\text{pos}}$ to move the hand to target size $S$ which is distance $D$ away is given by:

$$T_{\text{pos}} = a + b \log_2 (D/S + 1)$$

Fitts’ Law

• Time $T_{pos}$ to move the hand to target size $S$ which is distance $D$ away is given by:

$$T_{pos} = a + b \log_2 (D/S + 1)$$

Device Characteristics
(bandwidth of human muscle group & of device)

- $a$: start/stop time
- $b$: speed

Bandwidth of Human Muscle Groups

Why is the mouse fastest?

**Why these results?**

Time to position mouse proportional to Fitts’ Index of Difficulty $I_D$.

[i.e. how well can the muscles direct the input device]

Therefore speed limit is in the eye-hand system, not the mouse.

Therefore, mouse is a near optimal device.

Fitts’ Law Example

- Which will be faster on average?
  - pie menu (bigger targets & less distance)

Fitts’ Law in Windows & Mac OS

Windows 95: Missed by a pixel
Windows XP: Good to the last drop

The Apple menu in Mac OS X v10.4 Tiger.

Source: Jensen Harris, An Office User Interface Blog: Giving You Fitts. Microsoft, 2007; Apple
Fitts’ Law in Microsoft Office 2007

Larger, labeled controls can be clicked more quickly

Magic Corner: Office Button in the upper-left corner

Mini Toolbar: Close to the cursor

Everything is best for something and worst for something else.

- Bill Buxton
# 3-State Model of Input (Buxton)

<table>
<thead>
<tr>
<th>State</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td><em>Out Of Range:</em> The device is not in its physical tracking range.</td>
</tr>
<tr>
<td>1</td>
<td><em>Tracking:</em> Device motion moves only the cursor.</td>
</tr>
<tr>
<td>2</td>
<td><em>Dragging:</em> Device motion moves objects on the screen.</td>
</tr>
</tbody>
</table>

(Table from Hinckley Reading)
Mouse

(Figure from Hinckley Reading)

<table>
<thead>
<tr>
<th>State</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td><em>Out Of Range:</em> The device is not in its physical tracking range.</td>
</tr>
<tr>
<td>1</td>
<td><em>Tracking:</em> Device motion moves only the cursor.</td>
</tr>
<tr>
<td>2</td>
<td><em>Dragging:</em> Device motion moves objects on the screen.</td>
</tr>
</tbody>
</table>
Touch Screen

(Figure from Hinckley Reading)

<table>
<thead>
<tr>
<th>State</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td><em>Out Of Range</em>: The device is not in its physical tracking range.</td>
</tr>
<tr>
<td>1</td>
<td><em>Tracking</em>: Device motion moves only the cursor.</td>
</tr>
<tr>
<td>2</td>
<td><em>Dragging</em>: Device motion moves objects on the screen.</td>
</tr>
</tbody>
</table>
Stylus on Tablet

(Figure from Hinckley Reading)

<table>
<thead>
<tr>
<th>State</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>Out Of Range: The device is not in its physical tracking range.</td>
</tr>
<tr>
<td>1</td>
<td>Tracking: Device motion moves only the cursor.</td>
</tr>
<tr>
<td>2</td>
<td>Dragging: Device motion moves objects on the screen.</td>
</tr>
</tbody>
</table>
Mouse, in more detail

1. Hover
   - $dx, dy$
   - $\Delta t > T_{out}$; $dx, dy < \varepsilon_{enter}$

2. Drag
   - $dx, dy$
   - $dx, dy > \varepsilon_{drag}$

2. Left
   - $dx, dy$
   - L Button Up (Click)

2. Right
   - $dx, dy$
   - R Button Up (R Click)
(Multi-) Touch
Strengths

- Direct input allows maximal screen space for mobile devices (ocular centrism).
- More degrees of freedom.
- “Virtual input devices” are adaptable.
- No extra pieces to lose or break (styli!)
Challenges
(from Buxton)

• No tactile feedback.

• Requires free use of (both) hands and eyes.

• “Fat Finger” problems – precision & occlusion
Terminology

(from Buxton)

• Touch-tablets vs Touch screens
• Single-finger vs multi-finger
• Multi-person vs multi-touch
• Points vs Postures
• Hands and fingers vs Objects
Multi-point Gestures

Select Single: tap
Select Single: lasso
Select Group: hold and tap
Select Group, and Select Group: Use Select Single, or Select Single, on all items in the group.

Move: drag
Move: jump
Pan: drag hand
Rotate: drag corner
Object jumps to index finger location.
Finger touches corner to rotate.

Cut: slash
Paste 1: tap
Paste 2: drag from offscreen
Paste 3: Use Move 2 with off screen source and on-screen destination.

Delete 1: drag offscreen
Delete 2: Use Move 2 with on-screen source and off-screen destination.

Accept draw check
Reject draw ‘X’
Reject 1 Reject: If rejecting an object/dialog with an on-screen representation, use Delete 1 or Delete 2.

Help: draw ‘?’
Undo: scratch out
After duplicating, source object is no longer selected.

Posture-based Interaction

Golan Levin, Zach Lieberman

The Manual Input Sessions Workstation
The “Fat Finger” Problem

Graphics: Patrick Baudisch, nanoTouch
A Software Solution

scenario 1:
ambiguous target
due to occlusion

Graphics: D. Vogel, P. Baudisch - Shift
A Hardware Solution: Use the Backside

Graphics: Patrick Baudisch, nanoTouch
Hybrids: Keyboards on Interactive Tables

B. Hartmann, M. Morris, H. Benko, A. Wilson: Augmenting Interactive Surfaces With Keyboards and Mice. UIST 2009
Hybrids: Multi-touch on Mice

Mouse 2.0: Multi-touch Meets the Mouse
Nicolas Villar, Shahram Izadi, Dan Rosenfeld, Hrvoje Benko, John Helmes, Jonathan Westhues, Steve Hodges, Eyal Ofek, Alex Butler, Xiang Cao and Billy Chen.
Hybrids: SLAP widgets
Next Time

Model-View-Controller, Event-driven UIs

- *Basics of Event Handling.*
  Developing User Interfaces. Dan Olsen, Ch.4.
- Don’t forget to read and submit comment!