Keyboard: Half-QWERTY

- Naïve speed prediction for a keyboard layout
  - Two hand: Initial cap in sequence of caps = 2 strokes
  - One hand: Initial “flip” in sequence of flips = 2 strokes
  - Each cap = 2 strokes
  - Determine strokes for each keyboard layout typing a sample corpus
  - Speed of KeyboardX relative to two-hand QWERTY
    \[ \text{Speed of KeyboardX} = \frac{\text{strokes for two-hand QWERTY}}{\text{strokes for KeyboardX}} \]

Graph shows two-hand keyboard strokes normalized to 1


(This article also has a richer analysis of typing speed and the results of extended testing by a small number of users.)
Keyboard: Half-QWERTY

- Study: 10 QWERTY typists, within-subject; at most one 50-min session/day (2-hand pre/post tests, and 1-hand blocks); eyes blocked
- Exceeded hunt-and-peck performance after 3–4 hrs; after 10 hrs, all reached 41–73% of their two-hand speed

<table>
<thead>
<tr>
<th>Measure</th>
<th>Hands</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
<th>10</th>
</tr>
</thead>
<tbody>
<tr>
<td>Speed (wpm)</td>
<td>1</td>
<td>13.2</td>
<td>18.3</td>
<td>21.1</td>
<td>24.4</td>
<td>27.1</td>
<td>29.0</td>
<td>30.7</td>
<td>31.6</td>
<td>33.6</td>
<td>34.7</td>
</tr>
<tr>
<td></td>
<td>2</td>
<td>58.5</td>
<td>59.8</td>
<td>62.3</td>
<td>61.6</td>
<td>63.7</td>
<td>63.3</td>
<td>64.2</td>
<td>64.8</td>
<td>66.2</td>
<td>64.9</td>
</tr>
<tr>
<td>Errors (%)</td>
<td>1</td>
<td>5.96</td>
<td>5.13</td>
<td>8.93</td>
<td>9.70</td>
<td>9.21</td>
<td>8.98</td>
<td>7.55</td>
<td>8.23</td>
<td>7.54</td>
<td>7.44</td>
</tr>
<tr>
<td></td>
<td>2</td>
<td>3.25</td>
<td>3.49</td>
<td>2.45</td>
<td>3.05</td>
<td>3.40</td>
<td>3.55</td>
<td>3.55</td>
<td>3.55</td>
<td>3.55</td>
<td>4.20</td>
</tr>
</tbody>
</table>

E. Matias, IS Mackenzie, W. Buxton, INTERCHI 93

Keyboard: Hardware Device Issues

- Keyboard design
- Keyboard layout
  - CR, BSP, DEL, CTRL, FN
  - Cursor keys
  - Number pad (123 on top vs. bottom)
  - QWERTY, DSK,…
- Key size/shape
  - Flat vs. contoured top
- Key type
  - E.g., membrane
- Scale
Locator

- Specifies a location
- Can be used to select, position, orient, specify path, quantify, enter text

Locator: Hardware Device Issues

- Grip
  - Stylus vs. mouse
- Time to pick up
- Active vs. passive stylus
- Mouse tracking technology
- Case study: Mouse design
  - Shape
  - Buttons
  - ...
Case Study: Mouse Design

Soap

- Optical mouse internals repackaged in lozenge-shaped plastic "core" and surrounded by cloth "hull"
- Manipulation approaches
  - "Joystick"
    - Similar to handheld spring-loaded joysticks and "upside-down optical mouse"-like devices
  - "Belt"
    - Drag hull with core stationary
  - "Soap"
    - Flip core with hull stationary

Three-State Model


- **Mouse**
  - Two states: *tracking* (button up), *dragging* (button down)

- **Touch Tablet/Pad** (separate from display)
  - Two states: *out of range* (not touching), *tracking* (touching)
  - State 0 is used to designate a state the system cannot sense
  - State 1–State 0 transition = “pen has left the paper”
    - Does this generate an explicit event?

Note: Actual devices shown in photos may have a superset of the states listed here

---

Three-State Model


- **Stylus with Tip Switch**
  - Three states: *out of range* (not touching), *tracking* (touching, with switch open), *dragging* (touching, with switch closed)
### Three-State Model


- **Multi-button Mouse**
  - Three states: *tracking* (buttons up), *dragging a* (button a down), *dragging b* (button b down)
  - Could use double clicking on a one-button mouse to distinguish among states 2a/b
  - Could also have state 2ab (both buttons down)

- Note: State names (*tracking*, *dragging*) are just examples. E.g., state 2 could instead be drawing, interacting with a menu, ...

- Note: *Tracking* is sometimes known as *hovering* (esp. when the device doesn’t change position for some set amount of time)

---

### Three-State Model


- How to support selection in 0–1 state device?
  - Add a button to device
  - Use other hand on other device (e.g., to push a button)
  - Use state 1–state 0 transition (liftoff)
  - Use double tap/click
  - Use timeout cue (point at object for time $t \geq t_{\text{select}}$)

  - Use pressure threshold (if device can detect >1 level of pressure)
  - Use number of fingers or area of touch (if device can detect)

- Direct input devices (e.g., touch screen)
  - *Out of range* state 0 actually supports passive tracking since unsensed pointing device/finger can act as its own cursor
  - Not state 1, since system cannot sense it

- In contrast, compare with the [indirect] touch tablet/pad (in which finger on tablet/pad cannot be viewed in context of screen in state 0)
  - Same gesture, different context


- Mouse
- Legacy AM/FM radio
Fitts’s Law  P. Fitts, 1954

- Predictive model of time $MT$ to move a distance $A$ to target of width $W$.
  - $MT$ increases with increasing $A$, decreases with increasing $W$
  - Farther/smaller target $\rightarrow$ longer time to reach
    Closer/bigger target $\rightarrow$ shorter time to reach
  - $MT = C_1 + C_2 \ ID$
    - $ID$ = Index of Difficulty (function of $A$ and $W$)
    - $C_1$ = Device/appendage-dependent constant
    - $C_2$ = Device/appendage-dependent constant

Fitts’s Law  P. Fitts, 1954

- Original task used electrical contacts
- Parameters varied from $A = 1''$, $W = 1''$ to $A = 16''$, $W = .25''$
- $ID = \log_2 \left( 2A / W \right)$
  - Conventionally measured in bits, after Shannon
- $ID = \log_2 \left( A / W + 1 \right)$
  - Later formulation has slightly better fit, and assures positive $ID$ (Mackenzie)
Fitts’s Law  P. Fitts, 1954

- $MT = C_1 + C_2 \ ID$
  - where $ID$ measured in bits
- $C_2$ measured in secs/bit, ca. .1 sec/bit (range ca. 83 msec/bit – 430 msec/bit)
  - E.g., higher for button-down dragging
- $IP$ (Index of Performance) = $1 / C_2$
  - Measured in bits/sec (ca. 12 – 2.3 bits/sec)
  - Also known as throughput or bandwidth
- $MT = C_1 + ID / IP$