Interaction Device Categorization
Coordination of DOF

- **Separable vs. integral**
- **Example (from W. Buxton)**
  - **Etch A Sketch**
    - 2 *separable* positional DOF
    - separate control knobs
    - Good for vertical/horizontal lines
  - **Skedoodle**
    - 2 *integral* positional DOF
    - integrated control joystick
    - Good for free-form doodles
    - Templates provide constrained control
Interaction Device Categorization
Transfer Function

- Mapping of properties sensed to system action
  - Qualitative
    - Position (aka position control, zero-order control)
      - Isotonic devices better than isometric/elastic devices for position control
      - Position $\rightarrow$ position
      - Force $\rightarrow$ position
    - Velocity (aka rate control, first-order control)
      - Isometric/elastic devices better than isotonic devices for rate control
      - Force $\rightarrow$ velocity
      - Position $\rightarrow$ velocity
  - ...
  - Quantitative
    - Linear...
    - Nonlinear...
    - Variable...
    - ...

Stimulus–Response (S–R) Compatibility

- Whether mapping of user stimulus (e.g., limb motion) to system response (e.g., cursor motion) is compatible with regard to
  - Direction
  - Orientation
  - ...
  - ...
Control-to-Display (C/D) Ratio
[more generally, Control-to-Response (C/R) Ratio]

- Ratio of movement of input device (user stimulus) to movement of controlled display object (system response) for multiplicative transfer function

Alternatively, $CD\ gain = (C/D\ ratio)^{-1}$

- High C/D ratio $\rightarrow$ accuracy
- Low C/D ratio $\rightarrow$ speed, space savings

Control-to-Display (C/D) Ratio
[more generally, Control-to-Response (C/R) Ratio]

- Variable C/D ratio
  - E.g., typical mouse transfer function
    - high-speed $\rightarrow$ low C/D ratio
    - low-speed $\rightarrow$ high C/D ratio

G. Casiez and N. Roussel, UIST 2011
http://libpointing.org/user-guide/why/

Note: Graph shows CD gain, not C/D ratio
An Early GUI Framework: Interaction Tasks J. Foley et al., 80s

- Basic interaction task is the input by user of atomic “unit” of info
  - Text
  - Select
  - Position
  - Quantify
  - Orient
  - Path

- Basic interaction task is accomplished by
  - basic interaction technique
    - Approach for performing basic interaction task
  - using logical device
    - Abstraction of a device based only on its output
      - String, choice, pick, locator, valuator, stroke
  - implemented by physical device(s)
    - Actual device with characteristic properties
      - Keyboard, mouse, rotary control, linear slider, joystick, touch screen, scroll wheel, buttons,…
An Early GUI Framework: Interaction Tasks J. Foley et al., 80s

- Composite interaction task combines basic interaction tasks
  - Dialogue boxes
    - Specify multiple units of information
  - Construction
    - Create objects
  - Manipulation
    - Modify objects (e.g., by scaling)

An Early GUI Framework: Interaction Tasks J. Foley et al., 80s

- Problem: The notion of a “basic interaction task” accomplished by a single logical device is an oversimplification
- Best suited to “classical” devices processed by simple computation
- What about
  - Multi-touch?
  - Gesture?
  - Full-body?
  - Face?
  - Eyes?
  - Pulse?
  - Fingerprint?
  - Emotion?
Simple Gesture Recognition

$1$ Unistroke Recognizer

For each $T$ (template) and each $C$ (candidate to match)
- Resample point path to $N$ equidistant points
- Rotate "indicative angle" (vector from centroid to first point) to $0^\circ$
- Scale to reference square and translate centroid to origin

For each $C$ (additional steps)
- Refine rotation to compute (heuristic) minimum distance between corresponding points of $C$ and each $T$ (GSS = Golden Section Search)
- Select $T$ with smallest distance (best match)

Keyboard: QWERTY

- First used on typewriter by C. Sholes, C. Glidden, and S. Soulé, early 1870s
- Depicted in US Patent 207,559, filed 1875

Note: No shift key. This early machine typed capitals only.
Keyboard: QWERTY

- Designed to prevent jamming of keys struck in sequence
- Based on digraph frequency analysis

Keyboard: DSK (Dvorak Simplified Keyboard)

- Developed by A(ugust) Dvorak, based on time-motion studies, n-graph frequency analyses
  - Increase time fingers spend on “home keys”
  - Increase alternation between hands, fingers
- US Patent 2,040,248, filed 1932
- Vehement opinions for/against (e.g., http://www.dvorak-keyboard.com/, http://www.utdallas.edu/~liebowit/keys1.html)
- But, is it faster?
Keyboard: DSK (Dvorak Simplified Keyboard)

- Checkered history of comparison studies with QWERTY
  - 1943 US Navy studies “show” DSK faster
  - 1956 US General Services Admin study (Strong) “shows” QWERTY “brushup” practice more effective
    - Original data destroyed, possible experimenter bias
- Theoretical comparisons suggest 2.3–17% improvement
  - Studies of simulated typing based on digraph speed
- Anecdotal practical comparisons claim improvement
- Try it yourself

Keyboard: Half-QWERTY

- E. Matias, I.S. MacKenzie, and W. Buxton, INTERCHI 1993
- Builds on
  - User’s knowledge of QWERTY
  - L-R symmetry

Try the demo: http://www.matias.ca/halfkeyboard/demo
Keyboard: Half-QWERTY

- Keys of one hand remain the same
- Spacebar acts as shift to mirror image keys
- Press-release of spacebar with no other char
  - before timeout is a space
  - after timeout is a no-op
- Modifier keys (e.g., shift, ctrl) work as “sticky keys”
  - Press once to modify next key only
  - Press twice to lock and press again to release