Input: Hardware and Software
IAT351

Week 5 Lecture 1

6.02.2008

Lyn Bartram

lyn@sfu.ca

Some slides adapted from K. Edwards, Georgia tech

SCHOOL OF INTERACTIVE ARTS + TECHNOLOGY

Where we are….

- Two largest aspects of building interactive systems
  - Output basics
  - Now we need to look at input
- Generally, input is somewhat harder than output
  - Less uniformity, more of a moving target
  - MUCH more affected by human properties
  - Not as mature
- Need some way of characterising input dimensions and tasks
  - Logical devices ---> low level devices --> higher level

After this week ?

- Understand space of input devices with respect to tasks
- Understand event-driven programming model (Friday)
- Understand how to build virtual devices from physical input
  - Logical device capabilities
  - Physical input properties
  - using event listeners in Java

Interaction frameworks

- Atomic / Elemental tasks - interaction primitives
  - Select, position, orient, path, quantify, text entry
- Compound tasks
  - Chunking and phrasing of tasks
  - Gestural and Marking techniques
  - Multi-Channel Devices; e.g. scrolling.
- Higher-level primitives: sampling, scanning, and high-dimensional input; e.g. multi-touch tablets, scanners, haptic lens, "shape tape," motion capture, etc.
  - Foley and Van Dam, Fundamentals of Computer Graphics

Devices, tasks and techniques

- *Input device* communicates **sensed** information about the physical environment
- Coupled intimately with output for
  - **Feedback**
  - Virtual/specialised input (such as a scrolling control)
- *Interaction tasks* are low-level (elemental) inputs required from the user (such as entering a text string)
- *Interaction technique* is the fusion of input and output (hardware and software) to accomplish a task
  - Scroll through a document: click or drag a mouse (input) on a scroll bar (output)
  - Entering text: click a mouse (input) on an image of a keyboard (output)

Basic 2D Interaction Tasks

- Position/locate: specify a position to system
  - Pointing (spatial)
  - Symbolic specification (linguistic)
- Select
  - Variable
  - Fixed
- Text (symbols)
  - Keyboard
  - Handwriting
- Quantify
  - Spatial (relative)
  - Linguistic (absolute)

Logical devices

- Logical input devices
  - categorise how the task is accomplished

- 4 basic devices:
  - Keyboard
  - Choice
  - Valuator
  - Locator

SCHOOL OF INTERACTIVE ARTS + TECHNOLOGY

Input devices: text/symbol

- Variety of input devices
- Traditional keyboards (‘hard’ keyboards)
- Soft keyboards (on screen virtual keyboards)
- Speech input recognition
- Handwriting (see pen devices)
- Keyboard
  - Ubiquitous, but somewhat boring
  - Mature design

Traditional physical keyboards

Keyboard - discrete entry device

- QWERTY
  - originally designed to prevent typewriter keys jamming from adjacent key presses
- DVORAK
  - layout arranged based on the frequency
  - the frequency of letter patterns and language
  - hands (particularly the index fingers) rather than repetitive typing with one finger
- Chord
  - several keys must be pressed at once
  - small number of keys needed

Properties

- Procedural memory allows repetitive act to be “encoded”
- Touch typists can rapidly type on layout without interfering with mental composition of text
- Changes involve high re-learning
- keyboard-mouse text entry for the English language is more than twice as fast as automatic recognition of dictated speech
- No visual attention
- Provide tactile feedback
- Parallel input

Important Properties

- Size
- Shape
- Activation force
- Tactile and auditory feedback
- Ergonomic layout
  integrated pointing devices
- Miniature keyboards
- Additional extended functionality
- Wireless
- Small devices (cell phones! PDAs!)

QWERTY Keyboard layout

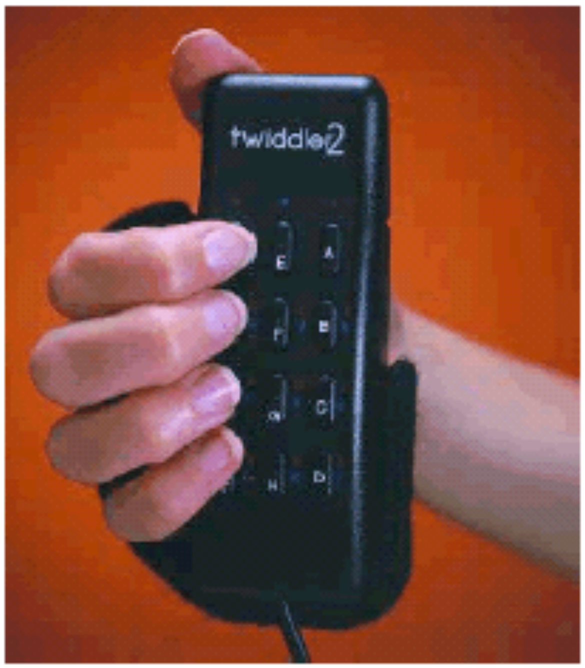
- Other layouts have been proposed
- Dvorak is best known
  - Designed to accommodate frequency of key presses
  - Widely touted as better

- Experimental evidence and theoretical analysis casts doubt on this
  - Alternating hands of QWERTY are a win since fingers move in parallel

- Note : What do we mean by theoretical approach? We'll get back to that

The Maltron keyboard: radical design sculpted to fit shape of hand and fingers.

Commercial Chord Keyboards



Retrieved Feb. 24, 2003 from
http://www.handykey.com/site/twiddler2.html

Retrieved Feb. 24, 2003 from
http://www.infogrip.com/product_view.asp?RecordNumber=12&sbcolor=%23FF9966&option=keyboard&subcategory=&CatTxt=&optiontxt=Keyboard

Other keyboards

Soft keyboards

- Require constant visual monitoring
- Lack tactile feedback
- No parallel input

Is there a future for the traditional keyboard?

- keyboard-mouse text entry for the English language is more than twice as fast as automatic recognition of dictated speech
- Whether or not Dvorak was better, it did not displace QWERTY
- Lesson: once there is critical mass for a standard it is nearly impossible to dislodge (even if there is an apparently good reason to do so)
- What happens with “new” devices?

Very small keyboards and ‘thumb’ entry

Cell phone keyboards for

- Numeric entry
- Some control functions
- TEXTING

Very small keyboards and ‘thumb’ entry

- Text entry techniques are either
  - Iterative (MultiTap, LessTap)
  - Predictive (T9)
- Accommodate numeric entry and dialpad legacy (overload not redefine)
  - alphabetical
- Text entry on cell phones is 2 to 5 times slower than on traditional keyboards

Very small keyboards and ‘thumb’ entry

- Multitap:
  - Tap until you get to the letter in the set assigned to that keyboard
  - Just need to remember alphabetic layout
- LessTap
  - Keys assigned a frequency and you enter the frequency index in taps
  - Need to remember frequency and layout
- T9
  - Enter words with fewer keypresses
  - Offers up most frequent words and you cycle through them

Key factors/properties

- MultiTap, LessTap:
  - Don’t need to look at the screen
  - (bad thing?)
- T9
  - Faster – but no motor memory
- Generational differences
  - Thumbing facility
  - Video game controllers

Very small keyboards and ‘thumb’ entry

- http://www.digitwireless.com/about/comparison.html

Ink and handwriting recognition

- Handwriting is much slower than typing or thumbing
  - Even on paper
- Handwriting recognition (OCR) still has latency issues on many platforms
- “naturalness” is appealing
- Arm/hand position is critical in both ease of entry and accuracy of signal
- Often systems use a compromise approach based on a smaller library of recognizable strokes
  - Palm Pilot Graffiti™

Buttons

- Enact Choice
  - Redio buttons, checkboxes, dialogue buttons ……
- Similar to keyboard but for entering symbols
  - Function keys
- Buttons are often bound to commands
- Would like software changeable labels on hardware buttons but this does not typically occur!
- Fundamental aspect of virtual (on-screen) buttons

Buttons

- Act as “meta” enablers

- Mouse buttons
  - Selection
  - Additional choices

- The one-button vs. 3 button mouse debate!

- In hardware, the button is the esential choice device
- In software?

Valuators

- Returns a single value in a range

- Major implementation alternatives
  - Potentiometer (variable resistor)
    - Similar to a volume control
  - Shaft encoder
    - Sense incremental movement

- Differences? (hint - this is important)

Valuator alternatives

- Bounded vs unbounded (absolute vs relative)

- Potentiometer
  - Normally bounded range of physical movement
  - Hence bounded range of input values

- Shaft encoder
  - Unbounded range of movement
  - No residual position of device
  - “Deltas”, relative range

Locators

- AKA pointing devices
- Returns a location (point)
  - Two values in range
  - Usually screen position - but not always
- Examples
  - Mice
  - Trackballs, tablets, joysticks, touch panels …….

Locators

- 2 major categories
  - Absolute vs. relative locators

- Based on input type
- It’s actually a little more complex than that, but let’s start with this

Absolute locators

- One-to-one mapping from device position to input value
  - Faster
  - Doesn't scale past fixed distances
  - Bounded input range
  - Less accurate (for same range of physical movement)
  - Examples?

Relative locators

- Provides the DIFFERENCE (delta) value between previous position and current position
- Maps movement into change in input
- Examples?

Relative locators

- More accurate (for same range of movement)
- Harder to develop motor skills
- Unbounded (can handle infinite spaces)
- Many physical parameters to consider

But there is actually more

- Q: is a mouse an absolute or a relative locator?

But there is actually more

- Q: is a mouse an absolute or a relative locator?
- A: No.
- There is a third major type:
- “Clutched absolute”.
  - Within a range it can be absolute (the mouse pad, the window)
  - Can disengage movement (pick it up) to extend beyond range

  - Picking up == clutch mechanism

Clutched absolute locators

- Very good compromise
- Get one-to-one mapping when in range
  - Easy to learn
  - Efficient
- Allow infinite range outside current local range context
  - Pick it up and reset
- What about trackballs?

Input > device: reprise

- Design challenge is to fit interaction tasks to input devices and interaction techniques
- Optimal choice for each elemental interaction task may lead to a poor design
  - Inconsistent and cluttered
- Multiple choices for a particular task
  - Mouse/command selection options
  - Popup menus and double clicking, circling, targeting …
- Design space: Compromise and constraints

Elemental tasks revisited

- **Select**: choose from set of alternatives
- **Position**: specify a position within a range (includes *picking)*
- (**Orient**): specify an angle or a 3D orientation (rotation)
- **Path**: specify a series of positions or orientations over time
- **Quantify**: provide an exact numeric value
- **Text**: enter symbolic data
- **Problem**: level of analyis is ill-defined!
  - Position can involve quantify (for purposes of precision)

Elemental tasks are too particulate

- Elemental model only maps to explicit interaction
- Doesn't provide analysis for goals of compound task
  - Etch-A-Sketch™ vs pen input for drawing
  - constraints, goals and methods
- Need to consider compound tasks *:*
  - Combinations of basic tasks
  - *Chunking:* higher level of the compound task
- The choice of device influences [and is influenced by] the level at which the user is required to think about the individual actions that must be performed to achieve a goal[1].
  - [1]Hinckley, K., Jacob, R., and Ware, C. Input/Output Devices and Interaction Techniques.

Properties of input devices in the 2D context

- But what (else) do we need to understand about input technologies to best map them to the user's task and capabilities?
- The choice of device is influenced by the important properties of the device related to physical sensor, feedback, ergonomics and interaction techniques easily supported.

Important Properties of input devices

- Common categories of input devices :
  - Continuous: sense a continuous range
  - Discrete: sensing one of *n* possible states (on/off)
- Continuous pointing devices
  - Physical property sensed
  - Transfer/transducer functions
  - Dimensionality
  - Direct/indirect control
  - Metrics of human performance
  - Device states supported (Buxton)
  - Engineering parameters (device performance)

Pointing devices: physical property sensed

- Traditional pointers sense position, motion, force or displacement
- **Position** sensing = absolute input (tablet)
  - Returns a coordinate in the 2D space
  - Absolute devices can support relative positioning (changes to position)
- **Motion** sensing = relative input (mouse)
  - Returns a delta from previous state
  - Relative devices emulate position with a screen cursor but can't fully support absolute positioning
- **Force** sensing: degree of force (isometric joystick)
- **Displacement** sensing: angle of displacement (isotonic joystick)

Pointing devices: physical property sensed

- Comparative performance issues:
  - Absolute devices can have the *nulling problem*, where the position of a device is not in agreement with a value in the system (comes from multiple mappings of the control)
- Doesn't occur with relative devices, but time can be wasted in clutching:
  - Picking up and repositioning a mouse
- ? Can you think of others ?

Transfer functions

- Mathematical function that scales the data from an input device to provide more stable, smooth and efficient operation
- mappings need to match the physical properties sensed
  - **Force-to-velocity** (isometric joystick):
    - force is mapped in a non-linear transformation to cursor movement speed
    - Rate mapping
  - **Position-to-position**: used for absolute positioning
    - (typically a linear scale, usually 1:1)
  - **Velocity-to-velocity**: mouse/relative
    - Common inappropriate mapping:Mouse cursor position to scrolling speed
      - No feedback of when scrolling will accelerate, hard to learn and control

Transfer functions cont.

- Device gain is a common multiplicative function
- Control-to-display (C:D) ratio
  - C: movement of input device
  - D: movement of controlled object
- Typically the gain is rarely constant:
  - Acceleration function varies gain according to velocity of C.
- Primary benefit: reduce the footprint (physical movement space) required for controlling the display space
- Concern: Gain confounds display size and device size
  - Speed/accuracy tradeoffs
- ? Discussion: when might acceleration be critical? When might it be an impediment?

Number of dimensions

- Devices measure one or more linear/angular dimensions.
  - Translation, rotation
- Degree of Freedom (DOF): # of independent position variables that would have to be specified in order to locate all parts of the mechanism
- Dimensionality: how many linear dimensions the device measures
  - A mouse is a 3 DOF device that is inherently 2D
- Multi-dimensional device senses >1 integrated dimensions
- A position sensing device could be a 1D slider, a 2D pointer or a 3D tracker…

Dimensions != channels

- If the number of dimensions required by the user's task exceeds the dimensions of the device then extra handling (device states,mouse buttons) and interaction techniques will need to be introduced.

- Multi-channel device –
  - provide multiple controls on a single device
  - Supports compound tasks
  - Avoid the need to move visual attention from task focus
  - Support control and meta-functions
  - Physical structure of some devices limits multiple channels

Input device states

- Users need to signal selection vs targeting/moving vs stopping
- Buxton's 3 states of input
  - Tracking, dragging, out of range
- Most pointing devices signal only 2 of the three

- Special procedures and/or interaction techniques needed to simulate the desired functionality
- Prone to error and inadvertent state change

Control

- Indirect control: user moves a device on a separate surface to control the position of a pointer on the display surface
  - Mice, tablets, …
  - Require a mechanical device that can be lost or damaged
  - Poor mappings can make these hard to learn and control

- Direct control: the input surface is the output surface
  - Touchscreens, light pens, ink input
  - kiosks
  - Issues of parallax (gap between input and display surfaces)
  - Occlusion (hands hide input)
  - Blurred displays

  - ? Can you think of other issues?

Device performance

- A transducer is:
  - Any device that converts one form of energy into another form of energy, specifically when one of the quantities is electrical. Thus, a loudspeaker converts electrical impulses into sound (mechanical impulses), a microphone converts sound into electrical impulses, a solar cell converts light into electricity, etc

- Engineering parameters (sampling, resolution, accuracy and linearity) have a direct impact on device and user performance.

metrics of device performance

- Resolution : Smallest incremental change that can be recognised
  - Mouse/touch: how fine the grid is (distance between points)
- Accuracy: how precisely it reports
  - Always less than resolution
- Lag/latency: total delay time between a user action and the system response
- Latency is a major issue, as a latency of 75 ms significantly impairs human performance generally using pointing devices
  - Smaller latencies seem to be problematic in haptic feedback

metrics of device performance cont.

- Sampling Rate
  - How often it collects/generates input
  - Measured in Hz (times/sec) number of samples per second, per channel
  - Only relevant to continuous devices such as mice or sensors
- Frame rate: the set of simultaneous samples in all channels at an instant in time
  - May be slower than sampling rate since it incorporates multiple samples but is often the same
- monitor refresh rate : How often the feedback is refreshed on screen

problems of device performance

- Noise
  - Unwanted signals in the system (unused data)
  - A measure of how clean the data is
  - Confounding information

- Aliasing
  - if the sample rate is not fast enough, the presence of totally nonexistent frequencies/values may be indicated
  - Introduces artifacts
  - Insufficient sampling or resolution
  - “smooth out” effect – anti-aliasing
- sensor nonlinearities
  - Analog devices are prone to variations in accuracy

- More on these when we study sensors

Metrics of effectiveness: human performance

- Pointing speed and accuracy
  - Fitts' law models of particular device performance

- Device acquisition time
  - Time to pick up, "start"/adjust to, and put down the device
  - *Homing*
- Learning time
- Footprint
  - How much space it takes up
  - Where in space it is
- ergonomic factors (how comfortable and adapted it is to use)

Human performance: movement vs force

- an *isometric* device connects the human limb and machines through force/torque while an *isotonic* device does this through movement.
  - Force joystick (trackpoint™)
  - Traditional mouse
- Spectrum : Isotonic (zero or constant resistance) to isometric (infinite resistance)
  - Spring-loaded or elastic: resistance increases with displacement (zero-order)
  - Viscous: resistance increases with velocity(1-order)
  - Inertial: resistance increases with acceleration (2-order)
- Which is “better”?

Human performance: transfer function

- *Position control* (also known as 0^{th} order motion)
  - User controls input position directly, either by relative or absolute method
- *Rate control* (also known as 1^{st} order motion)
  - maps human input to the *velocity* of the object while specifying a direction
- Position and rate control are superior to higher-order control such as acceleration control (2^{nd} order motion)

Human performance: position vs rate control

- position control can be considered more direct (more isomorphic) than rate control *as long as what is being controlled is the position*
  - No inherent movement smoothing (all limb movements transferred)
  - Maximum operating range is limited and forces clutching
  - Very difficult for small footprint devices
- Rate control provides smoother motion
  - more precision over velocity
  - Infinite range

There's an interaction!

- Studies looked at performance of devices for different mappings in a docking task:
  - isotonic vs isometric device
  - Position vs rate control
- Rankings were: (best – worst for performance)
  - Isotonic position
  - Isometric rate
  - Isotonic rate
  - Isometric position
- The compatibility principle: device performance is a function of both physical performance and mapping function.
- Match the transfer function to the task almost more than the device to the task?

So – aren't all pointing devices similar?

- Operating systems treat all devices as virtual
- How do particular properties differentiate them?

The mouse

The Original Mouse by Engelbart and English.

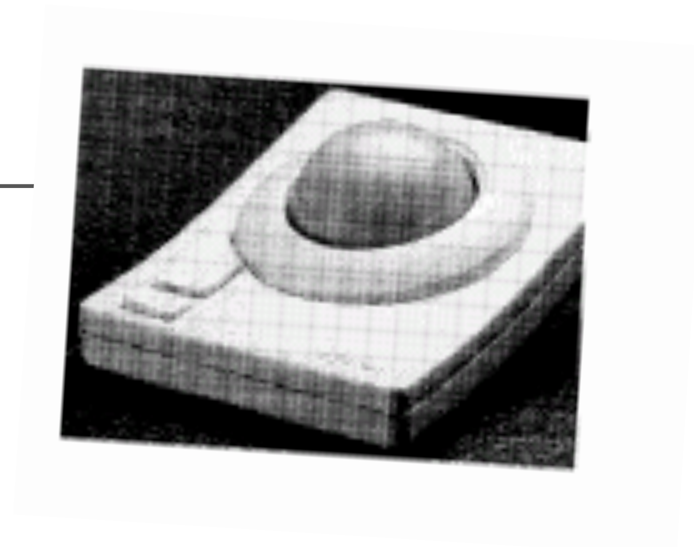
The mouse

- 2D device that senses movement relative to a flat surface.
- Properties match demands of desktop graphical interfaces
- It's stable: doesn't fall over or reposition itself (see a stylus)
- Integrates buttons
  - A mouse is not just a mouse …. It's a phone!
- Force to move it is orthogonal to movement plane, so it minimises accidental motion interference and clicking
- Doesn't tend to move when button clicked (muscle tension required to press the button has minimal interference with cursor motion compared to other devices)
- High performance for both rapid, coarse and slow, precise movements
- Fitts' law shows users can point as well with the mouse as with the hand
- ?? So what doesn't it do ??

Mouse cont.

- Mechanical :
  - Used a rolling wheel
  - Heavier and larger
  - Precision not particularly good
- Optical
  - Wide Beam of light and optical sensor
  - Higher resolution and accuracy
- Laser
  - Refinement of optical
  - Narrow beam of light

Trackballs



Trackballs

- Mechanical mouse turned upside down
- Rolls in place
  - No clutching per se
  - Small footprint
  - Boundary/constraint conditions
- Small footprint
- Can be mounted appropriate to ergonomics of station
  - Inclined surface
  - More comfortable, less repetitive strain
- Thumb, index finger – not shoulder, arm
- Digit precision better than arm precision

Trackballs

- Some indication that slower than mice
- Hard to manipulate integrated buttons
    - Buxton case study where limitation was desirable
    - Two-handed control for 3 states of input
- Size of cue ball and speed seem critically important
- ?? And they are bad/good at??

Tablets

Digitising tablets with stylus and puck

Tablets

- Most common absolute positioning device
- Can operate in relative mode or absolute mode
- Sense the position of a mechanical control object (stylus, puck)
- Physical affordances of control object are key
- Stylus is excellent for tracing, drawing, inking and free-hand input
  - Not stable in position
  - Can't accommodate many buttons
- Puck is equivalent to a mouse
- Tablets exploit kinesthetic memory
- Resolution of targeting dependent on physical surface (little gain, no velocity)

Tablets

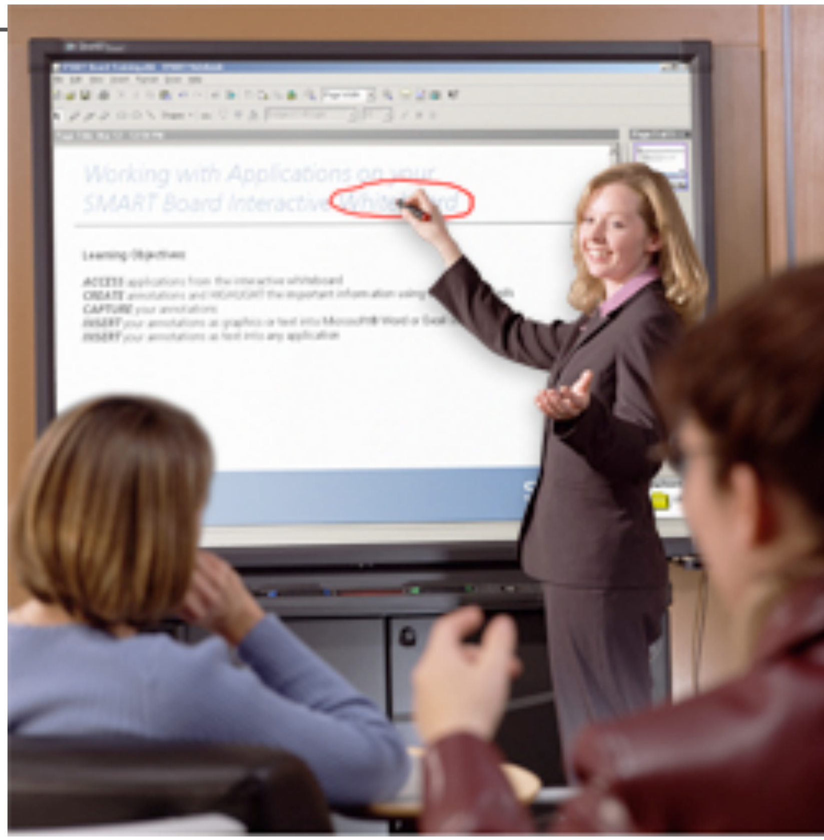
- Touch tablets sense position of the finger
- Touchpads are miniature touch tablets
- Resolution of pointing is limited by finger and needs to be handled

Touchpads

Touchpads

- Small touch-sensitive tablets
- Most commonly integrated into notebooks or smaller computing devices
- Typically respond in relative mode
  - Substantial clutching issues
- Some support absolute input mode
  - Asian characters or handwriting recognition
  - Requires a stylus
  - Overloads the device to separate control dimension from data dimension

Touch screens

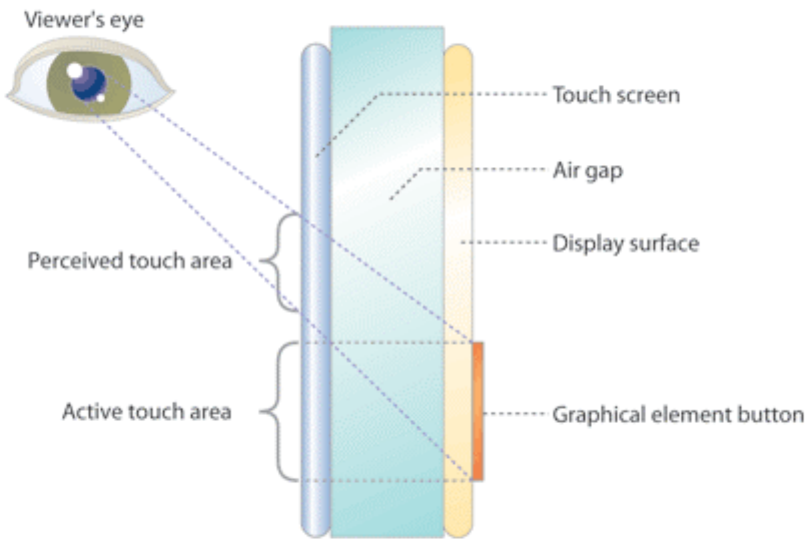


Wall-mounted touch screen (SMARTBoard™)

Touch screens:

- Direct-input device: the input surface is the display surface
  - Transparent touch-sensitive tablets mounted over a display
- Were most common as large screen displays and kiosks
- ATMS
- No moving parts to get lost
- Parallax can be an issue (see following)
- Need extra techniques to differentiate between selection and dragging
- Traditional touch screens can't sense "hover"
- Simultaneous touch resulted in averaging effect
- Arm fatigue and imprecise positioning

Parallax



- Parallax: the difference between the apparent position of an object on the screen and its actual active position on the touch panel.

Touch screens:

- Like touchpads, need extra work for select
  - Time on object
  - Occasions erroneous input
- Need to offset the selected position and information above the fingertip or pen
  - Displacement mapping function that needs to be carefully tailored
  - Vertical mounting requires different displacement than horizontal or angular
- Newer technology is vision based
- Can detect “hover” for tooltips and selection
- New ones combined with pen input

Pen input

- Touch screen using a stylus
- Inking, marking and gestural input
- Direct device – so no indication of selection position
- Hard to integrate buttons
- Pen dwell time can be used
- Need a double tap interaction technique

- Some tablets (TabletPC) use an inductive sensing technology that requires a special stylus
- Provides cursor feedback (tracking state) and hover
- Lacks the distinctive property of actually “touching” something to write or draw on!

Joysticks

Floatin
g

Self-Returning
(Measurement Systems)

Isometric

Joysticks

isotonic

- Floating:
  - absolute, position-sensitive coordinates.
- Spring-loaded/elastic
  - change is proportional to distance and direction of the shaft's offset from centre.
- isometric
  - change is determined by direction and magnitude of the force applied to the shaft.

Joysticks

- Recall performance discussion
- Have a very small footprint, so used when space is at a premium
- Isometric positioning requires significant practice
- Design of the stick affects isotonic feedback for displacement

Lots of other devices

- Still mostly KB+mouse, but increasing diversity
- Cameras
  - Lots of untapped potential in vision
  - Video (CCTV) presents HUGE scale problems
- Gesture
  - “natural” location mapping
  - Control/selection?
- Audio
  - Speech/sound as data
  - Speech as control