COMP327
Mobile Computing
Session: 2014-2015

Lecture Set 8 - User Interface Design

In this Lecture Set

- Human Computer Interaction
- General Principles
- PC vs Mobile Devices
- Interfaces
- Text Entry
- Touch and Gesture

The Challenges in design for Mobile Devices

- Mobile devices are fundamentally different to traditional PC based devices
  - PC's evolved from the notion of a desk providing a workstation surrounded by a mainframe
    - Static location, fixed wire, dynamic display, constant user attention and focus, desk-based input devices, (typically) dedicated peripheral support
  - Mobile Computing Devices broadly emerged from hand-held wireless phones
    - Dynamic location, (almost) always available wireless connectivity, intermittent user attention, limited real-estate to support input devices, handheld and movable, constrained display service, dynamic peripheral access within environment
- Differences in devices affects their interaction with the user
  - Hardware and software interfaces guided by HCI principles...

Human Computer Interaction (HCI)

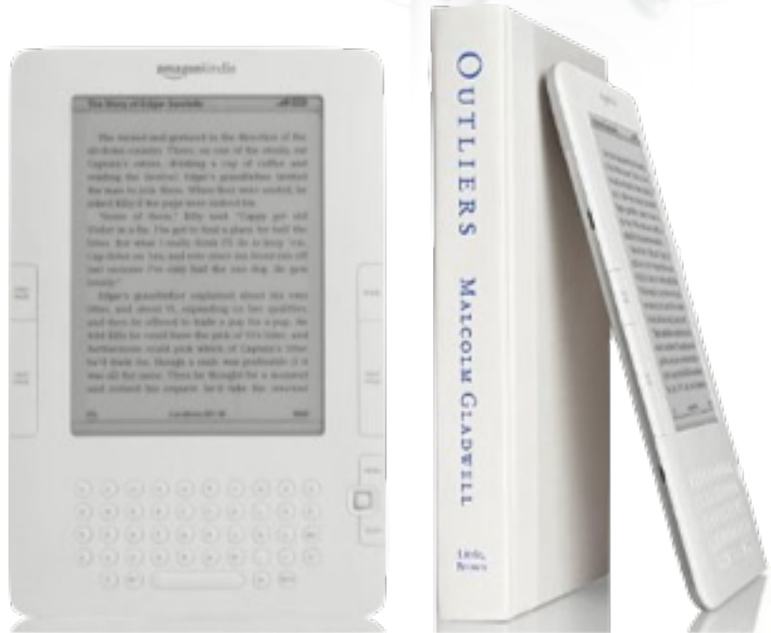
- HCI considers the man-machine interface
  - Three basic concepts:
    - ***Humans***: single or multiple users, with diverse abilities, acting competitively or cooperatively
    - ***Computers***: different computing devices (not just PCs)
    - ***Interaction***: through commands or manipulating virtual objects, or via speech, gesture or touch
- How can interaction with machines be improved?
  - Explicit interaction (eHCI)
  - Implicit interaction (iHCI)
- What are the different ways in which devices can interact with humans?
  - and vice versa!

The need for HCI in design

- For a given product, HCI traditionally refers to:
  - the processes and models for design
  - the operational interface
- However, poorly designed user interfaces lead to:
  - higher training costs
    - less intuitive interfaces require more time to learn or understand
    - retraining may be required for different systems that otherwise do the same task
  - higher usage costs (less efficiency)
    - tasks take longer due to having to fight with the interface
    - undermines users leaving them with no control
    - can result in higher error rates
  - ultimately lower adoption (and hence sales)
    - unless the system is fit-for-purpose and provides some value to the customer over the usage cost (both effort and financial), adoption will suffer

Motivation for HCI

- To support more effective use, an interface should be:
  - *useful*: accomplishes the task required by the user
  - *usable*: does the task easily, naturally, safely, with low error rates
  - *used*: enriches the user experience
- The success of the product depends largely on:
  - how usable the product is (i.e. utility), vs
  - the value it brings to the user
- Consider Apple's iPod...!
  - Will Amazon's Kindle be the iPod for print???



Heckel's Laws

- Heckel's Law:
    - The quality of the user interface of an appliance is relatively unimportant in determining its adoption if the perceived value is high!
- Heckel's Inverse Law:
    - The importance of the user interface design in the adoption of an appliance is inversely proportional to the perceived value of the appliance!

Whilst the usability of the interface is important, the utility of the device ultimately drives user adoption

Heckel's law: conclusions from the user interface design of a music appliance—the bassoon
Nigel Derrett, Personal and Ubiquitous Computing (2004) 8: 208–212

Goals for HCI

- The basic goal of HCI is to improve interaction between users and computers
    - by making the devices more usable
    - by making the devices receptive to the user's needs

- Specifically, HCI is concerned with:
    - methodologies and processes for designing interfaces
    - methods for implementing interfaces
    - techniques for evaluation and comparing interfaces
    - developing new interfaces and interaction techniques
    - developing descriptive and predictive models and theories of interaction

- Mobile devices present new challenges for several of these issues!

Implicit vs Explicit HCI

- Explicit HCI (eHCI)
  - concerned with the direct human interaction at set points during the devices normal operation
    - Typically context-free
    - Involves an explicit user model of the interface
      - e.g. switching on the light when walking into a dark room to retrieve an object
- Implicit HCI (iHCI)
  - the system responds to an act performed by the user, although the user act was not aimed at the system
    - System has a model of the user and their actions
    - Highly context dependent
    - Susceptible to noise
      - e.g. detecting a user walking into a dark room to retrieve an object, and automatically switching on the light

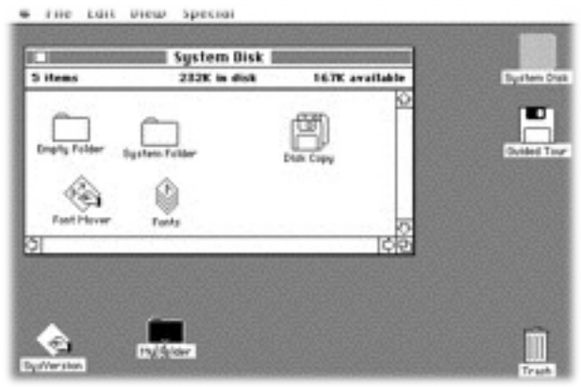
Case Study: Proximity and Ambient Light Detectors in the iPhone 3G

Sensors are used to infer both user and environmental activity.
A **proximity sensor** deactivates the display and touchscreen when the device is bought near to the face during a call. This not only saves battery power, but prevents inadvertent input from the user's face!
An **ambient light sensor** adjusts the display brightness to reduce battery power in dark environments.

http://en.wikipedia.org/wiki/IPhone

PC Interface

- Windows systems appeared as early as 1965
  - Original idea emerged in 1945 in the MEMEX system
  - Mainstream adoption not until mid ‘80ies
- Early interaction consisted of:
  - Keyboard and visual command line interface where commands were entered (terminated by a delimiter character)
    - Users needed knowledge of commands and usage
- WIMPS interface became mainstream in the ‘90ies
  - **W**indows, **I**cons, **M**enu, and **P**ointer device
  - Facilitated direct manipulation of visual objects on a display
    - Tasks achieved by activating and moving active window elements (menus etc)
    - Interaction device was mainly the mouse, but light-pen and (later) touch screens also used
  - Associated with the *desktop* metaphor



PC WIMP Interface

- Main Advantages
  - Order of commands or tasks can be much more ad hoc
  - Use of direct manipulation lowers entry barrier for new users
    - fewer commands need to be remembered
- Main Disadvantages
  - Doesn't help visually impaired
  - Consumes screen real-estate
    - problematic for small or low-resolution displays
  - Visual representations may not always be clear to all users
  - Mouse control requires good hand-eye coordination
    - Can be slow to use
  - Large number of virtual objects can be confusing and difficult to navigate

Mobile Device Interfaces

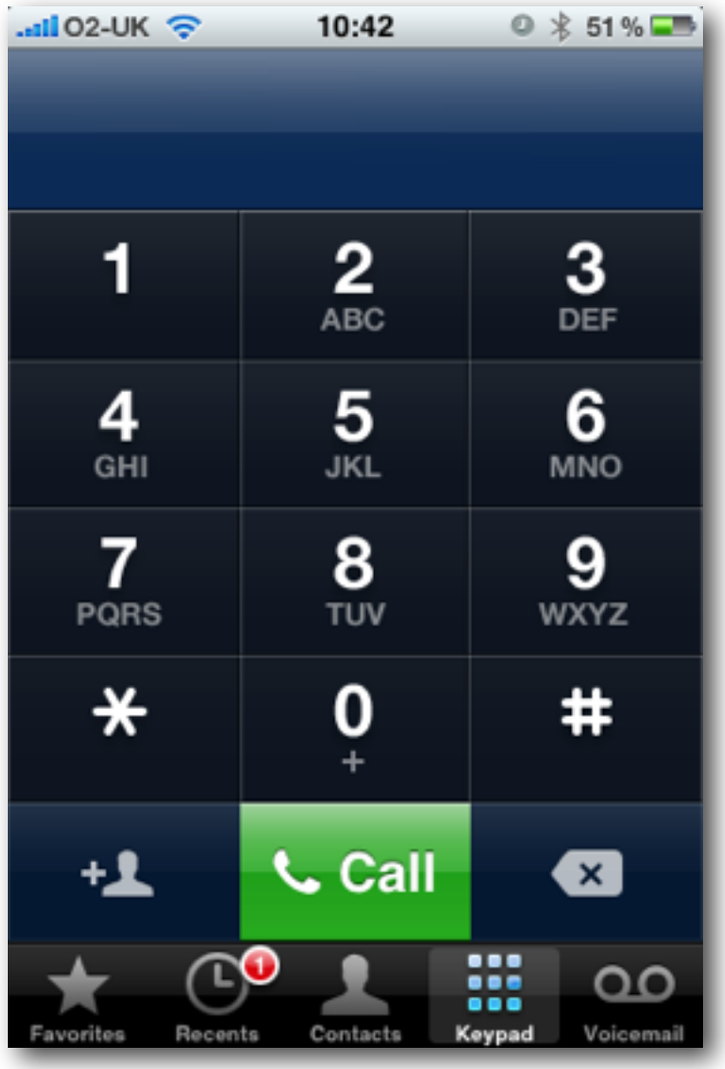
- PC style WIMPS interfaces simply don't work on Mobile Devices
  - Display area is much smaller:
    - Impractical to have several windows open on the display!
    - Difficult to find and resize windows and icons stacked on one another
- Pointer/mouse metaphor unsuited for mobile devices
  - Pointer accuracy is reduced on small screens
  - A lack of mouse requires other interaction modalities
    - Cursor keys, scroll wheel, touch screen stylus, etc
    - External portable peripherals can sometimes be used...

User input on mobile devices

- Most common input is through a numeric keypad
  - Telephone keypads bear letters
    - Originally, two digits referred to telephone exchanges
      - e.g. Aylesbury was assigned 01296 (01AY6)
      - Also used to remember numbers in the US (e.g. 1-800-PAINTER)
      - Advantage: *few keys*! Disadvantage: *few keys*!
        - Used heavily for SMS messaging with early GSM phones
- Mobile Mini keyboards
  - Simplify typing, but often too small to be used conventionally
    - Thumbing has emerged as a fast typing approach
- Touch Screens
  - Originally stylus based, but finger based approaches now common
  - Multitouch input emerging
- Other input through cursor keys, soft keys, scroll wheels, gestures, etc

Handling Limited Key Input: Multi-Tap

- Most numeric key input introduced modes
  - Multi-Tap keyboard
    - Keyboards typically have 12 keys
      - 10 digits, plus hash and star
      - Alphabet arranged over 8 keys, with 3 or 4 letters per key
    - Each letter is selected by repeatedly pressing the key until the desired character appears
  - Semantics of the key changes from context to context
  - Allowed early SMS messages to be sent from a numeric keypad
- Several predictive texting algorithms exist
  - T9, iTap, LetterWise



T9 and predictive texting

- T9 (Text on 9 keys)
  - Aims to simplify text entry by using a dictionary to predict typed word
    - Typically, only one key press per letter is required.
    - Multiple options are presented, in descending order of frequency (selected using the "Next" key)
    - Additional words can be added to the dictionary via Multi-Tap
    - Word completions can be proposed to speed up input
- Some implementations support *smart punctuation*
  - When using the number '1' key, punctuation is inserted

T9 and adapting to the user

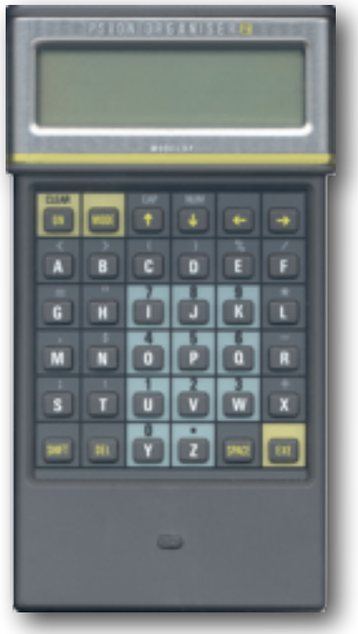
- The T9 model could be adapted to user behaviour
  - Consider the input pattern 4663
    - "good", "home", "gone", "hood", etc.
    - Known as *textonyms*
      - i.e. several words formed from the same digit sequences
    - By storing past usage, T9 can order textonym presentation based on frequency of us
  - Some implementations also learn n-grams
    - By recognising word pairs, a second word would be suggested given the entry of the first.
    - Similar phrase prediction found within iTap

3	def
4	ghi
6	mno

Numeric Keyboard input for other languages

- Q9 for Chinese
  - A "structural" method developed in Hong Kong
    - Non-Zero numeric keys display either 9 Chinese or stroke shapes
      - 0 is used to toggle between Chinese characters and stroke shapes
      - 5 corresponds to "other strokes"
    - First two keys correspond (approximately) to first two strokes
    - Third key (if necessary) corresponds to the last stroke
- Japanese Input Methods
  - Based on sequences of kana (syllabic) characters associated with different numbers
    - Repeated presses select different characters
    - Kana-to-Kanji conversion can then be performed
      - i.e. converting from syllabic/pronunciation to meaning/semantic

Mobile Mini Keyboards



- Early devices used alphabetic key ordering
  - e.g. Original Psion Organiser I and II
- Rapidly evolved to QUERTY due to PC keyboard familiarity
  - Access to keyboard typically clam-shell based
    - e.g. Psion Series 3 and Nokia 9000 Communicator

- The RIM Blackberry made use of a front-facing keyboard
  - Tiny keys provided motivation for name!
  - Aimed more at the Business Market
  - Input was primarily through "thumbing", similar to T9 etc

- More recent keyboards slide out from the side
  - Prevalent on many modern smart-phones, despite the increase in touchscreen use

Touchscreens

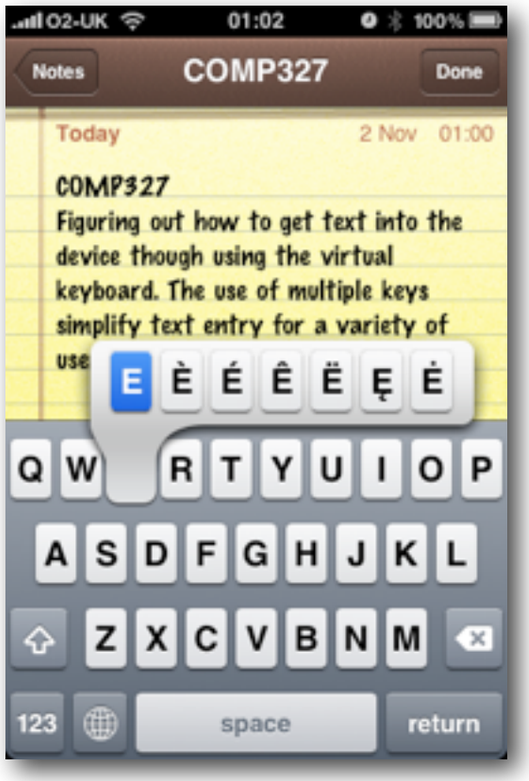
- A display that detects the presence and location of a "touch" on the display
  - Typically finger or stylus based
  - Used in a variety of settings
    - Retail, tourist information, ATMs, PDAs Game Consoles, etc
  - Until recently, could only detect one contact at a time
    - Multi-touch technology becoming prevalent
- Main technologies based on resistive and capacitive approaches
  - Other techniques include
    - Surface Acoustic Wave
    - Infrared arrays
    - Strain Gauge configurations

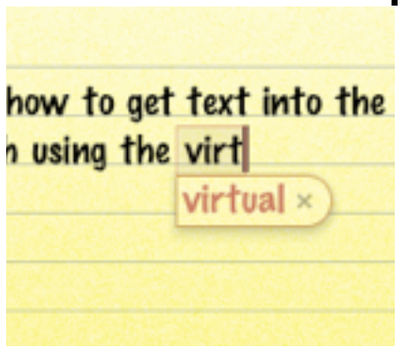
Touchscreen text input

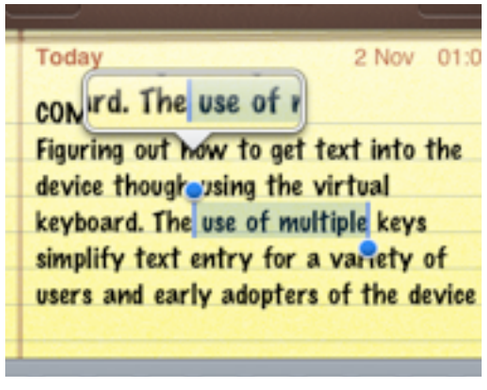
- Variety of text input methods exist:
  - *Virtual Keyboards*
    - User interacts with a keyboard that appears on the screen
  - *Single-stroke glyphs*
    - Known as Block Recognizer on Windows Mobiles
    - Assumes a set of single stroke glyphs to represent alphabetic characters
  - *Character Recognition*
    - Detects characters from multi-stroke language glyphs

Virtual Keyboards

- Graphical Approach:
  - A keyboard is displayed on the screen
  - User then types by selecting the relevant keys
  - Good for multi-lingual alphabets
- Simple to use, but requires accurate input
  - Mainly stylus only approach
  - Requires screen real-estate to display keyboard
    - Reduces the space for other content on the screen
- Apple's iPhone input uses several novel improvements
  - Predictive typing
  - Magnifier cursor
  - Cut and Paste







Single Stroke Glyphs

- Short, single stroke gestures are used instead of letters
  - Typically drawn on a touch sensitive panel
    - Often with no direct visual user feedback
    - Thus, letter glyphs are simplified
    - Recognition software simpler than character recognition
      - Thus suitable for lower-powered or slower devices
  - However...
    - Required users to learn a new set of glyphs!!!

- The best known language is Palm OS's Graffiti
  - Later replaced due to a Xerox patent infringement
  - Also runs on Windows Mobile platform

Palm's Graffiti 2

- Originally developed by Jeff Hawkins
  - Palm had seen Xerox's Unistrokes system (by David Goldberg) prior to development!
  - Thus, litigation resulted in changing the language to Graffiti 2

- Characters written on top of each other
  - Could use a dedicated touch-sensitive panel below the screen
    - Later versions also allowed writing over the screen with visual feedback
  - Avoided "running out of screen" after a few words
  - Maximised usable screen space
    - However, each letter had to be a continuous stroke
    - Pen off and on delimit character input

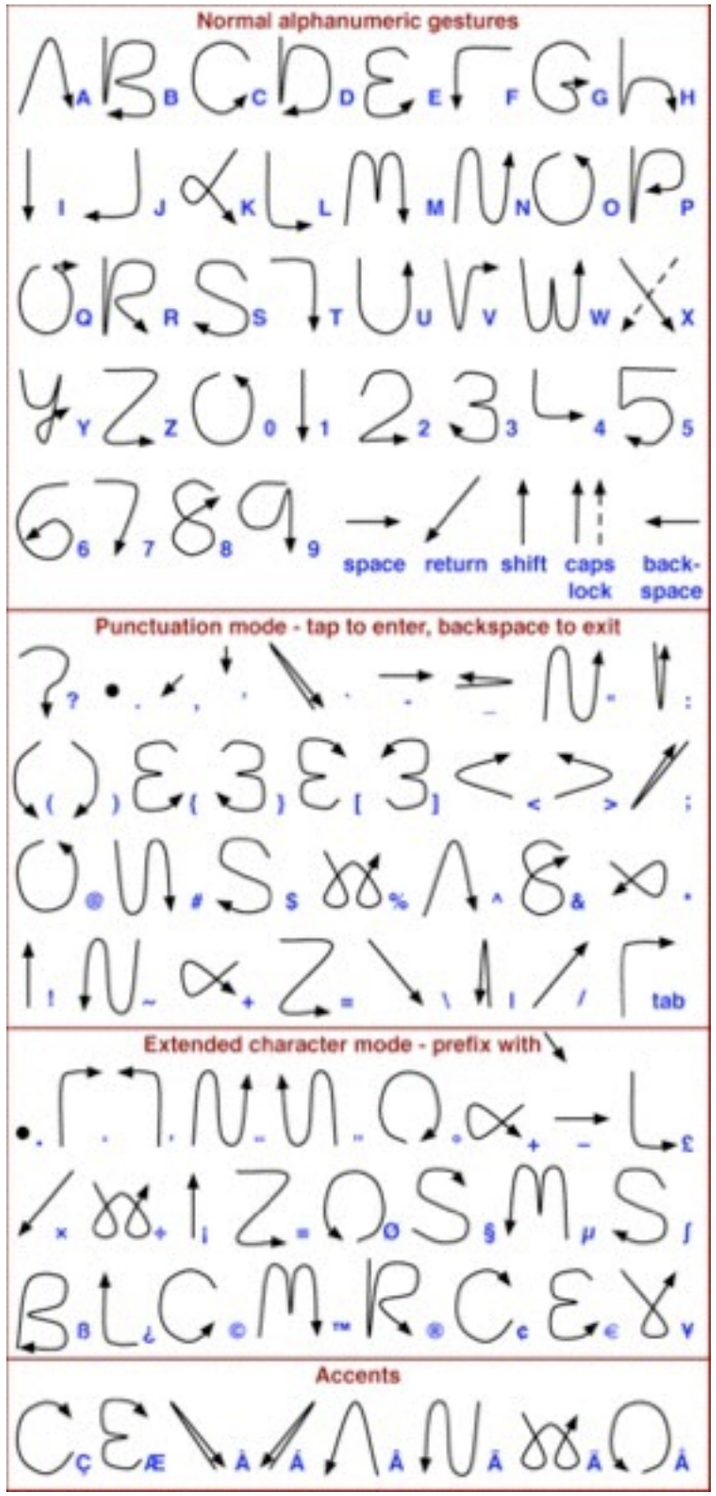
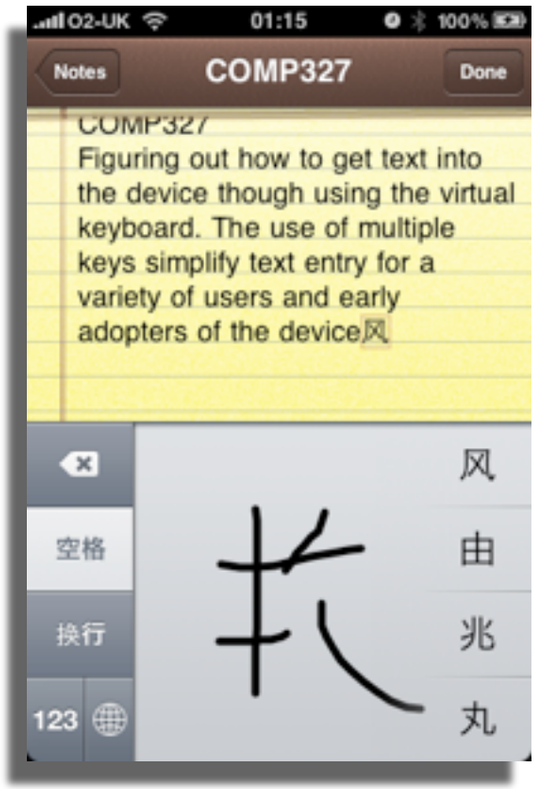


Image taken from Wikipedia, at http://en.wikipedia.org/wiki/File:Palm_Graffiti_gestures.png

GNU Free Documentation License, V1.2

Character Recognition

- Recognises handwriting based on pen-tip movements and pen-up / pen-down signalling
  - Also known as digital ink
  - Never really been successful!
    - Requires high processing power
    - Error rate is often high
- Adoption varies for different alphabets
  - Chinese Traditional & Simplified, Japanese and Korean
  - Recent Windows OSs include personalisation features for Latin alphabets
    - Require training based on samples of the user's handwriting
- Apple's Newton was the first PDA to use Character recognition
  - Was never a commercial success!
  - An evolution of this software exists within Mac OSX
    - Inkwell - appears whenever a tablet input device is used



Electronic Paper

- Typically used in eBooks
  - e.g. iRex iLiad, Sony Reader, Amazon Kindle, etc
  - Used within Motorola's FONE F3, to reduce display thickness, reduce power and improve outdoor use (a backlight can be used in dark environments)
  - Samsung's Alias 2 uses eInk to change the alphabet on the keyboard
- Reflective technology, requiring no emissive light source
  - Bistable, thus power is only required when changing the display
  - Works very well in bright light (compared to LCD displays), and at most viewing angles)
- Rely on millions of tiny microcapsules containing positively charged white particles, and negatively charged black particles
  - By using an electrical charge, the particles are either attracted to the front or back of the microcapsule.
- Very Low Refresh Rates !!!

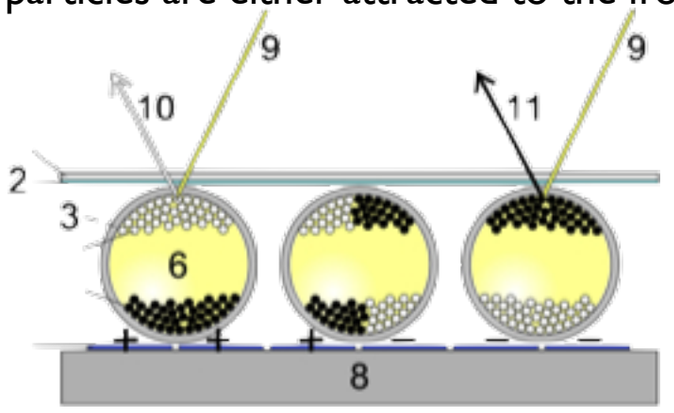


Image taken from Wikipedia, at http://en.wikipedia.org/wiki/File:Electronic_paper_(Side_view_of_Electrophoretic_display).PNG

Exercises...

- Why is HCI so important in getting a device adopted? Although Heckle's law suggests HCI will not prevent adoption if the device (or software's) capability is highly desirable, describe how good HCI improves the user experience
- What are the main challenges in typing in text on a numeric keypad? Describe two methods (a simple one, and a predictive one), and explain how approaches can be adapted to the user.
- Why were physical keyboards originally undesirable, yet why are they still generally preferred over the use of writing? Describe the challenges of character recognition, and how glyphs can be simplified

To Recap...

- In this lecture set, we covered:
  - The general principles of HCI design
    - How do they affect desktop computing
    - Why is mobile computing different
  - What are the general interaction modalities available in mobile phones
    - How do different technologies work
    - How can text be entered in small devices
    - How do display technologies differ?

Further Reading

- *Ubiquitous Computing: Smart Devices, Environments and Interactions*
  Stefan Poslad (Wiley, 2009)
  - Chapter 5