**Interaction Design (II)**

**Pointing Devices**

Direct-control devices:

* Work directly on the surface of the screen.
* Direct “touch” in VR.

Ex.: mobile phones…

* Old: light pen worked back in 1976.
* May produce fatigue:
  + Moving the lighten on the screen required much effort.
  + Should have a surface to rest the arm.

Issues of direct-control devices:

* Imprecision in pointing. Many factors:
  + Quality of the screen: capacitive screens less precise than resistive.
  + Size of the pointer: flat and no-so-fat fingers.
* Land-on strategy:
  + Select on clicking point.
  + Faster feedback.
  + Prone to errors.
* Lift-off strategy:
  + Initial click creates “cursor”, dragging used for precision pointing, lift-off selects.
  + More time consuming.

Advantages:

* Touch screens can be designed with no moving parts:
  + Durable.
  + Only device that has survived Walt Disney’s theme parks.
* Multi-touch allows for complex data entry or manipulation:
  + Pinch-to-zoom gestures.

Other issues:

* Pens may be more suitable for some tasks:
  + Reduce occlusion.
  + Familiar to users.
  + But require to be picked up and put down.
  + Pens are more accurate than fingers.
* Fingers are less precise than wrist-based movement.

Indirect-control devices:

* Work away from the surface.
* Mapping of the user movement to a pointing element (cursor/ray).

Ex.: keyboard, mouse, trackball, joystick, graphics tablets…

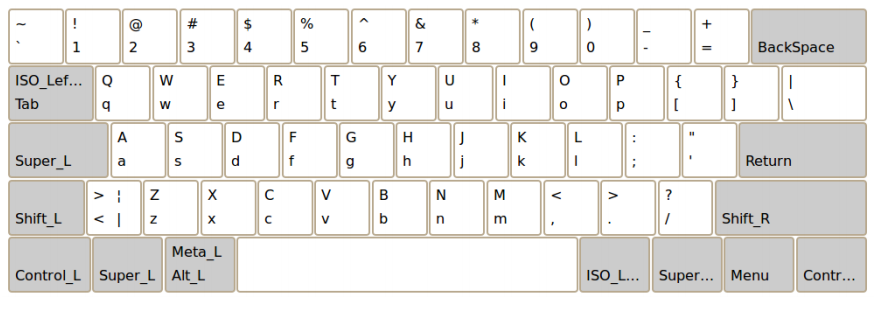
Issues:

* Alleviate hand fatigue.
* Eliminate screen occlusion.
* Mouse is the clear king:
  + Cost-effective.
  + Precise.
  + Hand has a surface to rest on.
  + Buttons easy to press.
  + Long movements require to pick up the mouse and replace (to accelerate movements).

**Typing and Keyboards**

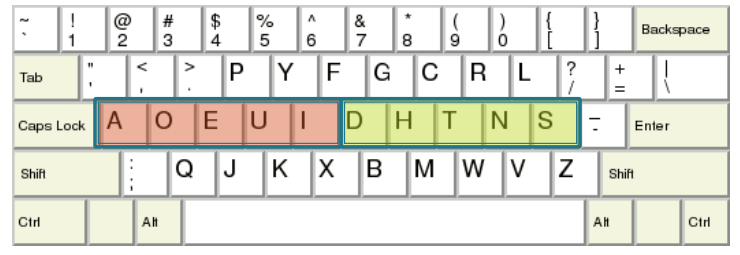
QWERTY keyboard layout:

* The placement of the keys reduces key jams.
* Keys commonly typed together are placed at large physical distance:
  + In a typing machine.
  + Changing hands.
  + Assuming language is English.
* Does not make sense with computers, because not everybody writes in English.



Dvorak layout:

* Vowels in one hand:
  + Combinations with consonants impose hand change.
* Most common letters at the places the fingers rest on the keyboard.



* Invented with the objective of reducing travel distances (10-finger typing).
* Improvements of up to 30%.
* Less errors.
* Also optimized for English.
* Low level of acceptance.

Keyboard layouts:

* Improves posture and reduces tension.
* No proven advantage.

Keyboard arrangements should be designed so that:

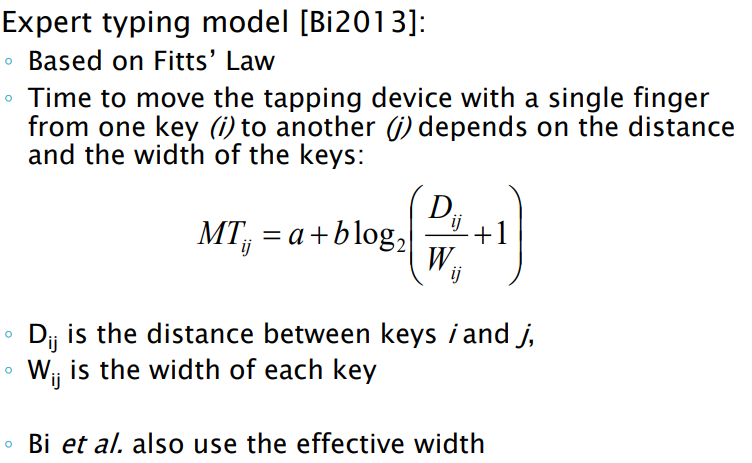
1. Balance the loads on the right and left hands.
2. Maximize the load on the home row.
3. Maximize the frequency of alternating hand sequences.
   * Alternating fingers avoids the need to wait for the end of the movement of the first finger before starting the second movement.
4. Minimizing the frequency of same finger typing.

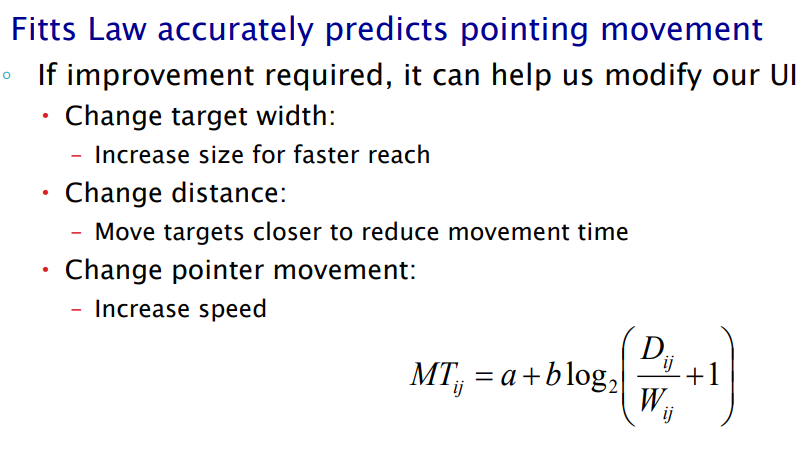


Practical issues:

* Experiment with keyboards layouts is difficult:
  + Users get their proficiency for practice.
  + It requires months of training in any layout.
  + The same people would require to be training back to original arrangement for starting a new experiment.
* It is commonly accepted formal results based as predictive human performance model rather than user testing for evaluation.

Touchable layouts (some issues):

* Size depends on screen size.
* Limited and occluded text.
* Require significant visual attention:
  + No physical feedback, sometimes sound.
* Errors: accidentally touching the screen.
* Touch and stylus based may be a good combined with stroke gestures or other ideas.



Improving mobile layouts:

* Different parameters to take into account:
  + 10-finger typing? As of tablets.
  + 2-thumb typing? Mobiles/tablets.
  + 1-finger typing? Most commonly mobile.
* Optimize for the number of fingers:
  + Tactile screen form factor.
  + Maybe hand positions too.

Mobile layouts:

*Proposed mobile layouts, Minnum:*

* Two or one finger typing.
* Compressing the three key rows into one:
  + Reduction of distance (in vertical).
  + Larger targets.
  + Proficient work prediction/correction is required.
* More room in your screen.

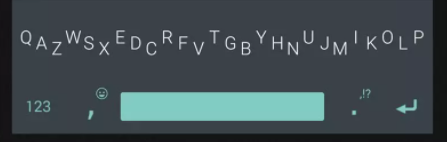
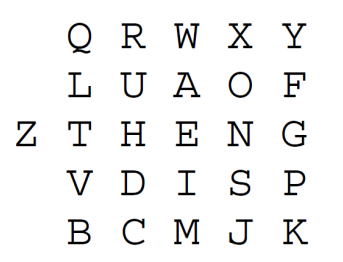


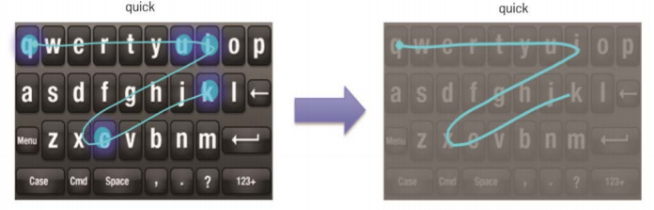
Diagram-based layout for single-finger typing:

* Optimized distances.
* Up to 25 wpm (over the typical 29 wpm on a complete QWERTY)



Single finger gesture typing:

* The finger traverses all the letters of a word without lifting off the screen.
* More comfortable (subjective evaluation) in tablets.
* Not faster than regular typing (objective evaluation) in tablets. Not so negative.



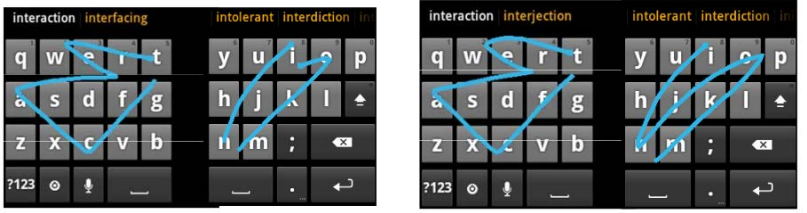
*Proposed mobile layouts, KALQ:*

* Optimize layout for better 2 thumb typing.
* Analyzed hand position, diagram frequency, tablet orientation…



Two fingers gesture typing:

* The two thumbs swipe to compose a word:
  + Lifting the finger when a part of the word belongs to the other thumb.
  + Or with a continuous trace.
* Finger traveling shortened by 50%.
* Speed does not increase over one finger entry (objective evaluation). Not so negative.
* High demand of attention (subjective evaluation).



Designing virtual keyboards. Elements to consider for usability:

*1. Auto-correction:*

Only suitable if proper dictionaries:

* Commonly, users do not notice the corrections.
* Some data such as address very prone to wrong correction.
* 92% sites do it wrong.

Best practices:

* Skip auto-correction for certain fields.
* Usually, it is sager to opt for a predictive approach and let the user to choose the best option.

*2. Auto-capitalization:*

In email addresses, disable auto-capitalization:

* Even if correct, people try to fix.

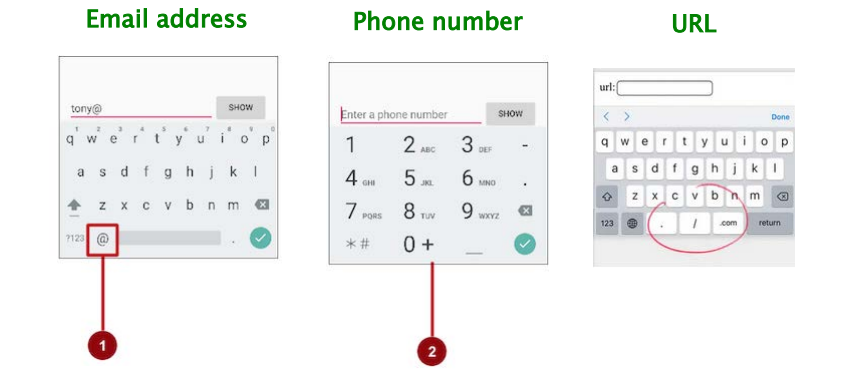
*3. Input data & custom keyboards:*

Virtual keyboards are small:

* Minimum recommended clickable size is 6.85 x 6.85 mm.
* Increase typos, validation errors…
* 60% top mobile websites do it wrong.

Dedicated keyboards may increase the size enough (phone numbers, ZIP codes, currency…):

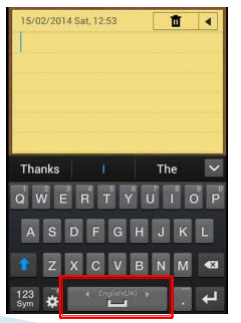
* Invoke them, and do it consistently.



*4. (Multiple) Language support:*

Most custom keyboards provide the possibility of changing the language on demand:

* In many cases correctors or word predictions mix languages.



**Mobile Interaction Design**

Mobile devices have different requirements for design, because:

* More personal.
* The environment where users use them competes for their attention.
* Entering data is difficult.
* Small screen sizes.

Desired features for mobile UIs:

* Quick find what they intend to.
* Minimum cognitive load for interaction.
* Information presented in small chunks.

User Interface and Interaction Design different from desktop.

Design guidelines:

*Keep navigation simple: communicating the current section of the app*

* Ensure navigation feels familiar.
* Design good information architecture.
* Navigation should not grab user attention.
* Ensure users know their location.
* Strive for consistency.
* Clear path to objectives.
* Clear visual hierarchy.

*Finger-friendly tap targets*

* Around 10 x 10mm minimum.
* Keep good spacing between elements.
* Predicted usability of a button according to its size (if it very used, make it bigger and if it is not very used make it smaller).
* For mobile take into account the thumb zones:
  + Consider Fitts only within the operation range of the thumb.
  + Outside elements require extra effort.



*Progressive disclosure and cognitive load reduction*

* Cognitive load: amount of brain power required to use the app\_
  + Keep amount of information (required to remember) low.
  + Progressively show new features or tasks.
  + Helps simplifying UI.

*Make text legible*

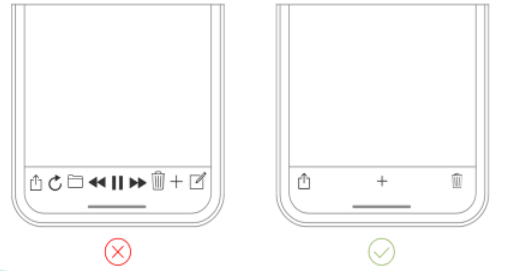
* Choose typeface that works well in multiple sizes and weights.
* Use legible font sizes: at least 11 points.
* Use adequate contrast.
* Correct vocabulary.

*Provide feedback on interactions*

* Use micro interactions if possible.
* Add progress indicators when required.

*Reduce clutter*

* Keep content to a minimum.
* Keep interface elements to a minimum.
* Alternatively, use progressive discovery.
* Strive for minimalism.



*Reduce user inputs*

* Simplifying procedures: onboarding, logon…
* Onboarding:
  + Break in multiple steps.
  + Delay information retrieval.
  + Inform properly on the needs.
* Logon:
  + Use one-time passwords or QRs when possible.

Recommendations:

* Keep forms as short as possible.
* Provide input masks.
* Use smart features such as autocomplete.
* Dynamically validate field values.
* Customize the keyboard for the type of entry.
* When possible, substitute text entry for options.

*Manage friction*

* Some alternative to increase the size that improve usability: visual stimulus, undo, ….
* Some “editing” actions must be dealt with care (send, upload, download, burn, share):
  + Possibility of undoing (even temporarily): for example, Google’s mail.
  + Highlight relevant elements: for example, call to action buttons (they guide users towards your goal conversation).
* Design with friction to avoid mistakes. Rule of the thumb:
  + Make destructive/delicate tasks more difficult-
  + Increasing the effort to prevent accidents:
    - Buttons for non-destructive.
    - Slides for destructive.

*Don’t make users wait for contents*

* Mobile connections are not stable: don’t present blank pages to the user.
* Use skeletons, lower resolution images…
* Update as soon as possible.

*Use gestures prudently*

* Gestures can save space: they do not require visual representations:
  + Hard to remember and use.
  + Not currently standardized.
  + Make use of standard gestures.
  + Don’t use standards gestures for non-standard tasks.

*Continuous integrated experience*

* When possible, synchronize app with desktop interaction (tasks can be continued on different devices).
* Do no replicate exact (web) experience on mobile.
* Be consistent with users’ expectations: in terms of visual elements, interactions…
* Don’t open external web browsers to complete tasks.
* Don’t create dead end pages.