Text Input

Interaction Techniques and Technologies (ITT), SS 2016

Session 6 (19.05.2015), Raphael Wimmer

Overview

These are slides/notes for the lecture, automatically generated from the slide set. Please extend this outline with your own notes.

Goals for this Week

Know

- text input methods
- typical text input speeds
- keyboard hardware

Learn

- measuring typing speed
- PyQT's signals and slots

Practice

- Python
- text input
- study design and conduction

Today

- **14:15 - 14:35** Review of last session, overview of today's session
- **14:35 - 15:20** Text input: metrics, numbers, methods, keyboard layouts
- **15:20 - 15:45** Discussion of upcoming assignment

Where are We?

- **Conducting and Logging Experiments** (+ intro to Python / PyQT)
- **Documenting and Visualizing Experiments** (+ intro to pylab, matplotlib)
- **Pointing** (pointing devices, Fitts' Law, Steering Law, CD gain, …)
- **Text Entry** (speed, models, keyboard layouts, input techniques)
- **Models of Interaction** (KLM, GOMS)

Quiz: Which of the following statements is true?

- Fitts' Law says that the time to select a target increases linearly with distance
- Eye movements can be modeled using Fitts' Law
- A high CD gain is important for pointing on large displays
- Touch screens are rate-control, direct, absolute pointing devices
- A *t test* indicates whether two values are statistically different

Retrospective: Reaction Times

The Model Human Processor

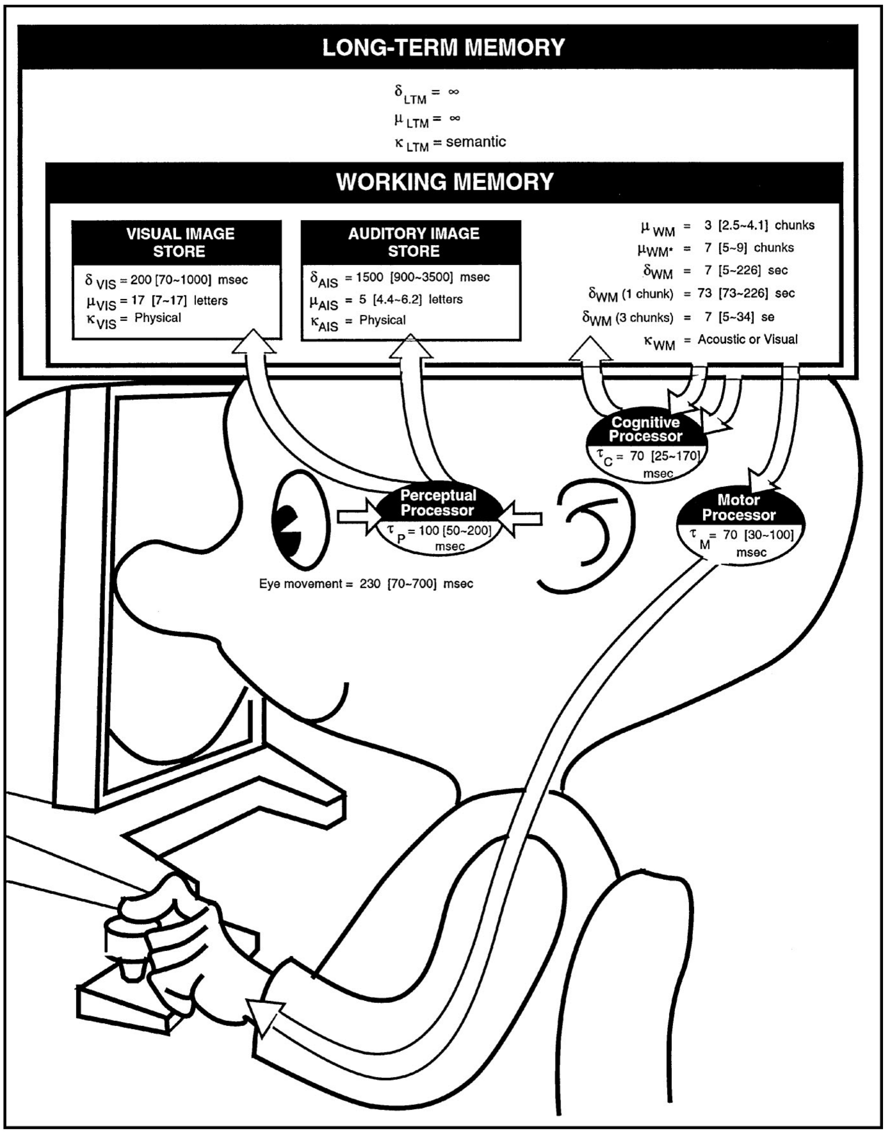


Figure 1: Card et al, 1983

Reaction and Processing Times

Table 1: Typical latencies in human sensorimotor control (Bailey, 1996, p.41)

Operation	Typical Time (ms)
Sensory reception	1 - 38
Neural transmission to brain	2 - 100
Cognitive processing	70 - 300
Neural transmission to muscle	10 - 20
Muscle latency and activation	30 - 70
Total	113 - 528

- approximate average reaction times (source[1]):
  - visual: 270 ms
  - auditory: 150 ms
  - touch: 155 ms
  - proprioception: similar to touch?
- minimal visually perceptible latency: 5 ms (Ng. et al, 2012[2])

Text Input / Text Entry

Overview

- Speech Input
- Handwriting
- Keyboards

Handwriting

- **OCR**
- **natural handwriting** (hard)
- **simplified alphabets**

. . .

. . .

Typing speed

- http://typing-speed-test.aoeu.eu/?lang=en

Hardware

Keyboard implementations

see blackboard

[1] http://biae.clemson.edu/bpc/bp/lab/110/reaction.htm
[2] http://dl.acm.org/citation.cfm?id=2380174

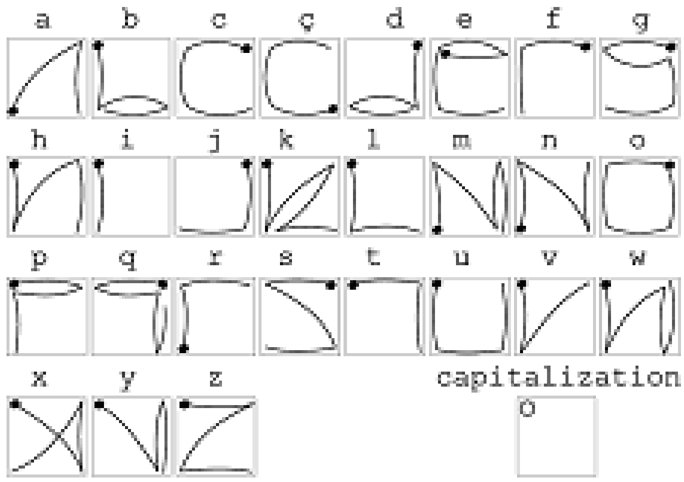


Figure 2: EdgeWrite

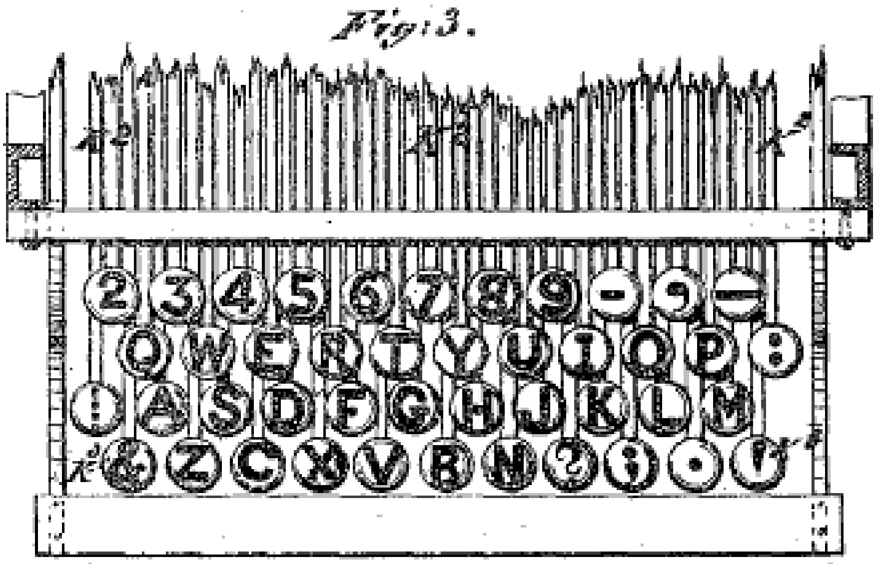
Figure 3: Graffiti

Ghosting / N-key rollover

- simple matrix scanning of contact mats leads to ignored keypresses
- Info in Geekhack.com wiki[3]
- Ghosting Demo by Microsoft Research[4]

Keyboard Layouts

QWERTY



o

- ~1870
- staggered rows required for key levers
- de-facto standard

Dvorak

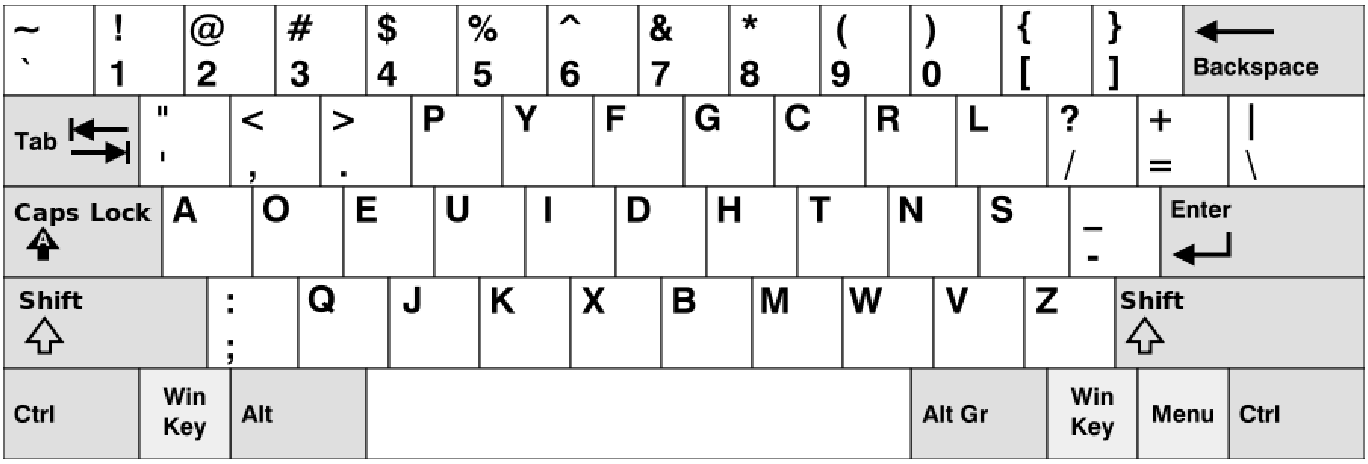


Figure 4: Wikimedia Commons, PD

[3] http://geekhack.org/showwiki.php?title=NKey+Rollover+-+Overview+Testing+Methodology+and+Results
[4] https://www.microsoft.com/appliedsciences/content/projects/KeyboardGhostingDemo.aspx

- ~ 1936
- optimize key locations to minimize finger movement
- shown to be faster than QWERTY (disputed![5])

Neo

Figure 5: neo-layout.org

- since 2004
- http://www.neo-layout.org/
- optimized for German language
- 6 layers, with many Unicode symbols, foreign characters

Colemak

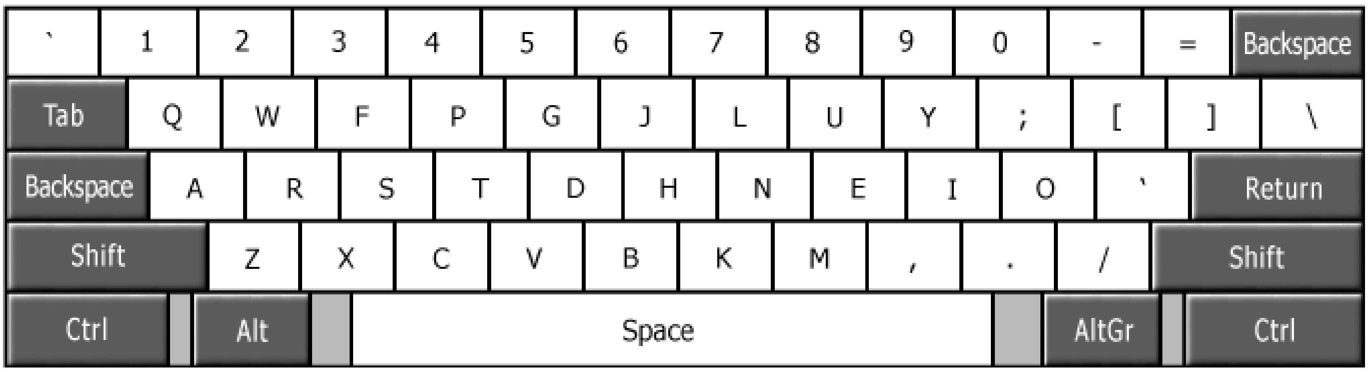


Figure 6: colemak.com

- since 2006
- http://colemak.com
- keys used for common shortcuts (Ctrl+Z/X/C/V) same as in QWERTY
- designed for English language

QFMLWY & Co.

- since 2005
- CarpalX project[6]
- automatically optimized layouts based on different corpora

[5] http://www.utdallas.edu/~liebowit/keys1.html
[6] http://mkweb.bcgsc.ca/carpalx/

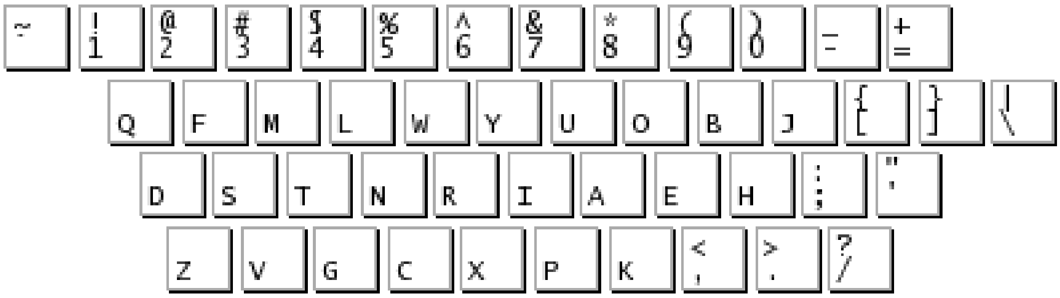


Figure 7: carpalx project

Stenotype

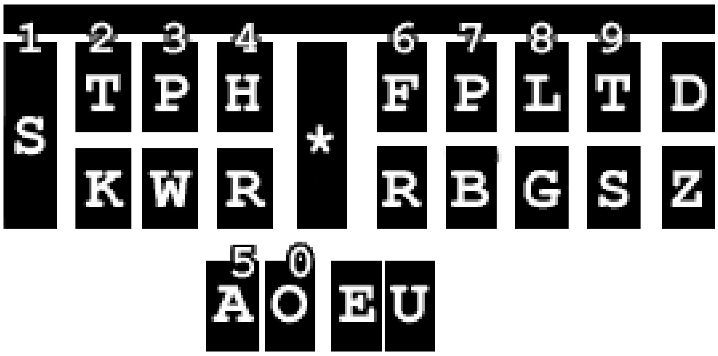


Figure 8: Wikimedia Commons, PD

- Open Steno Project[7]

Typical Text Entry Speeds

Source: WP:Words per minute[8]

- 1 word per minute (WPM) = 5 characters per minute (CPM)
- Handwriting: 30 wpm
- Stenography: 350 wpm
- Speaking: avg. 150 wpm, pro: 350-500 wpm, record: 637 wpm
- Reading: 250-300 wpm (typical adult)
- Morse: good: 20 wpm, pro: 60 wpm, record: 76 wpm
- One-key-keyboard (MacKenzie, 2010)[9]: 5 wpm

QWERTY

- Hunt-and-peck: 30-40 wpm
- Professional Typist: 50-80 wpm
- World Record: 216 wpm (1946, using the Dvorak layout)

[7] http://openstenoproject.org/
[8] https://en.wikipedia.org/wiki%20/Words_per_minute
[9] http://www.yorku.ca/mack/TOCHI2010.html

Stenotype

- Beginner: 100 wpm
- Professional Stenotypist: 200 wpm
- World Record: 360 wpm

Optimized Keyboard Concepts

Chording Keyboards

Figure 9: Buxton, 2010

http://research.microsoft.com/en-us/um/people/bibuxton/buxtoncollection/detail.aspx?id=7

See also: http://www.loper-os.org/?p=861

PianoText

https://www.youtube.com/watch?v=-ykkTXo2Zyg

http://pianotext.mpi-inf.mpg.de/

Tera-Keyboard (Ghost in the Shell)

https://youtu.be/YZX58fDhebc?t=14

Discussion on the SciFi Interfaces blog[10]

Mobile Phone Keyboards

- Gizmodo: 12 smartphone keyboards that are trying to reinvent mobile text input[11]

Learning Trade-off

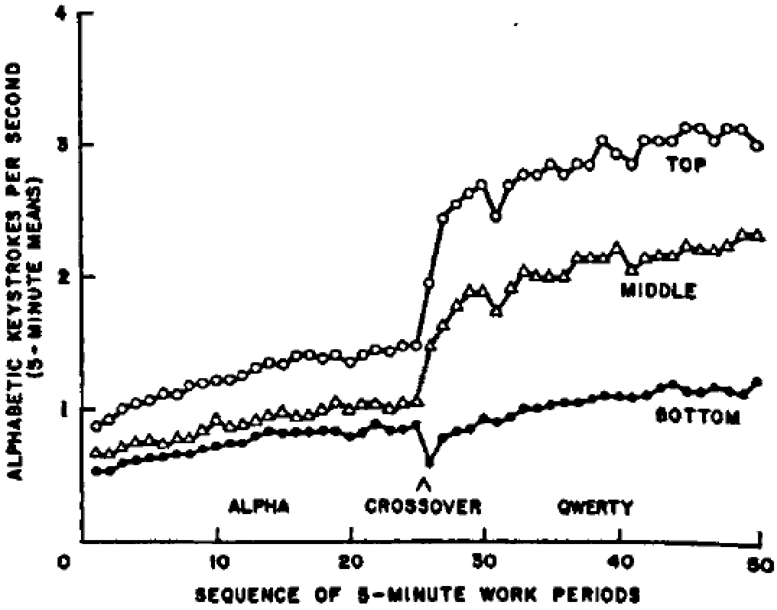


Figure 5. Keying rates for alphabetic characters on both keyboards for all A-Q subjects in each third of the rank order.

Figure 10: Michaels, 1971

S. Eugene Michaels (1971). Qwerty Versus Alphabetic Keyboards as a Function of Typing Skill. In *Human Factors: The Journal of the Human Factors and Ergonomics Society* 1971 13:DOI: 10.1177/001872087101300504

Slow Improvement

Predictive Techniques

- Do not require the user to learn something new - meet them where they are.
- **Autocomplete**:
  - provide suggestions for word completions once one or more letters are typed
  - Autocomplete in Python: github.com/rodricios/autocomplete[12]
  - See also QCompleter[13]
- **Autocorrect**:

[10] https://scifiinterfaces.wordpress.com/2013/07/24/the-secret-of-the-tera-keyboard/
[11] http://gizmodo.com/12-smartphone-keyboards-that-are-trying-to-reinvent-mob-1695151919
[12] https://github.com/rodricios/autocomplete
[13] http://doc.qt.io/qt-5/qcompleter.html

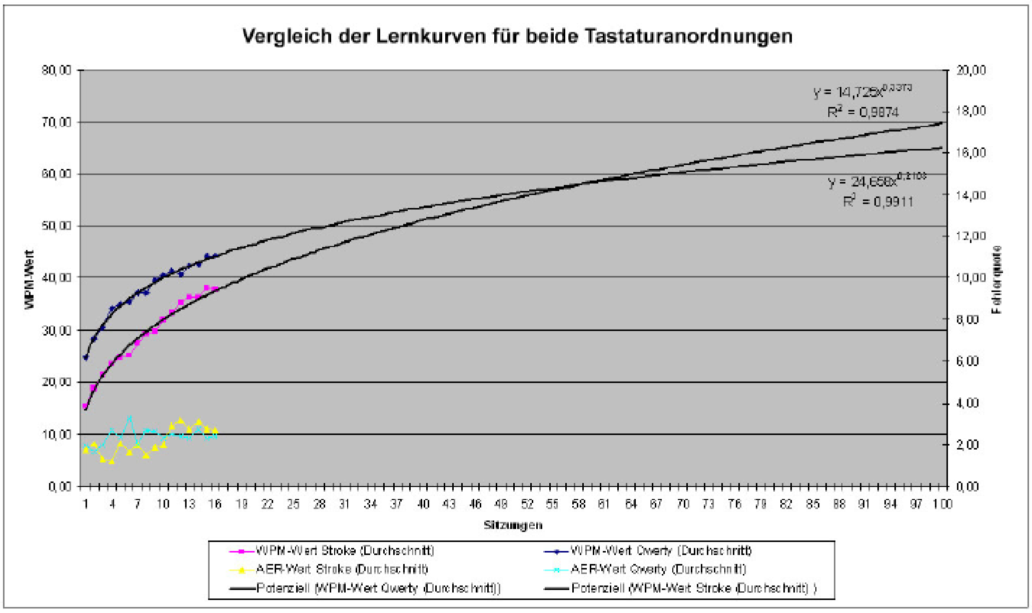


Figure 11: Johannes Jüngst, Diplomarbeit, 2010

- automatically change an incorrectly spelled word to the probably intended word
- also used for replacing abbreviations
- Python example: github.com/foobarmus/autocorrect[14]

Outlook

Course Assignment

Let's add chord input to a QWERTY keyboard.

Questions:

- Implementation?
- Choice of chords?
- Evaluation?

Next Session

more on Python:

- decorators
- QT widgets
- QT signals / slots

Further Reading

- http://careyryan.com/stenotype-can-we-type-much-faster/
- http://www.yorku.ca/mack/hci3.html

[14] https://github.com/foobarmus/autocorrect/

ENDE