UNIVERSITÀ DEGLI STUDI DI NAPOLI FEDERICO II

SOFTWARE ENGINEERING – LECTURE 13

# Models and Theories in Human-Computer Interaction

Prof. Luigi Libero Lucio Starace

luigiliberolucio.starace@unina.it

https://luistar.github.io

https://www.docenti.unina.it/luigiliberolucio.starace

# Today, on Software Engineering

**Usability Edition**

**Human Abilities**

(perception, memory, cognitive processes...)

**Technologies**

(VR, AR, mouse, windowing, touch screens, ...)

**Methodologies**

(prototyping, lifecycles, ...)

**Guidelines, Principles and Theories**

**Evaluation**

(experiments, observation, ...)

# What’s a model?

A **model** is a simplification (abstraction) of reality

- A perfect mapping of reality is not a model (and not useful!)
    - Precision vs generality
- «*All models are wrong, but some are useful*»

Models allow us to:

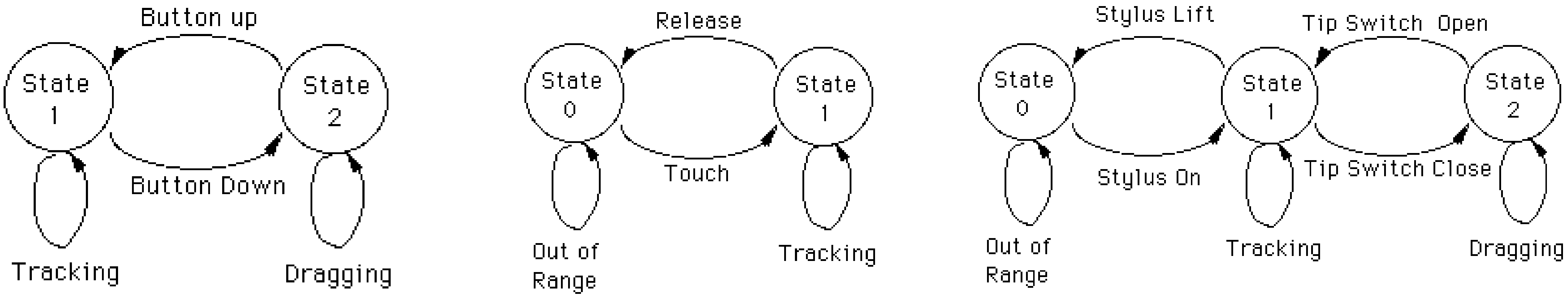
- **Represent** and **reason** about (aspects of) phenomena of interest
- Anticipate (**predict**) outcomes

# HCI models and theories

- In HCI, models and theories aim at explaing how humans interact with computers
- HCI models and theories can be classified as:
  - **Descriptive**: aim at developing a consistent terminology and useful taxonomies
  - **Explanatory**: describe sequence of events, possibly with causal relationship
    - E.g.: Norman’s seven stages of action
  - **Predictive**: Aim at enabling the comparison of design alternatives based on numeric predictions of speed or errors
  - **Prescriptive**: Offer guidelines for designers to make decisions

# The three-state model of graphical input

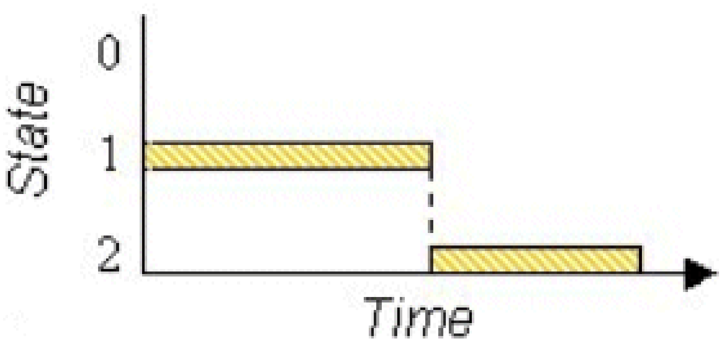
- Proposed by Will Buxton in 1990 [1]
- Describes graphical input with pointing devices
- Different technologies (mouse, trackpads, tablets with stylus, etc...)

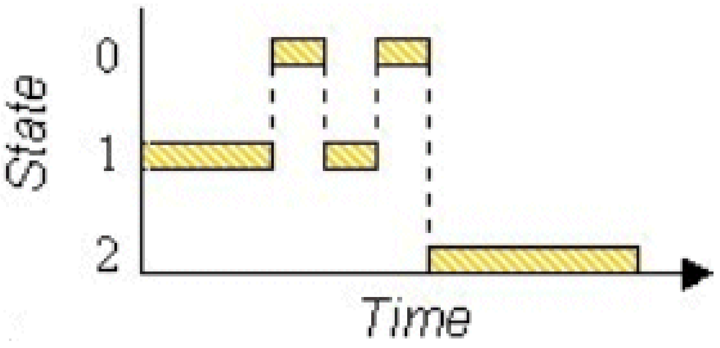



[1] Buxton, W. (1990). A Three-State Model of Graphical Input. In INTERACT '90.

# The three-state model of graphical input

- Dragging tasks with a mouse (left) and with a lift-and-tap touchpad (right) (after [1])



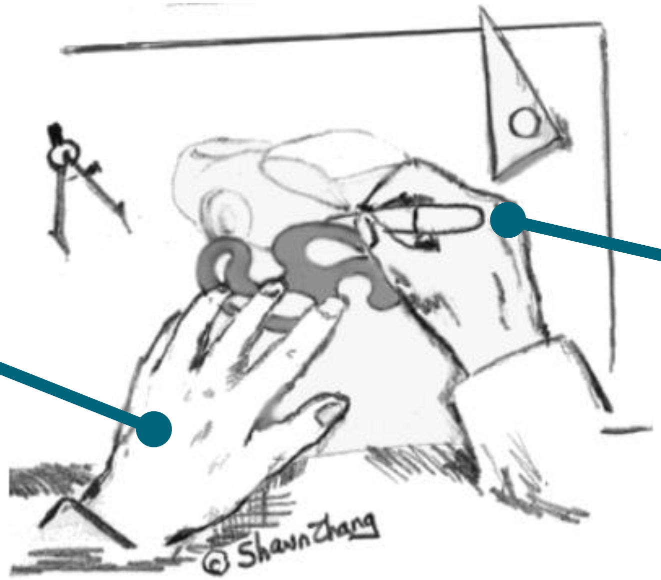




[1] MacKenzie, I. S., & Oniszczak, A. (1997). The tactile touchpad. *CHI '97 Conference on Human Factors in Computing Systems*

# Guiard's model of bimanual skill

- Many interactions are **asymmetric** w.r.t. left/right hand
- Guiard's model describes roles and actions of preferred/non-preferred hands

**Non-preferred**
- Leads the preferred hand
- Sets the spatial frame of reference for the preferred hand
- Performs coarse movements

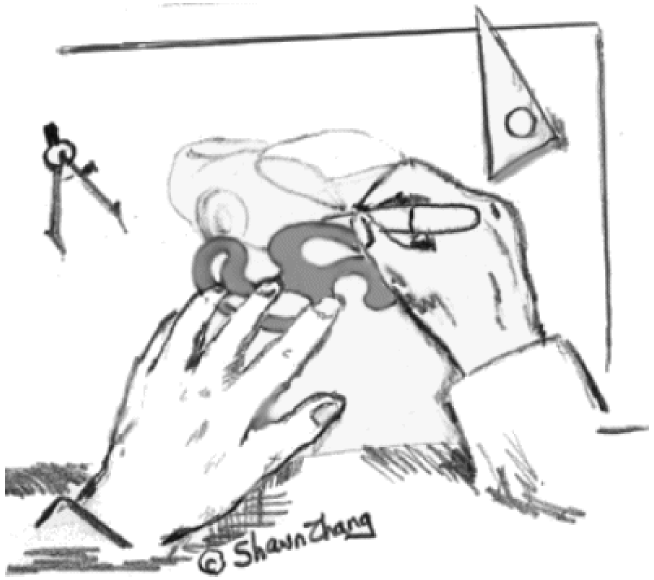



Two-handed interaction paradigm.
(Sketch by Shawn Zhang)

**Preferred**
- Follows the non-preferred hand
- Works within established frame of reference set by the non-preferred hand
- Performs fine movements

# Guiard’s model of bimanual skill

- The artist acquires the template with her left hand (*non-preferred hand leads*).
- The template is manipulated over the workspace (*coarse movement, sets the frame of reference*).
- The stylus is acquired in the right hand (*preferred hand follows*) and brought into the vicinity of the template (*works within frame of reference set by the non- preferred hand*).
- Sketching takes place (*preferred hand makes precise movements*).



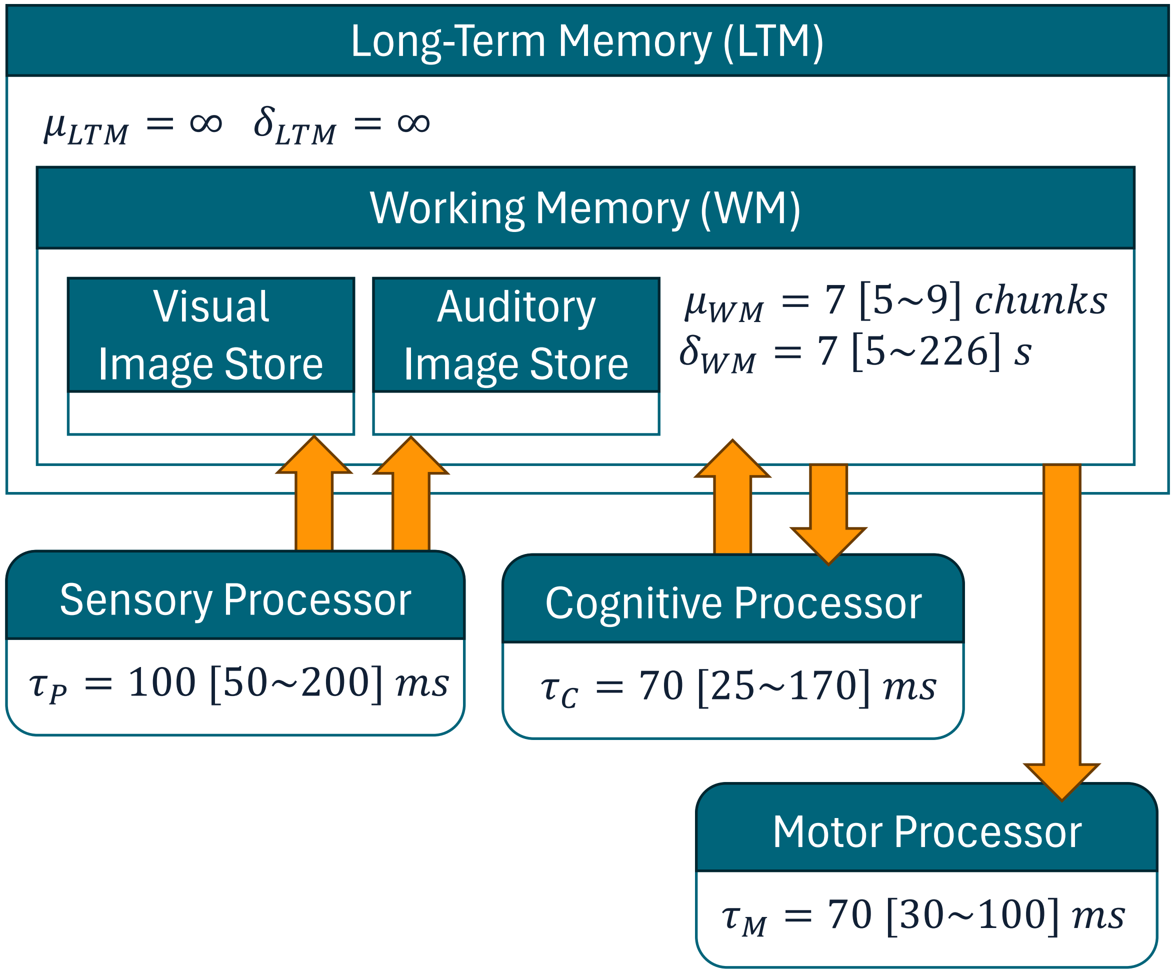

Two-handed interaction.
(Sketch by Shawn Zhang)

# The Model Human Processor (MHP)

- Developed by Card, Moran and Newell in 1983
- It’s a **predictive**, a-priori model
  - Can give approximations of user actions before real users are involved in the testing process (and before the UI is even implemented!)
- A human is modelled by a set of memories and processors that function according to a set of principles
- **Discrete** and **sequential** model

# The Model Human Processor

- Perceptual, Cognitive and Motor processors
- Different types of memories
- Model parameters:
  - Processor cycle times: $\tau$
  - Memory decay time: $\delta$
  - Memory capacity: µ



# MPH: Memories

- **Working Memory** (WM) is a subset of «activated» elements (chunks) from the **Long-Term Memory** (LTM)
  - Chunks can be composed of smaller units like letters in a word
  - A chunk might also consist of several words, as in a well-known phrase
- $\mu_{LTM} = \infty$ and $\delta_{LTM} = \infty$
- $\mu_{WM} = 7\ [5\sim9]\ chunks$
- $\delta_{WM} = 7\ [5\sim226]\ s$
  - Decay time for WM largely varies based on the number of stored chunks
  - $\delta_{WM}(1\ chunk) = 73\ [73\sim226]s$
  - $\delta_{WM}(3\ chunks) = 7\ [5\sim34]s$

# Perception of congruent and incongruent stimuli

- Perception times are also influenced by the nature of stimuli
- For example, there is a significant delay in reaction time (**Stroop effect**) between congruent and incongruent stimuli.
- Let’s do an experiment!
    - Volunteer needed
    - There will be three columns of words
    - You have to say the COLOR of each words
    - Say «done» when finished

# MHP: Why?

- You can see an interaction as a «program» for the MHP
- You can then «run» it on the MPH to estimate completion time!
  - Humans differ from each other. MPH allows to tune parameters (e.g.: processor cycle times) to target humans with different abilities

# MHP: Example application

- Suppose we are designing the main menu for a command-line utility with 16 different features
- Which alternative would be better?
  - 1x16 (breadth) or 4x4 (nested) menu?

```
*** MAIN MENU ***

 A. Feature A
 B. Feature B
 C. Feature C
    ...
 O. Feature O
 P. Feature P

Insert your choice>
```

```
*** MAIN MENU ***

 A. Submenu A
 B. Submenu B
 C. Submenu C
 D. Submenu D

Insert your choice>
```

```
*** MAIN MENU ***

 A. Feature A
 B. Feature B
 C. Feature C
 D. Feature D

Insert your choice>
```

```
*** MAIN MENU ***

 E. Feature E
 F. Feature F
 G. Feature G
 H. Feature H

Insert your choice>
```

# MHP Example: Breadth (1x16) Menu

```
foreach item in menu:
  Execute eye movement to item                        τM
  Perceive item text, transfer to WM                  τP
  Retrieve meaning of item, transfer to WM            τC
  Match code from displayed to needed item            τC
  if(Decide on match)                                 τC
    break

Execute eye movement to menu item letter              τM
Perceive menu item letter, transfer to WM             τP
Decide on key                                         τC
Press key in response                                 τM
```

Average number of iterations in a serial search on 16 items

$$(\tau_M + \tau_P + 3\tau_C) \cdot (16+1)/2$$

$$2\tau_M + \tau_P + \tau_C$$

$$T = (\tau_M + \tau_P + 3\tau_C) \cdot (16+1)/2 + 2\tau_M + \tau_P + \tau_C$$

$$= (70 + 100 + 3 \cdot 70) \cdot 8{,}5 + 2 \cdot 70 + 100 + 70 = \mathbf{3540\ ms}$$

# MHP Example: Depth (4x4) Menu

- Same procedure and steps as the depth menu
- But this time we do 2 serial searches over 4 items

$$T = 2 \cdot [(\tau_M + \tau_P + 3\tau_C) \cdot (4+1)/2 + 2\tau_M + \tau_P + \tau_C]$$

$$= 2 \cdot [(70 + 100 + 3 \cdot 70) \cdot 2{,}5 + 2 \cdot 70 + 100 + 70] = \mathbf{2520\ ms}$$

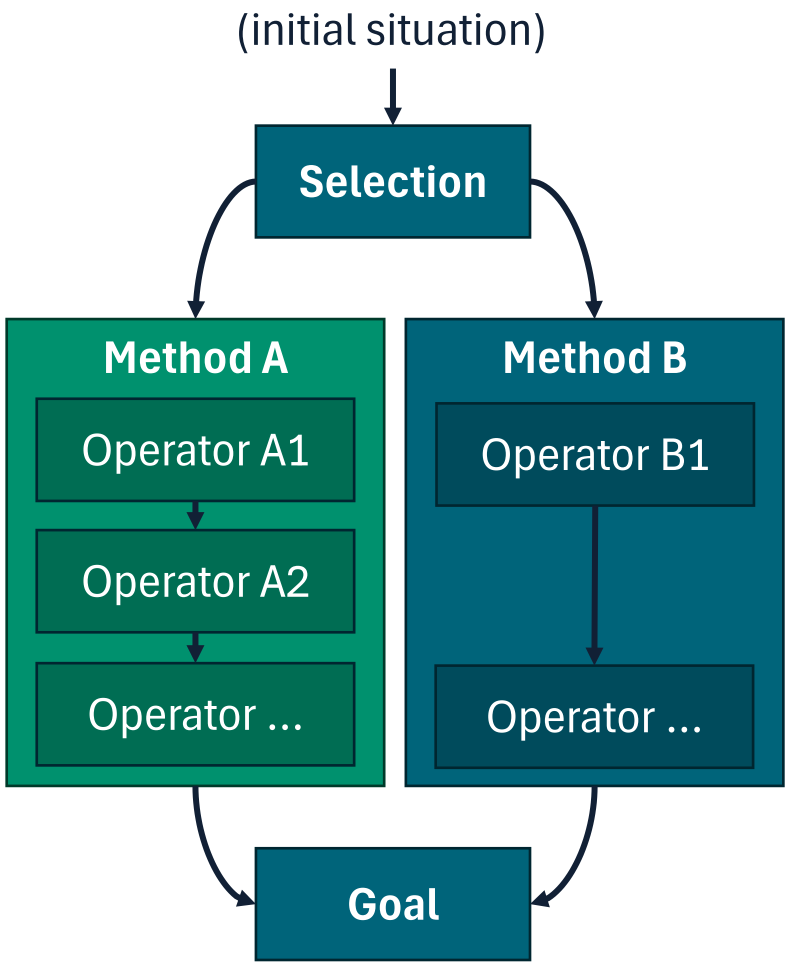
- The 4x4 menu is predicted to be ~30% faster than the 1x16 one!

# GOMS Model

- **G**oals, **O**perators, **M**ethods, **S**election rules
- Assumptions:
  - Interaction with a system is problem solving
  - Decompose interaction into subproblems
  - Determine goals to «attack» problem
  - Specify sequence of operations used to achieve goals
  - Timing values can be assigned to each operation

# GOMS

- **Goals**:
  - What the user wants to achieve (e.g., “start the correct utility function”).
- **Methods**:
  - Possibly alternative sequences of operators used to achieve the goal.
- **Selection rules**:
  - Criteria for choosing among different methods.
- **Operators**:
  - Basic actions performed by the user (e.g., “move mouse”, “click”, “check setting”).



# GOMS: Example

- **Goal**: delete a word in a document editor
- **Selection rule:** if cursor is at end of word to delete, use Method A, else use Method B
- **Method A**:
  - Press «backspace» key.
  - Check if word is deleted and return to previous operation if needed.
- **Method B**:
  - Move mouse cursor on word.
  - Perform double-click.
  - Press «backspace» key.

# Keyboard Level Model (KLM)

- KLM is one of the simplest GOMS variants
- Focuses on observable behaviour
  - Keystrokes, mouse movements, …
- Assumes error-free performance
- Common operators and the typical corresponding times:
  - **K (Keystroke):** 0.2 seconds (200 ms)
  - **P (Pointing with Mouse):** 1.1 seconds (1100 ms)
  - **B (Pressing/holding/releasing mouse button):** 0.1 seconds (100 ms)
  - **H (Homing Hands):** 0.4 seconds (400 ms)
  - **M (Mental Preparation):** 1.2 seconds (1200 ms)
  - **R (System Response):** Variable; typically around 0.1 seconds (100 ms)

# KLM Example: deleting a file

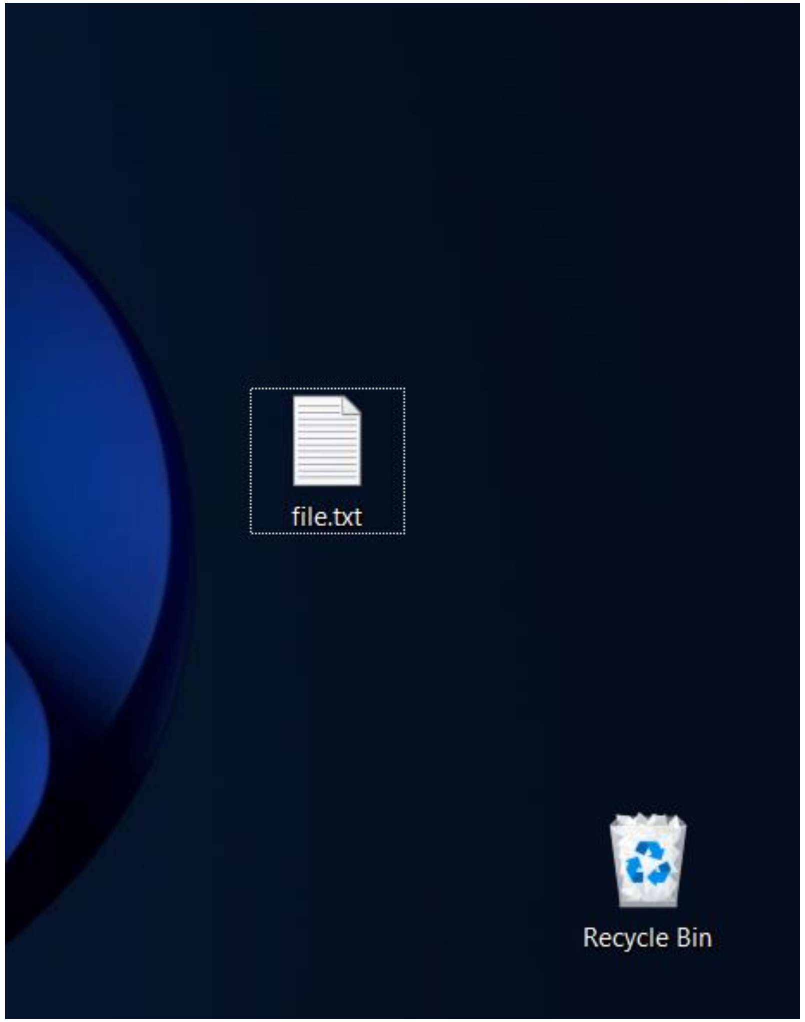
We can delete a file by:

A. Dragging and dropping it to the trash can icon

B. Selecting it and pressing the «delete» key

C. Performing a right click on it, then selecting the «move to trash» option

Which alternative is faster?



# KLM Example: deleting a file

**Alternative A** operators:

- Prepare to perform action (M)
- Point with the mouse over the file icon (P)
- Press and hold mouse button (B)
- Drag file icon to trash can icon (P)
- Release mouse button (B)

**Assumptions:**

- Hand already on mouse
- Trash icon visible
- File icon visible

Total time = M + 2P + 2B = 1.2 s + 2*1.1s + 2*0.1s = **3.6 s**

# KLM Example: deleting a file

**Alternative B** operators:

- Prepare to perform action (M)
- Point with the mouse over the file icon (P)
- Perform click (B)
- Move hand to keyboard (H)
- Press «delete» key (K)

**Assumptions:**

- Hand already on mouse
- Trash icon visible
- File icon visible

Total time = M + P + B + H + K = 1.2 s + 1.1 s + 0.1 s + 0.4 s + 0.2 s = **3 s**

# KLM Example: deleting a file

**Alternative C** operators:

- Prepare to perform action (M)
- Point with the mouse over the file icon (P)
- Perform right click (B)
- Point with mouse over «delete» option in context menu (P)
- Perform click (B)

**Assumptions:**

- Hand already on mouse
- Trash icon visible
- File icon visible

Total time = M + 2*P + 2*B = 1.2 s + 2*1.1s + 2*0.1s = **3.6 s**

# The Power Law of Practice

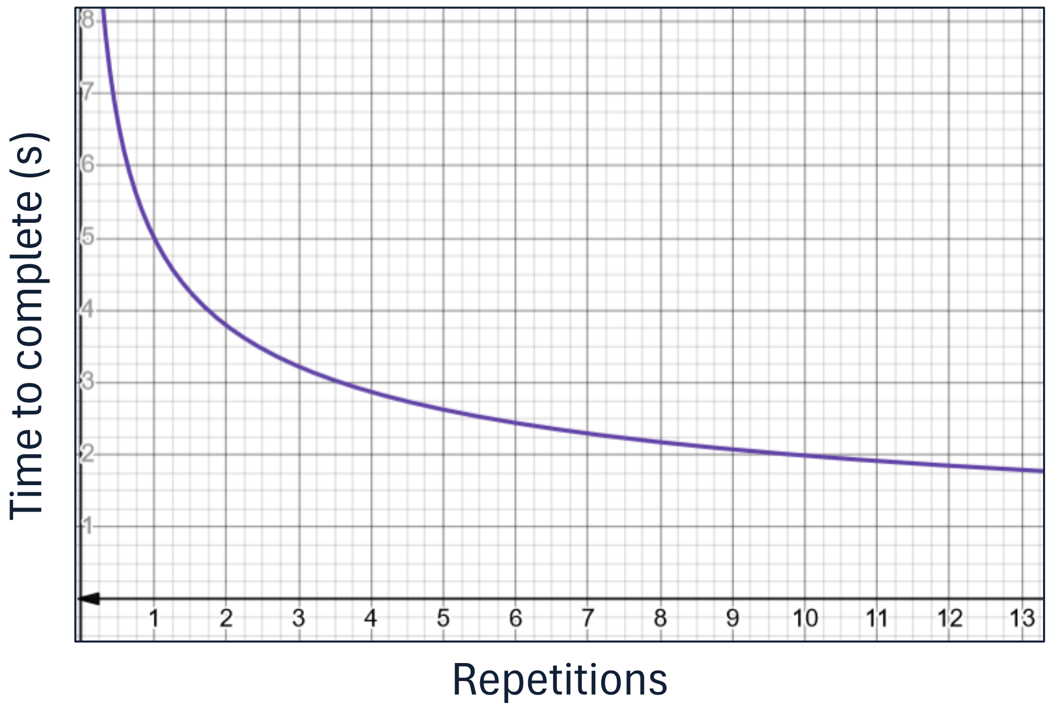
- Allen Newell (cognitive scientist) in the 1980s analyzed reaction times for a variety of tasks in learning experiments
- He noticed that the learning curves obtained in these studies have a very similar shape: that of a **power law**
- The time required to complete a task after $n$ trials ($T_n$) is close to the time required to complete that task the first time ($T_1$) times $n^{-a}$

$$T_n \approx T_1 \cdot n^{-a}$$

- $a$ is a parameter comprised between 0.2 and 0.6 (generally ~0.4)

# The Power Law of Practice, in practice

- A user took **5 seconds** to perform a given task the first time they were exposed to the new UI we developed
- How many repetitions would that user need to be able to perform the task in **2 seconds** or less?
  - We can compute an estimation wity the power law of practice: $T_n \approx T_1 \cdot n^{-a}$
  - We solve for $n$, assuming $a = 0.4$
  - $2\ s \leq 5\ s \cdot n^{-0.4}$
  - For $n = 10$, we get that $T_n \approx 1.99\ s$




Plot of the power law of practice, with the given parameters

# Hick’s Law

**Hick's law** describes the time it takes for a person to make a decision among a set possible choices.

- Hick’s law states that the time $T$ required to reach a decision increases *logarithmically* with the number of choices.
- In the case of equally probable alternatives:

$$T = a + b \cdot \log_2(n + 1)$$

  - $n$ is the number of choices
  - $a$ and $b$ are parameters depending on the context conditions (e.g.: the way choices are presented, the familiarity of the user,…)

# Applying Hick's Law

Which way is faster to select among 64 options?

- One-level 1x64 menu
  - $T = a + b \cdot \log_2(64) = a + 6b$
- Two-level 4x16 menu
  - $T = a + b \cdot \log_2(4) + a + b \cdot \log_2(16) = 2a + 6b$
- Two-level 8x8 menu
  - $T = 2 \cdot (a + b \cdot \log_2(8)) = 2a + 6b$
- Three-level 4x4x4 menu
  - $T = 3 \cdot (a + b \cdot \log_2(4)) = 3a + 6b$
- Six-level 2x2x2x2x2x2 menu
  - $T = 6 \cdot (a + b \cdot \log_2(2)) = 6a + 6b$

# Fitt’s Law

Models time to acquire targets in aimed movement

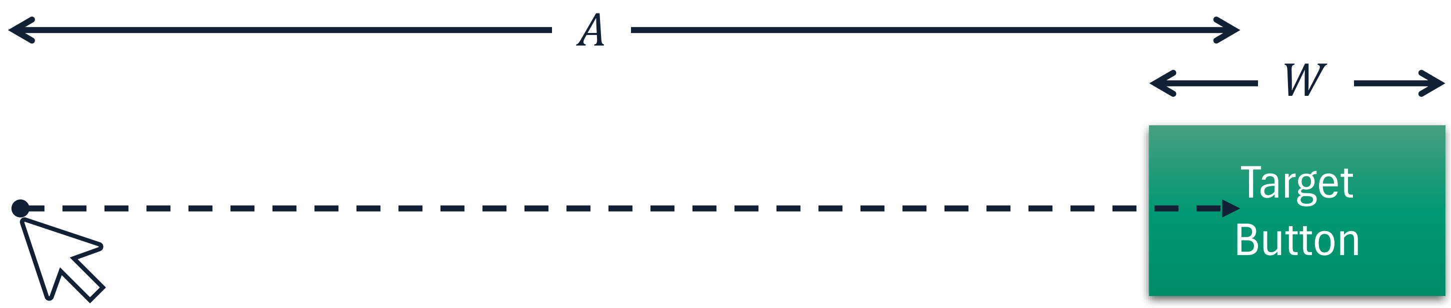
- Reaching for a control in a cockpit
- Moving across a dashboard
- Pulling a defective item from the conveyor belt
- Clicking on icons using a mouse

# Fitt's Law – Index of Difficulty (ID)

- The index of difficulty of a target acquisition task is defined as

$$ID = \log_2\left(\frac{A}{W} + 1\right)$$

- $A$ is the **Amplitude** of the movement (distance from start to target)
- $W$ is the **Width** of the target (permissible variability)



# Fitt's Law – Movement Times

- Movement Times (MT) depend on the Difficulty Index ID

$$MT = a + b \cdot ID = a + b \cdot \log_2\left(\frac{A}{W} + 1\right)$$

- Movement times also depend on the system, pointing device, user…
  - Can be fitted to specific cases with non-negative parameters $a$ and $b$
- It's the equation of a straight line ($y = mx + c$), where $b$ is the gradient
- $MT$ increases **linearly** with the $ID$

# Fitt's Law: Applications

If we need to reduce the time needed to perform a target seeking action

- Either we reduce the Amplitude of the movement (move target closer)
- Or we increase the Width of the target
- Or we could work on $a$ and $b$

1 cm button 7 cm away

1 cm button 3 cm away

3 cm button 3 cm away

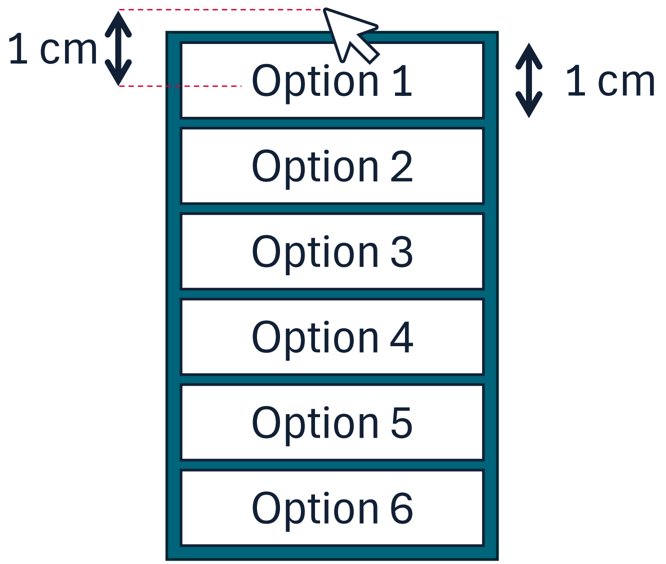
$$ID = \log_2\left(\frac{7}{1} + 1\right) = 3$$

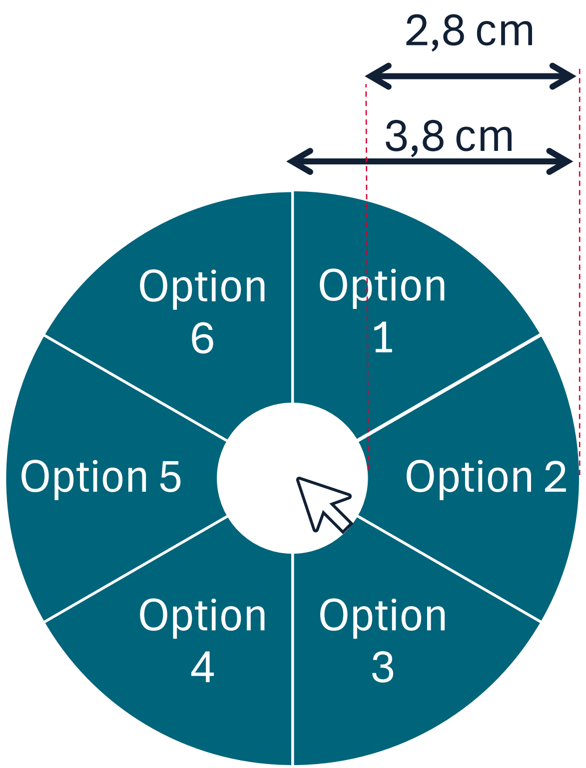
$$ID = \log_2\left(\frac{3}{1} + 1\right) = 2$$

$$ID = \log_2\left(\frac{3}{3} + 1\right) = 1$$

# Fitt's Law: Applications

- Which will be **faster** on average?

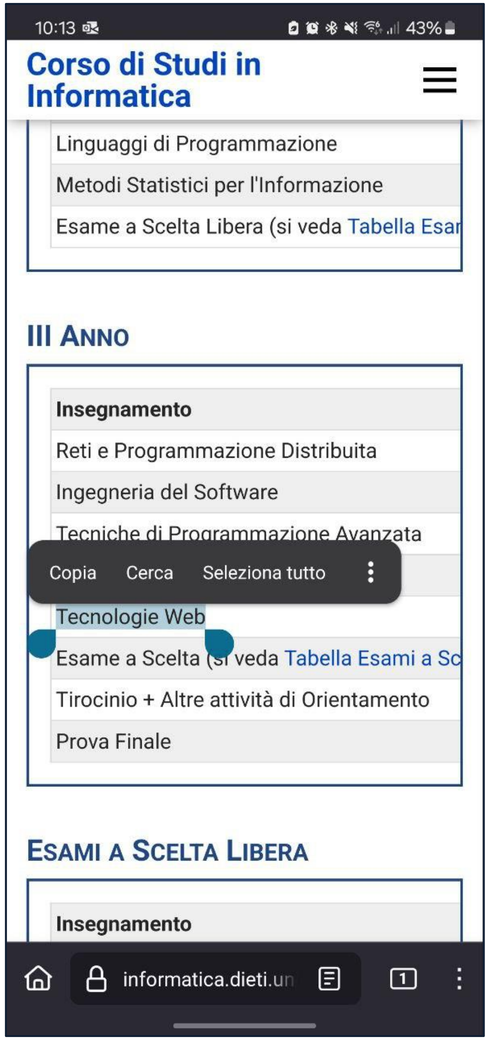


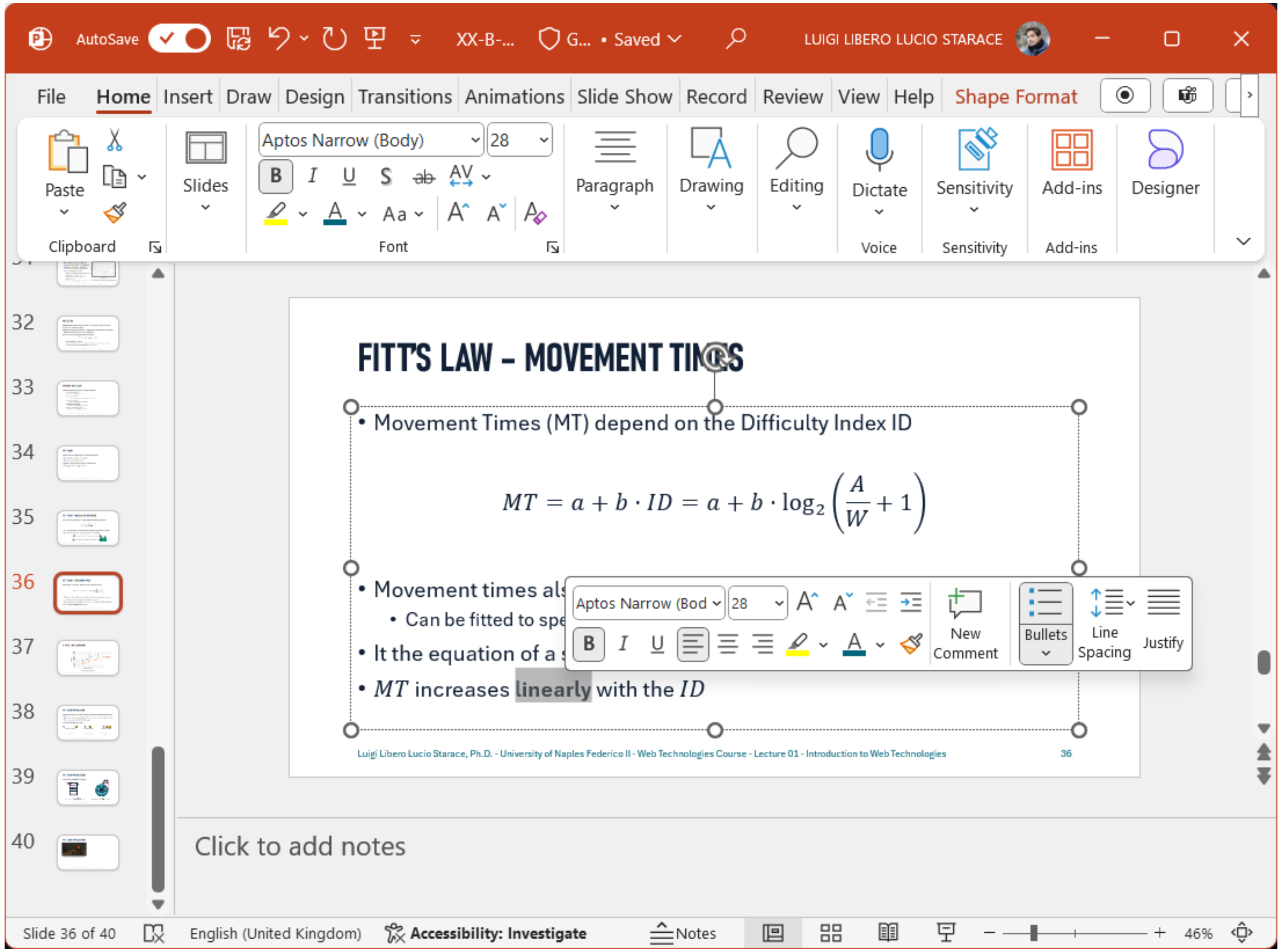




$$Average\ ID = \frac{\sum_{i=1}^{6} \log_2\left(\frac{i}{1}+1\right)}{6} \approx 2{,}04$$

$$Average\ ID = \log_2\left(\frac{2{,}4}{2{,}8}+1\right) \approx 0{,}89$$

# Fitt's Law: Applications



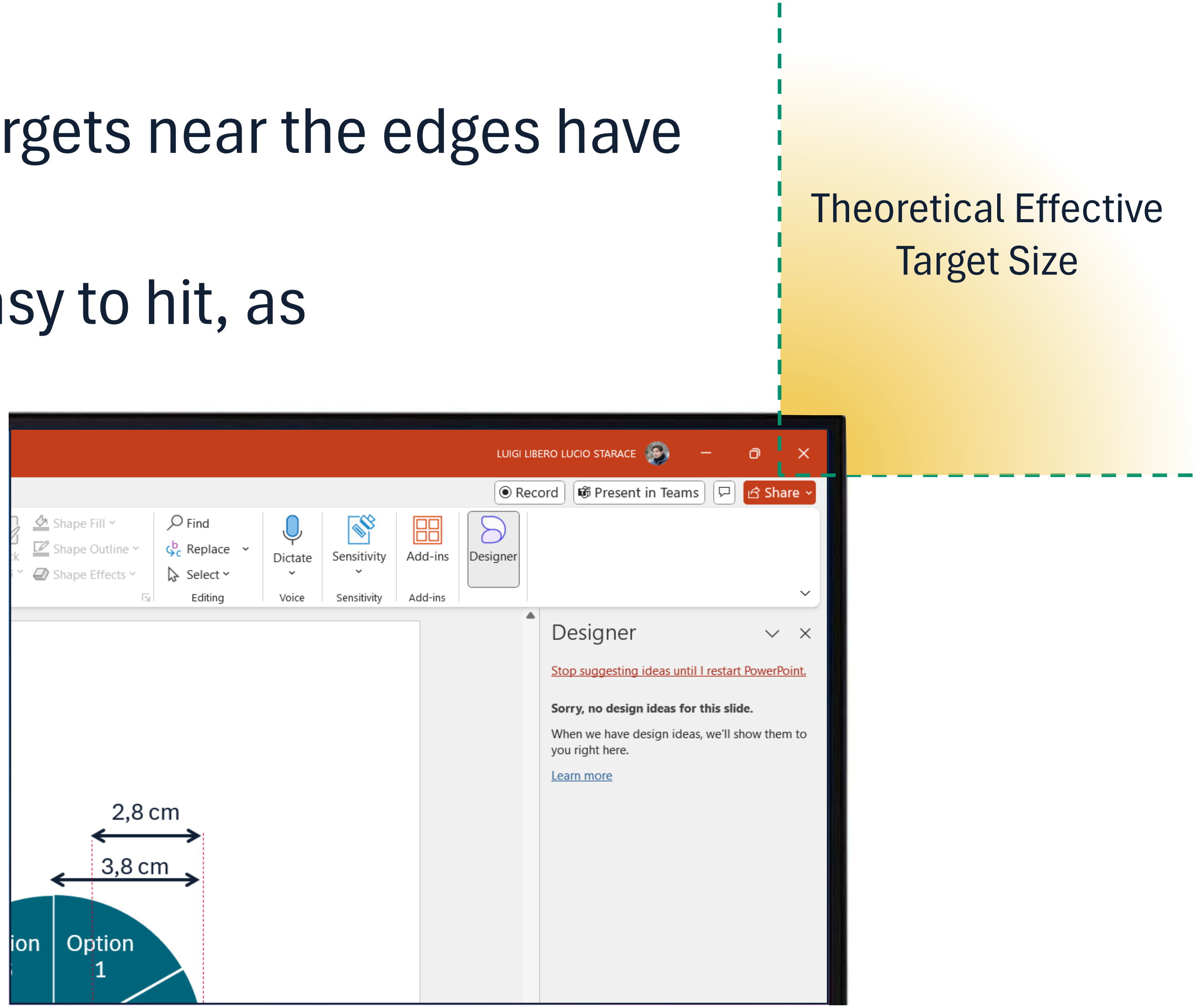
# Fitt’s Law: Applications

Shadow of the Tomb Raider
(videogame)

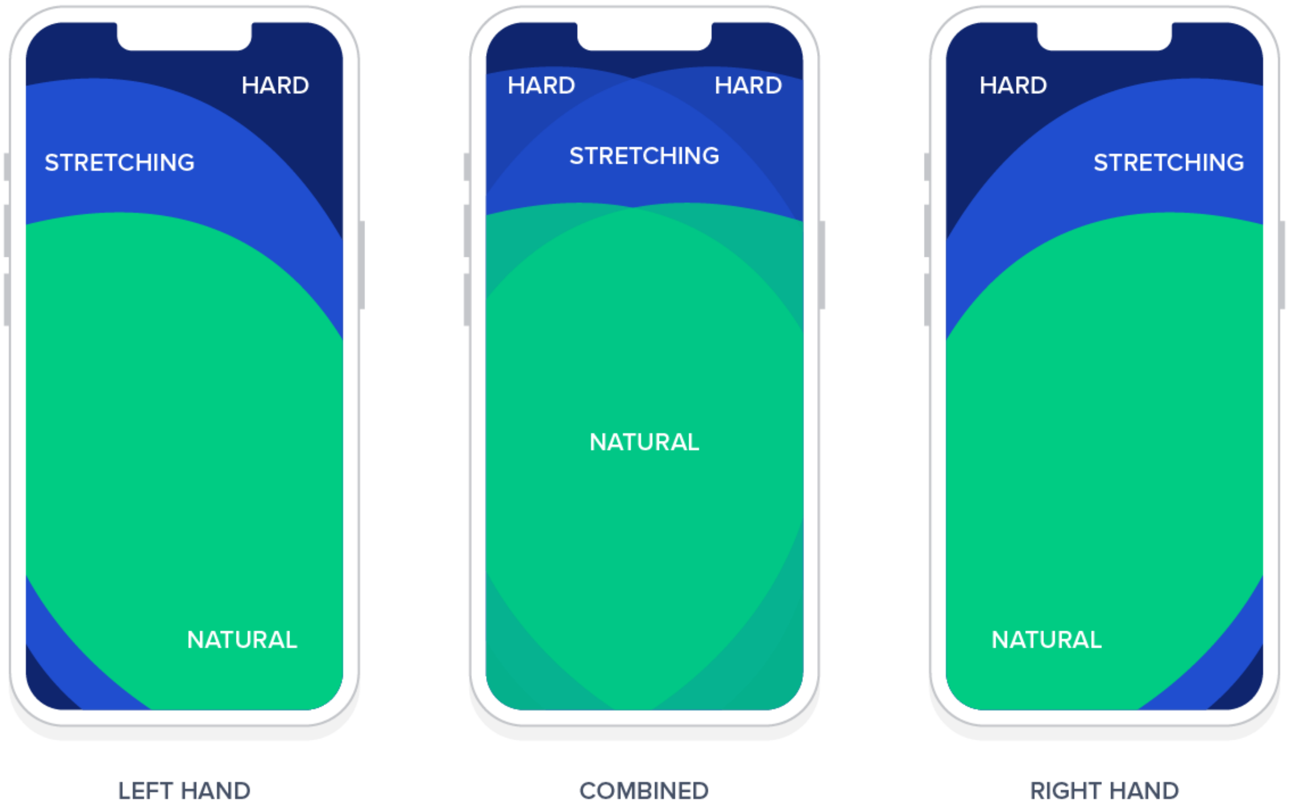

Grand Theft Auto: San Andreas – The Definitive Edition
(videogame)

# Fitt’s Law: Infinite Widths

- With a pointing device, targets near the edges have an infinite width
- These targets are fairly easy to hit, as $ID = \log_2(A/\infty + 1) = 0$!



# Fitt's Law: Mobile Devices




https://www.toptal.com/designers/mobile-ui/fitts-law-user-interface-design

# Readings and references


- Buxton, W. (1990). A Three-State Model of Graphical Input. In D. Diaper et al. (Eds), Human-Computer Interaction - INTERACT '90. Amsterdam: Elsevier Science Publishers B.V. (North-Holland), 449-456. https://www.dgp.toronto.edu/OTP/papers/bill.buxton/3state.html
- MacKenzie, I. S. (2003). Motor behaviour models for human-computer interaction. In *HCI models, theories, and frameworks: Toward a multidisciplinary science* https://www.yorku.ca/mack/mackenzie_chapter.html