

CS349

Input Performance

1

CS 349: Input Performance

17-Feb-12

- ❖ A02 marks should be posted today.
 - Remark requests by 5:00pm next Friday.
 - Speak to (email) TAs as your first step.
- ❖ Midterm Exam
 - Two weeks from today (March 2), 4:30-6:00pm
 - Watch Piazza for rooms, etc.
 - Coverage: to the end of today's lecture
 - Style: short answer, code, similar to questions you've been producing
 - Code: Exact library function names are not expected
 - Sample Exam: none

2

CS 349: Input Performance

Keystroke Level Model

- ❖ Problem: You're designing an interface and would like to model how long it takes to perform different tasks
 - Why?
- ❖ Keystroke Level Model
 - Describes a time cost for each action
 - A task is a sequence of actions, so simply sum the time costs involved.

3

CS 349: Input Performance

Keystroke Level Model

- ❖ Describe each task with a sequence of the following operators. Sum up times to estimate how long the task takes
 - K = Keystroking = 0.35s
 - P = Pointing = 1.10s
 - H = Homing = 0.4s
 - D = Drawing = variable time
 - M = Mental operator = 1.35s
 - R = Response by System = 1.2s

4

CS 349: Input Performance

Keystroke Level Model

- ❖ Benefits?
 -
 -
 -
- ❖ Drawbacks?
 - Some times estimates are out of date
 - Some time estimates are inherently variable
 - Doesn't model:
 - Errors
 - Learning time
 -
 -
 -
 -

5

CS 349: Input Performance

Keystroke Level Model

- ❖ Drawbacks (cont):
 - Equipment differs (eg: Trackpoint vs. touchpad)
 - All pointing takes 1.10s? *Really?*



6

CS 349: Input Performance

Better Estimates

❖ Fitts' Law: Pointing time without path constraints

- Published by Paul Fitts in 1954
- Psychologist at Ohio State University
- Theory based on rapid, aimed movement
- Most robust and highly adopted model of human hand movement



❖ Steering Law: Pointing time with path constraints

- Independently discovered three times:
 - Rashevsky (1959)
 - Drury (1971)
 - Zhai and Acott (1997)
- The last one was in the HCI community; most general mathematical result

7

CS 349: Input Performance

Fitts' Law

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

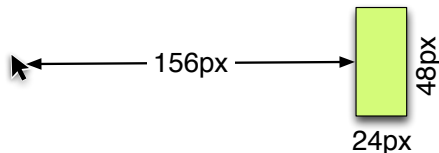
- ❖ T = time
- ❖ A = Amplitude of movement (distance between the starting point and the centre of the target)
- ❖ W = Constraining size of the target
- ❖ a and b are empirically determined based on the device and user

8

CS 349: Input Performance

Fitts' Law Example

- ❖ Assume $a = 1030$ and $b = 96$.



$$T = 1030 + 96 \log_2 \left(\frac{A}{W} + 1 \right)$$

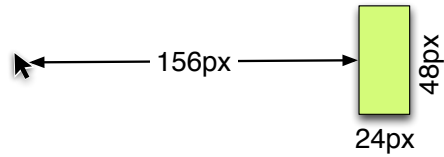
- ❖ What are the values for A and W ?

9

CS 349: Input Performance

Fitts' Law Example

❖ Assume $a = 1030$ and $b = 96$.



$$T = 1030 + 96 \log_2 \left(\frac{156 + 24 / 2}{24} + 1 \right)$$

$$T = 1030 + 96 \log_2 \left(\frac{168}{24} + 1 \right)$$

$$T = 1030 + 96 \log_2 (8)$$

$$T = 1030 + 96 \times 3$$

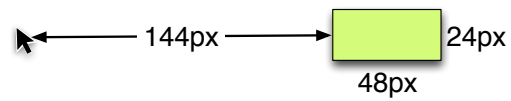
$$T = 1318ms$$

10

CS 349: Input Performance

Fitts' Law Example

❖ Assume $a = 1030$ and $b = 96$.



$$T = 1030 + 96 \log_2 \left(\frac{144 + 48 / 2}{24} + 1 \right)$$

$$T = 1030 + 96 \log_2 \left(\frac{168}{24} + 1 \right)$$

$$T = 1030 + 96 \log_2 (8)$$

$$T = 1030 + 96 \times 3$$

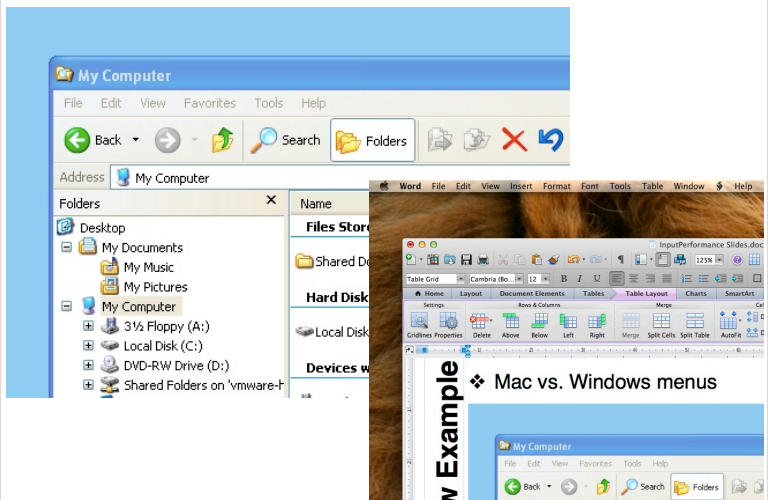
$$T = 1318ms$$

11

CS 349: Input Performance

Fitts' Law Example

❖ Mac vs. Windows menus



12

CS 349: Input Performance

Fitts' Law: ID

$$T = a + b \underbrace{\log_2 \left(\frac{A}{W} + 1 \right)}_{ID}$$

- ❖ ID = "Index of Difficulty"
- ❖ IP = "Index of Performance" = ID/T



13

CS 349: Input Performance

Physical Interpretation

- ❖ Larger objects are acquired faster than smaller objects; closer objects are acquired faster than more distant objects
- ❖ Gives mathematical rigor to what we already know intuitively:
 - the faster we move, the less precise our movements are
- ❖ Assumptions:
 - User is free to choose any path to the target.
 - Ignores the time to locate the target.

14

CS 349: Input Performance

Steering Law

- ❖ Steering Law is an adaptation of Fitts' Law
- ❖ Developed by Zhai and Acott
- ❖ Choose a paradigm which focuses on steering between boundaries

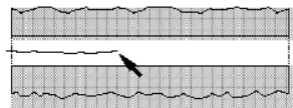


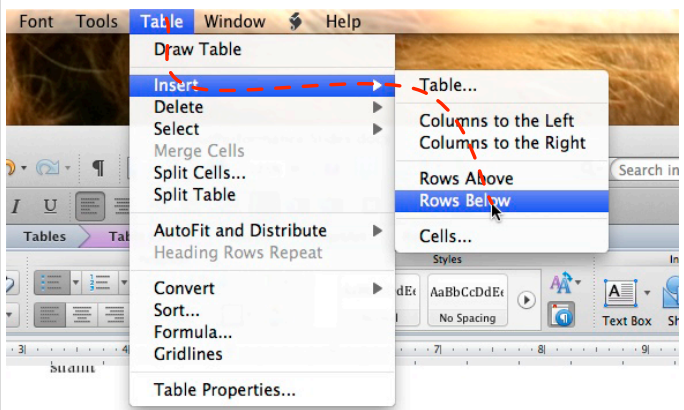
Figure 1: Self-paced movement with normal constraint

- ❖ Applicability?

15

CS 349: Input Performance

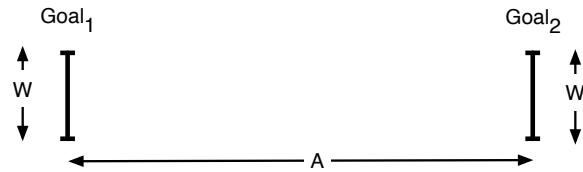
Steering Law



❖ Tracking a constrained path takes longer.

Steering Law

❖ 1st Experiment: Goal Passing



❖ Subjects passed a stylus from one end to the other

- As fast as possible
- Between each goal
- Several trials with different amplitudes (A) and widths (W)

❖ Result: Same law as Fitts' tapping task

Steering Law

❖ With only goals at the endpoints,

$$ID_1 = \log_2 \left(\frac{A}{W} + 1 \right)$$

❖ Adding N goals:



$$ID_N = \log_2 \left(\frac{A}{N \times W} + 1 \right)$$

Steering Law in a Tunnel

❖ When N approaches infinity, the task approaches steering through a tunnel (hierarchical menu).

❖ Index of Difficulty:

$$T = \lim_{N \rightarrow \infty} \sum_{i=1}^N b \log_2 \left(\frac{A}{N \times W} + 1 \right)$$

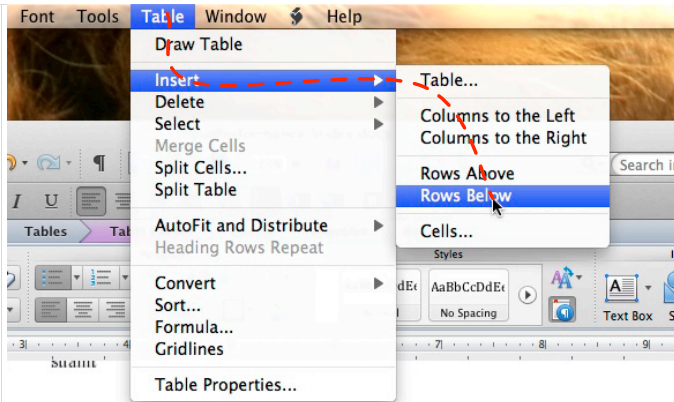
$$T = b \frac{A}{W}$$

❖ So difficulty is not related to $\log(A/W)$ but just A/W

19

CS 349: Input Performance

Hierarchical Menus



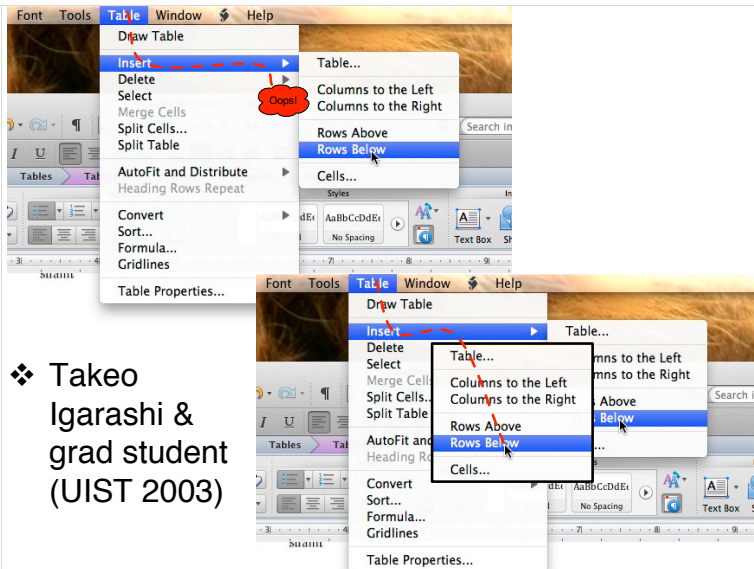
❖ Sum the parts of the path:

- Wide path (but short stopping distance)
- Narrow path (but wide stopping distance)
- Wide path (with short stopping distance)

20

CS 349: Input Performance

Improving Menus



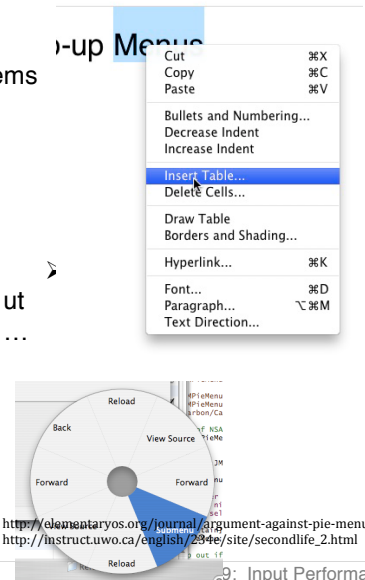
❖ Takeo Igarashi & grad student (UIST 2003)

21

CS 349: Input Performance

Improving Menus

- ❖ Pop-up Menus
 - Customized for action
 - Near mouse, but some items are still far away
- ❖ Pie Menus
 - All items are close

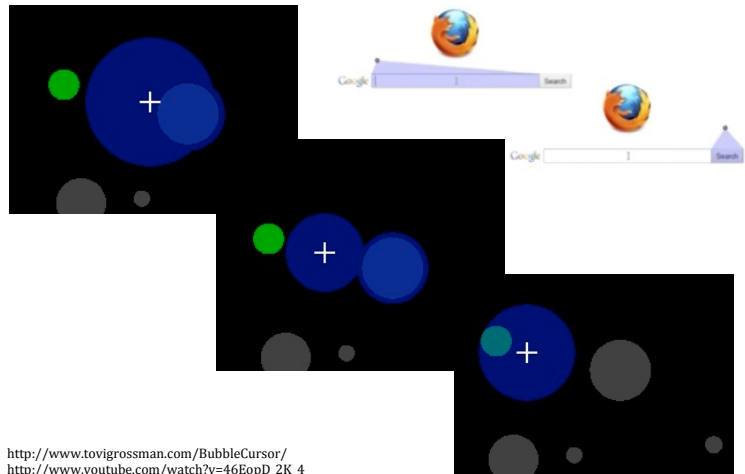


22

9: Input Performance

Bubble Cursors

- ❖ Change the size of the cursor to enclose exactly one nearby target.



23

CS 349: Input Performance

Motor vs. Screen Space

- ❖ How the cursor moves in response to mouse motion is under our control.
 - Making the cursor move more slowly when over the save button makes it larger in “motor space” even though it looks the same size in “screen space”.
 - It *LOOKS* the same on the screen, but the “Save” button is “sticky”.
 - Faster to click “Save” than Fitts’ Law (in pixels) would indicate.



24

CS 349: Input Performance

