Input Performance

KLM, Fitts' Law, Pointing Interaction Techniques

Input Performance Models

- You're designing an interface and would like to:
  - choose between candidate designs without building them
  - estimate performance with your new design
- Solution: use a model of how people use input devices and interfaces to predict time, error, fatigue, learning, etc.
  - models most often focus on time and error (easiest to measure)

Keystroke Level Model (KLM)

- Describe each task with a sequence of operators
- Sum up times to estimate how long the task takes
- Operator types
  - **K** Keystroke = 0.8 – 1.2s (based on expertise, type of string)
  - **P** Pointing = 1.10s
  - **B** Button press on mouse = 0.1s
  - **H** Hand move from mouse to/from keyboard = 0.4s
  - **M** Mental preparation = 1.2s
- KLM is simplified GOMS, so sometimes called KLM-GOMS
- Great online resource for KLM (Kieras, 1993):
  - ftp://ai.eecs.umich.edu/people/kieras/GOMS/KLM.pdf
- KLM Time Calculator
  - http://courses.csail.mit.edu/6.831/2009/handouts/ac18-predictive-evaluation/klm.shtml

KLM Operators

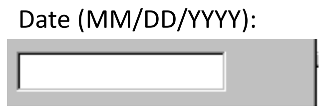
main physical operators

Code	Operation		Time
K	Key press and release (keyboard)	Best Typist (135 wpm)	0.08 seconds
		Good Typist (90 wpm)	0.12 seconds
		Poor Typist (40 wpm)	0.28 seconds
		Average Skilled Typist (55 wpm)	0.20 seconds
		Average Non-secretary Typist (40 wpm)	0.28 seconds
		Typing Random Letters	0.50 seconds
		Typing Complex Codes	0.75 seconds
		Worst Typist (unfamiliar with keyboard)	1.20 seconds
P	Point the mouse to an object on screen		1.10 seconds
B	Button press or release (mouse)		0.10 seconds
H	Hand from keyboard to mouse or vice versa		0.40 seconds
M	Mental preparation		1.20 seconds

KLM Example (Only Physical Operators)

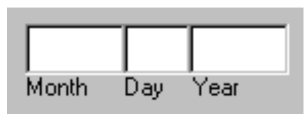
- Use KLM to compare the performance time of three different date entry widgets. (assume: hand already on mouse, 40 WPM typist)

PBH(Kx10)
- One text field



(PBPB)x3
- Three Dropdowns

PBH(Kx10)
- Three text fields



Op	Time
K	0.3
P	1.1
B	0.1
H	0.4
~~M~~	~~1.2~~

Including Mental Operators (M)

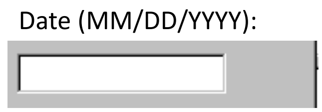
- People need to think about something before doing it
  - identify when people have to stop and think: **M**
  - difference between actions using *cognitive conscious* and *cognitive unconscious*
- Insert an **M** operation when people have to:
  - initiate a task
  - make a strategy decision
  - retrieve a chunk from memory
  - find something on the display (e.g. point to something)
  - think of a task parameter
  - verify that a specification/action is correct (e.g. display changes)
- Can use **M** to model novice and expert
  - add M in front of any action if they're a novice

KLM Example (Including Mental Operators)

- Use KLM to compare the performance time of three different date entry widgets. (assume: hand already on mouse, 40 WPM typist)

$MPBH(K*10)$

- One text field



$M(PBMPB*3)$

- Three Dropdowns

$MPBHKK +$
$HPBHKK +$
$HPBHKKKK$

- Three text fields

Op	Time
K	0.3
P	1.1
B	0.1
H	0.4
M	1.2

KLM Exercise

- Use KLM to compare different designs for deleting a file (assume: hand already on mouse, 40 WPM typist, file and trashcan are visible, return to original window when done)
- Do it without, and with, mental operators
- Designs:
  1. Select file and drag it trash can
  2. Select file and choose File/Delete from main menu
  3. Select file and delete with 'Del' shortcut key
  4. Select file and choose Delete from right-click context menu

- (solutions to 1,2,3 in ftp://ai.eecs.umich.edu/people/kieras/GOMS/KLM.pdf)

KLM Critique

Benefits?

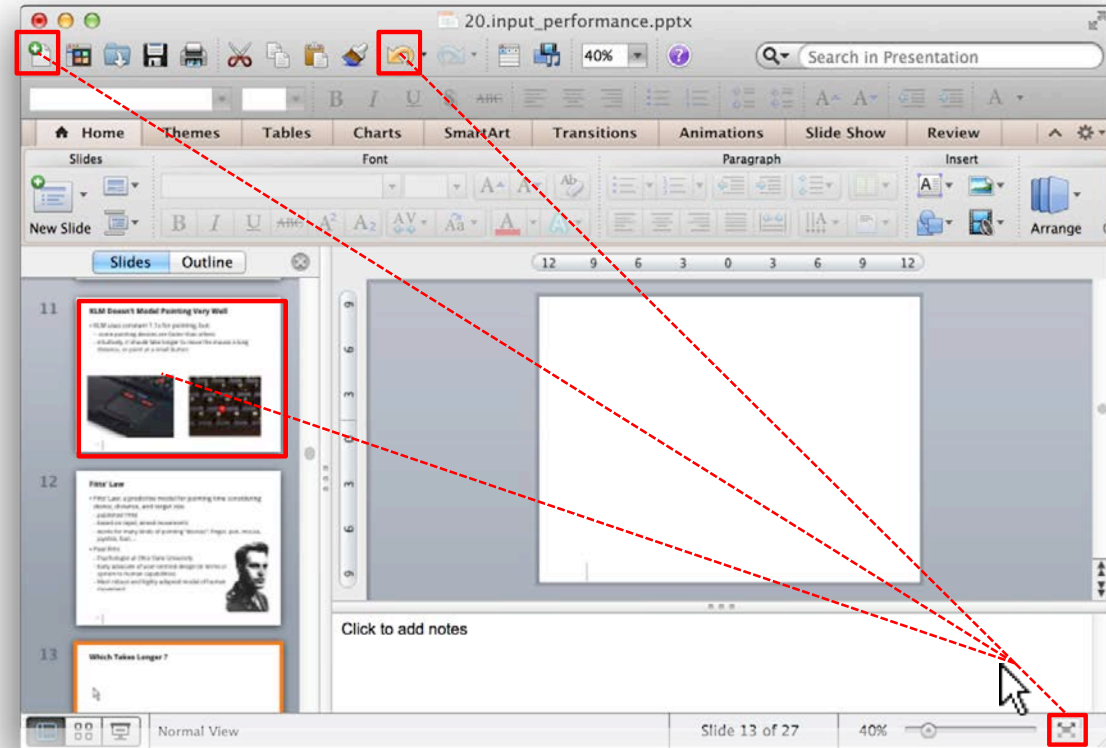
Drawbacks?

- Some time estimates are out of date
- Some time estimates are inherently variable
- Doesn’t model:
  - Errors
  - Learning time
  - etc.

KLM Doesn’t Model Pointing Very Well

- KLM uses constant 1.1s for pointing, but:
  - some pointing devices are faster than others
  - intuitively, it should take longer to move the mouse a long distance, or point at a small button

Which Takes Longer?



Fitts' Law

- Fitts' Law: a predictive model for pointing time considering device, distance, and target size
  - published 1954
  - based on rapid, aimed movements
  - works for many kinds of pointing "devices": finger, pen, mouse, joystick, foot, ..
- Paul Fitts
  - Psychologist at Ohio State University
  - Early advocate of user-centred design (in terms of matching system to human capabilities)

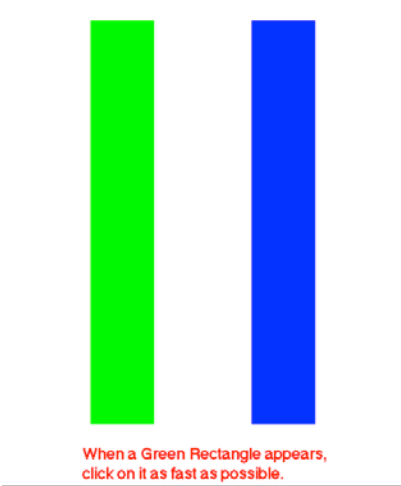
Distance vs. Size

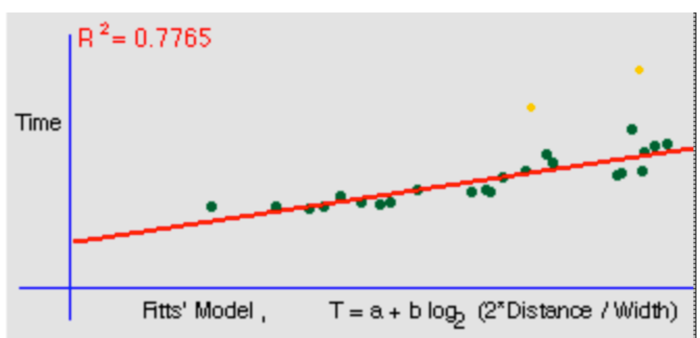
- The larger the **distance**, the longer the **time**
- The smaller the **size** of the target, the longer the **time**
- So, a proportional relationship between movement time and distance and size:

$$MT \propto \frac{D}{S}$$

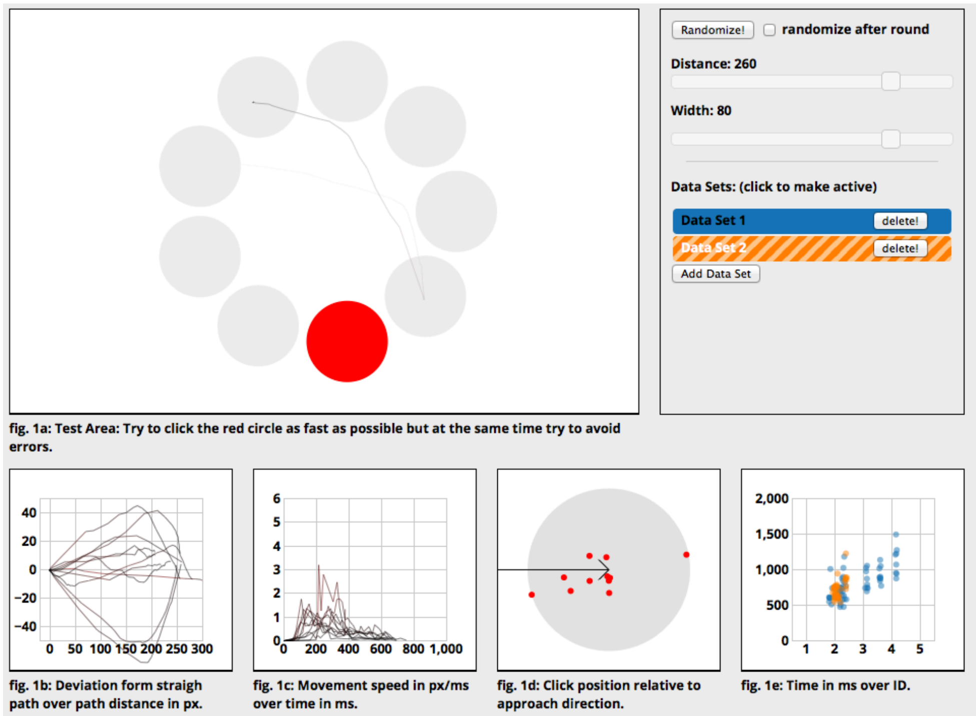
- But …
  – what is meant by target “size”?
  – a proportional relationship isn’t a model …

http://husk.eecs.berkeley.edu/projects/fitts/





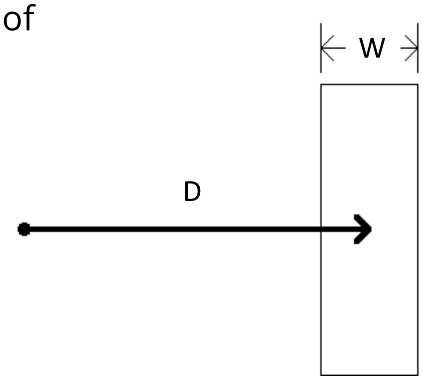
http://www.simonwallner.at/ext/fitts/



Linear Regression

- Movement time varies according to log of Distance and target "Width" (assume 1 dimension for the moment):

$$MT \propto \log \frac{D}{W}$$



- It's a linear regression, so it has a slope 'b' and intercept 'a' …

$$MT \propto a + b \log_2 \frac{D}{W}$$

Fitts' Law

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

- MT = movement time
- D = distance between the starting point and the centre of the target (D is often shown as 'A' for Amplitude)
- W = Constraining size of the target
- a and b are characteristics of input device

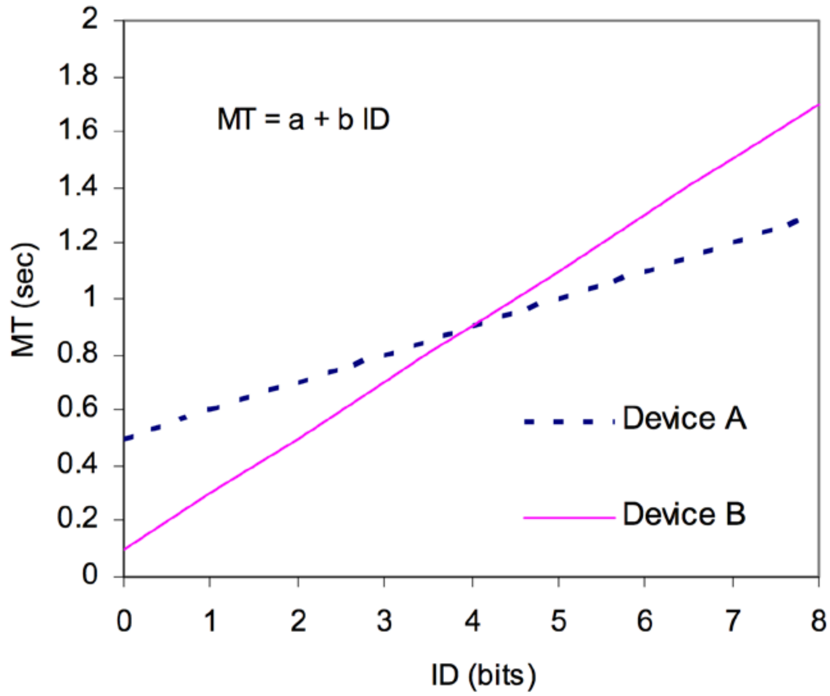
Fitts' Law: Index of Difficulty

$$MT = a + b\,\underbrace{\log_2\left(\frac{D}{W} + 1\right)}$$

IP = "Index of Performance" = 1/b

ID = "Index of Difficulty"

Device Characteristics (**a** and **b** parameters)



a, b, and IP for different devices

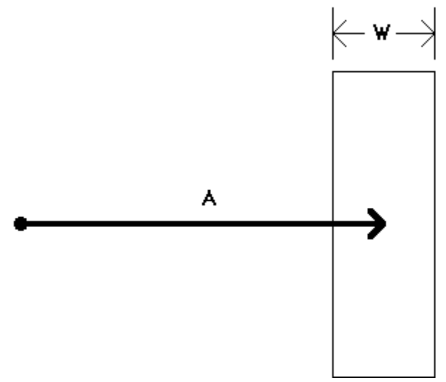
		Regression Coefficients		
Device	*r*[a]	Intercept, *a*(ms)	Slope, *b* (ms/bit>	*IP* (bits/s)[b]
		*** Pointing ***		
Mouse	.990	−107	223	4.5
Tablet	.988	−55	204	4.9
Trackball	.981	75	300	3.3
		*** Dragging ***		
Mouse	.992	135	249	4.0
Tablet	.992	−27	276	3.6
Trackball	.923	−349	688	1.5

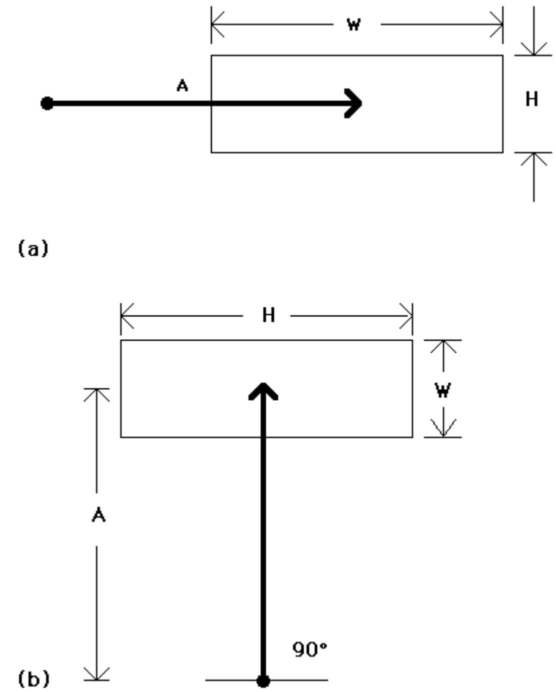
[a] $n = 16$, $p < .001$
[b] *IP* (index of performance) = 1/*b*

Figure 7. Fitts' law models. A regression analysis for each device-task combination shows the correlation (*r*), intercept (*a*), slope (*b*), and index of performance ($IP = 1/b$). Prediction equations are of the form $MT = a + b\ ID$, where $ID = \log_2(A/W + 1)$.

http://www.billbuxton.com/fitts91.html

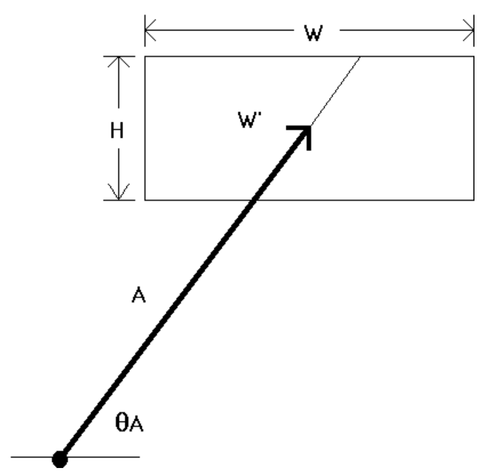
2D Targets?





http://www.yorku.ca/mack/CHI92.html
(remember ‘A’ = Amplitude = ‘D’ = Distance)

2D Targets: W’ as Cross Section Given Approach



- But hard to know approach angle a priori …

http://www.yorku.ca/mack/CHI92.html
(remember ‘A’ = Amplitude = ‘D’ = Distance)

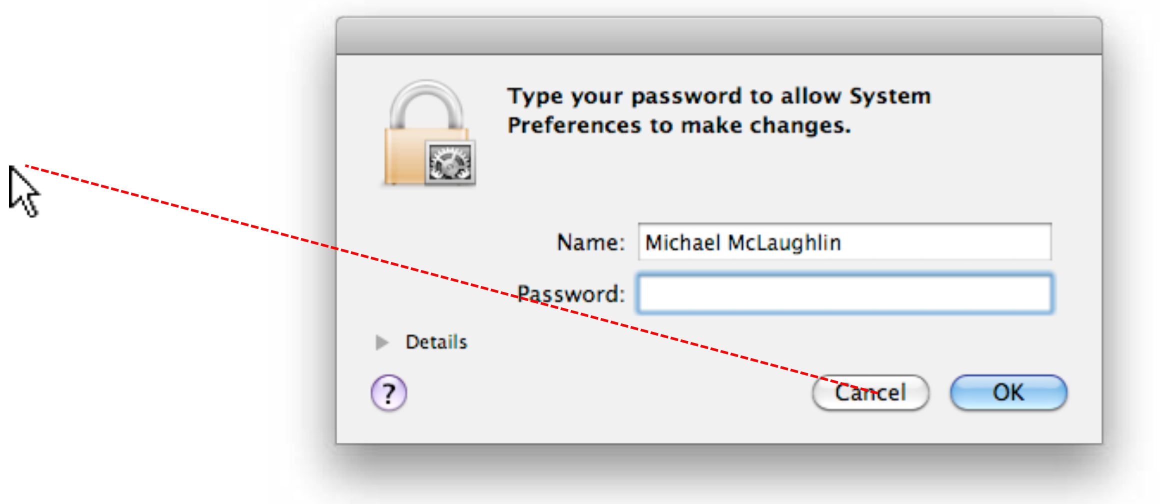
2D Targets: "W" is Minimum of Target W and H

$$MT = a + b \log_2 \left(\frac{D}{\min(W, H)} + 1 \right)$$

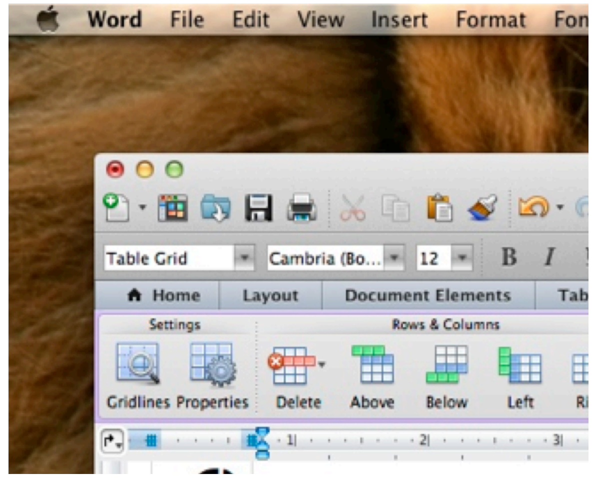
… but usually just write W assuming it's the minimum of target W and H

Fitts' Law Example

- Using a mouse to point (a = -107 and b = 223), what is the movement time to click on a 80 pixel by 32 pixel Cancel button located 400 pixels away?



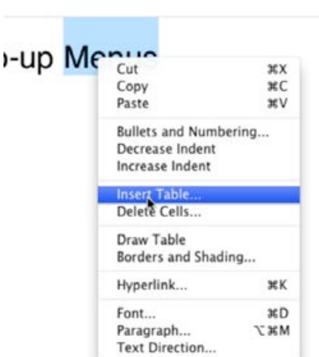
Menu Target Size in OSX and Windows



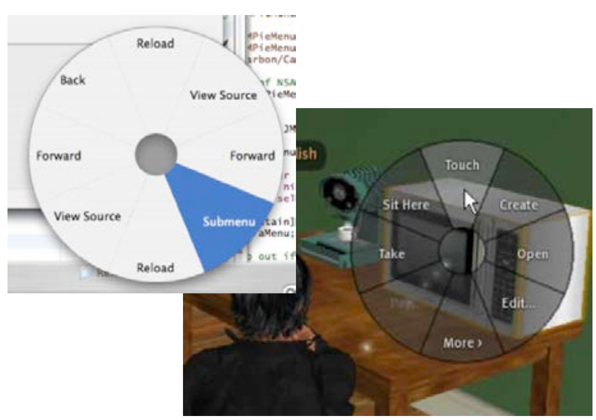
Context Menus, Pie Menus, Marking Menus

- Context Menu lowers D, but some items closer than others
- Pie Menus near mouse, all items same D (optimal)

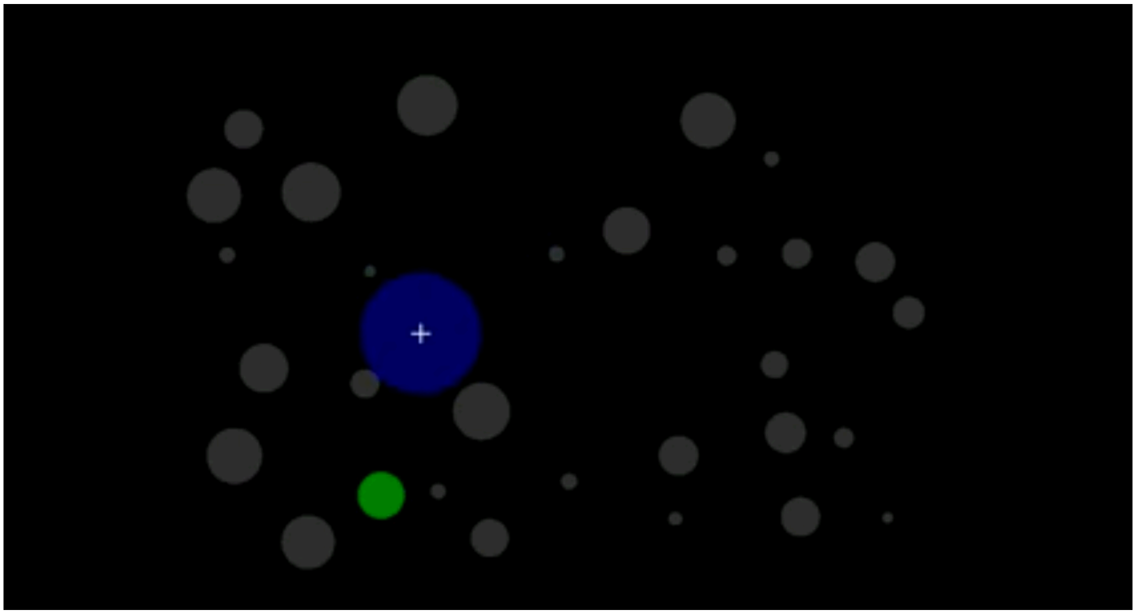
context menu



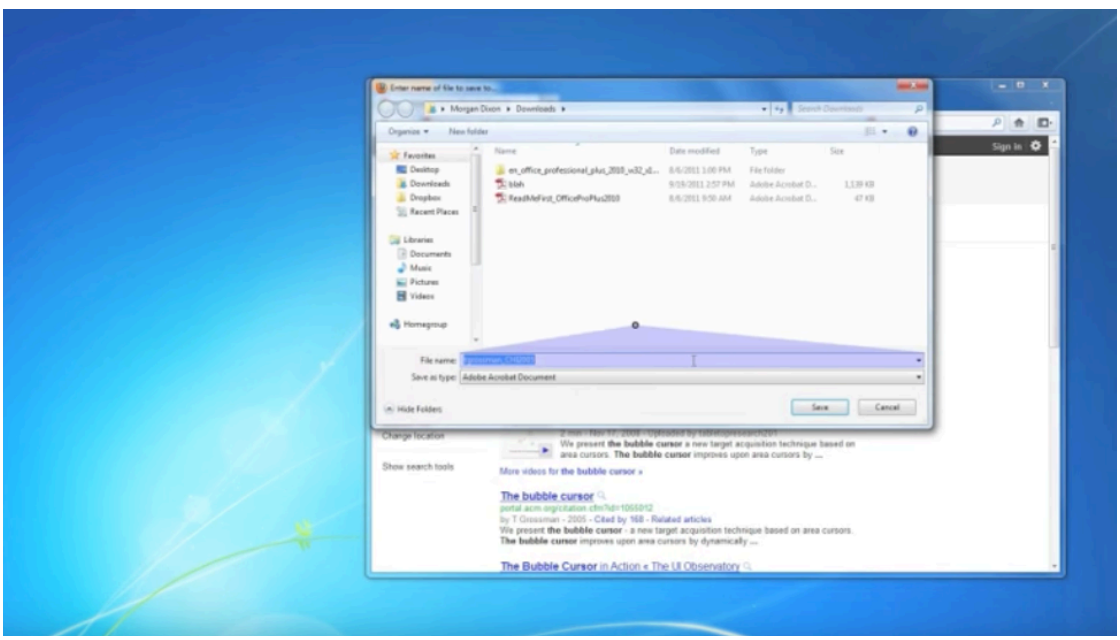
pie menu



http://elementaryos.org/journal/argument-against-pie-menus
http://instruct.uwo.ca/english/234e/site/secondlife_2.html



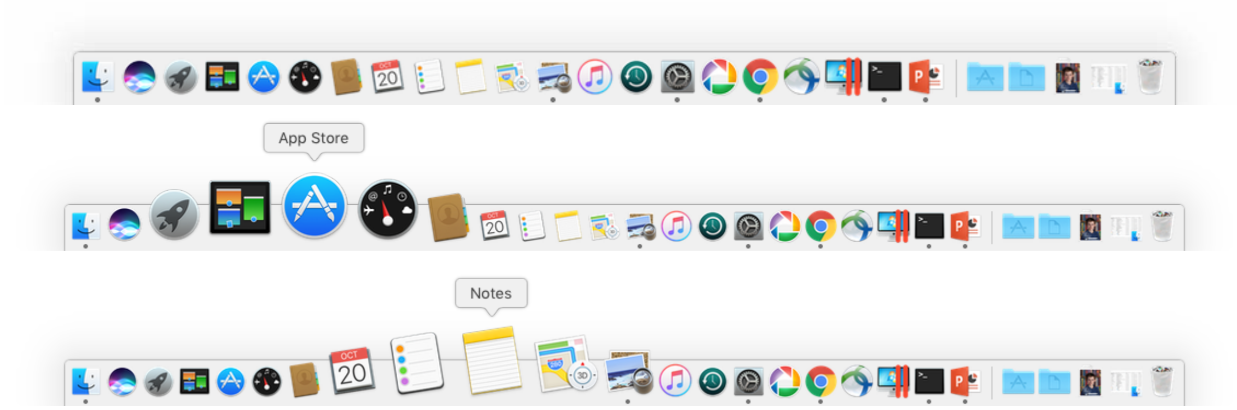
Bubble Cursor (Grossman and Balakrishnan, 2005)
– http://youtu.be/JUBXkD_8ZeQ



A General-Purpose Bubble Cursor using Prefab (Dixon et al. 2012)
– https://youtu.be/46EopD_2K_4

OSX Dock Expansion

- OSX Dock expands in visual space, but not motor space …
- Fitts's law says selecting an expanded target on the dock is no easier than the default small targets



McGuffin, M. J., & Balakrishnan, R. (2005). Fitts' law and expanding targets: Experimental studies and designs for user interfaces. ACM Transactions on Computer-Human Interaction (TOCHI), 12(4), 388-422.

Motor Space vs. Screen Space

- Dynamically change CD Gain based on position of cursor
  - 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".
  - LOOKS the same on screen, but "Save" button is "sticky".
  - Faster to click "Save" (if Fitts' Law calculated in motor space).

visual space

motor space

Steering Law

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

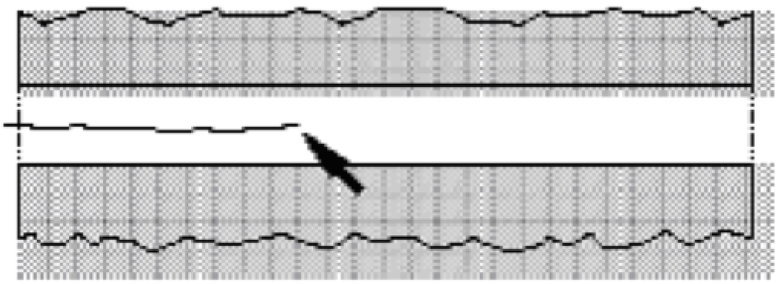
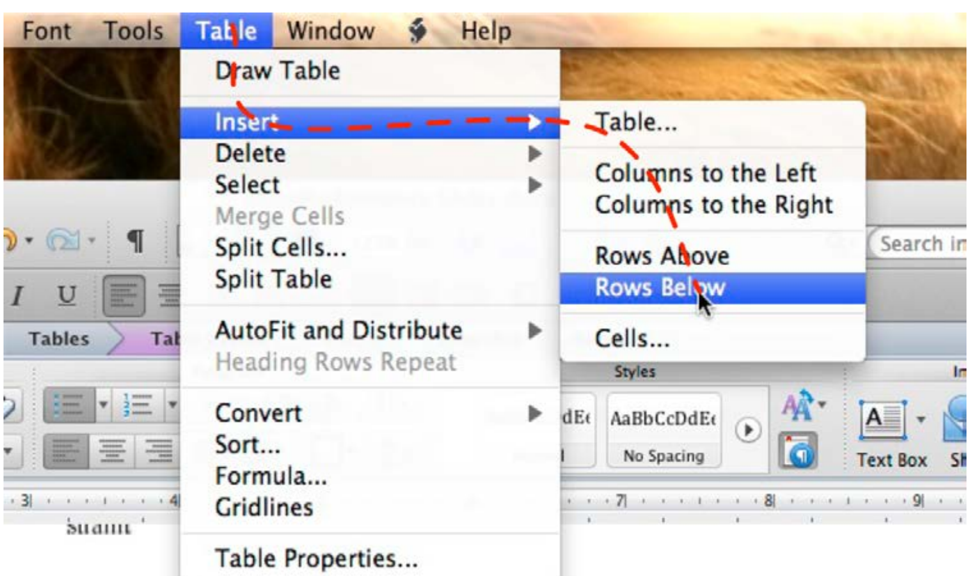


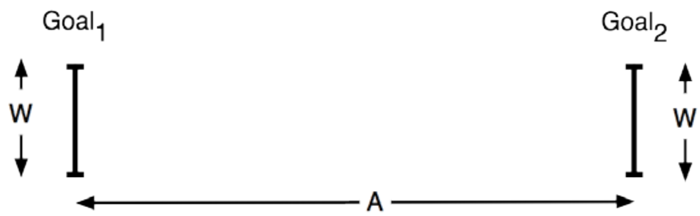
Figure 1: Self-paced movement with normal constraint

Steering Law

- Tracking a constrained path takes longer



Steering Law: 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: Goal Passing

- 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: Goal Passing

- When N approaches infinity, the task approaches steering through a tunnel (hierarchical menu).
- Index of Difficulty:

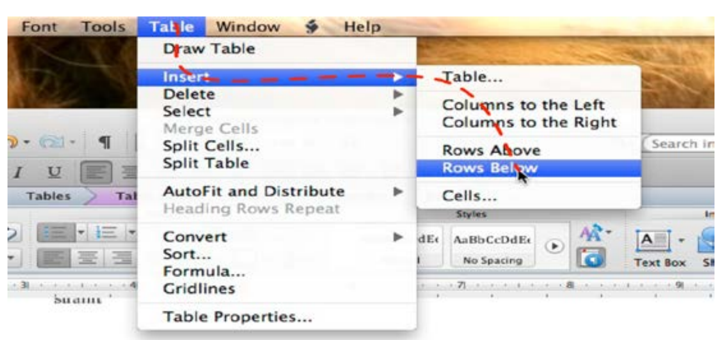
$$T = \lim_{N \to \infty} \sum_{i-1}^{N} b \log_2 \left(\frac{A/N}{W} + 1 \right)$$

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

- So difficulty is not related to log(A/W) but just A/W

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)



Summary

- We have mathematical models for acquiring a target, both when the path is unconstrained and constrained
  - Larger/closer is faster
- Gives some ideas for speeding things up
  - Keep things close (contextual, pie-menus)
  - Make things larger (bubble cursors)
  - Manipulate motor space to make intended targets stickier