



人机交互系统

交互设计模型

主讲教师：冯桂焕



背景

- 汽车上的刹车踏板和油门踏板相距很近，且刹车踏板要比油门踏板大很多
 - 经验告诉我们，可达到以最快的速度准确制动的目的
 - 但是，依据的原理是什么呢？
- 设计学科通常借助模型生成新的想法并对其测试
 - 如建筑学领域，有重量分布模型、空气环流模型、流体力学模型和光学模型等
- 交互设计领域
 - 计算用户完成任务的时间：KLM
 - 描述交互过程中系统状态的变化：状态转移网
 - 探讨任务的执行方法等：GOMS



■ 预测模型



预测模型

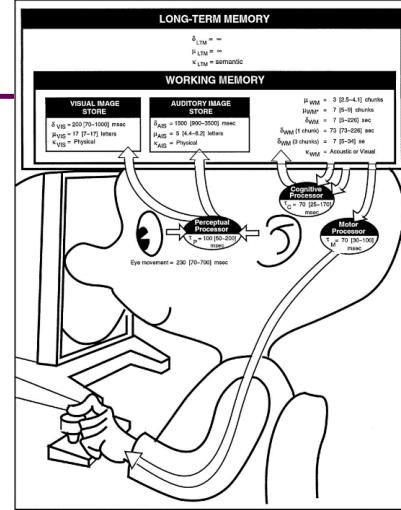
- 能够预测用户的执行情况，但不需要对用户做实际测试
- 特别适合于无法进行用户测试的情形
 - 举例：为改进对员工使用计算机的支持，设计了许多可行方案。如何判断那一种方法更有效？
- 不同模型关注用户执行的不同方面
 - GOMS
 - 击键层次模型KLM
 - Fitts定律



GOMS模型

■最著名的预测模型

- 1983年由Card, Morgan和Newell提出
- 基于人类处理机模型
- 泛指整个GOMS模型体系



■是关于人类如何执行认知—动作型任务以及如何与系统交互的理论模型

- 采用“分而治之”的思想，将一个任务进行多层次的细化
- 把每个操作的时间相加就可以得到一项任务的时间
- 操作指用户的目光从屏幕的一处移到另一处、识别出某个图标、手移到鼠标上



GOMS全称

■ Goal-目标

- 用户要达到什么目的

■ Operator-操作

- 任务执行的底层行为，不能分解
 - 为达到目标而使用的认知过程和物理行为
- 如点击鼠标

■ Method-方法

- 如何完成目标的过程，即对应目标的子目标序列和所需操作
- 如移动鼠标，输入关键字，点击Go按钮

■ Selection-选择规则

- 确定当有多种方法时选择和方法
- GOMS认为方法的选择不是随机的



举例

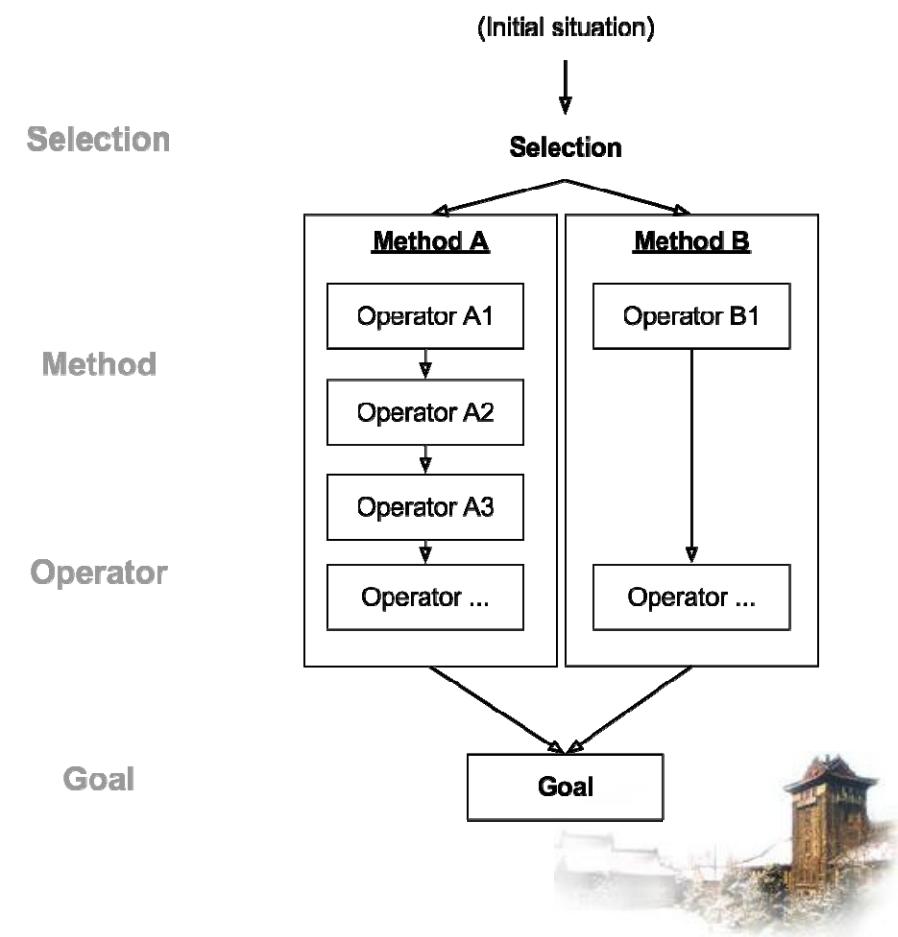
■ 使用GOMS模型描述在Word中删除文本的过程

- 目标：删除Word中的文本
- 方法1：使用菜单删除文本
 - 步骤1：思考，需要选定待删除的文本
 - 步骤2：思考，应使用“剪切”命令
 - 步骤3：思考，“剪切”命令在“开始”菜单中
 - 步骤4：选定待删除文本，执行“剪切”命令
 - 步骤5：达到目标，返回



GOMS方法步骤

- 选出最高层的用户目标
- 写出具体的完成目标的方法
 - 即激活子目标
- 写出子目标的方法
 - 递归过程，一直分解到最底层操作时停止
- 子目标的关系
 - 顺序关系
 - 选择关系
 - 以select: 引导



GOMS模型分析

■优点

- 能够容易地对不同的界面或系统进行比较分析
- 美国电话公司NYNEX
 - 利用GOMS分析一套即将被采用的新的计算机系统的应用效果不理想，放弃了使用新系统，为公司节约了数百万的资金。

■局限性

- 假设用户完全按一种正确的方式进行人机交互，没有清楚地描述错误处理的过程
- 只针对那些不犯任何错误的专家用户
- 任务之间的关系描述过于简单
- 忽略了用户间的个体差异



击键层次模型

■ Card等1983

■ 对用户执行情况进行**量化预测**

- 仅涉及任务性能的一个方面：时间

■ 用途

- 预测无错误情况下专家用户在下列输入前提下完成任务的时间

- 便于比较不同系统

- 确定何种方案能最有效地支持特定任务



操作符

操作符名称	描述	时间（秒）
K	按下一个单独按键或按钮	0.35（平均值）
	熟练打字员（每分钟键入 55 个单词）	0.22
	一般打字员（每分钟键入 40 个单词）	0.28
	对键盘不熟悉的人	1.20
	按下 shift 键或 ctrl 键	0.08
P	使用鼠标或其他设备指向屏幕上某一位置	1.10
P ₁	按下鼠标或其他相似设备的按键	0.20
H	把手放回键盘或其他设备	0.40
D	用鼠标画线	取决于画线的长度
M	做某件事的心理准备（例如做决定）	1.35
R(t)	系统响应时间——仅当用户执行任务过程中需要等待时才被计算	t



使用

■ 执行时间预测方法

- 列出操作次序，累加每一项操作的预计时间
- $T_{execute} = T_k + T_p + T_h + T_d + T_m + T_r$

■ 举例

- DOS环境下执行 “ipconfig”命令
 - MK[i] K[p] K[c] K[o] K[n] K[f] K[i] K[g] K[回车]
 - 简略表达版本： M9K[ipconfig回车]
- $T_{execute} = 1.35 + 9 \times 0.28 = 3.87\text{s}$
- 菜单选择
 - H[鼠标]MP[网络连接图标] P1[右键]P[修复] P1[左键]
 - $T_{execute} = 0.40 + 1.35 + 2P + 2K = 4.35\text{秒}$



编码方法

■ 举例：替换文字编辑器中长度为5个字符的单词

- 任务准备 M
- 将手放在鼠标上 H_{mouse}
- 将鼠标移到单词 P_{word}
- 选择单词 P_1P
- 回到键盘 $H_{keyboard}$
- 准备键入 M
- 键入新的5字符单词 $5K_{word}$



$$\begin{aligned} \text{■ } T_{execute} &= 2M + 2H + P + P_1 + P + 4K \\ &= 2.70 + 0.80 + 1.10 + 0.20 + 1.10 + 0.88 = 6.78s \end{aligned}$$



放置M操作符的启发规则

■问题

- 如何确定是否需要在具体操作之前引入一个思维过程呢？

■答案

-



放置M操作符的启发规则

以编码所有的物理操作和响应操作为开端。接着使用规则 0 放置所有的候选 M 操作符，然后循环执行规则 1 到 4，并对每一个 M 操作判断是否应该删除

规则 0	在所有 K 操作符前插入 M 操作符，要求 K 操作的值不能是参数字符串（如数字或文本）的一部分。在所有的对应于选择命令（非参数）的 P 操作符前放置 M 操作
规则 1	如果某个操作符前的 M 操作完全可以由 M 之前的操作符预测，则删除 M（如 PMK—PK）
规则 2	如果一串 MK 组成的字符串是一个认知单元（如一个命令的名字），则删除第一个 M 以外的所有 M
规则 3	如果 K 是一个冗余的终结符（如紧跟在命令参数终结符后面的命令终结符），则删除 K 之前的 M
规则 4	如果 K 是常量字符串（如一个命令名）的终结符，则删除 K 之前的 M。如果 K 是变量字符串（如参数字符串）的终结符，则保留 K 之前的 M



放置M的启发式规则

- 1: 在每一步需要访问长时记忆区的操作前放置一个M
- 2: 在所有K和P之前放置M

K -> MK; P -> MP

- 3: 删除键入单词或字符串之间的M

MKMKMK -> MKKK

- 4: 删除复合操作之间的M (如, 选中P和点击P₁)

MPMP₁ -> MPP₁



Adapted from Rob Miller

KLM分析

- 建模可以给出执行标准任务的时间
- 但没有考虑下面的问题

- 错误
- 学习性
- 功能性
- 回忆
- 专注程度
- 疲劳
- 可接受性



Fitts定律

- 用户访问屏幕组件的时间对于系统的使用效率是至关重要的
 - 哪些特性会影响访问效率呢?
 - Fitts, 1954
- 能够预测使用某种定位设备指向某个目标的时间
- 人机交互中，根据目标大小及至目标的距离，计算指向该目标的时间
 - 可指导设计人员设计按钮的位置、大小和密集程度
- 对图形用户界面设计有明显的意义
- “最健壮并被广泛采用的人类运动模型之一”



“轮流轻拍”实验

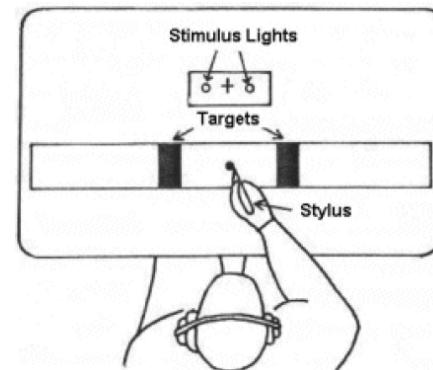
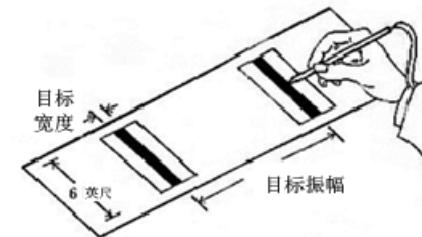
■记录拍中和失误的情况

■指令

●尽可能准确而不是快速的轮流轻拍两个薄板

■以实验数据为依据，得到困难指数如下

$$ID = \log_2(2A/W)$$



- Paul M. Fitts (1954). The information capacity of the human motor system in controlling the amplitude of movement. *Journal of Experimental Psychology*, volume 47, number 6, June 1954, pp. 381-391. (Reprinted in *Journal of Experimental Psychology: General*, 121(3):262–269, 1992).
- Paul M. Fitts and James R. Peterson (1964). Information capacity of discrete motor responses. *Journal of Experimental Psychology*, 67(2):103–112, February 1964.



- Fitts定律描述了人类运动系统的信息量
- 信息论中的Shannon定理

$$C = B \log_2(S/N+1)$$

- C是有效信息量(比特), B是通道带宽, S是信号能量, N是噪声

- Fitts定律
- S映射为运动距离或振幅(A), N映射为目标的宽度(W)



三个部分

■ 困难指数ID (Index of Difficulty) = $\log_2(2A/W)$ (bits)

- 对任务困难程度的量化
- 与宽度和距离有关

■ 运动时间MT (Movement Time) = $a + b * ID$ (secs)

- 在ID基础上将完成任务的时间量化

■ 性能指数IP (Index of Performance) = ID/MT (bits/sec)

- 基于MT和ID的关系
- 也称吞吐量



■ MacKenzie改写为

$$ID = \log_2(2A/W)$$

$$ID = \log_2(A/W+1)$$

- 更好地符合观察数据
- 精确地模拟了支撑Fitts定律的信息论 $C = B \log_2(S/N+1)$
- 计算出的任务困难指数总是整数

■ 平均时间 MT

$$MT = a + b \log_2(A/W+1)$$

- 常数a和b来自实验数据的线性回归



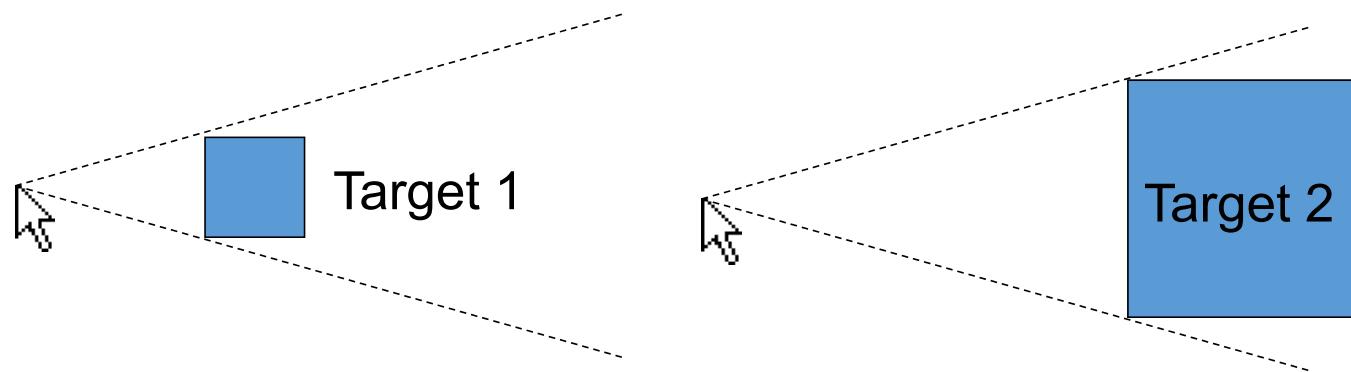
说明

- 如果MT的计算单位是秒，则a的测量单位是秒，b的测量单位是秒/比特
(ID的测量单位是比特)
- 系数a(截距)和b(斜率)由经验数据确定，且与设备相关
- 对于一般性计算，可使用 $a=50, b=150$ (单位是毫秒)
- A和W在距离测量单位上必须一致，但是不需要说明使用的具体单位



Fitts' Law

$$MT = a + b * ID, ID = \log_2(A/W+1)$$

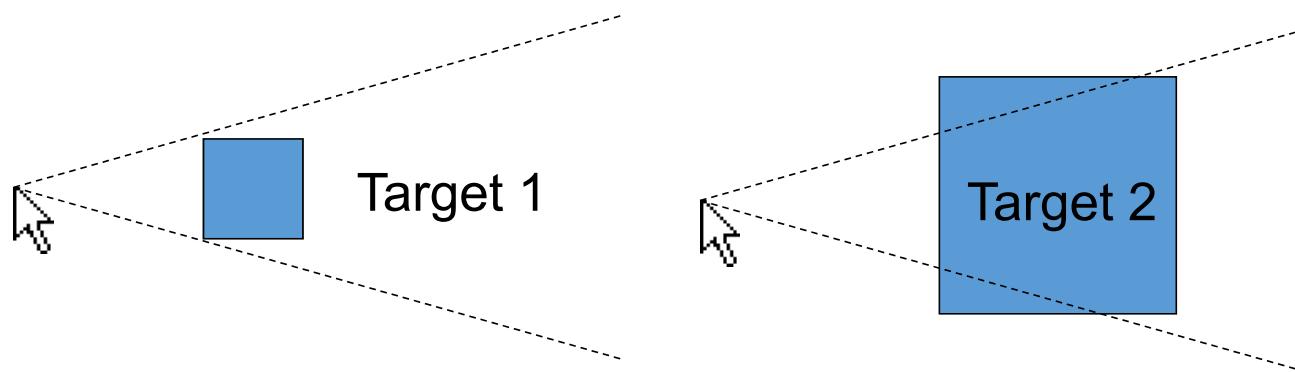


Same ID → Same Difficulty



Fitts' Law

$$MT = a + b * ID, ID = \log_2(A/W+1)$$

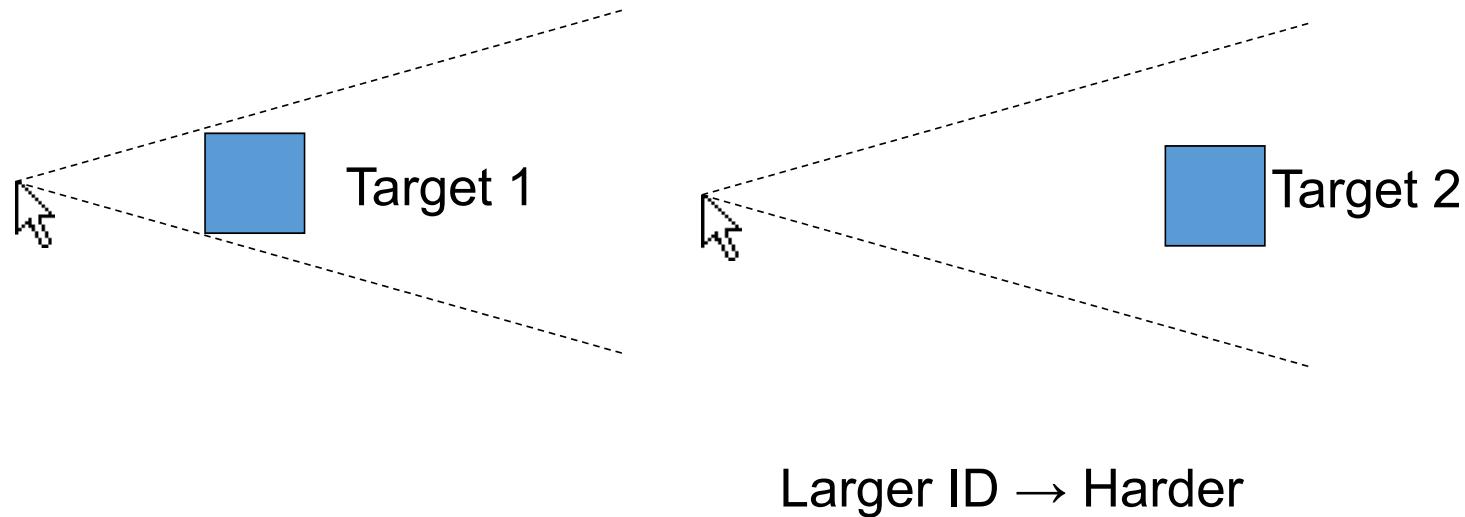


Smaller ID → Easier



Fitts' Law

$$MT = a + b * ID, ID = \log_2(A/W+1)$$



Fitts定律建议

■大目标、小距离具有优势

- 对选择任务而言，其移动时间随到目标距离的增加而增加，随目标的大小减小而增加

■屏幕元素应该尽可能多的占据屏幕空间

■最好的像素是光标所处的像素

■屏幕元素应尽可能利用屏幕边缘的优势

■大菜单，如饼型菜单，比其他类型的菜单使用简单

$$ID = \log_2(A/W+1)$$



Fitts定律应用

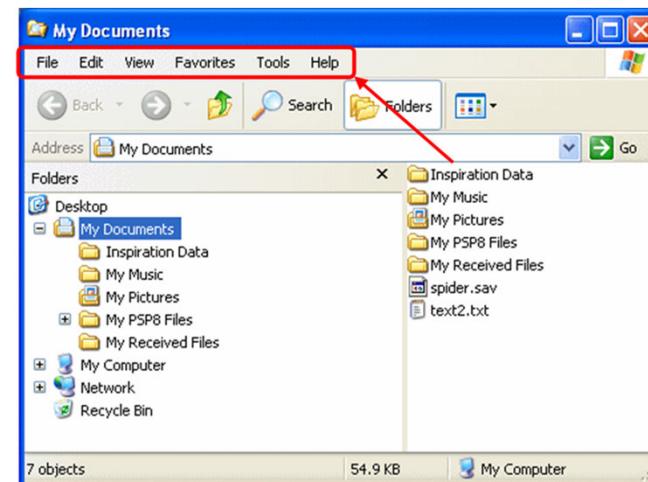
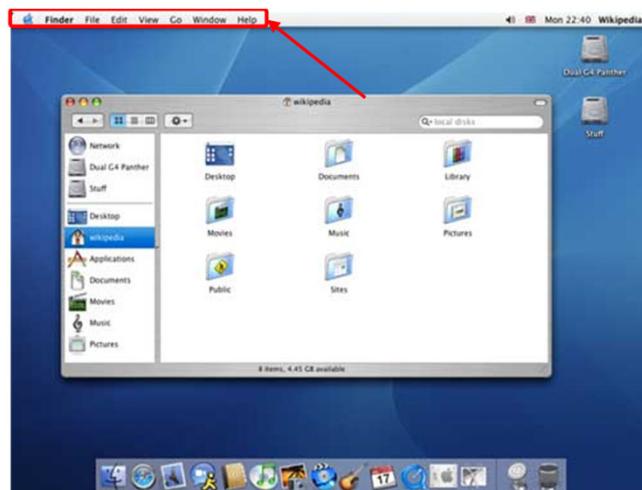
- 首先被Card等人应用在HCI领域
 - 鼠标的定位时间和错误率都优于其他设备
 - 鼠标速率接近最快速率
 - 使用鼠标完成运动任务比使用其他设备更加协调，这在交互设计中非常重要
- 策略一：缩短当前位置到目标区域的距离
 - 如右键菜单技术
- 策略二：增大目标大小以缩短定位时间
 - Windows操作系统和Macintosh操作系统中的应用程序菜单区域位置的设计



应用实例

■ Mac OS和Windows XP的比较 (苹果专利)

- Mac OS的菜单是沿着屏幕边缘排列的
- Windows OS的菜单位于标题栏下面



■ 用户往往在距离屏幕边缘50毫米处停下来

- 50毫米作为Mac OS的菜单宽度

■ 对于Mac OS

- $ID = 50 + 150 \log_2(80/50+1) = 256$ 微妙

■ 对于Windows OS

- $ID = 50 + 150 \log_2(80/5+1) = 663$ 微妙



Mac OS “dock”

■工具栏组件大小可以动态改变

- 为用户提供了一个放大的目标区域
- 可显示更多图标
- 新版Mac操作系统中都实现了扩展工具栏

■思考：

- 该工具栏存在何种优缺点？



Hick's Law

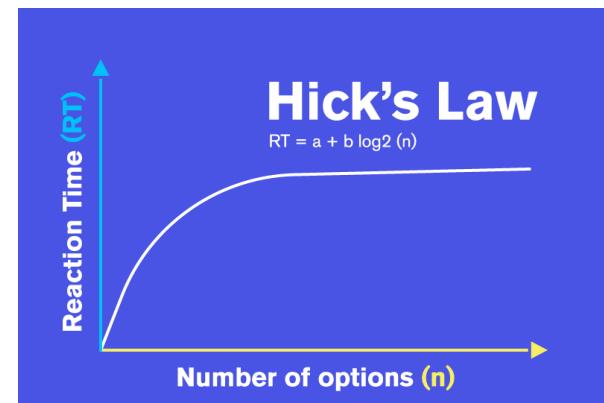
■ K.I.S.S Principle — Keep it Simple and Straightforward

■ 一个人拥有的选择越多，他们做决定的时间就越长。

■ $RT = a + b \log_2(n + 1)$

- T 是决策时间。
- a, b 是一个常数，代表个体的处理速度。
- n 是选择的数量。

■ 为用户提供更少、更集中的选项，
帮助他们快速做出决定，而不要让
他们陷入思维堵塞



选出你想购买的披萨



Hick's Law

- 应用：移除、减少、重复使用，以避免给用户带来压力！
 - 使用分析工具来找出最常用的功能。删除或重新设计那些很少使用的功能。

