人机交互期末重点

单选 + 多选（36分） 简答（30分） 设计（34分） 50%-60%

设计题

- **击键层次模型**
- **层次化任务分析**
- **实验设计**
- 交互原则

一章两个问题左右

概述

什么是人机交互 英文全称

人机交互（HCI）

- 是一门致力于**设计**、**评估**和**实现供人类使用**的**交互式**计算系统，并研究围绕这些系统的主要现象的学科
- Human Computer Interaction
- 重要性
    - 市场角度
        - 用户期望简单易用的系统
        - 对设计低劣系统的容忍度越来越差
    - 企业角度
        - 提高员工的生产效率
        - 降低产品的开发成本
        - 降低产品的后续支持成本
    - 用户角度
        - 获得较高的主观满意度
        - 减少时间、金钱、生命损失

人机交互 用户体验等等什么关系

用户体验 UX

- User Experience
- 用户体验**不仅仅是**指产品本身（比如软件好不好用、界面好不好看），而是包含了用户与企业打交道的**全流程**（包括售后服务、品牌印象、购买过程等）。这是一个宏观的、整体的概念
    - 用户体验是人与生俱来的感受能力
    - 不能够设计用户体验 只能为用户体验而设计

交互设计 IxD

- Interaction Design
- 是用户体验的一部分

人机交互历史

历史分为几个阶段 不同界面形式 优缺点

人机交互的各个阶段：

- 批处理阶段
    - 编写程序使用以“0|1”串表示的机器语言
    - 缺点
        - 每次只能由一个用户对计算机进行操作
        - 不符合习惯，耗时易错
- 联机终端阶段
    - 一维界面
    - 缺点
        - 回车后不能对命令内容进行修改
        - 需要用户大量记忆命令名称
- GUI阶段(Graphical User Interface, 当前的交互时代)
    - WIMP界面（windows icons menus pointers 窗口 图标 菜单 指针）
    - 优点
        - 用户可在窗口内选取任意交互位置，且不同窗口之间能够叠加
    - 二维半页面（二维指屏幕是平的 半指窗口可重叠）
    - 主要特征
        - **直接操纵**
- 未来人机交互 多媒体阶段
    - 多媒体画面
        - 引入动画、音视频等动态媒体
        - 二维半->三维 或更高
    - 多通道交互技术
        - 具有并行性 可同时接受来自多个通道的信息
    - 虚拟现实 语音交互 脑机交互
    - 主要风格 没有命令的用户界面
        - 由更多的媒体形式来构成更高的信息维度
        - 交互也将高度便携和个性化

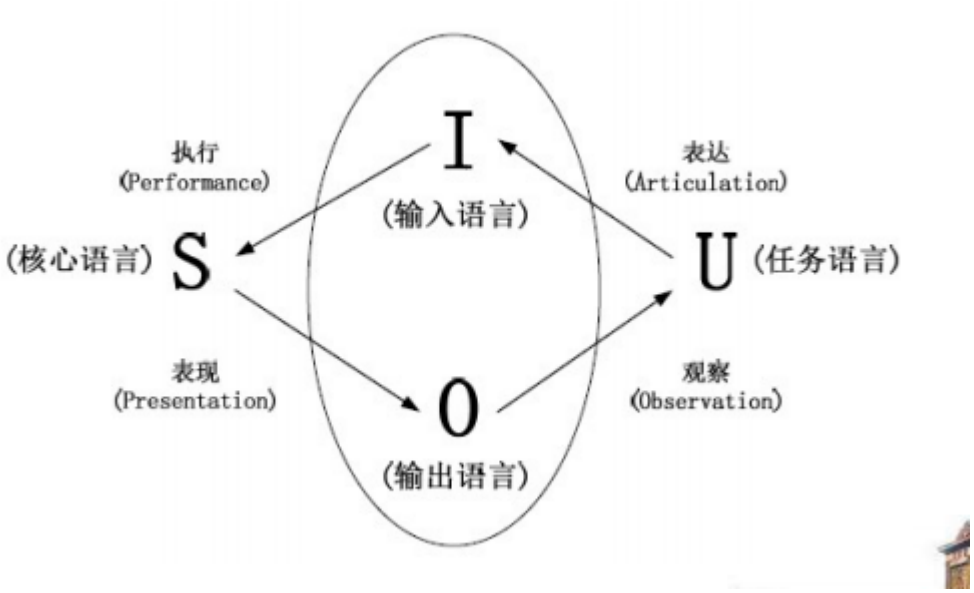
交互设计的原则和目标

重点！

交互框架 回答为什么不好用

- 作用
    - 提供理解或定义某种事物的一种结构
    - 能够帮助人们结构化设计过程
    - 认识设计过程中的主要问题
    - 有助于定义问题所涉及的领域
- EEC(最有影响力的框架)
    - 2个隔阂：执行隔阂
        - 用户为达目标而制定的动作与系统允许的动作之前的差别
    - 评估隔阂
        - 系统装填的实际表现与用户预期之间的差别
    - 4个组成部分
        - 目标（不等同于意图 单个目标可对应多个意图）
        - 执行
        - 客观因素
        - 评估
    - 7个阶段
        - 形成目标
        - 形成意图
        - 明确动作
        - 执行动作
        - 感知系统状态
        - 解释系统状态
        - 评估输出
    - 解释为什么有些界面的使用存在问题
- 扩展EEC模型
    - 解决了EEC模型不能描述人与系统通过界面进行的通信
    - 四个构成部分
        - 系统(内核语言) 指计算机内部的逻辑、状态和功能
        - 用户(任务语言) 指用户想要完成的目标和心理模型
        - 输入(输入语言) 指用户操作界面的方式
        - 输出(输出语言) 指系统向用户展示反馈的方式
    - 四个步骤
        - 表达 用户（U）将自己的意图转化为输入指令（I）
        - 执行 输入指令（I）被系统（S）接收并处理
        - 表现 系统（S）处理完后，将结果通过输出界面（O）呈现出来

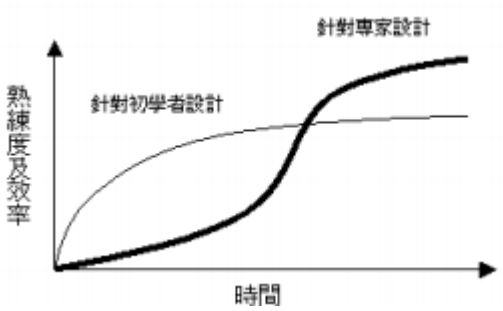
- 观察 用户（U）查看输出结果（O），评估任务是否完成



什么是可用性（客观）

可用性目标 目的是为交互设计人员提供一个评估交互式产品和用户体验各方面的具体方法

- 易学性(最基本的可用性属性)：使用系统的难易
- 易记性：学会使用并记住该产品如何使用的难易程度
    - 影响因素
        - 意义
        - 位置
        - 分组
        - 惯例
        - 冗余
    - 启发
        - 良好组织，使用用户已有的经验帮助提高易记性
- 效用性：一定程度上该产品提供了正确的功能，可以让用户做他们需要做的或想做的事情
- 高效率：产品在对用户执行任务的支持程度
    - 当用户学会使用产品之后，用户应该具有更高的生产力水平（效率）
    - 效率指熟练用户到达学习曲线上平坦阶段时的稳定绩效水平
- 安全性：避免用户发生危险和陷入不好的情形
    - 建议
        - 减少按键/按钮被误启动的风险 比如把save和close按键放置位置隔的远一点
        - 为用户提供各种出错时的恢复方法



用户体验是主观的 可用性是客观的 两者具有矛盾性

认识和理解可用性和其他用户体验目标之间的关系是交互设计的核心

简易可用性工程 特别是边做边说 启发式评估

- 用户和任务观察：了解和直接接触产品的目标用户
- 场景
    - 简便易行的原型 通过省略整个系统的若干部分来减少实现的复杂性
    - 水平原型 减少功能深度获取界面的表层
    - 垂直原型 减少功能的深度而对所选功能进行完整的实现
- 边做边说 **最有价值的单个可用性工程方法**
    - 让真实用户在使用系统执行一组特定任务的时候，讲出他们的所思所想
    - 可了解用户为什么这样做，并确定其可能对系统产生的误解
    - 实验人员需要不断地提示用户，或请他们事先观摩
- 启发式评估
    - 能够发现许多可用性问题 剩下的可以通过简化的边做边说来发现
    - 为避免个人的偏见，应当让多个不同的人来进行经验性评估
    - 5名专家能够发现约80%的可用性问题 被认为是最恰当的可用性测试用户数量
    - 建议将测试分阶段进行

设计原则 重点是十项启发式规则（可能用原则去解释 可能选择题选择某某场景对应那些原则）

- 基本原则
    - 可学习性 新用户能用它开始有效的交互并能获得最大的性能
    - 灵活性 用户和系统能以多种方式交换信息
    - 健壮性 在决定成就和目标评估方面对用户提供的支持程度
- 8条黄金原则：
    1. 尽可能一致：序列、术语、颜色、布局、字体（ 对应十项第4 ）
    2. 符合普遍可用性：不同用户不同需求
    3. 提供信息丰富的反馈：不同操作不同的反馈丰富程度
    4. 设计说明对话以生成结束信息：让用户知道什么时候他们已经完成了任务 使用户产生完成任务的满足感和轻松感 有助于让用户放弃临时的计划和想法（ 十项1 ）

5. 预防并处理错误：错误预防(将不适当的菜单选项功能以灰色显示屏蔽)、错误处理(提供简单的、有建设性的、具体的指导来帮助用户恢复操作)（十项5）
6. 让操作容易撤销：减轻焦虑 鼓励用户尝试新的选项（十项8）
7. 支持内部控制点：鼓励用户成为行为的主动者而不是响应者 避免模态对话框 避免很长引导、提供出口（十项3）
8. 减轻短时记忆负担：界面简单 风格统一 减少窗口间移动 提供足够学习时间 提供在线帮助信息（十项6）

- 10条启发式设计原则(熟练掌握)：Nielsen
  1. **系统状态可见度**
  - 对于用时较长的操作,需要给出显式的反馈
  2. **系统和现实世界的吻合**
  - 界面上的语言要使用用户熟悉的词汇、短语和概念，而不是内部术语
  - 遵循现实世界的惯例，使信息以自然和逻辑的顺序出现
  3. **用户享有控制权和自主权**
     - 提供“撤销”和“重做”功能，允许用户退出
  4. **一致性与标准化**
     - 产品内部对同一功能使用的术语和形式一致
     - 使用没有歧义的图标或图片
     - 一致的颜色、布局、大小写、字体等
  5. **避免出错**
     - 帮助用户减少出错可能
  6. **依赖识别而非记忆**
     - 减少用户的记忆负担，让选项可见。
  7. **使用的灵活性和高效性**
     - 充分考虑不同类型用户的使用偏好
     - 为专家用户提供快捷键或加速器。
  8. **帮助用户识别、诊断和恢复错误**
     - 错误信息要用通俗易懂的话，并提供解决方案。
  9. **帮助与文档**
     - 必须提供帮助/手册/用户指南
     - 用户指南的语言和格式应使用简单、标准的术语、
  10. **审美感和最小化设计**
      - 对话框中不应包含过多无关信息
      - 使用标准且通用的控件（如滑块、按钮等）
      - 选择适合屏幕显示的字体和字号，以确保最佳可读性

评估

可考的多 一定有一条大题关于实验设计

评估范型 四大类（场合 特点）

观察 询问 用户测试

可以用实证研究的方法 或者评估框架角度来说明实验如何开展

评估是设计过程的组成部分，系统化的数据搜集过程（并非独立的一环）

评估原则

- **评估应该依赖于产品的用户**
- **评估与设计应结合进行**
    - 仅靠用户最后对产品的一两次评估，是不能全面反映出软件可用性
- **评估应在用户的实际工作任务和操作环境下进行**
    - 根据用户完成任务的结果，进行客观的分析和评估
- **要选择有广泛代表性的用户**

评估范型

- **快速评估**
    - 设计人员非正式地向用户或顾问了解反馈信息，以证实设计构思是否符合用户需要
    - 强调“快速了解”，而非仔细记录研究发现
    - 得到的数据通常是非正式、叙述性的
        - 可以口语、书面笔记、草图、场景的形式反馈到设计过程
    - 是设计网站时常用的方法
    - 基本特征：快速
- **可用性测试**
    - 评测典型用户执行典型任务时的情况
        - 包括用户出错次数、完成任务的时间等
    - 基本特征
        - 是在评估人员的密切控制之下实行的
    - 主要任务
        - 量化表示用户的执行情况
    - 缺点
        - 测试用户的数量通常较少
        - 不适合进行细致的统计分析
- **实地研究**
    - 基本特征
        - 在自然工作环境中进行
    - 目的
        - 理解用户的实际工作情形以及技术对他们的影响

    - 作用
        - 探索新技术的应用契机
        - 确定产品的需求
        - 促进技术的引入
        - 评估技术的应用
    - 重难点
        - 如何不对受试者造成影响
        - 控制权在用户，很难预测即将发生和出现的情况
- **预测性评估**
    - 研究人员通过想象或对界面的使用过程进行建模
        - 专家们根据自己对典型用户的了解预测可用性问题的可用性评估
        - 逐步通过场景或基于问题回答的走查法
        - 用于比较相同应用不同界面的原型法，如使用Fitts定律预测使用设备定位目标的时间
    - 基本特征
        - 用户可以不在场
        - 整个过程快速、成本较低

区分评估技术的因素

- 评估在周期中的位置
    - 设计早期阶段的评估更快速、便宜
- 评估的形式
    - 实验室环境or工作环境
- 技术的主客观程度
    - 技术越客观，受评估人员知识的影响越大，如认知走查等
- 测量的类型
    - 主观技术：定性数据
    - 客观技术：定量数据
- 提供的信息
    - 底层信息：这个图标是可理解的吗？
    - 高层信息：这个系统是可用的吗？
- 响应的及时性
    - 边做边说法可及时记录用户行为
    - 任务后的走查取决于对事件的回忆
- 干扰程度
    - 直接观察可能会影响用户表现
- 所需资源
    - 设备、时间、资金、参与者、评估人员的专业技术及环境等

评估方法组合

- 启发式评估+边做边说等用户测试技术
    - 专家可通过启发性评估排除显而易见的可用性问题
    - 重新设计后，经用户测试，反复检查设计的效果
- 访谈+问卷调查
    - 先对小部分用户进行访谈，确定问卷中的具体问题
- 启发式评估vs.用户测试
    - 前者不需要用户参与
    - 二者发现的可用性问题不同，可以互补

实证研究方法

研究假设

- 零假设（不同的实验条件不会产生差异）
- 备择假设（往往是一个与零假设相反的陈述）

实验目标：是找到统计学证据来反驳或否定零假设，以支持备择假设

自变量Independent Variables：研究者感兴趣的设计因素或因变量变化的可能原因

- 自变量与受试者的行为无关，参与者无法对自变量施加任何影响
- 至少需要 2 个不同的取值， 这些取值被称为**实验条件**test conditions
- 如人的特性是天然的自变量。 如不同类型的技术，设备，设计

Factor (IV)	Levels (test conditions)
Device	mouse, trackball, joystick
Feedback mode	audio, tactile, none
Task	pointing, dragging
Visualization	2D, 3D, animated
Search interface	Google, custom

因变量 Dependent Variables 指研究者感兴趣的结果或效果

- “因”用于说明该变量依赖于受试者的行为或自变量的变化
- 因变量必须被明确定义
- 研究者希望找出自变量的变化是否会引起因变量的变化，以及如何引起因变量的变化
- 研究必须是可复现的

自变量通常是研究人员可以控制的实验条件；因变量通常是研究者需要测量的实验结果

实验构成

- 实验条件 变量取值

- 实验单位 应用实验条件的对象
- 分配方式 将实验单位分配到不同实验条件的方式
    - 完全随机化

实验设计

真正的实验

- 以至少一个可检验的研究假设为基础，并旨在验证它
- 通常**至少有两种条件**(实验条件和对照条件)或组(实验组和对照组)
- 因变量通常使用定量测量
- 通过各种**统计显著性检验**对结果进行分析
- 以消除潜在偏差为目标来设计和进行
- 具备不同的参与者样本，在不同的时间，不同的地点，由不同的参与者进行**复现**

组间设计 Between-group design (between-subject design)

- 每个参与者只暴露在一种实验条件下
- 参与组的数量直接对应于实验条件的数量
- 优点
    - 设计更简洁
    - 避免了学习效应
- 缺点
    - 结果受个体差异影响大
    - 样本量大

组内设计 Within-group design

- 优点
    - 样本量小
    - 隔离了个体差异
- 缺点
    - 学习效果的影响
    - 疲劳问题

组间设计和组内设计的优缺点完全相反

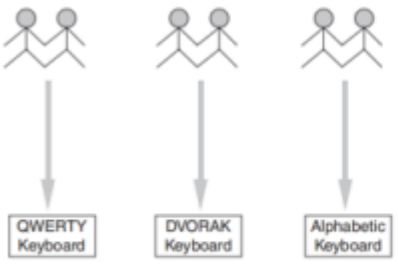


图 3.3 组间设计

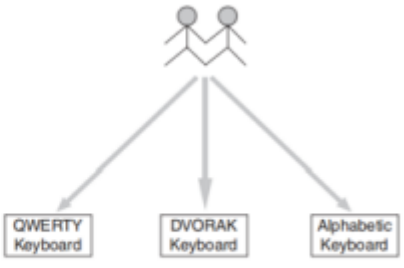


图 3.4 组内设计

如何选择

- 组间设计
    - 任务简单，个体差异有限
    - 受学习效果影响较大的任务
        - 有的实验不可能使用组内设计
            - 例子：在网上商店中，新手和有经验的用户在找到物品所需的时间上没有差别 一个人不可能同时是新手和有经验的在线商店用户
    - 注意
        - 参与者应尽可能**随机**分配到不同的条件下
        - 需要在不同条件下尽量平衡潜在的混杂因素，**即每个条件下的群体应尽量相似**
- 组内设计
    - 个体差异较大、学习效果不太容易受到影响的任务、或目标参与者群体很小的任务
        - 大多数测试复杂或学习技能或知识的任务——比如打字、阅读、写作和解决问题
    - 需要考虑如何控制与组内设计相关的学习效果、疲劳和其他潜在问题的负面影响
        - 控制学习效果 ：将实验条件的**顺序随机化。** 当研究的目标不是与应用的初始交互时，减少学习效果影响的有效方法是**提供充足的培训**
        - 解决疲劳问题 ： 单个实验的适当长度应该是60 - 90分钟或更短，并提供必要的休息机会

析因设计Factorial Design

	QWERTY	DVORAK	Alphabetic
Composition	1	2	3
Transcription	4	5	6

分析数据时，比较同一行可以检查不同键盘的影响；任务效果可以通过比较同一列的数据来检验

可以组间 可以组内

裂区设计split-plot design

析因研究中的一种设计，既有组间成分，也有组内成分

纵向比较：同一列的两个条件（如 1 和 4）由**同一批受试者**完成（组内对比）。

横向比较：同一行的三个条件（如 1、2、3）由**三批完全不同的人**完成（组间对比）

优点：允许在一个实验中研究两个或两个以上自变量之间的相互作用的影响

相互作用可被描述为“一个自变量对因变量的不同影响，取决于另一个自变量的特定取值

观察用户

观察法是所有可用性方法中最简单的方法

观察适用于产品开发的任何阶段

- 初期：理解用户的需要
- 开发过程：检查原型
- 后期：对最终产品进行评价

直接观察

- 实验室观察
- 现场观察

间接观察

- 日志和交互记录

观察的注意事项

观察人员自始至终应尽量保持安静 ，让用户感觉不到观察人员的存在 ，保证用户操作和平时工作的状态一样

当用户的操作令观察人员无法理解时 ，需要打断用户，请他对所做的某些操作进行解释 ，或把用户莫名其妙的操作行为记录下来

观察初期，应该拒绝用户的任何帮助请求 希望观察用户在没有系统专家指点的情况下如何操作，待评估完成后为用户提供适当帮助

记录观察数据，观察与访谈相结合。然后进行数据的定性定量分析。

询问用户和专家

访谈 有目的的对话过程

- 指导原则
    - 避免过长的问题
    - 避免使用复合句
    - 避免使用可能让用户感觉尴尬的术语或他们无法理解的语言
    - 避免使用有诱导性的问题
    - 尽可能保证问题是中性的

类型：

- 非结构化访谈：问题开放，不限定内容格式
- 结构化：根据预先的问题
- 半结构化
- 集体访谈：个别成员的看法是在应用的上下文中通过其他用户的交流而形成的。**焦点小组**是集体访谈的一种形式

焦点小组：

- 非正式
- 在界面设计之前和经过一段使用之后评估用户的需要和感受
- 人数限制：由大约6到9个典型用户组成
- 如在评估大学的网站时，可考虑由行政人员、教师和学生们组成3个分别的焦点小组

问卷调查（询问用户）

问卷调查是用于搜集统计数据和用户意见的常用方法

问卷设计原则：

- 确保问题具体明确
- 避免使用复杂的多重问题
- 等等一些三观原则

问卷问题类型：

- 常规问题 年龄啥的
- 自由回答 给个建议
- 量化分级问题
    - 之前的 Likert尺度 和 语义差异度尺度
    - 多选

问卷调查或访谈都属于间接方法，不对用户界面本身进行研究，而只是研究用户对界面的看法

认知走查（询问专家）

逐步检查使用系统执行任务的过程，从中找出可用性问题

主要目标是确定使一个系统如何易于学习

评估的具体过程就是把用户在完成这个功能时所做的所有动作讲述成一个 令人可以信服的故事

步骤：

- 标识用户特性，心理，知识经验啥的
- **设计样本任务**
- **制作原型或界面描述，明确执行步骤**（不一定要可运行的原型）
- 专家分析
- 逐步检查任务操作步骤
- 汇总关键信息
- 修改更正

优点：

- 不需要用户参与
- 不需要可运行的原型
- 能找出非常具体的用户问题

缺点：

- 工作量大，费时
- 关注面有限，**只适合评估一个产品的易学性**，不太容易发现使用效率方面的可用性问题协作走查：由用户、开发人员和可用性专家合作，逐步检查任务场景，讨论

启发式评估（询问专家）

评估步骤：

- 彻底检查界面
- 将界面与启发式规则对比
- 列举可用性问题
- 对每个问题解释确认

优点：

- 不涉及用户 面临的实际限制和道德问题较少
- 成本低，不需要特殊设备，而且较为快捷，又被称为”经济评估法“

缺点：

- 评估人员需要经过长时间的训练才能成为专家
- 可能出现虚假警报

用户测试

基于DECIDE框架的评估

步骤：

1. 决定评估需要完成的总体目标
2. 发掘需要回答的具体问题
3. 选择用于回答具体问题的评估范型和技术
4. 标识必须解决的实际问题，如测试用户的选择
5. 决定如何处理有关道德的问题
6. 评估解释并表示数据

用户测试

在受控环境中（类似于实验室环境）测量典型用户执行典型任务的情况，以DECIDE框架为基础。

为什么要进行显著性检验，显著性检验可以帮助确定我们可以将从样本群体中观察到的结果推广到整个

群体的把握有多大。如方差分析。

需求

重点是以下两条技术

用户建模

人物角色

- 不是真实的人
- 是基于观察到的那些真实人的行为和动机，并且在整个设计过程中代表真实的人
- 是在人口统计学调查收集到的实际用户的行为数据的基础上形成的综合原型
- 作用
    - 解决产品开发过程中出现的3个设计问题
        - 弹性用户
            - 为弹性用户设计赋予了开发者根据自己的意愿编码，而仍然能够为"用户"服务的许可
            - 如设计医院产品时，考虑设计能够满足所有护士的产品（用户定义的太宽泛 反而满足不了任何一种用户）
        - 自参考设计
            - 设计者或者程序员将其自己的目标、动机、技巧及心智模型投射到产品的设计中
        - 边缘情况设计
            - 必须考虑边缘情况，但它们又不应该成为设计的关注点
            - 问：A会经常进行这样的操作嘛（不能为了边缘情况而让本来的核心功能做得差强人意）

人物角色的构造基于以下问题

- 谁将使用系统
- 这些用户属于哪些类型的人群
- 是什么因素决定他们将怎样使用系统
- 他们与软件的关系有什么特征
- 他们通常需要软件提供什么支持
- 他们对软件会有怎样的行？他们对软件的行为有什么期望

留心焦点角色 最常见、最典型的角色

场景

需求的获取方式之一 场景

- 表示任务和工作结构的"非正式的叙述性描述"
- 以叙述的方式描述人的行为或任务，从中发掘出任务的上下文环境、用户的需要、需求
- 形式可以类似一篇故事、一个小品或者在给定环境下按照时间顺序的一段情节
- "讲故事"是人们解释自己做什么或者希望执行谬个任务的最自然方式
    - 故事的焦点就是用户希望达到的目标
    - 若场景说明不断提到某个特定形式、行为或者地点，就表明它是该活动的核心
- 场景说明通常来自专题讨论或者访谈，目的是解释或讨论有关用户目标的一些问题

层次化任务分析 HTA Hierarchical Task Analysis

把任务分解为若干子任务，再把子任务进一步分解为更细致的子任务。之后，把他们组织成一个“执行次序”，说明在实际情形下如何执行各项任务

执行次序

■0.借书

- ●1.前往图书馆
- ●2.检索需要的图书
  - ●2.1 访问图书馆目录
  - ●2.2 使用检索屏
  - ●2.3 输入检索准则
  - ●2.4 找出需要的图书
  - ●2.5 记录图书位置
- ●3.找到书架并取书
- ●4.到柜台办理借阅手续

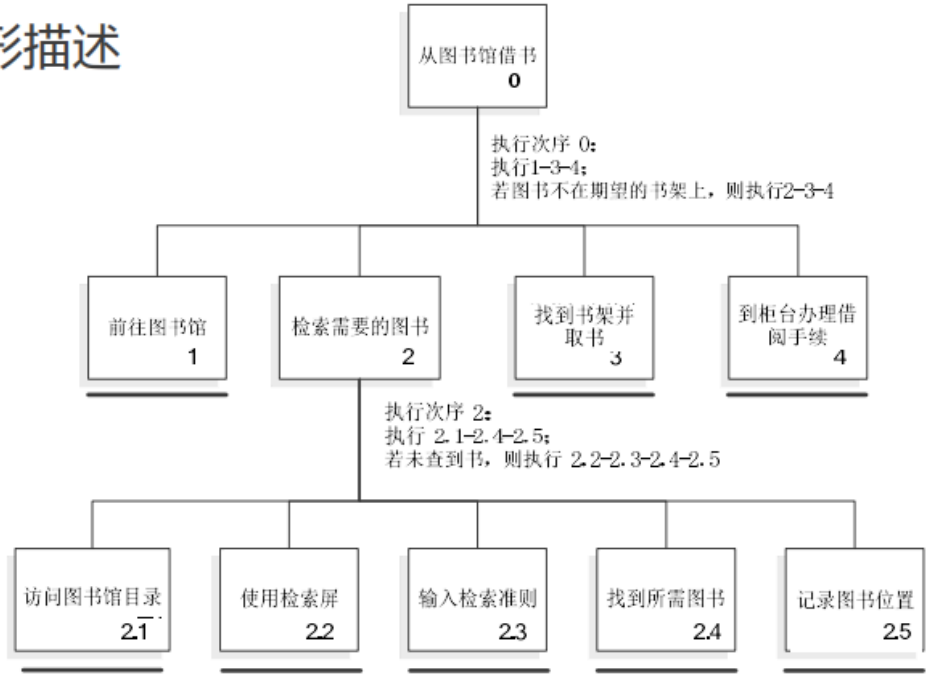
执行次序0：执行1-3-4；若图书不在期望的书架上，则执行2-3-4。

执行次序2：执行2.1-2.4-2.5；若未查到此书，则执行2.2-2.3-2.4-2.5

编号说明了对应步骤编号。

方框-线条图示

■HTA的图形描述



交互式系统的设计

设计框架（笔试不大容易考）

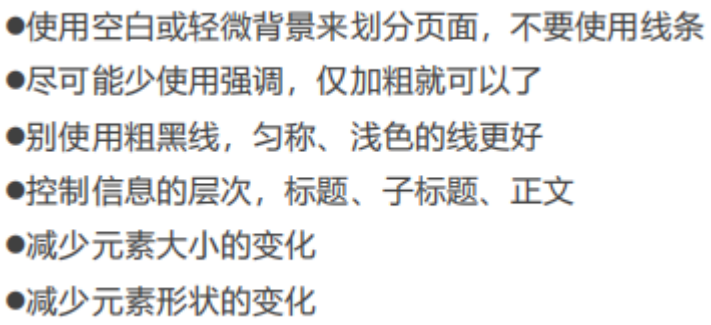
软件设计中的考虑

- 积极的文字表达 通过友善、鼓励性和非指责性的语言，提升用户的心理舒适度，即便是面对“低分”或“错误”的情况
- 消除歧义
- 让软件友好和体贴
    - 有趣的用语
- 加快系统的相应时间
- 减轻用户的记忆负担
- 减少用户的等待感
    - 以某种形式的反馈让用户了解操作进行的进度和状态
    - 以渐进方式向用户呈现处理结果
    - 给用户分配任务，分散用户的注意力
    - 减低用户的期望值
- 设计好的出错信息
    - 使用清晰的语言来表达，而不要使用难懂的代码

- 使用的语言应当精炼准确，而不是空泛而模糊的
- 对用户解决问题提供建设性的帮助
- 出错信息应当友好，不要威胁或责备用户

简化设计的策略 重点

- **删除**
  - 最明显的简化设计方法
  - 删除杂乱的特性
    - 可以让设计师专注于把有限的重要问题解决好
    - 有助于用户心无旁骛地完成自己的目标
    - 避免得到由简单功能叠加起来的毫无特色的产品
    - 保证只交付那些真正有价值的功能和内容
  - 如何删除
    - 关注核心
      - 与新增功能相比，客户更关注基本功能的改进
      - 影响到用户日常使用体验的功能
    - 砍掉残缺功能
      - 沉没成本误区
      - 为什么要留着它而非为什么应该去掉它
    - 目标用户经常会遇到这个问题吗？
    - 不要简单地因为客户要求就增加功能
  - 几种删除方法
    - 删除错误
      - 避免用户犯错
    - 删除视觉混乱
      - 减少用户必须处理的信息，集中注意力在真正重要的内容上



    - 删减文字
      - 删除不必要内容可以让读者对自己看到的内容更有自信
        - 删除引见性文字
        - 删除不必要的说明

        - 删除繁琐的解释
        - 使用描述性链接（如点击这里）
    - 精简句子
        - 让文字变得更加简洁、清晰、有说服力

- **组织**
    - 最快捷的简化设计方式
    - 分块
        - 用户界面设计离不开分块
        - 7±2法则 人类短期记忆（工作记忆）能够同时处理的信息块（Chunks）数量通常在 5 到 9 个之间，平均值为 7
        - 名词：可以按字母表、时间或空间顺序排列的清单
        - 动词：围绕行为进行组织
    - 确定清晰的分类标准

    - 利用不可见的网格来对齐界面元素
    - 大小和位置
        - 重要的元素要大一些，不太重要的界面元素应该小一些
            - 规则：如果一个元素的重要性为1/2，那就把它的大小做成1/4
        - 把相似元素放在一起
    - 期望路径
        - 在描述用户使用软件的路径时，千万不要被自己规划图中清晰的线条和整洁的布局所迷惑

- **隐藏**
    - 低成本的简化方案
        - 用户不会因不常用的功能分散注意力

        - 可作为删除不必要功能的开始
        - 必须仔细权衡要隐藏哪些功能
    - 隐藏什么
        - 主流用户很少使用，但自身需要更新的功能
        - 事关细节（对服务器进行配置或设计邮件签名）
        - 选项和偏好（修改绘图应用的单位）
        - 特定于地区的信息（如时间和日期需频繁自动更新的信息）
    - 渐进展示
        - 隐藏精确的控制部件
            - 常用的功能放出来，一些比较精确的专家级的功能可以隐藏起来
            - 自动保存用户的选择比自定义的效果好
            - 对于用户期望的功能，要在正确的环境下给出明确的提示
    - 适时出现
        - 过分强调隐藏功能（如为每个词加上超链接的选项）会导致混乱
        - 成功的隐藏
            - 尽可能彻底地隐藏所有需要隐藏的功能
            - **在合适的时机、合适的位置上显示相应功能**
    - 别让用户找信息
    - 让功能易于发现
        - 为隐藏功能打上标签：更多，高级
        - **把标签放在哪里比把标签做多大重要得多**
- **转移**
    - 被精简掉的按钮全部通过电视屏幕上的菜单来管理
    - 设备之间转移
        - 有时候，把某项任务的某些内容（如输入信息）转移到不同的平台上可能是一种更好的选择

■RunKeeper应用的例子

●功能：记录用户跑步的路线

●收集记录跑步数据很简单

●但小屏幕难以显示所记录的与一次跑步有关的所有信息

●RunKeeper网站适合输入数据，且能够很容易地查看各种细节信息

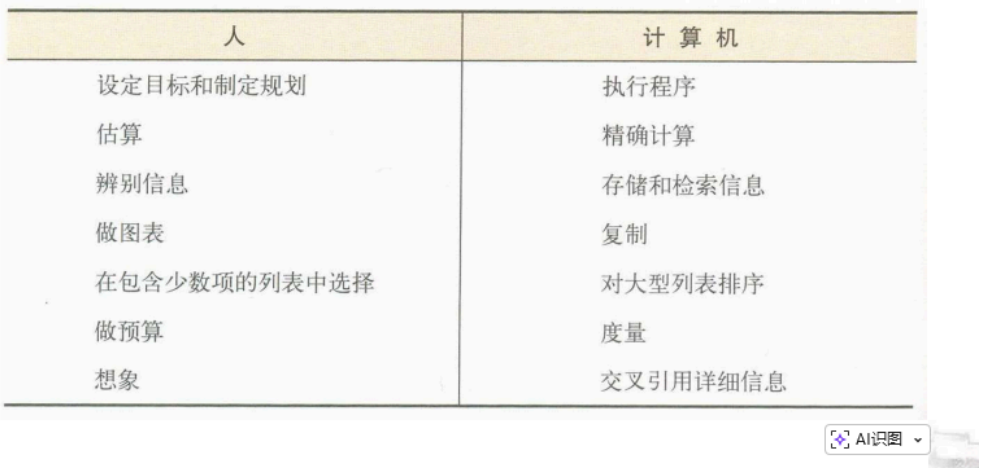
■利用两个平台的优势，各司其职

●按时间段手机信息：手机

●查看相应时间：网站

        - 向用户转移

- 把控制权交给用户 不要给用户的选择太多评判
- 搞清楚把什么工作交给计算机，把什么工作留给用户



人	计 算 机
设定目标和制定规划	执行程序
估算	精确计算
辨别信息	存储和检索信息
做图表	复制
在包含少数项的列表中选择	对大型列表排序
做预算	度量
想象	交叉引用详细信息

删除不必要的

组织要提供的

隐藏非核心的

转移……?

人机交互基础知识

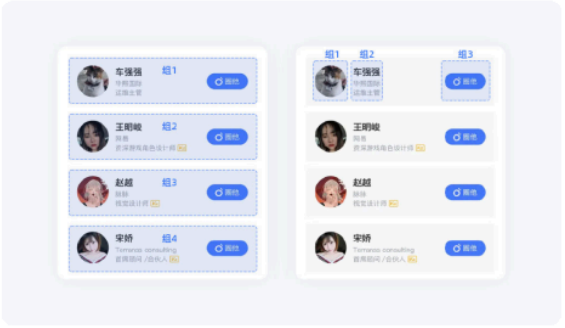
人类处理机模型 大头娃娃模型（重点）

- 最著名的信息处理模型
- 三个交互式组件
    - 感知处理器
        - 信息将被输出到声音存储和视觉存储区域
    - 认知处理器
        - 输入将输出到工作记忆
    - 动作处理器
        - 执行动作
- 存在的问题
    - 把认知过程描述为一系列处理步骤
    - 仅关注单个人和单个任务的执行过程
        - 忽视了复杂操作执行中人与人之间及任务与任务之间的互动
    - 忽视了环境和其他人可能带来的影响
- 改进：外部认知模型 分布式认知模型

格式塔心理学

- 研究人是如何感知一个良好组织的模式的，而不是将其视为一系列相互独立的部分记忆
    - 事物的整体区别于部分的组合
- 表明用户在感知事物的时候总是尽可能将其视为一个“好”的型式
    - **相近性原则**
        - 空间上比较靠近的物体容易被视为整体
            - 设计界面时，应按照相关性对组件进行分组
        - 应用
            - 合理运用法则，让界面层次清晰有序

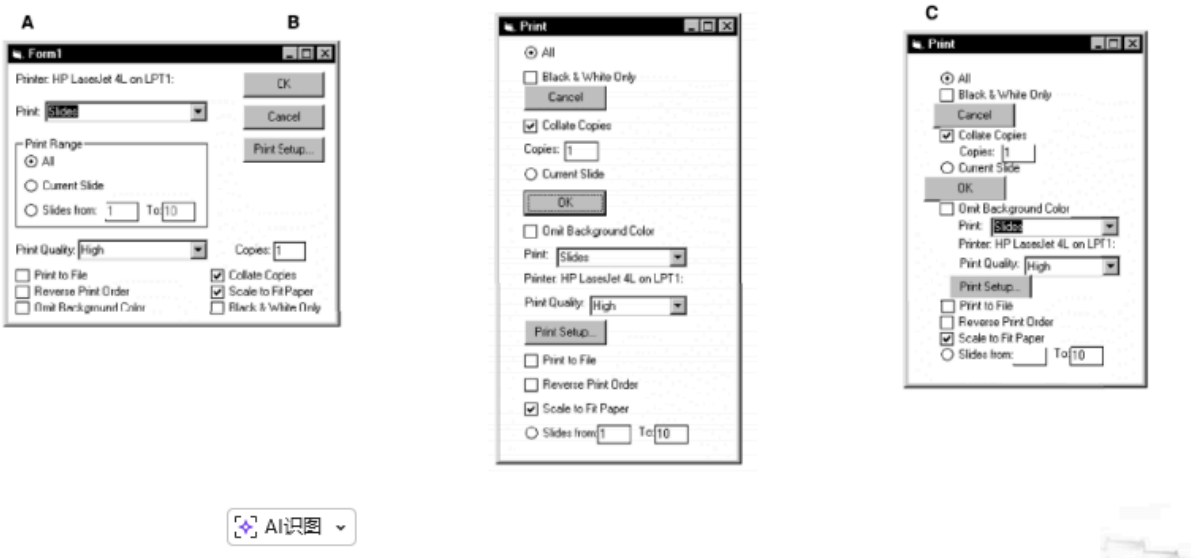
        ●例如列表页设计，将相关的信息组合在一起并重复排列出来，就能明显感知不同小组之间的界限，当同一小组内元素关系明确时，将其更加靠拢，用户视觉就会更聚焦。

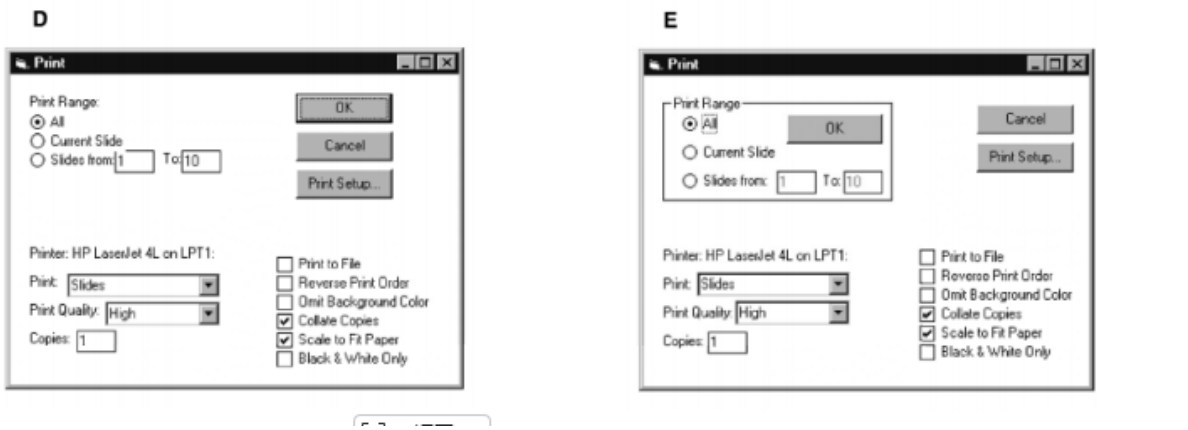
        

        
    - **相似性原则**
        - 人们习惯将看上去相似的物体看成一个整体
            - 功能相近的组件应该使用相同或相近的表现形式
        - 应用
            - 使用不同的大小、颜色、形状来创建对比或视觉权重，呈现出不一样的视觉效果，以达到弱化（降低视觉）或凸显（强化视觉）某些内容
    - **连续性原则**
        - 共线或具有相同方向的物体会被组合在一起
            - 将组件对齐，更有助于增强用户的主观感知效果
    - **对称性原则**
        - 相互对称且能够组合为有意义单元的物体会被组合在一起
    - **完整和闭合性原则**
        - 人们倾向于忽视轮廓的间隙而将其视作一个完整的整体
            - 页面上的空白可帮助实现分组
    - **前景与背景原则**
        - 前景和背景在某些情况下可以互换

    应用

    屏幕格式塔 Law of lines

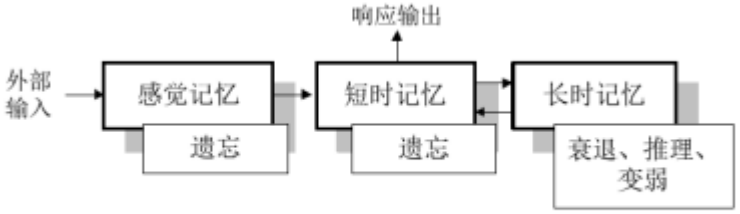




D更符合相近性和闭合性原则

人脑记忆特性

- 感觉记忆
  - 又称瞬时记忆
  - 在人脑中持续约 1s 钟
  - 帮助我们把相继出现的一组图片组合成一个连续的图像序列，产生动态的影像信息
- 短时记忆
  - 感觉记忆经编码后形成
  - 又称工作记忆，约保持 30s
  - 储存的是当前正在使用的信息，是信息系统加工的核心，可理解为计算机的内存
  - 短时记忆的存储能力约为7±2个单元
- 长时记忆
  - 短时记忆中的信息经进一步加工后会变为长时记忆
  - 只有与长时记忆区的信息具有某种联系的新信息才能够进入长时记忆
  - 长时记忆的信息容量几乎是无限的，但会**遗忘**，是指长时记忆中的信息有时是无法提取，不代表长时记忆区的信息丢失了



交互模型与理论

两种模型

- **击键层次模型**必考 操作序列合理即可拿分 可以加上文字部分内容 方便助教看
- 菲兹定律

hick's low 应该不会考

击键层次模型

GOMS模型 Goal-Operator-Method-Selection 目标-操作-方法-选择规则

击键层次模型

- 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

K (Keystroke/Button)：按下单个按键。

- **熟练打字员**：0.22s（约 55 wpm）。
- **一般打字员**：0.28s（约 40 wpm）。
- **不熟悉键盘的人**：1.20s。
- **按下 Shift 或 Ctrl**：0.08s。

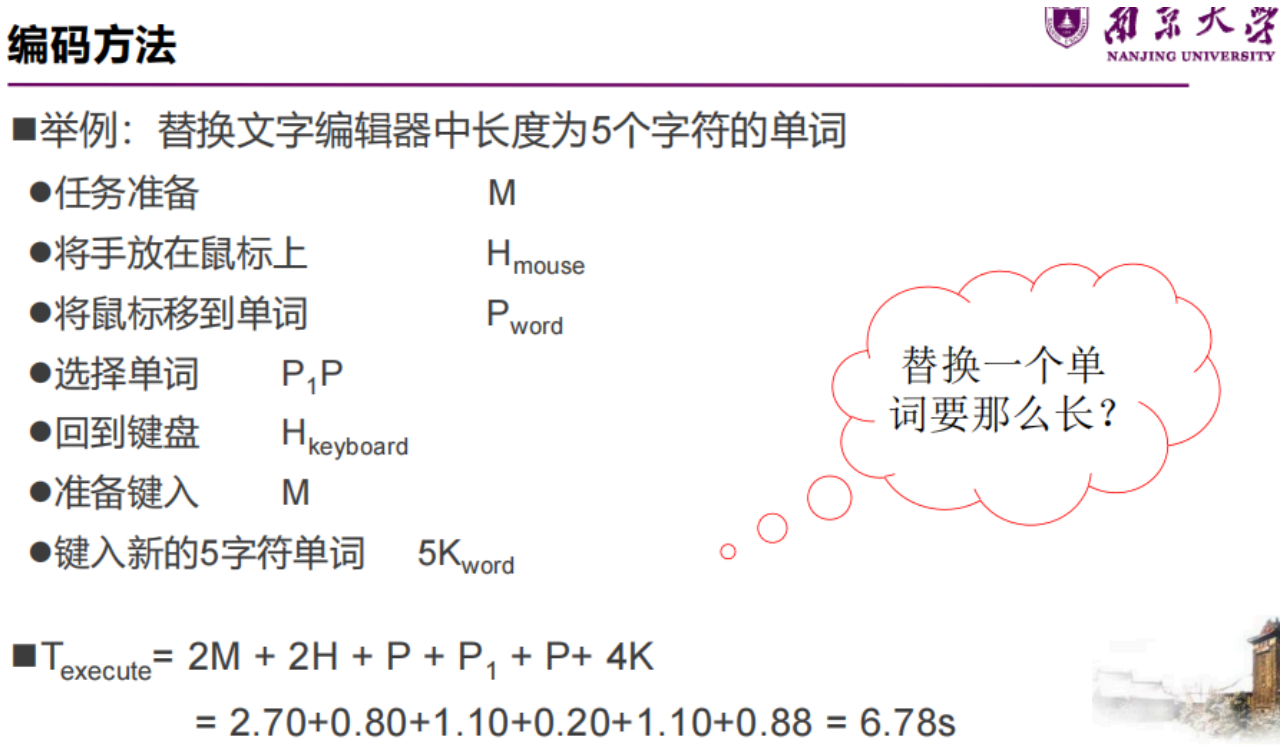
P (Pointing)：使用鼠标指向屏幕位置，耗时 **1.10s**。

P_1 (Clicking)：按下鼠标按钮（不移动），耗时 **0.20s**。

H (Homing)：手在键盘和鼠标之间切换，耗时 **0.40s**。

M (Mental)：执行动作前的心理准备（如做决定、搜索），耗时 **1.35s**。

R(t) (Response)：系统响应时间，仅当用户必须等待时才计入时间 **t**。



放置 M 操作符的五大规则解析

这些规则的逻辑是：**先假设每个关键动作前都有思考，然后再根据行为的连贯性剔除多余的部分。**

- **规则 0：初始插入**
  - 在所有的 **K（击键）** 操作符前插入 M。
  - 在所有的用于选择命令（非参数）的 **P（指向）** 操作符前插入 M。
- **规则 1：预测性剔除**
  - 如果某个操作（如击键 K）完全可以由之前的操作（如指向 P）预测到，则删除中间的 M。
  - *例子*：指向按钮后紧接着点击，常记为 $PMK \to PK$。
- **规则 2：认知单元合并**
  - 如果一串 MK 组合形成了一个熟练的“认知单元”（例如输入一个常见的命令单词），则除了第一个 M 外，删除后续所有的 M。

- **规则 3：冗余终结符剔除**
  - 如果 K 是一个紧跟在参数后面的冗余终结符（如输入完数字后的回车），则删除 K 之前的 M。
- **规则 4：变量与常量区分**
  - 如果是输入已知的常量字符串（命令名），删除其终结符前的 M。
  - 如果是输入变量字符串（如用户自定义的数字或文本），则**保留**终结符前的 M，因为用户需要确认输入是否正确。

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P_1)
$MPMP_1$ -> MPP_1

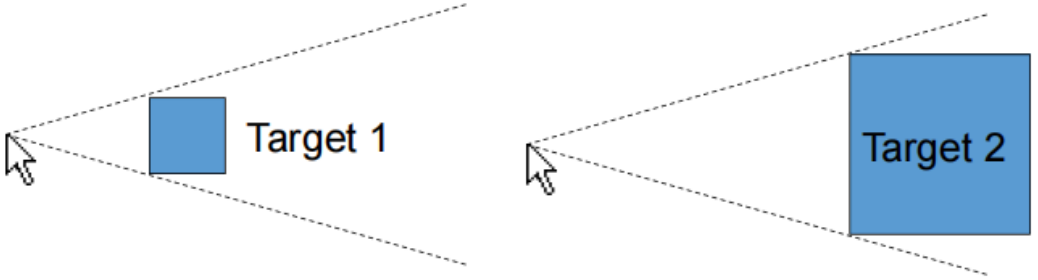
Fitts定律

- 能够预测使用某种定位设备指向某个目标的时间
- 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
- 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

MT平均时间 ID困难指数 IP性能指数 A运动距离或振幅 W目标宽度 一般用a=50,b=150

Fitts' Law

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



Same ID → Same Difficulty

定律的建议

- 大目标、小距离具有优势
- 屏幕元素应尽可能多地占据屏幕空间
- 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

应用

- 缩短当前位置到目标区域的距离
    - 如右键菜单技术
- 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（靠边 相当于W无限大）

Hick’s Law

RT = a + b log2 (n +1)

*RT*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

一个人拥有的选择越多，他们做决定的时间就越长

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

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

以用户为中心

以用户为中心 User-Centered Design (UCD)

- 以真实用户和用户目标作为产品开发的驱动力，而不仅仅是以技术为驱动力
- 应能充分利用人们的技能和判断力，应支持用户，而不是限制用户
- 需要透彻了解用户及用户的任务，并使用这些信息指导设计
- 原则
    - 及早以用户为中心
    - 综合设计
    - 及早并持续性地进行测试
    - 迭代设计
- 缺陷
    - 影响产品的创新性
    - 可操作性受到时间、预算和任务规模的限制

    - 忽视了人的主观能动性和对技术的适应能力

局限性体会一下即可