

《微交互》

图书基本信息

书名：《微交互》

13位ISBN编号：9787115331618

10位ISBN编号：7115331618

出版时间：2013-11-1

出版社：人民邮电出版社

作者：塞弗 (Dan Saffer)

页数：164

译者：李松峰

版权说明：本站所提供下载的PDF图书仅提供预览和简介以及在线试读，请支持正版图书。

更多资源请访问：www.tushu000.com

《微交互》

内容概要

平庸的产品与伟大的产品差就差在细节上。作者Dan Saffer将通过这本书展示怎么设计微交互，即位于功能之内或周边的那些交互细节。你的手机怎么静音？你怎么知道有新邮件了？怎么修改应用的设置？诸如此类的交互细节，既可以毁掉一个产品，也可以成就一个产品。高效而有趣的微交互，涉及触发器、规则、循环和模式，还有反馈。透过书中生动、真实的设备及应用示例，读者将理解微交互对于塑造产品个性、赋予产品卖点的重要意义。

《微交互》

作者简介

Dan Saffer 是 Smart Design 的交互设计总监，他还著有《交互设计指南》（ Designing for interaction: Creating Innovative Applications and Devices ）、《手势界面设计》（ Designing Gestural Interfaces ）和《设备设计》（ Designing Devices ）。自1995年起，他从事过家电、设备、应用、机器人、网站和服务设计，数百万用户每天都在使用他的设计。

书籍目录

- 第1章 设计微交互
- 第2章 触发器
- 第3章 规则
- 第4章 反馈
- 第5章 循环与模式
- 第6章 综合实例
- 附录 测试微交互

精彩短评

- 1、首先，作者用很大的篇章在强调微交互作用很大；其实，作者举了一堆他认为微交互做的很好的例子。觉得发现到周围产品微交互做很好并记住用处更大
- 2、以朴实的态度注重细节
- 3、简单易读，也很适合工程师看看
- 4、真的没什么接地气的内容，对于互联网交互设计没什么价值..
- 5、很好，能触发思考，看的时候不能反思自己想的太少
- 6、产品具备某种人格....人机啊...
- 7、不知道是不是翻译的问题，看书的过程脑子里总是冒出“又不说人话”的评价。挺简单一个意思偏要绕个圈子外加专业术语，看了几遍明白了一个肤浅的概念...比如设计成按钮就应该有按下的效果，说成“控件的视觉使用情境”
- 8、把这些其实大家都知道了的概念归纳成这么专业的说法也是需要能力的（细节很多时候是感性的，所以加强一些理性的思路也很有帮助），每部分最后的总结也很值得回顾。里面有很多案例还是会有争议，但是思考才是最重要的。
- 9、简洁易懂，有些章节也可以当为工具书用
- 10、内容很好，但是翻译一般。
- 11、UI和产品很有指导意义的一本书，触发器-规则-反馈-循环这个路径上随时查缺补漏
- 12、精读了一下。。。太值得阅读了，当今的移动应用大同小异，如果能在微交互这一块注入资源，定能打造更多用户体验非常棒的产品！
- 13、简单易读
- 14、细节决定成败。时不时翻下
- 15、不错的，干货比较多，虽然是一本小书但是受益匪浅！没有过多的概念和大话，非常贴合实际应用，作为用户体验入门感觉很受用。
- 16、将交互分成更小的部分，是一种新的看待交互的方式，学习了
- 17、内容简洁易懂，适合初学者阅读。若是从事几年业内工作的，上面的概念有些浅显，可用于巩固自己的观点。
- 18、非常不错的一本书
- 19、乱得没法看。
- 20、开始阅读
- 21、适合新人
- 22、不错
- 23、在读中虽然老了些，但是还是涨了一些姿势！画了些重点~
- 24、介绍了一些基础交互设计规则。
- 25、往小处想。
- 26、系统的讲了遍交互设计中的细节，验证了我平时工作中很多做法事对的
- 27、讲得蛮好的
- 28、翻译的质量堪忧啊
- 29、比较粗浅
- 30、内容似乎不错，但是翻译有误，语句不通，语义不明，感觉像翻译机器翻译的，个别地方还有错别字，还不如翻译机器呢。在亚马逊买的，是正版。
- 31、这么好的书评分这么低。交互入门经典，但是要建立在有过一点交互经验的基础上。
- 32、啃书中 从高度看问题 让我想起了高考复习刷题的认识 很久没想过这回事了
- 33、看到好几篇文章里推荐了这本书，确实不错。可以立刻带入到产品里，get了不少优化idea~
- 34、入门书，知识系统化。
- 35、几乎每个知识点，都能对应到工作中曾遇到的实例，感觉很不错。
- 36、不似预期 感觉一般~
- 37、最有价值结论：微交互（microinteraction）虽然可以通过 A/B test 改进，但它的原始创建过程，是不存在 MVP（最小可用版本）的，要搞就必须 all in。

《微交互》

- 38、有点用
- 39、确实有很多可以学习的地方.
- 40、是翻译的有点别扭么？
- 41、浅显易懂，要是后面的案例多配点图就完美了
- 42、还行吧，都是基本理论，早就掌握了。
- 43、总结的很好。

- 44、我读的有关交互的入门书，对交互有了一个简单的概括，列举了一些常见的场景和实例。
- 45、总结得很清楚简明。内容4星。
书中各观点理论详细标注的参考出处，严谨态度使得这本书的扩展阅读范围很广，增值很大。最终5星
- 46、触发器 规则 反馈 循环与模式
- 47、跳出了“搬定义”——“简单论述”——“举例”的类型书写作窠臼，细致、系统地阐述了微交互的历史、流程、法则、应用中的演进等方方面面，尤其是触发——规则——反馈——模式及循环这一模型的提出，堪称全书精华。缺点是后面的实例不够直观。
- 48、拼完口号拼细节...
- 49、逻辑性较强，有关微交互的场景设计。内容挺棒，翻译太烂，四星~
- 50、当工具书还可以，看了各种detail

《微交互》

精彩书评

- 1、定义了微交互的框架和体系，体系细分后，观点的论述乱78糟，缺乏逻辑，重复啰嗦，举例不贴切，细读不不下去，观大略就好.....
- 2、1.内容还没看，大概翻了一下，第一印象感觉看不下去，不说内容，排版和翻译就让人失望2.排版。虽然用了三四种字体依然感觉层次恒不清楚，字体黑和间隙大，显得文字很散漫，还有一些奇怪的图标去标记什么意义忘了3.翻译的也一般总之，先搁置一段时间等心情好了再读
- 3、更注重对细节的关注，这是读后此书最重要的感受。这里面举了很多实例，也教了很多方法，有很强的实用性，看完之后，还想再看第二遍。即使每个人都把关注细节挂在嘴边，但真正做到的人不多，真正沉下心去抠细节的人更少，我想这本书的作用就是让我们沉下心来，好好去思考，细节真正在哪里，而又带给我们什么，我们应该怎么做，如何把他做好。

章节试读

1、《微交互》的笔记-第99页

这一页对图4-11的说明也许是原作者的一个误解：

iPhone 上的 Gmail 应用是一个反面教材——连不应该开心的消息都放了笑脸图标。

从描述上，作者应该是想说明在没有网络链接的情况下（图左上角显示菊花圈...）提示无邮件时不应使用笑脸图标。但图片里看到 Gmail 用的 “You have no mail. Please enjoy your day!” 这句明显只是针对没有邮件需要处理的『舒适』状态。难不成没有邮件需要处理还是件难过的事？

如果这里一定要放置苦脸来表示载入不了邮件，那么说明文字相应的就要改动一下。由于图里看到数据还在 loading 中，所以 Gmail 在这里使用笑脸和对应的说明文字是没有任何问题的。

140518更新：

确认这是作者对这个交互细节的看法和通常理解有偏差了。

Little Big Details 最近一次更新的投稿认为『Hit inbox zero in gmail on your iPhone and get a smiley sunshine icon and a message to enjoy your day.』和书中注释完全相反

。 <http://littlebigdetails.com/post/86006922021/gmail-hit-inbox-zero-in-gmail-on-your-iphone-and>

2、《微交互》的笔记-第23页

微交互由四部分组成：触发器，规则，反馈，循环和模式。

手动触发器：源自用户期待或需求，每次触发相同操作，提前展示数据，不要破坏visual affordance，用得越多的触发器越要引人注目,仅在触发器本身无法提供相应信息情况下才使用标签。系统出错时会发生什么？再次尝试时间间隔多长？

微交互的核心是一组规则，是创建一个围绕微交互如何运作的简单的非技术性心智模型。

3、《微交互》的笔记-第102页

反馈是给人看的。用户对状况做出反应，而我们要对用户给出反馈，哪怕是通过机器。为产品赋予人性，赋予它们所不具备的动机和特质，早已不是新鲜事了。你的笔记本不会故意死机，你的手机也不会跟你生气。利用人类的这种特点，设计师应该有意识地让产品具备某种人格。特别是对于微交互，这种短暂又人性化的时刻不会让人觉得讨厌或无聊，而更可能是讨人喜欢。

——所以用户画像很重要，我们了解用户是什么类型，然后做一个气质匹配的产品陪他玩

出错或者其他可能让用户不耐烦的时刻都非常适合用个性化的反馈消解用户的情绪。

同样过于个性化的微交互也是有害的，实际上，一点点个性就能百战不殆，过度个性化不仅会抬高人的期望值（用户会假定微交互异乎寻常地聪明），而且还可能给人留下不着调、诡异或可恶的坏印象。

——个性是为了解决乏味、出错，而不是为了个性而个性，任何设计的初衷都是为解决问题。

4、《微交互》的笔记-第1章

微交互=触发器+规则+反馈+循环与模式

触发器：启动微交互

规则：规定微交互的工作过程

反馈：向用户说明规则

循环与模式：影响微交互的元规则

5、《微交互》的笔记-第108页

界面微动画要速度快，平滑，自然，简单，有目的。动画的作用：

- 1.改变视图的时候保持上下文。例如滚动列表可以看到之前和之后的内容。
- 2.解释刚刚发生了什么。
- 3.展示对象间的关系。比如在拖放结束时，让被拖动对象移动到放置目标中。
- 4.集中注意力。
- 5.提高感知效能，如进度条。
- 6.创造虚拟空间的假象，如怎样让面板滑入滑出，给人位置感和线路感。
- 7.鼓励深度交互。

6、《微交互》的笔记-第69页

对任何微交互而言，触发器被启动的第一个问题就是：我对用户以及用户所处的情境了解多少？你肯定了解一些情况，而有些情况可以用于改进微交互——不要从零开始。

7、《微交互》的笔记-第4章

老虎机的微交互：持续不断的反馈

反馈决定了微交互的特质与个性。

微交互反馈的目的是帮助用户了解微交互规则

- 1.别让反馈给用户造成压力：怎么通过最少的反馈传达同样的信息？

以下情况下，应该给出反馈：

- 手动启动触发器之后或者手动调整规则之后/期间
- 微交互（或周边功能）的状态因为系统启动触发器而发生显著变化

- 用户突破（或超越）了规则时
- 系统不能执行命令时
- 在关键过程特别是需要花较长时间的操作中显示进度
- 过程的开始或结束时
- 模式的开始或结束时，或者切换了模式时

2.最好的反馈永远不会高深莫测

3.用最少的反馈传达最丰富的信息

4.使用被人忽视的东西作为传达信息的手段

反馈方式：视觉、听觉、触觉

视觉：

用户发起的操作都应给出视觉反馈。系统发起的触发器/规则只有一部分（需要用户介入的）应给出反馈。

不要显示多余的视觉反馈。一般情况下，反馈应在原地或附近。

动画：速度快、平滑、自然、简单、有目的

- 改变视图时要保持上下文
- 解释刚刚发生了什么
- 展示对象间的关系
- 集中注意力
- 提高感知效能
- 创造虚拟空间的假象
- 鼓励深度交互

消息：准确、简洁

听觉：

强调（用户主动）、报警（系统发起）

1.耳标

联想其他听到过的声音、与传达的情绪吻合、独一无二

2.语音

简单明了

触觉：

- 强化物理操作
- （声音不可用或不合适）发出警告
- 在触摸屏表面创造人工纹理或摩擦力

反馈的规则：

- 环境变化
- 持续时间
- 强烈程度
- 重复次数

8、《微交互》的笔记-第34页

创新是没有办法的办法——反思自己的设计，保证基本可用是基础，在此之上才能锦上添花，求新求变不是设计的出发点

9、《微交互》的笔记-第110页

错误消息的反馈文本不仅要说明出了什么错，还要告诉人怎么纠正它。理想情况下，还应该随同消息提供一种纠正错误的途径，比如，不要仅仅告诉用户密码输错了，还要重新把焦点定位到密码框里或者提供一个找回密码的链接。

10、《微交互》的笔记-第56页

仅仅为了不同而不同很少能做好，而做得更好通常才会让人觉得不同。

11、《微交互》的笔记-第37页

不过要知道，不可见并非微交互（以及任何交互）设计过程中直接追求的目标，而是上下文和技术的副产品：在这种情境下，最好把什么隐藏起来？或者基于技术条件限制，如果不把什么隐藏起来，可见控件就没有地方显示？最好的微交互应该具有恰到好处的界面，而不是空间越多越好。

12、《微交互》的笔记-第139页

在本书写作时（2013年3月），iOS6只允许在角标中显示数字，而且只允许显示4个角标。怎么办？如果闹铃事件是6:30，可以在角标中显示630，那要是12:30呢？1230能表达清楚吗？而且这样使用角标也不合规则，因为角标里的数字通常都用来表示未读信息的鄂数量。这里提到的『只允许显示4个角标』是翻译错误还是原文错误？应该是『只允许显示4位数字』才准确吧？给完全不了解 iOS 系统的人看这要理解成同时在一个图标/屏幕上出现或只出现4个角标都是可能的。

这么不用心的考量以及把这书用非全彩出版和骗钱有什么区别？

13、《微交互》的笔记-第43页

与其他产品相关的（培训性或营销性）方案不同，微交互不是品牌创意的用武之地；相反，微交互非常注重实用性，目的就是清楚明确地传达功能。这并不是说微交互不重视奇思妙想或个性设计，而是说前提必须确保标签清楚明确。

14、《微交互》的笔记-第130页

循环可以让你创造出“长久的赞赏”，长久的赞赏意味着随时间的推移向用户交付新体验或新功能，而不是一成不变，借以建立用户忠诚度。从微交互角度讲，长久的赞赏意味着随时间的推移调整微交互，让用户感觉是专门给自己定制的，甚至是一个全新的东西。这要求使用长循环，也就是持续时间与微交互所在设备一样长，甚至更长时间的循环（如果能远程存储行为数据的话）。

15、《微交互》的笔记-第32页

多数人经常要用的微交互，应该最引人注目，少数人有时会用的微交互，应该容易注意到，极少数人不常用的微交互，应该通过搜索找。

16、《微交互》的笔记-第109页

在用户改变对象状态或切换使用模式时，过渡是微交互动画的重要实现方式。过渡能给人位置感和路线感，让用户知道自己身处何地、路在何方。

17、《微交互》的笔记-第87页

任何算法由四部分构成：顺序，决定，重复，变量

18、《微交互》的笔记-第3章

Mac OS X关于“另存为”操作的例子：

微交互的改变一般勿违背用户心理预期，除非明显好很多。

设计规则

确立目标（应为最终状态）

应自然而然，潜移默化

规则决定：

- 微交互如何相应被激活的触发器
- 交互期间用户可以进行什么操作
- 动作发生的顺序及时间
- 使用什么数据及数据的来源

《微交互》

- 简洁的相关配置及参数
- 什么时候应该提供什么反馈
- 微交互处于什么模式
- 微交互是否重复以及多久重复一次（一次性 or 循环往复？）
- 微交互结束时会发生什么

生成规则

可视化呈现：逻辑关系图（流程图）有助于描述规则。

动词（用户采取的动作）+名词（辅助实现动作的对象）

每个“名词”都应该独一无二。

两个（或多个）看起来相同的名词，他们的行为也应该相同。

相同的名词在不同的地方不能行为迥异。

微交互的最高境界通常是给用户多种多样的动词，最少的名词。

屏幕与状态

采用独立屏幕不如利用状态变化。

用户可以与之交互的对象至少有3种状态

- 1.邀请/默认状态（首次呈现）
- 2.活动状态（交互中）
- 3.更新后的状态（停止交互）

约束条件

- 可用的输入和输出方法
- 输入内容的类型和范围
- 要付出哪些代价
- 可用的数据有哪些
- 可以收集那些数据

不要从零开始

可以收集到用户哪些数据？通常非直接来源于用户的数据可能更有用。

善于利用情境和之前的行为预测或增强微交互。

理解复杂性

有些微交互必须由用户来控制，但系统仍然需要处理一定的复杂性。

最核心的复杂性在什么地方？用户掌握着哪一部分？何时需要介入？

微交互中涉及以下复杂性应该让系统处理：

- 快速计算
- 同时执行多个任务
- 记忆很多东西
- 检测复杂模式
- 从大数据中搜索特定的物品

有限的选项和聪明的默认项

提示用户最可能采取的下一个动作（去掉其他选项或视觉给出变化）。

减少规则最好的办法是给用户提供有限的选项。

替用户做默认选择，某些情况下应给出与决定相关的说明。

最明显的默认项应该是大多数人大多数情况下会用到的。

位于列表顶部或底部的项目，比位于列表中部的项目更容易被记住。

需要用户做一连串决定先从简单开始到复杂。

需要不断重复的微交互应该易于辨识。

只做一次或偶尔的微交互应该易操作，列出所有选项一目了然。

预防错误，而非弹出错误消息。

微文案

如果标签足够，绝不要使用说明性文案。

放置标签最佳位置实在操作地点上方，其次是被操作对象之上或之中。传统做法是把标签放在图标下方。

标签放字段上方，字段中放示例文本。

说明性文案与控件严格匹配。

不要使用多重否定。

算法由以下四部分构成：

- 顺序
- 决定
- 重复
- 变量

19、《微交互》的笔记-第20页

标志性时刻就是那些成为产品差异化标志的微交互。一个自定义的触发器控件或者一段漂亮的“加载中”动画或者一种好听的声音（“你有封新邮件，请注意查收”）都可以像功能一样加以宣传或营销，而且可以跨平台使用，或者在同一品牌下不同的产品中使用。这种标志性时刻有助于提升客户忠诚度和品牌差异化。就像Facebook的“like”按钮，已经成为其品牌不可分割的一部分。

20、《微交互》的笔记-第59页

用户可能知道也可能不知道规则的存在，而规则让自己可见的方式有两种：允许做什么和不允许做什么。显然规则决定什么事情发生，以什么顺序发生，进而影响用户体验。

21、《微交互》的笔记-第48页

虽然用户不必手动出发这些（系统）触发器，但最好是（比如在设置里）提供选项让用户能够调整它们。系统启动的每个触发器都应该有手动管理及禁用它们的选项。理想情况下，应该在微交互已经触发的时候给出这些选项（“不再显示这些提示”），至少有相应的设置。

此外，即使存在系统触发器，用户也可能需要一个手动控件（如印象笔记的同步按钮：后手动操作和系统提示合并）。比如，用户可能需要手动同步文本，而不是等待系统自动同步。手动控件可以提供一个保障，同时也能在系统出错（网络连接断开或传感器未指示）时保证能够触发微交互。

22、《微交互》的笔记-第5章

模式应该尽可能少用，最好没有模式，有也不要超过一种。

是微交互的分支，主要目的是执行一种不常见的操作，如设置、编辑模式。

模式越少，导致用户困惑的机会越少。

模式最好有自己单独的屏幕。

《微交互》

从一个模式返回之前的模式时，用户会希望原来的模式会处于自己离开前的状态。退出特殊模式回到主模式后，会希望主模式中的交互方式与自己离开时相同，当然要反映在特殊模式下所做的修改。

如果切换到一个模式就是为了执行一个简单的操作，会很令人讨厌。

可以快速完成的动作：

1) 弹簧模式：在用户按下不放的情况下才会被激活。好处在于，用户很少会忘记自己在不同的模式里，缺点是不适合费时的操作和复杂的输入。对微交互来说，弹簧模式最好少用，主要适合设备和家电产品。也可以是把用户引导到微交互的不可见触发器（如自动填充搜索字段）。

2) 一次性模式：在用户发起模式时，模式持续时间与动作执行时间相等，动作结束，模式就结束。最适合快速的任务切换或上下文工具。也可用在手势或语音微交互中。

循环通常用于设定持续时间，核心是计时，即确定微交互的速度和持续时间。由规则指明。

- 计数（for）循环
- 条件（while）循环
- 集合循环
- 无穷循环

开放循环：不响应反馈，执行完就结束

封闭循环：内置反馈机制

确定循环参数对用户体验影响极大

循环可以用来识别行为

长循环：持续时间与微交互所在设备一样长，甚至更长。扩展微交互，使其不仅限于一次使用。

随时间推移不断适应用户习惯。

渐进揭示/减少

23、《微交互》的笔记-第7页

产品设计的的好坏，取决于产品中最小的那一部分设计得如何

24、《微交互》的笔记-第15页

微交互的结构：触发器、规则、反馈、循环与模式，这四个部分是设计和分解微交互的一种方式：触发器启动微交互，规则规定微交互的工作过程，反馈向用户说明规则，循环与默认是影响微交互的元规则。

很多微交互都始于对用户需求的理解：用户想要做什么，什么时候会这么想，有多频繁。用户需求决定了触发器的使用情境、操作方式和存在时间。

触发器不一定由用户启动。如果设备或者应用检测到某些条件，同样可以由系统启动触发器，从而开始一次微交互。这种系统启动的触发器越来越多。触发条件多种多样，比如收到新邮件，到了预定时间，或者股票价格、用户位置变动等等。比如手机静音，让手机与日程关联可以使它在开会的时候自动静音。或者与位置关联，让它在进入电影院或音乐厅时自动静音。随着应用和设备的感应范围越来越广泛、环境感知能力越来越强，让它自己做决定的机会就越来越多。

25、《微交互》的笔记-第4页

微观交互与宏观功能完美契合是最强大的。用户体验设计的本质就在与此：既要关注细节，又要关注大局，只有这样，才能获得最好的体验。

26、《微交互》的笔记-第84页

把标签放到文本字段内部时要多加小心。因为一旦消失（必须的，因为用户得输入内容），用户可能会忘记这个字段是干什么用的，而且除了单击文本字段外部，没有别的办法再看到标签内容。更好的做法是把标签放在字段上方或者按钮上，同时再在字段中放一个示例文本。

27、《微交互》的笔记-第2章

手动触发器原则：

- 1.吻合用户需求
- 2.每次都触发相同动作
- 3.提前展示数据
- 4.不要破坏视觉使用情境
- 5.手动触发器要引人注目
- 6.不要引起用户对使用情境的错觉（似与第4条重复？）

手动触发器=控件+控件状态+文本或图示标签

控件状态：

- 默认状态
- 活动状态
- 悬停状态
- 翻转状态
- 鼠标点击/手指轻击/进行中
- 切换状态
- 设置状态

标签：

简洁、明确、一致性、用使用者的语言描述

系统触发器常见的启动条件：

- 错误
- 位置（来到某个地方）
- 收到数据
- 内部数据（如过一段时间屏幕自动降低亮度）
- 其他微交互（如向导式页面，第一步结束触发第二步）
- 其他人（的行为）

每个触发器都应该有手动管理或禁用的选项，在出发的时候给出设置选项

有系统触发器同时也需要手动控件的情况

系统触发器的规则：

- 多长时间启动一次？
- 已有的用户数据？
- 是否有已经启动的指示器？状态更新？启动前后变化？
- 出错时的反馈？是否重试，间隔多长？

最重要的是，触发器要能把用户带到实际的交互中。

28、《微交互》的笔记-第123页

如果是必须设计一种模式，对微交互而言，最好能给它设计一个屏幕。单独的屏幕可以减少错误和防止困惑，因为用户可以清楚地知道自己进入了一个不同的模式，而不是出于不熟悉的状态。过渡到新模式以及返回之前的默认模式的效果，也可以作为有用的提示，表示用户要到一个特殊的地方做一些特别的事。

用户进入另一个模式然后再返回之前的模式时，会希望原来的模式还处于自己离开的状态，尽管在另一个模式中操作的结果可能会反映到默认没收。用户在退出特殊模式返回主模式后，会希望主模式中的交互方式与自己离开时的相同，当然要反映出他们在特殊模式下所做的修改。

29、《微交互》的笔记-第80页

在设计产品和流程时，就要考虑根本不让用户出错。

理想情况下，你的微交互应该在用户没做错任何事（因为不应该有用户会做错的事）的情况下，一条错误消息都不出现，而只在系统本身不能正确响应的情况下才会出现错误消息。弹出错误警告是懒人的做法。如果确实发生了错误，那微交互应该首先想办法确定怎样解决问题。

30、《微交互》的笔记-第28页

触发器必须让用户在使用情境下认出来其是触发器。

31、《微交互》的笔记-第72页

所有活动都有内在的复杂性，超过了某个临界点，简化是不可能的。既然如此，唯一的问题就是如何对待复杂性。要么让系统来处理，不让用户介入；要么让用户来处理，把更多的决定权（包括控制权）交给用户。涉及微交互时，人们大都倾向于减少用户控制，而让微交互处理大多数复杂性。对此，我们想给出一个忠告：有些微交互必须由用户来控制，但即便如此，系统仍然需要处理一定的复杂性。

首先，要找出最核心的复杂性在什么地方，确定用户掌握着哪一部分以及何时需要介入。如果用户介入绝对必要，则适时将控制权交给用户。

——就如带孩子，不可包办，不可放置不管，需要有节制的智慧

32、《微交互》的笔记-第107页

理想情况下，视觉反馈应该出现在用户输入的附近或者原地。如果提交按钮在屏幕底部，就不要把错误信息显示到屏幕顶部。人在注意某些东西的时候，视野会变窄。视野外部的任何东西都将被无视。如果必须把视觉反馈放到远离视觉焦点的地方，可以通过让它运动（比如慢慢淡入）来吸引注意力。

33、《微交互》的笔记-第164页

版权说明

本站所提供下载的PDF图书仅提供预览和简介，请支持正版图书。

更多资源请访问:www.tushu000.com