

从苹果产品的设计错误学教训

前言

用 iPhone 的时候，你是否遇到过以下的情况？

1. 花不少时间整理屏幕上的图标，却总也不能得到满意的结果。（所有 iOS 版本）
2. 按 Home 键，指纹解锁手机，结果 Siri 跳了出来，问你想做什么，同时停掉你正在播放的音乐。被烦扰多次之后，决定关闭 Siri 功能。却发现问题仍然存在，只不过现在反复跳出来的是“语音控制”，或者问你“是否想启用 Siri？”（iOS 10）
3. 看到手机弹出通知，解锁去看，通知列表却不知去了哪里。以前从屏幕“上边沿”划下来，就能看到通知，可现在却只看到日期和天气预报。（iOS 10）
4. 想在手机上看个小视频，结果声音从蓝牙音箱出来了。想把声音切回手机上播放，从“下边沿”往上划，却发现以前的“媒体输出设备”菜单已经不在那里了，到处找不到。只好干脆把蓝牙给关了，然后下次还遇到同样的问题。（iOS 10）
5. 需要打开控制面板，从屏幕下边沿往上一划，却发现没有动静。
6. 锁屏时左划屏幕想启动相机，结果屏幕网上弹了一下又缩回去了，相机没能启动。反复几次都是一样的结果。（iOS 11）

每一次出现上述情况，作为一个讲求生活品质的人，你是否希望下次不要再这样？你是否拿着手机琢磨，却没找到解决方案？你是否上网搜索，甚至到苹果的用户论坛发帖提问？如果你需要为一个产品做这样的事情，说明里面存在着不人性化的设计。真正人性化的设计，是不需要用户去想，甚至去研究这些问题的。

懂一点编程的人可能认为这些是所谓“bug”，其实不是的。Bug 是指程序员犯了错误，写出的程序不符合设计要求，或者产生意想不到的结果。而我们这里谈的问题是“设计错误”，也就是程序员没有犯错误，而是设计本身出了问题。设计解决的问题是“做什么”，而编程解决的问题是“怎样

做”。如果“做什么”这个决定都错了，那么再仔细的程序员也无能为力。用很可靠的方式去解决错误的问题，终究不会得到好的结果。

这份报告就是为这些“设计错误”而产生的，它记录了 iPhone 设计中存在的一些不够人性化的地方。我们试图从现象出发，追寻问题产生的根源，探索设计中需要遵循的一些原理。有了这些原理，我们就可以在未来的产品设计中避免类似的错误再次发生。

这篇报告的目的当然不是否认或者贬低苹果公司的产品，而是为了探寻通用的设计原理。这些问题当然不是苹果手机所独有的，然而因为苹果公司一直处于世界设计领域的前沿，被众多其他厂家尊敬和模仿，我觉得利用苹果产品中存在的一些问题来解释这些原理，比用其它品牌作为例子，更加具有代表性和说服力。

客观的探讨这些设计问题，对于广大消费者，苹果公司，其它手机厂商，应该都具有一定的意义。

这份报告的版权属于王垠（及 CIIA）所有，授权给特定的顾客付费阅读之用。文中的内容受到国际版权法的保护，请勿拷贝，抄袭，抓图分享其中的内容。被授权的客户 email 被标注在每一页的底部。如果你在公共网络上发现了这份报告的拷贝，请向我们投诉。投诉时请注明下方的 email 地址和发现这份报告的网址。

投诉 email 是：legal@yinwang.org

目录

图标排列是一个智力游戏	3
Home 键的复杂度	9
滥用边沿操作	13
控件隐藏太深	16
容易混淆的划动操作	20
摄像启动了没有？	22
总结	25

图标排列是一个智力游戏

现在我们来看一下 iPhone 自诞生以来就存在的一个问题：调整 iPhone 桌面图标的位置，是一件很麻烦的事情。如果你花一些时间来研究它，就会发现这简直是一个智力游戏。

稍微有点条理，懂得“收纳”的人，都会希望把相近功能的 app 摆放在相邻的位置，希望把常用的 app 放在顺手的位置。如果你不需要确定它们具体的位置，至少应该可以决定它们位于哪个页面吧？然而 iPhone 的设计，使得这么简单的事情变得不必要的麻烦，导致各种意外的结果。

比如下面左图的屏幕中，我希望把“饿了么”和“大众点评”放在一起，因为它们都可以点外卖。所以我就把“饿了么”拖到“大众点评”的右边去。经过一阵眼花缭乱的动画之后，屏幕变成了右图的样子。不仔细看的人根本不知道发生了什么，要反复试验几次，你才会明白这些图标是按什么规律移动的。



发生了什么呢？原来“大众点评”和“饿了么”之间的图标，都被“右移”了一个位置。“淘宝”本来在第三行最后一列，现在跑到了第四行第一列，支付宝本来在第四行第一列，现在跑到了同一行的第二列，以此类推……直到遇到“饿了么”留下的空位，这个右移操作才停下来。所以“手机营业厅”没有移动位置，停在了最后的位置。

这个无辜的用户动作导致了什么后果呢？它导致近一半数目的图标被自动换了位置。很多用户对此不知情也不观察，所以不知道系统是按什么方式来移动这些图标，只是感觉 iPhone 的图标怎么不老老实实待在自己的位置，到处乱跑很讨厌。

这样的设计违反了一个重要而简单的原则：

人们希望上次用过的东西，下次用的时候还在原来的地方。

记得小时候妈妈告诉你，用了剪刀要放回它原来的地方吗？不然她下次用就不知道去哪里找了。这里是同样的原理，你根本没碰过的图标，被系统挪到其它地方去了，下次用你就找不到了。

从计算机科学的角度来看，人要查找一个物品，是一个“多层加速”的构架，就像电脑的 CPU 一样。在最快的级别，人的脑子里存在着一种对于物品位置的高速缓存（cache），或者叫做“空间模型”。这种模型与现实世界产生“对应关系”，你可以把模型看成是一个从“物品”到“位置”的哈希表。你输入“剪刀”，模型就会告诉你：“剪刀在客厅门口柜子的第二个抽屉里。”

如果物品的实际位置与模型一致，人就能依靠头脑里的这个模型快速的找到它们，而不需要对空间进行查找。模型是一个“心理”过程，类似查询哈希表，它的时间复杂度接近 $O(1)$ 。但如果物品的位置与模型不一致，人在模型给出的位置找不到它，就必须进入这个构架更慢的第二层：对空间进行查询。对空间的查询是一个物理的过程，它的复杂度是 $O(n)$ ，其中 n 代表空间的大小。

所以 iPhone 桌面图标的设计，会导致人脑中的高速空间模型失效。下次要付款的时候，人掏出手机想用“支付宝”。按照空间模型给出的位置，它应该是“屏幕左边往下第四个图标”，结果按下去却按到了“淘宝”。进小区大门，掏出手机按屏幕左下角的图标，以为会启动“门禁钥匙”，结果启动了“MiHome”……

稍微聪明点的人，都会希望常用的图标放在顺手的地方。像我就喜欢把“门禁钥匙”放在左下角，这样左手单手操作很容易按到。可是它现在跑到旁边去了，造成不必要的意外，而且不顺手。被系统自动移位的那十几个图标，都会在下次用的时候造成不必要的意外。人需要重新适应和学习十几个图标的新位置，仅仅因为他移动了一个图标，这是很不合理的。

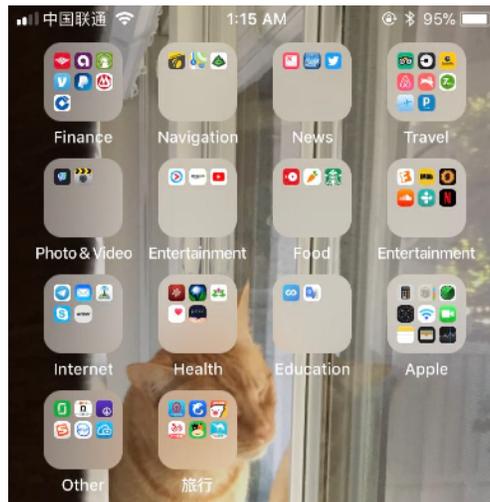
* * *

很多人察觉到了这个问题，所以他们采用了最懒的解决方案：从来不去移动和组织图标，让它乱着就乱着，从来不删除 app。有一些苹果粉丝，当你提到这个问题的时候，会跟你讲道理：“iPhone 这样做就是最好的办法。一般人寻找图标都是靠眼睛看，所以位置在哪里根本不重要！”

人眼查找图像的能力是有限的，大部分时候其实是靠位置。临时靠眼睛找图标，这种做法对于只有少数 app 的用户也许还行，多一点就不行了。一般用户用了一段时间 iPhone，都会有超过一屏的 app。而且如果你在一个满屏的桌面加入一个图标，屏幕最后的那个图标会被自动转移到下一屏，导致下一屏所有的图标都被右移。下次你再找那个换了桌面的图标，就会需要一屏一屏的翻，很费时间。

有人说这有什么关系呢？iPhone 有强大的搜索功能，你只需要在搜索栏输入 app 的名字就可以了。可是你发现没有，这样的做法本质上就是在敲命令行。想想吧，你在超市付款，后面跟着一队的人，在等你输入命令行，找到支付宝 app……

另外一种解决方案，是把图标进行分类管理，就像这样：



很快你就会发现这样的管理方式不可行。为什么呢？因为它给人脑增加了不必要的负担：

1. 首先，你不能很快的从 app 的“超小号缩略图”找到它，况且有些分类的图标超过了 9 个，根本不会在屏幕上显示。所以要找一个 app，你得先弄明白它属于哪个“类别”。
2. 但类别这东西，很多时候是模棱两可的。你又不是图书馆管理员，所以到用的时候就想不起之前给 app 分了哪些类别，这个 app 属于什么类别。

3. 等你弄明白了 app 的类别，你还得先找到类别所在的图标。如果你分了挺多类别，这是一件很难的事情，因为类别不能依靠人脑的“视觉记忆”进行高速匹配，而必须通过阅读它们名字，而名字字体又很小.....

啊好累！这就是为什么我曾经三度试图把 app 进行分类管理，最后都放弃了。反思分析一下自己脑子里需要进行什么样的操作，你就会发现这为什么不可行。

* * *

那么更好的图标管理方式是怎样的呢？其实很简单。我们只需要解决 iPhone 的图标系统造成的几个问题就行。然后你会发现，满足这两个条件的图标管理方式其实早就已经存在了。那就是古老的 Windows 桌面图标的管理方式。现在我们来看看 Windows 的桌面图标管理方式具有哪些优点：

1. 用户可以把图标摆放在（网格上）任何自己想要的位置，而不一定是连续的。这样他们可以随心所欲的组织图标的位置。你可以按照分类组织，按照使用频率组织，甚至可以摆出一些花样。
2. 用户移动一个图标到新的位置，必须首先把占据该位置的图标移走。要求用户主动移开这个图标而不是让系统自动移动它，看起来麻烦点，其实是让用户“知情”，避免下次找不到被移走的图标。
3. 用户没有碰的图标，全都待在它们原来的位置。这样避免了下次找不到图标，或者按错图标的情况。
4. 用户可以通过“矩形选择”操作，“control 选择”操作，选择出多个图标，进行多个图标的移动或者删除。

然后你还会发现 Android 系统的桌面图标管理也是那个样子的，只是它不能方便的进行“多图标选择”而已。很多人从 iPhone 换到 Android 系统，不习惯它移动图标还得先挪出个空位，而其实那才是更好的方式。

从这一节所采用的分析方式里面，你也许会悟出一个道理：

道理1:

把人看成一台机器，对它进行数据结构和算法分析，往往能够得到更加人性化的设计。

把人当成“机器”分析，结果得到“人性化”的设计，这个结论看似荒谬，但其实是有道理的。你会惊奇的发现，很多时候人们为了一台机器（比如自己的汽车）做出的各种考虑（比如保养啊，装饰啊，维修啊），很多时候比他们照顾自己的身体还周到很多。

非人性化设计的常见来源，就是把人当成一种地位还不如机器的东西，忽视他们生活的方便，效率和舒适程度。所以把人当成一台机器，考虑它的效率，损耗和保养，反而得到更加人性化的设计！

自觉付费

免费的部分到此结束，你现在阅读的这份文档后面的章节并不是免费的。

这个文档采用一种新的商业模式，我把它叫做“自觉付费”。自觉付费的原理就像一个水果摊。我懒得去守着这个摊子，我就把水果摆在那里，你们吃了之后自己付钱就行。

这跟我其他博文的“自愿付费”形式不同。“自愿付费”的意思是，你看了我的文章，如果你不喜欢，或者你根本没看完，那么你可以不付钱。“自觉付费”的意思是，你如果看完了书的内容，那么你就已经“享用了”它，你就应该付钱。如果你不付，就是有道德过错的，如果你不想付钱，就请不要看。

这就像我摆个“自觉付费”水果摊，你们吃了自己付钱就好，结果你吃了没付钱，这就不好了。你是否付费，将会影响到我的水果摊是否可以继续营业，会影响我继续写书的动力。我的博客含有代码，会统计这段时间的阅读人数。我会根据阅读人数与付费的比例，来推断是否有很多人没付钱。如果我发现付费的比例太小，我会适当调整作法。

如果你想继续读下去，请扫描下面的二维码付费之后阅读。付费前请核对收款人名字。



Home 键的复杂度

跟我合作过的一个教授曾经开过一个玩笑，他说：

“Every Apple product has just one button, and that button takes a picture!”

“每个苹果产品都只有一个按钮，那个按钮用来拍照！”

这显然不是一个准确的描述，但它用一种轻松的方式调侃了一种常见的偏执心理：只有一个……所以这是多么的简单和优雅！这个说法让我记忆犹新。每当想起它，我就会注意到 iPhone 的标志性设计：它正面的 Home 键。

你有没有想过，为什么 iPhone 正面只有一个按钮？通常的说法是为了简单：“只有一个按钮，当然简单了。不用考虑该按哪个按钮！”

只有一个按钮就是简单吗？从程序语言的角度来看，一个按钮就是一个函数，它接受某种输入，得到某种结果。



你有没有想过，iPhone 的 Home 键有多少种不同的输入？“单击”，“双击”，“长按”……虽然你不需要选择按哪个按钮，却需要从多种输入中选择一个。

所以虽然 iPhone 只有一个按钮，这个按钮上却存在着相当大的复杂性。用点术语，我们说这个按钮上存在着“编码”（coding），或者叫“多工”（multiplex）。虽然它看起来是一个按钮，实质上却是好几个按钮的功能，合并在一起。

如果你会写程序，懂得函数是什么，就会发现程序里的多个函数，几乎总是可以被合并成一个。比如，你有两个函数 f 和 g，它们接受不同类型的输入，那么你可以定义这样一个函数 h，它是 f 和 g 的合体：

```
function h(x)
{
  if (x 是 f 的输入类型) {
    return f(x);
  }
  else {
    return g(x);
  }
}
```

你可以把两个接受不同输入的函数合并成一个。同样的，你也可以把两个接受不同输入的按钮合并成一个。但这样的函数和按钮，它们真的简单吗？不，它们其实变复杂了。有经验的程序员都知道，把毫不相干功能的函数合并成一个，是程序员的大忌。

为什么这样不好呢？因为函数合并之后，你就不再能从函数的“名字”直接看出它是做什么的。同样的，按钮合并成之后，你就不再能从按钮的位置或者外形知道它是做什么的。在看到这样的函数或者按钮的时候，人的脑子里多了一层疑惑，多了一层判断。所以它们其实比分开的函数或者按钮还要复杂。

iPhone 的 Home 按钮，对于同一种输入，在不同的状态（state）下，还会有不同的功能。比如同样是“双击”，在锁屏的状态下会启动 Apple Pay。但在处于解锁状态的时候，却会启动任务管理器。

从程序语言理论的角度来看，这个函数依赖于某种全局或者外部状态（state），它不是一个纯函数（pure function）。这种不纯的函数很容易把人搞糊涂，因为同样的输入在不同的时候，会得到不同的输出。

如果你仍然觉得一个按钮就是简单，那么我可以设计这样一个手机给你，它绝对的“简单”。不但正面只有一个按钮，而且机身上不存在任何其它按钮，它没有触摸屏，也没有软键盘可以用来输入。所有的输入操作，完全通过那一个按钮来进行。

这是完全可行的，你知道为什么吗？因为只要有一个按钮，我就可以像早期的电报机一样，使用摩尔斯电码（morse code）输入任何信息！可是你愿意用这样的手机吗？它可是绝对“简单”的手机，真的只有一个按钮！当然你不会。为什么呢？因为它并不是真正的简单，它的输入包含了太多的复杂度。

爱因斯坦有句名言：

“Make things as simple as possible, but not simpler.”

“事情应该力求简单，但不能过于简单。”

我觉得这句话用来形容 iPhone 的 Home 键，是比较恰当的。过度偏执地追求“简单”或者“极简主义”，其实会让事情变得复杂，难用，不人性化。

回到我们最开头提出的困惑吧：按 Home 键，指纹解锁手机，结果 Siri 跳了出来，问你想做什么。这种情况出现的原因，往往是由于按下 Home 键停留的时间长了一些，结果被 iPhone 认为是“长按”，所以启动了 Siri。

可能有人会笑我，谁叫你按 Home 键那么久呢？然而有些时候，你没法很好的判断按 Home 键的时间。一个例子就是因为 iPhone 6 的指纹阅读器速度比较慢，所以按了 Home 键之后还得让手指在上面停留一会，才能解锁。这样“按键”和“停留”，很容易被人的运动神经自动转换为“长按”这个动作，按下去，等一会儿读指纹，再放开。这就被 iPhone 认为是长按，所以启动 Siri。这种情况在不经心操作手机的时候特别容易出现，比如开车的时候。

至于双击，这动作就更需要技巧了。很多用户电脑鼠标的双击都练不好，你还让他们在手机 Home 键上双击。按下两次之间的时间间隔，到底要多短才算双击呢？很多人都掌握不好。不但要双击，人家还得记住在不同“状态”下双击 Home 键的功能是不一样的。在解锁的时候双击，会打开任务管理器。在锁屏的时候双击，却会打开 Apple Pay。

总而言之，用户对于 Home 键的迷惑，很多时候来自对输入的混淆。我们应该避免让用户去掌握那些只有微妙差别的输入：单击，双击，长按……

这就是为什么每次遇到 Home 键的这些奇怪设计，我都会想起那句话：“每个苹果产品都只有一个按钮，那个按钮用来拍照！”仔细回味一下，真是别有深意。

* * *

既然一个按钮不等于简单，那么我们有什么更好的解决方案吗？

既然一个按钮其实可以很复杂，好一点的解决方案，就是把不同的功能分开在不同的按钮上，合理增加按钮的数目。既然你要完成多种不同的功能，那就不可避免要做出选择。就算只有一个按钮，你也需要选择如何操作它。与其让输入变得复杂多样，容易混淆，还不如多放几个按钮在那里，这样每个按钮只接受一种输入：按下。

由于每个按钮只有一种输入，所以他们不会在输入方面发生混淆。由于按钮的外形和位置可以“暗示”用户它的功能，用户不再需要去了解 and 掌握“单击”，“双击”或者“长按”这些不同的输入。他们只需要记得，左边的按钮表示“返回”，中间的按钮表示“任务管理器”，右边的按钮表示“Siri”。

注意到了吗，这左中右三个按钮，分别完成了 Home 键“单击”，“双击”，“长按”所完成的功能。现在它们分开了之后，用户只需要记得对它们的一种操作：按下。他们不会再为“双击”和“长按”造成的后果所困惑。

其实还可以更加灵活一点，你可以让其中一个按钮成为“可编程”按钮，就是让用户来决定它按下去之后做什么。比如要是用户不喜欢用 Siri，那么他可以用 Siri 的按钮来干其它事情。喜欢用对讲机 app 的用户，可以把这个按钮设置为“Push To Talk”按钮，这样他们不需要解锁手机打开 app，就可以迅速地进行对讲操作。经常听歌的人可以用它切歌。经常要用手电筒的用户，可以把它设置为手电筒开关。经常写语音备忘录的人，可以用它来启动录音。想把手机作为防身武器的人，可以用那个按钮启动高压电击…… 花样可以有很多，只需要一个可编程按钮，就可以给用户带来很多的方便。

我提出这个把功能分开在多个按钮上的作法，不等于你可以滥用这个方式，走到另外一个极端，在手机上制造太多的按钮。原因是用户分辨不同按钮的能力还是有限的。少数几个按钮可以接受，按钮多了形状还差不多，用户同样容易糊涂。

现在总结一下我们这一节学会的道理：

道理2：
数目少不等于简单。

滥用边沿操作

所谓“边沿操作”，就是从屏幕边沿往屏幕内划的操作。比如在 iOS10 里，从屏幕下边沿往上一划，你可以看到如下的“控制中心”：

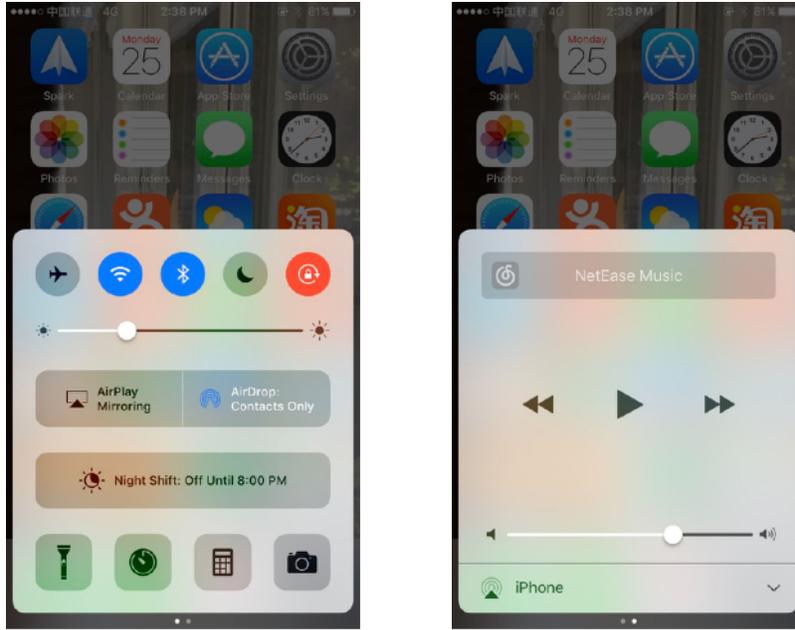


存在少数直观的边沿操作，问题不会很大。可是 iPhone 似乎在边沿操作上做了太多文章，发挥过度，这导致很多用户被迷糊了很久，才发现某些功能的存在。

就像我开头提到的问题：想在手机上看个小视频，结果声音从蓝牙音箱出来了。想把声音切回手机上播放，从“下边沿”往上划，却发现以前的声音控制已经不在那里了，到处找不到，只好干脆把蓝牙给关了。这个问题存在于 iOS 10 里面，iOS 11 修正了这个问题，但又带来了另外一些同类的问题。

问题出在哪里呢？在于 iOS 10 里面，把声音控制放到了控制中心的“第二页”。一般用户能够发现屏幕下边沿上划有个“控制中心”，已经算是对 iPhone 比较了解了。现在你来个 iOS 升级，把控制中心拆成两页。然后得寸进尺，期望用户不但知道如何打开控制中心，他还得知道控制中心在左右方向上分两页，第二页里面放着声音控制。

但是你给了用户什么提示信息，告诉他右边还有一页呢？你在控制中心的最下面，最不容易注意到的地方，画了两个小点，小得可怜……极少有人会注意到那两个点，极少有人会想到，从下边



沿划起来一个面板，它居然横着铺开有两页。这就好像从柜子里拉出一个抽屉，它有两个抽屉那么宽！把边沿操作划起来的面板，自然而然当成抽屉一样的物体，就是很多人察觉不到那第二页的存在的原因。这种面板根本就不应该有第二页。

不管你给出多少提示，告诉用户第二页的存在，音乐面板隐藏在面板的“第二页”，已经超越了普通用户可以忍受的深度了。我划起控制面板就应该直接看到音乐控制，为什么还要多划一下？控制面板应该只有一页才对。不然我把它划起来，还得判断我在其中哪一页，然后左划右划去找我需要的控制。要是不小心手指按到了亮度或者音量滑动条上，左划右划的动作还会不小心改变亮度或者音量，而不是翻页。所以有些地方是可以划的，有些地方不可以。我为什么需要去想这些事情？

另外，你会发现 iPhone 的“上边沿”操作也有类似的问题。从上边沿划下来的面板，包括了“提示信息”和“组件”两个页面，一左一右。这引起了很多混淆，导致很多人在 iOS 升级之后，找不到锁屏界面的提示信息解锁之后在哪里。iOS11 改进了这个问题，然而并没有完全的解决它。

所以解决方案是什么呢？那就是不要把这种边沿操作划出来的面板做成多页的。屏幕有多宽，面板就只有那么宽，而不应该隐藏更多的内容。

幸运的是在 iOS 11 中，苹果设计师们发现了这个问题，所以对这个面板做出了改动。可是不幸的是，他们并没有效的利用面板的空间，有好些重要的控制被隐藏到了对控件“重按”（force touch）之后弹起的“第二级面板”里面。

所以虽然 iOS 设计师们发现控制中心不应该有两页，他们仍然把某些控件隐藏在太深的位置。在下一节，我来重点讲一下如何决定功能在界面上的“深度”。

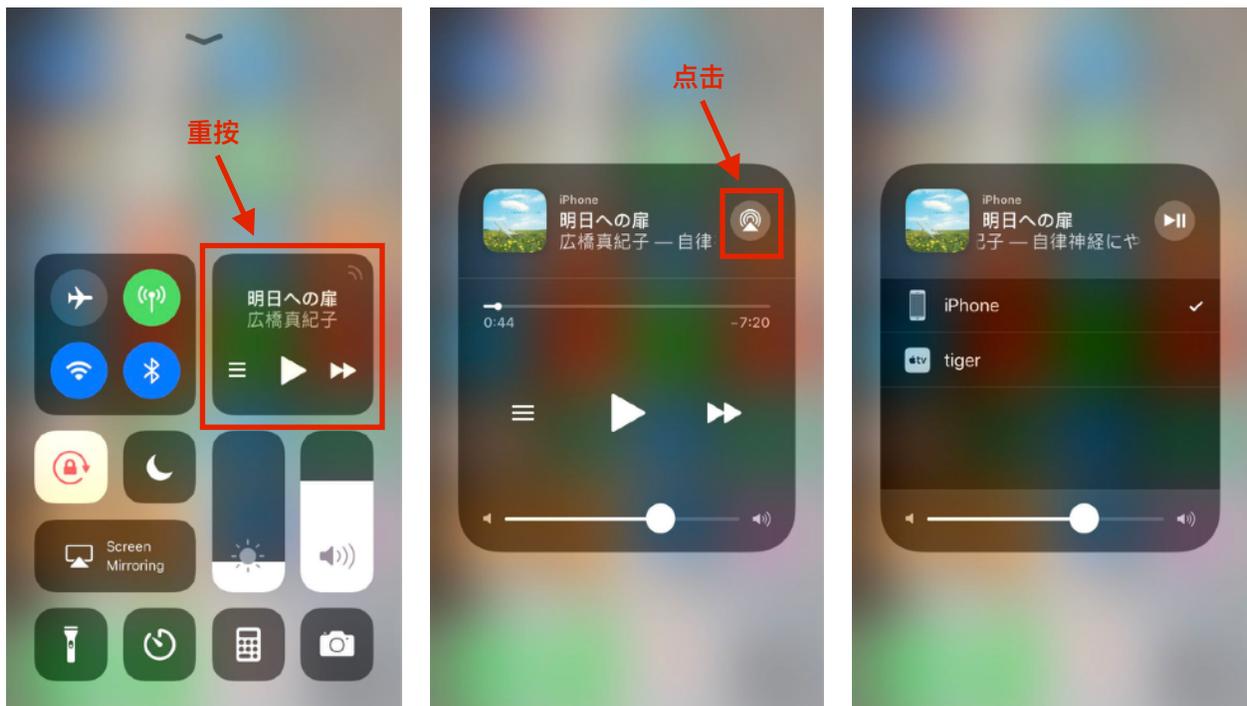
道理3:

你设想的，并不等于用户知道的。

控件隐藏太深

上一节，我们理解了控制中心那种“拉出来的抽屉”，不应该有多于一页。苹果公司在 iOS11 中意识到并且改正了这个问题，却并没有解决“控件隐藏太深”这个更深层次的问题。恰恰相反，iOS11 增加了某些控件的“逻辑深度”，并且大量采用“重按”（force touch）操作来调出这些控件，这使得它们比以前还更难发现。

其中一个控件，就是那个用于选择播放设备的菜单，下面我们把它叫做“媒体输出控件”好了。你开始播放音乐，结果发现声音从 Apple TV 出来到电视里去了。你想让它从 iPhone 的扬声器直接播放。在 iOS11 里，你需要做哪些事情呢？



1. 从屏幕下边沿，划起控制中心。（如图）
2. “重按”缩略版的媒体播放控件，展开得到完整的媒体控件。
3. 点击那个看起来像“电台”的按钮。下面的播放控件的位置，被换成了媒体设备列表。
4. 选择你想要的媒体设备。

（然后用户可能会发现播放，暂停之类的控制按钮不见了，不知道怎么弄回来）

你发现这样一件经常要做的事情，需要经过多少个步骤吗？实在有点太多了。我还没有算上你盲人摸象的时间，控制中心有好几个其它控件可以供你浪费时间。

用户完成任务需要进行的“操作”或者“判断”的数目，被我叫做“逻辑深度”。注意这种深度可以包含物理上的动作，也可以是心理上的判断。所以用户在 iPhone 上完成这个选择播放设备的任务，逻辑深度差不多是 4。

这里首要的设计问题，叫做“可发现性”（discoverability）。用户怎么知道某些控件“重按”之后会弹出东西？一个一个去试吗？如果他们根本没想过可以重按那些控件，那就永远不会发现它们后面还藏着东西了。

有一条重要的设计原则是，事物的外观应该给用户足够的提示，告诉他们可以做什么。可是我们发现 iOS11 的有些控件重按会弹出东西，有些却不会，而这两者在外观上并没有明显的差异。这种缺乏“标志信息”，把功能藏起来，模棱两可的作法，本身就是一种设计缺陷。

甚至有某些控件（比如锁屏开关）重按之后没有任何实际功能，却仍然装模作样抖动一下。这导致用户更加迷惑，到底哪些控件在重按之后有实际作用，而哪些只是在那里凑热闹的？碰巧重按了没有作用的控件之后，用户可能会失去信心，不再想去试其它的。

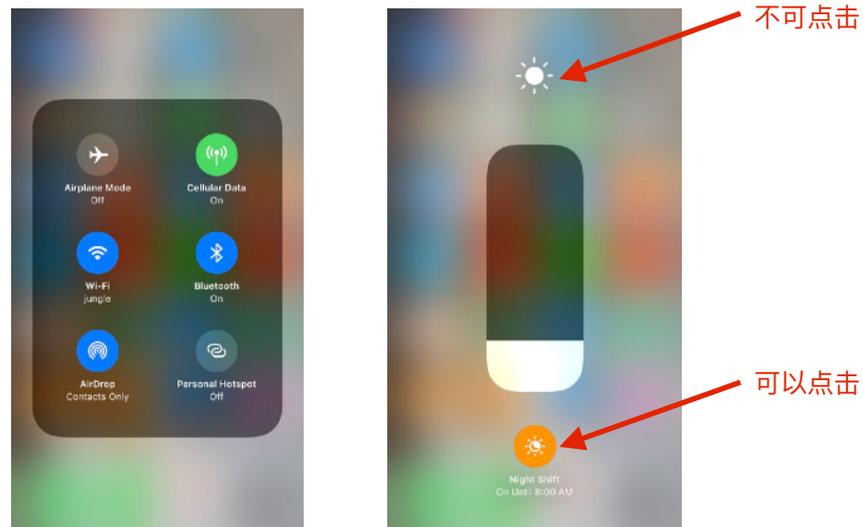
要知道，3D Touch 的“重按”并不是一个人们熟知的操作。iPhone 6 和之前的手机，根本就没有 3D Touch。即使有，因为要用力按，力道还要掌握好，而且功能太有限，很多人都不怎么喜欢用它。现在 iOS11 做出这么个新设计，必须要重按才能看到某些控件，实在有欠考虑。

然后，你怎么知道重按某个控件之后，就会弹出你需要的功能？并不是每个用户都那么会给事物分类的，他们又不是图书馆馆长。一般人发现需要改变声音的输出设备，会期望直接在控制中心看到它。他们不会想到这功能会隐藏在媒体播放控件重按之后弹出的面板里的一个按钮点击后出现的列表里。

实际上，“改变媒体输出设备”和“媒体播放控制”之间，本身就没有什么逻辑归属关系。用户对他们的操作可以是相互独立的，所以他们不会做出 iOS 设计者设想的那种归类。

iOS11 里面还有好几个隐藏过深的控件，它们必须通过“重按”小控件来获得。这些包括 AirDrop 和 Night Shift 的控制。你发现 Night Shift 控件有多难找到吗？它跟上面的亮度图标尺寸差不多

大，亮度图标不能点击，而 Night Shift 却是一个可以触摸的开关。人们很容易因为亮度图标不能



点击，所以觉得下面的 Night Shift 也是一个没有交互作用的图标！这是让人很容易混淆的设计。

过度的使用重按操作来安排控件的位置，不但使得它们很难被用户发现，同时也容易带来界面逻辑上的不一致。



我问你啊，为什么“媒体设备”的列表隐藏在媒体播放控件里面，而“Screen Mirroring”，这个完全类似的功能，被直接放在控制中心顶层，点击后弹出跟“媒体设备”几乎完全一样的菜单？同样是控制输出的导向（一个是屏幕，一个是媒体），为什么不把“媒体设备”做成跟“Screen Mirroring”

类似的按钮，放在它旁边？或许你已经发现，他们两个的图标都是很类似的，下面一个三角形，上面一个是屏幕，一个是声波。

* * *

那么有什么办法可以解决这种“功能深度”问题呢？哪些功能应该在顶层面板，哪些可以放到更深的位置呢？

在计算机科学理论里面，有一种概念叫做哈夫曼编码（Huffman Coding），可以被借用到这种场景。哈夫曼编码会给最常用的字符最短的编码。如果你把“逻辑深度”，也就是用户实现一个功能需要进行的操作总数（比如触摸屏幕的数目+思维判断的数目）作为编码的长度，那么你就可以构造出类似的编码。

最常用，需要在短时间做出操作的功能，应该被放在最顶层的界面里，甚至放到硬件按钮上。次要的，不急用的功能，位置可以稍微深一些。当然判断哪些操作是常用或者急用的功能，还是需要一定的经验和洞察力的。另外，处于较深位置的功能，它们的路径如何才能被用户发现，也需要考虑周到。否则很多用户会一直不知道它们的存在。

像之前的例子，媒体输出设备的控件，就应该处于控制中心的顶部。Night Shift 控件，Air Drop 等开关控件，也完全可以处于顶部。控制中心有那么大的空间可以用，不知道 iOS11 为何把其中一些藏起来，需要重按才能看到。

另外，应该避免给功能进行分类，使得一个控件位于另外一个里面。要记住一个道理：

道理4：

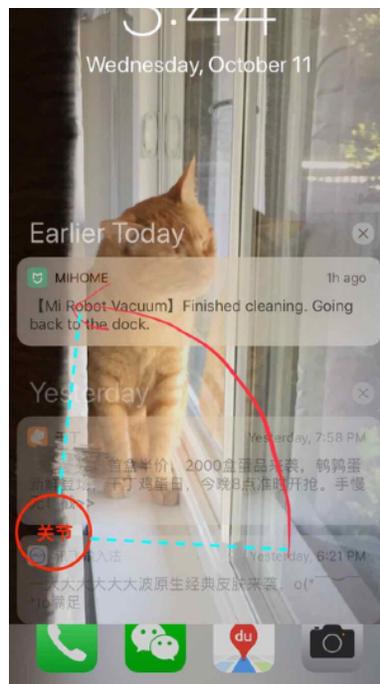
用户不是图书馆馆长，不要指望他们会对事物进行精确的分类。

容易混淆的划动操作

从 iOS10 升级到 iOS11，我发现锁屏界面出现一个新的问题，那就是原来左划可以迅速启动的相机功能，现在变得不可靠了。经常是左手拿出手机，单手往左一划，想启动相机，结果发现屏幕向上跳了一下，然后又弹回去了，相机没有启动。再划一下还那样。

在 iOS10 里面，这个问题不是那么明显，或者不存在。为什么 iOS11 之后出现这个问题呢？这不是一个 bug，它是一个设计错误。

观察之后你会发现，原因是 iOS11 给锁屏界面增加了一个“上划快捷操作”，用于调出之前看过的提示信息（notification），结果“上划”和“左划”产生了冲突。用户本来想“左划”，结果不经意的会先“上划”，结果导致了混淆。



用户本来想左划，为什么结果会先上划呢？经过简单的“人体工学”分析，你会发现单手进行“左划”操作的时候，拇指的运动方式是绕着大拇指关节画一条弧线，而不是一条直线。根据手握住手机的角度，拇指划动的“初始方向”可能是偏左，也可以是偏上。（如上图所示）

问题就在于，iOS 不看你“总体”的划动方向和角度，不去根据“手势”揣测你到底想做什么，而只是看“初始”方向。所以你的手指如果开头往斜上方运动，那么 iOS 就认为那是“上划”，认为你想

调出看过的提示信息。尽管手指运动的圆弧很快转为向左，iOS 却不改变最初的判断。它认为你转而向左就是想取消已经开始的上划操作，所以屏幕就弹回去了。

总结一下，iOS11 的问题出在哪里呢？

1. 给锁屏界面加入了太多不同功能的“快捷划动操作”，而手指划动的方向本来是一个不大可靠的信息，很容易引起混淆。左划启动相机是很有用的，因为照相往往对时效性要求很高，错过时机就没有了。然而上划调出看过的提示信息，就不是那么重要。

用户很少想翻出已经看过的提示。如果要看，他们完全可以先解锁手机，再拉下提示面板。这个操作不需要那么快。加入可有可无的上划操作，结果让本来很关键的“快捷相机”变得不可靠，是得不偿失的。

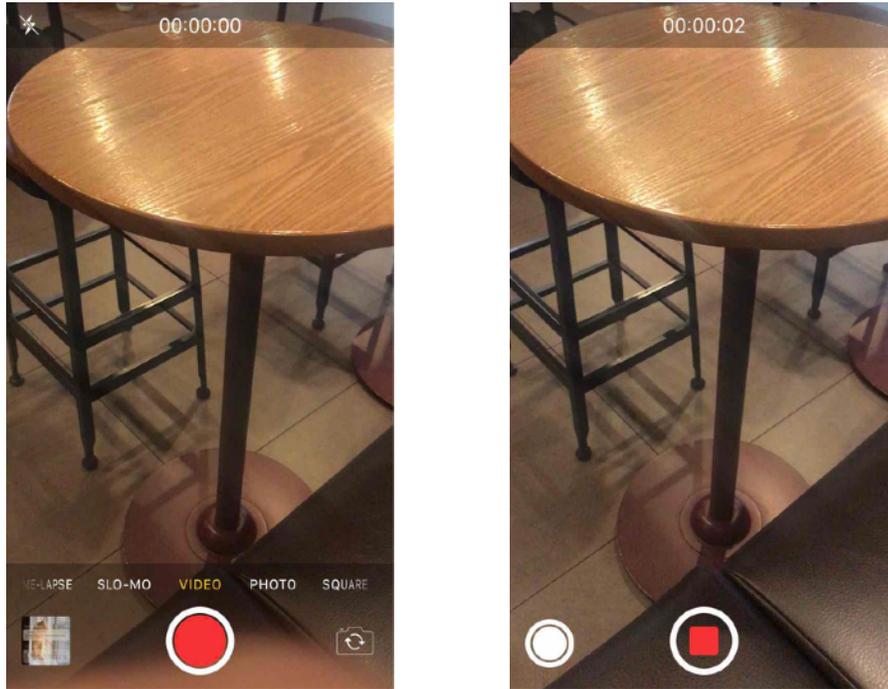
2. iOS11 不但加入了可有可无的“上划”操作，还加入了完全没有实际作用的“下划”操作。这个下划操作的“功能”，只是把整个屏幕往下拉动一点，一放手它就弹回去…… 加入这种毫无用处，纯粹视觉效果的功能，结果让重要的功能变得不可靠，是非常可惜的。

道理5:

不要为了加入可有可无的功能 而损害关键功能的可靠性。

摄像启动了没有？

iPhone 还有一个让用户容易混淆的设计，那就是它的摄像（video）按钮。不仅新用户和老年人容易弄错，连很多 iPhone 的资深用户，有时候也会因此出问题。



打开相机，切换到摄像界面。摄像按钮是一个白色圆圈，中间套着一个红色的圆形（表示摄像，常见符号）。点击圆圈启动摄像，红色圆圈变成了红色弧边方形（表示停止，常见符号）。外面的白色圆圈保持不变。

摄像的启动和停止按钮都是大家熟悉的符号，那有什么问题呢？问题就在于表示“启动”的圆形，长得太像表示“停止”的方形了。尺寸差别不大，都是一样的红色，方形还带有弧边，而且外面的白圈保持不变。所以不仔细看的话，很多人察觉不出这两者有什么差别。

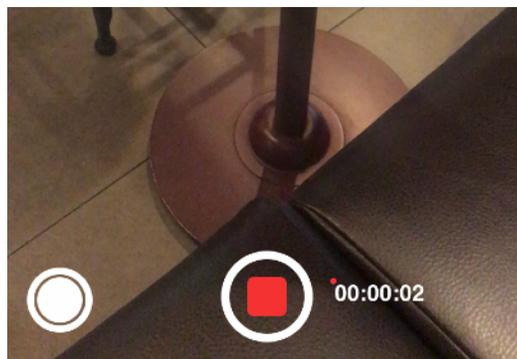
说一个真实的例子吧。我见过有人想拍视频，按了一下启动按钮。按钮已经变成方的，摄像已经启动，可是因为按钮视觉差异太小，所以他没察觉到，以为自己没按中启动按钮。所以疑惑了一秒钟之后，他又按了一下这按钮，当然这时摄像就停止了。可是他仍然举着手机，以为自己在拍视频。

精彩节目结束的时候，他又按了一下按钮，以为自己拍完了，而其实他启动了下一段视频。最后他按了 Home 键，所以这第二段视频也停止了。所以他拍了两个视频，每个只有一秒钟。打开看的时候，他还不知道到底发生了什么！

可能有人会说：“这人怎么那么笨，他不会看看上面的时钟有没有走吗？没有走就说明没有在拍啊！”这些人只是忘了自己“当年的痛”而已。

事实是，很多人都不看上面的时钟。为什么要看那里呢？人家在摄像呢，所以要看镜头里的东西。以时间有没有走来判断摄像是否在进行，这个信号太不直接，太隐晦了。要达到人性化，我们应该使用直接而明显的提示信号，而不是间接而隐晦的。

如果你真要用时钟作为“进行中”的标志，你至少可以把它放在按钮旁边，而不是最上面。我来冒充设计师给你画个图吧，就像这样：



用户点击按钮，旁边立即出现计数的时钟，还有个红点在闪，这是更加直接的提示。或者你可以在白色圆圈里加入旋转的动画效果，表示摄像正在进行，也是一个办法，就像 Time-Lapse 界面里的那种。

当然更好的设计，还是应该增加两种按钮的视觉差异。使用不同的颜色，更大差异的形状，或者去掉外面那层白圈……很多方式都可以达到这个目的。

出现这个设计错误的原因，我觉得是因为 iPhone 的设计者很在乎“外观”，却没有考虑到“功能”。他们觉得弧边的方形好看，外层的圆圈跟其它功能界面（照相，慢镜头等）的外形一致，所以就做成那样了。他们没有意识到，这个方形跟圆形的视觉特征太接近了。

有些人可能想说，外观和功能是鱼和熊掌，是矛盾的，所以你不能要求什么都做好。其实这两者不是矛盾的，你确实可以达到更人性化的功能，同时也有很好看的外观。我们不应该找借口损失其中的某一个。

道理6:

界面的视觉差异很重要。

总结

在这份报告中，我们指出了 iPhone 的设计中存在的一些问题。这些当然不是 iPhone 的所有问题，我们只希望这些例子起到抛砖引玉的效果。这些问题不是 iPhone 所特有的，它们完全可以存在于很多其它手机的设计里面。这些例子可以让我们认识到这些问题，以及它们背后的原理。

我们希望这份报告可以给手机的设计者，程序员提供一些有益的线索，帮助他们设计出更好的手机及其它产品。同时也希望用户们意识到，他们遇到的很多产品的问题，并不是他们自己的智商有问题，而是手机厂商的设计错误。用户们应该停止责备自己，应该向手机厂商反应他们遇到的这些问题。这样我们可以帮助整个社会改善各种产品的设计。