火龙果•整理
uml.org.cn

iOS 人机界面指南（二）：设计策略

ISUX 原创翻译

2014.10

目录

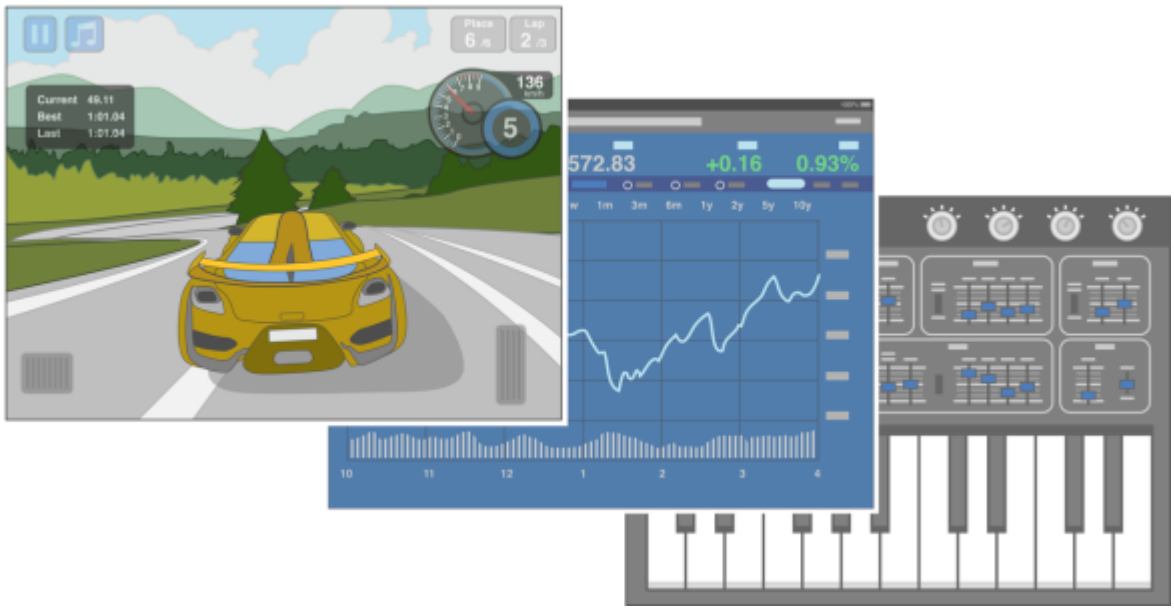
2 设计策略（Design Strategies）.....3
2.1 设计原则（Design Principles）.....3
2.1.1 美学完整性（Aesthetic Integrity）.....3
2.1.2 一致性（Consistency）.....3
2.1.3 直接操作（Direct Manipulation）.....4
2.1.4 反馈（Feedback）.....5
2.1.5 隐喻（Metaphors）.....6
2.1.6 用户控制（User Control）.....7
2.2 从概念到产品(From Concept to Product).....7
2.2.1 定义应用（Define Your App）.....7
2.2.2 为任务量身订制界面（Tailor Customization to the Task）.....9
2.2.3 原型与迭代（Prototype & Iterate）.....11
2.3 案例学习：从桌面到 iOS（Case Study: From Desktop to iOS）.....12
2.3.1 iPad 版 Keynote 应用（Keynote on iPad）.....12
2.3.2 iPhone 版邮件应用（Mail on iPhone）.....14
2.3.3 iOS 系统内的网页内容（Web Content in iOS）.....16

2 设计策略（Design Strategies）

2.1 设计原则（Design Principles）

2.1.1 美学完整性（Aesthetic Integrity）

美学完整性并不评判应用的视觉设计，或者用来描述应用的风格特征。美学完整性是指一款应用的视觉表现和交互行为在与功能结合后所传达出的整体一致性。



人们关心应用是否提供了应有的功能，但是也会潜移默化甚至是很直接地被应用的视觉表现和交互行为所影响。举个例子，一款协助用户完成任务的应用，可以通过使用精美而又无干扰的装饰性元素、标准的控件和可预期的交互行为来帮助用户聚焦在任务本身上。这样，应用就能传达出清晰一致的信息，使得人们信任它。但是，如果应用使用干扰的、琐碎或随意的 UI 来呈现任务，那么人们可能会对其可靠性和可信赖度产生怀疑。

另一方面，在沉浸式应用中—例如游戏—用户期待惊艳的视觉表现，为用户带来乐趣和刺激，并鼓励用户进行探索。人们不希望在游戏中完成严肃的或生产力式的任务，他们会期待游戏的视觉表现和交互行为能够整合进任务当中。

2.1.2 一致性（Consistency）

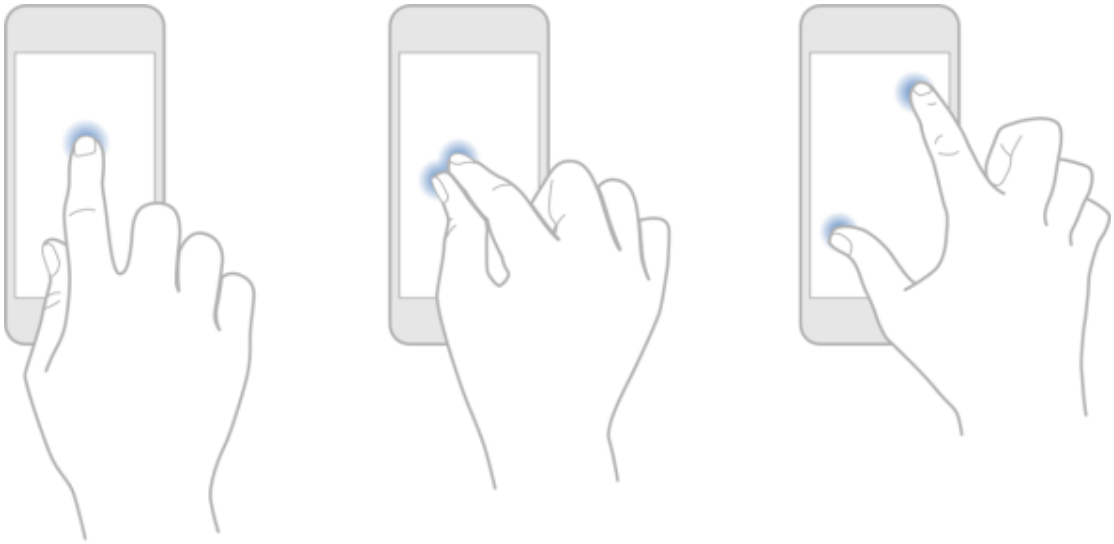
一致性可以让人们在一款应用中的不同部分甚至不同应用间复用他们的知识和技能。一款具备一致性的应用不应盲从地复制其他应用，也不应在风格上一成不变。应用应专注于让人们觉得舒适的标准和范例，并提供应用内部统一的体验。

在决定一款 iOS 应用是否要遵守一致性原则时，请思考如下问题：

- 应用是否和 iOS 标准一致？是否正确地使用了系统提供的控件、视图和图标？是否以用户所预期的方式整合了设备的特性？
- 应用是否内部统一？文案是否使用了一致的措辞和风格？同样的图标是否表意相同？在不同的位置执行同样的操作时，人们是否能能预期会发生什么？
- 应用是否和先前的版本保持一致？条款和意义是否保持不变？基本概念和主要功能是否发生了变化？

2.1.3 直接操作（Direct Manipulation）

当人们不再使用一堆控件进行操作，而是直接操作屏幕上的对象时，他们能更集中精力完成任务，也更容易理解这些行为所产生的结果。



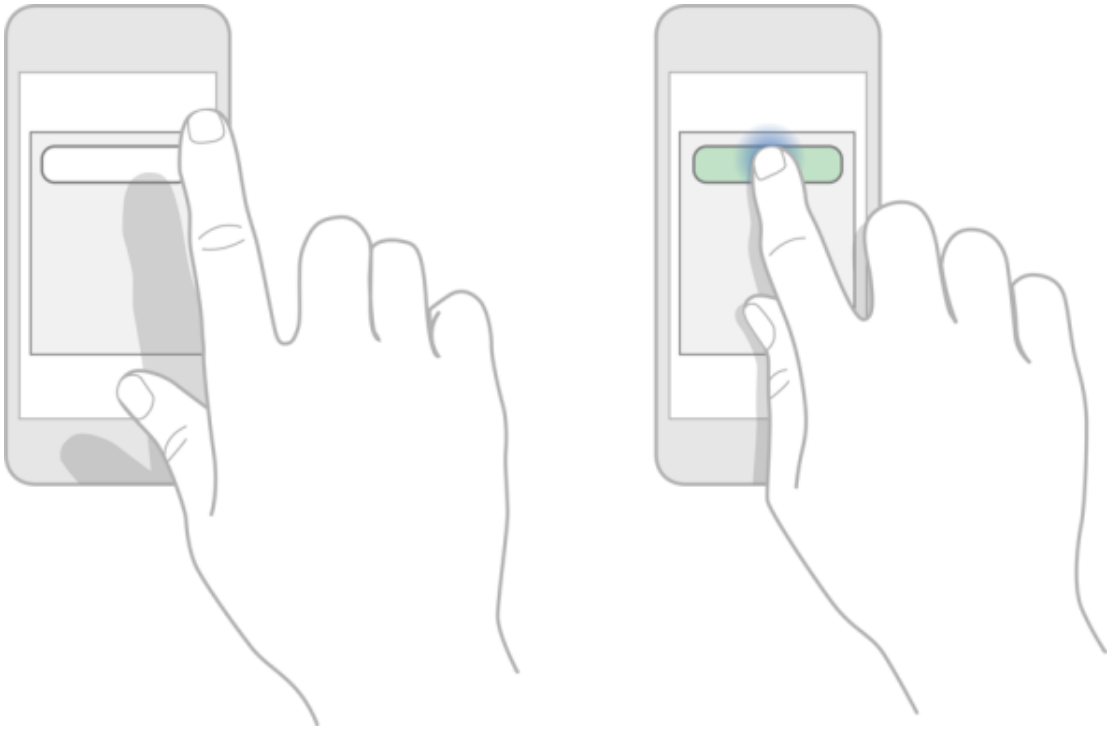
使用多点触摸界面，人们可以通过捏合操作来直接放大和缩小图片或文本内容。在游戏中，玩家应该直接移动和作用在屏幕上的对象。例如，游戏中可能会显示密码锁，用户可以通过转动它来打开。

在一款 iOS 应用中，如下情况中人们应该能够进行直接操作：

- 旋转或者移动设备来影响屏幕上的对象
- 使用手势来操作屏幕上的对象
- 显示即时可视的操作反馈

2.1.4 反馈（Feedback）

反馈可以明示人们的行为，展示操作的结果，并更新于任务进程之中。



iOS 内置的应用对用户的每个行为都提供了可感知的反馈。当人们点击列表项和控件时，它们会被临时高亮，并会在操作过程中持续一段时间，以此展示控件被执行的过程。

精细的动画会给人们带来有意义的反馈，帮助阐明行为的结果。例如，列表中新增一项时的动画可以从视觉上帮助人们发现列表的变化。

音效同样可以为人们提供有效的反馈，但不应成为唯一的反馈方式，因为人们不一定能听到。

2.1.5 隐喻（Metaphors）

当应用中的虚拟对象和交互行为与用户已经熟悉的体验相似时—无论这些体验是来源于真实或数字生活—用户就可以快速地掌握如何来使用这个应用。

当应用使用隐喻来传达某种用法或体验时，最好不要让隐喻突破所依赖的对象或交互行为本身的限制。（译者注：此处可理解为对于隐喻的使用应量力而为，不要过于牵强。）

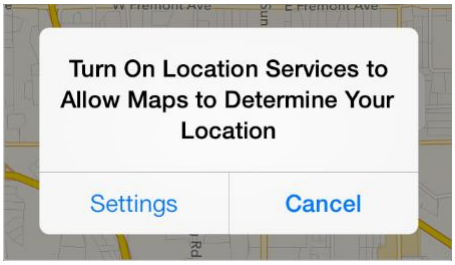
由于人们实际上是和屏幕进行物理上的交互，所以 iOS 应用有很大的余地来使用隐喻。iOS 中的隐喻包括：

- 移动分层视图来显示被遮挡的内容
- 拖曳、轻扫和滑动游戏中的对象
- 点击开关，滑动滑块，转动选择器

- 轻扫来翻阅书本或杂志

2.1.6 用户控制（User Control）

是人—而不是应用—发起和控制行为。应用可以对用户进行操作建议，对有危害的后果予以警示，但是不应替用户来做决策。好的应用会在给人们帮助他们避免不必要结果的能力时，找到正确的平衡。



当对应用的交互行为和控件都较为熟悉和可预期时，用户会觉得应用更易上手。那些简单直白的交互行为更容易被用户所理解和记住。

人们会希望在一个操作被执行之前有足够的机会来取消，也希望在执行一个不可逆的操作之前可以有机会来进行确认。最后，人们还会希望能够停止正在执行中的操作。

2.2 从概念到产品(From Concept to Product)

2.2.1 定义应用（Define Your App）

应用的定义是对应用主要功能和目标用户的简明具体的描述。

尽可能早地创建应用的定义可以帮助你将一个想法和功能清单转换为用户想要的条理清晰的产品。在开发过程中，可以使用定义来决定某些功能和行为是否合理。使用以下几个步骤来创建一个可靠的应用定义。

1、列出所有你认为用户可能喜欢的功能

可以直接进行头脑风暴。此时，你需要列出所有与产品中心有关的任务。不用担心清单太长，因为接下来会进行删减。

假设你一开始的想法是开发一个帮助人们购买食品杂货的应用。你可以思考在进行这项活动时，会涉及到哪些相关的任务，这些就是用户可能感兴趣的潜在功能。例如：

- 创建清单

- 查找食谱
- 比较价格
- 定位商店
- 给食谱做注释
- 查找可使用的优惠券
- 查看烹饪演示
- 探索不同的烹调方法
- 寻找某些食材的替代物

2、确定目标用户

现在你需要清楚地将你应用的用户与其他 iOS 应用的用户区分开来。确定在此情此景下，什么是对你的用户最重要的。在食品杂货例子中，你可能需要问问你的用户：

- 通常是在家里做饭还是更喜欢现成的食物
- 是忠实的优惠券用户还是认为优惠券没多大价值
- 喜欢寻找特别的食材还是喜欢基本食材
- 严格按照食谱做菜还是只把食谱当做灵感来源
- 喜欢少量多次购买还是一次性购买大量食物
- 希望能保留多个不同目的的清单还是只希望记录回家路上需要购买的几个东西
- 坚持使用固定的品牌还是会使用方便的替代品
- 习惯于购买固定的一些物品还是会按照食谱来购买

思考过这些问题之后，你可以从三个方面来描述目标用户的特征：喜欢按照食谱进行尝试，时常很匆忙，通常情况下很节俭。

3、根据目标用户过滤功能清单

如果在确定了一些用户特征后，你最终只得到了几个主要功能，那么你就已经步入了正轨：出色的 iOS 应用应该聚焦在帮助用户解决问题的核心任务上。

例如，即使第一步想出的那些可能需要的功能都是有用的，也不可能全都被第二步定义的目标用户认可。

当你在目标用户的使用情境下检查功能清单时，就可以判断你的应用应该聚焦在三个主要功能上：创建清单，获得并使用优惠券，获得食谱。

此时你就可以给出应用定义了，总结该应用做什么以及为谁做。食品杂货购买应用的定义可能如下：

"为热爱烹饪且节俭的用户订制的创建购物清单工具。"

4、不止于此

应用定义应该贯穿于整个开发过程，使用应用定义来确定功能，控件，措辞的合理性。例如：

当你想要新增一个功能时，问问你自己这对应用的主要目的和目标用户是否非常重要。如果不是，可以置之不理。例如，你已经确定了你的用户对大胆新颖的烹饪方法感兴趣，那么着重展示盒装蛋糕和现成的食物就不太合适。

当你考虑用户界面的外观和行为时，问问你自己你的用户更喜欢简单的、流线型的风格，还是有明显主题的风格。以用户目标为指导来完成你的应用，理解用户期望通过你的应用完成什么，例如完成任务的能力，快速找到答案的能力，深入综合内容的能力，或者获得娱乐的能力。例如，尽管你的食品杂货清单应用需要易于理解和快速上手，但你的用户还是可能倾向于一个有关食物的主题界面。

当你考虑应该使用怎样的措辞时，努力和用户对这个主题的专业程度相匹配。例如，尽管你的用户可能不是由专业的大厨组成，但你也可以肯定他们希望看到有关食材和技术专用的措辞。

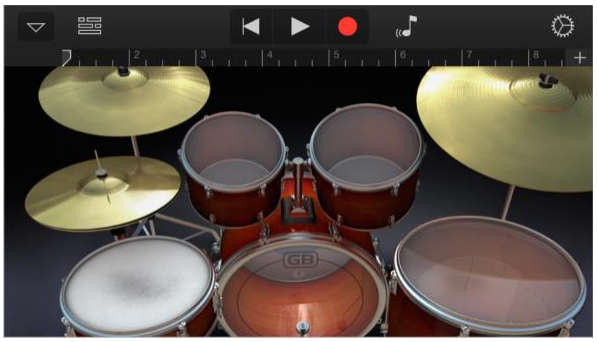
2.2.2 为任务量身订制界面（Tailor Customization to the Task）

好的 iOS 应用会根据清晰的目标和易用性来平衡用户界面的设计。为了达到这种平衡，要确保在设计阶段前期就考虑定制化。因为考虑品牌性、原创性和适销性通常会影响定制化的决策，所以专注于定制化怎样影响用户体验是有挑战性的。

可以由应用中的任务着手：用户执行这些任务的频率如何，在什么样的环境下进行？

举个例子，想象一个计算器应用使用的是精心设计的，充满艺术感的风格，并且使用了创新的布局来展示大家熟悉的计算器元素。这些像艺术品一样的细节渲染和创新的布局并不会影响用户去理解怎样点击按钮和查看计算结果。但是对于只想简单完成任务的用户，这种新奇的体验和美丽的界面很快就会失去效用，并且可能成为一种妨碍。

相反，GarageBand（随身录音室应用）可以不展示好看的、逼真的乐器来帮助用户制作音乐，但这样会使应用缺少身临其境的愉悦感。在 GarageBand 里，界面不只是向用户展示了如何使用，同样使得制作音乐的主任务更容易完成。



当你思考定制化如何增强或减弱用户完成任务的注意力时，记住以下几点：

定制总要有缘由。理想情况下，定制化的用户界面能促进用户完成任务并增强他们的体验。你最好尽可能地用任务驱动定制化决策。

尽量避免增加用户的认知负担。用户对标准界面元素的外观和行为都已经很熟悉了，所以他们不用停下来思考如何使用它们。当用户面对外观和行为与标准不同的元素时，他们就

失去了经验优势。除非你的独一无二的元素能够使任务更容易完成，否则用户很可能不喜欢被强制学习一些在其他应用都不通用的操作。

保持内在的一致性。你的应用中自定义元素越多，保持这些元素外观和行为的一致性就越重要。如果用户花费时间去学习了你创建的那些不熟悉的控件，那么他们会希望新学到的这些操作能够在整个应用中通用。

总是以内容为重点。因为标准元素很熟悉，所以它们不会分散用户在内容上的注意力。当你自定义用户界面时，注意确保界面元素不会抢走用户对内容的注意力。例如，如果你的应用允许用户观看视频，你可能选择设计一个自定义的重播控件。但是不管你用的是自定义还是标准的重播控件，都没有它是否在用户开始观看后隐藏，点击屏幕后出现来的重要。

在对标准控件进行重设计时再三思考。如果你不只是想自定义标准控件，而是想重设计，确保你的重设计能提供尽可能多的信息。例如，你设计了一个开关控件，它没有可以指明相反状态存在的信息，那么用户很可能意识不到这是个有两个状态的控件。

一定要彻底测试自定义的界面元素。在测试过程中，近距离的观察用户是否能预测你的元素如何使用以及是否能容易地与它们交互。例如，如果你创建的控件的可点击区域小于 44 x 44 点，用户点击时就会有困难。或者如果你创建了一个视图对点击和滑动的反馈不一样，确保这个视图提供的功能值得用户去额外关注交互的不同。

2.2.3 原型与迭代（Prototype & Iterate）

在你投入工程资源实现设计之前，创建原型来进行用户测试是个很好的主意。即使只有几个同事来帮你测试原型，你也会收获一些关于应用功能和用户体验的新鲜观点。

在设计的早期阶段，你可以使用纸面原型或者线框图去呈现主要的视图和控件，并且标明每个页面之间的跳转关系。你可以从线框图测试中获得一些有用的反馈，但是线框图的稀疏性有可能会误导用户。因为用户很难想象当线框被实际内容填满时体验会有什么样的变化。

如果你有一个可以在设备上运行的原型，那你可以得到更多有用的反馈。当用户能在设备上与你的原型进行交互时，他们能更容易发现应用中哪里功能不满足预期，哪里体验过于复杂。

创建可靠原型的最简单的方法是使用基于故事板（storyboard）的 Xcode 模板创建一个基础应用，然后使用一些类似于占位符的内容来进行填充。（**故事板（storyboard）**文件可以涵盖应用中的所有界面，并且包括界面之间的跳转关系。）接着，将这个原型导入到设备中，这样被测者就可以有一个尽可能真实的体验了。

你不需要在原型中提供大量的实际内容或者使每一个控件都可用，但是你确实需要营造足够的情境来保证真实的体验。并且需要在典型用户体验和非典型的边缘情况之间做好平衡。例如，如果你的应用需要处理很长的列表项，你的原型就不能只显示一两个条目。而

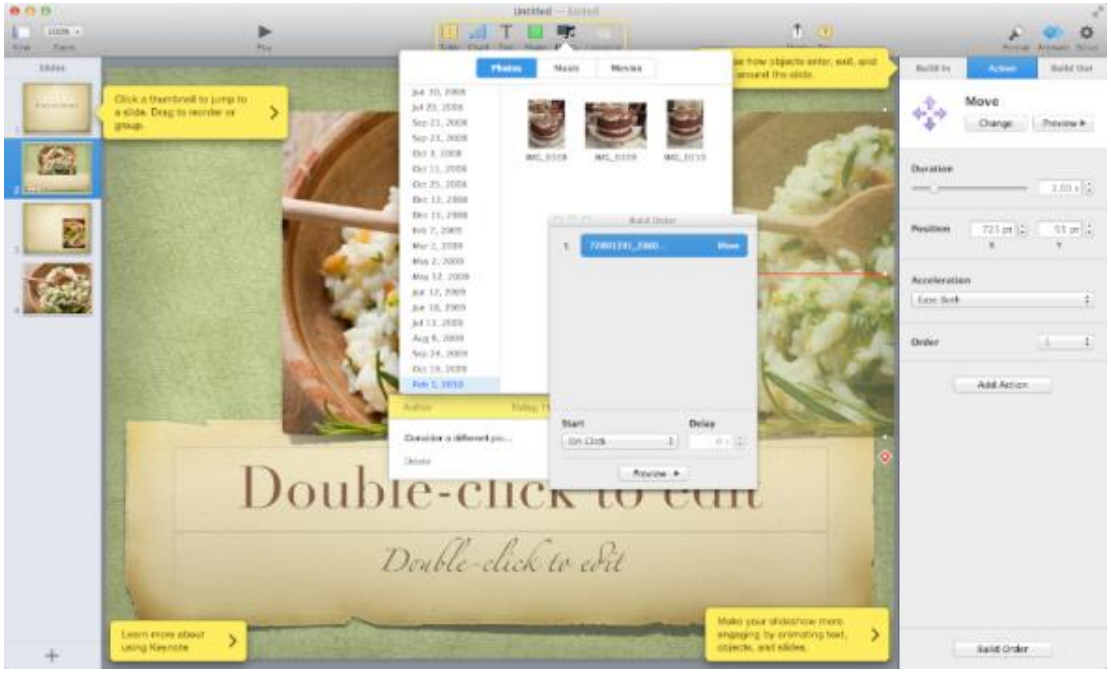
且在用户测试交互中，只要被测者能够点击屏幕上的一个区域进入到下一个逻辑页面或者完成主任务，那他们就可能会提供更有建设性的反馈。

当你使用 Xcode 应用模板来创建原型时，你可以免费使用很多功能，并且它可以相对容易地进行设计中的响应反馈调节。在你确定设计方案并投入资源进行实现之前，应该对原型进行多次迭代测试。想要开始学习 Xcode，请参考 *Xcode Overview*。

2.3 案例学习 :从桌面到 iOS(Case Study: From Desktop to iOS)

2.3.1 iPad 版 Keynote 应用（Keynote on iPad）

桌面版的 Keynote 应用是一个十分强大而又灵活的应用，可以创建非常优秀的幻灯片。人们喜爱 Keynote 将简单易用与良好的细粒度的操作结合进而控制无数精确细节的方式，如动画和文本属性等。



iPad 版的 Keynote 提取了 Keynote 桌面版的核心要素，并通过创造以下的用户体验使它在 iPad 上更舒适：

- 专注于用户内容
- 在不削减功能的基础上减少系统的复杂性
- 提供有用而又令人愉悦的快捷操作
- 延续桌面版本的体验
- 利用动人的动画提供良好的反馈与交流

Keynote 用户能很快理解如何使用 iPad 版，是因为它使用了 iPad 原生的范例，符合了用户对功能上的预期。新用户可以用简单、自然的方式直接操控内容，所以可以很容易学会如何使用 iPad 版的 Keynote。

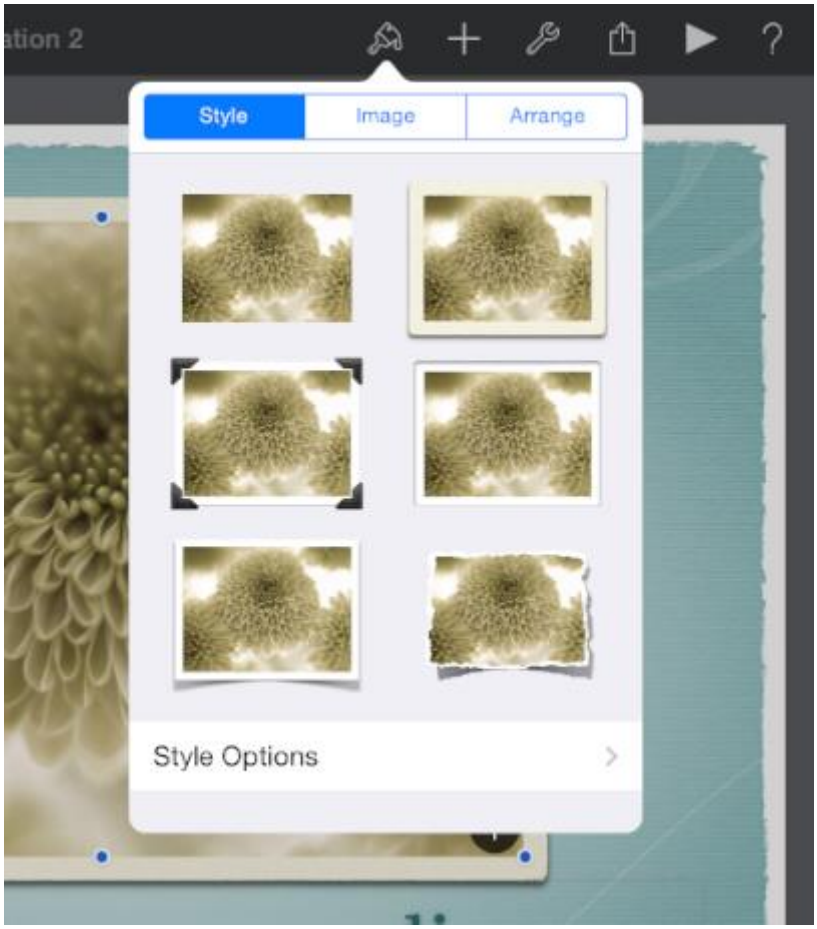
Keynote 从桌面版向 iPad 版的转变是基于从细节到深层的大量修改和重新设计的。这些都是一些最明显的适应性：

流线型的工具栏

工具栏中只有少数的元素，但是它们是用户在创建内容时所使用的全部功能和工具的统一入口。

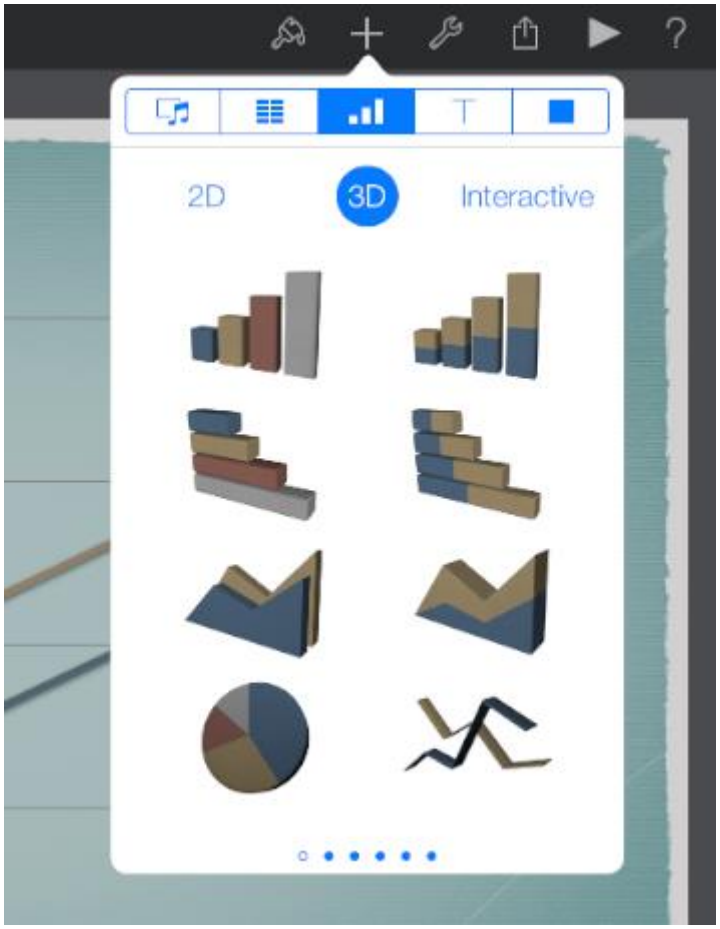
简化并优先响应用户焦点的检查器

iPad 版的 Keynote 能自动侦测各种工具，能通过对用户需求进行分类以修正被选择的对象。（译者注：特别是根据当前的操作对象而有限选择某些工具。）通常，人们可以在第一检查器视图中完成他们需要的所有修改操作。如果他们需要修改那些不常用的属性，他们可以下拉另一个检查器视图来进行。



丰富的预设样式集

人们可以利用预设的样式很简单地改变对象（如表格或图表）的外观或者感觉。除了颜色之外，每个集中，例如表格的标题和轴区分标识等的预设属性都被设计得与整体的主题和谐一致。

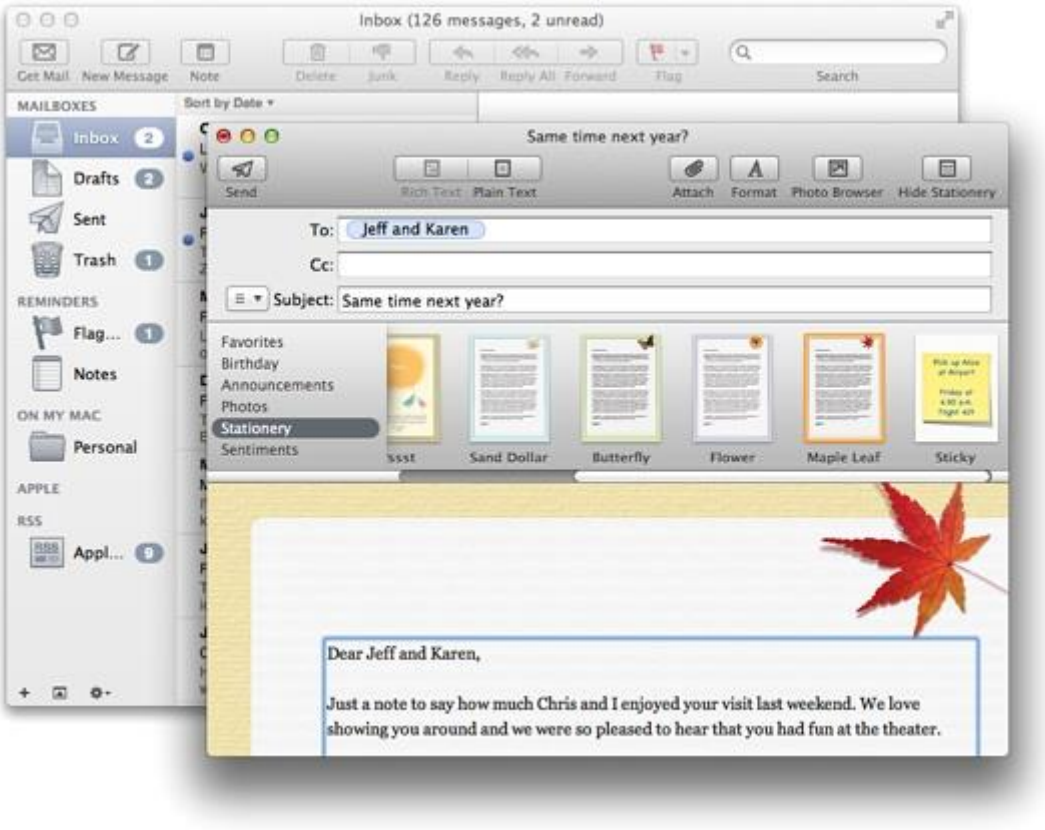


直接操作内容，丰富有意义的动画

在 iPad 版的 Keynote 中，用户可以拖动滑块到一个新的位置，可以扭动旋转一个对象，也可以轻击图片来选中它。iPad 版 Keynote 的响应动画进一步加强了这种可直接操作的印象。例如，用户在移动某个滑块时它通常会暂停，而当它被放置在一个新的位置时，环绕在周围的滑块将会向外扩散给它留出空间。

2.3.2 iPhone 版邮件应用（Mail on iPhone）

邮件应用是 OS X 中一款好用而又广受好评的常见应用。它也是一个很强大的程序，可以允许用户撰写、接收、分类和存储邮件，追踪行动和事件，也可以编写备忘录和邀请等。桌面版的邮件应用通过一系列的窗口提供了这些强大的功能。



iPhone 版的邮件专注于桌面版邮件的核心功能，帮助人们接收、撰写、发送和组织他们的信息。为了塑造移动 iPhone 版的邮件应用，将这些功能浓缩在为其量身定制的界面之中，做了如下的工作：

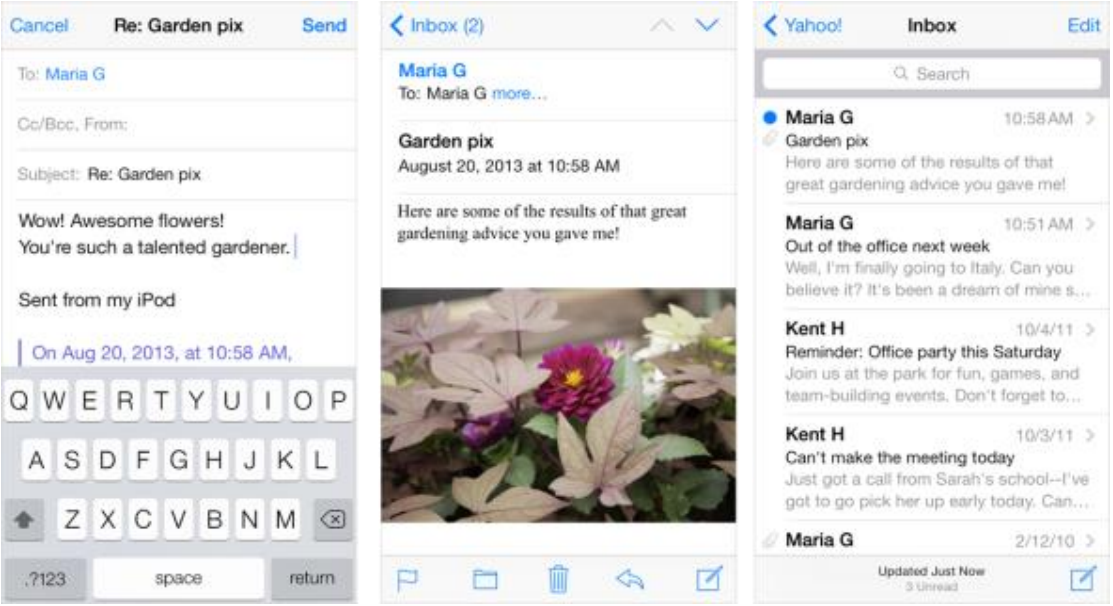
- 将人们的内容前置和居中的合理化呈现
- 专为处理不同任务而设计的不同视图
- 易于浏览并符合认知的信息结构
- 适时提供强大的编辑和组织性工具
- 传达动作和提供反馈的微妙且动人的动画

（我们）必须明白相较于桌面版的邮件应用，iPhone 版的邮件应用不是（译者注：或者说并不需要是）一个更好的应用，而是为移动端用户重新设计的邮件应用。iPhone 版的邮件应用专注于桌面版的功能子集并将它们呈现在一个吸引人的精简界面之中，据此为移动端的用户提供了核心的邮件体验。

为了使邮件应用的体验能适应移动场景，iPhone 版的邮件应用在几个关键的方面革新了用户界面。

直接、高度专注的页面

每个页面显示了邮件应用体验的一个方面：账户列表、邮箱列表、消息列表、消息查看和编辑视图。用户可以在一个屏幕内滑动查看完整的内容。



简单、可预见的导航

通过每屏的一次点击，用户可以逐层展开通用内容（账户列表）进入具体页面（一封消息）。每个页面会显示一个标题用以指示用户所在的位置，以及一个返回按钮用以更容易地回溯到他们之前的步骤。

需要时即可获取的、简单的点击性控件

基本上在任何场景之下，编写邮件和查阅新邮件都是人们首要希望进行的操作，因此 iPhone 版的邮件应用保证了这两个功能在多个页面中都可以便利地进行。当用户查看一封消息时，就会显示诸如回复、移动和删除等对消息的操作。

针对不同任务的不同类型的反馈

当人们删除一封消息时，它会动态地进入垃圾桶图标中。当人们发送一封消息时，可以看到它的发送过程；而当发送结束时，人们可以听到一个特别的声音提示。通过消息列表页面工具栏的副标题，用户通过简单一瞥就可以查看邮箱上次更新的时间。

2.3.3 iOS 系统内的网页内容（Web Content in iOS）

iOS 版的 Safari 应用在 iOS 设备上提供了出众的移动网页浏览体验。人们喜欢阅读清晰的文字和图片，也希望能通过旋转设备或者捏合和点击屏幕来调整视图。

基于标准建立的网站可以在 iOS 设备上显示得很好。特别是那些能侦测设备并不需要插件的网站可以同时在 iPhone 和 iPad 上都表现得很好，两者之间不会需要太多的修改，即使有也很小。

除此之外，成功的网站应具备以下的典型性：

- 如果页面宽度需要匹配设备宽度，可以设置合适的视窗（viewport）来适应设备
- 避免 CSS 中固定的定位，以便当用户缩放或拖动页面时内容无法被移出屏幕
- 拥有一套基于触控操作的用户界面，而不是依赖基于传统点击操作的交互

有时候，额外的一些修改可以（使页面）更合理。例如，在 iOS 系统中，很多网页应用会设置合适的视窗（viewport）宽度并通常隐藏 Safari 的 UI。如欲了解更多如何进行这些修改，参见 *Safari Web Content Guide* 章节中的 Configuring the Viewport 和 Configuring Web Applications。
网站也可以通过其他的方法适配桌面网页体验到 iOS 端的 Safari 浏览器中：

使键盘适应 iOS 端的 Safari

当键盘和格式辅助信息出现时，iPhone 上的 Safari 应用会将你的网页显示在 URL 地址下方和键盘与格式辅助信息上方。

使弹出式菜单适应 iOS 端的 Safari

在桌面版的 Safari 应用中，弹出式菜单会包含很多选项，就如在其他 OS X 应用中一样。在必要的情况下，菜单展开后可以超出应用窗口的边界以显示其中的所有选项。在 iOS 版的 Safari 应用中，弹出式菜单由原生的元素所呈现，这样能提供更好的用户体验。例如，在 iPhone 上，弹出式菜单会出现在选择器（**picker**）当中，选择器里会一个用户可选择的选项列表。（欲了解更多选择器控件的内容，可以参见 Picker。）