

Apple Watch 人机界面指南（中文版）

@微博 UDC-M

翻译说明

本文译自 Apple 官方推出的设计指南《Apple Watch Human Interface Guidelines》
(2014-11-19 版本, 原文地址见: <http://t.cn/Rz255g7>)。由于时间和水平有限, 翻译中难免存在错误和疏漏, 若有问题, 欢迎联系@微博 UDC-M 反馈建议

微博无线设计团队

2014 年 11 月 21 日



微博UDC-M

扫一扫二维码图案, 关注我吧

目录

UI 基础	4
为 Apple Watch 设计	4
应用间的适配	5
Glances	6
通知	7
模态界面（Modal Sheets）	9
布局	10
颜色和字体	12
动效	14
品牌	14
UI 元素	15
标签	15
图像	16
组	17
表格	17
按钮	18
开关	19
滑块	20
地图	20
日期和计时器	22
菜单	23
图标和图像设计	25
图标和图像尺寸	25
菜单图像	27

UI 基础

为 Apple Watch 设计



Apple Watch遵循以下设计宗旨：

个性化。由于Apple Watch是为穿戴设计，它的UI将贴合使用者的生活。在佩戴者抬起手肘时将显示时间和新的提示。数码化的输入——特别是心率测算以及草图功能，使得不可思议的新型个性化沟通方式成为可能。加速度传感器和一个心脏速率传感器提供着使用者每天行为的个性化信。在以往，没有任何Apple设备能够如此贴近用户，当你为Apple Watch设计应用时，重点考虑这种贴近用户的联系。

整体性。Apple Watch是设计用来模糊软硬件之间界限的产品。数码旋钮（Digital Crown）是一个用于微调的硬件，同时用来实现精妙的软件界面导航。Taptic发动机可产生与屏幕上的通知和交互行为相关联的微妙物理反馈。并且，由硬件驱动的物理手势——Force Touch，揭示了软件控制方式的另一个新维度。Retina显示屏使得UI设计的物理边界也被有效地模糊至不可见。周到的应用设计应该致力于软硬件之间无缝的体验。

轻量型。快捷和轻量级的交互可以最大化利用手腕上有限的呈现面积和位置。能够快捷简单地接收和取消信息，在私密性和可用性上都是有益的。比如，消息的缩略预览（The notification Short Look），被设计用来提供通知的最小化呈现，只在佩戴者想要保持进一步参与的情况下才展示更多信息，并提供易于确认

和轻扫的缩略信息浏览界面。为Apple Watch设计的应用应该尊重佩戴者的使用情境：在小尺寸表面上简易，随时地呈现一切。

Watch app应该是对于你的iOS app的补充，而不是替代。如果你用分钟计量iOS应用的交互敏捷度，那么你应该以秒计量Apple Watch上的交互敏捷度，也就是说，Apple Watch上的一切交互行为都应该更简洁，一切交互界面都应该更简单。

应用间的适配

Apple Watch 应该与佩戴者的 iPhone 一起配合着让你的应用更好地运行。

界面风格

Watch应用支持2种方式的导航：

层级导航（Hierarchical）。与iOS的层级导航一致，适合于多级信息的呈现。在多级导航应用中，用户通过在每个界面中做一次选择进行导航，直到到达目标界面。如果需要到达另一个目的界面，用户通常需要逐级返回或全部返回至初始界面，做不同的选择。

对于复杂的应用来说，层级导航在传统意义上通常优于平级，分页导航模型。

基于页面间的导航（Page-based）。分页界面能够让用户通过横滑来导航不同页面的内容。基于页面的导航方式适用于简单数据模型的应用，并且每一页的数据模型彼此并不紧密相关。

位于每个分页界面下方的点状指示器显示了用户在所有分页中的当前位置，最小化呈现分页总数量，从而简化导航。

你不能混合使用层级导航和分页导航。在设计构思阶段，你应该选定最适合你的应用的内容与设计的导航方式。

使用这2种导航形式都可以模式化呈现内容，标准的模式化界面能够让用户专注地完成任务或获取信息，但是暂时阻止用户与其余应用进行交互，更多信息参见[模式化表单（Modal Sheets）](#)

用户交互

基于行为的事件（Action-based events）。单击手势是用户与你的应用进行交互的主要手势，列表栏，按钮，滑块以及其他控件均将通过单击被激活。这些单击行为将被包括进你的WatchKit代码中。

手势（Gestures）。你不能为你的应用单独定义新的手势。而是应该采用系统定义好的手势，并利用这些手势去执行标准的行为：

- 垂直轻扫，滚动内容界面
- 在分页导航界面中，横滑切换上一页下一页
- 至左边界边缘的横滑将自动返回初始导航界面
- 点击表明着选择或交互行为，点击行为由系统进行处理，报告给你的WatchKit的行为方法库

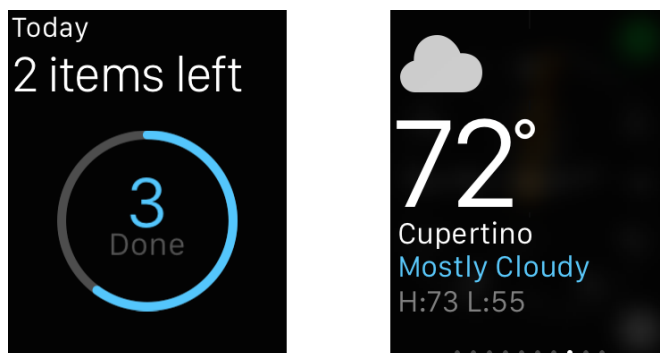
Apple Watch 不支持捏放等多指手势。

按压触摸（Force Touch）。小屏幕可以容纳的控件有限，因此Apple Watch引入了一种全新的交互模式：按压触摸。和感应点触一样，Retina屏幕也能感应压力。按压触摸用于呈现与当前内容相关的情境操作菜单（如果有的话）。

数码旋钮（The Digital Crown）。为微调设计，在不占用Apple Watch屏幕面积下提高滚动浏览效率，尤其便于翻阅长页面。

Glances

在Apple Watch中，Glance是app一系列内容的预览视图，理想情况下，它是实时的并与情境相关的。整体来看，Glances是Watch所有app提供的有意义信息的浏览集合。Glances是拉取，而不是推送，所以不同于推送到设备上的通知，Glances站在使用者的角度进行拉取信息。



Glances是：

- **基于模板的。** Glance有很多上部呈现或下部呈现的模板版式，使用Xcode选定一个你需要的模板版式，并让你的内容设计遵循相应模板版式的规范。
- **不支持滚动的。** 所有内容必须在1屏显示。
- **只提供阅览的。** 点击Glance的任何地方都将打开app。
- **可选的。** 不是所有app都需要Glances，并且用户可以选择需要显示的Glances。

Glances 底部的空间需预留给分页点状指示器。

基于用户当前使用情境去考虑Glance。 不实时以及不相关的信息将会降低Glance的可用性，尽可能多地考虑与用户实时相关的时间和地理位置信息。

Glances可以深层关联相应的app。 Glances能够通过切换告诉app当用户点击呈现过哪些缩略视图。app可以根据这些信息呈现不同的界面或者灵活呈现现有的界面。

缩略视图必须显示对用户最有用的信息。 不要将Glance简单当作启动你的应用的一种方式。

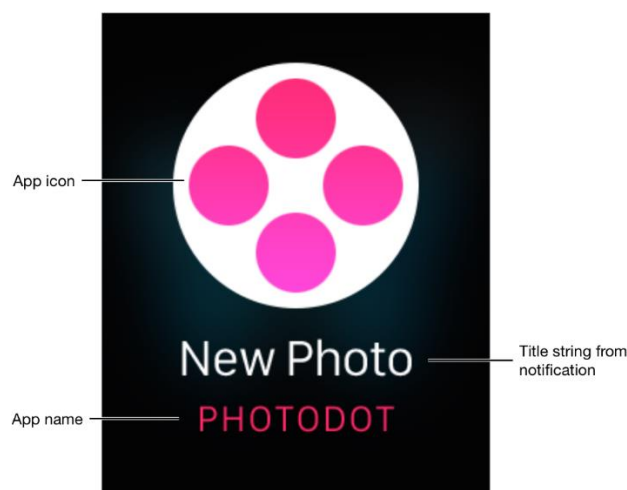
通知

Apple Watch中的通知快捷便利，并有着轻量级的交互方式，体现在2个方面：**Short Look**和**Long Look**。**Short Look**会在一个实时的本地或远程通知需要呈现给用户时出现。**Short Look**提供最精简的信息——一定程度上保护用户的隐私。如果佩戴者放下TA的手腕，**Short Look**便会消失。**Long Look**会在佩戴者手腕一直处于抬起状态或者佩戴者点击**Short Look**时出现。**Long Look**状态下会提供更多的细节信息和更多的功能——并且它必须通过佩戴者执行取消操作后消失。

仔细考虑你发送给用户通知的频率。 用户可能会认为过于频繁的通知是烦扰的，并因此禁用你的app通知在Watch上呈现。始终要确保呈现出的信息是用户想要的。

Short Look 通知

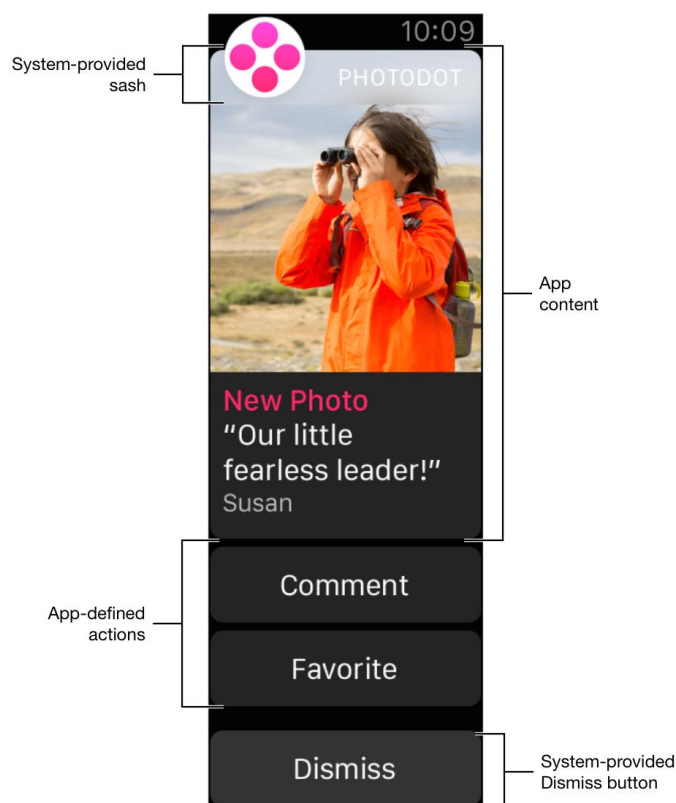
Short Look让用户知道通知来自于哪个app，并精简地呈现出来。**Short Look**界面是基于模板的，其中包含通知来源App的名称，图标，通知标题，系统将会以app的关键色展示app名称。



让通知标题简短专注。用于呈现通知标题的空间有限，因此尽可能让通知标题简明达意。标题字段不需要提供太多通知详情，只需要提供通知的内容的概要说明。

自定义的Long Look通知

Long Look将提供通知的更多细节信息。系统提供默认的Long Look呈现样式，但是各个app可以根据自身的品牌，代表图形去自定义Long Look的呈现样式。Long Look的结构对于所有app都是一致的，系统会提供一个浮于上层的边框用于承载app的图标和名称，并在底部提供取消按钮和app定义的操作按钮。顶部app图标和底部操作按钮区域中间用于呈现app内容。



app内容可以与系统提供的承载app图标边框位于一行，通过高斯模糊蒙层呈现，也可以在其下方开始呈现。一般高斯模糊蒙层呈现适用于图片或其他图形化内容。对于文本型的通知内容，使用Xcode的偏移通知内容选项来使得app内容正好从app图标边框下方开始呈现。



对于自定义的Long Look界面，app必须提供一个静态界面，在此基础上可以选择配置一个动态界面。动态界面比静态界面更加灵活呈现信息，但是请确保静态界面和动态界面呈现同样类型的通知内容。静态界面将为暂时读取不到的动态界面提供占位。

Long Look视图下的通知能够最多呈现4个自定义操作按钮。 Apple Watch会根据你提交的通知中自定义操作按钮类型呈现相应的操作按钮。

取消按钮始终存在。 取消按钮会在最多数量为4的操作按钮下存在。

想知道更多静态视图和动态视图信息，以及怎样配置操作按钮的信息，请参阅[WatchKit 编程指南](#)。

模态界面（Modal Sheets）

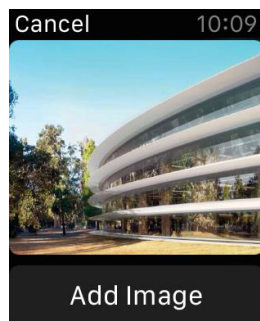
模态界面有优点也有缺点。它提供给用户一种暂时隔绝与其他app交互的情境，让用户去完成任务，在不受干扰的情况下获取信息，或是继续完成当前菜单下的操作选择。

当然，最理想的情况是你能够尽量减少模态视图的出现，建议只在以下情况下考虑创建模态情境：

- 当前获得用户注意力很重要

- 必须用户明确表示完成或明确表示放弃的任务，以便更好地判断是保存还是丢弃用户的数据

在分页模式中，模态界面可以只包含单屏界面，也可以包含多屏界面。单屏界面和多屏界面唯一的不同点只是有无配置界面下方的点状指示器。



Single



Paged

模态视图左上角的位置是预留给关闭按钮的。当用户点击关闭按钮，或横滑至最左边缘时，系统便会立即关闭模态视图。关闭按钮是必须呈现的，但是你可以改变按钮的文案以便更好地传达语义。你不需要在内容主体中放置关闭按钮。通常关闭按钮的文案为“关闭”和“取消”。且文案始终用白色呈现。

如果一项任务需要用户确认，那么请在模态视图主体中提供确认按钮。请使用标准的按钮去支持确认操作。确认按钮被点击后应关闭当前模态界面，并执行相应的确认后的后续操作。

保持模态任务的简单。避免从初始模态界面中呈现另一个模态界面。

布局

布局规范

控制在界面上并排呈现的操作数量。当并排呈现操作时，尽可能考虑使用图标而不是文字去表达操作的含义。不要并排放置3个以上的操作。太多的并排操作会让点触区域太小，以至于用户难以点击。

充分利用屏幕的边缘区域。Apple Watch 的表盘边缘提供了视觉上的边距留白，因此你不需要再在呈现内容的屏幕面积内再为内容周围留出边距。

使用相对位置布局app里的对象。因为App在不同屏幕尺寸Apple Watch中呈现相同的界面内容，相对位置可以使得被呈现的元素自适应填充不同屏幕尺寸的空间。

在图层显示上优先考虑左对齐方式。你的界面上的元素通常自上而下，自左而右进行布局，左对齐方式能够保证你有足够空间完整显示内容。

让有标签的按钮通栏显示。带标签的按钮通栏显示可以确保按钮标签文字能够完整显示。

使用情境菜单呈现次要操作。对于不常用的操作，通过呼出情境菜单呈现的方式要优于在界面上增加次要操作按钮的方式。

屏幕尺寸

在不同尺寸屏幕Apple Watch上，你的内容呈现都应该是一致的。当你设计内容布局时，让布局里的元素能够自适应填充不同尺寸空间。

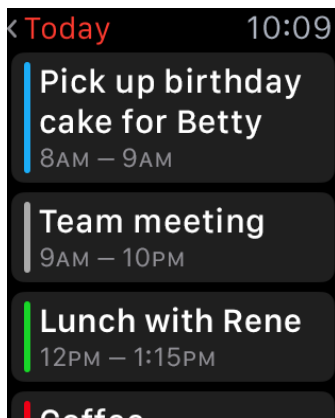


必要时，分别提供适合不同尺寸屏幕的图片。除非同一张图片资源在不同尺寸屏幕下均显示良好，否则，请分别提供适合各自屏幕尺寸的图片资源。

颜色和字体

颜色

颜色能够帮助你的app建立视觉和品牌的一致性。



使用黑色作为你app的背景颜色。黑色背景色可以柔和地与表盘边缘融合，使内容和表盘边缘间没有明显的边界。避免在你的app中使用明亮的背景颜色。

使用你的app的关键色建立品牌识别。每个app都有自己的关键色。系统将使用这个关键色呈现你的app通知的标题字段，高亮通知界面中你的app名称或关键内容。当然，你也应该同样使用关键颜色作为你的app品牌识别的一部分。

使用背景的高对比色呈现文字。背景的高对比色能让文字的可识别性更强。

避免使用颜色表达交互性。可以适当应用颜色建立你的品牌识别，但是不要使用颜色去暗示按钮或其他操作的交互性。

考虑有色彩识别困难的用户。大部分有色彩识别困难的用户很难区分红色和绿色。测试你的app，以确保你没有使用红色和绿色作为2种数值或状态的唯一区分方式（很多图片编辑类软件会提供帮助你验证颜色的工具）。

颜色能传递语义，但有时并不是按照你的初衷传递语义。每个人对颜色的语义理解都不同，很多文化中对于同种颜色的语义理解甚至是相悖的。因此，花点时间在调研你使用的颜色在不同国家和不同文化中是如何被看待的。尽可能地，确保你的app中使用的颜色传递了正确的语义信息。

字体

总而言之，文本必须具备易读性。如果用户不能有效地阅读你的app中的文字，那么使用再美丽的字体也无济于事。

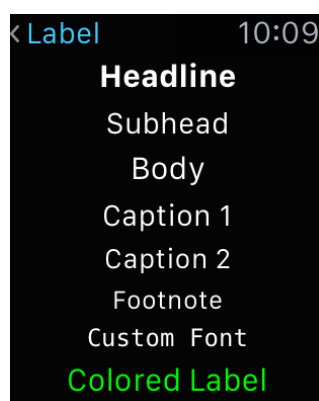
系统字体是专门为保证Apple Watch上文字易读性设计的。在全局字体大尺寸呈现下，系统会自动压缩字符间距以最大化利用横向空间。但是在全局字体小尺寸显示下，系统会自动松散化字符间距，并将“a”和“e”的孔洞空间增大，以便于用户阅读和快速浏览。当全局字体变小时，标点符号会相应变大。并且随着字体尺寸的变化，Apple Watch会灵活切换不同字体以确保清晰度和可读性。

app应该始终支持动态字体呈现，当你支持动态字体呈现时，你将得到：

- 在每种字体尺寸下均能自动调整字符间距和行高
- 根据不同语义的文本块，区别呈现文本样式，例如区别呈现主体文字，标注文字，标题文字
- 文本呈现灵活响应用户对于字体尺寸的设置

如果你使用内置的字体样式，你将免费得到动态自适应字体机制的支持。如果你使用自定义的字体集，你必须做一些工作以确保自定义的字体集也具备动态自适应字体机制的功能。想要学习如何使用文本风格以及如何确保你的app能够根据用户设置自适应呈现字体，请参阅[iOS文本编程指南](#)中的[文本风格](#)章节。

只要可能，尽量使用内置的字体风格。内置风格集能够自动支持动态自适应字体机制，并且确保字体在Apple Watch上完美呈现。



优先考虑在你的app中只使用1种字体。app会使用额外的字体来为品牌服务，但请有节制地使用额外的字体。混合使用不同的字体会让你的app看起来风格离散和品位不足。使用UIFont字体风格API来定义不同类型文本的显示区域，例如主体段落字段或者标题字段。

如果想要定制系统字体尺寸，请确保选择正确的尺寸。选择San Francisco Text字体呈现文本请选择19点或19点以下的。选择San Francisco Display字体呈现文本，请选择20点或20点以上的。

动效

Apple Watch充满了优雅微妙的动效，从而让用户有着愉悦的使用体验。优雅得体的动效能够：

- 传递各种状态和提供反馈
- 帮助用户视觉化感知他们的操作

使用一系列静帧图片来创建预渲染动效。在你的Watch app中事先封装好动效集可以让它们更快地呈现在用户面前。封装的动效使得你的app能够运行高帧速率和更平滑的动效。将你从WatchKit 中创建的动效导入到Apple Watch中时增加回放延迟效果。

回放控制只适用于图片和成组的对象。很多界面对象会循环播放动效化的图片系列。为了停止一段动效或者回放一段动效中的特定帧列，你必须使用图片或成组的对象。

品牌

成功的品牌塑造不仅仅能够为app自身带来无形资产，优秀的app通常能将独特的外观和感受整合到品牌塑造中，从而给予用户愉悦，难忘的体验。

有很多方式可以塑造你的app品牌，包括图标，颜色，自定义按钮等，当你设计你的app中的图形元素时，请确保每个自定义的元素均是美观且能够传达正确的功能语义，同时与app中其他界面中标准或自定义的元素风格保持一致性。

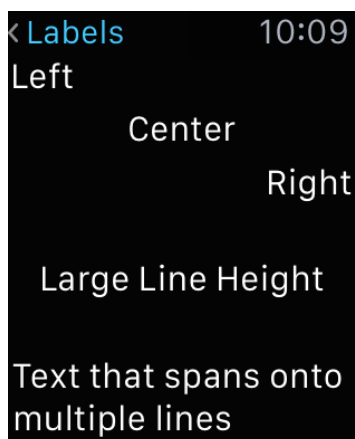
以优雅不突兀的方式引入品牌传达。用户使用你的app是为了更好地完成任务或得到有效的服务，用户不希望会有被迫看广告的感觉。为了达到最好的用户体验，你需要通过你对于字体，色系，图片样式的选择来潜移默化地向用户传递品牌价值。

尽量压制频繁在整个app界面或快速浏览界面显示你的Logo的想法。Apple Watch中的空间有限，且logo的每一次出现都会分散用户对于当前所专注内容的注意力。且在app界面中呈现logo并不能达到在网页中呈现logo一样的品牌传达效果，用户常常在不知道网站品牌所属的情况下打开网页，但用户往往在进入你的app之前就能看到你的app图标。

UI 元素

标签

标签用于显示静态文本。



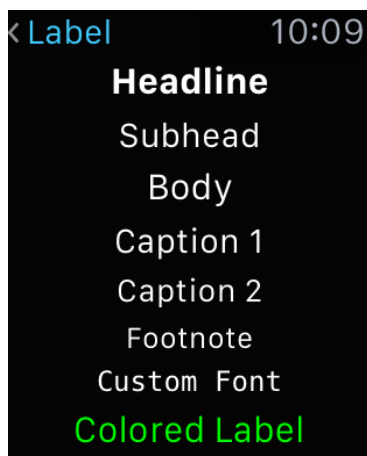
标签：

- 显示任意形式的静态文本
- 不支持直接的用户交互行为
- 可以通过编程方式进行更新
- 可以跨行显示

标签是 **app** 中最常见的元素之一。标签用于向用户显示短信息。标签非常适合显示相对较少的文本。

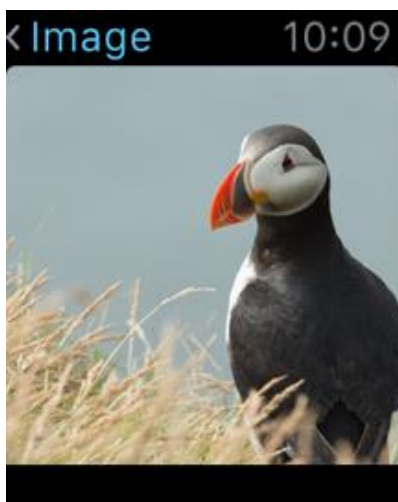
确保让你的标签清晰易读。 标签文本使用高对比度的颜色和动态字体，确保标签文本的尺寸适合于用户。内置系统字体（推荐使用）为 **Apple Watch** 提供了最好的易读性。如果你选择使用自定义字体，不要为了花哨的字体或亮丽的颜色而牺牲文本清晰度。（关于文本在 **app** 中的应用准则，其中包括有关动态字体信息，请参阅[颜色和版式](#)。）

尽可能地使用系统的内置样式。 内置的样式的设计通常具有高清晰度的特点，易于用户使用。



图像

图像可以展示一张图片或一系列动态图片。



图像

- 没有预先定义好的外观；只是用于显示相应的图片
- 不支持用户的交互行为
- 为开始播放和暂停动画提供控制方式

将 **Apple Watch** 展示的每一个内容，控制在一个适当的尺寸。不要因为屏幕尺寸不同，而尝试去拉伸或是缩放图片。根据设备的屏幕分辨率提供像素大小正确的图像资源。

将所有图像资源创建为@ 2x 图像。不需要建立非视网膜图像。

组

在界面中，组是对内容进行布局的一个非常重要的工具。组充当着其他对象的容器。组并没有预先设定好的外观，但是可以用自定义颜色或图片对其进行设置。组还有设定位置、尺寸、页边距和其他与布局相关的属性。

组：

- 项目的布局为水平或垂直
- 包含一个或一组交互元素
- 可自定义页边距和组内元素间距的属性
- 可以显示由一张图片或纯色构成的背景
- 可根据背景和内容配置圆角半径大小

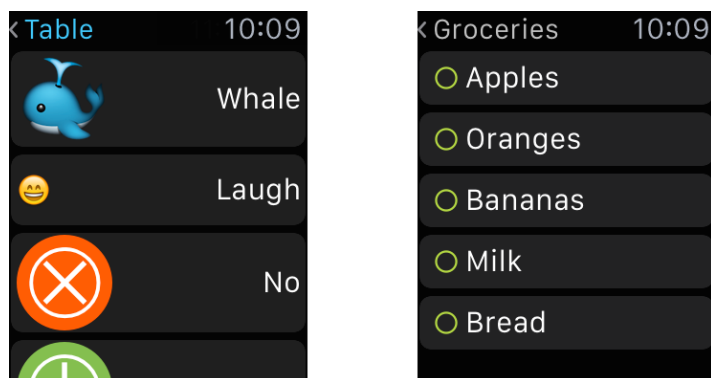
组是 **Xcode** 中进行设计的主要工具。组可以拥有自己的背景颜色或图像，这意味着可以使用组并将其作为视觉元素。

通过将组嵌套于其他的组中来创建复杂的布局。你可以通过这样做来同时显示水平项和垂直项。你也可以通过将组嵌套于其他组中，进而得到与被嵌套组相同的边距或是元素间距的属性。

为 Apple Watch 不同的显示器尺寸创建不同的背景图片。不要因为屏幕尺寸的不同，而尝试去拉伸或是压缩图片。根据设备的屏幕分辨率提供正确像素大小的图像。

表格

表格用于以单列，多行列表形式来呈现数据。使用表格可以动态地显示内容。



表格：

- 行的显示支持多种类型
- 可滚动
- 可以设置背景颜色或图片

在设计阶段确定表格中的行类型的布局。所有行必须设计为显示在最前端。你可以在运行时测试并选择你真正想要的行类型。

保持行类型的一致性。你可能会创建不同类型的行来分别显示内容、页眉和页脚。确保在使用这些行时具有一致性。

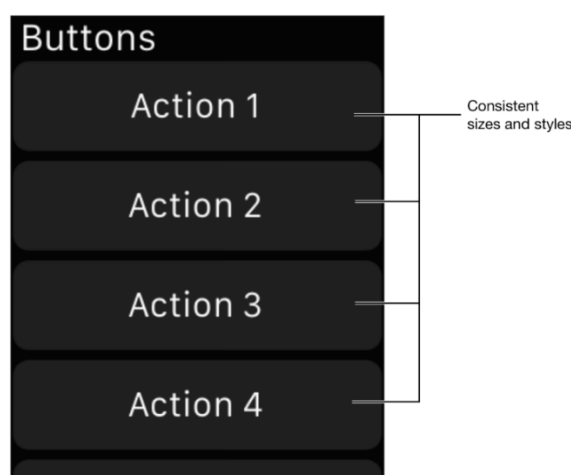
避免将类型明显不同的内容混行显示。当显示内容时，对应的行类型也要保持一致。只有在进行分离操作或是对行做调整操作时才可以使用其他的行类型。展示内容时使用相同的行类型不仅可以让行尺寸保持一致，而且容易进行导航。

在同时显示多个内容时限制表格中行的数量。表格中具有超过 20 行的内容时，滚动会变得很累。这时只显示一组相关的子集，同时为用户提供加载更多行的选项。

不要将表格嵌入到组中。表格的动态调整取决于其所包含的行的数量。不将表格嵌入组中可以避免由组为其带来的高度限制。

按钮

按钮用来执行 app 定义的操作。



按钮：

- 背景可自定义
- 有圆角
- 可以包含一个标签或一组对象

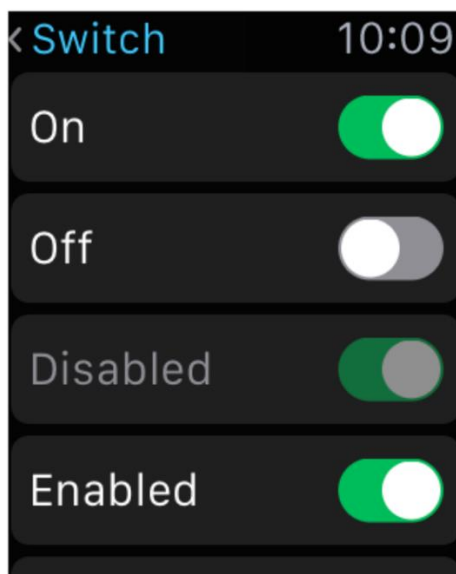
按钮为通栏按钮。可以在运行时更改通栏按钮背景的自定义颜色或图像。

尽可能的去匹配按钮的高度。如果多个按钮同时显示在同一个屏幕上，让每个按钮都具有相同的高度。

通过为按钮添加圆角，让其与其他元素区分开。标准的按钮圆角半径为 **6 个点**。

开关

开关表现了两种互斥的选项或状态。



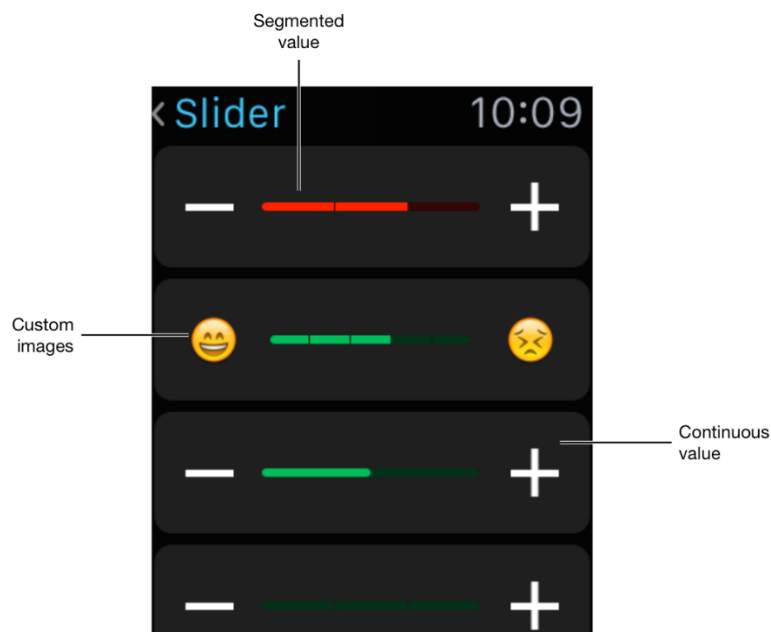
开关：

- 显示一个选项的二元状态
- 总是包含一个标签

使用开关来向用户提供一种在两个选项中指定一个的方式，例如是/否或开/关。

滑块

滑块控件允许用户在一个限定范围内调整数值。用户可以通过点击滑块两边的图像来调整数值。



滑块：

- 由一条水平的滑轨和两端用来调整数值的图像组成
- 可以使用分段或是连续的滑轨显示当前值
- 确保增加或减少的数值为预先设定的值。
- 不要把具体的数字展示给用户

可以放置自定义图像，用来传达左右两端数值的含义。如果不提供自定义图像，那么系统默认显示“+”和“-”的图像。

地图

地图可以将目的地信息呈现给用户。地图可以显示出目的地或指定的地点。地图属于静态快照，不能在你的 app 内进行交互。点击地图，用户将进入到地图 app 中。

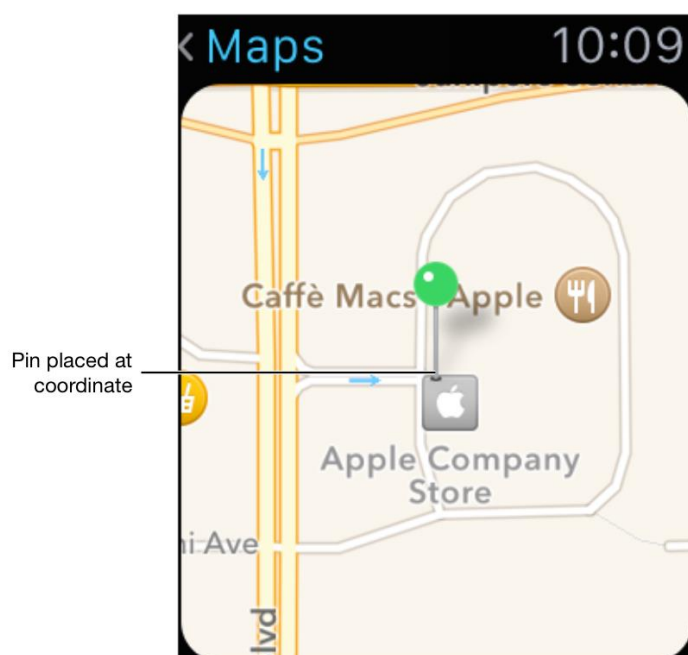
将地图区域设置成为包含相关位置信息的最小区域。在未设置 Watchkit extension 时，地图优先显示程序

设定好的区域。选择的区域既要包含所有指定地点，但是还要足够地小以确保用户可以方便使用。

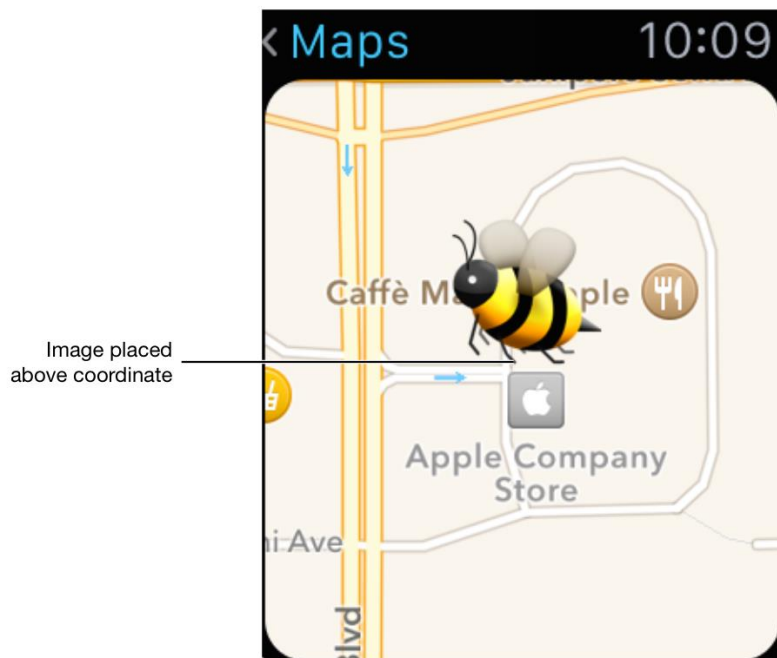
不要创建大小超过可用内容区域的地图。地图应该保证以适当的尺寸适配于当前 Apple Watch 屏幕，且用户不需要通过额外的滚动就可以看到完整的地图。

在地图中对需要突出的地点使用备注。备注是显示在地图上方用来标记地点或呼出信息的图像。在地图上显示的备注数量不要同时超过 5 个。

地图内置图钉提供绿色，红色，紫色 3 种颜色。绿色图钉用于标记起始点，红色图钉用于标记目的地，而紫色图钉用于标记指定的地点。



可以使用自定义图像创建备注。图像的底部边缘位于目标坐标之上。



日期和计时器

日期和计时器是 Apple Watch 里用于显示与时间相关的数值的一种特殊的标签。

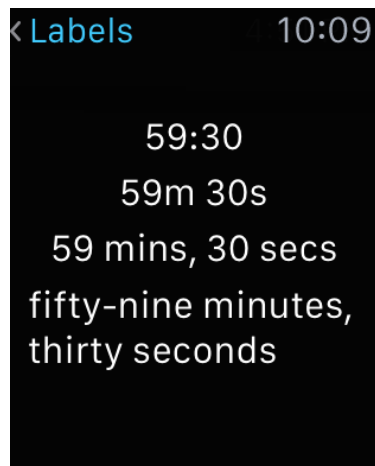
日期标签

日期标签：

- 显示日期、时间、或同时显示二者
- 可以通过设置，以可改变的形式、日历、时区来显示日期和时间
- 不需要去升级你的 WatchKit extension

无论什么时候你都可以使用日期标签显示想要看到的日期和时间。

计时器标签



计时器标签

- 倒计时到一个特定的时间或从一个特定的时间开始计时
- 可以通过设置，以不同形式来显示数值
- 不需要去升级你的 WatchKit extension

务必要通过使用计时器标签来实现一个精准的倒计时或计时器。

菜单

在 Retina 屏上，一个按压触摸手势可以触发与当前界面情境相关的菜单。使得用户不用离开当前界面就能在被呼起的菜单上执行相关操作。



菜单显示 1-4 种操作选项。操作选项按照被添加进菜单的顺序显示，并且通常为自左至右、自上而下排列。菜单不支持层级显示也不支持滚动。你可以在设计阶段自定义菜单上的操作选项，也可以选择让程序自动配置。

菜单操作适用于当前界面的情境。每个界面可以有特定的菜单，也可以没有菜单。对于当前界面呈现内容的操作或者是已选中对象的操作不适于放置在当前情境菜单中。

每个操作选项必须有一个图像和一个标签字段。菜单图像应采用系统线性风格，并采用标准的背景。标签字段必须在 1-2 行内显示。

菜单是可选择配置的。只在当前界面有相关情境操作时才需要显示菜单。系统能够通过使用动效暗示用户当前界面没有菜单存在。

有关创建菜单中的图标的信息，请参阅[菜单图像](#)

图标和图像设计

图标和图像尺寸

每一个 app 都需要一个漂亮难忘的主屏幕图标，以带给用户独特的识别感受。由于用户需要在主屏幕上通过图标识别不同的 app。因此你的 app 图标应与你的 iOS 上的图标一致并易于识别，同时达到传达应用目标的作用。



图标尺寸

主屏幕上的图标是圆形的，表 20-1 列出了相应的直径数值以及每个图标的用法。通过全出血的方形图像创建你的图标。系统将会自动附加圆形遮罩。

表格 20-1 图标尺寸

类型	Apple Watch (38mm)	Apple Watch (42mm)
通知中心图标	29px	36px
完整浏览通知视图 (Long Look Notification) 图标	80px	88px
主屏幕图标 缩略浏览视图 (Short Look) 图标	172px	196px

将所有图像资源创建为@2x 图像。不需要在你的 Watch app 程序包中包含非@2x 图像资源。

所有图像和图标，建议使用 PNG 格式，但要避免使用 interlaced PNGs。

图标和图像的标准色彩深度是 24 位，即红，绿和蓝分别为 8 位。你可以在图像中包含一个 8 位 Alpha 通道，但不是必须这样做。你还可以使用带有索引颜色的 PNG 图像，以便在你的图像文件中节省空间。

主屏幕的图标

Apple Watch 的主屏幕是独特而常见的。主屏幕图标与 iOS 的图标相仿，但是不伴有相应的文字说明。在如此小的界面中，这些图标必须清晰地传达 App 的语义。当 Apple Watch 的 app 与 iOS 的 app 为绑定关系时，图标可以保持视觉上的一致性。但当 Apple Watch 的 app 是作为 iOS 上 app 的补充应用时，图标可以相应地进行区别设计。

为了达到最佳效果，你可以向专业的平面设计师求助。一位有经验的平面设计师可以帮你的 app 制定一个整体的视觉风格，并将该样式应用到所有的图标和图像上。

使用易于人们识别的通用图像。在一般的情况下，要避免关注次要或模棱两可的语义元素。如邮箱的图标使用信封，而非邮箱、邮包或邮局的标志。

拥抱简洁。尤其要避免在你的图标中充斥太多图形。找到最能体现你的 app 本质的图形，以简单独特的形状表现它。考究地添加细节。如果图标的内容或形状过于复杂，细节就会变得更加复杂，可能在更小的尺寸中会显得模糊不清。

抽象传达你的 app 的含义。以艺术的方式演绎现实是更好的做法，因为这样做能强调出你希望用户记住的主体方面。

让你的图标与你 iOS app 图标类似。保持相似的外观帮助用户建立起你的 Watch app 和 iOS app 的联系。

为 Apple Watch 两个显示尺寸制作不同大小的主屏幕图标。你要确保设计的图标要适合所有型号的 Apple Watch 的尺寸。请参考表格 20-1 上的特定尺寸。

菜单图像

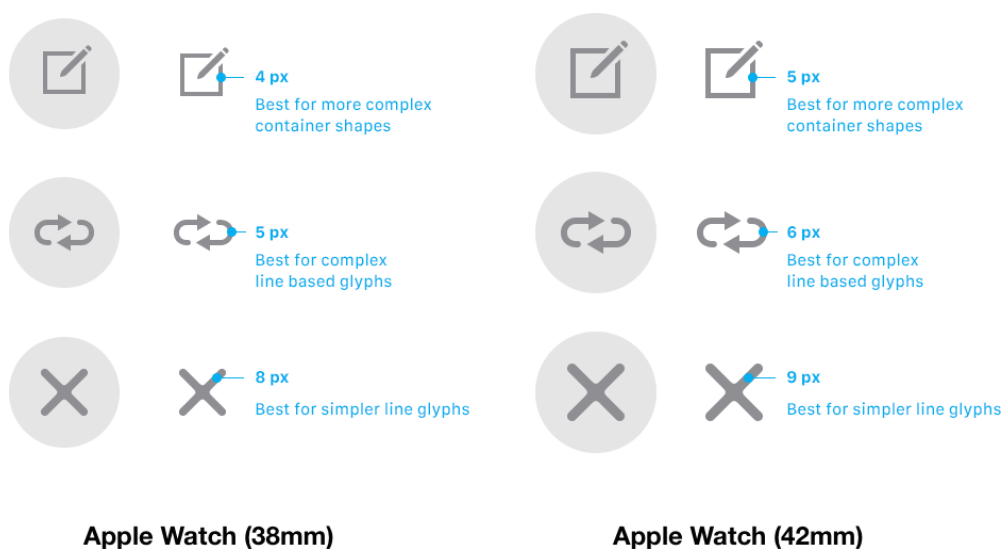
在按压触摸屏上的图标是系统模式化的图像，即通过 Alpha 通道定义最终的图像形状。忽视图像中的颜色信息。

对于菜单图像，画布尺寸通常大于实际内容尺寸。内容周围多余的空间确保菜单图标边缘和你的内容之间有足够的边界。

表格 21-1

设备	画布尺寸	内容尺寸
Apple watch (38mm)	70px	46px
Apple watch (42mm)	80px	54px

当设计菜单图像上显示的图形时，使用适合设备尺寸和图形复杂性的线形宽度。保持线形宽度至少 4 个像素，以防止它们难以辨认。



对于菜单图像，建议使用 PNG 格式，但要避免使用 interlaced PNGs。