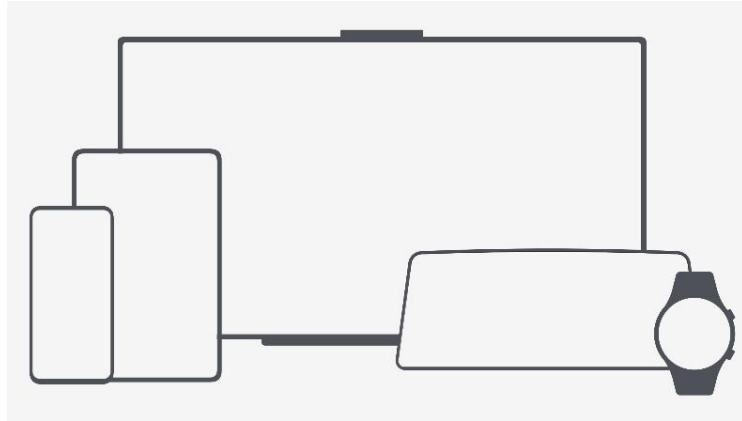


通用设计基础

概述



当为多款不同的设备开发应用时，除了要考虑每个设备的独特特性外，还需要考虑多个设备的共性，并使用通用性设计方法提供既符合设备独特特点，又具有跨设备一致性的设计，从而减少用户学习的难度，降低应用开发的成本。

在为 HarmonyOS 进行多设备开发时，可以从如下几个方面考虑设计和开发的拉通：

- 应用架构
- 人机交互
- 视觉风格
- 布局
- 界面用语

应用构架

应用中的导航结构

应用中的导航用于引导用户在应用的各个页面进行浏览。常用的应用导航有：

- 平级导航
- 上下级导航
- 混合导航

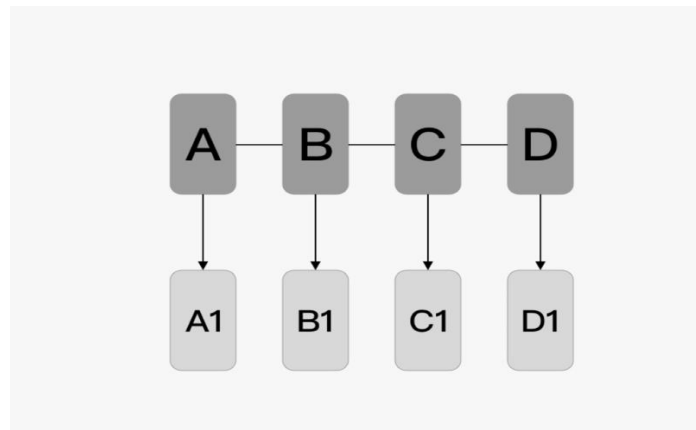
导航的原则：

- 一致。导航操作的结果应该与用户的期望保持一致。使用用户熟悉的界面布局 and 控件，让用户无论在什么页面，都知道如何导航。
- 清晰。导航应该提供清晰的路径。用户使用的时候，能够知道当前处在界面的什么位置，操作后将会跳转到什么位置。

平级导航

平级导航结构中，页面均处在同一层级。

使用场景：用于展示同等地位或同等层级的界面。例如以 TAB 方式组成的页面。

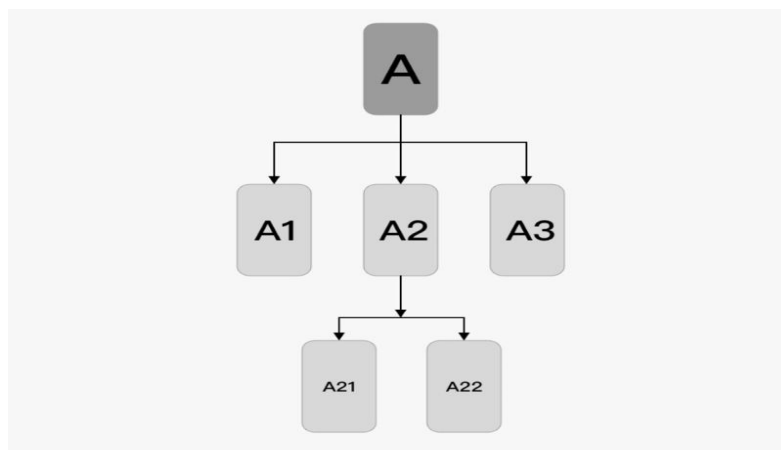


上下级导航

上下级导航结构由父页面和子页面组成。父页面可以有一个或多个子页面。每个子页面都有一个父页面。

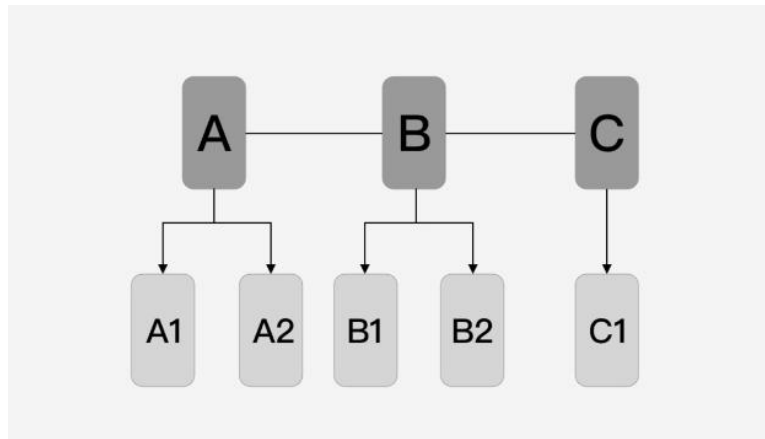
上下级结构导航适用于多层级的复杂结构。层级结构深的内容，用户访问的路径变长，效率降低，可以通过适当的层级穿透设计（如快捷方式）解决此问题。

使用场景：页面存在上下级关系的应用。



混合导航

在应用中，对同等地位或同等层级的页面使用平级导航结构，对具有复杂关系的页面使用上下级导航结构。



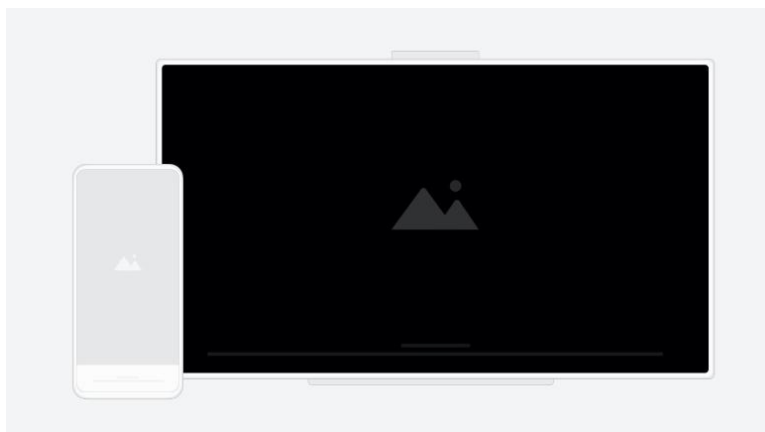
常用应用界面框架

常用应用界面框架有：

- 启动页
- 详情页
- 列表视图
- 宫格视图

启动页

针对内容型应用，应用的首页内容的获取需要花费一定的时间，此时可以使用启动页缓解页面加载内容的等待感。启动页可以展示应用的品牌形象或者广告。



- **仅针对内容型应用使用启动页。**内容型应用在启动时，需要花一定的时间获取内容，因此需要使用启动页，减少用户的等待感。没有网络加载内容的应用，不需要使用启动页。

- **避免让用户等待过长时间。**用户总是希望第一时间看到应用内容，因此在页面加载完成后，需要及时呈现内容。
- **从后台加载应用时，不应该显示启动页。**当应用被切换到后台后，再从后台加载回来时，不应该再次显示启动页。应用需要保留应用的状态，以便从后台恢复，方便用户继续浏览。

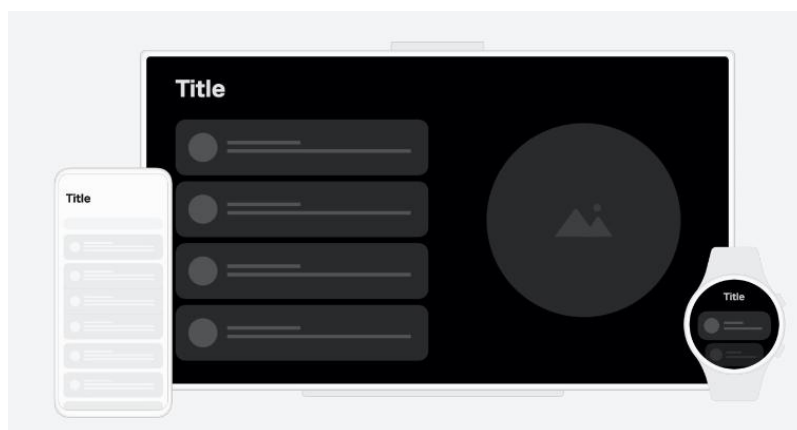
详情页

详情页用于展示应用的详细描述和操作。



列表视图

列表视图通常用于文字和数据内容的展示。



- **列表应该按照一定的逻辑排序，便于用户浏览和操作。**例如：按字母顺序排序、按时间排序。
- **列表应该是同类项的集合，应该对外呈现一致的布局样式。**常见的是单行列表、双行列表和三行列表。
- **列表显示的内容不宜过多。**用户应该一眼就能关注到重要信息和操作。

网格视图

网格视图通常用于图片和视频内容的展示。

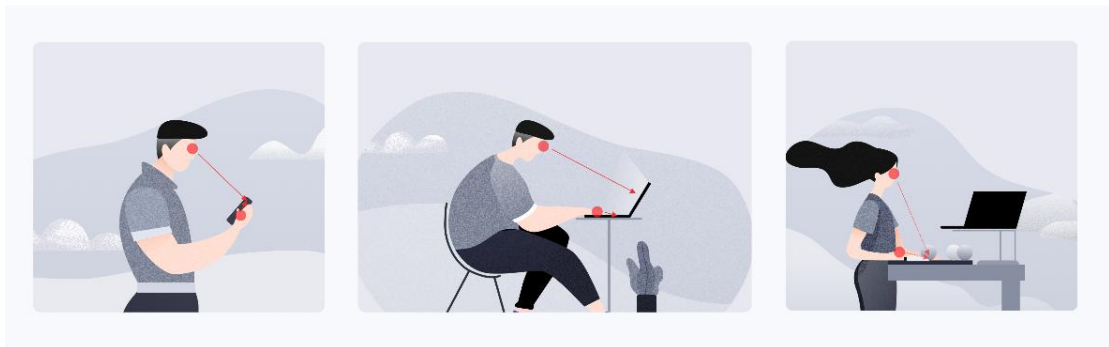


- 网格视图显示同等重要的项目，具有统一的布局。
- 网格视图以图像为主组织内容。例如：图库中用网格视图展示图片。
- 网格视图可以辅以文字和操作。例如：应用市场中使用网格展示应用程序图标，简单描述和下载按钮。
- 网格视图应该考虑响应式布局。在横竖屏切换时，网格视图应该能够按比例缩放内容以适配页面的宽度。

人机交互

概述

在全场景的数字体验中，越来越多类型的智能终端设备分布在用户的日常生活中，可交互的用户界面广泛存在于智能手机、平板、电脑（PC）、智能穿戴、电视、车机、虚拟现实（VR）和增强现实（AR）等设备上。应用可能在多种设备上运行或在单一设备上被用户通过多种输入方式操控，这需要其用户界面能够自动识别和支持不同的输入设备，以使用户以习惯的、舒适的方法与其进行交互。



在全场景的人机交互方面，HarmonyOS 的核心思想是“根据用户的状态，提供符合当前状态的交互方式，保证用户交互体验的一致性”。例如，当应用运行在触屏设备上时，用户可以通过手指长按打开上下文菜单；当应用运行在 PC 上时，用户则可以通过单击鼠标右键打开该菜单。

典型的输入方式（设备）包括但不限于触屏上手指/手写笔等直接交互、鼠标/触摸板/键盘/表冠/遥控器/车机摇杆/旋钮/手柄/隔空手势等间接交互、以及语音交互。

设计和开发应用时，设计师和开发者应考虑到应用具有使用多种输入方式（设备）的可能性，并实现相应的功能，保证在当前输入方式（设备）下应用能够以正确的、符合用户习惯的方式进行响应。

2 输入方式

2.1 基于触控的交互

很多设备都拥有支持多点触控的屏幕，允许用户使用手指和/或手写笔进行交互。它们与屏幕的接触状态、数量以及运动行为被识别成触控手势和操作，可以支持多种交互功能和体验（例如点击、滑动、缩放、旋转）。在多数情况下，应将触控交互作为用户首要的交互方式。以下内容描述了 HarmonyOS 所支持的核心手势。

滑动	<p>用户通过滑动来滚动列表或平移界面内容。</p> <p>场景：</p> <ul style="list-style-type: none"> 通过滑动滚动列表。 在内容区横向滑动切换页签。 通过滑动平移地图。 	
拖动	<p>用户将元素从一个位置移动到另外一个位置。</p>	
双击	<p>用户快速点击两下以放大/缩小内容、选择文字或触发特定的功能。</p>	
捏合	<p>用户使用两个手指按住屏幕向外展开以放大内容。</p> <p>使用两个手指按住屏幕向内收拢以缩小内容。</p> <p>场景：</p> <ul style="list-style-type: none"> 放大/缩小图片。例如，在查看图片界面。 放大/缩小内容。例如，浏览页面或者相机取景界面。 	

手势操作	功能描述	图示
点击	<p>用户通过点击某个元素触发功能或访问界面。</p>	
长按	<p>用户通过长按某个元素触发菜单或特定模式或进入界面。长按手势发现性差，常用功能不要使用长按来触发。</p> <p>场景：</p> <ul style="list-style-type: none"> 长按操作显示弹出菜单。例如，在联系人列表中长按某个联系人。 长按操作显示快捷菜单。例如，长按某个桌面图标。 长按操作进入多选。一般用于列表或表格界面。例如，长按信息的列表界面，长按图库的表格界面。 长按操作进入编辑模式或可排序模式，并伴随振动。例如，长按闹钟的列表界面，长按桌面的空白处。 长按操作选择文本。例如，在文本框中长按已经输入的文字。 长按图标进入该图标对应的功能详情。例如，在通知面板上长按功能开关图标。 	

敲击

适用于支持指关节手势的机型。



基于光标的交互

当用户使用指向设备（鼠标、触摸板、AR/VR 手柄、隔空手势等）与应用程序进行间接交互时，光标指向的对象和光标本身应提供适当的视觉反馈以表达对象的可交互性和到达的准确性。同时，应考虑利用光标支持精细化操作和悬浮状态的特性（相比手指触摸），以提升应用生产力、简化交互任务和增强信息展示。

本节规范主要适用于使用鼠标和触摸板来操作平面光标的场景，但基本的设计原则可推广至 AR/VR 手柄和隔空手势等空间交互场景。

光标形态

光标形态的设计遵循以下三个原则：

功能表达

当光标悬浮在特定界面对象或区域上时，可使用系统定义的标准的光标样式集合来表达其交互状态。

应用也可根据其自身具体场景自定义光标的形态。

简单性

为保证全局交互体验的一致性，应优先使用系统已定义的光标样式集合。如无必要，请勿增加新的光标样式。

对于自定义的光标样式，应尽可能简洁直观地表达在当前界面状态下用户可进行的操作和所需精度，不应在视觉上过分吸引用户的注意力。

自动隐藏和显现

对于以触摸交互为主的终端设备，如果光标在一段时间内没有移动或用户使用了触摸交互，则自动隐藏光标保证界面的干净整洁。

悬浮对象的形态

当光标进入并悬浮在界面元素上时，界面元素可通过适当的视觉和行为反馈来使其对用户聚焦和提高指向的准确率。

界面元素需要响应光标悬停的动作，以清晰表达该元素的可交互性

在光标范式下，用户是通过将光标移动到界面元素上，然后执行对应的操作。这种情况下，需要界面元素在光标悬停的时候就反馈出其是否可交互，避免用户点击后才发现该元素不可交互而产生挫败感。

在表达界面元素的可交互性时，应考虑以下设计原则：

显著性：在悬浮态下，与普通状态和其周围的界面元素相比，拥有清晰的功能可见性。

微妙性：视觉上不应过于强调或与普通状态具有较大差别，避免干扰用户注意力。

一致性：同类型控件的可交互性表达（视觉和动效上）应具有一致性。从平板/PC向电视、AR/VR 使用场景扩展时，不同设备间控件的悬浮态也应具备一定程度的表达一致性。

在悬浮态下，界面元素要表达其可交互的热区

图形界面中可能存在很多界面元素其视觉可见部分和实际可（触控）交互的热区相差很大。在此类场景下，光标悬停时准确表达可交互的热区有利于用户感知元素有效的触发区域，并进而提高交互的效率。

界面元素的悬浮态效果类型

HarmonyOS 提供 3 种界面元素的悬浮态效果：分别是浮起、叠加背板和高亮。



浮起：当光标进入控件时，光标消失，同时对象放大。对象可在一定范围内移动，同时产生 3D 倾斜效果响应操作位置；移动时对象上层有光泽变化，高光跟随位移动表达实际光标的位置。浮起的效果一般适用于可见部分和实际热区大小接近、有背景填充、且在布局上与其它元素不相邻的控件，典型的控件类型为强调按钮、导航点、勾选等。

叠加背板：当光标进入控件时，其转化为浅色的圆角矩形叠加于内容底层，且背板、内容、光泽能够在热区范围内移动以表达光标的运动趋势。叠加背板的悬浮效果适用于可见部分和实际可交互热区相差很大的控件，此类控件通常无背景填充，包括文本按钮、工具栏、Tab 栏等。

高亮：当光标进入控件时，在控件底部叠加颜色色块，但不改变当前控件的形态且光标不消失。适用于容器类控件，比如列表、卡片、菜单等。

展示附加信息

当光标移动到界面元素上时，在以下场景中，可考虑通过显示附加信息来提升光标体验。

1. **受限内容预览：**例如在邮件、日历、备忘录等应用中，当光标悬停在内容列表/网格上时，可以通过悬浮窗显示更多的内容详情，方便用户进行快速预览和选择。
2. **精确位置显示：**例如在截图、设计、办公应用中，当光标悬停在某些对象上时展示位置、尺寸等数值，便于用户进行精细控制。
3. **控件功能提示：**如果一个界面元素其功能的自我解释性较差，则可通过文本提示进行告知。

应谨慎使用悬停展示附加信息的功能。在使用时，展示的附加信息必须是必要的或能明显提升用户体验的。在使用展示附加信息的方案之前，应优先考虑界面的

清晰性、简洁性和表达性，以使用户无论是使用光标输入设备还是在直接用手指触摸交互，都可以很容易且舒适地使用应用。

精细化操作

光标交互区别于手指触摸交互的一个重要方面是光标支持精细化操作，设计师可考虑如何利用这一特性配合特定的光标形态来简化交互任务和提升生产力。

光标移动

应优化光标移动的显控比以提升其指向的速度和准确性。用户在使用光标指向目标时，可分为弹道加速和减速修正两个阶段。以鼠标为例，在弹道加速阶段，当焦点离目标较远时，用户快速移动鼠标以缩短光标与目标之间的距离，此时速度的重要性优于精度控制；在减速修正阶段，当光标离目标较近时，用户降低鼠标的移动速度，更为仔细的瞄准目标，此时精度控制重要性优于速度。

基于焦点的交互

当用户使用键盘、电视遥控器、车机摇杆/旋钮等非指向性输入设备与应用程序进行间接交互时，基于焦点的导航和交互是重要的输入手段。

本节描述了基于焦点交互的通用设计规范，针对各输入方式的细化的焦点交互方法请参考各输入方式的具体章节。

功能	设备类型	平板/PC	电视	车机		
	操作描述及按键设备	标准键盘	遥控器	旋钮	触摸板/方向键	摇杆
焦点导航	移至下一个焦点	Tab	N/A	顺时针旋转	N/A	N/A
	移至上一个焦点	Shift+Tab	N/A	逆时针旋转	N/A	N/A
	按方向移动焦点一个位置	左/右/上/下	方向键/触摸板滑动	N/A	方向键/触摸板	方向移动
	移至区域内第一个焦点	Home	N/A	N/A	N/A	N/A
	移至区域内最后一个焦点	End	N/A	N/A	N/A	N/A
	上一页/下一页	PageUp/PageDown	N/A	N/A	N/A	N/A

焦点交互	激活当前焦点操作	Space	确认键	确认键	确认键	确认键
	进入当前焦点内部	Enter	确认键	确认键	确认键	确认键
	取消当前操作/返回上一级	Esc	返回键	返回键	返回键	返回键
	打开当前项的上下文菜单	Menu键/快捷键组合	菜单键	N/A	N/A	N/A

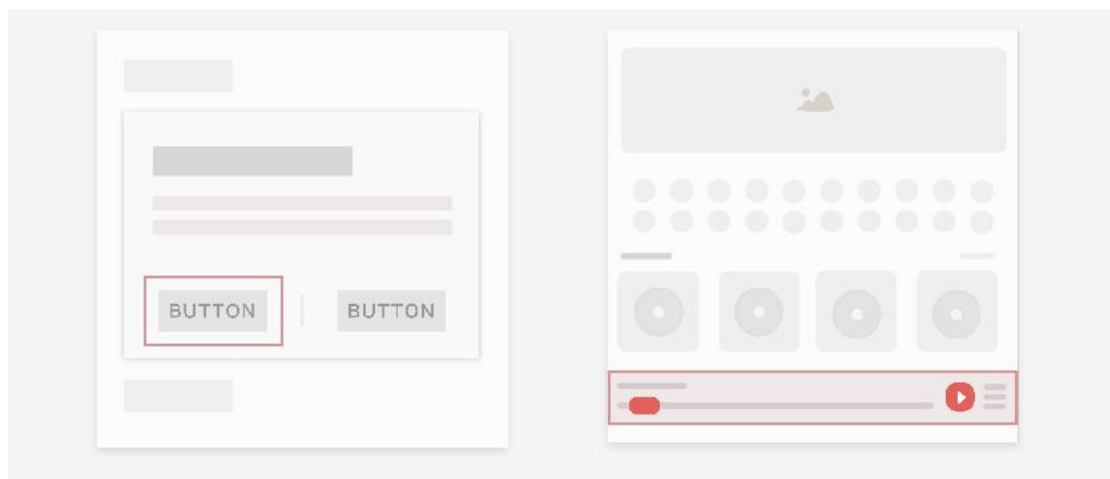
基本原则

内置支持

控件原生支持走焦操作，设计师和开发者可以针对具体的应用场景对是否获焦、焦点顺序进行调整。

提供初始

初始焦点的位置需明确、突出，让用户有效地识别该位置，以便顺利开展走焦操作。



焦点的默认位置与界面层级与内容相关，确认默认焦点的规则如下：

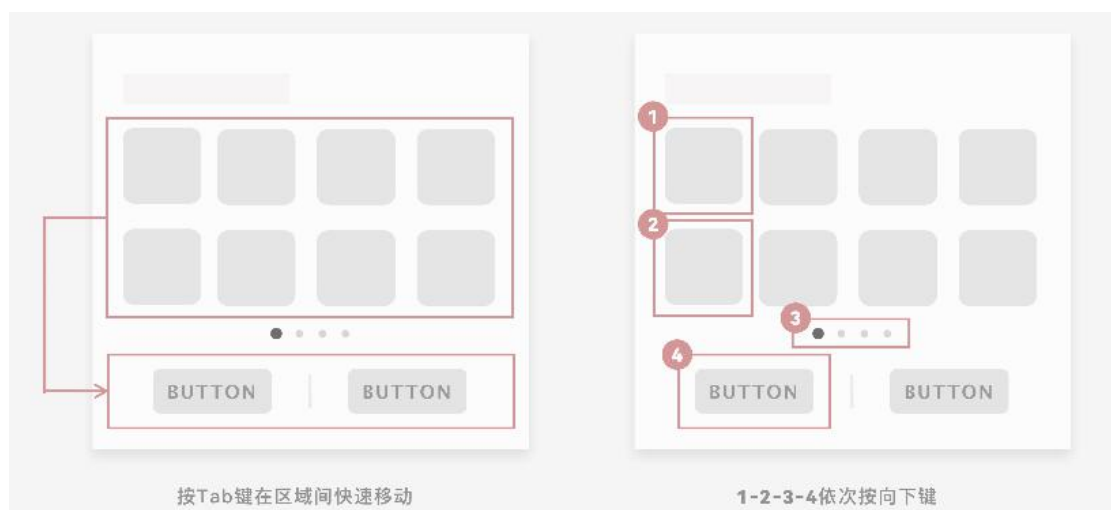
1. 层级优先，最顶层的界面优先。
2. 核心操作优先，根据应用提供的服务决定默认操作。
3. 内容优先，遵循从上至下从左至右的方向规则。
4. 不要在未加载完毕的区域显示默认焦点。

可遍历

为完成所有交互任务，焦点需要能够遍历所有可获得焦点的界面元素，以保证功能的完整性。

按区域获得焦点提高走焦效率

在由多种控件组成的可划分明显区域的界面中，可精简 Tab stop 的数量以提升走焦效率。



Tab stop 与普通焦点的异同（以键盘操作为例）：

Tab stop: 可以用 Tab 键获得焦点的控制元素，一般情况下，一个区域的首项为该区域的 Tab stop。

普通焦点: 不在 Tab 走焦序列之中，但在区域内可以通过方向键移动获得焦点。

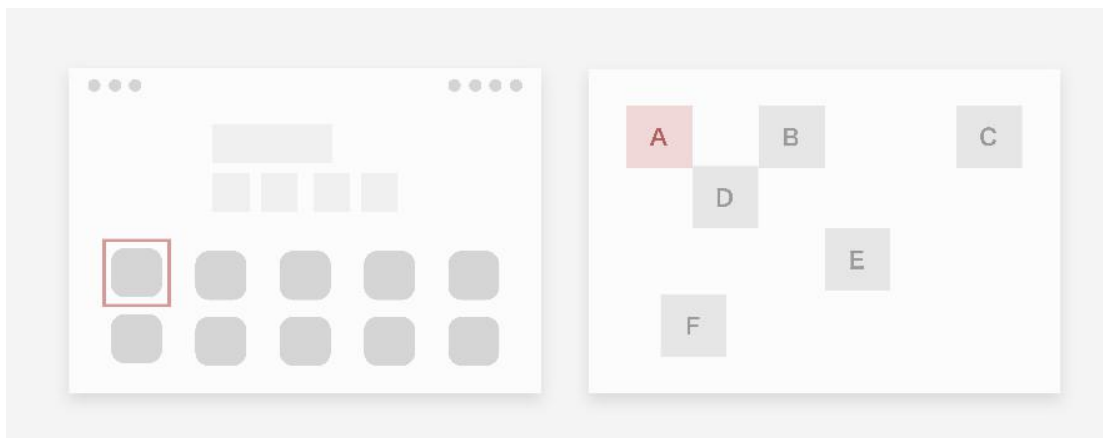
另一个典型场景是车机上通过旋钮和摇杆/方向键的配合使用来快速移动焦点。例如使用摇杆/方向键来切换页面、层级和区块，使用旋钮切换每个可点击的元素。

应根据输入设备支持情况决定是否开启按区域走焦，如无 Tab 键或其等价键支持，则不能开启区域间快速移动操作，此时，只能在单个焦点间遍历（Tab stop 数量与可获得焦点元素数量相同）。

走焦顺序

依据用户界面的形态和输入设备的不同，走焦场景可基本分为三类：绝对顺序走焦、相对方向走焦、和插入符走焦。

绝对顺序走焦



存在一个确定的走焦顺序，所有可获焦元素都会按照在屏幕上显示的顺序或指定的顺序进入走焦顺序中。例如当使用键盘时，Tab 键选择下一个焦点，Shift+Tab 键选择上一个焦点。

在设计应用时，可按照功能分区、视觉呈现等规则重新组织焦点顺序。

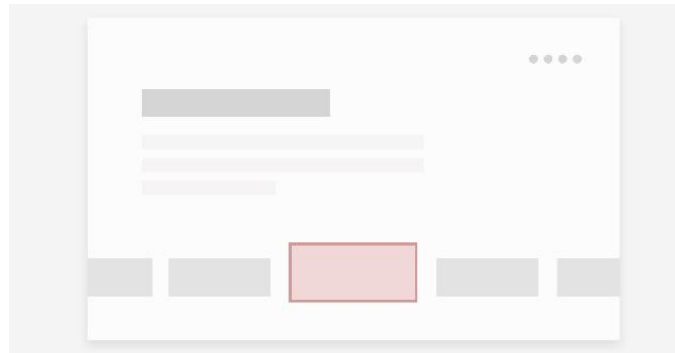
基本规则：

1. 用户只能按照给定的焦点顺序移动焦点。
2. 提供两种操作次序：正序和反序（例如键盘上的 Tab 和 Shift+Tab，车机旋钮的右旋和左旋）。
3. 一般来说，焦点列表首尾相连。

相对方向走焦

按照可获焦元素的分布方向关系进行走焦，存在多种不同路径的走焦顺序。

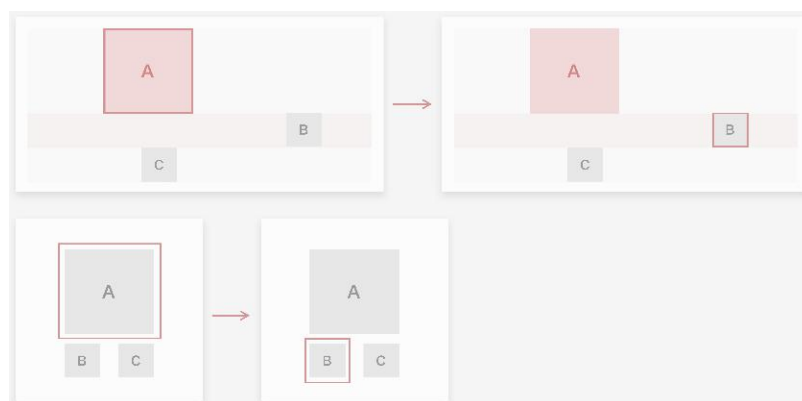
用户输入方向信息，系统通过某种算法计算出在相应的视觉方向上应获得焦点的元素，并使其获焦。典型的场景包括使用遥控器上的方向键与电视交互。



共用底层导航机制：

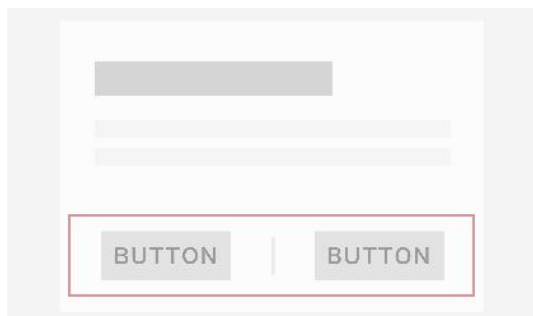
1. 无论输入设备是键盘、遥控器，或触摸板、摇杆等其他方向操作模拟设备，基本操作逻辑都保持一致。
2. 不同的导航策略将影响用户在按下方向键时下一个焦点的选择。

基于方向和距离优先的判断方法：



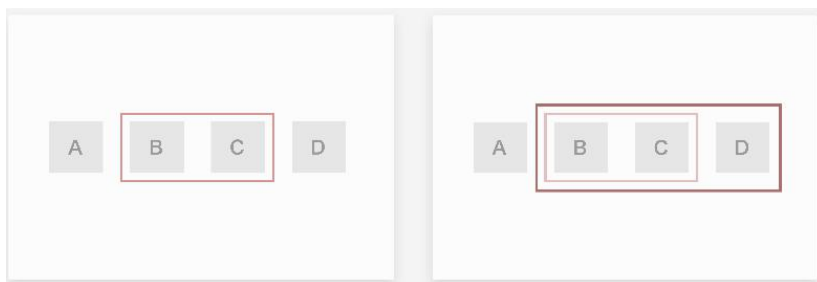
1. 收到用户按键输入
2. 将当前焦点边缘向该方向扩展
3. 第一个可投射至该扩展区域的即是目标焦点位置
4. 此时如果有多个候选焦点，选投影距离最短的那个
5. 如果仍有多个候选焦点，按照从上到下/从左到右的规则进行筛选

在绝对顺序走焦中嵌入相对方向走焦区域



此类导航优化主要有两个目的：

1. 提升操作效率。绝对顺序走焦可遍历所有可交互控件，在元素较多的情况下效率不高。
2. 使走焦顺序更具有逻辑性。按照功能组织分区，将逻辑顺序与视觉分区进行匹配，用户更容易学习。



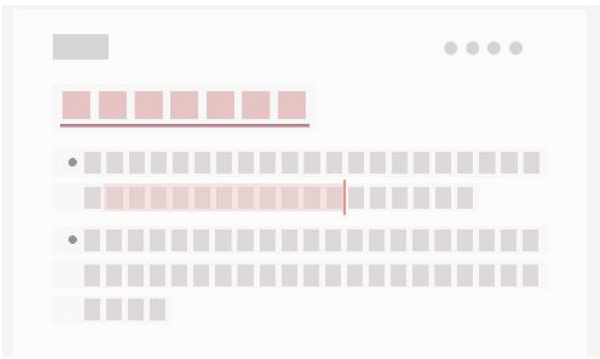
优化策略 1：

在相对方向区域内支持方向键导航，如上图所示，使用右键将焦点从 B 移至 C，使用左键将 C 移至 B；方向键只能在当前层级内使用，不能用于跳出区域，如：焦点位于 C 时，按右键无响应。

优化策略 2：

为每个区域指定一个默认 Tab 次序，使用 Tab 键能快速在区域间跳转，如：将 A 的 TabIndex 设置为 1, B 的 TabIndex 设置为 2, D 的 TabIndex 设置为 3，则按 Tab 键的走焦顺序为 A->B->D，可以不再遍历所有焦点，而在每个区域内可以使用方向键进行遍历，提供了一个效率更高的融合走焦方式。

插入符走焦



针对文本处理的操作，以插入符为操作指引，通过方向键移动、选择，常见于文本处理类、表格应用或控件之中，通常用来配合支持一些常见的文本操作类快捷键。

其它注意事项

对于支持触控交互或光标交互的系统，焦点仅在系统检测到用户正在使用特定的输入设备爱进行走焦操作时显示，避免视觉上的干扰。

在走焦操作状态下，如果检测到用户将输入方式切换为触控交互或光标交互，界面上的焦点视觉元素应自动隐藏。

鼠标

鼠标是一种典型的基于光标的、具备像素级精度的指向型输入设备，最为适用于对用户交互具有较高精度要求的生产力应用和高密度 UI 的场景。

一般地，鼠标由左键、右键和滚轮键组成，这些按键的交互应遵循业界标准的规范功能和用户的既有使用习惯。鼠标也可以通过和不同的键盘按键进行结合，提供额外的快捷操作体验。系统级一致的鼠标交互包括：

操作	预期行为
移动	移动光标，光标在不同元素界面上呈现和表达悬浮态样式，参考 基于光标的交互
左键点击	选择或启动一个项目的主功能（例如点击按钮启动应用或执行命令）
右键点击	展示上下文菜单（对应于触屏上长按）
滑动滚轮	沿上下方向或左右方向滑动或移动页面内容
左键按下并移动	拖拽对象或在一定区域内选择对象
按住Ctrl键同时滚动鼠标滚轮	缩放对象

鼠标与触摸交互的关系

在很多场景下，应用程序需要同时支持触屏交互和鼠标交互。

手指触摸交互通过对界面对象执行的物理世界真实的手势（例如滑动、旋转等）来模拟直接在屏幕上操作这些 UI 对象的能力。而鼠标受限于其间接交互的性质和仅有一个指针响应位置的特点，在一些交互任务上并不适合将手指触摸和鼠标按下操作完全对等。

1. 在一般的选择或启动对象上，鼠标左键点击可以直接等同于触屏上手指点击；
2. 在其它场景下，应优化应用以适配鼠标的交互习惯以提高生产力和用户体验的一致性。

例如，上下文菜单应通过右键单击来快速触发，而非模拟触屏上手指长按的左键长按；

拖拽对象时，允许鼠标左键按下即可拖拽，而无需长按才能触发拖拽状态；

尽量通过滚轮来控制页面的滑动和切换，而非左键按下后沿一定方向移动鼠标（这对用户来说是一个效率低下且难以控制的行为）。

关于更加详细的具体场景下触摸交互和鼠标交互的对应和转换关系，请参考[交互事件归一](#)。

关于光标和界面对象的悬浮态表现，请参考[基于光标的交互](#)。

触控板

触控板同时具备多指触控手势输入（触屏）和精细化指向型输入（鼠标）的特性，使得触控板既适合用于基于触摸交互优化的用户界面，也适合用于对指点精度有较高要求的生产力应用。

在为你的应用设计或适配触控板交互时，触控板交互应满足用户手眼分离状态下（眼睛看着屏幕，手在触控板上盲操作）的使用习惯，应遵循以下原则：

1. 触控板应该能起到取代鼠标的作用。
2. 应用在触屏上的手势操作功能可通过在触摸板上相应的手势来实现。

操作	预期行为
单指轻触后移动	移动光标，光标在不同元素界面上呈现和表达悬浮态样式，参考 基于光标的交互
单指单击	选择或启动一个项目的主功能（例如点击按钮启动应用或执行命令）
双指单击	展示上下文菜单（对应于触屏上长按和鼠标右键点击）
双指轻触后移动	沿某一方向滑动或移动页面内容（对应触屏上单指滑动操作）
单指按下并移动	拖拽对象或在一定区域内选择对象
双指远离和捏合	缩放对象
双指旋转	旋转对象
多指手势	系统导航

关于更加详细的具体场景下触摸交互和触控板交互的对应和转换关系，请参考[交互事件归一](#)。

关于光标和界面对象的悬浮态表现，请参考[基于光标的交互](#)。

键盘

键盘是一种重要的生产力输入设备，一个优秀的键盘使用体验应允许用户快速准确地进行文本输入、双手无需离开键盘即可在系统和应用内进行导航、访问所有的功能、以及支持无障碍体验。

走焦导航

焦点导航和交互方式、焦点划分和走焦顺序等设计原则请参考[基于焦点的交互](#)。

基础和标准快捷键

应结合应用场景提供必要的快捷键支持（例如：按键导航相关、剪切/复制/粘贴等），基础设计原则为：

1. 以传统 PC 为兼容对象，根据设备情况和应用场景进行裁剪。传统 PC 有最庞大的用户基础，用户习惯难以改变。但同时应移除过于复杂、频率较低、或缺乏设备使用场景的快捷键组合。
2. 新增快捷键支持。针对华为特有应用或功能（例如：分屏、多终端等）增加直接入口。

3. 以融合 PC 为目标为触屏手势操作添加快捷键支持，提高生产力（例如：打开通知栏、控制中心、显示/隐藏 Dock 等）。

类型		操作描述	快捷键
	系统功能	Home/显示桌面	LOGO+D / LOGO+H
		Back/上一级	ESC / LOGO+Backspace
		Recent/切换任务窗口	LOGO+Tab / Alt+Tab
		通知中心	LOGO+N
		锁屏	LOGO+L
		搜索	LOGO+S / Ctrl+F
		切换语言/输入法	LOGO+Space
		截屏	LOGO+Shift+S
		智慧分屏	LOGO+X
		全屏/多窗口切换	LOGO+W

全局快捷键	键盘走焦	将焦点移至下一个/上一个可获焦控件	Tab / Shift+Tab
		按照指定方向移动焦点/光标	上/下/左/右
		切换当前选择的复选框、单选按钮、开关控件的状态	Space
		激活获焦的按钮/菜单项等	Enter
	常见操作	剪切选中内容至剪贴板	Ctrl+X
		复制选中内容至剪贴板	Ctrl+C
		粘贴剪贴板内容至当前位置	Ctrl+V
		撤销最后一次操作	Ctrl+Z
		在当前焦点控件/窗口内选择所有项目	Ctrl+A
		选中项的上下文菜单/激活多选	LOGO+Shift+C

进阶操作	格式相关	加粗	Ctrl+B
		下划线	Ctrl+U
		斜体	Ctrl+I
	系统功能	关闭当前焦点窗口	Alt+F4 / Ctrl+W
		搜索	Ctrl+F
		弹出当前选中对象的上下文菜单	Shift+F10 / Menu
		打印	Ctrl+P
	Web浏览器	选择并聚焦地址栏	Ctrl+L / Alt+D
		刷新	Fn+F5 / Ctrl+R
	Tab类操作	新建Tab	Ctrl+T
		关闭当前Tab	Ctrl+W

	常用操作	重做最后一次操作	Ctrl+Y
		特殊粘贴	Ctrl+Shift+V
	系统功能	新建文件夹	Ctrl+Shift+N
		应用菜单	LOGO
		登出当前用户	Ctrl+Shift+Q
		任务管理器	LOGO+Esc
		重命名选中对象	F2 / Enter
		删除光标左侧的词组	Ctrl+Backspace
		去往行首	Home / Fn+←
		去往行末	End / Fn+→
		去往文档之首	Ctrl+Home / Ctrl+↑

	文本编辑	去往文档之末	Ctrl+End / Ctrl+↓
		前一词	Ctrl+←
		后一词	Ctrl+→
		前一行	↑
		后一行	↓
		去往下一个结果	F3 / Ctrl+G
		去往上一个结果	Shift+F3 / Shift+Ctrl+G
	资源管理器	去往地址栏	Ctrl+L / Alt+D
		去往历史中前一个位置	Alt+←
		去往历史中后一个位置	Alt+→

高阶操作		去往起始页（由用户或应用定义）	Alt+Home
	Web浏览器	给Url添加www.+ .com	Ctrl+Enter
		将当前页加入书签	Ctrl+D
		选择并聚焦搜索栏	Ctrl+E / Ctrl+K
		刷新（忽略缓存）	Ctrl+F5 / Ctrl+Shift+R
	Tab类操作	去往下一个Tab	Ctrl+Tab
		去往前一个Tab	Ctrl+Shift+Tab
		去往第n个Tab	Ctrl+n (n为Tab位置，最大为9)
		重新打开之前关闭的Tab	Ctrl+Shift+T

	窗口类操作	关闭当前应用的所有窗口	Alt+Shift+F4 或 Ctrl+Shift+W
		退出当前窗口的应用	Ctrl+Q
		打开Dock中的应用	LOGO+# (#是程序在Dock栏中的位置, 最大为9)
	控件与Widget	弹出当前控件的工具提示	Shift+F1
		显示与当前控件/窗口相关的帮助	Shift+F1
		将焦点移至下一个/上一个窗格	Ctrl+F6 / Alt+F6
		将焦点移至窗口的菜单栏	F10 / Alt
		在选中的Widget内选择/移动至第一个/最后一个项目	Home/End
	可访问性	允许用户使用快捷键时一次输入一个按键	连接Shift键5次
		停止/减慢当按键被按下时重复字符的速度	按住右侧Shift键8秒

自定义快捷键

在一些场景下, 用户可能更习惯于通过键盘来访问系统和应用功能, 例如在生产力软件中通过键盘快捷键来提高使用效率, 部分残障人士也更偏好使用键盘来进行交互。因此, 请确保应用中核心功能支持仅通过键盘操作即可访问。

- 对菜单中的功能提供键盘快捷键访问支持:** 键盘快捷键经常和应用内菜单中的功能关联出现, 应允许用户通过键盘快捷键访问这些功能。例如图库应用中照片列表支持通过快捷键进行全选、删除、复制、分享等操作。
- 对应用中其它核心功能提供键盘快捷键访问支持:** 应用中某些常用的功能没有对应的菜单项, 也需要相应的快捷键支持, 这类功能往往和控件上的直接操作对应。例如在聊天页面中, 应将 **Enter** 键与发送按钮的单击操作相关联, 允许用户在文本框输入消息内容后可以通过按 **Enter** 键来直接发送内容。
- 与系统标准快捷键相兼容:** 在所有应用中, 对于标准功能, 快捷键的设计应与系统标准快捷键的定义一致。另一方面, 在应用内尽量避免对标准快捷键的功能进行重新定义, 否则将对用户的使用造成混淆。

手写笔

在触屏上，手写笔是手指精细化操作的延伸，是一种像素级精度的指点设备。手写笔提供了一种直接的、自然的方式来进行数字内容书写、绘图和标注。目前手写笔分有无物理按键两种笔型，支持系统级一致的交互：

有物理按键

操作	预期行为
按住功能键画S型	截长屏
按住功能键圈画封闭性图形	局部截图
按住功能键单击屏幕	在熄屏状态下用笔点击屏幕，可快速打开备忘录书写
按住功能键长按屏幕	智慧识屏，长按可识别屏幕内容
按住功能键画直线	按住手写笔的功能键，屏幕中间画横线可直接进入分屏功能

无物理按键

操作	预期行为
用笔点击屏幕	在熄屏状态下用笔点击屏幕，可快速打开备忘录书写（需系统默认开启双击亮屏功能，后方可执行该功能。）

手绘套件开放

Pencil Engine：将手写效果能力以插件化方式集成，为应用开发者创造更多的手写应用场景。支持多种笔刷（圆珠笔、钢笔、铅笔、马克笔）选择、多色调色版；能设置笔刷颜色、粗细、撤销、恢复、擦除、套索等笔迹编辑能力。应用利用多笔刷可满足用户日常所有的笔记和绘图、涂鸦等需求，能达到较好的文字书写体验。

应用场景：Pencil Engine 可快速与应用集成，在应用中插入一个手写入口（如添加画板或标注），点击进入 Pencil Engine 编辑手写绘画，可保存和二次编辑修改。

隔空手势

隔空手势（非接触手势）是一种人与设备交互的新方式，用户可以在无需手持或接触设备的情况下与设备进行便捷的交互。随着技术的发展，隔空手势在手机、平板、车机、电视、音箱、AR/VR 等设备上都有一定的应用。

根据用户与交互界面/设备的空间维度上的关系，我们可以将隔空手势交互大致划分为三类：间接（光标）交互、直接（虚拟手）交互、以及指令性交互。

间接（远距离）交互

在间接交互的情况下，用户与交互界面/交互设备存在一定的距离，系统会通过光标将用户的隔空手势操作映射到系统界面上，并以特定的方式进行视觉反馈，以表达手势操控的对象的可交互性。通常包括导航移动、确认响应等手势。

操作	功能描述	手势描述	手势动作
导航	定位对象，即移动（光标/焦点）到用户想要操作的控件上	食指伸出，其余指头握拳，移动手即可导航。	
单击	当用户移动到目标控件后，用户需要触发控件功能，即点击目标控件	拇指食指触碰并释放	
双击	当用户移动到目标控件后，用户需要触发目标控件的其他功能	拇指食指触碰两次并释放	

长按	当用户移动到目标控件后，需要长时间按压控件以触发特定功能	拇指食指捏握超过一定时长（默认为500ms）	
拖拽	拖拽对象或在一定区域内选择对象	拇指与食指捏合，向左/向右或向上/向下移动。用户可连续滚动以控制界面上的元素，如进度条、音量条、页面滚动条等	
激活	特定的激活手势来触发用户进入到非接触手势交互状态	未进入非接触手势交互状态前，食指拇指触碰两次	

直接（虚拟手）交互

在直接交互的情况下，系统中能真实模拟出用户的双手，用户自己所见即为操作者所见。虚拟手势会随着手的移动而发生移动，虚拟手势的手型变化也会随着真实手型发生变化。

操作	功能描述	手势描述	手势动作
悬空	手势不与任何控件发生交互时只是在虚拟空间中发生移动	手呈自然状态，在空间中发生移动	
悬浮靠近	当准备与某一控件发生交互时，需要通过靠近控件以用来进行选择	手靠近待交互的控件	
按压	当需要触发控件功能时，需要通过按压手势来响应特定功能	手势接触控件并向内推动	
捏握	当需要移动被操作物体对象时	手接触控件表面，并将食指拇指捏合	

指令性交互

手势是指通过某些特定的手势来快速完成一个操作，而不需要通过与系统上控件进行一系列交互。快捷手势通常是符合用户直觉、文化习惯或者容易操作的动作，快捷手势对应的功能一般为用户常用的高频功能。

操作	功能描述	手势描述	手势动作
静音	系统音量静音或者将静音恢复	将食指靠近唇部中间	
菜单激活	通过特定手势触发弹出菜单，快捷菜单可以快速的调起该应用下的高频功能菜单，该菜单可根据应用场景的不同进行定义。弹出菜单后，用户可接着通过非接触手势的方式与菜单上的控件元素进行交互，触发相应功能	握拳展开成手掌	
菜单选择	菜单控件弹出后，选择用户需要操作的功能	移动手即可切换菜单中选择的焦点	
菜单确定	选择的到需要操作的功能后，需要用户确认触发，才可执行相应功能	将手掌握拳	

更多鸿蒙技术文章、课程、直播，都在 [HarmonyOS社区](#)



