

公共控件使用说明

公共控件版本信息

版本号	修订人	修订时间	控件 aar 包变更内容	使用文档变更 新内容
V7.0.18	张冠群	2024/12/27	1.调整 RecyclerView 抛滑时的速度缩放因子	
V7.0.17	张冠群	2024/12/24	1. 将 RecyclerView 的一些方法设为 public 供业务定制 2. 调整 RecyclerView 旋转表冠触顶和触底时的震感	
V7.0.16	张冠群	2024/12/9	1.修复 RecyclerView 滑动停止时结尾有明显顿挫感的问题	
V7.0.15	张冠群	2024/11/29	1. 开关控件点击 item 振动效果引入的内存泄漏	
V7.0.14	张冠群	2024/11/28	1. 新增字体使用说明 2. 解决 HeyNumberPicker 的 crash 问题 3. 解决 RecyclerView 滑到底部时有顿挫感的问题 4. 去掉 HeyNumberPicker 滚动时的声音	
V7.0.13	张冠群	2024/11/23	1. 优化 item 动画 2. 开关控件点击 item 振动效果 3. 解决状态栏编辑按钮点击无响应的问题 4. 修改 RecyclerView 底部间距	
V7.0.12	张冠群	2024/11/14	1.解决 VolumeSeekBar 内存泄漏的问题 2.表冠控制音量调节做无极效果适配 3.解决 RecyclerView 显示异常的问题 4.解决 HeyNumberPicker 快速转动表冠时不振动的问题	
V7.0.11	张冠群	2024/11/11	1. 修改 RecyclerView 部分页面的卡片相互叠在一起的问题	

V7.0.10	张冠群	2024/10/31	1. 为 HeyViewPager 添加是否响应表冠的接口 2. 解决 ViewPager 滑动时偶现黑屏的问题	
V7.0.9	张冠群	2024/10/25	1.解决 ViewPager 滑动时偶现黑屏的问题 2.解决 RecyclerView 滑动时 item 位置异常的问题	
V7.0.8	张冠群	2024/10/23	1.解决 VolumeSeekBar 滑动调节音量失败的问题	
V7.0.7	张冠群	2024/10/21	1.解决 RecyclerView 滚动在停靠下来的时候有一个不自然的吸附动画的问题	
V7.0.6	张冠群	2024/10/19	1.解决 RecyclerView 的 crash 问题	
V7.0.5	张冠群	2024/10/18	1.解决 RecyclerView 滑到底部, 首次点击 item 无响应的问题 2.解决 RecyclerView 慢滑时出现抛滑动画的问题 3.解决 HeyNumberPicker 出现 crash 的问题 4.解决 RecyclerView 导致的标题位置偏下的问题	
V7.0.4	张冠群	2024/9/16	1. 适配不支持高压马达下开关振动	
V7.0.3	张冠群	2024/9/13	1. 音量条更新为 7.0 的风格	
V7.0.2	张冠群	2024/9/2	1. 新增 ViewPager2 指示器 2. 更新 RecyclerView 列表滚动的效果	
V7.0.1	张冠群	2024/7/15	1.选择器风格修改 2.系统色彩更新 3.系统图标更新 4.系统按钮更新 5.开关, 单选, 多选更新	

1.主标题栏：HeyMainTitleBar.....5

2.返回标题栏：HeyBackTitleBar.....6

3.取消标题栏：HeyCancelTitleBar7

4.加载标题栏：HeyLoadingTitleBar8

5.单行默认列表：HeySingleDefaultItem.....9

6.单行带开关列表: HeySingleItemWithSwitch.....	10
7.单行带复选框列表: HeySingleItemWithCheckBox	11
8.单行两个图标列表: HeySingleDefaultItem.....	12
9.单行带左边图标列表: HeySingleItemLeftImage	13
10.单行仅带左边图标列表: HeySingleItemOnlyLeftImage	14
11.双行默认列表: HeyMultipleDefaultItem	15
12.双行带开关列表: HeyMultipleItemWithSwitch.....	17
13.双行带复选框列表: HeyMultipleItemWithCheckBox	18
14.双行两个图标列表: HeyMultipleDefaultItem	19
15.双行左边图标列表: HeyMultipleItemWithLeftImage.....	20
16.双行带开关列表 1: HeyMultipleSwitchSummaryAlone	21
17.双行带复选框列表 1: HeyMultipleCheckBoxSummaryAlone.....	22
18.双行默认列表 1: HeyMultipleDefaultSummaryAlone	23
19.圆角按钮: HeyShapeButton.....	24
20.图标按钮: HeyImageButton	26
21.下载按钮: HeyProgressButton	28
22.单组数字选择器: HeyNumberPicker.....	31
23.上、下午数字选择器: HeyTimePicker	32
24.双组数字选择器: HeyTimePicker	33
25.时分秒选择器: HeyCounterPicker.....	35
26.带单位选择器: HeyNumberPicker.....	36
27.翻页符: HeyPageIndicator	37

28.对话框(一个按钮): HeyDialog	41
29.对话框(两个按钮): HeyDialog	42
30.对话框(权限隐私申明): HeyDialog	43
31.对话框(自定义布局): HeyDialog	44
32.开关: HeySwitch.....	45
33.复选框: HeyCheckBox	47
34.单选框: RadioButton	48
35.消息提示: HeyToast	48
36.音量滑块: HeyControllerView.....	49
37.加载对话框: HeyLoadingDialog	52
38.抗锯齿圆角图片: HeyRoundImageAntiAlias.....	53
39.分组小标题: HeyGroupTitle	54
40.回弹列表 RecyclerView/ListView/ScrollView	55
41.脚注: HeyFootNoteView.....	57
42.说明: HeyInformationView	58
43.卡片: HeyBackgroundView	59
44.输入框: HeyEditTextStyle	60
45.音量条: HeySeekBar	61
46.ViewPager: HeyViewPager.....	62
47.动画差值器.....	62
48.右滑主题 1: HeytapTheme	63
49.右滑主题 2: HeytapCompatTheme.....	64
50.主题颜色.....	65

51.字体.....	66
52.常用动画.....	67
53.大小表适配.....	68
54.抗锯齿优化.....	71
55.控件 6.0 版本性能优化	72
56. 5.0 兼容性判断	72
57.圆表相关控件.....	73
1. 默认文字按钮	73
2. 默认图标按钮	73
3. 圆形默认按钮	74
4. 圆形小按钮	74
5. 圆形超小按钮	75
6. 圆形大按钮	76
7. HeyProcessSpinnerIndicator 进度提示控件.....	76
6. 圆表时间控件	79
7. 圆表 Dialog 左右按钮:	80
10. 圆表列表样式变更	80

1.主标题栏：HeyMainTitleBar

效果图：



使用方法：

```
<com.heytap.wearable.support.widget.HeyMainTitleBar
    android:id="@+id/default_titlebar_notime"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    app:heyText="@string/hey_title_bar"
    app:heyShowClock="true"/>
```

使用说明：

- 1.标题文本通过属性 `app:heyText` 控制, 也可以通过方法 `setTitle(title)` 控制
- 2.是否显示时间由属性 `app:heyShowClock` 控制, true 表示显示, false 表示不显示, 默认为 false, 不显示

2.返回标题栏：HeyBackTitleBar

效果图：



使用方法：

```
<com.heytap.wearable.support.widget.HeyBackTitleBar
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    app:heyShowClock="false"
    app:heyText="@string/hey_title_bar" />
```

使用说明：

- 1.标题文本通过属性 `app:heyShowClock` 控制，也可以通过方法 `setTitle(title)` 控制
- 2.返回操作可以通过方法 `setBackListener(null,activity)`控制
- 3.是否显示时间由属性 `app:heyShowClock` 控制，true 表示显示，false 表示不显示，默认为 false，不显示
- 4.是否显示返回箭头通过属性 `heyShowIcon` 控制，true 表示显示，false 表示不显示，默认为 true,显示

3.取消标题栏：HeyCancelTitleBar

效果图：



使用方法：

```
<com.heytap.wearable.support.widget.HeyCancelTitleBar  
    android:layout_width="wrap_content"  
    android:layout_height="wrap_content"  
    app:heyText="@string/hey_cancel"/>
```

使用说明：

- 1.标题文本通过属性 `app:heyText` 控制，也可以通过方法 `setTitle(title)` 控制
- 2.返回操作可以通过方法 `setBackListener(null,activity)`控制

4.加载标题栏：HeyLoaddingTitleBar

效果图：



使用方法：

```
<com.heytap.wearable.support.widget.HeyLoaddingTitleBar  
    android:layout_width="wrap_content"
```



```
android:layout_height="wrap_content"
app:heyText="正在加载"/>
```

使用说明：

- 1.标题文本通过属性 `app:heyText` 控制，也可以通过方法 `setTitle(title)` 控制

5.单行默认列表：HeySingleDefaultItem

效果图：



使用方法：

```
<com.heytap.wearable.support.widget.HeySingleDefaultItem
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
```

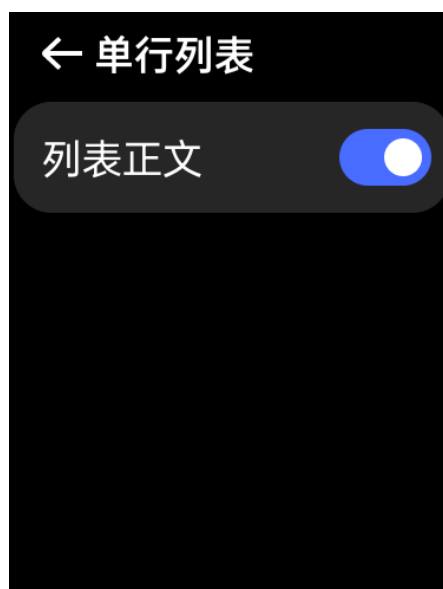
```
style="@style/HeySingleDefaultItemStyle"  
app:heyText="@string/hey_list_content"/>
```

使用说明：

- 1.列表正文通过属性 `app:heyText` 控制，也可以通过方法 `setTitle(title)` 控制
- 2.最右边图标可以通过 `app:heyPrimaryImage` 控制，也可以通过 `getPrimaryImage` 控制

6.单行带开关列表：HeySingleItemWithSwitch

效果图：



使用方法：

```
<com.heytao.wearable.support.widget.HeySingleItemWithSwitch  
style="@style/HeySingleDefaultItemStyle"
```

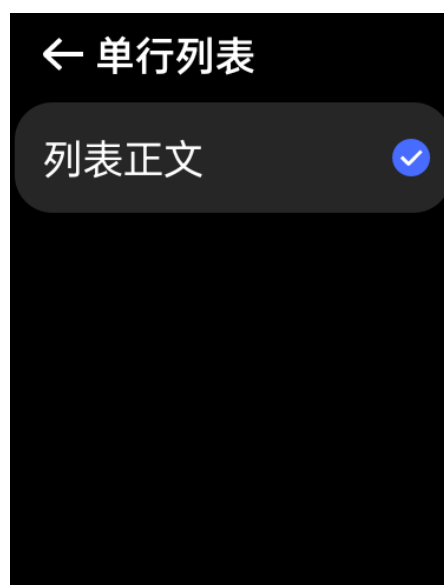
```
android:layout_width="match_parent"
android:layout_height="wrap_content"
app:heyText="@string/hey_list_content"/>
```

使用说明：

- 1.开关可以通过 `getHeySwitch` 获取
- 2.设置控件在点击卡片时是否需要改变开关状态，`setSupportClickCard` (boolean) ，默认为 true

7.单行带复选框列表：HeySingleItemWithCheckBox

效果图：



使用方法：

```
<com.heytap.wearable.support.widget.HeySingleItemWithCheckBox  
    style="@style/HeySingleDefaultItemStyle"  
    android:layout_width="match_parent"  
    android:layout_height="wrap_content"  
    app:heyText="@string/hey_list_content"/>
```

使用说明：

- 1.列表正文通过属性 `app:heyText` 控制，也可以通过方法 `setTitle(title)` 控制
- 2.复选框可以通过 `getCheckBox` 获取

8.单行两个图标列表：HeySingleDefaultItem

效果图：



使用方法：

```
<com.heytap.wearable.support.widget.HeySingleDefaultItem
    style="@style/HeySingleDefaultItemStyle"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    app:heyText="@string/hey_list_content"
    app:heyPrimaryImage="@drawable/hey_wifi"
    app:heyShowMinorImage="true"
    app:heyMinorImage="@drawable/hey_lock"/>
```

使用说明：

- 1.列表正文通过属性 `app:heyText` 控制，也可以通过方法 `setTitle(title)` 控制
- 2.最右边图标可以通过 `app:heyPrimaryImage` 控制，也可以通过 `getPrimaryImage` 控制
- 3.显示两个图标先将 `app:heyShowMinorImage` 设置成 `true`，再通过 `app:heyMinorImage` 指定第二张图片，也可以通过 `getMinorImage` 控制

9.单行带左边图标列表：HeySingleItemLeftImage

效果图：



使用方法：

```
<com.heytap.wearable.support.widget.HeySingleItemLeftImage  
    style="@style/HeySingleDefaultItemStyle"  
    android:layout_width="match_parent"  
    android:layout_height="wrap_content"  
    app:heyText="@string/hey_list_content"  
/>
```

使用说明：

- 1.列表正文通过属性 `getContentTextView` 控制
- 2.左边图片可以通过 `setLeftImageView` 控制，暂时不支持 xml 配置图片，如果需求迫切，后期可以添加支持
- 3.右边图片可以 `getRightImageView` 进行设置，暂时不支持 xml 配置图片，如果需求迫切，后期可以添加支持

10.单行仅带左边图标列表： `HeySingleItemOnlyLeftImage`

效果图：



使用方法：

```
<com.heytap.wearable.support.widget.HeySingleItemOnlyLeftImage  
    style="@style/HeySingleDefaultItemStyle"  
    android:layout_width="match_parent"  
    android:layout_height="wrap_content"  
    app:heyText="@string/hey_list_content"  
>
```

使用说明：

4.列表正文通过属性 `getContentTextView` 控制

5.左边图片可以通过 `setLeftImageView` 控制，暂时不支持 xml 配置图片，如果需求迫切，后期可以添加支持

11.双行默认列表：HeyMultipleDefaultItem

效果图：



使用方法：

```
<com.heytap.wearable.support.widget.HeyMultipleDefaultItem
    android:visibility="visible"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    app:heyText="@string/hey_list_content"
    app:heySummaryText="@string/hey_list_summary"
    app:heyShowIcon="true"
    app:heyPrimaryImage="@drawable/hey_arrow"
    style="@style/HeyMultipleDefaultItemStyle"/>
```

使用说明：

- 1.列表正文通过属性 **app:heyText** 控制，也可以通过 **setTitle(String title)** 控制
- 2.列表摘要文本可以通过 **app:heySummaryText** 控制，也可以通过 **setSummary(String summary)** 控制
- 3.右边图标可以通过 **app:heyShowIcon** 控制显示与隐藏，true 为显示，false 为隐藏，默认为 true，显示图片；
- 4.图片可以通过 **app:heyPrimaryImage** 设置，也可以通过 **getPrimaryImageView** 控制

12.双行带开关列表：HeyMultipleItemWithSwitch

效果图：



使用方法：

```
<com.heytap.wearable.support.widget.HeyMultipleItemWithSwitch
    style="@style/HeyMultipleDefaultItemStyle"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    app:heyText="列表正文的列表"
    app:heySummaryText="辅助文字"
/>
```

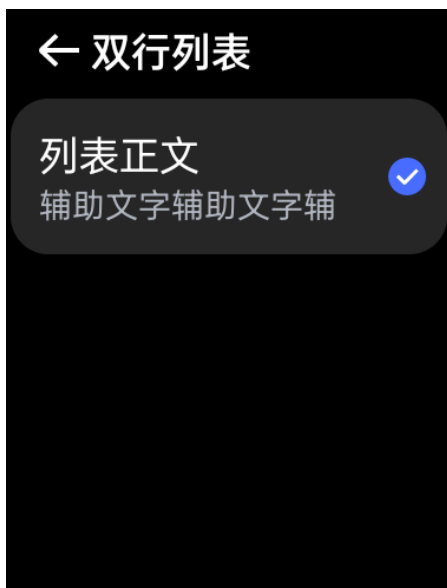
使用说明：

- 1.列表正文通过属性 `app:heyText` 控制，也可以通过方法 `setTitle(title)` 控制
- 2.列表摘要文本可以通过 `app:heySummaryText` 控制，也可以通过 `setSummary(summary)` 控制
- 3.开关可以通过 `getHeySwitch` 获取
4. 设置控件在点击卡片时是否需要改变开关状态，`setSupportClickCard` (boolean) ，默认为 true

PS:V5.0.2 引入开启振动效果的接口 `setVibrateEnable`, false 代表点击不振动, true 代表点击振动

13.双行带复选框列表: HeyMultipleItemWithCheckBox

效果图:



使用方法:

```
<com.heytap.wearable.support.widget.HeyMultipleItemWithCheckBox
    style="@style/HeyMultipleItemWithCheckBoxStyle"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    app:heyText="列表正文"
    app:heySummaryText="辅助文字辅助文字辅助"
/>
```

使用说明:

- 1.列表正文通过属性 `app:heyText` 控制, 也可以通过方法 `setTitle(title)` 控制
- 2.列表摘要文本可以通过 `app:heySummaryText` 控制, 也可以通过 `setSummary(summary)` 控制
- 3.复选框可以通过 `getCheckBox` 获取
- 4.复选框状态可以通过 `app:heyChecked` 控制, 如果为 `true`, 则为选中字体, 如果是 `false`, 则为未选中; 默认是 `true`, 选中状态

14.双行两个图标列表：HeyMultipleDefaultItem

效果图：



使用方法：

```
<com.heytao.wearable.support.widget.HeyMultipleDefaultItem
    style="@style/HeyMultipleDefaultItemStyle"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    app:heyText="列表正文"
    app:heySummaryText="辅助文字"
    app:heyPrimaryImage="@drawable/hey_wifi"
    app:heyShowMinorImage="true"
    app:heyMinorImage="@drawable/hey_lock"/>
```

使用说明：

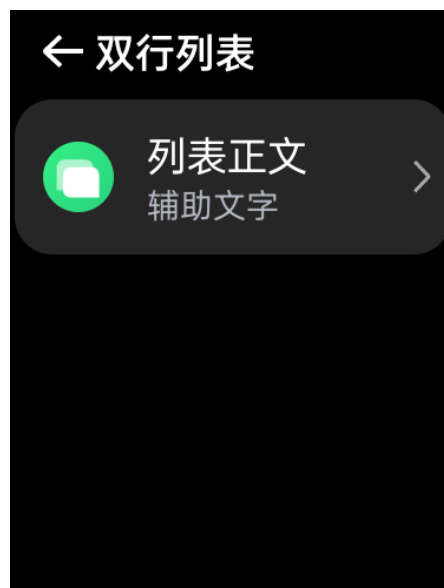
- 1.列表正文通过属性 `app:heyText` 控制，也可以通过方法 `setTitle(title)` 控制
- 2.列表摘要文本可以通过 `app:heySummaryText` 控制，也可以通过 `setSummary(summary)` 控制
- 3.最右边的图标可以通过 `app:heyPrimaryImage` 控制，也可以通过 `getPrimaryImage` 控制

4.右边靠左的图标可以通过 `app:heyMinorImage` 控制，也可以通过 `getMinorImage` 控制，前提是先将 `app:heyShowMinorImage` 设置成 `true`

15.双行左边图标列

表:HeyMultipleItemWithLeftImage

效果图:



使用方法:

```
<com.heytap.wearable.support.widget.HeyMultipleItemWithLeftImage
    style="@style/HeyMultipleItemWithLeftImageStyle"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    app:heyText="文本列表"
    app:heySummaryText="辅助文字"
    app:leftImage="@drawable/msg"
/>
```

使用说明:

- 1.列表正文通过属性 `app:heyText` 控制，也可以通过方法 `getContentTextView` 控制
- 2.列表摘要文本可以通过 `app:heySummaryText` 控制，也可以通过 `getSummaryTextView` 控制

- 3.左边图标可以通过 `app:leftImage` 控制，也可以通过 `getLeftImageView` 控制
- 4.右边图标可以通过 `getPrimaryImageView` 控制

16.双行带开关列表 1: HeyMultipleSwitchSummaryAlone

效果图:



使用方法:

```
<com.heytap.wearable.support.widget.HeyMultipleSwitchSummaryAlone
    style="@style/HeyMultipleDefaultItemStyle"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    app:heyText="列表正文"
    app:heySummaryText="辅助文字辅助文字辅助文字"
/>
```

使用说明:

- 3.列表正文通过属性 `app:heyText` 控制，也可以通过方法 `setTitle(title)` 控制
- 4.列表摘要文本可以通过 `app:heySummaryText` 控制，也可以通过 `setSummary(summary)` 控制
- 5.开关可以通过 `getHeySwitch` 获取

6. 设置控件在点击卡片时是否需要改变开关状态，`setSupportClickCard` (boolean) ，默认为 true

PS:V5.0.2 引入开启振动效果的接口 `setVibrateEnable`，false 代表点击不振动，true 代表点击振动

17.双行带复选框列表 1: HeyMultipleCheckBoxSummaryAlone

效果图：



使用方法：

```
<com.heytap.wearable.support.widget.HeyMultipleItemWithCheckBox
    style="@style/HeyMultipleItemWithCheckBoxStyle"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    app:heyText="列表正文"
    app:heySummaryText="辅助文字辅助文字辅助文字"
/>
```

使用说明：

- 2.列表正文通过属性 `app:heyText` 控制，也可以通过方法 `setTitle(title)` 控制
- 2.列表摘要文本可以通过 `app:heySummaryText` 控制，也可以通过 `setSummary(summary)` 控制
- 3.复选框可以通过 `getCheckBox` 获取
- 4.复选框状态可以通过 `app:heyChecked` 控制，如果为 `true`，则为选中字体，如果是 `false`，则为未选中；默认是 `true`，选中状态

18.双行默认列表 1: HeyMultipleDefaultSummaryAlone

效果图：



使用方法：

```
<com.heytap.wearable.support.widget.HeyMultipleDefaultItem
    style="@style/HeyMultipleDefaultItemStyle"
    android:visibility="visible"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    app:heyText="@string/hey_list_content"
    app:heySummaryText="@string/hey_list_summary"
    app:heyShowIcon="true"
    app:heyPrimaryImage="@drawable/hey_arrow"
    style="@style/HeyMultipleDefaultItemStyle"/>
```

使用说明：

2.列表正文通过属性 `app:heyText` 控制，也可以通过 `setTitle(String title)` 控制

2.列表摘要文本可以通过 `app:heySummaryText` 控制，也可以通过 `setSummary(String summary)` 控制

3.右边图标可以通过 `app:heyShowIcon` 控制显示与隐藏，true 为显示，false 为隐藏，默认为 true，显示图片；

4.图片可以通过 `app:heyPrimaryImage` 设置，也可以通过 `getPrimaryImageView` 控制

19.圆角按钮：HeyShapeButton

1.默认按钮

效果图：



使用方法：

```
<com.heytap.wearable.support.widget.HeyShapeButton  
    style="@style/HeyBigShapeButtonStyle"  
>
```

使用说明：

1. HeyShapeButton 继承自 Button，按钮文字的设置方法跟 Button 一样
2. 背景颜色可以通过 `app:heyBackgroundColor` 设置，也可以通过 `setHeytapBg` 设置
3. 默认效果使用 `style="@style/HeyBigShapeButtonStyle"` 即可
4. 拨号盘样式: `style="@style/HeyDialpadButtonStyle"`

2. 小按钮

效果图：



使用方法：

```
<RelativeLayout
    android:layout_width="match_parent"
    android:layout_height="wrap_content">
    <com.heytap.wearable.support.widget.HeyShapeButton
        style="@style/HeySmallShapeButtonStyle"
        android:layout_alignParentLeft="true" />

    <com.heytap.wearable.support.widget.HeyShapeButton
```

```
style="@style/HeySmallShapeButtonStyle"
    android:layout_alignParentRight="true"/>
</RelativeLayout>
```

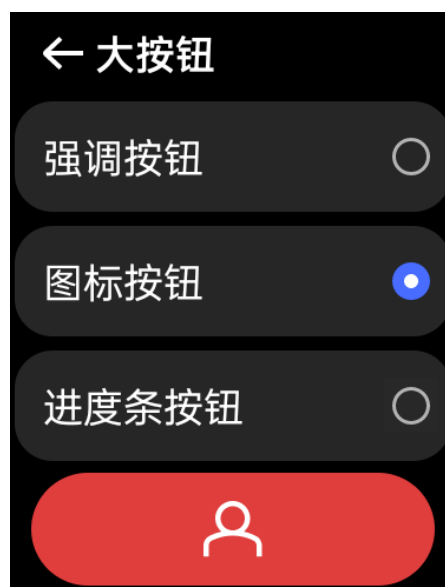
使用说明：

1. HeyShapeButton 继承自 Button，按钮文字的设置方法跟 Button 一样
2. 背景颜色可以通过 `app:heyBackgroundColor` 设置，也可以通过 `setHeytapBg` 设置
3. 小按钮默认效果使用 `style="@style/HeySmallShapeButtonStyle"` 即可
4. 如果要使用一行显示两个小按钮，直接使用上面的布局即可

20. 图标按钮：HeyImageButton

1. 腰圆按钮

效果图：



使用方法：

```
<com.heytap.wearable.support.widget.HeyImageButton
    android:src="@drawable/me"
    android:background="#FFFF8B26"
```

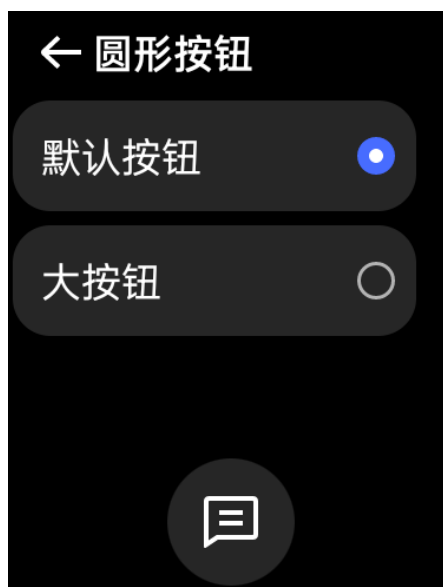
```
style="@style/HeyOvalButtonStyle"
/>
```

使用说明:

- 1.HeyImageButton 继承自 ImageButton 各个属性和方法跟 ImageButton 一样
- 2.默认效果可以使用 `style="@style/HeyOvalButtonStyle"`,应用只需要修改背景和 src 即可

2.圆形按钮

效果图



使用方法:

```
<com.heytao.wearable.support.widget.HeyImageButton
    style="@style/HeySmallCircleButtonStyle"
    android:background="@color/default_background_color"
    android:src="@drawable/hey_information"
/>
```

使用说明：

1.HeyImageButton 继承自 ImageButton 各个属性和方法跟 ImageButton 一样

2.默认效果可以使用 `style="@style/HeySmallCircleButtonStyle"`,应用只需要修改背景和 src 即可

21.下载按钮： HeyProgressButton

效果图：



使用方法：

Xml

```
<com.heytap.wearable.support.widget.HeyProgressButton
    android:id="@+id/download_btn"
    android:layout_width="match_parent"
    android:layout_height="48dp"/>
```

Java

```
mHeyProgressBtn.setOnClickListener(new View.OnClickListener() {  
    @Override  
    public void onClick(View view) {  
        switch (mStateType) {  
            case DownloadState.STATE_DEFAULT:  
                mProgress = 0;  
                mHeyProgressBtn.setProgress(mProgress);  
                mStateType = DownloadState.STATE_DOWNLOADING;  
                mHeyProgressBtn.setState(mStateType);  
                mDownloadHandler.sendEmptyMessageDelayed(0, 500);  
                break;  
  
            case DownloadState.STATE_DOWNLOADING:  
                mStateType = DownloadState.STATE_PAUSE;  
                mHeyProgressBtn.setState(mStateType);  
                mDownloadHandler.removeMessages(0);  
                break;  
  
            case DownloadState.STATE_PAUSE:  
                mStateType = DownloadState.STATE_DOWNLOADING;  
                mHeyProgressBtn.setState(mStateType);  
                mDownloadHandler.sendEmptyMessageDelayed(0, 500);  
                break;  
  
            case DownloadState.STATE_DOWNLOAD_FINISH:  
                mStateType = DownloadState.STATE_DEFAULT;  
                mHeyProgressBtn.setState(mStateType);  
                break;  
        }  
    }  
});  
  
private static class DownloadHandler extends Handler {  
  
    private WeakReference<Context> reference;  
  
    DownloadHandler(Context context) {  
        reference = new WeakReference<>(context);  
    }  
  
    @Override  
    public void handleMessage(Message msg) {  
        TestColorButton downloadActivity = (TestColorButton)
```

```

reference.get();
    if (downloadActivity != null) {
        switch (msg.what) {
            case 0:
                if (downloadActivity.mProgress < 100) {
                    downloadActivity.mProgress += 1.0;

downloadActivity.mHeyProgressBtn.setProgress(downloadActivity.mProgress);

downloadActivity.mDownloadHandler.sendMessageDelayed(0, 300);
                } else {
                    downloadActivity.mStateType =
DownloadState.STATE_DOWNLOAD_FINISH;

downloadActivity.mHeyProgressBtn.setState(downloadActivity.mStateType);
                }
                break;

            default:
                break;
        }
    }
}
}
}

```

使用说明：

此按钮继承自 View，自定义了部分属性：

```
<declare-styleable name="HeyProgressButton">
```

```
<!--按钮背景， layer-list 文件配置按钮圆角和颜色-->
```

```
<attr name="progressButtonBackground" format="dimension"/>
```

```
<!--字体大小-->
```

```
<attr name="progressbtnTextSize" format="dimension"/>
```

```
<!--默认状态字体颜色-->
```

```
<attr name="progressbtnTextColor" format="color"/>
```

```
<!--下载中字体颜色-->
```

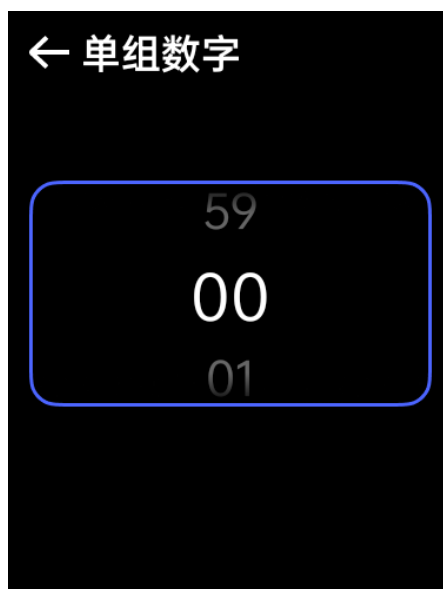
```
<attr name="progressbtnDownloadTextColor" format="color|reference" />
```

```
<!--按钮文字-->
```

```
<attr name="progressbtnText" format="string|reference" />
</declare-styleable>
```

22.单组数字选择器：HeyNumberPicker

效果图：



使用方法：

```
<com.heytap.wearable.support.widget.HeyNumberPicker
    android:id="@+id/number_picker"
    android:layout_width="match_parent"
    android:layout_height="@dimen/hey_widget_number_picker"
```

```
app:heyTextArray="@array/minute"
/>
```

使用说明：

heyTextArray 是时间选择器需要显示的内容，数组格式

23.上、下午数字选择器：HeyTimePicker

效果图：



使用方法：

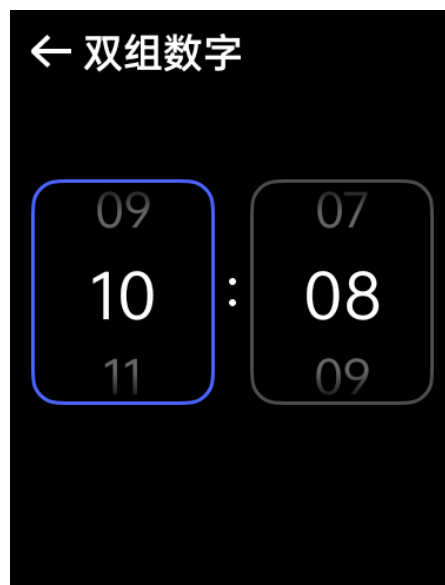
```
<com.heytap.wearable.support.widget.HeyTimePicker
    android:id="@+id/time_picker_2"
    android:layout_gravity="center"
    android:layout_width="match_parent"
    android:layout_height="@dimen/hey_widget_number_picker"/>
```

使用说明：

此控件继承自 `FrameLayout`，本质上是由 3 个 `HeyNumberPicker` 组合而成，并没有自定义任何属性

24.双组数字选择器：HeyTimePicker

效果图：



使用方法：

```
<LinearLayout
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_marginTop="15dp"
    android:layout_marginHorizontal=
        "@dimen/hey_content_horizontal_distance"
    android:gravity="center_vertical"
    android:orientation="horizontal">
```

```
<com.heytap.wearable.support.widget.HeyNumberPicker
    android:id="@+id/hour"
    app:heyTextArray="@array/hour_12"
    style="@style/HeyNormalPickerStyle"/>

<ImageView
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:src="@drawable/hey_colon"
    android:layout_marginHorizontal=
        "@dimen/hey_picker_colon_margin_horizontal"
/>

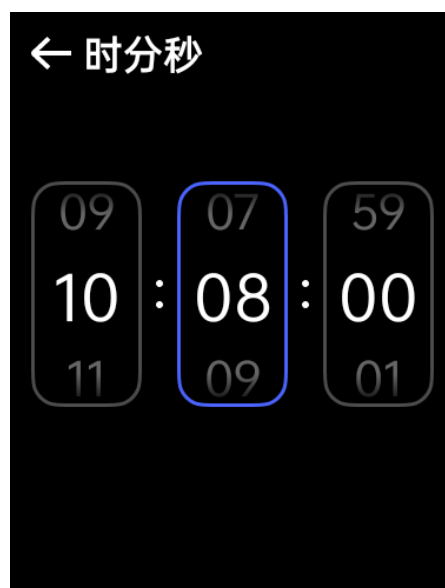
<com.heytap.wearable.support.widget.HeyNumberPicker
    android:id="@+id/minute"
    app:heyTextArray="@array/minute"
    style="@style/HeyNormalPickerStyle"/>
</LinearLayout>
```

使用说明：

此选择器是由两个 HeyNumberPicker 和一个 ImageView 拼接而成，应用可以根据实际需要调整样式；**需要注意的是，如果要显示冒号的话，最好让设计提供图片，不要用 TextView 显示冒号，因为使用 TextView 的话，垂直方向很难对齐的**

25.时分秒选择器：HeyCounterPicker

效果图：



使用方法：

```
<com.heytap.wearable.support.widget.HeyCounterPicker  
    android:id="@+id/count_picker"  
    android:layout_width="wrap_content"  
    android:layout_height="@dimen/hey_widget_number_picker"  
    android:layout_gravity="center"/>
```

使用说明：

此控件继承自 FrameLayout，本质上是由 3 个 HeyNumberPicker 和三个 TextView 组合而成，并没有自定义任何属性。

26.带单位选择器：HeyNumberPicker

效果图：



使用方法：

```
<TextView
    style="@style/HeyUnitStyle"
    android:layout_marginBottom="3dp"
    android:layout_marginTop="16dp"
    android:text="单位" />

<com.heytap.wearable.support.widget.HeyNumberPicker
```

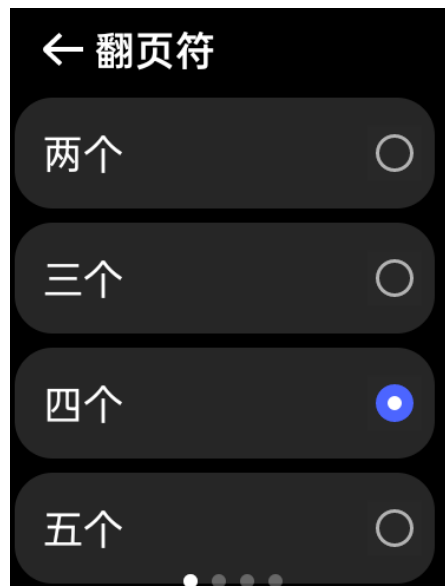
```
android:id="@+id/number_picker"
style="@style/HeyLargePickerStyle"
android:layout_marginVertical="0dp"
android:layout_marginHorizontal=
    "@dimen/hey_content_horizontal_distance"
app:heyTextArray="@array/minute" />
```

使用说明：

此效果由一个 TextView 和一个 HeyNumberPicker 组合而成，没有做成一个控件，TextView 统一使用 HeyUnitStyle 这个样式，内容根据实际情况调整

27.翻页符：HeyPageIndicator

效果图：



使用方法：

```
<FrameLayout xmlns:android="http://schemas.android.com/apk/res/android"
xmlns:app="http://schemas.android.com/apk/res-auto"
android:background="@drawable/bgg"
android:layout_width="match_parent"
android:layout_height="match_parent">
```

```
<androidx.viewpager.widget.ViewPager
    android:id="@+id/viewpager"
    android:background="@drawable/bgg"
    android:layout_width="match_parent"
    android:layout_height="match_parent" />
```

```
<com.heytap.wearable.support.widget.pageindicator.HeyPageIndicator
    android:id="@+id/indicator"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    style="@style/HeyPageIndicatorStyle"
/>
```

```
</FrameLayout>
```

```
private void initView() {
    final ViewPager viewPager = findViewById(R.id.viewpager);
    final HeyPageIndicator indicator = findViewById(R.id.indicator);
    mBannerAdapter = new ImagePagerAdapter();
    viewPager.setAdapter(mBannerAdapter);
    viewPager.addOnPageChangeListener(new ViewPager.OnPageChangeListener() {
        @Override
        public void onPageScrolled(int position, float positionOffset, int positionOffsetPixels) {
            indicator.onPageScrolled(position, positionOffset, positionOffsetPixels);
        }

        @Override
        public void onPageSelected(int position) {
            indicator.onPageSelected(position);
        }

        @Override
        public void onPageScrollStateChanged(int state) {
            indicator.onPageScrollStateChanged(state);
        }
    });
    indicator.setDotsCount(mBannerAdapter.getCount());
    indicator.setOnDotClickListener(new HeyPageIndicator.OnIndicatorDotClickListener() {
        @Override
        public void onClick(int position) {
            viewPager.setCurrentItem(position);
        }
    });
}
```

```
});  
}
```

使用说明：

继承自 `FrameLayout`，目前仅支持与 `ViewPager` 配合使用，不支持 `RecyclerView`，新增了部分自定义属性：

```
<declare-styleable name="HeyPageIndicator">  
    <!--指示器的颜色，即上图下方后面两个小圆圈的颜色-->  
    <attr name="dotColor" format="color"/>  
    <!--过渡器颜色，即上图第一个圆圈的填充颜色-->  
    <attr name="traceDotColor" format="color"/>  
    <!--指示器大小，即上图中底部小圆圈的大小，指示器是由正方形设置圆角而来-->  
    <attr name="dotSize" format="dimension"/>  
    <!--指示器之间的距离，即上图中底部小圆圈之间的距离-->  
    <attr name="dotSpacing" format="dimension"/>  
    <!--指示器的描边宽度-->  
    <attr name="dotStrokeWidth" format="dimension"/>  
    <!--指示器圆角半径，默认是 dotSize/2-->  
    <attr name="dotCornerRadius" format="dimension"/>  
    <!--指示器是否可以点击，点击指示器具有切换界面的效果-->  
    <attr name="dotClickable" format="boolean"/>  
</declare-styleable>
```

圆表适配：

可以配合 `ViewPager2` 使用，支持竖向排布。用法与上面保持一致，支持属性有些变化。

```
<style name="HeyPageIndicatorStyle">  
    <!--圆圈方向（横竖）-->  
    <item name="dotOrientation">horizontal</item>  
    <!--圆圈直径-->  
    <item name="dotSize">@dimen/hey_pageindicator_dot_size</item>  
    <!--圆圈距离底部的间距-->  
    <item name="dotRoundMargin">@dimen/hey_wear_page_indicator_padding</item>  
    <!--圆圈被选中的颜色-->  
    <item name="traceDotColor">@color/hey_page_color_selected</item>  
    <!--圆圈未选中的颜色-->  
    <item name="dotColor">@color/hey_page_color</item>  
    <!--圆圈之间的角度-->  
    <item name="dotRoundAngle">5</item>  
</style>
```

代码与之对应

```
indicator.setOrientation(HeyPageIndicator. ORIENTATION_HORIZONTAL);

indicator.setDotSize(getResources().getDimensionPixelSize(com.heytap.wearabl
e.R.dimen.hey_page_indicator_dot_size));

indicator.setRoundDotMargin(getResources().getDimensionPixelSize(com.heytap.
wearable.R.dimen.hey_wear_page_indicator_padding));

indicator.setTraceDotColor(getResources().getColor(com.heytap.wearable.R.col
or.hey_wear_page_indicator_selected_color));

indicator.setPageIndicatorDotsColor(getResources().getColor(com.heytap.weara
ble.R.color.hey_wear_page_indicator_unselected_color));

indicator.setRoundAngle(HeyWearPageIndicatorDrawable. SEPARATION_ANGLE);
```

另外说明：

- 1.圆表上指示器是以控件中心为圆点，最短边距为直径，根据角度依次画出。
- 2.可以通过平移控件的方式调整指示器位置。

28.对话框(一个按钮): HeyDialog

效果图:



使用方法:

```
HeyDialog.HeyBuilder builder = new HeyDialog.HeyBuilder(this);
builder.setContentStyle(HeyDialog.STYLE_TITLE_CONTENT) //设置对话框内容样式, 含标题
    .setTitle("允许使用位置权限") //设置对话框标题
    .setMessage("轨迹等数据, 允许") //设置对话框内容
    .setPositiveButton("确定", null); //设置对话框按钮文本和点击事件
dialog = builder.create();
dialog.show();
```

使用说明:

HeyBtnDialog 继承自 Dialog, 具备 Dialog 的一切特性; 使用构建者模式创建, 与原生保持一致; setContentStyle 用于设置对话框内容样式, 控件包预置了 3 种样式, 本例的样式是对话框包含标题的样式。如果对话框按钮点击事件不需要额外的逻辑, 那么 setPositiveButton 方法的第二个参数传入 null, 点击后对话框会自动消失。

对话框默认不支持右滑退出, 可以通过 `setSwipeEnable` 方法, 传入 `true` 来控制支持对话框右滑退出, 所有的对话框均如此

29.对话框(两个按钮): HeyDialog

效果图:



使用方法:

```
HeyDialog.HeyBuilder builder = new HeyDialog.HeyBuilder(this);
builder .setContentViewStyle(HeyDialog.STYLE_TITLE_CONTENT)
        .setMessage("一行的说明文本中对齐")
        .setTitle("允许使用位置权限")
        .setPositiveButton("按钮", new View.OnClickListener() {
            @Override
            public void onClick(View v) {
            }
        })
        .setNegativeButton("按钮", null)
        .setButtonOrientation(LinearLayout.VERTICAL);
Dialog dialog = builder .create();
dialog.show();
```

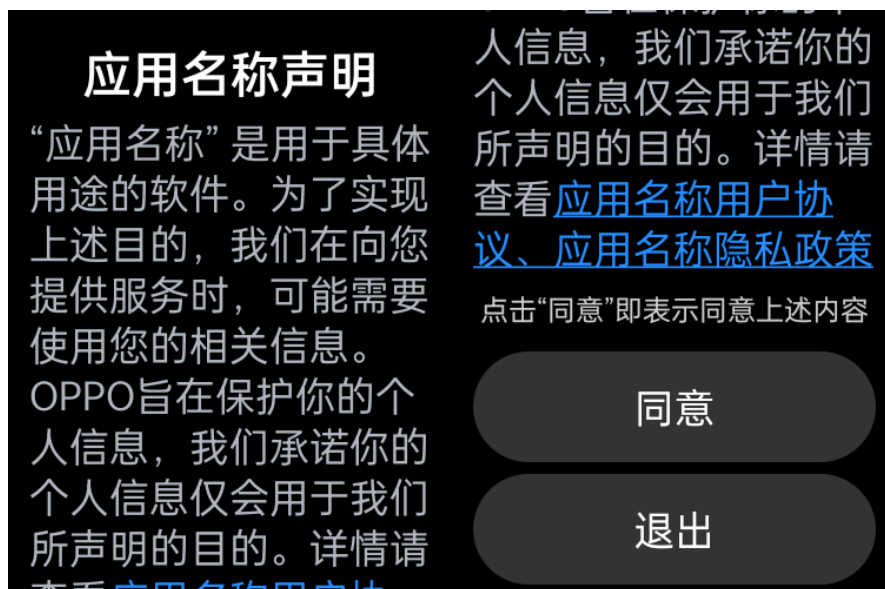
使用说明：

与一个按钮对话框相比，两个按钮对话框需要调用方法
setButtonOrientation 设置按钮方向，类型是 LinearLayout.HORIZONTAL 和
Vertical,默认是 Vertical

另外，对于水平布局的按钮来说，右边按钮是 PositiveButton；对于垂直布局的按钮来说，下面按钮是 PositiveButton

30.对话框(权限隐私申明)：HeyDialog

效果图：



使用方法：

```
HeyDialog.HeyBuilder builder = new HeyDialog.HeyBuilder(this);
builder.setContentStyle(HeyDialog.STYLE_PROTOCOL)
.setTitle("应用名称声明")
.setMessage("“应用名称”是用于具体用途的软件。为了实现上述目的，我们在向您提供服务时，可能需要使用您的相关信息。OPPO 旨在保护你的个人信息，我们承诺你的个人信息仅会用于我们所声明的目的。详情请查看应用名称用户协议、应用名称隐私政策。")
.setButtonOrientation(LinearLayout.HORIZONTAL)
.setSummary("点击“同意”即表示同意上述内容")
.setNegativeButton("按钮", null)
.setPositiveButton("按钮", new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        Toast.makeText(SecActivity.this, "隐私申明", Toast.LENGTH_LONG).show();
    }
});
```

```

    }
});
HeyDialog dialog = builder.create();

```

使用说明：同上

31.对话框(自定义布局): HeyDialog

效果图：



使用方法：

```

HeyDialog.HeyBuilder builder = new
HeyDialog.HeyBuilder(HeyDialogActivity.this);
builder.setHeyCustomLayout(R.layout.custom_layout) //自定义布局
.setPositiveButton("确定", null);
HeyDialog dialog = builder.create();
dialog.show();

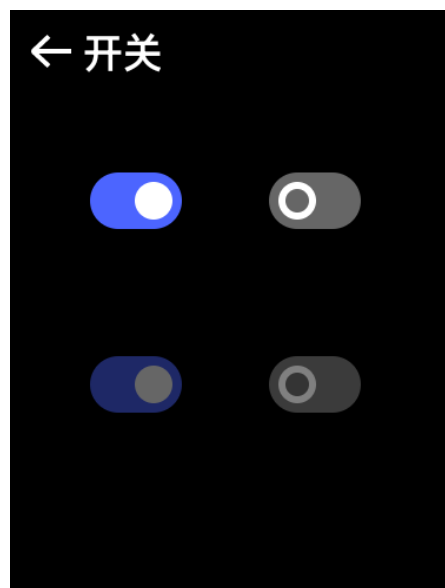
```

使用说明：

鉴于对话框样式千变万化，所以提供了自定义布局的对话框，使用方法如上所示，只需要传入自己的布局就行，**如果所需对话框不在上面列出的样式当中，请通过自定义布局实现，控件包不再单独提供别的对话框**

32.开关: HeySwitch

效果图:



使用方法:

```
<com.heytao.wearable.support.widget.HeySwitch
    android:id="@+id/switch1"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:clickable="true"
    style="@style/HeySwitchStyle"/>
```

使用说明:

此控件继承自 CompoundButton, 新增了部分自定义属性:

```
<declare-styleable name="HeySwitch">
    <!--开关高度-->
    <attr name="barHeight" format="dimension" />
    <!--开关宽度-->
    <attr name="barWidth" format="dimension" />
    <!--圆点半径-->
    <attr name="outerCircleWidth" format="dimension" />
    <!--关闭时, 中间灰圆与外面白圆之间的距离-->
    <attr name="outerCircleStrokeWidth" format="dimension" />
```

```

<!--关闭时，中间灰圆外面圆之间的颜色/打开时外圆的颜色-->
<attr name="outerCircleColor" format="dimension" />
<!--关闭时内圆半径-->
<attr name="innerCircleWidth" format="dimension" />
<!--内圆内边距-->
<attr name="circlePadding" format="dimension" />
<attr name="loadingDrawable" format="reference" />
<attr name="barCheckedColor" format="color" />
<attr name="barUncheckedColor" format="color" />
<attr name="innerCircleColor" format="color" />

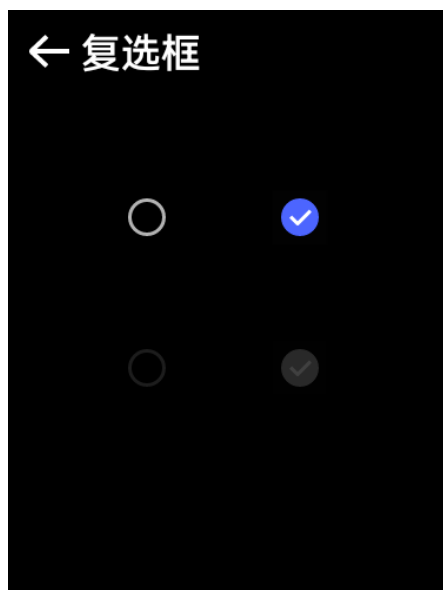
<!--内圆内边距-->
<attr name="barUncheckedDisabledColor" format="color" />
<attr name="barCheckedDisabledColor" format="color" />
<attr name="outerCircleUncheckedDisabledColor" format="color" />
</declare-styleable>

```

PS:V5.0.2 引入开启振动效果的接口 setVibrateEnable, false 代表点击不振动,
true 代表点击振动

33.复选框：HeyCheckBox

效果图：



使用方法：

```
<com.heytap.wearable.support.widget.HeyCheckBox  
    android:id="@+id/oppo_check_box"  
    android:layout_gravity="center_horizontal"  
    android:layout_width="wrap_content"  
    android:layout_height="wrap_content"  
    android:clickable="true"  
    android:focusable="true" />
```

使用说明：

HeyCheckBox,继承自 Button，新增了部分自定义属性：

```
<declare-styleable name="HeyCheckBox">  
    <!--CheckBox 的颜色，包括选中和不选中等-->  
    <attr name="HeyCheckBoxStateList" format="reference" />  
    <!--CheckBox 是否全选-->  
    <attr name="HeyCheckboxStateAllSelect" format="boolean" />  
    <!--CheckBox 是否部分选择-->  
    <attr name="HeyCheckboxStatePartSelect" format="boolean" />
```

```

<!--CheckBox 大小-->
<attr name="HeyCheckboxSize" format="dimension" />
<!--CheckBox 状态(全选，全不选，部分选中)-->
<attr name="HeyCheckBoxState">
    <enum name="unSelected" value="0" />
    <enum name="partSelected" value="1" />
    <enum name="allSelected" value="2" />
</attr>
</declare-styleable>

```

34.单选框：RadioButton

使用方法：

```

<com.heytao.wearable.support.widget.HeyCheckBox
    style="@style/HeyUIRadioStyle"
    android:id="@+id/rb"
    android:layout_width="match_parent"
    android:layout_height="wrap_content" />

```

35.消息提示：HeyToast

效果图：



使用方法：

```
HeyToast.showToast(getApplicationContext(),"Toast 文字, 居中示意超长折行。",  
Toast.LENGTH_SHORT);
```

使用说明：

HeyToast 继承自 Toast，在 showToast 里面 inflater 了自定义布局，此布局对 Toast 的 TextView 进行了部分属性设置；没有新增自定义属性。

请传入 ApplicationContext，否则可能会存在内存泄露问题

36.音量滑块：HeyControllerView

效果图：



使用方法：

```
<com.heytap.wearable.support.widget.HeyControllerView
```

```
    android:id="@+id/colorControllerView"
```

```
    android:layout_width="match_parent"
```

```
    android:layout_height="wrap_content"
```

```
    app:progressBarColor="@color/hey_colorAccent"/>
```

```
<com.heytap.wearable.support.widget.HeyControllerView
```

```
    android:id="@+id/colorControllerView2"
```

```
    android:layout_width="match_parent"
```

```
    android:layout_height="wrap_content"
```

```
    android:layout_marginTop="10dp"
```

```
    app:curProgress="1"
```

```
    app:maxProgress="3"
```

```
    app:progressBarType="segment"
```

```
    app:progressBarColor="@color/hey_colorAccent"/>
```

```
<com.heytap.wearable.support.widget.HeyControllerView
```

```
    android:id="@+id/colorControllerView3"
```

```
    android:layout_width="match_parent"
```

```
    android:layout_height="wrap_content"
```

```
    android:layout_marginTop="10dp"
```

```
    app:curProgress="2"
```

```
    app:maxProgress="4"
```

```
    app:progressBarType="segment" />
```

```
<com.heytap.wearable.support.widget.HeyControllerView
```

```
    android:id="@+id/colorControllerView4"
```

```
    android:layout_width="match_parent"
```

```

    android:layout_height="wrap_content"
    android:layout_marginTop="10dp"
    app:curProgress="2"
    app:maxProgress="5"
    app:progressBarType="segment" />

```

使用说明：

此控件主要用于设置模块的音量控制，继承自 FrameLayout，左右的"-"、"+"可以减少、增加音量，封装了 HeyHorizontalProgressBar，具体实现是左右两个 ImageView 加上中间的 HeyHorizontalProgressBar 构成，此控件自定义了属性：

```

<declare-styleable name="HeyHorizontalProgressBar">
    <!--此属性是指音量控制器中进度条的背景颜色，如上图进度条的背景颜色是灰色-->
    <attr name="backgroundBarColor" format="color|reference"/>
    <!--此属性是指音量控制器中进度条的前景颜色，如上图 2、3、4 中进度条的前景颜色是原谅色-->
    <attr name="progressBarColor" format="color|reference"/>
    <!--此属性当且仅当音量控制器的 progressBarType 为 segment 时才有效：代表音量控制器中有几个小刻度-->
    <attr name="maxProgress" format="integer|reference"/>
    <!--此属性当且仅当音量控制器的 progressBarType 为 segment 时才有效：代表音量控制器中的当前进度-->
    <attr name="curProgress" format="integer|reference"/>
    <!--音量控制器的类型，离散型和连续型，默认类型是连续型-->
    <attr name="progressBarType">
        <enum name="normal" value="0" />
        <enum name="segment" value="1" />
    </attr>
</declare-styleable>

```

6.0 版本：

若需要使用 6.0 版本控件，请调用 HeyControllerView_6_0 类
 标题 TextView 可以通过 getTitleText()方法获取。
 显示数值的 TextView 可以通过 getLevelText()方法获取。

37.加载对话框：HeyLoadingDialog

效果图：



使用方法：

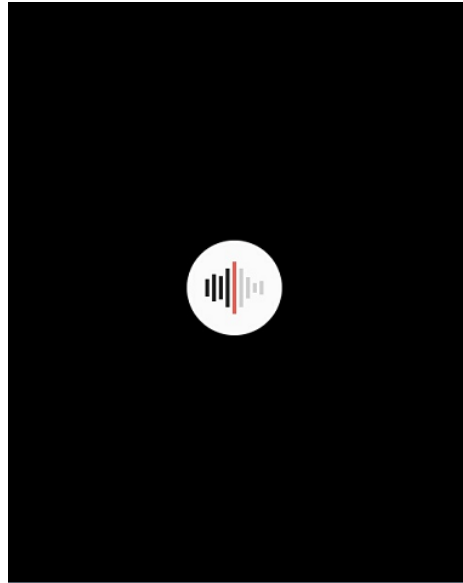
```
public void showLoadingDialog(View view) {  
    final HeyLoadingDialog colorLoadingDialog = new HeyLoadingDialog(this, "加载中...");  
    colorLoadingDialog.show();  
    new Handler().postDelayed(new Runnable() {  
        @Override  
        public void run() {  
            colorLoadingDialog.dismiss();  
        }  
    }, 15000);  
}
```

使用说明：

此控件继承自 Dialog，没有自定义属性

38.抗锯齿圆角图片：HeyRoundImageAntiAlias

效果图：



使用方法：

```
<com.heytap.wearable.support.widget.HeyRoundImageAntiAlias  
    android:layout_width="wrap_content"  
    android:layout_height="wrap_content"  
    android:gravity="center"  
    android:src="@drawable/record" />
```

使用说明：

此控件继承自 `ImageView`，具备 `ImageView` 的全部特性，没有新增自定义属性，直接当做普通 `ImageView` 使用即可。

39.分组小标题： HeyGroupTitle

效果图：



使用方法：

```
<com.heytap.wearable.support.widget.HeyGroupTitle
```

```
    android:id="@+id/title1"
```

```
    style="@style/HeyGroupTitleStyle"
```

```
    android:layout_width="match_parent"
```

```
    app:groupTitleText="分组小标题"
```

```
    app:groupTitleShowLoading="true"/>
```

```
<com.heytap.wearable.support.widget.HeyGroupTitle
```

```
    android:layout_width="match_parent"
```

```
    style="@style/HeyGroupTitleStyle"
```

```
    app:groupTitleText="分组小标题"/>
```

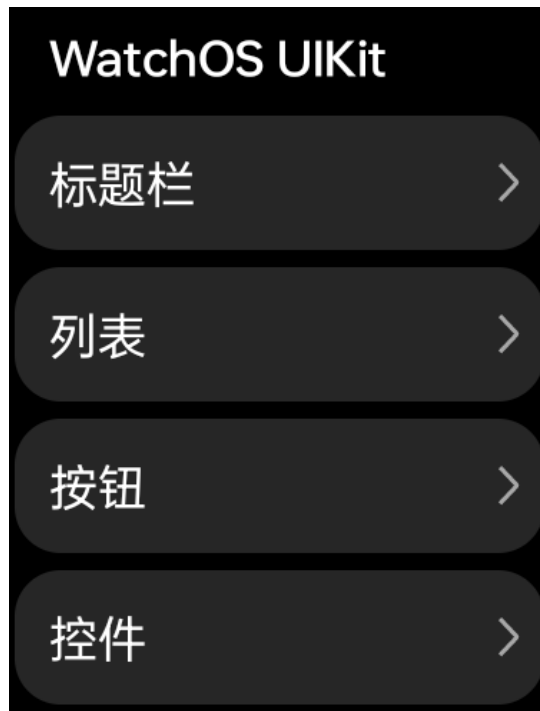
使用说明：

文本可以通过 `app:groupTitleText` 设置，文本位置可以通过 `app:groupTitlePosition` 设置

40.回弹列表 RecyclerView/ListView/ScrollView

1.RecyclerView

效果图：



使用方法：

```
<com.heytao.wearable.support.recyclerview.widget.RecyclerView
    android:id="@+id/recycler"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:fadeScrollbars="false"
    android:scrollbars="vertical" />
```

```
import com.heytao.wearable.support.recyclerview.widget.LinearLayoutManager;
import com.heytao.wearable.support.recyclerview.widget.RecyclerView;
mRecyclerView = findViewById(R.id.recycler);
mRecyclerView.setLayoutManager(new LinearLayoutManager(this));
```

使用说明：

此控件是在原生 RecyclerView 的基础上修改的，除了包名不一样，其他使用方法和原生 RecyclerView 一模一样

若需要顶部和底部的缩放效果以及底部对齐效果，则调用：

SetBottomAlignEnable (true)

若需要单独控制卡位效果和缩放效果,可调用 `setScaleAlignEnable(boolean scaleEnable, boolean alignEnable)`

是否需要响应表冠:

```
setSupportCrownFlag (boolean supportCrown)
```

supportCrown 为 true: 响应

supportCrown 为 false: 不响应

默认为 true

ListView/ScrollView:

ListView 和 ScrollView 均新增了是否支持回弹的接口 `setEnabledOverScroll(boolean b)`, false 表示不支持回弹, true 表示支持回弹, 默认为 false, 即默认不支持回弹, 如需支持, 可以通过反射调用 `setEnabledOverScroll` 设置

列表目前提供设置滚动条上边距和下边距的方法:

设置上边距: `setScrollbarPaddingTop`

设置下边距: `setScrollbarPaddingBottom`

注: 对于 gradle 编译的 APK, 设置上、下边距需要接入控件包里面提供的 `framework.jar`

41.脚注： HeyFootNoteView

效果图：



使用方法：

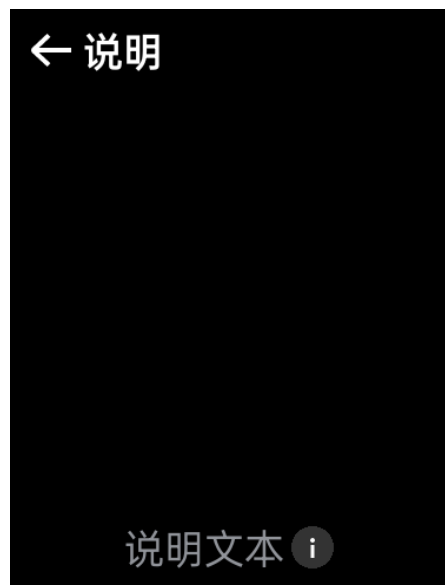
```
<com.heytap.wearable.support.widget.HeyFootNoteView  
    android:layout_width="match_parent"  
    android:layout_height="wrap_content"  
    app:heyFootIcon="@drawable/footnote"  
    app:heyFootText="@string/hey_foot_text"  
/>
```

使用说明：

此控件包含两个属性， heyFootIcon 和 heyFootText， 用于控制脚注左边的图标和右边的文字

42.说明： HeyInformationView

效果图：



使用方法：

```
<com.heytap.wearable.support.widget.HeyInformationView  
    android:layout_width="match_parent"  
    android:layout_height="wrap_content"  
    app:heyInfoIcon="@drawable/ic_info"  
    app:heyInfoText="@string/hey_info_text"  
/>
```

使用说明：

此控件包含两个属性，heyInfoIcon 和 heyInfoText，用于控制说明左边的图标和右边的文字

43.卡片：HeyBackgroundView

效果图：



使用方法：

```
<com.heytap.wearable.support.widget.HeyBackgroundView
    android:id="@+id/small"
    style="@style/HeyCardBackgroundStyle"
    android:layout_height="60dp" />

<com.heytap.wearable.support.widget.HeyBackgroundView
    android:id="@+id/normal"
    style="@style/HeyCardBackgroundStyle"
    android:layout_height="80dp"/>

<com.heytap.wearable.support.widget.HeyBackgroundView
    android:id="@+id/large"
    style="@style/HeyCardBackgroundStyle"
    android:layout_height="100dp"/>
```

使用说明：

此控件继承自 `FrameLayout`，在需要卡片的场景，应用应该使用它作为背景，高度根据卡片内容而自适应变化，不同高度的卡片，其圆角半径是不一样的

44.输入框：HeyEditTextStyle

效果图：



使用方法：

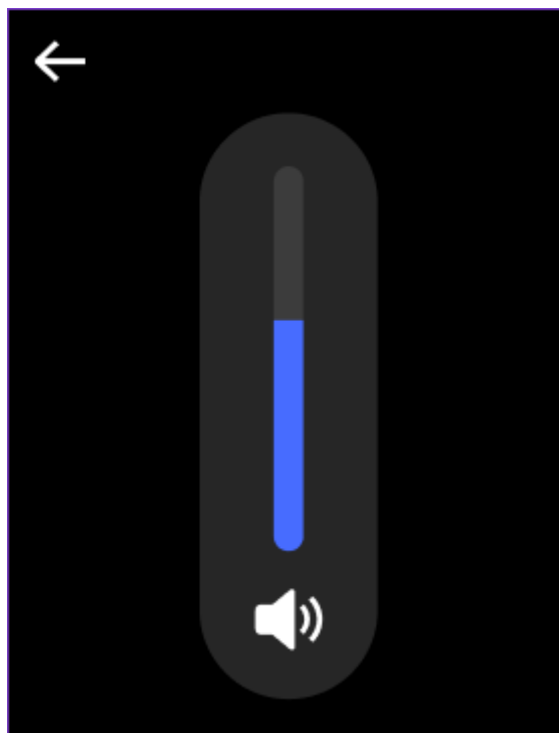
```
<EditText
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:text="REC"
    style="@style/HeyEditTextStyle"/>
```

使用说明：

此效果不是控件，而是一个 `style`，凡用到输入框的界面，引入此 `style` 即可实现如上效果

45.音量条： HeySeekBar

效果图：



使用方法：

```
<com.heytap.wearable.support.widget.HeySeekBar  
    android:id="@+id/seek"/>
```

使用说明：

V5.0 及以下：

此控件继承自源生的 SeekBar，除了监听事件由 OnSeekBarChangeListener 改成自定义的事件 HeyOnSeekBarChangeListener 以外，其他的功能与原生 SeekBar 并无差别。

SetCrownMax (int crownMax) 设置表冠转动时音量条的最大格数。

V6.0 及以上：与 5.0 的区别在于，HeySeekBar 继承原生的 FrameLayout，若想获取 SeekBar，需要调用 getSeekBar () 方法。V5.0 中 HeySeekBar 所提供的方法 都可以在 V6.0 中直接使用。若需要使用 6.0 版本控件，请调用 HeySeekBar_6_0 类。

46.ViewPager: HeyViewPager

说明：实现表冠转动时支持滑动的效，build.gradle 里需 implementation 'androidx.appcompat:appcompat:1.2.0' 或 androidx.appcompat:appcompat 1.2.0 之上的版本

47.动画差值器

效果图：

无

使用方法：

根据设计要求，新增四种动画差值器：

1.StandardCurve ：标准差值器

2.EaseCurve ：平缓差值器

3.AccelerationCurve ：加速差值器

4.DecelerationCurve ：减速差值器

上面四种动画差值器均基于贝塞尔曲线实现，在 HeyWidgetUtils 类中定义了四个常量

代表四种动画差值器：

HEY_STANDARD_INTERPOLATOR ：标准差值器

HEY_EASE_INTERPOLATOR ：平缓差值器

HEY_ACCELERATION_INTERPOLATOR ：加速差值器

HEY_DECELERATION_INTERPOLATOR ：减速差值器

通过 HeyWidgetUtils.getHeyInterpolator(HeyWidgetUtils.HEY_XXX_INTERPOLATOR)

即可获取所需要的动画差值器

48.右滑主题 1：HeytapTheme

效果图：

无

使用方法：

a.继承自此主题，在 styles.xml 中：

```
<style name="AppTheme" parent="@style/HeytapTheme">
    <!-- Customize your theme here. -->
    ....
</style>
```

b.直接使用此主题，在 AndroidManifest.xml 中：

```
<application
    .....
    android:theme="@style/HeytapTheme">
```

使用说明：

新增此主题是为了适配字体，所有用到右滑退出主题的应用，**务必**均接入此主题，界面直接继承自 Activity 的应用请使用此主题

49.右滑主题 2：HeytapCompatTheme

效果图：

无

使用方法：

c.继承自此主题在 styles.xml 中：

```
<style name="AppTheme" parent="@style/HeytapCompatTheme">
    <!-- Customize your theme here. -->
    ....
</style>
```

d.直接使用此主题，在 AndroidManifest.xml 中：

```
<application
    .....
    android:theme="@style/HeytapCompatTheme">
```

使用说明：

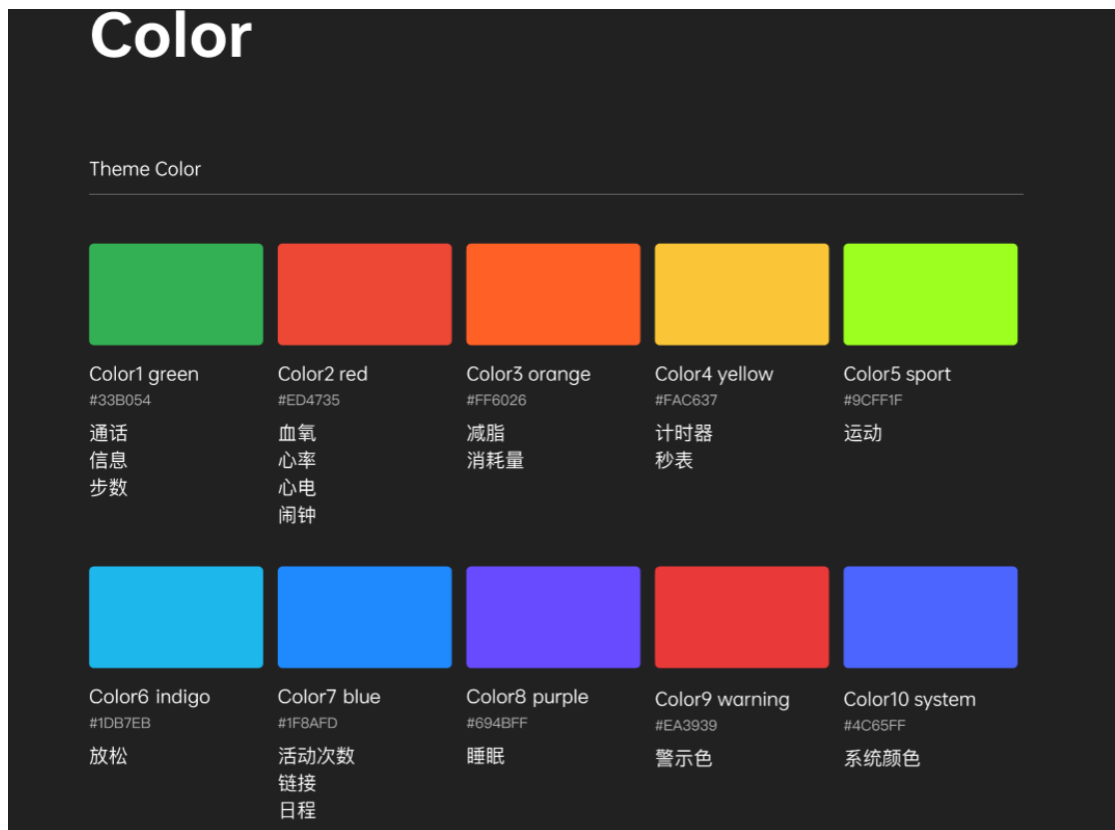
新增此主题是为了适配字体，所有用到右滑退出主题的应用，**务必**均接入此主题，界面继承自 AppCompatActivity 的应用请使用此主题

50.主题颜色

根据设计要求，控件包集成 10 种主题颜色，各个应用的主色调从这些颜色中选取，颜色如下：

<code><color name="hey_color_green">#33B054</color></code>
<code><color name="hey_color_red">#ED4735</color></code>
<code><color name="hey_color_orange">#FF6026</color></code>
<code><color name="hey_color_yellow">#FAC637</color></code>
<code><color name="hey_color_sport">#9CFF1F</color></code>
<code><color name="hey_color_indigo">#1DB7EB</color></code>
<code><color name="hey_color_blue">#1F8AFD</color></code>
<code><color name="hey_color_purple">#694BFF</color></code>
<code><color name="hey_color_waring">#EA3939</color></code>
<code><color name="hey_color_system">#4C65FF</color></code>

效果如下：



51.字体

为适配大小表字体效果，控件包新增以下字体大小，各应用务必引用下面的字体大小，否则小表显示效果会不对：

```
<dimen name="hey_xxxl_body">101sp</dimen>
<dimen name="hey_xxl_body">84sp</dimen>
<dimen name="hey_xl_body">69sp</dimen>
<dimen name="hey_l_body">57sp</dimen>
<dimen name="hey_m_body">46sp</dimen>
<dimen name="hey_s_body">38sp</dimen>
<dimen name="hey_xl_title">31sp</dimen>
<dimen name="hey_l_title">26sp</dimen>
<dimen name="hey_24M_Title">24sp</dimen>
<dimen name="hey_m_title">21sp</dimen>
<dimen name="hey_s_title">17sp</dimen>
<dimen name="hey_l_desription">17sp</dimen>
<dimen name="hey_m_desription">14sp</dimen>
```

```
<dimen name="hey_s_description">12sp</dimen>
<dimen name="hey_caption">9sp</dimen>
```

字体：

设计稿	代码
font-family: OPlusSans 3.0 font-weight: 400	android:fontFamily="oplus-sans-3"
font-family: OPlusSans 3.0 font-weight: 500	android:fontFamily="oplus-sans-3-medium"
font-family: OPlusSans 3.0 font-weight: 700	android:fontFamily="oplus-sans-3" android:textStyle="bold"
font-family:OPlusSans_Data font-weight: 400	android:fontFamily="oplus-sans-data"
font-family:OPlusSans_Data font-weight: 500	android:fontFamily="oplus-sans-data-medium"
font-family:OPlusSans_Data font-weight: 700	android:fontFamily="oplus-sans-data" android:textStyle="bold"

52.常用动画

为方便应用开发，控件包特意预置部分公用动画：

1.dialog_enter_from_bottom，此动画是从底部弹出动画，常用于对话框

```
<set xmlns:android="http://schemas.android.com/apk/res/android"
    android:zAdjustment="top">
    <translate
        android:fromYDelta="100%p"
        android:toYDelta="0%p"
        android:duration="133"/>
</set>
```

2.dialog_exit_from_top，此动画是从顶部退出动画，常用于对话框

```
<set xmlns:android="http://schemas.android.com/apk/res/android"
    android:zAdjustment="top">
    <translate
        android:fromYDelta="0%p"
        android:toYDelta="100%p"
        android:duration="133"/>
</set>
```

3.开放列表 item 点击动画给应用调用：

```
点击动画：HeyAnimationUtils.itemAnimateToPress(View view);
释放动画：HeyAnimationUtils.itemAnimateToNormal(View view);
```

53.大小表适配

鉴于当前手表屏幕尺寸和屏幕密度各不相同,以后可能还有更多不同参数的屏幕,所以屏幕适配刻不容缓,为统一效果,提升适配效率,故制定统一的适配方案。

目前存在 4 种屏幕:一代小表,一、二代大表,三代小表,三代大表四种表的屏幕参数各不相同。

针对这四种不同屏幕,采用最小宽度限定符方案, res 目录下新增如下四个文件夹目录:



不同尺寸和屏幕密度的尺寸资源文件放置于不同的 sw 文件夹下, sw 的含义是在屏幕的宽和高之间取数值最小的那个,一般都是宽比高小,所以 sw 很多时候取的是宽度,对于我们的手表而已,其计算方式如下:

$$sw = width * DisplayMetrics.DENSITY_DEFAULT / densityDPI$$

Width: 屏幕的宽度

DisplayMetrics.DENSITY_DEFAULT: google 的标准值, 一般是 160

densityDPI: 屏幕密度 (每寸像素数)

手表型号	分辨率	屏幕密度	values资源区分条件	values资源获取优先级	drawable资源区分条件	drawable资源获取优先级	规则
一代大表二代大表	402x476	320	values-sw200dp	values-sw200dp > values	drawable-sw200dp-xhdpi	drawable-sw200dp-xhdpi > drawable-xhdpi	1、默认资源目录values和drawable-xhdpi具有全量的资源 2、其他目录只定义差异化资源, 需注意资源获取优先级
一代小表	320x360	240	values-sw213dp	values-sw213dp > values-sw200dp > values	drawable-hdpi	drawable-hdpi > drawable-sw200dp-xhdpi > drawable-xhdpi	
二代小表 奥创SE 奥创	372x430	320	values (默认)	values	drawable-xhdpi (默认)	drawable-xhdpi	
奥斯特	378x496	320	values-h248dp	values-h248dp > values	drawable-h248dp-xhdpi	drawable-h248dp-xhdpi > drawable-xhdpi	

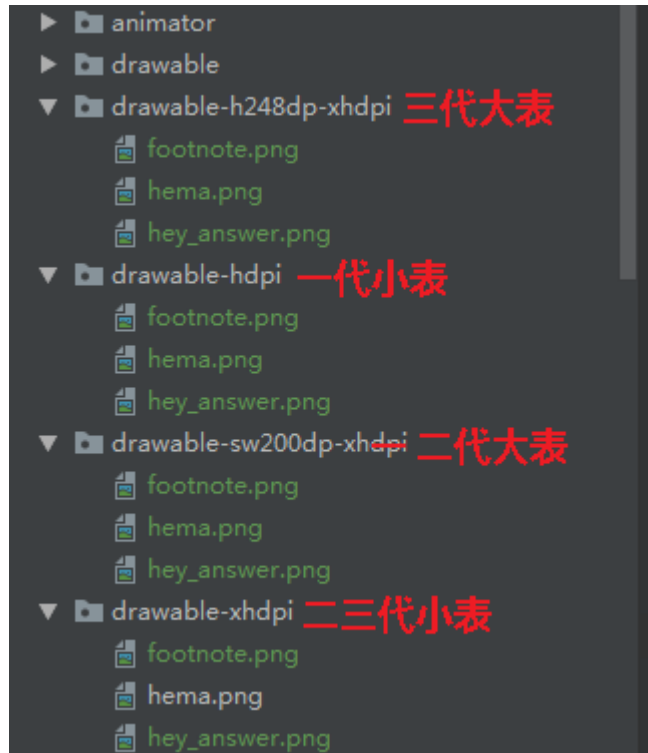
一二代大表的尺寸资源放置于 sw200dp 下, 一代小表放置于 213dp 下, 二代小表放置于 values 下; 三代大表放置于 values-h248dp 下, values 下面的 dimen 资源不可或缺; **与二代表适配方案不同, 三代表开始我们是以小表为基准, 同时舍弃二代大小表之间的缩放方案;** 上面是关于尺寸资源的适配, 下面说明 svg 图片的资源适配。对于 svg 图片, 其大小均是在 xml 中通过硬编码写死的:

```
<vector
  xmlns:android="http://schemas.android.com/apk/res/android"
  android:name="vector"
  android:width="24dp"
  android:height="24dp"
  android:viewportWidth="12"
  android:viewportHeight="12">
  <path
    android:name="path"
    android:pathData="M 27.4 12 L 44.6 12 C 49.9 12 51.9 12.6 53.8 13.6 C 55.8 14.6 57.3 16.2 58.3 18.1 C 59.3 20.1 59.9 22 59.9 27.3 L 59.9 44.5 C 59.9 49.8 59.3 51.8 5
    android:fillColor="#2AD181"
    android:strokeWidth="1"
    android:fillType="evenOdd"/>
  <path
    android:name="path_1"
    android:pathData="M 45.9 27.6 C 46.7 26.8 47.9 26.8 48.7 27.6 C 49.5 28.4 49.5 29.6 48.7 30.4 L 34.1 45.2 L 33.9 45.4 C 32.7 46.5 30.8 46.4 29.7 45.1 L 23.5 38 C 22.
    android:fillColor="#ffffff"
    android:strokeWidth="1"/>
</vector>
```

针对这种情况, 需要修改 xml 文件中的宽高, 通过@dimen/指向不同 sw 文件夹下面的尺寸资源; 注意: 这种方法要求 **minSdkVersion 为 24**, 适配后如下所示:

```
<vector
  xmlns:android="http://schemas.android.com/apk/res/android"
  android:name="vector"
  android:width="@dimen/hey_listitem_widget_size"
  android:height="@dimen/hey_listitem_widget_size"
  android:viewportWidth="12"
  android:viewportHeight="12">
  <path
    android:name="path"
    android:pathData="M 27.4 12 L 44.6 12 C 49.9 12 51.9 12.6 53.8 13.6 C 55.8 14.6 57.3 16.2 58.3 18.1 C 59.3 20.1 59.9 22 59.9 27.3 L 59.9 44.5 C 59.9 49.8 59.3 51.8 5
    android:fillColor="#737373"
    android:strokeWidth="1"
    android:fillType="evenOdd"/>
  <path
    android:name="path_1"
    android:pathData="M 45.9 27.6 C 46.7 26.8 47.9 26.8 48.7 27.6 C 49.5 28.4 49.5 29.6 48.7 30.4 L 34.1 45.2 L 33.9 45.4 C 32.7 46.5 30.8 46.4 29.7 45.1 L 23.5 38 C 22.
    android:fillColor="#666666"
    android:strokeWidth="1"/>
</vector>
```

还有一种需要适配的资源是 PNG 图片, 其适配方法和尺寸资源 dimen 一样:



不同尺寸和屏幕密度的 PNG 资源文件放置于不同的 sw 文件夹下，一二代大表的尺寸资源放置于 sw200dp-xhdpi 下，一代小表放置于 hdpi 下，二三代小表放置于 xhdpi 下，三代大表放置于 h248dp-xhdpi 下

适配注意事项：

1.任何能用控件包里面预置的尺寸， 必须用控件里面的， 减少应用适配工作量， 保持视觉效果的一致性。

2.对于控件， 能用控件包里面的 style 的， 必须用 style， 不要用的自己定义宽高和其他属性， 避免增加应用适配工作量； 不了解各个 style 的， 请看各个控件的使用方法， 直接拷贝粘贴即可(文本除外)

3.资源选取规则如下：

values 下面必须有默认的资源

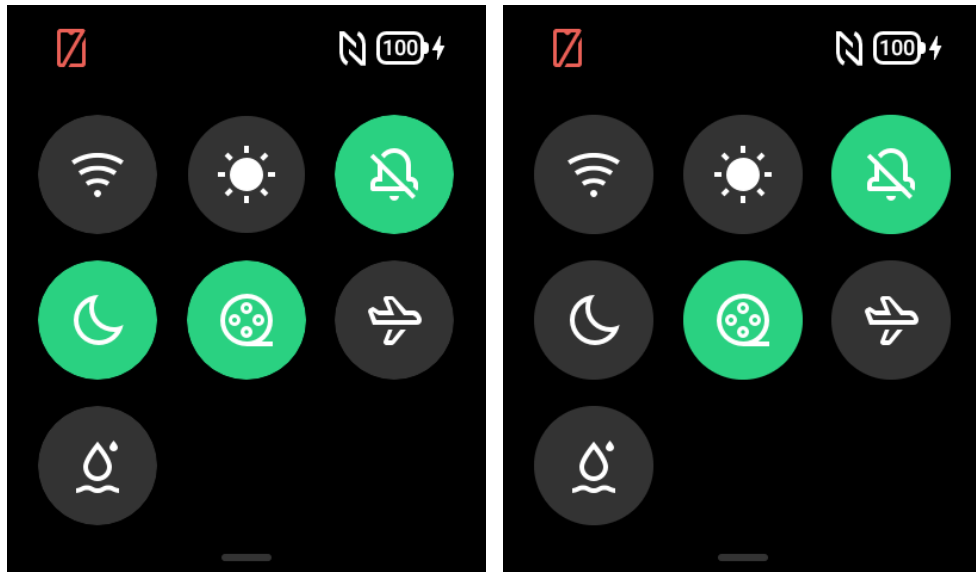
选取规则以一代小表为例， 先去 values-sw213 查找资源； 如果没有找到， 就去 values-sw200 查找； 如果还没找到， 就去 values 查找； 总体规则就是从大到小查找， 其他表也是这样的规则。

54.抗锯齿优化

鉴于当前手表大量使用圆角和圆形图片, 这些图片在大小表上表现均有不同程度的锯齿问题, 对用户体验影响较为明显, 为解决此问题, 特整理此方案, 便于各应用在以后的开发中避免此问题。

```
myView.setOutlineProvider(new ViewOutlineProvider() {  
    @Override  
    public void getOutline(View view, Outline outline) {  
        outline.setOval(0, 0, getWidth(), getHeight());  
    }  
});  
myView.setClipToOutline(true);
```

此方法是 View.java 中定义的, 所以任何继承自 view 类的控件都可以使用; 另外, 如果只需要圆角效果, 可以调用 setRoundRect 方法。如果自定义控件没有继承自 view 或者控件内没包含 view, 那么此方案将无效, 效果对比:



55.控件 6.0 版本性能优化

6.0 控件在点击是增加了卡片动画，导致启动速度变慢，优化方案是当点击卡片需要响应页面跳转时，接收到 tp 的 up 事件，则立刻停止卡片点击动画播放。

基类：

HeySingleBaseItem.java

HeyMultipleBaseItem.java

设置方法：setWindowSwitchFlag(boolean needWindowSwitch)

设为 true：接收到 tp 的 up 事件，立刻停止卡片点击动画播放

设为 false：接收到 tp 的 up 事件，不停止卡片点击动画播放

默认值为 false

56. 5.0 兼容性判断

通过 feature: opus.wear.ui.version.6.0 来判断是不是 6.0 版本

57.圆表相关控件

1. 默认文字按钮

效果图：



使用方法：

```
<com.heytap.wearable.support.widget.HeyShapeButton  
    style="@style/HeyRoundButtonTextDefaultStyle"  
>  
</>
```

2. 默认图标按钮

效果图：

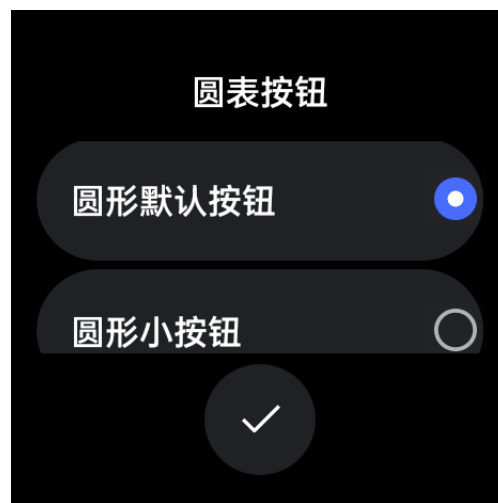


使用方法：

```
<com.heytap.wearable.support.widget.HeyImageButton  
    style="@style/HeyRoundButtonIconDefaultStyle"  
>
```

3. 圆形默认按钮

效果图：

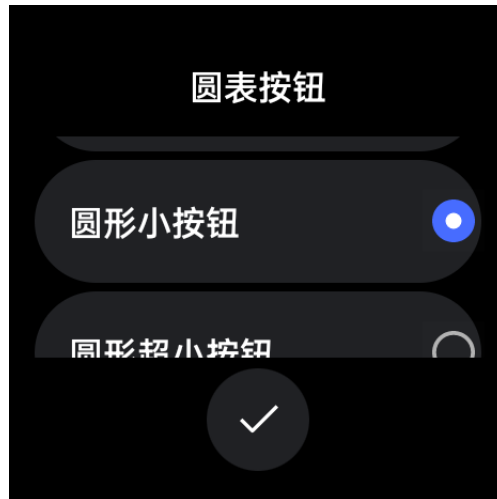


使用方法：

```
<com.heytap.wearable.support.widget.HeyImageButton  
    style="@style/HeyRoundButtonCircleNormalStyle"  
>
```

4. 圆形小按钮

效果图：



使用方法：

```
<com.heytap.wearable.support.widget.HeyImageButton  
    style="@style/HeyRoundButtonCircleSmallStyle"  
>
```

5. 圆形超小按钮

效果图：



使用方法：

```
<com.heytap.wearable.support.widget.HeyImageButton  
    style="@style/HeyRoundButtonCircleSmallestStyle"  
>
```

6. 圆形大按钮

效果图：

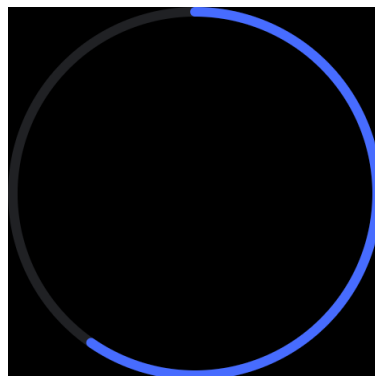


使用方法：

```
<com.heytap.wearable.support.widget.HeyImageButton  
    style="@style/HeyRoundButtonCircleLargeStyle"  
>
```

7. HeyProcessSpinnerIndicator 进度提示控件

效果图 1：

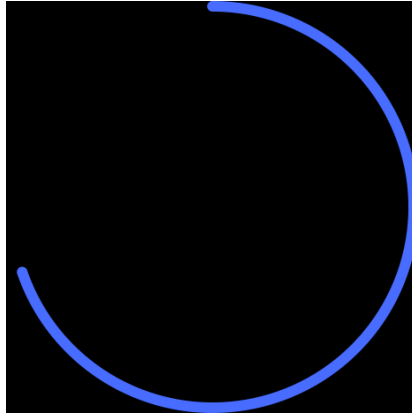


使用方法：

```
<com.heytap.wearable.support.widget.progressindicator.HeyProgressSpinnerIndicator  
    android:layout_width="match_parent"
```

```
android:layout_height="match_parent"
app:progress="0.0"
app:spinnerDirection="clockwise"
style="@style/HeyProgressSpinnerIndicatorDefault"
android:gravity="top" />
```

效果图 2:

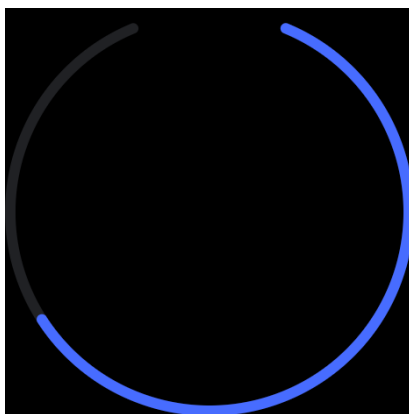


使用方法:

```
<com.heytao.wearable.support.widget.progressindicator.HeyProgressSpinnerIndicator
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    app:progress="0.0"
    app:spinnerDirection="clockwise"
    style="@style/HeyProgressSpinnerIndicatorDefault"
    android:gravity="top"
    app:spinner="@drawable/track_none_progress_indicator"/>
```

其中 `app:spinnerDirection="clockwise"` 为进度前进方向，`clockwise` 顺时针
`counter_clockwise` 逆时针（倒计时）

效果图 3:



使用方法：

```
<com.heytap.wearable.support.widget.progressindicator.HeyProgressSpinnerIndicator  
    android:layout_width="match_parent"  
    android:layout_height="match_parent"  
    app:progress="0.0"  
    app:spinnerDirection="clockwise"  
    style="@style/HeyProgressSpinnerIndicatorDefault"  
    android:gravity="top"  
    app:spinner="@drawable/track_progress_partial_indicator"/>
```

app:progress="0.0" 为当前进度条值，取值在 0.0-1.0 之间的任意 float 型值

您可以定制 app:spinner 如下：

```
<?xml version="1.0" encoding="utf-8"?>  
<drawable xmlns:app="http://schemas.android.com/apk/res-auto"  
    class="com.heytap.wearable.support.widget.progressindicator.ProgressSpinnerDrawable"  
    app:spinnerProgressColor="@color/hey_progress_spinner_indicator_spinner_progress_color"  
    app:spinnerTrackColor="@color/hey_progress_spinner_indicator_spinner_track_color"  
    app:spinnerTrackWidth="@dimen/hey_progress_indicator_spinner_track_width"  
    app:spinnerTrackStartAngle="22.5"  
    app:spinnerTrackEndAngle="337.5"/>
```

其中 app:spinnerProgressColor 为前景进度条颜色

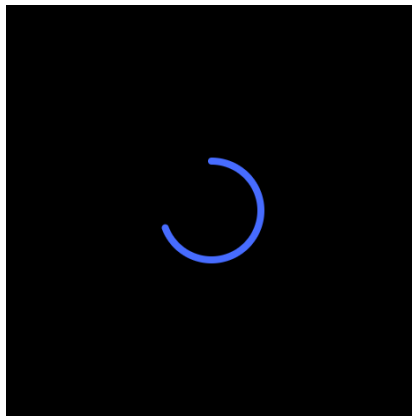
app:spinnerTrackColor 为背景轨道颜色

app:spinnerTrackWidth 为背景轨道宽度

app:spinnerTrackStartAngle 为开始的背景轨道角度

app:spinnerTrackEndAngle 为结束的背景轨道角度

效果图 4：



使用方法：

```
<com.heytap.wearable.support.widget.progressindicator.HeyProgressSpinnerIndicator  
    android:layout_width="60dp"  
    android:layout_height="60dp"  
    app:progress="0.0"  
    app:spinnerDirection="clockwise"  
    style="@style/HeyProgressSpinnerIndicatorDefault"  
    android:gravity="top"  
    app:spinner="@drawable/track_none_progress_small_indicator"  
    android:layout_centerInParent="true"/>
```

6.圆表时间控件



```
<com.heytap.wearable.support.widget.HeyArcTextClock  
    android:layout_width="wrap_content"  
    android:layout_height="wrap_content"  
    style="@style/HeyClockStyle" />
```

7.圆表 Dialog 左右按钮:



```
HeyDialog.HeyBuilder builder = new HeyDialog.HeyBuilder(this);
builder.setContentVisualStyle(HeyDialog.STYLE_TITLE_CONTENT)
    .setMessage("一行的说明文本中对齐")
    .setTitle("允许使用位置权限")
    .setImagePositiveButton(null, v -> { })
    .setButtonOrientation(LinearLayout.HORIZONTAL)
    .setIsRound(HeyWidgetUtils.isScreenRound(this));
dialog9 = builder.create();
dialog9.show();
```

使用说明:

```
.setImagePositiveButton(null, v -> { })
设置右边按钮图片和事件，图片为空则是默认对号
.setImageNegativeButton(null, v -> { })
设置左边按钮图片和事件，图片为空则是默认 x 号
.setPositiveButtonColor(getColor(R.color.default_background_color))
设置右边按钮背景色，默认颜色 default_background_color
.setNegativeButtonColor(getColor(R.color.hey_colorAccent))
设置左边按钮背景色，默认颜色 hey_colorAccent
.setIsShowTwoRoundBtn(true);
设置是否是双按钮，默认为 true
.setIsRound(true)
设置是否是圆形按钮，默认为 false
```

10. 圆表列表样式变更

圆表上右箭头统一设置为不可见 GONE，如有需要请手动设为可见。
涉及控件为：HeySingleDefaultItem，HeyMultipleDefaultItem，
HeySingleItemLeftImage，HeyMultipleDefaultSummaryAlone