



HiGV

FAQ

文档版本	00B03
发布日期	2018-12-29

版权所有 © 上海海思技术有限公司 2019。保留一切权利。

非经本公司书面许可，任何单位和个人不得擅自摘抄、复制本文档内容的部分或全部，并不得以任何形式传播。

商标声明



HISILICON、海思和其他海思商标均为海思技术有限公司的商标。

本文档提及的其他所有商标或注册商标，由各自的所有人拥有。

注意

您购买的产品、服务或特性等应受海思公司商业合同和条款的约束，本文档中描述的全部或部分产品、服务或特性可能不在您的购买或使用范围之内。除非合同另有约定，海思公司对本文档内容不做任何明示或默示的声明或保证。

由于产品版本升级或其他原因，本文档内容会不定期进行更新。除非另有约定，本文档仅作为使用指导，本文档中的所有陈述、信息和建议不构成任何明示或暗示的担保。

上海海思技术有限公司

地址： 深圳市龙岗区坂田华为总部办公楼 邮编：518129

网址： <http://www.hisilicon.com/cn/>

客户服务邮箱： support@hisilicon.com

前言

概述

本文为使用 HiGV 开发的程序员而写，目的是为您在开发过程中遇到的问题提供解决办法和帮助。



说明

本文以 Hi3559AV100 描述为例，未有特殊说明，其他芯片和 Hi3559AV100 一致。

产品版本

与本文档相对应的产品版本如下。

产品名称	产品版本
Hi3559A	V100
Hi3559C	V100
Hi3556A	V100
Hi3559	V200
Hi3556	V200
Hi3519A	V100
Hi3518E	V300
Hi3516D	V300
Hi3516C	V500

读者对象

本文档（本指南）主要适用于以下工程师：

- 技术支持工程师
- 软件开发工程师

修订记录

修订记录累积了每次文档更新的说明。最新版本的文档包含以前所有文档版本的更新内容。

文档版本 00B03 (2018-12-29)

新增 Hi3556AV100/Hi3519AV100/Hi3559V200/Hi3556V200 的相关内容

文档版本 00B02 (2018-02-26)

第 2 次临时版本发布

新增 1.15 及 1.16 小节

文档版本 00B01 (2017-08-26)

第 1 次临时版本发布



目 录

前 言.....	i
1 FAQ.....	1
1.1 HiGV 常见问题	1
1.1.1 如何设置控件皮肤透明度	1
1.1.2 如何设置点阵字库和矢量字库	1
1.1.3 如何动态设置窗口所在图层的像素格式.....	2
1.1.4 如何批量修改布局文件	3
1.1.5 如何批量缩放图片	3
1.1.6 如何解决字体变粗的问题	3
1.1.7 如何解决控件皮肤半透明不显示问题.....	4
1.1.8 如何解决竖屏显示 GUI 问题	4
1.1.9 如何解决消息队列满的问题	5
1.1.10 如何解决回放界面显示缩略图效果差的问题.....	5
1.1.11 如何解决动画框架移动不流畅的问题.....	5
1.1.12 如何解决接口并发调用异常问题.....	6
1.1.13 如何解决 Listbox 控件滑动时界面叠加的问题.....	6
1.1.14 如何解决控件快速移动导致的裂屏问题.....	7
1.1.15 如何解决适配触摸屏时触摸手势事件回调无效问题.....	7
1.1.16 如何解决文字内容不能正常显示的问题.....	8
1.2 HiGV 典型场景示例.....	9
1.2.1 小型触摸屏提升 Button 控件体验的方法.....	9
1.2.2 触摸屏提升轻扫动作灵敏度的方法.....	10
1.2.3 触摸屏设置界面的使用控件选择.....	11
1.2.4 运动 DV 产品典型菜单界面设计方法.....	12
1.2.5 Scrollbar 控件皮肤的设置方法	12
1.2.6 Scrollbar 控件悬浮的设置方法.....	14
1.2.7 触摸屏时间日期功能开发	16
1.2.8 非触屏场景回放界面翻页方法	17
1.2.9 多国语言设置方法	17
1.2.10 触摸屏相关消息注册回调函数的参数解析.....	18

插图目录

图 1-1 典型主界面	9
图 1-2 典型设置界面	11
图 1-3 典型菜单界面	12
图 1-4 水平方向皮肤示例.....	13
图 1-5 垂直方向皮肤示例.....	14
图 1-6 典型日期控件	17

1 FAQ

1.1 HiGV 常见问题

1.1.1 如何设置控件皮肤透明度

【现象】

某些 GUI 场景需要设置半透明皮肤背景。

【分析】

通过半透明图片或半透明的 color 属性设置皮肤。

【解决】

皮肤 skin 在皮肤描述文件 skin.xml 文件中设置，可以支持两种皮肤类型：pic 和 color。

半透明皮肤设置的方法如下：

- 设置半透明图片，通过图片编辑软件对背景图片进行处理，更改其透明度属性为半透明，把更改好的图片在 skin.xml 中进行引用即可。
- 设置半透明颜色，颜色属性信息格式如“0xFF000000”，表示 ARGB，最高位的两个字节“FF”表示 Alpha 透明度，从“00”全透明到“FF”全不透明，可以设置两个数据之间的一个中间值表示半透明状态。

1.1.2 如何设置点阵字库和矢量字库

【现象】

字体在 GUI 界面使用是非常频繁的，HiGV 可以支持点阵字库和矢量字库的设置。

【分析】

点阵字库和矢量字库相比占用 Flash 空间较小，但是不能缩放，字体边缘效果没有矢量字库圆润，为了更好的显示效果，推荐使用矢量字库。

【解决】



- 在字体描述文件 font.xml 文件中添加字库，如下所示，分别添加一个点阵字库和矢量字库。

点阵字库设置：

```
<font
    id="Font10"
    sbname="./res/font/ubf/Font10.ubf"
    mbname=""
    size=""
    isbold=""
    isitalic=""/>
```

矢量字库设置：

```
<font
    id="Font40"
    sbname="./res/font/ttf/basic.ttf"
    mbname=""
    size="40"
    isbold=""
    isitalic=""/>
```

- 点阵字库和矢量字库设置有一个明显的区别是矢量字库需要显式的设置 size 的数值，而点阵字库的大小是固定的，无需设置。
- 点阵字库的后缀名称是 ubf，而矢量字库的后缀名称是 ttf。
- 可以使用 HiFontTool 工具，把矢量字库转化成点阵字库。
- 当前 font.xml 中最多支持设置 8 个字体，如果超出范围则不会生效。

1.1.3 如何动态设置窗口所在图层的像素格式

【现象】

介绍如何在窗口进行切换时，动态的设置窗口对应图层的像素格式，例如由 HIGO_PF_4444 切换为 HIGO_PF_1555。典型的使用场景是解决应用程序需要同时支持图层旋转，半透明显示和缩略图显示较好效果的需求。

【分析】

HiGO 提供了接口 HI_GO_ChangeLayerFmt，利用此接口可以在窗口切换时，设置切换后窗口的像素格式。

【解决】

以 A 窗口切换到 B 窗口，像素格式从 HIGO_PF_4444 切换为 HIGO_PF_1555 为例，设置过程如下：

步骤 1 设置窗口格式

在 A 窗口的布局文件 xml 的窗口属性中增加“pixelformat”属性，设置为“argb4444”，类似在 B 窗口的布局文件 xml 中增加“pixelformat=argb1555”的属性。

步骤 2 注册回调函数



在切换 B 窗口的 xml 注册 onshow 回调函数，像素格式切换的接口在 onshow 回调函数中设置。

步骤 3 设置像素格式

在 onshow 回调函数中，首先调用接口 `HI_GV_Layer_GetHigoLayer` 获取 HiGO 图层句柄，然后再通过接口 `HI_GO_ChangeLayerFmt` 设置 B 窗口所在图层的像素格式。如果图层需要旋转，还需再调用一次 `HI_GV_Layer_SetRotateMode` 接口设置图层旋转。

----结束

1.1.4 如何批量修改布局文件

【现象】

介绍如何对布局 xml 文件的内容进行等比例的缩放。

【分析】

发布包中有 `xml2bin` 工具，可使用 `xml2bin` 工具在 Linux 服务器中对 xml 布局文件进行批量的修改。

【解决】

批量修改命令示例如下：

```
# ~/.xml2bin-xxml/-C0320024004800320-bbin/higv.bin-tc-esrc/extcfile/ -l src/extcfile/
```

- 此命令表示把 xml 目录下的布局文件的画布分辨率由 320*240 修改为 480*320。
- 命令参数含义可通过命令 `./xml2bin -H` 获取。

1.1.5 如何批量缩放图片

【现象】

介绍如何对数量较多的图片进行批量缩放。

【分析】

可通过开源软件 `FFmpeg` 的命令对图片进行缩放。

【解决】

- `FFmpeg` 命令需要到官网 (<http://ffmpeg.org/>) 下载源码，在 Linux 上编译并安装。如果把一张图片的宽高均缩放为原来的一半，可以用下面命令：

```
#ffmpeg -i SING0001.JPG -vf scale=iw*0.5:ih*0.5 output.JPG
```

- 批量缩放命令如下：

```
#find . -name "*.png" | xargs -I {} ffmpeg -y -i {} -vf scale=iw*0.75:ih*0.75 {}
```

其中 `iw` 表示原始图片宽度，`ih` 表示原始图片高度，缩放比例是 0.75。

1.1.6 如何解决字体变粗的问题

【现象】



字体在显示的过程中不断变粗或字体的粗度与预期不合。

【分析】

出现这个问题一般是字体重复绘制导致。

【解决】

- 排查布局 xml 文件，查看关于字体控件是否有叠加的问题，如果有的话，把叠加区域分开。
- 排查代码文件，查看是否有频繁重复调用字体绘制的接口，禁止重复绘制字体。

1.1.7 如何解决控件皮肤半透明不显示问题

【现象】

已经设置好的半透明控件皮肤，在屏幕上显示时要么不透明，要么全透明，不显示半透明皮肤状态。

【分析】

皮肤透明度会受到图层的像素格式影响，某些像素格式下不支持半透明显示。

【解决】

- 当前 Hi3559AV100 芯片支持的像素格式包含 HIGO_PF_4444、HIGO_PF_1555 和 HIGO_PF_8888，其中 HIGO_PF_4444 和 HIGO_PF_8888 两种像素格式可以支持设置半透明皮肤，HIGO_PF_1555 像素格式不支持直接设置为半透明皮肤。
- 如果设置了支持半透明皮肤显示的像素格式后，依然不能显示半透明效果，并且皮肤类型为 color 类型，可以尝试修改为 pic 类型皮肤，用半透明的图片当做背景皮肤。

1.1.8 如何解决竖屏显示 GUI 问题

【现象】

竖屏显示 GUI，串口报错，GUI 显示不出来。

【分析】

屏幕按照显示方向分为横屏和竖屏，运动 DV 或行车记录仪等产品的 GUI 布局一般是按照横屏来设置，更符合人的观看习惯。

【解决】

步骤 1 图层初始化。

图层初始化参数设置，画布的宽度和高度进行对调，如下所示：

```
HIGO_LAYER_INFO_S LayerInfo = {SCREEN_WIDTH, SCREEN_HEIGHT,  
SCREEN_HEIGHT, SCREEN_WIDTH, SCREEN_WIDTH, SCREEN_HEIGHT, .....};
```

图层这三组宽高信息分别为屏幕的宽度和高度，画布的宽度和高度，显示的宽度和高度，对其中第二组参数，即画布参数进行对调，使画布横向布局。当前 Hi3559AV100 芯片，支持旋转的像素格式为 HIGO_PF_4444、HIGO_PF_1555 和 HIGO_PF_8888 格式。



步骤 2 调用旋转接口。

在初始化图层后，通过调用旋转接口 `HI_GV_Layer_SetRotateMode`，把图层旋转 90 度或 270 度，接口设置请参考发布包目录文件“`middleware/sample/higv/scene_sample/main_sample.c`”。

----结束

1.1.9 如何解决消息队列满的问题

【现象】

GUI 显示卡顿，不连续，串口提示“`Send async msg failed, queue full.`”。

【分析】

HiGV 内部绘制的处理通过消息机制分发，有一个消息队列，当消息队列被阻塞后，绘制消息就不会分发到控件了，导致 UI 不会进行绘制，引起卡顿，不连续的现象。

【解决】

- 消息的分发是在 UI 主线程中进行的，如果主线程被阻塞，就会导致消息队列被阻塞的现象。
- 出现这种问题的原因一般是用户在 xml 控件中注册的回调函数中有比较耗时的操作导致，由于回调函数位于主线程，当其有耗时操作时，会阻塞整个主线程，导致 UI 的卡顿，不连续。解决此类问题比较好的办法是把比较耗时的操作放到子线程中进行，不要影响到主线程的正常运行。

1.1.10 如何解决回放界面显示缩略图效果差的问题

【现象】

使用 `ScrollGrid` 控件显示九宫格或六宫格缩略图时，缩略图的显示效果和原图比较效果差，色彩渐变少。

【分析】

图层的像素格式 `RGB` 的位数对色彩渐变影响较大，位数越低的像素格式，显示效果越差，可通过使用位数较多的像素格式，来达到一个较好的体验效果。

【解决】

图层初始化时，如果像素格式设置为 `HIGO_PF_4444`，则改为 `HIGO_PF_8888` 或 `HIGO_PF_1555`。

1.1.11 如何解决动画框架移动不流畅的问题

【现象】

使用 HiGV 动画接口移动控件，比如 `groupbox` 控件，从屏幕外移动到屏幕当中显示，移动的过程中，出现卡顿、不流畅的问题。

【分析】



控件的移动过程是一个控件不断绘制的过程，所以尽可能的减少绘制的时间，减少冗余的绘制，可有效的提高控件移动的性能表现。

【解决】

- 控件移动调用接口 `HI_GV_Widget_Move`，这个接口内部包含了绘制的过程，无需再调用 `HI_GV_Widget_Update` 或 `HI_GV_Widget_Paint` 等接口进行重复绘制了。
- 查看移动过程中的接口调用，是否有调试打印信息或其他多余的操作，一并都进行删除处理。

1.1.12 如何解决接口并发调用异常问题

【现象】

在多个子线程中调用 HiGV 接口进行显示，隐藏，绘制等操作，导致出现不可预期的异常。

【分析】

为了提高消息分发的效率和绘制的性能，HiGV 内部大多接口都是没有做多线程访问保护，导致在子线程中进行接口调用，可能会导致一些非预期的问题出现。

【解决】

- 可以理解 HiGV 是一个单线程的模块，所有接口的调用都在主线程中完成，不要用于子线程调用 HiGV 的接口。
- 如果有子线程需要通知控件进行业务处理，可通过发送异步消息来实现，异步消息发送接口是 `HI_GV_Msg_SendAsync`，这个接口是做过多线程保护的，子线程使用没有问题，前提是在主线程中有注册过控件对应的消息处理函数。Sample 代码实现请参考发布包目录文件
“middleware/sample/higv/scene_sample/asy_msg_scene.c”。

注意

异步消息发送接口请参考 `hi_gv_msg.h` 文件，`HI_GV_Msg_SendAsync` 可以满足大部分场景的需求，如果需发送一段内存数据给主线程，请使用 `HI_GV_Msg_SendAsyncWithData` 接口。

1.1.13 如何解决 Listbox 控件滑动时界面叠加的问题

【现象】

Listbox 控件开发完成后，在触屏上面上下滑动控件内容，UI 界面出现绘制叠加的问题。

【分析】

控件进行移动的过程，是一个不断进行 UI 绘制的过程，绘制依赖于皮肤资源，需要检测是否正确设置了控件的皮肤。

【解决】



出现这个问题的原因一般是只设置了 Listbox 控件的 Normalskin 皮肤，而没有设置 Activeskin 皮肤，需要检查布局 xml 文件的皮肤设置是否出现这种问题。

1.1.14 如何解决控件快速移动导致的裂屏问题

【现象】

在触摸屏场景下，手指在 Listbox 控件或 Scrollview 控件上来回移动，控件内容会高概率出现裂屏的问题。

【分析】

HIFB 的刷新方式有三种，分别是 0 buffer、1 buffer 和 2 buffer，含义如下。

- 0 buffer（即 HIFB_LAYER_BUF_NONE）
上层用户的绘制 buffer 即为显示 buffer。该刷新类型可以节省内存消耗，速度也最快，但是用户会看到图形的绘制过程。
- 1 buffer（即 HIFB_LAYER_BUF_ONE）
显示 buffer 由 HIFB 提供，因此需要一定的内存。该刷新类型是对显示效果和内存需求的一种折中考虑。但是会有锯齿。
- 2 buffer
显示 buffer 由 HIFB 提供。和前面的刷新类型相比，其要求内存最多，但图形显示效果最好。

【解决】

从分析中可以得出，0 buffer 和 1 buffer 刷新方式在显示上都存在一定的缺陷，可能会导致裂屏的问题，需要把刷新方式修改为 2 buffer。对应到 HiGV 应用程序，就是在图层初始化的时候，把刷新方式的参数设置为 2 buffer，即传入参数是 HIGO_LAYER_BUFFER_OVER。

```
HIGO_LAYER_INFO_S LayerInfo = {.....,  
(HIGO_LAYER_FLUSHTYPE_E)(HIGO_LAYER_BUFFER_OVER),  
HIGO_LAYER_DEFLICKER_AUTO, HIGO_PF_1555, HIGO_LAYER_HD_0};
```

1.1.15 如何解决适配触摸屏时触摸手势事件回调无效问题

【现象】

在触摸屏场景下，手指在触摸屏上滑动或点击操作时控件的手势回调函数无响应的问题。

【分析】

适配触摸屏需要修改 touchpadadp.c 文件，该文件主要做以下工作。

- HI_GV_TOUCH_OpenDevice; HI_GV_TOUCH_CloseDevice
可以通过以上接口打开或关闭触摸设备。
- HI_HAL_TOUCHPAD_Enable
该接口主要使能相关标志位，使能该标志位以便正常读取触摸数据。
- HI_GV_TOUCH_ReadData



该接口主要读取触摸操作时的相关数据。

```
typedef struct hiHIGV_TOUCH_INPUTINFO_S
{
    HI_S32      id;/**<input id info, one finger or two fingers*/
    HI_S32      x;/**<x coordinate absolute*/
    HI_S32      y;/**<y coordinate absolute*/
    HI_U32      pressure;/**<is press on screen: 0, 1*/
    HI_U32      timeStamp;/**<time stamp*/
} HIGV_TOUCH_INPUTINFO_S;
HI_GV_Register_TouchDevice
```

该接口主要注册相关回调函数以便将读取的触摸数据传送给 HIGV，以便 HIGV 识别相关手势动作。

【解决】

从分析中可以得出

- 在添加 touchpadadp.c 文件后首先要确定 HI_GV_TOUCH_ReadData 函数是否有正确调用，可以在其内部添加相关打印并将其接收到的数据通过串口打印出来。如果该接口并无数据打印则确认 HI_HAL_TOUCHPAD_Enable 使能标志位接口是否有调用，确认 s_bEnableFlg 该标志位使能；
- 如果 HI_GV_TOUCH_ReadData 函数有数据输出，则确认其输出数据是否正确，其中 id 为 0 代表单指触摸，id 为 1 代表两指触摸；x,y 分别代表触摸点的 x 轴和 y 轴坐标；pressure 值代表触摸屏的压力状态，手指与屏幕接触状态下 pressure 值一直为 1，离开屏幕时 pressure 值为 0；（如果手指在屏幕上滑动时读取的 pressure 值一直在变化，但是非 0，则要在驱动上报数据时将 pressure 值转化为 1。）

1.1.16 如何解决文字内容不能正常显示的问题

【现象】

在 xml 文件中设置文字内容后，UI 界面上不能正确的显示出来。

【分析】

字体内容不能正常显示原因可能有如下：

- 字体内容的宽高大于等于字体所在控件内容位置的宽高，例如：字体设置大小为 30 号，而字体所在位置的控件高度为 25，此时字体内容会显示不出来。
- 指定字库文件路径下没有 xml 中配置的字库文件。
- 字库文件不支持设置的字体格式，例如：字库文件仅支持中文和英文，而代码中设置字体格式为韩文。
- 语言 xml 文件编码格式不是 UTF-8 格式。

【解决】

- 保证字体设置的宽高小于所在位置控件内容的宽高。
- 在指定字库文件路径下放置正确的字库文件。
- 保证字库文件可以支持代码中设置的字体格式。

- 语言 xml 文件编码格式设置为 UTF-8。

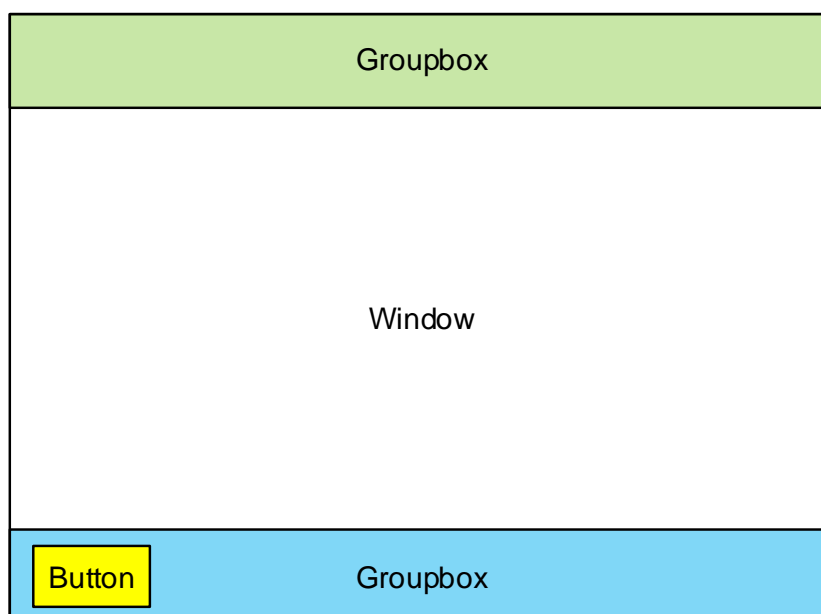
1.2 HiGV 典型场景示例

1.2.1 小型触摸屏提升 Button 控件体验的方法

【现象】

如图 1-1 所示，这是一个典型的运动 DV 主界面场景，背景是一个 Window 窗口，上下顶端是 Groupbox 控件，Groupbox 控件中包含了一些 Button 控件，为了简化流程，示例图中只包含了一个 Button 控件。由于屏幕尺寸本身比较小，Button 控件的布局相对会更加的小，那么在触屏上去点击 Button 控件的体验会比较差，容易出现误操作的情况。

图1-1 典型主界面



【分析】

除了点击 Button 控件可以进行操作外，再增加另外一个场景，即手指点击 Button 控件后，手指不离开屏幕，继续在屏幕上滑动一段距离后，再离开屏幕，此时也可以对最初点击的 Button 控件进行响应操作。

【解决】

步骤 1 注册事件。

在 Window 窗口的 xml 布局上注册一个 ontouchaction 回调函数，ontouchaction 事件是触摸手势的基础事件，能接收到所有的手势消息，包含 HIGV_TOUCH_START（按下事件），HIGV_TOUCH_END（抬起事件）和 HIGV_TOUCH_MOVE（移动事件）。

步骤 2 获取顶层控件句柄。



当手指在屏幕上按下时，通过 `HI_GV_Widget_GetWidgetByPos_TouchDevice` 接口可以获取到当前坐标对应的最顶层控件的句柄。

步骤 3 增加标志位。

判断顶层控件句柄是否是 `Button` 控件，如果是 `Button` 控件，则对这个 `Button` 控件增加一个标志位。

步骤 4 响应离开动作。

手指继续在屏幕上移动，此时 `Button` 控件不做任何响应，当手指离开屏幕的时候，可以解析出 `HIGV_TOUCH_END` 事件，此时通过异步消息接口，可以把事件通知给在前面做过标记的 `Button` 控件，然后 `Button` 控件做出对应的响应操作。

Sample 代码实现请参考发布包目录 “`middleware/sample/higv/scene_sample`”。

----结束

1.2.2 触摸屏提升轻扫动作灵敏度的方法

【现象】

在触摸屏主界面有一个典型的应用场景，即在主窗口上进行上下左右轻扫手势动作，弹出不同的界面或切换出其他窗口。这里一般的处理方式是在窗口的 `xml` 控件上注册一个 `ongesturefling` 事件的回调函数，然后在回调函数内部对方向进行判断后，进行响应的响应动作。这里存在着一个问题，就是 `fling` 事件的触发是需要一定的速度的，即手指要较快的在触摸屏上轻扫才会有响应，如果轻扫的速度不够，达不到触发 `fling` 事件的阈值，就不会产生任何反应，这个是有可能会不符合消费者的预期的。

【分析】

为了提高在主窗口进行轻扫动作的灵敏度，可以基于 `ontouchaction` 基础触摸事件进行分析组合，即通过按下、移动、抬起这三个动作来判断是否进行了轻扫动作。

【解决】

步骤 1 注册事件。

在 `Window` 窗口的 `xml` 布局上注册一个 `ontouchaction` 回调函数。

步骤 2 获取顶层控件句柄。

当手指在屏幕上按下时，通过 `HI_GV_Widget_GetWidgetByPos_TouchDevice` 接口可以获取到当前坐标对应的最顶层控件的句柄。

步骤 3 增加标志位。

判断顶层控件句柄是否是主窗口 `Window` 控件，如果是 `Window` 控件，则对这个 `Window` 控件增加一个标志位，并记录下按下位置的坐标信息。

步骤 4 获取离开屏幕事件信息。

手指继续在屏幕上移动，当手指离开屏幕的时候，可以解析出 `HIGV_TOUCH_END` 事件，并记录下抬起位置的坐标信息。

步骤 5 解析坐标。



对按下位置的坐标信息和抬起位置的坐标信息做差值计算，计算出 x 轴方向的位移和 y 轴方向的位移，并比较坐标值，计算出滑动方向，通过这两组信息即可得出轻扫的方向。

Sample 代码实现请参考发布包目录“middleware/sample/higv/scene_sample”。

----结束

1.2.3 触摸屏设置界面的使用控件选择

【现象】

如图 1-2 所示，是一个典型的运动 DV 或行车记录仪形态产品的参数设置界面，支持通过手势上下滑动可以进行参数的选择，控件的右侧绑定了一个 Scrollbar 滑动进度条。当前 HiGV 有两个控件可以达成类似的效果，分别是 Listbox 和 Scrollview，那么这两个控件区别是什么，如何进行选择呢。

图1-2 典型设置界面



【分析】

- Listbox 控件和 Scrollview 控件的详细描述请参考《HiGV 开发指南》文档 4.7 和 4.24 章节内容。
- Listbox 控件可以绑定数据模型，支持批量导入数据，比较适用于数据量较大，且类型单一的场景。
- Scrollview 是一个容器类控件，里面是可以放置子控件类型的，比如 Button、Label、Image 等，可以承载的类型更加丰富。
- Listbox 和 Scrollview 控件都可以绑定 Scrollbar 控件，这点使用上基本是一致的。

【解决】

- 如果设置界面设计承载的 UI 比较复杂，推荐使用 Scrollview 控件，通过其内部可以实现很多花样的变化。
- 如果设置界面内容比较单一，以文本为主，且设置信息条目较多，那么比较推荐使用 Listbox 控件，Listbox 在 xml 布局文件中只需设置一次，可以支持多个页面公用一个 Listbox 控件，切换页面时，调用刷新数据的接口，同步一次数据即可，工作量上相对较小。

1.2.4 运动 DV 产品典型菜单界面设计方法

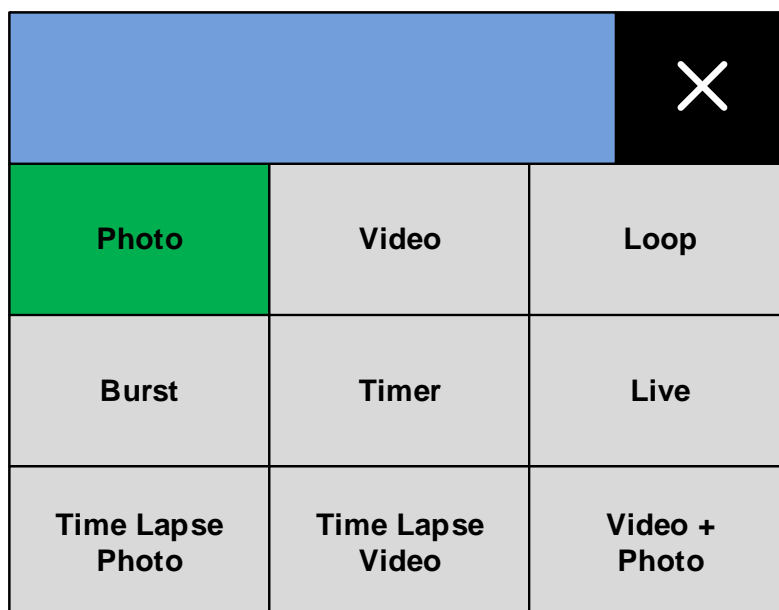
【现象】

介绍运动 DV 形态产品，菜单界面的典型设计方法。

【分析】

如图 1-3 所示，这个界面是一个典型的运动 DV 形态产品的菜单界面。

图1-3 典型菜单界面



【解决】

- 在 Window 窗口内部创建两个控件，分别是 Groupbox 和 Scrollview，GroupBox 控件位于窗口的顶端，Scrollview 控件位于 Groupbox 控件的下面。
- Groupbox 控件是容器类控件，可以放 Button 或 Label 控件，用于控制界面的关闭和显示信息。
- Scrollview 控件也是容器类控件，里面放一些 Button 控件来作为菜单显示，Button 控件需要设置 Normalskin 和 Activeskin，用于表征哪个处于选中状态。
- Scrollview 控件需要绑定一个 Scrollbar 控件，用于显示当前滑动进度。

1.2.5 Scrollbar 控件皮肤的设置方法

【现象】

介绍 Scrollbar 的两种类型的皮肤设置方法。

【分析】

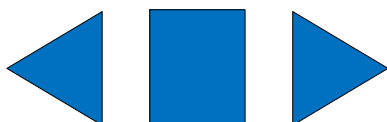
和其他控件一样，Scrollbar 的皮肤类型也分为 pic 和 color 两种，在 skin.xml 中配置。HiGV 的皮肤是由九宫格组成，分为上、下、左、右、左上、右上、左下、右下和中

间。设置为 **color** 类型的皮肤一般只需设置中间位置皮肤即可。**pic** 类型的皮肤就需要根据具体情况来分析了，**Scrollbar** 控件在方向上分为两种，即水平方向和垂直方向，不同的方向需要设置不同格式的的皮肤。

【解决】

水平方向的 **Scrollbar**，一般情况需要设置左、中、右这三个位置的皮肤。其中中间位置的皮肤是一个矩形区域，左和右位置的皮肤则是方向向左和方向向右的箭头图片。如图 1-4 所示，图中三张图片从左到右依次是左、中、右的皮肤。

图1-4 水平方向皮肤示例



XML 示例:

```
<skin
    id="horizontal_skin "
    type="pic"
    isnodrawbg="no"
    linewidth=""
    ltopidx=""
    toplineidx=""
    rtopidx=""
    llineidx="./res/pic/left.png"
    bgidx="./res/pic/middleware.png"
    rlineidx="./res/pic/right.png"
    lbtmidx=""
    rbtmidx=""
    btmlinedx=""
fgidx="0xFFFFFFFF" />
```

- 垂直方向的 **Scrollbar**，一般情况需要设置上、中、下这三个位置的皮肤。其中中间位置的皮肤是一个矩形区域，上和下位置的皮肤则是方向向上和方向向下的箭头图片。如图 1-5 所示，图中三张图片从上到下依次是上、中、下的皮肤。

图1-5 垂直方向皮肤示例



XML 示例:

```
<skin
  id="vertical_skin "
  type="pic"
  isnodrawbg="no"
  linewidth=""
  ltopidx=""
  toplineidx="./res/pic/top.png"
  rtopidx=""
  llineidx=""
  bgidx="./res/pic/middleware.png"
  rlineidx=" "
  lbtmidx=""
  rbtmidx=""
  btmlinedx="./res/pic/button.png"
  fgidx="0xFFFFFFFF" />
```

Sample 代码实现请参考发布包目录“middleware/sample/higv/scene_sample”。

1.2.6 Scrollbar 控件悬浮的设置方法

【现象】

介绍 Scrollbar 控件悬浮设置的方法。

【分析】

Listbox 控件和 Scrollview 控件可以支持绑定 Scrollbar，在触摸屏场景，Scrollbar 悬浮在底层控件上面体验更好。

【解决】

以 Listbox 控件为例，分别介绍 Scrollbar 控件非悬浮和悬浮的设置方法，主要是差异是布局文件 xml 的宽度或高度参数的设置。



- 非悬浮设置布局

```
<scrollbar
id="listbox_scrollbar"
top="0"
left="310"
width="10"
height="240"
normalskin="common_skin"
slidres="scrollbar_res_skin"
.....
/>

<listbox
id="MENUSET_LISTBOX"
top="0"
left="0"
width="310"
height="240"
isrelease="yes"
scrollbar="listbox_scrollbar"
normalskin=" common_skin"
activeskin="common_skin"
highlightskin=""
font="simhei_font_text"
.....
/>
```

- 悬浮设置布局

```
<scrollbar
id="listbox_scrollbar"
top="0"
left="310"
width="10"
height="240"
normalskin="common_skin"
slidres="scrollbar_res_skin"
transparent="yes"
.....
/>

<listbox
id="MENUSET_LISTBOX"
top="0"
left="0"
```



```
width="320"  
height="240"  
isrelease="yes"  
scrollbar="listbox_scrollbar"  
normalskin=" common_skin"  
activeskin="common_skin"  
highlightskin=""  
font="simhei_font_text"  
.....  
</>
```

- 通过比较 Scrollbar 非悬浮和悬浮布局，两者的主要区别就在于 Listbox 控件宽度的设置，非悬浮情况需要给 Scrollbar 预留一定的空间，而悬浮情况无需预留空间。
- Scrollbar 悬浮会把其背景属性设置为透明，即 `transparent="yes"`，而非悬浮布局一般不会设置背景为透明属性。
- Scrollview 控件绑定 Scrollbar 的情况和 Listbox 基本是一致的，需要关注的是 Scrollview 控件是由水平和垂直两个方向，而 Listbox 只有垂直方向，相应的如果是水平方向布局的 Scrollview 控件，需要修改的是布局中的高度属性，而不是宽度属性了。

Sample 代码实现请参考发布包目录“`middleware/sample/higv/scene_sample`”。

1.2.7 触摸屏时间日期功能开发

【现象】

触摸屏上开发时间或日期 UI 功能，需要支持通过手势滑动进行内容切换。

【分析】

推荐使用 Wheelview 控件组合实现类似的效果，三个 Wheelview 控件并排组合在一起可以达成年/月/日或时/分/上下午的效果。

【解决】

如图 1-6 所示，是一个典型的触摸屏日期控件，上半部分是三个 Label 或 Image 作为日期的标题，然后下面并排布局了三个 Wheelview 控件，并在控件内部设置不同的内容信息。

图1-6 典型日期控件

MONTH	DAY	YEAR
6	26	2016
7	27	2017
8	28	2018

Sample 代码实现请参考发布包目录“middleware/sample/higv/widget_sample”。

1.2.8 非触屏场景回放界面翻页方法

【现象】

介绍在非触屏场景，使用按键的方式，在回放界面（Scrollgrid 控件）进行翻页的接口。

【分析】

无。

【解决】

- 向下或向右翻页接口：HI_GV_ScrollGrid_PageForward。
- 向上或向左翻页接口：HI_GV_ScrollGrid_PageBackward。
- 直接翻页回到原点接口：HI_GV_ScrollGrid_MoveToOrigin。
- 直接翻页到最后位置接口：HI_GV_ScrollGrid_MoveToLast。
- 直接跳转到指定单元格接口：HI_GV_ScrollGrid_SetSelCell。

接口详细描述请参考头文件《hi_gv_scrollgrid.h》

1.2.9 多国语言设置方法

【现象】

介绍多国语言在 GUI 的设置方法。

【分析】

HiGV 支持多语言功能，通过接口设置，可以在不同国家语言之间进行切换。在实际应用中主要包括两种方式：

- 不同国家语言使用同一个字库。
- 不同国家语言使用不同的字库，可能某个或某些国家使用单独的字库。

**【解决】**

- 不同国家语言使用同一个字库，设置方法可参考《HiGV 开发指南》文档中 3.5.3 章节内容，设置方式比较简单。
- 不同国家语言使用不同的字库，需要通过字体集接口进行添加。
HI_GV_FontSet_Create 接口用于创建字体集，HI_GV_FontSet_AddFont 接口用于添加字体句柄到字体集，代码示例如下所示，首先创建一个字体集句柄 hFontSet，创建成功后，分别把 FONT20 和 FONT_KAI_20 加入到字体集中，这两个字体可以来源于两个字库，其中 FONT20 中包含了中文和日文，FONT_KAI_20 中包含了英文，在实际使用中，可以通过调用 HI_GV_Lan_Change 接口进行不同格式字体进行切换。

```
HI_HANDLE hFontSet;  
if(HI_SUCCESS == HI_GV_FontSet_Create(&hFontSet))  
{  
    (HI_VOID)HI_GV_FontSet_AddFont(FONT20, "zh;ja", hFontSet);  
    (HI_VOID)HI_GV_FontSet_AddFont(FONT_KAI_20, "en", hFontSet);  
}
```

Sample 代码实现请参考发布包目录“middleware/sample/higv/scene_sample”。

1.2.10 触摸屏相关消息注册回调函数的参数解析

【现象】

介绍如何解析触摸屏相关消息事件对应的回调函数的参数。

【分析】

触摸消息从类型上可分为两类，一类是基础触摸事件消息，另一类是触摸手势事件的消息。这两类消息对应的消息 ID 分别为：

- 基础触摸事件消息类型：HIGV_MSG_TOUCH。
- 手势事件消息类型：HIGV_MSG_GESTURE_TAP（轻触离开手势）、HIGV_MSG_GESTURE_LONGTAP（长按手势）、HIGV_MSG_GESTURE_FLING（轻扫手势）、HIGV_MSG_GESTURE_SCROLL（滑动手势）。

这两类消息类型对应的结构体是有差异的，解析方法也不同。

【解决】

基础触摸事件消息回调函数参数解析示例

```
HI_S32 CSlideUnlock::OnTouchAction(HI_U32 wParam, HI_U32 lParam)  
{  
    HI_S32 TouchLeft = 0;  
    HI_S32 TouchTop = 0;  
    HIGV_TOUCH_EVENT_S touchEvent = {0};  
    HIGV_TOUCH_E type = HIGV_TOUCH_BUTT;  
  
    HIGV_MemSet(&touchEvent, 0x0, sizeof(touchEvent));  
    HIGV_MemCopy(&touchEvent, (HIGV_TOUCH_EVENT_S*)lParam,
```




```
sizeof(touchEvent));

    TouchLeft = touchEvent.last.x;
    TouchTop = touchEvent.last.y;
    type = touchEvent.last.type;

    switch (type)
    {
        case HIGV_TOUCH_START:
        {
            (HI_VOID)TouchStart(TouchLeft, TouchTop);
            break;
        }
        case HIGV_TOUCH_END:
        {
            (HI_VOID)TouchEnd(TouchLeft, TouchTop);
            break;
        }
        case HIGV_TOUCH_MOVE:
        {
            (HI_VOID)TouchMove(TouchLeft, TouchTop);
            break;
        }
        default:
            break;
    }
    return HIGV_PROC_GOON;
}
```

解释:

- 消息 ID 类型对应为: **HIGV_MSG_TOUCH**。
- 触摸事件结构体类型: **HIGV_TOUCH_EVENT_S**, 结构体具体含义请参考《HiGV API 参考》文档 2.1.10 章节内容。
- **lParam** 参数是触摸事件结构体指针, 指向了一段内存数据; **wParam** 参数是触摸事件结构体的长度, 即 **lParam** 指向的内容的长度。
- **HIGV_TOUCH_START**、**HIGV_TOUCH_END** 和 **HIGV_TOUCH_MOVE** 这三个消息事件是触摸消息里面包含的最基础的三个事件, 即按下、抬起和移动, 所有的手势事件都是基于这三个事件进行的组合封装。
- 四个手势消息类型对应的结构体分别是 **HIGV_GESTURE_TAP_S**、**HIGV_GESTURE_LONGTAP_S**、**HIGV_GESTURE_FLING_S** 和 **HIGV_GESTURE_SCROLL_S**, 结构体具体描述请参考《HiGV API 参考》文档 2.1.10 章节内容, 参数解析方法和 **HIGV_MSG_TOUCH** 基本一致, 只需更换对应的数据结构体即可。