Mocor5 SimpleHome 框架介绍

版本： V1.0

日期：2018-05-31

声明

本文件所含数据和信息都属于紫光展锐机密及紫光展锐财产，紫光展锐保留所有相关权利。当您接受这份文件时，即表示您同意此份文件内含机密信息，且同意在未获得紫光展锐同意前，不使用或复制、整个或部分文件。紫光展锐有权在未经事先通知的情况下，对本文件做任何修改。紫光展锐对本文件所含数据和信息不做任何保证，在任何情况下，紫光展锐均不负责任何与文件相关的直接或间接的、任何伤害或损失。

前　言

文档说明

本文档主要介绍 Mocor5 中桌面应用 SimpleHome 的框架、两种不同 UI 模式和通用部分的客制化（按键定制、农历显示、手电筒功能以及语音播报功能）。

阅读对象

本文档适合基于 Mocor5 项目的桌面应用进行开发、维护以及客制化的人员参考。

内容介绍

本文档包括三个章节，分别为：

- 第一章：概述。介绍了简易桌面应用的开发背景；
- 第二章：SimpleHome 框架。描述了两种不同的 UI 模式以及 apk 的编译；
- 第三章：通用部分客制化。说明了按键、手电筒、农历日期以及语音播报功能的客制化。

文档约定

本文档采用下面醒目标志来表示在操作过程中应该特别注意的地方。

注意：

提醒操作中应注意的事项。

说明：

说明比较重要的事项。

相关文档

Mocor5 ClassicHome 客制化文档

Mocor5 GirdHome 客制化文档

目 录

图目录

第1章 概述

我司的 SimpleHome 桌面是一款针对 Mocor5 项目在 feature phone 上开发的 launcher。

该应用内部包含了两种不同的 UI 形态，分别称为 ClassicHome（图 1-1）和 GridHome（图 1-2）。可以根据项目需要自行选择终端显示的 UI 模式。另外，在桌面中可以通过长按某些按键达到快速打开某些界面的功能。

图 1-1 ClassicHome

图 1-2 GridHome

下面详细介绍 SimpleHome 应用的框架和两种 UI 模式以及通用部分客制化方法。

第2章 SimpleHome 框架

2.1 子目录介绍

如图 2-1 所示，在 SimpleHome 应用中其子目录主要包括 classichome、gridhome、common、WallpaperPicker、platform、libs、gradle 这七个部分。下面一一介绍各个子目录的主要功能。

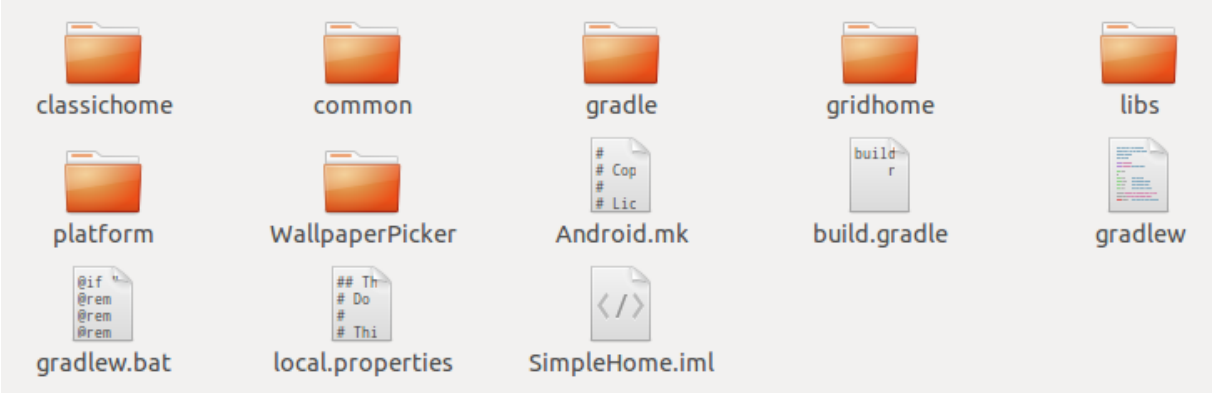

图 2-1 SimepleHome 的子目录图

1) classichome：classichome 模式是 SimpleHome 应用的两种 UI 模式中的一种。此目录主要包括对 classichome 模式的 UI 形态的定义，以及相应逻辑的处理。

2) gridhome：与 classichome 模式类似，gridhome 模式是 SimpleHome 应用的两种 UI 模式中的另一种。同理，此目录主要包括对 gridhome 模式的 UI 形态的定义，以及相应逻辑的处理。

3) common：该目录主要是封装了 classichome 和 gridhome 两种 UI 模式的通用逻辑，例如：按键事件的处理、手电筒的开启/关闭、农历日期的处理等。

4) WallpaperPicker：这是 SimpleHome 应用中内嵌的设置壁纸功能，包括壁纸的选择、预览以及设置。在 SimpleHome 应用以外可以通过 android.intent.action.SET_WALLPAPER 这个 action 隐式启动壁纸选择界面。

5) platform：此文件夹下的代码控制了在服务器和 android studio 中编译时，FeatureBarHelper 对象的创建与否。从而，避免了在 Android studio 上编译出的 SimpleHome apk 由于 FeatureBarHelper 对象而在运行时产生的 crash 问题。通常该部分无需关注。

6) libs：与一般应用下的 libs 文件夹一样，用于存放当前应用所涉及的库文件。通常无需关注。

7) gradle：这个目录下包含了 gradle wrapper 的配置文件，只有在 Android Studio 中编译应用时才会用到。通常不需要过多关心。

2.2 两种 UI 模式

如上一节所述，SimpleHome 应用提供了两种不同的 UI 模式，客户可根据项目需求自行配置。每种模式的 UI 界面具体如下。

A. ClassicHome 模式

图 2-2 主菜单界面-ClassicHome

图 2-3 亲情号界面-ClassicHome

如图 1-1 所示，ClassicHome 模式的 Idle 界面包含了时钟、日期（包括公历和农历，其中，农历只有在设备语言切换为中文时显示）的显示。具体农历日期的显示与否可在项目中自行配制，具体客制化方法见 3.2 章节。该界面的最下面一行是 feature bar，从文字描述可以看出：通过点击左、右软键可分别进入亲情号界面（图 2-3）和相机；中间 OK 键进入主菜单界面（图 2-2）。

这里的主菜单界面类似Launcher3中的主菜单界面。安装到手机的应用默认在主菜单界面显示，若需要隐藏应用入口，或者将应用入口移至 extra 列表中，则需要在

SimpleHome/classichome/res/xml/customize_apps.xml

中将相应应用的 app:group 属性配置为“hide”或者“extra”即可。具体修改可见 classichome 模式的详细文档《ClassicHome 客制化文档》。

B. GridHome 模式

图 2-4 亲情号界面-GridHome

这种模式下 UI 界面以六宫格形态展示，有且只有主界面的第一行由显示时钟和日期的 LinearLayout 代替，如图 1-2 所示。页面最下方的页面指示器则可实时显示当前页的位置以及页面数量。另外，不同页面的应用显示均是通过 xml 中的配置实现的，具体在

SinpleHome/gridhome/res/values/arrays.xml

中，包括了应用图标，宫格背景颜色以及页面中的位置等。这部分的客制化具体可见文档《GirdHome 客制化文档》。这里不再赘述。

最后，与 ClassicHome 中的农历日期显示方式类似，GridHome 中的农历日期的显示同样可以在项目工程文件中定制。具体可参考 3.2 章节。

说明：

GridHome 模式下，默认第二页为主界面（图 1-2），第一页为亲情号界面（图 2-4）。

2.3 APK 的编译

SimpleHome/Android.mk 文件主要可分为三个部分：

1. 库文件引用：

   include $(CLEAR_VARS)

   LOCAL_PREBUILT_STATIC_JAVA_LIBRARIES := \

   android-support-v4_new:libs/android-support-v4.jar

   include $(BUILD_MULTI_PREBUILT)

   无论编译哪种 UI 模式，该部分均会参与编译。所以，一旦修改了这里的静态库，两种 UI 模式都会被影响。

2. GridHome 模块编译部分：

   对应编译语句位于# Build rule for GridHome app 和# Build rule for ClassicHome app 这两句注释之间。

3. ClassicHome 模块编译部分：

   对应编译语句为# Build rule for ClassicHome app 这句注释之后的部分。

MakeFile 文件的这种写法保证了一旦选中某种模式，在编译过程中另一种模式的特有类和资源等均不会参与编译，更不会影响应用的最终运行。相比于那种应用的所有代码和资源参与编译的方式，这种方式可以说是在编译阶段就裁减了内存，有利于设备整体性能的提升。

在工程整体编译时，必须在/device/sprd/下项目的.mk 工程文件中配置 PRODUCT_PACKAGES 属性指定编译哪种模式，具体如下（以 ClassicHome 为例）：

PRODUCT_PACKAGES += \

ClassicHome

说明：

通过 mm 或者 mmm 指令单模块编译 SimpleHome 时，会同时生成 GridHome.apk 和 ClassicHome.apk，分别对应 GridHome 模式和 ClassicHome 模式。

第3章 通用部分客制化

无论是 ClassicHome 模式还是 GridHome 模式，均可引用 common 中的实现。目前， 在 common 中主要包括了一个焦点可以在上下左右方向任意移动的自定义控件 LoopGridView 以及各种工具类。接下来所说的客制化均是对工具类的客制化处理。

3.1 按键处理

SimpleHome 应用按键事件的处理主要是在 SimpleHome/common/src/com/sprd/common/util/KeyCodeEventUtil.java 中实现的。其中，

短按事件的处理：

public static boolean pressKeyEventForMainActivity(Context context, int keyCode, KeyEvent event)

长按事件的处理：

public static boolean longPressKeyEventForMainActivity(Context context, int keyCode)

注意：

个别按键的短按事件是在相应 UI 模式 Activity 的 onKeyUp 方法中处理的。例如，ClassicHome 模式的 Home.java 中就处理了 back 键、menu 键以及“确认”键的短按事件。

因此，对于按键的长按事件只需在 longPressKeyEventForMainActivity 中修改对应按键的处理逻辑或者添加对应的按键处理即可。而对于短按事件，建议将具有共性的按键事件在 pressKeyEventForMainActivity 中处理，那些具体 UI 模式所特有的一些按键事件则放到对应 activity 的 onKeyUp 中处理。

3.2 农历日期

1) 若项目需要支持桌面显示农历日期，只需在/device/sprd/路径下找到对应的.mk 工程文件，并添加如下属性即可：

```
PRODUCT_PROPERTY_OVERRIDES += \
ro.simplehome.lunar=true
```

不配置 ro.simplehome.lunar 属性，或者配置为 false 则均不支持显示农历日期。

2) 若需要调整农历日期的显示顺序，只需按需拼接农历日期的各个部分即可。

农历日期格式的对外接口在

SimpleHome/common/src/com/sprd/common/util/LunarCalendarConvertUtil.java

返回值为农历年份的方法：

public static String bulidLunarYear(Time time, Context context)

该方法的返回字符串为天干地支表示的年份与生肖年的拼接字串。可通过调整两部分年份的前后顺序来改变最终显示的农历年份。默认的农历年份显示格式是：天干地支年份+生肖年。

返回值为农历月份和日的方法：

public static String buildLunarMonthDay(Time time, Context context)。

在该方法种调用了 LunarCalendar.类的

public String[] getLunarCalendarInfo(boolean notDisplayLunarMonthForFirstDay)

方法，得到一个 length 为 6 的数组。其中，该数组的第一个元素为常用格式的年份，第二个元素为农历的月份，第三个元素为农历的日，第四个元素为传统的节日，第五个元素为国际上的规定的节日或者西方节日，第六个元素为农历中的二十四节气。在客制化农历时，可以通过对该数组元素的不同拼接方式得到不同的农历日期格式。这里默认农历日期的格式为：农历日期（不包括年份）+ 传统节日 + 二十四节气。

注意：

由于农历属于中国特有的一种历法， 因此，目前只有在中文简体和中文繁体下才支持该功能。

3.3 手电筒

手电筒的客制化与是否支持农历日期显示的客制化类似，同样需要在/device/sprd/路径下的相应.mk 工程文件中配置 PRODUCT_PROPERTY_OVERRIDES 属性，具体属性配置如下：

```
PRODUCT_PROPERTY_OVERRIDES += \
    ro.flashlight.node=/sys/devices/virtual/misc/sprd_flash/test \
    ro.flashlight.on_value=10 \
    ro.flashlight.off_value=11
```

其中，ro.flashlight.node 对应手电筒的节点文件所在路径；ro.flashlight.on_value 表示开启手电筒的节点值；ro.flashlight.off_value 表示关闭手电筒的节点值。

3.4 语音播报

工具类 SimpleHome/common/src/com/sprd/common/util/TextToSpeechUtils.java 采用单例模式，完全集成了语音播报功能。对外只有一个成员方法：

public static void speak(String str, Context context)

其中，第一个参数为要播报的字符串内容，第二个参数为上下文。

因此，在需要语音播报的地方直接调用 TextToSpeechUtils.speak 方法，并传入播报内容和上下文即可。


说明：

TextToSpeechUtils 使用单例模式可以有效防止多语音同时播报的现象。