android之activity横竖屏切换时的生命周期以及横竖屏切换时的资源适配方案

2015-01-12 21:20 1259人阅读 评论(0) 收藏 举报

分类:

Android进阶（21）

背景：之前有过两篇写activity的博客

android之activity的生命周期详解：详细介绍了activity的整个生命周期、各状态间的转换和返回桌面时保存activity的状态

android之activity中onSaveInstanceState和onRestoreInstanceState的触发时机：介绍了activity中这两个方法的触发时机和作用

本篇博客会牵扯到里面的内容，如果你都有所了解可以直接往下看，如果不了解可以进去回忆下。

问题：在做应用的退出对话框时，发现如果对话框弹出，然后把手机横竖屏切换一下，对话框就消失了。知道activity在横竖屏切换时会走一些生命周期，就误以为是在onPause和onResume的时候没有把对话框恢复出来，

首先想到就是设一个全局变量isDialogShow用来保存对话框是否显示，在onPause中检查下如果显示了，就把isDialogShow设置为true，然后在onResume中判断下isDialogShow的值，如果为true，则把对话框在显示出来。

以为这样就解决了，运行发现不对，对话框还是不显示。

然后debug发现，每次进入onResume的时候，isDialogShow都是false，也就是说每次isDialogShow都被重新初始化了。

这时候才突然意识到，横竖屏切换的时候生命周期可能不是我想的那样。

就借着这个机会去总结一下activity横竖屏切换时如何保存状态以及自适应布局。

解决方案：对于这个问题，有三种方案

1□顺应activity的生命周期，保存然后恢复状态，onCreate中动态适应布局。

2□在AndroidManifest.xml中设置，不让activity重走生命周期从而保存状态，然后通过onConfigurationChanged自适应布局

3□禁止activity的横竖屏转换，状态和布局保持原样。

方案一：

首先把activity的每个方法打印出来。

启动应用（此时是竖屏）：

Tag	Text
jason	onCreate
jason	onStart
jason	onResume
jason	0

切换为横屏：

jason	onPause
jason	1
jason	onSaveInstanceState
jason	onStop
jason	onDestroy
jason	onCreate
jason	onStart
jason	onRestoreInstanceState
jason	onResume
jason	0

里面的数字是加的一个全局变量（初始化为0，每进入一次onPause自动加1），用来看数值的变化，可见在onPause的时候还是1，等切换完成再回来的时候就成了0了。

也很容易理解，activity相当于完全重新做了一次初始化，完整走了一遍生命周期。

同时也注意到，切换为横屏的时候多了onSaveInstanceState和onRestoreInstanceState方法，上面第二篇博客里面说到的，activity在有“被销毁的可能”时，就会调用onSaveInstanceState方法，给你一个机会去保存activity中的数据，之后横竖屏切换时activity确实在“未经你许可的情况下被销毁了”，所以系统又会调用onRestoreInstanceState方法，让你把之前保存的数据恢复。

所以这样就明白了为什么我的对话框切换横竖屏之后不出来，而且用来记录的isDialogShow的值会被重置，如此解决方案也有了。

在onSaveInstanceState方法中保存isDialogShow的值，然后在onRestoreInstanceState方法中取回isDialogShow 的值，去判断该不该显示对话框就可以了。

那么如何自适应布局呢，像xml之类的资源，只要建两套就可以了，横屏资源放在res/layout-land里面,竖屏资源放在res/layout-port文件夹里面，资源名字都一样即可，系统会自己去找。

也可以通过以下方法获取目前横竖屏状态分别实现一些资源的加载。

```java
[java] view plain copy print ?
if (this.getResources().getConfiguration().orientation == Configuration.ORIENTATION_LANDSCAPE) {
 //do something
} else if (this.getResources().getConfiguration().orientation == Configuration.ORIENTATION_PORTRA
 //do something
}
```

以上就完成了在activity的完整生命周期里面保存状态以及自适应资源。

加载插

方案二:

在说方案二的时候要提一下，既然可以避免activity重新走生命周期，为什么还要设计这个横竖屏切换的时候去完整的重新走一遍生命周期呢?

因为既然手机切换了横竖屏，比如从竖屏切换为横屏了，那么原来竖屏的layout，drawable等资源可能无法适应横屏模式，设计完善的应用一般是会有两套资源来适配横竖屏模式的，所以在切换后重启activity，在新的创建过程中就会自

动使用横屏的资源，实现了动态适应。

方案二只需要在manifest文件中的activity标签下加如下代码

android:configChanges="orientation"

这段代码的含义是向系统说明，这个activity的orientation发生变化时，我不需要你去自适应，我自己会在onConfigurationChanged方法中做处理，我自己完成横竖屏切换的适应，即告诉activity不要重新走生命周期，放着我来。

加上这个代码声明之后，重新运行会发现，activity的生命周期没有出现，只出现了一个方法就是

那么切换过程中你要如何处理就是你自己的事情了，状态无需保存，都还是原来的，如果要做一些切换的话，可以通过如下代码获取到当前的横竖屏状态，然后自己操作。

```java
[java] view plain copy print ?
01. if (newConfig.orientation == Configuration.ORIENTATION_LANDSCAPE) {
02.  // do something
03. } else if (newConfig.orientation == Configuration.ORIENTATION_PORTRAIT) {
04.  // do something
05. }
```

如果只是加下面这个的话，你会发现activity还是会走完整的生命周期，onConfigurationChanged方法压根不会执行

android:configChanges="orientation"

这是因为从Android 3.2（API 13）开始，横竖屏切换时不仅orientation会发生变化，screenSize也会跟着发生变化，所以要改成下面这样。

加载插

android:configChanges="orientation|screenSize"

告诉activity，不管是orientation变化还是screenSize变化，都不要重新初始化，都通过onConfigurationChanged让我自己操作。

以上就是不让activity重新初始化，而是自己完成横竖屏切换的自适应操作。

方案三:

最简单最粗暴，禁止横竖屏转换，直接固定为横屏或者竖屏。

可以在配置Activity的地方进行如下的配置

```java
[java] view plain copy print ?
01. android:screenOrientation="portrait"
02. android:screenOrientation="landscape"
```

或者在onCreate里面

```
[java] view plain copy print ?
01. setRequestedOrientation(ActivityInfo.SCREEN_ORIENTATION_LANDSCAPE);
02. or
03. setRequestedOrientation(ActivityInfo.SCREEN_ORIENTATION_PORTRAIT);
```

这样就可以保证是竖屏总是竖屏了，或者横屏总是横屏。

以上，就不涉及状态保存和布局适应了，不管横屏竖屏一切如初。

参考：

http://www.jcodecraeer.com/a/anzhuokaifa/androidkaifa/2012/1106/516.html

http://blog.csdn.net/aliang775/article/details/8561075