**Launcher Design Doc**

**编 写 人：桌面品牌产品线**

版本历史

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 版本号 | 作者 | 操作 | 日期 | 说明 |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |

目录

[1 概述 3](#_Toc404672931)

[2 类图，框架 4](#_Toc404672932)

[3 代码分析 4](#_Toc404672933)

### 1 概述

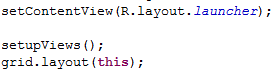
本文档主要介绍Launcher的整体布局情况，包括桌面的布局以及抽屉中的布局

### 2 类图，框架

### 3 代码分析

#### 一、整体概况预览



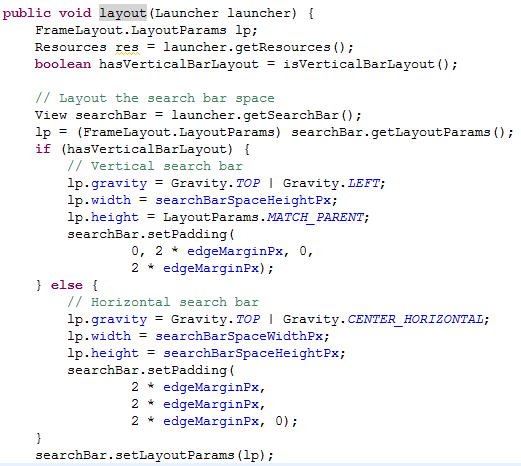




1. 布局的启示就是通过Launcher.java中的onCreate()方法中的setContentView()方法来加载，该方法一开始基本上就加载了整个桌面的整个布局，通过看launcher.xml文件我们可以看出整个桌面大致的布局方式，最外层是一个全屏的FrameLayout父容器，然后里面加了一个DragLayout，DragLayout是继承FrameLayout的，所以它的每个子项都是层叠的关系，首先加了一个WorkSpace布局，（我们暂且不去管这些自定义布局是怎么实现的，只用知道它是一个自定义的ViewGroup），后面又加了一层就是HotSeat，然后加了一层搜索条，然后的加的布局是长按状态下的下方显示的三个功能按钮，然后加的是当我们第一次使用桌面或文件夹等的新手引导布局部分，最后面才加的是抽屉的布局（它是一个TabHost）
2. setupViews()方法主要就是初始化在xml中相对应的view的变量；
3. grid.layout()方法主要是通过不同的设备调整相应的布局显示。



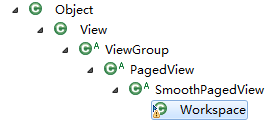




#### 二、主要自定义布局的分析

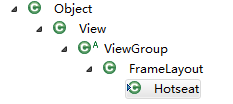
##### 1、WorkSpace.java

首先我们来看一下它的继承关系：



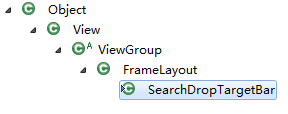
##### 2、HotSeat.java

首先我们来看一下它的继承关系：



##### SearchDropTargetBar.java

首先我们来看一下它的继承关系：



##### AppsCustomizeTabHost.java

首先我们来看一下它的继承关系：

