中兴Android系统界面软件设计与开发

**软件物理设计说明**

**版本：0.1**

编写： 张 翔

校对： 迟建涛

审核： 杨 帆

批准： 朱怡安

**西北工业大学－中兴通讯嵌入式系统联合开发实验室**

**2011年11月**

**目 录**

[1引言 5](#_Toc310697054)

[1.1文档标识 5](#_Toc310697055)

[1.2项目概述 5](#_Toc310697056)

[1.3文档概述 5](#_Toc310697057)

[1.4参考文档 6](#_Toc310697058)

[2 Launcher模块 7](#_Toc310697059)

[2.1 功能概述 7](#_Toc310697060)

[2.2 类图 7](#_Toc310697061)

[2.3 接口描述 9](#_Toc310697062)

[2.4 流程逻辑 28](#_Toc310697063)

[2.5 数据库设计 28](#_Toc310697064)

[3 动态壁纸模块 30](#_Toc310697065)

[3.1 星光漂移 30](#_Toc310697066)

[3.1.1 功能概述 30](#_Toc310697067)

[3.1.2 类图 31](#_Toc310697068)

[3.1.3 接口描述 31](#_Toc310697069)

[3.1.4 流程逻辑 33](#_Toc310697070)

[3.2 实时时间天气 33](#_Toc310697071)

[3.2.1 功能概述 33](#_Toc310697072)

[3.2.2 类图 35](#_Toc310697073)

[3.2.3 接口描述 36](#_Toc310697074)

[3.2.4 流程逻辑 37](#_Toc310697075)

[4桌面widget模块 38](#_Toc310697076)

[4.1 三维时间天气 38](#_Toc310697077)

[4.1.1 功能描述 38](#_Toc310697078)

[4.1.2 类图 39](#_Toc310697079)

[4.1.3 接口描述 39](#_Toc310697080)

[4.1.4 流程逻辑 40](#_Toc310697081)

[4.2 网络书签 40](#_Toc310697082)

[4.2.1 功能描述 40](#_Toc310697083)

[4.2.2 类图 42](#_Toc310697084)

[4.2.3 接口描述 42](#_Toc310697085)

[4.2.4 流程逻辑 43](#_Toc310697086)

[4.3 联系人 43](#_Toc310697087)

[4.3.1 功能描述 43](#_Toc310697088)

[4.3.2 类图 45](#_Toc310697089)

[4.3.3 接口描述 45](#_Toc310697090)

[4.3.4 流程逻辑 46](#_Toc310697091)

[4.4 社交网络 46](#_Toc310697092)

[4.4.1 功能描述 46](#_Toc310697093)

[4.4.2 类图 47](#_Toc310697094)

[4.4.3 算法及模型描述 47](#_Toc310697095)

[4.4.4 流程逻辑 48](#_Toc310697096)

[4.5 微博 48](#_Toc310697097)

[4.5.1 功能描述 48](#_Toc310697098)

[4.5.2 类图 50](#_Toc310697099)

[4.5.3 接口描述 50](#_Toc310697100)

[4.5.4 流程逻辑 50](#_Toc310697101)

[5 锁屏模块 51](#_Toc310697102)

[5.1 功能概述 51](#_Toc310697103)

[5.2 类图 51](#_Toc310697104)

[5.3 接口描述 51](#_Toc310697105)

[5.4 流程逻辑 52](#_Toc310697106)

[6 任务管理器模块 52](#_Toc310697107)

[6.1 功能概述 52](#_Toc310697108)

[6.2 类图 53](#_Toc310697109)

[6.3 接口描述 53](#_Toc310697110)

[6.4 流程逻辑 54](#_Toc310697111)

# 1引言

## 1.1文档标识

中文名称：《软件结构设计说明》。

英文名称：“Software Architecture Design Description（SADD）”。

文档版本：“0.1”。

文档编号：“SSM-ZTE-AndroidUI-SADD-0.1”。

## 1.2项目概述

本文档适用于“中兴Android系统界面软件设计与开发”项目（以下简称“AndroidUI项目”）的开发过程。AndroidUI项目由中兴通讯股份有限公司（以下简称“中兴通讯”）提出，由西北工业大学－中兴通讯嵌入式系统联合开发实验室（以下简称“联合实验室”）负责实施，该项目标识号为“SSM-ZTE-AndroidUI”，其软件产品版本号为“1.0”，包括三个内部版本，分别是0.1版、0.2版和0.3版。

项目内容为：

基于Android 3.2系统版本，针对中兴通讯设计开发的平板电脑，联合实验室依据中兴通讯提出的用户需求和设计的软件界面，深度定制Android系统的人机界面及应用软件界面，完成中兴Android系统界面软件的设计与开发。

## 1.3文档概述

本文档依据国家标准[《GB/T 8567-2006计算机软件文档编制规范》](../../资料/GBT%208567-2006%20计算机软件文档编制规范.pdf)制定，用来说明Android UI项目软件的架构，并设计软件模块划分、功能、接口以及关键算法。属于技术文档，仅限于联合实验室和中兴通讯的项目相关人员阅读。

## 1.4参考文档

* [《GB/T 8567-2006计算机软件文档编制规范》](../../资料/GBT%208567-2006%20计算机软件文档编制规范.pdf)，国家标准
* [《高校合作项目要求说明书－中兴Android系统界面软件设计与开发（讨论稿）》，](../../资料/高校合作项目要求说明书－中兴Android系统界面软件设计与开发（讨论稿）.doc)中兴通讯提供

# 2 Launcher模块

## 2.1 功能概述

Launcher，即启动器，作为Android操作系统的用户界面，为用户使用Android系统提供接口。它的主要功能包括应用程序菜单、多桌面以及页面编辑。

## 2.2 类图

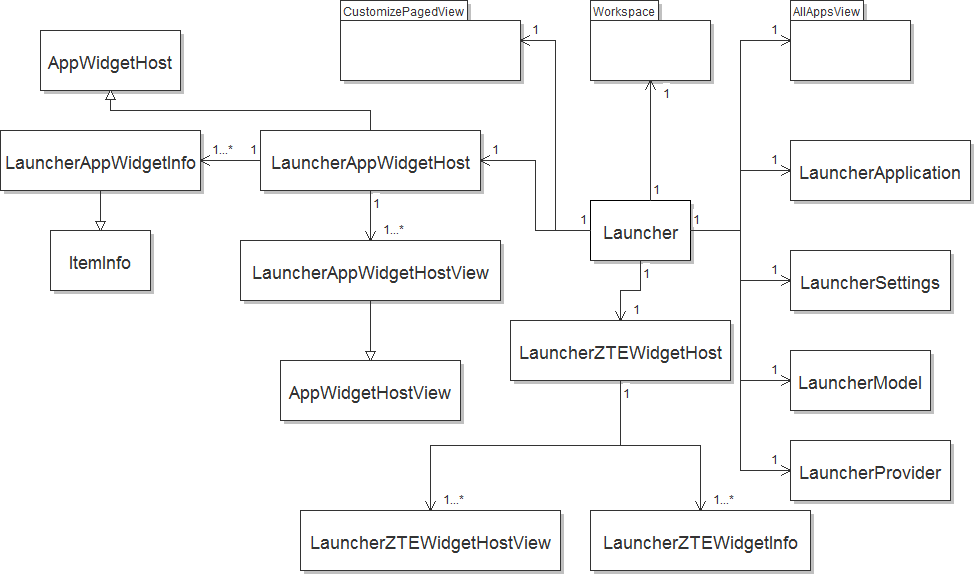


图2-1 Launcher主架构

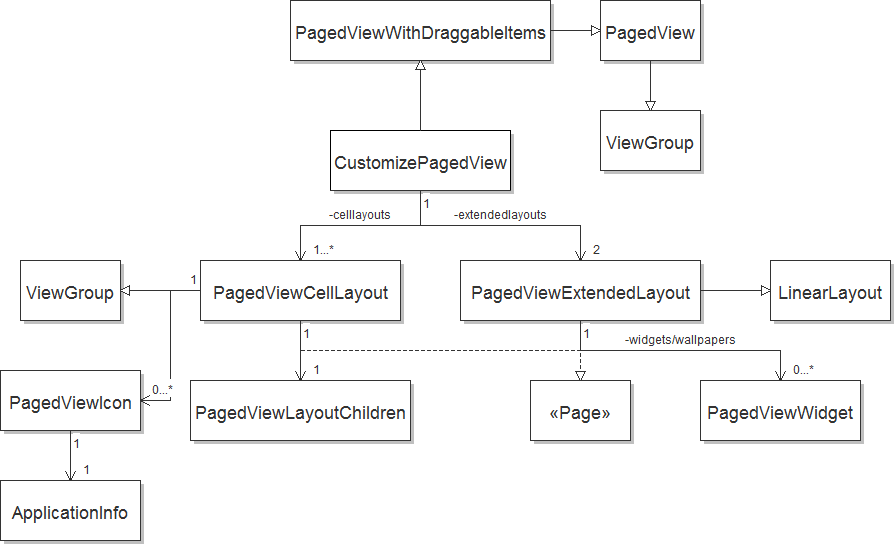


图2-2 页面编辑架构

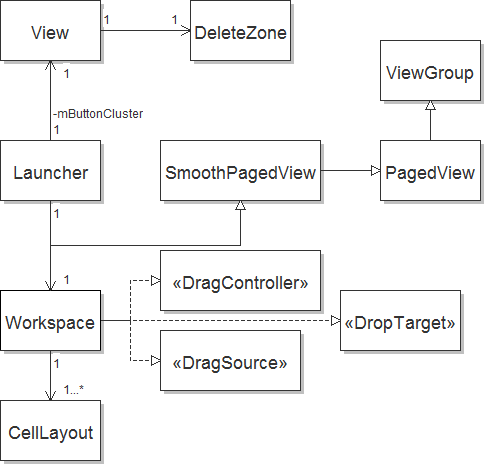


图2-3 多桌面架构

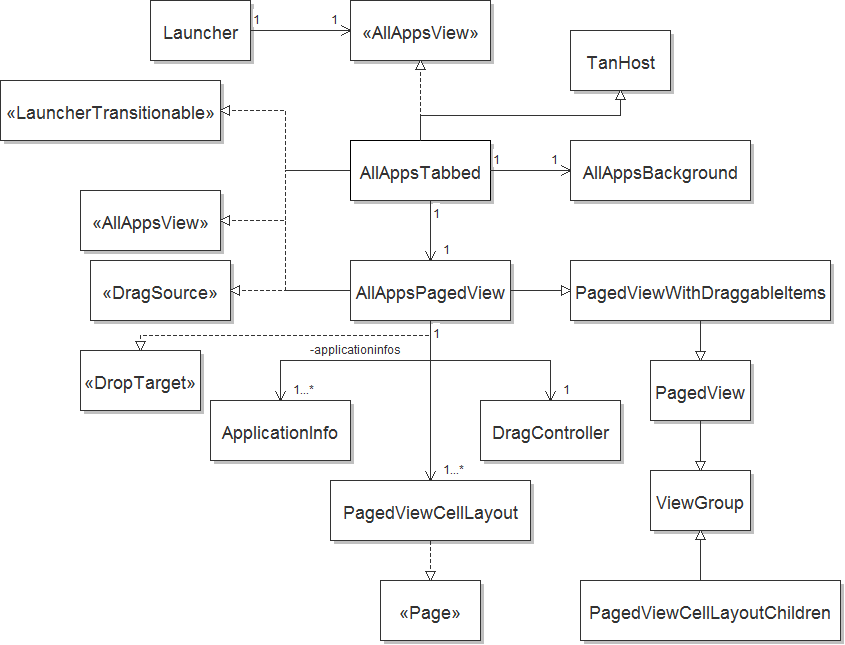


图2-4 应用程序菜单架构

## 2.3 接口描述

Launcher：Android启动器的主类，主要负责界面的初始化，数据库的读写，在桌面上加载widget，调用系统的接口，包括壁纸管理和应用程序管理等。



图2-5 Launcher类

LauncherApplication：配置Launcher应用信息，包括加载数据库，注册intent接收器。

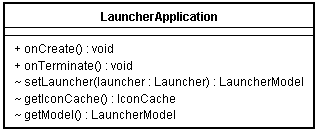


图2-6 LauncherApplication类

LauncherSettings：静态全局变量的定义。

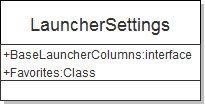


图2-7 LauncherSettings类

LauncherModel：提供数据库读写的接口。

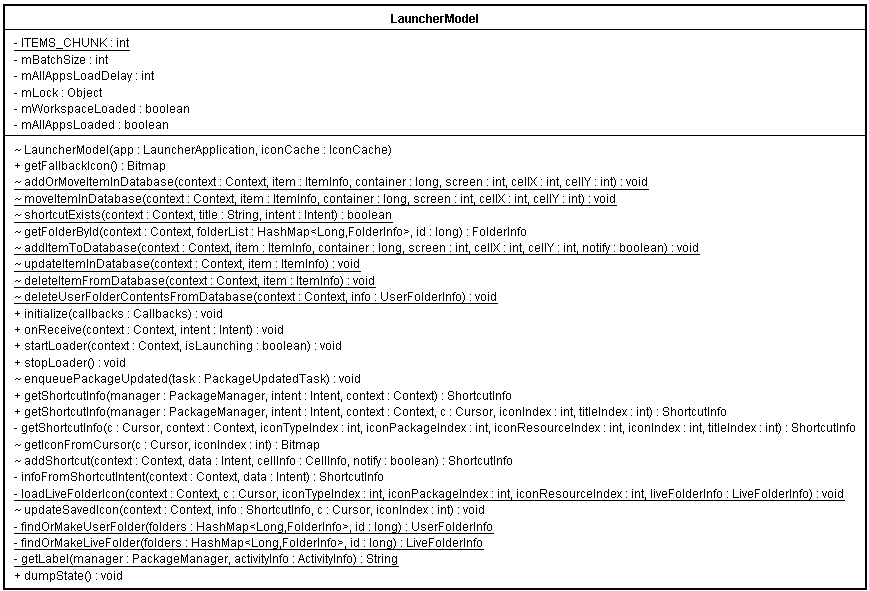


图2-8 LauncherModel类

LauncherProvider：创建数据库，表favorite保存Launcher的配置信息。

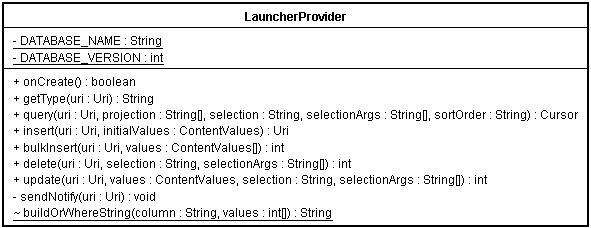


图2-9 LauncherProvider类

LauncherZTEWidgetHost：提供创建ZTE自定义widget视图的接口。

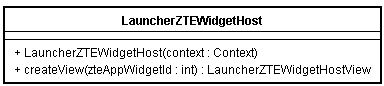


图2-10 LauncherZTEWidgetHost类

LauncherZTEWidgetInfo：主要负责维护ZTE自定义widget的信息，并写入数据库。

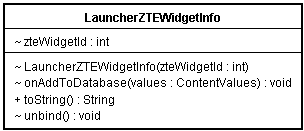


图2-11 LauncherZTEWidgetInfo类

LauncherZTEWidgetHostView：ZTE自定义widget的父类视图，并支持拖拽。

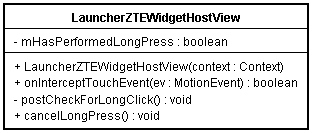


图2-12 LauncherZTEWidgetHostView类

LauncherAppWidgetHost：提供创建系统原生widget视图的接口。

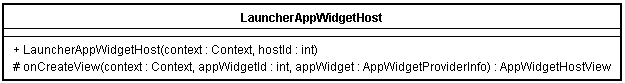


图2-13 LauncherAppWidgetHost类

LauncherAppWidgetHostView：系统原生widget的视图，并支持拖拽。

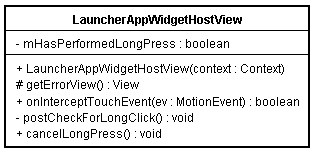


图2-14 LauncherAppWidgetHostView类

LauncherAppWidgetInfo：负责维护系统原生widget的信息，并写入数据库。



图2-15 LauncherAppWidgetInfo类

ItemInfo：维护数据库列表项，并统一存入数据库。

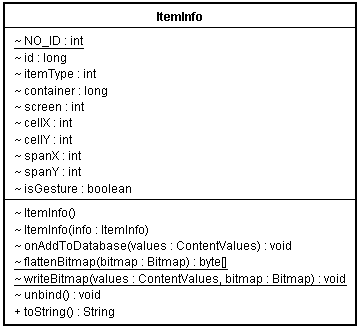


图2-16 ItemInfo类

Workspace：负责实现Launcher的多桌面，默认有5个桌面，并且可以水平方向上切换。

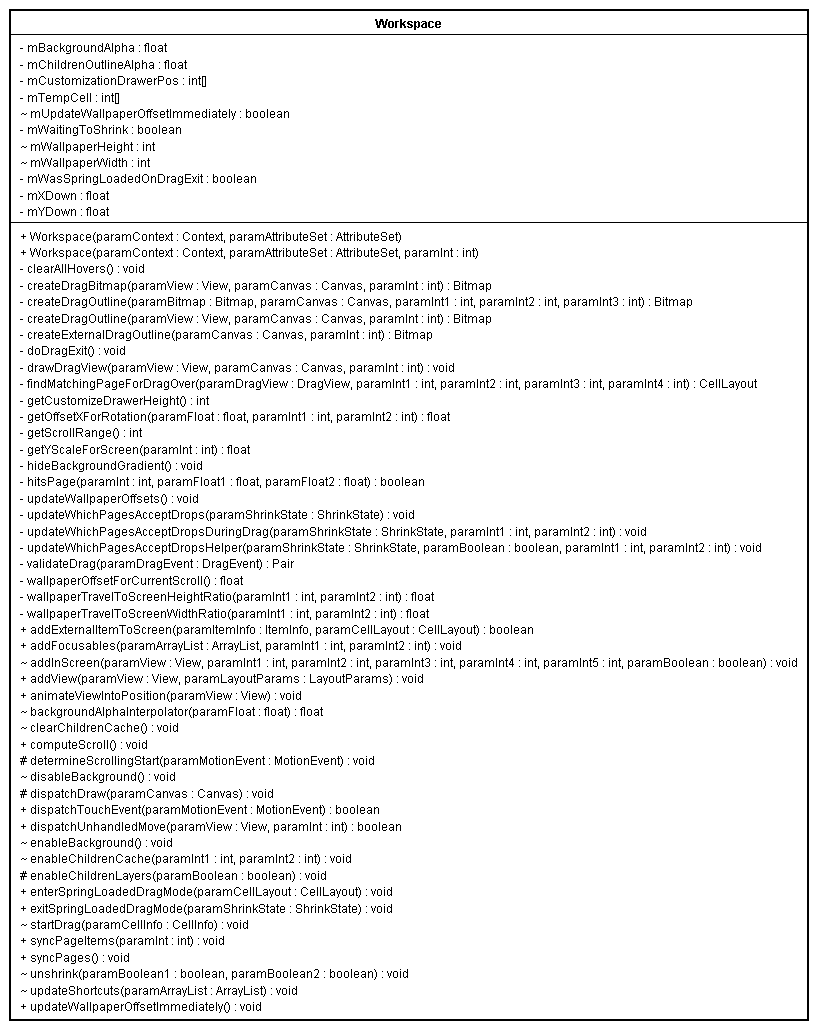


图2-17 Workspace类

CellLayout：负责实现单个桌面，桌面上可以加载widget和应用程序快捷方式。



图2-18 CellLayout类

SmoothPagedView：实现桌面平滑地切换功能。

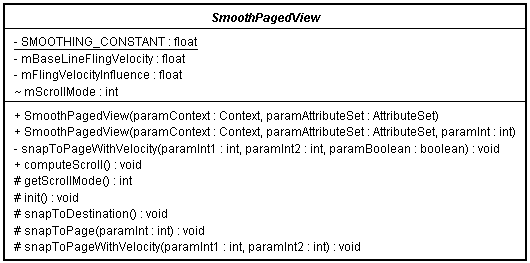


图2-19 SmoothPagedView类

PagedView：负责多桌面的布局。



图2-20 PagedView类

DragController：拖拽控制。

****

图2-21 DragController类

DragSource：Drag源接口。



图2-22 DragSource类

DropTarget：Drop目标接口。

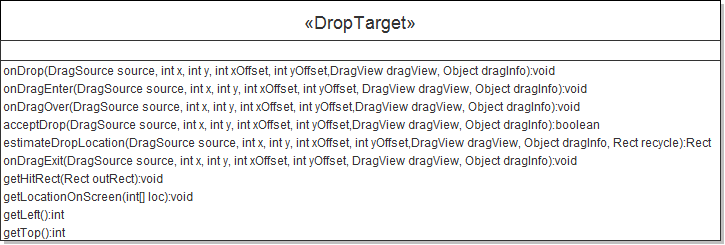


图2-23 DropTarget类

CustomizePagedView：实现页面编辑功能，负责与桌面进行交互，将widget、应用程序快捷方式拖入桌面。



图2-24 CustomizePagedView类

PagedViewWithDraggableItems：支持拖拽的多页面布局，用于应用程序菜单中。

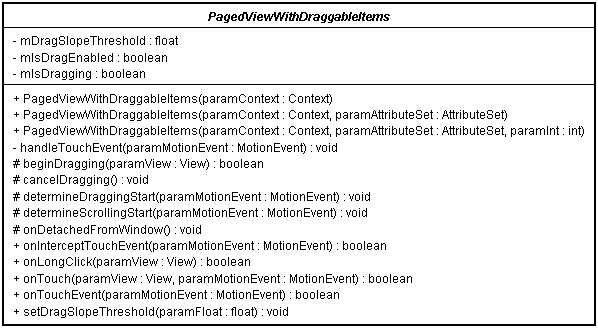


图2-25 PagedViewWithDraggableItems类

PagedViewCellLayout：负责实现应用程序菜单中的单页，以九宫格形式体现。



图2-26 PagedViewCellLayout类

PagedViewCellLayoutChildren：应用程序菜单中单个图标布局。

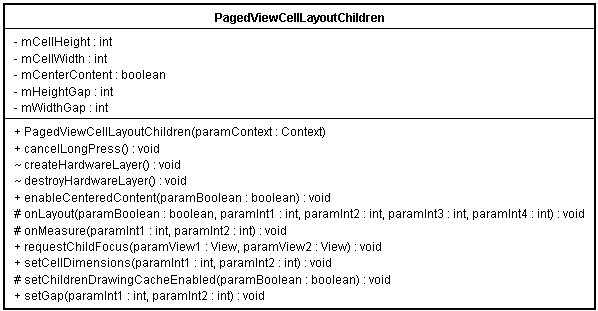


图2-27 PagedViewCellLayoutChildren类

PagedViewExtendedLayout：单行形式的多页面布局。

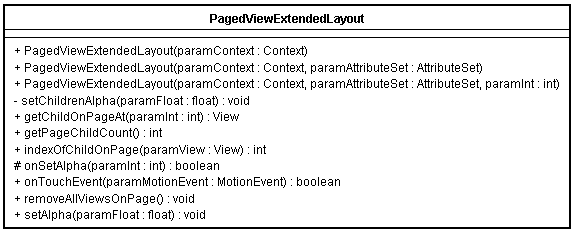


图2-28 PagedViewExtendedLayout类

Page：页面接口，负责提供实现child的索引。

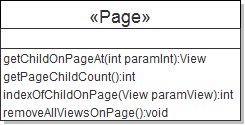


图2-29 Page类

PagedViewWidget：单行形式的多页面布局。

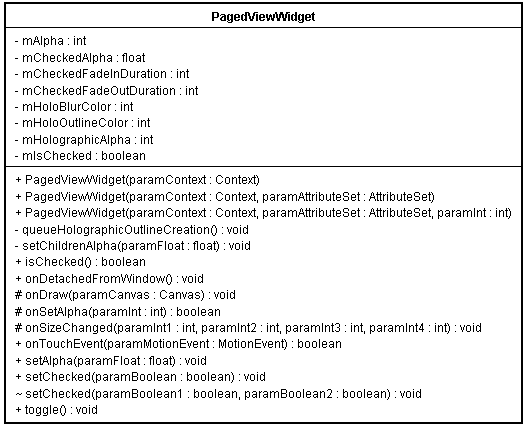


图2-30 PagedViewWidget类

PagedViewIcon：负责渲染页面中的单元。

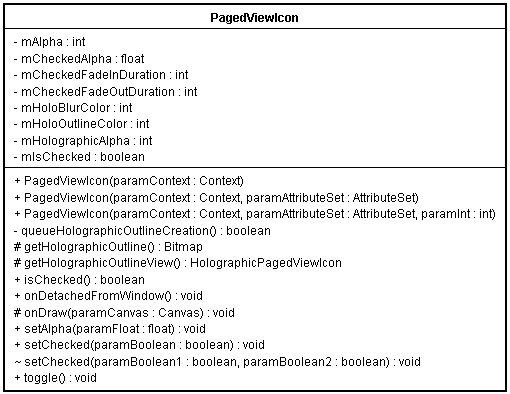


图2-31 PagedViewIcon类

AllAppsView：应用程序菜单视图。

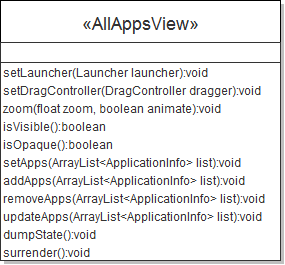


图3-32 AllAppsView类

AllAppsTabbed：Tab形式的应用程序菜单。

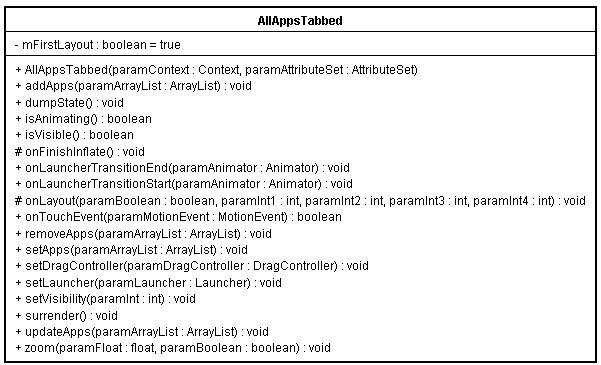


图2-33 AllAppsTabbed

AllAppsBackground：应用程序菜单背景。

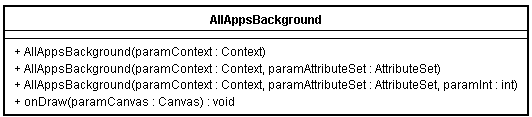


图2-34 AllAppsBackground类

AllAppsPagedView：应用程序菜单的多页面布局实现。

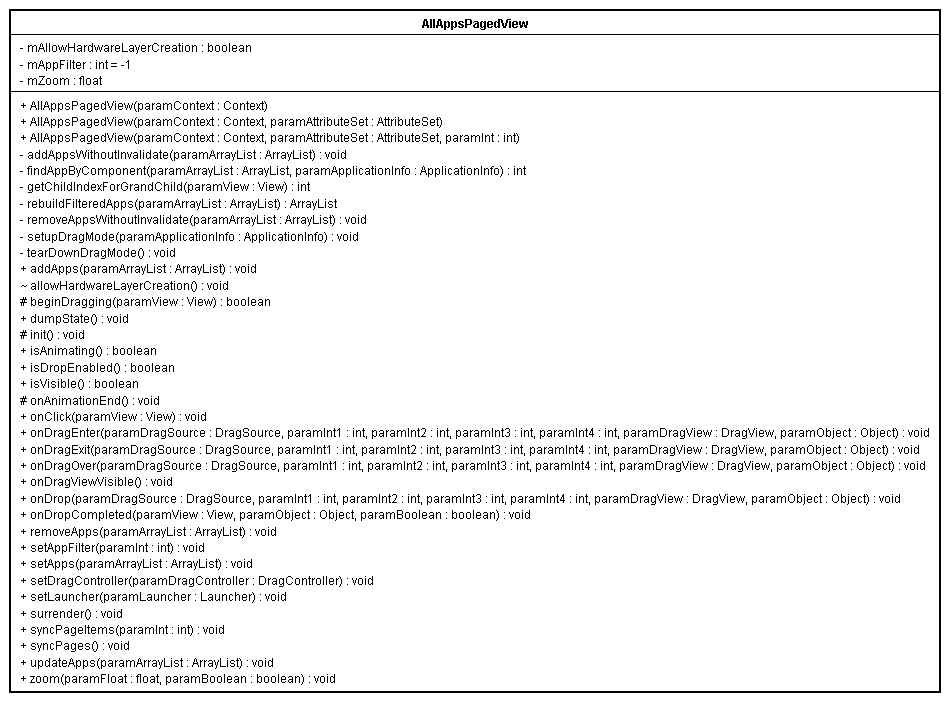


图2-35 AllAppsPagedView类

## 2.4 流程逻辑

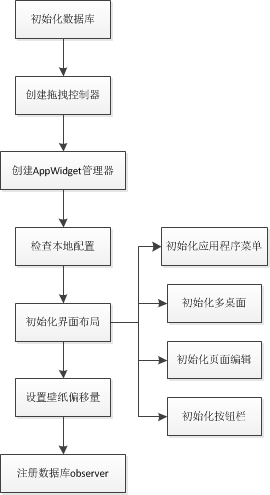


图2-36 Launcher数据表

## 2.5 数据库设计

Launcher数据库主要保存桌面widget的类型、位置以及大小信息，在Launcher启动时，从数据库读取信息并加载widget。在桌面添加或删除widget时，后台同时更新数据库。



图2-37 Launcher favorites数据表

# 3 动态壁纸模块

## 3.1 星光漂移

### 3.1.1 功能概述

桌面动态壁纸功能，在锁屏状态时正常显示，在调出Menu菜单和Edit设置时作为背景置灰显示，动态效果如图3-1所示。

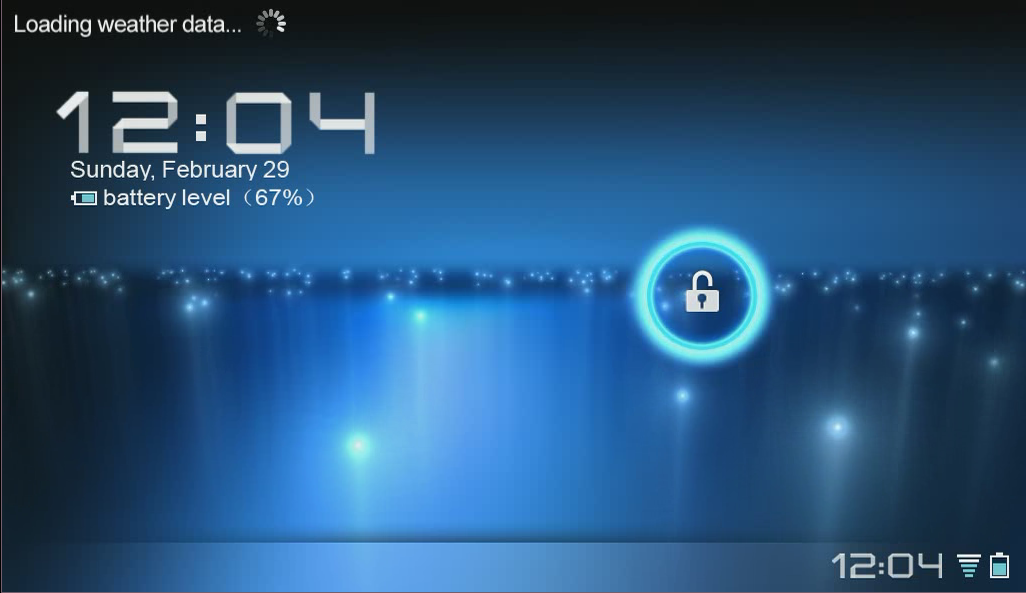


图3-1 星光漂移动态壁纸

### 3.1.2 类图

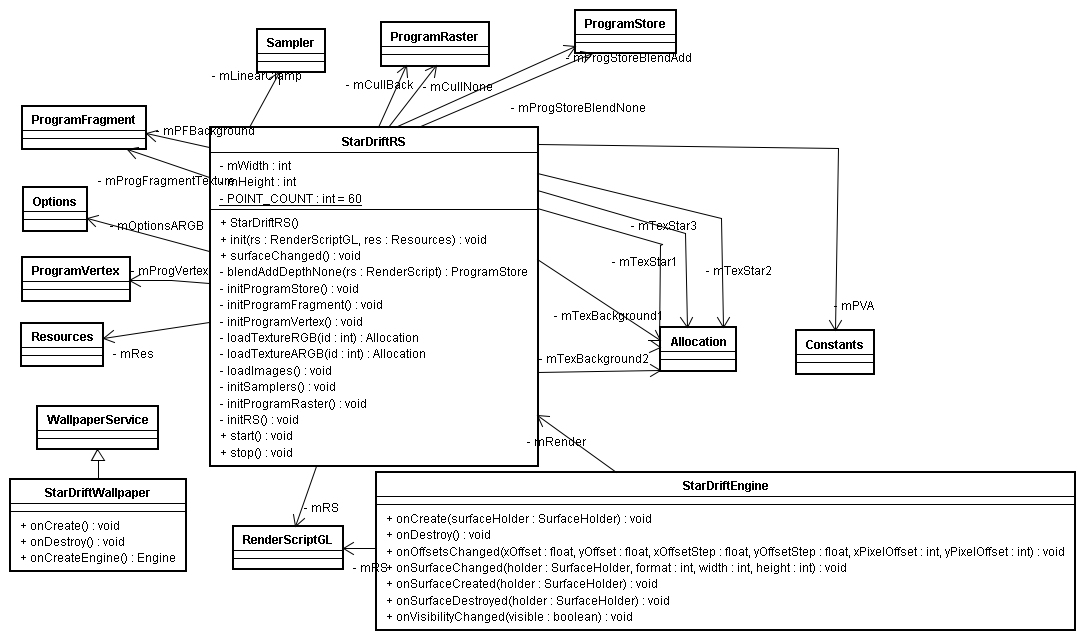


图3-2星光漂移类图

### 3.1.3 接口描述

1）功能部件

* 部件一：动态壁纸服务，构建一张动态壁纸必须继承实现系统定制的WallpapaerService，完成服务的部分函数实现，才可以被LivePicker动态壁纸加载器识别，进行正确加载。
* 部件二：动态壁纸服务引擎，构建动态壁纸必须继承实现动态壁纸引擎，完成引擎提供的画布具体实现。
* 部件三：RenderScript C代码句柄，创建RenderScript C代码对应的Java类，为Framework层调用提供接口。
* 部件四：RenderScript C代码，实现图片渲染功能。

2）内部接口设计

* 部件一与部件二：部件二为部件一的内部类，通过部件一的onCreateEngine（）方法返回部件二实例。
* 部件二与部件三：部件二中定义一类名为部件二的属性StarWallpaperRS mRender，通过部件二提供的公有方法完成在部件二中SurfaceView中的画图工作。
* 部件三与部件四：由系统工具为两个部件构建完整的接口。

3）外部接口设计

* 部件一外部接口：部件一必须实现系统提供的WallpaperService。
* 部件二外部接口：部件二必须实现系统提供的WallpaperService.Engine。

### 3.1.4 流程逻辑

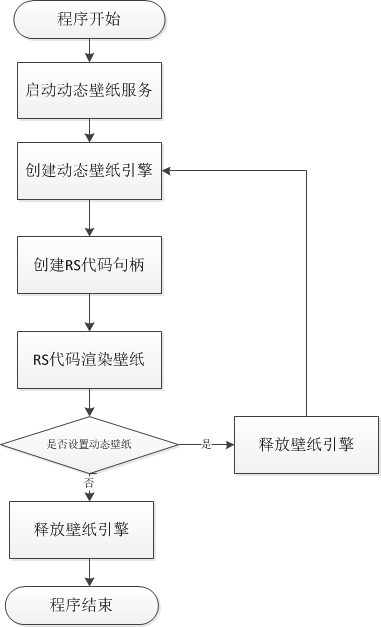


图3-3 星光漂移流程逻辑

## 3.2 实时时间天气

### 3.2.1 功能概述

符合桌面动态壁纸功能，可以通过Edit设置进行管理，在锁屏时正常显示，在调出Menu菜单和Edit设置时作为背景置灰显示，动态效果如图3-4所示。



图3-4 实时时间天气动态壁纸效果

增加随时间和天气而实时改变的效果，系统默认存储 “晴天”、“雨天”两种天气，以及“清晨”（05:00～08:00）、“上午”（09:00～12:00）、“中午”（13:00～17:00）、“傍晚”（18:00～23:00）、“凌晨”（00:00～04:00）五个时段，启动时默认显示当前时段晴天的动态效果，每隔1分钟更新时段效果，每隔3分钟自动更换天气效果，如图3-5、3-6所示。



图3-5 晴天清晨的实时时间天气动态壁纸效果



图3-6 雨天傍晚的实时时间天气动态壁纸效果

### 3.2.2 类图

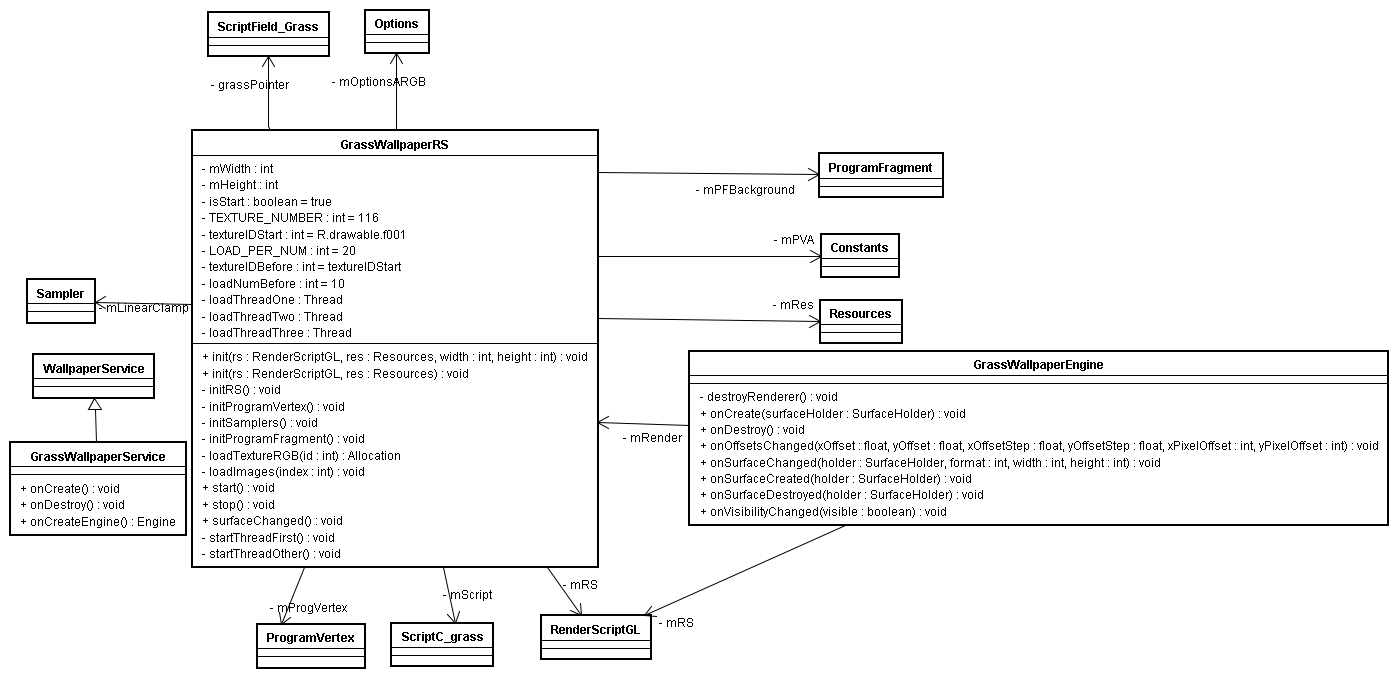


图3-7 实时时间天气类图

### 3.2.3 接口描述

1）功能部件

* 部件一：动态壁纸服务，构建一张动态壁纸必须继承实现系统定制的WallpaperService，完成服务的部分函数实现，才可以被LiveWallpaperPicker动态壁纸加载器识别，进行正确加载。
* 部件二：动态壁纸服务引擎，构建动态壁纸必须继承实现动态壁纸引擎，完成引擎提供的画布具体实现。
* 部件三：RenderScript C代码句柄，创建RenderScript C代码对应的Java类，为Framework层调用提供接口。
* 部件四：加载图片线程，创建线程，为RenderScript C渲染提供图片绑定。
* 部件五：RenderScript C代码，完成最终的图片渲染功能。
* 部件六：时段天气部件，定时完成时段和天气的循环变换。

2）内部接口设计

* 部件一与部件二：部件二为部件一的内部类，通过部件一的onCreateEngine（）方法返回部件二实例。
* 部件二与部件三：部件二中定义一类名为部件二的属性private StarWallpaperRS mRender，通过部件二提供的公有方法完成在部件二中SurfaceView中的画图工作。
* 部件三与部件四：部件四为部件三的内部类，在部件三中创建部件四的实例new LoadImageThread()。
* 部件三与部件五：由系统工具为两个部件构建完整的接口。
* 部件三与部件六：在部件三中创建部件六的实例WorldState = new WorldState()。
* 部件四与部件五：调用系统为部件三与部件五创建的接口，完成图片的加载工作。
* 部件五与部件六：调用系统为部件三与部件五创建的接口，完成天气和时段的变化工作。

3）外部接口设计

* 部件一外部接口：部件一必须实现系统提供的WallpaperService。
* 部件二外部接口：部件二必须实现系统提供的WallpaperService.Engine。

### 3.2.4 流程逻辑

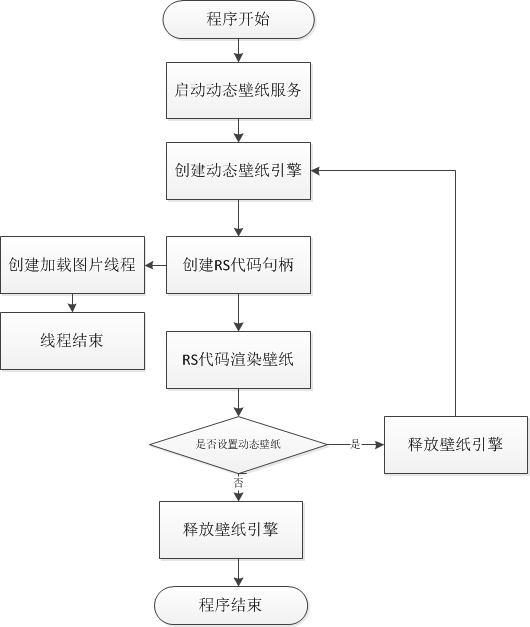


图3-8 实时时间天气流程逻辑

# 4桌面widget模块

## 4.1 三维时间天气

### 4.1.1 功能描述

实现桌面Widget的基本功能，可以被添加、删除、拖动等。

显示当前的时间、所在城市、天气信息。天气信息采用文字和动画效果结合的形式，根据不同的天气显示不同的动画效果。城市名称以文字形式显示在天气信息附近。当前时间采用三维立体数字形式显示，如图4-1所示。



图4-1 三维时间天气效果

系统默认存储“西安”、“北京”、“上海”三个城市、“晴天”、“雨天”两种天气，启动时默认显示当前时间西安晴天的信息。

当用户在Widget中上下滑动时，可以切换城市，切换时三维立体数字与滑动方向旋转一个角度后还原，依次切换三个城市，并更换天气信息。

### 4.1.2 类图

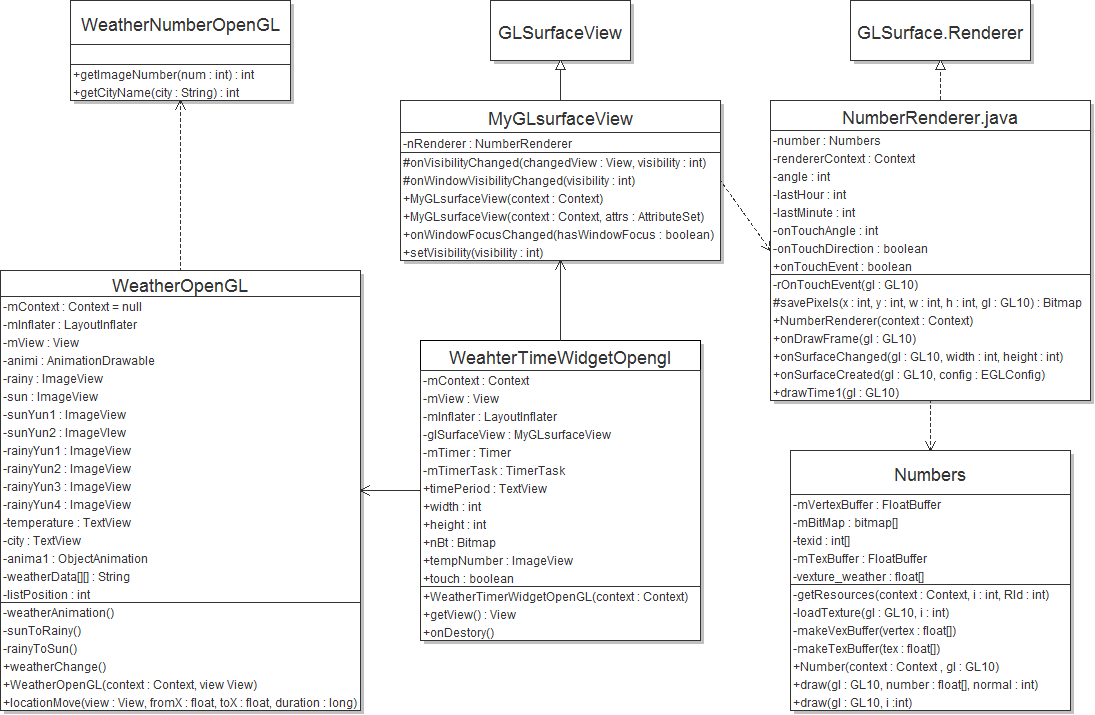


图4-2 三维时间天气类图

### 4.1.3 接口描述

MyGLsurfaceView：加载显示时间的Renderer，并对相应的事件进行监控和处理。

NumberRenderer：绘制三维时间，三维数字的渲染和动画控制。

Numbers：提供三维数字模型的建立。

WeatherNumberOpenGL：对于资源的重定位，通过统一的方法来获得资源的resourceID。

WeatherOpenGL：天气的显示以及动画效果的处理。

WeatherTimeWidgetOpenGL：提供getView()方法，将整个三维天气的View当做返回值。

### 4.1.4 流程逻辑

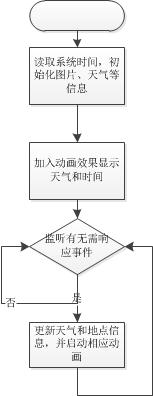


图4-3 三维时间天气流程逻辑

## 4.2 网络书签

### 4.2.1 功能描述

实现桌面Widget的基本功能，可以被添加、删除、拖动等。

以两排并列形式显示网络书签的缩略图，系统存储29个书签的缩略图，一屏长度可显示3张半缩略图，如图4-4所示。



图4-4 书签滑动查看效果

当用户在Widget中上下滑动时，所有书签缩略图滑动显示，并绕Y轴做一定角度的旋转，点击一张书签缩略图，会有震颤效果，且被点击的页面会从书签中飘出，并逐渐放大到整个屏幕，随后调用浏览器应用如图4-5所示。



图4-5 书签点击效果

### 4.2.2 类图

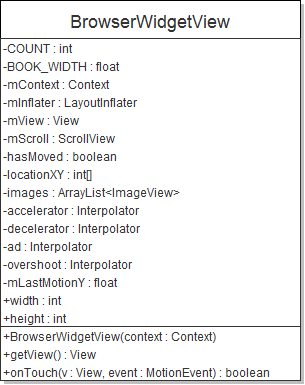


图4-6 网络书签类图

### 4.2.3 接口描述

BrowserWidgetView：对数据进行初始化，对于事件的监听以及动画效果的显示和控制，提供getView()方法，将网络书签的View当做返回值返回。

### 4.2.4 流程逻辑

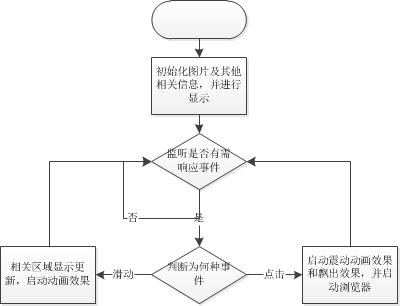


图4-7 网络书签流程逻辑

## 4.3 联系人

### 4.3.1 功能描述

实现桌面Widget的基本功能，可以被添加、删除、拖动等。

以两排并列形式显示联系人的缩略图，系统存储联系人的缩略图，一屏长度可显示2张半缩略图，如图4-8所示。



图4-8 联系人效果

当用户在Widget中上下滑动时，所有书签缩略图滑动显示，并绕Y轴做一定角度的旋转，如图4-9所示。



图4-9 联系人滑动效果

### 4.3.2 类图

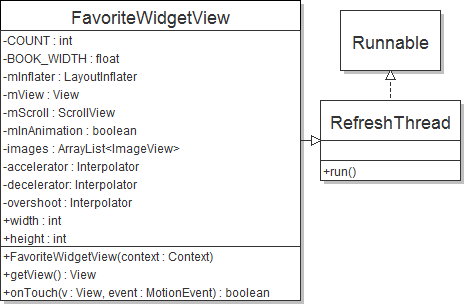


图4-10 联系人类图

### 4.3.3 接口描述

FavoriteWidgetView：对数据进行初始化，对于事件的监听以及动画效果的显示和控制，提供getView()方法，将联系人的View作为返回值返回。

### 4.3.4 流程逻辑

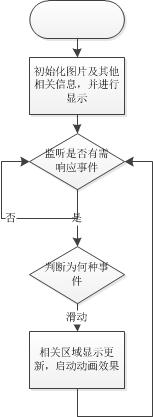


图4-11 联系人流程逻辑

## 4.4 社交网络

### 4.4.1 功能描述

显示好友的缩略页面，系统存储5个好友的缩略页面，当用户在Widget中向上滑动当前页面时，当前页以折纸方式向上退出，下个好友缩略页面由下往上滑入，向下滑动时页面向下平滑移动。

### 4.4.2 类图

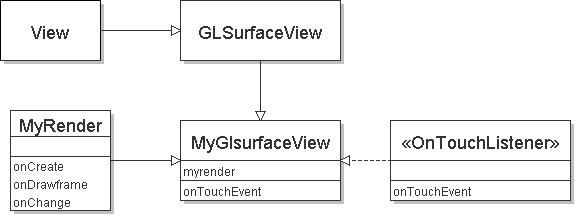


图4-12 社交网络类图

### 4.4.3 算法及模型描述

根据需求分析，结合Android Opengl ES的具体特点，需要使用动态建模来实现需求。

考虑到需求较为复杂，考虑以45\*135的二维数组来存储三维坐标点以此来构建模型，模型如下图所示：

|  |
| --- |
| 45\*45平面模型 |
| 45\*45 正弦模型 |
| 45\*45平面模型 |

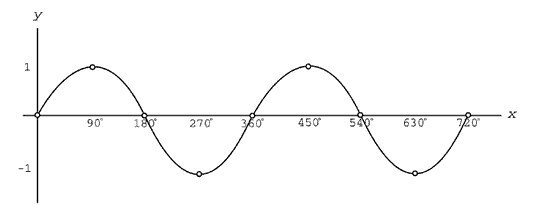


图4-13 模型示意图

该模型前后两个45\*45区域为平面模型且该模型在运动过程中不会变化，中间的45\*45区域使用正弦曲线模型（将y轴坐标与z轴坐标建立正弦关系），并通过数学计算将y轴坐标限定在正弦函数的第一象限中，以此来模拟折页效果。

### 4.4.4 流程逻辑

逻辑控制主要体现在GLSurfacevVew中OnTouchEvent事件与myrender类中创建的myCurvedSurface类（即建模的类）间的互动。具体逻辑流程如下图所示：

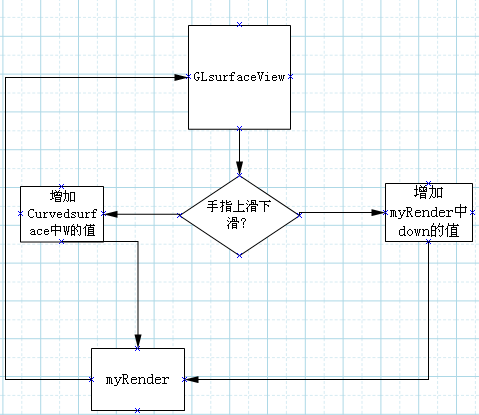


图4-14 社交网络流程逻辑图

## 4.5 微博

### 4.5.1 功能描述

实现桌面Widget的基本功能，可以被添加、删除、拖动等。

默认显示微博的界面，系统存储8条微博信息，一屏显示3条半微博信息，如图4-15所示。当用户在Widget中上下滑动时，所有微博信息滑动显示。



图4-15 微博默认界面

点击刷新按钮后为从上往下翻转滑入一条新微博。如图4-16所示。



图4-16 微博刷新动画效果

### 4.5.2 类图

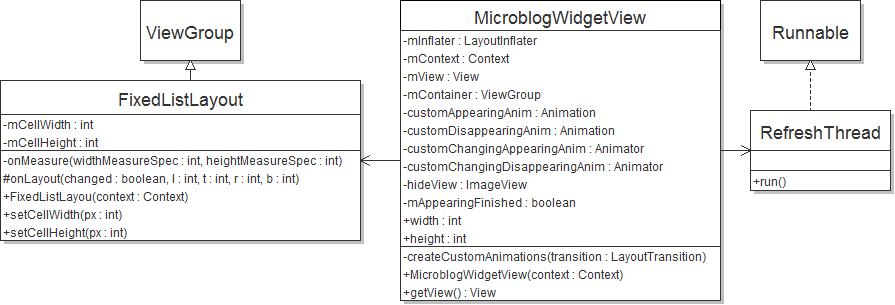


图4-17 微博类图

### 4.5.3 接口描述

MicroblogWidgetView：事件的监听以及动画效果的实现，提供getView()方法将微博的视图返回。

FixedListLayout：对于数据的初始化，以及视图布局的控制。

### 4.5.4 流程逻辑

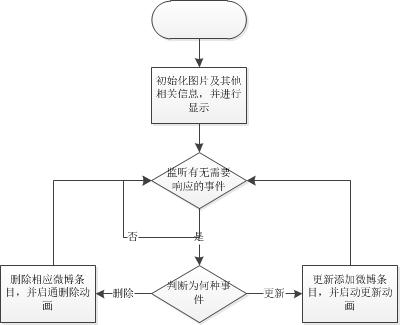


图4-18 微博流程逻辑

# 5 锁屏模块

## 5.1 功能概述

用户关闭屏幕时启动锁屏界面，再次开启屏幕时，显示锁屏界面。锁屏界面左侧显示时间、日期和电池状态。锁屏界面右侧为解锁按钮，长按按钮解锁。

## 5.2 类图

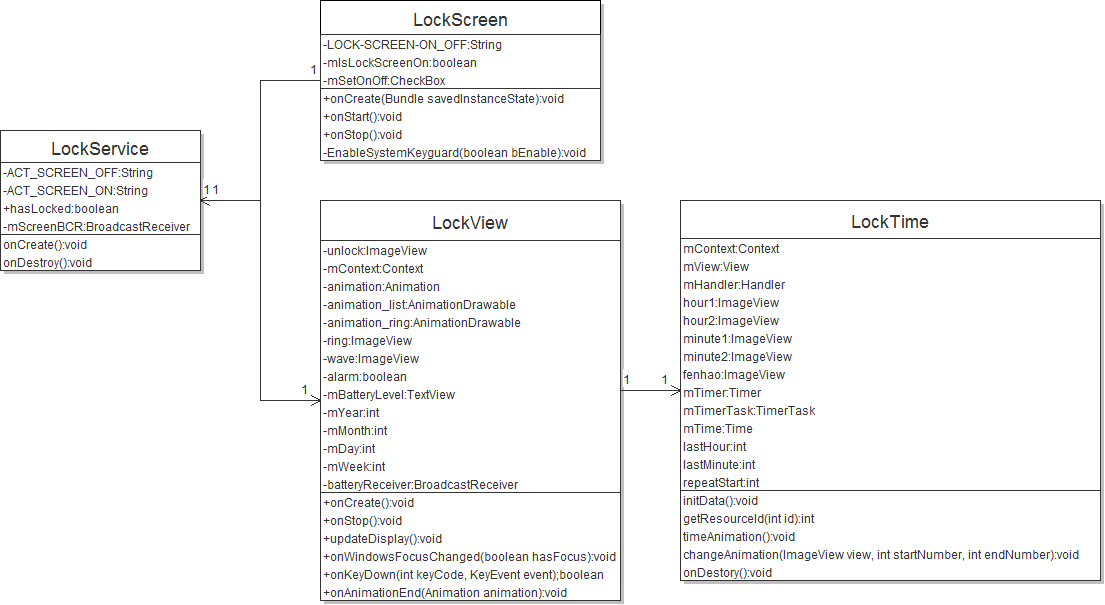


图5-1 锁屏类图

## 5.3 接口描述

LockScreen：主要负责实现开启/停止锁屏服务，禁用键盘；

LockService：锁屏服务，注册广播接收器，监控屏幕关闭操作，一旦屏幕关闭，启动锁屏界面；

LockView：锁屏界面，长按解锁；

LockTime：锁屏界面时的时间显示。

## 5.4 流程逻辑

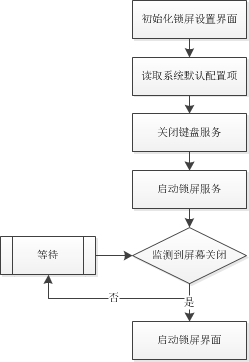


图5-2 锁屏流程

# 6 任务管理器模块

## 6.1 功能概述

任务管理器启动时，以弧形方式显示当前正在运行的程序的缩略图，从屏幕左边飞入，用户可上下滑动查看所有的程序缩略图，滑动时按照弧形方向转动，程序缩略图有前后震动效果，关闭程序时，上下两个临近程序缩略图会有上下震动效果。点击一个程序缩略图，被点击的缩略图会飘出，并逐渐放大到整个屏幕，然后激活该应用程序。

## 6.2 类图

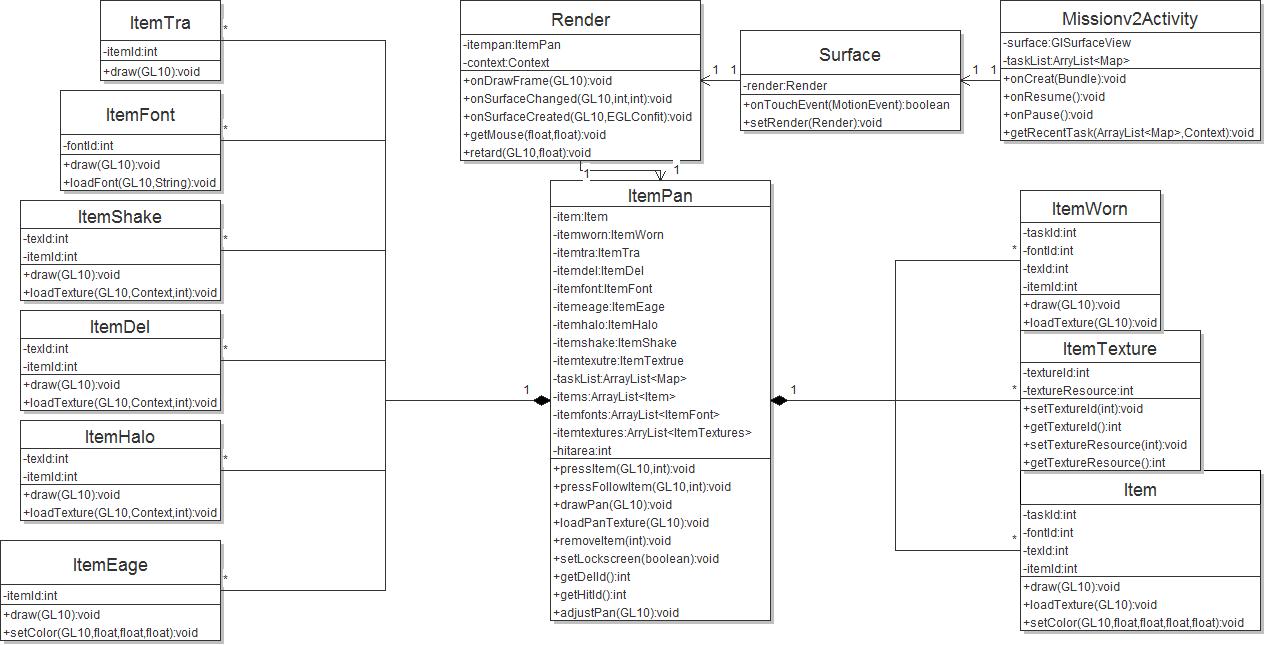


图6-1 任务管理器类图

## 6.3 接口描述

pressItem(GL10, int)：实现用户手指按下任务选项时的动画效果，包括弹性振动，高亮显示，颜色渐变，滑动淡出等动画效果。

pressFollowItem(GL10, int)：实现特定任务选项两侧选项的跟随振动效果。

getHitId()：返回用户点击选中的任务选项的选项标识。

getDelId()：返回用户点击删除的任务选项的选项标识。

drawPan(GL10)：以扇形的方式统一绘制所有任务选项，以及该扇形发生的动画效果，如删除任务选项后，两侧的选项合并的动画效果。

removeItem(int)：删除指定任务选项及其附带模型，更新各模型容器，关闭指定进程。

adjustPan(GL10)：调整扇形任务选项的位置，避免所有选项移出屏幕。

loadPanTexture(GL10)：统一装载各个模型需要的纹理、字符串等信息。

setLockscreen(boolean)：在完成一项操作的过程中锁定用户的操作。

getMouse(float, float)：返回鼠标在Android坐标系中的2维坐标。

retard(GL10, float)：实现扇形任务选项列表的减速和回转的动画效果。

getRecentTask(ArrayList<Map<String, Object>>, Context)：获得所有当前正在运行的进程，并过滤掉所有无法运行或无法停止的进程。

## 6.4 流程逻辑

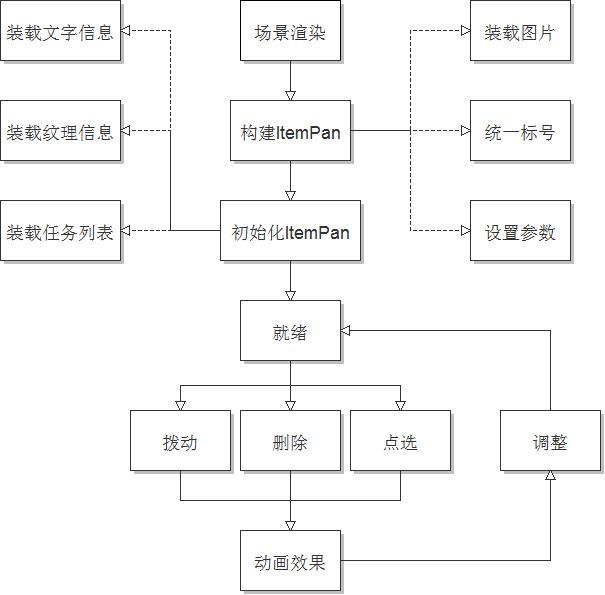


图6-2 任务管理器逻辑流程图