


七夜雪主 一点点幽默，一点点文艺，再加一点点技术，精心烹饪出来的会是什么呢？就是七夜雪了。。。

目录视图

摘要视图

RSS 订阅

个人资料



滕六

访问：266484次

积分：3488

等级：BLOG 5

排名：第4773名

原创：46篇

转载：23篇

译文：0篇

评论：349条

Android技术交流群

Android曙光集群：167630324

有兴趣的同学可以加一下

文章搜索

文章分类

推荐文章

最新评论

android源码定制之初探--定制anx
Fragmen: 楼主，怎么修改关机
界面的背景图片啊？

android源码定制之初探--定制anx
Fragmen: 楼主，怎么修改关机
界面的背景图片啊？

android应用开发之intent的妙用-
更年期般的小青年: 通俗易懂，
受益匪浅

android应用开发之intent的妙用-
更年期般的小青年: 写得这么
好，怎么没人顶了

android应用开发之intent的妙用-
UTuring8087: 写的很好，言简意
赅，谢谢(◕◕◕)...

Android-x86入门之--源代码的下
丁国华: 谢谢分享 学习了
'(*_*)'

如何让你的Android显示gif格式的
Tailyou: 1173752145@qq.com

android用户输入系统详细说明

写博客，送money、送书、送C币啦

7-8月博乐推荐文章

硬BUG 得大奖 100%中奖率

微信开发学习路线高级篇上线

恭喜博主

周兆熊新书发售

Android图形系统的分析与移植--一、Android GUI系统简介

分类：android图形系统分析与移植

2011-07-14 17:09

11607人阅读

评论(2)

收藏

举报

android

图形

module

框架

java

数据结构

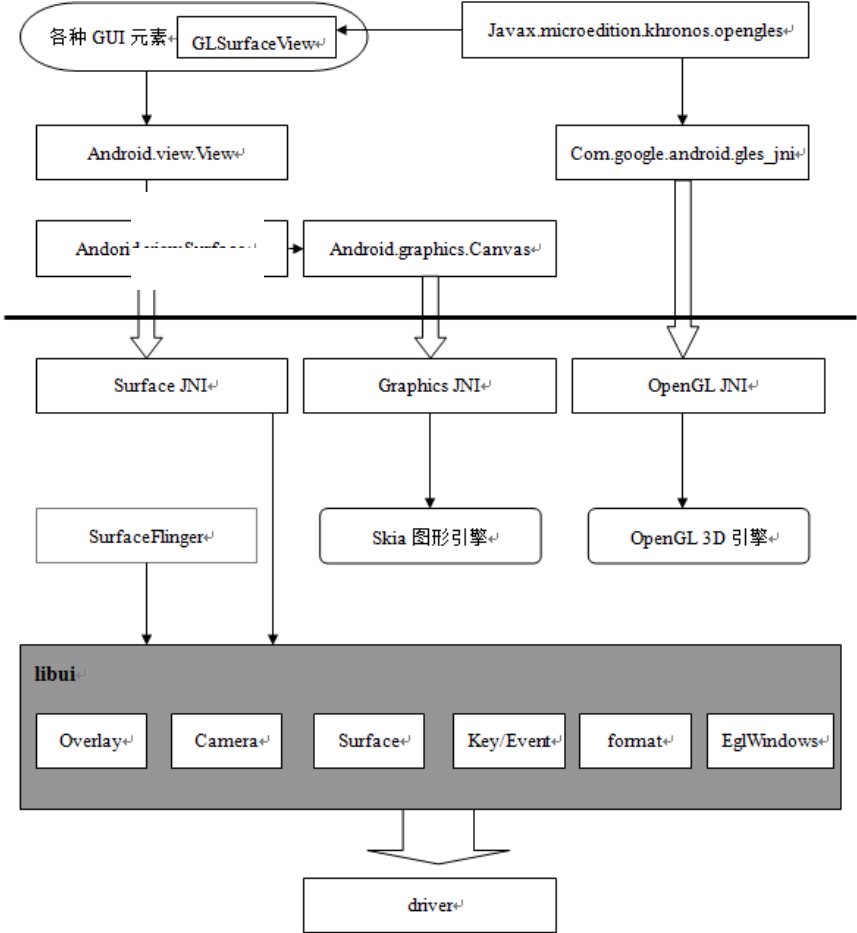
目录(?)

[+]

1 综述

Android的GUI系统由C语言框架和JAVA语言框架组成。对下层，Android的GUI系统通过调用显示输出设备和输入设备的驱动，从而将Android的软件系统和底层的硬件联系起来。对上层，Android的GUI系统提供了Java层次的绘图结构，Android的Java框架层调用这些绘图接口来构建各种UI元素，这些绘图接口也可以供Java应用程序层来调用。

Android的GUI系统架构图如图1所示：



```
graph TD
    subgraph Java_Layer [Java Layer]
        A[各种 GUI 元素<br/>GLSurfaceView<sup>+</sup>]
        B[Javax.microedition.khronos.opengles<sup>+</sup>]
    end
    subgraph Android_Framework [Android Framework]
        C[Android.view.View<sup>+</sup>]
        D[Android.graphics.Canvas<sup>+</sup>]
        E[Com.google.android.gles_jni<sup>+</sup>]
    end
    subgraph JNI_Layer [JNI Layer]
        F[Surface JNI<sup>+</sup>]
        G[Graphics JNI<sup>+</sup>]
        H[OpenGL JNI<sup>+</sup>]
    end
    subgraph Native_Layer [Native Layer]
        I[SurfaceFlinger<sup>+</sup>]
        J[Skia 图形引擎<sup>+</sup>]
        K[OpenGL 3D 引擎<sup>+</sup>]
    end
    subgraph LibUI [libui]
        L[Overlay<sup>+</sup>]
        M[Camera<sup>+</sup>]
        N[Surface<sup>+</sup>]
        O[Key/Event<sup>+</sup>]
        P[format<sup>+</sup>]
        Q[EglWindows<sup>+</sup>]
    end
    subgraph Driver [driver<sup>+</sup>]
    end

    A --> C
    B --> E
    C --> D
    D --> F
    D --> G
    E --> H
    F --> I
    G --> J
    H --> K
    I --> N
    J --> N
    K --> Q
    N --> Driver
```

图1 GUI系统框架图

其中驱动包括FrameBuffer驱动，Event输入驱动等。

liangbmw: 分析的太到位了，后面的没连载啊，，期待发两篇寓言共勉下--献给那些初入y_xing: 很有意义。

android应用开发之intent的妙用-jd_chows: 计算机源自我们的生活，博主 看了你的文章收获颇多谢谢了

文章存档

2011年12月	(2)
2011年11月	(1)
2011年09月	(1)
2011年08月	(25)
2011年07月	(40)

阅读排行

Android-x86入门之--让你如何让你的Android显示c	(17060)
Android-x86入门之--启动	(15577)
android应用开发之intent	(14051)
Android图形系统的分析!	(12997)
Android-x86入门之--源代码	(11606)
Android-x86入门之--各类	(10134)
Android-x86入门之--固化	(8000)
android应用程序开发另角	(7585)
Android图形系统分析与利	(7378)
	(7027)

评论排行

android应用程序开发另角	(45)
如何让你的Android显示c	(44)
Android-x86入门之--如何	(24)
Android-x86入门之--源代码	(23)
Android-x86入门之--让你	(23)
android源码定制之初探--	(18)
android应用开发之intent	(16)
Android-x86入门之--各类	(14)
android学习要点总结	(11)
Android技术交流群	(11)

由上图可知，libui是提供接口的框架库，其他的库大部分通过继承并实现libui中定义的类来实现相应的功能。因此，理解libui的构造对于了解GUI的整体框架存在必然性。以下就来介绍一下该库。

2 libui库

libui是Android在本地层次一个框架性质的集成库，它不仅是现实的中枢，也是整个AndroidGUI系统的中枢。

这个库提供了一些接口，由其他的库通过类继承方式来实现，而调用者只需要调用libui库的接口即可。libui库包含了颜色格式，Egl窗口（用户显示），按键及事件处理，Surface（显示界面），Overlay（显示叠加层接口），Camera（照相机接口）等多个方面的定义。

libui库中包含的内容如下：

1) Format（颜色格式）部分

这个部分本身定义颜色空间的枚举类型和数据结构，它需要重用pixelflinger中的一些数据格式的定义。

2) Native Windows（本地窗口）部分

本部分用于实现一个本地显示的接口，也就是实现一个egl_native_window_t类，程序通过调用从而实现基本的显示功能。

3) Key/Event（按键和时间处理）部分

本部分是系统输入的基础，其中定义按键的映射，通过操作Event时间设备来实现获取系统的输入。

4) Surface（显示界面）部分

本部分定义了显示界面较高层次的接口，包含部分显示界面的管理功能。

5) Overlay（显示部分的叠加层）部分

定义了一个叠加的显示输出层接口，覆盖在主显示层之上，常用于视频输出。

6) Camera（照相机）部分

定义摄像头的框架和接口。

3 显示输出与硬件的接口

显示输出接口最终需要与硬件交互，对于Linux操作系统，就是要和驱动程序打交道。这两部分原本也是属于“硬件抽象层”的内容，由于Android中使用的是FrameBuffer的标准显示驱动，因此本部分内容在libui库中使用标准方式来实现。

对于Android的显示部分，需要实现的接口是egl_native_window_t，它是一个OpenGL结构，也是给libEGL使用的。

Android使用标准的FrameBuffer作为驱动程序，同时也在本地框架中提供了系统和FrameBuffer驱动程序之间的适配层，即硬件抽象层。这个部分主要在Gralloc模块中实现调用标准的FrameBuffer驱动，它是系统和FrameBuffer设备之间的接口，以硬件模块的形式存在。Gralloc被libui库调用。

Gralloc模式是显示模块的实现，其中，framebuffer.cpp用于基于FrameBuffer的显示实现。其中定义了帧缓冲区为双缓冲：

```
// numbers of buffers for page flipping

#define NUM_BUFFERS 2

.....

//使用FBIOGET_FSCREENINFO和FBIOGET_VSCREENINFO的ioctl()接口来获取驱动程序的基本信息，并调用mmap()将驱动程序的内存映射到用户空间

struct fb_fix_screeninfo finfo;

if (ioctl(fd, FBIOGET_FSCREENINFO, &finfo) == -1)

return -errno;
```

```
structfb_var_screeninfo info;

    if (ioctl(fd, FBIOGET_VSCREENINFO,&info) == -1)

        return -errno;

//RequestNUM_BUFFERS screens (at lest 2 for page flipping)

info.yres_virtual= info.yres * NUM_BUFFERS;

libui对gralloc模块实现了调用，在FramebufferNativeWindow.cpp中打开了gralloc设备。

FramebufferNativeWindow::FramebufferNativeWindow()

    :BASE(),fbDev(0),grDev(0),mUpdateOnDemand(false)

{

    hw_module_tconst* module;

    if(hw_get_module(GRALLOC_HARDWARE_MODULE_ID, &module) == 0) {

        int stride;

        int err;

        err = framebuffer_open(module,&fbDev);

        LOGE_IF(err,"couldn't open framebuffer HAL (%s)", strerror(-err));

        err =gralloc_open(module, &grDev);

        LOGE_IF(err,"couldn't open gralloc HAL (%s)", strerror(-err));

        // .....

    }

}
```

版权声明：本文为博主原创文章，未经博主允许不得转载。

下一篇 Android图形系统的分析与移植--二、Android显示系统软件架构分析

主题推荐 android 移植 图形 gui

猜你在找

- Java基础核心技术：泛型(day14)
- Java基础核心技术：Java语言概述 (day01)
- Java基础核心技术：高级类特性1(day07-day09)
- Java基础核心技术：网络编程(day20)
- Java基础核心技术：面向对象编程(day05-day07)



查看评论

2楼 [lugechao](#) 2013-05-25 21:27发表

 受益匪浅啊！

1楼 [dish_bird123](#) 2011-08-01 15:04发表

good