



(12) 发明专利申请

(10) 申请公布号 CN 104572086 A

(43) 申请公布日 2015. 04. 29

(21) 申请号 201410812986. 2

(22) 申请日 2014. 12. 24

(71) 申请人 惠州 TCL 移动通信有限公司

地址 516006 广东省惠州市仲恺高新区和畅
七路西 86 号

(72) 发明人 李朝朝 李明江 曹捷

(74) 专利代理机构 深圳市君胜知识产权代理事
务所 44268

代理人 王永文 刘文求

(51) Int. Cl.

G06F 9/44(2006. 01)

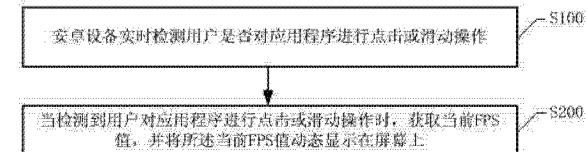
权利要求书2页 说明书8页 附图2页

(54) 发明名称

基于安卓设备的 FPS 值获取方法及获取系统

(57) 摘要

本发明公开了一种基于安卓设备的 FPS 值获
取方法及获取系统，所述方法包括：安卓设备实
时检测用户是否对应用程序进行点击或滑动操
作；当检测到用户对应用程序进行点击或滑动操
作时，获取当前 FPS 值，并将所述当前 FPS 值动态
显示在屏幕上。采用本发明可避免对安卓系统的
修改，更精确的获得了评价安卓设备的流畅性的
FPS 值，同时将 FPS 值显示在屏幕上，更方便用户
观看。



1. 一种基于安卓设备的 FPS 值获取方法, 其特征在于, 包括 :

A、安卓设备实时检测用户是否对应用程序进行点击或滑动操作 ;

B、当检测到用户对应用程序进行点击或滑动操作时, 获取当前 FPS 值, 并将所述当前 FPS 值动态显示在屏幕上。

2. 根据权利要求 1 所述的基于安卓设备的 FPS 值获取方法, 其特征在于, 所述步骤 B 具体包括 :

B1、建立动态链接库文件, 设置一交换缓冲函数, 判断是否进行显示初始化 ;

B2、如果进行显示初始化, 则将系统中的交换缓冲函数地址替换为所述自定义的函数地址, 检测触摸感应驱动是否开始刷新屏幕 ;

B3、如果是则调用自定义的交换缓冲函数, 对函数调用进行计数, 计算 FPS 值 ;

B4、将计算得到的 FPS 值写入用于数据通信的 Pipe 文件, 建立服务读取 Pipe 文件, 将得到 FPS 值显示到屏幕上。

3. 根据权利要求 2 所述的基于安卓设备的 FPS 值获取方法, 其特征在于, 所述步骤 B4 具体包括 :

B41、监听 Pipe 服务的启动, 进行 Pipe 的创建, 创建悬浮文字, 并显示悬浮文字, 执行注入的命令 ;

B42、在监听服务启动后, 启动线程, 线程每隔一预定时间可从 Pipe 读取触摸感应驱动传来的 FPS 值, 并显示在屏幕上。

4. 根据权利要求 2 所述的基于安卓设备的 FPS 值获取方法, 其特征在于, 所述步骤 B3 中 FPS 的计算方法为 :

通过计算交换缓冲函数在一秒钟执行的次数或是通过单位时间与单次执行函数所有时间的比值。

5. 根据权利要求 4 所述的基于安卓设备的 FPS 值获取方法, 其特征在于, 所述预定时间为 1s。

6. 一种基于安卓设备的 FPS 值获取系统, 其特征在于, 包括 :

检测模块, 用于安卓设备实时检测用户是否对应用程序进行点击或滑动操作 ;

获取与显示模块, 用于当检测到用户对应用程序进行点击或滑动操作时, 获取当前 FPS 值, 并将所述当前 FPS 值动态显示在屏幕上。

7. 根据权利要求 6 所述的基于安卓设备的 FPS 值获取系统, 其特征在于, 所述获取与显示模块具体包括 :

建立与设置单元, 用于建立动态链接库文件, 设置一交换缓冲函数, 判断是否进行显示初始化 ;

替换与检测单元, 用于如果进行显示初始化, 则将系统中的交换缓冲函数地址替换为所述自定义的函数地址, 检测触摸感应驱动是否开始刷新屏幕 ;

调用与计算单元, 用于如果是则调用自定义的交换缓冲函数, 对函数调用进行计数, 计算 FPS 值 ;

写入与显示单元, 用于将计算得到的 FPS 值写入用于数据通信的 Pipe 文件, 建立服务读取 Pipe 文件, 将得到 FPS 值显示到屏幕上。

8. 根据权利要求 7 所述的基于安卓设备的 FPS 值获取系统, 其特征在于, 所述步骤写入

与显示单元具体包括：

监听与创建单元，用于监听 Pipe 服务的启动，进行 Pipe 的创建，创建悬浮文字，并显示悬浮文字，执行注入的命令；

启动与显示单元，用于在监听服务启动后，启动线程，线程每隔一预定时间可从 Pipe 读取触摸感应驱动传来的 FPS 值，并显示在屏幕上。

9. 根据权利要求 7 所述的基于安卓设备的 FPS 值获取系统，其特征在于，所述调用与计算单元中 FPS 的计算方法为：

通过计算交换缓冲函数在一秒钟执行的次数或是通过单位时间与单次执行函数所有时间的比值。

10. 根据权利要求 9 所述的基于安卓设备的 FPS 值获取系统，其特征在于，所述预定时间为 1s。

基于安卓设备的 FPS 值获取方法及获取系统

技术领域

[0001] 本发明涉及智能设备领域，尤其涉及一种基于安卓设备的 FPS 值获取方法及获取系统。

背景技术

[0002] 随着移动通信的发展和人们生活水平的不断提高，各种安卓设备如手机、PAD 等设备的使用越来越普及，安卓设备已经成为人们生活中不可缺少的通信工具。

[0003] 目前，在移动产品市场，消费者对移动设备的性能关注度越来越高，而流畅性是影响用户对设备性能评价最直接和重要的因素之一。因此，设备的流畅性是用户关注的关键点，其评测、分析、优化方法更是各设备厂商重点研究的内容。而 FPS (Frame Per Second, 帧率，每秒帧的刷新频率) 是图形处理器每秒刷新次数，是评价安卓设备流畅性评测的重要参数，但是现有技术中提供的 FPS 值的计算不够准确，实时性也不够好，用户无法直观的看到结果。

[0004] 因此，现有技术还有待于改进和发展。

发明内容

[0005] 本发明要解决的技术问题在于，针对现有技术的上述缺陷，提供了一种基于安卓设备的 FPS 值获取方法及获取系统，通过本发明所述方案用户可以得到更精确的安卓设备 FPS 的值，从而直观的了解自己的安卓设备的流畅性。

[0006] 本发明解决技术问题所采用的技术方案如下：

一种基于安卓设备的 FPS 值获取方法，其中，包括：

A、安卓设备实时检测用户是否对应用程序进行点击或滑动操作；

B、当检测到用户对应用程序进行点击或滑动操作时，获取当前 FPS 值，并将所述当前 FPS 值动态显示在屏幕上。

[0007] 根据权利要求 1 所述的基于安卓设备的 FPS 值获取方法，其中，所述步骤 B 具体包括：

B1、建立动态链接库文件，设置一交换缓冲函数，判断是否进行显示初始化；

B2、如果进行显示初始化，则将系统中的交换缓冲函数地址替换为所述自定义的函数地址，检测触摸感应驱动是否开始刷新屏幕；

B3、如果是则调用自定义的交换缓冲函数，对函数调用进行计数，计算 FPS 值；

B4、将计算得到的 FPS 值写入用于数据通信的 Pipe 文件，建立服务读取 Pipe 文件，将得到 FPS 值显示到屏幕上。

[0008] 所述的基于安卓设备的 FPS 值获取方法，其中，所述步骤 B4 具体包括：

B41、监听 Pipe 服务的启动，进行 Pipe 的创建，创建悬浮文字，并显示悬浮文字，执行注入的命令；

B42、在监听服务启动后，启动线程，线程每隔一预定时间可从 Pipe 读取触摸感应驱动

传来的 FPS 值，并显示在屏幕上。

[0009] 所述的基于安卓设备的 FPS 值获取方法，其中，所述步骤 B3 中 FPS 的计算方法为：

通过计算交换缓冲函数在一秒钟执行的次数或是通过单位时间与单次执行函数所有时间的比值。

[0010] 所述的基于安卓设备的 FPS 值获取方法，其中，所述预定时间为 1s。

[0011] 一种基于安卓设备的 FPS 值获取系统，其中，包括：

检测模块，用于安卓设备实时检测用户是否对应用程序进行点击或滑动操作；

获取与显示模块，用于当检测到用户对应用程序进行点击或滑动操作时，获取当前 FPS 值，并将所述当前 FPS 值动态显示在屏幕上。

[0012] 所述的基于安卓设备的 FPS 值获取系统，其中，所述获取与显示模块具体包括：

建立与设置单元，用于建立动态链接库文件，设置一交换缓冲函数，判断是否进行显示初始化；

替换与检测单元，用于如果进行显示初始化，则将系统中的交换缓冲函数地址替换为所述自定义的函数地址，检测触摸感应驱动是否开始刷新屏幕；

调用与计算单元，用于如果是则调用自定义的交换缓冲函数，对函数调用进行计数，计算 FPS 值；

写入与显示单元，用于将计算得到的 FPS 值写入用于数据通信的 Pipe 文件，建立服务读取 Pipe 文件，将得到 FPS 值显示到屏幕上。

[0013] 所述的基于安卓设备的 FPS 值获取系统，其中，所述步骤写入与显示单元具体包括：

监听与创建单元，用于监听 Pipe 服务的启动，进行 Pipe 的创建，创建悬浮文字，并显示悬浮文字，执行注入的命令；

启动与显示单元，用于在监听服务启动后，启动线程，线程每隔一预定时间可从 Pipe 读取触摸感应驱动传来的 FPS 值，并显示在屏幕上。

[0014] 所述的基于安卓设备的 FPS 值获取系统，其中，所述调用与计算单元中 FPS 的计算方法为：

通过计算交换缓冲函数在一秒钟执行的次数或是通过单位时间与单次执行函数所有时间的比值。

[0015] 所述的基于安卓设备的 FPS 值获取系统，其中，所述预定时间为 1s。本发明提供了一种基于安卓设备的 FPS 值获取方法及获取系统，所述方法包括：安卓设备实时检测用户是否对应用程序进行点击或滑动操作；当检测到用户对应用程序进行点击或滑动操作时，获取当前 FPS 值，并将所述当前 FPS 值动态显示在屏幕上。本发明使移动终端增加了新功能：FPS 值动态显示功能；采用本发明可避免对安卓系统的修改，更精确的获得了评价安卓设备的流畅性的 FPS 值，同时将 FPS 值显示在屏幕上，更方便用户观看。

附图说明

[0016] 图 1 是本发明基于安卓设备的 FPS 值获取方法的较佳实施例的流程图。

[0017] 图 2 是本发明基于安卓设备的 FPS 值获取方法的具体应用实施例中 FPS 值动态显示过程的流程图。

[0018] 图 3 是本发明基于安卓设备的 FPS 值获取系统的较佳实施例的功能原理框图。

具体实施方式

[0019] 为使本发明的目的、技术方案及优点更加清楚、明确,以下参照附图并举实施例对本发明进一步详细说明。应当理解,此处所描述的具体实施例仅仅用以解释本发明,并不用于限定本发明。

[0020] 请参见图 1,图 1 是本发明基于安卓设备的 FPS 值获取方法的较佳实施例的流程图。方法包括:

步骤 S100、安卓设备实时检测用户是否对应用程序进行点击或滑动操作。所述安卓设备可为使用安卓系统的手机、PAD 等。较优的,所述安卓设备为用触屏操作的设备。因为只有用户在对应用程序进行操作时,系统才会对屏幕进行刷新,尤其是在玩游戏和看视频的时候,对系统的 FPS 值要求较高。

[0021] 步骤 S200、当检测到用户对应用程序进行点击或滑动操作时,获取当前 FPS 值,并将所述当前 FPS 值动态显示在屏幕上。

[0022] 其中, FPS(Frame Per Second, 帧率, 每秒帧的刷新频率)是图形处理器每秒刷新次数,是评价安卓设备流畅性评测的重要参数。

[0023] 具体实施时,步骤 S200 还包括:

步骤 S210、建立动态链接库文件,设置一交换缓冲函数,判断是否进行显示初始化;具体的所述动态链接库文件是后缀为 so 的文件,也称 so 文件。交换缓冲函数即为eglSwapBuffers 函数。

[0024] 步骤 S220、如果进行显示初始化,则将系统中的交换缓冲函数地址替换为所述自定义的函数地址,检测触摸感应驱动是否开始刷新屏幕;具体的,所述触摸感应驱动是 SurfaceFlinger 的中文译文。

[0025] 步骤 S230、如果是则调用自定义的交换缓冲函数,对函数调用进行计数,计算 FPS 值;

步骤 S240、将计算得到的 FPS 值写入用于数据通信的 Pipe 文件,建立服务读取 Pipe 文件,将得到 FPS 值显示到屏幕上。

[0026] 具体实施时,所述步骤 S240 具体包括:

步骤 S241、监听 Pipe 服务的启动,进行 Pipe 的创建,创建悬浮文字,并显示悬浮文字,执行注入的命令;

步骤 S242、在监听服务启动后,启动线程,线程每隔一预定时间可从 Pipe 读取触摸感应驱动传来的 FPS 值,并显示在屏幕上。

[0027] 具体实施时,所述步骤 S230 中 FPS 的计算方法为:

通过计算交换缓冲函数在一秒钟执行的次数或是通过单位时间与单次执行函数所有时间的比值。

[0028] 线程读取 Pipe 的值中,所述预定时间为 1s。具体的,线程每隔 1 秒通过通信管道 Pipe 读取触摸感应驱动传来的 FPS 的值,间隔时间也可根据需要确定,比如 2 秒等。

[0029] 由以上方法实施例可知,本发明提供的基于安卓设备的 FPS 获取方法,可避免对安卓系统的修改,更精确的获得了评价安卓设备的流畅性的 FPS 值,同时将 FPS 值显示在屏

幕上,更方便用户观看。

[0030] 本发明还提供了一种本发明基于安卓设备的 FPS 值获取方法的具体应用实施例中 FPS 值动态显示过程的流程图,如图 2 所示,步骤包括:

步骤 S10、建立 so 文件,之后执行步骤 S20;

步骤 S20、自定义eglSwapBuffers 函数,之后执行步骤 S30;

步骤 S30、判断是否进行显示初始化,如果否,则执行步骤 S30,如果是;如果否,则执行步骤 S40;

步骤 S40、将系统中的 eglSwapBuffers 函数地址替换为自定义的函数地址,之后执行步骤 S50;

步骤 S50、判断 SurfaceFlinger 是否开始刷新屏幕,如果不,则执行步骤 S50,如果是;如果不,则执行步骤 S60;

步骤 S60、调用自定义的 eglSwapBuffers 函数,对函数调用进行计数,计算 FPS 值;之后执行步骤 S70;

步骤 S70、将计算得到的 FPS 值写入 Pipe 文件,之后执行步骤 S80;

步骤 S80、建立服务读取 Pipe 文件,之后执行步骤 S90;

步骤 S90、将得到 FPS 值显示到屏幕上,之后执行步骤 S100。

[0031] 具体实施时,Pipe 服务的具体包括以下几个步骤:

1) 监听 Pipe 服务的启动,即监听服务启动、服务结束按钮,如果点击了启动按钮,则开启监听 Pipe 服务;如果点击了停止按钮,则停止监听 Pipe 的服务。具体的实现代码如下:

```
case R.id.buttonStart:  
    Log.d(TAG, "starting service"); // 监听服务启动按钮 //  
    startService(new Intent(this, FPSService.class)); // 如果点击了启动  
    按钮,则开启监听 Pipe 服务 //  
    break;  
case R.id.buttonStop:  
    Log.d(TAG, "stopping service"); // 监听服务结束按钮 //  
    stopService(new Intent(this, FPSService.class));  
    break; // 如果点击了停止按钮,则停止监听 Pipe 的服务 //。
```

[0032] 2) 服务的初始化。这里进行 Pipe 的创建,悬浮文字的创建,注入的执行等工作。具体实施时,首先建立 Pipe 文件,建立显示悬浮文字的布局,将更改文件系统权限值的命令,统一添加到一个字符串数组中。然后执行命令,给应用加 root 权限。如果命令执行不成功,则给用户提示请确保手机已经被 root。实现代码如下:

```
// Create the Pipe file for receiving data  
createPipe(); // 建立 Pipe 文件 //  
// Create floating textview to display FPS  
createLayout(); // 建立显示悬浮文字的布局 //  
// Inject and hook  
ArrayList<String> list = new ArrayList<String>();  
list.add("chmod 775 " + APP_PATH + "inject");
```

```

list.add("chmod 666 " + APP_PATH + "Pipe");
list.add("chmod 775 " + APP_PATH + "libfpsshow.so");
list.add(APP_PATH + "inject"); // 将更改文件系统权限值的命令,统一添加
到一个字符串数组中 //
// Execute as root
if (execute(list)) {
    Log.e(TAG, "OK\n");
} else {
    Toast.makeText(this, "Execute abnormally, Can not get root
permission.", Toast.LENGTH_LONG).show();
    Log.e(TAG, "Error\n");
    this.stopSelf();
} // 执行命令,给应用加 root 权限。如果命令执行不成功,则给用户提示请确保
手机应用不能获得 root 权限 //。

```

[0033] 3) 对于注入的命令的执行,需要 root 权限,方法如下:建立一个系统进程类 Process 对象 suProcess,建立一个输出流 os 用来获取 suProcess 的输出。然后从字符串数组中依次读取更改文件权限值的命令,并依次执行。最后执行退出命令。代码实现如下:

```

public final boolean execute(ArrayList<String> commands) {
    Process suProcess = Runtime.getRuntime().exec("su"); // 建立一个系
统进程类 Process 对象 suProcess//
    DataOutputStream os = new DataOutputStream(suProcess.
getOutputStream()); // 建立一个输出流 os 用来获取 suProcess 的输出 //
    for (String currCommand : commands) {
        os.writeBytes(currCommand + "\n");
        os.flush();
    } // 然后从字符串数组中依次读取更改文件权限值的命令,并依次执行 //
    os.writeBytes("exit\n");
    os.flush();
    int suProcessRetval = suProcess.waitFor();
} // 执行退出命令 //。

```

[0034] 4) 悬浮文字的显示。这里通过一个 Layout 即可。具体的,通过设置悬浮窗口,将其设置为系统提醒模式,不可获取焦点,不可触摸,位置在屏幕顶端中间,大小随内容而定。代码实现如下:

```

windowManager = (WindowManager) getSystemService("window");
layoutParams = new WindowManager.LayoutParams();
layoutParams.type = WindowManager.LayoutParams.TYPE_SYSTEM_ALERT;
layoutParams.flags = WindowManager.LayoutParams.FLAG_NOT_FOCUSABLE |
WindowManager.LayoutParams.FLAG_NOT_TOUCH_MODAL;

```

```

layoutParams.format = PixelFormat.RGBA_8888;
layoutParams.gravity = Gravity.TOP | Gravity.CENTER;
layoutParams.width = WindowManager.LayoutParams.WRAP_CONTENT;
layoutParams.height = WindowManager.LayoutParams.WRAP_CONTENT;
layoutParams.x = 0;
layoutParams.y = 0;
// myLayout is the customized layout which contains textView
myLayout = new MyLayout(this);
windowManager.addView(myLayout, layoutParams); //。

```

[0035] 5) 在监听服务的 onStartCommand() 被调用时,会启动线程,线程每一秒执行一次 run() 函数,该函数从 Pipe 读 Surfaceflinger 传来的 FPS,然后显示在屏幕上。这里不一定必须是每一秒 run 一次,可以根据需要调整。具体实施时,在线程的 run 函数中,通过 readFps() 方法读取 fps 值,然后将值传递到悬浮窗的对象 myLayout 中去,设置间隔执行时间为 1 秒。代码实现如下:

```

private Runnable myTasks = new Runnable() {
    @Override
    public void run() {
        int fps = readFps();
        Log.e(TAG, "Service FPS = " + fps + "\n");
        myLayout.setFPS(fps);
        // Do other customized computation.
        myLayout.setFPSAvg(fps_avg);
        myhandler.postDelayed(myTasks, 1000);
    }
}。

```

[0036] 6) FPS 的计算方法。eglSwapBuffers 函数每执行一次,相当于显示一次。在 1 秒钟内的次数,即 FPS 值。即获取系统当前时间,赋值给 Time1;执行缓存交换函数;重新获取系统当前时间,赋值给 Time2;交换时间 swapTime 等于两时间差;fps 等于 1 秒除以交换时间。具体代码如下方法一所示,也可以通过单位时间与单次执行函数所用时间的比值计算得到。通过当前时间 currentTime、上次时间 previousTime、开始时间 startTime 三个时间点来确立 framesNum,来计算 fps。currentTime 是系统当前时间,previousTime 是每次缓存交换函数开始执行的时间,startTime 是每次计算 fps 的起始时间,framesNum 是显示的帧数。缓存交换函数每执行一次,framesNum 增加 1,当前时间 currentTime 减去开始时间 startTime,即总共花费时间 totalTime。当 totalTime 大于 1000 毫秒,输出一次 fps。fps 的值等于总共的帧数除以所用总共时间。然后将当前时间重新设为开始时间 startTime,将 framesNum 重新设为 0。再进行下一次新的计算。代码实现如下所示:

方法一:

```

time1 = getSystemCurrentTime(); // 获取系统当前时间,赋值给 Time1 //
start_original_eglSwapBuffers(); // 执行缓存交换函数 //

```

```
time2 = getSystemCurrentTime(); // 重新获取系统当前时间, 赋值给 Time2 //
swapTime = t2 - t1; // 交换时间 swapTime 等于两时间差 //
fps = unitTime/swapTime; //fps 等于 1 秒除以交换时间 //
方法二：
currentTime = getSystemCurrentTime();
if (startTime < 0) {
    startTime = previousTime = currentTime;
    framesNum = 0;
} // 通过当前时间 currentTime、上次时间 previousTime、开始时间 startTime 三个时间点来确立 framesNum, 来计算 fps。 currentTime 是系统当前时间, previousTime 是每次缓存交换函数开始执行的时间, startTime 是每次计算 fps 的起始时间, framesNum 是显示的帧数 //
else {
    framesNum += 1; // 缓存交换函数每执行一次, framesNum 增加 1 //
    frameTime = currentTime - previousTime;
    totalTime = currentTime - startTime; // 当前时间 currentTime 减去开始时间 startTime, 即总共花费时间 totalTime //
    previousTime = nowTime;
    if (totalTime > 1000) {
        fps = framesNum * 1000 / totalTime;
        startTime = nowTime;
        framesNum = 0;
    } // 当 totalTime 大于 1000 毫秒, 输出一次 fps, fps 的值等于总共的帧数除以所用总共时间。然后将当前时间重新设为开始时间 startTime, 将 framesNum 重新设为 0。再进行下一次新的计算。 //
}
```

[0037] 由以上具体应用实施例可知, 本发明提供了一种基于安卓设备的 FPS 值获取方法, 通过在安卓设备刷新屏幕时, 对自定义的函数的调用次数进行计数, 计算出 FPS 值, 并将该值动态显示在屏幕上, 供用户参考安卓设备的流畅性。

[0038] 基于上述实施例, 本发明还提供了一种基于安卓设备的 FPS 值的获取系统的较佳实施例的功能原理框图, 如图 3 所示, 其中所述系统包括:

检测模块 410, 用于安卓设备实时检测用户是否对应用程序进行点击或滑动操作; 具体如上所述。

[0039] 获取与显示模块 420, 用于当检测到用户对应用程序进行点击或滑动操作时, 获取当前 FPS 值, 并将所述当前 FPS 值动态显示在屏幕上; 具体如上所述。

[0040] 所述的基于安卓设备的 FPS 值获取系统, 其中, 所述获取与显示模块 420 具体包括:

建立与设置单元, 用于建立动态链接库文件, 设置一交换缓冲函数, 判断是否进行显示初始化; 具体如上所述。

[0041] 替换与检测单元, 用于如果进行显示初始化, 则将系统中的交换缓冲函数地址替

换为所述自定义的函数地址,检测触摸感应驱动是否开始刷新屏幕;具体如上所述。

[0042] 调用与计算单元,用于如果是则调用自定义的交换缓冲函数,对函数调用进行计数,计算 FPS 值;具体如上所述。

[0043] 写入与显示单元,用于将计算得到的 FPS 值写入用于数据通信的 Pipe 文件,建立服务读取 Pipe 文件,将得到 FPS 值显示到屏幕上;具体如上所述。

[0044] 所述的基于安卓设备的 FPS 值获取系统,其中,所述步骤写入与显示单元具体包括:

监听与创建单元,用于监听 Pipe 服务的启动,进行 Pipe 的创建,创建悬浮文字,并显示悬浮文字,执行注入的命令;具体如上所述。

[0045] 启动与显示单元,用于在监听服务启动后,启动线程,线程每隔一预定时间可从 Pipe 读取触摸感应驱动传来的 FPS 值,并显示在屏幕上;具体如上所述。

[0046] 所述的基于安卓设备的 FPS 值获取系统,其中,所述调用与计算单元中 FPS 的计算方法为:

通过计算交换缓冲函数在一秒钟执行的次数或是通过单位时间与单次执行函数所有时间的比值;具体如上所述。

[0047] 所述的基于安卓设备的 FPS 值获取系统,其中,所述预定时间为 1s;具体如上所述。

[0048] 本发明提供了一种基于安卓设备的 FPS 值获取方法及获取系统,所述方法包括:安卓设备实时检测用户是否对应用程序进行点击或滑动操作;当检测到用户对应用程序进行点击或滑动操作时,获取当前 FPS 值,并将所述当前 FPS 值动态显示在屏幕上。采用本发明可避免对安卓系统的修改,更精确的获得了评价安卓设备的流畅性的 FPS 值,同时将 FPS 值显示在屏幕上,更方便用户观看。

[0049] 应当理解的是,本发明的应用不限于上述的举例,对本领域普通技术人员来说,可以根据上述说明加以改进或变换,所有这些改进和变换都应属于本发明所附权利要求的保护范围。

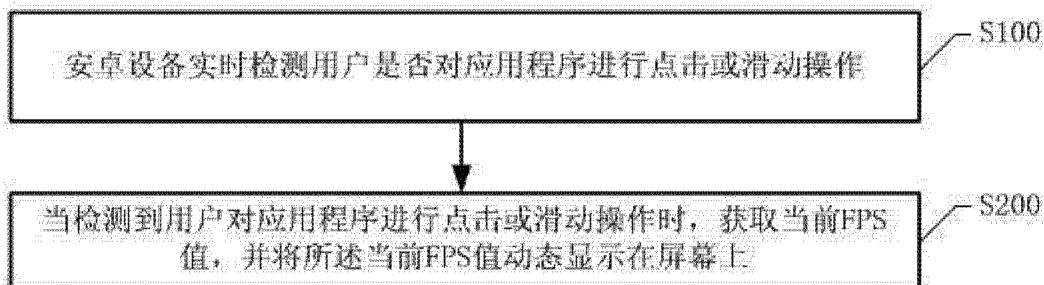


图 1

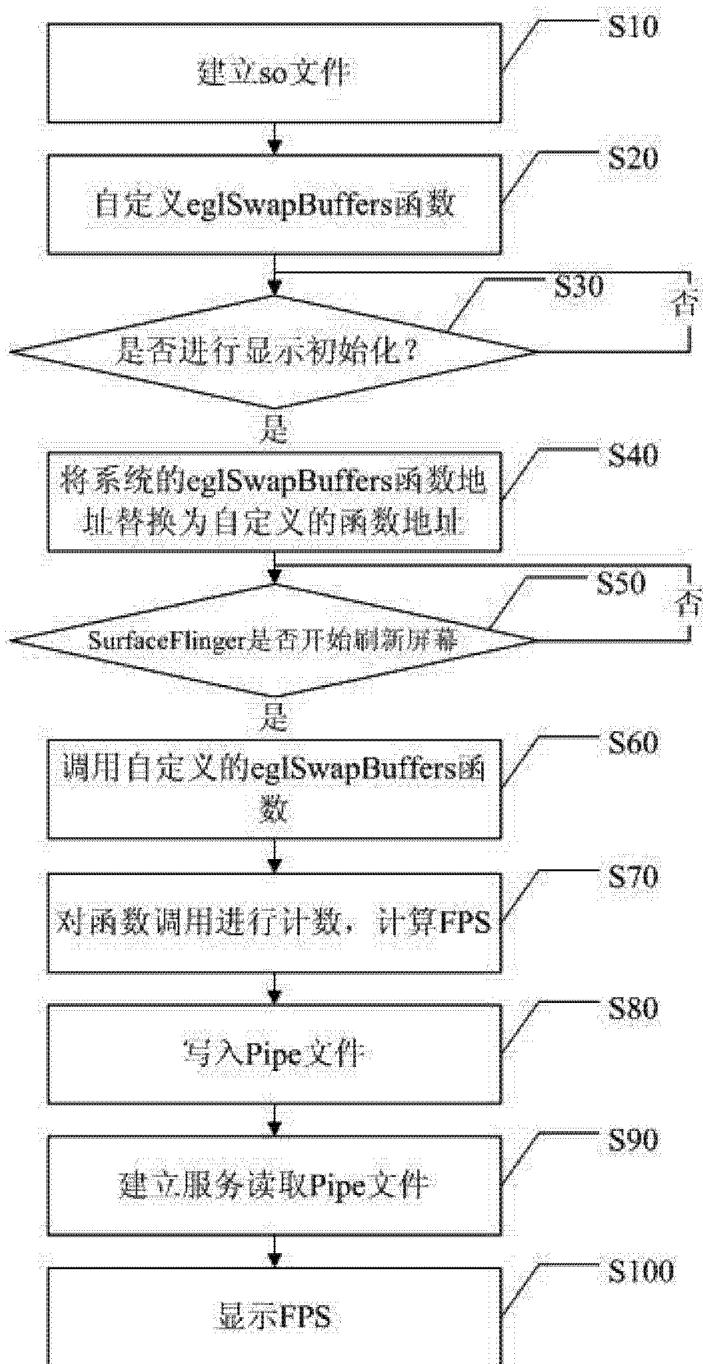


图 2

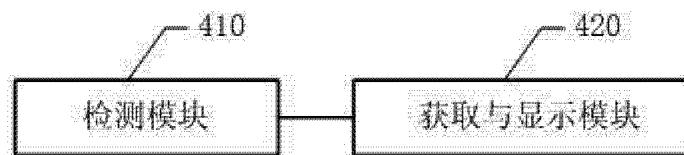


图 3