



(12) 发明专利申请

(10) 申请公布号 CN 106569652 A

(43) 申请公布日 2017. 04. 19

(21) 申请号 201510657328. 5

(22) 申请日 2015. 10. 13

(71) 申请人 福州瑞芯微电子股份有限公司

地址 350003 福建省福州市鼓楼区软件大道  
89 号 18 号楼

(72) 发明人 林銮云 沈志成

(74) 专利代理机构 福州市景弘专利代理事务所

(普通合伙) 35219

代理人 林祥翔 吕元辉

(51) Int. Cl.

G06F 3/0481(2013. 01)

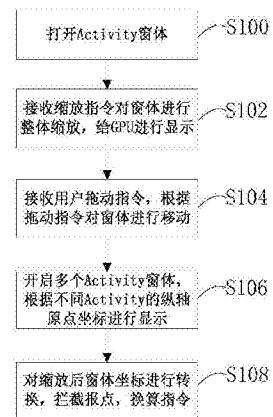
权利要求书2页 说明书6页 附图4页

(54) 发明名称

一种安卓多窗口显示方法及装置

(57) 摘要

本发明为一种新的安卓多窗体显示方法及装置,其中方法包括如下步骤,S100 接收用户指令,根据用户指令打开Activity窗体 ;S102接收缩放指令对窗体进行缩放 ;S104接收用户拖动指令,根据拖动指令对窗体进行移动 ;S106开启多个Activity窗体,根据不同Activity的纵轴原点坐标进行显示。达到了显示多窗体并能够移动缩放的技术效果。



1. 一种安卓多窗口显示方法,其特征在于,包括如下步骤,

接收用户指令,根据用户指令打开 Activity 窗体;

接收缩放指令对窗体进行缩放;

接收用户拖动指令,根据拖动指令对窗体进行移动;

开启多个 Activity 窗体,根据不同 Activity 的纵轴原点坐标进行显示。

2. 根据权利要求 1 所述的安卓多窗口显示方法,其特征在于,开启多个 Activity 窗体的方法为:

修改后台任务管理系统,只有当检测到用户最小化或关闭 Activity 对应的窗口时,才将该 Activity 关闭或切换至后台,并停止运行。

3. 根据权利要求 1 所述的安卓多窗口显示方法,其特征在于,对窗体进行缩放或移动的方法为,对每个 Activity 窗体加入横轴,竖轴原点坐标、横轴,竖轴缩放比例参数值以及纵轴原点坐标;通过调整上述参数对窗体进行缩放或移动;

其中所述横轴,竖轴原点坐标用于表示 Activity 窗体的位置;

所述横轴,竖轴缩放比例参数值用于表示 Activity 相对于屏幕显示面积的大小;

所述纵轴原点坐标用于表示不同 Activity 窗体显示的叠放次序。

4. 根据权利要求 1 所述的安卓多窗口显示方法,其特征在于,还包括步骤,

对缩放后的窗体坐标进行转换;

拦截缩放后的屏幕触摸报点,通过把当前触摸点换算到当前 Activity 在全屏时的位置得到换算指令;

根据换算指令对当前 Activity 进行操作。

5. 一种安卓多窗口显示装置,其特征在于,包括窗体打开模块、缩放模块、移动模块、多窗体显示模块,

所述窗体打开模块用于接收用户指令,根据用户指令打开 Activity 窗体;

所述缩放模块用于接收缩放指令对窗体进行缩放;

所述移动模块用于接收用户拖动指令,根据拖动指令对窗体进行移动;

所述多窗体显示模块用于开启多个 Activity 窗体,根据不同 Activity 的纵轴原点坐标进行显示。

6. 根据权利要求 5 所述的安卓多窗口显示装置,其特征在于,所述多窗体显示模块用于开启多个 Activity 窗体的方法为:

修改后台任务管理系统,只有当检测到用户最小化或关闭 Activity 对应的窗口时,才将该 Activity 关闭或切换至后台,并停止运行。

7. 根据权利要求 5 所述的安卓多窗口显示装置,其特征在于,所述缩放模块、移动模块还用于对每个 Activity 窗体加入横轴,竖轴原点坐标、横轴,竖轴缩放比例参数值以及纵轴原点坐标;通过调整上述参数对窗体进行缩放或移动;

其中所述横轴,竖轴原点坐标用于表示 Activity 窗体的位置;

所述横轴,竖轴缩放比例参数值用于表示 Activity 相对于屏幕显示面积的大小;

所述纵轴原点坐标用于表示不同 Activity 窗体显示的叠放次序。

8. 根据权利要求 5 所述的安卓多窗口显示装置,其特征在于,还包括触摸转换模块,

所述触摸转换模块用于对缩放后的窗体坐标进行转换;拦截缩放后的屏幕触摸报点,

通过把当前触摸点换算到当前 Activity 在全屏时的位置得到换算指令 ;根据换算指令对当前 Activity 进行操作。

## 一种安卓多窗口显示方法及装置

### 技术领域

[0001] 本发明涉及安卓系统屏幕显示技术,尤其涉及一种安卓多窗口显示方法及装置。

### 背景技术

[0002] 随着移动设备的普及,大尺寸的平板电脑和手机越来越多,用户希望在同一个界面可以操作更多的应用程序,比如聊天的同时看电影,浏览网页等,但目前市面上的大多操作系统都不支持分屏显示,用户体验不好。

[0003] 例如,在三星的 Galaxy Note Pro,可以实现了四个应用程序同时显示的功能,但这些显示的应用仅仅局限于三星自己定制化的应用,对很多第三方的应用无法支持分屏显示,给用户造成很大的困扰。

[0004] 苹果 ipad pro 实现的也是二分屏的形式。界面窗体都是同时显示,不可叠加,并且都不是通过整体缩放来显示。

### 发明内容

[0005] 为此,需要提供一种能够同时显示多窗体并能够按照用户意愿进行移动的安卓显示技术。

[0006] 为实现上述目的,发明人提供了一种安卓多窗口显示方法,包括如下步骤,

[0007] 接收用户指令,根据用户指令打开 Activity 窗体;

[0008] 接收缩放指令对窗体进行缩放;

[0009] 接收用户拖动指令,根据拖动指令对窗体进行移动;

[0010] 开启多个 Activity 窗体,根据不同 Activity 的纵轴原点坐标进行显示。达到了显示多窗体并能够移动缩放的技术效果。

[0011] 具体地,开启多个 Activity 窗体的方法为:

[0012] 修改后台任务管理系统,只有当检测到用户最小化或关闭 Activity 对应的窗口时,才将该 Activity 关闭或切换至后台,并停止运行。达到了能够开启多个 Activity 的效果。

[0013] 具体地,对窗体进行缩放或移动的方法为,对每个 Activity 窗体加入横轴,竖轴原点坐标、横轴,竖轴缩放比例参数值以及纵轴原点坐标;通过调整上述参数对窗体进行缩放或移动;

[0014] 其中所述横轴,竖轴原点坐标用于表示 Activity 窗体的位置;

[0015] 所述横轴,竖轴缩放比例参数值用于表示 Activity 相对于屏幕显示面积的大小;

[0016] 所述纵轴原点坐标用于表示不同 Activity 窗体显示的叠放次序。达到了能够对窗体进行缩放和移动的技术效果。

[0017] 进一步地,还包括步骤,

[0018] 对缩放后的窗体坐标进行转换;

[0019] 拦截缩放后的屏幕触摸报点,通过把当前触摸点换算到当前 Activity 在全屏时

的位置得到换算指令；

[0020] 根据换算指令对当前 Activity 进行操作。达到了能够在改变了的窗体大小中正常操作应用程序功能的效果。

[0021] 一种安卓多窗口显示装置,包括窗体打开模块、缩放模块、移动模块、多窗体显示模块,

[0022] 所述窗体打开模块用于接收用户指令,根据用户指令打开 Activity 窗体；

[0023] 所述缩放模块用于接收缩放指令对窗体进行缩放；

[0024] 所述移动模块用于接收用户拖动指令,根据拖动指令对窗体进行移动；

[0025] 所述多窗体显示模块用于开启多个 Activity 窗体,根据不同 Activity 的纵轴原点坐标进行显示。该装置达到了显示多窗体并能够移动缩放的技术效果。

[0026] 具体地,所述多窗体显示模块用于开启多个 Activity 窗体的方法为：

[0027] 修改后台任务管理系统,只有当检测到用户最小化或关闭 Activity 对应的窗口时,才将该 Activity 关闭或切换至后台,并停止运行。达到了能够开启多个 Activity 的效果。

[0028] 具体地,所述缩放模块、移动模块还用于对每个 Activity 窗体加入横轴,竖轴原点坐标、横轴,竖轴缩放比例参数值以及纵轴原点坐标；通过调整上述参数对窗体进行缩放或移动；

[0029] 其中所述横轴,竖轴原点坐标用于表示 Activity 窗体的位置；

[0030] 所述横轴,竖轴缩放比例参数值用于表示 Activity 相对于屏幕显示面积的大小；

[0031] 所述纵轴原点坐标用于表示不同 Activity 窗体显示的叠放次序。达到了能够对窗体进行缩放和移动的技术效果。

[0032] 优选地,还包括触摸转换模块,

[0033] 所述触摸转换模块用于对缩放后的窗体坐标进行转换；拦截缩放后的屏幕触摸报点,通过把当前触摸点换算到当前 Activity 在全屏时的位置得到换算指令；根据换算指令对当前 Activity 进行操作。达到了能够在改变了的窗体大小中正常操作应用程序功能的效果。

[0034] 区别于现有技术,上述技术方案不但能够同时显示多窗体还能够按照用户意愿进行移动,是一种先进的安卓显示技术。

## 附图说明

[0035] 图 1 为本发明具体实施方式所述的方法流程示意图；

[0036] 图 2 为本发明具体实施方式所述的窗体打开状态图；

[0037] 图 3 为本发明具体实施方式所述的窗体缩放状态图；

[0038] 图 4 为本发明具体实施方式所述的窗体移动状态图；

[0039] 图 5 为本发明具体实施方式所述的窗体超出屏幕边缘状态图；

[0040] 图 6 为本发明具体实施方式所述的多窗体打开状态图；

[0041] 图 7 为本发明具体实施方式所述的装置模块图。

[0042] 附图标记说明：

[0043] 700、窗体打开模块；

- [0044] 702、缩放模块；
- [0045] 704、移动模块；
- [0046] 706、多窗体显示模块；
- [0047] 708、触摸转换模块。

### 具体实施方式

[0048] 为详细说明技术方案的技术内容、构造特征、所实现目的及效果，以下结合具体实施例并配合附图详予说明。

[0049] 请参阅图 1，为本发明一种新的多窗体显示方法，包括如下步骤，

[0050] S100 接收用户指令，根据用户指令打开 Activity 窗体；

[0051] S102 接收缩放指令对窗体进行缩放；

[0052] S104 接收用户拖动指令，根据拖动指令对窗体进行移动；

[0053] S106 开启多个 Activity 窗体，根据不同 Activity 的纵轴原点坐标进行显示。达到了显示多窗体并能够移动缩放的技术效果。

[0054] 在某些实施例中，每个窗体都有一个标题栏，提供了最大化、最小化、关闭操作（标题栏示意图如附图 2 所示），通过对“最大化按钮”或者双击标题栏操作，用户可以较方便地对该窗体的大小进行调整，实现每个窗体三级大小显示。对各个窗体进行最大化、最小化、关闭操作，具体为：

[0055] 1)、当检测到用户点击一个应用，显示该应用界面后，通过标题栏的最大化按钮或双击标题栏，应用界面窗体会进行整体缩放，

[0056] 2)、为了方便安卓用户，该窗体的大小调整分为三级，每一级缩放的比例都不同，可以设置三级缩放比例分别为 0.8, 0.65, 0.5；

[0057] 原生的 Android 系统都是全屏显示，只有一个任务 (Task) 处于运行状态，Android 运行以 Task 为单位。Task 可包含多个 Activity，当另一个 activity 被激活至前台时，原先位于前台的 activity 被放置至后台同时停止运行，而在本实施例中，通过修改系统的后台任务管理系统，只有当检测到用户最小化或关闭 Activity 对应的窗口时，才将该 Activity 关闭或切换至后台，并停止运行，也就是说，显示在桌面之上的任务都可以处于运行状态，当新启动应用时，不关闭当前已经显示的应用，是否显示在桌面上新添加参数保存在该应用的数据结构中，只有通过最小化，后退、关闭等操作时，才将该应用移动到后台中，修改该参数状态，同时停止运行该程序并销毁窗体显示。通过上述步骤保证显示在桌面之上的任务不会被停止运行。可以在移动设备上同时显示多个应用程序的窗体，通过操作窗体标题栏调控按钮图标或者双击，可对窗体的大小进行缩放，可以比较方便地设置各个窗体的最大化、最小化、关闭等操作。

[0058] Android 系统一个任务 (Task) 可以包含多个 Activity，每个 Activity 可以包含多个窗口，相同 Activity 内窗口的排列是按照 Z 轴 Z-ordered 次序排列的，Z-ordered 次序可看作是屏幕法向量方向上的坐标轴，值越大的层意味着离用户越近，会把值小的窗口给盖住。系统窗口管理者根据各窗口的遮挡关系来做 layout 和分配释放 Surface，这个 Z-ordered 信息会转化为窗口对应 Surface 的 layer 属性输出到 GPU，指导 GPU 的渲染。

[0059] 系统保存历史次序的 Task 列表，每个 Task 又包含了 Z-ordered 信息的应用队

列,应用的成员 windows 又包含了一个 Z-ordered 的窗口列表。因此这个列表包含了这个 Activity 中的所有窗口,子窗口,开始窗口等。另一方面,系统显示框架中也会保存 windows 列表,它描述的是单个显示屏上的窗口集合。当点击不同的窗体时,切换任务 (TASK) 列表的次序,之后会根据这个 Z-order 列表信息得到每个窗口的 layer 值,重新绘制。

[0060] 在一些实施例中,对每个 Activity 加入横轴,竖轴原点坐标、横轴,竖轴缩放比例等参数值以及纵轴原点坐标,其中横轴、竖轴原点坐标分别用  $x, y$  表示,可以定义为相对于屏幕左上角点的位置,纵轴原点坐标用  $z$  表示,用于定义图层所在的高度,其中横轴竖轴所在平面与屏幕所在平面平行,所述纵轴与屏幕所在平面垂直。横轴竖轴纵轴的原点及朝向都可以根据习惯或实际需要而调整,在另一些实施例中,横竖轴的原点可以定义在屏幕左下角,纵轴的朝向可以定义为垂直于屏幕向内等等。通过这些参数重新计算窗体对应 surface 的 Matrix(图像矩阵,通过使用矩阵的方法可实现缩放,旋转等功能)来实现对窗体整体缩放,然后得出缩放后的窗体大小交给 GPU 显示。窗体数据仍为全屏数据,在这种模式下应用的界面不会改变,保证了窗体完整性。如图 2 所示的实施例中,进行步骤 S100 打开新的窗体 Activity1,图中所示 Activity1 占据了整个屏幕大小,我们以屏幕大小为  $1900 \times 1080$  为例,此时 Activity1 的横轴,竖轴原点坐标、横轴,竖轴缩放比例等参数值分别为  $(0, 0, 100\%, 100\%)$ ;当需要对 Activity1 进行缩放时,用户可以发出缩放指令,因此还包括步骤 S102 接收用户发出的缩放指令后对窗体进行整体缩放,给 GPU 进行显示。在我们的实施例中,缩放指令为双击 Activity1 的标题栏,当检测到用户发出缩放指令后,对对应的窗体缩放比例参数值进行调整,为了方便用户,可以设置多档缩放比例,每检测到一次缩放指令,所述缩放比例参数值便在  $(100\%, 70\%, 50\%, 35\%)$  中循环切换。在图 3 所示的实施例中,用户两次双击 Activity1 的标题栏后,Activity1 的横轴,竖轴原点坐标、横轴,竖轴缩放比例等参数值分别变为  $(0, 0, 50\%, 50\%)$ ,则此时 Activity1 的原点位于屏幕的左上角点,横纵长度均为原屏幕的一半,窗口大小变为  $950 \times 540$ 。

[0061] 在某些进一步的实施例中,还包括步骤 S104,接收用户拖动指令,根据用户拖动指令改变对应 Activity 的横轴,竖轴原点坐标,根据变更后的横轴,竖轴原点坐标进行显示。如图 4 所示的实施例中,以屏幕大小为  $1900 \times 1080$  为例,用户可以通过长按 Activity1 的标题栏来发出拖动指令,根据该拖动指令 Activity1 将向右下角移动,在本实施例中移动完毕后 Activity1 的  $(x, y)$  变为  $(475, 270)$ ,缩放比例不变,则 Activity1 显示的位置变为屏幕正中。当拖动后的窗体超出屏幕范围之后,超出屏幕显示范围的部分将不予显示,如图 5 所示,Activity5 的横轴,竖轴原点坐标、横轴,竖轴缩放比例等参数值分别为  $(950, 540, 100\%, 100\%)$ ,屏幕大小仍为  $1900 \times 1080$  的情况下,拖动后屏幕中只显示了 Activity5 的左上区域。

[0062] 在进一步的实施例中,包括多个正在运行的 Activity,因此还包括步骤 S106:打开多个 Activity 窗体,为不同的 Activity 赋予不同的横轴,竖轴,纵轴原点坐标,根据不同 Activity 的横轴,竖轴原点坐标、横轴,竖轴缩放比例进行显示,所述纵轴垂直于屏幕,纵轴原点坐标  $z$  定义为画面所在的高度,因此当不同的 Activity 的显示画面有重叠时,对纵轴原点坐标大的 Activity 画面进行在前显示。在图 5 所示的实施例中,当前运行的 Activity1 的  $(x, y, z)$  分别为  $(475, 270, 0)$  此时新运行的窗体 Activity6 一般赋值为当前

运行的  $(x+50, y+50, z+1)$  即  $(525, 320, 1)$ 。则本实施例中 Activity6 在新运行之后便默认在 Activity1 的上方进行显示。赋予新的横竖纵轴原点坐标是为了能够将多窗体应用进行错开位置显示。当需要将原本被覆盖或图层较为靠后的 Activity 时, 用户可以触摸该 Activity, 检测到触摸后, 调整对应的 Activity 纵轴原点坐标使对应的 Activity 纵轴原点坐标最大, 能够在靠前的图层进行显示。参数值的变更能够随时告知 GPU 进行处理, 实时进行窗口显示的切换。

[0063] 由于安卓系统默认当前应用为全屏显示状态,

[0064] 在某些进一步地实施例中, 还包括步骤 S108: 对缩放后的窗体坐标进行转换, 拦截缩放后的屏幕触摸报点, 通过把当前触摸点换算到当前 Activity 在全屏时的位置得到换算指令, 根据换算指令对当前 Activity 进行操作。如图 3 所示的实施例中, 用户触点为  $(925, 25)$  的屏幕位置时, 需要将其转化为全屏显示下的触摸位置, 即  $(1850, 50)$  的位置, 此处对应的换算指令是关闭当前 Activity, 那么系统将进行关闭当前 Activity1 的操作。

[0065] 在图 7 所示的实施例中, 一种安卓多窗口显示装置, 包括窗体打开模块 700、缩放模块 702、移动模块 704、多窗体显示模块 706,

[0066] 所述窗体打开模块 700 用于接收用户指令, 根据用户指令打开 Activity 窗体;

[0067] 所述缩放模块 702 用于接收缩放指令对窗体进行缩放;

[0068] 所述移动模块 704 用于接收用户拖动指令, 根据拖动指令对窗体进行移动;

[0069] 所述多窗体显示模块 706 用于开启多个 Activity 窗体, 根据不同 Activity 的纵轴原点坐标进行显示。该装置达到了显示多窗体并能够移动缩放的技术效果。

[0070] 具体地, 所述多窗体显示模块 706 用于开启多个 Activity 窗体的方法为:

[0071] 修改后台任务管理系统, 只有当检测到用户最小化或关闭 Activity 对应的窗口时, 才将该 Activity 关闭或切换至后台, 并停止运行。达到了能够开启多个 Activity 的效果。

[0072] 具体地, 所述缩放模块 702、移动模块 704 还用于对每个 Activity 窗体加入横轴, 竖轴原点坐标、横轴, 竖轴缩放比例参数值以及纵轴原点坐标; 通过调整上述参数对窗体进行缩放或移动;

[0073] 其中所述横轴, 竖轴原点坐标用于表示 Activity 窗体的位置;

[0074] 所述横轴, 竖轴缩放比例参数值用于表示 Activity 相对于屏幕显示面积的大小;

[0075] 所述纵轴原点坐标用于表示不同 Activity 窗体显示的叠放次序。达到了能够对窗体进行缩放和移动的技术效果。

[0076] 优选地, 还包括触摸转换模块 708,

[0077] 所述触摸转换模块 708 用于对缩放后的窗体坐标进行转换; 拦截缩放后的屏幕触摸报点, 通过把当前触摸点换算到当前 Activity 在全屏时的位置得到换算指令; 根据换算指令对当前 Activity 进行操作。达到了能够在改变了的窗体大小中正常操作应用程序功能的效果。

[0078] 需要说明的是, 在本文中, 诸如第一和第二等之类的关系术语仅仅用来将一个实体或者操作与另一个实体或操作区分开来, 而不一定要求或者暗示这些实体或操作之间存在任何这种实际的关系或者顺序。而且, 术语“包括”、“包含”或者其任何其他变体意在涵盖非排他性的包含, 从而使得包括一系列要素的过程、方法、物品或者终端设备不仅包括那些



要素,而且还包括没有明确列出的其他要素,或者是还包括为这种过程、方法、物品或者终端设备所固有的要素。在没有更多限制的情况下,由语句“包括……”或“包含……”限定的要素,并不排除在包括所述要素的过程、方法、物品或者终端设备中还存在另外的要素。此外,在本文中,“大于”、“小于”、“超过”等理解为不包括本数;“以上”、“以下”、“以内”等理解为包括本数。

[0079] 本领域内的技术人员应明白,上述各实施例可提供为方法、装置、或计算机程序产品。这些实施例可采用完全硬件实施例、完全软件实施例、或结合软件和硬件方面的实施例的形式。上述各实施例涉及的方法中的全部或部分步骤可以通过程序来指令相关的硬件来完成,所述的程序可以存储于计算机设备可读的存储介质中,用于执行上述各实施例方法所述的全部或部分步骤。所述计算机设备,包括但不限于:个人计算机、服务器、通用计算机、专用计算机、网络设备、嵌入式设备、可编程设备、智能移动终端、智能家居设备、穿戴式智能设备、车载智能设备等;所述的存储介质,包括但不限于:RAM、ROM、磁碟、磁带、光盘、闪存、U 盘、移动硬盘、存储卡、记忆棒、网络服务器存储、网络云存储等。

[0080] 上述各实施例是参照根据实施例所述的方法、设备(系统)、和计算机程序产品的流程图和/或方框图来描述的。应理解可由计算机程序指令实现流程图和/或方框图中的每一流程和/或方框、以及流程图和/或方框图中的流程和/或方框的结合。可提供这些计算机程序指令到计算机设备的处理器以产生一个机器,使得通过计算机设备的处理器执行的指令产生用于实现在流程图一个流程或多个流程和/或方框图一个方框或多个方框中指定的功能的装置。

[0081] 这些计算机程序指令也可存储在能引导计算机设备以特定方式工作的计算机设备可读存储器中,使得存储在该计算机设备可读存储器中的指令产生包括指令装置的制品,该指令装置实现在流程图一个流程或多个流程和/或方框图一个方框或多个方框中指定的功能。

[0082] 这些计算机程序指令也可装载到计算机设备上,使得在计算机设备上执行一系列操作步骤以产生计算机实现的处理,从而在计算机设备上执行的指令提供用于实现在流程图一个流程或多个流程和/或方框图一个方框或多个方框中指定的功能的步骤。

[0083] 尽管已经对上述各实施例进行了描述,但本领域内的技术人员一旦得知了基本创造性概念,则可对这些实施例做出另外的变更和修改,所以以上所述仅为本发明的实施例,并非因此限制本发明的专利保护范围,凡是利用本发明说明书及附图内容所作的等效结构或等效流程变换,或直接或间接运用在其他相关的技术领域,均同理包括在本发明的专利保护范围之内。

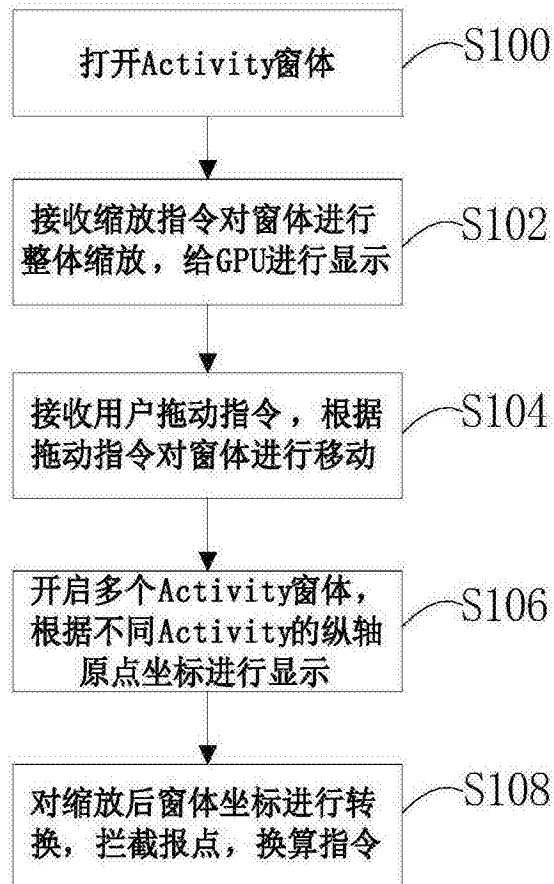


图 1



图 2



图 3

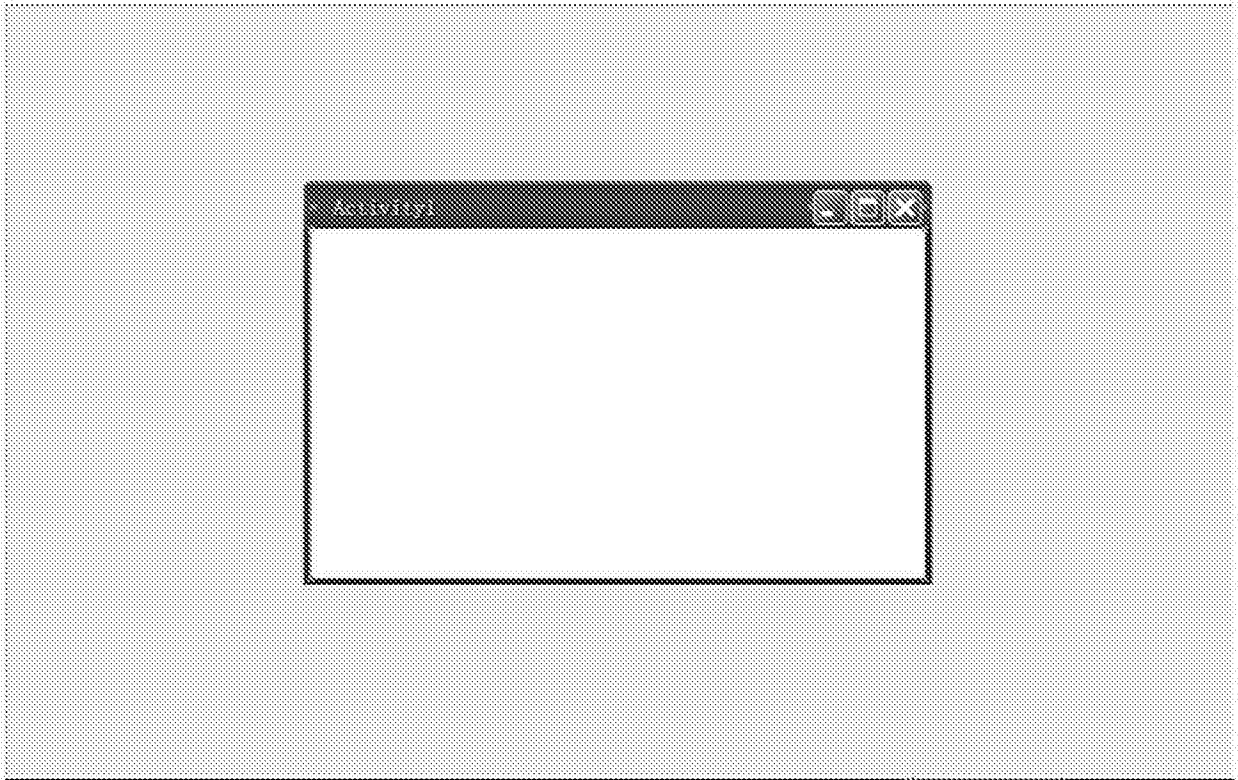


图 4

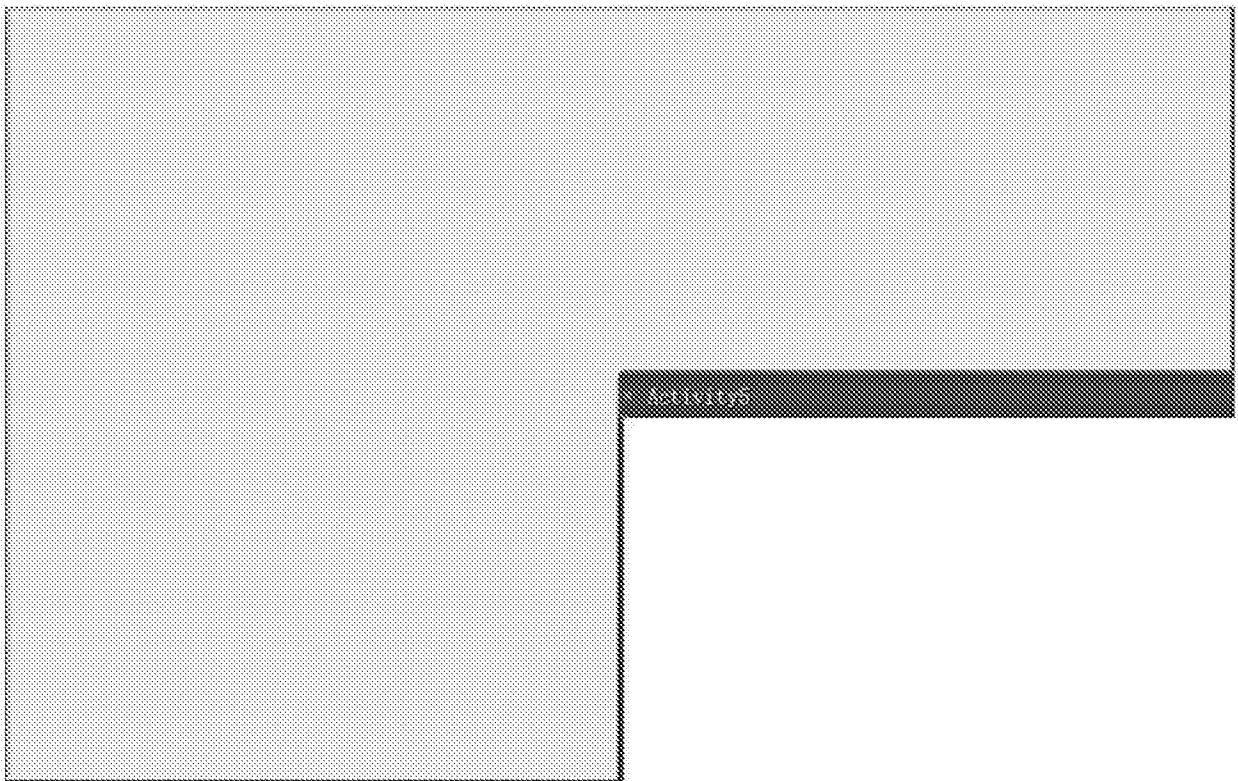


图 5

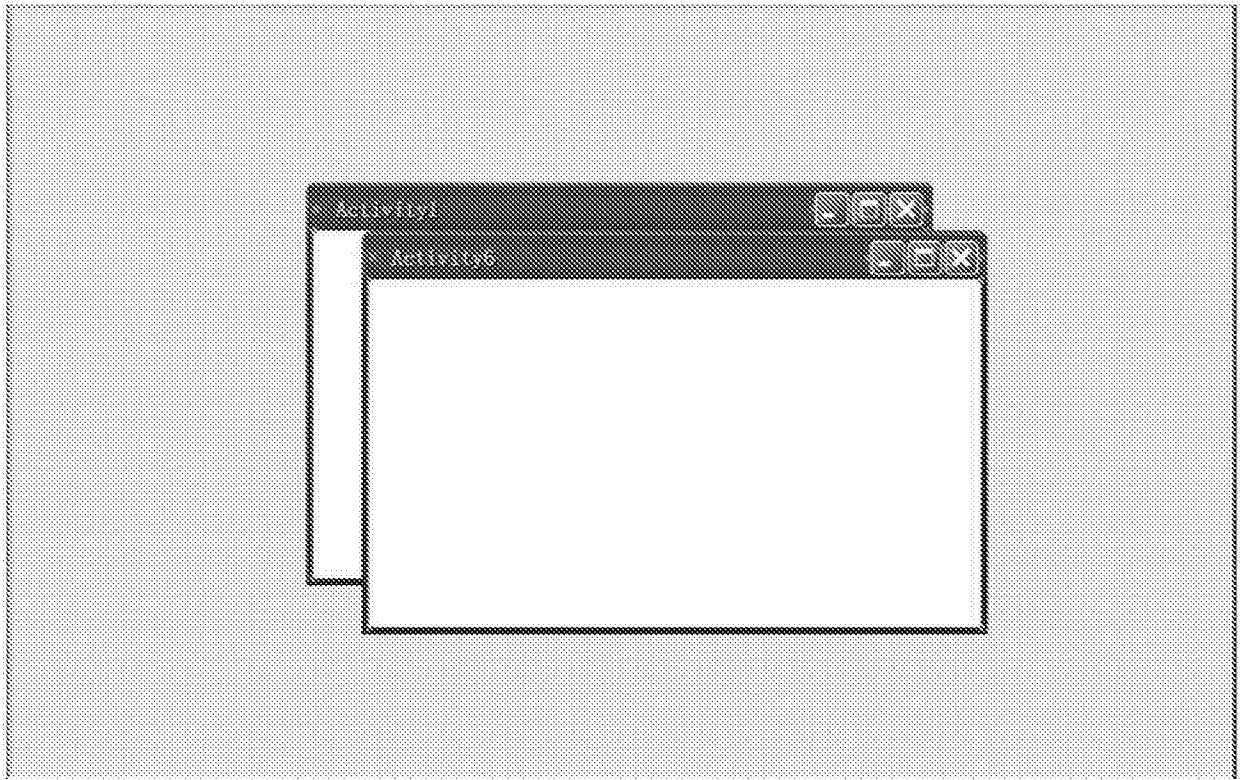


图 6

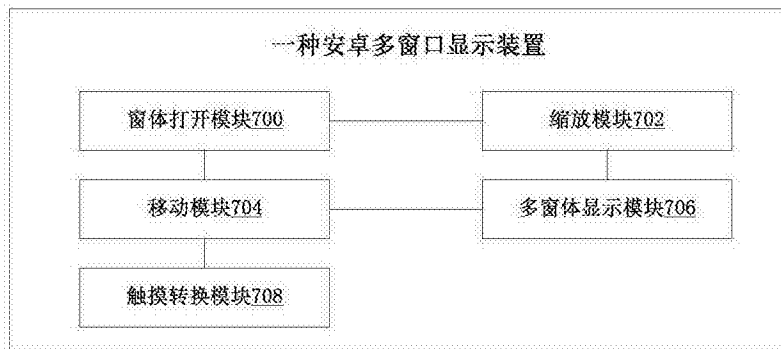


图 7