



(12)发明专利申请

(10)申请公布号 CN 110888635 A

(43)申请公布日 2020.03.17

(21)申请号 201911192904.8

(22)申请日 2019.11.28

(71)申请人 百度在线网络技术(北京)有限公司
地址 100085 北京市海淀区上地十街10号
百度大厦三层

(72)发明人 陈荣杭 邓小红 付嘉兴 雷志兴

(74)专利代理机构 北京市铸成律师事务所
11313
代理人 王一 杨瑾瑾

(51) Int. Cl.

G06F 8/30(2018.01)

G06F 8/38(2018.01)

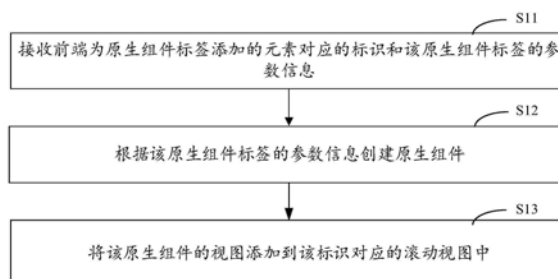
权利要求书3页 说明书19页 附图6页

(54)发明名称

同层渲染方法、装置、电子设备和存储介质

(57)摘要

本申请公开了同层渲染方法、装置、电子设备和存储介质,涉及网页技术领域。具体实现方案包括:接收前端为原生组件标签添加的元素对应的标识和原生组件标签的参数信息;根据该原生组件标签的参数信息创建原生组件;将该原生组件的视图添加到该标识对应的滚动视图中。本申请实施例通过为原生组件标签添加元素,利用该元素生成滚动视图,将原生组件的视图嵌套到滚动视图中,实现同层渲染。



1. 一种同层渲染方法,其特征在于,应用于客户端,所述同层渲染方法包括:
接收前端为原生组件标签添加的元素对应的标识和所述原生组件标签的参数信息;
根据所述原生组件标签的参数信息创建原生组件;
将所述原生组件的视图添加到所述标识对应的滚动视图中。
2. 根据权利要求1所述的同层渲染方法,其特征在于,还包括:
接收所述前端为原生组件标签添加的元素对应的标识;
查找所述标识对应的滚动视图。
3. 根据权利要求2所述的同层渲染方法,其特征在于,还包括:
接收所述元素对应的滚动视图的参数;
根据接收到的滚动视图的参数校验所述标识对应的滚动视图的参数;
在查找到的滚动视图的参数与接收到的滚动视图的参数一致的情况下,向所述前端返回成功找到滚动视图的结果。
4. 根据权利要求1所述的同层渲染方法,其特征在于,还包括:
保存所述标识与利用所述元素生成的滚动视图的对应关系。
5. 根据权利要求4所述的同层渲染方法,其特征在于,保存所述标识与利用所述元素生成的滚动视图的对应关系,包括:
在利用所述元素生成的滚动视图被添加到视图树时,触发视图移动到窗口方法:获取所生成的滚动视图的父视图,从所述父视图中解析得到所述标识,并保存所述标识和利用所述元素生成的滚动视图的内存地址的对应关系。
6. 根据权利要求1所述的同层渲染方法,其特征在于,还包括:
在所述元素发生回流的情况下,更新所述标识与利用所述元素重新生成的滚动视图的对应关系。
7. 根据权利要求6所述的同层渲染方法,其特征在于,更新所述标识与利用所述元素重新生成的滚动视图的对应关系,包括:
在利用所述元素重新生成的滚动视图被添加到视图树时,触发视图移动到窗口方法:获取重新生成的滚动视图的父视图,从所述父视图中解析得到所述标识,更新所述标识和利用所述元素重新生成的滚动视图的内存地址的对应关系。
8. 根据权利要求1所述的同层渲染方法,其特征在于,还包括:
在所述元素发生重绘的情况下,接收所述标识和更新的滚动视图参数;
查找所述标识对应的滚动视图;
在查找到的滚动视图的参数与接收到的滚动视图的参数一致的情况下,向所述前端返回成功找到滚动视图的结果;
根据重绘后的元素更新原生组件。
9. 根据权利要求1至7中任一项所述的同层渲染方法,其特征在于,所述元素包括HTML元素,所述HTML元素包括同层渲染容器占位标签,所述标识为容器标识。
10. 根据权利要求1至7中任一项所述的同层渲染方法,其特征在于,所述原生组件标签的参数信息包括以下至少之一:
所述原生组件为视频组件,所述视频组件标签的参数信息包括资源地址、播放方式和控件显示方式中的至少之一;

所述原生组件为相机组件,所述相机组件标签的参数信息包括应用模式、摄像头朝向、闪光灯开启方式、触发方式和帧数据尺寸中的至少之一;

所述原生组件为画布组件,所述画布组件标签的参数信息包括画布类型、画布标识、触摸动作响应方式和事件触发方式中的至少之一;

所述原生组件为地图组件,所述地图组件标签的参数信息包括经纬度、缩放级别、路线、显示方式和触发方式中的至少之一;

所述原生组件为音频组件,所述音频组件标签的参数信息包括音频组件标识、资源地址、播放方式、控件显示方式和事件触发方式中的至少之一;

所述原生组件为直播组件,所述直播组件标签的参数信息包括资源地址、模式、播放方式、显示方式和事件触发方式中的至少之一;

所述原生组件为输入框组件,所述输入框组件标签的参数信息包括输入框的内容、样式、控件显示方式和事件触发方式中的至少之一。

11. 根据权利要求1至7中任一项所述的同层渲染方法,其特征在于,还包括:

获取网页渲染组件根据触摸事件计算的第一视图;

重写网页渲染组件的寻找响应事件视图的方法,以将网页渲染组件的触摸事件转发给组件容器,由组件容器将触摸事件派发给所述组件容器上的所有原生组件,由各所述原生组件判断是否需要响应所述触摸事件。

12. 根据权利要求11所述的同层渲染方法,其特征在于,所述原生组件判断是否需要响应所述触摸事件包括:查找所述原生组件是否具有同层渲染标识,如果有,则确定所述原生组件被设置为同层渲染;在所述原生组件被设置为同层渲染且所述原生组件的父视图为在视图树中滚动视图的情况下,调用所述父视图的寻找响应事件视图的方法;

在接收到一个原生组件返回的需要响应触摸事件的第二视图的情况下,利用所述第二视图替换所述第一视图。

13. 一种同层渲染方法,其特征在于,应用于前端,所述同层渲染方法包括:

向客户端发送原生组件标签添加的元素对应的标识和所述原生组件标签的参数信息,以使得所述客户端根据所述原生组件标签的参数信息创建原生组件、并将所述原生组件的视图添加到所述标识对应的滚动视图中。

14. 根据权利要求13所述的同层渲染方法,其特征在于,还包括:

向客户端发送为原生组件标签添加的元素对应的标识,以使得所述客户端查找所述标识对应的滚动视图。

15. 根据权利要求13所述的同层渲染方法,其特征在于,还包括:

向所述客户端发送所述元素对应的滚动视图的参数,以使得所述客户端根据接收到的滚动视图的参数校验所述标识对应的滚动视图的参数。

16. 根据权利要求13至15中任一项所述的同层渲染方法,其特征在于,还包括:

在确定原生组件标签需要同层渲染的情况下,为所述原生组件标签添加所述元素。

17. 一种同层渲染装置,其特征在于,应用于客户端,所述同层渲染装置包括:

第一接收模块,用于接收前端为原生组件标签添加的元素对应的标识和所述原生组件标签的参数信息;

创建模块,用于根据所述原生组件标签的参数信息创建原生组件;

添加模块,用于将所述原生组件的视图添加到所述标识对应的滚动视图中。

18.根据权利要求17所述的同层渲染装置,其特征在于,还包括:

第二接收模块,用于接收所述前端为原生组件标签添加的元素对应的标识;

查找模块,用于查找所述标识对应的滚动视图。

19.一种同层渲染装置,其特征在于,应用于前端,所述同层渲染装置包括:

第一发送模块,用于向客户端发送原生组件标签添加的元素对应的标识和所述原生组件标签的参数信息,以使得所述客户端根据所述原生组件标签的参数信息创建原生组件、并将所述原生组件的视图添加到所述标识对应的滚动视图中。

20.根据权利要求19所述的同层渲染装置,其特征在于,还包括:

第二发送模块,用于向客户端发送为原生组件标签添加的元素对应的标识,以使得所述客户端查找所述标识对应的滚动视图。

21.一种电子设备,其特征在于,包括:

至少一个处理器;以及

与所述至少一个处理器通信连接的存储器;其中,

所述存储器存储有可被所述至少一个处理器执行的指令,所述指令被所述至少一个处理器执行,以使所述至少一个处理器能够执行权利要求1至16中任一项所述的方法。

22.一种存储有计算机指令的非瞬时计算机可读存储介质,其特征在于,所述计算机指令用于使所述计算机执行权利要求1至16中任一项所述的方法。

同层渲染方法、装置、电子设备和存储介质

技术领域

[0001] 本申请涉及一种计算机技术领域,尤其涉及一种网页技术领域。

背景技术

[0002] 在移动终端操作系统中,通过原生组件可以在客户端能够创建更多功能丰富的小程序。但是,由于小程序提供的原生组件并非H5 (HTML5,超文本标记语言(Hyper Text Markup Language)第五版)浏览器页面运行环境中的H5组件,存在原生组件与H5组件无法相互嵌套和较好兼容等问题。例如,由于原生组件的层级是最高,因此原生组件会始终覆盖在H5组件之上,导致H5组件的元素被遮挡。

发明内容

[0003] 本申请实施例提供一种同层渲染方法、装置、电子设备和存储介质,以解决现有技术中的一个或多个技术问题。

[0004] 第一方面,本申请实施例提供了一种同层渲染方法,应用于客户端,该同层渲染方法包括:

[0005] 接收前端为原生组件标签添加的元素对应的标识和该原生组件标签的参数信息;

[0006] 根据该原生组件标签的参数信息创建原生组件;

[0007] 将该原生组件的视图添加到该标识对应的滚动视图中。

[0008] 本申请实施例通过为原生组件标签添加元素,利用该元素生成滚动视图,将原生组件的视图嵌套到滚动视图中,实现同层渲染。这样,能够使得原生组件的视图不遮挡其他网页组件视图,并且能够随滚动视图联动。

[0009] 在一种实施方式中,该同层渲染方法还包括:

[0010] 接收前端为原生组件标签添加的元素对应的标识;

[0011] 查找该标识对应的滚动视图。

[0012] 上述实施方式利用为原生组件标签添加的元素对应的标识能够快速查找到滚动视图,能够提高同层渲染的处理速度和效率。

[0013] 在一种实施方式中,该同层渲染方法还包括:

[0014] 接收该元素对应的滚动视图的参数;

[0015] 根据接收到的滚动视图的参数校验该标识对应的滚动视图的参数;

[0016] 在查找到的滚动视图的参数与接收到的滚动视图的参数一致的情况下,向前端返回成功找到滚动视图的结果。

[0017] 上述实施方式通过为原生组件标签添加的元素对应的标识快速找到滚动视图后,通过滚动视图的位置、大小等参数对找到的滚动视图进行校验,有利于找到准确的滚动视图,提高客户端处理的准确性。

[0018] 在一种实施方式中,该同层渲染方法还包括:

[0019] 保存该标识与利用该元素生成的滚动视图的对应关系。

- [0020] 在一种实施方式中,保存该标识与利用该元素生成的滚动视图的对应关系,包括:
- [0021] 在利用该元素生成的滚动视图被添加到视图树时,触发视图移动到窗口方法:获取所生成的滚动视图的父视图,从该父视图中解析得到该标识,并保存该标识和利用该元素生成的滚动视图的内存地址的对应关系。
- [0022] 上述实施方式利用视图移动到窗口方法可以在监听到滚动视图被添加到视图树时,从滚动视图的父视图中得到标识,并保存标识和滚动视图的内存地址的对应关系,从而在后续可以利用该标识快速查找是否有对应的滚动视图。
- [0023] 在一种实施方式中,该同层渲染方法还包括:
- [0024] 在该元素发生回流的情况下,更新该标识与利用该元素重新生成的滚动视图的对应关系。
- [0025] 在一种实施方式中,更新该标识与利用该元素重新生成的滚动视图的对应关系,包括::
- [0026] 在利用该元素重新生成的滚动视图被添加到视图树时,触发视图移动到窗口方法:获取重新生成的滚动视图的父视图,从该父视图中解析得到该标识,更新该标识和利用该元素重新生成的滚动视图的内存地址的对应关系。
- [0027] 上述实施方式可以支持在元素发生重大变化的情况下启动回流。
- [0028] 在一种实施方式中,该同层渲染方法还包括:
- [0029] 在该元素发生重绘的情况下,接收该标识和更新的滚动视图参数;
- [0030] 查找该标识对应的滚动视图;
- [0031] 在查找到的滚动视图的参数与接收到的滚动视图的参数一致的情况下,向该前端返回成功找到滚动视图的结果;
- [0032] 根据重绘后的元素更新原生组件。
- [0033] 上述实施方式可以支持在元素发生微小变化的情况下启动重绘。
- [0034] 在一种实施方式中,该元素包括HTML元素,该HTML元素包括同层渲染容器占位标签,该标识为容器标识。
- [0035] 上述实施方式,通过为需要同层渲染的原生组件标签添加同层渲染容器占位标签,利用占位标签生成滚动视图,能够将原生组件的视图嵌套到滚动视图中,实现同层渲染。这样,能够使得原生组件的视图不遮挡其他网页组件视图,并且能够随滚动视图联动。利用同层渲染容器占位标签对应的容器标识能够快速查找到滚动视图,能够提高同层渲染的处理速度和效率。通过同层渲染容器占位标签对应的容器标识快速找到滚动视图后,通过滚动视图的位置、大小等参数对找到的滚动视图进行校验,有利于找到准确的滚动视图,提高客户端处理的准确性。
- [0036] 在一种实施方式中,该原生组件标签的参数信息包括以下至少之一:
- [0037] 该原生组件为视频组件,该视频组件标签的参数信息包括资源地址、播放方式和控件显示方式中的至少之一;
- [0038] 该原生组件为相机组件,该相机组件标签的参数信息包括应用模式、摄像头朝向、闪光灯开启方式、触发方式和帧数据尺寸中的至少之一;
- [0039] 该原生组件为画布组件,该画布组件标签的参数信息包括画布类型、画布标识、触摸动作响应方式和事件触发方式中的至少之一;

[0040] 该原生组件为音频组件,该音频组件标签的参数信息包括音频组件标识、资源地址、播放方式、控件显示方式和事件触发方式中的至少之一;

[0041] 该原生组件为直播组件,该直播组件标签的参数信息包括资源地址、模式、播放方式、控件显示方式和事件触发方式中的至少之一;

[0042] 该原生组件为输入框组件,该输入框组件标签的参数信息包括输入框的内容、样式、控件显示方式和事件触发方式中的至少之一。

[0043] 在一种实施方式中,该同层渲染方法还包括:

[0044] 获取网页渲染组件根据触摸事件计算的第一视图;

[0045] 重写网页渲染组件的寻找响应事件视图的方法,以将网页渲染组件的触摸事件转发给组件容器,由组件容器将触摸事件派发给组件容器上的所有原生组件,由各原生组件判断是否需要响应触摸事件。

[0046] 在一种实施方式中,原生组件判断是否需要响应触摸事件包括:查找原生组件是否具有同层渲染标识,如果有,则确定原生组件被设置为同层渲染;在原生组件被设置为同层渲染且原生组件的父视图为在视图树中滚动视图的情况下,调用父视图的寻找响应事件视图的方法;

[0047] 在接收到一个原生组件返回的需要响应触摸事件的第二视图的情况下,利用第二视图替换第一视图。

[0048] 上述实施方式可以支持原生组件返回需要响应触摸事件的视图。

[0049] 第二方面,本申请实施例提供了一种同层渲染方法,应用于前端,该同层渲染方法包括:

[0050] 向客户端发送原生组件标签添加的元素对应的标识和该原生组件标签的参数信息,以使得该客户端根据该原生组件标签的参数信息创建原生组件、并将该原生组件的视图添加到该标识对应的滚动视图中。

[0051] 在一种实施方式中,该同层渲染方法还包括:

[0052] 向客户端发送为原生组件标签添加的元素对应的标识,以使得该客户端查找该标识对应的滚动视图。

[0053] 在一种实施方式中,该同层渲染方法还包括:

[0054] 向客户端发送该元素对应的滚动视图的参数,以使得该客户端根据接收到的滚动视图的参数校验该标识对应的滚动视图的参数。

[0055] 在一种实施方式中,该同层渲染方法还包括:

[0056] 在确定原生组件标签需要同层渲染的情况下,为该原生组件标签添加该元素。

[0057] 第三方面,本申请实施例提供了一种同层渲染装置,应用于客户端,该同层渲染装置包括:

[0058] 第一接收模块,用于接收前端为原生组件标签添加的元素对应的标识和该原生组件标签的参数信息;

[0059] 创建模块,用于根据该原生组件标签的参数信息创建原生组件;

[0060] 添加模块,用于将该原生组件的视图添加到该标识对应的滚动视图中。

[0061] 在一种实施方式中,该同层渲染装置还包括:

[0062] 第二接收模块,用于接收该前端为原生组件标签添加的元素对应的标识;

[0063] 查找模块,用于查找该标识对应的滚动视图。

[0064] 第四方面,本申请实施例提供了一种同层渲染装置,应用于前端,该同层渲染装置包括:

[0065] 第一发送模块,用于向客户端发送原生组件标签添加的元素对应的标识和该原生组件标签的参数信息,以使得该客户端根据该原生组件标签的参数信息创建原生组件、并将该原生组件的视图添加到该标识对应的滚动视图中。

[0066] 在一种实施方式中,该同层渲染装置还包括:

[0067] 第二发送模块,用于向客户端发送为原生组件标签添加的元素对应的标识,以使得该客户端查找该标识对应的滚动视图。

[0068] 第五方面,本申请实施例提供了一种电子设备,电子设备的功能可以通过硬件实现,也可以通过硬件执行相应的软件实现。硬件或软件包括一个或多个与上述功能相对应的模块。

[0069] 在一个可能的设计中,电子设备的结构中包括处理器和存储器,存储器用于存储支持电子设备执行上述同层渲染方法的程序,处理器被配置为用于执行存储器中存储的程序。电子设备还可以包括通信接口,用于与其他设备或通信网络通信。

[0070] 第六方面,本申请实施例提供了一种存储有计算机指令的非瞬时计算机可读存储介质,用于存储电子设备及电子设备所用的计算机软件指令,其包括用于执行上述同层渲染方法所涉及的程序。

[0071] 上述申请中的一个实施例具有如下优点或有益效果:通过为原生组件标签添加元素,利用该元素生成滚动视图,将原生组件的视图嵌套到滚动视图中,实现同层渲染。这样,能够使得原生组件的视图不遮挡其他网页组件视图,并且能够随滚动视图联动。

[0072] 上述可选方式所具有的其他效果将在下文中结合具体实施例加以说明。

附图说明

[0073] 附图用于更好地理解本方案,不构成对本申请的限定。其中:

[0074] 图1是根据本申请一实施例同层渲染方法的示意图;

[0075] 图2是根据本申请另一实施例同层渲染方法的示意图;

[0076] 图3是根据本申请另一实施例同层渲染方法的示意图;

[0077] 图4是根据本申请另一实施例同层渲染方法的示意图;

[0078] 图5是根据本申请另一实施例同层渲染方法的示意图;

[0079] 图6是根据本申请另一实施例同层渲染装置的示意图;

[0080] 图7是根据本申请另一实施例同层渲染装置的示意图;

[0081] 图8是根据本申请另一实施例同层渲染装置的示意图;

[0082] 图9是根据本申请另一实施例同层渲染装置的示意图;

[0083] 图10是用来实现本申请实施例的同层渲染方法的电子设备的框图。

具体实施方式

[0084] 以下结合附图对本申请的示范性实施例做出说明,其中包括本申请实施例的各种细节以助于理解,应当将它们认为仅仅是示范性的。因此,本领域普通技术人员应当认识

到,可以对这里描述的实施例做出各种改变和修改,而不会背离本申请的范围和精神。同样,为了清楚和简明,以下的描述中省略了对公知功能和结构的描述。

[0085] 图1是根据本申请实施例同层渲染方法的示意图。如图1所示,该同层渲染方法可以应用于客户端,该同层渲染方法可以包括:

[0086] S11、接收前端为原生(NA,native)组件标签添加的元素对应的标识和该原生组件标签的参数信息。

[0087] S12、根据该原生组件标签的参数信息创建原生组件。

[0088] S13、将该原生组件的视图添加到该标识对应的滚动视图中。

[0089] 在一个示例中,前端可以为应用开发前端框架的简称,包括但不限于基于H5的网页(Web)前端框架、小程序前端框架、搜索结果页前端框架、Feed落地页前端框架等。客户端可以为某个应用例如手机百度的客户端。

[0090] 前端解析开发者标签,根据解析结果来确定该标签是否是原生组件标签。如果前端解析开发者标签符合某个原生组件的名称,可以确定该开发者标签为原生组件标签,需要进行同层渲染。例如,原生组件可以包括Camera(相机)、ARCamera(增强现实相机)、coverview(覆盖视图)、coverimage(覆盖图像)、animationview(动画视图)、button(按钮)、live(直播)、webview(网页视图)、input(单行输入框)、textarea(多行输入框)、video(视频)、map(地图)、audio(音频)、canvas(画布)等组件。例如,如果前端解析得到开发者标签名为video,则该开发者标签是一种用户实现视频播放的原生组件标签。

[0091] 原生组件标签的参数信息可以包括多种,不同原生组件标签的参数信息可能不同。这些参数信息的示例如下:

[0092] 原生组件为视频组件,视频组件标签的参数信息包括资源地址、播放方式和控件显示方式中的至少之一。例如:src表示要播放视频的资源地址,controls表示是否显示默认播放控件(播放/暂停按钮、播放进度、时间),autoplay表示是否自动播放,loop表示是否循环播放等。

[0093] 原生组件为相机组件,相机组件标签的参数信息包括应用模式、摄像头朝向、闪光灯开启方式、触发方式和帧数据尺寸中的至少之一。例如,Camera组件的mode表示应用模式,device-position表示摄像头朝向,flash表示闪光灯开启方式(值为auto是自动,on是打开,off是关闭),bindinitdone表示相机初始化完成时触发,frame-size表示相机帧数据尺寸。

[0094] 原生组件为增强现实相机组件,增强现实相机组件标签的参数信息包括增强现实项目标识、增强现实相机类型、闪光灯开启方式和触发方式中的至少之一。例如,ARCamera组件的key表示AR项目唯一标识,type表示AR相机类型,flash表示闪光灯开启方式,bindload表示AR加载成功时触发。

[0095] 原生组件为画布组件,画布组件标签的参数信息包括画布类型、画布标识、触摸动作响应方式和事件触发方式中的至少之一。例如,canvas组件的type表示指定画布类型,canvas-id表示画布组件的唯一标识符,bindtouchstart表示手指触摸动作开始,bindlongtap表示手指长按之后触发。

[0096] 原生组件为地图组件,地图组件标签的参数信息包括经纬度、缩放级别、路线、显示方式和触发方式中的至少之一。例如:map组件的longitude表示经度,latitude表示纬

度, polyline表示路线, show-compass表示显示指南针, bindtap表示点击地图时触发, bindlabeltap表示点击标签时触发。

[0097] 原生组件为音频组件, 音频组件标签的参数信息包括音频组件标识、资源地址、播放方式、控件显示方式和事件触发方式中的至少之一。例如: audio组件的id表示音频组件的唯一标识符, src表示要播放音频的资源地址, loop表示是否循环播放, controls表示是否显示默认控件, bindpause表示当暂停播放时触发pause事件。

[0098] 原生组件为直播组件, 直播组件标签的参数信息包括资源地址、模式、播放方式、显示方式和事件触发方式中的至少之一。例如: live-player (实时音视频播放) 组件的src表示音视频地址, mode表示模式(默认值可以为live), autoplay表示是否自动播放, muted表示是否静音, orientation表示画面方向, bindstatechange表示播放状态变化事件。再如: live-pusher (实时音视频录制) 组件的url表示推流地址, mode表示模式(默认值可以为RTC), autopush表示是否自动推流, muted表示是否静音, orientation表示画面方向, bindbgmcomplete表示背景音播放完成时触发。

[0099] 原生组件为输入框组件, 输入框组件标签的参数信息包括输入框的内容、样式、控件显示方式和事件触发方式中的至少之一。例如: textarea组件的value表示输入框的内容, placeholder-style表示样式, adjust-position表示键盘弹起时是否自动上推页面, bindfocus表示输入框聚焦时触发。再如: input组件的value表示输入框的初始内容, type表示输入的类型(例如默认为文本类型text), placeholder-style表示样式, adjust-position表示键盘弹起时是否自动上推页面, bindfocus表示输入框聚焦时触发。

[0100] 本申请实施例中, 对于原生组件以及参数信息的列举仅是示例, 在实际应用中, 具有更多类型的原生组件, 不同原生组件标签也可以具有更多的参数信息, 在本申请中不做具体限定。

[0101] 本申请实施例通过为原生组件标签添加元素例如同层渲染容器占位标签, 利用该同层渲染容器占位标签生成滚动视图, 将利用占位标签生成滚动视图作为原生组件的父视图, 能够将原生组件的视图嵌套到滚动视图中, 实现同层渲染。这样, 能够使得原生组件的视图不遮挡其他网页组件视图, 并且能够随滚动视图联动。

[0102] 在一种实施方式中, 该同层渲染方法还包括: 接收该前端为原生组件标签添加的元素对应的标识; 查找该标识对应的滚动视图。利用为原生组件标签添加的元素对应的标识能够快速查找到滚动视图, 能够提高同层渲染的处理速度和效率。

[0103] 在一种示例中, 前端为原生组件标签添加的元素可以包括HTML元素, HTML元素可以包括同层渲染容器占位标签, 该标签对应的标识可以为容器标识。

[0104] 在S11之前, 客户端可以接收来自前端的同层渲染容器占位标签对应的容器标识, 同层渲染容器占位标签为前端为需要同层渲染的原生组件标签添加的标签。查找容器标识对应的滚动视图。在成功找到滚动视图的情况下, 在步骤S11中, 客户端可以接收来自前端的父视图标识和原生组件标签的参数信息, 父视图标识的赋值为容器标识。然后在步骤S12中, 根据原生组件标签的参数信息创建原生组件。在步骤S13中, 将原生组件的视图添加到父视图标识对应的滚动视图中。

[0105] 在一个示例中, 可以在原生组件标签的预留位置, 插入同层渲染容器占位标签。例如, 所插入的同层渲染容器占位标签包括两个DIV (division, 划分) 标签。第一个DIV标签具

有溢出(overflow)属性,第二个DIV标签的高度大于第一个DIV标签。DIV标签,还可以称为区隔标签,可以设定字、画、表格等的摆放位置。overflow属性可以实现当内容溢出网页的元素框(例如网页中展现的某个组件视图)时,提供一种滚动机制,使得溢出元素框的内容被裁掉,并能够随DIV标签对应的滚动视图的滑动而将被裁掉的内容在元素框中显示。

[0106] 在前端添加同层渲染容器占位标签后,操作系统的网页渲染组件例如iOS系统的WKWebView可以根据所添加的同层渲染容器占位标签生成滚动视图,将滚动视图保存到某个位置,并可以获取该滚动视图的内存地址。然后,将该滚动视图发送给客户端,客户端可以在容器池中保存容器标识和滚动视图的内存地址的对应关系。

[0107] 前端可以通过调用插入容器的API(Application Programming Interface,应用程序编程接口)例如insertContainer通知客户端同层渲染容器占位标签对应的容器标识。客户端收到某个占位标签对应的容器标识后,可以在容器池中查找是否存在该容器标识对应的滚动视图。如果有,客户端可以向前端返回成功找到滚动视图的结果。在前端收到成功找到滚动视图的结果后,可以通过调用创建组件的API将赋值为该容器标识的父视图标识以及原生组件标签的参数信息发送给客户端。客户端在收到来自前端的赋值为该容器标识的父视图标识,以及原生组件标签的参数信息后,可以创建原生组件,还可以为所创建的原生组件添加同层渲染标识。并查找该父视图标识(即该容器标识)对应的滚动视图。然后,将所创建的原生组件的视图添加到父视图标识对应的滚动视图中。

[0108] 在一种实施方式中,该同层渲染方法还包括:

[0109] 接收该元素对应的滚动视图的参数。例如:接收同层渲染容器占位标签对应的滚动视图的参数。根据接收到的滚动视图的参数校验容器标识对应的滚动视图的参数。在查找到的滚动视图的参数与接收到的滚动视图的参数一致的情况下,向前端返回成功找到滚动视图的结果。通过为原生组件标签添加的元素对应的标识快速找到滚动视图后,通过滚动视图的位置、大小等参数对找到的滚动视图进行校验,有利于找到准确的滚动视图,提高客户端处理的准确性。

[0110] 例如,前端可以通过调用插入容器的API例如insertContainer通知客户端同层渲染容器占位标签对应的容器标识的同时,还可以通知客户端该同层渲染容器占位标签对应的滚动视图的参数例如位置、大小等。客户端收到某个占位标签对应的容器标识和滚动视图的参数后,可以在容器池中查找是否存在该容器标识对应的滚动视图。如果有,可以再通过收到的滚动视图的参数校验查找到的容器标识对应的滚动视图的参数。如果二者一致,客户端可以向前端返回成功找到滚动视图的结果。在前端收到成功找到滚动视图的结果后,可以通过调用创建组件的API将赋值为该容器标识的父视图标识以及原生组件标签的参数信息发送给客户端。客户端在收到来自前端的赋值为该容器标识的父视图标识,以及原生组件标签的参数信息后,可以创建原生组件,还可以为所创建的原生组件添加同层渲染标识。并查找该父视图标识(即该容器标识)对应的滚动视图。然后,将所创建的原生组件的视图添加到父视图标识对应的滚动视图中。

[0111] 如果查找到的滚动视图的参数与收到的滚动视图的参数不一致,可以向前端返回未找到滚动视图的结果或不返回结果。

[0112] 此外,在前端收到未找到滚动视图的结果后或经过一定时长未收到成功找到滚动视图的结果,也可以再次调用插入容器的API,直到调用次数大于或等于预先设置循环调用

次数为止。

[0113] 在一种实施方式中,该同层渲染方法还包括:保存该标识与利用该元素生成的滚动视图的对应关系。

[0114] 在一种实施方式中,保存该标识与利用该元素生成的滚动视图的对应关系,包括:

[0115] 在利用该元素生成的滚动视图被添加到视图树时,触发视图移动到窗口方法:获取所生成的滚动视图的父视图,从该父视图中解析得到该标识,并保存该标识和利用该元素生成的滚动视图的内存地址的对应关系。在后续可以利用该标识快速查找是否有对应的滚动视图。

[0116] 在一种实施方式中,客户端在监听到网页渲染组件将根据同层渲染容器占位标签生成的滚动视图添加到视图树时,触发视图移动到窗口方法:获取所生成的滚动视图的父视图,从父视图中解析得到容器标识,并在容器池中保存容器标识和所生成的滚动视图的内存地址的对应关系。

[0117] 例如,客户端在启动时可以hook(挂钩)滚动视图例如UIScrollView(iOS12.2以下)和WKChildScrollView(iOS12.2及以上),替换视图移动到窗口(didMoveToWindow)方法实现。在渲染时,网页渲染组件例如WKWebView根据前端插入的同层渲染容器占位标签生成滚动视图。客户端监听到WKWebView生成滚动视图且将该滚动视图加入到视图树时,生成attach事件,并触发didMoveToWindow方法。

[0118] 前端收到占位标签的attach事件时,可以调用客户端的插入容器接口例如insertContainer.insertContainer是一种API。前端可以通过调用insertContainer向客户端传递各种参数,例如向客户端发送insertContainer调用指示以告知客户端滚动视图位置、大小和containerID(容器标识)等。本申请实施例通过为需要同层渲染的原生组件标签添加同层渲染容器占位标签,能够将原生组件的视图嵌套到滚动视图中,实现同层渲染。这样,能够使得原生组件的视图不遮挡其他网页组件视图,并且能够随滚动视图联动。利用视图移动到窗口方法可以在监听到将滚动视图添加到视图树时,从滚动视图的父视图中得到容器标识,并保存容器标识和滚动视图的内存地址的对应关系,从而在后续可以利用容器标识快速查找是否有对应的滚动视图。通过容器标识可以快速找到滚动视图,有利于提高客户端处理速度。通过滚动视图的位置、大小等参数对找到的滚动视图进行校验,有利于找到准确的滚动视图,提高客户端处理的准确性。

[0119] 在一种实施方式中,该同层渲染方法还包括回流(reflow)过程,可以支持在元素布局变化等情况下启动回流。在该元素发生回流的情况下,更新该标识与利用该元素重新生成的滚动视图的对应关系。其中,更新该标识与利用该元素重新生成的滚动视图的对应关系,包括:

[0120] 在利用该元素重新生成的滚动视图被添加到视图树时,触发视图移动到窗口方法:获取重新生成的滚动视图的父视图,从该父视图中解析得到该标识,更新该标识和利用该元素重新生成的滚动视图的内存地址的对应关系。

[0121] 例如,在页面的DOM(Document Object Model,文档对象模型)元素由于窗口大小改变、文字大小改变、内容改变等导致发生回流的情况下,在监听到网页渲染组件将重新生成的滚动视图添加到视图树时,触发视图移动到窗口方法:获取重新生成的滚动视图的父视图,从父视图中解析得到容器标识,在容器池中更新容器标识和重新生成的滚动视图的

内存地址的对应关系。

[0122] 在一种实施方式中,该同层渲染方法还包括重绘过程,支持在元素大小变化等情况下启动重绘。在该元素发生重绘的情况下,接收该标识和更新的滚动视图参数;查找该标识对应的滚动视图;在查找到的滚动视图的参数与接收到的滚动视图的参数一致的情况下,向该前端返回成功找到滚动视图的结果;根据重绘后的元素更新原生组件。

[0123] 例如,在页面的DOM元素由于背景、文字、边框等颜色改变导致发生重绘的情况下,接收更新容器接口调用指示,更新容器接口调用指示中包括容器标识和更新的滚动视图参数;在容器池中查找容器标识对应的滚动视图;在查找到的滚动视图的参数与接收到的滚动视图的参数一致的情况下,向前端返回成功找到滚动视图的结果;根据重绘后的DOM元素更新组件池中的原生组件。

[0124] 在一种实施方式中,该同层渲染方法还包括事件处理过程,支持原生组件返回需要响应触摸事件的视图。例如,事件处理过程具体可以包括:

[0125] 获取网页渲染组件根据触摸事件计算的第一视图;

[0126] 重写网页渲染组件的寻找响应事件视图的方法,以将网页渲染组件的触摸事件转发给组件容器,由组件容器将触摸事件派发给组件容器上的所有原生组件,由各原生组件判断是否需要响应触摸事件。

[0127] 其中,原生组件判断是否需要响应触摸事件可以包括:查找原生组件是否具有同层渲染标识,如果有,则确定原生组件被设置为同层渲染;在原生组件被设置为同层渲染且原生组件的父视图为在视图树中滚动视图的情况下,调用父视图的寻找响应事件视图的方法;

[0128] 在接收到一个原生组件返回的需要响应触摸事件的第二视图的情况下,利用第二视图替换第一视图。

[0129] 图2是根据本申请实施例同层渲染方法的示意图。如图2所示,该同层渲染方法可以应用于前端,该同层渲染方法可以包括:

[0130] 步骤S21、向客户端发送原生组件标签添加的元素对应的标识和该原生组件标签的参数信息,以使得该客户端根据该原生组件标签的参数信息创建原生组件、并将该原生组件的视图添加到该标识对应的滚动视图中。

[0131] 在一种实施方式中,该同层渲染方法还包括:向客户端发送为原生组件标签添加的元素对应的标识,以使得该客户端查找该标识对应的滚动视图。

[0132] 在一种示例中,前端为原生组件标签添加的元素可以包括HTML元素,HTML元素可以包括同层渲染容器占位标签,该标签对应的标识可以为容器标识。

[0133] 在步骤S21之前,前端可以向客户端发送同层渲染容器占位标签对应的容器标识,以使得客户端查找容器标识对应的滚动视图,同层渲染容器占位标签为前端为需要同层渲染的原生组件标签添加的标签。

[0134] 在确定客户端成功找到滚动视图的情况下例如收到客户端返回的成功找到滚动视图的结果,在步骤S21中,前端可以向客户端发送父视图标识和原生组件标签的参数信息,并且父视图标识的赋值为容器标识,以使得客户端根据原生组件标签的参数信息创建原生组件、并将原生组件的视图添加到父视图标识对应的滚动视图中。

[0135] 在一种实施方式中,该同层渲染方法还包括:

[0136] 向客户端发送该元素对应的滚动视图的参数,以使得该客户端根据接收到的滚动视图的参数校验该标识对应的滚动视图的参数。

[0137] 例如,前端向客户端发送同层渲染容器占位标签对应的滚动视图的参数,以使得客户端根据接收到的滚动视图的参数校验容器标识对应的滚动视图的参数。

[0138] 在一种实施方式中,该同层渲染方法还包括:

[0139] 在确定原生组件标签需要同层渲染的情况下,为该原生组件标签添加该元素。例如,在确定原生组件标签需要同层渲染的情况下,前端为原生组件标签添加同层渲染容器占位标签,以使得网页渲染组件在渲染时根据同层渲染容器占位标签生成滚动视图并将所生成的滚动视图添加到视图树。

[0140] 在本实施例中,前端解析开发者标签,如果确定该标签是原生组件标签,需要进行同层渲染。在原生组件标签的预留位置,插入同层渲染容器占位标签。然后,操作系统的网页渲染组件可以根据所添加的同层渲染容器占位标签生成滚动视图,将滚动视图保存到某个位置,并可以获取该滚动视图的内存地址。

[0141] 客户端在监听到网页渲染组件将根据同层渲染容器占位标签生成的滚动视图添加到视图树时,生成添加事件,并触发视图移动到窗口方法:获取网页渲染组件生成的滚动视图的父视图,从父视图中解析得到容器标识,并在容器池中保存容器标识和所生成的滚动视图的内存地址的对应关系。

[0142] 前端收到占位标签的添加事件时,可以调用客户端的插入容器的API,向客户端传递占位标签的滚动视图参数和容器标识。客户端根据容器标识在容器池中查找对应的滚动视图。并利用收到滚动视图参数与查找到的滚动视图参数进行比较,以确定是否成功找到了利用占位标签生成的滚动视图。如果前端收到的调用结果为成功找到滚动视图,可以调用创建组件的API,向客户端传递父视图标识和原生组件标签的参数信息。

[0143] 本实施例中原生组件的类型可以包括多种,原生组件标签的参数信息包括示例如下:

[0144] 该原生组件为视频组件,该视频组件标签的参数信息包括资源地址、播放方式和控件显示方式中的至少之一;

[0145] 该原生组件为相机组件,该相机组件标签的参数信息包括应用模式、摄像头朝向、闪光灯开启方式、触发方式和帧数据尺寸中的至少之一;

[0146] 原生组件为画布组件,画布组件标签的参数信息包括画布类型、画布标识、触摸动作响应方式和事件触发方式中的至少之一;

[0147] 原生组件为地图组件,地图组件标签的参数信息包括经纬度、缩放级别、路线、显示方式和触发方式中的至少之一;

[0148] 原生组件为音频组件,音频组件标签的参数信息包括音频组件标识、资源地址、播放方式、控件显示方式和事件触发方式中的至少之一;

[0149] 原生组件为直播组件,直播组件标签的参数信息包括资源地址、模式、播放方式、显示方式和事件触发方式中的至少之一;

[0150] 原生组件为输入框组件,输入框组件标签的参数信息包括输入框的内容、样式、控件显示方式和事件触发方式中的至少之一。

[0151] 原生组件以及参数信息的具体示例可以参见上述同层渲染方法实施例的相关描

述,在此不再赘述。

[0152] 客户端根据原生组件标签的参数信息创建原生组件,为该原生组件添加同层渲染标识、并该原生组件的视图添加到收到的父视图标识对应的滚动视图中。本实施例中与上述同层渲染方法实施例相同的内容具有相同的含义,具体可以参见上述实施例的相关描述,在此不再赘述。

[0153] 在一种应用示例中,如图3所示,以在iOS系统中使用原生组件进行渲染为例,该同层渲染方法包括:

[0154] 步骤S301、前端解析开发者标签时根据当前的开关配置,如果NA组件需要同层渲染,添加同层渲染容器占位标签。

[0155] 步骤S302、客户端hook(挂钩)系统滚动视图例如UIScrollView(iOS12.2以下)和WKChildScrollView(iOS12.2及以上),替换didMoveToWindow(视图移动到窗口)方法实现。UIScrollView或WKChildScrollView可以简称scrollView。

[0156] 步骤S303、当开发者页面开始渲染,步骤S302中的滚动视图被添加到视图树时,客户端会触发didMoveToWindow,并会产生attach(添加)事件。

[0157] 步骤S304、客户端获取当前的滚动视图时,取出滚动视图的父视图,从父视图的layer(层)的name(名称)中解析containerID(容器标识)。通过isScrollViewFound判断是否找到了containerID对应的滚动视图(scrollView),如果是执行步骤S305,否则返回步骤S303。

[0158] 步骤S305、如果有containerID,说明是同层渲染场景。客户端将containerID当做key(键),滚动视图的内存地址当作value(值)保存到容器(container)池里。此外,还可以判断滚动视图的内存地址是否发生改变,如果是,则可以通过页面容器转发消息给组件池里的NA组件,NA组件可以更新父视图。例如,在重绘或者更新场景,如果重新生成了滚动视图,则滚动视图的内存地址可能发生改变。例如,页面容器:slaveWebViewController持有WKWebView渲染开发者页面。页面容器可以维护两个池子:容器(container)池可以管理前端插入的DIV标签生成的滚动视图,NA组件池可以管理NA组件。

[0159] 步骤S306、当前端收到占位标签的attach(添加)事件时,执行步骤S307调用端能力insertContainer通知客户端各种参数,例如参数可以包括滚动视图的位置、大小和containerID。在步骤S303中,触发didMoveToWindow,与产生attach事件没有时序要求。如果先产生attach事件,在步骤S303之后,也可能会直接执行步骤S306。

[0160] 步骤S308、客户端收到insertContainer调用时,先根据containerID在容器池里查找滚动视图,找到后重新换算滚动视图位置、大小等,并根据端能力参数的滚动视图的位置、大小校验查找到的滚动视图的参数是否相同。客户端回调前端,向前端返回是否找到了containerID对应的滚动视图的查找结果。通过containerID可以快速找到滚动视图,有利于提高客户端处理速度。通过滚动视图的位置、大小对找到的滚动视图进行校验,有利于找到准确的滚动视图,提高客户端处理的准确性。

[0161] 步骤S309、前端收到客户端的insertContainer回调时,通过isScrollViewFound判断是否找到了containerID对应的滚动视图。如果客户端没有找到containerID对应的滚动视图,则前端循环调用insertContainer。例如,执行步骤S310判断循环调用次数是否达到设定次数。如果没有,则调用insertContainer,直到客户返回找到滚动视图或达到循环

调用次数。例如，循环调用次数可以设置为至多5次。

[0162] 步骤S311、insertContainer调用结束时，前端会调用客户端的组件创建端能力。如果找到了containerID对应的滚动视图，前端调用客户端的组件创建的API时可以向客户端传递parentID参数，并且parentID值为containerID，没找到则可以不传。

[0163] 步骤S312、客户端收到组件创建的API调用时，如果传了父视图标识parentID（值为containerID），执行步骤S313根据parentID查找占位视图容器（container），执行步骤S314判断parentID是否存在对应的容器（container）。如果存在，则执行步骤S315将NA组件的视图添加到容器的滚动视图中，即实现同层渲染。否则执行步骤S316，根据是否是fix（固定）参数，将NA组件的视图添加到WKWebView或者WKWebView的滚动视图上，即不需要同层渲染。

[0164] 当用户点击NA组件视图或者其子视图时，执行事件处理流程。

[0165] 步骤S317、当开发者修改数据导致页面的DOM元素变化发生回流时，客户端也会触发didMoveToWindow，客户端会更新容器池对应的滚动视图，并更新组件的父视图。当开发者修改数据导致页面的DOM元素变化发送重绘时，执行步骤S318前端通过调用updateContainer告诉客户端。客户端根据传入的参数（例如变化后的滚动视图的背景、文字、边框等颜色和containerID）重新匹配滚动视图（匹配逻辑可以参见步骤S308中对insertContainer的调用），然后重新换算滚动视图的位置大小。此外，参见步骤S309前端收到客户端的回调时，如果客户端没有找到containerID对应的滚动视图，前端对updateContainer也可以循环调用。参见步骤S310如果调用次数超过阈值，例如5次，可以退化到非同层渲染的方式。

[0166] 在一种应用示例中，如图4所示，事件处理流程可以包括：

[0167] 步骤S401、当触摸事件发生后，iOS系统根据寻找响应事件视图的方法（hitTest:withEvent，以下简称hitTest方法）来确定触摸事件发生在哪个视图对象上。WKWebView计算本次需要响应事件的视图view。

[0168] 例如，-(nullable UIView*)hitTest:(CGPoint)point withEvent:(nullable UIEvent*)event。

[0169] 步骤S402、由于NA组件的渲染是在WKWebView的渲染流程之外，hitTest事件响应链到滚动视图例如WKChildScrollView就停止了。NA组件视图无法响应事件。因此，需要重写WKWebView的hitTest方法。

[0170] 步骤S403、客户端重写WKWebView的hitTest方法。

[0171] 步骤S404、客户端将WKWebView的hitTest事件转发给组件容器。

[0172] 步骤S405、组件容器将hitTest事件派发给当前容器上的所有NA组件。

[0173] 步骤S406、如果有组件返回了需要响应事件的视图，则用该视图替换调WKWebView的计算结果view。

[0174] 其中，如图5所示，组件判断是否需要响应事件的流程包括：

[0175] 步骤S501、组件实现hitTest接口，以寻找响应事件视图。

[0176] 步骤S502、判断是否是同层渲染。客户端在创建NA组件时，会在NA组件上添加是否是同层渲染的标识。根据该标识，可以判断该NA组件是否被设置为同层渲染。如果是，执行步骤S503，否则返回无法响应事件的结果例如nil。

[0177] 步骤S503、获取NA组件的父视图、即利用前端插入的占位标签所生成的滚动视图(标记为scrollView)。例如,可以调用系统的Superview(超级视图)方法来获取滚动视图。

[0178] 步骤S504、判断滚动视图是否在视图树中。如果是,则执行步骤S505,否则返回nil。

[0179] 步骤S505、判断滚动视图(scrollView)与view是否相等。如果是则执行步骤S506,否则返回nil。

[0180] 步骤S506、判断NA组件时是否需要显示。如果是,则执行步骤S507,否则返回nil。

[0181] 步骤S507、对滚动视图进行坐标换算。

[0182] 步骤S508、向滚动视图(scrollView)派发hitTest,以调用父视图的hitTest方法。

[0183] 步骤S509、判断滚动视图是否向原生组件返回了需要响应触摸事件的视图hitView。如果是,则执行步骤S510,否则返回nil。

[0184] 步骤S510、NA组件向组件容器返回需要响应触摸事件的视图hitView。

[0185] 采用本申请实施例的同层渲染方法,可以对原生组件进行层级控制。例如直接通过调整组件的z-index属性对原生组件进行层级控制。可以直接使用H5组件代替cover-view(覆盖视图)、cover-image(覆盖图像)组件来覆盖原生组件。可在操作系统内置组件例如:picker-view(拾取视图)、scroll-view(可滚动视图区域)、swiper(轮播)、movable-view(可移动视图)等中使用原生组件。此外,采用本申请的事件处理流程,video组件可以支持touch(触摸)事件。本申请实施例的同层渲染方法具有普适性,除了小程序之外,可以迁移到其它利用渲染组件进行渲染的业务场景,比如搜索落地页、视频落地页等。

[0186] 图6是根据本申请另一实施例同层渲染装置的示意图。如图6所示,该同层渲染装置可以应用于客户端,该同层渲染装置可以包括:

[0187] 第一接收模块61,用于接收前端为原生组件标签添加的元素对应的标识和该原生组件标签的参数信息;

[0188] 创建模块62,用于根据该原生组件标签的参数信息创建原生组件;

[0189] 添加模块63,用于将该原生组件的视图添加到该标识对应的滚动视图中。

[0190] 在一种实施方式中,如图7所示,该同层渲染装置还包括:

[0191] 第二接收模块71,用于接收该前端为原生组件标签添加的元素对应的标识;

[0192] 查找模块72,用于查找该标识对应的滚动视图。

[0193] 在一种实施方式中,该元素包括HTML元素,该HTML元素包括同层渲染容器占位标签,该标识为容器标识。

[0194] 例如,第二接收模块71接收来自前端为原生组件标签添加的同层渲染容器占位标签对应的容器标识。查找模块72查找容器标识对应的滚动视图。在成功找到滚动视图的情况下,第一接收模块61接收来自前端的父视图标识和原生组件标签的参数信息,父视图标识的赋值为容器标识。创建模块62根据原生组件标签的参数信息创建原生组件,添加模块63将原生组件的视图添加到父视图标识对应的滚动视图中。

[0195] 在一种实施方式中,该同层渲染装置还包括:

[0196] 第三接收模块73,用于接收该元素对应的滚动视图的参数;

[0197] 校验模块74,用于根据接收到的滚动视图的参数校验该标识对应的滚动视图的参数;

[0198] 返回模块75,用于在查找到的滚动视图的参数与接收到的滚动视图的参数一致的情况下,向该前端返回成功找到滚动视图的结果。

[0199] 例如第三接收模块73接收同层渲染容器占位标签对应的滚动视图的参数。在查找模块72查找到容器标识对应的滚动视图后,校验模块74可以校验查找到的滚动视图的参数与接收到的滚动视图的参数是否一致。如果一致,向该前端返回成功找到滚动视图的结果。如果不一致,可以向该前端返回未成功找到滚动视图的结果,也可以不返回结果。

[0200] 在一种实施方式中,该同层渲染装置还包括:

[0201] 保存模块76,用于保存该标识与利用该元素生成的滚动视图的对应关系。

[0202] 在一种实施方式中,该保存模块76还用于在利用该元素生成的滚动视图被添加到视图树时,触发视图移动到窗口方法:获取所生成的滚动视图的父视图,从该父视图中解析得到该标识,并保存该标识和利用该元素生成的滚动视图的内存地址的对应关系。

[0203] 例如,保存模块76在监听到网页渲染组件将根据同层渲染容器占位标签生成的滚动视图添加到视图树时,触发视图移动到窗口方法:获取所生成的滚动视图的父视图,从父视图中解析得到容器标识,并在容器池中保存容器标识和所生成的滚动视图的内存地址的对应关系。

[0204] 在一种实施方式中,该同层渲染装置还包括:

[0205] 回流模块77,用于在该元素发生回流的情况下,更新该标识与利用该元素重新生成的滚动视图的对应关系。

[0206] 在一种实施方式中,回流模块77更新该标识与利用该元素重新生成的滚动视图的对应关系,包括:在利用该元素重新生成的滚动视图被添加到视图树时,触发视图移动到窗口方法:获取重新生成的滚动视图的父视图,从该父视图中解析得到该标识,更新该标识和利用该元素重新生成的滚动视图的内存地址的对应关系。

[0207] 例如,回流模块77在页面的DOM元素发生回流的情况下,在监听到网页渲染组件将重新生成的滚动视图添加到视图树时,触发视图移动到窗口方法:获取重新生成的滚动视图的父视图,从父视图中解析得到容器标识,在容器池中更新容器标识和重新生成的滚动视图的内存地址的对应关系。

[0208] 在一种实施方式中,该同层渲染装置还包括:

[0209] 重绘模块78,用于在该元素发生重绘的情况下,接收该标识和更新的滚动视图参数;查找该标识对应的滚动视图;在查找到的滚动视图的参数与接收到的滚动视图的参数一致的情况下,向该前端返回成功找到滚动视图的结果;根据重绘后的元素更新原生组件。

[0210] 例如,重绘模块78在页面的DOM元素发生更新的情况下,接收更新容器接口调用指示,更新容器接口调用指示中包括容器标识和更新的滚动视图参数;在容器池中查找容器标识对应的滚动视图;在查找到的滚动视图的参数与接收到的滚动视图的参数一致的情况下,向前端返回成功找到滚动视图的结果;根据重绘后的DOM元素更新组件池中的原生组件。

[0211] 该原生组件标签的参数信息包括以下至少之一:

[0212] 该原生组件为视频组件,该视频组件标签的参数信息包括资源地址、播放方式和控件显示方式中的至少之一;

[0213] 原生组件为相机组件,相机组件标签的参数信息包括应用模式、摄像头朝向、闪光

灯开启方式、触发方式和帧数据尺寸中的至少之一；

[0214] 原生组件为画布组件，画布组件标签的参数信息包括画布类型、画布标识、触摸动作响应方式和事件触发方式中的至少之一；

[0215] 原生组件为地图组件，地图组件标签的参数信息包括经纬度、缩放级别、路线、显示方式和触发方式中的至少之一；

[0216] 原生组件为音频组件，音频组件标签的参数信息包括音频组件标识、资源地址、播放方式、控件显示方式和事件触发方式中的至少之一；

[0217] 原生组件为直播组件，直播组件标签的参数信息包括资源地址、模式、播放方式、显示方式和事件触发方式中的至少之一；

[0218] 原生组件为输入框组件，输入框组件标签的参数信息包括输入框的内容、样式、控件显示方式和事件触发方式中的至少之一。

[0219] 在一种实施方式中，该同层渲染装置还包括：

[0220] 事件处理模块79，用于获取网页渲染组件根据触摸事件计算的第一视图；重写网页渲染组件的寻找响应事件视图的方法，以将网页渲染组件的触摸事件转发给组件容器，由组件容器将触摸事件派发给组件容器上的所有原生组件，由各原生组件判断是否需要响应触摸事件。

[0221] 其中，原生组件判断是否需要响应触摸事件包括：查找原生组件是否具有同层渲染标识，如果有，则确定原生组件被设置为同层渲染；在原生组件被设置为同层渲染且原生组件的父视图为在视图树中滚动视图的情况下，调用父视图的寻找响应事件视图的方法；在接收到一个原生组件返回的需要响应触摸事件的第二视图的情况下，利用第二视图替换第一视图。

[0222] 图8是根据本申请另一实施例同层渲染装置的示意图。如图8所示，该同层渲染装置可以应用于前端，该同层渲染装置可以包括：

[0223] 第一发送模块81，用于向客户端发送原生组件标签添加的元素对应的标识和该原生组件标签的参数信息，以使得该客户端根据该原生组件标签的参数信息创建原生组件、并将该原生组件的视图添加到该标识对应的滚动视图中。

[0224] 在一种实施方式中，如图9所示，该同层渲染装置还包括：

[0225] 第二发送模块91，用于向客户端发送为原生组件标签添加的元素对应的标识，以使得该客户端查找该标识对应的滚动视图。

[0226] 在一种实施方式中，上述元素包括HTML元素，HTML元素包括同层渲染容器占位标签，上述标识为容器标识。

[0227] 例如，第二发送模块91向客户端发送同层渲染容器占位标签对应的容器标识，以使得客户端查找容器标识对应的滚动视图，同层渲染容器占位标签为前端为需要同层渲染的原生组件标签添加的标签。第一发送模块81在确定客户端成功找到滚动视图的情况下，向客户端发送父视图标识和原生组件标签的参数信息，并且父视图标识赋值为容器标识，以使得客户端根据原生组件标签的参数信息创建原生组件、并将原生组件的视图添加到父视图标识对应的滚动视图中。

[0228] 在一种实施方式中，该同层渲染装置还包括：

[0229] 第三发送模块92，用于向客户端发送该元素对应的滚动视图的参数，以使得该客

户端根据接收到的滚动视图的参数校验该标识对应的滚动视图的参数。

[0230] 例如,第三发送模块92向客户端发送同层渲染容器占位标签对应的滚动视图的参数,以使得客户端根据接收到的滚动视图的参数校验容器标识对应的滚动视图的参数。

[0231] 在一种实施方式中,该同层渲染装置还包括:

[0232] 添加模块93,用于在确定原生组件标签需要同层渲染的情况下,为该原生组件标签添加该元素。

[0233] 例如,添加模块93在确定原生组件标签需要同层渲染的情况下,为原生组件标签添加同层渲染容器占位标签,以使得网页渲染组件在渲染时根据同层渲染容器占位标签生成滚动视图并将所生成的滚动视图添加到视图树。客户端在监听到网页渲染组件将根据同层渲染容器占位标签生成的滚动视图添加到视图树时,触发视图移动到窗口方法:获取所生成的滚动视图的父视图,从父视图中解析得到容器标识,并在容器池中保存容器标识和所生成的滚动视图的内存地址的对应关系。前端收到占位标签的添加事件时,调用插入容器的API,通知客户端各种参数,例如第二发送模块91向客户端发送容器标识,第三发送模块92向客户端发送滚动视图的参数。客户端根据容器标识查找是否已经利用占位标签生成了滚动视图,如果是,再利用滚动视图的参数校验。校验成功后,客户端向前端返回成功找到滚动视图的结果。前端调用创建组件的API,第一发送模块81向客户端发送赋值为容器标识的父视图标识以及原生组件标签的参数信息。客户端根据原生组件标签的参数信息创建原生组件,为原生组件添加同层渲染标识,查找父视图标识对应的滚动视图,并将原生组件的视图添加到父视图标识对应的滚动视图中。

[0234] 在一种实施方式中,该原生组件标签的参数信息包括以下至少之一:

[0235] 该原生组件为视频组件,该视频组件标签的参数信息包括资源地址、播放方式和控件显示方式中的至少之一;

[0236] 该原生组件为相机组件,该相机组件标签的参数信息包括应用模式、摄像头朝向、闪光灯开启方式、触发方式和帧数据尺寸中的至少之一;

[0237] 该原生组件为画布组件,该画布组件标签的参数信息包括画布类型、画布标识、触摸动作响应方式和事件触发方式中的至少之一;

[0238] 该原生组件为地图组件,该地图组件标签的参数信息包括经纬度、缩放级别、路线、显示方式和触发方式中的至少之一;

[0239] 该原生组件为音频组件,该音频组件标签的参数信息包括音频组件标识、资源地址、播放方式、控件显示方式和事件触发方式中的至少之一;

[0240] 该原生组件为直播组件,该直播组件标签的参数信息包括资源地址、模式、播放方式、显示方式和事件触发方式中的至少之一;

[0241] 该原生组件为输入框组件,该输入框组件标签的参数信息包括输入框的内容、样式、控件显示方式和事件触发方式中的至少之一。

[0242] 本发明实施例各装置中的各模块的功能可以参见上述方法中的对应描述,在此不再赘述。

[0243] 根据本申请的实施例,本申请还提供了一种电子设备和一种可读存储介质。

[0244] 如图10所示,是根据本申请实施例的同层渲染方法的电子设备的框图。电子设备旨在表示各种形式的数字计算机,诸如,膝上型计算机、台式计算机、工作台、个人数字助

理、服务器、刀片式服务器、大型计算机、和其它适合的计算机。电子设备还可以表示各种形式的移动装置,诸如,个人数字处理、蜂窝电话、智能电话、可穿戴设备和其它类似的计算装置。本文所示的部件、它们的连接和关系、以及它们的功能仅仅作为示例,并且不意在限制本文中描述的和/或者要求的本申请的实现。

[0245] 如图10所示,该电子设备包括:一个或多个处理器901、存储器902,以及用于连接各部件的接口,包括高速接口和低速接口。各个部件利用不同的总线互相连接,并且可以被安装在公共主板上或者根据需要以其它方式安装。处理器可以对在电子设备内执行的指令进行处理,包括存储在存储器中或者存储器上以在外部输入/输出装置(诸如,耦合至接口的显示设备)上显示图形用户界面(Graphical User Interface, GUI)的图形信息的指令。在其它实施方式中,若需要,可以将多个处理器和/或多条总线与多个存储器和多个存储器一起使用。同样,可以连接多个电子设备,各个设备提供部分必要的操作(例如,作为服务器阵列、一组刀片式服务器、或者多处理器系统)。图10中以一个处理器901为例。

[0246] 存储器902即为本申请所提供的非瞬时计算机可读存储介质。其中,存储器存储有可由至少一个处理器执行的指令,以使至少一个处理器执行本申请所提供的同层渲染方法。本申请的非瞬时计算机可读存储介质存储计算机指令,该计算机指令用于使计算机执行本申请所提供的同层渲染方法。

[0247] 存储器902作为一种非瞬时计算机可读存储介质,可用于存储非瞬时软件程序、非瞬时计算机可执行程序以及模块,如本申请实施例中的同层渲染方法对应的程序指令/模块(例如,附图6所示的第一接收模块61、创建模块62和添加模块63。再如,附图8所示的第一发送模块81)。处理器901通过运行存储在存储器902中的非瞬时软件程序、指令以及模块,从而执行服务器的各种功能应用以及数据处理,即实现上述方法实施例中的同层渲染方法。

[0248] 存储器902可以包括存储程序区和存储数据区,其中,存储程序区可存储操作系统、至少一个功能所需要的应用程序;存储数据区可存储根据录音的电子设备的创建的数据等。此外,存储器902可以包括高速随机存取存储器,还可以包括非瞬时存储器,例如至少一个磁盘存储器件、闪存器件、或其他非瞬时固态存储器件。在一些实施例中,存储器902可选包括相对于处理器901远程设置的存储器,这些远程存储器可以通过网络连接至录音的电子设备的。上述网络的实例包括但不限于互联网、企业内部网、局域网、移动通信网及其组合。

[0249] 同层渲染方法的电子设备还可以包括:输入装置903和输出装置904。处理器901、存储器902、输入装置903和输出装置904可以通过总线或者其他方式连接,图10中以通过总线连接为例。

[0250] 输入装置903可接收输入的数字或字符信息,以及产生与录音的电子设备的用户设置以及功能控制有关的键信号输入,例如触摸屏、小键盘、鼠标、轨迹板、触摸板、指示杆、一个或者多个鼠标按钮、轨迹球、操纵杆等输入装置。输出装置904可以包括显示设备、辅助照明装置(例如,LED)和触觉反馈装置(例如,振动电机)等。该显示设备可以包括但不限于,液晶显示器(Liquid Crystal Display, LCD)、发光二极管(Light Emitting Diode, LED)显示器和等离子体显示器。在一些实施方式中,显示设备可以是触摸屏。

[0251] 此处描述的系统和技术各种实施方式可以在数字电子电路系统、集成电路系

统、专用集成电路 (Application Specific Integrated Circuits, ASIC)、计算机硬件、固件、软件、和/或它们的组合中实现。这些各种实施方式可以包括:实施在一个或者多个计算机程序中,该一个或者多个计算机程序可在包括至少一个可编程处理器的可编程系统上执行和/或解释,该可编程处理器可以是专用或者通用可编程处理器,可以从存储系统、至少一个输入装置、和至少一个输出装置接收数据和指令,并且将数据和指令传输至该存储系统、该至少一个输入装置、和该至少一个输出装置。

[0252] 这些计算程序(也称作程序、软件、软件应用、或者代码)包括可编程处理器的机器指令,并且可以利用高级过程和/或面向对象的编程语言、和/或汇编/机器语言来实施这些计算程序。如本文使用的,术语“机器可读介质”和“计算机可读介质”指的是用于将机器指令和/或数据提供给可编程处理器的任何计算机程序产品、设备、和/或装置(例如,磁盘、光盘、存储器、可编程逻辑装置(programmable logic device, PLD)),包括,接收作为机器可读信号的机器指令的机器可读介质。术语“机器可读信号”指的是用于将机器指令和/或数据提供给可编程处理器的任何信号。

[0253] 为了提供与用户的交互,可以在计算机上实施此处描述的系统和技术,该计算机具有:用于向用户显示信息的显示装置(例如,CRT (Cathode Ray Tube, 阴极射线管)或者LCD(液晶显示器)监视器);以及键盘和指向装置(例如,鼠标或者轨迹球),用户可以通过该键盘和该指向装置来将输入提供给计算机。其它种类的装置还可以用于提供与用户的交互;例如,提供给用户的反馈可以是任何形式的传感反馈(例如,视觉反馈、听觉反馈、或者触觉反馈);并且可以用任何形式(包括声输入、语音输入或者、触觉输入)来接收来自用户的输入。

[0254] 可以将此处描述的系统和技术实施在包括后台部件的计算系统(例如,作为数据服务器)、或者包括中间件部件的计算系统(例如,应用服务器)、或者包括前端部件的计算系统(例如,具有图形用户界面或者网络浏览器的用户计算机,用户可以通过该图形用户界面或者该网络浏览器来与此处描述的系统和技术实施方式交互)、或者包括这种后台部件、中间件部件、或者前端部件的任何组合的计算系统中。可以通过任何形式或者介质的数字数据通信(例如,通信网络)来将系统的部件相互连接。通信网络的示例包括:局域网(Local Area Network, LAN)、广域网(Wide Area Network, WAN)和互联网。

[0255] 计算机系统可以包括客户端和服务端。客户端和服务端一般远离彼此并且通常通过通信网络进行交互。通过在相应的计算机上运行并且彼此具有客户端-服务器关系的计算机程序来产生客户端和服务端的关系。

[0256] 根据本申请实施例的技术方案,通过为需要同层渲染的原生组件标签添加同层渲染容器占位标签,能够将原生组件的视图嵌套到滚动视图中,实现同层渲染。这样,能够使得原生组件的视图不遮挡滚动视图上的其他网页组件视图,并且能够随滚动视图联动。利用视图移动到窗口方法可以在监听到将滚动视图添加到视图树时,从滚动视图的父视图中得到容器标识,并保存容器标识和滚动视图的内存地址的对应关系,从而在后续可以利用容器标识快速查找是否有对应的滚动视图。通过容器标识可以快速找到滚动视图,有利于提高客户端处理速度。通过滚动视图的位置、大小等参数对找到的滚动视图进行校验,有利于找到准确的滚动视图,提高客户端处理的准确性。此外,还支持回流、重绘和原生组件返回需要响应触摸事件的视图。

[0257] 应该理解,可以使用上面所示的各种形式的流程,重新排序、增加或删除步骤。例如,本发申请中记载的各步骤可以并行地执行也可以顺序地执行也可以不同的次序执行,只要能够实现本申请公开的技术方案所期望的结果,本文在此不进行限制。

[0258] 上述具体实施方式,并不构成对本申请保护范围的限制。本领域技术人员应该明白的是,根据设计要求和因素,可以进行各种修改、组合、子组合和替代。任何在本申请的精神和原则之内所作的修改、等同替换和改进等,均应包含在本申请保护范围之内。

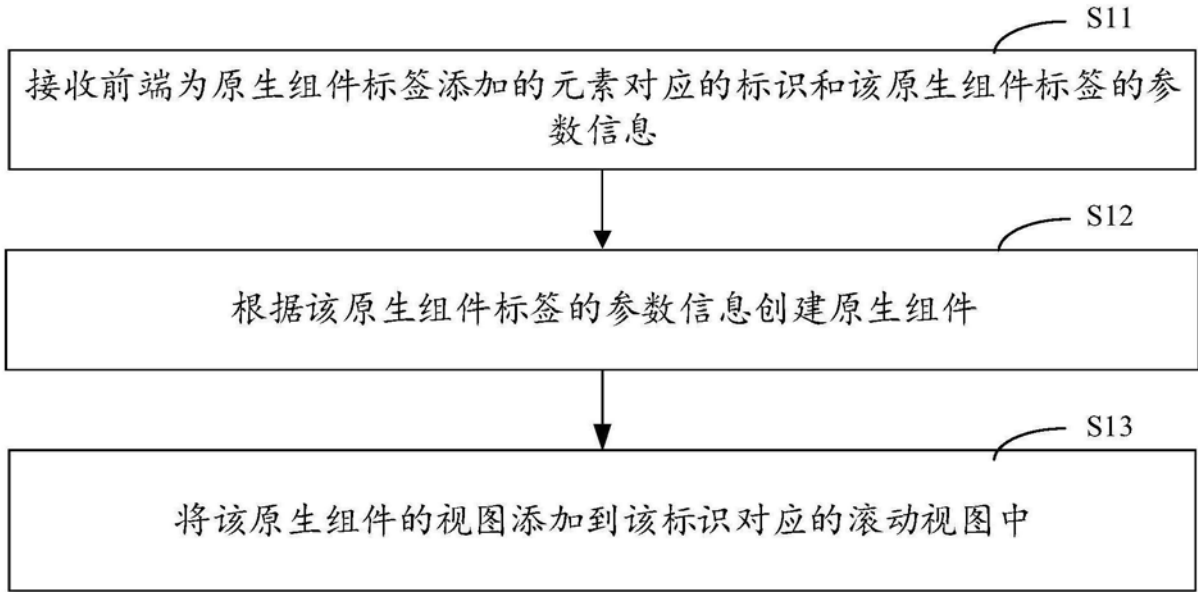


图1

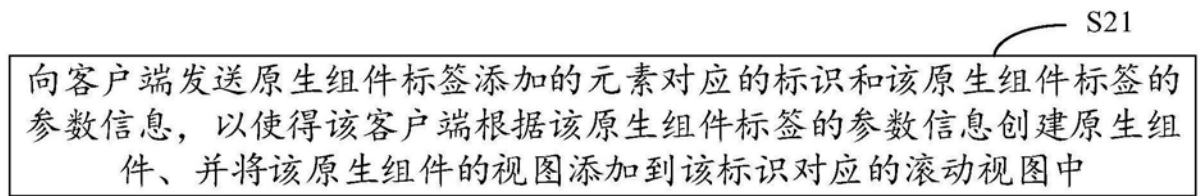


图2

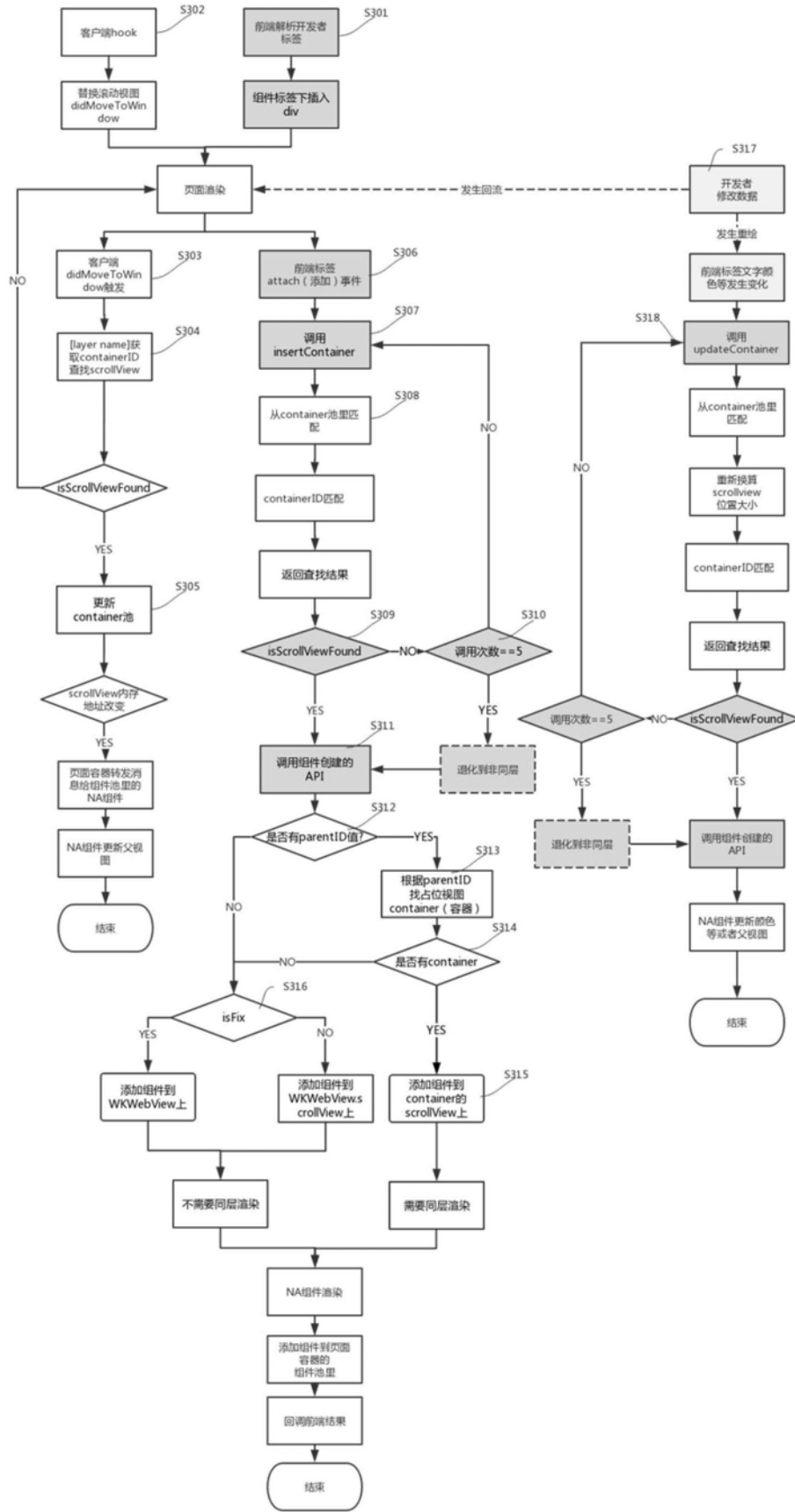


图3

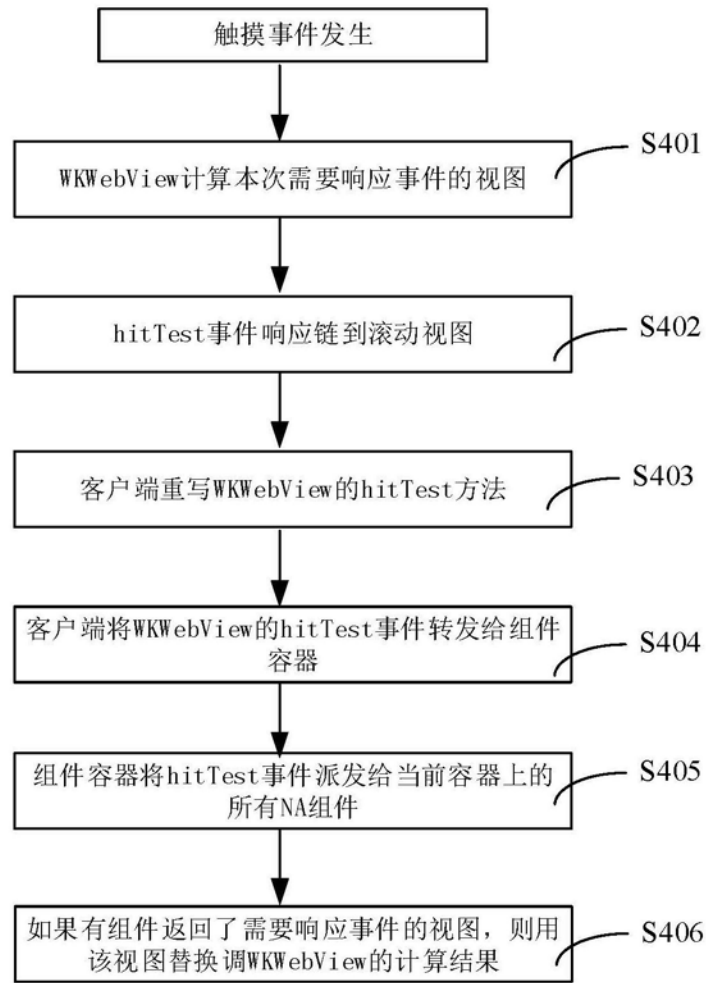


图4

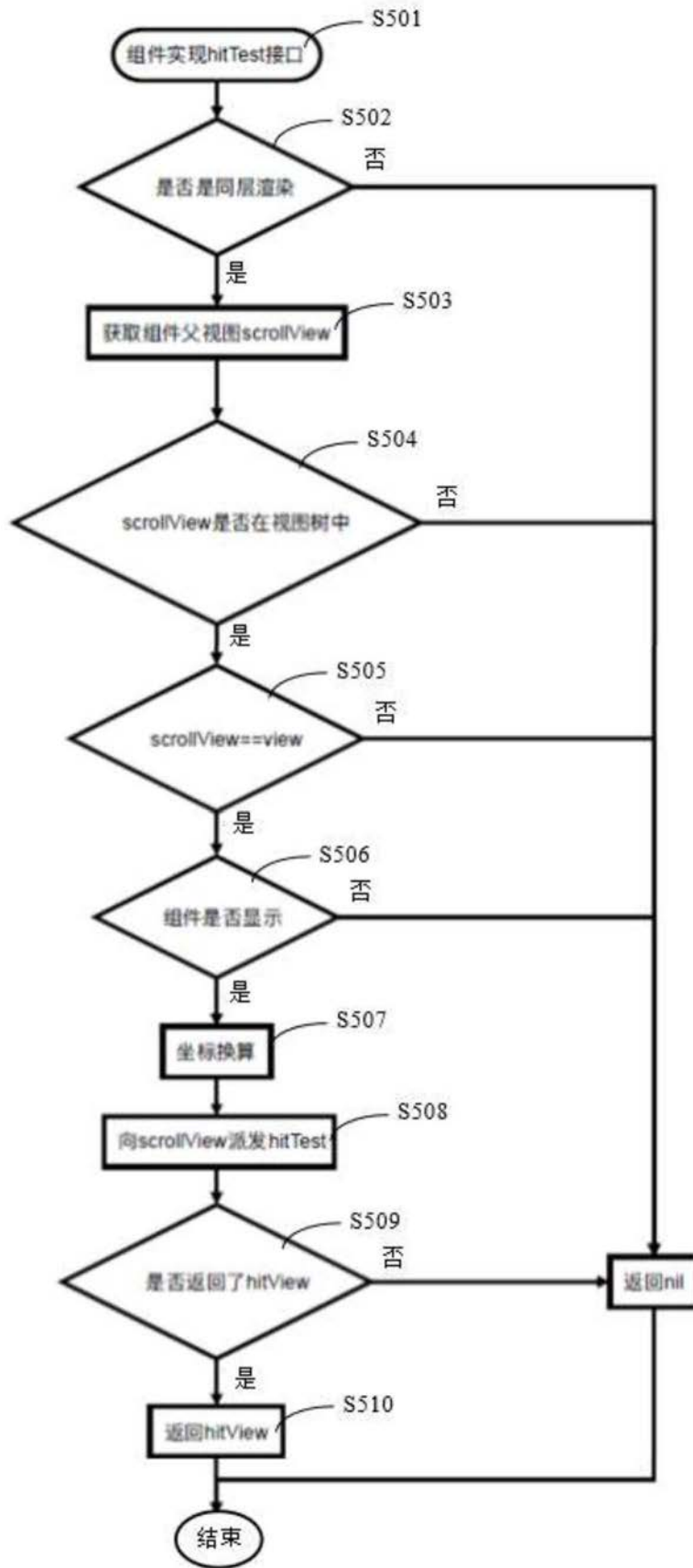


图5

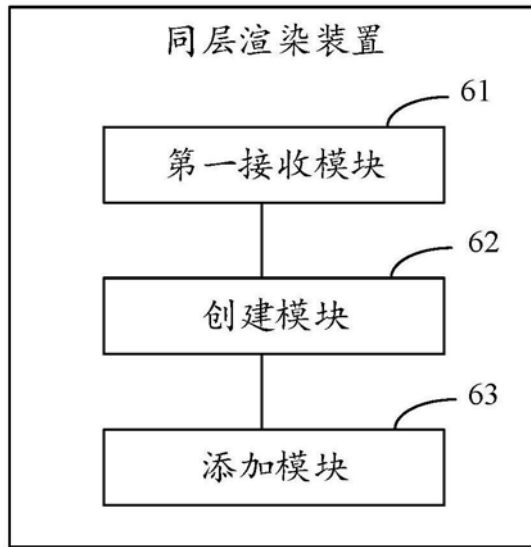


图6

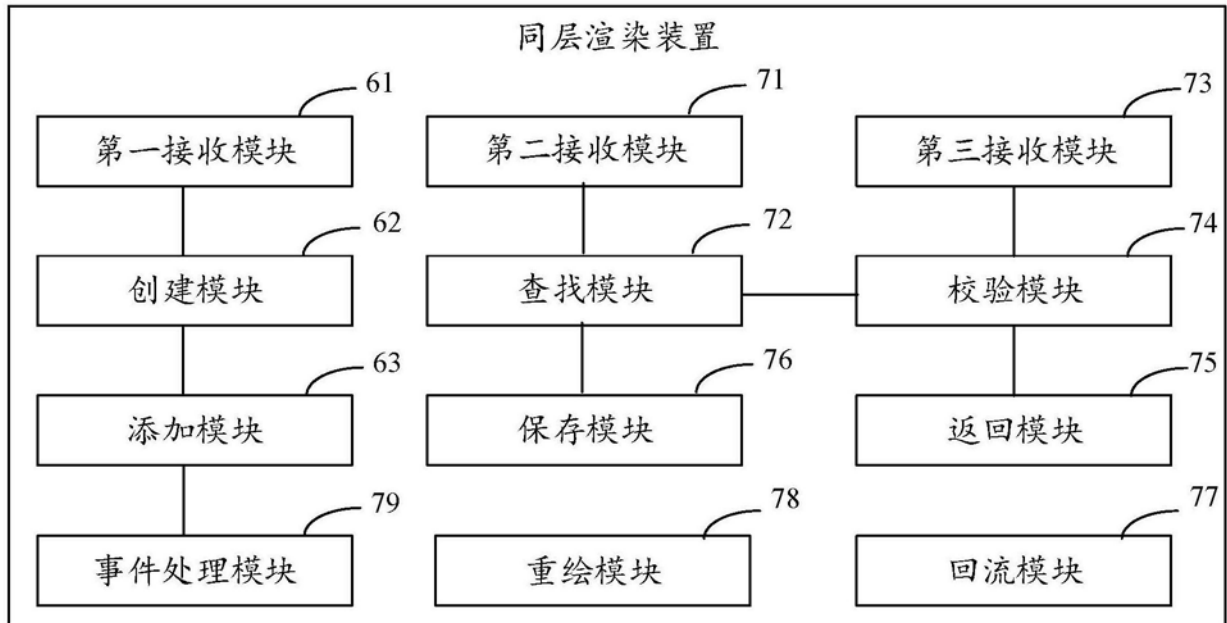


图7

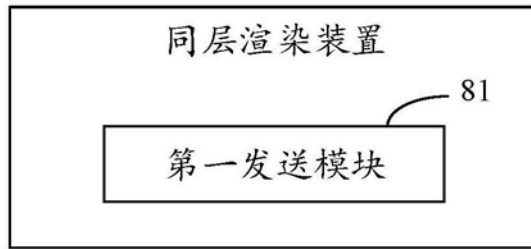


图8

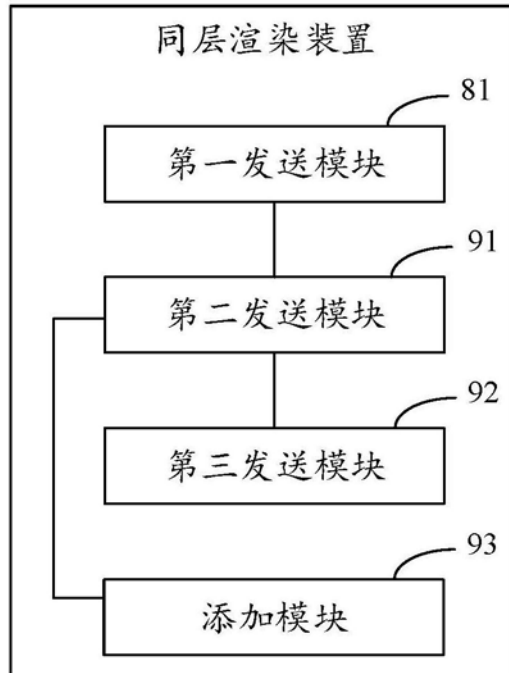


图9

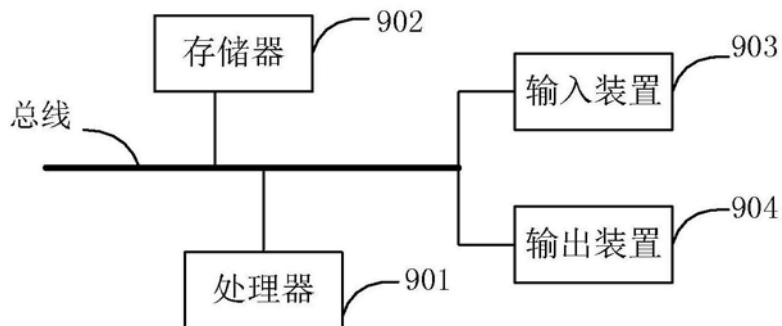


图10