



(12) 发明专利申请

(10) 申请公布号 CN 116594616 A

(43) 申请公布日 2023. 08. 15

(21) 申请号 202310528704.5

(22) 申请日 2023.05.11

(71) 申请人 腾讯科技(深圳)有限公司

地址 518064 广东省深圳市南山区高新区
科技中一路腾讯大厦35层

(72) 发明人 陈倚天

(74) 专利代理机构 深圳市智圈知识产权代理事

务所(普通合伙) 44351

专利代理师 韩绍君

(51) Int. Cl.

G06F 8/38 (2018.01)

G06F 3/04845 (2022.01)

G06F 8/34 (2018.01)

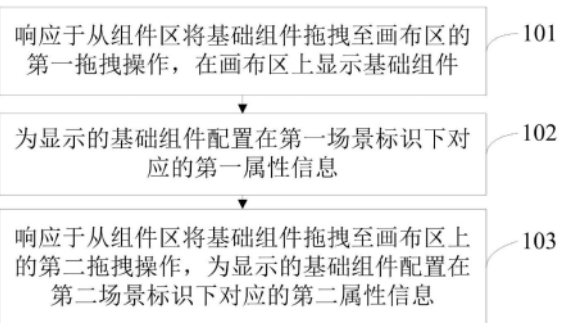
权利要求书2页 说明书18页 附图8页

(54) 发明名称

一种组件配置方法、装置及计算机可读存储介质

(57) 摘要

本申请实施例公开了一种组件配置方法、装置及计算机可读存储介质,本申请可应用于地图或交通场景,通过响应于从组件区将基础组件拖拽至画布区的第一拖拽操作,在画布区上显示基础组件;为显示的基础组件配置在第一场景标识下对应的第一属性信息;响应于从组件区将基础组件拖拽至画布区上的第二拖拽操作,为显示的基础组件配置在第二场景标识下对应的第二属性信息;其中,第二属性信息中至少包括过渡状态子属性和显示状态子属性。以此,通过对一个基础组件多次拖拽至画布区的拖拽操作,实现对基础组件在多个场景标识下的多场景配置,并配置不同场景之间过渡状态子属性相应的过渡动画,提升组件在不同场景状态切换的流畅性,提升用户的使用体验。



1. 一种组件配置方法,其特征在于,包括:
响应于从组件区将基础组件拖拽至画布区的第一拖拽操作,在所述画布区上显示所述基础组件;
为显示的基础组件配置在第一场景标识下对应的第一属性信息;
响应于从组件区将基础组件拖拽至所述画布区上的第二拖拽操作,为所述显示的基础组件配置在第二场景标识下对应的第二属性信息;
其中,所述第二属性信息中至少包括过渡状态子属性和显示状态子属性。
2. 根据权利要求1所述的组件配置方法,其特征在于,所述组件配置方法,还包括:
为组件区中的每个预设组件设置相应的组件标识;
所述基础组件关联目标组件标识,所述在所述画布区上显示所述基础组件,包括:
获取所述画布区上保存的预设组件相应的组件标识;
当检测到所述组件标识中不存在所述目标组件标识时,在所述画布区上新增所述基础组件进行显示。
3. 根据权利要求1或2所述的组件配置方法,其特征在于,所述在所述画布区上显示所述基础组件之后,还包括:
对所述画布区上显示的基础组件进行显示变化处理;
其中,所述显示变化处理至少包括亮度变化处理、透明度变化处理或者闪动变动处理中的一种处理方式。
4. 根据权利要求1至3任一项所述的组件配置方法,其特征在于,所述为显示的基础组件配置在第一场景标识下对应的第一属性信息,包括:
在属性配置区上展示显示的基础组件的第一场景标识以及相应的第一配置项;
接收用户基于第一配置项输入的第一属性信息。
5. 根据权利要求1至4任一项所述的组件配置方法,其特征在于,所述为所述显示的基础组件配置在第二场景标识下对应的第二属性信息,包括:
获取所述画布区上保存的预设组件相应的组件标识;
当检测到所述组件标识中存在所述目标组件标识时,不执行在所述画布区上新增所述基础组件的操作;
将所述显示的基础组件的第一场景标识更改为第二场景标识;
配置在所述第二场景标识下对应的第二属性信息。
6. 根据权利要求5所述的组件配置方法,其特征在于,所述将所述显示的基础组件的第一场景标识更改为第二场景标识,包括:
在属性配置区上将显示的基础组件对应的第一场景标识更改为第二场景标识;
生成所述第二场景标识下对应的第二配置项。
7. 根据权利要求6所述的组件配置方法,其特征在于,所述配置在所述第二场景标识下对应的第二属性信息,包括:
在所述属性配置区上展示显示的基础组件的第二场景标识以及相应的第二配置项;
接收用户基于第二配置项输入的第二属性信息。
8. 根据权利要求1至7任一项所述的组件配置方法,其特征在于,所述为所述显示的基础组件配置在第二场景标识下对应的第二属性信息之后,还包括:

根据所述第一场景标识下的第一属性信息对所述基础组件进行加载,生成并显示第一模型;

响应于对所述第一模型进行切换的触发操作,将所述第一场景标识切换至第二场景标识,并获取所述第二场景标识下的第二属性信息;

播放所述过渡状态子属性相应的过渡动画,并根据所述显示状态子属性对基础组件进行加载,生成并显示第二模型。

9. 根据权利要求8所述的组件配置方法,其特征在于,所述组件配置方法,还包括:

创建变量标识,所述变量标识携带第一变量值;

将所述第一变量值与所述基础组件的第一场景标识关联。

10. 根据权利要求9所述的组件配置方法,其特征在于,所述响应于对所述第一模型进行切换的触发操作,将所述第一场景标识切换至第二场景标识,包括:

响应于对所述第一模型进行切换的触发操作,将所述变量标识的第一变量值更改为第二变量值,并生成变更指令;

响应于所述变更指令,将所述第一场景标识切换至第二场景标识。

11. 一种组件配置装置,其特征在于,包括:

显示单元,用于响应于从组件区将基础组件拖拽至画布区的第一拖拽操作,在所述画布区上显示所述基础组件;

第一配置单元,用于为显示的基础组件配置在第一场景标识下对应的第一属性信息;

第二配置单元,用于响应于从组件区将基础组件拖拽至所述画布区上的第二拖拽操作,为所述显示的基础组件配置在第二场景标识下对应的第二属性信息;

其中,所述第二属性信息中至少包括过渡状态子属性和显示状态子属性。

12. 一种计算机可读存储介质,其特征在于,所述计算机可读存储介质存储有多条指令,所述指令适于处理器进行加载,以执行权利要求1至10任一项所述的组件配置方法中的步骤。

13. 一种计算机设备,其特征在于,包括存储器、处理器以及存储在所述存储器中并可以在所述处理器上运行的计算机程序,所述处理器执行所述计算机程序时实现权利要求1至10任一项所述的组件配置方法中的步骤。

一种组件配置方法、装置及计算机可读存储介质

技术领域

[0001] 本申请涉及计算机技术领域,具体涉及一种组件配置方法、装置及计算机可读存储介质。

背景技术

[0002] 低代码开发平台(LCDP)是无需编码(0代码)或通过少量代码就可以快速生成应用程序的开发平台。通过可视化进行应用程序开发的方法(参考可视编程语言),使具有不同经验水平的开发人员可以通过图形化的用户界面,使用拖拽组件和模型驱动的逻辑来创建网页和移动应用程序。

[0003] 相关技术中,一般都采用以组件的维度进行拖拽来实现低代码开发平台的开发。在对相关技术的研究和实践过程中,本申请的发明人发现,相关技术中,以组件的维度进行拖拽,无法实现组件对应多个状态的流畅切换,因为多个状态需要拖拽多个组件来布局实现,但是拖拽的多个组件之间的切换又无法实现不同状态的流畅过渡,导致显示效果较差,降低用户的使用体验。

发明内容

[0004] 本申请实施例提供一种组件配置方法、装置及计算机可读存储介质,可以提升组件在不同状态切换的流畅性,提升用户的使用体验。

[0005] 为解决上述技术问题,本申请实施例提供以下技术方案:

[0006] 一种组件配置方法,包括:

[0007] 响应于从组件区将基础组件拖拽至画布区的第一拖拽操作,在所述画布区上显示所述基础组件;

[0008] 为显示的基础组件配置在第一场景标识下对应的第一属性信息;

[0009] 响应于从组件区将基础组件拖拽至所述画布区上的第二拖拽操作,为所述显示的基础组件配置在第二场景标识下对应的第二属性信息;

[0010] 其中,所述第二属性信息中至少包括过渡状态子属性和显示状态子属性。

[0011] 一种组件配置装置,包括:

[0012] 显示单元,用于响应于从组件区将基础组件拖拽至画布区的第一拖拽操作,在所述画布区上显示所述基础组件;

[0013] 第一配置单元,用于为显示的基础组件配置在第一场景标识下对应的第一属性信息;

[0014] 第二配置单元,用于响应于从组件区将基础组件拖拽至所述画布区上的第二拖拽操作,为所述显示的基础组件配置在第二场景标识下对应的第二属性信息;

[0015] 其中,所述第二属性信息中至少包括过渡状态子属性和显示状态子属性。

[0016] 在一些实施方式中,所述组件配置装置,还包括:

[0017] 设置单元,用于为组件区中的每个预设组件设置相应的组件标识;

- [0018] 所述基础组件关联目标组件标识,所述显示单元,用于:
- [0019] 获取所述画布区上保存的预设组件相应的组件标识;
- [0020] 当检测到所述组件标识中不存在所述目标组件标识时,在所述画布区上新增所述基础组件进行显示。
- [0021] 在一些实施方式中,所述组件配置装置,还包括显示变化单元,用于:
- [0022] 对所述画布区上显示的基础组件进行显示变化处理;
- [0023] 其中,所述显示变化处理至少包括亮度变化处理、透明度变化处理或者闪动变动处理中的一种处理方式。
- [0024] 在一些实施例中,所述第一配置单元,用于:
- [0025] 在属性配置区上展示显示的基础组件的第一场景标识以及相应的第一配置项;
- [0026] 接收用户基于第一配置项输入的第一属性信息。
- [0027] 在一些实施例中,所述第二配置单元,包括:
- [0028] 获取子单元,用于响应于从组件区将基础组件拖拽至所述画布区上的第二拖拽操作,获取所述画布区上保存的预设组件相应的组件标识;
- [0029] 取消执行子单元,用于当检测到所述组件标识中存在所述目标组件标识时,不执行在所述画布区上新增所述基础组件的操作;
- [0030] 更改子单元,用于将所述显示的基础组件的第一场景标识更改为第二场景标识;
- [0031] 配置子单元,用于配置在所述第二场景标识下对应的第二属性信息。
- [0032] 在一些实施例中,所述更改子单元,用于:
- [0033] 在属性配置区上将显示的基础组件对应的第一场景标识更改为第二场景标识;
- [0034] 生成所述第二场景标识下对应的第二配置项。
- [0035] 在一些实施例中,所述配置子单元,用于:
- [0036] 在所述属性配置区上展示显示的基础组件的第二场景标识以及相应的第二配置项;
- [0037] 接收用户基于第二配置项输入的第二属性信息。
- [0038] 在一些实施例中,所述组件配置装置,还包括:
- [0039] 第一生成单元,用于根据所述第一场景标识下的第一属性信息对所述基础组件进行加载,生成并显示第一模型;
- [0040] 切换单元,用于响应于对所述第一模型进行切换的触发操作,将所述第一场景标识切换至第二场景标识,并获取所述第二场景标识下的第二属性信息;
- [0041] 第二生成单元,用于播放所述过渡状态子属性相应的过渡动画,并根据所述显示状态子属性对基础组件进行加载,生成并显示第二模型。
- [0042] 在一些实施例中,所述组件配置装置,还包括关联单元,用于:
- [0043] 创建变量标识,所述变量标识携带第一变量值;
- [0044] 将所述第一变量值与所述基础组件的第一场景标识关联。
- [0045] 在一些实施例中,所述切换单元,用于:
- [0046] 响应于对所述第一模型进行切换的触发操作,将所述变量标识的第一变量值更改为第二变量值,并生成变更指令;
- [0047] 响应于所述变更指令,将所述第一场景标识切换至第二场景标识,并获取所述第

二场景标识下的第二属性信息。

[0048] 一种计算机可读存储介质,所述计算机可读存储介质存储有多条指令,所述指令适于处理器进行加载,以执行上述组件配置方法中的步骤。

[0049] 一种计算机设备,包括存储器、处理器以及存储在所述存储器中并可以在所述处理器上运行的计算机程序,所述处理器执行所述计算机程序时实现上述组件配置方法中的步骤。

[0050] 一种计算机程序产品或计算机程序,所述计算机程序产品或计算机程序包括计算机指令,所述计算机指令存储在存储介质中。计算机设备的处理器从存储介质读取所述计算机指令,处理器执行所述计算机指令,使得所述计算机上述组件配置方法中的步骤。

[0051] 本申请实施例通过响应于从组件区将基础组件拖拽至画布区的第一拖拽操作,在画布区上显示基础组件;为显示的基础组件配置在第一场景标识下对应的第一属性信息;响应于从组件区将基础组件拖拽至画布区上的第二拖拽操作,为显示的基础组件配置在第二场景标识下对应的第二属性信息;其中,第二属性信息中至少包括过渡状态子属性和显示状态子属性。以此,通过对同一个基础组件多次拖拽至画布区的拖拽操作,实现对同一个基础组件在多个场景标识下的多场景配置,并且还可以实现不同场景之间过渡状态子属性相应的过渡动画的配置,相对于相关技术通过拖拽多个组件实现多个场景状态切换的方案而言,本申请提升了组件在不同场景状态切换的流畅性,提升用户的使用体验。

附图说明

[0052] 为了更清楚地说明本申请实施例中的技术方案,下面将对实施例描述中所需要使用的附图作简单地介绍,显而易见地,下面描述中的附图仅仅是本申请的一些实施例,对于本领域技术人员来讲,在不付出创造性劳动的前提下,还可以根据这些附图获得其他的附图。

[0053] 图1a是本申请实施例提供的组件配置系统的场景示意图;

[0054] 图1b是本申请实施例提供的组件配置系统的架构示意图;

[0055] 图2是本申请实施例提供的组件配置方法的流程示意图;

[0056] 图3a是本申请实施例提供的组件配置方法的场景示意图;

[0057] 图3b是本申请实施例提供的组件配置方法的另一场景示意图;

[0058] 图3c是本申请实施例提供的组件配置方法的另一场景示意图;

[0059] 图4a是本申请实施例提供的组件配置方法的另一流程示意图;

[0060] 图4b是本申请实施例提供的组件配置方法的另一场景示意图;

[0061] 图4c是本申请实施例提供的组件配置方法的另一流程示意图;

[0062] 图5是本申请实施例提供的组件配置装置的结构示意图;

[0063] 图6是本申请实施例提供的服务器的结构示意图。

具体实施方式

[0064] 下面将结合本申请实施例中的附图,对本申请实施例中的技术方案进行清楚、完整地描述,显然,所描述的实施例仅仅是本申请一部分实施例,而不是全部的实施例。基于本申请中的实施例,本领域技术人员在没有作出创造性劳动前提下所获得的所有其他实施

例,都属于本申请保护的范围。

[0065] 为了更好的说明本申请实施例,先对以下对一些相关概念进行介绍:

[0066] 低代码开发平台:无需代码或通过少量代码就可以快速生成应用程序的开发平台。

[0067] 组件:为将重复的代码提取出来合并成的一个个单位,组件可以实现复用,位于框架最底层,其他功能都依赖于组件,可供不同功能使用。组件化的特点是独立性强,方便调用和复用。

[0068] 请参阅图1a,图1a为本申请实施例所提供的组件配置系统的场景示意图,包括:终端11和服务器20,终端11和服务器20之间可以通过通信网络连接,该通信网络,包括无线网络以及有线网络,其中无线网络包括无线广域网、无线局域网、无线城域网、以及无线个人网中的一种或多种的组合。网络中包括路由器、网关等等网络实体,图中并未示意出。终端11可以通过通信网络与服务器20进行信息交互,例如,服务器20将组件数据发送至终端11。

[0069] 该组件配置系统可以包括组件配置装置,该组件配置装置具体可以集成在服务器20中,该服务器20可以是独立的物理服务器,也可以是多个物理服务器构成的服务器集群或者分布式系统,还可以是提供云服务、云数据库、云计算、云函数、云存储、网络服务、云通信、中间件服务、域名服务、安全服务、CDN、以及大数据和人工智能平台等基础云计算服务的云服务器,用于存储组件数据,该组件数据可以由代码组成,服务器20可以将该组件数据发送至终端11中,以使得终端11可以根据该组件数据生成多个组件进行展示。

[0070] 该组件配置系统还可以包括终端,该终端可以是智能手机、平板电脑、笔记本电脑、台式计算机、智能音箱、智能手表、车载终端、飞行器等。在图1a中,该终端即为图1a中的终端11,该终端11中可以安装有各种用户所需的客户端,比如低代码开发客户端。该终端11可以根据服务器20发送组件数据生成多个组件,进而响应于从组件区将基础组件拖拽至画布区的第一拖拽操作,在该画布区上显示该基础组件;为显示的基础组件配置在第一场景标识下对应的第一属性信息;响应于从组件区将基础组件拖拽至该画布区上的第二拖拽操作,为该显示的基础组件配置在第二场景标识下对应的第二属性信息;其中,该第二属性信息中至少包括过渡状态子属性和显示状态子属性。

[0071] 为了更好的说明低代码开发平台,请参阅图1b,图1b为本申请实施例所提供的组件配置系统的架构示意图,该架构示意图也可以理解为低代码开发平台的架构,该架构包括基础服务层、组件层、协议层、配置层和应用层。

[0072] 该基础服务层,用于提供应用环境的基础服务功能,例如账号管理服务可以提供用户账号登录验证的基础功能。

[0073] 该组件层,用于提供组件的基础功能,例如可以提供模型组件的组件化、能力抽象和渲染引擎的基础功能。

[0074] 该协议层,用于提供应用环境实现协议的实现功能,至少包括应用搭建协议、页面搭建协议和组件描述协议。

[0075] 该配置层,用于提供应用环境的操作配置实现功能。

[0076] 该应用层,用于提供具体应用环境的实现功能,即提供应用环境,例如PC(Personal Computer,个人计算机)应用。

[0077] 需要说明的是,图1a所示的组件配置系统的场景示意图以及图1b所示的组件配置

系统的架构示意图仅仅是一个示例,本申请实施例描述的组件配置系统、框架以及场景是为了更加清楚的说明本申请实施例的技术方案,并不构成对于本申请实施例提供的技术方案的限制,本领域普通技术人员可知,随着组件配置系统和框架的演变和新业务场景的出现,本申请实施例提供的技术方案对于类似的技术问题,同样适用。

[0078] 以下分别进行详细说明。

[0079] 本申请实施例提供了一种组件配置方法,该方法可以由终端或服务器执行,也可以由终端和服务器共同执行;本申请实施例以组件配置方法由终端执行为例来进行说明。

[0080] 请参阅图2,图2是本申请实施例提供的组件配置方法的流程示意图。该组件配置方法包括:

[0081] 在步骤101中,响应于从组件区将基础组件拖拽至画布区的第一拖拽操作,在画布区上显示基础组件。

[0082] 可以理解的是,在本申请的具体实施方式中,涉及到用户数据和属性信息等相关的数据,当本申请以上实施例运用到具体产品或技术中时,需要获得用户许可或者同意,且相关数据的收集、使用和处理需要遵守相关国家和地区的相关法律法规和标准。

[0083] 需要说明的是,在说明书、权利要求书和附图所描述的一些流程中,包含了按照特定顺序出现的多个步骤,但应该清楚了解,这些步骤可以不按照其在本文中出现的顺序来执行或并行执行,步骤序号仅仅是用于区分各个不同的步骤,序号本身不代表任何的执行顺序。此外,本文中的“第一”和“第二”等描述,是用于区别类似的对象,而不必用于描述特定的顺序或先后次序。

[0084] 低代码开发平台是无需编码或通过少量代码就可以快速生成应用程序的开发平台。通过可视化进行应用程序开发的方法,使具有不同经验水平的开发人员可以通过图形化的用户界面,使用拖拽组件的逻辑来创建网页和移动应用程序。

[0085] 一般可以设定组件区、画布区和属性配置区来实现低代码开发平台,该组件区支持展示每个组件,该组件的类型可以包括模型组件、地图类组件以及图标类组件等,上述举例不作为具体限定,为了更好的说明本申请实施例,后续以组件的类型为模型组件进行说明。

[0086] 该画布区用于实现承载组件的设置,支持组件的复制、粘贴和保存等常用快捷操作。该属性配置区提供对保存在画布区上的组件的样式、交互和数据的可视化定义。

[0087] 目前,一般都采用对组件进行拖拽来实现低代码开发平台的开发。但是,如果需要实现组件对应多个状态的切换时,需要从组件区拖拽多个组件到画布区,分别进行属性配置实现,由于拖拽的多个组件对应不同的页面,那么后续多个组件之间进行状态切换时,会导致无法实现流畅过渡,出现类似页面切换的卡顿现象,严重降低用户的体验感。

[0088] 为了解决上述技术问题,本申请实施例提出一种预设组件(过渡组件)来实现同一组件的多场景状态配置的方法,实现单组件进行多场景状态的分别配置以及过渡动画配置,提升后续组件的多个状态切换的流畅性,具体请继续参阅以下步骤。

[0089] 为了更好的说明本申请实施例,请一并参阅图3a所示,本申请实施例的低代码开发平台可以包括组件区、画布区以及属性配置区,与相关技术不同的是,本申请实施例可以通过对协议层中的组件搭建协议进行补充,增加组件相关的组件协议,即新增预设组件的相关字段描述(也可以理解为组件标识),使得预设组件关联相应的组件标识。

[0090] 在一实施方式中,本申请实施例还包括:为组件区中的每个预设组件设置相应的组件标识。

[0091] 其中,该预设组件也可以理解为过渡组件,即专门用于实现同一组件下多个场景状态流畅切换的组件,以此,可以为组件区中的每个预设组件设置相应的组件标识,用于区分其他普通组件。

[0092] 另外,还与相关技术不同的是,本申请实施例还可以为具有组件标识的预设组件增加当前场景标识,该当前场景标识可以实现同一组件在场景状态区分的效果,每个场景标识可以对应设置不同的属性信息,该属性信息用于对组件实现可视化属性设置和渲染样式设置,例如该属性信息可以包括相机、光照、模型状态、动画或者图像来源等等属性,该预设组件可以为地图类组件、图表类组件、表单类组件、容器类组件或者模型组件等。

[0093] 如图3a所示,在一实施方式中,可以通过在属性配置区为具有组件标识的预设组件添加当前场景标识控件A,该当前场景标识控件A用于实现为预设组件设置相应的当前场景标识,需要说明的是,非预设组件不具有当前场景标识控件A,无法设置相应的当前场景标识,即本申请实施例应用于预设组件的场景标识使用,非预设组件的使用方式没有变化。

[0094] 基于此,该基础组件即为预设组件中的一个组件,即该基础组件关联相应的目标组件标识,在一些实施方式中,该基础组件可以为具有目标组件标识的模型组件、具有目标组件标识的地图类组件、或者具有目标组件标识的容器类组件等,该基础组件在低代码开发平台的组件区进行展示,例如,如图3a所示,在低代码开发平台的组件区的模型组件区域上显示基础组件,此时画布区不存在该基础组件,以此,用户可以通过长按拖动的方式实现对该基础组件的拖拽操作,相应的,用户可以实现从组件区将基础组件拖拽至画布区的第一拖拽操作,请一并参阅图3b所示,终端响应于从组件区将基础组件拖拽至画布区的第一拖拽操作,在该画布区上显示该基础组件,需要说明的是,由于该画布区上不存在该基础组件,即不存在该基础组件相应的目标组件标识,所以该基础组件为第一次从组件区拖拽至画布区,可以实现对该基础组件的首个场景状态的配置,即可以在该画布区上新增该基础组件进行显示。

[0095] 在一些实施方式中,该基础组件关联目标组件标识,该在该画布区上显示该基础组件,包括:

[0096] (1) 获取该画布区上保存的预设组件相应的组件标识;

[0097] (2) 当检测到该组件标识中不存在该目标组件标识时,在该画布区上新增该基础组件进行显示。

[0098] 其中,该基础组件为预设组件中的一种,可以配置多个场景标识以实现多个场景状态,以此,低代码开发平台需要检测画布区上是否已经保存了相同的基础组件,由于每个预设组件均关联有相应的组件标识,因此可以通过组件标识来判定画布区上是否已经保存了相同的基础组件,这也是本申请实施例实现多个场景状态的关键点,请继续参阅图3b所示,可以预先获取画布区上保存的预设组件相应的组件标识,由于该画布区上暂时不存在预设组件,所以此时不存在组件标识。

[0099] 进一步的,该基础组件关联相应的目标组件标识,因此低代码开发平台可以检测该组件标识中是否存在该目标组件标识,来实现判定该画布区上是否已经保存了相应的基础组件,而此时该画布区上暂时不存在预设组件,所以不存在组件标识,即检测到组件标识

中不存在该目标组件标识,该基础组件为第一次添加至画布区,可以在该画布区上新增该基础组件进行相应的显示,并可以为该基础组件配置第一场景标识,即可以在当前场景标识控件A中为基础组件配置第一场景标识1。

[0100] 在一些实施方式中,该在所述画布区上显示所述基础组件之后,还包括:对所述画布区上显示的基础组件进行显示变化处理。

[0101] 其中,由于该基础组件为预设组件中的一种,即可以用于后续为基础组件配置多个场景标识以实现多个场景状态,因此,为了提示用户进行多个场景标识的配置,可以将显示在画布区上的基础组件进行显示变化处理,该显示变化处理指将画布区上显示的基础组件的显示状态进行变化,以区别于其他的非预设组件。

[0102] 所述显示变化处理至少包括亮度变化处理、透明度变化处理或者闪动变动处理中的一种处理方式。

[0103] 该亮度变化处理指可以将画布区上显示的基础组件的亮度进行改变,例如增加画布区上的基础组件的亮度,或者降低画布区上的基础组件的亮度,以使得画布区上的基础组件进行标志化显示,明显区别于其他非预设组件,提示用户可以根据亮度变化处理后的基础组件配置多个场景标识以实现多个场景状态。

[0104] 该透明度变化处理指可以将画布区上显示的基础组件的透明度进行改变,例如将画布区上的基础组件的透明度设置为百分之75,或者设置为百分之50,以使得画布区上的基础组件进行标志化显示,明显区别于其他非预设组件,提示用户可以根据透明度变化处理后的基础组件配置多个场景标识以实现多个场景状态。

[0105] 该闪动变化处理指可以将画布区上显示的基础组件的显示属性进行改变,例如将画布区上的基础组件的显示属性更改为按照每秒闪动一次的频率进行设置,或者按照每秒2秒闪动一次的频率进行设置,以使得画布区上的基础组件进行标志化显示,明显区别于其他非预设组件,提示用户可以根据闪动变化处理后的基础组件配置多个场景标识以实现多个场景状态。

[0106] 在步骤102中,为显示的基础组件配置在第一场景标识下对应的第一属性信息。

[0107] 其中,由于该场景标识可以实现该基础组件在不同的场景状态区分,因此需要为该基础组件配置每个场景标识下的属性信息,即需要为显示在画布区中的基础组件配置在第一场景标识下对应的第一属性信息,该第一属性信息可以由用户基于属性的配置项输入生成的,该第一属性信息用于配置该基础组件在第一场景标识下的可视化形态和渲染样式。

[0108] 需要说明的是,由于本申请实施例可以实现同一预设组件的不同场景状态的属性配置,因此,同一预设组件的不同场景状态均可以在同一页面中显示,所以本申请实施例还可以为该预设组件设置过渡动画的属性配置,即过渡状态子属性的设置,该过渡状态子属性用于描述预设组件从上一个场景标识的场景状态过渡到当前场景标识的场景状态。

[0109] 请继续参阅图3b所示,可以在属性配置区中为新增在画布区上的基础组件进行首个场景状态的配置,即在属性配置区的当前场景标识控件A中为该基础组件配置第一场景标识1,并且显示该第一场景标识1相应的多个第一配置项,用户可以通过该多个第一配置项为基础组件在第一场景标识1下配置相应的第一属性信息,例如进行模型配置、流光效果、相机特效、动画特效等属性配置,得到第一属性信息。

[0110] 在一些实施方式中,该为显示的基础组件配置在第一场景标识下对应的第一属性信息,可以包括:

[0111] (1)在属性配置区上展示显示的基础组件的第一场景标识以及相应的第一配置项;

[0112] (2)接收用户基于第一配置项输入的第一属性信息。

[0113] 其中,请继续参阅图3b所示,可以在属性配置区中展示新增在画布区上显示的基础组件的第一场景标识1,具体的,可以在属性配置区的当前场景标识控件A上显示该第一场景标识1,并同时显示该第一场景标识1下相应的第一配置项,该第一配置项主要配置该基础组件对应在第一场景标识1的场景状态,场景状态可以分为两大类,分别为静止状态(由显示状态子属性表征)和过渡状态(即过渡状态子属性表征),该静止状态用于描述基础组件在相应场景下的显示状态,该过渡状态用于描述基础组件从上一个场景标识的场景过渡到当前场景的过渡效果。

[0114] 该第一配置项可以包括模型配置子项、流光效果配置子项、相机配置子选项和动画配置子项等配置子选项,该模型配置子项可以用于配置该基础组件相应的模型形态,该流光效果配置子项可以配置该基础组件的流光特效,该相机配置子项可以配置该基础组件的展示角度,该动画配置子项可以配置从上一个场景标识的场景状态过渡到当前场景标识的动画效果,该模型配置子项、流光效果配置子项、相机配置子选项可以配置显示状态子属性,该动画配置子项可以配置过渡状态子属性。

[0115] 进一步的,用户可以通过该第一配置项配置基础组件在第一场景标识下的第一属性信息,设定基础组件在第一场景标识下的进行展示的模型可视化状态和渲染样式,例如通过第一属性信息设定基础组件为“鼎”的可视化状态,上方光照的渲染样式。

[0116] 在步骤103中,响应于从组件区将基础组件拖拽至画布区上的第二拖拽操作,为显示的基础组件配置在第二场景标识下对应的第二属性信息。

[0117] 其中,用户可以继续从组件区将基础组件继续拖拽至画布区上,实现基础组件在不同场景状态下的属性配置,与相关技术中,将同一组件拖拽至画布区上会生成多个相同的组件各自进行配置的实现方式不同的是,本申请实施例在将基础组件再次拖拽至画布区上之后,该画布区上不新增该基础组件,而是扩展该基础组件的场景标识相应的场景状态。

[0118] 即用户可以实现从组件区再次将基础组件拖拽至画布区的第二拖拽操作,来实现基础组件在另一场景状态下的属性配置,为了更好的说明本申请实施例,请一并参阅图3c所示,终端响应于从组件区将基础组件再次拖拽至画布区的第二拖拽操作,不在画布区上显示新的基础组件,而是对之前显示在画布区的基础组件配置第二场景标识2下的第二属性信息,该第二属性信息可以由用户基于新重置的属性的配置项输入生成的,该第二属性信息与第一属性信息不同,第二属性信息用于配置该基础组件在该第二场景标识下的可视化形态和渲染样式。

[0119] 相关技术中,不同的组件之间进行切换时,因为不同的组件对应不同的页面,所以切换时会出现页面变换,造成卡顿现象,导致不同场景状态之间的切换会有明显的停顿感,流畅性较差,给用户带来非常差的体验。在本申请实施例中,由于该第一场景标识1和第二场景标识2同属于一个基础组件,因此,两个场景标识相应的场景状态可以对应在同一页面,以此,由第一场景标识的场景状态切换至第二场景标识的场景状态时,不会造成页面变

换,带来卡顿现象,使得不同场景状态之间切换的流畅性得到极大的提升。

[0120] 相应的,由于基础组件不同的场景标识对应的场景状态均可以在同一页面中显示,为了更好的实现在不同场景状态之间的流畅衔接以及个性化表现,可以为第二场景标识2设置相应的过渡状态子属性,该过渡状态子属性用于描述基础组件从第一场景标识1的场景状态过渡到第二场景标识2的场景状态过程的过渡效果,例如该过渡状态子属性至少可以包括动画参数,可以理解的是,该第二场景标识2还可以包括显示状态子属性,该显示状态子属性用于描述基础组件在第二场景标识2的场景状态下的显示状态集合,例如该显示状态子属性至少可以包括模型配置参数、流光效果参数以及相机展示角度参数。

[0121] 在一些实施方式中,该为该显示的基础组件配置在第二场景标识下对应的第二属性信息,包括:

[0122] (1) 获取该画布区上保存的预设组件相应的组件标识;

[0123] (2) 当检测到该组件标识中存在该目标组件标识时,不执行在该画布区上新增该基础组件的操作;

[0124] (3) 将该显示的基础组件的第一场景标识更改为第二场景标识;

[0125] (4) 配置在该第二场景标识下对应的第二属性信息。

[0126] 其中,请继续参阅图3c所示,低代码开发平台响应于从组件区将基础组件拖拽至画布区上的第二拖拽操作,可以预先获取画布区上保存的预设组件相应的组件标识,由于该画布区上存在基础组件,因此,可以获取到该基础组件的目标组件标识,即组件标识中存在该目标组件标识。

[0127] 进一步的,低代码开发平台可以检测组件标识中是否存在该目标组件标识,来实现判定该画布区上是否已经保存了相应的基础组件,而此时该画布区上已经存在基础组件,以此,存在相应的目标组件标识,即检测到该组件标识中存在该目标组件标识,与相关技术在画布区再新增一个基础组件进行属性配置的方案不同的是,本申请实施例不在该画布区上新增该基础组件,而是为显示在画布区上的基础组件配置新的场景标识,即可以将画布区上显示的基础组件的第一场景标识1更改为第二场景标识2,进而对第二场景标识2的场景状态进行第二属性信息配置。

[0128] 在一些实施方式中,该将该显示的基础组件的第一场景标识更改为第二场景标识,可以包括:

[0129] (1.1) 在属性配置区上将显示的基础组件对应的第一场景标识更改为第二场景标识;

[0130] (1.2) 生成该第二场景标识下对应的第二配置项。

[0131] 其中,请继续参阅图3b所示,可以将属性配置区的当前场景标识控件A上显示的第一场景标识1自动加1,更改为第二场景标识2,并更新显示该第二场景标识2下相应的第二配置项,该第二配置项主要配置该基础组件对应第二场景标识2的场景状态。

[0132] 该第二配置项可以包括模型配置子项、流光效果配置子项、相机配置子选项和动画配置子项等配置子选项,该模型配置子项可以用于配置该基础组件在该第二场景标识2下相应的模型形态,该流光效果配置子项可以配置该基础组件在该第二场景标识2的流光特效,该相机配置子项可以配置该基础组件在该第二场景标识2的展示角度,该动画配置子项可以配置从第一场景标识1的场景状态过渡到当前的第二场景标识2的动画效果,该模型配

置子项、流光效果配置子项、相机配置子选项可以配置该基础组件在该第二场景标识2的显示状态子属性,该动画配置子项可以配置该基础组件在该第二场景标识2的过渡状态子属性。

[0133] 在一些实施方式中,该配置在该第二场景标识下对应的第二属性信息,可以包括:

[0134] (2.1) 在该属性配置区上展示显示的基础组件的第二场景标识以及相应的第二配置项;

[0135] (2.2) 接收用户基于第二配置项输入的第二属性信息。

[0136] 其中,用户可以通过该第一配置项配置基础组件在第二场景标识下的第二属性信息,设定基础组件在二场景标识下的进行展示的模式可视化状态和渲染样式,例如通过第二属性信息设定基础组件为“鼎”的细节可视化状态,过渡动画可以为逐渐拉近的动画效果以及正面光照的渲染样式。

[0137] 在一些实施方式中,用户可以通过继续将基础组件拖拽至画布区上,实现基础组件继续在新的场景标识的场景状态下的属性配置,实现基础组件的多个场景状态的配置。

[0138] 由上述可知,本申请实施例通过响应于从组件区将基础组件拖拽至画布区的第一拖拽操作,在画布区上显示基础组件;为显示的基础组件配置在第一场景标识下对应的第一属性信息;响应于从组件区将基础组件拖拽至画布区上的第二拖拽操作,为显示的基础组件配置在第二场景标识下对应的第二属性信息;其中,第二属性信息中至少包括过渡状态子属性和显示状态子属性。以此,通过对同一个基础组件多次拖拽至画布区的拖拽操作,实现对同一个基础组件在多个场景标识下的多场景配置,并且还可以实现不同场景之间过渡状态子属性相应的过渡动画的配置,相对于相关技术通过拖拽多个组件实现多个场景状态切换的方案而言,本申请提升了组件在不同场景状态切换的流畅性,提升用户的使用体验。

[0139] 结合上述实施例所描述的方法,以下将举例作进一步详细说明。

[0140] 在本实施例中,将以该组件配置装置具体集成在服务器中为例进行说明,具体参照以下说明。

[0141] 请参阅图4a,图4a为本申请实施例提供的组件配置方法的另一流程示意图。该方法流程可以包括:

[0142] 在步骤201中,终端为组件区中的每个预设组件设置相应的组件标识,响应于从组件区将基础组件拖拽至画布区的第一拖拽操作,获取画布区上保存的预设组件相应的组件标识,当检测到组件标识中不存在目标组件标识时,在画布区上新增基础组件进行显示。

[0143] 其中,请继续参阅图1b所示,与相关技术不同的是,本申请实施例可以通过对协议层中的组件搭建协议进行补充,增加组件相关的组件协议,即新增预设组件的相关字段描述(也可以理解为组件标识),使得预设组件关联相应的组件标识,该预设组件也可以理解为过渡组件,即专门用于实现同一组件下多个场景状态流畅切换的组件,以此,可以为组件区中的每个预设组件设置相应的组件标识,用于区分其他普通组件。

[0144] 请继续参阅图3a所示,终端为低代码开发平台中组件区中的每个预设组件设置相应的组件标识,以模型组件为例,该模型组件在本申请实施例即为基础组件,该模型组件为预设组件,可以为该模型组件设置相应的的组件标识AA。

[0145] 进一步的,在低代码开发平台的组件区的模型组件区域上显示该基础组件,此时

画布区不存在该基础组件,以此,用户可以通过长按拖动的方式实现对该基础组件的拖拽操作,相应的,用户可以实现从组件区将基础组件拖拽至画布区的第一拖拽操作,终端响应于从组件区将基础组件拖拽至画布区的第一拖拽操作,可以预先获取画布区上保存的预设组件相应的组件标识,由于该画布区上暂时不存在预设组件,所以此时画布区上不存在组件标识。

[0146] 由于该基础组件关联相应的目标组件标识,因此低代码开发平台可以检测该组件标识中是否存在该目标组件标识,来实现判定该画布区上是否已经保存了相应的基础组件,此时该画布区上暂时不存在预设组件,所以不存在组件标识,那么会检测到组件标识中不存在该目标组件标识,说明基础组件为第一次添加至画布区,可以在该画布区上新增该基础组件进行相应的显示

[0147] 在步骤202中,终端在属性配置区上展示显示的基础组件的第一场景标识以及相应的第一配置项,接收用户基于第一配置项输入的第一属性信息。

[0148] 其中,请继续参阅图3b所示,可以在属性配置区的当前场景标识控件A上显示该第一场景标识1,并同时显示该第一场景标识1下相应的第一配置项,该第一配置项主要配置该基础组件对应在第一场景标识1的场景状态,场景状态可以分为两大类,分别为静止状态(由显示状态子属性表征)和过渡状态(即过渡状态子属性表征),该静止状态用于描述基础组件在相应场景下的显示状态,该过渡状态用于描述基础组件从上一个场景标识的场景过渡到当前场景的过渡效果。

[0149] 如图3b所示,该第一配置项可以包括模型配置子项、流光效果配置子项、相机配置子项和动画配置子项等配置子项,该模型配置子项可以用于配置该基础组件相应的模型形态,该流光效果配置子项可以配置该基础组件的流光特效,该相机配置子项可以配置该基础组件的展示角度,该动画配置子项可以配置从上一个场景标识的场景状态过渡到当前场景标识的动画效果,该模型配置子项、流光效果配置子项、相机配置子项可以配置显示状态子属性,该动画配置子项可以配置过渡状态子属性。

[0150] 进一步的,用户可以通过该第一配置项配置基础组件在第一场景标识下的第一属性信息,例如,设定基础组件在第一场景标识下的进行展示的模型可视化状态和渲染样式,例如通过第一属性信息设定基础组件为“鼎”的模型形态,无流光效果,正面的展示角度,以及无动画效果。

[0151] 在步骤203中,终端响应于从组件区将基础组件拖拽至画布区上的第二拖拽操作,获取画布区上保存的预设组件相应的组件标识,当检测到组件标识中存在目标组件标识时,不执行在画布区上新增基础组件的操作。

[0152] 其中,为了扩展该基础组件的场景标识相应的场景状态,实现基础组件在另一场景状态下的属性配置,用户可以实现从组件区再次将基础组件拖拽至画布区的第二拖拽操作,请继续参阅图3c所示,低代码开发平台可以响应于从组件区将基础组件拖拽至画布区上的第二拖拽操作,可以预先获取画布区上保存的预设组件相应的组件标识,由于该画布区上存在基础组件,因此,可以获取到该基础组件的目标组件标识,即组件标识中存在该目标组件标识。

[0153] 进一步的,低代码开发平台可以检测组件标识中是否存在该目标组件标识,来实现判定该画布区上是否已经保存了相应的基础组件,而此时该画布区上已经存在基础组

件,以此,存在相应的目标组件标识,即检测到该组件标识中存在该目标组件标识,与相关技术在画布区再新增一个基础组件进行属性配置的方案不同的是,本申请实施例不在该画布区上新增该基础组件,而是为显示在画布区上的基础组件配置新的场景标识,即则不对画布中的页面描述新增组件,而是针对原有的组件进行场景协议的新增,从而达到拖入多个组件只维护一个组件实例,并增加“场景”值。

[0154] 在步骤204中,终端在属性配置区上将显示的基础组件对应的第一场景标识更改为第二场景标识,生成第二场景标识下对应的第二配置项。

[0155] 其中,请继续参阅图3b所示,可以将属性配置区的当前场景标识控件A上显示的第一场景标识1自动加1,更改为第二场景标识2,并更新显示该第二场景标识2下相应的第二配置项,该第二配置项主要配置该基础组件对应第二场景标识2的场景状态。

[0156] 该第二配置项可以包括模型配置子项、流光效果配置子项、相机配置子选项和动画配置子项等配置子选项,该模型配置子项可以用于配置该基础组件在该第二场景标识2下相应的模型形态,该流光效果配置子项可以配置该基础组件在该第二场景标识2的流光特效,该相机配置子项可以配置该基础组件在该第二场景标识2的展示角度,该动画配置子项可以配置从第一场景标识1的场景状态过渡到当前的第二场景标识2的动画效果,该模型配置子项、流光效果配置子项、相机配置子选项可以配置该基础组件在该第二场景标识2的显示状态子属性,该动画配置子项可以配置该基础组件在该第二场景标识2的过渡状态子属性。

[0157] 在步骤205中,终端在属性配置区上展示显示的基础组件的第二场景标识以及相应的第二配置项,接收用户基于第二配置项输入的第二属性信息。

[0158] 其中,请继续参阅图3b所示,终端可以在属性配置区上展示显示的基础组件的第二场景标识2以及相应的第二配置项。

[0159] 进一步的,用户可以通过该第二配置项配置基础组件在第二场景标识2下的第二属性信息,例如,设定基础组件在第二场景标识2下的进行展示的模型可视化状态和渲染样式,例如通过第二属性信息设定基础组件为“鼎的详细铭文”的模型形态,无流光效果,正面的展示角度,逐渐拉近动画效果。

[0160] 在一些实施方式中,用户可以通过继续将基础组件拖拽至画布区上,实现基础组件继续在新的场景标识的场景状态下的属性配置,实现基础组件的多个场景状态的配置。例如用户可以继续将基础组件拖拽至画布区上,实现基础组件继续在新的场景标识3的配置新的属性信息,如设定基础组件新的第三场景标识3下进行展示的模型可视化状态和渲染样式,可以通过第三场景标识3的新的第三属性信息设定基础组件为“鼎的详细信息展示”,的模型形态,无流光效果,正面的展示角度,逐渐拉远动画效果。

[0161] 在步骤206中,终端根据第一场景标识下的第一属性信息对基础组件进行加载,生成并显示第一模型,创建变量标识,将第一变量值与基础组件的第一场景标识关联。

[0162] 其中,在完成对该基础组件配置了不同场景标识下的属性信息之后,说明低代码开发平台实现同一组件在不同场景状态下的状态设置,完成低代码开发过程,进而,终端可以根据第一场景标识下的第一属性信息对基础组件进行加载,如图4b所示,图4b为低代码开发平台根据基础组件进行低代码开发生成的应用程序31,该应用程序31包括多个由基础组件对应不同场景状态下的应用页面,第一个应用页面即为根据第一场景标识下的第一属

性信息对基础组件进行加载生成的,该第一个应用页面包括基础组件生成的“鼎”的模型形态,无流光效果,正面的展示角度,以及无动画效果。

[0163] 进一步的,可以创建变量标识,该变量标识默认为第一变量值,该第一变量值可以为1,该第一变量值可以和第一应用页面中的基础组件的第一场景标识1进行关联,即后续该第一变量值的变化会触发场景标识相应的变化,该第一变量值可以通过预设事件触发进行改变,该预设事件可以为控件切换,例如第一应用页面中的第一控件311可以表示第一变量值1,当该第一控件311触发时,可以触发基础组件对应第一场景标识的场景状态的触发,显示页面,展示由基础组件生成的“鼎”的模型形态,无流光效果,正面的展示角度,以及无动画效果。后续可以通过对该第一控件311进行切换,来触发场景状态的变化。

[0164] 在步骤207中,终端响应于对第一模型进行切换的触发操作,将变量标识的第一变量值更改为第二变量值,并生成变更指令,响应于变更指令,将第一场景标识切换至第二场景标识,并获取第二场景标识下的第二属性信息。

[0165] 其中,为了展示该基础组件在其他场景标识的场景状态的展示,可以对第一模型进行切换,切换的方式可以为对控件切换,请继续参阅图4b所示,可以将第一控件311的切换到第二控件312,来实现对第一模型进行切换的触发操作,相应的,终端响应于该第一模型进行切换的触发操作,会将变量标识的第一变量值1更改为第二变量值2,由于该变量标识的变量值改变,会生成相应的变更指令,该应用程序31响应于变更指令,将场景标识由第一场景标识1切换为与第二变量值对应关联的第二场景标识2,并获取该第二场景标识2下的第二属性信息,实现场景状态的快速更新。

[0166] 在步骤208中,终端播放过渡状态子属性相应的过渡动画,并根据显示状态子属性对基础组件进行加载,生成并显示第二模型。

[0167] 其中,由于基础组件不同的场景标识对应的场景状态均在同一页面中显示,所以可以通过过渡动画实现第一场景标识的场景状态切换至第二场景标识的场景状态,即获取该第二属性信息中的过渡状态子属性,即“逐渐拉近动画效果”,来播放相应的逐渐拉近的过渡动画,并在该过渡动画播放完毕时,根据显示状态子属性,即“鼎的详细铭文”的模型形态,无流光效果,正面的展示角度来对基础组件重新记载,生成并显示该第二模型,即为应用程序31的第二个应用页面中展示的“鼎的详细铭文”。

[0168] 在一些实施方式中,还可以将第一控件312的切换到第二控件313,来实现对第二模型进行切换的触发操作,相应的,终端响应于该第二模型进行切换的触发操作,会将变量标识的第二变量值2更改为第三变量值3,由于该变量标识的变量值改变,会生成相应的变更指令,该应用程序31响应于变更指令,将场景标识由第二场景标识2切换为与第二变量值对应关联的第二场景标识3,并获取该第二场景标识2下的第三属性信息,实现场景状态的快速更新,即获取该第二属性信息中的过渡状态子属性,即“逐渐拉远动画效果”,来播放相应的逐渐拉远的过渡动画,并在该过渡动画播放完毕时,根据显示状态子属性,即“鼎的详细信息展示”的模型形态,无流光效果,正面的展示角度来对基础组件重新记载,生成并显示该第三模型,即为应用程序31的第三个应用页面中展示的“鼎的详细信息展示”。

[0169] 在一些实施方式中,请一并参阅图4c进行理解,用户可以通过新增基础组件的场景的操作,将场景标识改变,并通过对新的场景标识进行属性配置,实现配置新的场景对应的状态,进一步的,创建变量,将变量与场景标识进行绑定,以此,在后续用户点击切换按钮

之后,会自动修改相应的变量值,实现由过渡动画逐渐切换到基础组件新的场景下,实现单个组件的多场景流畅切换。

[0170] 由上述可知,本申请实施例通过响应于从组件区将基础组件拖拽至画布区的第一拖拽操作,在画布区上显示基础组件;为显示的基础组件配置在第一场景标识下对应的第一属性信息;响应于从组件区将基础组件拖拽至画布区上的第二拖拽操作,为显示的基础组件配置在第二场景标识下对应的第二属性信息;其中,第二属性信息中至少包括过渡状态子属性和显示状态子属性。以此,通过对同一个基础组件多次拖拽至画布区的拖拽操作,实现对同一个基础组件在多个场景标识下的多场景配置,并且还可以实现不同场景之间过渡状态子属性相应的过渡动画的配置,相对于相关技术通过拖拽多个组件实现多个场景状态切换的方案而言,本申请提升了组件在不同场景状态切换的流畅性,提升用户的使用体验。

[0171] 请参阅图5,图5为本申请实施例提供的组件配置装置的结构示意图,其中该组件配置装置可以包括显示单元301第一配置单元302和第二配置单元303,该组件配置装置应用于终端。

[0172] 显示单元301,用于响应于从组件区将基础组件拖拽至画布区的第一拖拽操作,在该画布区上显示该基础组件。

[0173] 在一些实施方式中,该组件配置装置,还包括:

[0174] 设置单元,用于为组件区中的每个预设组件设置相应的组件标识;

[0175] 该基础组件关联目标组件标识,该显示单元301,用于:

[0176] 获取该画布区上保存的预设组件相应的组件标识;

[0177] 当检测到该组件标识中不存在该目标组件标识时,在该画布区上新增该基础组件进行显示。

[0178] 第一配置单元302,用于为显示的基础组件配置在第一场景标识下对应的第一属性信息。

[0179] 在一些实施例中,该第一配置单元,用于:

[0180] 在属性配置区上展示显示的基础组件的第一场景标识以及相应的第一配置项;

[0181] 接收用户基于第一配置项输入的第一属性信息。

[0182] 在一些实施方式中,所述组件配置装置,还包括显示变化单元,用于:

[0183] 对所述画布区上显示的基础组件进行显示变化处理;

[0184] 其中,所述显示变化处理至少包括亮度变化处理、透明度变化处理或者闪动变动处理中的一种处理方式。

[0185] 第二配置单元303,用于响应于从组件区将基础组件拖拽至该画布区上的第二拖拽操作,为该显示的基础组件配置在第二场景标识下对应的第二属性信息。

[0186] 其中,该第二属性信息中至少包括过渡状态子属性和显示状态子属性。

[0187] 在一些实施例中,该第二配置单元303,包括:

[0188] 获取子单元,用于响应于从组件区将基础组件拖拽至该画布区上的第二拖拽操作,获取该画布区上保存的预设组件相应的组件标识;

[0189] 取消执行子单元,用于当检测到该组件标识中存在该目标组件标识时,不执行在该画布区上新增该基础组件的操作;

- [0190] 更改子单元,用于将该显示的基础组件的第一场景标识更改为第二场景标识;
- [0191] 配置子单元,用于配置在该第二场景标识下对应的第二属性信息。
- [0192] 在一些实施例中,该更改子单元,用于:
- [0193] 在属性配置区上将显示的基础组件对应的第一场景标识更改为第二场景标识;
- [0194] 生成该第二场景标识下对应的第二配置项。
- [0195] 在一些实施例中,该配置子单元,用于:
- [0196] 在该属性配置区上展示显示的基础组件的第二场景标识以及相应的第二配置项;
- [0197] 接收用户基于第二配置项输入的第二属性信息。
- [0198] 在一些实施例中,该组件配置装置,还包括:
- [0199] 第一生成单元,用于根据该第一场景标识下的第一属性信息对该基础组件进行加载,生成并显示第一模型;
- [0200] 切换单元,用于响应于对该第一模型进行切换的触发操作,将该第一场景标识切换至第二场景标识,并获取该第二场景标识下的第二属性信息;
- [0201] 第二生成单元,用于播放该过渡状态子属性相应的过渡动画,并根据该显示状态子属性对基础组件进行加载,生成并显示第二模型。
- [0202] 在一些实施例中,该组件配置装置,还包括关联单元,用于:
- [0203] 创建变量标识,该变量标识携带第一变量值;
- [0204] 将该第一变量值与该基础组件的第一场景标识关联。
- [0205] 在一些实施例中,该切换单元,用于:
- [0206] 响应于对该第一模型进行切换的触发操作,将该变量标识的第一变量值更改为第二变量值,并生成变更指令;
- [0207] 响应于该变更指令,将该第一场景标识切换至第二场景标识,并获取该第二场景标识下的第二属性信息。
- [0208] 以上各个单元的具体实施可参见前面的实施例,在此不再赘述。
- [0209] 由上述可知,本申请实施例通过显示单元301响应于从组件区将基础组件拖拽至画布区的第一拖拽操作,在画布区上显示基础组件;第一配置单元302为显示的基础组件配置在第一场景标识下对应的第一属性信息;第二配置单元303响应于从组件区将基础组件拖拽至画布区上的第二拖拽操作,为显示的基础组件配置在第二场景标识下对应的第二属性信息;其中,第二属性信息中至少包括过渡状态子属性和显示状态子属性。以此,通过对同一个基础组件多次拖拽至画布区的拖拽操作,实现对同一个基础组件在多个场景标识下的多场景配置,并且还可以实现不同场景之间过渡状态子属性相应的过渡动画的配置,相对于相关技术通过拖拽多个组件实现多个场景状态切换的方案而言,本申请提升了组件在不同场景状态切换的流畅性,提升用户的使用体验。
- [0210] 本申请实施例还提供一种计算机设备,该计算机设备可以为终端,如图6所示,其示出了本申请实施例所涉及的终端的结构示意图,具体来讲:
- [0211] 该计算机设备可以包括射频(RF, Radio Frequency)电路401、包括有一个或一个以上计算机可读存储介质的存储器402、输入单元403、显示单元404、传感器405、音频电路406、无线保真(WiFi, Wireless Fidelity)模块407、包括有一个或者一个以上处理核心的处理器408、以及电源409等部件。本领域技术人员可以理解,图6中示出的终端结构并不构

成对终端的限定,可以包括比图示更多或更少的部件,或者组合某些部件,或者不同的部件布置。其中:

[0212] RF电路401可用于收发信息或通话过程中,信号的接收和发送,特别地,将基站的下行信息接收后,交由一个或者一个以上处理器408处理;另外,将涉及上行的数据发送给基站。通常,RF电路401包括但不限于天线、至少一个放大器、调谐器、一个或多个振荡器、用户身份模块(SIM,Subscriber Identity Module)卡、收发信机、耦合器、低噪声放大器(LNA,Low Noise Amplifier)、双工器等。此外,RF电路401还可以通过无线通信与网络和其他设备通信。无线通信可以使用任一通信标准或协议,包括但不限于全球移动通讯系统(GSM,Global System of Mobile communication)、通用分组无线服务(GPRS,General Packet Radio Service)、码分多址(CDMA,Code Division Multiple Access)、宽带码分多址(WCDMA,Wideband Code Division Multiple Access)、长期演进(LTE,Long Term Evolution)、电子邮件、短消息服务(SMS,Short Messaging Service)等。

[0213] 存储器402可用于存储软件程序以及模块,处理器408通过运行存储在存储器402的软件程序以及模块,从而执行各种功能应用以及会话群组的管理。存储器402可主要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、至少一个功能所需的应用程序(比如声音播放功能、图像播放功能等)等;存储数据区可存储根据终端的使用所创建的数据(比如音频数据、电话本等)等。此外,存储器402可以包括高速随机存取存储器,还可以包括非易失性存储器,例如至少一个磁盘存储器件、闪存器件、或其他易失性固态存储器件。相应地,存储器402还可以包括存储器控制器,以提供处理器408和输入单元403对存储器402的访问。

[0214] 输入单元403可用于接收输入的数字或字符信息,以及产生与用户设置以及功能控制有关的键盘、鼠标、操作杆、光学或者轨迹球信号输入。具体地,在一个具体的实施例中,输入单元403可包括触敏表面以及其他输入设备。触敏表面,也称为触摸显示屏或者触控板,可收集用户在其上或附近的触摸操作(比如用户使用手指、触笔等任何适合的物体或附件在触敏表面上或在触敏表面附近的操作),并根据预先设定的程式驱动相应的连接装置。可选的,触敏表面可包括触摸检测装置和触摸控制器两个部分。其中,触摸检测装置检测用户的触摸方位,并检测触摸操作带来的信号,将信号传送给触摸控制器;触摸控制器从触摸检测装置上接收触摸信息,并将它转换成触点坐标,再送给处理器408,并能接收处理器408发来的命令并加以执行。此外,可以采用电阻式、电容式、红外线以及表面声波等多种类型实现触敏表面。除了触敏表面,输入单元403还可以包括其他输入设备。具体地,其他输入设备可以包括但不限于物理键盘、功能键(比如音量控制按键、开关按键等)、轨迹球、鼠标、操作杆等中的一种或多种。

[0215] 显示单元404可用于显示由用户输入的信息或提供给用户的信息以及终端的各种图形用户接口,这些图形用户接口可以由图形、文本、图标、视频和其任意组合来构成。显示单元404可包括显示面板,可选的,可以采用液晶显示器(LCD,Liquid Crystal Display)、有机发光二极管(OLED,Organic Light-Emitting Diode)等形式来配置显示面板。进一步的,触敏表面可覆盖显示面板,当触敏表面检测到在其上或附近的触摸操作后,传送给处理器408以确定触摸事件的类型,随后处理器408根据触摸事件的类型在显示面板提供相应的视觉输出。虽然在图6中,触敏表面与显示面板是作为两个独立的部件来实现输入和输入功

能,但是在某些实施例中,可以将触敏表面与显示面板集成而实现输入和输出功能。

[0216] 终端还可包括至少一种传感器405,比如光传感器、运动传感器以及其他传感器。具体地,光传感器可包括环境光传感器及接近传感器,其中,环境光传感器可根据环境光线的明暗来调节显示面板的亮度,接近传感器可在终端移动到耳边时,关闭显示面板和/或背光。作为运动传感器的一种,重力加速度传感器可检测各个方向上(一般为三轴)加速度的大小,静止时可检测出重力的大小及方向,可用于识别手机姿态的应用(比如横竖屏切换、相关游戏、磁力计姿态校准)、振动识别相关功能(比如计步器、敲击)等;至于终端还可配置的陀螺仪、气压计、湿度计、温度计、红外线传感器等其他传感器,在此不再赘述。

[0217] 音频电路406、扬声器,传声器可提供用户与终端之间的音频接口。音频电路406可将接收到的音频数据转换后的电信号,传输到扬声器,由扬声器转换为声音信号输出;另一方面,传声器将收集的声音信号转换为电信号,由音频电路406接收后转换为音频数据,再将音频数据输出处理器408处理后,经RF电路401以发送给比如另一终端,或者将音频数据输出至存储器402以便进一步处理。音频电路406还可能包括耳塞插孔,以提供外设耳机与终端的通信。

[0218] WiFi属于短距离无线传输技术,终端通过WiFi模块407可以帮助用户收发电子邮件、浏览网页和访问流式媒体等,它为用户提供了无线的宽带互联网访问。虽然图6示出了WiFi模块407,但是可以理解的是,其并不属于终端的必须构成,完全可以根据需要在不改变发明的本质的范围内而省略。

[0219] 处理器408是终端的控制中心,利用各种接口和线路连接整个手机的各个部分,通过运行或执行存储在存储器402内的软件程序和/或模块,以及调用存储在存储器402内的数据,执行终端的各种功能和处理数据,从而对手机进行整体监测。可选的,处理器408可包括一个或多个处理核心;优选的,处理器408可集成应用处理器和调制解调处理器,其中,应用处理器主要处理操作系统、用户界面和应用程序等,调制解调处理器主要处理无线通信。可以理解的是,上述调制解调处理器也可以不集成到处理器408中。

[0220] 终端还包括给各个部件供电的电源409(比如电池),优选的,电源可以通过电源管理系统与处理器408逻辑相连,从而通过电源管理系统实现管理充电、放电、以及功耗管理等功能。电源409还可以包括一个或一个以上的直流或交流电源、再充电系统、电源故障检测电路、电源转换器或者逆变器、电源状态指示器等任意组件。

[0221] 尽管未示出,终端还可以包括摄像头、蓝牙模块等,在此不再赘述。具体在本实施例中,终端中的处理器408会按照如下的指令,将一个或一个以上的应用程序的进程对应的可执行文件加载到存储器402中,并由处理器408来运行存储在存储器402中的应用程序,从而实现各种功能:

[0222] 响应于从组件区将基础组件拖拽至画布区的第一拖拽操作,在该画布区上显示该基础组件;

[0223] 为显示的基础组件配置在第一场景标识下对应的第一属性信息;

[0224] 响应于从组件区将基础组件拖拽至该画布区上的第二拖拽操作,为该显示的基础组件配置在第二场景标识下对应的第二属性信息;

[0225] 其中,该第二属性信息中至少包括过渡状态子属性和显示状态子属性。

[0226] 在上述实施例中,对各个实施例的描述都各有侧重,某个实施例中沒有详述的部

分,可以参见上文针对图像处理方法的详细描述,此处不再赘述。

[0227] 由上述可知,本申请实施例的计算机设备可以通过响应于从组件区将基础组件拖拽至画布区的第一拖拽操作,在画布区上显示基础组件;为显示的基础组件配置在第一场景标识下对应的第一属性信息;响应于从组件区将基础组件拖拽至画布区上的第二拖拽操作,为显示的基础组件配置在第二场景标识下对应的第二属性信息;其中,第二属性信息中至少包括过渡状态子属性和显示状态子属性。以此,通过对同一个基础组件多次拖拽至画布区的拖拽操作,实现对同一个基础组件在多个场景标识下的多场景配置,并且还可以实现不同场景之间过渡状态子属性相应的过渡动画的配置,相对于相关技术通过拖拽多个组件实现多个场景状态切换的方案而言,本申请提升了组件在不同场景状态切换的流畅性,提升用户的使用体验。

[0228] 本领域普通技术人员可以理解,上述实施例的各种方法中的全部或部分步骤可以通过指令来完成,或通过指令控制相关的硬件来完成,该指令可以存储于一计算机可读存储介质中,并由处理器进行加载和执行。

[0229] 为此,本申请实施例提供一种计算机可读存储介质,其中存储有多条指令,该指令能够被处理器进行加载,以执行本申请实施例所提供的任一种组件配置方法中的步骤。例如,该指令可以执行如下步骤:

[0230] 响应于从组件区将基础组件拖拽至画布区的第一拖拽操作,在该画布区上显示该基础组件;

[0231] 为显示的基础组件配置在第一场景标识下对应的第一属性信息;

[0232] 响应于从组件区将基础组件拖拽至该画布区上的第二拖拽操作,为该显示的基础组件配置在第二场景标识下对应的第二属性信息;

[0233] 其中,该第二属性信息中至少包括过渡状态子属性和显示状态子属性。

[0234] 根据本申请的一个方面,提供了一种计算机程序产品或计算机程序,该计算机程序产品或计算机程序包括计算机指令,该计算机指令存储在计算机可读存储介质中。计算机设备的处理器从计算机可读存储介质读取该计算机指令,处理器执行该计算机指令,使得该计算机设备执行上述实施例提供的各种可选实现方式中提供的方法。

[0235] 其中,该计算机可读存储介质可以包括:只读存储器(ROM,Read Only Memory)、随机存取记忆体(RAM,Random Access Memory)、磁盘或光盘等。

[0236] 由于该计算机可读存储介质中所存储的指令,可以执行本申请实施例所提供的任一种组件配置方法中的步骤,因此,可以实现本申请实施例所提供的任一种组件配置方法所能实现的有益效果,详见前面的实施例,在此不再赘述。

[0237] 以上对本申请实施例所提供的一种组件配置方法、装置及计算机可读存储介质进行了详细介绍,本文中应用了具体个例对本申请的原理及实施方式进行了阐述,以上实施例的说明只是用于帮助理解本申请的方法及其核心思想;同时,对于本领域的技术人员,依据本申请的思想,在具体实施方式及应用范围上均会有改变之处,综上所述,本说明书内容不应理解为对本申请的限制。

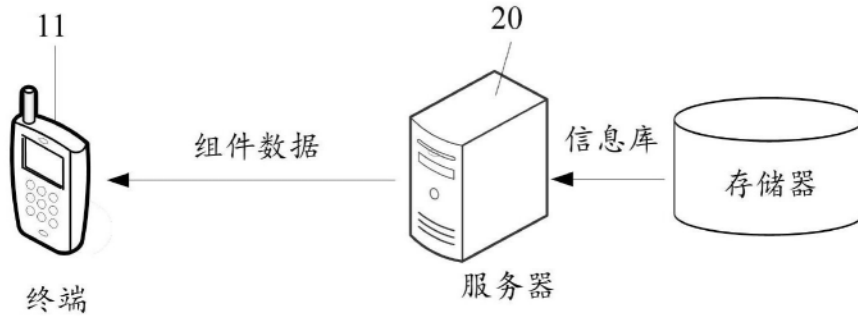


图1a

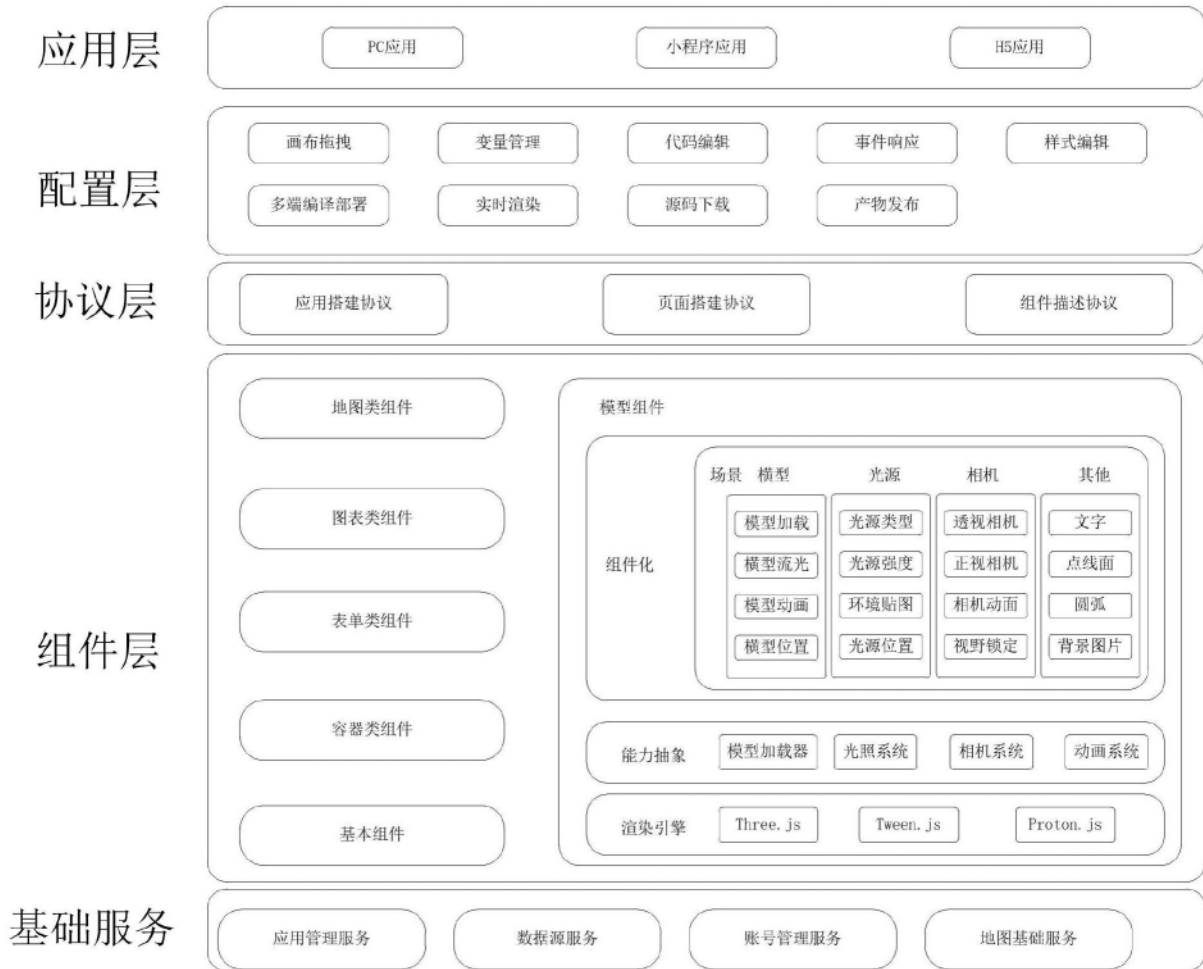


图1b

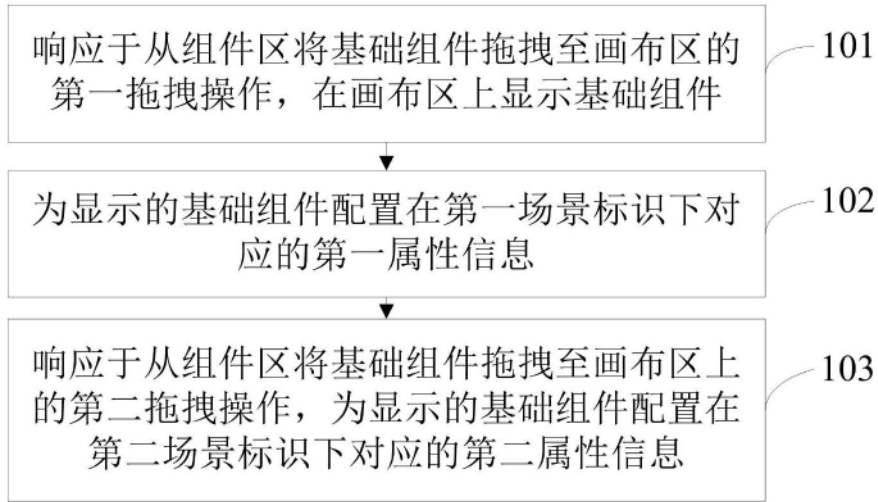


图2

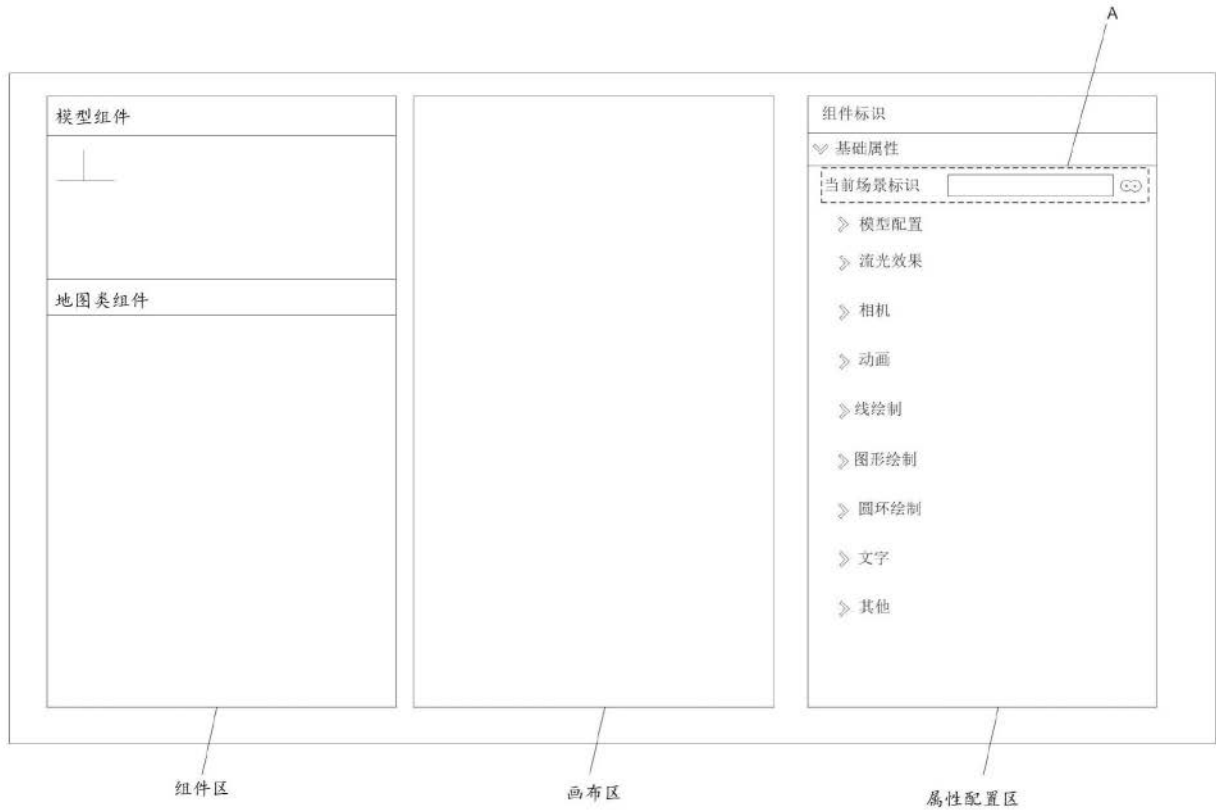


图3a

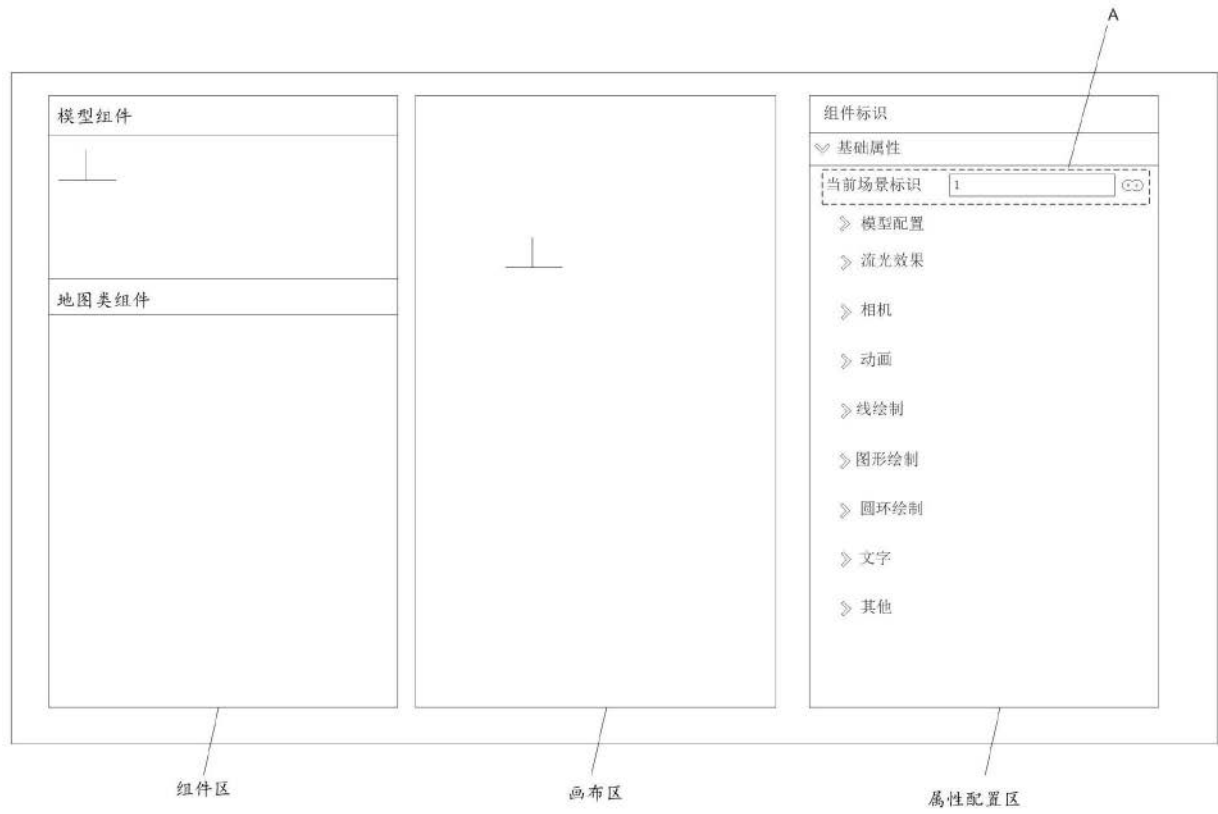


图3b

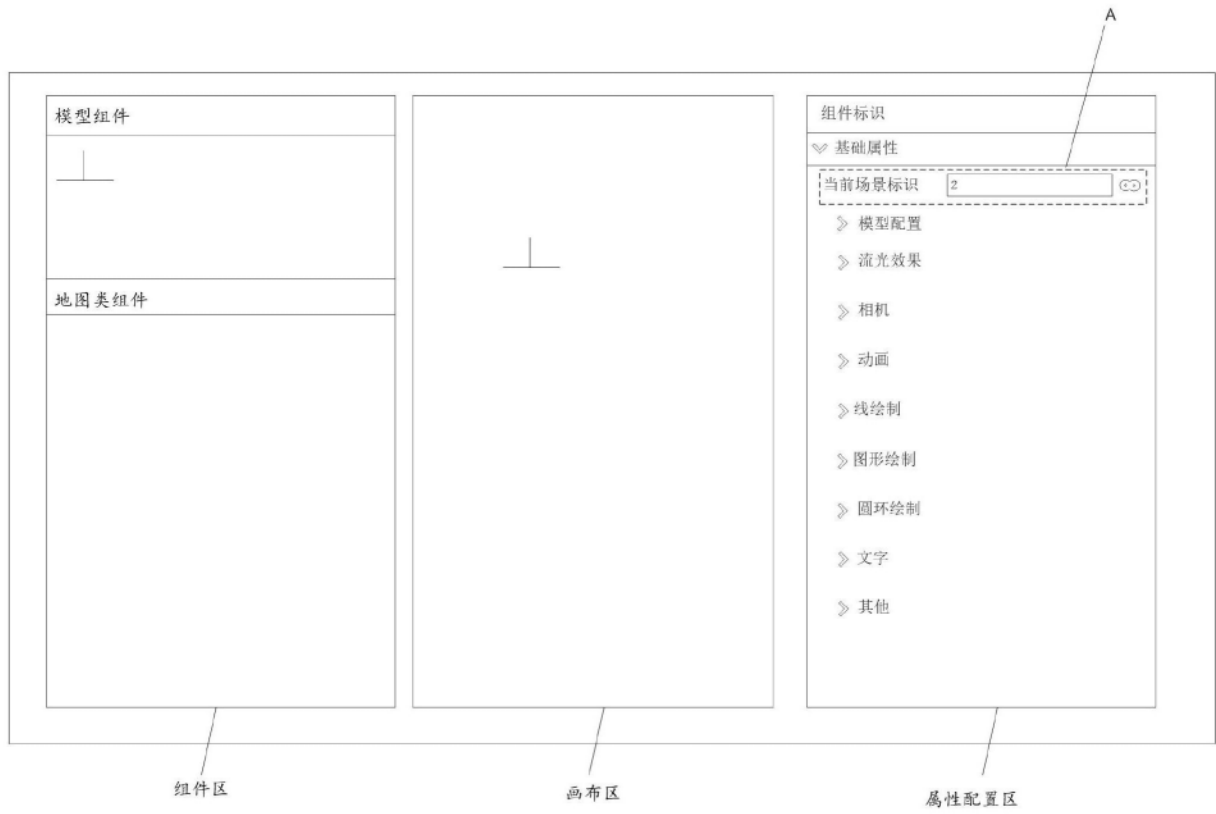


图3c

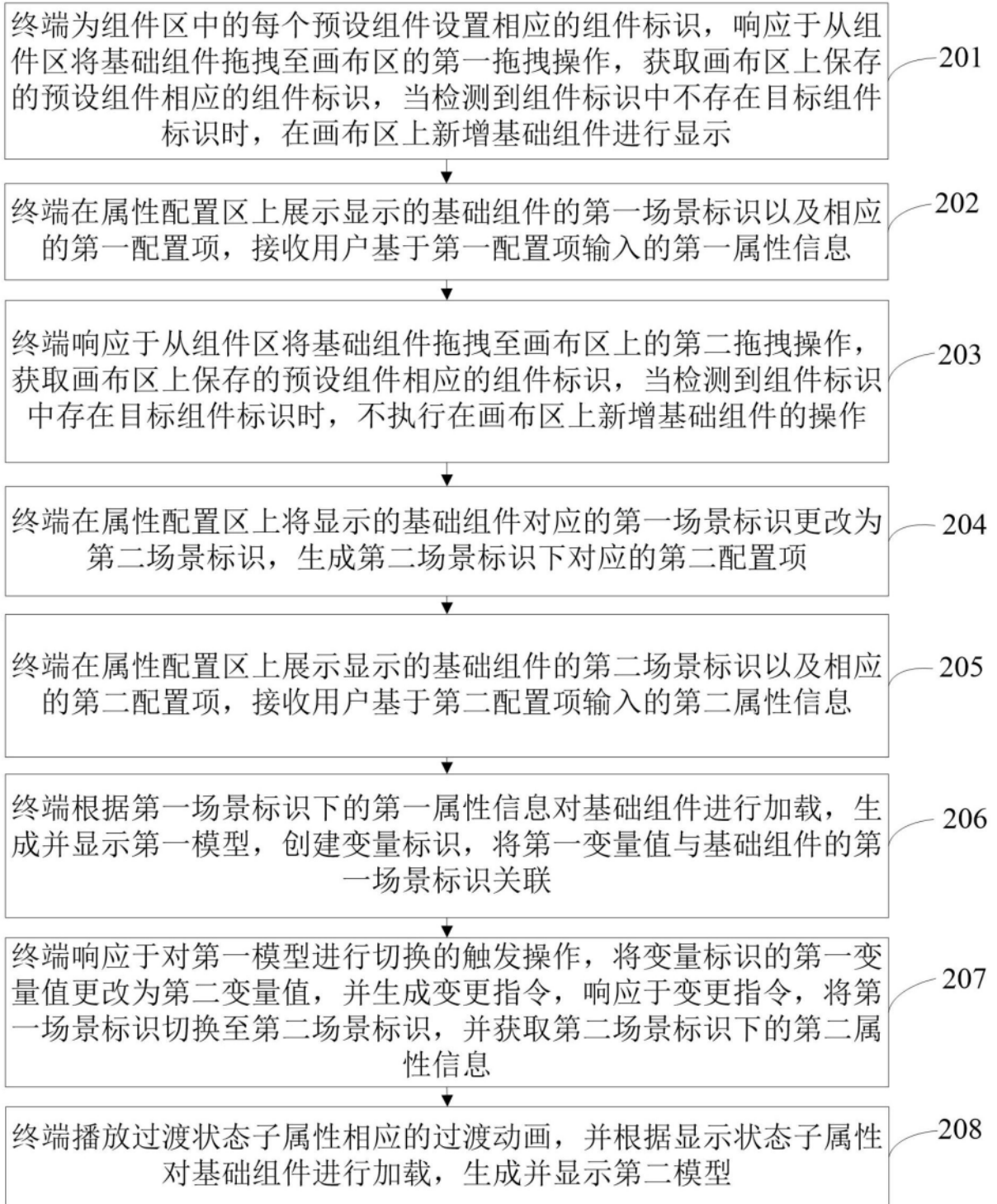


图4a

31

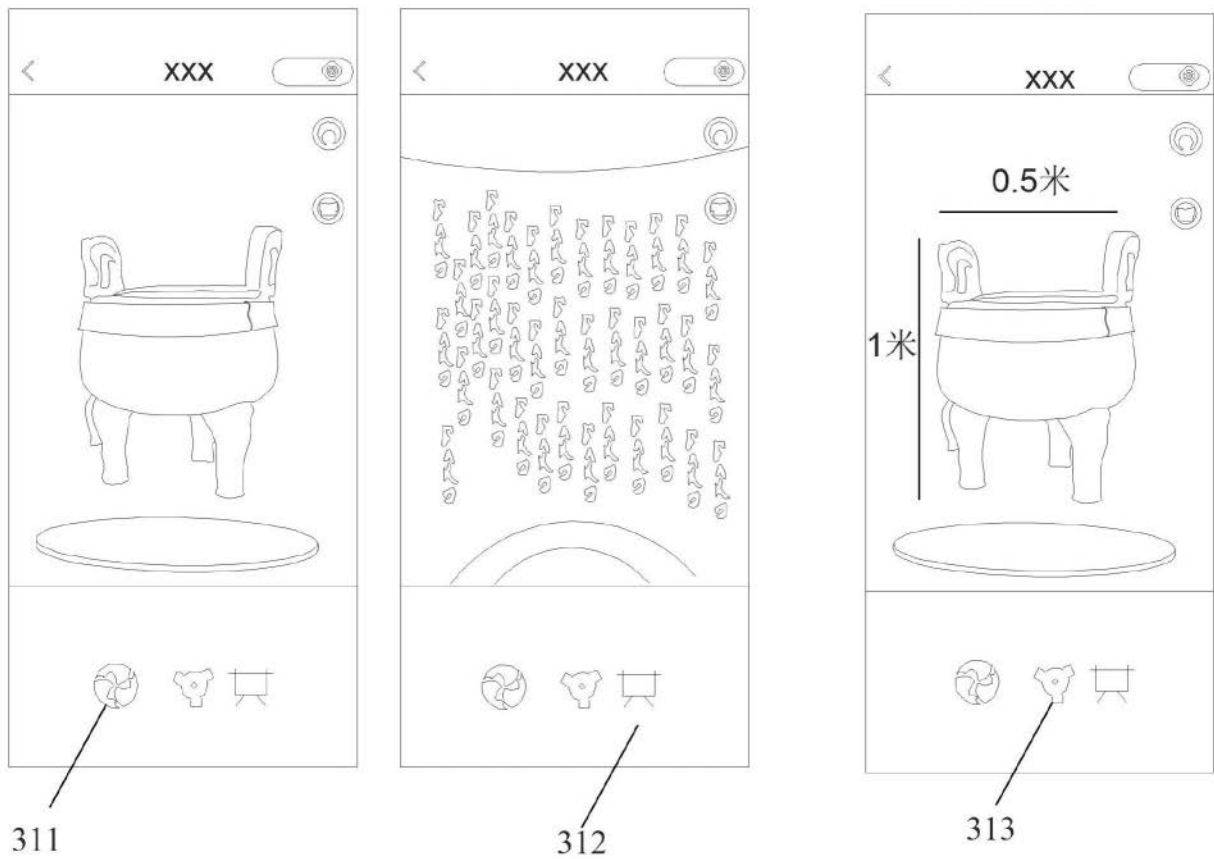


图4b

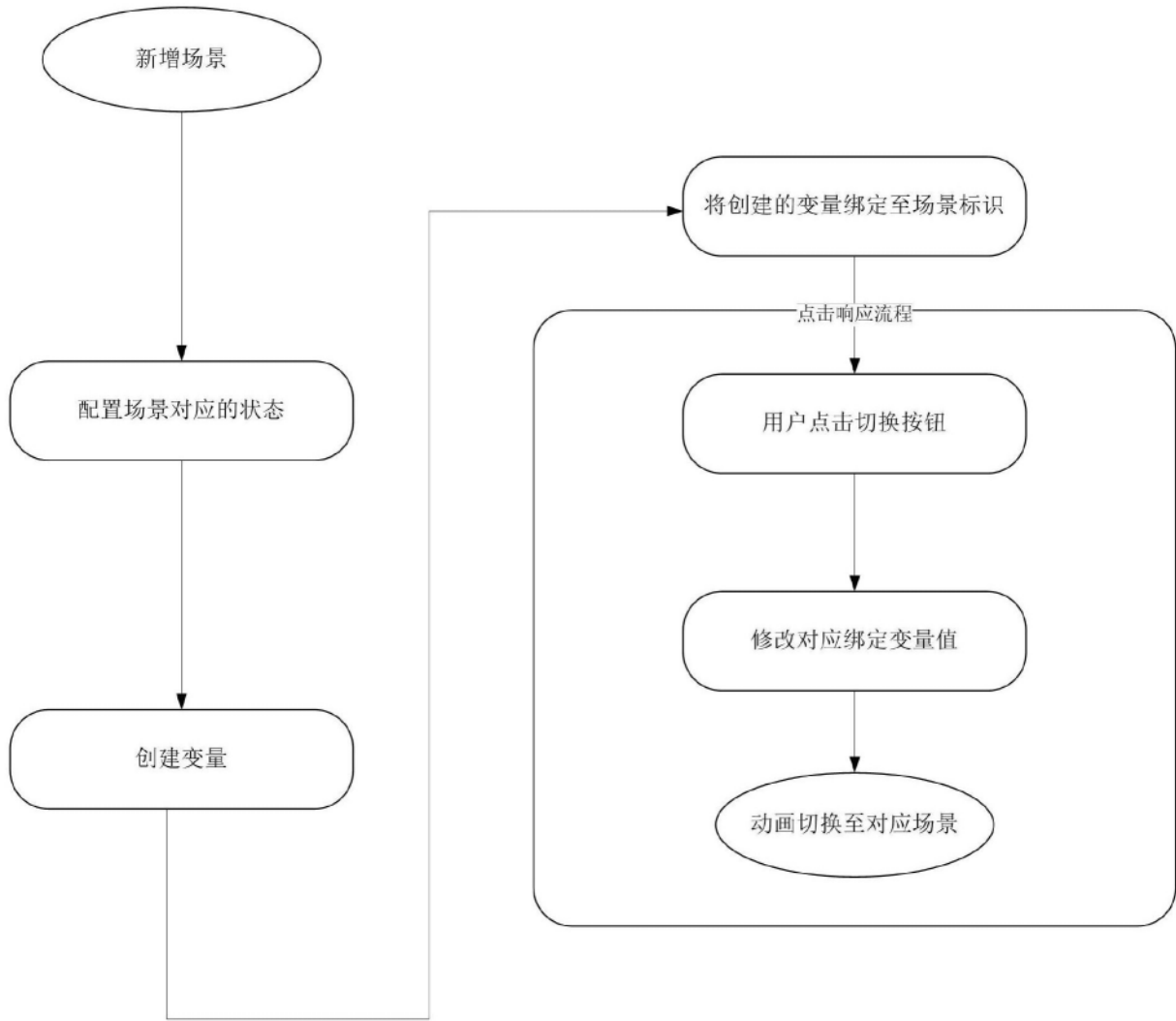


图4c

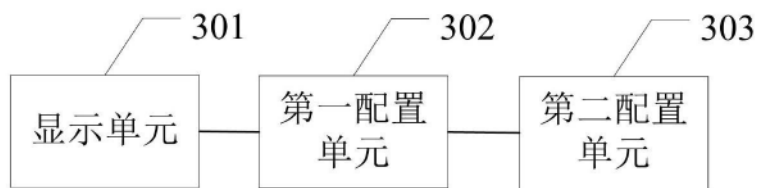


图5

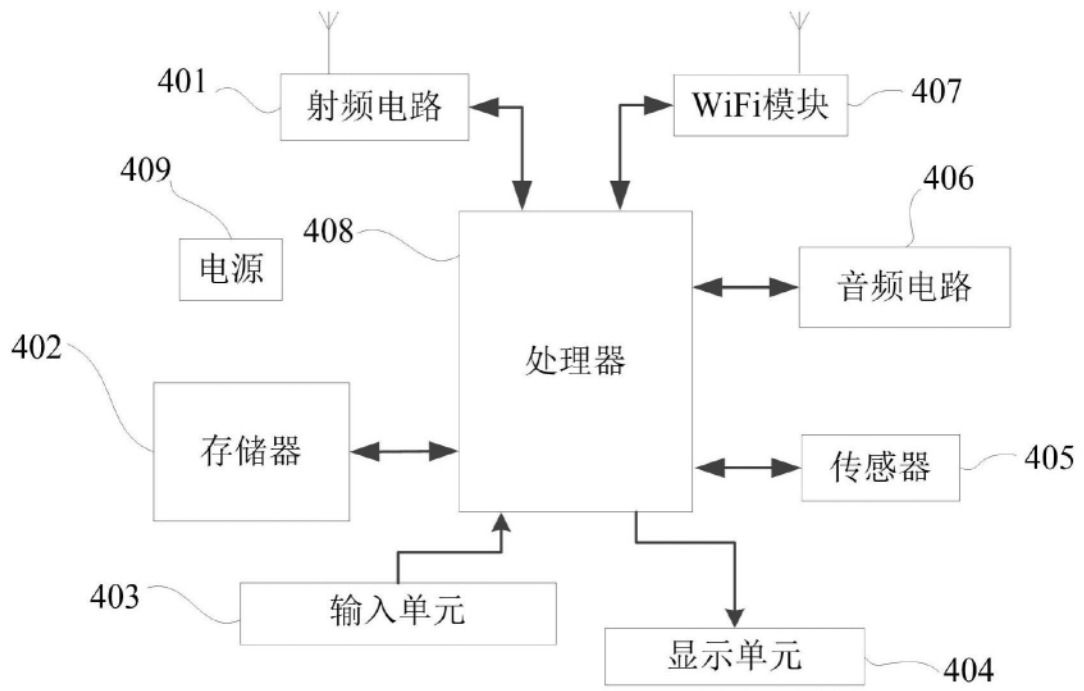


图6