



(12) 发明专利

(10) 授权公告号 CN 110136230 B

(45) 授权公告日 2023. 09. 05

(21) 申请号 201910253191.5

(22) 申请日 2019.03.29

(65) 同一申请的已公布的文献号

申请公布号 CN 110136230 A

(43) 申请公布日 2019.08.16

(73) 专利权人 北京达佳互联信息技术有限公司

地址 100085 北京市海淀区上地西路6号1

幢1层101D1-7

(72) 发明人 张一磊

(74) 专利代理机构 北京润泽恒知识产权代理有

限公司 11319

专利代理师 莎日娜

(51) Int. Cl.

G06T 13/00 (2011.01)

(56) 对比文件

CN 109359262 A, 2019.02.19

CN 101484921 A, 2009.07.15

JP 2000011199 A, 2000.01.14

US 2015370444 A1, 2015.12.24

审查员 孟子山

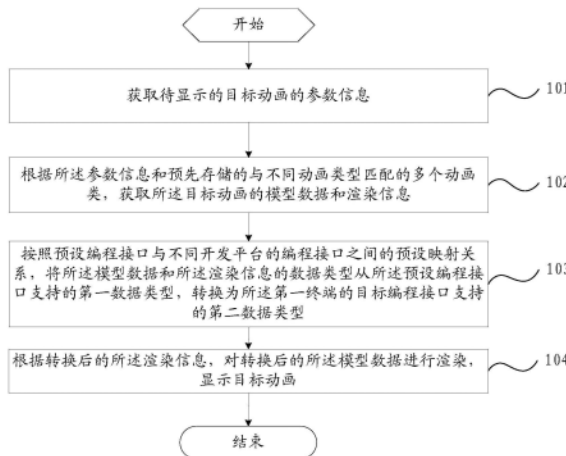
权利要求书3页 说明书13页 附图6页

(54) 发明名称

动画显示方法、装置、电子设备及存储介质

(57) 摘要

本公开是关于一种动画显示方法、装置、电子设备及存储介质,该动画显示方法包括:获取待显示的目标动画的参数信息,其中,目标动画为安装在第二终端上的动画应用程序所绘制的动画;根据参数信息和预先存储的与不同动画类型匹配的多个动画类,获取目标动画的模型数据和渲染信息;按照预设编程接口与不同开发平台的编程接口之间的预设映射关系,将所述模型数据和所述渲染信息的数据类型从所述预设编程接口支持的第一数据类型,转换为所述第一终端的目标编程接口支持的第二数据类型;根据转换后的渲染信息,对转换后的所述模型数据进行渲染,显示目标动画。本公开能够跨平台显示目标动画,无需修改代码。



1. 一种动画显示方法,其特征在于,应用于具有目标编程接口的第一终端,所述方法包括:

获取待显示的目标动画的参数信息,其中,所述目标动画为安装在第二终端上的动画应用程序所绘制的动画;

根据所述参数信息和预先存储的与不同动画类型匹配的多个动画类,获取所述目标动画的模型数据和渲染信息,其中,所述多个动画类为多个描述动画的类文件,所述动画类用于描述对应动画类型的动画参数和渲染信息;

按照预设编程接口与不同开发平台的编程接口之间的预设映射关系,将所述模型数据和所述渲染信息的数据类型从所述预设编程接口支持的第一数据类型,转换为所述第一终端的目标编程接口支持的第二数据类型;

根据转换后的所述渲染信息,对转换后的所述模型数据进行渲染,显示目标动画。

2. 根据权利要求1所述的动画显示方法,其特征在于,

所述获取待显示的目标动画的参数信息,包括:

获取所述目标动画的配置文件,其中,所述配置文件为安装在所述第二终端上的动画应用程序的与所述目标动画对应的配置文件;

根据所述配置文件,获取所述目标动画的参数信息。

3. 根据权利要求1所述的动画显示方法,其特征在于,所述根据所述参数信息和预先存储的与不同动画类型匹配的多个动画类,获取所述目标动画的模型数据和渲染信息,包括:

根据所述参数信息和预先存储的与不同动画类型匹配的多个动画类,获取与所述目标动画匹配的多个动画对象;

根据所述参数信息,获取所述多个动画对象中每个动画对象的模型数据、第一渲染信息,以及用于描述所述多个动画对象中不同动画对象之间动画关系的第二渲染信息。

4. 根据权利要求3所述的动画显示方法,其特征在于,所述目标动画由一个或多个原子动画构成,所述参数信息包括与所述目标动画对应的每个原子动画的动画类型、每个原子动画的动画参数、每个原子动画的模型标识以及不同原子动画之间的关联参数;

所述根据所述参数信息和预先存储的与不同动画类型匹配的多个动画类,获取与所述目标动画匹配的多个动画对象,包括:

根据预先存储的与不同动画类型匹配的多个动画类,按照所述参数信息中所述每个原子动画的所述动画类型和所述动画参数,创建与所述目标动画匹配的多个动画对象,其中,每个动画对象分别配置有对应的所述动画参数;

所述根据所述参数信息,获取所述多个动画对象中每个动画对象的模型数据、第一渲染信息,以及用于描述所述多个动画对象中不同动画对象之间动画关系的第二渲染信息,包括:

根据所述参数信息中与所述目标动画对应的每个原子动画的模型标识,获取所述多个动画对象中每个动画对象的模型数据;

根据所述每个动画对象配置的动画参数,生成每个动画对象的第一渲染信息;

按照所述目标动画对应的不同原子动画之间的关联参数,创建所述多个动画对象中不同动画对象之间的关联信息;

根据所述关联信息生成用于描述所述多个动画对象中不同动画对象之间动画关系的

第二渲染信息。

5. 根据权利要求4所述的动画显示方法,其特征在于,

所述根据所述参数信息中与所述目标动画对应的每个原子动画的模型标识,获取所述多个动画对象中每个动画对象的模型数据,包括:

获取模型文件;

根据所述参数信息中与所述目标动画对应的每个原子动画的模型标识,从所述模型文件中获取所述多个动画对象中每个动画对象的模型数据。

6. 根据权利要求1所述的动画显示方法,其特征在于,所述多个动画类对应的动画类型为所述动画应用程序支持的动画类型。

7. 一种动画显示装置,其特征在于,应用于具有目标编程接口的第一终端,包括:

第一获取模块,被配置为获取待显示的目标动画的参数信息,其中,所述目标动画为安装在第二终端上的动画应用程序所绘制的动画;

第二获取模块,被配置为根据所述参数信息和预先存储的与不同动画类型匹配的多个动画类,获取所述目标动画的模型数据和渲染信息,其中,所述多个动画类为多个描述动画的类文件,所述动画类用于描述对应动画类型的动画参数和渲染信息;

映射模块,被配置为按照预设编程接口与不同开发平台的编程接口之间的预设映射关系,将所述模型数据和所述渲染信息的数据类型从所述预设编程接口支持的第一数据类型,转换为所述第一终端的目标编程接口支持的第二数据类型;

渲染模块,被配置为根据转换后的所述渲染信息,对转换后的所述模型数据进行渲染,显示目标动画。

8. 根据权利要求7所述的动画显示装置,其特征在于,

所述第一获取模块包括:

第一获取子模块,被配置为获取目标动画的配置文件,其中,所述配置文件为安装在所述第二终端上的动画应用程序的与所述目标动画对应的配置文件;

第二获取子模块,被配置为根据所述配置文件,获取所述目标动画的参数信息。

9. 根据权利要求7所述的动画显示装置,其特征在于,所述第二获取模块包括:

第三获取子模块,被配置为根据所述参数信息和预先存储的与不同动画类型匹配的多个动画类,获取与所述目标动画匹配的多个动画对象;

第四获取子模块,被配置为根据所述参数信息,获取所述多个动画对象中每个动画对象的模型数据、第一渲染信息,以及用于描述所述多个动画对象中不同动画对象之间动画关系的第二渲染信息。

10. 根据权利要求9所述的动画显示装置,其特征在于,所述目标动画由一个或多个原子动画构成,所述参数信息包括与所述目标动画对应的每个原子动画的动画类型、每个原子动画的动画参数、每个原子动画的模型标识以及不同原子动画之间的关联参数;

所述第三获取子模块包括:

第一创建单元,被配置为根据预先存储的与不同动画类型匹配的多个动画类,按照所述参数信息中所述每个原子动画的所述动画类型和所述动画参数,创建与所述目标动画匹配的多个动画对象,其中,每个动画对象分别配置有对应的所述动画参数;

所述第四获取子模块包括:

第一获取单元,被配置为根据所述参数信息中与所述目标动画对应的每个原子动画的模型标识,获取所述多个动画对象中每个动画对象的模型数据;

第一生成单元,被配置为根据所述每个动画对象配置的动画参数,生成每个动画对象的第一渲染信息;

第二创建单元,被配置为按照所述目标动画对应的不同原子动画之间的关联参数,创建所述多个动画对象中不同动画对象之间的关联信息;

第二生成单元,被配置为根据所述关联信息生成用于描述所述多个动画对象中不同动画对象之间动画关系的第二渲染信息。

11. 根据权利要求10所述的动画显示装置,其特征在于,

所述第一获取单元包括:

第一获取子单元,被配置为获取模型文件;

第二获取子单元,被配置为根据所述参数信息中与所述目标动画对应的每个原子动画的模型标识,从所述模型文件中获取所述多个动画对象中每个动画对象的模型数据。

12. 根据权利要求7所述的动画显示装置,其特征在于,所述多个动画类对应的动画类型为所述动画应用程序支持的动画类型。

13. 一种电子设备,其特征在于,包括:

处理器;

用于存储所述处理器可执行指令的存储器;

其中,所述处理器被配置为执行以实现如权利要求1至权利要求6中任意一项所述的动画显示方法所执行的操作。

14. 一种非临时性计算机可读存储介质,其特征在于,当所述存储介质中的指令由电子设备的处理器执行时,使得所述电子设备能够执行一种以实现如权利要求1至权利要求6中任意一项所述的动画显示方法所执行的操作。

动画显示方法、装置、电子设备及存储介质

技术领域

[0001] 本公开涉及图形处理技术领域,尤其涉及一种动画显示方法、装置、电子设备及存储介质。

背景技术

[0002] 目前,计算机终端侧具有很多类型的动画制作软件,设计师可以使用计算机终端侧的动画制作软件(例如AfterEffects,简称AE,是Adobe公司推出的一款图形视频处理软件)设计制作各种动画特效。

[0003] 那么如果需要将计算机终端侧的动画制作软件制作的动画,显示在移动终端上,则需要开发人员根据该移动终端的操作系统支持的API(Application Programming Interface,应用程序编程接口),重新编写一个动画APP(应用程序,Application)安装在该移动终端上使用。

[0004] 但是不同操作系统的编程语言不同、所支持的API也不同,因此,针对同一个动画特效,需要熟悉各个操作系统开发语言的不同开发人员分别针对各个平台的终端开发出多套动画APP,以在不同平台的终端上显示同一种动画特效。

[0005] 显然,相关技术中的动画实现方案存在着动画代码难以复用、动画代码维护难度大、以及动画代码开发成本高的问题。

发明内容

[0006] 为克服相关技术中的动画实现方案存在着动画代码难以复用、动画代码维护难度大、以及动画代码开发成本高的问题,本公开提供一种动画显示方法、装置、电子设备及存储介质。

[0007] 根据本公开实施例的第一方面,提供一种动画显示方法,应用于具有目标编程接口的第一终端,所述方法包括:

[0008] 获取待显示的目标动画的参数信息,其中,所述目标动画为安装在第二终端上的动画应用程序所绘制的动画;

[0009] 根据所述参数信息和预先存储的与不同动画类型匹配的多个动画类,获取所述目标动画的模型数据和渲染信息;

[0010] 按照预设编程接口与不同开发平台的编程接口之间的预设映射关系,将所述模型数据和所述渲染信息的数据类型从所述预设编程接口支持的第一数据类型,转换为所述第一终端的目标编程接口支持的第二数据类型;

[0011] 根据转换后的所述渲染信息,对转换后的所述模型数据进行渲染,显示目标动画。

[0012] 在一种可能的实施方式中,所述获取待显示的目标动画的参数信息,包括:

[0013] 获取目标动画的配置文件,其中,所述配置文件为安装在所述第二终端上的动画应用程序的与所述目标动画对应的配置文件;

[0014] 根据所述配置文件,获取所述目标动画的参数信息。

[0015] 在一种可能的实施方式中,所述根据所述参数信息和预先存储的与不同动画类型匹配的多个动画类,获取所述目标动画的模型数据和渲染信息,包括:

[0016] 根据所述参数信息和预先存储的与不同动画类型匹配的多个动画类,获取与所述目标动画匹配的多个动画对象;

[0017] 根据所述参数信息,获取所述多个动画对象中每个动画对象的模型数据、第一渲染信息,以及用于描述所述多个动画对象中不同动画对象之间动画关系的第二渲染信息。

[0018] 在一种可能的实施方式中,所述目标动画由一个或多个原子动画构成,所述参数信息包括与所述目标动画对应的每个原子动画的动画类型、每个原子动画的动画参数、每个原子动画的模型标识以及不同原子动画之间的关联参数;

[0019] 所述根据所述参数信息和预先存储的与不同动画类型匹配的多个动画类,获取与所述目标动画匹配的多个动画对象,包括:

[0020] 根据预先存储的与不同动画类型匹配的多个动画类,按照所述参数信息中所述每个原子动画的所述动画类型和所述动画参数,创建与所述目标动画匹配的多个动画对象,其中,每个动画对象分别配置有对应的所述动画参数;

[0021] 所述根据所述参数信息,获取所述多个动画对象中每个动画对象的模型数据、第一渲染信息,以及用于描述所述多个动画对象中不同动画对象之间动画关系的第二渲染信息,包括:

[0022] 根据所述参数信息中与所述目标动画对应的每个原子动画的模型标识,获取所述多个动画对象中每个动画对象的模型数据;

[0023] 根据所述每个动画对象配置的动画参数,生成每个动画对象的第一渲染信息;

[0024] 按照所述目标动画对应的不同原子动画之间的关联参数,创建所述多个动画对象中不同动画对象之间的关联信息;

[0025] 根据所述关联信息生成用于描述所述多个动画对象中不同动画对象之间动画关系的第二渲染信息。

[0026] 在一种可能的实施方式中,所述根据所述参数信息中与所述目标动画对应的每个原子动画的模型标识,获取所述多个动画对象中每个动画对象的模型数据,包括:

[0027] 获取模型文件;

[0028] 根据所述参数信息中与所述目标动画对应的每个原子动画的模型标识,从所述模型文件中获取所述多个动画对象中每个动画对象的模型数据。

[0029] 在一种可能的实施方式中,所述多个动画类对应的动画类型为所述动画应用程序支持的动画类型。

[0030] 根据本公开实施例的第二方面,提供一种动画显示装置,应用于具有目标编程接口的第一终端,包括:

[0031] 第一获取模块,被配置为获取待显示的目标动画的参数信息,其中,所述目标动画为安装在第二终端上的动画应用程序所绘制的动画;

[0032] 第二获取模块,被配置为根据所述参数信息和预先存储的与不同动画类型匹配的多个动画类,获取所述目标动画的模型数据和渲染信息;

[0033] 映射模块,被配置为按照预设编程接口与不同开发平台的编程接口之间的预设映射关系,将所述模型数据和所述渲染信息的数据类型从所述预设编程接口支持的第一数据

类型,转换为所述第一终端的目标编程接口支持的第二数据类型;

[0034] 渲染模块,被配置为根据转换后的所述渲染信息,对转换后的所述模型数据进行渲染,显示目标动画。

[0035] 在一种可能的实施方式中,所述第一获取模块包括:

[0036] 第一获取子模块,被配置为获取目标动画的配置文件,其中,所述配置文件为安装在所述第二终端上的动画应用程序的与所述目标动画对应的配置文件;

[0037] 第二获取子模块,被配置为根据所述配置文件,获取所述目标动画的参数信息。

[0038] 在一种可能的实施方式中,所述第二获取模块包括:

[0039] 第三获取子模块,被配置为根据所述参数信息和预先存储的与不同动画类型匹配的多个动画类,获取与所述目标动画匹配的多个动画对象;

[0040] 第四获取子模块,被配置为根据所述参数信息,获取所述多个动画对象中每个动画对象的模型数据、第一渲染信息,以及用于描述所述多个动画对象中不同动画对象之间动画关系的第二渲染信息。

[0041] 在一种可能的实施方式中,所述目标动画由一个或多个原子动画构成,所述参数信息包括与所述目标动画对应的每个原子动画的动画类型、每个原子动画的动画参数、每个原子动画的模型标识以及不同原子动画之间的关联参数;

[0042] 所述第三获取子模块包括:

[0043] 第一创建单元,被配置为根据预先存储的与不同动画类型匹配的多个动画类,按照所述参数信息中所述每个原子动画的所述动画类型和所述动画参数,创建与所述目标动画匹配的多个动画对象,其中,每个动画对象分别配置有对应的所述动画参数;

[0044] 所述第四获取子模块包括:

[0045] 第一获取单元,被配置为根据所述参数信息中与所述目标动画对应的每个原子动画的模型标识,获取所述多个动画对象中每个动画对象的模型数据;

[0046] 第一生成单元,被配置为根据所述每个动画对象配置的动画参数,生成每个动画对象的第一渲染信息;

[0047] 第二创建单元,被配置为按照所述目标动画对应的不同原子动画之间的关联参数,创建所述多个动画对象中不同动画对象之间的关联信息;

[0048] 第二生成单元,被配置为根据所述关联信息生成用于描述所述多个动画对象中不同动画对象之间动画关系的第二渲染信息。

[0049] 在一种可能的实施方式中,所述第一获取单元包括:

[0050] 第一获取子单元,被配置为获取模型文件;

[0051] 第二获取子单元,被配置为根据所述参数信息中与所述目标动画对应的每个原子动画的模型标识,从所述模型文件中获取所述多个动画对象中每个动画对象的模型数据。

[0052] 在一种可能的实施方式中,所述多个动画类对应的动画类型为所述动画应用程序支持的动画类型。

[0053] 根据本公开实施例的第三方面,提供一种电子设备,包括:

[0054] 处理器;

[0055] 用于存储处理器可执行指令的存储器;

[0056] 其中,所述处理器被配置为执行以实现上述任意一项所述的动画显示方法所执行

的操作。

[0057] 根据本公开实施例的第四方面,提供一种非临时性计算机可读存储介质,当所述存储介质中的指令由电子设备的处理器执行时,使得电子设备能够执行一种以实现如上述任意一项所述的动画显示方法所执行的操作。

[0058] 根据本公开实施例的第五方面,提供一种应用程序,所述应用程序由电子设备的处理器执行时,使得电子设备能够执行一种以实现如上述任意一项所述的动画显示方法所执行的操作。

[0059] 本公开的实施例提供的技术方案可以包括以下有益效果:

[0060] 这样,本公开实施例通过获取待显示的目标动画的参数信息,并结合预先存储的与不同动画类型匹配的多个动画类,来获取目标动画的模型数据和渲染数据,然后,按照预设编程接口与不同开发平台的编程接口的预设映射关系,来将所述模型数据和渲染数据的数据类型均从预设编程接口支持的第一数据类型,转换为第一终端的目标编程接口所支持的第二数据类型,从而利用转换后的数据完成目标动画的显示。本公开实施例的方法可以分别应用于编程接口不同于该预设编程接口的各个第一终端,而无需修改动画代码,实现了对动画代码的复用,并降低了动画代码的维护难度以及开发成本,无需针对不同开发平台针对同一个动画分别进行多套代码实现。

[0061] 应当理解的是,以上的一般描述和后文的细节描述仅是示例性和解释性的,并不能限制本公开。

附图说明

[0062] 此处的附图被并入说明书中并构成本说明书的一部分,示出了符合本公开的实施例,并与说明书一起用于解释本公开的原理。

[0063] 图1是根据一示例性实施例示出的一种动画显示方法的流程图;

[0064] 图2是根据一示例性实施例示出的一种动画显示方法的流程图;

[0065] 图3是根据一示例性实施例示出的一种动画显示方法的流程图;

[0066] 图4是根据一示例性实施例示出的一种动画显示装置的结构框图;

[0067] 图5是根据一示例性实施例示出的一种用于动画显示的装置的框图;

[0068] 图6是根据一示例性实施例示出的一种用于动画显示的装置的框图。

具体实施方式

[0069] 这里将详细地对示例性实施例进行说明,其示例表示在附图中。下面的描述涉及附图时,除非另有表示,不同附图中的相同数字表示相同或相似的要素。以下示例性实施例中所描述的实施方式并不代表与本公开相一致的所有实施方式。相反,它们仅是与如所附权利要求书中所详述的、本公开的一些方面相一致的装置和方法的例子。

[0070] 图1是根据一示例性实施例示出的一种动画显示方法的流程图,该动画显示方法应用于具有目标编程接口的第一终端,该方法具体可以包括以下步骤:

[0071] 步骤101,获取待显示的目标动画的参数信息;

[0072] 其中,所述目标动画为安装在第二终端(例如计算机端)的动画应用程序(例如AE)所绘制的动画。那么为了能够使该目标动画显示在编程接口不同于第二终端的编程接口的

各个终端上,本公开实施例开发了一款通用于各个编程接口的目标APP(即目标动画应用程序),该目标APP可以获取需要显示的目标动画的参数信息,其中,该参数信息为该目标动画在已有的动画应用程序(例如安装在上述计算机端的AE)中的参数信息,该参数信息可以描述该目标动画。

[0073] 需要说明的是,本公开对于安装在第二终端上的已开发好的所述动画应用程序不做限制,其可以是任意一个已开发好的动画应用程序,设计师可以使用该动画应用程序制作目标动画。

[0074] 本发明实施例的方法的目的在于将该已开发好的动画应用程序所支持的各个动画重新开发,并使得重新开放的目标动画应用程序可以适用于各个编程接口的移动终端所安装使用,从而可以在具有各个编程接口的移动终端上绘制出该已开发好的动画应用程序所支持的各个动画,避免同一套动画代码的重复开发。

[0075] 在一种可能的实施方式中,在执行步骤101时,可以获取目标动画的配置文件;然后,根据所述配置文件,获取所述目标动画的参数信息。

[0076] 其中,所述配置文件为安装在所述第二终端上的动画应用程序的与所述目标动画对应的配置文件。

[0077] 后文以第二终端为计算机端,第一终端为移动终端为例来进行说明。但是根据应用场景的不同,本发明对于第一终端和第二终端的终端类型不做具体限制,可以都是移动终端,也可以都是计算机端,还可以一个是计算机端,另一个是移动终端,但是需要说明的是,第一终端和第二终端所支持的编程接口不同,或者说二者的操作系统不同。

[0078] 为了获取目标动画的参数信息,可以将该目标动画在计算机端的动画应用程序上的配置文件(例如json文件)读到内存,然后,本公开实施例的目标APP中设置有解析器,可以对内存中配置文件中的数据进行解析,从而获取到该目标动画的参数信息。

[0079] 这样,本公开实施例只需要获取待显示的目标动画在第二终端的动画应用程序上的配置文件,就可以根据该配置文件来获取到该目标动画的参数信息,所需要的外部数据较少,就可以基于该参数信息以及预先存储的多个动画类,来获取到该目标动画的模型数据和渲染信息,从而完成对模型数据和渲染信息的映射,并利用映射后的数据在第一终端上显示目标动画,操作简单,所需要的外部数据较少,且不需要根据不同平台重复开发,节省了开发成本。

[0080] 步骤102,根据所述参数信息和预先存储的与不同动画类型匹配的多个动画类,获取所述目标动画的模型数据和渲染信息;

[0081] 在一种可能的实施方式中,所述多个动画类对应的动画类型为安装在第二终端的所述动画应用程序支持的动画类型。

[0082] 由于AE可以制作的动画特效种类繁多,那么为了能够使本公开实施例的安装在第一终端的目标APP也可以实现对各种动画类型的动画的显示,本公开实施例的目标APP中包括动画系统,该动画系统预先存储了多个动画类,即多个描述动画的类(Class)文件。其中,每一个类文件描述一种类型的动画(即后文所述的原子动画),因此称作动画类。且不同动画类对应不同的动画类型,而该多个动画类所对应的多个动画类型均可以是例如上述AE所支持的动画类型。

[0083] 其中,动画类型包括但不限于旋转、位移等。以旋转类型的动画类来说,该动画类

可以描述旋转动画涉及的动画参数、渲染信息等。这样,不论目标动画是AE制作的哪些动画类型的动画所组合成的目标动画,本公开实施例的目标APP都可以实现该动画功能,从而使安装在第一终端的目标APP可实现的动画与安装在第二终端的动画应用程序可实现的动画是相同的。

[0084] 其中,预先存储的动画类中可以存储该动画类的各个动画参数的默认值。例如旋转动画的动画类中存储有默认向左旋转30度,旋转时长为2秒的动画参数的默认值。

[0085] 那么本步骤中,本公开实施例的目标APP中的动画系统,可以根据目标动画的所述参数信息和预先存储的多个动画类,来获取所述目标动画的模型数据和渲染信息。

[0086] 其中,所谓模型数据,即动画的应用对象。以旋转类型的动画为例,其旋转的动作需要有应用对象,而其应用对象即为模型数据,例如模型1,生成的动画就是该模型1进行旋转。

[0087] 所谓渲染数据,可以理解为渲染指令,该渲染指令输入给硬件(例如GPU(图形处理器,Graphics Processing Unit))后,可以使GPU在第一终端的屏幕上按照该渲染指令对模型数据进行渲染。

[0088] 在一种可能的实施方式中,在执行步骤102时,参照图2,可以通过S201和S202来实现:

[0089] S201,根据所述参数信息和预先存储的与不同动画类型匹配的多个动画类,获取与所述目标动画匹配的多个动画对象;

[0090] 其中,目标APP中预先存储的多个动画类定义了每个动画类型的动画参数、动画参数的默认值等,但是,目标动画的动画参数的数值并不一定为所述默认值,且目标动画可能包括多个动画类型的动画的组合。因此,本步骤可以根据预先存储的多个动画类,以及所述参数信息,来创建多个动画对象,其中,该多个动画对象分别对应的动画类型都是该目标动画的参数信息中包括的动画类型。且创建的多个动画对象已经按照所述参数信息配置有自定义数值的动画参数。

[0091] S202,根据所述参数信息,获取所述多个动画对象中每个动画对象的模型数据、第一渲染信息,以及用于描述所述多个动画对象中不同动画对象之间动画关系的第二渲染信息。

[0092] 其中,目标动画的参数信息中可以携带指示该目标动画涉及的动画模型的数据,因此,可以利用该参数信息获取到所述多个动画对象中每个动画对象的模型数据;此外,还可以根据目标动画的参数信息,来获取目标动画对应的每个动画对象的第一渲染信息,以及用于描述所述多个动画对象中不同动画对象之间动画关系的第二渲染信息。

[0093] 这样,本公开实施例通过在获取目标动画的模型数据和渲染信息时,利用目标动画的参数信息和预先存储的多个动画类,从而可以创建得到与目标动画匹配的多个动画对象,并利用该参数信息可以获取到每个动画对象的模型数据、第一渲染信息,用于描述所述多个动画对象中不同动画对象之间动画关系的第二渲染信息,简单快捷的绘制目标动画所需要的数据获取到,提升了目标动画在第一终端上的生成效率。

[0094] 在一种可能的实施方式中,所述目标动画由一个或多个原子动画构成,所述参数信息包括与所述目标动画对应的每个原子动画的动画类型、每个原子动画的动画参数、每个原子动画的模型标识以及不同原子动画之间的关联参数。

[0095] 其中,由于任意一个设计好的目标动画都是由多个子动画构成的,其中,每个子动画不可以再进行拆分,因此,这里可以将构成目标动画的且不可以再拆分的动画称作原子动画。一个目标动画可以包括一个或多个原子动画,原子动画的类型可以包括但不限于旋转、位移、缩放等。

[0096] 目标动画的配置文件中可以记录该目标动画由哪些类型的原子动画构成,构成该目标动画的每个原子动画的动画参数以及不同原子动画之间的关联参数,此外,目标动画的配置文件还可以记录每个原子动画的应用对象,即模型标识。

[0097] 因此,本公开实施例的方法通过对目标动画的参数信息进行解析,可以获取到上述列举的各个参数信息。

[0098] 对于动画参数,以旋转类型的原子动画举例来说,其动画参数可以包括但不限于:旋转角速度、旋转线速度、绕哪个轴旋转、旋转持续时长、旋转方向等;以位移类型的原子动画举例,其动画参数可以包括但不限于起始速度、加速度、位移轨迹等。

[0099] 对于不同原子动画之间的关联参数,以旋转动画和位移动画为例来说明,这两个动画之间的显示顺序是怎样的、二者如何组合(例如旋转过程中进行位移、旋转后再位移等组合方式)、不同动画之间的间隔时长等。其中,该不同原子动画之间的关联参数,体现在上文所述的第二渲染信息中。

[0100] 在本公开实施例中,参照图3,那么在执行S201时,可以通过S301来实现:

[0101] S301,根据预先存储的与不同动画类型匹配的多个动画类,按照所述参数信息中所述每个原子动画的所述动画类型和所述动画参数,创建与所述目标动画匹配的多个动画对象,其中,每个动画对象分别配置有对应的所述动画参数。

[0102] 具体而言,预先存储的动画类是动画应用程序支持的例如所有动画类型,但是,该目标动画可以只由部分原子动画构成,因此,可以按照目标动画的参数信息中的每个原子动画的动画类型(例如包括旋转和位移),来对预先存储的多个动画类中,只对旋转动画的动画类和位移动画的动画类分别创建两个动画对象,并根据参数信息中每个原子动画的动画参数,来对创建的动画对象分别配置对应的动画参数。其中,预先存储的动画类的动画类型对应于一种原子动画的动画类型,不同动画类的动画类型对应于不同原子动画的动画类型。

[0103] 举例来说,目标动画的动画参数包括三个原子动画,分别为位移动画1、位移动画2和旋转动画3,位移动画1具有动画参数1、位移动画2具有动画参数2、旋转动画3具有动画参数3。那么本步骤需要从预先存储的多个动画类中,对位移动画类创建两个位移动画对象,分别为位移动画对象1和位移动画对象2,且采用动画参数1对位移动画对象1的动画参数进行配置,采用动画参数2对位移动画对象2的动画参数进行配置;以及对旋转动画类创建一个旋转动画对象3,并采用动画参数3为旋转动画3配置动画参数。

[0104] 这样,创建的每个动画对象所配置的动画参数以及动画参数的数值,都是与目标动画匹配的,而非预先存储的动画类中的各个动画参数的默认数值。

[0105] 那么在执行S202时,参照图3,可以通过S302~S305来实现:

[0106] S302,根据所述参数信息中与所述目标动画对应的每个原子动画的模型标识,获取所述多个动画对象中每个动画对象的模型数据;

[0107] 其中,目标动画的参数信息可以记录该目标动画所包括的每个原子动画的模型标

识,那么本公开实施例的方法就可以利用该模型标识,来获取与该模型标识匹配的模型数据,将该模型数据作为该模型标识对应的原子动画的模型数据。而原子动画与上述实施例所述的动画对象又是一一对应的,因此,这里可以获取到目标动画对应的每个动画对象的模型数据。例如位移动画对象1的模型数据为模型1,位移动画对象2的模型数据为模型2,旋转动画对象3的模型数据为模型1。

[0108] 需要说明的是,不同动画对象对应的模型数据可以相同或不同,本公开对此不做限制。

[0109] 在一种可能的实施方式中,在执行S302时,可以首先,获取模型文件;然后,根据所述参数信息中与所述目标动画对应的每个原子动画的模型标识,从所述模型文件中获取所述多个动画对象中每个动画对象的模型数据。

[0110] 具体而言,在获取模型文件时,可以直接获取从动画应用程序处导出的模型文件,该模型文件为该动画应用程序制作的目标动画所对应的模型文件;也可以利用上述配置文件来获取该模型文件,例如配置文件中可以记录该模型文件的路径信息,那么通过读取该路径信息,则可以从相应路径处读取到该模型文件。其中,模型文件中的模型数据的类型可以包括二维模型数据、三维模型数据等,且模型文件中存储着模型标识与模型数据的对应关系。

[0111] 由于目标动画的参数信息中包括目标动画对应的每个原子动画的模型标识,而在模型文件中存储着模型标识与模型数据的对应关系,因此,可以从模型文件中,读取到所述每个原子动画的模型标识对应的模型数据,而每个原子动画都被创建了对应的动画对象,因此这里可以获取到目标动画对应的每个动画对象的模型数据。

[0112] 这样,本公开实施例通过将动画应用程序中关于目标动画的模型文件也导入到本公开实施例的目标APP中,从而可以借助于目标动画的参数信息中的,与所述目标动画对应的每个原子动画的模型标识,来从模型文件中获取到每个原子动画对应的模型数据,而本公开实施例的方法预先已经对每个原子动画创建了对应的动画对象,从而获取到每个动画对象的模型数据,使得本公开实施例的目标APP显示的目标动画与该目标动画在计算机端上显示的效果一致。

[0113] S303,根据所述每个动画对象配置的动画参数,生成每个动画对象的第一渲染信息;

[0114] 以动画对象为旋转动画对象3为例来说明,该旋转动画对象的动画参数包括:从45度旋转至90度,旋转时长为2s,旋转方向为顺时针,绕Y轴旋转,旋转次数为60次。

[0115] 那么本公开实施例的动画系统就可以对该旋转动画对象的上述动画参数进行分析计算,从而确定出该旋转动画对象3的渲染指令。例如某个渲染指令为在屏幕左上角,距离横轴10厘米,距离纵轴2厘米、纵轴画一个点的信息。

[0116] 由于渲染指令是交由GPU来执行的指令,因此,渲染指令只描述在第一终端的屏幕上如何绘制图形的信息。而本公开实施例的动画系统则是根据动画对象的动画参数来计算出该绘制信息,即所述渲染指令。

[0117] S304,按照所述目标动画对应的不同原子动画之间的关联参数,创建所述多个动画对象中不同动画对象之间的关联信息;

[0118] 其中,由于目标动画的参数信息可以包括该目标动画对应的不同原子动画之间的

关联参数,而上述步骤已经对每个原子动画创建了对应的动画对象,因此,这里还需要借助于该关联参数,来建立该目标动画对应的多个动画对象中不同动画对象之间的关联信息。

[0119] 关联参数和关联信息的实质内容可以相同,但是,它们的数据结构不同,因为,关联信息属于动画对象的属性,需要符合动画类中对该属性的定义要求。但是关联参数只是目标动画的参数信息。

[0120] S305,根据所述关联信息生成用于描述所述多个动画对象中不同动画对象之间动画关系的第二渲染信息。

[0121] 其中,S303步骤中对目标动画对应的每个动画对象生成了第一渲染信息,而一个目标动画是由多个原子动画构成的,而多个原子动画之间也是存在关联的,因此,本公开实施例的动画系统可以对目标动画对应的多个动画对象中不同动画对象之间的关联关系进行处理计算,从而生成目标动画中不同动画对象之间的渲染指令。

[0122] 由于渲染指令是交由GPU来执行的指令,因此,渲染指令只描述在第一终端的屏幕上如何绘制图形的信息。

[0123] 这样,本公开实施例通过获取与所述目标动画对应的每个原子动画的动画类型、动画参数、模型标识以及不同原子动画之间的关联参数等参数信息,从而可以借助于动画类型和动画参数,来利用预先存储的多个动画类,来创建与所述目标动画匹配的多个动画对象,其中,即便动画类型相同,但是动画参数不同,也会创建两个不同的动画对象;并利用参数信息中的模型标识,来获取每个动画对象对应的模型数据,以及根据每个对象的动画参数来生成每个动画对象的第一渲染信息,根据不同动画对象之间的关联关系,来生成用于描述所述多个动画对象中不同动画对象之间动画关系的第二渲染信息,生成了目标动画中各个原子动画的渲染信息,以及不同原子动画之间的渲染信息,以及每个原子动画应用的对象,即模型数据。避免了开发多套代码。

[0124] 步骤103,按照预设编程接口与不同开发平台的编程接口之间的预设映射关系,将所述模型数据和所述渲染信息的数据类型从所述预设编程接口支持的第一数据类型,转换为所述第一终端的目标编程接口支持的第二数据类型;

[0125] 具体而言,本公开实施例的目标APP采用预设编程接口(例如OpenGL(Open Graphics Library,开放图形库))的语言编写,其中,该预设编程接口是一种图形API,在开发目标APP时,可以采用任意一种图形API来开发。那么为了能够使编写的目标APP可以应用到各个开发平台上使用,无需更换代码,本公开实施例的目标APP包括中间层,该中间层存储有该预设编程接口与不同开发平台的编程接口(例如Vulkan(Vulkan是一个跨平台的2D和3D绘图API)、DirectX(Direct eXtension,简称DX,是由微软公司创建的多媒体编程接口)、Metal(是一种低层次的渲染应用程序编程接口)等)之间的预设映射关系。

[0126] 其中,在该预设映射关系中,预设编程接口与不同开发平台的编程接口分别具有一对一的映射关系,对于任意一个一对一的映射关系,映射的内容可以包括但不限于以下至少之一:内存信息(对应于下述举例中的字节长度)、坐标轴信息、裁剪空间信息(即深度信息的范围)。通过以上内容的映射,可以达到数据类型转换的目的。

[0127] 例如该目标APP所使用的预设编程接口为OpenGL,而该目标APP安装至的第一终端支持的目标编程接口为Vulkan,那么本公开实施例的中间层就可以将按照该预设映射关系中OpenGL与Vulkan之间的映射关系,来将所述模型数据的数据类型、所述渲染信息的数据

类型从OpenGL要求的数据类型,均转换为Vulkan支持的数据类型,从而使得第一终端的驱动、GPU可以识别转换数据类型后的所述模型数据和所述渲染信息,从而完成目标动画的显示。

[0128] 换句话说,不同编程接口所支持的数据类型不同,因此,需要将模型数据、渲染信息在预设编程接口所对应的数据类型,转换为目标APP目前所安装至的第一终端的编程接口所支持的数据类型。例如编程接口1的渲染函数1支持A类型的渲染信息,编程接口2的渲染函数2支持B类型的渲染信息,所以需要将渲染信息的数据类型从A类型转换为B类型,例如将渲染信息的字节长度从A类型支持的1k转换为B类型支持的0.5k。

[0129] 具体的,在进行数据类型的转换时,为了实现数据类型的转换的目的,可以对渲染信息进行内存空间的转换、坐标值信息的转换、裁剪空间的转换,从而到达将渲染信息从A类型,转换为B类型的数据的目的,使得转换后的属于B类型的渲染信息,可以供编程接口2的渲染函数2所调用。

[0130] 步骤104,根据转换后的所述渲染信息,对转换后的所述模型数据进行渲染,显示目标动画。

[0131] 其中,本公开实施例的中间层将转换后的所述渲染数据以及转换后的模型数据均下发给硬件驱动层,那么本公开实施例的GPU就可以根据转换后的所述渲染信息,对转换后的所述模型数据进行渲染,从而在第一终端的屏幕上显示目标动画。

[0132] 在一种可能的实施方式中,可以根据转换后的第一渲染信息,对每个动画对象对应的模型数据进行渲染,以及根据转换后的第二渲染数据,来对不同动画对象对应的不同模型数据之间的动画关系进行渲染,实现目标动画的实时显示。

[0133] 这样,本公开实施例通过获取待显示的目标动画的参数信息,并结合预先存储的与不同动画类型匹配的多个动画类,来获取目标动画的模型数据和渲染数据,然后,按照预设编程接口与不同开发平台的编程接口的预设映射关系,来将所述模型数据和渲染数据的数据类型均从预设编程接口支持的第一数据类型,转换为第一终端的目标编程接口所支持的第二数据类型,从而利用转换后的数据完成目标动画的显示。本公开实施例的方法可以分别应用于编程接口不同于该预设编程接口的各个第一终端,而无需修改动画代码,实现了对动画代码的复用,并降低了动画代码的维护难度以及开发成本,无需针对不同开发平台针对同一个动画分别进行多套代码实现。

[0134] 图4是根据一示例性实施例示出的一种动画显示装置的结构框图,应用于具有目标编程接口的第一终端。参照图4,该装置包括:

[0135] 第一获取模块41,被配置为获取待显示的目标动画的参数信息,其中,所述目标动画为安装在第二终端上的动画应用程序所绘制的动画;

[0136] 第二获取模块42,被配置为根据所述参数信息和预先存储的与不同动画类型匹配的多个动画类,获取所述目标动画的模型数据和渲染信息;

[0137] 映射模块43,被配置为按照预设编程接口与不同开发平台的编程接口之间的预设映射关系,将所述模型数据和所述渲染信息的数据类型从所述预设编程接口支持的第一数据类型,转换为所述第一终端的目标编程接口支持的第二数据类型;

[0138] 渲染模块44,被配置为根据转换后的所述渲染信息,对转换后的所述模型数据进行渲染,显示目标动画。

[0139] 在一种可能的实施方式中,所述第一获取模块41包括:

[0140] 第一获取子模块,被配置为获取目标动画的配置文件,其中,所述配置文件为安装在所述第二终端上的动画应用程序的与所述目标动画对应的配置文件;

[0141] 第二获取子模块,被配置为根据所述配置文件,获取所述目标动画的参数信息。

[0142] 在一种可能的实施方式中,所述第二获取模块42包括:

[0143] 第三获取子模块,被配置为根据所述参数信息和预先存储的与不同动画类型匹配的多个动画类,获取与所述目标动画匹配的多个动画对象;

[0144] 第四获取子模块,被配置为根据所述参数信息,获取所述多个动画对象中每个动画对象的模型数据、第一渲染信息,以及用于描述所述多个动画对象中不同动画对象之间动画关系的第二渲染信息。

[0145] 在一种可能的实施方式中,所述目标动画由一个或多个原子动画构成,所述参数信息包括与所述目标动画对应的每个原子动画的动画类型、每个原子动画的动画参数、每个原子动画的模型标识以及不同原子动画之间的关联参数;

[0146] 所述第三获取子模块包括:

[0147] 第一创建单元,被配置为根据预先存储的与不同动画类型匹配的多个动画类,按照所述参数信息中所述每个原子动画的所述动画类型和所述动画参数,创建与所述目标动画匹配的多个动画对象,其中,每个动画对象分别配置有对应的所述动画参数;

[0148] 所述第四获取子模块包括:

[0149] 第一获取单元,被配置为根据所述参数信息中与所述目标动画对应的每个原子动画的模型标识,获取所述多个动画对象中每个动画对象的模型数据;

[0150] 第一生成单元,被配置为根据所述每个动画对象配置的动画参数,生成每个动画对象的第一渲染信息;

[0151] 第二创建单元,被配置为按照所述目标动画对应的不同原子动画之间的关联参数,创建所述多个动画对象中不同动画对象之间的关联信息;

[0152] 第二生成单元,被配置为根据所述关联信息生成用于描述所述多个动画对象中不同动画对象之间动画关系的第二渲染信息。

[0153] 在一种可能的实施方式中,所述第一获取单元包括:

[0154] 第一获取子单元,被配置为获取模型文件;

[0155] 第二获取子单元,被配置为根据所述参数信息中与所述目标动画对应的每个原子动画的模型标识,从所述模型文件中获取所述多个动画对象中每个动画对象的模型数据。

[0156] 在一种可能的实施方式中,所述多个动画类对应的动画类型为所述动画应用程序支持的动画类型。

[0157] 关于上述实施例中的装置,其中各个模块执行操作的具体方式已经在有关该方法的实施例中进行了详细描述,此处将不做详细阐述说明。

[0158] 图5是根据一示例性实施例示出的一种用于动画显示的装置800的框图。例如,装置800可以是移动电话,计算机,数字广播终端,消息收发设备,游戏控制台,平板设备,医疗设备,健身设备,个人数字助理等。

[0159] 参照图5,装置800可以包括以下一个或多个组件:处理组件802,存储器804,电力组件806,多媒体组件808,音频组件810,输入/输出(I/O)的接口812,传感器组件814,以及

通信组件816。

[0160] 处理组件802通常控制装置800的整体操作,诸如与显示,电话呼叫,数据通信,相机操作和记录操作相关联的操作。处理组件802可以包括一个或多个处理器820来执行指令,以完成上述的方法的全部或部分步骤。此外,处理组件802可以包括一个或多个模块,便于处理组件802和其他组件之间的交互。例如,处理组件802可以包括多媒体模块,以方便多媒体组件808和处理组件802之间的交互。

[0161] 存储器804被配置为存储各种类型的数据以支持在设备800的操作。这些数据的示例包括用于在装置800上操作的任何应用程序或方法的指令,联系人数据,电话簿数据,消息,图片,视频等。存储器804可以由任何类型的易失性或非易失性存储设备或者它们的组合实现,如静态随机存取存储器(SRAM),电可擦除可编程只读存储器(EEPROM),可擦除可编程只读存储器(EPROM),可编程只读存储器(PROM),只读存储器(ROM),磁存储器,快闪存储器,磁盘或光盘。

[0162] 电源组件806为装置800的各种组件提供电力。电源组件806可以包括电源管理系统,一个或多个电源,及其他与为装置800生成、管理和分配电力相关联的组件。

[0163] 多媒体组件808包括在所述装置800和用户之间的提供一个输出接口的屏幕。在一些实施例中,屏幕可以包括液晶显示器(LCD)和触摸面板(TP)。如果屏幕包括触摸面板,屏幕可以被实现为触摸屏,以接收来自用户的输入信号。触摸面板包括一个或多个触摸传感器以感测触摸、滑动和触摸面板上的手势。所述触摸传感器可以不仅感测触摸或滑动动作的边界,而且还检测与所述触摸或滑动操作相关的持续时间和压力。在一些实施例中,多媒体组件808包括一个前置摄像头和/或后置摄像头。当设备800处于操作模式,如拍摄模式或视频模式时,前置摄像头和/或后置摄像头可以接收外部的多媒体数据。每个前置摄像头和后置摄像头可以是一个固定的光学透镜系统或具有焦距和光学变焦能力。

[0164] 音频组件810被配置为输出和/或输入音频信号。例如,音频组件810包括一个麦克风(MIC),当装置800处于操作模式,如呼叫模式、记录模式和语音识别模式时,麦克风被配置为接收外部音频信号。所接收的音频信号可以被进一步存储在存储器804或经由通信组件816发送。在一些实施例中,音频组件810还包括一个扬声器,用于输出音频信号。

[0165] I/O接口812为处理组件802和外围接口模块之间提供接口,上述外围接口模块可以是键盘,点击轮,按钮等。这些按钮可包括但不限于:主页按钮、音量按钮、启动按钮和锁定按钮。

[0166] 传感器组件814包括一个或多个传感器,用于为装置800提供各个方面的状态评估。例如,传感器组件814可以检测到设备800的打开/关闭状态,组件的相对定位,例如所述组件为装置800的显示器和小键盘,传感器组件814还可以检测装置800或装置800一个组件的位置改变,用户与装置800接触的存在或不存在,装置800方位或加速/减速和装置800的温度变化。传感器组件814可以包括接近传感器,被配置用来在没有任何的物理接触时检测附近物体的存在。传感器组件814还可以包括光传感器,如CMOS或CCD图像传感器,用于在成像应用中使用。在一些实施例中,该传感器组件814还可以包括加速度传感器,陀螺仪传感器,磁传感器,压力传感器或温度传感器。

[0167] 通信组件816被配置为便于装置800和其他设备之间有线或无线方式的通信。装置800可以接入基于通信标准的无线网络,如WiFi,运营商网络(如2G、3G、4G或5G),或它们的

组合。在一个示例性实施例中,通信组件816经由广播信道接收来自外部广播管理系统的广播信号或广播相关信息。在一个示例性实施例中,所述通信组件816还包括近场通信(NFC)模块,以促进短程通信。例如,在NFC模块可基于射频识别(RFID)技术,红外数据协会(IrDA)技术,超宽带(UWB)技术,蓝牙(BT)技术和其他技术来实现。

[0168] 在示例性实施例中,装置800可以被一个或多个应用专用集成电路(ASIC)、数字信号处理器(DSP)、数字信号处理设备(DSPD)、可编程逻辑器件(PLD)、现场可编程门阵列(FPGA)、控制器、微控制器、微处理器或其他电子元件实现,用于执行上述方法。

[0169] 在示例性实施例中,还提供了一种包括指令的非临时性计算机可读存储介质,例如包括指令的存储器804,上述指令可由装置800的处理器820执行以完成上述方法。例如,所述非临时性计算机可读存储介质可以是ROM、随机存取存储器(RAM)、CD-ROM、磁带、软盘和光数据存储设备等。

[0170] 在示例性实施例中,还提供了一种包括指令的应用程序,例如包括指令的存储器804,上述指令可由装置800的处理器820执行以完成上述方法。例如,所述非临时性计算机可读存储介质可以是ROM、随机存取存储器(RAM)、CD-ROM、磁带、软盘和光数据存储设备等。

[0171] 图6是根据一示例性实施例示出的一种用于动画显示的装置1900的框图。例如,装置1900可以被提供为一服务器。参照图6,装置1900包括处理组件1922,其进一步包括一个或多个处理器,以及由存储器1932所代表的存储器资源,用于存储可由处理组件1922的执行的指令,例如应用程序。存储器1932中存储的应用程序可以包括一个或一个以上的每一个对应于一组指令的模块。此外,处理组件1922被配置为执行指令,以执行上述方法。

[0172] 装置1900还可以包括一个电源组件1926被配置为执行装置1900的电源管理,一个有线或无线网络接口1950被配置为将装置1900连接到网络,和一个输入输出(I/O)接口1958。装置1900可以操作基于存储在存储器1932的操作系统,例如Windows Server™,Mac OS X™,Unix™,Linux™,FreeBSD™或类似。

[0173] 需要说明的是,本公开的执行主体可以是移动电话,计算机,数字广播终端,消息收发设备,游戏控制台,平板设备,医疗设备,健身设备,个人数字助理等;也可以是服务器。当电子设备如移动电话,计算机,数字广播终端,消息收发设备,游戏控制台,平板设备,医疗设备,健身设备,个人数字助理等时,如图5所示。当电子设备是服务器时,如图6所示。

[0174] 本领域技术人员在考虑说明书及实践这里公开的发明后,将容易想到本公开的其他实施方案。本公开旨在涵盖本公开的任何变型、用途或者适应性变化,这些变型、用途或者适应性变化遵循本公开的一般性原理并包括本公开未公开的本技术领域中的公知常识或惯用技术手段。说明书和实施例仅被视为示例性的,本公开的真正范围和精神由下面的权利要求指出。

[0175] 应当理解的是,本公开并不局限于上面已经描述并在附图中示出的精确结构,并且可以在不脱离其范围进行各种修改和改变。本公开的范围仅由所附的权利要求来限制。

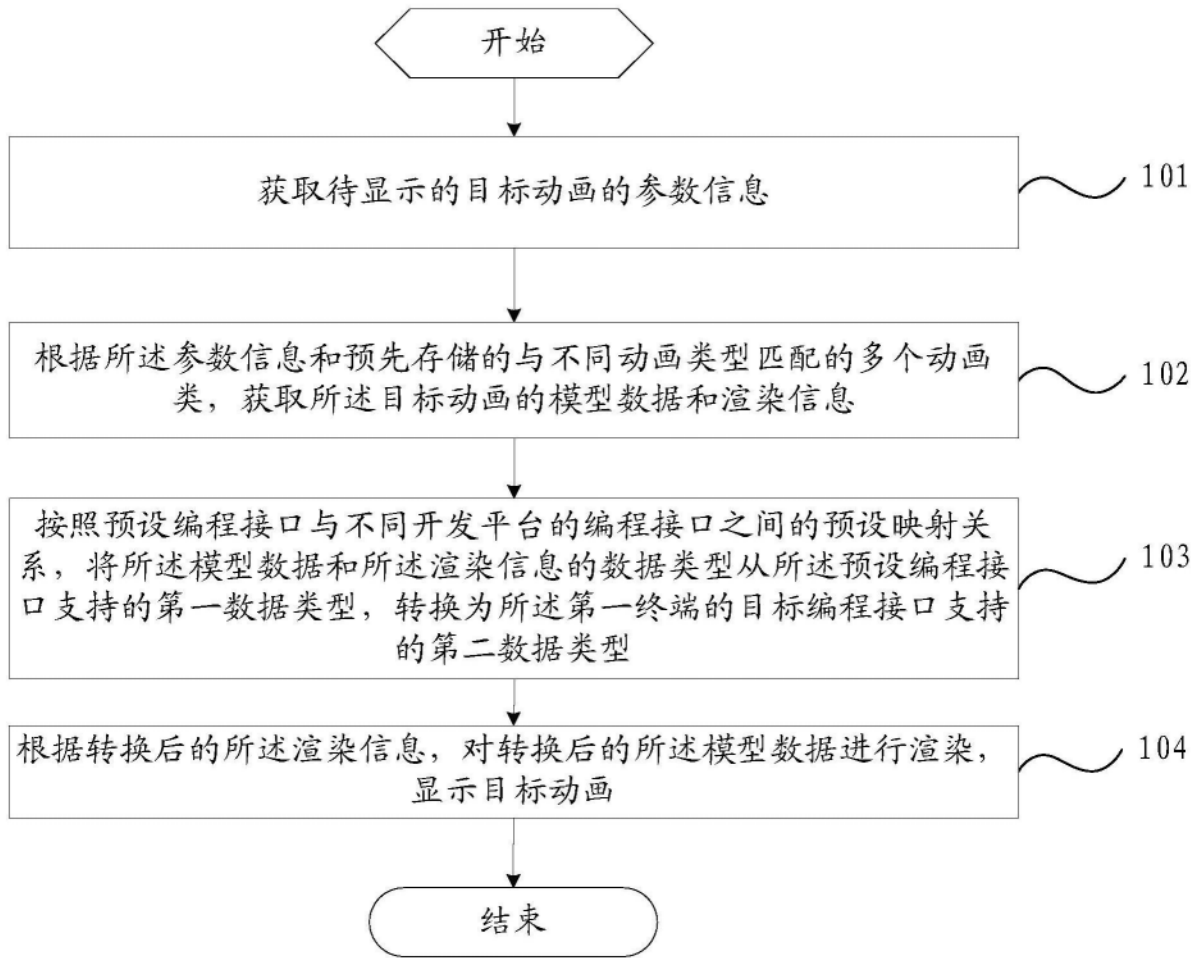


图1

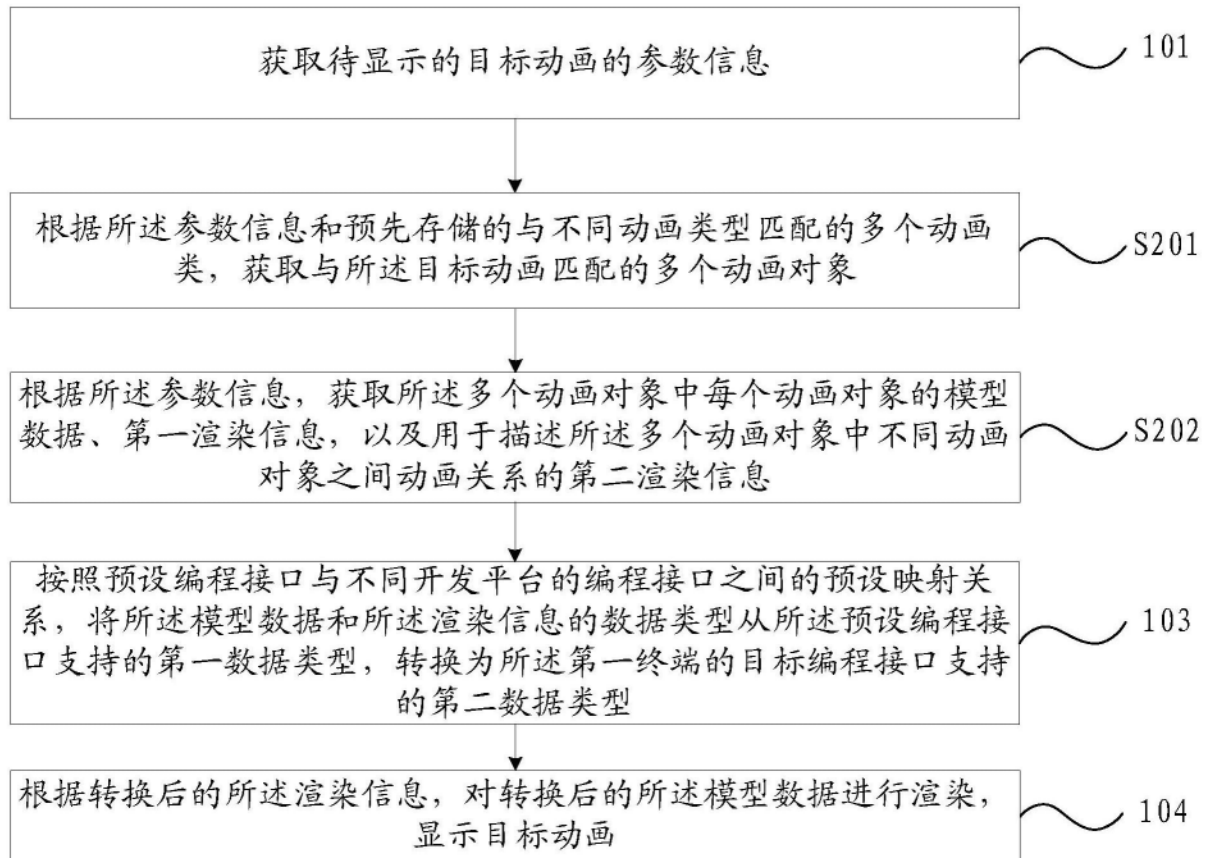


图2

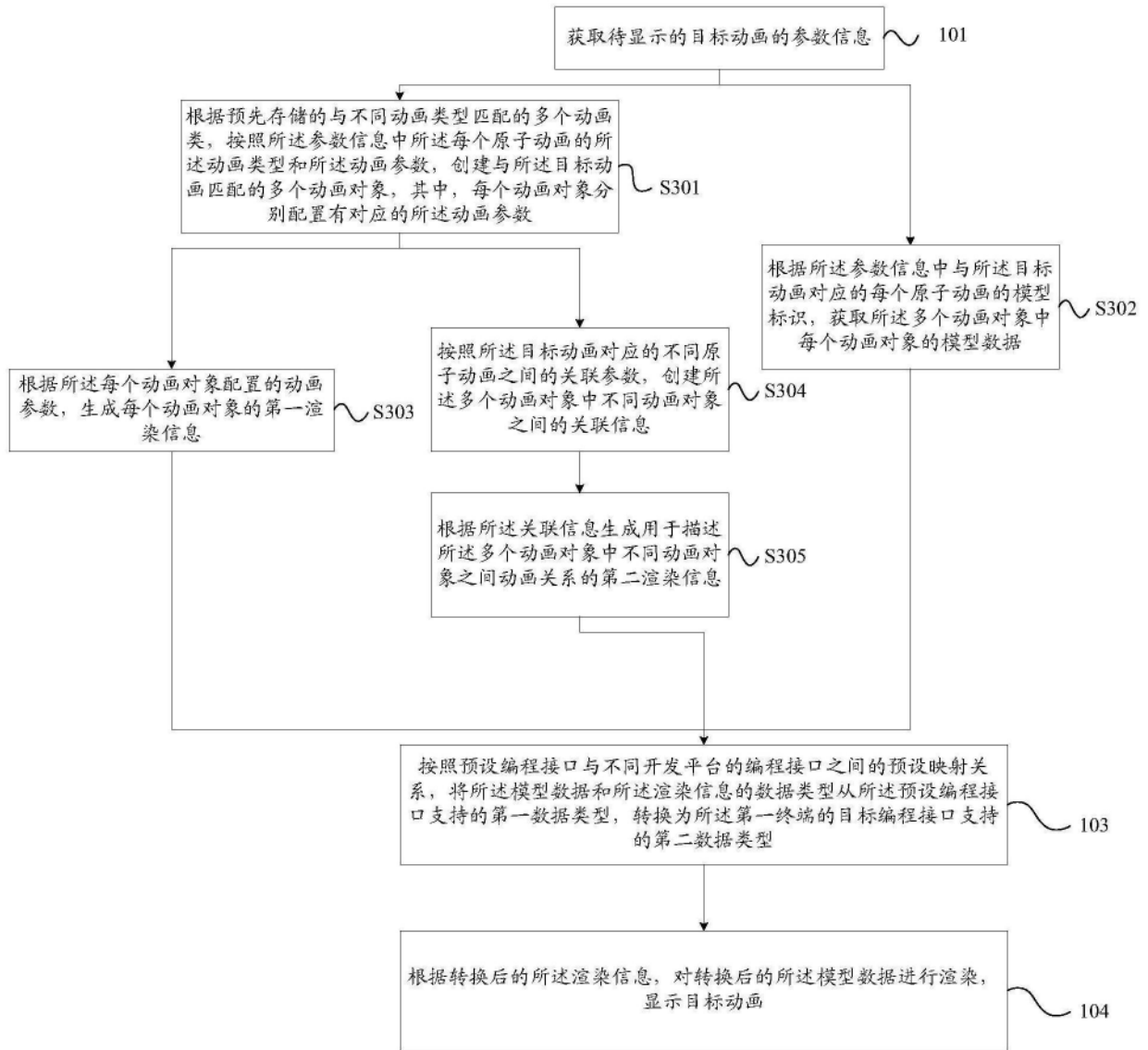


图3

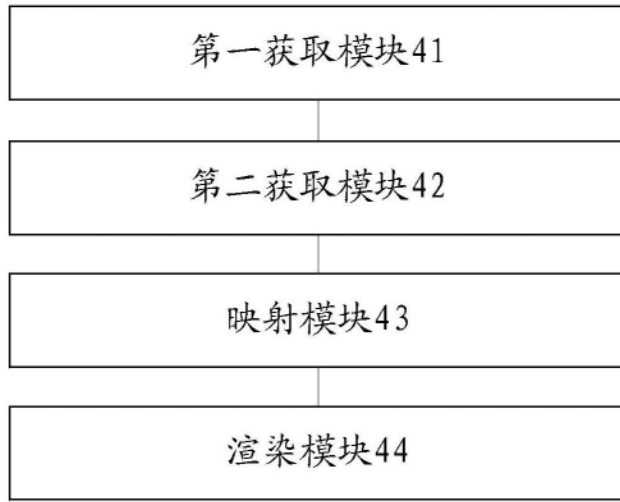


图4

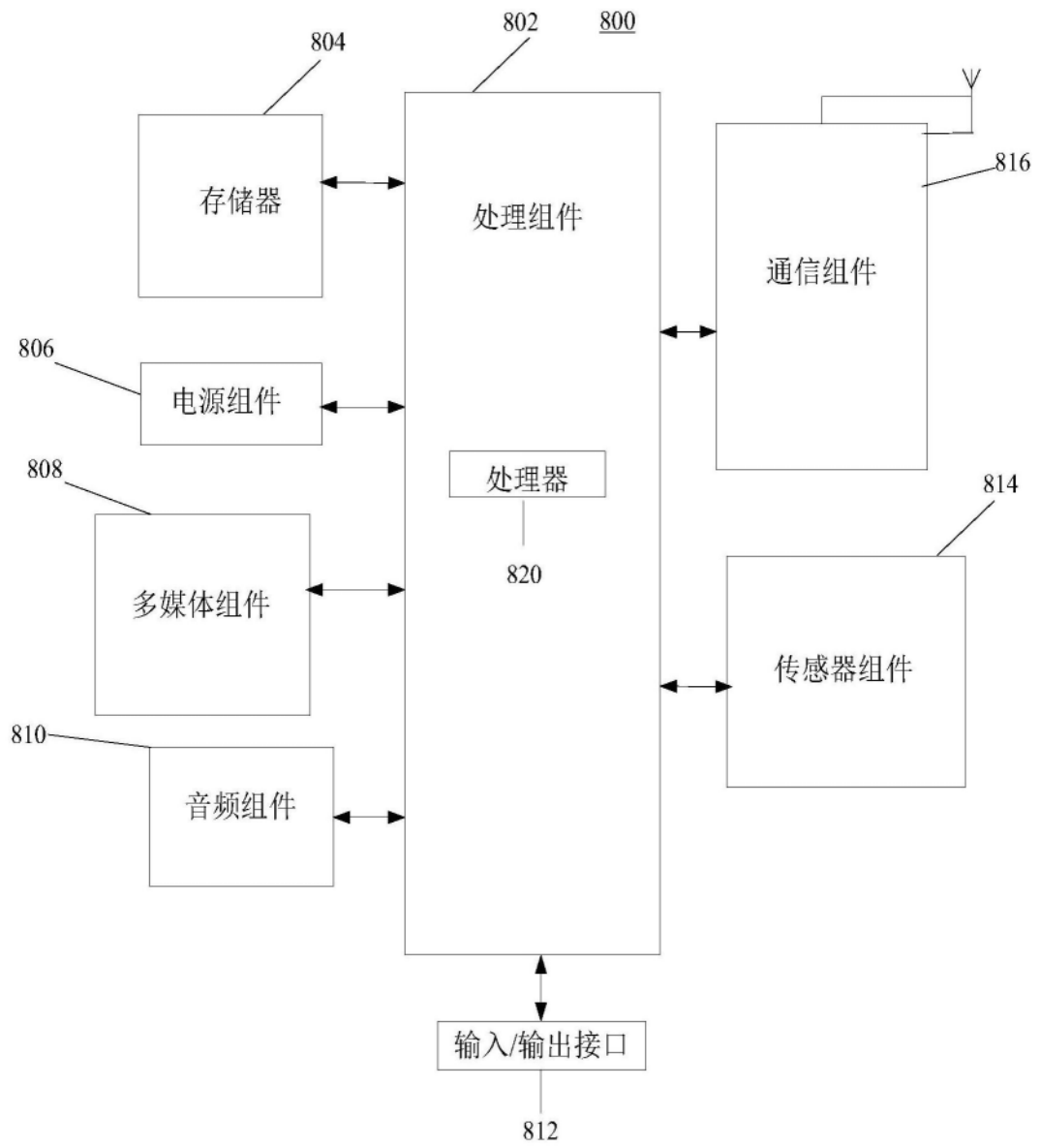


图5

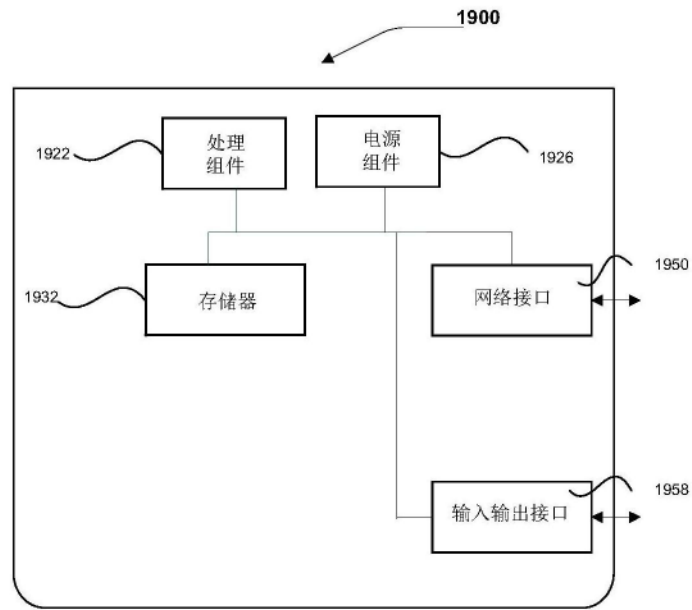


图6