



(12) 发明专利

(10) 授权公告号 CN 109582463 B

(45) 授权公告日 2021. 04. 06

(21) 申请号 201811457185.3

(22) 申请日 2018.11.30

(65) 同一申请的已公布的文献号
申请公布号 CN 109582463 A

(43) 申请公布日 2019.04.05

(73) 专利权人 OPPO广东移动通信有限公司
地址 523860 广东省东莞市长安镇乌沙海
滨路18号

(72) 发明人 陈岩

(74) 专利代理机构 北京三高永信知识产权代理
有限责任公司 11138
代理人 牟慧仙

(51) Int.Cl.
G06F 9/50 (2006.01)
G06K 9/62 (2006.01)

(56) 对比文件

US 2018108105 A1,2018.04.19

EP 2921973 A1,2015.09.23

CN 108769821 A,2018.11.06

审查员 沈晴

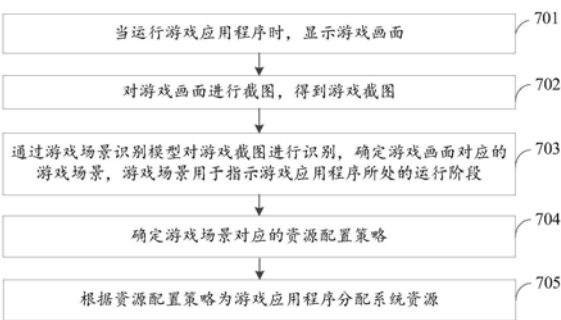
权利要求书2页 说明书12页 附图8页

(54) 发明名称

资源配置方法、装置、终端及存储介质

(57) 摘要

本申请实施例公开了一种资源配置方法、装置、终端及存储介质,属于应用优化领域。所述方法包括:当运行游戏应用程序时,显示游戏画面;对游戏画面进行截图,得到游戏截图;通过游戏场景识别模型对游戏截图进行识别,确定游戏画面对应的游戏场景,游戏场景用于指示游戏应用程序所处的运行阶段;确定游戏场景对应的资源配置策略;根据资源配置策略为游戏应用程序分配系统资源。本申请实施例提供的资源配置方法中,游戏应用程序无需接入终端操作系统,终端操作系统也能够智能分析出当前所处的游戏场景,从而免去了在游戏应用程序与操作系统之间建立数据通道的过程,并有助于提高操作系统的安全性。



1. 一种资源配置方法,其特征在于,所述方法应用于终端中,所述终端中运行有操作系统和游戏应用程序,所述方法包括:

当运行所述游戏应用程序时,显示游戏画面;

所述操作系统获取所述游戏应用程序的应用标识;

所述操作系统检测所述应用标识是否属于预设游戏列表,所述预设游戏列表中包含目标游戏应用程序的标识,所述目标游戏应用程序在运行过程中通过与操作系统之间的数据通道发送场景信息,所述操作系统用于根据所述场景信息指示的游戏场景确定资源配置策略;

若所述应用标识不属于所述预设游戏列表,所述操作系统对所述游戏画面进行截图,得到游戏截图;

所述操作系统通过游戏场景识别模型对所述游戏截图进行识别,确定所述游戏画面对应的游戏场景,所述游戏场景用于指示所述游戏应用程序所处的运行阶段,所述游戏场景识别模型是所述操作系统根据样本游戏截图和所述目标游戏应用程序对应场景信息所指示的游戏场景训练得到,所述样本游戏截图是所述操作系统在接收到所述目标游戏应用程序发送的所述场景信息时截图得到;

所述操作系统确定所述游戏场景对应的所述资源配置策略;

所述操作系统根据所述资源配置策略为所述游戏应用程序分配系统资源。

2. 根据权利要求1所述的方法,其特征在于,所述通过游戏场景识别模型对所述游戏截图进行识别,确定所述游戏画面对应的游戏场景,包括:

获取所述游戏应用程序对应的游戏类型;

将所述游戏类型和所述游戏截图输入所述游戏场景识别模型,得到所述游戏场景识别模型输出的所述游戏场景;

所述确定所述游戏场景对应的资源配置策略,包括:

根据所述游戏类型以及所述游戏场景,确定所述游戏场景对应的所述资源配置策略;

其中,所述游戏类型包括如下至少一种:第一人称射击游戏、第三人称射击游戏、多人在线战术竞技游戏、卡牌游戏、竞速游戏、音乐游戏、模拟经营游戏、冒险游戏、格斗游戏、角色扮演游戏和沙盒游戏。

3. 根据权利要求2所述的方法,其特征在于,所述确定所述游戏场景对应的资源配置策略之前,所述方法还包括:

所述操作系统获取当前剩余电量以及所述游戏应用程序的应用程序大小,所述应用程序大小用于指示所述游戏应用程序所占的存储空间;

所述根据所述游戏类型以及所述游戏场景,确定所述游戏场景对应的所述资源配置策略,包括:

将所述当前剩余电量、所述应用程序大小、所述游戏类型以及所述游戏场景输入资源配置模型,得到所述资源配置模型输出的所述资源配置策略。

4. 根据权利要求1所述的方法,其特征在于,所述方法还包括:

当所述操作系统接收到所述目标游戏应用程序发送的所述场景信息时,对所述目标游戏应用程序的当前游戏画面进行截图,得到所述样本游戏截图;

所述操作系统根据所述场景信息指示的所述游戏场景以及所述样本游戏截图,通过反

向传播算法对所述游戏场景识别模型进行训练。

5. 根据权利要求1所述的方法, 其特征在于, 所述对所述游戏画面进行截图, 得到游戏截图, 包括:

在所述游戏画面上层显示加速控件;

当接收到对所述加速控件的触发信号时, 对所述游戏画面进行截图, 得到所述游戏截图;

所述根据所述资源配置策略为所述游戏应用程序分配系统资源之后, 所述方法还包括:

所述操作系统每隔预定时间间隔识别所述游戏画面对应的所述游戏场景;

若所述游戏场景发生变化, 则所述操作系统重新确定所述游戏场景对应的所述资源配置策略。

6. 根据权利要求1至3任一所述的方法, 其特征在于,

所述系统资源的资源类型包括中央处理器资源、图形处理器资源、磁盘资源、内存资源和网络资源中的至少一种。

7. 一种资源配置装置, 其特征在于, 所述装置应用于终端中, 所述终端中运行有操作系统和游戏应用程序, 所述装置包括:

显示模块, 用于当运行所述游戏应用程序时, 显示游戏画面;

第二获取模块, 用于通过所述操作系统获取所述游戏应用程序的应用标识;

检测模块, 用于通过所述操作系统检测所述应用标识是否属于预设游戏列表, 所述预设游戏列表中包含目标游戏应用程序的标识, 所述目标游戏应用程序在运行过程中通过与操作系统之间的数据通道发送场景信息, 所述操作系统用于根据所述场景信息指示的游戏场景确定资源配置策略;

截图模块, 用于若所述应用标识不属于所述预设游戏列表, 通过所述操作系统对所述游戏画面进行截图, 得到游戏截图;

场景识别模块, 用于通过所述操作系统利用游戏场景识别模型对所述游戏截图进行识别, 确定所述游戏画面对应的游戏场景, 所述游戏场景用于指示所述游戏应用程序所处的运行阶段, 所述游戏场景识别模型是所述操作系统根据样本游戏截图和所述目标游戏应用程序对应场景信息所指示的游戏场景训练得到, 所述样本游戏截图是所述操作系统在接收到所述目标游戏应用程序发送的所述场景信息时截图得到;

策略确定模块, 用于通过所述操作系统确定所述游戏场景对应的所述资源配置策略;

资源分配模块, 用于通过所述操作系统根据所述资源配置策略为所述游戏应用程序分配系统资源。

8. 一种终端, 其特征在于, 所述终端包括处理器和存储器; 所述存储器存储有至少一条指令, 所述至少一条指令用于被所述处理器执行以实现如权利要求1至6任一所述的资源配置方法。

9. 一种计算机可读存储介质, 其特征在于, 所述存储介质存储有至少一条指令, 所述至少一条指令用于被处理器执行以实现如权利要求1至6任一所述的资源配置方法。

资源配置方法、装置、终端及存储介质

技术领域

[0001] 本申请实施例涉及应用优化技术领域,特别涉及一种资源配置方法、装置、终端及存储介质。

背景技术

[0002] 随着移动终端技术的不断发展,移动终端中应用程序的种类也越来越多。比如,移动终端中同时安装有游戏类应用程序、社交类应用程序、视频播放类应用程序、即时通信类应用程序和购物类应用程序。

[0003] 为了提高应用程序的使用体验,移动终端厂商都在不断提升自身的硬件配置,以提高应用程序的运行速度及质量。比如,移动终端通过配置高性能的中央处理器(CPU, Central Processing Unit)和图形处理器(Graphics Processing Unit,GPU),来提高游戏类应用程序的运行速度和画面质量。

发明内容

[0004] 本申请实施例提供了一种资源配置方法、装置、终端及存储介质。所述技术方案如下:

[0005] 一方面,提供了一种资源配置方法,所述方法包括:

[0006] 当运行游戏应用程序时,显示游戏画面;

[0007] 对所述游戏画面进行截图,得到游戏截图;

[0008] 通过游戏场景识别模型对所述游戏截图进行识别,确定所述游戏画面对应的游戏场景,所述游戏场景用于指示所述游戏应用程序所处的运行阶段;

[0009] 确定所述游戏场景对应的资源配置策略;

[0010] 根据所述资源配置策略为所述游戏应用程序分配系统资源。

[0011] 另一方面,提供了一种资源配置装置,所述装置包括:

[0012] 显示模块,用于当运行游戏应用程序时,显示游戏画面;

[0013] 截图模块,用于对所述游戏画面进行截图,得到游戏截图;

[0014] 场景识别模块,用于通过游戏场景识别模型对所述游戏截图进行识别,确定所述游戏画面对应的游戏场景,所述游戏场景用于指示所述游戏应用程序所处的运行阶段;

[0015] 策略确定模块,用于确定所述游戏场景对应的资源配置策略;

[0016] 资源分配模块,用于根据所述资源配置策略为所述游戏应用程序分配系统资源。

[0017] 另一方面,提供了一种终端,所述终端包括处理器和存储器;所述存储器存储有至少一条指令,所述至少一条指令用于被所述处理器执行以实现如上述方面所述的资源配置方法。

[0018] 另一方面,提供了一种计算机可读存储介质,所述存储介质存储有至少一条指令,所述至少一条指令用于被处理器执行以实现如上述方面所述的资源配置方法。

[0019] 游戏应用程序运行过程中,通过对游戏画面进行截图,并通过预先训练得到的游

戏场景识别模型对游戏截图进行识别,确定游戏画面对应的游戏场景,从而根据该游戏场景对应的资源配置策略为游戏应用程序分配系统资源,进而优化游戏应用程序在不同游戏场景下的性能以及功耗;本申请实施例提供的资源配置方法中,游戏应用程序无需接入终端操作系统,终端操作系统也能够智能分析出当前所处的游戏场景,从而免去了在游戏应用程序与操作系统之间建立数据通道的过程,并有助于提高操作系统的安全性。

附图说明

- [0020] 图1是本申请一个示例性实施例提供的终端的结构示意图;
- [0021] 图2是终端中应用程序与操作系统通信过程的实施示意图;
- [0022] 图3是本申请一个示例性实施例提供的终端的结构示意图;
- [0023] 图4和图5是图3所示终端中应用程序与操作系统通信过程的实施示意图;
- [0024] 图6是本申请另一个示例性实施例提供的终端的结构示意图;
- [0025] 图7示出了本申请一个示例性实施例示出的资源配置方法的流程图;
- [0026] 图8示出了本申请另一个示例性实施例示出的资源配置方法的流程图;
- [0027] 图9示出了本申请另一个示例性实施例示出的资源配置方法的流程图;
- [0028] 图10示出了本申请另一个示例性实施例示出的资源配置方法的流程图;
- [0029] 图11是在游戏画面上层显示加速控件的界面示意图;
- [0030] 图12示出了本申请一个实施例提供的资源配置装置的结构框图。

具体实施方式

[0031] 为使本发明的目的、技术方案和优点更加清楚,下面将结合附图对本发明实施方式作进一步地详细描述。

[0032] 在本文中提及的“多个”是指两个或两个以上。“和/或”,描述关联对象的关联关系,表示可以存在三种关系,例如,A和/或B,可以表示:单独存在A,同时存在A和B,单独存在B这三种情况。字符“/”一般表示前后关联对象是一种“或”的关系。

[0033] 请参考图1,其示出了本申请一个示例性实施例提供的终端100的结构方框图。该终端100可以是智能手机、平板电脑、便携式个人计算机等能够运行有游戏应用程序的电子设备。本申请中的终端100可以包括一个或多个如下部件:处理器110、存储器120和输入输出装置130。

[0034] 处理器110可以包括一个或者多个处理核心。处理器110利用各种接口和线路连接整个终端100内的各个部分,通过运行或执行存储在存储器120内的指令、程序、代码集或指令集,以及调用存储在存储器120内的数据,执行终端100的各种功能和处理数据。可选地,处理器110可以采用数字信号处理(Digital Signal Processing,DSP)、现场可编程门阵列(Field-Programmable Gate Array,FPGA)、可编程逻辑阵列(Programmable Logic Array,PLA)中的至少一种硬件形式来实现。处理器110可集成中央处理器(Central Processing Unit,CPU)、图像处理器(Graphics Processing Unit,GPU)和调制解调器等中的一种或几种的组合。其中,CPU主要处理操作系统、用户界面和应用程序等;GPU用于负责显示内容的渲染和绘制;调制解调器用于处理无线通信。可以理解的是,上述调制解调器也可以不集成到处理器110中,单独通过一块通信芯片进行实现。

[0035] 可选的,处理器110还可以包括AI(Artificial Intelligence,人工智能)处理器,该AI处理器用于处理有关机器学习的计算操作。

[0036] 存储器120可以包括随机存储器(Random Access Memory,RAM),也可以包括只读存储器(Read-Only Memory)。可选地,该存储器120包括非瞬时性计算机可读介质(non-transitory computer-readable storage medium)。存储器120可用于存储指令、程序、代码、代码集或指令集。存储器120可包括存储程序区和存储数据区,其中,存储程序区可存储用于实现操作系统的指令、用于实现至少一个功能的指令(比如触控功能、声音播放功能、图像播放功能等)、用于实现下述各个方法实施例的指令等,该操作系统可以是安卓(Android)系统(包括基于Android系统深度开发的系统)、苹果公司开发的IOS系统(包括基于IOS系统深度开发的系统)或其它系统。存储数据区还可以存储终端100在使用中所创建的数据(比如电话本、音视频数据、聊天记录数据)等。

[0037] 输入输出装置130可以包括触摸显示屏,该触摸显示屏用于接收用户使用手指、触摸笔等任何适合的物体在其上或附近的触摸操作,以及显示各个应用程序的用户界面。触摸显示屏通常设置在终端100的前面板。触摸显示屏可被设计成为全面屏、曲面屏或异型屏。触摸显示屏还可被设计成为全面屏与曲面屏的结合,异型屏与曲面屏的结合,本申请实施例对此不加以限定。

[0038] 除此之外,本领域技术人员可以理解,上述附图所示出的终端100的结构并不构成对终端100的限定,终端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。比如,终端100中还包括射频电路、输入单元、传感器、音频电路、无线保真(Wireless Fidelity,WiFi)模块、电源、蓝牙模块等部件,在此不再赘述。

[0039] 存储器120可分为操作系统空间和用户空间,操作系统即运行于操作系统空间,原生及第三方应用程序即运行于用户空间。为了保证不同第三方应用程序均能够达到较好的运行效果,操作系统针对不同第三方应用程序为其分配相应的系统资源。然而,同一第三方应用程序中不同应用场景对系统资源的需求也存在差异,比如,在本地资源加载场景下,第三方应用程序对磁盘读取速度的要求较高;在动画渲染场景下,第三方应用程序则对GPU性能的要求较高。而操作系统与第三方应用程序之间相互独立,操作系统往往不能及时感知第三方应用程序当前的应用场景,导致操作系统无法根据第三方应用程序的具体应用场景进行针对性的系统资源适配。

[0040] 在一种可能的实施方式中,如图2所示,为了使操作系统能够区分第三方应用程序的具体应用场景,需要打通第三方应用程序与操作系统之间的数据通信,使得操作系统能够随时获取第三方应用程序当前的场景信息,进而基于当前场景进行针对性的系统资源适配。

[0041] 以操作系统为Android系统为例,存储器120中存储的程序和数据如图3所示,存储器120中可存储有Linux内核层220、系统运行库层240、应用框架层260和应用层280,其中,Linux内核层220、系统运行库层240和应用框架层260属于操作系统空间,应用层280属于用户空间。Linux内核层220为终端100的各种硬件提供了底层的驱动,如显示驱动、音频驱动、摄像头驱动、蓝牙驱动、Wi-Fi驱动、电源管理等。系统运行库层240通过一些C/C++库来为Android系统提供了主要的特性支持。如SQLite库提供了数据库的支持,OpenGL/ES库提供了3D绘图的支持,Webkit库提供了浏览器内核的支持等。在系统运行库层240中还提供有安

卓运行时库(Android Runtime),它主要提供了一些核心库,能够允许开发者使用Java语言来编写Android应用。应用框架层260提供了构建应用程序时可能用到的各种API,开发者也可以通过使用这些API来构建自己的应用程序,比如活动管理、窗口管理、视图管理、通知管理、内容提供者、包管理、通话管理、资源管理、定位管理。应用层280中运行有至少一个应用程序,这些应用程序可以是操作系统自带的原生应用程序,比如联系人程序、短信程序、时钟程序、相机应用等;也可以是第三方开发者所开发的第三方应用程序,比如游戏类应用程序、即时通信程序、相片美化程序、购物程序等。

[0042] 操作系统与第三方应用程序之间一种可行的通信方式如图4所示,第三方应用程序中内嵌有用于与操作系统进行通信的软件开发工具包(Software Development Kit, SDK)。

[0043] 其中,SDK包含若干经过抽象的应用程序编程接口(Application Programming Interface, API),并由操作系统开发者提供给第三方应用程序开发者,并由第三方应用程序开发者将该SDK内嵌到第三方应用程序中。此类第三方应用程序安装并运行在操作系统后,即可调用SDK提供的API与操作系统进行通信。

[0044] 如图4所示,系统运行库层240可以额外包括接口通信系统242。该接口通信系统242可以视为操作系统中的一个子系统,或视为操作系统内嵌的一个应用程序。接口通信系统242中设置有SDK接口,第三方应用程序即调用内嵌SDK的API与该SDK接口之间通过粘合(Binder)的方式进行数据通信。这样,第三方应用程序的应用场景相关的数据就可以通过SDK传输给操作系统。借助内嵌SDK,操作系统还可以主动向第三方应用程序传输数据,或者,操作系统与第三方应用程序之间可以进行双向数据传输。

[0045] 在另一种可行的通信方式中,如图5所示,第三方应用程序还可以采用套接字(Socket)方式与接口通信系统242的Socket接口建立长连接,第三方应用程序的应用场景相关的数据即可以通过该长连接传输给操作系统。

[0046] 如图4和5所示,接口通信系统242中可设置有不同的策略模块,接收到第三方应用程序发送的数据后,接口通信系统242即采用第三方应用程序对应的策略模块对数据进行分析,得到相应的资源适配优化策略。基于分析得到的资源适配优化策略,接口通信系统242通过控制接口通知Linux内核层220进行系统资源适配优化。其中,该控制接口可以采用Sysfs的方式与Linux内核层220进行通信。

[0047] 可选的,接口通信系统242中不同的策略模块可以对应不同的第三方应用程序(即针对不同的应用程序设置策略模块),或者,不同的策略模块对应不同类型的第三方应用程序(即针对不同类型的应用程序设置策略模块),或者,不同的策略模块对应不同的系统资源(即针对不同系统资源设置策略模块),或者,不同的策略模块对应不同的应用场景(即针对不同的以应用场景设置策略模块),本申请实施例并不对策略模块的具体设置方式进行限定。。

[0048] 其中,接口通信系统242还可以通过Binder的方式与应用框架层260进行通信,用于接收应用框架层260发送的前景应用信息,从而基于前景应用信息,仅针对当前前台运行的第三方应用程序进行系统资源优化。

[0049] 以操作系统为IOS系统为例,存储器120中存储的程序和数据如图6所示,IOS系统包括:核心操作系统层320(Core OS layer)、核心服务层340(Core Services layer)、媒体

层360(Media layer)、可触摸层380(Cocoa Touch Layer)。核心操作系统层320包括了操作系统内核、驱动程序以及底层程序框架,这些底层程序框架提供更接近硬件的功能,以供位于核心服务层340的程序框架所使用。核心服务层340提供给应用程序所需要的系统服务和/或程序框架,比如基础(Foundation)框架、账户框架、广告框架、数据存储框架、网络连接框架、地理位置框架、运动框架等等。媒体层360为应用程序提供有关视听方面的接口,如图形图像相关的接口、音频技术相关的接口、视频技术相关的接口、音视频传输技术的无线播放(AirPlay)接口等。可触摸层380为应用程序开发提供了各种常用的界面相关的框架,可触摸层380负责用户在终端100上的触摸交互操作。比如本地通知服务、远程推送服务、广告框架、游戏工具框架、消息用户界面接口(User Interface,UI)框架、用户界面UIKit框架、地图框架等等。

[0050] 在图6所示出的框架中,与大部分应用程序有关的框架包括但不限于:核心服务层340中的基础框架和可触摸层380中的UIKit框架。基础框架提供许多基本的对象类和数据类型,为所有应用程序提供最基本的系统服务,和UI无关。而UIKit框架提供的类是基础的UI类库,用于创建基于触摸的用户界面,iOS应用程序可以基于UIKit框架来提供UI,所以它提供了应用程序的基础架构,用于构建用户界面,绘图、处理和用户交互事件,响应手势等等。

[0051] 其中,在IOS系统中实现第三方应用程序与操作系统数据通信的方式以及原理可参考Android系统,此处不再赘述。

[0052] 然而,采用打通第三方应用程序与操作系统之间数据通信的方式实现基于应用场景的系统资源适配时,第三方应用程序在开发过程中需要进行相应调整,比如内嵌操作系统开发商提供的SDK;并且,为了避免操作系统的数据接口暴露给非法应用程序,造成系统安全隐患,操作系统仅支持与部分第三方应用程序进行数据通信,导致系统资源适配的适用范围较窄。

[0053] 而本申请实施例提供的资源配置方法中,操作系统通过对运行中的游戏应用程序进行截图,并采集机器学习的方式,通过预先训练的游戏场景识别模型对游戏截图进行识别,确定出当前游戏画面对应的游戏场景,进而根据该游戏场景对应的资源配置策略为游戏应用程序分配系统资源。由于整个系统资源分配过程中,操作系统与游戏应用程序之间无需进行数据通信,因此游戏应用程序在开发过程中无需内嵌SDK,且操作系统的数据接口无需暴露给第三方应用程序,从而提高操作系统的安全性。下面采用示意性的实施例进行说明。

[0054] 请参考图7,其示出了本申请一个示例性实施例示出的资源配置方法的流程图。本实施例以该方法应用于图1所示的终端100来举例说明。该方法包括:

[0055] 步骤701,当运行游戏应用程序时,显示游戏画面。

[0056] 当游戏应用程序处于前台运行状态时,终端显示实时的游戏画面,该游戏画面在游戏运行过程中不断发生变化。比如,在游戏启动阶段,该游戏画面为游戏启动画面;完成游戏启动后,该游戏画面为游戏主界面;游戏进行中时,该游戏画面为游戏进行画面。

[0057] 步骤702,对游戏画面进行截图,得到游戏截图。

[0058] 由于不同游戏场景下,游戏画面具有不同的特点,因此本申请实施例中终端基于游戏画面识别当前所处的游戏场景。

[0059] 在一种可能的实施方式中,游戏应用程序启动后,终端操作系统每隔预定时间间隔对游戏画面进行截图,从而得到相应的游戏截图。

[0060] 在另一种可能的实施方式中,游戏应用程序启动后,终端在游戏画面上方显示加速控件,并在接收到对该加速控件的触发操作时进行截图。

[0061] 需要说明的是,操作系统在对游戏画面进行截图时,直接通过后台提取游戏图层,从而获取游戏截图,无需用户手动截图,用户在前台并不会感知。

[0062] 可选的,操作系统截取得到的游戏截图缓存在内存中,并在经过场景识别后直接删除。

[0063] 步骤703,通过游戏场景识别模型对游戏截图进行识别,确定游戏画面对应的游戏场景,游戏场景用于指示游戏应用程序所处的运行阶段。

[0064] 本申请实施例中,终端中预先存储有用于对游戏场景进行识别的游戏场景识别模型,该游戏场景识别模型可以是卷积神经网络模型、深度学习模型等其他用于进行图像识别的模型。

[0065] 针对该游戏场景识别模型的训练方法,在一种可能的实施方式中,该游戏场景识别模型根据携带有游戏场景标识的样本游戏截图训练得到。

[0066] 可选的,终端操作系统将游戏截图输入游戏场景识别模型,得到游戏场景模型输出的若干种候选游戏场景以及对应的概率,进而将概率最高的候选游戏场景确定为游戏画面对应的游戏场景。

[0067] 可选的,该游戏场景包括如下至少一种:游戏启动场景、游戏加载场景、游戏主界面场景和游戏进行场景。

[0068] 示意性的,操作系统将游戏截图输入游戏场景识别模型82后,得到游戏场景识别模型输出的游戏场景为“游戏进行场景”。

[0069] 步骤704,确定游戏场景对应的资源配置策略。

[0070] 终端操作系统根据确定出游戏应用程序当前所处的游戏场景,进一步确定游戏场景对应的资源配置策略,可选的,该资源配置策略中包含游戏场景下各项系统资源的配置信息,该系统资源包括CPU资源、GPU资源、磁盘资源、内存资源和网络资源中的至少一种。相应的,CPU资源对应的配置信息可以包括CPU工作频率和CPU核心开启数;GPU资源对应的配置信息可以包括GPU工作频率、GPU核心开启数;磁盘资源对应的配置信息可以包括磁盘转速和磁盘读写速度;内存资源对应的配置信息可以包括内存频率和内存读写速度;网络资源对应的配置信息可以包括网速和延迟。

[0071] 在一种可能的实施方式中,终端中存储有游戏场景与资源配置策略之间的对应关系,终端操作系统即基于该对应关系确定当前游戏场景对应的资源配置策略。示意性的,该对应关系如表一所示。

[0072] 表一

[0073]	游戏场景	资源配置策略
	游戏加载场景	资源配置策略A
	游戏主界面场景	资源配置策略B
	游戏进行场景	资源配置策略C

[0074] 示意性的,当识别出的游戏场景为“游戏进行场景”时,终端操作系统基于表一确

定出资源配置策略为“资源配置策略C”。

[0075] 步骤705,根据资源配置策略为游戏应用程序分配系统资源。

[0076] 可选的,根据资源配置策略中各项系统资源对应的配置信息,终端操作系统对游戏应用程序进行系统资源分配。

[0077] 在一种可能的实施方式中,当资源配置策略指示的目标系统资源配置低于当前系统资源配置,操作系统则将当前系统资源配置下调至资源配置策略指示的目标系统资源配置,从而在保证游戏应用程序运行质量的同时,降低终端的功耗;当资源配置策略指示的目标系统资源配置高于当前系统资源配置,操作系统则将当前系统资源配置上调至资源配置策略指示的目标系统资源配置,从而提高游戏应用程序的运行质量。

[0078] 综上所述,本申请实施例中,游戏应用程序运行过程中,通过对游戏画面进行截图,并通过预先训练得到的游戏场景识别模型对游戏截图进行识别,确定游戏画面对应的游戏场景,从而根据该游戏场景对应的资源配置策略为游戏应用程序分配系统资源,进而优化游戏应用程序在不同游戏场景下的性能以及功耗;本申请实施例提供的资源配置方法中,游戏应用程序无需接入终端操作系统,终端操作系统也能够智能分析出当前所处的游戏场景,从而免去了在游戏应用程序与操作系统之间建立数据通道的过程,并有助于提高操作系统的安全性。

[0079] 由于不同类型游戏应用程序中游戏画面的画面特征不同,比如游戏进行场景下,卡牌游戏的游戏画面中包含卡牌元素,而竞速游戏的游戏画面中包含赛道元素。因此,为了提高游戏场景识别的准确性,在一种可能的实施方式中,利用标记有游戏类型以及游戏场景标识的样本游戏截图训练游戏场景识别模型;相应的,使用游戏场景识别模型识别游戏场景时,操作系统将当前游戏应用程序的游戏类型和游戏截图作为模型输入,输入游戏场景识别模型中。下面采用示意性的实施例进行说明。

[0080] 请参考图8,其示出了本申请另一个示例性实施例示出的资源配置方法的流程图。本实施例以该方法应用于图1所示的终端100来举例说明。该方法包括:

[0081] 步骤801,当运行游戏应用程序时,显示游戏画面。

[0082] 本步骤的实施方式可以参考上述步骤701,本实施例在此不再赘述。

[0083] 步骤802,获取游戏应用程序的应用标识。

[0084] 当终端中安装有能够与操作系统建立数据通道,并进行数据通信的目标游戏应用程序时,此类目标游戏应用程序在运行过程中,可以通过数据通道向操作系统上报场景信息(包含游戏场景),操作系统即可基于获取到的场景信息为目标游戏应用程序进行系统资源配置。因此,针对目标游戏应用程序,操作系统无需进行截图和游戏截图识别。

[0085] 为了避免对目标游戏应用程序进行截图以及场景识别,造成终端处理资源浪费,在一种可能的实施方式中,操作系统获取当前运行游戏应用程序的应用标识。其中,该应用标识可以为游戏应用程序的应用包名。

[0086] 步骤803,检测应用标识是否属于预设游戏列表,预设游戏列表中包含目标游戏应用程序的标识,目标游戏应用程序在运行过程中通过与操作系统之间的数据通道向发送场景信息,操作系统用于根据场景信息指示的游戏场景确定资源配置策略。

[0087] 可选的,终端中存储有预设游戏列表,该预设游戏列表中即包含目标游戏应用程序的标识。其中,该预设游戏列表是终端从后台服务器出定期更新下载。

[0088] 终端操作系统检测当前游戏应用程序的应用标识是否属于预设游戏列表,若属于,则确定当前游戏应用程序为目标应用程序,后续无需进行截图场景识别;若不属于,则确定当前游戏应用程序无法直接向操作系统发送场景信息,并执行步骤804。

[0089] 在其他可能的实施方式中,当应用标识不属于预设游戏列表时,操作系统进一步检测应用标识是否属于游戏白名单,若属于游戏白名单,则执行步骤804。其中,该游戏白名单中包含游戏性能优化的游戏应用程序。

[0090] 步骤804,若应用标识不属于预设游戏列表,则对游戏画面进行截图,得到游戏截图。

[0091] 其中,终端操作系统进行截图的步骤可以参考上述步骤702,本实施例在此不再赘述。

[0092] 步骤805,获取游戏应用程序对应的游戏类型。

[0093] 为了提高后续游戏场景识别的准确性,操作系统获取游戏应用程序对应的游戏类型,该游戏类型可以包括如下至少一种:第一人称射击游戏(First-Person Shooter game, FPS)、第三人称射击游戏TPS(Third-Person Shooter game, FPS)、多人在线战术竞技游戏(Multiplayer Online Battle Arena, MOBA)、卡牌游戏、竞速游戏、音乐游戏、模拟经营游戏、冒险游戏(Adventure Game, AVG)、格斗游戏(Fighting Game, FTG)、角色扮演游戏(Role-Playing Game, RPG)和沙盒游戏。

[0094] 在一种可能的实施方式中,操作系统获取游戏应用程序的应用名称或应用包名,从而根据该应用名称或应用包名查询该游戏应用程序的游戏类型。本申请实施例并不对获取游戏类型的具体方式进行限定。

[0095] 步骤806,将游戏类型和游戏截图输入游戏场景识别模型,得到游戏场景识别模型输出的游戏场景。

[0096] 进一步的,在进行游戏场景识别时,操作系统将游戏类型和游戏截图共同输入游戏场景识别模型,从而得到游戏场景识别模型输出的若干种候选游戏场景及其概率,并将最高概率对应的候选游戏场景确定为当前游戏画面对应的游戏场景。

[0097] 步骤807,根据游戏类型以及游戏场景,确定游戏场景对应的资源配置策略。

[0098] 相同游戏场景下,由于不同游戏类型的游戏应用程序对系统资源的需求不同,比如,同样在游戏进行场景下,卡牌游戏对系统资源的需求低于MOBA游戏对系统资源的需求不同。因此,操作系统同时基于游戏类型以及游戏场景,确定当前游戏场景对应的资源配置策略。

[0099] 在一种可能的实施方式中,终端中存储有游戏类型、游戏场景以及资源配置策略三者之间的对应关系,终端操作系统即基于该对应关系确定当前游戏场景对应的资源配置策略。示意性的,该对应关系如表二所示。

[0100] 表二

[0101]	游戏类型	游戏场景	资源配置策略
	卡牌游戏	游戏加载场景	资源配置策略A1
	MOBA游戏	游戏加载场景	资源配置策略A2
	卡牌游戏	游戏运行场景	资源配置策略C1
	MOBA游戏	游戏运行场景	资源配置策略C2

[0102] 示意性的,当识别出的游戏场景为“游戏进行场景”,且游戏类型为“MOBA游戏”时,终端操作系统基于表二确定出资源配置策略为“资源配置策略C2”。

[0103] 由于资源配置策略所指示系统资源配置的高低还与游戏大小相关(游戏大小越大,对系统资源的需求越高),且与终端当前的电量相关(电量较低时,过高的系统资源配置会影响终端续航),因此为了进一步提高确定出的资源配置策略的准确性和适用性,在另一种可能的实施方式中,在确定游戏场景对应的资源配置策略之前,终端操作系统获取当前剩余电量以及游戏应用程序的应用程序大小,该应用程序大小用于指示游戏应用程序所占的存储空间。在确定游戏场景对应的资源配置策略时,终端操作系统将当前剩余电量、应用程序大小、游戏类型以及游戏场景输入资源配置模型,得到资源配置模型输出的资源配置策略。

[0104] 可选的,资源配置模型为基于深度学习的神经网络模型,且训练该资源配置模型时使用的样本输入参数包括样本电量、样本应用程序大小、样本类型和样本游戏场景,样本输出参数则包括样本资源配置策略。本申请实施例并不对资源配置模型的训练过程进行限定。

[0105] 示意性的,终端操作系统获取到当前剩余电量为“50%”、应用程序大小为“1.5GB”、游戏类型为“MOBA游戏”且游戏场景为“游戏进行场景”,将上述参数输入资源配置模型后,得到资源配置模型输出的资源配置策略。

[0106] 需要说明的是,终端还可以将其他与资源配置策略相关的参数作为模型输入进行模型训练,以此提高资源配置模型输出的资源配置策略的准确性。

[0107] 步骤808,根据资源配置策略为游戏应用程序分配系统资源。

[0108] 本步骤的实施方式可以参考上述步骤705,本实施例在此不再赘述。

[0109] 本实施例中,考虑到不同类型游戏具有不同的游戏画面特征,通过将当前游戏应用的游戏类型以及游戏截图一同输入游戏场景识别模型,得到游戏场景识别模型输出的游戏场景,提高了游戏场景识别的准确性。

[0110] 同时,考虑到不同类型游戏对系统资源的不同需求,根据当前游戏应用的游戏类型以及当前所处的游戏场景,确定游戏场景对应的资源配置策略,提高了确定出的资源配置策略的准确性。

[0111] 为了进一步提高游戏场景识别模型的识别准确性,在一种可能的实施方式中,操作系统根据目标应用程序上报场景信息是的游戏画面,对游戏场景识别模型进行反向训练。在图8的基础上,如图9所示,步骤808之后还可以包括如下步骤。

[0112] 步骤809,当接收到目标游戏应用程序发送的场景信息时,对目标游戏应用程序的当前游戏画面进行截图,得到样本游戏截图。

[0113] 在一种可能的实施方式中,当检测应用标识是否属于预设游戏列表时,终端操作系统在目标游戏应用程序运行过程中,接收目标游戏应用程序发送的包含游戏场景的场景信息,并基于场景信息对目标游戏应用程序进行资源配置;与此同时,操作系统对当前游戏画面进行截图,得到与目标游戏应用程序当前游戏场景对应样本游戏截图。

[0114] 步骤810,根据场景信息指示的游戏场景以及样本游戏截图,通过反向传播算法对游戏场景识别模型进行训练。

[0115] 进一步的,终端操作系统根据场景信息指示的游戏场景以及样本游戏截图,对原

有的游戏场景识别模型进行反向传播训练,以此提高游戏场景识别模型的识别准确性。

[0116] 可选的,终端操作系统可以根据场景信息指示的游戏场景、目标游戏应用程序的游戏类型以及样本游戏截图,对游戏场景识别模型进行反向传播训练。

[0117] 在一种可能的实施方式中,终端操作系统将样本游戏截图输入原有的游戏场景识别模型,得到模型输出的若干种候选游戏场景及其概率,并基于场景信息所指示游戏场景与各种候选游戏场景对应的概率,进行反向传播训练。

[0118] 本实施例中,终端操作系统在接收到目标游戏应用程序上报的场景信息时,对游戏画面进行截图,从而根据游戏截图和场景信息指示的游戏场景对游戏场景识别模型进行反向传播训练,以此提高游戏场景识别模型的识别准确性。

[0119] 在一种可能的实施方式中,在图8的基础上,如图10所示,步骤804可以包括如下步骤:

[0120] 步骤804A,显示加速控件。

[0121] 在一种可能的实施方式中,终端操作系统显示加速控件,并接收加速控件上的触发操作。其中,该加速控件可以是位于游戏画面上层的悬浮图标,或者显示在快捷栏中的快捷功能图标(比如显示在显示屏异性区域的加速图标),且触发操作可以是点击操作、按压操作或长按操作中的至少一种。本申请实施例对此不做限定。

[0122] 示意性的,如图11所示,游戏画面1101的上层显示有加速控件1102。

[0123] 步骤804B,当接收到对加速控件的触发信号时,对游戏画面进行截图,得到游戏截图。

[0124] 当接收到对加速控件的触发信号时,终端确定需要对游戏应用程序进行性能优化,从而执行后续截图以及游戏场景识别等操作。其具体实施方式可以参考图8实施例,本实施例在此不再赘述。

[0125] 由于游戏运行过程中,游戏场景在不断发生变化,比如游戏结束后,将由游戏进行场景切换至游戏主界面场景,因此操作系统需要及时调试资源配置策略,相应的,如图10所示,步骤808之后还包括如下步骤。

[0126] 步骤811,每隔预定时间间隔识别当前游戏画面对应的当前游戏场景。

[0127] 在一种可能的实施方式中,终端操作系统每隔预定时间间隔,通过截图和模型识别操作,重新识别当前游戏画面对应的当前游戏场景,并检测当前游戏场景与上一次识别出的游戏场景是否一致,若一致,则保持当前资源配置方式;若不一致,则通过下述步骤812重新确定当前游戏场景对应的资源配置策略。

[0128] 可选的,该预定时间间隔可以根据上一次识别游戏场景对应的预计持续时长确定。比如,对于MOBA游戏,当上一次识别出的游戏场景为“游戏进行场景”时,终端操作系统确定预定时间间隔为30分钟。

[0129] 步骤812,若当前游戏场景发生变化,则重新确定当前游戏场景对应的资源配置策略。

[0130] 当游戏场景发生变化时,终端操作系统重新确定资源配置策略,并根据重新确定的资源配置策略进行系统资源配置。

[0131] 请参考图12,其示出了本申请一个实施例提供的资源配置装置的结构框图。该资源配置装置可以通过软件、硬件或者两者的结合实现成为终端的全部或一部分。该装置包

括：

[0132] 显示模块1201,用于当运行游戏应用程序时,显示游戏画面；

[0133] 截图模块1202,用于对所述游戏画面进行截图,得到游戏截图；

[0134] 场景识别模块1203,用于通过游戏场景识别模型对所述游戏截图进行识别,确定所述游戏画面对应的游戏场景,所述游戏场景用于指示所述游戏应用程序所处的运行阶段；

[0135] 策略确定模块1204,用于确定所述游戏场景对应的资源配置策略；

[0136] 资源分配模块1205,用于根据所述资源配置策略为所述游戏应用程序分配系统资源。

[0137] 可选的,所述场景识别模块1203,包括：

[0138] 类型获取单元,用于获取所述游戏应用程序对应的游戏类型；

[0139] 场景预测单元,用于将所述游戏类型和所述游戏截图输入所述游戏场景识别模型,得到所述游戏场景识别模型输出的所述游戏场景；

[0140] 所述策略确定模块1204,用于：

[0141] 根据所述游戏类型以及所述游戏场景,确定所述游戏场景对应的所述资源配置策略；

[0142] 其中,所述游戏类型包括如下至少一种:FPS、TPS、MOBA、卡牌游戏、竞速游戏、音乐游戏、模拟经营游戏、AVG、FTG、RPG和沙盒游戏。

[0143] 可选的,所述装置,还包括：

[0144] 第一获取模块,用于获取当前剩余电量以及所述游戏应用程序的应用程序大小,所述应用程序大小用于指示所述游戏应用程序所占的存储空间；

[0145] 所述策略确定模块1204,还用于：

[0146] 将所述当前剩余电量、所述应用程序大小、所述游戏类型以及所述游戏场景输入资源配置模型,得到所述资源配置模型输出的所述资源配置策略。

[0147] 可选的,所述装置还包括：

[0148] 第二获取模块,用于获取所述游戏应用程序的应用标识；

[0149] 检测模块,用于检测所述应用标识是否属于预设游戏列表,所述预设游戏列表中包含目标游戏应用程序的标识,所述目标游戏应用程序在运行过程中通过与操作系统之间的数据通道向发送场景信息,所述操作系统用于根据所述场景信息指示的所述游戏场景确定所述资源配置策略；

[0150] 所述截图模块1202,用于若所述应用标识不属于所述预设游戏列表,则执行所述对游戏画面进行截图,得到游戏截图的步骤。

[0151] 可选的,所述装置还包括：

[0152] 样本截图模块,用于当接收到所述目标游戏应用程序发送的所述场景信息时,对所述目标游戏应用程序的当前游戏画面进行截图,得到样本游戏截图；

[0153] 训练模块,用于根据所述场景信息指示的所述游戏场景以及所述样本游戏截图,通过反向传播算法对所述游戏场景识别模型进行训练。

[0154] 可选的,所述截图模块1202,包括：

[0155] 控件显示单元,用于在所述游戏画面上层显示加速控件；

[0156] 截图单元,用于当接收到对所述加速控件的触发信号时,对所述游戏画面进行截图,得到所述游戏截图;

[0157] 所述装置还包括:

[0158] 定时识别模块,用于每隔预定时间间隔识别当前游戏画面对应的当前游戏场景;

[0159] 策略重置模块,用于若所述当前游戏场景发生变化,则重新确定所述当前游戏场景对应的所述资源配置策略。

[0160] 可选的,所述系统资源的资源类型包括CPU资源、GPU资源、磁盘资源、内存资源和网络资源中的至少一种。

[0161] 综上所述,本申请实施例中,游戏应用程序运行过程中,通过对游戏画面进行截图,并通过预先训练得到的游戏场景识别模型对游戏截图进行识别,确定游戏画面对应的游戏场景,从而根据该游戏场景对应的资源配置策略为游戏应用程序分配系统资源,进而优化游戏应用程序在不同游戏场景下的性能以及功耗;本申请实施例提供的资源配置方法中,游戏应用程序无需接入终端操作系统,终端操作系统也能够智能分析出当前所处的游戏场景,从而免去了在游戏应用程序与操作系统之间建立数据通道的过程,并有助于提高操作系统的安全性。

[0162] 本实施例中,考虑到不同类型游戏具有不同的游戏画面特征,通过将当前游戏应用的游戏类型以及游戏截图一同输入游戏场景识别模型,得到游戏场景识别模型输出的游戏场景,提高了游戏场景识别的准确性。

[0163] 同时,考虑到不同类型游戏对系统资源的不同需求,根据当前游戏应用的游戏类型以及当前所处的游戏场景,确定游戏场景对应的资源配置策略,提高了确定出的资源配置策略的准确性。

[0164] 本实施例中,终端操作系统在接收到目标游戏应用程序上报的场景信息时,对游戏画面进行截图,从而根据游戏截图和场景信息指示的游戏场景对游戏场景识别模型进行反向传播训练,以此提高游戏场景识别模型的识别准确性。

[0165] 本申请实施例还提供了一种计算机可读介质,该计算机可读介质存储有至少一条指令,所述至少一条指令由所述处理器加载并执行以实现如上各个实施例所述的资源配置方法。

[0166] 本申请实施例还提供了一种计算机程序产品,该计算机程序产品存储有至少一条指令,所述至少一条指令由所述处理器加载并执行以实现如上各个实施例所述的资源配置方法。

[0167] 本领域技术人员应该可以意识到,在上述一个或多个示例中,本申请实施例所描述的功能可以用硬件、软件、固件或它们的任意组合来实现。当使用软件实现时,可以将这些功能存储在计算机可读介质中或者作为计算机可读介质上的一个或多个指令或代码进行传输。计算机可读介质包括计算机存储介质和通信介质,其中通信介质包括便于从一个地方向另一个地方传送计算机程序的任何介质。存储介质可以是通用或专用计算机能够存取的任何可用介质。

[0168] 以上所述仅为本申请的较佳实施例,并不用以限制本申请,凡在本申请的精神和原则之内,所作的任何修改、等同替换、改进等,均应包含在本申请的保护范围之内。

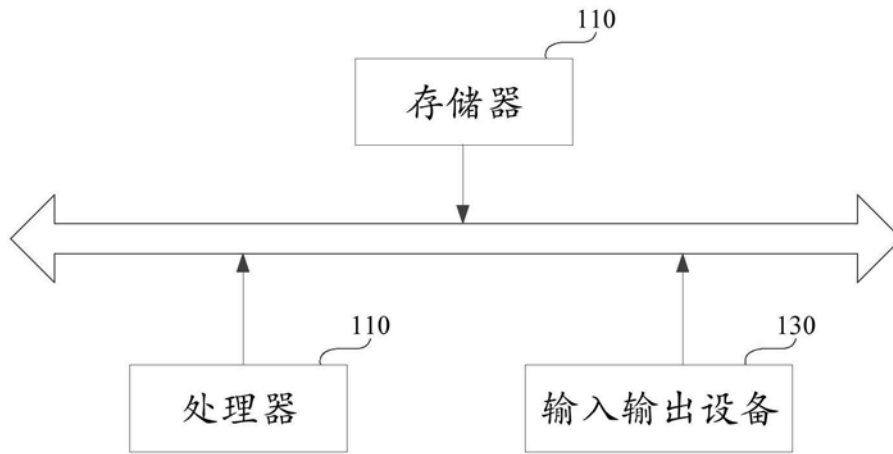


图1

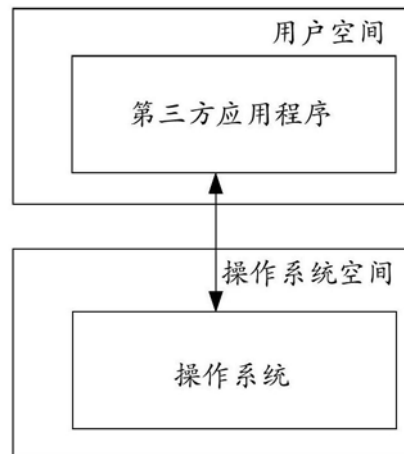


图2

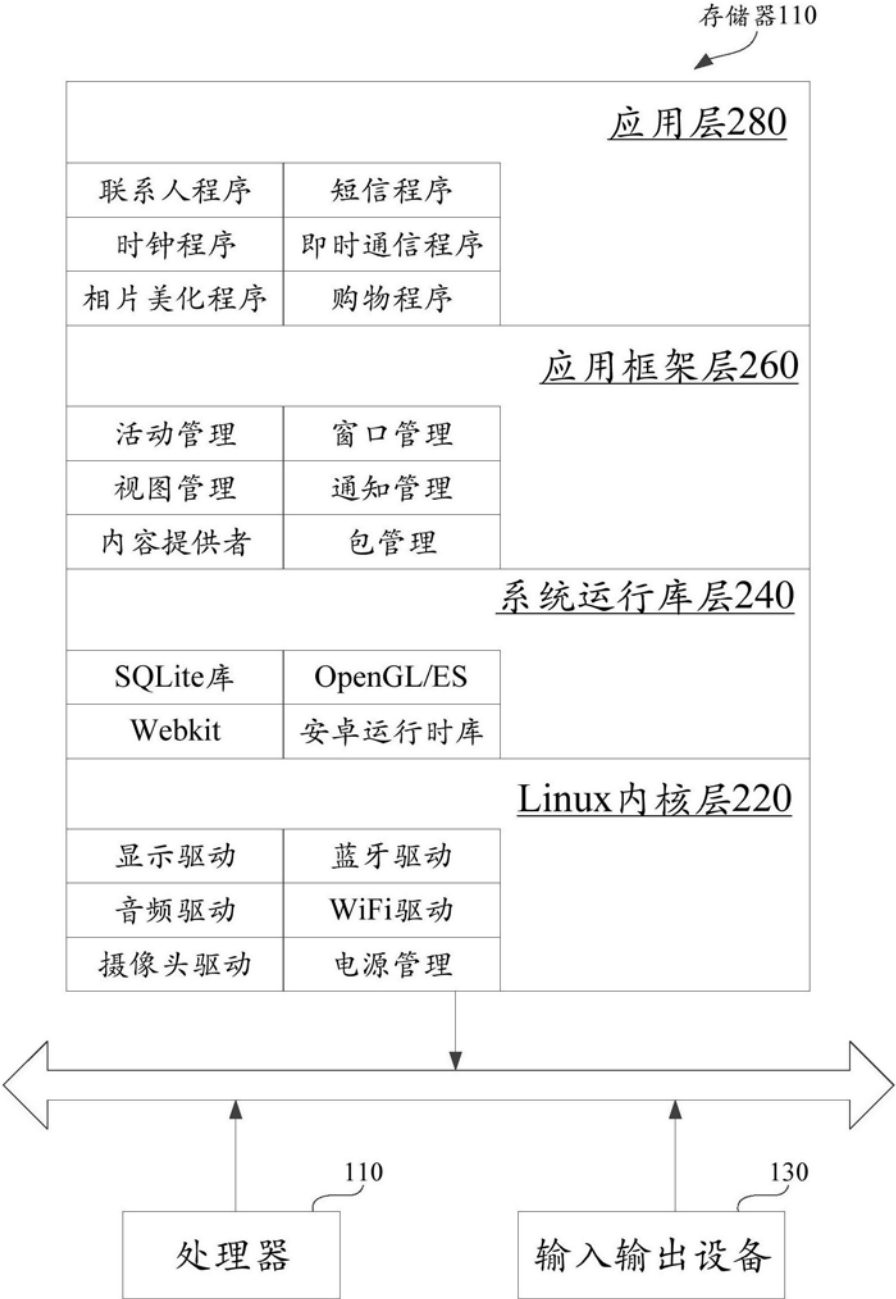


图3

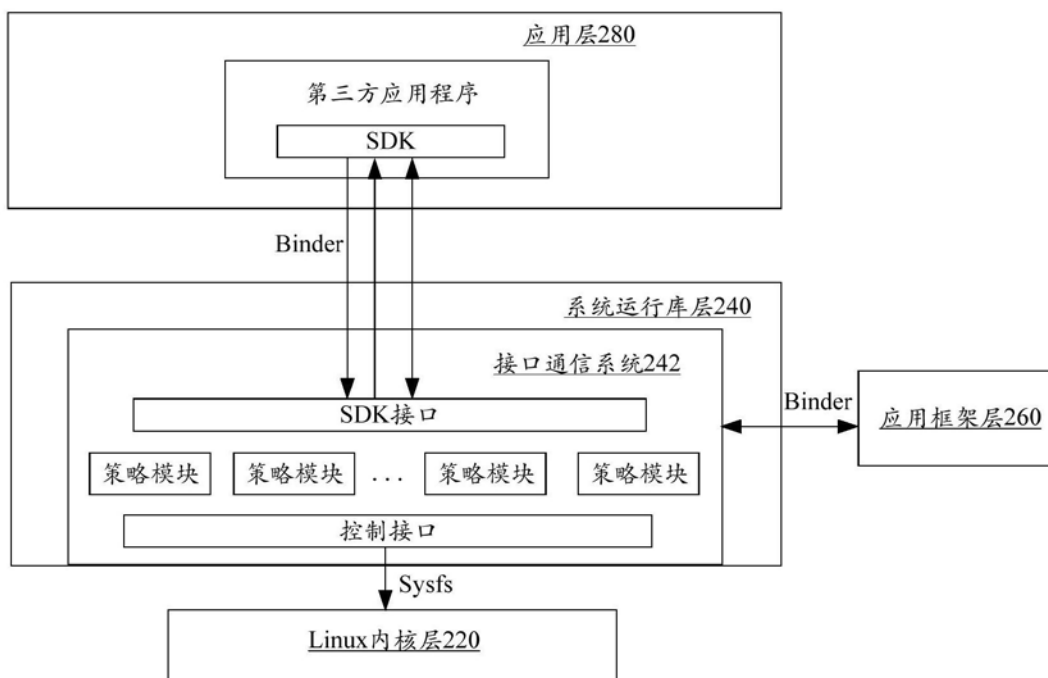


图4

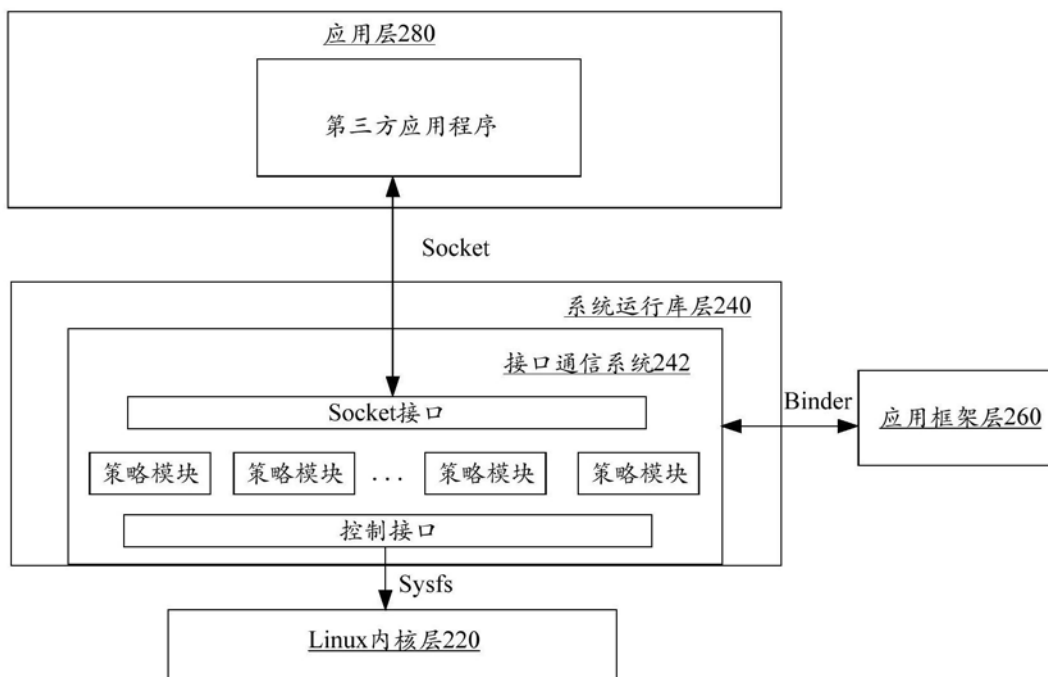


图5

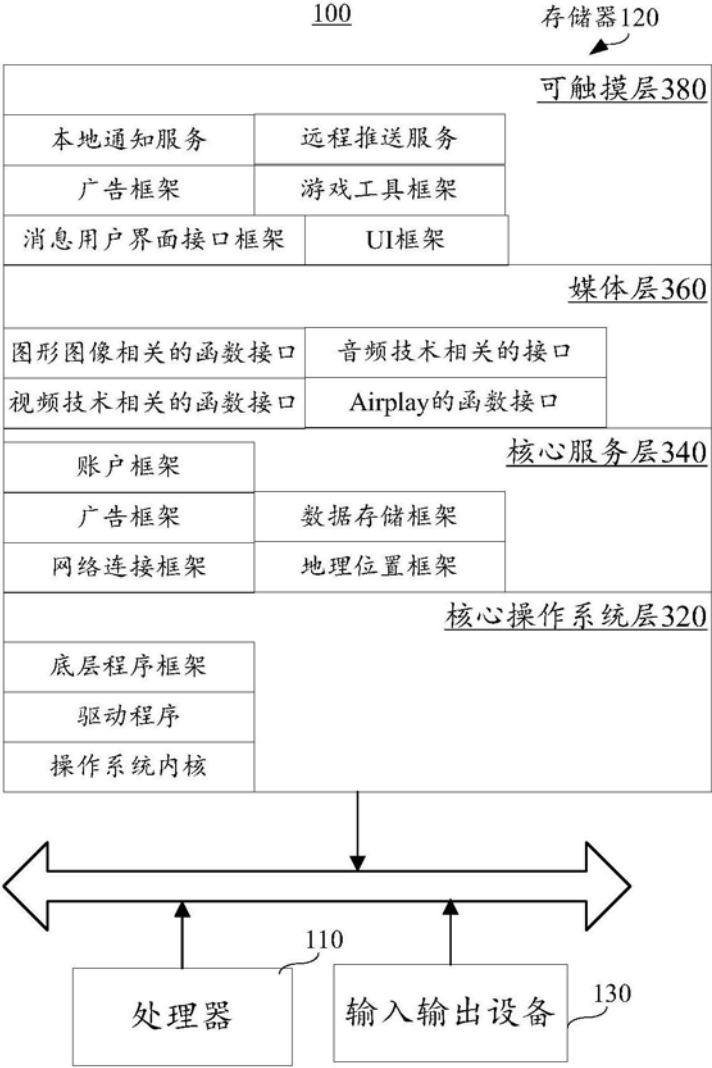


图6

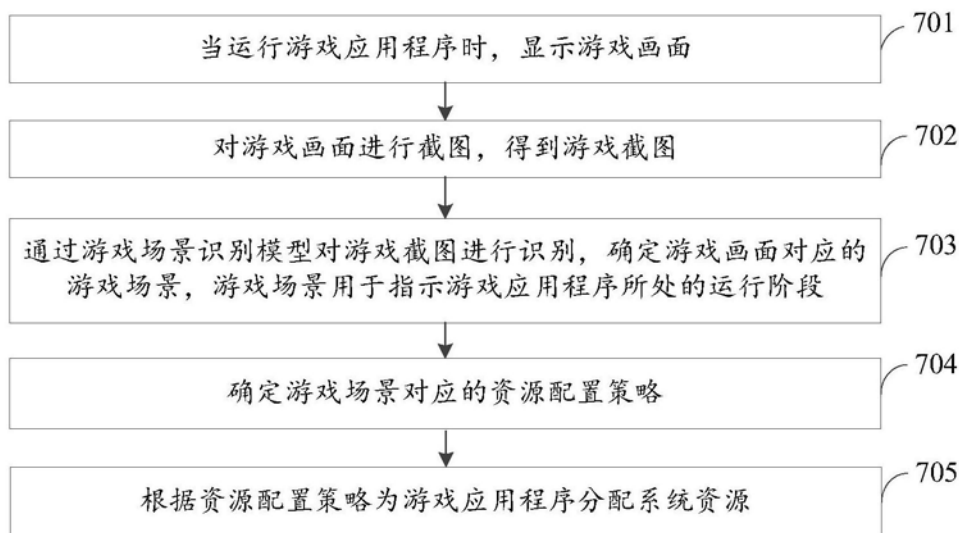


图7

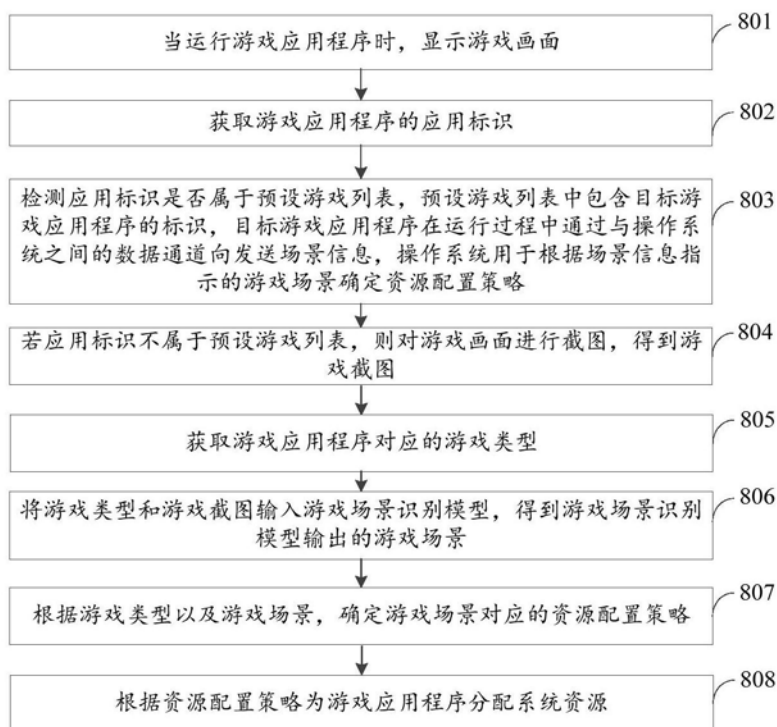


图8

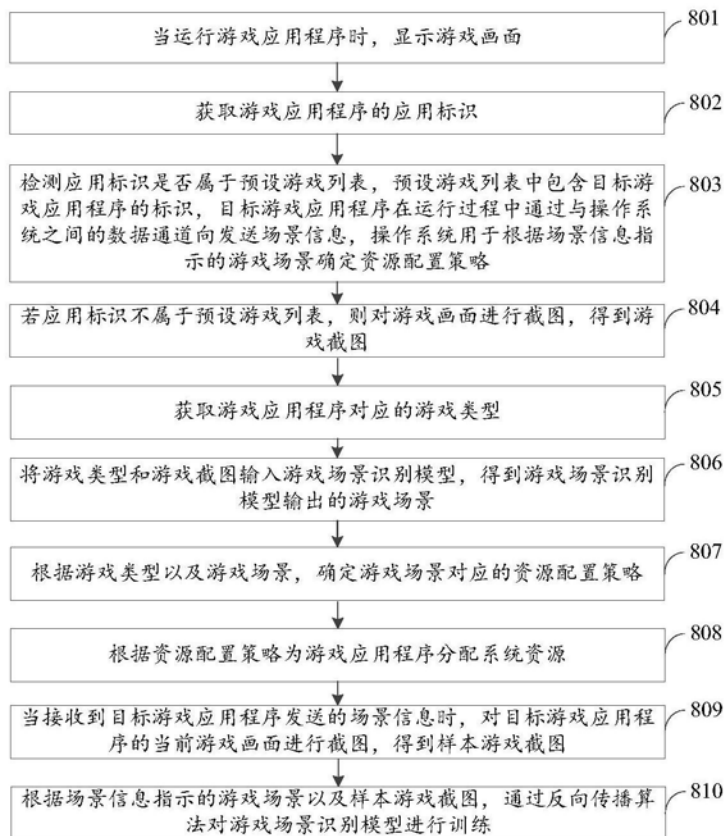


图9

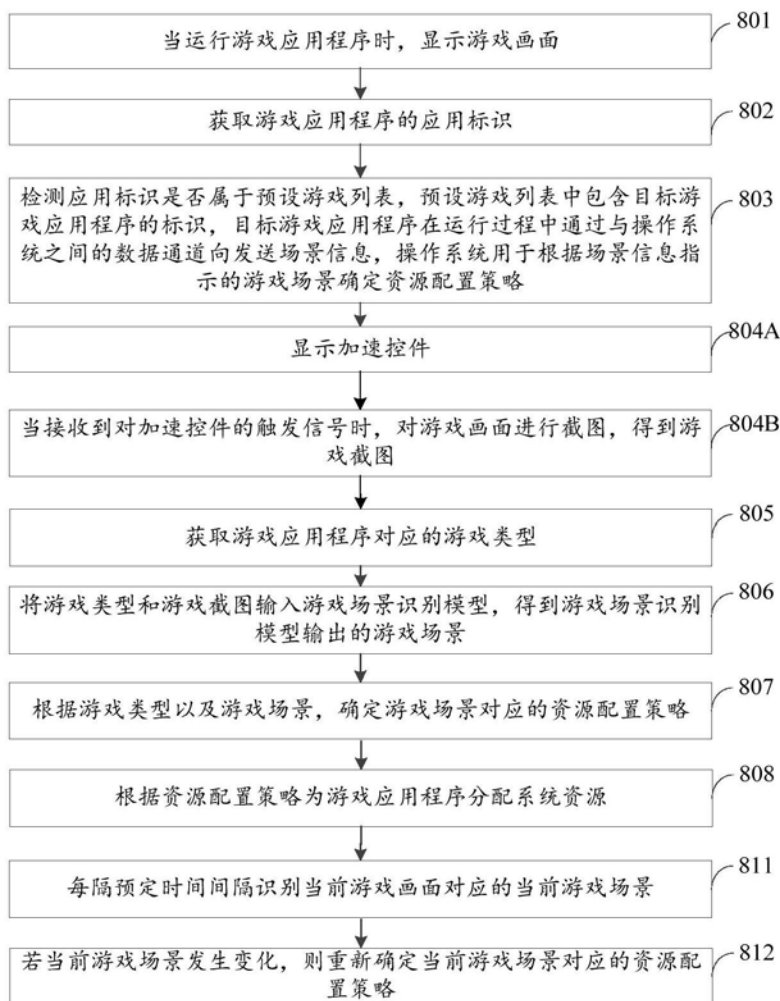


图10

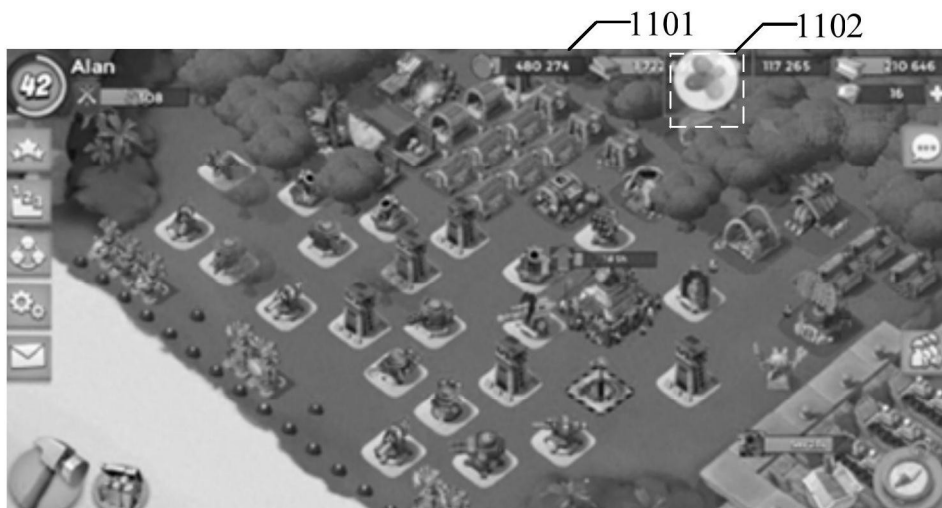


图11

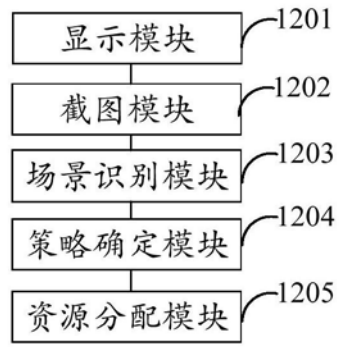


图12