



(12) 发明专利

(10) 授权公告号 CN 109718552 B

(45) 授权公告日 2022. 11. 04

(21) 申请号 201711038435.5

(22) 申请日 2017.10.30

(65) 同一申请的已公布的文献号
申请公布号 CN 109718552 A

(43) 申请公布日 2019.05.07

(73) 专利权人 腾讯科技(成都)有限公司
地址 610041 四川省成都市高新区拓东新街81号天府软件园二期C区4号楼

(72) 发明人 王怡然

(74) 专利代理机构 北京三高永信知识产权代理有限公司 11138
专利代理师 祝亚男

(51) Int. Cl.
A63F 13/55 (2014.01)

(56) 对比文件

CN 106341427 A, 2017.01.18

CN 104902105 A, 2015.09.09

CN 105338081 A, 2016.02.17

EP 1892023 A3, 2010.11.17

JP 2017012251 A, 2017.01.19

审查员 陈善学

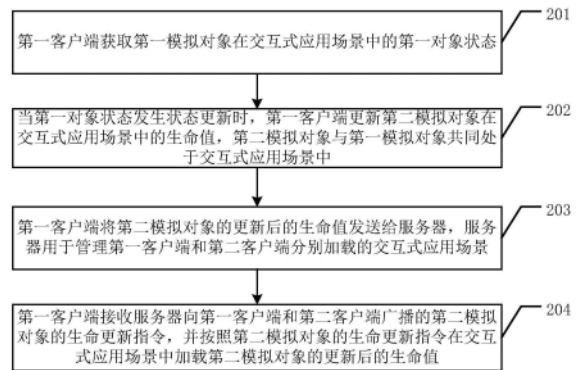
权利要求书2页 说明书14页 附图5页

(54) 发明名称

一种基于模拟对象的生命值控制方法和客户端

(57) 摘要

本发明实施例公开了一种基于模拟对象的生命值控制方法和客户端,用于实现对第二模拟对象的生命值控制,可以保证模拟对象的状态更新进度,不需要在交互式应用场景中增加更多的模拟对象,从而降低设计复杂度。该方法可包括:第一客户端获取第一模拟对象在交互式应用场景中的第一对象状态;当第一对象状态发生状态更新时,第一客户端更新第二模拟对象在交互式应用场景中的生命值,第二模拟对象与第一模拟对象共同处于交互式应用场景中;第一客户端将第二模拟对象的更新后的生命值发送给服务器,服务器用于管理第一客户端和第二客户端分别加载的交互式应用场景;第一客户端接收服务器向第一客户端和第二客户端广播的第二模拟对象的生命更新指令,并按照生命更新指令在交互式应用场景中加载第二模拟对象的更新后的生命值。



1. 一种基于防御型虚拟机关的生命值控制方法,其特征在于,包括:

第一客户端获取第一防御型虚拟机关在交互式应用场景中的第一状态,所述第一状态指示所述第一防御型虚拟机关是否被摧毁;

当所述第一状态发生状态更新时,所述第一客户端获取第三防御型虚拟机关在所述交互式应用场景中的第二状态,所述第二状态指示所述第三防御型虚拟机关是否被摧毁;

当所述第二状态发生状态更新时,所述第一客户端更新第二防御型虚拟机关在所述交互式应用场景中的生命值;其中,在所述交互式应用场景中敌方虚拟对象的第一预设进攻路线依次经过所述第一防御型虚拟机关、所述第三防御型虚拟机关和所述第二防御型虚拟机关;

所述第一客户端将所述第二防御型虚拟机关的更新后的生命值发送给服务器,所述服务器用于管理所述第一客户端和第二客户端分别加载的所述交互式应用场景;

所述第一客户端接收所述服务器向所述第一客户端和所述第二客户端广播的所述第二防御型虚拟机关的生命更新指令,并按照所述第二防御型虚拟机关的生命更新指令在所述交互式应用场景中加载所述第二防御型虚拟机关的更新后的生命值。

2. 根据权利要求1所述的方法,其特征在于,所述当所述第一状态发生状态更新时,所述第一客户端获取第三防御型虚拟机关在所述交互式应用场景中的第二状态,包括:

当所述第一状态发生状态更新时,所述第一客户端为所述第三防御型虚拟机关增加一个计时器并触发计时器开始计时;

所述第一客户端获取所述第三防御型虚拟机关在所述交互式应用场景中的所述第二状态;

所述当所述第二状态发生状态更新时,所述第一客户端更新第二防御型虚拟机关在所述交互式应用场景中的生命值,包括:

所述第一客户端判断所述第三防御型虚拟机关的第二状态是否在所述计时器的计时周期内发生状态更新;

当所述第二状态在所述计时器的计时周期内发生状态更新时,所述第一客户端更新所述第二防御型虚拟机关在所述交互式应用场景中的生命值。

3. 根据权利要求1所述的方法,其特征在于,

所述当所述第一状态发生状态更新时,所述第一客户端获取第三防御型虚拟机关在所述交互式应用场景中的第二状态,包括:

当所述第一状态更新为被摧毁时,所述第一客户端获取所述第三防御型虚拟机关在所述交互式应用场景中的所述第二状态;

所述当所述第二状态发生状态更新时,所述第一客户端更新第二防御型虚拟机关在所述交互式应用场景中的生命值,包括:

当所述第二状态更新为被摧毁时,所述第一客户端按照生命延长策略增加所述第二防御型虚拟机关在所述交互式应用场景中的生命值。

4. 根据权利要求1所述的方法,其特征在于,所述第一客户端更新第二防御型虚拟机关在所述交互式应用场景中的生命值,包括:

所述第一客户端为所述第二防御型虚拟机关增加持续预置时长的防御对象,所述防御对象用于保护所述第二防御型虚拟机关。

5. 根据权利要求1至4中任一项所述的方法,其特征在于,所述方法还包括:

在所述第二防御型虚拟机关的生命值存续期间,当所述敌方虚拟对象的第二预设进攻路线的另一第一防御型虚拟机关发生状态更新时,所述第一客户端禁止再次更新所述第二防御型虚拟机关在所述交互式应用场景中的生命值,所述第一预设进攻路线和所述第二预设进攻路线均经过所述第二防御型虚拟机关。

6. 一种客户端,其特征在于,所述客户端具体为第一客户端,所述第一客户端包括:

状态获取模块,用于获取第一防御型虚拟机关在交互式应用场景中的第一状态,所述第一状态指示所述第一防御型虚拟机关是否被摧毁;

生命值更新模块,包括状态获取子模块和生命值更新子模块;

所述状态获取子模块,用于当所述第一状态发生状态更新时,获取第三防御型虚拟机关在所述交互式应用场景中的第二状态,所述第二状态指示所述第三防御型虚拟机关是否被摧毁;

所述生命值更新子模块,用于当所述第二状态发生状态更新时,更新第二防御型虚拟机关在所述交互式应用场景中的生命值;其中,在所述交互式应用场景中敌方虚拟对象的第一预设进攻路线依次经过所述第一防御型虚拟机关、所述第三防御型虚拟机关和所述第二防御型虚拟机关;

发送模块,用于将所述第二防御型虚拟机关的更新后的生命值发送给服务器,所述服务器用于管理所述第一客户端和第二客户端分别加载的所述交互式应用场景;

显示模块,用于接收所述服务器向所述第一客户端和所述第二客户端广播的所述第二防御型虚拟机关的生命更新指令,并按照所述第二防御型虚拟机关的生命更新指令在所述交互式应用场景中加载所述第二防御型虚拟机关的更新后的生命值。

7. 根据权利要求6所述的客户端,其特征在于,所述生命值更新模块,还包括:

计时子模块,用于当所述第一状态发生状态更新时,为所述第三防御型虚拟机关增加一个计时器并触发计时器开始计时;

所述生命值更新子模块,具体用于判断所述第三防御型虚拟机关的第二状态是否在所述计时器的计时周期内发生状态更新;当所述第二状态在所述计时器的计时周期内发生状态更新时,更新所述第二防御型虚拟机关在所述交互式应用场景中的生命值。

8. 根据权利要求6所述的客户端,其特征在于,所述生命值更新模块,具体用于为所述第二防御型虚拟机关增加持续预置时长的防御对象,所述防御对象用于保护所述第二防御型虚拟机关。

9. 根据权利要求6至8中任一项所述的客户端,其特征在于所述生命值更新模块,还用于在所述第二防御型虚拟机关的生命值存续期间,当所述敌方虚拟对象的第二预设进攻路线的另一第一防御型虚拟机关发生状态更新时,禁止再次更新所述第二防御型虚拟机关在所述交互式应用场景中的生命值,所述第一预设进攻路线和所述第二预设进攻路线均经过所述第二防御型虚拟机关。

10. 一种计算机可读存储介质,包括指令,当其在计算机上运行时,使得计算机执行如权利要求1-5任意一项所述的方法。

一种基于模拟对象的生命值控制方法和客户端

技术领域

[0001] 本发明涉及计算机技术领域,尤其涉及一种基于模拟对象的生命值控制方法和客户端。

背景技术

[0002] 在多人在线战术竞技(英文全称:Multiplayer Online Battle Arena,英文简称:MOBA)游戏中,因为游戏的快节奏进展,有时候会出现一方用户所操作的角色在取得一定优势后,快速连续推进游戏进展的情况,直接致使本场游戏尽快结束,而这使得游戏中的另一方用户的不良体验大大提升。

[0003] 为了解决前述的被攻击一方的不良体验问题,现有技术对于游戏场景的管理方案中,通常采用如下的实现方案:需要在游戏设计时为游戏防守方增设更多的机关等道具,从而使得游戏进攻方的进攻节奏降下来,使得进攻方推进困难,节奏变慢。这种解决方案虽然保证了被进攻方的游戏体验,却使得进攻方的推进节奏明显降低,又导致进攻方的游戏体验下降,另外增设更多的道具也会增加游戏的设计复杂度。

发明内容

[0004] 本发明实施例提供了一种基于模拟对象的生命值控制方法和客户端,用于实现对第二模拟对象的生命值控制,可以保证模拟对象的状态更新进度,不需要在交互式应用场景中增加更多的模拟对象,从而降低设计复杂度。

[0005] 为解决上述技术问题,本发明实施例提供以下技术方案:

[0006] 第一方面,本发明实施例提供一种基于模拟对象的生命值控制方法,包括:

[0007] 第一客户端获取第一模拟对象在交互式应用场景中的第一对象状态;

[0008] 当所述第一对象状态发生状态更新时,所述第一客户端更新第二模拟对象在所述交互式应用场景中的生命值,所述第二模拟对象与所述第一模拟对象共同处于所述交互式应用场景中;

[0009] 所述第一客户端将所述第二模拟对象的更新后的生命值发送给服务器,所述服务器用于管理所述第一客户端和第二客户端分别加载的所述交互式应用场景;

[0010] 所述第一客户端接收所述服务器向所述第一客户端和所述第二客户端广播的所述第二模拟对象的生命更新指令,并按照所述第二模拟对象的生命更新指令在所述交互式应用场景中加载所述第二模拟对象的更新后的生命值。

[0011] 第二方面,本发明实施例还提供一种客户端,该客户端具体为第一客户端,第一客户端包括:

[0012] 状态获取模块,用于获取第一模拟对象在交互式应用场景中的第一对象状态;

[0013] 生命值更新模块,用于当所述第一对象状态发生状态更新时,更新第二模拟对象在所述交互式应用场景中的生命值,所述第二模拟对象与所述第一模拟对象共同处于所述交互式应用场景中;

[0014] 发送模块,用于将所述第二模拟对象的更新后的生命值发送给服务器,所述服务器用于管理所述第一客户端和第二客户端分别加载的所述交互式应用场景;

[0015] 显示模块,用于接收所述服务器向所述第一客户端和所述第二客户端广播的所述第二模拟对象的生命更新指令,并按照所述第二模拟对象的生命更新指令在所述交互式应用场景中加载所述第二模拟对象的更新后的生命值。

[0016] 本申请的第三方面,提供了一种计算机可读存储介质,所述计算机可读存储介质中存储有指令,当其在计算机上运行时,使得计算机执行上述各方面所述的方法。

[0017] 从以上技术方案可以看出,本发明实施例具有以下优点:

[0018] 在本发明实施例中,第一客户端获取第一模拟对象在交互式应用场景中的第一对象状态,当第一对象状态发生状态更新时,第一客户端更新第二模拟对象在交互式应用场景中的生命值,第二模拟对象与第一模拟对象共同处于交互式应用场景中,第一客户端将第二模拟对象的更新后的生命值发送给服务器,服务器用于管理第一客户端和第二客户端分别加载的交互式应用场景,第一客户端接收服务器向第一客户端和第二客户端广播的第二模拟对象的生命更新指令,并按照第二模拟对象的生命更新指令在交互式应用场景中加载第二模拟对象的更新后的生命值。本发明实施例第一客户端会监控第一模拟对象的对象状态,当第一对象状态发生状态更新时,可以为第二模拟对象更新生命值,因此第二模拟对象的对象状态是否更新与第一对象发生了状态更新密切相关,可以实现对第二模拟对象的生命值控制,第一模拟对象的状态可以得到更新,从而保证模拟对象的状态更新进度,第二模拟对象的生命值也因为第一对象状态的状态更新而更新,从而可以保证第二模拟对象所属的用户侧的体验,不需要在交互式应用场景中增加更多的模拟对象,从而降低设计复杂度。

附图说明

[0019] 为了更清楚地说明本发明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发明的一些实施例,对于本领域的技术人员来讲,还可以根据这些附图获得其他的附图。

[0020] 图1为本发明实施例提供的基于模拟对象的生命值控制方法的一种系统应用架构的示意图;

[0021] 图2为本发明实施例提供的一种基于模拟对象的生命值控制方法的流程方框示意图;

[0022] 图3-a为本发明实施例提供的在第一客户端的界面上显示交互式应用场景的示意图;

[0023] 图3-b为本发明实施例提供的游戏场景中基地产生护盾时增加血量的示意图;

[0024] 图4为本发明实施例提供的基于模拟对象的生命值控制方法的一种应用场景示意图;

[0025] 图5-a为本发明实施例提供的一种客户端的组成结构示意图;

[0026] 图5-b为本发明实施例提供的一种生命值更新模块的组成结构示意图;

[0027] 图5-c为本发明实施例提供的另一种生命值更新模块的组成结构示意图;

[0028] 图6为本发明实施例提供的基于模拟对象的生命值控制方法应用于终端的组成结

构示意图。

具体实施方式

[0029] 本发明实施例提供了一种基于模拟对象的生命值控制方法和客户端,用于实现对第二模拟对象的生命值控制,可以保证模拟对象的状态更新进度,不需要在交互式应用场景中增加更多的模拟对象,从而降低设计复杂度。

[0030] 为使得本发明的发明目的、特征、优点能够更加的明显和易懂,下面将结合本发明实施例中的附图,对本发明实施例中的技术方案进行清楚、完整地描述,显然,下面所描述的实施例仅仅是本发明一部分实施例,而非全部实施例。基于本发明中的实施例,本领域的技术人员所获得的所有其他实施例,都属于本发明保护的范围。

[0031] 本发明的说明书和权利要求书及上述附图中的术语“包括”和“具有”以及他们的任何变形,意图在于覆盖不排他的包含,以便包含一系列单元的过程、方法、系统、产品或设备不必限于那些单元,而是可包括没有清楚地列出的或对于这些过程、方法、产品或设备固有的其它单元。

[0032] 以下分别进行详细说明。

[0033] 本发明基于模拟对象的生命值控制方法的一个实施例,具体可以应用于在交互式应用场景中显示的模拟对象的生命值控制场景中。本发明实施例中交互式应用场景具体可以是游戏场景,也可以是应用程序的交互场景。举例说明,本发明实施例提供的基于模拟对象的生命值控制方法可以适用于为游戏角色搭建的场景,也可以适用于在软件应用系统中为用户对象搭建的场景。本发明实施例中所述的交互式应用场景中显示有模拟对象,该模拟对象可以是游戏场景中的游戏角色,也可以是游戏场景中的道具,例如模拟对象可以是二塔,或者高地塔,或者基地等场景道具,此处不做限定。

[0034] 请参阅图1所示,为本发明实施例提供的基于模拟对象的生命值控制方法的一种系统应用架构的示意图。在该系统中可以包括服务器和多个客户端,例如图1中的第一客户端10和第二客户端11、服务器20。在本发明实施例中,服务器和第一客户端、第二客户端之间分别建立有通信连接,该服务器用于管理各个客户端中分别加载的交互式应用场景,例如服务器和第一客户端、第二客户端之间通过无线网络相连接。例如,本发明实施例中服务器可以管理处于相同游戏场景下的两类客户端(即第一客户端和第二客户端),不限的是,本发明实施例中服务器还可以管理更多的客户端,例如服务器和处于相同游戏场景下的第一客户端、第二客户端、第三客户端保持有通信连接,这些客户端和服务器之间的交互式应用场景的显示方式与本发明实施例中两类客户端与服务器的交互过程相类似,可以参照本发明后续实施例中的多个应用场景的说明。

[0035] 如图1所示,第一客户端可以监控第一模拟对象是否发生状态更新,若第一模拟对象发生状态更新时,第一客户端更新第二模拟对象在交互式应用场景中的生命值,第一客户端将第二模拟对象的更新后的生命值通过无线网络发送给服务器,服务器根据该第二模拟对象的更新后的生命值生成第二模拟对象的生命更新指令,然后服务器通过无线网络广播该第二模拟对象的生命更新指令,第一客户端接收服务器向第一客户端和第二客户端广播的第二模拟对象的生命更新指令,并按照第二模拟对象的生命更新指令在交互式应用场景中加载第二模拟对象的更新后的生命值。因此第二模拟对象的对象状态是否更新与第一

对象发生了状态更新密切相关,可以实现对第二模拟对象的生命值控制,第一模拟对象的状态可以得到更新,从而保证模拟对象的状态更新进度,第二模拟对象的生命值也因为第一对象状态的状态更新而更新,从而可以保证第二模拟对象所属的用户侧的体验,不需要在交互式应用场景中增加更多的模拟对象,从而降低设计复杂度。

[0036] 本发明实施例中以第一客户端的交互式应用场景中显示有第一模拟对象和第二模拟对象为例进行说明,该第一模拟对象和第二模拟对象可以属于第一客户端中的用户角色所拥有,不限定的是,本发明实施例第一客户端的交互式应用场景中除了显示第一模拟对象和第二模拟对象之外,还可以显示有第三模拟对象。不限定的是,本发明实施例中第二客户端同样可以显示有交互式应用场景。举例说明,服务器可以通过帧同步命令向第一客户端和第二客户端发送数据,使得第一客户端和第二客户端能够显示相同的交互式应用场景。其中,帧同步是一种前后端数据同步的方式,可应用于对实时性要求很高的网络游戏,帧同步的过程可以包括如下内容:首先,各客户端实时上传操作指令集。然后服务器保存这些操作指令集,并在下一帧将其广播给所有客户端。最后客户端收到指令集后分别按帧序执行指令集中的操作。

[0037] 接下来从第一客户端一侧来说明本发明实施例提供的基于模拟对象的生命值控制方法,请参阅图2所示,本发明实施例提供的基于模拟对象的生命值控制方法,可包括如下步骤:

[0038] 201、第一客户端获取第一模拟对象在交互式应用场景中的第一对象状态。

[0039] 在本发明实施例中,第一模拟对象是指在交互式应用场景中显示的一个模拟对象,第一客户端首先监控该第一模拟对象的对象状态,为便于描述,将第一模拟对象的对象状态定义为“第一对象状态”。在交互式应用场景中每个模拟对象都可以有多种对象状态,并且这些对象状态是可以进行状态更新的。其中,模拟对象的状态更新可以由模拟对象自发进行,可以受到外界触发条件的触发再发生对象状态的更新。举例说明如下,以交互式应用场景为游戏场景为例,第一模拟对象可以是游戏场景中的第一道具,该第一道具可以受到攻击方角色的进攻之后发生状态更新,例如从在游戏场景中存在更新为被摧毁。

[0040] 202、当第一对象状态发生状态更新时,第一客户端更新第二模拟对象在交互式应用场景中的生命值,第二模拟对象与第一模拟对象共同处于交互式应用场景中。

[0041] 在本发明实施例中,第一客户端确定第一模拟对象需要进行实时监控时,第一客户端可以根据第一对象状态判断第一模拟对象是否发生了状态更新,在第一对象状态发生状态更新时,第一客户端更新第二模拟对象在交互式应用场景中的生命值,其中,第二模拟对象与第一模拟对象共同处于交互式应用场景中。例如,以交互式应用场景为游戏场景为例,第一模拟对象可以是游戏场景中的第一道具,第二模拟对象可以是游戏场景中的第二道具,则在游戏场景中可以同时显示第一道具和第二道具,其中,第一道具和第二道具都可以发生状态更新,例如,该第一道具可以受到攻击方角色的进攻之后发生状态更新,第二道具可以受到攻击方角色的进攻之后发生状态更新。

[0042] 在本发明实施例中,第一模拟对象和第二模拟对象在交互式应用场景中共同存在,这两个模拟对象的对象状态存在关联关系,即第二模拟对象的生命值会受到第一模拟对象的对象状态是否更新的影响,在第一模拟对象的对象状态没有发生更新时,第二模拟对象的生命值保持不变,在第一模拟对象的对象状态发生更新时,第一客户端更新第二模

拟对象的生命值。其中,生命值可以认为是模拟对象的生命周期,即生命值用于确定模拟对象是否显示在交互式应用场景中,更新了第二模拟对象的生命值,则第二模拟对象在交互式应用场景中的显示时长也会发生变化。

[0043] 需要说明的是,在本发明实施例中,第二模拟对象的生命值是可以依据第一模拟对象的第一对象状态是否发生状态更新进行动态调整,第一模拟对象的第一对象状态发生更新,使得交互式应用场景的进展得以进行,对于进攻方角色而言,能够体验到交互式应用场景的不断更新模拟对象的状态,例如第一模拟对象被进攻方角色攻击之后从显示状态转换为被摧毁状态,对于进攻方角色而言其用户体验可以得到保证,第二模拟对象的生命值又可以跟随第一模拟对象的对象状态发生更新进行动态更新,使得第二模拟对象的持有者(即防守方角色)仍可以按照更新后的生命值来显示第二模拟对象,因此防守方角色的用户体验也可以得到保证,因此本发明实施例解决了对战过程中一方被快速推进而导致防守方角色的不良体验,同时也没有让节奏变得更加拖沓,也可以保证进攻方角色的用户体验。

[0044] 在本发明的一些实施例中,步骤202当第一对象状态发生状态更新时,第一客户端更新第二模拟对象在交互式应用场景中的生命值,包括:

[0045] A1、当第一对象状态发生状态更新时,第一客户端获取第三模拟对象在交互式应用场景中的第二对象状态,其中,在交互式应用场景中第三模拟对象分别与第一模拟对象、第二模拟对象属于相邻对象;

[0046] A2、当第二对象状态发生状态更新时,第一客户端更新第二模拟对象在交互式应用场景中的生命值。

[0047] 其中,若交互式应用场景中同时存在三种模拟对象:第一模拟对象、第二模拟对象、第三模拟对象,例如请参阅图3-a所示,为本发明实施例提供的在第一客户端的界面上显示交互式应用场景的示意图。第一客户端可以是移动终端,图3-a中显示的是移动终端的用户界面(User Interface,UI),在移动终端的交互式应用场景中显示有三个模拟对象,分别为:模拟对象1、模拟对象2和模拟对象3。在交互式应用场景中模拟对象3分别与模拟对象1、模拟对象2属于相邻对象,图3-a中的箭头可以表示进攻方角色的进攻方向,若模拟对象3分别与模拟对象1、模拟对象2相邻,则进攻方角色可以首先攻击模拟对象1,当模拟对象1的状态更新为被摧毁时,进攻方角色继续攻击模拟对象,3、模拟对象2。第一客户端获取模拟对象3在交互式应用场景中的第二对象状态,第一客户端继续监控模拟对象3的第二对象状态,当第二对象状态发生状态更新时,第一客户端更新第二模拟对象在交互式应用场景中的生命值,即第二模拟对象的生命值可以跟随第一模拟对象的对象状态发生更新、第二模拟对象的对象状态发生更新进行动态更新,使得第二模拟对象的持有者仍可以按照更新后的生命值来显示第二模拟对象。

[0048] 进一步的,在本发明的一些实施例中,前述步骤202当第一对象状态发生状态更新时,第一客户端更新第二模拟对象在交互式应用场景中的生命值,除了包括步骤A1至步骤A2之外,步骤202还可以包括如下步骤:

[0049] A3、当第一对象状态发生状态更新时,第一客户端为第三模拟对象增加一个计时器并触发计时器开始计时。

[0050] 其中,当第一对象状态发生状态更新时,第一客户端除了执行前述的步骤A1,还可以执行步骤A3,即第一客户端在监控第三模拟对象的第二对象状态时还启用计时器,该计

时器用于对第三模拟对象在交互式应用场景中的显示进行计时,例如通过计时器可以确定第三模拟对象在交互式应用场景中已经存在的时长。

[0051] 在前述执行步骤A3的实现场景下,步骤A2当第二对象状态发生状态更新时,第一客户端更新第二模拟对象在交互式应用场景中的生命值,包括:

[0052] A21、第一客户端判断第三模拟对象的第二对象状态是否在计时器的计时周期内发生状态更新;

[0053] A22、当第二对象状态在计时器的计时周期内发生状态更新时,第一客户端更新第二模拟对象在交互式应用场景中的生命值。

[0054] 其中,第一客户端可以判断在该计时器销毁之前第三模拟对象的第二对象状态是否发生状态更新,若第二对象状态在计时器的计时周期内发生状态更新,则说明第一模拟对象和第三模拟对象的对象状态被连续更新,举例说明,若计时器的计时周期取值很小,则说明第一模拟对象和第三模拟对象的对象状态在很短的时间内就被连续更新,例如第一模拟对象和第三模拟对象在很短的时间内被连续摧毁,此时第一客户端更新第二模拟对象在交互式应用场景中的生命值,第二模拟对象的生命值可以跟随第一模拟对象的对象状态发生更新、第二模拟对象的对象状态发生更新进行动态更新,使得第二模拟对象的持有者仍可以按照更新后的生命值来显示第二模拟对象。

[0055] 在本发明的一些实施例中,步骤202当第一对象状态发生状态更新时,第一客户端更新第二模拟对象在交互式应用场景中的生命值,包括:

[0056] B1、当第一对象状态更新为被摧毁时,第一客户端按照生命延长策略增加第二模拟对象在交互式应用场景中的生命值。

[0057] 其中,模拟对象的对象状态更新可有多种实现方式,例如模拟对象在交互式应用场景中共有两种以上的对象状态:状态1和状态2,举例说明,状态1可以为显示状态,状态2可以为被摧毁状态。第一对象状态更新可以包括第一模拟对象被摧毁,即第一模拟对象从交互式应用场景中从存在状态更新为被摧毁状态,第一客户端可以按照生命延长策略增加第二模拟对象在交互式应用场景中的生命值,通过延长第二模拟对象的生命值可以使得第二模拟对象在交互式应用场景中的显示周期延长。

[0058] 在本发明的另一些实施例中,步骤202中的第一客户端更新第二模拟对象在交互式应用场景中的生命值,包括:

[0059] C1、第一客户端为第二模拟对象增加持续预置时长的防御对象,防御对象用于保护第二模拟对象。

[0060] 其中,更新第二模拟对象在交互式应用场景中的生命值可以有多种实现方式,例如为第二模拟对象设置防御对象,该防御对象可以用于保护第二模拟对象。具体的,以交互式应用场景具体为游戏场景为例,第一客户端还可以为第二模拟对象设置一个防御对象,例如该防御对象可以为3000点血量的护盾,通过防御对象来保护第二模拟对象,从而使得第二模拟对象的生命值得到更新。第一客户端设置的防御对象还可以包括持续预置时长,例如可以持续15秒,即该防御对象被增加到第二模拟对象上之后持续15秒,在15秒后该防御对象从交互式应用场景中消失,不再为第二模拟对象提供保护功能。

[0061] 在本发明的一些实施例中,若第一模拟对象被第一触发条件触发状态更新时,步骤202第一客户端更新第二模拟对象在交互式应用场景中的生命值之后,本发明实施例提

供的基于模拟对象的生命值控制方法,还可以包括如下步骤:

[0062] D1、在第二模拟对象的生命值存续期间,当第一模拟对象被第二触发条件触发状态更新时,第一客户端禁止再次更新第二模拟对象在交互式应用场景中的生命值。

[0063] 其中,为了避免第二模拟对象的生命值被重复更新,步骤202中第一客户端在监控第一模拟对象的第一对象状态时,可以监控第一模拟对象的触发条件,即第一模拟对象通过第一触发条件被触发状态更新。其中触发条件可以是第一模拟对象被哪个进攻方角色触发状态更新,或者触发条件可以是第一模拟对象所属的战斗序列被进攻方角色触发状态更新。在步骤202中第二模拟对象的生命值已经被更新之后,在第二模拟对象的生命值存续期间,当第一模拟对象被第二触发条件触发状态更新时,说明第一模拟对象又被不同于第一触发条件的另一个触发条件(即第二触发条件)触发了状态更新,即满足了步骤202中的“当第一对象状态发生状态更新时”,为了避免第二模拟对象的生命值被重复更新,第一客户端禁止再次更新第二模拟对象在交互式应用场景中的生命值。通过禁止对第二模拟对象重复更新生命值,使得交互式应用场景的进展节奏依然紧凑,进攻方角色的用户体验也可以得到保证。因此本发明实施例解决了对战过程中一方被快速推进而导致防守方角色的不良体验,同时也没有让节奏变得更加拖沓,也可以保证进攻方角色的用户体验。

[0064] 203、第一客户端将第二模拟对象的更新后的生命值发送给服务器,服务器用于管理第一客户端和第二客户端分别加载的交互式应用场景。

[0065] 在本发明实施例中,当第一对象状态发生状态更新时,第一客户端更新第二模拟对象在交互式应用场景中的生命值,第一客户端通过无线网络可以向服务器发送第二模拟对象的更新后的生命值,则服务器可以接收到第二模拟对象的更新后的生命值,由于服务器用于管理第一客户端和第二客户端分别加载的交互式应用场景,因此服务器可以根据第二模拟对象的更新后的生命值生成第二模拟对象的生命更新指令。

[0066] 204、第一客户端接收服务器向第一客户端和第二客户端广播的第二模拟对象的生命更新指令,并按照第二模拟对象的生命更新指令在交互式应用场景中加载第二模拟对象的更新后的生命值。

[0067] 在本发明实施例中,服务器向第一客户端和第二客户端广播的第二模拟对象的生命更新指令之后,第一客户端和第二客户端都可以接收到第二模拟对象的生命更新指令,则第一客户端和第二客户端都可以按照第二模拟对象的生命更新指令在交互式应用场景中加载第二模拟对象的更新后的生命值,从而在第一客户端和第二客户端上显示出了生命值更新后的第二模拟对象,因此可以实现第二模拟对象根据第一模拟对象的对象状态进行实时的生命值更新。

[0068] 在本发明的一些实施例中,步骤204第一客户端接收服务器向第一客户端和第二客户端广播的第二模拟对象的生命更新指令,包括:

[0069] E1、第一客户端接收服务器向第一客户端和第二客户端广播的帧同步命令;

[0070] E2、第一客户端解析帧同步命令得到生命更新指令。

[0071] 其中,服务器可以通过帧同步命令向第一客户端和第二客户端发送数据,使得第一客户端和第二客户端能够显示相同的交互式应用场景。帧同步是一种前后端数据同步的方式,可应用于对实时性要求很高的网络游戏,帧同步的过程可以包括如下内容:首先,各客户端实时上传操作指令集。然后服务器保存这些操作指令集,并在下一帧将其广播给所

有客户端。最后客户端收到指令集获取到服务器广播的生命更新指令,然后各个客户端分别按帧序执行指令集中的操作。

[0072] 需要说明的是,在本发明实施例中,以交互式应用场景具体为游戏场景为例。在游戏框架服务端的逻辑部分设计中,可以根据一致的初始化参数GameStartDocument,进行一致的初始化构造,如场景,对象,逻辑模块等。如此游戏会有一致的开始。对于游戏数据类型,包括int,bool,float,string,各种游戏数据的运算需要检查一致性。

[0073] 在本发明的一些实施例中,以交互式应用场景具体为游戏场景为例,在游戏场景下,第一模拟对象具体为第一游戏角色,第二模拟对象具体为第二游戏角色;第一客户端具体为第一游戏角色。步骤201第一客户端获取第一模拟对象在交互式应用场景中的第一对象状态,包括:

[0074] 第一游戏客户端获取第一游戏角色在游戏场景中的第一角色状态。

[0075] 步骤202、第一客户端更新第二模拟对象在交互式应用场景中的生命值,包括:

[0076] 第一游戏客户端更新第二游戏角色在游戏场景中的生命值。

[0077] 步骤204、按照第二模拟对象的生命更新指令在交互式应用场景中加载第二模拟对象的更新后的生命值,包括:

[0078] 第一游戏客户端按照第二游戏角色的生命更新指令在游戏场景中加载第二游戏角色的更新后的生命值。

[0079] 在游戏场景下,第一游戏客户端首先获取第一游戏角色在游戏场景中的第一角色状态。当所述第一角色状态发生状态更新时,所述第一游戏客户端更新第二游戏角色在所述游戏场景中的生命值,所述第二游戏角色与所述第一游戏角色共同处于所述游戏场景中。所述第一游戏客户端将所述第二游戏角色的更新后的生命值发送给服务器,所述服务器用于管理所述第一游戏客户端和第二游戏客户端分别加载的所述游戏场景。所述第一游戏客户端接收所述服务器向所述第一游戏客户端和所述第二游戏客户端广播的所述第二游戏角色的生命更新指令,并按照所述第二游戏角色的生命更新指令在所述游戏场景中加载所述第二游戏角色的更新后的生命值。本发明实施例第一游戏客户端会监控第一游戏角色的角色状态,当第一角色状态发生状态更新时,可以为第二游戏角色更新生命值,因此第二游戏角色的角色状态是否更新与第一对象发生了状态更新密切相关,可以实现对第二游戏角色的生命值控制,第一游戏角色的状态可以得到更新,从而保证游戏角色的状态更新进度,第二游戏角色的生命值也因为第一角色状态的状态更新而更新,从而可以保证第二游戏角色所属的用户侧的体验,不需要在游戏场景中增加更多的游戏角色,从而降低设计复杂度。

[0080] 通过以上实施例对本发明实施例的描述可知,第一客户端获取第一模拟对象在交互式应用场景中的第一对象状态,当第一对象状态发生状态更新时,第一客户端更新第二模拟对象在交互式应用场景中的生命值,第二模拟对象与第一模拟对象共同处于交互式应用场景中,第一客户端将第二模拟对象的更新后的生命值发送给服务器,服务器用于管理第一客户端和第二客户端分别加载的交互式应用场景,第一客户端接收服务器向第一客户端和第二客户端广播的第二模拟对象的生命更新指令,并按照第二模拟对象的生命更新指令在交互式应用场景中加载第二模拟对象的更新后的生命值。本发明实施例第一客户端会监控第一模拟对象的对象状态,当第一对象状态发生状态更新时,可以为第二模拟对象更

新生命值,因此第二模拟对象的对象状态是否更新与第一对象发生了状态更新密切相关,可以实现对第二模拟对象的生命值控制,第一模拟对象的状态可以得到更新,从而保证模拟对象的状态更新进度,第二模拟对象的生命值也因为第一对象状态的状态更新而更新,从而可以保证第二模拟对象所属的用户侧的体验,不需要在交互式应用场景中增加更多的模拟对象,从而降低设计复杂度。

[0081] 为便于更好的理解和实施本发明实施例的上述方案,下面举例相应的应用场景来进行具体说明。以MOBA游戏场景的设计为例,玩家可以在用户界面的操作层上操作玩家角色向对战方进行进攻,其中,操作层是玩家使用的角色所处的游戏场景层,需要说明的是,游戏场景中角色无法直接走动穿过的场景物件即是阻挡层,对于阻挡层可以包括墙体、机关等,这些构成前述实施例中的模拟对象,这些模拟对象可以在进攻方角色的对战过程中从显示状态转换为被摧毁状态。

[0082] 在手机的MOBA游戏中,因为游戏的快节奏,有时会出现一方在取得一定优势后,快速连续推塔的情况,而直接增加机关的血量或者防御值又会出现导致推线变困难,节奏变慢的情况,这是个玩家体验中的痛点,也是本发明实施例中需要寻找的平衡点,按照本发明实施例的解决方式,既可以减缓一方连续推塔造成的感受不好问题,又可以避免节奏变慢。本发明实施例可以在机关等道具上配备一系列的触发器,可以判断不同时间内,不同机关的摧毁状态,配置比如在N秒内连续被击爆了几座机关则会为下一个机关增加一个持续性的护盾,该护盾可抵御进攻方角色的伤害。如图3-b所示,为本发明实施例提供的游戏场景中基地产生护盾时增加血量的示意图。前述实施例中的第二游戏角色具体可以为图3-b中所示的是游戏场景中的基地,在游戏场景中可以为该基地产生护盾,图3-b中的血条即为增加的血量,通过这种方式,本发明实施例很好地达到了预期中的效果,解决了对战过程中,一方被快速推到多做机关的不良体验,同时也没有让节奏变得更加拖沓的问题。需要说明的是,图3-b中所示的游戏场景中的基地和增加的血量只是用于说明基地生命值的可更新,对于基地的形状仅做示意说明,该基地所包括的线条和连接关系不做限定。

[0083] 如图4所示,为本发明实施例提供的基于模拟对象的生命值控制方法的一种应用场景示意图,主要包括如下过程:

[0084] S01、假定一条线上的2塔,高地塔和基地分别为ABC三座塔。

[0085] 假定一条线上的2塔,高地塔和基地分别为ABC三座塔,在游戏场景下共有三条线,两个阵营,即一共有6条互补冲突ABC塔线,其中C都是基地,AB分别为每一条线上的2塔和高地塔,要想触发基地护盾保护,只能由一条线上的AB塔触发。

[0086] S02、系统检测A塔是否被击毁。

[0087] S03、若不是,继续执行步骤S02中的检测过程。

[0088] S04、若是,客户端为B塔增加一个计时器。

[0089] 客户端实施监控每条线上的A塔是不是被摧毁了,在A塔被击毁瞬间即会为该条线上的B塔增加一个计时器。

[0090] S05、计时器检测在X时间内,B塔有没有被摧毁。

[0091] 其中,客户端检测B塔在X时间内有没有被摧毁,X的取值可以根据游戏场景配置。

[0092] S06、若是,则为基地加一层护盾,例如持续M秒N血量的护盾,然后执行步骤S08和S11。

[0093] 如果B塔在X时间内被摧毁,则为C增加一层持续M秒N血量的护盾,M和N的取值可以根据游戏场景配置。

[0094] 举例说明,2塔和高地塔在30秒内被连续推掉时,水晶会增加一个4500点血量的护盾,持续15秒。同样道理,可以配置任意ABC三塔的护盾保护关系,即A塔和B塔在一定时间X内连续击毁后,给C塔增加一个持续M秒N血量的护盾。

[0095] S07、若否,计时器销毁,后面B塔被爆与否,C塔均不会增加护盾。

[0096] S08、检测在护盾持续期间有没有其他路的2塔、高地塔被击毁且符合加护盾条件。

[0097] S09、若否,继续检测。

[0098] S10、若是,效果禁止叠加,即不会再次触发保护。

[0099] 其中,客户端在此护盾持续期间,检测本方其它线上的2塔A有没有被摧毁,以及是否同样触发了护盾条件,如果触发则会禁止这个护盾再次添加到C上。

[0100] S11、发送帧同步命令。

[0101] 其中,客户端发送帧同步命令给服务器,服务器广播给游戏中所有客户端。

[0102] S12、服务器中转给每个客户端。

[0103] S13、每个客户端分别做一致性规则检查。

[0104] 每个客户端收到帧同步命令后,进行一致性规则检查,检查通过后,就会在所有客户端上同步C加护盾的命令。

[0105] S14、同步每个客户端。

[0106] 通过前述的举例说明,本发明实施例解决了在游戏场景的一场比赛中,连续爆三塔造成猝死的概率。

[0107] 需要说明的是,对于前述的各方法实施例,为了简单描述,故将其都表述为一系列的动作组合,但是本领域技术人员应该知悉,本发明并不受所描述的动作顺序的限制,因为依据本发明,某些步骤可以采用其他顺序或者同时进行。其次,本领域技术人员也应该知悉,说明书中所描述的实施例均属于优选实施例,所涉及的动作和模块并不一定是本发明所必须的。

[0108] 为便于更好的实施本发明实施例的上述方案,下面还提供用于实施上述方案的相关装置。

[0109] 请参阅图5-a所示,本发明实施例提供了一种客户端,该客户端具体为第一客户端500,可以包括:状态获取模块501、生命值更新模块502、发送模块503、显示模块504,其中,

[0110] 状态获取模块501,用于获取第一模拟对象在交互式应用场景中的第一对象状态;

[0111] 生命值更新模块502,用于当所述第一对象状态发生状态更新时,更新第二模拟对象在所述交互式应用场景中的生命值,所述第二模拟对象与所述第一模拟对象共同处于所述交互式应用场景中;

[0112] 发送模块503,用于将所述第二模拟对象的更新后的生命值发送给服务器,所述服务器用于管理所述第一客户端和第二客户端分别加载的所述交互式应用场景;

[0113] 显示模块504,用于接收所述服务器向所述第一客户端和所述第二客户端广播的所述第二模拟对象的生命更新指令,并按照所述第二模拟对象的生命更新指令在所述交互式应用场景中加载所述第二模拟对象的更新后的生命值。

[0114] 在本发明的一些实施例中,请参阅图5-b所示,所述生命值更新模块502,包括:

[0115] 状态获取子模块5021,用于当所述第一对象状态发生状态更新时,获取第三模拟对象在所述交互式应用场景中的第二对象状态,其中,在所述交互式应用场景中所述第三模拟对象分别与所述第一模拟对象、所述第二模拟对象属于相邻对象;

[0116] 生命值更新子模块5022,用于当所述第二对象状态发生状态更新时,更新所述第二模拟对象在所述交互式应用场景中的生命值。

[0117] 在本发明的一些实施例中,请参阅图5-c所示,相对于图5-b所示,所述生命值更新模块502,还包括:

[0118] 计时子模块5023,用于当所述第一对象状态发生状态更新时,为所述第三模拟对象增加一个计时器并触发计时器开始计时;

[0119] 所述生命值更新子模块5022,具体用于判断所述第三模拟对象的第二对象状态是否在所述计时器的计时周期内发生状态更新;当所述第二对象状态在所述计时器的计时周期内发生状态更新时,更新所述第二模拟对象在所述交互式应用场景中的生命值。

[0120] 在本发明的一些实施例中,所述生命值更新模块502,具体用于为所述第二模拟对象增加持续预置时长的防御对象,所述防御对象用于保护所述第二模拟对象。

[0121] 在本发明的一些实施例中,若所述第一模拟对象被第一触发条件触发状态更新时,所述生命值更新模块502,还用于在所述第二模拟对象的生命值存续期间,当所述第一模拟对象被第二触发条件触发状态更新时,禁止再次更新所述第二模拟对象在所述交互式应用场景中的生命值。

[0122] 在本发明的一些实施例中,所述显示模块504,具体用于接收所述服务器向所述第一客户端和所述第二客户端广播的帧同步命令;解析所述帧同步命令得到所述生命更新指令。

[0123] 在本发明的一些实施例中,所述生命值更新模块502,具体用于当所述第一对象状态更新为被摧毁时,按照生命延长策略增加所述第二模拟对象在所述交互式应用场景中的生命值。

[0124] 在本发明的一些实施例中,所述第一模拟对象具体为第一游戏角色,第二模拟对象具体为第二游戏角色;

[0125] 所述状态获取模块501,具体用于获取第一游戏角色在游戏场景中的第一角色状态;

[0126] 所述生命值更新模块502,具体用于更新所述第二游戏角色在游戏场景中的生命值;

[0127] 所述显示模块504,具体用于按照所述第二游戏角色的生命更新指令在所述游戏场景中加载所述第二游戏角色的更新后的生命值。

[0128] 通过以上实施例对本发明实施例的描述可知,第一客户端获取第一模拟对象在交互式应用场景中的第一对象状态,当第一对象状态发生状态更新时,第一客户端更新第二模拟对象在交互式应用场景中的生命值,第二模拟对象与第一模拟对象共同处于交互式应用场景中,第一客户端将第二模拟对象的更新后的生命值发送给服务器,服务器用于管理第一客户端和第二客户端分别加载的交互式应用场景,第一客户端接收服务器向第一客户端和第二客户端广播的第二模拟对象的生命更新指令,并按照第二模拟对象的生命更新指令在交互式应用场景中加载第二模拟对象的更新后的生命值。本发明实施例第一客户端会

监控第一模拟对象的对象状态,当第一对象状态发生状态更新时,可以为第二模拟对象更新生命值,因此第二模拟对象的对象状态是否更新与第一对象发生了状态更新密切相关,可以实现对第二模拟对象的生命值控制,第一模拟对象的状态可以得到更新,从而保证模拟对象的状态更新进度,第二模拟对象的生命值也因为第一对象状态的状态更新而更新,从而可以保证第二模拟对象所属的用户侧的体验,不需要在交互式应用场景中增加更多的模拟对象,从而降低设计复杂度。

[0129] 本发明实施例还提供了另一种终端,如图6所示,为了便于说明,仅示出了与本发明实施例相关的部分,具体技术细节未揭示的,请参照本发明实施例方法部分。该终端可以为包括手机、平板电脑、PDA(Personal Digital Assistant,个人数字助理)、POS(Point of Sales,销售终端)、车载电脑等任意终端设备,以终端为手机为例:

[0130] 图6示出的是与本发明实施例提供的终端相关的手机的部分结构的框图。参考图6,手机包括:射频(Radio Frequency,RF)电路1010、存储器1020、输入单元1030、显示单元1040、传感器1050、音频电路1060、无线保真(wireless fidelity,WiFi)模块1070、处理器1080、以及电源1090等部件。本领域技术人员可以理解,图6中示出的手机结构并不构成对手机的限定,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。

[0131] 下面结合图6对手机的各个构成部件进行具体的介绍:

[0132] RF电路1010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,给处理器1080处理;另外,将设计上的数据发送给基站。通常,RF电路1010包括但不限于天线、至少一个放大器、收发信机、耦合器、低噪声放大器(Low Noise Amplifier,LNA)、双工器等。此外,RF电路1010还可以通过无线通信与网络和其他设备通信。上述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(Global System of Mobile communication,GSM)、通用分组无线服务(General Packet Radio Service,GPRS)、码分多址(Code Division Multiple Access,CDMA)、宽带码分多址(Wideband Code Division Multiple Access,WCDMA)、长期演进(Long Term Evolution,LTE)、电子邮件、短消息服务(Short Messaging Service,SMS)等。

[0133] 存储器1020可用于存储软件程序以及模块,处理器1080通过运行存储在存储器1020的软件程序以及模块,从而执行手机的各种功能应用以及数据处理。存储器1020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手机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1020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。

[0134] 输入单元1030可用于接收输入的数字或字符信息,以及产生与手机的用户设置以及功能控制有关的键信号输入。具体地,输入单元1030可包括触控面板1031以及其他输入设备1032。触控面板1031,也称为触摸屏,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控面板1031上或在触控面板1031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,触控面板1031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1080,并能接收处理器1080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控面板1031。除了触控面板1031,输入单元1030还可以包括其他输入设备1032。具体地,其他输入设备1032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。

[0135] 显示单元1040可用于显示由用户输入的信息或提供给用户的信息以及手机的各种菜单。显示单元1040可包括显示面板1041,可选的,可以采用液晶显示器(Liquid Crystal Display,LCD)、有机发光二极管(Organic Light-Emitting Diode,OLED)等形式来配置显示面板1041。进一步的,触控面板1031可覆盖显示面板1041,当触控面板1031检测到在其上或附近的触摸操作后,传送给处理器1080以确定触摸事件的类型,随后处理器1080根据触摸事件的类型在显示面板1041上提供相应的视觉输出。虽然在图6中,触控面板1031与显示面板1041是作为两个独立的部件来实现手机的输入和输入功能,但是在某些实施例中,可以将触控面板1031与显示面板1041集成而实现手机的输入和输出功能。

[0136] 手机还可包括至少一种传感器1050,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1041的亮度,接近传感器可在手机移动到耳边时,关闭显示面板1041和/或背光。作为运动传感器的一种,加速计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;至于手机还可配置的陀螺仪、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。

[0137] 音频电路1060、扬声器1061,传声器1062可提供用户与手机之间的音频接口。音频电路1060可将接收到的音频数据转换后的电信号,传输到扬声器1061,由扬声器1061转换为声音信号输出;另一方面,传声器1062将收集的声音信号转换为电信号,由音频电路1060接收后转换为音频数据,再将音频数据输出处理器1080处理后,经RF电路1010以发送给比如另一手机,或者将音频数据输出至存储器1020以便进一步处理。

[0138] WiFi属于短距离无线传输技术,手机通过WiFi模块1070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1070,但是可以理解的是,其并不属于手机的必须构成,完全可以根据需要在不改变发明的本质的范围内而省略。

[0139] 处理器1080是手机的控制中心,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1020内的软件程序和/或模块,以及调用存储在存储器1020内的数据,执行手机的各种功能和处理数据,从而对手机进行整体监控。可选的,处理器1080可包括一个或多个处理单元;优选的,处理器1080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1080中。

[0140] 手机还包括给各个部件供电的电源1090(比如电池),优选的,电源可以通过电源管理系统与处理器1080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。

[0141] 尽管未示出,手机还可以包括摄像头、蓝牙模块等,在此不再赘述。

[0142] 在本发明实施例中,该终端所包括的处理器1080还具有控制执行以上由终端执行的基于模拟对象的生命值控制方法流程。

[0143] 另外需说明的是,以上所描述的装置实施例仅仅是示意性的,其中所述作为分离部件说明的单元可以是或者也可以不是物理上分开的,作为单元显示的部件可以是或者也可以不是物理单元,即可以位于一个地方,或者也可以分布到多个网络单元上。可以根据实际的需要选择其中的部分或者全部模块来实现本实施例方案的目的。另外,本发明提供的装置实施例附图中,模块之间的连接关系表示它们之间具有通信连接,具体可以实现为一条或多条通信总线或信号线。本领域普通技术人员在不付出创造性劳动的情况下,即可以理解并实施。

[0144] 通过以上的实施方式的描述,所属领域的技术人员可以清楚地了解到本发明可借助软件加必需的通用硬件的方式来实现,当然也可以通过专用硬件包括专用集成电路、专用CPU、专用存储器、专用元器件等来实现。一般情况下,凡由计算机程序完成的功能都可以很容易地用相应的硬件来实现,而且,用来实现同一功能的具体硬件结构也可以是多种多样的,例如模拟电路、数字电路或专用电路等。但是,对本发明而言更多情况下软件程序实现是更佳实施方式。基于这样的理解,本发明的技术方案本质上或者说对现有技术做出贡献的部分可以以软件产品的形式体现出来,该计算机软件产品存储在可读取的存储介质中,如计算机的软盘、U盘、移动硬盘、只读存储器(ROM,Read-Only Memory)、随机存取存储器(RAM,Random Access Memory)、磁碟或者光盘等,包括若干指令用以使得一台计算机设备(可以是个人计算机,服务器,或者网络设备等)执行本发明各个实施例所述的方法。

[0145] 综上所述,以上实施例仅用以说明本发明的技术方案,而非对其限制;尽管参照上述实施例对本发明进行了详细的说明,本领域的普通技术人员应当理解:其依然可以对上述各实施例所记载的技术方案进行修改,或者对其中部分技术特征进行等同替换;而这些修改或者替换,并不使相应技术方案的本质脱离本发明各实施例技术方案的精神和范围。

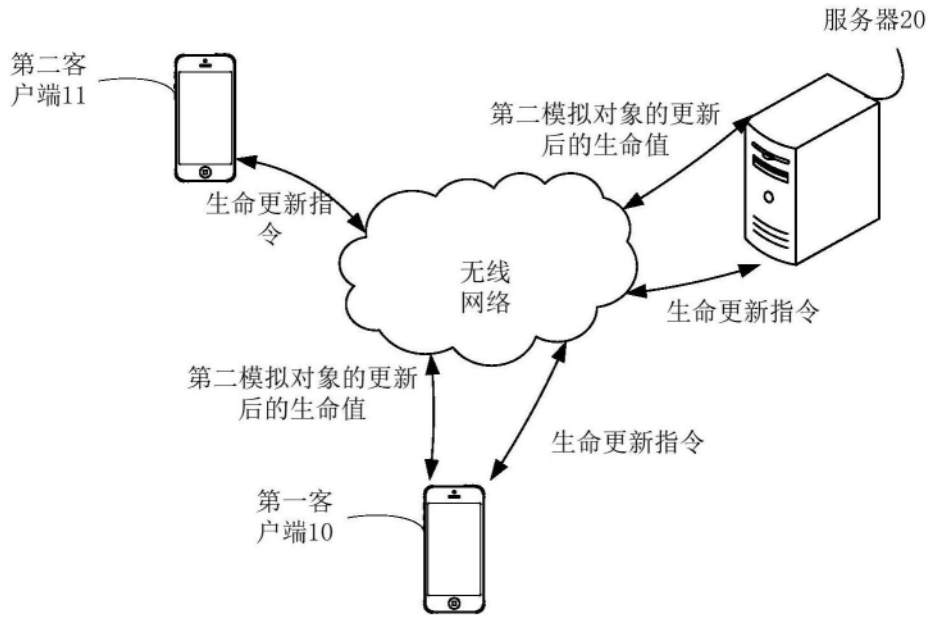


图1

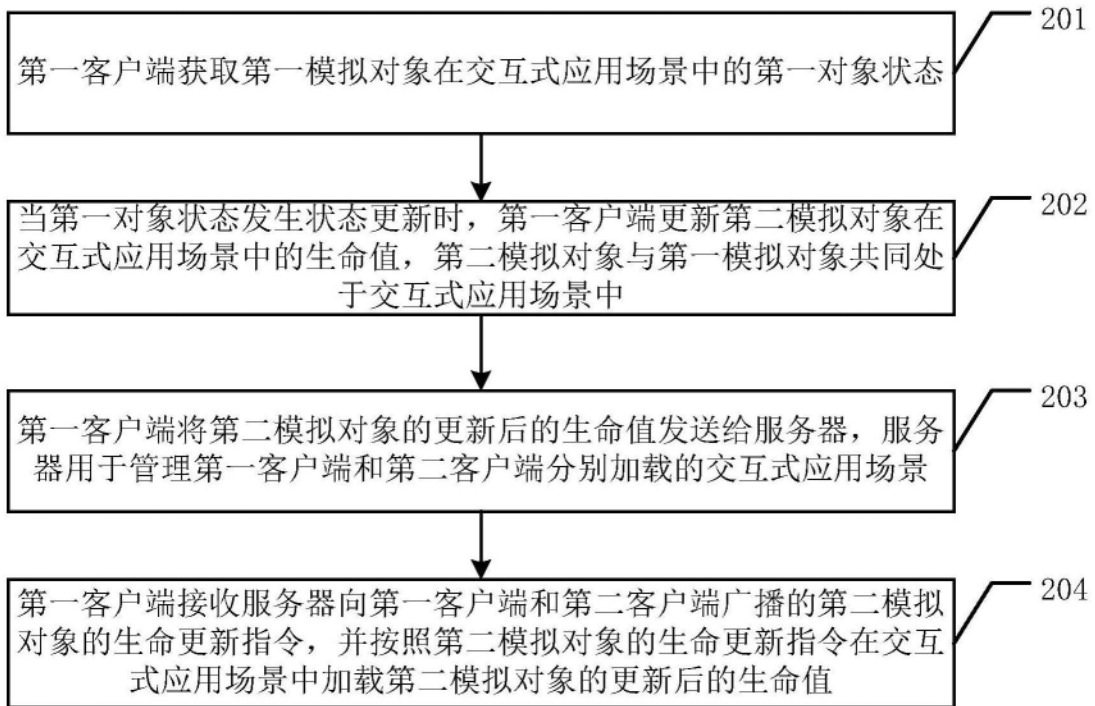


图2

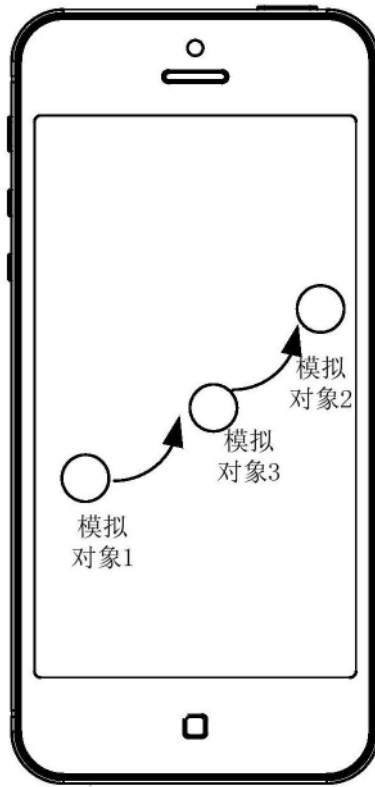


图3-a

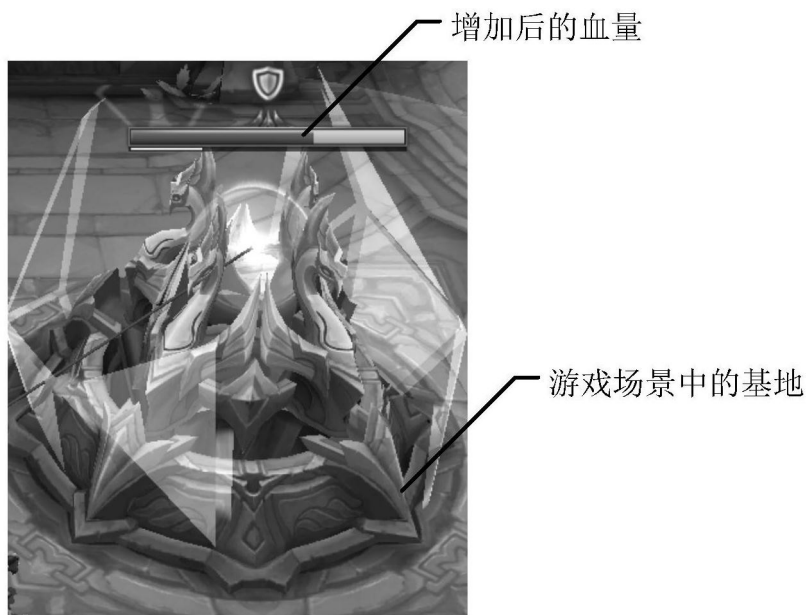


图3-b

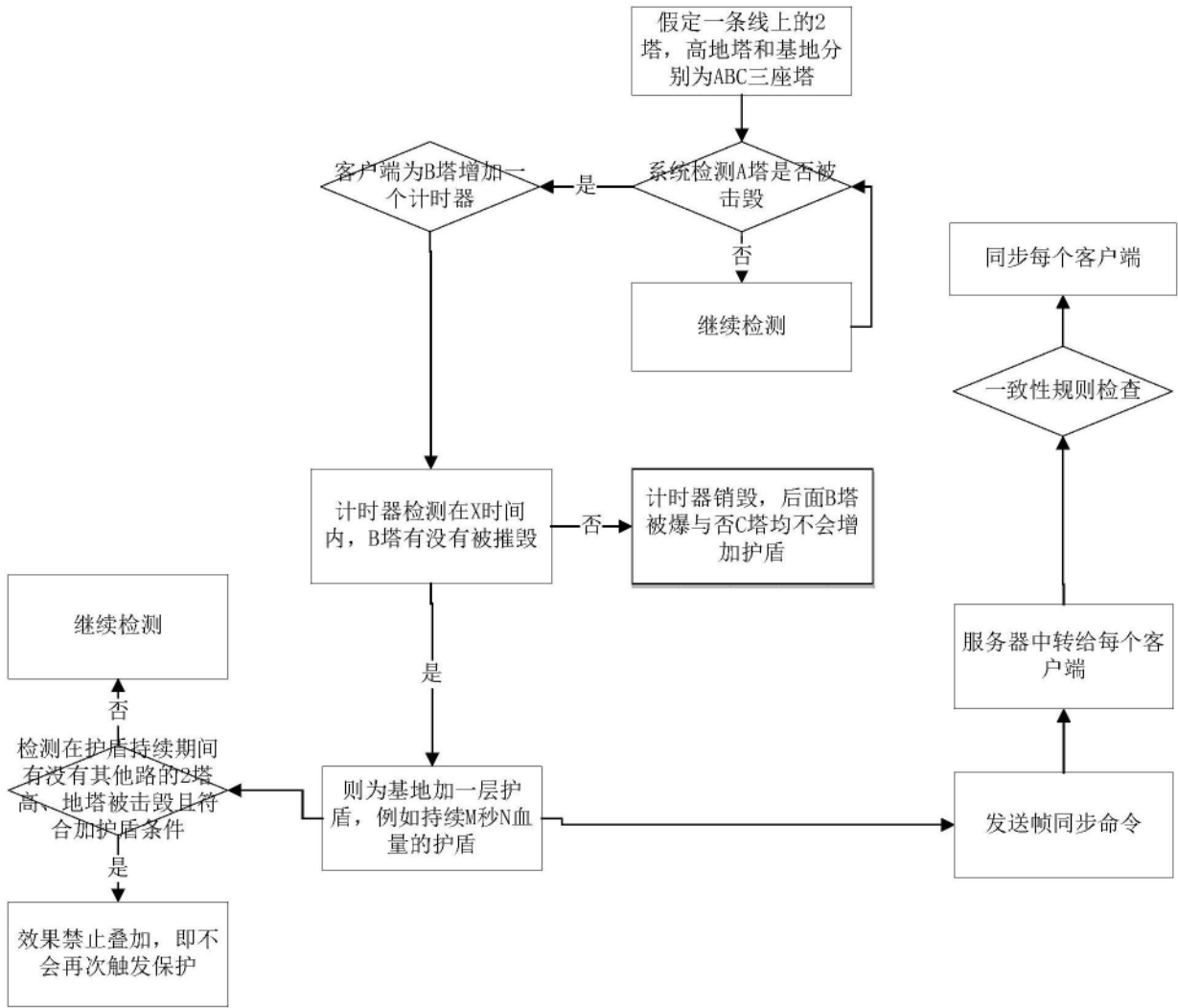


图4

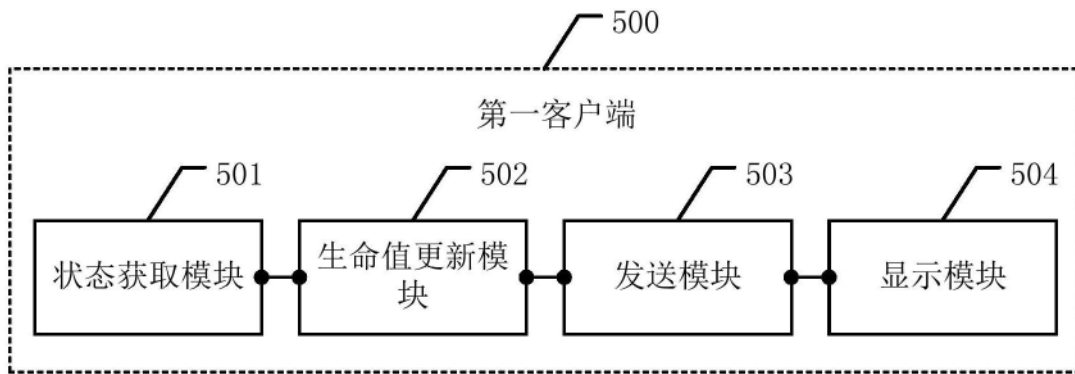


图5-a

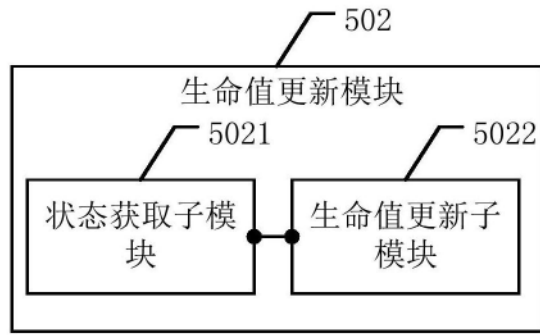


图5-b

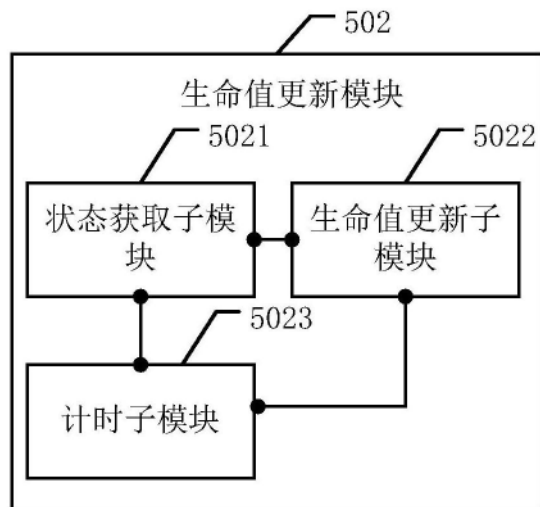


图5-c

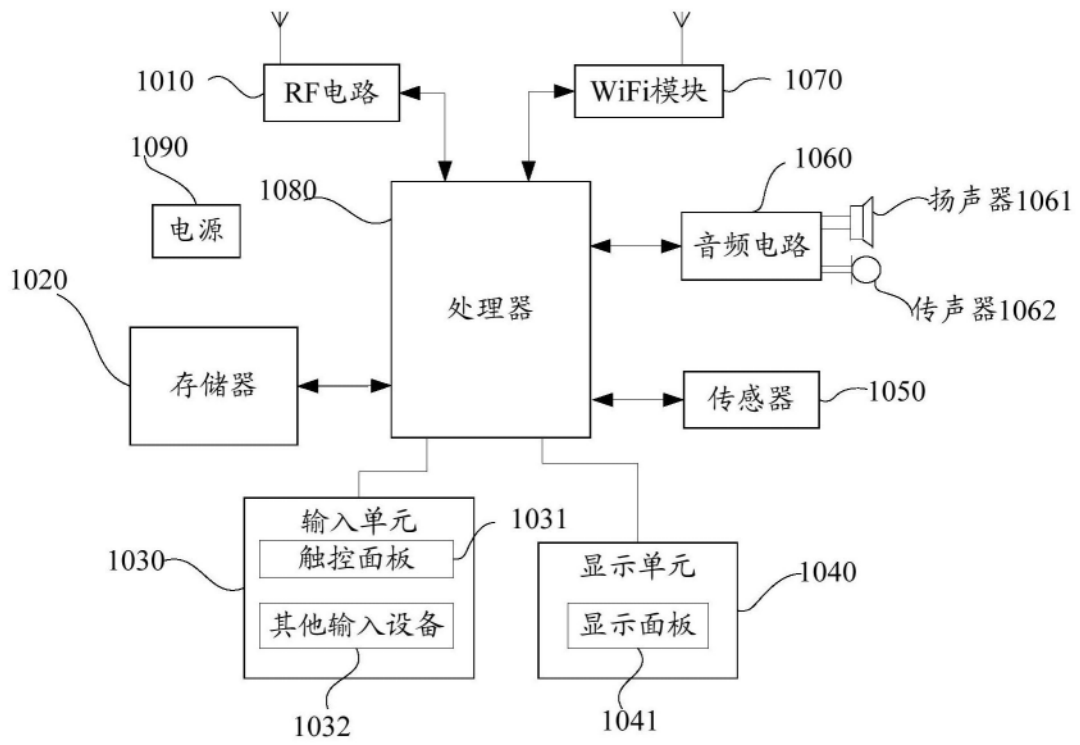


图6