



(12) 发明专利申请

(10) 申请公布号 CN 103678892 A

(43) 申请公布日 2014. 03. 26

(21) 申请号 201310643217. X

(22) 申请日 2013. 12. 03

(71) 申请人 腾讯科技(成都)有限公司

地址 610041 四川省成都市成都高新区拓东
新街 81 号天府软件园二期 C 区 4 号楼

(72) 发明人 钟勇 王旭新 姚建辉 邱林辉
杨敏

(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 刘映东

(51) Int. Cl.

G06F 19/00(2011. 01)

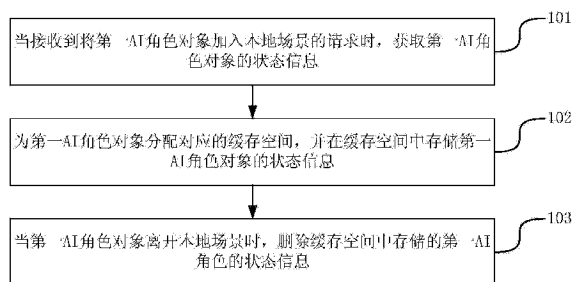
权利要求书2页 说明书10页 附图2页

(54) 发明名称

一种角色对象管理的方法和装置

(57) 摘要

本发明公开了一种角色对象管理的方法和装置,属于互联网技术领域。所述方法包括:当接收到将第一人工智能 AI 角色对象加入本地场景的请求时,获取所述第一 AI 角色对象的状态信息;为所述第一 AI 角色对象分配对应的缓存空间,并在所述缓存空间中存储所述第一 AI 角色对象的状态信息;当所述第一 AI 角色对象离开所述本地场景时,删除所述缓存空间中存储的所述第一 AI 角色的状态信息。采用本发明,可以提高对 AI 角色对象进行调度的灵活性。



1. 一种角色对象管理的方法,其特征在于,所述方法包括:

当接收到将第一人工智能 AI 角色对象加入本地场景的请求时,获取所述第一 AI 角色对象的状态信息;

为所述第一 AI 角色对象分配对应的缓存空间,并在所述缓存空间中存储所述第一 AI 角色对象的状态信息;

当所述第一 AI 角色对象离开所述本地场景时,删除所述缓存空间中存储的所述第一 AI 角色的状态信息。

2. 根据权利要求 1 所述的方法,其特征在于,所述当接收到将第一 AI 角色对象加入本地场景的请求时,获取所述第一 AI 角色对象的状态信息,包括:

当接收到将第一 AI 角色对象从其它服务器的场景迁移到本地场景的请求时,从所述其它服务器为所述第一 AI 角色对象分配的缓存空间中,获取所述第一 AI 角色对象的状态信息;或者,

当接收到在本地场景中新创建第一 AI 角色对象的请求时,获取所述请求中携带的所述第一 AI 角色对象的状态信息。

3. 根据权利要求 1 所述的方法,其特征在于,所述方法还包括:

当场景信息发生变化时,确定发生变化的场景信息所对应的本地场景中的关联角色对象;

如果所述关联角色对象中包括普通角色对象,则向所述关联角色对象中的普通角色对象对应的客户端发送所述发生变化的场景信息的变化信息;

如果所述关联角色对象中包括 AI 角色对象,则向 AI 处理平台发送所述发生变化的场景信息的变化信息和所述关联角色对象中包括 AI 角色对象的角色对象标识。

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:

当接收到客户端发送的携带有动作信息和角色对象标识的动作请求时,根据预设规则,验证所述动作信息的可行性,如果验证通过,则控制所述角色对象标识对应的普通角色对象执行所述动作信息对应的动作;或者,

当接收到 AI 处理平台发送的携带有动作信息和角色对象标识的动作请求时,根据预设规则,验证所述动作信息的可行性,如果验证通过,则控制所述角色对象标识对应的 AI 角色对象执行所述动作信息对应的动作。

5. 根据权利要求 4 所述的方法,其特征在于,所述根据预设规则,验证所述动作信息的可行性,包括:

将所述动作请求发送给逻辑平台,以使所述逻辑平台根据其存储的预设规则,验证所述动作信息的可行性,并反馈验证结果。

6. 根据权利要求 1 所述的方法,其特征在于,所述方法还包括:

记录本地分配有缓存空间的 AI 角色对象的角色类型为 AI 类型,记录本地分配有缓存空间的普通角色对象的角色类型为普通类型;

当接收到将本地场景中的第一普通角色对象转换为 AI 角色对象的请求时,将本地记录的所述第一普通角色对象的角色类型由普通类型修改为 AI 类型;

当接收到将本地场景中的第一 AI 角色对象转换为普通角色对象的请求时,将本地记录的所述第一 AI 角色对象的角色类型由 AI 类型修改为普通类型。

7. 一种角色对象管理的装置,其特征在于,所述装置包括:

获取模块,用于当接收到将第一人工智能 AI 角色对象加入本地场景的请求时,获取所述第一 AI 角色对象的状态信息;

存储模块,用于为所述第一 AI 角色对象分配对应的缓存空间,并在所述缓存空间中存储所述第一 AI 角色对象的状态信息;

删除模块,用于当所述第一 AI 角色对象离开所述本地场景时,删除所述缓存空间中存储的所述第一 AI 角色的状态信息。

8. 根据权利要求 7 所述的装置,其特征在于,所述获取模块,用于:

当接收到将第一 AI 角色对象从其它服务器的场景迁移到本地场景的请求时,从所述其它服务器为所述第一 AI 角色对象分配的缓存空间中,获取所述第一 AI 角色对象的状态信息;或者,

当接收到在本地场景中新创建第一 AI 角色对象的请求时,获取所述请求中携带的所述第一 AI 角色对象的状态信息。

9. 根据权利要求 7 所述的装置,其特征在于,所述装置还包括发送模块,用于:

当场景信息发生变化时,确定发生变化的场景信息所对应的本地场景中的关联角色对象;

如果所述关联角色对象中包括普通角色对象,则向所述关联角色对象中的普通角色对象对应的客户端发送所述发生变化的场景信息的变化信息;

如果所述关联角色对象中包括 AI 角色对象,则向 AI 处理平台发送所述发生变化的场景信息的变化信息和所述关联角色对象中包括 AI 角色对象的角色对象标识。

10. 根据权利要求 7 所述的装置,其特征在于,所述装置还包括控制模块,用于:

当接收到客户端发送的携带有动作信息和角色对象标识的动作请求时,根据预设规则,验证所述动作信息的可行性,如果验证通过,则控制所述角色对象标识对应的普通角色对象执行所述动作信息对应的动作;或者,

当接收到 AI 处理平台发送的携带有动作信息和角色对象标识的动作请求时,根据预设规则,验证所述动作信息的可行性,如果验证通过,则控制所述角色对象标识对应的 AI 角色对象执行所述动作信息对应的动作。

11. 根据权利要求 10 所述的装置,其特征在于,所述控制模块,用于:

将所述动作请求发送给逻辑平台,以使所述逻辑平台根据其存储的预设规则,验证所述动作信息的可行性,并反馈验证结果。

12. 根据权利要求 7 所述的装置,其特征在于,所述装置还包括转换模块,用于:

记录本地分配有缓存空间的 AI 角色对象的角色类型为 AI 类型,记录本地分配有缓存空间的普通角色对象的角色类型为普通类型;

当接收到将本地场景中的第一普通角色对象转换为 AI 角色对象的请求时,将本地记录的所述第一普通角色对象的角色类型由普通类型修改为 AI 类型;

当接收到将本地场景中的第一 AI 角色对象转换为普通角色对象的请求时,将本地记录的所述第一 AI 角色对象的角色类型由 AI 类型修改为普通类型。

一种角色对象管理的方法和装置

技术领域

[0001] 本发明涉及互联网技术领域,特别涉及一种角色对象管理的方法和装置。

背景技术

[0002] 随着互联网技术的不断发展,互联网已经成为了人们生活和工作中非常重要的组成部分,随着互联网的普及,网络游戏得到了快速的推广,成为了人们主要的娱乐方式之一。

[0003] 在网络游戏中,一般都会设置一个或多个场景,例如,角色扮演类游戏副本场景、野外场景等,棋牌类游戏桌面场景等。在场景中可以包括一个或多个角色对象(游戏人物),角色对象可以包括普通角色对象(由玩家控制的游戏人物)或 AI (Artificial Intelligence,人工智能)角色对象(由计算机采用人工智能方式控制的游戏人物)。AI 角色对象可以实现与玩家控制的普通角色对象的交互,如对话、对战、协作等,使网络游戏的内容更加丰富、更加多元化。

[0004] 在实现本发明的过程中,发明人发现现有技术至少存在以下问题:

[0005] 在现有技术中, AI 角色对象是作为游戏的场景的一部分存在的,在场景的游戏服务器中, AI 角色对象的状态信息(如人物的等级、位置、血量等)等信息是固化在主程序中的。因为主程序是很难动态修改的,所以导致一些动态的 AI 角色对象调度方式无法实现,例如,动态添加新 AI 角色对象(即由管理人员或玩家用户随机请求添加 AI 角色对象),或在不同服务器之间动态迁移 AI 角色对象(即由管理人员或玩家用户随机请求对某 AI 角色对象进行任意目的地的迁移)等,这些调度方式都很难动态实现。从而,导致对 AI 角色对象的进行调度的灵活性较差。

发明内容

[0006] 为了解决现有技术的问题,本发明实施例提供了一种角色对象管理的方法和装置,以提高对 AI 角色对象的进行调度的灵活性。所述技术方案如下:

[0007] 一方面,提供了一种角色对象管理的方法,所述方法包括:

[0008] 当接收到将第一人工智能 AI 角色对象加入本地场景的请求时,获取所述第一 AI 角色对象的状态信息;

[0009] 为所述第一 AI 角色对象分配对应的缓存空间,并在所述缓存空间中存储所述第一 AI 角色对象的状态信息;

[0010] 当所述第一 AI 角色对象离开所述本地场景时,删除所述缓存空间中存储的所述第一 AI 角色的状态信息。

[0011] 另一方面,提供了一种角色对象管理的装置,所述装置包括:

[0012] 获取模块,用于当接收到将第一人工智能 AI 角色对象加入本地场景的请求时,获取所述第一 AI 角色对象的状态信息;

[0013] 存储模块,用于为所述第一 AI 角色对象分配对应的缓存空间,并在所述缓存空间

中存储所述第一 AI 角色对象的状态信息；

[0014] 删除模块,用于当所述第一 AI 角色对象离开所述本地场景时,删除所述缓存空间中存储的所述第一 AI 角色的状态信息。

[0015] 本发明实施例提供的技术方案带来的有益效果是：

[0016] 本发明实施例中,当接收到将第一 AI 角色对象加入本地场景的请求时,获取第一 AI 角色对象的状态信息,为第一 AI 角色对象分配对应的缓存空间,并在缓存空间中存储第一 AI 角色对象的状态信息,当第一 AI 角色对象离开本地场景时,删除缓存空间中存储的第一 AI 角色的状态信息,这样,在服务器的主程序运行过程中,可以随时对 AI 角色对象进行添加和移出的操作,从而,可以提高对 AI 角色对象进行调度的灵活性。

附图说明

[0017] 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,对于本领域普通技术人员来讲,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。

[0018] 图 1 是本发明实施例提供的角色对象管理的方法流程图；

[0019] 图 2 是本发明实施例提供的网络系统结构示意图；

[0020] 图 3 是本发明实施例提供的角色对象管理的装置结构示意图；

[0021] 图 4 是本发明实施例提供的服务器的结构示意图。

具体实施方式

[0022] 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。

[0023] 实施例一

[0024] 本发明实施例提供了一种角色对象管理的方法,如图 1 所示,该方法的处理流程可以包括如下的步骤：

[0025] 步骤 101,当接收到将第一 AI 角色对象加入本地场景的请求时,获取第一 AI 角色对象的状态信息。

[0026] 步骤 102,为第一 AI 角色对象分配对应的缓存空间,并在缓存空间中存储第一 AI 角色对象的状态信息。

[0027] 步骤 103,当第一 AI 角色对象离开本地场景时,删除缓存空间中存储的第一 AI 角色的状态信息。

[0028] 本发明实施例中,当接收到将第一 AI 角色对象加入本地场景的请求时,获取第一 AI 角色对象的状态信息,为第一 AI 角色对象分配对应的缓存空间,并在缓存空间中存储第一 AI 角色对象的状态信息,当第一 AI 角色对象离开本地场景时,删除缓存空间中存储的第一 AI 角色的状态信息,这样,在服务器的主程序运行过程中,可以随时对 AI 角色对象进行添加和移出的操作,从而,可以提高对 AI 角色对象进行调度的灵活性。

[0029] 实施例二

[0030] 本发明实施例提供了一种角色对象管理的方法,该方法的执行主体可以为服务

器,该服务器可以是用于用户之间互动以及用户与计算机互动的网络应用后台的服务器,如网络游戏后台的服务器,本实施例以执行主体为网络游戏(如棋牌类游戏、角色扮演类游戏等)后台的场景服务器为例进行方案的详细说明,其它执行主体的情况下与此类似,本实施例中不再赘述。

[0031] 在一个网络应用中,可以将用户能够控制角色对象进入的区域划分为多个场景,分别设置不同的场景服务器来管理。其中,场景服务器是用于对某个场景(如某个桌面场景、副本场景等)中的场景信息进行管理的服务器。场景信息是用于反映场景中(如牌桌中、副本中等)角色对象、环境等的状态、变化的信息,场景信息中包括角色对象的状态信息。

[0032] 下面将结合具体实施方式,对图 1 所示的处理流程进行详细的说明,内容可以如下:

[0033] 步骤 101,当接收到将第一 AI 角色对象加入本地场景的请求时,服务器(场景服务器)获取第一 AI 角色对象的状态信息。

[0034] 其中,角色对象是网络应用中的可以由用户或 AI 控制的人物或其它事物的统称, AI 角色对象是由计算机采用人工智能方式控制的角色对象,普通角色对象是用户(网络应用的使用者,如游戏玩家)控制的角色对象。第一 AI 角色对象可以是任意的 AI 角色对象。本地场景是本服务器所管理的场景。状态信息是用于描述某个角色对象属性的信息的总和,是场景信息中的一种,如棋牌类游戏中角色对象的状态信息可以包括手牌信息、级别信息、道具信息等,角色扮演类游戏中角色对象的状态信息可以包括外形信息、掌握技能信息、攻击力信息、防御力信息、级别信息等。

[0035] 在实施中,请求将第一 AI 角色对象加入本地场景的情况可能有很多种,相应的获取第一 AI 角色对象的状态信息的方式可能也有所不同,以下给出了其中几种可能的情况下步骤 101 的处理过程:

[0036] 情况一

[0037] 当接收到将第一 AI 角色对象从其它服务器的场景迁移到本地场景的请求时,从该其它服务器为第一 AI 角色对象分配的缓存空间中,获取第一 AI 角色对象的状态信息。

[0038] 在实施中,第一 AI 角色对象可以是某其它服务器对应的场景中的 AI 角色对象,在之前第一 AI 角色对象加入该其它服务器时,该其它服务器可以按照本发明实施例的处理流程,为第一 AI 角色对象分配缓存空间,并将第一 AI 角色对象的状态信息存储在为其分配的缓存空间中。在某些情况下,例如,管理人员或用户请求将第一 AI 角色对象从该其它服务器的场景迁移到本地场景中,或者,第一 AI 角色对象在该其它服务器的场景中移动时,到达了该场景与本地场景的边界,需要进行场景切换,等等,这些情况下该其它服务器可以向本服务器发送迁移请求,用于请求将第一 AI 角色对象从其它服务器的场景迁移到本地场景,本地服务器接收到该请求时,可以向该其它服务器获取其中缓存的第一 AI 角色对象的状态信息。

[0039] 情况二

[0040] 当接收到在本地场景中新创建第一 AI 角色对象的请求时,获取该请求中携带的第一 AI 角色对象的状态信息。

[0041] 在实施中,管理人员可以请求在某个场景中添加第一 AI 角色对象,在服务器开始运营之前(用户还不能通过客户端登陆服务器的阶段,这个阶段管理人员可以对服务器进

行数据配置,例如创建若干个 AI 角色对象,或加载若干个预先配置好的 AI 角色对象)或在服务器运营过程中,管理人员可以通过其使用的管理终端,配置第一 AI 角色对象的状态信息,可以选择此第一 AI 角色对象的外形,输入其名称、血量、攻击力、防御力等信息,并点击确认添加此 AI 角色对象,这时,管理终端可以生成创建第一 AI 角色对象的请求,在其中携带此 AI 角色对象的状态信息,并将该请求发送给本服务器,本服务器接收该请求时,可以从该请求中获取第一 AI 角色对象的状态信息。

[0042] 步骤 102,服务器为第一 AI 角色对象分配对应的缓存空间,并在缓存空间中存储第一 AI 角色对象的状态信息。

[0043] 在实施中,服务器可以根据其载荷能力,预先设置一定数量的缓存空间,每个缓存空间用于分配给一个普通角色用户或 AI 角色对象,以存储相应角色对象的状态信息。服务器还可以维护一个缓存状态表,用于记录哪些缓存空间处于空闲状态,哪些缓存空间被使用。当某个 AI 角色对象或普通角色对象加入本地场景时,可以在各个处于空闲状态的缓存空间中,选择一个缓存空间,分配给相应的 AI 角色对象或普通角色对象,将相应的状态信息存储到选择的缓存空间中。服务器还可以记录各角色对象与缓存空间地址的对应关系,以便于对各角色对象的状态信息的管理。

[0044] 在服务器运营时,对场景信息进行维护管理的过程中,服务器会根据角色对象(普通角色用户或 AI 角色对象)的状态变化,对相应的缓存空间中的状态信息进行调整,例如,某 AI 角色对象受到伤害力为 1 的攻击且没有防御措施,那么,服务器就将该 AI 角色对象对应的缓存空间中的状态信息中的血量数值降低 1。

[0045] 这里,服务器对普通角色对象和 AI 角色对象的状态信息采用相同的管理方式。

[0046] 步骤 103,当第一 AI 角色对象离开本地场景时,服务器删除缓存空间中存储的第一 AI 角色的状态信息。

[0047] 在实施中,第一 AI 角色对象离开本地场景的情况有很多种可能。例如,一种情况是,将第一 AI 角色对象从本地场景迁移到某其它服务器的场景,该情况下,在此其它服务器从本服务器中获取第一 AI 角色对象的属性信息并存储在其自身的缓存空间中之后,本服务器可以对本地缓存空间中存储的第一 AI 角色对象的属性信息进行删除。又例如,还有一种情况是,管理人员请求取消第一 AI 角色对象,这时,服务器可以直接将第一 AI 角色对象的状态信息从缓存空间中删除。

[0048] 本发明实施例中,可以设置独立的 AI 处理平台,用于处理 AI 角色对象的行为逻辑以对各 AI 角色对象进行控制。AI 处理平台可以是上述场景服务器的一个模块,也可以是一个独立的服务器。

[0049] 在场景信息发生变化时,服务器可以进行如下的处理:

[0050] 步骤一,当场景信息发生变化时,确定发生变化的场景信息所对应的本地场景中的关联角色对象。

[0051] 其中,关联角色对象可以是基于网络应用中的预设规则能够感知到上述发生变化的场景信息的角色对象,预设规则可以根据需求任意设置。

[0052] 在实施中,用户通过客户端控制普通角色对象执行动作时,或 AI 处理平台控制 AI 角色对象执行动作时,则会触发场景信息发生变化,例如,用户控制某普通角色对象出“杀”牌,则会触发场景信息发生变化,其中包括该普通角色对象的状态信息发生变化(手牌少了

一张“杀”牌),还包括桌面上多了一张“杀”牌,等。根据服务器中预先制定的场景变化规则,也可以触发场景信息发生变化,例如,预先设置每间隔 20 分钟在某指定位置掉落一颗炸弹,那么每到 20 分钟的时候,在该指定位置就会掉落一个炸弹,这时场景信息发生变化,其中包括环境信息的变化(生成一个炸弹)等。

[0053] 在实施中,确定发生变化的场景信息所对应的关联角色对象的方法可以多种多样,例如,可以根据发生变化的场景信息的位置以及角色对象的视觉范围,确定视觉范围能包含该场景信息的角色对象,作为关联角色对象,又例如,可以针对特殊的事件(该事件发生即场景信息发生变化)预先设置对应的关联角色对象。

[0054] 另外,服务器还根据发生变化的场景信息,确定受到影响需要随之发生变化的其它场景信息(可以包括环境信息、角色对象的状态信息等),并对此其它场景信息进行相应的调整。

[0055] 步骤二,如果关联角色对象中包括普通角色对象,则向关联角色对象中的普通角色对象对应的客户端发送发生变化的场景信息的变化信息;如果关联角色对象中包括 AI 角色对象,则向 AI 处理平台发送发生变化的场景信息的变化信息和关联角色对象中包括 AI 角色对象的角色对象标识。

[0056] 其中,变化信息可以是发生变化的场景信息变化后的数值,也可以是变化值,例如,某角色对象的原血量为 3,变化后的血量为 4,那么,此场景信息的变化信息可以是 4 也可以是增加 1。

[0057] 在实施中,服务器可以记录各普通角色对象与客户端的对应关系,如果关联角色对象中包括普通角色对象,可以根据该对应关系确定相应的普通角色对象对应的客户端,并向确定出的客户端发送上述变化信息,客户端可以根据此变化信息在其显示的场景中显示相应的动画效果,例如,面前的房屋发生倒塌,用户看到场景发生变化后,可以判断自己需要作出的反应,如果决定需要进行某动作,可以进一步通过客户端向服务器发出动作请求。

[0058] 在实施中,如果关联角色对象中包括 AI 角色对象,可以向 AI 处理平台发送处理请求,处理请求中携带上述变化信息和关联角色对象中包括的各 AI 角色对象的角色对象标识, AI 处理平台中可以预先记录各 AI 角色对象对应的逻辑规则(可以由 AI 处理平台根据 AI 角色对象所处的场景和状态等信息来确定对应的行为逻辑规则,也可以由管理人员预先为 AI 角色对象指定对应的行为逻辑规则),当接收到处理请求时, AI 处理平台可以根据其中的角色对象标识查找对应的行为逻辑规则,并基于变化信息和查找到的行为逻辑规则,进行行为决策,如果决定需要进行某动作,则根据行为决策向服务器发送动作请求。在行为决策的过程中,如果根据逻辑规则,需要某些场景信息,可以向服务器获取相应的场景信息。

[0059] 本发明实施例中,服务器在接收到客户端或 AI 处理平台发送的动作请求时,可以根据动作请求对相应的场景信息进行控制,处理过程可以如下:

[0060] 情况一,当接收到客户端发送的携带有动作信息和角色对象标识的动作请求时,根据预设规则,验证动作信息的可行性,如果验证通过,则控制所述角色对象标识对应的普通角色对象执行动作信息对应的动作。

[0061] 情况二,当接收到 AI 处理平台发送的携带有动作信息和角色对象标识的动作请

求时,根据预设规则,验证所述动作信息的可行性,如果验证通过,则控制所述角色对象标识对应的 AI 角色对象执行所述动作信息对应的动作。

[0062] 具体的,根据预设规则验证动作信息的可行性的处理过程可以如下:将动作请求发送给逻辑平台,以使逻辑平台根据其存储的预设规则,验证动作信息的可行性,并反馈验证结果。

[0063] 其中,动作请求用于请求服务器控制角色对象执行某指定动作。预设规则可以是网络应用中基于自然规则和 / 或人为规定制定的基本行为规则。本发明实施例中,可以设置独立的逻辑平台,用于处理网络应用的基本逻辑,如判断某动作的执行是否可行,例如,某用户请求挪动某个箱子,而基于游戏的预设规则,该箱子现在的状态是无法移动的,则判定拒绝该动作请求。逻辑平台可以是场景服务器的一个模块,也可以是一个独立的服务器。

[0064] 在实施中,服务器接收到客户端或 AI 处理平台发送的动作请求后,可以将动作请求发送给逻辑平台,由逻辑平台根据预设规则验证动作请求中的动作信息的可行性,将验证结果反馈给服务器,服务器接收到的验证结果如果是通过,则控制相应的角色对象执行相应的动作,例如,出“杀”牌,使用某技能等,验证结果如果是不通过,则服务器可以拒绝该动作请求。执行相应的动作可以使场景信息发生变化,进而可以依据本实施例上面的方法进行变化信息的通知。

[0065] 上述处理中,服务器对普通角色对象和 AI 角色对象的动作请求采用了相同的处理方式。

[0066] 本发明实施例提供的网络系统结构示意图可以如图 2 所示。在该网络系统结构中, AI 处理平台和逻辑平台是独立于场景服务器之外的服务器,这样, AI 处理平台和逻辑平台可以为一台或多台场景服务器提供服务。

[0067] 本发明实施例中,服务器可以记录本地缓存空间中存储有状态信息的各角色对象的角色类型,可以记录本地分配有缓存空间的 AI 角色对象的角色类型为 AI 类型,记录本地分配有缓存空间的普通角色对象的角色类型为普通类型。相应的,可以建立角色类型表,用于记录各角色的角色类型。基于上述的通过分配缓存空间来管理角色对象的状态信息的处理方式,以及服务器记录的各角色对象的角色类型,服务器可以按如下的方式实现普通角色对象和 AI 角色对象之间的转换:

[0068] 当接收到将本地场景中的第一普通角色对象转换为 AI 角色对象的请求时,将本地记录的第一普通角色对象的角色类型由普通类型修改为 AI 类型。

[0069] 在实施中,可以在客户端向用户提供智能托管功能,用户在客户端中操作启动智能托管功能时,可以触发客户端向服务器发送智能托管请求,服务器接收到智能托管请求后,可以在上述角色类型表中,将该客户端对应的普通角色对象的角色类型,由普通类型修改为 AI 类型。另外,服务器也可以根据预先制定的规则,在某个触发条件发生时,主动将某普通角色对象转换为 AI 角色对象。

[0070] 当接收到将本地场景中的第一 AI 角色对象转换为普通角色对象的请求时,将本地记录的第一 AI 角色对象的角色类型由 AI 类型修改为普通类型。

[0071] 在实施中,当用户在客户端中操作结束智能托管时,可以触发客户端向服务器发送智能托管终止请求,服务器接收到智能托管终止请求后,可以在上述角色类型表中,将该客户端对应的普通角色对象的角色类型,由 AI 类型修改为普通类型。另外,服务器也可以

根据预先制定的规则,在某个触发条件发生时,主动将某 AI 角色对象转换为普通角色对象。

[0072] 本发明实施例中,当接收到将第一 AI 角色对象加入本地场景的请求时,获取第一 AI 角色对象的状态信息,为第一 AI 角色对象分配对应的缓存空间,并在缓存空间中存储第一 AI 角色对象的状态信息,当第一 AI 角色对象离开本地场景时,删除缓存空间中存储的第一 AI 角色的状态信息,这样,在服务器的主程序运行过程中,可以随时对 AI 角色对象进行添加和移出的操作,从而,可以提高对 AI 角色对象进行调度的灵活性。

[0073] 实施例三

[0074] 基于相同的技术构思,本发明实施例还提供了一种角色对象管理的装置,如图 3 所示,所述装置包括:

[0075] 获取模块 310,用于当接收到将第一人工智能 AI 角色对象加入本地场景的请求时,获取所述第一 AI 角色对象的状态信息;

[0076] 存储模块 320,用于为所述第一 AI 角色对象分配对应的缓存空间,并在所述缓存空间中存储所述第一 AI 角色对象的状态信息;

[0077] 删除模块 330,用于当所述第一 AI 角色对象离开所述本地场景时,删除所述缓存空间中存储的所述第一 AI 角色的状态信息。

[0078] 优选的,所述获取模块 310,用于:

[0079] 当接收到将第一 AI 角色对象从其它服务器的场景迁移到本地场景的请求时,从所述其它服务器为所述第一 AI 角色对象分配的缓存空间中,获取所述第一 AI 角色对象的状态信息;或者,

[0080] 当接收到在本地场景中新创建第一 AI 角色对象的请求时,获取所述请求中携带的所述第一 AI 角色对象的状态信息。

[0081] 优选的,所述装置还包括发送模块,用于:

[0082] 当场景信息发生变化时,确定发生变化的场景信息所对应的本地场景中的关联角色对象;

[0083] 如果所述关联角色对象中包括普通角色对象,则向所述关联角色对象中的普通角色对象对应的客户端发送所述发生变化的场景信息的变化信息;

[0084] 如果所述关联角色对象中包括 AI 角色对象,则向 AI 处理平台发送所述发生变化的场景信息的变化信息和所述关联角色对象中包括 AI 角色对象的角色对象标识。

[0085] 优选的,所述装置还包括控制模块,用于:

[0086] 当接收到客户端发送的携带有动作信息和角色对象标识的动作请求时,根据预设规则,验证所述动作信息的可行性,如果验证通过,则控制所述角色对象标识对应的普通角色对象执行所述动作信息对应的动作;或者,

[0087] 当接收到 AI 处理平台发送的携带有动作信息和角色对象标识的动作请求时,根据预设规则,验证所述动作信息的可行性,如果验证通过,则控制所述角色对象标识对应的 AI 角色对象执行所述动作信息对应的动作。

[0088] 优选的,所述控制模块,用于:

[0089] 将所述动作请求发送给逻辑平台,以使所述逻辑平台根据其存储的预设规则,验证所述动作信息的可行性,并反馈验证结果。

[0090] 优选的,所述装置还包括转换模块,用于:

[0091] 记录本地分配有缓存空间的 AI 角色对象的角色类型为 AI 类型,记录本地分配有缓存空间的普通角色对象的角色类型为普通类型;

[0092] 当接收到将本地场景中的第一普通角色对象转换为 AI 角色对象的请求时,将本地记录的所述第一普通角色对象的角色类型由普通类型修改为 AI 类型;

[0093] 当接收到将本地场景中的第一 AI 角色对象转换为普通角色对象的请求时,将本地记录的所述第一 AI 角色对象的角色类型由 AI 类型修改为普通类型。

[0094] 本发明实施例中,当接收到将第一 AI 角色对象加入本地场景的请求时,获取第一 AI 角色对象的状态信息,为第一 AI 角色对象分配对应的缓存空间,并在缓存空间中存储第一 AI 角色对象的状态信息,当第一 AI 角色对象离开本地场景时,删除缓存空间中存储的第一 AI 角色的状态信息,这样,在服务器的主程序运行过程中,可以随时对 AI 角色对象进行添加和移出的操作,从而,可以提高对 AI 角色对象进行调度的灵活性。

[0095] 实施例四

[0096] 图 4 是本发明实施例中服务器(该服务器可以用作上述网络应用的场景服务器)的结构示意图。该服务器 1900 可因配置或性能不同而产生比较大的差异,可以包括一个或一个以上中央处理器(central processing units,CPU)1922(例如,一个或一个以上处理器)和存储器 1932,一个或一个以上存储应用程序 1942 或数据 1944 的存储介质 1930(例如一个或一个以上海量存储设备)。其中,存储器 1932 和存储介质 1930 可以是短暂存储或持久存储。存储在存储介质 1930 的程序可以包括一个或一个以上模块(图示没标出),每个模块可以包括对服务器中的一系列指令操作。更进一步地,中央处理器 1922 可以设置为与存储介质 1930 通信,在服务器 1900 上执行存储介质 1930 中的一系列指令操作。

[0097] 服务器 1900 还可以包括一个或一个以上电源 1926,一个或一个以上有线或无线网络接口 1950,一个或一个以上输入输出接口 1958,一个或一个以上键盘 1956,和/或,一个或一个以上操作系统 1941,例如 Windows Server™, Mac OS X™, Unix™, Linux™, FreeBSD™ 等等。

[0098] 服务器 1900 可以包括有存储器,以及一个或者一个以上的程序,其中一个或者一个以上程序存储于存储器中,且经配置以由一个或者一个以上处理器执行所述一个或者一个以上程序包含用于进行以下操作的指令:

[0099] 当接收到将第一人工智能 AI 角色对象加入本地场景的请求时,获取所述第一 AI 角色对象的状态信息;

[0100] 为所述第一 AI 角色对象分配对应的缓存空间,并在所述缓存空间中存储所述第一 AI 角色对象的状态信息;

[0101] 当所述第一 AI 角色对象离开所述本地场景时,删除所述缓存空间中存储的所述第一 AI 角色的状态信息。

[0102] 优选的,所述当接收到将第一 AI 角色对象加入本地场景的请求时,获取所述第一 AI 角色对象的状态信息,包括:

[0103] 当接收到将第一 AI 角色对象从其它服务器的场景迁移到本地场景的请求时,从所述其它服务器为所述第一 AI 角色对象分配的缓存空间中,获取所述第一 AI 角色对象的状态信息;或者,

[0104] 当接收到在本地场景中新创建第一 AI 角色对象的请求时,获取所述请求中携带的所述第一 AI 角色对象的状态信息。

[0105] 优选的,还包括:

[0106] 当场景信息发生变化时,确定发生变化的场景信息所对应的本地场景中的关联角色对象;

[0107] 如果所述关联角色对象中包括普通角色对象,则向所述关联角色对象中的普通角色对象对应的客户端发送所述发生变化的场景信息的变化信息;

[0108] 如果所述关联角色对象中包括 AI 角色对象,则向 AI 处理平台发送所述发生变化的场景信息的变化信息和所述关联角色对象中包括 AI 角色对象的角色对象标识。

[0109] 优选的,还包括:

[0110] 当接收到客户端发送的携带有动作信息和角色对象标识的动作请求时,根据预设规则,验证所述动作信息的可行性,如果验证通过,则控制所述角色对象标识对应的普通角色对象执行所述动作信息对应的动作;或者,

[0111] 当接收到 AI 处理平台发送的携带有动作信息和角色对象标识的动作请求时,根据预设规则,验证所述动作信息的可行性,如果验证通过,则控制所述角色对象标识对应的 AI 角色对象执行所述动作信息对应的动作。

[0112] 优选的,所述根据预设规则,验证所述动作信息的可行性,包括:

[0113] 将所述动作请求发送给逻辑平台,以使所述逻辑平台根据其存储的预设规则,验证所述动作信息的可行性,并反馈验证结果。

[0114] 优选的,还包括:

[0115] 记录本地分配有缓存空间的 AI 角色对象的角色类型为 AI 类型,记录本地分配有缓存空间的普通角色对象的角色类型为普通类型;

[0116] 当接收到将本地场景中的第一普通角色对象转换为 AI 角色对象的请求时,将本地记录的所述第一普通角色对象的角色类型由普通类型修改为 AI 类型;

[0117] 当接收到将本地场景中的第一 AI 角色对象转换为普通角色对象的请求时,将本地记录的所述第一 AI 角色对象的角色类型由 AI 类型修改为普通类型。

[0118] 本发明实施例中,当接收到将第一 AI 角色对象加入本地场景的请求时,获取第一 AI 角色对象的状态信息,为第一 AI 角色对象分配对应的缓存空间,并在缓存空间中存储第一 AI 角色对象的状态信息,当第一 AI 角色对象离开本地场景时,删除缓存空间中存储的第一 AI 角色的状态信息,这样,在服务器的主程序运行过程中,可以随时对 AI 角色对象进行添加和移出的操作,从而,可以提高对 AI 角色对象进行调度的灵活性。

[0119] 需要说明的是:上述实施例提供的角色对象管理的装置在对角色对象的状态信息进行管理时,仅以上述各功能模块的划分进行举例说明,实际应用中,可以根据需要而将上述功能分配由不同的功能模块完成,即将装置的内部结构划分成不同的功能模块,以完成以上描述的全部或者部分功能。另外,上述实施例提供的角色对象管理的装置与角色对象管理的方法实施例属于同一构思,其具体实现过程详见方法实施例,这里不再赘述。

[0120] 上述本发明实施例序号仅仅为了描述,不代表实施例的优劣。

[0121] 本领域普通技术人员可以理解实现上述实施例的全部或部分步骤可以通过硬件来完成,也可以通过程序来指令相关的硬件完成,所述的程序可以存储于一种计算机可读

存储介质中,上述提到的存储介质可以是只读存储器,磁盘或光盘等。

[0122] 以上所述仅为本发明的较佳实施例,并不用以限制本发明,凡在本发明的精神和原则之内,所作的任何修改、等同替换、改进等,均应包含在本发明的保护范围之内。

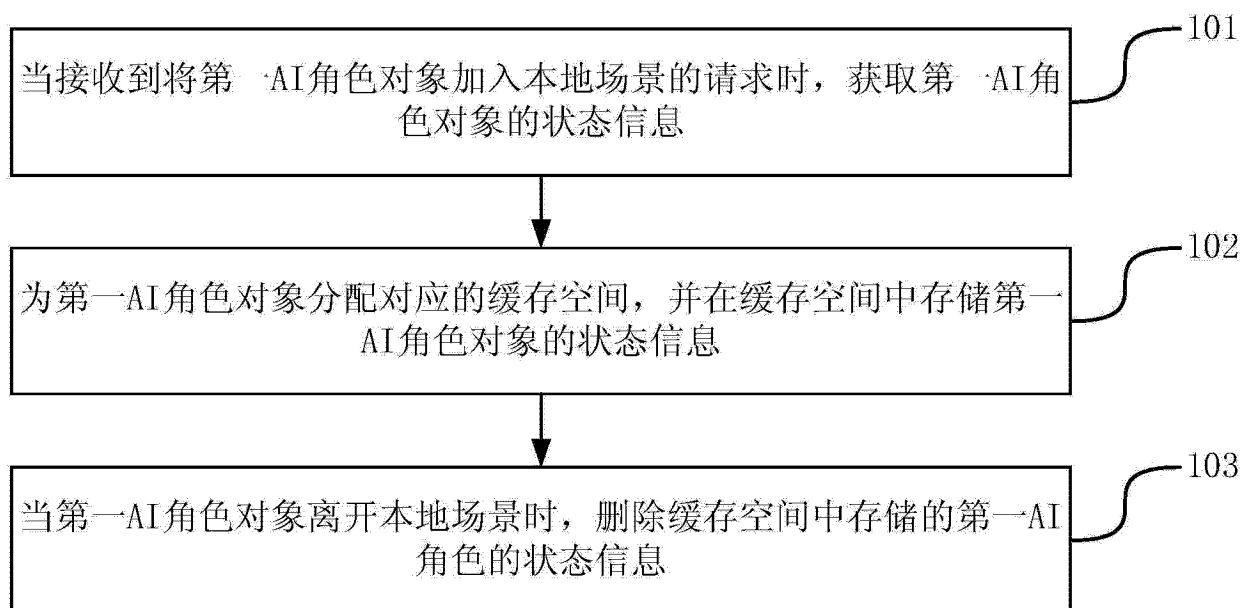


图 1

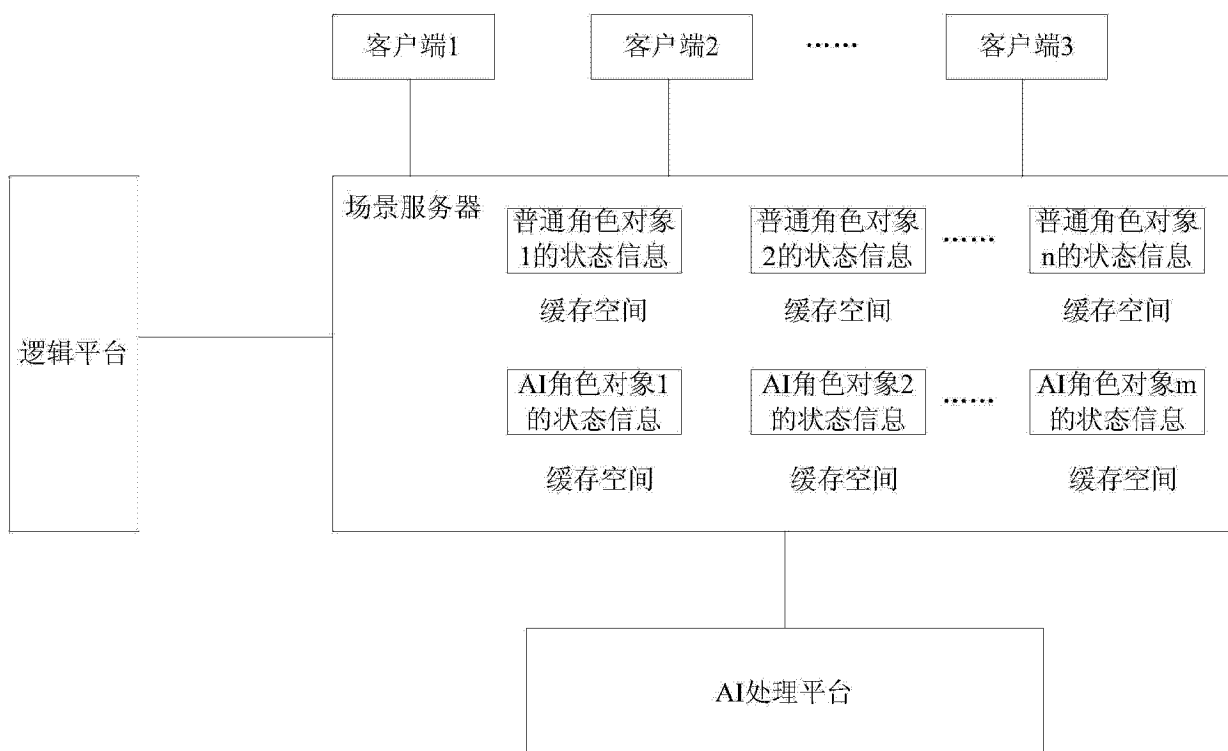


图 2

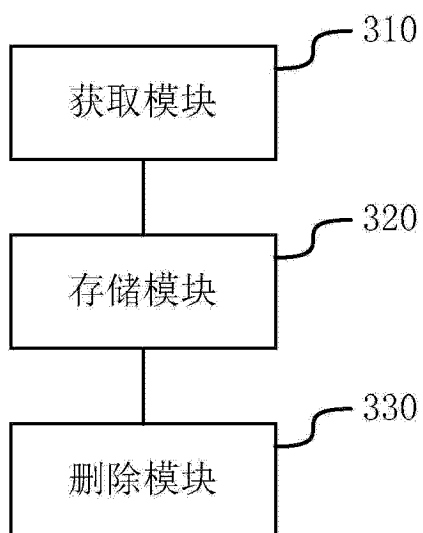


图 3

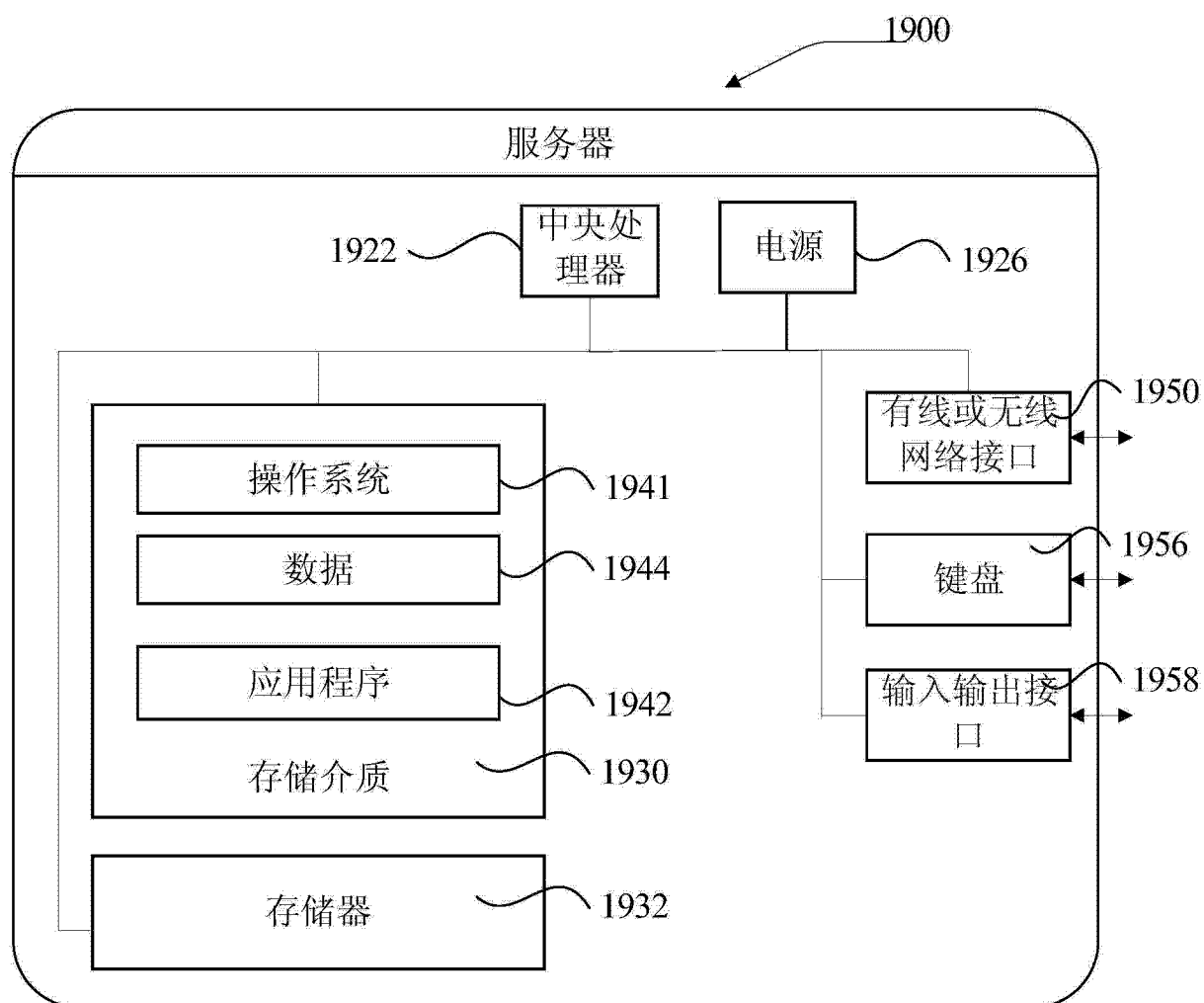


图 4