



(12)发明专利

(10)授权公告号 CN 105656688 B

(45)授权公告日 2019.09.20

(21)申请号 201610121881.1

(22)申请日 2016.03.03

(65)同一申请的已公布的文献号
申请公布号 CN 105656688 A

(43)申请公布日 2016.06.08

(73)专利权人 腾讯科技(深圳)有限公司
地址 518000 广东省深圳市福田区振兴路
赛格科技园2栋东403室

(72)发明人 江雷 单浩波 周锦源

(74)专利代理机构 广州三环专利商标代理有限
公司 44202

代理人 郝传鑫 贾允

(51)Int.Cl.

H04L 12/24(2006.01)

H04L 12/26(2006.01)

(56)对比文件

CN 105117575 A,2015.12.02,

CN 103049264 A,2013.04.17,

CN 101626420 A,2010.01.13,

CN 104077121 A,2014.10.01,

US 2014379100 A1,2014.12.25,

审查员 许晓娟

权利要求书2页 说明书14页 附图7页

(54)发明名称

状态控制方法和装置

(57)摘要

本发明公开了一种状态控制方法和装置。该方法包括:服务器检测到操作执行事件,操作执行事件用于请求对第一应用中的第一对象进行操作;服务器判断第一对象在混合状态机中的当前节点的下一个节点为第一状态节点还是第一行为树节点,混合状态机中的节点包括状态节点和行为树节点,混合状态机中的节点之间具有状态跳转关系;若判断出当前节点的下一个节点为第一状态节点,则服务器对第一对象执行第一状态节点所对应的第一状态中的操作;若判断出当前状态的下一个节点为第一行为树节点,则服务器对第一对象执行第一行为树节点所对应的第一行为树中的操作。本发明解决了相关技术使用有限状态机控制应用中个体执行操作效率低的技术问题。



1. 一种状态控制方法,其特征在于,包括:

服务器检测到操作执行事件,其中,所述操作执行事件用于请求对第一应用中的第一对象进行操作;

所述服务器判断所述第一对象在混合状态机中的当前节点的下一个节点为第一状态节点还是第一行为树节点,其中,所述混合状态机中的节点包括状态节点和行为树节点,所述混合状态机中的节点之间具有状态跳转关系;

若判断出所述当前节点的下一个节点为第一状态节点,则所述服务器对所述第一对象执行所述第一状态节点所对应的第一状态中的操作;

若判断出所述当前节点的下一个节点为所述第一行为树节点,则所述服务器对所述第一对象执行所述第一行为树节点所对应的第一行为树中的操作。

2. 根据权利要求1所述的方法,其特征在于,所述服务器判断所述第一对象在混合状态机中的当前节点的下一个节点为第一状态节点还是第一行为树节点包括:

若所述操作执行事件指示的是事件触发的操作或者异常的操作,则所述服务器判断所述当前节点的下一个节点为所述第一状态节点;

若所述操作执行事件指示的是流程类型的操作,则所述服务器判断所述当前节点的下一个节点为所述第一行为树节点。

3. 根据权利要求1所述的方法,其特征在于,所述服务器对所述第一对象执行所述第一行为树节点所对应的第一行为树中的操作包括:

所述服务器对所述第一对象执行所述第一行为树中的每个叶子节点所对应的操作。

4. 根据权利要求1所述的方法,其特征在于,在所述服务器检测到操作执行事件之前,所述方法还包括:

所述服务器获取第一配置信息,其中,所述第一配置信息用于指示所述状态节点、所述行为树节点、以及所述状态跳转关系;

所述服务器根据所述第一配置信息为所述第一对象生成所述混合状态机。

5. 根据权利要求4所述的方法,其特征在于,在所述服务器根据所述第一配置信息为所述第一对象生成所述混合状态机之后,所述方法还包括:

所述服务器获取第二配置信息;

所述服务器根据所述第二配置信息对所述混合状态机进行更新。

6. 根据权利要求5所述的方法,其特征在于,所述服务器根据所述第二配置信息对所述混合状态机进行更新包括以下至少之一:

所述服务器对所述混合状态机中已存在的状态节点、和/或已存在的行为树节点、和/或已存在的状态节点与已存在的行为树节点之间的状态跳转关系进行更新;

所述服务器删除所述混合状态机中已存在的状态节点、和/或已存在的行为树节点、和/或已存在的状态节点与已存在的行为树节点之间的状态跳转关系;

所述服务器在所述混合状态机中增加状态节点和/或行为树节点、和/或在增加的状态节点和/或行为树节点与已存在的状态节点和/或已存在的行为树节点之间建立状态跳转关系。

7. 一种状态控制装置,其特征在于,包括:

检测模块,用于服务器检测到操作执行事件,其中,所述操作执行事件用于请求对第一

应用中的第一对象进行操作；

判断模块,用于所述服务器判断所述第一对象在混合状态机中的当前节点的下一个节点为第一状态节点还是第一行为树节点,其中,所述混合状态机中的节点包括状态节点和行为树节点,所述混合状态机中的节点之间具有状态跳转关系;

第一操作模块,用于在判断出所述当前节点的下一个节点为第一状态节点,则所述服务器对所述第一对象执行所述第一状态节点所对应的第一状态中的操作;

第二操作模块,用于在判断出所述当前节点的下一个节点为所述第一行为树节点,则所述服务器对所述第一对象执行所述第一行为树节点所对应的第一行为树中的操作。

8. 根据权利要求7所述的装置,其特征在于,所述判断模块包括:

第一判定模块,用于在所述操作执行事件指示的是事件触发的操作或者异常的操作时,所述服务器判断所述当前节点的下一个节点为所述第一状态节点;

第二判定模块,用于在所述操作执行事件指示的是流程类型的操作,则所述服务器判断所述当前节点的下一个节点为所述第一行为树节点。

9. 根据权利要求7所述的装置,其特征在于,所述第二操作模块包括:

第一操作子模块,用于所述服务器对所述第一对象执行所述第一行为树中的每个叶子节点所对应的操作。

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:

第一获取模块,用于在所述服务器检测到操作执行事件之前,所述服务器获取第一配置信息,其中,所述第一配置信息用于指示所述状态节点、所述行为树节点、以及所述状态跳转关系;

生成模块,用于所述服务器根据所述第一配置信息为第一对象生成所述混合状态机。

11. 根据权利要求10所述的装置,其特征在于,所述装置还包括:

第二获取模块,用于在所述服务器根据所述第一配置信息为所述第一对象生成所述混合状态机之后,所述服务器获取第二配置信息;

更新模块,用于所述服务器根据所述第二配置信息对所述混合状态机进行更新。

12. 根据权利要求11所述的装置,其特征在于,所述更新模块包括以下至少之一:

第一更新子模块,用于所述服务器对所述混合状态机中已存在的状态节点、和/或已存在的行为树节点、和/或已存在的状态节点与已存在的行为树节点之间的状态跳转关系进行更新;

删除模块,用于所述服务器删除所述混合状态机中已存在的状态节点、和/或已存在的行为树节点、和/或已存在的状态节点与已存在的行为树节点之间的状态跳转关系;

增加模块,用于所述服务器在所述混合状态机中增加状态节点和/或行为树节点、和/或在增加的状态节点和/或行为树节点与已存在的状态节点和/或已存在的行为树节点之间建立状态跳转关系。

状态控制方法和装置

技术领域

[0001] 本发明涉及计算机领域,具体而言,涉及一种状态控制方法和装置。

背景技术

[0002] 目前,应用中控制个体执行操作的方式通常包括有限状态机和行为树,但是,有限状态机和行为树在执行操作过程中各自均存在缺陷,具体地:对于有限状态机,当状态非常多时,有限状态机需要处理大量的状态转换关系,不仅增加了设计者的工作量,而且不便于对有限状态机的维护,导致应用中个体执行操作效率低。对于行为树,行为树在状态跳转方面并不直观;行为树适合按计划执行一系列行为,不能有效地处理一些异常与特殊情况下的状态跳转。

[0003] 针对上述的问题,目前尚未提出有效的解决方案。

发明内容

[0004] 本发明实施例提供了一种状态控制方法和装置,以至少解决相关技术使用有限状态机控制应用中个体执行操作效率低的技术问题。

[0005] 根据本发明实施例的一个方面,提供了一种状态控制方法,包括:服务器检测到操作执行事件,其中,操作执行事件用于请求对第一应用中的第一对象进行操作;服务器判断第一对象在混合状态机中的当前节点的下一个节点为第一状态节点还是第一行为树节点,其中,混合状态机中的节点包括状态节点和行为树节点,混合状态机中的节点之间具有状态跳转关系;若判断出当前节点的下一个节点为第一状态节点,则服务器对第一对象执行第一状态节点所对应的第一状态中的操作;若判断出当前状态的下一个节点为第一行为树节点,则服务器对第一对象执行第一行为树节点所对应的第一行为树中的操作。

[0006] 根据本发明实施例的另一方面,还提供了一种状态控制装置,包括:检测模块,用于服务器检测到操作执行事件,其中,操作执行事件用于请求对第一应用中的第一对象进行操作;判断模块,用于服务器判断第一对象在混合状态机中的当前节点的下一个节点为第一状态节点还是第一行为树节点,其中,混合状态机中的节点包括状态节点和行为树节点,混合状态机中的节点之间具有状态跳转关系;第一操作模块,用于在判断出当前节点的下一个节点为第一状态节点,则服务器对第一对象执行第一状态节点所对应的第一状态中的操作;第二操作模块,用于在判断出当前状态的下一个节点为第一行为树节点,则服务器对第一对象执行第一行为树节点所对应的第一行为树中的操作。

[0007] 在本发明实施例中,通过服务器在检测到操作执行事件时判断第一对象在混合状态机中的当前节点的下一个节点为第一状态节点还是第一行为树节点,若为第一状态节点则对第一对象执行第一状态节点对应的第一状态中的操作,若为第一行为树节点则对第一对象执行第一行为树节点对应的第一行为树中的操作,达到了利用有限状态机和行为树共同控制应用中个体执行操作的目的,从而实现了提高应用中个体执行操作效率的技术效果,进而解决了相关技术使用有限状态机控制应用中个体执行操作效率低的技术问题。

附图说明

[0008] 此处所说明的附图用来提供对本发明的进一步理解,构成本申请的一部分,本发明的示意性实施例及其说明用于解释本发明,并不构成对本发明的不当限定。在附图中:

[0009] 图1是根据本发明实施例的状态控制方法的硬件环境的示意图;

[0010] 图2是根据本发明实施例的一种可选的状态控制方法的流程图;

[0011] 图3是根据本发明实施例的有限状态机的示意图;

[0012] 图4a是根据本发明实施例的状态分组之间有限状态机的示意图;

[0013] 图4b是根据本发明实施例的层次状态机的示意图;

[0014] 图5是根据本发明实施例的基于栈的状态机的示意图;

[0015] 图6是根据本发明实施例的行为树的示意图;

[0016] 图7是根据本发明实施例的混合状态机的示意图;

[0017] 图8是根据本发明实施例的状态控制装置的示意图;

[0018] 图9是根据本发明实施例的一种可选的状态控制装置的示意图;

[0019] 图10是根据本发明实施例的另一种可选的状态控制装置的示意图;

[0020] 图11是根据本发明实施例的再一种可选的状态控制装置的示意图;以及

[0021] 图12是根据本发明实施例的一种终端的结构框图。

具体实施方式

[0022] 为了使本技术领域的人员更好地理解本发明方案,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,所描述的实施例仅仅是本发明一部分的实施例,而不是全部的实施例。基于本发明中的实施例,本领域普通技术人员在没有做出创造性劳动前提下所获得的所有其他实施例,都应当属于本发明保护的范围。

[0023] 需要说明的是,本发明的说明书和权利要求书及上述附图中的术语“第一”、“第二”等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。应该理解这样使用的数据在适当情况下可以互换,以便这里描述的本发明的实施例能够以除了在这里图示或描述的那些以外的顺序实施。此外,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,例如,包含了一系列步骤或单元的过程、方法、系统、产品或设备不必限于清楚地列出的那些步骤或单元,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步骤或单元。

[0024] 实施例1

[0025] 根据本发明实施例,提供了一种状态控制方法的方法实施例。

[0026] 可选地,在本实施例中,上述状态控制方法可以应用于如图1所示的由服务器102和终端104所构成的硬件环境中。如图1所示,服务器102通过网络与终端104进行连接,上述网络包括但不限于:广域网、城域网或局域网,终端104并不限于PC、手机、平板电脑等。本发明实施例的数据处理方法可以由服务器102来执行,也可以由终端104来执行,还可以是由服务器102和终端104共同执行。其中,终端104执行本发明实施例的数据处理方法也可以是由安装在其上的客户端来执行。

[0027] 图2是根据本发明实施例的一种可选的状态控制方法的流程图,如图2所示,该方

法可以包括以下步骤：

[0028] 步骤S202,服务器检测到操作执行事件,其中,操作执行事件用于请求对第一应用中的第一对象进行操作;

[0029] 步骤S204,服务器判断第一对象在混合状态机中的当前节点的下一个节点为第一状态节点还是第一行为树节点,其中,混合状态机中的节点包括状态节点和行为树节点,混合状态机中的节点之间具有状态跳转关系;

[0030] 步骤S206,若判断出当前节点的下一个节点为第一状态节点,则服务器对第一对象执行第一状态节点所对应的第一状态中的操作;

[0031] 步骤S208,若判断出当前状态的下一个节点为第一行为树节点,则服务器对第一对象执行第一行为树节点所对应的第一行为树中的操作。

[0032] 通过上述步骤S202至步骤S208,通过服务器在检测到操作执行事件时判断第一对象在混合状态机中的当前节点的下一个节点为第一状态节点还是第一行为树节点,若为第一状态节点则对第一对象执行第一状态节点对应的第一状态中的操作,若为第一行为树节点则对第一对象执行第一行为树节点对应的第一行为树中的操作,达到了利用有限状态机和行为树共同控制应用中个体执行操作的目的,从而实现了提高应用中个体执行操作效率的技术效果,进而解决了相关技术使用有限状态机控制应用中个体执行操作效率低的技术问题。

[0033] 在步骤S202提供的技术方案中,第一应用可以为游戏类应用、即时通信类应用,或者其他类型应用,本发明对第一应用为作限定。第一应用中可以包括多个对象,这多个对象中包括第一对象,第一对象可以为这多个对象中的任意一个对象,需要说明的是,此处的对象可以为应用中某个执行事件的事件对象,比如游戏人物。第一应用可以安装在服务器中,也可以安装在客户端(比如智能手机、电能等),由服务器负责对该第一应用中的事件和对象进行处理和维护。操作执行事件可以用于请求对第一应用中的第一对象进行操作,其中,对第一对象进行的操作可以包括调整第一对象移动路径、控制第一对象释放技能等,本发明对此并未做具体限定。

[0034] 可选地,服务器可以实时检测该操作执行事件,以便于及时准确地响应该操作执行事件对第一应用中的第一对象进行操作。服务器在实时检测该操作执行事件的同时还可以实时检测第一应用中的第一对象是否正在执行某个操作,如果服务器检测到第一对象正在执行某个操作,则在检测到该操作执行事件时需要等待第一对象当前执行的操作执行完毕时才控制第一对象响应该操作执行事件;如果服务器检测到第一对象没有正在执行某个操作,则在检测到该操作执行事件时立即控制第一对象响应该操作执行事件。

[0035] 在步骤S204提供的技术方案中,混合状态机可以控制对第一对象进行操作,需要说明的是,混合状态机可以控制对一个或者多个第一应用中的对象进行操作。混合状态机由有限状态机和行为树组成,为了更好地说明混合状态机,本发明实施例首先对有限状态机和行为树做具体说明如下:

[0036] 有限状态机可以控制应用中个体对象执行操作,它包含了有限个状态和状态间的转换条件。同一时间内,个体对象只会处于某一个状态中,有限状态机需要从一个状态转换到另外一个状态实现执行不同的操作。例如,图3中表示了一个游戏应用中一个游戏人物的四个状态,分别为游荡、攻击、逃跑、寻找治疗,以及上述四个状态之间的转换关系,比如,游

戏人物从寻找治疗至游荡的转换条件为发现药物,从攻击至游荡的转换条件为对象不在视野范围内,此处不再一一举例上述四种状态之间的转换条件。

[0037] 有限状态机可以分为两种,分别为层次状态机(Hierarchical Finite State Machines,简称为HFSM)和基于栈的状态机(Stack based Finite State Machines,简称为SFSM)。其中,层次状态机提出了将状态分组的概念,不同组之间的状态共享相同的转换关系。它提升了状态控制的粒度,大大减少了重复的状态转换关系,可以让设计者更好的控制状态转换。如图4a所示,有限状态机中包括三个状态A、B、C,其中,A和C状态在 β 条件触发时都可以跳转到B状态,则可以将A和C状态分组,节省掉重复的 β 条件触发的状态转换,分组后的有限状态机,即有限状态机如图4b所示。

[0038] 基于栈的状态机可以使用栈结构管理状态的转换关系,如图5所示,栈顶保存了当前激活的状态C,状态之间的转换可以通过入栈和出栈操作来实现,该结构能够有效地处理基于事件触发的状态打断和继续的情况。例如,在游戏应用中需要添加一个新功能使得电脑AI随时暂停,则需要添加一个暂停状态。如果使用传统的有限状态机则需要为该暂停状态和每一个有限状态机中存在的状态之间设计转换关系。而使用基于栈的状态机,当事件触发需要暂停时只需要将暂停状态压栈即可,暂停结束时只需要将该暂停状态出栈即可以恢复到原有状态继续执行,大大节省了状态转换关系的设计。

[0039] 需要说明的是,本发明实施例中的混合状态机可以利用层次状态机,也可以利用基于栈的状态机,本发明对此并不做限定。

[0040] 行为树也可以控制应用中个体对象执行操作,行为树的节点可以分为以下几类:

[0041] 组合节点(Composite):行为树中的分支,即非叶子节点,可以包括选择节点(Selector)、序列节点(Sequence)、简单并行节点(Simple Parallel)。

[0042] 任务节点(Task):是行为树中的叶子节点,执行一项具体的任务,比如移动AI等。

[0043] 修饰节点(Decorator):依附于Composite节点或者Task节点中,用于判断某个分支或者某个叶子节点能否被执行。

[0044] 服务节点(Service):依附于Composite节点中,按一定频率被调用,用于更新该分支的信息。

[0045] 根节点(Root):行为树的根。

[0046] 选择节点(Selector):从左到右执行其子节点,若其子节点中有一个执行成功并返回,本节点就会立即停止执行。如果子节点返回成功,则该节点返回成功;若全部子节点返回失败,则该节点返回失败。

[0047] 序列节点(Sequence):从左到右执行其子节点,若其子节点中有一个执行失败并返回,本节点就会立即停止执行。如果有子节点返回失败,则该节点返回失败;若所有节点返回成功,则该节点返回成功。

[0048] 行为树的基本原理可以描述为:从根节点开始自顶向下,通过一些条件搜索这棵树,最终确定需要执行的叶子节点,并且执行它。树中除了根节点之外的每一个节点都有一个返回值,根据其执行情况返回成功或者失败。节点的返回值将会反馈给该节点的父节点,对其父节点的返回值造成一定的影响。如图6所示的行为树中,从根节点开始搜索,确定需要执行的叶子节点分别为任务节点0、任务节点1、任务节点2、任务节点3、任务节点4。一开始任务节点0执行失败,返回其给父节点选择节点0,继续执行任务节点1。任务节点1执行成

功,返回给其父节点选择节点0。由于任务节点0、任务节点1、任务节点2的父节点选择节点0是个选择节点,因此立即返回成功。同理,任务节点3执行成功,返回给其父节点顺序节点1,继续执行任务节点4。任务节点4执行失败,返回给其父节点顺序节点1。由于任务节点3、任务节点4的父节点顺序节点1是个顺序节点,因此立即返回失败。选择节点0、顺序节点1的父节点顺序节点0为顺序节点,在接收到选择节点0、顺序节点1的返回值时,确定顺序节点0的返回值为失败。顺序节点0将返回值返回至根节点,因此,整个行为树遍历完成之后,结果为失败。

[0049] 混合状态机中的节点可以包括状态节点和行为树节点,且混合状态机中的节点之间具有状态跳转关系,具体可以包括混合状态机中的状态节点之间、行为树节点之间、状态节点与行为树节点之间具有状态跳转关系。图7是根据本发明实施例的混合状态机的示意图,如图7所示,以游戏应用中游戏人物对应的混合状态机为例,混合状态机中可以包括多个节点,这多个节点中包括状态节点和行为树节点,其中,待机、寻路、追踪为状态节点,释放技能为行为树节点,混合状态机中的状态节点和状态节点之间,状态节点与行为树节点之间,行为树与行为树节点之间存在转换条件,图7中的箭头可以代表节点之间具有转换关系,节点之间具体的转换条件此处不做具体说明,行为树中包括多个节点,如图7所示,行为树节点中可以包括至少一个节点,比如顺序节点、选择节点、修饰节点、任务节点等。需要说明的是,本发明实施例对混合状态机中的状态节点和行为树节点的个数不做限定,对节点之间的转换条件不做限定,对行为树中节点个数,节点类型和节点关系不做限定。

[0050] 在步骤S204提供的技术方案中,服务器在检测到操作执行事件之后,判断第一对象在混合状态机中的当前节点的下一个节点为第一状态节点还是行为树节点。需要说明的是,操作执行事件中可以包括在混合状态机中第一对象所在的当前节点,也可以包括第一对象在执行当前节点所对应的操作之后转换的目标节点,即当前节点的下一个节点,在操作执行事件的触发下第一对象可以在混合状态机中从当前节点转换至下一个节点,其中,下一个节点可以是状态节点,也可以是行为树节点。

[0051] 在步骤S206和步骤S208提供的技术方案中,混合状态机中的任意一个状态节点(包括第一状态节点)所对应的状态中均包括对第一对象进行的操作,混合状态机中的任意一个行为树节点所对应的行为树中包括对第一对象进行的操作(包括第一行为树节点)。当服务器判断出当前节点的下一个节点为第一状态节点时,服务器对第一对象执行第一状态节点所对应的第一状态中的操作;当判断出当前状态的下一个节点为第一行为树节点时,服务器对第一对象执行第一行为树节点所对应的第一行为树中的操作。需要说明的是,本发明实施例对第一状态节点所对应的第一状态中的操作以及第一行为树所对应的操作不做限定。

[0052] 本发明实施例利用混合状态机融合了有限状态机和行为树的优点,对于需要遵照一定流程的计划,使用行为树处理,对于事件触发的行为或者一些异常情况,使用有限状态机处理。既能处理异常与特征情况下的状态跳转,又能根据某种计划执行一系列阶段性的行为。

[0053] 在实际应用场景中,如图7所示,对于游戏应用中的游戏人物来说主要有待机、追踪、寻路、释放技能四个状态。其中,释放技能的状态比较复杂,可能存在许多个不同的技能需要特定的排列组合来丰富游戏人物的技能表现。因此,本发明实施例将释放技能作为一

个大的通用状态,使用行为树去管理,释放技能状态和待机、追踪、寻路等其他状态的转换则使用有限状态机去控制。有限状态机可以使用程序代码维护,以保证大多数情况AI处理各种情况的稳定性。在实际使用中有限状态机选择基于栈的状态机,能够进一步减少多个状态转换带来的复杂性。在释放技能状态内部使用行为树去管理,游戏人物技能策划人员只需负责对该行为树进行编辑修改,而无需关心该行为树和外部其他状态的转换关系。由于该行为树内部大部分逻辑只是技能的组合与阶段的控制,并不会影响到整体的AI逻辑,策划人员完全可以自己操作而无需程序设计人员的协助。采用这样的方式,策划人员可以只关心自己需要的技能,控制出招流程,平衡游戏的节奏感,极大地节省了策划人员的工作量。此外,本发明实施例还可以进行非常方便地实现打断与恢复。考虑技能释放后造成游戏人物眩晕的例子,游戏人物眩晕结束后继续从中断点开始执行。使用传统的行为树解决这个问题将会非常复杂,或者需要很多任务才能办到。而本发明实施例通过借助于外部有限状态机,在有限状态机中新增一个眩晕状态,在触发眩晕的时候AI从当前状态跳转到眩晕状态,结束之后继续回到上一个状态。从而方便快速地实现了状态的打断与恢复。

[0054] 作为一种可选的实施例,步骤S204服务器判断第一对象在混合状态机中的当前节点的下一个节点为第一状态节点还是第一行为树节点可以包括:若操作执行事件指示的是事件触发的操作或者异常的操作,则服务器判断当前节点的下一个节点为第一状态节点;若操作执行事件指示的是流程类型的操作,则服务器判断当前节点的下一个节点为第一行为树节点。

[0055] 该实施例中服务器可以依据检测到的操作执行事件指示的操作决定当前节点的下一个节点为第一状态节点还是第一行为树节点,操作执行事件指示的操作可以包括事件触发的操作、异常的操作、流程类型的操作等,该实施例对操作执行事件指示的操作的内容不做限定,除了上述列举的内容外,还可以包括其他内容,此处不再一一举例说明。该实施例依据操作执行事件指示的操作判断当前节点的下一个节点是第一状态节点还是第一行为树节点,达到了准确确定当前节点的下一个节点的目的,进而能够达到根据不同的节点类型对第一对象执行不同操作的效果。

[0056] 作为一种可选的实施例,步骤S208服务器对第一对象执行第一行为树节点所对应的第一行为树中的操作包括:服务器对第一对象执行第一行为树中的每个叶子节点所对应的操作。

[0057] 该实施例中服务器在确定第一对象在混合状态机中的当前节点的下一个节点为第一行为树节点的情况下,服务器可以对第一对象执行第一行为树节点所对应的第一行为树中的操作,比如当第一对象为游戏人物,第一行为树节点为释放技能时,第一行为树中的操作可以包括调节技能强度、选择技能种类等。第一行为树中可以包括一个或者多个叶子节点,每个叶子节点对应一个第一行为树中的操作,即第一行为树中的操作可以包括一个,也可以包括多个。服务器对第一对象执行第一行为树节点所对应的第一行为树中的操作时,可以执行第一行为树中的每个叶子节点所对应的操作,具体的执行过程在介绍行为树的原理时已经说明,此处不再赘述。在执行完第一行为树中的每个叶子节点所对应的操作后,可以影响第一对象的属性,比如游戏人物的生命值、技能剩余量等。该实施例通过行为树对第一对象进行流程类型操作,能够避免使用有线状态机需要设置非常复杂的转换关系,进而达到了提高控制应用中对象执行操作的效率的技术效果。

[0058] 作为一种可选的实施例,在步骤S202服务器检测到操作执行事件之前,该实施例还可以包括以下步骤:

[0059] 步骤S2012,服务器获取第一配置信息,其中,第一配置信息用于指示状态节点、行为树节点、以及状态跳转关系。

[0060] 步骤S2014,服务器根据第一配置信息为第一对象生成混合状态机。

[0061] 在步骤S2012提供的技术方案中,第一配置信息可以是第一对象生成混合状态机所依据的信息,可以用于指示混合状态机中的状态节点、行为树节点以及状态跳转关系,需要说明的是,状态跳转关系可以包括状态节点之间、行为树节点之间、状态节点与行为树节点之间的状态跳转关系。第一配置信息可以包括混合状态机中状态节点的个数、行为树节点的个数、状态节点之间的转换条件、行为树节点之间的转换条件、状态节点与行为树节点之间的转换条件、状态节点所对应的状态中的操作,行为树节点所对应的行为树中的操作等。该实施例对第一配置信息不做具体限定,第一配置信息可以根据实际应用场景需求进行相应的调整。该第一配置信息可以由用户根据实际需求进行设定和调整。

[0062] 在步骤S2014提供的技术方案中,服务器在获取第一配置信息之后,依据该第一配置信息可以生成第一对象的混合状态机,比如,依据第一配置信息中的状态节点的个数可以设置混合状态机中的状态节点,依据行为树节点的个数可以设置混合状态机中的行为树节点,依据第一配置信息中的状态节点之间的转换条件可以设置混合状态机中的状态节点之间的跳转关系,依据第一配置信息中的行为树节点之间的转换条件可以设置混合状态机中的行为树节点之间的跳转关系,依据第一配置信息中的状态节点与行为树节点之间的转换条件可以设置混合状态机中的状态节点与行为树节点之间的跳转关系,依据第一配置信息中的状态节点所对应的状态中的操作可以配置混合状态机中的状态节点所对应的状态的操作,依据第一配置信息中的行为树节点所对应的行为树中的操作可以配置混合状态机中的行为树节点所对应的状态的操作等。上述只是列举了部分第一对象的混合状态机的配置操作,服务器根据第一配置信息为第一对象生成混合状态机的过程中还可以包括其他配置操作,此处不再一一举例说明。

[0063] 该实施例通过提前获取为第一对象生成混合状态机所需的第一配置信息,且该第一配置信息可以依据实际应用场景需求进行设定,能够达到满足用户实际需求的效果,同时,通过提前获取用于生成混合状态机所需的第一配置信息能够达到缩短混合状态机的生成时间的效果。

[0064] 作为一种可选的实施例,在步骤S2014服务器根据第一配置信息为第一对象生成混合状态机之后,该实施例还可以包括以下步骤:

[0065] 步骤S2016,服务器获取第二配置信息。

[0066] 步骤S2018,服务器根据第二配置信息对混合状态机进行更新。

[0067] 在步骤S2016提供的技术方案中,第二配置信息可以用于更新依据第一配置信息为第一对象生成的混合状态机,第二配置信息可以包括第一配置信息中的内容,还可以包括第一配置信息以外的内容,修改混合状态机中状态节点所对应的状态中的操作,修改混合状态机中行为树节点所对应的行为树中的操作,修改混合状态机中节点之间的转换条件和跳转关系,增加或者删除混合状态机中的节点,增加或者删除混合状态机中节点之间的转换条件和跳转关系等。该实施例对第二配置信息不做具体限定,第二配置信息可以包括

其他内容,此处不再一一举例说明。

[0068] 在步骤S2018提供的技术方案中,服务器在获取第二配置信息之后,依据该第二配置信息可以对混合状态机进行更新,可以包括对混合状态机中的节点进行更新,也可以包括对混合状态机中节点之间的跳转关系进行更新等。

[0069] 可选地,步骤S2018服务器根据第二配置信息对混合状态机进行更新可以包括以下至少之一:服务器对混合状态机中已存在的状态节点、和/或已存在的行为树节点、和/或已存在的状态节点与已存在的行为树节点之间的状态跳转关系进行更新;服务器删除混合状态机中已存在的状态节点、和/或已存在的行为树节点、和/或已存在的状态节点与已存在的行为树节点之间的状态跳转关系;服务器在混合状态机中增加状态节点和/或行为树节点、和/或在增加的状态节点和/或行为树节点与已存在的状态节点和/或已存在的行为树节点之间建立状态跳转关系。

[0070] 上述对混合状态机的更新操作只是本发明的一种可选实施例,需要说明的是,本发明并不对混合状态机的更新操作做限定,除了上述列举的更新操作外还可以包括其他更新操作,此处不再一一举例说明。该实施例通过提前获取用于更新混合状态机的第二配置信息,可以缩短混合状态机的更新时间,通过对混合状态机进行实时更新,能够提高混合状态机的实时性和准确性,进而达到利用混合状态机提高对第一对象执行操作的效率的效果。

[0071] 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。

[0072] 通过以上的实施方式的描述,本领域的技术人员可以清楚地了解到根据上述实施例的方法可借助软件加必需的通用硬件平台的方式来实现,当然也可以通过硬件,但很多情况下前者是更佳的实施方式。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一个存储介质(如ROM/RAM、磁碟、光盘)中,包括若干指令用以使得一台终端设备(可以是手机,计算机,服务器,或者网络设备等等)执行本发明各个实施例所述的方法。

[0073] 实施例2

[0074] 根据本发明实施例,还提供了一种用于实施上述状态控制方法的状态控制装置。图8是根据本发明实施例的状态控制装置的示意图,如图8所示,该装置可以包括:

[0075] 检测模块22,用于服务器检测到操作执行事件,其中,操作执行事件用于请求对第一应用中的第一对象进行操作;判断模块24,用于服务器判断第一对象在混合状态机中的当前节点的下一个节点为第一状态节点还是第一行为树节点,其中,混合状态机中的节点包括状态节点和行为树节点,混合状态机中的节点之间具有状态跳转关系;第一操作模块26,用于在判断出当前节点的下一个节点为第一状态节点,则服务器对第一对象执行第一状态节点所对应的第一状态中的操作;第二操作模块28,用于在判断出当前状态的下一个节点为第一行为树节点,则服务器对第一对象执行第一行为树节点所对应的第一行为树中的操作。

[0076] 需要说明的是,该实施例中的检测模块22可以用于执行本申请实施例1中的步骤S202,该实施例中的判断模块24可以用于执行本申请实施例1中的步骤S204,该实施例中的第一操作模块26可以用于执行本申请实施例1中的步骤S206,该实施例中的第二操作模块28可以用于执行本申请实施例1中的步骤S208。

[0077] 此处需要说明的是,上述模块与对应的步骤所实现的示例和应用场景相同,但不限于上述实施例1所公开的内容。需要说明的是,上述模块作为装置的一部分可以运行在如图1所示的硬件环境中,可以通过软件实现,也可以通过硬件实现。

[0078] 通过上述模块,通过服务器在检测到操作执行事件时判断第一对象在混合状态机中的当前节点的下一个节点为第一状态节点还是第一行为树节点,若为第一状态节点则对第一对象执行第一状态节点对应的第一状态中的操作,若为第一行为树节点则对第一对象执行第一行为树节点对应的第一行为树中的操作,达到了利用有限状态机和行为树共同控制应用中个体执行操作的目的,从而实现了提高应用中个体执行操作效率的技术效果,进而解决了相关技术使用有限状态机控制应用中个体执行操作效率低的技术问题。

[0079] 作为一种可选的实施例,图9是根据本发明实施例的一种可选的状态控制装置的示意图,如图9所示,判断模块24可以包括:第一判定模块242,用于在操作执行事件指示的是事件触发的操作或者异常的操作时,服务器判断当前节点的下一个节点为第一状态节点;第二判定模块244,用于在操作执行事件指示的是流程类型的操作,则服务器判断当前节点的下一个节点为第一行为树节点。

[0080] 该实施例中服务器可以依据检测到的操作执行事件指示的操作决定当前节点的下一个节点时第一状态节点还是第一行为树节点,操作执行事件指示的操作可以包括事件触发的操作、异常的操作、流程类型的操作等,该实施例对操作执行事件指示的操作的内容不做限定,除了上述列举的内容外,还可以包括其他内容,此处不再一一举例说明。该实施例依据操作执行事件指示的操作判断当前节点的下一个节点是第一状态节点还是第一行为树节点,达到了准确确定当前节点的下一个节点的目的,进而能够达到根据不同的节点类型对第一对象执行不同操作的效果。

[0081] 作为一种可选的实施例,图10是根据本发明实施例的另一种可选的状态控制装置的示意图,如图10所示,第二操作模块28可以包括:第一操作子模块282,用于服务器对第一对象执行第一行为树中的每个叶子节点所对应的操作。

[0082] 该实施例第一行为树中可以包括一个或者多个叶子节点,每个叶子节点对应一个第一行为树中的操作,即第一行为树中的操作可以包括一个,也可以包括多个。服务器在确定第一对象在混合状态机中的当前节点的下一个节点为第一行为树节点的情况下,服务器可以对第一对象执行第一行为树节点所对应的第一行为树中的操作,比如当第一对象为游戏人物,第一行为树节点为释放技能时,第一行为树中的操作可以包括调节技能强度、选择技能种类等。对第一对象执行第一行为树中的每个叶子节点所对应的操作,具体的执行过程在介绍行为树的原理时已经说明,此处不再赘述。在执行完第一行为树中的每个叶子节点所对应的操作后,可以影响第一对象的属性,比如游戏人物的生命值、技能剩余量等。该实施例通过行为树对第一对象进行流程类型操作,能够避免使用有线状态机需要设置非常复杂的转换关系,进而达到了提高控制应用中对象执行操作的效率的技术效果。

[0083] 作为一种可选的实施例,图11是根据本发明实施例的再一种可选的状态控制装置

的示意图,如图11所示,该实施例的状态控制装置还可以包括:第一获取模块212,用于在服务器检测到操作执行事件之前,服务器获取第一配置信息,其中,第一配置信息用于指示状态节点、行为树节点、以及状态跳转关系;生成模块214,用于服务器根据第一配置信息为第一对象生成混合状态机。

[0084] 需要说明的是,该实施例中的第一获取模块212可以用于执行本申请实施例1中的步骤S2012,该实施例中的生成模块214可以用于执行本申请实施例1中的步骤S2014。

[0085] 第一配置信息可以是第一对象生成混合状态机所依据的信息,可以用于指示混合状态机中的状态节点、行为树节点以及状态跳转关系,需要说明的是,状态跳转关系可以包括状态节点之间、行为树节点之间、状态节点与行为树节点之间的状态跳转关系。第一配置信息可以包括混合状态机中状态节点的个数、行为树节点的个数、状态节点之间的转换条件、行为树节点之间的转换条件、状态节点与行为树节点之间的转换条件、状态节点所对应的状态中的操作,行为树节点所对应的行为树中的操作等。该实施例对第一配置信息不做具体限定,第一配置信息可以根据实际应用场景需求进行相应的调整。该第一配置信息可以由用户根据实际需求进行设定和调整。服务器在获取第一配置信息之后,依据该第一配置信息可以生成第一对象的混合状态机,比如,依据第一配置信息中的状态节点的个数可以设置混合状态机中的状态节点,依据行为树节点的个数可以设置混合状态机中的行为树节点,依据第一配置信息中的状态节点之间的转换条件可以设置混合状态机中的状态节点之间的跳转关系,依据第一配置信息中的行为树节点之间的转换条件可以设置混合状态机中的行为树节点之间的跳转关系,依据第一配置信息中的状态节点与行为树节点之间的转换条件可以设置混合状态机中的状态节点与行为树节点之间的跳转关系,依据第一配置信息中的状态节点所对应的状态中的操作可以配置混合状态机中的状态节点所对应的状态的操作,依据第一配置信息中的行为树节点所对应的行为树中的操作可以配置混合状态机中的行为树节点所对应的状态的操作等。上述只是列举了部分第一对象的混合状态机的配置操作,服务器根据第一配置信息为第一对象生成混合状态机的过程中还可以包括其他配置操作,此处不再一一举例说明。

[0086] 该实施例通过提前获取为第一对象生成混合状态机所需的第一配置信息,且该第一配置信息可以依据实际应用场景需求进行设定,能够达到满足用户实际需求的效果,同时,通过提前获取用于生成混合状态机所需的第一配置信息能够达到缩短混合状态机的生成时间的效果。

[0087] 作为一种可选的实施例,如图11所示,该实施例的状态控制装置还可以包括:第二获取模块216,用于在服务器根据第一配置信息为第一对象生成混合状态机之后,服务器获取第二配置信息;更新模块218,用于服务器根据第二配置信息对混合状态机进行更新。

[0088] 需要说明的是,该实施例中的第二获取模块216可以用于执行本申请实施例1中的步骤S2016,该实施例中的更新模块218可以用于执行本申请实施例1中的步骤S2018。

[0089] 第二配置信息可以用于更新依据第一配置信息为第一对象生成的混合状态机,第二配置信息可以包括第一配置信息中的内容,还可以包括第一配置信息以外的内容,修改混合状态机中状态节点所对应的状态中的操作,修改混合状态机中行为树节点所对应的行为树中的操作,修改混合状态机中节点之间的转换条件和跳转关系,增加或者删除混合状态机中的节点,增加或者删除混合状态机中节点之间的转换条件和跳转关系等。该实施例

对第二配置信息不做具体限定,第二配置信息可以包括其他内容,此处不再一一举例说明。

[0090] 可选地,如图11所示,更新模块218可以包括以下至少之一:第一更新子模块2182,用于服务器对混合状态机中已存在的状态节点、和/或已存在的行为树节点、和/或已存在的状态节点与已存在的行为树节点之间的状态跳转关系进行更新;删除模块2184,用于服务器删除混合状态机中已存在的状态节点、和/或已存在的行为树节点、和/或已存在的状态节点与已存在的行为树节点之间的状态跳转关系;增加模块2186,用于服务器在混合状态机中增加状态节点和/或行为树节点、和/或在增加的状态节点和/或行为树节点与已存在的状态节点和/或已存在的行为树节点之间建立状态跳转关系。

[0091] 上述对混合状态机的更新操作只是本发明的一种可选实施例,需要说明的是,本发明并不对混合状态机的更新操作做限定,除了上述列举的更新操作外还可以包括其他更新操作,此处不再一一举例说明。该实施例通过提前获取用于更新混合状态机的第二配置信息,可以缩短混合状态机的更新时间,通过对混合状态机进行实时更新,能够提高混合状态机的实时性和准确性,进而达到利用混合状态机提高对第一对象执行操作的效率的效果。

[0092] 此处需要说明的是,上述模块与对应的步骤所实现的示例和应用场景相同,但不限于上述实施例1所公开的内容。需要说明的是,上述模块作为装置的一部分可以运行在如图1所示的硬件环境中,可以通过软件实现,也可以通过硬件实现,其中,硬件环境包括网络环境。

[0093] 实施例3

[0094] 根据本发明实施例,还提供了一种用于实施上述状态控制方法的服务器或终端。

[0095] 图12是根据本发明实施例的一种终端的结构框图,如图12所示,该终端可以包括:一个或多个(图中仅示出一个)处理器201、存储器203、以及传输装置205(如上述实施例中的发送装置),如图12所示,该终端还可以包括输入输出设备207。

[0096] 其中,存储器203可用于存储软件程序以及模块,如本发明实施例中的状态控制方法和装置对应的程序指令/模块,处理器201通过运行存储在存储器203内的软件程序以及模块,从而执行各种功能应用以及数据处理,即实现上述的状态控制方法。存储器203可包括高速随机存储器,还可以包括非易失性存储器,如一个或者多个磁性存储装置、闪存、或者其他非易失性固态存储器。在一些实例中,存储器203可进一步包括相对于处理器201远程设置的存储器,这些远程存储器可以通过网络连接至终端。上述网络的实例包括但不限于互联网、企业内部网、局域网、移动通信网及其组合。

[0097] 上述的传输装置205用于经由一个网络接收或者发送数据,还可以用于处理器与存储器之间的数据传输。上述的网络具体实例可包括有线网络及无线网络。在一个实例中,传输装置205包括一个网络适配器(Network Interface Controller, NIC),其可通过网线与其他网络设备与路由器相连从而可与互联网或局域网进行通讯。在一个实例中,传输装置205为射频(Radio Frequency, RF)模块,其用于通过无线方式与互联网进行通讯。

[0098] 其中,具体地,存储器203用于存储应用程序。

[0099] 处理器201可以通过传输装置205调用存储器203存储的应用程序,以执行下述步骤:服务器检测到操作执行事件,其中,操作执行事件用于请求对第一应用中的第一对象进行操作;服务器判断第一对象在混合状态机中的当前节点的下一个节点为第一状态节点还

是第一行为树节点,其中,混合状态机中的节点包括状态节点和行为树节点,混合状态机中的节点之间具有状态跳转关系;若判断出当前节点的下一个节点为第一状态节点,则服务器对第一对象执行第一状态节点所对应的第一状态中的操作;若判断出当前状态的下一个节点为第一行为树节点,则服务器对第一对象执行第一行为树节点所对应的第一行为树中的操作。

[0100] 处理器201还用于执行下述步骤:若操作执行事件指示的是事件触发的操作或者异常的操作,则服务器判断当前节点的下一个节点为第一状态节点;若操作执行事件指示的是流程类型的操作,则服务器判断当前节点的下一个节点为第一行为树节点。

[0101] 处理器201还用于执行下述步骤:服务器对第一对象执行第一行为树中的每个叶子节点所对应的操作。

[0102] 处理器201还用于执行下述步骤:在服务器检测到操作执行事件之前,服务器获取第一配置信息,其中,第一配置信息用于指示状态节点、行为树节点、以及状态跳转关系;服务器根据第一配置信息为第一对象生成混合状态机,其中,混合状态机中的状态节点和行为树节点之间具有状态跳转关系。

[0103] 处理器201还用于执行下述步骤:在服务器根据第一配置信息为第一对象生成混合状态机之后,服务器获取第二配置信息;服务器根据第二配置信息对混合状态机进行更新。

[0104] 处理器201还用于执行下述步骤:服务器对混合状态机中已存在的状态节点、和/或已存在的行为树节点、和/或已存在的状态节点与已存在的行为树节点之间的状态跳转关系进行更新;服务器删除混合状态机中已存在的状态节点、和/或已存在的行为树节点、和/或已存在的状态节点与已存在的行为树节点之间的状态跳转关系;服务器在混合状态机中增加状态节点和/或行为树节点、和/或在增加的状态节点和/或行为树节点与已存在的状态节点和/或已存在的行为树节点之间建立状态跳转关系。

[0105] 采用本发明实施例,提供了一种状态控制方法和装置。通过服务器在检测到操作执行事件时判断第一对象在混合状态机中的当前节点的下一个节点为第一状态节点还是第一行为树节点,若为第一状态节点则对第一对象执行第一状态节点对应的第一状态中的操作,若为第一行为树节点则对第一对象执行第一行为树节点对应的第一行为树中的操作,达到了利用有限状态机和行为树共同控制应用中个体执行操作的目的,从而实现了提高应用中个体执行操作效率的技术效果,进而解决了相关技术使用有限状态机控制应用中个体执行操作效率低的技术问题。

[0106] 可选地,本实施例中的具体示例可以参考上述实施例1和实施例2中所描述的示例,本实施例在此不再赘述。

[0107] 本领域普通技术人员可以理解,图12所示的结构仅为示意,终端可以是智能手机(如Android手机、iOS手机等)、平板电脑、掌上电脑以及移动互联网设备(Mobile Internet Devices, MID)、PAD等终端设备。图12其并不对上述电子装置的结构造成限定。例如,终端还可包括比图12中所示更多或者更少的组件(如网络接口、显示装置等),或者具有与图12所示不同的配置。

[0108] 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是可以通程序来指令终端设备相关的硬件来完成,该程序可以存储于一计算机可读存储介质

中,存储介质可以包括:闪存盘、只读存储器(Read-Only Memory,ROM)、随机存取器(Random Access Memory,RAM)、磁盘或光盘等。

[0109] 实施例4

[0110] 本发明的实施例还提供了一种存储介质。可选地,在本实施例中,上述存储介质可以用于执行状态控制方法的程序代码。

[0111] 可选地,在本实施例中,上述存储介质可以位于上述实施例所示的网络中的多个网络设备中的至少一个网络设备上。

[0112] 可选地,在本实施例中,存储介质被设置为存储用于执行以下步骤的程序代码:

[0113] S1,服务器检测到操作执行事件,其中,操作执行事件用于请求对第一应用中的第一对象进行操作;

[0114] S2,服务器判断第一对象在混合状态机中的当前节点的下一个节点为第一状态节点还是第一行为树节点,其中,混合状态机中的节点包括状态节点和行为树节点,混合状态机中的节点之间具有状态跳转关系;

[0115] S3,若判断出当前节点的下一个节点为第一状态节点,则服务器对第一对象执行第一状态节点所对应的第一状态中的操作;

[0116] S4,若判断出当前状态的下一个节点为第一行为树节点,则服务器对第一对象执行第一行为树节点所对应的第一行为树中的操作。

[0117] 可选地,存储介质还被设置为存储用于执行以下步骤的程序代码:若操作执行事件指示的是事件触发的操作或者异常的操作,则服务器判断当前节点的下一个节点为第一状态节点;若操作执行事件指示的是流程类型的操作,则服务器判断当前节点的下一个节点为第一行为树节点。

[0118] 可选地,存储介质还被设置为存储用于执行以下步骤的程序代码:服务器对第一对象执行第一行为树中的每个叶子节点所对应的操作。

[0119] 可选地,存储介质还被设置为存储用于执行以下步骤的程序代码:在服务器检测到操作执行事件之前,服务器获取第一配置信息,其中,第一配置信息用于指示状态节点、行为树节点、以及状态跳转关系;服务器根据第一配置信息为第一对象生成混合状态机,其中,混合状态机中的状态节点和行为树节点之间具有状态跳转关系。

[0120] 可选地,存储介质还被设置为存储用于执行以下步骤的程序代码:在服务器根据第一配置信息为第一对象生成混合状态机之后,服务器获取第二配置信息;服务器根据第二配置信息对混合状态机进行更新。

[0121] 可选地,存储介质还被设置为存储用于执行以下步骤的程序代码:服务器对混合状态机中已存在的状态节点、和/或已存在的行为树节点、和/或已存在的状态节点与已存在的行为树节点之间的状态跳转关系进行更新;服务器删除混合状态机中已存在的状态节点、和/或已存在的行为树节点、和/或已存在的状态节点与已存在的行为树节点之间的状态跳转关系;服务器在混合状态机中增加状态节点和/或行为树节点、和/或在增加的状态节点和/或行为树节点与已存在的状态节点和/或已存在的行为树节点之间建立状态跳转关系。

[0122] 可选地,本实施例中的具体示例可以参考上述实施例1和实施例2中所描述的示例,本实施例在此不再赘述。

[0123] 可选地,在本实施例中,上述存储介质可以包括但不限于:U盘、只读存储器(ROM, Read-Only Memory)、随机存取存储器(RAM, Random Access Memory)、移动硬盘、磁碟或者光盘等各种可以存储程序代码的介质。

[0124] 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。

[0125] 上述实施例中的集成的单元如果以软件功能单元的形式实现并作为独立的产品销售或使用,可以存储在上述计算机可读的存储介质中。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或者该技术方案的全部或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存储介质中,包括若干指令用以使得一台或多台计算机设备(可为个人计算机、服务器或者网络设备)执行本发明各个实施例所述方法的全部或部分步骤。

[0126] 在本发明的上述实施例中,对各个实施例的描述都各有侧重,某个实施例中未详述的部分,可以参见其他实施例的相关描述。

[0127] 在本申请所提供的几个实施例中,应该理解到,所揭露的客户端,可通过其它的方式实现。其中,以上所描述的装置实施例仅仅是示意性的,例如所述单元的划分,仅仅为一种逻辑功能划分,实际实现时可以有另外的划分方式,例如多个单元或组件可以结合或者可以集成到另一个系统,或一些特征可以忽略,或不执行。另一点,所显示或讨论的相互之间的耦合或直接耦合或通信连接可以是通过一些接口,单元或模块的间接耦合或通信连接,可以是电性或其它的形式。

[0128] 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单元来实现本实施例方案的目的。

[0129] 另外,在本发明各个实施例中的各功能单元可以集成在一个处理单元中,也可以是各个单元单独物理存在,也可以两个或两个以上单元集成在一个单元中。上述集成的单元既可以采用硬件的形式实现,也可以采用软件功能单元的形式实现。

[0130] 以上所述仅是本发明的优选实施方式,应当指出,对于本技术领域的普通技术人员来说,在不脱离本发明原理的前提下,还可以做出若干改进和润饰,这些改进和润饰也应视为本发明的保护范围。

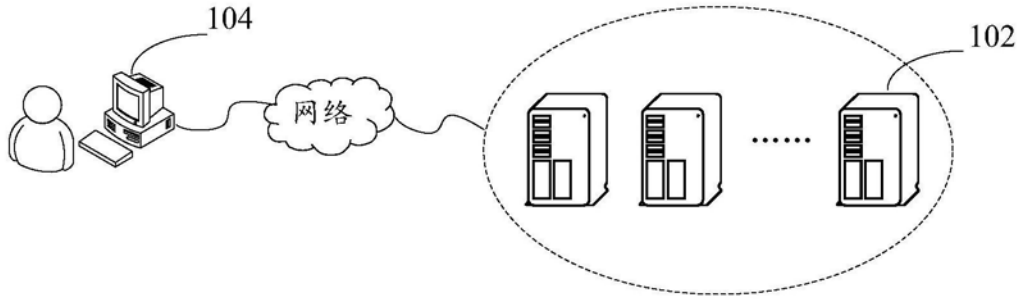


图1

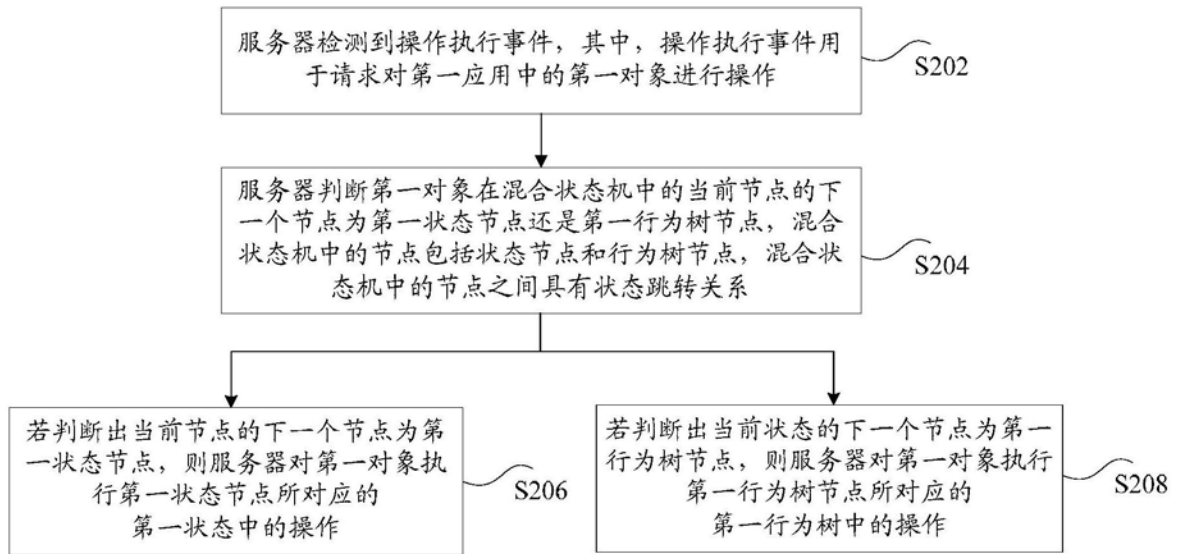


图2

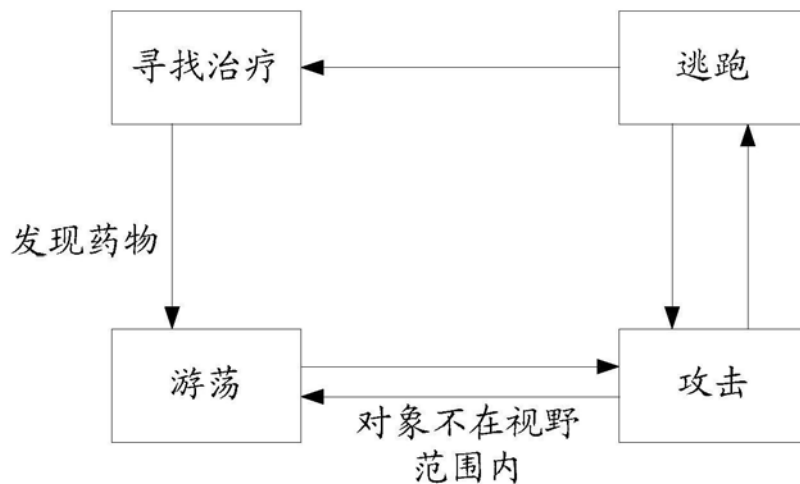


图3

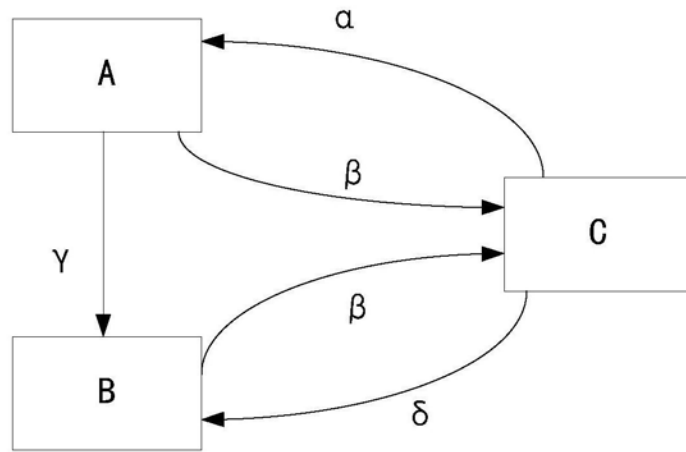


图4a

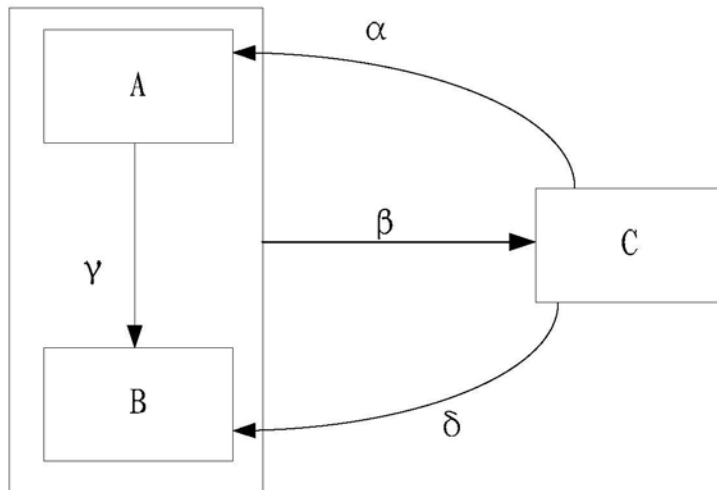


图4b

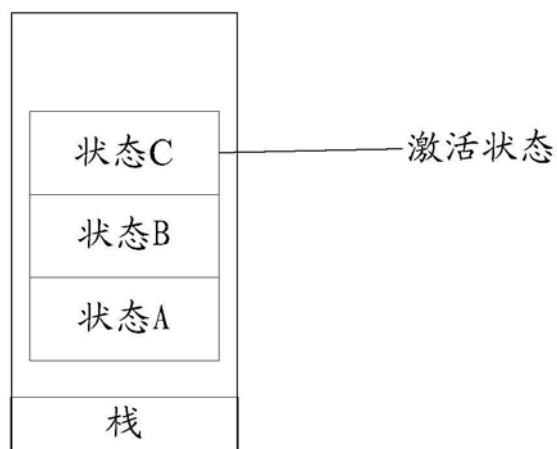


图5

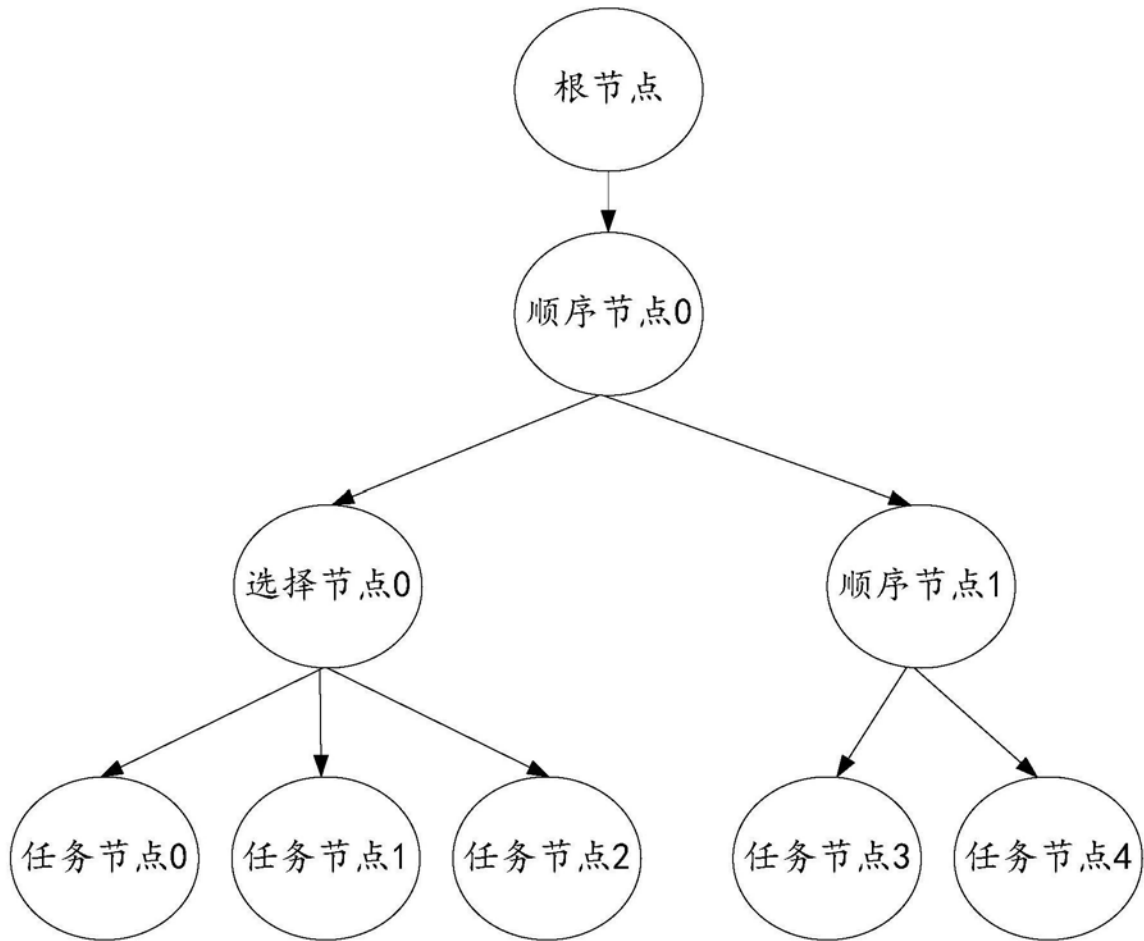


图6

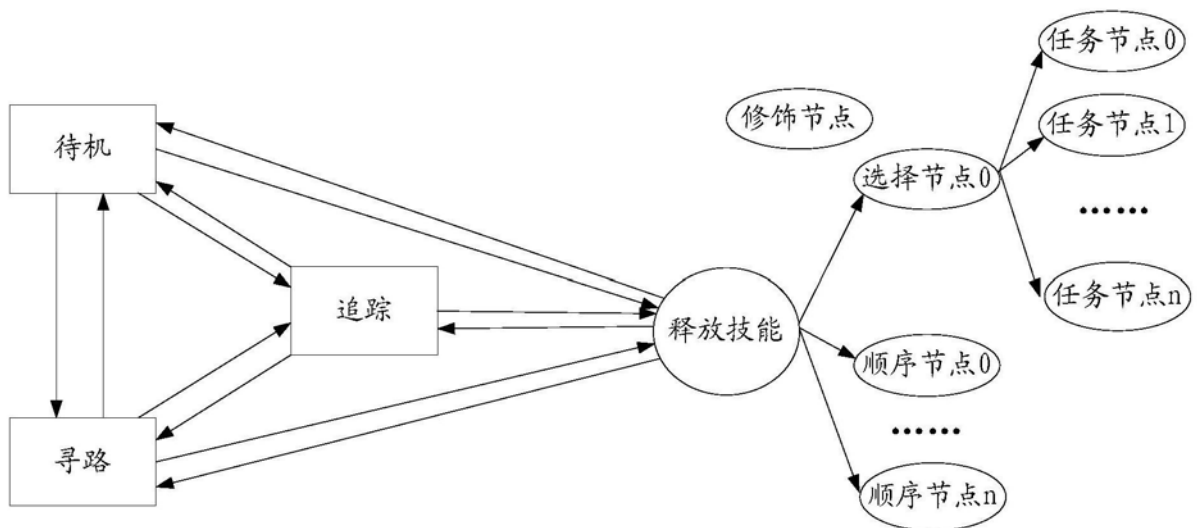


图7

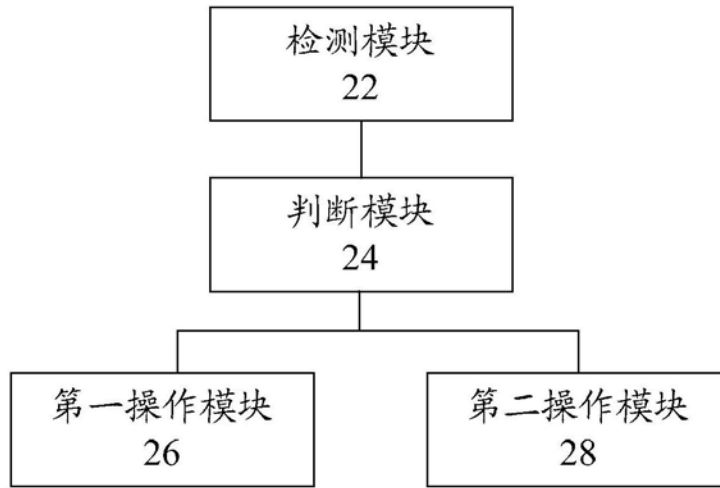


图8

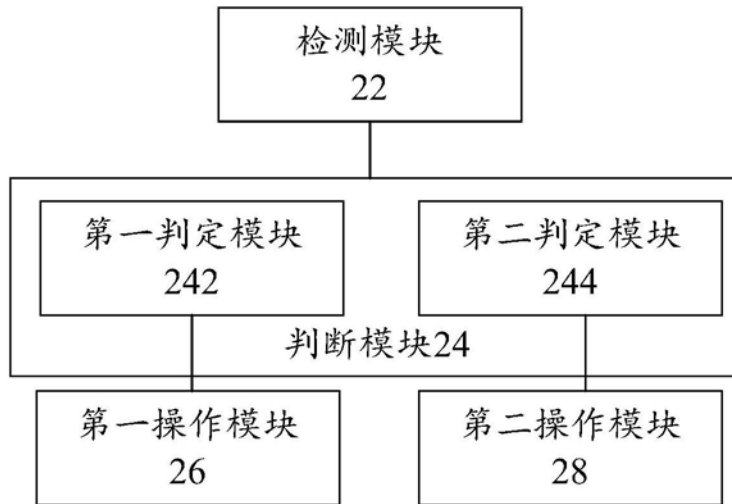


图9

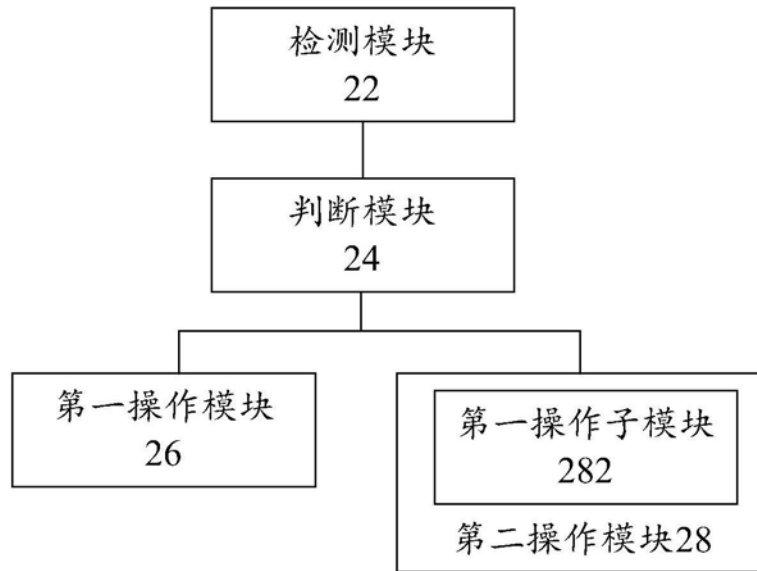


图10

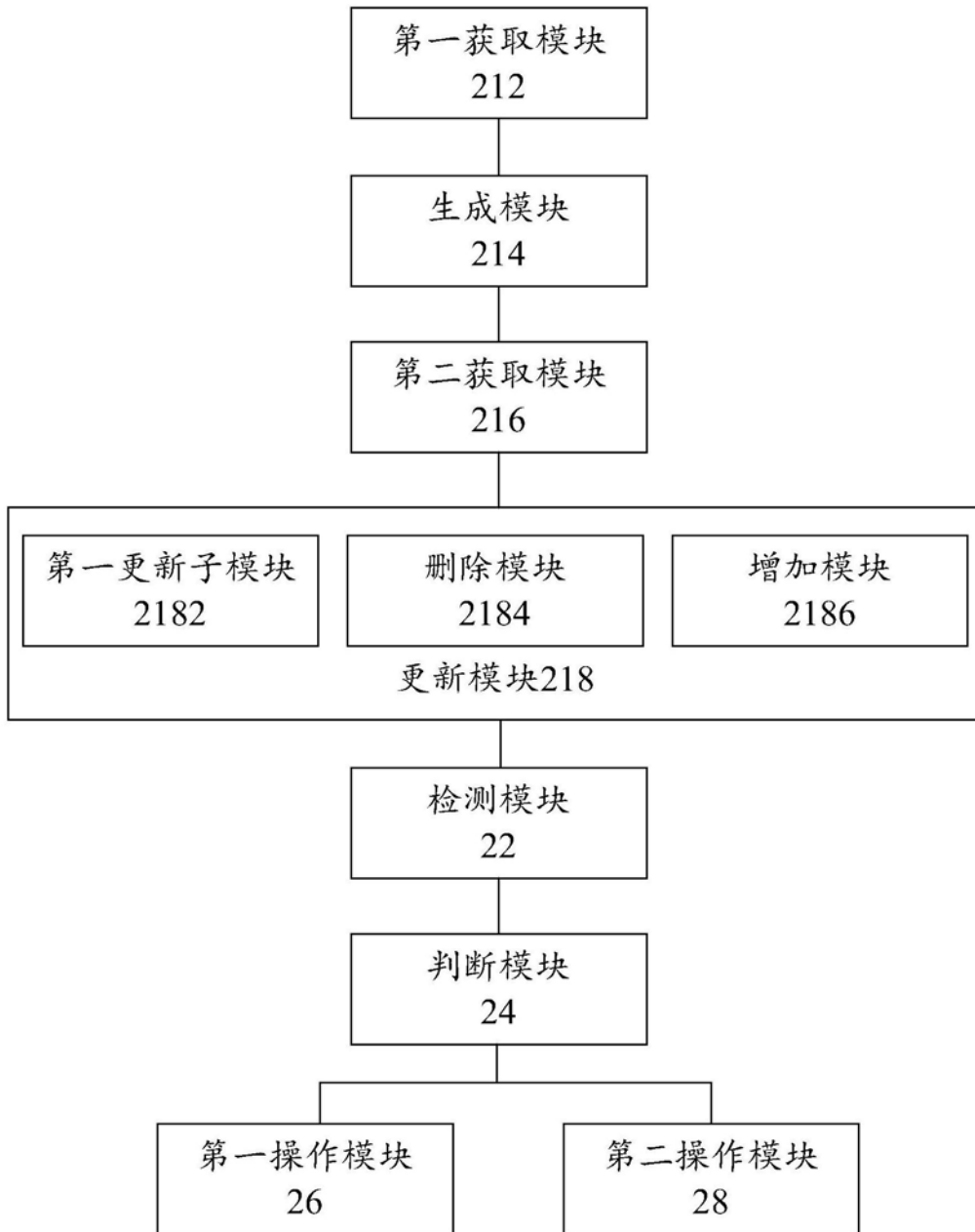


图11

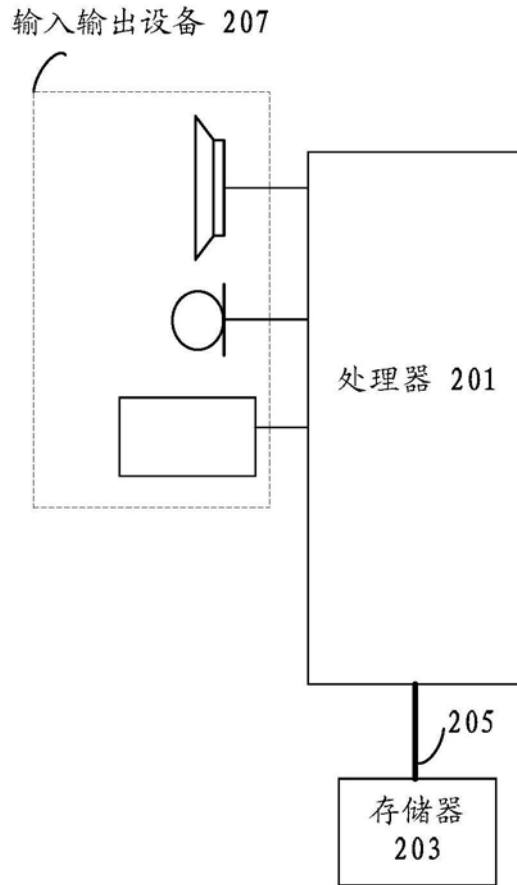


图12