



(12) 发明专利

(10) 授权公告号 CN 111481937 B

(45) 授权公告日 2023. 04. 07

(21) 申请号 202010277046.3

G06F 11/36 (2006.01)

(22) 申请日 2020.04.09

(56) 对比文件

(65) 同一申请的已公布的文献号

CN 108654089 A, 2018.10.16

申请公布号 CN 111481937 A

CN 110141864 A, 2019.08.20

CN 106095666 A, 2016.11.09

(43) 申请公布日 2020.08.04

CN 107943707 A, 2018.04.20

US 2013/0065694 A1, 2013.03.14

(73) 专利权人 网易(杭州)网络有限公司

审查员 金音

地址 310052 浙江省杭州市滨江区长河街

道网商路599号4幢7层

(72) 发明人 吴悦晨 李筱 陈赢峰 林磊

卢尚杰 范长杰 胡志鹏

(74) 专利代理机构 北京超凡宏宇专利代理事务

所(特殊普通合伙) 11463

专利代理师 于彬

(51) Int. Cl.

A63F 13/70 (2014.01)

权利要求书4页 说明书13页 附图4页

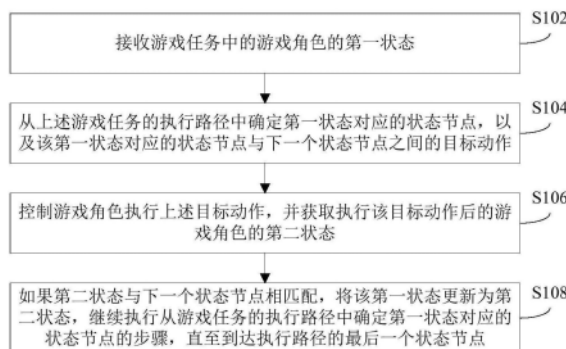
(54) 发明名称

游戏任务的测试方法、装置、测试终端和服务

器

(57) 摘要

本发明提供了一种游戏任务的测试方法、装置、测试终端和服务服务器,从游戏任务的执行路径中确定接收的游戏角色的第一状态对应的状态节点,以及该状态节点与下一个状态节点之间的目标动作;控制游戏角色执行目标动作,并获取执行目标动作后的第二状态;如果第二状态与下一个状态节点相匹配将第一状态更新为第二状态,继续执行从游戏任务的执行路径中确定第一状态对应的状态节点的步骤,直至到达执行路径的最后一个状态节点。该方式无需人工介入,可自动化完成游戏任务的测试,从而降低了游戏任务测试的人力物力成本,同时提高了测试效率,且该方式可以适用各种游戏任务,游戏任务的测试效果较好。



1. 一种游戏任务的测试方法,其特征在于,所述方法应用于测试终端;待测试的游戏任务预设有执行路径;所述执行路径包括顺序排列的多个状态节点,以及相邻的两个状态节点之间的动作;所述方法包括:

接收所述游戏任务中的游戏角色的第一状态;

从所述游戏任务的执行路径中确定所述第一状态对应的状态节点,以及所述第一状态对应的状态节点与下一个状态节点之间的目标动作;

控制所述游戏角色执行所述目标动作,并获取执行所述目标动作后的所述游戏角色的第二状态;

如果所述第二状态与所述下一个状态节点相匹配,将所述第一状态更新为所述第二状态,继续执行从所述游戏任务的执行路径中确定所述第一状态对应的状态节点的步骤,直至到达所述执行路径的最后一个状态节点;

所述游戏任务的执行路径,通过下述方式得到:

初始化所述游戏任务的路径搜索图,循环执行下述目标操作,直至所述路径搜索图停止更新,从所述路径搜索图中确定所述游戏任务的执行路径;

所述目标操作包括:

接收所述游戏任务中的游戏角色的第三状态,和所述游戏任务的任务描述信息;所述第三状态为所述游戏任务中所述游戏角色当前所处的状态;

根据所述第三状态和所述任务描述信息,确定待执行动作;

控制所述游戏角色执行所述待执行动作,并获取执行所述待执行动作后的所述游戏角色的第四状态;所述第四状态为所述游戏角色执行完所述待执行动作后在所述游戏任务中进入的新的时间点所对应的状态;

根据所述第三状态和所述待执行动作更新所述路径搜索图;将所述第四状态作为新的第三状态,继续执行确定当前执行动作的步骤,直至所述游戏任务执行完毕。

2. 根据权利要求1所述的方法,其特征在于,从所述路径搜索图中确定所述执行路径的步骤,包括:

如果从所述路径搜索图的第一个状态节点到最后一个状态节点,存在多条路径,将包含最少数量的状态节点的路径,确定为所述游戏任务的执行路径。

3. 根据权利要求1所述的方法,其特征在于,根据所述第三状态和所述任务描述信息,确定待执行动作的步骤,包括:

根据所述第三状态和所述任务描述信息,确定所述第三状态对应的动作空间;其中,所述动作空间中包括:所述游戏角色在所述第三状态下的合法动作;

从所述动作空间中确定待执行动作。

4. 根据权利要求3所述的方法,其特征在于,从所述动作空间中确定待执行动作的步骤,包括:

根据所述动作空间中各个合法动作在所述路径搜索图中所属的路径,以及所述各个合法动作在所述第三状态下已经被执行的次数,从所述各个合法动作中确定待执行动作。

5. 根据权利要求4所述的方法,其特征在于,根据所述动作空间中各个合法动作在所述路径搜索图中所属的路径,以及所述各个合法动作在所述第三状态下已经被执行的次数,从所述各个合法动作中确定待执行动作的步骤,包括:

针对所述动作空间中的每个合法动作,计算该合法动作的概率:

$$P_i = P_i^{\text{initial}} + \alpha * \sqrt{\log(\sum_{j=1}^n C_j)/C_i};$$

将所述概率最大的合法动作,确定为待执行动作;

其中, P_i 为第*i*个合法动作的概率; P_i^{initial} 为所述第*i*个合法动作的先验概率,如果所述第*i*个合法动作属于所述游戏任务的路径搜索图中指定路径, $P_i^{\text{initial}} = 1$,否则, $P_i^{\text{initial}} \in (0,0.1)$; α 为预设的权重参数; C_i 为所述第*i*个合法动作在所述第三状态下已经被执行的次数, C_j 为所述动作空间中的第*j*个合法动作在所述第三状态下已经被执行的次数, n 为所述动作空间中合法动作的总数; \log 为以2为底的对数函数。

6.根据权利要求5所述的方法,其特征在于,所述指定路径包括:所述游戏任务的当前的路径搜索图中,从第一个状态节点到最后一个状态节点的、包含有最少数量的状态节点的路径。

7.根据权利要求1所述的方法,其特征在于,根据所述第三状态和所述待执行动作更新所述路径搜索图的步骤,包括:

将所述第三状态更新为所述路径搜索图中的一个状态节点;

将所述待执行动作更新为所述第三状态对应的状态节点与下一个状态节点之间的动作。

8.根据权利要求1所述的方法,其特征在于,所述方法还包括:

如果所述第二状态与所述下一个状态节点不匹配,输出所述游戏任务的执行异常信息。

9.根据权利要求1所述的方法,其特征在于,所述测试终端分别与运行所述游戏任务的服务器和用户终端连接;

所述接收所述游戏任务中的游戏角色的第一状态的步骤,包括:

接收所述服务器发送的所述游戏任务中的游戏角色的第一状态;其中,所述游戏任务中的游戏角色的第一状态,由所述用户终端发送至所述服务器;

所述控制所述游戏角色执行所述目标动作的步骤,包括:

将所述目标动作发送至所述服务器,以通过所述服务器控制所述用户终端中的游戏角色执行所述目标动作。

10.一种游戏任务的测试方法,其特征在于,所述方法应用于运行有预设游戏任务的服务器;所述服务器分别与用于测试所述游戏任务的测试终端,以及运行有所述游戏任务的用户终端连接;所述游戏任务预设有执行路径;所述执行路径包括顺序排列的多个状态节点,以及相邻的两个状态节点之间的动作;所述方法包括:

向所述测试终端发送所述游戏任务中的游戏角色的第一状态,以通过所述测试终端从所述游戏任务的执行路径中,确定所述第一状态对应的状态节点,以及所述第一状态对应的状态节点与下一个状态节点之间的目标动作;

如果接收到来自所述测试终端的目标动作,控制所述用户终端中的游戏角色执行所述

目标动作,将执行所述目标动作后的所述游戏角色的第二状态发送至所述测试终端;如果所述第二状态与所述下一个状态节点相匹配,所述测试终端将所述第一状态更新为所述第二状态,继续执行从所述游戏任务的执行路径中确定所述第一状态对应的状态节点的步骤,直至到达所述执行路径的最后一个状态节点;

所述游戏任务的执行路径,通过下述方式得到:

初始化所述游戏任务的路径搜索图,循环执行下述目标操作,直至所述路径搜索图停止更新,从所述路径搜索图中确定所述游戏任务的执行路径;

所述目标操作包括:

接收所述游戏任务中的游戏角色的第三状态,和所述游戏任务的任务描述信息;所述第三状态为所述游戏任务中所述游戏角色当前所处的状态;

根据所述第三状态和所述任务描述信息,确定待执行动作;

控制所述游戏角色执行所述待执行动作,并获取执行所述待执行动作后的所述游戏角色的第四状态;所述第四状态为所述游戏角色执行完所述待执行动作后在所述游戏任务中进入的新的时间点所对应的状态;

根据所述第三状态和所述待执行动作更新所述路径搜索图;将所述第四状态作为新的第三状态,继续执行确定当前执行动作的步骤,直至所述游戏任务执行完毕。

11. 一种游戏任务的测试装置,其特征在于,所述装置设置于测试终端;待测试的游戏任务预设执行路径;所述执行路径包括顺序排列的多个状态节点,以及相邻的两个状态节点之间的动作;所述装置包括:

第一状态接收模块,用于接收所述游戏任务中的游戏角色的第一状态;

动作确定模块,用于从所述游戏任务的执行路径中确定所述第一状态对应的状态节点,以及所述第一状态对应的状态节点与下一个状态节点之间的目标动作;

第二状态接收模块,用于控制所述游戏角色执行所述目标动作,并获取执行所述目标动作后的所述游戏角色的第二状态;

循环执行模块,用于如果所述第二状态与所述下一个状态节点相匹配,将所述第一状态更新为所述第二状态,继续执行从所述游戏任务的执行路径中确定所述第一状态对应的状态节点的步骤,直至到达所述执行路径的最后一个状态节点;

所述游戏任务的执行路径,通过下述方式得到:

初始化所述游戏任务的路径搜索图,循环执行下述目标操作,直至所述路径搜索图停止更新,从所述路径搜索图中确定所述游戏任务的执行路径;

所述目标操作包括:

接收所述游戏任务中的游戏角色的第三状态,和所述游戏任务的任务描述信息;所述第三状态为所述游戏任务中所述游戏角色当前所处的状态;

根据所述第三状态和所述任务描述信息,确定待执行动作;

控制所述游戏角色执行所述待执行动作,并获取执行所述待执行动作后的所述游戏角色的第四状态;所述第四状态为所述游戏角色执行完所述待执行动作后在所述游戏任务中进入的新的时间点所对应的状态;

根据所述第三状态和所述待执行动作更新所述路径搜索图;将所述第四状态作为新的第三状态,继续执行确定当前执行动作的步骤,直至所述游戏任务执行完毕。

12. 一种游戏任务的测试装置,其特征在于,所述装置设置于运行有预设游戏任务的服务器;所述服务器分别与用于测试所述游戏任务的测试终端,以及运行有所述游戏任务的用户终端连接;所述游戏任务预设执行路径;所述执行路径包括顺序排列的多个状态节点,以及相邻的两个状态节点之间的动作;所述装置包括:

第一状态发送模块,用于向所述测试终端发送所述游戏任务中的游戏角色的第一状态,以通过所述测试终端从所述游戏任务的执行路径中,确定所述第一状态对应的状态节点,以及所述第一状态对应的状态节点与下一个状态节点之间的目标动作;

动作执行模块,用于如果接收到来自所述测试终端的目标动作,控制所述用户终端中的游戏角色执行所述目标动作,将执行所述目标动作后的所述游戏角色的第二状态发送至所述测试终端;如果所述第二状态与所述下一个状态节点相匹配,所述测试终端将所述第一状态更新为所述第二状态,继续执行从所述游戏任务的执行路径中确定所述第一状态对应的状态节点的步骤,直至到达所述执行路径的最后一个状态节点;

所述游戏任务的执行路径,通过下述方式得到:

初始化所述游戏任务的路径搜索图,循环执行下述目标操作,直至所述路径搜索图停止更新,从所述路径搜索图中确定所述游戏任务的执行路径;

所述目标操作包括:

接收所述游戏任务中的游戏角色的第三状态,和所述游戏任务的任务描述信息;所述第三状态为所述游戏任务中所述游戏角色当前所处的状态;

根据所述第三状态和所述任务描述信息,确定待执行动作;

控制所述游戏角色执行所述待执行动作,并获取执行所述待执行动作后的所述游戏角色的第四状态;所述第四状态为所述游戏角色执行完所述待执行动作后在所述游戏任务中进入的新的时间点所对应的状态;

根据所述第三状态和所述待执行动作更新所述路径搜索图;将所述第四状态作为新的第三状态,继续执行确定当前执行动作的步骤,直至所述游戏任务执行完毕。

13. 一种测试终端,其特征在于,包括处理器和存储器,所述存储器存储有能够被所述处理器执行的机器可执行指令,所述处理器执行所述机器可执行指令以实现权利要求1至10任一项所述的游戏任务的测试方法。

14. 一种服务器,其特征在于,包括处理器和存储器,所述存储器存储有能够被所述处理器执行的机器可执行指令,所述处理器执行所述机器可执行指令以实现权利要求1至10任一项所述的游戏任务的测试方法。

15. 一种计算机可读存储介质,其特征在于,所述计算机可读存储介质存储有计算机可执行指令,所述计算机可执行指令在被处理器调用和执行时,所述计算机可执行指令促使所述处理器实现权利要求1至10任一项所述的游戏任务的测试方法。

游戏任务的测试方法、装置、测试终端和服务端

技术领域

[0001] 本发明涉及游戏测试技术领域,尤其是涉及一种游戏任务的测试方法、装置、测试终端和服务端。

背景技术

[0002] 游戏开发过程中,游戏任务的回归测试是一个非常重要的环节。相关技术中,游戏任务的回归测试方式包括自动化测试和人工测试两种,该自动化测试方法需要测试人员理解游戏策划的游戏设计思路后,编写大量的自动化测试代码,然后通过该代码自动地在游戏中回归测试任务剧情,以寻找与游戏设计不符的问题,但是当游戏中的原任务被修改或者加入新任务时,自动化测试就可能无法正常运行,此时需要人工修改自动化测试代码或者改用人工测试的方式。

[0003] 上述人工测试方式需要测试人员先与游戏策划讨论确定策划的设计思路,然后人工手动地去不断测试回归游戏的任务剧情,找到与游戏设计不符或者影响剧情发展的问题。因此,无论是人工测试方式还是自动化测试方式,不仅需要对游戏有充分的理解,还需要进行大量的重复性的测试工作,导致测试工作需要消耗大量的人力和物力,且测试效率较低。

发明内容

[0004] 本发明的目的在于提供一种游戏任务的测试方法、装置、测试终端和服务端,以降低游戏任务测试的人力和物力,同时提高游戏任务的测试效率。

[0005] 第一方面,本发明实施例提供一种游戏任务的测试方法,该方法应用于测试终端;待测试的游戏任务预设有执行路径;该执行路径包括顺序排列的多个状态节点,以及相邻的两个状态节点之间的动作;该方法包括:接收游戏任务中的游戏角色的第一状态;从游戏任务的执行路径中确定第一状态对应的状态节点,以及第一状态对应的状态节点与下一个状态节点之间的目标动作;控制游戏角色执行目标动作,并获取执行目标动作后的游戏角色的第二状态;如果第二状态与下一个状态节点相匹配,将该第一状态更新为第二状态,继续执行从游戏任务的执行路径中确定第一状态对应的状态节点的步骤,直至到达执行路径的最后一个状态节点。

[0006] 在可选的实施方式中,上述游戏任务的执行路径,通过下述方式得到:初始化游戏任务的路径搜索图,循环执行下述目标操作,直至路径搜索图停止更新,从路径搜索图中确定游戏任务的执行路径;该目标操作包括:接收游戏任务中的游戏角色的第三状态,和游戏任务的描述信息;根据第三状态和任务描述信息,确定待执行动作;控制游戏角色执行待执行动作,并获取执行待执行动作后的游戏角色的第四状态;根据第三状态和待执行动作更新路径搜索图;将第四状态作为新的第三状态,继续执行确定当前执行动作的步骤,直至游戏任务执行完毕。

[0007] 在可选的实施方式中,上述从路径搜索图中确定执行路径的步骤,包括:如果从路

径搜索图的第一个状态节点到最后一个状态节点,存在多条路径,将包含有最少数量的状态节点的路径,确定为游戏任务的执行路径。

[0008] 在可选的实施方式中,上述根据第三状态和任务描述信息,确定待执行动作的步骤,包括:根据第三状态和任务描述信息,确定第三状态对应的动作空间;其中,该动作空间中包括:游戏角色在第三状态下的合法动作;从动作空间中确定待执行动作。

[0009] 在可选的实施方式中,上述从动作空间中确定待执行动作的步骤,包括:根据动作空间中各个合法动作在路径搜索图中所属的路径,以及各个合法动作在第三状态下已经被执行的次数,从各个合法动作中确定待执行动作。

[0010] 在可选的实施方式中,上述根据动作空间中各个合法动作在路径搜索图中所属的路径,以及各个合法动作在第三状态下已经被执行的次数,从各个合法动作中确定待执行动作的步骤,包括:针对动作空间中的每个合法动作,计算该合法动作的概率:

$$P_i = P_i^{\text{initial}} + \alpha * \sqrt{\log(\sum_{j=1}^n C_j) / C_i}; \text{将概率最大的合法动作,确定为待执行动作};$$

其中, P_i 为第*i*个合法动作的概率; P_i^{initial} 为第*i*个合法动作的先验概率,如果第*i*个合法动作属于游戏任务的路径搜索图中指定路径, $P_i^{\text{initial}}=1$,否则, $P_i^{\text{initial}} \in (0, 0.1)$; α 为预设的权重参数; C_i 为第*i*个合法动作在第三状态下已经被执行的次数, C_j 为动作空间中的第*j*个合法动作在第三状态下已经被执行的次数, n 为动作空间中合法动作的总数; \log 为以2为底的对数函数。

[0011] 在可选的实施方式中,上述指定路径包括:游戏任务的当前的路径搜索图中,从第一个状态节点到最后一个状态节点的、包含有最少数量的状态节点的路径。

[0012] 在可选的实施方式中,上述根据第三状态和待执行动作更新路径搜索图的步骤,包括:将第三状态更新为路径搜索图中的一个状态节点;将待执行动作更新为第三状态对应的状态节点与下一个状态节点之间的动作。

[0013] 在可选的实施方式中,上述方法还包括:如果第二状态与下一个状态节点不匹配,输出游戏任务的执行异常信息。

[0014] 在可选的实施方式中,上述测试终端分别与运行游戏任务的服务器和用户终端连接;上述接收游戏任务中的游戏角色的第一状态的步骤,包括:接收服务器发送的游戏任务中的游戏角色的第一状态;其中,该游戏任务中的游戏角色的第一状态,由用户终端发送至服务器;控制游戏角色执行目标动作的步骤,包括:将目标动作发送至服务器,以通过服务器控制用户终端中的游戏角色执行目标动作。

[0015] 第二方面,本发明实施例提供一种游戏任务的测试方法,该方法应用于运行有预设游戏任务的服务器;服务器分别与用于测试游戏任务的测试终端,以及运行有游戏任务的用户终端连接;该游戏任务预设有执行路径;该执行路径包括顺序排列的多个状态节点,以及相邻的两个状态节点之间的动作;该方法包括:向测试终端发送游戏任务中的游戏角色的第一状态,以通过测试终端从游戏任务的执行路径中,确定第一状态对应的状态节点,以及第一状态对应的状态节点与下一个状态节点之间的目标动作;如果接收到来自测试终端的目标动作,控制用户终端中的游戏角色执行目标动作,将执行目标动作后的游戏角色的第二状态发送至测试终端;如果第二状态与下一个状态节点相匹配,测试终端将第一状态更新为第二状态,继续执行从游戏任务的执行路径中确定第一状态对应的状态节点的步

骤,直至到达执行路径的最后一个状态节点。

[0016] 第三方面,本发明实施例提供一种游戏任务的测试装置,该装置设置于测试终端;待测试的游戏任务预设有执行路径;该执行路径包括顺序排列的多个状态节点,以及相邻的两个状态节点之间的动作;该装置包括:第一状态接收模块,用于接收游戏任务中的游戏角色的第一状态;动作确定模块,用于从游戏任务的执行路径中确定第一状态对应的状态节点,以及第一状态对应的状态节点与下一个状态节点之间的目标动作;第二状态接收模块,用于控制游戏角色执行目标动作,并获取执行目标动作后的游戏角色的第二状态;循环执行模块,用于如果第二状态与下一个状态节点相匹配,将该第一状态更新为第二状态,继续执行从游戏任务的执行路径中确定所述第一状态对应的状态节点的步骤,直至到达所述执行路径的最后一个状态节点。

[0017] 第四方面,本发明实施例提供一种游戏任务的测试装置,该装置设置于运行有预设游戏任务的服务器;该服务器分别与用于测试游戏任务的测试终端,以及运行有游戏任务的用户终端连接;该游戏任务预设有执行路径;该执行路径包括顺序排列的多个状态节点,以及相邻的两个状态节点之间的动作;该装置包括:第一状态发送模块,用于向测试终端发送游戏任务中的游戏角色的第一状态,以通过测试终端从游戏任务的执行路径中,确定第一状态对应的状态节点,以及第一状态对应的状态节点与下一个状态节点之间的目标动作;动作执行模块,用于如果接收到来自测试终端的目标动作,控制用户终端中的游戏角色执行目标动作,将执行目标动作后的游戏角色的第二状态发送至测试终端;如果第二状态与下一个状态节点相匹配,测试终端将第一状态更新为第二状态,继续执行从游戏任务的执行路径中确定第一状态对应的状态节点的步骤,直至到达执行路径的最后一个状态节点。

[0018] 第五方面,本发明实施例提供一种测试终端,包括处理器和存储器,该存储器存储有能够被处理器执行的机器可执行指令,处理器执行机器可执行指令以实现上述游戏任务的测试方法。

[0019] 第六方面,本发明实施例提供一种服务器,包括处理器和存储器,该存储器存储有能够被处理器执行的机器可执行指令,该处理器执行机器可执行指令以实现上述游戏任务的测试方法。

[0020] 第七方面,本发明实施例提供一种计算机可读存储介质,该计算机可读存储介质存储有计算机可执行指令,该计算机可执行指令在被处理器调用和执行时,该计算机可执行指令促使处理器实现上述游戏任务的测试方法。

[0021] 本发明实施例带来了以下有益效果:

[0022] 本发明提供了一种游戏任务的测试方法、装置、测试终端和服务器,首先接收游戏任务中的游戏角色的第一状态;该游戏任务预设有执行路径,该执行路径包括顺序排列的多个状态节点,以及相邻的两个状态节点之间的动作;进而从该游戏任务的执行路径中确定第一状态对应的状态节点,以及第一状态对应的状态节点与下一个状态节点之间的目标动作;再控制游戏角色执行目标动作,并获取执行目标动作后的游戏角色的第二状态;如果第二状态与下一个状态节点相匹配,将该第一状态更新为第二状态,继续执行从游戏任务的执行路径中确定第一状态对应的状态节点的步骤,直至到达执行路径的最后一个状态节点。该方式预先设置游戏任务的执行路径,在测试该游戏任务时,按照该执行路径执行各个

动作,以测试游戏角色是否能够按照该执行路径执行完毕,该方式无需人工介入,可自动化完成游戏任务的测试,从而降低了游戏任务测试的人力物力成本,同时提高了测试效率,且该方式可以适用各种游戏任务,游戏任务的测试效果较好。

[0023] 本发明的其他特征和优点将在随后的说明书中阐述,或者,部分特征和优点可以从说明书推知或毫无疑义地确定,或者通过实施本发明的上述技术即可得知。

[0024] 为使本发明的上述目的、特征和优点能更明显易懂,下文特举较佳实施方式,并配合所附附图,作详细说明如下。

附图说明

[0025] 为了更清楚地说明本发明具体实施方式或现有技术中的技术方案,下面将对具体实施方式或现有技术描述中所需要使用的附图作简单地介绍,显而易见地,下面描述中的附图是本发明的一些实施方式,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。

[0026] 图1为本发明实施例提供的一种游戏任务的测试方法的流程图;

[0027] 图2为本发明实施例提供的另一种游戏任务的测试方法的流程图;

[0028] 图3为本发明实施例提供的一种状态节点对应的动作空间的示意图;

[0029] 图4为本发明实施例提供的一种路径搜索图的结构示意图;

[0030] 图5为本发明实施例提供的另一种游戏任务的测试方法的流程图;

[0031] 图6为本发明实施例提供的另一种游戏任务的测试方法的流程图;

[0032] 图7为本发明实施例提供的一种游戏任务的测试装置的结构示意图;

[0033] 图8为本发明实施例提供的另一种游戏任务的测试装置的结构示意图;

[0034] 图9为本发明实施例提供的一种测试终端的结构示意图。

具体实施方式

[0035] 为使本发明实施例的目的、技术方案和优点更加清楚,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,所描述的实施例是本发明一部分实施例,而不是全部的实施例。通常在此处附图中描述和示出的本发明实施例的组件可以以各种不同的配置来布置和设计。

[0036] 因此,以下对在附图中提供的本发明的实施例的详细描述并非旨在限制要求保护的本发明的范围,而是仅仅表示本发明的选定实施例。基于本发明中的实施例,本领域普通技术人员在没有作出创造性劳动前提下所获得的所有其他实施例,都属于本发明保护的范围。

[0037] 目前游戏任务的回归测试主要是黑盒测试,分为自动化测试和人工测试两种,对于自动化测试,则是在了解游戏策划的设计思路后,测试人员通过编程的方式编写自动化测试的脚本代码,然后自动地在游戏中回归测试任务剧情,寻找与设计不符的问题;对于人工测试,测试人员会先与游戏策划讨论确定策划的设计思路,人工去不断测试回归游戏的任务剧情,找到与设计不符或者影响剧情发展的问题。

[0038] 由于游戏有数以千计的游戏任务,需要数百小时的时间回归整个游戏任务,同时,游戏又存在着迭代周期快,玩法复杂的特性,每天可能有3个版本的游戏任务需要回归;其

中,可能有大量的游戏任务需要人工测试回归或者编写修改测试脚本自动化测试回归,因此,测试人员需要不停的重复回归游戏任务,最终才能测试完成,确保游戏质量,提升玩家体验。

[0039] 而编程方式实现的自动化测试,效率低下、灵活性差,调整耗时耗力,并且很依赖测试人员自身的能力;而且在测试时只能按固定的规则测试游戏任务,无法对真实玩家行为进行模仿,只能覆盖游戏任务中部分测试情况,虽然测试效率高了,但效果不如人工测试好,所以在很多情况下,还是需要大量的人工测试的介入,这使得测试工作需要大量的人力和物力,并且要求所有测试人员都非常熟悉游戏,无形中增加了培训成本。因此,无论是人工测试方式还是自动化测试方式,不仅需要对游戏有充分的理解,还需要进行大量的重复性的测试工作,导致测试工作需要消耗大量的人力和物力,且测试效率较低。

[0040] 基于上述问题,本发明实施例提供了一种游戏任务的测试方法、装置、测试终端和服务器,该技术可以应用于新加入的游戏任务或者已有游戏任务的测试场景中,尤其是回归测试场景。为便于对本实施例进行理解,首先对本发明实施例所公开的一种游戏任务的测试方法进行详细介绍,该方法应用于测试终端;该测试终端通过运行有游戏任务的服务器与用户终端连接,该用户终端中运行有游戏任务中的游戏角色;待测试的游戏任务预设执行路径;该执行路径包括顺序排列的多个状态节点,以及相邻的两个状态节点之间的动作;如图1所示,该方法包括如下步骤:

[0041] 步骤S102,接收游戏任务中的游戏角色的第一状态。

[0042] 上述游戏任务可以是包含有预设的执行路径的游戏任务,也即是该游戏任务在之前的测试中已经规划出了执行路径。在具体实现时,该执行路径可以是在游戏任务第一次进行测试时得到的,在该游戏任务第一次进行测试时,需要控制游戏角色不断地与游戏任务进行交互,尝试完成游戏任务,在尝试完成游戏任务的过程中,在每个时间步,游戏角色都可以接收到游戏角色当前所处的状态信息,根据当前状态信息,游戏角色可以选择执行的动作,执行动作后跳转到游戏任务的下一个状态,不断循环整个过程直到游戏角色完成这个游戏任务,得到路径搜索图。

[0043] 该路径搜索图中包含有多个可完成游戏任务的路径,每条路径都包含有顺序排列的多个状态节点,以及相邻的两个状态节点之间的动作,通常可以用状态-动作树来表示。

[0044] 上述执行路径可以是路径搜索图中的任意一条可完成游戏任务的路径,也可以是完成游戏任务时路径最短的路径(也可称为最优路径)。该执行路径中各个状态节点的执行顺序和每两个状态节点之间的动作是固定好的,在测试已经预设执行路径的游戏任务时,首先需要根据游戏任务获取游戏角色在游戏任务对应的场景中的第一状态;该第一状态也即是游戏任务开始执行的初始状态,该游戏角色可以是游戏中的人物角色,可以是游戏中的飞船、舰艇、坦克等。

[0045] 步骤S104,从上述游戏任务的执行路径中确定第一状态对应的状态节点,以及该第一状态对应的状态节点与下一个状态节点之间的目标动作。

[0046] 由于游戏任务可以按照执行路径完成,上述第一状态可以是游戏任务的执行路径中的第一个状态节点对应的状态,也即是第一状态对应的状态节点为执行路径中的第一个状态节点。在执行路径中根据第一个状态节点与第二个状态节点(相当于上述下一个状态节点)之间的动作,可以得到目标动作。

[0047] 步骤S106,控制游戏角色执行上述目标动作,并获取执行该目标动作后的游戏角色的第二状态。

[0048] 当得到第一状态对应的状态节点与下一个状态节点之间的目标动作时,控制游戏角色执行该目标动作;游戏角色执行完该目标动作后,游戏任务会被推进,游戏角色所处的环境也会发生变化,以使该游戏角色处于新的状态,该新的状态也即是上述第二状态。

[0049] 步骤S108,如果第二状态与下一个状态节点相匹配,将该第一状态更新为第二状态,继续执行从游戏任务的执行路径中确定第一状态对应的状态节点的步骤,直至到达执行路径的最后一个状态节点。

[0050] 在测试过程中,需要判断第二状态是否与执行路径中的第一个状态节点的下一个状态节点相匹配,也即是判断执行路径中的下一个状态节点是否为第二状态对应的节点,如果是,将第一状态更新为第二状态,继续执行上述步骤S104,直到达到执行路径的最后一个状态节点。

[0051] 在具体实现时,如果第二状态与下一个状态节点相匹配,从游戏任务的执行路径中确定第二状态对应的状态节点,以及该第二状态对应的状态节点与下一个状态节点之间的目标动作;控制游戏角色执行目标动作,并获取执行该目标动作后的游戏角色的状态,如果该状态与下一个状态节点相匹配,继续确定该目标动作后的游戏角色的状态对应的状态,直到到达执行路径的最后一个状态节点。

[0052] 当到达执行路径的最后一个状态节点时,表明该游戏任务可以按照预设的执行路径完成游戏任务,该游戏任务正常,不存在漏洞。如果不能到达执行路径的最后一个状态节点,也可以理解为第二状态与下一个状态节点不匹配时,表明游戏任务无法按照预设的执行路径完成,游戏任务发生了料之外的变化,从而可以发现游戏测试中存在的漏洞。

[0053] 本发明提供了一种游戏任务的测试方法,首先接收游戏任务中的游戏角色的第一状态;该游戏任务预设执行路径,该执行路径包括顺序排列的多个状态节点,以及相邻的两个状态节点之间的动作;进而从该游戏任务的执行路径中确定第一状态对应的状态节点,以及第一状态对应的状态节点与下一个状态节点之间的目标动作;再控制游戏角色执行目标动作,并获取执行目标动作后的游戏角色的第二状态;如果第二状态与下一个状态节点相匹配,将该第一状态更新为第二状态,继续执行从游戏任务的执行路径中确定第一状态对应的状态节点的步骤,直至到达执行路径的最后一个状态节点。该方式预先设置游戏任务的执行路径,在测试该游戏任务时,按照该执行路径执行各个动作,以测试游戏角色是否能够按照该执行路径执行完毕,该方式无需人工介入,可自动化完成游戏任务的测试,从而降低了游戏任务测试的人力物力成本,同时提高了测试效率,且该方式可以适用各种游戏任务,游戏任务的测试效果较好。

[0054] 本发明实施例还提供另一种游戏任务的测试方法,该方法在上述实施例所述方法的基础上实现;该方法重点描述初始化游戏任务的路径搜索图的具体过程,以及从路径搜索图中确定游戏任务的执行路径的具体过程;如图2所示,该方法包括如下步骤:

[0055] 步骤S202,接收游戏任务中的游戏角色的第三状态,和游戏任务的描述信息。

[0056] 初始化游戏任务的路径搜索图的过程,也可以理解为对游戏任务第一次进行测试的过程,该游戏任务可以是新加入的游戏任务,也可以是对之前的游戏任务进行修改后的任务;在游戏任务第一次进行测试时,需要测试所有能完成该游戏任务的方式(相当于所有

的路径)是否都可以顺利运行,也即是需要得到完善的路径搜索图,如果所有路径都可以顺利运行,表明该游戏任务正常。

[0057] 上述第三状态为游戏任务中游戏角色当前所处的状态;上述游戏任务的任务描述信息可以包括描述该游戏任务的可实现功能的信息,以及每个游戏状态对应的参数信息等。

[0058] 步骤S204,根据上述第三状态和任务描述信息,确定待执行动作。

[0059] 在具体实现时,根据游戏角色当前所处的第三状态,从游戏任务的任务描述信息中提取有效的参数信息,将该参数信息和第三状态进行组合,可提取出该第三状态对应待执行动作,例如,第三状态为击杀了怪物,根据游戏任务的任务描述信息中的相应描述,确定待执行动作为逃跑到A点。

[0060] 在具体实现时,上述步骤S204可以通过下述步骤10-11实现:

[0061] 步骤10,根据第三状态和任务描述信息,确定第三状态对应的动作空间;其中,该动作空间中包括:游戏角色在第三状态下的合法动作。

[0062] 上述第三状态通常为游戏角色在游戏任务中的状态信息,从该状态信息中可提取出有效特征,并组合成一个当前状态节点,针对该状态信息可在游戏任务的任务描述信息中提取有效的参数信息,将参数信息和状态信息组合提取出当前状态节点的一个合法的动作空间;该动作空间中,包括有在当前状态节点下的可执行的所有的合法动作。通常每个状态节点都对应有一个动作空间,每个动作空间中至少包括一个合法动作,如图3所示为动作空间的示意图,图中S表示状态节点,A1、A2、…、An表示该状态节点S的动作空间中的合法动作。

[0063] 步骤11,从上述动作空间中确定待执行动作。

[0064] 由于在动作空间中通常包含有至少一个合法动作,需要根据一定的规则,从合法动作中选择一个动作,作为待执行动作。在具体实现时,可以根据动作空间中各个合法动作在路径搜索图中所属的路径,以及各个合法动作在第三状态下已经被执行的次数,从各个合法动作中确定待执行动作。

[0065] 上述路径搜索图中包含有多个路径,随着游戏角色与游戏任务不断的进行交互,路径搜索图也随之进行更新,在更新的过程中会从路径搜索图中确定一个路径为指定路径,该指定路径可以是最优路径,也可以是满足其他预设条件的路径。在从动作空间中确定待执行动作时,可以对动作空间中的每个合法动作依次进行判断,也即是针对每个合法动作,查看当前合法动作是否为当前路径搜索图中的指定路径中的动作,以及在第三状态下,当前合法动作在当前路径搜索图中的历史执行次数,根据查看结果从动作空间中的确定待执行动作。具体地,可以通过下述步骤20-21实现:

[0066] 步骤20,针对动作空间中的每个合法动作,计算该合法动作的概率:

$$[0067] \quad P_i = P_i^{\text{initial}} + \alpha * \sqrt{\log(\sum_{j=1}^n C_j) / C_i}$$

[0068] 其中, P_i 为第*i*个合法动作的概率; P_i^{initial} 为第*i*个合法动作的先验概率,如果第*i*个合法动作属于游戏任务的路径搜索图中指定路径, $P_i^{\text{initial}}=1$,否则, $P_i^{\text{initial}} \in (0, 0.1)$; α 为预设的权重参数; C_i 为第*i*个合法动作在第三状态下已经被执行的次数, C_j 为动作空间中的第*j*个合法动作在第三状态下已经被执行的次数, n 为动作空间中合法动作的总数; \log 为

以2为底的对数函数。

[0069] 上述 α 为预设的权重参数,可以根据测试人员需求设置,例如,可以设置为1.4,也可以设置为其他值。通常该值越小则说明越倾向于利用指定路径选择待执行动作,该值越大则越倾向于利用当前采取动作的情况来选择待执行动作。

[0070] 根据上述计算合法动作的概率的公式可知,越经常被执行的合法动作,本次被执行的概率越低,因此该公式更倾向于选择那些被执行次数少的合法动作,从而易于覆盖所有的路径。

[0071] 步骤21,将概率最大的合法动作,确定为待执行动作。

[0072] 在一些实施例中,上述指定路径可以包括:游戏任务的当前的路径搜索图中,从第一个状态节点到最后一个状态节点的、包含有最少数量的状态节点的路径。

[0073] 在当前的路径搜索图中,可完成游戏任务的路径中,包含有状态节点的数量最少的路径,可以确定为指定路径,通常包含有的状态节点最少也可以为执行时间最短或者路径最短。如图4所示的路径搜索图中,指定路径可以为状态节点A、动作1、状态节点B、动作2、状态节点C、动作4、状态节点D组成的路径。

[0074] 在具体实现时,可以利用动态规划算法,从当前的路径搜索图中规划出指定路径。该动态规划算法通常是通过拆分问题,定义问题状态和状态之间的关系,使得问题能够以递推的方式去解决。

[0075] 步骤S206,控制游戏角色执行上述待执行动作,并获取执行待执行动作后的游戏角色的第四状态。

[0076] 上述第四状态为游戏角色执行完待执行动作后,在游戏任务中所处的状态,也可以理解为游戏角色执行完待执行动作后,游戏任务将会被推进,从而改变了游戏角色当前所处的环境,也即是游戏角色进入了一个新的时间点,该时间点对应的状态即为第四状态。

[0077] 步骤S208,根据上述第三状态和待执行动作更新路径搜索图;将该第四状态作为新的第三状态,继续执行确定当前执行动作的步骤,直至游戏任务执行完毕。

[0078] 根据第三状态对应的状态节点和待执行动作,更新路径搜索图,例如,如果当前的路径搜索图中没有第三状态节点对应的状态节点时,将该状态节点以及该状态节点对应的待执行动作,添加至路径搜索途中;如果该状态节点为已经存在的节点,可以保存该节点与待执行动作的关系,以确定该节点与后续节点的关系,以完成该节点对应的整个路径。

[0079] 将上述第三状态更新为第四状态,继续执行根据第三状态和任务描述信息,确定待执行动作的步骤,直到游戏任务完成。游戏任务完成后,可以得到完成游戏任务的一条路径,该路径也是路径搜索图中的一条路径。

[0080] 在具体实现时,上述步骤S208中根据第三状态和待执行动作更新路径搜索图的步骤可以通过下述步骤30-31实现:

[0081] 步骤30,将第三状态更新为路径搜索图中的一个状态节点。

[0082] 步骤31,将待执行动作更新为第三状态对应的状态节点与下一个状态节点之间的动作。

[0083] 步骤S210,判断上述路径搜索图是否停止更新;如果是,执行步骤S212,否则,继续执行上述步骤S202。

[0084] 在具体实现时,如果路径搜索图中的所有状态节点和所有状态节点之间的动作都

被遍历到了,则停止更新,也即是路径搜索图已经建立完全,已经包含了所有的可完成游戏任务的路径。

[0085] 步骤S212,从上述路径搜索图中确定游戏任务的执行路径。

[0086] 在具体实现时,上述步骤S212可以通过下述方式确定执行路径:如果从路径搜索图的第一个状态节点到最后一个状态节点,存在多条路径,将包含有最少数量的状态节点的路径,确定为游戏任务的执行路径。也即是将建立完全的路径搜索图中包含最少数量的状态节点的路径,确定下一次测试该游戏任务的执行路径。通常包含有的状态节点最少也可以为执行时间最短或者路径最短,当路径搜索图建立完全时,该执行路径也可以成为全局最优路径,在具体实现时,可以利用动态规划算法,从当前的路径搜索图中规划出指定路径。

[0087] 上述方式中,游戏角色不断地与游戏任务对应的环境进行交互,以尝试完成游戏任务,从而可以更好的提高游戏任务测试中的状态覆盖率,同时在测试的过程中构建和更新关于游戏任务的路径搜索图,最终游戏角色可自主的学习完成游戏任务;同时该方式可以使用规划技术对路径搜索图进行规划,获得完成游戏任务的最优路径(相当于上述执行路径)。这样可以无需人工介入,游戏角色自主学会游戏任务的完成,测试了游戏任务剧情的流程是否正常,是否存在极端情况下游戏任务被不正常的完成了。

[0088] 该极端情况通常是指策划人员设计游戏任务完成时的漏洞,例如,策划人员设计游戏角色需要击杀30个怪,再逃跑到a点才能完成游戏任务,但是实际游戏过程中只需要跑到a点就能完成了游戏任务。

[0089] 上述游戏任务的测试方法,能够对游戏任务进行智能测试,且无需测试人员对游戏有充分的理解,也无需要进行大量的重复性的测试工作,从而提高了测试效率,也大幅降低了游戏测试人员的工作量,也降低了对开发者的整体能力需求,同时能使得测试效率更高、效果更好。

[0090] 本发明实施例还提供另一种游戏任务的测试方法,该方法在上述实施例所述方法的基础上实现;该方法重点描述在测试终端分别与运行游戏任务的服务器和用户终端连接时,实现游戏测试的具体过程;如图5所示,该方法应用于测试终端,该方法包括如下步骤:

[0091] 步骤S502,接收服务器发送的游戏任务中的游戏角色的第一状态;其中,游戏任务中的游戏角色的第一状态,由用户终端发送至服务器。

[0092] 当服务器接收到一个待测试的游戏任务时,服务器可以根据用户终端发送的在该游戏任务中游戏角色所处的第一状态,将通过远程过程调用(PRC)的方式,将该第一状态,以及该第一状态对应的游任务描述信息发送给测试终端。

[0093] 步骤S504,从上述游戏任务的执行路径中确定第一状态对应的状态节点,以及该第一状态对应的状态节点与下一个状态节点之间的目标动作。

[0094] 在一些实施例中,如果待测试的游戏任务为第一测试,可以根据第一状态和该第一状态对应的任务描述信息,确定第一状态对应的状态节点对应的动作空间,并从动作空间中选择概率最大的合法动作作为目标动作。

[0095] 本实施例中,待测试的游戏任务为预设有的执行路径的游戏任务,因此,可以直接将第一状态对应的状态节点与下一个状态节点之间的动作,确定为目标动作。

[0096] 步骤S506,将上述目标动作发送至服务器,以通过服务器控制用户终端中的游戏

角色执行该目标动作。

[0097] 测试终端可以通过远程过程调用的方式,将目标动作发送至服务器,以使服务器将该目标动作发送给用户终端,以用户终端中游戏角色指定该目标动作。

[0098] 步骤S508,获取执行该目标动作后的游戏角色的第二状态。

[0099] 当游戏角色执行完目标动作后,用户终端会将该游戏角色所处的第二状态发送给服务器,以使服务器将该第二状态,以及该第二状态对应的任务描述信息发送给测试终端。

[0100] 步骤S510,判断上述第二状态与下一个状态节点是否相匹配;如果是,执行步骤S512;否则,执行步骤S514。

[0101] 测试终端接收到第二状态后,将判断该第二状态是否与指定路径中的下一个状态节点相匹配。

[0102] 步骤S512,将该第一状态更新为第二状态,继续执行从游戏任务的执行路径中确定第一状态对应的状态节点的步骤,直至到达执行路径的最后一个状态节点。

[0103] 步骤S514,输出上述游戏任务的执行异常信息。

[0104] 上述异常信息通常包括游戏任务的日志、运行截图和相应的运行数据等,根据该异常信息可以确定游戏任务中出现漏洞的位置,以通知测试人员关注该漏洞。

[0105] 上述游戏任务的测试方法,在对预设有执行路径的游戏任务进行测试时,只需要判断该游戏任务是否可以按照该执行路径完成,如果可以,确定该游戏任务正常,如果不可以,则确定游戏任务中存在漏洞。该方式无需人工介入,节省了大量的人力和物力,而且该测试可以自动完成,提高了测试效率,而且该方式可以达到甚至超过人工测试的效果。

[0106] 针对于上述游戏任务的测试方法的实施例,本发明实施例还提供了另一种游戏任务的测试方法,该方法应用于运行有预设游戏任务的服务器;该服务器分别与用于测试游戏任务的测试终端,以及运行有游戏任务的用户终端连接;该游戏任务预设有执行路径,该执行路径包括顺序排列的多个状态节点,以及相邻的两个状态节点之间的动作;如图6所示,该方法包括如下步骤:

[0107] 步骤S602,向测试终端发送游戏任务中的游戏角色的第一状态,以通过该测试终端从游戏任务的执行路径中,确定第一状态对应的状态节点,以及该第一状态对应的状态节点与下一个状态节点之间的目标动作。

[0108] 在具体实现时,服务器可以通过远程过程调用(简称PRC)的方式,向测试终端发送数据。

[0109] 步骤S604,如果接收到来自测试终端的目标动作,控制用户终端中的游戏角色执行该目标动作,将执行目标动作后的游戏角色的第二状态发送至测试终端;如果第二状态与下一个状态节点相匹配,测试终端将该第一状态更新为第二状态,继续执行从游戏任务的执行路径中确定第一状态对应的状态节点的步骤,直至到达执行路径的最后一个状态节点。

[0110] 上述游戏任务的测试方法,通过游戏角色与游戏任务不断交互得到执行路径,并通过该执行路径测试游戏任务,该方式无需人工介入,可自动化完成游戏任务的测试,从而提高了测试效率,且该方式可以适用各种游戏任务,保证了游戏任务的测试效果。

[0111] 对应于上述方法实施例,本发明实施例提供了一种游戏任务的测试装置,该装置设置于测试终端;待测试的游戏任务预设有执行路径;该执行路径包括顺序排列的多个状

态节点,以及相邻的两个状态节点之间的动作;如图7所示,该装置包括:

[0112] 第一状态接收模块70,用于接收游戏任务中的游戏角色的第一状态。

[0113] 动作确定模块71,用于从游戏任务的执行路径中确定第一状态对应的状态节点,以及第一状态对应的状态节点与下一个状态节点之间的目标动作。

[0114] 第二状态接收模块72,用于控制游戏角色执行目标动作,并获取执行目标动作后的游戏角色的第二状态。

[0115] 循环执行模块73,用于如果第二状态与下一个状态节点相匹配,将第一状态更新为第二状态,继续执行从游戏任务的执行路径中确定第一状态对应的状态节点的步骤,直至到达执行路径的最后一个状态节点。

[0116] 上述游戏任务的测试装置,首先接收游戏任务中的游戏角色的第一状态;该游戏任务预设执行路径,该执行路径包括顺序排列的多个状态节点,以及相邻的两个状态节点之间的动作;进而从该游戏任务的执行路径中确定第一状态对应的状态节点,以及第一状态对应的状态节点与下一个状态节点之间的目标动作;再控制游戏角色执行目标动作,并获取执行目标动作后的游戏角色的第二状态;如果第二状态与下一个状态节点相匹配,将该第一状态更新为第二状态,继续执行从游戏任务的执行路径中确定第一状态对应的状态节点的步骤,直至到达执行路径的最后一个状态节点。该方式预先设置游戏任务的执行路径,在测试该游戏任务时,按照该执行路径执行各个动作,以测试游戏角色是否能够按照该执行路径执行完毕,该方式无需人工介入,可自动化完成游戏任务的测试,从而降低了游戏任务测试的人力物力成本,同时提高了测试效率,且该方式可以适用各种游戏任务,游戏任务的测试效果较好。

[0117] 具体地,上述装置还包括路径确定模块,用于:初始化游戏任务的路径搜索图,循环执行下述目标操作,直至路径搜索图停止更新,从路径搜索图中确定游戏任务的执行路径;该目标操作包括:第三状态接收模块,用于接收游戏任务中的游戏角色的第三状态,和游戏任务的任务描述信息;指定动作确定模块,用于根据第三状态和任务描述信息,确定待执行动作;第四状态接收模块,用于控制游戏角色执行待执行动作,并获取执行待执行动作后的游戏角色的第四状态;更新模块,用于根据第三状态和待执行动作更新路径搜索图;将第四状态作为新的第三状态,继续执行确定当前执行动作的步骤,直至游戏任务执行完毕。

[0118] 进一步地,上述路径确定模块,还用于:如果从路径搜索图的第一个状态节点到最后一个状态节点,存在多条路径,将包含最少数量的状态节点的路径,确定为游戏任务的执行路径。

[0119] 进一步地,上述执行动作确定模块,包括:动作空间确定单元,用于根据第三状态和任务描述信息,确定第三状态对应的动作空间;其中,该动作空间中包括:游戏角色在第三状态下的合法动作;动作选取单元,用于从动作空间中确定待执行动作。

[0120] 进一步地,上述动作选取单元,用于:根据动作空间中各个合法动作在路径搜索图中所属的路径,以及各个合法动作在第三状态下已经被执行的次数,从各个合法动作中确定待执行动作。

[0121] 具体地,上述动作选取单元,还用于:针对动作空间中的每个合法动作,计算该合

法动作的概率: $P_i = P_i^{\text{initial}} + \alpha * \sqrt{\log(\sum_{j=1}^n C_j) / C_i}$; 将概率最大的合法动作,确定为

待执行动作;其中, P_i 为第*i*个合法动作的概率; $P_i^{initial}$ 为第*i*个合法动作的先验概率,如果第*i*个合法动作属于游戏任务的路径搜索图中指定路径, $P_i^{initial}=1$,否则, $P_i^{initial} \in (0, 0.1)$; α 为预设的权重参数; C_i 为第*i*个合法动作在第三状态下已经被执行的次数, C_j 为动作空间中的第*j*个合法动作在第三状态下已经被执行的次数, n 为动作空间中合法动作的总数; \log 为以2为底的对数函数。

[0122] 在具体实现时,上述指定路径包括:游戏任务的当前的路径搜索图中,从第一个状态节点到最后一个状态节点的、包含有最少数量的状态节点的路径。

[0123] 进一步地,上述更新模块,用于:将第三状态更新为路径搜索图中的一个状态节点;将待执行动作更新为第三状态对应的状态节点与下一个状态节点之间的动作。

[0124] 进一步地,上述装置还包括异常信息输出模块,用于:如果第二状态与下一个状态节点不匹配,输出游戏任务的执行异常信息。

[0125] 在具体实现时,上述测试终端分别与运行游戏任务的服务器和用户终端连接;上述第一状态接收模块70,用于:接收服务器发送的游戏任务中的游戏角色的第一状态;其中,游戏任务中的游戏角色的第一状态,由用户终端发送至服务器;上述第二状态接收模块72,用于:将目标动作发送至服务器,以通过服务器控制用户终端中的游戏角色执行目标动作。

[0126] 本发明实施例所提供的游戏任务的测试装置,其实现原理及产生的技术效果和前述方法实施例相同,为简要描述,装置实施例部分未提及之处,可参考前述方法实施例中相应内容。

[0127] 对应于上述方法实施例,本发明实施例提供了另一种游戏任务的测试装置,该装置设置于运行有预设游戏任务的服务器;该服务器分别与用于测试游戏任务的测试终端,以及运行有游戏任务的用户终端连接;该游戏任务预设有执行路径;执行路径包括顺序排列的多个状态节点,以及相邻的两个状态节点之间的动作;如图8所示,该装置包括:

[0128] 第一状态发送模块80,用于向测试终端发送游戏任务中的游戏角色的第一状态,以通过测试终端从游戏任务的执行路径中,确定第一状态对应的状态节点,以及第一状态对应的状态节点与下一个状态节点之间的目标动作。

[0129] 动作执行模块81,用于如果接收到来自测试终端的目标动作,控制用户终端中的游戏角色执行目标动作,将执行目标动作后的游戏角色的第二状态发送至测试终端;如果第二状态与下一个状态节点相匹配,测试终端将第一状态更新为第二状态,继续执行从游戏任务的执行路径中确定第一状态对应的状态节点的步骤,直至到达执行路径的最后一个状态节点。

[0130] 上述游戏任务的测试装置,无需人工介入,可自动化完成游戏任务的测试,从而提高了测试效率较低,且该方式可以适用各种游戏任务,保证了游戏任务的测试效果。

[0131] 本发明实施例还提供了一种测试终端,用于运行上述游戏任务的测试方法,参见图9所示,该测试终端包括处理器101和存储器100,该存储器100存储有能够被处理器101执行的机器可执行指令,处理器101执行该机器可执行指令以实现上述游戏任务的测试方法。

[0132] 进一步地,图9所示的测试终端还包括总线102和通信接口103,处理器101、通信接口103和存储器100通过总线102连接。

[0133] 其中,存储器100可能包含高速随机存取存储器(RAM, Random Access Memory),也可

能还包括非不稳定的存储器 (non-volatile memory), 例如至少一个磁盘存储器。通过至少一个通信接口 103 (可以是有线或者无线) 实现该系统网元与至少一个其他网元之间的通信连接, 可以使用互联网, 广域网, 本地网, 城域网等。总线 102 可以是 ISA 总线、PCI 总线或 EISA 总线等。所述总线可以分为地址总线、数据总线、控制总线等。为便于表示, 图 9 中仅用一个双向箭头表示, 但并不表示仅有一根总线或一种类型的总线。

[0134] 处理器 101 可能是一种集成电路芯片, 具有信号的处理能力。在实现过程中, 上述方法的各步骤可以通过处理器 101 中的硬件的集成逻辑电路或者软件形式的指令完成。上述的处理器 101 可以是通用处理器, 包括中央处理器 (Central Processing Unit, 简称 CPU)、网络处理器 (Network Processor, 简称 NP) 等; 还可以是数字信号处理器 (Digital Signal Processor, 简称 DSP)、专用集成电路 (Application Specific Integrated Circuit, 简称 ASIC)、现场可编程门阵列 (Field-Programmable Gate Array, 简称 FPGA) 或者其他可编程逻辑器件、分立门或者晶体管逻辑器件、分立硬件组件。可以实现或者执行本发明实施例中的公开的各方法、步骤及逻辑框图。通用处理器可以是微处理器或者该处理器也可以是任何常规的处理器等。结合本发明实施例所公开的方法的步骤可以直接体现为硬件译码处理器执行完成, 或者用译码处理器中的硬件及软件模块组合执行完成。软件模块可以位于随机存储器, 闪存、只读存储器, 可编程只读存储器或者电可擦写可编程存储器、寄存器等本领域成熟的存储介质中。该存储介质位于存储器 100, 处理器 101 读取存储器 100 中的信息, 结合其硬件完成前述实施例的方法的步骤。

[0135] 本发明实施例还提供了一种服务器, 该服务器包括处理器和存储器, 存储器存储有能够被处理器执行的机器可执行指令, 该处理器执行机器可执行指令以实现上述游戏任务的测试方法, 该服务器的具体结构可以参考上述测试终端, 在此不再赘述。

[0136] 本发明实施例还提供了一种计算机可读存储介质, 该计算机可读存储介质存储有计算机可执行指令, 该计算机可执行指令在被处理器调用和执行时, 该计算机可执行指令促使处理器实现上述游戏任务的测试方法, 具体实现可参见方法实施例, 在此不再赘述。

[0137] 所属领域的技术人员可以清楚地了解到, 为描述的方便和简洁, 上述描述的装置和/或电子设备的具体工作过程, 可以参考前述方法实施例中的对应过程, 在此不再赘述。

[0138] 最后应说明的是: 以上所述实施例, 仅为本发明的具体实施方式, 用以说明本发明的技术方案, 而非对其限制, 本发明的保护范围并不局限于此, 尽管参照前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: 任何熟悉本技术领域的技术人员在本发明揭露的技术范围内, 其依然可以对前述实施例所记载的技术方案进行修改或可轻易想到变化, 或者对其中部分技术特征进行等同替换; 而这些修改、变化或者替换, 并不使相应技术方案的本质脱离本发明实施例技术方案的精神和范围, 都应涵盖在本发明的保护范围之内。因此, 本发明的保护范围应所述以权利要求的保护范围为准。

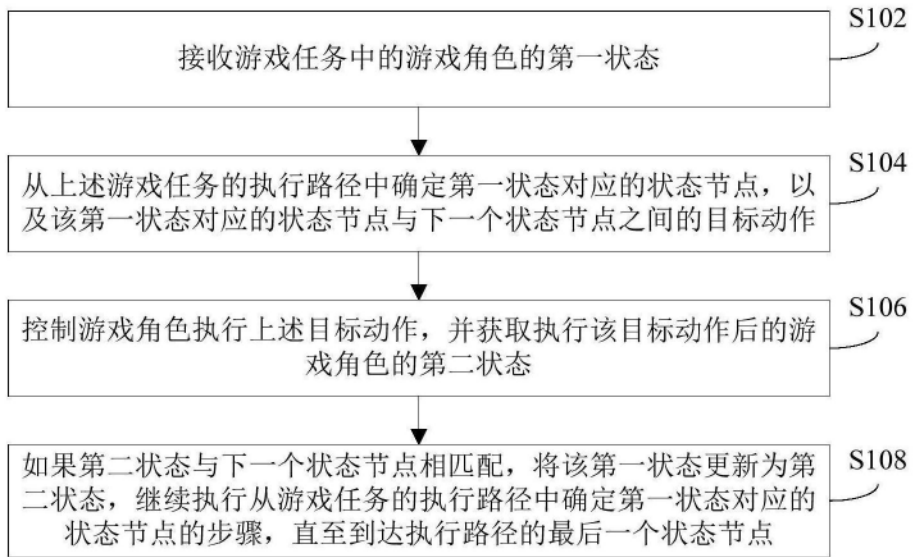


图1

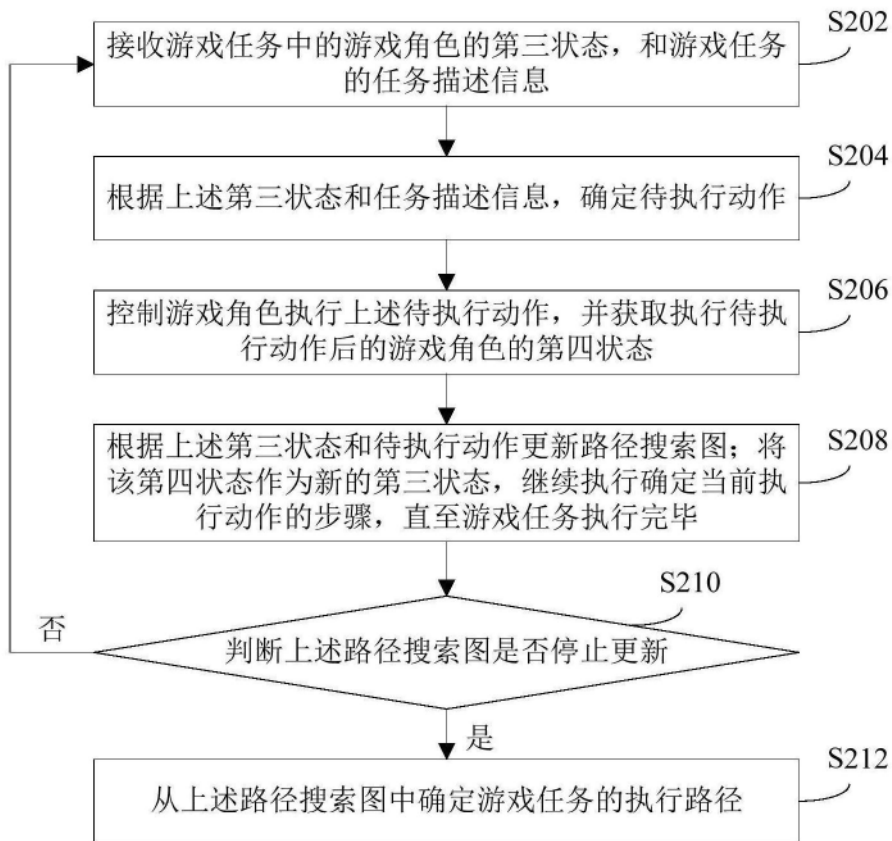


图2

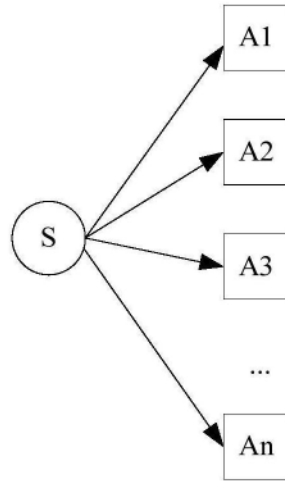


图3

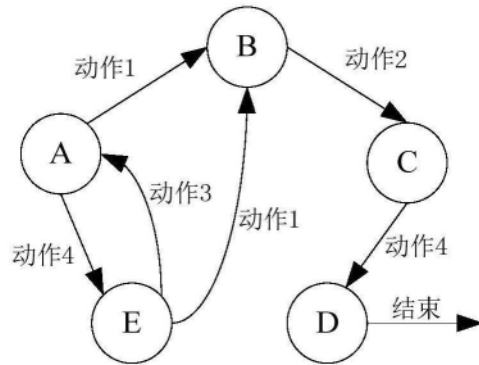


图4

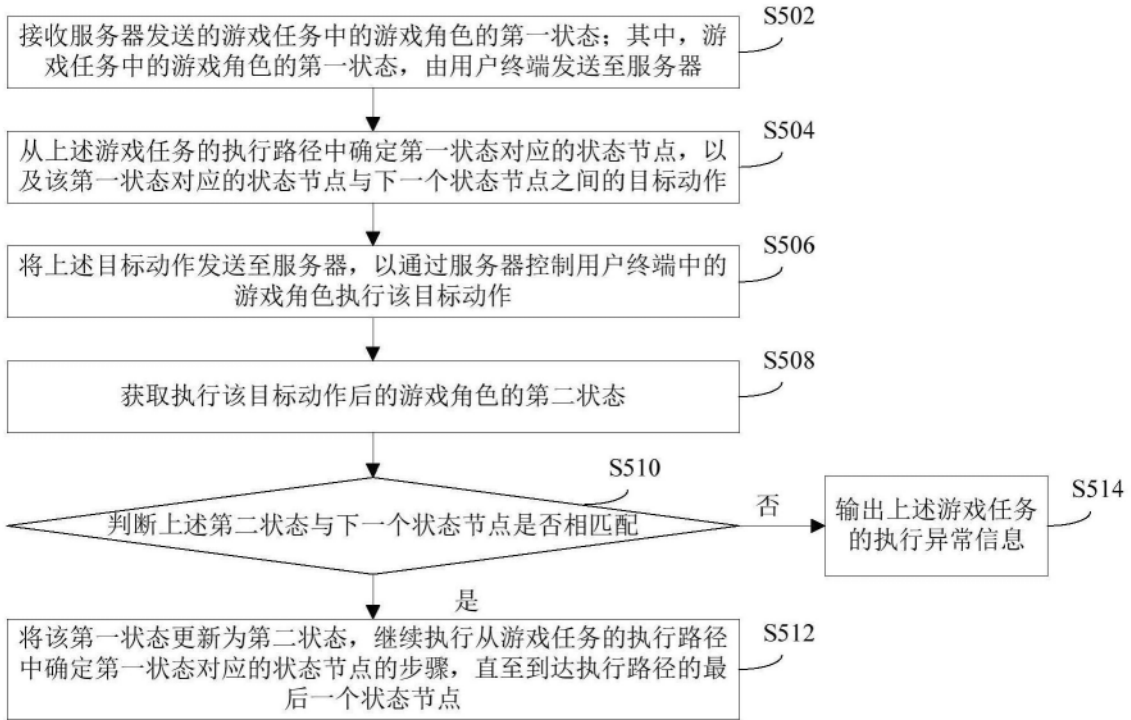


图5

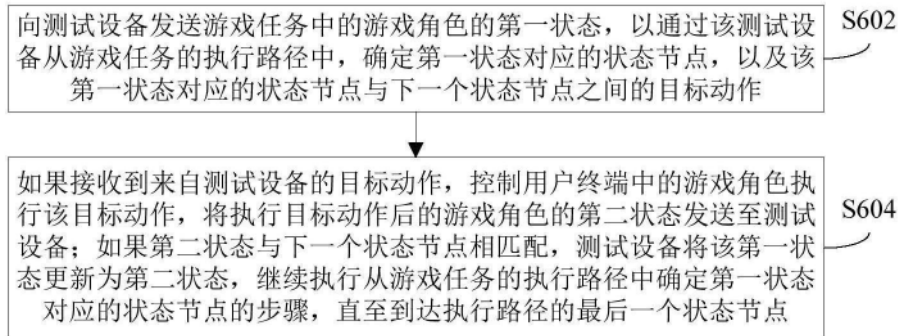


图6



图7



图8

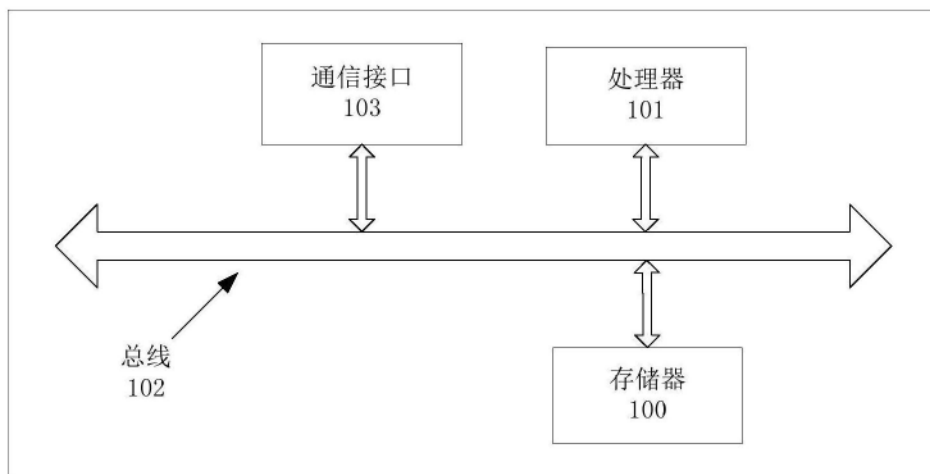


图9