



(12) 发明专利申请

(10) 申请公布号 CN 104582807 A

(43) 申请公布日 2015. 04. 29

(21) 申请号 201380043456. X

(51) Int. Cl.

(22) 申请日 2013. 06. 26

A63F 13/35(2014. 01)

(30) 优先权数据

A63F 13/69(2014. 01)

20125762 2012. 06. 29 FI

A63F 13/47(2014. 01)

(85) PCT国际申请进入国家阶段日

2015. 02. 15

(86) PCT国际申请的申请数据

PCT/FI2013/050702 2013. 06. 26

(87) PCT国际申请的公布数据

W02014/001639 EN 2014. 01. 03

(71) 申请人 玩族工作室

地址 芬兰赫尔辛基

(72) 发明人 维勒 - 卡勒 · 奥伯尼恩

蒂莫 · 卡马瑞纳恩

(74) 专利代理机构 北京信慧永光知识产权代理

有限责任公司 11290

代理人 曹正建 陈桂香

权利要求书2页 说明书9页 附图4页

(54) 发明名称

非线性游戏玩法的改进配置

(57) 摘要

一种改进的用于向两个以上的用户提供交互

1. 一种用于向两个以上的用户提供交互式游戏事件的计算机实施方法,所述计算机实施方法包括:

向所述用户分配游戏人物;

在所述游戏事件中发起一个以上的子过程,所述一个以上的子过程被配置为在单独的状态机中独立地进行,每个所述状态机响应于交互功能的特定集合;

针对每个所述子过程,选择游戏人物组,其中,针对至少两个同时进行的子过程选择至少一个游戏人物;

向所选择的游戏人物提供准许交互功能的集合以进行激活,所述准许交互功能的集合来自于在选择所选择的游戏人物时所针对的每个子过程的所述交互功能的特定集合;

响应于所选择的游戏人物在正在进行的子过程中激活它的所述准许交互功能的集合中的交互功能,改变该正在进行的子过程的状态;

响应于该正在进行的子过程的状态改变,更新该子过程中的所选择的游戏人物的所述准许交互功能的集合。

2. 根据权利要求 1 所述的计算机实施方法,其特征在于,所述游戏事件具有定义的开始点和结束点。

3. 根据权利要求 2 所述的计算机实施方法,其特征在于,所述定义的开始点和结束点中的至少一者是定时的。

4. 根据权利要求 1 至 3 中任一项所述的计算机实施方法,其特征在于,

向针对所述子过程选择的所述游戏人物分配游戏角色;并且

应用所述游戏角色,以限制所述交互功能的对所述游戏人物的可用性。

5. 根据权利要求 1 至 4 中任一项所述的计算机实施方法,其特征在于,根据预定的调度,发起所述子过程中的至少一部分。

6. 一种游戏服务器,其包括处理单元,所述处理单元被配置为:

向所述用户分配游戏人物;

在所述游戏事件中发起一个以上的子过程,所述一个以上的子过程被配置为在单独的状态机中独立地进行,每个所述状态机响应于交互功能的特定集合;

针对每个所述子过程,选择游戏人物组,其中,针对至少两个同时进行的子过程选择至少一个游戏人物;

向所选择的游戏人物提供准许交互功能的集合以进行激活,所述准许交互功能的集合来自于在选择所选择的游戏人物时所针对的每个子过程的所述交互功能的特定集合;

响应于所选择的游戏人物在正在进行的子过程中激活它的所述准许交互功能的集合中的交互功能,改变该正在进行的子过程的状态;

响应于该正在进行的子过程的状态改变,更新该子过程中的所选择的游戏人物的所述准许交互功能的集合。

7. 根据权利要求 6 所述的游戏服务器,其特征在于,所述处理单元被配置为:

向针对所述子过程选择的所述游戏人物分配游戏角色;并且

应用所述游戏角色,以限制所述交互功能的对所述游戏人物的可用性。

8. 根据权利要求 6 或 7 所述的游戏服务器,其特征在于,包括规则集,所述规则集包括所述游戏人物的所述交互功能在所述子过程的各种状态下的可用性的更新规则。

9. 根据权利要求6至8中任一项所述的游戏服务器,其特征在于,所述处理单元被配置为根据预定的调度发起所述子过程中的至少一部分。

10. 一种计算机可读的计算机程序产品,其在计算机装置中对用于执行权利要求1-5中任一项所述的方法的指令进行译码以向两个以上的用户提供交互游戏事件。

---

## 非线性游戏玩法的改进配置

### 技术领域

[0001] 本发明涉及多人视频游戏,更具体地涉及如独立权利要求的前序部分限定的计算机实施方法、设备和计算机程序产品。

### 背景技术

[0002] 在线性游戏玩法 (linear gameplay) 中,用户的游戏人物面临着预定顺序的挑战,并且玩家不能够真正改变故事情节或故事结局。非线性游戏玩法 (nonlinear gameplay) 给予更多的自由,且准许以各种顺序、支线任务 (side-quests) 和次要情节 (sub-plots) 来完成挑战。终极版本的非线性游戏玩法是如下开放式世界设计,该开放式世界设计提供虚拟世界,玩家几乎能够自由地在他们所希望的时间内漫游该虚拟世界。

[0003] 然而,对许多人而言,这样的开放式世界的访问没有提供合适的社会感;人们寻求更多的如下共享体验,在该共享体验中,游戏事件将玩家聚集在一起,并然后通过所有参与玩家的操作来逐步进行。这样的游戏非常少见,这是因为它们的设计是困难的设计挑战。在没有适当的测试的情况下,非线性游戏的故事情节中的缺陷和谬误的几率相当高。此外,参与到故事情节的过程中的游戏玩家越多,结构越复杂,并且执行所需的方法步骤需要越多的处理资源。迄今为止,传统技术还不能实现如下视频游戏,该视频游戏将共享的多人游戏事件提供至故事情节,其中该故事情节的所有参与用户可容易地通过他们的游戏操作作出贡献。

### 发明内容

[0004] 本发明的目的是解决或缓解上述问题的至少一部分。根据独立权利要求的特征部分的计算机实施方法、设备和计算机程序产品来实现本发明的目的通过。

[0005] 本发明可以被实施为用于向两个以上的用户提供交互式游戏事件的计算机实施方法。在所述方法中,向用户分配游戏人物,且在所述游戏事件中,发起一个以上的子过程,所述一个以上的子过程被配置为在单独的状态机中独立地进行,每个所述状态机响应于交互功能的特定集合。针对每个子过程,选择游戏人物的组,以便针对至少两个同时进行的子过程选择至少一个游戏人物。向所选择的游戏人物提供准许交互功能的集合以进行激活,所述准许交互功能的集合来自于在选择所选择的游戏人物时所针对的每个子过程的交互功能的特定集合,并且响应于所选择的游戏人物在正在进行的子过程中激活它的准许交互功能的集合中的交互功能,改变该正在进行的子过程的状态。响应于该正在进行的子过程的状态改变,更新该子过程中的所选择的游戏人物的准许交互功能的集合。

[0006] 本发明也可以被实施为一种游戏服务器,所述游戏服务器包括处理单元。所述处理单元被配置为:向用户分配游戏人物,且在所述游戏事件中,发起一个以上的子过程,所述一个以上的子过程被配置为在单独的状态机中独立地进行,每个所述状态机响应于交互功能的特定集合。所述处理单元还被配置为:针对每个子过程,选择游戏人物的组,以便针对至少两个同时进行的子过程选择至少一个游戏人物。所述处理单元还被配置为:向所选

择的游戏人物提供准许交互功能的集合以进行激活,所述准许交互功能的集合来自于在选择所选择的游戏人物时所针对的每个子过程的交互功能的特定集合,并且响应于所选择的游戏人物在正在进行的子过程中激活它的准许交互功能的集合中的交互功能,改变该正在进行的子过程的状态。响应于该正在进行的子过程的状态改变,更新该子过程中的所选择的游戏人物的准许交互功能的集合。

[0007] 本发明还可以被实施为一种计算机可读的计算机程序产品,该计算机程序产品在计算机装置中对用于执行本发明的方法的指令进行译码以向两个以上的用户提供交互游戏事件。

[0008] 在从属权利要求中记载了本发明的优选实施例。

[0009] 本发明基于一种新的游戏事件过程配置,该游戏事件过程配置包括用户可以参与其中的一个以上的独立进行的子过程。在子过程中,用户的游戏人物具有能够激活的预先选择的功能且在子过程中对用户可用的活动的集合随着子过程进行而变化。通过动态地控制可用的活动,能够明显减小子过程的复杂度。具有独立进行的子过程的游戏事件的模块化结构进一步减小复杂度。

[0010] 新的配置能够提供具有真正交互故事情节的多人的游戏事件。

## 附图说明

[0011] 下面将参照附图并结合优选实施例来更加详细地说明本发明,其中:

[0012] 图 1 图示了示例性网络系统;

[0013] 图 2 示出了用于图示示例性装置的配置的模块图;

[0014] 图 3 图示了游戏事件过程的实施例;

[0015] 图 4 示出了具有示例性游戏人物参数定义的表格;

[0016] 图 5 图示了具有预定的持续时间的游戏事件的简化示例;

[0017] 图 6 图示了子过程的状态机的更新方案。

## 具体实施方式

[0018] 下面的实施例是示例性的。尽管说明书可能参照“一”、“一个”或“一些”实施例,但是这并非必然意味着每个这样的参照是指相同的实施例,或者并非必然意味着特征仅适用于单个实施例。可以将不同实施例的单个特征进行组合以提供进一步的实施例。

[0019] 下面,将使用用于实施本发明的各种实施例的系统架构的简单示例来说明本发明的特征。仅详细说明用于举例说明实施例的相关要素。计算机实施方法、设备和计算机程序产品的各种实施包括被本领域技术人员通常已知的且在这里可没有被具体说明的要素。

[0020] 图 1 图示了可用于实施本发明实施例的示例性网络系统。所示的网络系统包括游戏服务器 14、网络 13 和一个或多个用户终端 10、11 和 12。这里,网络 13 表示硬件部件和软件部件的任何组合,该组合用于使一个通信端点中的过程能够将信息发送到另一个远程通信端点的另一个过程或能够从另一个远程通信端点的另一个过程接收信息。例如,网络 13 可以是个人区域网、局域网、家庭网、存储区域网、校园网、主干网、城域网、广域网、企业专用网、虚拟专用网或互联网,或这些网络的任何组合。

[0021] 用户终端 10、11 和 12 中的至少一者包括终端应用程序 APP-T 15。终端应用程序

15 是用户可控应用程序,该应用程序存储于或可以存储于用户终端的存储器中,并且提供指令,上述指令当被用户终端的处理单元执行时进行这里所述的功能。表述“用户可控”意

味着:执行应用程序的用户终端包括用户界面,且用户可以借助于用户界面来控制应用程序的执行。因此,用户可以发起和终止应用程序的运行,可以提供用于对在用户终端中被处理的指令的顺序进行控制的命令。

[0022] 用户终端 10、11 和 12 中的上述至少一者还包括浏览器 16,用户可以经由用户界面 UI 15 访问浏览器 16。因此,用户可以使用浏览器 16 来与连接至网络 13 的游戏服务器 14 通信。游戏服务器 14 可以是具有 IP 地址和域名的网页服务器。游戏服务器 14 也可以被实施为用于提供网页服务器功能的云。

[0023] 可以通过图 1 所示的可作为服务器的计算机装置来实施本发明的实施例。图 2 示出了用于图示示例性装置的配置的模块图。

[0024] 装置包括处理单元 20,以用于系统化地执行针对数据的操作。处理单元 20 是如下元件,该元件基本上包括一个或多个算法逻辑单元以及多个专用寄存器和控制电路。存储单元 21 提供能够存储用户数据或计算机可读数据或程序的数据媒介。存储单元连接至处理单元 20。存储单元 21 可以包括易失性或非易失性存储器,例如,EEPROM、ROM、PROM、RAM、DRAM、SRAM、固件和可编程逻辑等。

[0025] 装置可以包括接口单元 22,接口单元 22 具有至少一个用于将数据输入至装置的内部过程的输入单元和至少一个用于将数据从装置的内部过程输出的输出单元。装置的接口单元 22 可以包括用户接口单元 23,用户接口单元 23 具有用于输入用户数据的按键、触摸屏、麦克风及其等同物和用于输出用户数据的屏幕、触摸屏、扬声器及其等同物。装置的接口单元还可以包括网络接口单元 24,网络接口单元 24 提供用于网络连接的构件。如果采用有线接口,网络接口单元 24 通常包括插入式部件,该插入式部件对于被传送至它的外部连接点的信息和对于被馈送至与它的外部连接点连接的线路的信息起到网关作用。如果采用无线接口,网络接口单元 24 通常包括无线收发单元,该无线收发单元包括发送器和接收器,并且也电连接至处理单元 20。无线收发单元的发送器接收来自处理单元 20 的比特流,并将比特流转换成由天线发送的无线电信号。相应地,由天线接收的无线电信号被馈送至无线收发单元的接收器,接收器将无线电信号转换成比特流,且经转换的比特流被转发至处理单元 20 以进行进一步处理。

[0026] 处理单元 20、存储单元 21 和接口单元 22 电互连以提供如下构件,该构件根据装置的预定的、基本上经编程的过程对所接收和 / 或存储的数据系统性地执行操作。这些操作包括这里所述的用于应用程序服务器的构件、功能和过程。这些单元可以存在于一个物理元件中或可以连接成网络以用于分布式操作。

[0027] 一般而言,装置的各种实施例可以由硬件或专用电路、软件、逻辑或它们的任何组合来实施。一些方面可以由硬件来实施,而一些其它方面可以由固件或由控制器、微处理器或其它计算设备执行的软件来实施。软件程序(也被称为程序产品)是制造产品并能够存储于任何的装置可读数据存储媒介,并且它们包括用于进行特定任务的程序指令。

[0028] 虽然已经作为模块图、信息流图、流程图和逻辑流程图或使用一些其它图形表示图示和说明了本发明的各方面,但是容易理解:图示的单元、模块、装置、系统元件、步骤和方法可以例如由硬件、软件、固件、专用电路或逻辑、计算装置或它们的一些组合来实施

[0029] 游戏服务器被配置为用于实施如下过程,该过程向包括至少两个用户的组提供交互式联网游戏事件。有利地,游戏不是集体游戏,但组的大小限于预定的数量。在上下文中,交互是指用户组中的用户具有与游戏服务器的双向通信连接,且游戏服务器中的过程响应于经由通信连接而给出的用户命令。游戏是指游戏玩法的设置,并且包括游戏人物的定义、游戏人物的背景故事和虚拟场景等。游戏定义向游戏提供标记并且在改变游戏事件时通常保持不变。在这里,游戏事件是指在游戏中建立的过程并且具有定义的开始点和结束点。结束可以是定时的或者可以由游戏事件过程的特定状态触发。可以将游戏事件变成对非限制用户组来说是可访问的,其中可以从非限制用户组中选出参与者组。因此,可以在对游戏事件的设计不做本质改变的情况下对变化的用户组重复游戏事件。

[0030] 图 3 的流程图图示了根据本发明的游戏事件过程的实施例。在开始时,针对游戏事件,选择用户组(步骤 300)。有利地,游戏具有注册子过程,借助于该注册子过程,想要参与游戏事件的用户可以注册到一个以上的即将到来的游戏事件。注册子过程也可以包括具体的规则集,借助于该具体的规则集,当发起游戏的时间临近时,游戏服务器可以选择参与者。简单的规则集可以根据用户的注册时间将用户排列成行,并从行的顶部开始将用户  $U_i$  选择到参与者组。然而,在权利要求的范围内,可以应用更先进的选择方法。

[0031] 在发起游戏事件时,可以向参与用户提供游戏人物(步骤 302)。游戏人物是将用户与人物特征参数集合关联的规范,并且在整个游戏事件中对用户而言是有效的。在游戏事件结束时,可以重置游戏角色定义,以便在任何后续的游戏事件中,无论用户在先前的游戏事件中具有何种人物,但是可以向他或她分配相同或不同的人物。

[0032] 有利地,游戏人物具有与游戏设置相匹配的特定图形外观、背景故事以及特征化的特征和能力。游戏人物的特征可以由游戏人物参数定义。游戏人物参数可以在整个游戏事件中保持不变,或游戏人物参数中的一些参数可以在游戏事件期间动态地变化以将不可预测性和变化添加到情节片段。图 4 示出了用于图示游戏人物参数定义的简化示例的表格。表格的行对应于游戏事件的角色  $C1 \cdots C6$ ,且表格的列对应于可用于人物的不同游戏人物参数。可以看出,每个人物可以具有一个多个特征,且人物可以共用一些相同的特征。

[0033] 应注意,图 4 中的人物的数量和参数的数量仅是示例性的。游戏可以包括多个人物定义和多个人物,还可以在发起时在游戏服务器中决定要被应用于游戏事件的人物的身份。有利地,游戏人物参数是基于游戏设置的。作为示例,让我们假设游戏是如下剧情游戏:虚拟人物在虚拟环境中四处走动、彼此聊天和交互。游戏人物参数  $p1 \cdots p5$  可能涉及如下特征,这些特征在人际交互方面(例如,男性/女性、漂亮/平凡/丑陋、富有/贫穷、环保人士/技术专家和年长/年轻等)被认为是相关的。

[0034] 游戏事件被布置为包括多个独立进行的子过程,且用户可以经由他们的游戏人物参与这些子过程中的一个或多个。游戏服务器可以包括子过程触发规则集  $C_{sp}$ ,子过程触发规则集  $C_{sp}$  定义用于发起子过程的变化条件。例如,可以将子过程调度为在定义的时间点处发生,或者一个子过程可以在一些其它子过程已经结束时被触发。因此,子过程可以通过触发条件而是相关的,但是在发起后,子过程可以在没有与任何其它子过程交互的情况下独立地进行。

[0035] 图 5 图示了具有预定持续时间  $T$  的游戏事件的简化示例。游戏事件包括六个子过

程 SP1、SP2、SP3、SP4、SP5 和 SP6,每个子过程在定义的触发点(使用圆圈表示)处发起,且然后在没有彼此交互的情况下从它们的内部状态独立地进行到另一个状态。一些子过程(参看 SP6)可以在游戏事件期间多次地重复。图 5 采用图 4 的游戏人物来图示:游戏人物

选择以参与子过程 SP3、SP4 和 SB6,游戏人物 C2 被选择以仅参与子过程 SP3,并且游戏人物 C3 被选择以参与子过程 SP1、SP3 和 SP6。

[0036] 返回到图 3,当聚集用户且分配游戏人物时,游戏服务器被配置为检查子过程触发规则集  $C_{SP}$  中的任一规则是否得到满足(步骤 304)。如果是(步骤 306),发起相应的子过程  $SP_k$ (步骤 308)。

[0037] 子过程是游戏事件的可能涉及预定组的游戏角色的设计要素。在发起时,可以为这些游戏角色选出多个游戏人物。可以将子过程中的一个游戏角色仅分配给一个游戏人物,或分配给多个游戏人物。可以经由各自的用户界面将游戏人物引导或邀请到子过程。这可以通过本领域技术人员众所周知的各种方式来实现。例如,可以向针对子过程选择的游



所有游戏角色或人物可以具有不同的目标,或至少一些游戏角色或人物可以具有相同的目标。可以将一些目标定义为对立的,以便当一个游戏人物实现游戏角色的目标时,其它游戏人物中的一个或以上失败。在图 6 的示例性游戏设置中,游戏角色 R1 的目标可以是去发现游戏角色 R3 的秘密,且游戏角色 R3 的目标可以是维持秘密直至游戏事件的结束。游戏角色 R2 的目标可以是防止游戏角色 R1 发现游戏角色 R3 的秘密。对他们而言,这样不同的且有时甚至冲突的目标能够在游戏事件期间驱动用户活动并由此有助于非线性发展故事情节的印象。

[0041] 返回到图 3,在子过程的发起时(步骤 308),游戏服务器为子过程的游戏角色选择游戏人物。可以根据游戏人物的参数定义来选择,这些参数与次要情节的故事之间的相关性得到考虑。在任何一个子过程的发起后,游戏服务器开始待命以处理用于在任何一个进行的子过程中请求激活交互功能  $f(SP_{all})$  的用户命令(步骤 310)。如果检测到交互功能(步骤 312),游戏服务器可以确定交互功能所属的子过程  $SP_j$ (步骤 314),并且更新子过程  $SP_j$  的特定状态机  $SM(SP_j)$ (步骤 316)。同时,游戏服务器更新对子过程中的游戏人物可用的交互功能  $f(SP_j)$  的集合(步骤 318)。如图 3 所示,子过程的发起和子过程状态机的更新以及与交互功能的集合相关的更新可以继续,直至游戏事件的结束(步骤 320)。如上所述,结束可以是定时的,或可以由子过程的某一状态或活动触发。在游戏事件的结束时,游戏事件可以根据子过程的状态机的状态以及其它记录的或预先设计的材料来编译在游戏事件期间发生的情节以及突发事件的总结报告(步骤 322)。可以将将这个报告以游戏事件故事情节描述的形式提供给用户。

[0042] 让我们考虑剧情游戏中的子过程的示例,其中网页服务在服务平台上运行。在服务中,单个剧情可以具有 30 分钟至 2 小时的短的游戏时间。游戏事件提供如下真实的多人游戏:故事中的每个人物通过由图形化人物表示的真人扮演。该社交游戏玩法的人物被准许在虚拟环境中四处走动、通过给出的用户界面和聊天机构彼此聊天和交互。剧情每次发展和变化到不同的结局。

[0043] 在示例性剧情游戏中,向每个用户参与者给定游戏人物。在整个游戏中,游戏人物能够选择它自己的动作和反应,且用户可以在剧情中看见该动作和反应的结果。如上所述,对游戏人物可用的动作(交互功能)的集合根据游戏角色及故事情节的情形(即,游戏人物目前参与到的子过程的状态机的状态)而不同。可以使用用于共同地表明状态机的可用交互功能的可用用户界面选择的组合来激活交互功能。

[0044] 作为示例,子过程中的游戏人物可以具有游戏角色马克,马克的目标是追求特定女孩。在子过程中,游戏人物马克的用户可以被提供有例如诗或书本等物品以及例如阅读或给予等一些交互功能。马克的用户可以选择对象,且通过用户在用户显示器中选择诗-阅读-琳达这一组合并通过按下 OK 键,游戏人物马克将进行交互功能以将诗阅读给琳达。这个动作将把特定子过程的状态改变成所期望的琳达的响应的状态。在琳达响应马克之前,琳达不能够在子过程中发起其它活动,准许或拒绝读诗给她。在同一状态下,马克等待反应直至琳达应答。为了加快子过程的进行,子过程可以包括用于确定最大时间的定时器,在最大时间后,使用默认选择来更新状态机的状态。

[0045] 即使不存在技术链接,游戏角色或人物的目标也趋向于关联并由此链接子过程。例如,在示例性子过程中,游戏角色约翰可能需要扔掉物品,且如果另一游戏角色安从约翰

取走该物品,约翰因此同意与安签订另一个子过程的约定。最初,应用物品的一个子过程和应用约定的另一个子过程彼此之间毫无关系,但是游戏角色的用户通过他们的相互动作将这些子过程联系在一起。

[0046] 上述的个人动作将使给定的故事情节向各种方向发展。发生在游戏人物之间的谈判和谈话形成了游戏玩法的基本部分,但是由于子过程中的交互功能提供的动作和反应向游戏事件给出了涵义和结构 (meaning and structure) 的故事情节类型。

[0047] 如上所述,游戏角色能够以不同状态激活的交互功能起到动态地改变游戏的规则的作用。游戏角色中的游戏人物被准许做出某些动作,且游戏人物的和 / 或其它游戏人物的过去的动作确定了当前的状态且因此确定了在此状态下的当前规则。

[0048] 让我们考虑具有九个状态的明显简化的子过程及其状态机的规则集。被编号为 XXX、YYY 和 ZZZ 的状态描述了可能的方向 (为了简洁起见,没有对其描述)。

[0049] 子过程涉及两个不同的游戏角色 :

[0050] A :针对保护地址试图收集姓名的自然活动家

[0051] B、C :潜在的签名者。

[0052] 因此,子过程具有六个可能的标签 :

[0053] 地址已经被签署姓名 :B、C

[0054] 谁写下了谁的姓名 :B:B、B:C、C:C 和 C:B

[0055] 在这个示例中,这些标签的布尔存在性定义了不同的状态。在示例性的游戏事件中,子过程的状态机可以如下地进行 :

[0056] 状态 1 :开始

[0057] 标签 :-

[0058] A 动作 :

[0059] 向 B 要签名 -> 转到状态 2

[0060] 向 C 要签名 -> 转到状态 XXX

[0061] B 和 C 动作 :

[0062] 无

[0063] 状态 2 :A 要 B 的签名

[0064] 标签 :-

[0065] A 动作 :

[0066] 等待 B 的反应

[0067] B 动作 :

[0068] 签名 -> 转到状态 3

[0069] 拒绝 -> 转到状态 1

[0070] 假装,作为别人进行签名 (在这种情况下,作为 C) -> 转到状态 4

[0071] C 动作 :

[0072] 无,不能中断 A 和 B 交互

[0073] 状态 3 :B 作为 B 签名

[0074] 标签 :B、B:B

[0075] A 动作 :

- [0076] 要求 C 签名 -> 转到状态 5
- [0077] B 和 C 动作 :
- [0078] 无
- [0079] 状态 4 :B 假装 C 的名字
- [0080] ( 观察到 :从用户的观点来看,状态 3 和 4 看起来一样 )
- [0081] 标签 :C、B:C
- [0082] A 动作 :
- [0083] 要求 C 签名 -> 转到状态 6
- [0084] B 和 C 动作 :
- [0085] 无
- [0086] 状态 5 :B 签名、A 要求 C 签名
- [0087] 标签 :B、B:B
- [0088] A 动作 :
- [0089] 等待 C 的反应
- [0090] B 动作 :
- [0091] 无
- [0092] C 动作 :
- [0093] 签名 -> 转到状态 7
- [0094] 拒绝 -> 转到状态 3
- [0095] ( 在这种情况下不可能假装,这是因为这个示例中仅 3 个人 )
- [0096] 状态 6 :B 假装 C、A 要求 C 签名
- [0097] 标签 :C、B:C
- [0098] A 动作 :
- [0099] 等待 C 的反应
- [0100] B 动作 :
- [0101] 无
- [0102] C 动作 :
- [0103] 拒绝 -> 转到状态 4
- [0104] 生气 ( 因为 C 的名字已经在列表上 ) -> (A 得知谁假装 ) 转到状态 4
- [0105] 消除自己的名字 -> 转到状态 9。
- [0106] 假装,写下 B 的名字 -> 转到状态 8。
- [0107] 状态 7 :B 签名 B, C 签名 C
- [0108] 标签 :B、C、B:B、C:C
- [0109] 没有其它动作
- [0110] 状态 8 :B 假装 C、C 假装 B
- [0111] 标签 :B、C、B:C、C:B
- [0112] 没有其它动作
- [0113] 状态 9 :B 假装 C,但是 C 不再被列出
- [0114] 标签 :B:C

[0115] A 动作：

[0116] 要求 B 签名 -> 转到状态 YYY

[0117] 要求 C 签名 -> 转到状态 ZZZ

[0118] 可以看出,游戏人物的动作是具有设置标签的可能性的交互功能。当子过程的状态变化时,也更新对 A、B 和 C 可用的动作。当没有其它动作时,子过程可以结束。

[0119] 通过动态地更新交互功能的可用性,在不使交互式控制的用户体验妥协于游戏的情况下,明显减少用于管理的可能状态的数量。例如,在本子过程中,玩家之间 (A-B、A-C 和 B-C) 进行的交互也需要被视为状态定义要素。因此,标签的总数量是九,且在这个简化示例中,排列 9! 产生 362880 个可能的标签组合。容易理解,在使用传统状态机的任何实际实施中,状态的数量很快达到永远也不可能构建、存储和处理的数量。

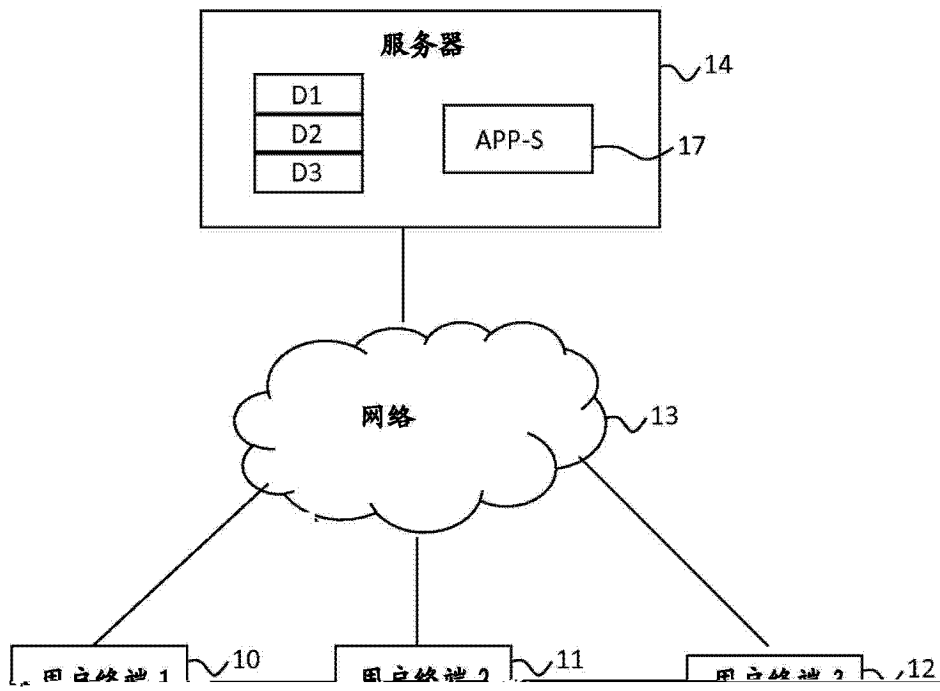
[0120] 容易看出,在实际的状态机中,状态的数量很快达到人们永远也不可能构建的数量且存储需求将会是巨大的。通过定义动作的更新规则(即,交互功能对游戏人物的可用性)来避免这种情况。然后,规则可以在游戏服务器中实时运行,且它们定义了运行中的当前状态。

[0121] 由于所提出的方案,通过所准许的交互功能链,在示例性次要情节中实际上仅可

达到 38 个状态。在四个玩家的情况下,存在 793 个可能状态,且在五个玩家的情况下,存在 37001 个可能状态等。容易想到这样的减小复杂度的技术优势。

[0122] 通过上述过程配置,每个单个游戏事件的用户体验是唯一的,且故事情节看起来根据参与到游戏事件的所有用户的活动来真实地进行。即使子过程不彼此技术交互,但单个玩家也连续地联合处理他的或她的子过程。单个子过程的状态机绝不是微不足道的,但是复杂度的数量仍保持为可管理的,这是因为状态机不需要对子过程之间的不可控制的排列进行负责。因此,在某种意义上,通过仅对故事情节的相关部分建模,实现了非线性故事情节。由于减小的复杂度,创建游戏的开发工作和运行游戏事件的计算资源明显少于使用传统游戏的情况。确保游戏的稳定运行所需的测试工作也大幅减少,这是因为能够将测试仅集中到有限数量的单独子过程状态机。

[0123] 对本领域技术人员来说,显然地,随着技术的进步,本发明的基本理念能够以各种方式实现。本发明及其实施例不限于上述示例,而可以在权利要求的范围内变化。



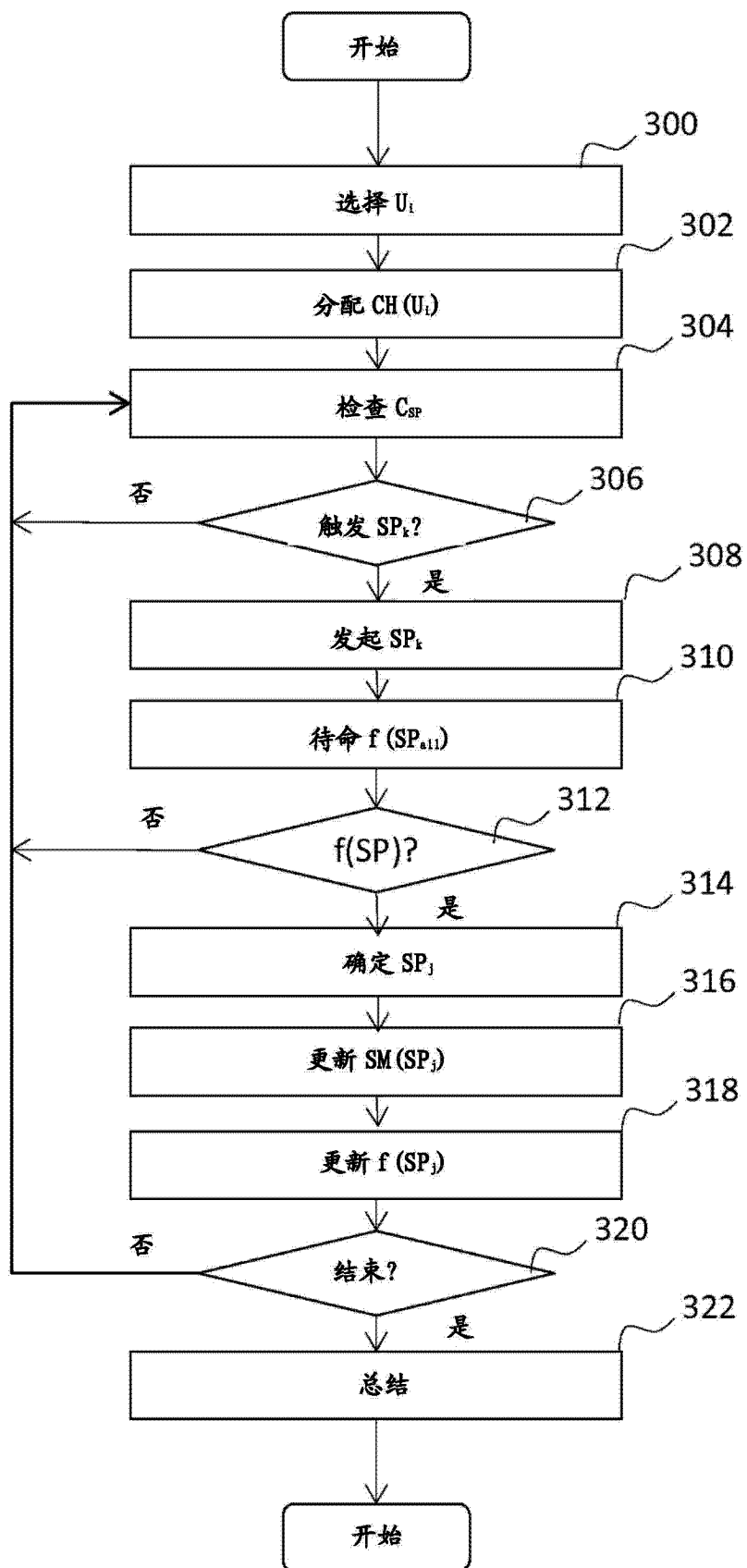


图 3

	p1	p2	p3	p4	p5
C1	x			x	
C2		x		x	x
C3	x	x			
C4	x		x		
C5				x	x
C6		x	x	x	

图 4

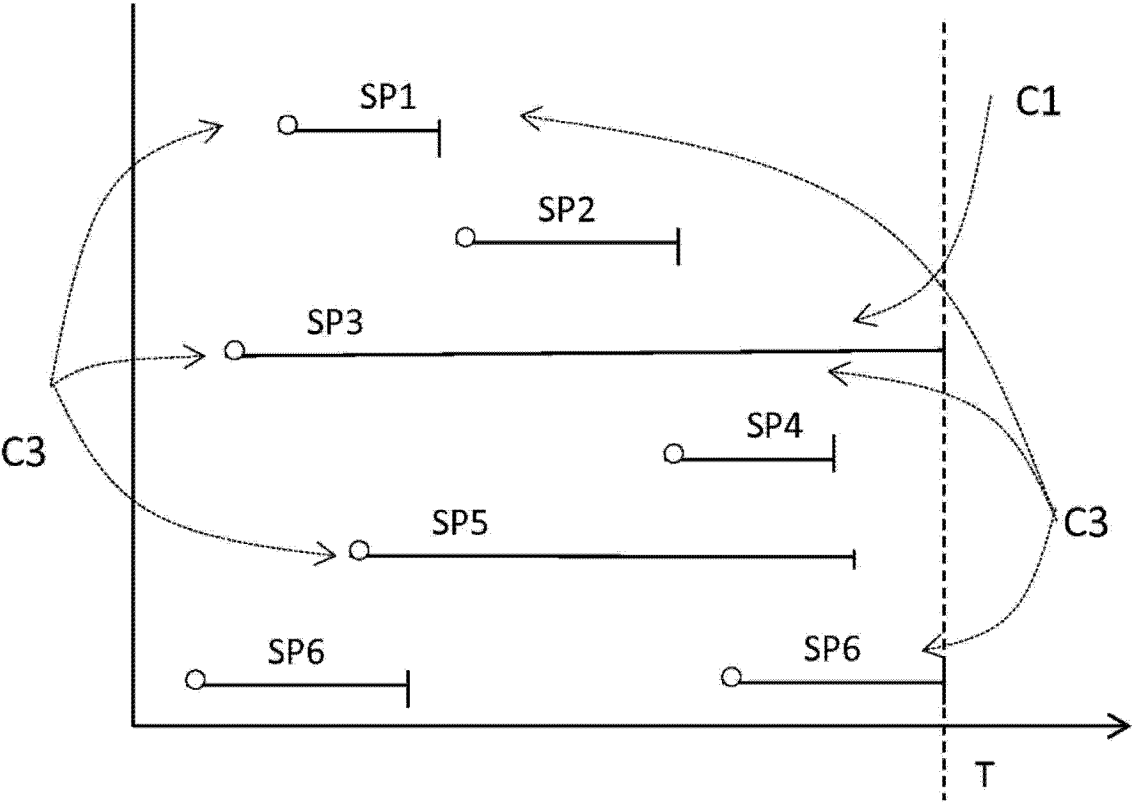


图 5

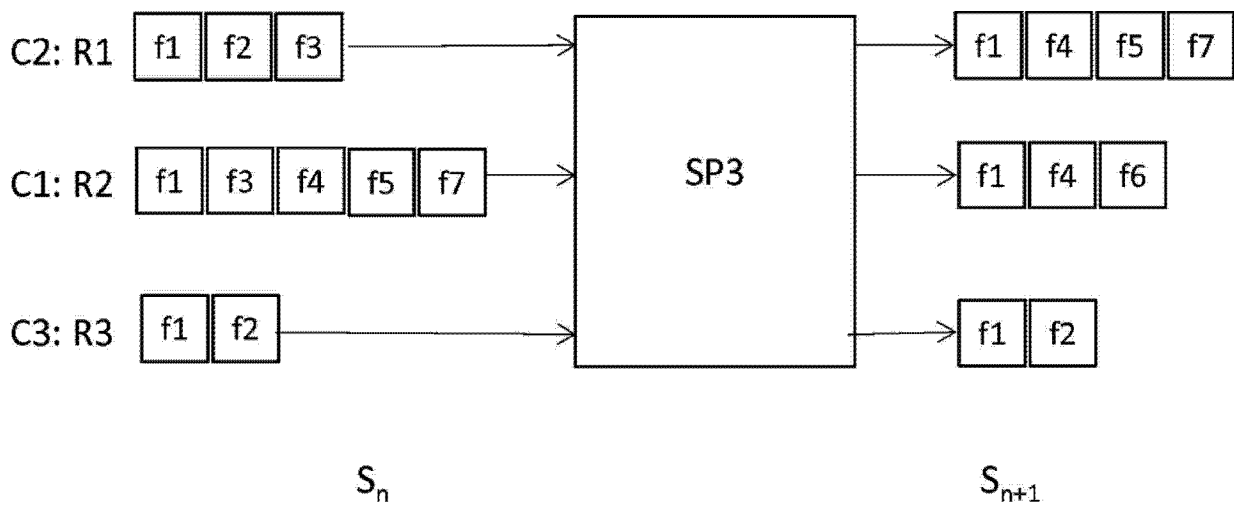


图 6