



(12)发明专利申请

(10)申请公布号 CN 110831677 A

(43)申请公布日 2020.02.21

(21)申请号 201880044708.3

(22)申请日 2018.05.02

(30)优先权数据

62/501,333 2017.05.04 US

(85)PCT国际申请进入国家阶段日

2020.01.02

(86)PCT国际申请的申请数据

PCT/US2018/030645 2018.05.02

(87)PCT国际申请的公布数据

WO2018/204494 EN 2018.11.08

(71)申请人 MZ知识产权控股有限责任公司

地址 美国加利福尼亚州

(72)发明人 A·帕里库奇 J·金姆

(74)专利代理机构 北京世峰知识产权代理有限公司 11713

代理人 卓霖 许向彤

(51)Int.Cl.

A63F 13/47(2006.01)

A63F 13/69(2006.01)

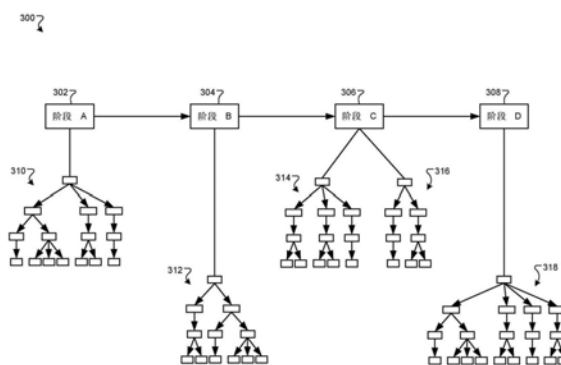
权利要求书2页 说明书12页 附图4页

(54)发明名称

用于在多玩家在线游戏中管理内容呈现的系统和方法

(57)摘要

本公开的实施方式涉及用于在多玩家在线游戏中管理内容呈现的方法、系统和物品。示例方法可以包括：向多个用户提供具有虚拟环境的多玩家在线游戏；向至少一个用户分配时间线，该时间线限定将为至少一个用户将内容项目引入虚拟环境的时间；在时间上向前推进虚拟环境；以及根据时间线限定的时间将内容项目引入虚拟环境。



1. 一种计算机实现的方法,包括:  
向多个用户提供多玩家在线游戏,所述多玩家在线游戏包括虚拟环境;  
向至少一个用户分配时间线,所述时间线限定将为所述至少一个用户将内容项目引入所述虚拟环境的时间;  
在时间上向前推进所述虚拟环境;和  
根据所述时间线限定的时间将所述内容项目引入所述虚拟环境。
2. 根据权利要求1所述的方法,其中,所述多玩家在线游戏包括生命周期,所述生命周期包括多个阶段。
3. 根据权利要求2所述的方法,其中,所述时间线对应于所述多个阶段中的单个阶段。
4. 根据权利要求2所述的方法,其中,每个阶段包括至少一个不同的时间线。
5. 根据权利要求2所述的方法,其中,至少一个阶段包括重叠的时间线。
6. 根据权利要求2所述的方法,其中,所述生命周期包括一系列时间线,并且其中,所述一系列中的每个时间线包括所述生命周期的一部分。
7. 根据权利要求1所述的方法,其中,所述至少一个用户包括所述在线游戏的一组新玩家。
8. 根据权利要求1所述的方法,其中,所述时间线包括树结构,所述树结构包括节点和分支。
9. 根据权利要求1所述的方法,其中,所述内容项目包括游戏事件、游戏特色、游戏规则、虚拟项目、虚拟角色及其任意组合中的至少一项。
10. 根据权利要求1所述的方法,其中,在时间上向前推进所述虚拟环境包括运行所述在线游戏的游戏时钟。
11. 一种系统,包括:  
被编程为执行操作的一个或多个计算机处理器,所述操作包括:  
向多个用户提供多玩家在线游戏,所述多玩家在线游戏包括虚拟环境;  
向至少一个用户分配时间线,所述时间线限定将为所述至少一个用户将内容项目引入所述虚拟环境的时间;  
在时间上向前推进所述虚拟环境;和  
根据所述时间线限定的时间将所述内容项目引入所述虚拟环境。
12. 根据权利要求11所述的系统,其中,所述多玩家在线游戏包括生命周期,所述生命周期包括多个阶段。
13. 根据权利要求12所述的系统,其中,所述时间线对应于所述多个阶段中的单个阶段。
14. 根据权利要求12所述的系统,其中,每个阶段包括至少一个不同的时间线。
15. 根据权利要求12所述的系统,其中,至少一个阶段包括重叠的时间线。
16. 根据权利要求12所述的系统,其中,所述生命周期包括一系列时间线,并且其中,所述一系列中的每个时间线包括所述生命周期的一部分。
17. 根据权利要求11所述的系统,其中,所述至少一个用户包括所述在线游戏的一组新玩家。
18. 根据权利要求11所述的系统,其中,所述时间线包括树结构,所述树结构包括节点

和分支。

19. 根据权利要求11所述的系统,其中,在时间上向前推进所述虚拟环境包括运行所述在线游戏的游戏时钟。

20. 一种物品,包括:

在其上存储有指令的非暂时性计算机可读介质,所述指令在由一个或多个计算机处理器执行时使所述计算机处理器执行操作,所述操作包括:

向多个用户提供多玩家在线游戏,所述多玩家在线游戏包括虚拟环境;

向至少一个用户分配时间线,所述时间线限定将为所述至少一个用户将内容项目引入所述虚拟环境的时间;

在时间上向前推进所述虚拟环境;和

根据所述时间线限定的时间将所述内容项目引入所述虚拟环境。

## 用于在多玩家在线游戏中管理内容呈现的系统和方法

[0001] 相关申请的交叉引用

[0002] 本申请要求2017年5月4日提交的美国临时专利申请No.62/501,333的权益,其全部内容通过引用合并于此。

### 技术领域

[0003] 本公开涉及多玩家在线游戏,并且在一些示例中,涉及用于在多玩家在线游戏中管理内容呈现的系统和方法。

### 背景技术

[0004] 通常,使用客户端设备与在线游戏的虚拟环境进行交互的数十万甚至数百万的玩家可以玩多玩家在线游戏。玩家通常在在线游戏中致力于完成任务、获取资产或取得特定分数或等级。一些游戏要求或鼓励玩家组成可以与其他玩家或玩家组对抗的组或队。

[0005] 在典型示例中,玩家通过向用户输入设备提供输入来与虚拟环境交互。用于游戏的一般用户输入设备可以包括例如触摸屏、小键盘、操纵杆和/或键盘。使用触摸屏,用户可以通过点击或选择屏幕上显示的项目来与虚拟环境进行交互。

### 发明内容

[0006] 总体上,本文所述的系统和方法提供了一致且自动化的方法,该方法用于在例如多玩家在线游戏之类的计算机游戏中管理内容(例如,游戏特色、事件、新规则、虚拟项目等)的呈现。可以创建一个或多个时间线,其限定将内容项目引入游戏的时间。可以将时间线之一分配给游戏的一组用户。随着游戏(或游戏的虚拟环境)在时间上推进,可以根据时间线将内容项目引入到该组用户中。

[0007] 有利地,本文描述的方法可以极大地简化和提高游戏管理内容呈现的能力,特别是当游戏的规模扩大到包括数百万可以互相交互并影响游戏事件的用户时。该方法可以允许为一个或多个用户管理时间抽象,并且可以以编程方式使引入或投送给用户的每一内容项目自动化。该方法通常是可缩放的,并且可以随着游戏普及性的增加允许游戏容纳和管理新用户的内容。与以前的方法不同,使用时间线可以避免人工干预和/或可以确保每个玩家以相似、一致的方式接收内容。这可以平衡游戏环境,并防止任何用户由于游戏内容的不一致投送而获得不公平的优势。另外地或替代地,通过避免或不需要人工干预,本文描述的方法可以减少运行游戏所需的运营商的数量,从而降低维护或管理游戏所需的成本和工作。

[0008] 一方面,本说明书中描述的主题涉及一种计算机实现的方法。该方法包括:向多个用户提供多玩家在线游戏,所述多玩家在线游戏具有虚拟环境;向至少一个用户分配时间线,所述时间线限定将为所述至少一个用户将内容项目引入所述虚拟环境的时间;在时间上向前推进所述虚拟环境;和根据所述时间线限定的时间将所述内容项目引入所述虚拟环境。

[0009] 在某些示例中,多玩家在线游戏可以包括具有多个阶段的生命周期。时间线可以对应于多个阶段中的单个阶段。每个阶段可以包括至少一个不同的时间线。至少一个阶段可以包括重叠的时间线。生命周期可以包括一系列时间线,并且该系列中的每个时间线可以包括或限定生命周期的一部分。

[0010] 在一些实施方式中,至少一个用户包括在线游戏的一组新玩家。时间线可以包括具有节点和分支的树结构。内容项目可以包括游戏事件、游戏特色、游戏规则,虚拟项目、虚拟角色及其任何组合中的至少一项。在时间上向前推进虚拟环境可以包括运行在线游戏的游戏时钟。

[0011] 在另一方面,本说明书中描述的主题涉及一种系统。该系统包括被编程为执行操作的一个或多个计算机处理器,该操作包括:向多个用户提供多玩家在线游戏,所述多玩家在线游戏具有虚拟环境;向至少一个用户分配时间线,所述时间线限定将为所述至少一个用户将内容项目引入所述虚拟环境的时间;在时间上向前推进所述虚拟环境;和根据所述时间线限定的时间将所述内容项目引入所述虚拟环境。

[0012] 在各种示例中,多玩家在线游戏可以包括具有多个阶段的生命周期。时间线可以对应于多个阶段中的单个阶段。每个阶段可以包括至少一个不同的时间线。至少一个阶段可以包括重叠的时间线。生命周期可以包括一系列时间线,并且该系列中的每个时间线可以包括或限定生命周期的一部分。

[0013] 在某些实施方式中,至少一个用户包括在线游戏的一组新玩家。时间线可以包括具有节点和分支的树结构。内容项目可以包括游戏事件、游戏特色、游戏规则、虚拟项目、虚拟角色及其任何组合中的至少一项。在时间上向前推进虚拟环境可以包括运行在线游戏的游戏时钟。

[0014] 在另一方面,本说明书中描述的主题涉及一种物品。所述物品包括在其上存储有指令的非暂时性计算机可读介质,所述指令在由一个或多个计算机处理器执行时使所述计算机处理器执行操作,所述操作包括:向多个用户提供多玩家在线游戏,所述多玩家在线游戏具有虚拟环境;向至少一个用户分配时间线,所述时间线限定将为所述至少一个用户将内容项目引入所述虚拟环境的时间;在时间上向前推进所述虚拟环境;和根据所述时间线限定的时间将所述内容项目引入所述虚拟环境。

[0015] 相对于本发明的给定方面描述的实施例的元素可以用于本发明的另一方面的各种实施例中。例如,可以想到的是,从属于一个独立权利要求的从属权利要求的特征可以用在任何其他独立权利要求的装置、系统和/或方法中。

## 附图说明

[0016] 图1是用于在多玩家在线游戏中管理内容呈现的示例系统的示意图。

[0017] 图2是多玩家在线游戏的示例时间线的示意图。

[0018] 图3是多玩家在线游戏的示例生命周期的示意图。

[0019] 图4是在多玩家在线游戏中使用时间线来管理内容呈现的示例方法的流程图。

## 具体实施方式

[0020] 图1示出了用于在多玩家在线游戏中管理内容呈现的示例系统100。服务器系统

112提供用于实现时间线的功能,该时间线限定何时将内容项引入在线游戏中。服务器系统112包括例如可以在一个或多个地理位置的一个或多个数据中心114处部署的软件组件和数据库。服务器系统112软件组件可以包括游戏模块116和时间线模块118。软件组件可以包括子组件,该子组件可以在相同或不同的单个数据处理装置上执行。服务器系统112数据库可以包括游戏数据120和用户数据122数据库。数据库可以驻留在一个或多个物理存储系统中。下面将进一步描述软件组件和数据。

[0021] 可以提供例如基于web的应用之类的应用作为终端用户应用,以允许用户与服务器系统112进行交互。终端用户应用可以由客户端设备(例如个人计算机128、智能电话130、平板计算机132和膝上型计算机134)的用户通过网络126(例如因特网)访问。其他客户端设备也是可能的。在替代示例中,游戏数据120和/或用户数据122或其任何部分可以被存储在一个或多个客户端设备上。附加地或替代地,系统100的软件组件(例如,游戏模块116和/或时间线模块118)或其任何部分可以驻留在一个或多个客户端设备上或用于在一个或多个客户端设备上执行操作。

[0022] 图1将游戏模块116和时间线模块118描绘为能够与数据库(例如,游戏数据120和用户数据122数据库)进行通信。游戏数据120数据库通常包括与使用系统100实现的多玩家在线游戏有关的信息。游戏数据120数据库可以包括例如与游戏的虚拟环境,游戏的图像、视频和/或音频数据,与先前、当前或将来的游戏事件对应的事件数据,游戏规则,游戏项目和/或限定游戏的当前状态的游戏状态数据有关的信息。用户数据122数据库通常包括与和在线游戏和/或虚拟环境的用户交互有关的数据。这样的信息可以是或包括例如用户与系统100的连接历史,用户购买,用户成就,用户任务,用户与其他用户的交互(例如,群聊),用户虚拟项目的获取或使用,和/或虚拟环境和/或现实世界中的其他用户状况。

[0023] 多玩家在线游戏可以是或包括例如冒险游戏、虚拟纸牌游戏、虚拟棋盘游戏、车辆操作游戏或其他类型的多玩家游戏。在优选示例中,多玩家在线游戏包括虚拟环境,在虚拟环境中用户或用户的替身可以具有各种能力。这些能力可以是或包括例如将替身或虚拟项目或对象移动到不同地理位置、与角色或其他用户交互、攻击其他用户、部署部队、防御来自其他用户的攻击、部署防御、建立或修改虚拟项目或对象(例如,虚拟建筑物或其他结构)、开发新技能、操作车辆、获取虚拟项目(例如,武器)、使用虚拟项目(例如纸牌或武器)或与之交互以及执行超自然任务(例如,魔法)。其他用户能力也是可能的。

[0024] 通常,游戏模块116和时间线模块118可以包括被配置为为多个用户提供和管理在线游戏的软件组件和/或处理组件。例如,游戏模块116可以被配置为在用户客户端设备(例如,个人计算机128、智能电话130、平板计算机132和膝上型计算机134)上实现游戏,并且为服务器系统112和客户端设备管理游戏状态。这可以涉及渲染在线游戏的虚拟环境和/或根据用户动作和游戏事件更新游戏状态。时间线模块118可以管理和/或利用一个或多个时间线,以将内容项目引入在线游戏中。内容项目可以是或包括例如游戏事件(例如,玩家之间的冲突或竞争)、游戏特色(例如,新用户能力)、游戏规则,虚拟项目(例如,虚拟武器或纸牌)、虚拟角色(例如虚拟动物或人)及其任意组合。随着在线游戏在时间上进展,游戏模块116和/或时间线模块118可以根据一个或多个时间线来引入内容项目。

[0025] 例如,图2是示例时间线200的示意图,该示例时间线200可用于(例如,由时间线模块112)在不同时间将内容项目引入在线游戏中。如所描绘的,时间线200可以被布置为包括

节点和分支的集合的树结构。时间线200的起点由根节点202表示。分支(由线段表示)可以将根节点202连接到时间表节点204层、事件节点206层和内容节点208层。通常,时间表节点204可以表示时间线200中可以发起或时间安排一个或多个游戏事件的位置。在该示例中,时间表节点204包括节点210、212和214,分别表示时间表A、B和C。同样,事件节点206可以表示时间线200中可以发生被时间安排的游戏事件的位置。在该示例中,事件节点206包括节点216、218、220和222,分别表示事件A、B、C和D。附加地或替代地,内容节点208(替代地称为叶节点)可以表示时间线200中与游戏事件相关联的内容项目被引入的位置。在该示例中,内容节点208包括分别表示内容项目A、B、C、D、E、F和G的节点224、226、228、230、232、234和236。尽管在时间线200中表示了七个内容项目(即,项目A至G),但是应当理解,其他时间线可以包括任何合适数量的内容项目(例如,数百、数千等)。还应当理解,任何合适数量的时间表节点和任何合适数量的事件节点可以被包括在时间线中。

[0026] 在各种示例中,时间线200可用于确定何时将内容项目(A到G)中的每一个引入在线游戏中。例如,时间线200中的每个分支可以与分支的一端上的第一节点与分支的相对侧上第二节点之间发生的时间量(例如,以秒、分钟、小时或天计)关联或限定该时间量。出于说明而非限制的目的,在图2中的括号中指示了时间线200中各个分支的时间量。例如,根节点202与时间表A节点210之间的时间量为+1(例如1秒)。另外地或替代地,沿着自根节点202起的路径的任意两个节点之间的时间量可以通过相加该两个节点之间的任何分支的时间来确定。例如,根节点202和节点224(对于项目A)之间的时间量是+2(即 $1+2-1=2$ )。换句话说,如果时间线200在时间 $t=0$ 应用于在线游戏中(例如,根节点202在时间 $t=0$ 发生),则项目A将在时间 $t=2$ (例如,2秒)被引入在线游戏中。按照这种方法,表1呈现了将项目A到G引入在线游戏的时间,假设根节点202在时间 $t=0$ 处被应用。如果根节点202在以后的时间(例如, $t=10$ )被应用,则可以将表1中的时间加到以后的时间(例如,可以在 $t=12$ 时引入项目A)。

项目	引入在线游戏中的时间
A	2
B	5
C	6
D	8
E	13
F	15
G	20

[0029] 表1。如果在时间 $t=0$ 时应用根节点202,则时间线200中项目A到G可以被引入在线游戏中的时间。

[0030] 有利地,用于时间线(例如时间线200)的树结构通常是模块化的,这可以促进时间线的开发和修改。例如,如果内容项目E和F将在游戏中被呈现多次,则包括内容项目E和F(以及时间表B和事件C)的时间线模块240可以被复制并插入到时间线中的不同位置和/或插入到不同时间线。通常,时间线中节点和分支的任何分组或集合(例如,也称为模块)可以在时间线的不同部分中重用或添加到一个或多个其他时间线中。附加地或替代地,当游戏

开发者修改或更新在线游戏时,可以将这样的模块从一个位置移动到不同位置。但是,一般而言,可能优选的是避免有一个以上的模块副本(例如,时间线模块240),而是维护和使用单个主副本,该主副本可以从多个时间线和/或由单个时间线内的一个以上的位置访问。在这种情况下,时间线可以包括到模块主副本的链接。每次使用主副本时,对主副本所做的任何更改都将自动被包括在内。

[0031] 附加地或替代地,用于时间线(例如时间线200)的树结构可以被认为包括根节点和一个或多个时间表。每个时间表可以包括一个或多个事件,并且每个事件可以包括一个或多个内容项目。此外,结构中的节点可以具有较接近根节点的父节点和较远离根节点的子节点。时间表节点(例如,时间表节点204)通常具有至少一个父节点(例如,根节点)和至少一个子节点(例如,事件节点)。事件节点(例如,事件节点206)通常具有至少一个父节点(例如,时间表节点)和至少一个子节点(例如,内容节点)。内容节点(例如,内容节点208)通常具有至少一个父节点(例如,事件节点)而没有子节点。在各种示例中,事件节点和/或内容节点可以具有一个以上的父节点。这允许从时间线中的一个以上的路径到达事件节点和/或内容节点。例如,当相关联的内容节点具有一个以上的父事件节点时,内容项目可以被引入游戏中一次以上。

[0032] 在各种实施方式中,在线游戏可以包括可以用多个时间线来管理的多个阶段。例如,参考图3,在线游戏可以包括具有下列四个阶段的生命周期300:阶段A 302、阶段B 304、阶段C 306和阶段D308。每个阶段可以表示在线游戏的用户在其中体验到一组独特的特色和/或事件的一部分。例如,当一个或多个用户首次加入在线游戏时,可以将用户置于阶段A 302。随着用户在游戏进展(例如,通过推进到更高等级或完成某些任务),可以将用户移动到阶段B 304、然后阶段C 306、再然后阶段D 308。如果需要,可以添加其他阶段。在某些情况下,从一个阶段推进到另一阶段可能需要满足某些转变条件。转变标准可能要求例如用户或用户组在从阶段A 302推进到阶段B 304之前达到某个等级(例如,等级10)。

[0033] 如图所示,生命周期300中的每个阶段可以包括一个或多个时间线或由一个或多个时间线管理。例如,阶段A 302可以包括时间线310,阶段B 304可以包括时间线312,阶段C 306可以包括第一时间线314和第二时间线316,并且阶段D 308可以包括时间线318。如本文中所述的,这些时间线中的每一个可用于确定何时将内容项目引入在线游戏中。对于阶段C 306,第一时间线314和第二时间线316可以串行地(例如,一个在另一个之前)或并行地执行。在并行配置中,两个时间线可以重叠,并且可以用于在基本上相同的时间或在重叠的时间段期间将内容引入在线游戏。

[0034] 图4示出了在在线游戏中管理内容呈现的示例计算机实现的方法400。具有虚拟环境的多玩家在线游戏被提供(步骤402)给多个用户。将时间线分配(步骤404)给用户中的至少一个。时间线限定将为至少一个用户将内容项目引入虚拟环境的时间。在时间上向前推进虚拟环境(步骤406)。根据时间线限定的时间,将内容项目引入(步骤408)虚拟环境。

[0035] 通常,在线游戏和/或游戏的虚拟环境可以与现实世界中的时间推进相同或相似的方式在时间上向前推进。游戏的一个或多个游戏时钟可以跟踪游戏和/或虚拟环境内的时间。在某些实施方式中,游戏可以在时间上推进而不管用户是否登录游戏或玩游戏。

[0036] 在各种示例中,本文所述的时间线的使用和管理可以涉及根据用户所驻留的特定时间线在游戏中构建内容。时间线可以表示游戏的一部分(例如游戏阶段)或全部的给定时



间跨度内存在的一些或所有游戏中事件(重要和/或不重要)。可以针对某些度量建立和维护每个时间线。度量可以是可度量的用户统计信息,其可用于对用户进行分类和/或建立时间线。出于说明而非限制的目的,在时间线的上下文中,度量可以包括例如用户已经在游戏中的时间量,用户自首次登录以来已经在游戏中的时间量,自上次登录以来用户已经在游戏中的时间量以及其他类似度量。

[0037] 随着用户在游戏中的时间进展(例如,从给定的开始时间),可以基于时间线以自动方式向用户投送内容片段。有利地,由于多种原因,本文描述的系统和方法使得内容更易于管理并且更容易投送给用户。例如,内容不需要立即或一次全部投送给整个游戏群体。而是,根据一个或多个时间线和/或游戏阶段,可以将内容随时间推移推出给特定的用户组。另外或替代地,可以显著减少投送游戏内容所需的运营商的数量。时间线模块118例如可以同时管理数百万个游戏用户,并且可以避免必须依靠大量的人工干预。另外地或可替代地,本文描述的方法可以通过确保所有用户以公平和一致的方式接收包括任何相关的优点和缺点的内容来平衡游戏环境。该方法可以防止任何用户例如由于在不同时间将有或有害的内容投送给用户而具有相对于其他用户的竞争优势。

[0038] 此外,本文中描述的多个时间线和/或游戏阶段的使用可以允许游戏运营商为每个时间线时间安排和维护较小的、更易于管理的内容列表,而不必为整个游戏时间安排和维护大的内容列表。附加地或替代地,代替维持整个游戏的当前状态,游戏运营商可以维持时间线内的一个或多个当前状态。在将附加用户添加到游戏时,运营商可以选择时间线中应当放置用户的位置。在优选示例中,一旦放置了新用户,该用户可以自动体验或重新运行时间线描述或以其他方式包含的状态。

[0039] 因此,游戏的整个生命周期可以被划分或以其他方式分解为一系列时间线,每个时间线是游戏生命周期的子集。每个时间线可以具有变化的长度和/或可以与其他时间线邻接或部分重叠。取决于例如时间线的时间长度,时间线相对于游戏生命周期的时间位置以及其他类似因素,每个时间线可包括一个或多个状态和相关联的内容。例如,在任何给定时间,不同的玩家组可以处于时间线的不同的状态或部分,例如不同的时间线模块、时间表或事件。

[0040] 在各个示例中,可以以从较小的构造块到较高的子时间线对象,以不同抽象级别来管理和构建本文所述的时间线。抽象(abstraction)可以是游戏内的分组,其中,整个时间线表示最高的抽象级别,并且时间线的部分或模块(例如,时间表或事件)表示较小的分组。可以是时间线的部分或模块的子时间线对象可以包括此类较小的分组。在优选的实施方式中,时间线的一些或全部可以被结合在另一时间线内(例如,通过将时间线模块从一个时间线复制或移动到另一时间线)。这可以极大地促进在两个或更多时间线之间移动或交换部分或模块的过程。特别地,本文中描述的用于时间线的树结构可以在交换和重新布置项目组时允许灵活性和简单性(例如,通过允许稍后添加不同的抽象级别)。游戏运营商可以将树结构中的不同级别视为不同的抽象。通常,树结构可以很容易地将时间线扩展到将来(例如,通过添加新的时间线模块)。该结构可以具有不同的抽象级别,这取决于游戏运营商希望如何查看时间线。该结构可以使游戏运营商更容易考虑与父代(例如,前一个节点)相关的每个子树(例如,时间表、事件或时间线模块),而不是需要维护的某些绝对状态。

[0041] 尽管树结构是本文描述的时间线的优选结构,但是也可以利用其他时间线结构。

例如,可以使用有向无环图(DAG)方法来实现时间线和/或游戏生命周期。DAG方法可以允许时间线中的一个或多个节点在时间线中具有多个父节点和/或从其他时间线进行访问。有利地,这可以使节点可重复和/或具有一个真实源(source of truth)。例如,当时间线的一部分(例如,模块、时间表或事件)被更新时,指向该部分的所有节点或时间线可以使用DAG方法被自动更新。

[0042] 在各种情况下,为了确定何时将内容项目引入在线游戏中,时间线模块118可以在时间线中平坦化(flatten)内容项目并按时间对项目进行排序。可以使用各种算法来实现此目的。例如,可以使用前序遍历或其他类似算法来计算时间线中每个内容节点的确切运行时间。附加地或替代地,可以使用排序算法等(例如,插入排序)。

[0043] 下面是PHP中的示例应用,其可以描述用于创建或以其他方式实例化时间线的类。在此示例中,可以声明三个视觉抽象:如本文所述,“时间线”涉及根节点;“时间表”涉及子时间线的中级别抽象;并且“事件”涉及子时间线的较低层级抽象。

```
class TimelineNode extends Model
{
    const kMaxTypeLength = 100;
    const kMaxNameLength = 100;
    const kMaxDescLength = 1000;

    const kEvent = 'Event';
    const kSchedule = 'Schedule';
    const kTimeline = 'Timeline';

    public static $displayTypes = [
        1 => self::kEvent,
        2 => self::kSchedule,
        3 => self::kTimeline
    ];

    public $node_id          = [Type::uint, PRIMARYAUTOFAST];
    public $display_type     = [Type::ascii => self::kMaxTypeLength,
    INDEXKEY]; public $name     = [Type::ascii => self::kMaxNameLength,
    INDEXKEY]; public $description = [Type::ascii => self::kMaxDescLength];
    public $create_admin_id  = [Type::uint32];
    public $create_ts        = [Type::uint32];
    public $modified_admin_id = [Type::uint32];
    public $modified_ts      = [Type::uint32];

    ...
}

class TreeNodeChild extends Model
```

```
{  
    //节点包含的被允许类的表。  
    public static $itemTypes = [  
        1 => self::kTreeNodeType,  
        2 => 'BlogPost'  
    ];  
  
    const kStart = 'Start';  
    const kEnd   = 'End';  
  
    //节点具有的被允许动作的表。  
    public static $itemActions = [  
[0045]         1 => self::kStart,  
                2 => self::kEnd  
    ];  
  
    public $node_id      = [Type::uint64, PRIMARYKEY];  
    public $type_id      = [Type::uint32, PRIMARYKEY];  
    public $child_id     = [Type::uint64, PRIMARYKEY];  
    public $action_id    = [Type::uint8, PRIMARYKEY];  
    public $timestamp    = [Type::int64];  
  
    ...  
}
```

[0046] 在一些实施方式中,人工智能和/或一个或多个经训练模型或分类器可用于为用户构建和/或选择时间表。例如,可以收集有关时间线或其组件(例如,模块)的性能的数据。性能数据可以提供时间线或其组件在在线游戏中执行得如何的指示。性能数据可以包括某些性能指标,例如计算效率、网络效率、完成时间、用户满意度、用户停机时间和/或收入。也可以使用其他性能指标。可以收集性能数据和关联的时间线和时间线组件的信息,并将其用于开发预测模型。例如,性能数据和时间线信息可以用作训练数据以训练一个或多个分类器。一旦经训练,分类器或其他预测模型(例如,回归方程)就可以用于预测时间线和/或其组件的性能。最终,预测模型可以提供对于哪些时间线和/或时间线组件要用于在线游戏

中的推荐(例如,基于预测的性能)。这样的预测和时间线选择可以改进用于实现在线游戏的计算机系统(包括例如服务器系统112、网络126和/或任何客户端设备)的性能。这样的改进可以导致更大的用户满意度和/或对在线游戏的更大的参与度。

[0047] 预测模型可以是或包括分类器,例如一个或多个线性分类器(例如费希尔(Fisher)线性判别、逻辑回归、朴素贝叶斯分类器和/或感知器)、支持向量机(例如最小平方支持向量机)、二次分类器、核估计模型(例如k近邻)、加强(元算法)模型、决策树(例如随机森林)、神经网络和/或学习向量量化模型。还可以使用其他分类器。

[0048] 本说明书中描述的主题和操作的实施方式可以在数字电子电路或在计算机软件、固件或硬件(包括本说明书中公开的结构以及其结构等效物)或在它们中的一个或多个的组合中实现。本说明书中描述的主题的实施方式可以被实现为一个或多个计算机程序,即编码在计算机存储介质上以供数据处理装置执行或控制数据处理装置的操作的计算机程序指令的一个或多个模块。替代地或额外地,程序指令可以编码在人工生成的传播的信号(例如,机器生成的电、光或电磁信号,其被生成为编码信息以传输到合适的接收器设备以供数据处理设备执行)上。计算机存储介质可以是或可以被包括在计算机可读存储设备、计算机可读存储基片、随机或顺序存取存储器阵列或设备或它们中一个或多个的组合中。另外,尽管计算机存储介质不是传播信号,但是计算机存储介质可以是以人工生成的传播信号进行编码的计算机程序指令的源或目的地。计算机存储介质还可以是或被包括在一个或多个分离的物理部件或介质(例如,多个CD、盘或其他存储设备)中。

[0049] 本说明书中描述的操作可以被实现为由数据处理装置对存储在一个或多个计算机可读存储设备上或从其他源接收的数据执行的操作。

[0050] 术语“数据处理装置”涵盖用于处理数据的所有种类的设备、装置和机器,举例来说包括可编程处理器、计算机、芯片上系统或前述中的多个或组合。所述设备可以包括特殊目的的逻辑电路,例如FPGA(现场可编程门阵列)或ASIC(专用集成电路)。除了硬件之外,装置还可以包括代码,该代码为所讨论的计算机程序创建执行环境,例如,构成处理器固件、协议栈、数据库管理系统、操作系统、跨平台运行环境、虚拟机或上述一个或多个的组合的代码。所述装置和执行环境可以实现各种不同的计算模型基础设施,例如网页服务、分布式计算和网格计算基础设施。

[0051] 计算机程序(也被称为程序、软件、软件应用、脚本或代码)可以以任何形式的编程语言编写,包括编译或解释型语言、陈述性或过程语言,并且可以以任何形式部署计算机语言,包括以独立程序的形式或以模块、组件、子程序、对象或其他适于用在计算环境中的单元。计算机程序可以但不必与文件系统中的文件对应。程序可以存储在保持其他程序或数据的文件的一部分(例如,存储在标记语言文件中的一个或多个脚本)中、在专用于所针对的单个文件中或在多个协作的文件(例如,存储一个或多个模块、子程序或代码的一些部分的文件)中。计算机程序可以部署为在一个或多个计算机上执行,这些计算机位于一个站点或分布在多个站点之间并通过通信网络连接。

[0052] 本说明书中描述的过程和逻辑流可以由执行一个或多个计算机程序的一个或多个可编程处理器执行,从而通过对输入数据进行操作并生成输出来执行动作。过程和逻辑流也可以由特殊目的的逻辑电路(例如FPGA(现场可编程门阵列)或ASIC(专用集成电路))来执行,并且设备可以被实现为特殊目的的逻辑电路。

[0053] 举例来说,适于执行计算机程序的处理器包括通用和专用微处理器,以及任何类型的数字计算机的一种或多种处理器。通常,处理器将从只读存储器或随机存取存储器或二者接收指令和数据。计算机的基本元件是用于根据指令执行动作的处理器和用于存储指令和数据的一个或多个存储器设备。一般来说,计算机将可以包括用于存储数据的一个或多个大容量存储装置(例如、磁盘、磁光盘、光盘或固态驱动器)或可以操作性地联接一个或多个大容量存储装置以从其接收数据或将数据传输到其处或者接收和传输二者。然而,计算机不必具有这样的设备。另外,计算机可以嵌入到另一装置(例如,仅举例几个,移动电话、个人数字助理(PDA)、移动音频或视频播放器、游戏控制台、全球定位系统(GPS)接收器或便携式存储装置(例如,通用串行总线(USB)快闪驱动器))中。适用于存储计算机程序指令和数据的设备可以包括所有形式的非易失性存储器、介质和存储设备,举例来说,包括半导体存储设备,例如EPROM、EEPROM和闪存设备;磁盘,例如内部硬盘或可移动盘;磁光盘;以及CD-ROM和DVD-ROM盘。处理器和存储器可以由特殊目的的逻辑电路来补充或可以结合到其中。

[0054] 为了提供与用户的交互,本说明书中描述的主题的实施方式可以实现在具有显示装置及键盘和指向装置的计算机上,显示装置例如用于为用户显示信息的CRT(阴极射线管)或LCD(液晶显示)监视器,用户可以通过键盘和指向装置向计算机提供输入,指向装置例如为鼠标、轨迹球或手写笔。也可以使用其他种类的装置来提供与用户的交互;例如,提供给用户的反馈可以是任何形式的感觉反馈,例如视觉反馈、听觉反馈或触觉反馈;并且可以以任何形式接收来自用户的输入,包括声学、语音或触觉输入。此外,通过向由用户使用的装置发送文件和接收来该装置的文件,计算机可以与用户交互;例如,通过响应于从浏览器接收的请求而将网页发送到用户的客户端装置上的浏览器。

[0055] 本说明书中描述的主题的实施方式可以在计算系统中实现,该计算系统包括后端组件(例如作为数据服务器)或包括中间件组件(例如应用服务器)或包括前端组件(例如具有图形用户界面和浏览器的客户端计算机,用户可通过其与本说明书中描述的主题的实施方式进行交互),或者一个或多个这样的后端、中间件或前端组件的任意组合。可以通过数字数据通信的任意形式或媒介(例如通信网络)将系统的组件互连。通信网络的示例包括局域网(“LAN”)和广域网(“WAN”)、互联网(例如,因特网)和点对点网络(例如,自组(ad hoc)点对点网络)。

[0056] 计算系统可以包括客户端和服务端。一般来说,客户端和服务端彼此远离,并且通常通过通信网络交互。通过运行在各个计算机并且彼此具有客户端-服务器关系的计算机程序而形成的客户端和服务器的关系。在一些实施方式中,服务器向客户端设备传输数据(例如,HTML页面)(例如,用于向与客户端设备交互的用户显示数据和从其接收用户输入的目的)。在客户端设备处生成的数据(例如,用户交互的结果)可以从在服务器处的客户端设备接收。

[0057] 尽管本说明书包含许多具体的实现细节,但是这些细节不应当被解释为对任何发明或请求保护的范疇进行限制,而应该被解释为特定于特定发明的特定实施例的特征的描述。在本说明书中在不同实现方式的背景中描述的某些特征也可以以组合方式实现在单个实现方式中。相反,在单个实现方式的背景中描述的各种特征也可以单独地实现在多个实现方式中或以任何合适的子组合的方式来实现。此外,尽管特征可以如上所述描述为在某

些组合中起作用并且最初也这样地请求,但在一些情况下,来自所请求的组合的一个或多个特征可以从组合中排除,并且所请求的组合可以涉及子组合或子组合的变体。

[0058] 类似地,尽管在附图中以特定顺序描绘了操作,但是这不应当被理解为需要以示出的特定顺序或以相继的顺序来执行这样的操作或者需要执行所有示意的操作来取得合意的结果。在某些情况下,多任务和并行处理可为有利。另外,在上述实现方式中对各种系统部件的分离不应当被理解为在所有实现方式中要求这样的分离,而是应当要理解,所描述的程序部件和系统可一般地在单个软件产品中被集成在一起或者被封装成多个软件产品。

[0059] 这样,已经描述了主题的特定实施方式。其他实现方式在下述权利要求的范围内。在一些情况下,记载在权利要求中的动作可以以不同的顺序执行而仍然取得合意的结果。此外,附图中描绘的过程不一定需要所示出的特定顺序或者连续顺序来实现期望的结果。在某些实施方式中,多任务和并行处理可为有利。

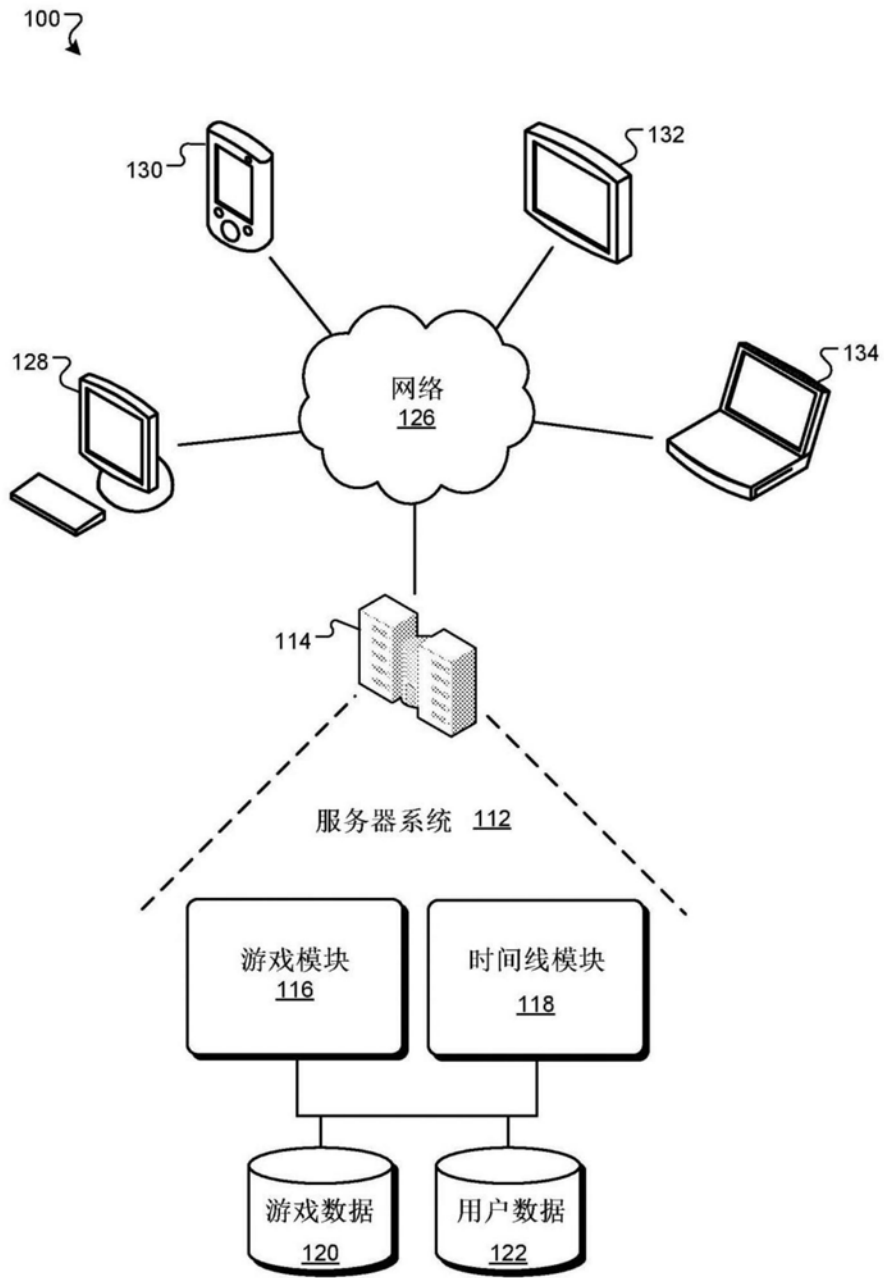


图1



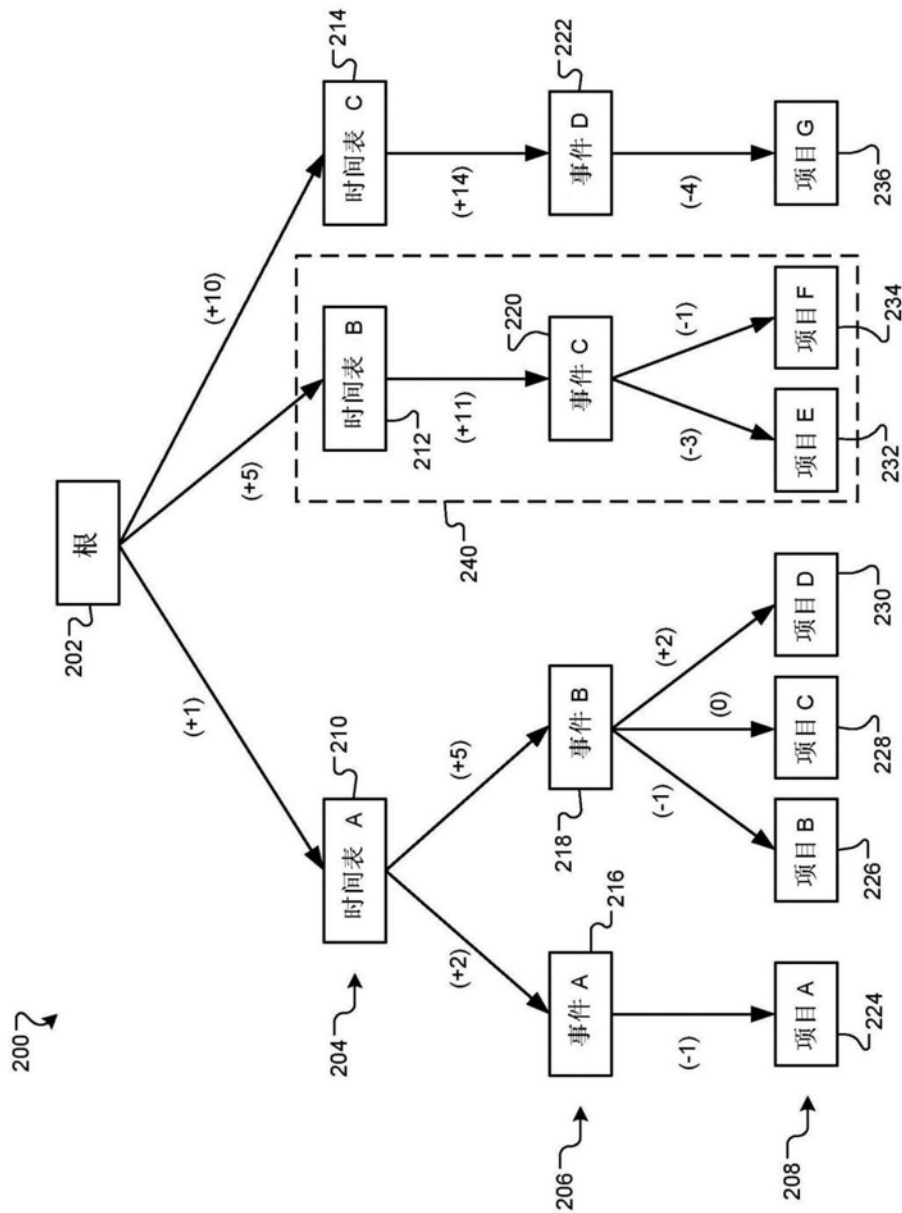


图2

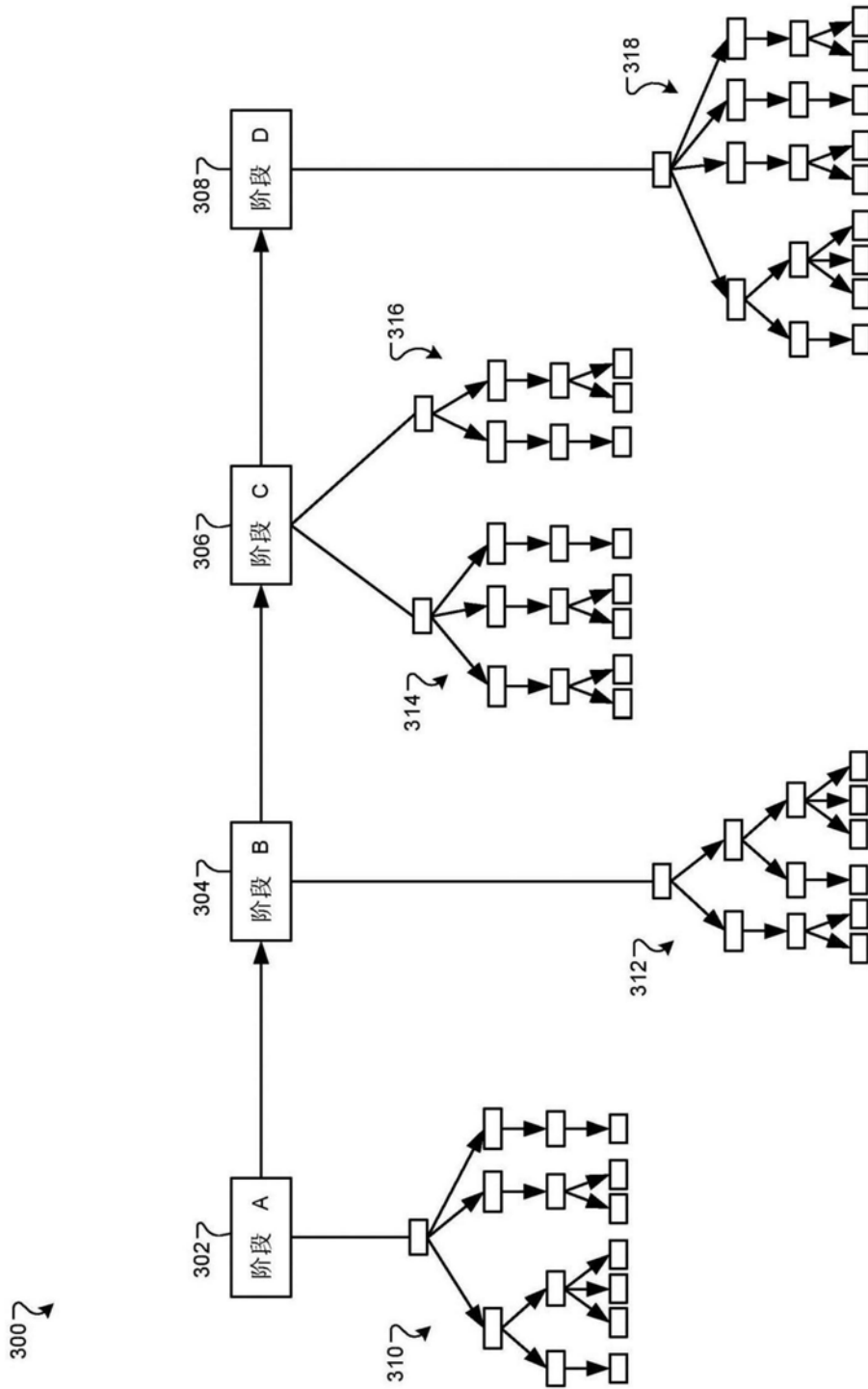


图3

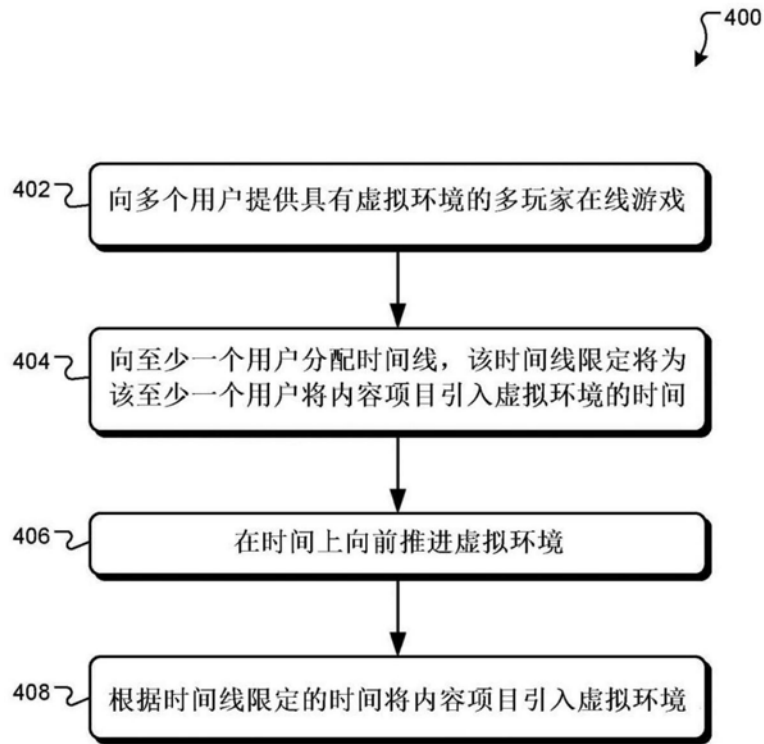


图4