

互动剧情跳转方法、互动剧情界面显示方法、装置及设备

本申请要求于2020年10月29日提交的、申请号为202011182523.4、发明名称为“互动剧情跳转方法、互动剧情界面显示方法、装置及设备”的中国专利申请的优先权，其全部内容通过引用结合在本申请中。

技术领域

本申请涉及计算机技术领域，特别涉及一种互动剧情跳转方法、互动剧情界面显示方法、装置及设备。

背景技术

互动影游是影视与游戏的互动结合，其包括根据影视作品改造的手游、页游、端游等游戏产品。互动影游以视频的形式展开游戏剧情，游戏剧情存在多个分支，随着剧情的发展会显示出不同的选项，玩家可以进行合理的选择，剧情会随着玩家的选择而走向不同的分支，播放不同的视频内容。

相关技术中，游戏产品的客户端可以显示剧情结构显示页面，剧情结构显示页面中包括多个剧情节点。其中，玩家当前所处的剧情对应的剧情节点为当前节点，玩家想要穿越到达的剧情节点为目标节点；玩家通过选择某些剧情节点到达当前节点，由这些经过的剧情节点组成的路径为当前路径，每个剧情节点上存储着玩家最近一次到达该剧情节点的路径为历史路径。当前路径和目标节点对应的历史路径之间存在交叉点，用户可以在剧情结构显示页面中，点击不同于当前节点的目标节点，并将当前路径在最后一个交叉点之前的部分路径，以及目标节点对应的历史路径在最后一个交叉点之后的部分路径进行拼接，以得到用以实现游戏剧情跳跃的剧情拼接路径。

上述相关技术实现的路径拼接方案，往往存在拼接的剧情路径逻辑不通的问题。

发明内容

本申请实施例提供了一种互动剧情跳转方法、互动剧情界面显示方法、装置及设备，能够保证拼接得到的路径在逻辑通畅的同时降低对剧情参数的影响。所述技术方案如下：

根据本申请实施例的一个方面，提供了一种互动剧情跳转方法，所述方法由计算机设备执行，所述方法包括：

获取当前节点信息，所述当前节点信息包括到达当前节点的第一剧情路径以及所述当前节点的剧情参数，所述当前节点用于指示用户当前所处的剧情情节，所述剧情参数反映用户在所述当前节点的用户状态；

响应于针对目标节点的剧情跳转指令，获取目标节点信息，所述目标节点信息包括所述目标节点的剧情参数以及到达过所述目标节点的历史剧情路径，所述剧情跳转指令用于指示剧情由所述当前节点跳转至所述目标节点；

对所述第一剧情路径以及所述历史剧情路径进行路径拼接，得到符合剧情逻辑规则且符合剧情参数条件的剧情拼接路径，所述剧情拼接路径通向所述目标节点；

基于所述剧情拼接路径，更新所述目标节点对应的剧情参数。

根据本申请实施例的一个方面，提供了一种互动剧情界面显示方法，所述方法由计算机设备执行，所述方法包括：

在剧情结构界面中显示当前节点以及到达所述当前节点的第一剧情路径，所述当前节点用于指示用户当前所处的剧情情节；

响应于针对目标节点的剧情跳转操作，获取所述当前节点的剧情参数以及到达过所述目标节点的历史剧情路径，所述剧情跳转操作用于指示剧情由所述当前节点跳转至所述目标节点

点；

对所述第一剧情路径与所述历史剧情路径进行路径拼接，显示拼接得到的剧情拼接路径，所述路径拼接用于在所述当前节点的剧情参数符合剧情参数条件的情况下，将所述第一剧情路径与所述历史剧情路径进行拼接，得到通往所述目标节点的且符合剧情逻辑规则的所述剧情拼接路径；

基于所述当前节点的剧情参数与所述剧情拼接路径，更新显示所述目标节点对应的剧情参数。

根据本申请实施例的一个方面，提供了一种互动剧情跳转装置，所述装置包括：

当前节点获取模块，用于获取当前节点信息，所述当前节点信息包括到达当前节点的第一剧情路径以及所述当前节点的剧情参数，所述当前节点用于指示用户当前所处的剧情情节，所述剧情参数反映用户在所述当前节点的用户状态；

目标节点获取模块，用于响应于针对目标节点的剧情跳转指令，获取目标节点信息，所述目标节点信息包括所述目标节点的剧情参数以及到达过所述目标节点的历史剧情路径，所述剧情跳转指令用于指示剧情由所述当前节点跳转至所述目标节点；

路径拼接模块，用于对所述第一剧情路径以及所述历史剧情路径进行路径拼接，得到符合剧情逻辑规则且符合剧情参数条件的剧情拼接路径，所述剧情拼接路径通向所述目标节点；

参数更新模块，用于基于所述剧情拼接路径，更新所述目标节点对应的剧情参数。

根据本申请实施例的一个方面，提供了一种互动剧情界面显示装置，所述装置包括：

界面显示模块，用于在剧情结构界面中显示当前节点以及到达所述当前节点的第一剧情路径，所述当前节点用于指示用户当前所处的剧情情节；

信息获取模块，用于响应于针对目标节点的剧情跳转操作，获取所述当前节点的剧情参数以及到达过所述目标节点的历史剧情路径，所述剧情跳转操作用于指示剧情由所述当前节点跳转至所述目标节点；

路径拼接模块，用于对所述第一剧情路径与所述历史剧情路径进行路径拼接，显示拼接得到的剧情拼接路径，所述路径拼接用于在所述当前节点的剧情参数符合剧情参数条件的情况下，将所述第一剧情路径与所述历史剧情路径进行拼接，得到通往所述目标节点的且符合剧情逻辑规则的所述剧情拼接路径；

参数更新模块，用于基于所述当前节点的剧情参数与所述剧情拼接路径，更新显示所述目标节点对应的剧情参数。

根据本申请实施例的一个方面，提供了一种计算机设备，所述设备包括处理器和存储器，所述存储器中存储有至少一条指令、至少一段程序、代码集或指令集，所述至少一条指令、所述至少一段程序、所述代码集或指令集由所述处理器加载并执行以实现上述互动剧情跳转方法。

根据本申请实施例的一个方面，提供了一种计算机设备，所述设备包括处理器和存储器，所述存储器中存储有至少一条指令、至少一段程序、代码集或指令集，所述至少一条指令、所述至少一段程序、所述代码集或指令集由所述处理器加载并执行以实现上述互动剧情界面显示方法。

根据本申请实施例的一个方面，提供了一种计算机可读存储介质，所述存储介质中存储有至少一条指令、至少一段程序、代码集或指令集，所述至少一条指令、所述至少一段程序、所述代码集或指令集由处理器加载并执行以实现上述互动剧情跳转方法。

根据本申请实施例的一个方面，提供了一种计算机可读存储介质，所述存储介质中存储有至少一条指令、至少一段程序、代码集或指令集，所述至少一条指令、所述至少一段程序、所述代码集或指令集由处理器加载并执行以实现上述互动剧情界面显示方法。

根据本申请实施例的一个方面，提供了一种计算机程序产品或计算机程序，该计算机程序产品或计算机程序包括计算机指令，该计算机指令存储在计算机可读存储介质中。计算机

设备的处理器从计算机可读存储介质读取该计算机指令，处理器执行该计算机指令，使得该计算机设备执行上述互动剧情跳转方法。

根据本申请实施例的一个方面，提供了一种计算机程序产品或计算机程序，该计算机程序产品或计算机程序包括计算机指令，该计算机指令存储在计算机可读存储介质中。计算机设备的处理器从计算机可读存储介质读取该计算机指令，处理器执行该计算机指令，使得该计算机设备执行上述互动剧情界面显示方法。

本申请实施例提供的技术方案可以带来如下有益效果：

通过基于到达当前节点的最新路径以及最新的剧情参数，在保证同时满足剧情逻辑规则以及剧情参数条件两方面限制的前提下，对当前路径与历史路径进行路径拼接，得到符合剧情逻辑规则且符合剧情参数条件的剧情拼接路径，保证了穿越操作后的新路径不会因为判断节点存在而导致拼接路径不通的问题，并且拼接出的路径满足最大程度的参数改变，有效保留最新的剧情状态。

应当理解的是，以上的一般描述和后文的细节描述仅是示例性和解释性的，并不能限制本申请。

附图说明

为了更清楚地说明本申请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申请的一些实施例，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。

图1是本申请一个实施例提供的应用程序运行环境的示意图；

图2是本申请一个实施例提供的互动剧情跳转方法的流程图；

图3是本申请一个实施例提供的剧情结构示意图；

图4是本申请另一个实施例提供的剧情结构示意图；

图5是本申请另一个实施例提供的剧情结构示意图；

图6是本申请另一个实施例提供的剧情结构示意图；

图7是本申请另一个实施例提供的剧情结构示意图；

图8是本申请另一个实施例提供的剧情结构示意图；

图9是本申请另一个实施例提供的互动剧情跳转方法的流程图；

图10是本申请一个实施例提供的互动剧情界面显示方法的流程图；

图11是本申请一个实施例提供的互动剧情跳转装置的框图；

图12是本申请一个实施例提供的互动剧情界面显示装置的框图；

图13是本申请一个实施例提供的计算机设备的结构框图。

具体实施方式

为使本申请的目的、技术方案和优点更加清楚，下面将结合附图对本申请实施方式作进一步地详细描述。以下示例性实施例中所描述的实施方式并不代表与本申请相一致的所有实施方式。相反，它们仅是与如所附权利要求书中所详述的、本申请的一些方面相一致的方法的例子。

首先，对本申请实施例涉及的相关术语进行解释说明，以便于本领域技术人员理解。

互动影游：也可以称为互动影视游戏，是影视与游戏的互动结合。互动影游包括根据影视作品改造的手游、页游、端游等游戏产品。在游戏中，剧情存在多个分支，随着剧情的发展会显示出不同的选项，玩家可以进行合理的选择，剧情会随着玩家的选择而走向不同的分支，播放不同的视频内容。互动影游的画面除了图片、动态影像、视频之外，还包括普通选项、互动选项、循环选项等游戏类的规则与方法以及游戏的运行程序，其中普通选项以及循

环选项为常规选项，互动选项并不多，主要包括角色转换、解密、对话选择、QTE（Quick Time Event，快速反应事件）等有限几种选项。其中，快速反应事件是指用户需根据指示在一定时间内操作（如触控操作、语音操作等），引导角色的动作，从而影响故事进展的交互事件。可选地，上述选项可以是文字选项，也可以是语音选项，还可以是图片选项，本申请实施例对此不作限定。

剧情节点：包含剧情视频块，剧情选项，参数变化操作与参数判断操作的属性集合的总称。可选地，一个剧情节点有一个或多个出口节点。可以根据节点上的选项主动通往某出口节点，也可以根据数值判断自动通往某出口节点。可选地，一个剧情节点的出口节点，是指该剧情节点的接下来的、且距离该剧情节点最近的节点。可选地，一个剧情节点的出口节点，与该剧情节点之间不存在其他剧情节点。

剧情选项：用于供玩家进行选择剧情的控件，每个剧情选项对应于不同的剧情分支，或者说通向不同的剧情节点，玩家可以进行合理的选择，剧情会随着玩家的选择而走向不同的分支，进一步，使得整个剧情走向不同的结局。用户可以自行从多个剧情选项中选择目标剧情选项，对应地，客户端可以获取到针对目标剧情选项的选择指令，该针对目标剧情选项的选择指令用于指示播放目标剧情选项对应的剧情视频。目标剧情选项可以是多个剧情选项中的任意一个剧情选项。

故事线：剧情节点的集合。

剧情参数：是一个全局变量，一些剧情节点上会对这些参数数值变量进行参数变化操作和参数判断操作。其中，剧情参数可以包括用户到达剧情节点对应的剧情时的剧情参数。比如，剧情参数可以包括道具参数、好感度值等参数值；此外，剧情参数还可以包括在剧情路径上的判定信息，比如，可以包括在剧情路径上的参数判定信息、进入下一游戏剧情的判定信息、道具获取判定信息，等等。

逻辑可达：发生在相邻节点间，反映前一节点到后一节点的路径符合逻辑规则。例如，从前一个节点a到后一个节点b，节点a的参数满足到达b节点的数值判断条件，或者到达b节点时没有数值判断操作，则认为节点a到b逻辑可达，否则逻辑不可达。

路径：若干相邻且逻辑可达的节点串联起来的节点集合。

路径标识：标识某条路径的唯一标识符，用整型数字表示。可选地，路径标识对应的路径可更新，比如，用户由剧情节点a走向下一个剧情节点b，那么此时路径为[a, b]，相应的，路径标识可以是1，当用户继续沿路径1走向下一剧情节点c时，路径1对应的剧情节点集合更新为[a, b, c]。

历史路径：每个剧情节点上存储着玩家最近一次到达该剧情节点的路径。

根节点：第一个视频块对应的节点，没有其他任何节点可以通向根节点。

当前节点：玩家当前所处的剧情对应的剧情节点；或者，可以是指玩家正在观看的视频块对应的节点。

当前路径：玩家从根节点通过选择某些剧情节点（观看某些剧情）到达当前节点，由经过的剧情节点组成当前路径。

交叉节点：两条或者多条路径上相同的剧情节点。

回溯操作：玩家从当前节点跳转到当前路径上其他剧情节点的操作。

穿越操作：玩家从当前节点跳转到所有走过的节点中不在当前路径上的其他节点的操作。

目标节点：在穿越操作和回溯操作中，用户希望前往的剧情节点。

当前状态：反映玩家到达当前节点时各剧情参数的值。到达一个节点会有多条不同的路径，如果每次按照不同的路径到达同一节点，那么到达该节点的剧情参数的数值可能会不同。

历史状态：每个节点都有一个历史状态，表示玩家最近一次到达该节点时各剧情参数的数值。

上下级关系：发生在两个节点之间，从节点a向其后面的节点遍历，经过若干节点可以

到达节点b, 则称节点a与b存在上下级关系。

起跳节点与目标节点: 发生在穿越/回溯操作中, 穿越操作/回溯操作前的当前节点为起跳节点, 希望穿越/回溯到的节点为目标节点。

请参考图 1, 其示出了本申请一个实施例提供的应用程序运行环境的示意图。该应用程序运行环境可以包括: 终端 10 和服务器 20。

终端 10 可以是诸如手机、平板电脑、游戏主机、多媒体播放设备、可穿戴设备、PC(Personal Computer, 个人计算机)、手持便携式游戏设备等电子设备。终端 10 中可以安装应用程序的客户端。

在本申请实施例中, 上述应用程序可以是任何能够显示画面与选项以供用户观看和选择的应用程序。典型地, 该应用程序为游戏应用程序, 如互动叙事类游戏、互动剧情类游戏、互动视频类游戏(如互动电影)、互动音乐类游戏、图像类冒险游戏、角色扮演冒险游戏和逻辑推理游戏, 等等。比如, 游戏可以设定一个大的游戏故事框架, 在游戏故事框架可以设定多个游戏剧情分支, 这些游戏剧情分支可以按照一定的剧情逻辑连接起来从而形成相应的游戏故事。玩家可以在游戏中进行互动操作, 终端可以基于玩家的互动操作发展相应的游戏剧情, 如随着剧情的发展会显示出不同的选项, 玩家可以进行合理的选择, 剧情会随着玩家的选择而走向不同的分支, 进一步, 使得整个游戏故事走向不同的结局。当然, 除了游戏应用程序之外, 其它类型的应用程序中也可以显示画面与选项以供用户观看和选择剧情。例如, 虚拟现实(Virtual Reality, VR)类应用程序、增强现实(Augmented Reality, AR)类应用程序、三维地图程序、军事仿真程序、社交类应用程序、互动娱乐类应用程序等, 本申请实施例对此不作限定。

可选地, 终端 10 中运行有上述应用程序的客户端。在一些实施例中, 上述应用程序是基于三维的虚拟环境引擎开发的应用程序, 比如该虚拟环境引擎是 Unity 引擎, 该虚拟环境引擎能够构建三维的虚拟环境、虚拟对象和虚拟道具等, 给用户带来更加沉浸式的游戏体验。

服务器 20 用于为终端 10 中的应用程序的客户端提供后台服务。例如, 服务器 20 可以是上述应用程序的后台服务器。服务器 20 可以是一台服务器, 也可以是由多台服务器组成的服务器集群, 或者是一个云计算服务中心, 可以提供云服务、云数据库、云计算、云函数、云存储、网络服务、云通信、中间件服务、域名服务、安全服务、CDN(Content Delivery Network, 内容分发网络)、以及大数据和人工智能平台等基础云计算服务。可选地, 服务器 20 同时为多个终端 10 中的应用程序提供后台服务。

可选地, 终端 10 和服务器 20 之间可通过网络 30 进行互相通信。

本申请实施例提供的互动剧情跳转方法、互动剧情界面显示方法, 各步骤的执行主体可以是服务器 20, 也可以是终端 10(如终端 10 中运行的应用程序的客户端), 还可以是由终端 10 和服务器 20 交互配合执行。为了便于说明, 在下述方法实施例中, 仅以各步骤的执行主体为计算机设备进行介绍说明, 但对此不构成限定。

下面, 通过几个实施例对本申请技术方案进行介绍说明。本申请技术方案主要提供一种在互动影视中应用的路径拼接方案, 该方案基于玩家当前路径与存储的历史路径按照规定算法进行路径拼接, 借助该方案可以使得玩家在穿越/回溯操作中, 在当前状态的基础上更新目标节点的最新状态。

请参考图 2, 其示出了本申请一个实施例提供的互动剧情跳转方法的流程图。该方法由计算机设备执行, 该方法可以包括以下几个步骤(210~240):

步骤 210, 获取当前节点信息。

当前节点信息包括到达当前节点的第一剧情路径以及当前节点的剧情参数。其中, 对剧情参数的理解介绍, 可以参考上述对本申请实施例涉及的相关术语进行的解释说明。

在一些实施例中，当前节点用于指示用户当前所处的剧情情节，第一剧情路径是到达当前节点的当前剧情路径，也可以理解为最新剧情路径，剧情参数反映用户在当前节点的用户状态。可选地，第一剧情路径是从根节点到达当前节点的剧情路径。在一些实施例中，客户端中登录有用户帐号。可选地，一个用户帐号对应有一个用户。可选地，用户状态包括用户当前节点的道具参数、好感度值等参数值；用户状态还可以包括用户当前节点的历史操作信息（如历史选择信息等）。

步骤 220，响应于针对目标节点的剧情跳转指令，获取目标节点信息。

可选地，接收到针对目标节点的剧情跳转操作之后，生成剧情跳转指令；剧情跳转指令用于指示剧情由当前节点跳转至目标节点。

在一些实施例中，目标节点信息包括目标节点的剧情参数以及到达过目标节点的历史剧情路径。可选地，历史剧情路径是最后一次到达过目标节点的剧情路径。可选地，历史剧情路径包括到达过目标节点的至少一条剧情路径。

步骤 230，对第一剧情路径以及历史剧情路径进行路径拼接，得到符合剧情逻辑规则且符合剧情参数条件的剧情拼接路径。

在一些实施例中，上述路径拼接是指将不同路径中部分路径或者全部路径进行拼接或组合处理，通过路径拼接可以得到一条新的剧情路径。可选地，剧情拼接路径通向目标节点。在进行路径探索过程中，当前节点可能会发生变化，因而剧情拼接路径可能经过当前节点，也可能不经过当前节点。在一些实施例中，基于当前节点对剧情路径进行逻辑判断，若确定其不符合剧情逻辑规则或剧情参数条件，则剧情拼接路径不经过当前节点。

在一些实施例中，上述剧情拼接路径符合剧情逻辑规则是指剧情拼接路径中各剧情节点之间的逻辑可达。其中，对逻辑可达和逻辑不可达的理解介绍，可以参考上述对本申请实施中对相关术语进行的解释说明。

在一些实施例中，上述剧情拼接路径符合剧情参数条件是指因路径拼接导致当前节点的剧情参数的数值被修改，但是数值变化范围要处于预设范围内，旨在尽量少的修改用户通过第一剧情路径到达当前节点所生成的剧情参数，并生成一条逻辑可达的通向目标节点的剧情路径。因此，在路径拼接时，需要从两方面对拼接路径进行限制，一是保证拼接路径逻辑可达，二是保证拼接路径中对于当前节点的剧情参数的改变要尽量小，符合用户想要达到的路径拼接结果，准备识别用户意图。

在示例性实施例中，上述步骤 230 可由如下步骤实现：

步骤 231，按照默认拼接规则，对第一剧情路径以及历史剧情路径进行路径拼接，得到默认剧情拼接路径。

在示例性实施例中，上述步骤 231 可以由如下过程实现：

确定当前路径与历史路径之间的剧情交叉节点；获取最后一个交叉节点前的当前路径以及最后一个交叉节点后的历史路径；将最后一个交叉节点前的当前剧情路径与最后一个交叉节点后的历史剧情路径进行路径拼接，得到默认剧情拼接路径。

步骤 232，对默认剧情拼接路径进行剧情逻辑校验处理。

在一些实施例中，上述剧情逻辑校验处理用于检测剧情路径中各剧情节点之间是否逻辑可达，若是，则可以判定该路径符合剧情逻辑规则，否则该路径不符合剧情逻辑规则。

在一些实施例中，对于上述步骤 232，可能得到两种情况，一是默认剧情拼接路径符合剧情逻辑规则，二是默认剧情拼接路径不符合剧情逻辑规则。

在一些实施例中，对于第一种情况，若默认剧情拼接路径符合剧情逻辑规则，则使用默认剧情拼接路径作为剧情拼接路径。此时说明该种情况下，由当前节点跳至目标节点的默认拼接路径的情况较为简单，不涉及数值判断操作或者数值修改操作，那么该默认拼接路径可以作为最终的剧情拼接路径，实现节点间的跳转。

对于第二种情况，若默认剧情拼接路径不符合剧情逻辑规则，则基于当前节点与目标节

点之间的上下级关系，对第一剧情路径以及历史剧情路径进行路径拼接处理，得到符合剧情逻辑规则且符合剧情参数条件的剧情拼接路径。

在一些实施例中，在默认剧情拼接路径不符合剧情逻辑规则的情况下，也不能简单地使用历史路径通往目标节点，这样会导致用户在当前节点得到的剧情参数被抹去，从而达不到预期的剧情效果。因此这里需要对第二种情况进行分析，针对不同的场景，选择不同的路径拼接方案。

本申请提供的技术方案的本意是在尽量少地修改用户所走的当前路径的前提下，生成一条可以到达目标节点的路径，并且要基于当前节点处的数值状态，更新目标节点的最新数值。在示例性实施例中，可根据当前节点与目标节点之间的上下级关系划分不同的路径拼接场景，一种场景是穿越前后的节点存在上下级关系，另一种是穿越前后的节点没有上下级关系。具体方案如下：

确定当前节点与目标节点之间是否存在上下级关系。可选地，在确定当前节点与目标节点之间的上下级关系之后，确定第一剧情路径与历史剧情路径之间的交叉节点中的目标判断节点，目标判断节点是指导致默认剧情拼接路径不符合剧情逻辑规则的涉及剧情参数判断的节点。

场景一：当前节点与目标节点之间存在上下级关系。

步骤 S10，在当前节点与目标节点之间存在上下级关系的情况下，在目标节点之前的剧情节点中进行深度优先搜索处理，得到符合剧情逻辑规则的剧情拼接节点。当前节点之后的剧情节点不会改变当前节点的剧情参数。

在示例性实施例中，因深度优先搜索处理 DFS 是一个递归操作，如果需要搜索的节点很多，那么显然单纯的 DFS 会很耗时且占用内存空间比较大，需要进行一些减枝操作，剔除掉一些不需要遍历的节点。因此，执行上述在目标节点之前的剧情节点中进行深度优先搜索处理的步骤之前，先进行一种计算量优化操作：

步骤 S00，获取目标判断节点之后的已读剧情节点。

上述目标判断节点是指导致默认剧情拼接路径不符合剧情逻辑规则以及涉及目标剧情参数判断的剧情节点。

相应地，上述步骤 S10 可由如下步骤 S1 替换：

S1，在目标判断节点之后的已读剧情节点中进行深度优先搜索处理，得到符合剧情逻辑规则的剧情拼接节点。

这样，深度优先搜索处理可以只遍历用户走过的节点（即已读剧情节点），原因也很容易理解，用户未走过的剧情节点不能凭空创造。

在示例性实施例中，还会存在判断节点之后的剧情结构较为复杂的情况，如图 3 提供的示例所示，其示例性示出了一种剧情结构示意图。如果判断节点 6 之后跟有若干层分叉+汇合的节点且用户曾经均到达过，那么此时仅通过遍历已读剧情节点的方法找到通向目标节点 9 的路径，仍然存在递归复杂度高的问题。针对如图 3 所示的情况，递归复杂度无疑会以指数速度递增，这样的耗时也不能满足要求。因此，需要在深度优先搜索处理的遍历过程中使用进一步的剪枝方案，具体如下（S20~S30）：

S20，检测深度优先搜索处理的递归次数。

S21，在递归次数未达到递归次数阈值的情况下，对当前搜索节点进行节点判断处理。

在一些实施例中，当前搜索节点是指当前时刻深度优先搜索处理到达的剧情节点，深度优先搜索处理的起始位置为目标判断节点的目标出口节点。

可选地，上述节点判断处理包括：检测当前搜索节点是否为目标节点，或者是否在历史剧情路径上；若是（即当前搜索节点为所述目标节点，或者在历史剧情路径上），则确定当前搜索节点通过节点判断处理，否则确定当前搜索节点未通过节点判断处理。

S22，在当前搜索节点未通过节点判断处理的情况下，将当前搜索节点的下一出口节点作

为当前搜索节点，以及累加递归次数，执行检测深度优先搜索处理的递归次数的步骤 S20。

S23，在当前搜索节点通过节点判断处理的情况下，对当前搜索节点进行剧情逻辑校验处理。

可选地，剧情逻辑校验处理包括：将第一剧情路径与在历史剧情路径上的当前搜索节点之后的历史路径进行路径拼接处理，生成临时拼接路径；若临时拼接路径符合剧情逻辑规则，则确定当前搜索节点通过剧情逻辑校验处理，否则确定当前搜索节点未通过剧情逻辑校验处理。

S24，若当前搜索节点通过剧情逻辑校验处理，则将当前搜索节点确定为剧情拼接节点。

S25，若当前搜索节点未通过剧情逻辑校验处理，则将当前搜索节点的下一出口节点作为当前搜索节点，以及累加递归次数，执行检测深度优先搜索处理的递归次数的步骤 S20。

S26，在递归次数达到递归次数阈值的情况下，检测当前搜索节点的所有出口节点是否均不在到达目标节点的历史剧情路径上。

S27，若当前搜索节点的所有出口节点均不在历史剧情路径上，则剔除当前搜索节点所在的递归分支。

S28，若当前搜索节点的出口节点中包含在历史剧情路径上的目标出口节点，则对目标出口节点进行剧情逻辑校验处理。

S29，若目标出口节点通过剧情逻辑校验处理，则将目标出口节点作为剧情拼接节点。

S30，若目标出口节点未通过剧情逻辑校验处理，则剔除当前搜索节点所在的递归分支。

可选地，在上述剪枝操作方案开始执行之前，可以先除去目标判断节点的出口节点中在历史路径上的剧情节点，得到剩余已读剧情节点。然后，在剩余已读剧情节点中进行深度优先搜索处理，得到符合剧情逻辑规则的剧情拼接节点。

上述过程可以简单理解为，如果遍历过程中的当前节点在历史路径上，则将该节点后面的历史路径与正在遍历的当前路径拼接成新路径，如果可达（即该新路径逻辑可达）则满足并退出；如果不可达（即该新路径逻辑不可达）则继续从该节点进行 DFS 递归查找。这里避免递归太深，我们可以设置一个上限，当递归深度达到该上限后，判断该节点出口的所有节点是否在目标节点的历史路径上，如果不在则直接返回，去寻找其他分支的节点，避免了后面非常耗时的递归。其中，该上限可以由相关技术人员根据实际情况进行设定，本申请实施例对此不作具体限定。虽然继续递归找下去有可能找到一条路径通往目标节点，但是本方案的目标是快速找到一条可用的路径即可，而历史路径此时就可以充分利用起来，这样可以快速减掉不必要的递归分支，极大的节省时间与空间的开销。

步骤 S2，基于剧情拼接节点生成剧情拼接路径。

可选地，上述步骤 S2 至少可通过两种方式进行实现，如下：

方式一：将第一剧情路径与在历史剧情路径上的剧情拼接节点之后的历史路径进行路径拼接，生成剧情拼接路径；

方式二：将符合剧情逻辑规则的临时拼接路径作为剧情拼接路径。

上述针对场景一所提供的方案的中心思想大致可以总结为：根据当前节点最新数值，更新目标节点数值。以互动游戏场景为例：其中的过程可以简单理解为：首先找到玩家当前路径与目标节点的历史路径的最后一个交叉节点。新路径的前半段为交叉节点前的当前路径，后半段为交叉节点后的历史路径，两段路径拼接作为前往目标节点的新路径。如果新路径逻辑可达则使用新路径；如果逻辑不可达，则找到逻辑不可达的判断节点，在判断节点之后不在历史路径上的节点（因为在历史路径上的点已经不可达）中，进行具有递归次数限制的深度优先搜索（DFS）直到找到目标节点。

在一个示例中，如图 4 所示，玩家最近一次到达节点 9 的历史路径为[1, 2, 5, 6, 7, 9]，当玩家回溯到节点 1 重新做出选择，将在节点 1 做出的选择修改为节点 4，此时到达节点 5 的最新路径是[1, 4, 5]。此时，若从节点 5 穿越到节点 9，应该在当前节点之后走过的节

点进行探索，依次找到满足逻辑可达且可以到达目标节点的节点集合作为最新路径。这里我们使用深度优先遍历算法（DFS）进行查找操作。以节点 5 处的数值为基础，深度遍历发现节点 6，当前节点改为节点 6，校验逻辑可达；将节点 6 塞入当前路径中，发现节点 6 的出口节点 7 和 8，将当前节点改为节点 7，校验发现逻辑不可达，当前节点改为 8，校验逻辑可达，依次类推，直到找到目标节点。如果节点 6 后还有其他节点但是没有走过，那么该节点就会被排除遍历范围外。

下面针对场景二所采取的技术方案进行介绍：

场景二：当前节点与目标节点之间不存在上下级关系。

在当前节点与目标节点之间不存在上下级关系的情况下，基于导致默认剧情拼接路径不符合剧情逻辑规则的目标剧情参数，更改默认剧情拼接路径中涉及目标剧情参数改变的剧情节点，得到符合剧情逻辑规则的剧情拼接路径。

其中，目标剧情参数改变用于满足剧情逻辑规则，剧情拼接路径中除目标剧情参数外的其他剧情参数符合剧情参数条件。

在示例性实施例中，上述方案可由如下步骤实现（A1~A8）：

A1，标记目标剧情参数。

A2，获取第一剧情路径与历史剧情路径之间所有的交叉节点。

A3，对第一剧情路径中每两个相邻的交叉节点之间的路径区间进行参数修改检查处理。

可选地，参数修改检查处理用于检查路径区间上的剧情节点是否涉及对被标记的目标剧情参数的数值修改操作。

A4，若目标路径区间内不存在涉及对目标剧情参数进行数值修改操作的剧情节点，则在目标路径区间内使用第一剧情路径，得到调整后的第一剧情路径。

A5，若目标路径区间内存在涉及对目标剧情参数进行数值修改操作的剧情节点，则在目标路径区间内使用历史剧情路径，得到调整后的第一剧情路径。

A6，将调整后的第一剧情路径与最后一个交叉节点后的历史剧情路径进行拼接，得到替换拼接路径。

A7，若替换拼接路径符合剧情逻辑规则，则确定替换拼接路径为剧情拼接路径，并停止对下一路径区间进行参数修改检查处理。

A8，若替换拼接路径不符合剧情逻辑规则，则对下一路径区间进行参数修改检查处理（对于下一路径区间，返回步骤 A3 开始执行）。

对于这种当前节点与目标节点不存在上下级关系，即无法从当前节点找到一条路径到达目标节点的情况，同样希望基于当前节点最新数值进行目标节点的数值更新。这样显然无法保证所有的数值都是最新，因为如果所有值都是最新，那么判断点就一定会指向当前节点的路径，目标节点则不满足逻辑可达。所以这里我们退而求其次，保证除该判断点所涉及的数值外，其他数值保持最新，而对于判断点的数值，我们可以做到尽量保持最新，即刚好满足判断点通向目标节点的路径逻辑可达。

方案思路为：首先生成默认拼接路径，若默认拼接路径满足逻辑可达则直接使用默认拼接路径；若默认拼接路径不满足逻辑可达，则找到导致默认拼接路径不满足逻辑可达的判断点，标记导致默认拼接路径不满足逻辑可达的目标参数。以当前路径为基础进行改造，判断点之后改为历史路径，找到当前路径与历史路径所有的交叉点，从后向前，每两个交叉点之间如果涉及目标参数的数值变化操作，则这两个交叉点之间使用对应的历史路径；如果不涉及目标参数的数值变化操作，则使用最新路径。在一些实施例中，每有一段路径区间内使用历史路径，就对改造后的新路径进行一次检查，判断是否可以到达目标节点，这样做的目的就是尽量少的改变最新的数值。如此从后向前，将交叉节点之间的路径一段一段拼接起来，即可达到预期需求。

在一个示例中，如图 5 所示，对于目标节点 18 来说，最近一次到达目标节点 18 的历史

路径为[1, 4, 5, 8, 9, 12, 13, 16, 17, 18]。到达当前节点 14 的当前路径为[1, 2, 5, 6, 9, 10, 13, 14, 15]。此时从当前节点 14 穿越到目标节点 18, 那么历史路径与当前路径最后一个交叉点是判断节点 13, 判断节点 13 为含有 B 参数的数值判断节点, 那么为了保证拼接路径能够到达目标节点 18, B 参数的数值需要满足由判断节点 13 走向普通节点 16 的数值判断条件, 因此可以确定此次路径拼接处理的预期效果为: 在参数 B 的数值满足判断节点 13 的数值判断的条件下不改变参数 A 的数值。

在此示例下, 应用本方案的路径拼接思路为: 找到当前路径与目标节点的历史路径的所有交叉节点, 对相邻的两个交叉节点确定的路径区间内的剧情节点进行检测, 检测目的在于发现路径区间内的节点中是否包括涉及目标参数 B 的数值变化操作节点。如果在某路径区间内存在涉及目标参数 B 的数值变化操作节点, 则在该路径区间内选择历史路径上的剧情节点; 否则使用当前路径上的剧情节点。同时, 为了尽量少的对玩家选择的当前路径进行修改, 一旦发现改变某个节点的选择后, 生成的拼接路径满足逻辑可达条件, 能够通向目标节点, 则停止其他交叉节点间的判断与改变。

如此, 最后得到的最新路径为[1, 4, 5, 8, 9, 12, 13, 16, 17, 18], 节点 1 与节点 5 之间包括涉及对参数 B 进行数值变化操作的节点 2、3、4, 因此在节点 1 与节点 5 之间选择穿过节点 4 的历史路径作为此路径区间内的拼接路径。节点 9 与节点 13 之间包括涉及对参数 B 进行数值变化操作的节点 10、11、12, 因此在节点 9 与节点 13 之间选择穿过节点 12 的历史路径作为此路径区间内的拼接路径。节点 5 与节点 9 之间不包括涉及对参数 B 进行数值变化操作的节点, 仅包括涉及对参数 A 进行数值变化操作的节点 6、7、8, 因此在节点 5 与节点 9 之间选择穿过节点 6 的当前路径作为此路径区间内的拼接路径。这样, 就不会改变用户当前路径得到的参数 A 的最新数值, 同时也使参数 B 满足在判断节点 13 指向剧情节点 16 的数值判断条件, 因此, 该拼接路径达到了路径拼接的预期效果。

如果只改变一个对数值 B 变化的节点即可满足判断节点的数值到达目标节点, 则生成的新路径可能为[1, 2, 5, 6, 9, 12, 13, 16, 17, 18]。

步骤 240, 基于剧情拼接路径, 更新目标节点对应的剧情参数。

在一些实施例中, 基于剧情拼接路径中的各剧情节点中存在的参数变化操作以及参数判断操作, 更新目标节点对应的剧情参数。

综上所述, 本申请实施例提供的技术方案, 通过基于到达当前节点的最新路径以及最新的剧情参数, 在保证同时满足剧情逻辑规则以及剧情参数条件两方面限制的前提下, 对当前路径与历史路径进行路径拼接, 得到符合剧情逻辑规则且符合剧情参数条件的剧情拼接路径, 保证了穿越操作后的新路径不会因为判断节点存在而导致拼接路径不通的问题, 并且拼接出的路径满足最大程度的参数改变, 有效保留最新的剧情状态。

下面结合几个示例对上述有益效果进行解释说明:

相关技术中所使用的方案一般是在穿越操作中, 找到玩家当前路径与目标节点的历史路径的最后一个交叉节点。新路径的前半段为交叉节点前的当前路径, 后半段为交叉节点后的历史路径, 两段路径拼接作为前往目标节点的新路径。在一个示例中, 如图 6 所示, 其示例性示出了一种剧情结构示意图。其中, 到达剧情节点 G 的历史路径为加粗的路径 60, 我们看路径 60 上几个代表性的节点, 用户若想到达节点 G 需要在几个分歧节点中选择 A 和 C。在另一个示例中, 如图 7 所示, 其示例性示出了一种剧情结构示意图。如果当前节点在 E, 到达 E 的路径为图 7 的加粗路径 70, 此时用户在几个代表节点处的选择为 B 和 D。结合上述两个示例, 在本示例中, 如图 8 所示, 此时, 从当前节点 E 穿越到 G, 到达 G 的路径 80 变为: 此时代表节点的分歧由之前的 A, C 变为最新的 B, D, F, E。如果节点 E 和 F 涉及数值改变, 那么到达 G 点的最新数值也会改变。

上述方案在简单的剧情场景下进行节点穿越操作没有问题, 但是如果应用场景中在穿越

的目标节点的历史路径上有数值判断的节点,那么其方案将无法使用。

这里通过图4列举一个示例进行说明。在这个示例中,节点2、3、4的样式表示含有数值变化操作的节点,节点6的样式表示含有数值判断操作的节点,节点1的样式3表示没有数值判断操作和数值变化操作的普通节点,图4中的节点2、3、4会对某一数值进行修改,节点6对该数值进行值判断,比如值大于某一阈值 x 则剧情走向节点7,否则走向节点8。其中, x 的具体数值可以由相关技术人员根据实际情况进行设定,本申请实施例对此不作具体限定。在一种可能的情况中,玩家最近一次到达节点9是按照路径[1,2,5,6,7,9]到达的。如果玩家回溯到节点1进行重新选择从而修改数值,然后穿越到节点9,此时按照相关技术提供的路径拼接方案进行路径拼接,得到的拼接路径可能无法到达,因为在节点6处的判断完全有可能指向节点8而非拼接后的节点7。当路径逻辑不可达时仍会使用目标节点的历史路径,即又调整到了[1,2,5,6,7,9],此时到达目标节点9时的剧情参数的数值没有因为之前回溯到节点1的修改而更新,无法满足用户的预期效果。

而本申请的技术方案提供的思路是基于当前最新路径的基础上进行的路径探索,保证了穿越操作后的新路径不会因为判断节点存在而导致拼接路径不通的问题。此外,本方案对应用场景进行了进一步扩展:穿越操作前后,拼接出的路径满足最大程度的只改变某些数值,其他数值保持不变。

请参考图9,其示出了本申请另一个实施例提供的互动剧情跳转方法的流程图。该方法由计算机设备执行,该方法可以包括以下几个步骤(1~22):

步骤1,响应于针对目标节点的剧情跳转操作,启动路径拼接流程。

步骤2,按照默认拼接规则生成默认剧情拼接路径。

步骤3,判断默认剧情拼接路径是否逻辑可达。若是,执行步骤21;否则执行步骤4。

步骤4,判断起跳节点与目标节点之间是否存在上下级关系。若是,则执行步骤5;否则执行步骤13。

步骤5,查找导致默认剧情拼接路径不可达的判断节点,除去该判断节点出口节点中在历史路径上的节点,在剩余节点中进行深度优先搜索处理。

步骤6,判断此时递归深度是否达到预设上限,若是,则执行步骤10;否执行步骤7。

步骤7,判断当前搜索节点是否是目标节点,或者是否在目标节点的历史路径上。若是,则执行步骤8;否则执行步骤9。

步骤8,将当前路径与当前节点之后的历史路径拼接生成新路径,新路径可达执行步骤21;否则执行步骤9。

步骤9,从当前搜索节点的出口节点中继续进行深度优先搜索处理,执行步骤6。

步骤10,判断当前搜索节点的所有出口节点是否均不在目标节点的历史路径上。若是,则执行步骤11;否则执行步骤12。

步骤11,剔除该递归分支,对其他节点进行深度优先搜索处理,执行步骤6。

步骤12,将当前路径与在历史路径上的出口节点之后的历史路径拼接生成的新路径,新路径逻辑可达则执行步骤21;否则执行步骤11。

步骤13,标记出导致目标节点逻辑不可达的判断节点对应的参数。

步骤14,以当前路径为基础进行路径改造。

步骤15,获取当前路径与目标节点的历史路径的所有交叉节点。

步骤16,将最后一个交叉节点到目标节点的历史路径作为新路径的最后一段路径区间。

步骤17,从后向前,对每两个交叉节点之间的路径区间进行检查,判断路径区间上的节点是否涉及对被标记参数的修改操作。若是,则执行步骤18;否则执行步骤19。

步骤18,新路径在这两交叉节点之间的一小段路径区间使用历史路径,对新路径判断是否可达目标节点。可达执行步骤21;否则执行步骤20。

步骤 19, 新路径在这两交叉节点之间的一小段路径区间使用当前路径。

步骤 20, 此处两个交叉节点处理完毕, 前往前两个交叉节点。执行步骤 17。

步骤 21, 使用该新路径作为最终路径, 更新目标节点状态。

步骤 22, 路径拼接完成。

综上所述, 本申请实施例提供的技术方案, 通过基于到达当前节点的最新路径以及最新的剧情参数, 在保证同时满足剧情逻辑规则以及剧情参数条件两方面限制的前提下, 对当前路径与历史路径进行路径拼接, 得到符合剧情逻辑规则且符合剧情参数条件的剧情拼接路径, 保证了穿越操作后的新路径不会因为判断节点存在而导致拼接路径不通的问题, 并且拼接出的路径满足最大程度的参数改变, 有效保留最新的剧情状态。

此外, 本申请提供的方法对互动影视的剧情设计给予了更大的灵活度, 对较为复杂的剧情逻辑提供了技术支持, 适用场景更加广泛。

请参考图 10, 其示出了本申请一个实施例提供的互动剧情界面显示方法的流程图。该方法由计算机设备执行, 该方法可以包括以下几个步骤 (1001~1004):

步骤 1001, 在剧情结构界面中显示当前节点以及到达当前节点的第一剧情路径。

可选地, 当前节点用于指示用户当前所处的剧情情节。

游戏剧情结构界面: 用于向玩家展示整个游戏剧情结构的界面, 玩家可通过游戏剧情结构界面了解自己经历过游戏剧情、玩家的游戏剧情路线、以及各游戏剧情之间的关联关系等, 比如, 游戏剧情结构界面可以为游戏故事线界面等。本申请实施例中, 游戏剧情结构界面可以包括玩家之前进入过的, 且存储有数据的剧情节点, 以便玩家可以重新选择进入历史剧情节点的剧情。

步骤 1002, 响应于针对目标节点的剧情跳转操作, 获取当前节点的剧情参数以及到达过目标节点的历史剧情路径。

可选地, 剧情跳转操作用于指示剧情由当前节点跳转至目标节点。在一些实施例中, 根据接收到的剧情跳转操作生成剧情跳转指令, 基于当前跳转指令获取当前节点的剧情参数以及到达过目标节点的历史剧情路径。

步骤 1003, 对第一剧情路径与历史剧情路径进行路径拼接, 显示拼接得到的剧情拼接路径。

在一些实施例中, 可以由客户端进行路径拼接, 也可以由服务器进行路径拼接, 还可以由客户端和服务器交互进行路径拼接, 本申请实施例对此不作具体限定。

可选地, 在用户从剧情画面返回游戏剧情结构界面后显示拼接得到的剧情拼接路径。

可选地, 在用户点击目标节点后显示拼接得到的剧情拼接路径。

可选地, 在用户点击目标节点后显示目标节点对应的剧情画面展示控件的同时, 显示拼接得到的剧情拼接路径。

路径拼接用于在当前节点的剧情参数符合剧情参数条件的情况下, 将第一剧情路径与历史剧情路径进行拼接, 得到通往目标节点的且符合剧情逻辑规则的剧情拼接路径。

步骤 1004, 基于当前节点的剧情参数与剧情拼接路径, 更新显示目标节点对应的剧情参数。

可选地, 在剧情结构界面中显示更新后的目标节点对应的剧情参数。

本实施例是上文提供的互动剧情跳转方法在产品侧的实施例, 涉及的相关术语解释请参照上文实施例中有关术语的解释。示例性地, 在叙事类产品中, 有些节点会对数值进行修改, 选择不同的节点会对数值进行不同的修改, 从不同的路径到达同一节点时数值有可能不同。当到达对该数值的判断节点处时, 数值的不同就会通往不同的节点, 导致玩家看到不同的剧情内容。玩家可以在故事线的若干节点上来回穿越重新做出选择来修改数值, 看过的节点视频不需要重复观看, 从而快速体验到新剧情的效果。

综上所述，本申请实施例提供的技术方案，通过基于到达当前节点的最新路径以及最新的剧情参数，在保证同时满足剧情逻辑规则以及剧情参数条件两方面限制的前提下，对当前路径与历史路径进行路径拼接，得到符合剧情逻辑规则且符合剧情参数条件的剧情拼接路径，保证了穿越操作后的新路径不会因为判断节点存在而导致拼接路径不通的问题，并且拼接出的路径满足最大程度的参数改变，有效保留最新的剧情状态。

下述为本申请装置实施例，可用于执行本申请方法实施例。对于本申请装置实施例中未披露的细节，请参照本申请方法实施例。

请参考图 11，其示出了本申请一个实施例提供的互动剧情跳转装置的框图。该装置具有实现上述互动剧情跳转方法的功能。该装置 1100 可以包括：当前节点获取模块 1101、目标节点获取模块 1102、路径拼接模块 1103 以及参数更新模块 1104。

当前节点获取模块 1101，用于获取当前节点信息，所述当前节点信息包括到达当前节点的第一剧情路径以及所述当前节点的剧情参数，所述当前节点用于指示用户当前所处的剧情情节，所述剧情参数反映用户在所述当前节点的用户状态。

目标节点获取模块 1102，用于响应于针对目标节点的剧情跳转指令，获取目标节点信息，所述目标节点信息包括所述目标节点的剧情参数以及到达过所述目标节点的历史剧情路径，所述剧情跳转指令用于指示剧情由所述当前节点跳转至所述目标节点。

路径拼接模块 1103，用于对所述第一剧情路径以及所述历史剧情路径进行路径拼接，得到符合剧情逻辑规则且符合剧情参数条件的剧情拼接路径，所述剧情拼接路径通向所述目标节点。

参数更新模块 1104，用于基于所述剧情拼接路径，更新所述目标节点对应的剧情参数。

在示例性实施例中，所述路径拼接模块 1103，用于：

按照默认拼接规则，对所述第一剧情路径以及所述历史剧情路径进行路径拼接，得到默认剧情拼接路径；

若所述默认剧情拼接路径符合所述剧情逻辑规则，则使用所述默认剧情拼接路径作为所述剧情拼接路径；

若所述默认剧情拼接路径不符合所述剧情逻辑规则，则基于所述当前节点与所述目标节点之间的上下级关系，对所述第一剧情路径以及所述历史剧情路径进行路径拼接，得到符合所述剧情逻辑规则且符合所述剧情参数条件的所述剧情拼接路径。

在示例性实施例中，所述路径拼接模块 1103，用于：

确定所述当前节点与所述目标节点之间是否存在上下级关系；

在所述当前节点与所述目标节点之间存在所述上下级关系的情况下，在所述目标节点之前的剧情节点中进行深度优先搜索处理，得到符合所述剧情逻辑规则的剧情拼接节点，所述当前节点之后的剧情节点不会改变所述当前节点的剧情参数；基于所述剧情拼接节点生成所述剧情拼接路径；

在所述当前节点与所述目标节点之间不存在所述上下级关系的情况下，基于导致所述默认剧情拼接路径不符合所述剧情逻辑规则的目标剧情参数，更改所述默认剧情拼接路径中涉及所述目标剧情参数改变的剧情节点，得到符合所述剧情逻辑规则的所述剧情拼接路径；其中，所述剧情拼接路径中除所述目标剧情参数外的其他剧情参数符合所述剧情参数条件。

在示例性实施例中，所述路径拼接模块 1103，还用于：

获取目标判断节点之后的已读剧情节点，所述目标判断节点是指导致所述默认剧情拼接路径不符合所述剧情逻辑规则以及涉及所述目标剧情参数判断的剧情节点，所述已读剧情节点是指已到达过的剧情节点；

所述在所述目标节点之前的剧情节点中进行深度优先搜索处理，得到符合所述剧情逻辑规则的剧情拼接节点，包括：

在所述目标判断节点之后的已读剧情节点中进行所述深度优先搜索处理，得到符合所述剧情逻辑规则的剧情拼接节点。

在示例性实施例中，所述路径拼接模块 1103，用于：

除去所述目标判断节点的出口节点中在所述历史路径上的剧情节点，得到剩余已读剧情节点；

在所述剩余已读剧情节点中进行所述深度优先搜索处理，得到符合所述剧情逻辑规则的剧情拼接节点。

在示例性实施例中，所述路径拼接模块 1103，还用于：

检测所述深度优先搜索处理的递归次数；

在所述递归次数未达到递归次数阈值的情况下，对当前搜索节点进行节点判断处理，所述当前搜索节点是指当前时刻所述深度优先搜索处理到达的剧情节点，所述深度优先搜索处理的起始位置为所述目标判断节点的目标出口节点；在所述当前搜索节点未通过所述节点判断处理的情况下，将所述当前搜索节点的下一出口节点作为所述当前搜索节点，以及累加所述递归次数，执行所述检测所述深度优先搜索处理的递归次数的步骤；在所述当前搜索节点通过所述节点判断处理的情况下，对所述当前搜索节点进行剧情逻辑校验处理；若所述当前搜索节点通过所述剧情逻辑校验处理，则将所述当前搜索节点确定为所述剧情拼接节点；若所述当前搜索节点未通过所述剧情逻辑校验处理，则将所述当前搜索节点的下一出口节点作为所述当前搜索节点，以及累加所述递归次数，执行所述检测所述深度优先搜索处理的递归次数的步骤；

在所述递归次数达到所述递归次数阈值的情况下，检测所述当前搜索节点的所有出口节点是否均不在到达所述目标节点的所述历史剧情路径上；若所述当前搜索节点的所有出口节点均不在所述历史剧情路径上，则剔除所述当前搜索节点所在的递归分支；若所述当前搜索节点的出口节点中包含在所述历史剧情路径上的目标出口节点，则对所述目标出口节点进行所述剧情逻辑校验处理；若所述目标出口节点通过所述剧情逻辑校验处理，则将所述目标出口节点作为所述剧情拼接节点；若所述目标出口节点未通过所述剧情逻辑校验处理，则剔除所述当前搜索节点所在的递归分支。

在示例性实施例中，所述路径拼接模块 1103，用于：

检测所述当前搜索节点是否为所述目标节点，或者是否在所述历史剧情路径上；若所述当前搜索节点为所述目标节点，或者在所述历史剧情路径上，则确定所述当前搜索节点通过所述节点判断处理，否则确定所述当前搜索节点未通过所述节点判断处理；

所述对所述当前搜索节点进行剧情逻辑校验处理，包括：

将所述第一剧情路径与在所述历史剧情路径上的所述当前搜索节点之后的历史路径进行路径拼接，生成临时拼接路径；若所述临时拼接路径符合所述剧情逻辑规则，则确定所述当前搜索节点通过所述剧情逻辑校验处理，否则确定所述当前搜索节点未通过所述剧情逻辑校验处理。

在示例性实施例中，所述路径拼接模块 1103，用于：

将所述第一剧情路径与在所述历史剧情路径上的所述剧情拼接节点之后的历史路径进行路径拼接，生成所述剧情拼接路径；

或者，

将符合所述剧情逻辑规则的临时拼接路径作为所述剧情拼接路径。

在示例性实施例中，所述路径拼接模块 1103，还用于：

标记所述目标剧情参数；

获取所述第一剧情路径与所述历史剧情路径之间所有的交叉节点；

对所述第一剧情路径中每两个相邻的交叉节点之间的路径区间进行参数修改检查处理，所述参数修改检查处理用于检查所述路径区间上的剧情节点是否涉及对被标记的所述目标剧

情参数的数值修改操作；

若目标路径区间内不存在涉及对所述目标剧情参数进行数值修改操作的剧情节点，则在所述目标路径区间内使用所述第一剧情路径在所述目标路径区间对应的相邻交叉节点之间形成的路径；

若所述目标路径区间内存在涉及对所述目标剧情参数进行数值修改操作的剧情节点，则在所述目标路径区间内使用所述历史剧情路径在所述目标路径区间对应的相邻交叉节点之间形成的路径，得到调整后的第一剧情路径；

将所述调整后的第一剧情路径与所述最后一个交叉节点后的历史剧情路径进行拼接，得到替换拼接路径；

若所述替换拼接路径符合所述剧情逻辑规则，则确定所述替换拼接路径为所述剧情拼接路径，并停止对下一路径区间进行所述参数修改检查处理；

若所述替换拼接路径符合所述剧情逻辑规则，则对下一路径区间进行所述参数修改检查处理。

在示例性实施例中，所述路径拼接模块 1103，还用于：

确定所述第一剧情路径与所述历史剧情路径之间的交叉节点中的目标判断节点，所述目标判断节点是指导致所述默认剧情拼接路径不符合所述剧情逻辑规则的涉及剧情参数判断的节点。

综上所述，本申请实施例提供的技术方案，通过基于到达当前节点的最新路径以及最新的剧情参数，在保证同时满足剧情逻辑规则以及剧情参数条件两方面限制的前提下，对当前路径与历史路径进行路径拼接，得到符合剧情逻辑规则且符合剧情参数条件的剧情拼接路径，保证了穿越操作后的新路径不会因为判断节点存在而导致拼接路径不通的问题，并且拼接出的路径满足最大程度的参数改变，有效保留最新的剧情状态。

请参考图 12，其示出了本申请另一个实施例提供的互动剧情界面显示装置的框图。该装置具有实现上述互动剧情界面显示方法的功能。该装置 1200 可以包括：界面显示模块 1201、信息获取模块 1202、路径拼接模块 1203 以及参数更新模块 1204。

界面显示模块 1201，用于在剧情结构界面中显示当前节点以及到达所述当前节点的第一剧情路径，所述当前节点用于指示用户当前所处的剧情情节。

信息获取模块 1202，用于响应于针对目标节点的剧情跳转操作，获取所述当前节点的剧情参数以及到达过所述目标节点的历史剧情路径，所述剧情跳转操作用于指示剧情由所述当前节点跳转至所述目标节点。

路径拼接模块 1203，用于对所述第一剧情路径与所述历史剧情路径进行路径拼接，显示拼接得到的剧情拼接路径，所述路径拼接用于在所述当前节点的剧情参数符合剧情参数条件的情况下，将所述第一剧情路径与所述历史剧情路径进行拼接，得到通往所述目标节点的且符合剧情逻辑规则的所述剧情拼接路径。

参数更新模块 1204，用于基于所述当前节点的剧情参数与所述剧情拼接路径，更新显示所述目标节点对应的剧情参数。

综上所述，本申请实施例提供的技术方案，通过基于到达当前节点的最新路径以及最新的剧情参数，在保证同时满足剧情逻辑规则以及剧情参数条件两方面限制的前提下，对当前路径与历史路径进行路径拼接，得到符合剧情逻辑规则且符合剧情参数条件的剧情拼接路径，保证了穿越操作后的新路径不会因为判断节点存在而导致拼接路径不通的问题，并且拼接出的路径满足最大程度的参数改变，有效保留最新的剧情状态。

请参考图 13，其示出了本申请一个实施例提供的计算机设备 1100 的结构框图。该计算机设备 1300 可以是诸如手机、平板电脑、游戏主机、多媒体播放设备、可穿戴设备、PC(Personal

Computer, 个人计算机)、手持便携式游戏设备等电子设备。该计算机设备用于实施上述实施例中提供的方法。该计算机设备可以是图 1 所示应用程序运行环境中的终端 10, 或者服务器 20。具体来讲:

通常, 计算机设备 1300 包括有: 处理器 1301 和存储器 1302。

处理器 1301 可以包括一个或多个处理核心, 比如 4 核心处理器、8 核心处理器等。处理器 1301 可以采用 DSP (Digital Signal Processing, 数字信号处理)、FPGA (Field Programmable Gate Array, 现场可编程门阵列)、PLA (Programmable Logic Array, 可编程逻辑阵列) 中的至少一种硬件形式来实现。处理器 1301 也可以包括主处理器和协处理器, 主处理器是用于对在唤醒状态下的数据进行处理的处理单元, 也称 CPU (Central Processing Unit, 中央处理器); 协处理器是用于对在待机状态下的数据进行处理的低功耗处理器。在一些实施例中, 处理器 1301 可以在集成有 GPU (Graphics Processing Unit, 图像处理器), GPU 用于负责显示屏所需要显示的内容的渲染和绘制。一些实施例中, 处理器 1301 还可以包括 AI (Artificial Intelligence, 人工智能) 处理器, 该 AI 处理器用于处理有关机器学习的计算操作。

存储器 1302 可以包括一个或多个计算机可读存储介质, 该计算机可读存储介质可以是非暂态的。存储器 1302 还可包括高速随机存取存储器, 以及非易失性存储器, 比如一个或多个磁盘存储设备、闪存存储设备。在一些实施例中, 存储器 1302 中的非暂态的计算机可读存储介质用于存储至少一个指令, 至少一段程序、代码集或指令集, 所述至少一条指令、至少一段程序、代码集或指令集, 且经配置以由一个或者一个以上处理器执行, 以实现上述方法。

在一些实施例中, 计算机设备 1300 还可选包括有: 外围设备接口 1303 和至少一个外围设备。处理器 1301、存储器 1302 和外围设备接口 1303 之间可以通过总线或信号线相连。各个外围设备可以通过总线、信号线或电路板与外围设备接口 1303 相连。具体地, 外围设备包括: 射频电路 1304、显示屏 1305、摄像头组件 1306、音频电路 1307、定位组件 1308 和电源 1309 中的至少一种。

本领域技术人员可以理解, 图 13 中示出的结构并不构成对计算机设备 1300 的限定, 可以包括比图示更多或更少的组件, 或者组合某些组件, 或者采用不同的组件布置。

在示例性实施例中, 还提供了一种计算机可读存储介质, 所述存储介质中存储有至少一条指令、至少一段程序、代码集或指令集, 所述至少一条指令、所述至少一段程序、所述代码集或所述指令集在被处理器执行时以实现上述互动剧情跳转方法。

在示例性实施例中, 还提供了一种计算机可读存储介质, 所述存储介质中存储有至少一条指令、至少一段程序、代码集或指令集, 所述至少一条指令、所述至少一段程序、所述代码集或所述指令集在被处理器执行时以实现上述互动剧情界面显示方法。

可选地, 该计算机可读存储介质可以包括: ROM (Read Only Memory, 只读存储器)、RAM (Random Access Memory, 随机存取记忆体)、SSD (Solid State Drives, 固态硬盘) 或光盘等。其中, 随机存取记忆体可以包括 ReRAM (Resistance Random Access Memory, 电阻式随机存取记忆体) 和 DRAM (Dynamic Random Access Memory, 动态随机存取存储器)。

在示例性实施例中, 还提供了一种计算机程序产品或计算机程序, 该计算机程序产品或计算机程序包括计算机指令, 该计算机指令存储在计算机可读存储介质中。计算机设备的处理器从计算机可读存储介质读取该计算机指令, 处理器执行该计算机指令, 使得该计算机设备执行上述互动剧情跳转方法。

在示例性实施例中, 还提供了一种计算机程序产品或计算机程序, 该计算机程序产品或计算机程序包括计算机指令, 该计算机指令存储在计算机可读存储介质中。计算机设备的处理器从计算机可读存储介质读取该计算机指令, 处理器执行该计算机指令, 使得该计算机设备执行上述互动剧情界面显示方法。

应当理解的是，在本文中提及的“多个”是指两个或两个以上。“和/或”，描述关联对象的关联关系，表示可以存在三种关系，例如，A 和/或 B，可以表示：单独存在 A，同时存在 A 和 B，单独存在 B 这三种情况。字符“/”一般表示前后关联对象是一种“或”的关系。另外，本文中描述的步骤编号，仅示例性示出了步骤间的一种可能的执行先后顺序，在一些其它实施例中，上述步骤也可以不按照编号顺序来执行，如两个不同编号的步骤同时执行，或者两个不同编号的步骤按照与图示相反的顺序执行，本申请实施例对此不作限定。

以上所述仅为本申请的示例性实施例，并不用以限制本申请，凡在本申请的精神和原则之内，所作的任何修改、等同替换、改进等，均应包含在本申请的保护范围之内。