



(12) 发明专利

(10) 授权公告号 CN 111589140 B

(45) 授权公告日 2023.03.14

(21) 申请号 202010408457.1

审查员 蒋婷

(22) 申请日 2020.05.14

(65) 同一申请的已公布的文献号

申请公布号 CN 111589140 A

(43) 申请公布日 2020.08.28

(73) 专利权人 腾讯科技(深圳)有限公司

地址 518057 广东省深圳市南山区高新区
科技中一路腾讯大厦35层

(72) 发明人 魏嘉城 胡勋 粟山东

(74) 专利代理机构 北京三高永信知识产权代理
有限责任公司 11138

专利代理人 张所明

(51) Int.Cl.

A63F 13/56 (2014.01)

A63F 13/58 (2014.01)

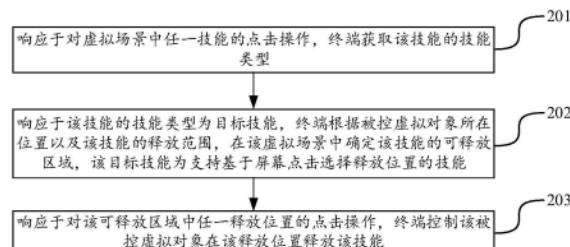
权利要求书3页 说明书19页 附图7页

(54) 发明名称

虚拟对象的控制方法、装置、终端及存储介质

(57) 摘要

本申请公开了一种虚拟对象的控制方法、装置、终端及存储介质，属于计算机领域。本申请通过响应于对虚拟场景中任一技能的点击操作，获取该技能的技能类型，响应于该技能的技能类型为支持基于屏幕点击选择释放位置的目标技能，根据被控虚拟对象所在位置以及该技能的释放范围，在该虚拟场景中确定该技能的可释放区域，响应于对该可释放区域中任一释放位置的点击操作，控制该被控虚拟对象在该释放位置释放该技能，无需基于技能轮盘以及技能摇杆进行复杂的拖拽操作，具有更高的瞄准精度、更低的操作难度、更高效的人机交互效率，改善了用户体验。



1.一种虚拟对象的控制方法,其特征在于,所述方法包括:

在检测到对虚拟场景中任一位置的点击操作时,响应于所述位置上存在用户界面控件,获取所述用户界面控件的控件类型;

响应于所述用户界面控件的控件类型为技能控件,获取其他技能控件的触发状态,所述其他技能控件为除了点击操作对应的技能控件之外的其他技能的技能控件;其中,具有屏蔽标识的技能为处于触发状态的技能,所述屏蔽标识为终端在技能的技能控件被点击时分配的;

响应于所述其他技能控件不处于触发状态,确定检测到对所述虚拟场景中技能的点击操作;

响应于对所述虚拟场景中技能的点击操作,获取所述虚拟场景中技能的技能类型;

响应于所述虚拟场景中技能的技能类型为目标技能,在所述虚拟场景中显示所述技能的释放倒计时,所述释放倒计时用于提示所述虚拟场景中技能剩余多长时间即将释放;

在所述释放倒计时过程中,根据被控虚拟对象所在位置以及所述虚拟场景中技能的释放范围,在所述虚拟场景中确定所述技能的可释放区域,所述目标技能为支持基于屏幕点击选择释放位置的技能;

响应于对所述虚拟场景中地图控件的点击操作,在所述地图控件中确定被点击的目标区域;

将所述地图控件中的所述目标区域映射为所述虚拟场景中的目标场景,所述目标场景中包括所述可释放区域;

在所述目标场景中检测对所述可释放区域中任一释放位置的点击操作;

响应于对所述可释放区域中任一释放位置的点击操作,控制所述被控虚拟对象在所述释放位置释放所述技能。

2.根据权利要求1所述的方法,其特征在于,所述响应于对所述可释放区域中任一释放位置的点击操作,控制所述被控虚拟对象在所述释放位置释放所述技能包括:

响应于对所述可释放区域中任一释放位置的点击操作,获取在所述屏幕中所述释放位置的屏幕坐标;

将所述屏幕坐标映射为在所述虚拟场景中所述释放位置的场景坐标;

控制所述被控虚拟对象在所述场景坐标释放所述技能。

3.根据权利要求1所述的方法,其特征在于,若所述可释放区域为圆形区域,所述根据被控虚拟对象所在位置以及所述技能的释放范围,在所述虚拟场景中确定所述技能的可释放区域包括:

在所述虚拟场景中,将所述被控虚拟对象所在位置确定为所述可释放区域的圆心,将所述技能的释放范围确定为所述可释放区域的半径;

在所述虚拟场景中为所述可释放区域增加释放模板,所述释放模板用于表示在所述释放模板之外的区域释放所述技能无效。

4.根据权利要求1所述的方法,其特征在于,所述响应于对所述可释放区域中任一释放位置的点击操作,控制所述被控虚拟对象在所述释放位置释放所述技能之前,所述方法还包括:

响应于对所述可释放区域中任一释放位置的拖动操作,基于拖动操作的拖动轨迹,在

所述虚拟场景中更新所述释放位置。

5. 一种虚拟对象的控制装置,其特征在于,所述装置包括:

获取模块,用于在检测到对虚拟场景中任一位置的点击操作时,响应于所述位置上存在用户界面控件,获取所述用户界面控件的控件类型;响应于所述用户界面控件的控件类型为技能控件,获取其他技能控件的触发状态,所述其他技能控件为除了点击操作对应的技能控件之外的其他技能的技能控件;其中,具有屏蔽标识的技能为处于触发状态的技能,所述屏蔽标识为终端在技能的技能控件被点击时分配的;响应于所述其他技能控件不处于触发状态,确定检测到对所述虚拟场景中技能的点击操作;响应于对所述虚拟场景中任一技能的点击操作,获取所述虚拟场景中技能的技能类型;

确定模块,用于响应于所述虚拟场景中技能的技能类型为目标技能,在所述虚拟场景中显示所述技能的释放倒计时,所述释放倒计时用于提示所述虚拟场景中技能剩余多长时间即将释放;在所述释放倒计时过程中,根据被控虚拟对象所在位置以及所述虚拟场景中技能的释放范围,在所述虚拟场景中确定所述技能的可释放区域,所述目标技能为支持基于屏幕点击选择释放位置的技能;

映射检测模块,用于响应于对所述虚拟场景中地图控件的点击操作,在所述地图控件中确定被点击的目标区域;将所述地图控件中的所述目标区域映射为所述虚拟场景中的目标场景,所述目标场景中包括所述可释放区域;在所述目标场景中检测对所述可释放区域中任一释放位置的点击操作;

控制模块,用于响应于对所述可释放区域中任一释放位置的点击操作,控制所述被控虚拟对象在所述释放位置释放所述技能。

6. 根据权利要求5所述的装置,其特征在于,所述控制模块用于:

响应于对所述可释放区域中任一释放位置的点击操作,获取在所述屏幕中所述释放位置的屏幕坐标;

将所述屏幕坐标映射为在所述虚拟场景中所述释放位置的场景坐标;

控制所述被控虚拟对象在所述场景坐标释放所述技能。

7. 根据权利要求5所述的装置,其特征在于,若所述可释放区域为圆形区域,所述确定模块用于:

在所述虚拟场景中,将所述被控虚拟对象所在位置确定为所述可释放区域的圆心,将所述技能的释放范围确定为所述可释放区域的半径;

在所述虚拟场景中为所述可释放区域增加释放模板,所述释放模板用于表示在所述释放模板之外的区域释放所述技能无效。

8. 根据权利要求5所述的装置,其特征在于,所述装置还包括:

显示模块,用于响应于所述技能为目标技能,在所述虚拟场景中显示所述技能的释放倒计时或者释放提示信息中至少一项。

9. 一种终端,其特征在于,所述终端包括一个或多个处理器和一个或多个存储器,所述一个或多个存储器中存储有至少一条程序代码,所述至少一条程序代码由所述一个或多个处理器加载并执行以实现如权利要求1至权利要求4任一项所述的虚拟对象的控制方法所执行的操作。

10. 一种存储介质,其特征在于,所述存储介质中存储有至少一条程序代码,所述至少

一条程序代码由处理器加载并执行以实现如权利要求1至权利要求4任一项所述的虚拟对象的控制方法所执行的操作。

虚拟对象的控制方法、装置、终端及存储介质

技术领域

[0001] 本申请涉及计算机技术领域,特别涉及一种虚拟对象的控制方法、装置、终端及存储介质。

背景技术

[0002] 随着计算机技术的发展以及终端功能的多样化,出现了越来越多的手机游戏,其中,MOBA(Multiplayer Online Battle Arena,多人在线战术竞技)游戏逐渐成为手机游戏中极为重要的一类游戏。在目前的MOBA游戏中,玩家可以控制目标虚拟对象向某一位置释放技能,以在该位置上触发该技能的作用效果。

[0003] 通常,玩家在释放技能时,需要先点击某一技能控件,在该技能控件上触发显示技能轮盘,技能轮盘与技能在虚拟场景中的释放范围呈映射关系,再通过拖动技能轮盘中的技能摇杆来调整技能在释放范围内的释放方向和释放位置,在调整完毕后玩家松手,才能在技能摇杆所映射的位置上释放出这一技能。

[0004] 应用上述基于技能轮盘的释放方式,由于技能轮盘与释放范围呈映射关系,而技能轮盘的大小通常是不变的,对于一些释放范围较大的技能,导致其映射比例过大,玩家在拖动技能摇杆的过程中,即使产生轻微移动都会导致最终释放位置发生大范围的偏移,导致瞄准精度较低、操作难度较大、人机交互效率较低,进而影响用户体验。

发明内容

[0005] 本申请实施例提供了一种虚拟对象的控制方法、装置、终端及存储介质,能够提升MOBA中释放技能的瞄准精度、降低操作难度、提升人机交互效率、优化用户体验。该技术方案如下:

[0006] 一方面,提供了一种虚拟对象的控制方法,该方法包括:

[0007] 响应于对虚拟场景中任一技能的点击操作,获取所述技能的技能类型;

[0008] 响应于所述技能的技能类型为目标技能,根据被控虚拟对象所在位置以及所述技能的释放范围,在所述虚拟场景中确定所述技能的可释放区域,所述目标技能为支持基于屏幕点击选择释放位置的技能;

[0009] 响应于对所述可释放区域中任一释放位置的点击操作,控制所述被控虚拟对象在所述释放位置释放所述技能。

[0010] 一方面,提供了一种虚拟对象的控制装置,该装置包括:

[0011] 获取模块,用于响应于对虚拟场景中任一技能的点击操作,获取所述技能的技能类型;

[0012] 确定模块,用于响应于所述技能的技能类型为目标技能,根据被控虚拟对象所在位置以及所述技能的释放范围,在所述虚拟场景中确定所述技能的可释放区域,所述目标技能为支持基于屏幕点击选择释放位置的技能;

[0013] 控制模块,用于响应于对所述可释放区域中任一释放位置的点击操作,控制所述

被控虚拟对象在所述释放位置释放所述技能。

[0014] 在一种可能实施方式中,所述装置还包括:

[0015] 更新模块,用于响应于对所述可释放区域中任一释放位置的拖动操作,基于拖动操作的拖动轨迹,在所述虚拟场景中更新所述释放位置。

[0016] 一方面,提供了一种终端,该终端包括一个或多个处理器和一个或多个存储器,该一个或多个存储器中存储有至少一条程序代码,该至少一条程序代码由该一个或多个处理器加载并执行以实现如上述任一种可能实现方式的虚拟对象的控制方法所执行的操作。

[0017] 一方面,提供了一种存储介质,该存储介质中存储有至少一条程序代码,该至少一条程序代码由处理器加载并执行以实现如上述任一种可能实现方式的虚拟对象的控制方法所执行的操作。

[0018] 本申请实施例提供的技术方案带来的有益效果至少包括:

[0019] 通过响应于对虚拟场景中任一技能的点击操作,获取该技能的技能类型,响应于该技能的技能类型为支持基于屏幕点击选择释放位置的目标技能,根据被控虚拟对象所在位置以及该技能的释放范围,在该虚拟场景中确定该技能的可释放区域,响应于对该可释放区域中任一释放位置的点击操作,控制该被控虚拟对象在该释放位置释放该技能,这种通过先点击技能再点击释放位置进行技能释放的人机交互方式,能够基于简单的两次点击操作释放技能,无需基于技能轮盘以及技能摇杆进行复杂的拖拽操作,并且避免了由技能轮盘映射到释放范围时产生的偏移误差,因此具有更高的瞄准精度、更低的操作难度,从而带来更高效的人机交互效率,改善了用户体验。

附图说明

[0020] 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。

[0021] 图1是本申请实施例提供的一种虚拟对象的控制方法的实施环境示意图;

[0022] 图2是本申请实施例提供的一种虚拟对象的控制方法的流程图;

[0023] 图3是本申请实施例提供的一种虚拟对象的控制方法的流程图;

[0024] 图4是本申请实施例提供的一种界面示意图;

[0025] 图5是本申请实施例提供的一种界面示意图;

[0026] 图6是本申请实施例提供的一种界面示意图;

[0027] 图7是本申请实施例提供的一种界面示意图;

[0028] 图8是本申请实施例提供的一种触屏技能释放原理性示意图;

[0029] 图9是本申请实施例提供的一种虚拟对象的控制装置的结构示意图;

[0030] 图10是本申请实施例提供的一种终端的结构示意图。

具体实施方式

[0031] 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。

[0032] 本申请中术语“第一”“第二”等字样用于对作用和功能基本相同的相同项或相似项进行区分，应理解，“第一”、“第二”、“第n”之间不具有逻辑或时序上的依赖关系，也不对数量和执行顺序进行限定。

[0033] 本申请中术语“至少一个”是指一个或多个，“多个”的含义是指两个或两个以上，例如，多个第一位置是指两个或两个以上的第一个位置。

[0034] 为了便于理解本申请实施例的技术过程，下面对本申请实施例所涉及的一些名词进行解释：

[0035] 虚拟场景：是应用程序在终端上运行时显示（或提供）的虚拟场景。该虚拟场景可以是对真实世界的仿真环境，也可以是半仿真半虚构的虚拟环境，还可以是纯虚构的虚拟环境。虚拟场景可以是二维虚拟场景、2.5维虚拟场景或者三维虚拟场景中的任意一种，本申请实施例对虚拟场景的维度不加以限定。例如，虚拟场景可以包括天空、陆地、海洋等，该陆地可以包括沙漠、城市等环境元素，用户可以控制虚拟对象在该虚拟场景中进行移动。可选地，该虚拟场景还可以用于至少两个虚拟对象之间的虚拟场景对战，在该虚拟场景中具有可供至少两个虚拟对象使用的虚拟资源。可选地，该虚拟场景中可以包括对称的两个区域，属于两个敌对阵营的虚拟对象分别占据其中一个区域，并以摧毁对方区域深处的目标建筑/据点/基地/水晶来作为胜利目标，其中，对称的区域比如左下角区域和右上角区域，又比如左侧中部区域和右侧中部区域等。

[0036] 虚拟对象：是指在虚拟场景中的可活动对象。该可活动对象可以是虚拟人物、虚拟动物、动漫人物等，比如：在虚拟场景中显示的人物、动物、植物、油桶、墙壁、石块等。该虚拟对象可以是该虚拟场景中的一个虚拟的用于代表用户的虚拟形象。在虚拟场景中可以包括多个虚拟对象，每个虚拟对象在虚拟场景中具有自身的形状和体积，占据虚拟场景中的一部分空间。可选地，当虚拟场景为三维虚拟场景时，可选地，虚拟对象可以是一个三维立体模型，该三维立体模型可以是基于三维人体骨骼技术构建的三维角色，同一个虚拟对象可以通过穿戴不同的皮肤来展示出不同的外在形象。在一些实施例中，虚拟对象也可以采用2.5维或2维模型来实现，本申请实施例对此不加以限定。

[0037] 可选地，该虚拟对象可以是通过客户端上的操作进行控制的玩家角色，也可以是通过训练设置在虚拟场景对战中的人工智能（Artificial Intelligence, AI），还可以是设置在虚拟场景互动中的非玩家角色（Non-Player Character, NPC）。可选地，该虚拟对象可以在虚拟场景中进行竞技的虚拟人物。可选地，该虚拟场景中参与互动的虚拟对象的数量可以是预先设置的，也可以是根据加入互动的客户端的数量动态确定的。

[0038] MOBA（Multiplayer Online Battle Arena, 多人在线战术竞技）游戏：是一种在虚拟场景中提供若干个据点，处于不同阵营的用户控制虚拟对象在虚拟场景中对战，占领据点或摧毁敌对阵营据点的游戏。例如，MOBA游戏可将用户分成至少两个敌对阵营，分属至少两个敌对阵营的不同虚拟队伍分别占据各自的地图区域，以某一种胜利条件作为目标进行竞技。该胜利条件包括但不限于：占领据点或摧毁敌对阵营据点、击杀敌对阵营的虚拟对象、在指定场景和时间内保证自身的存活、抢夺到某种资源、在指定时间内互动比分超过对方中的至少一种。例如，MOBA游戏可将用户分成两个敌对阵营，将用户控制的虚拟对象分散在虚拟场景中互相竞争，以摧毁或占领敌方的全部据点作为胜利条件。

[0039] 可选地，每个虚拟队伍包括一个或多个虚拟对象，比如1个、2个、3个或5个，根据参

与战术竞技中各队伍内虚拟对象的数量,可以将战术竞技划分为1V1竞技比拼、2V2竞技比拼、3V3竞技比拼、5V5竞技比拼等,其中,1V1是指“1对1”的意思,这里不做赘述。

[0040] 可选地,MOBA游戏可以以局(或称为回合)为单位来进行,每局战术竞技的地图可以相同,也可以不同。一局MOBA游戏的持续时间是从游戏开始的时刻至达成胜利条件的时刻。

[0041] 在MOBA游戏中,用户可以控制虚拟对象在该虚拟场景的天空中自由下落、滑翔或者打开降落伞进行下落等,在陆地上中跑动、跳动、爬行、弯腰前行等,也可以控制虚拟对象在海洋中游泳、漂浮或者下潜等,在此仅以上述场景进行举例说明,本申请实施例对此不作具体限定。

[0042] 在MOBA游戏中,用户还可以控制虚拟对象释放技能从而与其他虚拟对象进行战斗,例如,该技能的技能类型可以包括攻击技能、防御技能、治疗技能、辅助技能、斩杀技能等,每个虚拟对象都可以具有各自固定的一个或多个技能,而不同的虚拟对象通常具有不同的技能,不同的技能可以产生不同的作用效果。比如,若虚拟对象释放攻击技能击中了敌对虚拟对象,那么会对敌对虚拟对象造成一定的伤害,通常表现为扣除敌对虚拟对象的一部分虚拟生命值,又比如,若虚拟对象释放治疗技能命中了友方虚拟对象,那么会对友方虚拟对象产生一定的治疗,通常表现为回复友方虚拟对象的一部分虚拟生命值,其他各类技能均可以产生相应的作用效果,这里不再一一枚举。

[0043] 以下,对本申请涉及的系统架构进行介绍。

[0044] 图1是本申请实施例提供的一种虚拟对象的控制方法的实施环境示意图。参见图1,该实施环境包括:第一终端120、服务器140和第二终端160。

[0045] 第一终端120安装和运行有支持虚拟场景的应用程序。该应用程序可以是MOBA游戏、大型多人在线角色扮演游戏(Massively Multiplayer Online Role Playing Game, MMORPG)、第一人称射击游戏(First-Person Shooting game,FPS)、第三人称射击游戏、虚拟现实应用程序、三维地图程序或者多人枪战类生存游戏中的任意一种。第一终端120可以是第一用户使用的终端,第一用户使用第一终端120操作位于虚拟场景中的第一虚拟对象进行活动,该活动包括但不限于:释放技能、调整身体姿态、爬行、步行、奔跑、骑行、跳跃、驾驶、拾取、射击、攻击、投掷中的至少一种。示意性的,第一虚拟对象是第一虚拟人物,比如仿真人角色或动漫人物角色。

[0046] 服务器140可以包括一台服务器、多台服务器、云计算平台或者虚拟化中心中的至少一种。服务器140用于为支持虚拟场景的应用程序提供后台服务。可选地,服务器140可以承担主要计算工作,第一终端120和第二终端160可以承担次要计算工作;或者,服务器140承担次要计算工作,第一终端120和第二终端160承担主要计算工作;或者,服务器140、第一终端120和第二终端160三者之间采用分布式计算架构进行协同计算。

[0047] 第二终端160安装和运行有支持虚拟场景的应用程序。该应用程序可以是MOBA游戏、MMORPG游戏、FPS游戏、第三人称射击游戏、虚拟现实应用程序、三维地图程序或者多人枪战类生存游戏中的任意一种。第二终端160可以是第二用户使用的终端,第二用户使用第二终端160操作位于虚拟场景中的第二虚拟对象进行活动,该活动包括但不限于:释放技能、调整身体姿态、爬行、步行、奔跑、骑行、跳跃、驾驶、拾取、射击、攻击、投掷中的至少一种。示意性的,第二虚拟对象是第二虚拟人物,比如仿真人角色或动漫人物角色。

[0048] 第一终端120以及第二终端160可以通过有线或无线通信方式与服务器140进行直接或间接地连接,本申请实施例在此不对连接方式进行限制。

[0049] 在一些实施例中,第一终端120控制的第一虚拟对象和第二终端160控制的第二虚拟对象处于同一虚拟场景中,此时第一虚拟对象可以在虚拟场景中与第二虚拟对象进行互动。上述第一虚拟对象以及第二虚拟对象可以为敌对关系,例如,第一虚拟对象与第二虚拟对象可以属于不同的队伍和阵营,敌对关系的虚拟对象之间,可以以互相释放技能的方式进行对战,比如,第一虚拟对象向第二虚拟对象释放攻击技能。在另一些实施例中,第一虚拟对象以及第二虚拟对象还可以为队友关系,例如,第一虚拟人物和第二虚拟人物可以属于同一个队伍、同一个组织、具有好友关系或具有临时性的通讯权限,在这种情况下,第一虚拟对象可以向第二虚拟对象释放治疗技能。

[0050] 上述服务器140可以是独立的物理服务器,也可以是多个物理服务器构成的服务器集群或者分布式系统,还可以是提供云服务、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、内容分发网络(Content Delivery Network, CDN)以及大数据和人工智能平台等基础云计算服务的云服务器。

[0051] 上述第一终端120或者第二终端160可以是智能手机、平板电脑、笔记本电脑、台式计算机、智能音箱、智能手表、MP3(Moving Picture Experts Group Audio Layer III, 动态影像专家压缩标准音频层面3)播放器、MP4(Moving Picture Experts Group Audio Layer IV, 动态影像专家压缩标准音频层面4)播放器、电子书阅读器等,但并不局限于此。

[0052] 需要说明的是,第一终端120和第二终端160上安装的应用程序可以是相同的,或两个终端上安装的应用程序是不同操作系统平台的同一类型应用程序。第一终端120可以泛指多个终端中的一个,第二终端160可以泛指多个终端中的一个,本实施例仅以第一终端120和第二终端160来举例说明。第一终端120和第二终端160的设备类型可以相同或不同。本领域技术人员可以知晓,上述终端的数量可以更多或更少。比如上述终端可以仅为一个,或者上述终端为几十个或几百个,或者更多数量。本申请实施例对终端的数量和设备类型不加以限定。

[0053] 图2是本申请实施例提供的一种虚拟对象的控制方法的流程图。参见图2,该实施例应用于终端,该终端可以是图1所示的第一终端120或者第二终端160,该实施例包括下述步骤:

[0054] 201、响应于对虚拟场景中任一技能的点击操作,终端获取该技能的技能类型。

[0055] 其中,该终端可以为任一用户所使用的终端,该终端当前所控制的虚拟对象为被控虚拟对象,被控虚拟对象可以通过释放技能来与虚拟场景中的其他虚拟对象之间进行对抗行为,这里所涉及释放的技能可以是被控虚拟对象所固有的一个或多个技能,或者,该技能还可以是被控虚拟对象在装配了某种虚拟道具之后应用该虚拟道具而释放的一个或多个技能,本申请实施例不对技能来源进行具体限定。可选地,该技能的技能类型可以包括攻击技能、防御技能、治疗技能、辅助技能、斩杀技能等,本申请实施例不对技能类型进行具体限定。

[0056] 在一些实施例中,终端响应于用户的开局操作,显示操作界面,该操作界面可以包括虚拟场景、虚拟对象的选择控件、虚拟道具的选择控件等。该终端可以实时检测用户对该操作界面中各个虚拟对象的选择操作,将被选择的虚拟对象确定为本局出战的被控虚拟对

象,可选地,该选择操作可以为点击操作、长按操作等,也可以为对任一虚拟对象所对应快捷键的触发操作,本申请实施例对此不作限定。由于在一些MOBA游戏中,用户可以一次性选择多个虚拟对象出战,并在对局中同时控制多个虚拟对象作战,因此在本申请实施例中的被控虚拟对象是指用户当前所操控的虚拟对象,俗称为“主操虚拟对象”。

[0057] 用户在操作界面中选择好被控虚拟对象之后,终端开启本局游戏,加载虚拟场景,并在虚拟场景中显示被控虚拟对象以及该被控虚拟对象所具有的至少一个技能的技能控件,该技能控件用于提供技能的用户交互接口,比如,该技能控件可以是以图形按钮的方式在虚拟场景中悬浮显示,该图形按钮可以是圆形按钮、矩形按钮或者不规则形状的按钮,本申请实施例不对技能控件的显示方式进行具体限定。可选地,在技能控件上可以展示对应技能的技能标识,该技能标识可以是一个标识图案,或者是一串标识文本,或者是一串标识码。

[0058] 在一些实施例中,终端还可以在虚拟场景中显示其他的UI (User Interface, 用户界面) 控件,比如地图控件、商店控件、移动摇杆等,本公开实施例不对虚拟场景中显示的控件内容进行具体限定。

[0059] 用户可以通过点击该虚拟场景中任一技能的技能控件,使得终端响应于用户对该技能的点击操作,获取该技能的技能类型,其中,该技能类型可以包括目标技能和除了目标技能之外的技能,该目标技能为支持基于屏幕点击选择释放位置的技能。上述过程可以视为检测该技能的技能类型是否为目标技能的过程,终端可以为目标技能的技能类型增加目标技能标识,从而通过检测该技能的技能类型是否具有目标技能标识,即可确定该技能的技能类型是否为目标技能,当检测到该技能的技能类型具有目标技能标识时,确定该技能的技能类型为目标技能,执行下述步骤202,否则,当检测到该技能的技能类型没有目标技能标识时,确定该技能的技能类型不是目标技能,退出流程,可选地,该目标技能标识可以为IsFullScreenSkill标识,由于目标技能可以在终端的整个屏幕中通过点击操作来触发释放,这种释放方式区别于相关技术中基于技能轮盘的操作机制,因此目标技能也可以俗称为“全屏技能”、“触屏技能”。

[0060] 在相关技术中,用户在释放技能时,需要先点击某一技能控件,在该技能控件上触发显示技能轮盘,技能轮盘与技能在虚拟场景中的释放范围呈映射关系,再通过拖动技能轮盘中的技能摇杆来调整技能在释放范围内的释放方向和释放位置,在调整完毕后用户松手,才能在技能摇杆所映射的位置上释放出这一技能。这种基于技能轮盘的释放方式,由于技能轮盘与释放范围呈映射关系,而技能轮盘的大小通常是不变的,对于一些释放范围较大的技能,导致其映射比例过大,用户在拖动技能摇杆的过程中,即使产生轻微移动都会导致最终释放位置发生大范围的偏移,导致瞄准精度较低、操作难度较大、人机交互效率较低,进而影响用户体验。

[0061] 而在本申请实施例中,目标技能则无需基于技能轮盘进行释放,而是一种可以直接基于用户对屏幕的点击操作来选择释放位置的技能,由于用户能够直接在虚拟场景中点击某一释放位置来释放技能,达到“所见即所得”的交互效果,而无需经过技能轮盘到虚拟场景之间的映射,因此即使针对一些释放范围较大的技能仍然可以保持良好的瞄准精度,也即能够大大提升释放技能的瞄准精度、降低释放技能的操作难度,从而提升人机交互效率,优化用户体验,以下,将针对目标技能如何进行释放进行详细说明。

[0062] 202、响应于该技能的技能类型为目标技能，终端根据被控虚拟对象所在位置以及该技能的释放范围，在该虚拟场景中确定该技能的可释放区域，该目标技能为支持基于屏幕点击选择释放位置的技能。

[0063] 在上述过程中，当检测到该技能的技能类型为目标技能时，终端可以将被控虚拟所在位置确定为可释放区域的圆心，将该技能的释放范围确定为该可释放区域的半径，从而确定出一个圆形的可释放区域。比如，被控虚拟对象位于屏幕左下角时，用户点击了技能1，技能1的释放范围为六尺，因此在虚拟场景中，将屏幕左下角中被控虚拟对象所在位置周围六尺的圆形区域确定为技能1的可释放区域。

[0064] 在一些实施例中，终端还可以在该虚拟场景中为该可释放区域增加释放模板，该释放模板用于表示在该释放模板之外的区域释放该技能无效。此时如果检测到用户对屏幕中任一位置点的点击操作，终端通过检测该位置点是否位于该释放模板中，即可确定该位置点是否位于可释放区域内，具体地，当该位置点位于该释放模板中，确定该位置点位于可释放区域内，执行下述步骤203，否则，当该位置点不位于该释放模板中，确定该位置点位于可释放区域外，则不会释放技能，将本次点击操作视为无效操作或者误操作。

[0065] 203、响应于对该可释放区域中任一释放位置的点击操作，终端控制该被控虚拟对象在该释放位置释放该技能。

[0066] 在上述过程中，当终端检测到用户对该可释放区域中任一释放位置的点击操作之后，终端控制该被控虚拟对象在该释放位置上释放该技能，结算该技能的作用效果。根据技能类型的不同，作用效果的结算规则也不尽相同。

[0067] 在一个示例性场景中，对于常规的攻击技能来说，终端可以检测在该释放位置上是否存在其他虚拟对象，若该释放位置上存在其他虚拟对象，确定该攻击技能命中其他虚拟对象，并对其他虚拟对象的虚拟生命值进行一定数值的扣除，否则，若该释放位置上不存在其他虚拟对象，确定该攻击技能未命中其他虚拟对象，意味着本次释放的技能偏离了目标。

[0068] 在一个示例性场景中，对于一些AOE (Area Of Effect, 范围攻击) 技能来说，用户所点击的释放位置其实是技能攻击范围的中心点，终端可以检测在该释放位置周围的目标距离内是否存在其他虚拟对象，若该释放位置周围的目标距离内存在其他虚拟对象，确定该攻击技能命中其他虚拟对象，并对其他虚拟对象的虚拟生命值进行一定数值的扣除，否则，若该释放位置周围的目标距离内不存在其他虚拟对象，确定该攻击技能未命中其他虚拟对象，意味着本次释放的技能偏离了目标。

[0069] 在上述两种示例性场景中，其他虚拟对象可以指其他用户通过终端所操控的虚拟对象，还可以指虚拟道具、虚拟NPC、虚拟士兵、虚拟建筑等，本公开实施例不对技能所能够命中的其他虚拟对象的类型进行具体限定。

[0070] 本申请实施例提供的方法，通过响应于对虚拟场景中任一技能的点击操作，获取该技能的技能类型，响应于该技能的技能类型为支持基于屏幕点击选择释放位置的目标技能，根据被控虚拟对象所在位置以及该技能的释放范围，在该虚拟场景中确定该技能的可释放区域，响应于对该可释放区域中任一释放位置的点击操作，控制该被控虚拟对象在该释放位置释放该技能，这种通过先点击技能再点击释放位置进行技能释放的人机交互方式，能够基于简单的两次点击操作释放技能，无需基于技能轮盘以及技能摇杆进行复杂的

拖拽操作，并且避免了由技能轮盘映射到释放范围时产生的偏移误差，因此具有更高的瞄准精度、更低的操作难度，从而带来更高效的人机交互效率，改善了用户体验。

[0071] 图3是本申请实施例提供的一种虚拟对象的控制方法的流程图。参见图3，该实施例应用于终端，该终端可以是图1所示的第一终端120或者第二终端160，该实施例包括下述步骤：

[0072] 300、终端显示虚拟场景。

[0073] 在上述过程中，终端响应于用户的开局操作，显示操作界面，在该操作界面中可以包括虚拟场景、虚拟对象的选择控件、虚拟道具的选择控件等。其中，开局操作用于触发一局游戏的启动，其可以为对开局控件的点击操作等，本申请实施例对开局操作的类型不做限定。

[0074] 该终端可以实时检测用户对该操作界面中各个虚拟对象的选择操作，将被选择的虚拟对象确定为本局出战的被控虚拟对象，可选地，该选择操作可以为点击操作、长按操作等，也可以为对任一虚拟对象所对应快捷键的触发操作，本申请实施例对此不作限定。由于在一些MOBA游戏中，用户可以一次性选择多个虚拟对象出战，并在对局中同时控制多个虚拟对象作战，因此在本申请实施例中的被控虚拟对象是指用户当前所操控的虚拟对象，俗称为“主操虚拟对象”。

[0075] 用户在操作界面中选择好被控虚拟对象之后，终端开启本局游戏，加载虚拟场景，并在虚拟场景中显示被控虚拟对象以及该被控虚拟对象所具有的至少一个技能的技能控件，该技能控件用于提供技能的用户交互接口，比如，该技能控件可以是以图形按钮的方式在虚拟场景中悬浮显示，该图形按钮可以是圆形按钮、矩形按钮或者不规则形状的按钮，本申请实施例不对技能控件的显示方式进行具体限定。可选地，在技能控件上可以展示对应技能的技能标识，该技能标识可以是一个标识图案，或者是一串标识文本，或者是一串标识码。

[0076] 在一些实施例中，终端还可以在虚拟场景中显示其他的UI (User Interface，用户界面) 控件，比如地图控件、商店控件、移动摇杆等，本公开实施例不对虚拟场景中显示的控件内容进行具体限定。需要说明的是，在本申请实施例中，对于除了技能控件之外的UI控件，称之为“非技能控件”。

[0077] 在上述显示技能控件的过程中，终端可以读取目标配置文件，该目标配置文件中存储有该被控虚拟对象的配置信息，基于该目标配置文件中的配置信息，可以确定出该被控虚拟对象所具有的至少一个技能，以及该至少一个技能各自的元信息，该元信息中可以包括各个技能的释放机制、冷却时长、持续时长、作用效果等，当然，在该元信息中还可以指示各个技能是否具有目标技能标识(比如IsFullScreenSkill标识)，用以检测被点击的技能是否为目标技能，将在后续步骤304中进行说明，这里不做赘述。

[0078] 301、终端在检测到对该虚拟场景中任一位置的点击操作时，响应于该位置上存在用户界面UI控件，获取该UI控件的控件类型。

[0079] 其中，该UI控件的控件类型包括技能控件和非技能控件，该技能控件为任一技能的UI控件，该非技能控件为除了技能控件之外的UI控件，该非技能控件包括但不限于：地图控件(俗称为“小地图”)、商店控件(用于购买虚拟装备或者虚拟道具的控件)、移动摇杆(用于控制被控虚拟对象进行位移的控件)等，需要说明的是，这里的移动摇杆和相关技术中的

技能摇杆不同,移动摇杆用于控制被控虚拟对象的位移,而技能摇杆则用于控制技能在释放范围内的释放方向和释放位置。

[0080] 在上述过程中,用户可以点击虚拟场景中的任一位置,终端在检测到用户对该虚拟场景中任一位置的点击操作时,检测该位置上是否存在UI控件,如果该位置上存在UI控件,获取该UI控件的控件类型。具体地,终端可以获取用户点击的该位置的屏幕坐标,检测该位置的屏幕坐标是否位于所有的UI控件的覆盖区域内,当该位置的屏幕坐标位于任一UI控件的覆盖区域内,确定该位置上存在UI控件,获取该UI控件的控件类型,当该UI控件的控件类型为技能控件时,执行下述步骤302,否则,当该位置的屏幕坐标不位于所有UI控件的覆盖区域内(确定该位置上不存在UI控件),或者当该UI控件的控件类型为非技能控件时,退出流程。

[0081] 这是由于在虚拟场景中,UI控件通常会悬浮显示在虚拟场景上层,因此用户点击任一位置之后,终端需要检测在该位置上是否存在UI控件,以及获取UI控件的控件类型,若UI控件的控件类型为非技能控件,则说明本次点击操作所触发的事件并非是技能释放事件,退出流程,否则,若UI控件的控件类型为技能控件,需要通过下述步骤302进行进一步地判断。

[0082] 302、终端响应于该UI控件的控件类型为技能控件,获取其他技能控件的触发状态,该其他技能控件为除了该技能之外的其他技能的技能控件。

[0083] 终端通过上述步骤301可以检测出该位置上是否存在技能控件,如果该位置上存在技能控件,继续检测其他技能控件的触发状态,如果存在处于触发状态的其他技能控件,说明已经用户正在释放其他技能,而由于用户不能同时释放两个技能,因此将本次点击操作确定为无效操作或者误操作,退出流程,否则,如果不存在处于触发状态的其他技能控件,执行下述步骤303。

[0084] 在一些实施例中,当任一技能的技能控件被点击时,终端可以为该技能分配一个屏蔽标识CurSkill,该屏蔽标识CurSkill用于屏蔽除了该技能之外的其他技能的点击事件,以保证当前操作技能的唯一性,因此,终端在检测是否存在处于触发状态的其他技能控件时,只需要检测当前是否存在任一技能具有屏蔽标识CurSkill,当存在任一技能具有屏蔽标识CurSkill时,确定存在处于触发状态的其他技能控件,退出流程,否则,当所有技能都不具有屏蔽标识CurSkill时,确定不存在处于触发状态的其他技能控件,执行下述步骤303。

[0085] 在上述过程中,终端响应于该位置上存在UI控件,并且该UI控件的控件类型为技能控件时,说明用户点击了该技能的技能控件,而由于用户无法同时操纵两个技能,因此需要通过上述步骤302来判断是否已经有其他技能被按下了,只有在没有任何其他技能被按下时,才执行下述步骤303,否则将退出流程。

[0086] 303、终端响应于该其他技能控件不处于触发状态,确定检测到对该虚拟场景中该技能的点击操作。

[0087] 在上述过程中,通过步骤301保证了用户点击的位置上存在UI控件,通过步骤302保证了用户已点击了某一技能,并且当前并未同时操作两个技能,因此可以确定到用户对该技能的点击操作,执行下述步骤304。

[0088] 图4是本申请实施例提供的一种界面示意图,请参考图4,在虚拟场景400中包括技

能1~4各自对应的技能控件401~404,用户在点击技能控件404之后,终端检测到触摸点上的UI控件的类型为技能控件,且不存在处于触发状态的其他技能控件,从而可以确定检测到对技能4的点击操作。

[0089] 304、终端响应于对虚拟场景中该技能的点击操作,获取该技能的技能类型。

[0090] 在上述步骤304中,用户可以通过点击该虚拟场景中任一技能的技能控件,使得终端响应于用户对该技能的点击操作,获取该技能的技能类型,其中,该技能类型可以包括目标技能和除了目标技能之外的技能,该目标技能为支持基于屏幕点击选择释放位置的技能,由于目标技能可以在终端的整个屏幕中通过点击操作来触发释放,这种释放方式区别于相关技术中基于技能轮盘的操作机制,因此目标技能也可以俗称为“全屏技能”、“触屏技能”。

[0091] 上述过程可以视为检测该技能的技能类型是否为目标技能的过程,终端可以在目标配置文件中为被控虚拟对象的所具备的至少一个技能中的目标技能的技能类型增加目标技能标识,从而通过在目标配置文件中检测该技能的技能类型是否具有目标技能标识,即可确定该技能的技能类型是否为目标技能,当检测到该技能的技能类型具有目标技能标识时,确定该技能的技能类型为目标技能,执行下一步骤305,否则,当检测到该技能的技能类型没有目标技能标识时,确定该技能的技能类型不是目标技能,退出流程。

[0092] 在一个示例性场景中,该目标技能标识可以为IsFullScreenSkill标识,在目标配置文件中为被控虚拟对象配置技能的时候,可以针对目标技能的技能类型增加一个IsFullScreenSkill标识,作为目标技能标识的相关参数,当检测到全屏触摸事件(也即用户对任一技能的点击操作)时,终端可以检测当前被点击的技能控件上所对应的技能是否在配置文件中被打上了IsFullScreenSkill标识,当该技能的技能类型拥有IsFullScreenSkill标识时,可以视为这个技能被全屏触摸事件触发。

[0093] 305、终端响应于该技能的技能类型为目标技能,在该虚拟场景中显示该技能的释放倒计时或者释放提示信息中至少一项,该目标技能为支持基于屏幕点击选择释放位置的技能。

[0094] 终端通过上述步骤304获取该技能的技能类型,当该技能的技能类型为目标技能时,终端可以在虚拟场景中显示该技能的释放倒计时或者释放提示信息,其中,该释放倒计时用于提示用户该技能剩余多长时间即将释放,该释放提示信息用于提示用户如何操作以释放该技能。

[0095] 可选地,该释放倒计时的时长或者该释放提示信息的内容可以由目标配置文件进行指示,该释放倒计时或者该释放提示信息可以以文本形式显示在虚拟场景中,或者还可以以动画形式在虚拟场景中悬浮播放,或者还可以以语音播报的形式进行智能提示,本公开实施例不对释放倒计时或者释放提示信息的显示方式进行具体限定。

[0096] 图5是本申请实施例提供的一种界面示意图,请参考图5,在虚拟场景500中包括技能1~4各自对应的技能控件501~504,用户在点击技能控件504之后,终端检测到触摸点上的UI控件的类型为技能控件,且不存在处于触发状态的其他技能控件,从而可以确定检测到对技能4的点击操作。此后终端检测到技能4属于目标技能,从而在虚拟场景中显示技能4的释放倒计时505以及释放提示信息506,可以看出来当前释放倒计时剩余5秒钟,且释放提示信息506为文字消息“点击屏幕即可释放技能”。

[0097] 306、终端根据被控虚拟对象所在位置以及该技能的释放范围,在该虚拟场景中确定该技能的可释放区域。

[0098] 在一些实施例中,若该可释放区域为圆形区域,终端在该虚拟场景中,可以将该可释放区域的圆心确定为该被控虚拟对象所在位置为圆心,将该可释放区域的半径确定为该技能的释放范围,当然,若该可释放区域不局限于圆形区域,还可以是扇形区域、矩形区域、不规则区域等,比如,当可释放区域为扇形区域的时候,用户可以通过移动摇杆来控制扇形区域的朝向,并通过旋转操作来旋转扇形区域的朝向。

[0099] 在一个示例性场景中,假设可释放区域为圆形区域,当被控虚拟对象位于屏幕左下角时,用户点击了技能1,技能1的释放范围为六尺,因此在虚拟场景中,将屏幕左下角中被控虚拟对象所在位置周围六尺的圆形区域确定为技能1的可释放区域。

[0100] 在一些实施例中,终端在该虚拟场景中为该可释放区域增加释放模板,该释放模板用于表示在该释放模板之外的区域释放该技能无效。此后如果再次检测到用户对屏幕中任一位置点的点击操作,终端通过检测该位置点是否位于该释放模板中,即可确定该位置点是否位于可释放区域内,具体地,当该位置点位于该释放模板中,确定该位置点位于可释放区域内,执行下述步骤307,否则,当该位置点不位于该释放模板中,确定该位置点位于可释放区域外,则不会释放技能,将本次点击操作视为无效操作或者误操作。

[0101] 307、终端响应于对该虚拟场景中地图控件的点击操作,在该地图控件中确定被点击的目标区域。

[0102] 在上述过程中,由于虚拟场景通常较大,而终端屏幕的大小有限,在终端屏幕中只能展示出一部分的虚拟场景,因此通常在虚拟场景中可以显示地图控件,该地图控件用于随时切换将屏幕中当前显示的部分虚拟场景切换至其他的部分虚拟场景,这样就无需用户控制虚拟对象进行位移才能切换视角,而是可以通过直接点击地图控件中的目标区域来进行视角切换,极大地方便了用户操作。

[0103] 在一些实施例中,在地图控件中的每个地图点都对应于虚拟场景中的一个位置点,使得整个虚拟场景按照一定的比例缩小显示于地图控件中,终端在检测到用户对地图控件的点击操作时,可以在地图控件中确定用户点击的地图点,并将该地图点作为中心,确定一个矩形区域,该矩形区域的长宽比与终端屏幕的长宽比相等,该矩形区域的尺寸可以由用户进行预设,从而将该矩形区域确定为被点击的目标区域。

[0104] 308、终端将该地图控件中的该目标区域映射为该虚拟场景中的目标场景,该目标场景中包括该可释放区域。

[0105] 在上述过程中,终端根据地图控件与虚拟场景的映射关系,将该目标区域映射为目标场景,并在终端屏幕中显示该目标场景,由于该目标场景中包括部分或全部的可释放区域,此时用户即可通过再次的点击操作完成技能释放。

[0106] 图6是本申请实施例提供的一种界面示意图,请参考图6,在虚拟场景中包括技能1~4各自对应的技能控件601~604,用户在点击技能控件604之后,终端检测到触摸点上的UI控件的类型为技能控件,且不存在处于触发状态的其他技能控件,从而可以确定检测到对技能4的点击操作。此后终端检测到技能4属于目标技能,从而在虚拟场景中显示技能4的释放倒计时605以及释放提示信息606,可以看出来当前释放倒计时剩余3秒钟,且释放提示信息606为文字消息“点击屏幕即可释放技能”。进一步地,在虚拟场景中还显示了地图控件

607, 用户在点击了地图控件607之后, 终端在该地图控件607中确定被点击的目标区域608, 将目标区域608所映射的目标场景600显示在终端屏幕中。

[0107] 309、终端在该目标场景中检测对该可释放区域中任一释放位置的点击操作。

[0108] 在上述过程中, 终端在该目标场景中检测用户对可释放区域内任一释放位置的点击操作, 由于终端在上述步骤306中可以为可释放区域增加释放模板, 而该释放模板用于表示在该释放模板之外的区域释放该技能无效。此后如果再次检测到用户对目标场景中任一释放位置的点击操作, 终端通过检测该释放位置是否位于该释放模板中, 即可确定该释放位置是否位于可释放区域内, 具体地, 当该释放位置位于该释放模板中, 确定该释放位置位于可释放区域内, 执行下述步骤310, 否则, 当该释放位置不位于该释放模板中, 确定该释放位置位于可释放区域外, 则不会释放技能, 将本次点击操作视为无效操作或者误操作。

[0109] 310、终端响应于对该可释放区域中任一释放位置的点击操作, 控制该被控虚拟对象在该释放位置释放该技能。

[0110] 在上述过程中, 当终端检测到用户对该可释放区域中任一释放位置的点击操作之后, 终端响应于用户对该可释放区域中任一释放位置的点击操作, 可以获取在终端屏幕中该释放位置的屏幕坐标, 将该屏幕坐标映射为在该虚拟场景中该释放位置的场景坐标, 从而控制该被控虚拟对象在该场景坐标释放该技能。

[0111] 在一些实施例中, 终端在将屏幕坐标映射为场景坐标时, 可以采用下述公式的逆矩阵, 从而能够将二维屏幕内的屏幕坐标转化为三维场景内的场景坐标FocusPoint:

$$[0112] \quad Z_c \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \begin{bmatrix} \frac{1}{dx} & 0 & u_0 \\ 0 & \frac{1}{dy} & v_0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} f & 0 & 0 & 0 \\ 0 & f & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} R & T \\ \vec{0} & 1 \end{bmatrix} \begin{bmatrix} X_w \\ Y_w \\ Z_w \\ 1 \end{bmatrix} = \begin{bmatrix} f_x & 0 & u_0 & 0 \\ 0 & f_y & v_0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} R & T \\ \vec{0} & 1 \end{bmatrix} \begin{bmatrix} X_w \\ Y_w \\ Z_w \\ 1 \end{bmatrix}$$

[0113] 其中, $\begin{bmatrix} f_x & 0 & u_0 & 0 \\ 0 & f_y & v_0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}$ 为相机内参矩阵, f_x 表示相机在x轴方向上的焦距, f_y 表示相机在y轴方向上的焦距, u_0 和 v_0 均为相机的光圈中心。

[0114] 其中, $\begin{bmatrix} R & T \\ \vec{0} & 1 \end{bmatrix}$ 为相机外参矩阵, R 表示旋转矩阵, T 表示偏移向量。

[0115] 其中, X_w 、 Y_w 、 Z_w 分别表示在三维场景内释放位置在x轴、y轴、z轴上的场景坐标, u 、 v 分别表示在二维屏幕内释放位置在u轴、v轴上的屏幕坐标, Z_c 表示释放位置在相机坐标系内的z轴坐标值, f 为相机焦距。

[0116] 进一步地, 在确定了技能的释放位置的场景坐标之后, 可以调用技能指示器, 从而在虚拟场景中显示出该技能的预计释放位置, 用户可以通过技能指示器观察预计释放位置是否为想要释放的位置, 如果两者一致, 用户松手之后即可触发被控虚拟对象在被点击的释放位置上释放该技能。

[0117] 在一些实施例中, 如果两者不一致(也即技能指示器所显示的预计释放位置并不理想), 此时用户可以不松手, 而是在屏幕中执行拖动操作, 使得终端响应于用户对该可释

放区域中任一释放位置的拖动操作,基于拖动操作的拖动轨迹,在该虚拟场景中更新该释放位置。最终,当用户拖动完毕松手之后,触发被控虚拟对象在拖动轨迹中最新更新的释放位置上释放该技能。

[0118] 需要说明的是,在释放该技能之后,终端还可以结算该技能的作用效果。根据技能类型的不同,作用效果的结算规则也不尽相同。

[0119] 在一个示例性场景中,对于常规的攻击技能来说,终端可以检测在该释放位置上是否存在其他虚拟对象,若该释放位置上存在其他虚拟对象,确定该攻击技能命中其他虚拟对象,并对其他虚拟对象的虚拟生命值进行一定数值的扣除,否则,若该释放位置上不存在其他虚拟对象,确定该攻击技能未命中其他虚拟对象,意味着本次释放的技能偏离了目标。

[0120] 在一个示例性场景中,对于一些AOE (Area Of Effect,范围攻击) 技能来说,用户所点击的释放位置其实是技能攻击范围的中心点,终端可以检测在该释放位置周围的目标距离内是否存在其他虚拟对象,若该释放位置周围的目标距离内存在其他虚拟对象,确定该攻击技能命中其他虚拟对象,并对其他虚拟对象的虚拟生命值进行一定数值的扣除,否则,若该释放位置周围的目标距离内不存在其他虚拟对象,确定该攻击技能未命中其他虚拟对象,意味着本次释放的技能偏离了目标。

[0121] 在上述两种示例性场景中,其他虚拟对象可以指其他用户通过终端所操控的虚拟对象,还可以指虚拟道具、虚拟NPC、虚拟士兵、虚拟建筑等,本公开实施例不对技能所能够命中的其他虚拟对象的类型进行具体限定。

[0122] 图7是本申请实施例提供的一种界面示意图,请参考图7,在虚拟场景700中包括技能1~4各自对应的技能控件701~704,用户在点击技能控件704之后,终端检测到触摸点上的UI控件的类型为技能控件,且不存在处于触发状态的其他技能控件,从而可以确定检测到对技能4的点击操作。此后终端检测到技能4属于目标技能,从而在虚拟场景中显示技能4的释放倒计时705以及释放提示信息706,可以看出来当前释放倒计时剩余4秒钟,且释放提示信息706为文字消息“点击屏幕即可释放技能”。用户通过点击虚拟场景中可释放区域内的释放位置707,触发被控虚拟对象在释放位置707上直接释放技能4。

[0123] 上述所有可选技术方案,可以采用任意结合形成本公开的可选实施例,在此不再一一赘述。

[0124] 本申请实施例提供的方法,通过响应于对虚拟场景中任一技能的点击操作,获取该技能的技能类型,响应于该技能的技能类型为支持基于屏幕点击选择释放位置的目标技能,根据被控虚拟对象所在位置以及该技能的释放范围,在该虚拟场景中确定该技能的可释放区域,响应于对该可释放区域中任一释放位置的点击操作,控制该被控虚拟对象在该释放位置释放该技能,这种通过先点击技能再点击释放位置进行技能释放的人机交互方式,能够基于简单的两次点击操作释放技能,无需基于技能轮盘以及技能摇杆进行复杂的拖拽操作,并且避免了由技能轮盘映射到释放范围时产生的偏移误差,因此具有更高的瞄准精度、更低的操作难度,从而带来更高效的人机交互效率,改善了用户体验。

[0125] 图8是本申请实施例提供的一种触屏技能释放原理性示意图,请参考图8,在一个示例性场景中,如800所示,终端可以通过如下流程来释放触屏技能(即目标技能):当玩家手指触摸屏幕时,处于屏幕按下状态,终端首先会判断当前触摸点是否在其他UI控件上,也

即检测触摸点是否有其他虚拟按钮,比如小地图(地图控件)、商店(商店控件)、移动摇杆等非技能控件的虚拟按钮,不在这些非技能控件上的触摸事件才会被认为是本次释放技能所需要的触摸事件,接着检测当前触摸点上技能按钮是有按下,也即检测当前触摸点是否位于任一技能的技能控件上,如果当前触摸点位于任一技能的技能控件上,又因为玩家不能同时操作两个技能,所以需要去判断是否已经有其他技能控件已经按下,也即查找是否有已触发的其他触屏释放技能,在没有任何其他技能控件按下时,才能接受本次触摸技能操作。

[0126] 并且,在配置技能的时候,针对触屏技能配置相关参数,比如给技能增加IsFullScreenSkill标识,当检测到全屏触摸事件时,终端检测当前技能控件所对应的技能是否有IsFullScreenSkill标识,当该技能拥有IsFullScreenSkill标识时,认为该技能被全屏触摸事件触发,此时可以设置当前全屏触摸事件的触摸标识FingerID和屏蔽标识CurSkill,其中,触摸标识FingerID用来区别当前全屏触摸事件与按下的其他触摸事件,而屏蔽标识CurSkill则用来屏蔽其他控件按下的事件,保证当前操作技能的唯一性。当以上所有步骤都顺利通过时,终端根据技能控件上的触摸点(释放位置的屏幕坐标)计算出在虚拟场景中的场景坐标FocusPoint,并根据场景坐标FocusPoint处理技能指示器的表现,以向玩家展示出预计释放位置的预览效果。

[0127] 进一步地,当触摸点在屏幕上移动的时候,处于屏幕拖动状态,首先检测当前按下事件的触摸标识FingerID与之前触发的全屏触摸事件的FingerID是否一致,也即检测触摸点与之前的触摸标识FingerID是否相同以及是否存在屏蔽标识CurSkill,当触摸标识FingerID相同且存在屏蔽标识CurSkill的时候,接着判断是否存在当前按下事件所触发的技能,当这两个条件都满足的时候,终端用当前触摸点对场景坐标FocusPoint进行更新,也即是按照新的触摸点的屏幕坐标进行映射,得到新的场景坐标FocusPoint。

[0128] 进一步地,当触摸点抬起(玩家松手)时,处于屏幕松手状态,终端同样检测当前按下事件的FingerID与之前触发的全屏触摸事件的FingerID是否一致,也即检测触摸点与之前的触摸标识FingerID是否相同以及是否存在屏蔽标识CurSkill,当触摸标识FingerID相同且存在屏蔽标识CurSkill的时候,如果存在当前按下事件所触发的技能,当这两个条件都满足的时候,终端将相关技能数据(包括屏蔽标识CurSkill、最终的场景坐标FocusPoint等)发出,向服务器请求释放技能,并基于帧同步技术同步到参与游戏的其他终端。

[0129] 具体地,对于允许点击屏幕施法的触屏技能,玩家点击技能控件后,进入屏幕施法模式,在屏幕中下方区域会显示屏幕施法模式的关闭倒计时(也即技能的释放倒计时)和释放提示信息“点击屏幕即可施放技能”。此外,在技能的释放范围内会增加一层释放模板,释放模板表示该模板所覆盖的区域内可点击施法。如果玩家点击到区域外的地方,则不会进行施法。在倒计时过程中,玩家点击屏幕,则会在虚拟场景的可释放区域内被点击的释放位置上直接释放技能。如果需要向非屏幕内的位置进行释放,玩家可以先点击小地图使屏幕移动到对应位置,再点击屏幕进行释放技能。

[0130] 在本申请实施例中,对于技能的释放范围较远的技能,增加了点击屏幕施法(也即释放)的机制,玩家可以通过点击屏幕,直接对点击的释放位置进行施法,达到“所见即所得”的交互效果,提升了施法过程中的瞄准精度,使玩家的游玩过程更加舒适,优化了用户体验。当然,对于允许点击屏幕施法的触屏技能,终端也可以提供两种兼容的施法方式,使

得用户既可以通过点击屏幕施法，也可以基于传统的技能轮盘施法，提供更加丰富多样化的
人机交互方式。

[0131] 图9是本申请实施例提供的一种虚拟对象的控制装置的结构示意图，请参考图9，
该装置包括：

[0132] 检测模块901，用于响应于对虚拟场景中任一技能的点击操作，检测该技能是否为
目标技能，该目标技能为支持基于屏幕点击选择释放位置的技能；

[0133] 确定模块902，用于响应于该技能为目标技能，根据被控虚拟对象所在位置以及该
技能的释放范围，在该虚拟场景中确定该技能的可释放区域；

[0134] 控制模块903，用于响应于对该可释放区域中任一释放位置的点击操作，控制该被
控虚拟对象在该释放位置释放该技能。

[0135] 本申请实施例提供的装置，通过响应于对虚拟场景中任一技能的点击操作，获取
该技能的技能类型，响应于该技能的技能类型为支持基于屏幕点击选择释放位置的目标技
能，根据被控虚拟对象所在位置以及该技能的释放范围，在该虚拟场景中确定该技能的可
释放区域，响应于对该可释放区域中任一释放位置的点击操作，控制该被控虚拟对象在该
释放位置释放该技能，这种通过先点击技能再点击释放位置进行技能释放的人机交互方
式，能够基于简单的两次点击操作释放技能，无需基于技能轮盘以及技能摇杆进行复杂的
拖拽操作，并且避免了由技能轮盘映射到释放范围时产生的偏移误差，因此具有更高的瞄
准精度、更低的操作难度，从而带来更高效的人机交互效率，改善了用户体验。

[0136] 在一种可能实施方式中，该控制模块903用于：

[0137] 响应于对该可释放区域中任一释放位置的点击操作，获取在该屏幕中该释放位置
的屏幕坐标；

[0138] 将该屏幕坐标映射为在该虚拟场景中该释放位置的场景坐标；

[0139] 控制该被控虚拟对象在该场景坐标释放该技能。

[0140] 在一种可能实施方式中，基于图9的装置组成，该装置还包括：

[0141] 映射检测模块，用于响应于对该虚拟场景中地图控件的点击操作，在该地图控件
中确定被点击的目标区域；将该地图控件中的该目标区域映射为该虚拟场景中的目标场
景，该目标场景中包括该可释放区域；在该目标场景中检测对该可释放区域中任一释放位
置的点击操作。

[0142] 在一种可能实施方式中，若该可释放区域为圆形区域，该确定模块902用于：

[0143] 在该虚拟场景中，将该可释放区域的圆心确定为该被控虚拟对象所在位置为圆
心，将该可释放区域的半径确定为该技能的释放范围；

[0144] 在该虚拟场景中为该可释放区域增加释放模板，该释放模板用于表示在该释放模
板之外的区域释放该技能无效。

[0145] 在一种可能实施方式中，基于图9的装置组成，该装置还包括：

[0146] 显示模块，用于响应于该技能为目标技能，在该虚拟场景中显示该技能的释放倒
计时或者释放提示信息中至少一项。

[0147] 在一种可能实施方式中，该检测模块901还用于：

[0148] 在检测到对该虚拟场景中任一位置的点击操作时，检测该位置上是否存在非技能
控件，该非技能控件为除了该技能的技能控件之外的用户界面控件；

[0149] 响应于该位置上不存在该非技能控件且存在该技能的技能控件,检测是否存在处于触发状态的其他技能控件,该其他技能控件为除了该技能之外的其他技能的技能控件;
[0150] 响应于不存在处于触发状态的其他技能控件,确定检测到对该虚拟场景中该技能的点击操作。

[0151] 在一种可能实施方式中,基于图9的装置组成,该装置还包括:

[0152] 更新模块,用于响应于对该可释放区域中任一释放位置的拖动操作,基于拖动操作的拖动轨迹,在该虚拟场景中更新该释放位置。

[0153] 需要说明的是:上述实施例提供的虚拟对象的控制装置在控制虚拟对象时,仅以上述各功能模块的划分进行举例说明,实际应用中,可以根据需要而将上述功能分配由不同的功能模块完成,即将终端设备的内部结构划分成不同的功能模块,以完成以上描述的全部或者部分功能。另外,上述实施例提供的虚拟对象的控制装置与虚拟对象的控制方法实施例属于同一构思,其具体实现过程详见虚拟对象的控制方法实施例,这里不再赘述。

[0154] 图10是本申请实施例提供的一种终端的结构示意图。该终端1000可以是:智能手机、平板电脑、MP3播放器(Moving Picture Experts Group Audio Layer III,动态影像专家压缩标准音频层面3)、MP4(Moving Picture Experts Group Audio Layer IV,动态影像专家压缩标准音频层面4)播放器、笔记本电脑或台式电脑。终端1000还可能被称为用户设备、便携式终端、膝上型终端、台式终端等其他名称。

[0155] 通常,终端1000包括有:处理器1001和存储器1002。

[0156] 处理器1001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1001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1001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器,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1001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1001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。

[0157] 存储器1002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1002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1002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,该至少一个指令用于被处理器1001所执行以实现本申请中各个实施例提供的虚拟对象的控制方法。

[0158] 在一些实施例中,终端1000还可选包括有:外围设备接口1003和至少一个外围设备。处理器1001、存储器1002和外围设备接口1003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1003相连。具体地,外围设备包括:射频电路1004、触摸显示屏1005、摄像头组件1006、音频电路1007和电源1009中的至少一种。

[0159] 外围设备接口1003可被用于将I/O(Input/Output,输入/输出)相关的至少一个外

围设备连接到处理器1001和存储器1002。在一些实施例中，处理器1001、存储器1002和外围设备接口1003被集成在同一芯片或电路板上；在一些其他实施例中，处理器1001、存储器1002和外围设备接口1003中的任意一个或两个可以在单独的芯片或电路板上实现，本实施例对此不加以限定。

[0160] 射频电路1004用于接收和发射RF (Radio Frequency, 射频) 信号，也称电磁信号。射频电路1004通过电磁信号与通信网络以及其他通信设备进行通信。射频电路1004将电信号转换为电磁信号进行发送，或者，将接收到的电磁信号转换为电信号。可选地，射频电路1004包括：天线系统、RF收发器、一个或多个放大器、调谐器、振荡器、数字信号处理器、编解码芯片组、用户身份模块卡等等。射频电路1004可以通过至少一种无线通信协议来与其它终端进行通信。该无线通信协议包括但不限于：城域网、各代移动通信网络(2G、3G、4G及5G)、无线局域网和/或WiFi (Wireless Fidelity, 无线保真) 网络。在一些实施例中，射频电路1004还可以包括NFC (Near Field Communication, 近距离无线通信) 有关的电路，本申请对此不加以限定。

[0161] 显示屏1005用于显示UI (User Interface, 用户界面)。该UI可以包括图形、文本、图标、视频及其它们的任意组合。当显示屏1005是触摸显示屏时，显示屏1005还具有采集在显示屏1005的表面或表面上方的触摸信号的能力。该触摸信号可以作为控制信号输入至处理器1001进行处理。此时，显示屏1005还可以用于提供虚拟按钮和/或虚拟键盘，也称软按钮和/或软键盘。在一些实施例中，显示屏1005可以为一个，设置终端1000的前面板；在另一些实施例中，显示屏1005可以为至少两个，分别设置在终端1000的不同表面或呈折叠设计；在再一些实施例中，显示屏1005可以是柔性显示屏，设置在终端1000的弯曲表面上或折叠面上。甚至，显示屏1005还可以设置成非矩形的不规则图形，也即异形屏。显示屏1005可以采用LCD (Liquid Crystal Display, 液晶显示屏)、OLED (Organic Light-Emitting Diode, 有机发光二极管) 等材质制备。

[0162] 摄像头组件1006用于采集图像或视频。可选地，摄像头组件1006包括前置摄像头和后置摄像头。通常，前置摄像头设置在终端的前面板，后置摄像头设置在终端的背面。在一些实施例中，后置摄像头为至少两个，分别为主摄像头、景深摄像头、广角摄像头、长焦摄像头中的任意一种，以实现主摄像头和景深摄像头融合实现背景虚化功能、主摄像头和广角摄像头融合实现全景拍摄以及VR (Virtual Reality, 虚拟现实) 拍摄功能或者其它融合拍摄功能。在一些实施例中，摄像头组件1006还可以包括闪光灯。闪光灯可以是单色温闪光灯，也可以是双色温闪光灯。双色温闪光灯是指暖光闪光灯和冷光闪光灯的组合，可以用于不同色温下的光线补偿。

[0163] 音频电路1007可以包括麦克风和扬声器。麦克风用于采集用户及环境的声波，并将声波转换为电信号输入至处理器1001进行处理，或者输入至射频电路1004以实现语音通信。出于立体声采集或降噪的目的，麦克风可以为多个，分别设置在终端1000的不同部位。麦克风还可以是阵列麦克风或全向采集型麦克风。扬声器则用于将来自处理器1001或射频电路1004的电信号转换为声波。扬声器可以是传统的薄膜扬声器，也可以是压电陶瓷扬声器。当扬声器是压电陶瓷扬声器时，不仅可以将电信号转换为人类可听见的声波，也可以将电信号转换为人类听不见的声波以进行测距等用途。在一些实施例中，音频电路1007还可以包括耳机插孔。

[0164] 电源1009用于为终端1000中的各个组件进行供电。电源1009可以是交流电、直流电、一次性电池或可充电电池。当电源1009包括可充电电池时，该可充电电池可以支持有线充电或无线充电。该可充电电池还可以用于支持快充技术。

[0165] 在一些实施例中，终端1000还包括有一个或多个传感器1010。该一个或多个传感器1010包括但不限于：加速度传感器1011、陀螺仪传感器1012、压力传感器1013、光学传感器1015以及接近传感器1016。

[0166] 加速度传感器1011可以检测以终端1000建立的坐标系的三个坐标轴上的加速度大小。比如，加速度传感器1011可以用于检测重力加速度在三个坐标轴上的分量。处理器1001可以根据加速度传感器1011采集的重力加速度信号，控制触摸显示屏1005以横向视图或纵向视图进行用户界面的显示。加速度传感器1011还可以用于游戏或者用户的运动数据的采集。

[0167] 陀螺仪传感器1012可以检测终端1000的机体方向及转动角度，陀螺仪传感器1012可以与加速度传感器1011协同采集用户对终端1000的3D动作。处理器1001根据陀螺仪传感器1012采集的数据，可以实现如下功能：动作感应(比如根据用户的倾斜操作来改变UI)、拍摄时的图像稳定、游戏控制以及惯性导航。

[0168] 压力传感器1013可以设置在终端1000的侧边框和/或触摸显示屏1005的下层。当压力传感器1013设置在终端1000的侧边框时，可以检测用户对终端1000的握持信号，由处理器1001根据压力传感器1013采集的握持信号进行左右手识别或快捷操作。当压力传感器1013设置在触摸显示屏1005的下层时，由处理器1001根据用户对触摸显示屏1005的压力操作，实现对UI界面上的可操作性控件进行控制。可操作性控件包括按钮控件、滚动条控件、图标控件、菜单控件中的至少一种。

[0169] 光学传感器1015用于采集环境光强度。在一个实施例中，处理器1001可以根据光学传感器1015采集的环境光强度，控制触摸显示屏1005的显示亮度。具体地，当环境光强度较高时，调高触摸显示屏1005的显示亮度；当环境光强度较低时，调低触摸显示屏1005的显示亮度。在另一个实施例中，处理器1001还可以根据光学传感器1015采集的环境光强度，动态调整摄像头组件1006的拍摄参数。

[0170] 接近传感器1016，也称距离传感器，通常设置在终端1000的前面板。接近传感器1016用于采集用户与终端1000的正面之间的距离。在一个实施例中，当接近传感器1016检测到用户与终端1000的正面之间的距离逐渐变小时，由处理器1001控制触摸显示屏1005从亮屏状态切换为息屏状态；当接近传感器1016检测到用户与终端1000的正面之间的距离逐渐变大时，由处理器1001控制触摸显示屏1005从息屏状态切换为亮屏状态。

[0171] 本领域技术人员可以理解，图10中示出的结构并不构成对终端1000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。

[0172] 在示例性实施例中，还提供了一种计算机可读存储介质，例如包括至少一条程序代码的存储器，上述至少一条程序代码可由终端中的处理器执行以完成上述实施例中虚拟对象的控制方法。例如，该计算机可读存储介质可以是ROM(Read-Only Memory，只读存储器)、RAM(Random-Access Memory，随机存取存储器)、CD-ROM(Compact Disc Read-Only Memory，只读光盘)、磁带、软盘和光数据存储设备等。

[0173] 本领域普通技术人员可以理解实现上述实施例的全部或部分步骤可以通过硬件

来完成,也可以通过程序来指令相关的硬件完成,该程序可以存储于一种计算机可读存储介质中,上述提到的存储介质可以是只读存储器,磁盘或光盘等。

[0174] 以上所述仅为本申请的可选实施例,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。

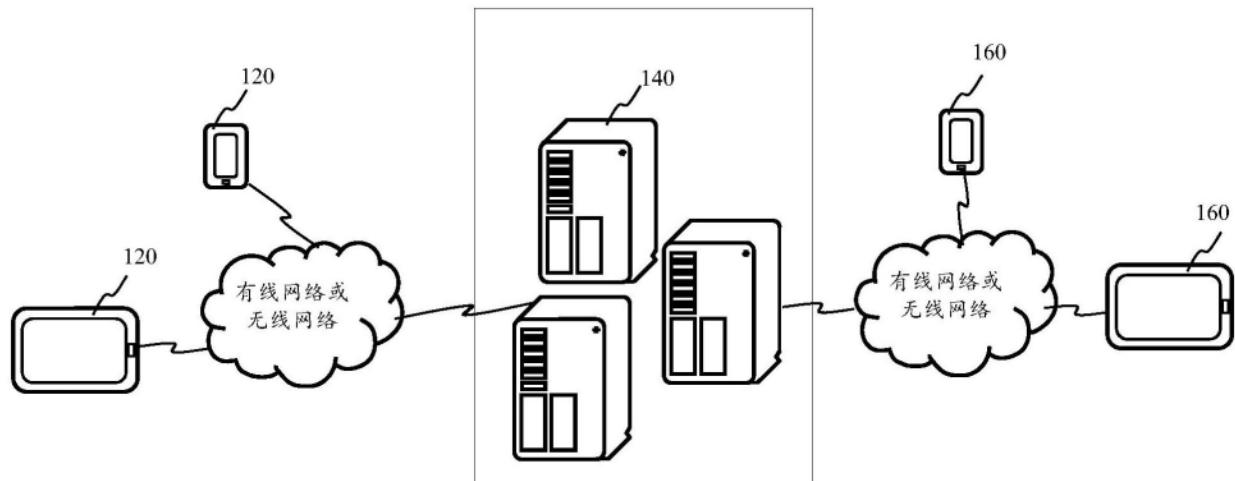


图1

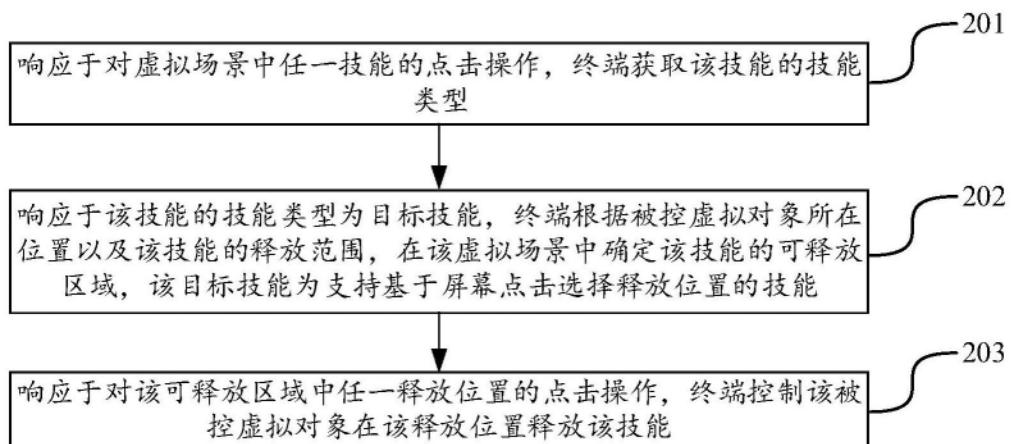


图2

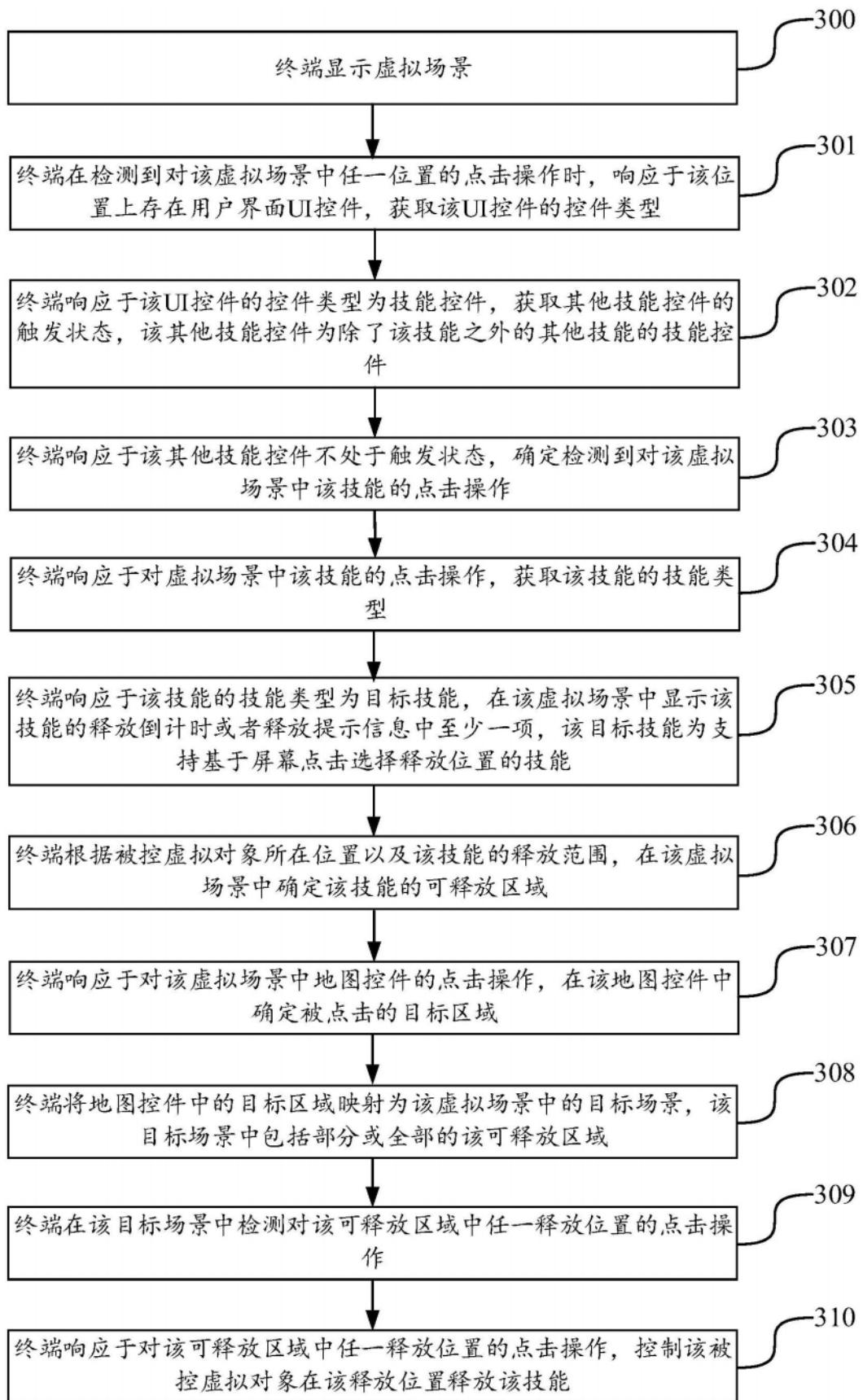


图3

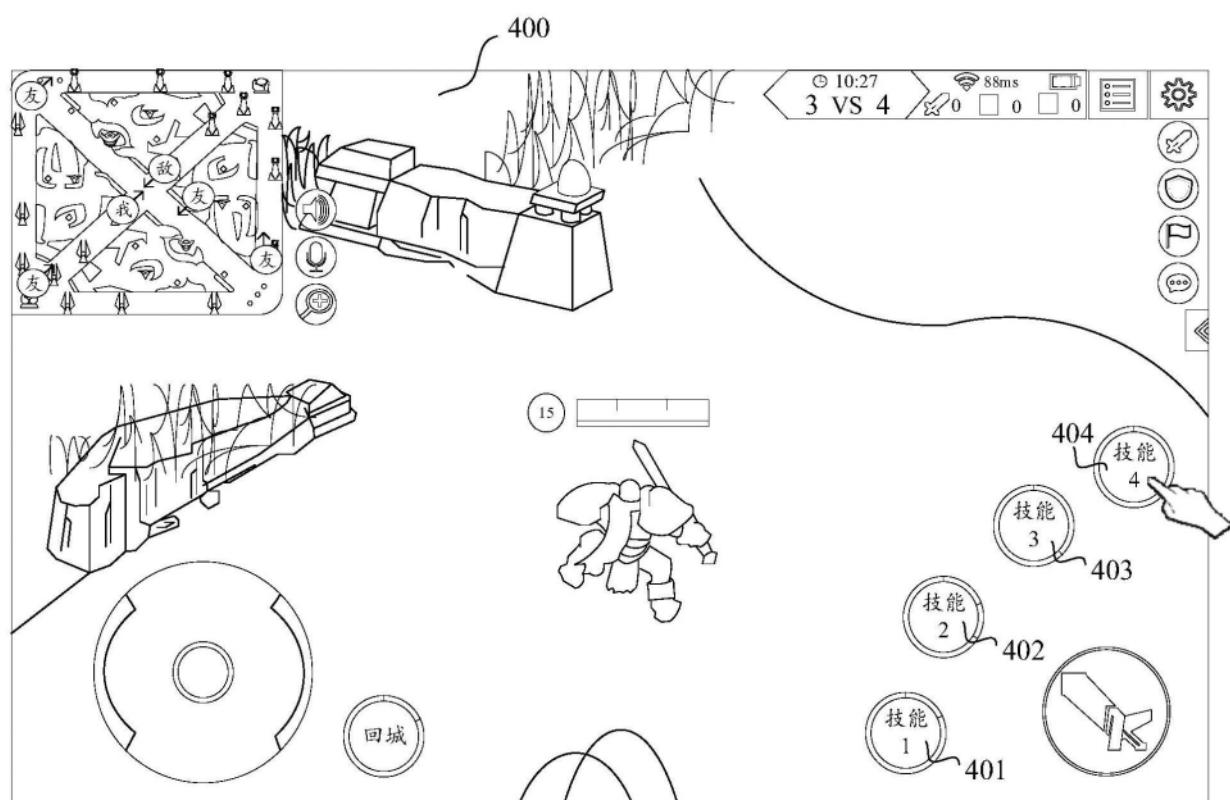


图4

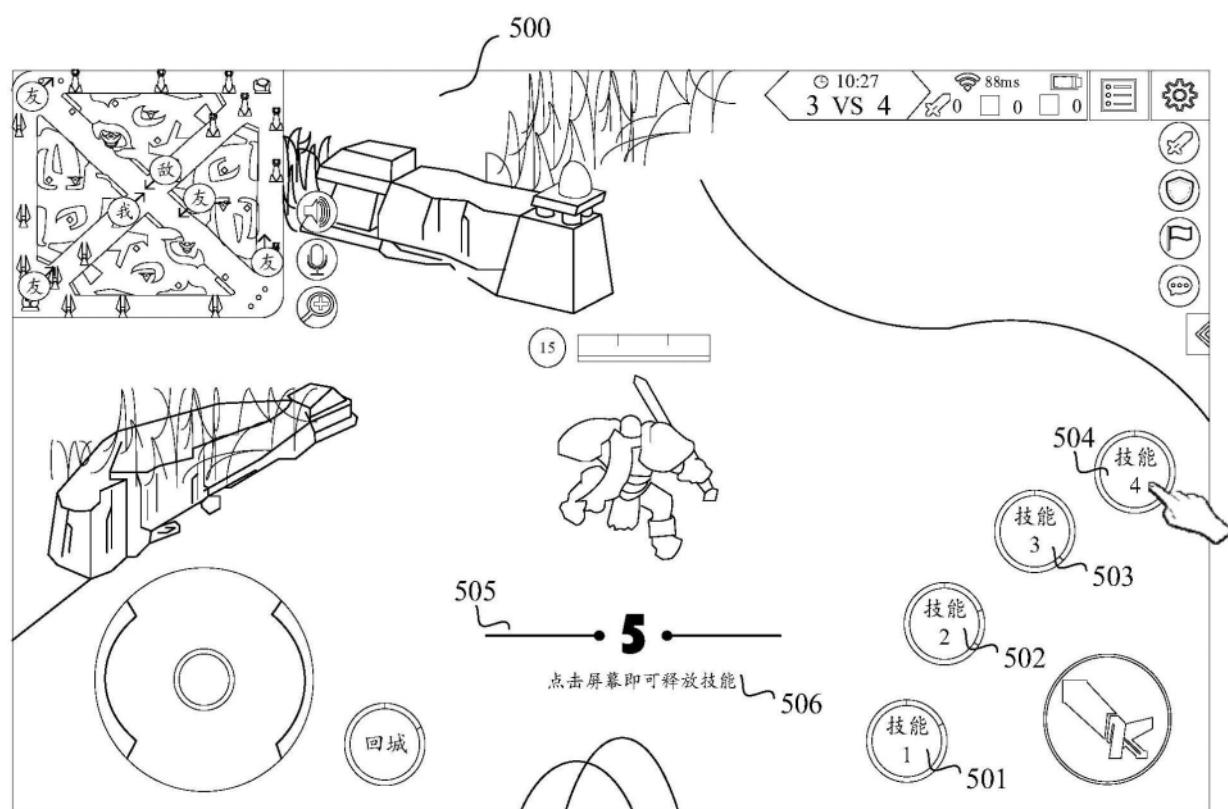


图5

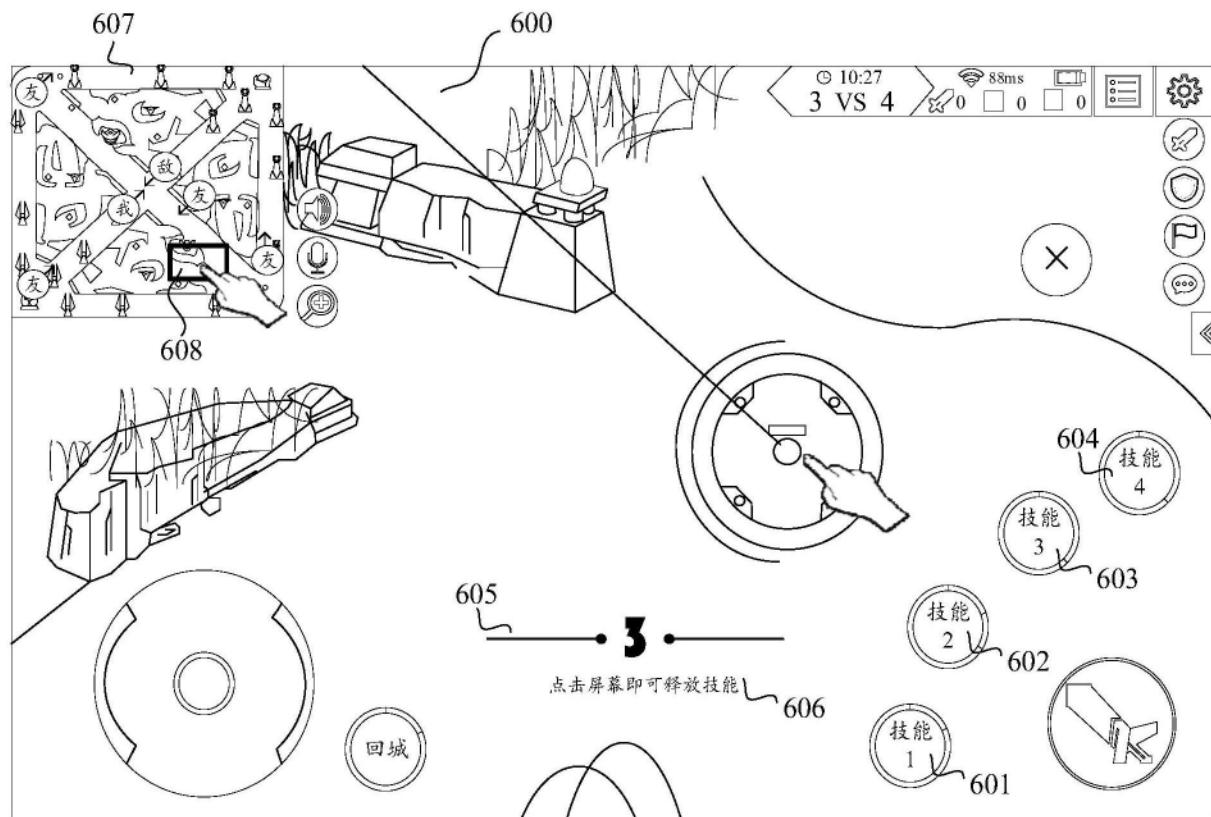


图6

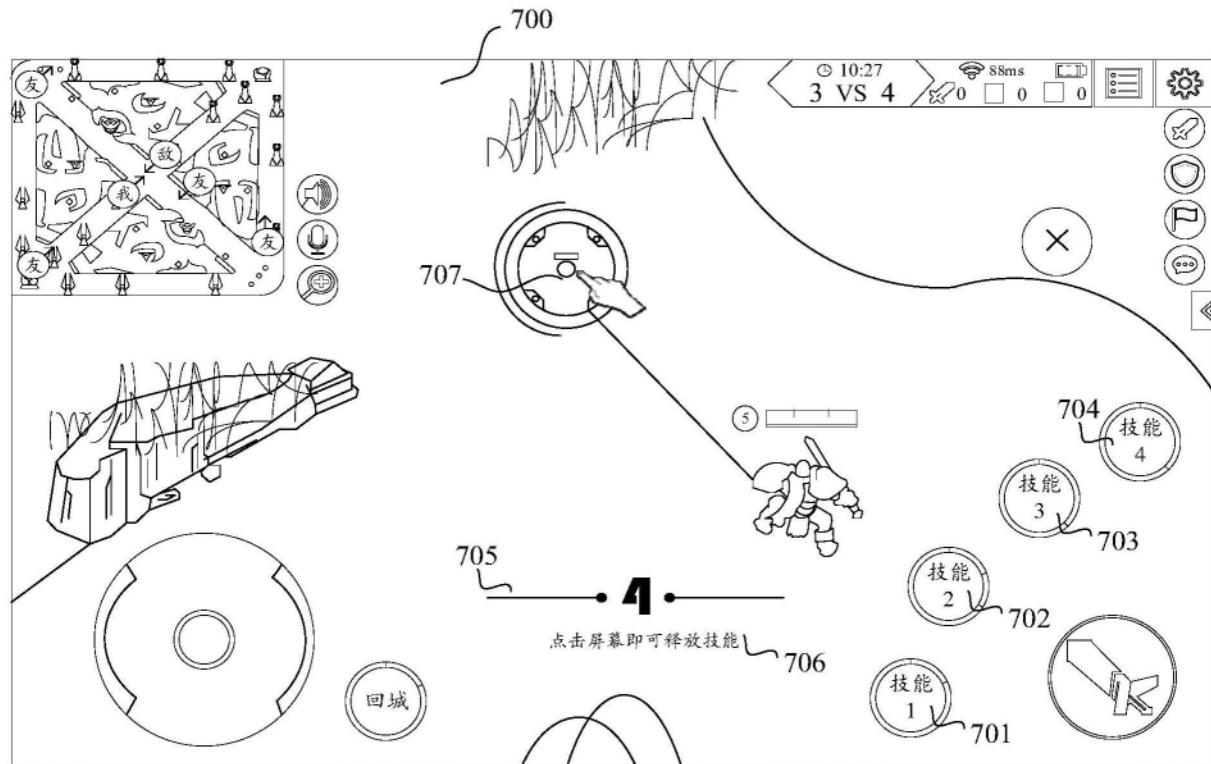


图7

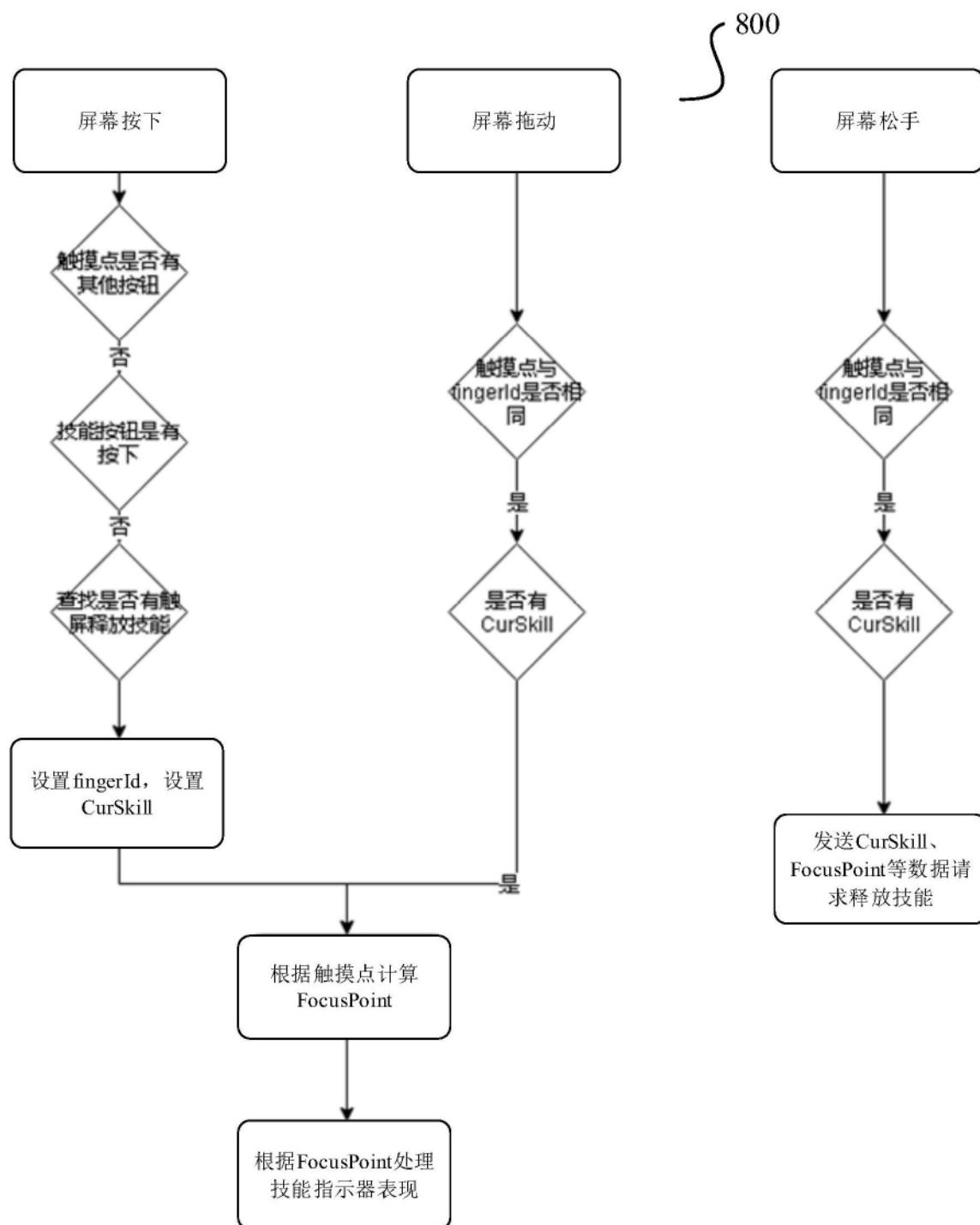


图8

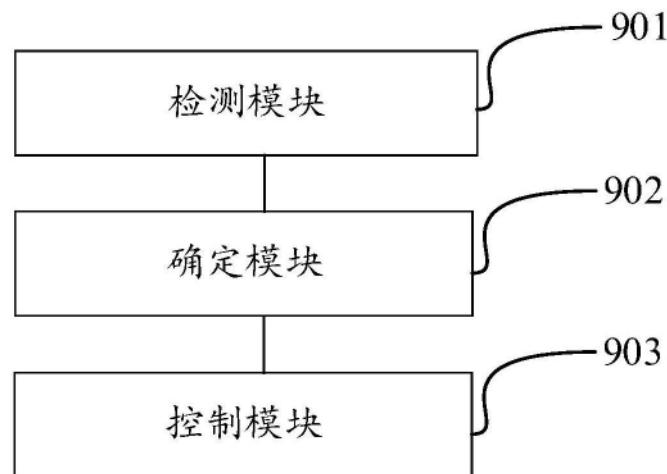


图9

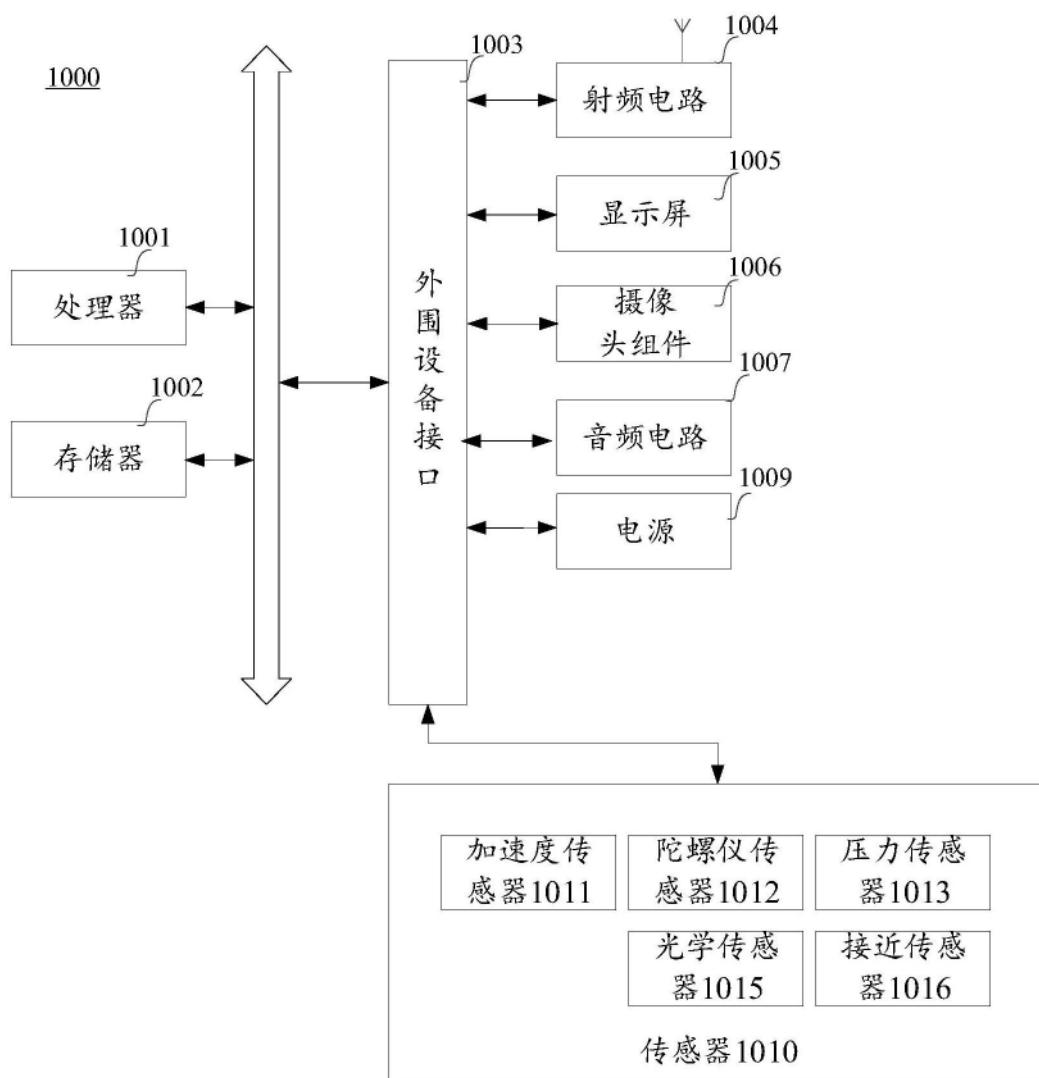


图10