

基于可飞行道具的互动方法、装置、电子设备及存储介质

本申请要求于 2022 年 12 月 12 日提交、申请号为 202211610140.1、发明名称为“基于可飞行道具的互动方法、装置、电子设备及存储介质”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，特别涉及一种基于可飞行道具的互动方法、装置、电子设备及存储介质。

背景技术

随着计算机技术的发展和终端功能的多样化，MOBA（Multiplayer Online Battle Arena，多人在线对抗竞技游戏）游戏逐渐变得盛行。在 MOBA 游戏提供的虚拟场景中，隶属于相同或不同队伍的虚拟对象之间能够通过互相释放虚拟技能等方式来进行协作或对抗。

发明内容

本申请实施例提供了一种基于可飞行道具的互动方法、装置、电子设备及存储介质。该技术方案如下：

一方面，提供了一种基于可飞行道具的互动方法，所述方法包括：

响应于第一虚拟对象对虚拟技能的释放操作，在虚拟场景中创建第一可飞行道具，所述第一可飞行道具与所述虚拟技能关联，所述第一虚拟对象属于第一阵营；

控制所述第一可飞行道具沿第一飞行轨迹进行移动，所述第一飞行轨迹为基于所述释放操作确定的轨迹；

在所述第一可飞行道具的目标范围内检测到第二可飞行道具的情况下，控制所述第一可飞行道具沿第二飞行轨迹进行移动，所述第二可飞行道具由第二虚拟对象释放，所述第二虚拟对象属于第二阵营，所述第二飞行轨迹与所述第一飞行轨迹不同。

一方面，提供了一种基于可飞行道具的互动装置，所述装置包括：

创建模块，用于响应于第一虚拟对象对虚拟技能的释放操作，在虚拟场景中创建第一可飞行道具，所述第一可飞行道具与所述虚拟技能关联，所述第一虚拟对象属于第一阵营；

第一控制模块，用于控制所述第一可飞行道具沿第一飞行轨迹进行移动，所述第一飞行轨迹为基于所述释放操作确定的轨迹；

第二控制模块，用于在所述第一可飞行道具的目标范围内检测到第二可飞行道具的情况下，控制所述第一可飞行道具沿第二飞行轨迹进行移动，所述第二可飞行道具由第二虚拟对象释放，所述第二虚拟对象属于第二阵营，所述第二飞行轨迹与所述第一飞行轨迹不同。

一方面，提供了一种电子设备，该电子设备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计算机程序，该至少一条计算机程序由该一个或多个处理器加载并执行以实现如上述基于可飞行道具的互动方法。

一方面，提供了一种计算机可读存储介质，该计算机可读存储介质中存储有至少一条计算机程序，该至少一条计算机程序由处理器加载并执行以实现如上述基于可飞行道具的互动方法。

一方面，提供一种计算机程序产品，所述计算机程序产品包括一条或多条计算机程序，所述一条或多条计算机程序存储在计算机可读存储介质中。电子设备的一个或多个处理器能够从计算机可读存储介质中读取所述一条或多条计算机程序，所述一个或多个处理器执行所述一条或多条计算机程序，使得电子设备能够执行上述基于可飞行道具的互动方法。

本申请实施例提供的技术方案带来的有益效果至少包括：

通过在第一可飞行道具的目标范围内检测到第二可飞行道具时，由于第一可飞行道具和第二可飞行道具的阵营不同，控制第一可飞行道具从第一飞行轨迹切换到第二飞行轨迹，这使得第一可飞行道具的飞行轨迹随着游戏对局的进行而富于变化，提升了对第一可飞行道具的道具资源利用率，丰富了基于可飞行道具的交互方式，提高了人机交互效率。

附图说明

图1是本申请实施例提供的一种典型的MOBA游戏的虚拟世界；
图2是本申请实施例提供的一种以蓝方视角观察到的虚拟世界的示意图；
图3是本申请实施例提供的一种以红方视角观察到的虚拟世界的示意图；
图4是本申请实施例提供的一种在终端上显示用户界面的示意图；
图5是本申请实施例提供的一种在终端上显示用户界面的示意图；
图6是本申请实施例提供的一种虚拟世界的示意图；
图7是本申请实施例提供的一种虚拟世界的示意图；
图8是本申请实施例提供的另一种典型的MOBA游戏的镜像虚拟世界的示意图；
图9是本申请实施例提供的另一种典型的MOBA游戏的镜像虚拟世界的示意图；
图10是本申请实施例提供的另一种典型的MOBA游戏的镜像虚拟世界的示意图；
图11是本申请实施例提供的一种基于可飞行道具的互动方法的实施环境示意图；
图12是本申请实施例提供的一种基于可飞行道具的互动方法的流程图；
图13是本申请实施例提供的一种单个第一可飞行道具的射线轨迹的示意图；
图14是本申请实施例提供的一种多个第一可飞行道具的射线轨迹的示意图；
图15是本申请实施例提供的一种单个第一可飞行道具的环形轨迹的示意图；
图16是本申请实施例提供的一种多个第一可飞行道具的环形轨迹的示意图；
图17是本申请实施例提供的一种基于可飞行道具的互动方法的流程图；
图18是本申请实施例提供的一种碰撞检测方式的原理图；
图19是本申请实施例提供的一种筛选第二可飞行道具的原理图；
图20是本申请实施例提供的一种蓝图工具的示意图；
图21是本申请实施例提供的一种基于可飞行道具的互动方法的原理性示意图；
图22是本申请实施例提供的一种基于可飞行道具的互动装置的结构示意图；
图23是本申请实施例提供的一种终端的结构示意图；
图24是本申请实施例提供的一种电子设备的结构示意图。

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。

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。

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，多个可飞行道具是指两个或两个以上的可飞行道具。

本申请中术语“包括A或B中至少一项”涉及如下几种情况：仅包括A，仅包括B，以及包括A和B两者。

本申请中涉及到的用户相关的信息（包括但不限于用户的设备信息、个人信息、行为信息等）、数据（包括但不限于用于分析的数据、存储的数据、展示的数据等）以及信号，当以本申请实施例的方法运用到具体产品或技术中时，均为经过用户许可、同意、授权或者经过各方充分授权的，且相关信息、数据以及信号的收集、使用和处理需要遵守相关国家和地区

的相关法律法规和标准。例如，本申请中涉及到的用户的操作或指令都是在充分授权的情况下获取的。

相关技术中，存在一类虚拟技能，虚拟对象在释放这类虚拟技能后，在虚拟场景中创建一个按照预设的飞行轨迹进行移动的可飞行道具。由于可飞行道具的飞行轨迹固定不变、生效时长也有限，因此，可飞行道具的道具资源利用率较低，基于可飞行道具的交互方式单一，人机交互效率低。而本申请实施例中，提供了一种基于可飞行道具的互动方法，能够提升对第一可飞行道具的道具资源利用率，丰富基于可飞行道具的交互方式，提高人机交互效率，详见下述说明。

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抗，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，该虚拟场景中包括对称的两个区域，属于两个不同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。

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、虚拟精灵、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。

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，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。

虚拟技能：指虚拟环境中的虚拟对象可以释放的固有互动技能，通常，不同的虚拟对象绑定有不同的专属虚拟技能，但也存在一些不同虚拟对象可以共享的通用虚拟技能，用户可以选择装配哪些通用虚拟技能到本次对局中，局内不可变更已装配的通用虚拟技能，并且局内可装配的通用虚拟技能的数量存在上限。虚拟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、收割技能等。通常，虚拟对象在释放虚拟技能后，虚拟技能具有一段技能 CD（Cool Down，冷却），用户在技能 CD 期间无法再次释放该虚拟技能，直到技能 CD 结束以后，虚拟技能才会再次恢复到可释放的状态，不同的虚拟技能可用具有相同或不同的技能 CD 长短，并且，也并非每种虚拟技能都具有技能 CD，没有技能 CD 的虚拟技能可以视为技能 CD 为 0。

可飞行道具：指虚拟对象释放虚拟技能后，由虚拟技能所产生、召唤、释放或创建出来的可飞行的虚拟道具，可飞行道具也可视为是虚拟技能的召唤物，或者视为是虚拟对象的召唤物。通常，由虚拟技能所产生的可飞行道具会沿着预设的飞行轨迹进行移动，如，沿着从虚拟对象出发指向摇杆方向的射线轨迹进行移动，直到可飞行道具到达生效时长（或者到达飞行射程，或者碰撞到其他阵营的虚拟对象造成一定的虚拟伤害值）。

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抗，占领据点或摧毁对方阵营据点的游戏。例如，MOBA 游戏可将用户分成至少两个阵营，分属至少两个阵营的不同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：占领据点或摧毁对方阵营据点、击败对方阵营的虚拟对象、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，以摧毁或占领对方区域深处的目标建筑/据点/基地/水晶作为胜利条件。

可选地，每个队伍包括一个或多个虚拟对象，比如 1 个、2 个、3 个或 5 个，根据参与游戏对局中的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。

可选地，MOBA 游戏以对局（或称为回合）为单位来进行，每场对局中选取的场景地图可以相同，也可以不同。每局 MOBA 游戏的持续时间是从小游戏开始的时刻至任一队伍或阵营达成胜利条件的时刻。

在 MOBA 游戏中，用户可以在不同对局中选取主操的不同虚拟对象，需要说明的是，每局中选取完毕的主操虚拟对象在本场对局中不可变更，但可以在其他对局中选取不同的主操虚拟对象。用户在选取完毕主操虚拟对象以后，能够控制主操虚拟对象在虚拟场景中释放虚拟技能，从而达到与对方阵营的其他虚拟对象进行互动的效果。

下面分别介绍两种典型的 MOBA 游戏。

第一种典型的 MOBA 游戏。

图 1 是一种典型的 MOBA 游戏的虚拟世界二维地图。在这种典型的 MOBA 游戏中虚拟角色被分为红方和蓝方两个阵营，每个阵营分别有五个虚拟角色，一共十个虚拟角色共同进行一场 MOBA 游戏对局。

如图 1 所示，虚拟世界地图呈正方形，且被分为以下几个部分，在正方形的一个对角线的两端分别有两个阵营的基地（水晶），即蓝方基地 1001 和红方基地 1002；连接蓝方基地 1001 和红方基地 1002 的三条进攻路线，分别是：上路 1003、中路 1004、下路 1005；公共区域：河道 1006，以及野区 1007。

两个阵营的虚拟角色分别出生在各自的基地位置，同一阵营的五个虚拟角色分别沿三个进攻方向向对方发起进攻，摧毁对方阵营的基地即可获得游戏的胜利。其中，蓝方阵营出生在蓝方基地 1001，红方阵营出生在红方基地 1002，并且两个阵营的虚拟角色都以我方基地位于观察视角的左下角的视角观察虚拟世界。即蓝方虚拟角色以第一视角 1008 观察虚拟世界，红方虚拟角色以第二视角 1009 观察虚拟世界，并且针对各自的视角，三个进攻方向从左向右分别是上路、中路和下路。示例性的，如图 2 所示，是以蓝方虚拟角色的第一视角 1008 观察到的虚拟世界，其中蓝方基地 1001 位于虚拟世界画面的左下角；如图 3 所示，是以红方虚拟角色的第二视角 1009 观察到的虚拟世界，其中红方基地 1002 位于虚拟世界画面的左下角。

这样设置两个阵营的视角，无论用户控制的虚拟角色属于红方阵营或是属于蓝方阵营，对方阵营的基地永远在虚拟世界画面的右上角，虚拟角色的进攻方向也永远是虚拟世界画面的右上方向，有助于用户操控虚拟角色。但这样的设置也存在一个问题，即蓝方的下路是红方的上路，当蓝方虚拟角色和红方虚拟角色都位于蓝方下路与红方上路的交界处（河道）时，蓝方用户在终端上看到的用户界面如图 4 所示，有部分虚拟世界画面被 UI（User Interface，用户界面）控件 1010 遮挡住，但较危险的河道 1006 区域（红方的虚拟角色，比如刺客，可能从河道 1006 突然发起进攻）并没有被遮挡，因此蓝方用户的视野较为开阔。而红方在终端上看到的用户界面如图 5 所示，同样有部分虚拟世界画面被 UI 控件 1010 遮挡，并且较危险的河道 1006 区域会被 UI 控件遮挡住，影响了红方用户的视野，导致红方用户不便于观察河道 1006 区域，容易被蓝方的刺客击败。

因此，下路 1005 相比于上路 1003 更加安全。

同一阵营的五个虚拟角色通常为五个不同类型的虚拟角色，示例性的，虚拟角色的类型可以是：

战士：血量较多，防御力较高，攻击力较高，攻击距离短，移动较灵活，通常具有一定的位移技能，可以在一定程度上抵挡对方伤害，或对对方造成伤害。位移技能是可以使虚拟角色加快移动速度，或向某一方向冲出一段距离，或从一个点瞬间移动到另一点的技能。

法师：血量极低，防御力极低，攻击力非常高且为法术伤害，攻击距离长，移动不灵活，容易被击败，因此通常在战士或坦克/辅助的保护下对对方进行攻击。

坦克/辅助：血量非常多，防御力非常高，攻击力极低，攻击距离短，通常适合在队伍前方为队友抵挡伤害，保护其他队友。

射手：与法师类似，不同的是射手具有的是非常高的物理伤害，适合持续性输出，以及攻击防御塔和基地。

刺客：血量较低，防御力较低，攻击力高，攻击距离短，移动非常灵活，通常具有很多位移技能，适合对对方的法师、或射手发起突击，有瞬间击败对方法师或射手的能力。

由于不同类型的虚拟角色有各自特点，结合上路和下路在视野上的优劣势，不同类型的虚拟角色通常以固定的进攻方向向对方发起进攻。通常由射手（和坦克/辅助）从较安全的路 1005 向对方发起进攻；由法师从中路 1004 向对方发起进攻；由具有一定位移优势的战士从较危险的上路 1003 向对方发起进攻；刺客则主要活动在野区 1007，伺机支援上路中路 1004 或下路 1005 的队友。

这样会导致虚拟角色会同与自己类型不同的对方虚拟角色对抗，蓝方的射手对抗红方的战士，而蓝方的战士对抗红方的射手，影响游戏的公平和用户的体验。示例性的，如图 6 所示，蓝方的射手一号 1011 从蓝方的下路 1005 向红方进攻，蓝方的战士一号 1012 从蓝方的上路 1003 向红方进攻，红方的射手二号 1013 从红方的下路 1005 向蓝方发起进攻，红方的战士二号 1014 从红方的上路 1003 向蓝方发起进攻。即射手一号 1011 对抗战士二号 1014，战士一号 1012 对抗射手二号 1013。

而为了使游戏更加公平，更加合理的对抗方式如图 7 所示，蓝方的射手一号 1011 对抗红方的射手二号 1013，蓝方的战士一号 1012 对抗红方的战士二号 1014。为了实现这样的对抗方式，需要解决如何使蓝方的下路和红方的下路为同一条路线的问题，即，将蓝方或红方其中一方的上下路互换，使原本的下路变为上路，原本的上路变为下路。示例性的，将红方的上下路变为如图 7 所示的上路 1003 和下路 1005 的位置。使蓝方的下路 1005 同样为红方的下路 1005，蓝方的上路 1003 同样为红方的上路 1003。

第二种典型的 MOBA 游戏即实现了这种更加合理的对抗方式。

第二种典型的 MOBA 游戏。

第二种典型的 MOBA 游戏模式与第一种典型的 MOBA 游戏在游戏玩法上相同，同样是正方形的虚拟世界，第一阵营和第二阵营的基地同样位于正方形的对角线上，同样是个阵营五个虚拟角色分别沿三个进攻方向向对方发起进攻。不同的是，第一阵营的下路同时也是第二阵营的下路，第一阵营的上路同时也是第二阵营的上路。第二种典型的 MOBA 游戏是以下面的方法实现这种更加合理的对抗方式的。

游戏对局中具有第一虚拟世界和关于第一虚拟世界地平面镜像的第二虚拟世界。如图 8 所示，游戏对局中具有第一虚拟世界 1101 以及第一虚拟世界 1101 关于地平面 1102 对称的第二虚拟世界 1103，即，第二虚拟世界是第一虚拟世界的镜像倒影。

若以垂直于第一虚拟世界地平面且指向天空的方向为 y 轴正半轴方向 1104，则，控制第一阵营虚拟角色的用户看到的虚拟世界，是在视角位于 y 轴正半轴的空间所观察到的第一虚拟世界，如图 9 所示，是控制第一阵营虚拟角色的用户观察到的第一虚拟世界。控制第二阵营虚拟角色的用户看到的虚拟世界，是在视角位于 y 轴负半轴的空间所观察到的第二虚拟世界，如图 10 所示，是控制第二阵营虚拟角色的用户观察到的第二虚拟世界。可以看出，第一

虚拟世界 1101 和第二虚拟世界 1103 是左右相反的世界，这种方法可实现将第二阵营的上下路对调，控制第二阵营虚拟角色的用户看到的下路同样也是控制第一阵营虚拟角色的用户看到的下路。

第二种典型的 MOBA 游戏，将两个互为镜像的虚拟世界分别展示给两个阵营的用户，其中第一阵营的用户以 y 轴正半轴的视角观察第一虚拟世界，并控制虚拟角色在第一虚拟世界活动；第二阵营的用户以 y 轴负半轴的视角观察第二虚拟世界，并控制虚拟角色在第二虚拟世界活动。由于第一虚拟世界和第二虚拟世界是完全相反的两个世界，服务器需要为第一虚拟世界和第二虚拟世界分别设置两套运算逻辑，用第一运算逻辑来计算第一阵营的虚拟角色在第一虚拟世界的活动信息，例如：移动位置、技能释放方向等，用第二运算逻辑来计算第二阵营的虚拟角色在第二虚拟世界的活动信息，之后还需要将一个虚拟世界的运算结果显示到另一个虚拟世界上。

以下，对本申请涉及的系统架构进行介绍。

图 11 是本申请实施例提供的一种基于可飞行道具的互动方法的实施环境示意图。参见图 11，该实施环境包括：第一终端 120、服务器 140 和第二终端 160。

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器械类生存游戏中的任何一种。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：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、释放虚拟技能中的至少一种。示意性的，第一虚拟对象是第一虚拟人物，比如仿真人物角色或动漫人物角色。

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。

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器械类生存游戏中的任何一种。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：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、释放虚拟技能中的至少一种。示意性的，第二虚拟对象是第二虚拟人物，比如仿真人物角色或动漫人物角色。

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。

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第二虚拟对象进行互动。

在一些实施例中，上述第一虚拟对象以及第二虚拟对象可以为对抗关系，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，对抗关系的虚拟对象之间，能够以互相释放虚拟技能的方式进行对抗，比如，第一虚拟对象向第二虚拟对象释放攻击技能。在后续实施例中，以第一虚拟对象和第二虚拟对象互为对抗关系为例进行说明。

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，第一虚拟对象和第二虚拟对象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辅助技能。

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。

上述第一终端 120 或者第二终端 160 可以是智能手机、智能掌机、便携式游戏设备、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。

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。

图 12 是本申请实施例提供的一种基于可飞行道具的互动方法的流程图。参见图 12，该实施例由电子设备执行，以电子设备为终端为例进行说明，该实施例包括以下步骤：

1201、终端响应于第一虚拟对象对虚拟技能的释放操作，在虚拟场景中创建第一可飞行道具，该第一可飞行道具与该虚拟技能关联，该第一虚拟对象属于第一阵营。

本申请实施例涉及的第一虚拟对象，是指用户使用终端主操的虚拟对象，第一虚拟对象所属的阵营称为第一阵营。示意性地，在 MOBA 游戏中涉及到两个阵营，第一虚拟对象所属的第一阵营，以及第二虚拟对象所属的第二阵营，第一阵营和第二阵营是不同的阵营，例如，第一阵营和第二阵营互为对抗关系。

本申请实施例涉及的虚拟技能，是指第一虚拟对象能够释放的任一可召唤可飞行道具的虚拟技能，即，该虚拟技能可以是第一虚拟对象绑定或专属的虚拟技能，也可以是所有虚拟对象通用的虚拟技能，这里对虚拟技能不进行具体限定。

本申请实施例涉及的可飞行道具，是指第一虚拟对象释放该虚拟技能后所产生、召唤、释放或创建的能够在虚拟场景中飞行的虚拟道具，例如，该可飞行道具是虚拟技能的发射物、投掷物、召唤物、飞行跟宠、飞行载具等。该第一可飞行道具与该虚拟技能关联，可以理解为该虚拟技能用于产生、召唤、释放或者创建该第一可飞行道具。

在一些实施例中，用户在终端上启动游戏应用，在游戏应用中执行开局操作，以开启一场游戏对局，接着，终端在游戏应用中加载并显示游戏对局的虚拟场景，并在虚拟场景中显示终端所主操的第一虚拟对象。可选地，在开局之前，用户可以在开局界面中，从可供选择的多个虚拟对象中，选择本次游戏对局要出场的第一虚拟对象，例如，用户从自身已拥有的英雄池中选择符合本次分路定位的主操英雄作为第一虚拟对象，或者如果用户未作出选择，系统可以为用户分配本次游戏对局的第一虚拟对象，分配方式可以是随机分配，可以优先选择用户所绑定账号中出场频次最高的英雄，或者选择用户所绑定账号中操作评分最高的英雄等，本申请实施例对此不进行具体限定。

在一些实施例中，在完成对第一虚拟对象的选择或分配后，在游戏对局中将自动携带或装配第一虚拟对象的专属虚拟技能，同时，在开局界面中，用户还可以选择指定数量的通用虚拟技能，以将通用虚拟技能也携带到本次游戏对局中，使得第一虚拟对象既能够释放专属虚拟技能也能够释放通用虚拟技能，其中，指定数量由游戏应用的业务逻辑决定，这里不做限定。

在一些实施例中，在对虚拟场景加载完毕后，用户能够控制第一虚拟对象在虚拟场景中进行互动，可选地，在第一虚拟对象具有可产生、召唤、释放或创建可飞行道具的虚拟技能

的情况下，终端若检测到用户对该虚拟技能的释放操作，将在虚拟场景中创建与该虚拟技能所绑定的第一可飞行道具，第一可飞行道具是指第一虚拟对象通过虚拟技能所创建的可飞行道具。

在一些实施例中，上述对虚拟技能的释放操作包括但不限于：对虚拟技能的技能控件的点击操作、触摸操作、按压操作、拖拽操作、滑动操作等，用户还能够通过摇杆控件来控制虚拟技能的释放方向，本申请实施例对释放操作的实施方式不进行具体限定。

在一些实施例中，终端检测到对虚拟技能的释放操作以后，判断该虚拟技能是否能够召唤可飞行道具，如果该虚拟技能能够召唤可飞行道具，在缓存中以虚拟技能的技能 ID (Identification, 标识) 为索引进行查询，以获取到与该技能 ID 关联存储的道具资源，这一道具资源即用于显示第一可飞行道具，接着，终端利用渲染引擎对该道具资源进行渲染，以将第一可飞行道具显示在虚拟场景中。可选地，终端中缓存有技能配置信息，该技能配置信息包括技能标识与道具资源标识之间的关联关系，技能标识与道具标识之间的关联关系表示该技能标识指示的虚拟技能与该道具资源标识指示的虚拟道具关联，因此终端可以根据该技能配置信息确定与该虚拟技能关联的该第一可飞行道具的道具资源，然后通过对该道具资源进行渲染，得到该第一可飞行道具。

1202、终端控制该第一可飞行道具沿第一飞行轨迹进行移动，该第一飞行轨迹为基于该释放操作确定的轨迹。

在一些实施例中，终端可以基于用户在步骤 1201 中执行的对虚拟技能的释放操作，确定该虚拟技能的释放方向，接着，根据该虚拟技能的释放方向来生成第一飞行轨迹。

可选地，该虚拟技能的释放方向可以是用户通过摇杆控件来指定的，比如，用户手指按住摇杆控件选定释放方向以后，再执行对虚拟技能的释放操作，这样可以直接获取基于摇杆控件所确定的释放方向，该释放方向可以是终端屏幕中摇杆控件的中心点指向用户手指的触摸点所指示的方向。这样用户可以一只手按下技能控件来触发释放操作，另一只手控制摇杆控件来精细调整释放方向，具有较高的操作自由度。

可选地，该虚拟技能的释放方向还可以是用户通过技能控件来指定的，比如，用户手指按住虚拟技能的技能控件以后，可以向释放方向进行滑动操作，并在松手后（即滑动操作结束后）触发释放虚拟技能，该释放方向可以是该滑动操作的滑动方向。这样用户可以直接单手对技能控件进行操作，同时选定释放方向并触发释放操作，能够简化用户的人机交互流程。

可选地，该虚拟技能的释放方向还可以是用户选定释放目标以后，由终端自动确定出来的释放方向，比如，用户将互动焦点锁定在某个释放目标以后，用户所触发的虚拟技能都将自动锁定到释放目标上，其中，释放目标可以是对方阵营的虚拟对象，也可以是一些如野怪、小兵、基地等非玩家控制的虚拟对象。例如，终端自动将当前帧中从第一虚拟对象指向释放目标的方向作为虚拟技能的释放方向。

在一些实施例中，在虚拟技能仅释放单个第一可飞行道具时，第一飞行轨迹可以是以第一虚拟对象当前位置为起点、且指向虚拟技能的释放方向的一条射线轨迹，这样终端可以控制第一可飞行道具沿着该射线轨迹进行移动。

如图 13 所示，图 13 是本申请实施例提供的一种单个第一可飞行道具的射线轨迹的示意图，在虚拟场景 1300 中显示有第一虚拟对象 1301 和第二虚拟对象 1302，第一虚拟对象 1301 属于第一阵营，第二虚拟对象 1302 属于第二阵营，第一阵营和第二阵营互为对抗关系。在一个示例中，第一虚拟对象 1301 装配有专属虚拟技能 1311~1313 以及通用虚拟技能 1314~1315。可选地，用户控制第一虚拟对象 1301 释放虚拟技能 1311 时，将会在虚拟场景 1300 中创建虚拟技能 1311 召唤出来的第一可飞行道具 1321，并且，第一可飞行道具 1321 会沿着一条射线轨迹 1322 进行移动，射线轨迹 1322 是一条从第一虚拟对象 1301 指向第二虚拟对象 1302 的射线，射线轨迹 1322 为第一飞行轨迹的一种示例性说明。

在另一些实施例中，在虚拟技能可释放多个第一可飞行道具时，第一飞行轨迹可以是以第一虚拟对象当前位置为起点，且指向虚拟技能的释放方向的多条射线轨迹，多条射线轨迹

之间的角度差可以是相同或者不同的。比如，将多个第一可飞行道具中位于中心的第一可飞行道具的射线轨迹设置为与释放方向重叠，其余的第一可飞行道具均与相邻的第一可飞行道具的射线轨迹之间保持恒定的角度差，这样能够营造出来类似于多个第一可飞行道具的散射效果。

如图 14 所示，图 14 是本申请实施例提供的一种多个第一可飞行道具的射线轨迹的示意图，在虚拟场景 1400 中显示有第一虚拟对象 1401 和第二虚拟对象 1402，第一虚拟对象 1401 属于第一阵营，第二虚拟对象 1402 属于第二阵营，第一阵营和第二阵营互为对抗关系。在一个示例中，第一虚拟对象 1401 装配有专属虚拟技能 1411~1413 以及通用虚拟技能 1414~1415。可选地，用户控制第一虚拟对象 1401 释放虚拟技能 1411 时，将会在虚拟场景 1400 中创建虚拟技能 1411 召唤出来的多个第一可飞行道具 1421，这里以召唤出来 5 个第一可飞行道具 1421 为例进行说明。每个第一可飞行道具 1421 都会沿着一条射线轨迹 1422 进行移动，射线轨迹 1422 同样是第一飞行轨迹的一种示例性说明。

1203、终端在该第一可飞行道具的目标范围内检测到第二可飞行道具的情况下，控制该第一可飞行道具沿第二飞行轨迹进行移动，该第二可飞行道具由第二虚拟对象释放，第二虚拟对象属于第二阵营。

在一些实施例中，终端在第一可飞行道具沿着第一飞行轨迹进行移动的过程中，可以对第一可飞行道具进行碰撞检测，即，检测在第一可飞行道具的目标范围内是否存在第二可飞行道具，第二可飞行道具是由第二阵营的第二虚拟对象所释放的可飞行道具，这样，如果在第一可飞行道具的目标范围内检测到了第二可飞行道具，则控制第一可飞行道具进行飞行轨迹的变换，比如，为第一可飞行道具生成一条新的第二飞行轨迹，并控制第一可飞行道具沿着第二飞行轨迹进行移动，第二飞行轨迹与第一飞行轨迹不同。需要说明的是，如果在第一可飞行道具的目标范围内检测到了第一阵营的其他虚拟对象释放的可飞行道具，由于这些可飞行道具均属于第一阵营，因此将不会触发进行飞行轨迹的变换。

在一些实施例中，第一可飞行道具的目标范围是指三维的虚拟场景中与第一可飞行道具之间的距离不大于预设距离的范围，例如该目标范围是以第一可飞行道具为圆心、以预设距离为半径的球形区域，也即是当第二可飞行道具与该第一可飞行道具之间的距离不大于预设距离时，则认为在该第一可飞行道具的目标范围内检测到第二可飞行道具。在另一些实施例中，该目标范围是指与该第一可飞行道具位于同一水平面且与该第一可飞行道具之间的距离不大于预设距离的范围，例如该目标范围是第一可飞行道具所在的水平面中以该第一可飞行道具为圆心、以预设距离为半径的圆形区域，也即是当第二可飞行道具与该第一可飞行道具位于同一水平面，且第二可飞行道具与该第一可飞行道具之间的距离不大于预设距离时，则认为在该第一可飞行道具的目标范围内检测到第二可飞行道具。或者，该第一可飞行道具的目标范围是指该第一可飞行道具所占据的位置，也即是当第二可飞行道具所占据的位置与该第一可飞行道具所占据的位置存在重叠部分时，则认为在该第一可飞行道具的目标范围内检测到第二可飞行道具。

在一些实施例中，在虚拟技能仅释放单个第一可飞行道具时，第二飞行轨迹可以是第一虚拟对象包围在内的环形轨迹，例如，第二飞行轨迹是以第一虚拟对象为圆心的环形轨迹，这样终端在检测到第一可飞行道具的目标范围内存在第二可飞行道具以后，此后将控制第一可飞行道具沿着第二飞行轨迹进行移动，即，相当于在第一可飞行道具碰撞到了第二可飞行道具以后，将第一可飞行道具原本沿着射线轨迹进行移动，切换到第一可飞行道具沿着环形轨迹围绕第一虚拟对象进行移动。

如图 15 所示，图 15 是本申请实施例提供的一种单个第一可飞行道具的环形轨迹的示意图，在图 13 中所示的用户控制第一虚拟对象 1301 释放了虚拟技能 1311，从而在虚拟场景 1300 中召唤出来第一可飞行道具 1321 的基础上，如果第一可飞行道具 1321 在沿着第一飞行轨迹进行飞行的过程中，在第一可飞行道具 1321 的目标范围内检测到了第二虚拟对象 1302 所释放的第二可飞行道具（图 15 中未示出），那么视为第一可飞行道具 1321 碰撞到了第二可

飞行道具，这时，将为第一可飞行道具 1321 生成一条新的环形轨迹 1500，环形轨迹 1500 可以是一个将第一虚拟对象 1301 包围在内的圆环或椭圆环，第一虚拟对象 1301 可以处于环形轨迹 1500 的中心或者内部任一位置，环形轨迹 1500 是第二飞行轨迹的一种示例性说明。

对比图 13 和图 15，可以看出，原本第一可飞行道具将会按照预设的第一飞行轨迹进行移动，并在到达生效时长，或者到达飞行射程，或者碰撞到其他虚拟对象（并造成一定的作用效果，可以是增益效果或者是减益效果等）后消失。但本申请实施例提供的技术方案，能够在第一可飞行道具碰撞到第二可飞行道具以后，提供一种新型的变换飞行轨迹的交互方式，第一可飞行道具将从原本的射线轨迹切换到环形轨迹上，这极大提升了第一可飞行道具的道具资源利用率，并且飞行方式也变得更加丰富多彩，提高了人机交互效率。

在另一些实施例中，在虚拟技能可释放多个第一可飞行道具时，第二飞行轨迹可以是将第一虚拟对象包围在内的多条环形轨迹，多条环形轨迹之间可以是呈同心圆的布局方式并以第一虚拟对象为圆心，或者，多条环形轨迹之间还可以是交错的一系列非同心圆，本申请实施例对此不进行具体限定。这样终端在检测到任一第一可飞行道具的目标范围内存在第二可飞行道具以后，此后将控制虚拟技能释放的全部第一可飞行道具均沿着各自的第二飞行轨迹进行移动，即，相当于在任一第一可飞行道具碰撞到了第二可飞行道具以后，将全部第一可飞行道具原本沿着各自的射线轨迹进行移动，切换到全部第一可飞行道具沿着各自的环形轨迹围绕第一虚拟对象进行移动。

如图 16 所示，图 16 是本申请实施例提供的一种多个第一可飞行道具的环形轨迹的示意图，在图 14 中所示的用户控制第一虚拟对象 1401 释放了虚拟技能 1411，从而在虚拟场景 1400 中召唤出来多个第一可飞行道具 1421 的基础上，如果任一第一可飞行道具 1421 在沿着第一飞行轨迹进行飞行的过程中，在任一第一可飞行道具 1421 的目标范围内检测到了第二虚拟对象 1402 所释放的第二可飞行道具（图 16 中未示出），那么视为该第一可飞行道具 1421 碰撞到了第二可飞行道具，这时，将为每个第一可飞行道具 1421 均生成一条新的环形轨迹 1600，环形轨迹 1600 可以是一个将第一虚拟对象 1401 包围在内的圆环或椭圆环，不同环形轨迹 1600 可以构成同心圆环或非同心圆环，第一虚拟对象 1401 可以处于各个环形轨迹 1600 的中心或者内部任一位置，环形轨迹 1600 是第二飞行轨迹的一种示例性说明。

对比图 14 和图 16，可以看出，原本多个第一可飞行道具将会按照各自预设的第一飞行轨迹进行移动，并在到达生效时长，或者到达飞行射程，或者碰撞到其他虚拟对象（并造成一定的作用效果，可以是增益效果或者是减益效果等）后消失。但本申请实施例提供的技术方案，能够在任一第一可飞行道具碰撞到第二可飞行道具以后，提供一种新型的变换飞行轨迹的交互方式，将每个第一可飞行道具都将从原本的射线轨迹切换到环形轨迹上，这极大提升了第一可飞行道具的道具资源利用率，并且飞行方式也变得更加丰富多彩，提高了人机交互效率。

在以上过程中，由于原本 MOBA 游戏中，每类可飞行道具通常都专属于指定虚拟对象的指定虚拟技能，而可飞行道具的生效时长有限、飞行方式固定（沿着第一飞行轨迹移动），这使得大量技术人员设计好的可飞行道具的道具资源利用率低，而在本申请实施例提供的技术方案中，由于设计了一套创新的基于可飞行道具的交互方式，使得不同阵营的可飞行道具发生碰撞时，将会改变一方的可飞行道具的飞行轨迹，使得可飞行道具的飞行轨迹富于变化，而不会局限于预设的飞行轨迹，丰富了游戏表现力，提高了道具资源利用率，提高了人机交互效率。

并且，由于可飞行道具在沿固定飞行轨迹移动时碰撞到其他阵营的可飞行道具后会改变飞行轨迹，因此不同阵营的虚拟对象在进行对局的过程中，如果第一阵营的虚拟对象释放虚拟技能召唤了第一可飞行道具沿着第一飞行轨迹去攻击第二阵营的虚拟对象，则第二阵营的虚拟对象可以释放虚拟技能来召唤第二可飞行道具去碰撞第一可飞行道具，以使第一可飞行道具改变飞行轨迹导致无法攻击到第二阵营的虚拟对象，相当于可以通过碰撞可飞行道具该对可飞行道具进行拦截，使得可飞行道具无法发挥作用，一方面提高了可飞行道具的灵活性，

一方面也扩展了可飞行道具的功能，促使玩家围绕可飞行道具积极制定对局策略以及使用可飞行道具灵活调整对局策略，丰富了游戏玩法，有利于激发玩家参与对局的积极性，提高了人机交互效率，提高了玩家的对局体验。

上述所有可选技术方案，能够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。

本申请实施例提供的技术方案，通过在第一可飞行道具的目标范围内检测到第二可飞行道具时，由于第一可飞行道具和第二可飞行道具的阵营不同，控制第一可飞行道具从第一飞行轨迹切换到第二飞行轨迹，这使得第一可飞行道具的飞行轨迹随着游戏对局的进行而富于变化，提升了对第一可飞行道具的道具资源利用率，丰富了基于可飞行道具的交互方式，提高了人机交互效率。

在上一实施例中，简单介绍了基于可飞行道具的互动流程，而在本申请实施例中，将详细说明各个步骤的细节实施方式。图 17 是本申请实施例提供的一种基于可飞行道具的互动方法的流程图。参见图 17，该实施例由电子设备执行，以电子设备为终端为例进行说明，该实施例包括以下步骤：

1701、终端响应于第一虚拟对象对虚拟技能的释放操作，在虚拟场景中创建第一可飞行道具，该第一可飞行道具与该虚拟技能关联，该第一虚拟对象属于第一阵营。

上述步骤 1701 与上一实施例中步骤 1201 同理，这里不做赘述。

在一些实施例中，终端可以通过配置文件来创建并管理第一可飞行道具，配置文件中涉及到多个轨道，每个轨道用于管理第一可飞行道具的一种业务逻辑。示意性地，第一可飞行道具的配置文件可以涉及有五个轨道，从上到下依次包括：

- 1 | SpawnObjectDuration0 //生成一个实体
- 2 | SetCollisionTick0 //为该实体添加碰撞检测框
- 3 | TriggerParticle3 //为该实体添加特效
- 4 | MoveBulletDuration4 //移动该实体
- 5 | HitTriggerDuration0 //对该实体添加碰撞检测逻辑

上述实体即指代虚拟场景中创建的第一可飞行道具的第一实体，也即是该第一可飞行道具的初始实体，上述特效即指代虚拟场景中创建的第一可飞行道具的第一实体携带的第一特效，也即是该第一可飞行道具的初始特效。

在游戏应用的运行过程中，终端可以在缓存中查找到第一可飞行道具的配置文件，并运行该配置文件，以按照从上到下的顺序来依次执行每个轨道的业务逻辑，从而在虚拟场景中创建出来一个携带有特效的实体作为第一可飞行道具，并且第一可飞行道具会按照释放操作所指定的释放方向来进行移动，即，第一可飞行道具会按照第一飞行轨迹来进行移动，并对沿途中碰撞到的非友方的虚拟对象产生一定的作用效果。上述配置文件即为第一可飞行道具的道具资源的一种可能实施方式。

在一个示例性场景中，以第一可飞行道具为虚拟技能的发射物为例，终端在检测到用户控制第一虚拟对象释放虚拟技能时，会调用游戏应用的功能节点“1 | SpawnBulletTick0”来产生一个 Bullet（弹丸）对象，该 Bullet 对象为游戏应用内的一种 Actor（作用物）对象，接着，以虚拟技能的技能 ID 为索引，在缓存中查询到第一可飞行道具的配置文件，并按照配置文件中各个轨道所提供的业务逻辑，来实现对 Bullet 对象的初始化配置，从而能够在虚拟场景中显示出第一可飞行道具，并能够控制第一可飞行道具沿着第一飞行轨迹进行移动。

1702、终端控制该第一可飞行道具沿第一飞行轨迹进行移动，该第一飞行轨迹为基于该释放操作确定的轨迹。

上述步骤 1702 与上一实施例中步骤 1202 同理，这里不做赘述。

示意性地，终端可以通过第一可飞行道具的配置文件中的轨道“4 | MoveBulletDuration4”来定义第一飞行轨迹，例如，将第一飞行轨迹定义为：以第一虚拟对象为起点、指向虚拟技

能的释放方向的射线轨迹，或者，将第一飞行轨迹定义为：指向虚拟技能的释放方向的直线轨迹，或者还可以将第一飞行轨迹按照设定的轨迹函数来定义成弧线轨迹、曲线轨迹、抛物线轨迹等，这里对第一飞行轨迹的具体类型不进行限定。

需要说明的是，第一飞行轨迹是指配置文件中对第一可飞行道具所预设的固定轨迹，是不会受到游戏对局的进行而改变的，即，不管用户在游戏对局的哪一时刻来控制第一虚拟对象释放虚拟技能，创建的第一可飞行道具都会先按照第一飞行轨迹进行移动。

1703、终端在该第一可飞行道具的目标范围内检测到第二可飞行道具的情况下，控制该第一可飞行道具在该第一飞行轨迹上停止移动，该第二可飞行道具由第二阵营的第二虚拟对象释放。

在一些实施例中，终端可以为第一可飞行道具添加一个碰撞检测框，碰撞检测框可以是一个挂载在第一可飞行道具的道具模型上的碰撞检测盒子（也可称为碰撞检测范围），使得碰撞检测框可以随着第一可飞行道具移动而随之移动，从而保证第一可飞行道具在移动过程中能够实时进行碰撞检测。可选地，碰撞检测框所圈定的范围即为目标范围，技术人员可以在配置文件中通过轨道“2|SetCollisionTick0”来定义目标范围的尺寸，这里对目标范围的尺寸不进行具体限定。需要说明的是，如果虚拟技能释放多个第一可飞行道具，可以对每个第一可飞行道具都挂载一个碰撞检测框，或者，如果多个第一可飞行道具的第一飞行轨迹比较紧凑，方便起见，也可以对全部第一可飞行道具挂载一个较大范围的碰撞检测框，这里不进行具体限定。

在一些实施例中，终端在为第一虚拟对象释放的第一可飞行道具挂载碰撞检测框以后，同理，第二虚拟对象所释放的第二可飞行道具也会挂载碰撞检测框。这样，终端可以通过判断第一可飞行道具的碰撞检测框是否与第二可飞行道具的碰撞检测框存在交集，从而判断第一可飞行道具的目标范围内是否包含有第二可飞行道具。当第一可飞行道具的碰撞检测框与第二可飞行道具的碰撞检测框存在交集时，确定第一可飞行道具的目标范围内包含有第二可飞行道具。当第一可飞行道具的碰撞检测框与第二可飞行道具的碰撞检测框不存在交集时，确定第一可飞行道具的目标范围内未包含第二可飞行道具。

如图18所示，图18是本申请实施例提供的一种碰撞检测方式的原理图，在虚拟场景1800中，假设第二虚拟对象1802释放虚拟技能后，召唤出来一个第二可飞行道具1810，这时第二可飞行道具1810的道具模型上会挂载一个球形的碰撞检测框1811，可以看出，第二可飞行道具1810的碰撞检测框1811与第一虚拟对象1801的角色模型上挂载的碰撞检测框1803存在交集，说明第二可飞行道具1810命中了第一虚拟对象1801，因此，第二可飞行道具1810将对第一虚拟对象1801产生相应的作用效果，例如，对第一虚拟对象1801扣除一定的虚拟生命值，或者，对第一虚拟对象1801扣除一定的虚拟防御值，或者，对第一虚拟对象1801在设定时长内降低攻击力系数等，这里对可飞行道具的作用效果不进行具体限定。

由于虚拟场景中除了第二可飞行道具以外，友方虚拟对象释放的可飞行道具、虚拟对象自身、野怪等中立虚拟对象、甚至一些虚拟物品也可能会挂载碰撞检测框，因此，终端在检测到第一可飞行道具的碰撞检测框与任一碰撞检测框存在交集的情况下，视为在第一可飞行道具的目标范围内检测到了障碍物，但需要判断障碍物是否为第二可飞行道具，才决策是否要改变第一可飞行道具的飞行轨迹。

在一些实施例中，检测目标范围内的障碍物是否为第二可飞行道具的方式包括以下步骤1713~1733：

1713、在第一可飞行道具的目标范围内检测到虚拟对象的情况下，获取该虚拟对象的对象类型。

在一些实施例中，终端在检测到第一可飞行道具的碰撞检测框与任一碰撞检测框存在交集的情况下，视为在第一可飞行道具的目标范围内检测到了障碍物，接着，查询该障碍物的类型，如果该障碍物的类型是可飞行道具，进入步骤1723，如果该障碍物的类型是虚拟对象，则根据虚拟技能的技能设定，控制第一可飞行道具对虚拟对象产生作用效果，例如降低虚拟

对象的移动速度、加长虚拟对象的技能 CD、降低虚拟对象的虚拟生命值或者虚拟防御力等。

1723、在该类型为可飞行道具的情况下，获取该可飞行道具所属的阵营。

在一些实施例中，在障碍物的对象类型为可飞行道具的情况下，还可以进一步判断可飞行道具所属的阵营，如果可飞行道具属于第二阵营，进入步骤 1733，如果可飞行道具属于第一阵营，则可以不做任何处理，继续沿着第一飞行轨迹进行移动。

1733、在该阵营为该第二阵营，且该可飞行道具满足吸附条件的情况下，将该障碍物确认为该第二可飞行道具，该吸附条件指示该可飞行道具能够变换其他可飞行道具的飞行轨迹。

在一些实施例中，在可飞行道具属于第二阵营的情况下，还可以进一步判断可飞行道具是否满足吸附条件，比如，在终端上存储有满足吸附条件的道具 ID 列表，进而在道具 ID 列表中查询是否含有在目标范围内检测到的第二阵营的可飞行道具的道具 ID，如果在满足吸附条件的道具 ID 列表中查询到该可飞行道具的道具 ID，可以确定该可飞行道具满足吸附条件，将该可飞行道具确定为一个第二可飞行道具，如果在满足吸附条件的道具 ID 列表中没有查询到该可飞行道具的道具 ID，可以确定该可飞行道具不满足吸附条件，即，该可飞行道具不支持变换碰撞到的其他可飞行道具的飞行轨迹，则可以执行其他的游戏主业务逻辑，判断是不做任何处理，还是抵消或者吸收掉一方的可飞行道具。

图 19 是本申请实施例提供的一种筛选第二可飞行道具的原理图，如图 19 所示，以第一可飞行道具为第一发射物、第二可飞行道具为第二发射物为例进行说明，在第一发射物和第二发射物各自的道具模型上分别挂载一个碰撞检测框。在第一发射物沿着第一飞行轨迹进行移动的过程中，一旦第一发射物的碰撞检测框与任一碰撞检测框存在交集，说明第一发射物碰撞到一个障碍物，接着，判断该障碍物的类型是否为可飞行道具，如果是，继续判断可飞行道具是否属于第二阵营，如果是，继续判断可飞行道具是否满足吸附条件，如果是，则将该可飞行道具确定为第二可飞行道具；否则，如果任一判断逻辑的判断结果为否，则退出流程。

在一些实施例中，为了方便判断任一可飞行道具是否满足吸附条件，可以在创建每个可飞行道具的配置文件时，配置文件中通常还可用于记录可飞行道具的属性信息，并在属性信息中可以记录可飞行道具的道具名称、道具 ID、所属的虚拟技能的技能 ID、所属的虚拟对象的对象 ID、所属的阵营 ID 等，接着，在属性信息中新建一个吸附属性字段，吸附属性字段取值为 True 时，代表可飞行道具满足吸附条件，吸附属性字段取值为 False 时，代表可飞行道具不满足吸附条件，同时还可以由技术人员设定吸附属性字段缺省时默认满足吸附条件或者默认不满足吸附条件。这样，在第一可飞行道具的目标范围内碰撞到障碍物以后，只需要访问障碍物的配置文件中的属性信息，即可根据障碍物的 ID 判断该障碍物是否是可飞行道具，根据阵营 ID 判断可飞行道具是否属于第二阵营，并根据吸附属性字段来判断是否满足吸附条件，在同时满足对象类型是可飞行道具、属于第二阵营且满足吸附条件的情况下，确定在第一可飞行道具的目标范围内检测到了第二可飞行道具，进入步骤 1704。

在一个示例中，终端通过以下代码来筛选得到第二可飞行道具，以下代码的逻辑即为上述确定第一可飞行道具的目标范围内碰撞到障碍物是否为第二可飞行道具的逻辑。

```
bool SpecialBulletDuration::FilterTargetBullet(const PooledHandle<SGCore::ActorRoot>& hitActor)
{
    const PooledHandle<SGCore::BaseBulletController>& hitBulletController = hitActor->BulletController;
    if (AGEProp(bOnlyCheckCollison))
    {
        return(Intersects(hitActor, _coordShape, AGEProp(bEdgeCheck))&& SGCore::BulletHelper::ContainsTag(hitActor, (BulletTag)AGEProp(targetTag)));
    }
    if (hitBulletController
        && hitBulletController->m_belongedBulletSkill
```

```

        && !hitBulletController->GetAbilityFlag(SGCore::BulletAbility_IgnoreSpecialBullet)
        && ((hitBulletController->GetAbilityFlag(SGCore::BulletAbility_CanDestroyed)) || (AGEProp(bAbsorbBullet) && hitBulletController->GetAbilityFlag(SGCore::BulletAbility_ForceDestroyedByAbsorbBullet)))
        && ((AGEProp(targetBulletType) == TargetBulletType_SelfBullet
            && (hitBulletController->m_belongBulletSkill->skillContext
            && hitBulletController->m_belongBulletSkill->skillContext->Instigator
            && hitBulletController->m_belongBulletSkill->skillContext->Instigator == attackerActor))
            || (AGEProp(targetBulletType) == TargetBulletType_EnemyBullet && hitActor->IsEnemyActor(attackerActor))
            || (AGEProp(targetBulletType) == TargetBulletType_NotThis && hitActor != triggerActor))
        && (AGEProp(bulletTypeId) <= 0 || hitBulletController->m_belongedBulletSkill->bulletParam->bulletSkillTypeId == AGEProp(bulletTypeId))
        && Intersects(hitActor, _coordShape, AGEProp(bEdgeCheck))
        && !bDestroyedSelf
        && !hitBulletController->m_belongedBulletSkill->IsFinish()
        && (AGEProp(bDestroyedTarget) || collidedBulletList.Find(hitActor->m_ObjID) == INVALID_INDEX))
    {
        if (AGEProp(bAbsorbBullet) && hitBulletController->GetAbilityFlag(SGCore::BulletAbility_NotDestroyByAbsorbBullet))
        {
            return false;
        }
        return false;
    }
    return false;
}

```

在上述步骤 1713~1733 中，设定了多重的判断逻辑来检测目标范围内的障碍物是否为第二可飞行道具，可选地，根据游戏业务需求，可以针对不同虚拟技能释放的不同第一可飞行道具设置相同或者不同的判断逻辑，例如，某类第一可飞行道具不判断是否满足吸附条件，这样能够带来更加灵活多变的对抗策略和玩法体验，有利于丰富基于可飞行道具的互动方式。此外，通过步骤 1713~1733 的多重判断逻辑，能够保证第二可飞行道具是属于第二阵营且满足吸附条件的可飞行道具，这样能够避免在碰撞到某些友方的虚拟对象释放的可飞行道具被改变飞行轨迹后，引发后续游戏业务逻辑出错。

在一些实施例中，在通过上述步骤 1713~1733，检测到第一可飞行道具的目标范围内存在第二可飞行道具，即，第一可飞行道具的碰撞检测框与第二可飞行道具的碰撞检测框存在交集以后，需要控制第一可飞行道具在第一飞行轨迹上停止移动，此外，还可以立即停止第一可飞行道具的碰撞检测功能和道具移动功能，这样防止第一可飞行道具在第一飞行轨迹上继续移动并造成伤害。

在一个示例中，以第一可飞行道具为第一发射物为例，通过第一发射物的 Actor 对象获取到第一发射物正在执行的 Action（行动），然后在第一发射物的配置文件中逐个轨道进行分析，以检测当前的 Action 所属的轨道 Type 类型，接着中止当前的 Action 所属的目标轨道。

例如，假设当前的 Action 所属的轨道是道具移动功能的轨道，则可以通过以下代码逻辑来控制第一发射物停止移动：

```

for (uint32 i = 0; i < bulletTracks.Count(); i++)
{
    const PooledHandle<AGE::Track>& bulletTrack = bulletTracks[i];
}

```

```

    if (bulletTrack && bulletTrack->enabled
        &&(bulletTrack->TrackPrefix GetTrackEventType() == AGE::EventType_MoveBullet
Duration
        || bulletTrack->TrackPrefix GetTrackEventType() == AGE::EventType_BulletTriggerD
uration
        || bulletTrack->TrackPrefix GetTrackEventType() == AGE::EventType_BezierBulletD
uration
        || bulletTrack->TrackPrefix GetTrackEventType() == AGE::EventType_Trigonometric
BulletDuration))
    {
        bulletTrack->Stop();
        bulletTrack->SetEnabled(false);
    }
}

```

1704、终端为该第一可飞行道具生成第二飞行轨迹，控制该第一可飞行道具沿该第二飞行轨迹进行移动。

在一些实施例中，停止第一可飞行道具在第一飞行轨迹的移动以后，可以使用一个新的第二飞行轨迹将第一可飞行道具重新发射出去，这样能够实现对第一可飞行道具的飞行轨迹变换，该第二飞行轨迹与第一飞行轨迹不同。

在一些实施例中，终端可以先将第一可飞行道具在第一飞行轨迹上的碰撞检测功能和道具移动功能的业务逻辑停止，再重新赋予该第一可飞行道具新的业务逻辑，这样能够避免去修改第一可飞行道具的固有配置文件，以避免破坏掉第一可飞行道具的固有业务逻辑，避免后续用户再次释放虚拟技能召唤第一可飞行道具时的表现不可控。有鉴于此，为了改变第一可飞行道具的飞行轨迹，可以为第一可飞行道具重新创建一个第二实体，再将第二实体与第一可飞行道具的第一实体进行结合，该第一实体为创建第一可飞行道具时的初始实体，该第二实体为第一可飞行道具与第二可飞行道具碰撞后创建的新实体。

在一些实施例中，终端可以先为该第一可飞行道具创建携带第二特效的第二实体，该第二实体继承该第一可飞行道具的第一实体，第二特效继承第一实体携带的第一特效。即，如上述步骤 1701 中介绍的，在召唤第一可飞行道具时，已经对第一可飞行道具创建过一个实体并添加特效了，该第一实体即为创建的初始实体，该第一特效即为添加的初始特效。因此，在对第一可飞行道具重新创建第二实体和添加第二特效时，可以跳过配置文件中的创建实体、添加特效的两步，通过继承携带第一特效的第一实体，来得到携带第二特效的第二实体，接着，再创建第一可飞行道具在第二飞行轨迹上的道具移动功能。

在一些实施例中，在创建第二实体后，终端可以基于该第一可飞行道具和该第二可飞行道具的碰撞位置，创建该第二实体的该第二飞行轨迹，例如，通过 Loop（循环）函数来设定第二飞行轨迹是将第一虚拟对象包围在内的环形轨迹，这样能够控制第一可飞行道具沿着第二飞行轨迹进行移动时，实际上执行的是围绕第一虚拟对象的绕身运动，环形轨迹的圆心、半径等均由 Loop 函数来进行定义和调整。

在一个示例中，为第一可飞行道具创建第二实体的配置文件如下：

- 1 | 获取第一可飞行道具碰撞到的第二可飞行道具
- 2 | LoopBulletDuration0 //设定第二实体执行围绕第一虚拟对象的绕身运动
- 3 | ExtendBulletDuration0 //延长第二实体的生效时长

在上述过程中，通过对第一可飞行道具重新创建第二实体，并通过第二实体来生成第二飞行轨迹，这样能够通过一个全新的配置文件来控制第一可飞行道具的移动，由于第二实体和第二特效继承了第一实体和第一特效，因此简化了全新的配置文件的创建流程，节约了终端的处理资源。

此外，上述全新的配置文件中的最后一个轨道“3 | ExtendBulletDuration0”，能够控制第一可飞行道具在改变飞行轨迹以后的生效时长，能够按照业务需求灵活地延续生效时长（甚

至可以无限延续), 或者销毁轨道 (即销毁第一可飞行道具)。

在上述步骤 1703-1704 中, 提供了在该第一可飞行道具的目标范围内检测到第二可飞行道具的情况下, 控制该第一可飞行道具沿第二飞行轨迹进行移动的一种可能实施方式, 通过停止第一可飞行道具在第一飞行轨迹上的移动, 并创建一个携带第二特效的第二实体来继承携带第一特效的第一实体, 并为第二实体生成第二飞行轨迹, 并通过第二实体的配置文件来控制第一可飞行道具在第二飞行轨迹上移动, 能够将第一可飞行道具的飞行轨迹从第一飞行轨迹切换到第二飞行轨迹, 相当于在第二飞行轨迹上将第一可飞行道具重新发射 (或称为释放), 这样能够避免破坏第一可飞行道具的固有配置文件, 此外, 通过第二实体的配置文件还能够灵活修改第一可飞行道具的生效时长, 具有很高的开发自由度。

在一些实施例, 当虚拟技能一次性释放了多个第一可飞行道具时, 可以对每个第一可飞行道具都执行以上创建第二实体、生成第二飞行轨迹的操作, 但为了便于统一管理同一虚拟技能释放的多个第一可飞行道具, 保证将同一虚拟技能释放的多个第一可飞行道具同时变换飞行轨迹, 终端还可以维护一个可飞行道具管理数组, 该可飞行道具管理数组用于管理该虚拟场景中变换过飞行轨迹的可飞行道具。例如, 终端在游戏引擎的蓝图工具中维护上述可飞行道具管理数组, 或者, 也可以使用其他开发工具来维护该可飞行道具数组。

在上述基础上, 终端在创建该第二实体后, 可以向该可飞行道具管理数组通知第一可飞行道具的轨迹变换事件, 将该第一可飞行道具添加到该可飞行道具管理数组中。这样, 每当一个第一可飞行道具的第二飞行轨迹生成完毕后, 就将该第一可飞行道具的轨迹变换事件通知到该可飞行道具管理数组, 并在该可飞行道具管理数组中添加该第一可飞行道具, 这样方便了管理同一虚拟技能一次性释放的多个第一可飞行道具, 保证这些第一可飞行道具同时变换飞行轨迹、延长生效时长以及被销毁。

一个示例中, 为了方便可飞行道具管理数组进行管理, 第二实体的配置文件可以被修改成如下的轨道逻辑:

- 1 | 获取第一可飞行道具碰撞到的第二可飞行道具
- 2 | 生成第二飞行轨迹
- 3 | 生成第二实体 (蓝图工具使用)
- 4 | 通知蓝图, 有一个第一可飞行道具切换飞行轨迹
- 5 | 第一可飞行道具沿着第二飞行轨迹重新发射
- 6 | 延长第一可飞行道具的 AGE 生效时长
- 7 | PrintTick0

如图 20 所示, 图 20 是本申请实施例提供的一种蓝图工具的示意图, 在蓝图工具中通过 ActorRoot 数组添加功能, 来创建一个目标数组即可飞行道具管理数组, 并在可飞行道具管理数组中添加一个已经变换了飞行轨迹以后的第一可飞行道具 Bullet。

在上述方式中, 通过可飞行道具管理数组, 能够方便地管理一场游戏对局中各个虚拟对象在释放虚拟技能以后召唤并变换过飞行轨迹的可飞行道具, 这样, 在可飞行道具变换了飞行轨迹以后, 不管是继续延迟可飞行道具的生效时长, 还是统一销毁这些可飞行道具, 都只需要由游戏主业务逻辑通过一个事件来通知蓝图工具中的可飞行道具管理数组, 就能够对每个变换过飞行轨迹的可飞行道具进行统一管控, 具有很高的灵活性和自由度, 并且能够按照游戏业务需求进行灵活修改。

在一些实施例, 在第一可飞行道具的目标范围内检测到所述第二可飞行道具的情况下, 控制第一可飞行道具沿第二飞行轨迹进行移动, 且取消显示第二可飞行道具。其中, 取消显示第二飞行道具表示该第二飞行道具被抵消、摧毁或者说被吸附。也即是, 在两个飞行道具发生碰撞时, 一个飞行道具改变飞行轨迹, 另一个飞行道具被抵消、摧毁或者说被吸附。

在另一些实施例, 在对第一可飞行道具生成第二飞行轨迹以后, 由于第一可飞行道具是在和第二可飞行道具碰撞以后才改变了飞行轨迹, 这时可以根据虚拟技能的技能设定, 或者对比第一虚拟对象和第二虚拟对象的技能等级, 来判断哪一方的可飞行道具会被改变飞行

轨迹, 剩余一方的可飞行道具则可以被抵消, 或者被摧毁, 或者被第一可飞行道具吸附 (或吸收), 或者继续沿着原本的固定轨迹进行移动等, 可由技术人员按照游戏业务需求进行灵活设定, 在本申请实施例, 将以第一可飞行道具改变飞行轨迹为例进行说明。

在第一可飞行道具改变飞行轨迹, 且第二可飞行道具被第一可飞行道具吸收的情况下, 终端还可以基于该第二可飞行道具, 改变该第一可飞行道具的形态, 该形态包括该第一可飞行道具的实体形状或飞行特效中的至少一项。即, 由于第二可飞行道具被第一可飞行道具吸收, 这样可以根据第二可飞行道具, 来改变第一可飞行道具的形态, 以使得改变形态后的第一可飞行道具具有部分第二可飞行道具的实体形状或飞行特效, 或者两者的合成形状或者合成特效, 接着, 终端控制改变形态后的该第一飞行道具沿该第二飞行轨迹进行移动。这样能够一目了然地向用户提示已经有两个可飞行道具在碰撞之后发生融合, 以提升用户的信息获取效率。

在一些实施例中, 由于形态包括实体形状和飞行特效两个方面, 因此, 终端可以通过以下方式 1714~1734 中的任一项, 来改变第一可飞行道具的形态:

1714、终端保留该第一可飞行道具的实体形状, 将该第二可飞行道具的飞行特效赋值给该第一可飞行道具。

在一些实施例中, 终端在创建第二实体以后, 令第二实体继承第一可飞行道具的实体形状以及第二可飞行道具的飞行特效, 这样能够使得第二实体保留第一可飞行道具的实体形状并兼具第二可飞行道具的飞行特效, 达到较为融洽自然的融合效果, 并且能够直观体现出来第二可飞行道具被第一可飞行道具吸附。

1724、终端保留该第一可飞行道具的飞行特效, 将该第二可飞行道具的实体形状赋值给该第一可飞行道具。

在一些实施例中, 终端在创建第二实体以后, 令第二实体继承第二可飞行道具的实体形状以及第一可飞行道具的飞行特效, 这样能够使得第二实体保留第二可飞行道具的实体形状并兼具第一可飞行道具的飞行特效, 达到较为融洽自然的融合效果, 同样能够直观体现出来第二可飞行道具被第一可飞行道具吸附。

1734、终端基于该第一可飞行道具的形态和该第二可飞行道具的形态确定目标形态, 将目标形态赋值给第一可飞行道具。

在一些实施例中, 终端在创建第二实体以后, 对第一可飞行道具的实体形状和第二可飞行道具的实体形状进行融合, 得到第二实体的实体形状, 例如, 使用第一可飞行道具的实体外形和第二可飞行道具的实体配色, 组合形成第二实体的实体形状, 或者, 使用第二可飞行道具的实体外形和第一可飞行道具的实体配色, 组合形成第二实体的实体形状, 或者, 保留第一可飞行道具的实体外形, 再通过一个训练完毕的图像融合模型, 来融合第一可飞行道具和第二可飞行道具各自的纹理贴图, 得到一张新的目标纹理贴图, 通过目标纹理贴图来渲染第一可飞行道具的道具模型, 目标纹理贴图可以融合两者纹理贴图的纹理特征或者风格特征等, 这里对图像融合模型不进行具体限定。

在一些实施例中, 终端在创建第二实体以后, 对第一可飞行道具的飞行特效和第二可飞行道具的飞行特效进行融合, 得到第二实体的飞行特效, 比如, 对两个可飞行道具的飞行特效进行简单叠加, 或者, 如果涉及到虚拟技能释放了多个第一可飞行道具, 可以部分第一可飞行道具保留自身飞行特效, 另一部分第一可飞行道具采用第二可飞行道具的飞行特效, 本申请实施例对此不进行具体限定。

在以上方式中, 通过基于第二可飞行道具, 来改变第一可飞行道具的形态, 这样可以改变第一可飞行道具的实体形状或者飞行特效中的至少一项, 以使得两个可飞行道具达到较为融洽自然的融合效果, 同样能够直观体现出来第二可飞行道具被第一可飞行道具吸附, 提升了用户的信息获取效率。

1705、终端延长该第一可飞行道具的生效时长。

在一些实施例中, 终端可以通过如上述步骤 1704 中介绍过的第二实体的配置文件中的轨

道“3 | ExtendBulletDuration0”，来延长第一可飞行道具的生效时长。对生效时长的延长可由技术人员进行设定，例如，每当碰撞到一个第二可飞行道具，都将生效时长延长 5 秒或者其他数值，或者，仅首次碰撞到第二可飞行道具时，将生效时长延长 10 秒或者其他数值，或者，还可以将第一可飞行道具的生效时长不断延长，并在对局中通过主业务逻辑来进行统一销毁，来满足灵活多变的业务需求。

在上述步骤 1705 中，在该第一可飞行道具的目标范围内检测到第二可飞行道具的情况下，延长该第一可飞行道具的生效时长，这样能够使得第一可飞行道具在 MOBA 游戏中的表现时长得以大幅提升，从而从延长生效时长的角度来提高了道具资源利用率。

上述所有可选技术方案，能够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。

本申请实施例提供的技术方案，通过在第一可飞行道具的目标范围内检测到第二可飞行道具时，由于第一可飞行道具和第二可飞行道具的阵营不同，控制第一可飞行道具从第一飞行轨迹切换到第二飞行轨迹，这使得第一可飞行道具的飞行轨迹随着游戏对局的进行而富于变化，提升了对第一可飞行道具的道具资源利用率，丰富了基于可飞行道具的交互方式，提高了人机交互效率。

在上一实施例中，详细介绍了两个不同阵营的可飞行道具碰撞后的处理逻辑，而在本实施例中，将以两个不同阵营的可飞行道具是两个不同阵营的发射物为例进行说明，将第一阵营中第一虚拟对象释放的第一可飞行道具称为第一发射物，将第二阵营中第二虚拟对象释放的第二可飞行道具称为第二发射物。

如图 21 所示，图 21 是本申请实施例提供的一种基于可飞行道具的互动方法的原理性示意图，该方法由电子设备执行，以电子设备为终端为例进行说明，该实施例包括以下步骤：

步骤 2101、用户控制第一阵营的第一虚拟对象释放虚拟技能后，终端创建虚拟技能的第一发射物，开启第一发射物的吸附功能。

步骤 2102、终端为第一发射物生成碰撞检测框，用于实现碰撞检测功能。

步骤 2103、终端基于碰撞检测框，对第一发射物进行实时碰撞检测。

步骤 2104、在碰撞检测框中检测到障碍物的情况下，终端判断障碍物是否为发射物，如果是，进入步骤 2105，如果否，不做任何处理。

步骤 2105、终端判断发射物是否属于第二阵营，如果是，进入步骤 2106，如果否，不做任何处理。

步骤 2106、终端判断发射物是否满足吸附条件，如果是，进入步骤 2107，如果否，不做任何处理。

步骤 2107、终端控制第一发射物在第一飞行轨迹上停止移动。

步骤 2108、终端停止对第一发射物的碰撞检测逻辑。

步骤 2109、终端为第一发射物生成新的实体，通过新的实体重新对第一发射物进行控制。

步骤 2110、终端使用蓝图工具记录第一发射物，用于后续游戏业务流程。

步骤 2111、终端执行后续流程，如在第二飞行轨迹上重新发射第一发射物，延长第一发射物的生效时长。

本申请实施例提供的技术方案，通过针对以虚拟技能为载体的发射物，通过碰撞检测功能，在检测到两个不同阵营的发射物碰撞后，控制一方阵营的发射物被另一方阵营的发射物吸附，从而改变另一方阵营的发射物的飞行轨迹并延长生效时长，能够延长发射物的表现时长，丰富发射物的飞行轨迹，极大提升了发射物道具资源利用率，丰富基于发射物的互动方式，提高了人机交互效率。

图 22 是本申请实施例提供的一种基于可飞行道具的互动装置的结构示意图，如图 22 所示，该装置包括：

创建模块 2201, 用于响应于第一虚拟对象对虚拟技能的释放操作, 在虚拟场景中创建第一可飞行道具, 该第一可飞行道具与该虚拟技能关联, 该第一虚拟对象属于第一阵营;

第一控制模块 2202, 用于控制该第一可飞行道具沿第一飞行轨迹进行移动, 该第一飞行轨迹为基于该释放操作确定的轨迹;

第二控制模块 2203, 用于在该第一可飞行道具的目标范围内检测到第二可飞行道具的情况下, 控制该第一可飞行道具沿第二飞行轨迹进行移动, 该第二可飞行道具由第二虚拟对象释放, 该第二虚拟对象属于第二阵营。

本申请实施例提供的技术方案, 通过在第一可飞行道具的目标范围内检测到第二可飞行道具时, 由于第一可飞行道具和第二可飞行道具的阵营不同, 控制第一可飞行道具从第一飞行轨迹切换到第二飞行轨迹, 这使得第一可飞行道具的飞行轨迹随着游戏对局的进行而富于变化, 提升了对第一可飞行道具的道具资源利用率, 丰富了基于可飞行道具的交互方式, 提高了人机交互效率。

在一些实施例中, 基于图 22 的装置组成, 该装置还包括检测模块, 用于:

在该第一可飞行道具的目标范围内检测到障碍物的情况下, 获取该障碍物的类型;

在该类型为可飞行道具的情况下, 获取该可飞行道具所属的阵营;

在该阵营为该第二阵营, 且该可飞行道具满足吸附条件的情况下, 将该障碍物确定为该第二可飞行道具, 该吸附条件指示该可飞行道具能够变换其他可飞行道具的飞行轨迹。

在一些实施例中, 基于图 22 的装置组成, 该第二控制模块 2203 包括:

第一控制单元, 用于控制该第一可飞行道具在该第一飞行轨迹上停止移动;

生成单元, 用于为该第一可飞行道具生成该第二飞行轨迹;

第二控制单元, 用于控制该第一可飞行道具沿该第二飞行轨迹进行移动。

在一些实施例中, 该生成单元用于:

为该第一可飞行道具创建携带第二特效的第二实体, 该第二实体继承所述第一可飞行道具的第一实体, 所述第二特效继承所述第一实体携带的第一特效;

基于该第一可飞行道具和该第二可飞行道具的碰撞位置, 创建该第二实体的该第二飞行轨迹。

在一些实施例中, 该生成单元还用于:

向可飞行道具管理数组通知该第一可飞行道具的轨迹变换事件, 将该第一可飞行道具添加到该可飞行道具管理数组中, 该可飞行道具管理数组用于管理该虚拟场景中变换过飞行轨迹的可飞行道具。

在一些实施例中, 基于图 22 的装置组成, 该装置还包括:

延长模块, 用于在该第一可飞行道具的目标范围内检测到第二可飞行道具的情况下, 延长该第一可飞行道具的生效时长。

在一些实施例中, 该第一飞行轨迹为直线轨迹, 该第二飞行轨迹为将该第一虚拟对象包围在内的环形轨迹。

在一些实施例中, 基于图 22 的装置组成, 该装置还包括:

改变模块, 用于基于该第二可飞行道具, 改变该第一可飞行道具的形态, 该形态包括该第一可飞行道具的实体形状或飞行特效中的至少一项;

该第二控制模块 2203, 还用于控制改变形态后的该第一可飞行道具沿该第二飞行轨迹进行移动。

在一些实施例中, 该改变模块用于:

保留该第一可飞行道具的实体形状, 将该第二可飞行道具的飞行特效赋值给该第一可飞行道具; 或,

保留该第一可飞行道具的飞行特效, 将该第二可飞行道具的实体形状赋值给该第一可飞行道具; 或,

基于该第一可飞行道具的形态和该第二可飞行道具的形态确定目标形态, 将目标形态赋

值给第一可飞行道具。

在一些实施例中，该第二控制模块 2203，用于：

在第一可飞行道具的目标范围内检测到第二可飞行道具的情况下，控制第一可飞行道具沿第二飞行轨迹进行移动，且取消显示第二可飞行道具。

上述所有可选技术方案，能够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。

需要说明的是：上述实施例提供的基于可飞行道具的互动装置在进行基于可飞行道具的互动时，仅以上述各功能模块的划分进行举例说明，实际应用中，能够根据需要而将上述功能分配由不同的功能模块完成，即将电子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，上述实施例提供的基于可飞行道具的互动装置与基于可飞行道具的互动方法实施例属于同一构思，其具体实现过程详见基于可飞行道具的互动方法实施例，这里不再赘述。

图 23 是本申请实施例提供的一种终端的结构示意图，如图 23 所示，终端 2300 是电子设备的一种示例性说明。可选地，该终端 2300 的设备类型包括：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 2300 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。通常，终端 2300 包括有：处理器 2301 和存储器 2302。

可选地，处理器 2301 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。可选地，处理器 2301 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。在一些实施例中，处理器 2301 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 2301 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 2301 还包括 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。

在一些实施例中，存储器 2302 包括一个或多个计算机可读存储介质，可选地，该计算机可读存储介质是非暂态的。可选地，存储器 2302 还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 2302 中的非暂态的计算机可读存储介质用于存储至少一个程序代码，该至少一个程序代码用于被处理器 2301 所执行以实现本申请中各个实施例提供的基于可飞行道具的互动方法。

在一些实施例中，终端 2300 还可选包括有：外围设备接口 2303 和至少一个外围设备。处理器 2301、存储器 2302 和外围设备接口 2303 之间能够通过总线或信号线相连。各个外围设备能够通过总线、信号线或电路板与外围设备接口 2303 相连。具体地，外围设备包括：射频电路 2304、显示屏 2305、摄像头组件 2306、音频电路 2307 和电源 2308 中的至少一种。

外围设备接口 2303 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 2301 和存储器 2302。在一些实施例中，处理器 2301、存储器 2302 和外围设备接口 2303 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 2301、存储器 2302 和外围设备接口 2303 中的任意一个或两个在单独的芯片或电路板上实现。

射频电路 2304 用于接收和发射 RF（Radio Frequency，射频）信号，也称电磁信号。射频电路 2304 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 2304 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 2304 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。

显示屏 2305 用于显示 UI (User Interface, 用户界面)。可选地, 该 UI 包括图形、文本、图标、视频及其它们的任意组合。当显示屏 2305 是触摸显示屏时, 显示屏 2305 还具有采集在显示屏 2305 的表面或表面上方的触摸信号的能力。该触摸信号能够作为控制信号输入至处理器 2301 进行处理。

摄像头组件 2306 用于采集图像或视频。可选地, 摄像头组件 2306 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, 音频电路 2307 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 2301 进行处理, 或者输入至射频电路 2304 以实现语音通信。出于立体声采集或降噪的目的, 麦克风为多个, 分别设置在终端 2300 的不同部位。

电源 2308 用于为终端 2300 中的各个组件进行供电。可选地, 电源 2308 是交流电、直流电、一次性电池或可充电电池。当电源 2308 包括可充电电池时, 该可充电电池支持有线充电或无线充电。该可充电电池还用于支持快充技术。

本领域技术人员能够理解, 图 23 中示出的结构并不构成对终端 2300 的限定, 能够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。

图 24 是本申请实施例提供的一种电子设备的结构示意图, 该电子设备 2400 可因配置或性能不同而产生比较大的差异, 该电子设备 2400 包括一个或一个以上处理器 (Central Processing Units, CPU) 2401 和一个或一个以上的存储器 2402, 其中, 该存储器 2402 中存储有至少一条计算机程序, 该至少一条计算机程序由该一个或一个以上处理器 2401 加载并执行以实现上述各个实施例提供的基于可飞行道具的互动方法。可选地, 该电子设备 2400 还具有有线或无线网络接口、键盘以及输入输出接口等部件, 以便进行输入输出, 该电子设备 2400 还包括其他用于实现设备功能的部件, 在此不做赘述。

在示例性实施例中, 还提供了一种计算机可读存储介质, 例如包括至少一条计算机程序的存储器, 上述至少一条计算机程序可由电子设备中的处理器执行以完成上述各个实施例中的基于可飞行道具的互动方法。例如, 该计算机可读存储介质包括 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。

在示例性实施例中, 还提供了一种计算机程序产品, 包括一条或多条计算机程序, 该一条或多条计算机程序存储在计算机可读存储介质中。电子设备的一个或多个处理器能够从计算机可读存储介质中读取该一条或多条计算机程序, 该一个或多个处理器执行该一条或多条计算机程序, 使得电子设备能够执行以完成上述实施例中的基于可飞行道具的互动方法。

本领域普通技术人员能够理解实现上述实施例的全部或部分步骤能够通过硬件来完成, 也能够通过程序来指令相关的硬件完成, 可选地, 该程序存储于一种计算机可读存储介质中, 可选地, 上述提到的存储介质是只读存储器、磁盘或光盘等。

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。

权利要求书

1.一种基于可飞行道具的互动方法，由电子设备执行，所述方法包括：

响应于第一虚拟对象对虚拟技能的释放操作，在虚拟场景中创建第一可飞行道具，所述第一可飞行道具与所述虚拟技能关联，所述第一虚拟对象属于第一阵营；

控制所述第一可飞行道具沿第一飞行轨迹进行移动，所述第一飞行轨迹为基于所述释放操作确定的轨迹；

在所述第一可飞行道具的目标范围内检测到第二可飞行道具的情况下，控制所述第一可飞行道具沿第二飞行轨迹进行移动，所述第二可飞行道具由第二虚拟对象释放，所述第二虚拟对象属于第二阵营，所述第二飞行轨迹与所述第一飞行轨迹不同。

2.根据权利要求1所述的方法，其中，所述在所述第一可飞行道具的目标范围内检测到第二可飞行道具包括：

在所述第一可飞行道具的目标范围内检测到障碍物的情况下，获取所述障碍物的类型；

在所述类型为可飞行道具的情况下，获取所述可飞行道具所属的阵营；

在所述阵营为所述第二阵营，且所述可飞行道具满足吸附条件的情况下，将所述障碍物确定为所述第二可飞行道具，所述吸附条件指示所述可飞行道具能够变换其他可飞行道具的飞行轨迹。

3.根据权利要求1或2所述的方法，其中，所述控制所述第一可飞行道具沿第二飞行轨迹进行移动包括：

控制所述第一可飞行道具在所述第一飞行轨迹上停止移动；

为所述第一可飞行道具生成所述第二飞行轨迹；

控制所述第一可飞行道具沿所述第二飞行轨迹进行移动。

4.根据权利要求3所述的方法，其中，所述为所述第一可飞行道具生成所述第二飞行轨迹包括：

为所述第一可飞行道具创建携带第二特效的第二实体，所述第二实体继承所述第一可飞行道具的第一实体，所述第二特效继承所述第一实体携带的第一特效；

基于所述第一可飞行道具和所述第二可飞行道具的碰撞位置，创建所述第二实体的所述第二飞行轨迹。

5.根据权利要求4所述的方法，其中，所述方法还包括：

向可飞行道具管理数组通知所述第一可飞行道具的轨迹变换事件，将所述第一可飞行道具添加到所述可飞行道具管理数组中，所述可飞行道具管理数组用于管理所述虚拟场景中变换过飞行轨迹的可飞行道具。

6.根据权利要求1所述的方法，其中，所述方法还包括：

在所述第一可飞行道具的目标范围内检测到所述第二可飞行道具的情况下，延长所述第一可飞行道具的生效时长。

7.根据权利要求1所述的方法，其中，所述第一飞行轨迹为直线轨迹，所述第二飞行轨迹为将所述第一虚拟对象包围在内的环形轨迹。

8.根据权利要求1所述的方法，其中，所述在所述第一可飞行道具的目标范围内检测到

第二可飞行道具的情况下，控制所述第一可飞行道具沿第二飞行轨迹进行移动包括：

基于所述第二可飞行道具，改变所述第一可飞行道具的形态，所述形态包括所述第一可飞行道具的实体形状或飞行特效中的至少一项；

控制改变形态后的所述第一可飞行道具沿所述第二飞行轨迹进行移动。

9.根据权利要求8所述的方法，其中，所述基于所述第二可飞行道具，改变所述第一可飞行道具的形态包括：

保留所述第一可飞行道具的实体形状，将所述第二可飞行道具的飞行特效赋值给所述第一可飞行道具；或，

保留所述第一可飞行道具的飞行特效，将所述第二可飞行道具的实体形状赋值给所述第一可飞行道具；或，

基于所述第一可飞行道具的形态和所述第二可飞行道具的形态确定目标形态，将所述目标形态赋值给所述第一可飞行道具。

10.根据权利要求1所述的方法，其中，所述在所述第一可飞行道具的目标范围内检测到第二可飞行道具的情况下，控制所述第一可飞行道具沿第二飞行轨迹进行移动包括：

在所述第一可飞行道具的目标范围内检测到所述第二可飞行道具的情况下，控制所述第一可飞行道具沿所述第二飞行轨迹进行移动，且取消显示所述第二可飞行道具。

11.一种基于可飞行道具的互动装置，配置于电子设备中，所述装置包括：

创建模块，用于响应于第一虚拟对象对虚拟技能的释放操作，在虚拟场景中创建第一可飞行道具，所述第一可飞行道具与所述虚拟技能关联，所述第一虚拟对象属于第一阵营；

第一控制模块，用于控制所述第一可飞行道具沿第一飞行轨迹进行移动，所述第一飞行轨迹为基于所述释放操作确定的轨迹；

第二控制模块，用于在所述第一可飞行道具的目标范围内检测到第二可飞行道具的情况下，控制所述第一可飞行道具沿第二飞行轨迹进行移动，所述第二可飞行道具由第二虚拟对象释放，所述第二虚拟对象属于第二阵营，所述第二飞行轨迹与所述第一飞行轨迹不同。

12.一种电子设备，所述电子设备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计算机程序，所述至少一条计算机程序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10任一项所述的基于可飞行道具的互动方法。

13.一种计算机可读存储介质，所述计算机可读存储介质中存储有至少一条计算机程序，所述至少一条计算机程序由处理器加载并执行以实现如权利要求1至权利要求10任一项所述的基于可飞行道具的互动方法。

14.一种计算机程序产品，所述计算机程序产品包括至少一条计算机程序，所述至少一条计算机程序由处理器加载并执行以实现如权利要求1至权利要求10任一项所述的基于可飞行道具的互动方法。