



(12) 发明专利申请

(10) 申请公布号 CN 114570023 A

(43) 申请公布日 2022. 06. 03

(21) 申请号 202210129276.4

(22) 申请日 2022.02.11

(71) 申请人 深圳市锐昌智能科技有限公司
地址 518000 广东省深圳市南山区粤海街道高新区社区沙河西路1819号深圳湾科技生态园7栋B1010

(72) 发明人 冀恒 侯本贵

(74) 专利代理机构 深圳市鼎泰正和知识产权代理事务所(普通合伙) 44555
专利代理师 缪太清

(51) Int. Cl.
A63F 13/537 (2014.01)
A63F 13/577 (2014.01)
A63F 13/60 (2014.01)

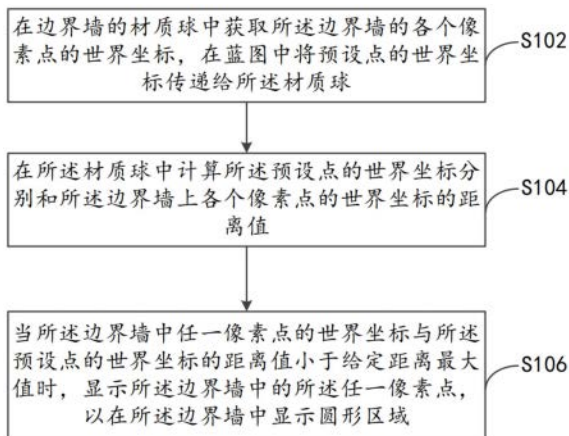
权利要求书2页 说明书7页 附图2页

(54) 发明名称

UE4引擎中虚拟智能边界墙实现方法和装置

(57) 摘要

本发明涉及一种UE4引擎中虚拟智能边界墙实现方法、装置、计算机设备和存储介质,所述方法包括:在边界墙的材质球中获取所述边界墙的各个像素点的世界坐标,在蓝图中将预设点的世界坐标传递给所述材质球;在所述材质球中计算所述预设点的世界坐标分别和所述边界墙上各个像素点的世界坐标的距离值;当所述边界墙中任一像素点的世界坐标与所述预设点的世界坐标的距离值小于给定距离最大值时,显示所述边界墙中的所述任一像素点,以在所述边界墙中显示圆形区域。上述方法能够具备更炫酷效果以对用户更有警示感的边界墙显示。



1. 一种UE4引擎中虚拟智能边界墙实现方法,其特征在于,所述方法包括:

在边界墙的材质球中获取所述边界墙的各个像素点的世界坐标,在蓝图中将预设点的世界坐标传递给所述材质球;

在所述材质球中计算所述预设点的世界坐标分别和所述边界墙上各个像素点的世界坐标的距离值;

当所述边界墙中任一像素点的世界坐标与所述预设点的世界坐标的距离值小于给定距离最大值时,显示所述边界墙中的所述任一像素点,以在所述边界墙中显示圆形区域。

2. 根据权利要求1所述的虚拟智能边界墙实现方法,其特征在于,所述当所述边界墙中任一像素点的世界坐标与所述预设点的世界坐标的距离值小于给定距离最大值时,显示所述边界墙中的所述任一像素点,包括:

将所述给定距离最大值减去所述边界墙中任一像素点的世界坐标与所述预设点的世界坐标的距离值,得到第一差值;

当所述第一差值小于零时,不显示所述边界墙中的任一像素点;

当所述第一差值大于或等于零时,显示所述边界墙中的任一像素点,以在所述边界墙中显示圆形区域。

3. 根据权利要求2所述的虚拟智能边界墙实现方法,其特征在于,所述圆形区域中包含多个像素点,所述多个像素点中与所述预设点的差值越小的像素点,像素点的亮度越大。

4. 根据权利要求2所述的虚拟智能边界墙实现方法,其特征在于,所述圆形区域中包含多个像素点,所述方法还包括:

获取所述圆形区域中的任一像素点与所述预设点的第二差值;

将所述第二差值除以所述给定最大距离值,得到商值;

将所述商值赋值给所述材质球的不透明度,以得到所述圆形区域中的任一像素点的亮度。

5. 根据权利要求1所述的虚拟智能边界墙实现方法,其特征在于,所述方法还包括:

获取所述预设点到所述边界墙的最短距离;

当所述最短距离小于预设的最大闪烁距离时,开启所述边界墙的闪烁。

6. 根据权利要求5所述的虚拟智能边界墙实现方法,其特征在于,所述开启所述边界墙的闪烁之后,还包括:

启动所述材质球的时间节点和三角函数节点;

通过所述时间节点的时间函数获得时间值;

根据所述时间值、所述最大闪烁距离以及所述最短距离获得所述三角函数节点的三角函数的周期系数;

根据所述三角函数和所述周期系数得到闪烁控制值;

根据所述闪烁控制值控制所述边界墙的闪烁。

7. 根据权利要求6所述的虚拟智能边界墙实现方法,其特征在于,所述根据所述时间值、所述最大闪烁距离以及所述最短距离获得所述三角函数节点的三角函数的周期系数,包括:

获取所述最大闪烁距离与所述最短距离的第三差值;

计算所述第三差值与所述最短距离的比值;

获取所述时间值与所述比值的乘积,所述所述时间值与所述比值的乘积为所述周期系数;

所述根据所述三角函数和所述周期系数得到闪烁控制值,包括:

将三角函数sine和所述周期系数的乘积作为所述闪烁控制值,所述三角函数为所述三角函数sine。

8. 一种UE4引擎中虚拟智能边界墙实现装置,其特征在于,所述装置包括:

获取模块,用于在边界墙的材质球中获取所述边界墙的各个像素点的世界坐标,在蓝图中将预设点的世界坐标传递给所述材质球;

计算模块,用于在所述材质球中计算所述预设点的世界坐标分别和所述边界墙上各个像素点的世界坐标的距离值;

显示模块,用于当所述边界墙中任一像素点的世界坐标与所述预设点的世界坐标的距离值小于给定距离最大值时,显示所述边界墙中的所述任一像素点,以在所述边界墙中显示圆形区域。

9. 一种计算机设备,包括存储器、处理器及存储在存储器上并可在处理器上运行的应用程序,其特征在于,所述处理器执行所述应用程序时实现权利要求1至7中任一项所述方法的步骤。

10. 一种计算机可读存储介质,其上存储有应用程序,其特征在于,所述应用程序被处理器执行时实现权利要求1至7中任一项所述方法的步骤。

UE4引擎中虚拟智能边界墙实现方法和装置

技术领域

[0001] 本发明涉及虚拟引擎技术领域,特别是涉及一种UE4引擎中虚拟智能边界墙实现方法、装置、计算机设备和存储介质。

背景技术

[0002] UE4是美国Epic游戏公司研发的一款3A级次时代游戏引擎,渲染效果强大,采用物理材质系统,是开发者最喜爱的引擎之一。UE4画面效果完全达到3A游戏水准,光照和物理渲染效果强大。UE4蓝图系统让游戏策划也能编辑代码,各种官方插件齐全也让开发者不用在自编第三方插件并担心兼容接口问题。更重要的是针对虚拟现实游戏,UE4为手柄、VR控制器提供了良好支持。

[0003] 一些VR应用中常常会有墙、悬崖之类的边界,用户走到边界附近时,需要有一些视觉或者听觉提示提醒用户已接近边界区域,不能继续往前走了。现有的VR应用会在用户走到边界区域时显示一面墙,警醒用户,但是这样效果比较单一。

发明内容

[0004] 本发明的目的在于克服现有技术的不足,提供了一种UE4引擎中虚拟智能边界墙实现方法、装置、计算机设备和存储介质,能够具备更炫酷效果以对用户更有警示感的边界墙显示。

[0005] 为了解决上述中至少一个技术问题,本发明实施例提供了一种UE4引擎中虚拟智能边界墙实现方法,所述方法包括:

[0006] 在边界墙的材质球中获取所述边界墙的各个像素点的世界坐标,在蓝图中将预设点的世界坐标传递给所述材质球;

[0007] 在所述材质球中计算所述预设点的世界坐标分别和所述边界墙上各个像素点的世界坐标的距离值;

[0008] 当所述边界墙中任一像素点的世界坐标与所述预设点的世界坐标的距离值小于给定距离最大值时,显示所述边界墙中的所述任一像素点,以在所述边界墙中显示圆形区域。

[0009] 在一个实施例中,所述当所述边界墙中任一像素点的世界坐标与所述预设点的世界坐标的距离值小于给定距离最大值时,显示所述边界墙中的所述任一像素点,包括:

[0010] 将所述给定距离最大值减去所述边界墙中任一像素点的世界坐标与所述预设点的世界坐标的距离值,得到第一差值;

[0011] 当所述第一差值小于零时,不显示所述边界墙中的任一像素点;

[0012] 当所述第一差值大于或等于零时,显示所述边界墙中的任一像素点,以在所述边界墙中显示圆形区域。

[0013] 在一个实施例中,所述圆形区域中包含多个像素点,所述多个像素点中与所述预设点的差值越小的像素点,像素点的亮度越大。

- [0014] 在一个实施例中,所述圆形区域中包含多个像素点,所述方法还包括:
- [0015] 获取所述圆形区域中的任一像素点与所述预设点的第二差值;
- [0016] 将所述第二差值除以所述给定最大距离值,得到商值;
- [0017] 将所述商值赋值给所述材质球的不透明度,以得到所述圆形区域中的任一像素点的亮度。
- [0018] 在一个实施例中,所述方法还包括:
- [0019] 获取所述预设点到所述边界墙的最短距离;
- [0020] 当所述最短距离小于预设的最大闪烁距离时,开启所述边界墙的闪烁。
- [0021] 在一个实施例中,所述开启所述边界墙的闪烁之后,还包括:
- [0022] 启动所述材质球的时间节点和三角函数节点;
- [0023] 通过所述时间节点的时间函数获得时间值;
- [0024] 根据所述时间值、所述最大闪烁距离以及所述最短距离获得所述三角函数节点的三角函数的周期系数;
- [0025] 根据所述三角函数和所述周期系数得到闪烁控制值;
- [0026] 根据所述闪烁控制值控制所述边界墙的闪烁。
- [0027] 在一个实施例中,所述根据所述时间值、所述最大闪烁距离以及所述最短距离获得所述三角函数节点的三角函数的周期系数,包括:
- [0028] 获取所述最大闪烁距离与所述最短距离的第三差值;
- [0029] 计算所述第三差值与所述最短距离的比值;
- [0030] 获取所述时间值与所述比值的乘积,所述所述时间值与所述比值的乘积为所述周期系数;
- [0031] 所述根据所述三角函数和所述周期系数得到闪烁控制值,包括:
- [0032] 将三角函数sine和所述周期系数的乘积作为所述闪烁控制值,所述三角函数为所述三角函数sine。
- [0033] 一种UE4引擎中虚拟智能边界墙实现装置,所述装置包括:
- [0034] 获取模块,用于在边界墙的材质球中获取所述边界墙的各个像素点的世界坐标,在蓝图中将预设点的世界坐标传递给所述材质球;
- [0035] 计算模块,用于在所述材质球中计算所述预设点的世界坐标分别和所述边界墙上各个像素点的世界坐标的距离值;
- [0036] 显示模块,用于当所述边界墙中任一像素点的世界坐标与所述预设点的世界坐标的距离值小于给定距离最大值时,显示所述边界墙中的所述任一像素点,以在所述边界墙中显示圆形区域。
- [0037] 另外,本发明实施例还提供了一种计算机设备,其包括:存储器、处理器及存储在存储器上并可在处理器上运行的应用程序,处理器执行应用程序时实现上述任一实施例方法的步骤。
- [0038] 另外,本发明实施例还提供了一种计算机可读存储介质,其上存储有应用程序,应用程序被处理器执行时实现上述任一实施例方法的步骤。
- [0039] 在本发明实施例中,在边界墙的材质球中获取所述边界墙的各个像素点的世界坐标,在蓝图中将预设点的世界坐标传递给所述材质球;在所述材质球中计算所述预设点的

世界坐标分别和所述边界墙上各个像素点的世界坐标的距离值；当所述边界墙中任一像素点的世界坐标与所述预设点的世界坐标的距离值小于给定距离最大值时，显示所述边界墙中的所述任一像素点，以在所述边界墙中显示圆形区域。因此，能够在边界墙中显示出圆形区域，而并不是显示全部的一面边界墙，从而能够具备更炫酷效果以对用户更有警示感的边界墙显示。

附图说明

- [0040] 图1是本发明实施例中的一种UE4引擎中虚拟智能边界墙实现方法的流程示意图；
[0041] 图2是本发明实施例中的一种UE4引擎中虚拟智能边界墙实现方法的边界墙显示效果的示意图；
[0042] 图3是本发明实施例中的一种UE4引擎中虚拟智能边界墙实现装置的结构示意图；
[0043] 图4是本发明实施例中的计算机设备的结构组成示意图。

具体实施方式

[0044] 下面将结合本发明实施例中的附图，对本发明实施例中的技术方案进行清楚、完整地描述，显然，所描述的实施例仅仅是本发明一部分实施例，而不是全部的实施例。基于本发明中的实施例，本领域普通技术人员在没有作出创造性劳动前提下所获得的所有其它实施例，都属于本发明保护的范围。

[0045] 本发明实施例提供了一种UE4引擎中虚拟智能边界墙实现方法，如图1所示，该方法包括以下步骤：

[0046] S102，在边界墙的材质球中获取所述边界墙的各个像素点的世界坐标，在蓝图中将预设点的世界坐标传递给所述材质球。

[0047] 具体地，在边界墙的材质球获取到各个像素点的世界坐标，在蓝图中将预设点的世界坐标传递给该材质球。

[0048] S104，在所述材质球中计算所述预设点的世界坐标分别和所述边界墙上各个像素点的世界坐标的距离值。

[0049] 具体地，声明一个float4类型的变量，命名为TriggerPoint，作为触发边界墙预设点的位置。在材质球中获取到各个像素点的世界坐标，进而减去TriggerPoint的世界坐标，计算出所得各个向量的大小，即为各个像素点到预设点的距离值。

[0050] S106，当所述边界墙中任一像素点的世界坐标与所述预设点的世界坐标的距离值小于给定距离最大值时，显示所述边界墙中的所述任一像素点，以在所述边界墙中显示圆形区域。

[0051] 具体地，使用距离触发边界墙的显示。当预设点靠近边界墙到最大显示距离时，显示边界墙的一个圆形区域的像素点，圆形区域的圆心位置是预设点的坐标投影在边界墙的位置。实现的方式为：参见图2所示，计算各个像素点与预设点之间的距离值，当任一距离值小于给定距离最大值时，则显示该距离值对应的像素点，从而实现在边界墙显示圆形区域。

[0052] 在一个实施例中，所述当所述边界墙中任一像素点的世界坐标与所述预设点的世界坐标的距离值小于给定距离最大值时，显示所述边界墙中的所述任一像素点，包括：将所述给定距离最大值减去所述边界墙中任一像素点的世界坐标与所述预设点的世界坐标的

距离值,得到第一差值;当所述第一差值小于零时,不显示所述边界墙中的任一像素点;当所述第一差值大于或等于零时,显示所述边界墙中的任一像素点,以在所述边界墙中显示圆形区域。

[0053] 具体地,计算各个像素点与预设点之间的距离值,给定距离最大值减去任一距离值,结果小于0的返回0,不显示这个任一距离值对应的像素点。反之,结果大于0时,则显示这个任一距离值对应的像素点。

[0054] 在一个实施例中,所述圆形区域中包含多个像素点,所述多个像素点中与所述预设点的差值越小的像素点,像素点的亮度越大。

[0055] 具体地,圆形区域中预设点投影到圆形区域的位置为圆形区域的圆心。圆形区域内包含多个像素点,圆形区域中像素点与预设点的差值越小,该像素点的亮度越大。也即是,圆形区域的圆心的像素点与预设点的差值最小,圆心的像素点最亮。越远离圆心的像素点,其与预设点的差值越大,对应的像素点亮度越小,即越靠近圆形区域的边缘的像素点的亮度越小,对应越暗。如上述例子并结合图2所示,边界墙中离圆心越近的像素点越亮,离圆心越远的像素点越暗,且预设点越靠近边界墙,圆形区域的像素点越亮。

[0056] 在一个实施例中,所述圆形区域中包含多个像素点,S106之后还包括:获取所述圆形区域中的任一像素点与所述预设点的第二差值;将所述第二差值除以所述给定最大距离值,得到商值;将所述商值赋值给所述材质球的不透明度,以得到所述圆形区域中的任一像素点的亮度。

[0057] 具体地,计算各个像素点与预设点之间的距离值,给定距离最大值减去任一距离值再除以给定距离最大值,得到结果值,结果值处于(0,1)之间。将结果值赋值给材质球的不透明度,会使越靠近圆心的像素点的Alpha值越高,距离越接近给定距离最大值的像素点的Alpha值越接近0,计算出来的值越接近0,则像素点的颜色越虚。

[0058] 如,声明一个float类型的变量,命名为TriggerToWallDistance,作为触发边界墙预设点到边界墙所在平面的最短距离;声明一个float类型的变量,命名为MaxDistance,作为显示边界墙最远距离。

[0059] 在材质球中获取到各个像素点的世界坐标,进而减去TriggerPoint的世界坐标,计算出所得各个向量的大小,即为各个像素点到预设点的距离值。再用MaxDistance减去各个像素点到预设点的距离值,将结果使用clamp材质节点换算,使该结果小于0的时候为0,大于MaxDistance的时候为MaxDistance,再将该结果乘以(MaxDistance/各个像素点到预设点的距离值)的n次方,最后得到记过Alpha1,n的取值可根据实际效果来调整,使显示的区域明暗程度更加明显。

[0060] 在一个实施例中,S106之后还包括:获取所述预设点到所述边界墙的最短距离;当所述最短距离小于预设的最大闪烁距离时,开启所述边界墙的闪烁。

[0061] 优选地,所述开启所述边界墙的闪烁之后,还包括:启动所述材质球的时间节点和三角函数节点;通过所述时间节点的时间函数获得时间值;根据所述时间值、所述最大闪烁距离以及所述最短距离获得所述三角函数节点的三角函数的周期系数;根据所述三角函数和所述周期系数得到闪烁控制值;根据所述闪烁控制值控制所述边界墙的闪烁。

[0062] 优选地,所述根据所述时间值、所述最大闪烁距离以及所述最短距离获得所述三角函数节点的三角函数的周期系数,包括:获取所述最大闪烁距离与所述最短距离的第三

差值;计算所述第三差值与所述最短距离的比值;获取所述时间值与所述比值的乘积,所述所述时间值与所述比值的乘积为所述周期系数;所述根据所述三角函数和所述周期系数得到闪烁控制值,包括:将三角函数sine和所述周期系数的乘积作为所述闪烁控制值,所述三角函数为所述三角函数sine。

[0063] 具体地,获取预设点到边界墙的最短距离,当该最短距离小于预设的最大闪烁距离时,使用材质球的Time节点和三角函数节点,开启边界墙的闪烁,并将预设点到边界墙的最短距离作为三角函数的周期系数,使距离越短闪烁越快。

[0064] 如,声明一个float类型的变量,命名为ShiningDistance,作为开始闪烁的最大闪烁距离。声明一个float类型的变量,命名为TriggerToWallDistance,作为触发预设点到边界墙所在平面的最短距离。

[0065] 在材质球中,将ShiningDistance和TriggerToWallDistance使用if节点进行比较,如果 $\text{TriggerToWallDistance} \geq \text{ShiningDistance}$,则返回1;如果 $\text{TriggerToWallDistance} < \text{ShiningDistance}$,则使用材质球中的时间函数,时间函数会传出一个随着时间进行的时间值,比如过了2秒,会传出一个2。进而时间值乘以 $(\text{ShiningDistance} - \text{TriggerToWallDistance})$ 后再除以TriggerToWallDistance,得到周期系数。周期系数再乘以sine函数,得到Alpha2,此Alpha2结果会随着时间的推移,在(-1,1)之间呈周期变化,且会随着TriggerToWallDistance越来越小,变化越来越快。因此,通过Alpha2结果可以实现在预设点靠近边界墙到最大闪烁距离时,边界墙开始闪烁,且距离越近闪烁越快,加强警示效果。

[0066] 在一个实施例中,将上述 $\text{Alpha1} * \text{Alpha2}$ 赋值给材质球的不透明度,即可让该边界墙的材质会满足效果。

[0067] 需要说明的是,在UE4引擎中实现上述各个实施例所述的一种UE4引擎中虚拟智能边界墙实现方法时,各个参数的声明参见如下处理方式:

[0068] 蓝图方面:新建一个Actor,以添加一个静态模型组件,该静态模型组件为边界墙模型。可以提供一个比较炫酷的边界墙模型,按照上面的藐视修改该边界墙模型的材料。

[0069] 使用虚幻自带的函数CreateDynamicMaterialInstance将边界墙的材料材质转化为动态材料实例,如可在蓝图中修改材料的参数。然后使用函数SetVectorParameterValue设置TriggerPoin,使用函数SetScalarParameterValue设置TriggerToWallDistance、MaxDistance和ShiningDistance即可在蓝图中自定义材料中声明的参数。

[0070] 在本发明实施例中,在边界墙的材质球中获取所述边界墙的各个像素点的世界坐标,在蓝图中将预设点的世界坐标传递给所述材质球;在所述材质球中计算所述预设点的世界坐标分别和所述边界墙上各个像素点的世界坐标的距离值;当所述边界墙中任一像素点的世界坐标与所述预设点的世界坐标的距离值小于给定距离最大值时,显示所述边界墙中的所述任一像素点,以在所述边界墙中显示圆形区域。因此,能够在边界墙中显示出圆形区域,而并不是显示全部的一面边界墙,从而能够具备更炫酷效果以对用户更有警示感的边界墙显示。

[0071] 在一实施例中,本发明还提供一种UE4引擎中虚拟智能边界墙实现装置,如图3所示,该装置包括:

[0072] 获取模块302,用于在边界墙的材质球中获取所述边界墙的各个像素点的世界坐

标,在蓝图中将预设点的世界坐标传递给所述材质球。

[0073] 计算模块304,用于在所述材质球中计算所述预设点的世界坐标分别和所述边界墙上各个像素点的世界坐标的距离值。

[0074] 显示模块306,用于当所述边界墙中任一像素点的世界坐标与所述预设点的世界坐标的距离值小于给定距离最大值时,显示所述边界墙中的所述任一像素点,以在所述边界墙中显示圆形区域。

[0075] 关于一种UE4引擎中虚拟智能边界墙实现装置的具体限定可以参见上文中对于一种UE4引擎中虚拟智能边界墙实现方法的限定,在此不再赘述。上述一种UE4引擎中虚拟智能边界墙实现装置中的各个模块可全部或部分通过软件、硬件及其组合来实现。上述各模块可以硬件形式内嵌于或独立于计算机设备中的处理器中,也可以以软件形式存储于计算机设备中的存储器中,以便于处理器调用执行以上各个模块对应的操作。

[0076] 本发明实施例提供的一种计算机可读存储介质,所述计算机可读存储介质上存储有应用程序,该程序被处理器执行时实现上述实施例中任意一个实施例的一种UE4引擎中虚拟智能边界墙实现方法。其中,所述计算机可读存储介质包括但不限于任何类型的盘(包括软盘、硬盘、光盘、CD-ROM、和磁光盘)、ROM(Read-Only Memory,只读存储器)、RAM(Random Access Memory,随机存储器)、EPROM(Erasable Programmable Read-Only Memory,可擦写可编程只读存储器)、EEPROM(Electrically Erasable Programmable Read-Only Memory,电可擦可编程只读存储器)、闪存、磁性卡片或光线卡片。也就是存储设备包括由设备(例如,计算机、手机)以能够读的形式存储或传输信息的任何介质,可以是只读存储器,磁盘或光盘等。

[0077] 本发明实施例还提供了一种计算机应用程序,其运行在计算机上,该计算机应用程序用于执行上述中任意一个实施例一种UE4引擎中虚拟智能边界墙实现方法。

[0078] 此外,图4是本发明实施例中的计算机设备的结构组成示意图。

[0079] 本发明实施例还提供了一种计算机设备,如图4所示。所述计算机设备包括处理器402、存储器403、输入单元404以及显示单元405等器件。本领域技术人员可以理解,图4示出的设备结构器件并不构成对所有设备的限定,可以包括比图示更多或更少的部件,或者组合某些部件。存储器403可用于存储应用程序401以及各功能模块,处理器402运行存储在存储器403的应用程序401,从而执行设备的各种功能应用以及数据处理。存储器可以是内存储器或外存储器,或者包括内存储器和外存储器两者。内存储器可以包括只读存储器(ROM)、可编程ROM(PROM)、电可编程ROM(EPROM)、电可擦写可编程ROM(EEPROM)、快闪存储器、或者随机存储器。外存储器可以包括硬盘、软盘、ZIP盘、U盘、磁带等。本发明所公开的存储器包括但不限于这些类型的存储器。本发明所公开的存储器只作为例子而非作为限定。

[0080] 输入单元404用于接收信号的输入,以及接收用户输入的关键字。输入单元404可包括触控面板以及其它输入设备。触控面板可收集用户在其上或附近的触摸操作(比如用户使用手指、触笔等任何适合的物体或附件在触控面板上或在触控面板附近的操作),并根据预先设定的程序驱动相应的连接装置;其它输入设备可以包括但不限于物理键盘、功能键(比如播放控制按键、开关按键等)、轨迹球、鼠标、操作杆等中的一种或多种。显示单元405可用于显示用户输入的信息或提供给用户的信息以及终端设备的各种菜单。显示单元405可采用液晶显示器、有机发光二极管等形式。处理器402是终端设备的控制中心,利用各

种接口和线路连接整个设备的各个部分,通过运行或执行存储在存储器403内的软件程序和/或模块,以及调用存储在存储器内的数据,执行各种功能和处理数据。

[0081] 作为一个实施例,所述计算机设备包括:一个或多个处理器402,存储器403,一个或多个应用程序401,其中所述一个或多个应用程序401被存储在存储器403中并被配置为由所述一个或多个处理器402执行,所述一个或多个应用程序401配置用于执行上述实施例中的任意一实施例中的一种UE4引擎中虚拟智能边界墙实现方法。

[0082] 另外,以上对本发明实施例所提供的一种UE4引擎中虚拟智能边界墙实现方法、装置、计算机设备和存储介质进行了详细介绍,本文中应采用具体个例对本发明的原理及实施方式进行了阐述,以上实施例的说明只是用于帮助理解本发明的方法及其核心思想;同时,对于本领域的一般技术人员,依据本发明的思想,在具体实施方式及应用范围上均会有改变之处,综上所述,本说明书内容不应理解为对本发明的限制。

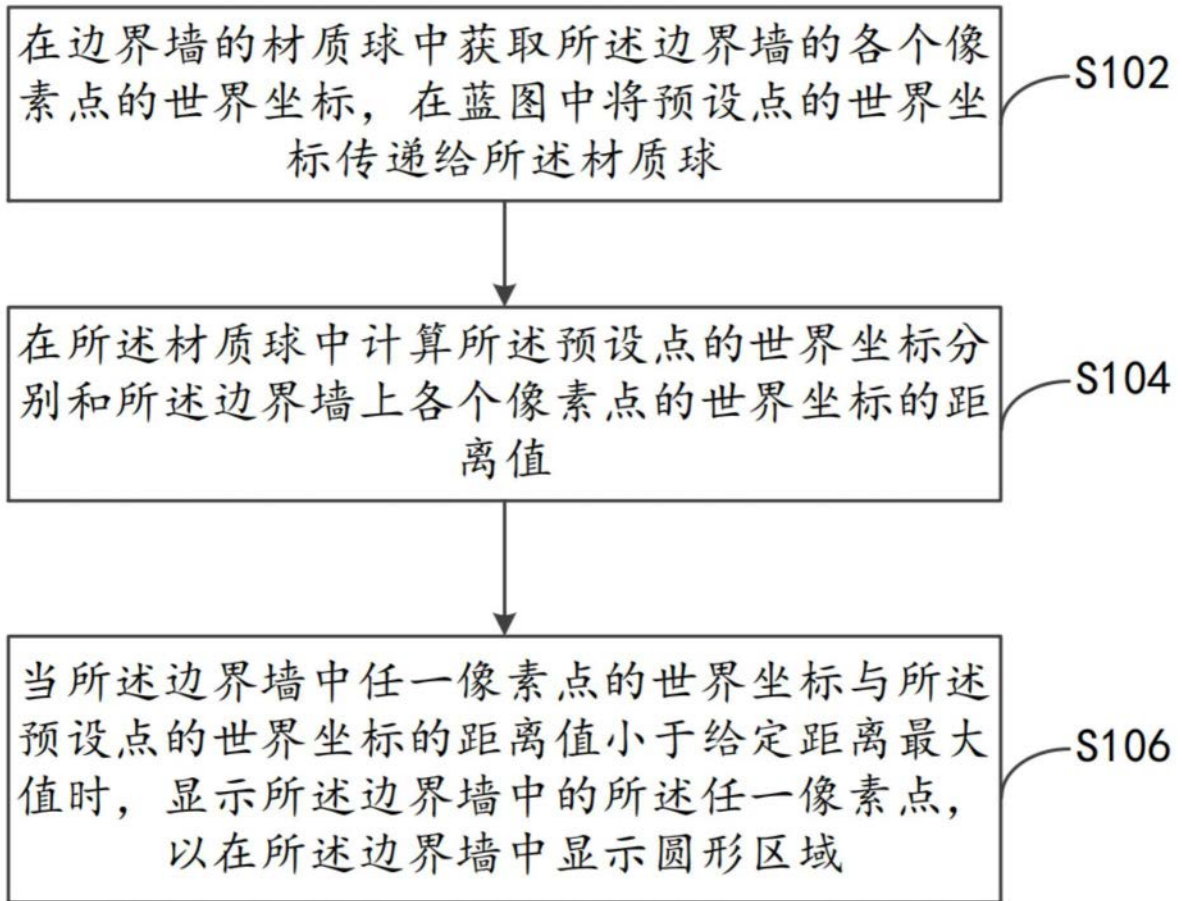


图1

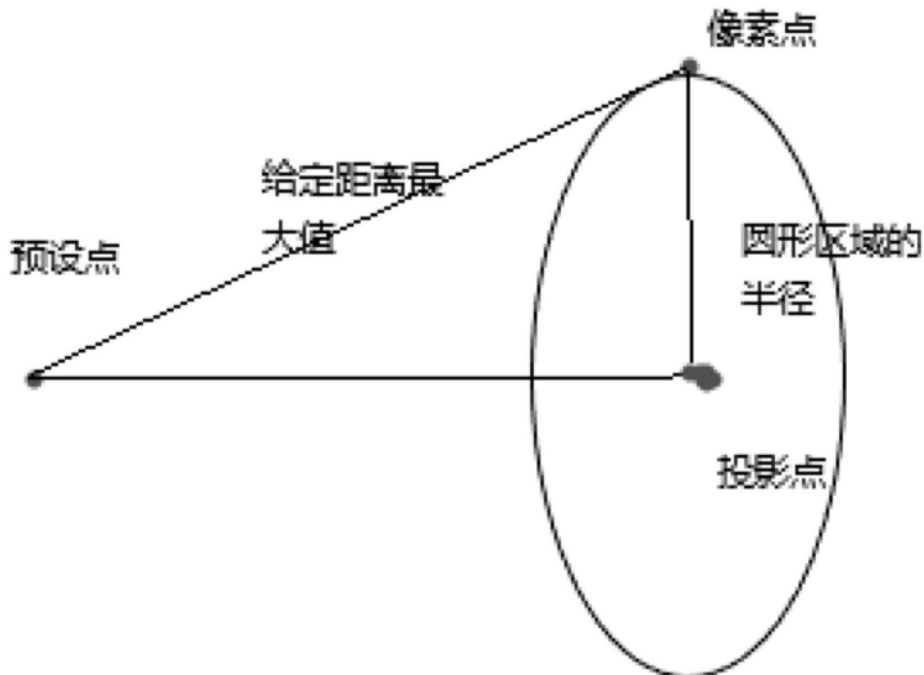


图2

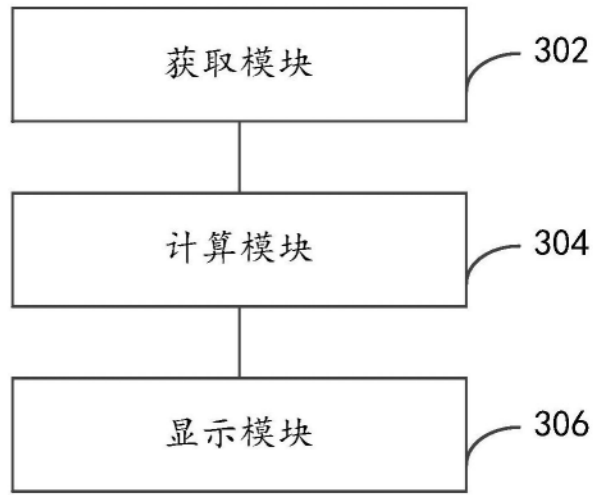


图3

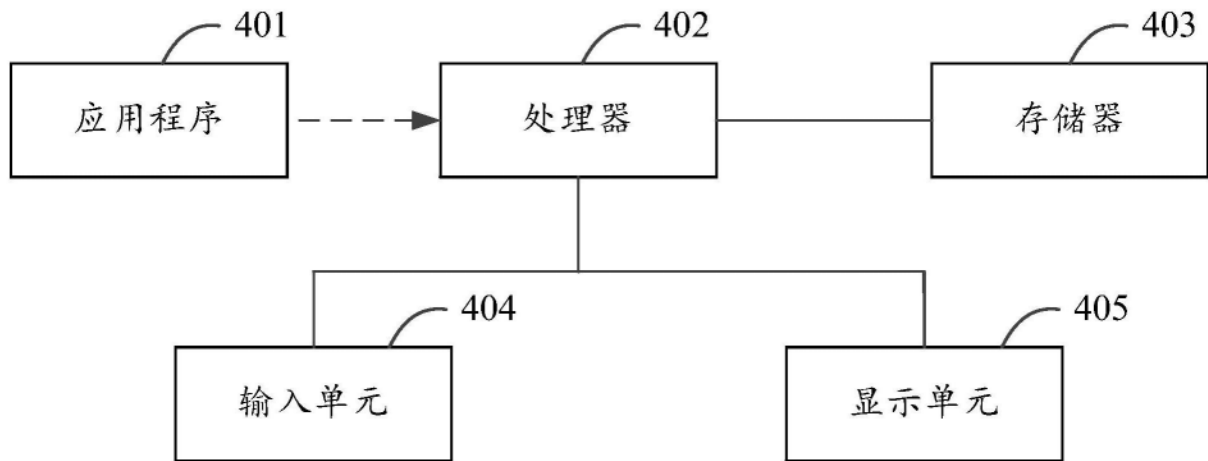


图4