

QUEST中的定位

Unity quest开发：https://blog.csdn.net/qq_46044366/category_12118293.html

QUEST LINK 空间定位

QUEST中的坐标系

https://blog.csdn.net/qq_46044366/article/details/131616046

1、eye level

以头显的现实坐标为参考系中心，所有物体的位置是相对于头显的位置。

选择 Eye Level，会以头显设备本身作为参考系，头部高度相当于摆放在场景中的玩家物体的高度加上头显相对于设备自身坐标系的高度（每次设备定位成功后会建立设备自身的坐标系，以现实世界中头显的位置作为原点），不会额外加上现实中头显到地面的高度。

2、floor level

选择 Floor Level，会以安全区的地面（打开 VR 设备一般都会先设置地面高度，然后划安全区）作为参考系，运行程序后头部高度会以地面作为参考点，高度相当于摆放在场景中的玩家物体的高度加上现实中玩家头显到安全区地面的距离。

3、stage

选择 Stage，和 Floor Level 一样也是以地面作为参考系，但是**不会受头显的重定位影响**。

意思就是不随着摆头和移动和重置改变物体的位置

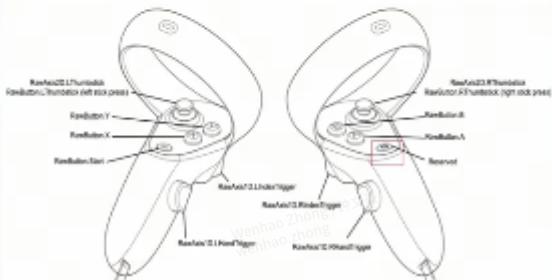
但是如果重启头显或者重划安全区，还是会重新定位

重定位

重定位

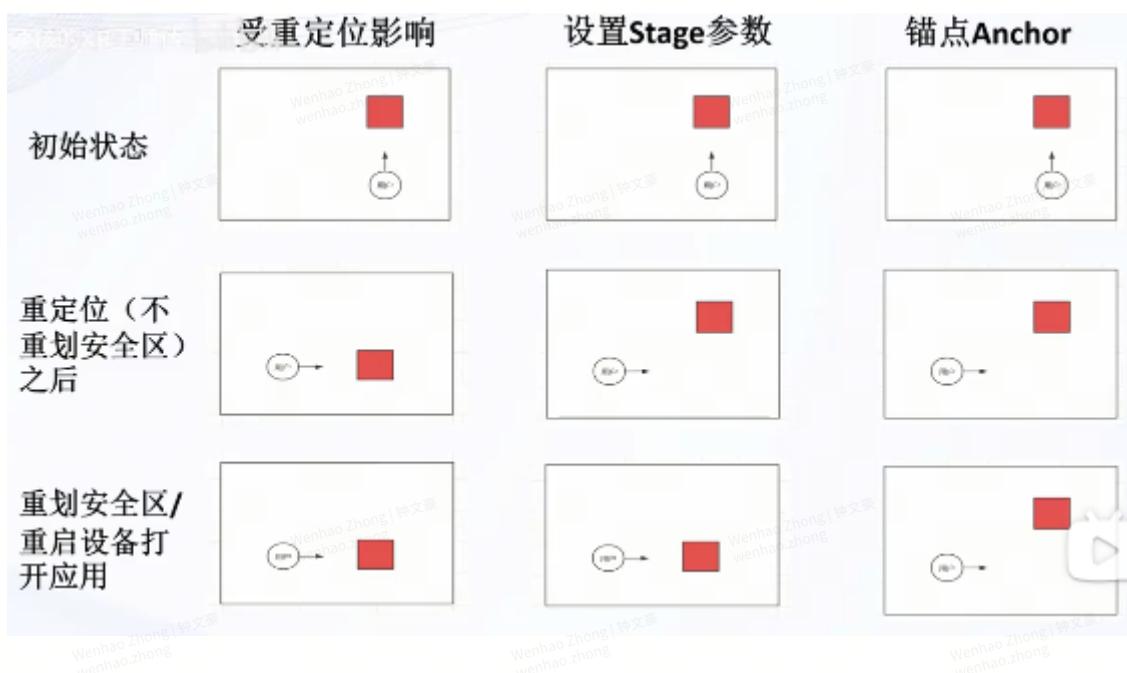
设备将世界坐标原点重置在用户当前的位置，原点的z轴朝向和用户的z轴朝向一致。

- 一般在摘下头显熄屏后重新戴上/重新划定安全区/长按 Quest 右手柄的 Home 键/用 Quest 系统手势来重置中心的时候会进行重定位



将原点置为当前用户的位置，z轴朝向与当前用户的z轴朝向一致（也就是面朝方向）

stage模式应该是不受重定位影响的，但是重新划定安全区或者重启头显进入应用（也相当于重新设置安全区，不需要重新划定，但是需要确定原本的安全边界是否正确）后也会重置



空间设置

Space Setup 空间设置

- Space Setup 空间设置(以前称为场景捕获)是捕获场景模型的过程。它由 Quest 系统管理，因此在设备上运行的所有应用都可以访问相同的环境数据。空间设置是一个用户引导的过程:在设置之前，需要先允许应用访问设备的空间数据。开启了获取空间数据的权限后，首先要扫描环境，获取空间网格，提取空间信息(如地板和天花板的高度，墙壁的位置)，然后用户通过修正错误(校准墙壁位置)和添加缺失信息(房间物体)来完成这个过程。
- 空间设置无法在串流模式下进行。

如果需要设置，要先退出串流模式，然后设置完再进入串流模式

空间锚点

总的来说就是将一个虚拟物体固定在真实位置上

将虚拟物体锚定（固定）在现实世界中的某个位置。每次回到这个位置附近的时候可以看到虚拟物体仍处于完全相同的位置。

- 不受设备重定位的影响
- 退出应用/重启头显后再次打开应用，锚定过的虚拟物体仍然在原来的位置

场景锚点（Scene Anchor）和空间锚点（Spatial Anchor）的区别

场景锚点

在 Quest 系统进行空间设置的时候被创建出来，由系统进行管理。空间设置完毕后，系统会将现实房间的布局以及房间内物体的信息保存在场景锚点中，保证重建出来的虚拟房间和现实的房间匹配，墙壁、地板等物体的位置固定不变。比如一个房间有4面墙，空间设置的时候会给每面墙添加一个场景锚点，然后将虚拟的墙创建在场景锚点的位置，让虚拟的墙贴合现实的墙，并且保证墙的位置固定不动。

空间锚点

由应用本身创建，让用户进行管理。比如用户打开应用后，在现实中创建一个虚拟物体，然后可以在虚拟物体身上创建空间锚点，保存空间锚点的位置信息，这样拥有空间锚点的物体就会被固定在当前的位置。应用重新打开后再加载之前保存的空间锚点数据，这样之前锚定的虚拟物体就知道自己应该出现在什么位置，并且能够出现在相同的位置。

空间锚点只能在一个应用内生效

Wenhao.Zhong | 钟文豪
wenhao.zhong

Wenhao.Zhong | 钟文豪
wenhao.zhong