

Vector3 可以表示(三维向量 ,点,欧拉角等)

向量是一个有方向有大小,没有位置的量

静态属性

-	-
back:	Vector3(0, 0, -1)的简码。
down:	写Vector3(0, -1, 0)的简码。
forward:	写Vector3(0, 0, 1)的简码,也就是向z轴。
left:	写Vector3(-1, 0, 0)的简码。
one:	写Vector3(1, 1, 1)的简码。
right:	写Vector3(1, 0, 0)的简码,也就是向x轴。
up	写Vector3(0, 1, 0)的简码,也就是向y轴
zero	写Vector3(0, 0, 0)的简码。

静态方法

- 1 Angle(); 由from和to两者返回一个角度。
- 2 ClampMagnitude();返回长度限制后向量
- 3 Distance();返回a点和b点之间的距离。
- 4 Cross();两个向量的交叉乘积,返回向量 几何意义:法线 两者长度相乘再*Sin (夹角)
- 5 Dot();两个向量的点乘积。返回标量 几何意义:投影 两者长度相乘再*Cos (夹角)
- 6 Lerp();两个向量之间的线性插值。
- 7 LerpUnclamped();两个向量之间的线性插值。改插值t在小于0或大于1时的返回值不会被限
- 8 MoveTowards();当前的地点移向目标。
- 9 Normalize();使向量的方向不变长度为1。
- 10 OrthoNormalize();使向量规范化并且彼此相互垂直。
- 11 Project();投影一个向量到另一个向量。
- 12 ProjectOnPlane();投影向量到一个平面上由垂直到该平面的法线定义。
- 13 Reflect();沿着法线反射向量。参数是法线 案例:打砖块会弹效果
- 14 RotateTowards();当前的向量转向目标。

属性

-	-
magnitude:	返回向量的长度（只读）。求模长
normalized:	返回长度为1的该向量（只读）。归一化 标准化 方向不变长度变为1
sqrMagnitude:	返回这个向量的长度的平方（只读）。(长度的平方)(为了优化效率)

叉乘:

(4) 叉乘: 两个向量的叉乘得到一个新的向量, 新向量垂直于原来的两个向量, 并且长度等于原向量长度相乘后再乘夹角的正弦值。

点乘:

(3) 点乘: 两个向量点乘得到一个标量, 数值等于两个向量长度相乘后再乘以二者夹角的余弦, 如图18-11所示。

Quaternion 四元数

因为欧拉角表示旋转会出现万向锁问题, Unity所以使用四元数表示旋转
四元数和欧拉角可以互相转换

属性

`eulerAngles`; 返回四元数对应的欧拉角

静态方法

Quaternion.identity//零旋转
Quaternion.Euler();//欧拉角 转 四元数
Quaternion.Angle();//两个四元数的夹角(两个物体前方的夹角)
Quaternion.Lerp();//插值 (可以做缓动旋转效果)
Quaternion.RotateTowards();//匀速旋转到另一个旋转
Quaternion.LookRotation();//返回朝向某个向量的旋转
Quaternion.Inverse();//反旋转 (并非方向反转)
四元数和四元数相乘: 旋转增加
四元数和向量相乘:
四元数*向量 返回把向量沿着四元数表示的旋转量 旋转完成后的向量

作业

实现Unity编辑器窗口中选中物体, 和对选中物体按下f键后镜头靠近的效果

周末作业: 实现一个简单的塔防游戏

包含功能:

- 1 三种怪物, 随机生成
- 2 怪物沿着事先铺好的路点移动
- 3 场景中事先布置好的炮台会自动攻击距离小于x的怪物
- 4 怪物有自己的血量, 受伤后血量减少, 血量小于等于0则死亡