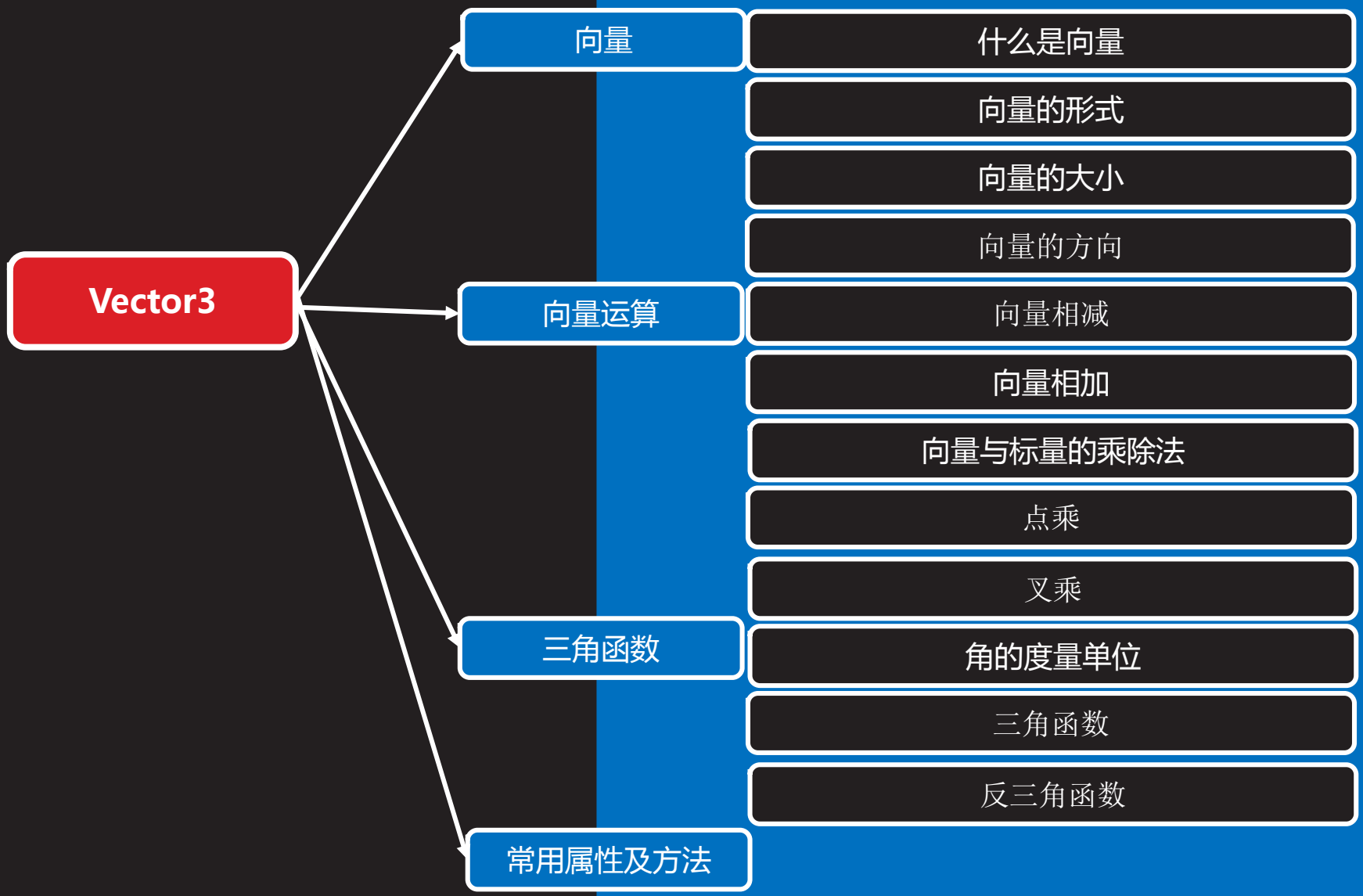
Technology Tarena 达内科技

3D 数学基础

3D MATH

向量 | 欧拉角 | 四元数 | 坐标系统

Vector3


Vector3
向量
什么是向量
向量的形式
向量的大小
向量的方向
向量运算
向量相减
向量相加
向量与标量的乘除法
点乘
叉乘
三角函数
角的度量单位
三角函数
反三角函数
常用属性及方法

Technology Tarena 达内科技

向量

什么是向量

Technology Tarena 达内科技

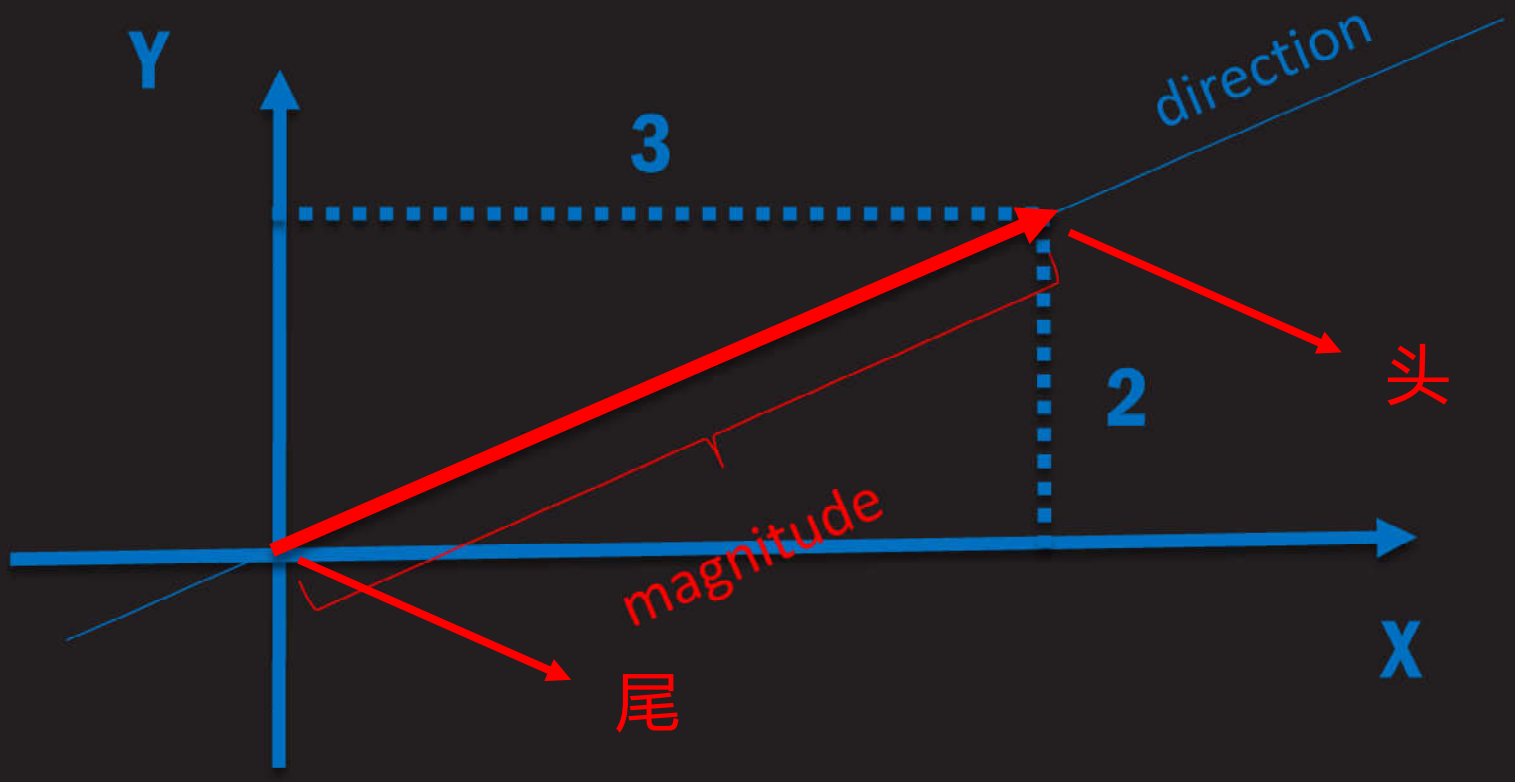
- 一个数字列表，表示各个维度上的有向位移。
- 一个有大小有方向的物理量。
  -- 大小就是向量的模长。
  -- 方向描述了空间中向量的指向。
- 可以表示物体的位置和方向。

知识讲解

向量的形式

Technology Tarena 达内科技

知识讲解



向量的大小(模)

Technology Tarena 达内科技

- 向量各分量平方和的平方根。
- 公式：$\sqrt[2]{X^2+y^2+z^2}$
- API： float dis=vector.magnitude;

  -- 模的平方　vector.sqrMagnitude

  因为平方根的计算耗时长，所以效率高于magnitude。

知识讲解

向量的方向

- 获取向量方向也称“标准化向量”，或“归一化向量”；即获取该向量的单位向量。
- 单位向量：大小为1的向量。
- 公式： V / |V|
- 几何意义：将该向量拉长或者缩短，使模长等于1。
- API：Vector3 vector2=vector1.normalized;

  -- vector2为vector1的单位向量

  -- vector1.Normalize(); 将vector1自身设置为单位向量

知识讲解

Technology Tarena 达内科技

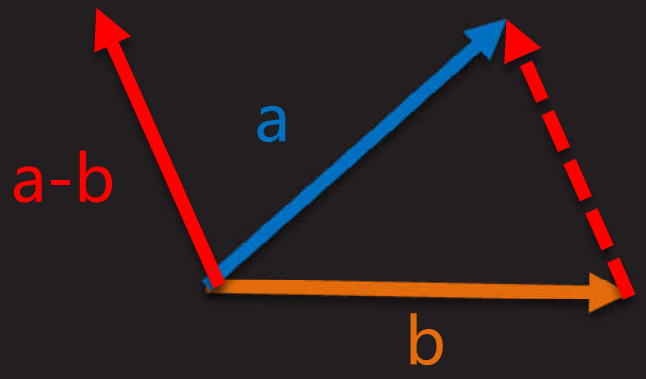
向量运算(1)

向量相减

Technology Tarena 达内科技

- 等于各分量相加减
- 公式： [x1,y1,z1] - [x2,y2,z2] = [x1-x2,y1-y2,z1-z2]
- 几何意义：向量a与向量b相减，结果理解为以b的终点为始点，以a的终点为终点的向量。方向由b指向a。
- 应用：计算两点之间的距离和相对方向。

知识讲解

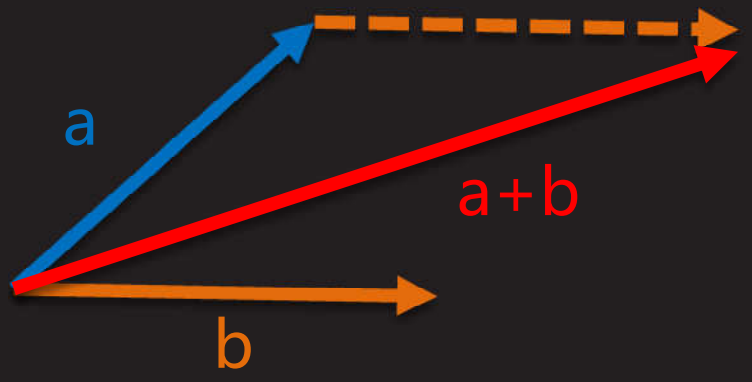


向量相加

Technology Tarena 达内科技

- 等于各分量相加和。
- 公式： [x1,y1,z1] + [x2,y2,z2] = [x1+x2,y1+y2,z1+z2]
- 几何意义: 向量a与向量b相加，平移使b的始点与a的终点重合，结果为以a的始点为始点，以b的终点为终点的向量。
- 应用：物体移动

知识讲解

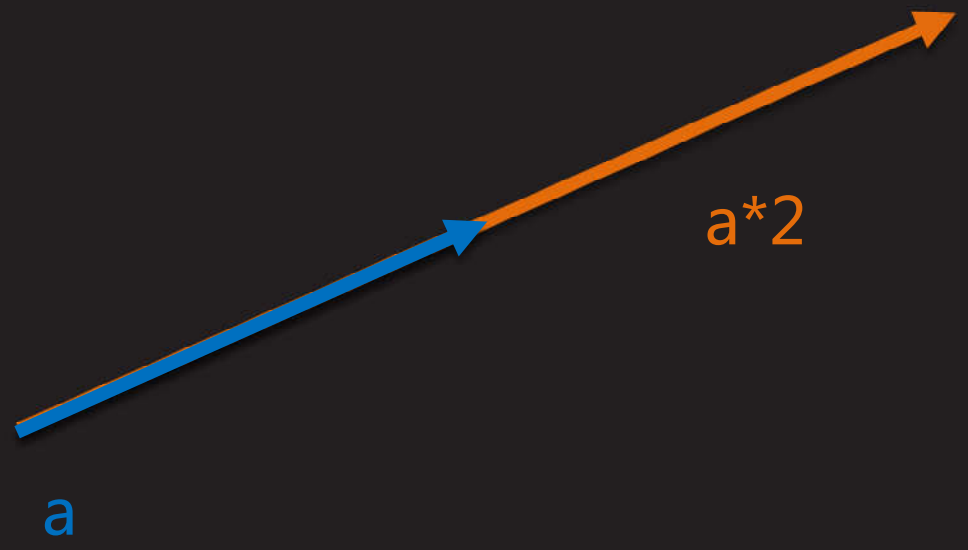


向量与标量的乘除

Technology Tarena 达内科技

- 乘法：该向量的各分量与标量相乘 k[x,y,z] = [xk,yk,zk]
- 除法：该向量的各分量与标量相除 [x,y,z]/k = [x/k,y/k,z/k]
- 几何意义：缩放向量长度。

知识讲解



Technology Tarena 达内科技

三角函数

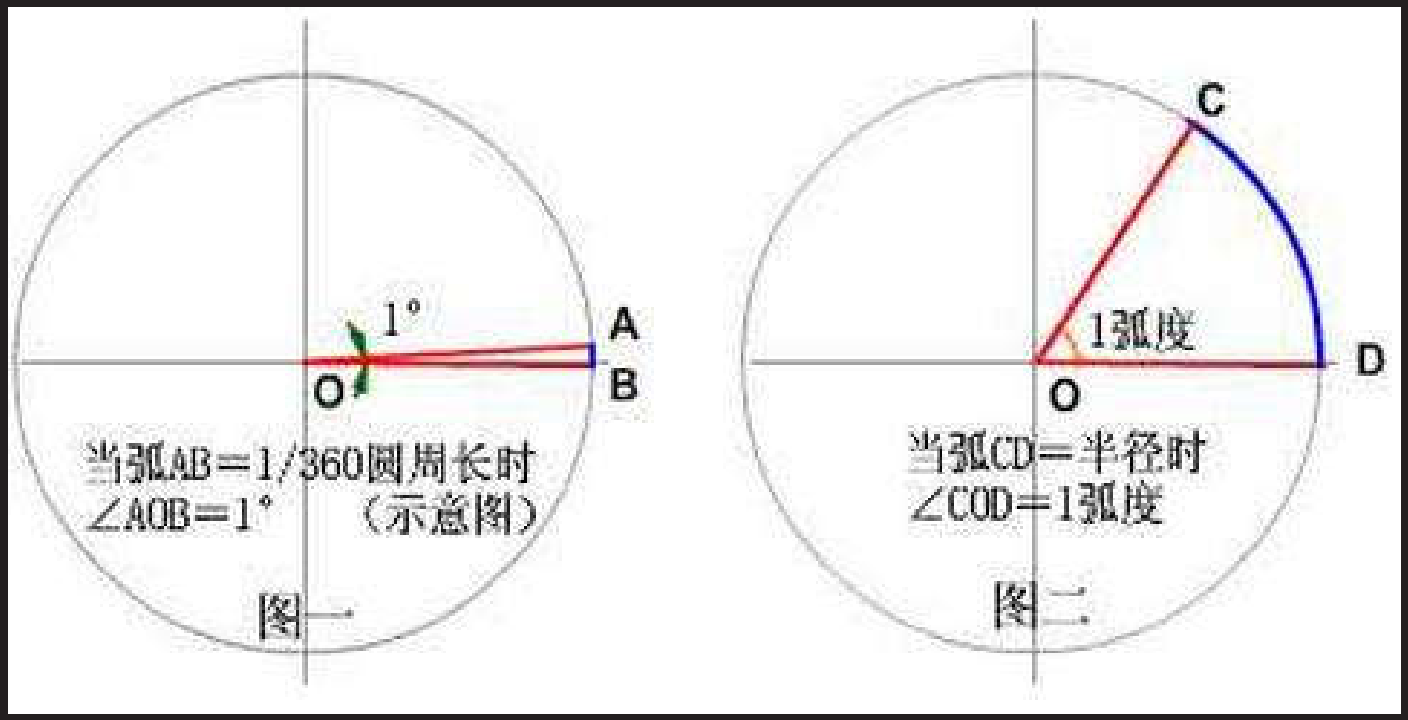
角的度量方式

Technology Tarena 达内科技

知识讲解

- 角度Degree与弧度Radian

两条射线从圆心向圆周射出，形成一个夹角和夹角正对的一段弧。当弧长等于圆周长的360分之一时，夹角为1度。弧长等于圆的半径时，夹角为1弧度。



角的度量方式(续1)

知识讲解

- 角度与弧度的换算：

  PI=180度　　1弧度=180度/PI　　1角度=PI/180度

  角度=>弧度：弧度=角度数*PI/180

  API：弧度=角度数*Mathf.Deg2Rad

  弧度=>角度：角度=弧度数*180/PI

  API：角度=弧度数*Mathf.Rad2Deg

- 在日常生活中角度制应用比较广泛。
- 在三角函数中弧度制可以简化计算。

三角函数

Technology Tarena 达内科技

知识讲解

- 建立了直角三角形中角与边长比值的关系。
- 可用于根据一边一角，计算另外一边长。
- 公式：

  正弦 sin x = a / c ;

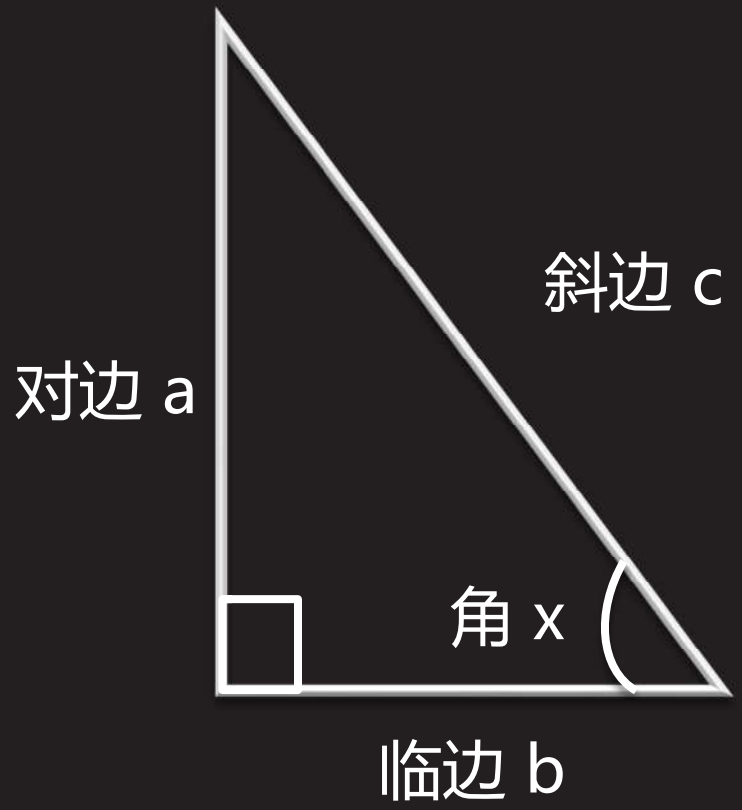
  余弦 cos x = b / c ;

  正切 tan x = a / b ;
- API：

  Mathf.Sin(float radian)

  Mathf.Cos(float radian)

  Mathf.Tan(float radian)



反三角函数

Technology Tarena 达内科技

知识讲解

- 反正弦，反余弦，反正切等函数的总称。
- 可用于根据两边长，计算角度。
- 公式：

  反正弦 arcsin a / c = x ;

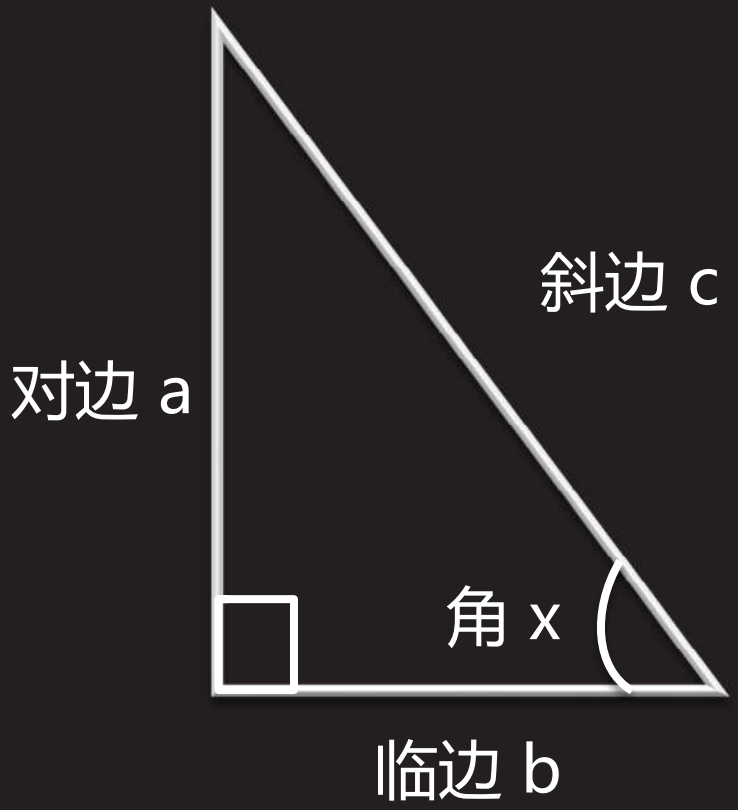
  反余弦 arccos b / c = x ;

  反正切 arctan a / b = x ;
- API：

  Mathf.Asin(float radian)

  Mathf.Acos(float radian)

  Mathf.Atan(float radian)



Technology Tarena 达内科技

向量运算(2)

点乘

- 又称“点积”或”内积” 。
- 公式：各分量乘积和

  [x1,y1,z1] · [x2,y2,z2] = x1x2+y1y2+z1z2
- 几何意义：a·b=|a| · |b| cos<a,b>

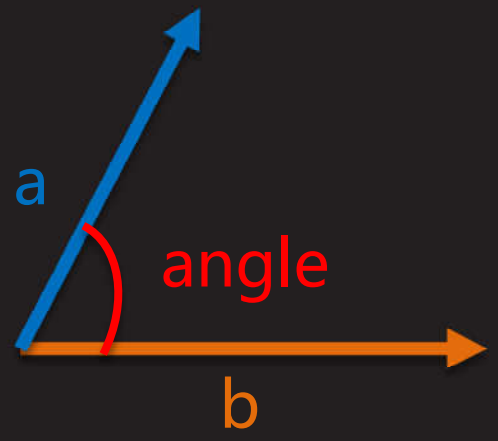
  两个向量的单位向量相乘后再乘以二者夹角的余弦值。
- API：float dot=Vector3.Dot(va, vb);

知识讲解

应用

- 对于标准化过的向量，点乘结果等于两向量夹角的余弦值。
- 应用：计算两向量夹角

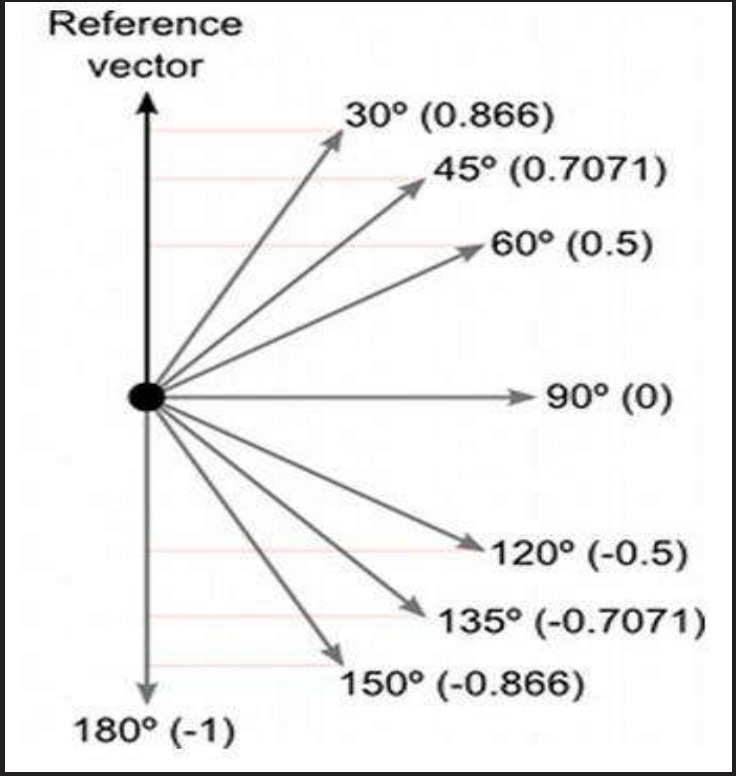
```
float dot =Vector3.Dot(a.normalized, b.normalized);
float angle = Mathf.Acos(dot) * Mathf.Rad2Deg
```



知识讲解

结果与角度关系

- 对于标准化过的向量，方向完全相同，点乘结果为1，完全相反，点乘结果为-1，互相垂直结果为0。



知识讲解

叉乘

Technology Tarena 达内科技

- 又称“叉积”或”外积” 。
- 公式： [x1,y1,z1] x [x2,y2,z2] =

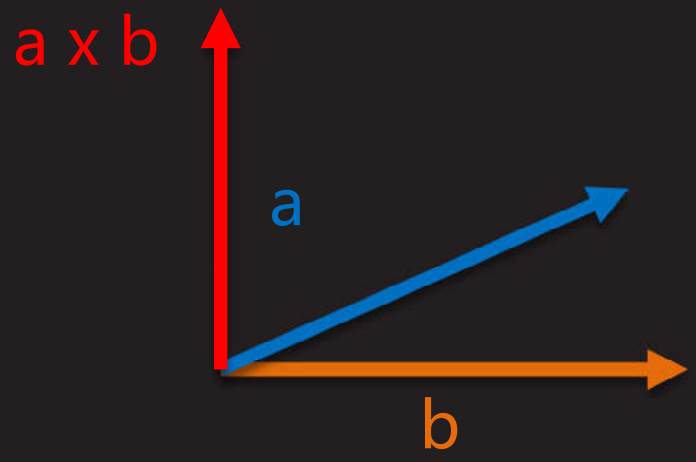
  [y1 * z2 - z1 * y2, z1 * x2 - x1 * z2, x1 * y2 - y1 * x2]
- 几何意义：结果为两个向量所组成面的垂直向量，模长为两向量模长乘积再乘夹角的正弦值。
- 脚本: Vector vector=Vector3. Cross (a, b);

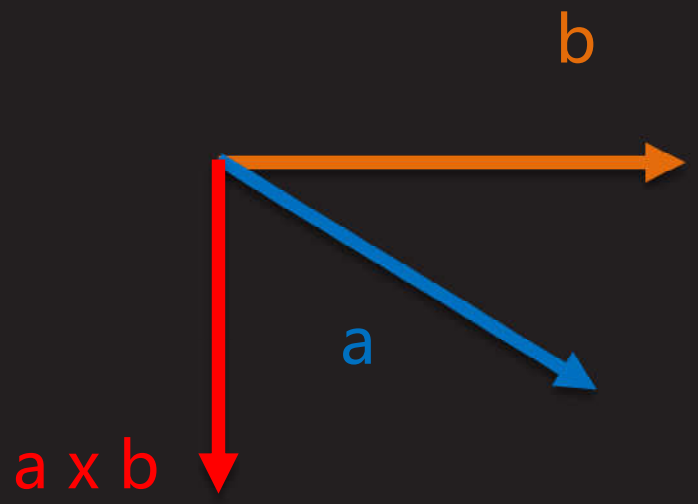
知识讲解

应用

- 创建垂直于平面的向量。
- 判断两条向量相对位置。

知识讲解

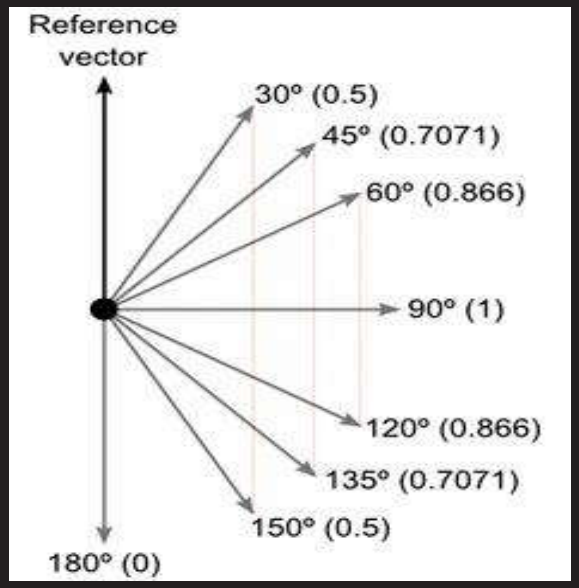




结果与角度关系

- 叉乘所得向量的模长与角度关系： 0~90度角

```
Vector3 cross=
        Vector3.Cross(a.normalized, b.normalized);
float angle =
        Mathf.Asin(cross.magnitude) * Mathf.Rad2Deg;
```



知识讲解

Technology Tarena 达内科技

常用属性及方法

静态属性

- Vector3.up => new Vector3(0,1,0)
  Vector3.down => new Vector3(0,-1,0)
  Vector3.left => new Vector3(-1,0,0)
  Vector3.right => new Vector3(1,0,0)
  Vector3.forward => new Vector3(0,0,1)
  Vector3.back => new Vector3(0,0,-1)

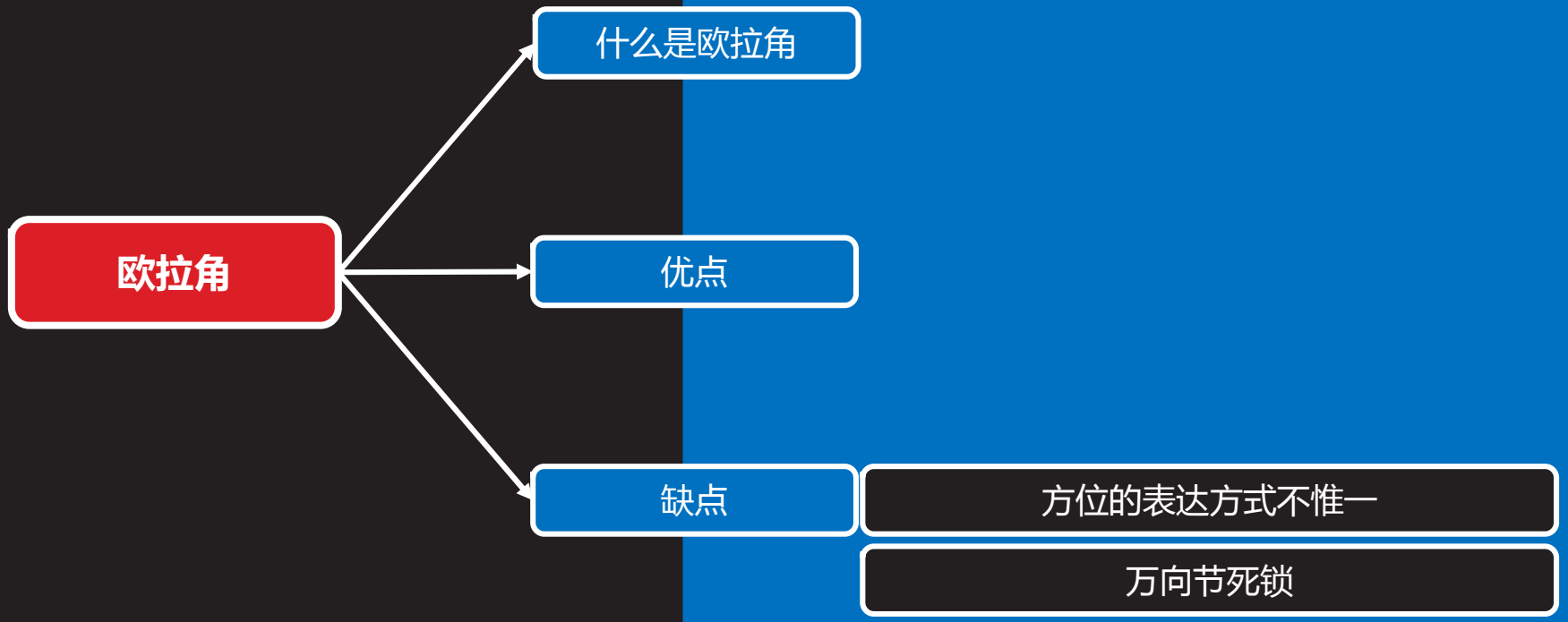
知识讲解

静态方法

- Vector3.Lerp
  Vector3.MoveTowards
  Vector3.SmoothDamp
  Vector3.Angle
  Vector3.ProjectOnPlane
  Vector3.Reflect
  ……

知识讲解

欧拉角



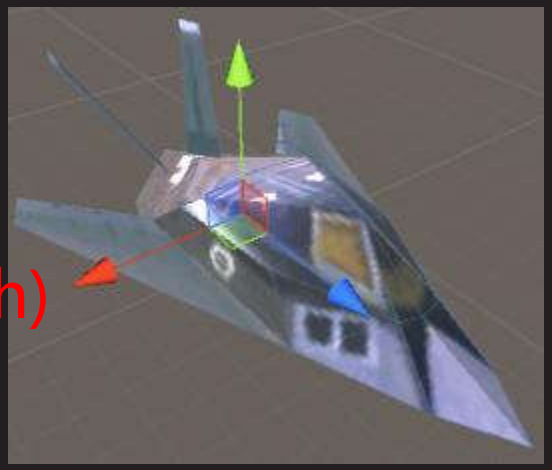
Technology Tarena 达内科技

什么是欧拉角

什么是欧拉角

- 使用三个角度来保存方位。
- API：Vector3 eulerAngle = this.transform.eulerAngles;

知识讲解



Technology Tarena 达内科技

优点

优点

- 仅使用三个数字表达方位，占用空间小。
- 沿坐标轴旋转的单位为角度，符合人的思考方式。
- 任意三个数字都是合法的，不存在不合法的欧拉角。

知识讲解

Technology Tarena 达内科技

缺点

方位的表达方式不维一

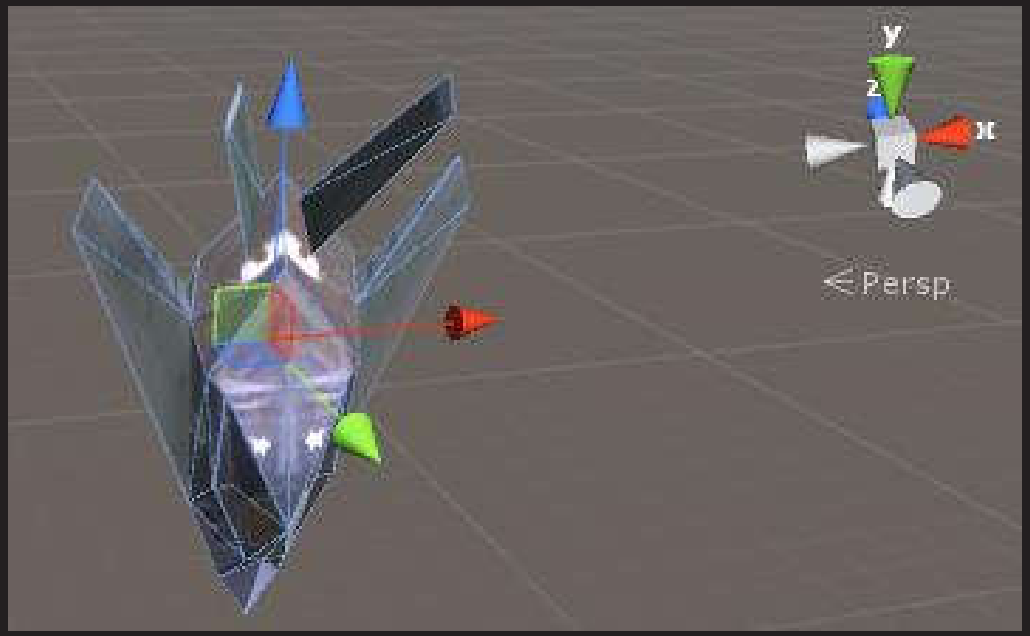
Technology Tarena 达内科技

- 对于一个方位，存在多个欧拉角描述，因此无法判断多个欧拉角代表的角位移是否相同。
- 例如：
  -- 角度0,5,0与角度0,365,0
  -- 角度0,-5,0与角度0,355,0
  -- 角度250,0,0与角度290,180,180
- 为了保证任意方位都只有独一无二的表示，Unity引擎限制了角度范围，即沿X轴旋转限制在-90到90之间，沿Y与Z轴旋转限制在0到360之间。

知识讲解

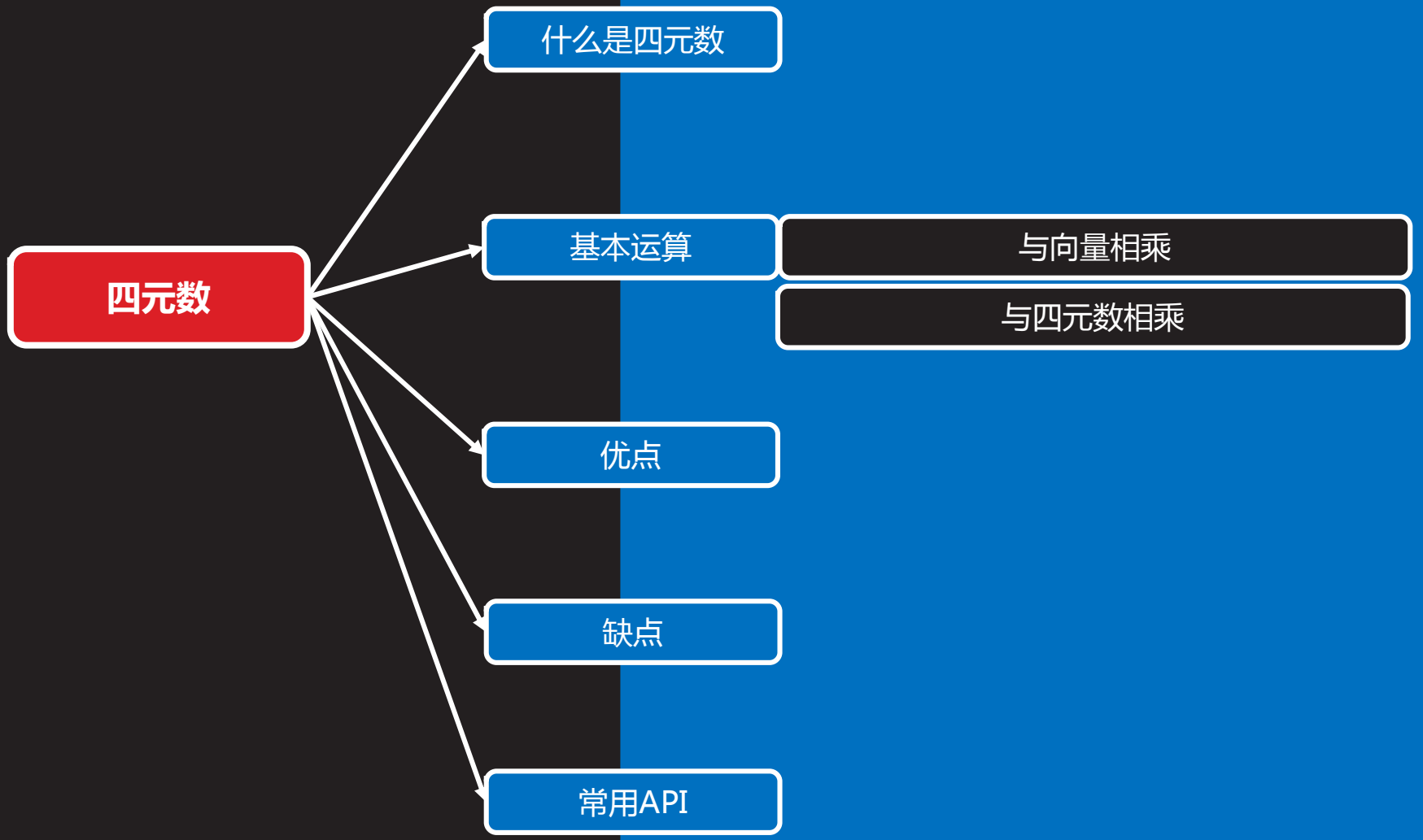
万向节死锁

- 物体沿X轴旋转±90度，自身坐标系Z轴与世界坐标系Y轴将重合，此时再沿Y或Z轴旋转时，将失去一个自由度。
- 在万向节死锁情况下，规定沿Y轴完成绕竖直轴的全部旋转，即此时Z轴旋转为0。



知识讲解

四元数



Technology Tarena 达内科技

什么是四元数

什么是四元数

Technology Tarena 达内科技

知识讲解

- Quaternion 在3D图形学中代表旋转，由一个三维向量(X/Y/Z)和一个标量(W)组成。
- 旋转轴为V，旋转弧度为θ ，如果使用四元数表示，则四个分量为：

  x=sin(θ /2)*V.x　　y=sin(θ /2)*V.y

  z=sin(θ /2)*V.z　　w=cos(θ /2)
- X、Y、Z、W的取值范围是-1到1。
- API：Quaternion qt = this.transform.rotation;

Technology Tarena 达内科技

基本运算

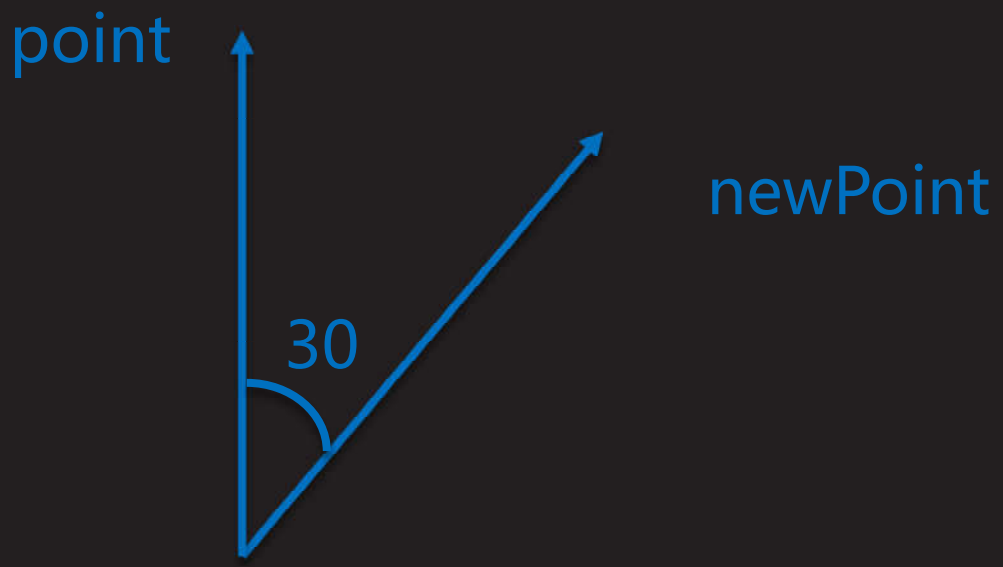
与向量相乘

Technology Tarena 达内科技

- 四元数左乘向量，表示将该向量按照四元数表示的角度旋转。
- 例如：

```
Vector3 point = new Vector3(0,0,10);
Vector3 newPoint = Quaternion.Euler(0,30,0) * point ;
```

知识讲解



与四元数相乘

- 两个四元数相乘可以组合旋转效果。
- 例如：

```
Quaternion rotation01 =
    Quaternion.Euler(0, 30, 0) * Quaternion.Euler(0, 20, 0);
Quaternion rotation02 = Quaternion.Euler(0, 50, 0);
-- rotation01 与 rotation02 相同
```

知识讲解

Technology Tarena 达内科技

优点

避免万向节死锁

- this.transform.rotation *= Quaternion.Euler(0, 1, 0);
  -- 可使物体沿自身坐标Y轴旋转
- this.transform.Rotate(Vector3 eulerAngles)
  -- 内部就是使用四元数相乘实现

知识讲解

Technology Tarena 达内科技

缺点

缺点

- 难于使用，不建议单独修改某个数值。
- 存在不合法的四元数。

知识讲解

Technology Tarena 达内科技

常用API

常用API

- Quaternion.LookRotation
- Quaternion.Euler
- Quaternion.Lerp
- Quaternion.FromToRotation
- Quaternion.AngleAxis

知识讲解

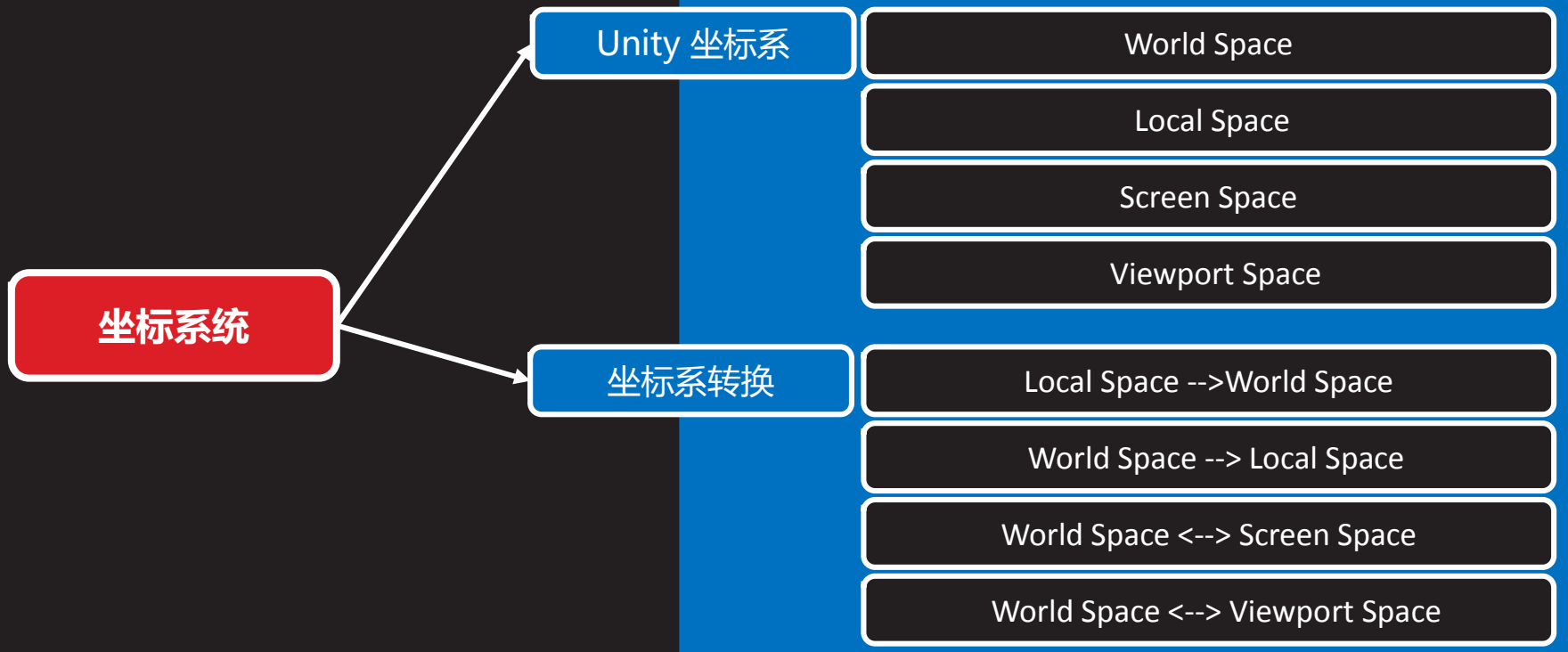
Technology Tarena 达内科技

练习

- 根据用户输入的方向旋转角色，并向前移动。

课堂练习

坐标系统


坐标系统
Unity 坐标系
World Space
Local Space
Screen Space
Viewport Space
坐标系转换
Local Space -->World Space
World Space --> Local Space
World Space <--> Screen Space
World Space <--> Viewport Space

Technology Tarena 达内科技

Unity 坐标系

World Space

Technology Tarena 达内科技

- 世界(全局)坐标系：整个场景的固定坐标。
- 作用：在游戏场景中表示每个游戏对象的位置和方向。

知识讲解

Local Space

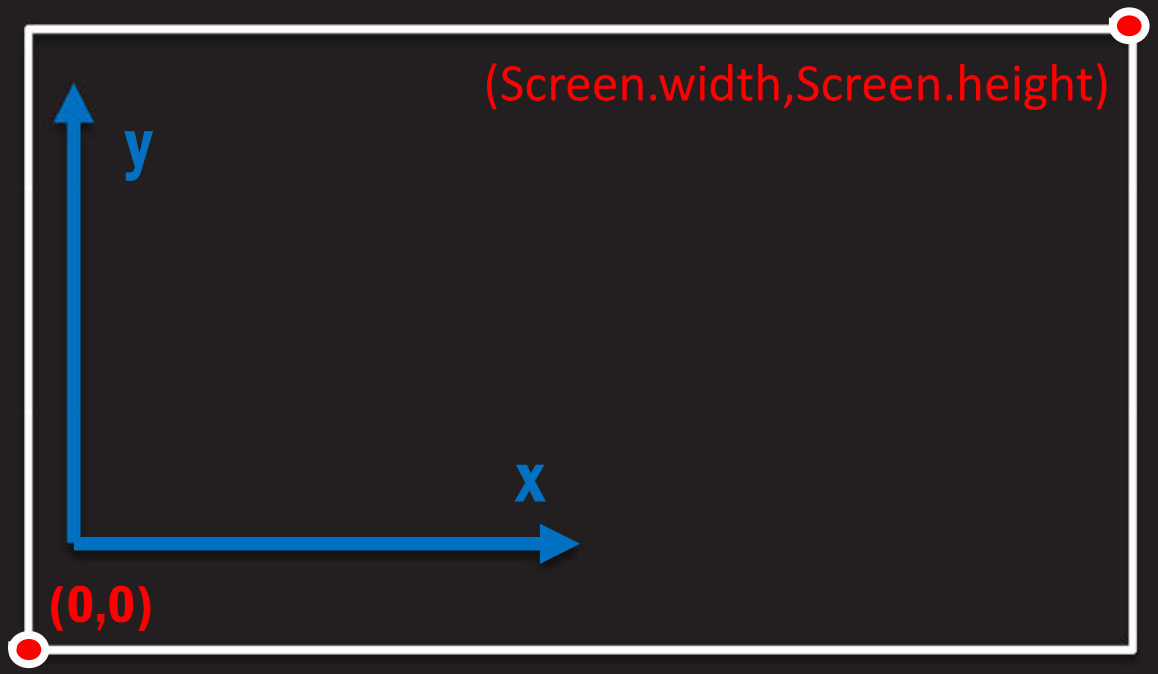
Technology Tarena 达内科技

- 物体(局部)坐标系：每个物体独立的坐标系，原点为模型轴心点，随物体移动或旋转而改变。
- 作用：表示物体间相对位置与方向。

知识讲解

Screen Space

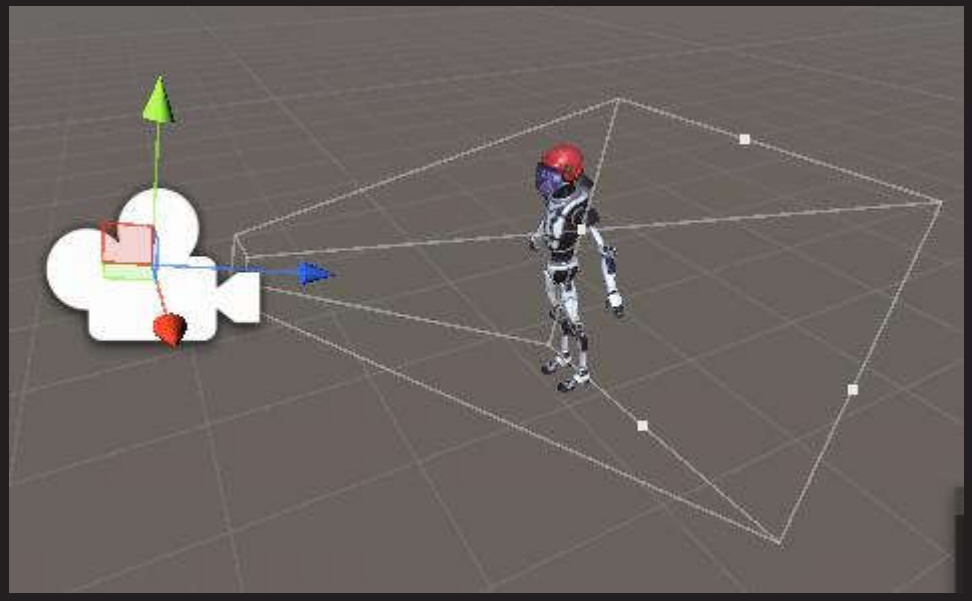
- 屏幕坐标系：以像素为单位，屏幕左下角为原(0，0)点，右上角为屏幕宽、高度(Screen.width，Screen.height)，Z为到相机的距离。
- 作用：表示物体在屏幕中的位置。



知识讲解

Viewport Space

- 视口(摄像机)坐标系：屏幕左下角为原(0，0)点，右上角为(1,1)，Z为到相机的距离。
- 作用：表示物体在摄像机中的位置。



知识讲解

Technology Tarena 达内科技

坐标系转换

Local Space --> World Space

知识讲解

- transform.forward 在世界坐标系中表示物体正前方。
- transform.right 在世界坐标系中表示物体正右方。
- transform.up 在世界坐标系中表示物体正上方。

- transform.TransformPoint
  转换点，受变换组件位置、旋转和缩放影响。
- transform.TransformDirection
  转换方向，受变换组件旋转影响。
- transform.TransformVector
  转换向量，受变换组件旋转和缩放影响。

World Space --> Local Space

Tarena Technology 达内科技

知识讲解

- transform.InverseTransformPoint
  转换点，受变换组件位置、旋转和缩放影响。
- transform.InverseTransformDirection
  转换方向，受变换组件旋转影响。
- transform.InverseTransformVector
  转换向量，受变换组件旋转和缩放影响。

World Space <--> Screen Space

Technology Tarena 达内科技

- Camera.main.WorldToScreenPoint
  将点从世界坐标系转换到屏幕坐标系中
- Camera.main.ScreenToWorldPoint
  将点从屏幕坐标系转换到世界坐标系中

知识讲解

World Space <--> Viewport Space

- Camera.main.WorldToViewportPoint
  将点从世界坐标系转换到视口坐标系中
- Camera.main.ViewportToWorldPoint
  将点从屏幕坐标系转换到世界坐标系中

知识讲解