

[首页](#) [资讯](#) [精华](#) [论坛](#) [问答](#) [博客](#) [专栏](#) [群组](#) [更多 ▼](#)

[您还未登录！](#) [登录](#) [注册](#)

zhboy666666

- [博客](#)
- [微博](#)
- [相册](#)
- [收藏](#)
- [留言](#)
- [关于我](#)

Unity Mathf 数学运算(C#)

博客分类：

- [Unity3D](#)

Mathf.Abs绝对值

计算并返回指定参数 f 绝对值。

Mathf.Acos反余弦

static function Acos (f : float) : float

以弧度为单位计算并返回参数 f 中指定的数字的反余弦值。

Mathf.Approximately近似

static function Approximately (a : float, b: float) : bool

比较两个浮点数值，看它们是否非常接近，由于浮点数值不精确，不建议使用等于来比较它们。例如，1.0==10.0/10.0也许不会返回true。

```
public class example : MonoBehaviour {  
    public void Awake() {  
        if(Mathf.Approximately(1.0F, 10.0F / 10.0F))  
            print("same");  
    }  
}
```

关闭

Mathf.Asin反正弦

static function Asin (f : float) : float

以弧度为单位计算并返回参数 f 中指定的数字的反正弦值。

Mathf.Atan2反正切

static function Atan2 (y : float, x :float) : float

以弧度为单位计算并返回 y/x 的反正切值。返回值表示相对直角三角形对角的角，其中 x 是临边边长，而 y 是对边边长。

返回值是在x轴和一个二维向量开始于0个结束在(x,y)处之间的角。

```
public class example : MonoBehaviour {
    public Transform target;
    void Update() {
        Vector3 relative = transform.InverseTransformPoint(target.position);
        float angle = Mathf.Atan2(relative.x, relative.z) * Mathf.Rad2Deg;
        transform.Rotate(0, angle, 0);
    }
}
```

Mathf.Atan反正切

static function Atan (f : float) :float

计算并返回参数 f 中指定的数字的反正切值。返回值介于负二分之 pi 与正二分之 pi 之间。

Mathf.CeilToInt最小整数

static function CeilToInt (f : float) : int

返回最小的整数大于或等于f。

Mathf.Ceil上限值

static function Ceil (f : float) : float

返回 f 指定数字或表达式的上限值。数字的上限值是大干等于该数字的最接近的整数。

Mathf.Clamp01限制0~1

static function Clamp01 (value : float) :float

限制value在0,1之间并返回value。如果value小于0，返回0。如果value大于1,返回1，否则返回value。

Mathf.Clamp限制

static function Clamp (value : float, min :float, max : float) : float

限制value的值在min和max之间，如果value小于min，返回min。如果value大于max，返回max，否则返回value

static function Clamp (value : int, min :int, max : int) : int

限制value的值在min和max之间，并返回value。

[关闭](#)
Mathf.ClosestPowerOfTwo最近的二次方

static function ClosestPowerOfTwo (value :int) : int

返回距离value最近的2的次方数。

Mathf.Cos余弦

static function Cos (f : float) : float

返回由参数 f 指定的角的余弦值 (介于 -1.0 与 1.0 之间的值) 。

Mathf.Deg2Rad度转弧度

static var Deg2Rad : float

度到弧度的转化常量。(只读)

这等于 $(\pi * 2) / 360$ 。

Mathf.Mathf.Rad2Deg 弧度转度

static var Rad2Deg : float

弧度到度的转化常量。(只读)

这等于 $360 / (\pi * 2)$ 。

Mathf.DeltaAngle增量角

static function DeltaAngle (current :float, target : float) : float

计算给定的两个角之间最短的差异。

// Prints 90

Debug.Log(Mathf.DeltaAngle(1080,90));

Mathf.Epsilon小正数

static var Epsilon : float

一个很小的浮点数值。(只读)

最小的浮点值，不同于0。

以下规则：

- anyValue + Epsilon = anyValue
- anyValue - Epsilon = anyValue
- 0 + Epsilon = Epsilon
- 0 - Epsilon = -Epsilon

一个在任意数和Epsilon的之间值将导致在任意数发生截断误差。

public class example : MonoBehaviour {

 bool isEqual(float a, float b) {

 if(a >= b - Mathf.Epsilon && a <= b + Mathf.Epsilon)
 return true;

 else

 return false;

 }

}

[关闭](#)

Mathf.Exp指数

static function Exp (power : float) : float

返回 e 的 power 次方的值。

Mathf.FloorToInt最大整数

static function FloorToInt (f : float) : int

返回最大的整数，小于或等于f。

Mathf.Floor下限值

static function Floor (f : float) : float

返回参数 f 中指定的数字或表达式的下限值。下限值是小于等于指定数字或表达式的最接近的整数。

Mathf.Infinity正无穷

static var Infinity : float

表示正无穷，也就是无穷大， ∞ （只读）

Mathf.InverseLerp反插值

计算两个值之间的Lerp参数。也就是value在from和to之间的比例值。

//现在参数是3/5

float parameter =Mathf.InverseLerp(walkSpeed, runSpeed, speed);

Mathf.IsPowerOfTwo是否2的幂

static function IsPowerOfTwo (value : int): bool

如果该值是2的幂，返回true。

// prints false

Debug.Log(Mathf.IsPowerOfTwo(7));

// prints true

Debug.Log(Mathf.IsPowerOfTwo(32));

Mathf.LerpAngle插值角度

static function LerpAngle (a : float, b :float, t : float) : float

和Lerp的原理一样，当他们环绕360度确保插值正确。

a和b是代表度数。

```
public class example : MonoBehaviour {
```

```
    public float minAngle = 0.0F;
```

```
    public float maxAngle = 90.0F;
```

```
    void Update() {
```

```
        float angle = Mathf.LerpAngle(minAngle, maxAngle, Time.time);
```

```
        transform.eulerAngles= new Vector3(0, angle, 0);
```

```
    }
```

```
}
```

关闭

Mathf.Lerp插值

static function Lerp (from : float, to :float, t : float) : float

基于浮点数t返回a到b之间的插值，t限制在0~1之间。

当t = 0返回from，当t = 1 返回to。当t = 0.5 返回from和to的平均值。

Mathf.Log10基数10的对数

static function Log10 (f : float) : float

返回f的对数，基数为10。

Mathf.Log对数

static function Log (f : float, p : float): float

返回参数 f 的对数。

// logarithm of 6 in base 2

//以2为底6的对数

// prints 2.584963

print(Mathf.Log(6, 2));

Mathf.Max最大值

static function Max (a : float, b : float): float

static function Max (params values :float[]) : float

返回两个或更多值中最大的值。

Mathf.Min最小值

static function Min (a : float, b : float): float

static function Min (params values :float[]) : float

返回两个或更多值中最小的值。

Mathf.MoveTowardsAngle移动角

static function MoveTowardsAngle (current :float, target : float, maxDelta : float) : float

像MoveTowards,但是当它们环绕360度确保插值正确。

变量current和target是作为度数。为优化原因，maxDelta负值的不被支持，可能引起振荡。从target角推开current，添加180度角代替。

Mathf.MoveTowards移向

static function MoveTowards (current :float, target : float, maxDelta : float) : float

改变一个当前值向目标值靠近。

这实际上和 Mathf.Lerp相同，而是该函数将确保我们的速度不会超过maxDelta。maxDelta为负值将目标从推离。

Mathf.NegativeInfinity负无穷

static var NegativeInfinity : float

表示负无穷，也就是无穷小， $-\infty$ （只读）

[关闭](#)

Mathf.NextPowerOfTwo下个2的幂

Mathf.PingPong乒乓

static function PingPong (t : float, length: float) : float

0到length之间往返。t值永远不会大于length的值，也永远不会小于0。

The returned value will move back and forth between 0 and length.

返回值将在0和length之间来回移动。

Mathf.PI圆周率

static var PI : float

PI (读pai) 的值, 也就是圆周率 (π) 的值3.14159265358979323846... (只读)

Mathf.Pow次方

static function Pow (f : float, p : float): float

计算并返回 f 的 p 次方。

Mathf.Repeat重复

static function Repeat (t : float, length :float) : float

循环数值t, 0到length之间。t值永远不会大于length的值, 也永远不会小于0。

这是类似于模运算符, 但可以使用浮点数。

```
public class example : MonoBehaviour {  
    void Update() {  
        transform.position= new Vector3(Mathf.Repeat(Time.time, 3),  
transform.position.y,transform.position.z);  
    }  
}
```

Mathf.RoundToInt四舍五入到整数

static function RoundToInt (f : float) :int

返回 f 指定的值四舍五入到最近的整数。

如果数字末尾是.5, 因此它是在两个整数中间, 不管是偶数或是奇数, 将返回偶数。

Mathf.Round四舍五入

static function Round (f : float) : float

返回浮点数 f 进行四舍五入最接近的整数。

如果数字末尾是.5, 因此它是在两个整数中间, 不管是偶数或是奇数, 将返回偶数。

Mathf.Sign符号

static function Sign (f : float) : float

返回 f 的符号。

当 f 为正或为0返回1, 为负返回-1。

[关闭](#)

Mathf.Sin正弦

static function Sin (f : float) : float

计算并返回以弧度为单位指定的角 f 的正弦值。

Mathf.SmoothDampAngle平滑阻尼角度

static function SmoothDampAngle (current :float, target : float, ref currentVelocity : float,
smoothTime : float,maxSpeed : float = Mathf.Infinity, deltaTime : float = Time.deltaTime) :
float

参数

current

当前的位置。

target

我们试图达到的位置。

currentVelocity

当前速度，这个值在你访问这个函数的时候会被随时修改。

smoothTime

the target faster.

要到达目标位置的近似时间，实际到达目标时要快一些。

maxSpeed

可选参数，允许你限制的最大速度。

deltaTime

上次调用该函数到现在的时间。缺省为Time.deltaTime。

随着时间的推移逐渐改变一个给定的角度到期望的角度。

这个值通过一些弹簧减震器类似的功能被平滑。这个函数可以用来平滑任何一种值，位置，颜色，标量。最常见的是平滑一个跟随摄像机。

//一个简单的平滑跟随摄像机

//跟随目标的朝向

```
public class example : MonoBehaviour {
```

```
    public Transform target;
```

```
    public float smooth = 0.3F;
```

```
    public float distance = 5.0F;
```

```
    private float yVelocity = 0.0F;
```

```
    void Update() {
```

```
        //从目前的y角度变换到目标y角度
```

```
        float yAngle = Mathf.SmoothDampAngle(transform.eulerAngles.y,
```

```
target.eulerAngles.y, ref yVelocity, smooth);
```

```
        //target的位置
```

```
        Vector3 position = target.position;
```

```
        //然后，新角度之后的距离偏移
```

```
        position += Quaternion.Euler(0, yAngle, 0) * new Vector3(0, 0, -distance);
```

```
        //应用位置
```

```
        transform.position = position;
```

```
        //看向目标
```

```
        transform.LookAt(target);
```

```
    }
```

```
}
```

[关闭](#)

Mathf.SmoothDamp平滑阻尼

static function SmoothDamp (current :float, target : float, ref currentVelocity : float, smoothTime : float,maxSpeed : float = Mathf.Infinity, deltaTime : float = Time.deltaTime) : float

参数

current

当前的位置。

target

我们试图达到的位置。

currentVelocity

当前速度，这个值在你访问这个函数的时候会被随时修改。

smoothTime

要到达目标位置的近似时间，实际到达目标时要快一些。

maxSpeed

可选参数，允许你限制的最大速度。

deltaTime

上次调用该函数到现在的时间。缺省为Time.deltaTime。

描述

随着时间的推移逐渐改变一个值到期望值。

这个值就像被一个不会崩溃的弹簧减振器一样被平滑。这个函数可以用来平滑任何类型的值，位置，颜色，标量。

```
public class example : MonoBehaviour {
    public Transform target;
    public float smoothTime = 0.3F;
    private float yVelocity = 0.0F;
    void Update() {
        float newPosition = Mathf.SmoothDamp(transform.position.y, target.position.y,
        ref yVelocity, smoothTime);
        transform.position = new Vector3(transform.position.x, newPosition,
        transform.position.z);
    }
}
```

Mathf.SmoothStep平滑插值

static function SmoothStep (from : float, to : float, t : float) : float

和lerp类似，在最小和最大值之间的插值，并在限制处渐入渐出。

```
public class example : MonoBehaviour {
    public float minimum = 10.0F;
    public float maximum = 20.0F;
    void Update() {
        transform.position = new Vector3(Mathf.SmoothStep(minimum, maximum,
        Time.time), 0, 0);
    }
}
```

Mathf.Sqrt平方根

static function Sqrt (f : float) : float



计算并返回 f 的平方根。

关闭

Mathf.Tan正切

static function Tan (f : float) : float

计算并返回以弧度为单位 f 指定角度的正切值。

分享到： 

[复杂材质烘焙成一张贴图](#) | [保护你的私密文件 for NTFS](#)

• 2012-09-17 10:27

- 浏览 52689
- [评论\(0\)](#)
- 分类:[编程语言](#)
- [相关推荐](#)

参考知识库



[语音识别与合成知识库](#) 521 关注 / 316 收录



[计算机视觉知识库](#) 1257 关注 / 251 收录



[自然语言理解和处理知识库](#) 464 关注 / 87 收录



[知识工程知识库](#) 150 关注 / 74 收录

评论

发表评论



[您还没有登录,请您登录后再发表评论](#)



zhboy666666

- 浏览: 147955 次
- 性别:
- 来自: 大连
- 我现在离线

关闭

最近访客 [更多访客>>](#)



[jacktian1973](#)



[holtek452301](#)[u010050918](#)[wangweibin](#)

文章分类

- [全部博客 \(65\)](#)
- [cocos2d \(17\)](#)
- [XCode \(16\)](#)
- [objective-c \(3\)](#)
- [Unity3D \(15\)](#)
- [IOS \(2\)](#)
- [VS MayaAPI \(1\)](#)
- [OGRE \(1\)](#)
- [VC++ \(1\)](#)

社区版块

- [我的资讯 \(0\)](#)
- [我的论坛 \(0\)](#)
- [我的问答 \(0\)](#)

存档分类

- [2013-05 \(1\)](#)
- [2013-03 \(1\)](#)
- [2012-09 \(4\)](#)
- [更多存档...](#)

最新评论

声明：ITeye文章版权属于作者，受法律保护。没有作者书面许可不得转载。若作者同意转载，必须以超链接形式标明文章原始出处和作者。

© 2003-2017 ITeye.com. All rights reserved.

网安备110105010620]

关闭