

配合「[导航窗口](#)」查阅本文档体验最佳。

## UI: singleton

集合了 UI 的一系列缓动

### doArea2(wnd, to, duration)

使用缓动的方式，改变一个窗口的尺寸或大小

- params :

参数名	参数类型	说明
wnd	Window	引擎的窗口实例
to	Area2	这是缓动系统封装的类型，用于表示窗体的大小和尺寸
duration	number	缓动持续时间

- return :

TweenerOperator

- example :

```
--为了方便调用，本接口做了不少封装，效果如下

--改变位置
--这代表 x 轴和 y 轴都移动到相对坐标 100% 的位置
UI:doArea2(wnd, {x = {1, 0}, y = {1, 0}}, 0.5)
--改变绝对坐标
UI:doArea2(wnd, {x = {0, 100}, y = {0, 100}}, 0.5)
--当然也可以只填一项，另外的项会取窗口当前的值
UI:doArea2(wnd, {x = {1,0}}, 0.5)

--改变尺寸
UI:doArea2(wnd, {width = {1, 0}, height = {1,0}}, 0.5)

--同时改变位置和尺寸
UI:doArea2(wnd, {
    x = {0, 0},
    y = {0, 0},
    width = {1, 0},
    height = {1, 0}
}, 0.5)
```

```
-- to 参数可以传入的项有

-- x = {0, 0}
-- y = {0, 0}
-- width = {0, 0}
-- height = {0, 0}

-- 这些项可以选填，不填的项，默认取当前值
```

## doFade(wnd, to, duration)

以缓动的形式改变透明度

- params :

参数名	参数类型	说明
wnd	Window	引擎的窗口实例
to	Area2	透明度终点值
duration	number	缓动持续时间

- return :

TweenerOperator

- example :

```
--窗体的透明度在 0.5 秒内以缓动的形式变为 1
UI:doFade(wnd, 1, 0.5)
```

---

## Object: singleton

集合了对象的一系列缓动接口

## doMove(to, duration)

以缓动的行为移动位置

- params

参数名	参数类型	说明
to	Vector3	位置的终点值
duration	number	缓动持续时间

- return :

TweenerOperator

- example :

```
player.doMove(Vec3.(0,2,0), 1)
```

---

## LoopType : enum

**全局实例** 缓动的循环类型

### Restart

每一次循环，重新开始播放

### Yoyo

每一次循环，动画会自动反转，像悠悠球荡来荡去

---

## EaseType : enum

**全局实例** 缓动类型

该类型实在太多，这里不一一列举，但是该枚举中提供了所有国际通用的缓动类型，可以上该网站查看类型以及效果：[缓动函数速查表\(easings.net\)](http://easings.net) `Ctrl` + `鼠标左键` 点击链接

还有一些：

### linear

线性动画，一条直线。

---

## TweenerOperator : class

对 Tweener 进行参数设置

**注意：**

本类的大部分方法都会返回本身实例，方便修改参数，但是有一部分并不返回

例如：`playCancel()`，`pause()` ... 因为这些是操作型参数，用户应该保存实例然后再调用

### delay(time)

延时启动**整个**动画。

这里要注意，是整个动画，包括设置了循环之后的动画。

- **params :**

参数名	参数类型	说明
time	number	延时启动的时间，秒为单位。

- **return :**

`TweenerOperator`

### from(value)

表示本次缓动从哪个值开始。这个值的类型和你要操作的值保持一致。

例如：

如果你调用 `doFade()` 接口，你改变的是透明度，那就传入透明度的值即为：`number`

如果你调用 `doMove()` 接口，你改变的是位置，那就传入位置的值即为：`Vector3 / Vector2`

- **params :**

参数名	参数类型	说明
value	any	表示缓动从哪个值开始

- **return :**

`TweenerOperator`

### reverses()

反转本次缓动

**注意：**该接口目前有缺陷，不是真正意义上的反转，只是把起始值和终点值调换了

### setAutoKill(enable)

设置自动销毁，代表动画完成会自动销毁，默认为 true

- **params**

参数名	参数类型	说明
enable	boolean	是否自动销毁

- **return :**

`TweenerOperator`

### setLoops(count, loopType)

设置本次缓动循环

- **params :**

参数名	参数类型	说明
count	number	循环次数，默认为 1
loopType	LoopType	循环类型，默认为 Yoyo

- **return :**

TweenerOperator

- **example :**

```
UI:doFade(wnd, 1, 0.5)
    :setLoops(1, LoopType.Yoyo)
```

## setEase(ease)

设置缓动类型

- **params :**

参数名	参数类型	说明
ease	EaseType	缓动类型，默认为 outExpo

- **return :**

TweenerOperator

- **example :**

```
UI:doFade(wnd, 1, 0.5)
    :setEase(EaseType.linear)
```

## go()

跳过延迟马上启动

## pause()

暂停缓动

## continue()

继续缓动

## restart()

重新开始缓动

## reset()

重置缓动，会将位置设置为起始位置，并且暂停

可以调用 `continue()` 来继续

## destoty()

销毁该缓动

## playCancel()

取消缓动，动画未播放完成时，以缓动的形式回到起始点

之后可以调用 `restart()` 重新开始缓动

## onStart(callBack)

设置一个回调函数，在缓动开始时调用

- **params**

参数名	参数类型	说明
callBack	function	回调函数

- **return :**

`TweenerOperator`

## onFinish(callBack)

设置一个回调函数，在缓动结束时调用

- **params**

参数名	参数类型	说明
callBack	function	回调函数

- **return :**

`TweenerOperator`

## next

这是一个字段，可以设置本次缓动完成后的下一次缓动

可写不可读

- **type :**

TweenerOperator

- **example :**

```
UI:doArea2(wnd, {x = {0,0}}, 0.5)
.next =
UI:doArea2(wnd, {width = {1.1, 0}}, 0.2)
.next =
UI:doArea2(wnd, {width = {1, 0}}, 0.2)
```