

Tween

这是一个用于物理缓动的插件，能够作用于任何在视觉上有物理运动的东西。

例如：UI、实体 / Entity ...

使用方式

方便、简单的调用接口

下面我们以 UI 的物理缓动为例，介绍一下系统的使用方式。



如上图这个动画，嗯，效果非常好。

那么怎么达到这个效果呢？只需要一行代码。

```
UI.doArea2(page, {x = {0, 0}}, 0.5)
```

如果有使用过 Unity 的插件中大名鼎鼎的 DOTween，肯定会觉得熟悉，没错，这个插件的 **调用接口** 就是模仿它来的。

上面的接口的参数分别为：

page：对应一个窗体实例

{x = {0, 0}}：缓动的目标位置。在这里指该窗体的 x 轴指移动到相对坐标为 0，绝对坐标为 0 的位置。

0.5：表示本次缓动的持续时间，单位：秒。

丰富的参数

如果有的时候我希望在动画结束后，播放某个事件，但是接口里似乎并没有？

不用担心，开发者早就考虑到这一点，为了尽可能使接口简洁，所有非必要的参数都封装到另一个地方。

使用方式：

```
UI:doArea2(page, {x = {0, 0}}, 0.5)
    :onFinish(function()
        print("UI动画结束了。")
    end)
```

`doArea2` 接口会返回一个 `TweenerOperator` 类型的实例，通过调用这个实例的方法，可以做非常多的事。

除了在动画结束时注册事件，还有这些：

- `delay()` 延时播放
- `reverses()` 反转播放
- `setEase()` 设置缓动类型
- `setLoop()` 设置循环
- `setAutoKill()` 设置自动销毁
- ...

详情请参考《API参考手册》或 源代码

然后你就可以这样用：

```
UI:doArea2(page, {x = {0,0}}, 0.5)
    :setEase(EaseType.outExpo)
    :setLoop(5, LoopType.Yoyo)
    :delay(2)
    :onStart(function()
        print("UI 动画开始了。")
    end)
    :onFinish(function()
        print("UI 动画结束了")
    end)
```

当然，不是所有的接口都可以这样玩，一些 **操作性接口** 是不会返回自身的，也就是不能像上面这样一层一层调用，例如重置接口、暂停接口。

不这样做的原因是这些接口的使用应该保存 `operator` 实例来调用，而上面展示的接口都是在缓动开始之前设置参数。

--操作性接口的正确使用方法。

```
local operator = UI:doArea2(page, {x = {0,0}}, 0.5)
```

--在缓动运行0.2秒后暂停。

```
world.Timer(0.2 * 20, function()
    operator:pause()
end)
```

