

# OAC Devkit

Version: 4.1

[PDF Document](#)

## 功能

- 平滑的效果，平滑的视觉。
- 灵活的配置和组合。
- 变形过程中，可以随时改变方向。

## 快速开始。按照这个，一步一步

1. 准备您的物品
2. 根据自己的需要，选择以".OAC"开头的配置文件，修改其参数，拖入清单。
3. 将名为OAC.KERNEL 的主脚本拖到清单中。
4. 选择您需要的触发脚本，将其拖放到对象中。 "Extra"中已经为您预设了一些触发脚本。 当然，您可以根据需要自定义它们。
5. 完成。

## 做一个简单的推拉门



创建一个盒子，像门一样      选择您需要功能的配置文件，将它们拖放到目录中



改变参数

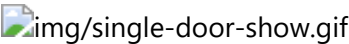
X方向移动2米

持续时间2秒

使用 ease-in-out 效果功能

拖放脚本

## 点击看效果



更详细的例子，在"Example"目录中，放出来进行测试和编辑

## 包含脚本

名称	描述
OAC.KERNEL	(必要的) 主脚本

### Extra

名称	描述
----	----

名称	描述
TouchToggle	使 prim 可点击，触发并切换往复运动，它只会触发当前 prim(LINK_THIS)。
TouchToggleSync	使 prim 可点击，触发并切换往复运动，它会触发(LINK_SET)中所有prim，通常用在根prim。
AutoClose 30s	打开30秒后自动关闭。
AutoToggle after end 20s	转换结束后，等待20秒切换状态，循环。
AgentSensorOpen	附近有人时打开。
AgentSensorToggle	附近有人时打开，无人时关闭。
SoundTrigger	运行过程中播放声音，此脚本预设为电动门，可任意更换。

配置

一个notecard(记事卡)代表一个配置字段，拖拽到内容栏，编辑它名字以修改参数。

格式: .OAC {关键字} {值}

关键字	类型	取值	默认	描述	版本
BROADCAST2	integer	大于 -5 且不 为 0	-4	广播发送范围，-4:LINK_THIS, -3:LINK_ALL_CHILDREN, -2:LINK_ALL_OTHERS, -1:LINK_SET, 1:LINK_ROOT, 和其它	3.3
DURATION	float	任何	0.0	时长，如果小于0.1，则视为0.0， 0.0表示没有运动过程，瞬间完成	1.7
DISTANCE	vector	任何	<0.0,0.0,0.0>	距离，移动变化	4.0
ROTATION	vector	任何	<0.0,0.0,0.0>	旋转，旋转变化，这个向量的含义是 <ROLL, PITCH, YAW>。 * 旋转总是相对于prim的局部(local)方向 向量。	1.8
SCALE	vector	大于 <0.0,0.0,0.0>	<1.0,1.0,1.0>	缩放，缩放变化，不可出现负值，如果 等于ZERO_VECTOR (<0.0,0.0,0.0>)， 则视为无效的	3.0
ORIGIN	integer	0/1/2	0	参照物，见下方特别说明	2.0
TIMING_FUNC	integer	0/1/2/3	0	过渡效果，见下方特别说明	2.0
QUEUE	string			Queue模式，详见下文	3.0

DISTANCE 特殊用法

4.0版本之后 DISTANCE 的取值添加了相对于物体尺寸的选项，支持后缀 x,y,z,X,Y,Z。

- $x, y, z$ : 本prim的尺寸
- $X, Y, Z$ : root prim的尺寸

```
DISTANCE <1.2x, 2X, 0.5z> // 沿 x 方向运动 1.2 倍的 当前 prim 尺寸 x, 沿 y 方向运动 2 倍 root prim 尺寸 x, 沿 z 方向运动 0.5 倍 当前 prim 的 尺寸 z。
```

### 用例子来说明

1. 有一扇滑动开关的门，它的宽度是  $x$ ，高度是  $z$ ，厚度是  $y$ 。开启这扇门需要沿着  $x$  轴移动 0.8 倍门的宽度，如以下写法：

```
DISTANCE <0.8x, 0, 0>
```

2. 一个可以缩放的滑块，我们无法确定它的尺寸，所以更没法确定它移动的具体的距离，只知道它会沿着  $z$  轴升起 root prim 尺寸  $y$  2 倍的高度。如以下写法：

```
DISTANCE <0, 0, 2Y>
```

### 关于 参照物 ORIGIN

#### 局部(local) (0)

运动方向将参考局部(local)方向向量。

例子:

```
.OAC DISTANCE <1.0, 0.0, 0.0>  
.OAC ORIGIN 0
```



#### 根prim(root) (1)

运动方向将参考根prim的全局方向向量。

例子:

```
.OAC DISTANCE <1.0, 0.0, 0.0>  
.OAC ORIGIN 1
```



它仅适用于链接集中的子 prim。当对象是根 prim 或者它是一个独立的 prim 时，视为全局。

全局(world) (2)

转换将参考全局(world)方向向量。

例子:

```
.OAC DISTANCE <1.0, 0.0, 0.0>
.OAC ORIGIN 2
```



关于 过度效果 TIMING\_FUNC

0: linear 线性	1: ease-in-out 缓入/出	2: ease-in 缓入	3: ease-out 缓出
img/timing-func-0.png	img/timing-func-1.png	img/timing-func-2.png	img/timing-func-3.png
.OAC TIMING_FUNC 0	.OAC TIMING_FUNC 1	.OAC TIMING_FUNC 2	.OAC TIMING_FUNC 3

since 3.2

两个特殊值，正向移动与反向移动是对称的。

比如正向是 ease-in，那么反向自动切换为 ease-out。

102: ease-in 缓入(反向自动翻转为 ease-out 缓出)	103: ease-out 缓出(反向自动翻转为 ease-in 缓入)
.OAC TIMING_FUNC 102	.OAC TIMING_FUNC 103

Queue 模式

在 3.0 版本中新增Queue模式，它可以连续演绎多个变化过程（正向、反向），并且延续了任意时间点随时切换方向的特性。

```
.OAC QUEUE {编号}/{时长}/{参照}/{时间函数}/{距离}/{旋转}/{缩放}
```

是的，它将以前所支持的参数写在一行，并赋予给QUEUE，然后，您可以添加多个QUEUE。

{编号}代表了QUEUE顺序，在PRIM的内容里，文件是按照文件名升序顺序排列的，所以只要能保证顺序的正确，编号可以随意指定，无论是 1234... 或者 ABCD...。

如果两个QUEUE中需要等待，可以加入一个只带有时长的QUEUE，像下面这样：

```
.OAC QUEUE 1/5.0///<10.0,0.0,0.0>//
.OAC QUEUE 2/2.0/////
```

```
.OAC QUEUE 3/5.0///<0.0,10.0,0.0>//
```

## 本地消息接口

### 本地控制与数据提交

Num: **802840**

#### 开/正向变换

正向移动/变换

```
llMessageLinked(..., 802840, "OPEN", "");
```

#### 关/反向变换

反向移动/变换

```
llMessageLinked(..., 802840, "CLOSE", "");
```

#### 正反向切换

切换当前移动/变换方向

```
llMessageLinked(..., 802840, "TOGGLE", "");
```

#### 设置方向值

手动提交并改变当前所处运行方向状态

值: 大于 0: 设置为打开（待关闭）状态，此时可以执行关闭（逆向变换） 小于等于 0: 设置为关闭（待打开）状态，此时可以执行打开（正向变换）

```
llMessageLinked(..., 802840, "DIRECTION|1", "");  
llMessageLinked(..., 802840, "DIRECTION|-1", "");
```

#### 设置全局缩放

作用于 DISTANCE，子PRIM在有缩放状态下的移动距离倍率。

默认: 1.0，如果给予的值 <0，则使用默认。

```
llMessageLinked(..., 802840, "SCALE|1.0", "");
```

重载

手动提交以重载脚本（重新读取所有配置参数）

```
llMessageLinked(..., 802840, "RELOAD", "");
```

本地事件广播

Num: **802841**

变换开始

发送至: **BROADCAST2**指定, 默认 -4:**LINK\_THIS**

```
TRANSFORM_STARTED|{方向}
```

方向:

- 1: 开, 正向变换(Queue)
- -1: 关, 逆向变换(Queue)

变换结束

发送至: **BROADCAST2**指定, 默认 -4:**LINK\_THIS**

```
TRANSFORM_FINISHED|{方向}
```

方向:

- 1: 开, 正向变换(Queue)
- -1: 关, 逆向变换(Queue)

变换中 (Queue 模式)

发送至: **BROADCAST2**指定, 默认 -4:**LINK\_THIS**

```
TRANSFORM_PROCESS|{方向}|{队列编号}|{有效性}
```

方向:

- 1: 开, 正向变换(Queue)
- -1: 关, 逆向变换(Queue)

有效性:

- 0: 如果 DISTANCE、ROTATION、SCALE 均无变化
- 1: 如果 DISTANCE、ROTATION、SCALE 任意一项有变化

## 本地 LinksetData 触发器

OAC.KERNEL 会监听 `LINKSETDATA_UPDATE`

name: (string)llGetLinkKey(LINK\_ROOT) + "-oac-stat"

- 当 value 为 **偶数** (**0 [2 4 6...]**) 时触发 **CLOSE**
- 当 value 为 **奇数** (**1 [3 5 7...]**) 时触发 **OPEN**

```
llLinksetDataWrite((string)llGetLinkKey(LINK_ROOT) + "-oac-stat", "1"); // OPEN
llLinksetDataWrite((string)llGetLinkKey(LINK_ROOT) + "-oac-stat", "0"); // CLOSE

llLinksetDataWrite((string)llGetLinkKey(LINK_ROOT) + "-oac-stat", "2"); // CLOSE
llLinksetDataWrite((string)llGetLinkKey(LINK_ROOT) + "-oac-stat", "3"); // OPEN
```