## 概述

### 插件介绍

参数条的有许多作用范围，菜单界面、战斗界面、地图界面都可以作用到：

◆Drill\_CoreOfGaugeMeter 系统 - 参数条核心

以下是参数条的子插件：

◆Drill\_GaugeOfBufferTimeBar UI - 缓冲时间条

◆Drill\_GaugeForBoss.js UI – 高级BOSS生命固定框

◆Drill\_GaugeForVariable.js UI – 高级变量固定框

**核心只对参数条的各个部件进行详细说明，如果你想找”从零开始设计”的教程，去看看子插件相关的文档说明**。

### 名词索引

以下你可以按住ctrl键点击下面的词，可以直接定位到想了解的名词：

|  |  |
| --- | --- |
| 基本结构 | [参数条](#参数条) [遮罩](#遮罩) [整体旋转角度](#整体旋转角度) |
| 段 | [段](#段) [段上限](#段上限) [缩短方式](#缩短方式) |
| 段 – 多段结构 | [段数](#段数) [上段](#上段) [下段](#下段) [段循环](#段循环) |
| 段 – 流动效果 | [头段](#头段) [过渡段](#过渡段) [尾段](#尾段) [段长度](#段长度) |
| 凹槽条 | [凹槽条](#凹槽条) [段阻塞](#段阻塞) |
| 弹出条 | [弹出条](#弹出条) [弹出条弹道](#弹出条弹道) |
| 粒子 | [粒子](#粒子) |
| 游标 | [游标](#游标) [多段复位](#多段复位) |
| 加满动画 | [加满动画](#加满动画) |

### 插件关系

参数条核心的关系如下图所示：



## 主体

### 定义

**参数条：**指用于实时显示某些参数数值的**整体结构**。比如实时显示生命值、魔法值、时间、变量值等参数。参数条是贴图。

参数条的结构如下图所示：



### 基本结构

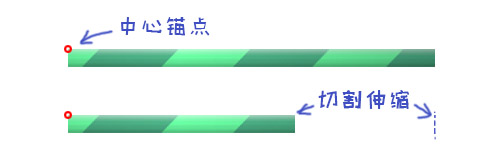
参数条有下列固定且硬性结构：

1.只能为长方形。（环形非常难实现）

2.只有一个裸露的条，没有外框。（子插件会提供背景框架配置）

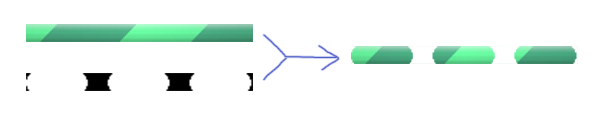
3.锚点在左上角，只能 从左往右 伸缩，且伸缩的方式为图片切割。

4.参数条的像素大小与资源图片像素大小紧密相关，且参数条 不能 缩放。（你需要根据参数条的大小来适配外框）



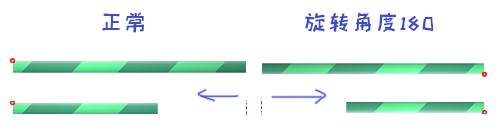
你可以控制4个的基本参数：**X坐标, Y坐标, 旋转角度, 遮罩**

**遮罩：**在上述固定结构的基础上，你可以使用遮罩做成平行四边形或圆角矩形。你也可以设置一个正方形的条，加上圆遮罩，制作出一个球状条：





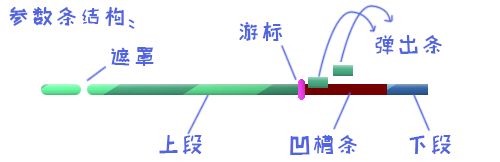
**旋转角度：**你可以修改旋转角度使其看起来为 从右向左 或 从下往上 的伸缩结构：（还可以做垂直的结构、45度角斜向结构等）



需要注意的是，最好在其它参数全部配置好之后，再进行旋转。因为有一定旋转角度的条，对你配置数值预估会有一定影响偏差。

### 附加结构

在基本结构的基础上，参数条的还可以附带许多额外效果，如下图所示：



具体内容可以往后面章节查看，这里简单介绍一下：

**上段：**多段时，处在上方的 段。

**下段：**多段时，处在下方的 段。如果参数值小于 段上限，则下段是空的。

**遮罩：**能掩盖参数条，使其变成透明的部分。与ps的蒙版一样，黑色区域完全透明，白色区域完全不透明，灰色区域半透明。

**凹槽条：**处于上段与下段中间的条。当参数值（比如生命值）被打出空缺时，凹槽条不会立即扣除，停留一段时间后再缩短。

**弹出条：**受到伤害时，直接被切出一小段的用于动画效果的弹出块。

**游标：**用于指示当前参数所处位置的较醒目贴图，游标可以是gif。

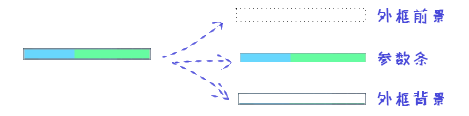
## 参数条与外框

### 组合关系

参数条主体是一个完全裸露的条，没有外框，而相关子插件会提供外框的设计：

**1) 2框+1参数条**

以 Drill\_GaugeOfBufferTimeBar UI-缓冲时间条 插件为例，

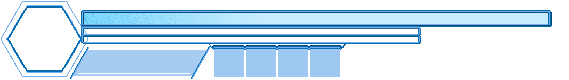


图中有外框前景和外框背景，分别在参数条的上方和下方。其中外框前景是空图片。

**2) 2框+6参数数字+3参数条**

以 Drill\_GaugeForBoss UI-高级BOSS生命固定框 插件为例：







图中有背景框和3个参数条：（前景框为空图片，图中参数数字都未显示）

生命条：段数8 + 流动 + 凹槽条 + 弹出条

魔法条：段数1 + 流动

怒气条：段数1 + 流动

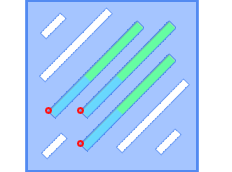
### 旋转关系

**分别配置**：由于参数条和外框的参数是分开的，如果参数条主体整体旋转了之后，外框是不会变化的。你需要另外控制外框的形状、或者旋转角度。

注意，时间条因为只有一个参数条，所以外框可以旋转，但是其它的就不一样了。

**多参数条位置分配**：多个斜向参数条与外框就不是上述的一对一那么简单的关系了，你要手动控制参数条所在的位置。下图中的外框，就不会提供旋转角度的功能，这需要你手动配置偏移，使得参数条嵌入外框。



（注意子插件中的参数条锚点说明，部分参数条的锚点不一定在左上角）

（参数条和外框之所以分离，就是为了适应上述可能多参数条组合的特殊情况）

## 段

### 定义

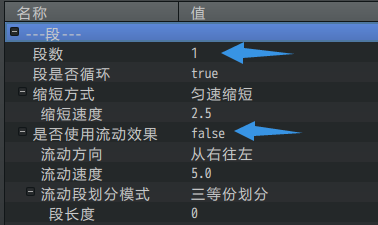
**段**：表示 参数条图片 被分割的贴图部分。**段是贴图**。（旧插件中叫”层级”）



由于旧插件中的定义比较模糊，且容易混淆。这里经过梳理，所有与”段”贴图相关的内容和定义，都有”段”字。

**最简单原始的段设置**：段数：1，流动效果：false 即可。

设置后资源图片大小就是段的大小，没有多段切割处理，也没有段长度划分处理。



### 段上限

**段上限**：表示 单段 能够容纳的最大参数值。（旧插件中叫”单层层值”）

**公式**：当前参数值 / 段上限 = 当前段长度 / 资源图片长度

最大生命值等于段上限，则显示单段。如果比段上限大，那么会显示多段。

以生命条为例：

段上限：200，生命值：200； 200/200 = 1。参数条为标准的1段。



段上限：100，生命值：200； 200/100 = 2。参数条会有2段。



段上限：150，生命值：200； 200/150 = 1余50。参数条会有2段，且上段只有三分之一。

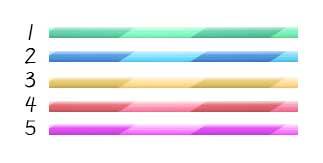


注意 参数和段 之间的关系，段是贴图，参数是数字。贴图包含高度和宽度，**你不能直接把参数值直接当成段长度，你需要参考之前的公式来进行换算。**

比如生命值256000000000000，满血状态的贴图宽度肯定不是这个宽度值，而是你配置的资源（比如300像素）那么宽。

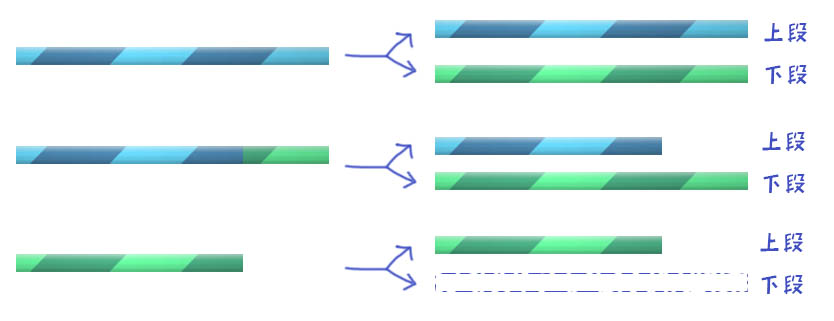
### 多段结构

**段数**：表示 参数条图片 被分割的数量，只能横向等比例分割。如下图所示，下图的段数为5，图片被分割成了5段。



**上段**：多段时，处在下方的 段。（旧插件中叫”上层条”）

**下段**：多段时，处在下方的 段。如果参数值小于 段上限，则下段是空的。（旧插件中叫”下层条”）



**段循环：**开启段循环后，分割的段贴图可以循环使用。

资源配置中，如果你设置了5段，而参数的值非常高，使得 参数值/段上限 要比5还要大，这时候，你可以开启段循环，使得第5段之后，又是第1段的结构。

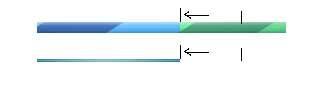


如果你没有开启段循环，那么 参数条 将一直保持最大值（最后一段全满）的状态。因为已经超出了能够显示的最大上限，所以只能保持最满状态。

### 缩短效果

缩短有两种方式，瞬间缩短 和 匀速缩短。

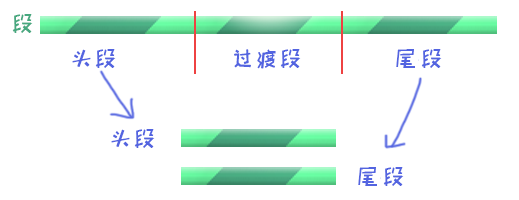
一般情况下，这缩短效果不明显。你需要关闭凹槽条和弹出条，才可能注意到，上段会根据当前参数值，有一个缩短/伸长的过程。这个过程不会延迟，变化即伸缩。



如果你使用了 凹槽条、弹出条 效果，建议使用瞬间缩短。因为如果匀速缩短的的速度太慢，会影响美观效果。

### 流动效果

流动效果开启后，段 会被划分为下面三个部分：（不是切割，而是划分）

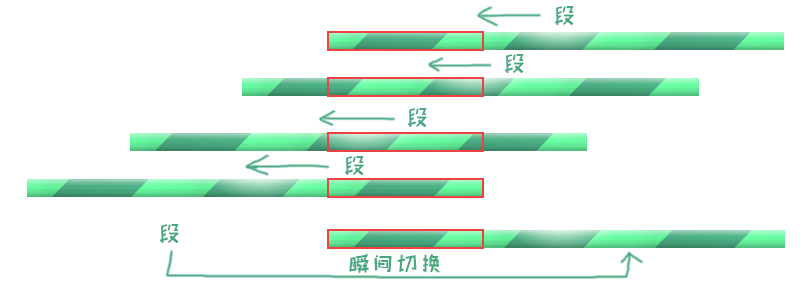


**头段**：参数条显示出来后，最先（第1帧）看见的部分。

**过渡段**：段在流动过程中，每一帧看见的部分。

**尾段**：循环流动至末尾的部分。只有尾段与头段**完全一样**，才能实现无缝循环流动。

流动效果的实际原理如下图：



**图中红色方框为游戏中实际看见的 段 部分**。并且，流动时，会有一道白光闪过，这道白光就是过渡段中画的亮光。另外，由图可知以下信息：

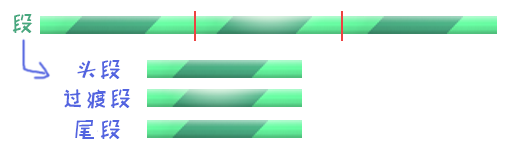
1. 头段和尾段 之所以必须一样，是为了实现无缝的瞬间切换。

2. 图中为 从右往左 流动，从左往右 流动原理一样，只是方向相反。

3. 流动效果开启时，**参数条在游戏中实际长度 = 头段的长度**。

**1）流动效果 - 三等分划分**

大部分与参数条相关的插件，都会默认：**流动参数条的长度是资源图片长度的三分之一**。



因为这样比较容易理解，头段长度 = 尾段长度 = 过渡段长度。如果我配置150像素宽的图片，那么实际就能得到50像素宽的 段 。

（旧文档中的解释：分成3等分后，图像从第1部分到第3部分时是连贯性的效果，框架到达第3部分时，会瞬间回归第1部分，进入新一轮的循环。）

**2）流动效果 -** **段长度划分**

实际上，过渡段可以为任意长度，甚至0长度（不能为负长度）。这取决于你设置的头段的段长度。你可以画一条超长的循环流动结构，然后再回归到最初的 头段。



注意，因为 段长度 = 头段的长度 = 尾段的长度，所以 段长度 必须大于或等于资源长度的一半。假设段长度设置为120，你可以配置资源图片的长度必须为240像素以上。



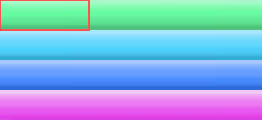
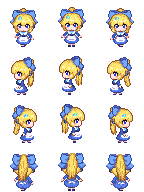
**3）流动效果 - 资源影响**

需要注意的是，如果你设置了“三等分划分”，那么**遮罩、凹槽条**的**资源长度**也应该设置为三分之一。如果设置的是“段长度划分”，那么遮罩、凹槽条的长度应该等于 段长度。

### 多段 与 流动

**1）类比**

上述概念可能比较复杂，这里其实不妨可以用行走图类比一下：

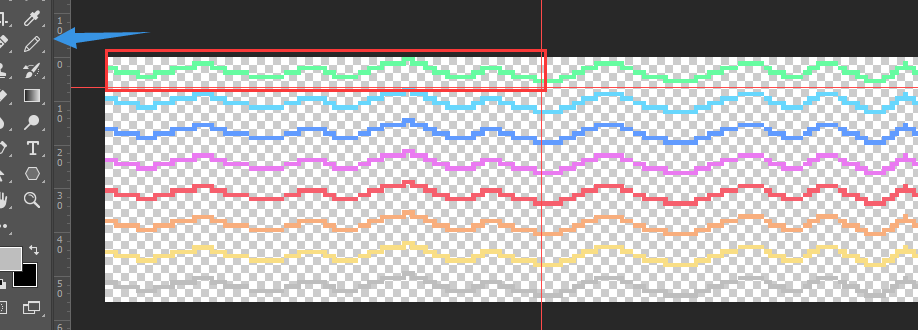
段 的多段 好比 行走图 的方向。

段 的流动 好比 行走图 的帧数。

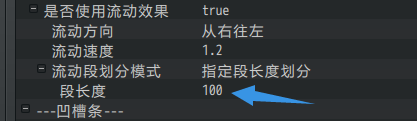
多段和流动互不干扰，所以不需要担心交互问题。不管是参数条还是行走图，在游戏中的实际大小，都是经过切割或规划后的大小，所以在配置时要考虑清楚。

**2）快速上手**

要画出 流动效果 并不难。举个例子，就拿ps的铅笔画波形。如下图中，简单画一个长长的随意的波形。



既然 头段 的长度是关键，那么我一开始就预先设置段长度100像素：

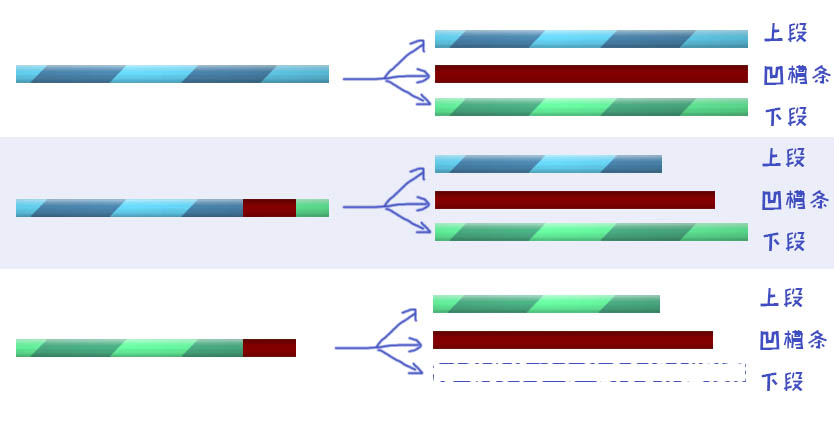


这样，中间段就可以随意画了，想画多长画多长，画完了把 头段 复制一下，放在最后作为尾段，就完成啦。这样配置参数条，显示都是固定100像素。不需要考虑资源长度。

## 凹槽条

### 定义

**凹槽条**：处于上段与下段中间的条。当参数值（比如生命值）被打出空缺时，凹槽条不会立即扣除而留下红印，停留一段时间后再缩短。





凹槽条有延迟时间，一般情况下，连续收到攻击，凹槽条会一直留着，可以比较直观地看出你这次连续猛烈的攻击下，造成了什么程度的伤害。

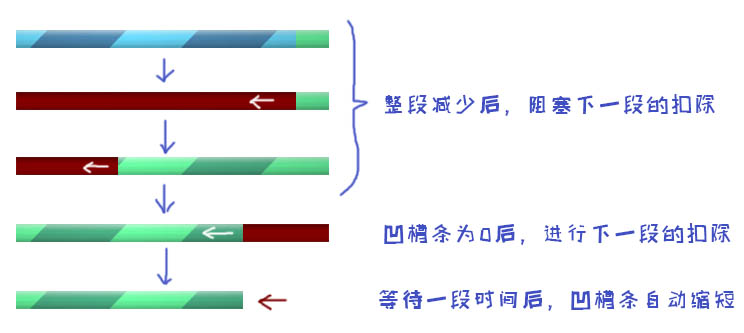
（注意，凹槽条 的图片配置资源，要考虑是否开启了流动效果。流动开启时，只需要配置与参数条资源的三分之一的长度。）

另外，凹槽条并不是在所有地方添加都合适，如果参数值的减少/扣除是持续不断的（比如时间条），那么凹槽条反而会影响整个参数条的观感。

### 段 阻塞

多段时，如果一整段被你打掉了。**凹槽条会立即结束延迟，开始缩短，直到完全缩短为0后，再从下一段开始重新计算凹槽条缩短**。

缩短过程如下图所示：



整段缩短期间，下一段不会显示扣除的生命，所以一般情况下建议凹槽条的缩短速度不要设置的太慢。

## 弹出条

### 定义

**弹出条**：参数值减少时，上段会切掉减少的部分，形成弹出条，用于播放 段的扣除动画效果。

（下图中白色的为弹出条，样式中可以设置弹出条块模式，图中设置为白色块；）

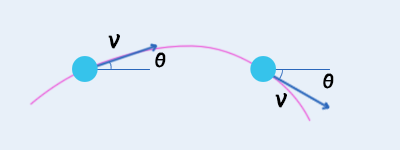
（红色的圆圈为游标，显示方式为 受伤时显示 。）



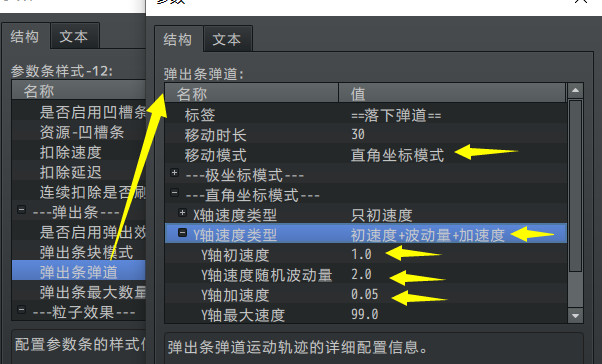
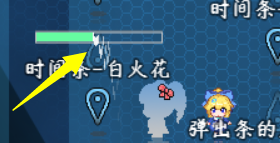
**不支持遮罩**：需要注意的是，弹出条是不会被遮罩挡住的。如果你的遮罩遮挡后是一个不规则形状，那么弹出条弹出的形状仍然为长方形不变，可能会影响视觉效果。

### 弹出条弹道

弹道的具体配置说明，去看看”关于弹道.docx”。



弹出条的轨迹不一定必须是抛物线，不移动、只向下移动、扩散移动都可以做出非常好的动画效果。具体可以去可视化管理层看看，火花事件。



## 粒子

### 定义

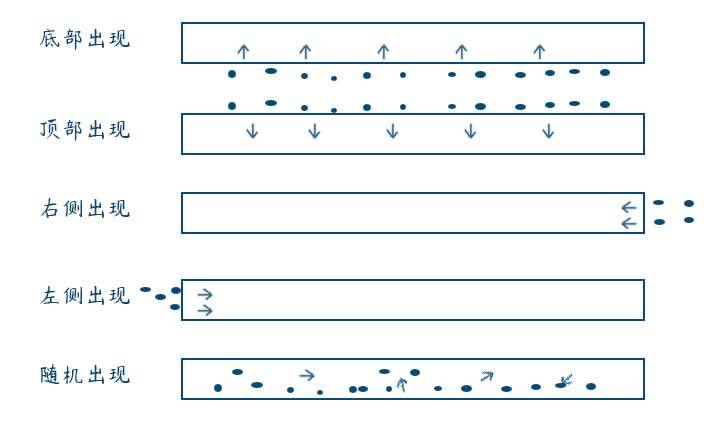
**粒子**：是只在参数条内部冒出的粒子效果。

常见的有生命条中的云雾效果、生命罐中的气泡效果等。



### 出现方式

下图为粒子的出现模式：



**需要注意的是，xy速度要根据你的出现位置进行调整**。

底部出现，y速度使用负数。

顶部出现，y速度使用正数。

右侧出现，x速度使用负数。

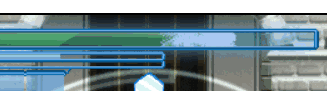
左侧出现，x速度使用正数。

随机出现，x、y速度可以设置0。

**另外，粒子出现与消失，与 段 的伸缩有联系。**

空的部分，将不会出现粒子，并且，由于出现的范围越来越短，20个粒子会越来越密集地挤在一起。

随机分布缩短时：

右侧出现缩短时：

### 仿粒子效果

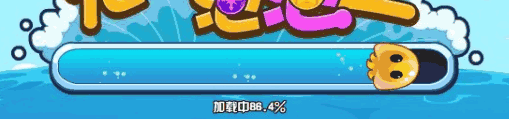
由于粒子方向随机，是会乱跑的，反而实际效果没有理想那么均匀。

这时候，你可以换一种思路，**把颗粒直接画在参数条图片中，并开启流动效果**。因为流动效果相当于平移图片，而这里面的颗粒就是完全均匀的，并且无限循环。

## 游标

### 定义

**游标**：是跟随当前条进度移动的一个贴图。可以是单张贴图，也可以是gif贴图。比如图中的黄色章鱼为游标。

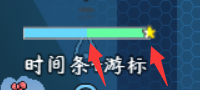
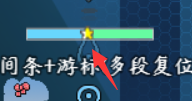


再比如图中的白色亮光，也是游标。



**多段复位**：游标会根据加满的情况浮动，默认情况下，只要 单段 加满了，游标则会一直处于满状态。多段情况下，需要开启多段复位，实现游标复位。

（左图为默认情况，右图开启了多段复位功能。）

另外，如果启用了多段复位，游标在满格时，会因为正好能整除，且余数为0，而造成游标回到起点位置。（比如 段上限135，参数值正好270，游标此时会处在起点位置）

这时候，你需要稍微修改一下 段上限 或 参数值，让其余数不为0。（比如修改段上限设为134，或者参数值设为269）

**遮罩遮挡**：默认游标处在最上方，如果你希望游标像 段 一样，嵌入在框内，并且能被遮罩遮挡，可以开启遮罩遮挡功能。



### 显示方式

游标有许多特殊的显示方式：

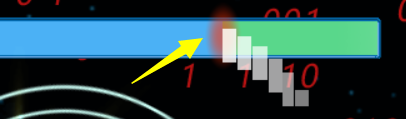
**光亮模式**：游标在条的两侧时，透明度最低，中间的透明度最高。

（有时候游标就是一个圆形的渐变光，最经典的例子，就是win7配置界面）



**闪烁模式：**游标周期性地闪烁。

**受伤模式**：在数值降低时，突然显现游标，随后消失。



**增量模式**：在数值增加时，突然显现游标，随后消失。

**变化模式**：在数值增加/降低时，突然显现游标，随后消失。

**一直显示：**游标不做任何处理。

## 加满动画

### 定义

**加满动画**：是指参数条从无到有的一个动画过程。这里的加满是直接对整个参数条进行切割伸缩而控制的。与参数值没有关系。

（下图中，不管段数如何，都是从左往右填充的。）



部分子插件会屏蔽此功能，比如时间条，时间条是持续减少/增加的，不需要加满动画。