## 概述

### 基础要求

|  |  |
| --- | --- |
| 先尝试回答下面的问题：  什么是GIF文件和序列大图？  什么是GIF动画序列？  什么是状态元、状态节点、动作元？  是否已完成 ”配置GIF文件并在游戏中播放” 流程？ | 需要先了解基础知识哦！ |

如果你对上述问题有疑问，那么说明你还不了解 动画序列。

需要去看看：”小工具 > GIF动画序列编辑器” 中 入门篇+高级篇 的知识。

|  |
| --- |
| 小工具>入门篇 教你如何用小工具配置GIF，并转入到游戏中并播放。  小工具>高级篇 介绍更复杂的GIF配置，并在游戏中执行复杂播放。  本文档 总结一下动画序列的概念，并且介绍最初始的插件配置方法，  不是重要内容，了解即可。 |

### 插件介绍

核心插件：

◆Drill\_CoreOfActionSequence 系统 - GIF动画序列核心

相关子插件：

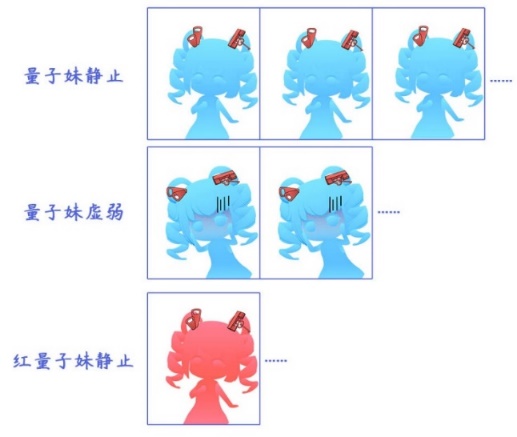
◆Drill\_ActorPortraitureExtend 战斗UI - 高级角色肖像

◆Drill\_PictureActionSequence 图片 - GIF动画序列

◆Drill\_EventActionSequence 行走图 - GIF动画序列

GIF动画序列可将多张png图片组合，可随时播放不同序列。

可以去看看：[思维导图](#_思维导图) 。



该插件含配置小工具：

GIF动画序列编辑器（DrillGIFActionSequenceEditor）

你可以直接点开GIF动画序列核心插件，配置里面的数据：

[直接配置 - 循环GIF](#_直接配置_-_循环GIF)

[直接配置 - 单次播放GIF](#_直接配置_-_单次播放GIF)

作为对比，这里也列出小工具的操作：

[小工具配置 - 循环GIF](#_小工具配置_-_循环GIF)

[小工具配置 - 单次播放GIF](#_小工具配置_-_单次播放GIF)

### 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：

|  |  |
| --- | --- |
| 动画序列 | [动画序列](#_动画序列) |
| 直接配置 | [小工具标记](#_1）小工具标记) [参数简介](#_2）参数简介) [默认的状态元集合](#_命名规则) |

### 插件关系

GIF动画序列核心的插件关系如下图。



### 思维导图

GIF动画序列核心结构的思维导图如下：（调整一下word右下角的缩放率，可以看清小字）



## 动画序列

### 定义

根据 ”小工具 > GIF动画序列编辑器” 中 入门篇+高级篇 的知识。

已经介绍了 **动画序列、状态元、状态节点、动作元、动画帧** 这些定义。

这里不重复介绍了，以免对知识点产生混淆。

## 直接配置

### 直接配置说明

#### 1）小工具标记

本文的 [基础要求](#_基础要求) 就要求走完小工具的“配置GIF文件并在游戏中播放”的流程。

走完流程后，你会发现在插件配置中，多出“【编辑器编辑】”的文本。



如果你不使用小工具，也可以直接对 GIF动画序列核心进行配置，

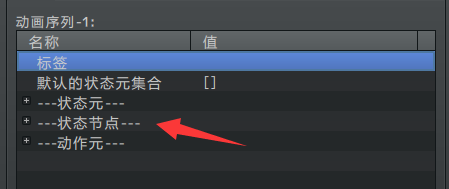
只不过点开后，“【编辑器编辑】”的文本会消失。



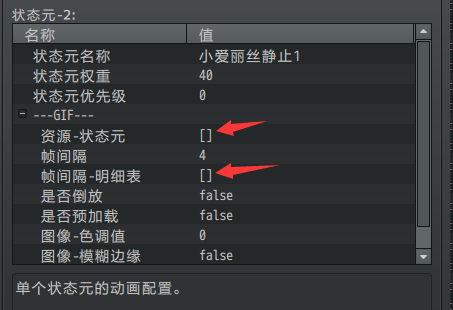
#### 2）参数简介

选择一个动画序列后，你可以看到下面的参数配置。

这些配置在小工具中也对应了 状态元、状态节点、动作元。



状态元下需要一张一张配置图片，并且要配置对应的帧间隔。

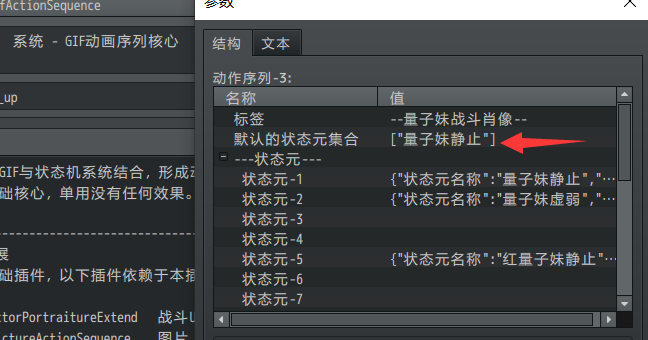


#### 3）默认的状态元集合

**默认的状态元集合**：指动画序列启用后，默认执行的状态元集合。

该集合至少需要配置一个状态元，不然在动画序列导入后，你将看不见任何图像。

注意，这里的配置，需要手填 状态元名称，如果名称文本对应不上，则没有效果。

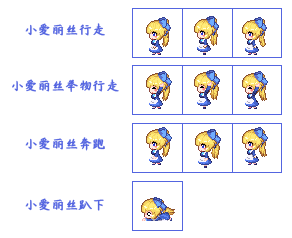


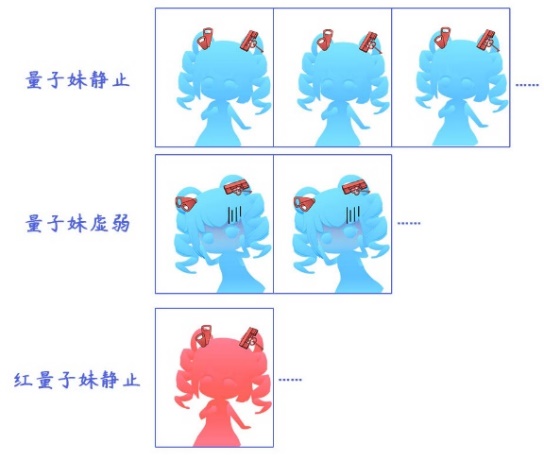
### 资源设置

#### 1）资源可大可小

资源的大小没有限制，可以是大的肖像，也可以是小像素行走图。

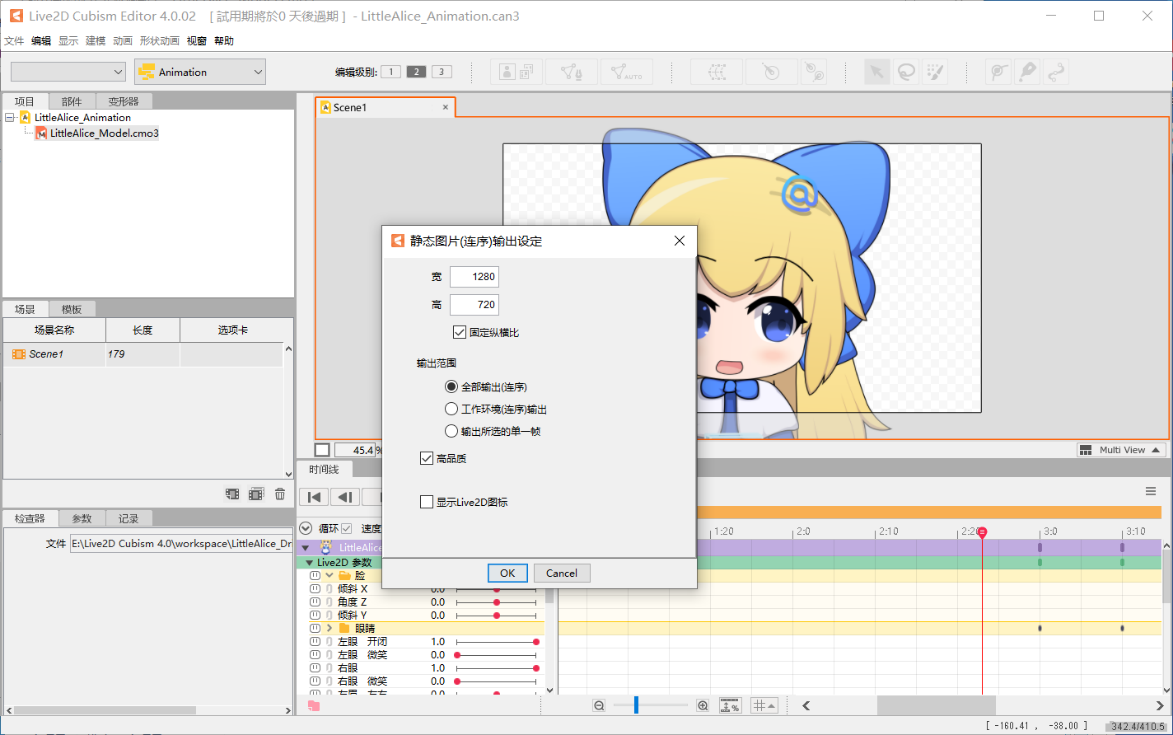
需要注意的是，所有帧都必须拆成单张图片，才能配置到动画序列。





#### 2）资源帧数不限

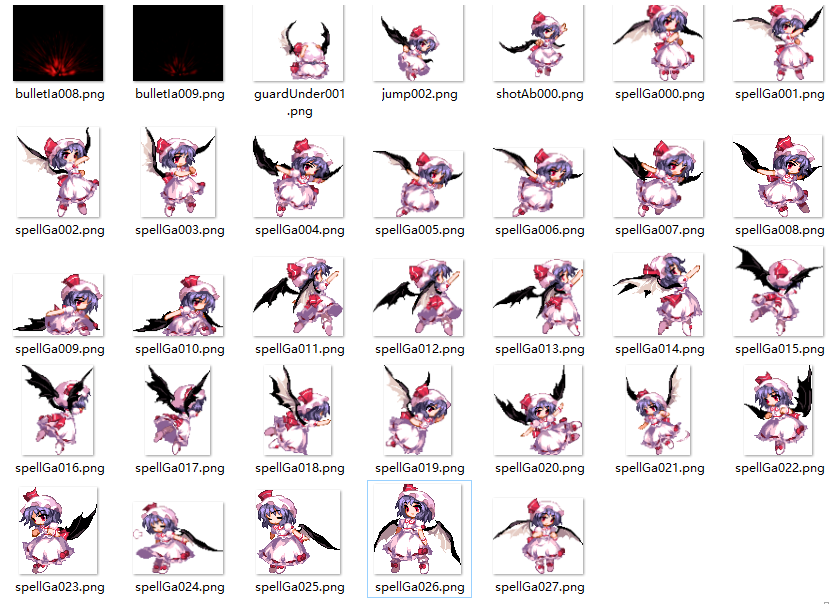
GIF动画序列的帧数不限，可以通过动态立绘软件（比如live2d）导出100张以上的图片，然后配置到 动作元或状态元 中。



#### 3）资源名不能重复

如果你是从外面找到的素材，最好将这些素材全部进行 **重新命名** ，因为后期这些素材将会全部放在 同一个 文件夹，必须保证这些文件名不会重名，否则会被覆盖。

（下图为东方非想天则的素材示意图……好像不小心截取到了威严满满的抱头蹲防。。）



## 从零开始设计（DIY）

你可以了解 直接配置，直接配置与小工具配置原理是一样的。

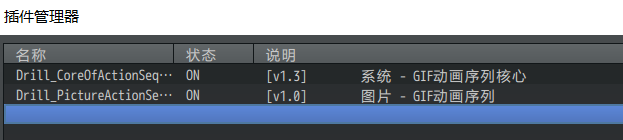
不过注意，后期主要以 小工具配置 为主，因为直接配置的内容太多，会疯掉的。

### 直接配置 - 循环GIF

#### 1）创建工程

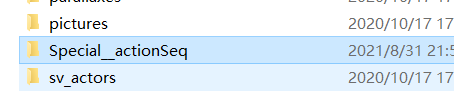
创建一个初始工程，加入两个动画序列插件。





#### 2）复制资源

将用到的资源png图片文件放入 Special\_\_actionSeq 文件夹中。

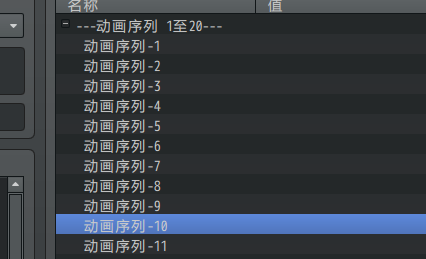


这里打算配置： 图片GIF-波浪点（20个图片）



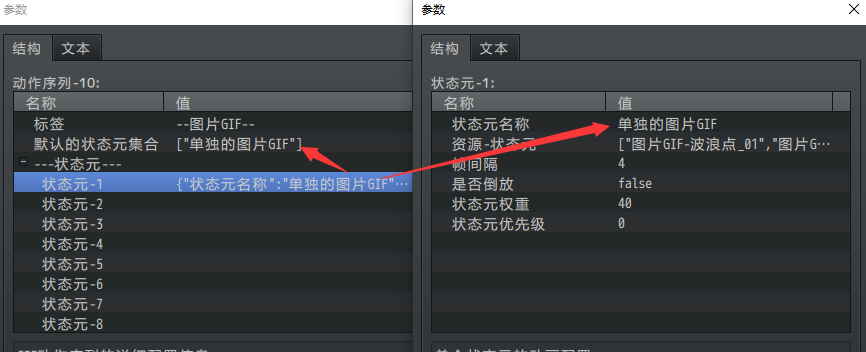
#### 3）配置动作序列

进入动画序列插件核心，选择一个动画序列。



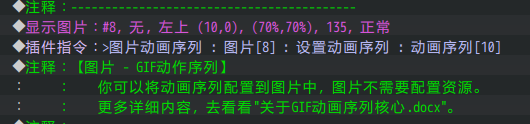
配置状态元即可，将20张图片都放入 资源-状态元 列表中。

注意，需要保证 默认状态节点 与 状态元名称 保持一致。



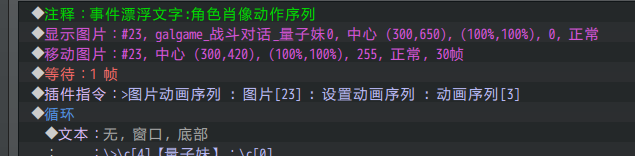
#### 4）插件指令绑定

配置后，用插件指令将 动画序列10 绑定到 图片8 上，就完成了。





**图片延迟问题：**有时候动画序列在图片建立后，不能立即设置动画序列，不然会使得动画序列找不到对象进行初始化。这时应该等1帧再设置。

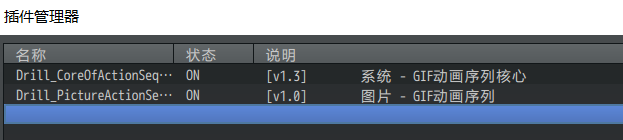


### 直接配置 - 单次播放GIF

#### 1）创建工程

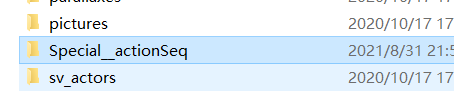
创建一个初始工程，加入两个动画序列插件。





#### 2）复制资源

将用到的资源png图片文件放入 Special\_\_actionSeq 文件夹中。

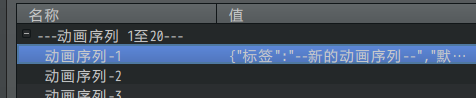


这里打算配置： 图片GIF-小闪烁（10个图片）



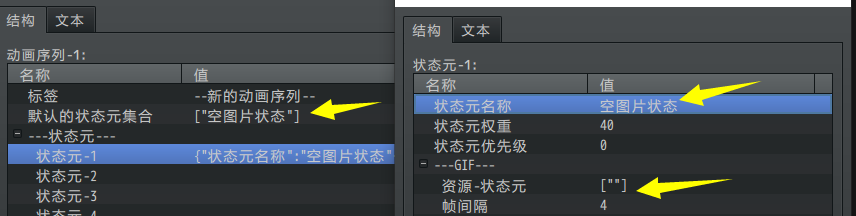
#### 3）配置动作序列

进入动画序列插件核心，选择一个动画序列。



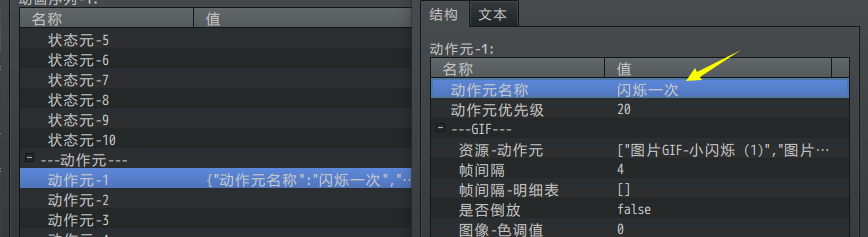
配置一个空的状态元名字为“空图片状态”。

注意，需要保证 默认状态节点 与 状态元名称 保持一致。



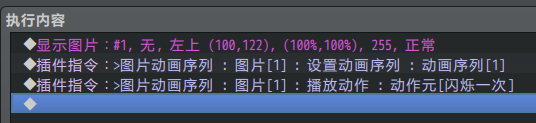
接下来，配置一个动作元，设置名字为“闪烁一次”。

将资源的10个图片，配置到 资源-动作元 列表中。

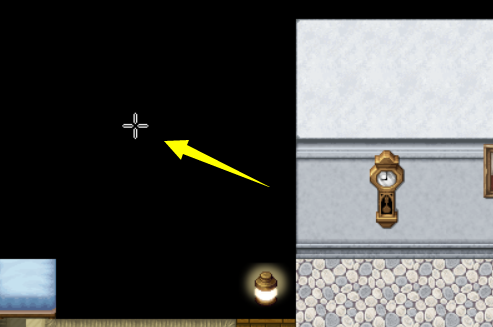


#### 4）插件指令绑定

建立一个事件，执行显示图片，然后 设置动画序列。最后播放动作。



可以看到，GIF的图片播放了一次后，就消失了。



#### 5）原理说明

动画序列中，**必须**要有默认状态节点，**必须**要有至少一个状态元。

状态元需要作为循环的对象，不停的循环播放GIF。

所以，需要给一个空资源图片的状态元。

这时候，再播放一次动作元，动作元播放完毕后，就又回到 空图片 循环的状态元了。

### 直接配置 - 完整动画序列

#### 1）配置关系

完整的配置就是将上述的两个配置结合起来。

一个动画序列中，可以配置很多 状态元、状态节点、动作元，是一个大的盒子。

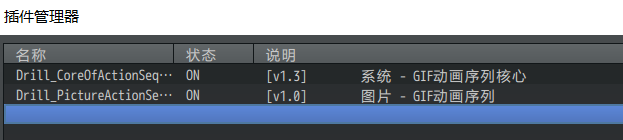
你需要将资源填充到盒子中，完成盒子配置后，再通过插件指令控制播放。



#### 2）创建工程

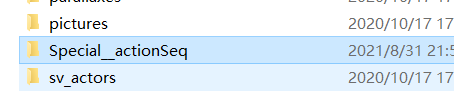
创建一个初始工程，加入两个动画序列插件。





#### 3）复制资源

将用到的资源png图片文件放入 Special\_\_actionSeq 文件夹中。

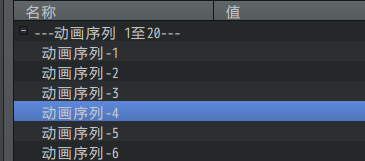


这里打算配置： 量子妹（8个图片，2个状态元，2个动作元）



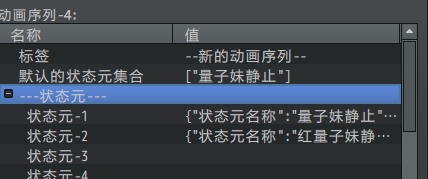
#### 4）配置动作序列

进入动画序列插件核心，选择一个动画序列。

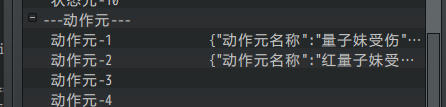


配置两个状态元：量子妹静止和 红量子妹静止

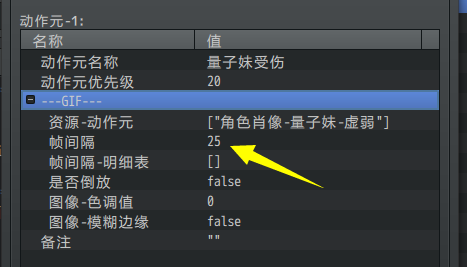
默认状态节点为 量子妹静止 。



配置两个动作元：量子妹受伤 和 红量子妹受伤

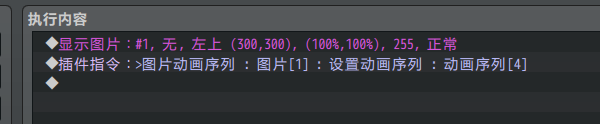


注意，配置动作元的帧间隔为25帧，拉长受伤图片的显示时间。



#### 5）插件指令绑定

建立一个事件，执行显示图片，然后 设置动画序列。



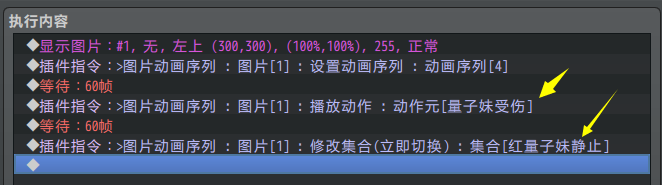
可以看到，绑定后，动画序列将按照默认的状态，循环播放。

（默认播放速度4帧，有点快，略显鬼畜）



随后，添加插件指令，可以控制播放受伤动作。

以及 修改状态节点，使其切换至红量子妹版本。



以上是完整的配置与调用方法。

#### 6）直接配置大量资源方法

如果状态元有非常多的图片，一个个配置非常麻烦。（比如从live2d中导出的大量图片）

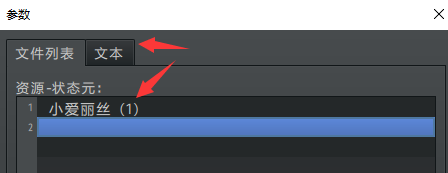
这里推荐使用小工具 动画序列GIF编辑器，可以进行批量导入。

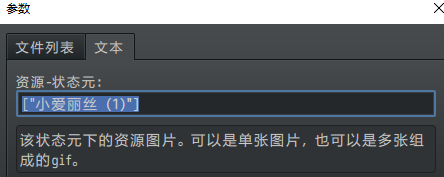
不过，直接导入也是可以的，

比如下图有45张小爱丽丝图片需要配置。



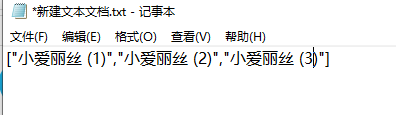
先配置一个”小爱丽丝 (1)”后，点击文本。



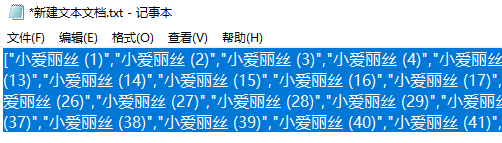


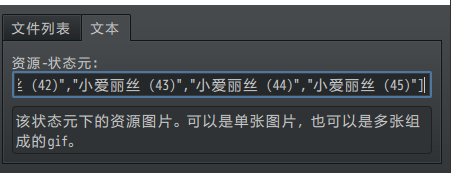
将文本复制到记事本中，按照下面的格式，一次写2、3、4、5、6……直到45.

逗号必须是英文逗号，引号也必须是英文引号。

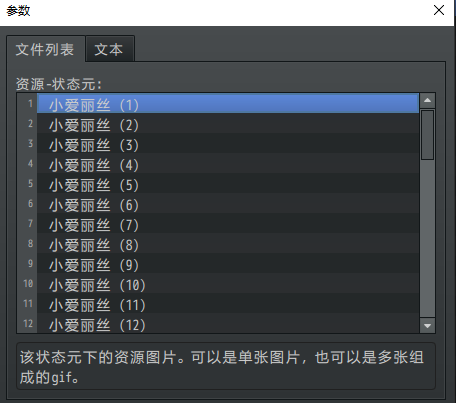


写完后，按Ctrl+A全选，然后复制粘贴到刚才的文本中。





通过这种方式，可以将一堆文件配置上，可以免去一个个点资源图片的麻烦。



### 小工具配置 - 循环GIF

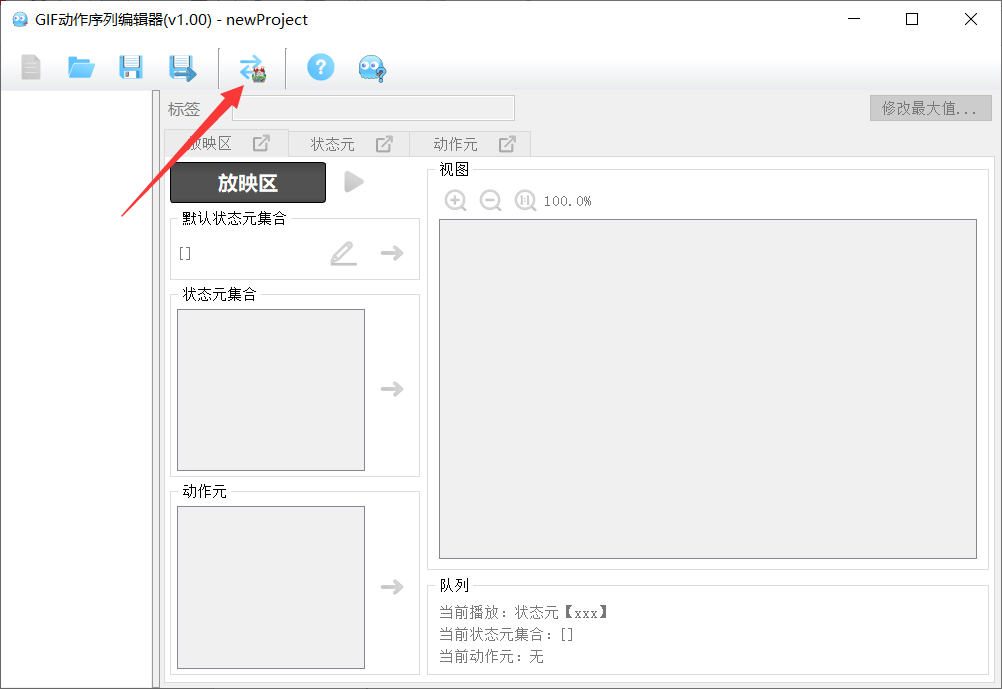
此配置步骤在小工具的操作教程中也有。

可以看看帮助文档：“关于GIF动画序列核心编辑器（入门篇）.docx”。

#### 1）小工具中导入工程

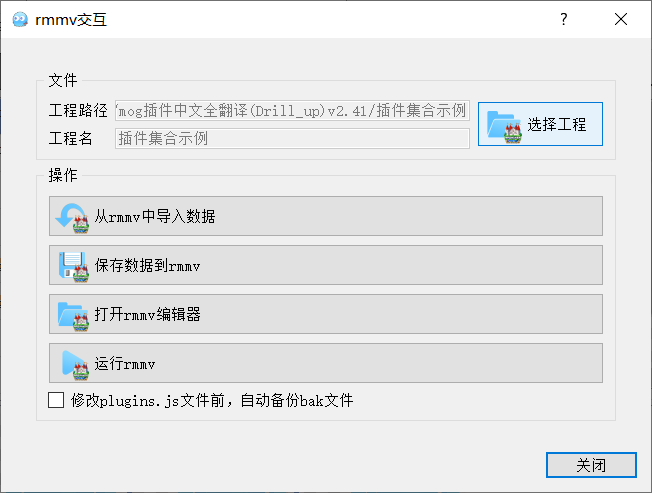
你可以用小工具 新建/导入/打开 工程，这里按 导入的流程 走。

进入软件后，点击rmmv交互按钮。

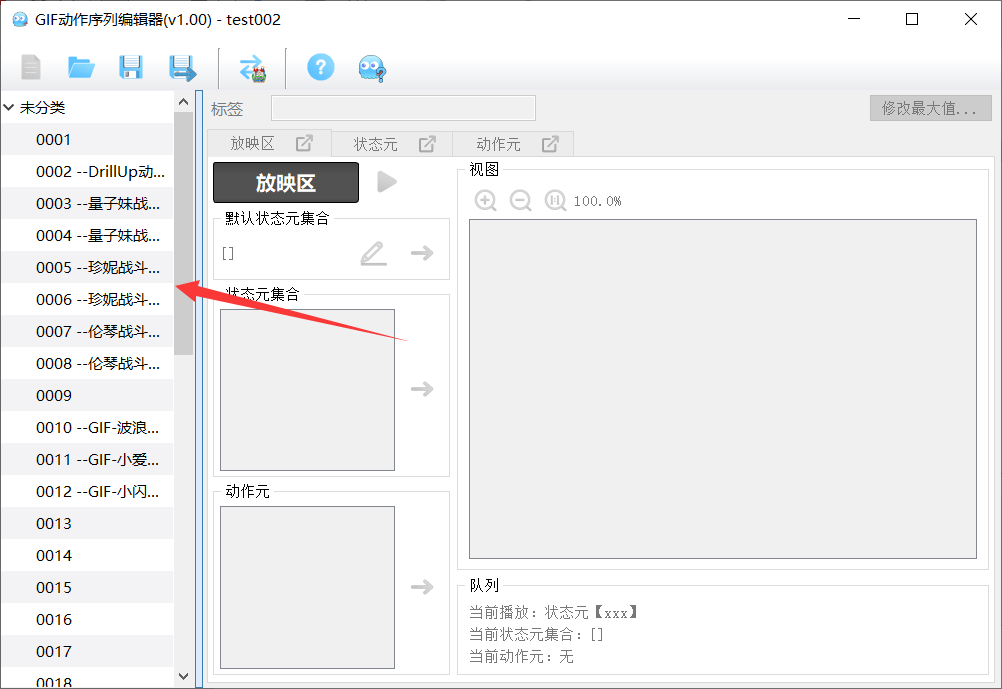


选择一个工程，然后点击“从rmmv中导入数据”。

如果指定的rmmv工程没有加动画序列插件和配置，那么会系统会提示你自动添加新的数据，并走新建的流程。



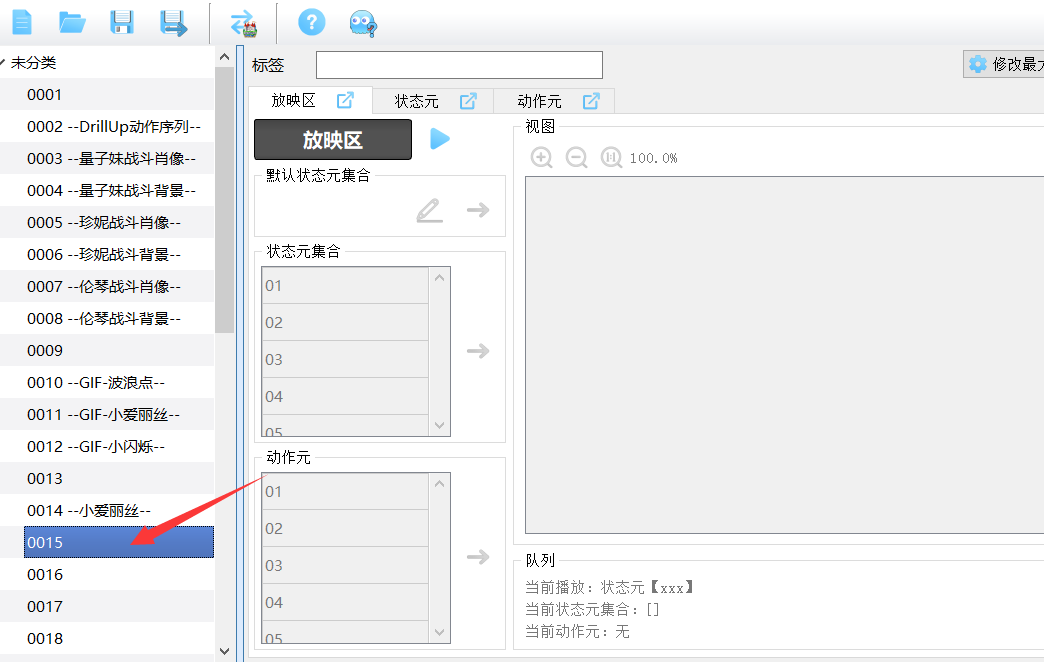
导入 插件示例集合的工程 后，可以看到动画序列数据全部显示了。



#### 2）新建动画序列

选择一个空白区域，准备输入一个新的动画序列。

这里选择的动画序列编号为15。

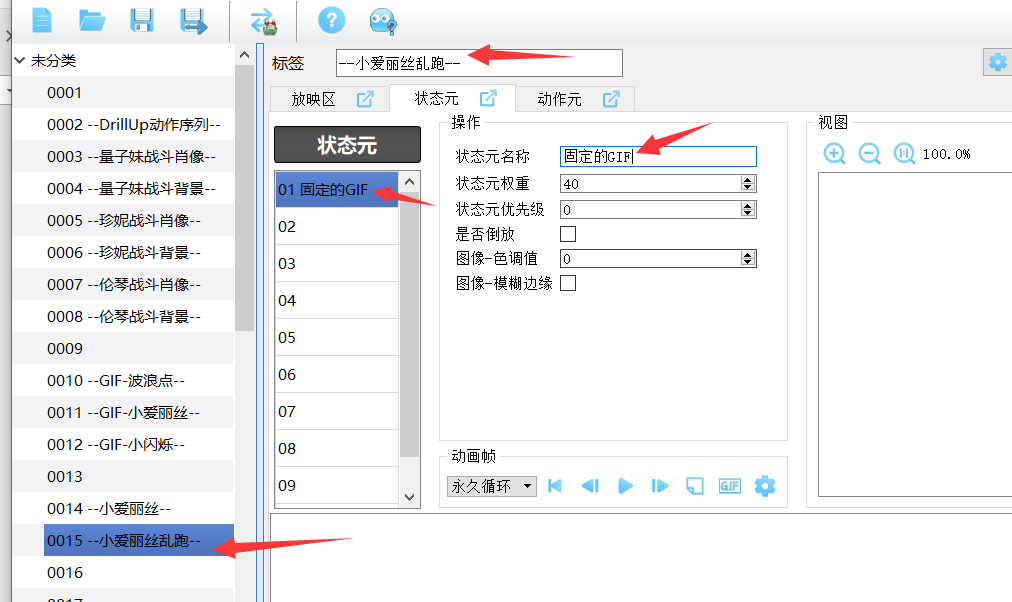


#### 3）新建状态元

在图中下列输入框填写内容。

动画序列的标签随便起一个名字，并不作用到游戏中。

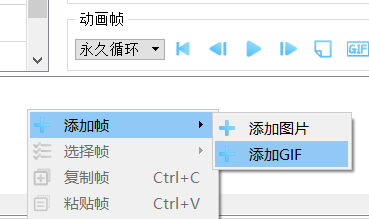
状态元的名称要留意一下，子插件会作为标识来调用。由于只当成一个GIF，所以这里起名为”固定的GIF”。



#### 4）状态元配置GIF

在下方空白帧区域右键菜单，添加一个GIF。比如编辑器中help文件夹中的”小爱丽丝\_示例GIF.gif”文件。（你也可以批量添加图片、序列大图）

添加GIF时，有两种读取器可选择，根据你的情况选择一个。





#### 5）保存工程

保存工程，选择一个文件夹保存为工作空间即可。



需要说明的是，编辑器会自动识别，并帮你填”默认状态节点”参数。

游戏中会自动根据这个默认集合对GIF动画序列进行初始化。

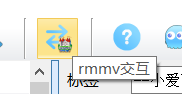
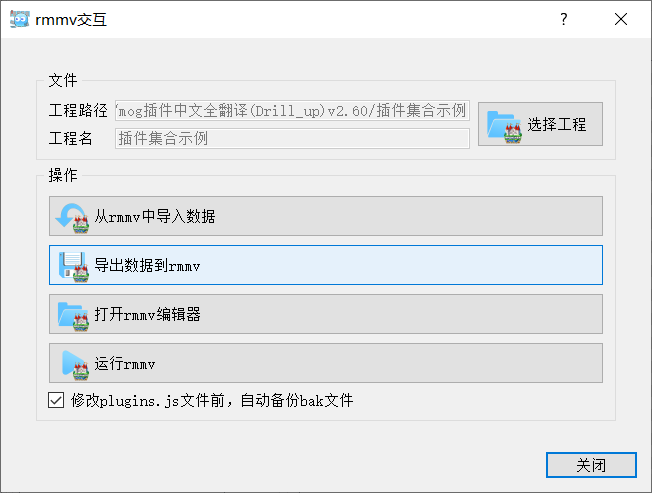


#### 6）导出配置到rmmv

点击选项栏的rmmv交互按钮。

点击”导出数据到rmmv”，即可将动画序列配置转移到rmmv中。

这里软件会提示你，必须要先关rmmv工程，才能导出。

到这里，编辑器的工作就完成了，接下来我们要进入游戏进行测试配好的GIF。

（注意，img/Special\_\_actionSeq 文件夹的图片资源，会自动覆盖，不需要去管）

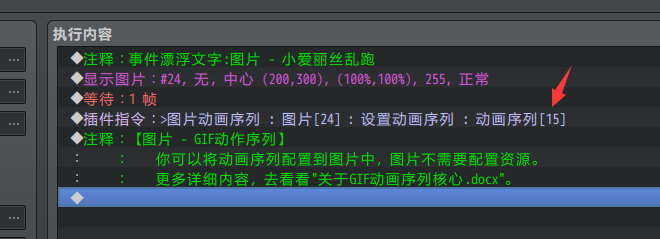
上述操作全都在小工具中进行，完全不需要 经过 rmmv的插件编辑器 哦。

#### 7）插件指令绑定

GIF动画序列核心 单独不能使用，所以这里还需要一个 子插件 ：

◆Drill\_PictureActionSequence 图片 - GIF动画序列

加了子插件后，建立一个事件，如下：（注意动画序列编号对应15）



#### 8）测试

在游戏中接触这个事件，就可以看到图片显示了。

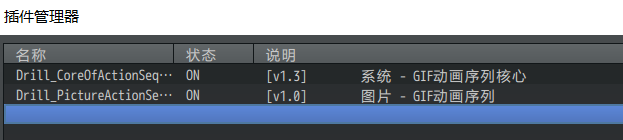


### 小工具配置 - 单次播放GIF

#### 1）创建工程

创建一个初始工程，加入两个动画序列插件。



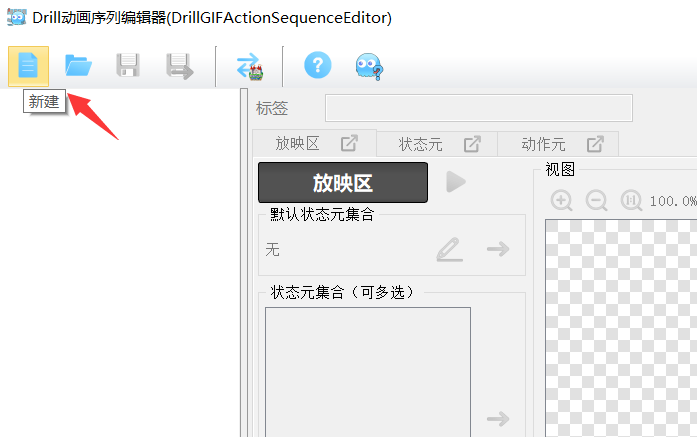


#### 2）小工具中新建工程

注意，这里操作的是 没有动画序列配置 的新工程，所以新建覆盖没问题。

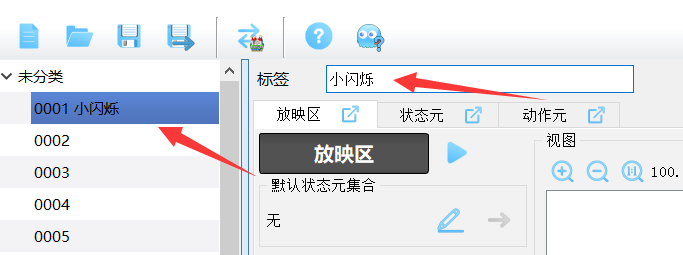
（你可以用小工具 新建/导入/打开 工程，这里按 新建的流程 走）

进入软件后，点击新建。



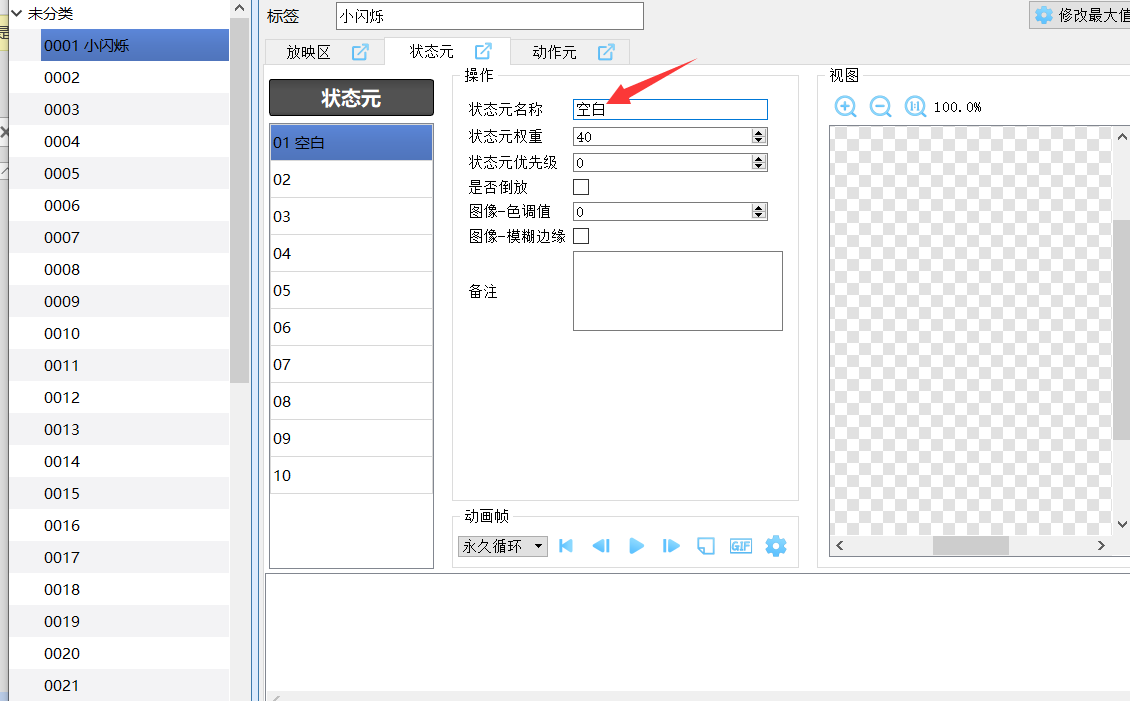
#### 3）新建动画序列

新建工程后，随便在空白处选择一个动画序列，并命名。



#### 4）新建空白状态元

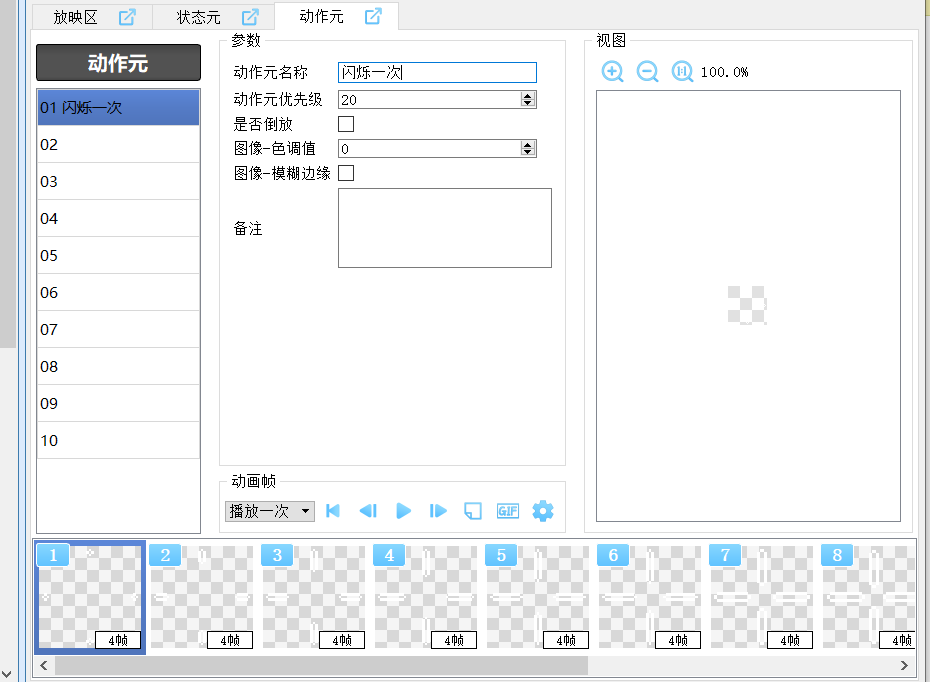
新建一个状态元，写上名称即可，不需要编辑其他内容。



#### 5）新建动作元

新建一个动作元，命名：闪烁一次。

在下面空白处右键菜单添加 图片/GIF/序列大图，配置动作元。



#### 6）保存工程

保存工程，选择一个文件夹保存为工作空间即可。

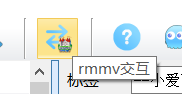
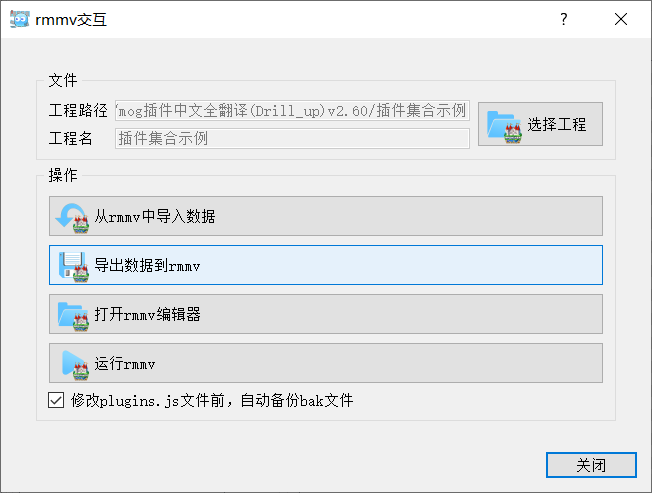


#### 7）导出配置到rmmv

点击选项栏的rmmv交互按钮。

点击”导出数据到rmmv”，即可将动画序列配置转移到rmmv中。

这里软件会提示你，必须要先关rmmv工程，才能导出。

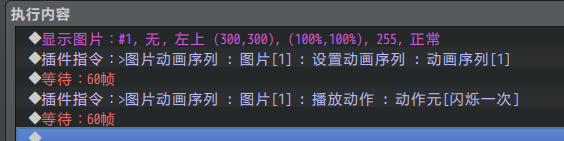
 

到这里，编辑器的工作就完成了，接下来我们要进入游戏进行测试配好的GIF。

（注意，img/Special\_\_actionSeq 文件夹的图片资源，会自动覆盖，不需要去管）

#### 8）插件指令绑定

建立一个事件，执行如下指令：（注意动画序列编号对应1，播放动作元时要对应名称）



#### 9）测试

在游戏中接触这个事件，就可以看到图片显示，并只播放一次动作。

（注意，核心插件v1.3及之前版本，播放结束后会停留在第一帧动作）

（此bug在v1.4中被修复）



### 小工具配置 - 完整动画序列

|  |
| --- |
| 小工具可以将 动画序列 配置得非常灵活、复杂，  可以去看看小工具中的文档 ”关于GIF动画序列核心编辑器（入门篇）.docx”。 |

### 游戏中测试动画序列

#### 1）动画序列配置

首先，要确保动画序列中，已经配置了至少一个**状态元**，并配置了**默认的状态元集合**。



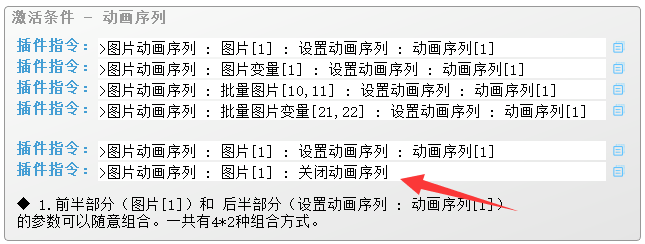
#### 2）子插件配置

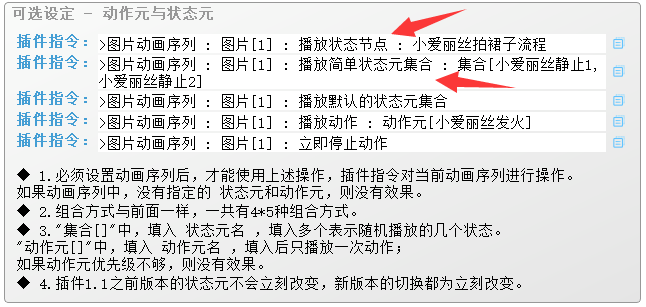
播放功能是基于 子插件 而定的，具体要看子插件的说明内容。

以 图片GIF动画序列 插件为例：

Drill\_EventActionSequence 图片 - GIF动画序列

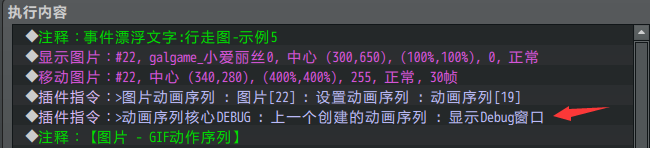
该插件的插件指令如下：

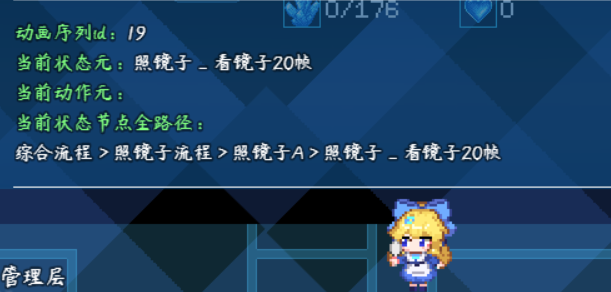




#### 3）插件指令设置

在游戏中，可以开启debug功能，查看当前图片的播放情况：





具体可以去 插件示例中 的 动画序列管理层 看看。