## 概述

### 插件介绍

基于核心：

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

当前插件：

◆Drill\_EventActionSequence 行走图 - GIF动画序列

◆Drill\_EventActionSequenceAutomation 行走图 - GIF动画序列自动化

你可以把行走图变成GIF动画序列，也可以还原。

### 插件关系

行走图动画序列的关系如下图。



## 行走图动画序列

### 原理

该插件实际上内嵌了一套自主向动画序列进行调用的方法，所以设置都是固定好的。

基本的GIF动画序列，通过插件指令即可调用。

自动化功能，通过插件内固定的”@xxx”的格式，进行自动调用。



你可以使用插件指令，播放指定的动作元、状态元；

也可以通过 自动注解集合，匹配 静止、行走、奔跑、跳跃 等的行走图。

### 自动化

自动化 与 行走图切换 的功能相似，自动化能适应更加复杂的情况。

**1) 注解单元**

目前可用的注解单元如下：（后期会增加更多）

|  |  |  |  |
| --- | --- | --- | --- |
| 第一条件 | 第二条件 | 第三条件 | 第N条件… |
| @向下  @向左  @向右  @向上 | @静止  @移动  @奔跑  @跳跃  @滑行  @被举起 | @空手  @举花盆 | …… |

上述注解都是程序固定的字段，不可修改。

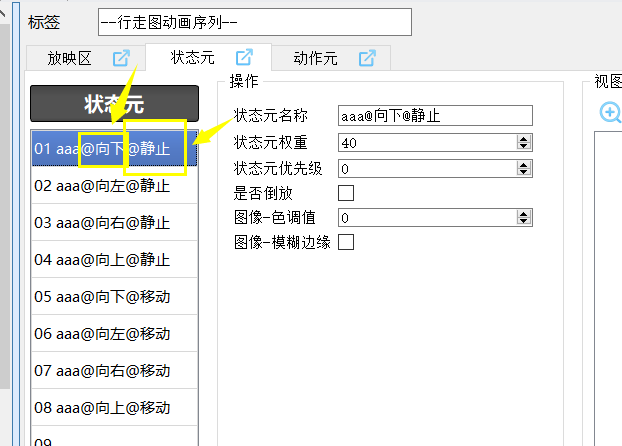
**2) 特殊条件**

条件有一些特殊情况：

|  |  |
| --- | --- |
| 条件 | 说明 |
| @奔跑 | 事件没有奔跑状态，只有 玩家和玩家队员 有奔跑状态。 |
| @滑行 | 只有玩家和玩家队员 有滑行状态。 |
| @举花盆 | 只有玩家有举花盆状态。 |

**3) 扫描方式**

举个例子，一个事件处于以下状态：@向下、@静止、@空手



那么，系统会找全部满足的状态元 ”xxxx@向下@静止@空手”。

如果没有全部满足的，则找次要满足的“”xxxx@向下@静止”；

如果仍然没有，则继续找”xxxx@向下”；

如果一个符合的条件都没有，则系统 不操作 状态元的修改。

简单来说，就是每次系统都会找满足注解条件最多的状态元。

注解条件越详细，对应的状态元越精准。

**4) 组合注解**

下表为 第一条件和第二条件 的全部组合。与行走图的结构有些相似。

如果你的美术资源够多，你可以添加大量含注解的状态元。

|  |  |
| --- | --- |
| @向下@静止  @向左@静止  @向右@静止  @向上@静止 | @向下@移动  @向左@移动  @向右@移动  @向上@移动 |
| @向下@奔跑  @向左@奔跑  @向右@奔跑  @向上@奔跑 | @向下@跳跃  @向左@跳跃  @向右@跳跃  @向上@跳跃 |
| @向下@滑行  @向左@滑行  @向右@滑行  @向上@滑行 | @向下@被举起  @向左@被举起  @向右@被举起  @向上@被举起 |

**5) 多个相同注解**

你可以写两个或以上的相同注解：

比如：

小爱丽丝@向下@静止@举花盆

小爱丽丝2@向下@静止@举花盆

小爱丽丝3@向下@静止@举花盆

这3个 状态元 都满足条件，那么这三个都会被送入序列中。

按照 状态元集合 的结构，根据权重随机抽取并播放。

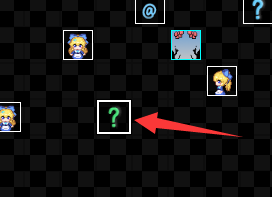
## 配置方法

### 直接配置

单用行走图动画序列插件就可以实现。

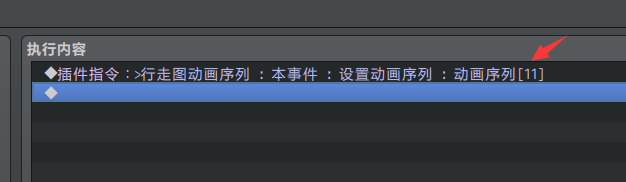
◆Drill\_EventActionSequence 行走图 - GIF动画序列

1）创建一个事件



2）添加执行的插件指令

比如给事件设置动画序列11。



对应到示例默认的45帧GIF。



3）接触事件即可激活

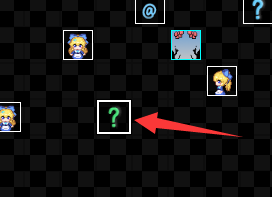
需要注意的是，事件的中心锚点是正下方。

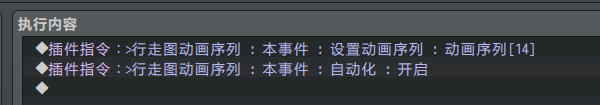
### 自动化配置

自动化配置一样，只是多了个指令。

1）创建一个事件

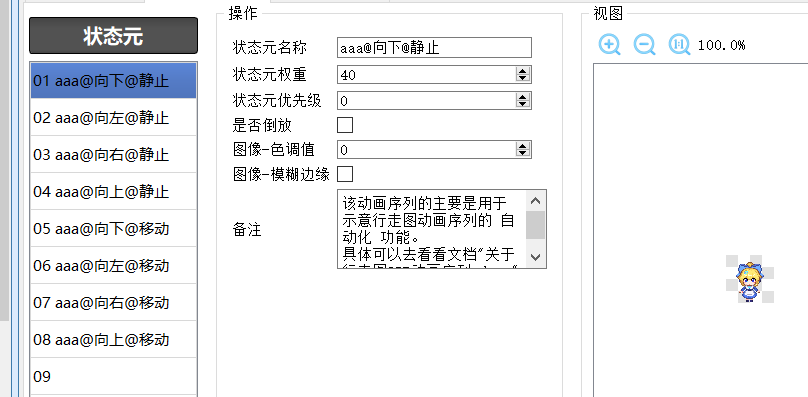


2）添加执行的插件指令



你至少要有：@向上、下、左、右@静止 和 @向上、下、左、右@移动。

动画序列播放时才看起来自然。



关于自动化的 注解说明，可以看看前面章节：[自动化](#_自动化) 。

3）接触事件即可激活

## 其他说明

如果你在配置过程中遇到了无法解决的问题，

可以去看看”关于GIF动画序列核心.docx”文档中的 常见问题 章节。