## 概述

### 相关插件

这里主要对rmmv内置的事件页、独立开关作详细结构说明。

相关插件如下：

◆Drill\_EventOtherCondition 物体 - 事件页出现条件

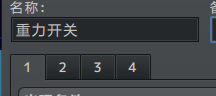
◆Drill\_EventSelfSwitch 物体 - 独立开关

插件起到辅助作用，可以使你设计更多独立开关E、F、G、H……以及更多变量、开关条件。

## 事件页机制

### 出现条件

事件页的变化，是根据 **出现条件+最大的页面编号** 来控制的。



如果事件页4的条件为空，且没有”<<出现条件>>”注释，那么前面3页将永远没有效果。**因为4一直满足条件，没有不满足条件的情况，所以无法跳至3**。

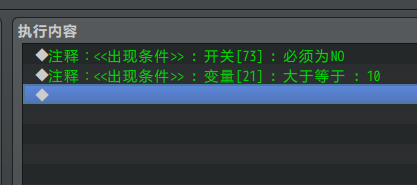


**通常事件页的出现条件主要分为三种：开关、变量、独立开关**。所有其他复杂条件，都可以先绑定开关/变量，再对开关/变量进行赋值。



举个例子，如果你想设置背包中的物品达到一定数量时开启事件页。那么你应该让事件页的出现条件绑定到变量，再将物品数量赋给那个变量。

通过 事件页出现条件插件，你可以设置多个开关和变量的条件。而独立开关不一样，只能写一个。下图中左边和右边的效果一样。

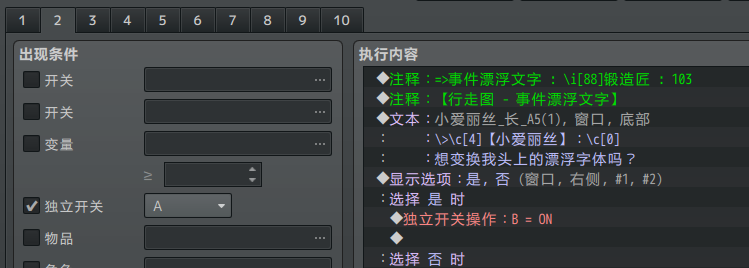
 

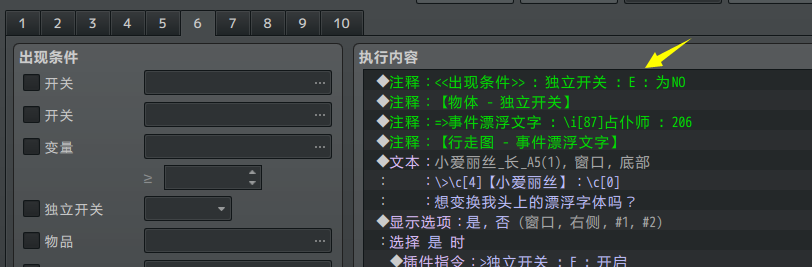
### 独立开关

独立开关是控制事件自身特有的条件结构，一般规定一个独立开关控制一个事件页，这样可以使得设计事件的逻辑变得比较清晰。

如果你有多个变量、开关条件，写多条注释即可。但是**独立开关写多条没有效果**。

（下图为地图管理层的”任务师”小爱丽丝，10个事件页分别对应：无条件、A、B、C、D、E、F、G、H、I ）





## 事件页流程

### 基本流程

下面，这里开始对事件页流程进行设计，**这里只考虑事件页与独立开关，而开关、变量这两个更复杂的条件不考虑**：

|  |  |
| --- | --- |
| 事件页 | 条件 |
| 1 | 无 |
| 2 | 独立开关A |
| 3 | 独立开关B |
| 4 | 独立开关C |

开启独立开关A，那么进入事件页2；开启B，进入3；开启C，进入4。用活动图来表示，则为下面的情况：



如果反过来，如果依次关闭，则为：



由于事件页会受到**最大页面编号**的影响，如果直接开启独立开关C，则会出现下面的情况：



### 环形流程

如果事件跳转到事件页2，而事件页2执行完后，关闭A开关，回到事件页1，那么就可以看做这是一个环形的流程。

（你可以对应去看看 计时开关 的原理，计时开关就是下图的环形流程。）



下面，以鼠标悬停为例，小爱丽丝接触到鼠标之后，会响一声，然后变为红发。**如果没有响一声的动作，直接变为红发，那么就是上图的12环形结构**。但是正因为有响一声的动作，所以，1应该先跳转至2，然后2跳3。鼠标离开了，再跳1。





那么跳转的控制应该如下表：

|  |  |  |  |
| --- | --- | --- | --- |
| 事件页 | 条件 | 描述 | 指令 |
| 1 | 无 | 默认状态 | 鼠标接触，开启独立开关A。 |
| 2 | 独立开关A | 鼠标悬停动作 | 发出点击声。打开B。 |
| 3 | 独立开关B | 鼠标悬停状态 | 如果A被关闭，立即关闭B、C。 |

### 复合流程

一般情况下，一个事件可能会有两个或者三个环形流程，但是他们都被安排在一个事件页序列里，于是就造成了更加复杂的情况。



**+**



**=**



图中的结构是示例中，设计-逻辑图形 的多状态开关，第一个环（蓝）是状态切换的流程，第二个环（淡蓝）是鼠标点击的流程。

淡蓝点击一次之后，蓝色的环进行一次跳转，即1->4，再点击一次，即4->5……以此类推。