## 概述

### 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：

|  |  |
| --- | --- |
| 独立开关 | [独立开关](#独立开关) [自定义独立开关](#_自定义独立开关) |
| 事件页 | [事件页](#事件页) [出现条件](#出现条件) [自定义出现条件](#_自定义出现条件)  [事件删除](#_事件删除) [事件页设计](#_事件页设计) |

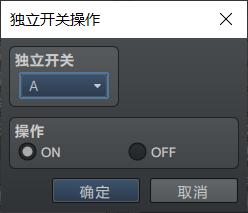
## 独立开关

### 机制

**独立开关：**指专用于事件自身的开关。

一般通过开启/关闭独立开关，实现切换事件页。

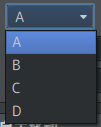
开关也能实现切换事件页，只不过开关可以跨地图，而独立开关不能跨地图。

|  |
| --- |
| 如果事件指令只影响事件自身，那么就应该使用独立开关。  一来可以确保在复制事件时，不用考虑外部关联问题；二来可以节省开关的占用。 |

默认情况下，每个事件的独立开关只给了四个。

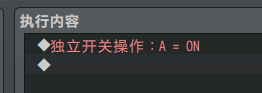
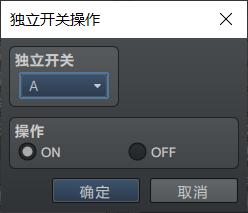
你可以通过插件，自定义更多的独立开关，可以去看看后面章节：[自定义独立开关](#_自定义独立开关)。



### 独立开关的控制

#### 1）事件指令

通过事件指令能修改独立开关，但只能修改事件自己的独立开关，且只有ABCD四个。

#### 2）插件指令控制

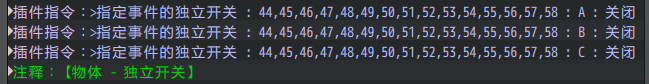
需要用到下面的插件：

◆Drill\_EventSelfSwitch 物体 - 独立开关

使用下面插件指令，可以直接 开启/关闭 独立开关。



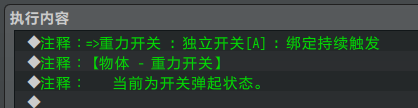
你也可以直接 开启/关闭 地图中 指定事件/多个事件 的独立开关。

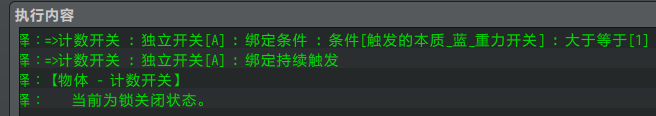


#### 3）物体触发控制

物体类的开关插件、物体触发类插件 都可以触发 开启/关闭 一个自定义的独立开关。

详细内容可以去看看 ”8.物体 > 大家族-开关.docx”。





|  |
| --- |
| 这些插件经常作为 触发 的基础插件，实现游戏中复杂的触发功能。  可以去看看文档：”8.物体 > 触发的本质.docx”。 |

### 自定义独立开关

需要用到下面的插件：

◆Drill\_EventSelfSwitch 物体 - 独立开关

该插件能支持更多的独立开关E、F、G、H……并且独立开关的名称可以自定义。

#### 1）使用方法 - 插件指令

要切换到自定义独立开关，使用下面的插件指令即可：



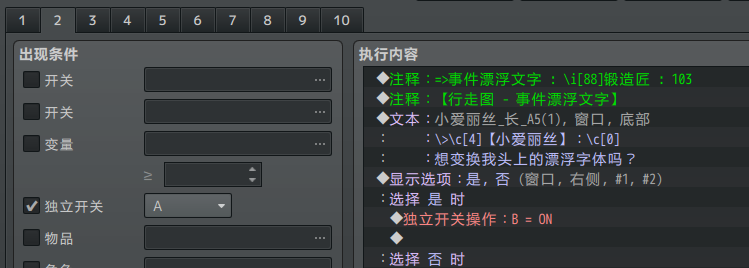
#### 2）使用方法 - 出现条件

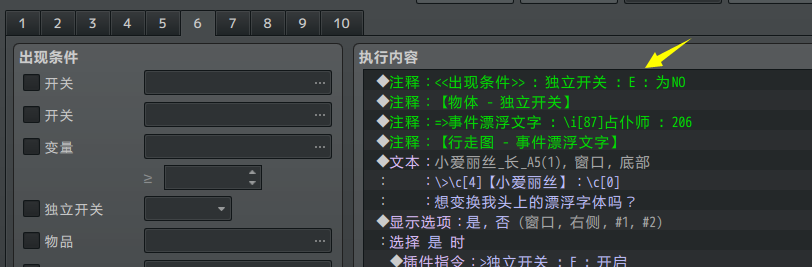
出现条件设置需要后面章节 [自定义出现条件](#_自定义出现条件) 的插件的配合。

如下图，示例中 行走图管理层 的”任务师”小爱丽丝就使用了大量独立开关。

10个事件页分别用到了：无条件、A、B、C、D、E、F、G、H、I。



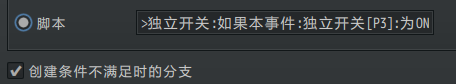


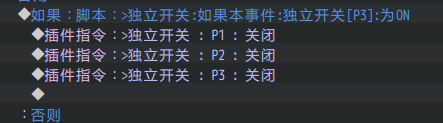


#### 3）使用方法 - 分支条件

独立开关插件能支持通过写 分支条件指令 的方式，

对自定义的独立开关进行判断。



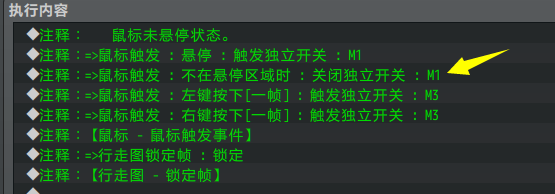


#### 4）自定义名称设计

以鼠标管理层为例，与鼠标点击相关的独立开关，专门用 M1，M2，M3 来命名。

这样可以与一般的独立开关区分开来。





独立开关需要与后面章节 [自定义出现条件](#_自定义出现条件) 的插件的配合使用。

### 独立开关与触发

**触发：**指在某个时机或某个条件下，执行自定义指令。

**触发的本质：**触发的本质就是**命题** ，若p则q。



**触发与独立开关：**若p则开启独立开关A，若开启独立开关A则r，独立开关A作为中间过程，能实现p到r的触发设计。



|  |
| --- |
| 使用独立开关，最终目的是为了 实现触发功能。  你可以去了解触发的基础知识，结合文档看看：”8.物体 > 触发的本质.docx” |

触发能够被拆分，还能套娃拆分，

将触发分解为更多小功能，通过独立开关依次实现。

**触发与独立开关拆分：**要实现若p则r，可以插入独立开关A作为中间过程，将其变成 “若p则开启独立开关A” 与 “若独立开关A开启则执行r”。



**触发与独立开关套娃：**根据触发的传递性，你能用套娃的方式将独立开关AB作为中间过程，实现p到r的触发设计。



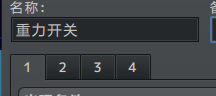
|  |
| --- |
| 具体 设计触发 的方法与流程，去看看文档 ”8.物体 > 触发的本质.docx” 。  这里只做简单介绍，因为独立开关只是 设计触发 的一个中间过程而已。 |

## 事件页

### 机制

**事件页：**指事件自带的事件指令集合。

一个事件带有多个事件页，每个事件页都按顺序进行编号。



**出现条件：**指激活事件页所需的条件。

事件会根据 **出现条件+最大的页面编号** 来确认事件页，然后执行这个事件页的指令。

通常事件页的出现条件主要分为三种：**开关、变量、独立开关**。

其他的出现条件，都可以通过 先绑定开关/变量，再对开关/变量进行赋值 来实现。



### 事件页设计

#### 1）基本流程

下面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**，开关、变量条件暂不考虑：

|  |  |
| --- | --- |
| 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：



#### 2）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，所以需要添加一个事件页来专门执行动作。

于是为：1先跳转至2，然后2跳3。鼠标离开了，再跳1。





此跳转的控制如下表：

|  |  |  |  |
| --- | --- | --- | --- |
| 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。 |

#### 3）复合流程

有时，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。



**+**



**=**



图中的结构为示例中 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（白）是鼠标点击的流程。

白环流程点击一次之后，蓝环流程进行一次跳转，即1->4，再点击一次，即4->5……以此类推。

### 自定义出现条件

需要用到下面的插件：

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

#### 1）使用方法

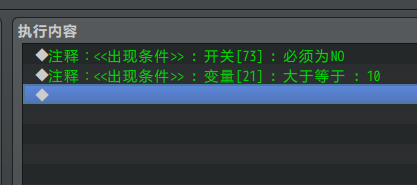
默认情况下，假设你需要设置背包中的物品达到一定数量时开启事件页。

那么你需要先设置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。



而通过 事件页出现条件插件，你可以直接用注释设置开关和变量的条件。

下图和上图的设置效果，一模一样。



你可以写多个 开关或变量 的注释，来作为出现条件。

但需要注意，独立开关的出现条件，只能写一个。

#### 2）注意事项

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页将永远跳转不了。**因为4一直满足条件，没有不满足条件的情况，所以无法跳至前面事件页**。

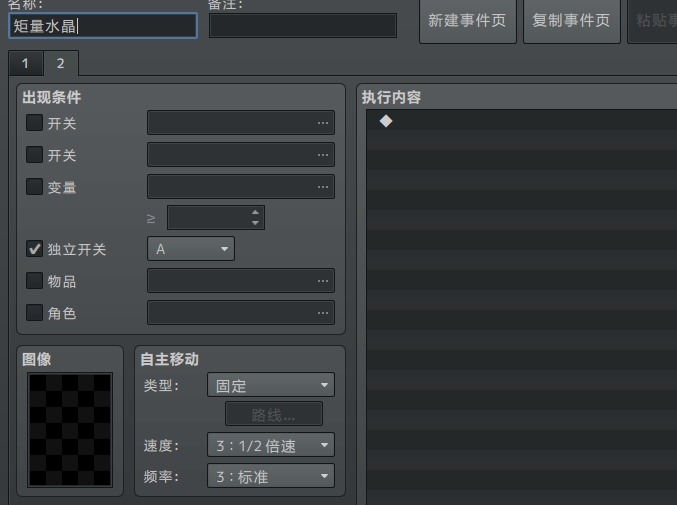


### 事件删除方法

#### 1）空事件页方法

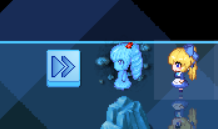
通常来说，开启独立开关，然后让事件进入一个全空的事件页即可。

但是此操作并不会真的删除事件，事件仍然在那里，只是以低消耗方式站在那里。



需要注意一种情况：比如你站在一个空的事件页脚下，你无法对面前的npc进行对话。因为这个空事件会吸收掉你对前面npc的对话请求。

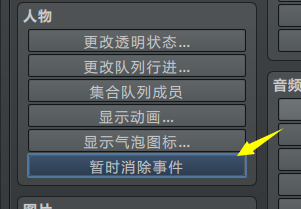
如下图，站在瓦砾事件上，无法与前面的方块进行互动；站在圆盘上，无法与前面的小爱丽丝对话。

#### 2）用指令消除方法

使用事件指令”暂时消除事件”，能够临时消除事件。

但是离开地图然后回来，事件又会出现。



#### 3）事件管理核心

考虑到有时需要 复制大量事件 的情况。

若不彻底删除，即使是静态的事件，也会不断地累积，拖慢游戏的运行速度。

为了确保删除完全，可以通过 事件管理核心 来将事件彻底删除。

具体可以去看看文档 “28.物体管理 > 关于事件管理核心.docx”。

## 从零开始设计（DIY）

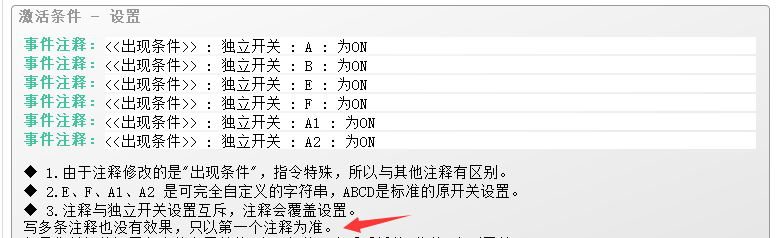
### 设计“独立开关A+B都为ON”的出现条件

#### 1. 设置一个目标

2024年2月6日，作者我使用插件 “物体-独立开关” 的时候，

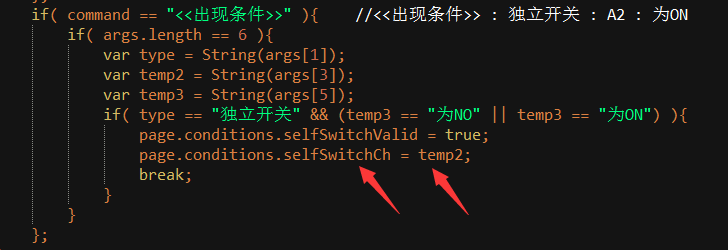
发现了一个常用指令 <<出现条件>> 中，非常反人类的设定。

“与独立开关设置互斥” “只能以第一个注释为准”。



如果我想要多个独立开关都为 ON，怎么办？

于是作者我跑去看了一眼脚本，



哦，独立开关的确只能设置一个，本身底层设定就只能写一个……改不了。

作者我看看有没有什么其他的方法。

#### 2. 结构规划/流程梳理

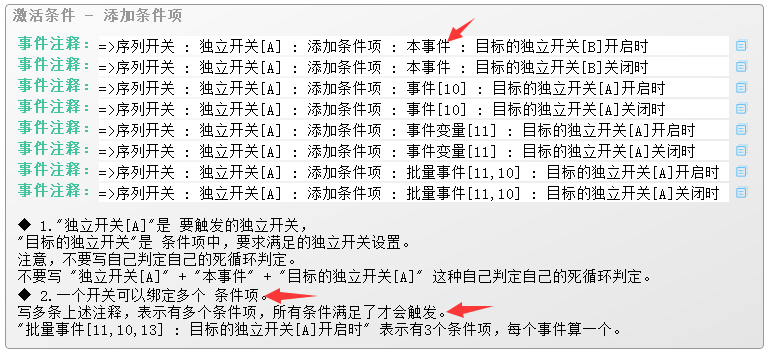
本来作者我打算开新坑，

但转念一想，不对，这个插件我好像已经写过了。

没错！就是插件 物体-序列开关 。

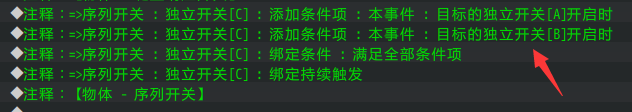
Drill\_EventSequentialSwitch 物体 - 序列开关

这个插件能够添加任意多个 固定条件项，全部满足才能开启独立开关。

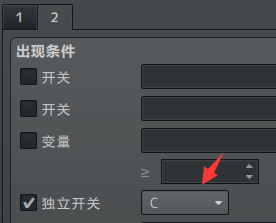


#### 3. 事件编写

在事件页第一页添加下面的指令，



第二个事件页，使用 序列开关 指向的 独立开关C。



即可实现 “独立开关A+B都为ON” 的出现条件功能。

|  |
| --- |
| 你可以去看看文档：“8.物体 > 大家族-开关.docx”。  了解各个开关插件，通过组合实现各种有趣的功能。 |