## 概述

**基础要求**

|  |  |
| --- | --- |
| 先尝试回答下面的问题：  什么是触发？  什么是区域触发？  什么是主动方和被动方？  什么是持续触发和单次触发？  事件的串行、并行如何设置？ | 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 触发的本质。

需要去看看：”8.物体 > 触发的本质.docx”。

也可以去看看示例中 机关管理层 的 触发的本质 介绍。

### 相关插件

主要插件如下：

◆Drill\_CoreOfFixedArea 物体触发 - 固定区域核心

◆Drill\_EventAutoTrigger 物体触发 - 固定区域 & 玩家接近 & 条件触发

这里主要介绍 玩家接近触发 的相关功能。

如果你想了解 事件接近触发，去看看文档“9.物体触发 > 关于事件接近触发.docx”。

|  |
| --- |
| 若已了解触发的本质，不多废话，先打开插件，把DEBUG调试用的区域显示功能打开，然后进入游戏看。很多思路就清晰了，打开方法去看章节：[DEBUG区域显示](#_DEBUG区域显示) 。 |

### 快速区分

|  |  |
| --- | --- |
|  |  |
| **玩家接近** | **事件接近** |

更详细的区分，可以看后面章节：[相似插件区别](#相似插件区别) 。

### 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：

|  |  |
| --- | --- |
| 插件名概念 | [固定区域](#固定区域) [玩家接近](#玩家接近) [条件触发](#条件触发) |
| 区域主体 | [玩家](#玩家) [玩家触发区域](#玩家触发区域) [目标事件](#目标事件) |
| 触发 | [持续性](#_持续性) [条件触发](#条件触发)  [目标事件触发](#目标事件触发) [筛选器](#_筛选器) |

### 插件关系

插件关系如下，玩家接近和事件接近 插件相互不干扰：

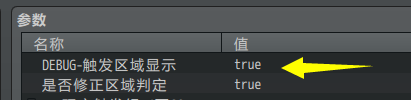


## 玩家接近

### DEBUG区域显示

#### 1）开启DEBUG

在参数中，设置为true即可。

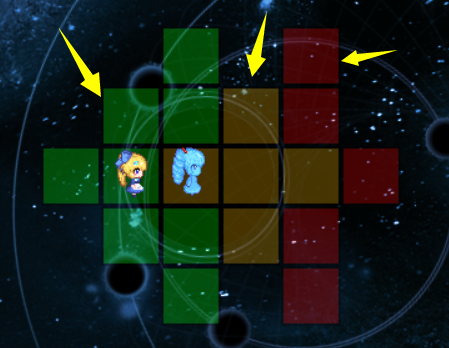


由于玩家的大量区域都是叠加在一起的，所以你还需要进入区域配置中，单独开启/关闭，并配置颜色。

#### 2）叠加色

两个以上的区域显示时，会出现重叠的情况，重叠的部分可能会出现颜色叠加，而产生不同的颜色，比如 红+绿 = 黄。留意一下这是正常现象即可。



### 区域主体关系

#### 1）定义

**玩家：**即区域的所有者，区域跟随 玩家的位置与方向 变化。

**玩家触发区域：**插件中所有的区域，都强制绑定于玩家。

**目标事件：**符合区域的条件触发的目标事件，进入区域范围后，触发开关。



由于玩家本身不是事件，没有事件页机制，所以区域只会对目标事件进行触发。

当区域符合触发条件后，会执行：[触发事件](#_触发事件) 。

#### 2）相似插件区别

**玩家接近：**指多个区域固定在玩家身上，事件接近了玩家的区域，会开启独立开关。

**事件接近**：指多个区域固定在事件身上，玩家/其它事件接近了该事件的区域，可以开启 该事件或接近事件 的独立开关。

因为玩家不属于事件，没有自身的独立开关和事件页，所以 玩家接近和事件接近 被分成了两种不同的插件结构。

### 筛选器

#### 1）固定区域

该插件基于固定区域，可以使用 形状区域或自定义区域 作为接近触发范围，

详细可以去看看“9.物体触发 > 关于物体触发-固定区域.docx”

#### 2）筛选器

如下图，前者是无筛选器，后者是去除了 不可通行 区域的筛选器。

如果事件处于 **被筛选掉的区域** 时，则事件不会被触发。

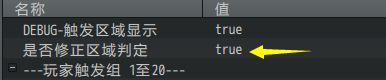
开DEBUG显示，玩家移动时，区域经过石头的位置可能会闪一下，这是因为显示区域刷新延迟问题。但这并不会触发到事件，实际情况下 不符合的位置 已经被筛选了。

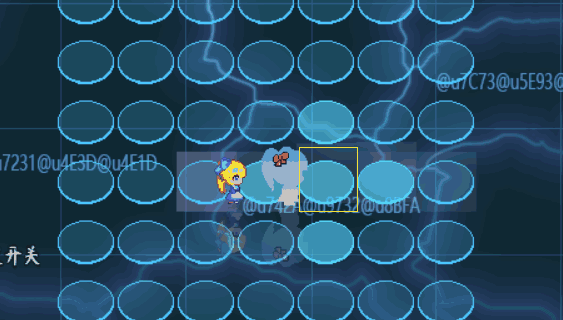
### 区域修正

**游戏原设置有个缺陷，前进一步，在你开始前进的那一瞬间，你就已经被判定为在前进的位置了**。

插件默认开启了区域修正，你可以关闭，然后去物体触发管理层看看效果。

如下图的判定，你前进之前，亮片就已经变化了。





因此默认开启修正，能够确保未完全离开当前位置时，仍然保持当前区域的触发情况。

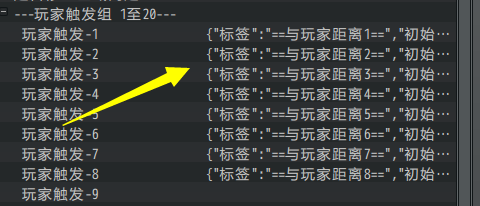
## 触发

### 持续性

玩家接近触发是 持续不断的 。

插件中添加区域后，游戏中每一帧都进行这些区域的 接近判断。

一旦区域发生变化，触发会即时变化。



与插件指令控制的单次触发不同，

插件指令执行了单次物体触发时，触发才会变化一次。





举个例子，下图玩家站在“与玩家距离1”的触发范围中，如果你使用 事件指令 将下图范围内的亮片的 独立开关 全部关闭，是不起作用的。

因为持续触发，亮片会在关闭后立即重新开启。



### 触发事件

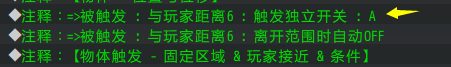
看本章节前，先了解下对象间基本关系：[区域主体关系](#_区域主体关系) 。

#### 1）条件触发

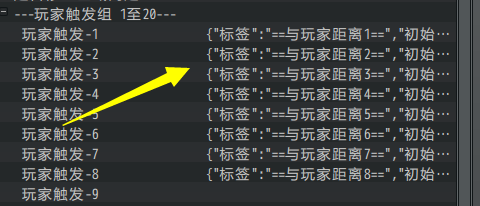
每个区域都有一个固定的**关键字**，这个关键字用作为 被触发 的条件。

只有关键字匹配，才能被触发。

以下图为例：目标事件只有 ”与玩家距离6” 的条件，条件匹配才会开启独立开关A。



玩家默认具备1-8的区域，但由于目标事件只与条件”与玩家距离6”，因此触发将按距离6的情况来算。



#### 2）目标事件触发

目标事件触发指 **目标事件** 进入范围后触发 **目标事件** 的独立开关。

比如物体触发管理层中的乱跑小爱丽丝，

事件进入了玩家的范围，会开启独立开关A的事件页，执行逃跑行为。



由于玩家自身不是事件，所以玩家不需要考虑被触发关系。

### 触发条件的简化思想

玩家自动触发中，

可能会有不同的区域，触发同一个条件。

而且，不同的条件之间，有很多相交的区域图块，而且还有筛选器问题……

**如果你考虑各种 复杂组合 多了，就很容易把自己绕晕，实际上在你设计时，应该减少每一个变量，只考虑单一效果**。

举个例子，下图的小爱丽丝，同时具有“与玩家距离3”区域触发、“与玩家距离4”区域触发，这两个区域是相交的，当玩家突然出现在她面前时（ 比如切换地图/瞬间位移），是3先触发还是4先触发？



回答问题前，你不应该去想哪个先触发，**而是应该想办法简化思路**。

触发的流程是这样的：

玩家主动触发 > 被触发条件距离3激活 > 事件独立开关A开启 > 切换A的事件页

玩家主动触发 > 被触发条件距离4激活 > 事件独立开关B开启 > 切换B的事件页

而上面一大串的内容，简化后是这样的：

与玩家3距离的事件页

与玩家4距离的事件页

最后你才会发现，你写那么一大串，只是为了使得这个**事件页具有一个特殊条件**而已。至于谁先触发，当然是看 事件页顺序，哪个在后面哪个算了。

**物体触发的本质是切换事件页，区域、关键字、筛选器、触发开关 都是局部概念而已**。

## 性能消耗

### 多次优化

玩家接近触发的性能消耗非常考验脚本内部的算法。

假设玩家身上有20个区域，平均每个区域有12个坐标点。每个坐标点要判断地图中30个事件是否激活触发。1秒60帧。

那么每秒都要计算 20x12x30x60 = 432000次。

这计算压力对于计算机来说非常大。

经过多次版本更新优化，使用了最新的棋盘算法（v2.1版本以上），此算法能将区域压缩在一起，并把事件判断次数减半。

大概能减少到每秒21600次。

这个消耗对于只考虑玩家的范围情况来说，已经比较可观了。

（注意这里只是举例，次数并不是真实次数，代码内部要复杂的多，实际计算量更复杂）

但在设计时，最好尽可能减少 **地图事件数量、玩家区域数量、区域的坐标点数量**。

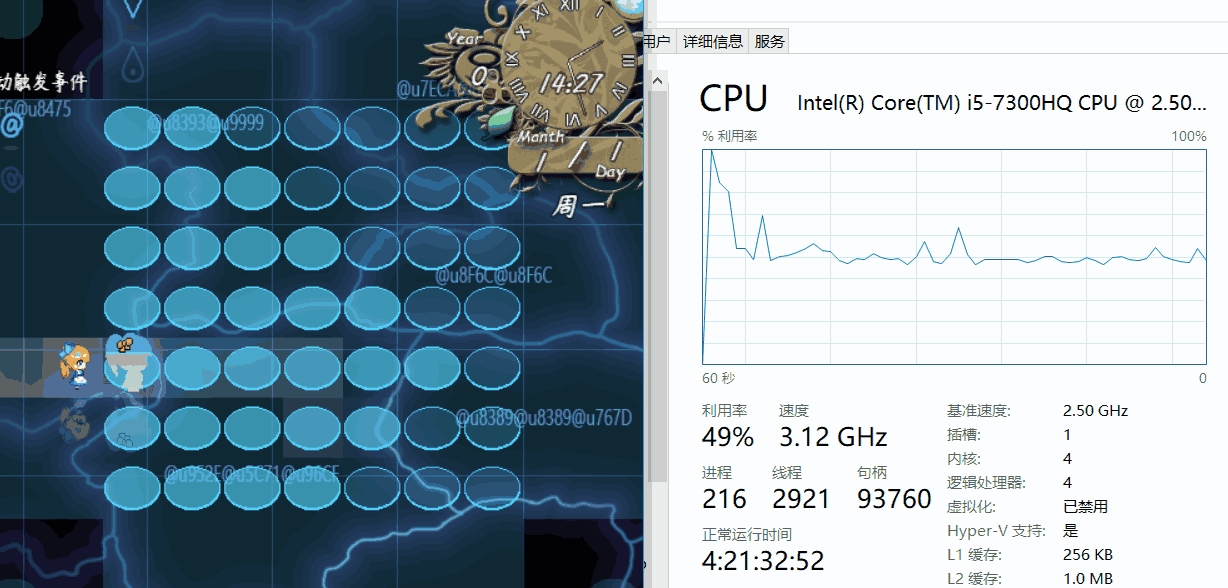
### 查看方式

一般的电脑中，地图中同时放300个事件，不会引起特别的卡顿。

**但手机的能力就不行了，最多100个事件，不然超级卡。**

（fps 30帧以上都属于流畅运行，在游戏时按F2可以查看帧数。）

你可以按Ctrl+Shift+Esc打开任务管理器，然后去物体触发管理层的76个亮片那里转悠，看看CPU的情况，CPU最高峰为游戏载入时的情况，次高峰为进入地图时的情况。



具体性能的分数与消耗，可以去看看“性能测试统计表.xlsx”。

## 从零开始设计（DIY）

### 看见玩家就逃/看见玩家就抓的小爱丽丝

#### 1. 设置一个目标

此功能虽然老生常谈了，但是其中的逻辑并不简单，尤其在事件接近触发加入后，概念非常容易混淆。

你必须先想好要做的功能，再去寻找挖掘其中的限制条件，构建设计思路。

这里，作者我打算制作一个小爱丽丝事件，事件看见玩家，就赶紧逃跑。



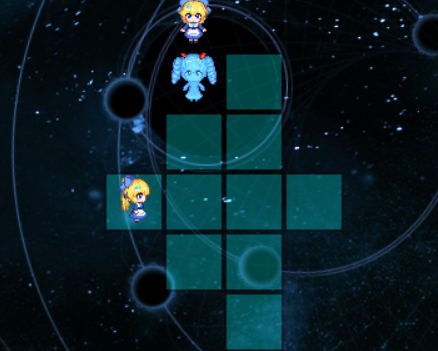
#### 2. 结构规划/流程梳理

这里，“看见玩家”的定义是什么？

这种“看见”有两种定义：

一种是玩家接近到事件一定的距离后，事件产生了警觉。

另一种是事件通过视野范围，真的看见了玩家。

前者，即 **目标事件** 进入了玩家的范围，玩家触发了 **目标事件**。

此功能，当前插件能实现，关系可见：[区域主体关系](#_区域主体关系) 。

后者，即 玩家进入了 **主动事件** 的范围内，玩家触发了 **主动事件** 。

此功能，当前插件不能实现，

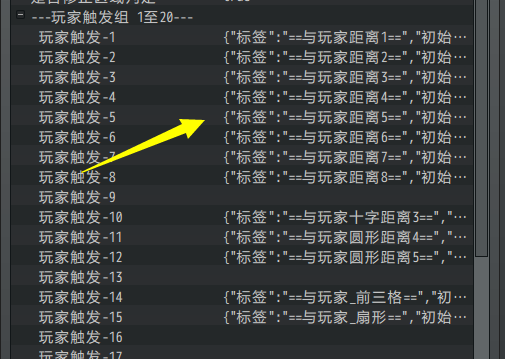
你应该去看“9.物体触发 > 关于事件接近触发.docx”中的设计章节 小爱丽丝看见玩家后打招呼 。

#### 3. 区域设置

当你确认了使用 玩家接近触发 后，

相关设置，直接可以从插件中进行编辑设置。

因为所有区域都是直接绑定与玩家的，因此只需要添加触发即可。

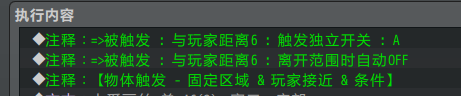
 

区域形状的概念可以去了解下 “9.物体触发 > 关于物体触发-固定区域.docx”。

如果你要设置 **自定义区域**，去看看 那个文档中 的 **自定义区域配置** 章节。

#### 4. 事件触发设置

目标事件添加对应的 玩家触发的 条件关键字 即可。

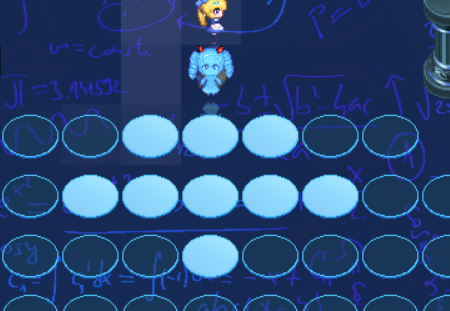


游戏中测试，能够自动根据对应的区域进行触发。



你也可以使用 自定义区域，

自定义区域能够根据玩家的不同朝向而变化。

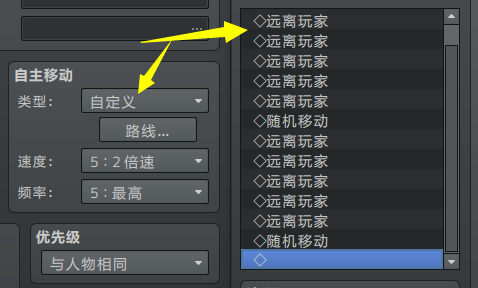


#### 5. 抓/逃设置

抓玩家，即移动路线为 接近玩家。

逃跑，即移动路线为 远离玩家。

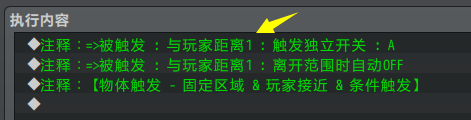
事件触发切换了事件页后，在指定事件页设置 自定义移动路线 即可。

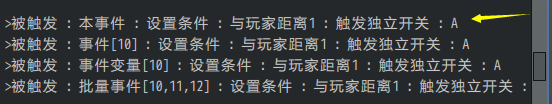


#### 6. 事件注释与插件指令 细节

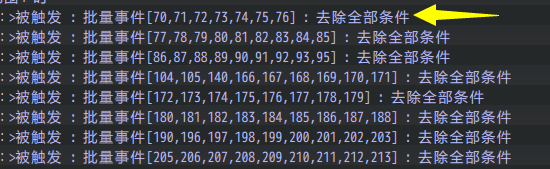
这里需要注意一下，

你可以通过 事件注释 添加条件，也可以通过 插件指令 临时添加条件。





但是，通过插件指令去除，**有时候 无法去除 事件注释 的触发条件**。



这是因为“离开范围时自动OFF”，会使得事件页切换回第一页，这样”与玩家距离1”的条件，就又被重新设置上去了。