## 概念

### 定义

**动画帧：**界面中的最小时间单位。1秒=60帧。

**帧刷新：**脚本中，每帧都会update刷新一次，这种过程称为帧刷新。

**帧数（FPS）：**在游戏过程中，并不是每秒真的会完美刷新60次，每秒实际的刷新的帧数称为FPS。

## 动画帧

**1）动画帧与FPS**

rmmv游戏基于动画帧，每帧对 **当前界面** 进行一次计算，变化游戏内的图形，1秒计算60次。

在游戏中按F2可以看到当前游戏 帧数（FPS）。



**2）动画帧与界面**

如果当前处于菜单界面，则战斗界面、地图界面都不做任何计算。因此这些界面处于暂停状态。

**3）动画帧与游戏时间**

游戏时间分为两种，一种是游戏内时间，一种是真实时间。

真实时间是根据计算机系统时间来决定的，与动画帧没有关系。

（可以参见脚本SceneManager.\_currentTime的定义。）

**4）动画帧与变速齿轮**

变速齿轮的原理，在于游戏请求时，系统每次都给一个欺骗的时间。

因为游戏1秒需要计算60次，则每1/60秒执行一次计算。

如果系统每1/30秒给游戏一次时间，则游戏变成了2秒才能计算60次。

游戏里的1秒 = 现实的2秒。所以游戏速度变慢了。

（这里的 变速齿轮插件 只改变动画帧时间，不会影响 真实游戏时间 的统计。）

## 并行事件

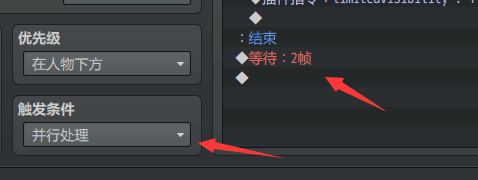
并行事件的所有动作，如果没有设置等待帧，那么将默认视为1秒执行60次。

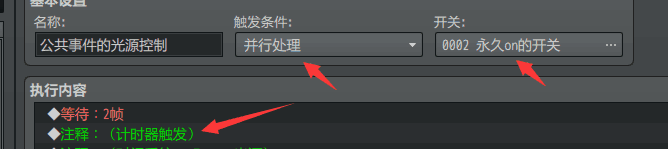
这也是并行事件拖慢游戏速度最大的原因，频繁地执行大量与时间没有关系的重复计算。

因此，当你建立一个并行事件时，**请务必优先考虑添加等待帧**。

等待1帧 = 1秒执行30次，等待2帧 = 1秒执行20次，等待3帧 = 1秒执行15次

公式为： 计算次数 = 60/ ( 等待帧 + 1 )





自动执行 与 并行事件的原理一样。

但是自动执行会阻塞你的其他事件动作。如果出现了bug，你会一直动不了，有助于发现问题。而并行事件没有直观的影响，所以需要时刻注意。