## 概念

### 简单原则

后面会介绍一大堆概念，可能会比较难理解。这里先说结论：

**要查看哪些插件是低、中、高消耗的插件，去看看”** **性能测试统计表.xlsx”**。

**如果你的游戏考虑手机端，低消耗流畅运行**，你需要：

1.确保每张地图的事件 < 80个。

2.确保单次插件指令执行数量 < 40个。

3.少用或者不用 中等消耗 或 高消耗 的插件。

4.并行事件尽可能 < 10个。

**如果你的游戏考虑PC电脑端，但是低配电脑也要流畅运行**，你需要：

1.确保每张地图的事件 < 200个。

2.确保单次插件指令执行数量 < 80个。

3.少用或者不用 高消耗 的插件。

**如果你的游戏几乎不考虑性能**，在较高配置的游戏本中运行，以上问题你全部不需要担心，直接肝就是了。

另外，变速齿轮可以使得所有消耗爆炸式上升，设置速度越快，消耗越大。

（高配电脑都能卡成2帧……）

### 硬件与软件

要玩一个游戏，必须具备两个东西：**硬件 和 软件**。

硬件就是你的电脑，有高配电脑，也有低配电脑。

软件就是你的游戏，有高效程序，也有垃圾程序。

**软件之所以要进行性能测试，是为了使程序更为高效，确保在低配电脑里，游戏也能流畅运行。**

插件也是如此，你安装的每个插件，都有自己的性能，有些插件由于计算量非常大，很可能在低配电脑中造成游戏卡顿。

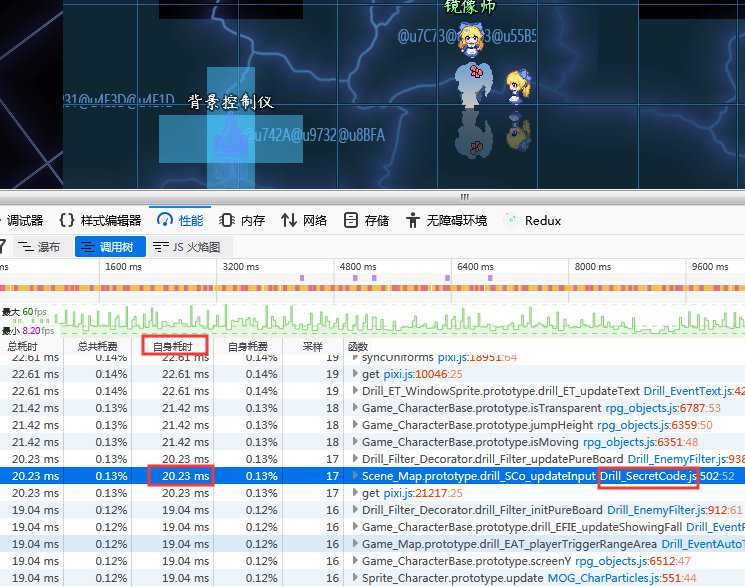
**所以，如果你对软件的性能有一定要求，你需要考虑是否排除一些高消耗的插件。**





**硬件性能**（图中是低配电脑╭(°A°`)╮）

（该电脑跑不动吃鸡，大型3d游戏都带不动，只能玩一些中低端的3d游戏。然而，该电脑却出人意料地能流畅运行英雄联盟，这令我有些敬佩。）



**软件性能**（图中是火狐浏览器的检测工具╭(°A°`)╮）

这里测试性能使用的是低配电脑，同样的插件，低配电脑消耗为20.23ms，而高配电脑的消耗为9.14ms。所以如果你的电脑不是特别低配，插件还是大可放心使用的。

高性能电脑固然支持清晰的画面和流畅的体验，但rmmv毕竟是2d游戏，如果2d游戏比3d游戏还卡，实在说不过去。优化是必要的。

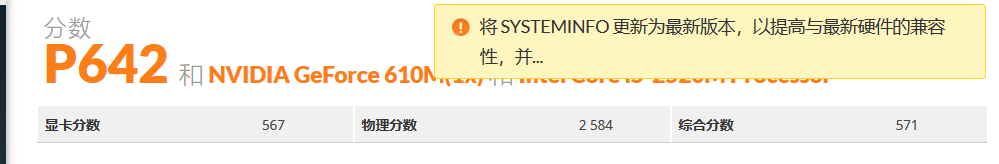
### 电脑性能

下图为**低配电脑**的鲁大师跑分：（综合分数为48456）



下图为**低配电脑**的3dmark跑分：（综合分数为571）

（结果在3dmark可查： <https://www.3dmark.com/3dm11/13658378> ）



另外，介绍一下标准游戏本的分数：（2018年的新本，7000左右软妹币）

下图为**游戏本**的鲁大师跑分：（综合分数为201441）



下图为**游戏本**的3dmark跑分：（综合分数为5559）

（结果在3dmark可查： <https://www.3dmark.com/3dm11/13658876> ）

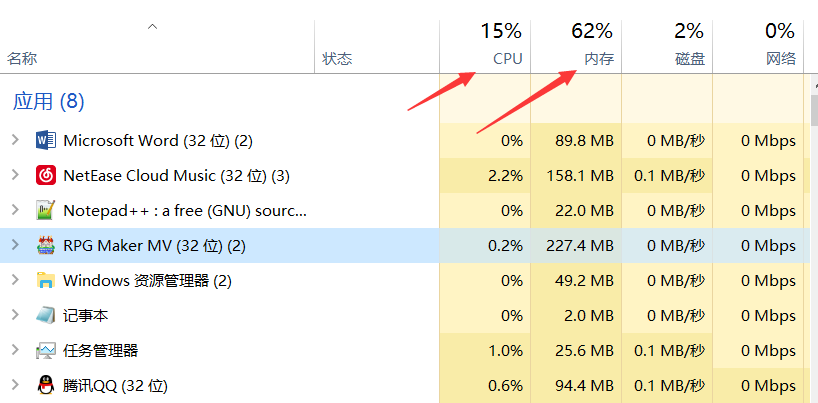


### 性能与内存

**性能就是游戏消耗速度。**

**内存就是游戏占用空间。**

打开win10的任务管理器，可以看到所有程序的性能和内存的关系。



内存属于瞬间爆炸型，系统的内存空间不够了，软件继续如果要求内存，会直接软件停止工作或死机。

性能属于经久不衰型，游戏速度慢了，会掉帧，卡顿，严重影响游戏体验，但是游戏本身就是不挂掉，一直卡。

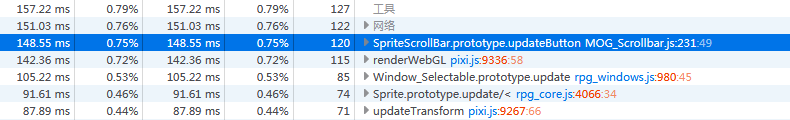
这里只讨论插件的性能占用情况，游戏的内存不计。

### 为什么要记录性能

由于的插件特别多，有许多插件计算量大，混杂在里面难以被发现。

单独关掉一个插件，也看不出明显的卡顿，所以必须通过内存工具来找。

比如下图的mog插件计算量是我从来都想不到的：



参照下面的20000.00ms检测条件的对照表：

|  |  |
| --- | --- |
| 范围 | 程度 |
| 0.00ms - 40.00ms | 几乎无消耗 |
| 40.00ms - 80.00ms | 低消耗 |
| 80.00ms - 120.00ms | 中消耗 |
| 120.00ms以上 | 高消耗 |

MOG\_Scrollbar（菜单滚动条）插件 的消耗量居然为148.55ms！

看来之后不得不把这个垃圾插件推翻重写了。

另外需要提一下的是，插件好比树枝树叶，而rpg\_core.js是最底层的树根，所以它的消耗自然是最大的，**因为集合了树枝树叶所有的计算量**。

## 帧数（FPS）

rmmv标准为60帧。在游戏中 **按F2** 可以调出帧数变化窗口。

以下是个人总结的FPS观感与体验：

|  |  |  |
| --- | --- | --- |
| 效果 | 帧数 | 感受 |
|  | 50~60 | 完美运行 |
|  | 40~50 | 稳定运行 |
|  | 30~40 | 流畅运行 |
|  | | |
|  | 26~30 | 轻微卡顿 |
|  | 22~26 | 掉帧严重 |
|  | 20~22 | 卡成狗 |
|  | 18~20 | 卡成翔 |
|  | 15~18 | 卡成PPT |
|  | 15以下 | 游戏没法玩了 |

如果帧数一直在30帧以上，还是比较靠谱的。

只要帧数低于30帧，游戏时就能感受到部分卡顿，对于玩家来说，可能会影响游戏体验。

**另外，在加载界面中，帧数可能会突然掉很多，这是因为加载时要初始化大量数据，分配内存，属于正常现象，加载完毕后会恢复稳定帧数。**

## 一些经验之谈

### 并行事件与插件

值得一提的是，并行事件 与 **持续执行** 的插件 在功能上是等效的。

但不同的是，插件由于是独立出去的脚本模块，可以进行性能监听或算法优化。

而并行事件基于纯事件的设置，执行次数**不可把控**，在优化与性能上存在明显劣势。如果你会js脚本，最好自己写相关插件去实现。

### 切换菜单时掉帧现象

**切换菜单掉帧属于正常现象**。切换掉帧后，会恢复平稳状态。

之前”界面.docx”中有介绍。切换界面后，上一个界面所有数据处于暂停状态。所有计算量都转移到下一个界面的刷新。

而从菜单界面转回到地图界面后，地图界面中大量贴图数据需要**重建**。所以就造成了帧数骤降，全力重建贴图数据。



其实这种重建现象非常常见，所有游戏都需要重建。只是不同的游戏重建的时间非常长，比如3d游戏。Rmmv为2d，所以不需要担心那么长的重建时间。

重建时间内，玩家肯定是不会耐心等它一直卡在那里的，那么如何让玩家耐心等呢？于是 游戏载入界面 就诞生了。

### 贴图变化的性能

贴图变化是最难评估插件性能的一种情况。下面因素会影响性能：

1.插件对贴图控制的计算量

2.图像投射显示的面积

3.图像刷新的频率

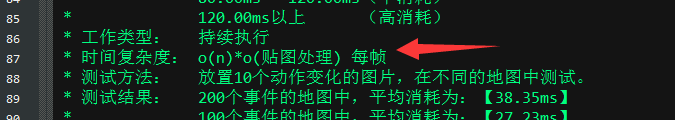
**有时候你可能会发现，插件性能里面，为什么明明是图片贴图的性能消耗，50个事件和200个事件的消耗截然不同，图片贴图明明和事件毫不相干**。其实这里有两个原因，一是计算机计算量(CPU)有限，大量事件的计算挤兑了资源分配，其他插件的需要更多时间等待资源分配。二是图形显示计算(GPU)有限，图片越多，GPU的显示控制的负担就越大。

我们都知道，图像是由一个个像素点组成的，每个点都存储了一条RGB（红绿蓝）数据。如果是png，则存储RGBA（红绿蓝+透明度）数据。

一个普通的行走图的资源是576x384的图像，那么程序把数据展开后，得到的是221184个像素点。一共 221184 \* 4 = 884736 个数据。

如果我要把一个图片旋转、缩放，那么红、绿、蓝、透明度数据就需要至少遍历一次，遍历一次80万的计算消耗就用掉了……

因此，性能测试中我会把 o(贴图处理) 单独列出来，因为这是一个非常不稳定的量。



当然，计算机并不会真的老老实实吧这80万次的计算全部执行，**如果计算负荷超过一定值，计算机会自主选择掉帧，”我不去计算图形，那么这么多的计算量也就不存在了”**。

但是，对玩家就不友好了，有时候直接卡成ppt。