## 插件大坑

以下是插件的血的教训，踩坑了将会耗费**巨量的时间和代价**去挽回。**千万不要做**：

1.修改了插件的文件名。

2.添加新插件后，读取未装插件的旧存档。

（如果只是修改插件配置，读取旧存档，也有部分问题；）

（可见“0.基本定义 > 数据更新与旧存档.docx”）

3.修改rpg\_core.js等内核文件的中脚本，不留注释和修改记录。

4.无视插件的先后顺序。

5.复制粘贴文件时，没有备份工程。

大部分插件错误都是因为配置错误产生的，当然也不排除插件本身的问题，如果你在上述情况之外仍然遇到了问题，可以找我修复插件的bug哦。

### 1.修改了插件的文件名

**原理概述**：插件的文件名在脚本中就是硬性规定的，脚本通过文件名识别插件。

**解决方法**：如果修改了插件名，那么赶紧改回去，文件名多任何一个符号、空格、大小写字母都会不识别。

|  |
| --- |
| **其他说明**：如果一定要修改名字也可以，去修改脚本中的代码，但是如果你的对游戏脚本.imported控制与parameters(“”)函数不熟悉，那么还是止步吧。 |

### 2.添加新插件后，读取未装插件的旧存档

**原理概述**：大部分插件中都有与存档一起存入的数据。读取存档时，插件也会向存档读取系统要数据，系统没有数据，那么插件会直接报错。

**解决方法**：删除旧存档，包括全局存储相关文件，新开游戏。

如果你只是修改插件配置，读取旧存档，可以去了解下“0. 基本定义 > 数据更新与旧存档.docx”

|  |
| --- |
| **原理详细**：存档是一系列插件、功能、数据的最大集合体。新开游戏时，系统会将所有插件的内容初始化，插件也会将初始信息加入系统中，并与存档建立了硬性关系。  **常见报错**：在读取旧存档时，容易引起定位错误，找不到id对应的内容，常见下图的错误：    **其他说明**：任何游戏基本都难以支持旧的存档。因为旧存档存储了大量未更新前的旧数据，这些数据会对游戏造成未知的影响。  不过也有一种补救措施，在你游戏中每个剧情点都设置一个开关，达到某个阶段关闭前一个剧情开关，开启下一个。这样，开新游戏就能直接进入指定剧情，而不需要靠存档进入了。比如某些小游戏，密码、秘籍解锁指定剧情。 |

### 3.修改rpg\_core.js等内核文件的中脚本，不留注释和修改记录

**原理概述**：没记录容易误事，过去改的脚本可能会折腾今天的你半死不活。

**解决方法**：养成记录的习惯，多写一点文字不费内存也不费钱更不费时间。

|  |
| --- |
| **其他说明**：论坛里面会有许多教你如何快速修改一些简单脚本的方法，比如系统的文字颜色之类的。这些方法很容易满足你一时的需求。但是，只要时间一长，你的旧工程死活不能运行某些插件，但是新开的工程可以运行，找不到原因。  一定要做记录，你修改了这一块要标注哪个文件哪一行的位置。 |

### 4.无视插件的先后顺序

**原理概述**：所有插件都有先后的载入初始化顺序。

**解决方法**：遵循示例的顺序进行排序，或者看插件的”★”要求，控制顺序。

|  |
| --- |
| **其他说明**：有些需要核心的插件会对核心进行检查，如果你的顺序错了，插件会提示你缺少核心。但是，也存在需求性不强的半扩展插件，如果这类插件没找到自己的扩展插件，则插件会自己关闭相关的扩展功能。 |

### 5.复制粘贴文件时，没有备份工程

**原理概述**：Windows系统会在你覆盖文件的时候会问你是否覆盖。覆盖了，旧文件就彻底没了。

**解决方法**：没有解决办法。

|  |
| --- |
| **其他说明**：过一段时间，给自己的工程打一个压缩包，这样能够挽回一定的损失。许多群友经常看见了更好的素材、音乐，就会立即把当前的换掉，装上新的。由于覆盖操作过于顺手，以至于原来的素材再也找不回了。 |