

目录

一、	Mtool 教程	2
1.	注入游戏	2
2.	翻译/导出翻译	2
3.	关于 mtool 的更新	3
二、	申请百度 API	3
三、	mytrs 教程	4
1.	基本使用说明	4
a)	设置配置文件	4
b)	进行机翻	5
2.	进阶使用	6
a)	清除无用字符功能	6
b)	翻译修正功能	7
c)	二次翻译功能	7
四、	我的其他工具	8
1.	untrsfix	8
2.	trs_extractor	8
a)	基本使用	9
b)	进阶使用	9

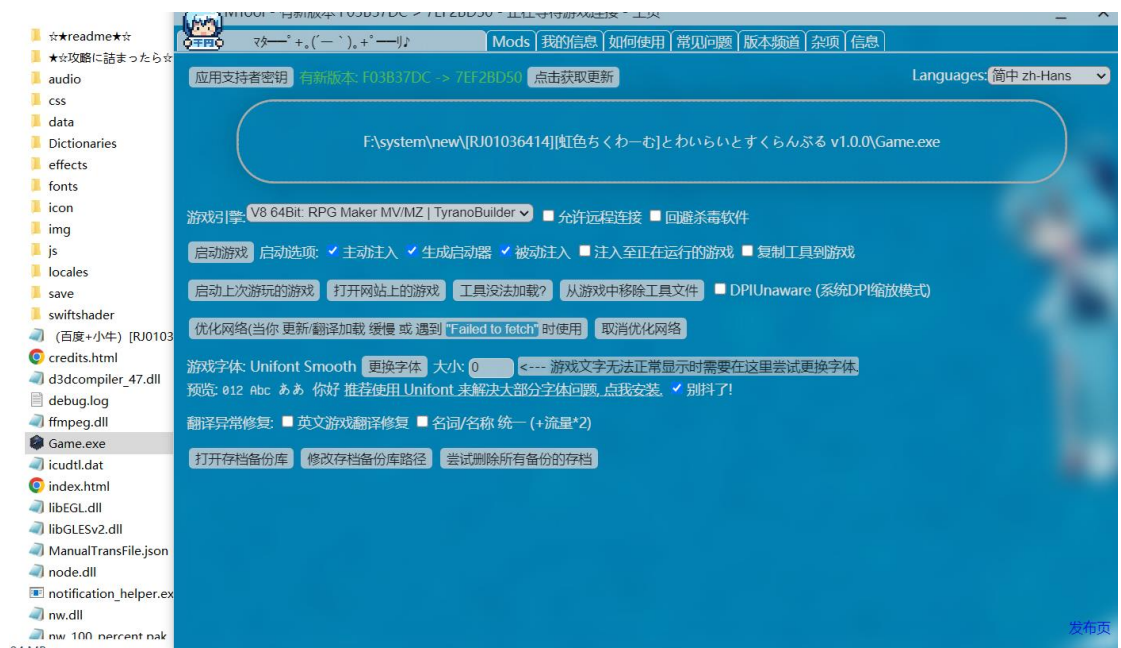
- 先宣传一下我的 QQ 群: 852682626, 用于 mytrs 技术交流和机翻文件分享。禁止涉黄涉政, 在群不会被爆破的基础上可以随意水群。

一、Mtool 教程

作者爱发电网址: <https://afdian.net/a/AdventCirno>

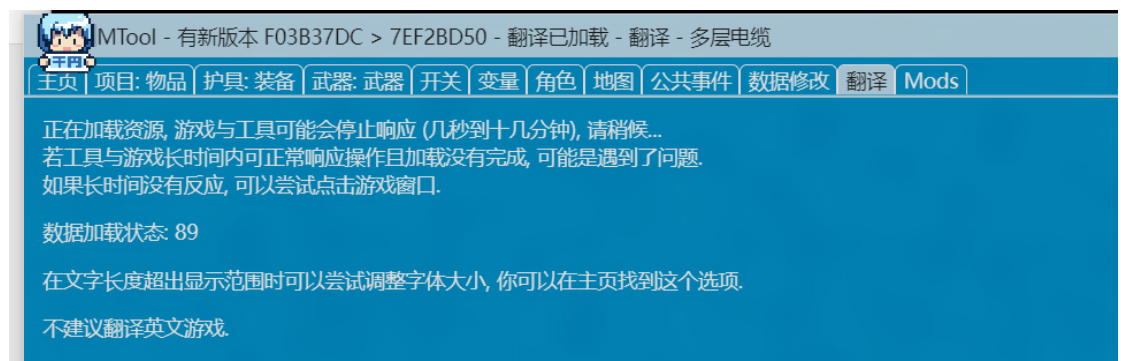
1. 注入游戏

如图, 把游戏的可运行文件 (exe) 拖入到 mtool, 点启动游戏即可。如果选择了被动注入, 以后打开游戏都会自动打开 mtool。否则需要用“与工具一同启动.bat”打开游戏。



2. 翻译/导出翻译

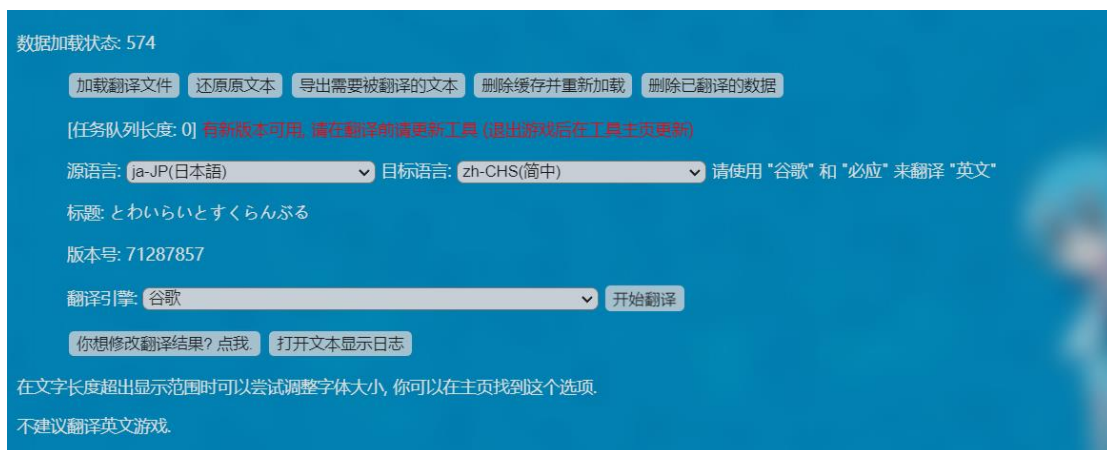
利用 mtool 打开游戏后, 选择“翻译”界面, 工具会自动加载游戏内的文本, 等待加载完成。



- 如果想使用 mtool 自带翻译功能, 需要保证工具是最新版本, 并且免费用户只能用谷歌翻译 (效果极差)。自带翻译会在游戏根目录生成一个“TrsData.bin”文件, 其中储存着加密过的机翻文本。非免费用户翻译的机翻文本只有使用

对应支持者密钥才能读取，也就是说不能分享。

- b) “加载翻译文件”为手动加载翻译文件，通常来说，如果游戏根目录内存在“TrsData.bin”文件，在游戏启动时会自动加载翻译文件，不需要手动加载。
- c) “导出需要被翻译的文本”会将游戏内文本导出到根目录下名为“ManualTransFile.json”的文件内，以供用户自行翻译。



3. 关于 mtool 的更新

通常来说，mtool 更新是为了修复某些新发售游戏内存在漏翻的情况，作者大大及其勤勉，所以更新很快，也很频繁。

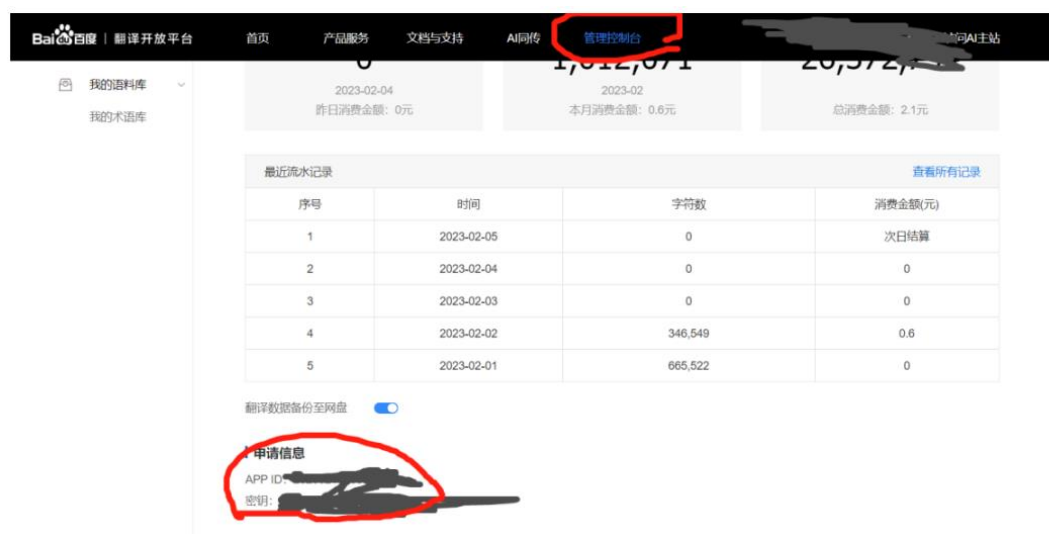
当你发现你玩的游戏内，有很多文本都没有被翻译，而且每次没被翻译的文本都不同，就说明你需要更新 mtool 了。

二、申请百度 API

自行导出翻译，需要有翻译引擎的 api 账号。

百度翻译 api 的注册，网站：<https://fanyi-api.baidu.com/>，注册需要实名认证，并且会询问你公司名称和 app 用途等信息，填个人使用即可。

注册号之后去管理控制台就能看见自己的 APP ID 和密码了，注意不要泄露给别人。









需要注意的是，百度翻译 api 有高级版和标准版两种版本，高级版每个月 100W 免费字符，QPS（每秒请求数）为 10，而标准版是每月 5W 免费字符，QPS 为 1，所以一般不考虑标准版。

三、mytrs 教程

mytrs 是我个人写的一款能够翻译“ManualTransFile.json”类格式文件的工具，目前支持百度，彩云和小牛三种翻译引擎。Github 地址：<https://github.com/stevetr/mytrs1>

1. 基本使用说明

工具压缩包内应该存在这些文件

 config.json	2023/6/16 11:29	JSON 文件	2 KB
 dict.json	2023/6/15 22:54	JSON 文件	33 KB
 mytrs_v2.1.exe	2023/6/16 19:11	应用程序	6,804 KB
 mytrs_v2.1.py	2023/6/16 19:07	PY 文件	20 KB
 必看!! 使用说明.txt	2023/6/16 11:37	文本文档	8 KB
 更新记录.txt	2023/6/16 19:15	文本文档	7 KB

a) 设置配置文件

初次使用前，以及每次更新工具后，需要打开配置文件 config.json 进行设置。

其中必须要设置的有

{	
"backupengine": [
"caiyun",	二次翻译使用的引擎，详见进阶使用，不想仔细看的话，就设置成[]，不使用此功能。
"xiaoniu"	
],	
"recordtime": {	
"year": 2023,	修改为当前年月
"month": 4	
},	
"enginelist": [使用的翻译引擎列表，把你想用的翻译引擎填进来
"baidu",	注意是英文引号，以及每一项之间要带英文逗号，
"baidu2",	从前到后依次使用这些引擎，当前一个引擎报错或
"caiyun",	字符耗尽，会自动切换到下一个。
"xiaoniu"	最后一个的后面不能有逗号
],	
"baidu": {	翻译引擎的名称，也就是填在上面列表里的东西
	百度引擎名称里必须要有关键字"baidu"，彩云有"caiyun"，小牛有"xiaoniu"，不能混搭，比如"baiducaiyun"
"dic": "dict.json",	翻译修正功能，填字典文件名，不使用填 0，详见进阶使用
"appid": "自己填",	api 的 id，在百度翻译控制台看
"appkey": "自己填",	api 的 key
"from_lang": "jp",	从什么语言，jp 日语，en 英语，zh 中文
"to_lang": "zh",	到什么语言
"NPQ": 3000,	每次翻译请求，文本的长度，最大 6000，不建议设的太高
"QPS": 10,	翻译引擎的 qps，高级版 10，标准版 1
"limits": 996000,	每月字符消耗上限，高级版 1000000，建议在此基

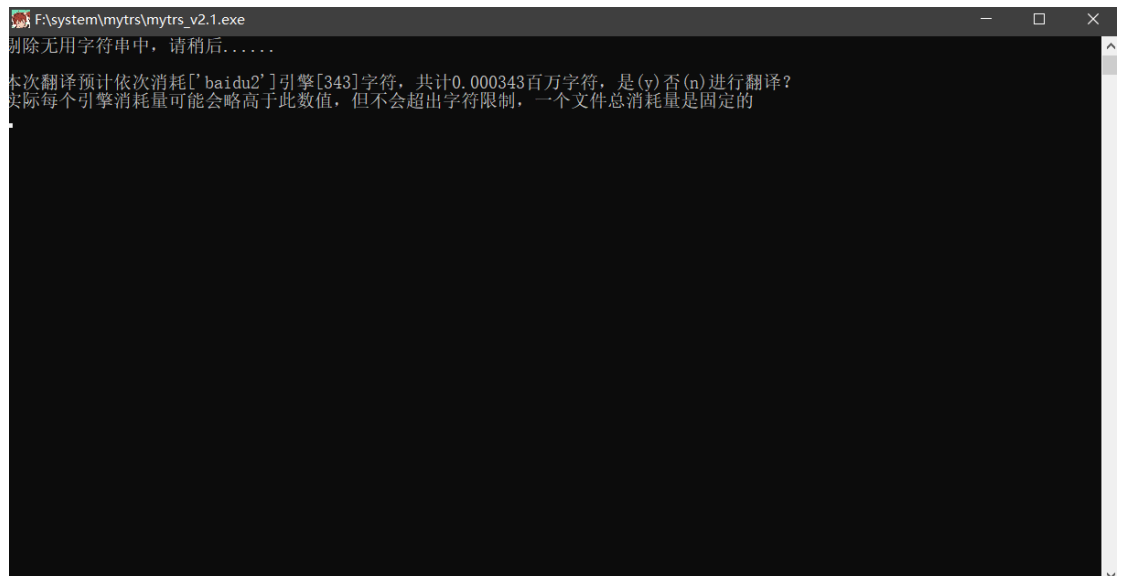
<pre> "usage": 0 }, "baidu2": { }, "caiyun": { "dic": "dict.json", "token": "自己填", "direction": "ja2zh", "NPQ": 3000, "limits": 980000, "usage": 0 }, "xiaoniu": { "dic": 0, "dictNo": "14d9e11102", "apikey": "自己填", "from_lang": "ja", "to_lang": "zh", "NPQ": 3000, "QPS": 50, "limits": 996000, "usage": 0 } } </pre>	<p>础上减去一点，留作保险</p> <p>使用本工具本月消耗的字符数，会控制在不超过 limits, 月初自动清零（在其他地方消耗的不会统计在内）</p> <p>如果你有两个或以上百度 api，命名示例</p> <p>后面这些设置，如果你不用这些引擎，可以不管彩云小译</p> <p>相当于百度的 appid+appkey</p> <p>从语言到语言，代码官网查</p> <p>限制为 5000</p> <p>小牛翻译，之前有每天签到领 50w 字符，现在没了，不推荐</p> <p>小牛自带字典，需自行前往官网设置，不用的话，设为 0</p> <p>限制为 5000</p>
---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------	---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

b) 进行机翻

把 mtool 导出的“ManualTransFile.json”复制到 mytrs 所在文件夹内，双击打开 mytrs.exe 即可，如果会用 python，可以运行 py 文件，不会用的话，py 文件删掉也无所谓。

打开后你会看到这个界面

“剔除无用字符串中，请稍后……”表示工具正在剔除掉“ManualTransFile.json”文件中不需要翻译的文本，具体剔除规则，见进阶使用。剔除完成后，会显示翻译整个文件需要的字符数，以及预计要用到的翻译引擎。输入 y 后按回车，开始翻译。



每个引擎翻译结束后, 会进行利用字典文件进行翻译修正 (详见进阶使用), 全部翻译完成后, 会进行二次翻译 (详见进阶使用), 按 y 进行二次翻译或按 n 不进行翻译直接导出翻译结果 (按 n 会提示翻译出错, 但实际没问题)。翻译的结果会自动保存在 exe 所在的文件夹内, 文件名为“TrsData.bin”。把它复制到游戏根目录, 用 mtool 打开游戏就会自动进行导入翻译了。



2. 进阶使用

a) 清除无用字符功能

官方命名 cleaner, 作用是节省翻译引擎字符量消耗, 以及避免可能的翻译引擎报错。开启后有两种工作模式。以“ManualTransFile.json”内的行为单位, 日语模式会清除掉所有不含日语假名的文本行 (日语汉字不算日语假名); 英文模式会清除掉所有不含大小写英文字母的文本行。日语和英语模式不能同时打开。本功能在在 config.json 中有三个配置项。

```
{
    "cleaner":1,      是否开启 cleaner, 1 开启 0 关闭
    "ja":true,       日语模式, true 开启, false 关闭
    "en":false       英语模式
}
```

b) 翻译修正功能

或者叫字典功能，和 mtool 的 mods 中的翻译修正相同，需要单独的 json 文件，并在 config.json 内对应翻译引擎的"dic"项设置其文件名。

文件内的格式如下

```
{
    "B1":{
        "C11":"A11",
        "C12":"A12",
        "C13":""
    },
    "B2":{
        "C21":"A21",
        "C22":"A22",
    }
}
```

在翻译引擎翻译结束后，会自动查找译文中含有 B 的文本，如果它的原文含有对应的 A，就自动将所有的 B 替换对应的为 C。A 可以是空字符串 ("")，这时不会检测原文，直接将 B 替换为 C。对单行文本来讲，存在生效先后顺序，某个 AC 对生效后，后续 AC 对均不会生效

如，翻译文本在 trsdata 内的格式为

"asdas**A11**sd**A22**"： "wdas**B2**dsd**B1**dds**B1B2**"

在应用字典以后，会变为

"asdas**A11**sd**A22**"： "wdas**C22**dsd**C11**dds**C11C22**"

通常来说，不同翻译引擎的 B 会不同，所以不同引擎的字典文件不能通用。

c) 二次翻译功能

对于某些文本，翻译引擎会翻译出异常结果，如图

```
"あっ・・・かっ！？・・・あああああっ！？！？"："啊.....啊！？啊啊啊啊啊！？！？",
"いやあああああ——っ！！"："啊啊啊啊啊啊啊啊！！",
"う・・・ああああ・・・うわああ・・・っ！！"："啊啊啊啊啊啊啊啊啊啊啊啊！！",
"「うああああ・・・あっ！"："啊啊啊啊啊啊啊啊！！",
"あっ！？くはあああああああああっ！！！！"："啊！？啊啊啊啊啊啊啊啊啊啊！！！！",
"ふあああああああああ——っ！！！！"："啊啊啊啊啊啊啊啊啊啊！！！！",
"「あああああ"："啊啊啊啊啊啊",
"・・・ひあああああああ——っ！！！！"："啊啊啊啊啊啊啊啊啊啊！！！！",
"くあああああああ——っ！！！！"："啊啊啊啊啊啊啊啊啊啊！！！！",
"「う・う・う・うわあああああっ！！！！"："啊啊啊啊啊啊啊啊！！！！"！！？",
"あああああああああ——っ！！！！"："啊啊啊啊啊啊啊啊啊啊！！！！",
"「あああ・・・も、もう許せえ・・・！"："啊啊啊啊啊啊啊啊啊啊啊啊啊啊啊啊啊啊啊啊啊啊啊啊啊啊啊啊啊啊！！",
"「あああ・・・あああ・・・ああ・・・っ！！！！"："啊啊啊啊啊啊啊啊啊啊啊啊啊啊啊啊啊啊啊啊啊啊！！",
"くはあああああああああ～っ！！！！"："啊啊啊啊啊啊啊啊啊啊～！！！！",
"うあああああああっ！！！！"："啊啊啊啊啊啊啊啊！！！！",
"あああああああっ！！！！"："啊啊啊啊啊啊啊啊！！！！",
"やあああああああああ——っ！！！！"："啊啊啊啊啊啊啊啊啊啊！！！！",
"うわあああああああああっ！！！！"："哇啊啊啊啊啊啊！！！！",
"ぐあああああ——っ！！！！？"："啊啊啊啊啊啊！！！！？",
"「うわあああ・・・っ！！？やめっ・・・"："哇啊啊啊啊啊啊啊啊啊啊啊啊啊啊啊啊啊啊啊啊啊啊啊啊啊啊啊啊啊啊",
"くあああああ——っ！！！！"："啊啊啊啊啊啊！！！！",
"くあああああ！！！！"："啊啊啊啊啊啊！！！！",
"うはあああああ——っ！！！！"："啊啊啊啊啊啊啊啊！！！！",
"これ以上、わしの巫力をお・・・くあああああっ！！！！"："再加上我的巫力啊啊啊啊啊啊啊啊！！！！",
"ふあああああああ——っ！！！！？"："啊啊啊啊啊啊啊啊！！！！？",
```

本功能的作用就是尽可能减少这类文本，在 config.json 有三个配置项

```
{
    "src_rep": 3,
    "dst_rep": 5,
```


13

字

檢

11

证

不

四

1.

相

“L

具

- 后

- 笙

等

2.

并

其

$$\{$$

"

"

"

"

"

"

是否导出为容易看懂的文件的格式, true 会导致 data 文件夹大小变大, 具体来说, 从 50M 变为 150M。


```
}
```

a) 基本使用

设置好两个 data 路径，打开 exe 文件即可。工具处理结束后，将 exe 所在文件夹内的 data 文件夹，覆盖到新版游戏内即可。

随后可以用 mtool 分别导出两个版本游戏的待翻译文件，将已汉化旧版游戏的待翻译文件改名为 TrsData.bin，使用 untrsfix 即可得到游戏版本更新后，新增的文本。把它翻译后用 mtool 加载，即可得到一个“旧版的汉化文本+机翻的新增文本”的新版游戏。

b) 进阶使用

由于本工具涉及到对游戏数据的修改，所以有一定可能会使游戏出现 bug，包括但不限于：旧版游戏中已汉化的部分文本未能提取到新版游戏（漏翻），对话文本错位，丢失版本新增事件等等。这些 bug 并不是很容易被发现，但一旦出现，影响巨大。

以下 config 设置项便是为了减少这些 bug 而设置的，当出现 bug 时，需要手动调整这些配置项，来减少 bug 的产生。没有编程基础的人可能难以读懂下面的内容

```
{  
"diff":0.1,  
"difflist":true,  
"upperkey":["list"],  
"difflist2":true,  
}
```

先来说明工具运行逻辑（python 语言为例）：

对于工具能适用的游戏，它们的游戏数据是以 dict（对应 json 的大括号{}）和 list（对应中括号[]）的组合的形式储存在 json 文件内，这些数据中含有游戏的文本字符串，同时也有一些用来标记这些字符串在何处出现的数据。

以每一个 dict 和 list 为一级，工具会自动匹配 json 内的数据格式，搜索到旧版数据中，含有中文汉字的字符串所在的路径，并比对新旧版游戏这个路径途中所有的数据。如果除了这些汉字字符串所在路径以外的所有数据都相同，会标记这个路径为“same”，被标记为“same”的路径，就会将对应的字符串覆盖到新版游戏中。

“diff”：用来调整在一个 dict 中，有多少数据不同仍会被标记为 same，用于减少漏翻情况的发生。

“difflist”：为 true 时，当 list 长度不匹配，且 list 内存在文本数据时，会直接用已翻译的旧版 list 覆盖新版 list。为 false 时，会导致这种文本漏翻，但是设置为 true 可能会导致 bug，丢失版本新增事件。

“upperkey”：是一个 list，作用是对上面的功能增加限制，只有在这个 list 的上级路径是 dict，且 dict 的 key 包含在 upperkey 内时，才会启用上面的功能。整体来讲，会降低 bug 几率

“difflist2”：在 list 长度不相等，但是不满足上面两项共同判断的条件时，是否对文本进行替换。开启此功能，可能会导致涉及到的文本错位，但关闭此功能会导致这部分文本漏翻。

以某个游戏举例来说：

关闭“difflist”，出现了游戏中部分文本漏翻的情况。但是只打开“difflist”，又会

导致回想房间丢失版本更新后多出的新 cg。这是因为储存 cg 的数据是一个 list, 多出的新 cg 会导致新旧两个版本的 cg list 长度不匹配, 而在开启"difflist"的情况下, 选择保留了旧版的 cg list。

"upperkey"是为了解决上述问题新增的判断条件, 储存游戏文本的 list 和储存游戏事件的 list 分属不同的层级, 它们的 key 是不同的, 那么只需筛选出文本 list 所对应的 key, 就可以实现只对游戏文本应用"difflist"功能。"upperkey"就是用来储存这些文本 list 所对应的 key 的。

但是增加"upperkey"后又会出现新问题, 那就是被"upperkey"屏蔽掉的 list, 如果它的内部也存有游戏文本, 这些文本就会被漏翻掉。以我用来测试的游戏来说就是回想房间内的所有文本 (CG 名称和选项等) 都被漏翻了。而这些文本都有一个特点, 就是按顺序从上往下替换, 基本不会出错。

开启"difflist2"就可以实现这一功能, 最终在三个条件共同作用下, 我实现了“游戏文本没有漏翻, cg 没有缺失”的最理想效果 (至少我还没发现其他 bug)。