

Mao 汉化日志 2023/10/27

此日志将较简略地叙述关于《墙国：猫的遗产》汉化工作中 Monobehaviour 文本提取与文件导回的过程。

汉化探索者：Dorin

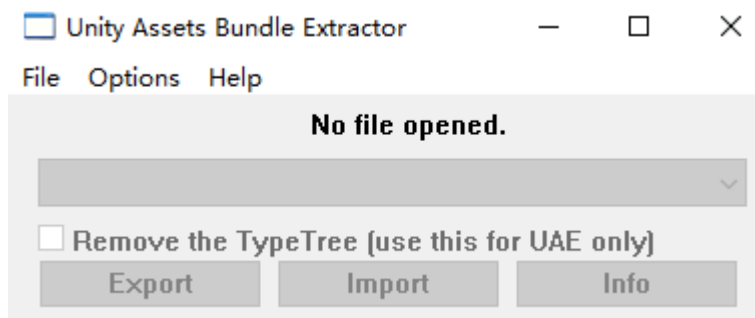
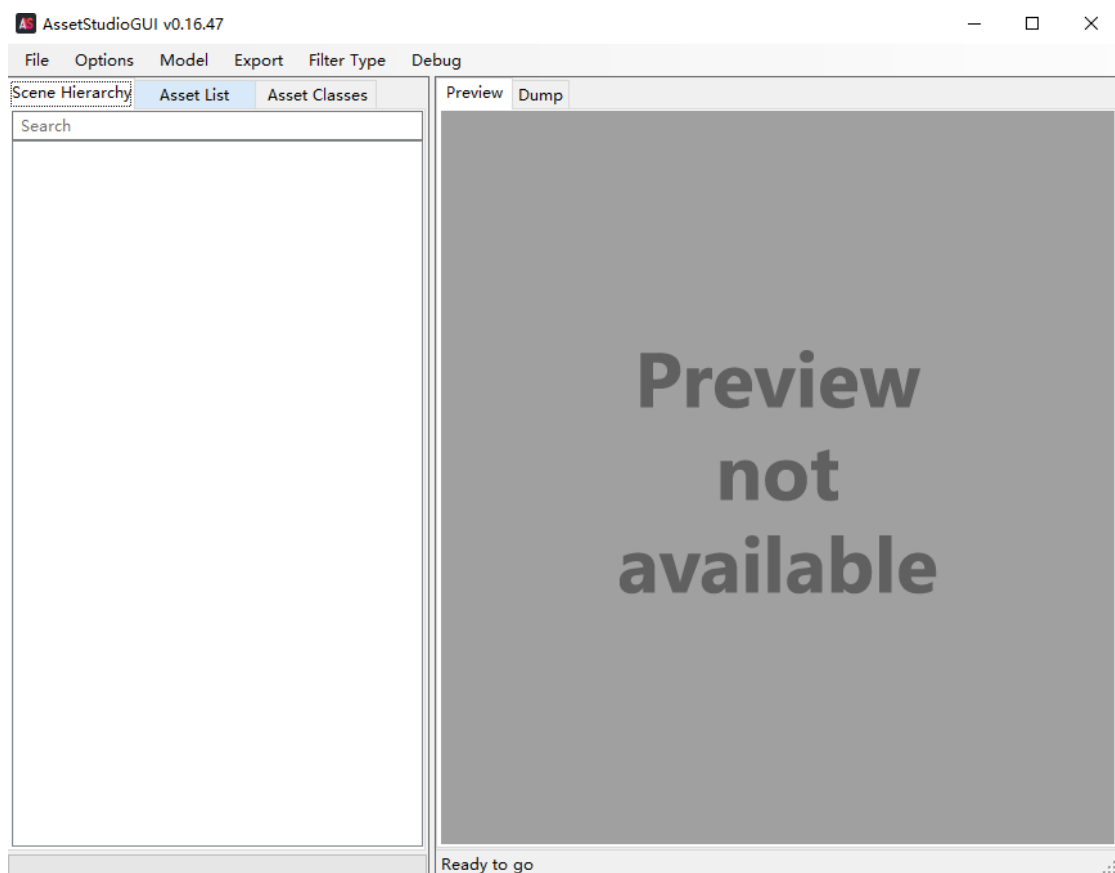
游戏版本：1.6.7.2（之前版本亦适用）

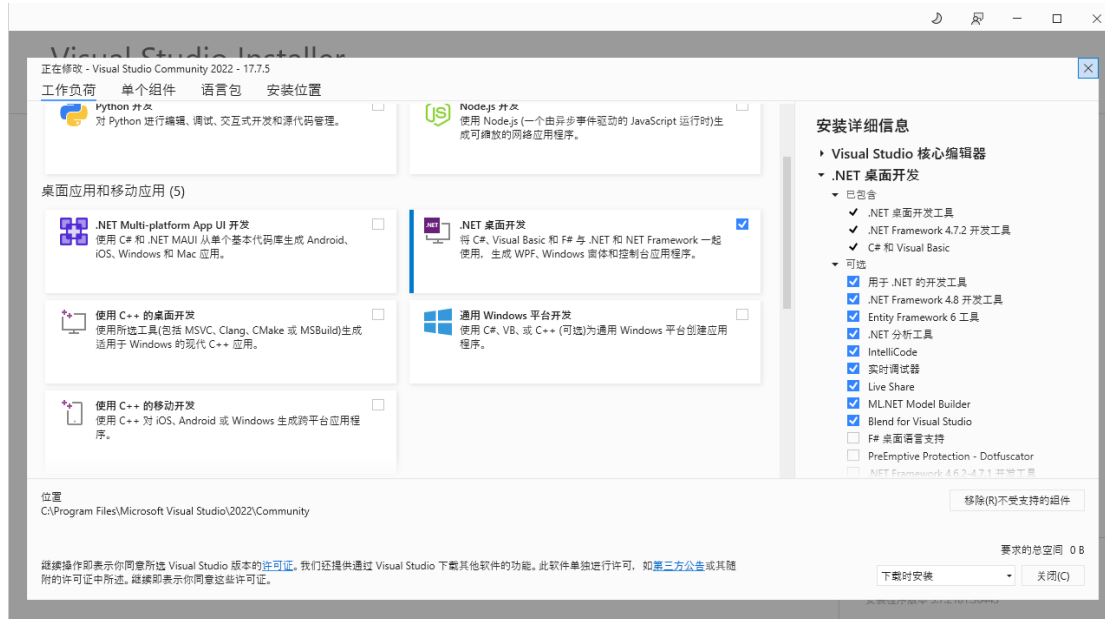
1. 所需工具

AssetStudioGUI-v0.16.47

AssetBundleExtractor-v2.2stabled (请使用给定版本, 其他版本会造成未知问题)

Visual Studio 2022 或任何能打开 json 文件的编译器

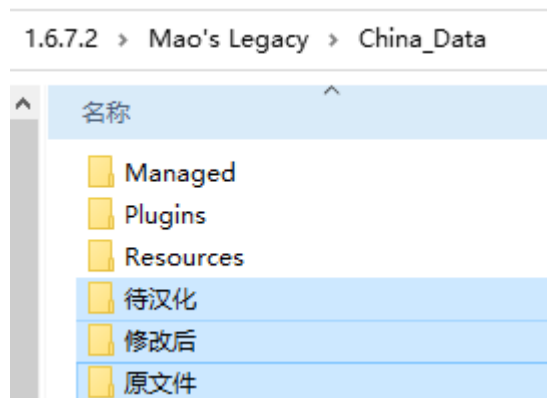




2. 提取流程

2.1 事前准备

打开 Mao's Legacy\China_Data 文件夹，新建三个文件夹，分别命名为：待汉化、修改后、原文件。

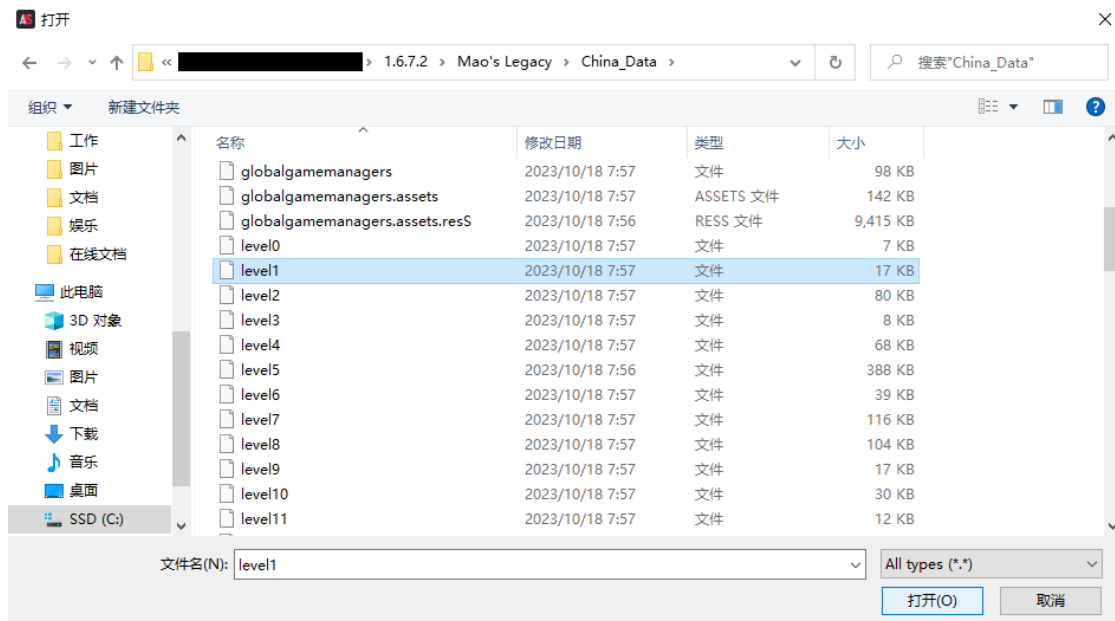
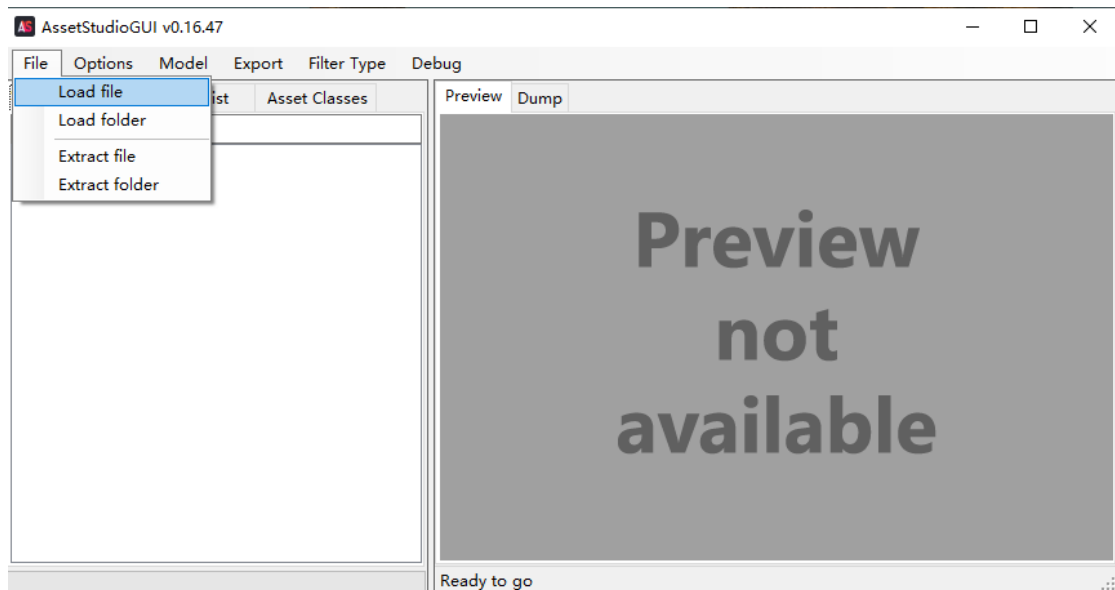


将 level、resources.assets、sharedassetsX.assets 三类文件复制进“原文件”文件夹以进行备份。

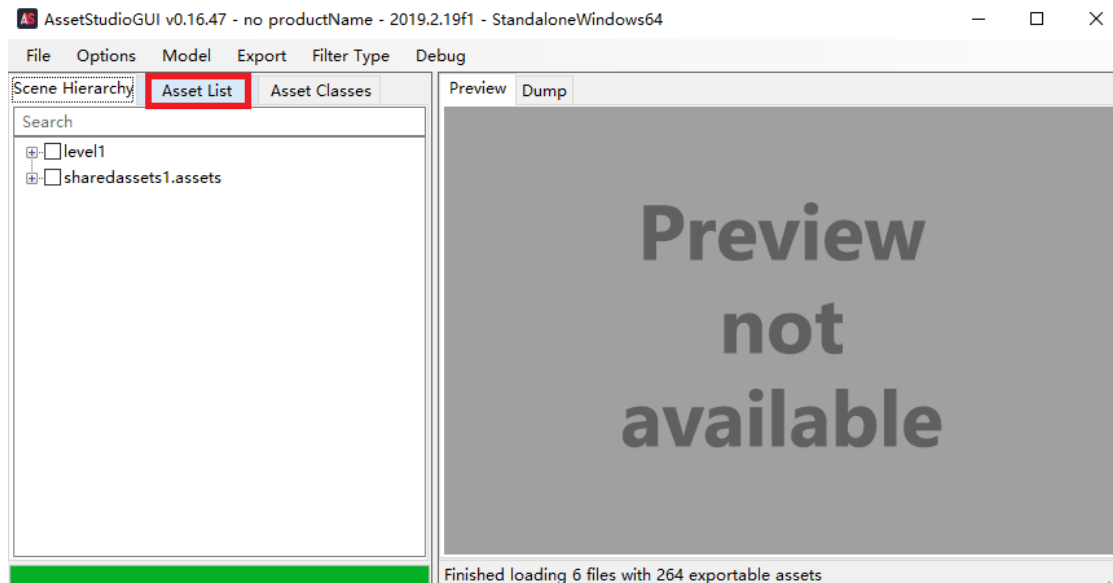
Mao's Legacy > China_Data > 原文件				搜索“原文件”
名称	修改日期	类型	大小	
level0	2023/10/18 7:57	文件	7 KB	
level1	2023/10/18 7:57	文件	17 KB	
level2	2023/10/18 7:57	文件	80 KB	
level3	2023/10/18 7:57	文件	8 KB	
level4	2023/10/18 7:57	文件	68 KB	
level5	2023/10/18 7:56	文件	388 KB	
level6	2023/10/18 7:57	文件	39 KB	
level7	2023/10/18 7:57	文件	116 KB	
level8	2023/10/18 7:57	文件	104 KB	
level9	2023/10/18 7:57	文件	17 KB	
level10	2023/10/18 7:57	文件	30 KB	
level11	2023/10/18 7:57	文件	12 KB	
level12	2023/10/18 7:57	文件	12 KB	
level13	2023/10/18 7:56	文件	333 KB	
level14	2023/10/18 7:57	文件	30 KB	
level15	2023/10/18 7:57	文件	22 KB	
level16	2023/10/18 7:57	文件	29 KB	
level17	2023/10/18 7:57	文件	10 KB	
level18	2023/10/18 7:57	文件	9 KB	
level19	2023/10/18 7:57	文件	23 KB	
level20	2023/10/18 7:57	文件	84 KB	
level21	2023/10/18 7:57	文件	9 KB	
resources.assets	2023/10/18 7:56	ASSETS 文件	3,483 KB	
sharedassets0.assets	2023/10/18 7:57	ASSETS 文件	9 KB	
sharedassets1.assets	2023/10/18 7:56	ASSETS 文件	768 KB	
sharedassets2.assets	2023/10/18 7:57	ASSETS 文件	212 KB	
sharedassets3.assets	2023/10/18 7:57	ASSETS 文件	6 KB	
sharedassets4.assets	2023/10/18 7:57	ASSETS 文件	175 KB	
sharedassets5.assets	2023/10/18 7:56	ASSETS 文件	1,130 KB	
sharedassets6.assets	2023/10/18 7:57	ASSETS 文件	19 KB	
sharedassets7.assets	2023/10/18 7:57	ASSETS 文件	35 KB	
sharedassets8.assets	2023/10/18 7:57	ASSETS 文件	17 KB	

2.2 使用 AssetStudioGUI 打开 Mao's Legacy\China_Data 文件夹下的 level 文件

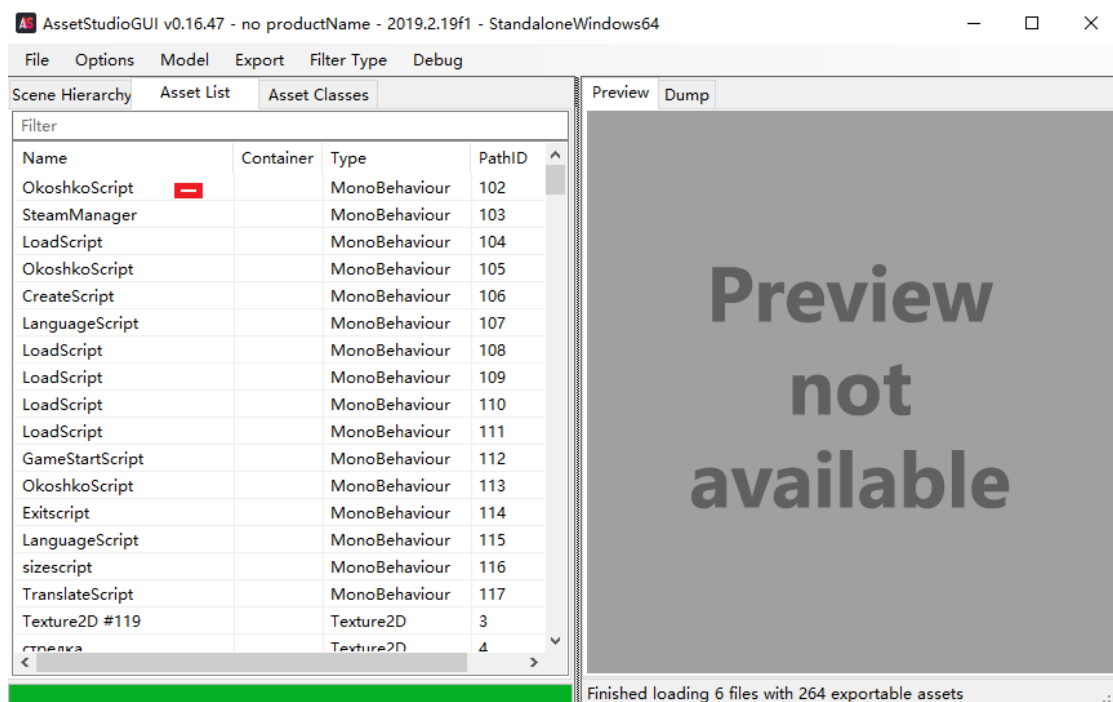
在此处，我们选择打开 level1 文件。



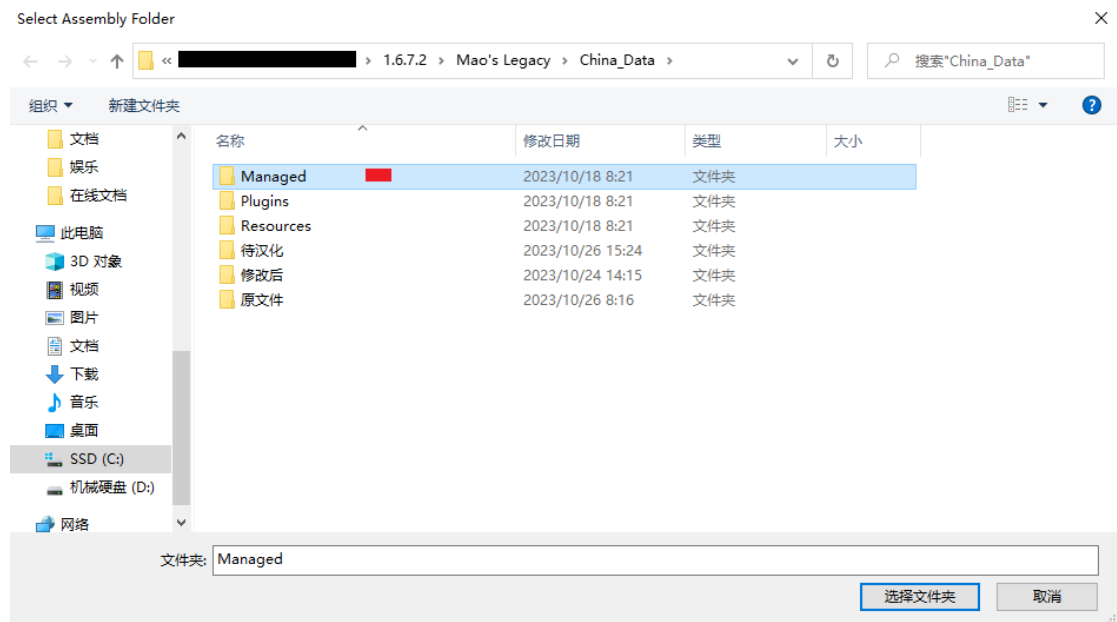
选择 Asset List



在列表中选择任意一个 MonoBehaviour 类型文件



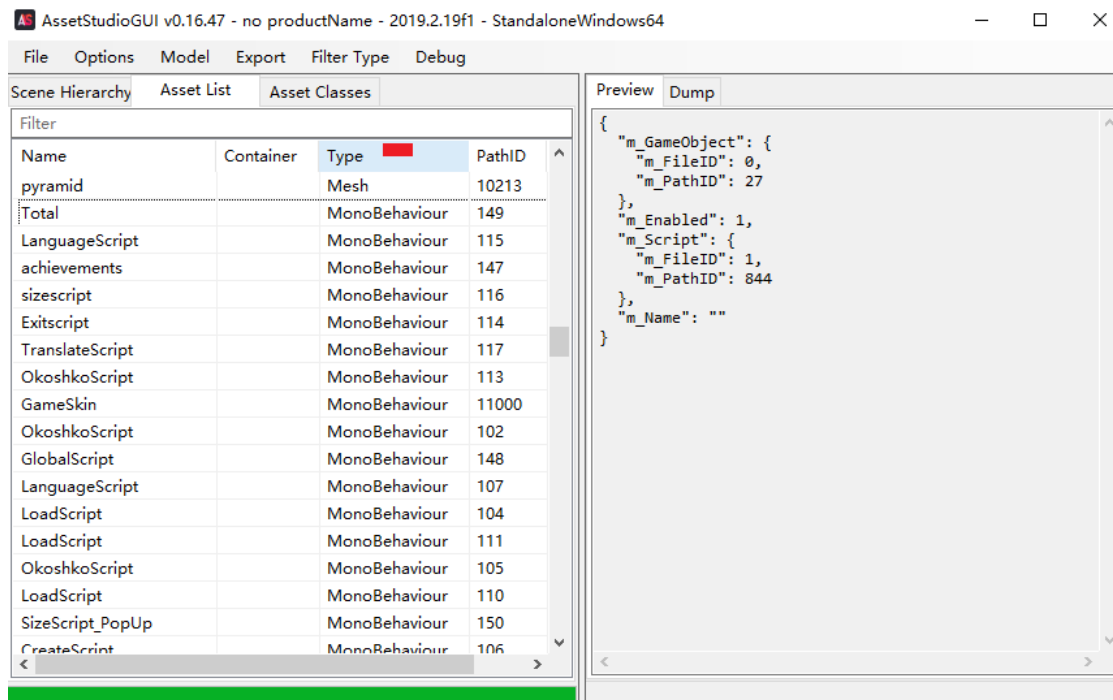
点击后弹出一个 Select Assembly Folder，请选择 Mao's Legacy\China_Data 下的 Managed 文件夹，以将 Monobehaviour 文件与其中的 Unity Assembly 引用建立链接。



每次新打开一个包含 Monobehaviour 文本的 level 文件，都要求进行此动作。

2.3 查看需要汉化的 Monobehaviour 文件

筛选 Monobehaviour 类型的文件。



一般来说，需要导出的文件名为“OkoshkoScript”、“TranslateScript”、“TutorialScript”等，手动点击相应文件，并在右侧的 Preview 中判断它们是否需要汉化。若需要汉化，则记录下相应文件的位置、名称以及 Path ID。

Scene Hierarchy
Asset List
Asset Classes

Filter

Name	Container	Type	PathID
achievements		MonoBehaviour	147
sizescrypt		MonoBehaviour	116
Exitscript		MonoBehaviour	114
TranslateScript		MonoBehaviour	117
OkoshkoScript		MonoBehaviour	113
GameSkin		MonoBehaviour	11000
OkoshkoScript		MonoBehaviour	102
GlobalScript		MonoBehaviour	148
LanguageScript		MonoBehaviour	107
LoadScript		MonoBehaviour	104
LoadScript		MonoBehaviour	111
OkoshkoScript		MonoBehaviour	105

Preview

Dump

```

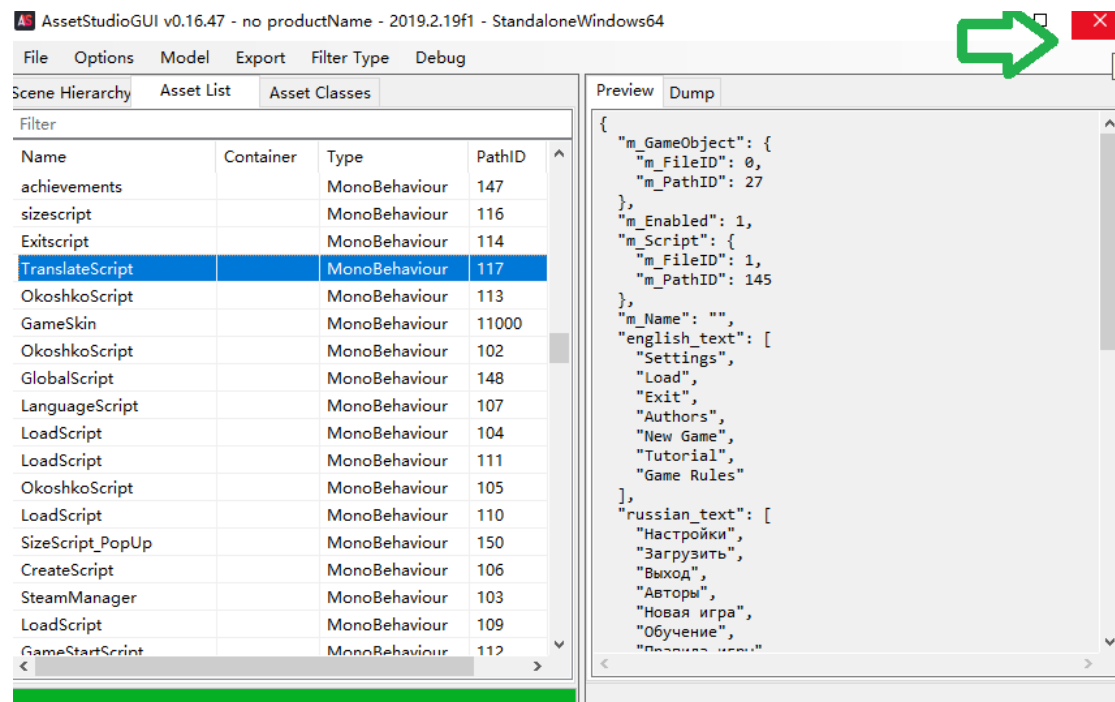
{
  "m_GameObject": {
    "m_FileID": 0,
    "m_PathID": 27
  },
  "m_Enabled": 1,
  "m_Script": {
    "m_FileID": 1,
    "m_PathID": 145
  },
  "m_Name": "",
  "english_text": [
    "Settings",
    "Load",
    "Exit",
    "Authors",
    "New Game",
    "Tutorial",
    "Game Rules"
  ]
}

```

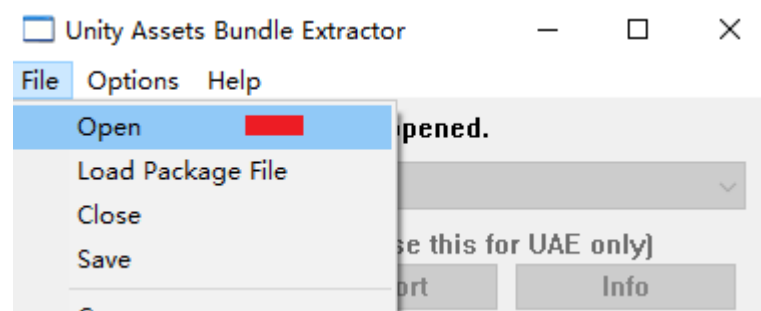
比如，我在 TranslateScript 中找到了游戏主菜单的按钮，我需要记录的信息为：level1 中的 TranslateScript，PathID-117。

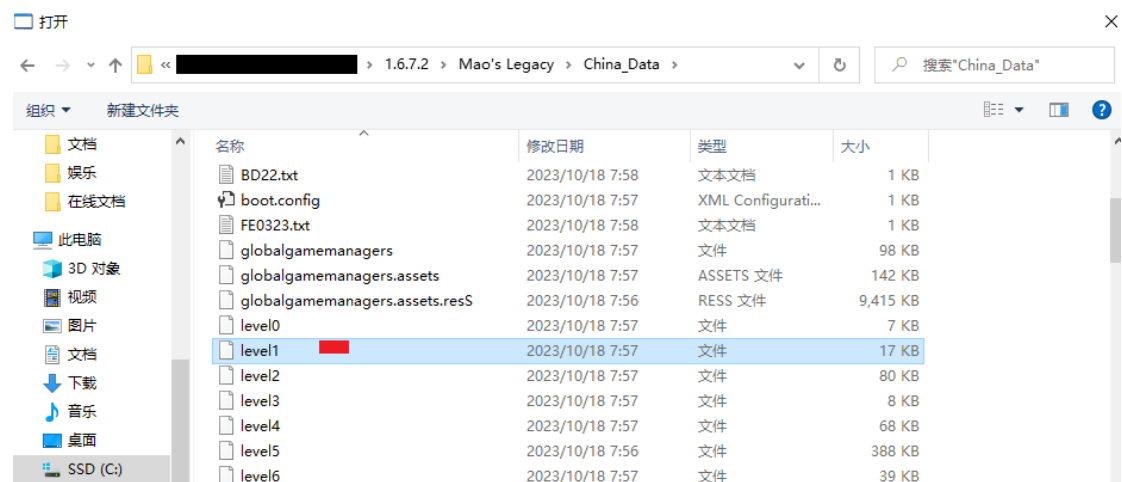
2.4 使用 AssetBundleExtractor 导出需要汉化的文件

首先，关闭之前所使用的 AssetStudioGUI。

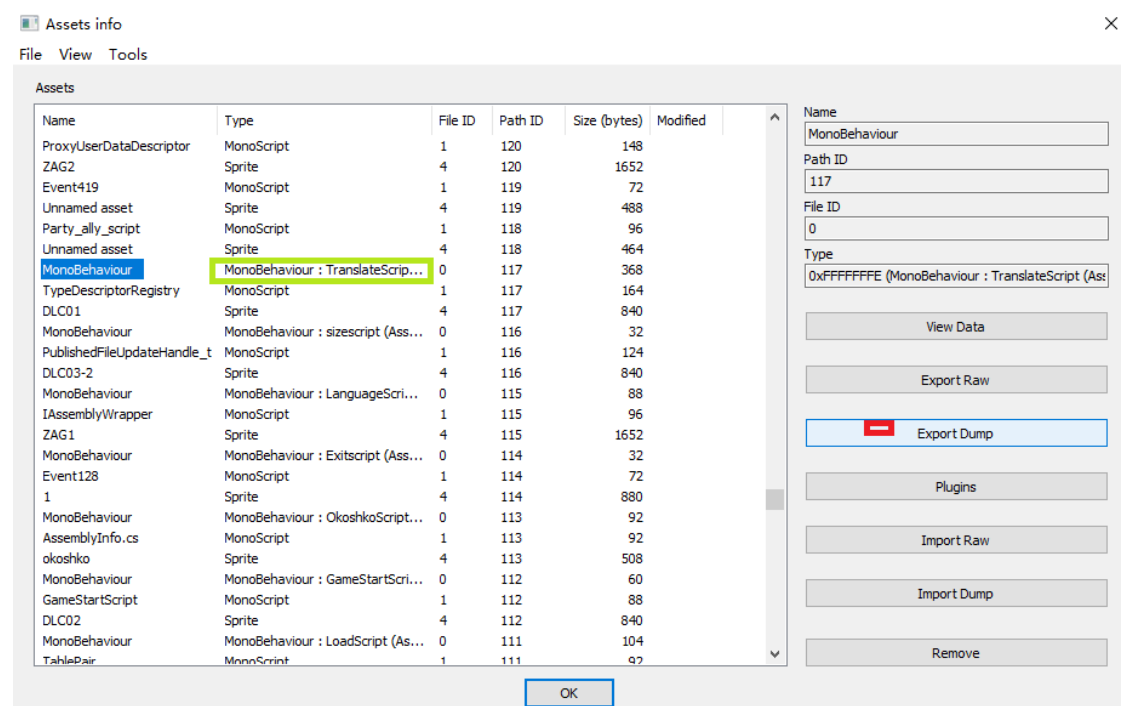


使用 AssetBundleExtractor 打开 Mao's Legacy\China_Data\level1。





回顾之前所记录的信息，level1 中的 TranslateScript，Path ID-117；
使用 Path ID 进行排序，找到 117，并选择其中名为 TranslateScript 的那一个，之后在右侧选择 Export Dump（导出并转储）。



Dump export



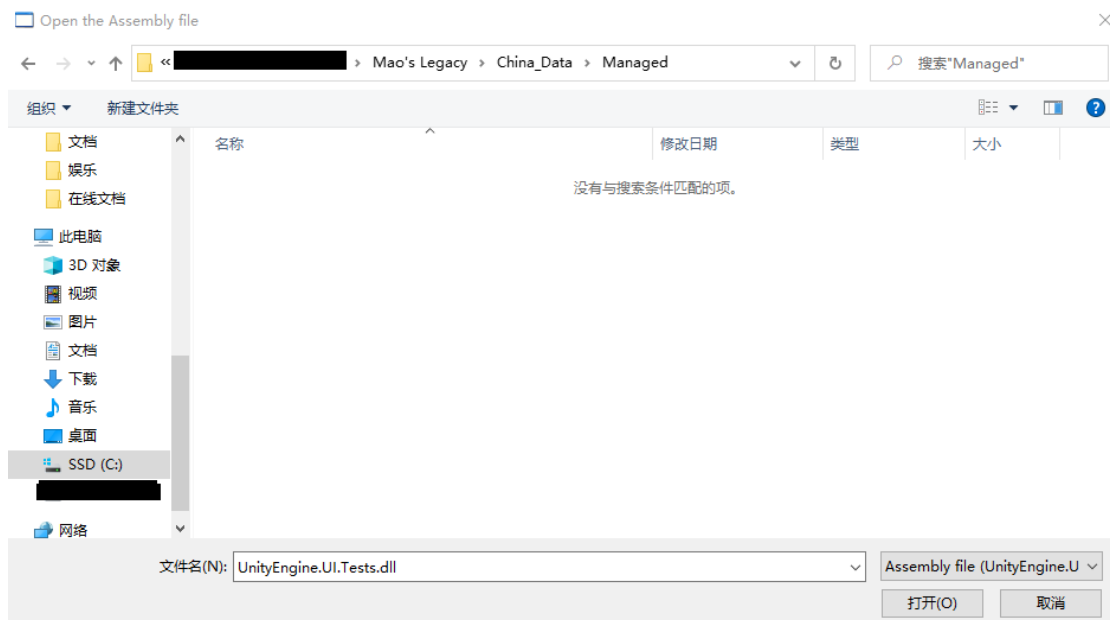
Additional MonoBehaviour type information can possibly be retrieved in order to enhance dump output.

Do you want to do that now? Alternatively, this can be done through Edit->Get script information.

是(Y)

否(N)

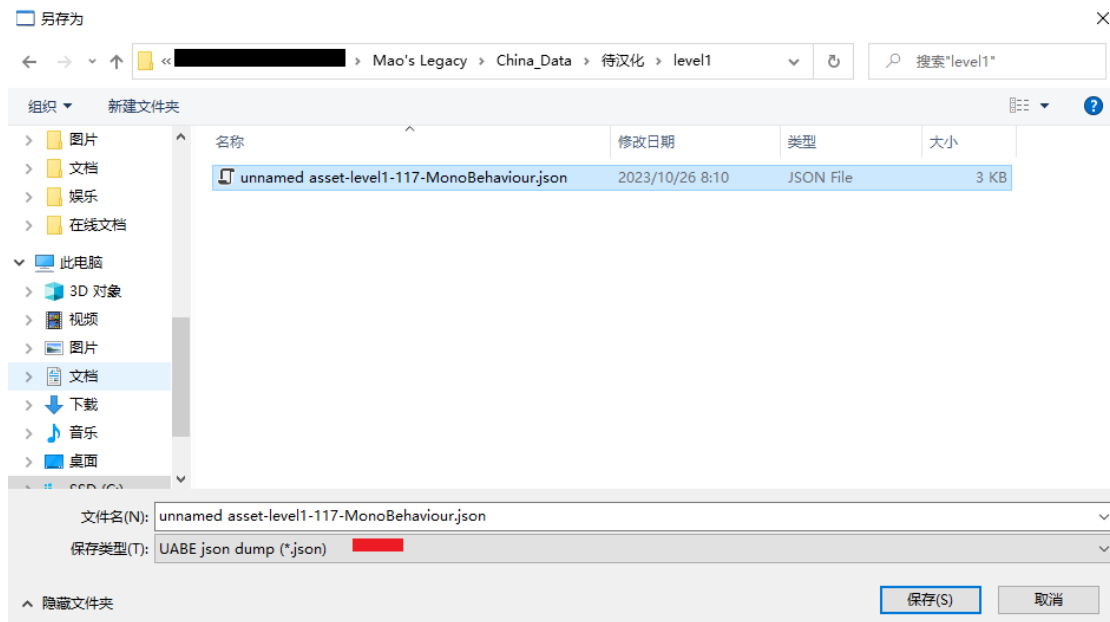
仍然是选择 Mao's Legacy\China_Data 下的 Managed 文件夹



当然，如果没有找到相应 dll 的话，就点击“取消”。

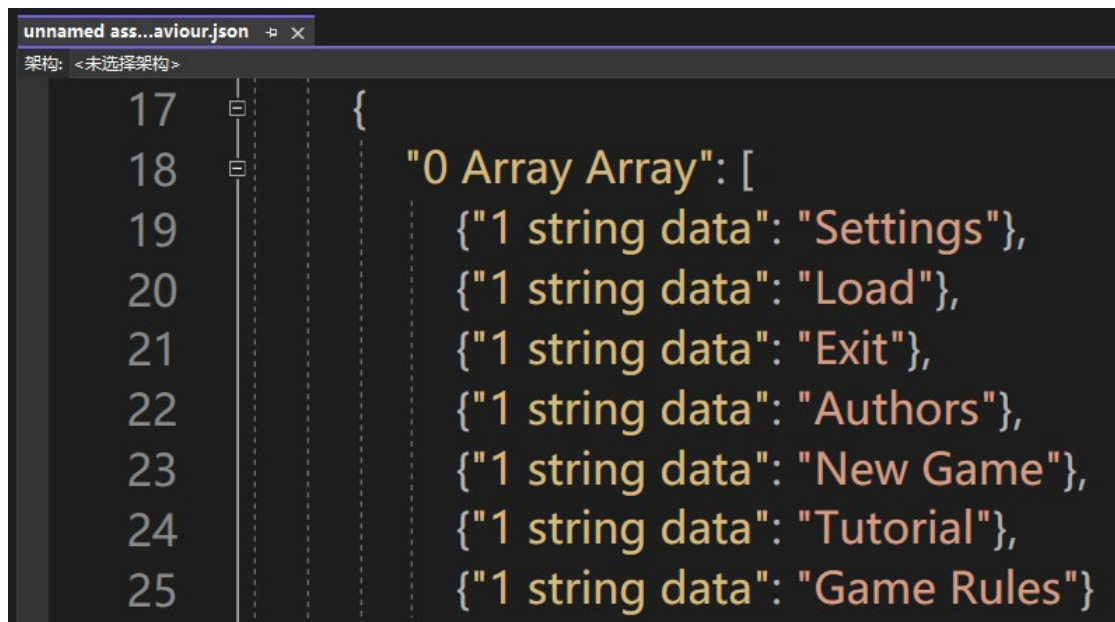
取消多次之后会跳出一个另存为选项，将其保存在 Mao's Legacy\China_Data\待汉化\level1 文件夹下(如果没有请自行创建)。

手动选择保存类型为 UABE json dump，并点击保存。



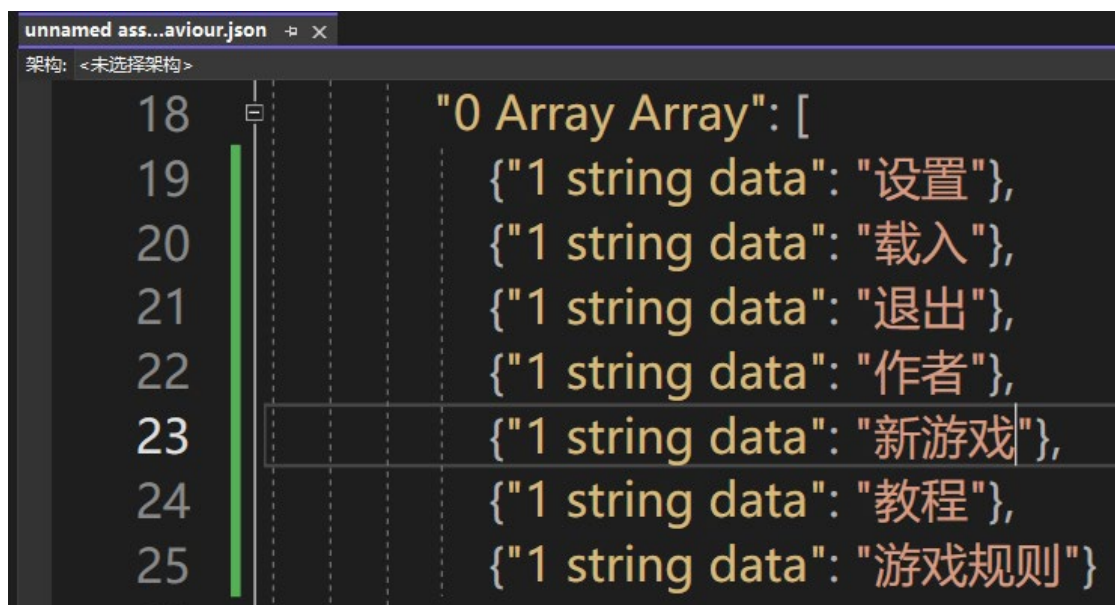
2.5 汉化相应的 json 文件

打开 Mao's Legacy\China_Data\待汉化\level1 下的 unnamed asset-level1-117-MonoBehaviour.json，汉化并保存。



The screenshot shows a code editor window titled 'unnamed ass...aviour.json'. The file content is as follows:

```
17 {  
18     "0 Array Array": [  
19         {"1 string data": "Settings"},  
20         {"1 string data": "Load"},  
21         {"1 string data": "Exit"},  
22         {"1 string data": "Authors"},  
23         {"1 string data": "New Game"},  
24         {"1 string data": "Tutorial"},  
25         {"1 string data": "Game Rules"}  
19
```

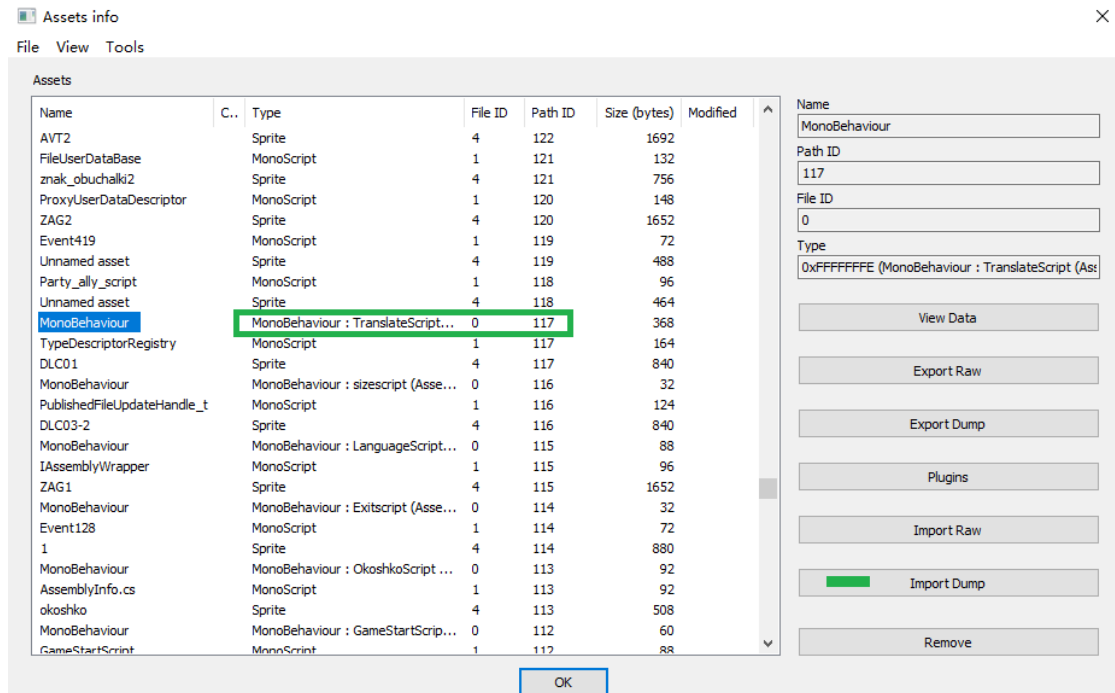


The screenshot shows the same code editor window after translation. The file content is now:

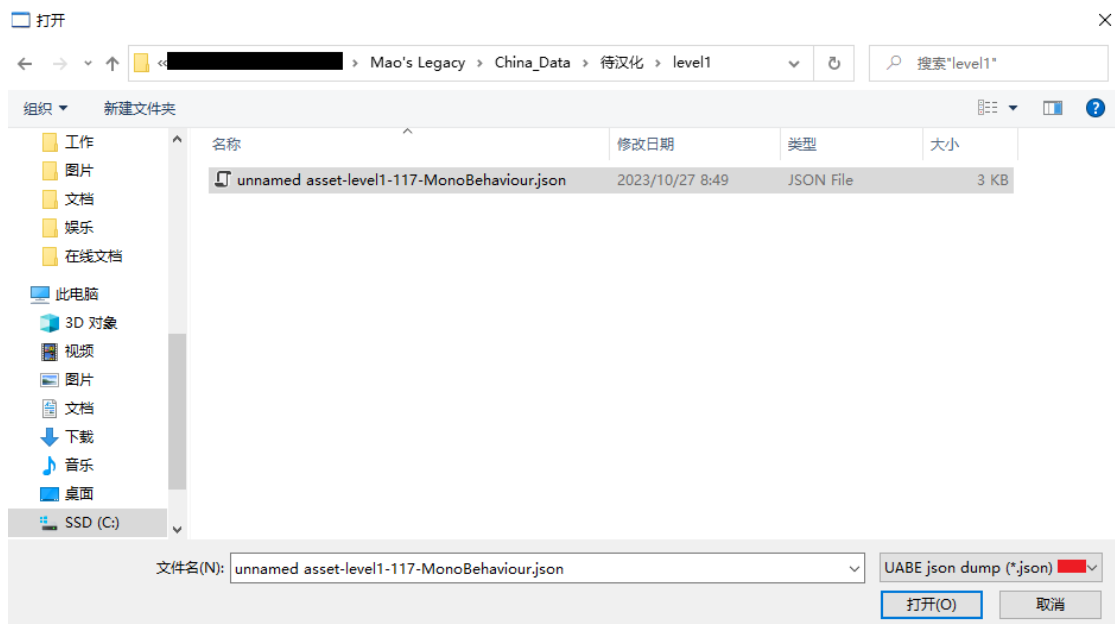
```
18     "0 Array Array": [  
19         {"1 string data": "设置"},  
20         {"1 string data": "载入"},  
21         {"1 string data": "退出"},  
22         {"1 string data": "作者"},  
23         {"1 string data": "新游戏"},  
24         {"1 string data": "教程"},  
25         {"1 string data": "游戏规则"}  
19
```

2.6 使用 AssetBundleExtractor 导回已汉化的 json 文件

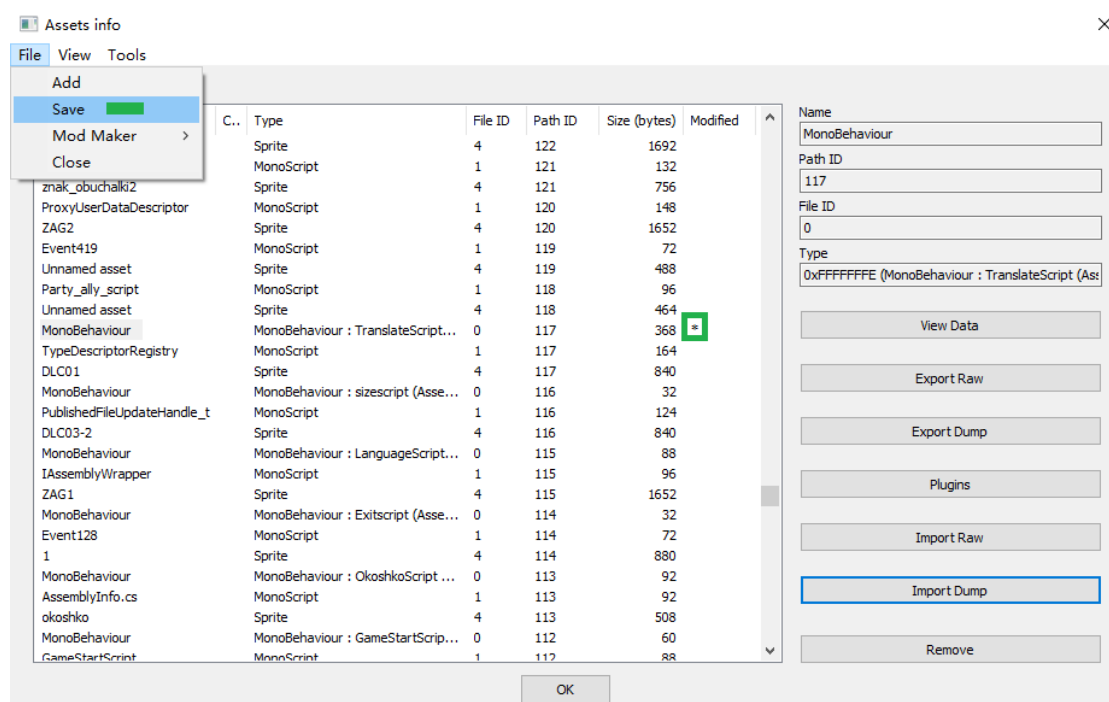
使用 AssetBundleExtractor 打开 Mao's Legacy\China_Data\level1 并找到之前所导出的 TranslateScript, PathID-117, 选择 Import Dump。



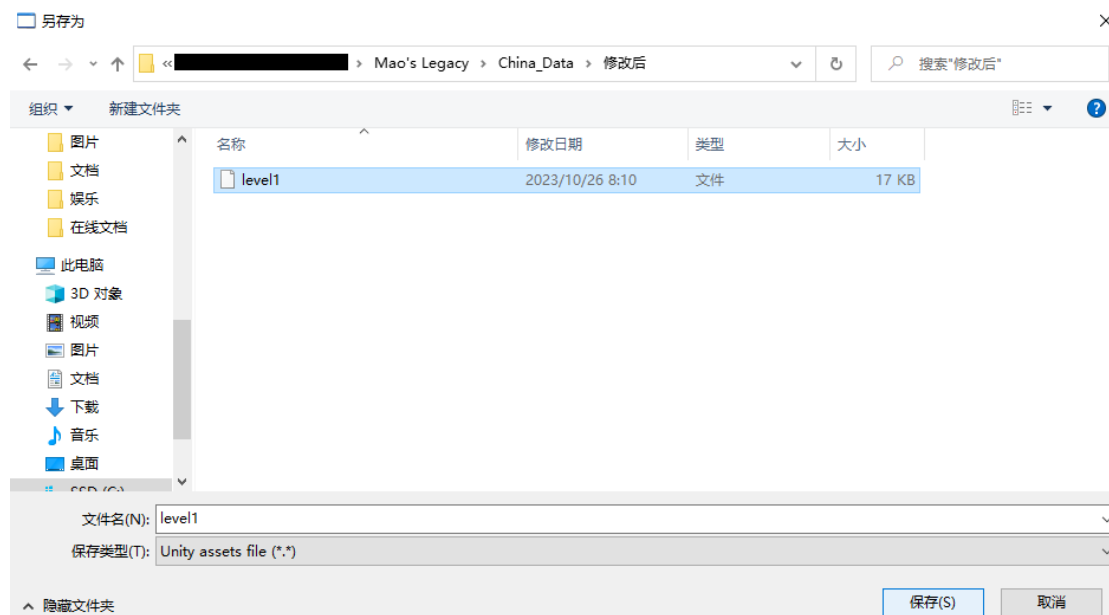
选择 Mao's Legacy\China_Data\待汉化\level1 下的 unnamed asset-level1-117-MonoBehaviour.json



可以看到，修改后的文件在 Modified 那一列出现了一个*号。



在 AssetBundleExtractor 中将修改后的 level1 文件保存到 Mao's Legacy\China_Data\修改后。



2.7 替换 Mao's Legacy\China_Data 下的文件

关闭 AssetBundleExtractor。

将 Mao's Legacy\China_Data\修改后的 level1 复制到 Mao's Legacy\China_Data，并替换目标中的文件。



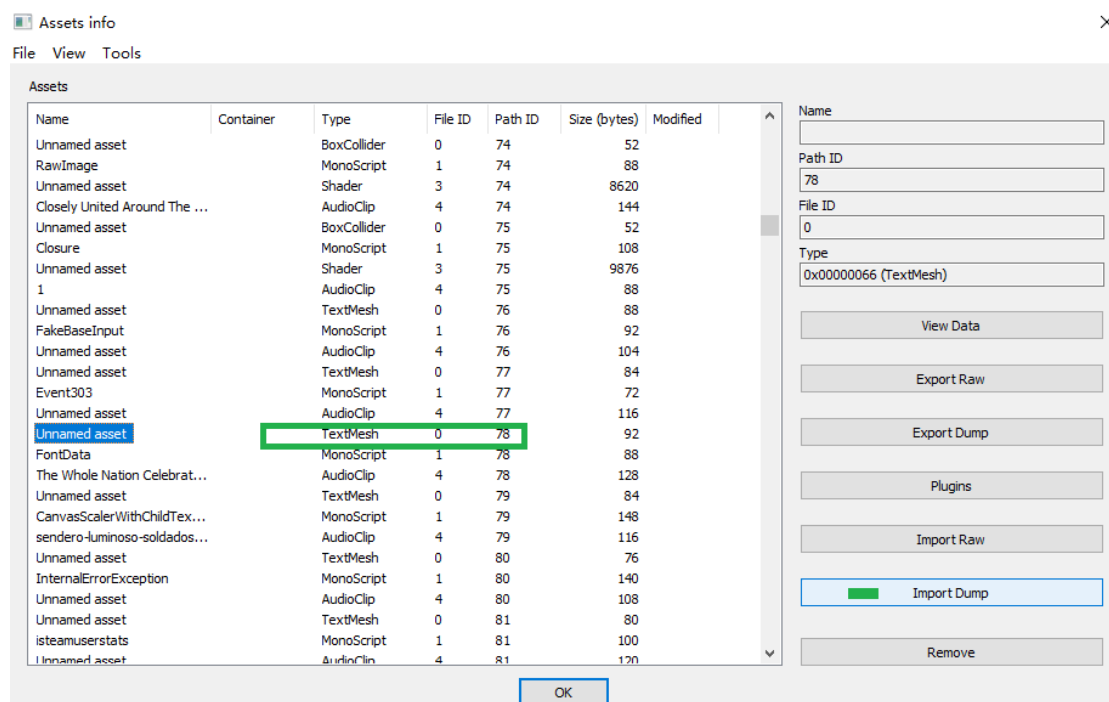
打开 Mao's Legacy 下的 China.exe，查看汉化效果。



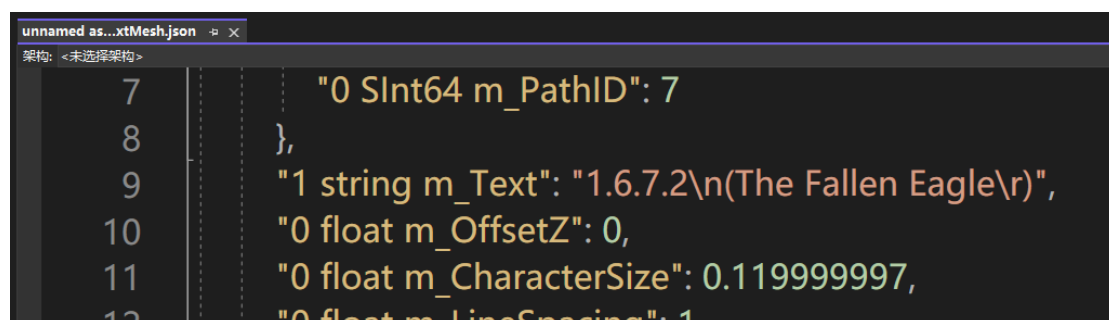
看起来还不错？ 很好的样子！

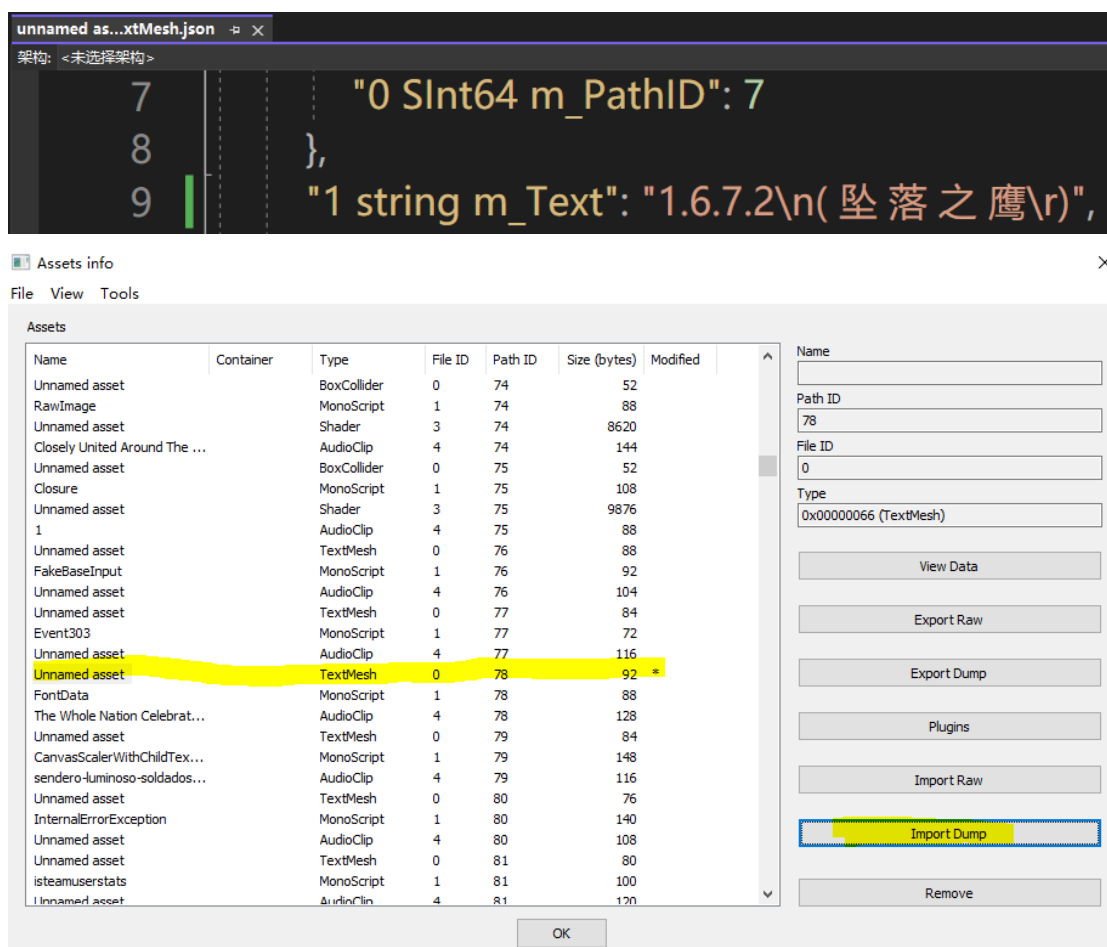
接着，我们注意到左上角，The Fallen Eagle 仍然是英文。那么，该去哪儿找到它呢？

经过一番寻找，我使用 AssetBundleExtractor 在 level1 中的 PathID-78 找到了它。



和之前的操作一样，导出转储为 json，汉化后再导回去。





接着是再次保存 level1 到“修改后”文件夹，关闭 AssetBundleExtractor，再将修改后的 level1 与 Mao's Legacy\China_Data 下的 level1 替换。



再次打开 Mao's Legacy 下的 China.exe。



左上角看起来有点奇怪，是吧？两侧分别多了个 N 和 R。



```
9 | "1 string m_Text": "1.6.7.2\n(The Fallen Eagle\r)",
```



原文是“The Fallen Eagle”，在原始英文版本中显示“THE FALLEN EAGLE”；显而易见，它使用了 string 类的 ToUpper()方法。

```
//测试场

// \n和\r为转义序列，表示换行、回车
string m_Text = "1.6.7.2\n(The Fallen Eagle)\r";
string upperCaseString = m_Text.ToUpper();
Console.WriteLine(upperCaseString); // 此方法将所有小写字母转为大写字母
```

Microsoft Visual Studio 调试控制台

```
1.6.7.2
(THE FALLEN EAGLE)
```

因为不明的原因，在使用中文字符替换英文后，ToUpper()方法将两侧的\n和\r转换成了N、R，但是这发生在转义序列奏效之后（不然“1.6.7.2”就会和“坠落之鹰”处于同一行了）。



如果它是一个C#脚本，我们可以将它转写为如下形式，以实现相同的效果，并规避一些问题。

```
//测试场

// \n和\r为转义序列，表示换行、回车
string m_Text = "1.6.7.2\n(The Fallen Eagle)\r"; //英文原文
string m_Text2 = "1.6.7.2" + $"{Environment.NewLine}" + "(坠落之鹰)";
string upperCaseString = m_Text2.ToUpper();
Console.WriteLine(upperCaseString); // 此方法将所有小写字母转为大写字母
```

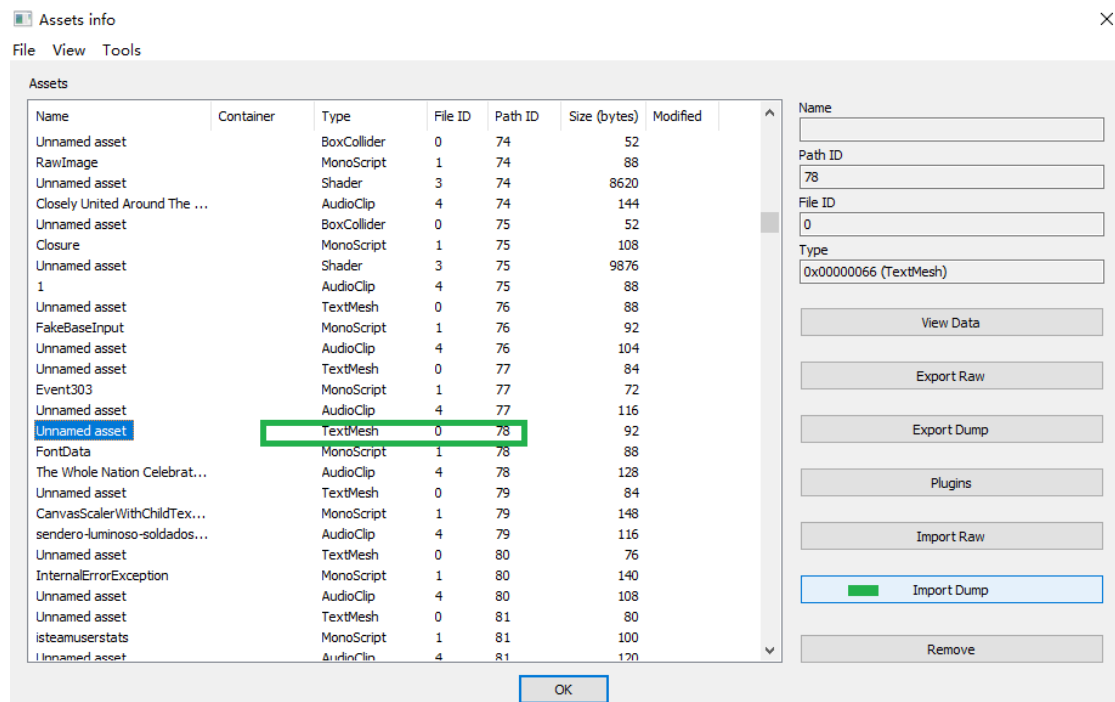
Microsoft \

```
1.6.7.2
(坠落之鹰)
```

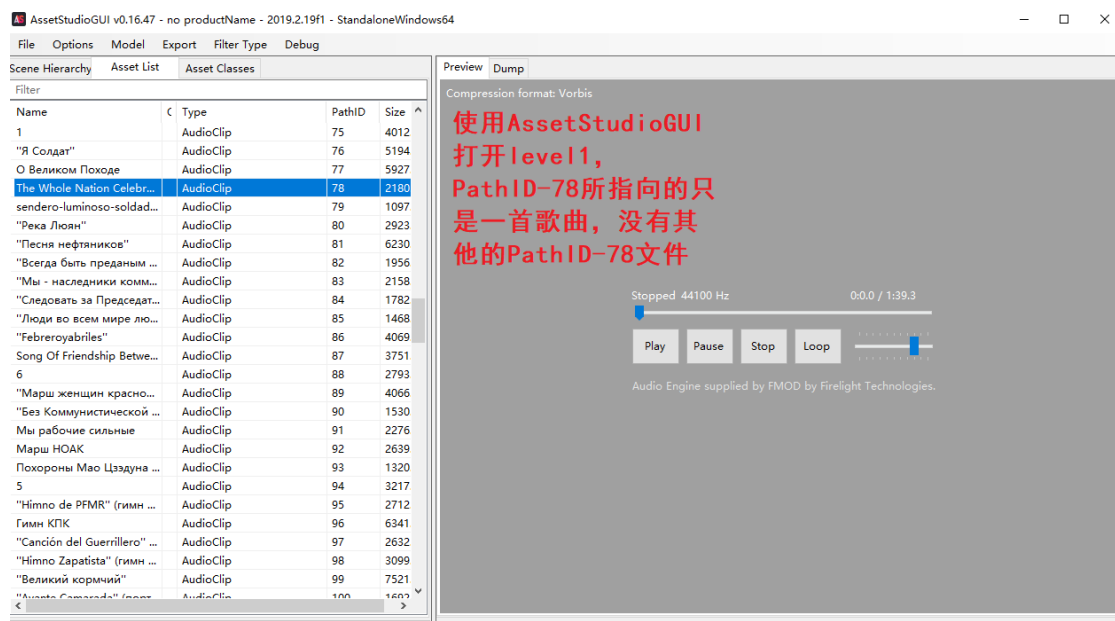
可惜这是一份导出转储的json文件，这个问题只能暂且搁置。

2.8 经由 2.7 末尾的 PathID-78 文件指明在汉化过程中使用 AssetStudioGUI 的一些问题

在 2.7 的第二部分，我使用 AssetBundleExtractor 找到了 level1 中的 PathID-78 文件。



但是, 使用 AssetStudioGUI 打开 level1, 我是找不到这个文件的。



关于此点的解决方案：汉化过程中，使用 AssetBundleExtractor 逐个打开 level 文件，将其中所有的 TextMesh、MonoBehaviour 文件全部提取，记录信息，逐个检查并汉化。

（这样做将极大地增加时间成本。）

3. 后记

Dorin 在近一周内的汉化过程中遭受到了极其强烈的精神折磨。

在翻阅了简中网络大量关于 Unity 汉化的资料后，他发现居然是 AssetBundleExtractor 版本的问题！需要从 v3.0-beta 退回到 v2.2-stabled 。

这太荒谬了！有鉴于此，编者建议大家要好好填写汉化日志或文档，以免后人再次掉坑。

编者：Dorin、KKND

游戏版本：1.6.7.2

时间：2023/10/27