序言

此文档将较简略地叙述关于《墙国:猫的遗产》汉化工作中提取相关文件 dll 及修改错误的过程。

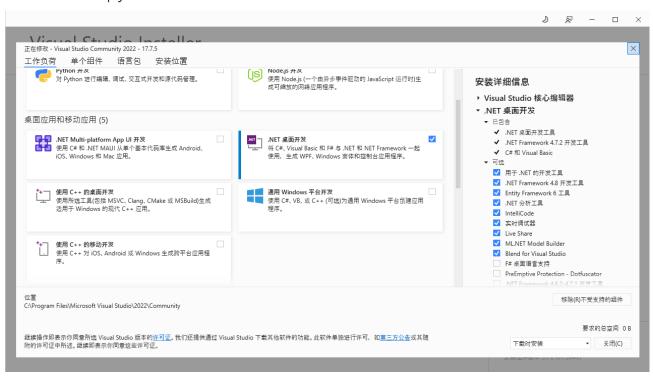
编者: Dorin

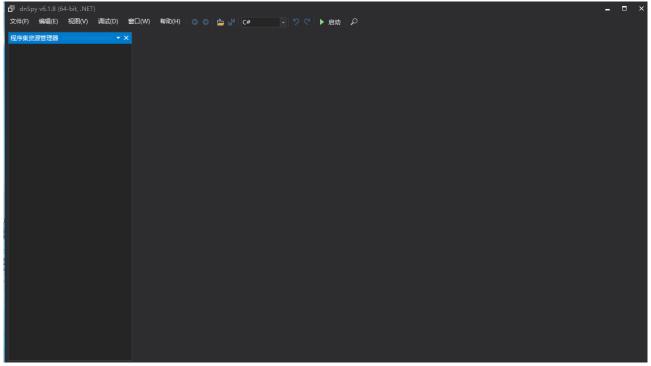
游戏版本: 1.6.8 (之前版本亦适用)

1. 所需工具

Visual studio 2022

dnSpy v6.1.8

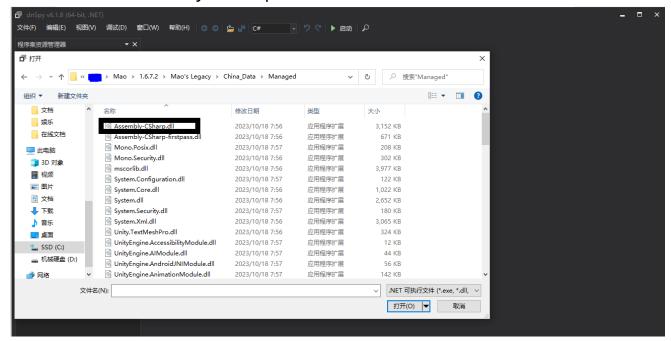




2. 提取流程

2.1 使用 dnSpy 打开 Mao's Legacy\China_Data\Managed 文件夹

下的 Assembly-CSharp.dll

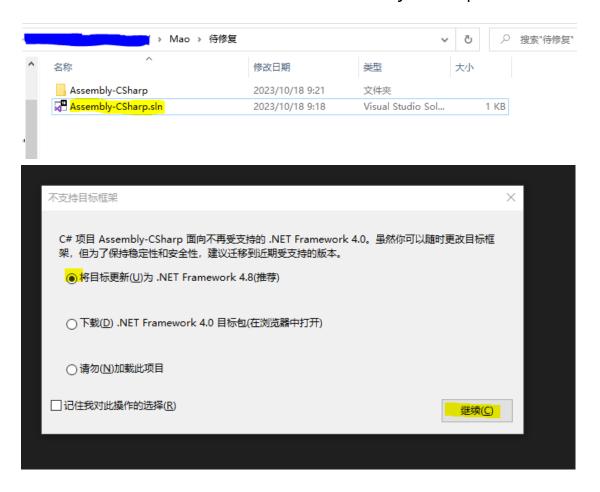


2.2 使用 dnSpy 将打开后的 Assembly-CSharp.dll 导出到工程

```
文件(F) 编辑(E)
                   视图(V)
                             调试(D)
                                       窗口(W)
                                                 帮助(H) 🕒 😊 造 📲 C# 🔻 💆 🤈 🤈
                                                                                                     ▶ 启动
   导出到工程(J)..
                                           ssembly-CSharp (0.0.0.0)
                                                     // C:\LJW\Mao\1.6.7.2\Mao's Legacy\China_Data\Managed\Assembly-CSharp.dll
// Assembly-CSharp, Version=0.0.0.0, Culture=neutral, PublicKeyToken=null
     保存模块(M)...
   打开(O)...
                                  Ctrl+O
                                                    using System;
using System.Diagnostics;
   从GAC打开(G)...
                            Ctrl+Shift+O
                                                    using System. Reflection;
using System. Runtime. Compiler Services
     打开列表(I)...
     最近的文件(F)
                                                     [assembly: AssemblyVersion("0.0.0.0")]
     重新加载所有程序集(R)
☑ 全部关闭
                                                     [assembly: \ Debuggable (Debuggable Attribute. Debugging Modes. \ Ignore Symbol Store Sequence Points)] \\
図 退出(X)
                                  Alt+F4
```

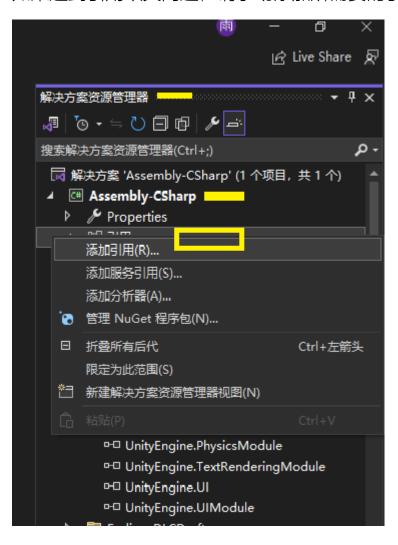
导出到工程		x			
文件夹(F)	`'\Mao\待修复				
解决方案(S)	Assembly-CSharp.sln	✓			
版本(V)	VS2019				
语言(L)	C#				
项目GUID (P)	98aa098d-dbb0-4731-bc3e-743cbb7c52	ea 新建(N)			
✓ 解包资源(U)				
■ 创建 .resx文件(X)					
✓ 将BAML反编译为XAML (B)					
☐ 在Visual Studio中打开项目(O)					
即将反编译一个文件。若要反编译多个文件,请选中多个程序集。					
	导出(E)	取消(C)			

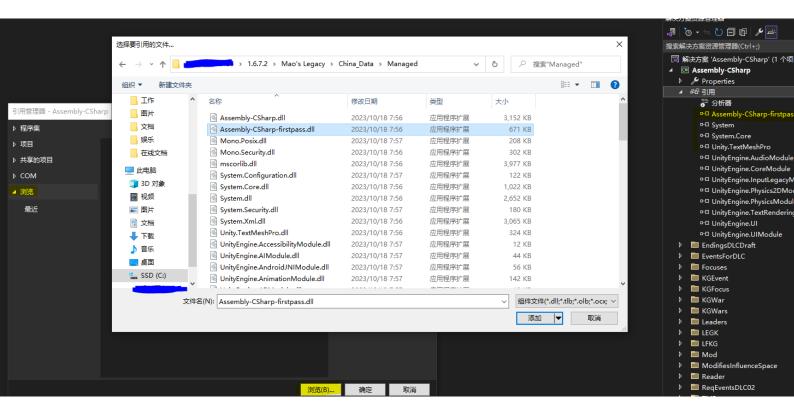
2.3 打开"待修复"文件夹下的 Assembly-CSharp.sln



2.4 添加相关引用(可能非必需)

如果遇到引用缺失问题,请手动添加所需要的引用



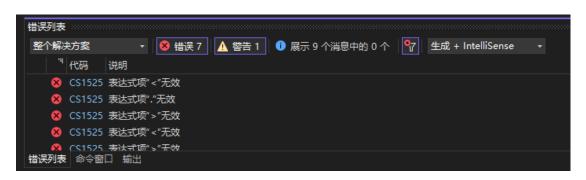


引用的所在位置,在 Mao's Legacy\China_Data\Managed 下 右侧解决方案管理器中,缺什么引用,就手动添加什么引用。 添加引用时请注意游戏版本。

2.5 重新生成解决方案



接着便浮现出错误。



3. 修改错误流程

修改错误的顺序可能各有不同,编者会列出错误所出现的文件及相应位置。

在每一步修错结束后,若 IDE 没有及时更新错误列表,请在 *菜* 单栏-生成 重新生成解决方案。

3.1 错误 1: Comparison<int> <>9_0 相关

出现位置:

VertexZoom.cs 的第 53 行、113 行、115 行

错误原因: 反编译过程中丢失了原变量名, 且原变量名未赋初始值。

修错方法:将变量<>9_0改为 comparisonX 并赋予初始值。

```
VertexZoom.cs + ×
Assembly-CSharp
                                          → <sup>©</sup>

TMPro.Examples.VertexZoom
         52
                          this.hasTextChanged = true;
                          Comparison<int> comparisonX=null;
         53
                          for (;;)
         54
         55
VertexZoom.cs ⇒ ×
Assembly-CSharp
                            - %TMPro.Examples.VertexZoom
                           Comparison < int > comparison;
     112
     113
                           if ((comparison = comparisonX) == null)
     114
     115
                              comparison = (comparisonX = ((int a, int b) =>
     116
```

3.2 错误 2: <PrivateImplementationDetails>相关

出现位置:

EventReader.cs 的第 112 行

```
EventReader.cs * x

Assembly-CSharp

- **LEGK.EventReader

- **CreateEvents(List<Lexem> list_lexems)

- **Creat
```

错误原因: 反编译过程中丢失了相应文件的链接。

修错方法: 前往底层将相应的 ComputeStringHash 方法找回。

具体的寻找在此不予细讲,我最初是在 1.6.5 的原生英文版本中寻得,在之后的游戏版本里,此方法因不明示于文件表面,故屡次造成了这个问题。

```
internal static uint ComputeStringHash(string s)
{
    uint num = 0U;
    bool flag = s != null;
    if (flag)
    {
        num = 0x811C9DC5U;
        for (int i = 0; i < s.Length; i++)
        {
            num = ((uint)s[i] ^ num) * 0x1000193U;
        }
        }
        return num;
}</pre>
```

//复制粘贴过程中可能会发生错误

```
internal static uint ComputeStringHash(string s)
{
    uint num = 0U;
    bool flag = s != null;
    if (flag)
    {
         num = 0x811C9DC5U;
         for (int i = 0; i < s.Length; i++)
         {
             num = ((uint)s[i] ^ num) * 0x1000193U;
         }
    }
    return num;
}
```

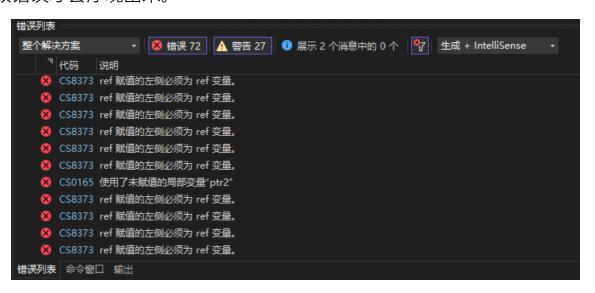
现将此方法提取上来,放置于 EventReader.cs 文件的 internal static class EventReader 类下,并在 112 行修改相应的语句。

```
→ %LEGK.EventReader
           // Token: 0x0600065E RID: 1630 RVA: 0x0014E6DC File Offset: 0x0014C8DC
247
           internal static uint ComputeStringHash(string s)
248
249
250
             uint num = 0U;
251°
             bool flag = s != null;
                                                         放置于EventReader.cs文件的internal
252
             if (flag)
                                                         static class EventReader类下
253
                                                         //编者选择在EventReader类的末尾放置
                num = 0x811C9DC5U;
254
                for (int i = 0; i < s.Length; i++)
255
256
                  num = ((uint)s[i] ^ num) * 0x1000193U;
257
258
259
260
             return num;
261
```

修改后重新生成解决方案

```
编辑(E) 视图(V) Git(G) 项目(P) 生成(B) 调试(D) 测试(S) 分析(N) 工具(T) 扩展(X) 窗口(W) 帮助(H) 🗸 搜索 🕶
    御・昏 🗎 🗐 り・マ - Debug ៉ 生成解決方案(B)
                                                    Ctrl+Shift+B 👝 🕼 🖫 🧏 🔲 刽 刃 闪 束
                              重新生成解决方案(R)
EventReader.cs* ⇒ ×
                              清理解决方案(C)
# Assembly-CSharp
                              对解决方案运行代码分析(Y)
                                                             Crading W
                           益 生成 Assembly-CSharp (U)
       108
                             重新生成 Assembly-CSharp (E)
                             清理 Assembly-CSharp(N)
       109
                       wl
                             对 Assembly-CSharp 运行代码分析(A)
                              批生成(T)...
       110
                             配置管理器(O)...
       111
                           string lexem type = array[i].lexem type;
                           uint num = ComputeStringHash(lexem_type);
       112
                           if (num <= 2369371622U)
```

可以看见,IDE 又报出了许多错误。这并非是之前的改错所造成的,而是因为 IDE 会优先显示较为明显的错误;在主要错误修改后,次级错误才会浮现出来。



3.3 错误 3: ptr 相关一系列错误

出现位置:

Results text.cs 中的较多行

```
Results_text.cs + ×
ः Assembly-CSharp
                                                                          + SaAzkaban()
      270
                              i = num5;
     271
                           int ptr2;
     272
                           if (GlobalScript.inst.gameState.party_number[1] > 1
     273
     274
                              text = "We won with a devastating result, taking
     275
                              ptr2 = ref GlobalScript.inst.gameState.data[3];
     276
                              ptr2 += 10;
     277
                              ptr2 = ref GlobalScript.inst.gameState.data[4];
     278
     279
                              ptr2 -= 20;
                              ptr2 = ref GlobalScript.inst.gameState.data[1];
     280
                              ptr2 += 50;
     281
     282<sup>®</sup>
```

语句解释: 所谓 ptr, 是 pointer 的缩写, 也就是"指针"。 在 C/C++相关编程语言中, 这段语句首先创建了一个空指针, 然后将空指针指向相应数据, 根据判断条件来使用指针修改数据。

错误原因:在.NET6/7/8 (C#标准)中,您不能直接声明一个空引用 (null reference)或空指针,然后将其用作 ref 参数。

修错方法: 将相关 ptr 在定义时首先赋值以一个距离最近、与 ptr 对应的 ref 值。(注意定义时的 ref int, 此修改并不会改变游戏体验)

```
📆 文件(F) 编辑(E) 视图(V) Git(G) 项目(P) 生成(B) 调试(D) 测试(S) 分析(N) 工具(T) 扩展(X) 窗口(W) 帮助(H) 🗸 搜索 🕶 Assembly-CSharp
     御・営 🗒 📵 り・♡ - Debug - Any CPU
                                 ▼ ▶ 启动 ▼ ▶ ∅ ▼ 📭 局 📲 🤝 ե 僱 🖫 🧏 闪 気 気 気 臭
 Assembly-CSharp
                                        + 🕸 Results_text
                                                                               → 🗞 Azkaban()
       270
                                 j = num5;
       271
                              ref int ptr2 = ref GlobalScript.inst.gameState.data[3];
       272<sup>®</sup>
                              if (GlobalScript.inst.gameState.party_number[1] > 1500)
       273
       274
       275
                                 text = "We won with a devastating result, taking the ma
                                 ptr2 = ref GlobalScript.inst.gameState.data[3];
       276
       277
                                 ptr2 += 10;
                                 ptr2 = ref GlobalScript.inst.gameState.data[4];
       278
       279
                                 ptr2 -= 20;
                                 ptr2 = ref GlobalScript.inst.gameState.data[1];
       280
                                 ptr2 += 50;
       281
       282
```

其余 ptr 问题解决方法类似,注意在修改时相应变量有 ptr、ptr2、ptr3 之分,切勿对应错误。

修改完后重新生成解决方案

```
Results_text.cs +> X
                      清理解决方案(C)
对解决方案运行代码分析(Y)
 Assembly-CSharp
     9584
                      重新生成 Assembly-CSharp (E)
                      清理 Assembly-CSharp(N)
     9585<sup>®</sup>
     9586
                                            alScript.inst.gameState.data[3];
                       if (GlobalScript.inst.gameState.party number[1] > 1500)
     9587
     9588
                         text = "Мы победили с разгромным результатом, за
     9589
     9590
                         ptr2 = ref GlobalScript.inst.gameState.data[3];
     9591
                         ptr2 += 10;
                         ptr2 = ref GlobalScript.inst.gameState.data[4];
     9592
     9593
                         ptr2 -= 20;
     9594
                         ptr2 = ref GlobalScript.inst.gameState.data[1];
     9595
                         ptr2 += 50;
     9596
```

3.4 错误 4: 只有 assignment、call、increment、decrement 和 new 对象表达式可用作语句(常规问题)

出现位置:

TMP_TextSelecter_A.cs 中的 56 行

错误原因: 赋值语句写成判断语句。

修错方法: "=="改为"="。

3.5 错误 5: Lambda 相关问题(1.6.7.2 版本)

出现位置:

doneventscript.cs 中的原 6087 行等处

```
doneventscript.s * x

Assembly-CSharp

6084

6085

if (GlobalScript.inst.gameState.faction_lead)
6086

6087

GlobalScript.inst.gameState.faction_lead
6088

6089

"pol",
6090

"index"

6091

6092

} ValueTuple < Politic, int > pol) = > pol.l
6092
```

错误原因:游戏制作组所编写的 C#版本过于古早 (?)。

修错方法: 元组在.NET6/7/8 的 Lambda 表达式中有更好的解决方法。

原 6087 行附近

原代码:

GlobalScript.inst.gameState.faction_leader[4] =
GlobalScript.inst.gameState.politics.Select((Politic pol, int i) =>
new ValueTuple<Politic, int>(pol,
i)).FirstOrDefault(([TupleElementNames(new string[]

```
ntscript.cs* 💠 🤉
bly-CSharp
         136 II (Olobalocriptiliistigamestateilialiibel evelit – – 57)
6084
6085
        if (GlobalScript.inst.gameState.faction_leader[4] >= 200 || GlobalScript.inst.gameState.faction_lea
6086
6087
             GlobalScript.inst.gameState.faction leader[4] =
               GlobalScript.inst.gameState.politics.Select((Politic pol, int i)
6088
               => new ValueTuple < Politic, int > (pol, i)). FirstOrDefault(([TupleElementNames(new string[]
6089
6090
               "pol",
6091
               "index"
6092
             })] ValueTuple < Politic, int > pol) => pol.Item1.traits[0] == 3).Item2;
6093
```

修改代码:

```
GlobalScript.inst.gameState.faction_leader[4] =
GlobalScript.inst.gameState.politics.Select((Politic pol, int i) =>
new ValueTuple<Politic, int>(pol, i))
.FirstOrDefault(polTuple => polTuple.ltem1.traits[0] == 3).ltem2;
```

```
原 6093 行附近
原代码:
GlobalScript.inst.gameState.faction leader[4] =
GlobalScript.inst.gameState.politics.Select((Politic pol, int i) =>
new ValueTuple < Politic, int > (pol,
i)).FirstOrDefault(([TupleElementNames(new string[]
{
             "pol",
             "index"
})] ValueTuple < Politic, int > pol) => pol.Item1.traits[0] == 2).Item2;
修改代码:
             GlobalScript.inst.gameState.faction_leader[4] =
GlobalScript.inst.gameState.politics.Select((Politic pol, int i) =>
new ValueTuple < Politic, int > (pol, i))
.FirstOrDefault(polTuple => polTuple.ltem1.traits[0] == 2).ltem2;
```

```
原 6858 行附近
原代码:
GlobalScript.inst.gameState.faction leader[4] =
GlobalScript.inst.gameState.politics.Select((Politic pol, int i) =>
new ValueTuple < Politic, int > (pol,
i)).FirstOrDefault(([TupleElementNames(new string[]
{
             "pol",
             "index"
})] ValueTuple<Politic, int> pol) => pol.Item1.traits[0] == 3).Item2;
修改代码:
          GlobalScript.inst.gameState.faction_leader[4] =
GlobalScript.inst.gameState.politics.Select((Politic pol, int i) =>
new ValueTuple < Politic, int > (pol, i))
.FirstOrDefault(polTuple => polTuple.ltem1.traits[0] == 3).ltem2;
```

```
原 6866 行附近
原代码:
GlobalScript.inst.gameState.faction leader[4] =
GlobalScript.inst.gameState.politics.Select((Politic pol, int i) =>
new ValueTuple < Politic, int > (pol,
i)).FirstOrDefault(([TupleElementNames(new string[]
{
             "pol",
             "index"
})] ValueTuple < Politic, int > pol) => pol.Item1.traits[0] == 2).Item2;
修改代码:
          GlobalScript.inst.gameState.faction_leader[4] =
GlobalScript.inst.gameState.politics.Select((Politic pol, int i) =>
new ValueTuple < Politic, int > (pol, i))
.FirstOrDefault(polTuple => polTuple.ltem1.traits[0] == 2).ltem2;
```

3.7 重新生成解决方案

在以上错误都修复完毕后,只剩下一些 warning (警告),请忽视 它们,并在 *菜单栏-生成* 重新生成解决方案。



```
351
                                                                                       GlobalScript.inst.gameState.faction leader[4] = GlobalScript.ins
352
                                               .FirstOrDefault(polTuple => polTuple.ltem1.traits[0] == 3).ltem2;
 354
 355
                                                                         if (GlobalScript.inst.gameState.faction leader[4] >= 200 || GlobalS
 356
357
                                                                                      GlobalScript.inst.gameState.faction leader[4] = GlobalScript.ins
                                               .FirstOrDefault(polTuple => polTuple.ltem1.traits[0] == 2).ltem2;
 358
859®
                                                                         text2 = "Тяньаньмэньский инцидент. Снова?!";
 360
                                                                                                                             = ≥ | ≥ | ©
 | 一名 | 日本の | 日本の
           p - Mao\特修复\Assembly-CSharp\bin\Debug\Assembly-CSharp.dll
重新生成": 1 成功, 0 失败, 0已跳过 ————
; 开始于 12:39,并花费了 01.430 秒 —————
```

以上便重新生成了 Assembly-CSharp.dll, 可以在相应项目的 bin/Debug 文件夹下找到它。

称^	修改日期	类型	大小
Assembly-CSharp.dll	2023/10/18 12:39	应用程序扩展	3,401 KB
Assembly-CSharp.pdb	2023/10/18 12:39	Program Debug	2,572 KB
Assembly-CSharp-firstpass.dll	2023/10/18 7:56	应用程序扩展	671 KB
System.dll	2023/10/18 7:56	应用程序扩展	2,652 KB
Unity.TextMeshPro.dll	2023/10/18 7:56	应用程序扩展	324 KB
UnityEngine.AnimationModule.dll	2023/10/18 7:57	应用程序扩展	142 KB
UnityEngine.AudioModule.dll	2023/10/18 7:57	应用程序扩展	54 KB
UnityEngine.CoreModule.dll	2023/10/18 7:56	应用程序扩展	892 KB
UnityEngine.IMGUIModule.dll	2023/10/18 7:57	应用程序扩展	146 KB
UnityEngine.InputLegacyModule.dll	2023/10/18 7:57	应用程序扩展	25 KB
UnityEngine.Physics2DModule.dll	2023/10/18 7:57	应用程序扩展	104 KB
UnityEngine.PhysicsModule.dll	2023/10/18 7:57	应用程序扩展	86 KB
UnityEngine.SharedInternalsModule.dll	2023/10/18 7:57	应用程序扩展	19 KB
UnityEngine.TextCoreModule.dll	2023/10/18 7:57	应用程序扩展	176 KB
UnityEngine.TextRenderingModule.dll	2023/10/18 7:57	应用程序扩展	27 KB
UnityEngine.UI.dll	2023/10/18 7:57	应用程序扩展	219 KB
UnityEngine.UIModule.dll	2023/10/18 7:57	应用程序扩展	22 KB

4. 注意事项与后记

4.1 汉化的任务

主要任务是替换相应的英文文本。

根据以往的汉化经验,替换文本的过程中经常会出现小的问题,比

如缺失一侧的大括号,将英文分号写成中文的分号(;和;),一些

转义序列上的问题等等。

编者在此文档中专注于程序修错,这类汉化文本相关问题还请咨询

汉化组群友或者其他 Github 组员。

4.2 后记

编者认为,由于汉化组群友缺乏行之有效的沟通,以及在任务分配上

的管理欠缺, 故而经常性出现重复劳动的悲剧。

因此,编者建议,每位群友最好在汉化组群或者 GitHub 专门开辟一

个工作日志或者文档, 每完成一项任务, 根据日期和内容为分类标签

逐次填写,藉此以提高游戏汉化的整体效率。

编者: Dorin

游戏版本: 1.6.7.2

时间: 2023/10/18