

# HOI4DEV 钢4MOD开发工具

更新日期：2024.12.10

## 示例：PIHC2模组

GitHub链接（暂时为私密，需要联系作者申请访问；我们将很快开放）。

## 工程源文件结构

值得强调的是，HOI4DEV仅仅是一个Python工具包，其核心在于与Python的灵活配合。因此，一方面，构造一个自动编译的模组需要开发者具有一定Python编程能力。另一方面，灵活性也导致所展示的工程示例也并不能够代表HOI4DEV必须被这样使用。开发者可以根据自己的喜好组织工程源文件的格式。此处仅展示PIHC2的一个实践示例。

### PIHC2的工程源文件文件夹结构:

```

17 | | └─ achievements/          # 成就
18 | |   └─ ...
19 | | └─ bops/                # 权力平衡
20 | |   └─ ...
21 | | └─ characters/          # 角色
22 | |   └─ ...
23 | | └─ characters_random/    # 随机角色肖像
24 | |   └─ ...
25 | | └─ copies/              # 复制内容
26 | |   └─ ...
27 | | └─ ...
28 | | └─ superevents/         # 超事件
29 | |   └─ ...
30 | | └─ technologies/        # 科技
31 | |   └─ ...
32 | | └─ traits/              # 特质
33 | |   └─ ...
34 | | └─ buildings_delta.json  # 快速在个别地块上添加新的建筑
35 | | └─ delta_vic.txt         # 快速在个别地块上修改胜利点的名字
36 | | └─ relation_grid.json    # 快速调整国家间两两的初始外交关系
37 | |   └─ ...
38 | |
39 | └─ scripts/                # 一些临时用到的、方便的脚本
40 | | └─ adv_gen/              # 从Excel表格中自动生成顾问角色到characters/中
41 | |   └─ ...
42 | | └─ region_gui/          # 从地图中提取若干个省份构建gui（类似于TN0乌克兰的面包蓝地图）
43 | |   └─ ...
44 | | └─ states_editor/        # 地块数据编辑器
45 | |   └─ ...
46 | | └─ wiki/                # 自动列举mod中存在的所有意识形态和多语言本地化，上传到mod的wiki页面
47 | |   └─ ...
48 | | └─ C00_create_a_new_mod.py  # mod正式内容：第00步，创建一个新的mod
49 | | └─ C01_build_the_map.py    # mod正式内容：第01步，编辑地图（相邻关系、地块归属、建筑资源等）
50 | | └─ C02_add_mirror_portal.py  # mod正式内容：第02步，PIHC模组特色，在地图上添加传送门
51 | | └─ C03_add_gfxs.py        # mod正式内容：第03步，添加加载图
52 | | └─ C04_add_countries.py    # mod正式内容：第04步，添加所有国家
53 | | └─ ...
54 | | └─ C11_add_intel_agencies.py  # mod正式内容：第11步，添加所有情报机构
55 | | └─ C12_add_achievements.py  # mod正式内容：第12步，添加所有成就
56 | | └─ C13_add_inventory_items.py  # mod正式内容：第13步，PIHC模组特色，添加所有物品栏中的物品
57 | | └─ C14_add_state_lores.py    # mod正式内容：第14步，PIHC模组特色，添加所有地块的地块故事、介绍
58 | |   └─ ...

```

```

59 |
60 |─ steamdown/          # 一个第三方的steam工具，可以自动把模组信息转为steam页面
61 |   └─ ...
62 |─ README_generator.py # 自动节选模组信息，使用steamdown转换
63 |
64 |─ v0.2.2.py           # 当前模组版本v0.2.2.py，主脚本，调用mod正式内容脚本，一键生成完整的
   整个mod
65 |─ ...

```

在这样一个结构下，mod开发只需要在resources中添加JSON代码、本地化TXT或者png图标，然后反复运行 `python v0.2.2.py` 检查结果即可。

特别地，如果需要将自动开发与传统制作组的手动开发结合，在 `resources/` 中有 `copies/` 文件夹，其中的内容都可以使用传统的钢4语言、文件结构编写，会被原封不动复制到编译完的模组中。同时，也支持在 `resources/copies/data/` 中手动以HOI4DEV的语法进行编辑。如此，传统制作组的开发方式、手动使用HOI4DEV开发方式、自动使用HOI4DEV开发方式，可以有机结合起来。

## 示例1：添加角色

角色文件都在 `resources/characters/` 中，一个典型的角色文件夹格式为：

```

1 resources/
2 |   └─ characters/
3 |     └─ STARLIGHT_GLIMMER/      # 角色 CHARACTER_STARLIGHT_GLIMMER
4 |       └─ portraits/
5 |         └─ alt.png             # 替代图像，类似于EAW暮光理论中暮光肖像随着精神状态变换
6 |         └─ default.png        # 默认领袖肖像，无论大小格式都会被先缩放后裁剪至标准
           210x156的dds文件，同时自动生成顾问小肖像
7 |         └─ info.json           # 角色代码（领袖、顾问效果）
8 |         └─ locs.txt            # 角色本地化（名称、描述）
9 |         └─ ...
10 |      └─ ...
11 |   └─ ...

```

而角色特质在 `resources/traits/` 中：

```

1 resources/
2 |   └─ traits/
3 |     └─ STARLIGHT_GLIMMER_RESISTANCE_LEADER/      # 特质
        TRAIT_STARLIGHT_GLIMMER_RESISTANCE_LEADER
4 |     |   └─ info.json          # 特质代码（修正、是否随机）
5 |     |   └─ locs.txt           # 特质本地化（名称），特质无需图标
6 |     └─ STARLIGHT_GLIMMER_FORMER_PRINCIPLE/      # 特质
        TRAIT_STARLIGHT_GLIMMER_FORMER_PRINCIPLE
7 |     |   └─ info.json          # 特质代码（修正、是否随机）
8 |     |   └─ locs.txt           # 特质本地化（名称），特质无需图标
9 |     └─ ...
10 |   └─ ...
11 └─ ...

```

角色代码 resources/characters/STARLIGHT\_GLIMMER/info.json :

```

1 {
2   "gender": "female",
3   "country_leader": {
4     "ideology": "harmonicisism_republic",
5     "traits": [
6       "TRAIT_STARLIGHT_GLIMMER_RESISTANCE_LEADER",
7       "TRAIT_STARLIGHT_GLIMMER_FORMER_PRINCIPLE"
8     ]
9   }
10 }

```

角色本地化 resources/characters/STARLIGHT\_GLIMMER/locs.txt.json :

```

1 [en.@NAME]
2 Starlight Glimmer
3 [zh.@NAME]
4 星光熠熠
5
6 [zh.@DESC]
7 星光熠熠的过去与现在仿佛处于两个不同的时空，在过去，她是平等镇的残酷独裁者，为了实现自己的“平等”而毫不犹豫给镇民们带去了痛苦，还差点制造了一场毁灭世界的危机。而现在，她是暮光闪闪最为器重的学生与最好的伙伴，前友谊学院的辅导员以及抵抗军当之无愧的领导者。但无论她是什么身份，熟悉星光熠熠的小马都能看出她是一名理想主义者，只要是星光熠熠认为正确的目标，那么她就会毫不犹豫地朝着那个方向努力。而她强大的、寻遍整个小马利亚都难出其二的魔法天赋，则成了她实现理想的强大助力。

```

- 8 在终末之战后，星光熠熠成功逃脱了和煦光流的追捕并辗转来到了平等镇，这里位于帝国边境，位置偏僻，是藏身的绝佳地点。星光熠熠安顿下来后，便马不停蹄地开始了根据地的建设，她以平等镇为核心，团结起了镇民，召集了周边小马，宣告了“星光抵抗军”的成立，开始号召小马们起来反对并推翻和煦光流伪帝国的统治。
- 9 随着时间的推移，星光熠熠也联系上了曾经的好友——无序和余晖烁烁，他们都在以自己的方式反击帝国的统治。但令星光熠熠伤心的是，自己曾经的挚友特丽克西已经臣服于和煦光流的统治。同时，阵线内部也还有许多问题函待解决，而帝国对阵线的敌意开始与日俱增，越来越多的部队被布置到边境，冲突一触即发。星光熠熠明白，留给自己的时间已经不多了。
- 10 每当星光熠熠感到困难时，她就会回忆起与暮光闪闪，与特丽克西，与伙伴们的点点滴滴。她会回忆起暮光闪闪曾经的教导：“别再牵挂那段失败的去，想象还有很多美好的未来在等着你。”是的，前路漫漫但并非绝境，这不只是星光熠熠一马的抗争，这是全小马利亚，每一匹处于压迫之下的小马的抗争！无论自己成功与否，至少小马们明白光明仍在。终有一日，笼罩天空的乌云将被驱散，寒冷的大地将会回暖，一缕友谊的阳光将穿破黑暗，重新照耀小马利亚的每一片土地。
- 11 [en.@DESC]
- 12 The current Starlight Glimmer is nothing like the one from the past. In the past, she was a cruel dictator of the Our Town, who brought pain to the residents in the name of “equality” and almost caused a world-ending crisis. Now, she is the most trusted student and best friend of Twilight Sparkle, the former counselor of the School of Friendship, and the leader of the Starlight Resistance. But no matter what her identity is, those who are familiar with Starlight Glimmer can tell that she is an idealist. As long as she believes in a goal, she will strive towards it without hesitation. Her powerful magic talent, which is second to none in all of Equestria, has become a powerful aid in achieving her ideals.
- 13 After the Ultimate Battle, Starlight Glimmer successfully escaped the pursuit of Cozy Glow and came to Our Town, which is located on the border of the Empire and is a remote location, making it an excellent hiding place. After settling down, Starlight Glimmer immediately began to build a base, using Our Town as the core, uniting the residents, gathering the surrounding ponies, and declaring the establishment of the “Starlight Resistance”, calling on the ponies to rise up against and overthrow the illegal empire of Cozy Glow.
- 14 As time passed, Starlight Glimmer also contacted her former friends – Discord and Sunset, who were fighting against the Empire in their own ways. But what saddened Starlight Glimmer was that her former friend Trixie had already submitted to the rule of Cozy Glow. At the same time, there were still many problems within the Front that needed to be resolved, and the Empire’s hostility towards the Front was increasing day by day, with more and more troops being deployed to the border, and conflict was imminent. Starlight Glimmer understood that she didn’t have much time left.

15 Whenever Starlight Glimmer felt difficult, she would recall the moments with Twilight Sparkle, with Trixie, and with her friends. She would recall the teachings of Twilight Sparkle: “When one journey ended it led us here, but just imagine all the other good things that are out there waiting for you if you just give them a chance.” Yes, the road ahead is long but not impassable. This is not just Starlight Glimmer’s struggle, but the struggle of all of Equestria, every pony under oppression! Whether she succeeds or not, at least the ponies understand that the light is still there. One day, the dark clouds that shroud the sky will be dispelled, the cold earth will warm up, and a ray of friendship’s sunlight will pierce the darkness, shining on every corner of Equestria.

编译所有角色的代码 scripts/C05\_add\_characters.py :

```
1  # ===== #
2  # ==== CHAPTER 5: Add Characters ==== #
3  # ===== #
4
5  # %%
6  # Import hoi4dev
7  from hoi4dev import *
8  # Fix the random seed
9  import numpy as np
10 np.random.seed(42)
11
12 # First add all traits. The resources of the traits are located in
13 # `resources/traits`.
14 for trait in TQDM(ListFolders("resources/traits", ordered=True), desc='Building
15 # traits...'):
16     path = pjoin("resources", "traits", trait)
17     if ExistFolder(path) and ExistFile(pjoin(path, "info.json")):
18         AddTrait(path, translate=False) # 使用HOI4DEV的`AddTrait`方法添加一个特质到mod带
19         # 编译文件夹的`data/`中
20
21 # %%
22 # Now add all characters. The resources of the characters are located in
23 # `resources/characters`.
24 for character in TQDM(ListFolders("resources/characters", ordered=True),
25 # desc='Building characters...'):
26     path = pjoin("resources", "characters", character)
27     if ExistFolder(path) and ExistFile(pjoin(path, "info.json")):
28         AddCharacter(path, translate=False) # 使用HOI4DEV的`AddCharacter`方法添加一个角
29         # 色到mod带编译文件夹的`data/`中
30
31 # %%
```

```
26 AddRandomCharacters("resources/characters_random") # 使用H0I4DEV的
    `AddRandomCharacters`方法将随机角色肖像添加到角色池中
```

在执行 scripts/C05\_add\_characters.py 后，我们可以用 CompileMod() 来编译得到一个可以运行的mod，在游戏中检查我们添加的角色是否正确。

## 示例2：添加决议

添加决议的代码与上述添加角色的代码类似，但是注意决议包含在决议组内。首先，源文件结构：

```
1 resources/
2 |   └─ decisions/
3 |   |   └─ C01_UNIFICATION_WAR/      # 决议组：C01通过战争进行国家统一的决议
4 |   |   |   └─ icon.png              # 决议组小图标
5 |   |   |   └─ info.json             # 决议组代码（优先级、触发条件）
6 |   |   |   └─ locs.txt              # 决议组本地化（名称、描述）
7 |   |   |   └─ C01_C02_WAR/          # 决议：准备对C02国家宣战
8 |   |   |   |   └─ default.png       # 决议小图标
9 |   |   |   |   └─ info.json         # 决议/任务代码（时间、效果、优先级、触发条件等）
10 |   |   |   |   └─ locs.txt         # 决议本地化（名称、描述）
11 |   |   |   └─ ...
12 |   |   └─ ...
13 |   └─ ...
14 └─ ...
```

决议代码 resources/decisions/C01\_UNIFICATION\_WAR/C01\_C02\_WAR/info.json：

```
1 {
2     "name": "Declare Ponyville",
3     "priority": 100,
4
5     "cost": 150,
6     "days_remove": 42,
7     "fire_only_once": true,
8     "visible": {
9         "has_completed_focus": "FOCUS_C01_LAST_RESORT",
10        "C02": {
11            "has_country_flag": "PIHC_COUNTRY_FLAG_INDEPENDENT_FROM_C01_TIMEOUT"
12        }
13    },
14    "available": {
15        "if": {
```

```

16         "limit": {
17             "ROOT": {
18                 "has_country_flag": "PIHC_COUNTRY_FLAG_NO_FOCUS"
19             }
20         },
21         "custom_trigger_tooltip": {
22             "tooltip": "PIHC_COUNTRY_FLAG_NO_FOCUS_TOOLTIP",
23             "always": false
24         }
25     }
26 },
27     "war_with_on_remove": "C02",
28     "remove_effect": {
29         "declare_war_on": {
30             "type": "puppet_wargoal_focus",
31             "target": "C02"
32         }
33     },
34     "ai_will_do": {
35         "base": 0
36     }
37 }

```

决议本地化 resources/decisions/C01\_UNIFICATION\_WAR/C01\_C02\_WAR/locs.txt :

```

1  [en.@]
2  Prepare for Suppressing Ponyville
3  [zh.@]
4  对小马谷的镇压准备
5
6  [en.@desc]
7  Ponyville has been independent from the Empire, we need to suppress their resistance
   and show the power of the Empire.
8  [zh.@desc]
9  小马谷已经从帝国治下独立，我们要镇压他们的反抗，展示帝国的力量。

```

编译所有决议和决议组的代码 scripts/C08\_add\_decisions.py :

```

1  # ===== #
2  # ==== CHAPTER 8: Add Decisions ==== #
3  # ===== #
4
5  # %%

```



```
6 # Import hoi4dev
7 from hoi4dev import *
8 # Fix the random seed
9 import numpy as np
10 np.random.seed(42)
11
12 # %%
13 # PIHC has a core decision for every central equestria region. Let's add them.
14 # from scripts.central_equestria_script_legacy import
15 #     gen_central_equestria_core_decisions
16 # gen_central_equestria_core_decisions()
17
18 # %%
19 # Let's add all decision categories. The resources of the decisions are located in
20 #     `resources/decisions`.
21 for category in TQDM(ListFolders("resources/decisions", ordered=True),
22     desc='Building decisions...'):
23     path = pjoin("resources", "decisions", category)
24     AddDecisionCategory(path, translate=False) # 使用HOI4DEV的`AddDecisionCategory`方
25     法添加一个决议组到mod带编译文件夹的`data/`中
```

在执行 `scripts/C08_add_decisions.py` 后，我们可以用 `CompileMod()` 来编译得到一个可以运行的mod，在游戏中检查我们添加的决议组是否正确。