

WARNO 模组制作手册 v1

创造一个新的模组

转到游戏安装目录(从steam中下载的: Library -> WARNO -> Properties -> Local Files -> Browse Local Files),并进入 `Mods` 子目录

在Mods目录中打开控制台 (使用(DOS命令) `cd <模组文件夹路径>` 或者 按住 `shift + 右键` 在windows资源管理器中 > "在此处打开控制台(/在此处打开PowerShell)") 并以mod名称作为参数运行

`CreateNewMod.bat`. 例如:

```
CreateNewMod.bat MyAwesomeMod
```

名称必须唯一; 如果已经存在具有该名称的文件夹,则它将不起作用. 如果一切顺利, 你应该会看到一条消息,表明mod已经创建,并且出现了一个名为 `MyAwesomeMod` 的新文件夹. 它应该包含以下目录:

- **AssetsModding**
- **CommonData**
- **GameData**
- **Gen**

以及以下文件:

- **base.zip**
- **GenerateMod.bat**
- **UpdateMod.bat**
- **UploadMod.bat**

这4个文件 (`base.zip` 和3个 `.bat` 后缀文件) 是模组制作工具的一部分, 请不要修改它们,否则你可能会损坏你的mod.

如果缺少什么,请尝试删除mod文件夹并重新创建它.

你现在可以开始制作你的mod了,请继续阅读!

生成mod

在游戏中激活mod或上传到Steam创业工坊之前,需要一个生成步骤.如果你想接着做,请在mod目录中启动**GenerateMod.bat**. 屏幕将在短时间内变黑.

当你生成了你的mod,你的mod就会得到一个自动生成的默认的SteamID. 你需要上传你的mod以赋予其一个正式的SteamID. 即使你还没有完成你的工作,你也可以上传并将其隐藏在您的工作室mod页面上.

编辑mod配置文件

转到 `C:\Users\USERNAME\保存的游戏\EugenSystems\WARNO\mod\ MyAwesomeMod` 文件夹, 并打开文件 `Config.ini`:

```
Name = MyAwesomeMod ; 你的mod的名称
Description = My Mod is Awesome ; 你对mod的描述
TagList = ; 请不要编辑该字段
PreviewImagePath = C:\my_mod_image.png ; 在你的硬盘上显示出的图像
CosmeticOnly = 0 ; 你对mod是否会影响游戏性 ? 如果是,那么创建大厅时不会分享该mod
ID = 902495510 ; mod的steamID,请不要修改
Version = 0 ; 当此mod中的更新与当前版本不兼容时要增加的值
DeckFormatVersion = 0 ; 增加该值可使所有卡槽失效
ModGenVersion = 0 ; 请不要编辑该字段
```

如果你想更改mod名称,你必须:更改配置中的"名称"字段文件,更改C:\Users\USERNAME\保存的游戏\EugenSystems\WARNO\mod中的文件夹名称,更改steam目录中的文件夹名.

UploadMod.bat启动时会自动填写"ID"和"TagList".

上传mod

要在steam上上传你的mod,请在你的mod目录中启动**UploadMod.bat**.屏幕将在短时间内变黑.

此过程将更新"TagList","ID"和"ModGenVersion"字段.

一旦你上传了你的mod,它就可以在steam workshop上使用了,你可以访问它自己的页面,设置它的可见性,描述,图像等...

更新mod - 当新游戏补丁发布时

随着游戏通过Steam更新,需要不时更新你的mod,以便它仍然可以与新版本一起使用. 我们创建了一个小工具来帮助您完成这项工作,只需最少的手动操作. 在谈论它之前,让我们看看问题出在哪里.

为什么我们需要更新mods?

想象一下,你有一个修改文件a.ndf的mod.当这个mod创建时,原始文件

(来自游戏数据)是:

```
Thing is TThing
(
    SomeInt = 42
)
```

你把它修改成:

```
Thing is TThing
(
    SomeInt = 43
)
```

没问题,这就是修改的重点. 现在,如果我们Eugen也需要修改这个文件,并最终向其中添加内容,使其在新版本的游戏中看起来像这样,该怎么办:

```
Thing is TThing
(
    SomeInt = 42
)

Thing2 is TThing
(
    SomeInt = 80
)
```

你会想保持你对Thing/SomeInt值的更改,并拥有新的Thing2.
你的mod的理想结果:

```
Thing is TThing
(
    SomeInt = 43
)

Thing2 is TThing
(
    SomeInt = 80
)
```

跟新mod

解决方案非常简单,只需在mod文件夹中运行UpdateMod.bat. 这将把你的更改与上一版本的游戏数据[合并](#).

如果你幸运的话,这个过程会自动运行,并显示一条成功的消息. 但是,在操作过程中可能会遇到冲突. 当新版本的游戏和你的mod同时修改了NDF文件的同一部分时,就会发生冲突. 在这种情况下,该程序无法猜测您想要的结果,此时需要手动干预.

解决冲突

再次想象一下,你正在修改A.ndf.
原始版本(创建mod时的游戏数据):

```
Thing is TThing
(
    SomeInt = 42
)
```

修改后的版本:

```
Thing is TThing
(
    SomeInt = 43
)
```

随着以下更新,游戏版本的文件变成了:

```
Thing is TThing
(
    SomeInt = 80
)
```

游戏和mod都修改了同一行. 你可以很容易地看到,Thing/SomeInt在你的mod中应该有什么最终值完全取决于你,没有任何工具可以自主的决定它.

更新工具将警告您文件的冲突并停止它. 为了处理这种情况,该工具将用特殊的分隔符标记像这样的冲突:

```
Thing is TThing
(
<<<<<<<
    SomeInt = 80
|||||||

    SomeInt = 42
\=====
    SomeInt = 43
\>>>>>>>
)
```

- 在 <<<<<<< 之后的是新版本中的游戏数据
- 在 ||||||| 之后的是共同的基础源数据(在我们的案例中是原始游戏的数据)
- 在 ===== 之后的是你的当前版本mod中的数据
- >>>>>>> 表示着冲突代码结束

解决冲突包括修改文件,使游戏能正确加载它(即,去掉分隔符共有的一行SomeInt=...),并修改为您的mod所需的值. 你可以自由地保留你的mod值 (43),回到原始值 (42), 获得新值 (80) 或者使用一个其它值.

这个过程与git等版本控制软件中的三方合并非常相似.

Mod的备份

在你的mod目录中,你会发现2个必要的脚本:

- CreateModBackup.bat :这将备份你当前的mod状态. 备份基本上是存档并存储位于备份/文件夹中的mod文件副本. 每次上传你的mod时都会生成一个备份,考虑到既然你都上传了你的mod,那么你目前的mod版本八成是稳定的-_-您可以使用此脚本自己生成备份,该脚本将要求你提供名称, 您可以留空,它将生成并使用基于当前时间的名称.
- RetrieveModBackup.bat : 该脚本将把你的mod还原到你存储在Backup/中的历史版本, 它只需复制并强制粘贴备份版本到当前的工作空间中, 因此在进行RetrieveModBackup之前请您三思. 脚本将提示您输入名称, 您可以将条目留空, 它将选择最近生成的备份.

NDF Map

您可能要编辑的每个文件位置.

游戏界面

如果你想修改界面, 你需要查看GameData/UserInterface/ .

Style/文件夹下的所有文件都将与全局对象相关, 并在不同的屏幕上重复使用样式.

文件夹 Use/ 下的所有文件将描述每个屏幕或屏幕的一部分，当在游戏中使用后需在文件夹中区分为：

- InGame : 当比赛启动后游戏界面上的所有东西
- OutGame : 与主菜单、加载等相关的一切...
- Common : InGame和OutGame的所有共同界面.
- ShowRoom : 不言自明,与展厅(游戏里的军械库,也许-_-)有关.
- Strategic : 与Steelman模式相关的所有内容

游戏设置

单位

所有单位描述文件都位于

GameData/Generated/Gameplay/Gfx/UniteDescator.ndf中. 您会在同一文件夹中找到其他各种有趣的描述文件，如WeaponDescriptor.ndf和Ammunity.ndf.

师/团

师/团位于GameData/Generated/Gameplay/Decks/Divisions.ndf中 .师/团由队组成，队是由相同单位组成的，所有队都位于GameData/Generated/Gameplay/Decks/Packs.ndf .

游戏参数

GameData/Gameplay/Constantes/包含定义通用游戏功能的所有文件,例如:

- 在GDConstants.ndf中，您可以找到定义重影(原文为ghosts)颜色的属性DefaultGhostColors
- RelativeBonusMoneyByIADeficity允许你给AI更多的收入

武器

武器主要在WeaponDescriptor.ndf和Ammunity.ndf中进行描述，它们位于GameData/Generated/Gameplay/Gfx中.

第一个文件描述了每个单位的炮塔是如何配置的，以及安装了哪些武器. 例如，您可以找到定义每种武器齐射次数、炮塔的角度范围和旋转速度的属性，以及一些与UI相关的参数. 您还可以找到对TAcommunicationDescriptor对象的引用，这些对象在Ammunity.ndf中定义 . 在这些对象中有许多有用的属性，以下是其中的一些:

- Arme: 武器将造成的伤害类型. 对于AP武器，您可以使用从AP_1到AP_30的值（请参阅CommonData/Gameplay/Constantes/Enumerations/ArmeType.ndf） .
- PorteeMaximale : 武器的射程。你们中的一些人已经明白了这个单元的含义（参见[本文](#)） .
- HitRollRuleDescriptor : 描述了如何计算命中和刺穿几率（请参阅GameData/Gameplay/Constantes/HitRollConstants.ndf） .
- IsHarmlessForAllies : 启用/禁用友军伤害。

您可能还想查看其他文件:

- GameData/Gameplay/Unit/CriticalModules/CriticalEffectModule_GroundUnit.ndf: 描述了地面部队（通常是坦克）的关键影响分布.

AI

GameData/Gameplay/Skirmish包含有关人工智能的文件。人工智能被称为策略。根据游戏难度和设置，有多种策略（用于遭遇战、多人游戏、历史战斗和steelman），位于GameData/Gameplay/skirmish/strategies/子文件夹中。

策略插入GameData/Gameplay/Skirmish/IASkirmishDescriptors.ndf.

策略包含Generators和Transitions列表。AI根据TIAGeneralStrategyTransition从一个TSequenceGeneratorDescriptor导航到另一个。

TSequenceGeneratorDescriptor描述AI活动时的优先级。生成器中使用的MacroActionBehavior描述符位于GameData/Gameplay/Skirmish/Skirmish Macros文件夹中。大多数MacroActions控制具有特定目标（攻击、防御等）的单位组。每个操作所需的单位在TPoolModel对象中进行了描述，这些对象可以自由修改。注意：新的TPoolModel对象需要不同的GUID字段。

关于AI的全局设置在

GameData/Gameplay/Constantes/IStratConstantes.ndf中定义，并影响遭遇战和战役AI。

逐步创建你的第一个mod.

本节的目标是创建一个mod，允许创建更多单位的甲板（100个槽点，而不是50个）。

按照本手册第一部分的说明，创建一个新的mod，创建你的mod MyAwesomeMod.

从您的mod文件夹中，打开文件GameData/Generated/Gameplay/Decks/Divisions.ndf

将所有MaxActivationPoints=50替换为MaxActivationPoints=100.

按照生成mod部分的说明生成您的mod.

现在你可以在游戏中通过菜单中的模组中心激活和测试你的mod .

按照上传mod部分的说明进行操作 .

祝贺你在WARNO工作室创建并发布了你的第一个mod.