Руководство по созданию дополнений

вер. 2011.06.30

Graviteam ®

СОДЕРЖАНИЕ

1 ОБЩИЕ СВЕДЕНИЯ О СОЗДАНИИ ДОПОЛНЕНИЙ

Для создания дополнений необходимо установить патч №5 либо более новый, поверх оригинальной игры. Для распаковки архивных файлов игры и создания собственных дополнений предназначены специальные команды. Для вызова команды используется следующий формат?

starter.exe <название команды>, <параметры команды>

Название команды и параметры разделяются знаком ",". Для запуска команды необходимо воспользоваться либо файловой оболочкой типа Far (http://www.farmanager.com/download.php), либо создать командный файл операционной системы (с расширением bat или cmd) в корневой папке игры.

Результаты преобразований записываются в log-файлы в папке out в корневом каталоге игры (с названием соответствующим названию команды).

При запуске команды без параметров, появляется диалоговое окно позволяющее выбрать файл для распаковки либо упаковки. Командные файлы для быстрого запуска команд находятся в папке "docs\modwork\" в каталогах:

- asets – для работы с ресурсами;
- cfgtext – для работы с текстами и настройками;
- flatwork – для работы с архивами.

При использовании команды mkflat необходимо чтобы в том месте, где создается архив, присутствовал каталог с названием, совпадающим с названием архива, в котором должны находится файлы, помещаемые в архив, и файл описания архива: <имя архива>.!flatpack

Распакованные файлы помещаются в папку "users\modwork", которую целесообразно создать перед началом работы с модификациями.

Примеры командных файлов для групповой обработки приведены в папке "docs\modwork\examples", а шаблоны файлов параметров в "docs\modwork\stencil".

1.1 Работа с архивами игры

Для работы с архивами игры используются команды mkflat и unflat, для создания и распаковки архива. Файлы архивов должны иметь расширение flatdata.

Пример создания нового архива:

```
starter.exe  mkflat,  users\modwork\my_addon.flatdata, users\modwork\my_addon.!flatlist
```

Эту команду необходимо вызывать из корневого каталога игры. В результате будет создан (в папке "users\modwork") архив my_addon.flatdata, в который будут добавлены файлы перечисленные в файле my_addon.!flatlist.

Файл содержащий описание добавляемых файлов (my_addon.!flatlist) должен быть построен по следующим правилам:

- начинаться с заголовка i_unflat:unflat();
- на следующей строке должна быть фигурная скобка "{";
- дальше должен быть перечислен список файлов, их формат и локаль для которой они актуальны, эти параметры должны быть разделены запятыми;
- заканчиваться файл должен символом "}".

Пример:

```
i_unflat:unflat()
{
	acrates		, config	, loc_def ;
	ai_plans	, config	, loc_def ;
	ammo		, config	, loc_def ;
	anims		, config	, loc_def ;
}
```

Название файла строится следующим образом: <имя файла>.<локаль>.<тип>

Для распаковки существующего архива необходимо воспользоваться командой unflat.

```
starter.exe  unflat,  users\modwork\game_archive.flatdata, users\modwork\game_archive
```

Эта команда распакует файл game_archive.flatdata в папку "users\modwork\game_archive" и создаст список распакованных файлов

game_archive.!flatlist который можно будет использовать для последующей запаковки.

Типы файлов перечислены в таблице 1

Таблица 1

Типы файлов

Расширение	Тип файла
text	Текстовые файлы
config	Файлы настроек
program	Описание команд и частей программ
mesh	Геометрия объектов
sound	Звуки и музыка
fontmap	Карты шрифтов
armor	Карты бронирования
image	Изображения
texture	Текстуры

1.2 Текстовые файлы

Для работы с текстовыми файлами применяются команды text2pd и pd2text.

Пример распаковки текстового файла:

starter.exe text2pd, users\modwork\text_file.loc_rus.text, users\modwork\ text_file.loc_rus.engcfg2

преобразует файл text_file.loc_rus.text в конфигурационный файл text_file.loc_rus.engcfg2. Если второй параметр не задан полученный файл будет находиться в той же папке что и распаковываемый, но будет иметь расширение engcfg2.

Текст представляет собой набор таблиц, каждая из которых начинается с индентификатора состоящего, из латинских букв (в нижнем регистре) и цифр, за которым, в фигурных скобках, следуют одна или несколько строк разделенных ";". Все таблицы помещены в общий блок определяющий локаль.

Длина индентификатора таблицы не должна превышать 31 символ.

Например:

```
//локаль
loc_rus()
{
	//таблица состоящая из 1-й строки
	txt_text1[s]() { Текст №1; }
	txt_text2[s]() { Текст №2; }

	//таблица состоящая из нескольких строк
	txt_big_text[s]()
	{
		Таблица. Текст 1;
		Таблица. Текст 2;
		Таблица. Текст 3;
		Таблица. Текст 4;
	}

}
```

В тексте недопустимы символы "{", "}" и ";". Если возникает необходимость в задании таких символов, а так же специальных символов перевода строки и табуляции, необходимо воспользоваться предваряющим

символом "$". Для задания табуляции "$t", для перевода строки "$n", для задания цвета $<номер цвета>: 1 – черный, 2 – зеленый, 3 – желтый, 4 – красный, 5 – белый, 6 – серый, 7 – синий, 8 – фиолетовый.

Для запаковки текстового файла используется команда:

```
starter.exe pd2text, users\modwork\text_file.loc_rus.engcfg2, users\modwork\ text_file.loc_rus.text
```

преобразует конфигурационный файл text_file.loc_rus.engcfg2 в текстовый файл text_file.loc_rus.text. Если второй параметр не задан полученный файл будет находиться в той же папке что и распаковываемый, но будет иметь расширение text.

Пример задания текстового конфигурационного файла stencil\text_example.loc_rus.engcfg2

1.3 Файлы настроек

Для работы с файлами настроек применяются команды cfgp2pd и pd2cfgp.

Пример распаковки файла настроек:

starter.exe cfgp2pd, users\modwork\tab.loc_def.config, users\modwork\tab.loc_def.engcfg2

преобразует файл настроек tab.loc_def.text в конфигурационный файл tab.loc_def.engcfg2. Если второй параметр не задан полученный файл будет находиться в той же папке что и распаковываемый, но будет иметь расширение engcfg2.

Настройки представляют собой набор блоков двух типов: список констант и таблица. Каждый блок начинается с индентификатора состоящего, из латинских букв (в нижнем регистре) и цифр, за которым, в фигурных скобках, следует несколько строк разделенных ";". Для каждой константы должно быть указано название и формат, в квадратных скобках, после которого через знак "=" идет значение константы.

Длина индентификатора блока или названия константы не должна превышать 31 символ.

В названии таблицы (в квадратных скобках) должен быть указан формат для каждой ячейки таблицы, а для списка констант символ "=". Формат конфигурационного файла рассмотрен в разделе 3.2.

Для запаковки файла настроек используется команда:

starter.exe pd2cfgp, users\modwork\tab.loc_def.engcfg2, users\modwork\tab.loc_def.config

преобразует конфигурационный файл tab.loc_def.engcfg2 в файл настроек tab.loc_def.config. Если второй параметр не задан полученный файл будет находиться в той же папке что и распаковываемый, но будет иметь расширение config.

Пример задания файла конфигурации для описания настроек stencil\desc_example.addpack.engcfg2.

1.4 Ресурсы игры

Для хранения ресурсов игры используются специальные форматы: ATF для хранения текстур и изображений, AAF для хранения звуков и музыки, GO2 для хранения геометрии объектов. Эти форматы не предназначены для непосредственного изменения и редактирования, поэтому для работы с ними необходимо преобразовать файлы в этих форматах в другие форматы, предназначенные для непосредственного редактирования. И после редактирования провести обратное преобразование.

1.4.1 Текстуры и изображения

Для конвертирования текстур предназначены команды atf2dds и dds2atf, которые позволяют преобразовать текстуры из формата ATF в DDS и обратно.

Пример преобразования текстуры:

```
starter.exe atf2dds, users\modwork\reg_tex_dift.loc_def.texture, users\modwork\ reg_tex_dift.loc_def.dds
```

преобразует текстуру reg_tex_dift.loc_def.texture в reg_text_dift.loc_def.dds. Если второй параметр не задан полученный файл будет находиться в той же папке что и распаковываемый, но будет иметь расширение dds.

Для редактирования текстур в формате dds можно использовать ряд программ:

- Paint.NET, ссылка http://www.getpaint.net/index.html;
- GIMP, ссылка http://gimp-win.sourceforge.net/stable.html;
  DDS plugin http://nifelheim.dyndns.org/~cocidius/dds/;
- плагин к Adobe® PhotoShop® от nVidia®
  http://developer.nvidia.com/object/photoshop_dds_plugins.html.

При редактировании текстуры обращайте внимание на ее формат и наличие MIP-уровней. Эти параметры не должны изменяться!

Суффикс "dift" в названии текстуры обозначает диффузную карту (каналы RGB) и карту прозрачности (канал A), суффикс "norsp" – карту нормалей (каналы RG) карту блеска (канал A) и карту шероховатостей (канал B).

В таблице 2 приведены характеристики основных форматов текстур.

Таблица 2

Форматы текстур

Префикс	Формат/ MIP уровни	Описание
bump	DXT5	бамп текстуры (нормали, блеск и шероховатость)
trans	DXT5	диффузные текстуры с полупрозрачностью
reg	DXT1	диффузные текстуры с однобитным альфаканалом
lbump	DXT5	текстуры ландшафта
clouds	DXT5	текстуры облаков и небосвода
detail	DXT5/1	детальные текстуры
coc	DXT1	текстуры кабин (не используются)
menu	DXT5	текстуры меню и интерфейса
map	DXT5	текстуры карты
mapback	DXT1	подложка тактической карты
font	DXT5/1	изображения шрифтов
uncomp	RGB8	некомпрессируемые текстуры
user	DXT5	пользовательские картинки (не используются)

1.4.2 Звуки и музыка

Для конвертирования звуков в формат, используемый игрой, предназначена команда wav2aaf, которая предназначена для преобразования звуков в формате WAV.

Пример преобразования звука:

starter.exe wav2aaf, users\modwork\my_snd.loc_def.wav, users\modwork\ my_snd.loc_def.sound

Для "трехмерных" звуков (выстрелы, взрывы, звуки техники и пр.) используется формат 44100 Гц (44КГц) 16 бит МОНО. Для системных звуков 44100 Гц (44КГц) 16 бит СТЕРЕО.

Для музыки и фоновых звуков формат xWMA, который можно получить утилитой из состава DirectX SDK® xWMAEncode, сконвертировав некомпрессированный звуковой файл в формате WAV в формате 44КГц, 16 бит СТЕРЕО.

Пример преобразования:

```
xWMAEncode -b 160000 amb_can_0.wa_ amb_can_0.wav
```

Файл в формате WAV "amb_can_0.wa_" будет преобразован в файл в формате xWMA "amb_can_0.wav".

DirectX SDK можно скачать по следующей ссылке: http://www.microsoft.com/downloads/details.aspx?FamilyID=b66e14b8-8505-4b17-bf80-edb2df5abad4&displaylang=en (553.3 Мб)

1.4.3 Смена материала объекта

Для смены материала у существующих геометрических объектов (техники, солдат, сооружений, растительности и т.д.) предназначена команда tex_changer.

Пример смены материала:

```
starter.exe tex_changer, users\modwork\mesh.loc_def.mesh, ginf_dift, ginf_norsp
```

Первым параметром передается файл геометрии (с расширением mesh), вторым название текстуры цвета и прозрачности, без префикса, но с суффиксом "dift". Последним параметром передается текстура нормалей, яркости блика и шероховатости, без префикса, но с суффиксом "norsp".

Эта команда не может быть вызвана без параметров!

1.4.4 Геометрия объектов

Для конвертирования геометрии из формата X предназначена команда x2go. Пример преобразования геометрии:

starter.exe x2go, users\modwork\my_meshl0.X, users\modwork\ my_mesh.loc_def.mesh

преобразует геометрию my_meshl0.X, my_meshl1.X, …. в my_mesh.loc_def.mesh, собирая все уровни детализации в один файл. Если второй параметр не задан полученный файл будет находиться в той же папке что и распаковываемый, но будет иметь расширение mesh.

Уровни детализации и физический уровень должны быть сконвертированы в формат X при помощи встроенных средств DCC среды (например, Blender) либо при помощи сторонних плагинов, например **Panda DirectX Exporter,** который можно скачать по следующей ссылке http://www.andytather.co.uk/Panda/directxmax_downloads.aspx.

Названия файлов должны быть следующими:

1) <название>l0.X для физического уровня;

2) <название>ln.X для видимых уровней детализации (которых должно быть 3 штуки), где n – номер уровня от 1 до 3; **l – буква "L" в нижнем регистре**.

При запуске команды без параметров появится диалог выбора файла, в котором необходимо выбрать любой из уровней детализации. Все уровни детализации должны находиться в одной папке!

Для просмотра сконвертированной геометрии (либо любой геометрии из игры) используется команда "model_view".

Пример просмотра модели:

starter.exe model_view, users\modwork\ my_mesh.loc_def.mesh

покажет на экране внешний вид модели с установленными на нее текстурами. Если команда запущена без параметров, откроется окно, в котором нужно выбрать модель для просмотра. **Текстуры, наложенные на модель, должны находиться в ресурсах игры!**

1.5 Карты бронирования

Для преобразования карт бронирования из формата TGA в формат, используемый игрой предназначена команда tga2am.

Пример преобразования карты бронирования:

```
starter.exe tga2am, users\modwork\armor.loc_def.tga,
```

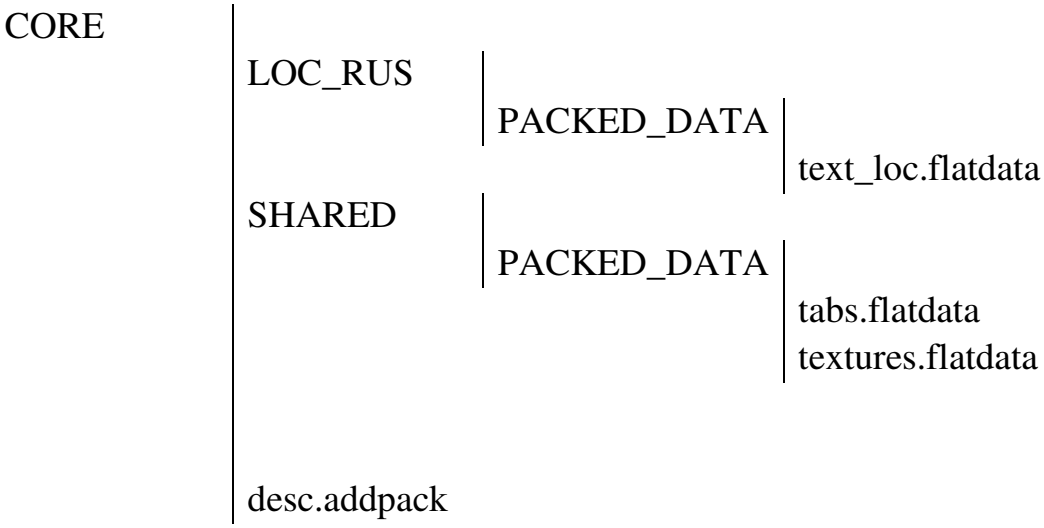
При вызове команды без параметров появится диалог в котором можно будет выбрать файл содержащий карту. Формат TGA файла должен быть 32 битным. Каналы R, G и B должны быть идентичными, в альфа-канале белым цветом должны быть обозначены точки имеющие броню, черным полностью прозрачные точки.

Сконвертированные карты бронирования должны иметь расширение "armor".

2 СОЗДАНИЕ ДОПОЛНЕНИЯ

Пользовательское дополнение должно иметь следующую структуру:
- файл readme.txt – текстовое описание дополнения;
- папка CORE – содержащая файлы дополнения и информацию для установки.

В папке CORE должен содержаться файл "desc.addpack" с информацией для установки дополнения, а так же папки loc_rus и shared в которые помещены локальные или общие ресурсы дополнения (в папки packed_data в архивах). Пример иерархии дополнения показан ниже:



readme.txt

Создать описание дополнения (desc.addpack) можно путем редактирования и последующего преобразования шаблона "stencil\desc_example.addpack.engcfg2".

Пример шаблона для установки дополнения:

```
i_updater:updater=()
{
	//путь к дополнению
	path[s] = <my_updates>;
	//название дополнения
	desc[s]  = <My Addon>;
	//автор(ы) дополнения
	authors[s] = <Vasya Pupkin>;
	//версия дополнения
	version[u] = 100;
	//тип дополнения
	//CAMP - кампании/полигоны,
	//RES - обновление ресурсов,
	//ADDN - аддон
	type[*] =  RES;
	//удалять пердыдущую версию дополнения
	//рекомендуется ставить true
	clear_prev[b] = true;
	//версия игры на которую ставится дополнение (в 16-ричной системе)
	//если установлен флаг 0x80000000 – ставится
	//только на указанную версию
	eng_ver[u] = 0x0000050b;
	//путь к системным файлам
	//(для пользовательских дополнений оставлять пустым)
	sys_path[s] = ;
	//файл используемый для сохранения заменяемых файлов
	//оставлять пустым
	recover[s] = ;

}
```

ВНИМАНИЕ! Строки, записанные в угловых скобках (<>), необходимо заменить своими собственными без угловых скобок.

2.1 Использование командных файлов ОС

Для обработки нескольких файлов желательно применять командные файлы ОС – это текстовые файлы с расширением bat или cmd, которые позволяют выполнять несколько команд последовательно.

В корневом каталоге игры создайте текстовый файл my_addon.cmd, в который в текстовом редакторе (notepad) внесите команды необходимые для сборки дополнения. Теперь чтобы пересобрать дополнение достаточно только запустить этот файл при помощи менеджера файлов.

Примеры использования командных файлов:
- для работы с архивами unflat_example.cmd и mkflat_example.cmd;
- для работы с текстом text2pd_example.cmd и pd2text_example.cmd;
- для работы с настройками cfgp2pd_example.cmd и pd2cfgp_example.cmd;

3 ПРОЧЕЕ

3.1 Локали

Для каждого ресурса помещаемого в архив необходимо задавать локаль. Если ресурс используется во всех версиях игры, локаль должна быть "loc_def". Для таких ресурсов как изображения с надписями, тексты и шрифты, необходимо указать конкретную локаль языка для которого они созданы: loc_rus – для русского, loc_eng – для английского, loc_ger – для немецкого.

Общие ресурсы помещаются в папку shared, а локальные в папки с соответствующими названиями локалей.

3.2 Формат конфигурационного файла

Каждый конфигурационный файл состоит из одного либо нескольких блоков. Каждый блок может быть двух типов: список констант и таблица. Внутри каждого списка констант могут находиться вложенные блоки.

Каждый блок состоит из названия длиной до 32 символов – латинских букв и цифр в нижнем регистре. После названия для списка констант идет символ "=", а для таблицы фортмат для каждой ячейки помещенный в квадратные скобки. Конец названия отмечается символами "( )".

Тело блока находится в фигурных скобках. Внутри которых расположены константы, таблицы и вложенные блоки.

Например:

```
//список констант (символ = указывает на это)
build0=()
{
	type[*] = BLD;
	mesh[s] = build01_s01_c0;
	mass[f]= 4000;
	dynamic[b] = true;
	j[v]	= 1, 1, 1, 0;
	no_coll[b] = false;
	imp[v] = 30000, 0, 20000, 30000;
	chunk[s] = d_base01;
	armor_map[s] = arm_ubuild_dift;
	armor_tal[f] = 25;
	material[s] = wood;
	force_max_mip[u] = 2;


	//таблица с форматом для каждой ячейки
	col_bounds[sfuf]()
	{
		d_coll_01, 0, 0, 0.2;
		d_coll_02, 0, 0, 0.2;

	} //endof col_bounds

} //endof build0
```

Каждая константа состоит из названия (латинскими буквами в нижнем регистре не более 31 символа с учетом формата) и формата находящегося в квадратных скобках. После чего идет знак "=" и значение константы.

Допустимые форматы показаны в таблице 3.

Таблица 3

Форматы констант

Формат	Описание
u	Целое число без знака (в десятичной или 16-ти ричной системе) или цвет
i	Целое число со знаком
b	Логическая константа, принимает 2 значения: true или false
f, c	Вещественное число (для разделения целой и дробной части используется символ ".")
v	Вектор из 4-х вещественных чисел разделенных запятой
a	Вектор из 4-х целых чисел разделенных запятой
*	FCC код (целое число без знака) – буквенный код из 2-4 латинских букв в верхнем регистре

Комментарии в файле ставятся при помощи символов "//", для однострочного комментария и "/*" и "*/" для открытия и закрытия многострочного комментария. В однострочном комментарии игнорируются все символы от "//" и до конца строки, а в многострочном – помещенные между символами "/*" и "*/".

3.3 Сводная таблица команд

В таблице 4 перечислены названия команд для работы с модификациями.

Таблица 4

Команды для работы с модификациями

Работа с архивами	
mkflat	Создать архив <название и путь создаваемого архива>, <название и путь к описанию файлов которые будут добавлены в архив>
unflat	Распаковать файлы из архива <название и путь к файлу архива>, <путь к папке куда будут помещены распаковываемые файлы>
Работа с текстом и файлами настроек	
text2pd	Преобразовать файл с текстом в конфигурационный файл <название и путь к текстовому файлу>, <название и путь файла конфигурации>
pd2text	Преобразовать конфигурационный файл в текстовый файл <название и путь файла конфигурации>, <название и путь к текстовому файлу>
cfgp2pd	Преобразовать файл настроек в конфигурационный файл <название и путь к файлу настроек>, <название и путь файла конфигурации>
pd2cfgp	Преобразовать конфигурационный файл в файл настроек <название и путь файла конфигурации>, <название и путь к файлу настроек>

Работа с ресурсами	
atf2dds	Конвертировать текстуру из формата ATF в формат DDS <название и путь текстуры в формате ATF>, <название и путь к текстуре в формате DDS>
dds2atf	Конвертировать текстуру из формата DDS в формат ATF <название и путь текстуры в формате DDS>, <название и путь к текстуре в формате ATF>
wav2aaf	Конвертировать звук из формата WAV в формат AAF <название и путь звука в формате WAV>, <название и путь к звуку в формате AAF>
tex_changer	Сменить текстуры в материале объекта <название и путь геометрии объекта GO2>, <первая текстура без префикса и расширения >, <вторая текстура без префикса и расширения >
tga2am	Конвертировать текстуру из формата TGA в формат карт бронирования <название и путь текстуры в формате TGA>, <название и путь к карте бронирования>
x2go	Конвертировать геометрию из X файла в формат GO2 <название и путь геометрии в формате X>, <название и путь к геометрии в формате GO2 >
model_view	Просмотреть модель в формате GO2 <название и путь геометрии в формате GO2>

3.4 Назначение файлов

В таблице 5 приведены основные файлы игры и описано их назначение.

Таблица 5

Назначение файлов игры

Файл	Описание
Тексты	
loc_kit	Основной текст игры
loc_encycl	Тексты энциклопедии
loc_qbattle	Тексты для редактора быстрого боя
sold_fam_names	Фамилии и имена солдат
loc_redef	Назначение кнопок
Настройки	
common_res_mod	Ресурсы игры для модификации (без дублирования)
common_res	Основные ресурсы игры
ammo	Боеукладки для техники
div_units_rus(ger)	Подразделения, солдаты, техника, поддержка (без дублирования) соответствующей стороны
ger_hum_base	База немецких солдат
rus_hum_base	База советских солдат
markers_01	Надписи и знаки на технике
qbattle	Подразделения для редактора быстрого боя
season_ua_winter	Параметры сезона (зима)
sound_base	Параметры звуковых групп
techn_base	База параметров техники
ui_params	Параметры для энциклопедии и эпюр
Архивы игры	
effects	Шейдеры
gos_builds	Геометрия сооружений
gos_misc	Вспомогательные геометрические объекты
gos_objects	Геометрия объектов ландшафта
gos_techn	Геометрия техники
music	Музыкальные файлы
phys_maps	Карты бронирования и окопов
sounds	Звуки
speech_ger	Немецкая речь
speech_rus	Русская речь

tabs	Файлы настроек
tex_humans	Текстуры солдат
tex_misc	Вспомогательные текстуры меню и некоторые спрайты
tex_dummy	Текстуры небосвода и интерфейса
tex_objects	Текстуры объектов и строений
tex_techns	Текстуры техники
Локальные архивы игры	
text_loc	Текстовые файлы
textures_loc	Текстуры шрифтов

4 КАК СДЕЛАТЬ …?

4.1 10 шагов по созданию простейшего мода

Цель мода – добавить текст для энциклопедии, рассказывающий о танке Т-34.

0) **Создаем папку "users\modwork" если она еще не создана**.

1) Создаем в папке "users\modwork" папку "test_mod".

2) Создаем в папке "test_mod":

- папку "test_pack".
- текстовый файл desc.engcfg2 скопировав его из шаблона "docs\modwork\stencil\desc_example.addpack.engcfg2" и переименовав.

3) Заполняем созданный файл desc.engcfg2 (см. раздел 2):

- path - папка, в которую будет ставиться мод (латинскими буквами в нижнем регистре без пробелов);
- desc - название мода, которое будет показываться при установке (желательно латинскими буквами);
- authors - имя автора мода (желательно латинскими буквами);
- version – версия мода (100 – показывается как 1.00);
- type – тип дополнения (CAMP - кампании/полигоны, RES - обновление ресурсов, ADDN – аддон).

4) В папке "test_pack" создаем текстовый файл **test_pack.!flatlist** и пишем в него следующий текст (см. раздел 1.1):

```
i_unflat:unflat()
{
    t34_enc_text        , text      , loc_rus;
}
```

5) В папке "test_pack" создаем текстовый файл t34_enc_text.loc_rus.engcfg2 заполняя его следующим текстом (см. раздел 1.2):

```
loc_rus()
{
    txt_enci_t34_stz_m41[s]()
    {
        $t$5Т-34 обр. 1941 года выпуска, завод СТЗ.$n$t$4$n$n
        <…текст статьи….> ;
    }
}
```

6) Запускаем команду pd2text и выбираем файл "t34_enc_text.loc_rus.engcfg2", в результате должен появиться запакованный текстовый файл "t34_enc_text.loc_rus.text".

7) Запускаем команду mkflat и указываем название файла архива "test_pack.flatpack" **в папке "test_mod"**. В результате должен появиться этот файл.

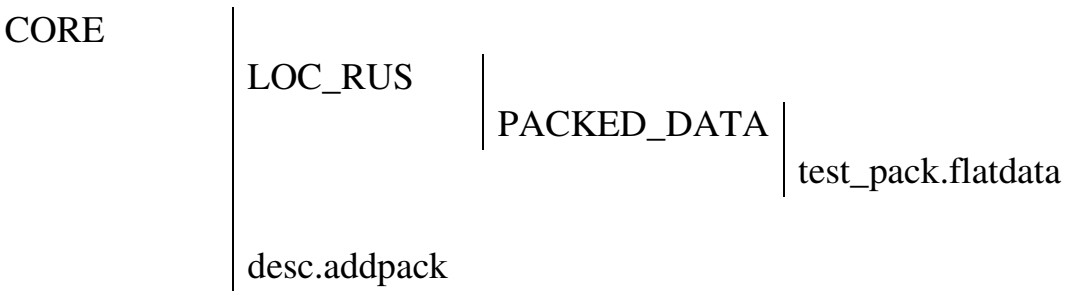
8) Запускаем команду pd2cfgp для создания описания мода и выбираем файл desc.engcfg2 в папке "test_mod". В результате появится файл desc.config, который **переименовываем в desc.addpack**.

9) Собираем мод в одном месте (см. раздел 2), в новой папке "my_addon", для этого копируем файлы:

- test_pack.flatdata в "my_addon\core\loc_rus\packed_data";
- desc.addpack в "my_addon\core";

10) Создаем файл readme.txt в папке "my_addon" в котором пишем краткую информацию о моде.

В результате должно получиться (папка "my_addon"):



Мод готов и его можно установить при помощи встроенной в игру утилиты!

4.2 Создание новых и изменение существующих подразделений

Описание подразделений РККА хранится в файле div_units_rus.loc_def.config, для Вермахта в файле div_units_ger.loc_def.config, а для современных армий (после ВМВ) в файле div_units_misc.loc_def.config. Эти файлы находятся в архиве tabs.flatpack. Для его извлечения необходимо:

1) Распаковать архив tabs.flatpack (из патча) командой unflat.

starter.exe root\programs\unflat.progpack,
data\k43t\dev_updates\shared\packed_data\tabs.flatdata, users\modwork\tabs_uf

2) Сконвертировать файл div_units_rus.loc_def.config командой cfgp2pd.

starter.exe root\programs\cfgp2pd.progpack, users\modwork\tabs_uf\div_units_rus.loc_def.config,

Скопировать распакованный файл настроек "div_units_???.loc_def.engcfg2" для дальнейшей работы в другую папку. Файлы подразделений работают по накопительной системе, т.е. каждый установленный патч или дополнение, в котором есть файл с таким названием, добавляет описание единиц либо подразделения в общий список. В случае дублирования подразделений используется первое найденное *в порядке установки патчей* и дополнений.

В разделе **units()** находится описание отдельных единиц техники и солдат, с префиксом "rkkau_" – техника и солдаты СССР, а с префиксом "weru_" – техника и солдаты Германии.

В разделе **squads()** находится описание отделений и техники вместе с расчетами, с префиксом "rkka_" – подразделения СССР, а с префиксом "wer_" – подразделения Германии.

Например, мы хотим **создать новое отделение за СССР с 5 солдатами с винтовками и одним сержантом и изменить экипаж танка Т-34 до 3-х человек.**

1) Удаляем содержимое блока "units()", оставляя только фигурные скобки и название блока, т.к. новых единиц техники либо солдат мы добавлять не будем.

2) Удаляем содержимое блока "supports()", аналогично предыдущему.

3) Удаляем все строки из блока "squads()" кроме строк:

rkka_squad_inf_43a, sq_inf, txt_ce_rkka_squad_inf_43a, ….;
rkka_crew_t34, sq_crew, txt_ce_rkka_crew_t34, …..;

4) Переименовываем в первой строке:

-"rkka_squad_inf_43a" в "rkka_squad_inf_43b",
- "txt_ce_rkka_squad_inf_43a" в "txt_ce_rkka_squad_inf_43b".

5) Ищем в первой строке "rkkau_inf_sergant, 1" – с этого места начинается список единиц и техники, которая входит в подразделение (здесь же можно указывать объявленные выше подразделения). После каждого вхождения стоит количество единиц/техники (в данном случае 1 шт.). Оставляем сержанта, и меняем следующую запись "rkkau_inf_rifle, 6" на "rkkau_inf_rifle, 5".

Удаляем оставшиеся записи "rkkau_inf_arifle, 1, rkkau_inf_mgun, 1," заменив их на " , 0, , 0, ". Таким образом, мы получаем отделение из 6-ти человек с названием "rkka_squad_inf_43b".

6) Во второй строке "rkka_crew_t34" ищем " rkkau_tank_agun, 2," и заменяем на " rkkau_tank_agun, 1,". Получив экипаж танка Т-34 из 3-х человек. При этом название подразделения остается прежним и изменяет значение оригинального подразделения.

Дополнительно необходимо создать текстовый файл с таблицей "txt_ce_rkka_squad_inf_43b" (см. раздел 1.2) содержащей две записи разделенных ";": "Отделение" и " Стрелковое обр. нач. 1943г. №2". Которые будут отображаться в списке и интерфейсе. Этот файл должен быть включег в состав мода, таким же образом, как и созданный файл с описанием подразделений.

Процесс создания мода подробно описан в разделе 4.1. **Созданый мод нужно ставить выше по списку, чем обновления от разработчиков.**

Использовать новое подразделение можно:

- добавив его в резервы подразделений для быстрого боя "qbattle.loc_def.config" (tabs.flatpack) например, в блок "p_ussr_rd_04_550()" либо в состав активного взвода, например, в блок "ussr_rd_rifles", заменив там одно из подразделений;

- добавив его в состав подразделений в одной из операций, для этого нужно сконвертировать файл:
"data\k43t\dev_updates\shared\camps\..\<операция>\<операция>.campack",
командой cfgp2pd, и добавить подразделение в резерв (блок "reserves") или в один из активных взводов (блок "act_platoons"), а затем запаковать файл командой pd2cfgp.

Все измененные файлы необходимо добавить в состав мода. Файлы описания операции запаковывать в архив не нужно, необходимо только задать правильный путь!