

SpellForce 3: Fallen God

Quick Start Guide for Fallen God Custom Campaigns

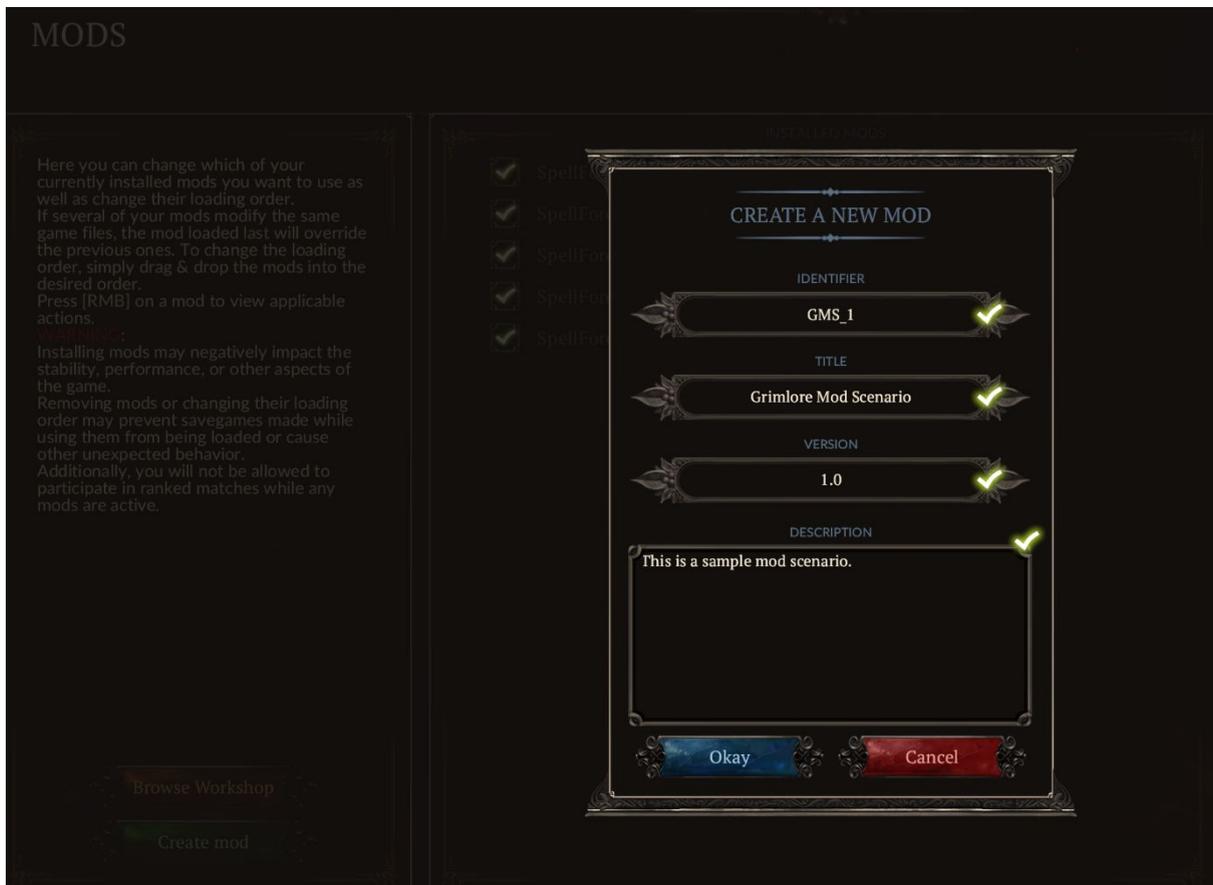
This tutorial goes over all the steps required to get a complete custom campaign up and running. While we do not go over the details of how to create gameplay logic, it covers every topic required to start your own custom campaign map(s)!

Creating the Base Mod	2
Setting up a new Articy project	3
Adding a custom scenario definition	5
Adding custom map logic	6
Adjusting map configuration	7
Exporting the map description	8
Setting up the Script and Scenario	9
Adding the script logic	9
Text and Localization	11
The Level Editor	12
Asset Tool and Data Editor	13
MakeRes	16
Testing and Publishing your mod	17

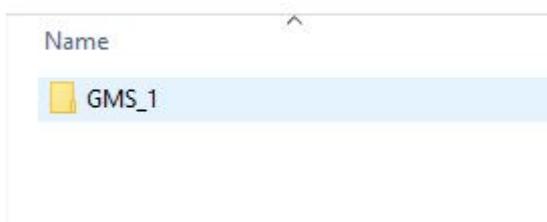
Creating the Base Mod

To start, open SpellForce 3: Fallen God.

Find the “mods” tab and create a new mod. Make sure to pick a unique Identifier that you can easily recognize and copy/paste.



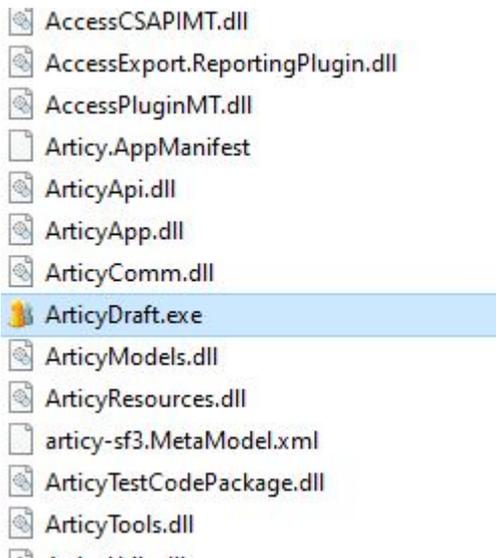
Pressing “okay” will create a folder in your `_game_root_folder/data_mods/`, and in there you place all your files for your mod.



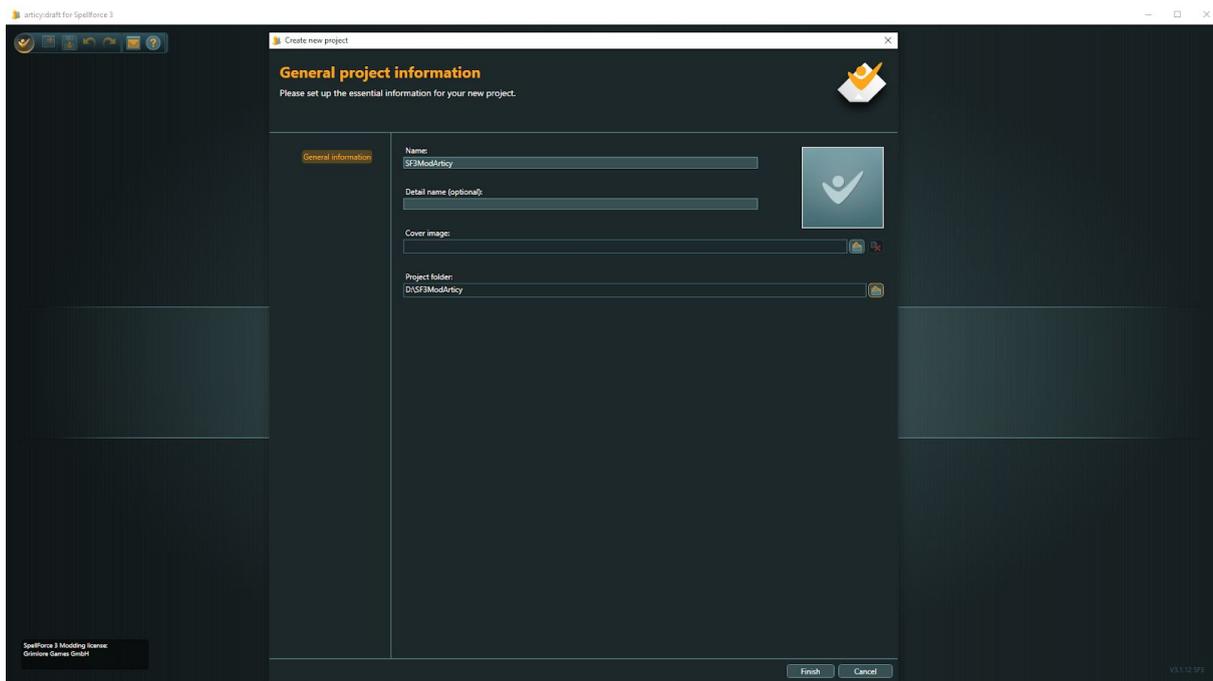
For this Quick Start Guide, we will copy the Scenario Map “Old-Haayalash” and make it editable for us.

Setting up a new Articy project

In your_game_root_folder/editor/articy, find ArticyDraft.exe and open it.



Click on “Create New Project”



Articy will ask for a “root” folder. You will need to select your SpellForce 3: Fallen God installation folder.

Once Articy opens, you will find all kinds of content that we placed there for you. It is read-only and enables you to use characters, buildings and units from all SpellForce 3 campaigns.

The “Flow” contains all Dialogues and Quests, in this case only for our Scenario Map.

“Entities” contains all types of objects that can be present in the world: Buildings, Units, Characters, Items. We have shipped everything available for maximum flexibility!

“Locations” are where you can edit your map configurations.

“Documents”, “Template Design” and “Assets” refer to various technical implementations and additional assets you can use in Articy itself.

For our testing purposes, open up “Locations”.



You will find the map data in Main/Gamescom_2016/ where we stored our Map Data for the Scenario “Old-Haayalash”. If you want more details about the different data entries, simply click on one of them and press F8. F8 will be your best friend in Articy, as it opens up the Properties-Window for the data entry.

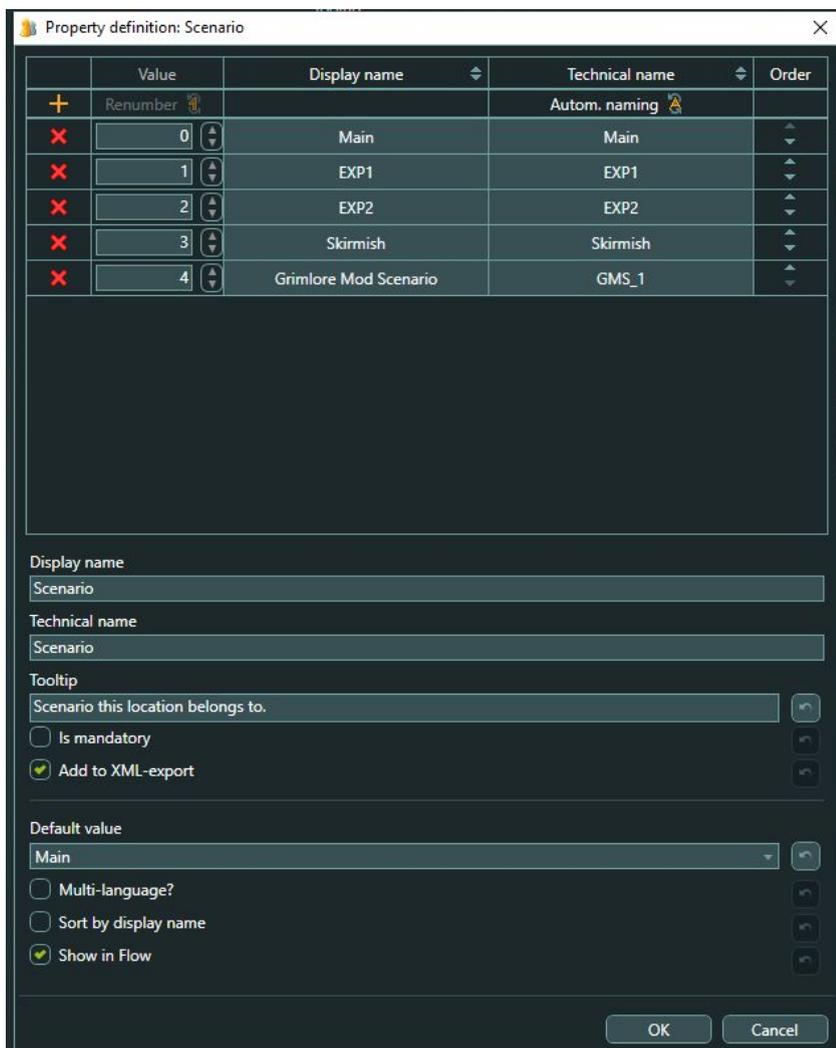
Before we continue, we need to create a proper playable technical Scenario type.

Adding a custom scenario definition

To do this, find the “Scenario” Drop-Down List under “Property Definitions”



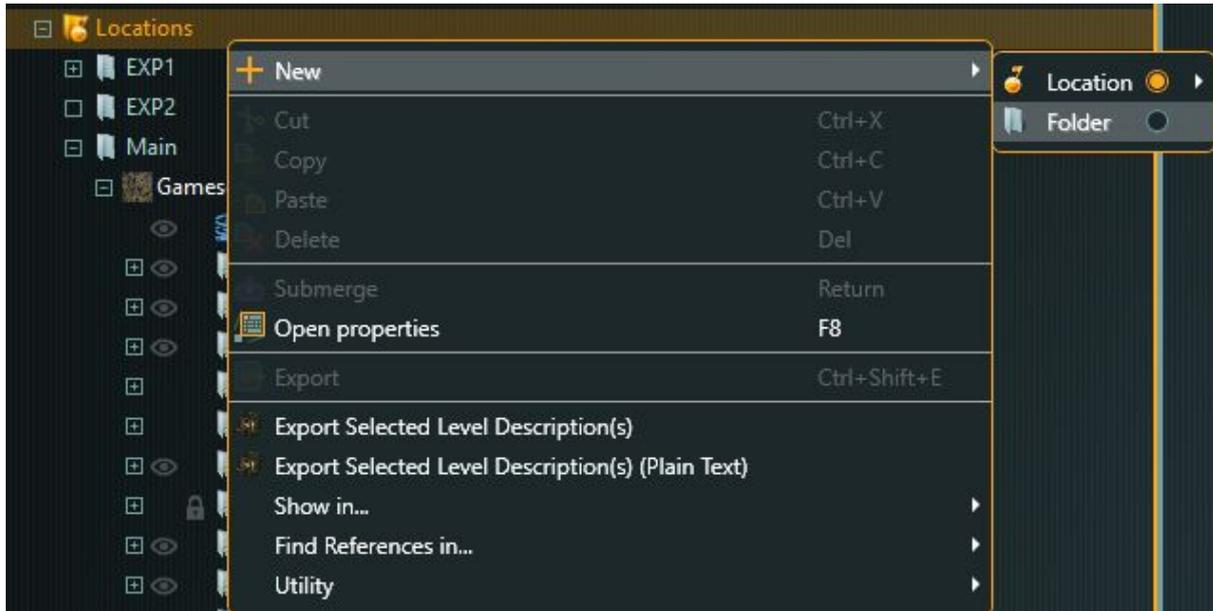
Hit “Edit”, then enter your custom scenario and press “OK”.



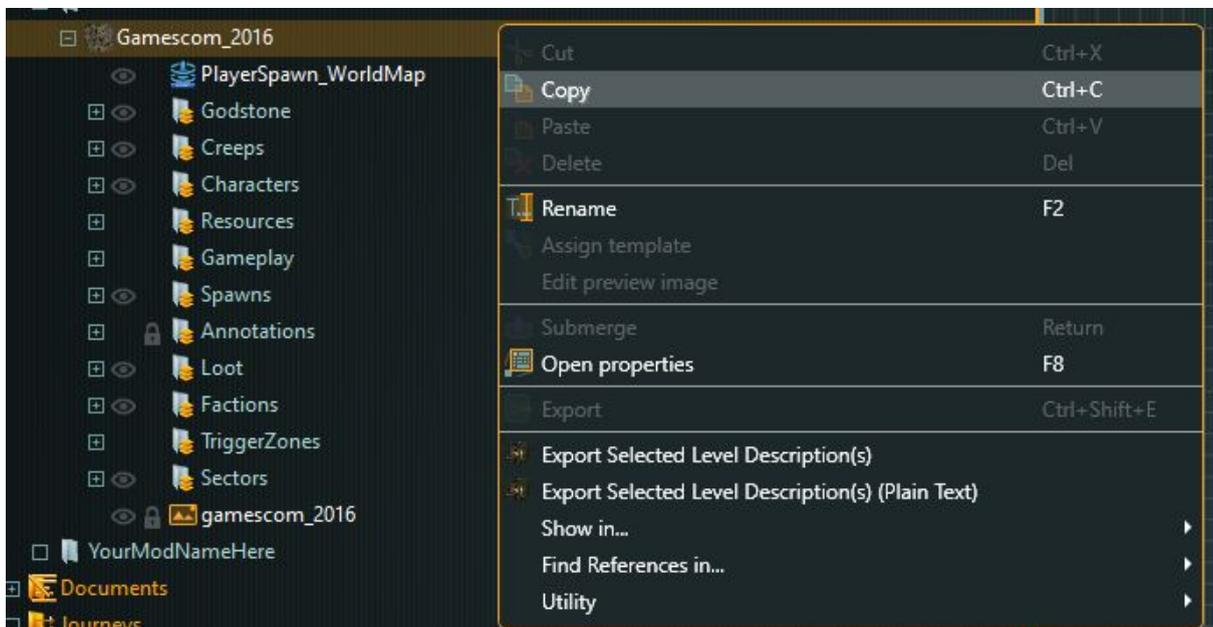
Adding custom map logic

Back in “Locations”, we now create our own folder there where we put all our data.

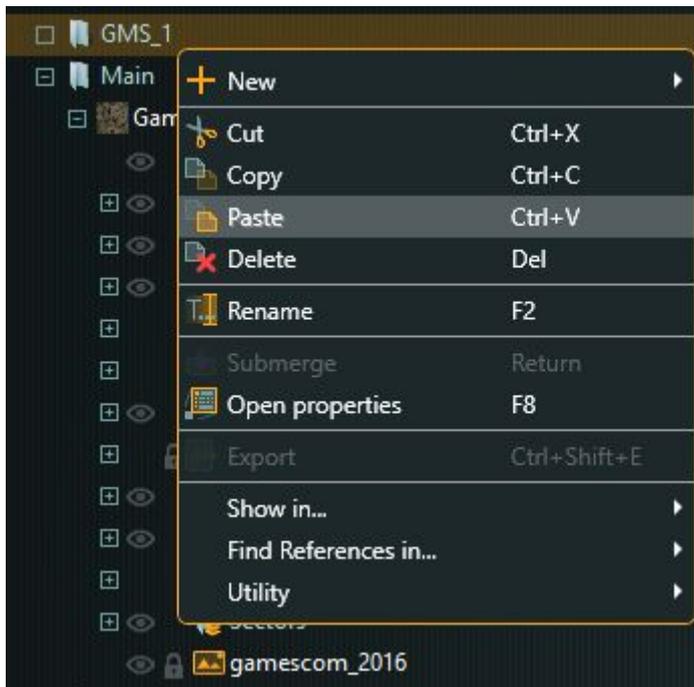
Create a new folder for our map under “Locations”



Normally, you would add your own logic here. For testing purposes, we use existing logic. Copy the whole map of the Old-Haayalash Scenario, the technical name being “Gamescom_2016”



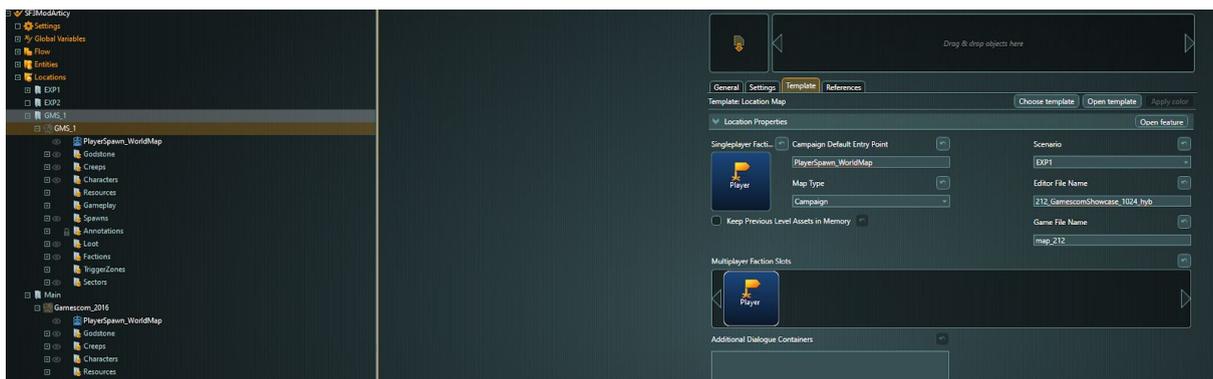
Then paste it into your new folder.



Adjusting map configuration

You can now start to make adjustments to this map. For demonstration purposes, we recommend only changing the following:

Rename “Gamescom_2016” to a more fitting name (we recommend using technical names, as real names are bound to change) and when done, click on the entry and press F8 to open the properties. In the properties, click on “Template”. Your screen should now look like this:



You can see various options for this map. We have a defined player faction, we have a faction for joining multiplayer players and we also have various properties.

Because this is a custom scenario where we want to define our own rules, we will change the “Scenario” type from “EXP1” to our “Grimlore Mod Scenario” (or whatever name you picked for your Scenario).

The “Editor File Name” is also incorrect, but this is just a personal reference for yourself. Make sure to change it to a name you can recognize later.

The “Game File Name” needs to be in the format of “map_XXX”, where XXX can be any id between 900 and 65000. We recommend trying to find IDs for your map that will not interfere with other mods.

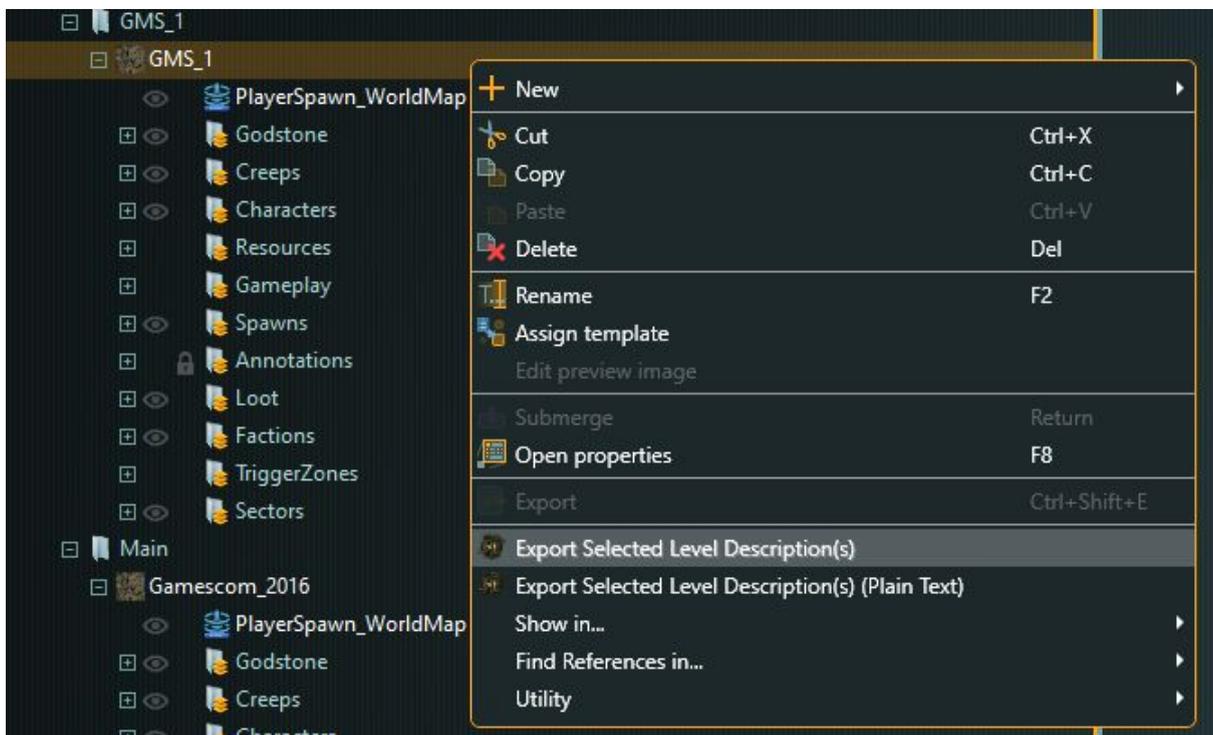
For testing purposes, we will pick the id “924”, so our Game File Name is going to be map_924.

Also, for this map we already have Dialogue and Quests set up in our “Flow”. The id of that map is 212, and we want to use all of that. Because of this, we add “map_212” in “Additional Dialogue Containers”.



Exporting the map description

This is all the work that is needed to have this Articy Map Description working for our purposes. To get this working in your mod now, simply export the map.



If everything goes well, you will find a file with the chosen id in your mod folder. For our case, this is data_mods/GMS_1/configs/levels/map_924.content

If you have this file, continue.

Setting up the Script and Scenario

Adding the script logic

Normally this would be the start of your scripting adventure.

However, because this script has already been written, we can simply go into /editor/SingleplayerTemplateMap/configs/levels and copy the folder “map_924”. This folder contains another subfolder called “Scripts” and there we find our actual scripts. One for handling game logic, another for handling the weather.

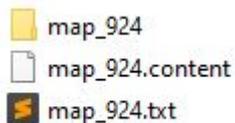
Find your_mod_root_folder/configs/levels/ and copy the “map_924” folder in there.

For custom script logic, please refer to our scripting wiki:

http://sf3scripting.spellforce.com/index.php?title=Main_Page

Additionally, copy “map_924.txt” as well and paste it into the same folder. This file serves as a more general description of the map and can determine if a map should use different logic compared to others.

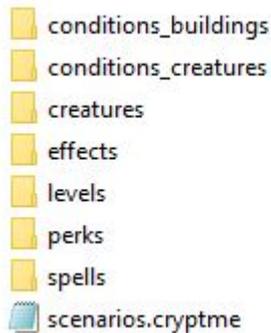
Having done that, your configs/levels/ folder should look like this



There is no need for changes here, as our .content file is already up to date and the scripts are just our standard scripts.

Adding scenario configuration

In the “SingleplayerTemplateMap/configs” folder, you can find a file called “scenarios.cryptme”; copy this into your mod’s /configs/ folder.



In this file, you will find a description of your custom scenario. To make sure that the Articy Map Description knows about this scenario, use the technical name that you have referenced earlier.

In our case, this is “GMS_1”. Also, use a DisplayNameIndex between 1 and 100 for this map. We have picked the id “99” for this level, and as it is the only level we use, we are setting it as our starting level, too.

```
// GMS_1
Scenario "GMS_1"
{
    // First map and setting of this scenario
    StartingLevel 924 0
```

In your own mod, you could set a different level as the starting level and then allow the player to travel to this one instead. Just like in a regular campaign!

You can set up various other things in here, such as available factions to play, if Co-Op play should be allowed and so on.

Text and Localization

To make sure that our maps and mods have proper descriptions, and any new content we add also can be read properly, we need to add some xml files.

In the “SingleplayerMapTemplate”-folder find the folder “texts”. Copy the 4 xml-files there into your mod’s “text” folder.

To categorize different texts, we use a system “Identifiers” and [IDs] and depending on what type of object you want to name, you will have to use a different file. Naming the spell with the ID of 12345 would require “Spell[12345” for a name and “SpellDesc[12345” for a description in the “names.xmls” file.

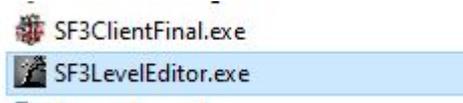
If you are curious about all the identifiers and IDs, you can simply open the debug client and start the following command (Use ^ to open the console)

```
[INPUT]clientsystem dumptimeindices
[INFO]Unused name index ranges below:
[INFO]Folder: Main
[INFO]- Scenario: 8 - 100
[INFO]- GlossaryCategory: 11 - 20
[INFO]- OptionsMenuHint: 78 - 1000
[INFO]- Notification: 111 - 1000
[INFO]- TutorialName: 22, 24 - 26, 30 - 32, 67 - 99, 120 - 149, 165 - 199, 207 - 209, 211, 214 - 300, 317 - 319, 335 - 1000
[INFO]- CtrRaceName: 15 - 100
[INFO]- CtrAttributeName: 8 - 20
[INFO]Folder: Names
[INFO]- Item: 717, 736 - 799, 848 - 849, 854 - 856, 861 - 999, 1009 - 1499, 1527 - 1599, 1612 - 1699, 1711 - 1799, 1878 - 2499, 2524 - 2599, 2611 - 2699, 2711 - 2799, 287
[INFO]- Creature: 12 - 14, 17, 27 - 29, 31, 43 - 476, 478 - 1595, 1597 - 2433, 2436 - 2495, 2498 - 2500, 2502 - 2503, 2505 - 2519, 2521 - 2545, 2551 - 2563, 2565 - 2596, 26
[INFO]- CreatureAction: 25 - 4665, 4667 - 65535
[INFO]- Building: 30, 60, 92 - 149, 165 - 999, 1016 - 4999, 5033 - 5099, 5125 - 5199, 5234 - 5299, 5326 - 5399, 5420 - 23999, 24020 - 26999, 27001 - 30000, 30053 - 39499
[INFO]- BuildingAction: 23 - 199, 203 - 209, 211 - 65535
[INFO]- Technology: 0 - 45 - 49, 52 - 99, 107 - 119, 126 - 139, 146 - 199, 203 - 4999, 5015 - 5099, 5116 - 5199, 5217 - 5299, 5305, 5314, 5319 - 5329, 5333 - 5399, 5414 - 59
[INFO]- Resources: 28 - 65535
[INFO]- Spell: 0 - 13, 15 - 999, 1002, 1005, 1008 - 1022, 1025 - 1049, 1052 - 2999, 3001 - 3004, 3006 - 3009, 3011 - 3014, 3016 - 3019, 3021 - 3024, 3026 - 3029, 3031 - 30
[INFO]- SpellEffectDesc: 1 - 7525, 7527 - 7815, 7817 - 20649, 20651 - 20654, 20656 - 20659, 20661 - 20729, 20731 - 20734, 20736 - 20739, 20741 - 20919, 20921 - 20924
[INFO]- Preset: 26 - 65535
[INFO]- AbilityTree: 24 - 899, 903 - 909, 920 - 949, 971 - 999, 1002 - 1899, 1915 - 1919, 1924 - 65535
[INFO]- Condition: 13 - 14, 24, 30 - 49, 51 - 99, 141, 143 - 173, 176, 183 - 1019, 1023 - 2135, 2137 - 2139, 2141, 2143 - 2160, 2163 - 2209, 2212 - 2213, 2215 - 2240, 2242 -
[INFO]- ConditionEffectDesc: 0 - 25023, 25026 - 31784, 31786 - 39100, 39102 - 39150, 39157 - 39199, 39201 - 39239, 39241, 39243 - 39299, 39302 - 42059, 42062 - 420
[INFO]- BuildingCondition: 7 - 25955, 25957 - 44994, 44996 - 65535
[INFO]- BuildingConditionEffectDesc: 0 - 65535
[INFO]- Perk: 1 - 9, 11 - 19, 21 - 29, 31 - 39, 41 - 49, 51 - 59, 61 - 69, 71 - 79, 81 - 84, 91 - 199, 201 - 209, 211 - 219, 221 - 229, 231 - 239, 241 - 249, 251 - 259, 261 - 269, 271
[INFO]- PerkEffectDesc: 0 - 25040, 25044 - 25137, 25141 - 25613, 25616 - 25729, 25733 - 25749, 25753 - 26300, 26304 - 26359, 26363 - 31999, 32001 - 32009, 32011 - 3
[INFO]- Area: 81 - 65535
[INFO]- Timer: 2 - 65535
[INFO]- Blueprint: 0 - 65535
[INFO]- Race: 0, 10 - 65535
[INFO]- EntityFlag: 14 - 19, 23 - 65535
[INFO]- Achievement: 0, 55 - 56, 58, 86 - 100, 136 - 65535
[INFO]- AchievementStat: 0, 59 - 65535
[INFO]- AchievementReward: 0, 6 - 65535
[INFO]- AchievementCategory: 0, 7 - 65535
[INFO]- ArenaThemeName: 0, 19 - 65535
[INFO]Folder: Inspection
[INFO]- Description: 0 - 46, 60 - 999, 1030 - 1999, 2002 - 2999, 3010 - 3029, 3035 - 3999, 4007 - 4999, 5005 - 5999, 6007 - 6999, 7006 - 7999, 8006 - 8999, 9005 - 9999, 10
```

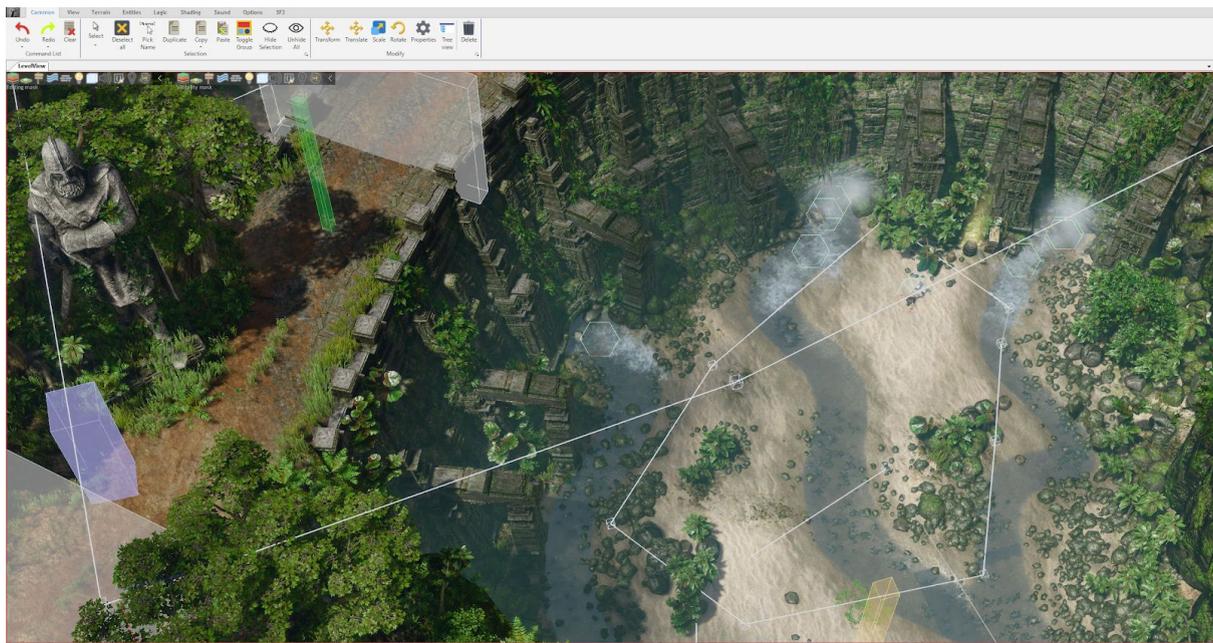
The Level Editor

We have our map description set up, as well as the scripts and technical connections for text. The last thing that is missing is the actual level file!

To get you started, we have shipped the complete level files for “Old-Haayalash”. Open the “SF3LevelEditor.exe” in your game’s root folder.



Click on the Grimlore Games Logo on the top left, select “Load” and find the editor\SingleplayerTemplateMap\levels folder. In there, you should find the file “GScenarioMap.mogler” – this is our map file.



You will find the map in its technical overview, with all logic and beauty behind it. Explaining this Editor is worthy of a whole extra tutorial – we just want to get to the basics here, so do not edit anything at this point. If you are interested in using the Level Editor, refer to “Grimlore World Editor Documentation.pdf” in the /editor/ folder.

Instead, click on the Logo again, but this time select “Export”. Find your game’s root folder and export the map in the /levels/ (not /configs/levels!) folder of your mod. In our case, this is: data_mods\GMS_1\levels.

Be sure to name the file “map_XXX”, so for us “map_924”. Then wait for the export. This can take a long time, so if you want to skip this waiting time, simply copy the pre-exported file “map_924.lv” and rename it according to your chosen ID.

Everytime you change something in the level editor, a new export is needed.

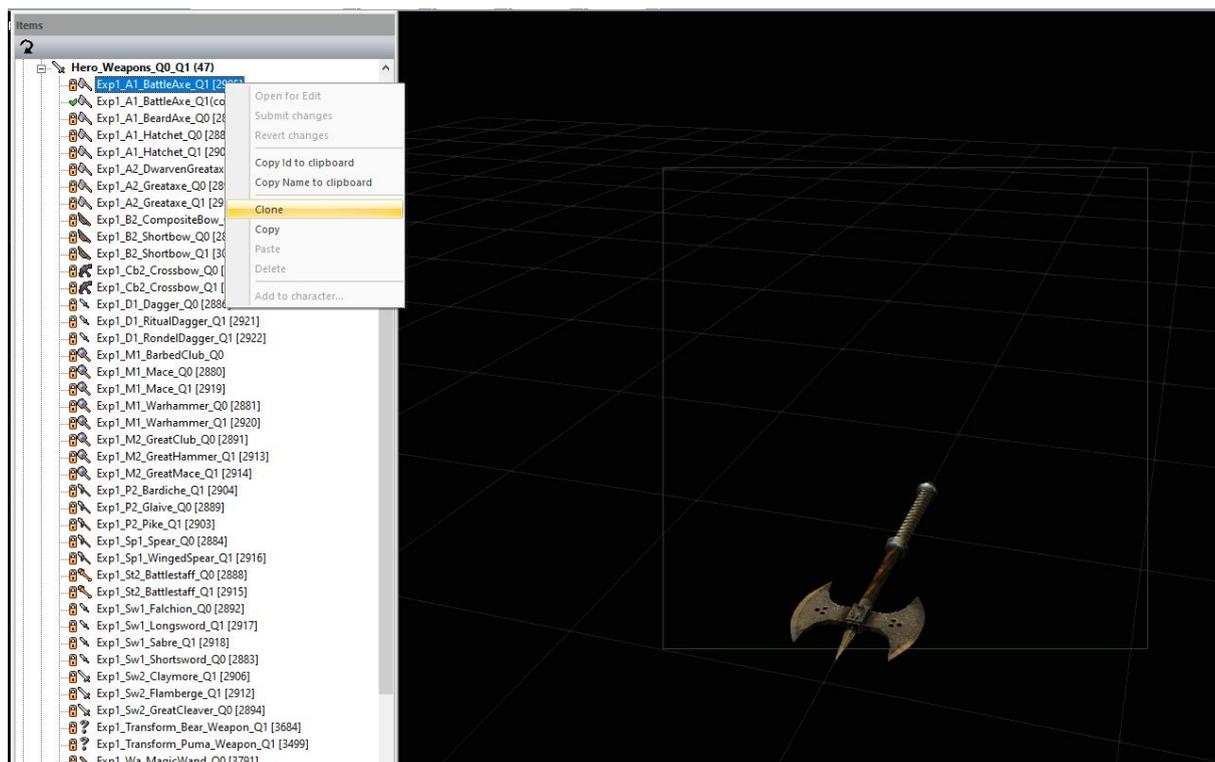
Asset Tool and Data Editor

There are 2 tools left to cover. The Asset Tool serves as a reference viewer for everything visual. Here you can take a look at models, effects and other entities that can be used in the Data Editor.

The Data Editor combines these visual entities with gameplay logic and prepares them to be used in the actual game.

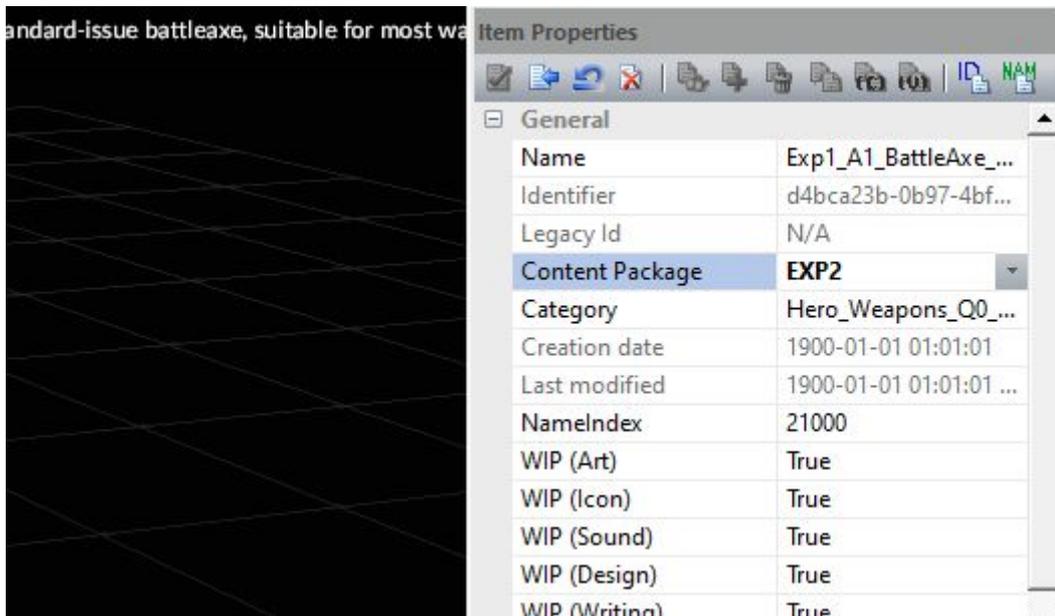
When opening the Data Editor, you will see a lot of existing data that we have used to create the various SpellForce 3 games. While you can not modify these, you are free to add your own data entities: new items, buildings or characters!

For the tutorial map, we will not add any new content. But the steps to adding new content is actually really easy. Say we wanted to add a new Dwarven Weapon, the “Demon Slayer”.



We can either

- 1) Find a weapon that we like and clone it
- 2) Create a new entity and modify it's Item Properties to match those of other weapons, to your liking.

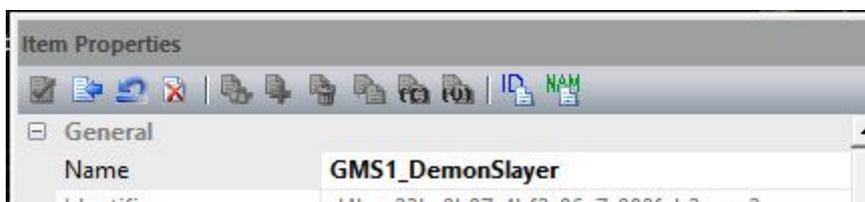


The “General” section requires a few changes

- Name: Set to a unique name that is uniquely unique to your item/building/character!
- Identifier: Used to reference the item by ID, either in the Data Editor or in scripts
- Content Package: Needs to be set to your Mod Name
- Category: Should be set to something that fits the new entity
- NameIndex: Will be used to give a name and description to the entity ingame. So make sure to match it with the one specified in the appropriate text file (See section “Text and Localization”)

Also, in Articy, you will need to add an entity that has the same technical **name** that you used in the Data Editor.

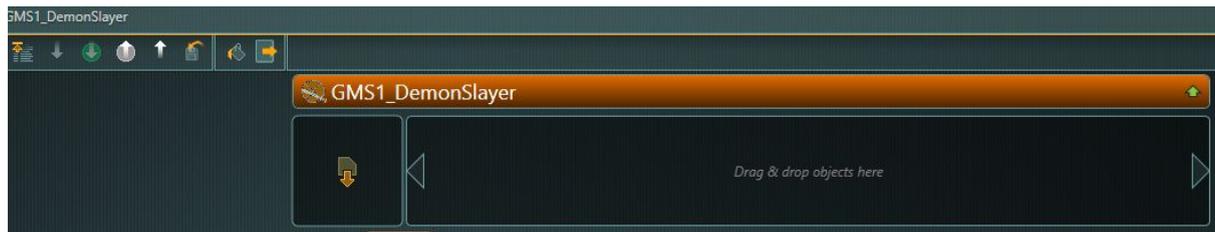
For our DemonSlayer, we picked “GMS1_DemonSlayer” as the technical name in the Data Editor.



Now, in Articy, we create a new Item Entity within “Entities” (Sub folders are up to you)



And we match the name to that in the Data Editor.



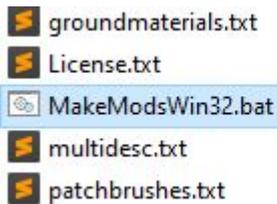
Now we can use this Entity in Articy to fill loot containers, quest rewards or other things. We can also use the technical name instead of the identifier in scripts to give it to the player.

The general process is the same for buildings and characters, but the articy entries are different and the data you can configure in the Data Editor also depends on the type of entity. We recommend taking a look at how our entities are set up to get a better idea on how they work.

MakeRes

All your map data is now in the data folder. This is good, but the game only recognizes files in “bin_X” folders. To avoid copying everything manually, we created a helpful batch file.

Find “MakeModsWin32.bat” in your_game_root_folder/editor/.



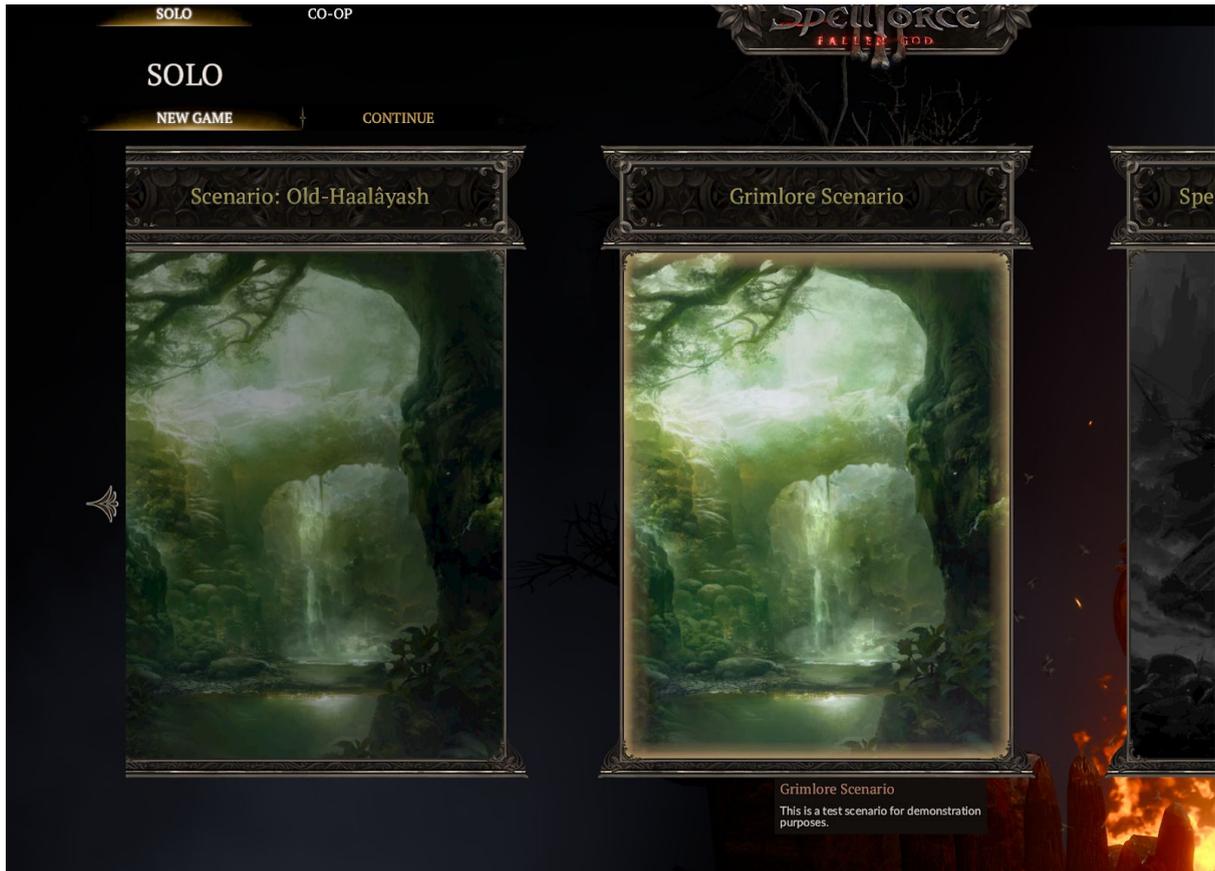
Simply execute it and you will see a bunch of updated files listed. If no errors show up, close it. If certain errors show up, go back through this tutorial and repeat the necessary steps!

```
[Win32] "bin_mods_win32\gms_1\configs\scenarios.crypt" has been encrypted successfully.  
[Win32] "bin_mods_win32\gms_1\configs\levels\map_924\Scripts\Weather.as" has been updated.  
[Win32] "bin_mods_win32\gms_1\configs\levels\map_924.txt" has been updated.  
[Win32] "bin_mods_win32\gms_1\configs\levels\map_924\Scripts\Level.as" has been updated.
```

Do not forget to do this step everytime you change something – if it’s only in the data-folder, the game will not recognize it!

Testing and Publishing your mod

If you have followed every step of this tutorial, you will find your new Campaign Mod in the game. Since we have not made any adjustments besides the name, the symbol will look similar to Old-Haayalash, so do not be surprised:



Feel free to play through this map if you have not done this already, it gives a great overview of how the campaign logic works.

If you are done with editing your map, you can go into the “Mods”-tab again. Right-click your map and hit “Publish”. This will create .pak files you can distribute to your friends or directly via the Steam Workshop!

That is it! ... at least for now. The real magic starts in crafting beautiful maps, connecting them with your custom logic and writing compelling dialogue and quests.

We absolutely cannot wait to see what you have in store for SpellForce 3!

Have questions? Head on over to our Discord: <http://discord.gg/spellforce> or visit the steam SpellForce 3 Steam Forums!