**Lesson 3 The first steps to customizing your own station-GRF**

Well done mate! :W  
Good thing indeed is that you now have a testbed to check out your graphics and see if you like them, how they turn out ingame.  
I would recommend to split the browns and the blacks and make two separate sets of graphicsfiles from them. These we can then use to expand the code.  
  
But before expanding the code I will try to explain some of the basic elements of the code we have so far. That will make it hopefully easier for you to see what needs to be changed when expanding the code or if you want to make an update for public release or even an entirely new GRF.  
Many of these things you will recognise from NML-coding, its just done differently ;-)  
  
The most important thing in NFO-coding I would like you to know about is hexadecimal counting.  
In normal life we count in decimals, 0 to 9, 10 to 19 etc etc.   
Hexadecimal adds 6 extra "numbers":  
Decimal: 00 01 02 03 04 05 06 07 08 09 (10 11 ......)  
Hexadecimal: 00 01 02 03 04 05 06 07 08 09 0A 0B 0C 0D 0E 0F (10 11 12 13 14 15 16 17 18 19 1A 1B ..... 99 9A 9B 9B 9C 9D 9E 9F A0 A1 ..... )  
So everything upwards from 09 is a different (larger) number, 0A is actually 10, 10 is actually 16.  
This means you can get more "real" numbers from just 2 bits, eventually leading up to FF, which is 255 (and that is a magic number you probably know about ;-)For example its the max number of ID's in a single GRF)  
As I said, Hexadecimals are very important in NFO-coding, you will encounter them a lot, so its good to get to know about them early on ;-)  
A very handy website is this one:  
<http://www.statman.info/conversions/hexadecimal.html>  
A simple hexadecimal to decimal and viceversa converter, for when you loose track of what number you are on ;-)  
  
**Ok back to the basics....**  
First of all, the name of the .nfo

**Code:**

auzstations\_1\_v001.nfo

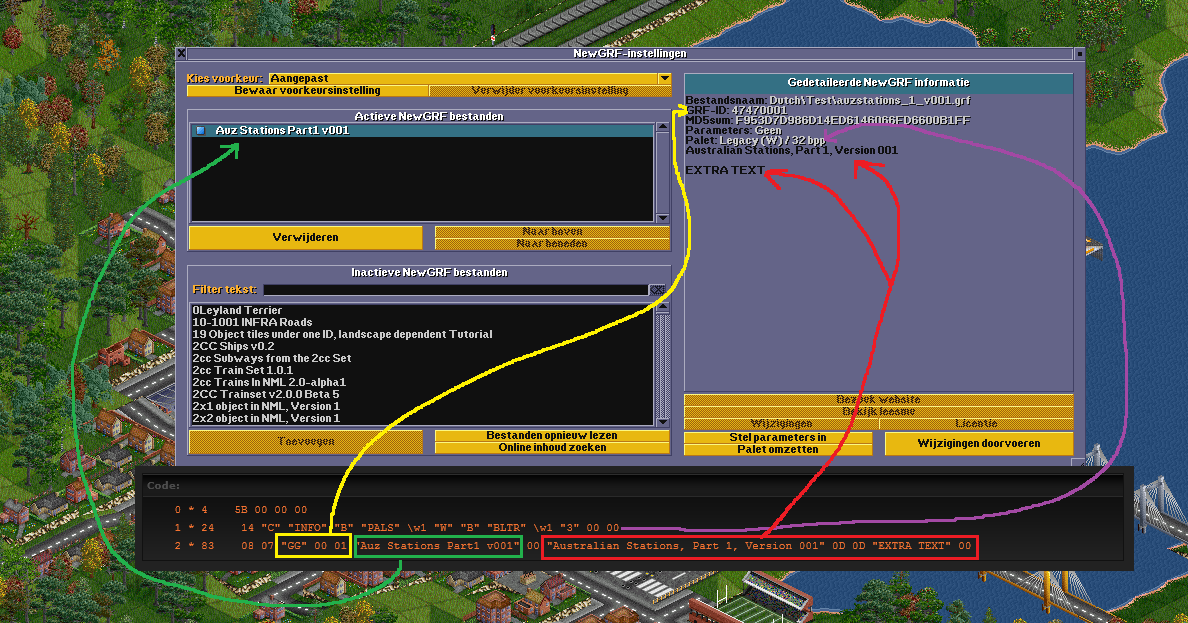
Choosing a clear and updateable name from the beginning is very important (just like with NML-coding)  
auzstations speaks for itself  
\_1\_ means part-1 of the auzstations family (trust me, when you get into stations you just keep adding and adding, like I did with the Dutch Station Addition set, I am almost ready to start with part-3 of that ;-))  
v001 That is the current version number, when you are ready for a publicly released update, save the file then with a higher versionnumber. Basically the same as you do with your other GRFs. Nothing really new here ;-)  
  
Now, inside the .nfo file  
All GRFs in NFO that I coded have this kind of "header"

**Code:**

    0 \* 4    5B 00 00 00  
    1 \* 24    14 "C" "INFO" "B" "PALS" \w1 "W" "B" "BLTR" \w1 "3" 00 00  
    2 \* 83    08 07 "GG" 00 01 "Auz Stations Part1 v001" 00 "Australian Stations, Part 1, Version 001" 0D 0D "EXTRA TEXT" 00

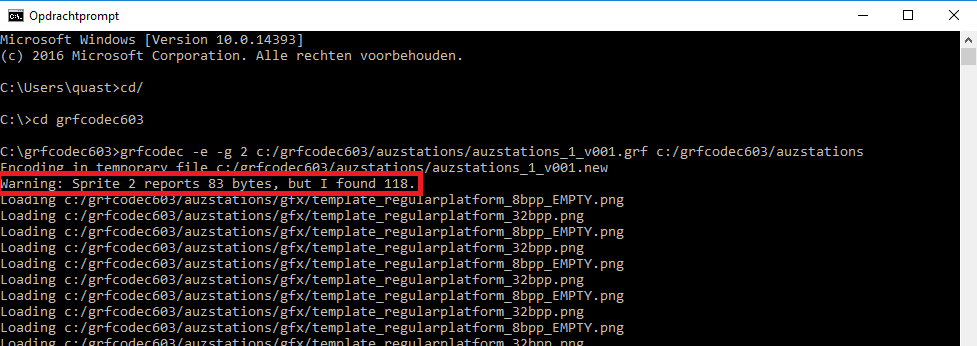
line 0 states the total number of lines of code in the .nfo in hexadecimal, we have a total of 91 lines right now, that is 5B in hexadecimal.   
As you can see, there are three 00's, meaning we are not limited to just 2 bits and thus can go further than FF (255 lines).  
Adding extra lines of code means that this number will go up, but you dont have to do that manually, we will use the handy tool nforenum.exe for that later on.  
  
line 1 Is in this case basic graphics information, it says that the 8bpp file is in Windows palette and that there are also 32bpp graphics. This you dont have to change, leave it as is.  
EDIT made on July 18th 2017:  
Something does have to change ;-)  
the "W" has to be "D"   
as it turns out I am using the Dos-palette for the 8bpp graphics-templates, not the Windows-one :oops:This is important regarding using an 8bpp animated color mask, in combination with the 32bpp graphics   
  
line 2 states the GRF-ID "GG" 00 01 , the GRF-name WITH VERSIONNUMBER "Auz Stations Part1 v001" as seen in the grf-list ingame and finally the GRF-description "Australian Stations, Part 1, Version 001" 0D 0D "EXTRA TEXT"  
So basically, the same information you give in a NML-header.  
For every new publically released update you need to change the versionnumber, just like with the .nfo name.   
The "EXTRA TEXT" you can alter to what you like.  
If you want to make a completely new GRF, you also need to change the GRF-ID and general names ofcourse. Basically the same like you do for your object-sets.  
So ingame you will see this information like this:

**Attachment:**

  
Zoom in (real dimensions: 1188 x 623)  
Example246.png [ 248.5 KiB | Viewed 62 times ]

For now I would like to give you a little assignment 8-)  
- Change the EXTRA TEXT, to a text that you would like. Maybe something like: Stations as found Down Under, Made by GarryG. Whatever you like ;-)  
- Save the .nfo and try to compile it to a GRF  
  
- It will probably compile to a GRF, but at the beginning of the compiling you will see a warningmessage like this:

**Attachment:**

  
Zoom in (real dimensions: 977 x 346)  
Example247.png [ 29.99 KiB | Viewed 62 times ]

- This now shows you what the number means next to the codeline number. Its the number of bytes in a line of code. So changing stuff in a line of code may probably mean that you need to change that number. Handy that grfcodec.exe gives this information, so you dont have to count it yourself ;-)and thus you can change the number to the correct one.  
However, you will probably have to do this often (for example when adding/changing the names of the stationtiles for the purchasemenu) and when you have a lot of code it is a bit of a pain to scroll back everytime in the DOS-prompt ;-)  
  
- So now we come to the real point of this assignment, using the handy tool nforenum.exe :twisted:  
  
- Copy the .nfo with the changed information to the folder that has nforenum.exe in it  
- Drag and drop the .nfo onto the nforenum.exe  
- It will now automatically renumber what needs to be renumbered (you will probably see a quick DOS-prompt pop up and dissappear again)  
- Now, cut (or copy, but cutting is handier so you dont end up with too many copies of the .nfo in various folders) and paste the .nfo back to its original folder and let it replace the old one.  
- Recompile the GRF again and you will see that there is no more warning. :mrgreen:  
  
You are going to do this often, as nforenum not only renumbers the number of bytes in a line of code, the total numbers of codelines as stated in codeline-0, but also the codelinenumbers at the beginning of a line of code!! :bowing::bowing:  
Meaning that you can later on simply copypaste parts of code, without having to worry about changing the codelinenumbers manually!! And this is really really handy!!  
  
I will let you digest this information first, let me know if you can get nforenum to do its work and then I will go on to the real magic, the pieces of code that actually gets you stations ingame. 8-)

Last edited by [**Quast65**](http://www.simuscape.net/simutalk/memberlist.php?mode=viewprofile&u=102) on 19 Jul 2017, 03:58, edited 2 times in total.