MATERIAL AND TEXTURE ANIMATION WITH NIFSKOPE:

PART 2: MATERIAL ANIMATIONS

A tutorial from Pixelhate. 2014.

Contact Information: Pm Pixelhate at http://forums.nexusmods.com/index.php?showuser=1205226

This guide is the result of compiling information on the use of NifSkope by reading tutorials (some quite outdated) and Wikis, tracking help posts on forums or by trying to figure things out by myself. This work was possible thanks to too many people to name them all. Some of their words where just copy/pasted here, as things where explained better than I will ever be able to do.
I am forever grateful for their generous act of sharing knowledge.

So, if this guide is ever useful to you, it is thanks to them.

Please, see credits and links at the end of this document.

This part comes from a package with 4 parts in Pdf or Doc format and with a series of meshes and textures, referenced in the following pages. A series of flow charts images summarize the steps for each animation type.

If you received this tutorial without these resources, please visit http://www.nexusmods.com/fallout3/ and download the full package.

Tutorial is in four parts:
Part 1 An Introduction.
Part 2 This part about Material Animations.
Part 3 covering Textures Animations.
Part 4 will deal with Controller Managers, Sequences and Scripts.

TABLE OF CONTENT

This tutorial is focused on Static or Havoked object. No skins, armors or weapons have been tested.

All information shared here is meant for Fallout 3, but most probably will work for FNV as well.

ALPHA MATERIAL ANIMATION

STEP BY STEP

Open 01DemoAlpha in Geck preview (Right click, open with Geck) to have a quick view.

1. Open 01DemoAlpha in NifSkope and expand all the arrays. Examine the differences between the two posters.

It's part of my workflow to rename Properties. There are 2 reasons for that.

When using Duplicate or Copy functions, error messages may appear if arrays are not named.

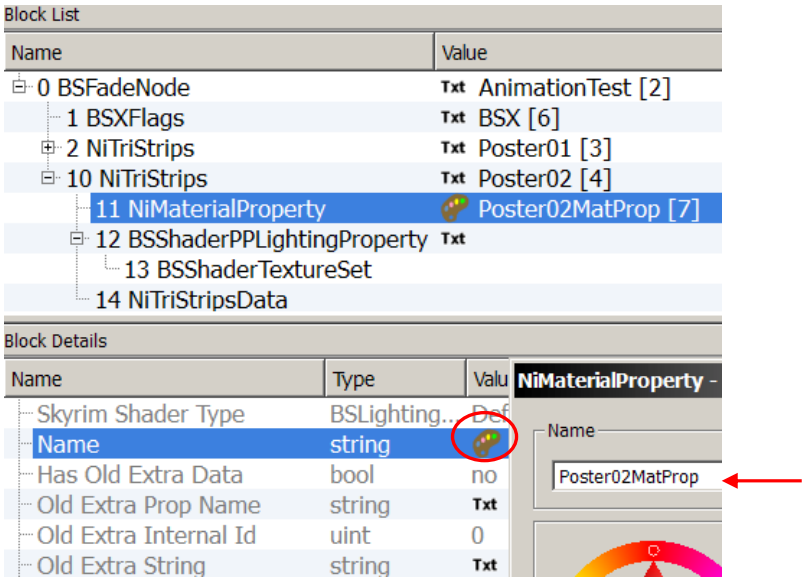
If several Controllers animate several Properties, it might be useful to have a coherent naming to quickly find connections.

2. In Poster02, rename the NiMaterialProperty.

   2 A. Select the NiMaterialProperty.

   2 B. In Block Details, click on the painter's palette.

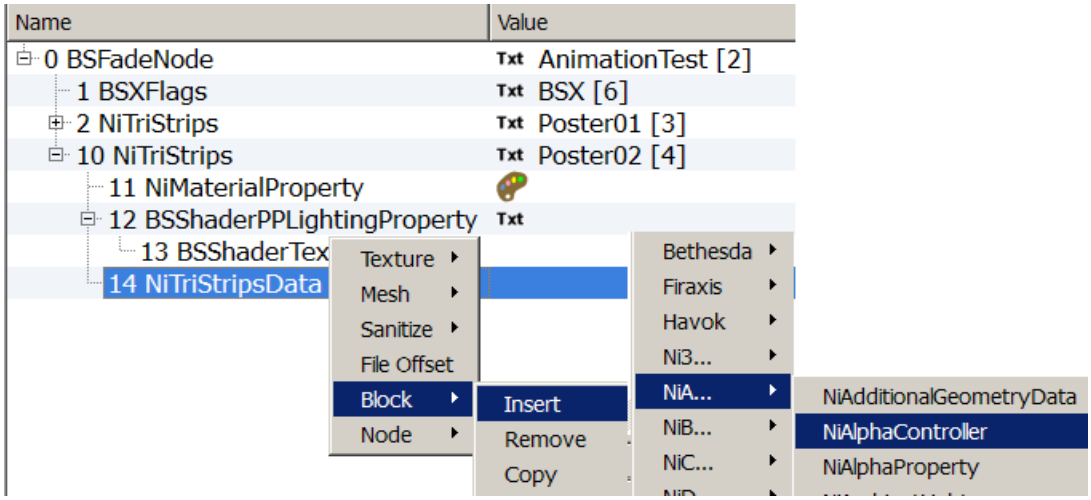
   2 C. A window should appear where you can enter a suitable name. Accept.



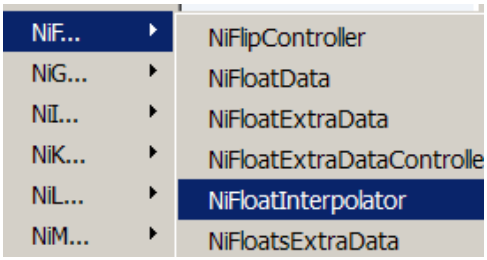
3. Insert all the elements needed to build an animation block. A Controller, an Interpolator and some key Data storage.

In Block List, select the last element then,

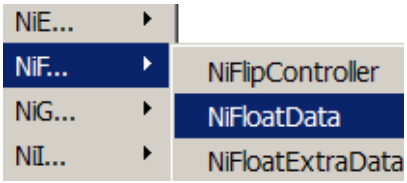
3A. Right click Block ⇨ Insert ⇨ NiA ⇨ NiAlphaController



3B. Right click Block ⇨ Insert ⇨ NiF ⇨ NiFloatInterpolator.



3C. Right click Block ⇨ Insert ⇨ NiF ⇨NiFloatData

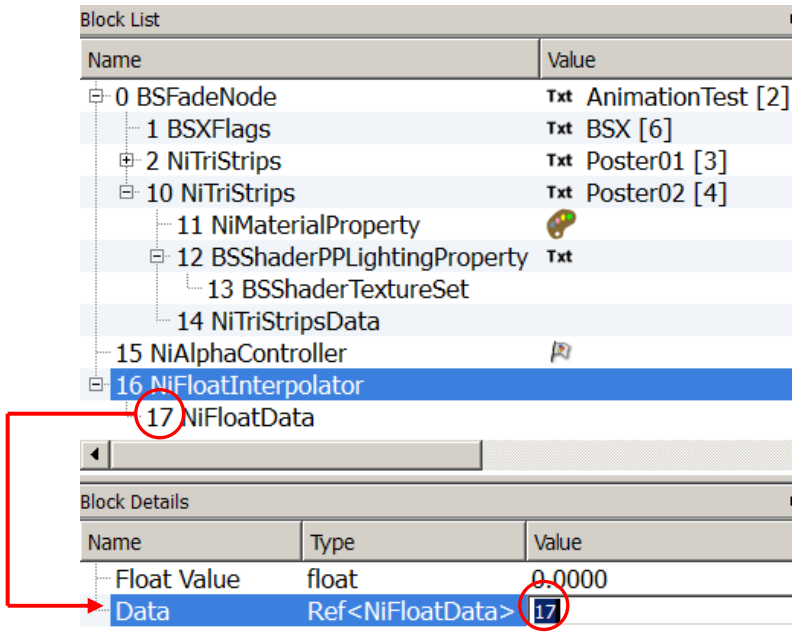


These 3 elements need to be connected.

4. Linking the Data to the Interpolator.

Select the Interpolator, in Block Details:

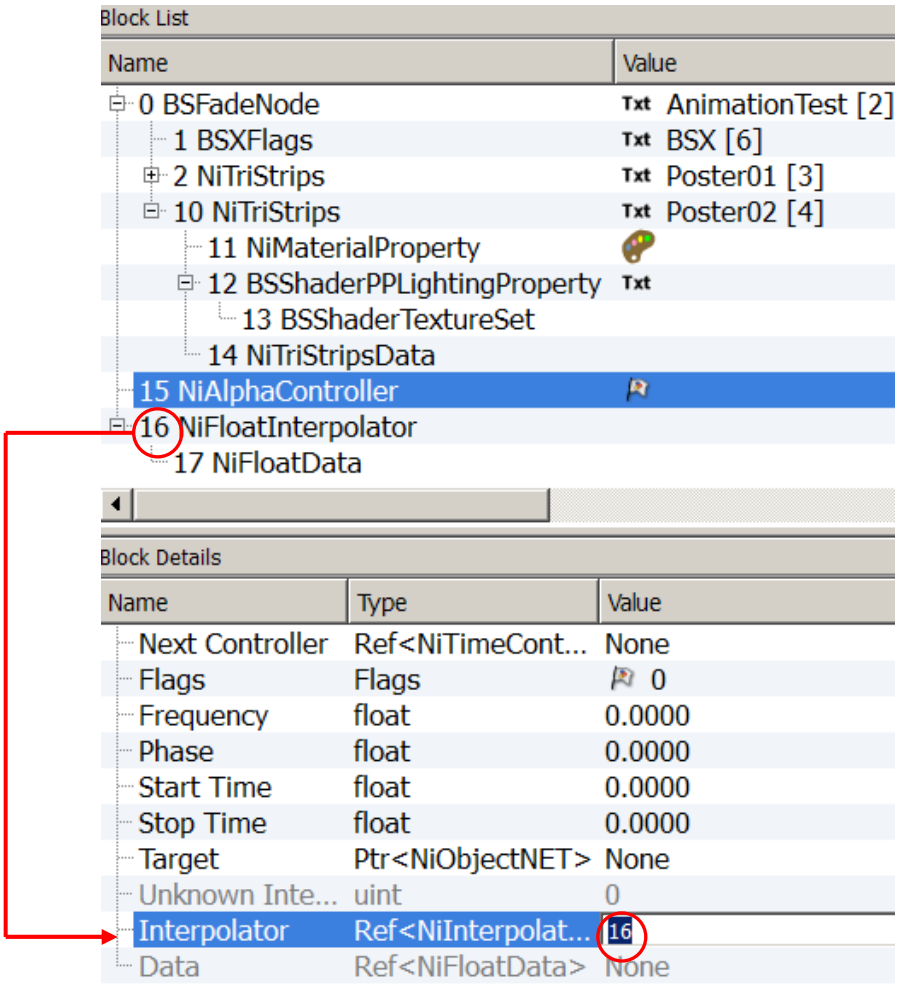
Set the value of the Data with the value of the NiFloatData.



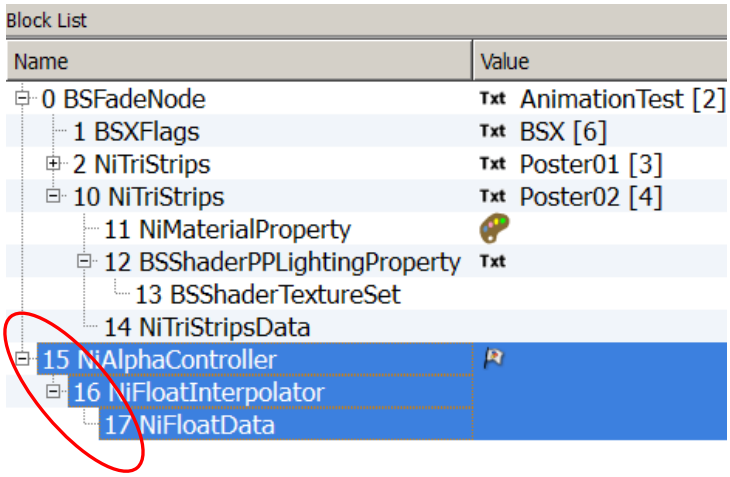
5. Linking the Interpolator to the Controller.

Select the Controller, in Block Details:

Set the value of the Interpolator with the value of the NiFloatInterpolator.



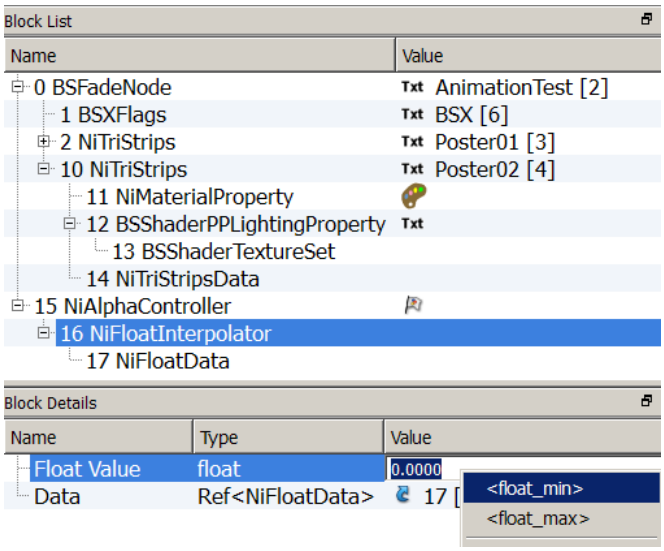
You have now a linked up Animation Block.



Let's adjust the settings of the different elements to have it workable.

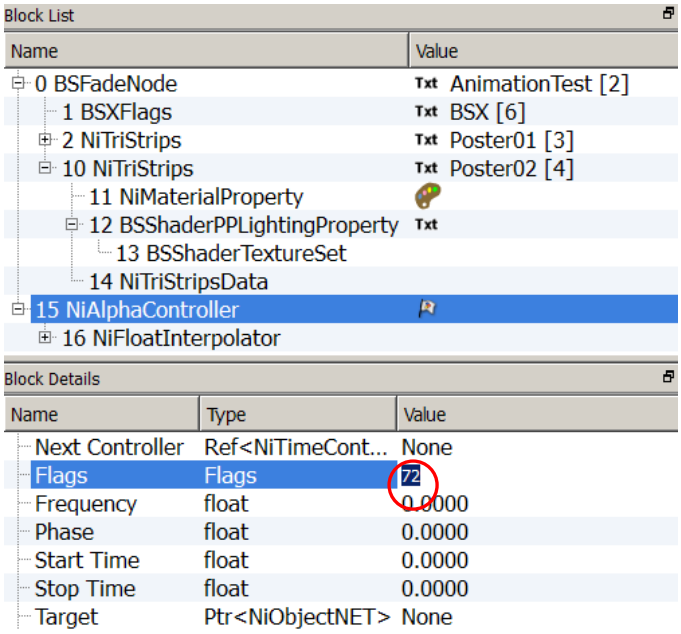
6. Select the Interpolator, in Block Details :

Double click on the Value field of the Float Value, then right click and select <float_min>.

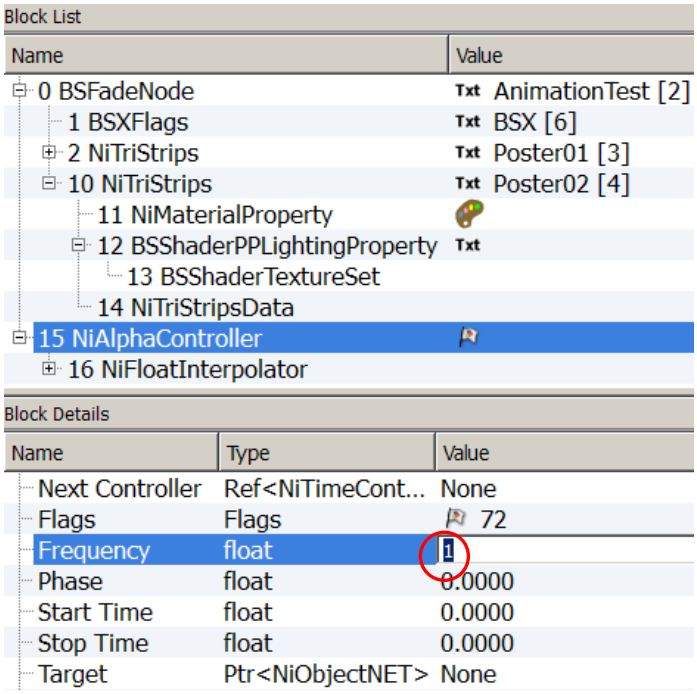


7. Select the Controller, in Block Details :

7A. Set the Flag to 72. This will set the Controller in Loop mode, so that the animation will be continuously playing. If set to 76, it will be in Clamp mode, meaning the animation will be playing once then stop.



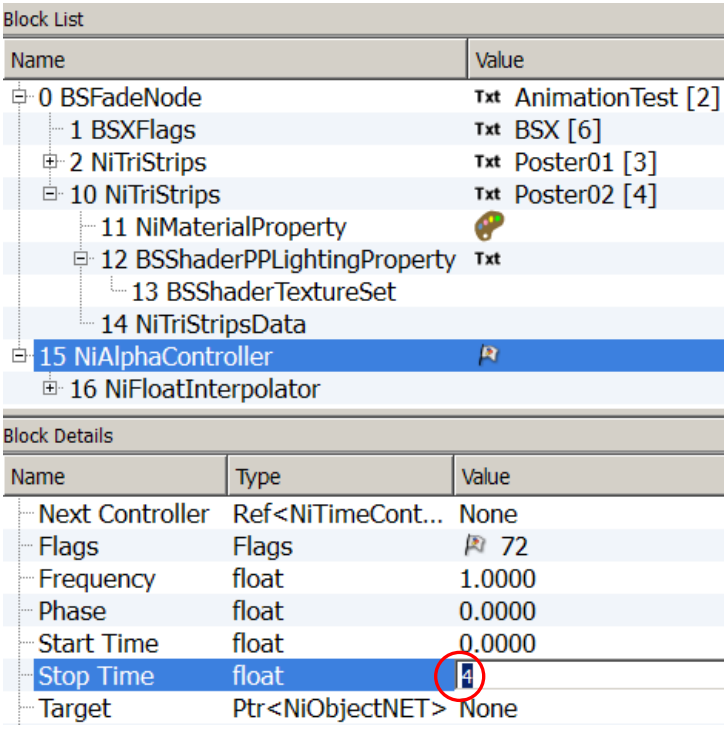
7B. Set the frequency to 1. This is the standard setting. Changing it will result in accelerated or slowed down animations.



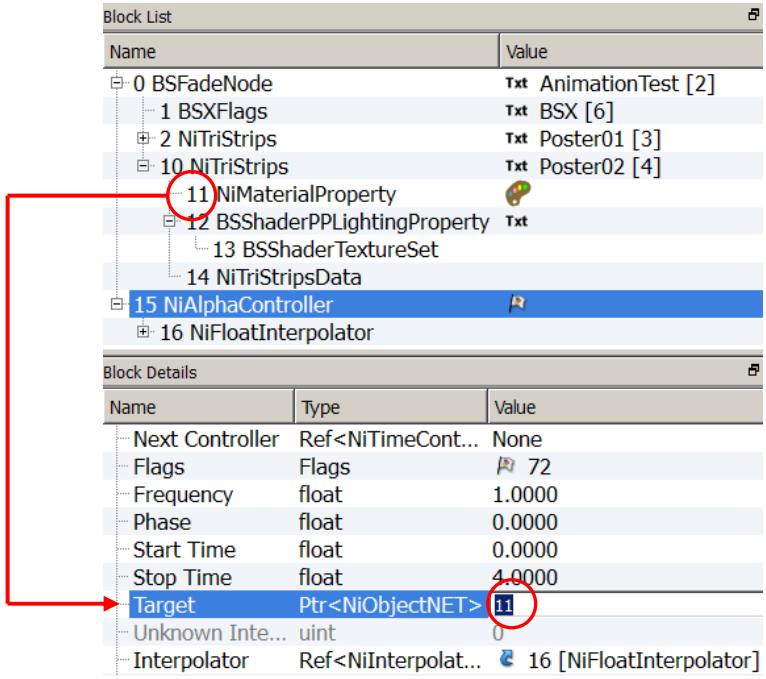
Phase is left on 0.

The Start Time will be left on 0 too.

7C. Setting Stop Time. This is, actually, where you set the global length of your animation. For our demonstration, we will choose a 4 seconds length. Set 4 in the Stop Time Value.

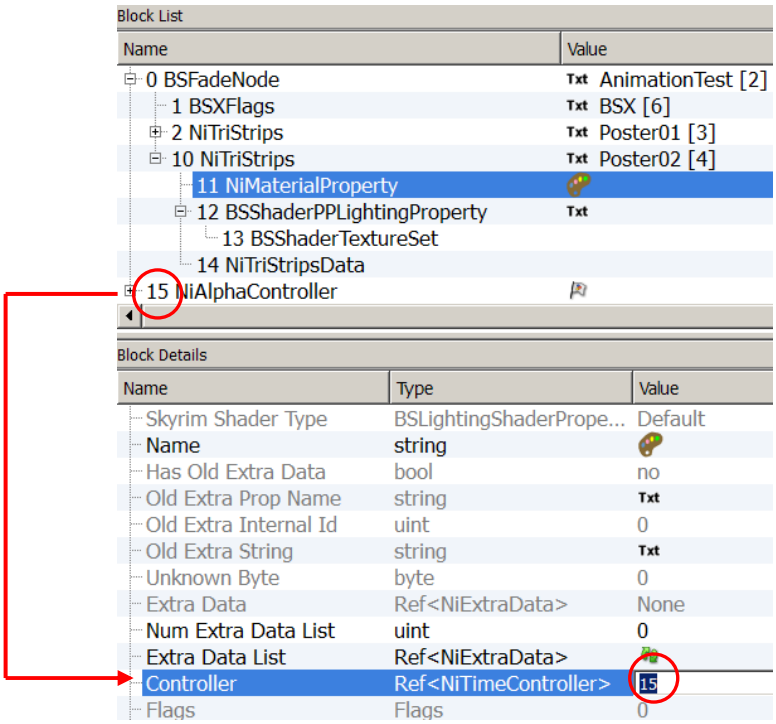


7D. We will now target the Material Property that needs to be animated. Set the Value of the Target with the Value of the NiMaterialProperty.



To finalize the linkage, we will tell the Material Property who is its controller.

8. Select the NiMaterialProperty and set the value of the NiAlphaController in the Controller Value.



It might be a good thing to save at this point.

Make sure to save under a new name (like MyAlpha01.nif)

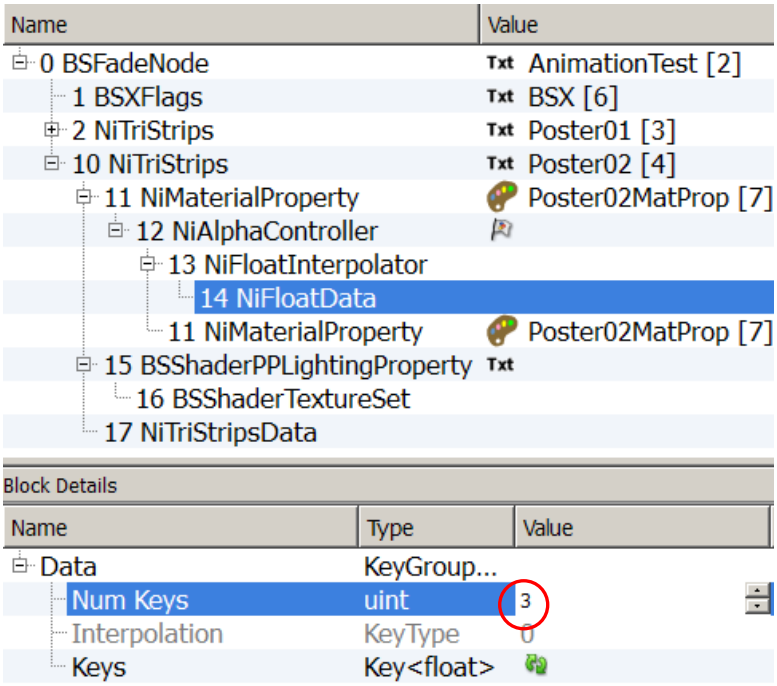
You have now a workable Alpha Animation block. Yet, it doesn't contain any Key Data.

That will be the next step.

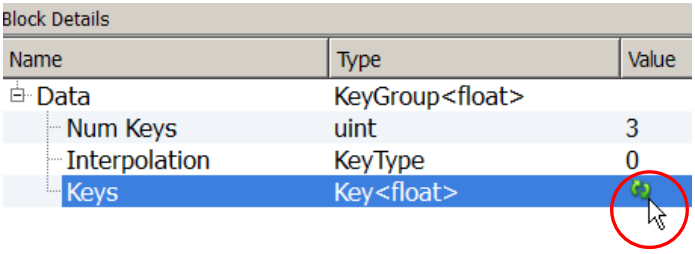
9. Keys settings.

Expand the Poster02 tree till you select the NiFloatData.

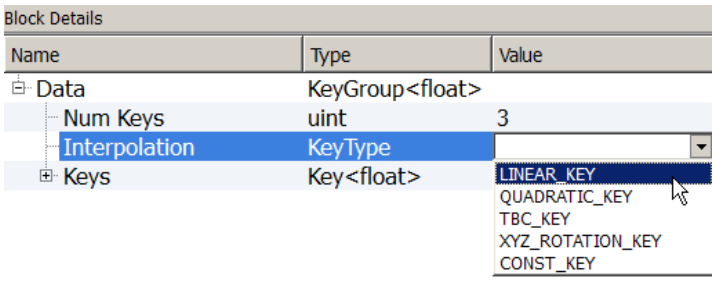
9A. In Block Details, insert the number of Keys you'll want to use. In this case, 3



9B. Update.



9C. Select the Interpolation mode. Double click to access the drop down menu.



You must insert a number of Keys first to be able to choose Interpolation.

The Key Value can be set to a range from 0 to 1.

0 will set the texture invisible, 0.5 will make it 50% transparent and 1 fully opaque.

9D. Expand all the Keys and set the Time and Values as followed.

Block Details

Name	Type	Value
Data	KeyGroup<float>	
Num Keys	uint	3
Interpolation	KeyType	LINEAR_KEY
Keys	Key<float>	
Keys	Key<float>	
Time	float	0.0000
Value	float	1.0000
Forward	float	0.0000
Backward	float	0.0000
TBC	TBC	X 0.0000 Y 0.
Keys	Key<float>	
Time	float	3.5000
Value	float	0.0000
Forward	float	0.0000
Backward	float	0.0000
TBC	TBC	X 0.0000 Y 0.
Keys	Key<float>	
Time	float	4.0000
Value	float	1.0000
Forward	float	0.0000
Backward	float	0.0000
TBC	TBC	X 0.0000 Y 0.

At the beginning of the animation (Time 0) the texture will be fully visible (Value 1).

It will linearly disappear till it's fully transparent at 3.5 seconds (Value 0).

As soon as it has reached this point, it will regain visibility in half a second till the end (Time 4, Value 1), then it will seamlessly loop and restart its sequence. Not rendered in NifSkope preview.

Save as MyAlpha02.nif. Open in Geck preview.

It's up to you to play and experiment a bit, now.

You might want to add keys or change the Time and Values. (8D).

You might want to change the length of the sequence in the NiAlphaController. (6C).

You might want to change the Interpolation mode. CONST_KEY will act as a switch. (8C).

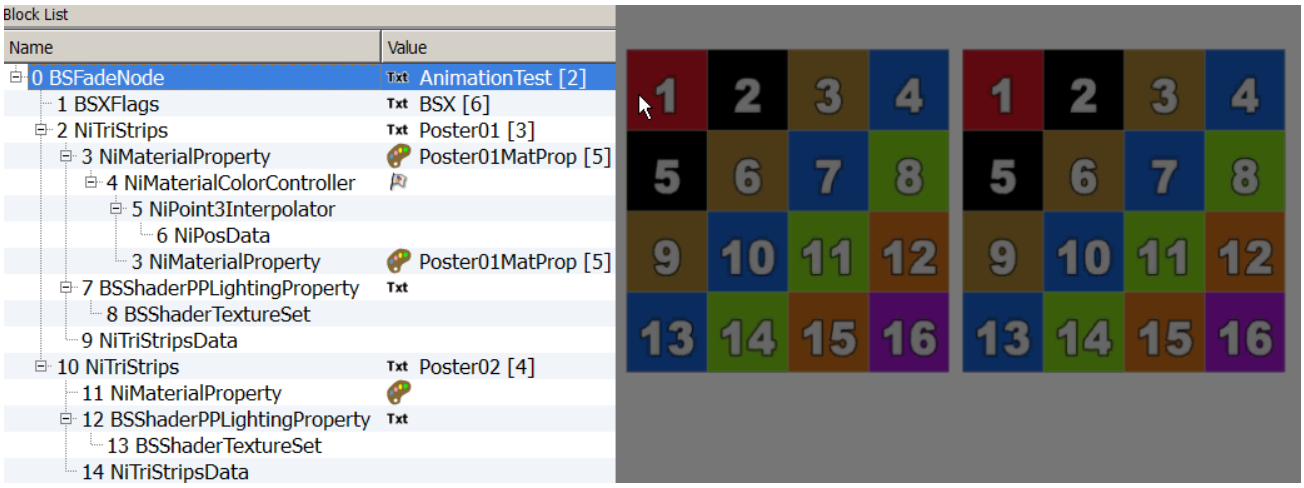
Don't use XYZ_ROTATION_KEY though, as it seems to be dedicated to object animation and crashes with Texture Animation.

EMISSIVE MATERIAL ANIMATION

STEP BY STEP

Open 02DemoEmissive in Geck preview to have a quick view.

1. Open 02DemoEmissive in NifSkope and expand all the arrays. Examine the differences between the two posters.



2. Rename the NiMaterialProperty.

3. To build an Emissive animation block, the procedure is the same as for Alpha animation: first inserting the elements.
   Only the Controller, the Interpolator and Data storage are from a different type.

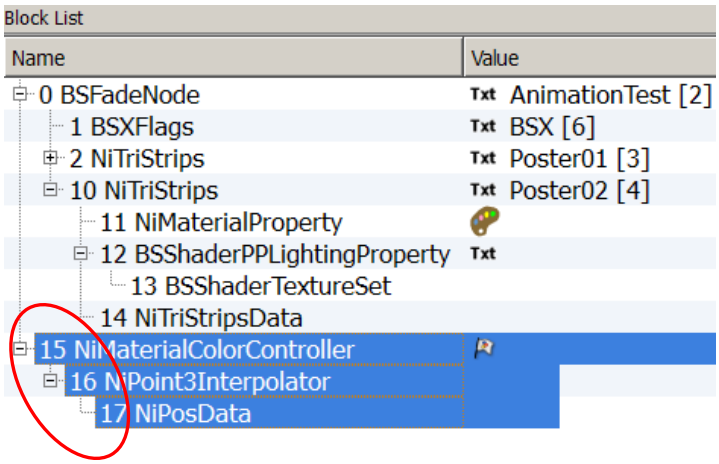
   Use Poster02. In Block List, at the bottom of the list,

   3A. Right click Block ⇨ Insert ⇨ NiMaterialColorController

   3B. Right click Block ⇨ Insert ⇨ NiPoint3Interpolator

   3C. Right click Block ⇨ Insert ⇨ NiPosData

Connect these three elements the same way as before, to have a linked Block.



4. Link the Data to the Interpolator.

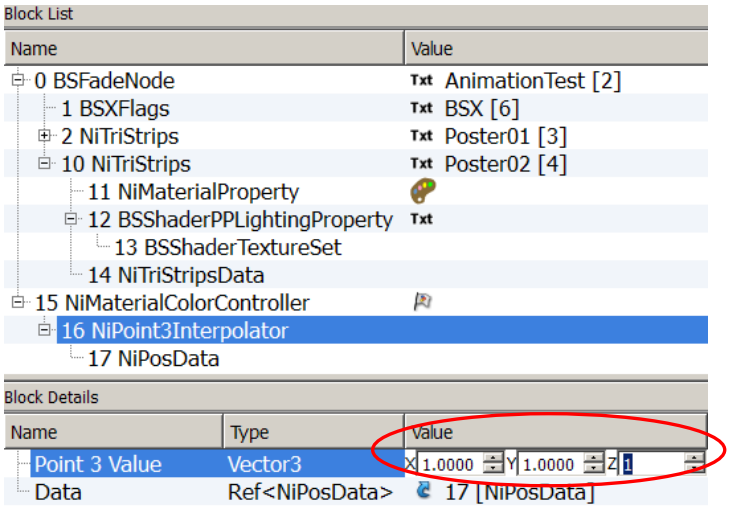
   Select the Interpolator, set the value of the Data with the value of the NiPosData.

5. Link the Interpolator to the Controller.

   Select the Controller, set the value of the Interpolator with the value of the NiPoint3Interpolator.

Adjust the settings to have the block workable.

6. Select the Interpolator.
In Block Details, set its three Value fields to 1



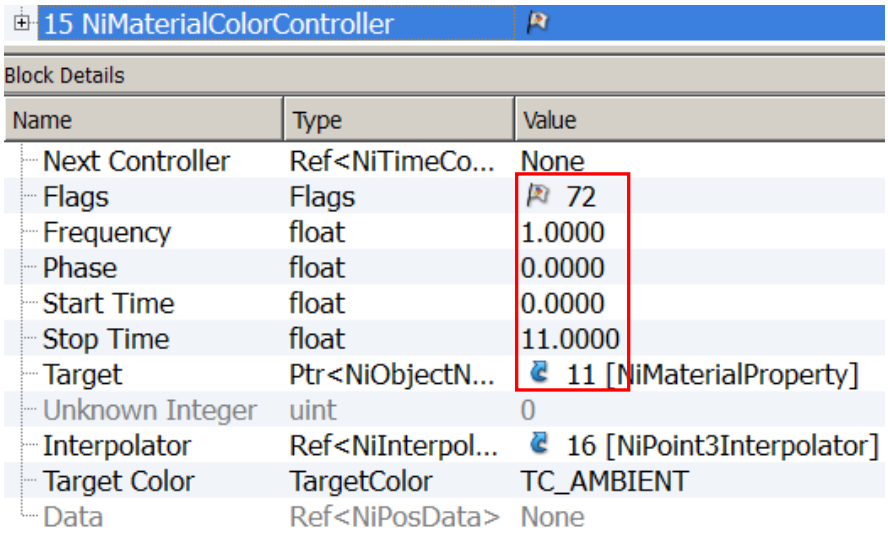
7. Select the Controller.
In Block Details:

7 A. Set the Flag to 72 for Loop mode. (76 if you want Clamp mode.)

7 B Set the Frequency to 1

7 C Set the length of the animation, eleven seconds. Start Time 0, Stop Time 11

7 D Target the NiMaterialProperty that needs to be animated.



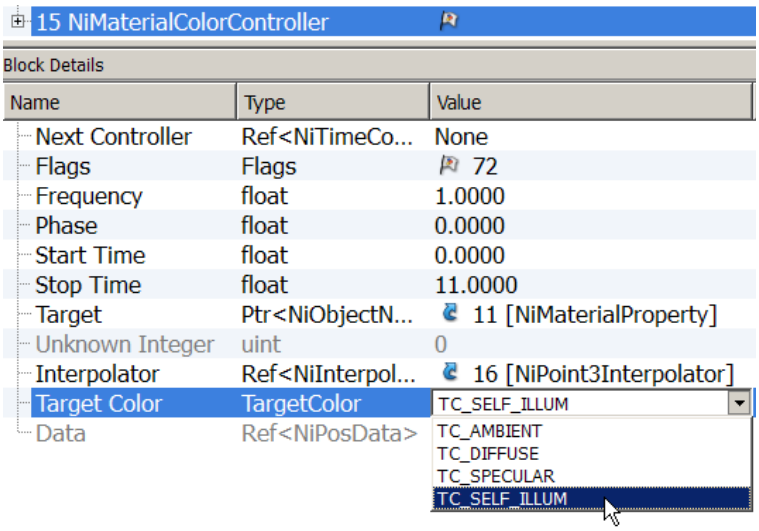
Till now, every step to build a Material animation Alpha or Emissive have been similar, only the type of Controller, Interpolator and Data storage are different. Knowing (and understanding) this procedure until it becomes automatic, is very useful to quickly build animation blocks.

The next step is specific to Emissive animations.

You'll have to choose a Target Color.

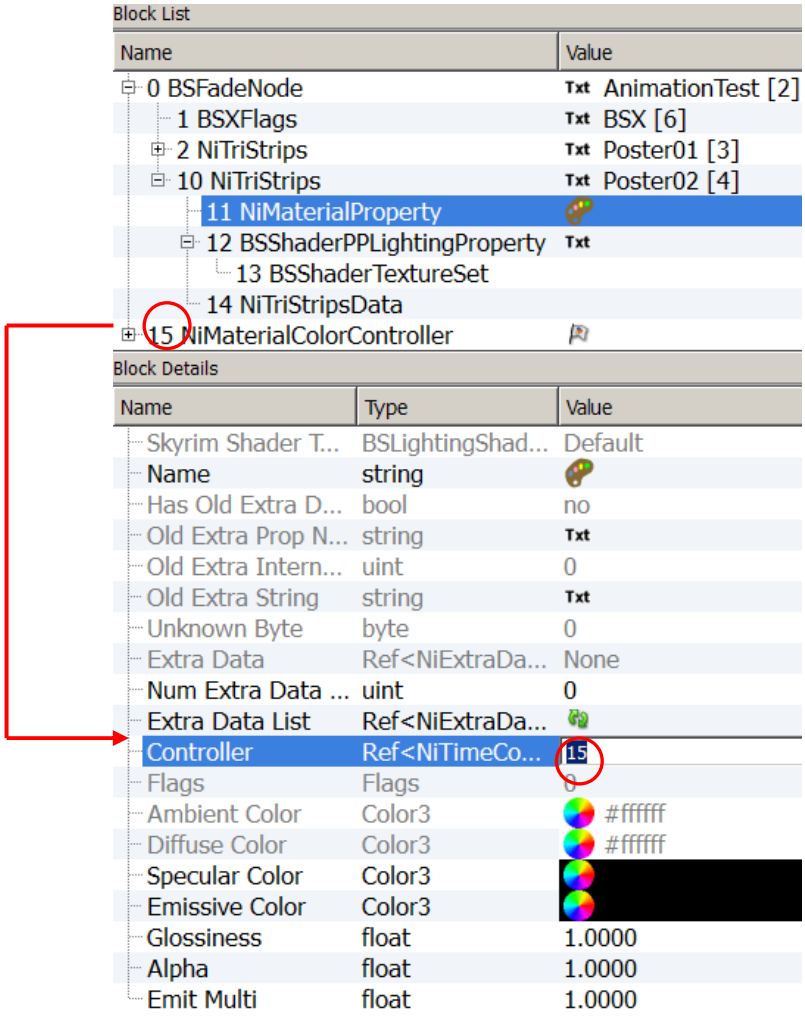
8. Select the Controller.

In Block Details: Double click on the Target Color Value to access the drop down menu and select TC_SELF_ILLUM.



Don't forget to finalize the linkage, by telling the Material Property who is its controller.

9. Select the NiMaterialProperty and set the value of the NiMaterialColorController.



Again, it would be wise to save your progression at this point. Firstly, to let the structure reorganize itself and secondly, to see if there's any major problem by checking in Geck preview before we set the Keys Data.

The three Vector3 Values X Y Z are corresponding to R G B color Values with a minimum of 0 and a maximum of 1. If you want a specific color to be emitted, you have to know its RGB Value.

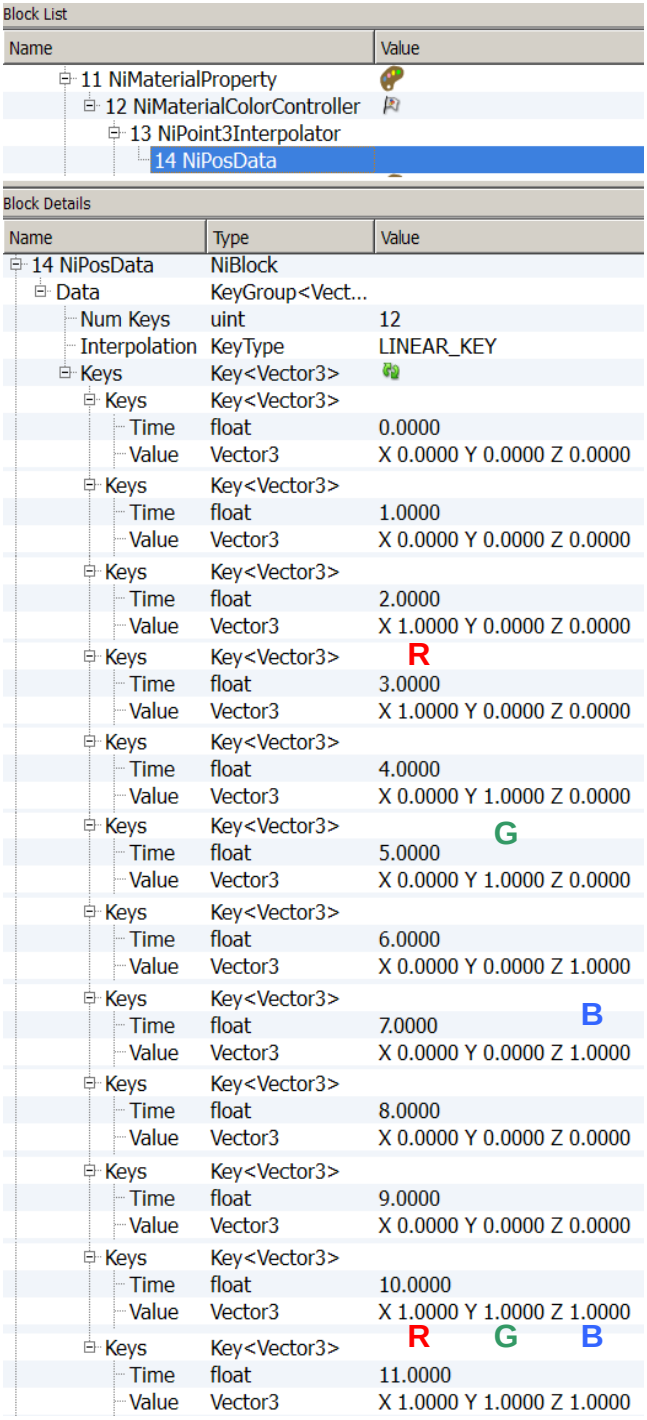
For this demo, we've chosen to set the following sequence:
The animation start with the poster emitting nothing and staying so for 1 second, then it will slowly and linearly turn red in 1 second and stay glowing red at its max for 1 second, then turning green and staying, turning blue and staying, stop emitting and staying, emitting fully and staying at its max for 1 second and finally cutting neat to restart the sequence.

10. Key setting. Select the NiPosData.

10 A. In Block Details, insert the number of Keys needed. In this case: 12. Update.

10 B. Choose the Interpolation Key Type: LINEAR_KEY.

10 C. Expand the keys and set the keys as followed then save your file with a distinct name.



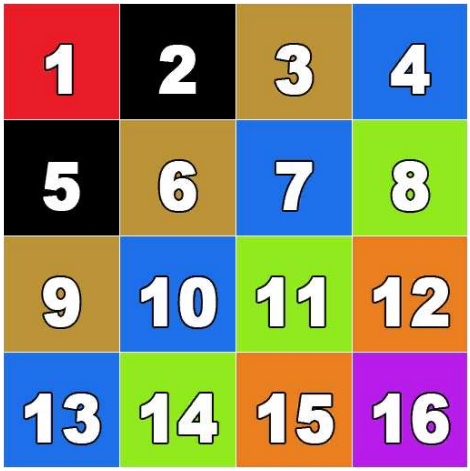
You can save and check your work in Geck. Again it's up to you to play and experiment with settings.

GLOW MAP

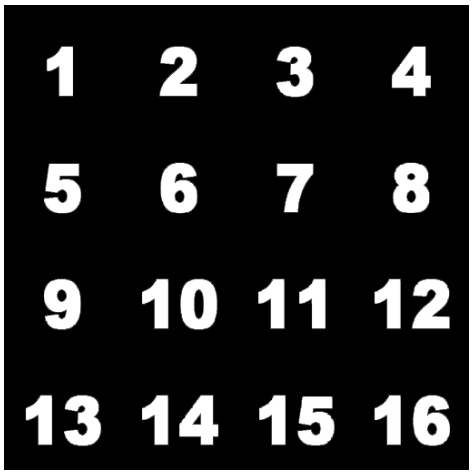
Emissive or Glowing Animations can achieve various effects: from flickering screen to pulsating bits on a machine, Glowing Chems, etc.

It can be combined with a Glow Map. The Glow Map will act as a mask, allowing certain parts of the texture to be affected by the Emissive Value of the NiMaterialProperty, while other parts are not.

The white parts will be showed fully glowing; the black part will remain un-glowing. Colors can be used too.



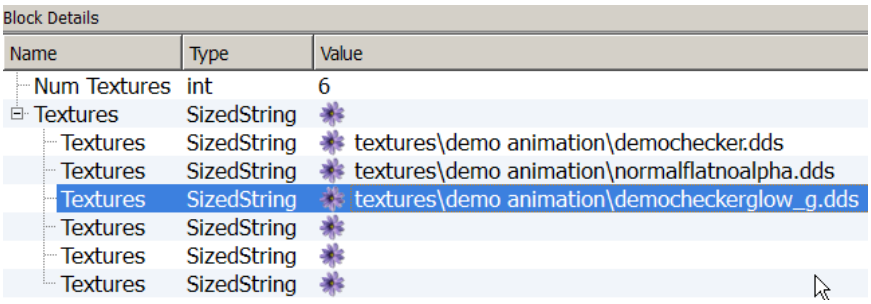
Diffuse Map



Glow Map

Glow Maps usually have an “underscore g” following their names, like: TextureName_g.dds.

Glow Maps are placed in the third slot in the BSShaderTextureSet.



To see an example, open 02DemoEmissiveGlowMapExample.nif in Geck preview.

A WORD ABOUT SPECULAR MATERIAL ANIMATION

Specular animation should be possible, theoretically, but:

As with other features, it is fairly possible that it is not available in FO3 /FNV engine. I found nothing on Internet about it.
I have been told that it is working in Skyrim, though and maybe in Morrowind.

I never managed to have a visible result in my testing, apart from in NifSkope preview.

(See 03SpecularExample.nif)

I have never seen a Vanilla Nif using Specular animations.

According to Ghogiel, specular animations are overridden by Specular Map (alpha channel of the Normal map), which means that the Normal Map must be saved in DXT1-NoAlpha format.

http://wiki.tesnexus.com/index.php/Adding_material_controllers_to_objects_in_Nifskope

Adding or removing SF_Specular in the BSShaderPPLightingProprety doesn't seem to change anything. Working with BSShaderNoLightingProperty either.

I think I've tried all the possible combinations, with all kind of Normal Maps but I might be missing a setting somewhere.

To build a specular animation block, the procedure is should be exactly the same as for Emissive Animation.
Only the NiMaterialColorController will Target Color TC_SPECULAR.

If you have any information or knowledge to share about this, you are most welcome to contact me.

Pm Pixelhate at http://forums.nexusmods.com/index.php?showuser=1205226

I will be happy to update!

CREDITS (ALPHABETIC)

I would like to thank deeply and sincerely.

- **Anoxeron** for his dedication, enthusiasm and benevolence. Thank you for testing each exercise, each step without respite and for giving me countless opportunities to improve this tutorial. Thank you for proof reading and corrections.
  http://www.nexusmods.com/fallout3/users/3694499/?
- **BrettM** for his tutorial about Textures Animations for Skyrim, despite a big difference in the engine, his tutorial was an inspiration for layout and presentation.
  http://www.nexusmods.com/skyrim/mods/47104/?
- **Ghogiel** for his guide about NiMaterialControlers, his resources and mods which are real learning material and for his kind private answers about Specularity.
  http://wiki.tesnexus.com/index.php/Adding_material_controllers_to_objects_in_Nifskope
  http://www.nexusmods.com/fallout3/users/209926/?tb=mods&pUp=1
- Nexus for hosting this.
  http://www.nexusmods.com/fallout3/
- **Prensa** for her kindness, constant support and patient explanations. For sending me back faulty experiments, miraculously repaired and working. Her posts, on Nexus, can be compiled and will answer most of the troubles you'll encounter in FO3. Thank you for proof reading and corrections.
  http://www.nexusmods.com/fallout3/users/751212/?tb=mods&pUp=1
  http://forums.nexusmods.com/index.php?/user/751212-prensa/
- **Sullyvanj93** for partial proof reading and writing advices.
- **TrickyVein** for his very clear tutorial on NiControllerManager on Nexus which has helped me to tame "this intimidating beast" and understand the necessity to develop a workflow.
  http://forums.nexusmods.com/index.php?/topic/984792-tutorial-working-with-the-nicontrollermanager/
- **Turboscalpeur** for his friendship, support and beautiful images.
  http://forums.nexusmods.com/index.php?/user/4708873-turboscalpeur/
  https://www.flickr.com/photos/turboscalpeur/
- **Weijiesen** for overall theory consistency scrutinizations and awesomeness!
  http://www.nexusmods.com/newvegas/users/1026866/?tb=images&pUp=1
- The good people who takes time to write articles, tutorials, guides and make them available to others.
  https://www.google.com
- The nice people helping on forums like Nexus, NifTools Forum, etc.
  http://forums.nexusmods.com/
  http://niftools.sourceforge.net/forum/
- The makers of The Elder Scrolls NifSkope guide for their very detailed guide offering the bases on what this tutorial is build
  http://cs.elderscrolls.com/index.php?title=Working_With_Nifs_101_:_An_Introduction
  http://cs.elderscrolls.com/index.php?title=Working_With_Nifs_101_:_Basic_Use
- The makers of the tools that allow us to express our creativity and share it.
  NifSkope http://sourceforge.net/projects/niftools/files/nifskope/1.1.3/
  Geck http://geck.bethsoft.com/index.php?title=Main_Page
  Geck PowerUp http://www.nexusmods.com/fallout3/mods/15067/?