MATERIAL AND TEXTURE ANIMATION WITH NIFSKOPE:

PART 3: TEXTURE 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This guide is the result of compiling information on the use of NifSkope by reading tutorials (some quite outdated) and Wikis, tracking help posts on forums or by trying to figure things out by myself.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 about Material Animations.
Part 3 This part covering Textures Animations.
Part 4 will deal with ControllerManager, Sequence and Script.

TABLE OF CONTENT

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

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

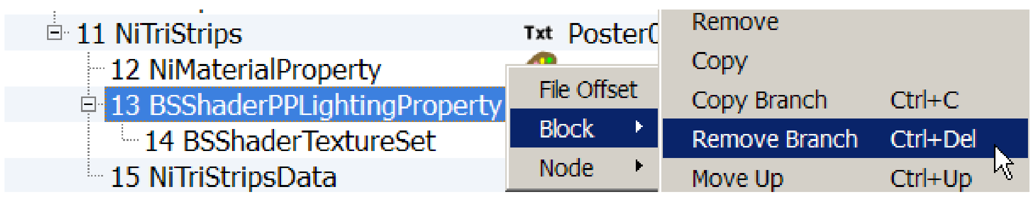
Shader Preparatory Work

The part you want to Texture Animate **must** have a BSShaderNoLightingProperty, a NiTexturingProperty and a NiSourceTexture in charge of the texture, in order to work. If your NiTriStrips doesn't have the correct structure, you'll need to prepare it.

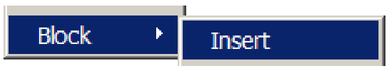
(Use Poster02 in 01DemoRotate.nif for exercise.)

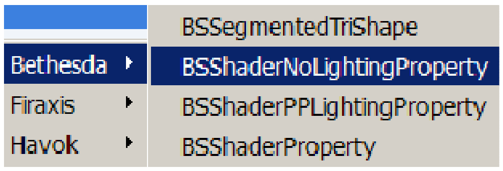
Here's the procedure:

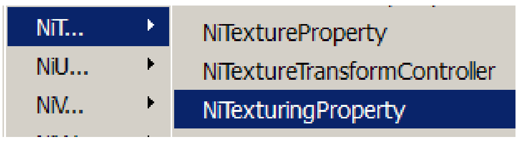
A. Removing the BSShaderPPLightingProprety and the BSShaderTextureSet.

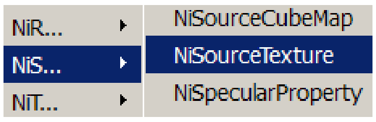


B. Inserting a BSShaderNoLightingProperty, a NiTexturingProperty and a NiSourceTexture.



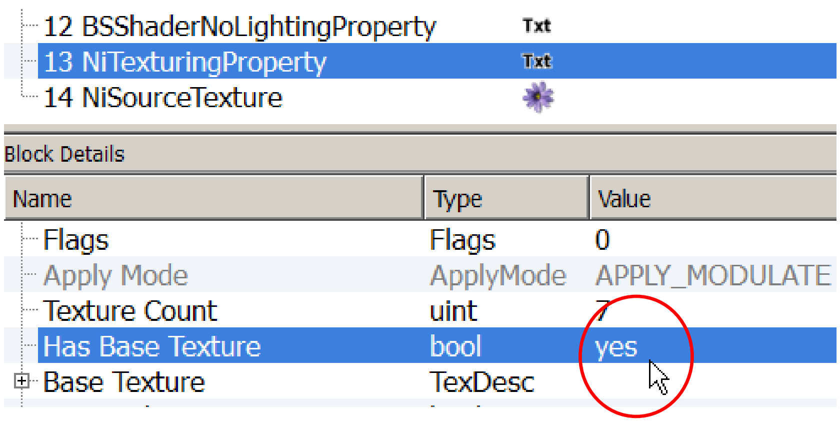






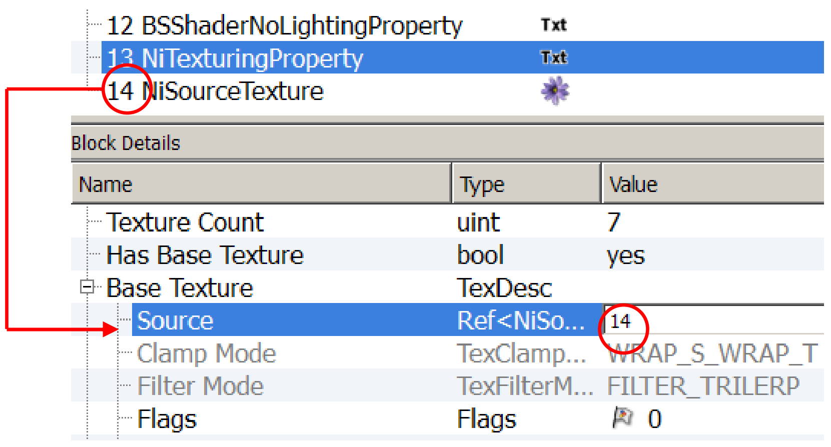
C. Linking the NiSourceTexture to the NiTexturingProperty:

- Select the NiTexturingProperty. In Block Details, double click in the Value field of the **Has Base Texture** to switch it from No to Yes.



Name	Type	Value
Flags	Flags	0
Apply Mode	ApplyMode	APPLY_MODULATE
Texture Count	uint	7
Has Base Texture	bool	yes
Base Texture	TexDesc	

- Expand the Base Texture tree now available. Set the Value of the Source with the Value of the NiSourceTexture.



12 BSShaderNoLightingProperty Txt
13 NiTexturingProperty Txt
14 NiSourceTexture

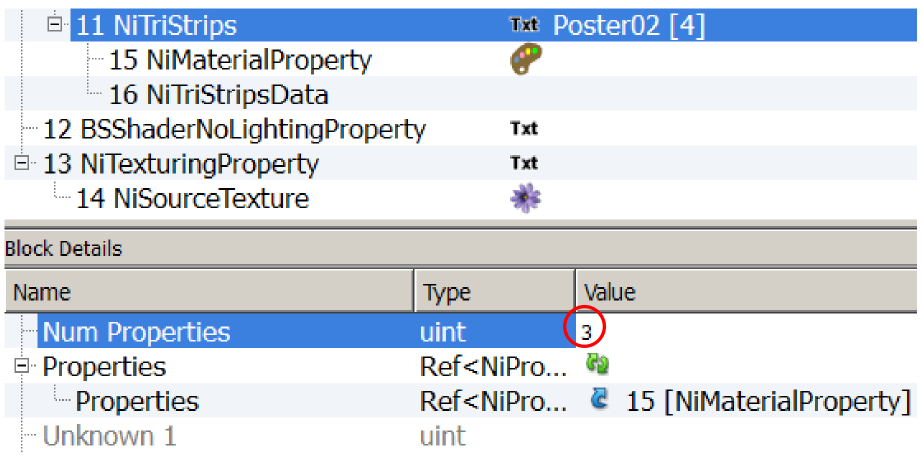
Block Details

Name	Type	Value
Texture Count	uint	7
Has Base Texture	bool	yes
Base Texture	TexDesc	
Source	Ref<NiSo...	14
Clamp Mode	TexClamp...	WRAP_S_WRAP_T
Filter Mode	TexFilterM...	FILTER_TRILERP
Flags	Flags	0

We will look after the other settings later.

D. Assigning the Properties to the NiTriStrips.

- Select the NiTriStrips. In Block Details, expand Properties.
  Add the number of properties needed. Update.
  (We need three in total, the existing one plus our Shader and Texturing properties.)

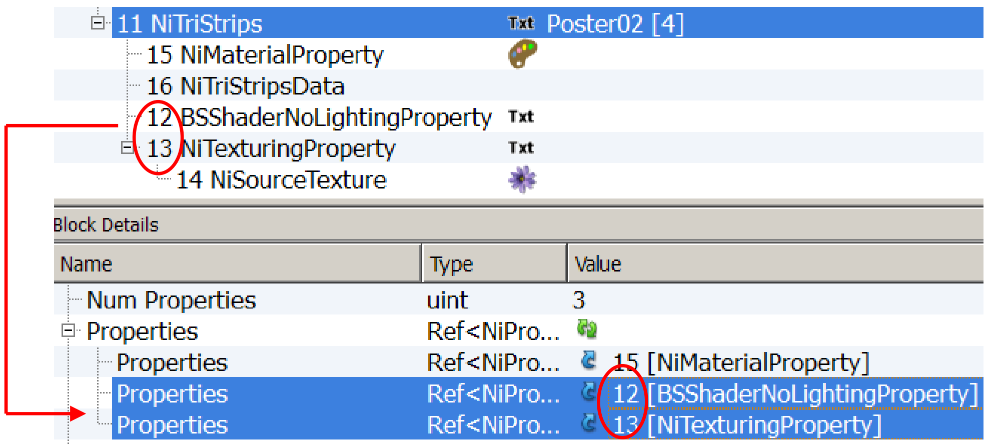


11 NiTriStrips Txt Poster02 [4]
15 NiMaterialProperty
16 NiTriStripsData
12 BSShaderNoLightingProperty Txt
13 NiTexturingProperty Txt
14 NiSourceTexture

Block Details

Name	Type	Value
Num Properties	uint	3
Properties	Ref<NiPro...	
Properties	Ref<NiPro...	15 [NiMaterialProperty]
Unknown 1	uint	

- Double click in the Value fields of the blank Properties and enter the Values of the BSShaderNoLightingProperty and the NiTexturingProperty.



11 NiTriStrips Txt Poster02 [4]
15 NiMaterialProperty
16 NiTriStripsData
12 BSShaderNoLightingProperty Txt
13 NiTexturingProperty Txt
14 NiSourceTexture

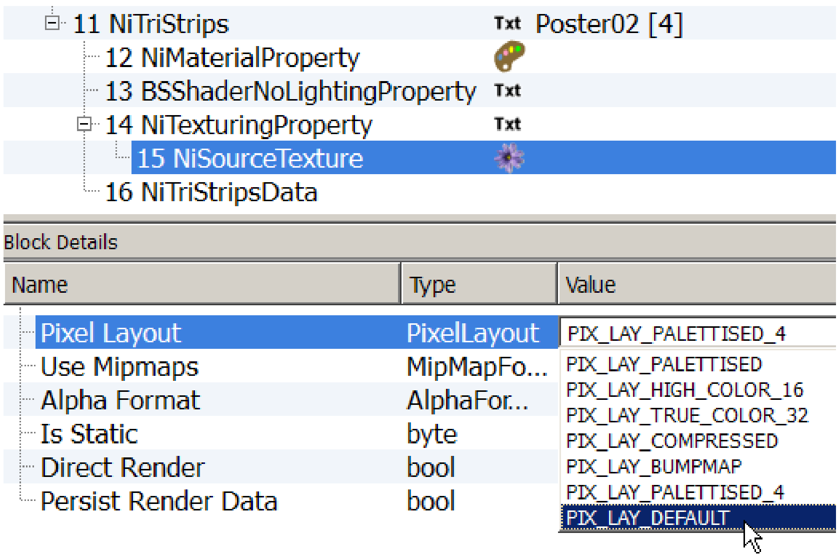
Block Details

Name	Type	Value
Num Properties	uint	3
Properties	Ref<NiPro...	
Properties	Ref<NiPro...	15 [NiMaterialProperty]
Properties	Ref<NiPro...	12 [BSShaderNoLightingProperty]
Properties	Ref<NiPro...	13 [NiTexturingProperty]

It is advisable to save at this point, so blocks will reorganize themselves.

E. Adjusting the settings of the NiSourceTexture.

- Select The NiSourceTexture. In Block Detail: double click in the Value field of the Pixel Layout to access the drop down menu and select PIX_LAY_DEFAULT.



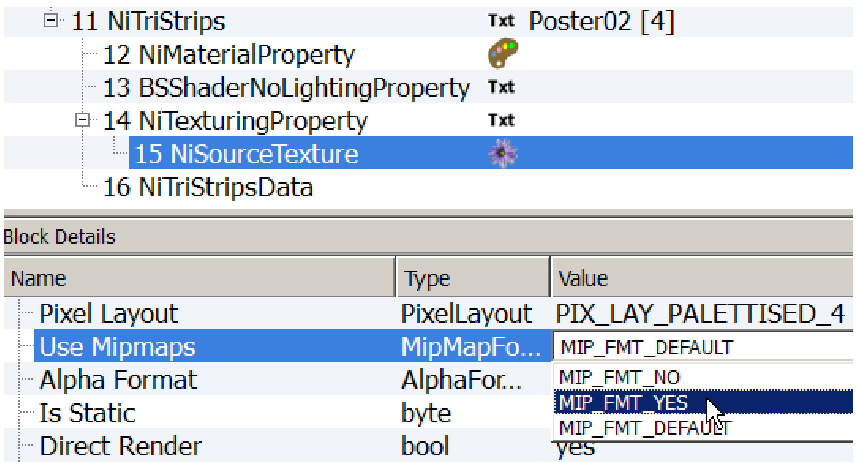
11 NiTriStrips Txt Poster02 [4]
12 NiMaterialProperty
13 BSShaderNoLightingProperty Txt
14 NiTexturingProperty Txt
15 NiSourceTexture
16 NiTriStripsData

Block Details

Name	Type	Value
Pixel Layout	PixelLayout	PIX_LAY_PALETTISED_4
Use Mipmaps	MipMapFo...	
Alpha Format	AlphaFor...	
Is Static	byte	
Direct Render	bool	
Persist Render Data	bool	

PIX_LAY_PALETTISED
PIX_LAY_HIGH_COLOR_16
PIX_LAY_TRUE_COLOR_32
PIX_LAY_COMPRESSED
PIX_LAY_BUMPMAP
PIX_LAY_PALETTISED_4
PIX_LAY_DEFAULT

- Same way, select MIP_FMT_YES for Use Mipmaps.



11 NiTriStrips Txt Poster02 [4]
12 NiMaterialProperty
13 BSShaderNoLightingProperty Txt
14 NiTexturingProperty Txt
15 NiSourceTexture
16 NiTriStripsData

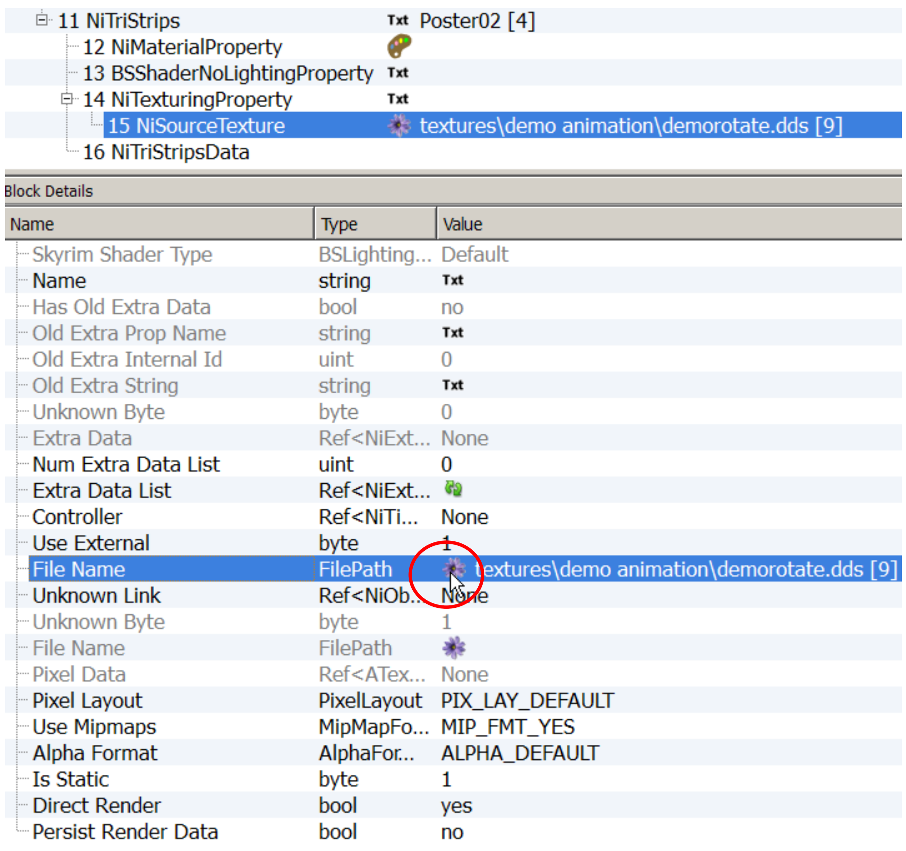
Block Details

Name	Type	Value
Pixel Layout	PixelLayout	PIX_LAY_PALETTISED_4
Use Mipmaps	MipMapFo...	MIP_FMT_DEFAULT
Alpha Format	AlphaFor...	
Is Static	byte	
Direct Render	bool	yes

MIP_FMT_NO
MIP_FMT_YES
MIP_FMT_DEFAULT

- Alpha Format should be set on APLHA_DEFAULT

- Click on the flower to choose your texture.



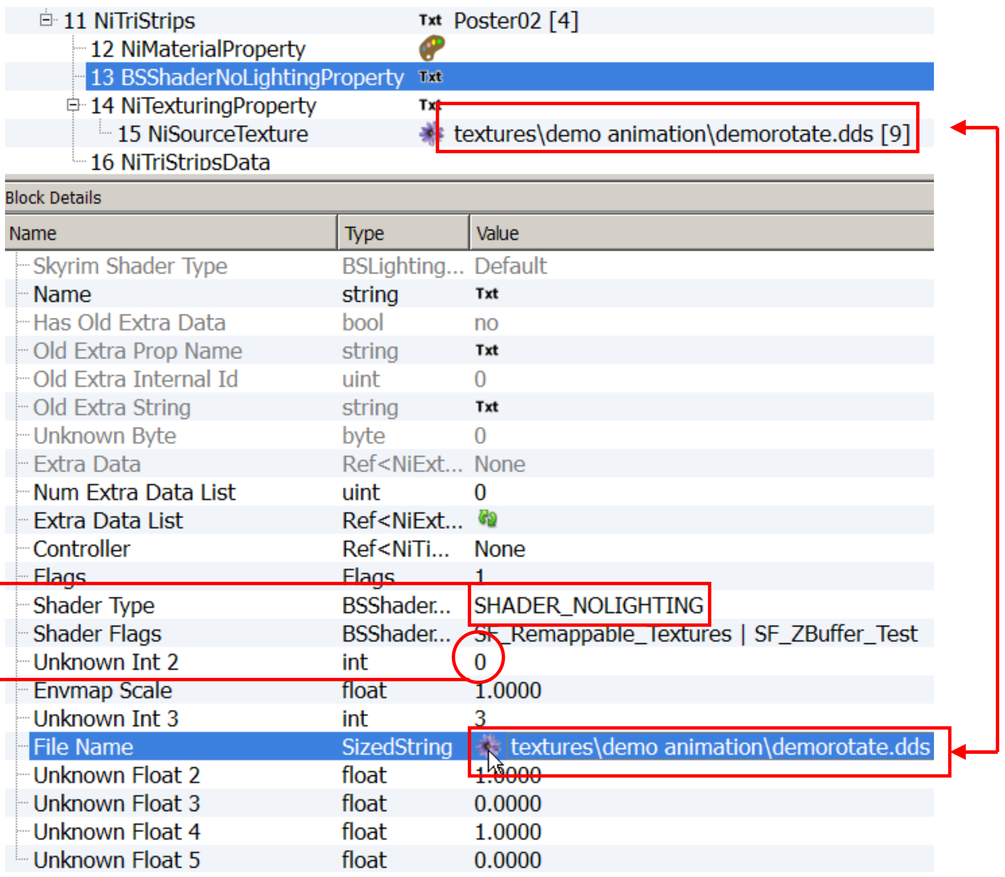
11 NiTriStrips	Txt Poster02 [4]
12 NiMaterialProperty	
13 BSShaderNoLightingProperty	Txt
14 NiTexturingProperty	Txt
15 NiSourceTexture	textures\demo animation\demorotate.dds [9]
16 NiTriStripsData	

Block Details

Name	Type	Value
Skyrim Shader Type	BSLighting...	Default
Name	string	Txt
Has Old Extra Data	bool	no
Old Extra Prop Name	string	Txt
Old Extra Internal Id	uint	0
Old Extra String	string	Txt
Unknown Byte	byte	0
Extra Data	Ref<NiExt...	None
Num Extra Data List	uint	0
Extra Data List	Ref<NiExt...	
Controller	Ref<NiTi...	None
Use External	byte	1
File Name	FilePath	textures\demo animation\demorotate.dds [9]
Unknown Link	Ref<NiOb...	None
Unknown Byte	byte	1
File Name	FilePath	
Pixel Data	Ref<ATex...	None
Pixel Layout	PixelLayout	PIX_LAY_DEFAULT
Use Mipmaps	MipMapFo...	MIP_FMT_YES
Alpha Format	AlphaFor...	ALPHA_DEFAULT
Is Static	byte	1
Direct Render	bool	yes
Persist Render Data	bool	no

The Texture needs to be referenced twice: in the NiSourceTexture and in the Shader.

- Select the BSShaderNoLightingProperty and click on the flower in the File Name Value Field to select your texture. It must be the same as in the NiSourceTexture.



11 NiTriStrips	Txt Poster02 [4]
12 NiMaterialProperty	
13 BSShaderNoLightingProperty	Txt
14 NiTexturingProperty	Txt
15 NiSourceTexture	textures\demo animation\demorotate.dds [9]
16 NiTriStripsData	

Block Details

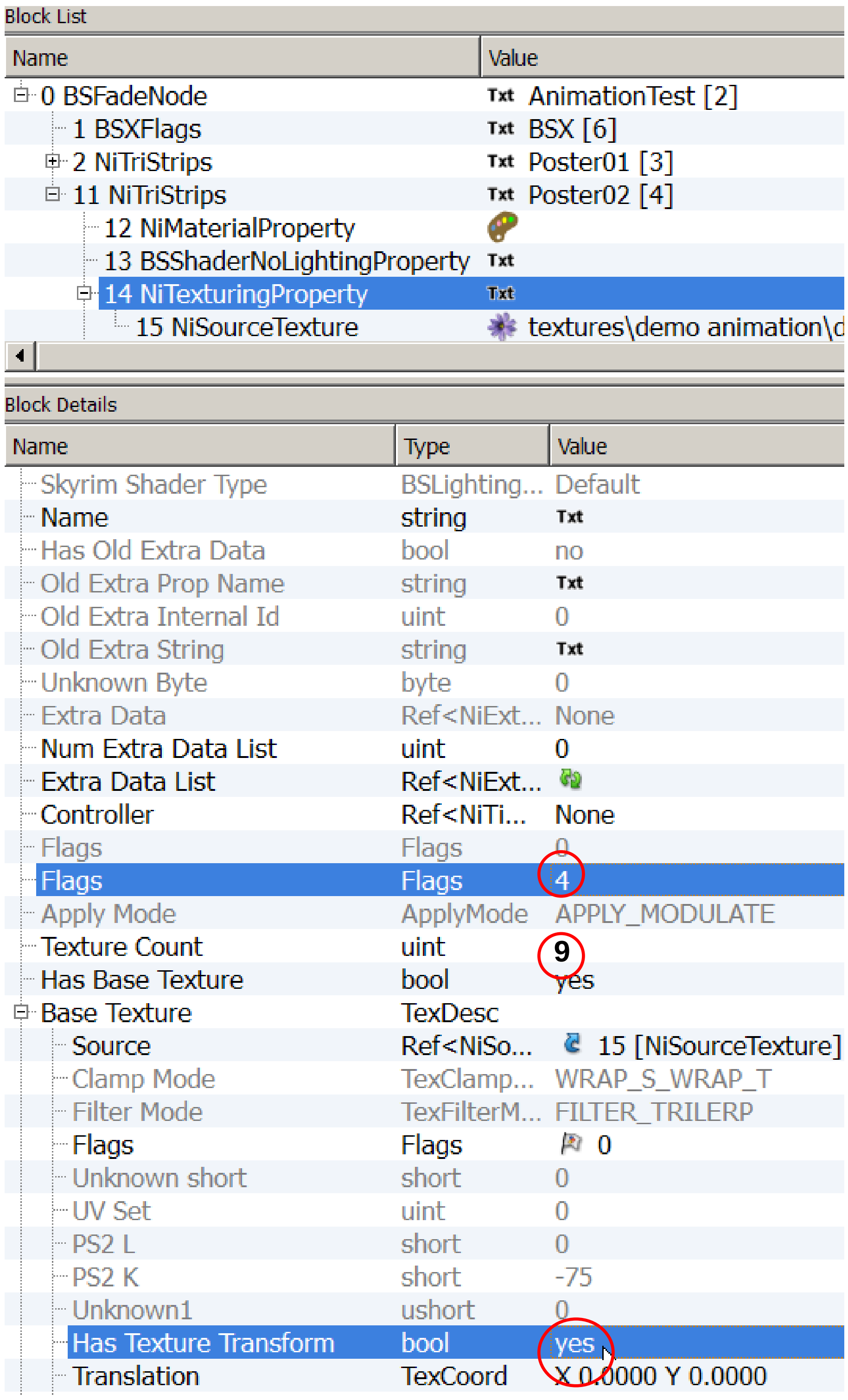
Name	Type	Value
Skyrim Shader Type	BSLighting...	Default
Name	string	Txt
Has Old Extra Data	bool	no
Old Extra Prop Name	string	Txt
Old Extra Internal Id	uint	0
Old Extra String	string	Txt
Unknown Byte	byte	0
Extra Data	Ref<NiExt...	None
Num Extra Data List	uint	0
Extra Data List	Ref<NiExt...	
Controller	Ref<NiTi...	None
Flags	Flags	1
Shader Type	BSShader...	SHADER_NOLIGHTING
Shader Flags	BSShader...	SF_Remappable_Textures | SF_ZBuffer_Test
Unknown Int 2	int	0
Envmap Scale	float	1.0000
Unknown Int 3	int	3
File Name	SizedString	textures\demo animation\demorotate.dds
Unknown Float 2	float	1.0000
Unknown Float 3	float	0.0000
Unknown Float 4	float	1.0000
Unknown Float 5	float	0.0000

Select SHADER_NOLIGTHING

Make sure that this is set to Zero.

F. Adjusting the settings of the NiTexturingProperty.

- Set the Flag to 4.
- Set the Texture Count to 9
- Expand Base Texture, then set the **Has Texture Transform** to Yes. This is the main switch allowing animation.



Block List

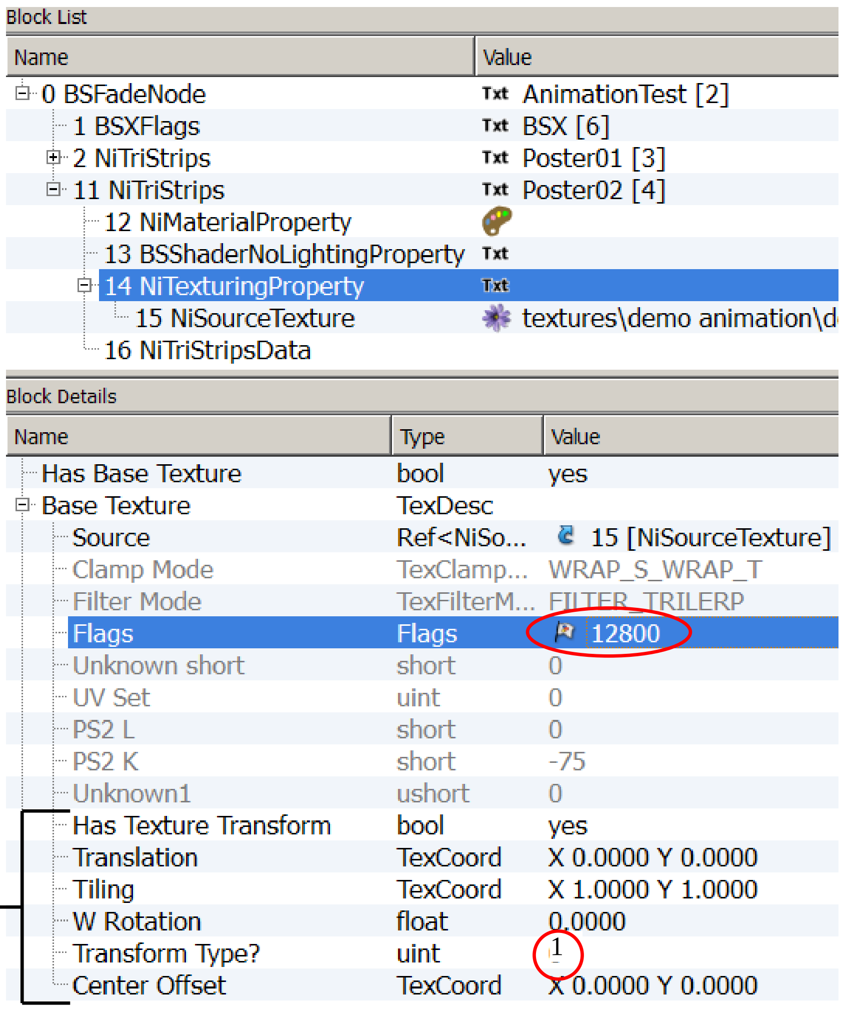
Name	Value
0 BSFadeNode	AnimationTest [2]
1 BSXFlags	BSX [6]
2 NiTriStrips	Poster01 [3]
11 NiTriStrips	Poster02 [4]
12 NiMaterialProperty	
13 BSShaderNoLightingProperty	
14 NiTexturingProperty	
15 NiSourceTexture	textures\demo animation\d

Block Details

Name	Type	Value
Skyrim Shader Type	BSLighting...	Default
Name	string	Txt
Has Old Extra Data	bool	no
Old Extra Prop Name	string	Txt
Old Extra Internal Id	uint	0
Old Extra String	string	Txt
Unknown Byte	byte	0
Extra Data	Ref<NiExt...	None
Num Extra Data List	uint	0
Extra Data List	Ref<NiExt...	
Controller	Ref<NiTi...	None
Flags	Flags	0
Flags	Flags	4
Apply Mode	ApplyMode	APPLY_MODULATE
Texture Count	uint	9
Has Base Texture	bool	yes
Base Texture	TexDesc	
Source	Ref<NiSo...	15 [NiSourceTexture]
Clamp Mode	TexClamp...	WRAP_S_WRAP_T
Filter Mode	TexFilterM...	FILTER_TRILERP
Flags	Flags	0
Unknown short	short	0
UV Set	uint	0
PS2 L	short	0
PS2 K	short	-75
Unknown1	ushort	0
Has Texture Transform	bool	yes
Translation	TexCoord	X 0.0000 Y 0.0000

A series of settings have become available inside the Has Texture Transform section.

- Set Flags to 12800 and Transform Type? to 1.



Block List

Name	Value
0 BSFadeNode	Txt AnimationTest [2]
1 BSXFlags	Txt BSX [6]
2 NiTriStrips	Txt Poster01 [3]
11 NiTriStrips	Txt Poster02 [4]
12 NiMaterialProperty	
13 BSShaderNoLightingProperty	Txt
14 NiTexturingProperty	Txt
15 NiSourceTexture	textures\demo animation\d
16 NiTriStripsData	

Block Details

Name	Type	Value
Has Base Texture	bool	yes
Base Texture	TexDesc	
Source	Ref<NiSo...	15 [NiSourceTexture]
Clamp Mode	TexClamp...	WRAP_S_WRAP_T
Filter Mode	TexFilterM...	FILTER_TRILERP
Flags	Flags	12800
Unknown short	short	0
UV Set	uint	0
PS2 L	short	0
PS2 K	short	-75
Unknown1	ushort	0
Has Texture Transform	bool	yes
Translation	TexCoord	X 0.0000 Y 0.0000
Tiling	TexCoord	X 1.0000 Y 1.0000
W Rotation	float	0.0000
Transform Type?	uint	1
Center Offset	TexCoord	X 0.0000 Y 0.0000

These settings allow performing operations on the UV Map.

These settings replace conveniently the operations that can be made in the UV Editor, as they can be modified and reset any time. We will leave them default, apart of the **Transform Type?** which we would set on 1. Occasionally we'll change some settings to match our need.

Translation	Move the UV to the left or right XY=UV
Tiling	Resize the UV Map
W Rotation	Rotate it (In degree)
Transform Type?	0 = Reversed, 1 = Normal, 2 = Flipped UV
Center Offset	Determine the center of the UV Map

It is good to save and have a check in Geck Preview to be sure that the base for animations is healthy.

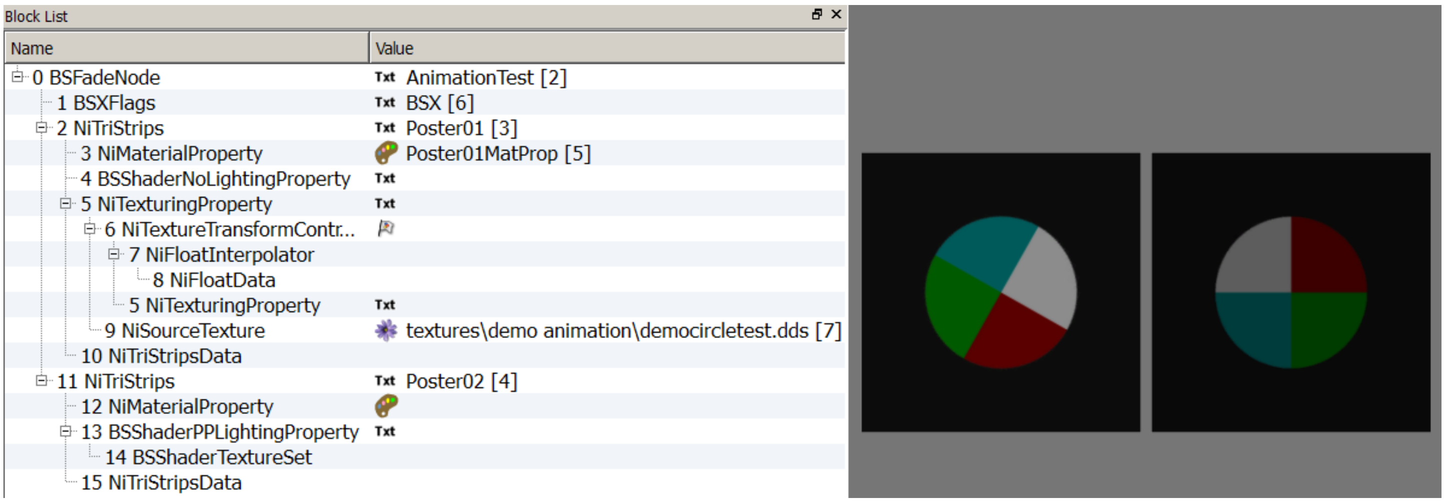
These previous steps need to be made **each time** a mesh doesn't have a BSShaderNoLightingProperty.

ROTATION TEXTURE ANIMATION

STEP BY STEP

Open 01DemoRotate in Geck preview for a quick view.

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



Block List

Name	Value
0 BSFadeNode	Txt AnimationTest [2]
1 BSXFlags	Txt BSX [6]
2 NiTriStrips	Txt Poster01 [3]
3 NiMaterialProperty	Poster01MatProp [5]
4 BSShaderNoLightingProperty	Txt
5 NiTexturingProperty	Txt
6 NiTextureTransformContr...	
7 NiFloatInterpolator	
8 NiFloatData	
5 NiTexturingProperty	Txt
9 NiSourceTexture	textures\demo animation\democircletest.dds [7]
10 NiTriStripsData	
11 NiTriStrips	Txt Poster02 [4]
12 NiMaterialProperty	
13 BSShaderPPLightingProperty	Txt
14 BSShaderTextureSet	
15 NiTriStripsData	

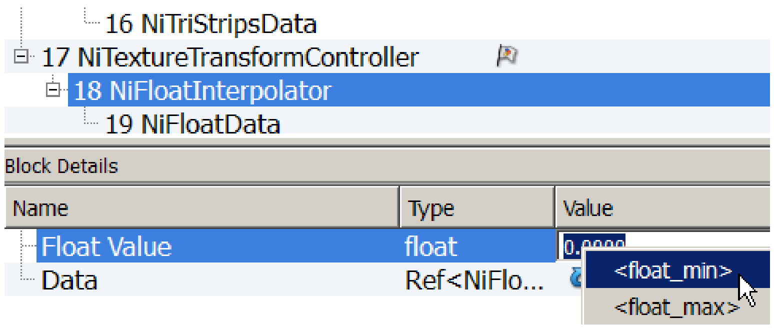
2. Once you have set the correct Shader in poster02 by following the steps described in Shader Preparatory Work, you can:

3. Rename the Property you will work on, the NiTexturingProperty. (Click on **Txt** in the Value field to access the string pop up window.)

4. Insert a NiTextureTransformController, a NiFloatInterpolator and a NiFloatData.

5. Link the three elements to make a texture Animation Block. *(As seen previously in detail in Alpha Material Animation – Step By Step, in Part 2 of this tutorial.)*
   - In the Interpolator, set the value of the Data with the value of the NiFloatData.
   - In the Controller, set the value of the Interpolator with the value of the NiFloatInterpolator.

6. Adjust the setting of the NiFloatInterpolator.
   - Set it to <float_min>



16 NiTriStripsData
17 NiTextureTransformController
18 NiFloatInterpolator
19 NiFloatData

Block Details

Name	Type	Value
Float Value	float	0.0000
Data	Ref<NiFlo...	

7. Adjust the settings of the NiTextureTransformController.

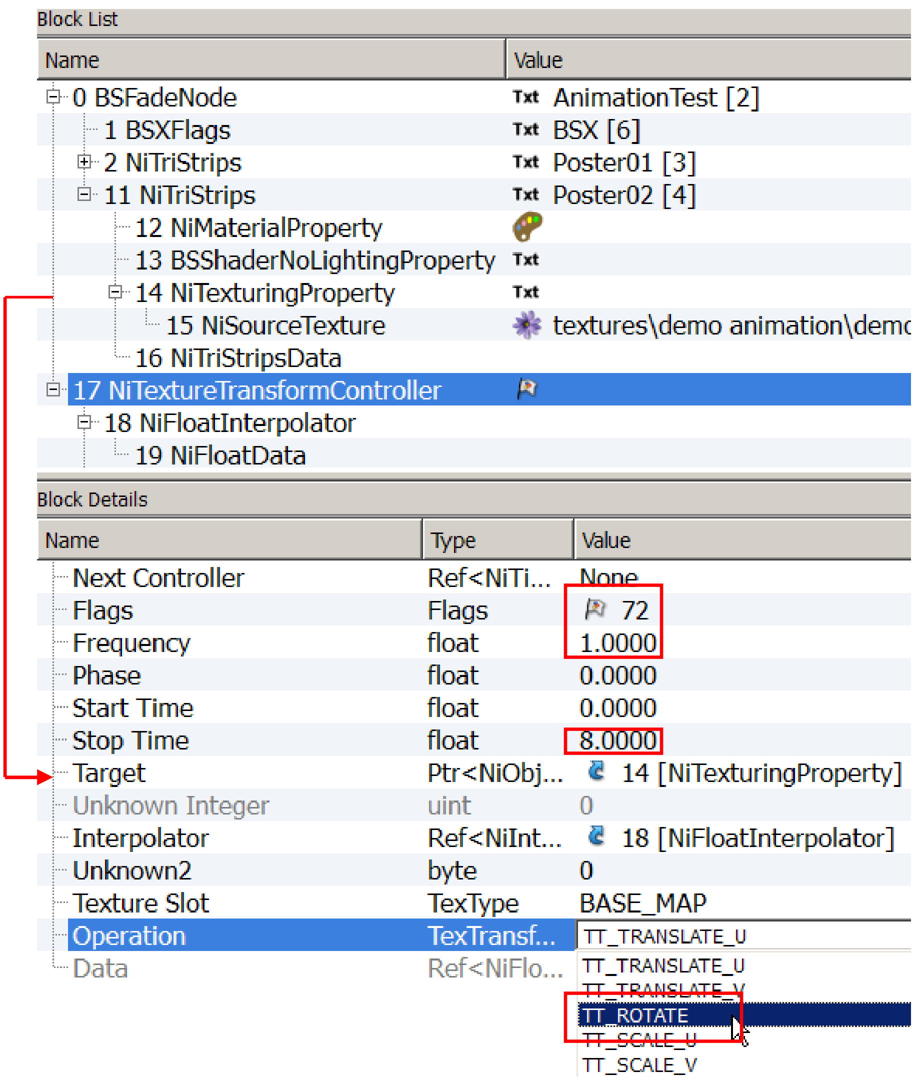
7 A. Flags to 72.

7 B. Frequency to 1.

7 C. Stop time 8 (length of the animation).

7 D. Target the NiTexturingProperty that need to be animated.

7 E. Select TT_ROTATE in the Operation Values.



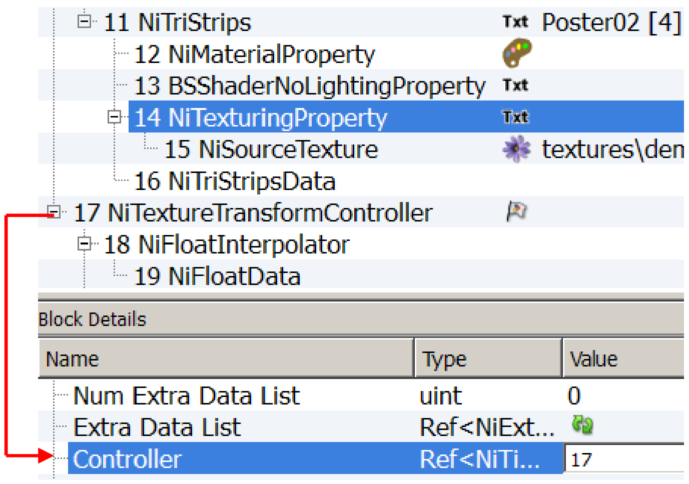
Block List

Name	Value
0 BSFadeNode	Txt AnimationTest [2]
1 BSXFlags	Txt BSX [6]
2 NiTriStrips	Txt Poster01 [3]
11 NiTriStrips	Txt Poster02 [4]
12 NiMaterialProperty	
13 BSShaderNoLightingProperty	Txt
14 NiTexturingProperty	Txt
15 NiSourceTexture	textures\demo animation\demo
16 NiTriStripsData	
17 NiTextureTransformController	
18 NiFloatInterpolator	
19 NiFloatData	

Block Details

Name	Type	Value
Next Controller	Ref<NiTi...	None
Flags	Flags	72
Frequency	float	1.0000
Phase	float	0.0000
Start Time	float	0.0000
Stop Time	float	8.0000
Target	Ptr<NiObj...	14 [NiTexturingProperty]
Unknown Integer	uint	0
Interpolator	Ref<NiInt...	18 [NiFloatInterpolator]
Unknown2	byte	0
Texture Slot	TexType	BASE_MAP
Operation	TexTransf...	TT_TRANSLATE_U
Data	Ref<NiFlo...	

8. Finalize the linkage, by telling the NiTexturingProperty who is its controller.



Name	Value
11 NiTriStrips	Txt Poster02 [4]
12 NiMaterialProperty	
13 BSShaderNoLightingProperty	Txt
14 NiTexturingProperty	Txt
15 NiSourceTexture	textures\den
16 NiTriStripsData	
17 NiTextureTransformController	
18 NiFloatInterpolator	
19 NiFloatData	

Block Details

Name	Type	Value
Num Extra Data List	uint	0
Extra Data List	Ref<NiExt...	
Controller	Ref<NiTi...	17

Time to save your work.

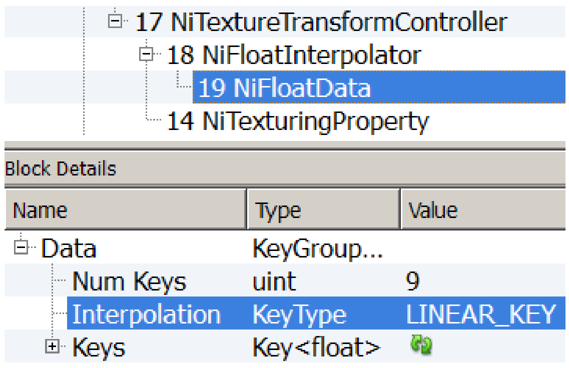
The rotation values aren't expressed in angle-degree, but in number values instead. In my experiment, I've found the following:

- 1.5709 = 90°
- 3.1418 = 180°
- 4.7127 = 270°
- 6.2836 = 360° or full rotation

9. Setting the Keys. Select the NiFloatData and in Block Details:

9 A. Insert the number of Keys needed (9). Update.

9 B. Select the Interpolation mode (LINEAR_KEY).



17 NiTextureTransformController
18 NiFloatInterpolator
19 NiFloatData
14 NiTexturingProperty

Block Details

Name	Type	Value
Data	KeyGroup...	
Num Keys	uint	9
Interpolation	KeyType	LINEAR_KEY
Keys	Key<float>	

- And set the Keys as followed.

Key	Time	0.0000
	Value	0.0000
Key	Time	1.0000
	Value	0.0000
Key	Time	2.0000
	Value	1.5709
Key	Time	3.0000
	Value	1.5709
Key	Time	4.0000
	Value	3.1418
Key	Time	5.0000
	Value	3.1418
Key	Time	6.0000
	Value	4.7127
Key	Time	7.0000
	Value	4.7127
Key	Time	8.0000
	Value	6.2836

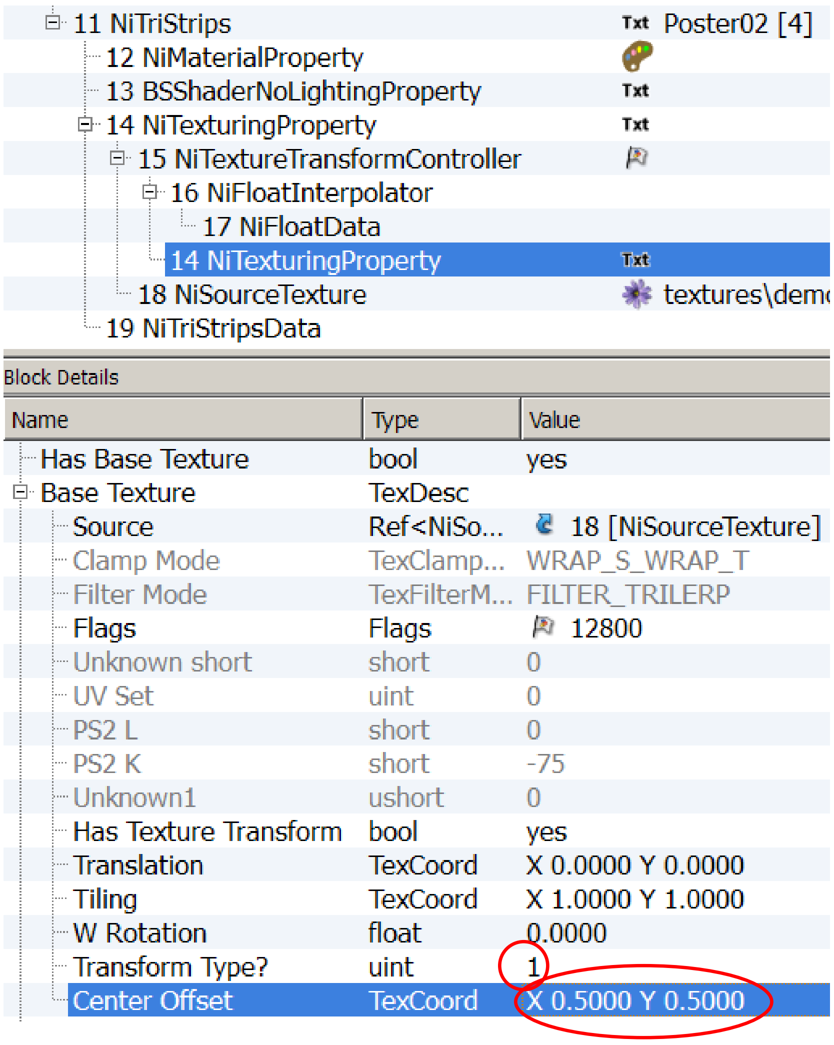
If you have saved and checked your work in Geck preview, as it is advisable at this stage, you've noticed that the animation is not centred.

There is one last step to perform to have the UV Map centred in the middle.

10. In the NiTexturingProperty, adjust settings of the **Base Texture** section.

10 A. Make sure the Transform Type? is set to 1.

10 B. Center Offset set to X 0.5000, Y 0.5000



```
11 NiTriStrips                          Txt Poster02 [4]
  12 NiMaterialProperty
  13 BSShaderNoLightingProperty         Txt
  14 NiTexturingProperty                Txt
    15 NiTextureTransformController
      16 NiFloatInterpolator
        17 NiFloatData
      14 NiTexturingProperty            Txt
    18 NiSourceTexture                  textures\dem
  19 NiTriStripsData
```

Block Details

Name	Type	Value
Has Base Texture	bool	yes
Base Texture	TexDesc	
Source	Ref<NiSo...	18 [NiSourceTexture]
Clamp Mode	TexClamp...	WRAP_S_WRAP_T
Filter Mode	TexFilterM...	FILTER_TRILERP
Flags	Flags	12800
Unknown short	short	0
UV Set	uint	0
PS2 L	short	0
PS2 K	short	-75
Unknown1	ushort	0
Has Texture Transform	bool	yes
Translation	TexCoord	X 0.0000 Y 0.0000
Tiling	TexCoord	X 1.0000 Y 1.0000
W Rotation	float	0.0000
Transform Type?	uint	1
Center Offset	TexCoord	X 0.5000 Y 0.5000

The animation should work as expected now. The texture is designed to give the illusion that only the circle in the center is moving, but, of course, the whole texture is rotating. Assign another texture to see what I mean.

Poster01 and Poster 02 are rotating differently, examine how.

I suggest that you play a bit with different settings (Transform Type?, Interpolation Mode, Keys Time & Values, etc.). Possibilities are endless…

Translation U Texture Animation

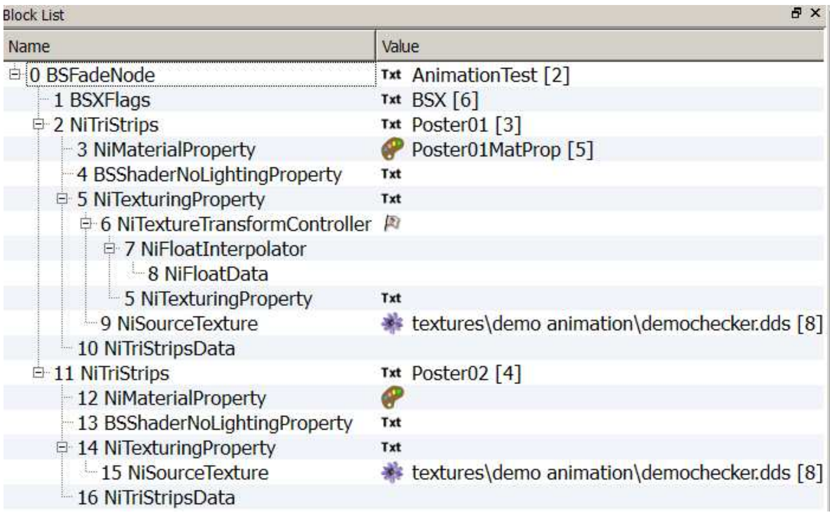
Step by Step

This will demonstrate how to realise a translation animation on the U Axis (horizontally).

Most of the steps are similar to the previous animation. These steps will be listed solely, only the new or specific ones will be illustrated.

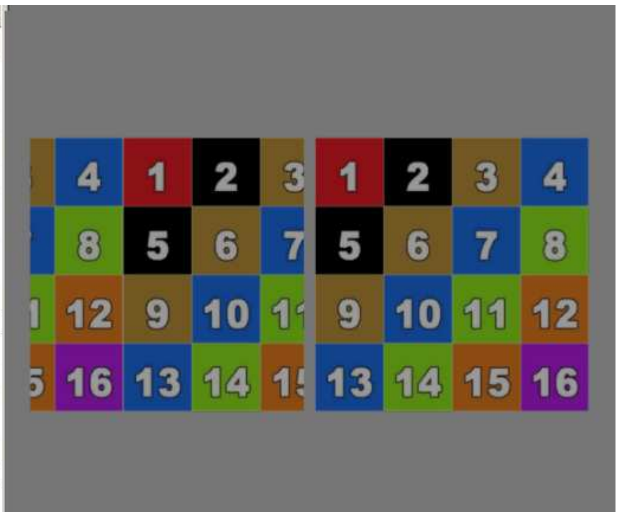
Open 02DemoTranslate-U in Geck preview for a quick view.

1. 02DemoTranslate-U in NifSkope and expand all the arrays. Examine the differences between the two posters.



Block List

Name	Value
0 BSFadeNode	Txt AnimationTest [2]
1 BSXFlags	Txt BSX [6]
2 NiTriStrips	Txt Poster01 [3]
3 NiMaterialProperty	Poster01MatProp [5]
4 BSShaderNoLightingProperty	Txt
5 NiTexturingProperty	Txt
6 NiTextureTransformController	
7 NiFloatInterpolator	
8 NiFloatData	
5 NiTexturingProperty	Txt
9 NiSourceTexture	textures\demo animation\demochecker.dds [8]
10 NiTriStripsData	
11 NiTriStrips	Txt Poster02 [4]
12 NiMaterialProperty	
13 BSShaderNoLightingProperty	Txt
14 NiTexturingProperty	Txt
15 NiSourceTexture	textures\demo animation\demochecker.dds [8]
16 NiTriStripsData	



You've noticed that Translate Animations are played reversed in NifSkope.

2. BSShaderNoLightingProperty is already set, so you save some work.

3. You have to rename the NiTexturingProperty, tough.

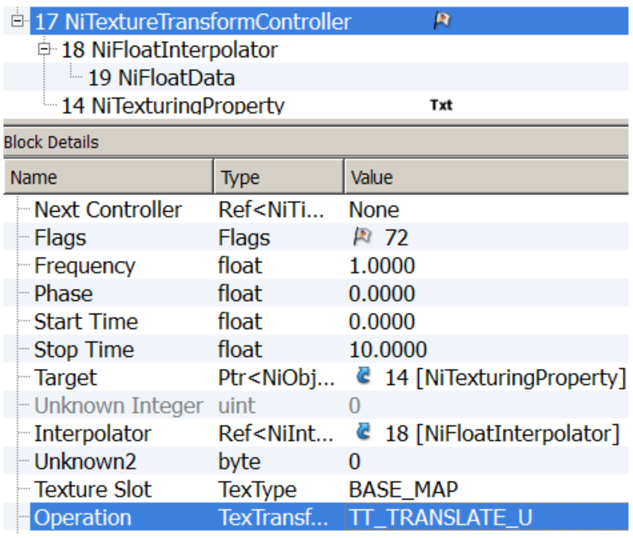
4. Insert a NiTextureTransformController, a NiFloatInterpolator and a NiFloatData.

5. Link the three elements to make a texture Animation Block.
   - In the Interpolator, set the value of the Data with the value of the NiFloatData.
   - In the Controller, set the value of the Interpolator with the value of the NiFloatInterpolator.

6. Adjust the setting of the NiFloatInterpolator.
   - Set it to <float_min>

7. Adjust the settings of the NiTextureTransformController.
   - Flags to 72.
   - Frequency to 1.
   - Stop time 10 (length of the animation).
   - Target the NiTexturingProperty that need to be animated.
   - Make sure TT_TRANSLATE_U Is selected in the Operation Values.

NiTextureTransformController settings



17 NiTextureTransformController
18 NiFloatInterpolator
19 NiFloatData
14 NiTexturingProperty Txt

Block Details

Name	Type	Value
Next Controller	Ref<NiTi...	None
Flags	Flags	72
Frequency	float	1.0000
Phase	float	0.0000
Start Time	float	0.0000
Stop Time	float	10.0000
Target	Ptr<NiObj...	14 [NiTexturingProperty]
Unknown Integer	uint	0
Interpolator	Ref<NiInt...	18 [NiFloatInterpolator]
Unknown2	byte	0
Texture Slot	TexType	BASE_MAP
Operation	TexTransf...	TT_TRANSLATE_U

8. Finalize the linkage, by telling the NiTexturingProperty who is his controller. Save and test your work.

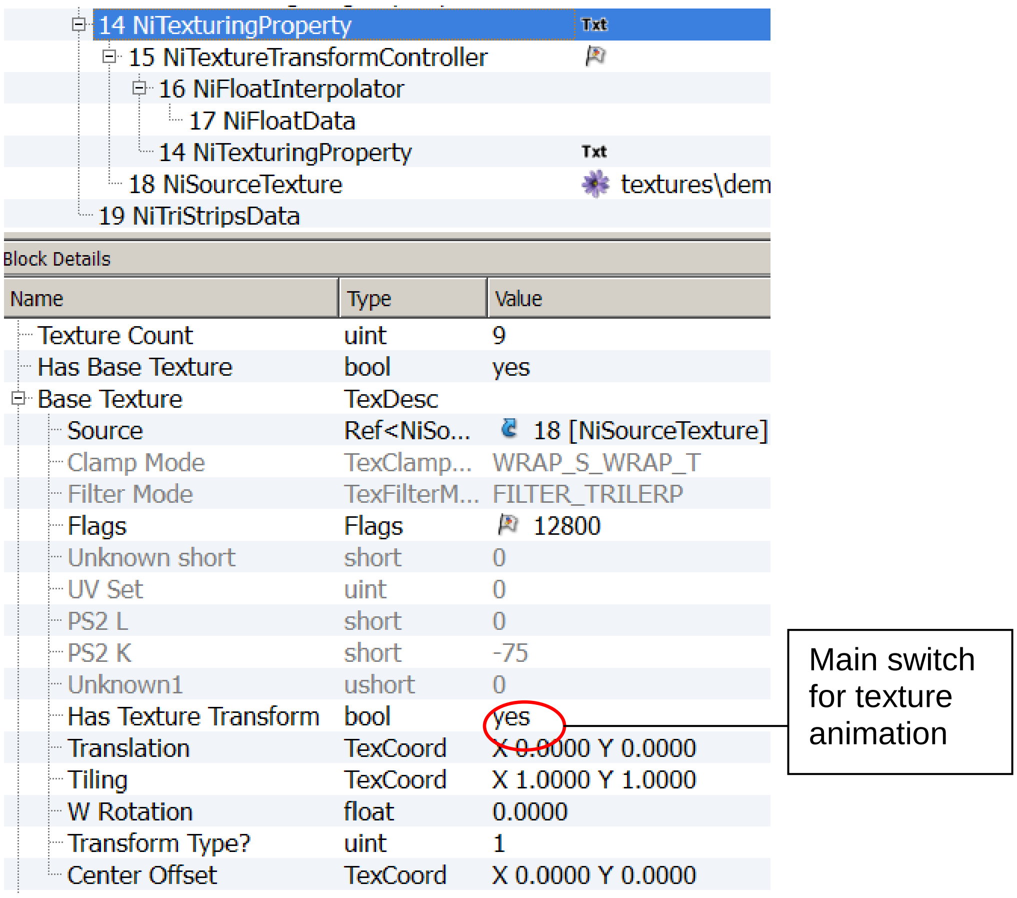
With **Transform Type?** (in NiTexturingProperty's Base Texture) set to 1, a key Value of 0.2500 will move the texture 25% to the right. If set to -0.2500, it'll move to the left. Set to 0 will be the other way around.

We want to have the texture moving to the right, by quarter section, with 1 second duration for the translation and 1 second of pause in between. Pause for the start is a little longer.

9. Setting the Keys. Select the NiFloatData, expand Data.
9 A. Insert the number of Keys needed (9). Update.
9 B. Select the Interpolation mode (LINEAR_KEY).
9 C. And set the Keys as followed.

Key	Time	0.0000
	Value	0.0000
Key	Time	1.0000
	Value	0.2500
Key	Time	2.0000
	Value	0.2500
Key	Time	3.0000
	Value	0.5000
Key	Time	4.0000
	Value	0.5000
Key	Time	5.0000
	Value	0.7500
Key	Time	6.0000
	Value	0.7500
Key	Time	7.0000
	Value	1.0000
Key10	Time	10.0000
	Value	0.0000

10. Check the Base Texture settings. Center Offset doesn't seem to matter here.



- 14 NiTexturingProperty — Txt
  - 15 NiTextureTransformController
    - 16 NiFloatInterpolator
      - 17 NiFloatData
    - 14 NiTexturingProperty — Txt
  - 18 NiSourceTexture — textures\dem
- 19 NiTriStripsData

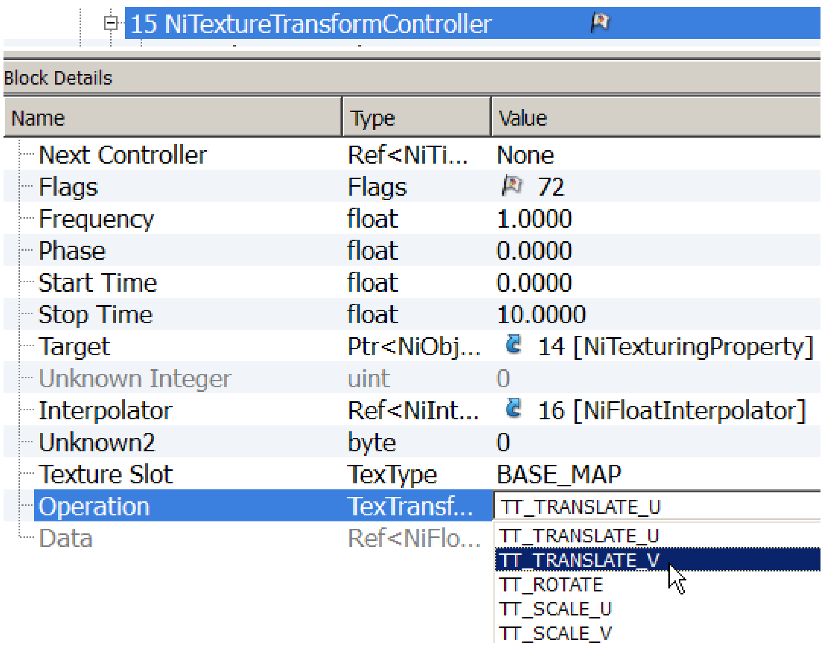
Block Details

Name	Type	Value
Texture Count	uint	9
Has Base Texture	bool	yes
Base Texture	TexDesc	
Source	Ref<NiSo...	18 [NiSourceTexture]
Clamp Mode	TexClamp...	WRAP_S_WRAP_T
Filter Mode	TexFilterM...	FILTER_TRILERP
Flags	Flags	12800
Unknown short	short	0
UV Set	uint	0
PS2 L	short	0
PS2 K	short	-75
Unknown1	ushort	0
Has Texture Transform	bool	yes
Translation	TexCoord	X 0.0000 Y 0.0000
Tiling	TexCoord	X 1.0000 Y 1.0000
W Rotation	float	0.0000
Transform Type?	uint	1
Center Offset	TexCoord	X 0.0000 Y 0.0000

Save and preview your work.

There's a slight setting difference between Poster01 & 02. See if you spot it…

Building a Translation Animation on the V Axis (vertically) would follow the exact same steps, except for the step **7** where you will select TT_TRANSLATE_V in the Operation Values.



15 NiTextureTransformController

Block Details

Name	Type	Value
Next Controller	Ref<NiTi...	None
Flags	Flags	72
Frequency	float	1.0000
Phase	float	0.0000
Start Time	float	0.0000
Stop Time	float	10.0000
Target	Ptr<NiObj...	14 [NiTexturingProperty]
Unknown Integer	uint	0
Interpolator	Ref<NiInt...	16 [NiFloatInterpolator]
Unknown2	byte	0
Texture Slot	TexType	BASE_MAP
Operation	TexTransf...	TT_TRANSLATE_U
Data	Ref<NiFlo...	

Having both UV Axis moving simultaneously can display interesting effects, in case of Translate or Scale Animations

TRANSLATE UV TEXTURE ANIMATION

STEP BY STEP

This will demonstrate how to realise a short film sequence (Gif type) through a Translation Animation on the UV Axis (horizontally & vertically).

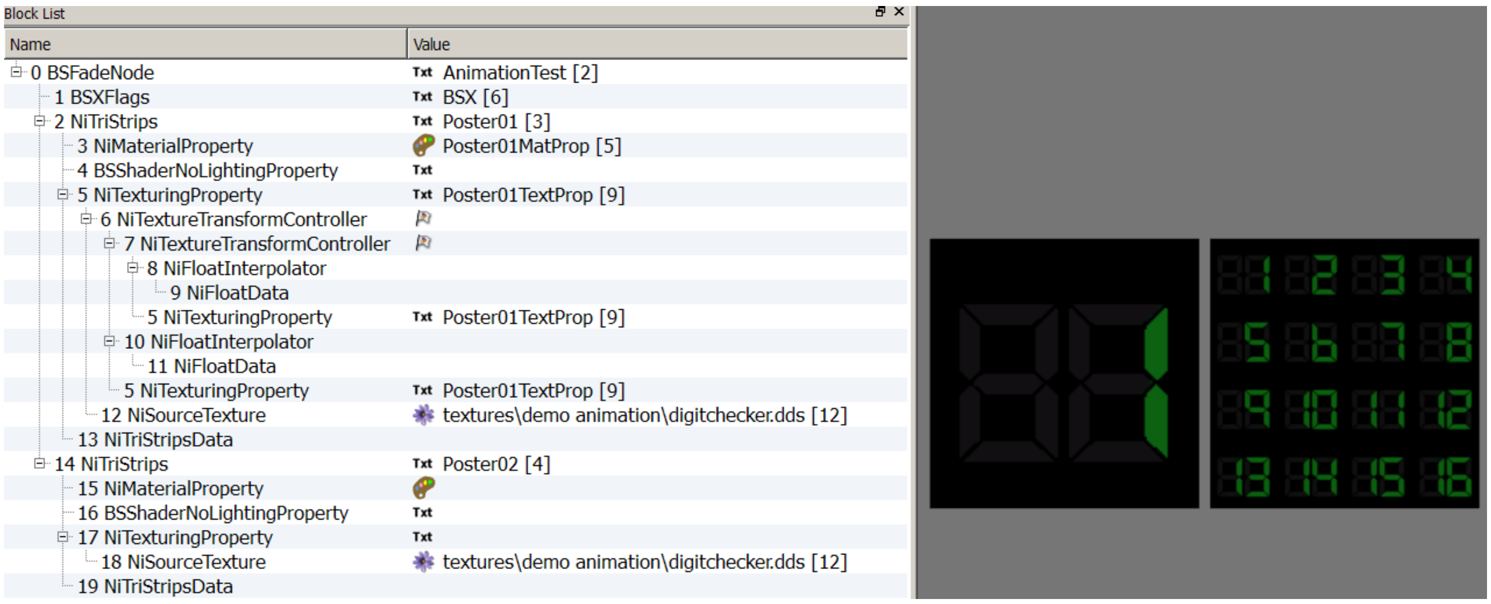
The texture is divided in 16 equal parts. Each part will be equivalent to a frame.

In this demo, each frame will be displayed for a half a second.

Again, NifSkope preview isn't representative on how thing will look in game. Geck gives a better idea.

Open 02DemoTranslate-UV in Geck preview for a quick view.

1. 02DemoTranslate-UV in NifSkope and expand all the arrays. Examine the differences between the two posters.



Block List

Name	Value
0 BSFadeNode	Txt AnimationTest [2]
1 BSXFlags	Txt BSX [6]
2 NiTriStrips	Txt Poster01 [3]
3 NiMaterialProperty	Poster01MatProp [5]
4 BSShaderNoLightingProperty	Txt
5 NiTexturingProperty	Txt Poster01TextProp [9]
6 NiTextureTransformController	
7 NiTextureTransformController	
8 NiFloatInterpolator	
9 NiFloatData	
5 NiTexturingProperty	Txt Poster01TextProp [9]
10 NiFloatInterpolator	
11 NiFloatData	
5 NiTexturingProperty	Txt Poster01TextProp [9]
12 NiSourceTexture	textures\demo animation\digitchecker.dds [12]
13 NiTriStripsData	
14 NiTriStrips	Txt Poster02 [4]
15 NiMaterialProperty	
16 BSShaderNoLightingProperty	Txt
17 NiTexturingProperty	Txt
18 NiSourceTexture	textures\demo animation\digitchecker.dds [12]
19 NiTriStripsData	

No Shader preparatory work, it's all set.

Let's make some adjustments on the NiTexturingProperty.

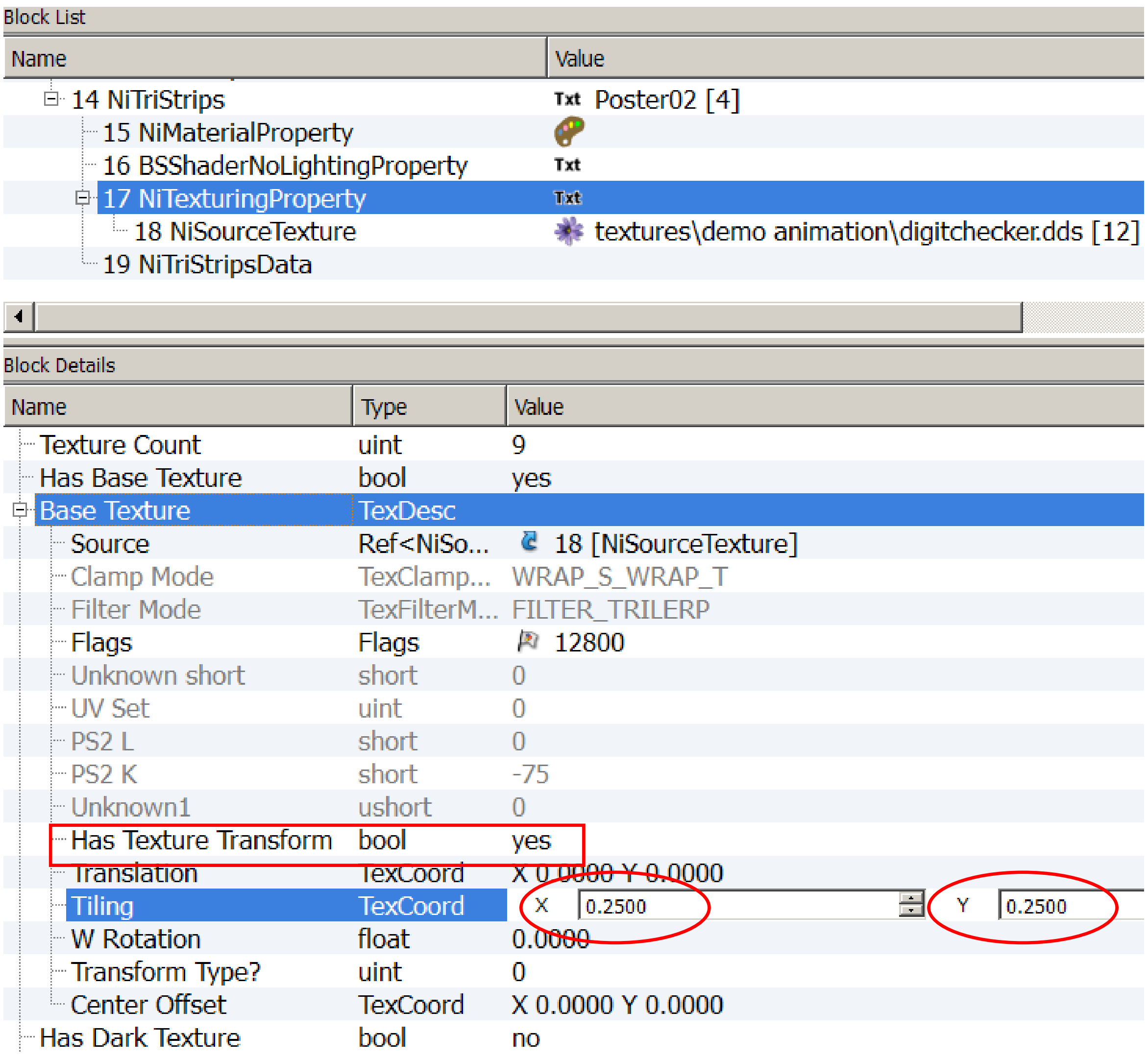
2. Rename the NiTexturingProperty.

Since each frame or part of the texture is a quarter the size of the global texture, we will have to resize the UV Map displayed.

Two way of doing:
By resizing the UV Map itself (*as described in Generalities - Part 1 of this tutorial*)
Or
By adjusting settings in NiTexturingProperty in a non destructive way.

To reduce the UV map to a quarter of its size:

3. Select the NiTexturingProperty, expand the Base Texture tree and make sure Has Texture Transform is on yes and set Tiling to 0.2500.


Block List

Name	Value
14 NiTriStrips	Txt Poster02 [4]
15 NiMaterialProperty	
16 BSShaderNoLightingProperty	Txt
17 NiTexturingProperty	Txt
18 NiSourceTexture	textures\demo animation\digitchecker.dds [12]
19 NiTriStripsData	

Block Details

Name	Type	Value
Texture Count	uint	9
Has Base Texture	bool	yes
Base Texture	TexDesc	
Source	Ref<NiSo...	18 [NiSourceTexture]
Clamp Mode	TexClamp...	WRAP_S_WRAP_T
Filter Mode	TexFilterM...	FILTER_TRILERP
Flags	Flags	12800
Unknown short	short	0
UV Set	uint	0
PS2 L	short	0
PS2 K	short	-75
Unknown1	ushort	0
Has Texture Transform	bool	yes
Translation	TexCoord	X 0.0000 Y 0.0000
Tiling	TexCoord	X 0.2500 Y 0.2500
W Rotation	float	0.0000
Transform Type?	uint	0
Center Offset	TexCoord	X 0.0000 Y 0.0000
Has Dark Texture	bool	no

Other settings will be left on zero.

To have both UV axes moving, you'll need two Animation Blocks, one for each Translation.

So we will build an Animation Block as usual, duplicate it at a certain point, adjust some more settings, finalize the linkage and add the keys.

4. Insert a NiTextureTransformController, a NiFloatInterpolator and a NiFloatData.

5. Link them into a Block.

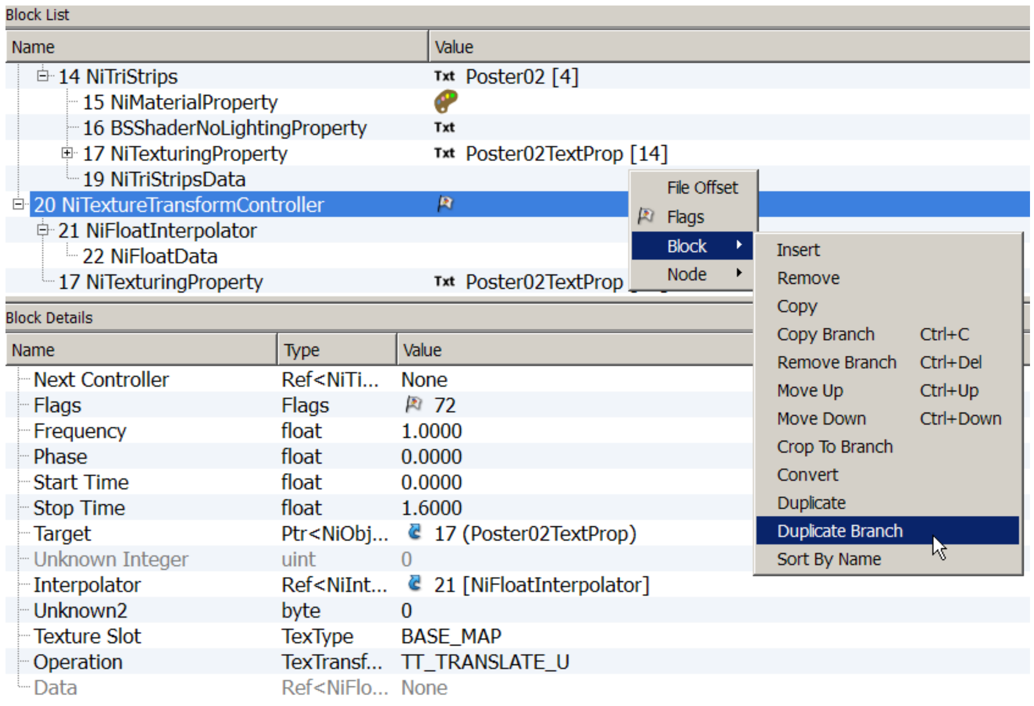
6. Adjust the setting of the NiFloatInterpolator.
   - Set it to <float_min>

7. Adjust the settings of the NiTextureTransformController.
   - Flags to 72.
   - Frequency to 1.
   - Stop time 8 (length of the animation).
   - Target the NiTexturingProperty that need to be animated.

Make sure the NiTexturingProperty is properly named.

Once you have a workable Animation Block.

8. Select the Controller and Duplicate Branch.



Block List

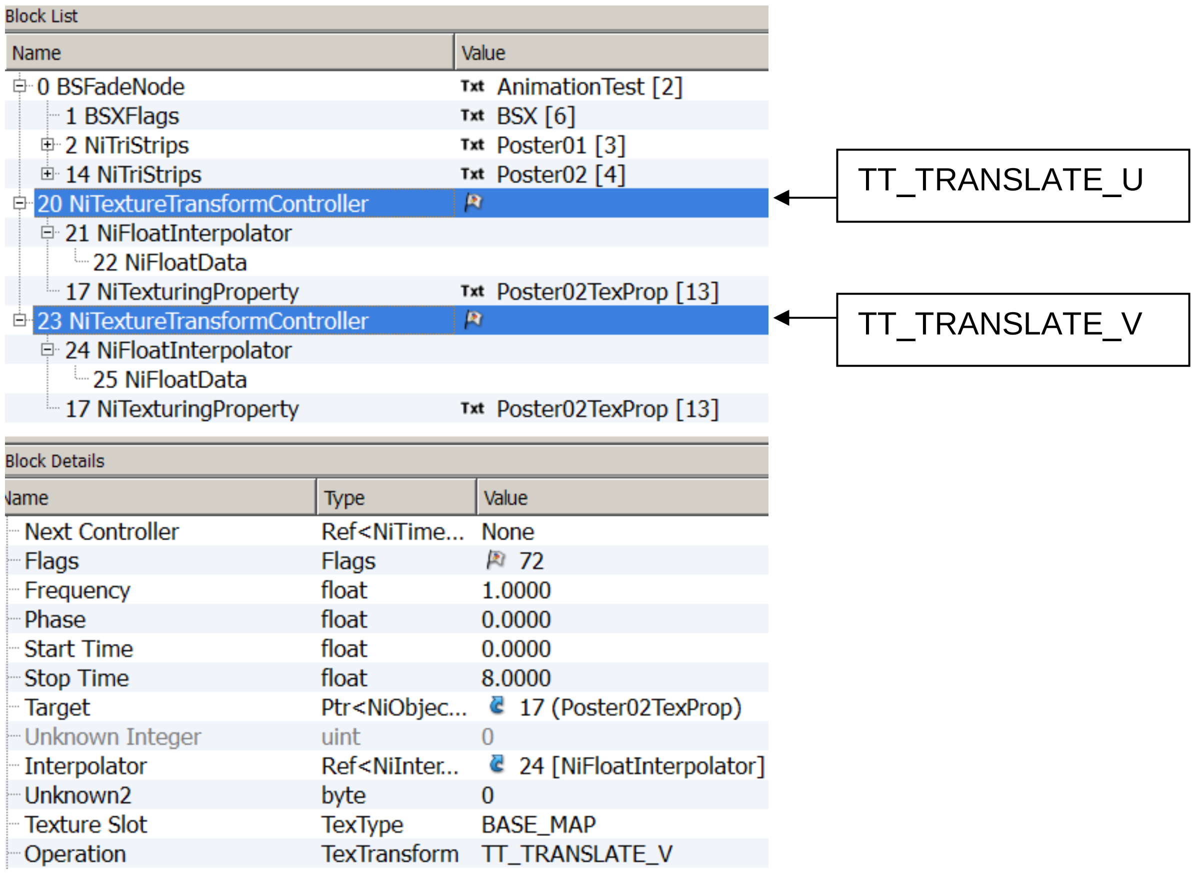
Name	Value
14 NiTriStrips	Txt Poster02 [4]
15 NiMaterialProperty	
16 BSShaderNoLightingProperty	Txt
17 NiTexturingProperty	Txt Poster02TextProp [14]
19 NiTriStripsData	
20 NiTextureTransformController	
21 NiFloatInterpolator	
22 NiFloatData	
17 NiTexturingProperty	Txt Poster02TextProp

Block Details

Name	Type	Value
Next Controller	Ref<NiTi...	None
Flags	Flags	72
Frequency	float	1.0000
Phase	float	0.0000
Start Time	float	0.0000
Stop Time	float	1.6000
Target	Ptr<NiObj...	17 (Poster02TextProp)
Unknown Integer	uint	0
Interpolator	Ref<NiInt...	21 [NiFloatInterpolator]
Unknown2	byte	0
Texture Slot	TexType	BASE_MAP
Operation	TexTransf...	TT_TRANSLATE_U
Data	Ref<NiFlo...	None

You should now have two distinct Animation Blocks with identical settings.

9. Make sure the first NiTextureTransformController Operation is set to TT_TRANSLATE_U Set the second NiTextureTransformController Operation to TT_TRANSLATE_V.



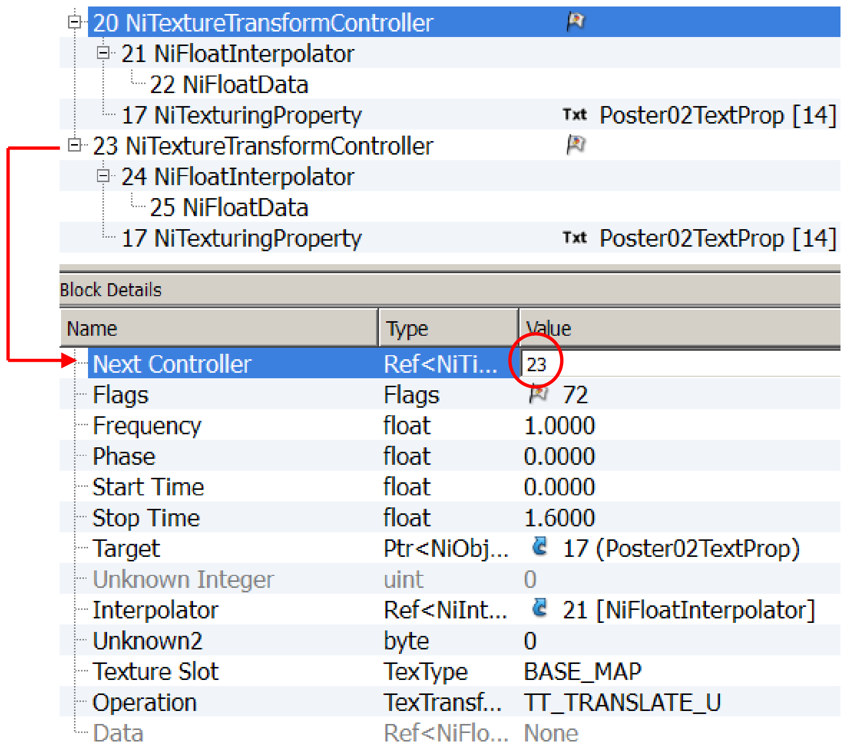
Block List

Name	Value
0 BSFadeNode	Txt AnimationTest [2]
1 BSXFlags	Txt BSX [6]
2 NiTriStrips	Txt Poster01 [3]
14 NiTriStrips	Txt Poster02 [4]
20 NiTextureTransformController	
21 NiFloatInterpolator	
22 NiFloatData	
17 NiTexturingProperty	Txt Poster02TexProp [13]
23 NiTextureTransformController	
24 NiFloatInterpolator	
25 NiFloatData	
17 NiTexturingProperty	Txt Poster02TexProp [13]

Block Details

Name	Type	Value
Next Controller	Ref<NiTime...	None
Flags	Flags	72
Frequency	float	1.0000
Phase	float	0.0000
Start Time	float	0.0000
Stop Time	float	8.0000
Target	Ptr<NiObjec...	17 (Poster02TexProp)
Unknown Integer	uint	0
Interpolator	Ref<NiInter...	24 [NiFloatInterpolator]
Unknown2	byte	0
Texture Slot	TexType	BASE_MAP
Operation	TexTransform	TT_TRANSLATE_V

10. Chain-link the two Controllers. Select the first NiTextureTransformController and set the Value of the second one in the Next Controller field.

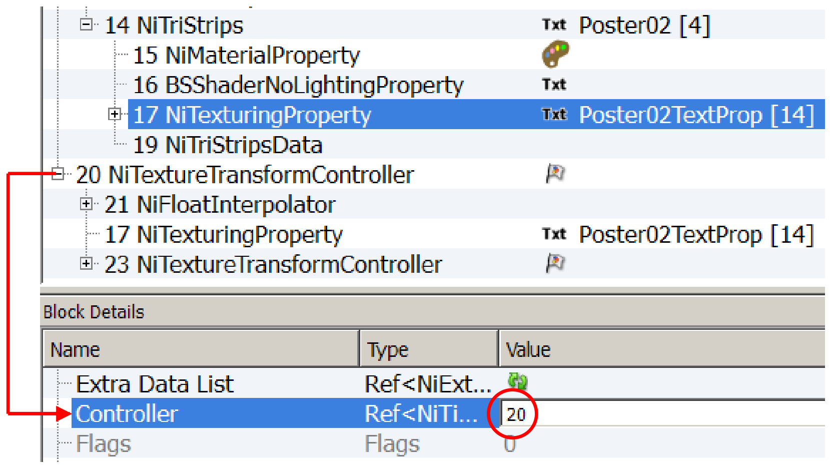


```
20 NiTextureTransformController
  21 NiFloatInterpolator
    22 NiFloatData
  17 NiTexturingProperty                Txt  Poster02TextProp [14]
23 NiTextureTransformController
  24 NiFloatInterpolator
    25 NiFloatData
  17 NiTexturingProperty                Txt  Poster02TextProp [14]
```

Block Details

Name	Type	Value
Next Controller	Ref<NiTi...	23
Flags	Flags	72
Frequency	float	1.0000
Phase	float	0.0000
Start Time	float	0.0000
Stop Time	float	1.6000
Target	Ptr<NiObj...	17 (Poster02TextProp)
Unknown Integer	uint	0
Interpolator	Ref<NiInt...	21 [NiFloatInterpolator]
Unknown2	byte	0
Texture Slot	TexType	BASE_MAP
Operation	TexTransf...	TT_TRANSLATE_U
Data	Ref<NiFlo...	None

11. Finalize the linkage by telling the NiTexturingProperty who is his main Controller (that would be the first in link).



```
14 NiTriStrips                          Txt  Poster02 [4]
  15 NiMaterialProperty
  16 BSShaderNoLightingProperty         Txt
  17 NiTexturingProperty                Txt  Poster02TextProp [14]
  19 NiTriStripsData
20 NiTextureTransformController
  21 NiFloatInterpolator
  17 NiTexturingProperty                Txt  Poster02TextProp [14]
  23 NiTextureTransformController
```

Block Details

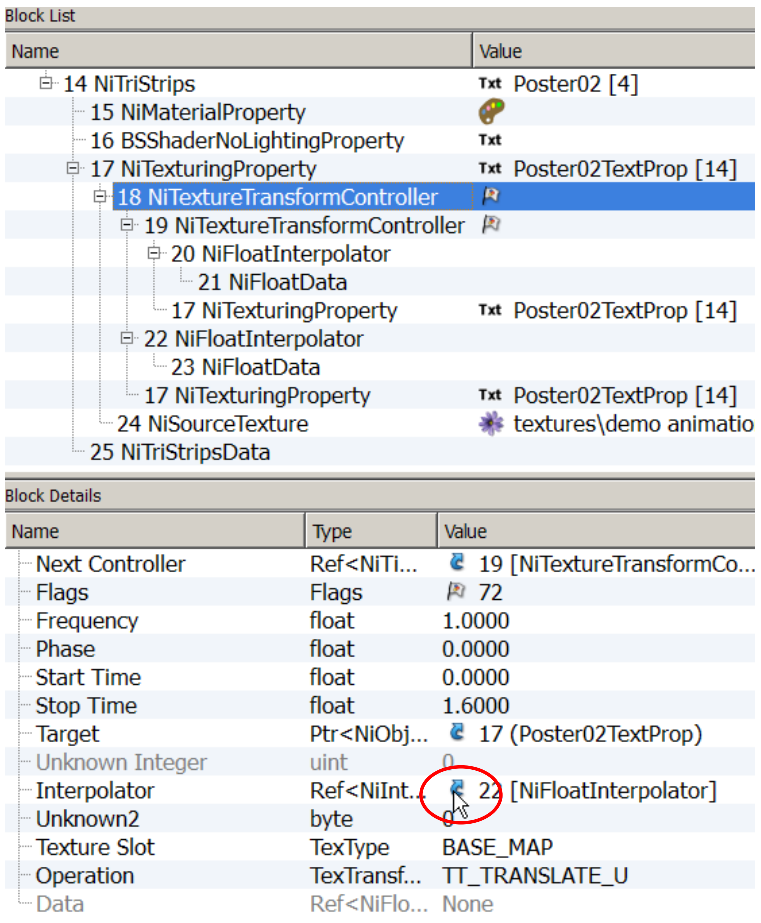
Name	Type	Value
Extra Data List	Ref<NiExt...	
Controller	Ref<NiTi...	20
Flags	Flags	0

Save to rearrange Blocks and check in Geck preview if everything is fine, so far.

The structure of both Posters should be identical. The only difference would be the keys settings.

12. Key Settings

Select the first NiTextureTransformController, the U one. Click the arrow of its interpolator.



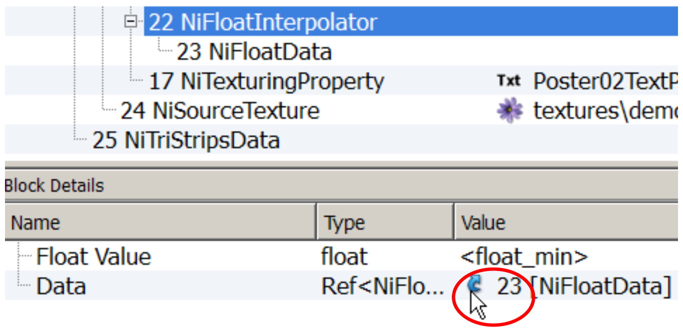
Block List

Name	Value
14 NiTriStrips	Txt Poster02 [4]
15 NiMaterialProperty	
16 BSShaderNoLightingProperty	Txt
17 NiTexturingProperty	Txt Poster02TextProp [14]
18 NiTextureTransformController	
19 NiTextureTransformController	
20 NiFloatInterpolator	
21 NiFloatData	
17 NiTexturingProperty	Txt Poster02TextProp [14]
22 NiFloatInterpolator	
23 NiFloatData	
17 NiTexturingProperty	Txt Poster02TextProp [14]
24 NiSourceTexture	textures\demo animatio
25 NiTriStripsData	

Block Details

Name	Type	Value
Next Controller	Ref<NiTi...	19 [NiTextureTransformCo...
Flags	Flags	72
Frequency	float	1.0000
Phase	float	0.0000
Start Time	float	0.0000
Stop Time	float	1.6000
Target	Ptr<NiObj...	17 (Poster02TextProp)
Unknown Integer	uint	0
Interpolator	Ref<NiInt...	22 [NiFloatInterpolator]
Unknown2	byte	0
Texture Slot	TexType	BASE_MAP
Operation	TexTransf...	TT_TRANSLATE_U
Data	Ref<NiFlo...	None

You will be transported to the NiFloatInterpolator, again, click the arrow of its Data. This is the safest way to end at the corresponding NiFloatData.



Name	Value
22 NiFloatInterpolator	
23 NiFloatData	
17 NiTexturingProperty	Txt Poster02TextP
24 NiSourceTexture	textures\demo
25 NiTriStripsData	

Block Details

Name	Type	Value
Float Value	float	<float_min>
Data	Ref<NiFlo...	23 [NiFloatData]

You'll need 17 keys for the U Translation

Similarly, find the NiFloatData corresponding to the V Translation, 5 keys only are needed.

The Interpolation is set on CONST_KEY for both Data.

Translate U Key Settings

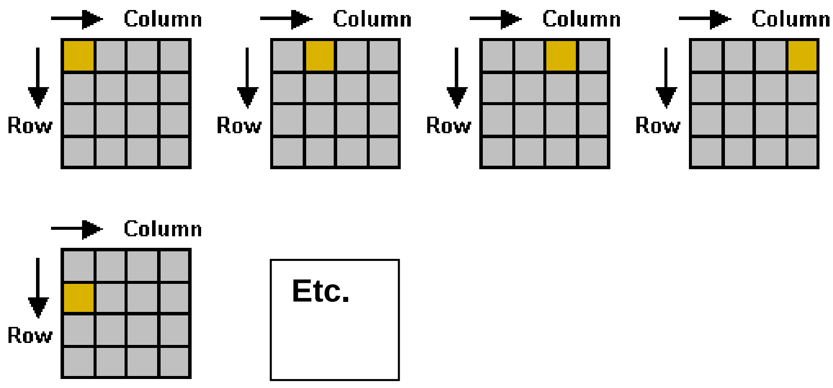
Key	Time	0.0000
	Value	0.0000
Key	Time	0.5
	Value	0.2500
Key	Time	1
	Value	0.5000
Key	Time	1.5
	Value	0.75
Key	Time	2
	Value	0.0000
Key	Time	2.5
	Value	0.2500
Key	Time	3
	Value	0.5000
Key	Time	3.5
	Value	0.7500
Key	Time	4
	Value	0.0000
Key	Time	4.5
	Value	0.2500
Key	Time	5
	Value	0.5000
Key	Time	5.5
	Value	0.7500
Key	Time	6
	Value	0.0000
Key	Time	6.5
	Value	0.2500
Key	Time	7
	Value	0.5000
Key	Time	7.5
	Value	0.7500
Key	Time	8
	Value	0.0000

Translate V Key Settings

Key	Time	0.0000
	Value	0.0000
Key	Time	2
	Value	0.2500
Key	Time	4
	Value	0.5000
Key	Time	6
	Value	0.75
Key	Time	8
	Value	0.0000

You can save and check your work in Geck Preview.

The horizontal U axis will act as column and the vertical V axis as row.



Our resized UV Map will move to the left, over the main Texture Map, column by column on the first row, then go to second row and start again, then to the third, to the fourth and back to the first, etc.

Each column and row is a quarter of the main Map.

In the illustration above the Values would be:

U = 0
V = 0

U = 0.25
V = 0

U = 0.5
V = 0

U = 0.75
V = 0

U = 0
V = 0.25

If you want more a “movie like” effect, ten frames per second is a minimum, twelve per second is better.

Have a look at 02DemoTranslate-UVMatrixExample.nif in Geck preview.

For “HD” animations you can go up to 25 fps.

16 frames at 10 fps is a 1.6 second animation.

If each frame is 128 x 128 pixels, which gives a decent image in game, on a 2048 x 2048 pixels textures you can set 256 images for a 25.6 seconds film.

To have longer animation, you’ll have to size down frames (thus quality), size up texture or multiply your screens. Of course, UV values in Key Data must be adapted.

Calculations allow some compromise …

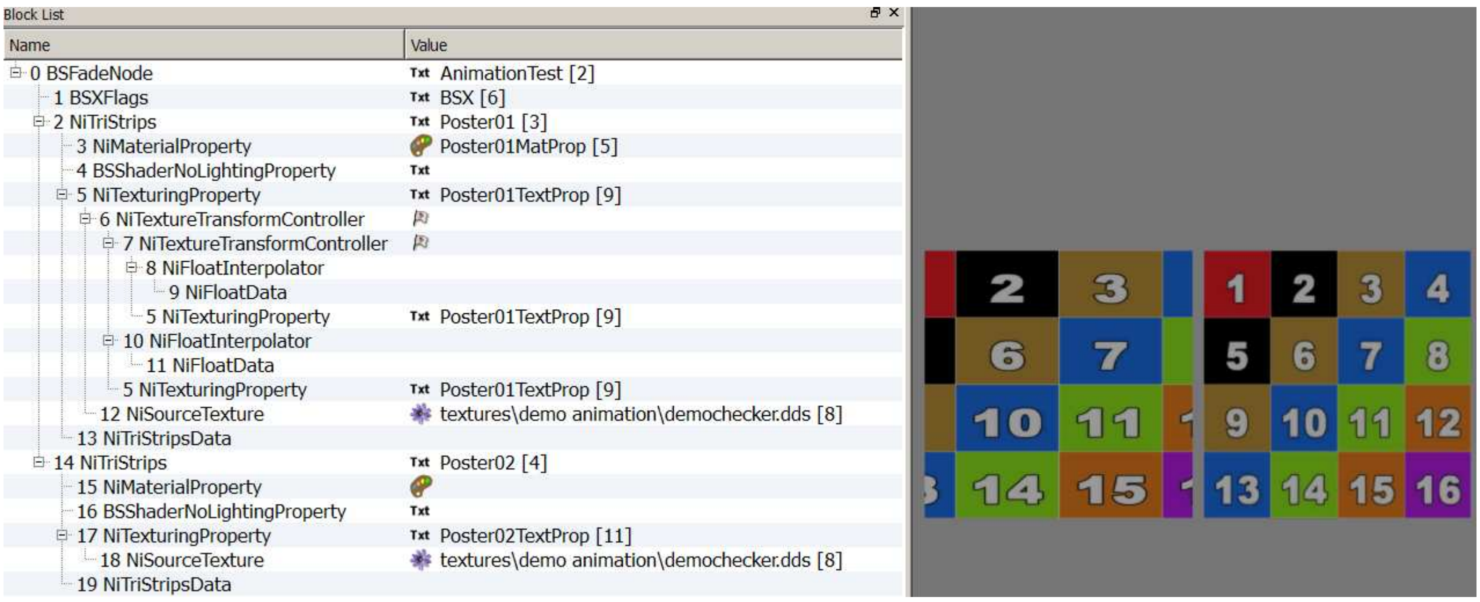
Scale UV Texture Animation

Step by Step

This will demonstrate how to achieve a zoom in & out effect, using Scaling on UV Axis.

The texture will stretch along the horizontal axis (U), come back to its original state, then scale on the vertical axis, come back to its original state, and then by combining both previous movement in one we'll obtain a zooming effect.

1. Open 03DemoScaleUV in Geck preview for a quick view.



Block List

Name	Value
0 BSFadeNode	Txt AnimationTest [2]
1 BSXFlags	Txt BSX [6]
2 NiTriStrips	Txt Poster01 [3]
3 NiMaterialProperty	Poster01MatProp [5]
4 BSShaderNoLightingProperty	Txt
5 NiTexturingProperty	Txt Poster01TextProp [9]
6 NiTextureTransformController	
7 NiTextureTransformController	
8 NiFloatInterpolator	
9 NiFloatData	
5 NiTexturingProperty	Txt Poster01TextProp [9]
10 NiFloatInterpolator	
11 NiFloatData	
5 NiTexturingProperty	Txt Poster01TextProp [9]
12 NiSourceTexture	textures\demo animation\demochecker.dds [8]
13 NiTriStripsData	
14 NiTriStrips	Txt Poster02 [4]
15 NiMaterialProperty	
16 BSShaderNoLightingProperty	Txt
17 NiTexturingProperty	Txt Poster02TextProp [11]
18 NiSourceTexture	textures\demo animation\demochecker.dds [8]
19 NiTriStripsData	

2. Shader is correctly set, the NiTexturingProperty is already renamed. The procedure would have been similar to the previous one.

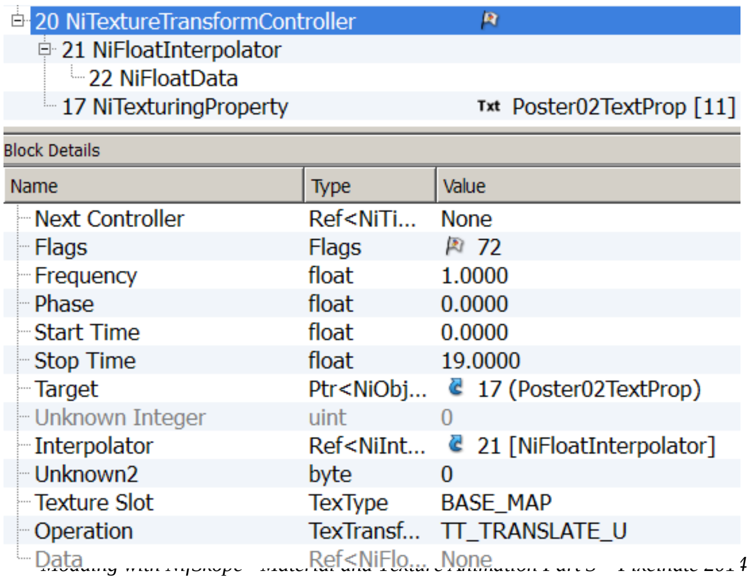
3. Build a Scale Block Animation.

   3 A. Insert a NiTextureTransformController, a NiFloatInterpolator and a NiFloatData.
   3 B. Link them to form a Block.

4. Adjust the settings of the NiFloatInterpolator.

   - Set it to <float_min>

5. Adjust the settings of the NiTextureTransformController as Following:



20 NiTextureTransformController	
21 NiFloatInterpolator	
22 NiFloatData	
17 NiTexturingProperty	Txt Poster02TextProp [11]

Block Details

Name	Type	Value
Next Controller	Ref<NiTi...	None
Flags	Flags	72
Frequency	float	1.0000
Phase	float	0.0000
Start Time	float	0.0000
Stop Time	float	19.0000
Target	Ptr<NiObj...	17 (Poster02TextProp)
Unknown Integer	uint	0
Interpolator	Ref<NiInt...	21 [NiFloatInterpolator]
Unknown2	byte	0
Texture Slot	TexType	BASE_MAP
Operation	TexTransf...	TT_TRANSLATE_U
Data	Ref<NiFlo...	None

6. Duplicate the Block. (Duplicate Branch)
7. Set the first NiTextureTransformController Operation to TT_SCALE_U
   And the second one to TT_SCALE_V
8. Chain-link the two Controllers. (See point 10 of the previous demonstration).
9. Finalize the linkage. (In NiTexturingProperty set the first of the two Controller)
10. Setting of the Keys.

   8 keys are needed for each of the NiFloatData. Interpolation is set on LINEAR_KEY.

Scale U Key Settings

Key	Time	0.0000
	Value	1.0000
Key	Time	2.0000
	Value	1.0000
Key	Time	4.0000
	Value	0.5000
Key	Time	6.0000
	Value	1.0000
Key	Time	14.0000
	Value	1.0000
Key	Time	16.0000
	Value	0.5000
Key	Time	18.0000
	Value	1.0000
Key	Time	19.0000
	Value	1.0000

Scale V Key Settings

Key	Time	0.0000
	Value	1.0000
Key	Time	8.0000
	Value	1.0000
Key	Time	10.0000
	Value	0.5000
Key	Time	12.0000
	Value	1.0000
Key	Time	14.0000
	Value	1.0000
Key	Time	16.0000
	Value	0.5000
Key	Time	18.0000
	Value	1.0000
Key	Time	19.0000
	Value	1.0000

Save and check your work in Geck preview.

Till now you have the animation continuously playing in game, with no interaction possible.

The next part will bring the NiControllerManager and the possibility to trigger several sequences of animations via script or activation. As it is a bit more complex, make sure the previous matter is well known and understood.

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/?