

## Introduction

The main target of this tutorial is to give a short introduction how custom animation can be created using Blender. For this you must use the newest (atm beta) script, 2.3.12. In this version the export option was activated. What else is needed, basic knowledge about modelling, a model to animate, the Kfm editor and nifscope.

So far i created one custom animation, so there still could be problems, but because the main principle should be same and this option is awaited for a long time, at least by me, i start with this tutorial. But keep in mind it was my first try ever to create an animation and also learned the basics while doing this.

The tutorial will cover the following parts, creating a new bone skeleton with nifscope, rigging vertices to the bones in Blender, animating in Blender and the final work in nifscope. Again most information are reused from other tutorials. Thanks goes to the authors of the other tutorials and the niftools team for making the tools and allow us to use them for free.

Ciao

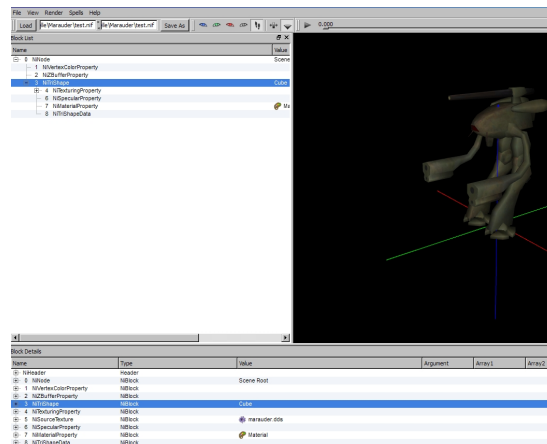
The\_Coyote

## Outline

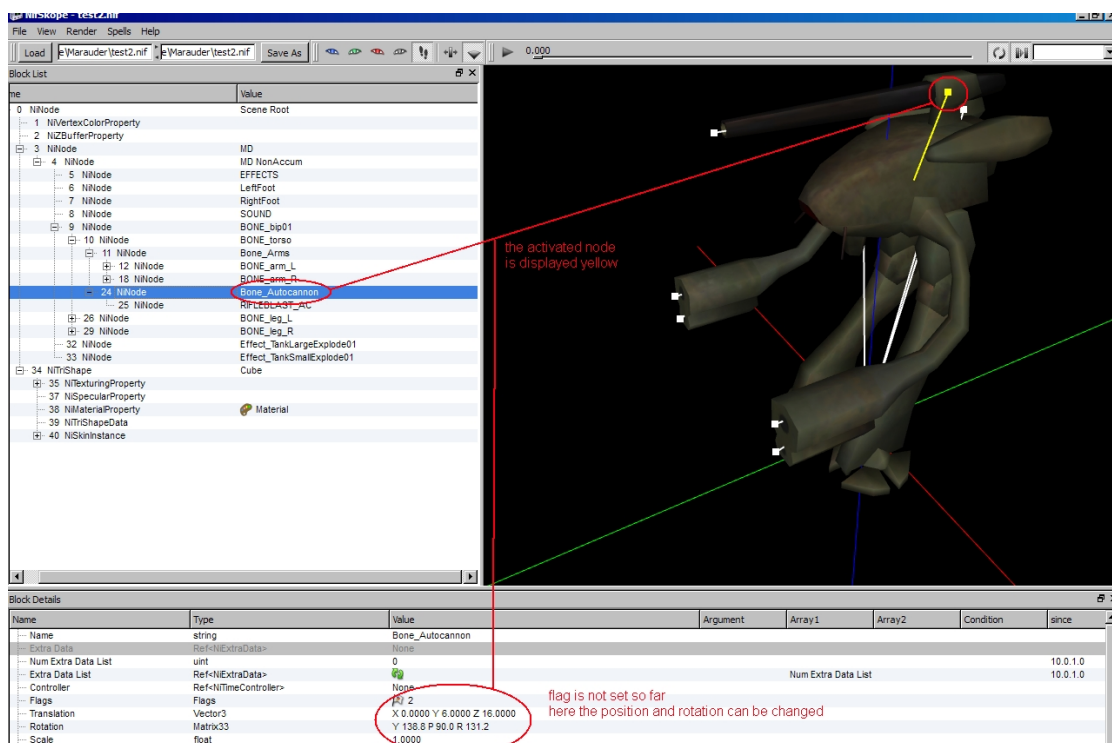
I. Creating a bone skeleton.....	2
II. Rigging the model.....	4
1. Assign vertices.....	4
2. Weight painting.....	5
III. Animating.....	6
1. Base set up.....	6
2. Animating.....	7
3. Export.....	8
IV. Afterwork.....	8
V. Options?.....	9
VI. Appendix.....	11

## Creating a bone skeleton

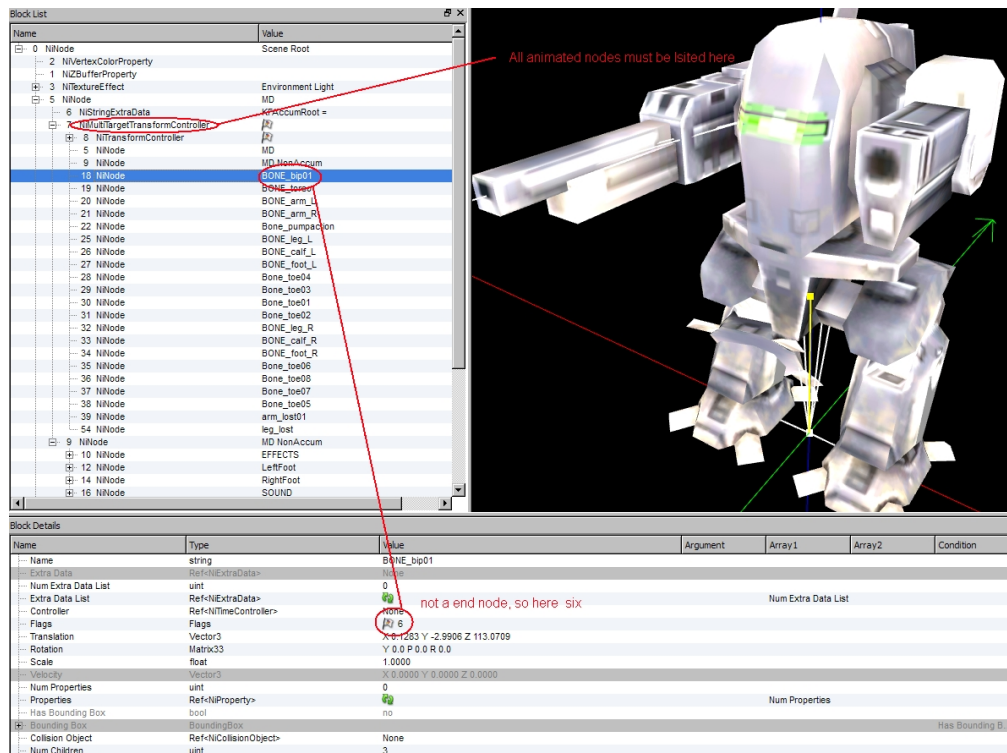
An interesting option when making a model and have the option to animate it is that you are no longer bound to the limits of existing animation. One example is the bone structure, you want a ship with 10 turrets, a four armed beserker, it is possible. What I did, first I had the model finished in Blender and exported it as unrigged nif. Now I open a bone structure of a similar unit – even if there are no a lot similarities I would start with a standard nif – and copy the new model in this nif. After the new model is added I remove all other models which were in the nif before.



Now the skeleton creation can start, if there are similarities move the bones to the new positions and / or change the rotation, if you need more bones add a new one and move the bone to the position. But be aware of a bone limit per model iirc – look at the Firaxis tutorial for more informations – in case you need to many bones it could be necessary that you must split your model. Another reason for splitting the model is if there are certain parts that shall be removed later in the animation, eg lost arm. While changing the position of the bone change the view often if your are really in the position where you want them, sometimes the perspective can give you the wrong imagination.



The final steps to finish your skeleton, set the flag the bones and add your new bones to the NiMultiTargetTransformController. The flag of the bones of your skeleton must be set to 6, or if it is the last bone in the three to 22. Both is necessary for a correct playing animation.

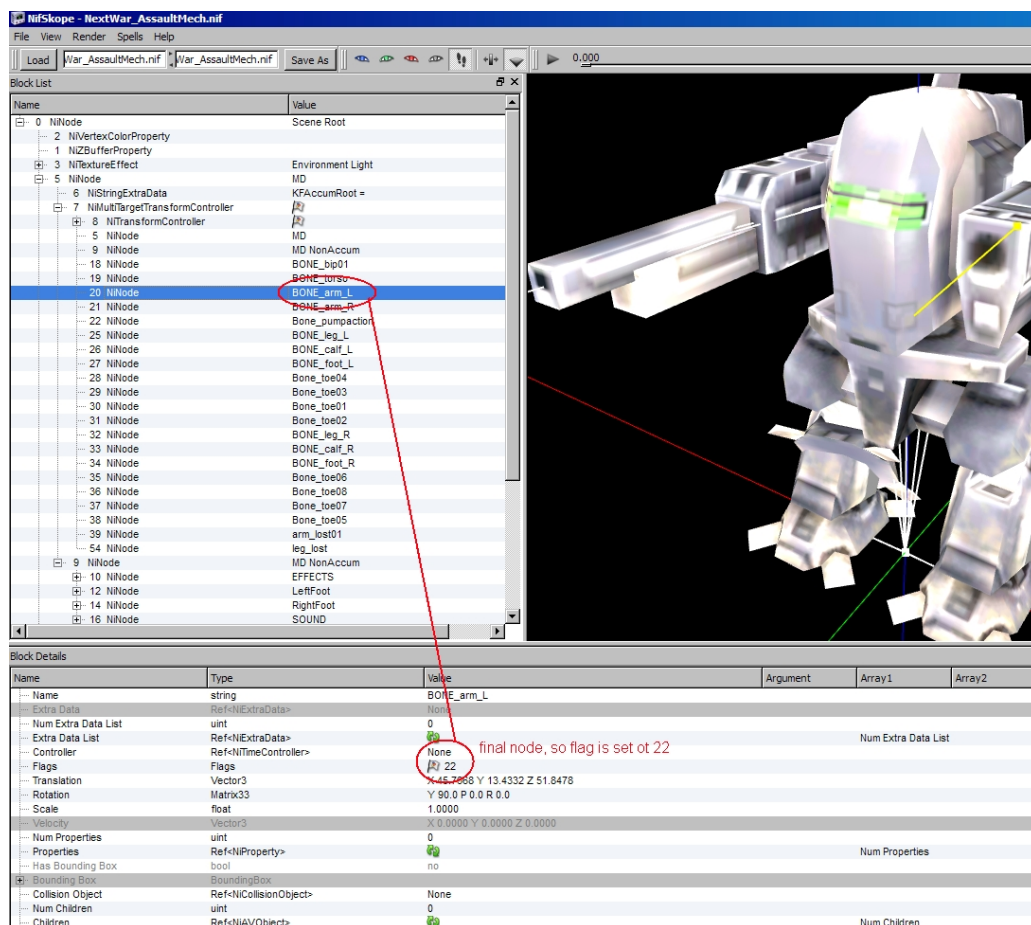


Block List

Name	Value
0 NiNode	Scene Root
2 NiVertexColorProperty	
1 NiZBufferProperty	
3 NiTextureEffect	Environment Light
5 NiNode	MD
6 NiStringExtraData	KFAccumRoot =
7 NiMultiTargetTransformController	[?]
8 NiTransformController	[?]
5 NiNode	MD
9 NiNode	MD NonAccum
18 NiNode	BONE_bip01
19 NiNode	BONE_toe01
20 NiNode	BONE_arm_L
21 NiNode	BONE_arm_R
22 NiNode	Bone_pumpaction
25 NiNode	BONE_leg_L
26 NiNode	BONE_calif_L
27 NiNode	BONE_foot_L
28 NiNode	Bone_toe04
29 NiNode	Bone_toe03
30 NiNode	Bone_toe01
31 NiNode	Bone_toe02
32 NiNode	BONE_leg_R
33 NiNode	BONE_calif_R
34 NiNode	BONE_foot_R
35 NiNode	Bone_toe06
36 NiNode	Bone_toe08
37 NiNode	Bone_toe07
38 NiNode	Bone_toe05
39 NiNode	arm_ist01
54 NiNode	leg_ist01
9 NiNode	MD NonAccum
10 NiNode	EFFECTS
12 NiNode	LeftFoot
14 NiNode	RightFoot
16 NiNode	SOUND

Block Details

Name	Type	Value	Argument	Array1	Array2	Condition
Name	string	BONE_bip01				
Extra Data	Ref<NiExtraData>	None				
Num Extra Data List	uint	0				
Extra Data List	Ref<NiExtraData>	None				
Controller	Ref<NiTimeController>	None				
Flags	Flags	6				
Translation	Vector3	X=-1253 Y=-2.9906 Z=113.0709				
Rotation	Matrix33	Y 0.0 P 0.0 R 0.0				
Scale	float	1.0000				
Velocity	Vector3	X 0.0000 Y 0.0000 Z 0.0000				
Num Properties	uint	0				
Properties	Ref<NiProperty>	None				
Has Bounding Box	bool	no				
Bounding Box	BoundingBox					
Collision Object	Ref<NiCollisionObject>	None				
Num Children	uint	3				



Block List

Name	Value
0 NiNode	Scene Root
2 NiVertexColorProperty	
1 NiZBufferProperty	
3 NiTextureEffect	Environment Light
5 NiNode	MD
6 NiStringExtraData	KFAccumRoot =
7 NiMultiTargetTransformController	[?]
8 NiTransformController	[?]
5 NiNode	MD
9 NiNode	MD NonAccum
18 NiNode	BONE_bip01
19 NiNode	BONE_toe01
20 NiNode	BONE_arm_L
21 NiNode	BONE_arm_R
22 NiNode	Bone_pumpaction
25 NiNode	BONE_leg_L
26 NiNode	BONE_calif_L
27 NiNode	BONE_foot_L
28 NiNode	Bone_toe04
29 NiNode	Bone_toe03
30 NiNode	Bone_toe01
31 NiNode	Bone_toe02
32 NiNode	BONE_leg_R
33 NiNode	BONE_calif_R
34 NiNode	BONE_foot_R
35 NiNode	Bone_toe06
36 NiNode	Bone_toe08
37 NiNode	Bone_toe07
38 NiNode	Bone_toe05
39 NiNode	arm_ist01
54 NiNode	leg_ist01
9 NiNode	MD NonAccum
10 NiNode	EFFECTS
12 NiNode	LeftFoot
14 NiNode	RightFoot
16 NiNode	SOUND

Block Details

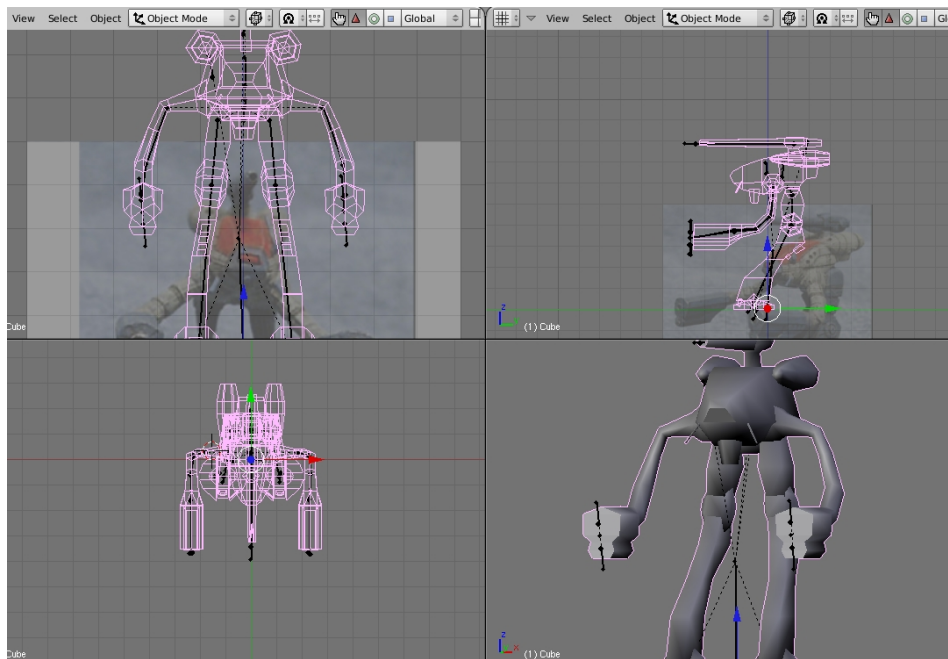
Name	Type	Value	Argument	Array1	Array2	Condition
Name	string	BONE_arm_L				
Extra Data	Ref<NiExtraData>	None				
Num Extra Data List	uint	0				
Extra Data List	Ref<NiExtraData>	None				
Controller	Ref<NiTimeController>	None				
Flags	Flags	22				
Translation	Vector3	X=45.7668 Y=13.4332 Z=51.8478				
Rotation	Matrix33	Y 90.0 P 0.0 R 0.0				
Scale	float	1.0000				
Velocity	Vector3	X 0.0000 Y 0.0000 Z 0.0000				
Num Properties	uint	0				
Properties	Ref<NiProperty>	None				
Has Bounding Box	bool	no				
Bounding Box	BoundingBox					
Collision Object	Ref<NiCollisionObject>	None				
Num Children	uint	0				
Children	Ref<NiAVObject>					

## Rigging the model

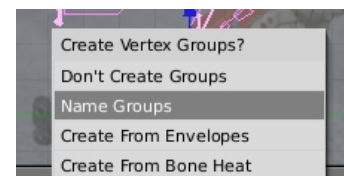
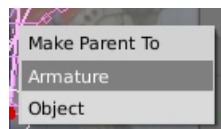
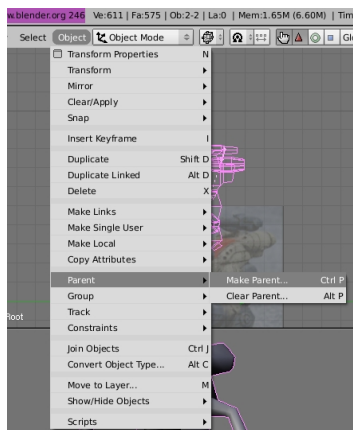
The next step after creating our skeleton is to rig the model to this. It is necessary to say which vertices shall be affected by the movement of one of the bone. There are two methods achieving this target I'm aware of in Blender. Also be aware that Blender offers a symmetrical rigging option if the bones follow a certain naming convention, see the Blender Gus tutorial listed in the appendix for details. If you don't plan to assign vertices to more than one bone the first method is enough for you, direct assigning vertices to a bone.

### Assign vertices

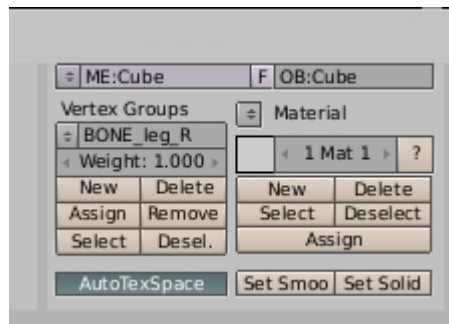
To assign vertices to a bone you first have to parent the skeleton to the model. For this import both in Blender, after this is completed Blender should look like this.



The selected part is the model, the black lines is our skeletons. To parent the skeleton to the model, select first your model and add after this the skeleton to your selection, you must have both selected and the order is important. After this follow the steps shown in the pictures below, first select parent, make parent, armature and finally name groups.



If the second window does not show up, most probability your selection order was wrong. After this step is finished the actual rigging can happen. For this select only your model and go to the edit mode. There in the panel view you can see a menu that looks like this

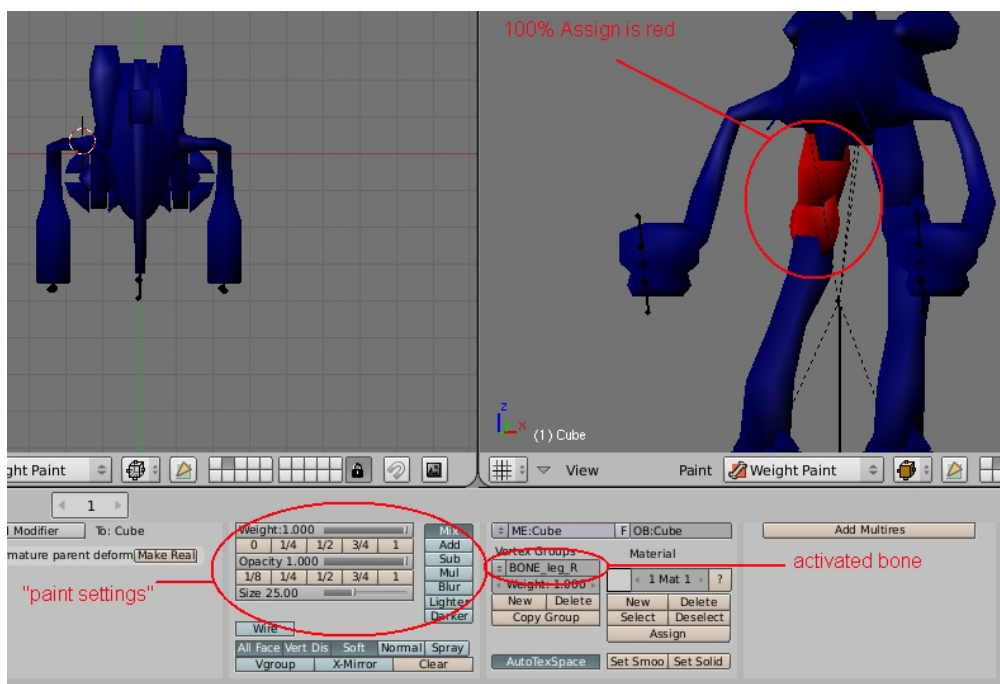


Interesting for us is the left side, under vertex group you can select the actual bone, the line below you can change the weight of the assign and the other six menu options are the actual actions. So select the first bone you want, after this select all but also only the vertices you want to rig to this bone and press *Assign*. Finished, that were all necessary steps. To control your work you can select *Desel.*(ect) and all rigged vertices are deselected. The *Weight* option is for the one vertices one bone case not important. Now you only have to repeat this step until all vertices are assigned to one of the bones, if you missed one the export script will show you your error.

After all vertices are rigged export your model again and play a bit in nifscope with the rotation of the bones. Now the rigged bones should also be moved, if not there was an error.

## Weight painting

So the following will be only guessing, i never did it. Using weight values when assigning bones is only necessary when vertices shall be affected by more than one bone. In this case you can use the *Weight* option from above or paint the weight on the vertices. If you want to do this, import a standard human unit in Blender, go to weight paint mode and look how the bones are painted there.



Now try to mimic this for your model. After finishing the job, again export your model and try it out in nifscope. If you have never rigged to an animation (or better an existing skeleton) you have to do the exact same steps, but to mention again i only did the 100% assign so far, ships and airplanes are that much easier.

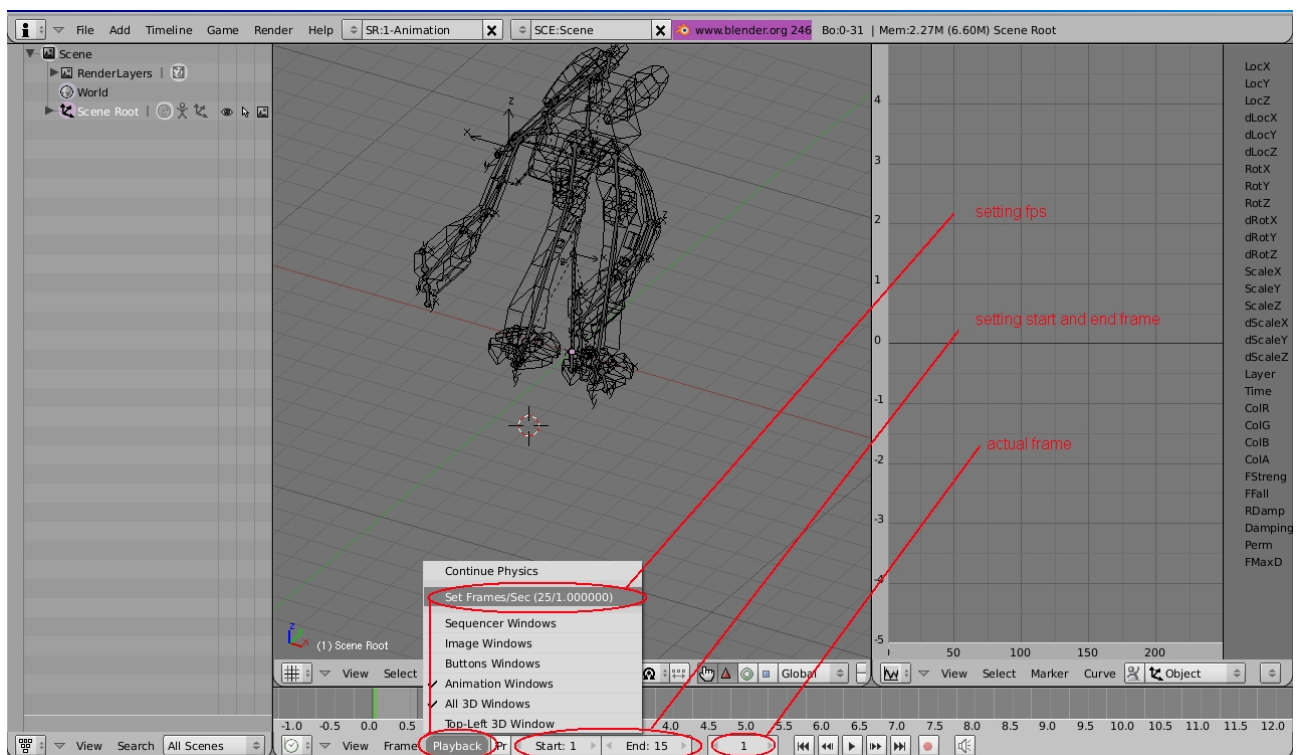
## Animating

Now go back to Blender (or reopen your blend file with your rigged model) and go to the animation mode. This can be done pressing ctrl and left. But before actual animation some basics i know so far, and that is not too much.

### Base Set Up

- ➔ First you have to know what kind of animation you want, there are different flow charts for different animation sets.
- ➔ Second the frame rate should be set to 30 fps.
- ➔ Third, try to be exact as possible if a fixed frame number is given in the flow chart.
- ➔ Fourth, think before starting to work, try to imagine what which animation sequence should play and how it will look in game. A busy idle animation will often not look so good.
- ➔ And fifth, look which animation need which start position, eg the strike sequence starts from the fortify position.
- ➔ Sixth, the way the animation is played (one time or cycle), while set also later, is also important, so when you have a cycling animation, eg run, your end position should be the same like the start pose.
- ➔ Have fun, you can give 'life' to something

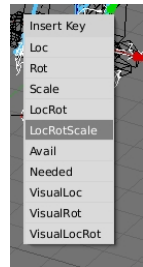
So but no the real base set up. First select your start frame, 1, and your end frame, the number from the flow chart - here 15. After this set the frame rate to 30 fps





## Animating

Afterward deselect your model, select your skeleton and change to pose mode. Now select all bone and press in the window the *i-key*. In the pop up select LocRot, then the location and rotation is stored for this point, LocRotScale would also store the scaling of the bone. I have no idea what the last five do, but the the first five keys should be enough to animate the object.



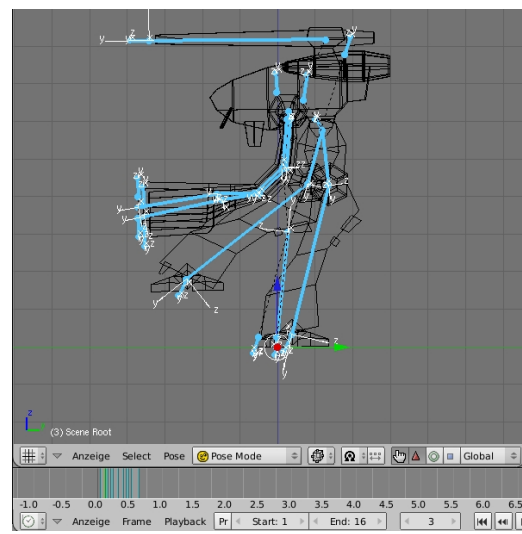
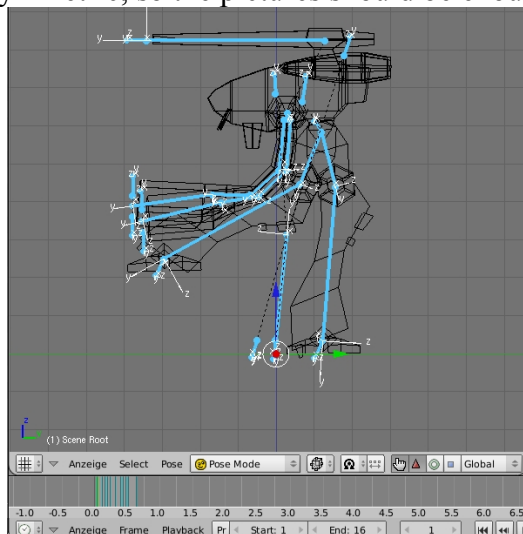
Now after setting the start position set also the end position key if they must the same. Go to the frame and repeat the step from above. Now the actual animation can start. Be aware you don't must set every frame, if you have a movement it is enough to set the start and end position Blender will do the rest. I set some key positions and corrected manual possible errors, like flying while running. Also be aware, your nodes and your model will be a (sub)child of the MD or MD NonAccum node, so every movement there will have a double effect on your model. Also you must no select every bone before pressing the *i-key*, but be aware only the position of the selected nodes is saved.

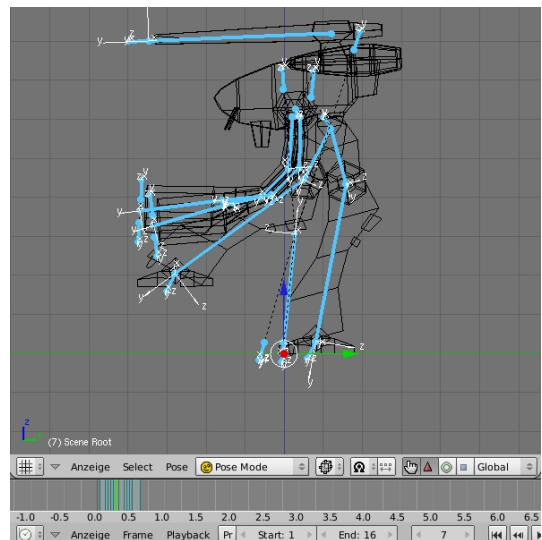
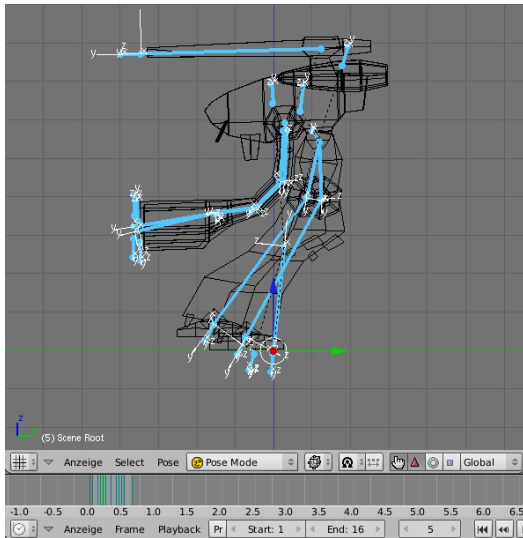
Finally there is an option to add text key informations to the kf files, it is an important part of the file, eg the information what kind of animation, when effects are played (sound and visual) and so on are stored there. For closer information look at the tutorial in the appendix. If you plan certain entries it would be good to mark the frames, the entry itself can be done later in nifscope, but when the frame is set, at least a text key for this frame is created.

Now you only need the creative part of your mind and create every needed animations. I would recommend to save every file separately. Also, even when i'm not sure – when trying it i could have done another mistake, do not try to save an animation with a start key different from one. At least consider yourself as warned, my try gave me a wrong start and end time, the actual time of the frame in Blender.

What to say more, try it, it is not so difficult, but for the start try a small animation only including all minimal necessary files, eg for a machine unit: Idle, Run, Fortify, FortifyIdle, Strike, Hurt, Die, DieFade would be enough. Also try not too much, the available time is really short sometimes.

As example some picture of the run animation of the Marauder. Showed are frame number 1, 3, 5 and 7. Because it must stop like the sequence started, it is a cycling animation, the animation is symmetric, so the pictures should be enough





Now after finishing the animation the next job is the export.

## Export



The picture shows all you need. You start the export script and after choosing the name the option menu pops up. There you select *Export Animation Only (.kf)* and Civilization IV as game. After pressing *Ok* the export script automatic create the kf file.

This steps must been done for every animation independently, After all animations file are created the Blender part is finished. As next step before finally polish the kf files you could create the kfm file. For this purpose use the KFM Editor from the utilities section. Again i would start with a similar kfm file and remove all unneeded entries and change the name of the linked files. Don't forget to change the link to your nif right in the top of the file.

## Afterwork

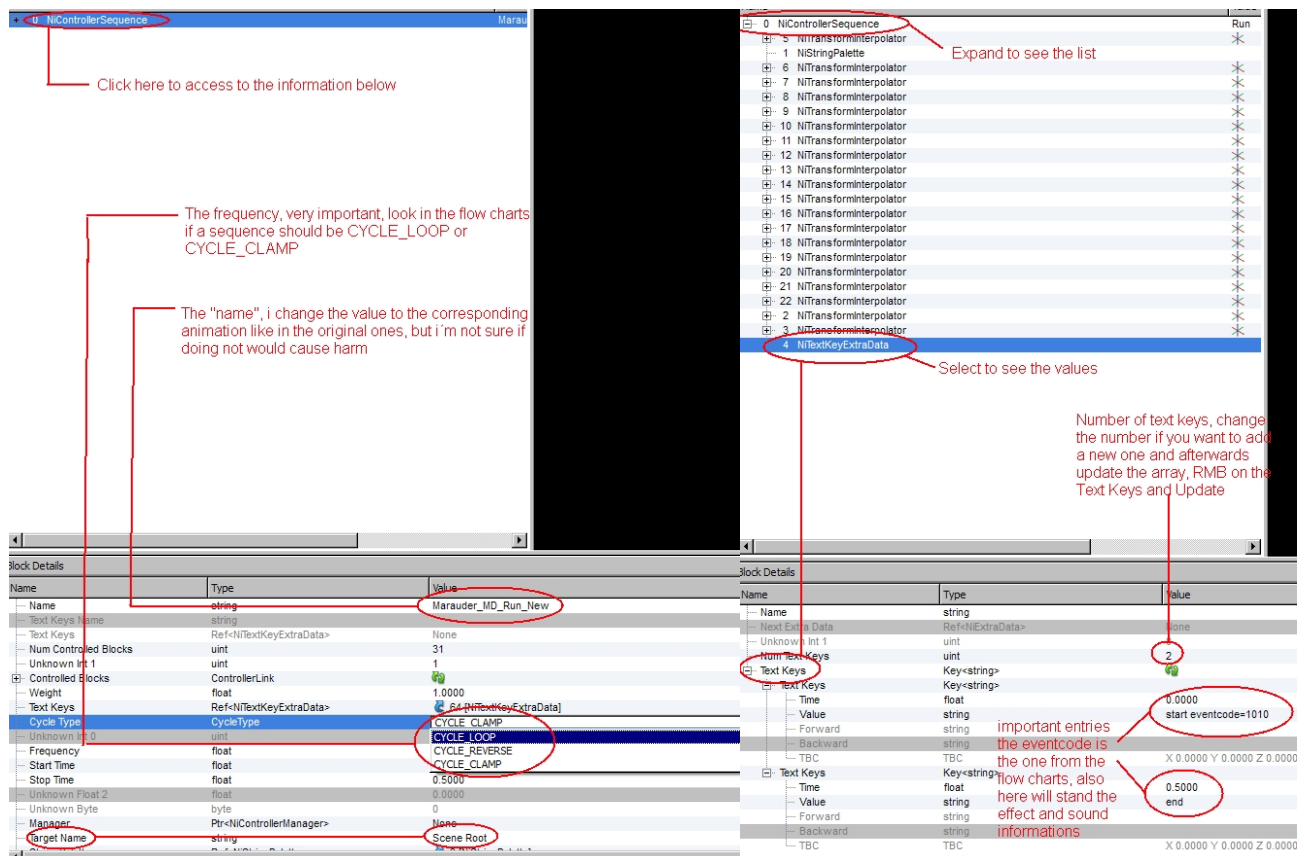
Now a bit afterwork is necessary, so far no effects or sound effects are played. For example when your unit is hit you want a explosion. For this open a nif from which you want to use an effect, also open the corresponding kf file where the effect is played.

Before doing everything else, copy the effect node from the nif in your, copying because then no subnode can be missed. Copying alone but will not play the effect. Now open your created kf file and look at the text entry values of the original kf files. There you will see entries like SOUND:.... or EFFECT:..., that are the now interesting values, search for the entry where after effect the name of the copied node is written - the third value is the name of the effect in the xml – and add this to your kf files. The sound entries controls the played sound, and because an explosion without



BOOM would sound silly, add this entry also your kf. Which line should be self explaining, the effects and sound have easy to recognise names. What sometimes is a bit tricky is the correct rotation and scale of the effect node, you have to try it in game and if your unit fire backwards change the direction. If an effect is too small or to big you can change the scale of the node, you really should try scale 100 for the death explosion one time. When playing with the units you will see a lot of different effects and sounds, eg the foot stamps of the mech are also generated this way.

The last important step is the changing of some values in the kf files. For this purpose i would open an original files for the same sequence and compare the values. The most important part will be the adding of the start with the eventcode information entry and the stop entry. For more informations see the pictures.



So after this was done try your animation and hopeful all worked like it should. If not check the values, loop yes or not, are there missing reference in the nif, eg not added to the NiMultiTargetTransformController, correct entries in the kfm file and so one. There are enough places where an error could occur. Also again, be aware that the option is only test wise activated so far.

## Options?

What can be done, hard to say, i only used the method one time so far, but the complete process described above, starting with a new armature and ending with a new animation, but i only used the LocRotScale keys. Other options i have in my mind could be the change of one part of an animation, eg the strike animation to have a two sword attack, for this i would create a nif in the fortify pose, import this one in Blender and make the animation. If the method could be applied to leaderheads, no idea, if they don't differ in the way the animations works it should work, but this

would be something for the leaderhead experts to try out. Also i didn't experimented with all possible kf entries, eg the boolean values, in most cases i saw them they were used to control the visibility of an object.

## Appendix

### Software:

Blender: <http://www.blender.org/>  
Gimp: <http://www.gimp.org/>  
Paint.net: <http://www.getpaint.net/index.html>  
NifScripts: [http://sourceforge.net/project/showfiles.php?group\\_id=149157&package\\_id=166219](http://sourceforge.net/project/showfiles.php?group_id=149157&package_id=166219)  
NifScope: [http://sourceforge.net/project/showfiles.php?group\\_id=149157&package\\_id=170735](http://sourceforge.net/project/showfiles.php?group_id=149157&package_id=170735)  
Niflib: [http://sourceforge.net/project/showfiles.php?group\\_id=149157&package\\_id=166113](http://sourceforge.net/project/showfiles.php?group_id=149157&package_id=166113)  
PyFFI: [http://sourceforge.net/project/platformdownload.php?group\\_id=199269](http://sourceforge.net/project/platformdownload.php?group_id=199269)  
Python: <http://www.python.org/>

### Tutorials:

Export from Blender: <http://forums.civfanatics.com/showthread.php?t=167335>  
Texturing: [http://www.colacola.se/howto\\_texttut.htm](http://www.colacola.se/howto_texttut.htm)  
Assign vertices to bones: <http://forums.civfanatics.com/showthread.php?t=252568>  
some basics and assign vertices: <http://forums.civfanatics.com/showthread.php?t=267233>  
Nifviewer, Nifscope: <http://forums.civfanatics.com/showthread.php?t=263814>  
UV edit in nifscope: <http://forums.civfanatics.com/showthread.php?t=258966>  
Nifscope: <http://forums.civfanatics.com/showthread.php?t=183742>  
Nifviewer, Part I: <http://forums.civfanatics.com/showthread.php?t=163585>  
Nifviewer, Part II: <http://forums.civfanatics.com/showthread.php?t=165689>  
Nifviewer, Part III: <http://forums.civfanatics.com/showthread.php?t=176106>  
Nifviewer, shader, teamcolour: <http://forums.civfanatics.com/showthread.php?t=245363>  
Animations: <http://forums.civfanatics.com/showthread.php?t=271331>

### new:

<http://forums.civfanatics.com/showthread.php?t=279548>  
[http://wiki.blender.org/index.php/Manual/PartI/Your\\_First\\_Animation\\_in\\_30\\_plus\\_30\\_Minutes\\_Part\\_II](http://wiki.blender.org/index.php/Manual/PartI/Your_First_Animation_in_30_plus_30_Minutes_Part_II)  
[http://niftools.sourceforge.net/wiki/Blender/Oblivion\\_Character\\_Animation](http://niftools.sourceforge.net/wiki/Blender/Oblivion_Character_Animation)