Blender : export d'un mesh sans squelette

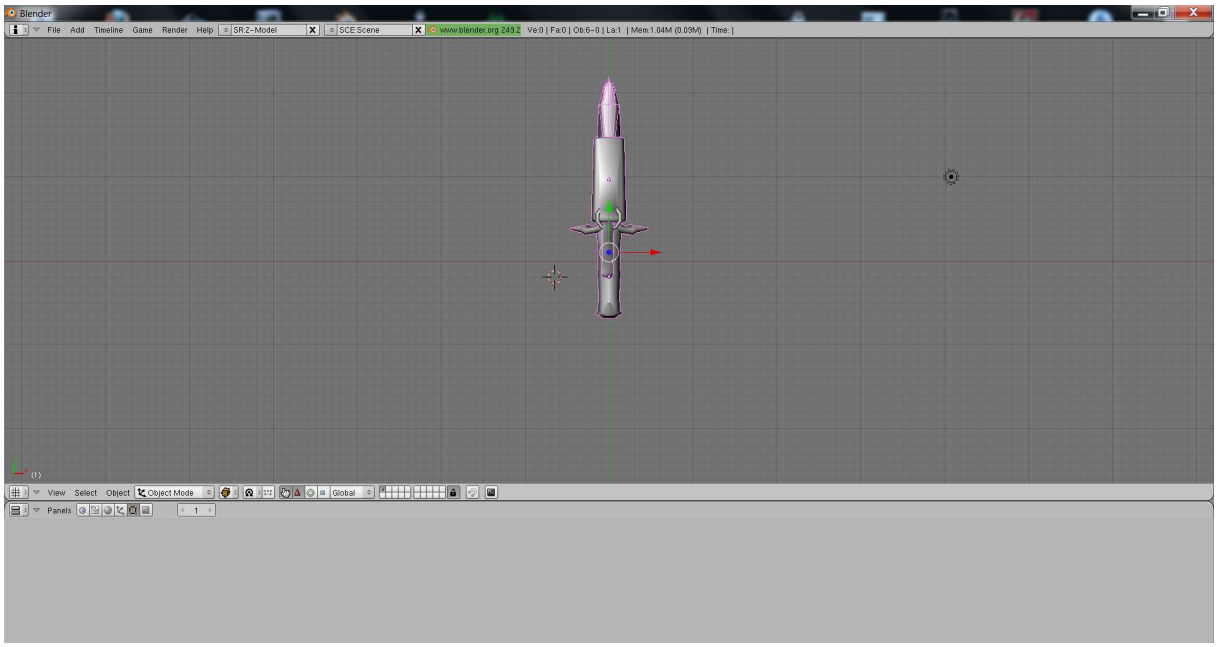
Par Gérauld (pour et distribué sur le site de la Confrérie des Traducteurs)

Ce tutoriel explique la marche à suivre pour exporter un mesh de Blender et le remettre au format de Skyrim. L'exemple est l'export du mesh importé lors du tuto précédent : la dague de fer.

Préparer un material pour l'export de l'UV map :

L'UV map est la carte de correpondance entre la texture et les sommets de la forme de l'objet. Pour que Blender exporte l'UV map, il faut assigner une texture à chaque objet composant le mesh.

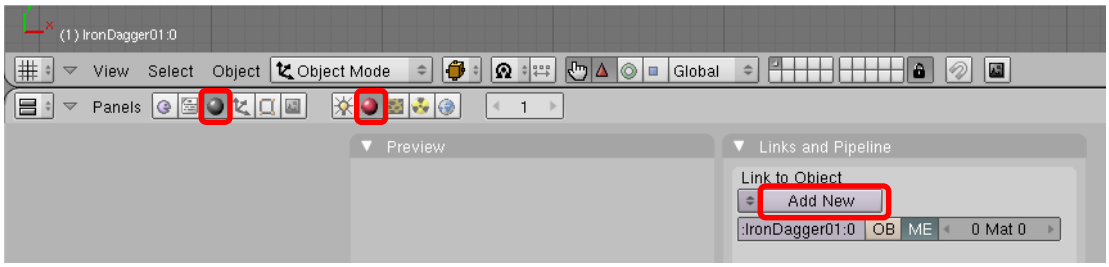
Nous sommes en object mode et nous avons ça sur l'écran :



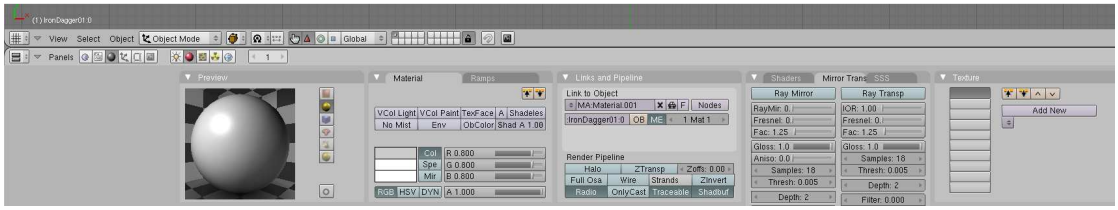
Sélectionner un objet en faisant un clic droit sur la forme. Son nom apparait en bas à gauche de la fenêtre de rendu.

Le nom de l'objet est IronDagger01 :0. Cela correspond au nom du NiTriShape importé (voir tuto Nifskope sur les meshes d'armes)

Avant d'affecter une texture, il faut affecter un material à l'objet. Pour se faire, il faut cliquer sur le bouton en forme de sphère dans le bloc de gauche, faire de même dans le bloc de boutons suivant et cliquer sur ADD NEW dans la fenêtre du bas :

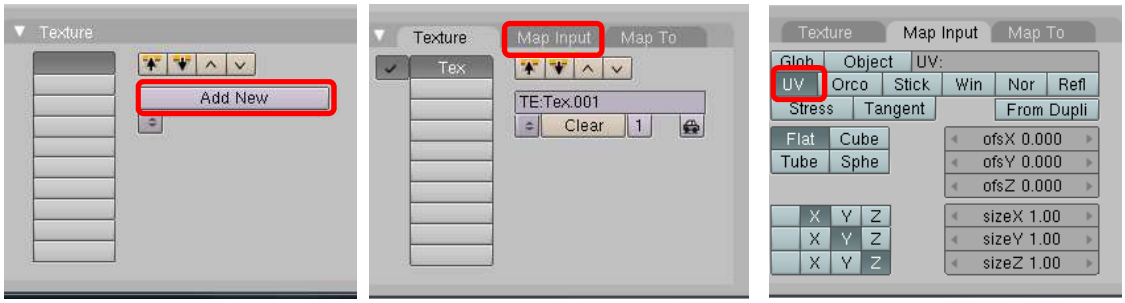


Le bloc du bas ressemble maintenant à ceci :



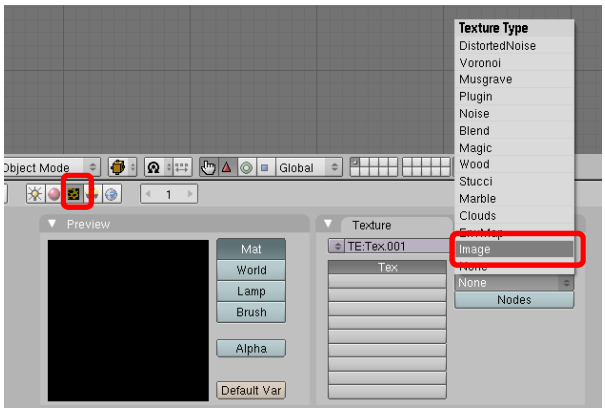
Il va maintenant falloir lui dire que le material contient une texture.

Dans le bloc en bas, tout à droite (Texture), cliquer sur ADD NEW. On obtient l'affichage de droite. Choisir l'onglet Map input et sélectionner UV.



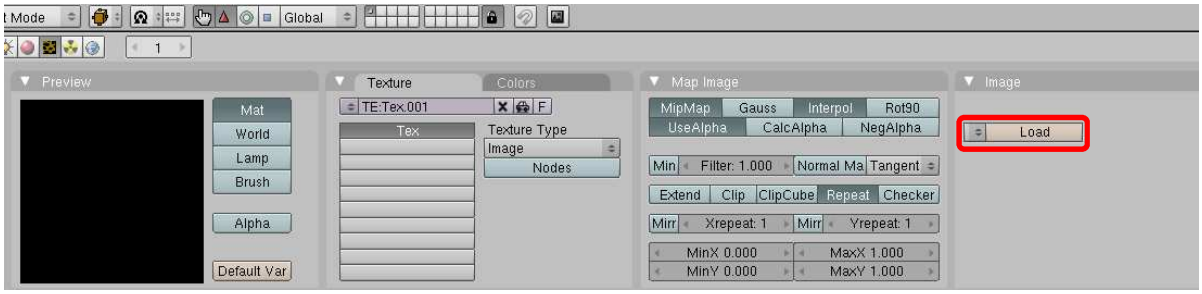
Nous avons donc un material avec une texture dont le mappage provient d'une UVmap.

Nous allons définir la texture associée au material. Le choix de la texture n'a aucune importance car ce ne sera de toute façon pas celle-ci qui servira dans le mesh final. Pour ma part, j'ai mis une texture dans le répertoire par défaut de Blender que j'utilise pour tous mes exports.

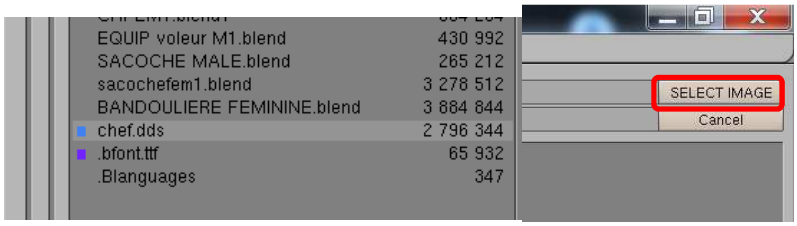


Cliquer sur le bouton juste à côté de la sphère du deuxième bloc de boutons puis choisir Image dans le menu déroulant Texture Type.

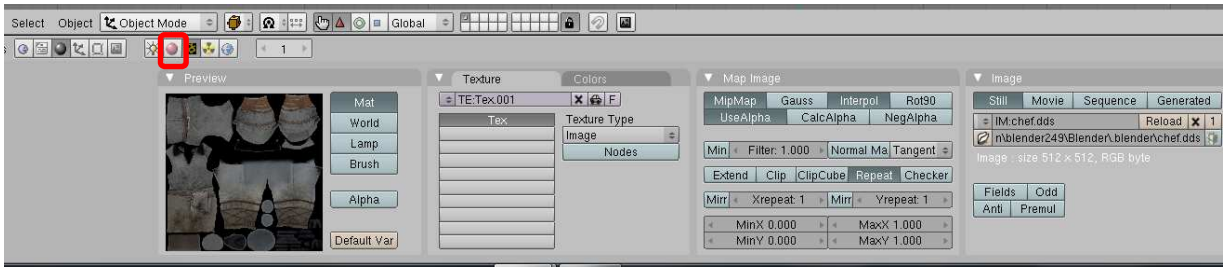
Cliquer ensuite sur le bouton LOAD dans le dernier bloc apparu à droite de celui dans lequel vous avez choisi Image :



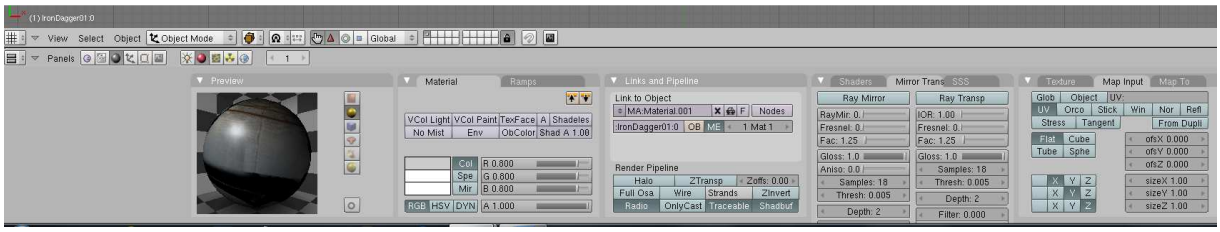
Choisir une texture au format DDS (n'importe laquelle) et cliquer sur SELECT IMAGE en haut à droite.



La texture apparait ensuite dans la partie basse de l'écran. Cliquer sur le 2ème bouton du deuxième bloc de boutons pour revenir à la fenêtre de Material :



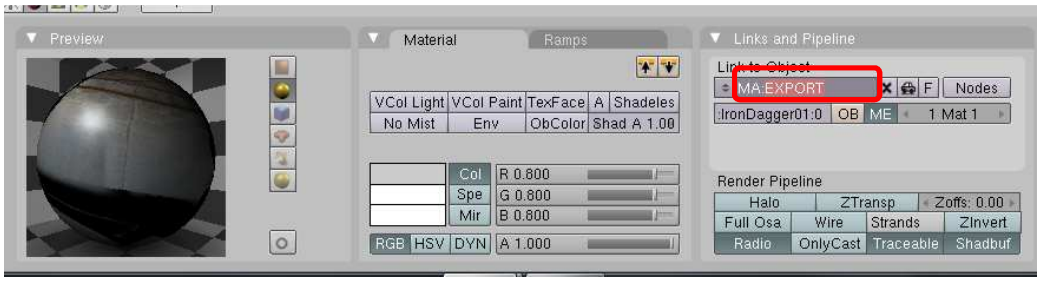
La fenêtre matérial représente maintenant la sphère blanche du début avec la texture choisie. La manipulation a été réussie pour cet objet :



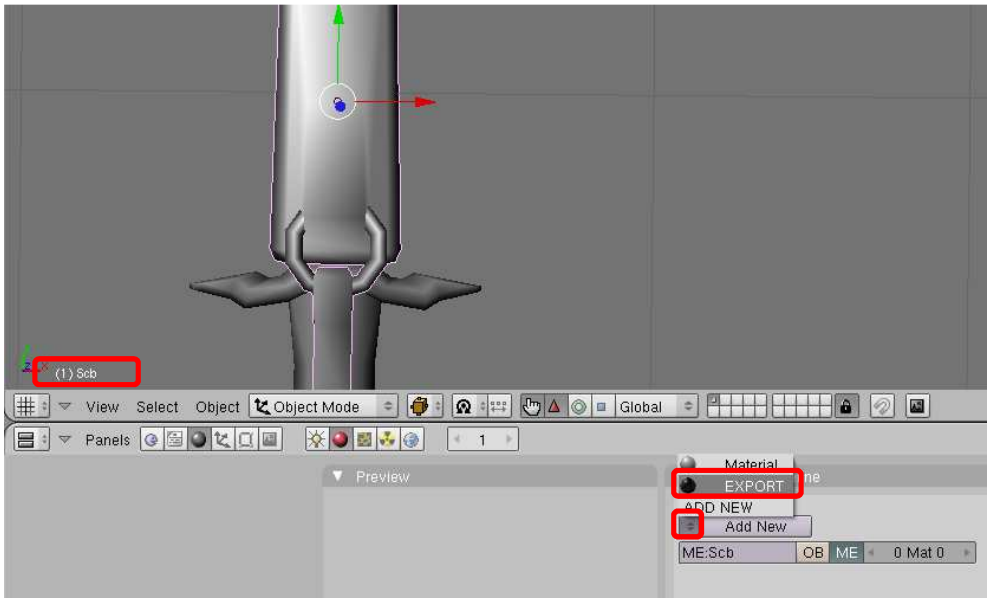
Notre material est prêt. Il reste maintenant à l'appliquer à tous les objets.

Application du material à tous les objets du mesh :

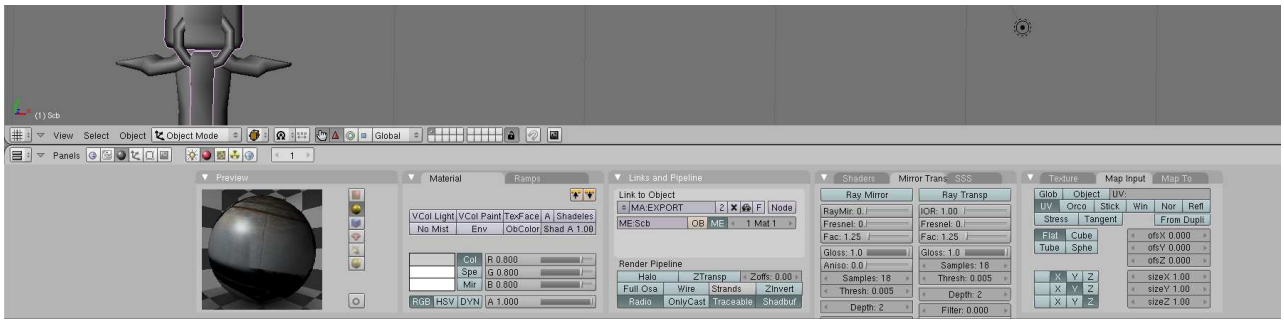
Renommer le material pour le retrouver plus facilement :



Placer la souris dans le fenêtre de rendu et faire un clic droit sur un autre objet pour le sélectionner. Les contours de l'objet sélectionné deviennent roses et son nom est affiché en bas à gauche de la fenêtre de rendu. Cliquer sur la double flèche à gauche de ADD NEW et sélectionner le material EXPORT que nous avons créé.



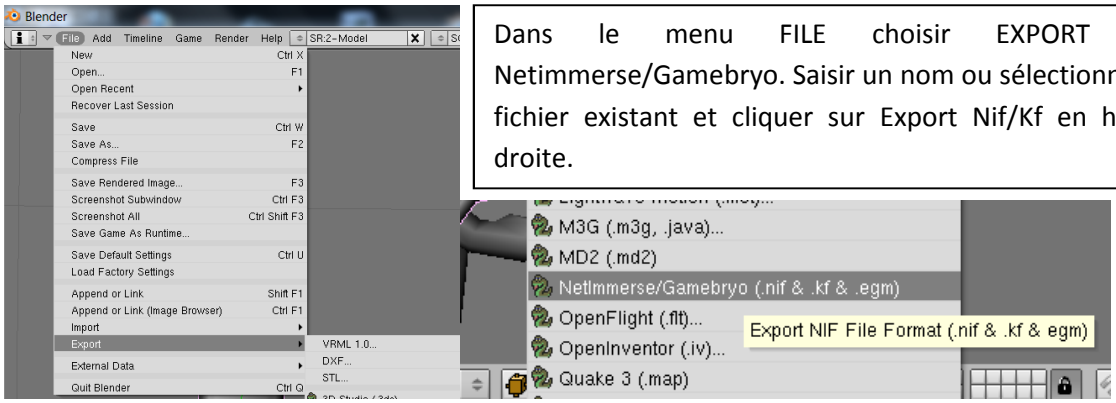
Nous retrouvons l'affichage obtenu avec le précédent objet.



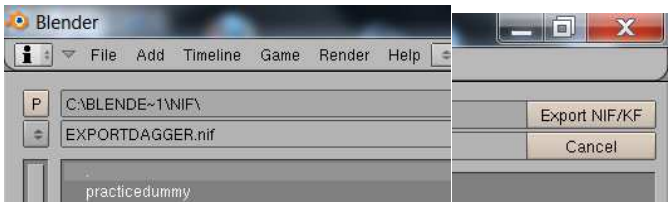
Faire de même pour TOUS les objets du mesh (un objet par NiTriShape importé. Il suffit de faire des clics droits pour que Blender sélectionne à tour de rôle les meshes qui sont superposés.

Export de Blender en format Nif :

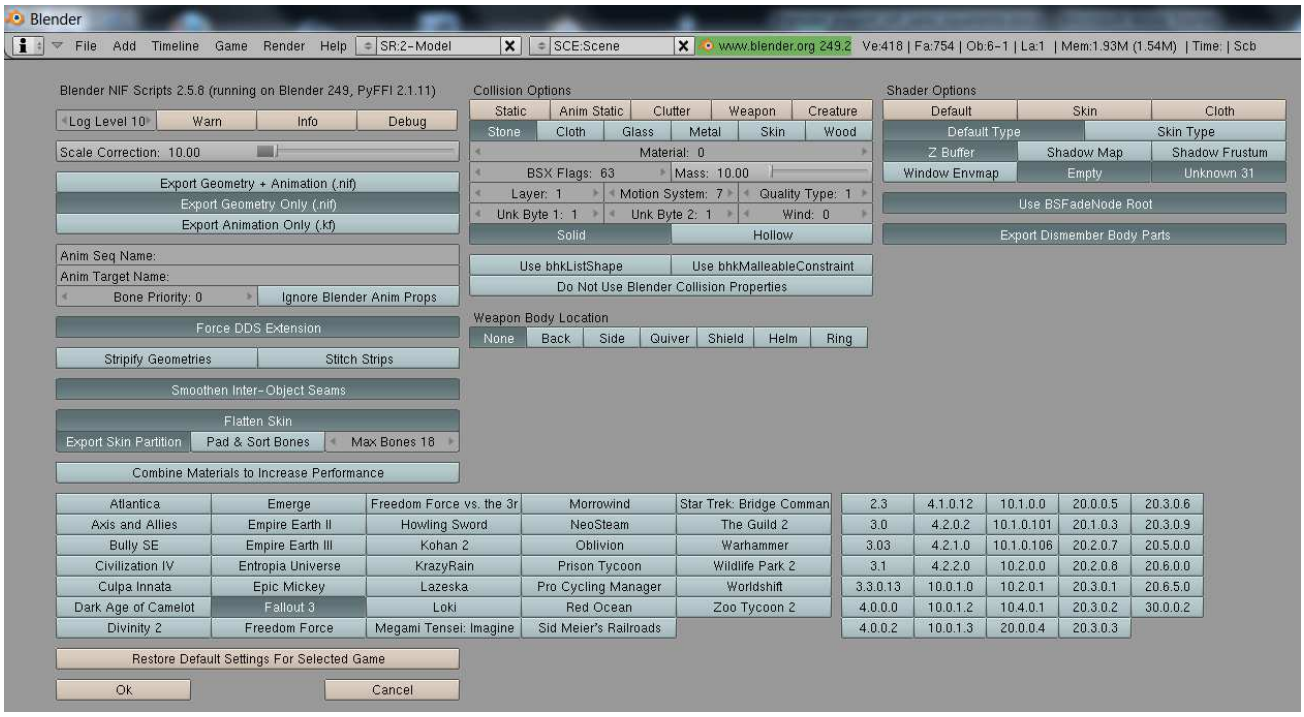
Placer la sourie dans la fenêtre de rendu en Object mode et appuyer sur A pour tout sélectionner. Si les contoures ne sont pas roses, appuyer à nouveau sur la touche A.



Dans le menu FILE choisir EXPORT puis Netimmerse/Gamebryo. Saisir un nom ou sélectionner un fichier existant et cliquer sur Export Nif/Kf en haut à droite.



Reproduire les réglages suivant et cliquer sur OK :

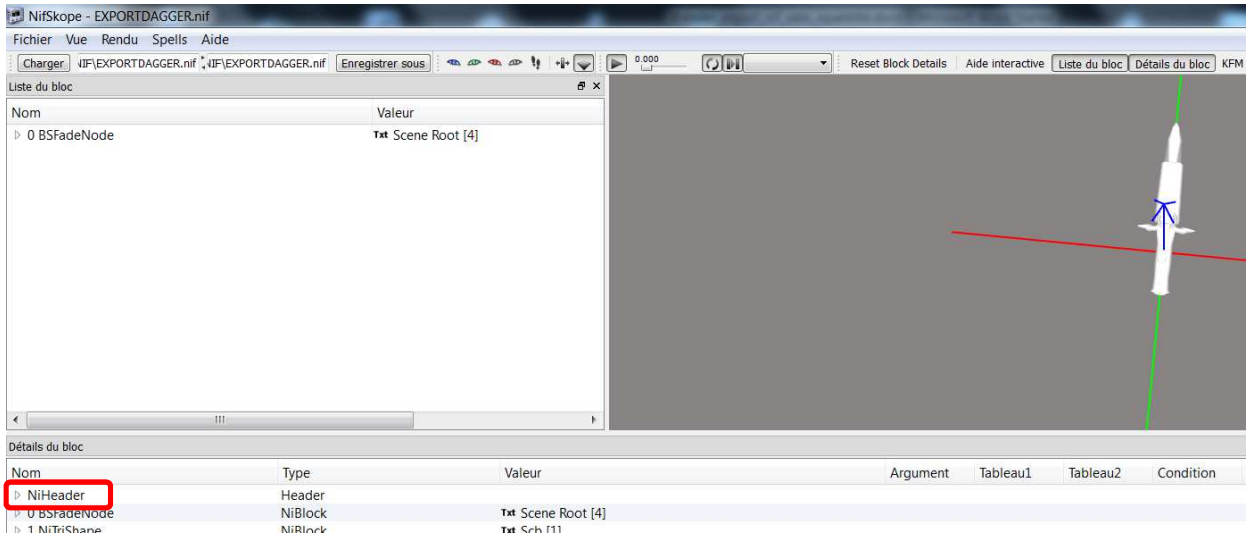


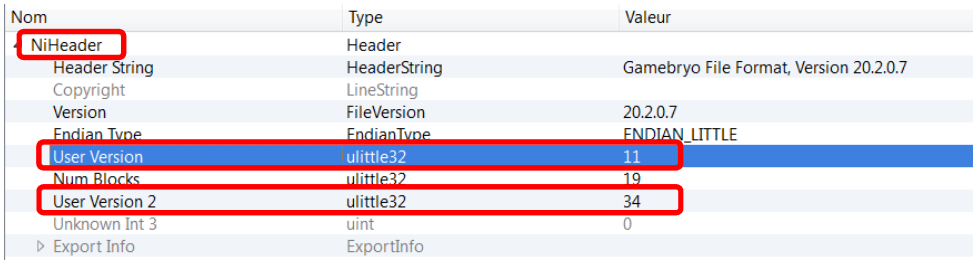
A ce moment-là. Le mesh est exporté mais ne peut pas encore servir dans le CK ou en jeu. Il faut le retravailler sous Nifskope.

Préparation du mesh sous Nifskope :

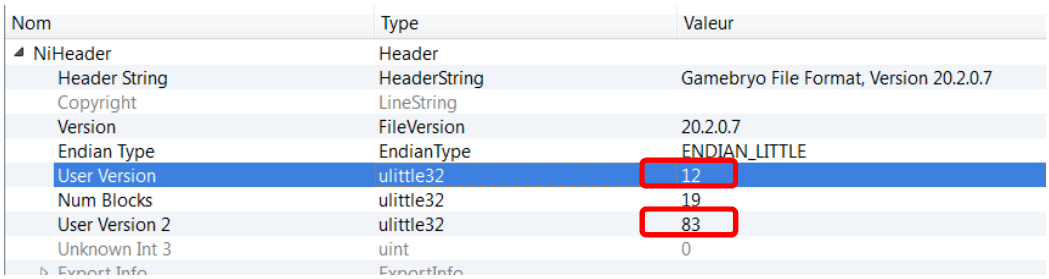
Nous allons faire l'inverse des manipulations de l'import vers Blender dans un premier temps.

Tous d'abord, charger le mesh et cliquer sur la petite flèche à côté de HEADER dans la fenêtre du bas.

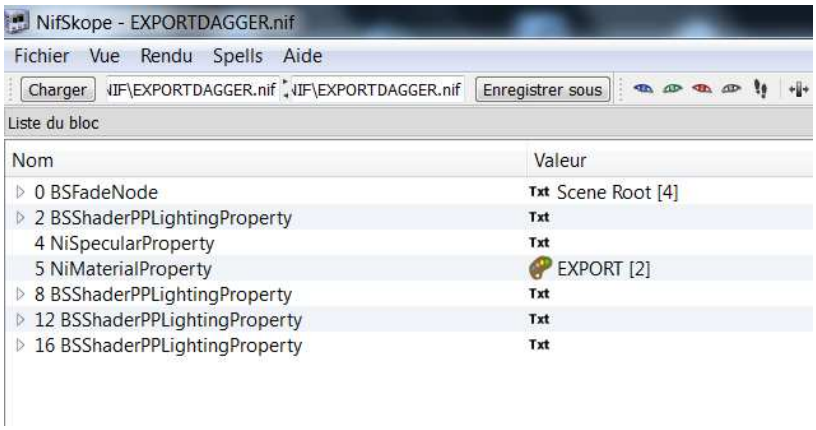




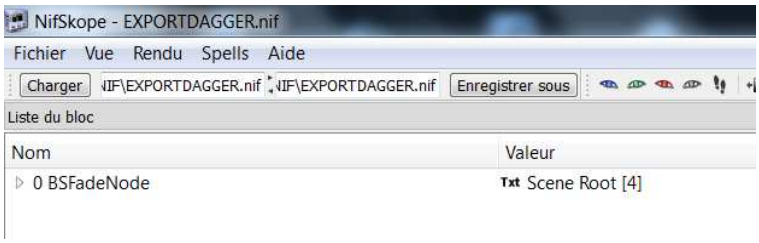
Dans les lignes User Version et User Version 2, remplacer 11 par 12 et 34 par 83.



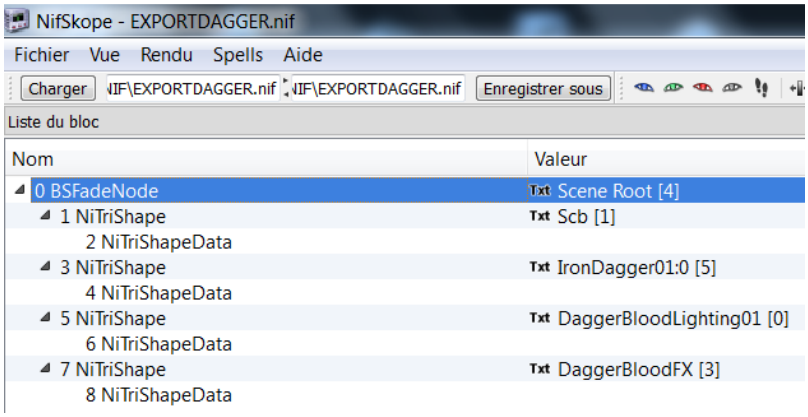
Sauvegarder le mesh et charger le à nouveau dans Nifskope pour que les modifications soient actives.



Quand le mesh est à nouveau ouvert, on aperçoit des blocs qui ne sont plus inclus dans l'arborescence du BSFadeNode. Il faut les supprimer. Il faut les sélectionner un par un et faire CTRL + Suppr pour les supprimer. Nous obtenons :



Maintenant, nous allons jeter un coup d'œil à l'arborescence du BSFadeNode :



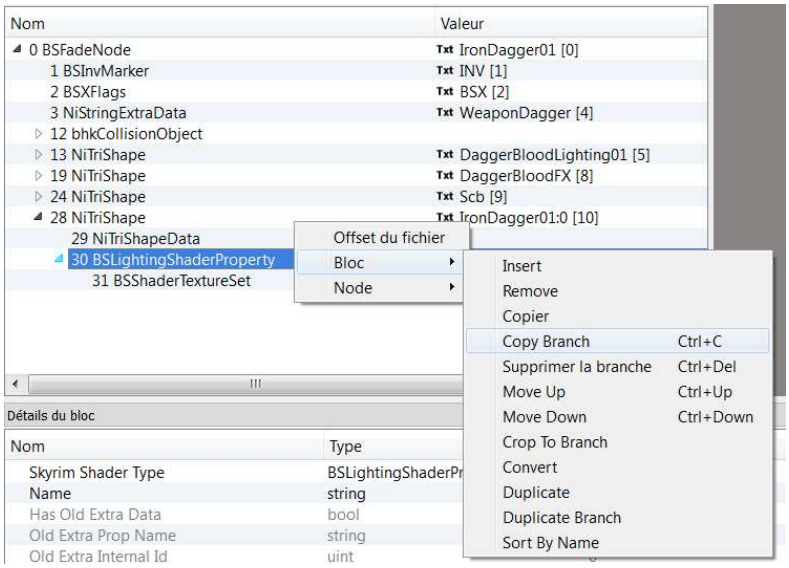
Nous retrouvons bien les 4 Nitrishapes (et leur NiTrishapeData associé) en children du BSFadeNode.

Il manque donc les blocs BSShaderLightingProperty, les BSTextureSet, les NiStringExtraData et les NiAlphaProperty dépendant des Nitrishapes. Nous allons nous occuper de ceci tout de suite.

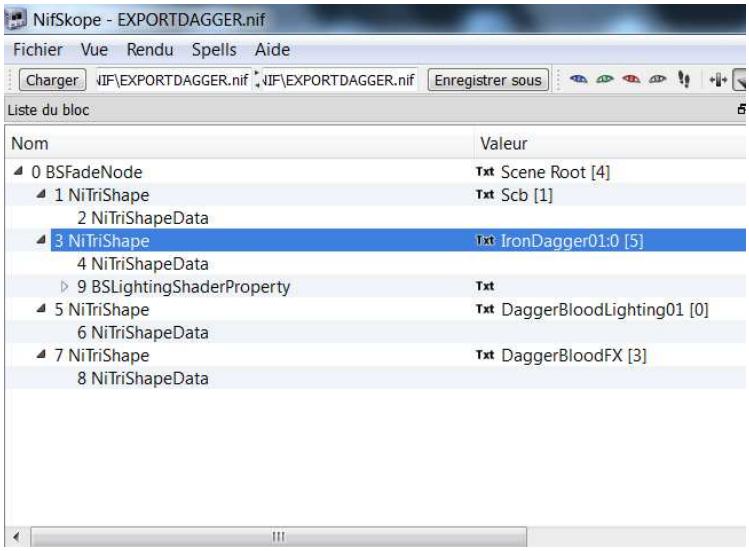
Pour se faire, nous allons récupérer ces blocs à partir du mesh vanilla qui nous a servi lors de l'import (car les Nitrishapes ont le même nom, mais ça marche avec les autres meshes du même type). Ces blocs seront collés dans le meshe et placés dans l'arborescence de notre mesh exporté.

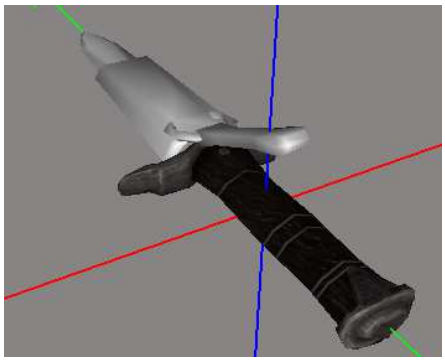
Une autre méthode, plus courte, consiste à importer les NiTriShapes modifiées dans un mesh vanilla. Par soucis pédagogique, nous utiliserons la version longue.

Pour copier une branche (bloc + son arborescence), il faut le sélectionner et faire CTRL + V (ou un clic droit sur le bloc concerné, Bloc, Copy Branch).

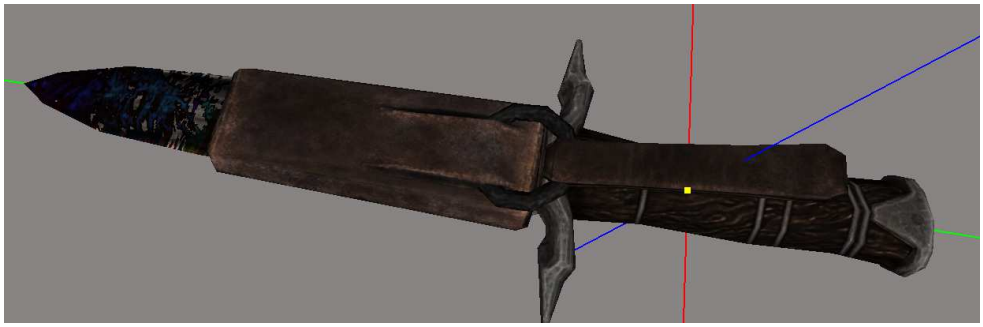


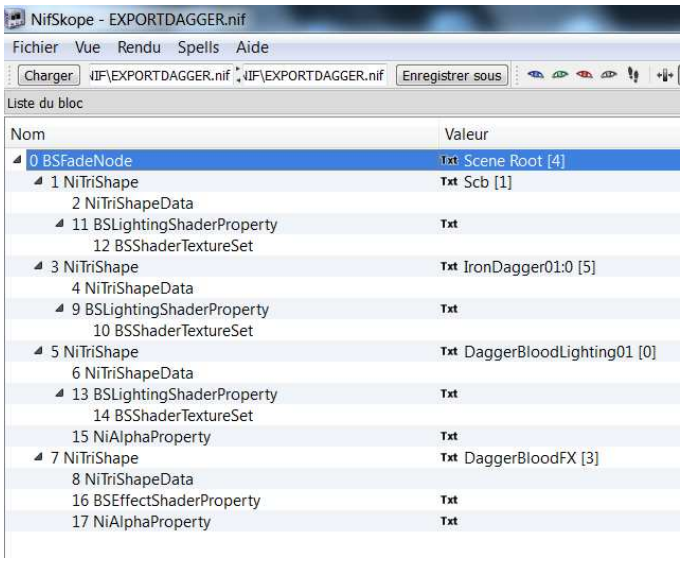
Puis, dans notre mesh exporté, on sélectionne le Nitrishape correspondant et on fait un CTRL + V :



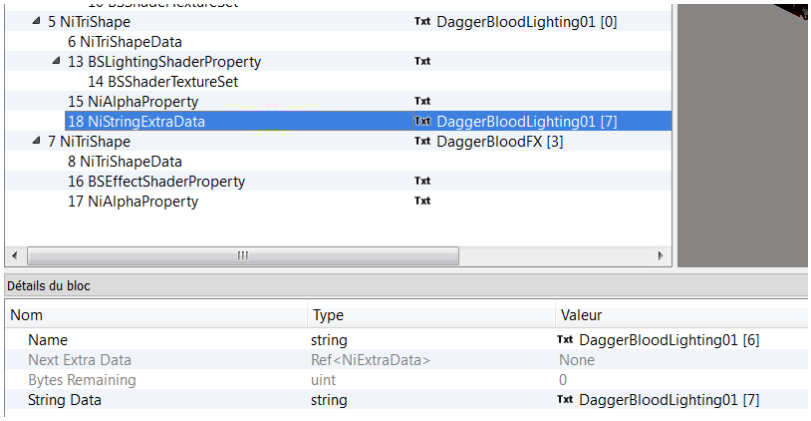


Notre arme commence à retrouver des couleurs. Faire de même avec les BSShaderLightingProperty et les NiAlphaProperty des autres Nitrishapes jusqu'à obtenir ça :





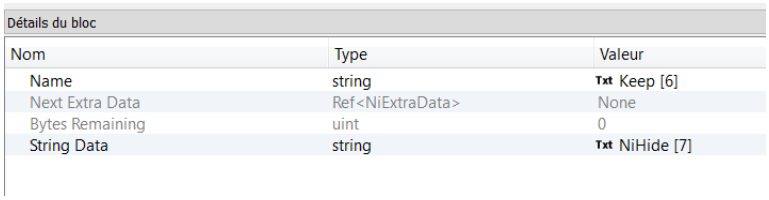
Nous allons maintenant copier les NiStringExtraData dépendant des Nitrishape du sang (lire tuto Nifskope sur les armes pour plus de détails)



Les noms contenus dans les blocs ne sont pas bons. Nous allons les changer en clisaunt sur les icones txt des lignes Name et Sting Data. Cliquer sur l'icone txt à droite de Name.

Taper Keep dans le champ du bas puis OK. Faire la même manipulation pour la ligne String Data en saisissant NiHide. **ATTENTION AUX MAJUSCULES.**

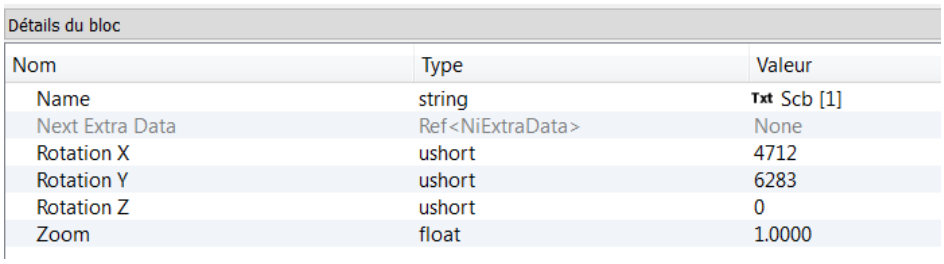
On obtient :



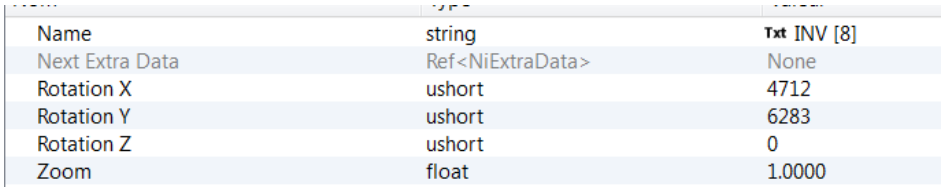
Un fois tous les éléments devant se trouver dans les Nitrishapes collés dedans avec les bonnes chaines de caractères, nous allons ajouter les blocs liés au mesh d'arme en lui-même (la collision sera vue en dernier).

BSInvMarker :

Copier le BSInvMarker du mesh vanilla dans le mesh exporté :

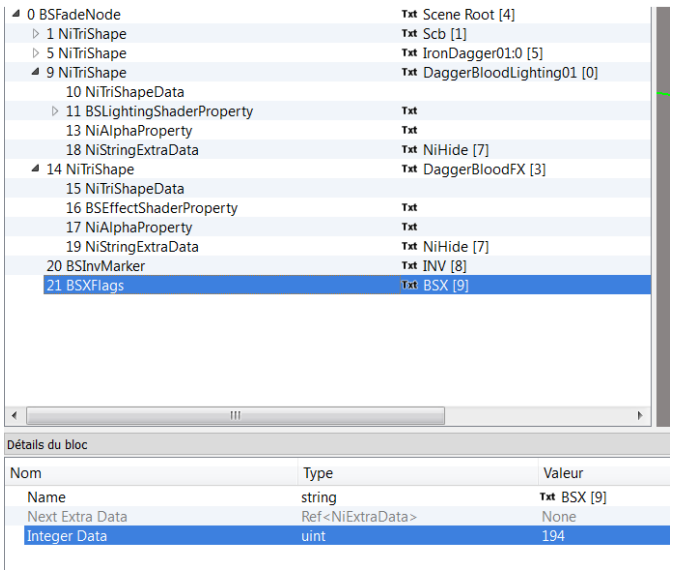


Remplacer la valeur dans la ligne Name par INV (en majuscules)



BSXFlags :

Faire la même chose avec le bloc BSXFlags. Ramplacer lea valeur de sa ligne name par BSX (en majuscules) :

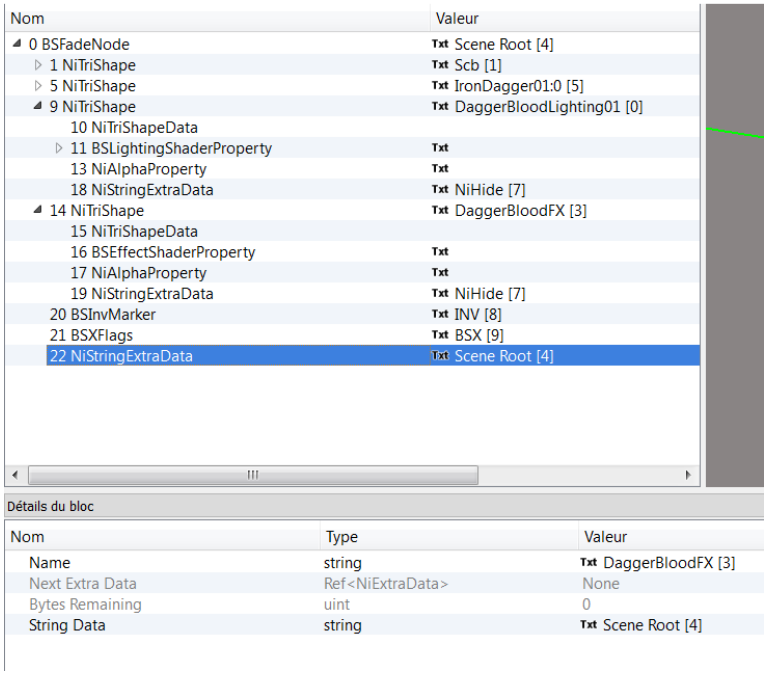


La valeur de la ligne Integer Data doit être à 194 pour une arme.

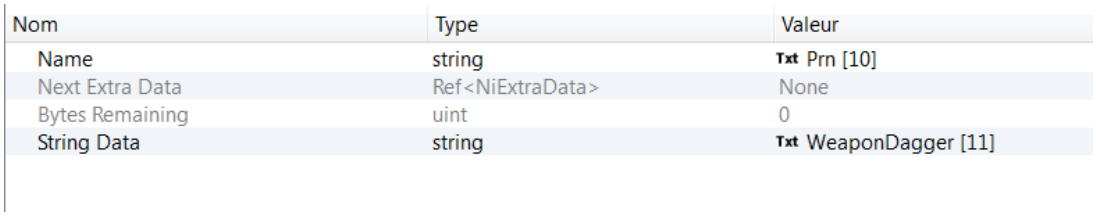
Bien vérifier cette valeur. Pour plus d'informations voir tuto Nifskope - mesh d'armes.

NiStringExtraData du BSFadeNode :

Le copier à partir du mesh d'origine et le coller en sélectionnant le BSFadeNode :



Remplacer les valeurs comme suit :



La ligne String Data est liée au type d'arme du mesh. WeaponDagger correspond à une arme de type dague, WeaponBack à une armes à deux mains (voir tuto Nifskope-meshes d'armes).

Il ne nous manque que le bloc de collision à ajouter.

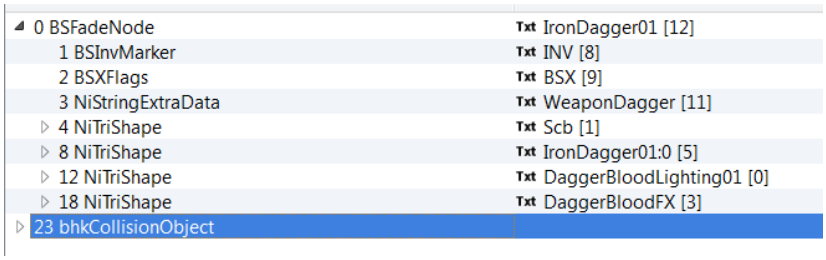
Bloc de collision :

Si l'arme exportée a une forme voisine de celle du mesh vanilla, il est plus simple de récupérer directement le bloc de l'autre mesh. Si votre arme a une forme peu courante ou n'ayant pas grand-chose à voir avec celle d'origine, il faudra créer un nouvelle forme de collision mais il faudra quand même récupérer un bloc vanilla.

Dans tous les cas, il faut récupérer le bloc de l'ancien mesh.

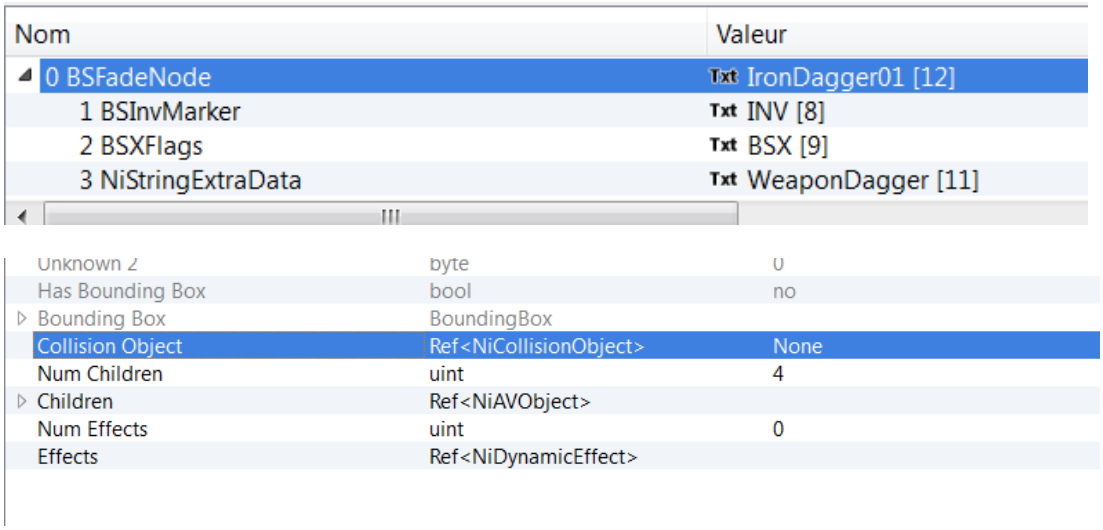
Un copier-coller direct donne souvent ceci :

Il suffit de renommer notre BSFadeNode du même nom que celui qui est présent dans le mesh vanilla pour pouvoir importer le bloc.



Le bloc s'est mis en bas du mesh et n'est pas dans l'arborescence du BSFadeNode. On va l'y mettre.

Sélectionner le BSFadeNode et chercher la ligne Collision Object dans les détails :



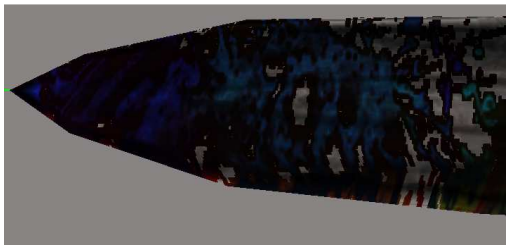
Remplacer None par le numéro du bhkCollisionObject importé, donc 23 dans notre exemple :

Bounding Box	BoundingBox	
Collision Object	Ref<NiCollisionObject>	23 [bhkCollisionObject]
Num Children	uint	4
Children	Ref<NiAVObject>	

S'il apparait, c'est que le numéro était bien le bon.

Les flags des Nitrishapes

Si vous regardez l'image du mesh vanilla et du mesh exporté de Blender, vous allez voir une énorme différence au niveau de la lame. Cela provient des deux Nitrishapes du sang. Ils sont visibles sur le mesh exporté et pas sur le mesh vanilla (avec l'option Show Hidden déselectionnée dans le menu de Nifskope).




Sélectionner à tour de rôle les deux NitriShapes dont le sang dépend et repérer la ligne flag avec une valeur 14.

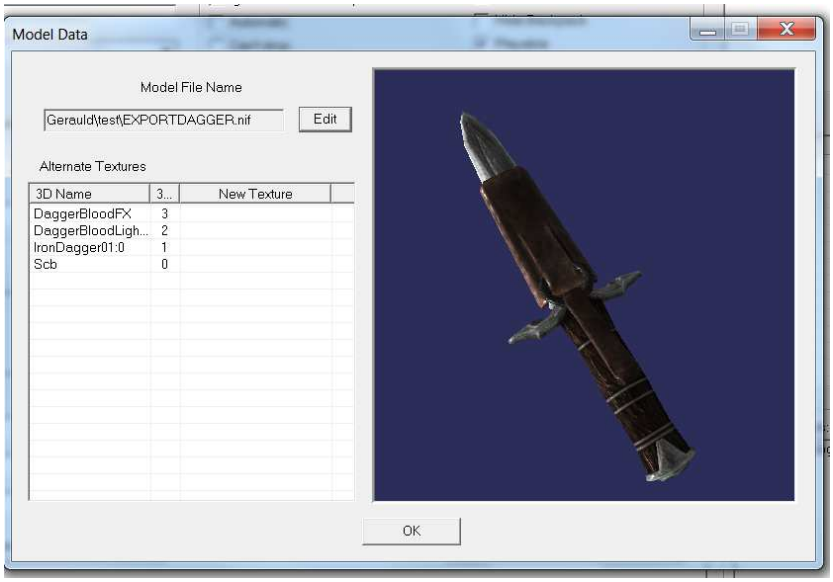
17 NiTriShape	Txt IronDagger01:0 [5]
21 NiTriShape	Txt DaggerBloodLighting01 [0]
27 NiTriShape	Txt DaggerBloodFX [3]
28 NiStringExtraData	Txt NiHide [7]
29 NiTriShapeData	
30 BSEffectShaderProperty	Txt
31 NiAlphaProperty	Txt

Détails du bloc

Nom	Type	Valeur
Skyrim Shader Type	BSLightingShaderPropertyShaderT...	Default
Name	string	Txt DaggerBloodLighting01 [0]
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<NiExtraData>	None
Num Extra Data List	uint	1
Extra Data List	Ref<NiExtraData>	
Controller	Ref<NiTimeController>	None
Flags	Flags	14
Unknown Short 1	ushort	8

Remplacer la valeur par 15 dans les deux NiTriShape dont le nom contient Blood. Normalement, l'affichage du mesh vanilla et de celui d'origine est maintenant identique.

Il suffit maintenant de sauvegarder le mesh et le tester dans le CK (pas besoin de charger d'esp ou d'esm pour vérifier un mesh).



Notre mesh de dague a donc survécu à un passage dans Blender via Nifskope sans soucis.

Je ne me suis pas amusé à déformer la dague ou à mettre d'autres textures car ça ne change rien à la procédure d'export du mesh et au travail à faire détaillé ci-dessus. Bon modding.

Gérauld