

ARMES ET ANIMATIONS DANS NWN2

1. Fonctionnement	2
1.1. Armes	2
1.1.1. Armes standards.....	2
1.1.2. Baseitems.2da	3
1.2. Animations	3
1.2.1. Pour les créatures.....	3
1.2.2. Pour les armes	4
2. Réparation du modèle "fouet de Balor"	6
2.1. Modèles.....	6
2.2. Animations	6
3. Création de vos propres armes	8
3.1. Modèle	8
3.2. Animations	9

1. Fonctionnement

1.

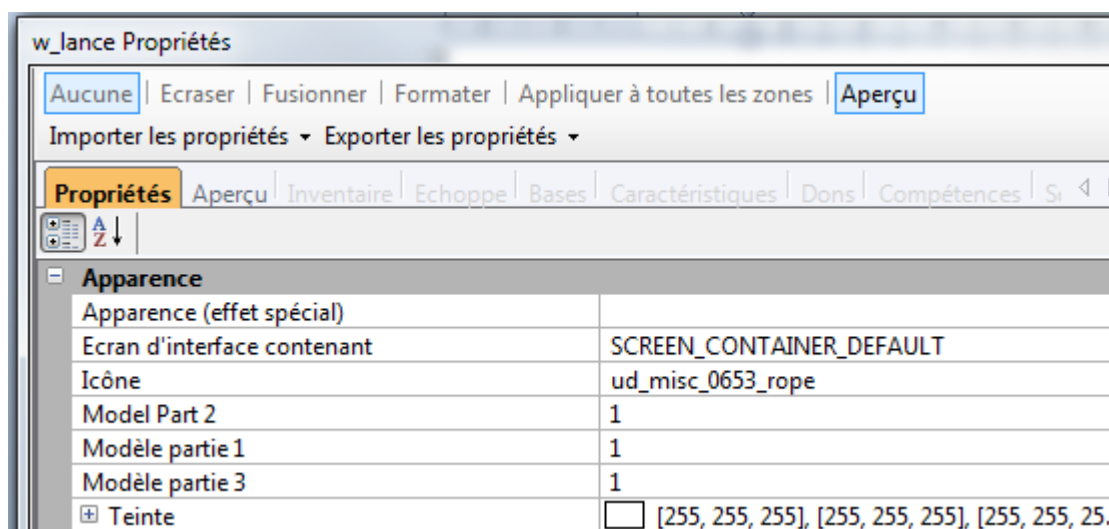
1.1. Armes

Il existe déjà de nombreux tutoriels qui expliquent comment créer des armes pour NWN2, aussi je ne rentrerai pas les détails pour ce faire. Je rappelle néanmoins les bases pour comprendre comment ajouter des armes animées dans le jeu.

1.1.1. Armes standards

Deux sortes d'armes sont disponibles : celles qui peuvent être personnalisées, en choisissant l'aspect et la forme de la poignée, de la garde et de la lame (pour une épée par exemple). Pour NWN2, ces parties sont appelées Modèle Partie 1, Model Part 2 et Modèle Partie 3 (eh oui, la traduction pour la partie 2 a apparemment été oubliée!). Pour incorporer des armes additionnelles dans le jeu, les créateurs devront suivre la convention suivante pour les noms :

- Pour une arme générique "w_arme", les différents modèles de la famille devront être désignés par "w_arme01", "w_arme23", etc...
- Chaque partie aura un nom découlant de cette désignation, avec l'ajout d'un suffixe **_a** pour la partie 1, **_b** pour la partie 2 et **_c** pour la partie 3 : dans le cas de "w_fleuret45", nous aurons donc *w_fleuret45_a* pour la lame, *w_fleuret45_b* pour la garde et *w_fleuret45_c* pour la poignée.
- Ces parties pouvant être sélectionnées indépendamment les unes des autres, il est tout à fait possible de créer une épée à partir de la lame **w_sword01_a**, de la garde **w_sword02_b** et de la poignée **w_sword03_c**. Cela implique qu'un modèle nouveau ajouté à une famille existante devrait avoir des dimensions similaires à celles des modèles déjà présents dans la famille en question.



L'alternative consiste à créer une arme d'un seul tenant (c'est-à-dire que poignée, garde et lame ne sont plus interchangeables). La convention de désignation à respecter dans ce cas est "w_arme67" (pas de lettre ajoutée à la fin).

NWN2 reconnaît si les armes sont composées d'un ou plusieurs éléments à la lecture des informations contenues dans baseitems.2da.

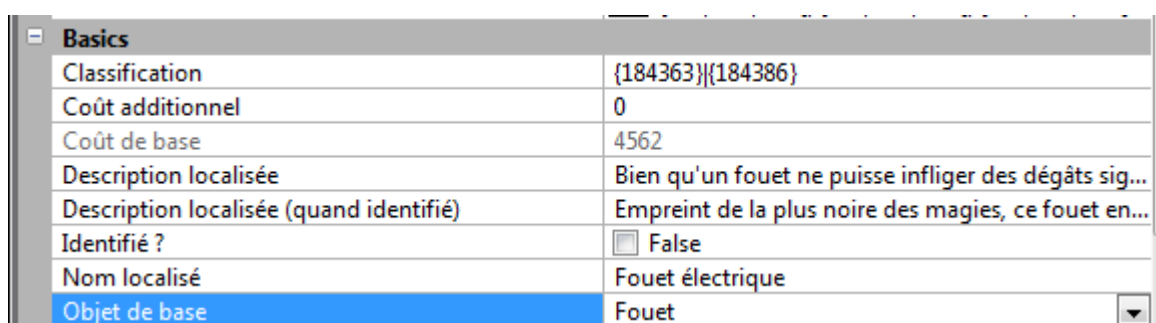
1.1.2. Baseitems.2da

Le tableau à deux dimensions (plus simplement, des données en lignes et en colonnes : chaque ligne correspond à un objet en particulier, et chaque colonne définit une propriété valable pour chaque objet) *baseitems.2da* liste toutes les caractéristiques des objets utilisées par NWN2.

Name	label	InvSlotWidth	InvSlotHeight	EquipableSlots	CanRotateIcon	ModelType	NWN2_Anim	ItemClass	GenderSpecific	Part1EnvMap	Part2EnvMap	Part3EnvMap	DefaultModel	NWN2_DefaultIcon	DefaultIcon	Container	WeaponWield	WeaponType	WeaponSize
4	176 lightflail	1	3	0x1C090	1	2	1	w_flail	0	1	1	1	it_bag	9	****	0	****	2	3
31	1525 dart	1	1	0x00010	0	0	0	w_dart	0	1	1	1	it_bag	33	****	0	****	11	1
35	**** DELETED	2	4	0x1C090	1	2	1	w_flail	0	1	1	1	it_bag	37	****	0	****	2	4
61	1555 sling	1	2	0x00090	0	0	1	w_sling	0	****	****	****	it_bag	57	****	0	****	10	2
111	83617 Whip	2	2	0x1C090	1	0	1	w_devilwhip	0	1	1	1	it_bag	64	****	0	****	0	2
201	Fouet Fouet	2	2	0x1C090	1	0	1	w_fouet	0	1	1	1	it_fouet	64	****	0	****	3	2

Ce qui nous intéresse plus particulièrement ici, ce sont :

- Label → le nom qui apparaît dans la liste déroulante “objet de base”



- ModelType → 0 pour les armes en une pièce, 2 pour celles composées de trois éléments.
- NWN2_Anim → 0 pour les objets rigides, 1 pour ceux qui disposent d’animations.
- ItemClass → nom du modèle (nom du fichier .mdb)
- GenderSpecific si l’apparence de l’objet change en fonction du genre de son utilisateur (0 = non, 1 = oui).

C’est là que toutes les informations concernant les objets que les créatures peuvent manier dans le jeu. Jetez donc un coup d’œil à la ligne 111 : c’est celle que nous allons devoir modifier pour faire fonctionner le fouet (tous les détails dans la suite du document).

1.2. Animations

Pour NWN2, les animations utilisent un squelette (qui définit les positions et liens des différents « os ») et des fichiers d’animation (qui définissent comment les os sont orientés et bougent par rapport à l’os de référence).

Ce principe s’applique aux créatures comme aux objets animés (à ne pas confondre avec les plaçables).

La seule différence, mais de taille, concerne la convention de désignation des fichiers d’animation.

1.2.1. Pour les créatures

Le corps d’une créature appelée “*c_monstre*” doit être nommé “*c_monstre_CL_Body01*” (où 01 est un numéro indiquant les différentes apparences possibles, qui peuvent être choisies dans l’onglet “ensemble d’armure” dans les propriétés de la créature).

Son squelette devra se terminer par *_skel* (en règle générale, il reprend le nom de la créature, comme “*c_monstre_skel*”, mais ce n’est pas une obligation).

Ce qui est obligatoire en revanche est que les fichiers d'animation soient nommés comme le squelette qu'ils animent (ainsi, si le squelette de "c_monstre" est "c_creature_skel", toutes ses animations devront commencer par "c_creature_").

Elles devront aussi indiquer quelle animation elles représentent :

- "Idle" pour l'animation "ne rien faire",
- 1attack01 pour l'animation d'attaque (première variante, la deuxième sera 1attack02 et la troisième 1attack03 etc...).

Enfin, elles doivent aussi préciser quelle posture (position de combat) est concernée ("1hS" pour les armes à une main, "1hSS" pour les armes à une main et bouclier, "D2h" pour les armes à deux mains etc...)

Prenons un exemple : notre créature "c_monstre", dont le corps "c_monster_cl_body03" est animé par l'intermédiaire du squelette "c_creature_skel", utilisera l'animation "c_creature_1hSS_1attack04" pour la quatrième variante d'une attaque avec une arme à une main et un bouclier.

1.2.2. Pour les armes

Notes:

- Ce qui suit découle de mes découvertes et peut n'être pas applicable dans tous les cas.
- Les armes étant décrites dans le même fichier 2da que les autres objets, il est possible que des drapeaux qui ondulent au gré du vent ou des livres dont les pages se tournent puissent être ajoutés sans qu'il soit besoin d'en faire une créature.
- Un modèle .mdb nommé selon les règles pour les créatures ("c_creature_CL_Body01") et le même modèle mais cette fois nommé selon les règles pour les armes ("w_weapon01") sont nécessaires : la "créature" pour les animations et l'arme pour l'apparence (du moins dans l'éditeur).

Pour les armes animées, il y a aussi besoin d'un squelette. Les conventions pour désigner les animations sont cependant modifiées comme suit (pour simplifier la lecture, on supposera que "w_arme" est le nom de l'arme pour laquelle les animations are envisagées) :

- En plus du préfixe "w_arme", les fichiers d'animation doivent inclure dans leur nom une lettre (pour utilisateur féminin/masculin, voir [1.1.2](#) à propos de l'apparence spécifique au genre),
- Après la lettre F/M, une seconde lettre L/R (pour utilisation avec la main gauche/dRoite, R est probablement l'animation par défaut L utilisée uniquement dans le cas du combat à deux armes). Même pour les armes à deux mains, la partie FR ou MR est requise,
- Ensuite vient la posture (_UNA_, _1hs_, etc...),
- Et enfin le type d'animation (même principes que pour les créatures: "idle", "1attack01", "death02", etc...).

Information supplémentaire : la convention ci-dessus suppose que les animations normales pour l'utilisateur sont applicables. Si le maniement d'une arme animée requiert en outre une animation particulière pour l'attaquant, elle devra bien évidemment être créée et nommée pour refléter la position de combat et l'arme utilisée. Pour la fronde par exemple, on trouve **P_??M/F_Thrw_slingattack01** où ?? est l'une des quatre possibilités pour les

joueurs [DD pour les nains ; GG pour les gnomes ; OO pour les demi-orques ; HH pour les humains, elfes, halfelins – en fait toutes les autres races jouables].

Je n'ai pas vérifié, mais cela signifie vraisemblablement que les trolls ne peuvent pas utiliser de frondes (ou plutôt, que leurs attaques avec une fronde vont certainement paraître bizarres).

Une elfe noire attaquant avec un fouet (dont le modèle est appelé "elfwhip") et maniant un bouclier devra avoir une animation **P_HHF_1hSS_elfwhipattack01** si les gestes habituels pour une épée n'étaient pas adaptés. Mais il s'agit là d'une simple supposition, je n'ai pas vérifié non plus.

La question qui reste toutefois sans réponse à ce stade est : pourquoi le fouet de Balor reste-t-il toujours rigide alors que tous les fichiers d'animation et les mdb ont été renommés comme indiqué ?

2. Réparation du modèle “fouet de Balor”

La réponse tient en ce que plusieurs exigences n'étaient pas respectées :

- Le nom du maillage à l'intérieur du fichier .mdb n'était pas le même que celui du fichier .mdb ;
- L'échelle à laquelle le fouet a été créé n'était pas la même que celle utilisée pour son squelette ;
- Les animations respectaient les conventions pour une créature, pas pour une arme (les lettres FR / FL / MR / ML n'étaient pas présentes).
- La référence indiquée dans baseitems.2da n'était pas cohérente avec le nom du squelette et du modèle.

2.

2.

2.1. Modèles

Commençons par le commencement, et faisons en sorte que baseitems.2da, le maillage, le modèle, le squelette et les animations utilisent tous le même nom. Comme nous n'allons pas refaire toutes les animations, le plus simple est d'utiliser le nom du squelette et donc de remplacer en ligne numéro 111 la référence « *w_whip* » par « *w_devilwhip* ».

Puisque le 2da indique maintenant que le fouet utilise le modèle ***devilwhip.mdb***, assurons-nous qu'il en existe bien un (*w_whip*, *w_whip01* et même *w_whipdevil* existent, mais malheureusement ils utilisent tous un maillage *w_whip* ou *w_whip_cl_body01* alors que baseitems.2da indique en toutes lettres au moteur de NWN2 de rechercher ***devilwhip.mdb***, ***devilwhip01.mdb*** [ModelType = 0, donc en un seul tenant] et ***devilwhip_cl_body01.mdb***).

Voici la procédure :

- ✓ Renommer *w_whip01.mdb* en *w_devilwhip01.mdb*.
- ✓ Ouvrir *w_devilwhip01.mdb*.
- ✓ Renommer le maillage *w_whip* en *w_devilwhip01*.
- ✓ Exporter à 5500% de la taille d'origine (inclure les points d'ancrage *hp_0*, *hp_1* et *hp_2*).

- ✓ Ouvrir *w_devilwhip.mdb*.
- ✓ Renommer le maillage *w_whip* en *w_devilwhip*.
- ✓ Exporter à la taille d'origine (pas besoin d'inclure les sphères de collision).

- ✓ Ouvrir *w_DevilWhip_CL_Body01.mdb*.
- ✓ Renommer le maillage *w_whip* en *w_devilwhip_CL_Body01*.
- ✓ Exporter à la taille d'origine (inclure les sphères de collision cette fois).

Maintenant, les fichiers relatifs au fouet de Balor sont cohérents : le modèle animé, le modèle rigide pour l'éditeur, le squelette et les animations utilisent tous le même nom “*w_devilwhip*”.

2.2. Animations

Conformément à ce qui a été expliqué précédemment, les animations devraient indiquer une posture *_1hS_* or *_D2h_* etc... Cependant, en ce qui concerne le fouet, nous allons considérer une posture *UNArmed* (« sans arme »). En conséquence, renommer toutes les

animations `w_DevilWhip_D2h_animation` en `w_DevilWhipFR_UNA_animation` permettra enfin de bénéficier du fouet de Balor dans le jeu.

Note : Je n'ai toujours pas compris pourquoi les animations pour le fouet de Balor sont supposées être UNA et pas D2h. Après tout, les animations pour la fronde et le fléau (ainsi que celles que j'ai créées pour ma champagne) respectent les règles énoncées en [1.2.2](#). J'ai toutefois remarqué que les animations pour les fouets de Balor faisaient référence à la fois à "`w_devilwhip_skel`" (ce qui est attendu !) et "`w_whip01`" (ce qui est plus surprenant...) Je suppose que ceci explique cela.

Quoi qu'il en soit, si vous exportez vos propres animations sans le maillage, il ne devrait y avoir aucun problème.

3. Création de vos propres armes

Le titre peut être trompeur : je ne vais pas expliquer comment créer une arme ou une animation, mais plusieurs tutoriaux sont disponibles – un que je trouve particulièrement détaillé est celui de Frank Perez (Elysium), qui peut être trouvé sur son blog <http://nwn2faithless.blogspot.com> (rechercher “creating new life with Expotron and max”). En revanche il est uniquement en anglais.

3.

3.

3.1. Modèle

Vous l’avez probablement déjà deviné, le fonctionnement des animations pour les armes implique que seul le modèle numéro 01 apparaîtra en jeu. Si vous ne me croyez pas, regardez à quoi ressemble un fléau dans l’éditeur :



Et comparez avec ce que vous voyez en jeu :



Effets visuels:

Pour l'instant, hp_0 et hp_1 ne donnent pas les résultats escomptés... Pour ma part il s'agit d'un bug du jeu, dans la mesure où le fléau (l'autre arme animée avec la fronde du jeu original) est lui aussi totalement affecté par l'effet appliqué, alors que seule la boule à pointes devrait l'être (regardez l'effet « fx_torchhand » sur le fléau et sur la torche, vous comprendrez).

La bonne nouvelle est que les points d'ancrage pour creature sont reconnus : ajouter un "ap_forehead" ou "ap_mouth" ou ce que vous voulez au squelette de l'arme, et créez l'effet visuel avec les mêmes points d'ancrage de référence (la création des effets visuels ne fait pas partie de ce tutoriel, mais il en existe de nombreux sur le sujet).

3.2. Animations

Comme expliqué en [section 1.2](#), il existe une différence fondamentale entre la convention pour dénommer les créatures et les armes : les animations des créatures font référence au squelette qu'elles utilisent, alors que les animations pour les armes utilisent aussi le nom du modèle.

Voici un exemple pour fixer les idées :

Les araignées spectrales ont pour modèle *C_SpidWrai_CL_Body01*, mais comme toutes les araignées du jeu utilisent le squelette *c_spid_skel* et par suite les animations *c_spid_una_idle*, *c_spid_una_1attack01*, etc....

Le fouet de Balor en revanche, utilisant le squelette *w_devilwhip_skel*, n'était pas animé au départ car son modèle était nommé *w_whip* : il y avait incohérence entre le squelette, le modèle et le maillage.

Pour résumer : seul un modèle sera visible, et le même nom doit être utilisé pour :

- Le modèle animé : **w_weapon_cl_body01** (utilisé dans le jeu)
- Le modèle rigide **w_weapon01** (utilisé dans l'éditeur pour afficher l'arme)
- Le squelette : **w_weapon_skel**

- Les fichiers d'animation : don't le nom doit commencer par "**w_weapon**" (ne pas oublier de préciser le genre et le côté avant la posture et le nom de l'animation, comme expliqué au paragraphe 1.2.2).
- Les effets visuels fonctionneront avec les ancrages "ap_xxx" et pas les habituels hp_0, hp_1, etc...