

# **La compilation des tutoriaux du net en français par NutsCracker Version 1.2**

<b>1</b>	<b>INTRODUCTION</b>	<b>8</b>
<b>2</b>	<b>UN PLUG-INS, UN MOD ?</b>	<b>8</b>
<b>2.1</b>	<b>L'INSTALLATION D'UN PLUG-IN</b>	<b>8</b>
<b>2.2</b>	<b>POUR LA DESINSTALLATION</b>	<b>9</b>
<b>2.3</b>	<b>LE TESCO</b>	<b>9</b>
<b>2.4</b>	<b>PRECAUTIONS A PRENDRE</b>	<b>9</b>
<b>3</b>	<b>L'INTERFACE - IHM</b>	<b>10</b>
<b>3.1</b>	<b>LA FENÊTRE OBJECT WINDOW</b>	<b>11</b>
3.1.1	LES ELEMENTS D'ARCHITECTURE ET D'EXTERIEUR – LES STATICS	11
3.1.2	LES CREATURES	14
3.1.3	LES PNJ (NPC)	17
3.1.4	LES TRESORS SIMPLES	22
3.1.5	ALCHEMY – ALCHEMIE	23
3.1.6	APPARATUS	24
3.1.7	ARMOR - ARMURES	24
3.1.8	BOOK	25
3.1.9	CLOTHING – LES HABITS	26
3.1.10	INGREDIENT	27
3.1.11	LOCKPICK	27
3.1.12	MISC ITEM	27
3.1.13	PROBE – LES SONDAS	28
3.1.14	REPAIR ITEM – LES OUTILS DE REPARATION	28
3.1.15	WEAPON – LES ARMES	29
3.1.16	LES CONTENEURS	30
3.1.17	LES OBJETS DE NIVEAU - LEVELED ITEMS	31
3.1.18	LES PORTES - DOORS	31
3.1.19	ACTIVATOR- LES ACTIVEURS	33
3.1.20	LES PARTIES DU CORPS - BODY PART	33
3.1.21	LES ENCHANTEMENTS - ENCHANTING	34
3.1.22	LES LUMIERES – LIGHT	35
3.1.23	LES OBJETS STATIQUES – STATIC	35
<b>3.2</b>	<b>LE FENETRE CELL VIEW</b>	<b>36</b>
<b>3.3</b>	<b>LA FENETRE DE RENDU</b>	<b>37</b>
3.3.1	MOUVEMENT DANS LA FENETRE DE RENDU	37
3.3.2	DEPLACER ET REDIMENSIONNER DES OBJETS	38
3.3.3	RACCOUCIS CLAVIER	38
<b>3.4</b>	<b>LA FENETRE PREVIEW</b>	<b>39</b>
<b>3.5</b>	<b>LA FENETRE DES REGIONS</b>	<b>40</b>
3.5.1	MODIFIER LES PROPRIETES D'UN CELLULE	41
<b>3.6</b>	<b>LES MENUS DE L'IHM</b>	<b>41</b>
3.6.1	FILE	41
EDIT		42
3.6.2	VIEW	42
3.6.3	WORLD	42

3.6.4	CHARACTER	42
3.6.5	GAMEPLAY	47

## **4 CHOISIR VOS REGLAGES** **51**

## **5 DIALOGUE** **53**

5.1	TOPIC – LES SUJETS	54
5.2	VOICE	55
5.3	GREETING – SALUTATIONS	55
5.4	PERSUASION	55
5.5	JOURNAL	55
5.6	%DEFINES TAGS	56
5.7	FUNCTION / VARIABLE	56
5.8	LA SECTION RESULT	57

## **6 LES FICHIERS DE L'EDITEUR** **59**

6.1	LES FICHIERS ESM ET ESP	59
6.2	CHARGER DES DONNEES	59
6.3	LES MODELES 3D ET LES TEXTURES	60
6.3.1	LES FICHIERS DDS	61
6.3.2	MODELISER DANS 3DS	61
	UNE AIDE AU SKINNAGE DES MODELES 3D	61
	DES SITES DE MODELES	61
	LES ICONES	61
	LES FICHIERS DE SON	61
	OU SE TROUVENT LES FICHIERS DU JEU?	62
	L'ORGANISATION DES FICHIERS	62
	LE FICHIER ESM	62
	LE SYSTEME DE NOMMAGE	62
	LA STRUCTURE DES REPERTOIRES	62

## **7 LA CREATION DES INTERIEURS** **64**

7.1	UN OBJET POUR DES REFERENCES	64
7.2	CREER UN CELLULE D'INTERIEUR	65
7.3	LA STRUCTURE DE BASE	66
7.4	LA CONSTRUCTION DE LA CELLULE	68
7.4.1	LE DEBUT	69
7.4.2	L'AJOUT DES ELEMENTS DE L'ARCHITECTURE	70
7.4.3	LA CONSTRUCTION DE LA CELLULE	71
7.4.4	ASPECT GENERAL	71
7.4.5	REMARQUES SUR LES PERFORMANCES	72
7.4.6	LES LUMIERES	73
7.4.7	LIER L'INTERIEUR A L'EXTERIEUR	74
7.4.8	ON AURAIT PU FAIRE PLUS SIMPLE	75
7.5	PEUPLER ET EQUIPER NOTRE CELLULE	75
7.6	PATH GRID – EDITION ET MODIFICATION	75
7.7	CONNAITRE LE CONTENU DE LA CELLULE	79

<b>7.8</b>	<b>INONDER UNE CELLULE D'INTERIEUR</b>	<b>80</b>
<b>7.9</b>	<b>REMARQUES POUR LES CREATEURS DE STATIC</b>	<b>80</b>
<b>7.10</b>	<b>UN OUTIL POUR ACCELERER LE DEVELOPPEMENT DES CELLULES : GENMOD</b>	<b>80</b>
7.10.1	OU LE TROUVER?	81
<b>8</b>	<b><u>CREER DES CELLULES D'EXTERIEUR</u></b>	<b>82</b>
<b>8.1</b>	<b>FORMER LE PAYSAGE LE PAYSAGE</b>	<b>82</b>
8.1.1	L'ALTITUDE - HEIGHT	84
8.1.2	TEXTURE	84
8.1.3	LES COULEURS - VERTEX COLOR	85
<b>8.2</b>	<b>AJOUTER DES OBJETS, DES CREATURES, DES LUMIERES OU DES PNJ.</b>	<b>85</b>
<b>8.3</b>	<b>DEFINIR LE CLIMAT DE LA CELLULE</b>	<b>86</b>
8.3.1	CREER UN CLIMAT PERSONNALISE	86
8.3.2	MODIFIER LES PROPRIETES D'UN CELLULE	88
<b>8.4</b>	<b>N'OUBLIEZ PAS DE TESTER...</b>	<b>88</b>
<b>9</b>	<b><u>LA CREATION DES PNJ PAS A PAS.</u></b>	<b>89</b>
<b>9.1</b>	<b>DES PRECAUTIONS A PRENDRE</b>	<b>89</b>
<b>9.2</b>	<b>LA CREATION DE L'APPARENCE DU PNJ</b>	<b>89</b>
<b>9.3</b>	<b>EQUIPER VOTRE PNJ</b>	<b>92</b>
9.3.1	UNE QUANTITE INFINIE D'OBJET	92
9.3.2	UNE PRECAUTION A PRENDRE	92
<b>9.4</b>	<b>LES PARAMETRES I.A.</b>	<b>92</b>
9.4.1	L'ONGLET PACKAGE	92
9.4.2	L'ONGLET SERVICES	98
<b>9.5</b>	<b>ANIMATION</b>	<b>102</b>
<b>10</b>	<b><u>DIALOGUE/JOURNAL-BASED QUEST CREATION</u></b>	<b>103</b>
<b>10.1</b>	<b>L'IDEE DE BASE</b>	<b>103</b>
<b>10.2</b>	<b>L'ENTREE EN MATIERE DE RA'VIRR</b>	<b>104</b>
<b>10.3</b>	<b>CREER UN TOPIC – UN SUJET DE DISCUSSION.</b>	<b>105</b>
10.3.1	AJOUTER UN TOPIC	106
<b>10.4</b>	<b>FILTRE LES PNJ QUI PEUVENT DIRE UNE PHRASE</b>	<b>107</b>
<b>10.5</b>	<b>POSER UNE QUESTION AU JOUEUR, LUI OFFRIR UN CHOIX</b>	<b>107</b>
<b>10.6</b>	<b>INTERPRETER UN CHOIX DU JOUEUR</b>	<b>108</b>
<b>10.7</b>	<b>ECRIRE DANS LE JOURNAL</b>	<b>109</b>
10.7.1	LA STRUCTURE DU JOURNAL	109
10.7.2	AJOUTER UNE ENTREE DANS LE JOURNAL	110
10.7.3	LA FAIRE APPARAITRE DANS LE JOURNAL DU JOUEUR	110
<b>10.8</b>	<b>FAIRE EVOLUER LA QUETE</b>	<b>110</b>
<b>10.9</b>	<b>FINALISER LA QUETE</b>	<b>111</b>
<b>10.10</b>	<b>DERNIERS CONSEILS</b>	<b>111</b>
<b>11</b>	<b><u>SCRIPTING – LES SCRIPTS</u></b>	<b>112</b>
<b>11.1</b>	<b>PRESENTATION GENERALE</b>	<b>112</b>
<b>11.2</b>	<b>LE TESCO ET LA PROGRAMMATION ORIENTEE OBJET</b>	<b>113</b>
11.2.1	QUELQUES CONCEPT D'OBJET	113

11.2.2	LE TESCS ET LES OBJETS	114
<b>11.3</b>	<b>DEUX OU TROIS CHOSES A SAVOIR</b>	<b>114</b>
11.3.1	L'EXECUTION DES SCRIPT	114
11.3.2	LES VARIABLES	114
<b>11.4</b>	<b>CONVENTIONS DE BASE</b>	<b>114</b>
11.4.1	ASSIGNER UNE VALEUR A UNE VARIABLE	115
11.4.2	UTILISER DES VALEURS ALEATOIRES	115
11.4.3	UTILISER LES FONCTIONS	115
11.4.4	VIRGULES, ESPACES ET PARENTHESES	116
11.4.5	LES TESTS IF	116
11.4.6	LE TEST WHILE (TANT QUE)	117
11.4.7	LES COMMENTAIRES	117
<b>11.5</b>	<b>LES OPERATEURS BOOLEENS</b>	<b>117</b>
<b>11.6</b>	<b>SCRIPTS LOCAUX – SCRIPTS GLOBAUX</b>	<b>118</b>
11.6.1	MODIFICATION DE VERSION D'UN MOD ET SCRIPT GLOBAL	118
<b>11.7</b>	<b>DU BON USAGE DU BOUTON SAVE DE LA FENETRE D'EDITION</b>	<b>118</b>
<b>11.8</b>	<b>LES FONCTIONS RECONNUES DANS LES SCRIPTS</b>	<b>118</b>
<b>11.9</b>	<b>LA MACHINE A ETATS</b>	<b>119</b>
<b>11.10</b>	<b>DES ASTUCES ET DES CHOSES A CONNAITRE POUR FAIRE DES SCRIPTS</b>	<b>119</b>
11.10.1	COMPTER LE NOMBRE D'EXECUTIONS	119
11.10.2	LES CHRONOMETRES	120
11.10.3	SI LE JOUEUR EST DANS LE MODE MENU	121
<b>11.11</b>	<b>DEBUGGER UN SCRIPT</b>	<b>121</b>
<b>11.12</b>	<b>LES DEPLACEMENTS DES PNJ</b>	<b>122</b>
11.12.1	FAIRE SE DEPLACER UN PNJ VERS UNE NOUVELLE DESTINATION	122
11.12.2	VERIFIER QU'UN PNJ A BIEN EXECUTE SON MOUVEMENT	122
11.12.3	MOUVEMENT ALEATOIRE D'UN PNJ OU D'UNE CREATURE	124
11.12.4	FAIRE SUIVRE UN ACTEUR PAR UN AUTRE	124
11.12.5	VERIFIER LA NATURE DU PACKAGE IA QUI S'EXECUTE	125
11.12.6	FORCER UN MOVEMENT DISCRET	125
11.12.7	AMMENER UN ACTEUR OU UN OBJET A UNE NOUVELLE POSITION	126
<b>11.13</b>	<b>TEXTES ET DIALOGUE</b>	<b>127</b>
11.13.1	AFFICHER UN MESSAGE ET DES CHOIX	127
11.13.2	AJOUTER UN TOPIC DANS LES DIALOGUES	128
11.13.3	AJOUTER DES ENTREES DANS LE JOURNAL ET VERIFIER SON CONTENU.	128
<b>11.14</b>	<b>COMBAT</b>	<b>129</b>
11.14.1	DEBUTER ET FINIR UN COMBAT	129
11.14.2	DETECTER UNE ATTAQUE DE LA PART DU JOUEUR	130
11.14.3	CONNAITRE LA PRIME SUR LA TETE D'UN JOUEUR	130
<b>11.15</b>	<b>LA MAGIE</b>	<b>130</b>
11.15.1	INVOCATION DE SORTS	130
11.15.2	GERER ET TESTER LES SORTS	132
11.15.3	GERER ET TESTER LES EFFETS DES SORTS	132
<b>11.16</b>	<b>LES BRUITAGES</b>	<b>132</b>
11.16.1	JOUEUR UN SON	132
<b>11.17</b>	<b>GARDER UNE TRACE DU PASSAGE DU TEMPS</b>	<b>133</b>
11.17.1	CHRONOMETRE	133
11.17.2	CONNAITRE L'HEURE DU JOUR	133
11.17.3	GARDER UNE TRACE DU PASSAGE DES JOURS	134
11.17.4	LES PHASES DE LA LUNE	134
<b>11.18</b>	<b>UTILISER LES OBJETS</b>	<b>134</b>
11.18.1	UTILISER L'INVENTAIRE	134
11.18.2	VERIFIER QU'UN OBJET EST UTILISE ET L'UTILISER	137
11.18.3	VEROULLER ET DEVEROULLER DES PORTES ET DES COFFRES	137
11.18.4	ACTIVER ET DESACTIVER UN OBJET	138

11.18.5	DEPLACER ET FAIRE TOURNER DES OBJETS	139
11.18.6	DETERMINER LA LOCALIZATION	141
<b>11.19</b>	<b>STATISTIQUES ET ATTRIBUTS</b>	<b>143</b>
<b>11.20</b>	<b>LES CONTROLES DU JOUEUR</b>	<b>144</b>
<b>11.21</b>	<b>LE CLIMAT</b>	<b>145</b>
11.21.1	CHANGER LE TEMPS QU'IL FAIT	145
11.21.2	MODIFIER LES PARAMETRES METEO D'UNE REGION	145
11.21.3	CONNAITRE LE TEMPS COURANT.	145
11.21.4	CONNAITRE LA VITESSE DU VENT	146
<b>11.22</b>	<b>DIVERSES FONCTIONS ET VARIABLES</b>	<b>146</b>
11.22.1	SAVOIR SI LE JOUEUR A OUVERT UN MENU	146
11.22.2	SAVOIR SI LE JOUEUR EST A L'INTERIEUR	146
11.22.3	INTERROMPRE L'EXECUTION D'UN SCRIPT	147
<b>11.23</b>	<b>CONTROLLER LES SCRIPTS GLOBAUX</b>	<b>147</b>
<b>11.24</b>	<b>TRUCS ET ASTUCES</b>	<b>147</b>
11.24.1	PERMETTRE A UN ACTEUR DE CHANGER D'ARMES (GHAN BURI GHAN – GRAND MAITRE SCRIPTEUR)	147
11.24.2	UNE SEQUENCE CINÉMATIQUE- GIANLUCA (MORROWIND SUMMIT FORUMS)	149
<b><u>12</u></b>	<b><u>FUNCTION / VARIABLE</u></b>	<b><u>151</u></b>
<b>12.1</b>	<b>FUNCTION – LES FONCTIONS</b>	<b>151</b>
<b>12.2</b>	<b>GLOBAL – LES VARIABLES GLOBAL</b>	<b>153</b>
<b><u>13</u></b>	<b><u>INTERPRETATION DE CERTAINES VALEURS DONNEES PAR LES FONCTIONS</u></b>	<b><u>154</u></b>
<b>13.1</b>	<b>A PROPOS DU TEMPS</b>	<b>154</b>
<b>13.2</b>	<b>LES RACES</b>	<b>154</b>
<b>13.3</b>	<b>LES CLANS DE VAMPIRES</b>	<b>154</b>
<b>13.4</b>	<b>LES PACKAGES IA</b>	<b>154</b>
<b>13.5</b>	<b>LES PHASES DE LA LUNE</b>	<b>154</b>
<b>13.6</b>	<b>LES SIGNIFICATIONS DES COEFFICIENTS DE WANDER (PACKAGE IA)</b>	<b>155</b>
<b>13.7</b>	<b>LES SALUTATIONS (GREETINGS)</b>	<b>155</b>
<b><u>14</u></b>	<b><u>LES EFFETS DES SORTS - SPELLEFFECT</u></b>	<b><u>156</u></b>
<b><u>15</u></b>	<b><u>LES FONCTIONS DE SCRIPT</u></b>	<b><u>158</u></b>
<b>15.1</b>	<b>I.A.</b>	<b>158</b>
<b>15.2</b>	<b>ANIMATION</b>	<b>159</b>
<b>15.3</b>	<b>COMBAT</b>	<b>159</b>
<b>15.4</b>	<b>DIALOGUE</b>	<b>159</b>
<b>15.5</b>	<b>FACTION</b>	<b>160</b>
<b>15.6</b>	<b>ITEM/OBJECT</b>	<b>161</b>
<b>15.7</b>	<b>LES VARIABLES LOCALES POUR L'INVENTAIRE - INVENTORY LOCAL VARIABLES</b>	<b>161</b>
<b>15.8</b>	<b>MAGIC</b>	<b>161</b>
<b>15.9</b>	<b>DIVERS - MISCELLANEOUS</b>	<b>162</b>
<b>15.10</b>	<b>LES MOUVEMENTS ET LES DEPLACEMENTS</b>	<b>163</b>
<b>15.11</b>	<b>LES CONTROLES DU JOUEURPLAYER CONTROLS</b>	<b>164</b>
<b>15.12</b>	<b>LE REPOS DU GUERRIER</b>	<b>165</b>

<b>15.13</b>	<b>LE TEMPS QUI PASSE</b>	<b>165</b>
<b>15.14</b>	<b>LES TESTS SUR LES SCRIPTS</b>	<b>165</b>
<b>15.15</b>	<b>LES BRUITAGES</b>	<b>165</b>
<b>15.16</b>	<b>STATISTIQUES</b>	<b>166</b>
<b>15.17</b>	<b>LA PLUIE ET LES TEMPETES DE CENDRE</b>	<b>166</b>
<b>15.18</b>	<b>CONSOLE (DANS LE JEU UNIQUEMENT)</b>	<b>166</b>

---

## **16 FAIRE UN MOD PROPRE** **169**

<b>16.1</b>	<b>UN MOD “SALE”</b>	<b>169</b>
<b>16.2</b>	<b>ANALYSE DES MODIFICATIONS FAITES</b>	<b>169</b>
<b>16.3</b>	<b>LES FONCTIONS DE TEST FOURNIES AVEC LE TESCS</b>	<b>170</b>
16.3.1	MODEL TEST – LE TEST DES OBJETS 3D	170
16.3.2	DIALOGUE AND CONFLICTS – LES PROBLEMES DANS LES DIALOGUES.	171
16.3.3	NON TELEPORTING DOOR – LES PORTES NON RELIEES A DES CELLULES	171
16.3.4	NON TELEPORTING INTERIORS – LES INTERIEURS ISOLES	171
16.3.5	TEST ALL LANDS – VERIFICATION DES PAYSAGES	171
16.3.6	TEST/FIX ALL PATH GRID	171
16.3.7	TEST ALL CELLS	171
16.3.8	TEST ALL INTERIOR CELL	171
<b>16.4</b>	<b>UTILISER UN EDETEUR HEXADECIMAL POUR NETTOYER UN MOD</b>	<b>172</b>
<b>16.5</b>	<b>TESAME – THE ELDER SCROLL ADVANCED MOD EDITOR</b>	<b>174</b>
16.5.1	LES FICHIERS DU TESAME	175
16.5.2	PRESENTATION GENERALE	175
16.5.3	UTILISATION	175
16.5.4	MERGE – FUSION ET COMPARAISON	176
16.5.5	EDITION DE L’EN TETE	176
<b>16.6</b>	<b>UN FAÇON D’EVITER DE MODIFIER LE SCRIPT PRINCIPAL.</b>	<b>176</b>
<b>16.7</b>	<b>LES PNJ ET VOS MODS</b>	<b>176</b>
<b>16.8</b>	<b>NETTOYER UNE SAUVEGARDE FAITE AVEC UN MOD SALE.</b>	<b>176</b>

---

## **17 FORMAT DES FICHIERS ESP DE MORROWIND** **177**

<b>17.1</b>	<b>PRESENTATION GENERALE</b>	<b>177</b>
<b>17.2</b>	<b>LES TYPES D’OBJETS</b>	<b>177</b>
<b>17.3</b>	<b>DETAILS</b>	<b>178</b>

# 1 Introduction

Vous venez d'avoir une idée pour faire un plug. Que devez vous faire ?

Cela fait maintenant un certain temps que vous entendez parler de cet outil le TESC (The Elder Scroll Construction Set). On vous dit qu'il est utilisé par les moddeurs du monde entier pour faire les mods disponibles sur le net.

Vous écrivez votre idée sur papier. Et vous ouvrez cet éditeur..... Horreur. C'est quoi toutes ces fenêtres. Le moins que l'on puisse dire, c'est que l'utilisation n'en a pas l'air simple.

Se présente alors à vous deux alternatives, dans la première, vous mettez mon idée au placard et vous attendez quelqu'un d'autre fasse le travail. La seconde.... Vous vous jetez la tête la première dans cet éditeur et apprenez sur le tas.

Bon c'est vrai qu'il existe de nombreux tutoriels sur le Net, mais beaucoup sont dans la langue de shakespeare... et votre niveau en anglais n'est pas top.

Il existe aussi des forums sur lesquels vous pourrez trouver des infos, mais c'est un peu éclaté, dispersé partout.

Ouf voici votre grill....

Une compilation traduite dans la langue de Molière qui nous explique le fonctionnement de TESC.

Ce document est assez conséquent. J'y ai regroupé des informations éparses que j'ai trouvées sur le net. Synthétiser et restructurer pour reproduire ça.

Evidemment, tout n'est pas de moi et les sources d'inspiration sont nombreuses. (vous les trouverez dans la partie référence de ce document).

Ne me jetez pas la pierre si vous reconnaissez votre travail (DragonSong ou d'autres), je ne vous plagie pas mais vous admire.

Une rubrique contient les liens vers vos documents originaux.

Bien évidemment, j'attends vos remarques et vos contributions. Alors n'hésitez pas, envoyez moi un mail : [nutscracker\\_mw@yahoo.fr](mailto:nutscracker_mw@yahoo.fr)

NutsCracker

## 2 Un plug-ins, un mod ?

Un plug-ins (ou mod) est un petit morceau de logiciel qui permet d'enrichir un logiciel par l'ajout de fonctionnalité non prévues par les développeurs initiaux.

Dans le cas de Morrowind, cela peut-être des créatures, des quêtes des maisons ou quoi que ce soit que votre imagination puisse inventer. Morrowind est basé sur le principe des jeux open-ended (fin-ouverte) qui ne se terminent jamais. Vous évoluez dans un monde imaginé par les développeurs du jeu, et lorsque vous pensez avoir tout découvert vous découvrez sur Internet qu'il existe de nombreuses extensions (toutes gratuites, à cause d'une clause légale dans le jeu initial) qui vous permettent de découvrir de nouvelles régions, de faire de nouvelles quêtes etc...

### 2.1 L'installation d'un plug-in

La plupart des mods que vous trouverez en téléchargement sont des fichiers compressés qui contiennent :

- un fichier esp
- un readme
- un (ou des) sous répertoires

Il vous suffit de décompresser ce fichier dans votre répertoire d'installation de Morrowind, de façon que le fichier esp soit dans le répertoire Data Files.

Ensuite lancer le jeu par le raccourci habituel et dans la fenêtre de lancement sélectionner l'option Data Files, vous pouvez maintenant voir tous les plugs disponibles. Pour jouer avec un plug spécifique, il faut le sélectionner. Et voilà c'est aussi simple que ça...

Le fichier esp contient les modifications apportées au jeu initial par le moddeur.

Le readme contient une description du plug.

Les sous répertoires contiennent des modèles, des textures ou tout autre chose ajoutée par le moddeur.

## 2.2 *Pour la désinstallation*

Si vous ne voulez tout simplement plus jouer avec ce mod, il suffit de le désélectionner dans la liste des plug actif (accessibles par Data files au lancement du jeu)

Si en revanche vous désirez l'enlever de votre disque, il s'agit d'une opération délicate car il faut enlever tous les fichiers du plug (ce que vous ne connaissez pas toujours). Faites de toutes façons attention à l'avoir désélectionner dans la liste des plug actifs.

## 2.3 *Le TESCS*

TESCS est l'acronyme pour The Elder Scroll Construction Set. C'est l'éditeur fourni par Bethesda pour que la communauté des joueurs de MW puisse créer ses propres mods.

Il s'agit d'un outil extrêmement puissant une fois qu'on a compris comment s'en servir. Ceci est destiné à vous expliquer comment on s'en sert.

Alors bon courage et bonne lecture.

Et surtout n'oubliez pas la seule limite du TESCS est votre imagination.

Si vous ne trouvez pas l'info voulue dans ces pages posez la question sur les forums des développeurs (après avoir utilisé la fonction search) et une fois que vous aurez la réponse prévenez moi pour que je fasse évoluer ceci.

## 2.4 *Précautions à prendre*

Cette précaution vaut aussi bien pour les moddeurs que pour les utilisateurs de mods. Il faut TOUJOURS avoir une sauvegarde en double.

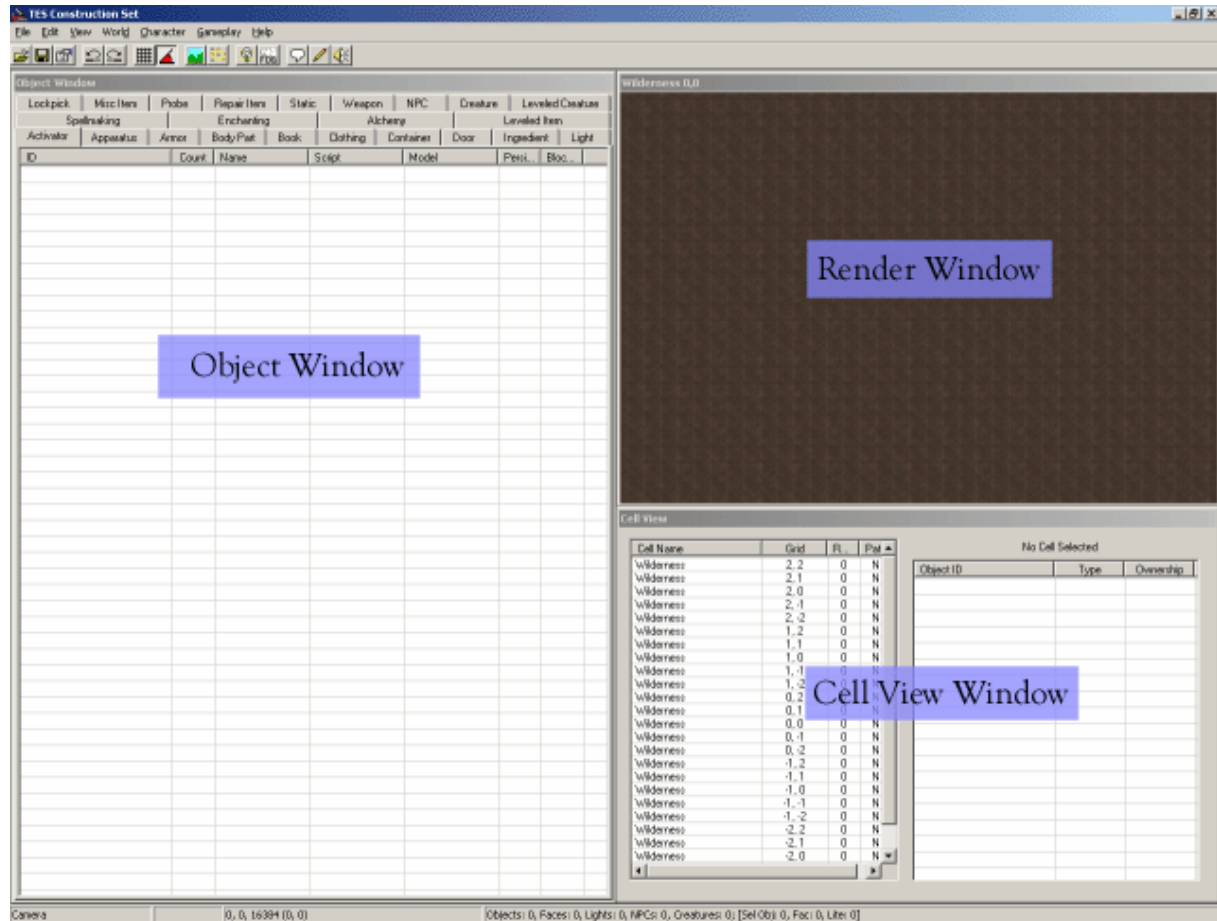
Pour les moddeurs, cela semble évident que vous ne voulez pas détruire votre partie en cours avec un plug que vous êtes en train de réaliser. Ayez donc une sauvegarde que vous utilisez spécialement pour tous vos tests. Personnellement, j'en utilise en permanence deux, l'une pour tester des fonctions dont le fonctionnement ne m'est pas parfaitement connu, et l'autre pour le test du plug en cours de développement.

D'une manière générale, je vous supplie tous de ne pas rendre disponible des mods dont vous n'êtes pas certains qu'ils ne vont pas nuire au fonctionnement du jeu. (un chapitre de ce document explique et présente des méthodes de test et de nettoyage d'un mod). Par ailleurs, le développement d'un mod, comme tout travail utilisant un ordinateur est sujet au plantage de ce dernier. Ayez donc une sauvegarde de votre mod en cours de développement (par rapport à la version que vous êtes en train d'éditer). Cela présente l'autre avantage d'avoir une base saine lorsqu'une modification provoque une destruction complète de l'univers.

Pour les joueurs.... Et bien..... il y a fort à parier que tout le monde n'aura pas lu ce tutorial. (humour) et peut être que ce que vous allez installer risque de détruire votre jolie sauvegarde de votre perso à 300 heures de jeu... ayez donc une sauvegarde de secours.

### 3 L'interface - IHM

Le TESCS (The Elder Scrolls Construction Set) est divisé en quatre parties. Tout d'abord, il ya une barre standard qui vous permet d'éditer, sauvegarder, ouvrir les data files, de régler vos préférences, de modifier les races, les aptitudes, etc...



La fenêtre "**Object Window**" vous montre tous les objets du fichier maître courant ( le fichier esm). Ceux ci sont répartis en différentes catégories:

**Activator** Ceux sont les objets qui fonctionnent comme des déclencheurs (par ex : enseigne de magasins – si vous vous approchez un message apparaît).

**Alchemy** les potions prédéfinies et déjà existantes dans le jeu. Par opposition à celles que le joueur crée durant sa partie.

**Apparatus** les ustensiles pour l'alchimie (Mortier, calcineur etc..)

**Armor** tous les différents types d'armures

**Body Part** (pas utilisé dans le TESC) les différentes parties qui constituent un modèle.

**Book** les livres et les ouvrages présents dans le jeu ( on y trouve aussi les parchemins)

**Clothing** les robes et autres habits (robes, chaussures, maillot, ...)

**Container** tous les objets qui sont susceptibles d'en contenir d'autre – les urnes, tonneau, buisson, fleurs, ...

**Creature** les ennemis que le joueur peut rencontrer

**Door** les différentes portes – qui correspondent aux différents styles architecturaux.

**Enchanting** les effets d'enchantement possibles

**Ingredient** les ingrédients ( fleurs, pétales, cendres...)

**Leveled Creature** les créatures un peu spéciales: plus un joueur est de niveau élevé plus les ennemis qu'ils rencontrent sont difficiles à tuer.

**Leveled Item** l'équivalent des créatures de niveau, mais pour les objets.

**Light** les sources de lumières, torches, lampion

**Lockpick** les pics de crochetages, ceux qu'on utilise pour forcer des serrures

**Miscellaneous** tout ce qui ne peut pas rentrer dans les autres catégories (par ex : or, les ustensiles comme les fourchettes, clés)

**NPC** (Non Player Character) équivalent de PNJ, tous les personnages que l'on est amené à rencontrer dans le monde

**Probe** toutes les sondes, ce sont les accessoires utilisés pour désamorcer les pièges

**Repair Item** les outils pour la réparation des armes et des armures ( marteau,

**Spellmaking** les capacités spéciales, maladies, enchantement fabrication de sorts

**Static** les objets statiques, c'est à dire ceux qui sont là surtout pour la décoration (de plus amples détails au 4.1)

**Weapon** les armes

Dans chaque catégories on voit les ID des objets, le nombre de fois où ils sont présents dans le monde et d'autres informations qui dépendent de la catégorie ( par exemple le poids, un script, le nom dans le monde...).

La fenêtre "**Cell View**" monstre toutes les cellules du maître courant et des plugs qui lui sont rattachés. Cela incluse leur nom, leur position sur la carte du monde (si ce sont des cellules extérieurs) et le nombre de "references" (cela sera expliqué plus tard) qu'elles contiennent. "Path" indique si la cellule dispose – ou non – d'une grille de déplacement ( Path Grid). De plus il y a la liste de tous les objets que la cellule actuellement active contient. Cela signifie que l'on y trouve leur ID, le type de l'objet, le NPC propriétaire (si ils sont la propriété de quelqu'un).

La fenêtre de rendu « **Render Window** » montre la cellule actuellement chargée. C'est dans cette fenêtre que l'on fait toute les ajouts, les déplacements et les modifications sur les objets.

Il y a deux astuces à connaître, elles existent dans tous les logiciels ou presque, mais ici c'est vraiment pratique. Le TESCS représente un ensemble très riche d'informations, et il est très facile de s'y perdre. Heureusement, il existe une fonction de recherche. Vous rentrez l'ID de l'objet recherché et la cellule qui contient la référence est automatiquement chargée et la vue est centrée sur l'objet.

L'autre astuce vous aidera à trouver plus vite les objet que vous cherchez : la plupart des objets sont représentées dans des tableaux, ces tableaux peuvent devenir très vite très grand, vous pouvez classer les cellules par ordre alphabétique (ou inverse) en cliquant sur l'intitulé de la commande.

### 3.1 La fenêtre Object Window

Dans cette fenêtre se trouvent de nombreux onglets. Chacun correspond une catégorie des objets qui constituent l'environnement de MW. Chacun de ces objets peut exister en de multiples endroits de l'univers de MW, chacune de ces existences est en fait une référence à la définition de l'objet telle qu'elle se trouve dans cette fenêtre, ainsi les modifications faites sur une référence se répercutent instantanément sur toutes les références qui existent dans le monde de MW.

Nous détaillons maintenant les informations pour chacune de ces catégories.

#### 3.1.1 Les éléments d'architecture et d'extérieur – les statics

Dans tous les mondes que vous allez créer, il existe des constructions, ou des ruines de construction. Il est important d'assurer une certaine cohérence au niveau des styles architecturaux que vous créer. Vous pouvez également utiliser les éléments existants.

Avant de vous lancer dans une création, quelle qu'elle soit, je vous conseille d'avoir au moins une petite idée, dans le meilleur des cas un dessin de votre projet.

Vous devez tout d'abord vous intéressé au style architectural de votre cellule, sera ce une hutte de Ashlander (Cendrés) un palace Telvani ou encore une grotte?

Nom de l'architecture	Exemple dans le monde
Ashlander	Les quatre campements des Ashlanders : Urshilaku Ahemmusa, Zainab, Erabeninsum
Bonecave	Utilisé pour toutes le grottes du monde
Common	L'architecture impériale standard : Pelagiad, Caldera
Daedric	L'architecture inimitable des ruines daedric Ald Sotha, Bal Fel...
Dwarven	Utilisée pour les ruines des constructions du peuple Dwener Arkngthand, Dagoth Ur

Hlaalu	Ce qu'on peut voir a Balmora ou Suran
Imperial Fort	Les forts de l'empire : Buckmoth, Moonmoth
Lavacave	Les grottes d'origine volcanique
MoldCave, MudCave	Les grottes standard, une différence dans les textures appliquées sur les parois
Nord	Architecture nordique : Dagon Fel
pycave	Autre grotte d'origine volcanique mais la roche est plus sombre (un peu comme du basalte)
Redoran	Ald'ruhn ou Maar Gan
Sewer	Les égouts comme à Vivev
Stronghold, Strongruin	Les anciennes forteresses des dunmer : Kogoruhn, Hlormaren
Telvani	Ils sont fous, ils font pousser leur maison, c'est génial Sadrith Mora
Veloithi	Les tombes ancestrales dans les quartiers de Vivec

Toutes les pièces sont créées par drag'n drop.

Sélectionner l'onglet 'Static' dans la fenêtre des objets. Vous avez maintenant la liste de tous les static utilisables dans le monde. Les objets Static, sont des objets avec lesquels le PJ ne peut pas interagir excepté pour se cogner, les murs, les tables, les chaises...

Il existe plusieurs catégories de static. Elles sont identifiables par l'ID qui apparaît dans la colonne de gauche. Cette ID commence par un mot qui caractérise la catégorie que l'on traite.

Pour que ce que vous construisez soit réutilisables par la communauté des moddeurs, essayez de faire en sorte que le l'ID que vous attribuez à vos créations corresponde à ce que les développeurs de Bethesda ont mis en place.

Object Window					
Activator		Apparatus	Armor	Body Part	Book
Clothing		Container	Door	Ingredient	Light
Lockpick		Misc Item	Probe		
Repair Item	Static	Weapon	NPC	Creature	Leveled Creature
Spellmaking		Enchanting	Alchemy	Leveled Item	
ID	Count	Model	Persists	Bloc...	
ex_vivec_lp_02	31	x\Ex_Vivec_lp_02.NIF	no	no	
Ex_Vivec_P_water_01	1	x\Ex_Vivec_P_water_01.NIF	no	no	
ex_vivec_palace_01	0	x\Ex_Vivec_palace_01.NIF	no	no	
Ex_Vivec_pd_01	1	x\Ex_Vivec_pd_01.NIF	no	no	
Ex_Vivec_pq_01	1	x\Ex_Vivec_pq_01.NIF	no	no	
Ex_Vivec_pq_02	1	x\Ex_Vivec_pq_02.NIF	no	no	
Ex_Vivec_pq_03	1	x\Ex_Vivec_pq_03.NIF	no	no	
Ex_Vivec_pq_04	1	x\Ex_Vivec_pq_04.NIF	no	no	
Ex_Vivec_pqs_01	1	x\Ex_Vivec_pqs_01.NIF	no	no	
Ex_Vivec_pqs_02	1	x\Ex_Vivec_pqs_02.NIF	no	no	
ex_vivec_prisonmoon_01	1	x\Ex_Vivec_prisonmoon_01.NIF	no	no	
Ex_Vivec_ps_01	1	x\Ex_Vivec_ps_01.NIF	no	no	
ex_vivec_sbase_01	1	x\Ex_Vivec_Sbase_01.NIF	no	no	
ex_vivec_telt_01	1	x\Ex_Vivec_TelT_01.NIF	no	no	
ex_vivec_w_01	70	x\Ex_Vivec_W_01.NIF	no	no	
ex_vivec_w_02	0	x\Ex_Vivec_W_02.NIF	no	no	
ex_vivec_w_03	1	x\Ex_Vivec_W_03.NIF	no	no	
ex_vivec_w_c_01	76	x\Ex_Vivec_W_C_01.NIF	no	no	
ex_vivec_w_e_01	94	x\Ex_Vivec_W_E_01.NIF	no	no	
ex_vivec_w_g_01	47	x\Ex_Vivec_W_G_01.NIF	no	no	
ex_vivec_w_slope_01	0	x\Ex_Vivec_W_slope_01.NIF	no	no	
ex_vivec_waterspout_01	70	x\Ex_Vivec_waterspout_01.NIF	no	no	
ex_vivec_waterspout_02	446	x\Ex_Vivec_waterspout_02.NIF	no	no	
ex_vivec_waterspout_03	18	x\Ex_Vivec_waterspout_03.NIF	no	no	
Ex_Vivec_waterspout_05	8	x\Ex_Vivec_waterspout_05.NIF	no	no	
Ex_Vivec_wspout_d_02	8	x\Ex_Vivec_wspout_d_02.NIF	no	no	
ex_wg_cave_entrance_01	2	x\Ex_WG_cave_entrance_01.NIF	no	no	
flora_ash_grass_b_01	1065	\Flora_Ash_Grass_B_01.NIF	no	no	
Flora_Ash_Grass_R_01	786	\Flora_Ash_Grass_R_01.NIF	no	no	
flora_ash_grass_w_01	2049	\Flora_Ash_Grass_W_01.NIF	no	no	
flora_ash_log_01	40	x\Flora_ash_log_01.nif	no	no	
flora_ash_log_02	110	x\Flora_ash_log_02.nif	no	no	
flora_ash_log_03	78	x\Flora_ash_log_03.nif	no	no	
flora_ash_log_04	51	x\Flora_ash_log_04.nif	no	no	
Flora_ashtree_01	441	x\Flora_ashtree_01.NIF	no	no	
flora_ashtree_02	442	x\Flora_ashtree_02.NIF	no	no	
flora_ashtree_03	660	x\Flora_ashtree_03.NIF	no	no	
flora_ashtree_04	1078	x\Flora_ashtree_04.NIF	no	no	
flora_ashtree_05	735	x\Flora_ashtree_05.nif	no	no	

**dwrv** parties d'une grande statue

**ex** toutes les apparences extérieurs des constructions, elles sont en plus réparties en fonction du style architectural.

**flora** toutes les plantes qui ne sont pas des containers (par exemple les arbres)

**furn** les objets de décoration (lit, table...)

**in** tous les éléments intérieurs mur, escalier couloir, eux aussi répartis en fonction de leur style architectural

**terrain** des éléments supplémentaires pour les extérieurs, des ponts, des rochers

**VFX** les effets visuels comme les sorts. Ceux-ci ne sont d'aucun usage.

Tous les autres éléments ne rentrent pas dans ces catégories (indicateur du nord NorthMarker ou le volcan). Si vous n'avez d'idée précise de ce que vous voulez construire ou si vous cherchez un objet spécifique, déplacer simplement les objets vers la fenêtre de rendu. Elle doit être active sur la cellule que vous êtes en train de construire. Si ce n'est pas le cas vous pouvez la rendre active en double cliquant sur la ligne de votre cellule dans la fenêtre 'Cell View'.

N'oubliez pas alors de toujours effacer ce que vous venez d'ajouter. Sauf si c'est vraiment voulu de votre part essayez de conserver une cohérence dans le style de votre intérieur. Ne faites, par exemple, pas cohabiter un style Daedric avec le style impérial.

Lors de la construction de la structure de base, il faut faire particulièrement attention à ne pas laisser d'espace ou de trous entre les éléments de la structure. Cela peut prendre un peu de temps mais vous n'aurez pas alors à

reprendre totalement l'aménagement intérieur si vous réalisez plus tard avoir fait une erreur dans la structure de base.

Cela ne présente pas d'intérêt particulier pour la création d'un plug normal, mais tous les éléments static ont des paramètres spécifiques de réglages. On y accède en double cliquant sur la référence du static dans la fenêtre de Rendu.

La fenêtre est classique.

**ID** est l'identité de l'objet, son type.

**Add Art File** : permet de spécifier le fichier qui contient la représentation visuelle de l'objet, le fichier nif – NetImmerseFile

**Save** ferme la fenêtre en prenant en compte les modifications apportées

**Cancel** ferme la fenêtre sans prendre en compte les modifications apportées.

Il peut arriver que l'objet que vous insérez dans une cellule soit trop grand ou trop petit par rapport au reste de votre univers. Il est alors possible de jouer sur l'échelle (3D Scale) du modèle, une valeur inférieure à 1 fera un objet plus petit alors qu'une valeur supérieure fera un objet plus grand.

Après avoir définis la structure de la cellule que vous venez de créer, c'est-à-dire les murs, plafonds cadre de porte etc... il faut la peupler.

### 3.1.2 Les créatures

Les créatures sont l'ensemble des animaux, ennemis, ... qui sont susceptibles d'apparaître dans le jeu. Par opposition avec les éléments static, elles sont animées.

Ce sont les créatures qui constituent la vie et les embûches du monde que vous créez.

Pour peupler un donjon, on procède de la même façon que pour le construire. C'est-à-dire que l'on procède par drag'n drop des créatures depuis l'onglet 'Creatures' de la fenêtre des objets, vers la fenêtre de rendu.

### 3.1.2.1 Les créatures simples

Toutes les créatures de “base” sont disponibles dans "Creature". On les ajoute à l'univers par drag'n drop. En double-cliquant sur une créature on fait apparaître la fenêtre des propriétés pour cette créature.

**Creature**

ID

Name

Script  ...

Type  Level

Essential ☐ Corpses Persist ☐ Respawn ☐

Weapon & Shield ☐

Movement  
☐ Flies ☒ Walks ☐ Swims  
☐ Biped ☐ None

Encumbrance ???/250

Attributes

Str	<input type="text" value="50"/>	Spd	<input type="text" value="9"/>	Health	<input type="text" value="200"/>
Int	<input type="text" value="50"/>	End	<input type="text" value="50"/>	Spell Pts	<input type="text" value="195"/>
Wil	<input type="text" value="90"/>	Per	<input type="text" value="50"/>	Fatigue	<input type="text" value="600"/>
Agi	<input type="text" value="50"/>	Luc	<input type="text" value="50"/>	Soul	<input type="text" value="150"/>

Skills

Combat	<input type="text" value="70"/>	Attack 1	<input type="text" value="15"/>	<input type="text" value="45"/>
Magic	<input type="text" value="90"/>	Attack 2	<input type="text" value="15"/>	<input type="text" value="45"/>
Stealth	<input type="text" value="20"/>	Attack 3	<input type="text" value="15"/>	<input type="text" value="45"/>

Blood Texture:

Sound Gen Creature:

Save Cancel

Dialogue Animation AI

☐ Blocked

Items Spells

Count	Object ID	Type
1	random_vo...	Lev...

**ID** l'identité de la créature. C'est à travers cet identifiant que l'on pourra faire référence à la créature dans les scripts. Chaque Objet a une identité unique, qui est limitée à 24 caractères.

**Name** Il s'agit du nom sous lequel la créature sera connue par le joueur dans l'univers. C'est ce qui apparaît à coté de la créature lorsque l'on déplace le viseur dessus. Ce nom est limité à 32 caractères.

**Script** Permet d'associer un script à la créature, soit un script qui existe déjà soit un script spécial que vous écrirez alors.

**Type** On choisit si la créature est une "Creature", un "Daedric", un "humanoid" ou 'Undead'. Cela affecte notamment les immunités au sort, aux maladies, etc...

**Level** Le niveau de la créature, par opposition avec le niveau du joueur, le niveau d'une créature n'affecte en rien ses autres caractéristiques.

**Essential** En cochant cette case, la créature restera dans le donjon, même morte. Cela peut être utilisé pour une créatures qui transporte un objet important pour le déroulement du jeu.

**Corpses Persist** (Le corps Reste) Vous permettez ou non au joueur de faire disparaître le corps du mort.

**Respawn** (Ressuscite) En cochant cette option, la créature réapparaîtra régulièrement. Le joueur pourra la tuer, partir puis revenir plus tard, il se trouvera toujours face à une créature du même type. Un fantôme dans les tombes, par exemple, réapparaît régulièrement.

**Attributes** Pour modifier les attributs de la créature. La valeur varie de 1 à 100 (comme pour le joueur)

**Health** détermine le nombre de PV de la créature

**Spell Points** détermine la quantité de magicka dont dispose la créature pour lancer des sorts.

**Fatigue** détermine le niveau de fatigue de la créature.

**Soul** Détermine la valeur de l'âme de la créature pour les enchantements. Plus cette valeur est importante, plus la charge de l'âme est importante.

**Skills** Par opposition avec les PNJ, les créatures ne disposent pas de toutes les compétences. Elles sont déterminées par trois sortes de spécialisation.

**Attack 1/2/3** Ce sont les dommages minimum/maximum que la créature peut infliger en défonçant/taille/estoc avec son arme ou l'équivalent (griffe, points)

**Blood Texture** la couleur du sang de la créature, le petit nuage qui apparaît quand on touche les créatures.

**Sound Gen Creature** Pour déterminer le son produit par la créature.

**Weapon&Shield** En cochant cette option, la créature peut utiliser des armes et des boucliers plutôt que ses griffes ou ses dents...

**Movement** Pour choisir si la créature peut/doit voler, marcher ou nager. En sélectionnant "Biped" la créature utilisera le squelette et l'animation des humanoïdes.

**Items** C'est la liste de ce que la créature transporte. Comme tous les autres 'personnages', elles ont un encombrement limité. Pour ajouter quelque chose à la liste des objets transportés, on le fait par drag'n drop depuis la liste des objets dans la fenêtre des objets. Si vous sélectionnez un objet dans la liste et que vous recliquez sur la colonne "Count", vous pouvez alors définir la quantité transportée par la créature. En choisissant une valeur négative vous spécifiez qu'il y en a une infinité. Le seul intérêt de cela est pour les marchands, c'est pourquoi nous l'expliquerons en détail dans la section PNJ. Pour retirer un objet de la liste, il suffit de le sélectionner et d'appuyer sur la touche [SUPPR]

**Spells** Pour spécifier les capacités spéciales, les pouvoirs et sorts dont la créature dispose. Le fonctionnement en est le même que pour les objets transportés.

**Save/Cancel** Vous pouvez soit sauvegarder soit annuler les modifications que vous venez de faire.

**Dialogue** Cela fait apparaître la fenêtre des dialogues possibles avec la créature. Puisque les créatures ont la désagréable habitude de vous attaquer avant de vous parler, nous approfondirons cela dans la section concernant les PNJ.

**Animation** fait apparaître la fenêtre "Animation"

**AI** (Artificial intelligence) pour spécifier les paramètres AI de la créature. Cela est plus intéressant pour les PNJ.

Le bouton en haut à gauche permet d'associer un nouveau modèle à la créature. Ces modèles doivent être faits sous 3DS Max puis importer grâce à un outil de Bethesda qui les convertit en fichier nif.

Nous verrons cela plus tard.

### **3.1.2.2 Les créatures de niveau**

L'intérêt des "Créatures de niveau" se présente lorsque vous voulez créer un donjon qui n'a pas de niveau prédéfini; mais qui s'adapte au niveau du joueur. Ainsi la difficulté du donjon augmentera en même temps que le niveau du joueur. Pour cela, chaque « Créature de niveau » dispose d'une liste de créatures que le joueur rencontrera en fonction de son niveau. La fenêtre de configuration des "créatures de niveau" est légèrement différente de celle des créatures classiques.

Vous remarquerez également que le drag'n drop d'une créature de niveau dans la fenêtre de rendu fait apparaître un singe.

**Leveled Creature**

ID

☒ Calculate from all levels <= PC's level

Chance None

☐ Blocked

PC ...	Creature Name
1	scrib
1	Rat
1	kwama forager
2	scrib diseased
3	rat diseased
3	netch_bull
3	netch_bull
9	netch_betty
9	netch_betty

**Reference Data**

**Position**

X

Y

Z

**Rotation**

X

Y

Z

**3D Scale**

☐ Extra Data

Health Left

Soul

**Assignable Data**

Owner

Global Variable/Rank

☐ Teleport

Load Cell

☐ Locked

Level

Key

Trap

☐ Reference Blocked

**ID** L'identité de la créature, elle est limitée à 32 caractères, et doit, comme pour les créatures normales être unique.

**Calculate from all levels <= PC's level** En cochant cette case le joueur rencontrera l'une des créatures dont le niveau est inférieur ou égal au sien. Si l'option est désélectionnée, alors le joueur ne rencontrera que des créatures dont le niveau est égal au sien.

**Chance None** La valeur spécifiée dans ce qui suit indique le pourcentage pour que le joueur ne rencontre rien du tout.

La liste sur la droite contient les créatures que le joueur peut rencontrer, elles sont associées au niveau du joueur. Vous pouvez modifier la liste en ajoutant/retirant des créatures. Toujours par drag'n drop depuis la liste des créatures dans la fenêtre des objets. Ou bien, vous pouvez changer les niveaux du joueur de la même façon que pour modifier le nombre d'objets transportés. La plupart des informations présentes sous la barre 'Reference Data' ne sont pas intéressantes, elles se réfèrent à la position de la créatures dans le monde.

### 3.1.3 Les PNJ (NPC)

Les PNJ sont essentiels dans un jeu de rôle. Ceux sont eux qui permettent au joueur de marchander, de se tenir au courant des dernières rumeurs, d'avoir un avis, des informations. Ils nous offrent des quêtes ou bien au contraire sont des ennemis à abattre. Les PNJ ou NPCs (Non-Player-Characters) constituent la vie de l'univers dans lequel le joueur évolue. Tous le PNJ existant sont disponibles dans l'onglet "NPC" de la fenêtre des Objets. Pour assurer une cohérence dans le monde, chaque PNJ doit être unique. Si vous voulez ajouter un PNJ dans le jeu, il existe une méthode très simple qui garantira son unicité dans l'univers.

Placer un PNJ dans le monde, peu importe lequel, puisque vous allez ensuite modifier ses propriétés.

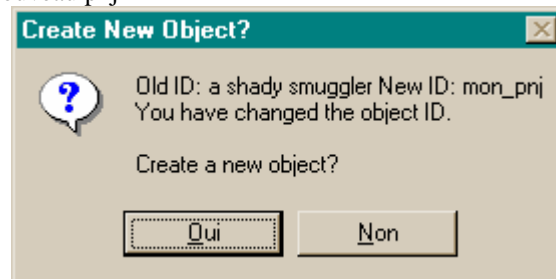
Ouvrez la fenêtre des propriétés en double cliquant dessus.

La première chose à faire ensuite est de modifier l'ID du PNJ. Elle doit être unique et est limitée à 24 caractères. Il est très important de commencer par là, dans le cas contraire vous modifieriez un PNJ existant dans le jeu. Ce

qui pourrait avoir des conséquences dramatiques. Pour vous éviter ce genre de désagréments, vous pouvez créer un plug qui ne contient qu'un PNJ, qui n'est jamais utilisé et faire en sorte que tous vos PNJ soient construits à partir de lui. Ainsi, si vous oubliez de modifier l'ID du PNJ que vous créez, les dégâts seraient limités.

Puis après avoir modifié l'ID du pnj, vous cliquez Save

Et acceptez la création d'un nouveau pnj



La fenêtre des propriétés vous offre de nombreuses options.

**NPC**

ID:  Name:  Script:  ...

Race:  Female: ☐ Class:  Level:  Faction & Rank:

Essential: ☐ Corpses Persist: ☐ Respawn: ☐ Add Animation File:

Attributes:

Str	45	Spd	54
Int	42	End	42
Wil	32	Per	64
Agi	33	Luc	40

Health:  Magicka:  Fatigue:  Disp:  Rep:

Blood Texture:  Auto Calculate Stats: ☒

Value	Skill
5	Marksman
5	Acrobatics
5	Security
5	Unarmored
5	Alchemy
5	Restoration
5	Mysticism

Encumbrance 11/225

Items | Spells

Count	Object ID	Type
1	common_p...	Clot...
1	common_s...	Clot...
1	common_s...	Clot...
1	common_b...	Clot...
1	steel dagger	We...

Dialogue Animation AI Save Cancel

☐ Blocked

**ID** Comme cela est expliqué plus haut, l'ID doit être unique et est limitée à 24 caractères.

**Name** C'est le nom du pnj tel qu'il apparaîtra dans le jeu. Pour assurer une cohérence dans le jeu, il est conseillé d'éviter de choisir un nom qui ne fasse pas 'local'. Evitez donc les noms ressemblant à « Patrick Dupond », il est évident que cela ne sonne pas vraiment Tamrielesque.

**Script** Pour choisir (ou créer) un script pour ce pnj.

**Race** La race du PNJ, vous pouvez utiliser toutes les races que vous trouvez dans le master ou celles qui sont disponibles par le biais d'autres plug'in. Dans ce cas n'oubliez pas le ou les auteurs originaux des modèles que vous créez.

**Class** Pour attribuer une profession au pnj. De même que pour les races vous pouvez choisir autant dans les classes existantes que dans celles qui sont ajoutées.

**Female** En cochant, cette case, le pnj sera une femme (ou son équivalent), sinon ce sera un homme.

**Level** Le niveau du pnj. Si l'option "Auto Calculate Stats" est cochée, l'éditeur fait les calculs des attributs et des compétences correspondantes au niveau que vous choisissez. Je n'ose vous recommander de laisser cette option cochée (cela vous fera gagner du temps).

**Faction & Rank** Pour choisir la faction et le rang dans cette faction pour le pnj.

**Essential** En cochant cette option, vous signalez que le pnj est indispensable au déroulement de l'intrigue. C'est-à-dire que si le joueur tue ce pnj, il verra apparaître une boîte de dialogue l'informant qu'il vient de tuer un individu essentiel et qui l'invite à reprendre une partie sauvegardée plus tôt.

**Corpses Persist** En cochant cette option, vous empêchez que le joueur retire le corps après l'avoir tué.

**Respawn** En cochant cette option, le pnj réapparaîtra au même endroit au retour du joueur si il a tué le pnj. Il va de soit que de cocher cette option risque d'introduire dans le jeu une dimension peu réaliste. Je vous recommande donc de réfléchir à deux fois avant de la cocher.

**Attributes** Tous les pnj ont les mêmes caractéristiques que le joueur. Elles sont évaluées automatiquement si l'option "Auto Calculate Stats" est cochée.

**Health** Le nombre de PV du pnj. On peut également les calculer automatiquement. Si vous désirez créer un pnj mort, il faut désélectionner l'option "Auto Calculate Stats", puis mettre la valeur de Health à 0.

**Magicka** La réserve d'énergie magique dont le pnj dispose pour lancer des sorts.

**Fatigue** La quantité d'énergie dont le pnj dispose (l'équivalent de la barre verte du PJ). Elle est consommée quand le pnj court se bat ou est frappé.

**Disp** La disposition du pnj par rapport au joueur. (La barre bleue dans la fenêtre de dialogue du jeu). Vous ne pouvez mettre ici qu'une valeur indicative. En effet la disposition d'un pnj par rapport au joueur est également fonction de différents facteurs tels que la personnalité, la réputation, la faction etc. du joueur

**Rep** La réputation du pnj. Cela affecte le taux de réussite des tentatives de persuasion (pour augmenter la disposition du pnj à votre égard). Plus la réputation est élevée, plus il est difficile de le « persuader ».

**Bloody Texture** Pour définir la texture du sang du pnj.

**Auto Calculate Stats** Si cette option est cochée, l'éditeur calcul automatiquement les caractéristiques du pnj. Sa santé, sa magicka, sa fatigue, sa réputation, ses compétences les sorts et les services qu'il peut rendre en fonction de son niveau, sa faction, sa race et sa classe. De sorte que si vous désirez modifier une des compétences, il faut désactiver cette option.

**Skill list** La liste des compétences (27) et le niveau atteint par le pnj pour chacune. Contrairement au reste vous pouvez les modifier manuellement même si l'option "Auto Calculate Stats" est sélectionnée. Pour modifier le niveau dans une des compétences, double cliquez sur la compétence et choisir une valeur (entière) entre 1 et 100.

**Dialogue** On approfondit cela plus loin.

**Animation** Pour modifier les animations des pnj. Aussi longtemps que vous ferez des plug simples pour MW, vous n'aurez rien à changer ici.

**AI** À voir plus loin

**Encumbrance** Cela montre la charge du pnj. Comme pour le PJ, cela limite le nombre d'objet que le pnj peut transporter.

**Items** C'est la liste de tout ce que le pnj transporte. Cela fonctionne comme la liste des items pour les créatures ou pour les conteneurs. Vous pouvez ajouter des objets des catégories : Alchemy, Apparatus, Armor, Book, Clothing, Ingredient, Leveled Item, Lockpick, Misc Item, Probe, Repair Item, Weapon. Le PNJ porte les habits ou armures et armes qui sont le plus utiles pour lui. (Par exemple un sorcier préfère ne pas porter d'armure mais une robe, un maillot et un pantalon). Si la quantité d'item transportée par le pnj est fixée à une valeur négative, cela signifie que le pnj dispose de l'item en quantité illimitée. C'est une caractéristique très utile pour les commerçants. Le fonctionnement en est le suivant: quand le joueur achète l'item au commerçant, l'item disparaît de l'inventaire du marchand. Mais la prochaine fois que le joueur viendra parler au marchand, il pourra de nouveau acheter cet objet. En revanche si le joueur tue le marchand, il ne trouvera qu'une fois les objets sur le corps. Ils ne réapparaîtront pas.

**Spells** La liste des sorts que le pnj peut utiliser. La liste peut être déterminer automatiquement, aussi si vous désirez ajouter un sort il faut désactiver l'option "Auto Calculate Stats". On peut toujours retirer les sorts.

**Save** Sauvegarde les modifications faites et retourne dans l'éditeur.

**Cancel** Retourne dans l'éditeur sans conserver une seule modification.

En haut à droite de la fenêtre de dialogue, il y a deux listes. Ces deux listes permettent de définir l'apparence de votre pnj. Sa tête et sa coupe de cheveux, exactement comme vous l'avez fait lors de la création de votre personnage au début de votre partie. Le bouton "Add Animation File" vous permet de charger de nouveaux modèles pour les pnj (ils doivent se présenter sous forme de fichier NetImmerse file nif).

### 3.1.3.1 Les dialogues

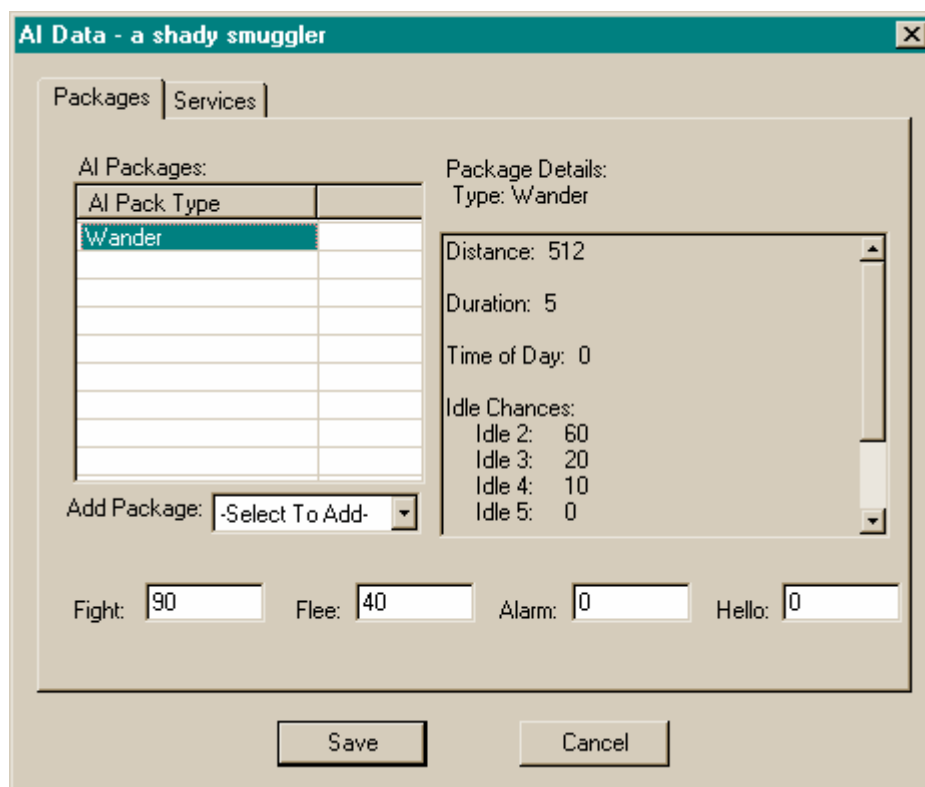
On se reportera utilement à la section dialogue qui entre dans les détails de cet élément essentiel du TESCS.

### 3.1.3.2 AI (Artificial Intelligence)- Intelligence Artificielle

La fenêtre de l'I.A. est constituée de deux onglets, l'un contient la description du comportement global du pnj ,et l'autre les services qu'il peut rendre.

#### 3.1.3.2.1 Packages – Le comportement général

Le premier contient les informations relatives au comportement général du pnj. Il est possible de paramétrer les différentes familles de l'IA qui affectent le comportement du personnage. Si vous choisissez plusieurs types de comportements, ceux ci seront effectués dans l'ordre dans lequel vous les avez figés, et ceci en boucle. C'est à dire qu'une fois que toute la liste a été parcourue, on la reprend au début.



Nous donnons maintenant une description des fonctionnalités couvertes dans les familles de l'IA ( les packages).

**Activate** Avec ce package, le pnj utilise certains objets de son environnement. La seule information utile pour ce package est l'ID de l'objet à utiliser. Le pnj réalisera toujours l'action standard associée avec l'objet (il s'équippa avec les armures ou les habits....)

**Escort** Avec ce package, le pnk escortera (suivra) un autre pnj jusqu'à un endroit spécifique. Cela implique que le pnj suivra et protégera un autre pnj. Pour fonctionner ce package a besoin de connaître le pnj à escorter (vous devez connaître son ID). On notera que la personne escortée peut aussi bien être un autre pnj que le joueur lui-même. Il faut également définir le temps durant lequel le pnj devra escorter l'autre et l'endroit où l'escorte

s'arrêtera, après avoir choisis une cellule dans la liste, il est encore possible de sélectionner un point dans la cellule jusqu'où l'escorte doit aller, ceci est déterminé par des coordonnées spatiales en x-y-z. Le bouton "View Location" permet de vérifier la validité des coordonnées que vous avez saisies. Attention : lorsque vous demandez à quelqu'un d'escorter une autre personne, le temps que cela prendra n'est pas une donnée...

**Follow** hormis la défense du pnj suivi, je n'ai pas trouvé de différence avec le package Escort.

**Travel** Ceci impose au pnj de se rendre à un certain endroit dans la cellule. Cet endroit est là encore spécifié en coordonnées x-y-z et peut être visualisée grâce au bouton "View Location".

**Wander** un pnj qui erre se déplace aléatoirement le long de la grille de déplacement de la cellule dans laquelle il est. Distance donne le rayon d'un cercle imaginaire limitant l'excursion du pnj. Duration, indique la durée de son errance. Time of Day indique l'heure du jour à laquelle il se déplacera. Les Idle Chances indique un taux de probabilité pour que le personnage s'arrête et fasse une autre sorte de geste. Voici les activités possible : determine the chances of the NPC stopping and performing some kind of gesture. There are the following possible gestures:

A droite de l'écran, il y a un panneau qui donne des détails sur le package AI actuellement sélectionné. Si vous désirez que le pnj se contente d'être à un certain endroit sans bouger (cela peut être le cas pour un marchand), il n'y a qu'à effacer tous les packages.

Sous le menu déroulant des packages AI il y a 4 champs de text qui déterminent le comportement du pnj lorsque vous êtes dans son entourage.

**Fight** Si cette valeur est importante le pnj attaquera à vue le joueur. Dans le cas contraire, cette valeur détermine la probabilité pour que le pnj attaque le joueur si il commet un crime.

**Flee** C'est la probabilité pour que le pnj fuit le joueur.

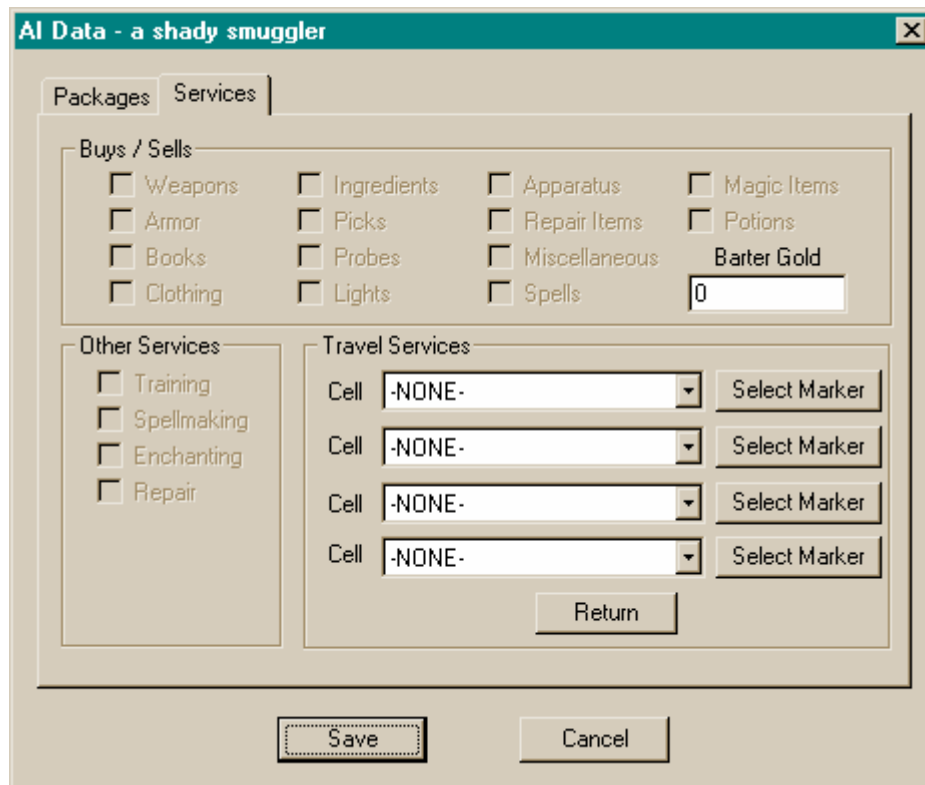
**Alarm** donne des informations sur la réaction du pnj par rapport aux crimes que le joueur commet devant lui (que ce soit en les signalant ou en attaquant le pj).

**Hello** indique la distance à partir de laquelle le pnj saluera le joueur.

### 3.1.3.2.2 Les services

Le second panneau contient la description des services que le pnj offre au joueur. (Commerce, enchantement...) Cela a déjà été précisé au dessus, pour pouvoir modifier les données de ce panneau il faut avoir désactivé l'option "Auto Calculate Stats". Ce panneau se compose de trois parties. La première, "Buys / Sells", permet de fixer les biens que le commerçant peut vendre ou acheter. Les catégories sont les mêmes que celles que l'on trouve dans la fenêtre des Objets. Dans le cas des sorts, la liste des sorts disponibles va dépendre de la profession du pnj. Les autres services ("Other Services") sont bien connus dans le jeu.

Les services d'apprentissage (Training) permettent au pnj d'offrir des formations au joueur, les compétences qu'il peut enseigner dépendent de sa profession. La réparation pour les forgerons qui peuvent réparer les armes et les armures. La fabrication de sorts (Spellmaking) et d'enchantement (Enchanting) ont le même fonctionnement que dans le jeu. Les propositions de voyage permettent au joueur via une certaine somme d'argent de se rendre dans une autre cellule. Il est possible d'offrir de transporter le joueur vers 4 destinations différentes. L'endroit où le joueur est téléporté est marqué par des 'Travel marker', le bouton "Select Marker" amène la caméra vers le repère associé.



Return permet de retourner au pnj.

### 3.1.4 Les trésors simples

Les éléments pour les trésors peuvent être trouvés dans les onglets

Alchemy, Apparatus, Armor, Book, Clothing, Ingredient, Lockpick, Misc Item, Probe, Repair Item, Weapon.

Les objets à niveau "Leveled Item" sont des trésors mais ne peuvent être placés que dans les conteneurs ou les inventaires de personnages. Les trésors et conteneurs sont placés et déplacés comme les autres objets dans l'univers. Nous allons parler des propriétés des objets de ces catégories.

**ID** Chaque objet a un identifiant unique, limité à 32 caractéristiques.

**Name** C'est le nom sous lequel l'objet sera connu dans l'univers. Il est limité à 32 caractères.

**Script** le script pour cet objet

**Weight** indique le poids de l'objet en livre ( ben oui c'est un jeu américain ;) ), c'est la mesure utilisée dans le jeu.

**Value** la valeur financière moyenne de l'objet lors de transaction avec les commerçants.

**3D Scale** détermine la taille de l'objet

**Extra Data/Owner** pour spécifier le propriétaire de l'objet, cela peut être un pnj ou une faction. Si vous choisissez un pnj, vous pouvez adjoindre un variable à cet objet qui indique si le joueur peut (true) ou non (false) prendre l'objet. Si vous avez sélectionné une faction vous pouvez spécifier un rang minimum que le joueur doit atteindre pour prendre l'objet. Si le joueur essaye de prendre l'objet sans y être autorisé, alors il commet un crime.

**Extra Data/Apply To Selection** Cette option est utile lorsque vous voulez attribuer les mêmes données à un groupe d'objet (par exemple le propriétaire). Sélectionner un groupe de référence en utilisant la touche [CTRL]. Puis double-cliquez sur une des références et changer les données et appuyer sur "Apply To Selection", toutes les références sélectionnées auront les mêmes Extra Data.

**Position** indique la position de l'objet dans le monde.

**Save** permet de prendre en compte les modifications apportées.

**Cancel** retourne à l'éditeur sans prendre en compte les modifications effectuées.

Deux boutons sous "Script" indiquent:

1. le fichier graphique contenant la représentation 3D de l'objet (nif)
2. le fichier représentant l'icone 2D qui apparaît dans l'inventaire pour l'objet.(tga).


### 3.1.5 Alchemy – Alchimie

**Alchemy**

ID:

Name:

Script:



Weight:

☐ References Persist

☐ Blocked

Effects	Cost	Duration	Magnitude	Total Cost
Detect Animal	0.75	15	10	22.58
		0	0	
		0	0	
		0	0	
		0	0	
		0	0	
		0	0	
		0	0	

☐ Auto Calculate Value

Potion Value:

**Reference Data**

Position: X:   Y:   Z:

Rotation: X:   Y:   Z:

3D Scale:

☐ Extra Data

Health Left:

Soul:

Assignable Data

Owner:

Global Variable/Rank:

☐ Teleport

Load Cell:

☐ Locked

Level:

Key:

Trap:

☐ Reference Blocked

**Effects** les effets de la potion

**Duration** le nombre de secondes durant lesquelles les effets de la potion se font sentir.


**Magnitude** c'est la puissance de la potion. Par exemple le pourcentage d'effet caméléon.

**Potion Value** la valeur financière moyenne de ce type de potion.

**Auto Calculate Value** Si cette option est sélectionnée la valeur de la potion est la somme des "Total Costs" affichés dans la partie gauche de la fenêtre.

### 3.1.6 Apparatus

**Apparatus**

ID:   

Name:

Type:

Script:  ...

Weight:  Value:

Quality:

☐ References Persist

☐ Blocked

**Reference Data**

Position: X:   Y:   Z:

Rotation: X:   Y:   Z:

3D Scale:

☐ Extra Data

Health Left:

Soul:

**Assignable Data**

Owner:

Global Variable/Rank:

☐ Teleport

Load Cell:

☐ Locked

Level:

Key:

Trap:


☐ Reference Blocked

**Type** Indique la nature de l'appareil. Est ce un alembic, un calcineur un mortier... et ainsi les effect sur les potions préparées.

**Quality** indique la qualité de l'appareil. Plus l'appareil est de bonne qualité plus le joueur a de chancede réussir sa potion et plus elle sera puissante.

### 3.1.7 Armor - Armures

**Armor**

ID:   

Name:

Type:

Script:  ...

Weight:  Medium:

Health:  Value:

Enchantment:

Enchanting:

☐ References Persist

☐ Blocked

**Biped Object**

Head:

**Male Armor**

**Female Armor**

**AR "AR"** signifie "Armor Rating", c'est à dire Classe d'Armure en quelque sorte. Plus le nombre est grand plus l'armure offre une protection efficace au joueur.

**Health** L'état de l'objet. Son état de deterioration. Lorsqu'il atteint 0, l'objet ne peut plus servir et doit être réparé avant de pouvoir s'en resservir.

**Enchantment** la quantité d'enchantement que l'objet peut supporter. Chaque enchantement, pouvoir magique, ajouté à l'arme consomme un certain nombre de points d'enchantement. Si on essaye d'ajouter sur l'objet plus d'enchantement qu'il ne peut en supporter, l'enchantement ne pourra pas être achevé.

**Enchanting** pour spécifier si l'objet supporte déjà un enchantement.

**Biped Object** indique la partie du corps concernée par l'objet.

**Male / Femal Armor** indique quelle partie modèle doit être utilisé en fonction du sexe du joueur.

**Extra Data/Health Left** indique le taux d'usure de l'objet trouvé par le joueur. Dans certains cas cela peut se transformer en "Count", dans le cas par exemple des flèches ou des étoiles de jets... Il est alors possible de spécifier le nombre d'éléments représenté par cette item.

### 3.1.8 Book

**Teaches** on précise la compétence qui est augmentée par la lecture de l'ouvrage.

**Scroll** si c'est coché, l'ouvrage est affiché comme un parchemin.

**Enchantment** le nombre de points d'enchantement don't l'objet dispose.

**Enchanting** pour choisir un enchantement déjà présent sur l'objet.

**Book Text** le texte de l'ouvrage. C'est le texte que le joueur pourra (ou non) lire. On utilise une version simplifiée de l'HTML et est limité à 64 000 caractères. Les seuls tag HTML autorisés sont les suivants: [BR];

[P]; [FONT]; [DIV]. Pour avoir une idée du rendu du code que vous écrivez, il suffit d'ouvrir la fenêtre des propriétés d'un livre.

### 3.1.9 Clothing – les habits

**Enchantment** C'est la capacité d'enchantement d'un objet. Le nombre de points d'enchantement qu'on peut lui attribuer.

**Enchanting** pour ajouter un enchantement sur l'objet.

**Biped Object** indique la partie du corps couverte par cet habit.

**Male / Female Clothing** reference de l'objet qui couvre la dite partie du corps.

### 3.1.10 Ingredient

**Effects** on peut spécifier jusqu'à quatre effets pour l'objet. On peut être amené à donner plus d'information quand on choisit "Restore Skill" ou "Restore Attribute".

### 3.1.11 Lockpick

**Uses** le nombre maximum d'utilisation pour cet objet.

**Quality** la qualité du pic de crochétage. Plus cette valeur est élevée plus la probabilité d'une utilisation réussie est élevée.

**Extra Data/Uses Left** le nombre d'utilisations restantes.

### 3.1.12 Misc Item

Ces objets n'ont pas d'autres options que celles expliquées ci-dessus, excepté pour les Soul Gems.

**Extra Data/Soul** pour indiquer le type de l'âme enfermée dans la gemme.

### 3.1.13 *Probe – les sondes*

On retrouve les memes informations que pour les pics de crochetages.

**Uses** le nombre maximum d'utilisation pour cet objet.


**Quality** la qualité du pic de crochetage. Plus cette valeur est élevée plus la probabilité d'une utilisation réussie est élevée.

**Extra Data/Uses Left** le nombre d'utilisations restantes.

### 3.1.14 *Repair Item – les outils de réparation*

Ce sont les outils que l'on utilise pour réparer les armures ou les armes.

**Repair Item**

ID:   

Name:

Script:  ...

Weight:  Value:  ☐ References Persist

Uses:  Quality:  ☐ Blocked

**Reference Data**

Position: X:  1.00 Y:  1.00 Z:  1.00

Rotation: X:  1.00 Y:  1.00 Z:  1.00

3D Scale:  ☐ Extra Data Uses Left:  Soul:

Assignable Data: Owner:  Global Variable/Rank:

☐ Teleport Load Cell:

☐ Locked Level:  Key:  Trap:

☐ Reference Blocked


**Uses** le nombre d'utilisation possible

**Quality** la qualité de l'outil. Plus elle est élevée, plus la réparation a des chances de réussir.

**Extra Data/Uses Left** le nombre d'utilisation qui reste.

### 3.1.15 Weapon – Les armes

**Weapon**

ID:   

Name:

Type:

Script:  ...

Weight:  Value:

Health:  Speed:

Enchantment:  Reach:

Enchanting:

☒ Ignores Normal Weapon Resistance ☐ References Persist ☐ Blocked

	Minimum	Maximum
Chop	<input type="text" value="1"/>	<input type="text" value="44"/>
Slash	<input type="text" value="1"/>	<input type="text" value="33"/>
Thrust	<input type="text" value="1"/>	<input type="text" value="4"/>

**Reference Data**

Position: X:   Y:   Z:

Rotation: X:   Y:   Z:

3D Scale:

☐ Extra Data: Health Left:  Soul:

Assignable Data: Owner:  Global Variable/Rank:

☐ Teleport: Load Cell:

☐ Locked: Level:  Key:  Trap:

☐ Reference Blocked

**Type** on y spécifie le type de l'arme de cette façon

[la famille d'arme(hâche, épée, ...)] [le nombre de main nécessaire pour se servir de l'arme] [la distance d'utilisation (corps à corps, jet, ...)]

Ces trois informations ne sont pas toujours écrites.

**Health** La condition maximum d'utilisation de l'objet. Dans MW les armes et les armures s'usent lorsque l'on s'en sert, cette valeur est un image du nombre d'utilisation possible de l'arme. Lorsque la valeur atteint 0, l'arme doit être réparée avant de pouvoir être utilisée.

**Speed** Indique la vitesse de l'arme, le nombre de coups que l'on peut donner en une minute par exemple.

**Enchantment** le nombre de point d'enchantement de l'objet.

**Enchanting** si l'arme a déjà un enchantement on précise sa nature à cet endroit

**Reach** indique la distance maximum à la quelle la cible doit se trouver si on veut la toucher.

**IgnoresNormalWeaponResistance** cette option permet de signaler que l'arme infligera des dommages même si la cible est immune aux attaques avec des armes das le matériau( fer, argent...)

**Damage** détermine les valeurs maximum et minimum des dégats que l'on peut infliger avec cette arme. En coup assené (chop) de taille (slash) ou d'estoc (thrust).

**Extra Data/Health Left** indique l'état d'usure actuel de l'arme

### 3.1.16 Les conteneurs

Les conteneurs sont tous les objet qui peuvent contenir d'autre objet. Par exemple, les caisses, les coffres, les corps, certaines plantes. Dans certains conteneurs, vous pouvez y prendre et y ajouter des objets, dans l'autre vous ne pouvez que prendre des objets.

Count	Object ID	Type
1	misc_lw_bowl_chapel	Misc Item
20	random_ingredient	Leveled...

**ID** l'ID unique de cet objet. Limitée à 24 caractères

**Name** le nom de l'objet qui apparaîtra dans le jeu pour le joueur, limité à 32 caractères.

**Script** on y ajoute un script, par exemple un créature apparaît lorsque le joueur essaie d'ouvrir le coffre

**Organic Container** si cette option est cochée, il est impossible d'ajouter des objets dans le conteneur pendant le jeu. C'est le cas des plantes par exemple

**Weight** Le poids d'un conteneur a une autre fonction que le poids d'un objet. Il s'agit de sa capacité de contenance.

**Extra Data/Owner** on peut préciser que le conteneur appartient à une personne, auquel cas vous commettez un crime en prenant des objets qui s'y trouvent.

**Extra Data/Apply To Selection** Cette option permet d'appliquer les mêmes paramètres à une sélection de référence que vous avez construite en utilisant la touche [CTRL].

**3D Scale** l'échelle de l'objet, pour augmenter ou diminuer sa dimension.

**Locked** verrouillage: si le conteneur est verrouillé ou piégé, on peut préciser ici le niveau de verrouillage, la clé qui peut ouvrir le verrou et la nature du piège. Le niveau de verrouillage est défini par une valeur comprise entre 1 et 100. 'Key' vous permet de faire référence à la clé qui permet d'ouvrir ce verrou, et si le joueur n'en dispose pas, alors il pourra essayer de cocher la serrure et de désarmer les pièges en utilisant les objets adaptés ou un sort. "Trap" vous permet de choisir les effets du piège.

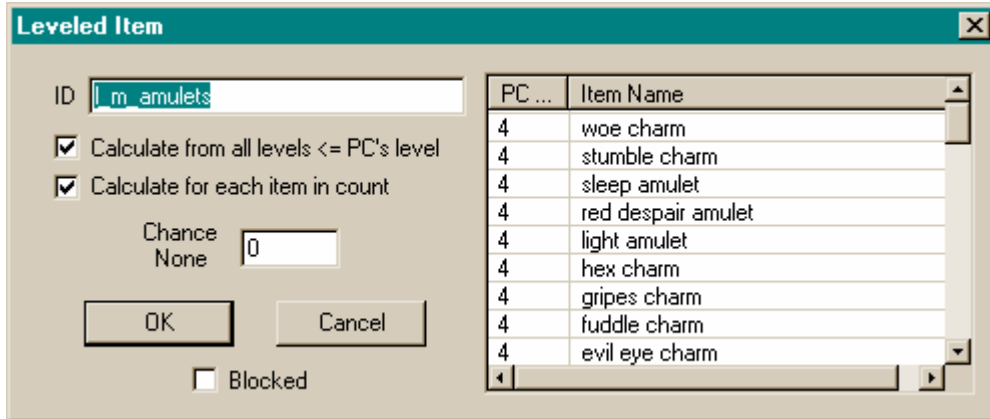
**Save** enregistre les modifications et retourne dans l'éditeur

**Cancel** ne sauvegarde aucune modification en retournant dans l'éditeur.

Le bouton au dessus de "References Persist" vous permet de modifier l'apparence du conteneur en lui attribuant un nouveau fichier nif. La liste en haut à droite décrit le contenu de l'objet (quantité, ID des objets, et type). Vous ne pouvez modifier que la quantité des items que vous avez ajouté. Pour ajouter des objet personnalisés il faut la créer dans l'éditeur puis la sauvegarder et l'ajouter dans le conteneur par drag'n drop dans la liste. Les conteneurs peuvent aussi contenir des Objets de niveau (Leveled Item).

### 3.1.17 Les Objets de niveau - Leveled Items

Ce type d'objet ne peut être ajouté que dans les conteneurs ou les inventaires. Elle fonctionne de la même façon que les créatures de niveau.



**ID** C'est l'identifiant unique pour cet objet. Il est limité à 32 caractères.

**Calculate from all levels <= PC's level** Cette option a le même comportement que sa soeur pour les créatures de niveau. Si elle est sélectionnée, tous les objets qui sont associés à un niveau inférieur ou égale à celui du joueur apparaitront. Sinon, seules les items dont le niveau est égal à celui du joueur apparaissent.

**Calculate for each item in count** Cette option est intéressante si le conteneur contient plus d'un objet. Si elle est cochée, le moteur du jeu générera un nouvel objet pour chaque reference de l'objet de niveau dans le conteneur. Si elle n'est pas cochée, alors le moteur détermine une fois la nature de l'objet et le multiplie par le nombre de reference dans le conteneur. (Par exemple, si vous disposer d'un conteneur qui contient "l\_m\_amulets" trois fois et que cette option n'est pas cochée, et que le résultat donne l'objet "stumble charm", il y aura trois "stumble charm" identiques dans le conteneur. En revanche, si l'option est cochée, vous pourrez trouver une "woe charm", une "sleep amulet" et une "light amulet", ou alors deux "hex charm" et une "red despair amulet". LA raison de ce résultat est que les objets sont déterminés aléatoirement).

**Chance** la probabilité de ne rien trouver du tout

**OK** sauvegarde les modifications et retourne dans l'éditeur

**Cancel** retourne dans l'éditeur sans prendre en compte les modifications.

La liste à droite contient les objets parmi lesquels le choix aléatoire sera fait. Elle associe un niveau du joueur et une ID d'objet. Il est possible d'ajouter/retirer les objets comme dans toutes les autres liste. Par drag'n drop depuis la fenêtre des objets.

### 3.1.18 Les portes - Doors

Il existe deux types de portes, les portes "Static" que l'on trouve dans le panneau des "static", elles ne peuvent être ouverte ou fermée. Vous pouvez vous en servir pour limiter les explorations du joueur, si vous ne voulez pas qu'il voit une pièce que vous n'avez pas finie, par exemple.

Toutes les autres portes sont disponibles dans l'onglet 'Doors', et il y en a au moins une par style architectural.

**ID** l'ID (identifiant) unique de la porte. Limité à 32 caractères.

**Name** le nom sous lequel le joueur verra la porte dans le jeu.

**Script** pour associer un script à la porte

**Animation** Ici vous pouvez voir l'animation de la porte. Cette option est souvent désactivée.

**3D Scale** Vous permet de modifier la taille de la porte en augmentant ou diminuant la valeur du nombre dans le champ de texte à droite.

**Extra Data/Owner** Vous pouvez spécifier le propriétaire de la porte, et si le joueur essaye de forcer la porte alors il commet un crime.

**Extra Data/Apply To Selection** Cette option est utile lorsque vous voulez attribuer les memes données à un groupe d'objet (par exemple le propriétaire). Sélectionner un groupe de référence en utilisant la touche [CTRL]. Puis double-cliquez sur une des références et changer les données et appuyer sur "Apply To Selection", toutes les references sélectionnées auront les mêmes Extra Data.

**Teleport** Cette option est utilisée pour connecter des cellules entre elles ou vers l'extérieur. Si l'option est validée, vous pouvez choisir la cellule dans laquelle vous serez téléporté. Le bouton au dessus permet de basculer entre la porte et son marqueur de positionnement de sorte que si le 'Door Marker' n'est pas placé correctement vous pouvez ajuster sa position. Si le joueur utilise la porte, il se retrouvera téléporté à l'endroit du repère réagissant dans la direction que celui-ci indique.

**Locked** de même que pour les conteneurs, on peut signaler que la porte est verrouillée, et si nécessaire piégée. Le 'Lock Level' (niveau de Verrouillage) est défini par un nombre compris entre 1 et 100. "Key" permet de donner une clé qui ouvrira le verrou et désamorcera le piège. "Trap" permet de piéger la porte, en choisissant la nature du piège.

**Save** Sauvegarde les modifications faites et retourne dans l'éditeur.

**Cancel** retourne dans l'éditeur sans prendre en compte les modifications apportées.

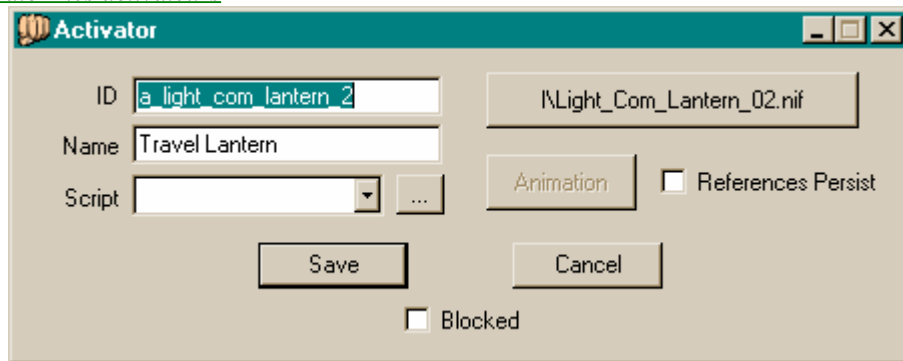
En haut à droite de la fenêtre, il y a trois boutons. Le premier permet d'assigner un modèle visuel à la porte (doit un fichier nif). Les deux autres boutons vous permettent de modifier les sons que le joueur entendra en ouvrant la porte (ou en la fermant).

Il y a deux façons d'utiliser les portes.

dans la première utilisation possible, vous pouvez donner une structure plus logique à votre donjon En compartimentant le donjon, vous éviterez que le joueur ne se fasse attaquer par toutes les créatures en même temps entre autre. Vous pourrez également produire une architecture plus cohérente.

Le deuxième mode d'utilisation est le plus élémentaire : relier l'intérieur de votre donjon à l'extérieur. Dans ce cas, il faut utiliser l'option "Teleport" des propriétés de la porte.

### 3.1.19 Activator- les activateurs



**ID** une ID unique pour les objets, limitée à 24 caractères

**Name** le nom qui apparaît dans le jeu

**Script** pour adjoindre un script à l'objet (dans ce cas il faut utiliser la variable OnActivate).

**Add Art File** pour associer un fichier .nif qui represent L'activateur dans le jeu.

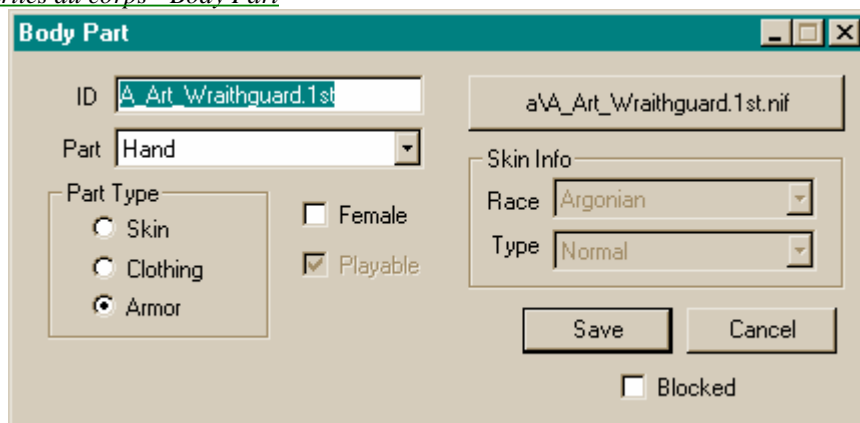
**Extra Data/Owner** le propriétaire de l'objet. Si le PJ le prend, il commet un crime.

**Extra Data/Apply To Selection** Cette option est utile lorsque vous voulez attribuer les memes données à un groupe d'objet (par exemple le propriétaire). Sélectionner un groupe de référence en utilisant la touché [CTRL]. Puis double-cliquez sur une des références et changer les données et appuyer sur "Apply To Selection", toutes les references sélectionnées auront les mêmes Extra Data.

**OK** sauvegarde les modifications et ferme la fenêtre.

**Cancel** annule toutes les modifications et retourne dans l'éditeur.

### 3.1.20 Les parties du corps - Body Part



**ID** un identifiant unique, qui est utilisé pour les propriétés d'autres objets (par exemple, les armures).

**Part** détermine la partie du corps représentés. Il y a: le bassin (Ankle), Chest, les épaules(Clavicle), les pieds (Foot), l'avant bras (Forearm), le museau (Groin), les cheveux (Hair), les mains (Hand), la tête (Head), les genoux (Knee), Neck, Tail, le bras (Upper Arm), la cuisse (Upper Leg), Wrist.

**Part Type** détermine dans quel cas on utilise tel ou tel partie du corps lorsque le personnage porte une armure, des habits ou rien.

**Female** cette partie sera utilisée pour les femmes.

**Add Art File** pour définir le modèle 3D qui représente cette partie du corps (.nif).

**Skin Info** permet de sélectionner la race qui utilisera cette partie de corps et si il s'agit d'un vampire ou non.

**OK** enregistre les modifications et ferme la fenêtre.

**Cancel** annule toutes les modifications et retourne dans l'éditeur

### 3.1.21 Les enchantements - Enchanting

Cette catégorie contient la liste de tous les enchantement prédéfinies que l'on peut ajouter sur des objets dans le jeu.

Effects	Cost	Range	Area	Duration	Magnitude	Total Cost
Summon Ancestral Ghos	7.00	Self	0	100	1	35.00
Fortify Skill	1.00	Self	0	100	10	50.00
			0	0	0	
			0	0	0	
			0	0	0	
			0	0	0	
			0	0	0	
			0	0	0	

**ID** l'ID unique qui est reconnue dans les autres fenêtres de propriétés.

**Cast Type** pour choisir entre utilisé une fois (Cast Once), lancé lorsqu'on le demande (Cast When Used) – le joueur doit être en mode sort pour l'utiliser – ou lancé lorsque l'objet est utilisé (Cast When Strikes) – valable uniquement pour les armes- ou effet constant (Constant Effect).

**Effects** on peut associer jusqu'à 8 effets magiques qui sont choisis dans la liste Spellmaking. Il peut arriver que l'on doive donner plus d'information sur le sort (Absorb Fatigue XXXXpoints), vous pouvez aussi choisir la distance d'effet du sort (contact - Touch, à distance - Target, le pj - Self). La zone (Area) d'effet de l'enchantement, la durée du sort (en second) et sa puissance its Magnitude (par exemple le nombre de points de dégâts infligés par une boule de feu)

**Charge Amount** le nmbre de charge de l'objet

**Enchantment Cost** c'est le nombre de point d'enchantement que l'objet doit avoir pour être enchanté.

**Auto Calculate** permet d'évaluer automatique les deux valeurs ci dessus.

**OK** enregistre les modifications et ferme la fenêtre.

**Cancel** annule toutes les modifications et retourne dans l'éditeur

### 3.1.22 Les lumières – Light

**ID** l'ID unique de l'objet

**Radius** la rayon de la sphère affectée par cette lumière.

**Script** pour ajouter un script associé à cette éclairage. Par exemple, la lumière s'éteint lorsque le pj entre dans une certaine zone.

**Add Art File** pour attribuer le modèle 3D qui sera utilisé dans le jeu..

**Color** permet de définir la couleur de la lumière. On peut choisir une couleur, ou bien entrer son code RGB.

**Flicker Effect** permet de définir la vitesse de la pulsation ou du clignotement de la lumière.

**Sound ID** c'est le son qui répété en boucle lorsque la lumière est allumée.

**Fire** cette option doit être validée si la source lumineuse est une flamme.

**Negative** cette option sur une source lumineuse aura l'effet inverse. C'est-à-dire que la zone dans le rayon d'action de la source lumineuse ne sera pas plus claire mais au contraire plus sombre.

**Dynamic** permet de définir si cette source lumineuse est affectée par d'autres sources lumineuses (par exemple, les zones d'ombre doivent pouvoir être éclairées par la présence d'une torche).

**Can carry** si cette option est validée, le joueur peut transporter la source lumineuse. Cela active également les champs (Name, Weight, Time et Value).

**Off by default** si cette option est cochée, il n'y aura pas de lumière dans le monde du jeu.

**Name** le nom qui apparaîtra dans le jeu.

**Weight** le poids en livre.

**Time** détermine la durée pendant laquelle la source lumineuse éclairera l'environnement.(en secondes)

**Value** la valeur financière de la lampe.

**OK** enregistre les modifications et ferme la fenêtre.

**Cancel** annule toutes les modifications et retourne dans l'éditeur

### 3.1.23 Les objets statiques – Static

**ID** l'ID unique de l'objet, qui sert à identifier l'objet dans le jeu (par exemple dans les scripts)

**Add Art File** pour définir le modèle 3D utilisé dans le jeu (doit un fichier nif).

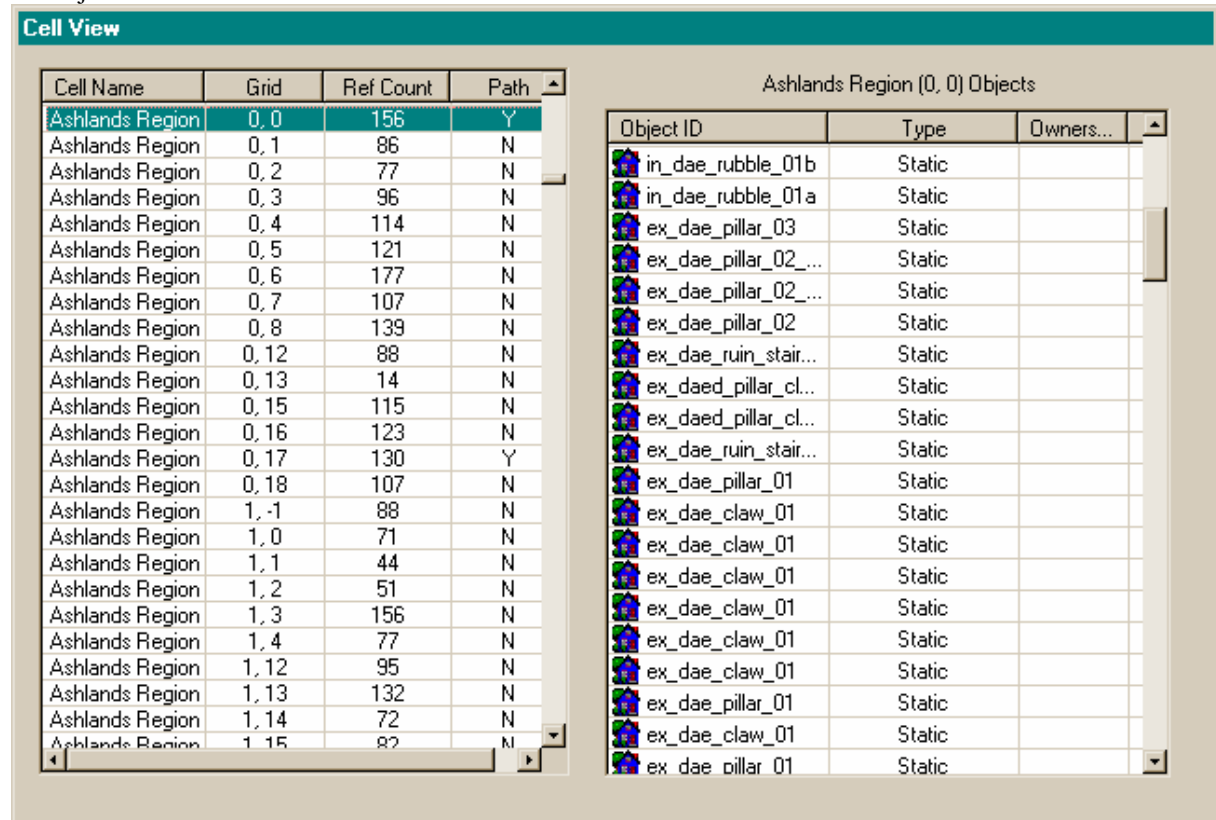
**OK** enregistre les modifications et ferme la fenêtre.

**Cancel** annule toutes les modifications et retourne dans l'éditeur

### 3.2 Le fenêtre Cell View

Dans cette fenêtre, on trouve la liste de toutes les cellules (intérieures et extérieures) qui sont définies dans le fichier maître et dans les plugs chargées.

On y trouve la liste des cellules identifiées par leur nom (rappelez vous, la grosse erreur de Bethesda) et la liste des objets contenus dans cette cellule.



La zone de gauche contient la liste des cellules.

**Cell Name** est le nom des cellules, il doit être unique et est le même que celui que le joueur verra dans le jeu.

**Grid** correspond à la position de la cellule par rapport à une référence qui se situe dans les Ashlands au sud-est de Caldera. (on se reportera à la visualisation des régions dans la fenêtre des Regions)

**Ref Count** contient le compte du nombre de références dans la cellule.

**Path** indique l'absence ou la présence de grille de déplacement dans la cellule.

La zone de droite contient la liste des références présentes dans la cellule mise en surbrillance dans la zone de droite.

**Object ID** est le type de la référence

**Type** est la catégorie dans laquelle ce type d'objet est enregistrée

**Owners** correspond au propriétaire éventuel de la référence.

Un double clic sur l'ID d'une cellule ordonne à l'éditeur de charger la cellule indiquée et son contenu dans la fenêtre de rendu.

Les sélections qui sont faites dans la colonne de droite permettent de choisir la référence qui est mise en surbrillance dans la fenêtre de rendu. Un double-clic permet de centrer la vue de la fenêtre de Rendu sur la référence sélectionnée.

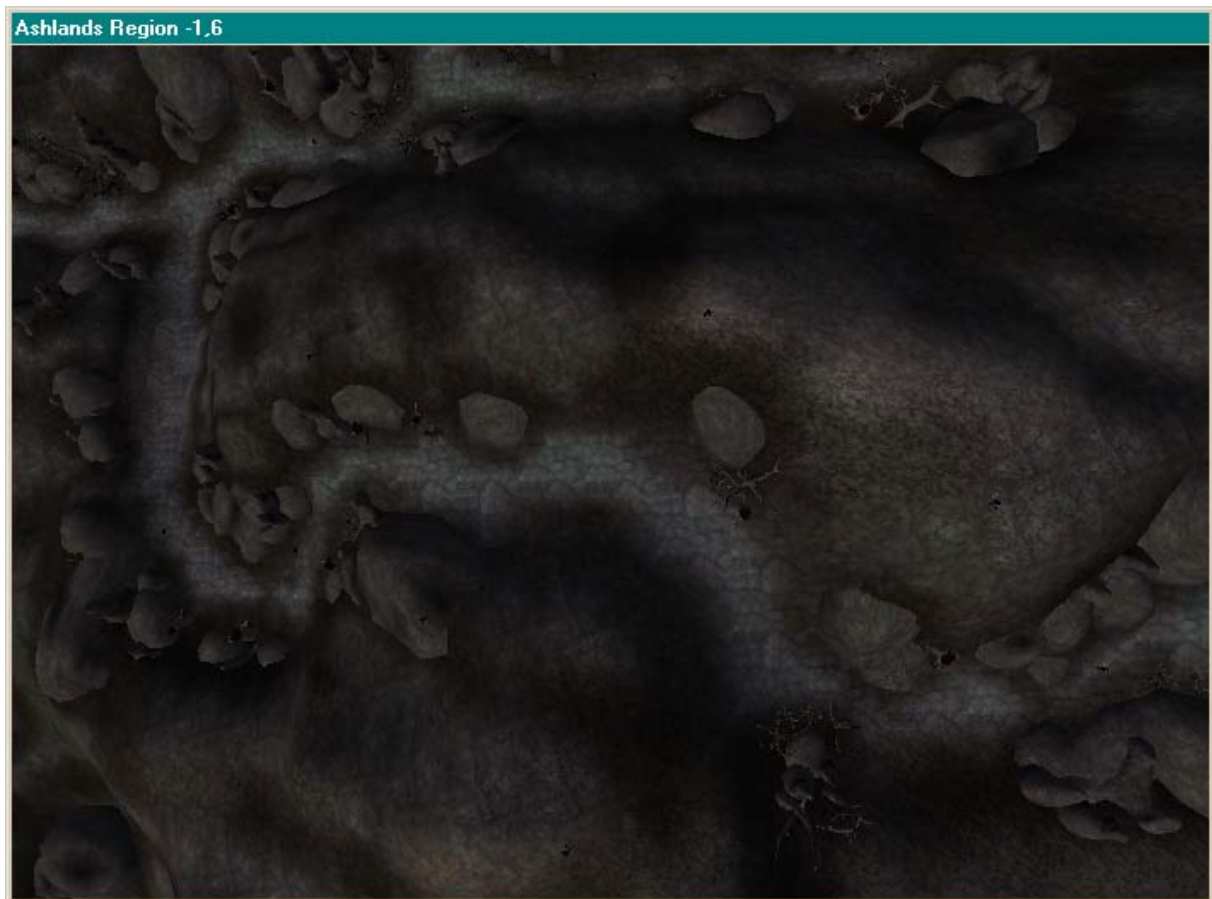
### 3.3 La fenêtre de rendu

Dans cette fenêtre, vous construisez les cellules que vous désirez ajouter à l'univers de MW (intérieure ou extérieure). Vous pouvez également y visionner ou modifier les cellules existantes.

En fait, c'est le cœur visuel du TESCS, c'est là que vous construisez l'univers visuel par ajouts des éléments que vous trouvez dans la fenêtre des objets.

Vous pouvez construire ces cellules et les peupler, les remplir par un simple drag'n drop des objets depuis la fenêtre des objets vers l'endroit où vous voulez placer la référence ainsi créée.

Ce type de développement que nous présentons est inspiré du monde logiciel. Il s'agit du RAD (Rapid Application Design), qui permet à une personne non experte dans le développement logiciel de construire des applications de haut niveau par l'assemblage de pièces élémentaires (un peu comme un lego).



#### 3.3.1 Mouvement dans la fenêtre de rendu

Nous allons maintenant expliquer comment déplacer les points de vue de la caméra dans la fenêtre de rendu. Intéressons nous d'abord au déplacement de la caméra.

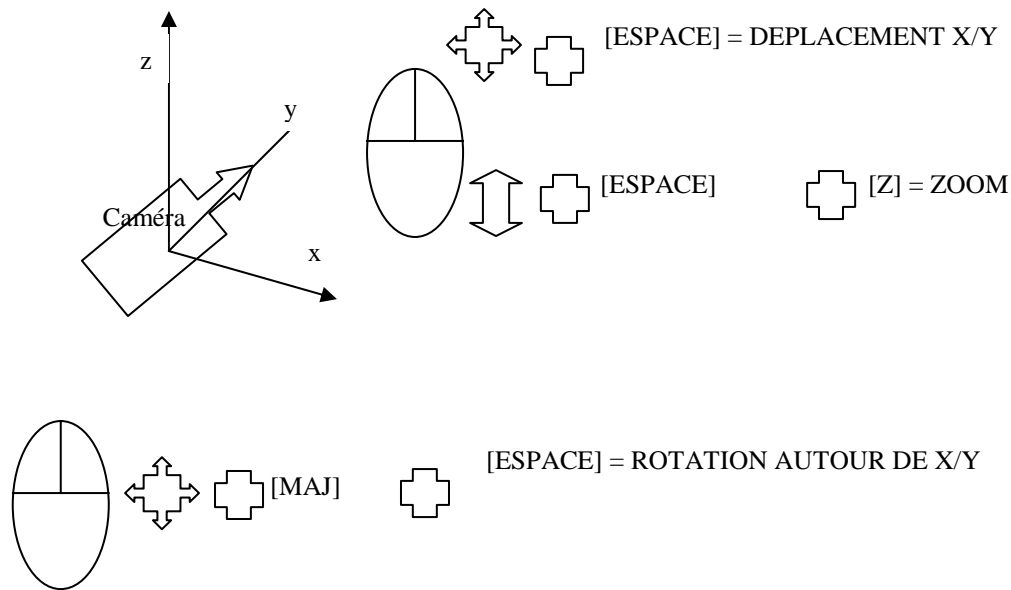
##### 3.3.1.1 Déplacer la caméra

La manière la plus simple de comprendre le mouvement de la caméra est de s'imaginer un système de coordonnées qui se déplacent par rapport à la position courante de la caméra. Pour déplacer la caméra le long des axes X et Y, il faut appuyer sur la barres ESPACE et simultanément déplacer la souris. Pour se déplacer le long de l'axe z (altitude), soit on utilise la roulette de la souris, soit on appuie sur [V] et on déplace la souris de haut en bas. Pour faire tourner la caméra, appuie sur [MAJ] et on bouge la souris. Quand on fait tourner la caméra alors qu'un objet est sélectionné, on tourne autour de l'objet.

Pour faire un zoom sur un objet, on le sélectionne et on fait la même chose que pour se déplacer le long de l'axe z.

Ces mouvements sont valables dans deux cas :

- si aucune référence n'est sélectionnée, les mouvements se font dans le repère du monde
- si une référence est sélectionnée, les mouvements de la caméra se font dans le repère de la référence sélectionnée. (cette dernière solution est souvent plus confortable que la précédente). Attention le repère de l'objet sélectionné à les mêmes directions que celui du monde. Seule l'origine change.



### 3.3.2 Déplacer et redimensionner des objets

Pour déplacer un objet le long des axes x et y, il faut le sélectionner, puis on place le curseur sur l'objet, on clic gauche et tout en restant enfoncé on le déplace (c'est une opération classique de Drag'n Drop) Si vous voulez modifier son altitude, on appuie et on reste enfoncé sur la touche [Z] et on déplace la souris. Pour faire tourner l'objet autour de l'axe Y, on le sélectionne, on clic dessus et on fait un clic droit tout en déplaçant la souris. Pour le faire tourner autour des axes X et Z on appuie les touches [X] et [Z] du clavier tout en restant enfoncé. Il est également possible de redimensionner un objet dans la fenêtre de rendu, pour cela il faut sélectionner l'objet, puis la combinaison de la touche [S] et d'un mouvement vertical de la souris permet de faire grossir (vers le haut) ou rétrécir (vers le bas) un objet ou une créature. Cela modifie l'échelle (3D Scale) de l'objet sélectionné.

### 3.3.3 Raccourcis clavier

[SOURIS HAUT / BAS / DROITE / Gauche] fait déplacer (sur de grandes distances) la zone observée dans la fenêtre de rendu. (on se déplace dans le cellule).

[CTRL]+[Z] Défaire

[CTRL]+[Y] Refaire

[CTRL]+[X] Couper

[CTRL]+[C] Copier

[CTRL]+[V] Coller

[CTRL]+[MAJ]+[V] recopie de l'objet cloné à la même place que l'objet original

[CTRL]+[D] Duplique

[CTRL]+[F] Trouver

[MAJ]+[C] – cache tous les objets ajoutés dans une cellule d'extérieur (Pratique pour voir les altitudes).

[F2] Trouver le précédent

[F3] Trouver le suivant

[F4] Active-Désactive le mode collision (Cela montre les endroits où le joueur se heurte à des objets)

[A] Active – Désactive l'éclairage (on ignore ou pas les sources de lumières dans la fenêtre de rendu, tout devient visible)

[C] Vue Isométrique (de l'objet actuellement sélectionné dans la fenêtre Cell View)

[F] Tombe (Fall) pour amener l'objet courant au contact du sol, l'objet actuellement sélectionné « tombe » jusqu'à heurter un autre objet. Cette fonction est très pratique pour placer des static ou des objets de décoration non affecté d'un effet de lévitation permanente.

[H] Active-Désactive le mode d'édition de paysage

[L] Active – Désactive le mode de visualisation des zones d'éclairages des différentes sources lumineuses

[T] Top View – pour avoir la vue du dessus, toujours par rapport à l'objet actuellement sélectionné.

[W] Active – Désactive le mode Fil de Fer  
[DEL] on retire l'objet actuellement sélectionné

En utilisant les touches du pavé directionnel, il est également possible de se déplacer suivant les axes X et Y du monde. Faites tout de même attention, les orientations peuvent être un peu déconcertantes (la flèche droite déplace vers le gauche)

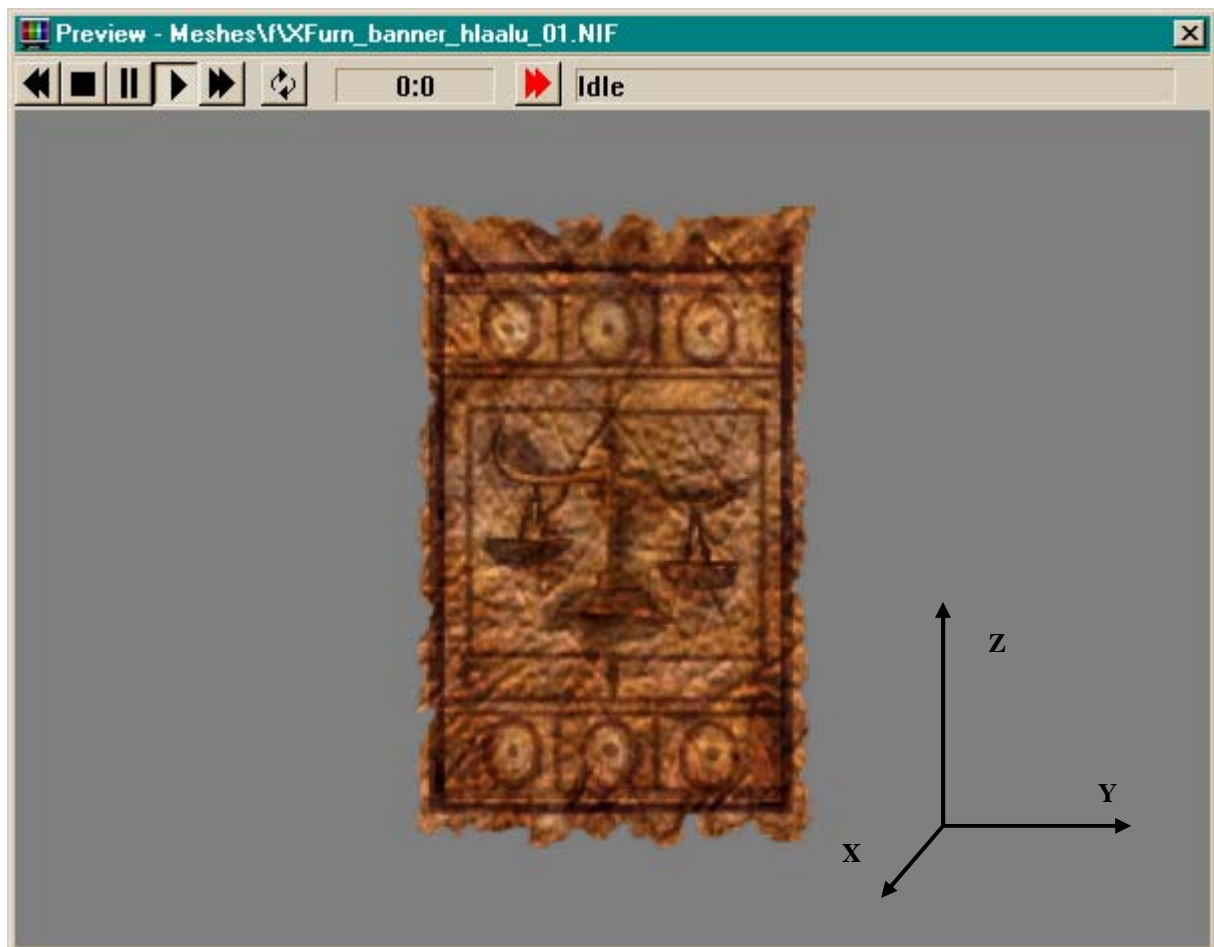
[PAGE UP] et [PAGE DOWN] permettent de faire tourner le monde autour de l'axe Z du repère choisi.

### 3.4 La fenêtre Preview

Pour voir la fenêtre preview il faut utiliser le menu View/Preview, l'activation de cette fenêtre fait immédiatement disparaître la fenêtre de rendu. Dans cette fenêtre vous pouvez visualiser l'allure des différents objets qui ont une représentation dans l'univers du jeu.

Vous pouvez également y prévisualiser les différentes animations des objets qui en ont une.

Pour visualiser un objet dans cette fenêtre, il suffit de l'avoir ouverte et de sélectionner un objet dans la fenêtre des Objets.



Dans cette fenêtre comme dans la fenêtre de Rendu, vous pouvez bouger les objets et modifier leur position par rapport à la caméra.

Les déplacements dans le plan de vue se font avec les touches

[A] – translate l'objet vers la droite (sur l'image)

[D] – translate l'objet vers gauche

[W] – translate l'objet vers le bas

[S] – translate l'objet vers le haut

[PAGE UP] – permet de faire un zoom sur l'objet

[PAGE DOWN] – permet de faire un zoom arrière sur l'objet.

Mais il est également possible de le faire tourner autour de ces différents axes. Pour l'explication des mouvements je suppose que lorsque vous chargez un objet dans la fenêtre il est orienté avec son axe Z vertical, l'axe X sort de l'écran et pointe vers vous et l'axe Y est horizontal dans la fenêtre.

En utilisant les commandes de rotation de l'objet, il faut bien garder à l'esprit que vous faites également tourner ce repère.

Les touches [INSER] et [DEBUT] font tourner l'objet autour de son axe X.

Les flèches [HAUT] et [BAS] font tourner l'objet autour de son axe Y.

Les flèches [DROITE] et [GAUCHE] font tourner l'objet autour de son axe Z.

DE plus lorsqu'il y a une animation associée à l'objet que vous visualisez (ou plusieurs) vous verrez ces animations dans la fenêtre. Vous remarquerez que dans certains cas ce n'est pas l'objet qui est animé mais une association de cubes. C'est dans ce cas l'animation des créatures de type humanoïdes.

Si il y a a plusieurs animations associées à l'objet vous pouvez les voir les unes après les autres grace au bouton rouge :




La zone de texte à droite du bouton indique le nom du groupe d'animation actuellement jouée, tandis que la zone de gauche indique le numéro de la frame actuelle et celui de la fin pour cette animation.

### 3.5 La fenêtre des régions

C'est dans cette fenêtre que vous pouvez définir de nouvelles régions, modifier les existantes et surtout spécifier l'appartenance d'une cellule extérieure à une région donnée. Ce qui permettra entre autre de lui attribuer un climat particulier.

On accède à cette fenêtre par le menu "World/Regions...".

Regions																																																																							
Region ID:	Bitter Coast Region	Name: Bitter Coast Region																																																																					
New	Delete	Change ID																																																																					
Sleep Creature:	ex_bittercoast_sleep																																																																						
Weather Chances																																																																							
Clear:	10	Rain:																																																																					
Cloudy:	60	Thunder:																																																																					
Foggy:	10	Ash:																																																																					
Overcast:	0	Blight:																																																																					
Region Paint																																																																							
Region Map Color																																																																							
R	34																																																																						
G	39																																																																						
B	255																																																																						
Select Color																																																																							
<table border="1"> <thead> <tr> <th>Chance</th> <th>Priority</th> <th>Sound ID</th> </tr> </thead> <tbody> <tr><td>4</td><td>1</td><td>wind calm1</td></tr> <tr><td>4</td><td>2</td><td>wind calm2</td></tr> <tr><td>4</td><td>3</td><td>wind calm3</td></tr> <tr><td>4</td><td>4</td><td>wind calm4</td></tr> <tr><td>4</td><td>5</td><td>wind calm5</td></tr> <tr><td>5</td><td>6</td><td>wind trees1</td></tr> <tr><td>5</td><td>7</td><td>wind trees2</td></tr> <tr><td>5</td><td>8</td><td>wind trees3</td></tr> <tr><td>5</td><td>9</td><td>wind trees4</td></tr> <tr><td>5</td><td>10</td><td>wind trees5</td></tr> <tr><td>5</td><td>11</td><td>wind trees6</td></tr> <tr><td>5</td><td>12</td><td>wind trees7</td></tr> <tr><td> </td><td> </td><td> </td></tr> <tr><td> </td><td> </td><td> </td></tr> <tr><td> </td><td> </td><td> </td></tr> <tr><td> </td><td> </td><td> </td></tr> <tr><td> </td><td> </td><td> </td></tr> <tr><td> </td><td> </td><td> </td></tr> <tr><td> </td><td> </td><td> </td></tr> <tr><td> </td><td> </td><td> </td></tr> <tr><td> </td><td> </td><td> </td></tr> <tr><td> </td><td> </td><td> </td></tr> </tbody> </table>			Chance	Priority	Sound ID	4	1	wind calm1	4	2	wind calm2	4	3	wind calm3	4	4	wind calm4	4	5	wind calm5	5	6	wind trees1	5	7	wind trees2	5	8	wind trees3	5	9	wind trees4	5	10	wind trees5	5	11	wind trees6	5	12	wind trees7																														
Chance	Priority	Sound ID																																																																					
4	1	wind calm1																																																																					
4	2	wind calm2																																																																					
4	3	wind calm3																																																																					
4	4	wind calm4																																																																					
4	5	wind calm5																																																																					
5	6	wind trees1																																																																					
5	7	wind trees2																																																																					
5	8	wind trees3																																																																					
5	9	wind trees4																																																																					
5	10	wind trees5																																																																					
5	11	wind trees6																																																																					
5	12	wind trees7																																																																					
OK																																																																							

**New** vous permet de créer une nouvelle région. La principale caractéristique d'une cellule qui dépend de la région à laquelle elle appartient est son climat. Toutes les cellules dans une région ont le même climat.

**Delete** efface la région actuellement sélectionnée.

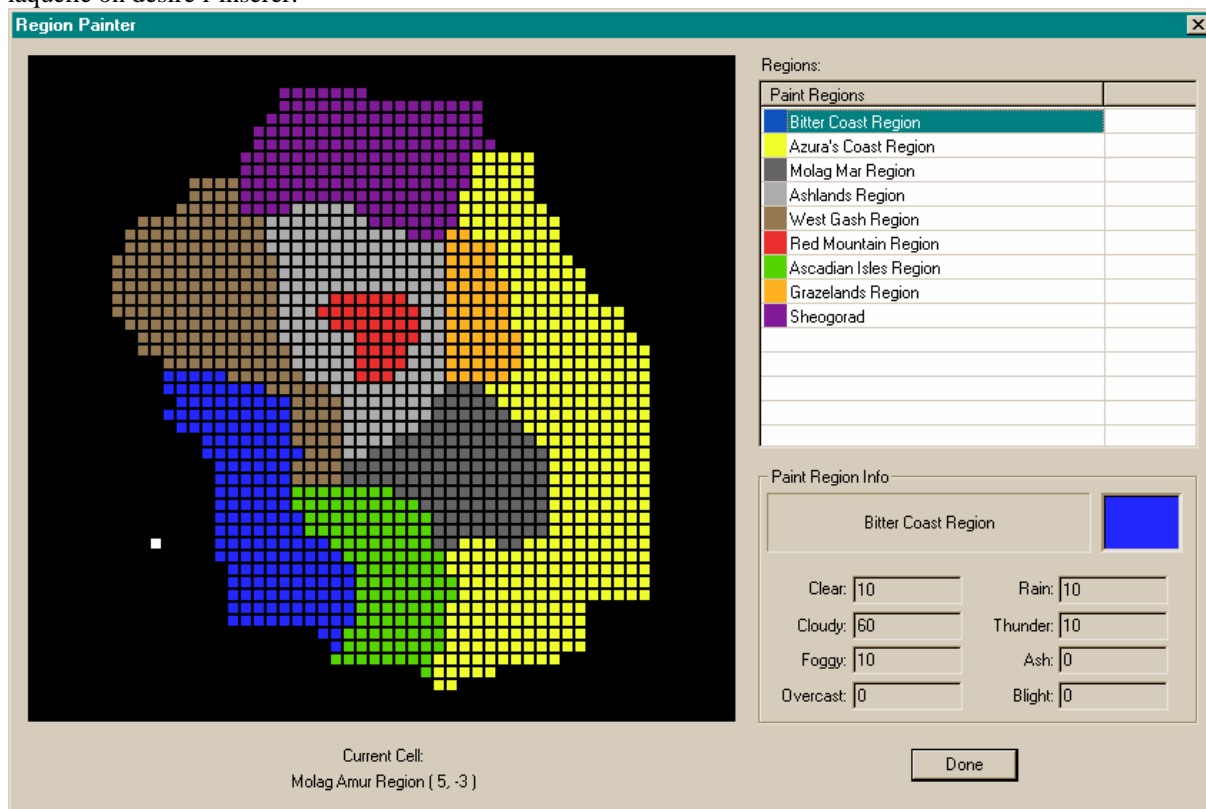
**Change ID** vous permet de modifier l'ID d'une region existante.

**Name** C'est le nom de la region. A ne pas confondre avec les noms des cellules. Le nom de la région n'apparaît jamais dans le jeu.

**Sleep Creature** Ici, vous spécifier une créature de niveau qui attaque le joueur lorsqu'il dort.

**Weather Chances** vous définissez les poids de chacun des 8 climats élémentaires. Il s'agit de pourcentage, la somme des valeurs que vous entrez doit être égale à 100.

**Region Paint** fait apparaître la fenêtre "Region Paint". Cela vous montre toutes les cellules existantes dans le monde, colorée d'une certaine couleur. Cette couleur est la couleur associée à la région dont elles dépendent. On peut choisir une région dans la liste de droite et attribuer à la nouvelle cellule la couleur de la région dans laquelle on désire l'insérer.



**Region Map Color** indique la région associée à la couleur sélectionnée dans la fenêtre "Region Paint". Comme d'habitude, vous pouvez choisir une couleur dans la liste (en cliquant à droite dans la liste) ou bien créer une nouvelle couleur en donnant son code RGB.

**OK** Enregistre vos modifications et ferme la fenêtre.

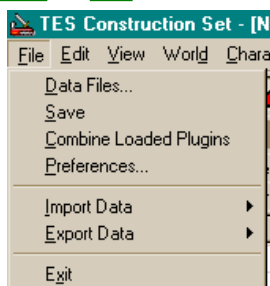
La liste à droite contient l'énumération de tous les sons d'ambiance qui peuvent être entendus dans la région. Vous pouvez ajouter des sons en sélectionnant "Gameplay/ Sounds..." puis en ajoutant les sons dans la liste par Drag'n drop. Puis, vous pouvez définir les probabilités que le joueur entende tel ou tel son. Comme pour les climats, cette valeur est un pourcentage et la somme des valeurs attribuées ne doit pas être supérieure à 100, mais peut être inférieure à 100. La priorité des sons ne peut pas être modifiée. Mais vous pouvez enlever des sons de la liste en les sélectionnant puis en appuyant sur la touche [DEL].

### 3.5.1 *Modifier les propriétés d'une cellule*

Pour modifier les propriétés d'une cellule, sélectionner "World/Exterior Cell...". La fenêtre "Exterior Cell" apparaît. Dans cette fenêtre vous pouvez choisir une des cellules extérieures. Vous pouvez alors modifier sa région d'appartenance et l'illégalité du repos dans cette cellule.

## 3.6 *Les menus de l'IHM*

### 3.6.1 *File*



**Data Files...** Pour charger le master file et les plug

**Save** enregistre les modifications dans le plug actif.

**Preferences** pour régler vos préférences d'édition

**Import Data** importe les données depuis un fichier txt.

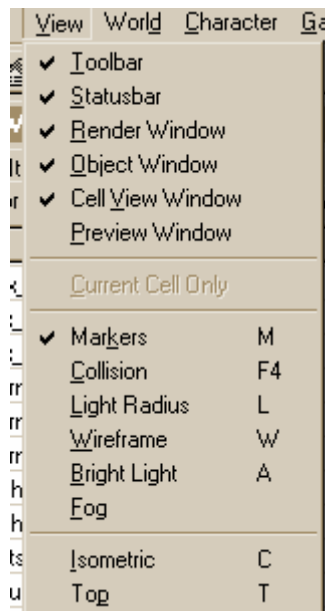
**Export Data** exporte les données vers un fichier txt

**Exit** ferme le TESCS

### Edit

Les fonctions d'édition standard de Windows, annule la dernière action, refait ce qui a été annulé, recherche d'un objet, remplacement....

### 3.6.2 View



**Toolbar** rend la barre d'outil (in)visible.

**Statusbar** rend la barre de status (in)visible (celle qui est en bas de l'éditeur).

**Render Window** rend la fenêtre 'Render Window' (in)visible. Si la fenêtre de rendu est active, la fenêtre de Prévisualisation ( Preview) est automatiquement fermée.

**Object Window** rend la fenêtre des Objets (in)visible.

**Cell View Window** rend la fenêtre des cellules (in)visible.

**Preview Window** rend la fenêtre de prévisualisation (in)visible ce qui permet de visualiser une animation. Ouvrir cette fenêtre ferme automatiquement la fenêtre de rendu.

**Markers** rend les 'Markers' (nord et porte) (in)visible.

**Collision** montre les endroits où le joueur entre en collision avec des éléments du décor (ce qui permet de vérifier si le joueur peut ou non passer à un endroit).

**Light Radius** montre les rayons d'éclairage de toutes les sources lumineuses.

**Wireframe** passe du mode fil de fer au mode texturé.

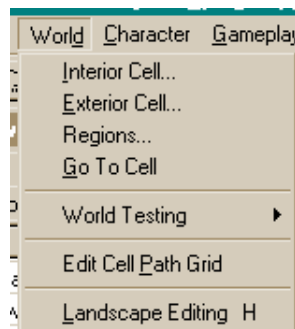
**Bright Light** désactive toutes les lumières. C'est une option très utile lorsque l'on place les objets static, les créatures ou les pnj. Il est recommandé de basculer entre ce mode et le mode normal lorsque vous placez les éclairages afin d'avoir une idée de l'effet des éclairages.

**Fog** active le brouillard. C'est très gourmand en temps de calcul. Je vous recommande donc de le désactiver.

**Isometric** si un objet est sélectionné, cela permet de passer à un mode de visualisation isométrique de l'objet.

**Top** pour avoir une vue aérienne de l'objet sélectionné

### 3.6.3 World



**Interior Cell...** pour la création des cellules d'intérieur (ou la modification) fait apparaître la fenêtre "Interior Cell".

**Exterior Cell...** pour la création de cellule d'extérieur. Fait apparaître la fenêtre "Exterior Cell".

**Regions** pour l'édition des régions. Fait apparaître la fenêtre des régions.

**Go To Cell** vous permet de vous rendre directement dans une cellule donnée.

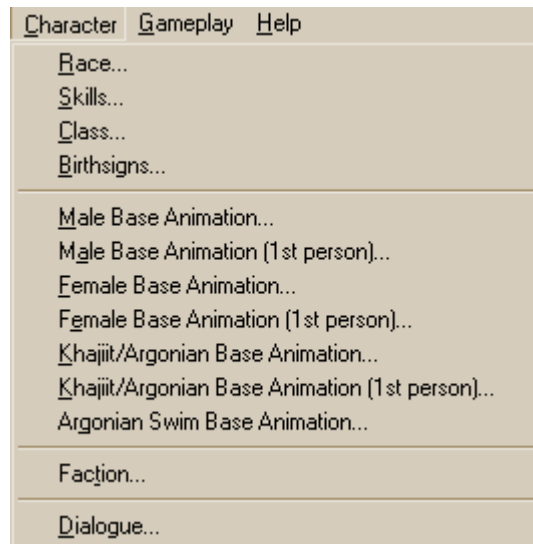
**World Testing** des outils pour tester vos créations. D'un usage limité.

**Edit Cell Path Grid** pour éditer la grille de déplacement des cellules.

**Landscape Editing** pour passer en mode édition de paysage ou le quitter.

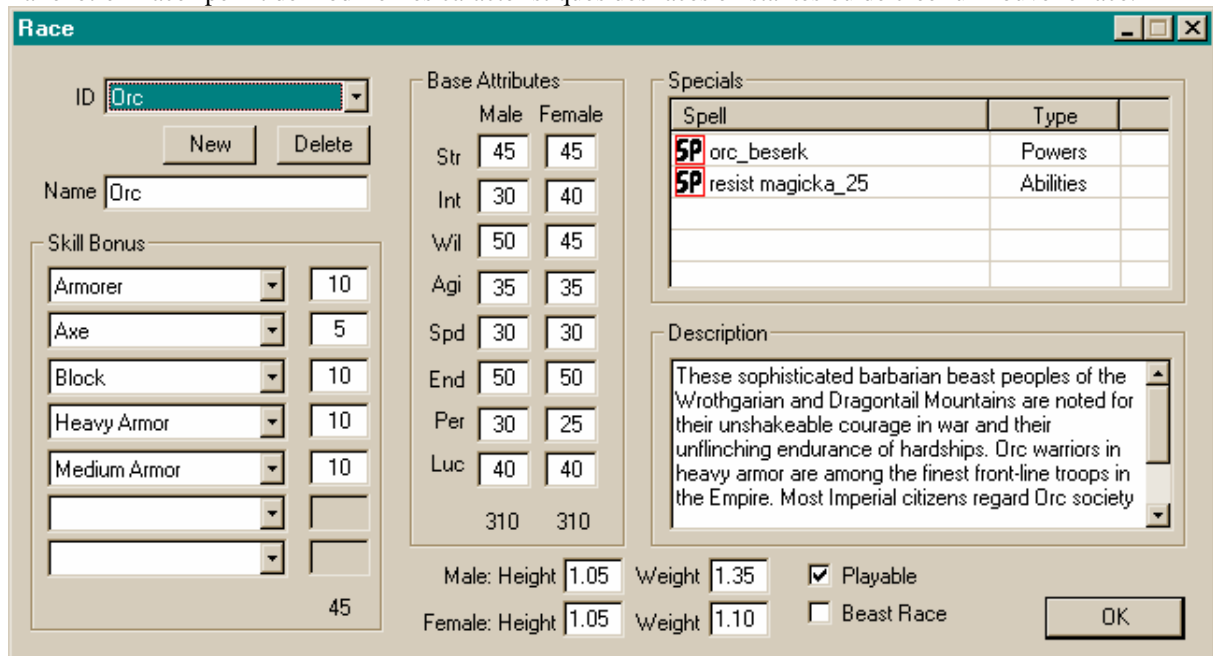
### 3.6.4 Character

Vous pouvez ajouter de nouvelles caractéristiques, races ou classes qui pourront être jouées (ou non) dans le monde de MW.



### 3.6.4.1 Race

La fenêtre "Race" permet de modifier les caractéristiques des races existantes ou de créer une nouvelle race.



**ID** L'ID unique de la race sélectionnée.

**New** pour créer une nouvelle race.

**Delete** permet d'effacer la race sélectionnée.

**Name** le nom utilise pour designer cette race dans le jeu.

**Skill Bonus** pour attribuer les bonus sur certaines compétences (au maximum 7). On peut choisir la compétence (Skill) et le nombre de points en bonus.

**Base Attributes** les caractéristiques de base des membres de cette race. Il faut bien entendu faire une différence entre les mâles et les femelles.

**Male / Female Height / Weight** pour définir la taille et le poids des membres de la race. Il s'agit d'une valeur relative par rapport à ce qu'on a coutume de rencontrer dans MW. Une valeur supérieure à 1.0 indique une taille/un poids plus grand que la moyenne. Alors qu'une valeur inférieure indique une race plus petite/légère.

**Playable** indique si le joueur peut choisir de faire partie de cette race.

**Beast Race** indique si la race est humanoïde ou non. Les races non humanoïdes ne peuvent pas porter de bottes, de chaussures ou de casque. Ce type de race utilise les animations des Argonians et des Khajiit.

**Specials** pour ajouter/modifier/retirer des capacités de races, des sorts ou des compétences particulières. On peut utiliser tous les éléments de la liste "Spellmaking" par drag'n drop depuis cette liste.

**Description** il s'agit d'une description brève de la race qui apparaît dans le jeu sous forme de Tooltip.

### 3.6.4.2 Les compétences - Skills

Ceci vous permet de modifier les propriétés des compétences de MW. Il n'est pas possible de créer de nouvelles compétences.

Skill	Actions	Use Value
Acrobatics	Jump	0.15
Governing Attribute	Fall	3.00
Strength		
Specialization		
Stealth		

Description

Acrobatics skill enables one to jump long distances and to avoid damage when falling from great heights. Nimble acrobats can reach areas others cannot get to and can direct their paths while falling.

OK

**Skill** on choisit l'un des 27 compétences du jeu.

**Governing Attribute** l'attribut gouvernant de la compétence. C'est à dire la caractéristique qui sera renforcée par une progression dans cette compétence.

**Specialization** la spécialisation à laquelle la compétence appartient.

**Actions** Montre toutes les actions possibles qui sont affectées par cette compétence.

**Use Value** détermine de quelle façon l'action associée affecte l'acquisition d'expérience pour cette compétence.

**Description** C'est la description de la compétence telle qu'elle apparaît dans le jeu.

### 3.6.4.3 Classes - Class

Ceci vous permet de voir et de modifier les classes existantes dans le jeu. Pour l'essentiel, on suit le même fonctionnement que celui qui procède à la création d'un personnage au début d'une partie.

**ID** l'ID de la classe sélectionnée

**New** pour créer une nouvelle classe

**Delete** détruit la classe sélectionnée

**Name** le nom de la classe tel qu'il apparait dans le jeu

**Primary Attributes** les deux caractéristique favorites pour cette classe.

**Specialization** la specialisation de la classe (Combat, Magie, Soins).

**Playable** si cette option est validée, le joueur se verra offrir la possibilité de jouer cette classe.

**Major Skills** les 5 compétences principales pour cette classe

**Minor Skills** les 5 compétences secondaires pour cette classe.

**Description** c'est la description de la classe. Cela ne doit être rempli que si le joueur peut choisir de jouer cette classe.

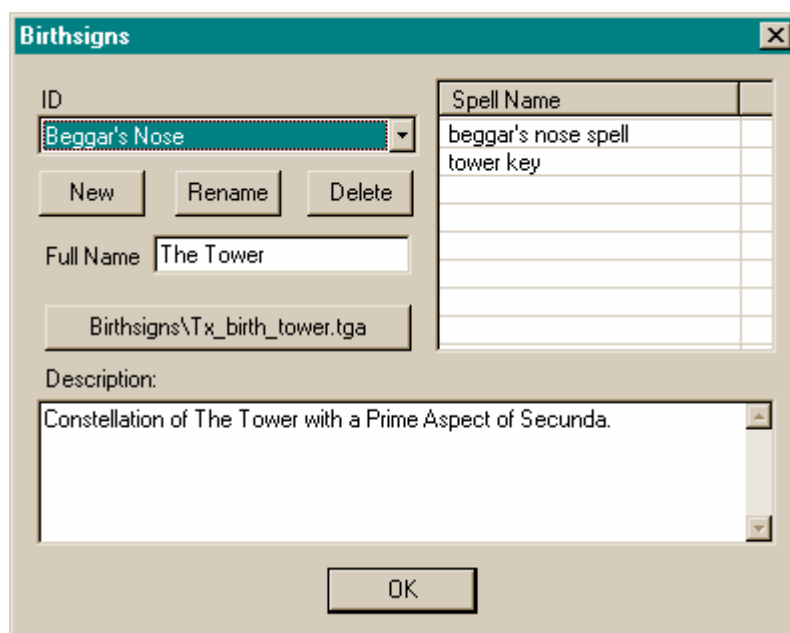
**Auto-Calc Buys / Sells** cela permet de fixer les objets vendus/achetés par les pnj quand l'option "Auto Calculate Stats" est activée.

**Auto-Calc Other** permet de fixer le type de services offert par un pnj lorsque "Auto Calculate Stats" est activée.

**OK** sauvegarde les modifications et ferme la fenêtre.

#### 3.6.4.4 Birthsigns – les signes astraux

Pour ajouter ou modifier les signes astraux et leurs propriétés.



**ID** L'ID des signes de naissance

**New** pour créer un nouveau signe astral

**Rename** pour modifier l'ID d'un signe

**Delete** efface le signe de naissance courant.

**Full Name** le nom complet du signe de naissance, tel qu'il apparaît dans le jeu.

**Constellation Image** pour associer l'image de la constellation du signe de naissance (un fichier tga).

**Description** la description du signe tel qu'elle apparaît dans le jeu.

**OK** sauvegarde les modifications et ferme la fenêtre.

La liste à droite est celle des effets du signe sur le personnage. On les ajoute par drag'n drop depuis la liste des "Spellmaking". Pour les effacer, on sélectionne la ligne et on appuie sur [DEL].

#### 3.6.4.5 Faction

Vous pouvez modifier toutes les factions du jeu ou en créer de nouvelles.

**ID** l'ID de la faction

**New** pour créer une nouvelle faction

**Delete** pour effacer la faction actuellement sélectionnée.

**Name** le nom de la faction tel qu'il apparaît dans le jeu.

**Hidden from PC** permet de signaler que la faction est cachée ou non au joueur. ( ? )

**Requirements** ici vous pouvez configurer de nombreuses choses.

les caractéristiques et les compétences (jusqu'à 7) favorites de la faction.

La liste vous permet de créer les différents rangs de la faction. Il est possible d'en créer des nouveaux en double-cliquant. Vous insérez alors le nom du rang que vous voulez créer et les exigences pour accéder à ce rang. Les deux premières valeurs sont pour les deux caractéristiques favorites. La troisième pour les deux compétences préférées et la dernière valeur indique la réputation que l'on doit avoir dans la faction ( elle augmente en faisant des travaux )

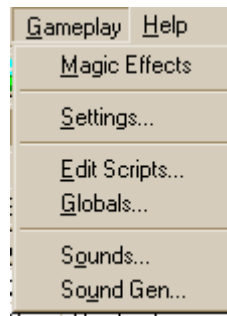
**Reactions** montre comment les membres des autres factions réagissent si le joueur est membre de la faction. Pour créer de nouvelles réactions, un clic droit et le choix de l'option "New", permettent de choisir une Faction dans la liste à droite et d'insérer une valeur dans le champ texte. Une valeur négative indique une réaction de rejet alors qu'une valeur positive indique une réaction positive. (Les valeurs choisies par Bethesda ne sont jamais importantes, comprises entre -3 et 3).

**OK** enregistre les modifications et ferme la fenêtre.

Pour plus d'explication sur les dialogues, on se référera utilement à ce qui est plus haut. Les options "Male Base Animation", "Male Base Animation (1st person)", "Female Base Animation", "Female Base Animation (1st person)", "Khajit / Argonian Base Animation", "Khajit / Argonian Base Animation (1st person)" et "Argonian Swim Base Animation" vous permettent de modifier les animations proposées pour ces mouvements.

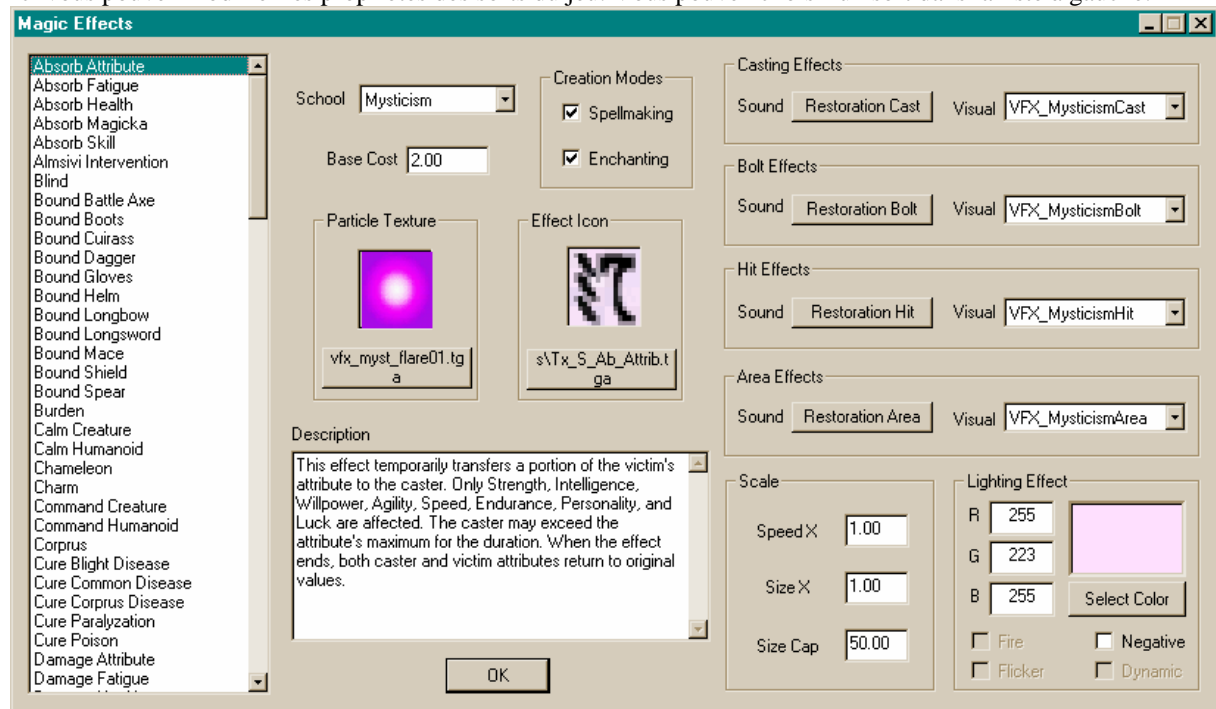
### 3.6.5 *Gameplay*

Le gameplay est l'ensemble des petites choses que vous pouvez faire dans le domaine de MW.



### 3.6.5.1 Magic Effects – la magie

Ici vous pouvez modifier les propriétés des sorts du jeu. Vous pouvez choisir un sort dans la liste à gauche.



**School** l'école de magie à laquelle le sort appartient.

**Creation Modes** pour autoriser l'utilisation du sort dans la creation de nouveaux sorts ou d'enchantement.

**Base Cost** c'est la quantité de magicka nécessaire pour invoquer le sort. Il ne s'agit pas du coût réel du sort, mais cela sert de base pour calculer ce qui sera effectivement nécessaire dans le jeu.

**Particle Texture** la texture qui est utilisée lorsque l'on invoque le sort. On peut choisir une nouvelle texture à condition que ce soit un fichier tga.

**Effect Icon** l'icone qui est affichée dans la barre de status du personnage pendant le jeu. Vous pouvez en choisir une autre en ajoutant un nouveau fichier tga.

**Description** explication sur le fonctionnement du sort.

**Casting / Bolt / Hit / Area Effects** ces quatre options determine la façon dont le sort est représenté visuellement et accoustiquement lors des différentes phases de son invocation. (invocation, déplacement jusqu'à la cible – boule de feu -, choc, ce qui se passe autour du point d'impact).

**Lighting Effect** determine la modification de la lumière ambiante par le sort. On peut choisir une couleur de la même façon que d'habitude.

**Scale** determine la vitesse de déplacement du sort ( dans le cas d'un sort à distance) (Speed X), la dimension de l'effet standard (Size X) et sa taille maximum (Scale Cap).

**OK** sauvegarde les modifications et ferme la fenêtre.

### 3.6.5.2 Settings... / Edit Scripts... / Globals...

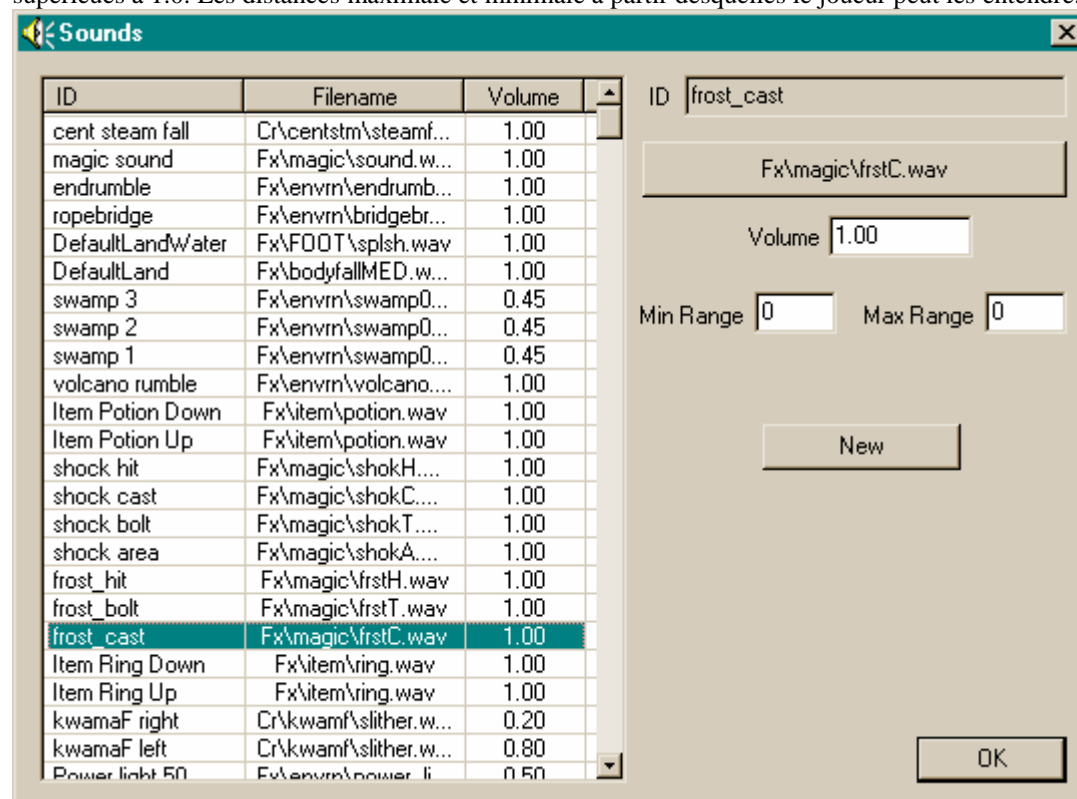
**Settings** montre essentiellement les texts affichés dans le menu du jeu et quelques autres variables sans réelle importance pour la plupart des plug normaux.

**Edit Scripts** ouvre l'éditeur de script.

**Globals** montre toutes les variables globales.

### 3.6.5.3 Sounds...

Cette fenêtre liste tous les sons (excepté les texts parlés). La liste associe les ID des sons avec les fichiers et le volume auquel ils doivent être joués. Pour chacun, il est possible de modifier le son en cliquant sur le bouton en haut à droite et en sélectionnant un nouveau fichier (qui doit être un wav). Le volume peut avoir des valeurs supérieures à 1.0. Les distances maximale et minimale à partir desquelles le joueur peut les entendre.



"New" permet de créer un nouveau son et

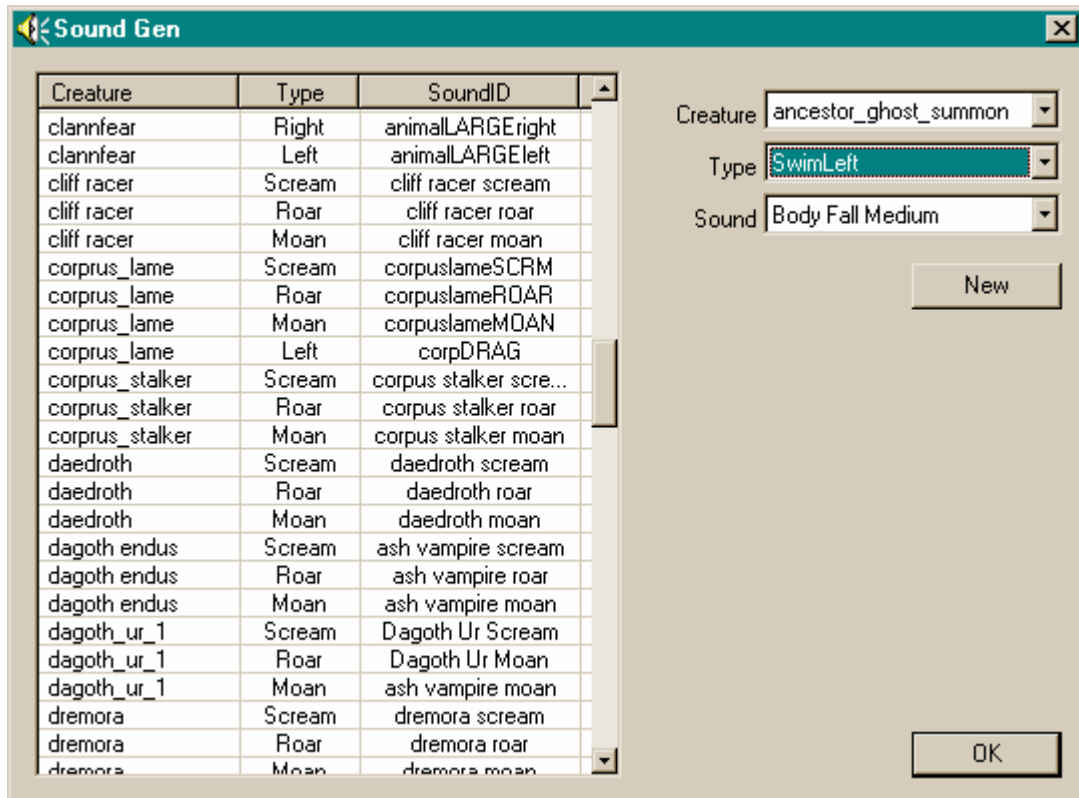
"OK" sauvegarde les modifications et ferme la fenêtre.

### 3.6.5.4 Sounds Gen...

Ici, on peut définir les sons émis par les créatures en fonction de ce qu'elle font. La liste montre toutes les associations. On peut y voir quel créatures fait quoi et le bruit qu'elle émet par son ID.

On peut créer de nouvelles associations en cliquant sur "New".

Sous "créature" on peut choisir parmi toutes les créatures qui sont chargées (celles du jeu ou celles du plug).



"Type" permet de modifier les options suivantes.

**Land** le bruit produit lorsque quelqu'un tombe et heurte le sol (SPROTCH)

**Left** bruit produit lorsque le pied gauche de quelqu'un touche le sol durant son déplacement.

**Moan** bruit produit lorsque la créature mugit.

**Right** bruit produit par le contact du pied droit avec le sol pendant le déplacement de la créature.

**Roar** bruit émis lorsque la créature rugit.

**Scream** bruit émis lorsque la créature sanglote.

**SwimLeft** bruit produit lorsque la créature nage à gauche.

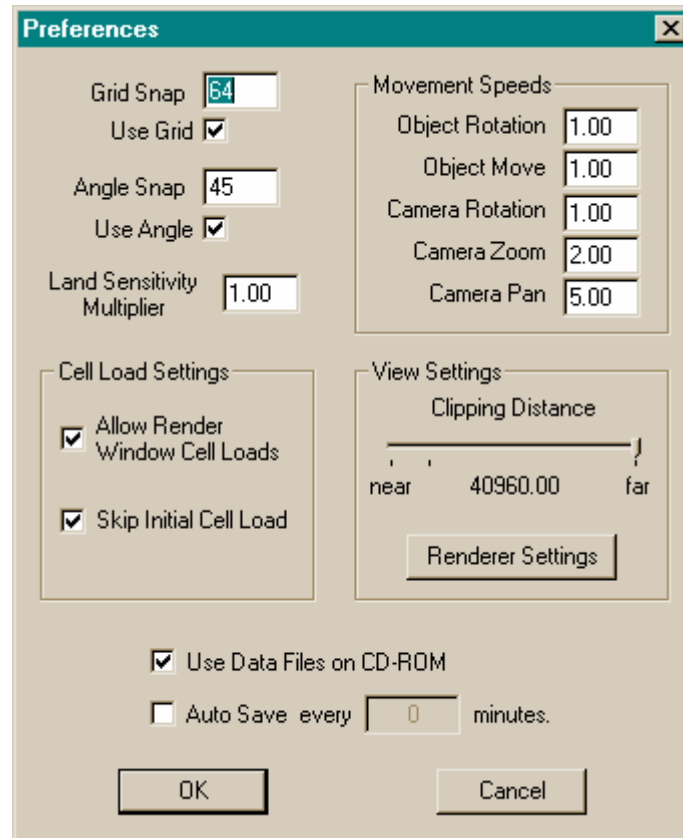
**SwimRight** bruit produit lorsque la créature nage à droite.

"Sound" permet de choisir le son émis lorsque le personnage/créature fait l'action sélectionnée.

"OK" enregistre les modifications et ferme la fenêtre.

## 4 Choisir vos réglages

C'est une étape très importante pour faire de bons plugs. En effet, les réglages standards ne sont pas optimal (loin s'en faut – cf TESAME) pour bien faire les choses. Tout d'abord, il faut ouvrir la fenêtre d'édition des préférences, c'est accessible par le biais du menu « File/ Preferences... ».



Dans la fenêtre qui apparaît, on peut ajuster quelques options qui seront très utiles pour la construction de plugs. Il faut valider les boîtes "Use Grid" et "Use Angle".

"**Use Grid**" permet de déplacer les objets sur un grille imaginaire ( comme les dessins dans Word). Le nombre que l'on choisit dans "Grid snap" indique le nombre d'unités pour le déplacement d'un objet. C'est à dire de combien de pas élémentaire on déplace l'objet. Lors de la construction de la structure de base d'une cellule d'intérieur, il est recommandé de choisir une valeur de 32 par 64, cela permet de lier plus facilement les différentes éléments du monde. Lorsque l'on place des objets tels que les fournitures, des ordures ou autre chose, il vaut mieux choisir une valeur plus faible pour donner une impression « moins géométrique » au joueur.

"**Use angle**" permet de faire tourner les objets d'un angle prédéfini dont la valeur est donnée par "Angle Snap". La prochaine étape est l'adaptation de la vitesse d'exécution des différentes actions. Dans tous les cas, il faut changer la valeur de "Object Rotation" ; elle est choisie un peut trop grande et augmente inutilement la difficulté pour aligner les objets. Il est recommandé de choisir environ 0.8 comme valeur pour la vitesse de rotation ("Object Rotation"). En revanche, la valeur pour la vitesse de rotation de la caméra est un peu faible, je vous recommande de l'augmenter légèrement, les autres paramètres sont également ajustables à votre guise pour obtenir le résultat souhaité.

NB : les réglages proposés ici ont été choisis à mon goût sur MA machine. Certains paramètres de vitesse ont une action différente en fonction de la machine sur laquelle vous travaillez

L'option "**Cell Loading Settings**" définit le comportement du TESC lors que vous charger un cellule. "Allow Render Window Cell Loads" permet au TESC de charger une nouvelle cellule, si vous sélectionnez un objet qui est à la fois intégrer dans la cellule courante et dans une cellule adjacente qui n'a pas encore été chargé, ou tout simplement si vous approcher la caméra trop près d'une cellule non encore chargée. "Skip Initial Cell Load"

empêche le chargement de la première cellule apparaissant dans la fenêtre "Cell View" lorsque vous chargez un nouveau fichier maître (\*.esm).

**"View Settings"** contient deux éléments. La barre de réglage de "Clipping Distance" (Distance de Coupure : c'est la distance à partir de laquelle on ne veut plus avoir de rendu des objets) spécifie la distance maximale de rendu dans la zone de rendu. Cela est particulièrement intéressant si votre système est un peu lent. Si vous sélectionnez "Renderer Settings", vous verrez une nouvelle fenêtre, dans laquelle vous pourrez choisir entre un mode de rendu plein-écran ou dans une fenêtre (sur mon système le choix est désactivé, et je suppose que c'est le cas sur de nombreux systèmes), si vous désirez utiliser l'accélération hardware ou software, alors que la fenêtre de rendu utilise un « mip-mapping », il faut spécifier l'adaptateur à utiliser (si vous en avez plusieurs) et la résolution avec laquelle la fenêtre de rendu.

De plus, vous pouvez choisir si les « data files » doivent être récupérées depuis le CD-ROM ou le disque Dur, et si l'éditeur doit faire des sauvegardes automatiques de vos plug.

## 5 Dialogue

Tout bon JDR se doit de permettre des interactions entre le PJ et les différents PNJ de l'univers. Sans cela, rien n'aurait de réelle saveur dans le jeu. Pas de quêtes, pas de journal rien.

La fenêtre des Dialogues que vous faites apparaître en cliquant sur la bulle dans la barre d'outils, ou en appuyant sur Dialogue dans les fenêtres des créatures ou de PNJ (NPC) est une représentation de toutes les informations de dialogues de MW.

Il est donc conseillé de manier tout ce qui apparaît dans cette fenêtre avec la plus grande précaution. Une règle de base est de n'effacer que des choses qui ont été apportés par un plug (de préférence le votre) ou vous pourriez bien définitivement perturber le fonctionnement du jeu.

Avant de commencer la moindre explication, nous allons tout d'abord décrire l'IHM. En bas à gauche, dans le menu déroulant, vous pouvez sélectionner un pnj, les possibilités de dialogue du pnj sélectionné apparaissent alors dans la fenêtre.

The screenshot shows the 'Dialogue' window with the 'Topic' tab selected. The left pane lists topics like 'Background', 'Imperial', 'J'Saddha', etc. The right pane shows a table of dialogues. The 'Speaker Condition' section includes fields for ID, Race, Class, Faction, Rank, Cell, PC Faction, and PC Rank, along with a 'Function/Variable' section. The 'Result' section shows the current state of the game. The 'Shared By' section on the right contains buttons for updating shared data, journal preview, hyperlinks, error checking, and sound filename.

Si vous ne choisissez pas de pnj, vous aurez accès à la totalité des dialogues dans la liste à gauche. C'est dans cette configuration que vous devez vous mettre si vous voulez ajouter/modifier des sujets (topic) ou des réponses pour vos pnj.

En haut à gauche, il y a plusieurs onglets, ils correspondent aux types d'informations que vous êtes en train de modifier.

**Topic** il s'agit de la liste de tous les sujets que les pnj peuvent aborder avec le joueur.

**Voice** il s'agit de tous les textes parlés qui peuvent être utilisés par les pnj. Ce que vous entendez lorsque vous passez à côté d'eux dans le jeu.

**Greeting** Les salutations, c'est la première chose que les pnj vous disent lorsque vous les activez.

**Persuasion** Les réponses que les pnj lorsque vous essayez de les persuader.

**Journal** Toutes les entrées dans le journal du joueur. Les informations à propos des quêtes, commences ou pas...

**"Shared By"** à droite, montre la liste de tous les pnj qui peuvent accéder au topic courant. C'est-à-dire la liste des pnj qui parleront du même sujet (même en disant des choses différentes). Si vous choisissez un nouveau

Topic/réponse il faut demander de mettre à jour l'information dans le panneau. Pour le faire, il suffit de cliquer le bouton "Update Shared By".

**"Journal Preview"** fait apparaître une fenêtre qui vous donne un aperçu des entrées du journal relatives au topic courant. Il faut choisir un ID de topic et un Index pour voir l'entrée de journal. Nous verrons l'utilité de ceci plus tard.

**"Update Hyperlinks"** construit et montre les liens entre les différents topics. (Attention, c'est une activité qui demande du temps et je vous recommande de l'éviter sur les systèmes lents, ma configuration est relativement solide et j'ai réussi à la planter).

**"Error Check Results"** parcourt la base de données des dialogues et ramène toutes les erreurs que l'on peut trouver dans les instructions données dans les champs "Results" (Le rapport peut également être consulté dans le fichier nommé "Warnings.txt"). Il est recommandé d'exécuter ce test avant de considérer que votre plug est fini, au moins pour vous assurer du bon fonctionnement des dialogues.

Le bouton **"Sound Filename"** n'est actif que lorsque que vous vous trouvez dans l'onglet "Voice".

Il y finalement une zone qui s'intitule **Speaker Condition**. Cette zone est primordiale. C'est elle qui régit le comportement des pnj, en fonction de certaines conditions.

**ID** permet de spécifier l'ID d'un pnj qui doit pouvoir accéder à ce topic.

**Race** Pour avoir accès à la réponse, le pnj doit être d'une certaine race.

**Class** pour associer une réponse à un type de profession du pnj.

**Faction** Certains sujets de conversation ne sont connues que certaines factions.

**Rank** Un pnj doit avoir au moins un certains rang pour connaître cette réponse. Il est évident que cette option n'a pas de valeur sans la condition de Faction.

**Cell** Le pnj doit être dans une certaine cellule. Cela a une certaine importance, si vous vous souvenez de mon commentaire concernant la création des cellules et les noms des cellules par rapport aux villes. Si vous choisissez une certaine cellulen tous les pnj qui sont dans cette cellule connaîtront la réponse ou le sujet de conversation.

**PC Faction** Pour aborder certains sujets de conversation avec le joueur, certains pnj peuvent désirer que le joueur fasse partie de certaines factions.

**PC Rank** ou ai au moins atteint un certain rang dans la faction en question. Là encore cette option n'a pas de sens si on ne s'intéresse pas à la faction du joueur.

**Sex** certains sujets de conversation sont délicats et ne peuvent aborder par tout le monde, en fonction du sexe du pnj.

**Disp** Il faut que le pnj aie une disposition minimale envers le joueur.

**Function / Variable** Ce point sera abordé plus tard.

**Result** permet de signaler à l'éditeur ce qui se passé après avoir aborder certains sujet de conversation ou bien donné certaines réponses. Les instructions que l'on donne à ce niveau sont en langage de scripts. Je vous recommande de vous reporter au paragraphe qui aborde ce point.

Lorsque l'on s'adresse à un PNJ, il parcourt le moteur du jeu parcourt la liste des entrées de dialogues et regarde si TOUTES les conditions posées dans cette section de l'IHM sont validées. Lorsqu'il en trouve une, le PNJ donne cette information, ce type de salutations etc...

Le moteur de jeu donnera toujours la première réponse qu'il trouve. Si vous voulez que être certain que le PNJ dise ce que vous voulez lui faire dire, il faut donc placer cette information au tout début de la table.

Par exemple, si je désire que Ra'Virr me salue autrement qu'en proposant ses armes daédrique, il faut que je place l'entrée de ces nouvelles salutations avant.

Chaque dialogue peut faire apparaître de nouveau sujet de discussion (à droite dans la fenêtre de dialogue du jeu), les Topic.

### 5.1 Topic – les sujets

Pour ajouter un nouveau sujet, aucun pnj ne doit être sélectionné. Dans le menu déroulant en bas à gauche vous ne sélectionnez aucun pnj.

Un clic droit dans la liste des topic fait alors apparaître un menu contextuel, dans lequel il faut choisir l'option "New". Vous pouvez maintenant créer le nouveau topic que vous vouliez créer. Le texte que vous saisissez à ce niveau est celui qui apparaîtra dans le menu de dialogue du joueur.

Vous pouvez maintenant accéder à la fenêtre "Info / Response". A ce niveau vous pouvez ajouter, modifier ou supprimer les réponses que le pnj pourra donner. Pour ajouter une nouvelle réponse, un clic droit sur la liste des réponses fait apparaître un menu contextuel dans lequel il faut choisir l'option "New". Vous pouvez maintenant

écrire la réponse qui sera donnée par le pnj. Etant donné que des pnj différents auront probablement des réactions différentes sur un même sujet en fonction de leur disposition par rapport au PJ, il est conseillé de créer plusieurs entrées. Pour chaque entrée vous pouvez définir les conditions que le pnj doit remplir avant que le sujet ou la réponse ne fasse partie de sa base de données. Le numéro de la liste des réponses est une image de la priorité qui lui est donnée. Si vous choisissez un sujet de conversation, le moteur du jeu parcourt la liste de toutes les réponses à ce sujet et choisit la réponse qui remplit toutes les conditions et qui a la plus grande priorité. Ainsi, si le pnj a quelque chose d'important à dire au joueur, il faut lui attribuer une forte priorité.

## 5.2 Voice

L'onglet 'Voice' contient tous les textes effectivement parlé dans MW. C'est à dire les réactions qu'ont certains pnj lorsque vous passez à coté d'eux. Ici vous ne pouvez pas ajouter de nouveau sujet de conversation, mais vous pouvez ajouter de nouveau échantillon de salutations aux catégories existantes. Ces échantillons seront alors utilisables par les pnj existants ou par les nouveaux. Les conditions qui apparaissent dans cette zone fonctionnent de la même manière que les conditions pour les Topics.

Le bouton "Sound Filename" permet d'ajouter un nouveau fichier son qui sera joué quand la condition sera activée, ce fichier doit être au format mp3.

## 5.3 Greeting – Salutations

Cet onglet contient toutes les salutations (ce sont les premières choses qui apparaissent dans la fenêtre de dialogue lorsque le joueur s'adresse à un pnj). Les modifications se font de la même façon que dans l'onglet Topic. Mais vous ne pouvez pas créer de nouvelles catégories de salutations. (Bethesda a créé certaines catégories, mais l'endroit où vous ajouter vos salutations n'a pas d'influence sur le jeu, seulement sur la maintenabilité de votre plug).

Vous trouvez ici les correspondances données par Bethesda aux salutations (Même si cela n'a pas d'importance essayez de les respecter)

Greeting 0	Si le PJ est un criminel
Greeting 1	Apparemment des saluts vraiment spécifiques, quand le PJ a accepté une quête donnée par le PNJ...
Greeting 2	Le PJ est un vampire
Greeting 3	Si le PNJ est membre de la guilde Morag Tong (une seule salutation)
Greeting 4	Si le PJ est malade
Greeting 5	Le PJ est malade
Greeting 6	Il y en a beaucoup, tout ce qu'il y a là dedans est relié aux quêtes
Greeting 7	Relié au faction apparemment
Greeting 8	Basés sur la réputation du personnage et sur son apparence extérieure (ses habits)
Greeting 9	Le genre de chose que les personnes pourraient vous dire dans les 'dernières rumeurs', et il y en a BEAUCOUP

## 5.4 Persuasion

Contient toutes les réponses faites lorsque le joueur use de persuasion sur les pnj, elles sont fonctions de la réussite ou de l'échec de l'action. Il n'est pas possible de créer de nouvelles catégories, celles qui sont présentes (dont les noms sont facilement compréhensibles) contiennent tout ce qui marche habituellement.

## 5.5 Journal

Cet onglet contient toutes les entrées du journal qui ont un rapport avec les quêtes. Chaque ID est associée avec un topic (principalement des quêtes). En fonction des sujets il peut y avoir une ou plusieurs entrées dans le journal. Si il n'y en a qu'une vous pouvez lui attribuer l'index 1, en revanche si il y en a plusieurs vous devez attribuer un index différent à chaque entrée. Il est important, lorsque l'on crée une entrée dans le journal de lui attribuer une ID et un index.

Cette section est très importante pour la réussite d'un bon plug. Vous ne voudriez pas que les joueurs qui utilisent votre plug soit obligés d'avoir un papier et un crayon à portée de main pour noter les informations apportées.

Une entrée du journal est en quelque un nouveau topic. Elle dispose de son ID, qui doit être unique. Puis il peut y avoir plusieurs informations à ajouter pour cette entrée. A chaque nouvelle informations vous attribuer une nouvelle valeur de l'index.

L'instruction Journal « ID » <index> ajoutera automatiquement dans le journal du joueur le texte correspondant.

### 5.6 %Defines tags

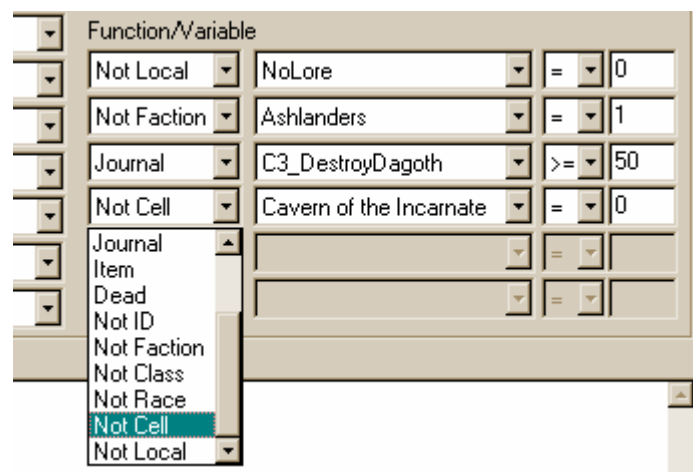
Il y a un certain nombre de mots clés qui permettent d'accéder dans un dialogue à certaines informations contextuelles.

Variable	Ce qu'on verra dans le dialogue
%Name	Le nom du pnj qui parle
%PCName	Le nom du joueur
%Race	La race du pnj
%PCRace	La race du joueur
%Class	La classe du pnj qui parle
%PCClass	LA classe du joueur
%Faction	La faction du pnj
%Rank	Le rang du pnj dans sa faction
%PCRank	Le rang du joueur dans la faction du pnj
%NextPCRank	Le prochain rang du joueur dans la faction du personnage qui parle
%Cell	Le nom de la cellule dans laquelle le pnj se trouve
%Global	Ici, on peut remplacer Global par n'importe quelle variable globale qui existe

### 5.7 Function / Variable

Dans la section **Speaker Condition**, il y a une zone qui contient de nombreux menus déroulants. Cette zone permet d'affiner les conditions pour qu'un PNJ dise une chose ou une autre.

Il est possible d'« interroger » l'état du PNJ, ou du journal ou d'une variable...



Le premier menu déroulant vous permet de sélectionner le type d'information que vous cherchez.

Le second menu déroulant change en fonction de ce que vous avez sélectionné dans le premier et vous offre le choix parmi ce qui existe dans le jeu pour l'option sélectionnée.

Le troisième vous permet de choisir le test que vous faites sur la valeur de retour de la fonction.

Enfin la dernière section permet de donner une valeur à vérifier.

Le tableau suivant donne la signification de ce que vous sélectionnez dans le premier menu déroulant. Des informations plus détaillées sont fournies dans l'annexe Function/Variables pour les dialogues

Choix	Ce qui est fait
Function	Permet de récupérer un certain nombre d'information du monde (1 vrai 0 faux) Les descriptions des fonctions proposées sont fournies en annexes

Global	Permet de récupérer la valeur d'une variable globale du jeu
Local	Permet de récupérer la valeur d'une variable locale
Journal	Permet de tester si le journal contient une entrée (on sélectionne ensuite l'ID de l'entrée puis son index) La valeur proposée est comparée à l'état actuel de l'index dans le journal.
Item	Permet de vérifier si le joueur a un objet spécifique dans son inventaire. La valeur à préciser est alors la quantité d'objet détenue par le joueur.
Dead	Donne le nombre de d'individus ayant la même ID que l'interlocuteur tués par le joueur
Not ID	Le PNJ n'a pas une ID donnée (1 vrai, 0 faux)
Not Faction	Le PNJ n'appartient pas à une certaine faction
Not Race	Le PNJ n'est pas d'une certaine race
Not Class	Le PNJ n'est pas d'une certaine classe
Not Cell	Le PNJ n'est pas dans une cellule
Not Local	De même que pour Local mais la condition n'est pas validée

### 5.8 La section Result

Dans cette section, vous disposez d'un certain nombre de fonctions qui permettent de préciser des opérations à faire lorsque le PNJ prononce le texte sélectionné.

Les instructions données ici utilisent le langage de script de l'éditeur.

## 6 Les Fichiers de l'éditeur

### 6.1 Les fichiers esm et esp

Le TESC fonctionne avec deux types de fichiers les fichiers esm et les fichiers esp (respectivement pour Elder Scroll Master et Plugs).

Le fichier esm est le fichier maître, c'est à dire celui qui contient toutes les informations pour la gestion du monde, du gameplay, les modèles 3D et tout le reste de base.

Si on veut complètement recréer un nouveau jeu, il existe ( probablement chez wiwi <http://perso.club-internet.fr/lowi/>, sinon ailleurs chercher vous trouverez quelque chose qui s'appelle LeifMaster) un esm vide qui permet de tout recréer depuis zéro. Il n'est pas nécessaire de préciser qu'il s'agit d'un travail considérable ...

Une autre solution existe, un convertisseur de fichier esp en fichier esm.

Dans le cas de morrowind, le fichier de base pour le monde est le fichier morrowind.esm que l'on trouve dans le répertoire d'installation de MW.

Les fichiers esp, sont les fichiers qui contiennent les informations supplémentaires par rapport au contenu de l'esm pour que vos plugs fonctionnent. Ils sont donc en théorie de petites tailles sauf cas contraires où les auteurs ont rajoutés des modèles ou des textures.

En annexe de ce document vous trouverez une analyse de la structure des fichiers esp.

C'est à partir de ces analyses que des programmes tels que mGen, TESAME, GenMod on pu voir le jour.

Nota Bene : Les changements que l'on apporte dans les plugs sont toujours prioritaires par rapport aux données du esm. C'est ce qui a rendu possible le plug de traduction de Bamb et de sa fine équipe.

### 6.2 Charger des données

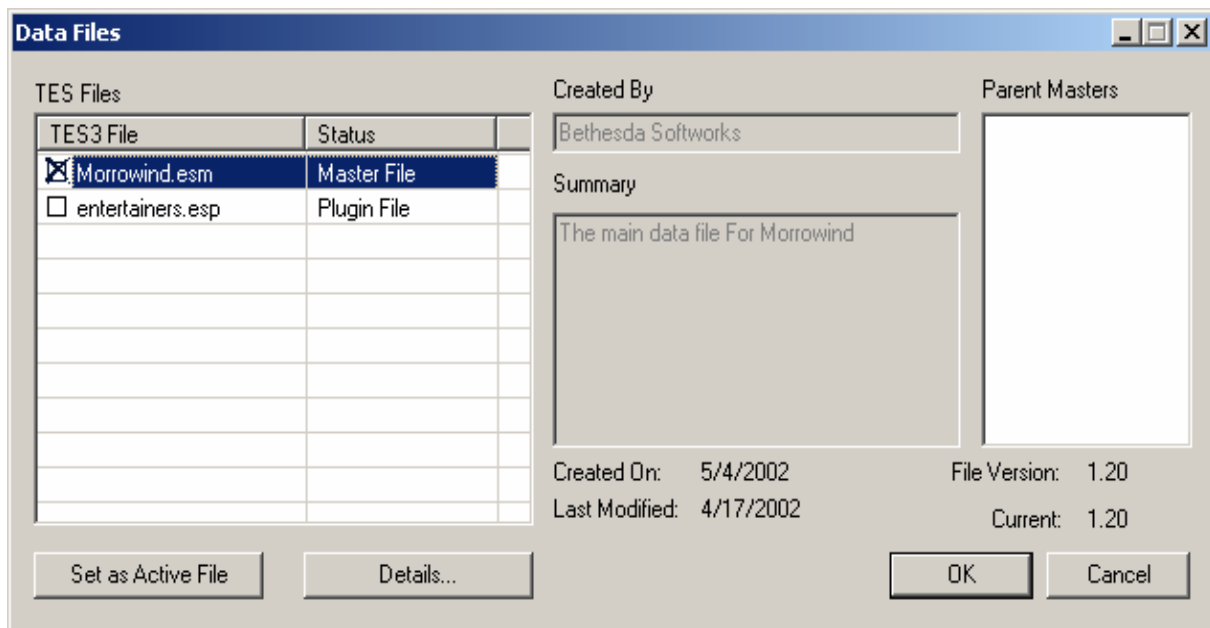
Avant de commencer la création de vos plugs, il est nécessaire de charger les « data-files » sur lesquels vous allez travailler.

Pour faire cela cliquer sur le bouton « Open file » dans la barre d'outil ou sélectionner "File/Data Files...". Cela fait apparaître la fenêtre des « Data Files ».

Là vous choisissez "Morrowind.esm" en double cliquant dessus. Morrowind.esm est le « master file » de MW. Il est également possible de charger d'autres fichiers de plug pour travailler dessus. (Si vous utilisez le travail d'autres moddeurs, alors soyez polis et signalez le. Après lui avoir demandé, par mail, si il vous y autorise à la faire).

Si vous voulez sauvegarder vos changements sur un plug existant alors déclarez le comme fichier actif, c'est à dire qu'il faut le mettre en surbrillance puis appuyer sur le bouton "Set As Active File". En revanche, si vous voulez créer un nouveau plug, ne déclarer aucun plug comme fichier actif. On vous donnera la possibilité de choisir un nom la première fois que vous sauvegarderez les changements.

Il est inutile de préciser qu'il est déconseillé de déclarer Morrowind.esm comme fichier actif... les changements seraient irréversibles et peuvent s'avérer dangereux. (Pour tout dire je ne sais même pas si c'est possible, mais dans le doute, je préfère prévenir).



Maintenant que nous connaissons les bases de l'ihm et du chargement de fichier nous allons regarder la création des éléments du monde.

Pour la construction d'une extension, on peut être amené à construire deux types de cellules, les cellules extérieures et les cellules d'intérieures.

Les cellules d'extérieures sont des modifications ou des créations de secteur géographiques. On y édite des paysages par modifications des élévations de zones de paysages, on y ajoute ensuite des arbres, des plantes etc...

Les cellules d'intérieures sont les zones du monde qui correspondent à l'intérieur d'une maison, d'une tombe, d'une mine etc...

### 6.3 Les modèles 3D et les textures

Les modèles 3D que vous voyez dans le jeu sont des fichiers NIF (NetImmerse File). Il s'agit d'un format propriétaire et vous pouvez trouver des informations sur Internet à son sujet.

Ce qui nous intéresse à ce niveau, c'est lorsque vous voulez créer une nouvelle créature, une nouvelle race, un nouvel objet ou quoi que ce soit qui doit être vu dans l'univers du jeu, il faudra spécifier à l'éditeur le fichier nif qu'il devra utiliser.

Bethesda a mis à la disposition de la communauté des moddeurs un outil qui permet de convertir des fichiers 3DStudio Max en fichier nif.

Il est donc nécessaire de disposer soit de 3DS soit d'un modèleur 3D et d'un ami qui pourra vous faire les conversions.

Pour l'information, il existe une version gratuite de 3DS **Gmax**, qui est plus limitée mais qui présente l'avantage d'un prix tout à fait abordable. (<http://www.discreet.com>)

À l'heure où j'écris ces lignes, je n'ai pas testé ce modèleur et surtout la compatibilité avec le convertisseur de Bethesda, mais je n'ai pas vu d'informations allant dans ce sens non plus.

Tout modèle 3D doit être Skinné, c'est-à-dire qu'on doit lui appliquer des textures. Ces textures doivent être soit au format tga ou au format bmp soit au format dds (ce dernier étant plus compressé – 50% de moins, je ne peux que vous recommander de le choisir, pour limiter la taille des mods créés et ainsi permettre à nos pauvres modems de ne pas trop ramer). L'avantage des fichiers dds, c'est qu'il n'est pas besoin de changer le nom de la texture dans le fichier nif – (Expertise des fichiers binaires). Les textures doivent obligatoirement avoir un format de 16x(un multiple de 2) (ou bien 16 par 64) dans les deux directions pour être acceptées dans MW. (par 16x32 ou 32x64...)

Le format dds est particulier et là encore Bethesda met à disposition de la communauté un plug pour les lire et les modifier dans Photoshop. (L'installation en est simple, il suffit de copier un fichier dans le répertoire plugins de photoshop). Ce même plug peut être ajouté à 3DS, pour y lire les textures.

La seule limite à l'utilisation de ces textures (dds) est pour les sols, en extérieur. Là seul les fichiers tga sont acceptés.

#### 6.3.1 Les fichiers dds

Pour les créer dans Photoshop,

Télécharger le fichier **dds.8bi** sur le site <http://developer.nvidia.com/docs/IO/1176/ATT/dds.8bi>

Le mettre dans le répertoire 'modules externes/formats de fichier/ de photoshop. (version 6 fr)

Pour les utiliser dans 3DS :

Télécharger le fichier **dds.bmi** sur le site <http://developer.nvidia.com/docs/IO/1176/ATT/dds.bmi>

Ou sur le site <http://developer.nvidia.com/docs/IO/1176/ATT/3dsmax5dds.zip>

Le mettre dans le répertoire 'stdplugins/' de 3DS. Faites attention à avoir le bon plug en fonction de votre version de 3DS.

#### 6.3.2 Modéliser dans 3DS

Ce n'est pas le lieu de vous apprendre la modélisation sous 3DS, il existe de nombreux tutoriaux sur le net pour ça. Tous mieux fait que ce que je ne pourrais faire.

En revanche, voici quelques astuces que j'ai pu trouver sur les divers forums MW :

- faites attention au positionnement des objets dans 3DS. En effet il faut faire attention lors de l'exportation de la scène à ne pas avoir coché l'option 'Pivot au centre de la sélection'. Il faut savoir que le TESCS utilise comme centre le point de pivot de l'objet au moment de l'exportation.
- Le TESCS récupère les informations de rotation du pivot, c'est-à-dire que lors de la conversion, une fois que vous avez placé l'objet comme vous voulez l'exporter, il faut penser à remettre à 0 les informations de rotation du pivot.
- Enfin le TESCS récupère aussi la position du pivot (qui est enregistrée par rapport à l'origine de l'univers). Un objet enregistré à Z=5m apparaîtra dans le TESCS à 5m
- Pour éliminer les problèmes de taille entre ce que vous dessinez dans 3DS et ce que vous voyez dans le TESCS. Pour que vos modèles apparaissent directement à la bonne taille dans le TESCS ; il suffit de dessiner en unité standard dans 3DS et savoir que de cette façon, un pnj mesure 128 unités de haut. Une unité standard de 3DS est strictement égale à une unité MW.

#### Une aide au skinnage des modèles 3D

Il existe disponible sur le net des outils qui fournissent une aide non négligeable pour réaliser les textures de modèles 3D.

Je vous en cite en vrac quelques uns :

Unwrap3D, ChilliSkinner.....

Remarque : il est possible de modifier les textures appliquées sur un objet. Ceci est expliqué en annexe, Reskinner un nif. Il faut avoir un éditeur hexadécimal.

#### Des sites de modèles

Il y a également de modeleurs qui mettent leur travail à disposition de la communauté (Lady Eternity par exemple).

Vous pouvez aussi utiliser des modèles construits par d'autres moddeurs, à la condition de le leur signaler et de citer les auteurs dans le readme de votre mod. (C'est du savoir vivre).

#### Les icônes

Les icones, c'est ce qui apparaît dans votre inventaire ou dans les informations en cours de jeu. Elles doivent au format tga.

#### Les fichiers de son

Dans MW il existe deux types de sons, ceux qui sont enregistrés dans des fichiers au format wav et ceux qui utilisent le format mp3.

Les fons sonore (musique d'ambiance) sont au format mp3.

Les sons associés à des mouvements de créatures, à ces paroles dites par les pnj sont au format wav.

### Où se trouvent les fichiers du jeu?

Pour tous ceux qui ont acheté le jeu, vous trouverez tous les fichiers modèles, textures, icônes et sons sur le second CD.

Pour les autres, qu'attendez vous pour être honnête ?

### L'organisation des fichiers

Cette information a été construite en s'inspirant d'un tutoriel écrit par Sorcerer pour le site Darkblade.

### Le fichier esm

Le fichier morrowind.esm contient TOUTES les données disponible sur le second CD du jeu, en plus du monde du jeu. Il contient également la description de tous les scripts et du reste.

Mais ne vous inquiétez pas, vous n'avez pas besoin de connaître tous les noms de tous les fichiers de données. Si jamais vous créer un modèle avec le même nom qu'un modèle existant dans le jeu, le moteur chargera celui qui est stocké dans le répertoire « Data Files », qui contient les données des plugs.

Les modifications apportées par un plug sont toujours prioritaires sur le contenu du fichier esm, de même les données dans le répertoire 'Data Files' sont toujours prioritaires sur celles qui sont contenues dans le fichier esm.

Une application de cette propriété simple, le changement des textures du sol. Vous n'avez qu'à mettre les fichiers de textures au bon endroit dans le répertoire 'Data Files' et ce seront celles-ci qui seront prises pour la représentation dans le jeu.

Pour faciliter la gestion des fichiers nombreux qui existent, et pour savoir ce qu'ils contiennent sans les visualiser, Bethesda propose un système de nommage des fichiers.

### Le système de nommage

Intéressons nous à un exemple simple d'un fichier modèle:

B\_N\_DarkElf\_M\_Head01.nif

B: indique qu'il s'agit d'une partie de corps ET que ce fichier doit se trouver dans le repertoire B

N : une facilité de Bethesda pour dire que ce n'est pas un vampire

DarkElf : la race correspondante

M : indique qu'il s'agit d'un homme (Male) et F indiquerait une femme (Female)

Head : la partie du corps concernée

01 : au cas où il y a le choix

nif : c'est un fichier NIF

Et pour un fichier de texture ?

TX\_B\_N\_DarkElf\_M\_Head01.bmp

Le préfixe TX indique qu'il s'agit d'une texture, et vous pouvez reconnaître les autres options.

En connaissant ce système vous trouverez facilement ce que vous cherchez et comprendrez rapidement à quoi servent les fichiers que vous trouvez.

### La structure des répertoires

Le second CD du jeu contient, comme nous l'avons déjà vu, tous les modèles et toutes les données du jeu. L'arborescence ressemble à ceci :

Game Data

- \_ Meshes : contient tous les fichiers nif
- \_ Textures : contient les textures du jeu
- \_ Icons : contient les fichiers des icônes utilisées pour le jeu
- \_ Bookart : contient les images et les polices utilisées pour représenter les livres et les parchemins

Le répertoire Meshes contient les sous répertoires :

- A – Armure
- B – les parties de corps (Body Part)
- C – les habits (Clothes)
- D – les portes (inclue les entrées des cavernes)

- E – Les effets des sorts
- F – les fournitures - les accessoires divers
- I – des éléments d'intérieur et les ponts
- L – les sources de lumière
- M – des choses inclassables ailleurs (Misc)
- N – les ingrédients pour l'alchimie
- O – les conteneurs
- R – les modèles des créatures
- W – les armes
- X – d'autres éléments d'intérieur
- Dans le répertoire Meshes toutes les animations spéciales, ciel, eau et les curseurs

Comme vous pouvez le constater, il est relativement simple de trouver rapidement ce que l'on cherche. Je vous conseille d'essayer de respecter cette arborescence quand vous ajouterez vos modèles, cela vous permettra de mieux vous y retrouver.

Le répertoire des Textures contient toutes les textures normales et les sous répertoires :

- BirthSign : qui contient les images des signes astro (tga)
- Levelup : contient les images qui apparaissent lorsque le PJ change de niveau (elles varient avec la classe)
- MagicItem : l'effet visuel (certains l'appellent sac poubelle) qui montre qu'un objet est magic, c'est une animation de 0-31 au format tga
- Nvwater : le déplacement des textures pour l'eau
- Water : pareil qu'au dessus

Le répertoire des Icones contient les icônes par défaut et les icône des menus ainsi que les sous répertoires :

- A – les armures
- C – les habits (dans l'inventaire)
- K – les icônes des compétences
- L – les icônes des sources lumineuses
- M – des choses diverses de l'inventaire
- N – les ingrédients pour la magie
- S – les sorts
- W – les armes dans l'inventaire/

Enfin le répertoire Bookart, ne contient que des images des polices pour les livres.

## 7 La création des intérieurs

Pour comprendre le TESCS et s'exercer la création d'intérieurs est la meilleure option qui se présente à vous. De plus si votre système est un peu lent, je vous le recommande doublement du fait que la création d'extérieur est extrêmement gourmande en ressource. (Moins de 1fps dans la fenêtre de rendu). Dans la suite de ces explications, nous suivrons la même démarche que le tutorial de création d'un dungeon dans le TESCS.

### 7.1 Un objet pour des references

Avant de commencer, nous allons définir un nouveau terme: "référence". Jusqu'à présent nous avons essentiellement parlé d'« Objets », ce terme est peu approprié. En effet, il se réfère aux créatures, objets (choses) tels qu'ils existent dans la base de donnée principale de MW. Il faut savoir que si on place un objet par "drag'n drop" dans le monde du jeu, on crée une référence vers celui-ci, pour parler en terme de programmation orientée objet, on crée une instance de cet objet dans le monde. C'est à dire que l'on crée un élément qui correspond à, et qui aura le même comportement que celui que l'on a défini pour l'objet. Cette « référence » aura le même comportement que celui que l'on a défini pour tous les objets de la même famille. On peut par exemple créer plusieurs références de Guar qui auront un comportement similaire, et surtout les mêmes caractéristiques de bases. La distinction entre la 'référence' et 'l'objet' est très importante quand on change les caractéristiques d'un 'objet'/une 'référence'. La majorité des fenêtres de spécifications sont constituées de deux parties: l'une en haut de l'écran, et l'autre sous l'intitulé "Reference Data". Les changements apportés dans la zone "Reference Data" n'ont d'effet que sur cette référence de l'objet. (par exemple, on peut modifier les coordonnées de la créature), cela n'affectera que la référence sélectionnée).

PC ...	Creature Name
2	scrib
2	Rat
3	kwama forager
4	nix-hound
5	scrib diseased
5	kagouti
5	guar_feral
6	rat_diseased
6	alit_diseased
6	kannuti

**Reference Data**

**Position**  
 X: -52800.000 1.00  
 Y: 32000.000 1.00  
 Z: 512.000 1.00

**Rotation**  
 X: 0.0 1.00  
 Y: 0.0 1.00  
 Z: 0.0 1.00

**3D Scale** 1.00

**Extra Data**  
 Health Left:   
 Soul:

**Assignable Data**  
 Owner:   
 Global Variable/Rank:   
 Apply to Selection

**Teleport**  
 Load Cell: Abaelun Mine - Interior  
 Select Marker

**Locked**  
 Level:   
 Key:   
 Trap:

☐ Reference Blocked

En revanche, les modifications apportées au dessus de la barre 'Reference Data ' auront une répercussion sur toutes les références existantes dans l'univers. Dans la fenêtre ci dessus, une modification du pourcentage de rencontre aura une répercussion dans l'ensemble du jeu.

Il existe néanmoins une manière très simple d'apporter des modifications dans la partie supérieure de cete fenêtre sans modifier l'ensemble de l'univers. Pour cela, faites les modifications que vous désirez dans la partie supérieure de la fenêtre, attribuer une nouvelle ID (on se rappellera qu'elle doit être unique dans l'univers du jeu, il s'agit en quelque sorte du 'type ' d'objet), puis confirmer la création d'un nouvel objet lorsque vous sauvegarderez votre plugs.

## 7.2 Créer une cellule d'intérieur

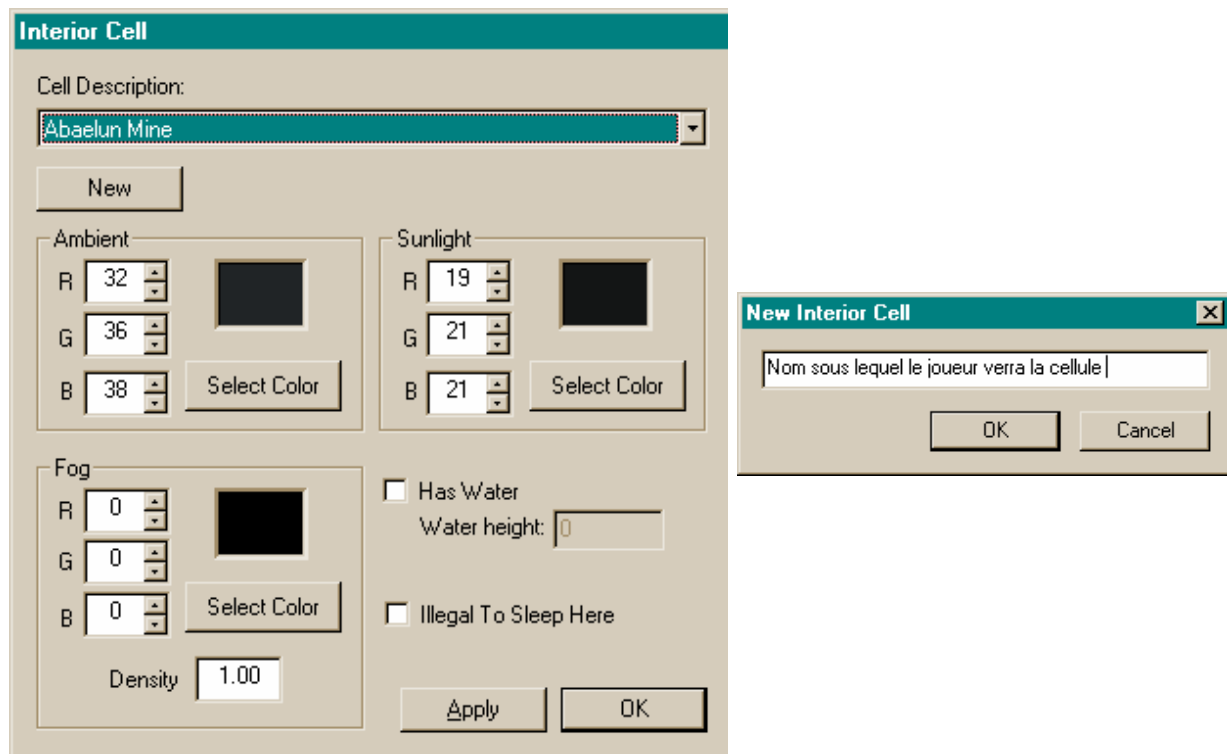
Avant toute chose, il faut créer la 'cellule', pour cela on suit le menu "World/Interior Cell...", cela fait apparaître la fenêtre "Interior Cell". On sélectionne "New" pour créer une nouvelle cellule.

Dans le cas où on désire reprendre un travail que l'on effectuait sur une cellule, on ne crée pas de nouvelle cellule, on recherche dans le menu déroulant 'Cell Description' la nom de la cellule que l'on a créée.

En ce qui concerne les noms de cellules que vous allez créer, il faut être très précautionneux. En effet, le nom de cellule que vous allez choisir, sera celui que le joueur verra dans le jeu. Il doit être unique, car il s'agit également de l'identifiant (ID) de la cellule dans le jeu. Pour la petite histoire c'est cette 'boulette' des développeurs de Bethesda qui a rendu incompatible les mods VF avec la VO et vice-versa.

Pour les noms de cellule, il faut donc éviter les mots de code du type 'Mon\_ModAmoi\_Le\_Shrine\_3'J .

Soyez également vigilant à ne pas commencer le nom de la cellule par le nom d'une ville du jeu. Sinon tous les PNJ que vous pourriez ajouter dans la cellule disposeraient automatiquement des choix de dialogues de la ville en question. Et aussi loin que je sache, il est impossible de les enlever sans enlever en meme temps pour tous les habitant de la ville. La conséquence de ceci, est que si vous voulez créer une nouvelle maison dans une ville, et y ajouter des PNJ, pour la cohérence du jeu, il est conseillé de nommer cette maison en commençant par le nom de la ville. Par exemple : 'Seyda Neen Maison des Plaisirs' ;) A ce moment là tous les PNJ ajoutés dans la maison seront automatiquement des citoyens de la Seyda Neen.



Après avoir choisit un nom, on peut modifier quelques paramètres. Premièrement, on peut redéfinir les couleurs des lumières, ambiante et solaire. Les modifications que vous apportez à ce niveau donne un cachet à votre cellule. Les couleurs "Ambient Light" et "Sunlight" sont définies en mode RVB (Rouge Vert Bleu ou RGB).

Vous pouvez jouer sur les poids des différentes composantes de couleur les valeurs sont des entiers compris entre 0 et 255.

La zone "Fog" (Brouillard) vous permet de créer un brouillard avec une couleur spécifique et une densité. (Plus la valeur est grande plus on obtient une purée de pois épaisse). En intérieur, le brouillard apparaît si la profondeur de champ choisie par le joueur ne permet pas d'afficher toute la zone.

En sélectionnant "Has Water" on spécifie si il y a de l'eau à l'intérieur de la cellule, en conséquence on définit aussi la hauteur de l'eau, elle est donnée en relatif à partir de z=0.

Une remarque, l'eau sera affichée dans la fenêtre de rendu, il est donc recommandé de la rajouter après avoir bien équipé les éléments de la cellule qui seront sous l'eau.

Finalement, vous pouvez spécifier qu'il est illégal de dormir dans la cellule, c'est ce qui se passe dans les maisons, dans les villes. Si cette option est cochée le PJ verra un message différent lorsqu'il désirera se reposer.

Une fois que vous avez fait tous ces choix, vous cliquez sur OK et...

Et rien ne se passe. Ne vous inquiétez pas le travail que vous venez de faire n'est pas perdu. Dans la liste de gauche de la fenêtre Cell View, recherchez le nom de la cellule que vous venez de créer et double cliquez dessus... maintenant c'est bon, vous ne voyez toujours rien dans la fenêtre de Rendu, mais c'est un rien d'une couleur différente.

### 7.3 La structure de base

Avant de commencer un plug, vous devez au moins avoir une vague idée de ce que vous voulez faire. Il est évident que plus l'idée est claire, plus vous irez vite et meilleur sera le résultat. Le top est d'avoir un plan sur papier, même grossièrement tracé.








































Vous devez tout d'abord vous intéresser au style architectural de votre cellule, sera ce une hutte de Ashlander (Cendrés) un palace Telvani ou encore une grotte?

Nom de l'architecture	Exemple dans le monde
Ashlander	Les quatre campements des Ashlanders : Urshilaku Ahemmusa, Zainab, Erabeninsum
Bonecave	Utilisé pour toutes les grottes du monde
Common	L'architecture impériale standard : Pelagiad, Caldera
Daedric	L'architecture inimitable des ruines daedric Ald Sotha, Bal Fel...
Dwarven	Utilisée pour les ruines des constructions du peuple Dwener Arkngthand, Dagoth Ur
Hlaalu	Ce qu'on peut voir à Balmora ou Suran
Imperial Fort	Les forts de l'empire : Buckmoth, Moonmoth
Lavacave	Les grottes d'origine volcanique
MoldCave, MudCave	Les grottes standard, une différence dans les textures appliquées sur les parois
Nord	Architecture nordique : Dagon Fel
pycave	Autre grotte d'origine volcanique mais la roche est plus sombre (un peu comme du basalte)
Redoran	Ald'ruhn ou Maar Gan
Sewer	Les égouts comme à Vivec
Stronghold, Strongruin	Les anciennes forteresses des dunmer : Kogoruhn, Hlormaren
Telvani	Ils sont fous, ils font pousser leur maison, c'est génial Sadrith Mora
Veloithi	Les tombes ancestrales dans les quartiers de Vivec

Toutes les pièces sont créées par drag'n drop.

Sélectionner l'onglet 'Static' dans la fenêtre des objets. Vous avez maintenant la liste de tous les static utilisables dans le monde. Les objets Static, sont des objets avec lesquels le PJ ne peut pas interagir excepté pour se cogner, les murs, les tables, les chaises...

Il existe plusieurs catégories de static. Elles sont identifiables par l'ID qui apparaît dans la colonne de gauche. Cette ID commence par un mot qui caractérise la catégorie que l'on traite.

Object Window												
Activator	Apparatus	Armor	Body Part	Book	Clothing	Container	Door	Ingredient	Light	Lockpick	Misc Item	Probe
Repair Item	Static	Weapon	NPC	Creature	Leveled Creature	Spellmaking	Enchanting	Alchemy	Leveled Item			
ID	Count	Model	Persists	Bloc...								
 ex_vivec_lp_02	31	x\Ex_Vivec_lp_02.NIF	no	no								
 Ex_Vivec_P_water_01	1	x\Ex_Vivec_P_water_01.NIF	no	no								
 ex_vivec_palace_01	0	x\Ex_Vivec_palace_01.NIF	no	no								
 Ex_Vivec_pd_01	1	x\Ex_Vivec_pd_01.NIF	no	no								
 Ex_Vivec_pq_01	1	x\Ex_Vivec_pq_01.NIF	no	no								
 Ex_Vivec_pq_02	1	x\Ex_Vivec_pq_02.NIF	no	no								
 Ex_Vivec_pq_03	1	x\Ex_Vivec_pq_03.NIF	no	no								
 Ex_Vivec_pq_04	1	x\Ex_Vivec_pq_04.NIF	no	no								
 Ex_Vivec_pqs_01	1	x\Ex_Vivec_pqs_01.NIF	no	no								
 Ex_Vivec_pqs_02	1	x\Ex_Vivec_pqs_02.NIF	no	no								
 ex_vivec_prisonmoon_01	1	x\Ex_Vivec_prisonmoon_01.NIF	no	no								
 Ex_Vivec_ps_01	1	x\Ex_Vivec_ps_01.NIF	no	no								
 ex_vivec_sbase_01	1	x\Ex_Vivec_Sbase_01.NIF	no	no								
 ex_vivec_telt_01	1	x\Ex_Vivec_TelT_01.NIF	no	no								
 ex_vivec_w_01	70	x\Ex_Vivec_W_01.NIF	no	no								
 ex_vivec_w_02	0	x\Ex_Vivec_W_02.NIF	no	no								
 ex_vivec_w_03	1	x\Ex_Vivec_W_03.NIF	no	no								
 ex_vivec_w_c_01	76	x\Ex_Vivec_W_C_01.NIF	no	no								
 ex_vivec_w_e_01	94	x\Ex_Vivec_W_E_01.NIF	no	no								
 ex_vivec_w_g_01	47	x\Ex_Vivec_W_G_01.NIF	no	no								
 ex_vivec_w_slope_01	0	x\Ex_Vivec_W_slope_01.NIF	no	no								
 ex_vivec_waterspout_01	70	x\Ex_Vivec_waterspout_01.NIF	no	no								
 ex_vivec_waterspout_02	446	x\Ex_Vivec_waterspout_02.NIF	no	no								
 ex_vivec_waterspout_03	18	x\Ex_Vivec_waterspout_03.NIF	no	no								
 Ex_Vivec_waterspout_05	8	x\Ex_Vivec_waterspout_05.NIF	no	no								
 Ex_Vivec_wspout_d_02	8	x\Ex_Vivec_wspout_d_02.NIF	no	no								
 ex_wg_cave_entrance_01	2	x\Ex_WG_cave_entrance_01.NIF	no	no								
 flora_ash_grass_b_01	1065	\Flora_Ash_Grass_B_01.NIF	no	no								
 Flora_Ash_Grass_R_01	786	\Flora_Ash_Grass_R_01.NIF	no	no								
 flora_ash_grass_w_01	2049	\Flora_Ash_Grass_W_01.NIF	no	no								
 flora_ash_log_01	40	x\Flora_ash_log_01.nif	no	no								
 flora_ash_log_02	110	x\Flora_ash_log_02.nif	no	no								
 flora_ash_log_03	78	x\Flora_ash_log_03.nif	no	no								
 flora_ash_log_04	51	x\Flora_ash_log_04.nif	no	no								
 Flora_ashtree_01	441	x\Flora_ashtree_01.NIF	no	no								
 flora_ashtree_02	442	x\Flora_ashtree_02.NIF	no	no								
 flora_ashtree_03	660	x\Flora_ashtree_03.NIF	no	no								
 flora_ashtree_04	1078	x\Flora_ashtree_04.NIF	no	no								
 flora_ashtree_05	735	x\Flora_ashtree_05.nif	no	no								

**dwrv** parties d'une grande statue

**ex** toutes les apparences extérieurs des constructions, elles sont en plus repartees en fonction du style architectural.

**flora** toutes les plantes qui ne sont pas des containes (par exemple les arbres)

**furn** les objets de decoration (lit, table...)

**in** tous les elements intérieurs mur, escalier couloir, eux aussi répartis en fonction de leur style architectural

**terrain** des elements supplémentaires pour les extérieurs, des ponts, des rochers

**VFX** les effets visuels comme les sorts. Ceux-ci ne sont d'aucun usage.

Tous les autres elements ne rentrent pas dans ces categories (indicateur du nord NorthMarker ou le volcan). Si vous n'avez d'idée précise de ce que vous voulez construire ou si vous cherchez un objet spécifique, déplacer simplement les objets vers la fenêtre de rendu. Elle doit être active sur la cellule que vous êtes en train de construire. Si ce n'est pas le cas vous pouvez la rendre active en double cliquant sur la ligne de votre cellule dans la fenêtre 'Cell View'.

N'oubliez pas alors de toujours effacer ce que vous venez d'ajouter. Sauf si c'est vraiment voulu de votre part essayez de conserver une cohérence dans le style de votre intérieur. Ne faites, par exemple, pas cohabiter un style Daedric avec le style impérial.

Lors de la construction de la structure de base, il faut faire particulièrement attention à ne pas laisser d'espace ou de trous entre les éléments de la structure. Cela peut prendre un peu de temps mais vous n'aurez pas alors à

reprendre totalement l'aménagement intérieur si vous réalisez plus tard avoir fait une erreur dans la structure de base.

Cela ne présente pas d'intérêt particulier pour la création d'un plug normal, mais tous les éléments static ont des paramètres spécifiques de réglages. On y accède en double cliquant sur la référence du static dans la fenêtre de Rendu.

La fenêtre est classique.

**ID** est l'identité de l'objet, son type.

**Add Art File** : permet de spécifier le fichier qui contient la représentation visuelle de l'objet, le fichier nif – NetImmerseFile

**Save** ferme la fenêtre en prenant en compte les modifications apportées

**Cancel** ferme la fenêtre sans prendre en compte les modifications apportées.

Il peut arriver que l'objet que vous insérez dans une cellule soit trop grand ou trop petit par rapport au reste de votre univers. Il est alors possible de jouer sur l'échelle ( 3D Scale) du modèle, une valeur inférieure à 1 fera un objet plus petit alors qu'une valeur supérieure fera un objet plus grand.

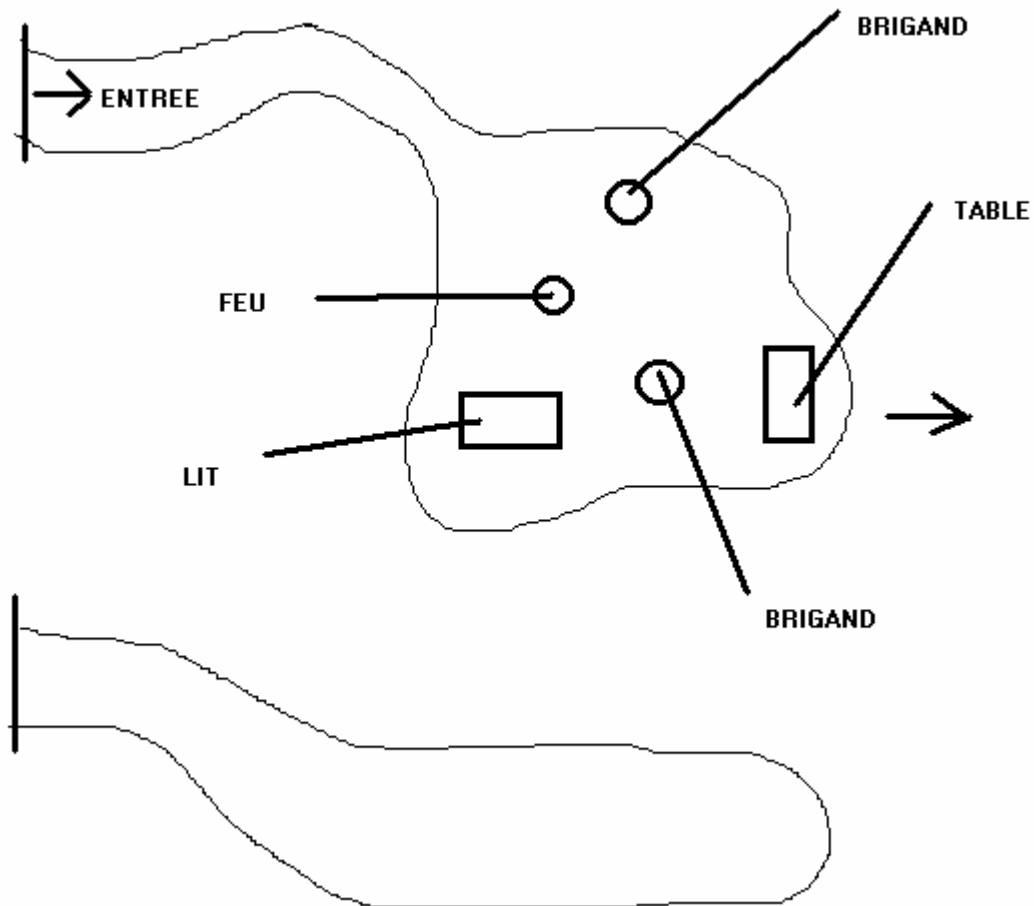
Après avoir définis la structure de la cellule que vous venez de créer, c'est-à-dire les murs, plafonds cadre de porte etc... il faut la peupler.

#### 7.4 La construction de la cellule

La première chose à connaître, ici encore, est l'objectif que l'on désire atteindre. Quelle doit être l'apparence de la cellule que l'on est en train de créer. Il faut avoir un plan même grossier de ce que l'on va créer. Autrement, on court à la catastrophe.

Afin de vous éviter la recherche de ce que vous allez créer et pour avoir un support d'explication, je vous propose le schéma suivant. Une grotte, comme vous pouvez le voir, il y a deux vues de l'objectif à atteindre :

- une vue de dessus
- une vue latérale pour prendre en compte le changement de niveau



Puisque je veux créer une grotte, je vais choisir les éléments de l'architecture de type MudCave, c'est-à-dire que je chercherais les statics qui s'appellent in\_mud\_XXX.

#### 7.4.1 Le début

Pour commencer, on va créer un repère visuel dans la cellule. Pour se simplifier la tâche, je vous conseille de toujours commencer la construction de vos cellules depuis l'origine du repère, X=0, Y=0 et Z=0. Il sera ainsi beaucoup plus simple de rajouter des éléments d'architecture autour de ce point.

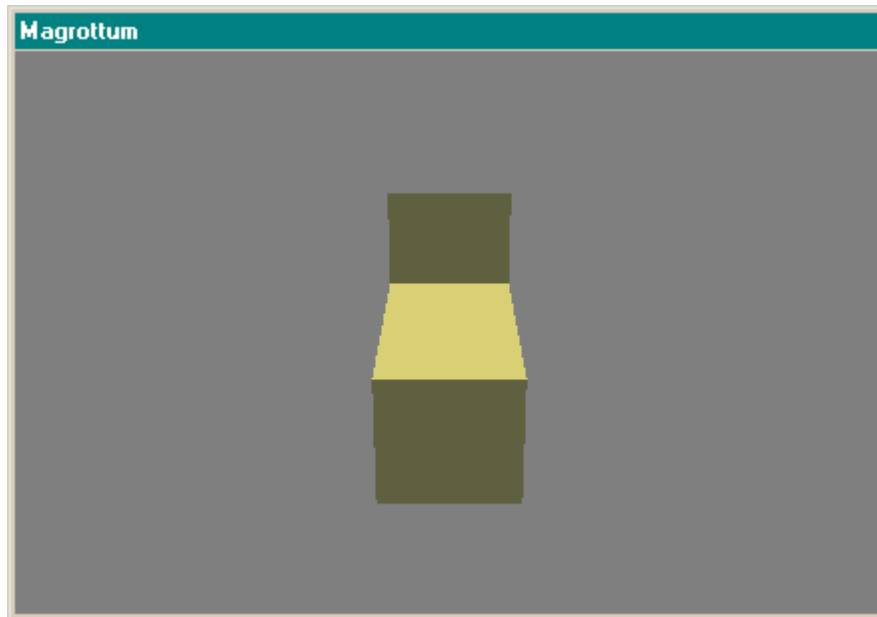
La première chose à ajouter est le repère du nord magnétique. Cela vous évitera de l'oublier plus tard.

La position (surtout, l'orientation du nord) sera à retoucher plus tard pour assurer la cohérence lorsque le PJ entre dans une cellule d'intérieur. Il faut éviter que le Nord se déplace de 90° en passant une porte.....

Vous trouverez le nord dans l'onglet static de la fenêtre des Objets (NorthMarker). Vous le placez par simple drag'n drop depuis l'onglet static dans la fenêtre de Rendu. A priori, il n'y a aucune raison que l'endroit où vous le placerez soit au centre du repère. Pour le placer exactement à l'endroit souhaité, on va modifier les données de la référence, en double-cliquant sur le NorthMarker.

Dans la fenêtre qui apparaît alors, mettez à 0 les trois coordonnées du marqueur, puis fermer la fenêtre.

A ce moment, il est possible que vous ne voyiez plus rien dans la fenêtre de Rendu. Là encore pas de panique, il est possible de recentrer la vue sur l'objet en double-cliquant sur sa référence dans la fenêtre Cell View. Voici ce que vous devriez alors voir :



Maintenant on peut ajouter les éléments de notre grotte par rapport à cet élément.

#### 7.4.2 *L'ajout des éléments de l'architecture*

Maintenant, on ajoute les éléments de l'architecture. La principale difficulté est l'ajustement des éléments les uns par rapport aux autres. Il est en effet très important de faire en sorte qu'il n'y ait pas d'espace visible en les statics.

Fort heureusement l'ensemble des statics produits par Bethesda ont des dimensions multiples de 16 (128, 256, 384, 512 et 1024 pour les plus fréquents). Cela doit permettre de mieux positionner les statics les uns par rapport aux autres en faisant de simple additions (aider vous de votre calculette si besoin), en effet les pivots des statics de Bethesda sont tous positionnés au centre des différents statics, sauf pour les escaliers (le pivot est à la base de l'escalier), dans les préférences de l'éditeur, sélectionnez un Grid Snap de 64, cela sera encore plus simple.

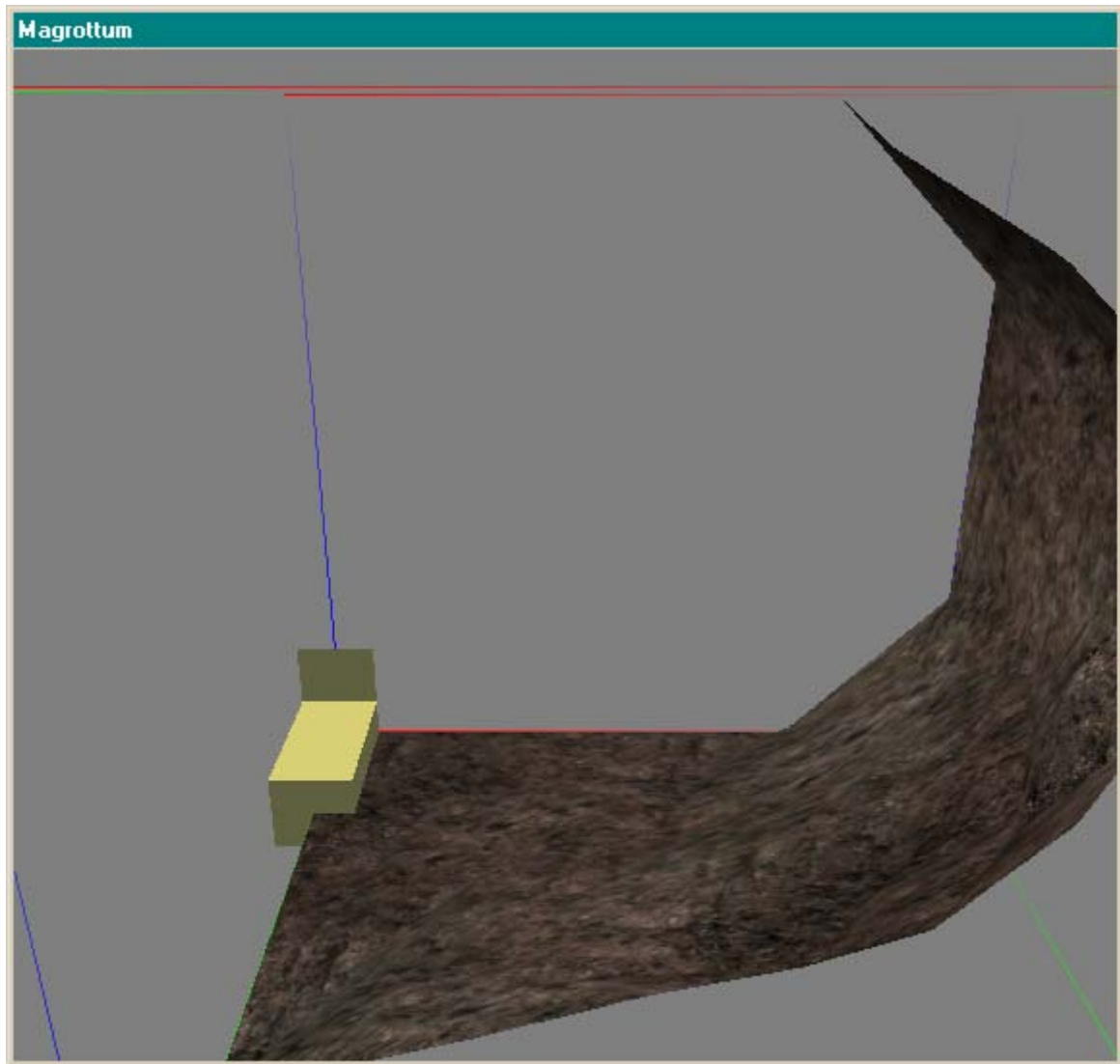
in_mudcave2_09	Static	-128, 960, -192 [0, 0, 0] (Magrottum)
----------------	--------	---------------------------------------

Dans la barre d'état de l'éditeur, vous avez de nombreuses informations relatives à l'élément actuellement sélectionné dans la fenêtre de rendu. Son ID, le type de l'élément, les coordonnées en X, Y, Z et les angles de rotation par rapport au repère de la cellule. Et enfin, la cellule dans laquelle on est en train de travailler.

<b>Objects: 2, Faces: 380, Lights: 0, NPCs: 0, Creatures: 0; [Sel Obj: 1, Fac: 358, Lite: 0]</b>
--

On peut également y connaître le nombre d'objets dans la cellule, le nombre de face, d'éclairage, de PNJ, de créatures...

On amène donc le premier élément de notre grotte, dans la fenêtre de rendu et on le place à l'origine du repère, il est alors positionné de la façon suivante :



#### 7.4.3 La construction de la cellule

On ajoute ensuite les éléments de la cellule les uns après les autres.

On peut positionner les éléments aussi bien avec la souris (clic-gauche et mouvement de la souris pour déplacer l'objet suivant les axes X et Y, enfoncer les touches X, Y ou Z permet de restreindre les mouvements à ces axes, pour faire une rotation par rapport aux différents axes, il faut enfoncer le bouton droit de la souris et se déplacer, attention les mouvements se font alors dans le repère propre de l'objet), qu'en spécifiant les données de la références dans la fenêtre des propriétés.

Lors de la première construction de cellule, vous allez éprouver certaines difficultés qui sont dues essentiellement à la méconnaissance que vous avez sur la signification des éléments que vous associez entre eux. Ne vous en faites pas, ce n'est pas grave. Avec le temps et de la patience, vous arriverez à connaître la signification des codes utilisés par Bethesda.

#### 7.4.4 Aspect general

Faites attention à l'aspect naturel de ce que vous construisez. Autant les angles droits ne sont pas choquants dans une construction artificielle, autant le regard est immédiatement attiré par eux dans une grotte par exemple. Veillez à la cohérence des cellules que vous construisez. A moins qu'il n'y ait une vraie raison, ne mélangez pas l'architecture Velothi avec l'architecture Daédrique par exemple...



Voici l'exemple de quelque chose de pas naturel dans une grotte. Lorsque vous construisez vos cellules prenez régulièrement de la « hauteur » par rapport à vos créations, changez de point de vue cela fera apparaître des défauts cachés.

#### 7.4.5 Remarques sur les performances

Vous l'avez sans doute constaté en jouant, le moteur 3D de MW n'est pas des plus performants. Apparemment tout est calculé pour chaque image, y compris les faces non visibles. En conséquence, essayez de limiter le nombre de face des cellules que vous construisez, 60 000 semble une bonne limite.

De toutes façons tester toujours vos créations pour voir si le moteur 3D n'est pas mis à genoux par votre imagination.

En travaillant la mise en place des éléments, ce qui peut être long, on obtient quelque chose comme ça.



Avant de peupler notre grotte, nous allons la rattacher à l'univers de MW afin de pouvoir la tester en grandeur réelle.

Pour pouvoir tester la cellule, il faut fournir un peu de lumière. On y place donc un feu de camp pour fournir un éclairage.

#### 7.4.6 Les lumières

Il existe deux types de sources lumineuses, celles qui ont une représentation 3D dans le jeu et celle qui n'en ont pas. Pour le test, cela n'a pas d'importance, choisissez simplement une source lumineuse puissante, cela montrera tous les défauts éventuels.

NB : le raccourci [F] pour ajuster la coordonnée en Z d'un objet de façon à ce qu'il soit au contact du sol (ou d'un support) ne marche pas pour les lumières. Pensez à changer de point de vue pour vous assurer qu'elle est bien au contact d'un support.

Pour faciliter l'aménagement de votre cellule, pensez à réduire le pas de la grille de déplacement.... (64 est trop gros, essayez d'autre valeur, jusqu'à pouvoir placer vos objets comme vous le voulez dans l'espace.

Puis il faut prévoir un point d'accès.

#### 7.4.7 Lier l'intérieur à l'extérieur

Tout d'abord, il faut trouver la cellule extérieure (ou intérieure) dans laquelle le joueur doit être téléporté lorsqu'il utilisera la porte.

Dans notre cas nous allons prendre une cellule extérieure, ce qui la rendra plus accessible pour les tests. Nous choisissons de poser l'entrée de notre caverne dans la colline au Sud-Ouest du SiltRider de Balmora.

On pose donc une porte extérieure de caverne dans cette colline. Et afin de ne pas prendre de mauvaises habitudes, on décore un peu l'entrée pour la rendre plus naturelle. On fait également attention à l'accessibilité pour de la porte pour tous les joueurs.

Les portes qui permettent de téléporter le joueur dans une autre cellule se trouvent dans l'onglet Door de la fenêtre des Objets. Il existe également des portes dans l'onglet static, mais celles-ci ne sont pas utilisables pour téléporter le PJ. En revanche, vous pouvez les utiliser pour limiter les accès dans votre donjon, pour empêcher l'accès à des zones non finies par exemple.

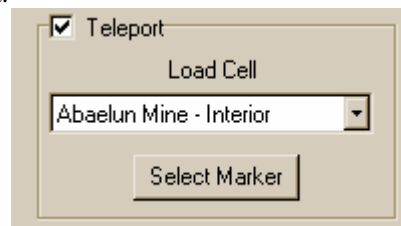
Pour placer une porte de téléportation, commencer par placer les pas de porte, en effet les portes présentes dans l'onglet door ne sont représentées que par leur battant.

Voici le placement du pas de porte du coté de la grotte :



Puis dans l'onglet Doors, vous trouvez un objet qui s'appelle 'doors\_cavern\_door10', et vous l'ajoutez à la place qu'il doit occuper. Faites attention à bien la positionner.

Une fois qu'elle est bien posée, double cliquez sur sa référence. Dans la fenêtre des propriétés de cette porte, vous pouvez cocher l'option teleport.



Il est maintenant possible de sélectionner la cellule de destination de cette téléportation.

Une fois que vous avez choisi la cellule de téléportation, il faut encore placer correctement le marqueur de téléportation. Cliquez sur Select Marker, l'éditeur charge alors la cellule de destination et place une flèche rose

devant l'écran. C'est le marqueur de téléportation, il est positionné par défaut à la coordonnées (0,0,0) avec l'angle (0,0,0). Il est possible de le déplacer par drag'n drop, de la même façon que les autres objets.

Il ne reste plus qu'à le positionner au bon endroit de l'autre côté. C'est-à-dire au niveau de la porte qui téléporte à l'intérieur de la grotte.

Petite idée : si votre PC est un peu lent, je vous conseille de repérer les coordonnées de la porte de l'autre côté (dans la cellule de destination) notez les et lorsque vous voudrez placer le repère de téléportation, vous n'aurez qu'à changer ses coordonnées dans la fenêtre des propriétés, et à ajuster son orientation.

Maintenant, il est possible de sortir de la grotte. Pour que l'on puisse y rentrer, il faut faire la même chose pour la zone d'entrée...

#### 7.4.8 On aurait pu faire plus simple

Commencer la construction des cellules intérieures par une grotte n'est pas vraiment ce qu'il y a de plus simple, il faut faire attention à l'aspect naturel des choses.... L'absence d'angle droit est frustrante et pénible à gérer pour les déplacements des objets. On aurait pu commencer par faire une chambre carrée avec un lit.

C'est vrai. Mais peut être faut il goûter la difficulté pour apprécier les outils comme GenMod ou mGen qui génère automatiquement des cellules internes ou externes.

Rassurez vous, il existe par ailleurs la possibilité de faire un copier coller depuis des cellules déjà existantes pour récupérer un bout d'architecture qui vous plaît.

Pour faire ce copier coller et ce piratage honteux, il faut charger la cellule que vous désirez copier. Sélectionner la zone qui vous intéresse (clic gauche à l'extérieur puis on reste enfoncé pour faire apparaître un cadre blanc autour de ce qui nous intéresse, lorsque l'on relâche le bouton gauche, l'ensemble des objets sélectionnés apparaît avec leur boîte Rouge-Verte-Bleu), copier (Ctrl+C), vous chargez la cellule que vous construisez et vous collez (Ctrl+V). Voilà c'est fait, comme il est possible que la position ne soit pas exactement ce que vous vouliez, vous pouvez ajuster en déplaçant la sélection de la même façon qu'un objet unique (attention à ne pas cliquer en dehors de la sélection, vous perdriez alors toute la sélection).

#### **7.5 Peupler et équiper notre cellule**

Créer une architecture pour une cellule, c'est bien beau. Mais ce n'est pas encore suffisant pour que l'on puisse considérer que le travail est fini...

Il faut encore ajouter des décors, des PNJ ou des créatures pour peupler cette cellule. C'est un travail long et souvent minutieux. Pour ajouter des éléments dans la cellule on procède de la même façon que pour ajouter les murs : par drag'n drop depuis la fenêtre des objets.

Il y a quand même quelques précautions élémentaires à prendre :

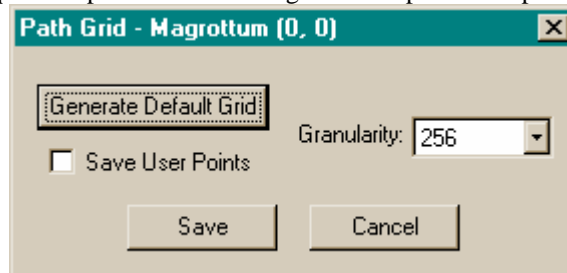
- si vous ajoutez un PNJ, il faut d'abord le créer (n'ajoutez JAMAIS un PNJ qui existe déjà dans le jeu)
- si vous voulez ajouter des coffres avec un contenu spécifique, créer d'abord un nouvel objet coffre, puis attribuez lui son contenu et finalement poser le dans la cellule. Si vous ne procédez pas de cette façon, et que vous utilisez un coffre existant dans le jeu. Vous risquez un GROS PROBLEME, les modifications de contenu que vous faites sur votre coffre (croyez vous) se répercutent sur tous les coffres du jeu qui utilisent la même ID..... imaginez qu'à cause d'une erreur de ce type, tous les coffres de MW contiennent l'épée surpuissante que vous donnez en récompense au joueur...il y aurait un problème dans la logique du jeu puisqu'il lui suffirait d'ouvrir n'importe quel coffre pour y trouver la dite épée. Faites donc TRES ATTENTION à ce genre de problème. (référez vous à la section un mod propre).
- Faites en sorte d'éviter les objets en lévitation. N'hésitez pas à utiliser la touche [F], qui fait tomber l'objet jusqu'à ce qu'il entre en contact avec une surface. Cette commande marche pour tous les éléments sauf pour les lumières.

#### **7.6 Path Grid – Edition et modification**

La grotte est maintenant peuplée avec des brigands. Il faut que ces PNJ sachent se déplacer à l'intérieur de la cellule. Le moteur de jeu de MW utilise un algorithme qui repose sur une grille de déplacement pour permettre aux PNJ/créatures de se déplacer. Il existe des cellules dans lesquelles il n'y a pas grille, ou bien le PNJ/créature peut être amené à sortir des chemins balisés, dans ce cas MW utilise un moteur assez rudimentaire de détection d'obstacle pour permettre aux PNJ/créatures de se déplacer malgré cela.

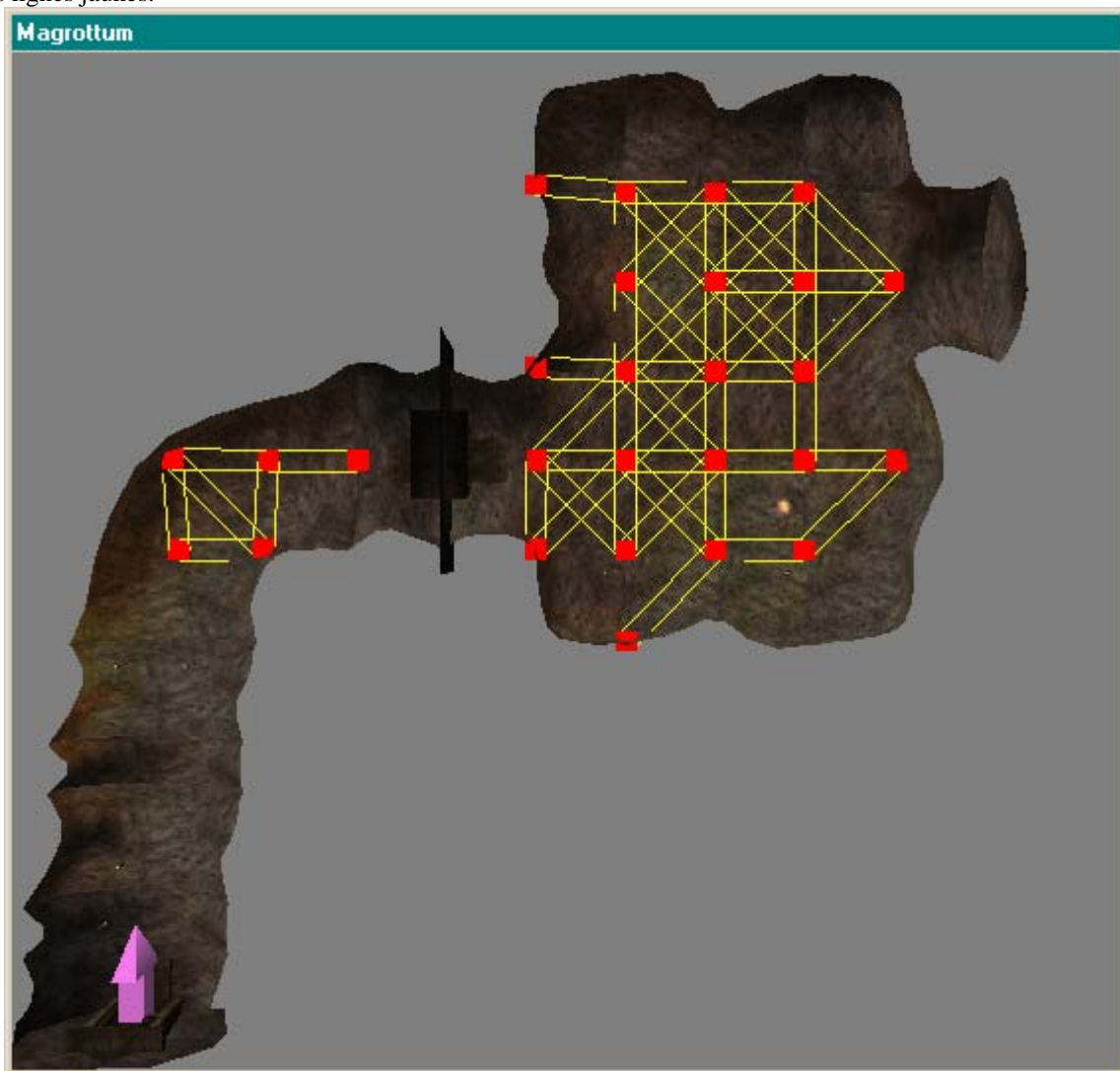
Dans la mesure où l'on désire créer une cellule peuplée de bandits, il est conseillé de créer une grille de déplacement pour votre cellule. Pour faire cela, cliquer sur le bouton Edit Path Grid dans la barre de menu, ou bien par le menu World/Edit PathGrid.

Il apparaît alors une fenêtre qui vous permet de créer la grille de déplacement pour votre cellule.



Il est possible de générer automatiquement une grille avec une granularité/précision définie. Plus ce chiffre est petit plus il y aura de points générés.

Lorsque vous appuyez sur Generat Default Grid, vous voyez apparaître dans votre cellule des points rouges et des lignes jaunes.



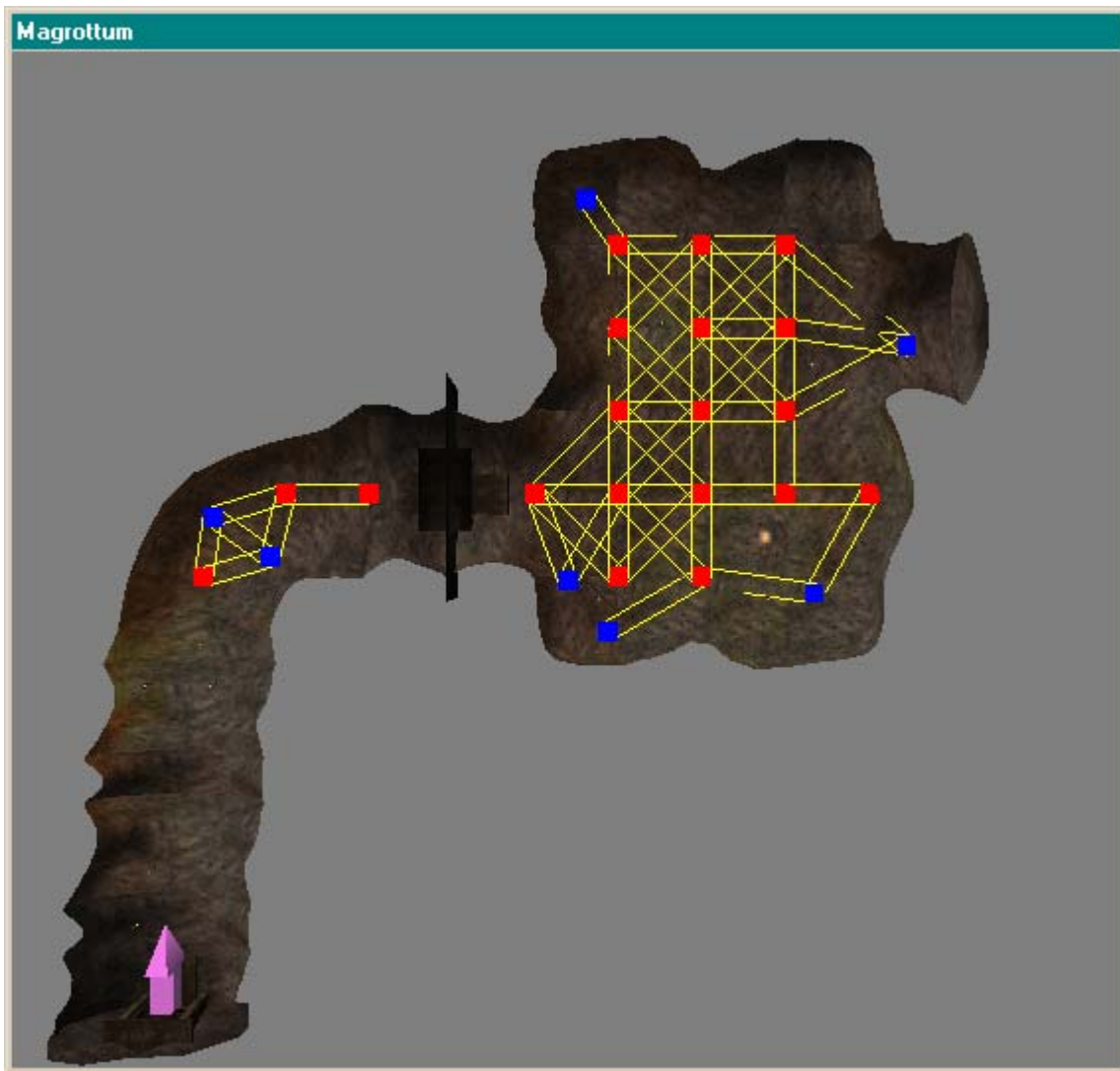
Les points rouges sont les extrémités des chemins possibles et les traits jaunes sont les chemins.

Lorsque l'on regarde la grille générée par défaut, on peut voir qu'il y a des points qui ne seront pas accessibles aux PNJ, cela risque de provoquer un comportement bizarre de celui-ci.

Vous pouvez les déplacer pour éviter les comportements indésirables.

Toutefois, si vous souhaitez que les changements que vous apportez à la grille soient pris en compte, il faut cocher l'option Save User Points.

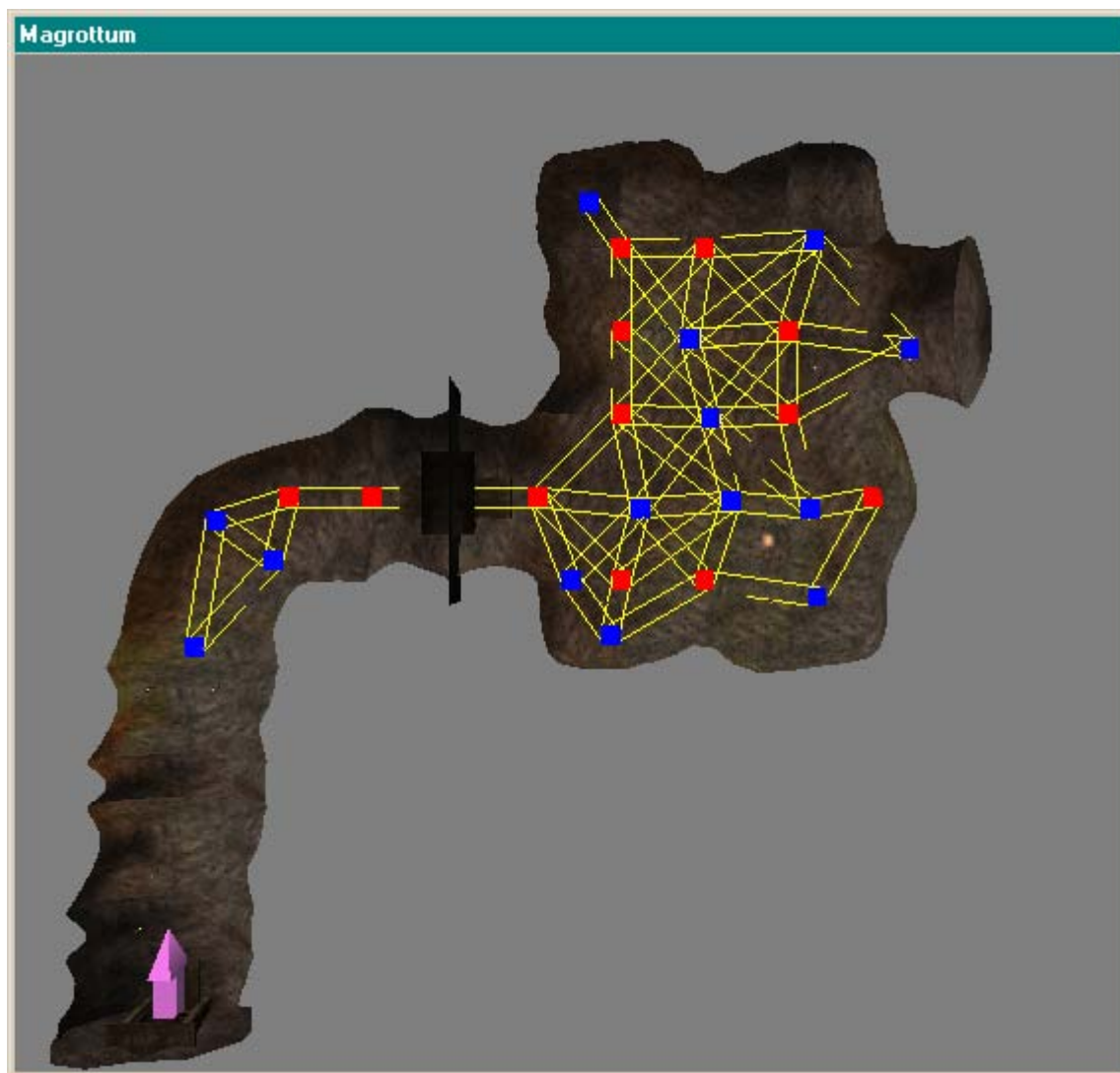
Pour déplacer des sommets de la grille, il suffit de faire un drag' drop de ceux-ci. Vous verrez que les chemins sont immédiatement modifiés.



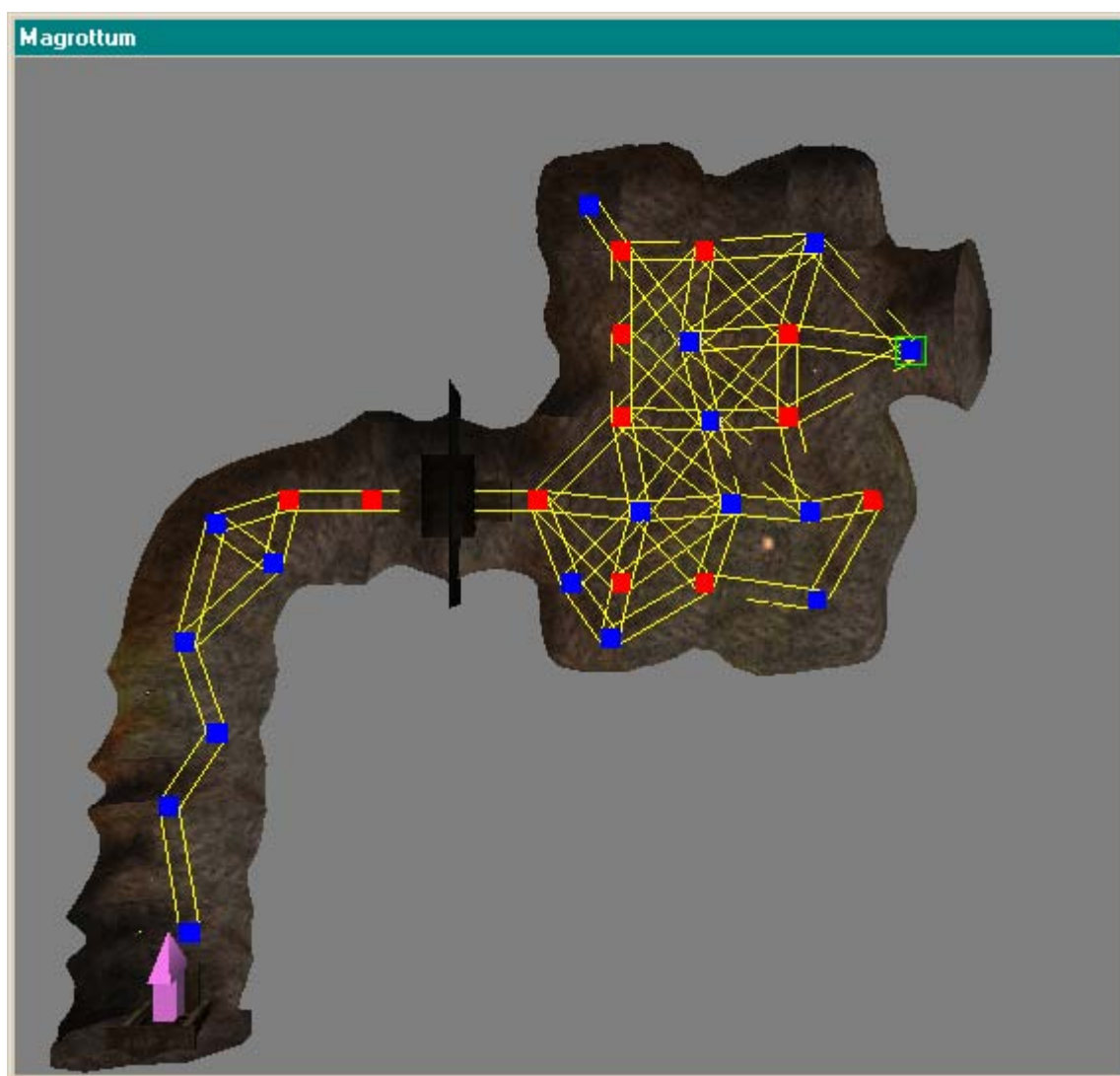
Les points que vous avez déplacés apparaissent en bleu sur la grille. Pour supprimer des sommets gênants, il suffit de les sélectionner et d'appuyer sur la touche [SUPPR] du clavier. De cette façon vous interdisez aux PNJ d'aller à certains endroits. Cela peut être utile pour les forcer à contourner des objets que vous avez ajoutés.

En regardant la grille actuelle pour notre cellule, on se rend compte qu'il y a un petit problème. En effet, il y a deux grilles séparées qu'aucun chemin ne joint. Les PNJ ne pourront pas traverser la porte (qui est pourtant ouverte)...

Il est possible de créer des chemins entre deux points. Sélectionner l'origine, puis en appuyant sur la touche [CTRL] cliquez sur la destination du chemin que vous voulez faire. Ajuster les points de la grille pour lui enlever sa rigueur et ajouter un peu de naturel dans la répartition des sommets.



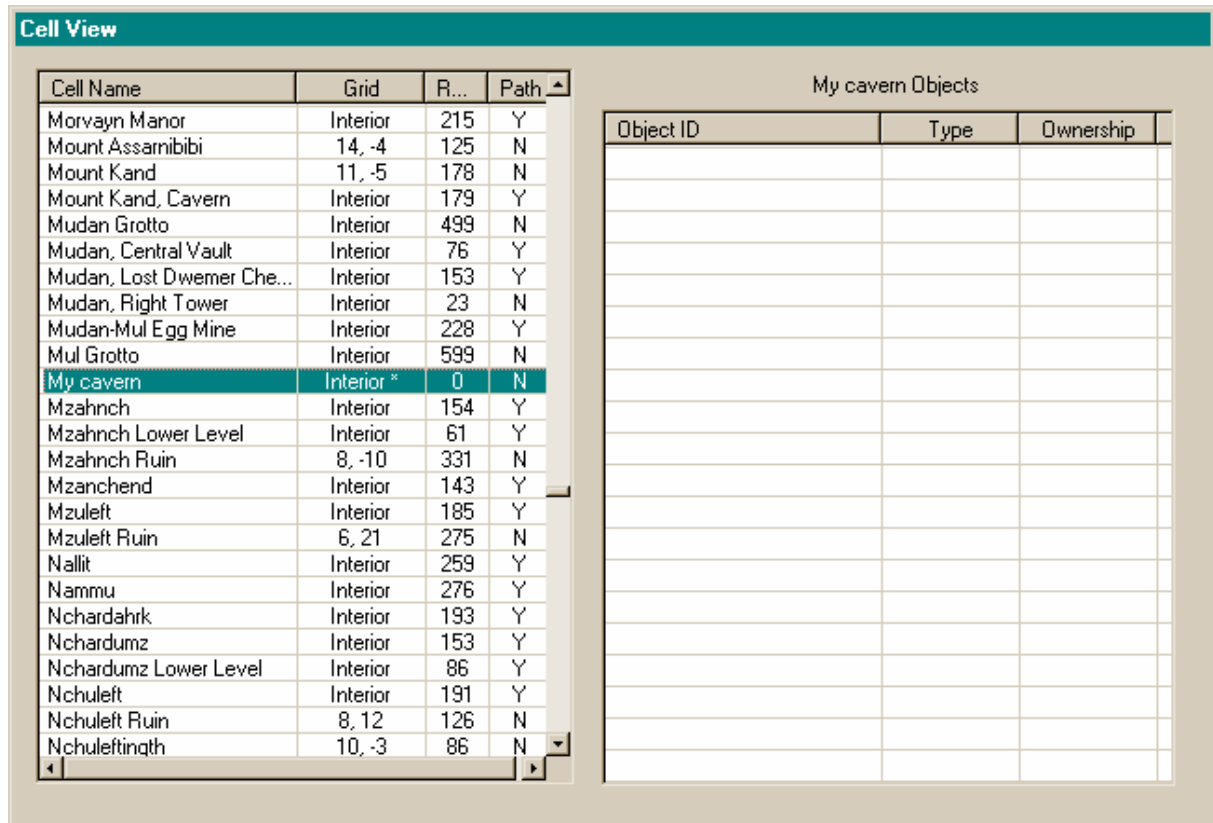
Il y a encore des endroits de la cellule qui ne sont pas accessibles. Il est possible de créer des points. Pour cela il suffit de faire un clic droit à l'endroit où vous désirez ajouter un sommet. Puis connecter le au reste de la grille.



#### 7.7 Connaître le contenu de la cellule

Pour connaître le contenu de la cellule, reporter vous à la fenêtre Cell View. Elle se compose de deux listes, celle de gauche contient la liste de toutes les cellules du jeu. Celle de droite, contient la liste du contenu de la cellule qui est sélectionnée dans la colonne de gauche.

Vous remarquerez les astérisques à côté de certains éléments des listes, ces astérisques vous informent que les éléments ainsi marqués ont été modifiés par vous, et que les modifications seront prises en compte dans le mod.



En double cliquant sur un élément de la colonne de droite, vous chargerez automatiquement la cellule concernée (si ce n'est pas déjà fait) et vous centrerez la vue de la fenêtre de Rendu sur cet objet.

Les touches T et C vous permettent de basculer respectivement à une vue de dessus et une vue isométrique (centrée sur l'objet sélectionné).

### 7.8 Inonder une cellule d'intérieur

Il est possible d'inonder une cellule d'intérieur, pour cela sélectionner l'item Interior Cell du menu, World. Cela fera apparaître la fenêtre des cellules d'intérieur, dans le menu déroulant sélectionnez le nom de la cellule concernées par la montée des eaux.

Vous pouvez maintenant modifier tous les paramètres génériques de la cellule, éclairage, brouillard etc.

Cochez l'option Has Water. Vous pouvez maintenant choisir la hauteur de l'eau. Elle est donnée en relatif par rapport à  $z=0$  (c'est-à-dire que si vous voulez avoir un niveau d'eau supérieur à  $z = 0$  il faut rentrer un nombre positif et dans le cas contraire, un nombre négatif)

NB : le niveau d'eau est valable pour toute la cellule, il n'est pas possible de créer des niveaux d'eau différents dans une même cellule.

NB : pensez à placer vos objets avant d'inonder la cellule, l'usage de la touche [F] poserait tous les objets sur l'eau...

## 7.9 Remarques pour les créateurs de static

Vous avez pu le constater, les choix de Bethesda pour les volumes englobant leur static sont loin d'être idiots, ils nous simplifient grandement la tâche. Pour les nombreux moddeurs qui vont désirer créer leur propre style architectural (qu'Azura les bénisse), essayer de faire en sorte que les dimensions de vos statics soient des multiples de 16 (et Azura n'aura de cesse de vous remercier encore et encore et encore).

### 7.10 Un outil pour accélérer le développement des cellules : GenMod

Vous pourrez trouver sur le Net, grâce à Jim Adam, un outil qui permet de générer des cellules d'intérieur à vitesse grand V et sans avoir à vous préoccuper de l'ajustement des static entre eux. Cet outil utilise la connaissance des dimensions des différents static pour les agencer entre eux de façon simple.

Dans sa version actuelle il permet même de générer des cellules d'intérieur de façon aléatoire pour tous les types de pièces excepté les grottes. Mais l'auteur travaille sur un générateur aléatoire de cellules pour les grottes aussi.

Comme je trouve cet outil génial, je vous fait partager ce que j'ai compris de son fonctionnement. Comme le mode d'emploi fournit est en anglais, je traduit les explications qu'il y donne.

#### *7.10.1 Où le trouver?*

Cet utilitaire est disponible dans toutes les bonnes échopes MW. C'est un utilitaire, vous le trouverez donc dans les rubriques utilitaires.

Il est également téléchargeable directement chez son créateur.

## 8 Créer des cellules d'extérieur

Les extérieurs englobent la totalité des paysages du monde. Le monde est divisé en cellules, ce sont des carré d'environ un kilomètre de coté (385 pieds). Elles sont repérées par un système de coordonnées dont l'origine se situe des les Ashlands au Sud-Est de Caldera.

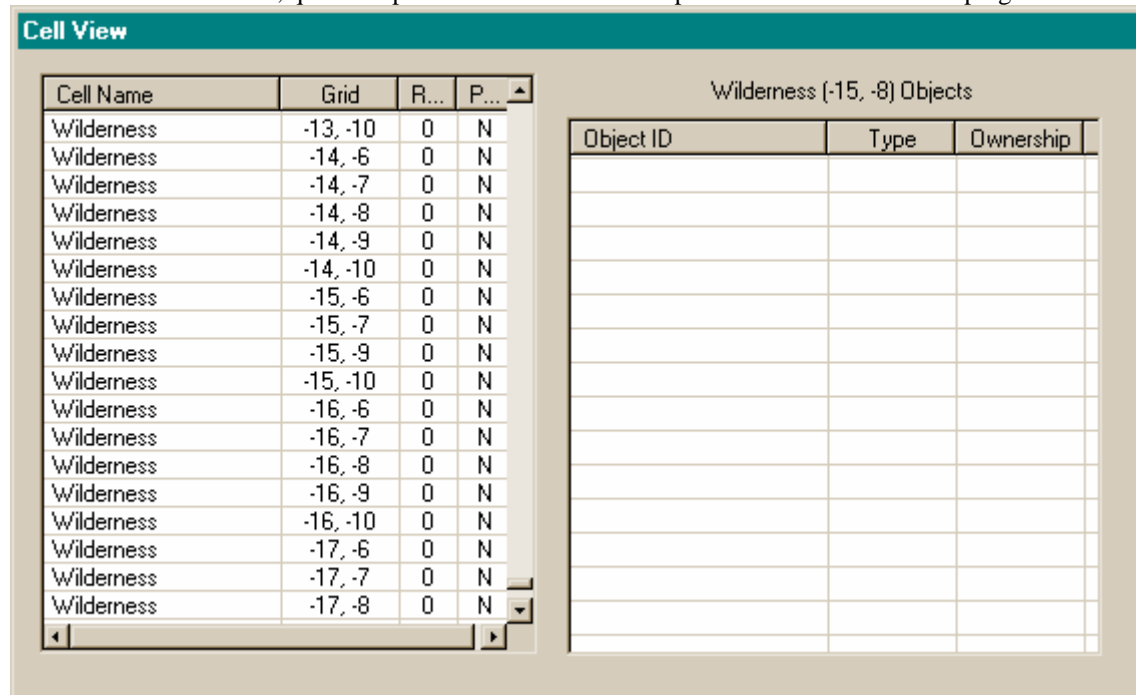
Il est important de savoir que lorsque l'on charge une cellule extérieure, toutes les cellules dans son voisinage sont également chargées. L'éditeur traite des cellules d'extérieur un unique paysage, c'est à dire, que si vous déplacez le point de vue vers une bordure, l'éditeur charge automatiquement la cellule avoisinante (mais seulement si l'option "Allow Render Winow Cell Loads" dans les préférences a été sélectionnée.

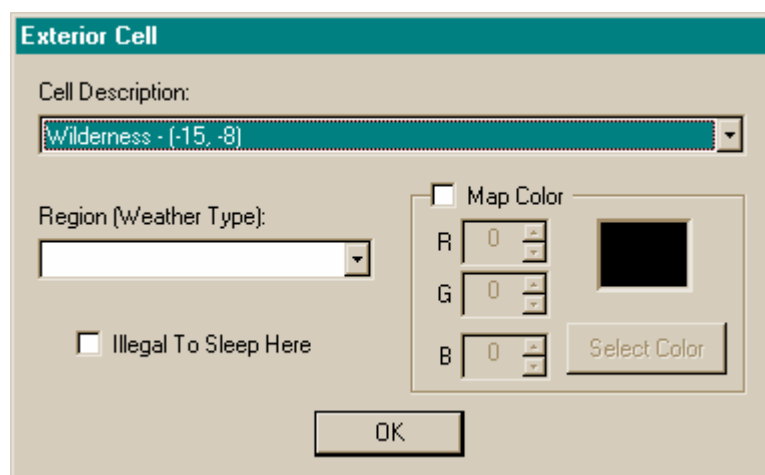
Beaucoup de choses fonctionnent de la même façon dans la création d'extérieur que dans la création d'intérieur (l'ajout d'un pnj, d'un objet, de créatures ou de portes..). Il est bien sûr possible de modifier les cellules extérieures existantes.

Nous allons nous concentrer sur la création de nouvelles cellules d'extérieur, ce qu'il vous apprendrez ici sera également valable pour la modification d'un existant.

## 8.1 Former le paysage

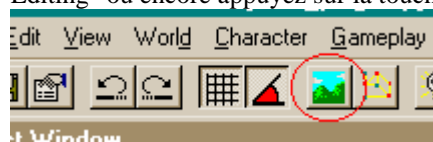
Avant de commencer, il faut trouver une cellule à éditer. Afin d'éviter les conflits avec l'existant, je vous recommande de choisir une cellule hors de l'île actuelle de Vvardenfell. Pour cela, vous pouvez choisir une cellule sur le bord de la carte (soit  $x=23$ , soit  $x=-18$ , soit  $y=27$ , soit  $y=-12$ , ces valeurs peuvent varier, je ne les donne qu'à titre indicatif. Lorsque vous quittez, la zone des cellules existantes, de nouvelles cellules "Wilderness" sont créées, que vous pouvez ensuite renommer pour les utiliser dans votre plug.



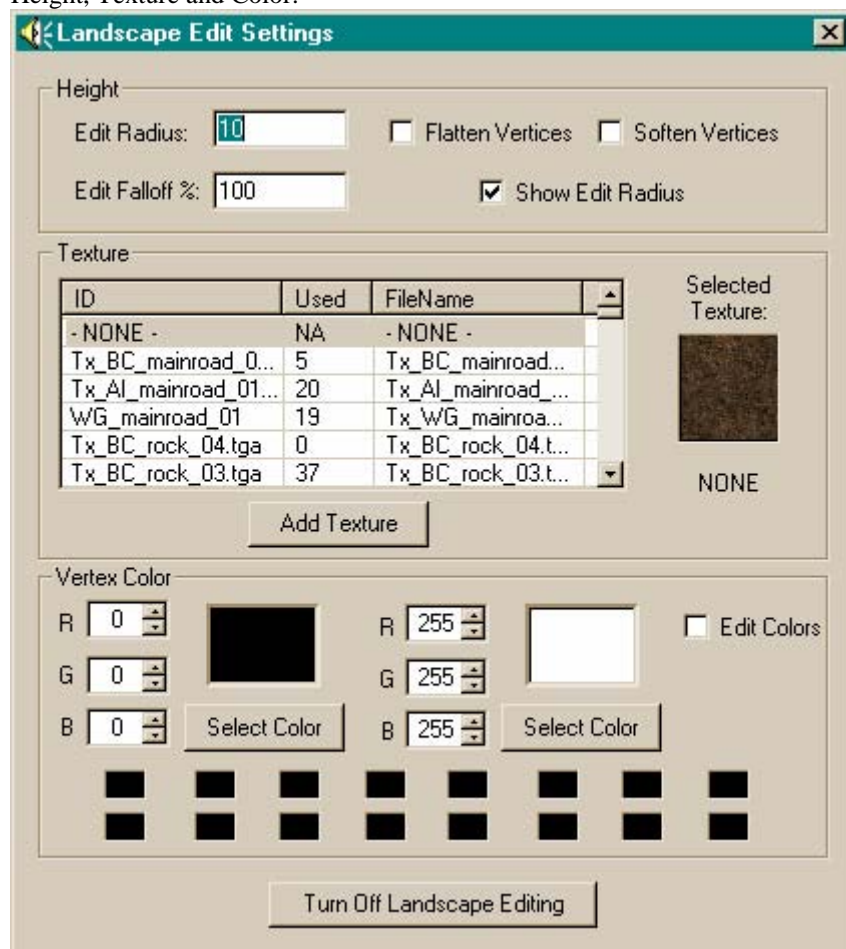


Cela peut s'avérer long de créer des cellules 'Wilderness', mais de cette façon vous ne risquez pas de faire des bêtises en éditant une cellule de MW...

La première étape est la création de la structure générale du paysage (les montagnes, les plaines et les rivières ...). Pour faire cela, cliquer sur le bouton "Edit Landscape" dans la barre d'outils ou sélectionnez "World/Landscape Editing" ou encore appuyez sur la touche [H].



Vous voyez maintenant la fenêtre "Landscape Edit Settings", qui vous permet de modifier différents paramètres: Height, Texture and Color.



Il vaut éditer l'altitude 'Height' en mode fil de fer, même si cela impose de basculer régulièrement en mode texturé pour visualiser les zones qui sont sous l'eau et celles qui ne le sont pas. En revanche, les couleurs et les textures du paysage doivent être appliqués en mode texture (sinon, on ne voit rien)

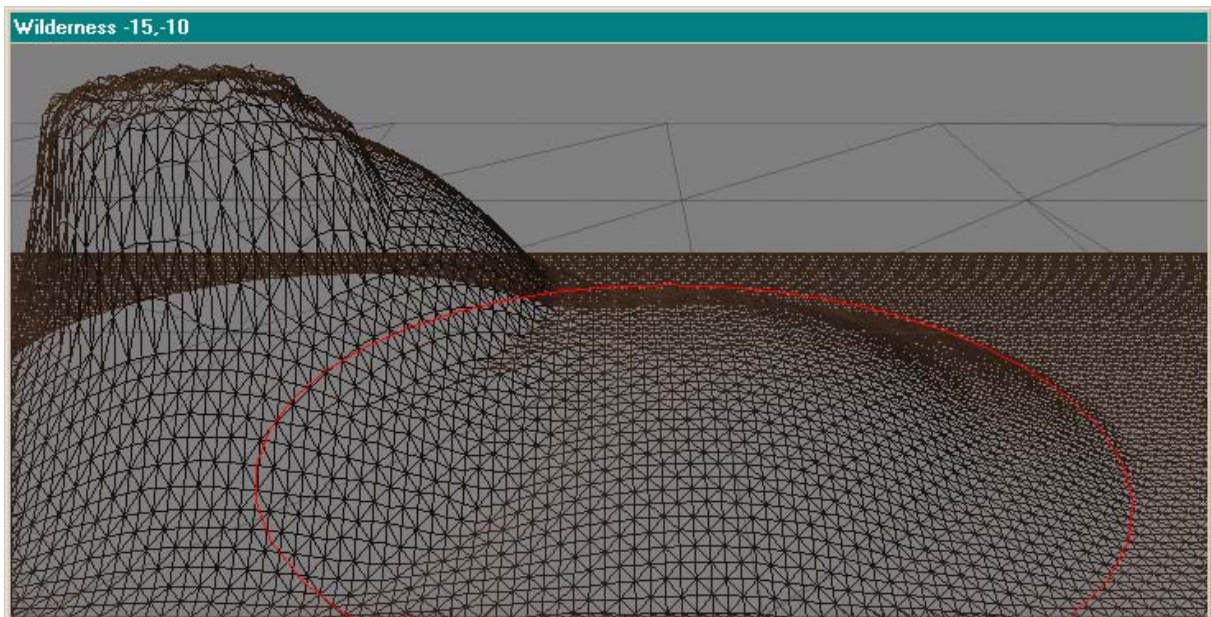
#### 8.1.1 L'altitude - Height

L'altitude d'un paysage détermine son allure générale (colline, vallée etc...). Les trois dimensions d'un paysage sont séparées en trois unités élémentaires.

L'eau est au niveau 0, et l'altitude des cellules (sauf celles qui sont éditées) est à -2000. Tout travail d'édition de cellule extérieure doit donc commencer par une phase où l'on fait sortir les terres de l'eau.

All three dimensions of the landscape are separated into single units. The water height is at 0.

L'augmentation et la diminution de l'altitude des terres fonctionnent de la même façon. L'outil que l'on utilise est un cercle rouge. Lorsque l'on maintient le bouton gauche enfoncé et que l'on déplace la souris vers le haut on augmente l'altitude de la zone qui est à l'intérieur du cercle. Lorsque l'on déplace la souris vers le bas, on diminue l'altitude.



**"Edit Radius"** vous permet de modifier le rayon du cercle rouge.

**"Edit Falloff %"** indique l'arrondi de l'édition du paysage. Cela permet de paramétrer ce qui se passe en bordure du cercle. Plus la valeur est faible, plus les bords de la zone élevée/abaissée seront abrupts.

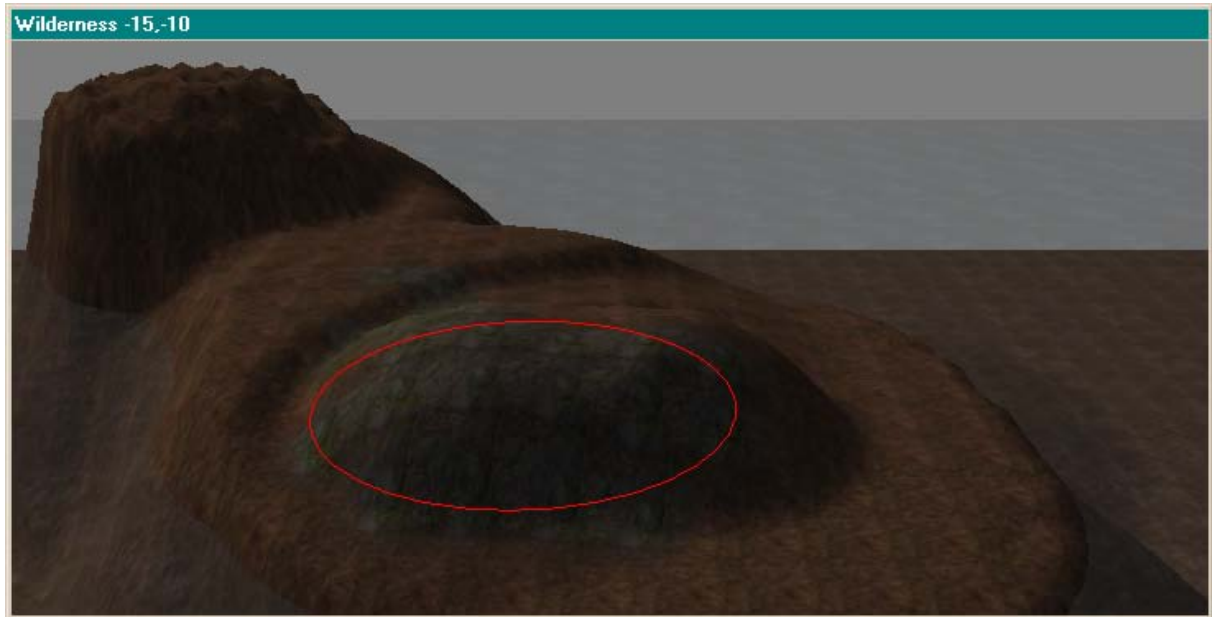
**"Flatten Vertices"** permet de ramener la zone à l'intérieur du cercle à sa valeur initiale (-2000).

**"Soften vertices"** vous permet de réduire les ruptures d'édition et d'adoucir (d'éroder) les sommets de collines ou les fonds des vallées.

**"Show Edit Radius"** rend (in)visible le cercle rouge dans la zone de rendu.

#### 8.1.2 Texture

Aussi longtemps que l'édition des couleurs ("Edit Colors") n'est pas activée, vous pouvez ajouter des textures au paysage par un clic droit dans la zone de rendu. Vous pouvez choisir la texture que vous attribuez dans la liste à gauche de la fenêtre d'édition des paysages.

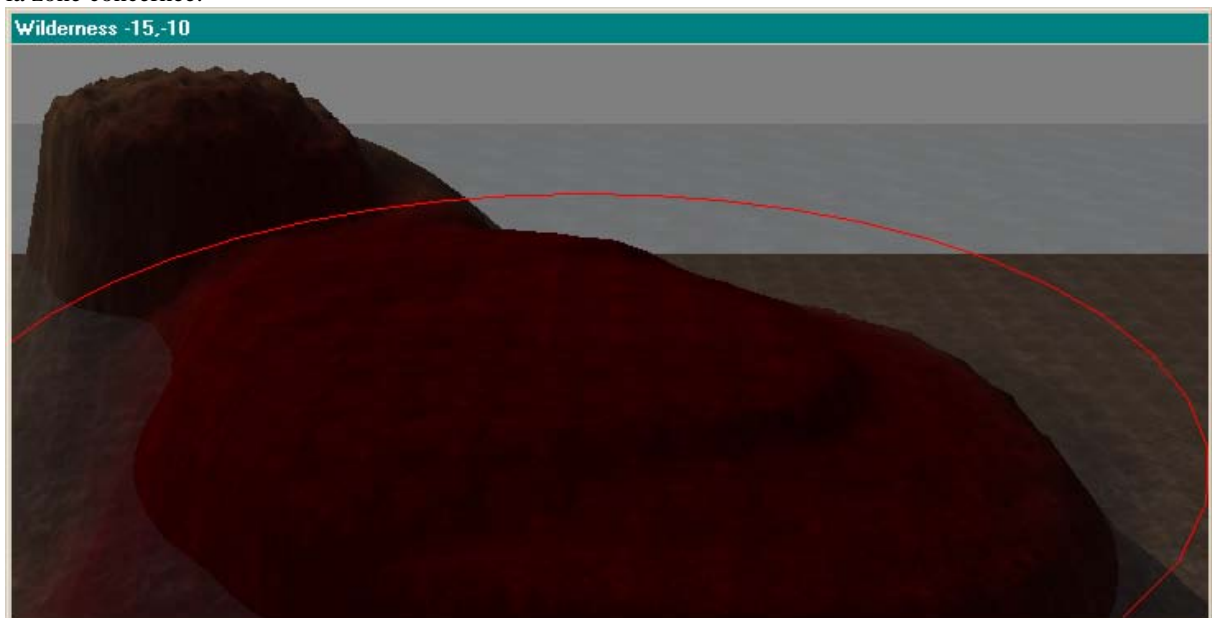


"Add Texture" vous permet d'ajouter des textures de votre propre production (ce doit être des fichiers tga).

"Selected Texture" cette zone vous donne un aperçu de la texture que vous êtes en train d'appliquer, mais c'est souvent difficile de se faire une idée de ce que cela donne.

#### 8.1.3 Les couleurs - Vertex Color

En plus des textures, vous pouvez attribuer des couleurs à vos paysages. Vous pouvez choisir deux couleurs qui sont choisies soit dans une liste prédéfinie(en appuyant sur "Select Color") soit par vous en donnant les valeurs RGB de votre couleur. Pour ajouter les couleurs sur vos créations, il faut activer "Edit Colors" puis cliquer à droite ou à gauche dans la zone de rendu, la couleur associée avec le clic droit/gauche est alors ajoutée au paysage. Pour retirer une couleur attribuée, il faut sélectionner la couleur blanche (255,255,255) et l'attribuer à la zone concernée.



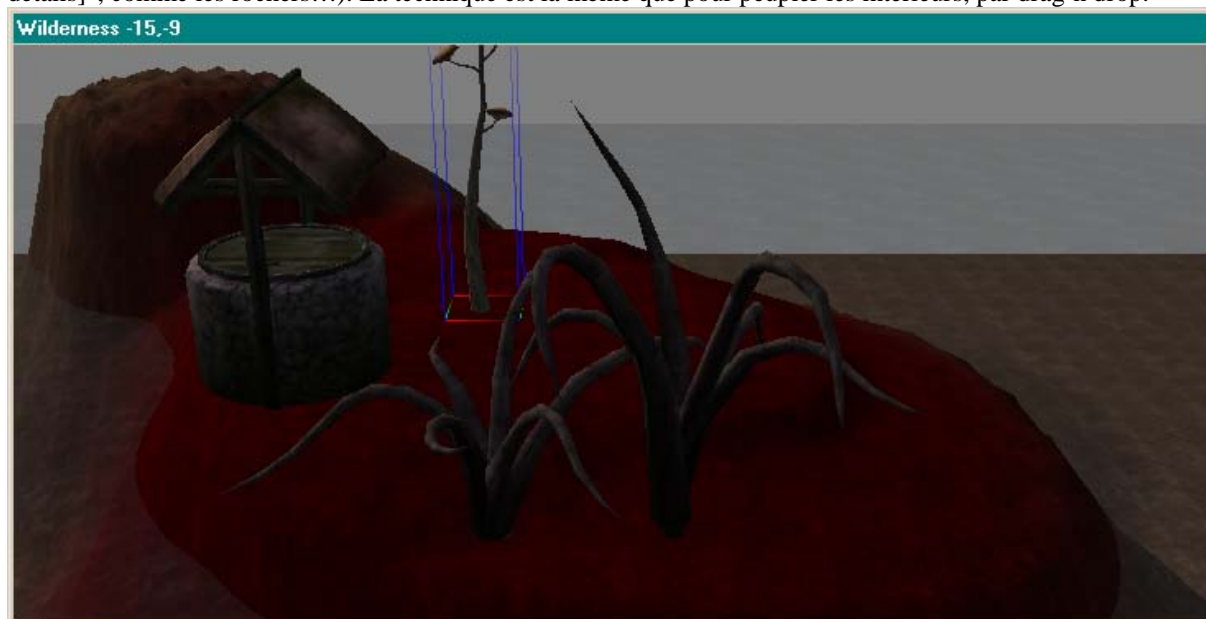
(on ajoute un beau rouge sang par terre....)

Attribuer une couleur au paysage est une technique qui peut être utile pour marquer des chemins ou des zones de terres brûlées (si vous ne disposez pas de la texture adéquate). Il va de soit que les textures sont plus jolies et on a un meilleur effet pour le rendu de votre monde.

"Turn Off Landscape Editing" permet de fermer la fenêtre d'édition de paysage.

#### 8.2 Ajouter des objets, des creatures, des lumières ou des pnj.

Comme nous l'avons vu plus haut, la méthode à suivre pour ajouter la vie dans votre cellule est la même que pour la création de cellule d'intérieur. Vous disposez simplement de plus d'objet à ajouter. Les plantes (flora\_[et plus de détails]), les objets d'extérieur ("ex\_[plus de détails]"), et les objets de terrain (named "terrain\_[plus de détails]"), comme les rochers...). La technique est la même que pour peupler les intérieurs, par drag'n drop.




### 8.3 Définir le climat de la cellule

Pour rendre plus "réelle" la cellule que vous venez de créer, il faut lui attribuer un climat. Vous pouvez lui attribuer l'un des climats prédéfinis ou bien créer un climat personnalisé pour votre cellule.

#### 8.3.1 Créer un climat personnalisé

Pour créer un climat spécifique, choisissez "World/Regions..." afin de faire apparaître la fenêtre "Regions".

Regions																																																																							
Region ID:	Bitter Coast Region	Name: Bitter Coast Region																																																																					
New	Delete	Change ID																																																																					
Sleep Creature:	ex_bittercoast_sleep																																																																						
Weather Chances																																																																							
Clear:	10	Rain:																																																																					
Cloudy:	60	Thunder:																																																																					
Foggy:	10	Ash:																																																																					
Overcast:	0	Blight:																																																																					
Region Paint																																																																							
Region Map Color																																																																							
R	34																																																																						
G	39																																																																						
B	255																																																																						
<button>Select Color</button>																																																																							
<table border="1"> <thead> <tr> <th>Chance</th> <th>Priority</th> <th>Sound ID</th> </tr> </thead> <tbody> <tr><td>4</td><td>1</td><td>wind calm1</td></tr> <tr><td>4</td><td>2</td><td>wind calm2</td></tr> <tr><td>4</td><td>3</td><td>wind calm3</td></tr> <tr><td>4</td><td>4</td><td>wind calm4</td></tr> <tr><td>4</td><td>5</td><td>wind calm5</td></tr> <tr><td>5</td><td>6</td><td>wind trees1</td></tr> <tr><td>5</td><td>7</td><td>wind trees2</td></tr> <tr><td>5</td><td>8</td><td>wind trees3</td></tr> <tr><td>5</td><td>9</td><td>wind trees4</td></tr> <tr><td>5</td><td>10</td><td>wind trees5</td></tr> <tr><td>5</td><td>11</td><td>wind trees6</td></tr> <tr><td>5</td><td>12</td><td>wind trees7</td></tr> <tr><td> </td><td> </td><td> </td></tr> <tr><td> </td><td> </td><td> </td></tr> <tr><td> </td><td> </td><td> </td></tr> <tr><td> </td><td> </td><td> </td></tr> <tr><td> </td><td> </td><td> </td></tr> <tr><td> </td><td> </td><td> </td></tr> <tr><td> </td><td> </td><td> </td></tr> <tr><td> </td><td> </td><td> </td></tr> <tr><td> </td><td> </td><td> </td></tr> <tr><td> </td><td> </td><td> </td></tr> </tbody> </table>			Chance	Priority	Sound ID	4	1	wind calm1	4	2	wind calm2	4	3	wind calm3	4	4	wind calm4	4	5	wind calm5	5	6	wind trees1	5	7	wind trees2	5	8	wind trees3	5	9	wind trees4	5	10	wind trees5	5	11	wind trees6	5	12	wind trees7																														
Chance	Priority	Sound ID																																																																					
4	1	wind calm1																																																																					
4	2	wind calm2																																																																					
4	3	wind calm3																																																																					
4	4	wind calm4																																																																					
4	5	wind calm5																																																																					
5	6	wind trees1																																																																					
5	7	wind trees2																																																																					
5	8	wind trees3																																																																					
5	9	wind trees4																																																																					
5	10	wind trees5																																																																					
5	11	wind trees6																																																																					
5	12	wind trees7																																																																					
<button>OK</button>																																																																							

**New** vous permet de créer une nouvelle région. La principale caractéristique d'une cellule qui dépend de la région à laquelle elle appartient est son climat. Toutes les cellules dans une région ont le même climat.

**Delete** efface la région actuellement sélectionnée.

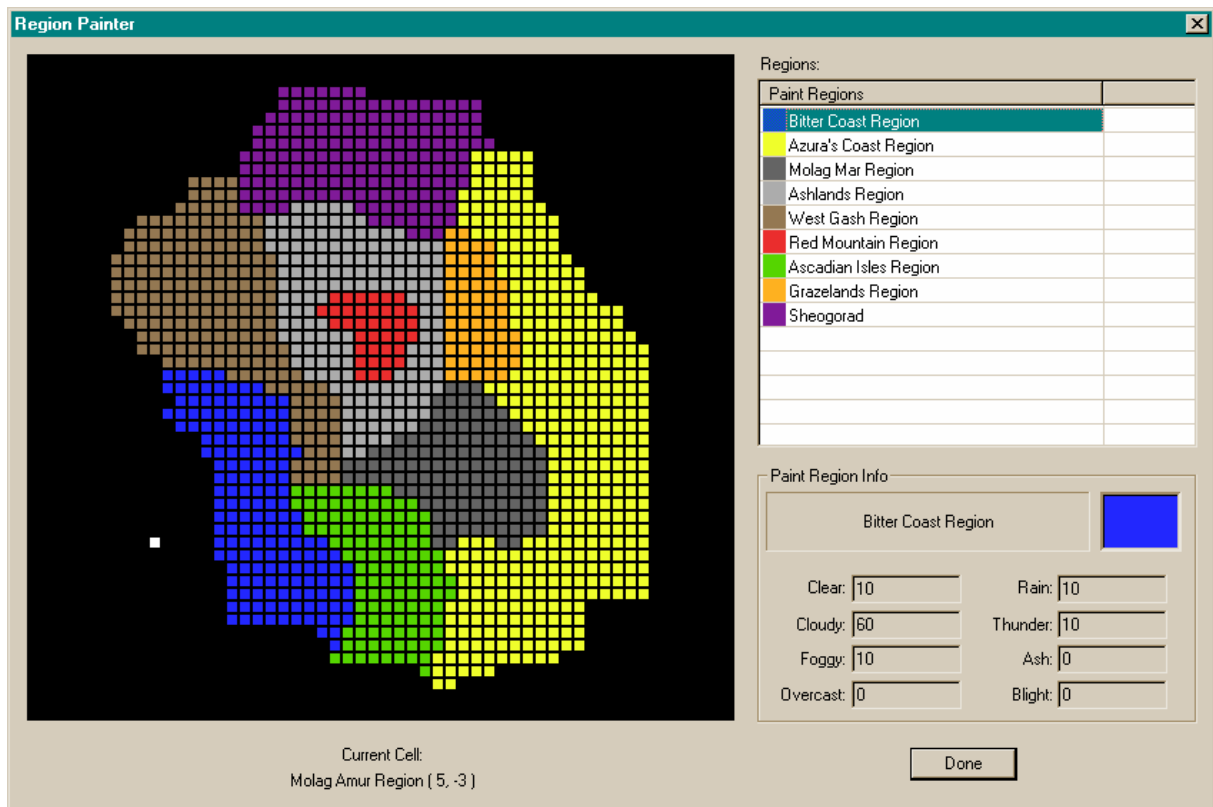
**Change ID** vous permet de modifier l'ID d'une region existante.

**Name** C'est le nom de la region. A ne pas confondre avec les noms des cellules. Le nom de la région n'apparaît jamais dans le jeu.

**Sleep Creature** Ici, vous spécifier une creature de niveau qui attaque le joueur lorsqu'il dort.

**Weather Chances** vous définissez les poids de chacun des 8 climats élémentaires. Il s'agit de pourcentage, la somme des valeurs que vous entrez doit être égale à 100.

**Region Paint** fait apparaître la fenêtre "Region Paint". Cela vous montre toutes les cellules existantes dans le monde, colorée d'une certaine couleur. Cette couleur est la couleur associée à la région dont elles dépendent. On peut choisir une région dans la liste de droite et attribuer à la nouvelle cellule la couleur de la région dans laquelle on désire l'insérer.



**Region Map Color** indique la region associée à la couleur sélectionnée dans la fenêtre "Region Paint". Comme d'habitude, vous pouvez choisir une couleur dans la liste (en cliquant à droite dans la liste) ou bien créer une nouvelle couleur en donnant son code RGB.

**OK** Enregistre vos modifications et ferme la fenêtre.

La liste à droite contient l'énumération de tous les sons d'ambiance qui peuvent être entendus dans la région. Vous pouvez ajouter des sons en sélectionnant "Gameplay/ Sounds..." puis en ajoutant les sons dans la liste par Drag'n drop. Puis, vous pouvez définir les probabilités que le joueur entende tel ou tel son. Comme pour les climats, cette valeur est un pourcentage et la somme des valeurs attribuées ne doit pas être supérieure à 100, mais peut être inférieure à 100. La priorité des sons ne peut pas être modifiée. Mais vous pouvez enlever des sons de la liste en les sélectionnant puis en appuyant sur la touche [DEL].

### 8.3.2 Modifier les propriétés d'une cellule

Pour modifier les propriétés d'une cellule, sélectionner "World/Exterior Cell...". La fenêtre "Exterior Cell" apparaît. Dans cette fenêtre vous pouvez choisir une des cellules extérieures. Vous pouvez alors modifier sa région d'appartenance et l'illégalité du repos dans cette cellule.

### 8.4 N'oubliez pas de tester...

De même que pour les cellules d'intérieur, il faut tester vos créations avant de la rendre disponible pour les autres joueurs. Utiliser la même démarche et la même rigueur.

## 9 La création des PNJ pas à pas.

### 9.1 *Des précautions à prendre*

Tout d'abord, IL FAUT TOUJOURS SAVOIR CE QUE L'ON VEUT FAIRE. Et encore une fois, un support écrit quelque part, même succinct, vous sera d'un secours énorme. Ce support vous servira de guide pour les modifications que vous faites, un guide qui tient le rôle d'un fil conducteur et d'un garde-fou. Vous saurez ce qu'il faut faire et vous n'en ferez pas plus, ce qui évitera la création d'un mod peu propre.

Je vous conseille aussi fortement de NE PAS COPIER un autre PNJ pour démarrer votre création. En faisant un Copier - Coller vous récupérez toutes les spécificités de l'original... et vous pourriez avoir un comportement qui ne correspond pas à ce que vous attendiez.

### 9.2 *La création de l'apparence du PNJ*

Dans la fenêtre des Objet, sélectionnez l'onglet « NPC ».

Vous avez maintenant devant vous la liste des PNJ qui interviennent dans l'univers de MW. Ils sont identifiés de manière unique par leur ID. Les autres colonnes du tableau vous donnent des renseignements complémentaires sur les caractéristiques des PNJ.

Pour créer un nouveau PNJ, faites un clic droit dans le tableau et sélectionnez 'New'. Cela va faire apparaître une nouvelle fenêtre.

(Vous pouvez vous reportez à la section IHM de ce document pour des explications sur la significations des éléments de cette interface)

La première chose à faire est de choisir une ID pour votre PNJ. Elle doit bien sûr être unique pour tout le jeu, mais rassurez vous, si jamais ce que vous proposez existe déjà, l'éditeur vous le signalera.

Un bon système pour faire des ID unique est d'accolé des lettres devant le nom de votre PNJ. Personnellement, je trouve qu'il y a une certaine logique à choisir un système de nommage qui ressemblerait à ceci :

<vos\_initiales>\_<initiales\_de\_votre\_mod>\_votreID

Ce qui donnerait dans ce cas

\_NCr\_es\_MonPNJ

L'ID que vous choisissez est limitée en nombre de caractères (24), ce n'est pas le nom du PNJ tel qu'il sera vu dans le jeu.

On vient de régler le point crucial de la création d'un PNJ, son identifiant. Tout le reste des réglages est extrêmement simple et de toute façon, vous pourriez les modifier plus tard.

Le Name du PNJ, est l'identité sous laquelle il est connu dans l'univers de MW (ou de votre mod). Dans la section de droite, vous pouvez sélectionner l'apparence (en fait uniquement la tête) de votre PNJ. En choisissant

un modèle pour la tête et la chevelure de votre PNJ, ce sont les mêmes que ceux qui sont proposés au joueur lorsqu'il crée son personnage au tout début (oui, je sais cela n'aide pas beaucoup, mais bon pour l'instant je n'ai pas le temps de vous montrer à quoi cela ressemble). L'apparence extérieure de votre PNJ est fonction de la race que vous lui attribuez dans la zone dédiée. Si vous sélectionnez l'option Female, votre PNJ sera une femme (ou son équivalent dans la race du PNJ).

Il est possible d'attribuer un script pour un PNJ, c'est de cette façon que sont construits les compagnons qui nous sont tellement utiles. A moins d'un cas vraiment particulier, vous n'avez pas à modifier cette section.

Pour chaque PNJ, il est possible de lui attribuer une classe (dans cette liste se trouvent également les classes que vous avez créées pour votre mod), une faction et un rang dans la faction. Vous pouvez également attribuer un niveau au PNJ dans sa classe. Pour ce qui est du niveau de ce PNJ, vous pouvez vous inspirer des PNJ que vous connaissez dans l'environnement de MW. Dans le cas où votre PNJ est membre d'une guild, veillez à respecter une certaine logique dans les niveaux en fonction du rang dans la faction (le grand maître ne devrait pas être plus faible que le grouillot de base...)

Si vous désirez que votre PNJ soit un commerçant ou bien qu'il vende des sorts, choisissez lui une classe qui offre ces services, mais Notez bien que ce choix ne sera pas en soit suffisant pour que le PNJ soit un commerçant (barter service).

Toutes ces opérations devraient être relativement simple puisque vous avez noté quelque part ce que vous vouliez faire.

En ce qui concerne les options « Essential » et « Corpses Persist », je pense qu'elle parle d'elle-même, il faut valider « Essential » si vous pensez que la mort du PNJ va être fatale au prolongement de votre quête (c'est le cas de Caius, un message vous invite alors à reprendre une sauvegarde antérieure). D'une manière générale, essayez d'éviter de sélectionner cette option, et prévoyez des alternatives multiples pour le prolongement de la quête.

L'option Corpse Persist empêchera le joueur de se débarrasser du corps de son innocente victime... J'ose à peine demander l'utilité réelle de cette fonctionnalité.

L'option Respawn est l'une de celles qui est à employer avec le plus de précautions, peu de PNJ reviennent discuter avec le PJ qui les a occis, si vous cochez cette option, j'espère que vous savez ce que vous faites.

Sauf cas vraiment, vraiment particulier ne touchez pas à "Add Animation File.", c'est le bouton qui va vous permettre de modifier l'animation attachée à votre PNJ...

Toujours pareil, sauf cas particulier, ne changez pas la texture du Sang...

Vous imaginez sans problème la difficulté (et l'ennui) qu'il peut y avoir à modifier toutes les caractéristiques de votre PNJ individuellement. Heureusement Bethesda se propose de le faire automatiquement pour vous si vous cochez l'option "Aut Calculate Stats.": cela présente l'autre avantage de faire un personnage équilibré sans que VOUS ayez à vous en occuper... cela fera un mod plus équilibré que ce vous pourriez faire.

Il y a des cas où il est malgré tout nécessaire de désactiver cette option, notamment lorsque vous désirez modifier la liste des sorts vendus par un magicien...ou modifier les services rendus par un PNJ. Lorsque cette option est activée de nombreuses options du PNJ sont évaluées automatiquement en fonction de la classe que vous lui avez attribuée (cf Création d'une nouvelle classe)

Vous pouvez classer les compétences par ordre décroissant ou croissant de leur valeur en cliquant sur l'intitulé « Value » de la colonne du tableau des Compétences en bas à droite.

Le bouton AI vous permet d'accéder aux paramètres de l'IA que vous attribuez à votre PNJ. Il existe quelque part dans ce document une section qui explique le fonctionnement des ces différents paramètres. Par défaut, attribuer leur le package AI nommé « Wander », le PNJ aura alors le comportement qui s'approche le plus possible de *'le PNJ s'ennuie et attends que quelque chose se passe...'*

Il ne reste plus qu'à cliquer sur "Save" pour enregistrer les modifications que vous avez faites sur le PNJ. Pour le voir, charger une cellule (de préférence d'intérieur, c'est moins long) et glissez votre PNJ dedans, pour voir son apparence.

Vous pouvez modifier la tête et les cheveux de votre PNJ et voir immédiatement les modifications apportées sur son apparence.

Mais nous verrons cela en plus tard.

### 9.3 *Equiper votre PNJ*

Dans le tableau en bas à droite, vous pouvez voir deux onglets;

- Item : contient la liste des objets que le PNJ porte (armes, armures, vêtements, bijoux....)
- Spell : contient la liste des sorts que le PNJ connaît et peut potentiellement utiliser lorsqu'il est attaqué.

Pour ajouter des éléments dans ces deux listes, on procède de la même manière. On fait un drag'n drop depuis, la fenêtre des objets vers ces listes.

Pour l'équipement de votre PNJ, il faut là encore savoir à l'avance ce que vous voulez faire... faites une liste des « commissions ».

Pour voir de quoi votre PNJ aura l'air, placé le dans la fenêtre de Rendu (Render View)

Une façon simple de trouver les éléments que vous voulez donner au PNJ est de trier les éléments par ordre alphabétique en fonction de leur nom (pas de leur ID), pour faire cela, cliquez sur l'en-tête de colonne Name dans les onglets « Clothes » (Vêtements) et « Armor » (Armure)...

#### 9.3.1 *Une quantité infinie d'objet*

Si votre PNJ est un marchand, vous pouvez souhaité qu'il dispose d'un stock d'un certain type d'objet. Un marchand de flèches, par exemple, devrait pouvoir vendre des flèches au PJ chaque fois que celui-ci vient le voir. Mais, il serait absurde de lui donner un nombre très grand de flèches qui décroîtrait en fonction des achats successifs du PJ. Mettons qu'il en dispose de 20 à chaque transaction.

Il suffit pour faire cela, de rentrer un nombre négatif (-20) dans la colonne quantité d'objet. Le marchand aura alors son stock d'objet (flèches) régulièrement renouvelé.

Si le joueur tue le PJ qui dispose d'une quantité infinie d'objets ce processus de renouvellement s'arrête.... (le marchand ne fabrique plus de flèche puisqu'il est mort )

#### 9.3.2 *Une précaution à prendre*

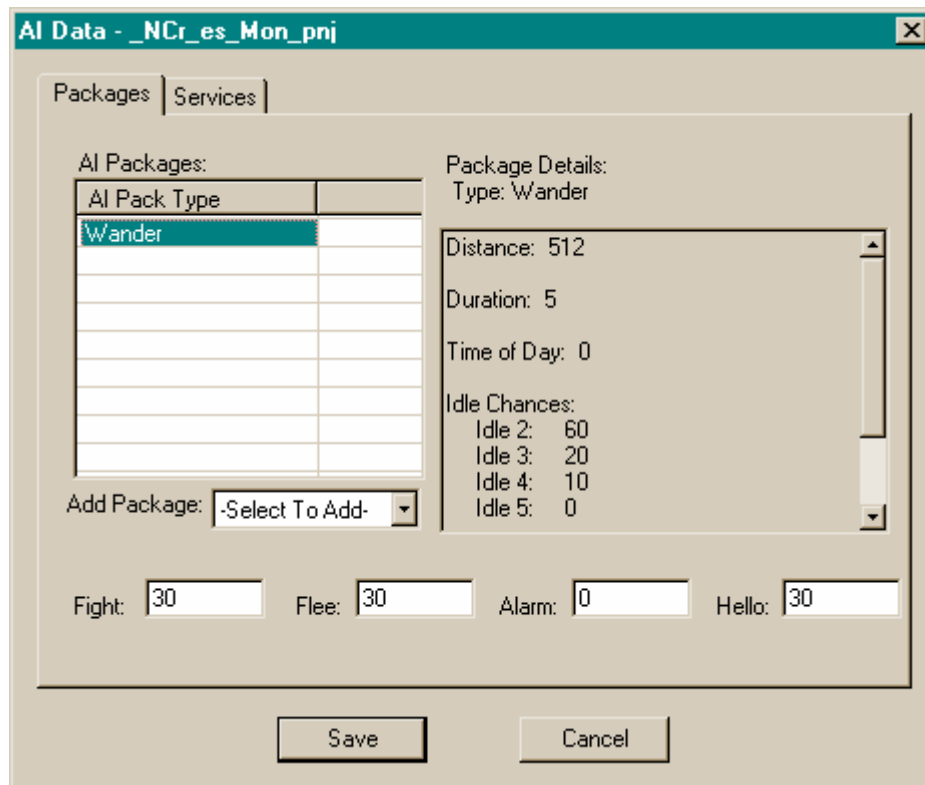
Si vous êtes obligé de modifier un PNJ existant, sachez que les modifications que vous pourriez vouloir apporter à son inventaire ne seront pas prises en compte par le jeu, si le joueur a déjà parlé au PNJ et a fortiori si ils ont échangé des objets ensemble. Lors des sauvegardes, l'inventaire du PNJ est stocké dans la sauvegarde du joueur, et c'est cette information qui est retenue par le moteur du jeu.

Donc si vous avez besoin de donner un objet au PJ, n'utiliser jamais un PNJ déjà existant. Vous n'avez aucune maîtrise de son inventaire.

### 9.4 *Les paramètres I.A.*

#### 9.4.1 *L'onglet Package*

Lorsque vous appuyez sur le bouton I.A. de la fenêtre de création du PNJ, vous faites apparaître une nouvelle fenêtre. Dans cette fenêtre vous pouvez paramétrer le comportement de votre PNJ, en utilisant des types de comportements I.A. qui sont préprogrammés par Bethesda. Vous pouvez également choisir un type de services que le PNJ pourra offrir au joueur.



Dans cette fenêtre vous pouvez choisir des types de comportements pour le PNJ en utilisant des groupes de fonctions de l'I.A.

Le comportement du PNJ tel qu'il apparaîtra dans le jeu est paramétré par les choix que vous faites à ce niveau. Ses déplacements, ses errances et ses actions sont paramétrés dans cette fenêtre.

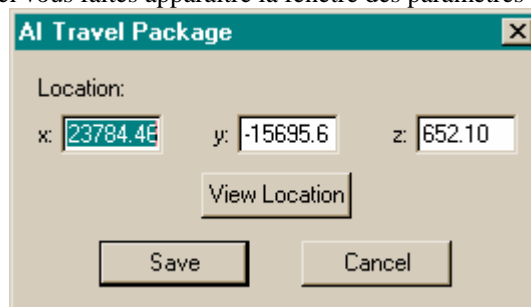
Il est possible d'ajouter de nombreux packages IA, tous sont choisis parmi la liste proposée dans le menu déroulant. Il est possible d'en ajouter autant que l'on veut (y compris ajouter plusieurs fois le même package avec des paramètres différents), il faut simplement savoir que les packages sont activés successivement et en boucle. C'est-à-dire que dans le tableau nommés AI Package vous allez construire une liste de comportement, ils seront effectués les uns après les autres successivement jusqu'à la fin de la liste. A la fin de cette liste, le moteur de jeu reprend au début de la liste.

Il est possible d'ajouter des comportements pris dans les différents types de comportements offerts par l'interface. Pour ajouter un comportement, vous sélectionnez un package A.I dans le menu déroulant, et une fenêtre apparaît qui vous permet de paramétrer les packages IA que vous utilisez.

Nous allons maintenant les détailler successivement.

#### 9.4.1.1 Le package Travel

En sélectionnant le menu Travel vous faites apparaître la fenêtre des paramètres du package IA.



En utilisant le package I.A. le PNJ va se déplacer à l'intérieur de la cellule dans laquelle il se trouve jusqu'à la coordonnées donnée par les trois paramètres X, Y et Z.

En appuyant sur le bouton ‘View Location’ vous faites apparaître une croix rouge dans la fenêtre de Rendu. Vous pouvez déplacer cette croix par drag’n drop à l’intérieur de la cellule. (Attention quand même, chez moi la croix ne suit pas vraiment, c’est le moins que l’on puisse dire, le mouvement de la souris).



#### 9.4.1.2 Le packageWander

En anglais, Wander signifie “error”, en clair cela veut dire qu’en utilisant le package Wander, le PNJ aura l’apparence externe de ‘je ne sais pas trop ce que je fais à part attendre’. En pratique c’est le package I.A. le plus riche de l’univers de MW.

AI Wander Package	
Distance:	512
Duration: (in hours)	5
Time of Day: (?:00)	0
Idle Chances	
Idle 2:	60
Idle 3:	20
Idle 4:	10
Idle 5:	0
Idle 6:	0
Idle 7:	0
Idle 8:	0
Idle 9:	0
<div>Save</div> <div>Cancel</div>	

Pour déterminer le comportement d’un PNJ qui utilise ce package, le moteur du jeu réalise un tirage aléatoire puis en fonction des valeurs que vous avez données à chacun des types de comportement, il attribue une animation spécifique au PNJ. Les valeurs des différents coefficients que vous attribuez dans les cases Idle X sont des pourcentages, donc la somme des toutes les valeurs doit être égale à 100. Le tableau suivant donne la signification des coefficients.

Idle0	Ne fait rien
Idle2	regarde autour de lui
Idle3	regarde derrière lui
Idle4	se gratte la tête
Idle5	époussette ses habits ou son armure
Idle6	se frotte les mains (pratique pour un marchand)
Idle7	regarde ses doigts, puis furtivement autour de lui (prépare un mauvais coup)
Idle8	absorbé dans ses pensées
Idle9	cherche son arme

L'information Distance, indique le rayon dans lequel le PNJ est libre de se déplacer autour de sa position initiale lorsque le Package Wander a été activé.

Duration (in hours), indique la durée en heure (de jeu) durant laquelle ce package est actif.

Time Of Day, indique l'heure à partir de laquelle le PNJ exécute cette action.

#### 9.4.1.3 Le package Escort

Vous utilisez le package Escort lorsque vous désirez que le PNJ escorte quelqu'un ou quelque chose jusqu'à un certain endroit. En l'escortant, le PNJ ouvrira le chemin pour la personne escortée, et la défendra contre tout ce qui pourrait l'attaquer.

Le package Escort fonctionne avec l'ID de la personne ou chose à escorter. (l'ID du joueur est Player). Faites attention à ne pas confondre l'ID d'un PNJ avec son nom..

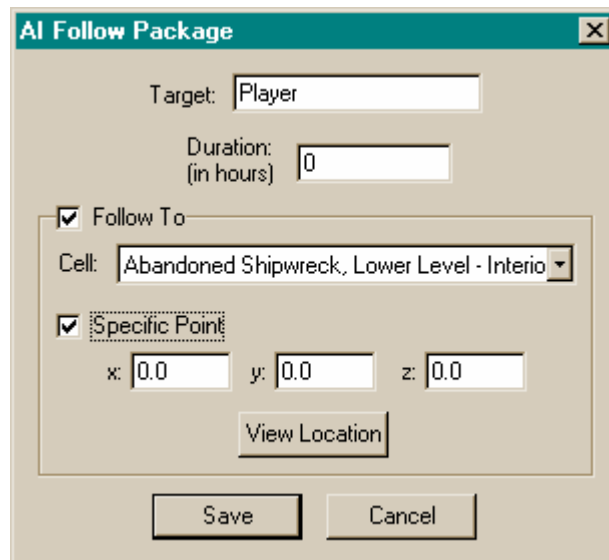
Vous pouvez donner une durée pour l'escorte (en heure de jeu), si vous la fixez à la 0, le PNJ escortera toujours la cible.

Si vous choisissez d'escorter, non pas pendant une durée précise mais jusqu'à un endroit précis, vous cochez l'option Escort To. Ceci devrait rendre éditable le nom de la cellule jusqu'à laquelle l'escorte aura lieu. Dans ce cas le PNJ cessera d'escorter la cible dès l'entrée dans la cellule.

Comme ceci peut ne pas être suffisant, vous pouvez spécifier une coordonnée à l'intérieur de la cellule (en x, y z) et vérifier (en appuyant sur View Location) que les coordonnées saisies sont les bonnes. Si elles ne vous conviennent pas, vous pouvez déplacer le repère par drag'n drop.

#### 9.4.1.4 Le package Follow

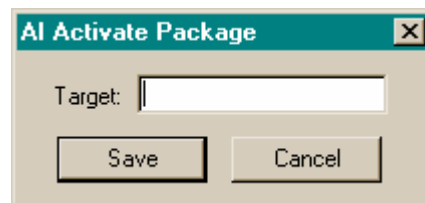
Le package Follow fonctionne de la même manière que le package Escort, à ceci près que le PNJ qui suit n'ouvre pas la voie. Je ne crois pas qu'il défende son compagnon (mais c'est à vérifier). C'est un package qui peut être utile pour constituer des caravanes qui se déplacent dans le MW, ou des groupes d'animaux.



Tous les paramètres de ce package ont la même signification que les paramètres que l'on attribue à une action d'escorte.

#### 9.4.1.5 Le package Activate

En utilisant le package Activate, vous dites à votre PNJ d'effectuer l'action par défaut sur l'objet qu'il donne dans le champ cible. C'est-à-dire que si la cible est une porte, par exemple, il l'ouvrira, si la cible est une personne il lui parlera etc..



Dans le champ Target, on spécifie l'ID de l'objet à Activer.

En combinant ces différents package dans les paramètres IA de votre PNJ vous pouvez arriver à lui donner un comportement « humain ». Puisqu'il est possible d'enchaîner les actions les unes derrière les autres, vous pouvez le faire se déplacer dans une ville, puis rentrer dans un bâtiment, le faire ressortir etc.

Il faut toutefois faire attention au fait que les PNJ ne sont activés (et donc leur comportement) qu'à partir du moment où vous rentrez dans la cellule (où à partir du moment où elle est activée).

Sur l'onglet I.A., il y a encore 4 champs, Fight, Flee, Hello et Alarm. Vous pouvez mettre des valeurs numériques comprises entre 1 et 100 dans ces champs. Ils déterminent l'attitude du PNJ envers le joueur. Il peut l'attaquer à vue, ou fuir lorsqu'il le voit. Vous pouvez y paramétrer la réaction qu'il aura si le joueur commet un crime dans son entourage.

Les valeurs que vous rentrez à ce niveau ne sont pas celles qui sont utilisées dans le jeu. Ce sont des valeurs de bases qui sont ensuite modifiées par les paramètres du jeu. (Ces paramètres sont accessibles par le menu Gameplay/settings, je vous recommande de ne pas trop modifier ces valeurs à moins de vraiment savoir ce que vous faites).

#### 9.4.1.6 Fight

L'élément Fight, détermine la propension que le PNJ a à attaquer le PJ. Lorsque cette valeur est à 100, il attaquera toujours le PJ. Typiquement, lorsque vous créez une cache de brigand, sauf cas exceptionnel, les PNJ

que vous mettez dans cette cachette doivent avoir leur valeur Fight à 100. Sinon, en les attaquant le PJ pourrait commettre un crime (puisqu'il ne se défend pas).

Les différentes actions du joueur dans l'entourage du PNJ vont modifier sont attitude de la façon suivante :

Action du joueur	Valeur par défaut	Formule de modification avec les paramètres du jeu
Distance du joueur	20-(Char Distance*0.005	iFightDistanceBase - (Char Distance * fFightDistMult)
Attaque le PNJ	100	iFightAttack
Disposition du PNJ envers le joueur	( 50 - Disposition ) * 1	(50 - Disposition) * fFightDispMult
Le joueur vole quelque chose	5 * Item Value	fAlarmStealing * Item Value
Pick pocket sur le PNJ	25	iAlarmPickPocket
Commets un crime	25	iAlarmTresspass
Taunt (Admire)	D'après les formule des tables de persuasion	
Intimide	D'après les formule des tables de persuasion	
Corromp (Bribe)	D'après les formule des tables de persuasion	

Ces modificateurs influent sur le comportement du PNJ. Mais on peut donner les équivalences suivantes, qui ne sont qu'indicatives ;

100	Attaquera toujours
95	Attaque en fonction de la distance (3000 unités)
90	Attaque en fonction de la distance (2000 unités)
80	Attaque si le PJ s'approche trop ou si le PNJ n'aime pas le PJ (1000 unités, disposition 10)
70	Attaque si vous êtes trop proche et qu'il ne vous aime vraiment pas (1000 unités, 35 disposition)
60	Attaque si il ne vous aime pas du tout (Disposition <30)
50	Attaque si il vous hait – disposition = 0
40	Attaque si il n'aime pas le PJ et qu'il est proche (500 unités, Disposition 10)
30	Attaque si il hait le PJ et qu'il commis un crime
20	Attaque si il hait le PJ et qu'il a commis de nombreux crimes
10	Attaque si il hait le PJ et que le PJ a commis des crimes contre lui
0	N'attaque que si le PJ commence

#### 9.4.1.7 Alarm & Crimes

Lorsque le PJ commet une action illégale dans le voisinage d'un PNJ, et que ce PNJ s'en aperçoit, il dira quelque chose au joueur. Par cette action, il informera également les autres PNJ dans la cellule. A la suite de cela, tous les PNJ modifient leur réglages qui sont fonction de ce la valeur que vous avez rentré dans Alarm.

Si un PNJ a une Alarm réglée à 100, il mettra une prime sur la tête du PJ.

Les actions du PJ ont évidemment un impact différent suivant qu'il vole une cuillère en bois ou qu'il tue un PNJ.

Crime commis	Prime attribuée	Variable des paramètres de jeu
Attaque un PNJ	40	IcrimeAttack
Tue un PNJ	1000	IcrimeKilling
Vole un objet	1	fCrimeStealing * Item Value
Vole une personne (Pic Pocket)	25	IcrimePickPocket
Enfreint la loi (force une serrure)	5	IcrimeTresspass
Tautn	5	IcrimeTaunt
Intimide	5	IcrimeIntimidate

De plus, si le PNJ est un garde (si sa classe est Guard), il aura un comportement supplémentaire: il interceptera le PJ, si nécessaire il le poursuivra et l'arrêtera.

Si la prime sur la tête du PJ est supérieure à 10 000, il l'attaquera directement au lieu d'amorcer un dialogue d'arrestation

On peut signaler que les gardes attaquent également toutes les créatures qui s'en prennent aux gens (y compris le PJ).

#### 9.4.1.8 Flee – La fuite

C'est la propension que le PNJ a à fuir le PJ. L'utilisation en est assez délicate car de nombreux paramètres sont pris en compte pour déterminer si le PNJ s'enfuit ou non.

Par exemple la quantité de dommage absorbée par le PNJ mais ce n'est pas le seul car on peut aussi prendre en compte la quantité de dégâts infligés au PJ...

Mettre cette valeur à 100 devrait faire fuir le PNJ en permanence mais ce n'est même pas encore le cas.

#### 9.4.1.9 Hello – Prise en compte du joueur

C'est la distance en dessous de laquelle le PNJ s'arrêtera pour saluer le joueur.

Cette valeur qui est à 30 par défaut est multipliée par le paramètre iGreetDistanceMultiplier, (à 7 par défaut) pour donner la distance en unité de distance du jeu.

Par exemple, choisir 30 provoquera un salut adressé au joueur pour une distance de 210 unités (environ 3 mètre, 10 pieds)

#### 9.4.2 L'onglet Services

A chaque PNJ, vous pouvez attribuer des services qu'il pourra rendre au joueur. C'est dans cet onglet que l'on spécifie si un PNJ est un marchand, la nature de ce qu'il vend et achète. Les sorts qu'il peut vendre, les enchantements etc...

Si vous avez coché l'option Auto Calc Stat, vous ne pourrez pas changer beaucoup de chose dans cette zone. En effet, les services offerts sont fonction de la classe que vous avez attribuée au PNJ. Vous pouvez néanmoins modifier la quantité d'or d'un marchand, ou les cellules de destination pour les voyages.

**AI Data - a shady smuggler**

Packages Services

**Buys / Sells**

☐ Weapons ☐ Ingredients ☐ Apparatus ☐ Magic Items  
☐ Armor ☐ Picks ☐ Repair Items ☐ Potions  
☐ Books ☐ Probes ☐ Miscellaneous  
☐ Clothing ☐ Lights ☐ Spells

Barter Gold

**Other Services**

☐ Training  
☐ Spellmaking  
☐ Enchanting  
☐ Repair

**Travel Services**

Cell    
 Cell    
 Cell    
 Cell

Pour information, la quantité d'or que vous donnez à marchand est réinitialisé tous les jours à une heure du matin (heure du jeu) à la valeur que vous spécifiez.

Dans le cas particulier d'un marchand, il ne faut pas oublier de lui attribuer des objets en quantité infinie.... Pour qu'il puisse en vendre à plusieurs reprises. (regardez la section Equiper un PNJ)

IL est également possible de modifier les services offerts par un PNJ et de les attribuer à votre convenance. Pour cela il faudra désactiver l'option Auto-Calc-Stat, dans la fenêtre principale de création du PNJ.

**AI Data - a shady smuggler**

Packages Services

**Buys / Sells**

☐ Weapons ☐ Ingredients ☐ Apparatus ☐ Magic Items  
☐ Armor ☐ Picks ☐ Repair Items ☐ Potions  
☐ Books ☐ Probes ☐ Miscellaneous  
☐ Clothing ☐ Lights ☐ Spells

Barter Gold

**Other Services**

☐ Training  
☐ Spellmaking  
☐ Enchanting  
☐ Repair

**Travel Services**

Cell    
 Cell    
 Cell    
 Cell

Pour que votre PNJ puisse vendre et acheter des armes, par exemple, il faut cocher l'option Weapons. Si vous désirez qu'il puisse entraîner le PJ dans une des caractéristiques principales de la classe du PNJ vous sélectionnez l'option Training, etc.

#### 9.4.2.1 Les transports

Les PNJ peuvent offrir au PJ de le faire voyager vers une destination au choix du PJ parmi 4 possibilités. Cela aura le même effet qu'une téléportation.

Pour chaque destination à proposer au PJ, il faut sélectionner la cellule de destination (dans le menu déroulant Cell) puis positionner le Marker de voyage associé.

En cliquant sur Select Marker, vous chargez la cellule de destination dans la fenêtre de Rendu et vous êtes centré sur une flèche verte.

Vous pouvez déplacer et réorienter la flèche à votre guise comme n'importe quel objet. La direction indiquée par la flèche est celle dans laquelle le joueur regardera à son arrivée. (c'est le même fonctionnement que pour les marqueurs de portes).

#### 9.4.2.2 Les services proposés par les classes de Morrowind.esm

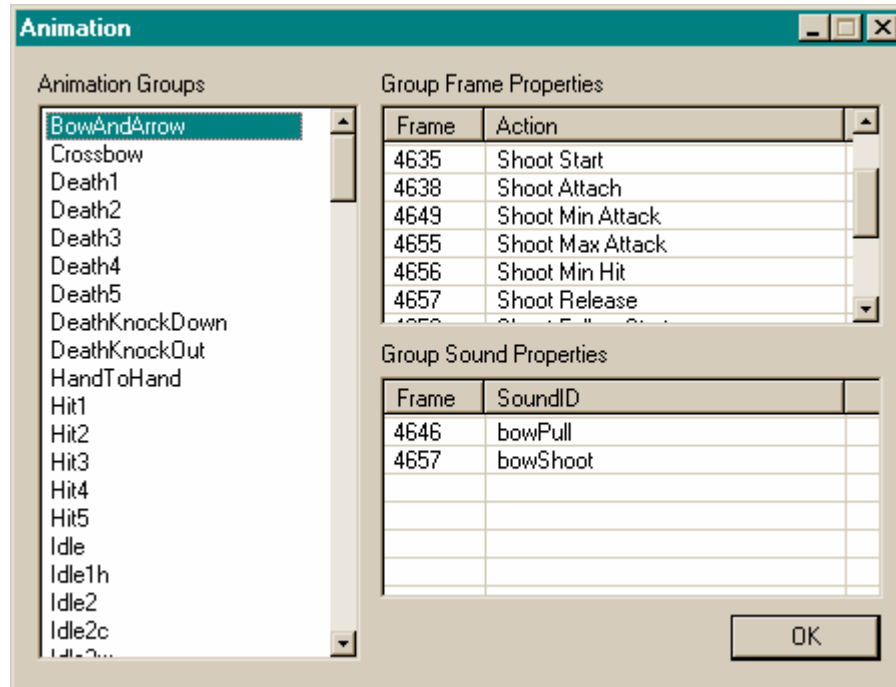
Dans le fichier morrowind.esm, de nombreuses classes sont déjà construites. Vous pouvez les réutiliser en fonction de ce que vous voulez faire ou en créer d'autres. Le tableau ci-dessous résume les services offerts par défaut (avec l'option Auto-Calc-Stat) par les membres des dites classes.

N'oubliez que vous pouvez toujours créer votre propre classe. (Malheureusement, on ne peut pas créer ses propres services...)

	Achat / Vente													
	Armes	Armures	Livres	Habits	Ingédients	Crochet	Sonde	Lumière	Appareil (Alch.)	Outils de réparation	Divers	Sorts (Parch.)	Objets Magiques	Potions
Classe													Entraînement	Creation de Sorts
Enchant.														
Réparation														
Acrobat														
Agent														
Alchemist														
Alchemist Service														
Apothecary														
Apothecary Service														
Archer														
Assassin														
Assasin Service														
Barbarian														
Bard														
BattleMage														
BattleMage Service														
Bookseller														
Buoyant Armiger														
Caravaner														
Champion														
Clothier														
Commoner														
Crusader														
Dreamers														
Drillmaster														
Drillmaster Service														
Enchanter														
Enchanter Service														
Enforcer														
Farmer														
Gondolier														
Guard														
Guild Guide														
Healer														
Healer Service														
Herder														
Hunter														
Knight														
Mabrigash														
Mage														
Mage Service														
Master of Arms														
Merchant														
Miner														
Monk														
Monk Service														
Necromancer														
NightBlade														
NightBlade Service														
Noble														
Ordinator														
Ordinator Guard														
Pauper														
Pawnbroker														
Pilgrim														
Priest														
Priest Service														
Publican														
Rogue														
Savant														
Savant Service														
Scout														
Sharpshooter														
Shipmaster														
Slave														
Smith														
Smuggler														
Sorcerer														
Sorcerer Service														
Spellsword														
Thief														
Thief Service														
Trader														
Trader Service														
Warlock														
Warrior														
Wise Woman														
Wise Woman Service														
Witch														
Witch Hunter														

### 9.5 Animation

Le bouton Animation fait apparaître une fenêtre qui contient la description du comportement gestuel de votre PNJ. Les animations sont prises sur l'animation classique fournie par Bethesda. Je vous livre ici une capture d'écran de la chose.



La zone de gauche contient une liste de toutes animations qui sont décrites dans le fichier d'animation que vous avez spécifié (grâce à Add Animation File).

Dans les scripts vous avez la possibilité de jouer des animations ou des groupes d'animations grâce aux commandes qui s'y réfèrent (PlayGroup, SkipGroup...), les paramètres que vous passez à ces fonctions sont précisés ici... n'hésitez pas à vous y référer de temps en temps.

A chaque animation est associé (ou non) un son, ces sons sont modifiés par vous dans le menu Sounds de Gameplay.

## 10 Dialogue/Journal-Based Quest Creation

Comment faire une bonne quête?

Très largement inspirée de Dialogue/Journal-Based Quest Creation

Tutorial de Dale French

artydog@bellsouth.net

### 10.1 L'idée de base

Une idée sur un papier....

La première chose à faire est de prendre des notes sur papier, ou n'importe quel autre support. N'ouvrez jamais le TESCS sans avoir avant tout une idée relativement précise de ce que vous voulez faire.

Cela vous sera d'une grande aide et c'est surtout un début de garantie sur la qualité du résultat final.

Le plus important est la chronologie des événements. Avoir quelque part une liste chronologique de l'enchaînement des événements est quelque chose de primordial.

Prenons l'exemple de la première quête que vous confie le capitaine qui vous accueille : aller porter des papier à Caius Cossades.

1. récupérer les papiers
2. aller à Balmora
3. se renseigner sur Caius
4. lui donner les papiers.

C'est relativement simple, mais sans cette trame temporelle, on aurait pu faire n'importe quoi...

Je vous propose d'articuler cette explication autour d'une quête simple. Cela vous évitera d'avoir à lutter pour trouver une idée, et cela me permettra d'être un peu plus clair.

Appelons cette quête 'Où sont mes chaussures?'

Dans cette quête, vous irez voir Ra'Virr à Balmora. Il vous demandera de l'aider à retrouver ses chaussures. Vous les trouverez et les lui rapporterez.

Oui, je vous entend d'ici, mais c'est nul. Et moi, je vous répondrai d'apprendre à marcher avant de vouloir battre le record du monde de 110 m haies.

Nous verrons ici, comment construire un dialogue autour de cette quête, et comment ajouter des entrées dans le journal.

Voici la chronologie que adopterons:

Rencontrer Ra'Virr  
Ecouter son histoire à propos de ses chaussures  
Lui trouver ses pompes  
Et lui rapporter.

Qui est impliqué dans l'affaire?

Bon là, je vous l'accorde c'est simple, il n'y a qu'une seule personne impliquée. La liste sera vite faite.

Mais lorsque vous vous lancerez dans la création d'un vrai plug, vous serez content d'avoir sous la main la liste des personnes impliquées. (Quand je dit personne, je pense PNJ).

Cette liste contiendra évidemment les pnj existants mais aussi et surtout ceux qu'il vous faudra créer.

Un dernier conseil, pour vous simplifier la tâche, noter surtout les ID des pnj (c'est de cette façon qu'ils sont connus dans l'éditeur).

Dans notre cas, la liste des pnj se limite à

Ra'Virr

Une clé pour les entrées dans le journal: une ID

Puisque nous allons apprendre à utiliser le journal et les entrées dans le journal, il nous faut une ID (unique) pour créer une section dans le journal. Dans un futur pas si lointain, vous aurez peut être besoin de plus d'une section, mais, ici, une seule est suffisante.

Je vous recommande pour les ID du journal, comme pour toutes les autres ID de choisir un nom simple et reconnaissable.

Vous pouvez choisir n'importe quoi, le joueur ne la verra jamais, mais .... Vous ne ne la retrouverez peut être plus.

Ecrivez là, et faites attention à son orthographe (Casse aussi)

Ici, nous choisissons :

[Les\\_chaussures](#)

Et maintenant, le reste !

## 10.2 L'entrée en matière de Ra'Virr

Pour ce type de quête, les seules actions que nous allons menées dans le TESCS concernent les dialogues et le journal.

Nous allons donc utiliser la fenêtre des dialogues.

Pour la faire apparaître cliquez sur la bulle de la barre d'outils. (Evidemment avant vous aurez chargé le fichier Morrowind.esm....). Il faudra donc un certain temps pour le contenu de la fenêtre de dialogue s'affiche.

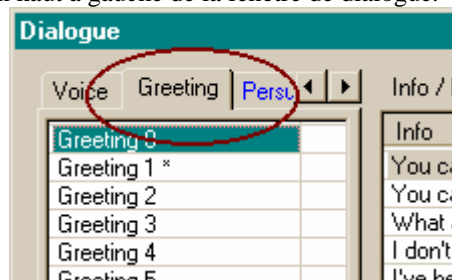


Si c'est la première fois que vous ouvrez cette fenêtre, je pense que vous serez (à juste titre) impressionné. Pour briser le mythe, il y a dans ce document, quelque part un descriptif de cette fenêtre et de ses fonctionnalités.

Si vous ne l'avez pas fait avant, c'est le moment ou jamais de lire ce qui y est écrit.

En ce qui concerne notre affaire, toute conversation commence par des salutations (Greetings). Nous allons donc nous intéresser à l'onglet Greetings.

Cliquez sur l'onglet « Greeting » en haut à gauche de la fenêtre de dialogue.



Dans la zone Info/Response, vous voyez maintenant toutes les salutations que l'on trouve dans MW.

Vous remarquerez que les salutations sont réparties en 10 catégories, numérotées de 0 à 9. Il s'agit d'une convention choisie par Bethesda pour permettre de mieux appréhender l'énorme quantité de saluts autorisés. Essayez un moment d'imaginer de ce que cela aurait pu être si il n'y avait pas eu de catégories...

La salutation d'un personnage est la première chose qui apparaît lorsque vous l'activez (barre espace ou E suivant les cas...). Voyons maintenant à quoi correspondent les différentes catégories de salut telles que Bethesda les voulues.

Greeting 0	Si le PJ est un criminel
Greeting 1	Apparemment des saluts vraiment spécifiques, quand le PJ a accepté une quête donnée par le PNJ...
Greeting 2	Le PJ est un vampire
Greeting 3	Si le PNJ est membre de la guilde Morag Tong (une seule salutation)
Greeting 4	Si le PJ est malade
Greeting 5	Relié aux propositions de quêtes
Greeting 6	Il y en a beaucoup, tout ce qu'il y a là dedans est relié aux quêtes
Greeting 7	Relié au faction apparemment
Greeting 8	Basés sur la réputation du personnage et sur son apparence extérieure (ses habits)
Greeting 9	Le genre de chose que les personnes pourraient vous dire dans les 'dernières rumeurs', et il y

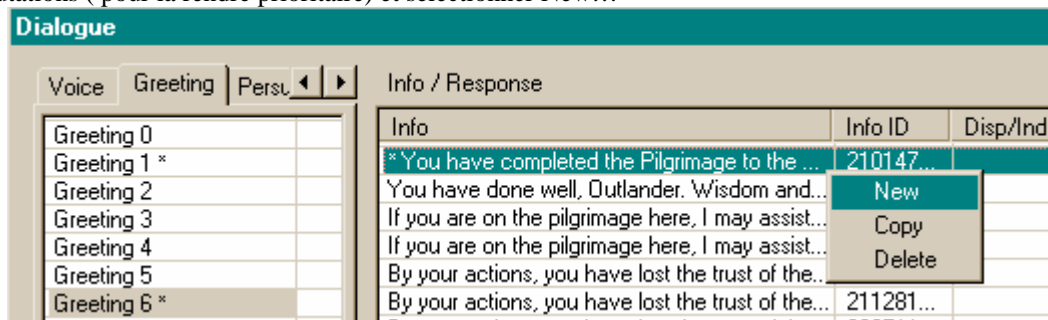
en a BEAUCOUP
---------------

En regardant un peu ce tableau, on se rend compte que le meilleur endroit pour placer ce qui concerne les quêtes est les Greetings 6... il y a peut être une raison à cela ;)

Il est particulièrement déconseillé de placer des salutations dans les salutations 8 et 9, à cause la première salutation de la série 7 (toujours valable à partir d'un certain niveau de la quête principale, ou alors avant les salutations relatives au Nerevarine), vous risqueriez de frustrer un certain nombre des personnes qui ont fini la quête principale.

On veut ajouter une salutation pour Ra'Virr, il faut donc que seul Ra'Virr puisse prononcer cette salutation (dans la série 6).

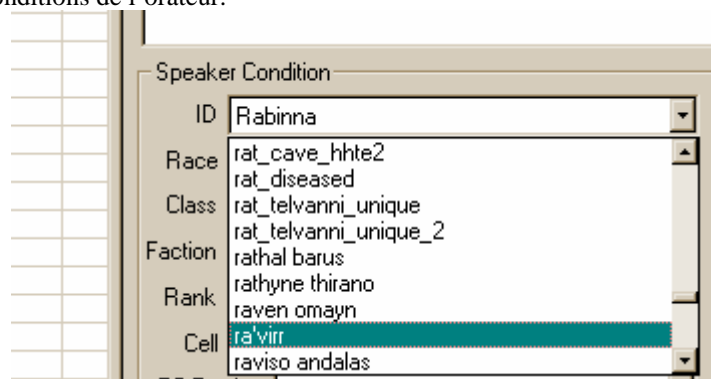
Dans la grande section Info/Response, ajouter une nouvelle saluation. Faites un clic-droit tout en haut de la liste des salutations ( pour la rendre prioritaire) et sélectionner New...



Vous pouvez ensuite ajouter le contenu de la salutation de Ra'Virr... :

« Bonjour Etranger, avez-vous vu mes chaussures ? »
---

Puis, il faut spécifier que seul Ra'Virr doit pouvoir vous accueillir de cette façon. Sélectionner l'ID de Ra'Virr dans le filtre ID des conditions de l'orateur.



A ce niveau de travail, si vous sauvegardez les modifications que vous avez faites, et si vous allez voir Ra'Virr, en n'étant ni malade, ni un vampire, ni invisible; au lieu de vous parler des ses armes daédric, il vous dira ce que vous venez de taper.

### 10.3 Créer un Topic – un sujet de discussion.

Les Topics sont les mots ou les expressions qui apparaissent dans la zone droite de la fenêtre de dialogue du jeu. Ce sont les mots qui apparaissent en bleu dans le discours des personnages.

C'est grâce au Topic que vous pouvez parler avec un PNJ d'un sujet spécifique ou d'un autre.

La gestion des Topic de discussion est très sensible, en effet les Topic sont identifiés par le moteur du jeu au travers de leur contenu, des mots choisis pour être le Topic, et pas par une ID séparée qui ne serait pas vue par le joueur...

Il faut donc faire très attention aux mots que l'on choisit pour un Topic. Il faut éviter les expressions ou les mots que de nombreux PNJ de MW pourraient être amenés à prononcer.

Si vous choisissez d'ajouter des Topic dans notre langue maternelle (le français), éviter par exemple de donner comme simple mot clés «Le»... tout le monde est susceptible de prononcer ce mot dans ses dialogues. Et il apparaîtrait alors toujours en bleu...

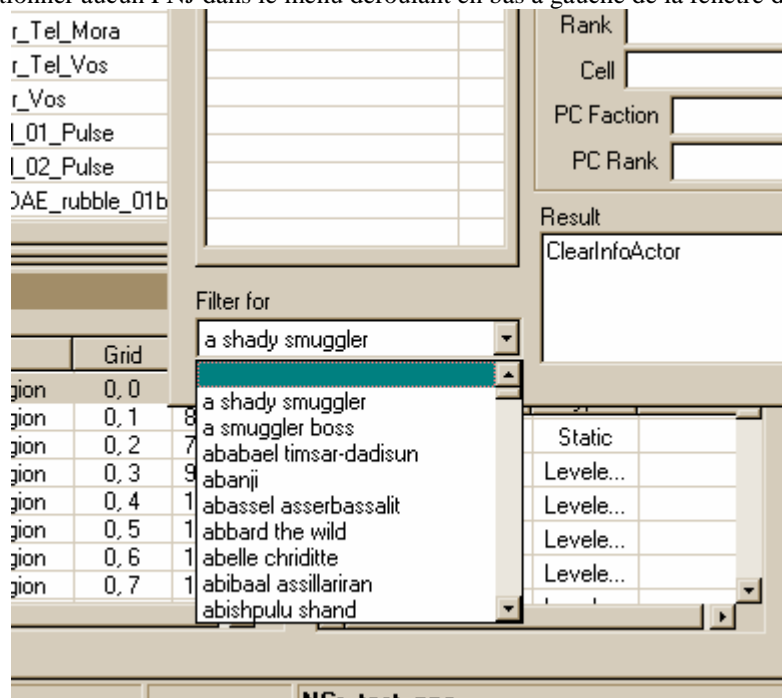
Les Topic sont connus de tous le monde dans le l'univers de MW, pour chaque PNJ, le moteur du jeu parcourt la liste complète des Topic construit la liste par addition des Topic rencontrés.

Pour éviter les conflits avec d'autres Topic, il faut faire attention l'énoncé de celui que vous créez, qu'il ne soit pas identique à un existant.

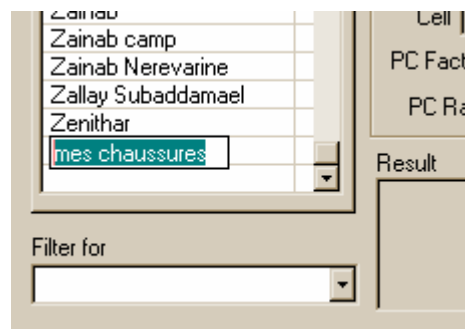
Pour le cas, qui nous occupent nous allons choisir de prendre « mes chaussures » comme Topic.

### 10.3.1 Ajouter un Topic

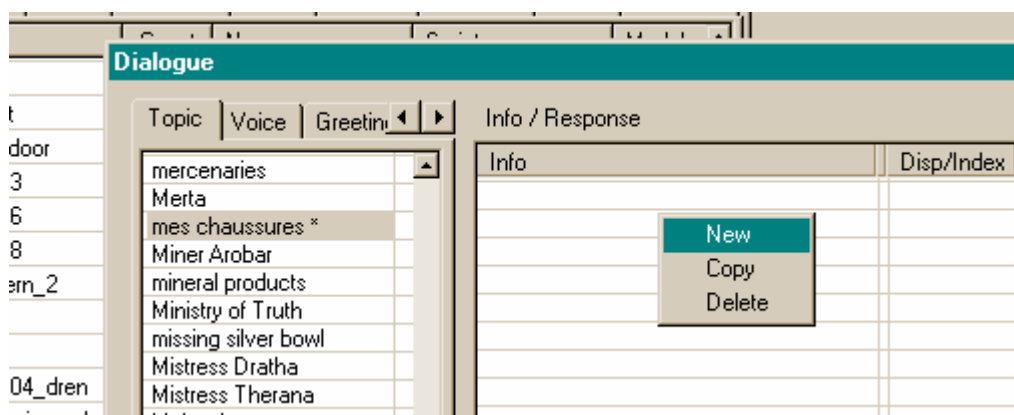
Pour ajouter un Topic, il faut que les informations de la fenêtre de dialogue ne soient pas filtrées. Pour faire cela, il suffit de ne sélectionner aucun PNJ dans le menu déroulant en bas à gauche de la fenêtre de Dialogue.



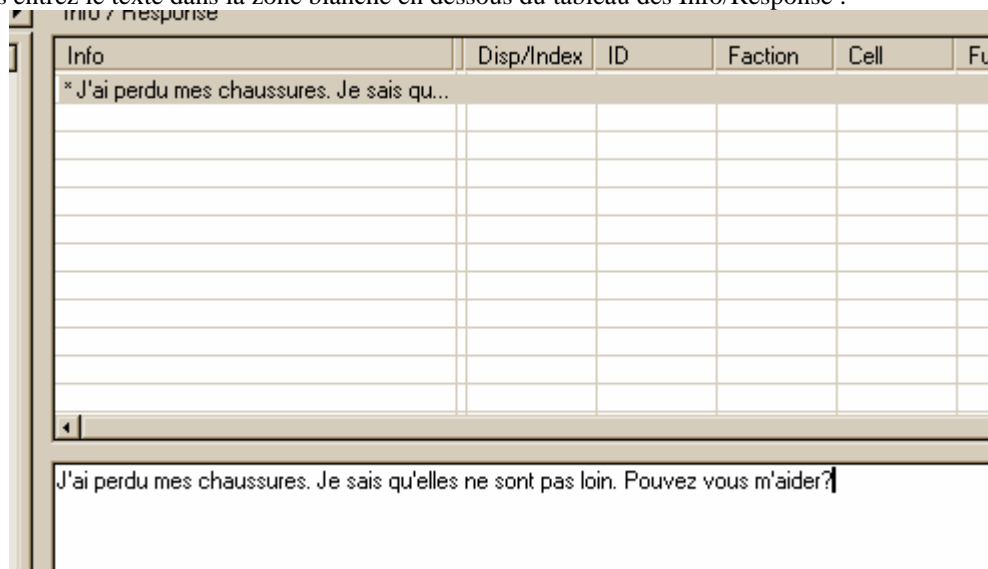
Ensuite, vous sélectionnez l'onglet Topic (en haut à gauche), et dans la liste qui apparaît, vous faites un clic droit et sélectionnez «New» dans le menu contextuel. Vous pouvez maintenant taper le sujet de votre Topic.



Voilà , vous disposez maintenant d'un nouveau Topic. Vous le sélectionnez et par un clic droit dans la zone Info/Response, en sélectionnant 'New' vous pouvez ajouter des paroles prononcées par le PNJ.



Puis, vous entrez le texte dans la zone blanche en dessous du tableau des Info/Response :



#### 10.4 Filtrer les PNJ qui peuvent dire une phrase

Bien. Maintenant que nous avons créé un nouveau Topic, et ajouter des choses dans ce Topic, il faudrait s'assurer que seul Ra'Virr prononce cette phrase.

Imaginer la déconvenue du joueur si tous les PNJ le saluait en lui parlant de ses chaussures... ce ne serait pas sérieux et produirait ce que nous pouvons appeler un mod « sale ».

Heureusement pour nous, il est possible de faire en sorte que seul Ra'Virr prononce cette phrase.

En dessous de tout ce que nous venons de voir, il y a une zone qui s'appelle « Speaker Condition ». C'est dans cette zone que vous allez placer des filtres successifs qui vont permettre d'éviter que tous les pnj du jeu puisse accéder à cette phrase.

Il faut faire attention quand même lorsque l'on choisit les conditions. Il ne faut pas hésiter à être très restrictif, en particulier lorsque l'on ajoute une nouvelle phrase (rappelez vous, toujours le plus haut possible dans la liste), en effet, le moteur du jeu considérera qu'un PNJ peut prononcer la phrase si l'ensemble des conditions est vérifiée... (si vous estimez que la seule condition à respecter est d'être dans l'échoppe de Ra'Virr, et qu'un autre moddeurs décide de placer un pnj dans cette même échoppe, le pnj pourrait prononcer la phrase...)

Dans notre cas nous appliquons le même filtre que celui que nous avons pris pour les salutations. Sur l'ID du pnj, nous choisissons Ra'Virr.

Dans ce cas spécifique, où nous avons créé le Topic depuis 0, cela n'a pas beaucoup d'importance. Mais il vaut mieux prendre de bonnes habitudes dès le début.

#### 10.5 Poser une question au joueur, lui offrir un choix

Vous l'avez vu, Ra'Virr posera une question au pnj. Il faut donc que le pnj ait le choix de la réponse à donner. Il ne sera pas nécessairement intéresser par la tâche « passionnante » que nous lui confions.

Lorsque l'on pose une question au joueur, il faut lui laisser le choix de réponse. Pour cela on dispose de la fonction **Choice**, que nous pouvons utiliser dans la zone Result de la fenêtre de dialogue.

Dans cette zone, il est possible de donner des instructions, en langage de script, qui seront exécutées au moment où le pnj prononcera la phrases ou les paroles associées.

Nous offrons un choix au PJ de la façon suivante :

The screenshot shows a script editor window. At the top, there are dropdown menus for 'Cell', 'PC Faction', and 'PC Rank'. Below these, a 'Result' field contains the text: `choice "OK. Je vais les chercher" 1 "J'ai autre chose à faire" 2`. The numbers 1 and 2 represent the two choices offered to the player.

Faites attention aux guillemets et aux nombres, et surtout à leur séquence.

L'instruction fera apparaître deux alternatives pour le joueur, dans l'une le PJ accepte d'aider Ra'Virr, et dans l'autre non. Les deux propositions apparaîtront en blanc en bas de la fenêtre de dialogue du jeu.

#### 10.6 Interpréter un choix du joueur

Et maintenant que se passera t'il lorsque le joueur fera son choix?

Nous voulons que Ra'Virr soit content et donne la quête (oui je sais, c'est exagéré) au joueur, si celui-ci accepte de l'aider, sinon le joueur baissera dans l'estime de Ra'Virr.

Ra'Virr aura donc deux réactions en fonction de la réponse du joueur. On les ajoute dans la zone Info/Response de la fenêtre de la même façon que l'on ajouté la demande de Ra'Virr...

On filtre les nouvelles phrases de sorte que seul Ra'Virr puisse les prononcer. Comme ce ne sera pas suffisant, on modifie le filtre, en ajoutant l'interprétation de la fonction Choice dans la colonne de droite :

The screenshot shows the 'Speaker Condition' window. The 'ID' is set to 'ra'virr'. The 'Function/Variable' section has 'Function' set to 'Choice' and the value set to '1'. Other fields like 'Race', 'Class', 'Faction', 'Rank', 'Cell', 'PC Faction', and 'PC Rank' are empty. The 'Disp' field is set to '0'.

(Lorsque le PJ accepte de chercher les chaussures)

Cela signifie que si le joueur a fait le choix « 1 », Ra'Virr le remerciera..

The screenshot shows the 'Speaker Condition' window. The 'ID' is set to 'ra'virr'. The 'Function/Variable' section has 'Function' set to 'Choice' and the value set to '2'. Other fields like 'Race', 'Class', 'Faction', 'Rank', 'Cell', 'PC Faction', and 'PC Rank' are empty. The 'Disp' field is set to '0'.

(Lorsque le joueur refuse)

Sinon, Ra'Virr sera déçu.

Chacun des choix du joueur aura des conséquences, celles-ci sont explicitées dans la zone Result.

Si le joueur fait le choix 2 (refus d'aider Ra'Virr), il baissera dans son estime :

Ajouter

ModDisposition -5

dans la zone Result.

Pour le reste, nous allons faire en sorte d'inscrire l'évènement dans le journal.

#### 10.7 *Ecrire dans le journal*

Pour écrire dans le journal, il existe une instruction très simple:

Journal ID\_journal index

Pour cette instruction, vous ajoutez une entrée dans le journal. Mais, comme ça c'est peut-être un peu brutal...

Oui ? Alors on va rentrer dans les détails.

Pour accéder aux entrées du journal, on clique sur l'onglet journal en haut à gauche.

##### 10.7.1 La structure du journal

Tout d'abord, nous allons analyser un peu la structure du journal.

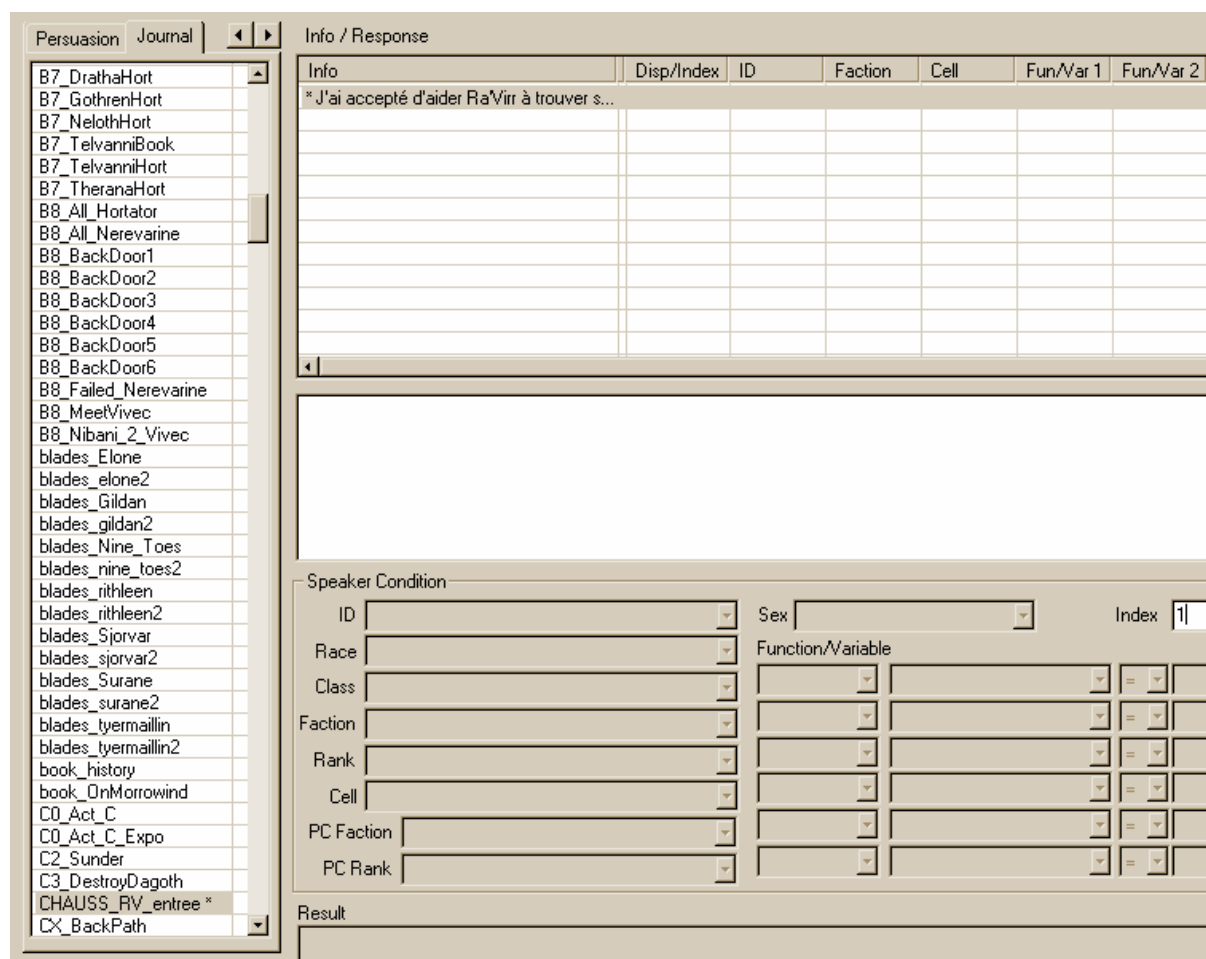
Le journal se compose d'entrée, des thèmes du journal : c'est ce qu'on trouve dans la colonne à gauche de la fenêtre.

Pour y ajouter un thème (une ID), il faut que les entrées ne soient pas filtrées (c'est-à-dire aucune sélection dans la boîte en bas à gauche).

A chaque thème, on peut associer plusieurs éléments. Ils apparaissent dans le grand tableau à droite, et sont identifiés par des Index (en bas à droite du tableau). Si il n'y a qu'un seul élément pour un thème, il doit obligatoirement prendre la valeur 1, si il y en a plusieurs, les valeurs prises doivent être différentes.

L'ID (ou l'intitulé du thème) n'est jamais vue par le joueur, de sorte que vous pouvez choisir un nom exotique (par exemple : CHAUS\_RV\_entree). Le joueur ne verra que les éléments dans son journal.

Par exemple :



### 10.7.2 Ajouter une entrée dans le journal

Pour ajouter une nouvelle entrée dans le journal, on procède de la même façon que lorsque l'on désire ajouter un nouveau Topic.

Il faut que les dialogues ne soient pas filtrés. On peut alors faire un clic-droit dans la liste des Id du journal et sélectionné l'item 'New'. Vous pouvez alors entrer une nouvelle ID, rappelez vous que celle-ci ne sera jamais connue du joueur. Vous pouvez donc choisir une ID aussi exotique que possible.

Je vous conseille tout de même de ne pas choisir quelque chose de complètement aberrant. Essayez, par exemple de respecter le même système que pour les ID des objets.

Une fois que vous avez choisit une ID (un thème) pour les entrées du journal, il faut les éditer. Pour ce faire, il faut effectuer les mêmes séries d'action que lorsque vous voulez ajouter un nouveau dialogue dans un Topic.

Il faut toutefois faire attention à bien spécifier un Index pour chaque entrée que vous créer dans le journal. Si vous n'ajoutez qu'une unique entrée pour un thème, il faut que celle-ci ait au moins l'index 1. En revanche, si vous devez ajouter plus d'une entrée pour une ID, faites attention à leur attribuer des index différents.

### 10.7.3 La faire apparaître dans le journal du joueur

L'instruction

**Journal ID\_journal index**

Permet de faire apparaître, l'entrée correspondant à l'index 'index' du thème 'ID\_journal'.

## 10.8 Faire évoluer la quête

Maintenant que nous avons mis en place les différents éléments de la quête, il faut s'arranger pour conserver une cohérence dans les dialogues de Ra'Virr.

Que diriez vous si Ra'Virr devait vous saluer toujours de la même manière ? Et commencer toujours la conversation de la même façon ?

Pour éviter ce genre d'inconvénients, il faut affiner les filtres que nous avons mis en place pour les salutations et pour les dialogues successifs.

Nous allons maintenant faire un test sur l'état interne du journal, il faut s'assurer que le PJ n'a pas déjà accepté la mission de Ra'Virr. C'est-à-dire que l'index courant du journal pour la quête des chaussures n'est pas 1.

[illegible]

On procède de même pour chacune des paroles prononcées par Ra'Virr.

## 10.9 Finaliser la quête

Et maintenant que nous avons ramené les chaussures à Ra'Virr, qu'allons nous faire.

Tout d'abord, il faut avoir créer une entrée dans le journal, sous l'ID que vous venez de créer (CHAUSS\_RV\_entree) qui correspond à l'évènement où vous rapportez les chaussures à Ra'Virr.

Supposons que vous lui ayez attribuer l'index 10.

Une fois que vous allez trouver les chaussures de Ra'Virr, vous lui rapportez. Cela provoquera un dialogue différent, pendant lequel vous rendrez les chaussures à Ra'Virr. Cette action fera apparaître une nouvelle entrée dans le journal. Celle qui correspond au remerciement de Ra'Virr.

Il faudra également modifier les filtres des différents dialogues de Ra'Virr pour prendre en compte le retour des chaussures. De la même façon que vous aviez évité les répétitions des dialogues de salutations pour vous donner la quête.

### 10.10 Derniers conseils

N'oubliez pas de vérifier toujours que les dialogues que vous avez sélectionnés sont valides. N'oubliez pas de cliquer le bouton Check Error Results.

Ceci vous permettra de vérifier que les dialogues sont valides et ne que toutes les actions codées dans Result sont valides.

Faites attention à la cohérence des dialogues que vous construisez. Ils sont la saveur de votre mod.

Enfin, faites en sorte de toujours positionner les dialogues les plus restrictifs par les conditions (dans speaker condition) le plus haut possible dans la liste des paroles prononcées à propos d'un sujet spécifique.

## 11 Scripting – les scripts

### 11.1 *Présentation générale*

A ce point de la lecture, vous savez déjà faire beaucoup de choses. Mais il vous manque encore une chose qui apportera la vraie vie dans le monde et qui augmentera l'interactivité entre le joueur et les pnj, voire même le monde : les scripts.

Vous avez du le voir, de nombreuses fenêtres de propriété permettent de faire référence à des scripts. Les objets, portes pnj ou créatures peuvent avoir un comportement particulier qui est spécifié par des scripts.

Les possibilités offertes par les scripts sont multiples. Pour donner quelques exemples, ils vous permettent d'ajouter ou de retirer des sujets de conversation pendant le déroulement du jeu, de modifier le son émis par un élément, de créer des quêtes, de faire apparaître des personnages ou des créatures sous certaines conditions, de modifier les inventaires des personnages.

Très simplement les scripts sont des petits morceaux de code simplifiés.

Pour ceux qui ont une expérience dans le domaine de la programmation, le nom de script est usurpé, puisque les scripts doivent être compilés, et ce n'est pas le cas ici. Mais vous nous pardonnerez cet abus de langage.

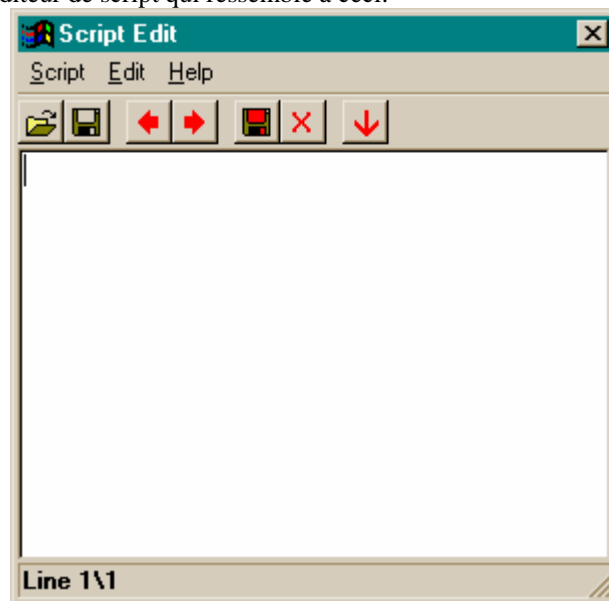
Si vous voulez créer des scripts, il vaut mieux connaître quelques langages de programmation, mais ce n'est pas une condition nécessaire.

Il existe deux moyens d'accéder à l'éditeur de Script pour modifier des scripts ou en créer de nouveaux.

la première est par la fenêtre d'édition de propriétés, et sélectionner le bouton "... " qui se trouve à côté de la zone nommée script

la seconde passe par les menus "Gameplay / Edit Scripts..."

L'une et l'autre ouvrent l'éditeur de script qui ressemble à ceci:



Outre le fait qu'il s'agit ni plus ni moins que d'un vulgaire éditeur de texte, les fonctionnalités en sont les suivantes :

**Script/New...** permet de créer un nouveau script

**Script/Open...** ouvre un script déjà existant

**Script/Previous Script** affiche le script précédent en fonction de l'ordre alphabétique (si vous préfixez vos scripts avec un identifiant défini, vous naviguerez plus facilement dans l'éditeur)

**Script/Next Script** affiche le script suivant dans la liste en fonction de l'ordre alphabétique.

**Script/Save** sauvegarde le script courant.

**Script/Recompile All** cette commande permet de compiler tous les scripts existants (ceci doit être fait avant que les modifications apportées ne soient prises en compte).

**Script/Delete** efface le script courant.

**Script/Exit** ferme l'éditeur de script.

**Edit/Undo** annule la dernière action effectuée dans l'éditeur.

**Edit/Redo** permet de refaire une action annulée.

**Help/Contents** ouvre le chapitre approprié dans l'aide du TESCS.

**Help/Commands** ouvre le chapitre approprié dans l'aide en ligne du TESCS.

**Help/Functions** ouvre le chapitre associé dans l'aide en ligne du TESCS.

## 11.2 *Le TESCS et la programmation Orientée Objet*

Dans cette partie nous abordons quelques concepts qui nous semblent intéressants pour tout moddeur. Sachez que la lecture des informations données ici n'est absolument pas nécessaire pour la suite, ce sera un plus. Cette partie est présente uniquement pour la culture. Elle vous aidera à comprendre les concepts manipulés dans le TESCS et l'architecture sous jacente à tout ce que vous pouvez voir.

Pour les scripteurs, elle pourra les aider à mieux comprendre ce qu'ils font et les raisons pour lesquelles les choses doivent être faites d'une certaine façon.

Le développeur habitué aux concepts d'objets pourra aisément se passer de la lecture de la première partie qui présente rapidement les concepts sous jacents à la POO (Programmation Orientée Objet), et trouvera plus intéressant la seconde partie qui raccorde le TESCS et MW à cela.

### 11.2.1 *Quelques concept d'Objet*

Je tiens ici à signaler que ce chapitre s'inspire fortement du tutorial de script produit par Daederius, qui explique d'une manière extrêmement didactique les concepts de la programmation orientée objet.

En informatique on distingue de nombreux types de langages ; les plus couramment rencontrés aujourd'hui sont les langages procéduraux (comme le C) et les langages Objets (C++ ou JAVA par exemple).

Les premiers repose sur le concept de procédures, des fonctions qui s'appellent les unes les autres. Les seconds mettent en communication des Structures complexes qui sont définies par des données et les traitements que l'on peut leur appliquer.

Un objet est une sorte de définition générale d'un type de comportement, et la matérialisation de cet objet s'appelle une instance.

Dans l'univers de MW, les références qui sont manipulées sont en fait des instances d'un objet.

Mais qu'est ce qu'un objet.

Il s'agit d'un type, qui dispose d'un certain nombre de caractéristiques, des variables, et d'un certain nombre de comportements (des méthodes).

NOM\_DE\_L\_OBJET

..... ; des variables

..... ; des méthodes

FIN

Prenons l'exemple d'une arme :

Objet\_Arme

ID IDnameArme ; c'est l'ID unique pour ce type d'arme

String VarName ; le nom dans le jeu

short poids ; le poids de l'objet

short degats ; les degats qu'elle inflige

image modèle ; le modèle que l'on voit dans le jeu

fonction Attaque() ; utiliser pour l'attaque

fonction Esquive()

fonction reparerArme()

...

FIN Objet\_ARME

Il s'agit d'une sorte de patron que tous les objets armes du jeu respectent. Tous attribuent une valeur différentes aux variables et tous ont des comportements différents.

Les variables des objets sont choisies par vous dans les différentes interfaces qui vous permettent d'ajouter des types d'armes, de coffre...

Vous retrouvez cette structure dans le TESCS, dans les différents tableaux de l'interface et dans les zones de créations des objets.

Rassurez vous, vous n'avez pas à créer de nouveaux objets cela aurait demandé un trop haut niveau de compétence informatique pour que tout le monde puisse faire des mods. Et le TESCS n'aurait pas pu être ce qu'il est (un outil puissant et presque convivial).

Et les fonctions ? Et bien les fonctions sont justement les fonctions que vous utilisez dans les langages de script. Lorsque vous utilisez la ligne suivante :

`Player->AddItem Gold_001 75`

Vous utilisez en fait la fonction AddItem de l'objet Player avec les paramètres **Gold\_001** et **75**. Il est possible d'accéder aux variables des objets par un POINT (.)

`Player.strength`

Ou par l'appel à la méthode

`Player->GetStrength...`

### 11.2.2 Le TESCS et les objets

Le TESCS manipule des objets. Vous pouvez créer de nouveaux type pour chaque objet –PNJ, Weapon.... Mais vous ne pouvez pas créer de nouveaux objets ou de nouvelles fonctions pour les objets existants ( C'est dommage mais bon...)

Lorsque vous faites un drag'n drop d'un type d'objet vers la fenêtre de rendu, vous créez une nouvelle instance de cet objet, le TESCS l'appelle Référence. Cette référence partage un certain nombre d'information avec toutes les références du jeu, et dispose d'un certain nombre d'information qui lui sont propres.

Parmi les variables partagées, on peut trouver les scripts, ATTENTION donc quand vous attachez un script à un objet vous l'attachez également à toutes les références existantes dans le jeu, il y a également le fichier modèle... etc. Parmi les variables non partagées, il y a notamment la position X Y Z de l'objet, ses angles de rotations et parfois quelques autres données.

## 11.3 Deux ou trois choses à savoir

### 11.3.1 L'exécution des script

Un script s'exécute:

- 1** toujours s'il s'agit d'un script global
- 2** si le PJ se trouve dans la Cellule dans laquelle se trouve l'objet auquel le script est attaché
- 3** si le PJ a l'objet en sa possession

L'ensemble des scripts s'exécutent tous pour chaque image (comme il y en a plusieurs par seconde, vos scripts s'exécutent donc plusieurs fois par seconde). Comme cela est consommateur en temps CPU, je vous demande d'épargner les configurations peu solides et faites en sorte que vos scripts s'exécutent un nombre minimum de fois ou soit rapide si ils n'ont rien à faire.

Nous verrons plus loin comment éviter qu'un script ne s'exécute trop souvent, ou trop longtemps.

Ne vous inquiétez pas pour les fonctions d'animation, les vitesses sont toujours données en unité par seconde... le moteur de rendu fait les calculs nécessaires.

### 11.3.2 Les variables

Les variables locales au démarrage d'un script sont toujours nulles. Mais, les initialisations ne sont pas des considérées comme des instructions au même titre que le reste du script. Lorsqu'un script a été exécuté une fois, il ne retourne jamais dans la zone d'initialisation. (Cf ; nombre de fois).

En revanche, les commandes stopScript puis startScript impose de repasser dans les zones d'initialisation....

## 11.4 Conventions de base

Tous les scripts doivent suivre certaines conventions. Tout d'abord, l'apparence générale d'un script est la suivante.

`Begin [ScriptName]`

`[Instructions]`

```
End [ScriptName]
```

**[ScriptName]** doit être remplacé par le nom que vous désirez donner au script. Le nom d'un script doit être unique, toujours. Le script "Main" est toujours exécuté à partir du moment où le jeu commence.

Les **[Instructions]** peuvent avoir plusieurs formats. D'une manière générale, les instructions peuvent créer des variables contenant des informations à utiliser plus tard, attribuer des valeurs aux variables, vérifier certaines conditions ou activer des fonctions prédéfinies.

Chaque instruction est associée à une ligne. Lors de l'activation du script, les instructions sont effectuées les unes après les autres. Par opposition avec la plupart des langages connus, la fin d'une ligne n'est pas marquée par un point-virgule.

Les variables peuvent être de trois types différents:

short: ce type est également connu comme entier (int ou integer) dans beaucoup d'autres langages. Les variables de ce type peuvent prendre des valeurs comprises entre  $2^{15}$  et  $2^{15}-1$  (-32768 to 32767).

long c'est le type également connu sous le même nom dans les langages comme le C ou le C++, la valeur est entière et est comprise entre  $-2^{31}$  et  $2^{31}$  (-2,147,483,648 to 2,147,183,647).

float c'est le type de grandeur utilisée pour les valeurs réelles comprises entre  $3.4e-38$  et  $3.4e+38$  (environ -28.75784178 et 47.24215822).

#### 11.4.1 Assigner une valeur à une variable

L'assignation d'une valeur à une variable se fait avec la fonction:

```
set [VariableName] to [Value].
```

**[VariableName]** corresponde au nom de la variable dont la valeur doit être change.

**[Value]** peut être une valeur, une autre variable ou le résultat d'une évaluation numérique (le calcul doit être entre parenthèses : par exemple, set damage to (a\*b)).

Pour récupérer la valeur d'une variable que vous utilisez :

```
get [VariableName]
```

Il est possible de modifier les variables locales d'un autre script. Il y a toutefois quelques précautions à prendre :

- le script que l'on veut modifier doit être attaché à un objet dont vous connaissez l'ID
- l'objet doit être unique pour que cela puisse fonctionner, en effet le moteur du jeu va chercher la première référence qu'il trouve sous cette ID.
- l'objet doit avoir un script attaché, sinon cela ne marchera pas

```
set ObjectID.variable to 100
```

Attention, l'inverse ne marche pas... **set local\_variable to ObjectID.variable**

En revanche, l'instruction suivante semble donner des résultats :

```
if(ObjectID.variable > 10)
```

#### 11.4.2 Utiliser des valeurs aléatoires

L'utilisation de valeur aléatoire est très fréquente, et donne une apparence plus vivante à votre univers. Pour assigner une valeur aléatoire à une variable :

```
Set ma_variable to Random 50
```

La fonction Random retourne un nombre aléatoire compris entre 0 et la valeur que vous donné en paramètre. Dans ce cas, j'attribue à ma\_variable une valeur comprise entre 0 et 49.

#### 11.4.3 Utiliser les fonctions

Pour utiliser une fonction sur un objet spécifique, on utilise la syntaxe :

```
'ObjectID' -> MaFonction [Paramètres]
```

Cela signifie que l'on appelle la fonction MaFonction sur l'objet 'ObjectID'.

Les paramètres de la fonction sont (ou non) séparés par des virgules, mais en tout cas par des espaces.

Retenez bien que les instructions de script doivent toujours tenir sur une ligne. Si une fonction a besoin de nombreux paramètres, n'allez pas à la ligne sous prétexte de rendre le script plus lisible.

Remarque : certaines fonctions acceptent de prendre des variables en paramètres d'autres non... donc méfiance en utilisant des variables en paramètres des fonctions.

Si ObjectID n'est pas précisé dans un script local, la fonction est appelée sur l'objet auquel le script est attaché.

#### 11.4.4 Virgules, espaces et parenthèses

La syntaxe de l'éditeur n'est pas trop rigide, et les tests effectués par le compilateur ne sont pas très rigoureux. Faites donc attention aux points suivants, qui pourraient provoquer des erreurs difficiles à détecter.

- 4** Préférer les virgules aux espaces pour séparer les paramètres des fonctions
- 5** Si une ID contient des espaces, entourée la de guillemets (mais essayez autant que possible d'éviter de créer des ID avec des espaces)
- 6** Laisser un espace après les parenthèses, les opérateurs et les noms de variables, il arrive que cela produise pour une raison inconnue des comportements étranges.

#### 11.4.5 Les tests if

Pour tester la valeur de certaines conditions, il faut utiliser la syntaxe suivante:

```
if([VariableNameOrValue][Operator][VariableNameOrValue])  
  
[Orders]  
  
endif
```

[VariableNameOrValue] est remplacé soit par une variable, soit par une valeur, mais il est également possible d'introduire une opération qui donne une valeur.

[Operator] correspond aux opérateurs logiques, de comparaison. Ils retournent un "booléen". Si le résultat du booléen est vrai, les instructions dans les blocs suivants seront exécutées, dans le cas contraire, les instructions seront ignorées.

Les opérateurs de comparaisons sont les suivants:

== donne "true" si les deux membres de l'égalité sont identiques.  
!= donne "true" si les deux membres de la comparaison sont différents.  
< donne "true" si la valeur à gauche est inférieure à la valeur à droite.  
> donne "true" si la valeur à gauche est supérieurs à celle de droite.  
<= donne "true" si la valeur à gauche est inférieure ou égale à la valeur de droite.  
>= returns "true" si la valeur à gauche est supérieure ou égale à la valeur de droite.

Vous pouvez ajouter d'autres tests dans les Instructions. **Mais il est impossible d'utiliser des combinaisons booléennes comme AND, OR, ou XOR. Vous devrez donc construire des structures imbriquées de tests, mais dans la limite de 10 tests imbriqués.** Il existe une astuce pour programmer des tests avec des combinaisons.

La structure suivante est également autorisée et permet des tests complexes sur les valeurs des variables.

```
if([VariableNameOrValue][Operator][VariableNameOrValue])  
  
[Orders]  
  
elseif([VariableNameOrValue][Operator][VariableNameOrValue])  
  
[Orders]  
  
else
```

```
[Orders]
```

```
endif
```

**elseif** permet de vérifier la validité d'une autre condition, si la première n'est pas validée.

**else** est active si aucune des conditions testées auparavant n'est validée avant. Else et Elseif sont utilisables séparément.

#### 11.4.6 Le test while (tant que)

Il est également possible d'utiliser des boucles, tant que la condition est validée, les instructions sont exécutées.

```
while([VariableNameOrValue][Operator][VariableNameOrValue])
```

```
[Orders]
```

```
endwhile
```

La boucle while permet de réaliser un ensemble d'opération autant de fois qu'il est nécessaire pour valider une condition. Puisqu'elle est dans un script, elle est réalisée pour une image. Cela permet d'aller plus vite pour certaines actions.

**MAIS ATTENTION, SI UNE CONDITION N'EST JAMAIS VALIDÉE, LE JEU N'AVANCERA PLUS ET SERA COMPLETEMENT PARALYSÉ....**

#### 11.4.7 Les commentaires

Vous pouvez ajouter des commentaires après un point virgule

```
; commentaires
```

**NB:** une unique référence peut être utilisée par ligne d'instruction.

### 11.5 Les opérateurs booléens

Le TESCS ne permet pas de faire des opérations booléennes dans les tests. Ainsi, les opérateurs ET, OU ne fonctionnent pas. Toutefois, on peut se débrouiller pour retrouver leur équivalent.

Le test

```
if(variable1== 1 ET variable2==1)
```

```
    Blablabla
```

```
endif
```

Se transcrit sous la forme

```
short boolean
```

```
if(variable1 == 1)
```

```
    if(variable2 == 1)
```

```
        set boolean to 1
```

```
    endif
```

```
endif
```

```
if(boolean == 1 )
```

```
    blablabla
```

```
endif
```

ou bien

```
if(variable1 == 1)
```

```
    if(variable2 == 1)
```

```

blabla
endif
endif
    
```

et ainsi de suite. Cela risque d'allonger considérablement vos scripts....  
Et garder en mémoire que la profondeur de ces structures ne soit pas dépasser 10 tests...

### 11.6 Scripts locaux – scripts globaux

Il existe deux sortes de scripts dans l'univers de MW. Les scripts locaux et les scripts globaux. La différence entre les deux tient au fait que les scripts locaux sont toujours associés à un objet de l'univers, alors que les scripts globaux, eux, ne sont attachés à aucun objet.

L'autre différence réside dans le mode d'exécution des scripts. Les scripts globaux ne sont pas exécutés par défaut. Il faut utiliser une commande spéciale pour les lancer StartScript nom\_du\_script. En revanche une fois qu'ils ont été démarrés, ils sont toujours actifs et sont évalués pour chaque image, et ils sont relancés à chaque reprise de jeu. Il est possible de les interrompre par la méthode StopScript nom\_du\_script.

Les scripts globaux sont dangereux car si on oublie de les interrompre, ils continuent à utiliser ressources, et ceci même si vous « retournez » tout de suite. Si vous ne voulez les utiliser qu'une fois, n'oubliez pas de les stopper.

Les scripts locaux en revanche s'exécutent sans qu'il soit nécessaire de les démarrer. Ils sont exécutés dès que la zone qui contient l'objet auquel ils sont attachés est chargée.

Il est possible de faire en sorte qu'un script 'local' s'exécute toujours, par exemple en l'attachant à un objet que le joueur garde sur lui.

Mais alors comment lancé un script global vous demandez vous ?

Et c'est là que je vous réponds qu'il faut faire très attention avec la méthode choisie pour démarrer un script global. En effet, Bethesda nous a laissé l'accès au script main, et vous allez sans doute vous dire...« Ben voilà , je rajoute une ligne qui démarre mon script, dans le principal et le tour est joué ! »

SURTOUT PAS, cela est la cause d'une incompatibilité entre mods. Imaginez que tout le monde fasse comme vous, il n'y aurait qu'un seul script lancé, celui qui est lancé par le dernier mod chargé par le moteur.

Rassurez vous, il existe un artifice élégant et fiable pour lancer malgré tout votre script global :

Vous attaché un script local, à un objet et vous signalez que le joueur doit se rendre à coté de cet objet pour que le mod soit actif. L'objet en question n'a pas besoin d'être visible pour le joueur, caché le dans une pièce, ou bien attachez le à une source lumineuse.

Un autre artifice est de demandé au joueur d'aller parler à un PNJ, et dans les Result du dialogue avec ce PNJ, vous lancer votre script principal. Mais pour l'amour d'Azura, ne modifier pas le script principal.

#### 11.6.1 Modification de version d'un mod et script global

Si vous utilisez un mod qui a un script global, il faut prendre certaines précautions avant de changer la version utilisée.

Il faut d'abord désinstaller le plug puis lancer le jeu, faire une sauvegarde, installez la nouvelle version et relancer le jeu.

### 11.7 Du bon usage du bouton Save de la fenêtre d'édition

Le bouton Save que vous trouvez dans la barre d'outils de la fenêtre d'édition de script permet de prendre en compte les modifications que vous avez fait sur le script sur lequel vous travaillez.

Le script est alors compilé et des messages d'erreur vous sont adressés si des erreurs ont été trouvés.

ATTENTION : ce bouton n'enregistre pas les modifications dans le plug courant. Le script n'est enregistré dans le plug que lorsque vous cliquez sur le bouton Save de la fenêtre principale de l'éditeur.

Lorsque vous travaillez sur de gros scripts pensez à enregistrer les modifs au niveau du plug, histoire de ne pas avoir de mauvaises surprises si le TESCS plante....

### 11.8 Les fonctions reconnues dans les scripts

En annexe à ce document, vous trouverez un tableau récapitulatif des fonctions reconnues dans les scripts. Ce tableau décrit leur syntaxe, ce qu'elles font et donnent des exemples d'utilisation.

Avant d’aller plus loin, je voudrais vous présenter un concept qui vous servira dans le développement de vos scripts : la machine à état.

### 11.9 La machine à états

La machine à états est un des concepts de bases pour décrire le fonctionnement de la quasi totalité des programmes, et en particulier celui des ordinateurs.

Une machine à états est un système qui est décrit par un nombre fini d’état. Un état peut être dans notre cas la valeur d’une variable, ou un ensemble de valeurs pour un jeu de variables. La machine à états est définie par ses états et par les transitions possibles entre chacun de ses états. Elle a donc un comportement complètement déterministe et liée à l’évolution des variables qui la caractérisent.

Quel rapport entre les machines à états et les scripts dans MW ?

Si vous regardez les scripts existants dans le jeu, ou ceux développés par les moddeurs, vous vous apercevrez très vite qu’ils ont presque tous une structure commune. Cette structure associe un (ou des grandeurs) et des séries de test sur la valeur de ces variables.

```
Begin MaMachine
```

```
short state
```

```
if(state == 0)
```

```
blablabla
```

```
set state to 1
```

```
else if (state = 1)
```

```
bliblibli
```

```
set state to 2
```

```
....etc
```

```
etc
```

```
endif
```

Le fonctionnement de ces scripts est alors assez simple à comprendre, la variable state caractérise l’état de la machine. C’est une sorte d’indicateur de position. Puis le script fait un premier test sur la valeur de state, comme lors de la première exécution du script les variables sont initialisées à 0, le premier test est vérifié.

On exécute donc les actions **blablabla** et on change la machine d’état. De sorte que à la prochaine exécution elle ne fera que les instructions **bliblibli**. Etc...

Si vous comprenez les principes de la machine à états, et la façon d’en réaliser une dans MW, vous pourrez faire des scripts très puissants et très riches (malheureusement, le texte en sera très long, du fait qu’il n’y a pas de possibilités d’utiliser les opérateurs booléen AND, OR ou XOR dans les tests.

Il y a encore quelques astuces à maîtriser pour éviter de faire des scripts trop gourmands.

### 11.10 Des astuces et des choses à connaître pour faire des scripts

Il y a trois astuces de bases à connaître pour écrire un bon script dans MW. Ces trois astuces sont nécessaires afin d’éviter que les scripts que vous allez écrire ne viennent complètement ralentir l’exécution du jeu. En effet, comme nous ‘avons vu plus haut, les scripts sont exécutés pour chaque image calculée, si votre machine tourne à 30 fps, ils seront donc exécutés 30 fois par seconde. Mais si votre machine tourne à 30 fps, vous avez de la chance, ce n’est pas le cas pour tout le monde. Et pour peu que vous créiez un mod avec de nombreux scripts dans une zone, vous pourriez perdre quelques fps, si vous ne prenez pas certaines précautions.

Je n’ose pas écrire le mot règle pour ces astuces, mais certaines en sont presque.

#### 11.10.1 Compter le nombre d’exécutions

Il est souvent utile de connaître le nombre d’exécution pour un script donné. En effet, il y a de nombreux cas où nous ne désirons qu’une seule exécution d’un script donné.

Prenons l'exemple d'une porte magique, elle ne s'ouvrira que si le joueur trouve le mot de passe. Et nous sommes particulièrement vicieux, nous n'accordons qu'une chance au joueur.

Lorsqu'il essaiera d'ouvrir la porte on fera apparaître le message suivant :

« Connaissez vous le mot de passe ? », le joueur pourra répondre par oui ou non. Dans un premier temps, on fera confiance au joueur qui réponds oui et on lui ouvrira. Si il répond non...tant pis pour lui.

L'intérêt de cet exemple, est que si le joueur répond non, on ne veut plus lui ouvrir la porte. Il faut donc empêcher que le script ne s'exécute à nouveau.

Pour cela nous allons utiliser une variable de contrôle qui permettra de connaître le nombre d'exécution du script.

#### Begin MagicalDoor

short controlvar

```
if(controlvar == 1)
    return
endif
```

```
if(OnActivate == 1)
```

```
    MessageBox « Connaissez vous le mot de passe ? » « oui » « non »
```

```
    Set controlvar to 1
```

```
endif
```

```
...
```

```
end
```

Le fonctionnement de ce script est le suivant, pour chaque image, on regarde si on a déjà exécuté la suite du script, si c'est le cas, on signale la fin d'exécution du script (instruction return), si ce n'est pas le cas, on regarde si le joueur a activé la porte, si c'est le cas on lui pose la question, et on fait basculer la variable de contrôle à 1. Ainsi, lors des prochaines exécutions du script on retournera tout de suite, sans faire la suite du code.

NB : je vous conseille de toujours faire le plus tôt possible les tests permettant de déterminer si vous devez ou non faire des choses dans ce script. Les instructions **return** devraient dans la majorité des cas se trouver tout en haut de votre code, de façon à favoriser la rapidité de calcul des images.

#### 11.10.2 Les chronomètres

Il existe de nombreux cas dans lesquels vous désirez qu'un groupe d'instruction s'exécute à intervalles réguliers, mais pas pour toutes les images. Si vous désirez, par exemple, répéter une phrase au joueur toutes les 10 secondes, il ne faut pas que cette phrase s'affiche pour chaque image. Comme il n'est pas possible d'utiliser le même genre d'astuce que précédemment (un compteur) il faut trouver autre chose.

Il y a une fonction qui donne temps écoulé depuis la dernière exécution du script, rappelez vous que le script est exécuté à chaque image, cette valeur peut donc être très petite, ce sera une variable de type float.

Imaginons que vous avez créé un PNJ qui est un compagnon pour le joueur. Vous lui donnez un comportement très poussé et vous voulez qu'il informe le joueur qu'il a besoin de soins lorsque son niveau de santé est inférieur à 25% du maximum. Cette information passerait par une boîte de dialogue informant le joueur de la mort prochaine de son compagnon, mais il est particulièrement désagréable d'avoir toujours le même message à l'écran...le PNJ informera le joueur toutes les 5 secondes par exemple.

L'usage de la fonction **GetSecondsPassed** permet de connaître le temps écoulé depuis la dernière exécution du script.

#### Begin NeedHealing

float chrono

```
if( GetHealthGetRatio > 0.25)
```

```
    return
endif
```

```
set chrono to chrono+GetSecondsPassed
```

```
if( chrono >= 5)
```

```
    MessageBox "J'ai besoin d'être soigné"
```

```

    Set chrono to 0
Endif
end

```

Ici nous utilisons la variable chrono qui permet de chronométrer le temps total. C'est une sorte de compteur, on le fait évoluer chaque fois que l'on exécute le script.

Vous noterez que là encore, nous avons fait le test qui permet de cesser l'exécution du script au tout début, si le PNJ est en bonne santé on ne fait rien.

Si il n'est pas en bonne santé, alors on débute on incrémente chrono.

Lorsque ce chrono atteint la valeur de 5 secondes, on répète au joueur la demande de soin, puis on réinitialise le chrono.

### 11.10.3 Si le joueur est dans le mode menu

Dans le cas de l'exemple précédent, on peut encore améliorer la chose. En effet, lorsque le joueur regarde son inventaire ou la carte, lorsqu'il utilise une fenêtre de menu, le script s'exécute toujours, et oui, le joueur regarde son inventaire mais le moteur du jeu continue de calculer les images, et dans le même temps d'exécuter les scripts actifs. Il ne veut pas être ennuyé par ces messages alors qu'il consulte son inventaire.... Il ne faut donc pas lui en afficher.

Un autre exemple, vous réaliser un script qui piège une porte, et qui inflige des dégats pendant 10 secondes au joueur qui essaye de l'ouvrir, si il passe dans le mode menu, il n'encaissera pas de dégats mais votre script continuera de tourner, si le joueur reste dans le mode menu pendant suffisamment longtemps, il ne subira aucun dommage...

Il existe une instruction qui permet de savoir si le joueur est actuellement dans le menu : MenuMode, cette fonction retourne 1 si le joueur est dans le mode menu et 0 sinon. Il est donc facile de ne pas exécuter d'instructions si le joueur est dans le menu...

```

Begin NeedHealing
float chrono
if(MenuMode ==1)
return
endif
if( GetHealthGetRatio > 0.25)
return
endif
set chrono to chrono+GetSecondsPassed
if( chrono >= 5)
    MessageBox "J'ai besoin d'être soigné"
    Set chrono to 0
Endif
end

```

Vous remarquerez que c'est le premier test que je fais...

Si le joueur est dans le menu, je cesse l'exécution du script. Quasiment tous les scripts que vous allez créer devraient commencer de cette manière, en effet les scripts que vous rendez actifs pendant le menu ralentissent le déplacement de la souris (elle n'est déplacée que pour chaque image calculée). D'autre part, les scripts qui utilisent des instructions à effectuer à intervalle régulier risque de se trouver perturbés si ils sont actifs lorsque le joueur est dans le menu.

### **11.11 Débugger un script**

Malheureusement, il n'existe pas de débogueur pour le TESCS. Pour vérifier le fonctionnement d'un script il ne vous reste plus que messages et les lignes de commandes de la console.

Si vous avez un doute sur le fonctionnement d'un script, insérez une ligne de commande qui fait apparaître une boîte de message aux points clés de votre script. N'oubliez pas de commenter ces lignes ou de les effacer avant de rendre votre mod disponible.

La ligne de commande dans la console vous sera d'une grande utilité également, la commande sv fait apparaître une liste de toutes les variables globales ainsi que leur valeur.

Pour connaître les variables attribuées à un objet, il faut que celui-ci ait un script qui lui soit attaché. Faites alors apparaître la console, faites un clic droit sur l'objet taper la commande sv, vous visualisez la liste des variables attachées à l'objet.

Maintenant que vous connaissez ces astuces, nous allons pouvoir regarder des scripts classiques, l'utilisation de certaines fonctions.

Je n'ai évidemment pas testé personnellement tous ce que je présente ensuite, mais je l'ai pris de forum (US ou autre) et en partie aussi sur le Morrowind Scripting for Dummies de GhanBuriGhan (un excellent travail sur les scripts).

### 11.12 Les déplacements des PNJ

#### 11.12.1 Faire se déplacer un PNJ vers une nouvelle destination

On utilise la fonction AiTravel :

```
AiTravel, x, y, z, [reset]
```

Les variables x, y, z sont données dans le système de coordonnées du monde. Il est possible de les connaître en déplaçant la caméra dans la zone de Rendu jusqu'à l'endroit qui vous intéresse, ou bien en sélectionnant l'un des points de la grille de déplacement, ou encore en sélectionnant un objet proche de la destination souhaitée, les coordonnées sont alors affichées dans la fenêtre des objets.

L'utilisation de cette fonction requiert absolument de s'assurer qu'elle ne sera appelée qu'une unique fois. Le script suivant ne marchera pas :

```
Begin Voyage
```

```
AiTravel, 1359, 2700, 1045
```

```
End Voyage
```

En effet le script est exécuté à chaque calcul d'image, le package I.A. utilisé est donc constamment réinitialisé, la conséquence directe est que le PNJ ne se déplacera pas. Il faut donc employer une variable qui indiquera que le package I.A. a déjà été initialisé.

```
Begin Voyage
```

```
Short deja_fait
```

```
If (deja_fait==0)
```

```
    AiTravel, 1359, 2700, 1045
```

```
    Set deja_fait to 1
```

```
endif
```

```
End Voyage
```

Ce script fonctionnera, et le PNJ se déplacera jusqu'au point désiré aussitôt que le script sera actif, c'est-à-dire dès l'instant de chargement de la cellule.

#### 11.12.2 Vérifier qu'un PNJ a bien exécuté son mouvement

On utilise une fonction classique GetAIPackageDone

```
GetAIPackageDone    (returns short)
```

```
if ( GetAIPackageDone == 1 )
```

```

[blablabla]
endif

```

La fonction GetAIPackageDone retourne un entier qui vaut 1, lors de l'image qui suit la fin de l'exécution du package courant. Cette fonction associée à celle du dessus va permettre d'enchaîner des déplacements successifs pour le PNJ.

Le script suivant présente un exemple pour enchaîner des déplacements successifs, vous remarquerez que l'on utilise une sorte de machine à états qui permet de vérifier que le PNJ est arrivé ou non à un certain point avant de l'envoyer ailleurs.

```

Begin Voyages
short etat
float chrono

if ( MenuMode == 1 ) ;si on est dans le mode de menu on ne fait rien
    return
endif
;*****debut du deplacement
if ( etat == 0 )
    if ( Player -> GetDistance HB_adros_darani < 5000 ) ; si le joueur n'est pas trop pres
        set etat to 5
    endif
;***** Debut du voyage
elseif ( etat == 5 )
    SetHello 0
    AITravel -8144, -19409, 728 ; coordonnees du point 1
    set etat to 10
elseif ( etat == 10 )
    if ( GetAIPackageDone == 1 ) ; a t on atteint le point 1
        set etat to 40 ; si oui on se prepare pour la prochaine destination
    endif
elseif ( etat == 40 )
    AITravel -9147, -19459, 720 ; en route vers le point 2
    set etat to 50
elseif ( etat == 50 )
    if ( GetAIPackageDone == 1 ) ; a t on atteint le point 2
        set etat to 60
    endif
elseif ( etat == 60 )
    AITravel -8144, -19409, 728 ;on retourne au point 1
    set etat to 70
elseif ( etat == 70 )
    if ( GetAIPackageDone == 1 ) ; le point 1 a ete atteint
        set etat to 80
    endif
elseif ( etat == 80 )
    AITravel -6640, -18496, 1040 vers le point 0
    set etat to 90
elseif ( etat == 90 )
    if ( GetAIPackageDone == 1 ) ; on a atteint le point 0
        set etat to 0 ; on recommence le pelerinage
    endif
endif

End TravelLoop

```

Il y a aussi de bons exemples utilisant AITravel dans le script "lookoutsript" (Fargoth planque son trésor à Seyda Neen) ou encore dans le script CharGenWalkNPC script (c'est le garde qui vous guide dans le bateau au début du jeu).

GhanBuriGhan, qui a développé le mod des marchands ambulants signale qu'il en fait utilisation dans son mod.

Si vous avez l'intention de beaucoup utiliser cette fonction, vous devez connaître quelques limitations. Lorsque vous quittez une cellule, ou bien si vous dormez alors que le PNJ est en train de se déplacer, le script ne détecte pas le signal GetAIPackageDone. La conséquence est que le PNJ est complètement perdu et coincé dans son voyage lorsque que vous revenez dans la cellule ou lorsque vous vous réveillez. Le code suivant (valable pour le script précédent) permet de s'affranchir de cette limitation et de faire en sorte que le script continue.

```
; ***** Sauvegarde du script – on retrouve l'état du script après un repos ou avoir quitter la cellule
If ( Player -> GetDistance, HB_adros_darani < 5000 )
    if (GetCurrentAIPackage == -1) ; si aucun package IA n'est en cours d'exécution
        set timeout to ( timeout + GetSecondsPassed ); utilisation d'un chrono pour se retrouver
        if ( timeout >= 3 ) ; attend un peu, il y a toujours des mouvements qui se produisent
            set etat to (etat - 10) ; on revient en arrière dans l'historique des etats pour se deplacer.
            set timeout to 0
        endif
    else
        set timeout to 0
    endif
endif
```

Note du traducteur: je ne suis pas expert, et je n'ai pas testé la solution proposée, mais il me semble qu'il peut y avoir un problème si au moment où le joueur quitte la cellule, la variable etat est à 0 (elle se retrouverait alors valoir -10, et il n'y a pas d'action associée. A tester donc.

### 11.12.3 Mouvement aléatoire d'un PNJ ou d'une créature

Utilisation de AiWander

```
AiWander, distance, duree, heure, [idle2], [idle3], ...[idle9], [reset]
Actor_ID -> AiWander, 512, 5, 0, 0,20,0,0,10,30,0,0
```

C'est l'algorithme le plus utilisé dans le jeu pour les mouvements du PNJ. Le PNJ se déplace sur la grille de déplacement de manière toute à fait aléatoire (si il n'y a pas de grille, il utilise un algo de détection d'obstacle et se déplace aléatoirement) et il fait des mouvements divers.

- distance: donne la distance maximum dont l'acteur peut s'éloigner de son point d'origine.
- duree: la durée en heures pendant laquelle l'acteur execute ce package, après quoi le package est réinitialiser et un signal GetAIPackageDone est envoyé.
- heure: on suppose que c'est l'heure de début d'exécution du package, ce n'est valable que si on lui attribue une durée.
- [idle2] , ...[idle9] : pourcentage pour que l'acteur effectue les mouvements associés. Les significations des Idle sont les suivantes:
  - Idle2: regarde autour de lui
  - Idle3: regarde derrière
  - Idle4: se gratte la tête
  - Idle5: époussette ses habits ou son armure
  - Idle6: se frotte les mains
  - Idle7: regarde ses doigts puis furtivement autour de lui
  - Idle8: est plongé dans ses pensées ( ???)
  - Idle9: cherche son arme.

Si vous utilisez les Idle, tous les pourcentages doivent être attribué dans l'ordre ci dessus.

Pour faire en sorte qu'un acteur reste à la place qu'on lui a attribuée, on utilise la commande :

```
AiWander, 0, 0, 0
```

### 11.12.4 Faire suivre un Acteur par un autre

Pour faire suivre un acteur par un autre, on utilise le package I.A. AiFollow, qui contient deux fonctions :

```
AiFollow, ActorID, duration, x, y, z, [reset]
AiFollowCell, ActorID, CellID, duration, x, y, z, [reset]
```

```
"MobID" -> AIFollow, "Mob2ID", 0,0,0,0
```

La première fonction, AIFollow, est utilisée pour suivre un acteur jusqu'à une position particulière dans la cellule courante.

La seconde, AIFollowCell est utilisée pour suivre un acteur jusqu'à une position spécifique dans une cellule donnée.

Le package IA Follow (suivre) est utilisé pour qu'un PNJ ou une créature en suive un autre ou pour qu'il suive le PNJ. Cela peut être utilisé entre créature pour former des groupes d'animaux ou des caravanes par exemple.

L'exemple suivant donne l'ordre à un guar de charge de suivre un PNJ spécifique ou que ce dernier aille :

```
elseif ( etat == 20 )
    HB_guar_pack_adros_ -> AIFollow, HB_adros_darani, 0, 0, 0, 0
    AITravel -8144, -19409, 728 ;nouvelle destination
    set etat to 30
```

Comme il n'y ni durée, ni destination spécifiée, le guar suivra le PNJ jusqu'à ce qu'un ordre contraire soit donné. Dans ce cas, et comme pour tous les packages I.A., il faut faire attention à ne pas répéter infiniment l'ordre, l'utilisation d'une condition assurant un unique appel de la fonction est recommandée, vous pouvez reprendre l'exemple donné pour AITravel.

Les informations de durée, de cellule de destination ou de coordonnées, sont utilisées pour déterminer l'endroit (ou le moment) où la fonction de suivi doit s'arrêter. A ce moment, le signal GetAIPackageDone est envoyé de la même façon que pour la fonction AITravel.

Pour l'instant, la signification du paramètre reset reste encore inconnue.

#### 11.12.5 Verifier la nature du package IA qui s'exécute

Pour les scripts attachés à des acteurs, on peut avoir intérêt, pour l'intérêt du jeu, à faire des choses complexes. Et pour les scripts en question, il peut être extrêmement utile de savoir quel est le package I.A. qui est actuellement en train de s'exécuter avant de décider quelle action nouvelle entreprendre. Pour cela on dispose de la fonction GetCurrentAIPackage :

GetCurrentAIPackage (returns short)

```
If ( GetCurrentAIPackage == 2 )
    [blablabla]
endif
```

Cette fonction retourne un entier dont les valeurs potentielles sont les suivantes :

None = -1  
 Wander = 0  
 Travel = 1  
 Escort = 2  
 Follow = 3  
 Activate = 4  
 Pursue = 5

#### 11.12.6 Forcer un mouvement discret

Par exemple, lorsque l'on crée un PNJ compagnon pour le PJ, il peut être intéressant de le forcer à se déplacer discrètement (si il veut courir il marchera), on dispose alors de la fonction ForceSneak :

```
ForceSneak      (force le PNJ à être discret, si il veut courir, il ne pourra que marcher)
ClearForceSneak (annule l'ordre de discrétion)
GetForceSneak   (retourne 1 si le PNJ est actuellement forcé à la discrétion)
```

Malheureusement, il ne semble pas exister de commande similaire pour forcer un PNJ à courir. Vous pouvez regarder le script Lookout pour avoir un exemple d'utilisation. En voici un extrait :

```

elseif ( walkstate == 2 )
    Fargoth->ForceSneak ; on force Fargoth à être discret
    Fargoth->AiTravel -11468.595,-71511.531,173.728 ;il va vers l'arbre
    set walkstate to 3

elseif ( walkstate == 3 )
    if ( Fargoth->GetAiPackageDone == 1 )
        ;Fargoth->Equip "torch_infinite_time_unique"
        set walkstate to 4
        ;MessageBox "DOIT ETRE A TROIS"
    endif

elseif ( walkstate == 4 )

    set timer to timer + GetSecondsPassed
    Fargoth->ClearForceSneak ; termine l'action silencieuse
    Fargoth->AiWander 0 0 0 0 0 0 0 ; Fargoth erre autour de l'arbre

        if ( timer > 3 )
            Fargoth->ForceSneak ; de nouveau discret
            Fargoth->AiTravel -11410.590,-72057.188,133.644 ;va vers le mur
            set walkstate to 5
        endif
endif

```

#### 11.12.7 Amener un acteur ou un objet à une nouvelle position

Position, x, y, z, zRot (fonctionne uniquement pour les extérieurs)  
 PositionCell, x, y, z, zRot, "cellID" (pour les intérieurs et les extérieurs)  
 (ATTENTION x,y,z et zRot:doivent être des valeurs, les variables ne sont pas tolérées)

Player -> position -23515, -15355, 3355, 90  
 "Actor\_ID" -> PositionCell, -254, 475, -376, 360, "Balmora, Council Club"

Les cas d'application classique de cette fonction sont les scripts de téléportation, l'anneau de téléportation classique qui transporte le joueur jusqu'à un lieu spécifique. Faites attention au fait que cette fonction n'accepte pas les variables en argument, uniquement des valeurs. Voici un script classique de téléportation:

```

Begin TeleportScript
;ce script est par exemple attaché à un anneau

short status
short button
short OnPCEquip

if ( MenuMode == 1 )
    return
endif

if ( OnPCEquip == 1 )
    Set Status to 10
    Set OnPCEquip to 0
Endif

If ( status == 10 ); affiche un menu de choix
MessageBox "Où voulez vous aller?", "Balmora", "Vivec", "Cancel"
    Set Status to 20
Elseif ( status == 20 ); attend une réponse

```

```

Set button to GetButtonPressed
If ( button == 0 )
Player -> PositionCell -21278, -17613, 534, 0, "Balmora (-3, -3)"
Elseif ( button == 1 )
Player -> Position 29872, -82108, 578, 180
Elseif ( button >= 2 )
    Set status to 0
Endif
Endif
End

```

Vous remarquerez que les deux cellules de destination sont des cellules d'extérieur et pourtant, deux commandes sont utilisées. La différence entre Position et PositionCell réside dans le fait que la première commande ne peut pas être utilisée pour les téléporter vers une cellule d'intérieur.

Si vous essayez de téléporter le joueur (ou un objet) vers une place normalement inaccessible (coincé dans un objet, un mur, ou à une coordonnée qui n'existe pas) l'objet téléporté est en fait placé à la position la plus proche qui ne vérifie pas une de ces conditions problématiques.

Ces fonctions peuvent également être utilisées pour déplacer des PNJ ou des objets.

### 11.13 Textes et Dialogue

#### 11.13.1 Afficher un message et des choix

Associée à la fonction GetButtonPressed, la fonction MessageBox permet de poser des questions au joueur.

```

MessageBox, "Message", [var1], [var2], ["button1"], ["button2"]
GetButtonPressed (donne un entier au format short, qui donne le numéro du choix du joueur, on commence à 0
et la fonction retourne -1 tant qu'aucun choix n'est fait)

```

La commande Message Box vous permet de donner des informations au joueur, ou de lui poser des questions. Il y a plusieurs façons de l'utiliser, la plus simple affiche un simple message à l'écran durant quelques secondes. L'exemple suivant fait apparaître un message lorsque le joueur s'équipe avec un objet spécifique :

```

Begin informJoueur

if ( MenuMode == 1 )
    return
endif

if (OnPCEquip ==1 )
MessageBox, "L'épée vibre dans votre main"
Endif

End informJoueur

```

La seconde utilisation de cette commande fait apparaître un message à l'écran jusqu'à ce qu'il presse un bouton:

```

MessageBox, "Ulyah lève ses mains et commence une incantation. Vous allez être téléporté vers le Sheogorad",
"ok"

```

Essayez de n'utiliser ce mode d'opération que lorsque vous avez une très bonne raison de le faire, il n'est que rarement agréable pour le joueur de devoir presser un bouton OK simplement pour signaler qu'il a bien reçu une information.

La troisième utilisation possible vous permet de poser une question au joueur et d'en attendre une réponse, un choix par l'intermédiaire de la fonction GetButtonPressed fonction:

```

Begin choix

Short button

```

**Short status**

Short OnPCEquip ; il faut declarer cette grandeur comme variable du script, sinon il y aura des erreurs

```

if (OnPCEquip ==1 )
    MessageBox, "L'épée vibre dans votre main. Voulez vous la garder ?", "oui", "non"
    Set OnPCEquip to 0 ;n'affiche le message qu'une fois
    Set status to 1
Endif
If ( status == 1); puis on attend une decision du joueur
    Set button to GetButtonPressed
    If (button == 0); si le joueur choisi oui, on continue normalement
        Return
    Elseif (button == 1)
        Player -> drop, "power_sword" ; sinon le joueur jette l'épée à terre
        Set PCEquip to 1
    Endif
Endif
End

```

Dans ce cas de script, vous remarquerez qu'il manque des instructions qui permettent d'interrompre le script prématurément pour éviter de ralentir le jeu. Si vous avez bien compris ce que je vous répète depuis le début, vous saurez qu'il y a un test des tests à faire pour interrompre l'exécution du script. Je vous laisse les placer vous-même.

11.13.2 Ajouter un Topic dans les dialogues

Dans la section Dialogue, de ce document, nous avons vu comment créer des nouveaux sujets de conversation. Si vous n'avez fait qu'ajouter le Topic dans la liste des sujets de conversation, vous avez du vous trouver bête en ne pouvant toujours pas atteindre ce sujet de conversation avec le PNJ. La réponse est simple, vous ne connaissez toujours pas le sujet de conversation, donc vous ne pouvez pas en parler avec les PNJ. Pour rendre accessible le sujet de conversation au joueur, il faut l'informer de l'existence de ce sujet de conversation. Pour cela vous avez deux solutions, la première consiste à introduire le sujet de conversation grace à une autre parole, si vous faites en sorte qu'un PNJ prononce les mots associés à votre topic par exemple dans les salutations (comme nous l'avons fait dans la quête des chaussures...).

La seconde alternative est d'ajouter le sujet de conversation par script. C'est une manière élégante de le faire quand il ne semble pas logique qu'un PNJ parle de ce sujet au joueur spontanément. Il y a des choses que le joueur découvre petit à petit, en lisant des livres ou des parchemins, il va ensuite interroger les PNJ. Ou bien un PNJ peut vous surprendre par son attitude, mais il n'abordera pas spontanément le sujet sauf si on l'interroge, un PNJ défiguré ne dira pas spontanément « Bonjour, vous avez vu je suis défiguré ».

Dans ce cas, on attache un script au PNJ :

**Begin AjouteDialogueSpecial**

```

;ajoute la possibilite de demander pourquoi le PNJ est défiguré
AddTopic, " défiguré"

```

**End AjouteDialogueSpecial**

ATTENTION : avant d'éditer ce script, vous devez avoir ajouter le sujet de conversation correspondant (Topic), dans le cas contraire, lors de la sauvegarde du script vous auriez un message d'erreur.

Il n'y a pas de possibilité de retirer un sujet de conversation par le biais des scripts. Si vous voulez malgré tout faire quelque chose de la sorte, il vous faut modifier la valeur d'une variable locale du PNJ, puis vous assurer, par le biais des conditions du PNJ, que ce dernier n'abordera le sujet de conversation que sous une condition sur la valeur de cette variable locale.

11.13.3 Ajouter des entrées dans le journal et vérifier son contenu.

Journal, ID, Index

Journal, MG\_BCSroomsCombat, 10

Cette instruction ajoute une entrée au journal pendant le jeu. Comme pour les sujets de conversation, il faut s'assurer que l'entrée a d'abord été créée dans la fenêtre des dialogues. Le paramètre Index, indique la section du journal qui doit être ajoutée dans le journal. Cela n'est pas une règle, mais il est conseillé d'éviter les noms simples pour les sujets du journal. En les préfixant de deux lettres (comme Bethesda a pu le faire), ces deux lettres vont permettre d'éviter que le sujet du journal ne soit confondu avec un sujet de conversation, et risquerait d'apparaître dans les conversations.

GetJournalIndex, JournalID (returns short)

If (GetJournalIndex, MG\_BCSroomsCombat == 10)

Cette fonction retourne un entier qui correspond à l'index (le plus haut ou le plus récent???) de l'entrée correspondante dans le journal. C'est une manière très simple de connaître l'état d'avancement d'une quête et d'avoir une réaction de la part des PNJ en accord avec cet avancement.

Voici un exemple, bref, de script utilisant ces deux fonctions dans le jeu:

Begin attackEsclave

short nolore

If ( GetDeadCount "Vorar Helas" > 0 )

return

endif

if ( GetJournalIndex "MV\_SlaveMule" < 102 )

If ( GetDistance, "Rabinna" < 512 )

Rabinna->AiWander 0 0 0 0 0 0

StartCombat, "Rabinna"

Journal "MV\_SlaveMule", 102

endif

endif

End

## 11.14 Combat

### 11.14.1 Débuter et finir un combat

StartCombat, ActorID

StopCombat

Actor\_ID1 -> StartCombat, ActorID2

Actor\_ID -> StopCombat

Les fonctions StartCombat et StopCombat sont utilisées pour qu'un PNJ (ou une creature) passe en mode de combat ou le quitte. Alors que l'utilisation de la fonction StopCombat à chaque image, semble ne pas poser de problème, il est en revanche nécessaire, pour la fonction StartCombat, d'utiliser un système d'invocation unique du même type que pour les fonction I.A., dans le cas contraire, il semblerait que le PNJ ne fasse rien du tout. D'un autre côté, l'utilisation de StopCombat sans sécurité du même type, risque de créer un PNJ sans défense face une attaque du PNJ, cela peut en faire un Gandhi en puissance qui attendra simplement que le PJ le tue sans se défendre ni répliquer.

Dès le que le PNJ est en mode de combat, les réglages I.A. de l'acteur sont appliqués par défaut, c'est-à-dire que l'on récupère les paramètres flee fight réglé dans la fenêtre A.I. du PJ.

*11.14.2 Détecter une attaque de la part du joueur*

OnPCHitMe (retourne un entier du type short)

Short OnPCHitMe  
If (OnPCHitme == 1)

Il s'agit d'une variable du jeu qui vaut 1 lorsque le joueur touche en combat le PNJ ou la créature. Il faut savoir que cette variable doit être explicitement remise à 0. Voici un exemple extrait du mod des marchands voyageur de GhanBuriGhan, il attaché ce script au guar de charge:

Begin\_HB\_Adros\_GuarDefend

```
float timer
short attackstate
short OnPCHitMe

if ( OnPCHitMe == 1 )
    set attackstate to 1
    set OnPCHitMe to 0
endif

if (attackstate == 1)
    StartCombat, player
    set timer to ( timer + GetSecondsPassed)
    if ( timer >= 1 )
        set timer to 0
        if ( GetLOS, HB_adros_darani == 1)
            HB_adros_darani -> StartCombat, Player
            set attackstate to 0
        endif
    endif
endif

End
```

*11.14.3 Connaître la prime sur la tête d'un joueur*

Cette fonction n'est pas documentée.

GetPCCrimeLevel (returns short)

Donne le niveau de crime d'un joueur (une sorte d'indicateur pour savoir si un crime commis par le joueur est détecté par les PNJ dans l'entourage). On se référera au script "Bill\_MT\_writxxx" pour un exemple d'utilisation de cette fonction.

**11.15 La magie***11.15.1 Invocation de sorts*

Dans le domaine de l'invocation des sorts, on dispose de 3 fonctions.

AddSpell, SpellID  
RemoveSpell, SpellID  
Cast, SpellID, TargetID

La fonction AddSpell permet d'ajouter un sort à l'objet sur lequel la fonction est invoquée. Cela peut recouvrir deux sens, soit le sort est ajouté à la liste des sorts de l'acteur, soit ajouter une malédiction, une maladie quoique ce soit d'autre...La signification de la fonction RemoveSpell a un comportement similaire, les sorts classiques

sont simplement retirées de la liste du joueur alors que les malédictions ou les effets des maladies sont tout simplement interrompus.

La fonction Cast invoque le sort donnée par "SpellID" et le dirige sur la cible "TargetID", la conséquence est que la cible souffrira directement des effets bénéfiques ou non du sort.

La fonction Cast peut être utilisée pour réaliser des pièges, comme dans l'exemple suivant pour un coffre. Vous remarquerez qu'il y a un test du type 'faire une fois', afin que les effets du sort ne soit pas permanents sur le joueur :

```
Begin Trap_script

short done

if ( OnActivate == 1 )
    if ( done == 1 ) ;bloque l'invocation multiple du sort
        Activate
        return
    else
        Cast, "flame", Player ;inflige des dégats au joueur
        set done to 1
        Activate
    endif
endif

End trap_script
```

Puis, voici un exemple de script qui utilise la fonction Addspell:

```
begin Item_Cast

short OnPCEquip

short CurseAdded
float Timer

if ( CurseAdded )
    set Timer to ( Timer + GetSecondsPassed )
    if ( Timer >= 25 ) ; après 25 secondes on cesse d'infliger des dégats au joueur.
        set Timer to 0
        Player->RemoveSpell "ItemFlame"
    set CurseAdded to 0
    endif
endif

if ( OnPCEquip );lorsque le PJ s'équipe avec l'objet
    if ( MenuMode )
        return
    elseif ( CurseAdded == 0 )
        Player->AddSpell "ItemFlame" ; un sort de type malédiction!
;c'est un sort que l'on a créé pour le besoin qui inflige 1 point dégât par secondes
        set Timer to 0
        set CurseAdded to 1
    elseif ( CurseAdded ) ; on invoque une seule fois le sort
        Player->RemoveSpell "ItemFlame"
        set CurseAdded to 0
    endif
endif

end Item_Cast

(script by Patrin, edited)
```

Le résultat de ce script est l'invocation d'un sort du type malediction qui inflige un point de dégât de feu par seconde. Vous remarquerez la présence d'une structure du type 'on ne le fait qu'une fois' dans ce script. **Dans ce cas, l'absence de structure de ce type peut provoquer le plantage du jeu !**

#### 11.15.2 Gérer et tester les sorts

```
GetSpell, SpellID  
If (Player -> GetSpell, "heal companion" == 1)
```

La fonction GetSpell permet de savoir si le joueur connaît ou non le sort donné. Elle retourne 1 si la créature ou le PNJ (ou le PJ) dispose du sort dans sa liste, cela fonctionne aussi pour connaître les malédictions ou les maladies.

```
GetSpellEffects, SpellID
```

La fonction GetSpellEffects permet de savoir si la personne est affectée des effets d'un sort ou d'une maladie. Elle retourne 1 si c'est le cas. L'instruction suivante pourrait être ajoutée au script du piège présenté plus haut :

```
if ( Player -> GetSpellEffects, "flame" == 1 )  
    MessageBox "Vous brûlez"  
endif
```

C'est la meilleure possibilité pour ajouter de "nouveaux effets" magiques. Un exemple idiot est la création d'un sort qui a un effet minime – augmenter de 1 point par seconde la caractéristique Chance. La fonction GetSpellEffects est alors utilisée pour détecter si le sort a été invoqué sur le joueur, le reste du script est alors soumis à la validation de ce test.

#### 11.15.3 Gérer et tester les effets des sorts

```
GetEffect, Effect  
If ( GetEffect, sEffectShield == 1)
```

Cette fonction retourne 1, si lors de son invocation l'acteur est soumis aux effets du sort cité. IL faut bien savoir que dans MW, les effets ne sont pas les sorts mais les éléments qui constituent la réalité du sort. Vous trouverez en appendice la liste des effets des sorts.

```
RemoveEffects, Effect  
RemoveEffect sEffectShield
```

Cette action annule tous les sorts qui utilisent cet effet.

**Important: Il faut bien comprendre que les effets ne sont pas les sorts mais les éléments qui constituent les sorts.**

### **11.16 Les bruitages**

#### 11.16.1 Jouer un son

```
PlaySound, "soundID"  
  
PlaySoundVP, "soundID", volume, pitch  
  
PlaySound3D, "soundID"  
  
PlaySound3DVP, "soundID", volume, pitch  
  
"ex_gg_portcullis_02"->Playsound3DVP "Dwemer Door Open" 1.0 1.0
```

Ces fonctions ne marchent qu'avec des fichiers wav.

La fonction PlaySond permet de jouer un son sans aucune modification. Cette fonction ne tient pas compte de l'objet auquel le script est attaché, le son est toujours joué au volume maximal. Il semble intervenir directement dans l'oreille du PJ.

La fonction PlaySound3D permet d'ajouter un effet tridimensionnel au son, il semble alors émis par l'objet auquel le script est attaché.

Les variantes en "VP" de ces commandes permettent de modifier le volume et la tonalité du son. Bethesda n'a pas beaucoup utilisé ces fonctions, et, apparemment, les utilisations donnent toujours les valeurs 1.0 aux deux variables supplémentaires que sont le volume et la tonalité. De l'expérience de GhanBuriGahn, ces fonctions ne semblent pas prendre de variable en entrée, uniquement des valeurs.

Le son utilisé est celui dont l'ID a été ajoutée (ou non) dans la fenêtre des sons accessibles par le menu Gameplay. Pour ajouter un son à la liste de ce qui est jouable dans le jeu, il faut ajouter le fichier wav le répertoire Data Files/Sound.

Une bonne source de son pour ajouter à l'ambiance sonore de MW est <http://www.findsounds.com/>.

#### 11.17 Garder une trace du passage du temps

Il existe de nombreuses possibilités pour suivre le passage du temps, certaines d'entre ne sont pas ou peu documentée dans le fichier d'aide du TESCS.

##### 11.17.1 Chronomètre

Nous avons déjà vu plus haut comment réaliser un chronomètre en utilisant la fonction GetSecondsPassed.

#### GetSecondsPassed

Cette fonction retourne la durée en secondes qui s'est écoulée depuis la dernière execution du script, en d'autre termes depuis la dernière image. Il s'agit évidemment d'une valeur non entière puisqu'il y a plusieurs images par seconde. L'utilisation classique en est la suivante :

```

Begin Chronometre

Float chrono
Short etat

Set chrono to (chono + GetSecondsPassed)

If (chrono > 10)
    MessageBox "Affiché toutes les 10 secondes"
    Set chrono to 0
Endif

End Chronometre
    
```

##### 11.17.2 Connaitre l'heure du jour

Il est également possible de connaître l'instant courant de la journée par l'intermédiaire de la fonction GameHour. Cette fonction peut être utilisée pour créer des PNJ plus normaux, qui ne sont pas dans les rues toute la nuit par exemple.

```
;ce script pourrait par exemple etre attache à un PNJ
Begin NuitNormal

If (GameHour >= 20 )
    disable
elseif (GameHour < 7)
    enable
endif

End NuitNormal
```

### 11.17.3 Garder une trace du passage des jours

Il y a une autre variable du jeu, qui permet de connaître le 'jour du mois'. Elle est accessible par Day. Et elle retourne une chaîne de caractère qui peut par exemple contenir "17, last seed" pour indiquer qu'on est le 17 jour du mois. Cette fonctionnalité peut être utilisée pour connaître le nombre de jour écoulés depuis un événement précis, il suffit par exemple de lancer le script suivant par la commande StartScript, qui en fera un script global :

```
Begin compteLes Jours

Short nombreDeJourEcoule
Short aujourd'hui

if ( aujourd'hui != Day ) ;si le jour a change (a augmente)...
    set aujourd'hui to Day
    set nombreDeJourEcoule to nombreDeJourEcoule + 1 ;on increment le compteur
endif
```

Cette fonctionnalité peut être utilisée pour créer des quêtes qui doivent être réalisées en un certain nombre de jour afin de s'assurer que le décompte du temps est bien réalisé. On pourrait également créer des scripts originaux utilisant cette fonction pour par exemple créer des armes maudites dont les effets iroient croissant avec le temps.

### 11.17.4 Les phases de la lune

Une fonction absolument indispensable pour tout mod incluant des loups-garous.

```
GetMassePhase
GetSecundaPhase
```

La fonction GetMassePhase permet de récupérer la phase de la lune principale, alors que GetSecundaPhase permet de récupérer la phase de la petite lune.

```
If (GetMassePhase == 4)
    [active la transformation en garous]
endif
```

**Note:** Le fichier d'aide fait apparaître la fonction GetSecundusPhase, la syntaxe donnée ici est la seule correcte. On notera également que les deux fonctions retournent 0 dès que le PJ est en intérieur. A la fin du document se trouve la liste des significations des valeurs de retour de ces fonctions.

## **11.18 Utiliser les objets**

### 11.18.1 Utiliser l'inventaire

#### 11.18.1.1 Ajouter et retirer des objets de l'inventaire

```
AddItem, ObjectID, count
```

```
RemoveItem, objectID, count
```

```
Actor -> AddItem, "item_ID", 1
```

Je pense que la signification de ces fonctions est évidente, AddItem, ajoute l'objet ObjectID à l'inventaire avec la quantité count. De même la fonction RemoveItem enlève les objets de l'inventaire. Ce type de fonction est à utiliser pour faire des échanges entre les PNJ et le PJ par exemple.

L'invocation de la méthode RemoveItem fait simplement disparaître l'objet de l'univers. Si vous désirez poser l'objet à même le sol, il faut utiliser la fonction DropItem qui posera l'objet au pied de l'acteur.

Retirer un objet qui n'est pas présent dans l'inventaire semble ne pas poser de problème, mais, retirer un objet sur lequel un script est attaché risque de provoquer un plantage du jeu.

Le script suivant a été proposé sur un forum, il permet de demander au joueur si il désire réutiliser un objet usage ce qui permet de le replacer dans l'inventaire une version de recyclage de l'objet.

```
Begin scr_thing
```

```
short button
```

```
short OnPcEquip
```

```
short etat
```

```
if ( MenuMode == 1 ) ; si le joueur est dans le mode menu... On termine vite
```

```
    return
```

```
endif
```

```
if ( OnPCEquip == 1 ) ; lorsque l'objet est equipe
```

```
    set etat to 1
```

```
    set OnPCEquip to 0 ; on remet la variable à 0 sinon on a des problemes
```

```
endif
```

```
if (etat ==1)
```

```
    MessageBox "Reutiliser?" , "oui" , "non"
```

```
    set etat to 2
```

```
elseif (etat ==2)
```

```
    set button to GetButtonPressed
```

```
    if ( button == 0 )
```

```
        PlaySound "mysticism cast"
```

```
        player->removeitem "item_a", 1 ; cette ligne fait planter le jeu!!!
```

```
        player->additem "item_b", 1
```

```
        set etat to 0
```

```
    elseif ( button == 1 )
```

```
        set etat to 0 ; quand c'est fait on reinitialise tout
```

```
    endif
```

```
endif
```

```
end
```

Ce script fonctionne bien sans la ligne RemoveItem, mais à la ligne signalé, le jeu plante. La raison en ai que le script est attaché à l'objet, et qu'il est « détruit » en même temps que l'objet.... Avant d'avoir fini de s'exécuter.

Moralité, on ne doit pas utiliser la commande RemoveItem dans un script, si l'objet a retire est justement l'objet auquel le script est attaché. On peut toujours le faire par le biais d'un autre script local, ou bien en lançant un script global qui fera la tâche à la fin de l'exécution du script.

#### 11.18.1.2 Jeter un objet par terre

```
Drop, ObjectID, count
```

```
"Actor_ID" -> Drop, "ITEM_ID", 1
```

Cette fonction est censée pousser l'acteur à poser un objet à ses pieds, mais elle semble ne fonctionner correctement que pour le PJ. Lorsqu'on utilise cette fonction sur un PNJ, l'objet est bien retiré de l'inventaire, mais il est posé aux pieds du PJ...

#### 11.18.1.3 Vérifier qu'un objet est dans l'inventaire

```
GetItemCount, ObjectID      (retourne un entier du type short, qui donne la quantité d'objet dans
l'inventaire)

Short objectcount
Set objectcount to "Mob_ID" -> GetItemCount, "Object_ID"

If ( GetItemCount, "Object_ID" >= 1 )
```

Cette fonction interroge l'inventaire de l'acteur et retourne la quantité d'objet du type donné par ObjectID qui sont présents dans l'inventaire de l'acteur.

#### 11.18.1.4 Savoir si un objet est équipé

**OnPCEquip**

Cette variable prend la valeur 1 à l'instant où le joueur équipe l'objet. Elle garde cette valeur tant que l'objet est équipé, vous pouvez la remettre à 0 manuellement pour créer un évènement d'équipement.

```
if ( OnPCEquip == 1 )    ; lorsque l'objet est équipé
    [faire quelque chose]
    set OnPCEquip to 0    ; on veut ne le faire qu'une fois...
endif
```

Le script précédent remet la variable à 0 de sorte que les instructions [faire quelque chose], ne seront exécutées qu'au moment précis où le joueur s'équipe de l'objet. Il est également possible d'utiliser une variable d'état qui permet de contrôler l'instant où les instructions seront effectivement effectuées.

Notez bien que l'on peut également utiliser cette variable lorsque le joueur est dans un menu :

```
If (MenuMode ==1)
    if ( OnPCEquip == 1 )    ;lorsque l'objet est équipé
        [faire quelque chose]
        set OnPCEquip to 0    ; ne le faire qu'une fois
    endif
endif
```

Ce script s'exécutera pendant que le PJ est dans le menu, et dès que l'objet est équipé. Alors que l'exemple suivant ne fera les actions que lorsque le joueur sortira du menu :

```
If (MenuMode ==1)
    Return
Endif

if ( OnPCEquip == 1 )    ;lorsque l'objet est équipé
    [faire quelque chose]
    set OnPCEquip to 0    ; ne le faire qu'une fois
endif
```

#### 11.18.1.5 S'équiper d'un objet

**EquipErreur ! Signet non défini., ObjectID**

```
Actor_ID -> Equip, "p_restore_health_q"
```

**Attention cette fonction ne marche pas dans MW, mais elle semble être fonctionnelle dans Tribunal...**

L'utilisation de cette fonction peut servir pour faire des armures maudites que le joueur ne peut enlever etc... mais la fonction semble ne pas fonctionner dans la majorité des cas pour MW simple.

### 11.18.2 Vérifier qu'un objet est équipé et l'utiliser

#### 11.18.2.1 Vérifier l'utilisation

On dit qu'un objet est « utilisé » au moment où le joueur presse la barre [ESPACE] (réglage par défaut). La plupart des objets ont une action standard qui leur est associée et qui est exécutée au moment de l'utilisation de l'objet. Les portes s'ouvrent, les coffres montrent leur contenu, les acteurs entament un dialogue etc....

Il est possible d'intercepter ces actions par la fonction OnActivate pour faire autre chose ou vérifier une condition avant d'exécuter l'action:

```
OnActivate
```

```
If ( OnActivate == 1 )
```

OnActivate retourne la valeur 1 durant une image après l'activation de l'objet. Pour réaliser l'action associée à un objet, on utilise la commande activate. Cela permet de faire qu'un PNJ par exemple ouvre une porte...:

```
Activate
```

```
"Object_ID" -> activate
```

**Notes:** GhanBuriGhan dit avoir testé cette fonction sur les portes sans succès. En revanche, il ne parle pas de problème pour l'activation des conteneurs, des acteurs ou des activateurs autres que les portes.

Le script suivant montre un usage sympathique de ces fonctions. Il est attaché à un objet de type conteneur, un coffre piégé qui n'en informe pas le joueur. (l'option trapped n'est pas cochée):

```
Begin Trap_script
short done
if ( OnActivate == 1 )
.....if ( done == 1 ) ;on ne le fait qu'une fois
.....Activate
.....return
.....else
.....Cast, "flame", Player ;inflige des dégât au joueur
.....set done to 1
.....Activate ; puis on appelle la fonction classique d'ouverture du coffre
.....endif
endif
End trap_script
```

### 11.18.3 Verrouiller et déverrouiller des portes et des coffres

```
Lock, locklevel
Unlock
GetLocked (retourne un short, 1 ou 0)
(valable pour les portes et les coffres)

My_Door -> Lock, 50

If (Getlocked == 1)
    Unlock
```

Endif

Ces fonctions sont utilisées pour verrouiller ou déverrouiller des coffres et des portes. Le fonction `GetLocked` retourne (1) lorsque l'objet sur lequel on invoque la fonction est verrouillé.

`Lock` verrouille l'objet avec le niveau de verrouillage donné, et `Unlock` retire tout verrouillage sur l'objet sans s'occuper du niveau de verrouillage.

Voici un exemple de script produit par qwert, qui permet d'augmenter le niveau de Sécurité en le reverrouillant en permanence:

Begin PC\_Security\_Skill\_Trainer

float timer

if ( menumode == 1)  
return  
endif

set timer to timer + GetSecondsPassed  
if ( timer > 10 )  
set timer to 0  
endif

if ( timer == 0 ) ;utilisation d'un chrono pour reverrouiller le coffre toutes les 10 secondes  
"Storm\_Chest\_Trainer"->Lock 50  
endif  
End

#### 11.18.4 Activer et désactiver un objet

Il est possible de faire en sorte qu'un objet ou une personne ne soit pas visible et donc totalement ignore dans le jeu. Pour cela on utilise les fonctions suivantes qui permettent d'activer ou de désactiver les objets.

Enable  
Disable  
GetDisabled (retourne 1 si l'objet est désactivé)

ObjectID -> enable  
(valable pour tous les types d'objet)

La fonction `disable` fait complètement disparaître l'objet du monde du jeu. Il n'est pas visible et n'est pas non plus calculé (bien que les scripts qui lui sont attachés soit actifs), la fonction `enable` fait réapparaître un objet dans le monde, il est de nouveau visible et le joueur peut interagir avec.

`GetDisable` permet de savoir si un objet est ou non visible dans le monde du jeu. Ces fonctions sont très puissantes et on peut les utiliser pour de multiples objectifs, faire apparaître des portes secrètes, faire disparaître des maisons etc..... Ces fonctions sont par exemple utilisées pour la construction de votre forteresse par votre maison noble.

ATTENTION, il ne faut pas désactiver un objet depuis un script qui lui est attaché pour la même raison que pour la fonction `RemoveItem`.

Un exemple extrait du jeu est le `SlaveScript` qui permet de « libérer » les esclaves dès que le joueur quitte la cellule:

begin slaveScript

short slaveStatus  
short doOnce  
short NoLore

[pour les autres statuts des esclaves reportez vous au script du jeu]

```

if ( slaveStatus == 3 )
    if ( GetCurrentAIPackage == 3 )
        AIWander 512 0 0 0 0 0 0 0 0 0 0
    endif
    if ( GetItemCount Slave_Bracer_Left > 0 )
        Drop Slave_Bracer_Left 1
    endif
    if ( GetItemCount Slave_Bracer_Right > 0 )
        Drop Slave_Bracer_Right 1
    endif
    if ( CellChanged == 1 )
        Disable ;*****fait disparaitre les esclaves lorsque le joueur quitte la cellule
    endif
endif
end slaveScript

```

#### 11.18.4.1 Attention : désactiver les lumières

Apparemment il y a un problème avec le moteur du jeu lorsque l'on désactive des sources lumineuses, les acteur et les autres objets restent illuminés... alors que le reste du monde ne l'est plus. GhanBuriGhan dit n'avoir pas tester de solution de contournement du problème, mais une solution possible pourrait être de déplacer la source lumineuse ailleurs (dans un endroit où elle n'éclaire rien) A TESTER DONC.

#### 11.18.5 Déplacer et faire tourner des objets

Les fonctions suivantes ne fonctionnent pas sur le joueur, les PNJ ou les créatures. Elles fonctionnent sur tous les autres types d'objets.

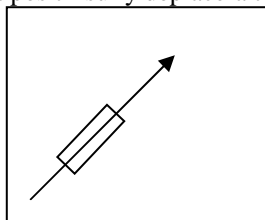
##### 11.18.5.1 Déplacer un objet le long de son axe

```

Move    axe, unité/sec
Move    x, 100

```

Déplace un objet le long de l'axe sélectionné (x, y, ou z) à la vitesse donnée. Cette vitesse est donnée en unités par seconde sachant on l'on compte 21.3 unités par pied (environ 33 cm). Ce mouvement est réalisé dans le repère local de l'objet. Ainsi un mouvement positif sur y déplacera toujours l'objet vers ce qui est son avant.



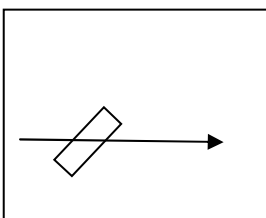
##### 11.18.5.2 Se déplacer le long d'un axe du monde

```

MoveWorld Erreur ! Signet non défini. axis, units/sec
MoveWorld z, 100

```

Déplace un objet le long du l'axe sélectionné dans le repère du monde (x, y, ou z) à la vitesse donnée. Ainsi un mouvement positif en z fera toujours monter l'objet quelque soit son orientation locale.



ordonnées du monde, l'axe Z est toujours orienté du bas vers haut, X est orienté d'Est en Sud.

Voici un exemple pris sur un forum, ce script permet de faire déplacer une plateforme lorsque le joueur se trouve dessus:

```

Begin platform_script

Short PlatformMoving
Short ActivateMe
Float Timer

If ( GetStandingPC == 1 )
    Set ActivateMe to 1
Endif

If ( ActivateMe == 1 )
    If ( PlatformMoving == 0 )
        Set Timer to Timer + GetSecondsPassed
        If ( Timer <= 15 )
            "floating_platform_01"->MoveWorld X 10
        Else
            Set Timer to 0
            Set PlatformMoving to -1
        Endif
    Endif
    If ( PlatformMoving == -1 )
        Set Timer to Timer + GetSecondsPassed
        If ( Timer <= 15 )
            "floating_platform_01"->MoveWorld X -10
        Else
            Set Timer to 0
            Set PlatformMoving to 0
            Set ActivateMe to -1
        Endif
    Endif
Else
    "floating_platform_01"->SetAtStart
Endif

End platform_script
    
```

#### 11.18.5.3 CellUpdate

CellUpdate

**ATTENTION CETTE FONCTION SEMBLE NE PAS FONCTIONNER!** Si on s'en tient à ce que dit Bethesda, cette fonction permet de mettre à jour le contenu d'une cellule. Cette fonction devrait être utilisée lorsque l'on déplace des objets sur de grandes distances. Le moteur du jeu garde une trace de la position d'origine des objets et si un objet est déplacé trop loin, il peut ne pas être calculé convenablement.

La partie concernant le mauvais calcul des objets est certainement vraie, ils semblent disparaître si ils sont déplacés trop loin. Mais les tentatives d'utilisation de cette fonction se sont apparemment toujours soldées par un échec et le message d'erreur suivant: "need to add function code for function CellUpdate"

#### 11.18.5.4 Connaître la position d'un objet et le regarder

GetPos, axis

Object\_ID -> GetPos, z

Lorsque l'on déplace un objet par l'invocation des méthodes précédentes, il peut être utile de récupérer les informations relatives à leur position actuelle. Dans le script suivant, on utilise cette fonction pour contrôler le mouvement d'une source lumineuse (un feu) pour faire un sorte de feu de camp qui apparaît lentement et disparaît de même à une heure donnée de la journée., la position initiale du feu est à Z= 511:

Begin \_HB\_Scheduled fire

short control\_fire

; le script est attaché à un PNJ qui garde le feu de camp.

;\*\*\*\*\* ce controle n'intervient que si le PNJ s'éloigne trop du camp:

if ( GetDistance, "HB\_Furn\_De\_Firepit\_camp" < 600 )

    If ( GameHour < 17 )

        if ( HB\_Light\_Fire\_camp -> GetPos Z >= 400 )

            HB\_Light\_Fire\_camp -> MoveWorld z, -0.1 ; éteint le feu

        else

            HB\_Light\_Fire\_camp -> disable

        endif

    elseif ( GameHour >= 17 )

        HB\_Light\_Fire\_camp -> enable

        if ( HB\_Light\_Fire\_camp -> GetPos Z < 511 )

            HB\_Light\_Fire\_camp -> MoveWorld z, 0.1 ; rallume le feu

        else

            HB\_Light\_Fire\_camp -> enable

        endif

    endif

endif

end

#### 11.18.5.5 Faire tourner un objet

Ces fonctions sont similaires aux fonctions de déplacement que nous venons de voir, il est également possible d'orienter un objet par rapport à son repère local ou par rapport au repère du monde, et déterminer l'angle courant.

Rotate , axis, angle/sec

RotateWorld, axis, angle/sec

GetAngle , axis (retourne un réel)

La fonction GetAngle donne l'angle par rapport au repère du monde, pas par rapport au repère local.

#### 11.18.5.6 Ramener un objet à sa position initiale

SetAtStart

Object\_ID -> SetAtStart

Ceci permet de ramener un objet à la position qui lui a été assignée dans l'éditeur avant qu'on ne lui applique le moindre mouvement de translation ou de rotation. Référer vous au script de la plate forme mobile.

#### 11.18.6 Déterminer la localisation

##### 11.18.6.1 Distance:

GetDistance, ObjectID

"ObjectID1" -> GetDistance, "ObjectID2"

Cette fonction donne la distance en unite d'un objet à un autre. La syntaxe simple (sans référence) permet de connaître la distance entre l'objet auquel le script est attaché et l'objet désigné, utilisée avec une référence comme dans l'exemple, elle permet de connaître la distance entre l'objet ObjectID1 et l'objet ObjectID2. This function returns the distance (in units) of one object to another.

On peut utiliser cette fonction pour faire des scripts qui permettent de faire des déclencheurs d'attaque.

Voici un exemple d'utilisation pris dans les scripts du jeu:

```
; Ce script est attaché à un PNJ nommé Ashamanu:
; Ashamanu ajoute l'entrée 60 du sujet "MS_WhiteGuar" lorsque le PNJ se trouve à une distance inférieure à 256
unités

if ( GetDisabled != 1 )
    if ( GetDistance Player <= 256 )
        if ( GetDistance "guar_white_unique" <= 256 )
            if ( GetJournalIndex "MS_WhiteGuar" <= 50 )
                Journal "MS_WhiteGuar" 60
            endif
        endif
    endif
endif
```

#### Limitations:

- Notez bien que vous ne devez utiliser cette fonction avec une ID unique ou bien dans un environnement dans lequel vous êtes certain qu'il n'y a qu'une instance de l'objet cité – sinon le moteur du jeu prendra comme référence la première qu'il trouvera et donnera la réponse par rapport à celle là, la plupart du temps ce n'est d'ailleurs pas ce que vous vouliez. Ainsi, un script qui permettra d'informer le joueur de la présence d'un guar à une distance inférieure à 800 unités devra être attaché à l'ID du guar et vérifier la distance vers le joueur qui lui est unique.
- Si vous détermine la distance à un objet qui se déplace dans le monde en utilisant les fonctions Move et MoveWorld, la commande GetDistance retournera toujours la distance à la position initiale de l'objet. Dans un cas comme celui-ci, il faut utiliser la fonction GetPos qui retournera la bonne information.

#### 11.18.6.2 Ligne de vision:

Cette fonction permet de savoir si il existe une ligne de vision entre deux acteurs. C'est utilisable pour savoir si un acteur en a vu un autre, ou pour savoir si il peut tirer dessus.

GetLOS, ObjectID  
Actor\_ID -> GetLOS, Player

#### Pas documentée:

GetLineOfSight  
(fonctionne peut être mieux! Mais pas encore testée)

#### 11.18.6.3 Détecteur de presence d'un acteur sur un objet

```
GetStandingPC           retourne 1 si un le PJ est dessus
GetStandingActor        retourne 1 si un acteur quelconque est sur l'objet.

If ( GetStandingPC == 1)
    [... active un piège horrible]
endif
```

C'est une fonction immense, en particulier pour les intérieurs. Déclenchement de piège, détection de présence... tout ce qui est envisageable à partir d'une fonction de detection de presence. Vous pouvez faire un 'Activeur' en utilisant le fichier nif de n'importe quel objet statique (inclus les tapis, les dalles..) et déclencher des événements en détectant la presence du joueur.

Cet exemple simple est utilisé pour lancer une boule de feu sur le joueur dès qu'il marche à un endroit donné:

```
Begin _HB_Hall_Lighting

if ( GetStandingPC == 1 )
    set HB_hallfire to 1
endif

end
```

HB\_hallfire est une variable globale utilisée pour activer le jeu de la boule de feu.

```
Begin _HB_Hall_fire_on

if ( HB_hallfire == 1)

    if ( GetPos, z, < -736 )
        MoveWorld, z, 3
        if ( GetPos, z, > -780)
            enable
        endif
    endif

else
    disable
endif

end
```

### 11.19 Statistiques et Attributs

```
GetStat
SetStat, float_variable
ModStat, float_variable (les valeurs positives sont ajoutées et les valeurs negatives sont retranchées)
```

Ici, on remplace *Stat* par n'importe quel statistiques ou attribut du jeu. Paramètre I.A., résistance, réputation etc.. (Reporter vous à la fin de ce document pour trouver les mots à utiliser)

GetStat donne la valeur actuelle de grandeur concernée.

SetStat donne la nouvelle valeur à la grandeur.

ModStat ajoute ou retire la valeur donnée à la grandeur concernée.

Ces commandes ont un champ d'application assez important. Elles peuvent être utilisées pour des objets magiques, maudits ou bénis, pour avoir des informations sur l'état du joueur. On peut rendre un PNJ plus agressifs ou plus peureux en modifiant les paramètre I.A., les rendre très inamicaux à la nuit tombée etc..

```
Set floatvar to Player -> GetHealth
Player -> SetWillpower, 20
Player -> ModHealth, floatvar
```

Cas d'utilisation spéciale pour Mod:

```
ModCurrentHealth, float
ModCurrentMagicka, float
ModCurrentFatigue, float
```

ModHealth modifie le niveau de santé maximale d'un acteur, ModCurrentHealth touche uniquement la santé actuelle de l'acteur et ne peuvent pas attribuer une valeur supérieure au maximum autorisé (le niveau de PV du joueur)

Pas documenté:

```
GetHealthGetRatio (returns float)
```

Cette fonction retourne le pourcentage de santé restante à l'acteur. Elle retourne un float dont la valeur est comprise entre 0 et 1. 1 correspond à 100% et 0 à 0%... probablement que le perso est mort ʘ . Cette fonction doit remplacer la fonction GetHealthRatio qui apparaît à tort dans le fichier d'aide.

### 11.20 Les contrôles du joueur

GhanBuriGhan avoue ne pas les avoir utilisés pour l'instant. ET comme je ne l'ai pas fait non plus, je vous avoue que ce qui suit est quasiment une copie du fichier d'aide. A mon avis, il y a beaucoup de choses à faire avec ces fonctions, mais c'est à vous de voir.

DisablePlayerControls	(le joueur ne peut plus que regarder autour de lui, ou accéder au menu des option)
DisablePlayerFighting	(Le joueur ne peut plus se battre)
DisablePlayerJumping	(le joueur ne peut plus sauter)
DisablePlayerLooking	(le joueur ne peut plus regarder autour de lui)
DisablePlayerMagic	(le joueur ne peut plus utiliser la magie)
DisablePlayerViewSwitch	(le joueur ne peut plus changer de mode de vue)
DisableVanityMode	(le joueur ne peut plus se regarder)
EnableLevelUpMenu	(fait apparaître le menu de passage de niveau)
EnablePlayerControls	(reactive les contrôles du joueur)
EnablePlayerJumping	
EnablePlayerFighting	
EnablePlayerLooking	
EnablePlayerMagic	
EnablePlayerViewSwitch	
EnableRest	(active le menu de repos)
EnableVanityMode	
GetPlayerControlsDisabled	(retourne 1 si le joueur peut se contrôler)
GetPlayerFightingDisabled	(retourne 1 si le joueur peut se battre)
GetPlayerJumpingDisabled	
GetPlayerMagicDisabled	
GetPlayerLookingDisabled	
GetPlayerViewSwitch	( <b>ne fonctionne pas</b> , apparemment)
GetVanityModeDisabled	

PCGet3rdPerson	(retourne si le joueur se regarde)
PCForce3rdPerson	(oblige la vue à la troisième personne, doit attendre que l'animation courante s'achève)
PCForce1stPerson	(pareil mais pour la vue à la première personne)

### Pas documentées:

Ces fonctions sont utilisées pour la création des personnages:

EnableBirthMenu
EnableClassMenu
EnableRaceMenu
EnableNameMenu
RemoveEffects
EnableMagicMenu
EnableMapMenu
EnableInventoryMenu
EnableStatsMenu

Pour l'instant, on ne sait pas comment ces fonctions se comportent dans le cours du jeu, mais elles peuvent être utilisées et l'ont été...

## 11.21 Le climat

### 11.21.1 Changer le temps qu'il fait

ChangeWeather, RegionID, TypeEnum

ChangeWeather, "West Gash", 4

Cette fonction permet de modifier le climat dans une region donnée en attribuant différents poids aux types de temps spécifiée dans TypeEnum. Les changements prendront effet au prochain changement météorologique. (on suppose que la fréquence des changements est donnée dans le fichier morrowind.ini, dans la section Weather, essayez de modifier la valeur de Hours Between Weather Changes=20)

Les valeurs que l'on peut mettre dans la liste TypeEnum correspondent à:

- 0 Clair
- 1 légers nuages
- 2 Brumeux
- 3 Nuageux
- 4 Pluie
- 5 Orage
- 6 tempête de cendre
- 7 Fléau

### 11.21.2 Modifier les paramètres météo d'une région

ModRegion, RegionID, clear\_var, cloudy\_var, foggy\_var, overcast\_var, rain\_var, thunder\_var, ash\_var, blight\_var (all literal values)

ModRegion, "West Gash", 10, 20, 10, 5, 5, 40, 10, 0

Modifie les probabilités des différentes météo pour une région. Elle permet de rendre les climats plus réalistes en les faisant évoluer dans le temps, elle permet d'éviter un type de météo ou au contraire d'en imposer un en permanence en jouant sur les pourcentages.

Puisqu'il s'agit de pourcentage, la somme des valeurs données ne doit pas excéder 100.

### 11.21.3 Connaitre le temps courant.

GetCurrentWeather (return un entier)

If ( GetCurrentWeather == 1)  
[c'est légèrement nuageux]

Cette fonction retourne un entier qui correspond aux conditions météorologiques actuelles, à voir en fonction des valeurs données plus haut et dans un tableau à la fin du tuto. Exemple d'utilisation, Bethesda utilise des tests sur le temps pour faire bouger les bannières à l'entrée des villes.

begin OutsideBanner

;ce script est attaché à un objet bannière

;bouge à cause du vent

;Idle est toujours actif, Idle2 est actif pour une petite brise, et Idle3 pour la tempête

short ran

if ( MenuMode == 0 )

set ran to random 100

if ( ran < 30 ) ;30% de chance que la bannière bouge

;contrôle l'activité météo de la zone

if (GetCurrentWeather >= 5 )

;orage, tempête de cendre ou fléau

LoopGroup, Idle3, 5

endif

```

;la dernière animation invoquée dans le script est celle qui est effectivement jouée
if ( ran <= 10 )
    PlayGroup, Idle

elseif ( GetCurrentWeather < 5 )
    PlayGroup, Idle2

endif

endif

endif

```

#### 11.21.4 Connaître la vitesse du vent

##### **Pas documentée:**

GetWindSpeed (retourne float)

Cette fonction retourne toujours 0 en intérieur et une valeur non entière en extérieur (elle varie rapidement, environ 2.0 pour un temps nuageux)  
(XPCagey l'a trouvée)

#### **11.22 Diverses fonctions et variables**

##### 11.22.1 Savoir si le joueur a ouvert un menu

MenuMode

MenuMode retourne 1 si le joueur a activé un menu ( par exemple l'écran d'inventaire). C'est un moyen rapide et efficace pour ne pas exécuter un script dans certaines conditions. Reportez vous au début de cette fonction.

```

If ( MenuMode == 1 )
    Return
Endif

```

##### 11.22.2 Savoir si le joueur est à l'intérieur

GetInterior (retourne 1 si le joueur est en intérieur, 0 sinon)  
If ( GetInterior == 1 )

**Pas documentée!** (Merci à XP-Cagey et Killgore)

Le script suivant est un exemple d'utilisation proposé par Killgore. Si vous désirez le tester, taper la commande "StartScript Outside\_Check" dans la console.

```

Begin Outside_Check
short doonce

if (MenuMode == 1)
Return
endif

if (doOnce == 0) ;si vous êtes dans une nouvelle cellule
;ou au tout début

if ( GetInterior == 1 )
MessageBox "1: dedans"
elseif ( GetInterior == 0 )
MessageBox "0: dehors"
else

```

```

MessageBox "la on ne sait pas"
endif

set doOnce to 1
Return
endif

if (doOnce == 1)
if (CellChanged == 0) Return
else ;si le joueur vient de changer de cellule
set doOnce to 2 ;attend une nouvelle image
endif
Return
endif

if (doOnce == 2) ;puis commence et affiche les resultats
set doOnce to 0
Return
endif

End Outside_Check
    
```

### 11.22.3 Interrompre l'exécution d'un script

```
Return
```

Return signale au moteur de jeu qu'il n'a plus rien dans ce script pour cette image.

```

If ( MenuMode == 1 )
    Return
Endif
    
```

Attention: tout ce qui se trouve après cette instruction ne sera pas exécuté.

### **11.23 Contrôler les scripts globaux**

```

ScriptRunning, ScriptName (retourne 1 si le script nommé est actif)

StartScript, ScriptName (début un script)

StopScript, ScriptName (l'interrompt)
    
```

On utilise ces fonctions pour contrôler l'exécution des scripts globaux.

```

if ( ScriptRunning, CharGen == 0 )
    StartScript CharGen
endif
    
```

### **11.24 Trucs et astuces**

Cette section devrait s'agrandir avec le temps. Elle présente des astuces d'utilisation des scripts pour obtenir des fonctionnalités originales. (Envoyez moi vos idées, et je les ajouterai en vous citant (bien entendu))

#### 11.24.1 Permettre à un acteur de changer d'armes (Ghan Buri Ghan – grand maître scripteur)

Peut être que l'utilisation de cette astuce disparaîtra avec Tribunal qui annonce avoir corrigé la fonction equip... En attendant.... Voici un exemple qui permet à un garde de changer d'arme (épée ou arc)

```
Begin_HB_CaravanGuard_AI
```

```

; ce script rend l'I.A des gardes plus dangereuse pour le joueur en lui permettant de choisir entre l'arc ou l'épée
en fonction de la distance du joueur
short currentarrows
short storearrows
short doonce

set currentarrows to GetItemCount "arrow of wasting flame"

if ( doonce == 0 )
    set storearrows to currentarrows
endif

if ( GetDistance, Player < 120 )
    set currentarrows to GetItemCount "arrow of wasting flame"
    if ( currentarrows > 0 )
        RemoveItem "arrow of wasting flame", 1
        set doonce to 1
    endif

elseif ( GetDistance, Player >= 120 )
    if ( currentarrows < storearrows )
        AddItem "arrow of wasting flame", 1
    else
        set doonce to 0
    endif
endif

End

```

L'exemple suivant est utilisé par Bethesda....:

```

begin marksmanToggle

short counter
short myMarksman

if ( MenuMode == 1 )
    return
endif

if ( counter < 20 )
    Set counter to counter + 1
    Return
endif

if ( myMarksman == 0 )
    set myMarksman to GetMarksman
endif

if ( GetMarksman > 0 )
    if ( GetDistance Player < 400 )
        SetMarksman 0
    endif
else
    if ( GetDistance Player > 600 )
        SetMarksman myMarksman
    endif
endif

endif

```

```
;pour les créateurs de mods, ceci permet de forcer l'I.A à faire ce qu'il faut
;en combat rapproche on utilise les armes de meles
;a distance on utilise les armes de jets
;il y a un controle toutes les 20 images pour la vitesse
;Note: ceci n'affecte pas l'I.A. pour les sorts.
```

End

#### 11.24.2 Une sequence cinématique- gianluca (Morrowind Summit forums)

Une façon de créer une sequence cinématique. Désactive les contrôles du joueurs, et le place dans une pièce de mur invisible, puis les déplace et donc la caméra avec...

```
if menumode==1
return
endif

if doOnce==0

"Collision wall2"->disable
"Collision wall3"->disable
"Collision wall4"->disable

set doOnce to 1

endif

if doOnce==1
"Collision wall1"->moveworld X 800
messagebox "moving"
if ("Player"->getPos Z < 570 )
set doOnce to 2
set playxx to "Player"->getPos X
set playyy to "Player"->getPos Y
set playzz to "Player"->getPos Z
"Collision wall2"->enable
"Player"->position -114679 -4119 590 90
endif
endif

if doOnce==2
"Collision wall2"->moveworld X 800
messagebox "moving"
if ("Player"->getPos Z < 570 )
set playxx to "Player"->getPos X
set playyy to "Player"->getPos Y
set playzz to "Player"->getPos Z
"Collision wall3"->enable
"Player"->position -112634 -4119 590 90
set doOnce to 3
endif
endif

if doOnce==3
"Collision wall3"->moveworld X 200
"Collision wall3"->moveworld Y -800
if ("Player"->getPos Z < 570 )
set doOnce to 4
set playxx to "Player"->getPos X
set playyy to "Player"->getPos Y
set playzz to "Player"->getPos Z
"Collision Wall4"->enable
"Player"->position -112126 -6150 590 90
endif
endif
```

```
endif
```

```
if doOnce==4  
"Collision wall4"->moveworld X 600  
"Collision wall4"->moveworld Y -450  
if ("Player"->getPos Z < 570 )  
set doOnce to 5  
set playxx to "Player"->getPos X  
set playyy to "Player"->getPos Y  
set playzz to "Player"->getPos Z  
endif  
endif
```

```
if doOnce==5  
stopscript ELDQ_visualforbattle  
endif  
end ELDQ_visualforbattle
```

## 12 Function / Variable

Cette annexe est une description des fonctions et des variables globales utilisées pour définir les **Speaker Condition** dans la fenêtre de dialogue. Il est possible de jouer sur 6 paramètres de ce type.

Remarque : il est possible d'ajouter ses propres variables globales dans un mod. Elles sont alors accessibles dans la liste des variables globales.

Le résultat est vrai si...

< la valeur testée est plus petite que la valeur donnée  
 > la valeur testée est plus grande que la valeur donnée.  
 = la valeur testée est égale à la valeur donnée.  
 != la valeur testée est différente à la valeur donnée.  
 <= la valeur testée est inférieure ou égale à la valeur donnée.  
 >= la valeur testée est supérieure ou égale à la valeur donnée.  
 Il faut d'abord choisir la famille des éléments que vous voulez tester:

### 12.1 Function – les fonctions

Cela vous permet de comparer un nombre à différentes caractéristiques du monde du jeu. Etant donnée que vous ne testez que des grandeurs numériques, les valeurs vraie et fausse sont représentées par 1 ou 0 :

Fonction invoquée	Effet
<b>Alarm</b>	la valeur du coefficient d'Alarm qui a été réglé dans la fenêtre AI. (pour un pnj)
<b>Alarmed</b>	la valeur est vraie, si le pnj a détecté un crime actuellement commis par le pj.
<b>Attacked</b>	retourne vrai si le pnj a déjà été attaqué par le pj.
<b>Choice</b>	cette condition marche avec la fonction "Choice", que vous utilisez dans le champ 'Result' de la fenêtre Dialogue. La fonction choice permet au joueur de choisir une réaction possible à propos d'un sujet de conversation. Chaque réaction est associée à une valeur numérique. Lorsque le joueur choisit une réaction, la valeur correspondante à son choix est attribuée à Choice. Après cela, le moteur du jeu parcourt à nouveau le sujet de conversation et affiche la première réponse dont la condition Choice est égale à la valeur sélectionnée
<b>CreatureTarget</b>	Retourne vrai si l'orateur vise une créature
<b>Detected</b>	Retourne vrai si le pnj a détecté la présence du joueur
<b>FactionRankDifference</b>	La différence de rang entre le joueur et le pnj ( par ex : le pnj a un rang égal à 5 et le pj un rang égal à 2, le résultat est de 3), si le joueur n'est pas membre de la faction du pj le résultat est -1.
<b>Fight</b>	Retourne la valeur de ce qui a été réglé dans la fenêtre IA.
<b>Flee</b>	La valeur du coefficient de fuite attribué au pnj dans la fenêtre IA.
<b>FriendHit</b>	si le joueur a attaqué le pnj alors qu'ils étaient amis ( par exemple, membre de la même faction) la valeur est égale à 0 ; de même si le pnj n'a pas été frappé par le joueur. Dans les autres cas, cela indique le nombre de coups portés par le pj au pnj.
<b>HealthPercent</b>	le pourcentage de PV qui restent au pnj.
<b>Hello</b>	La valeur de l'option « Hello » qui a été assignée au pnj dans la fenêtre IA.
<b>Level</b>	Le niveau du pnj
<b>PCAcrobatics</b>	La valeur de la compétence Acrobatie du PJ
<b>PCAgility</b>	La valeur de la caractéristique Agilité du PJ
<b>PCAlchemy</b>	La valeur de la compétence Alchimie du PJ
<b>PCAlteration</b>	La valeur de la compétence Altération du PJ
<b>PCArmorer</b>	La valeur de la compétence Armurier du PJ
<b>PCAthletics</b>	La valeur de la compétence Athlétisme du PJ
<b>PCAxe</b>	La valeur de la compétence Hâche du PJ
<b>PCBLightDisease</b>	Vrai si le PJ est affecté du Fléau
<b>PCBlock</b>	La valeur de la compétence Blocage du PJ

<b>PCBluntWeapon</b>	La compétence en arme de taille
<b>PCClothingModifier</b>	Somme des valeurs de ce que le PJ porte actuellement
<b>PCCommonDisease</b>	Vrai si le PJ est atteint d'une maladie commune
<b>PCConjuration</b>	La valeur de la compétence Conjuration du PJ
<b>PCCorpus</b>	Vrai si le PJ est atteint du Corpus
<b>PCCrimeLevel</b>	Le niveau de crime du PJ, reporte le nombre et la gravité des crimes commis par le PJ)
<b>PCDestruction</b>	La valeur de la compétence Destruction du PJ
<b>PCEnchant</b>	La valeur de la compétence Enchantement du PJ
<b>PCEndurance</b>	La valeur de la caractéristique Endurance du PJ
<b>PCExpelled</b>	Vrai si le PJ a été expulsé de la faction du PNJ
<b>PCFatigue</b>	Le niveau de fatigue du PJ
<b>PCHandToHand</b>	La valeur de la compétence corps à corps du PJ
<b>PCHealth</b>	Le nombre de PV du joueur
<b>PCHealthPercent</b>	Le taux de point de vie qui restent au joueur
<b>PCHeavyArmor</b>	La valeur de la compétence en Armure lourde du PJ
<b>PCIllusion</b>	La valeur de la compétence Illusion du PJ
<b>PCIntelligence</b>	La valeur de la caractéristique Intelligence du PJ
<b>PCLevel</b>	La niveau du PJ
<b>PCLightArmor</b>	La valeur de la compétence Armure Légère du PJ
<b>PCLongBlade</b>	La valeur de la compétence Lame Longue du PJ
<b>PCLuck</b>	La valeur de la caractéristique Chance du PJ
<b>PCMagicka</b>	La quantité de Magicka du joueur
<b>PCMarksman</b>	La valeur de la compétence d'Archer du PJ
<b>PCMediumArmor</b>	La valeur de la compétence Armure Moyenne du PJ
<b>PCMercantile</b>	La valeur de la compétence Marchandage du PJ
<b>PCMysticism</b>	La valeur de la compétence Mysticisme du PJ
<b>PCPersonality</b>	La valeur de la caractéristique Personnalité du PJ
<b>PCReputation</b>	La valeur de la réputation du PJ
<b>PCRestoration</b>	La valeur de la compétence Guérison du PJ
<b>PCSecurity</b>	La valeur de la compétence Sécurité du PJ
<b>PCSex</b>	Le sexe du PJ (0=male, et 1=female)
<b>PCShortBlade</b>	La valeur de la compétence lame courte du PJ
<b>PCSneak</b>	La valeur de la compétence discrétion du PJ
<b>PCSpear</b>	La valeur de la compétence Alchimie du PJ
<b>PCSpeechcraft</b>	La valeur de la compétence Eloquence du PJ
<b>PCSpeed</b>	La valeur de la caractéristique Vitesse du PJ
<b>PCStrength</b>	La valeur de la caractéristique Force du PJ
<b>PCUnarmored</b>	La valeur de la compétence sans armure du PJ
<b>PCVampire</b>	Indique si le PJ est un vampire ou non
<b>PCWillpower</b>	La valeur de la caractéristique Volonté du PJ
<b>RankRequirement</b>	Permet de vérifier si le PJ correspond aux exigences de prochain niveau pour la Faction( 0=ni la reputation dans la faction, ni les compétences ne sont suffisantes, 1=seules les compétences sont suffisantes, 2=seule la reputation est suffisante, 3=le PJ est qualifié)
<b>ReactionHigh</b>	Retourne la plus haute valeur de la réaction entre la faction du pnj et toutes les factions auxquelles le PJ appartient.
<b>ReactionLow</b>	Même fonctionnement que la fonction précédente, mais pour la valeur la plus basse
<b>Reputation</b>	Donne la reputation du joueur
<b>SameFaction</b>	Vrai si le pnj et le pj sont de la même faction
<b>SameRace</b>	Vrai si le pnj et le pj sont de la même race
<b>SameSex</b>	Vrai si le pnj et le pj sont du même sexe
<b>ShouldAttack</b>	Vrai si le pnj veut se battre avec le pj
<b>TalkedToPC</b>	Vrai si le pnj a déjà parlé au PJ.
<b>Weather</b>	Donne le temps dans la cellule courante.

**12.2 Global – les variables globales**

Il s'agit d'un ensemble de variables qui ne dependent pas du pnj auquel le PJ parle, de sa position dans le monde, etc...

<b>CharGenState</b>	Indique si le joueur a commence une nouvelle partie, i.e. si il a créé un nouveau personnage
<b>Day</b>	Le jour courant dans le jeu
<b>DestroyBligh</b>	Cette variable vaut -1, si la maladie du Fléau a été détruite
<b>DuelActive</b>	Indique si il y a actuellement un combat dans l'arène de Vivec.
<b>Exp[FactionName]</b>	Indique si le joueur a été expulsé d'une certaine faction
<b>fargothwalk</b>	Cette variable indique le progrès dans la quête confiée par Hrisskar pendant que Fargoth va se cacher
<b>FreedSlavesCounter</b>	Le nombre d'esclaves libérés
<b>GameHour</b>	L'heure du jour dans le jeu
<b>GG_Gate1_State</b>	Indique si la première porte de Ghostgate est fermé ou non
<b>GG_Gate2_State</b>	Indique si la deuxième porte de Ghostgate est fermé ou non
<b>HeartDestroyed</b>	Indique si le Cœur d'Akulakhan a été détruit
<b>HortatorVotes</b>	Indique le nombre de vote pour être Hortator chez les Telvanni
<b>MaduraRescued</b>	Indique si Madura Seran a été secouru (quête impériale)
<b>MonopolyVotes</b>	Le nombre de vote obtenu par le joueur, si le joueur a tué trop de monde qui pouvait voter pour lui, sa valeur est -1 (c'est une quête que je ne connais pas...)
<b>Month</b>	Le mois courant dans le jeu
<b>MT_LegitKills</b>	Le nombre d'assassinat commis pour la Morag Tong (les gentils assassins)
<b>MT_NewCrimeLevel</b>	Le nouvel indicateur de crime du joueur, après avoir été en prison
<b>PCRace</b>	La race du joueur
<b>PCVampire</b>	Indique si le joueur est un vampire ou non
<b>Random</b>	Crée une valeur aléatoire, on peut lui associer un coefficient multiplicateur. Random crée un nombre compris entre 0 et 1, la multiplie par le coefficient puis l'arrondi.
<b>ratskilled</b>	Le nombre de rat tués pendant que le joueur fait une quête pour Audenian Valius à Vivec
<b>RedoranMurdered</b>	Le nombre des Conseil Redoran tués par le joueur
<b>Rent_[Town]_[Tavern ShortCut]</b>	Indique si le joueur pris une chambre, dans une ville et dans une auberge
<b>Stronghold</b>	Indicateur de progression de la construction de votre Stronghold
<b>TelvanniDead</b>	Le nombre de conseillers Telvanni morts
<b>TGlove</b>	Vaut -1 si le joueur a équipé un gant : Right Bal Molagmer Glove
<b>TimeScale</b>	Le multiplicateur pour l'écoulement du temps dans le jeu par rapport au temps réel (la valeur standard est 30, ce qui signifie qu'une minute de temps réel correspond à 30 minutes dans le jeu)
<b>VampClan</b>	Le clan vampire auquel le joueur appartient
<b>VampKills</b>	Le nombre de vampire n'appartenant pas au clan du joueur et tués par lui
<b>WearingHelmHHDA</b>	Indicateur de port du casque dans la première quête Hlaalu
<b>WearingLegionUni</b>	Indique si le joueur porte une armure qui fait partie des armures impériales
<b>WearingOrdinatorUni</b>	Indique si le joueur porte un élément de l'armure d'Indoril
<b>WraithguardEquipped</b>	Indique si le joueur a équipé Wraithguard
<b>Year</b>	L'année dans le jeu

## 13 Interprétation de certaines valeurs données par les fonctions

### 13.1 A propos du temps

Valeur	Temps correspondant
0	Clair
1	Couvert
2	Brouillard
3	Overcast
4	Pluie
5	Tonnerre
6	Tempête de cendre
7	Tempête de cendre rouge (Fléau)

### 13.2 Les races

Valeur	Race correspondante
1	Argonien
2	Breton
3	Elfe noir (Dunmer)
4	Haut Elfe
5	Impérial
6	Khajit
7	Nord
8	Orc
9	RedGuard ( garde Rouge)
10	Elfe des bois

### 13.3 Les clans de Vampires

Valeur	Nom du clan
1	Aundae
2	Berne
3	Quarra

### 13.4 Les packages IA

Valeur	Package IA correspondant
-1	Aucun
0	Wander
1	Travel
2	Escort
3	Follow
4	Activate (utilisation d'un objet)
5	Pursue (ne peut pas être utilisé dans les scripts)

### 13.5 Les phases de la lune

Valeur	Lune ..
0	Nouvelle
1	Lune montante (premier quartier)
2	Première moitié

3	Lune descendante
4	Pleine lune

### 13.6 Les significations des coefficients de Wander (package IA)

Idle0	Ne fait rien
Idle2	regarde autour de lui
Idle3	regarde derrière lui
Idle4	se gratte la tête
Idle5	époussette ses habits ou son armure
Idle6	se frotte les mains (pratique pour un marchand)
Idle7	regarde ses doigts, puis furtivement autour de lui (prépare un mauvais coup)
Idle8	absorbé dans ses pensées
Idle9	cherche son arme

### 13.7 Les salutations (Greetings)

Greeting 0	Si le PJ est un criminel, ou les alarmes
Greeting 1	Apparemment des saluts vraiment spécifiques, quand le PJ a accepté une quête donnée par le PNJ...
Greeting 2	Le PJ est un vampire
Greeting 3	Si le PNJ est membre de la guilde Morag Tong (une seule salutation)
Greeting 4	Si le PJ est malade
Greeting 5	Ce qui se réfère aux quêtes
Greeting 6	Il y en a beaucoup, tout ce qu'il y a là dedans est relié aux quêtes
Greeting 7	Relié au faction apparemment
Greeting 8	Basés sur la réputation du personnage et sur son apparence extérieure (ses habits)
Greeting 9	Le genre de chose que les personnes pourraient vous dire dans les 'dernières rumeurs', et il y en a BEAUCOUP

## 14 Les effets des sorts - SpellEffect

Voici la liste des effets des sorts, elle est livrée telle quelle sans commentaire ( pas eu la force) :

Effet	Sort Associé	Effet	Sort Associé
sEffectWaterBreathing		sEffectDemoralizeCreature	
sEffectSwiftSwim		sEffectRallyHumanoid	
sEffectWaterWalking		sEffectRallyCreature	
sEffectShield		sEffectDispel	
sEffectFireShield		sEffectSoultrap	
sEffectLightningShield		sEffectTelekinesis	
sEffectFrostShield		sEffectMark	
sEffectBurden		sEffectRecall	
sEffectFeather		sEffectDivineIntervention	
sEffectJump		sEffectAlmsiviIntervention	
sEffectLevitate		sEffectDetectAnimal	
sEffectSlowFall		sEffectDetectEnchantment	
sEffectLock		sEffectDetectKey	
sEffectOpen		sEffectSpellAbsorption	
sEffectFireDamage		sEffectReflect	
sEffectShockDamage		sEffectCureCommonDisease	
sEffectFrostDamage		sEffectCureBlightDisease	
sEffectDrainAttribute		sEffectCureCorpusDisease	
sEffectDrainHealth		sEffectCurePoison	
sEffectDrainSpellpoints		sEffectCureParalyzation	
sEffectDrainFatigue		sEffectRestoreAttribute	
sEffectDrainSkill		sEffectRestoreHealth	
sEffectDamageAttribute		sEffectRestoreSpellPoints	
sEffectDamageHealth		sEffectRestoreFatigue	
sEffectDamageMagicka		sEffectRestoreSkill	
sEffectDamageFatigue		sEffectFortifyAttribute	
sEffectDamageSkill		sEffectFortifyHealth	
sEffectPoison		sEffectFortifySpellpoints	
sEffectWeaknessToFire		sEffectFortifyFatigue	
sEffectWeaknessToFrost		sEffectFortifySkill	
sEffectWeaknessToShock		sEffectFortifyMagickaMultiplier	
sEffectWeaknessToMagicka		sEffectAbsorbAttribute	
sEffectWeaknessToCommonDisease		sEffectAbsorbHealth	
sEffectWeaknessToBlightDisease		sEffectAbsorbSpellPoints	
sEffectWeaknessToCorpusDisease		sEffectAbsorbFatigue	
sEffectWeaknessToPoison		sEffectAbsorbSkill	
sEffectWeaknessToNormalWeapons		sEffectResistFire	
sEffectDisintegrateWeapon		sEffectResistFrost	
sEffectDisintegrateArmor		sEffectResistShock	
sEffectInvisibility		sEffectResistMagicka	
sEffectChameleon		sEffectResistCommonDisease	
sEffectLight		sEffectResistBlightDisease	
sEffectSanctuary		sEffectResistCorpusDisease	
sEffectNightEye		sEffectResistPoison	
sEffectCharm		sEffectResistNormalWeapons	
sEffectParalyze		sEffectResistParalysis	
sEffectSilence		sEffectRemoveCurse	
sEffectBlind		sEffectTurnUndead	
sEffectSound		sEffectSummonScamp	
sEffectCalmHumanoid		sEffectSummonClannfear	
sEffectCalmCreature		sEffectSummonDaedroth	
sEffectFrenzyHumanoid		sEffectSummonDremora	
sEffectFrenzyCreature		sEffectSummonAncestralGhost	
sEffectDemoralizeHumanoid		sEffectSummonSkeletalMinion	

Effet	Sort Associé	Effet	Sort Associé
sEffectSummonLeastBonewalker sEffectSummonGreaterBonewalker sEffectSummonBonelord sEffectSummonWingedTwilight sEffectSummonHunger sEffectSummonGoldensaint sEffectSummonFlameAtronach sEffectSummonFrostAtronach sEffectSummonStormAtronach sEffectFortifyAttackBonussEffectCommandCreatures sEffectCommandHumanoids sEffectBoundDagger sEffectBoundLongsword sEffectBoundMace		sEffectBoundBattleAxe sEffectBoundSpear sEffectBoundLongbow sEffectExtraSpell sEffectBoundCuirass sEffectBoundHelm sEffectBoundBoots sEffectBoundShield sEffectBoundGloves sEffectCorpus sEffectVampirism sEffectSummonCenturionSphere sEffectSunDamage sEffectStuntedMagicka	

## 15 Les fonctions de Script

Cette section parcourt l'ensemble des fonctions de script, donne leur utilisation et un exemple d'utilisation. Il s'agit d'une table de rappel des fonctions étudiées dans la parties Script du document principal.

### 15.1 I.A.

Signature	Ce qu'elle fait	Un exemple
AiActivate , ObjectID, [reset]	Le pnj ou la creature active l'objet ObjectID. L'activation correspond à: PNJ = lui parlerDialog Conteneur = l'ouvrir Porte = l'ouvrir Armes, armure,...= ramasser Livre/parchemins = lire	AiActivate, "in_dae_door_01"
AiEscort, ActorID, duration, x, y, z, [reset]	Escorte un pnj/une creature pendant un certain temps ou jusqu'à un lieu donné (spécifié en x,y,z). La fonction d'escorte permet de protéger également la cible escortée. La fonction AiFollow fonctionne de la même façon	AiEscort, "stargel", 30, -25668.379, -11195.562, 1138.720
AiEscortCell, ActorID, CellID, duration, x, y, z, [reset]	Escorte un pnj/une créature jusqu'à une cellule donnée (autre que la courante) et une fois dans cette cellule l'escorte encore pendant un certain temps, ou jusqu'à une position L'utilisation de AiFollowCell est le pendant de cette fonction	AiEscortCell, "stargel", "Moonmoth Legion Fort", 30, -4772.017, -18893.484, 1056.590
AiFollow, ActorID, duration, x, y, z, [reset]	Même chose que AiEscort mais sans la protection Rq : la cible peut aussi être le joueur dans tous les cas escorte ou suit.	AiFollow, "hlaalu guard_outside", 30, -25668.379, -11090, 1138.720
AiFollowCell, ActorID, CellID, duration, x, y, z, [reset]	Même chose que AiEscortCell mais sans la protection	AiFollowCell, "hlaalu guard_outside", "Moonmoth Legion Fort", 30, -4772.017, -18786, 1056.590
AiTravel, x, y, z, [reset]	Se déplace jusqu'à la coordonnée x,y,z	AiTravel -25668.379, -11090, 1138.720
AiWander, range, duration, time, [idle2], [idle3], . . . [idle9], [reset]	S'occupe autour de sa position actuelle, il peut se déplacer jusqu'à une certaine distance maximum, pendant un certain temps. Cette activité commence à une certaine heure du jour. Les autres paramètres sont optionnels. Apparemment le moteur de jeu lance un D100 et en fonction du résultat joue une animation différente : Idle2: regarde autour de lui Idle3: regarde derrière lui Idle4: se gratte la tête Idle5: change ses habits ou son armure Idle6: se frotte les mains (pratique pour un marchand) Idle7: regarde ses doigts, puis furtivemetn autour de lui (prépare un mauvais coup) Idle8: absorbé dans ses pensées	AiWander, 2000, 5, 0, 60, 20, 20, 20, 0, 0, 0, 20

	Idle9: cherche son arme C'est un point obscur de la doc, donc si quelqu'un a plus d'information...	
GetCurrentAIPackage	Donne le valeur du package AI actif (cf plus haut pour la signification)	if (GetCurrentAIPackage) . . .
GetAIPackageDone	Retourne 1 si l'exécution du package IA est achevée 0 sinon	if (GetAiPackageDone) . . .
GetDetected, ActorID	Retourne 1 si le pnj/creature peut détecter l'objet. Une fonction très lente : à éviter	if (GetDetected, "stargel")...
ForceSneak	Force le pnj à se déplacer furtivement	ForceSneak
ClearForceSneak	Annule la commade précédente	ClearForceSneak
GetForceSneak	Retourne 1 si le pnj est en mod furtif	if(GetForceSneak)...

### 15.2 Animation

PlayGroup, [Flags]	GroupName, Joue une animation pour un pnj, spécifie le groupe d'animation à effectuer. Un flag optionnel : 0 = termine ce qu'il est en train de faire puis fait l'animation demandée 1 = arête ce qu'il fait et joue l'animation depuis le début 2 = arrete l'animation au milieu et boucle depuis le début.	PlayGroup, Walk, 1
LoopGroup, Number, [Flags]	GroupName, Même chose qu'au dessus mais l'animation est répétée en boucle. La signifiacion du flage est la même.	LoopGroup, Walk, 11, 1
SkipAnim	Ignore l'animation courante pour cette image.	SkipAnim

### 15.3 Combat

GetAttacked	Retourne 0 l'acteur n'a jamais été attaqué 1 sinon	if(GetAttacked==0)
GetTarget, ActorID	Retourne 1 si l'acteur s'attaque à la cible donnée	if(GetTarget, "Stargel")
HitAttemptOnMe, ObjectID	Retourne 1 si l'acteur a failli être touché par cet objet	if (HitAttemptOnMe, "Silver Longsword")
HitOnMe, ObjectID	Retourne 1 si l'acteur a été touché par cet objet	if (HitOnMe, "Silver Longsword")
OnDeath	Retourne 1 si l'acteur est mort	if (OnDeath)
OnKnockout	Retourne 1 si l'acteur est sonné	if (OnKnockout==0)
OnMurder	Retourne 1 si l'acteur a été assassiné	if (OnMurder)...
OnPCHitMe	Ce n'est pas une fonction, mais est utilisé comme variable qui prend sa valeur quand l'objet est touché ( ???)	long OnPCHitMe if (OnPCHitMe) . . .
StartCombat, ActorID	Engage le combat avec le pnj/créature donné	StartCombat Player
StopCombat, ActorID	Arrête le combat avec l'acteur	StopCombat, Player

### 15.4 Dialogue

AddTopic, TopicID	Ajoute le topic à la liste des topic du PJ	Addtopic, "Orvas Dren"
ClearInfoActor	Pas très clair. Apparemment, l'info n'apparaît pas dans le journal. Utilisée dans la zone Result de la fenêtre de dialogue.	ClearInfoActor
ForceGreeting	Le pnj force le dialogue avec le PJ (même si ils ne sont pas au même endroit...télépathie)	ForceGreeting
GetJournalIndex, JournalID	Retourne la valeur du plus grand index dans le journal correspondant à L'ID	if (GetJournalIndex, A1_Dreams>5)
Goodbye	Force la fin du dialogue. Le PJ ne peut plus appuyer que sur « Au revoir »	Goodbye
Journal, ID, Index	Ajoute un index dans le journal sous la réf ID	Journal, A1_Dreams, 1
Choice, "Button1", Value . . . ["Button5"], value	Semblable a « Message Box » mais dans les dialogues. Permet au joueur de faire un choix	Choice, "Yes", 1, "No", 2

### 15.5 Faction

GetPCRank, [FactionID]	Donne le rang du PJ dans la faction. Si la faction n'est pas précisée, on prend celle de l'acteur. Retourne une valeur entre 0 et 9 et -1 si le PJ n'es pas membre de la faction	if (GetPcRank, Redoran >= 1)
GetPCFacRep, [FactionID]	?? probablement la reputation du joueur dans la faction	
GetRace, "RaceID"	Retourne 1 si le PJ est de la race donnée	if ( Player->GetRace "Argonian" == 1 )
LowerRank	Diminue d'un niveau le rang de l'acteur	Player->LowerRank
ModPCFacRep, var, [FactionID]	??Modifie la réputation du personnage d'un nombre de points dans la faction considérée	ModPcFacRep, 5, Redoran
ModFactionReaction, factionID1, factionID2, var	Modifie les réactions des factions ???	modFactionReaction, "Temple", "Nerevarine", 4
PCClearExpelled	Efface les informations d'expulsion du pj de la faction pnj	PCClearExpelled
PCExpell [FactionID]	Expluse le pj de la faction donnée (ou de celle de l'acteur si ce n'est pas spécifié)	PCExpell "Imperial Guard"
PCExpelled [factionID]	Retourne 1 si le pj a été exclu de la faction du pj	PCExpelled "Nerevarine"
PCJoinFaction [FactionID]	Le pj fait partie de la faction	PCJoinFaction "Redoran"
PCLowerRank	Rétrograde le PJ dans la faction du pnj	PCLowerRank
PCRaiseRank	Donne une promotion au PJ dans la faction. Si il n'en faisait pas partie lui donne le rang 1	PCRaiseRank
RaiseRank	Donne une promotion au PNJ dans sa faction	RaiseRank
SameFaction	Retourne 1 si le pj et le pnj sont membres d'une même faction	If (SameFaction == 1)
SetPCFacRep, var, [FactionID]	Fixe la réputation du pj dans la faction ( ??)	SetPCFacRep, 44, Redoran
SetFactionReaction, factionID1, factionID2, var	Modifie les réactions des factions entre elles	SetFactionReaction, "Temple", "Nerevarine", 55

**15.6 Item/Object**

AddItem, ObjectID, count	Ajoute un certain nombre d'objet dans l'inventaire de l'acteur	AddItem, "Silver Longsword", 1
Activate	Utilise l'actio par défaut de l'objet (voir au début pour les actions par défaut)	Activate
Drop, ObjectID, count	Pose des objets (un certain nombre) à ses pieds	Drop, "Silver Longsword", 1
Equip, ObjectID	Pas vu souvent, équipe l'objet sur son propriétaire	
GetItemCount, ObjectID	Retourne la quantité d'objet de ce type en possession de l'acteur	if (GetItemCount, "Silver Longsword">0)
OnActivate	Retourne 1 si l'objet est activé durant cette image	if (OnActivate)
OnRepair	Retourne 1 si l'objet est réparé durant cette image	if (OnRepair)
RemoveItem, objectID, count	Retire une certaine quantité d'objet du l'inventaire. Ils ne sont pas déposés, ils disparaissent	RemoveItem, &quot;Silver Longsword", 1
RepairedOnMe, objectID	Retourne 1 si l'objet est réparé par un objet du type objectID ???	If(RepairedOnMe, hammer_repair = 1 )
UsedOnMe, objectID	Retourne 1 un objet du type objectID est utilisé sur l'appelant ???	if(UsedOnMe,Misc_pot_redware_01)

**15.7 Les variables locales pour l'inventaire - Inventory Local Variables**

OnPCEquip	Le PJ vient de s'équiper d'un objet : reste vraie tant le PJ est équipé de cet objet	
OnPCAdd	Le PJ ajoute un objet dans son inventaire	
OnPCDrop	Le PJ se débarrasse (par terre) d'un objet	
OnPCRepair	Le PJ répare un objet	
OnPCSoulGemUse	L'objet doit être un SoulGem et elle est utilisée pour recharger un objet ou enchanter un objet	
PCSkipEquip	Assignée à 1 permet d'empêcher l'équipement d'un objet. Utile pour de pop-up qui informe qu'un objet est cassé	

**15.8 Magic**

AddSoulgem, CreatureID, SoulgemID	Ajoute une soulgem dans l'inventaire d'un acteur. La soulgem contient l'âme de la créature spécifiée	AddSoulGEM, rat, misc_soulgem_lesser
AddSpell, SpellID	Ajoute le sort donné à la liste des invocations de l'acteur	AddSpell, Burden
Cast, SpellID, TargetID	Invoke le sort donné et lance sur la cible	Cast Burden, "Adosi Darano"
DropSoulgem, CreatureID	Se débarrasse (à terre) de la sougem contenant l'âme de la créature	DropSoulgem, rat
DisableTeleporting	Désactive la téléportation	DisableTeleporting
EnableTeleporting	Active la téléportation	EnableTeleporting
GetBlightDisease	Retourne 1 si la créature est atteinte du Fléau (Blight)	if (GetBlightDisease) ...
GetCommonDisease	Retourne 1 si la créature est atteinte d'une maladie commune	if(GetCommonDisease)
GetEffect, Effect	Retourne 1 si l'acteur subit un effet spécifique	if (GetEffect, Burden)
GetSpell, SpellID	Retourne 1 si l'acteur dispose du sort donné	if (GetSpell, "calming touch")

GetSpellEffects, SpellID	Retourne 1 si l'acteur subit les effets d'un sort	if(GetSpellEffects, "Burden of Sin")
HasSoulgem, CreatureID	Retourne 1 si l'acteur a une soulgem contenant l'âme de la créature donnée	if (HasSoulgem, rat)
RemoveSoulgem, CreatureID	Retire de l'inventaire de l'acteur la soulgem contenant l'âme de la créature	RemoveSoulgem, rat
RemoveSpell, SpellID	Retire le sort donné de la liste des invocations de l'acteur	RemoveSpell, "cure poison touch"
RemoveSpellEffects, SpellID	Retire les effets du sort donné	RemoveSpellEffects, "curse endurance"
RemoveEffects, Effect	Retire les effets donnés	RemoveEffects, blind

### 15.9 Divers - Miscellaneous

Enable	Rend l'objet visible et actif dans le jeu	Enable
DontSaveObject	A appeler lorsque vous voulez empêcher la sauvegarde d'un objet	DontSaveObject
Disable	Rend l'objet inactif et hors fonction	Disable
FadeIn <time>	Fait apparaître l'objet avec un effet de fondu qui s'étale dans le temps ; time est compris entre 0 et 10.0	FadeIn 1.0
FadeOut time	Fait disparaître l'objet avec un effet de fondu (pareil que FadeIn)	: FadeOut 2.0
FadeTo alpha speed	Crée un fondu sur l'écran. Alpha prend une valeur entre 0 (clair) et 100 (tout noir)	FadeTo 50 2.0
GetButtonPressed	Retourne le choix depuis une MessageBox (0=premier choix, 1=2ème choix, ...)	if (GetButtonPressed=0)
GetDisabled	Retourne 1 si l'objet n'est pas actif.	if (GetDisabled)
GetLocked	Retourne 1 si l'objet est verrouillé	if (GetLocked)
GetPCCell, Cell ID	Retourne 1 si le PJ est dans la cellule correspondante. Le test peut aller en profondeur. Par exemple, à Vivec dans la Fred'House, la fonction retourne 1 pour Vivec	if (GetPCCell, "Balmora")
GotoJail	Envoie le PJ en prison	GotoJail
GetMasserPhase	Retourne la phase de la lune principale (cf tableau plus haut)	if(GetMasserPhase == 4 )
GetSecundusPhase	Retourne phase de la seconde lune	if(GetSecundusPhase == 4)
GetStandingPC	Retourne 1 si le PJ se tient sur l'objet	if (GetStandingPC)
GetStandingActor	Retourne 1 si un acteur (quel qu'il soit) se tient sur l'objet	if (GetStandingActor)
HurtStandingActor, value	Inflige des dégâts à tout acteur qui se tient sur l'objet. Il s'agit de dégâts par seconde et la valeur doit être négative	HurtStandingActor, -20
Lock, Var	Fixe le verrouillage du conteneur ou de la porte	Lock, 50
MenuMode	Retourne 1 si un menu est ouvert	if (MenuMode) return Endif

MessageBox, "Message", [var1], [var2], ["button1"], ["button2"]	Fait apparaître une boîte dialogue avec des choix pour le PJ. Il faut définir un message, et des choix à afficher qui sont associés à des boutons. Utilisez la fonction GetPressedButton pour connaître le choix du PJ. Vous pouvez utiliser %.2f pour afficher un chiffre avec deux chiffres après la virgule (%.0f pour aucun chiffre après la virgule)	MessageBox, "Dois je commencer à un tour? GameHour = %.2f", GameHour, "OK", "Non"
PayFine	Appeler cette méthode après que le joueur ait payé son amende, afin que l'I.A. sache que le joueur est OK	PayFine
PlayBink "filename" flag	Pour jouer une séquence video. Le flag indique si le joueur peut ou non l'interrompre.	PlayBink "mw_end.bik" 1
Random, Value	Retourne une valeur aléatoire comprise entre 0 et 1 et multipliée par Value	set myNumber to (random, 100)
ShowMap cellID	Monter toutes les cellules dont le nom commence par cette chaîne de caractère	ShowMap "Balmora"
Unlock	Déverrouille un conteneur ou une porte	Unlock
Xbox	Retourne 1 si c'est la version Xbox...	if (Xbox)

#### 15.10 Les mouvements et les déplacements

CellChanged	Retourne 1 pour une image après le changement de cellule	If (CellChanged)
CellUpdate	Met à jour les positions des différents objets à cause de problème qui arrivent parfois lorsque l'on déplace un objet sur de grande distance.	CellUpdate
GetPos, axis	Donne la position de l'objet dans le monde. Il est nécessaire de donner l'axe pour lequel la donnée nous intéresse	set myX to (getPos, x)
GetAngle, axis	Donne l'angle de l'objet par rapport à l'une des directions	set myAngleX to (getAngle, x)
GetLOS, ObjectID	Retourne 1 si l'on a une ligne de vision dégagée pour « voir » l'objet	if (GetLOS, Player)
GetDistance, ObjectID	Retourne la distance jusqu'à l'objet	if (GetDistance, Player < 200)
GetStartingPos, axis	Donne la position initiale de l'objet dans le monde. Celle qui a été attribuée dans l'éditeur. Précisez l'axe x, y ou z	set startX to (GetStartingPos, X)
GetStartingAngle, axis	Même fonction qu'au dessus mais pour l'angle	set stAglX to (GetStartingAngle,x)
Move, axis, units/sec	Déplace un objet le long de cet axe à la vitesse précisée. (il s'agit des axes de l'objet)	Move, y, 100
MoveWorld, axis, units/sec	Déplace un objet le long de cet axe à la vitesse précisée. (il s'agit des axes du monde)	Move, x, 100
PlaceAtPC, ObjectID, count, distance, direction	Place un objet au voisinage du PJ. Avec la quantité spécifiée, à la distance donnée et dans la direction imposée (0= devant, 1=derrière, 2= à gauche, 3 = à droite)	PlaceAtPC, "Silver Longsword", 1, 200, 2
Rotate, axis, angle/sec	Fait tourner un objet autour de son axe à la vitesse angulaire donnée	Rotate, x, 100
RotateWorld, axis, angle/sec	Pareil qu'au dessus mais autour de l'axe du monde.	Rotate, x, 100

Position, x, y, z, zRot	Téléporte l'objet à l'extérieur à la position donnée et regardant dans la direction angulaire donnée (autour de z)	<a href="#">Position 76345, 100.56, 215, 176</a>
PositionCell, x, y, z, zRot, "cellID"	Similaire à la commande précédente mais dans la cellule citée	<a href="#">PositionCell 76345, 100.56, 215, 176, "Gnisis Barracks"</a>
SetAtStart	Apparemment, renvoie les objets à leur place initiale	<a href="#">SetAtStart</a>
SetAngle, axis, angle	Attribue l'orientation angulaire de l'objet à la valeur donnée autour de l'axe cité	<a href="#">SetAngle, X, 35</a>
SetPos, axis, pos	Met l'objet à la position donnée sur l'axe choisit	<a href="#">SetPox, X, 2353</a>

### 15.11 Les contrôles du joueur-Player Controls

DisablePlayerControls	Désactive tous les contrôles du joueur. Il ne peut plus que regarder autour de lui. Il n'a plus accès au menu.	<a href="#">DisablePlayerControls</a>
DisablePlayerFighting	Le joueur ne peut plus se battre	<a href="#">DisablePlayerFighting</a>
DisablePlayerJumping	Le joueur ne peut plus sauter	<a href="#">DisablePlayerJumping</a>
DisablePlayerMagic	Le joueur ne peut plus faire de magie	<a href="#">DisablePlayerMagic</a>
DisablePlayerLooking	Le joueur ne peut plus regarder autour de lui	<a href="#">DisablePlayerLooking</a>
DisablePlayerViewSwitch	Le joueur ne peut plus changer le mode de vue ( 1 <sup>ère</sup> ou 3 <sup>ème</sup> personne)	<a href="#">DisablePlayerViewSwitch</a>
DisableVanityMode	Le joueur ne peut plus passer en vue à la troisième personne	<a href="#">DisableVanityMode</a>
EnableLevelUpMenu	Fait apparaître le menu de passage de niveau	<a href="#">EnableLevelUpMenu</a>
EnablePlayerControls	Réactive les contrôles du joueur	<a href="#">EnablePlayerControls</a>
EnablePlayerJumping	Le joueur peut de nouveau sauter	<a href="#">EnablePlayerJumping</a>
EnablePlayerFighting	Le joueur peut de nouveau se battre	<a href="#">EnablePlayerFighting</a>
EnablePlayerLooking	Le joueur peut de nouveau regarder autour de lui	<a href="#">EnablePlayerLooking</a>
EnablePlayerMagic	Le joueur peut de nouveau faire de la magie	<a href="#">EnablePlayerMagic</a>
EnablePlayerViewSwitch	Le joueur peut de nouveau de nouveau changer les modes de vue	<a href="#">EnablePlayerViewSwitch</a>
EnableRest	Le joueur peut de nouveau se reposer	<a href="#">EnableRest</a>
EnableVanityMode	Le joueur peut de nouveau se mettre en vue à la troisième personne	<a href="#">EnableVanityMode</a>
GetPlayerControlsDisabled	Retourne 1 si le joueur n'a plus de contrôle	<a href="#">GetPlayerControlsDisabled</a>
GetPlayerFightingDisabled	Retourne 1 si le joueur ne peut plus se battre	<a href="#">GetPlayerFightingDisabled</a>
GetPlayerJumpingDisabled	Retourne 1 si le joueur ne peut plus sauter	<a href="#">GetPlayerJumpingDisabled</a>
GetPlayerMagicDisabled	Retourne 1 si le joueur ne peut plus faire de magie	<a href="#">GetPlayerMagicDisabled</a>
GetPlayerLookingDisabled	Retourne 1 si le joueur ne peut plus regarder autour de lui	<a href="#">GetPlayerLookingDisabled</a>
GetPlayerViewSwitch	Retourne 1 si le joueur ne peut plus changer de mode de vue	<a href="#">GetPlayerViewSwitch</a>
GetVanityModeDisabled	Retourne 1 si le joueur ne peut plus passer en vue à la troisième personne	<a href="#">if (GetVanityModeDisabled)...</a>

PCGet3rdPerson	Retourne 1 si le joueur est en vue à la troisième personne	if (PCGet3rdPerson)
PCForce3rdPerson	Force la vue à la troisième personne	PCForce3rdPerson
PCForce1stPerson	Force la vue à la première personne	PCForst1stPerson
'GetInterior'	Retourne 1 si le joueur est dans une cellule intérieure	
'GetHealthRatio'	Retourne la fraction de PV qui reste à l'acteur (utile pour la fuite...)	

#### 15.12 Le repos du guerrier

GetPCSleep	Retourne 1 si le PJ se repose	if (GetPCSleep)
ShowRestMenu	Fait apparaître le menu de repose (même quand c'est illégal)	ShowRestMenu
WakeUpPC	Réveille le PJ	WakeUpPC

#### 15.13 Le temps qui passe

GetSecondsPassed	Retourne le nombre de secondes écoulées depuis la dernière image	Set Timer to (timer + GetSecondsPassed)
------------------	--	---

#### 15.14 Les tests sur les scripts

ScriptRunning, ScriptName	Retourne 1 si le script nommé s'exécute.	If ( ScriptRunning, MessageTest == 1 )
StartScript, ScriptName	Commence un script GLOBAL	StartScript, GrandCouncil
StopScript, ScriptName	Interrompt un script GLOBAL	StopScript, GrandCouncil

#### 15.15 Les bruitages

GetSoundPlaying soundID	Retourne 1 si le SoundID est actuellement joué.	if (GetSoundPlaying, "volcano rumble")
PlayLoopSound3D, "soundID"	Joue en boucle un son en mode 3D	PlayLoopSound3D "volcano rumble"
PlayLoopSound3DVP, "soundID", volume, pitch	Joue en boucle un son en mode 3D et ajuste le volume et le pitch	PlayLoopSound3DVP, "volcano rumble", 89, 12
PlaySound, "soundID"	Joue le son identifié	PlaySound "volcano rumble"
PlaySoundVP, "soundID", volume, pitch	Joue un son en ajustant le volume et la pitch	Playsound "volcano rumble", 89, 12
PlaySound3D, "soundID"	Joue un son en mode 3D	PlaySound3D, "volcano rumble"
PlaySound3DVP, "soundID", volume, pitch	Joue un son en 3D (pas en boucle) en ajustant le volume et le pitch	PlaySound3DVP "volcano rumble", 89, 12
Say, "file name", "text"	Dit un texte et fait apparaître le texte en sous titre (puis je suggérer de faire parler que les objet animé)	Say "bonjour.wav", "Bonjour Madame"
SayDone	Retourne 1 si l'objet a finit de dire son texte	if (SayDone)
StreamMusic, "filename.ext"	Joue une musique en arrière plan sonore	StreamMusic, "party.mp3"
StopSound, SoundID	Interrompt un son	Stopsound, "volcano rumble"

### 15.16 Statistiques

GetDeadCount, ObjectID	Donne le nombre d'objet ayant la même ID tués par le joueur	if (GetDeadCount, "Imperial Guard" > 1)
Resurrect	R&mène un acteur à la vie...	Resurrect

Il est possible de modifier les caractéristiques du joueur. Ses statistiques. Employer les mots Get, Set ou Mod suivis de la statistique qui vous intéresse.

Par exemple:

ModStrength -10 (retire 10 à la force du PJ)

Caractéristique du PJ	Compétences du PJ		Des modificateurs du PJ		
Strength Intelligence Willpower Agility Speed Endurance Personality Luck	Block Armorer MediumArmor HeavyArmor BluntWeapon LongBlade Axe Spear Athletics Enchant Destruction Alteration Illusion Conjuration	Mysticism Restoration Alchemy Unarmored Security Sneak Acrobatics LightArmor ShortBlade Marksman Mercantile Speechcraft HandToHand	Etat du PJ	Fatigue Health Magicka Level – ne peut pas être modifié par Mod...	
			Modificateurs de combat	AttackBonus DefendBonus ResistMagicka ResistFire ResistFrost ResistShock ResistDisease ResistBlight ResistCorpus ResistPoison ResistParalysis Chameleon	ResistNormalWeapons WaterBreathing WaterWalking SwimSpeed SuperJump Flying ArmorBonus CastPenalty Silence Blindness Paralysis Invisible
			IA	Fight Flee Alarm Hello	Attention: il s'agit de propriétés IA des objets. Les modifications portent sur toutes les référence qui existent dans le monde du jeu
			Divers	Reputation Disposition PCCrimeLevel – ne peut être utilisé que pour le PJ	

### 15.17 La pluie et les tempêtes de cendre

ChangeWeather, RegionID, TypeEnum	Change le climat d'une REGION, donnez le temps qu'il fait par un choix dans le tableau des climats	ChangeWeather, "West Gash", 3
GetCurrentWeather	Donne le temps actuel (là où est le joueur)	Example: if (GetCurrentWeather == 7). .
ModRegion, RegionID, clear_var, cloudy_var, foggy_var, overcast_var, rain_var, thunder_var, ash_var, blight_var	Modifie le climat général d'une région en attribuant les pourcentages à chaque type de climat	ModRegion, "West Gash", 10, 20, 10, 5, 5, 40, 10, 0

### 15.18 Console (dans le jeu uniquement)

CenterOnCell (coc), CellID	Place le PJ au centre de la cellule nommée	CenterOnCell "Moonmoth Legion Fort"
CenterOnExterior (coe), X, Y	Place le PJ dans le cellule à la position X,Y (par rapport à l'origine au sud de Caldera)	Example: Coe -10, -5
CreateMaps "Filename.esp"	Crée les images de cartes (pour Xbox)	
FillJournal	Ajoute toutes les entrées dans le journal (Attention c'est long)	FillJournal
FillMap	Ajoute toutes le ville sur la carte principale	FillMap
FixMe	Déplace de 128 unités le PJ. Utile pour se sortir d'un endroit où les collisions nous coincent	FixMe
GetFactionReaction factionID factionID	Les noms de faction ne sont pas optionnels. Cela ne marche qu'en mode console	GetFactionReaction Nerevarine, Redoran
Help	Affiche les raccourcis pour la plupart des commandes.	Help
Show	???????	????
ShowVars (sv)	Montre les valeurs des variables locales	Showvars
StopCellTest (sct)	Interrompt le test en cours d'une cellule	sct
TestCells (tc)	Déplace le joueur dans toutes les cellules pour les tester	TC
TestInteriorCells (tic)	Déplace le joueur dans toutes les cellules intérieures pour les tester	tic
TestModels (t3d)	Vérifie les modèles ???	
ToggleAI (ta)	Interrompt ou réactive (toutes ?) les AI	ta
ToggleBorders (tb)	Lorsque les bords sont actifs, une ligne jaune marque les frontières des cellules	Toggleborders
ToggleCombatStats (tcs)	Affiche les statistiques de combat dans la console	TCS
ToggleCollision (tcl)	Lorsque c'est désactivé, on peut tout traverser et voler	tcl
ToggleCollisionBoxes (tcb)	Affiche les blocs de collision autour des objets (la où il y a des faces)	tcb
ToggleCollisionGrid (tcg)	Affiche un paquet de nombre a l'écran	tcg
ToggleDebugText (tdt)	Affiche la direction, l'angle de vue et d'autre info à l'écran	Tdt
ToggleDialogueStats (tds)	Montre des informations sur les dialogues dans la console	Tds
ToggleFogOfWar	Active ou désactive le brouillard de guerre	tfow
ToggleFullHelp (tfh)	Affiche le propriétaire et le script associé quand la souris passe sur un objet.	tfh
ToggleGodMode (tgm)	Active ou désactive le mode invincibilité	tgm
ToggleGrid (tg)	Des chiffres ....	
ToggleKillStats (tk)	Affiche les statistique à propose de tueries dans la console	tk
ToggleLoadFade	???	ToggleLoadFade
ToggleMagicStats (tms)	Montre les informations de magie dans la console	Tms
ToggleMenus (tm)	Dés/active les menus	
ToggleScripts	Dés/active tous ( ?) les scripts	
ToggleStats (tst)	Dés/Active les menus de statistique	
ToggleSky(ts)	Affiche ou non le ciel	
ToggleTextureString (tts)	???	
ToggleWorld (tw)	Retire tout le monde. Ne reste que le ciel et l'océan	
ToggleWireframe (twf)	Passe en mode fil de fer	
TPG : Toggle path grid display	Affiche les grilles de déplacement dans les cellules	

SG	Montre les membres du group de l'acteur (?)	
ST	Montre les cibles de l'acteur	
ShowScenegraph (ssg)	Fait tout planter...	

## 16 Faire un mod propre

### 16.1 Un mod “sale”

Il est extrêmement facile de produire un mod de taille gargantuesque. Le TESCS enregistre dans un mod toutes les modifications, même accidentelle (il ne peut pas savoir qu’elles le sont), faites par rapport au fichier maître.

Si sans le vouloir, vous déplacer une table dans un cellule, vous la remettez à la même place (ou à peu près), cela ne changera rien au déroulement du jeu, pourtant cette modification est enregistrée dans votre mod.

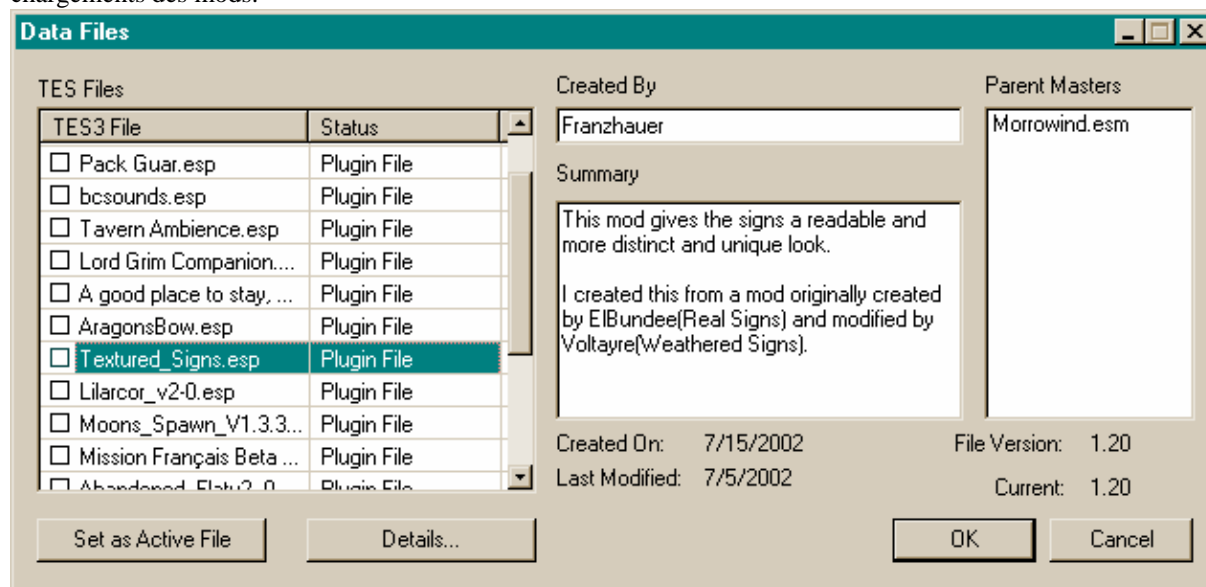
Le plus dangereux et le plus pervers reste les modifications que vous faites sans le vouloir sur les caractéristiques (hors celle qui touchent aux références) d’un objet peuvent être infime et pourtant perturber le jeu.

Il est possible que cela affecte tout l’univers de Morrowind, sans même parler des univers, des quêtes d’autres moddeurs, voire même des sauvegardes des joueurs.

### 16.2 Analyse des modifications faites

Il existe un outil fournit avec le TESCS qui vous permet de connaître l’ampleur des modifications que vous apportez à l’univers.

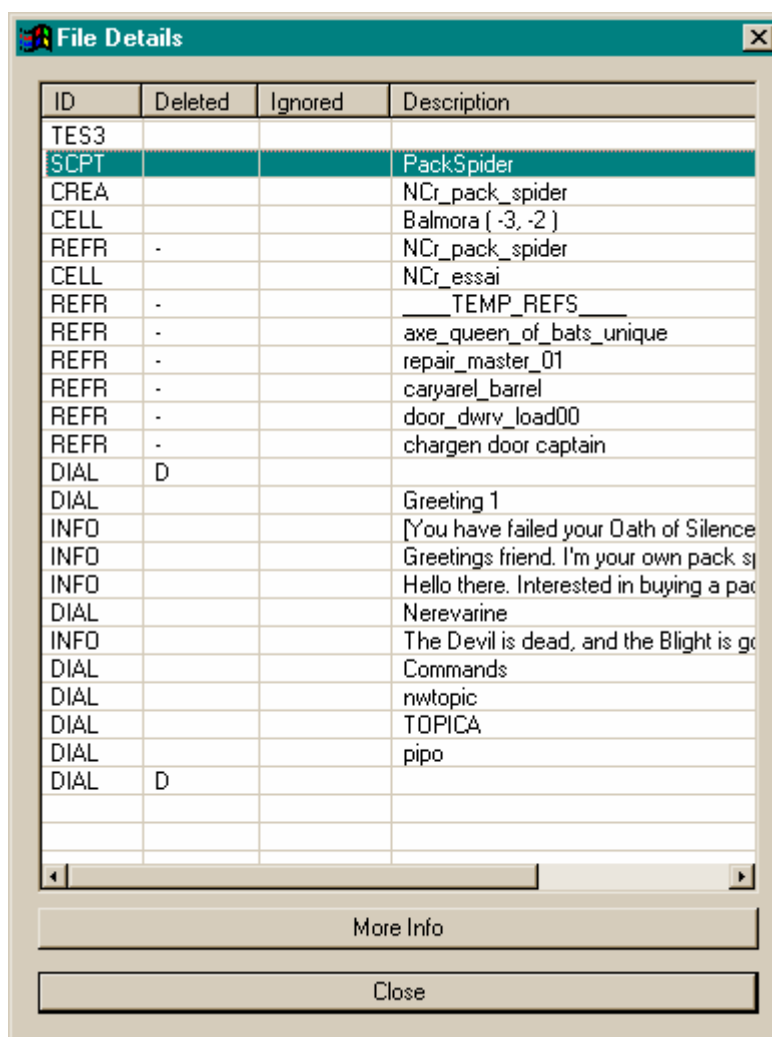
Sauvegardez votre ESP. Puis dans le menu File sélectionnez Data File. Cela fait apparaître la fenêtre de chargements des mods.



Sélectionnez votre mod et cliquez sur le bouton ‘Details’.

Cela fait apparaître la liste de tout ce que votre mod modifie. Nous allons apprendre à comprendre la signification des éléments de cette liste.

Si vous voyez une liste très longue alors que vous n’avez ajouter qu’une table dans une cellule... c’est qu’il y a un problème quelque part.



Dans ce cas, il vous reste trois alternatives :

- mettre un mod sale en téléchargement
- recommencer à zéro
- retirer les entrées inutiles à l'aide d'un éditeur Hexadécimal (ce n'est pas aussi dur que cela en a l'air).

### 16.3 Les fonctions de test fournies avec le TESCS

Le TESCS ne vient pas seul, il est accompagné de fonctions pour le test des mods que nous développons. Elles sont accessibles par le menu World/World Testing.

Une chose à avoir absolument avant d'utiliser ces fonctions ; chacune d'elle réalise les tests sur l'ensemble du monde, et pas uniquement sur votre mod. En conséquence il faut être vraiment sûr de vouloir tester avant de tester.

Toute activité de Test est très gourmande en temps, et c'est un euphémisme. En clair, n'utilisez ces fonctions que si vous êtes persuadés à 500% de l'utilité du test. Et dans tous les cas, faites une sauvegarde de votre plug avant de lancer un test, on a trop souvent vu un moddeur assassiner ses dernières modifications en utilisant le gestionnaire de tâche pour interrompre sauvagement l'éditeur.

Pour compléter ce tutorial, je vais néanmoins vous présenter les rôles des différents tests proposés. Sachez simplement que je décline toute responsabilité. Ce sont des outils puissants mais LENTS. Leur utilisation doit être réservée pour des modifications profondes et multiples. (Plusieurs cellules modifiées...grosses modif sur les scripts...). Dans de nombreux cas, il est plus simple et plus rapide de lancer le jeu et tester vous-même votre plug, que cela ne vous empêche pas d'être sérieux et exigeant envers votre mod.

#### 16.3.1 Model Test – le test des objets 3D

Vérifie que tous les objets 3D du monde ont une ID valide.

Si il manque des objets dans votre univers, mains, membres ou toute autre chose, vous devriez connaître avec précision l'origine du problème.

#### 16.3.2 Dialogue and Conflicts – Les problèmes dans les dialogues.

Cette fonction crée une liste de toutes les entrées de Dialogues qui partagent une ID avec un élément (quel qu'il soit) dans le monde. Cela pourrait poser des problèmes qui peuvent apparaître en utilisant la fonction replace ID dans un script.

De façon toute à fait primaire, c'est un moyen de vous prévenir contre les fonctions que vous utilisez dans les scripts pour modifier des instance. Ce test montrera les endroits où vous utilisez une ID pour deux dialogues différents.

Utiliser ce test pour une simple création de réplique peut sembler exagéré, mais cela évite les conflits et les actions et interférences non voulues.

Cette fonction vous permettra de déceler des problèmes qui n'apparaissent qu'en cours de jeu, et qui sont autrement difficiles à détecter.

#### 16.3.3 Non teleporting door – les portes non reliées à des cellules

Cette fonction crée une liste de toutes les portes de l'extérieur qui ne sont pas reliées à des cellules.

Il est en effet absurde d'avoir des portes en extérieur qui ne sont reliées à aucune cellule d'intérieur, ce test vous permet de mettre en évidence les localisations de ces portes que vous auriez placées par accident. Ce test vérifie la liste complète des cellules d'extérieur, c'est long. Et il est souvent plus rapide de vérifier « à la main » les cellules qui ont été modifiées (elles postfixées d'un astérisque \*).

Pour éviter les ennuis détectés par cette fonction, je peux vous conseiller de ne jamais relier votre cellule à l'extérieur avant d'être certain de l'endroit où vous vouliez la mettre.

#### 16.3.4 Non teleporting interiors – les intérieurs isolés

Cette fonction est le pendant de la précédente, en effet elle permet de récupérer la liste des cellule d'intérieurs qui ne sont reliées à rien.

Evidemment cette fonction n'est pas infaillible, il suffit d'avoir deux cellules d'intérieurs correctement liées entre elles pour qu'elles n'apparaissent pas dans la liste, et ceci même si finalement, il n'existe aucun moyen de s'y rendre depuis l'extérieur.

N'utilisez cette fonction que si vous avez créé un donjon énorme, pour vérifier que les cellules sont bien connectées entre elles et que vous ne laissez aucune cellule inaccessible depuis l'extérieur.

#### 16.3.5 Test All Lands – verification des paysages

N'utilisez cette fonction que si vous avez créé ou modifier un grand nombre de cellules d'extérieur, l'utilisation de cette fonction retournera une liste contenant l'ensemble des cellules qui ont des problèmes de frontières, trous ou autre chose....

#### 16.3.6 Test/Fix All path grid

Parcourt la liste de tous les chemins et élimine les points qui sont sous l'eau et tous les points qui ne sont connectés à rien.

#### 16.3.7 Test all cells

Ce test n'est à effectuer que si vous avez beaucoup de temps à perdre, et encore....

Ce test permet de vérifier la cohérence graphique des cellules Il simule le déplacement d'une caméra dans chaque cellule pour vérifier que les frontières sont bien placée ce test est effectuer pour vérifier que les animations se déroulent sans accro.

#### 16.3.8 Test All Interior Cell

Fait la même chose que la fonction précédente mais pour les cellules d'intérieurs.

#### 16.4 Utiliser un éditeur hexadécimal pour nettoyer un mod

Nous allons ensemble voir comment on peut modifier le fichier esp d'un mod, à l'aide d'un éditeur hexadécimal. Pour ma part, j'utilise une version de Hex WorkShop.

L'ensemble des éditeurs hexa se présente de la même façon, il ne sera donc pas trop difficile de se raccrocher aux explications avec un autre éditeur.

Nous allons travailler sur le fichier esp qui correspond à la capture d'écran au dessus.

La liste doit pouvoir se comprendre sans trop de mal, mais nous y reviendrons quand même pour expliquer chacune des lignes.

Ce mod est un support que j'utilise pour faire des tests, on peut donc enlever tout ce qui s'y trouve.

Les noms utilisés sont absurdes et lorsque je fait un mod, j'utilise une convention de nomage qui me permet de retrouver assez rapidement les informations qui m'intéressent. Je vous conseille de faire de même, par exemple en préfixant tout les noms (et ID) que vous créez de quelques lettres commune (par exemple ; ici j'ai choisit NCr\_...).

Voici la démarche de nettoyage que je vous propose :

- regarder chaque ligne de la liste

- pour chacune posez vous la question : Est-ce un nouvel objet ? si oui en ai-je vraiment besoin ?

si il s'agit d'un objet existant et que vous le modifiez, alors posez vous trois la question de la nécessité de faire une modification sur cet objet !!

- pour chacune de ces lignes si la réponse est non : alors la ligne est probablement superflue. Notez sa référence et continuez.

Lorsque vous découvrez un objet superflu, allez dans la cellule dans laquelle il est ajouté supprimez le simplement de la liste des références.

Pour les programmeurs de script, vous devez comprendre la nécessité qu'un objet ai toujours un nom unique. Dans la plupart des cas, pour éviter de modifications trop profonde du jeu, je vous conseille de créer un nouvel objet chaque fois que vous voulez attribuer un script à un objet (imaginer que tous les coffres de l'île se trouvent affublés d'un génie qui en sort parce que vous vouliez le mettre dans le coffre de votre mod...)

Ne vous contentez pas de changer le nom et les propriétés d'un objet pour satisfaire vos besoins, cela n'est ni propre, ni élégant...

Cette remarque s'applique également à tous les conteneurs, faites attention à ne modifier que les informations pour une référence donnée et pas pour tout l'univers.

Deux scénarios catastrophes peuvent se présenter :

1) Lors de l'édition de votre mod, vous manipulez des objets et accidentellement vous renommez misc\_com\_bottle\_05. Aïe, ce que vous vouliez faire était simplement renommer une copie et pas l'original. CE n'est pas grave, vous renommez la copie et c'est bon, pensez vous. Erreur, votre mod contient des modifications pour les 991 références à misc\_com\_bottle\_05. Pour le TESCS, vous n'avez pas annulé une action, vous avez exécuté deux actions différentes (qui de votre point de vue s'annulent). Si vous essayez cette manipulation, regardez la liste des modifications.... c'est énorme, et votre fichier esp a probablement doublé de taille.

2) Vous pensez avoir annulé des modifications faites sur un objet, mais vous avez oublié une modification critique... votre mod est un cauchemar pour tous les joueurs de MW.

Premier conseil, le plus important créer vos propres objets. Ne modifier pas les originaux. Garder une liste de ce que vous faites, créer....

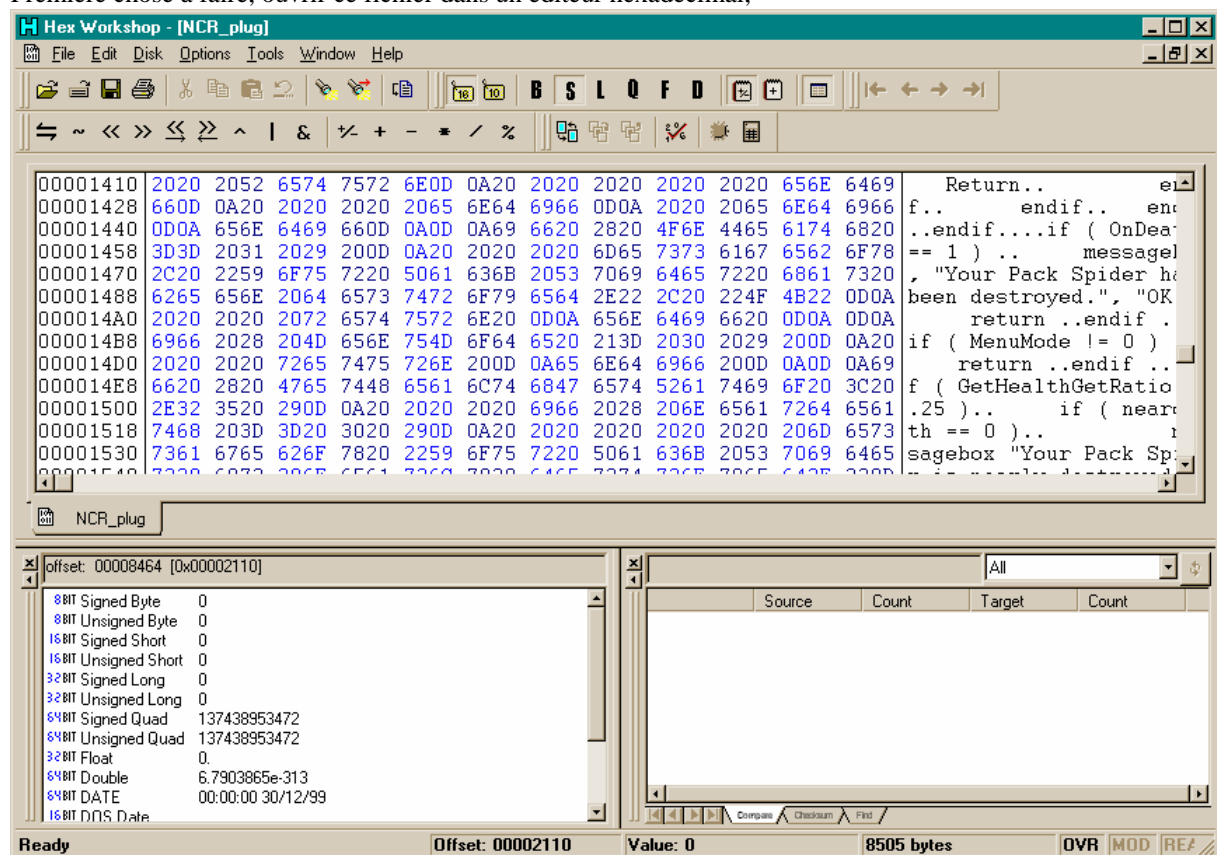
Deuxième conseil, assurez vous avant de faire la distribution de votre mod, via la fenêtre de Détails qu'il n'y a pas de choses inutiles dans votre mod.

Voyons comment utiliser un éditeur hexadécimal pour corriger les esp de manière très fine. La première chose est de s'assurer que l'on connaît un peu le format de ces fichiers (une annexe à ce tutorial, contient une description partielle de ce format).

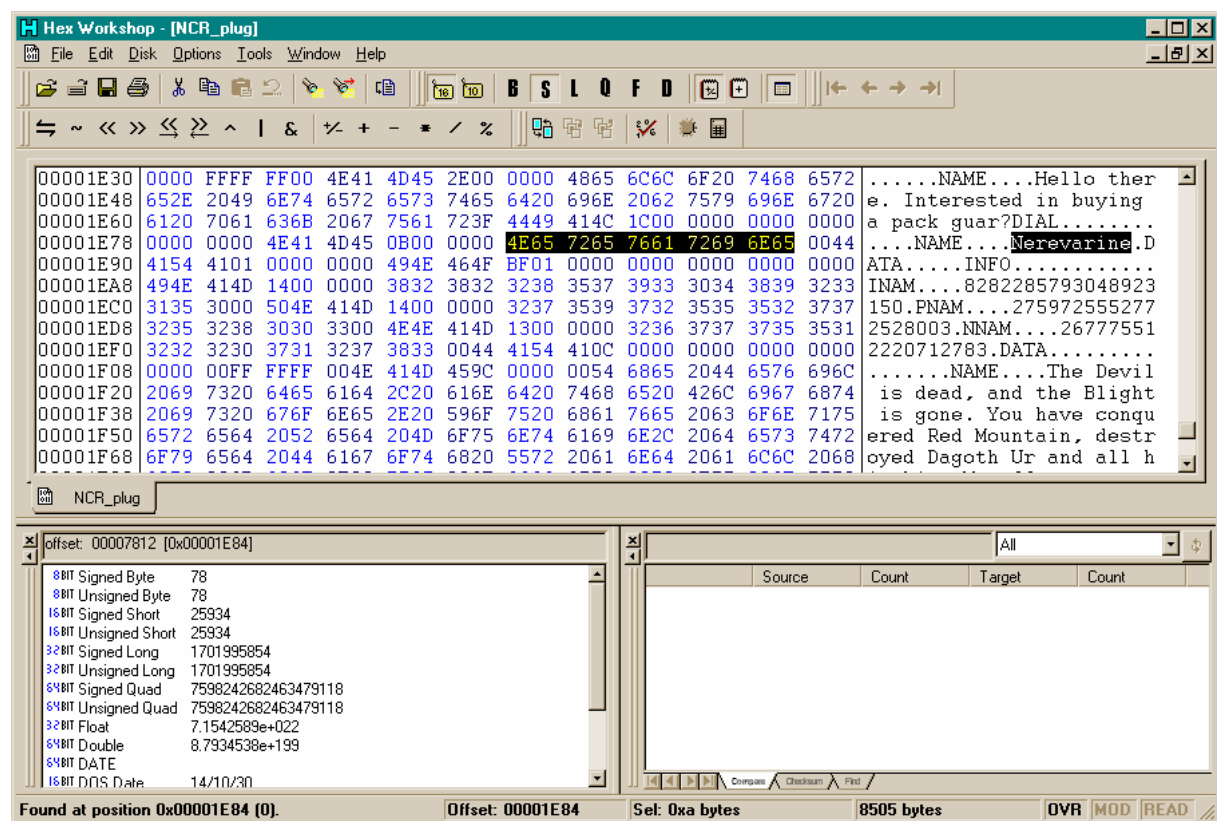
Nous allons retirer les modifications faites dans la base de dialogue concernant le Nerevarine...

DIAL			Nerevarine
INFO			The Devil is dead, and the Blight is go
DIAL			Commands

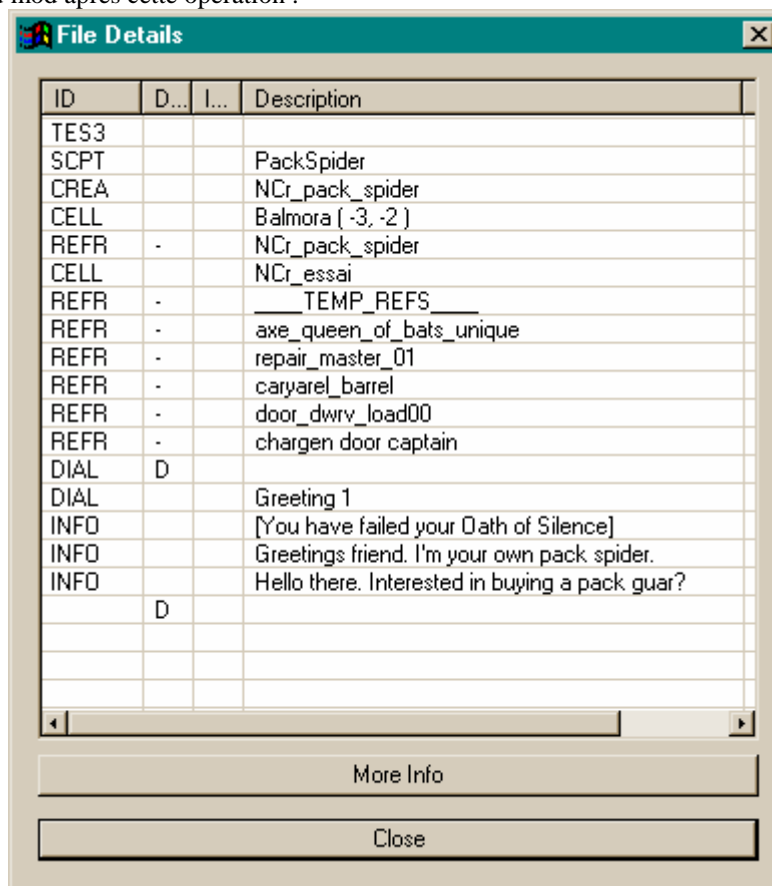
Première chose à faire, ouvrir ce fichier dans un éditeur hexadécimal,



Puis trouver la référence aux modifications dans le dialogue : (en utilisant la fonction Find, cela va tout seul)



Ensuite on remplace ce que l'on veut supprimer par des espaces. On se référera utilement à la description de la structure des esp pour éviter les bêtises, on sauvegarde les modifications faites et on recharge dans le TESCS... Voici les Détails du mod après cette opération :



La zone que l'on désirait supprimer a disparu.

Faites attention toutefois, ce genre de manipulation peut être risquée si vous faites une erreur vous pourriez tout perdre.

Garder donc en vue la structure des ESP lorsque vous faites ces modifications.

#### 16.5 TESAME – The Elder Scroll Advanced Mod Editor

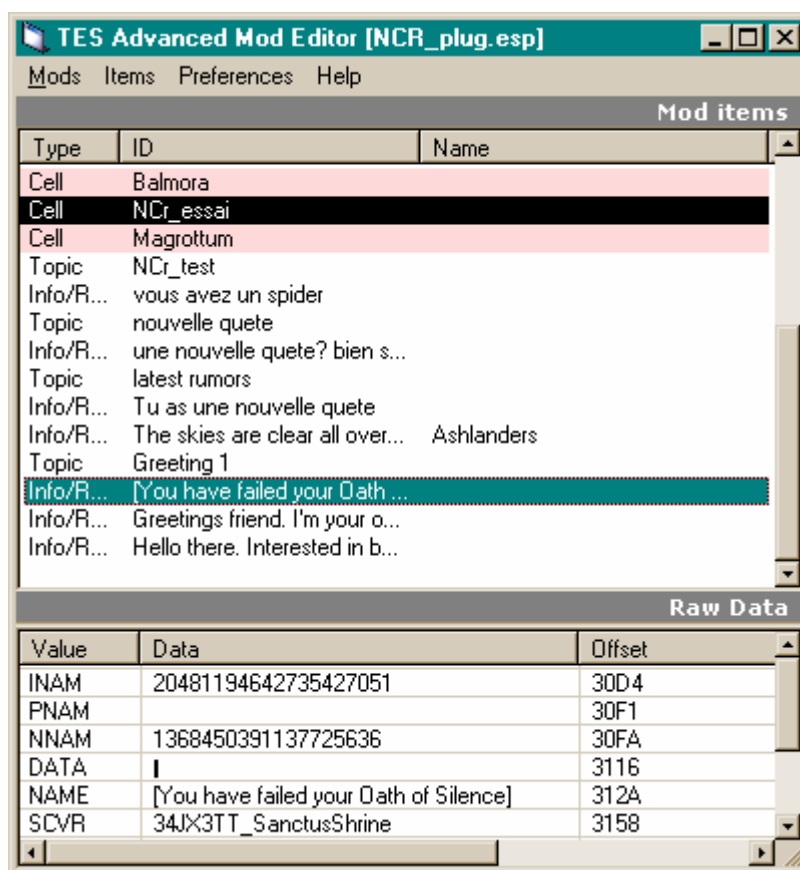
Voilà, maintenant que vous connaissez la manière de modifier un mod en vous prenant la tête... (si, on peut dire que c'est une grosse prise de tête d'utiliser un éditeur hexadécimal pour faire ce boulot de dingue).

Faites tout de même attention en utilisant ce soft, il ne s'appelle pas éditeur avancé pour rien...

Je vais vous parler d'un utilitaire que vous trouverez dans toutes les bonnes épiceries... TESAME. C'est un utilitaire qui a été développé pour permettre de nettoyer les mods réalisés. Il permet notamment d'éliminer les modifications involontaires que vous auriez pu faire.

TESAME, The Elder Scroll Advanced Mod Editor.... Tout est dans le nom.

Maintenant que vous êtes alléchés, je vous présente l'animal.



Lorsque vous exécutez le programme, vous voyez apparaître cette fenêtre. Il ne vous reste plus qu'à vous familiariser avec son interface pour le moins austère.

#### 16.5.1 Les fichiers du TESAME

TESAME manipule deux sortes de fichiers. Les fichiers esp, c'est normal puisqu'on les modifie, et les fichiers esd.

Les fichiers ESD (Elder Scroll Data) sont des morceaux de fichier esm. TESAME les utilise pour son système d'édition avancée.

#### 16.5.2 Présentation générale

Le TESAME se divise en deux parties, dans la partie supérieure vous retrouvez une liste analogue à celle que vous pourriez voir dans les détails du TESC.

La partie inférieure vous montre les détails de chaque modification apportée par votre mod au monde de MW. Vous avez en fait une visualisation plus conviviale du contenu du fichier esp que ce que vous auriez avec un éditeur hexa.

Il est possible de choisir le classement des informations dans les deux tableaux en cliquant sur les en têtes de colonnes.

#### 16.5.3 Utilisation

L'utilisation du TESAME doit, malgré sa simplicité, se faire avec précaution. En effet, les modifications que vous faites à ce niveau sont définitives pour votre mod, et risquent de perturber profondément le fonctionnement de votre plug si vous ne faites pas attention. L'auteur de l'outil n'a pas intégré de possibilité de revenir en arrière sur les modifications.

Tout se joue dans la partie supérieure de l'interface. Si vous trouvez une donnée qui vous intrigue, double cliquez dessus (clic gauche). Cela devrait modifier le contenu de la partie inférieure, vous y voyez maintenant des précisions sur l'élément sélectionné.

Si vous pensez que les modifications apportées n'ont pas lieu d'être, faites un clic droit dans la partie supérieure, sur la ligne de l'item à éliminer et appuyez sur la touche [SUPPR], ou utilisez le menu Item/Delete.

La ligne concernée devrait disparaître de la zone supérieure.

Il est possible de sélectionner plusieurs éléments.

Si vous avez un doute sur une modification, sélectionnez la, puis utilisez le menu Item/Save As. Il vous sera alors demandé de donner un nom pour un fichier esd.

Une fois que vous aurez sauvegardé la donnée qui vous intéresse, effacer la, sauvegardez le fichier esp par le menu Mods/save as... Puis tester votre mod ainsi modifié, si vous notez un dysfonctionnement de votre mod à la suite de la suppression que vous venez de faire... vous pourrez récupérer les informations et les réintégrer dans votre mod par le biais du menu Item/Insert. (Sélectionner le fichier esd créé plus tôt et cliquez sur ok).

#### 16.5.4 Merge – Fusion et comparaison

En plus de toutes ces fonctionnalités, TESAME vous permet de fusionner et comparer deux fichiers esp. C'est une fonctionnalité très utile, lorsque l'on est nombreux à travailler sur un même mod.

Cette fonction est accessible par le biais du menu Mods/Merge.

Pour remarque, la fonction analogue du TESCS, File/Combined Plugins est plus puissante, et probablement plus fiable.

#### 16.5.5 Edition de l'en tête

Pour finir, TESAME vous offre la possibilité de modifier les informations de l'en-tête d'un fichier esp, auteur et description du contenu. (Mods/EditHeader)

#### **16.6 Une façon d'éviter de modifier le script principal.**

De trop nombreux moddeurs pensent que le meilleur moyen de lancer leurs scripts globaux est de modifier le main MW.....

CE N'EST PAS UNE BONNE IDEE !!!!!!!!!!!!!!!!!!!!!!!

Cela est la cause d'incompatibilité avec d'autres mods. Le meilleur moyen d'activer vos scripts globaux est d'utiliser un artifice comme d'envoyer le PJ voir une personne, ou dans une pièce.

Lorsque le PJ arrivera dans la pièce ou parlera au PNJ voulu, vous lancerez alors vos scripts globaux pour la première fois. Ils seront ensuite relancer automatiquement pour chaque partie.

Pour un personnage, ajouter dans les Result de sa salutation par défaut l'exécution du script global (StartScript monScript), pour un objet, attaché un script à un objet qui votre script global lorsque le PJ se trouve dans la même pièce....

#### **16.7 Les PNJ et vos mods**

Essayez autant que faire se peut de ne pas utiliser dans vos quêtes des PNJ du jeu initial. Il y a toujours un joueur qui l'aura tué et qui du coup ne pourra pas utiliser votre mod.

Même si cela est coûteux créer vos propres PNJ et inventez leur une histoire...

#### **16.8 Nettoyer une sauvegarde faite avec un mod sale.**

Le problème apparaît régulièrement, j'ai ajouté un objet dans le monde, et maintenant je n'arrive plus à le faire disparaître. Pourquoi ?

La réponse est simple tiens en une phrase : vous avez sauvegardé une partie en prenant en compte un mod qui n'était pas propre. Lorsque l'on sauvegarde une partie, on sauvegarde toutes informations relatives à l'objet avec lequel on est entré en contact. Ainsi on sauvegarde la position d'un PNJ de test et après on le trouve toujours au même endroit quelques soient les modifications que l'on peut faire.

Pour nettoyer une sauvegarde, on peut tenter de modifier son extension en esp, la charger dans TESAME et effacer les références qui nous dérangent. Après quoi on enregistre et on renomme le fichier en ess.

Cette manipulation devrait vous permettre de ne pas prendre en compte un mod sale.

## 17 Format des fichiers ESP de Morrowind

Cette section a été construite grâce aux informations extraites d'un document construit par Jim Adam.

### 17.1 Présentation générale

D'une manière tout à fait basique un fichier esp est une liste d'objets. Chacun de ces objets représente une information qui a été modifiée par rapport au fichier maître auquel il fait référence.

En ouvrant un fichier esp dans un éditeur hexadécimal, vous remarquerez que la majeure partie du texte lisible est constituée d'objets et d'informations relatives à ces objets (par exemple une porte est un objet, alors que son nom est une information).

Les premiers bytes (362) constituent l'en-tête du fichier (le header), c'est vrai pour la majorité des fichiers esp, pourtant cela peut varier en fonction du fichier maître (un autre que morrowind.esm demanderait un header différent).

Cet en-tête contient des données comme le nom de l'auteur, une description du mod, le nom du fichier esm, etc...

Puis à la suite de l'en-tête du fichier, suit la description des objets. Chaque objet commence par son type et sa longueur. Puis viennent les informations relatives à cet objet.

Le type est un mot de 4 lettres (DOOR, LIGH, STAT...). La longueur du champ d'un objet est de 12 bytes alors que chaque information n'occupe que 4 bytes.

La taille, comme pour toutes les informations binaires d'un esp est définie sur une base 256:  $A*256^0 + B*256^1 + C*256^2...$

### 17.2 Les types d'objets

Mot (TAG)	Description
ACTI	Activateur
ALCH	Alchimie
APPA	Appareil
ARMO	Armure
BOOK	Livre
BODY	Partie du corps (une hypothèse)
CELL	Cellule
CLOT	Les habits
CONT	Conteneurs
CREA	Créature
DOOR	Porte
DIAL	Sujet de conversation pour les dialogues
ENCH	Enchantement
FACT	Faction
FRMR	Référence à un objet (soit un nouvel objet, soit la copie d'un objet)
GLOB	Une variable globale
INFO	Info/Réponse de dialogue
INGR	Ingrédients
LAND	Modification dans l'altitude d'un paysage
LEVC	Créatures de niveau (hypothèse faite)
LEVI	Item de niveau
LIGH	Lumière
LTEX	Modification de la texture d'un paysage
MGEF	Effet magique
MISC	Objet divers
NPC_	Les pnj, NPC
PGRD	Grille de déplacement
PROB	Sonde de désamorçage
REPA	Objet de réparation
SCPT	Script

SKIL	Compétence
SNDG	Définition du type de bruit qu'une créature/un objet doit produire
SOUN	Un nouveau type de son. Ainsi, SNDG fait référence à un nouveau son. Celui-ci se réfère à SOUN qui réfère à un WAV.
SPEL	Les sorts et quelques types d'informations.
STAT	Les statiques
WEAP	Une arme
<b>Field Names</b>	
BNAM	le visage d'un pnj
CNAM	La classe d'un pnj (sa profession)
FNAM	Son identité (ID)
ITEX	Image d'un objet pour l'inventaire
KNAM	La coiffe d'un pnj
MODL	Le modèle d'un pnj
NAME	Le nom
NAM0	Apparait avant une liste de références pour une cellule. (pas de réponse claire)
NPC0	Contenu d'un conteneur
RNAM	Race d'un pnj
SCHD	En tête d'un script ( nom)
SCDT	Script compilé (hypothèse)
SCTX	Code du script

### 17.3 Détails

Les fichiers ESP, ont, apparemment un ordre préférentiel qui reste toujours le même. Si un fichier esp n'a pas de données à ajouter dans une section spécifique, alors la place est laissée libre. D'autres sont ignorées si aucune modification ou addition n'a été faite.

L'ordre de stockage d'information est le suivant:

Les scripts

Les nouvelles portes

Les nouveaux activateurs

Les nouveaux PNJ

Les nouveaux conteneurs

Les cellules

Les donnée de référence des créatures et pnj

Les données de référence des objets dans leur ordre de création.

Champs	Type	Taille	Description
<b>Header (En tête)</b>			Attention: faites attention; la taille de l'en-tête du fichier peut varier. Il contient en effet le nom du fichier Maître dont le plug dépend. Comme ce n'est pas toujours morrowind.esm.
File Signature Signature du fichier	String	4	C'est toujours "TES3" pour indiquer qu'il s'agit d'un fichier pour The Elder Scroll 3.
Header Size La taille de l'en-tête	Integer	4	Donne la taille de l'en-tête. Habituellement, 0x5A 0x01 0x00 0x00.
Reserved Réservé	Byte	8	Ces bytes doivent toujours être NULL
Begin Header On commence le header	String	4	"HEDR"
Header Size	Binary	4	Indique la distance entre la fin de cette information et le debut de l'instruction MAST. Habituellemn c'est 0x2C 0x01 0x00 0x00.
Version Number	Float	4	Le numéro de version du fichier (1.2) sous forme d'un réel (un nombre avec une virgule): 0x9A 0x99 0x99 0x3F

Unknown (inconnu)	Binary	4	Quatre valeur nulle, But INCONNU: 0x00 0x00 0x00 0x00
Nom de l'auteur	String	32	Les caractères non utilisés doivent être NULL.
Description	String	260	Description du mod par l'utilisateur. Les caractères non utilisés doivent être NULL..
Début des infos du fichier maître	String	4	"MAST"
Longueur du nom du fichier maître	Binary	4	Indique la taille(en bytes) du nom du fichier maître. Ceci incluse également le NULL (obligatoire à la fin). Si vous utilisez "Morrowind.esm" comme fichier maître, la valeur est 0x0E.
Master Filename	String	Varies	Il s'agit du nom du fichier maître. Il est terminé par le caractère NULL. Habituellement "Morrowind.esm", il peut changer si vous utiliser un autre esm.
Debut des données	String	4	"DATA"
La taille des données	Integer	4	0x08 0x00 0x00 0x00
Taille en byte du master	Integer	8	Taille en byte du fichier maître: 0x1C 0xF9 0xC1 0x04 0x00 0x00 0x00 0x00
<b>Script Data</b>			La section des scripts du plug
Début d'un script	String	4	"SCPT"
Taille du script	Integer	12	Taille du script de puis le mot "SCHD" jusqu'à la fin.
Begin SCHD	String	4	"SCHD"
SCHD Length	Integer	4	Apparemment cette valeur vaut toujours 0x34, et les espaces manquant sont mis à 0.
SCHD données	String	Varies	Apparemment, contient le nom du script, et d'autres données non null à partir d'un certain offset.
Début de la donnée du script SCDT	String	4	"SCDT"
Taille du texte du script	Integer	4	Taille de l'information de la taille du source du texte.
Données SCDT	Inconnu	Varies	Apparemment un jeu de 4 entiers suivi de texte.
[Incomplet]			A completer. Il y a encore de nombreuses informations nécessaires pour les scripts.
<b>Les données des pnj</b>	?	?	A completer.
<b>Les données des creatures</b>	?	?	A completer
<b>Activateur</b>			Les données relatives aux activateurs ajoutées aux données des objets
Début d'un activateur	String	4	"ACTI"
Taille de l'activateur	Integer	12	Taille de l'information relative à cet activateur.
Debut de l'information sur le nom	String	4	"NAME"
Taille de l'information	Integer	4	Varie.
Nom	String	Variable	C'est l'ID unique de l'activateur. (par exemple, "active_com_bed_01".
Modèle de l'activateur	String	4	MODL
Taille de l'information modèle	Integer	4	C'est le nom du fichier NIF utilisé pour cet activateur (Ex., "fFurn_Com_Bed_01.nif").
Début du nom utilisé dans le jeu	String	4	"FNAM"
Taille de l'information	Integer	4	Variable. C'est la taille de l'information de nom. Elle incluse le byte NULL de la fin.
Le nom	String	4	Le nom utilisé dans le jeu par ex, "Lit".
Debut du script associé	String	4	"SCRI"
Taille du nom du script associé	Integer	4	Variable. C'est la taille du nom du script associé à l'objet. Le byte NULL de la fin est inclus dans le calcul.

Les données du SCRI	String	4	C'est le nom du script qui s'achève par le byte NULL. Ex., "Lit_Classique".
<b>Container</b>			Pour les informations relatives à cet Référence, reporter vous au début de ce chapitre. Pour créer un nouveau conteneur... (lorsqu'un nouveau conteneur est créé cette section n'est pas vide...)
Début d'un conteneur	String	4	"CONT"
Taille totale de l'information	Integer	12	Variable d'un conteneur à l'autre. C'est la taille du reste des informations.
Début des informations de noms	String	4	"NAME"
Taille du nom	Integer	4	Variable.
Le nom	String	Variable	C'est l'ID unique du conteneur.
Début MODL	String	4	"MODL"
Taille de MODL	Integer	4	Variable.
Les données de MODL	String	Variable	C'est le nom du fichier NIF qui contient le modèle utilisé dans le jeu, par ex: "o\Contain_Com_Chest_01.nif"
Début du nom usuel FNAM	String	4	"FNAM"
Taille du nom usuel	Integer	4	Variable
FNAM Data	String	Variable	Le nom usuel du conteneur ( par ex: 'Coffre').
Début des informations CNDT	String	4	"CNDT"
Taille de CNDT	Integer	4	0x04 0x00 0x00 0x00
Données de CNDT	Unknown	4	Par défaut: 0x00 0x00 0x00 0x00
FLAG	String	4	"FLAG"
Taille des FLAG	Integer	4	0x04 0x00 0x00 0x00
Données du coffre FLAG Data	Integer	4	Par défaut "0x08".
Début des info du propriétaire NPC0	String	4	"NPC0" – (C'est un zero pas un o majuscule). C'est une information qui se répète, une pour chaque type d'objet contenu dans le conteneur.
Taille du NPC0	Integer	4	Peut varier mais on l'a toujours vu égal à 0x24, et les espaces inutilisés sont à la fin de l'information remplis de NULLs.
Quantité des objets	Integer	4	Variable. Indique la quantité de l'objet présent dans le conteneur.
ID de l'objet	String	Variable	Variable. C'est l'ID de l'objet dans le conteneur.
<b>Les sons</b>			
Début de l'information relative au son	String	4	"SOUN"
Taille de l'info	Integer	12	Variable. C'est la taille totale de l'information du son qui commence au champ suivant (NAME).
Début du Nom	String	4	"NAME"
Taille de l'info	Integer	4	Taille du nom, inclus le byte NULL de fin.
Données NAME	String	Variable	C'est l'ID unique du son. Par ex. "Stone_Door_Open_01"
Début du nom du fichier	String	4	"FNAM"
Taille de FNAM	Integer	4	Variable. C'est la taille du nom du fichier, qui inclue le byte NULL de fin.
Données de FNAM	String	4	C'est le nom du fichier wav à utiliser pour ce son. Ex., "Fx\Stone01.wav".
Préfixe de la donnée	String	4	"DATA"
Taille de la donnée.	Integer	4	Longueur en byte de l'information du son. Apparemment toujours 0x03.

Les données du son	Set of Values	3	Pas certain de la signification de ces informations, apparemment le plus courant est: 0xFF 0x00 0x00
Données pour les cellules d'extérieur	?	?	A finir.
Données pour les cellules d'intérieur			
Début la Cellule	String	4	"CELL"
Taille de la cellule	Integer	12	Longueur en bytes de cette cellule.
Préfixe de la cellule	String	4	"NAME"
Taille du nom de la cellule	Integer	4	Longueur en byte du nom de la cellule, inclue le byte NULL de fin.
Nom de la cellule	String	Variable	Le nom de la cellule, inclue le byte NULL de fin.
Début des donnée	String	4	"DATA"
Taille de la donnée	Integer	4	Longueur en byte des informations de la cellule.
Données de la cellule	Jeu de valeur	12	<p>Les valeurs par défaut sont: 0x01 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x80 0x3F</p> <p>Les 4 premier bytes sont un masque qui informe de la presence d'eau dans la cellule et si il est illegal d'y dormir.</p> <p>(a) si il y a de l'eau, la première valeur devient 0x03. ( bascule sur le byte 0x02.) Contrôle la hauteur de l'eau dans la section INTC (la prochaine).</p> <p>(b) si il est illégal de dormir ici, le premier nombre doit être 0x05. (par opposition au byte 0x04).</p> <p>(c) si il y a de l'eau et qu'il est illegal dormir dans la cellule, le byte deviant 0x07 (le jeu des trios premiers bytes).</p> <p>Les quatres suivants indique la densité du brouillard dans la cellule par un réel variant de 0.0f à 1.0f.. NB : cela copie l'information donnée dans la donnée AMBI (dans la section INTV).</p>
Debut de INTV	String	4	"INTV" – la signification de INTV n'est pas claire, peut pour différencier les cellules internes des cellules externs ?
Taille de l'info INTV	Integer	4	Taille de l'info INTV, par défaut: 0x04 0x00 0x00 0x00.
Les données de INTV	Int	4	<p>Par défaut: 0x00 0x00 0x00 0x00</p> <p>Hauteur de l'eau dans la cellule (si il y a de l'eau dans la cellule, voir au dessus). Par exemple pour 1m50 d'eau (5 pieds) il faut écrire: 0x05 0x00 0x00 0x00. [Je ne sais pas où est la reference pour cette hauteur ( peut être z=0)?]</p>
Début de AMBI	String	4	"AMBI" – contient les données de la lumière ambiante dans la cellule.
Taille de l'info AMBI	Integer	4	Taille du champ AMBI field data. Par défaut : 0x10 0x00 0x00 0x00.
Donnée de l'éclairage ambiant AMBI	Jeu de entiers	16	<p>Les valeurs par défaut sont: 0x47 0x47 0x47 0x00 0xF2 0xD9 0xD9 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x80 0x3F</p> <p>Les quatres premières valeur représentent le code RGB de la lumière ambiante bien que les valeurs par défaut (quatre fois 0x47) ne correspondent pas au valeur proposes par défaut dans l'éditeur (0x47 = 66) et l'éditeur indique 71.... Toutes les modifications font coller parfaitement pour le reste. Les quatres valeurs pour l'éclairage ambiant sont toujours 0x00.</p> <p>Les 4 valeurs 0xF2 0xD9 0xD9 0x00 correspondent à la lumière solaire dans la cellule, avec la quatrième toujours égale à zero (0x00).</p> <p>Les 4 valeurs suivantes correspondent à la couleur RGB du brouillard dans la cellule (la 4<sup>ème</sup> est toujours égale à 0x00).</p> <p>Les 4 bytes sont un reel qui exprime la densité du brouillard variant de 0.0f à 1.0f (information dupliquée de celle données dans la section des données de cellule).</p>

Liste des habitants Créatures et PNJ			Toutes les créatures et les PNJ qui apparaissent dans la cellule sont listés ensuite. Avant la liste des objets. Si vous ne référencez qu'une créature déjà existante ou un PNJ existant (plutôt que de créer un nouveau perso) regardez d'abord la partie Référence à un Objet.
Début du nom NAM0	String	4	"NAM0" – (c'est un zéro à la fin) Apparemment c'est le même que celui qui précède les types d'objets (FRMRs) présents dans une cellule. Il n'apparaît qu'une fois, quelque soit le nombre d'objet dans la cellule.
Taille de NAM0	Integer	4	Par défaut 0x04 0x00 0x00 0x00.
NAM0 : les données	Integer	4	Avoir!! Pour l'instant semble indiquer le nombre d'enregistrement FRMR dans la liste, mais ce n'est pas toujours le cas....
<b>Référence à un Objet</b>			A chaque référence que vous faites, quelque soit le type de l'objet (static, conteneur, creature, activator...), vous pouvez utiliser le même format. Seulement si vous créer un nouveau type de créature ou si vous faites une copie complète de quelque chose, alors il sera nécessaire de rentrer dans les détails.
Début de FRMR	String	4	"FRMR" – On ne sait pas ce que c'est.
Taille de cette info	Integer	4	0x04 0x00 0x00 0x00
Données de FRMR	Index	4	C'est l'index de cet objet dans la cellule. Le premier ajouté à l'index 0x01. Le suivant prend 0x02, et ainsi de suite.
Début du Nom	String	4	"NAME"
Taille de l'info	Integer	4	Taille du nom, inclue Le caractère terminal NULL
La donnée de NAME	String	Variable	Le nom de l'objet référencé, par exemple:"in_v_s_corner01"
Echelle [Optionnel]	String	4	"XSCL" – Ce champ n'apparaît que si le Scale de l'objet est différent de 1.0.
Taille de l'info	Integer	4	0x04 0x00 0x00 0x00
Valeur de l'échelle	float	4	L'agrandissement (Scale) de l'objet si il est différent de 1.0.
Début des données	String	4	"DATA"
Taille des données	Integer	4	0x18 0x00 0x00 0x00
Les données	Jeu de valeur	24	Les 3 premières valeurs sont les coordonnées X, Y, Z de l'objet, sous la forme de 3 réels (4 bytes chacun). Les 3 suivantes sont les valeurs des rotations autour de X,Y et Z , données sous la forme de 3 réels (4 bytes chacun). Les valeurs sont données ici en radians même si l'éditeur les montre en degrés.
<b>Porte de téléportation</b>			Pour les portes qui ne téléporte pas, regardez la section Référence à un objet. Les portes de téléportation apparaissent dans la section FRMR des enregistrements à l'intérieur d'une cellule et pas au début du fichier dans une section différente, comme c'est le cas pour la plupart des définitions d'objets.
Début de FRMR	String	4	"FRMR" – ne sait pas ce que c'est.
Taille du champ	Integer	4	0x04 0x00 0x00 0x00.
Données du champ FRMR	Index	4	C'est l'index de l'objet dans la cellule.
Début du nom	String	4	"NAME"
Taille du champ	Integer	4	Longueur du champ nom. Inclue le caractère NULL de fin.
Le nom	String	Variable	Le nom (ID) de la porte, par exemple : "door_cavern_doors10".
Début de DODT	String	4	"DODT" – Destination de la téléportation.
Taille du champ	Integer	4	Taille de la section DODT 0x18 0x00 0x00 0x00.
Les données	Jeu de valeur	24	La localisation de l'endroit exact où la téléportation amène. C'est à dire les coordonnées X,Y,Z puis, les angles de rotation X,Y,Z du marqueur de voyage qui se trouve à l'arrivée du téléporteur.
Début de l'information Nom [Optionel]	String	4	"DNAM" – Utilisé uniquement quand la téléportation mène à une cellule d'intérieur.
Taille du champ	Integer	4	Taille de l'information nom, inclue le caractère NULL de fin.

La donnée DNAM	String	Variable	Le nom de la cellule d'intérieur dans laquelle la porte nous amène, par ex : "Balmora, Guild of Mages"
Début des informations	String	4	"DATA"
Taille de l'info	Integer	4	Taille de l'info: 0x18 0x00 0x00 0x00.
Données	Jeu de valeur	24	Ce sont les coordonnées X,Y,Z et les angles de rotation X,Y,Z de la porte elle-même. Un réel par valeur (4 bytes).

---

**Créateurs Originaux:**

Erik Benerdal (aka Scarabus of [Fudge Entertainment](#)) DizzyJim Adam

Vous pouvez le contacter pour ajouter vos informations: email [Jim Adam](#).