

Importation de contenu



Étude des moteurs de jeu

Plan de la présentation

- Exportation
- Textures
- Matériaux
- Collisions
- LODs



NAND168 : Étude des moteurs de jeu

2

Éléments de contenu

- Chaque élément de contenu est enregistré dans un fichier .uasset
 - Plus simple pour les outils comme Perforce
 - Temps de synchronisation plus rapide lors de changement
- Chaque élément est lié à un chemin précis
 - Il n'est pas recommandé de déplacer un objet à l'extérieur du *Content Browser*



NAND168 : Étude des moteurs de jeu

3

Éléments de contenu

- Utiliser une bonne structure de répertoires pour s'y retrouver facilement
- Utiliser des noms d'objets **significatifs**



NAND168 : Étude des moteurs de jeu

4

Unité et échelle

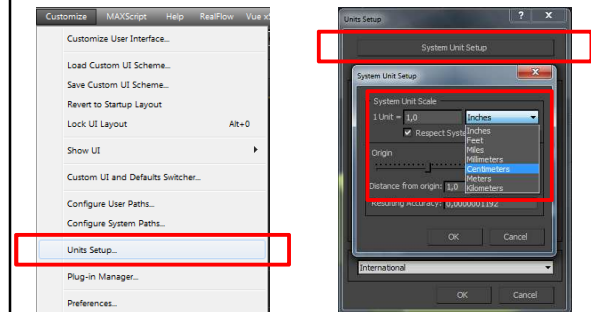
- Unreal utilise le système métrique (cm)
- Préférable de configurer 3DS pour avoir le même système d'unité



NAND168 : Étude des moteurs de jeu

5

Unité et échelle



NAND168 : Étude des moteurs de jeu

6

Unité et échelle

- À titre de référence, un personnage de taille moyenne mesure environ 192 unités dans Unreal



NAND168 : Étude des moteurs de jeu

7

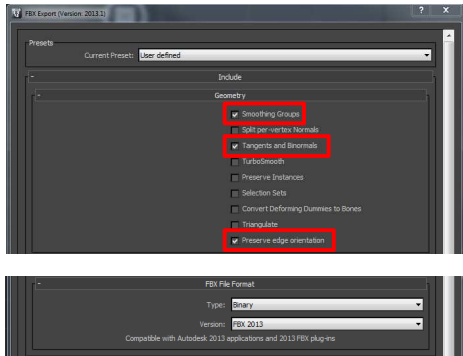
Exportation d'un objet simple

- Porter attention :
 - Point de pivot dans Unreal : origine dans 3DS
 - Grille
 - Triangles
- Export selected / export all



NAND168 : Étude des moteurs de jeu

8



NAND168 : Étude des moteurs de jeu

9

Exportation d'un objet simple

- Exercice – Exportation / importation (10 minutes)
 - Créer une chaise dans 3DS Max
 - 80 unités de haut (approx.)
 - 30 unités de large (approx.)
 - Exporter vers le format FBX
 - Importer dans l'engin



NAND168 : Étude des moteurs de jeu

10

Structure des répertoires

- Important d'avoir une structure de répertoire où l'on se retrouve facilement
- Il n'y a pas UNE bonne structure, il faut cependant que tous les membres de l'équipe utilise la même



NAND168 : Étude des moteurs de jeu

11

Structure des répertoires

- Vous devrez utiliser celle-ci :

Content\AI	all AI specific content (blackboard – controller – pawn – tasks)
Content\Base	base assets (eg. master materials) material functions and other "foundations" assets
Content\Characters	character meshes / blueprints and skeletons
Content\Experimental	development assets / mockup meshes / special textures and icons. Not part of final build
Content\Effects	particle effects and dependencies
Content\Environment	environment assets (meshes – materials – textures)
Content\Gameplay	gameplay specific assets (eg. flag mesh & dependencies for Capture The Flag)
Content\Sounds	sounds and sound cues
Content\UI	menu and HUD assets
Content\Weapons	weapons and projectiles

<http://www.tomlooman.com/ue4-naming-convention/>


NAND168 : Étude des moteurs de jeu

12

Structure des répertoires

- Tous les fichiers de niveaux (.umap)
 - Content\Maps



NAND168 : Étude des moteurs de jeu

13

Textures

- Toujours prendre des textures carrés ayant une puissance de 2
 - $2^0 \times 2^0$ jusqu'à $2^{12} \times 2^{12}$ pixels
 - 1×1 jusqu'à 4096×4096 pixels
- Les textures ne sont pas appliquées directement sur un objet, elles sont plutôt utilisées par un matériel



NAND168 : Étude des moteurs de jeu

14

Textures

Resolution	Total Mips from 1x1	DXT1	DXT5
16x16	5 mips	312 bytes	496 bytes
32x32	6 mips	824 bytes	1.48kb (1.520 bytes)
64x64	7 mips	2.80kb (2.872 bytes)	5.48kb (5.616 bytes)
128x128	8 mips	10.8kb (11.064 bytes)	21.4kb (22.000 bytes)
256x256	9 mips	42.8kb (43.832 bytes)	85.4kb (87.536 bytes)
512x512	10 mips	170kb (174.904 bytes)	341kb (349.680 bytes)
1024x1024	11 mips	682kb (699.192 bytes)	1.33MB (1.398.256 bytes)
2048x2048	12 mips	2.66MB (2.796.344 bytes)	5.33MB (5.592.560 bytes)
4096x4096	13 mips	10.6MB (11.184.962 bytes)	21.3MB (22.369.776 bytes)
8192x8192	14 mips	42.6MB (44.739.384 bytes)	85.3MB (89.478.640 bytes)



NAND168 : Étude des moteurs de jeu

15

Textures

- Les textures peuvent être utilisées pour être rendues (ex. diffuse) ou pour communiquer de l'information (ex. normale)
 - Le paramètre sRGB indique si le gamma doit être ajusté ou non. On ne veut pas que le gamma soit ajusté sur une normale par exemple.



NAND168 : Étude des moteurs de jeu

16

Textures

- Formats supportés
 - .bmp, .float, .pcx, .png, .psd, .tga, .jpg
 - .dds - Cubemap Texture (32bits/Channel, 8.8.8.8 ARGB 32 bpp, unsigned)
 - .hdr - Cubemap Texture (LongLat unwrap)
- RGB / RGBa
- Utiliser seulement 8 bits / canal
 - **À vérifier**

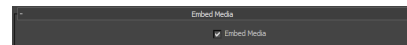


NAND168 : Étude des moteurs de jeu

17

Textures

- Les textures peuvent être exportées automatiquement
 - Coché l'option « Embed Media »



NAND168 : Étude des moteurs de jeu

18

Textures

- Le nom des textures dans 3DS peut être transféré automatiquement
- Utiliser la nomenclature suivante:
(Préfixe_)NomTexture[_Nombre][_Suffixe]

Exemple :

T_Roche_D



NAND168 : Étude des moteurs de jeu

19

Textures

- Suffixes :

_BC	Base color
_MT	Metallic
_S	Specular
_R	Roughness
_N	Normal
_DP	Displacement
_AO	Ambient Occlusion
_H	Height Map
_FM	Flow Map
_L	Light Map (fake)
_M	Mask



NAND168 : Étude des moteurs de jeu

20

Textures

- Masque pour environnement :

R = Metallic
G = Roughness
B = Ambient Occlusion

- Masque pour personnage :

R = Metallic
G = Roughness
B = Subsurface Opacity

- Masque pour personnage (cheveux):

R = Hair Alpha
G = Specular/Roughness map
B = Anisotropic direction map



NAND168 : Étude des moteurs de jeu

21

Textures

- Dans le répertoire, utiliser le groupe « Textures »
 - Exemple pour une texture destinée à un élément d'environnement
 - Content\Environment\Textures
 - Exemple pour une texture destinée à un élément de *gameplay*
 - Content\Gameplay\Textures



NAND168 : Étude des moteurs de jeu

22

Matériaux

- Les matériaux de base sont transférés avec FBX
 - Standard
 - Multi/Sub-object
- Le noms des matériaux créés dans Unreal seront les mêmes que ceux dans 3DS Max



NAND168 : Étude des moteurs de jeu

23

Matériaux

- Seulement certaines textures sont transférées :

Map / textures	Configuration 3DS MAX
Diffuse	Diffuse > Bitmap
Normal Map	Bump > Normal Bump > Normal > Bitmap



NAND168 : Étude des moteurs de jeu

24

Matériaux

- Le nom des matériaux dans 3DS peut être transféré automatiquement
- Utiliser la nomenclature suivante:
 - M_NomMatériel
- Dans le répertoire, utiliser le groupe « Materials »



NAND168 : Étude des moteurs de jeu

25

Matériaux

- Exercice – matériel (10 minutes)
 - Choisir une texture de bois et créer un matériel pour la chaise
 - Exporter vers le format FBX
 - Importer dans l'engin

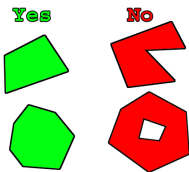


NAND168 : Étude des moteurs de jeu

26

Collisions

- Peuvent être créés dans 3DS ou Unreal
 - Unreal = seulement des collisions **très** simples
- Les collisions doivent toujours être convexes



NAND168 : Étude des moteurs de jeu

27

Collisions

- Que doit-on considérer lors de la création de collision?

<http://www.allreadable.com/fb6aAVVK>



NAND168 : Étude des moteurs de jeu

28

Collisions

- Fonction de l'objet
- Échelle de l'objet
- Minimisation du nombre de polygone
- Interaction avec l'objet?



NAND168 : Étude des moteurs de jeu

29

Collisions

- 3DS : Les collisions sont associées à un objet en fonction de la nomenclature
 - Objet : SM_IlotA
 - Collision : UCX_SM_IlotA
- Il peut y avoir plusieurs collisions associées à un même objet
 - Collision 1 : UCX_NomObjet
 - Collision 2 : UCX_NomObjet



NAND168 : Étude des moteurs de jeu

30

Collisions

- Comment feriez-vous les collisions?



NAND168 : Étude des moteurs de jeu

31

Collisions

- Exercice – créer collision (10 minutes)
 - Créer la collision pour la chaise en tenant compte qu'on peut s'y asseoir
 - Exporter vers le format FBX
 - Importer dans l'engin



NAND168 : Étude des moteurs de jeu

32

Niveaux de détails (LOD)

- On désire généralement limité le coût d'affichage lorsqu'un objet s'éloigne de la caméra
 - Solution : LOD
- On associe à un même objet plusieurs niveau où chacun a un nombre moins élevé de triangles que le précédent



NAND168 : Étude des moteurs de jeu

33

Niveaux de détails (LOD)

- Chaque LOD peut avoir ses propres textures et matériaux et leur nombre peut varier
- On associe à un même objet plusieurs niveau où chacun a un nombre moins élevé de triangles que le précédent



NAND168 : Étude des moteurs de jeu

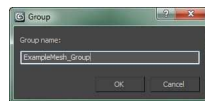
34

LOD - Procédure

- Choisir tous les objets et choisir *Group*



- Choisir un nom significatif



NAND168 : Étude des moteurs de jeu

35

LOD - Procédure

- Cliquer  et choisir *Level of Detail*
 - Utiliser  au besoin
- En ayant le groupe sélectionné, cliquer sur 
 - Les objets seront automatiquement triés en fonction de leur complexité



NAND168 : Étude des moteurs de jeu

36

LOD - Procédure

- Sélectionner le groupe (et les collisions le cas échéant) et exporter en utilisant les mêmes étapes qu'un *Static mesh*

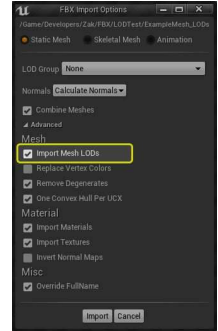


NAND168 : Étude des moteurs de jeu

37

LOD - Procédure

- Lors de l'importation dans Unreal, assurez-vous de cocher l'option *Import Mesh LODs*



NAND168 : Étude des moteurs de jeu

38

LOD - Procédure

- Exercice – créer LOD (10 minutes)
 - Créer un groupe de 4 mesh avec différents niveaux de détail
 - Exporter vers le format FBX
 - Importer dans l'engin
 - Inclure dans une scène et ajuster la distance où l'objet change de niveau de détails



NAND168 : Étude des moteurs de jeu

39

Convention des noms

- Utiliser la nomenclature suivante:
 - (Préfixe_)NomItem

BP_	Blueprint
SK_	Skeletal Mesh
SM_	Static Mesh
AD_	Apex Destructible Asset
AC_	Apex Cloth Asset
MT_	Morph Target
ST_	Speed Tree
PS_	Particle System
LF_	Lens Flare
VF_	Vector Field
S_	Sound
SC_	Sound Cue

M_	Material
MI_	Material Instance
MITV_	Material Instance Time Varying
MF_	Material Function
MPC_	Material Parameter Collection
T_	Texture
TC_	Texture Cube
RT_	Render Target
PM_	Physical Material



NAND168 : Étude des moteurs de jeu

40