Originally posted by java23
A la demande de Thomas, je vous donne quelques explications sur cette technique (ça devrait aussi intéresser quelques autres) .

L.O.D. = Level Of Detail ou niveaux de détails en Français .

_____

Avant le LOD dans TrainZ, les "meshes" (modèles) étaient de la forme :
-- .im (indexed mesh) : c'est à dire un modèle immuable faisant toujours le même nombre de polys . S'applique à tous les modèles sauf le matériel roulant (scenery, trackside, track, road, bridge, tunnel, vehicle, etc .)
-- .pm (progressive mesh) : c'est à dire un modèle progressif dont le jeu réduit autant que faire se peut le nombre de polys au fur et à mesure de l'éloignement . Ne s'applique qu'au matériel roulant et n'agit que sur le nombre de polys, pas sur la taille et le nombre des textures .

Dans le répertoire de l'objet chose, il y a un fichier chose.im ou chose.pm, créé dans GMax, que le jeu utilise pour matérialiser l'objet .

_____

Maintenant, avec le LOD dans TrainZ, les meshes sont de la forme .lm (level of detail mesh), c'est à dire un modèle constitué de plusieurs versions, chacune ayant differents niveaux de polygonage et de texturage .
Chaque version est un fichier .im que le jeu va chercher, selon la distance, dans un fichier chose.lm.txt placé dans le répertoire de l'objet chose .

Je prends un exemple :

dans le répertoire de l'objet chose, il y a le config.txt qui spécifie, entre autres :
PHP:

```
...
mesh-table
{
  default
    {
      mesh chose.lm
      auto-create 1
    }
}
...
```

Ce petit paragraphe indique au jeu de chercher les modèle de l'objet dans le fichier chose.lm.txt, placé dans le répertoire chose .

Le fichier chose.lm.txt se présente sous la forme :
PHP:

```
version 1.0


  offset = 0.01;

  calcPoint = center;

  multiplier = 1.0;

  animationCutOff = 0.00;

  renderCutOff = 0.00;

  attachmentCutOff = 0.00;


  mesh("0.05")

  {

    name="chose_minuscule.im";

  }


  mesh("0.1")

  {

    name="chose_petit.im";

  }


  mesh("0.2")

  {

    name="chose_moyen.im";

  }


  mesh("1.0")
```

```
  {
    name="chose.im";
  }
```

Celui-ci veut dire en Français :

-- version 1

-- "flottement" entre deux modèles = 0,01 . Evite au jeu de faire "clignoter" l'objet entre deux modèles si on se trouve à la distance à laquelle le jeu est censé changer de modèle . Plus la valeur est petite, plus le flottement est petit .

-- le point de l'objet à partir duquel sont calculés les niveaux de rendus (center / near / far)

-- multiplicateur = 1 . Laisser cette valeur .

-- niveau de détail à partir duquel cesser l'animation des parties ... animées  **. Exprimé en taille par rapport à l'écran (1 plein écran, 0.5 moitié de l'écran, 0.00 ne jamais cesser l'animation .)**

-- niveau de détail à partir duquel l'objet n'est plus rendu (devient invisible)

-- niveau de détail à partir duquel les bogies ne sont plus rendus (deviennent invisibles)

Ensuite est énumérée la liste des modèles à rendre, du plus petit (le plus éloigné) au plus grand (le plus près) .

-- objet de taille = 5 centièmes de l'ecran, utiliser le modèle "chose_minuscule"

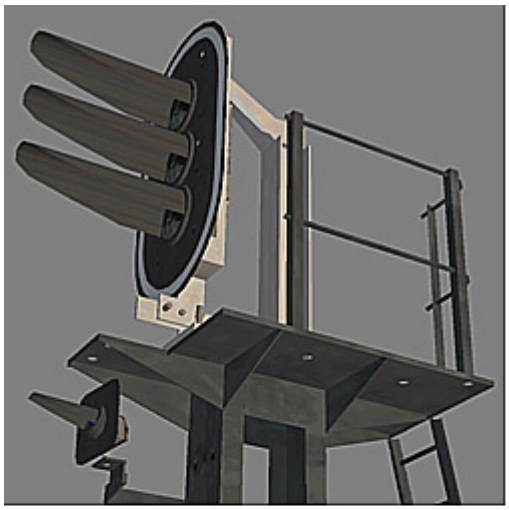
-- objet de taille = 1 dizième de l'ecran, utiliser le modèle "chose_petit"

-- objet de taille = 2 dizièmes de l'ecran, utiliser le modèle "chose_moyen"

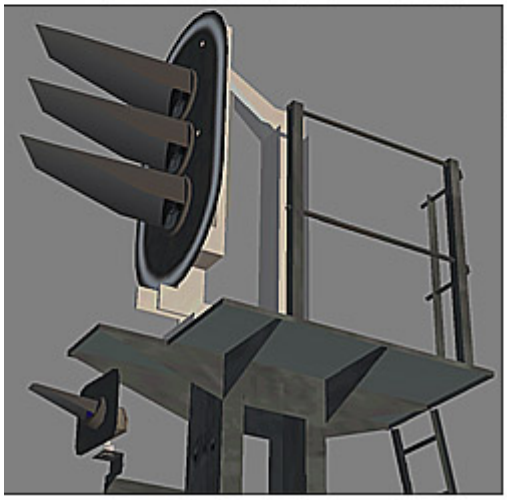
-- objet de taille = plein écran, utiliser le modèle "chose"

Vous pouvez donner n'importe quel noms à vos modèles . Il peut y en avoir 2 comme 50 suivant la progressivité avec laquelle vous voulez que celui-ci évolue avec sa taille apparente sur l'écran . L'exemple ci-dessus utilise quatre modèles différents, construits chacun dans GMax, chacun avec ses textures, différentes ou communes entre plusieurs modèles . Tous les modèles chose.im / chose_x.im / chose_xx.im / chose_xxx.im / chose_xxxx.im / etc.im et leurs textures sont placés dans le répertoire chose .

A titre d'exemple, mon panneau A sur mât droit à hauteur normale, utilise quatre modèles ayant les caractéristiques suivantes :



modèle vu de 0 à 45 m de distance :
- 2358 polygones
- 5 textures de 256x256 pixels
- 4 textures de 64x64 pixels
144 ko de 3D
905 ko de textures



modèle vu de 45 à 93 m de distance :
- 792 polygones
- 1 texture de 256x128 pixels
- 2 textures de 64x64 pixels
56 ko de 3D
113 ko de textures

modèle vu de 93 à 180 m de distance :
- 293 polygones
- 3 textures de 64x64 pixels
27 ko de 3D
29 ko de textures

modèle vu de 180 m à l' infini:
- 157 polygones
- 1 texture de 32x64 pixels
15 ko de 3D
7 ko de textures

Si vous l'avez téléchargé, examinez le contenu du répertoire "kuid 23299 24047" .

Pour plus d'informations sur le L.O.D., télécharger ce fichier

Amicalement,

Philippe

Donc en bref le but est de construire son objet en autant de fichiers gmax qu'il y aura de changement dans la distance, à vous de voir pour l'esthétique par des essais.
C'est plus long à faire mais ça vaut le coup...
N'ayez pas peur de sucrer, de loin ya que la forme qui compte!

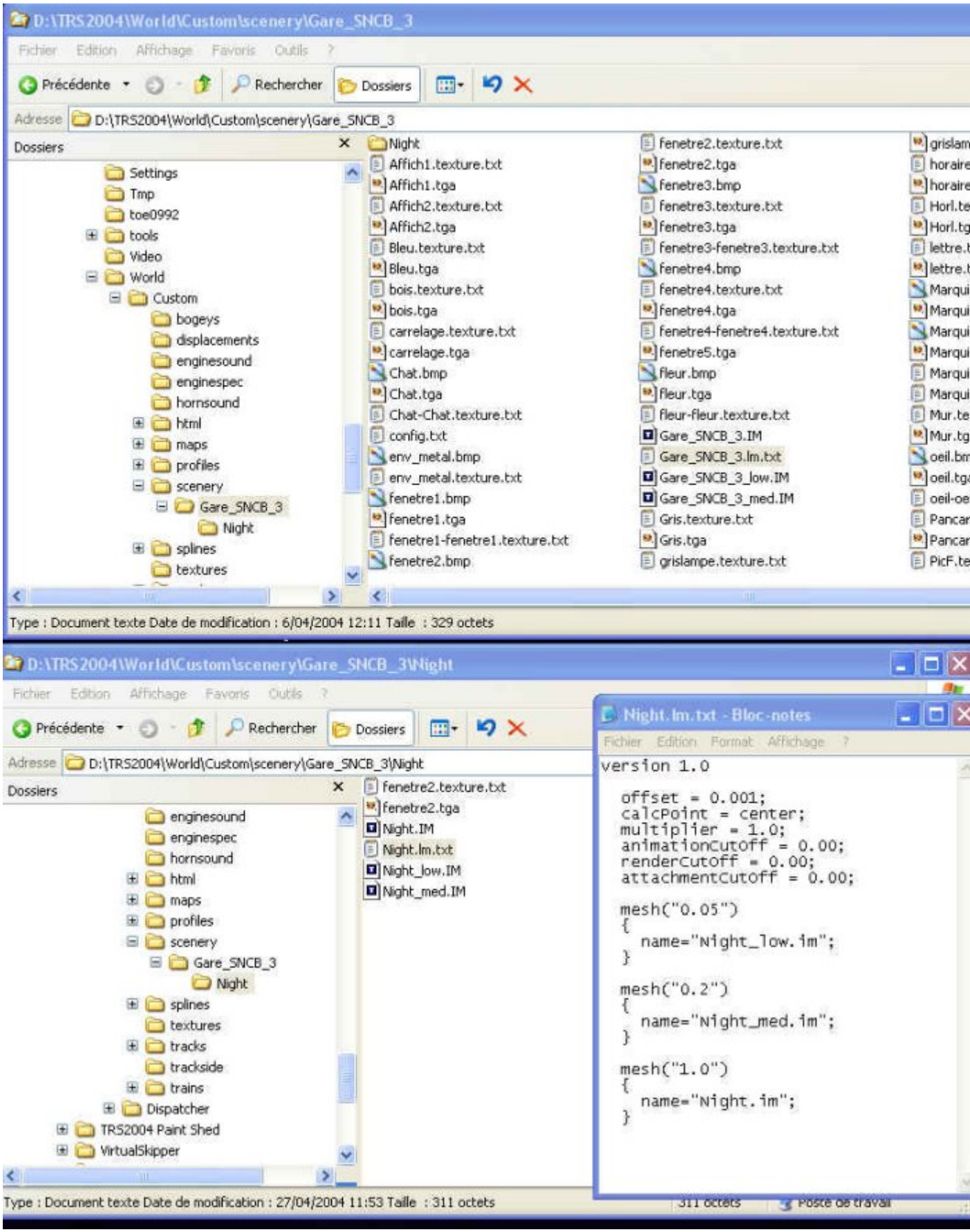
En export, tout va dans le même dossier.

Pour le mode nuit (je n'ai fait qu'un même gmax pour les 3 modes), cela fonctionne indépendamnent comme ci-dessus mais dans un sous-dossier.

Le config va diriger la lecture de l'objet sur un fichier lm (pas im mais LM) que vous aurez créé et qui va regrouper tous les im en indiquant à quelle distance ils seront lus (à vous de régler au mieux).

Le reste est indiqué dans mes screens

Au fait, ne faites pas comme moi pour les textures, il faut que je me décide à faire une map de texturage


D:\TRS2004\World\Custom\scenery\Gare_SNCB_3
Fichier Edition Affichage Favoris Outils ?
Précédente Rechercher Dossiers
Adresse D:\TRS2004\World\Custom\scenery\Gare_SNCB_3
Dossiers
Settings
Tmp
toe0992
tools
Video
World
Custom
bogeys
displacements
enginesound
enginespec
hornsound
html
maps
profiles
scenery
Gare_SNCB_3
Night
splines
textures
Night
Affich1.texture.txt
Affich1.tga
Affich2.texture.txt
Affich2.tga
Bleu.texture.txt
Bleu.tga
bois.texture.txt
bois.tga
carrelage.texture.txt
carrelage.tga
Chat.bmp
Chat.tga
Chat-Chat.texture.txt
config.txt
env_metal.bmp
env_metal.texture.txt
fenetre1.bmp
fenetre1.tga
fenetre1-fenetre1.texture.txt
fenetre2.bmp
fenetre2.texture.txt
fenetre2.tga
fenetre3.bmp
fenetre3.texture.txt
fenetre3.tga
fenetre3-fenetre3.texture.txt
fenetre4.bmp
fenetre4.texture.txt
fenetre4.tga
fenetre4-fenetre4.texture.txt
fenetre5.tga
fleur.bmp
fleur.tga
fleur-fleur.texture.txt
Gare_SNCB_3.IM
Gare_SNCB_3.lm.txt
Gare_SNCB_3_low.IM
Gare_SNCB_3_med.IM
Gris.texture.txt
Gris.tga
grislampe.texture.txt
Type : Document texte Date de modification : 6/04/2004 12:11 Taille : 329 octets
D:\TRS2004\World\Custom\scenery\Gare_SNCB_3\Night
Fichier Edition Affichage Favoris Outils ?
Précédente Rechercher Dossiers
Adresse D:\TRS2004\World\Custom\scenery\Gare_SNCB_3\Night
Dossiers
enginesound
enginespec
hornsound
html
maps
profiles
scenery
Gare_SNCB_3
Night
splines
textures
tracks
trackside
trains
Dispatcher
TRS2004 Paint Shed
VirtualSkipper
fenetre2.texture.txt
fenetre2.tga
Night.IM
Night.lm.txt
Night_low.IM
Night_med.IM
Night.lm.txt - Bloc-notes
Fichier Edition Format Affichage ?
version 1.0
offset = 0.001;
calcPoint = center;
multiplier = 1.0;
animationCutoff = 0.00;
renderCutoff = 0.00;
attachmentCutoff = 0.00;
mesh("0.05")
{
name="Night_low.im";
}
mesh("0.2")
{
name="Night_med.im";
}
mesh("1.0")
{
name="Night.im";
}
Type : Document texte Date de modification : 27/04/2004 11:53 Taille : 311 octets
311 octets
Poste de travail