IUT de Provence – Site d'Arles
Département d'Informatique
2ème année

2006 - 2007

**IN Connaissances complémentaires**

**TP N°3- Max Script**

# Export d'un objet 3DSMax / Lecture en C++/OpenGL

*Nous allons voir au cours de ce TP comment écrire dans un fichier texte de manière à sauver la structure d'un objet 3D (sommets, faces, coordonnées de texture, texture). Ce fichier sera ensuite chargé et affiché en temps réel avec OpenGL dans un programme en C++.*

# Affichage d'un fichier texte

Il est possible de faire apparaître le contenu d'un fichier texte dans un éditeur afin de le consulter ou de le modifier :

```
edit "fichier.txt"
```

# Sélection d'un fichier

On peut faire apparaître une boîte de sélection de fichiers en lecture ou en écriture.

**Lecture :**

```
nom_fichier = GetOpenFileName()
if nom_fichier != undefined then
(
    edit nom_fichier
)
```

**Ecriture :**

```
nom_fichier = GetSaveFileName()
if nom_fichier != undefined then
(
    print nom_fichier
)
```

# Ecriture dans un fichier texte

**Exemple :**
On demande à l'utilisateur d'entrer un nom de fichier dans un sélecteur de fichiers avec la fonction **`GetSaveFileName`**. Si le nom entré est valide, on crée le fichier avec la fonction **`createfile`**. On y sauve une ligne de texte suivie d'un retour chariot, ainsi que le contenu de la variable **`une_variable`**. On ferme le fichier avec la fonction **`close`**.

```
une_variable = 10
nom_fichier = GetSaveFileName()
if nom_fichier != undefined then
(
    out_file = createfile nom_fichier
    format "Un peu de texte\n" to:out_file
    format "%" une_variable to:out_file
    close out_file
)
```

**Exemple :**
On crée une sphère, que l'on transforme en mesh éditable. On crée le fichier « sphere.txt » en écriture, dans lequel on écrit le nombre de sommets de la sphère suivit des coordonnées de chacun des sommets séparés par des espaces.

```
obj = sphere()                          -- Création d'une sphère
convertToMesh obj                       -- Conversion en mesh editable
out_file = createfile "sphere.txt"      -- Création du fichier
num_verts = obj.numverts                -- Nombre de sommets
format "%\n" num_verts to:out_file      -- Ecriture du nombre de sommets
for v = 1 to num_verts do               -- Boucle sur tous les sommets
(
    vert = getVert obj v                -- On obtient le v-ième sommet
    format "% % %\n" vert.x vert.y vert.z to:out_file   --Ecriture
)

close out_file                          -- Fermeture du fichier
```

**Ecrire un entier**

Pour écrire un float sous la forme d'un entier dans un fichier texte, on peut utiliser une conversion de type :

```
format "%" (une_variable as integer) to:out_file
```

# Lecture dans un fichier texte

**Exemple :**
Lecture d'une chaîne de caractères dans un fichier texte :

```
in_file = openFile "fichier.txt"
if in_file != undefined then
(
      chaine = readLine in_file
)
close in_file
```

# Texture

Il existe deux possibilités en MAXScript pour accéder aux informations de texture. Nous utiliserons la plus ancienne, maintenue pour des raisons de compatibilité, et qui est aussi la plus simple à utiliser. Elle suppose que chaque sommet ne peut avoir qu'une information de couleur et une information de texture. Vous regarderez dans l'aide de MAXScript en quoi consiste l'autre méthode, qui permet de gérer plusieurs textures par face.

Le principe est le suivant : à chacun des 3 sommets d'une face est associé un entier. Cet entier est un indice dans un tableau de coordonnées de textures. **Attention, ces indices sont numérotés à partir de 1**. En pratique, les 3 indices des 3 sommets de la face sont rangés dans un point3.
Si **`tmesh`** est un objet de type mesh editable dont on veut connaître les informations de texture :

```
-- Liste d'indices de texture des faces
for f = 1 to num_faces do
(
    indice = getTVFace tmesh f
    print indice.x              -- indice du 1er sommet de la face
    print indice.y              -- indice du 2ème sommet de la face
    print indice.z              -- indice du 3ème sommet de la face
)
```

Le nombre de coordonnées de texture s'obtient avec le champ **`numtverts`** d'un objet de type mesh. La liste des coordonnées de texture s'obtient avec la fonction **`getTVert`**. Cette fonction retourne un point3, dont les champs x et y correspondent aux coordonnées de texture 2D.

```
num_tverts = tmesh.numtverts

for v = 1 to num_tverts do
(
    t = getTVert tmesh v
    print t.x
    print t.y
)
```

Pour connaître le matériau associé à un objet obj :

```
materiau = obj.material
```

Ce qui nous intéresse dans ce matériau, c'est le nom de l'image utilisée comme texture de couleur diffuse. Pour cela, on accède au champ **`diffuseMap`** du matériau. Cet objet **`diffuseMap`** possède un champ **`bitmap`** correspondant à l'image. Enfin, le champ **`filename`** de l'objet **`bitmap`** correspond au nom du fichier.

```
Nom_texture_diffuse = mat.diffuseMap.bitmap.filename
```

**Exemple :**

```
utility texture_objet "Infos texture objet"
(

    button bouton_infos "Choisir objet" width:60 height:20

    fn GetGeometry o =
    (
        Superclassof o == Geometryclass and classof o != TargetObject
    )

    on bouton_infos pressed do
    (
        obj = pickobject filter:GetGeometry
        if isValidNode obj then
        (
            materiau = obj.material
            if( materiau != undefined )
            (
                nom_texture_diffuse = materiau.diffuseMap.bitmap.filename
                print nom_texture_diffuse
            )
            else
            (
                print "Aucun materiau"
            )
        )
    )
)
```

# Question 1

On veut écrire un script permettant de sélectionner un objet dans la scène et de le sauver dans un fichier dont on décide de l'organisation : tout d'abord la chaîne « FORMAT1 » servant d'identificateur, puis le nombre de sommets, le nombre de faces, la liste des sommets et la liste des faces.

1.1 Ecrire un utilitaire affichant un bouton portant le label « Choisir objet ».

1.2 En cliquant sur ce bouton, on devra pouvoir sélectionner un objet dans la scène (fonction **`pickobject`**).

1.3 Si un objet a été sélectionné, faire apparaître un sélecteur de fichiers en écriture (fonction **`GetSaveFileName`**).

1.4 Si un nom valide de fichier a été entré, créer un fichier (fonction **`createfile`**).

1.5 Ecrire dans le fichier la chaîne de caractères « FORMAT1 ». Cette chaîne servira à identifier notre format d'objet : le programme qui utilisera ce fichier vérifiera la présence de cet identificateur en début de fichier pour s'assurer qu'il s'agit d'un fichier au bon format.

1.6 Ecrire dans le fichier le nombre de sommets.

1.7 Ecrire dans le fichier le nombre de faces.

1.8 Ecrire dans le fichier la liste des sommets, en sautant une ligne pour chaque sommet. Sur chaque ligne, on écrira les valeurs de x, y et z du sommet en les séparant par un espace.

1.9 Ecrire dans le fichier la liste des faces, en sautant une ligne pour chaque face. Sur chaque ligne, on écrira les indices des 3 sommets des faces en les séparant par un espace.

1.10 Fermer le fichier (fonction **`close`**).

1.11 Testez le programme. Le fichier généré doit avoir l'allure suivante (ici pour un cube) ***sans les commentaires*** :

```
FORMAT1                  Identificateur
8                        Nombre de sommets
12                       Nombre de faces
-50.0 50.0 -50.0         Coordonnées des 8 sommets
50.0 50.0 -50.0              |
-50.0 50.0 50.0              |
50.0 50.0 50.0               |
-50.0 -50.0 -50.0            |
50.0 -50.0 -50.0             |
-50.0 -50.0 50.0             |
50.0 -50.0 50.0              |
0 2 3                    Indices dans la liste des sommets pour chacun
3 1 0                    des 3 sommets de chacune des 12 faces
4 5 7                        |
7 6 4                        |
0 1 5                        |
5 4 0                        |
1 3 7                        |
7 5 1                        |
3 2 6                        |
6 7 3                        |
2 0 4                        |
4 6 2                        |
```

1.12 Utilisez le programme de visualisation OpenGL se trouvant sur la page web pour tester la classe **`Objet3D`** fournie. Dans la fonction **`initialise_scene()`**, modifiez l'appel à la méthode **`charge()`** de l'objet 3D afin de préciser le nom du fichier à charger.
Attention, il faut impérativement que le fichier généré soit semblable à celui ci-dessus. Il ne faut pas, entre autres, rajouter des lignes vides entre les données.

# Question 2

On veut maintenant améliorer le script précédent en offrant la possibilité de sauver en plus des informations sur la texture de l'objet. L'organisation du fichier sauvé devra être la suivante : tout d'abord la chaîne « FORMAT2 » servant d'identificateur, puis le nombre de sommets, le nombre de faces, le nombre de coordonnées de textures, la liste des sommets, la liste des coordonnées de texture, la liste des faces, et la liste des indices dans la liste des coordonnées de texture pour chaque sommet de facette.

2 Ecrire un utilitaire affichant un bouton portant le label « Choisir objet ».

2.1 En cliquant sur ce bouton, on devra pouvoir sélectionner un objet dans la scène (fonction **`pickobject`**).

2.2 Si un objet a été sélectionné, faire apparaître un sélecteur de fichiers en écriture (fonction **`GetSaveFileName`**).

2.3 Si un nom valide de fichier a été entré, créer un fichier (fonction **`createfile`**).

2.4 Ecrire dans le fichier la chaîne de caractères « FORMAT2 ». Cette chaîne servira à identifier notre format d'objet : le programme qui utilisera ce fichier vérifiera la présence de cet identificateur en début de fichier pour s'assurer qu'il s'agit d'un fichier au bon format.

2.5 Ecrire dans le fichier le nombre de sommets.

2.6 Ecrire dans le fichier le nombre de faces.

2.7 Ecrire dans le fichier le nombre de coordonnées de texture.

2.8 Ecrire dans le fichier la liste des sommets, en sautant une ligne pour chaque sommet. Sur chaque ligne, on écrira les valeurs de x, y et z du sommet en les séparant par un espace.

2.9 Ecrire dans le fichier la liste des coordonnées de texture, en sautant une ligne à chaque fois. Sur chaque ligne, on écrira les valeurs de x, y des coordonnées en les séparant par un espace.

2.10 Ecrire dans le fichier la liste des faces, en sautant une ligne pour chaque face. Sur chaque ligne, on écrira les indices des 3 sommets des faces en les séparant par un espace.

2.11 Ecrire dans le fichier la liste des indices dans la liste des coordonnées de texture pour chaque sommet de face, en sautant une ligne pour chaque face. Sur chaque ligne, on écrira les 3 indices correspondant aux 3 sommets des faces en les séparant par un espace.

2.12 Ecrire dans le fichier le nom de l'image utilisée comme texture diffuse.

2.13 Fermer le fichier (fonction **`close`**).

2.14 Testez le programme. Le fichier généré doit avoir l'allure suivante (ici pour un cube) :

```
FORMAT2                     Identificateur
8                           Nombre de sommets
12                          Nombre de faces
-50.0 50.0 -50.0            Coordonnées des 8 sommets
50.0 50.0 -50.0                 |
-50.0 50.0 50.0                 |
50.0 50.0 50.0                  |
-50.0 -50.0 -50.0               |
50.0 -50.0 -50.0                |
-50.0 -50.0 50.0                |
50.0 -50.0 50.0                 |
12                          Nombre de coordonnées de texture
0.0 0.0                     Liste des 12 coordonnées de texture
1.0 0.0                         |
0.0 1.0                         |
1.0 1.0                         |
0.0 0.0                         |
1.0 0.0                         |
0.0 1.0                         |
1.0 1.0                         |
0.0 0.0                         |
1.0 0.0                         |
0.0 1.0                         |
1.0 1.0                         |
9 11 10                     Indices de texture des 3 sommets de chacune
10 8 9                      des 12 faces
8 9 11                          |
11 10 8                         |
4 5 7                           |
7 6 4                           |
0 1 3                           |
3 2 0                           |
4 5 7                           |
7 6 4                           |
0 1 3                           |
3 2 0                           |
0 2 3                       Indices dans la liste des sommets pour chacun
3 1 0                       des 3 sommets de chacune des 12 faces
4 5 7                           |
7 6 4                           |
0 1 5                           |
5 4 0                           |
1 3 7                           |
7 5 1                           |
3 2 6                           |
6 7 3                           |
2 0 4                           |
4 6 2                           |
c:\temp\briques.tga         Nom de la texture
```

2.15 Modifiez le programme de visualisation OpenGL de la question précédente de manière à lire dans le fichier les nouvelles données. On chargera l'image utilisée comme texture de couleur diffuse. On plaquera cette image sur l'objet dans la méthode d'affichage **`affiche()`** de la classe **`Objet3D`**.

2.16 Sauvez également dans le fichier au FORMAT2 les propriétés de couleur ambiante, diffuse et spéculaire de l'objet, et utilisez les dans le programme en C++.