

# Initiation à la programmation COFFEE. (partie 5.)

Par Tengaal (tengaal@libertysurf.fr) 01/07/2002.

## Les matériaux

---

Le langage COFFEE permet de tout contrôler dans Cinema 4D, tout ce qui est créé dans une scène 3D est manipulable par la programmation. C'est pourquoi nous allons maintenant sortir du gestionnaire d'objets pour nous tourner vers le gestionnaire de matériaux, car ce sont également des objets de Cinema 4D dont les paramètres sont accessibles par leurs containers, il est donc très simple d'agir sur chaque canaux de la matière et de modifier les couleurs, les intensités, les images et tous les paramètres spécifiques à chaque canal.

Info : Les matériaux sont des objets de type BaseMaterial et les canaux sont des objets de type BaseChannel. (voir documentation SDK COFFEE)

## Trouver et accéder à un matériaux.

---

Pour commencer, nous allons voir comment pointer sur une matière, en effet, si notre scène 3D comporte différents matériaux et que l'on souhaite agir sur l'un en particulier, nous devons être capable de le localiser.

Souvenez-vous, dans la partie 3, une fonction permettait de trouver un objet par son nom :

**Document -> FindObject (Nom\_objet) ;**

Et bien, pour les matériaux, une fonction similaire est spécialement adaptée:

**Document -> FindMaterial( Nom\_matière) ;**

Voyons donc, par l'exemple, comment utiliser ce pointage :

- Créons dans une scène, 4 matériaux.
- Donnez à ces matériaux les noms suivants : "mat01" , "mat02" , "mat03" et "mat04" .
- On crée ensuite un objet Neutre auquel on ajoute une nouvelle expression COFFEE.
- Puis créez un autre objet Neutre avec comme nom "bouton".

Voici le listing de notre expression :

```
main(doc,op)
{
    var clic=GetActiveObject(doc) ;           // clic est donc l'objet sélectionné dans le gestionnaire
    if ( !clic) return ;
    var clic_nom=clic->GetName() ;           //on lit le nom de cet objet cliqué
    if (clic_nom != "bouton") return ;      // si ce n'est pas notre "bouton" on quitte

    // sinon on exécute notre programme
    doc -> SetActiveObject(op) ;             // important : on désélectionne aussitôt notre "bouton"
                                           // en sélectionnant notre objet op
    var matiere=doc -> FindMaterial("mat02") ; // nous allons modifier la matière "mat02"
    if ( !matiere)
    {
        op ->SetName("matière inconnue") ; // si aucune matière de ce nom n'existe,
        return ;                             // on l'indique dans le nom de notre objet Neutre
                                           // puis on quitte le programme.
    }

    // sinon, la matière à été trouvée, on l'indique dans le nom de notre objet Neutre (op)
    op -> SetName("matière trouvée !") ;

// fin
}
```

Fermez la fenêtre du script puis cliquez sur "bouton" et là le nom de l'objet Neutre qui contient notre expression va vous informer sur la recherche de la matière "mat02".

Cette petite astuce qui consiste à utiliser le nom d'un objet pour afficher des informations sur le déroulement du programme est très pratique pendant le développement d'un outil en COFFEE !

Une autre fonction permet d'accéder au premier matériau du document sans tenir compte de son nom : **matiere = document -> GetFirstMaterial()**.

Ensuite, on accède aux autres matières avec le même principe que pour les objets :

**matiere\_suivante = matiere -> GetNext()** et **matiere\_precedente = matiere -> GetPrev()**.

## Les paramètres des matériaux

---

Maintenant que nous savons comment accéder à une matière de notre document, nous allons voir comment agir sur les paramètres de chaque canal. Tout d'abord, il faut accéder aux canaux que l'on souhaite modifier, on peut activer et désactiver chacun d'eux mais également changer les valeurs des paramètres.

```
var matiere = doc -> GetFirstMaterial(); // on accède ici à la première matière du document
var canal = matiere -> GetChannel(ID_canal); // on accède au canal défini par son identifiant
```

Comme pour les paramètres des containers, Cinema 4D contient des identifiants (ID) prédéfinis qui permettent d'accéder à chaque canal de notre matériau, voici la liste :

Identifiant canal	Description
CHANNEL_COLOR	Canal Couleur
CHANNEL_DIFFUSION	Canal Diffusion
CHANNEL_LUMINANCE	Canal Luminescence
CHANNEL_TRANSPARENCY	Canal Transparence
CHANNEL_REFLECTION	Canal Réflexion
CHANNEL_ENVIRONMENT	Canal Environnement
CHANNEL_FOG	Canal Brouillard
CHANNEL_BUMP	Canal Relief
CHANNEL_ALPHA	Canal Alpha
CHANNEL_SPECULAR	Canal Spécularité
CHANNEL_SPECULARCOLOR	Canal Couleur spéculaire
CHANNEL_GLOW	Canal Lueur
CHANNEL_DISPLACEMENT	Canal Déplacement

Note : Les matériaux de la version 7 de Cinema 4D, ont des paramètres supplémentaires regroupés sous le nom Illumination (pour les caustiques, la radiosité et l'atténuation de réflexion.). La documentation actuelle du SDK COFFEE ne mentionne pas cette nouvelle option des matériaux.

Avec la fonction **matiere -> SetChannelState(ID\_canal, TRUE ou FALSE)**, on peut activer (TRUE) ou désactiver (FALSE) un canal précis (ID\_canal), de même, on peut connaître cet état par la fonction **var etat = matiere -> GetChannelState(ID\_canal)** où la valeur etat reçoit TRUE ou FALSE.

**Exemple** : On veut activer le canal Luminescence et désactiver le canal Couleur.

```
var matiere = doc -> FindObject("Ma_matiere"); // on pointe sur "Ma_matiere"
matiere -> SetChannelState(CHANNEL_COLOR, FALSE); // désactiver Couleur
matiere -> SetChannelState(CHANNEL_LUMINANCE, TRUE); // activer Luminescence
matiere -> Update(); // actualiser la matière
```

L'accès aux paramètres d'un canal nécessite tout d'abord d'accéder au container du canal (à l'ensemble de ses paramètres.) puis de modifier la valeur souhaitée en la localisant encore une fois avec son identifiant, dont voici la liste :

Identifiants paramètres	Types	Description
CH_COLOR	[vector]	Couleurs R, V, B de 0.00 (0%) à 1.00 (100%)
CH_BRIGHTNESS	[float]	Intensité de 0.00 (0%) à 1.00 (100%)
CH_TEXTURE	[string]	Nom de l'image Texture
CH_BLUR_OFFSET	[float]	Offset du flou
CH_BLUR_STRENGTH	[float]	Intensité du flou
CH_INTERPOLATION	[int]	Interpolation (voir la fenêtre concernée)
CH_MIXMODE	[int]	Mode de mélange MIX (voir fenêtre)
CH_MIXSTRENGTH	[float]	Intensité du Mix

Dans le cas d'une texture vidéo, voici les paramètres accessibles par le bouton Modifier du canal :

Identifiants paramètres	Types	Description
CH_TIME_FROM	[int]	De la position...
CH_TIME_TO	[int]	A la position...
CH_TIME_FPS	[int]	Nombre d'images par seconde de la séquence
CH_TIME_MODE	[int]	Mode de la séquence mode (voir fenêtre concernée)
CH_TIME_TIMING	[int]	Durée de la séquence (voir fenêtre concernée)
CH_TIME_START	[BaseTime]	Début de la séquence
CH_TIME_END	[BaseTime]	Fin de la séquence
CH_TIME_LOOPS	[int]	Boucles de la séquence

**Exemple** : On veut changer la couleur de la matière "mat01" en R :80% V :20% B :100%, et définir l'intensité à 85%.

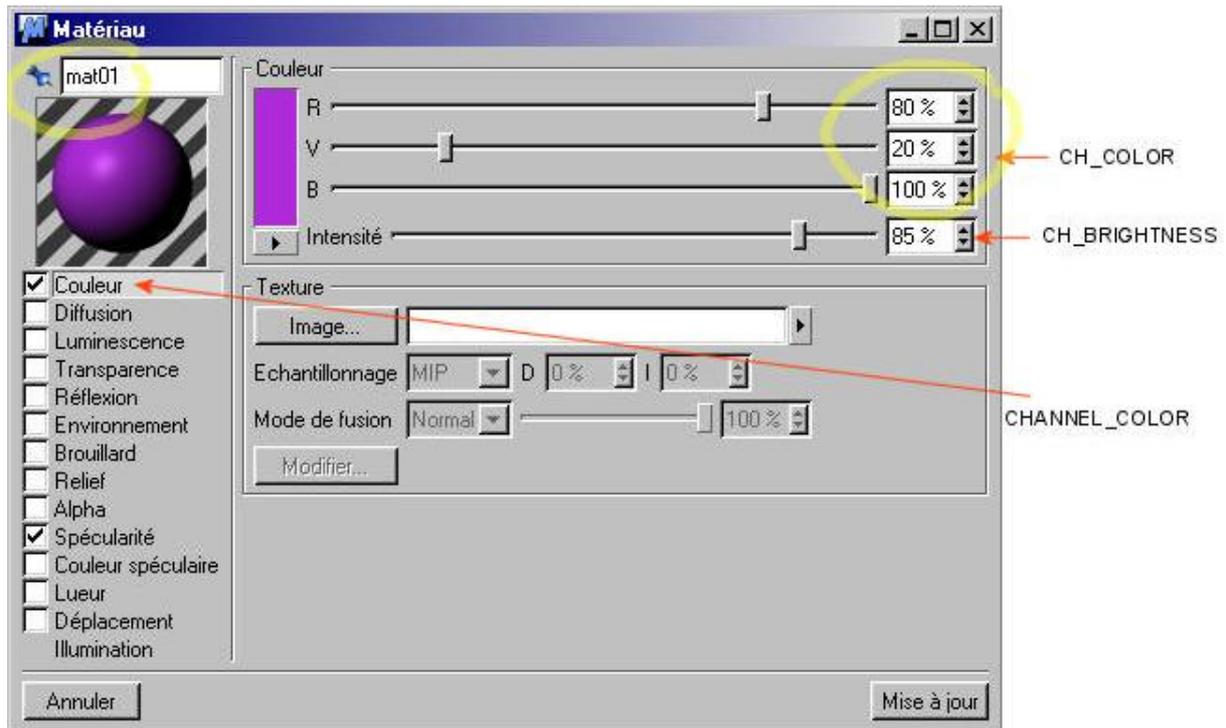
```

var matiere=doc -> FindMaterial("mat01") ; // on pointe sur la matière "mat01"
var canal=matiere -> GetChannel(CHANNEL_COLOR) ; // on accède au canal Couleur
var canal_cont=canal -> GetContainer() ; // on accède au container du canal
canal_cont -> SetData(CH_COLOR, vector(0.80,0.20,1.00)) ; // vector ( R , V , B )
canal_cont -> SetData(CH_BRIGHTNESS,0.85) ; // intensité à 85%
canal -> SetContainer(canal_cont) ; // on applique ces changements au canal

// Indispensable : actualiser le matériau
matiere -> Update() ;

```

Voici le résultat de cette action :



Cette 5<sup>e</sup> partie arrive à son terme, vous avez donc en main les bases indispensables pour manipuler les paramètres des matériaux, les fonctions particulières énoncées dans cette partie s'ajoutent aux fonctions communes à tous les objets de Cinema 4D, c'est à dire que la création d'un matériau se fait de la même manière que pour un objet classique, ainsi que la définition de son nom... Par exemple pour créer un matériau en COFFEE :

```

var matiere = new (BaseMaterial) ; // On créé un objet Matière
matiere -> SetName ("mat01") ; // on lui donne un nom
document -> InsertMaterial (matiere,NULL) ; // on l'insère dans notre scène en tant que matière
matiere -> Update () ; // on actualise la matière créée

```