

IN Connaissances complémentaires

TP N°1 - MaxScript

Introduction

MAXScript est le langage de script de **3DS MAX**. Il permet de :

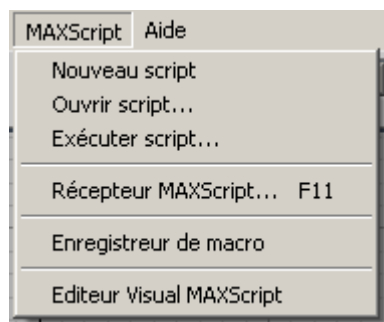
- Automatiser certains aspects de l'utilisation du programme, comme la modélisation, l'animation, les matériaux, et le rendu, ...
- De contrôler interactivement le programme grâce à des commandes que l'on peut taper en ligne
- De mettre en place des scripts utilisant des panneaux déroulants, boutons, ...
- De transformer ces scripts en macro et de les installer comme bouton dans la barre de menu, ou comme choix, dans un menu, ou de leur assigner un raccourci composé de combinaisons de touches.
- De construire des importations et exportations utilisant des fichiers ASCII et binaires.
- D'écrire des contrôleurs procéduraux qui peuvent accéder à tous les états de la scène.
- De construire des outils automatisés, comme par exemple des scripts de rendus automatisés.

Accéder à MAXScript

Il y a différentes façons d'accéder à MAXScript.

Vous pouvez ouvrir le **Récepteur MAXScript** à partir du menu **MAXScript** :


Menu Principal → MAXScript → Récepteur MAXScript.

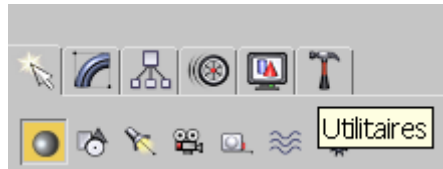


Vous pouvez aussi faire un clic droit sur le mini récepteur de MAXScript situé dans le coin gauche de l'interface de 3DS max.

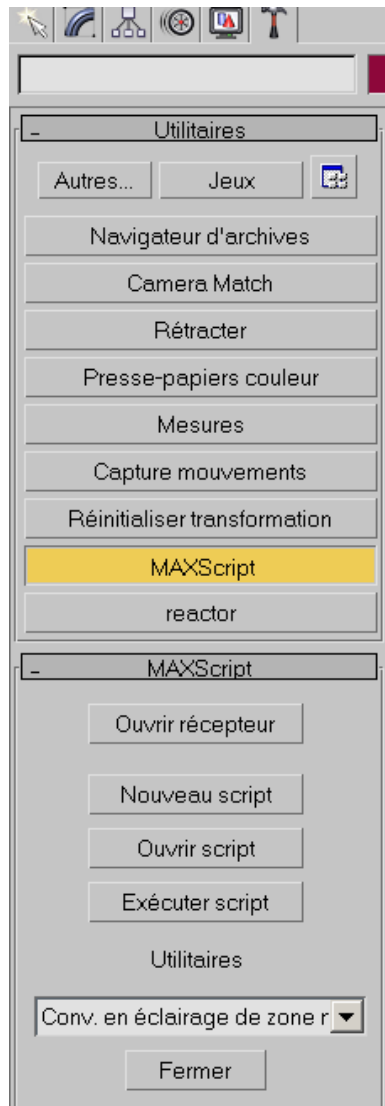


Une autre façon d'ouvrir le récepteur de MAXScript est de suivre la procédure suivante :

1. Dans le panneau de commande, choisissez **Utilitaires** ()



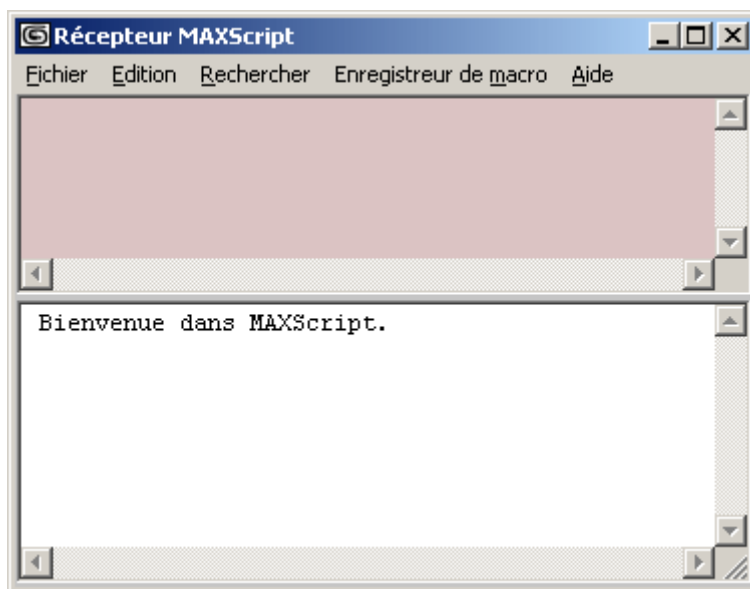
2. Cliquez ensuite sur le bouton **MAXScript**. Le panneau déroulant concernant MAXScript apparaît.



3. Cliquez sur le bouton **Ouvrir récepteur**.

Le récepteur de MAXScript

Le récepteur de MAXScript est un interpréteur interactif de langage MAXScript. Vous pouvez entrer des commandes MAXScript dans cette fenêtre et valider avec la touche **entrée** afin de l'exécuter immédiatement.



Le Mini- récepteur de MAX

La partie gauche de la barre de statut contient un mini récepteur de MAXScript dont on peut redimensionner la taille.



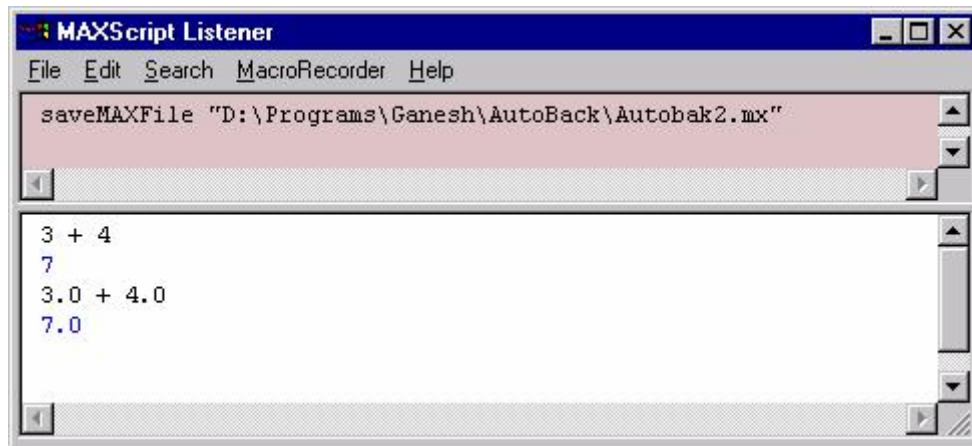
Entrer des informations dans MAXScript

Il y a plusieurs types d'informations que vous pouvez vouloir entrer dans le récepteur de MAXScript, comme des valeurs numériques, des chaînes de caractères ou encore des tableaux.

Les informations sont généralement introduites dans MAXScript en tapant une valeur ou une expression et en pressant la touche **entrée**.

Entrer un nombre dans MAXScript

MAXScript permet de faire la distinction entre deux types de nombres : entiers et réels.



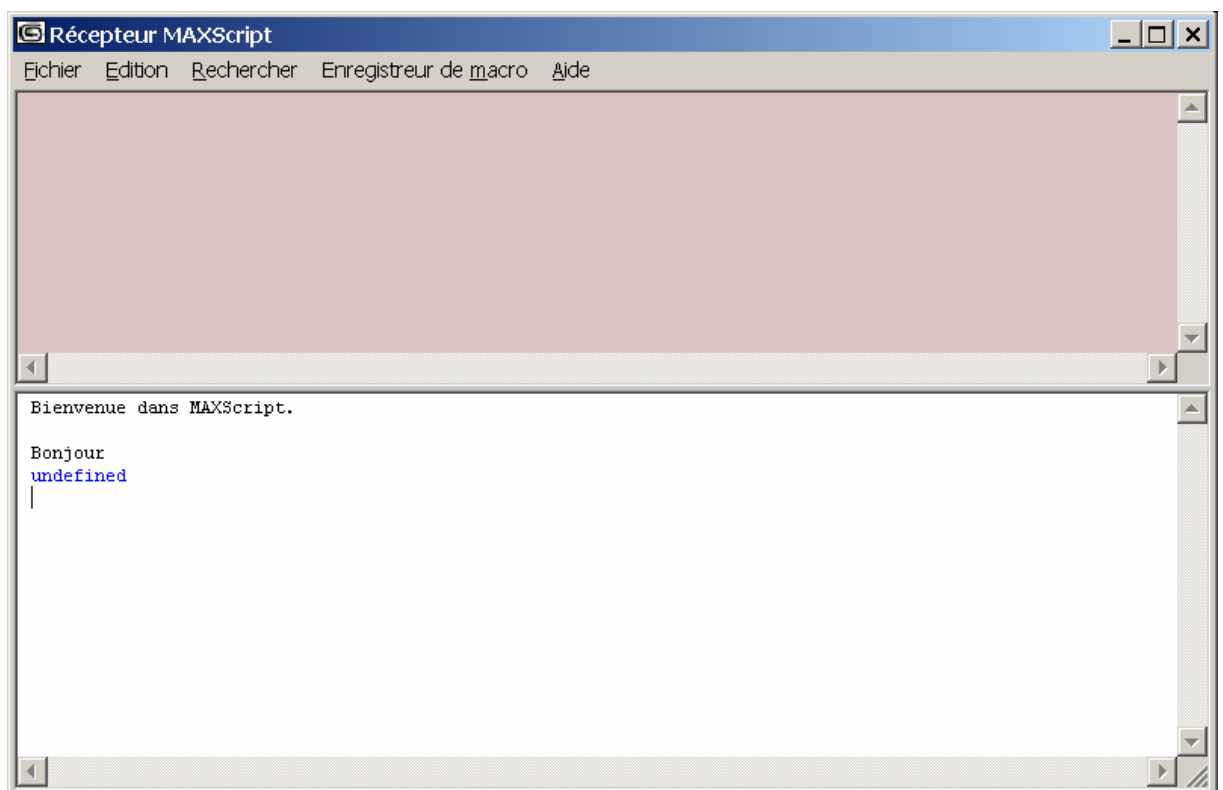
Entrer une chaîne de caractère dans MAXScript

MAXScript est un langage basé sur les expressions. Chaque expression doit donner une valeur. Les chaînes sont introduites entre guillemets.

Questions :

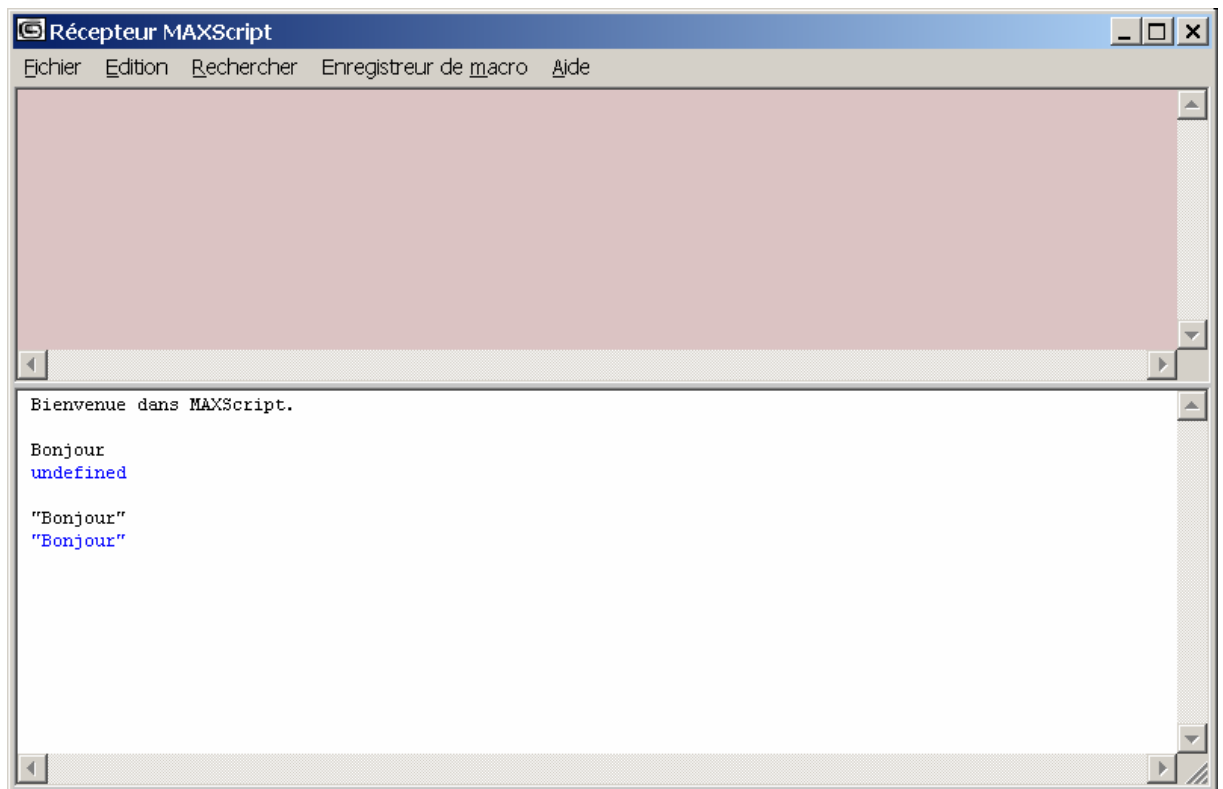
Avec le récepteur de MAXScript, écrire BONJOUR

1. Tapez à la place du prompt, BONJOUR et pressez entrée.
2. MAXScript répond en disant **undefined**. MAXScript ne connaît pas le mot BONJOUR.



3. Tapez "BONJOUR" (entre guillemets) puis Entrée.
4. La réponse est ici Bonjour.

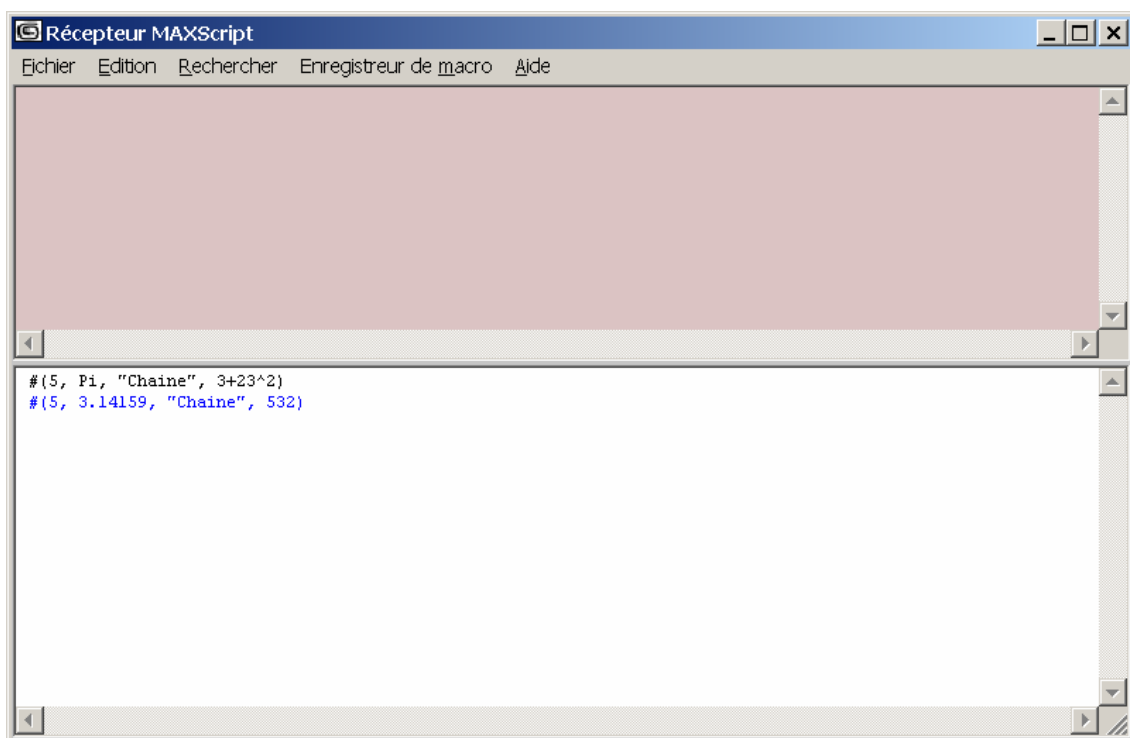
MAXScript sait donc traiter les chaînes de caractères délimitées par des guillemets.



Entrée d'un tableau dans MAXScript

Vous pouvez utiliser un tableau afin de grouper plusieurs éléments dans une collection.

Dans MAXScript, chaque élément dans le tableau peut contenir n'importe quel type de valeurs.



Un tableau doit être défini par le caractère # et des parenthèses.

Assigner des valeurs aux variables

La syntaxe pour affecter des valeurs à des variables est :

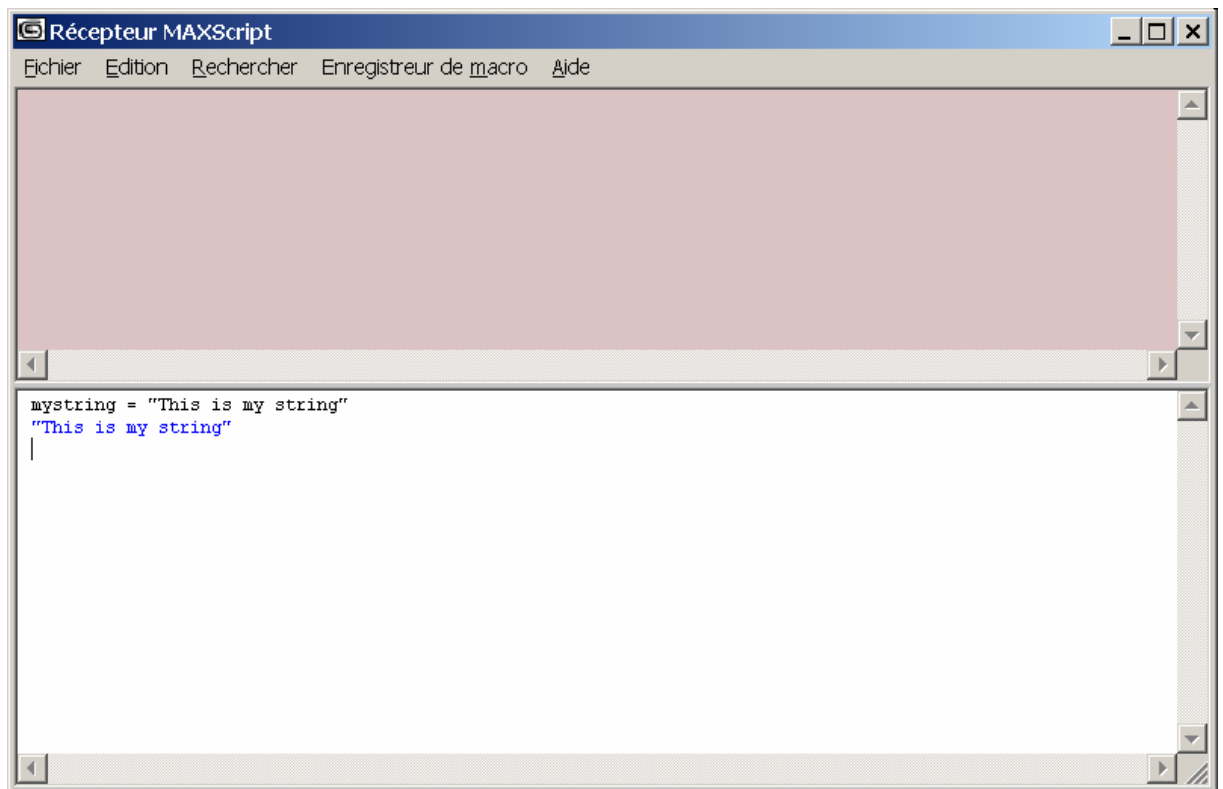
```
Nom_variable = Valeur_variable
```

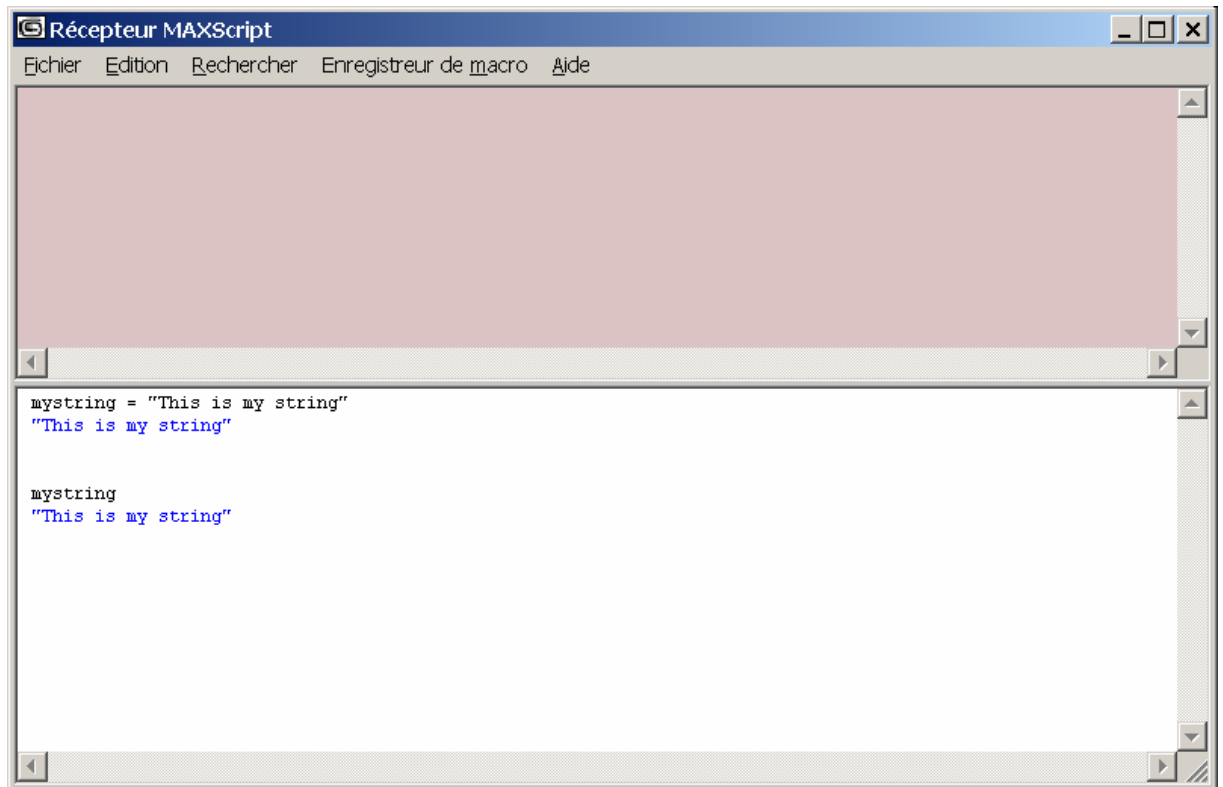
Une variable commence par un caractère ou "_" (souligné), suivi par des nombres ou des caractères. La valeur de la variable peut être une chaîne de caractères, un nombre, ou une expression.

Pour assigner une valeur à une variable, tapez la commande suivante :

```
mystring = "This is my string."
```

MAXScript retourne la valeur de la variable chaîne. Vous pouvez maintenant faire référence à cette variable par son nom.





Il faut noter que MAXScript ne distingue pas les différences de casses (majuscules/ Minuscules).

L'enregistreur de Macro

L'enregistreur de Macro MAXScript capture les actions faites par l'utilisateur et génère les commandes MAXScript qui correspondent à ces actions. Les lignes de codes générées par l'enregistreur de macro peuvent être visualisées dans le récepteur MAXScript.

Nous allons tout d'abord réinitialiser la scène.

Questions :

- 1- Pour lancer l'enregistreur de macro, allez dans le menu **Enregistreur de Macro** du récepteur MAXScript. L'activer. Toutes vos manipulations sont maintenant enregistrées. Ne pas fermer le récepteur de MAXScript afin de suivre les commandes qui vont y apparaître en cours de manipulation.
- 2- Créer une sphère
- 3- Créer une boîte
- 4- Aller dans le panneau modifier et sélectionner la sphère
- 5- La déplacer
- 6- Sélectionner la boîte et la supprimer.
- 7- Arrêter l'enregistreur de MACRO

Les commandes suivantes sont apparues dans le récepteur de MAXScript :

```
Sphere radius:60 smooth:on segs:32 chop:0 slice:off sliceFrom:0 sliceTo:0
mapCoords:on recenter:off transform:(matrix3 [1,0,0] [0,0,1] [0,-1,0] [-73.8931,-
3.88397e-006,88.855]) isSelected:on
```

```

Box lengthsegs:1 widthsegs:1 heightsegs:1 length:72.6718 width:75.1145
height:47.0229 mapCoords:on transform:(matrix3 [1,0,0] [0,0,1] [0,-1,0] [9.46565,-
1.17454e-006,26.8702]) isSelected:on

select $Sphère01

move $ [70.9678,-12.3813,0]

select $Boîte01

actionMan.executeAction 0 "40020" -- Editer: Supprimer objets

delete $

```

Dessiner une boîte avec MAXScript

Vous devez réinitialiser 3DSMax et effacer les lignes contenues dans le récepteur de MAXScript.

Comme nous l'avons vu dans le paragraphe précédent, nous pouvons créer une boîte grâce à la ligne de commande `box()`.

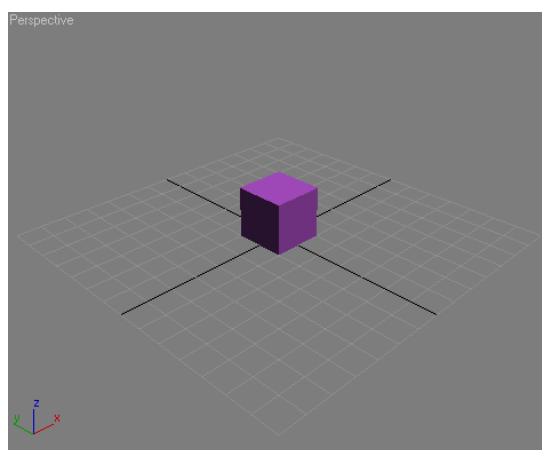
Cette ligne permet de créer une boîte ayant les paramètres par défaut. Il est en général pratique d'assigner un objet dans une variable. Donnez- lui le nom `mybox` grâce à la commande suivante :

```
mybox = box()
```

Les parenthèses permettent de signaler qu'il faut prendre les paramètres par défaut.

Si vous voulez spécifier les paramètres pendant la création, comme la taille de la boîte par exemple, utilisez la syntaxe suivante :

```
mybox = box length:20 width:20 height:20
```



Modification des paramètres de la Boîte

Les propriétés de la boîte sont accessibles en suffixant le nom de la variable associée à la boîte par la propriété à laquelle vous voulez accéder. Par exemple, pour accéder à la hauteur, vous utilisez la ligne de commande : `mybox.height`

Plutôt que le nom de la variable, on peut aussi utiliser le nom de l'objet (champ **name** de la variable) en le faisant précéder du symbole **\$**. Par exemple, si la boîte s'appelle **Boîte01**, on pourra accéder à sa hauteur avec : **\$Boîte01.height**

Tous les objets 3ds max ont des paramètres de création comme la longueur, la largeur, la hauteur pour les boîtes, le rayon pour les cercles... Ces objets ont aussi des propriétés liées aux transformations comme la mise à l'échelle, la rotation, la position, et des propriétés générales comme la couleur, ...

Syntaxe de différentes opérations :

```
mybox.pos = [10,-75,20]
```

```
mybox.scale = [1.5,1.5,1.5]
```

Pour modifier la couleur, vous pouvez utiliser des constantes prédéfinies comme **blue**, **black**,... ou utiliser directement le code RGB. Exemples :

```
mybox.wireColor = blue
```

ou

```
mybox.wireColor = (color 255 0 255)
```

Vous pouvez aussi trouver tous les paramètres d'un objet grâce à la commande **showClass()**. Par exemple, pour la boîte :

```
showclass "box.*"
```

* permet de visualiser tous les paramètres de la boîte.

Questions :

- 1- Renommez la boîte **mybox** grâce à la propriété **Name**.
- 2- Changer sa couleur grâce à la propriété **wireColor**. Créer une boîte verte.
- 3- Placer la boîte en **[0,-75,0]** grâce à la propriété **pos**.

Appliquer des transformations Standard

Déplacer la boîte

Afin de déplacer l'objet, utilisez la commande suivante :

```
move name_obj [<x,y,z>]
```

Exemple :

```
move mybox [10,0,0]
```

```
move mybox [10,0,0]
move mybox [10,0,0]
```

va déplacer votre boîte **mybox** de 30 unités le long de l'axe des X.

Mise à l'échelle de la boîte

```
scale name_obj [<x,y,z>]
```

Rotation de la boîte

```
rot_angle = eulerangles x y z
```

```
rotate mybox rot_ange
```

Autres transformations sur la boîte

Les lignes suivantes permettent de préparer la boîte à des transformations plus complexes. On change par exemple le nombre de segments sur chaque face et prépare à un mappage de texture. Exemple :

```
mybox.lengthsegs = 10
mybox.widthsegs = 10
mybox.heightsegs = 10
```

```
mybox.mapCoords = true
```

Questions :

- 1- Déplacer votre boîte de 30 vers le haut
- 2- Appliquer une rotation à la boîte de 30 degrés selon l'axe des Y.

Enregistrement de script dans un fichier

Dans la suite du TP, nous allons enregistrer les scripts dans un fichier. Nous devons ensuite exécuter le script en l'appelant.

Procédure à suivre :

- a- Dans le menu MAXScript, faire **nouveau Script**.
- b- Un éditeur s'ouvre à l'écran. Enregistrer ce script (l'extension sera alors **.ms**)
- c- Pour lancer un script que vous aurez réalisé, aller dans le menu MAXScript et sélectionnez **Exécutez Script**.

Créer et assigner un Modificateur

Il faut utiliser la commande **addModifieur** et spécifier le nom du modificateur que vous souhaitez utiliser avec les paramètres appropriés.

Par exemple, pour créer un modificateur Courbure réglé à 30 ° et l'appliquer à votre boîte, vous devez écrire :

```
addModifieur mybox (twist angle:30)
```

Si l'angle ne vous paraît pas correct, vous pouvez modifier un des paramètres du modificateur :

```
mybox.twist.angle = 60
```

Remarque : Vous pouvez avoir besoin de mettre des informations entre commentaires.

Vous utiliserez `/* ... */`

Questions :

- 1- Créez un nouveau script que vous nommerez **TP5.ms**
- 2- Créer une nouvelle boîte
- 3- Lui appliquer une courbure de 30° d'angle
- 4- Enregistrer le script
- 5- L'exécuter

Grouper des objets

Afin de grouper des objets, il faut dans un premier temps les sélectionner. A supposer que l'on veuille grouper une boîte, une sphère et un cylindre, on écrira :

```
-- Création des objets
b = box()
s = sphere()
c = cylinder()

-- Sélection de plusieurs objets
select #(b,s,c)

-- Groupage des objets sélectionnés
group selection
```

Le nom donné au groupe sera par défaut «**Group01**» (puis «**Group02**», etc.). Pour donner explicitement un nom au groupe lors de sa création, on utilisera la syntaxe suivante :

```
group selection name:"un_groupe"
```

Animer un objet

MAXScript a plusieurs formes de syntaxes qui imitent le fonctionnement du bouton **Clé auto** dans l'interface 3ds max. Voici un exemple où l'on déplace une boîte tout en modifiant son échelle :

```
animate on
(
at time 0 (
    mybox.pos = [-100, 0, 0]
    mybox.scale = [1, 1, 0.25]
)
at time 100 (
    mybox.pos = [100, 0, 0]
    mybox.scale = [1, 1, 3]
)
)
```

Dans les questions suivantes, vous devez modifier le script ci dessus afin d'animer la boîte.

- 1- Faire tourner la boîte au cours de l'animation de 90 selon l'axe des X
- 2- Exécuter le script
- 3- Lancer l'animation

Définition de macro scripts

Les macro scripts sont des scripts associés à des boutons de barres d'icônes, et qui sont exécutés lorsqu'on clique sur ces boutons. Ces scripts doivent être définis selon la syntaxe suivante :

```
macroScript <name> [category:<string>]
                    [buttonText:<string>]
                    [toolTip:<string>]
                    [icon:#(<string>, <index>) | icon:<string>]
                    [ silentErrors:<boolean> ]
( <macro_script_body> )
```

Exemple :

```
macroScript ma_macro category:"Cameras"
                    tooltip:"exemple de macro"
                    icon:#("Cameras",2)
(
    box()
)
```

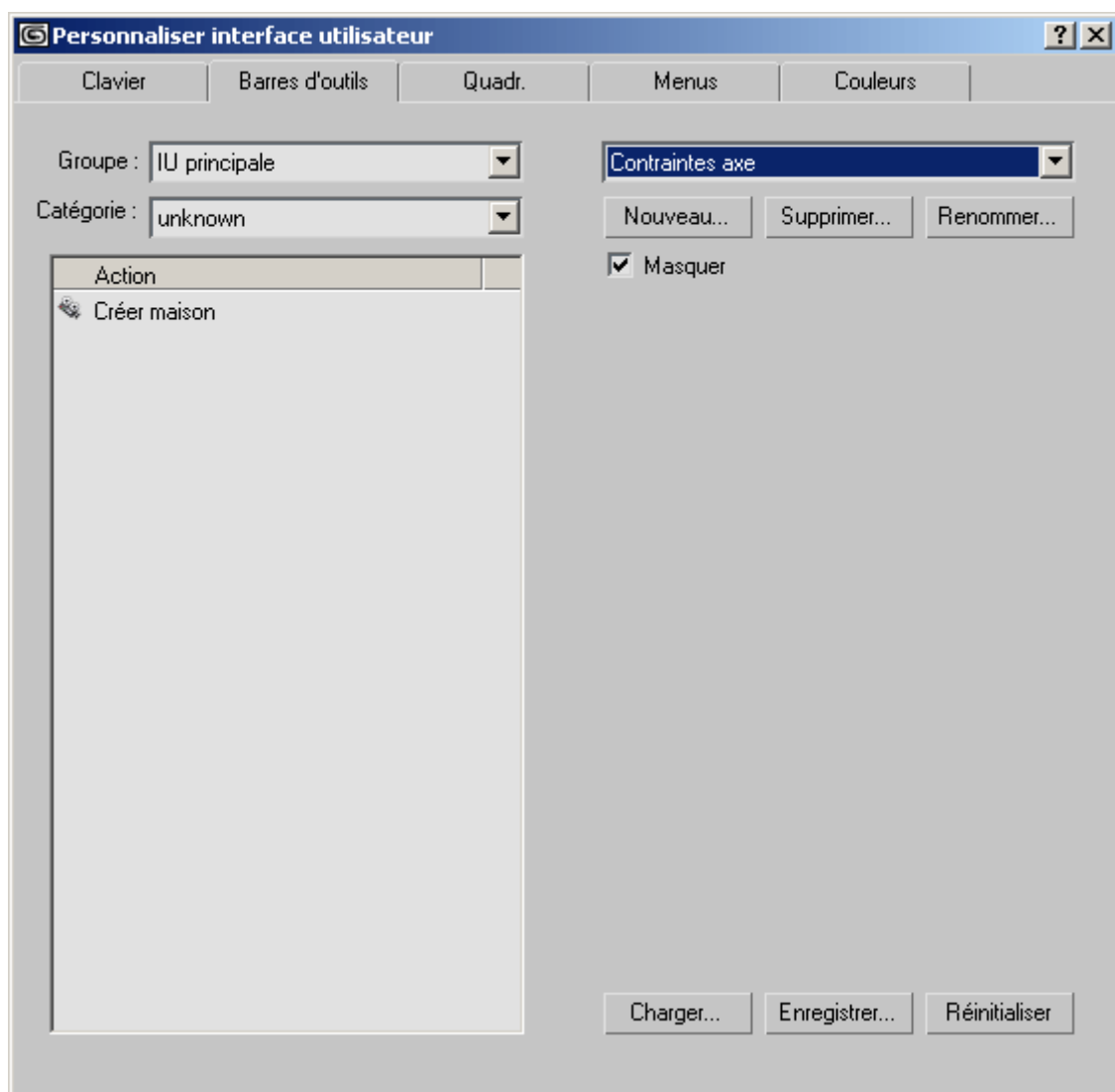
category (optionnel) indique la catégorie dans laquelle doit être rangée la macro. Si cette information n'est pas spécifiée, la catégorie sera **unknown**.

tooltip (optionnel) indique le message d'aide qui apparaîtra à l'écran lorsqu'on place le curseur de la souris sur le bouton associé à la macro.

icon (optionnel) indique l'icône à utiliser dans la barre d'outil pour représenter la macro. Dans l'exemple ci-dessus, on utilise la 2^{ème} icône de la bibliothèque d'icône relative aux caméras. On pourrait également spécifier le nom d'un fichier bmp.

Pour associer ce script à un bouton de barre d'icônes, cliquer avec le bouton droit sur une barre d'icônes et sélectionnez **Personnaliser**. La boîte de dialogue **Personnaliser interface utilisateur** apparaît. Sélectionnez la catégorie à laquelle appartient votre macro. Les noms des macros appartenant à cette catégorie sont affichés dans une liste. En sélectionnant la macro dans la liste, il est possible de placer son icône par drag&drop dans une barre d'icônes.

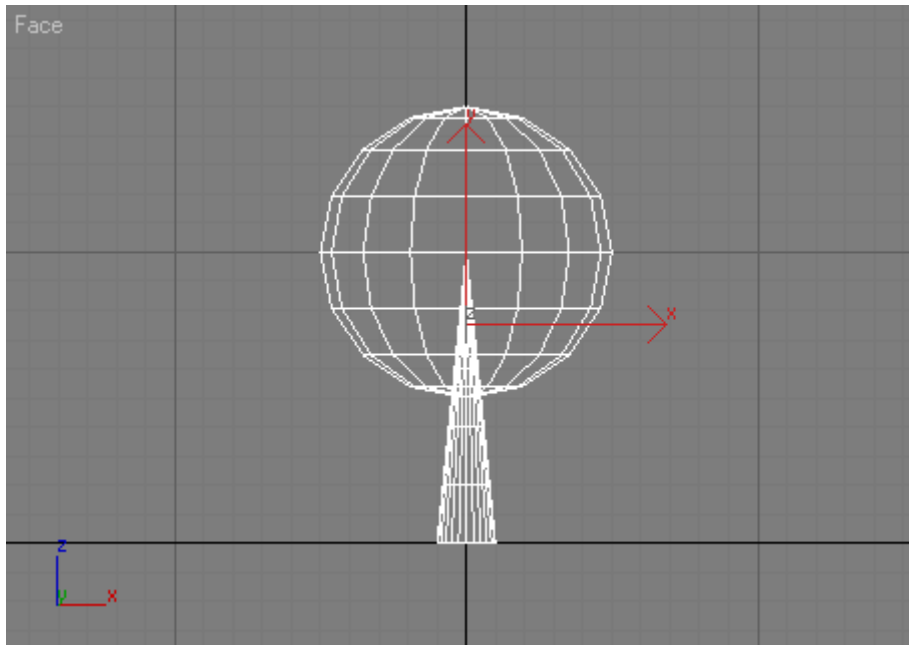
Lorsque vous modifiez le script dans sa fenêtre d'édition, il faut le réactualiser avec l'option **Evaluer tout** du menu **Fichier**.



Grâce à l'onglet **Menus** de la boîte de dialogue **Personnaliser interface utilisateur**, il est aussi possible de placer la macro dans l'un des menus.

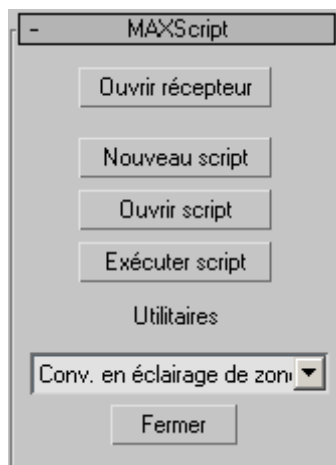
Question :

Ecrire une macro script **Arbre** permettant d'afficher un arbre constitué d'un cône pour le tronc et d'une sphère pour le feuillage. Le cône sera de couleur marron, avec un rayon1=10 (**radius1**), un rayon2=0 (**radius2**), une hauteur=100 (**height**). La sphère sera verte, avec un rayon=50 (**radius**). La macro devra grouper ces deux primitives.



Scripts utilitaires

Il est possible de faire des scripts présentant une interface graphique avec des boutons, des champs éditables, des cases à cocher, etc., permettant ainsi de régler interactivement les paramètres du script. En sélectionnant le nom du script dans la liste déroulante **Utilitaires** du panneau **MAXScript**, l'interface du script apparaît en dessous.



La syntaxe d'un script utilitaire est la suivante :

```
utility <var_name> <description_string>
  [ rolledUp:<boolean> ] [ silentErrors:<boolean> ]
( <utility_body> )
```

Exemple

```
utility creation_objet "Création objet"
(
  local   taille_boite=30

  checkbox check_cone "Ajout d'un cône"

  spinner spinner_taille "Taille:" range:[-1000,1000,taille_boite]

  bouton bouton_creer "Créer" width:60 height:20

  bouton bouton_a_propos "A propos..." width:60 height:20

  colorpicker couleur "Couleur:" color:[250,0,0]

  on bouton_a_propos pressed do
  (
    messagebox "Création\n\npar S. Thon\nVersion 1.0" title:"A propos..."
  )

  on bouton_creer pressed do
  (
    if check_cone.checked then
    (
      c=cone()
      c.wireColor = couleur.color
    )
    b=box width:taille_boite height:taille_boite length:taille_boite
    b.wireColor = couleur.color
  )

  on spinner_taille changed value do  -- quand la valeur du spinner change
  (
    taille_boite = spinner_taille.value
  )
)
```



Le bouton **Fermer** est automatiquement rajouté par 3DS MAX.

Question

Ecrire un script utilitaire permettant comme précédemment de créer des arbres. L'interface comportera une case à cocher permettant de rajouter ou non la sphère correspondant au feuillage. Grâce à des **spinner**, il devra être possible de régler la hauteur et le rayon du cône, ainsi que le rayon de la sphère. Deux **colorpicker** permettront de régler la couleur du tronc et du feuillage. L'arbre sera créé lorsqu'on cliquera sur un bouton **Créer**.