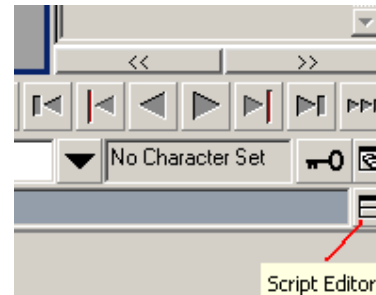


Langage MEL introduction

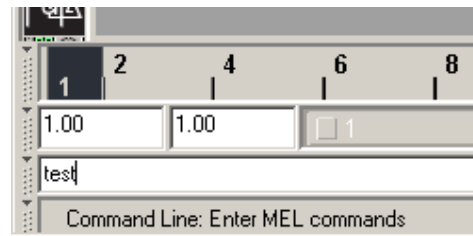
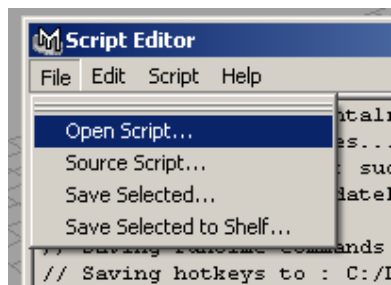
Faire un fichier MEL

Tout d'abord, il faut ouvrir la fenêtre permettant de gérer les scripts MEL. Pour cela cliquez sur l'icône « script editor » en bas à droite. Vous pouvez aussi passer par Window → General Editors → Script Editor

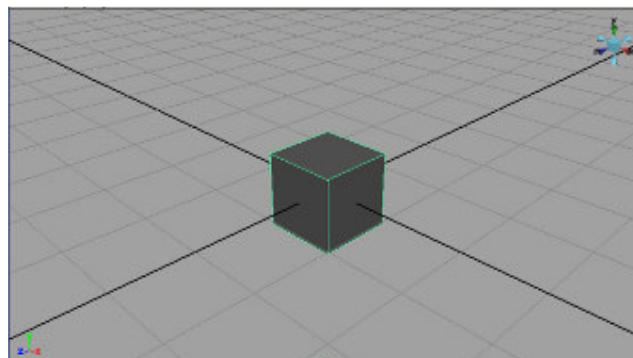


Ouvrez votre éditeur préféré et enregistrez votre travail sous « tp.mel ». Chaque fois que vous voulez lancer un script, il faut d'abord le charger dans le logiciel : « File → Source Script ». Il faut alors donner votre fichier avec l'extension .mel. Ce fichier doit être un fichier texte standard. Lorsque vous modifiez votre fichier, il faut le recharger en mémoire pour que les modifications soient prises en compte. Exemple :

```
global proc test()
{
    polyCube -w 1 -h 1 -d 1 -sx 1 -sy 1 -sz 1 -ax 0 1 0 -tx 1 -ch 1;
}
```

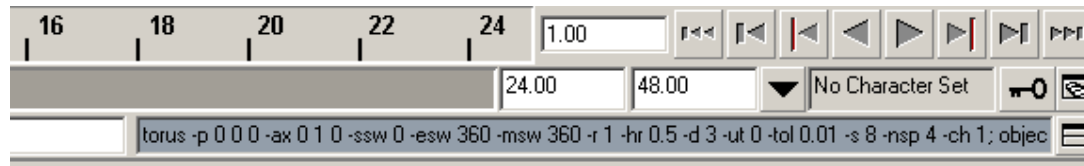


Ensuite tapez « test » + ENTREE dans la ligne de commande pour activer le script :

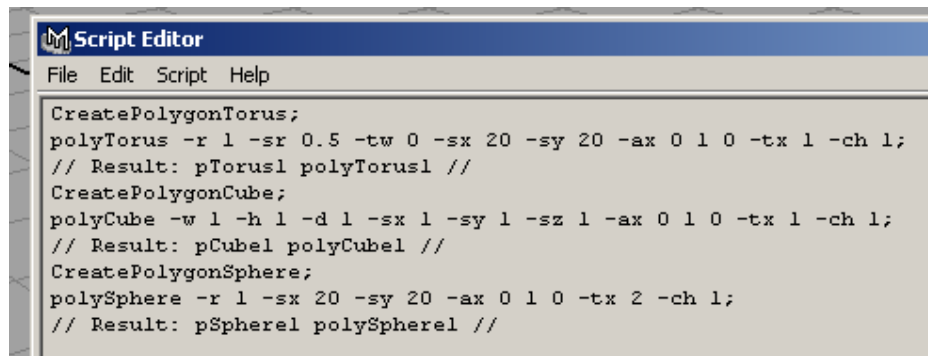


Maya et le MEL

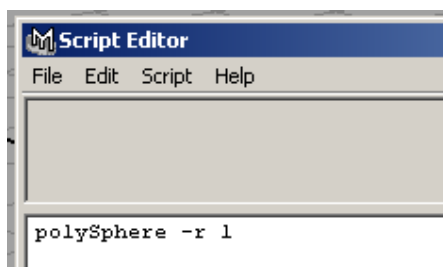
Créez un tore et regardez ce qu'il se passe dans votre fenêtre en bas à droite.



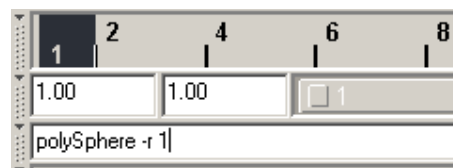
Toute action / opération dans Maya correspond au déclenchement d'une commande MEL. La dernière commande exécutée est affichée en bas à droite de la fenêtre de Maya. Créez un cube et une sphère et regardez le script editor. Vous constatez que l'ensemble des commandes y figurent :



Ainsi si vous savez effectuer une opération sous Maya, il vous suffit de l'effectuer et de regarder le script editor pour connaître les commandes utilisées. Vous connaissez ainsi les commandes MEL dont vous avez besoin. Vous pouvez taper vos commandes MEL en utilisant la fenêtre de texte en bas du script editor ou la command line en bas de la fenêtre Maya :



ou



Attention dans la command line tapez **ENTREE** pour exécuter votre code et **CTRL + ENTREE** dans le script editor (Maya est farceur). Par la suite, pour exécuter votre code, vous allez effectuer un copier coller de votre fichier à l'intérieur du script editor (la command line ne pouvant contenir qu'une ligne). Cela permet de contourner le passage par le menu « open script » à chaque modification de votre fichier.

La documentation

Pour accéder aux aides du MEL, il faut momentanément désactiver le proxy ESIEE.
Pour cela dans internet explorer faites :

- IE : Outils > Options Internet > Connexions > Paramètres réseau / paramètres LAN > décocher //cache.esiee.fr/scripts/proxy.pac
- FireFox : Outils > Options > Avancé > Réseau > Paramètres > Pas de proxy

Nous avons remarqué que beaucoup de paramètres sont apparus dans les lignes générées dans le script editor lors de nos essais. A quoi correspondent ces paramètres et quelles sont les noms des différentes commandes ? Pour cela, utilisons la documentation de Maya en faisant : Help → MEL Command Reference. Dans le script editor, vous trouverez en plus l'aide du langage MEL : Help → Help on MEL. Regardons maintenant les options de la commande PolySphere. Nous trouvons par exemple :

-radius (-r) ou encore -name (-n). Nous pouvons donc maintenant créer une sphère de rayon 5 s'appelant sp1 en tapant :

```
polySphere -r 10 -n sp1;
```

Vérifiez que les attributs correspondent :

Channels	Object
sp1	
Translate X	0
Translate Y	0
Translate Z	0
Rotate X	0
Rotate Y	0
Rotate Z	0
Scale X	1
Scale Y	1
Scale Z	1
Visibility	on
SHAPES	
sp1Shape	
INPUTS	
polySphere1	
Radius	10
Subdivisions Axis	20
Subdivisions Height	20

Le langage MEL

Opérateurs

Affectation : =

Opérateurs d'affectation évolués : +=, -=, /=, *=, ++, --

Variable : \$nom_variable

Types

```
int      $a = 5;
float    $b = 3.456;
vector   $v = <<1.2, 3.4, 6.5>>;
float    $ar[] = {1.2, 3.4, 4.5}; // tableau
matrix   $mtx[3][2];           // une matrice 3x2 de flottant
```

Le type vector est un cas particulier du tableau, il doit être plus efficace à l'usage.
Pas de tableau de tableau sous Maya.

Opérations sur les tableaux & matrices

```
$un_tableau = {"un", "deux"};
```

```

int $taille = size($un_tableau);
vector $B= <<1, 1, 1 >>;
$B.x = 2; il faut utiliser : $B = << 2 , $B.y, $B.z>> ;
matrix $mat[2][2] = << 1, 2; 3, 4>>;
$mat[0][0] = 3;
vector $A,$B;
$v = $A ^ $B ; // produit vectoriel
$a = $A * $B

```

Les structures conditionnelles

Les structures - if / else / switch / while / for - sont les meme qu'en C & JAVA

IF

```
if ( $a == $b) { ... } else if ( $a > $b) {...} else {...}
```

SWITCH

```
switch ($color){ case "blue": ... break; case $c1: ... break; default: ... break; }
```

WHILE

```
while ($a < size($arry)) { ... }
```

DO WHILE

```
do {...} while ($a > 0);
```

FOR

```
for ($i = 10; $i > 0; $i--) { print($i+"...\n"); }
string $arry[3] = {"red","green","blue"};
for ($k in $arry) { print($k + "\n"); };
```

Procédures

Syntaxe : **global proc <return type> <name>(<arg list>) { ... ; return <exp>; }**

Exemple :

```
global proc float squareAndAdd(float $x, float $y) { return $x * $x + $y; }
square(5.0, 2.0);
```

Commentaires

Utilisez les traditionnels // pour une ligne et /* */ pour un bloc.

Modifier les paramètres d'un objet

Utilisez les fonctions suivantes pour un objet s'appelant mySphere :

Ecriture : `setAttr("mySphere.translateX",10);`
Lecture : `$a = getAttr("mySphere.translateX");`

Utiliser des expressions dans les commandes MEL

Veillez à les entourer par des parenthèses :

```
int $a = 7 ;  
int $b = 8;  
polySphere -r ($a+$b);
```

Sélectionner des objets

Pour ajouter un objet à la sélection actuelle :

```
select -add ("nurbsCircle" + $i);
```

Pour tout désélectionner :

```
select -cl
```

Fonctions mathématiques

Les habituelles fonctions sinus/cosinus sont disponibles : `cos()` et `sin()`. Pour donner un peu de variation aléatoire à vos formes, utilisez les fonctions `rand()` et `gauss()` .

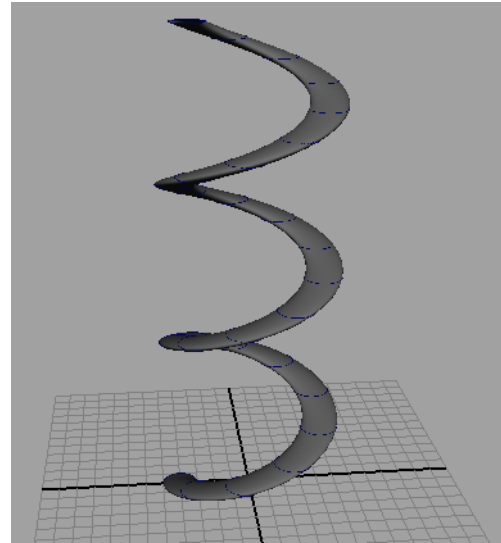
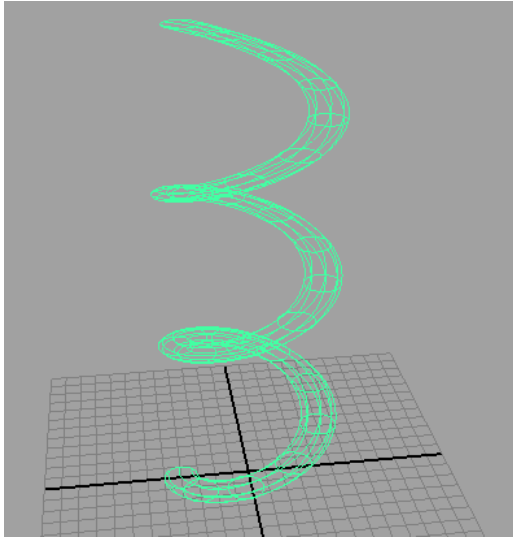
Fonctions avancées du MEL

`eval` permet d'évaluer une expression contenue dans une chaîne de caractères.

```
string $t = "print(\"hello\")";  
eval($t) ;
```

Exercices :

Réalisez un script permettant de réaliser l'objet suivant :



Donnez tout ce que vous avez et réalisez la figure suivante :



Création de [GEORGE KATODRYTIS](#)