

MAXScript – výukový kurz

Díl čtvrtý – jazyk MAXScript, část I.

© Jan Melichar, březen 2008

OBSAH

ÚVOD	4
ZÁKLADNÍ PŘÍKAZY	5
OPERÁTORY	6
PROMĚNNÉ.....	6
POLE.....	7
ZÁVĚREM	9

Upozornění

Tento materiál je určen výhradně pro soukromé výukové účely. Nesmí být v žádném případě použit ve spojení s komerční činností, prodáván, šířen v pozměněné podobě či jinak modifikován.

Úvod

Vážení čtenáři,

vítám vás u čtvrtého pokračování seriálu o jazyce a nástroji MAXScript, tentokrát nazvaného **Jazyk MAXScript, část první**.

Dnes se podíváme již na samotný jazyk MAXScript (dále jen MXS). Blíže se budeme zabývat základními příkazy MXS, operátory, proměnnými a naši pozornost soustředíme také na pole (arrays).

Základní příkazy

Pro začátek práce s MXS je vhodné znát několik základních příkazů; na valnou většinou z nich při psaní vlastních skriptů určitě narazíte. Zde jsou:

- **příkaz pro vyčištění okna Listener window:**

```
clearlistener()
```

- **příkaz pro výběr všech objektů ve scéně:**

```
max select all
```

- **příkaz pro odznačení všech objektů (počet vybraných objektů: 0):**

```
clearselection()
```

- **příkaz pro odstranění vybraných objektů:**

```
max delete
```

- **symbol dolaru (\$) slouží k odkázání na určitý objekt:**

```
$nějaký_objekt
```

- **příkaz pro vložení jiného - skriptu (table.ms):**

```
include "table.ms"
```

- **příkaz pro výpis řetězce uzavřeného do uvozovek:**

```
print "Hello world!"
```

- **jednořádkový komentář - za zdvojenou pomlčkou:**

```
if ( chboxKeepRatio.state == true ) then ...
```

```
-- text - komentář, který není součástí vlastního kódu
```

- **více řádkový komentář - uzavřený mezi znaky '/*' a '*/' - používá se např. pro zápis úvodních - autorských informací o příslušném skriptu na začátku kódu:**

```
/*  
Script for creating a table  
Author: autor  
Email: email  
Datum: August 11, 2005  
*/
```

- **Příkaz pro výpis libovolného textu (v uvozovkách) do okna:**

```
messagebox "Text, který se objeví v nově otevřeném okně"
```

- **příkaz pro otevření souboru:**

```
max file open
```

- **příkaz pro zobrazení (znovu zviditelnění) všech dosud schovaných (hidden) objektů ve scéně:**

```
max unhide all
```

- příkaz pro vyrenderování aktuální scény:

```
max quick render
```

Operátory

Základní matematické operátory

Syntaxe jazyka MXS se nijak neliší od syntaxe jiných jazyků. I zde se tedy můžeme setkat (a jistě se setkáme :) se základními matematickými operátory pro sčítání ('+'), odečítání ('-'), násobení ('*'), dělení ('/'). Přitom, operátory pro násobení a dělení mají větší prioritu než operátory pro sčítání a odečítání s tím, že patřičným uzávorkováním můžeme prioritu změnit. Za zmínku možná stojí, že MXS Listener může krom jiného sloužit také jako pomocná kalkulačka :).

Dalšími základními matematickými operátory jsou tyto: operátor pro porovnání proměnných ('=='), nerovnítko ('!='), větší ('>'), menší ('<'), větší a rovno ('>='), menší a rovno ('<=').

Proměnné

Základní typy proměnných

Mezi základní typy proměnných lze v MXS zařadit následující:

- **Float**
- **Integer**
- **String**
- **Time**
- **Double (nově v 3ds Max 9)**
- **Integer64 (nově v 3ds Max 9)**
- **IntegerPtr (nově v 3ds Max 9)**

Přiřazování proměnných

Základní práce s přiřazováním proměnných není v MXS nijak složitá. Podobně jako v jiných programovacích jazycích je hodnota napravo od rovnítka přiřazována do názvu proměnné nalevo od rovnítka. Tedy např.:

```
MujPrvniRezetec = "Hello World"
```

Obsah proměnné **MujPrvniRezetec** pak můžeme libovolně vypsát:

```
MujPrvniRezetec  
/* výsledek: Hello World */
```

Podobně můžeme proměnné využívat pro přiřazování číselných hodnot. Např.:

```
c = sqrt(9)^3
```

Důležité: Nachází-li se na řádku pouze jeden příkaz, není nutno za něj psát středník. Při zápisu více příkazů na jeden řádek je však nutné každý takovýto příkaz ukončit středníkem!

Pozn: Mezi proměnné MXS patří také tzv. **globální** a **lokální proměnné**. Ty si však vzhledem ke svému většímu rozsahu a návaznosti na pokročilejší konstrukty, jako jsou např. metody či třídy, zaslouží své místo v příštím, tj. pátém díle tohoto seriálu.

Pole

Pole je speciální uspořádaná datová struktura, v níž každý její prvek může být jiného typu a nabývat určité hodnoty. Ke každému takovému prvku v poli přistupujeme pomocí indexu, resp. pořadí prvku v tomto poli. Např.

Deklarace pole

```
/* deklarace pole s hodnotami 10, 20 a 30: */
#(10, 20, 30)

/* deklarace pole s hodnotami 10, 20, 30 a jeho přiřazení do proměnné
"pole": */
pole = #(10, 20, 30)

/* chceme vypsat druhý prvek v poli: */
pole[2]
/* výsledek: 20 */
```

Budeme-li chtít určitý prvek v poli změnit, nahradíme jeho hodnotu opět přes index, a to následovně:

```
/* přepsání třetího prvku pole na 120: */
pole[3] = 120
/* výpis pole: */
pole
/* výsledek: #(10, 20, 120) */
```

Jednotlivé prvky pole mohou tedy být různého typu. Pole můžeme tedy deklarovat např. takto:

```
x = 180;
pole2 = #("Hodnota cos x je", cos x)
/* výsledek: #("Hodnota", "cos x je", -1) */
```

Jistě jste si všimli, že jsme v deklaraci pole použili proměnnou x. Té je třeba ještě před vlastní deklarací pole přiřadit určitou hodnotu.

Operace s prvky pole

Do deklarovaného pole lze přidávat další prvky. Lze to provést několika příkazy.

Jedním z nich je příkaz **insertItem**.

Mějme tedy pole deklarované takto:

```
pole3 = #(10, 20, 30)
insertItem 40 pole3 4
insertItem 50 pole3 2
```

Příkazem **insertItem** jsme vložili hodnotu **40** do pole s názvem **pole3** na pozici **4** a hodnotu **50** na pozici **2**.

Výsledkem tedy bude:

```
pole3
/* výsledek: #(10, 50, 20, 30, 40) */
```

Budeme precizní a řekneme si, že se nám pořadí prvků příliš nevyhovuje. Co s tím? Prvky setřídíme. K tomuto účelu použijeme příkaz **sort** "pole". Zápis tedy bude následující:

```
sort pole3
/* výsledek: #(10, 20, 30, 40, 50) */
```

Dalšími příkazy pro vložení prvku do pole jsou např.: **join** nebo **appendUnique**, což je nový příkaz v 3ds Max 2008. Rozdíl mezi příkazem **insertItem** a **appendUnique** spočívá (jak již sám příkaz napovídá) v kontrole, zda se prvky v poli neopakují. Zatímco příkazem **insertItem** můžeme do pole přidat prvek, který se v poli již vyskytuje, při použití příkazu **appendUnique** bude vrácena hodnota *false* a další prvek do pole přidán nebude.

Duplicitní prvky pole lze odstranit pomocí příkazu **makeUniqueArray** "pole". Jakýkoli prvek lze z pole odstranit pomocí příkazu **deleteItem** "pole" "pozice". Tedy například odstraníme hodnotu, jejíž index se nachází na pozici 3. Odstraněna tedy bude hodnota 30.

```
deleteItem pole3 3
/* výsledek: #(10, 20, 40, 50) */
```

Vícerozměrná pole

Vícerozměrná pole se od jednorozměrných liší v tom, že ke stávající "řadě" řadě prvků (v tomto případě prvky (1, 2, 3)) je přidána nová "řada", a taktéž v přidávaném indexu, který je při výpisu určitých prvků pole vložen do nových hranatých závorek. Deklarace takového pole může vypadat následovně (nově vložený rozměr-řada sestává z prvků (10, 20, 30)):

```
pole4 = #( #(1, 2, 3), #(10, 20, 30) )
pole4[2][1]
/* výsledek: 10; byla vrácena hodnota prvního prvku (indexu) z druhého rozměru pole */
```

Závěrem

Příští díl tohoto kurzu bude zaměřený na základní větvící příkazy, cykly, funkce a třídy a objekty, a taktéž již zmíněné globální a lokální proměnné.

Literatura: Autodesk 3ds Max 2008 MAXScript Reference