

Max Script for 3D Studio MAX R4 CZ. II

by Adrian Majchrzak

Witam wszystkich w drugiej części mojego arta na temat Max Script. W pierwszej części napisaliśmy prosty skrypt którym kontrolowaliśmy zachowanie się kulki w naszej scenie. Dzisiaj poznamy trochę więcej szczegółów dotyczących skryptów więc zaczynamy.

Zmienne

Na początku powiemy sobie o zmiennych. Zmienne w każdym języku programowania reprezentują najczęściej jakiś przedział liczbowy, jakiś ciąg znaków lub coś jeszcze. Tu mamy trochę inaczej to przedstawione a mianowicie zmienna w maxscript może być dowolnego typu. Nie ma tu czegoś takiego jak zmienna int czy float albo string. Jeśli chcemy przypisać sobie do zmiennej jakiś napis to wymyślamy sobie nazwę zmiennej i przypisujemy do niej wartość za pomocą znaku równości np.

`napis = "to jest jakiś ciąg znaków przypisany zmiennej napis."`

Uwaga

Jeżeli tworzysz ciąg znaków to pamiętaj że zanim zamkniesz cudzysłów musisz postawić na końcu napisu kropkę

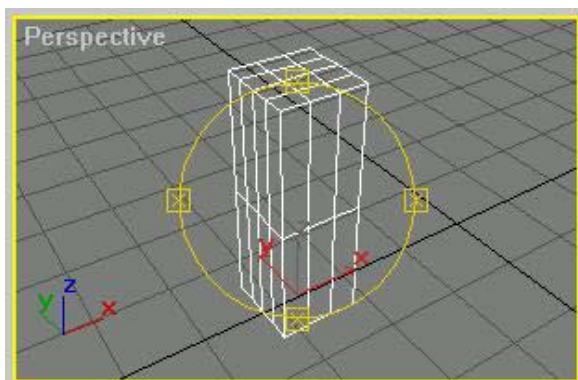
Jeżeli chcemy zrobić sobie zmienne dotyczące np. wysokości to piszemy

```
max_wys = 10
```

tak więc nie musimy się przejmować typami zmiennych. Tu zmiennymi są nazwy które najłatwiej jest nam zapamiętać i używać w programie.

Przykład 1

Utwórz sobie w dowolnym miejscu obiekty np. Box o dowolnej wielkości, segmentach itp. Teraz napiszemy skrypt który ustawi takie parametry i położenie jakie sobie życzymy. Wiem że to można zrobić na samym początku modelowania ale tu chodzi o pokazanie działania zmiennych oraz zasadę pracy z obiektami w scenie w oparciu o ich właściwości.



Tak wygląda nasz obiekt Box01 po opisaniu go skryptem

Skrypt kontrolujący nasz obiekt Box01

```
-- parametry obiektu Box01
dlug = 10
szer = 10
wys = 30
leng_segs=4
wid_segs=3
heig_segs=2

-- przypisywanie parametrów do obiektu Box01
$Box01.Length=dlug
$Box01.Width=szer
$Box01.Height=wys
$Box01.LengthSegs=leng_segs
$Box01.WidthSegs=wid_segs
$Box01.HeightSegs=heig_segs
```

Jak widać skrypt jest banalny ale teraz już wiesz jak się odwołać do właściwości każdego obiektu. Wygodnie stworzyć sobie zmienne które będą przechowywały parametry obiektu a potem je przypisać do obiektu. Łatwiej jest go potem kontrolować. Zauważyłeś też pewnie coś nowego w kodzie. Dołożyłem linijkę komentarza. Znacznie ułatwia programowanie. Komentarze rozpoczynamy od dwóch myślników a potem możemy pisać co chcemy.

Instrukcja warunkowa if .. then .. else

W dalszej części będziemy posługiwali się dalej naszym pudełkiem. Dla pewności że wszystko będzie dobrze, zresetuj scenę i utwórz nowy obiekt Box01 o dowolnych parametrach ale nie ruszaj ilości segmentów, niech każdy segment ma wartość 1. Od razu podam kod skryptu a potem omówię jego działanie.

Przykład skryptu opisującego zasadę działania instrukcji if

```
-- parametry obiektu Box01
l=2

-- dla warunku if
seg1 = 2
seg2 = 2
seg3 = 4
seg4 = 4

-- przypisywanie parametrów do obiektu Box01
$Box01.LengthSegs=1

-- warunek if
if l < 5 then
  (
    $Box01.WidthSegs=seg1
    $Box01.HeightSegs=seg2
  )
else
  (
    $Box01.WidthSegs=seg3
    $Box01.HeightSegs=seg4
  )
```

W pierwszej linijce nic nowego nie ma, sam komentarz. Dalej mamy zmienną do której przypisaliśmy wartość 2. Zmienna ta będzie kontrolować zachowanie się warunku if i będzie wpływała na to ile segmentów będzie miał nasz Box01.

Dalej mamy znowu komentarz i 4 kolejne zmienne które będą wykorzystane w instrukcji warunkowej. Mają one za zadanie przechowywać wartości ilości segmentów. Dalej mamy znowu komentarz a za nim z góry przypisana wartość 1 dla długości segmentów naszego pudełka. Czas na opis naszej instrukcji warunkowej

Postać instrukcji warunkowej if

```
if <warunek> then
(
< jakieś instrukcje>
)
else
(
< jakieś instrukcje >
)
```

Warunkiem tu jest nasza zmienna „l” którą deklarowaliśmy na samym początku. Dokonujemy porównania. Jeżeli **l < 5** wtedy wykonuje się część instrukcji po słowie **then** jeżeli warunek jest fałszywy czyli zmienna **l** wynosi **6** wtedy wykonywane są instrukcje po słowie **else**. I to jest całe działanie tej funkcji. Można też stosować funkcję bez słowa **else**, wtedy nie są potrzebne nawiasy. Poniżej w tabelce zamieszczam wszystkie możliwe operatory porównań które są dozwolone dla warunku if.

Tablica funkcji których możesz używać w warunku if	
==	<i>Porównanie</i>
!=	<i>Różne od</i>
<	<i>Większe od</i>
>	<i>Mniejsze od</i>
>=	<i>Mniejsze lub równe</i>
<=	<i>Większe lub równe</i>

O instrukcji if jeszcze będziemy mówić ponieważ jej funkcje warunkowe są znacznie szersze ale o tym innym razem. Jeżeli podobają Ci się te lekcje zapraszam za jakiś czas, niedługo będzie kolejna część.

Autor :	Adrian Majchrzak
Przeznaczenie :	Strona www.max3d.pl