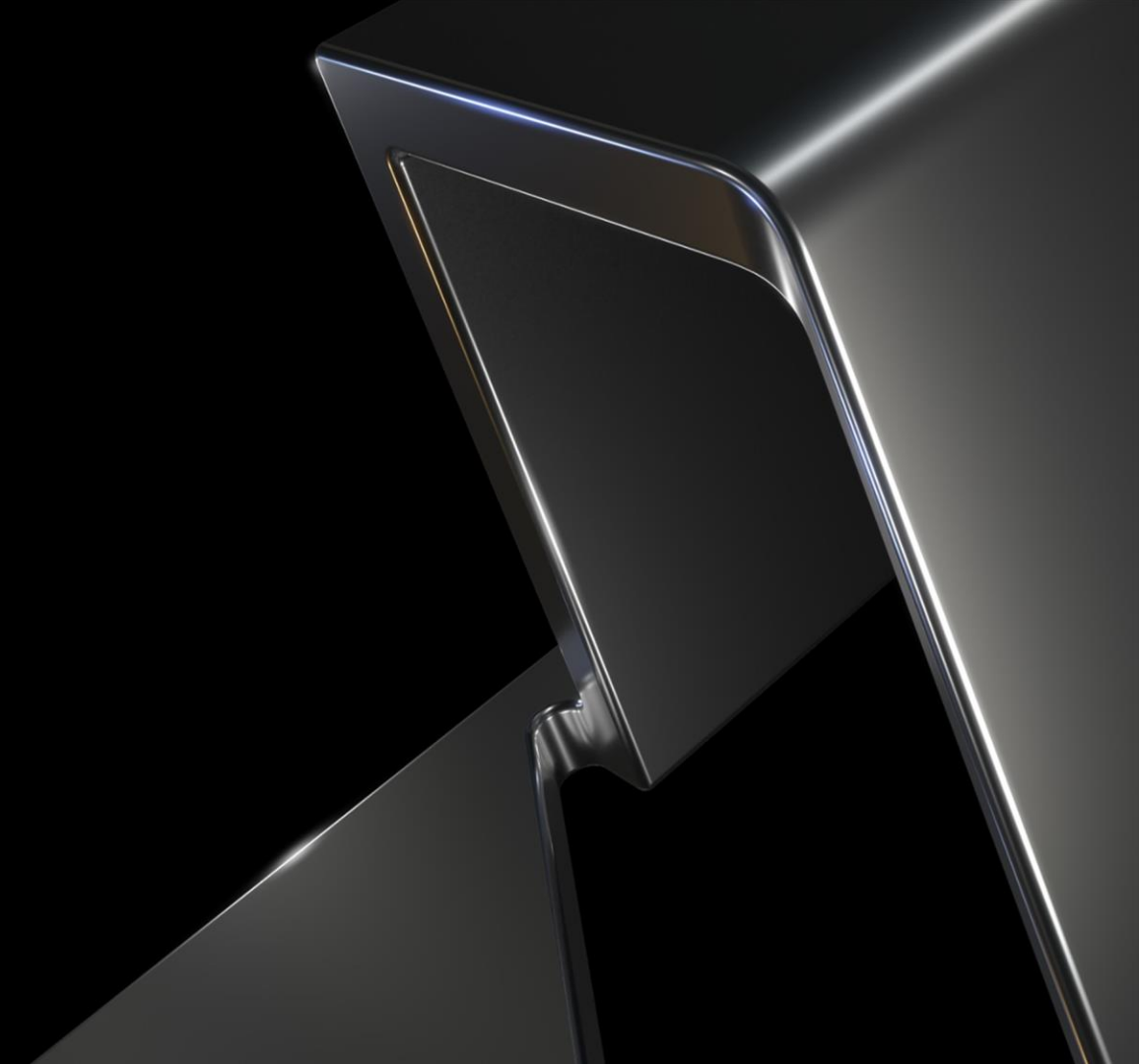
AUTODESK UNIVERSITY

MaxScript

ShovenZhang（章文涵）



前言

前言

- 课程说明
  - 本次课程涵盖的脚本知识和专业的脚本书籍相比，在许多方面都比较匮乏，如果有需要进一步了解的部分，我的建议是通过官方文档或者是查看一些专业的书籍

- 课程目的
  - 希望能够通过我个人学习经验的总结分享、以及一些案例讲解，让没有接触过max脚本的同学能够快速入门，并且能够独立编写工具，以及在之后学习脚本、编写脚本的过程中，提供一些开发的思路

学习脚本能够带来什么？

- 如果只对角色模型设置一些简单的绑定，那么max脚本和脚本控制器不一定是必须要了解的
- 如果是纯美术制作的岗位，如果没有代码基础，也能够学习脚本，美术不要惧怕写代码。通过脚本简化重复性的工作、创建一些操作的快捷方式等等。。能够提高非常大的工作效率，所以说，学习脚本是一件非常有意义的事情。
- 如果是T A或者是专业的绑定师，则至少需要了解脚本，比如拿到一个脚本文件，至少能够去分析实现逻辑，有能力去修改它
- DCC脚本的学习，前提是要会用软件，假设目前使用的是Maya，那么Mel和Python是必须掌握的；如果用的是3DSMax，那么Max脚本是必不可缺的技能之一。不管是3DSMax还是Maya，每个软件对应的脚本语言，只是不同的语法，开发的时候都是同一个套路。学习脚本除了掌握基础的语法之外，就是要靠完成需求之后经验的积累。总结一句话就是“实践大于理论”。

成为一个优秀的动画TA、或者想要转岗成为动画TA，需要具备什么技能？

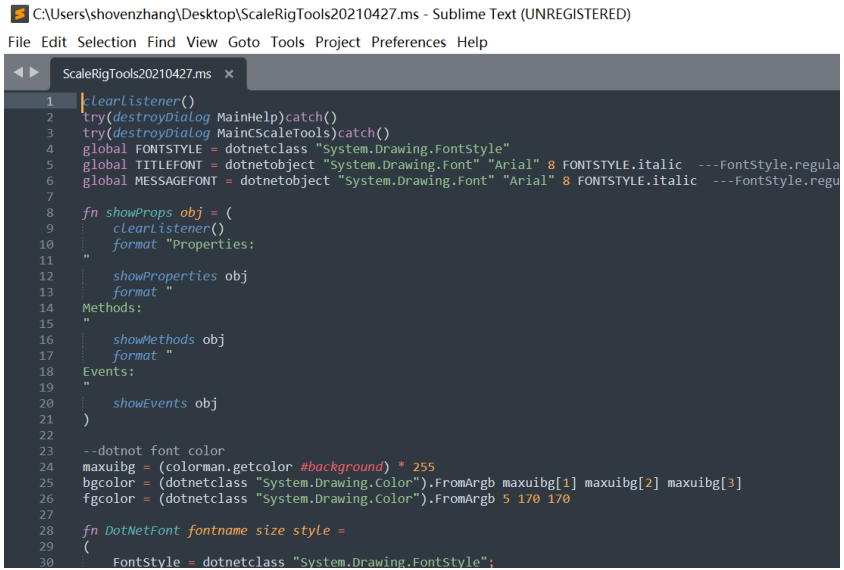
- 首先是代码基础，至少一门高级语言的编程经验
- 其次是原型开发能力，比如能够独立制作一个FPS游戏demo、编写一个角色绑定插件、一个模型制作插件等等
- 创造力和独立分析问题的能力也是不可缺少的
- 需要掌握的技能包括但不限于帮助制作解决问题、研究解决技术难题、维护动画制作管线等等
- Key过动画，且了解制作流程;动画不需要做的很好，但这是加分项。

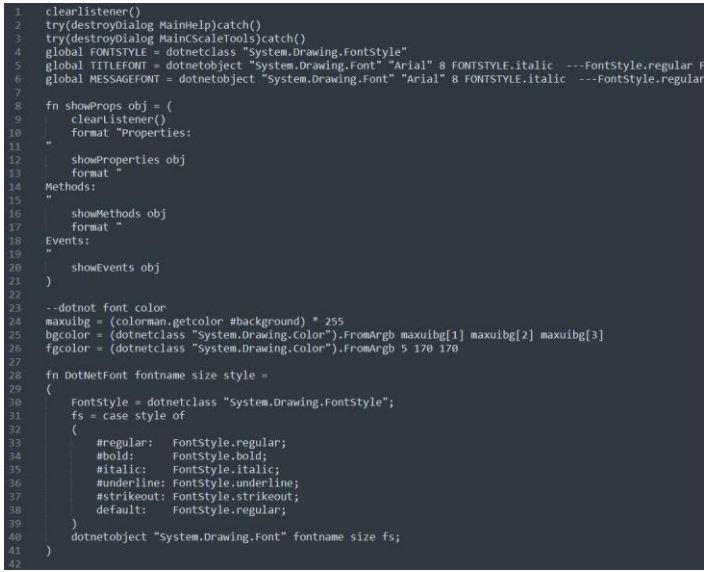
开发工具

Sublime Text

SubLime Text

- 下载地址
  - https://www.sublimetext.com/
- Maxscript语法高亮支持插件及安装教程
  - https://github.com/cb109/sublime3dsmax

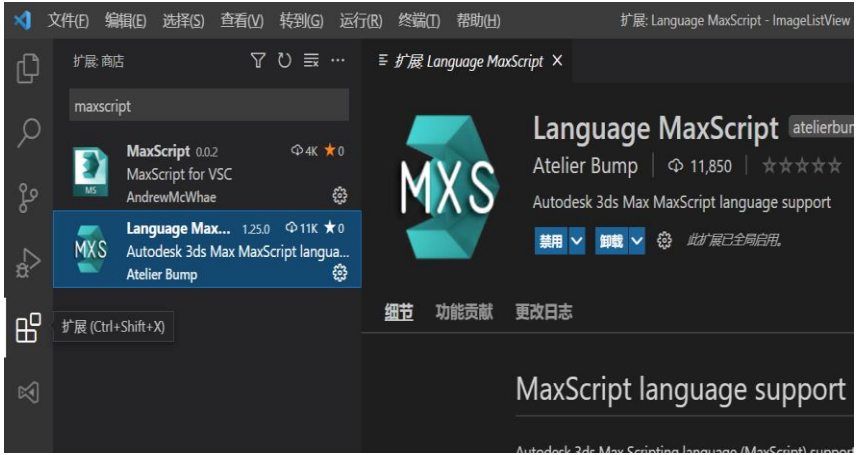


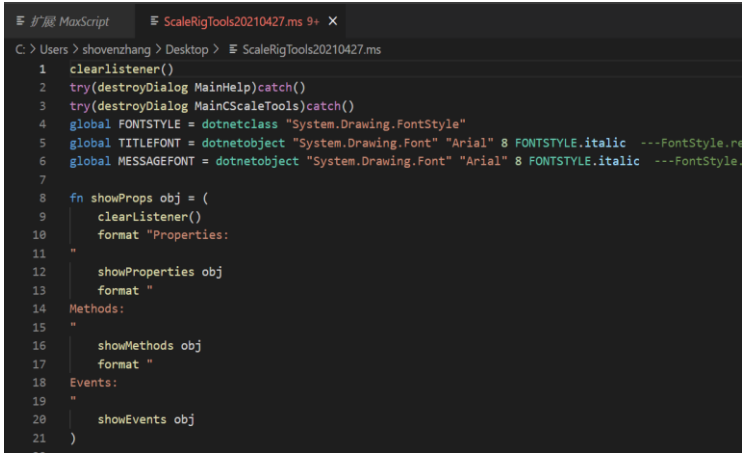


Visual Studio Code

Visual Studio Code

- 下载地址
  - https://code.visualstudio.com/
- Maxscript语法高亮支持插件
  - 使用快捷键【Ctrl+Shift+X 】进入【扩展:商店】，搜索关键字【maxscript】



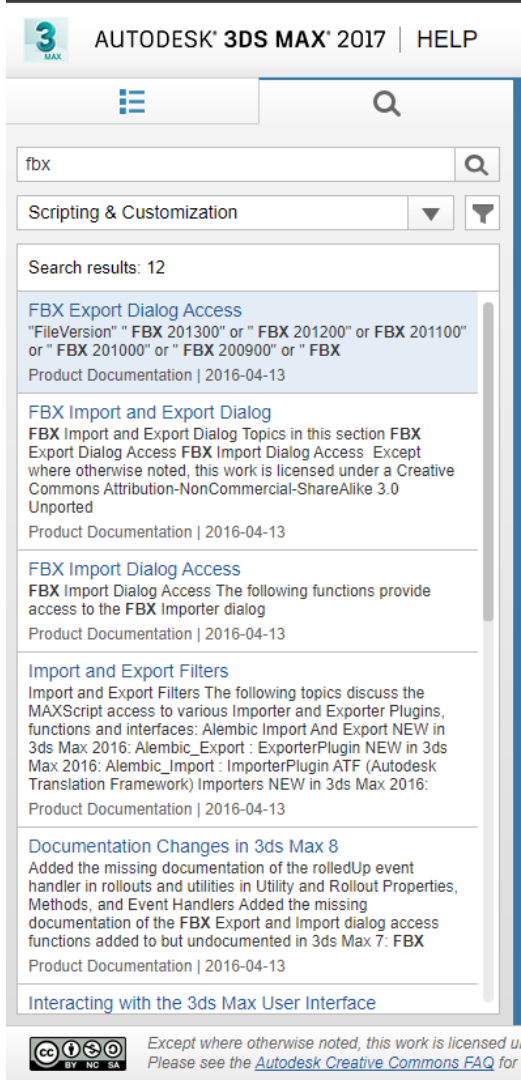


学习途径

文档书籍

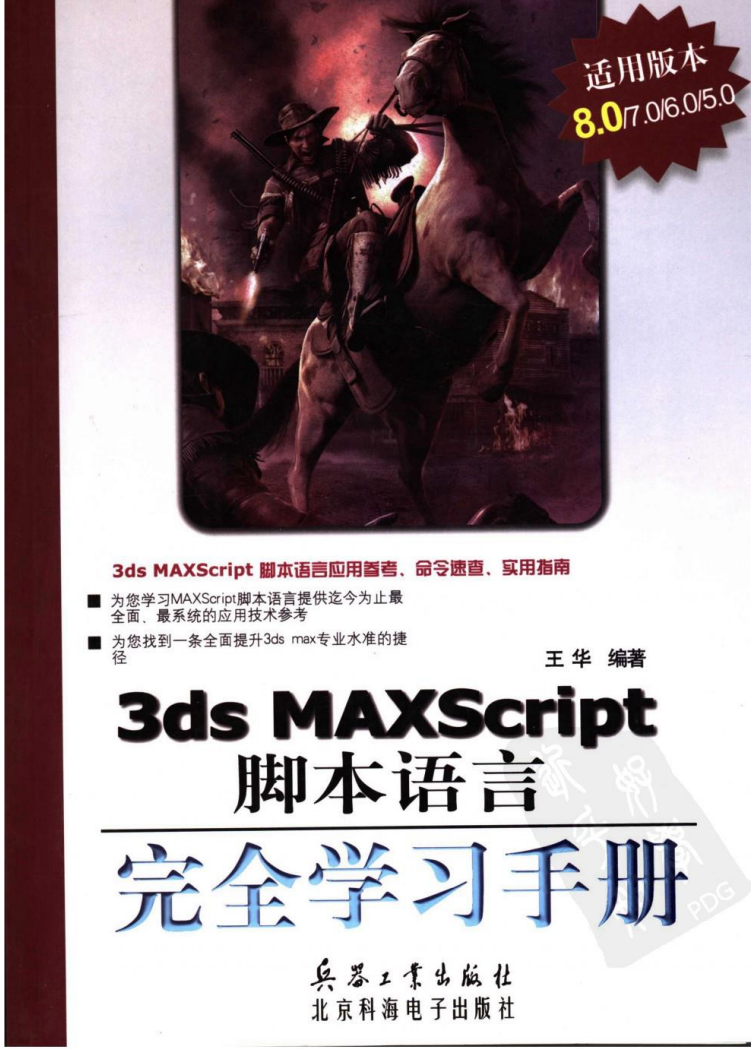
文档书籍

- 官方帮助文档
  - 快捷键“F1”
  - https://help.autodesk.com/view/3DSMAX/2017/ENU/
  - 选择对应的类别，搜索框内输入需要的关键字



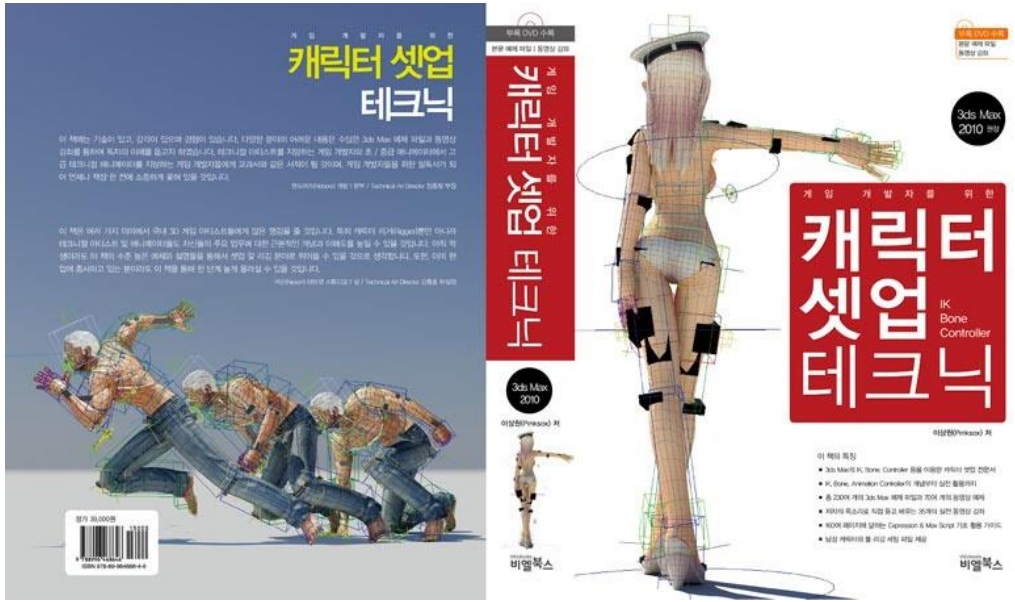
文档书籍

- 3ds MaxScript脚本语言完全学习手册
  - 类似官方帮助文档



文档书籍

- 角色设定技术 / Character Setup Techniques
  - 非常专业的3dsMax绑定书籍，能够深入了解脚本控制器、约束系统等使用
  - 实用，案例讲解居多，适合有一定软件基础、脚本基础



其他学习途径

其他学习途径

- 学习其他脚本的写法
  - “.ms”、“.mcr”，直接打开
  - “.MZP”，改后缀名，MZP改为ZIP，然后通过压缩软件进行解压
- MaxScript Listener /max脚本侦听器
  - 在3dsMax里执行一些操作时，侦听器会记录百分之七十以上的代码
- 通过内置函数查询相应语句
  - show <object>、showProperties <object>、showMethods <object>、showEvents <object>
- 搜索引擎
  - Google搜索，关键字：maxscript+“对应关键字”
  - 百度搜索，中文资料相对较少
- 号称全球最大的MAXScript脚本网站：scriptspot
  - http://www.scriptspot.com/

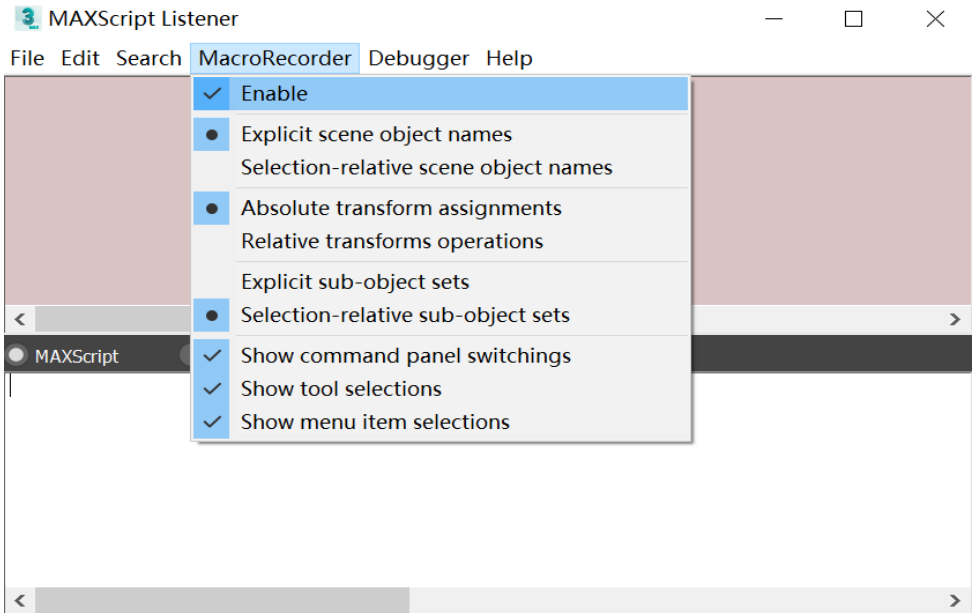
脚本基础

脚本侦听器
（MAXScript Listener）

脚本侦听器

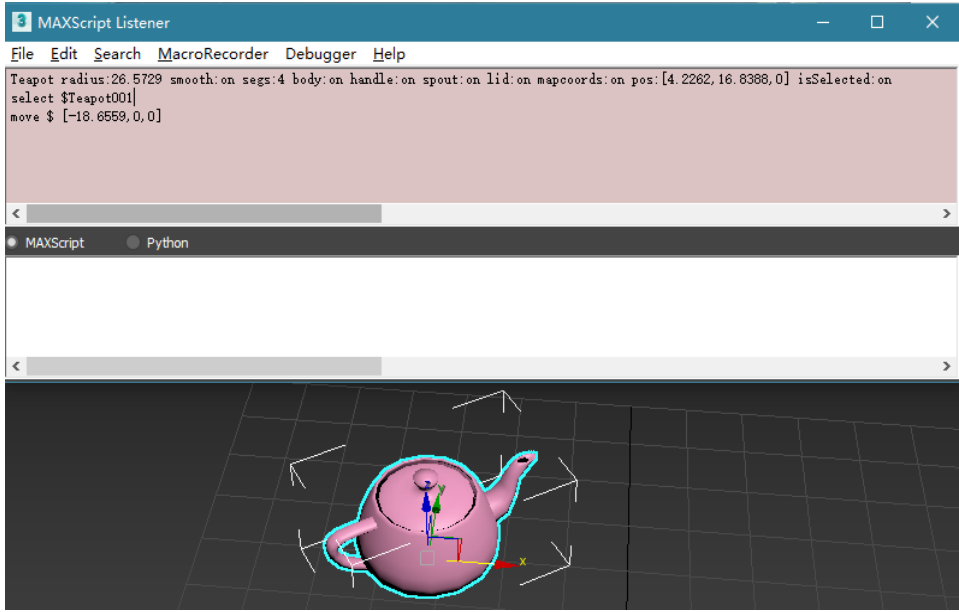
- 常用打开方式：快捷键F11、菜单
  Scripting → MAXScript Listener

- 将Macro Recorder（宏记录器）菜单中的Enable选项激活（默认为关闭）



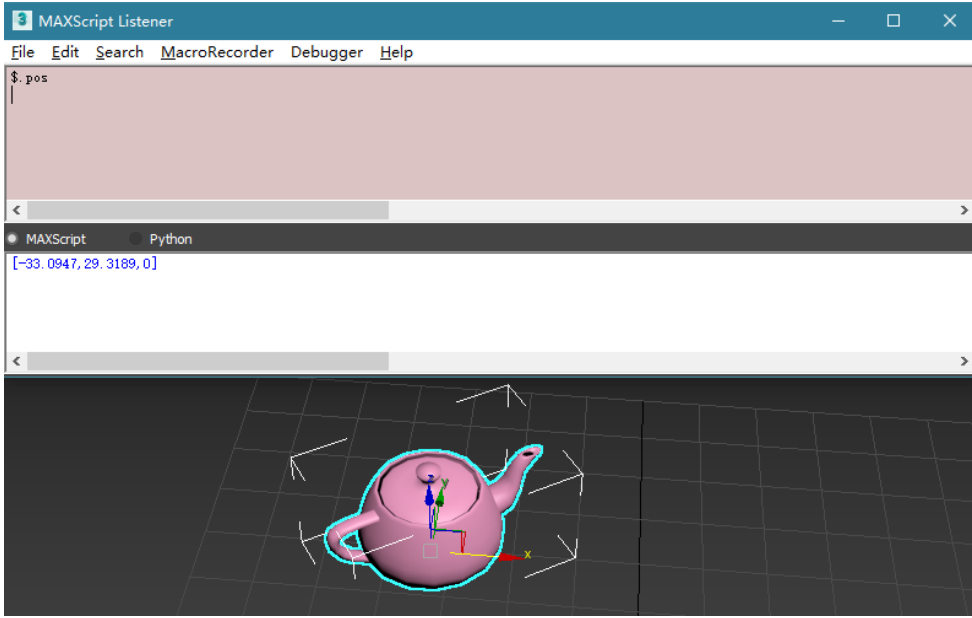
脚本侦听器

- 举例：创建一个茶壶，然后将其沿着X轴移动



脚本侦听器

- 举例：执行一个简单的脚本并返回值
- 小技巧：快捷键Ctrl+D，清除窗体内代码

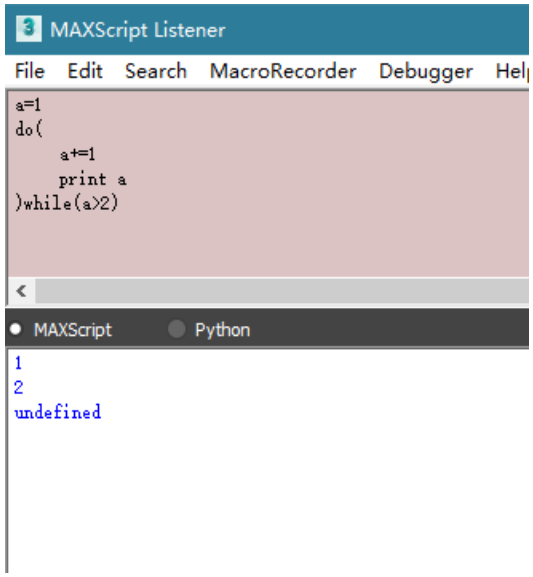


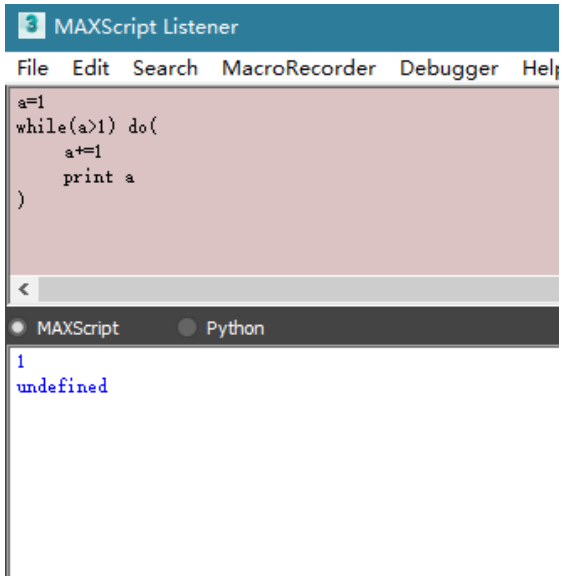
Do While循环

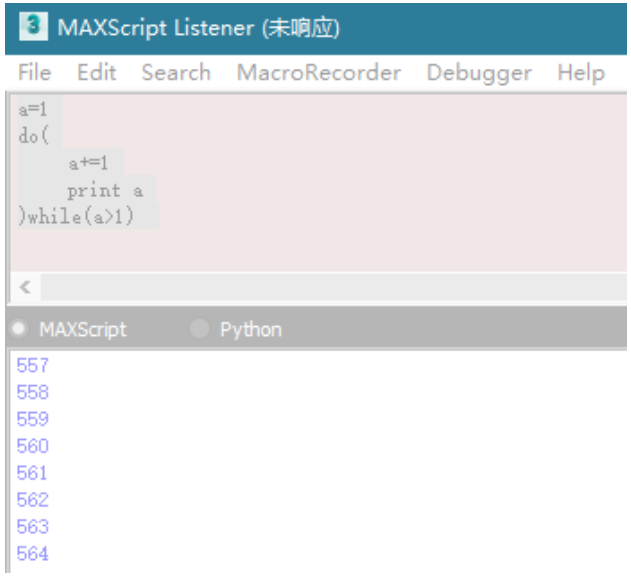
Do While 循环

```
do( <代码> )while( <条件> )
--先执行代码,然后判断条件是否成立。如果条件一直成立，就会一直执行代码。若条件一开始就不成立，代码也会被执行一次。

while( <条件> )do( <代码> )
--先判断条件是否成立，如果成立，再执行代码。如果条件一直成立，就会一直执行代码。若条件一开始就不成立，则不会执行代码。
```







For 循环

For循环

```
--语法
    for <var_name> = <sequence> do(
        <expression>...
    )

--示例
    for i = 1 to 8 do(
        move $ [i,0,0]
        print i
    )
```

```
1       move $ [1,0,0] ; print 1 ;
2       move $ [2,0,0] ; print 2 ;
3       move $ [3,0,0] ; print 3 ;
4       move $ [4,0,0] ; print 4 ;
5       move $ [5,0,0] ; print 5 ;
6       move $ [6,0,0] ; print 6 ;
7       move $ [7,0,0] ; print 7 ;
8       move $ [8,0,0] ; print 8 ;
```

```
for i = 1 to 8 do (
    move $ [i,0,0]
    print i

)
```

MAXScript Python

```
1
2
3
4
5
6
7
8
OK
```

IF 语句

IF 语句

```
--语法
    if <条件> then (
        <代码>
    )

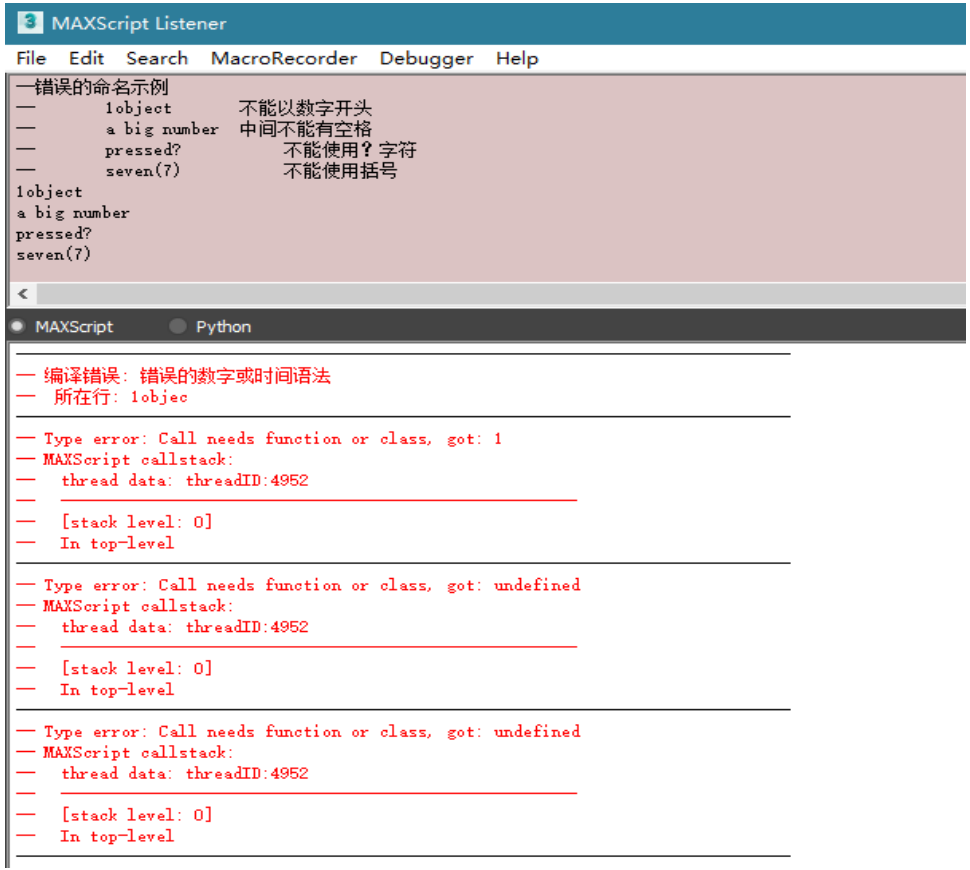
--示例
    for i = 1 to 8 do(
        if (mod i 2 == 0) then(
            move $ [i*2,0,0]
        )else(
            move $ [i,0,0]
        )
    )
```

```
1    move $ [1,0,0]
2    move $ [4,0,0]
3    move $ [3,0,0]
4    move $ [8,0,0]
5    move $ [5,0,0]
6    move $ [12,0,0]
7    move $ [7,0,0]
8    move $ [16,0,0]
```

```
1        move $ [1,0,0] ; print 1 ;
2        move $ [2,0,0] ; print 2 ;
3        move $ [3,0,0] ; print 3 ;
4        move $ [4,0,0] ; print 4 ;
5        move $ [5,0,0] ; print 5 ;
6        move $ [6,0,0] ; print 6 ;
7        move $ [7,0,0] ; print 7 ;
8        move $ [8,0,0] ; print 8 ;
```

变量

变量


MAXScript Listener
File Edit Search MacroRecorder Debugger Help
--错误的命名示例
--		1object		不能以数字开头
--		a big number	中间不能有空格
--		pressed?		不能使用？字符
--		seven(7)		不能使用括号
1object
a big number
pressed?
seven(7)
MAXScript
Python
-- 编译错误：错误的数字或时间语法
-- 所在行：1objec
-- Type error: Call needs function or class, got: 1
-- MAXScript callstack:
-- thread data: threadID:4952
-- [stack level: 0]
-- In top-level
-- Type error: Call needs function or class, got: undefined
-- MAXScript callstack:
-- thread data: threadID:4952
-- [stack level: 0]
-- In top-level
-- Type error: Call needs function or class, got: undefined
-- MAXScript callstack:
-- thread data: threadID:4952
-- [stack level: 0]
-- In top-level

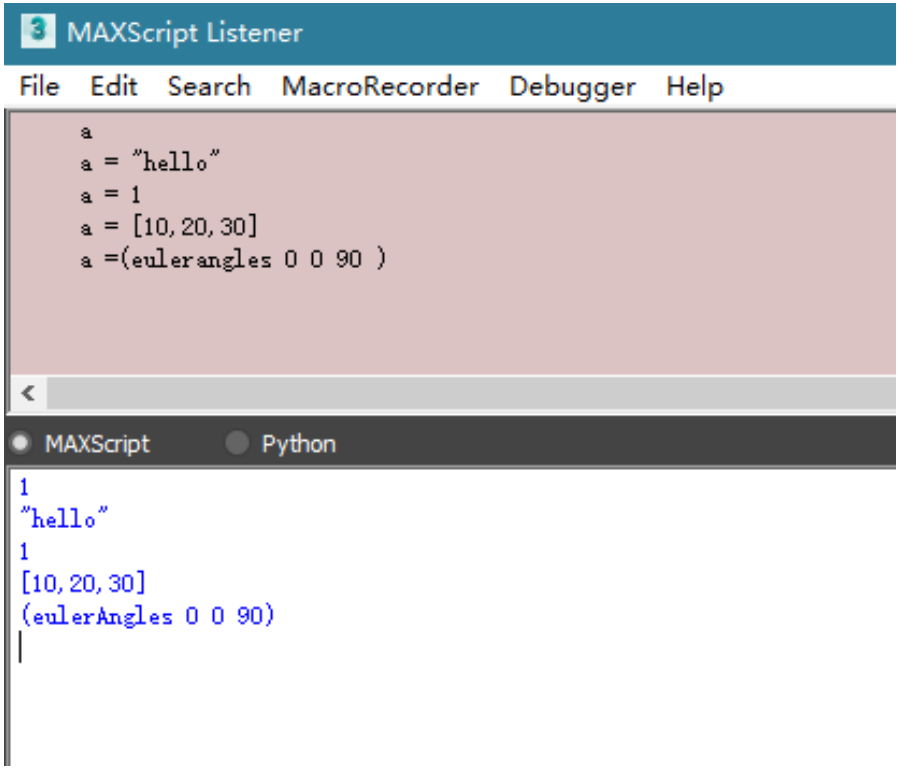
变量

常用数据类型

- Integer 整形、整数
  - a = 1
  - b = 2
  - c = a + b
  - c = 3
- float 浮点型、小数点
  - a = 1.1
  - b = 2.0
  - c = a + b
  - c = 3.1
- String 字符串
  - a = "max"
  - b = "脚本"
  - c = a +b
  - c = "max脚本 “
- Array 数组
  - a=#(1,2,"hello",8)
  - a[1]的值为1
  - a[2]的值为2
  - a[3]的值为"hello"
  - a[4]的值为8

AS 类型转换

AS 类型转换


MAXScript Listener
File Edit Search MacroRecorder Debugger Help
a
a = "hello"
a = 1
a = [10,20,30]
a =(eulerangles 0 0 90)
MAXScript
Python
1
"hello"
1
[10,20,30]
(eulerAngles 0 0 90)

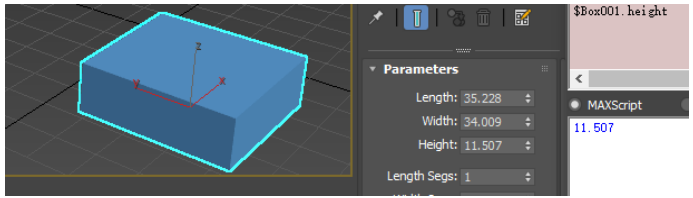
常用操作符、内置函数、内置变量

常用操作符、内置函数、内置变量

- 平方根
  - ^
  - ```
    print 2^3
    ```
    MAXScript
    ```
    8
    ```
- +=　、　-=
  - ```
    a=1
    a+=2
    ```
    MAXScript
    ```
    1
    3
    ```

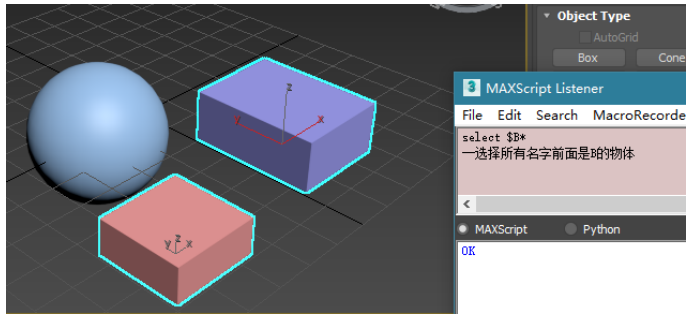
- == 、 > 、 < 、 >= 、 <=
  - ```
    a=1
    b=2
    if (a>=b) then (print "a>=b") else (print "a<b")
    ```
    MAXScript　Python
    ```
    1
    2
    "a<b"
    "a<b"
    ```
    ```

常用操作符、内置函数、内置变量

- = 赋值
  - ```
    x = 1
    s = "x="
    s += x as string
    ```
    MAXScript P
    ```
    1
    "x="
    "x=1"
    ```
- $ 物体选择器
  - 

- random 随机函数
  - ```
    random 1 100        —生成1-100的随机整数
    random 1 100.0      —生成1-100的随机整数
    random 1.0 100      —生成1-100的随机小数
    ```
    MAXScript Python
    ```
    43
    78
    53.5126
    ```
- mod
  - ```
    mod 10 3
    mod 10 2
    ```
    MAXScript
    ```
    1.0
    0.0
    ```
    ```

常用操作符、内置函数、内置变量

- * 通配符
  - 

- ‘ ’ 单引号，变量包括器
  - ```
    一作用1：包裹后可以使用任意符号作为变量名，比如空格开头，数字开头
    一作用2：选择不规则名字的物体

    select $'Bip001 L Thigh'
    '一个茶壶' = "yi ge cha hu"
    select $Bip001 L Thigh
    ```

    MAXScript Python

    ```
    OK
    "yi ge cha hu"
    -- Argument count error: select wanted 1, got 3
    -- MAXScript callstack:
    --   thread data: threadID:4952
    --   ------------------------------------------------
    --   [stack level: 0]
    --   In top-level
    ```
    ```

消息提示框
（MessageBox）

消息提示框（MessageBox）

```
messageBox <message_string> [ title : <window_title_string> ] [ beep : <boolean> ]
```

显示包含消息字符串和确定按钮的模式消息框。消息框窗口标题可以设置为 `title:` 关键字参数。您可以控制是否发出蜂鸣声 `beep:` 关键字参数，默认为 `true` .

```
queryBox <message_string> [ title : <window_title_string> ] [ beep : <boolean> ]
```

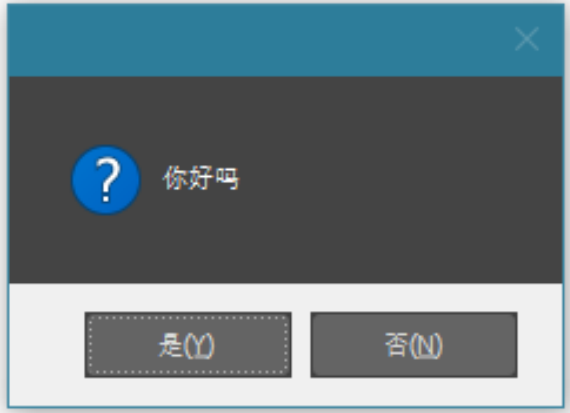
显示一个模式消息框，类似于 `messageBox()` 函数创建，除了它包含是和否按钮。这 `queryBox()` 函数返回 `true` 如果用户单击 Yes 和 `false` 如果用户点击否。

```
yesNoCancelBox <message_string> [ title : <window_title_string> ] [ beep : <boolean> ]
```

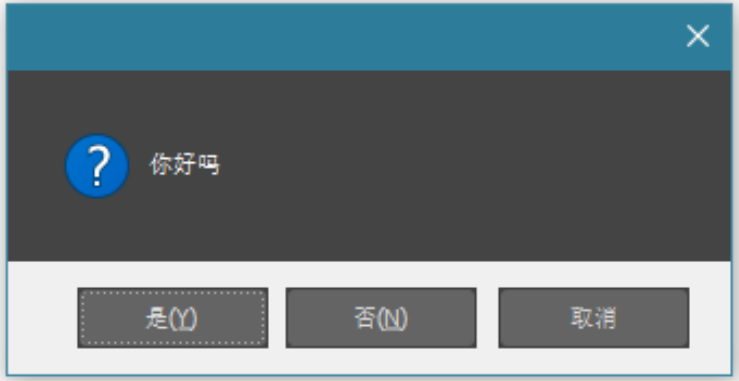
显示一个模式消息框，类似于 `messageBox()` 函数创建，但它包含一个是、一个否和一个取消按钮。这 `yesNoCancelBox()` 函数返回 `#yes` , `#no` 或者 `#cancel` 取决于用户单击哪个按钮来关闭消息框。

消息提示框（MessageBox）

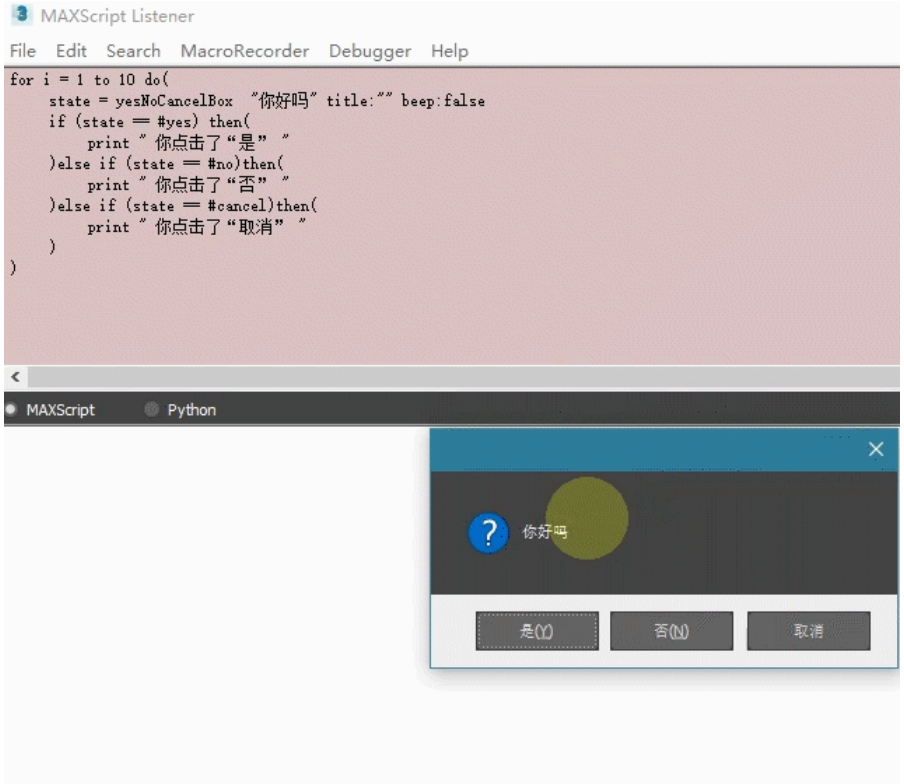
```
state = querybox "你好吗" title:"" beep:false
false
true
```




```
state = yesNoCancelBox  "你好吗" title:"" beep:false
#cancel
#no
#yes
```



消息提示框（MessageBox）



自定义函数

自定义函数

```
—语法
    function <函数名> <形参> = ( <代码> )

—示例
    —弹出出一个信息提示框，提示内容为“data”
      fn dayin temp =
      (
          messagebox ( temp) title:"Tips" beep:False
      )
      dayin  "这是一个自定义函数"
```

MAXScript Python

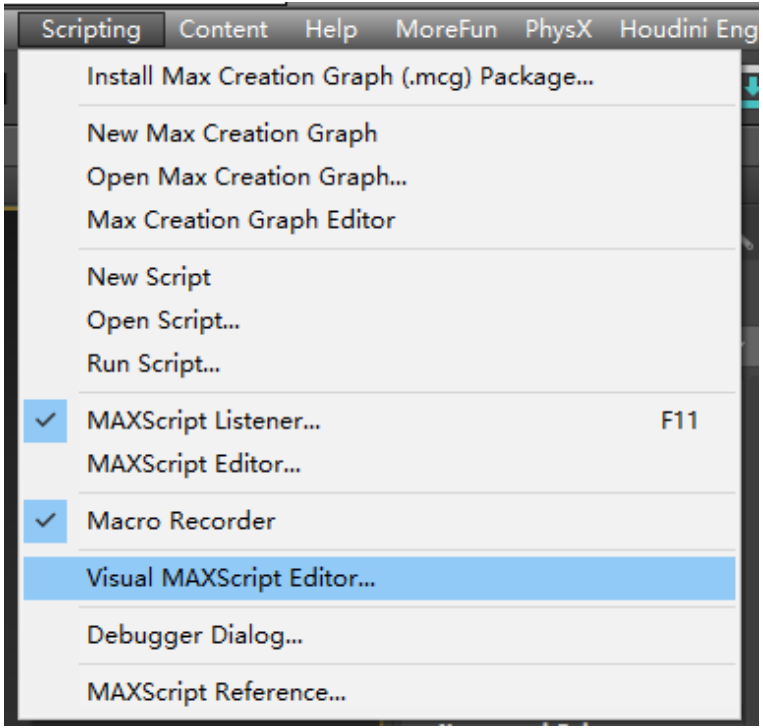
```
dayin()
```

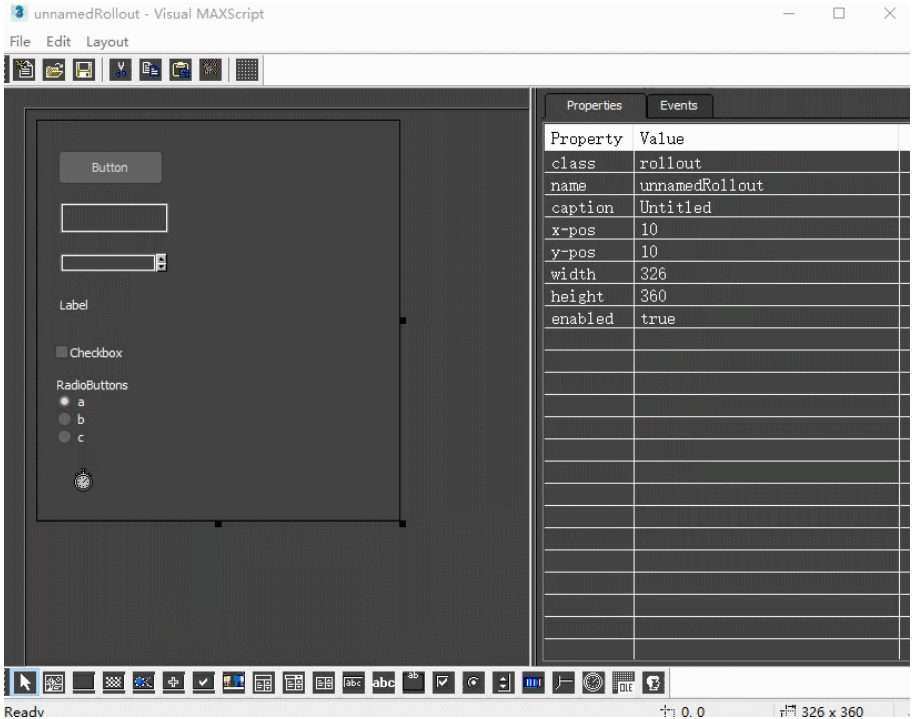
案例讲解

脚本框架

脚本框架

界面编辑器


Scripting
Content
Help
MoreFun
PhysX
Houdini Eng
Install Max Creation Graph (.mcg) Package...
New Max Creation Graph
Open Max Creation Graph...
Max Creation Graph Editor
New Script
Open Script...
Run Script...
MAXScript Listener... F11
MAXScript Editor...
Macro Recorder
Visual MAXScript Editor...
Debugger Dialog...
MAXScript Reference...


unnamedRollout - Visual MAXScript
File Edit Layout
Button
Label
Checkbox
RadioButtons
a
b
c
Properties
Events
Property Value
class rollout
name unnamedRollout
caption Untitled
x-pos 10
y-pos 10
width 326
height 360
enabled true
Ready
0, 0
326 x 360

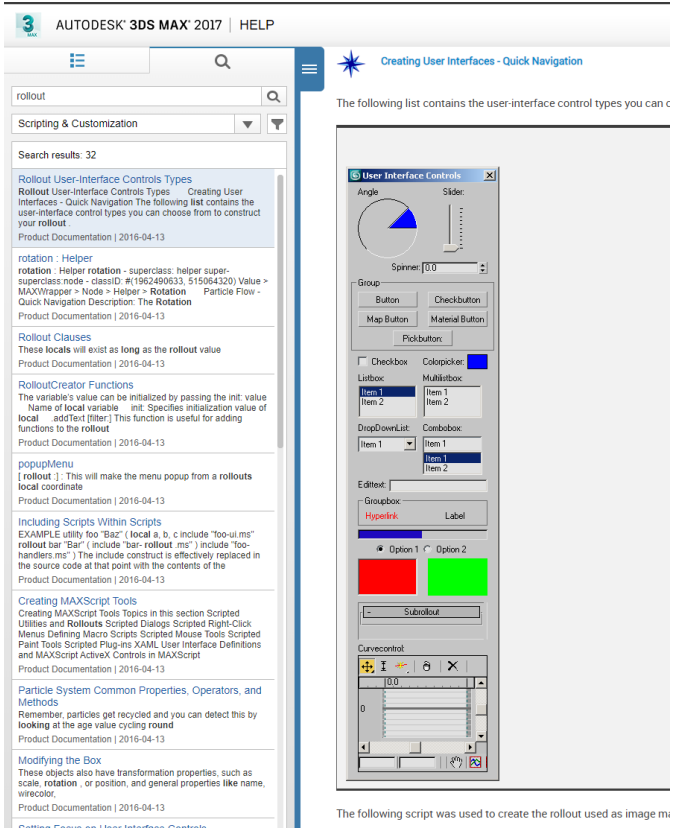
脚本框架

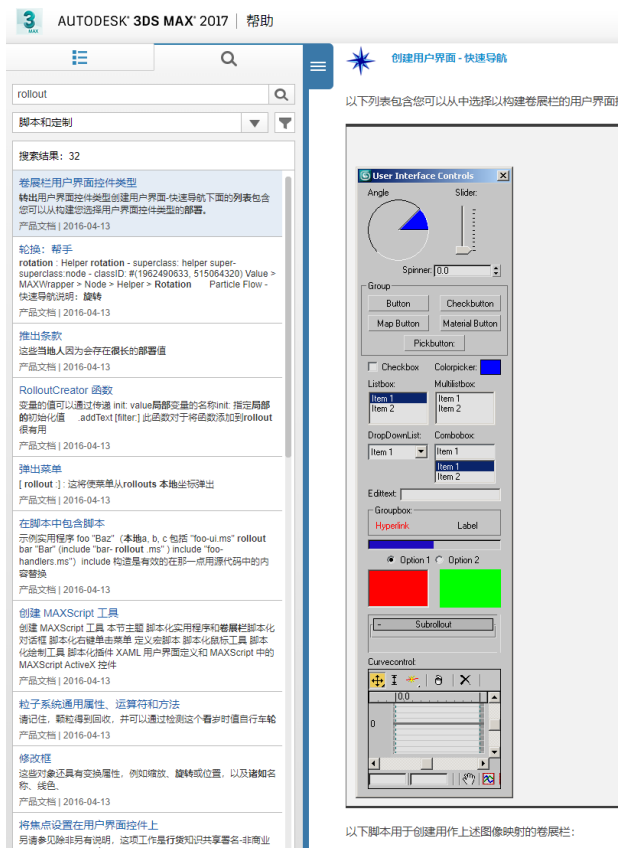
界面编辑器

```
rollout unnamedRollout "Untitled" width:326 height:360
(
    button 'btn5' "Button" pos:[19,27] width:94 height:30 align:#left
    editText 'edt1' "" pos:[17,74] width:101 height:27 align:#left
    label 'lbl1' "Label" pos:[21,159] width:119 height:29 align:#left
    checkbox 'chk1' "Checkbox" pos:[16,194] width:81 height:29 align:#left
    radioButtons 'rdo1' "RadioButtons" pos:[18,231] width:30 height:62 labels:#("a", "b", "c") align:#left
    spinner 'spn2' "" pos:[20,120] width:99 height:16 align:#left
    timer 'tmr1' "Timer" pos:[30,312] width:24 height:24 align:#left
    label 'lbl3' "12312312312" pos:[157,36] width:71 height:25 align:#left
    editText 'edt3' "path:" pos:[157,102] width:122 height:41 align:#left
    button 'btn7' "click" pos:[154,151] width:150 height:30 align:#left
)
```

脚本框架

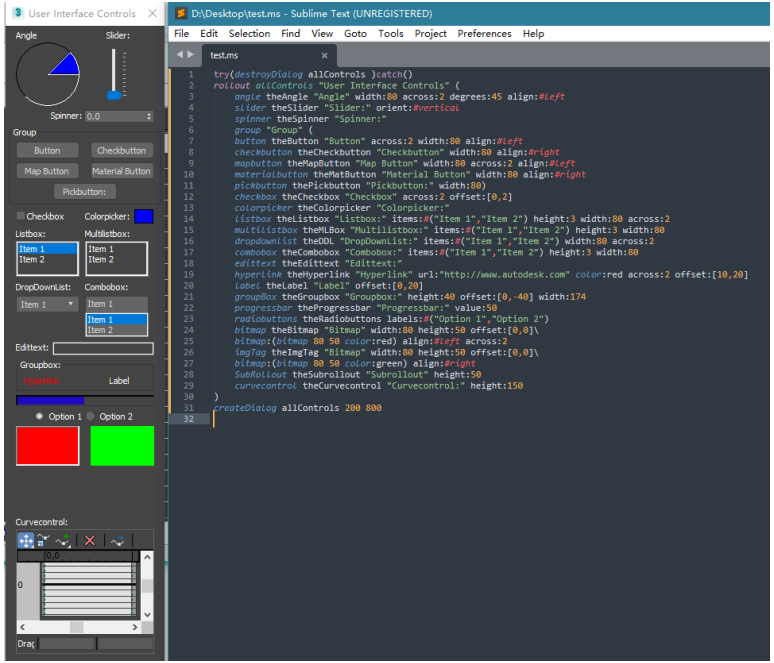
界面编辑器

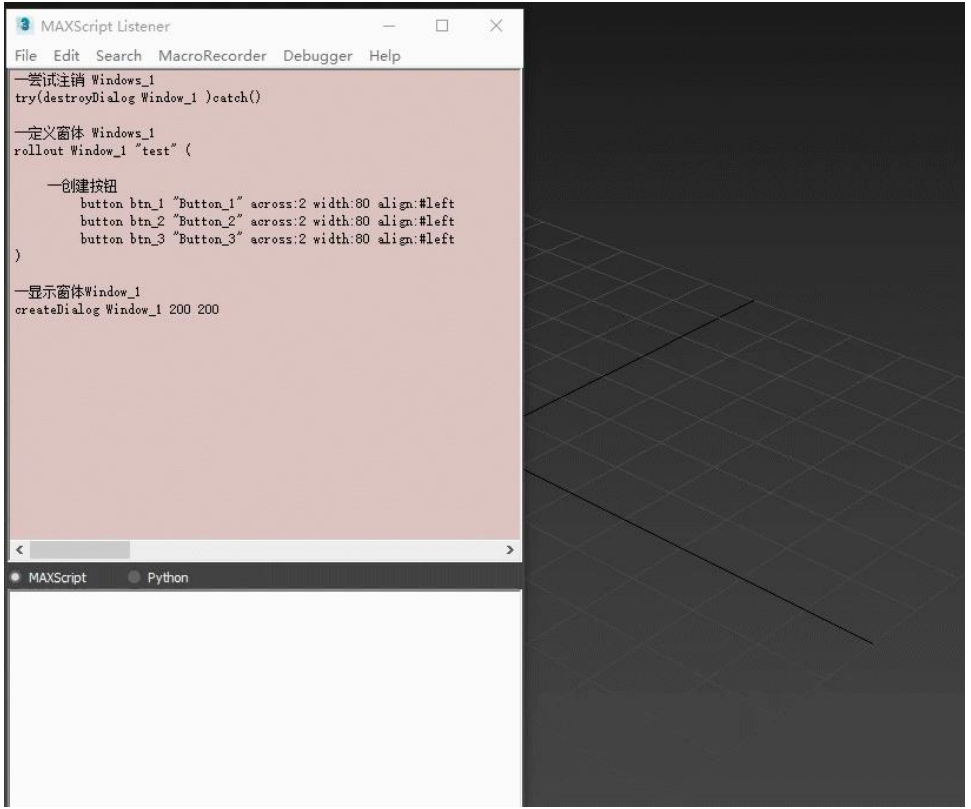

AUTODESK 3DS MAX 2017 | HELP
rollout
Scripting & Customization
Search results: 32
Rollout User-Interface Controls Types
Rollout User-Interface Controls Types Creating User Interfaces - Quick Navigation The following list contains the user-interface control types you can choose from to construct your rollout .
Product Documentation | 2016-04-13
rotation : Helper
rotation : Helper rotation - superclass: helper super-superclass:node - classID: #(1962490633, 515064320) Value > MAXWrapper > Node > Helper > Rotation Particle Flow - Quick Navigation Description: The Rotation
Product Documentation | 2016-04-13
Rollout Clauses
These locals will exist as long as the rollout value
Product Documentation | 2016-04-13
RolloutCreator Functions
The variable's value can be initialized by passing the init: value Name of local variable init: Specifies initialization value of local .addText [filter:] This function is useful for adding functions to the rollout
Product Documentation | 2016-04-13
popupMenu
[rollout :] : This will make the menu popup from a rollouts local coordinate
Product Documentation | 2016-04-13
Including Scripts Within Scripts
EXAMPLE utility foo "Baz" (local a, b, c include "foo-ui.ms" rollout bar "Bar" (include "bar- rollout .ms") include "foo-handlers.ms") The include construct is effectively replaced in the source code at that point with the contents of the
Product Documentation | 2016-04-13
Creating MAXScript Tools
Creating MAXScript Tools Topics in this section Scripted Utilities and Rollouts Scripted Dialogs Scripted Right-Click Menus Defining Macro Scripts Scripted Mouse Tools Scripted Paint Tools Scripted Plug-ins XAML User Interface Definitions and MAXScript ActiveX Controls in MAXScript
Product Documentation | 2016-04-13
Particle System Common Properties, Operators, and Methods
Remember, particles get recycled and you can detect this by looking at the age value cycling round
Product Documentation | 2016-04-13
Modifying the Box
These objects also have transformation properties, such as scale, rotation , or position, and general properties like name, wirecolor,
Product Documentation | 2016-04-13
Creating User Interfaces - Quick Navigation
The following list contains the user-interface control types you can c
User Interface Controls
Angle
Slider:
Spinner:
Group
Button
Checkbox
Map Button
Material Button
Pickbutton:
Checkbox
Colorpicker:
Listbox:
Multilistbox:
Item 1
Item 2
DropDownList:
Combobox:
Editext:
Groupbox:
Hyperlink
Label
Option 1
Option 2
Subrollout
Curvecontrol:
The following script was used to create the rollout used as image ma


AUTODESK 3DS MAX 2017 | 帮助
rollout
脚本和定制
搜索结果：32
卷展栏用户界面控件类型
转出用户界面控件类型创建用户界面-快速导航下面的列表包含您可以从构建您选择用户界面控件类型的部署。
产品文档 | 2016-04-13
轮换：帮手
rotation : Helper rotation - superclass: helper super-superclass:node - classID: #(1962490633, 515064320) Value > MAXWrapper > Node > Helper > Rotation Particle Flow - 快速导航说明：旋转
产品文档 | 2016-04-13
推出条款
这些当地人因为会存在很长的部署值
产品文档 | 2016-04-13
RolloutCreator 函数
变量的值可以通过传递 init: value局部变量的名称init: 指定局部的初始化值 .addText [filter:] 此函数对于将函数添加到rollout很有用
产品文档 | 2016-04-13
弹出菜单
[rollout :] : 这将便菜单从rollouts 本地坐标弹出
产品文档 | 2016-04-13
在脚本中包含脚本
示例实用程序 foo "Baz" （本地a, b, c 包括 "foo-ui.ms" rollout bar "Bar" (include "bar- rollout .ms") include "foo-handlers.ms") include 构造是有效的在那一点用源代码中的内容替换
产品文档 | 2016-04-13
创建 MAXScript 工具
创建 MAXScript 工具 本节主题 脚本化实用程序和卷展栏脚本化对话框 脚本化右键单击菜单 定义宏脚本 脚本化鼠标工具 脚本化绘制工具 脚本化插件 XAML 用户界面定义和 MAXScript 中的 MAXScript ActiveX 控件
产品文档 | 2016-04-13
粒子系统通用属性、运算符和方法
请记住，颗粒得到回收，并可以通过检测这个看岁时值自行车轮
产品文档 | 2016-04-13
修改程
这些对象还具有变换属性，例如缩放、旋转或位置，以及诸如名称、线色、
产品文档 | 2016-04-13
将焦点设置在用户界面控件上
另请参见除非另有说明，这项工作是行货知识共享署名-非商业
创建用户界面 - 快速导航
以下列表包含您可以从中选择以构建卷展栏的用户界面
User Interface Controls
Angle
Slider:
Spinner:
Group
Button
Checkbox
Map Button
Material Button
Pickbutton:
Checkbox
Colorpicker:
Listbox:
Multilistbox:
Item 1
Item 2
DropDownList:
Combobox:
Editext:
Groupbox:
Hyperlink
Label
Option 1
Option 2
Subrollout
Curvecontrol
以下脚本用于创建用作上述图像映射的卷展栏：

脚本框架

界面编辑器

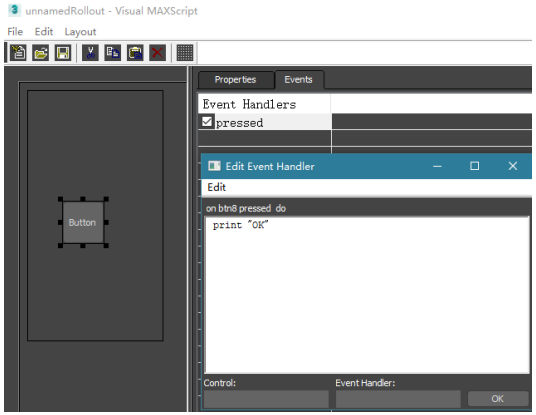

User Interface Controls
Angle
Slider:
Spinner: 0.0
Group
Button
Checkbutton
Map Button
Material Button
Pickbutton:
Checkbox
Colorpicker:
Listbox:
Multilistbox:
Item 1
Item 2
DropDownList:
Combobox:
Edittext:
Groupbox:
Hyperlink
Label
Option 1
Option 2
Curvecontrol:
D:\Desktop\test.ms - Sublime Text (UNREGISTERED)
File Edit Selection Find View Goto Tools Project Preferences Help
test.ms
try(destroyDialog allControls)catch()
rollout allControls "User Interface Controls" (
angle theAngle "Angle" width:80 across:2 degrees:45 align:#left
slider theSlider "Slider:" orient:#vertical
spinner theSpinner "Spinner:"
group "Group" (
button theButton "Button" across:2 width:80 align:#left
checkbutton theCheckbutton "Checkbutton" width:80 align:#right
mapbutton theMapButton "Map Button" width:80 across:2 align:#left
materialbutton theMatButton "Material Button" width:80 align:#right
pickbutton thePickbutton "Pickbutton:" width:80)
checkbox theCheckbox "Checkbox" across:2 offset:[0,2]
colorpicker theColorpicker "Colorpicker:"
listbox theListbox "Listbox:" items:#("Item 1","Item 2") height:3 width:80 across:2
multilistbox theMLBox "Multilistbox:" items:#("Item 1","Item 2") height:3 width:80
dropdownlist theDDL "DropDownList:" items:#("Item 1","Item 2") width:80 across:2
combobox theCombobox "Combobox:" items:#("Item 1","Item 2") height:3 width:80
edittext theEdittext "Edittext:"
hyperlink theHyperlink "Hyperlink" url:"http://www.autodesk.com" color:red across:2 offset:[10,20]
label theLabel "Label" offset:[0,20]
groupBox theGroupbox "Groupbox:" height:40 offset:[0,-40] width:174
progressbar theProgressbar "Progressbar:" value:50
radiobuttons theRadiobuttons labels:#("Option 1","Option 2")
bitmap theBitmap "Bitmap" width:80 height:50 offset:[0,0]\
bitmap:(bitmap 80 50 color:red) align:#left across:2
imgTag theImgTag "Bitmap" width:80 height:50 offset:[0,0]\
bitmap2(bitmap 80 50 color:green) align:#right
SubRollout theSubrollout "Subrollout" height:50
curvecontrol theCurvecontrol "Curvecontrol:" height:150
)
createDialog allControls 200 800

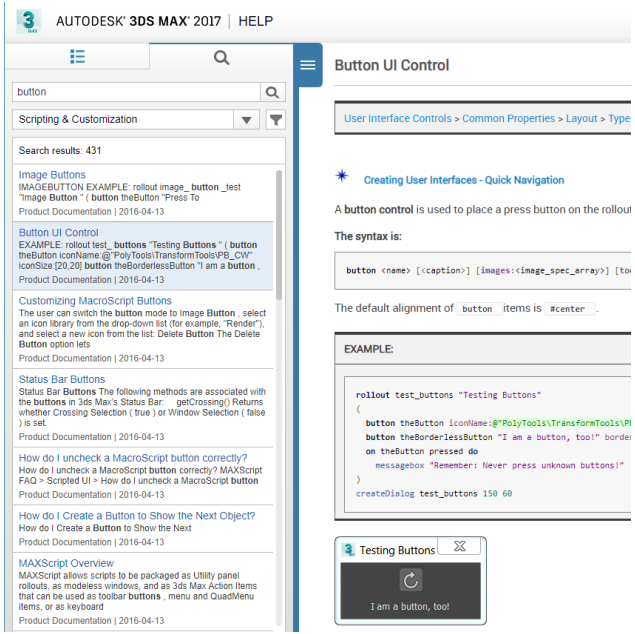

MAXScript Listener
File Edit Search MacroRecorder Debugger Help
一尝试注销 Windows_1
try(destroyDialog Window_1)catch()
一定义窗体 Windows_1
rollout Window_1 "test" (
一创建按钮
button btn_1 "Button_1" across:2 width:80 align:#left
button btn_2 "Button_2" across:2 width:80 align:#left
button btn_3 "Button_3" across:2 width:80 align:#left
)
一显示窗体Window_1
createDialog Window_1 200 200
MAXScript
Python

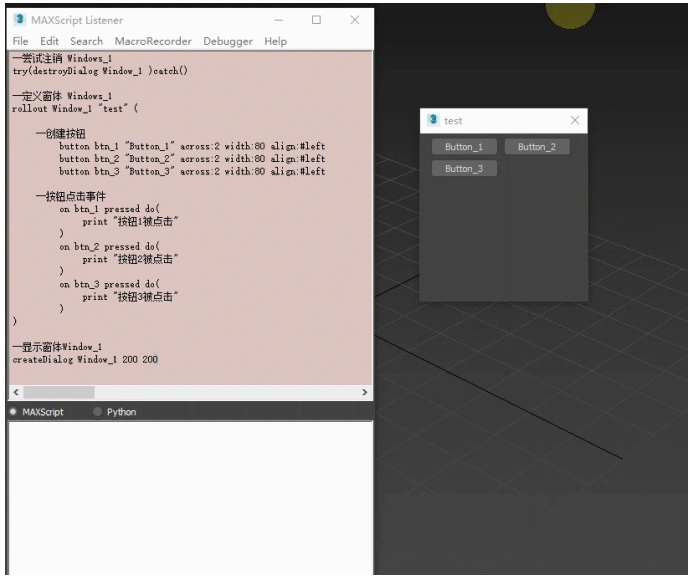
事件

脚本框架

界面编辑器


unnamedRollout - Visual MAXScript
File Edit Layout
Properties
Events
Event Handlers
pressed
Edit Event Handler
Edit
on btn8 pressed do
print "OK"
Button
Control:
Event Handler:
OK


AUTODESK 3DS MAX 2017 | HELP
button
Scripting & Customization
Search results: 431
Image Buttons
Button UI Control
Customizing MacroScript Buttons
Status Bar Buttons
How do I uncheck a MacroScript button correctly?
How do I Create a Button to Show the Next Object?
MAXScript Overview
Button UI Control
User Interface Controls > Common Properties > Layout > Type
Creating User Interfaces - Quick Navigation
A button control is used to place a press button on the rollout
The syntax is:
The default alignment of button items is #center.
EXAMPLE:
rollout test_buttons "Testing Buttons"
Testing Buttons
I am a button, too!


MAXScript Listener
File Edit Search MacroRecorder Debugger Help
一尝试注销 Windows_1
try(destroyDialog Window_1)catch()
一定义窗体 Windows_1
rollout Window_1 "test" (
一创建按钮
button btn_1 "Button_1" across:2 width:80 align:#left
button btn_2 "Button_2" across:2 width:80 align:#left
button btn_3 "Button_3" across:2 width:80 align:#left
一按钮点击事件
on btn_1 pressed do(
print "按钮1被点击"
)
on btn_2 pressed do(
print "按钮2被点击"
)
on btn_3 pressed do(
print "按钮3被点击"
)
)
一显示窗体Window_1
createDialog Window_1 200 200
MAXScript
Python
test
Button_1
Button_2
Button_3

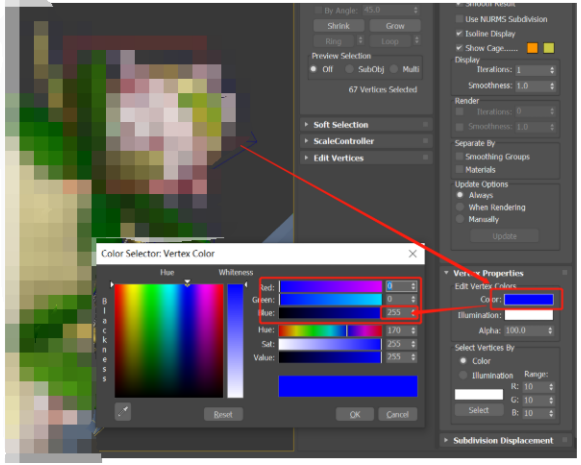
模型顶点色设置脚本

问题描述

- 为了实现一些效果，模型max文件中有部分mesh需要设置顶点色
- 并且顶点色的设置有一定的规则，手动设置效率比较低

- 表情对应的Mesh命名设置
  - 如图，“Eye”这个表情类别有3个Mesh，那么除了默认的Mesh命名保持不变，另外两个分别加上后缀“_EC1、“_EC2”。

  - 同类型表情最多支持6个。除了默认的Mesh命名之外，剩下的5个Mesh，其命名分别加上后缀“_EC1”、“_EC2”、“_EC3、“_EC4”、”_EC5“
- 表情Mesh顶点色设置
  - 例：如图
    - 如果当前Mesh为默认的，其顶点色的B通道为255，即1*255
    - 如果当前Mesh为 EC1 ，那么其顶点色的B通道为204，即0.8*255
    - 如果当前Mesh为 EC2 ，那么其顶点色的B通道为204，即0.6*255
    - 以此类推....



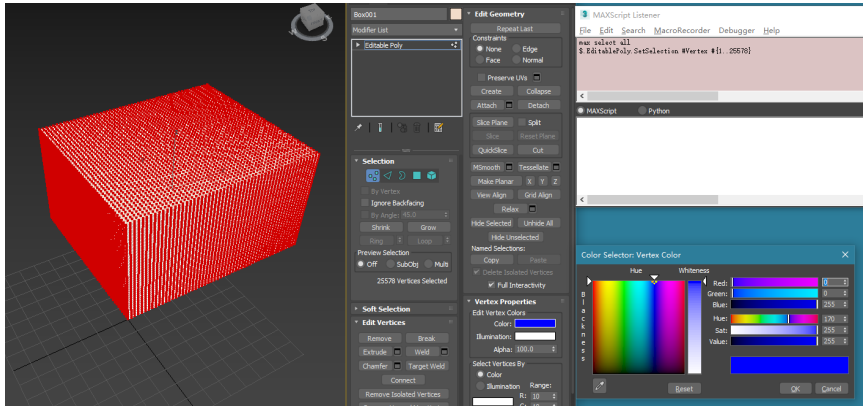
脚本流程

插件所需功能

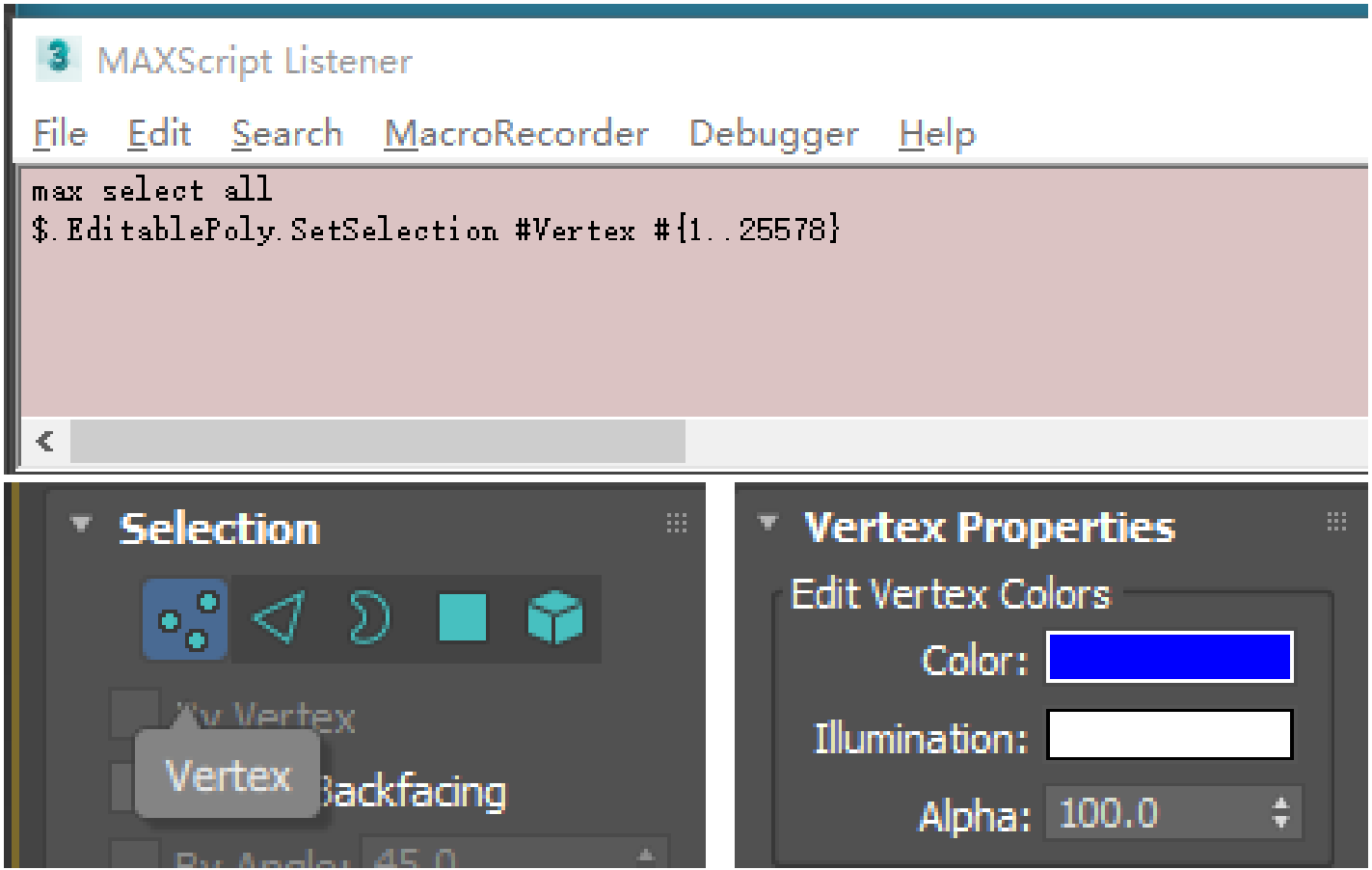
- “选中对应模型”，然后“点击按钮”实现顶点色赋予
- 需要支持可以同时选中多个模型进行顶点色的赋予

手动操作流程

- 选中模型，进入“点级别”选择模式
- 选中所有顶点
- 计算出需要设置的顶点色，然后进行赋予



插件编写思路


MAXScript Listener
File Edit Search MacroRecorder Debugger Help
max select all
$.EditablePoly.SetSelection #Vertex #{1..25578}
Selection
Vertex
Backfacing
Vertex Properties
Edit Vertex Colors
Color:
Illumination:
Alpha:
100.0

插件编写思路


```
fn setVertexColor node vert col = if iskindof node Editable_Poly do (
    if not polyop.getmapsupport node 0 do polyop.setmapsupport node 0 on
    for face in polyop.getfacesusingvert node vert do
    (
        vv = polyop.getfaceverts node face
        tv = polyop.getmapface node 0 face
        polyop.setmapvert node 0 tv[finditem vv vert] (col as point4)
    )
)
```

插件编写思路

```
fn setVertexColor node vert col = if iskindof node Editable_Poly do (
    if not polyop.getmapsupport node 0 do polyop.setmapsupport node 0 on
    for face in polyop.getfacesusingvert node vert do
    (
        vv = polyop.getfaceverts node face
        tv = polyop.getmapface node 0 face
        polyop.setmapvert node 0 tv[finditem vv vert] (col as point4)
    )
)
```

面对面

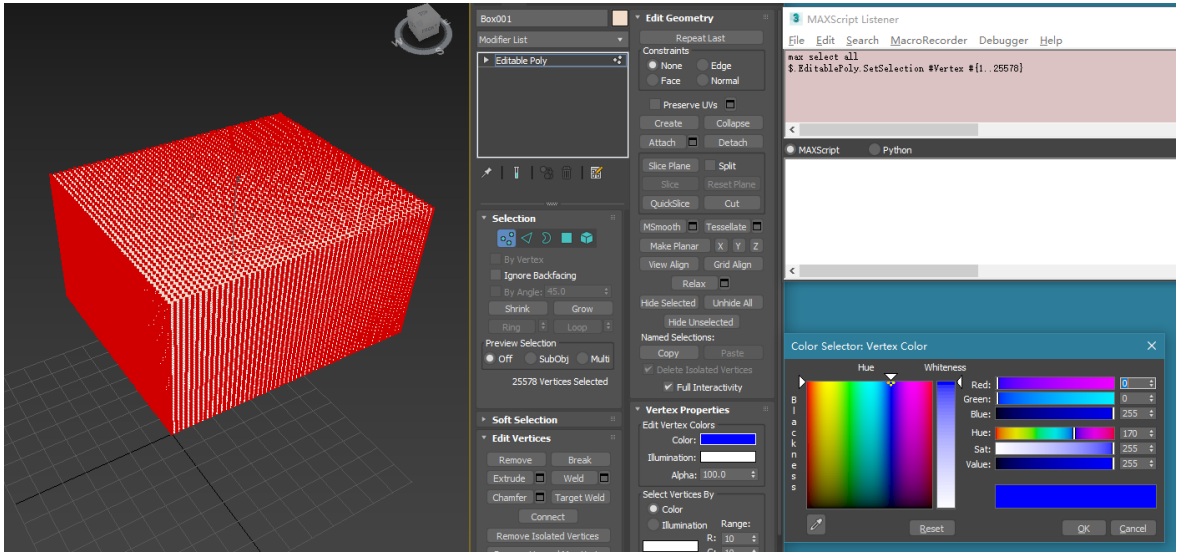
返回主题导航

息肉。getFaceVerts <Poly poly> <int face>

以数组形式返回面的顶点。数组中顶点的顺序对应于面中顶点的顺序。

插件编写思路

```
fn setVertexColor node vert col = if iskindof node Editable_Poly do (
    if not polyop.getmapsupport node 0 do polyop.setmapsupport node 0 on
    for face in polyop.getfacesusingvert node vert do
    (
        vv = polyop.getfaceverts node face
        tv = polyop.getmapface node 0 face
        polyop.setmapvert node 0 tv[finditem vv vert] (col as point4)
    )
)
```



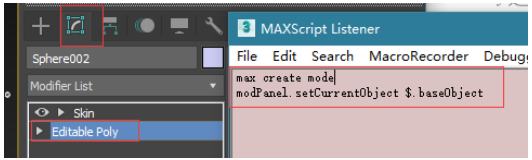
插件编写思路

```
fn setVertexColor node vert col = if iskindof node Editable_Poly do (
    if not polyop.getmapsupport node 0 do polyop.setmapsupport node 0 on
    for face in polyop.getfacesusingvert node vert do
    (
        vv = polyop.getfaceverts node face
        tv = polyop.getmapface node 0 face
        polyop.setmapvert node 0 tv[finditem vv vert] (col as point4)
    )
)
fn setObjVertexColor R G B = (
    try(
        SelectMeshArray=#()
        SelectMeshName=#()
        SelectMeshArrayCount=0
        try(
            x = $
            x as array
            SelectMeshArray=x
        )catch(
            x = $
            SelectMeshArray=#(x)
        )
        SelectMeshArrayCount =  SelectMeshArray.count
        for k = 1 to SelectMeshArrayCount do(
            SelectMeshName[k]=SelectMeshArray[k].name
        )
        for i = 1 to SelectMeshArrayCount do (
        --switch editablepoly
            clearSelection()
            max modify mode
            bi = getNodeByName SelectMeshName[i]
            select bi
            modPanel.setCurrentObject $.baseObject
            subobjectLevel = 0
        --setvertex
            MeshVertexCount = getNumVerts $
            for j = 1 to MeshVertexCount do(
                setVertexColor $ j [R,G,B]
            )
        )
        --
        max modify mode
        max create mode
        clearSelection()
    )catch(messagebox "请选择模型" title:"Error" beep:false)
)
```

```
SelectMeshArray=#()
SelectMeshName=#()
SelectMeshArrayCount=0
try(
    x = $
    x as array
    SelectMeshArray=x
)catch(
    x = $
    SelectMeshArray=#(x)
)
```

```
SelectMeshArrayCount =  SelectMeshArray.count
for k = 1 to SelectMeshArrayCount do(
    SelectMeshName[k]=SelectMeshArray[k].name
)
```

```
for i = 1 to SelectMeshArrayCount do (
--switch editablepoly
    clearSelection()
    max modify mode
    bi = getNodeByName SelectMeshName[i]
    select bi
    modPanel.setCurrentObject $.baseObject
    subobjectLevel = 0
--setvertex
    MeshVertexCount = getNumVerts $
    for j = 1 to MeshVertexCount do(
        setVertexColor $ j [R,G,B]
    )
)
```




```
MeshVertexCount = getNumVerts $
for j = 1 to MeshVertexCount do(
    setVertexColor $ j [R,G,B]
)
```

插件编写思路

```
fn setVertexColor node vert col = if iskindof node Editable_Poly do (
    if not polyop.getmapsupport node 0 do polyop.setmapsupport node 0 on
    for face in polyop.getfacesusingvert node vert do
    (
        vv = polyop.getfaceverts node face
        tv = polyop.getmapface node 0 face
        polyop.setmapvert node 0 tv[finditem vv vert] (col as point4)
    )
)
fn setObjVertexColor R G B = (
    try(
        SelectMeshArray=#()
        SelectMeshName=#()
        SelectMeshArrayCount=0
        try(
            x = $
            x as array
            SelectMeshArray=x
        )catch(
            x = $
            SelectMeshArray=#(x)
        )
        SelectMeshArrayCount =  SelectMeshArray.count
        for k = 1 to SelectMeshArrayCount do(
            SelectMeshName[k]=SelectMeshArray[k].name
        )
        for i = 1 to SelectMeshArrayCount do (
        --switch editablepoly
            clearSelection()
            max modify mode
            bi = getNodeByName SelectMeshName[i]
            select bi
            modPanel.setCurrentObject $.baseObject
            subobjectLevel = 0
        --setvertex
            MeshVertexCount = getNumVerts $
            for j = 1 to MeshVertexCount do(
                setVertexColor $ j [R,G,B]
            )
        )
        --
        max modify mode
        max create mode
        clearSelection()
    )catch(messagebox "请选择模型" title:"Error" beep:false)
)
```

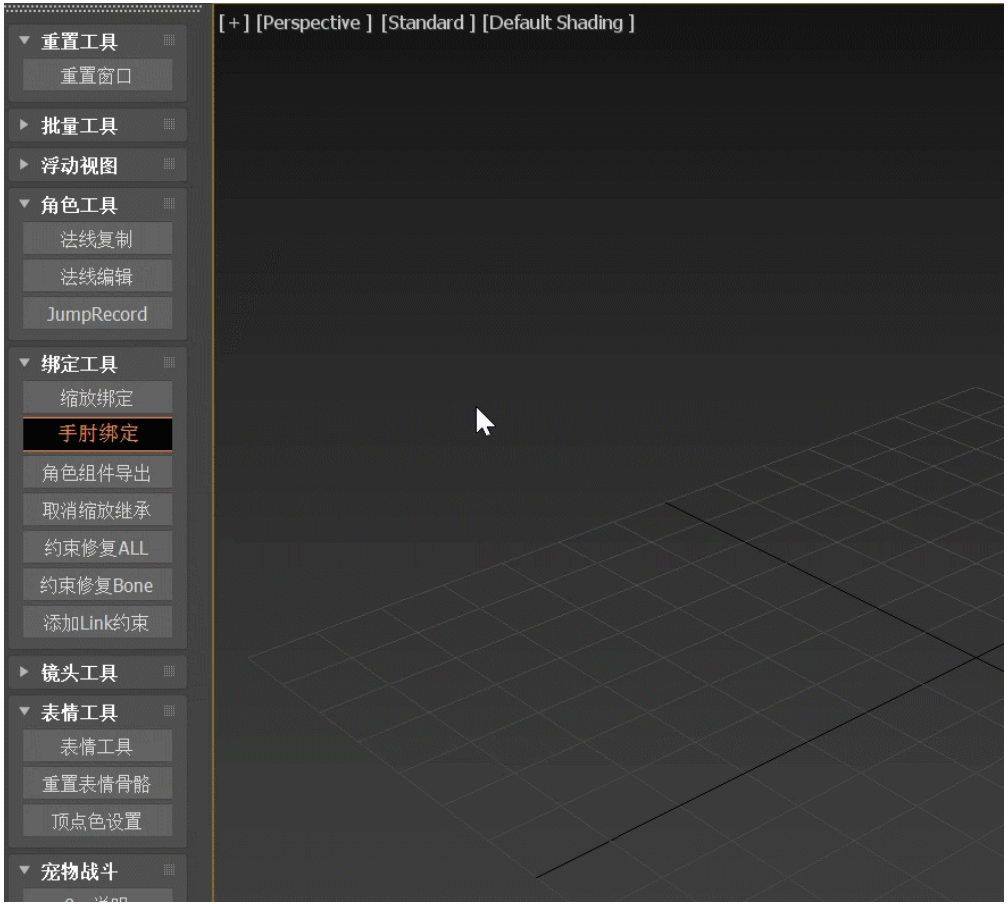
```
fn setObjVertexColor R G B = (
    try(
```

```
 MeshVertexCount = getNumVerts $
 for j = 1 to MeshVertexCount do(
     setVertexColor $ j [R,G,B]
 )
```

```
fn setDefault =(
    setObjVertexColor 0 0 1.0
)
fn setEC1 =(
    setObjVertexColor 0 0 0.8
)
fn setEC2 =(
    setObjVertexColor 0 0 0.6
)
fn setEC3 =(
    setObjVertexColor 0 0 0.4
)
fn setEC4 =(
    setObjVertexColor 0 0 0.2
)
fn setEC5 =(
    setObjVertexColor 0 0 0.0
)
```

额外知识点
DOTNET 界面框架

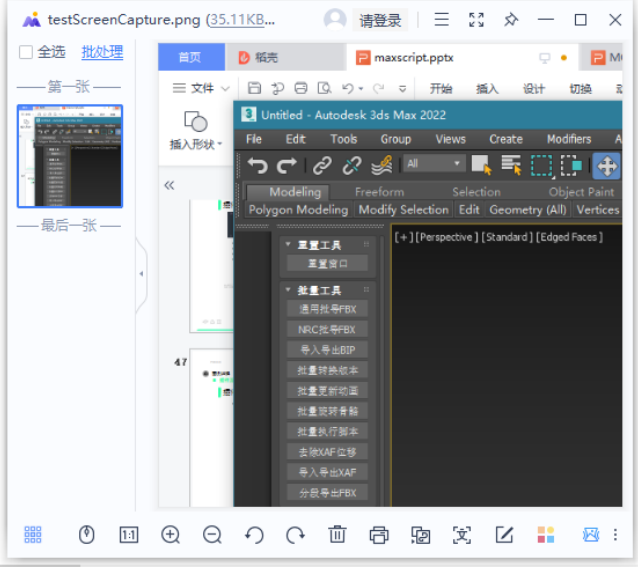
Dotnet


[+] [Perspective] [Standard] [Default Shading]
重置工具
重置窗口
批量工具
浮动视图
角色工具
法线复制
法线编辑
JumpRecord
绑定工具
缩放绑定
手肘绑定
角色组件导出
取消缩放继承
约束修复ALL
约束修复Bone
添加Link约束
镜头工具
表情工具
表情工具
重置表情骨骼
顶点色设置
宠物战斗

Dotnet

```
Fn ScreenCapture PosX PosY ImgW ImgH ImgFileName =
(
try
    (
        local DotNetBmp,DotNetGraphics,DotNetPoint
        local tempBmp,tempGraphic
        DotNetBmp = DotNetClass "System.Drawing.Bitmap"
        DotNetGraphics = DotNetClass "System.Drawing.Graphics"
        DotNetPoint = DotNetClass "System.Drawing.Point"
        tempBmp = DotNetObject "System.Drawing.Bitmap" ImgW ImgH
        tempGraphic = DotNetGraphics.FromImage tempBmp
        tempGraphic.CopyFromScreen (DotNetObject DotNetPoint PosX  PosY) \
                                   (DotNetObject DotNetPoint 0 0) tempBmp.Size
        tempBmp.Save ImgFileName
        tempGraphic.Dispose()
true
    )
catch (false)
)

ScreenCapture  0 0 500 500 "d:\\testScreenCapture.png"
ScreenCapture()
true
```



Dotnet

MAXScript Listener

File Edit Search MacroRecorder Debugger Help

MAXScript Python

```
CSharpCodeSource ="using System;
using System.Windows.Forms;
namespace TestNameSpace
{
    public class TestClass
    {
        public void Test(string theString)
        {
            MessageBox.Show(theString);
        }
    }
}
"

CSharpProvider = dotnetobject "Microsoft.CSharp.CSharpCodeProvider"
CompilerParams = dotnetobject "System.CodeDom.Compiler.CompilerParameters"

CompilerParams.ReferencedAssemblies.Add("System.dll")
CompilerParams.ReferencedAssemblies.Add("System.Windows.Forms.dll")
CompilerParams.GenerateExecutable = false
CompilerParams.GenerateInMemory = false

CompilerResults = CSharpProvider.CompileAssemblyFromSource CompilerParams #(CSharpCodeSource)

WinAssembly = CompilerResults.CompiledAssembly
TestAssembly =  WinAssembly.CreateInstance "TestNameSpace.TestClass"
TestAssembly.Test "动态编译的C#"

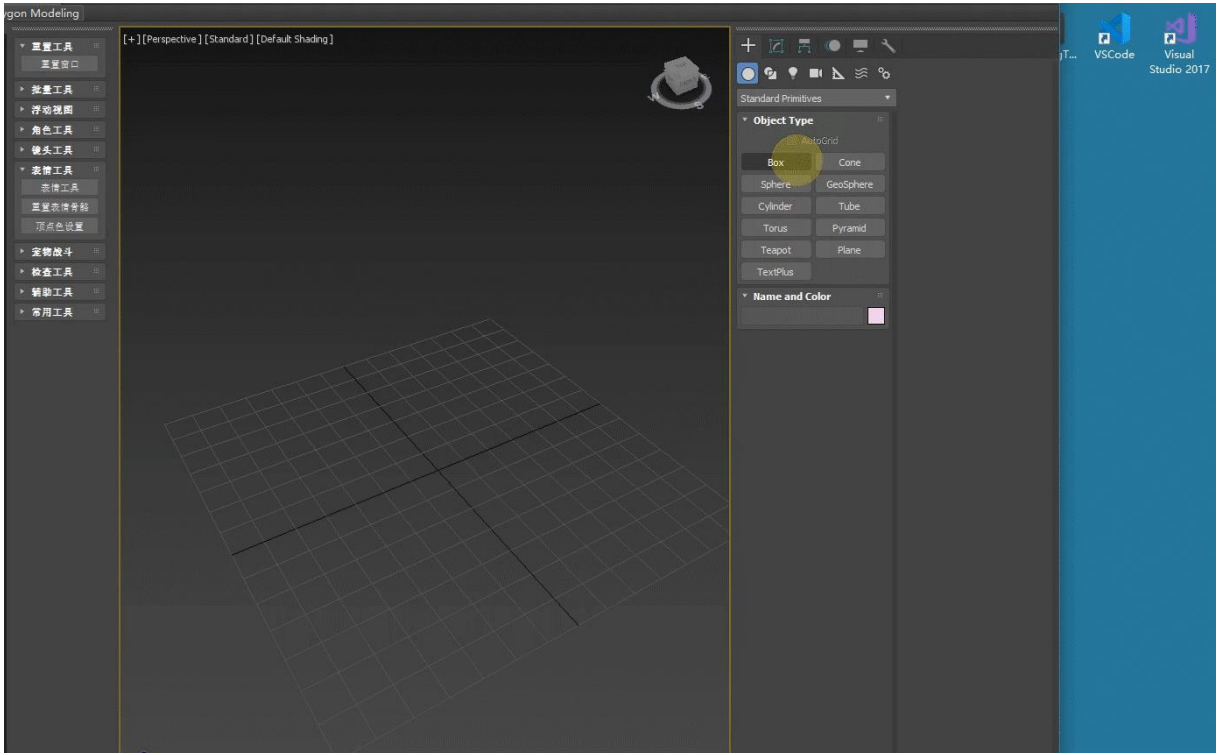
(DotNetObject "TestNameSpace.TestClass").Test "123"
"using System;
using System.Windows.Forms;
namespace TestNameSpace
{
    public class TestClass
    {
        public void Test(string theString)
        {
            MessageBox.Show(theString);
        }
    }
}
"
dotNetObject:Microsoft.CSharp.CSharpCodeProvider
dotNetObject:System.CodeDom.Compiler.CompilerParameters
0
1
false
false
dotNetObject:System.CodeDom.Compiler.CompilerResults
dotNetObject:System.Reflection.RuntimeAssembly
dotNetObject:TestNameSpace.TestClass
```

Dotnet

```
fn setVertexColor node vert col = if isKindOf node Editable_Poly do (
)
fn setObjVertexColor R G B = (
)
fn setDefault =(
)
fn setEC1 =(
)
fn setEC2 =(
)
fn setEC3 =(
)
fn setEC4 =(
)
fn setEC5 =(
)

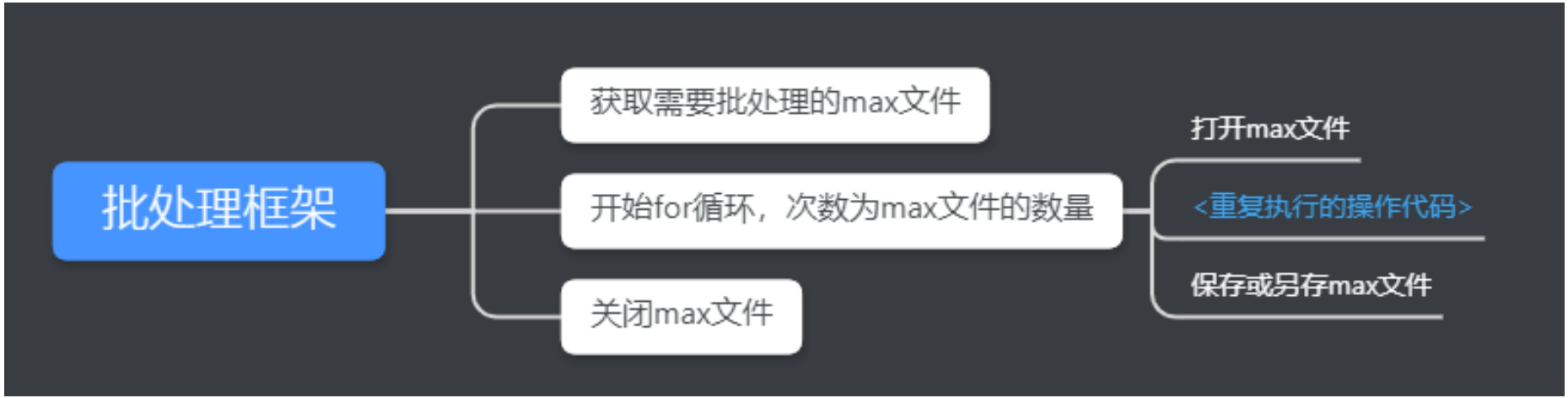
--创建dotnet组件，赋值给变量
    button1 = dotNetObject "System.Windows.Forms.Button"
    button2 = dotNetObject "System.Windows.Forms.Button"
    button3 = dotNetObject "System.Windows.Forms.Button"
    button4 = dotNetObject "System.Windows.Forms.Button"
    button5 = dotNetObject "System.Windows.Forms.Button"
    button6 = dotNetObject "System.Windows.Forms.Button"
    MainForm = dotNetObject "System.Windows.Forms.Form"
-- button1的一些参数设置
    button1.BackColor = (dotNetClass "System.Drawing.Color").Black
    button1.FlatStyle = (dotNetClass "System.Windows.Forms.FlatStyle").Flat
    button1.ForeColor = (dotNetClass "System.Drawing.Color").Coral
    button1.Location = dotNetObject "System.Drawing.Point" 60 12
    button1.Name = "button1"
    button1.Size = dotNetObject "System.Drawing.Size" 107 45
    button1.TabIndex = 0
    button1.Text = "Default"
    button1.UseVisualStyleBackColor = False
-- button2
-- button3
-- button4
-- button5
-- button6
--窗体组件的一些参数设置
    MainForm.location = dotNetObject "System.Drawing.Point" -100 -100
    MainForm.topmost = true
    FormBorderStyle = dotNetClass "System.Windows.Forms.FormBorderStyle"
    MainForm.FormBorderStyle = (dotNetClass "System.Windows.Forms.FormBorderStyle").SizableToolWindow
    MainForm.BackColor=MainForm.BackColor.black
    MainForm.Opacity = 0.75
    MainForm.text ="TEC_setVertexColor"
    MainForm.bounds=dotNetObject "system.drawing.rectangle" 10 90 250 400
    MainForm.StartPosition = (dotNetClass "System.Windows.Forms.FormStartPosition").CenterScreen
--将按钮添加到窗体上
    MainForm.Controls.Add(button1)
    MainForm.Controls.Add(button2)
    MainForm.Controls.Add(button3)
    MainForm.Controls.Add(button4)
    MainForm.Controls.Add(button5)
    MainForm.Controls.Add(button6)
--为对应按钮的"点击事件"添加代码
    dotNet.addEventHandler button1 "click" (setDefault)
    dotNet.addEventHandler button2 "click" (setEC1)
    dotNet.addEventHandler button3 "click" (setEC2)
    dotNet.addEventHandler button4 "click" (setEC3)
    dotNet.addEventHandler button5 "click" (setEC4)
    dotNet.addEventHandler button6 "click" (setEC5)

--显示窗体  功能类似createdialog
MainForm.show()
```

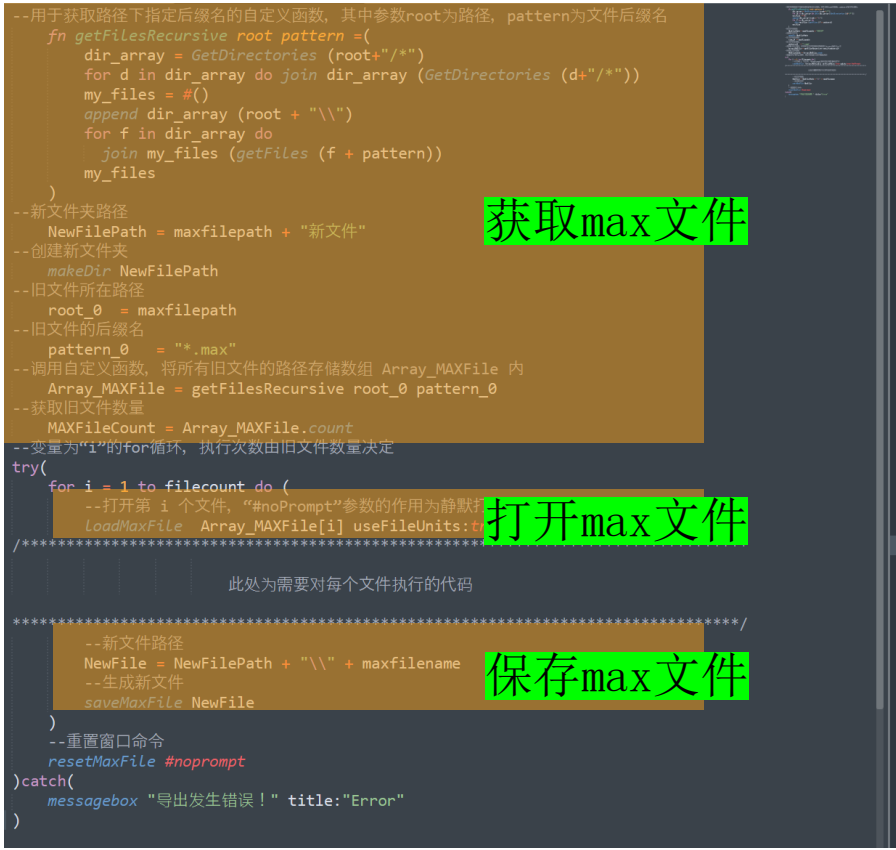


通用批处理框架

通用批处理框架


批处理框架
获取需要批处理的max文件
开始for循环，次数为max文件的数量
关闭max文件
打开max文件
<重复执行的操作代码>
保存或另存max文件

通用批处理框架




```
--用于获取路径下指定后缀名的自定义函数，其中参数root为路径，pattern为文件后缀名
    fn getFilesRecursive root pattern =(
        dir_array = GetDirectories (root+"/*")
        for d in dir_array do join dir_array (GetDirectories (d+"/*"))
        my_files = #()
        append dir_array (root + "\\")
        for f in dir_array do
          join my_files (getFiles (f + pattern))
        my_files
    )
--新文件夹路径
    NewFilePath = maxfilepath + "新文件"
--创建新文件夹
    makeDir NewFilePath
--旧文件所在路径
    root_0  = maxfilepath
--旧文件的后缀名
    pattern_0   = "*.max"
--调用自定义函数，将所有旧文件的路径存储数组 Array_MAXFile 内
    Array_MAXFile = getFilesRecursive root_0 pattern_0
--获取旧文件数量
    MAXFileCount = Array_MAXFile.count
--变量为"i"的for循环，执行次数由旧文件数量决定
try(
    for i = 1 to filecount do (
        --打开第 i 个文件，"#noPrompt"参数的作用为静默打
        loadMaxFile  Array_MAXFile[i] useFileUnits:tr
/*****************************************************

                    此处为需要对每个文件执行的代码

*****************************************************/
        --新文件路径
        NewFile = NewFilePath + "\\" + maxfilename
        --生成新文件
        saveMaxFile NewFile
    )
    --重置窗口命令
    resetMaxFile #noprompt
)catch(
    messagebox "导出发生错误！" title:"Error"
)
```

```
    fn getFilesRecursive root pattern =(
        dir_array = GetDirectories (root+"/*")
        for d in dir_array do join dir_array (GetDirectories (d+"/*"))
        my_files = #()
        append dir_array (root + "\\")
        for f in dir_array do
          join my_files (getFiles (f + pattern))
        my_files
    )
    NewFilePath = maxfilepath + "新文件"
    makeDir NewFilePath
    root_0  = maxfilepath
    pattern_0   = "*.max"
    Array_MAXFile = getFilesRecursive root_0 pattern_0
    MAXFileCount = Array_MAXFile.count
try(
    for i = 1 to filecount do (
        loadMaxFile  Array_MAXFile[i] useFileUnits:true quiet:true #noPrompt
/*****************************************************************************

                    此处为需要对每个文件执行的代码

*****************************************************************************/
        NewFile = NewFilePath + "\\" + maxfilename
        saveMaxFile NewFile
    )
    resetMaxFile #noprompt
)catch(messagebox "导出发生错误！" title:"Error")
```

通用批处理框架(案例1：批量重命名)

- 情况说明：有一批max文件，其中所有Bone骨骼的名字都是默认的，现需要对骨骼名字进行修改
- 需求说明：对所有的Bone骨骼命名进行修改，在“Bone”后面加一个下划线“_”
  - 比如“Bone001”要修改为“Bone_001”
- 伪代码
  - 套用批处理框架
  - 接着获取文件内所有的Bone骨骼，将其存到数组内，以供之后调用
  - 将字符“Bone”替换为“Bone_”


```
--for循环的第二种写法，详细语法可以查询官方文档
--变量为“j”的for循环，其循环次数为max文件内“对象”的个数
--where循环，如果当前对象为“BoneGeometry”类型，即所谓的“Bone骨骼”类型，那么就收集到数组“Array_Bone”内
    Array_Bone = for j in objects where (classof j == BoneGeometry) collect j

--获取骨骼数量
    BoneCount = Array_Bone.count

--变量为“K”的for循环，循环次数为骨骼的根数
    for k = 1 to BoneCount do(
        --数组中第“K”个骨骼的名字
            oldname = Array_Bone[k].name

        -- 返回字符串中从第5个到最后一个字符
            oldnameCount = oldname.count
            name1 = substring oldname 5 oldnameCount

        --设置新的骨骼名字
            Array_Bone[k].name = "Bone_" + name1
    )
```

通用批处理框架(案例2：批量SkinWrap)

- 情况说明：有一批已经完成了绑定的角色Avatar文件，每个部件都是一个单独的mesh。由于某些原因，模型同学对原模型的布线进行了修改，动作同学需要对模型重新进行蒙皮，测试发现使用skinwrap可以满足需求，但是部件很多，手动skinwrap的效率非常低

- 需求说明：将修改布线后的模型重新进行绑定

- 前置条件：
  - 除了后缀名不同，【修改后的FBX模型文件命名】与【max文件】命名一致，且都在同一个文件夹内
  - 修改后的mesh命名与原mesh命名相对比，后缀增加了字符“_new”

通用批处理框架(案例2：批量SkinWrap)

- 伪代码
  - 先套用批处理框架
  - 打开max文件，得到所有的mesh，把他存储到变量A中，以便后续调用
  - “合并导入”对应的FBX文件
  - for循环，循环次数为变量A的数量，即mesh的数量，设循环变量为“k”（
    - 获取第k个mesh的名字为“X”，那么对应新mesh的名字就是“X_new”
    - 根据新的mesh命名，选中它，并添加skinwrap修改器
    - 在skinwrap修改器中添加第“k”个mesh，勾选“权重到所有顶点”选项
    - 执行skinwrap命令
    - 删除第k个mesh
  - ）

通用批处理框架(案例2：批量SkinWrap)

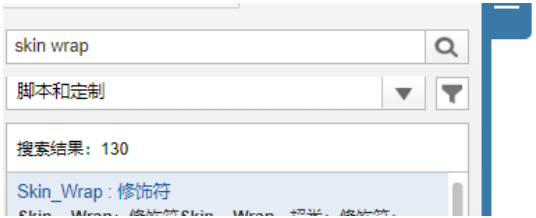
- 编写思路
  - ```
    --选择新模型
    select $body_new

    --切换到修改器模型
    max modify mode

    --添加skinwrap修改器
    modPanel.addModToSelection (Skin_Wrap ()) ui:on

    --勾选“权重到所有顶点”选项
    $.modifiers[#Skin_Wrap].weightAllVerts = on


    --执行skinwrap命令
    $.modifiers[#Skin_Wrap].meshDeformOps.convertToSkin on

    --选择并删除旧模型
    select $body
    max delete
    ```
  - 

    “参数”卷展栏

    ```
    <Skin_Wrap>。meshList ArrayParameter默认值：#() -- 节点数组
    ```

    获取/设置网格列表节点数组。对应于“参数”卷展栏中的节点列表。

    ```
    --添加旧模型到列表中
    oldmesh=$body
    $.modifiers[#Skin_Wrap].meshList = #(oldmesh)
    ```
    ```

通用批处理框架(案例2：批量SkinWrap)

- 编写思路
  -
```
--选择新模型
select $body_new

--切换到修改器模型
max modify mode

--添加skinwrap修改器
modPanel.addModToSelection (Skin_Wrap ()) ui:on

--勾选“权重到所有顶点”选项
$.modifiers[#Skin_Wrap].weightAllVerts = on

--添加旧模型到列表中
oldmesh=$body
$.modifiers[#Skin_Wrap].meshList = #(oldmesh)

--执行skinwrap命令
$.modifiers[#Skin_Wrap].meshDeformOps.convertToSkin on

--选择并删除旧模型
select $body
max delete
```

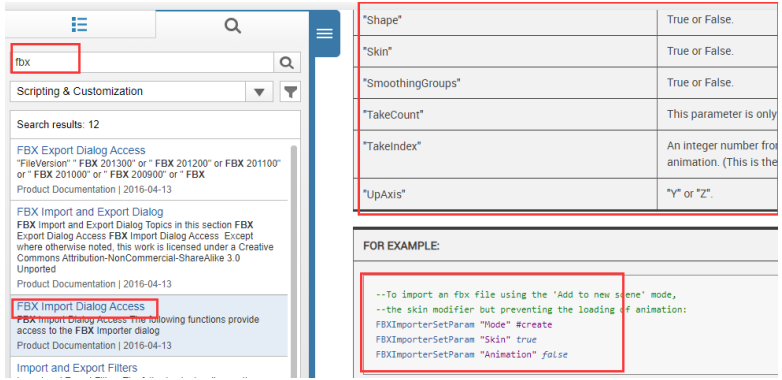
    - 套用批处理框架
    - 打开max文件，获取所有mesh，存储到变量A中，以便后续调用
    - 合并导入对应的FBX文件
    - for循环，循环次数为变量A的数量，即mesh的数量，设循环变量为“k”（
      - 获取第i个mesh的命名，从而得到新的mesh命名为XXX_new
      - 根据新的mesh命名，选中它，并添加skinwrap修改器
      - 在skinwrap修改器中添加第“k”个mesh，勾选“权重到所有顶点”选项
      - 执行skinwrap命令
      - 删除第i个mesh
    -)

  -
```
Mod_All=for i in objects where(classof i==Editable_Poly or classof i==Editable_mesh or classof i==PolyMeshObject) collect i
```

通用批处理框架(案例2：批量SkinWrap)

- 编写思路
  - 


```
AlembicImport.ImportToRoot = false --parent to an Alembic Container
importFile (GetDir #export +"\\alembictest_hdm5_max.abc") #noPrompt using:AlembicImport
--RESULT: The loaded Cone matches the original animation
```

  - 


  - ```
fn replaceSub str1 str2 str3 = (
    while ((i=findstring str1 str2) != undefined) do
    (
        str1 = replace str1 i str2.count str3
)
    return str1
)
```
```

通用批处理框架(案例2：批量SkinWrap)

- 编写思路

```
--自定义函数，用于替换字符串
fn replaceSub str1 str2 str3 = (
    while ((i=findstring str1 str2) != undefined) do
    (
        str1 = replace str1 i str2.count str3
    )
    return str1
)

--获取所有mesh，存储到变量Mod_All中
Mod_All=for i in objects where(classof i==Editable_Poly or classof i==Editable_mesh or classof i==PolyMeshObject) collect i

--导入fbx文件
    NowMaxFBXfilepath = maxfilepath
    NowMAXFileName = maxfilename
    replaceSub NowMAXFileName ".max" ".fbx"
    --NowMAXFileName变量的值经过了自定义函数的处理，变成了对应fbx文件的名字
        importFile NowMAXFileName #noPrompt

for k = 1 to Mod_All.count do(

    --选择新模型
        NewMesh = getnodebyname (Mod_All[k].name+"_new")
        select NewMesh


    max modify mode
    modPanel.addModToSelection (Skin_Wrap ()) ui:on
    $.modifiers[#Skin_Wrap].weightAllVerts = on

    --添加旧模型到列表中
        oldmesh = Mod_All[k]
        $.modifiers[#Skin_Wrap].meshList = #(oldmesh)

    $.modifiers[#Skin_Wrap].meshDeformOps.convertToSkin on

    --选择并删除旧模型
    select oldmesh
    max delete
)
```

总结

总结

- **刷书**
  - 把学习资料当作字典来使用，但前提是要先“刷”一遍，只有都看过、用过，知道有这个东西存在，遇到某些情况的时候，才知道怎么去使用它，否则都不知道有这样一个东西存在，学习起来就比较困难
- **搜索引擎大法**
  - 网上的资料很多，或许要实现的功能已经有人写过了，拿过来稍微改一改就能够使用
- **“列大纲”**
  - 实现一个功能之前，将功能先逐步拆解，提前规划好要实现的子功能，以“伪代码”的形式罗列出来。

总结

- **“学习其他脚本”**
  - 除了“刷书”，查看其他脚本代码的时候，或许会收获一些之前没有见过的东西，可以收集起来。
  - 有时候写的代码不能够一次性运行成功，就需要多敲、多运行，只有这样才能加深对代码的理解。
- **“脚本库”**
  - 当编程经验积累到一定程度的时候，相信已经写了很多脚本，在之后生产中遇到的一些需求，可能只要拿之前的代码改一改就能用。
  - 自己写过的脚本、网上下载的脚本都可以收集起来，形成一个个人的脚本库，就好像“通用批处理框架”一样，可以直接套用。
  - 一些觉得好用 或者 觉得之后可能会用到的自定义函数，都可以用一些笔记软件存起来。

Thanks For Watching!

AUTODESK UNIVERSITY