MAXscript\_功能助记

目录

[1. 实用功能 2](#_Toc466903174)

[1.1 方法 2](#_Toc466903175)

[1.1.1 批量获取目录下文件路径 返回数组 2](#_Toc466903176)

[1.1.2 从一个多边形的已选中的边提取一个SpLine 2](#_Toc466903177)

[1.1.3 把一个对象的点坐标写进文档中 2](#_Toc466903178)

[1.2 关于子级的选择 3](#_Toc466903179)

[1.2.1 已选中的子物体 ---> 数组 3](#_Toc466903180)

[1.2.2 数组 ---> 选中子物体 3](#_Toc466903181)

[1.2.3 其它Mesh 子对象选择方法 4](#_Toc466903182)

[1.3 应用功能 5](#_Toc466903183)

[1.3.1 加密脚本 5](#_Toc466903184)

[1.4 C# & MaxScript 5](#_Toc466903185)

[3.1.1 常用命令 5](#_Toc466903186)

[3.1.2 示例： 6](#_Toc466903187)

[3.1.3 Maxscript里正则表达式 6](#_Toc466903188)

[2. 相关技巧 7](#_Toc466903189)

[2.1 自动提示补全方法 7](#_Toc466903190)

[2.2 占位 11](#_Toc466903191)

[3. 牛B案例 11](#_Toc466903192)

[3.1 自己写的 11](#_Toc466903193)

[3.2.1 skyBox 11](#_Toc466903194)

[3.2.2 CATIA\_V5\_To\_FBX 12](#_Toc466903195)

[3.2.3 处理倾斜模型的工具，包括清理小物体和重建拟合三角面 13](#_Toc466903196)

[3.2 牛人写的 18](#_Toc466903197)

[3.2.1 占位 18](#_Toc466903198)

[4. 深度好文 18](#_Toc466903199)

[4.1 占位 18](#_Toc466903200)

[4.2 占位 18](#_Toc466903201)

[4.3 占位 18](#_Toc466903202)

[5. 待整理的资料 18](#_Toc466903203)

[5.1 占位 18](#_Toc466903204)

[5.2 占位 18](#_Toc466903205)

[5.3 占位 18](#_Toc466903206)

[5.4 占位 18](#_Toc466903207)

# 实用功能

## 方法

### 批量获取目录下文件路径 返回数组

--遍历数组，拿到每个件的路径进行操作

filePaths =getFiles @"E:\test\5-1-3.fbm\\*.fbx"

for f in files do(

--打开 loadMAXFile f

--导入 importFile f #noPrompt

format "% \n" f

)

### 从一个多边形的已选中的边提取一个SpLine

--poly

myPoly= $

--拿到线（点ID集）

myEdes = polyOp.getEdgeSelection myPoly

--把poly转成SPline

polyOp.createShape myPoly myEdes smooth:False name:"myShape" node:unsupplied

### 把一个对象的点坐标写进文档中

filePath = "C:\\Users\\Administrator\\Desktop\\NewTest.txt"

if CreateFile filePath == undefined then

messagebox "无法创建文档"

else fileStrea =

CreateFile filePath

pointCount = numKnots myShape 1

for i=1 to pointCount do

(

format "%\n" (getKnotPoint myShape 1 i) to:fileStrea

)

close fileStrea

)

messagebox "OK"

## 关于子级的选择

### 已选中的子物体 ---> 数组

通过已经选中的点集 获取一个点的数组

<mesh,poly>.selectedVerts -------------> <mesh,poly>.verts 获取包括全部点的数组

getVertSelection <mesh,poly>

Eg:

a = $.selectedVerts

b = getVertSelection $

Bv:

#verts(186,208,404 : $Editable\_Mesh:Tile @ [2876.225098,-4576.110352,227.571991])

#{186, 208, 404}

通过已经选中的线集 获取一个线的数组

<mesh,poly>.selectedEdges -------------> <mesh,poly>.edges 获取包括全部线的数组

getEdgeSelection <mesh,poly>

Eg:

a = $.selectedEdges

b = getEdgeSelection $

Bv:

#edges(908,1612,1614,2906,2907,3684:$Editable\_Mesh:Tile@ 2876.225098,-4576.110352,227.571991])

#{908, 1612, 1614, 2906..2907, 3684}

通过已经选中的面集 获取一个面的数组

<mesh,poly>.selectedFaces -------------> <mesh,poly>.faces 获取包括全部面的数组

getFaceSelection <mesh,poly>

Eg:

a = $.selectedFaces

b = getFaceSelection $

Bv:

#faces(303,2298,2507:$Editable\_Mesh:Tile@ 2876.225098,-4576.110352,227.571991])

#{303, 2298, 2507}

### 数组 ---> 选中子物体

通过一个点数组选中包括的点 ，是getVertSelection $ 的镜像操作

setVertSelection <mesh> <vertlist>

polyOp.setVertSelection <poly> <vertlist>

Eg:

setVertSelection $ a

polyOp.setVertSelection $ a

Bv:

OK

通过一个数组选中包括的线 ，是getEdgeSelection $ 的镜像操作

setEdgeSelection <mesh> <edgelist>

polyOp.setEdgeSelection <Poly poly> <edgelist>

通过一个数组选中包括的面 ，是getFaceSelection $ 的镜像操作

setFaceSelection <mesh> <facelist>

polyOp.setFaceSelection <Poly poly> <facelist>

### 其它Mesh 子对象选择方法

-- 返回一个<BitArray>值，元素为 Mesh 对象里开放边的序号。

meshop.getOpenEdges <Mesh mesh>

-- 返回一个整数，表示 Mesh 对象的面数，等于属性<mesh>.numfaces 的值。

getNumFaces <mesh>

-- 从 Mesh 对象删除指定面，并自动对面进行重新编号。

deleteFace <mesh> <face\_index\_Integer>

-- 返回一个 BitArray 值，元素为 Mesh 对象的顶点序号，这些顶点被参数<facelist>指定的面使用。

meshop.getVertsUsingFace <Mesh mesh> <facelist>

-- 返回一个 BitArray 值，元素为 Mesh 对象的边序号，这些边被参数<facelist>指定的面使用。

meshop.getEdgesUsingFace <Mesh mesh> <facelist>

-- 返回一个 BitArray 值，元素为 Mesh 对象的面序号，这些面被参数<facelist>组成的多

-- 边形所包含。参数 threshold:的默认值为 45 度。如果参数 ignoreVisEdges:为 True，边的可见性被忽略，但参数 threshold:仍有效。

meshop.getPolysUsingFace <Mesh mesh> <facelist> \

ignoreVisEdges:<boolean=False> threshold:<Float=45.>

-- 删除指定面。如果参数 delIsoVerts:为 True，任何单独顶点都将被删除。

meshop.deleteFaces <Mesh mesh> <facelist> delIsoVerts:<boolean=True>

-- 返回一个 BitArray 值，元素为 Mesh 对象的面序号，这些面所在元素的边中至少有一

-- 个面在参数<facelist>所指定的面列表中。如果有指定参数 fence： ，其指定面所在的元素不会被处理。

meshop.getElementsUsingFace <Mesh mesh> <facelist> \

fence:<facelist=unsupplied>

-- 返回一个 BitArray 值，元素为 Mesh 对象的面序号，这些面使用了参数<vertlist>指定

-- 的顶点。

meshop.getFacesUsingVert <Mesh mesh> <vertlist>

-- 返回指定 Mesh 对象的面数

meshop.getNumFaces <Mesh mesh>

-- 返回一个 BitArray 值，元素为 Mesh 对象的顶点序号，这些顶点被参数<facelist>指定

-- 的面使用

meshop.getVertsUsedOnlyByFaces <Mesh mesh> <facelist>

-- 返回一个 BitArray 值，元素为 Mesh 对象的面序号，这些面处在由顶点列表<vertlist>

-- 定义的多边形里。

meshop.getPolysUsingVert <Mesh mesh> <vertlist> \

ignoreVisEdges:<boolean=False> threshold:<Float=45.>

## 应用功能

### 加密脚本

encryptScript "111111111111111.ms"

## C# & MaxScript

### 常用命令

showMethods --显示方法

showproperties --显示方法

showevents --显示事件

dotNet.showConstructors --显示构造函数

### 示例：

C#

public class TestClassObject

{

string Name;

public TestClassObject(string thisName)

{

Name = thisName;

}

public string GetLongName()

{

return ("You can call me " + Name + "!");

}

public int Getsss(int es)

{

return es \* 3;

}

}

MaxScript

dotnet.loadAssembly (@"D:\test\testClasss.dll")

ccc = dotnetclass "testClass.TestClassObject"

ThisTestObject = dotNetObject ccc "ddddddd"

ThisTestObject.GetLongName()

ThisTestObject.Getsss 55

### Maxscript里正则表达式

--Function：Regex

--@param expression - 正则表达式字符串

--@return 返回.NET下的正则表达式对象

fn Regex expression=

(

rgx = dotNetObject "System.Text.RegularExpressions.Regex" expression

return rgx

)

--使用方法举例

R1 = Regex "^(LG){0,1}([YnNnTtLl]||YP)(\d{1,})\S{0,}\_(\d{1,})$"

bResult = R1.isMatch("LGY01\_001")

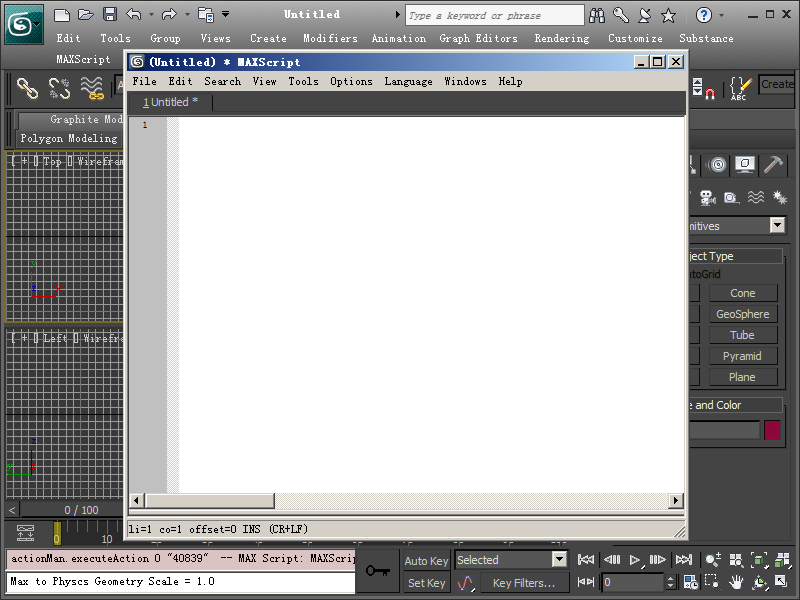
返回的结果是bResult=true

# 相关技巧

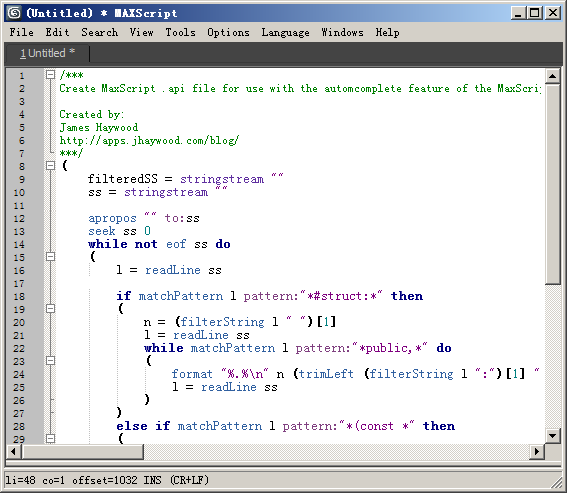
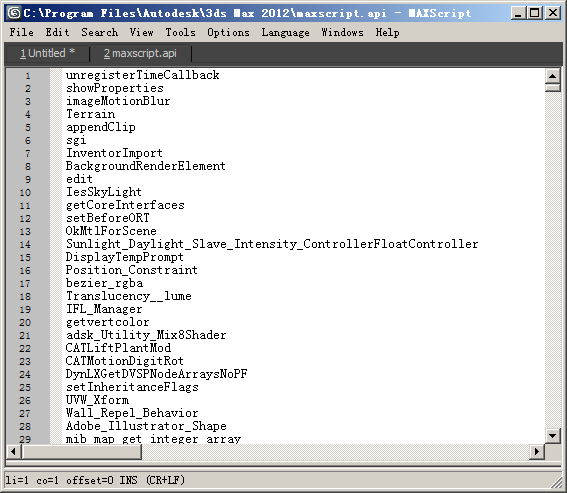
## 自动提示补全方法

步骤：

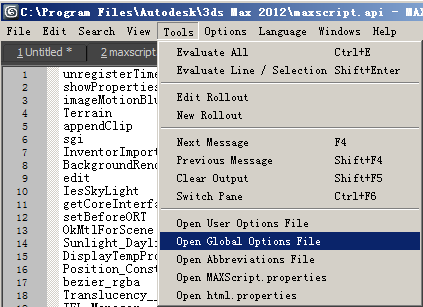
1、启动MAX，新建一个编辑窗口（需要管理员身份）



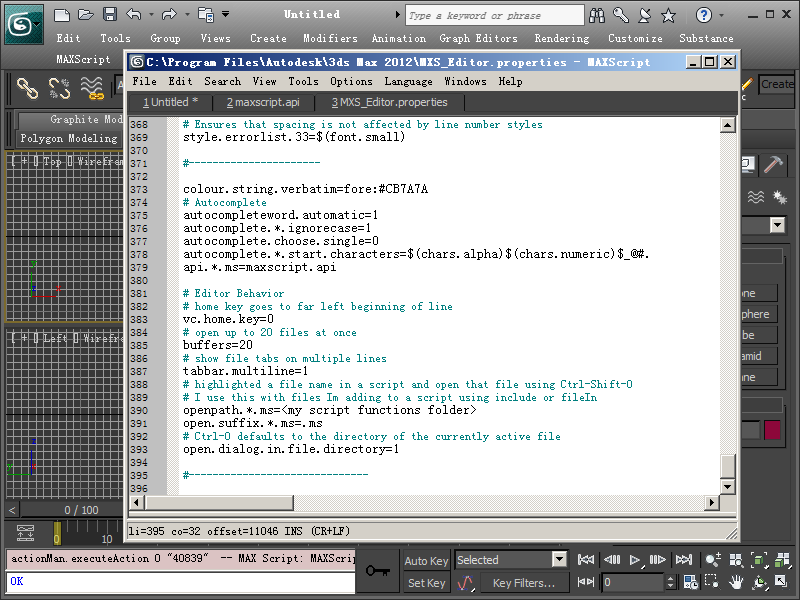
2、把下面那段红色的代码copy到窗口中，按Ctrl+E 执行，会自动生成一个maxscript.api文件，这个文件的位置看标题栏上的路径。（没有什么是一下Ctrl+E解决不了的，如果有，那就两下！）

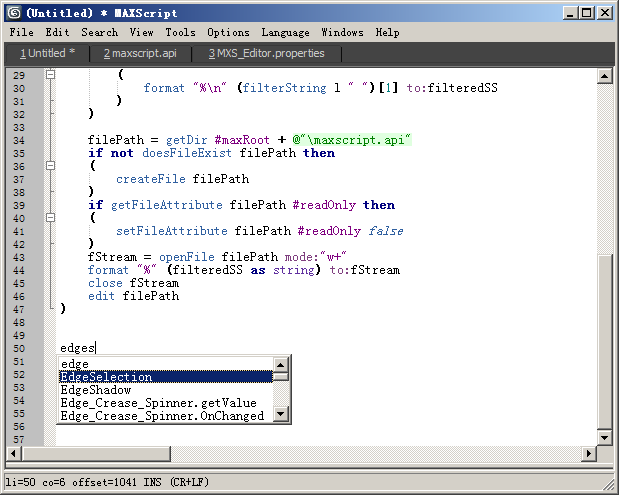
3、打开Tools->OpenGlobalOptionsFile



4、找到文件末尾，把下面蓝色代码贴进来, save!



5、重起MAX，更多内容看图2中第7、8、9条



(

filteredSS = stringstream ""

ss = stringstream ""

apropos "" to:ss

seek ss 0

while not eof ss do

(

l = readLine ss

if matchPattern l pattern:"\*#struct:\*" then

(

n = (filterString l " ")[1]

l = readLine ss

while matchPattern l pattern:"\*public,\*" do

(

format "%.%\n" n (trimLeft (filterString l ":")[1] " ") to:filteredSS

l = readLine ss

)

)

else if matchPattern l pattern:"\*(const \*" then

(

format "%\n" (filterString l " ")[1] to:filteredSS

)

)

filePath = getDir #maxRoot + @"\maxscript.api"

if not doesFileExist filePath then

(

createFile filePath

)

if getFileAttribute filePath #readOnly then

(

setFileAttribute filePath #readOnly false

)

fStream = openFile filePath mode:"w+"

format "%" (filteredSS as string) to:fStream

close fStream

edit filePath

)

#----------------------

colour.string.verbatim=fore:#CB7A7A

# Autocomplete

autocompleteword.automatic=1

autocomplete.\*.ignorecase=1

autocomplete.choose.single=0

autocomplete.\*.start.characters=$(chars.alpha)$(chars.numeric)$\_@#.

api.\*.ms=maxscript.api

# Editor Behavior

# home key goes to far left beginning of line

vc.home.key=0

# open up to 20 files at once

buffers=20

# show file tabs on multiple lines

tabbar.multiline=1

# highlighted a file name in a script and open that file using Ctrl-Shift-O

# I use this with files Im adding to a script using include or fileIn

openpath.\*.ms=<my script functions folder>

open.suffix.\*.ms=.ms

# Ctrl-O defaults to the directory of the currently active file

open.dialog.in.file.directory=1

#------------------------------

## 占位

# 牛B案例

## 自己写的

### skyBox

resetmaxfile #noPrompt;

mapPaht = getOpenFileName types:"JPG|\*.jpg" -->"Data(\*.dat)|\*.dat|Excel(\*.csv)|\*.csv|All|\*.\*|";

savePath = getsavepath caption:"Save Folder";

try(

--> create camera

Targetcamera fov:90 pos:[0,0,0] name:"UP" target:(Targetobject transform:(Matrix3 [1,0,0] [0,1,0] [0,0,1] [0,0,100]));

Targetcamera fov:90 pos:[0,0,0] name:"DN" target:(Targetobject transform:(Matrix3 [1,0,0] [0,1,0] [0,0,1] [0,0,-100]));

Targetcamera fov:90 pos:[0,0,0] name:"RT" target:(Targetobject transform:(Matrix3 [1,0,0] [0,1,0] [0,0,1] [100,0,0]));

Targetcamera fov:90 pos:[0,0,0] name:"BK" target:(Targetobject transform:(Matrix3 [1,0,0] [0,1,0] [0,0,1] [0,-100,0]));

Targetcamera fov:90 pos:[0,0,0] name:"LF" target:(Targetobject transform:(Matrix3 [1,0,0] [0,1,0] [0,0,1] [-100,0,0]));

Targetcamera fov:90 pos:[0,0,0] name:"FR" target:(Targetobject transform:(Matrix3 [1,0,0] [0,1,0] [0,0,1] [0,100,0]));

--> create skyball

ball = Sphere radius:1000 segs:128 pos:[0,0,0] isselected:true;

--> change Normals

convertToMesh ball;

meshop.flipNormals ball ball.faces;

--> UVWMap

max modify mode;

addModifier ball (Uvwmap());

ball.modifiers[1].maptype = 2;

--> material

ball.material = meditMaterials[1];

ss=ball.material;

ss.selfIllumamount = 100;

ss.diffuse = color 255 255 255;

ss.ambient = color 255 255 255;

ss.specular = color 255 255 255;

ss.diffuseMap = Bitmaptexture filename:mapPaht;

--> outPut

render camera:$UP outputSize:[1024,1024] outputFile:(savePath + @"\UP.jpg");

render camera:$DN outputSize:[1024,1024] outputFile:(savePath + @"\DN.jpg");

render camera:$RT outputSize:[1024,1024] outputFile:(savePath + @"\RT.jpg");

render camera:$BK outputSize:[1024,1024] outputFile:(savePath + @"\BK.jpg");

render camera:$LF outputSize:[1024,1024] outputFile:(savePath + @"\LF.jpg");

render camera:$FR outputSize:[1024,1024] outputFile:(savePath + @"\FR.jpg");

resetmaxfile #noPrompt;

messageBox("OK!");

)catch(

resetmaxfile #noPrompt;

messageBox("Failed!");

)

### CATIA\_V5\_To\_FBX

resetmaxfile #noPrompt

rootFolde = getsavepath caption:"DataSourcePath"

saveRoot = getsavepath caption:"TagetPath"

foldes = append #() (rootFolde+"\\")

for itm in foldes do

(

newfoldes = getDirectories (itm + "\*.\*")

join foldes newfoldes

)

files = #()

for i=1 to foldes.count do

(

subPathFBX = getfiles (foldes[i] + "\*.cgr")

join files subPathFBX

subPathTXT = getfiles (foldes[i] + "\*.CATPart")

join files subPathTXT

)

sort files

for i = 1 to files.count do

(

thisFile = files[i]

importFile thisFile #noPrompt

-- max select all --resetName+ID

-- s=1

-- for itm in selection do

-- (

-- itm.name = "reset"+ s as string

-- s = s+1

-- )

newPath = saveRoot + (substring thisFile (rootFolde.count+1) -1)

makeDir (getFilenamePath newPath)

savePath = getFilenamePath newPath + getFilenameFile newPath + ".fbx"

max select all

exportFile savePath #noPrompt selectedonly:true

resetmaxfile #noPrompt

)

messagebox("OK!")

### 处理倾斜模型的工具，包括清理小物体和重建拟合三角面

/\*

删除FBX文件中过小的独立结构

fileDir ： 指定一个文件夹

maxFaceCount ： 指定一个阈值

2016.6.9

\*/

fn cleanSmallObj =

(

messagebox "请备份原文件，删除操作不可逆！"

--初始化

resetMaxFile()

fileDIR = getSavePath caption:"选择打开文件路径:"

filePaths =getFiles (fileDIR + @"\\*.fbx")

for afile in filePaths do

(

--导入一个

importFile afile #noPrompt

--指定变量名

tar = $

--打开 Modify 命令面板

max modify mode

--选中所有的点

subobjectLevel = 1

max Select All

--焊接点集

meshOps.weld $

--获取对象面数

faceCount = getNumFaces tar

--少于这个数量将被删除

delCount = faceCount / 10

subobjectLevel = 3

--结果集合

delFaecs = #()

--遍历面

for f=1 to faceCount do

(

--拿到一个面

aface = #(f)

--通过个这面获取与其相接的面集

tt = meshop.getElementsUsingFace tar aface

--通过面集选中面

tar.selectedFaces = tt

--获取选中面的数量

sel = tar.selectedFaces.count

--判断是否小于参考值，是则加如结果集

if sel < delCount do

append delFaecs f

)

--删除结果集中的面

meshop.deleteFaces tar delFaecs delIsoVerts:True

--进入父级

if subobjectLevel != 0 then subobjectLevel = 0

--输出

newPath = substring afile 1 (findstring afile ".") + "osg"

exportFile newPath #noPrompt

--全选、删除

max select all

delete selection

)

messagebox "OK"

)

/\*

提取轮廓坐标值输出到文档

\*/

fn getLunKuo =

(

try

(

filePath = (getSaveFileName types: "all(\*.\*)") + ".txt"

if classof $ != Editable\_Poly then convertToPoly $ ; subobjectLevel = 3

--拿到线（点ID集）

myEdes = polyOp.getEdgeSelection $

--把poly转成SPline

polyOp.createShape $ myEdes smooth:False name:"myShape" node:unsupplied

myShp =$myShape

if CreateFile filePath == undefined

then messagebox "无法创建文档"

else

(

fileStrea = CreateFile filePath

pointCount = numKnots myShp 1

for i=1 to pointCount do

(

format "%\n" (getKnotPoint myShp 1 i) to:fileStrea

)

close fileStrea

)

messagebox "OK"

)catch(messagebox "有BUG联系BT")

)

fn TheEnd =

(

-- try

-- (

-- if (readLine ( openfile @"D:\aa")) =="aaa"

-- then

-- (

createdialog 'Gvitech.Tools@TB'

-- addrollout 'Gvitech.Tools@TB'

--encryptScript "BT\_tools.ms"

-- )

-- else messagebox ("未授权！")

-- )

-- catch messagebox ("未授权！")

)

fn addBack =

(

filePath = getOpenFileName caption:"打开网格构建文档" types:"\*.txt"

str1 = readLine (openfile filePath)

execute str1

addModifier $ (Uvwmap ())

$.material = meditMaterials[12]

mapPath = getOpenFileName caption:"打开贴图" types:"\*.tif"

meditMaterials[12].diffuseMap = Bitmaptexture fileName:mapPath

actionMan.executeAction 0 "63545"

)

/\*

主界面

\*/

rollout 'Gvitech.Tools@TB' "Gvitech.Tools@TB" width:300 height:340

(

-------------------------------------------------------------------------------

groupBox aboutMe "说明" pos:[10,270] width:275 height:50 enabled:false

label lbl2 "working..." pos:[25,290] width:150 height:20

-------------------------------------------------------------------------------

groupBox classA "classA" pos:[10,10] width:275 height:98

button btn1 "清理小结构" pos:[30,30] width:230 height:25 toolTip:""

button btn2 "提取修补轮廓" pos:[30,65] width:150 height:25 toolTip:""

button btn3 "修补" pos:[193,64] width:65 height:25 toolTip:""

-------------------------------------------------------------------------------

-------------------------------------------------------------------------------

on btn1 pressed do

(

cleanSmallObj()

)--end on

on btn2 pressed do

(

getLunKuo()

)--end on

on btn3 pressed do

(

addBack()

)--end on

)TheEnd()

## 牛人写的

### 占位

# 深度好文

## 占位

## 占位

## 占位

# 待整理的资料

## 占位

## 占位

## 占位

## 占位