

Autodesk 3ds Max

Pythonデモンストレーション

Akira Kudo – akira.kudo@autodesk.com

Developer Consulting Engineer – M&E Sparks Autodesk Developer Network



本日のアジェンダ

1. 概要
2. 何故Pythonなのか？
3. Pythonの3ds Maxでの使用
4. MAXScriptとの比較
5. MaxPlus APIについて
6. UI拡張
7. 機能拡張
8. Q & A

概要

3ds MaxがPythonをサポートすると言う事とは?

A. DLLとしてのPython 2.7.3

3ds Max 2014 ExtensionにPython 2.7.3の標準ディストリビューションがDLLとして存在

B. スクリプトの実行

GUP plug-in (MaxPythonHost.gup)にてMAXScript及び3ds Max command lineからPythonスクリプトを実行

C. 新規3ds Max API

CPython拡張モジュール(MaxPlus)によりSDKをPythonに開放

_MaxPlus.pydより機能開放

```
enum Python_Result
{
    Python_Success = 0,           // method executed successfully
    Python_Path_Error,           // Can't find the python distribution folder
    Python_Init_Error,           // the python initialization method failed
    Python_Script_File_Error,   // the python script file could not be opened for read
    Python_Execute_Error,        // script was not executed successfully
};

Python_Result InitPythonInterpreter();
Python_Result ExecutePythonString(const wchar_t* script,
    const wchar_t* fileName = nullptr, bool clearUndoBuffer = true);
Python_Result ExecutePythonFile(const wchar_t* fileName, bool clearUndoBuffer = true);
const wchar_t* GetLastPythonError();
```

3ds Max Python – 実行方法は?

- MAXScriptからファイルを実行:
`python.executeFile "demoHelloWorld.py"`
- MAXScriptからコードを実行:
`python.execute "print 'hello world!'"`
- コマンドラインからファイルを実行:
`3dsmax.exe -U PythonHost "demoHelloWorld.py"`

3ds Max Pythonの機能

- 組み込みの標準 CPython 2.7.3 64ビット
 - 標準のPythonにてコンパイルされたモジュールとはバイナリー互換
 - CPythonバイナリーは 3ds Max にても動作
 - 例: <http://www.lfd.uci.edu/~gohlke/pythonlibs/>
- APIによる3ds Maxの操作
- MAXScriptとの連携

3ds Max Pythonの機能

- エディター, デバッガー, コンソール, マクロレコーダー
- 制限があるAPI
- マルチスレッド
- 標準のPythonディストリビューション以外のモジュール(PySideとのリリースを検討中)

何故Pythonなのか?

Pythonが選ばれる幾つかの理由

- 既に多くのライブラリが存在
 - 想像を超えた多さ
- 多くのユーザー数とナレッジ
 - StackOverflow.comには250,000の質問と回答
- 再利用し易いスクリプト
- 拡張し易い
 - CPython API, SIP, SWIG, Boost, その他.
- 学習し易い
- その他・・・

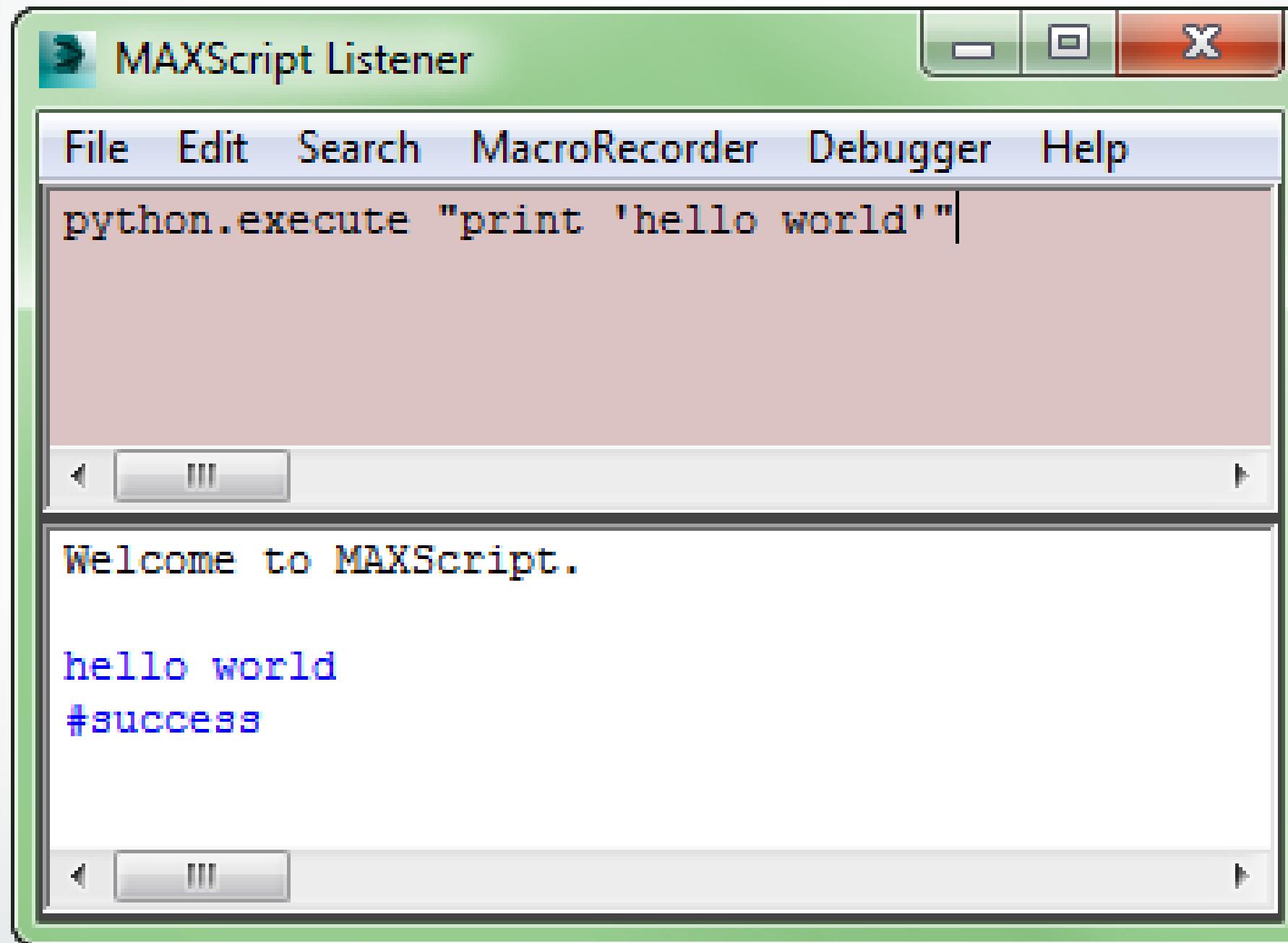
3ds Max Python – 提供しない機能

Pythonにより…

- パイプラインのPythonツール・ライブラリの機能向上が容易
- スイート製品間のツール開発を容易
- ツール開発者やテクニカルディレクターの方々へ
- PythonはMAXScriptに取って代わるものではない
- PythonがMAXScriptより優れている:
 - OSとの作業
 - PyQt 又は PySideによる洗練されたUI
 - NumPyや PyCUDAによるハイパフォーマンスな計算処理
 - インターネット接続: アップロード/ダウンロード
 - その他…

Pythonの3ds Maxでの使用

Hello World



Python <=> MAXScript

MAXScript から Python を実行

- 引数を渡せない
- 実行結果を返せない

Python から MAXScript を実行

- 引数を渡せない
- 実行結果の取得が可能
 - `box = MaxPlus.Core.EvalMAXScript('Box()')`

3ds MaxにてPython はどの様に動作のするか

Python codeが実行されると、Pythonセッションが開始される

3ds Maxの再起動迄、セッションは継続される：

再開する方法は無い

モジュールがロードされると、削除は出来ない
(Pythonの仕様)

Pythonはメインスレッドにて実行

3ds MaxでのPythonの作法

グローバルな名前空間を汚染しない

- 例. モジュールを全てインポートしない

if __name__ == '__main__' を使用

- 全てのファイルがライブラリーになる可能性が有り

設定を強く推奨: sys.argv = [__file__]

- 幾つかのモジュールはこの方法に依存 (例. Unittest等)

3ds Maxはスレッドセーフではない

MAXScript, .NET, 又は C++ を適切に利用

MAXScriptとPythonの比較

MAXScriptによるBent Cylinderの作成

```
1 cyl = cylinder()
2 cyl.Radius = 10
3 cyl.Height = 30
4 bd = bend()
5 bd.Angle = 45
6 addmodifier cyl bd
```

PythonによるBent Cylinderの作成

```
1  """
2   . . . Demonstrates · creating · a · cylinder · and · applying · a · bend · modifier. . .
3  """
4
5  import · MaxPlus
6
7  obj · = · MaxPlus · Factory · CreateGeomObject (MaxPlus · ClassIds · Cylinder)
8  obj · ParameterBlock · Radius · Value · = · 10.0
9  obj · ParameterBlock · Height · Value · = · 30.0
10 node · = · MaxPlus · Factory · CreateNode (obj)
11 mod · = · MaxPlus · Factory · CreateObjectModifier (MaxPlus · ClassIds · Bend)
12 mod · ParameterBlock · BendAngle · Value · = · 45.0
13 node · AddModifier (mod)
14
```

MAXScriptによるライトとカメラの取得

```
1 lights_and_cameras =
2   for o in objects
3     where (iskindof o light or iskindof o camera)
4       collect o
5
6   for x in lights_and_cameras do print x.name
7
```

Pythonによるライトとカメラの取得

```
3     import MaxPlus
4
5     def allNodes(root = MaxPlus.Core.GetRootNode()):
6         for n in root.Children:
7             for n2 in allNodes(n):
8                 yield n
9         yield root
10
11    def lightsAndCameras():
12        for n in allNodes():
13            o = n.BaseObject
14            if o:
15                if MaxPlus.LightObject._CastFrom(o):
16                    yield n
17                elif MaxPlus.CameraObject._CastFrom(o):
18                    yield n
19
20    def main():
21        for n in lightsAndCameras():
22            print n.Name
23
24    if __name__ == '__main__':
25        main()
```

MaxPlus API について

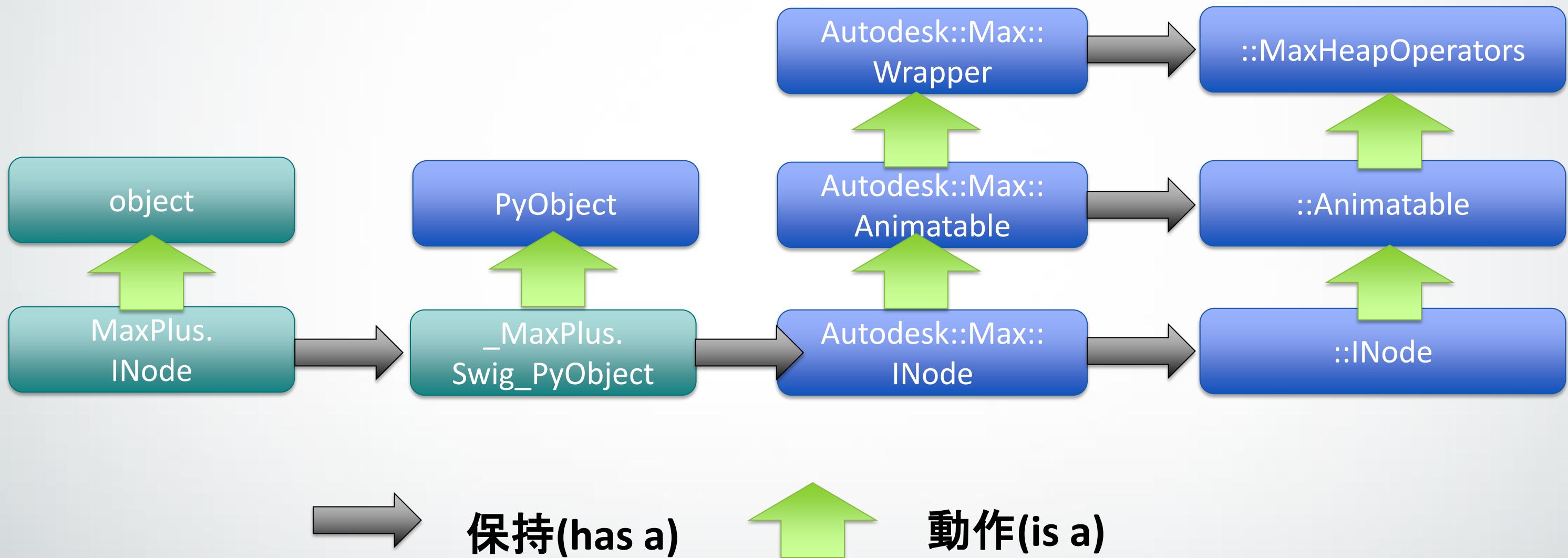
MaxPlus

MaxPlusは、3ds Max SDKのハイレベルオブジェクトで C++ API のラッパー

SWIGを使用しモジュールとして Python に公開

MaxPlus のオブジェクトは、a::MaxHeapOperator へのポインタや参照を保持するラッパークラスから派生

MaxPlus のオブジェクトモデル



Pythonから3ds Maxを操作

モジュール“MaxPlus”による共通機能へのアクセス:

- シーングラフの走査
- オブジェクトの作成
- ファイルのオープン / クローズ
- メッシュの操作
- パラメータの取得 / 設定
- 簡単なレンダリング

help vs dir

- オブジェクトの中身を確認
 - dir() - Python Build-in
 - help()

```
import MaxPlus
help(MaxPlus.Asset)
```

例) help.py

Inspectと__dict__

- Python APIのクラスを抽出

- `import inspect`

```
classes = inspect.getmembers(MaxPlus, inspect.isclass)
```

- `__dict__`

```
totalcnt = 0
for c in classes:
    name = str(c[0])
    membercnt = len(c[1].__dict__)
    totalcnt += membercnt
    api[name] = membercnt
```

例) demoClassesInMaxPlus.py

プロパティのアクセス

- アクセス方法は2つ
 - プロパティ名
 - Get/Set ... ()メソッド
 - リードオンリー
 - 例 ParameterBlock.Name

```
import MaxPlus

m = MaxPlus.Factory.CreateDefaultStdMat()

c = m.Ambient
c = m.Diffuse
c = m.GetAmbient()
c = m.GetDiffuse()

m.Ambient = c
m.Diffuse = c
m.SetAmbient(c)
m.SetDiffuse(c)
```

例) property.py

オブジェクトの操作

- 作成
 - Factory.CreateObject() -> object
 - Factory.CreateGeomObject() -> GeomObject
 - Factory.CreateNode() -> node

```
obj = MaxPlus.Factory.CreateGeomObject(MaxPlus.ClassIds.Sphere)
node = MaxPlus.Factory.CreateNode(obj)
```

例) demoMakeInstances.py

オブジェクトの操作

- 変更
 - パラメータブロック
 - モディファイナー

```
obj = MaxPlus.Factory.CreateGeomObject(MaxPlus.ClassIds.Cylinder)
obj.ParameterBlock.Radius.Value = 10.0
obj.ParameterBlock.Height.Value = 30.0
node = MaxPlus.Factory.CreateNode(obj)
mod = MaxPlus.Factory.CreateObjectModifier(MaxPlus.ClassIds.Bend)
mod.ParameterBlock.BendAngle.Value = 45.0
node.AddModifier(mod)
```

- メッシュ

```
geom = MaxPlus.Factory.CreateGeomObject(MaxPlus.ClassIds.TriMeshGeometry)
tri = MaxPlus.TriObject._CastFrom(geom)
mesh = tri.GetMesh()
```

例) demoBentCylinder.py demoMeshAndCPV.py

マテリアルの操作

- 作成

- Factory
 - 例 m = MaxPlus.Factory.CreateDefaultStdMat()
- ClassId

```
def GeneratePlugins(sid, cls):  
    for cd in MaxPlus.PluginManager.GetClassList().Classes:  
        if cd.SuperClassId == sid:  
            anim = MaxPlus.Factory.CreateAnimatable(sid, cd.ClassId, False)  
            if anim:  
                inst = cls._CastFrom(anim)  
                if inst:  
                    yield inst  
  
def CreateMaterials():  
    materials = GeneratePlugins(MaxPlus.SuperClassIds.Material, MaxPlus.Mtl)  
    materialList = list(materials)  
    numMaterials = len(materialList)  
    #for m in materialList:  
    #    print m  
    print "%d materials" % numMaterials  
    return materialList
```

例) demoMaterials.py

マテリアルの操作

- アサイン
 - 例 node.Material = m
- マテリアルエディタ
 - オープン
 - 例 MaxPlus.MaterialEditor.OpenMtlDlg(MtlDlgMode.basic)
 - スロットへの設定(2方法)
 - 例 MaxPlus.MaterialManager.PutMtlToMtlEditor(m,index)
 - 例 MaxPlus.MaterialEditor.SetSlot(index,m)
 - アクティブスロット
 - 例 MaxPlus.MaterialEditor.SetActiveSlot(index)

例) demoMaterials.py

ビットマップの操作

- 作成
 - Factory
 - 例 `b = MaxPlus.Factory.CreateBitmap()`
 - Storage作成
 - 例 `s = MaxPlus.Factory.CreateStorage(BitmapTypes.BMM_TRUE_32)`
 - BitmapInfo
 - `Storage::GetBitmapInfo()`にて取得
 - `SetWidth() / SetHeight()`等
 - ピクセル領域の確保
 - `BitmapStorage.Allocate()`

例) `createbitmap.py`

ビットマップの操作

- ピクセル設定
 - BitmapStorage.PutPixel()

例

```
for x in range(720):
    for y in range(620):
        storage.PutPixel(x, y, (MaxPlus.Color64(64000-(x*(64000/720)), (y*(64000/720)), 64000, 64000)))
```

- Storage設定
 - Bitmap.SetStorage()

例) createbitmap.py

Playボタンの操作

- Animation
 - MaxPlus.Animation

例

```
anim = MaxPlus.Animation  
PrintInterval(anim.GetAnimRange())
```

- 範囲設定
 - インターバル設定
 - 各フレーム160ticks

例

```
newFrames = 200 * 160  
newRange = MaxPlus.Interval(0, newFrames)  
anim.SetRange(newRange)
```

例) demoAnimation.py

Playボタンの操作

- 各種設定
 - Auto Keyボタン
 - Animation.SetAnimateButtonState()
 - Startボタン
 - Animation.StartPlayback()
 - PlayBack Loopボタン
 - Animation.SetPlaybackLoop()
 - Time Slider
 - Animation.SetTime()
 - 再生スピード
 - Animation.SetPlaybackSpeed()

例) demoAnimation.py

アセットの操作

- 作成

例

```
asset = MaxPlus.AssetManager.CreateAsset ("maps\\fir.tga")
print asset.GetResolvedFileName ()
```

- 取得

- アットマネージャより

- 例 MaxPlus.AssetManager.GetAssets()

例) demoAssets.py

プラグインファイルの情報

- PluginManager
 - .PluginDllsにプラグインのリスト

例

```
'''  
    Demonstrates using the PluginManager to extract information about loaded plugins.  
import MaxPlus  
  
# List all plug-in dlls  
pluginsDlls = MaxPlus.PluginManager.PluginDlls  
print "Total PluginDlls: {0}\n".format(MaxPlus.PluginManager.GetNumPluginDlls())  
for pd in pluginsDlls:  
    print "PluginDll:", pd.FilePath  
    print "Description:", pd.Description  
    print "Loaded:", pd.Loaded  
    print "NumClasses:", pd.NumClasses  
    for cd in pd.Classes:  
        if cd:  
            print "  ", cd.GetClassName()
```

例) demoOutputPluginClasses.py

シーングラフの走査

- 必要な情報

- Core.GetRootNode()
- INode.Parent
- INode.Children

```
...
    Creates a simple text representation of the scene graph
...
import MaxPlus

def outputNode(n, indent = ''):
    print indent, n.Name
    for c in n.Children:
        outputNode(c, indent + '--')

if __name__ == '__main__':
    outputNode(MaxPlus.Core.GetRootNode())
```

例) demoSceneGraph.py

シングラフの走査

- セレクションの情報
 - SelectionManager.Nodes
 - セレクションリストのイテレータ
 - 直接アクセス出来ない
 - 例 MaxPlus.SelectionManager.Nodes[3]
 - コレクションを作成
 - list(MaxPlus.SelectionManager.Nodes)
 - tuple(MaxPlus.SelectionManager.Nodes)

例

```
nodes = list(MaxPlus.SelectionManager.Nodes)
print nodes[0].Name
```

シーンファイルの操作

- FileManager
 - オープン
 - .Open()
 - セーブ
 - .Save()
 - ファイルパスの取得
 - .GetFileNameAndPath()

例 fm = MaxPlus.FileManager
fm.Save(MaxPlus.PathManager.GetTempDir() + r"\test.max")
print fm.GetFileNameAndPath()

Maxシステムパスの操作

- MaxPlus.PathManagerにて
 - 取得
 - GetScriptsDir(),GetTempDir()等
 - 設定
 - SetScriptsDir(),SetTempDir()等
 - ファイルパスの取得
 - .GetFileNameAndPath()

```
import MaxPlus
print 'The Scripts directory is', MaxPlus.PathManager.GetScriptsDir()
print 'The Temp directory is', MaxPlus.PathManager.GetTempDir()
```

アクションとメニューの操作

- アクション
 - 作成
 - MaxPlus.ActionFactory.Create()
 - 実行
 - ActionItem.Execute()

例

```
import MaxPlus

def doSomething():
    print "I sleep all night and I work all day!"

action = MaxPlus.ActionFactory.Create('Do something', 'Python demo', doSomething)
action.Execute()
```

例) createAction.py

アクションとメニューの操作

- メニュー
 - メニュービルダー作成
 - MaxPlus.MenuBuilder()
 - アイテム/セパレータ登録
 - MenuBuilder.AddItem() / MenuBuilder.AddSeparator()
 - 作成
 - MenuBuilder.Create()

例

```
mb = MaxPlus.MenuBuilder(MenuName)
mb.AddItem(action)
mb.AddSeparator()
menu = mb.Create(MaxPlus.MenuManager.GetMainMenu())
```

例) demoMenu.py

レンダリング

- 実行
 - MaxPlus.RenderExecute.QuickRender()
- MaxPlus.RenderSettings
 - 出力設定
 - SetOutputFile()
 - SetSaveFile()
 - イメージの幅・高さ設定
 - SetWidth() / SetHeight()
 - Pre/Post render script
 - SetPreScriptFile() / SetUsePreScript() 等
 - その他(グローバルオプション設定)

例) demoRender.py

ビューポート

- MaxPlus.ViewportManager

- アクティブビュー
 - Get/SetActiveViewport()
- 走査
 - ViewportManager.Viewports
 - Lenは使用不可(GetNumViewports())

```
| for view in MaxPlus.ViewportManager.Viewports:  
|     viewType = view.GetViewType()  
|     print "%d - %s - %s (%d)" % (index, MaxPlus.ViewportManager.getViewportLabel(index), viewType.GetKey(viewType), viewType)
```

UI擴張

PythonによるUI作成

Win32

- `ctypes.windll`
- `PyWin32`

.NET

- MAXScript
- `PyDotNet`
- `ctypes`

Qt

- PySide / PyQt

Tcl/Tk

- 標準であるが、時代遅れ

wxWidgets

- サポートされない

Qtとは?

- 有名なopen-source cross-platform C++ UI toolkit
- Pythonとのバインディングは2つ: PySide / PyQt
 - 3rd パーティーによる拡張
- Windowsでは Win32のラッパー
- 3ds Max 2014 は 4.8.2を使用
- 3ds Maxは Qt DLLが存在
 - QtGui4.dll とQtGui4_Ad.dll

SIPによるQtのラッパー

[Google code](#)にてBlur Studioが3ds Max 2014 で動作するバ
イナリバージョンを公開

- 1月15日まで
- Google code がシャットダウン

PySide

Mayaが使用

[Kelly Michelのブログ](#)よりバイナリがダウンロード可能

多くのお客様よりサポート要請

PyQtと類似

- ライセンスの違いだけ

3ds MaxにおけるQt UI

- ガーベージコレクションからウィジェットを保護
- QApplication.instance()を使用
 - UIファイルのインポート前にインスタンスを取得
- 3ds Max 親ウインドウに
 - 又は 3ds Maxの子ウインドウに
- QApplication._exec()を呼び出さない
 - ウィジェットを表示するのみ
- ウィンドウのフォーカスを取得するにはアクセラレータを無効に
 - MaxPlus.CUI.DisableAccelerators()

Qt UI を3ds Maxの親ウインドウに

QtWinMigrate.QWinWidget を使用

- ウィンドウを親ウインドウに
- PyQt の標準ではない
- 3ds Max 2014互換のモジュールがBlurより公開

ドッキングには

- C++にてCUIFrameを作成し、CtypesによりAPIを実行
 - Christopher Digginsのブログ にて公開予定

機能拡張

3ds MaxにおけるPythonの拡張

拡張方法は幾つか存在

- CPython 拡張モジュール
 - CPython APIを介して手法
 - SWIG, SIP, Boost, Cython, その他.
- ctypes モジュール
- PyWin32 モジュール(3rd パーティ)
- Python for .NET (3rd パーティ)

C++によるDLLの作成

```
1 #pragma once
2
3 #include <istdplug.h>
4 #include <maxtypes.h>
5 #include <utilapi.h>
6 #include <maxscript/maxscript.h>
7 #include <maxscript/util/listener.h>
8
9 // prevent name mangling
10 extern "C"
11 {
12     DllExport void PrintString(const TCHAR* text);
13     DllExport HWND GetMaxHWND();
14     DllExport AnimHandle GetAnimHandle(Animatable* anim);
15     DllExport Animatable* GetAnimFromHandle(AnimHandle h);
16     DllExport const MCHAR* GetNodeName(Animatable* node);
17     DllExport void MakeDockable(HWND h);
18 }
19
```

DLLは3ds Maxルートフォルダに配置

PythonからカスタムDLLの呼び出し

```
6 import ctypes
7
8 def main():
9     ... # Create a wchar_ptr object
10    text = ctypes.c_wchar_p(u'Hello AU!\n')
11
12    ... # Load the au_demo.dll
13    au_demo = ctypes.windll.au_demo
14
15    ... # Call a function from the DLL
16    au_demo.PrintString(text)
17
18 if __name__ == '__main__':
19    main()
```

御質問は御座いますか？

E-mail akira.kudo@autodesk.com にお気軽に問い合わせ下さい

