

Python  
+  
NoSQL

in Animations

太極  
digimax

官順暉

# **What can you expect?**

---

One man's journey of learning  
python in animation industry.



<http://www.flickr.com/photos/nidalm/3523404908/>

Conf. Room #2: Sphinx Documentation

Conf. Room #1: Pyramid Pro

- In Gaming, animation, VFX industry?
- Coding in Autodesk Maya?



# NoSQL? CouchDB? MongoDB?





**官順暉 (Drake)**

**R&D Manager, Digimax Inc.**

**@drakeguan**

批踢踢活動部年度鉅獻

# 批踢踢大學

創意徵稿

windgodvocxl

投稿請至 Ptt\_Monopoly

踢踢總部

又去\●\跳啦  
出! \ / \$ 3000

亞門町

●●●\ ト几●刁  
\$ 0204

台北 1 0 1

呼 \ = \$ 5000

← START!  
● >  
< □

時間：2010.xx.xx~xx.xx

歡迎來到 批踢踢實業坊 目前有 [140118] 位鄉民在 批踢踢大亨 Online

請輸入代號，或以 guest 參觀，或以 new 註冊：

觀光局邀您分享遊記、相片，福斯汽車、百萬獎勵讓您玩遍台灣！<http://ppt.cc/w4vV>



# acm International Collegiate Programming Contest

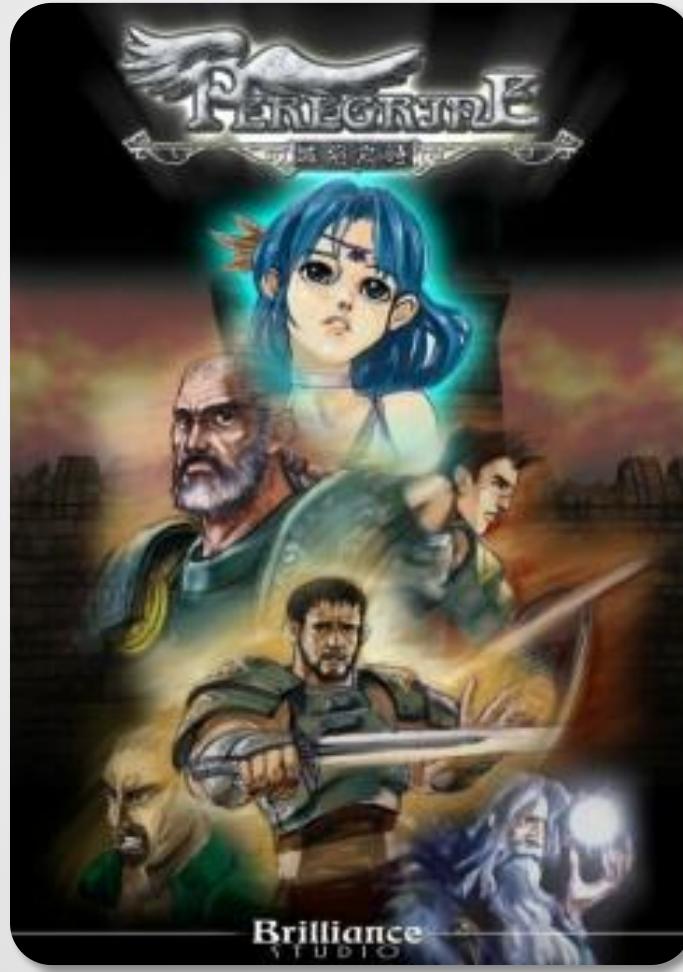
IBM®

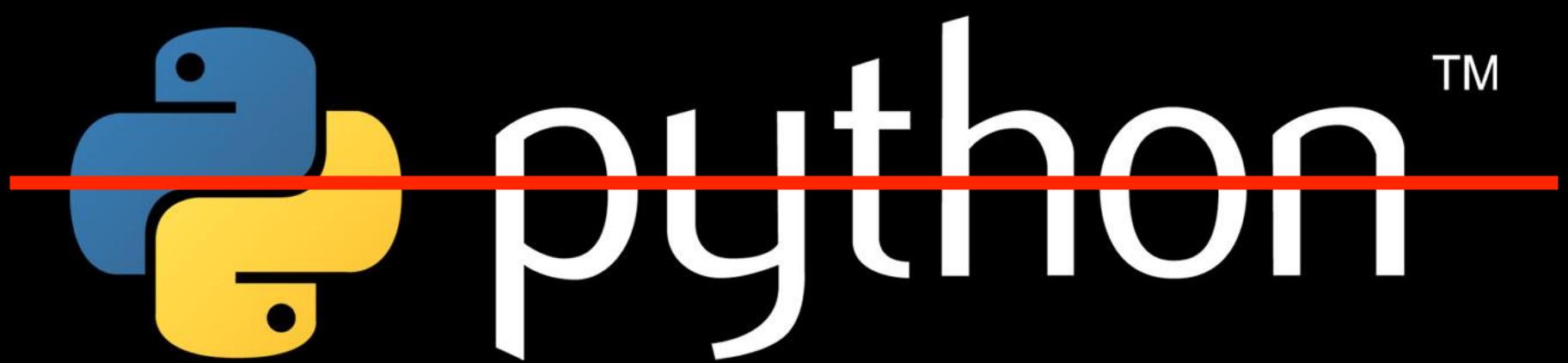
event  
sponsor



<http://www.nakuz.com/bbs/viewthread.php?action=printable&tid=4352983&sid=3XQfMO>

# Peregrine / Brilliance Studio



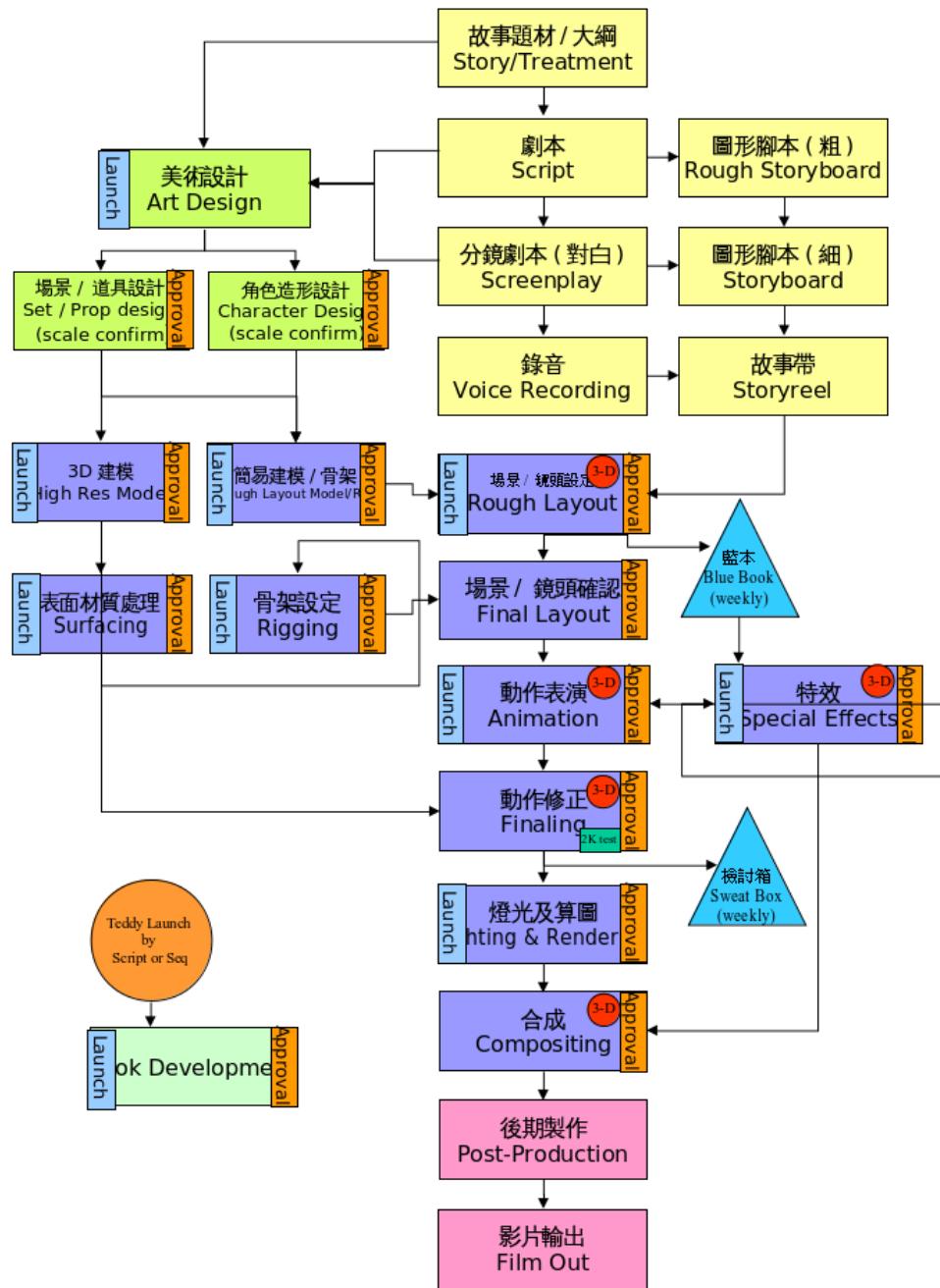


- 太極影音 (Digimax Inc.)
- Animation Studio / Post-production
- ~100 employees
- artist : engineer = 70% : 10%

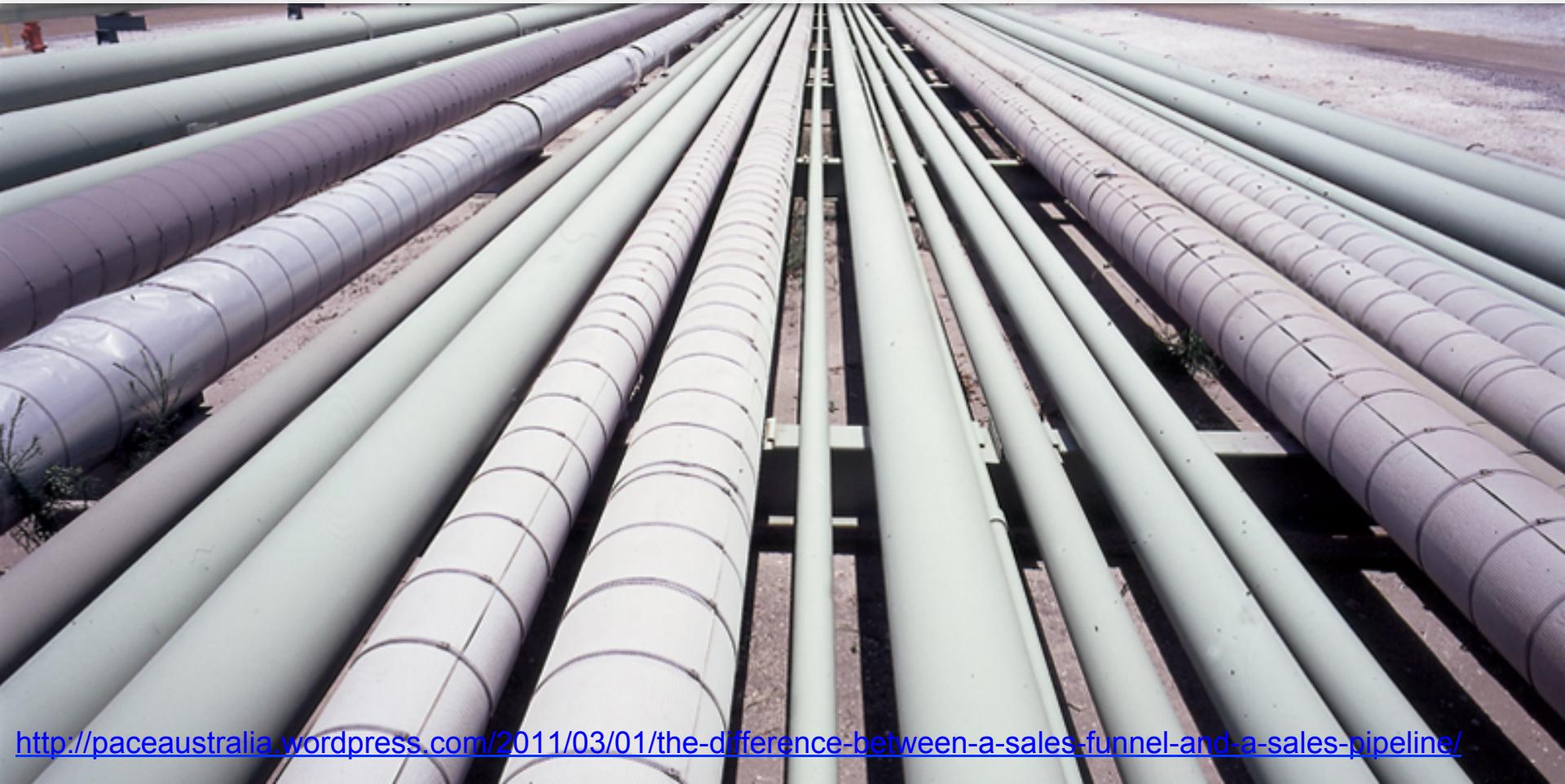


# Digimax Production





# **Rookies! Let's build our animation pipeline!**



太極 3D 動畫開發系統 - 子計劃之一 - 製作掌標系統子計劃書  
Digimax 3D Production Systems - No. 1 - Production Tracking System

## 太極 3D 動畫開發系統

(Digimax 3D Film Production Systems)

### 子計劃之一：製作掌標

(No.1 - Production Tracking System)

Version: 1.0  
Release Date: 2005/02/01

太極 3D 動畫開發系統 - 子計劃之二 - Maya Scene I/O Tool  
Digimax 3D Production Systems - No. 2 - Maya Scene I/O Tool

## 太極 3D 動畫開發系統

(Digimax 3D Film Production Systems)

### 子計劃之二：Maya Scene I/O

(No.2 - Maya Scene I/O Tool & Tagging System)

Version: 1.0  
Release Date: 2005/02/01

太極 3D 動畫開發系統 - 子計劃之五 - Maya Animation Baking System  
Digimax 3D Production Systems - No. 5 - Maya Animation Baking System

## 太極 3D 動畫開發系統計劃書

(Digimax 3D Film Production Systems)

### 子計劃之二：Maya Animation Baking System

(No.5 - Maya Animation Baking System)

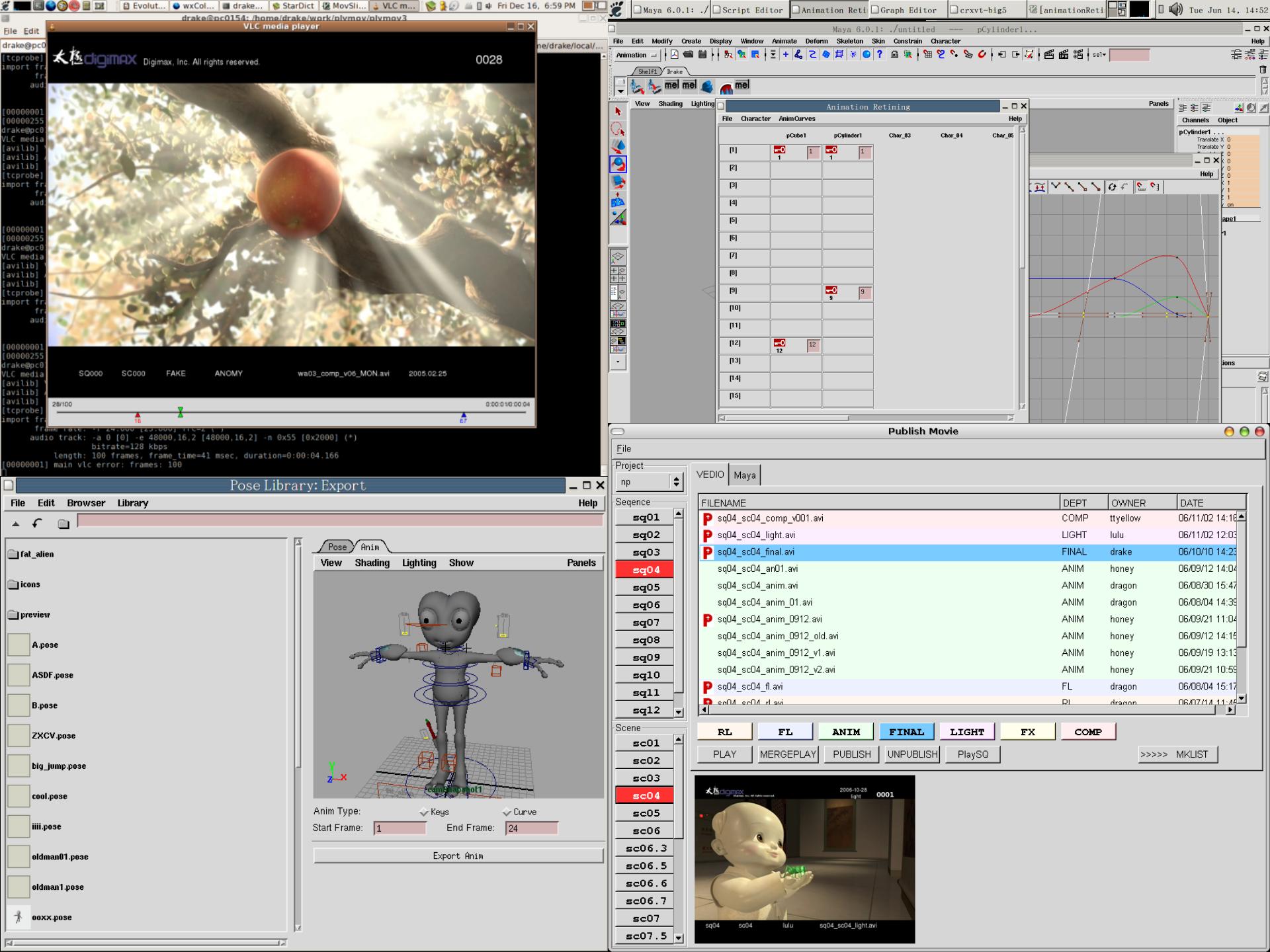
Version: 1.0  
Release Date: 2005/04/22

## 太極 3D 動畫開發系統計劃書

(Digimax 3D Film Production Systems)

### 子計劃之三：影片製作與播放工具

(No.3 - Make Movie Tools & Sequence Playback Tools)



# A project for coding in Python

**Joel on Software**

# User Interface Design For Programmers

by Joel Spolsky

Wednesday, October 24, 2001

## **Chapter 1: Controlling Your Environment Makes You Happy**

Most of the hard core C++ programmers I know *hate* user interface programming. This surprises me, because I find UI programming to be quintessentially easy, straightforward, and fun.

It's *easy* because you usually don't need algorithms more sophisticated than how to center one rectangle in another. It's *straightforward* because when you make a mistake, you immediately see it and can correct it. It's *fun*, because the results of your work are immediately visible. You feel like you are sculpting the program directly.

<http://www.joelonsoftware.com/uibook/fog0000000249.html>

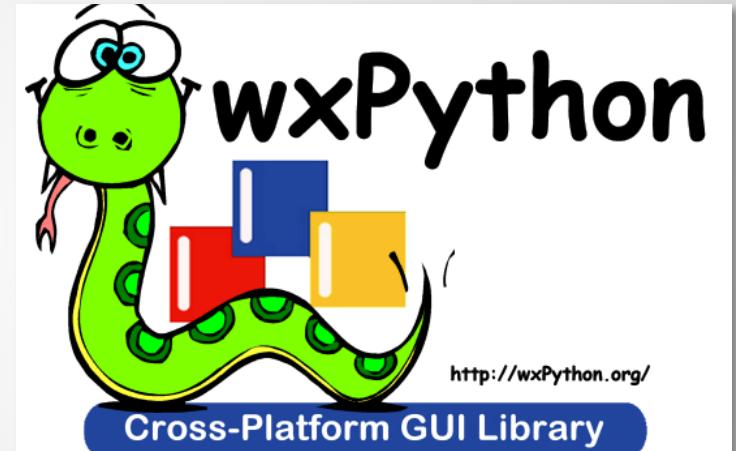
# PyGTK, PyQt or WxPython?



<http://www.pygtk.org/>



<http://aralbalkan.com/1675/>



<http://commons.wikimedia.org/wiki/File:WxPython-logo.png>

- wxPython Overview
  - Recent Additions/Upda
- Frames and Dialogs
  - AUI\_DockingWindowMgt
  - AUI\_MDI
  - Dialog
  - Frame
  - MDIWindows
  - MiniFrame
  - Wizard
- Common Dialogs
  - AboutBox**
  - ColourDialog
  - DirDialog
  - FileDialog
  - FindReplaceDialog
  - FontDialog
  - MessageDialog
  - MultiChoiceDialog
  - PageSetupDialog
  - PrintDialog
  - ProgressDialog
  - SingleChoiceDialog
  - TextEntryDialog
- More Dialogs
  - ImageBrowser
  - ScrolledMessageDialog
- Core Windows/Controls
- "Book" Controls
- Custom Controls
- Advanced Generic Wid
- More Windows/Control
- Window Layout
- Process and Events
- Clipboard and DnD
- Using Images
- Miscellaneous

Active Version:  Original  Modified  Save Changes  Delete Modified

```

1  import wx
2  from wx.lib.wordwrap import wordwrap
3
4
5
6
7  class TestPanel(wx.Panel):
8      def __init__(self, parent, log):
9          self.log = log
10         wx.Panel.__init__(self, parent, -1)
11
12         b = wx.Button(self, -1, "Show a wx.AboutBox", (50,50))
13         self.Bind(wx.EVT_BUTTON, self.OnButton, b)
14
15
16     def OnButton(self, evt):
17         # First we create and fill the info object
18         info = wx.AboutDialogInfo()
19         info.Name = "Hello World"
20         info.Version = "1.2.3"
21         info.Copyright = "(C) 2006 Programmers and Coders Everywhere"
22         info.Description = wordwrap(
23             "A \"hello world\" program is a software program that prints out \"Hello world!\" on a display device. It is used in many introductory tutorials for teaching a programming language."
24
25             "\n\nSuch a program is typically one of the simplest programs possible in a computer language. A \"hello world\" program is a \"sanity test\" to make sure that a language's compiler, development environment, and run-time environment are correctly installed."
26
27
28
29
30
31         info.WebSite = ("http://en.wikipedia.org/wiki/Hello_world")
32         info.Developers = [ "Joe Programmer",
33                            "Jane Coder",
34                            "Vippy the Mascot" ]
35
36
37         info.License = wordwrap(licenseText, 500, wx.ClientDC(self))
38
39         # Then we call wx.AboutBox giving it that info object
40         wx.AboutBox(info)
41
42
43
44

```

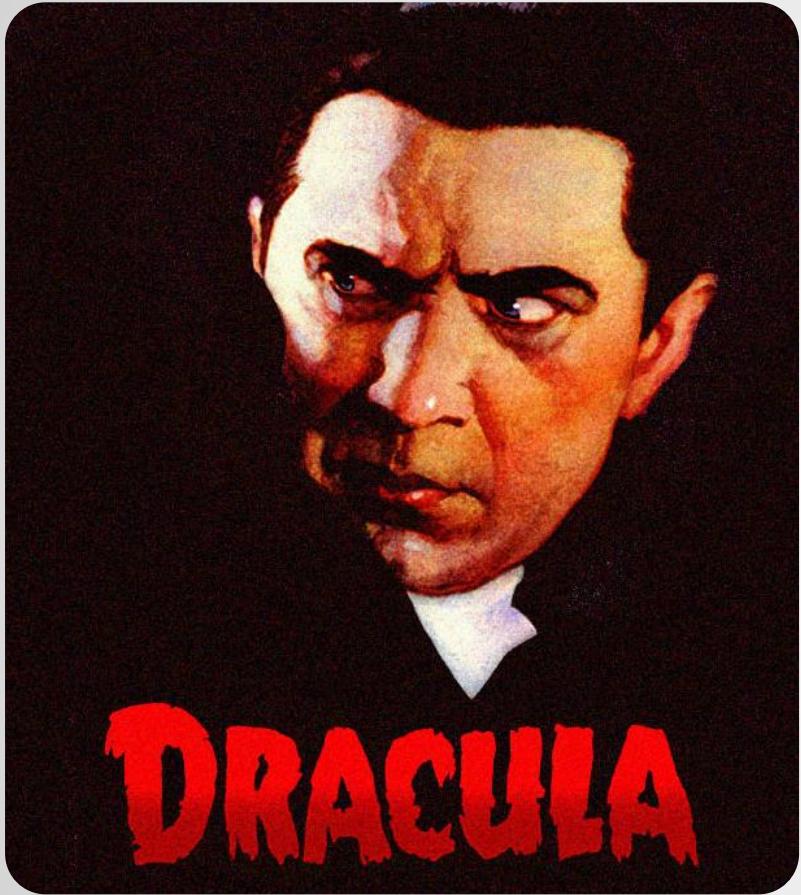


The screenshot shows the Dracula 2.5 software interface with the following details:

- Top Bar:** Dracula 2.5 [Scene], 檔案, 設定, Programs, 幫助.
- Toolbar:** 新增 'Scene' program, 新增 'MkMov' program.
- Project Panel:** npfmt
- Sequence List:** sq010, sq020, sq900, sq910
- Scene List:** 進行中的 Scene 檔案 (170 項)

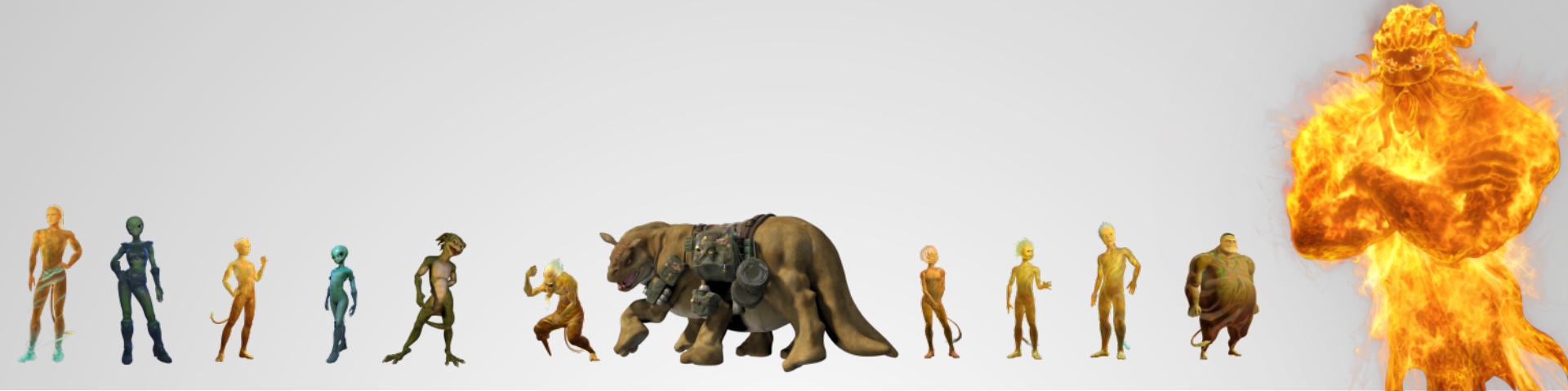
Stage	Layer	鏡頭資訊, 路徑縮寫	所有人	日期
anim		/ycliao/npfmt_sq010_sh0010_anim_0018.ma	ycliao	2012/03/23 16:39:30
anim		/stella_chen/npfmt_sq010_sh0010_anim_0018.ma	stella_chen	2012/03/13 11:26:35
anim_final		/bruno/npfmt_sq010_sh0010_anim_final_ma.ma	bruno	2012/03/07 09:36:01
anim_final		/sharon/npfmt_sq010_sh0010_anim_final_ma.ma	sharon	2012/02/23 11:00:44
anim		/damon/npfmt_sq010_sh0010_anim_0019.ma	damon	2012/02/01 16:49:00
anim_bake		/npfmt_sq010_sh0010_anim_bake_ma.ma	jeter_chuang	2012/01/16 17:48:49
anim_final		/npfmt_sq010_sh0010_anim_final_ma.ma	stone	2012/01/16 17:47:18
anim_bake_ma		/stone/npfmt_sq010_sh0010_anim_0019.ma	stone	2012/01/16 17:28:33
anim_final_ma		/stone/npfmt_sq010_sh0010_anim_0018.ma	stone	2012/01/16 16:41:41
anim		/stone/npfmt_sq010_sh0010_anim_0017.ma	stone	2012/01/13 17:32:08
anim		/stone/npfmt_sq010_sh0010_anim_0016.ma	stone	2012/01/13 16:56:39
anim		/bruno/npfmt_sq010_sh0010_anim_0015.ma	bruno	2012/01/12 19:39:20
anim		/bruno/npfmt_sq010_sh0010_anim_0013.ma	bruno	2012/01/12 18:38:26
anim		/stone/npfmt_sq010_sh0010_anim_0014.ma	stone	2012/01/10 14:29:55
anim		/stone/npfmt_sq010_sh0010_anim_0013.ma	stone	2012/01/10 13:53:50
anim		/stone/npfmt_sq010_sh0010_anim_0012.ma	stone	2012/01/05 10:33:18
anim		/stone/npfmt_sq010_sh0010_anim_0011.ma	stone	2012/01/05 10:01:05
anim		/rainmaker/npfmt_sq010_sh0010_rebuild.ma	rainmaker	2012/01/03 16:20:23
anim_0008		/rainmaker/npfmt_sq010_sh0010_anim_0008.ma	rainmaker	2012/01/03 16:05:16
- Scene List:** 已發佈的 Scene 檔案 (6 項已發佈 scene 檔案)

Stage	Layer	鏡頭資訊, 路徑縮寫	所有人	發佈日期
rl	final_ma	/bruno/npfmt_sq101_sh0010_rl_0005.ma	bruno	2011/12/12 18:56:44
fl	final_ma	/jeter_chuang/npfmt_sq101_sh0010_fl_0004.ma	jeter_chuang	2011/12/16 09:53:44
fl	bake_ma	/timothy/npfmt_sq101_sh0010_fl_0006.ma	timothy	2012/01/07 01:35:44
anim	final_ma	/stone/npfmt_sq010_sh0010_anim_0018.ma	stone	2012/01/16 17:47:48
anim	bake_ma	/stone/npfmt_sq010_sh0010_anim_0019.ma	stone	2012/01/16 17:48:48
sim		/aging_huang/npfmt_sq010_sh0010_sim_bake.ma	aging_huang	2012/01/04 10:33:44
- Review Notes:**
  - ekijo@2012/1/20,light (120120 review by bruno)
    - ▲選用燭火閃爍較明顯的版本。請調整燭火閃動的 timing，盡量錯開，不要一致。
    - ▲盡可能做出燭火閃動的感覺，而不要整片方塊的光閃動。
    - ▲麻煩請算exr圖檔並發佈avi於dracula上。
    - ▲之後於3D化階段，再討論風吹動而水面上霧氣移動的相關製作方案。
  - kathryn\_lee@2012/1/30,light (20120130, reviewed by Bruno)
    - \*最左側山腳下的霧氣請加濃，更明顯一些。
    - \*請確認光宅左側石柱樹葉叢周邊圖層問題，確認有對準。
    - \*中景房屋裡的亮光請調大明暗之間的差距。
  - ekijo@2012/2/1,light (20120201 review by bruno)
    - ▲中景右1窗內燭火閃動的明暗程度請降低一點點。之後請算exr圖檔並發佈avi於dracula上
  - ekijo@2012/2/3,comp (20120202 review by bruno)
    - matte painting:
      - ▲全部已調整完畢。請算exr圖檔並發佈avi於dracula上。
  - ekijo@2012/2/7,comp (20120207 review by bruno)
    - ▲請小新協助將matte painting上的水波紋去除，但保留大塊亮面。
    - ▲請將湖面在遠noise製作出來的水波紋寬度減少一倍。
  - ekijo@2012/2/13,light (20120209 review by bruno)
    - ▲請算exr圖檔並發佈avi於dracula上。
  - ekijo@2012/2/14,comp (20120210 review by bruno)
    - 水波調小版本ok.
  - ekijo@2012/3/13,comp (20120212 review by bruno/tom)
    - 增加備註到目前的scene中



<http://misswargoenglish.wikispaces.com/Dracula>

- Version control for artists
- Meaningful color/layout
- All-in-one >> separated
- GUI + multithreading



# Let's talk about animation





Autodesk Maya 2011 Hotfix 3 x64: /home/alf/work/auto rig/body\_test/body\_rig\_01.ma\* --- spineend\_joint...

File Edit Modify Create Display Window Assets Animate Geometry Cache Create Deformers Edit Deformers Skeleton Skin Constrain Character Muscle Help

Animation DGTOOL ModelTools RiggingTools TextureTools Custom alf\_dgtool alf\_rig alf\_rig2 General Curves Surfaces Polygons Subdivs Deformation Animation Dynamics

View Shading Lighting Show Renderer Panels

**dgPSDUI - rigging**

create/edit pose deformer edit pose reader mirror pose import / expo

1. Choose Geo: man\_body <<>

2. PoseDeformer: poseDeformer1 Edit-Vtx-Mem

3. Pose Name: L\_clavicle\_up 0. CREATE NEW

4. Start Sculpt Mesh. Color:  Del Sculpt Mesh

5. Choose influences. Auto Choose selected

r\_leg\_joint3  
r\_ball\_joint  
spine\_joint1  
spine\_joint2  
spine\_joint3  
spine\_joint4  
spine\_joint5  
spine\_joint6  
chest\_joint  
**L\_clavicle\_joint**  
L\_arm\_bind\_joint1  
L\_arm\_bind\_joint2  
L\_arm\_bind\_joint3  
L\_elbow\_bind\_joint1

Filter: \*  
current selected influences: 1

6. Create/Edit Pose and Readers

advanced options  
Based on Orig Shape

select ctls, and record pose

Rename Delete Mirror  
Export Import Auto-Mem Edit-Mem  
blendMode: RBF-LUFactorize deformSpace: joint-space  
avgPoseSepRBF: 180.000

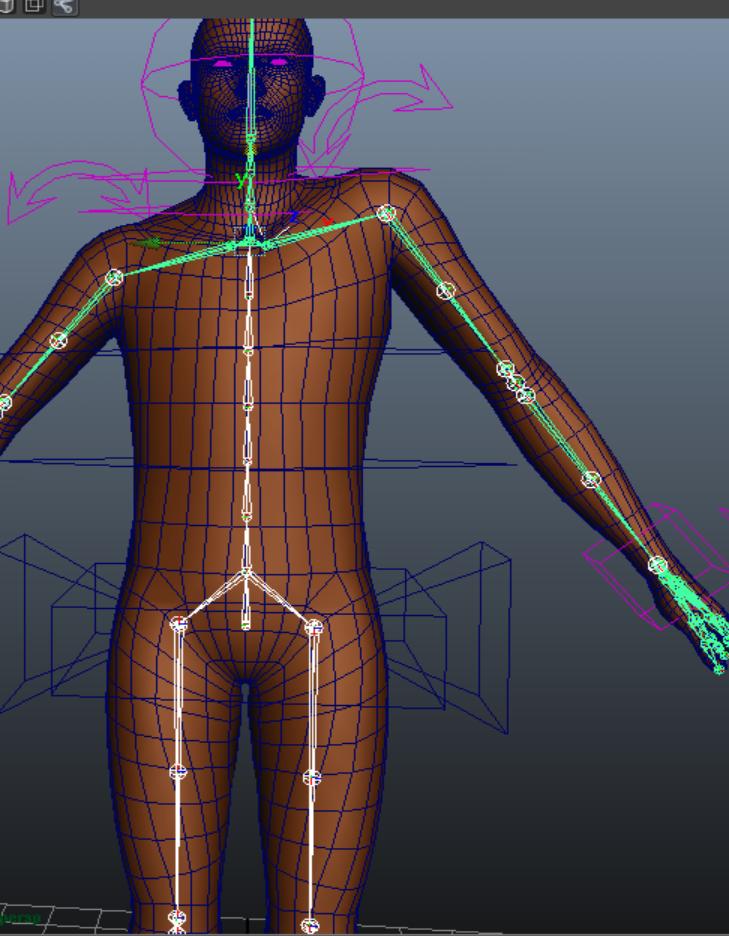
**Channel Box / Layer Editor**

**spineend\_joint . . .**

Translate X 0.66  
Translate Y 0  
Translate Z 0  
Rotate X 0  
Rotate Y 0  
Rotate Z 0  
Scale X 1  
Scale Y 1  
Scale Z 1  
Visibility on  
Radius 0.1

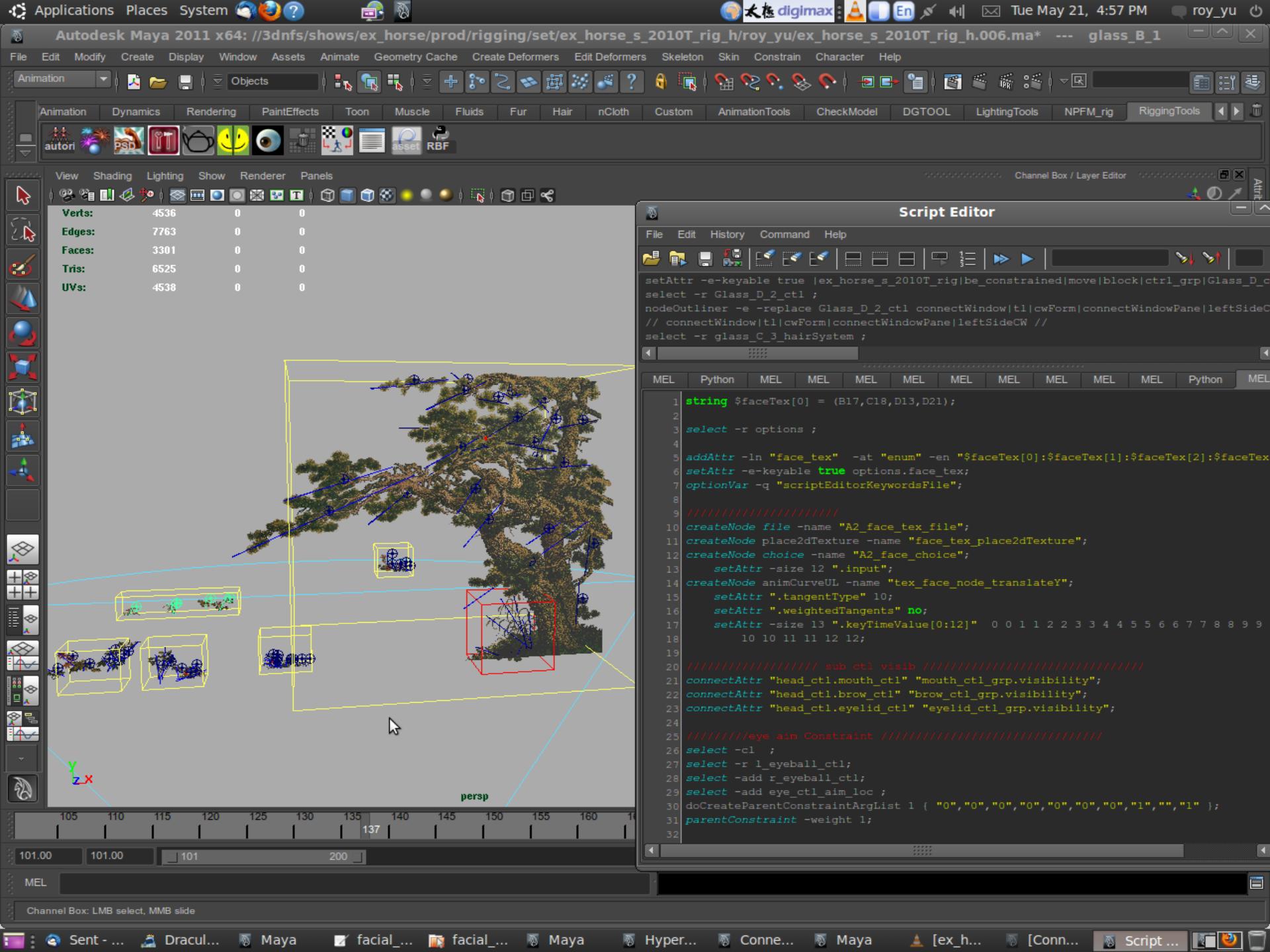
**INPUTS**  
spineend\_joint\_scale  
spine\_curve\_scale  
spine\_curve\_info  
spine\_scale\_fix  
**OUTPUTS**  
bindPose1

Display Render Anim  
Layers Options Help



14 16 18 20 22 24 1.00 24.00 48.00 No Anim Layer No Character Set

[Inbox - Mozilla Thund... [body\_test - File Brow... [\*Unsaved Document ... Maya dgPSDUI - rigging



## Example [\[edit\]](#)

This is an example of a script which copies a selected object through its path:

```
// animated duplicates/instances script
proc animatedDuplication (int $rangeStart, int $rangeEnd, int $numOfDuplicates, int $duplicateOrInstance)
{
    int $range_start = $rangeStart;
    int $range_end = $rangeEnd;
    int $num_of_duplicates = $numOfDuplicates;
    int $step_size = ($range_end - $range_start) / $num_of_duplicates;
    int $i = 0;
    int $temp;

    currentTime $range_start;      // set to range start

    string $selectedObjects[];    // to store selected objects
    $selectedObjects = `ls -sl`; // store selected objects
    select $selectedObjects;

    while ($i <= $num_of_duplicates)
    {
        $temp = $range_start + ($step_size * $i);
        currentTime ($temp);
        // selected the objects to duplicate or instance
        select $selectedObjects;
        if($duplicateOrInstance == 0)
        {
            duplicate;
        }
        else
        {
            instance;
        }
        $i++;
    }
}
```



```
// Usage example:
//  duplicate the current selection 5 times --
//  evenly distributed between frame 1 and 240
animatedDuplication(1, 240, 5, 0);
```

Long name (short name)	Argument types	Properties
<b>-long (-l)</b>		C
	Return full path names for Dag objects. By default the shortest unique name is returned.	
<b>-shortNames (-sn)</b>		C
	Return short attribute names. By default long attribute names are returned.	
<b>-head (-hd)</b>	int	C
	This flag specifies the maximum number of elements to be returned from the beginning of the list of items. Note: each type flag will return at most this many items so if multiple type flags are specified then the number of items returned can be greater than this amount.	
<b>-tail (-tl)</b>	int	C
	This flag specifies the maximum number of elements to be returned from the end of the list of items. Note: each type flag will return at most this many items so if multiple type flags are specified then the number of items returned can be greater than this amount.	
<b>-type (-typ)</b>	string	C M
	List all objects of the specified type. This flag can appear multiple times on the command line. Note: the type passed to this flag is the same type name returned from the -showType flag. Note: some selection items in Maya do not have a specific object/data type associated with them and will return "untyped" when listed with this flag. This flag cannot be used in conjunction with the -exactType flag.	
<b>-exactType (-et)</b>	string	C M
	List all objects of the specified type, but <b>not</b> objects that are descendants of that type. This flag can appear multiple times on the command line. Note: the type passed to this flag is the same type name returned from the -showType flag. This flag cannot be used in conjunction with the -type flag.	
<b>-showType (-st)</b>		C
	List the type of each object after its name.	
<b>-objectsOnly (-o)</b>		C
	When this flag is set only object names will be returned and components/attributes will be ignored.	
<b>-selection (-sl)</b>		C
	List objects that are currently selected.	
<b>-live (-lv)</b>		C
	List objects that are currently live.	
<b>-hilite (-hl)</b>		C
	List objects that are currently hilited for component selection.	
<b>-preSelectHilite (-psh)</b>		C

Go to: [Synopsis](#). [Return value](#). [Flags](#). [MEL examples](#).

## Synopsis

```
ls [-allPaths] [-assemblies] [-cameras] [-containers] [-dagObjects]
[-dependencyNodes] [-exactType string] [-flatten] [-geometry] [-]
[-head int] [-hilite] [-intermediateObjects] [-invisible] [-leaf]
[-live] [-lockedNodes] [-long] [-materials] [-noIntermediate] [-]
[-objectsOnly] [-partitions] [-persistentNodes] [-planes]
[-preSelectHilite] [-readonly] [-recursive boolean] [-referenced]
[-references] [-renderGlobals] [-renderQualities] [-renderResolu
[-renderSetups] [-selection] [-sets] [-shapes] [-shortNames] [-s]
[-tail int] [-templated] [-textures] [-transforms] [-type string
[-undeletable] [-untemplated] [-visible] [object [object...]]]
```

ls is undoable, NOT queryable, and NOT editable.

The **ls** command returns the names (and optionally the type names) of objects in the scene. A common use of **ls** is to filter or match objects based on their name (using wildcards) or type. By default **ls** will match any object in the scene but it can also be used to filter or objects when used in conjunction with the **-selection** flag. If type names are requested, in **showType** flag, they will be interleaved with object names so the result will be pairs of values. Internal nodes (for example itemFilter nodes) are typically filtered so that only some are returned. However, using a wildcard will cause all the nodes matching the wild card to be returned, including internal nodes. For example, **ls \*** will list all nodes whether internal or not. In **relativeNames** mode, the **ls** command will return names *relative* to the current namespace. For more details, please refer to the **-relative** flag in the **namespace** command.

## Return value

*string[]*

## Flags

[allPaths](#), [assemblies](#), [cameras](#), [containers](#), [dagObjects](#), [dependencyNodes](#), [exactType](#), [geometry](#), [ghost](#), [head](#), [hilite](#), [intermediateObjects](#), [invisible](#), [leaf](#), [lights](#), [live](#), [lockedNodes](#), [materials](#), [noIntermediate](#), [nodeTypes](#), [objectsOnly](#), [partitions](#), [persistentNodes](#), [planes](#), [preSelectHilite](#), [readOnly](#), [recursive](#), [referencedNodes](#), [references](#), [renderGlobals](#), [renderResolutions](#), [renderSetups](#), [selection](#), [sets](#), [shapes](#), [shortNames](#), [showType](#), [tail](#), [textures](#), [transforms](#), [type](#), [undeletable](#), [untemplated](#), [visible](#)

## MEL examples

```
// create some objects to operate on and select them all.
// Note that there are two objects named circle1;
circle -n circle1; group; circle -n circle1;
sphere -n spherel; group; instance;
select -ado;

// list all objects
ls;

// List all selected objects
ls -selection;

// List all hilited objects
ls -hilite;

// List last selected object
ls -selection -tail 1;

// List all objects named "spherel". Note that since spherel is
// instanced, the command below lists only the first instance.
ls spherel;

// To list all instances of spherel, use the -ap/allPaths flag.
ls -ap spherel;

// List all selected objects named "group*"
ls -sl "group*";

// List all geometry, lights and cameras in the DAG.
ls -geometry -lights -cameras;

// List all shapes in the dag.
ls -shapes;

// One thing to note is that it is better to always use the
// -l/long flag when listing nodes without any filter. This is
// because there may be two nodes with the same name (in this
// example, circle1). 'ls' will list the names of all the objects
// in the scene. Objects with the same name need a qualified
// path name which uniquely identifies the object. A command
// to select all objects such as "select `ls`" will fail because
// the object lookup can't resolve which "circle1" object is
// intended. To select all objects, you need the following:
// select `ls -l`;

// When trying to find a list of all objects of a specific
// type, one approach might be to list all objects and then
// use the nodeType command to then filter the list. As in:
// string $allObjects[];
// string $obj;
// $allObjects = `ls -l`;
// for ( $obj in $allObjects ) {
//   if ( `nodeType $obj` == "surfaceShape" ) {
//     print ($obj + "\n");
//   }
// }
```

```
// path name which uniquely identifies the object. A command
// to select all objects such as "select `ls`" will fail because
// the object lookup can't resolve which "circle1" object is
// intended. To select all objects, you need the following:
// select `ls -l`;

// When trying to find a list of all objects of a specific
// type, one approach might be to list all objects and then
// use the nodeType command to then filter the list. As in:
// string $allObjects[];
// string $obj;
// $allObjects = `ls -l`;
// for ( $obj in $allObjects ) {
//   if ( `nodeType $obj` == "surfaceShape" ) {
//     print ($obj + "\n");
//   }
// }
// The problem with this is that 'nodeType' returns the
// most derived type of the node. In this example, "surfaceShape"
// is a base type for nurbsSurface so nothing will be printed.
// To do this properly, the -typ/type flag should be used
// to list objects of a specific type as in:
// $allObjects = `ls -type surfaceShape`;
// for ( $obj in $allObjects ) {
//   print ($obj + "\n");
// }
```

```
// List all geometry shapes and their types
ls -type geometryShape -showType;

// List all paths to all leaf nodes in the DAG
ls -dag -lf -ap;

// List all nodes below the selected node
ls -dag -ap -sl;

// List all dag nodes that are read-only (i.e. referenced nodes)
ls -dag -ro;

// List all ghosting objects
ls -ghost;

// List reference nodes associated with specific files
ls -references;

// List all reference nodes, including unknown and shared reference
// nodes
ls -type reference;
```

# Python in Maya!



Cart (0) | Company | Contact Us | Partners



Products ▾ Solutions ▾ Purchase ▾ Support ▾ Community ▾ Store ▾

United States Worldwide Sites

Home > Services & Support > Autodesk Maya Services & Support

## Autodesk Maya Services & Support



Support

Documentation

Data & Downloads

Training

Feature Videos

Learning Path

Events and Seminars

Self-Paced Training

Tutorials

Advanced Techniques

Training Centers and Certification

Community

Consulting

## Maya 8.5 Features

Python | Python | Python

```
#depth and width o:  
depth = random.uni:  
width = random.uni:
```

```
# create a poly cul
```

View Now

### Python Scripting

Maya 8.5 delivers all the benefits of the Python scripting language – greatly accelerating the development of custom scripts and plug-ins. Using a sample "House" script (i.e. one that randomly generates simple polygonal houses), this video will show you some of the new scripting functionality within Maya (MEL and Python), such as multiple Script Editor tabs, new toolbar icons, and text zooming and scaling. You'll see how Python has been integrated into Maya at the same level as MEL – with particular focus given to a favorite MEL workflow that has been extended to Python scripting. Finally, you'll learn how you can take advantage of the sample Python plug-ins that ship with Maya 8.5.

<http://usa.autodesk.com/adsk/servlet/item?linkID=9242256&id=13587662&siteID=123112>

## with Mel

```
string $objs[] = `ls -type transform`;
for ($x in $objs) {
    print (longNameOf($x)); print "\n";

    // make and break some connections
    connectAttr( $x + ".sx" ) ($x + ".sy");
    connectAttr( $x + ".sx" ) ($x + ".sz");
    disconnectAttr( $x + ".sx" ) ($x + ".sy");
    string $conn[] = `listConnections -s 0 -d 1 -p 1 ($x + ".sx")`;
    for ($inputPlug in $conn)
        disconnectAttr ( $x + ".sx" ) $inputPlug;

    // add and set a string array attribute with the history of this tr
    if ( !`attributeExists "newAt" $x` )
        addAttr -ln newAt -dataType stringArray $x;
    string $shape[] = `listRelatives -s $x`;
    string $history[] = `listHistory $shape[0]`;
    string $elements = "";
    for ($elem in $history)
        $elements += "" + $elem + " ";
    eval ("setAttr -type stringArray " + $x + ".newAt " + `size $histo
    print `getAttr ( $x + ".newAt" )`;

    // get and set some attributes
    setAttr ($x + ".rotate") 1 1 1;
    float $trans[] = `getAttr ($x + ".translate")`;
    float $scale[] = `getAttr ($x + ".scale")`;
    $trans[0] *= $scale[0];
    $trans[1] *= $scale[1];
    $trans[2] *= $scale[2];
    setAttr ($x + ".scale") $trans[0] $trans[1] $trans[2];

    // call some other scripts
    myMelScript( `nodeType $x` , $trans );
}
```

## default Python

```
objs = cmds.ls( type= 'transform')
if objs is not None: # returns None when it f
    for x in objs:
        print mm.eval('longNameOf("%s")' % x)

    # make and break some connections
    cmds.connectAttr( '%s.sx' % x, '%s.sy' % x )
    cmds.connectAttr( '%s.sx' % x, '%s.sz' % x )
    cmds.disconnectAttr( '%s.sx' % x, '%s.sy' % x)

    conn = cmds.listConnections( x + ".sx", s=0, d=1, p=1)
    # returns None when it finds no matches
    if conn is not None:
        for inputPlug in conn:
            cmds.disconnectAttr( x + ".sx", inputPlug )

    # add and set a string array attribute with the history
    if not mm.eval( 'attributeExists "newAt" "%s"' % x):
        cmds.addAttr( x, ln='newAt', dataType='stringArray')
    shape = cmds.listRelatives( x, s=1 )
    if shape is not None:
        history = cmds.listHistory( shape[0] )
    else:
        history = []
    args = tuple( ['%s.newAt' % x, len(history)] + history )
    cmds.setAttr( *args , type= 'stringArray' )

    # get and set some attributes
    cmds.setAttr ( '%s.rotate' % x, 1, 1, 1 )
    scale = cmds.getAttr ( '%s.scale' % x )
    scale = scale[0] # maya packs the previous result in a l
    trans = list( cmds.getAttr ( '%s.translate' % x )[0] )
    trans[0] *= scale[0]
    trans[1] *= scale[1]
    trans[2] *= scale[2]
    cmds.setAttr ( '%s.scale' % x, trans[0], trans[1], trans[2] )
    mm.eval('myMelScript("%s",%s,%s,%s)') % (cmds.nodeType(
```



PyMEL

Python in Maya Done Right

Search projects

Project Home

Downloads

Wiki

Issues

Source

Summary People

#### Project Information

 +3 Recommend this on Google

 Starred by 131 users  
[Project feeds](#)

 **Code license**  
[New BSD License](#)

 **Labels**  
python, maya, mel, 3d,  
graphics, VFX, CG

 **Members**  
[chad...@gmail.com](mailto:chad...@gmail.com),  
[elron...@gmail.com](mailto:elron...@gmail.com)  
[3 committers](#)

#### Featured

 **Downloads**  
[pymel-1.0.3.zip](#)  
[Show all »](#)

#### Links

 **External links**  
[Docs](#)  
[Github Repo](#)  
[Maya Python Bugs](#)  
[Luma Pictures](#)  
[Altitude Studio](#)

## Python in Maya Done Right

PyMEL makes python scripting with Maya work the way it should. Maya's command module is a direct translation of mel commands into python commands. The result is a very awkward and unpythonic syntax which does not take advantage of python's strengths -- particularly, a flexible, object-oriented design. PyMEL builds on the cmds module by organizing many of its commands into a class hierarchy, and by customizing them to operate in a more succinct and intuitive way.

### Project Goals

- Create an open-source python module for Maya that is intuitive to MEL users and python users alike
- Fix bugs and design limitations in Maya's python modules, maya.cmds and maya.mel
- Keep code concise and readable
- Add organization through class hierarchy and sub-modules
- Provide documentation accessible via html and the builtin help() function
- Make it "just work"

### Production Proven

Since its release over a year ago PyMEL, has accumulated an impressive resume in both feature films and games:

- DreamWorks: Fung Fu Panda, Shrek 4, Monsters Vs Aliens, and How to Train Your Dragon
- Luma Pictures: Pirates of the Caribbean: At World's End, Harry Potter 6, and Wolverine
- ImageMovers Digital: Robert Zemeckis' upcoming A Christman Carol

Here's what Seth Gibson of Bungie Studios, makers of the hit game Halo, has to say:

"Having done production python code myself for many years, wrapping my head around Maya's native implementation took a little bit of time. With PyMEL, I can think and write the python code and syntax I'm already used to, which speeds up my development time considerably. It's also going to help our other Technical Artists with their Python learning curve, since PyMEL's syntax is consistent with most other python packages. Kudos to the PyMEL team for such a well thought out project!"

File Edit View Terminal Help

```
moduleNameShortToLong[moduleName] = getModuleCommandList( longname, long_version )

tmpCmdlist = inspect.getmembers(cmds, callable)
cmdlist = {}
#moduleCmds = defaultdict(list)
moduleCmds = dict( (k,[]) for k in moduleNameShortToLong.keys() )
moduleCmds.update( {'other':[], 'runtime': [], 'context': [], 'uiClass': [] } )

for funcName, data in tmpCmdlist :
    # determine to which module this function belongs
    module = None
    if funcName in ['eval', 'file', 'filter', 'help', 'quit']:
        module = None
    elif funcName.startswith('ctx') or funcName.endswith('Ctx') or funcName.endswith('Context'):
        module = 'context'
    elif funcName in uiClassList:
        # module = 'uiClass'
    elif funcName in nodeHierarchyTree or funcName in nodeTypeToNodeCommand.values():
        # module = 'node'
    else:
        for moduleName, commands in moduleNameShortToLong.iteritems():
            if funcName in commands:
                module = moduleName
                break
    if module is None:
        if mm.eval('whatIs "%s"' % funcName ) == 'Run Time Command':
            module = 'runtime'
        else:
            module = 'other'

    cmdInfo = {}

    if module:
        moduleCmds[module].append(funcName)

    if module != 'runtime':
        cmdInfo = getCmdInfo(funcName, long_version)

    if module != 'windows':
        if funcName in nodeFunctions:
            nodeCommandList.append(funcName)
            cmdInfo = testNodeCmd( funcName, cmdInfo, nodeCmd=True, verbose=True )
        #elif module != 'context':
        #    cmdInfo = testNodeCmd( funcName, cmdInfo, nodeCmd=False, verbose=True )

    cmdInfo['type'] = module
    flags = getCallbackFlags(cmdInfo)
    if flags:
        cmdInfo['callbackFlags'] = flags
    cmdlist[funcName] = cmdInfo
```

## with Mel

```
string $objs[] = `ls -ty
for ($x in $objs) {
    print (longNameOf($x
    // make and break so
    connectAttr( $x + ".c
    connectAttr( $x + ".c
    disconnectAttr( $x +
    string $conn[] = `li
    for ($inputPlug in $c
        disconnectAttr (
    // add and set a str
    if ( !`attributeExis
        addAttr -ln newA
    string $shape[] = `l
    string $history[] =
    string $elements =
    for ($elem in $histo
        $elements += """
    eval ("setAttr -type
    print `getAttr ( $x
    // get and set some
    setAttr ($x + ".rota
    float $trans[] = `ge
    float $scale[] = `ge
    $trans[0] *= $scale[
    $trans[1] *= $scale[
    $trans[2] *= $scale[
    setAttr ($x + ".scal
    // call some other s
    myMelScript( `nodeTy
}
```

## default Python

```
objs = cmds.ls( type='t
if objs is not None:
    for x in objs:
        print mm.eval('l
        # make and break
        cmds.connectAttr(
        cmds.connectAttr(
        cmds.disconnectAttr(
        conn = cmds.list
        # returns None w
        if conn is not N
            for inputPlu
                cmds.disconnectAttr(
        # add and set a
        if not mm.eval(
            cmds.addAttribute(
        shape = cmds.list
        if shape is not N
            history = cmd
        else:
            history = []
        args = tuple(
            ['%s.newAt' % x, len(history)] + history
        cmds.setAttr( *args
        # get and set some
        cmds.setAttr ( '%s.rota
        scale = cmds.getAttr ( '%s.scale' % x
        scale = scale[0] # maya packs the previous result in a list for no apparent reason
        trans = list( cmds.getAttr ( '%s.translate' % x )[0] ) # the tuple must be converted to a list for item
        trans[0] *= scale[0]
        trans[1] *= scale[1]
        trans[2] *= scale[2]
        cmds.setAttr ( '%s.scale' % x, trans[0], trans[1], trans[2] )
        mm.eval('myMelScript("%s",%s,%s,%s)' % (cmds.nodeType(x), trans[0], trans[1], trans[2] )
```

## with Pymel

```
from pymel import *
for x in ls( type='transform' ):
    print x.longName()
    # safe to import into main namespace
    # object oriented design
    # make and break some connections
    x.sx >> x.sy
    x.sx >> x.sz
    x.sx // x.sy
    x.sx.disconnect()
    # add and set a string array attribute with the history of this transform's shape
    x.setAttr( 'newAt', x.getShape().history(), force=1 )
    # get and set some attributes
    x.rotate.set( [1,1,1] )
    trans = x.translate.get()
    trans *= x.scale.get() # vector math
    x.translate.set( trans ) # ability to pass list/vector args
    mel.myMelScript(x.type(), trans) # automatic handling of mel procedures
```

# Ever wondering users' behaviors?



# Dracula 2.9.2 [Asset]

File Settings Programs Help



Task 1 Asset 1 Scene 1 Movie 1 Movie Maker 1

## Project

npds

## Type

all

## Asset

npds\_c\_boy

npds\_c\_duck

npds\_c\_fish

npds\_c\_fisher

npds\_c\_man\_A

npds\_c\_man\_B

npds\_c\_man\_C

npds\_c\_pixie

npds\_c\_poet\_A

npds\_c\_poet\_B

npds\_c\_tang

npds\_c\_wang

npds\_c\_zhen\_er

## Stage

all

## Working Asset Files



Stage	Asset Folder	Asset File	Owner	Date
rigging	roy_yu	npds_c_fisher_rig_h_v18.ma	roy_yu	2012/09/03
rigging	None	npds_c_fisher_rig_h_v18.ma	roy_yu	2012/09/03
rigging	roy_yu	npds_c_fisher_rig_h_v17.ma	roy_yu	2012/08/29
rigging	roy_yu	npds_c_fisher_rig_h_v16.ma	roy_yu	2012/08/29
rigging	roy_yu	npds_c_fisher_rig_h_v15.ma	roy_yu	2012/08/29
rigging	monkey	npds_c_fisher_rig_h_v15.ma	monkey	2012/08/29

32 working assets

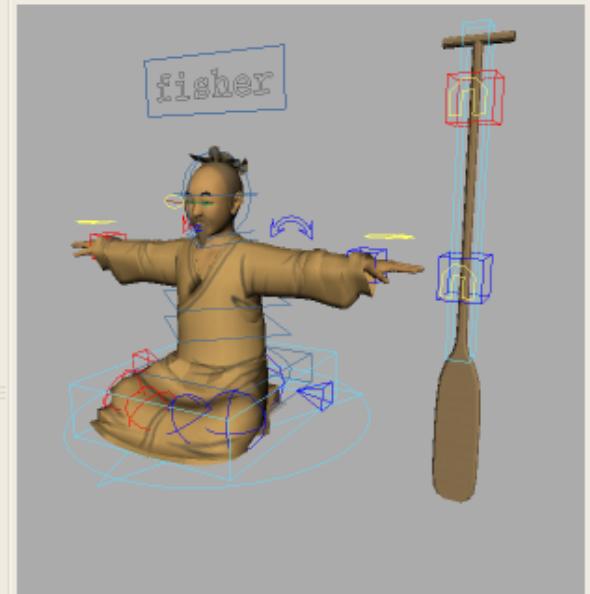
## Published Asset Files



Stage	State	Asset File	Ass	Publisher	Publish Date
rigging	CBB	npds_c_fisher_rig_h_v18.ma	10	roy_yu	2012/09/03
model	CBB	npds_c_fisher_mod_h.ma	2	rainmaker	2012/08/06

2 published assets

## Info



pub by ? at ?



2012/08/31 #tex

review by edward

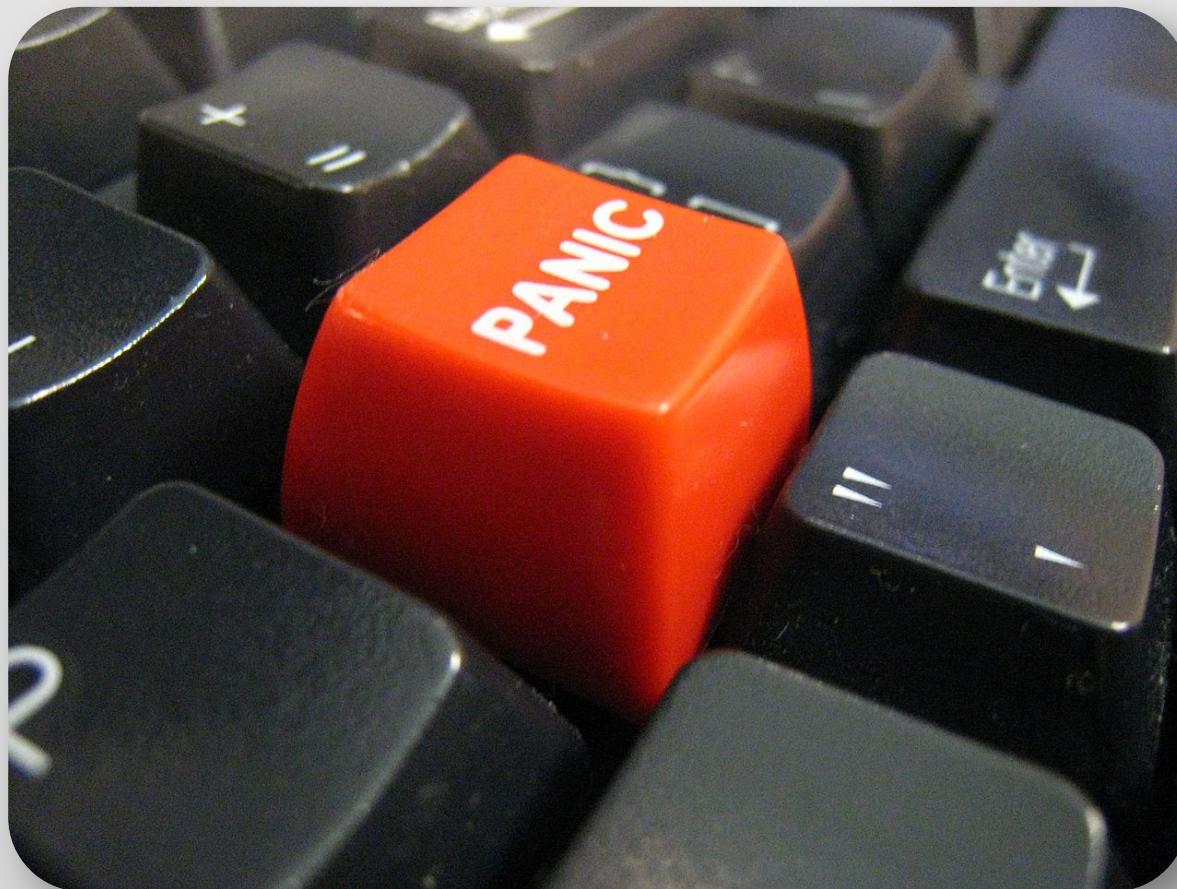
▲ 通過

Post Remark

# Logging first

name	value
level	DEBUG
user	stella
host	pc0111
ip	192.168.101.111
os	posix
pid	12039
timestamp	2011-12-22T16:46:55Z
klass	CinemaSelectorController
function	OnModelUpdateShots
params	(['sh0010', 'sh0020', 'sh0030', 'sh0040', 'sh0050'], )
filename	src/core/controller/CinemaSelectorController.py

# The most clicked buttons (functions)?



[http://commons.wikimedia.org/wiki/File:Panic\\_button.jpg](http://commons.wikimedia.org/wiki/File:Panic_button.jpg)

[About](#)[Contribute](#)[Mailing List](#)[Download](#)[Quick Links](#)

**Apache CouchDB™** is a database  
that uses **JSON** for documents,  
**JavaScript** for **MapReduce** queries,  
and regular **HTTP** for an **API**

**DOWNLOAD**  
Version 1.3.0



A Database for the Web

<http://couchdb.apache.org/>

# The most clicked buttons (functions)?

▼ View Code

Map Function:

```
function(doc) {  
  if(doc.ts) {  
    emit([doc(klass, doc.function], 1);  
  }  
}
```

Reduce Function (optional):

```
function(keys, values, rereduce) {  
  if (rereduce) {  
    return sum(values);  
  } else {  
    return 1;  
  }  
}
```

Run    Language: javascript ▾    Revert    Save As...    Save

# Pre-defined behavior analysis

- The crash rate of this app?
- Who and how many people works in weekend?
- The heartbeat of the production team?
- ....

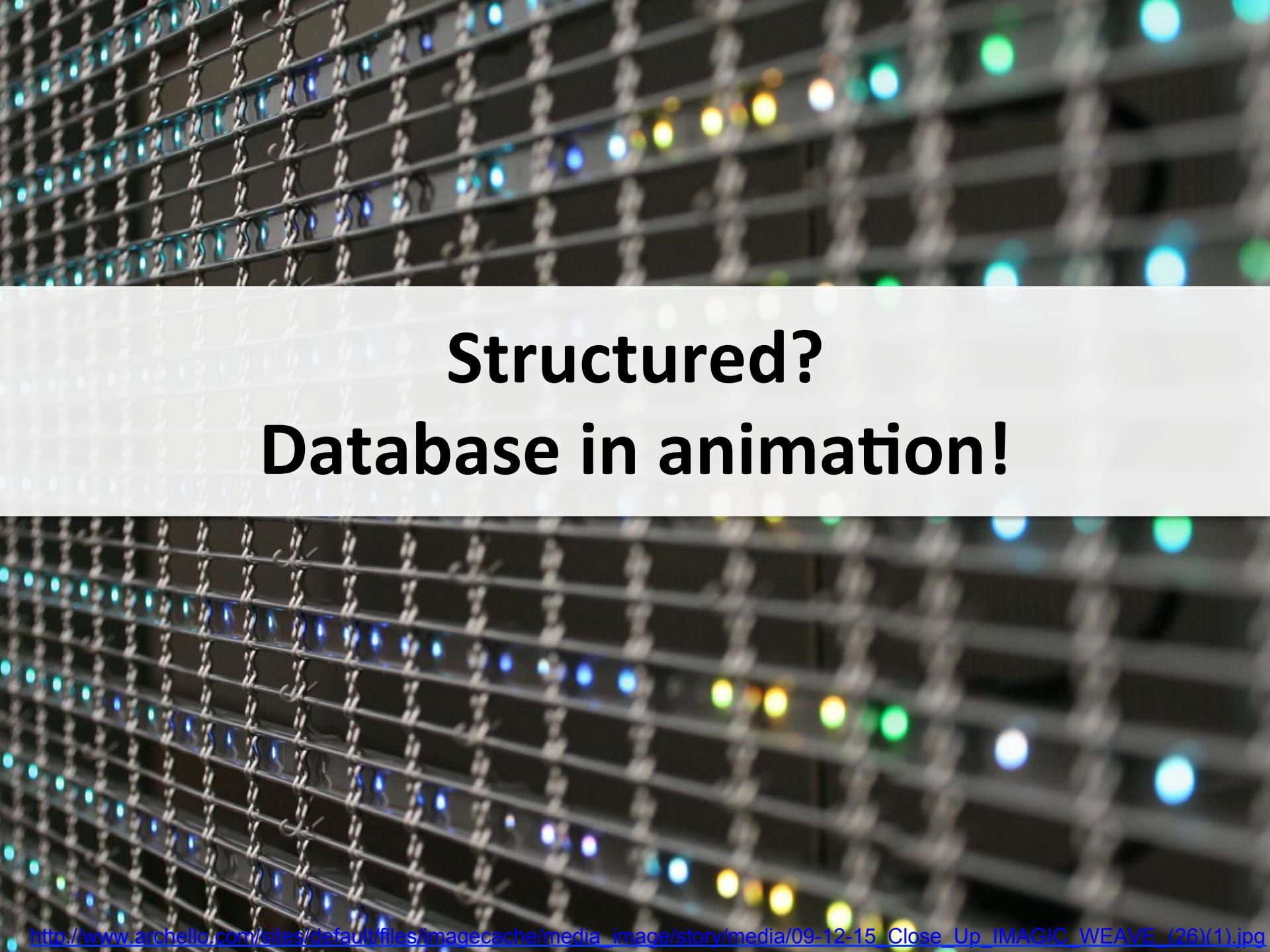
# Behavior logging/mining in CouchDB

## Pros

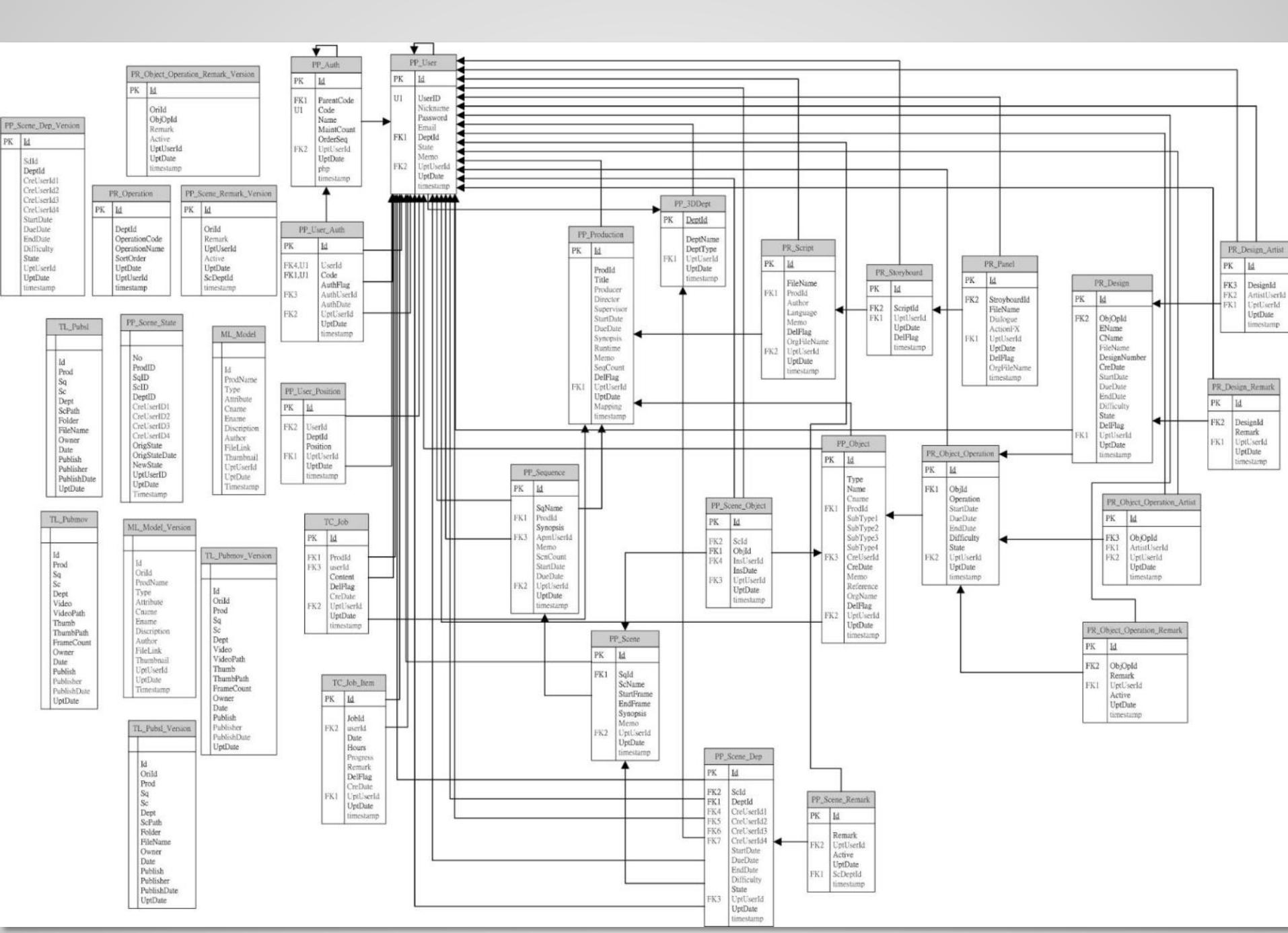
- 秒殺
- 易用
- JSON loves Dict

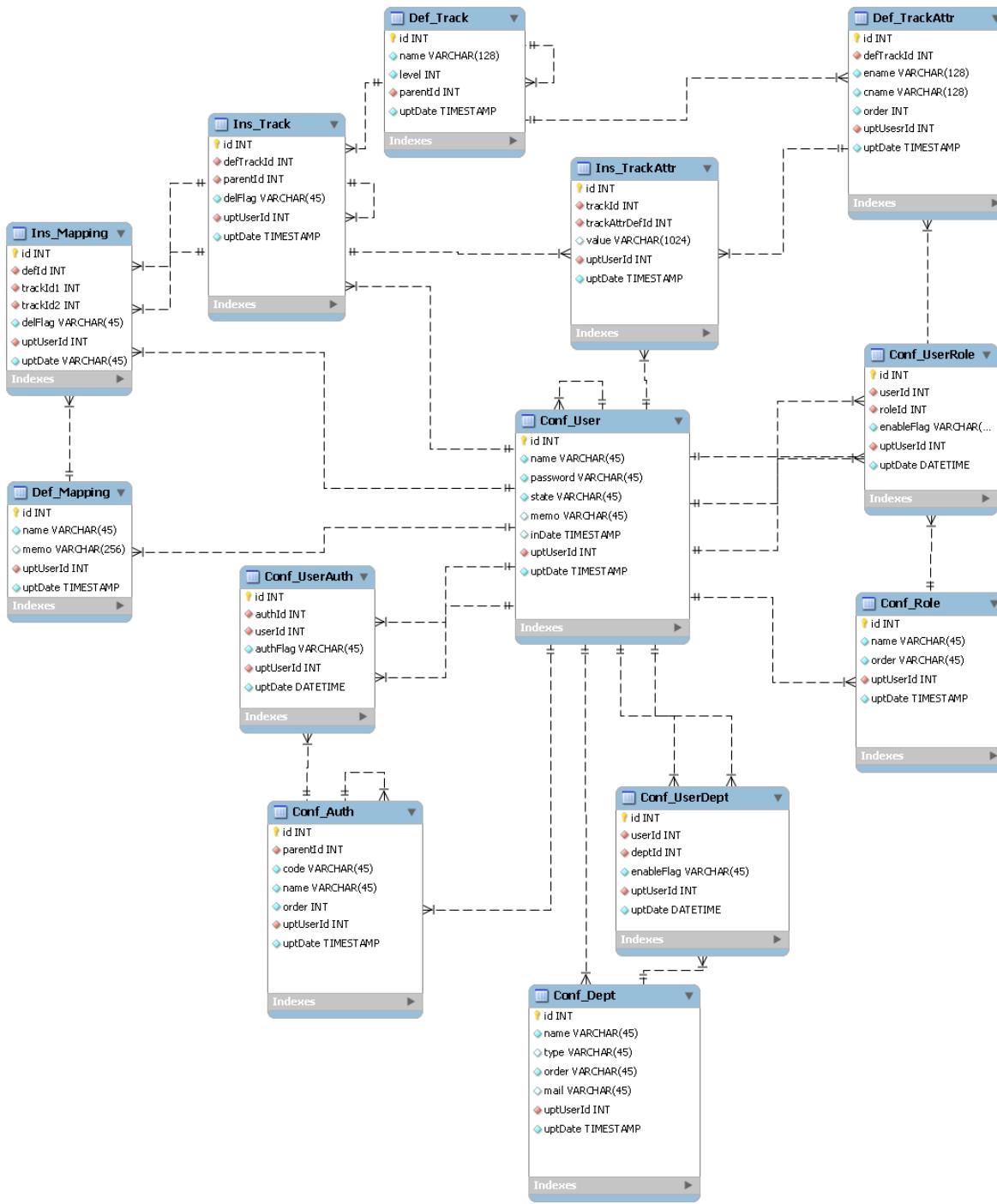
## Cons

- View by MapReduce
- 第一次產生 View 超慢
- Disk monster
- Javascript?!



# **Structured? Database in animation!**







<http://blog.prowork.me/post/41349307980/inserts-and-the-new-mongodb-php-driver-1-3>

## Big Data is a Matter of Speed, not Size

We risk overlooking the much more important story here, the real revolution, which is the mass democratisation of the means of access, storage and processing of data. ***This story isn't about large organisations running parallel software on tens of thousand of servers***, but about more people than ever being able to collaborate effectively around a distributed ecosystem of information, ***an ecosystem of small data***.

# Asset management for creative environment

- DB schema?
- Query performance?
- JIT dev/deploy?



# Takeaway

- Version control for non-programmer is an un-explored area.
- Python dominates animation/VFX industry.
- Anyone can benefit from NoSQL and other cloud techniques.





Thank You