Python + Maya - Part 1

January 24, 2008 by Rodrigo Araujo

First, what's Maya?

> Maya is a high end 3D computer graphics and 3D modeling software package originally developed by Alias Systems Corporation, but now owned by Autodesk as part of the Media and Entertainment division. Autodesk acquired the software in October 2005 upon purchasing Alias. Maya is used in the film and TV industry, as well as for computer and video games.*Wikipedia on Autodesk Maya*

On its 8th version, Maya introduced Python scripting as an alternative for their native scripting language, MEL (Maya Embedded Language) which was being used for quite a long time.
After spending some months learning Python, today started to study maya bindings to Python and decided to make a series of posts showing some of the things I learn on the process.

The idea is that Python and MEL are alternatives for a programmer as they provided similar functionalities. So one must choose between one each and start their program.

Inputing code to Maya might be done in a series of different ways:

- **On start** Maya always runs a script named "userSetup.py" which must be located at:
  - Windows: <drive>:\Documents and Settings\<user>\My Documents\maya\<version>\scripts
  - Mac OS X: ~/Library/Preferences/Autodesk/maya/<version>/scripts
  - Linux: ~/maya/<version>/scripts

  So you can write anything you'd like Maya to do on its first moments of life at this file;

- **Command Line** It's possible to input simple commands through Maya command line on the bottom of the window, like some select command of a hard to click object;
- **Script Editor** The best way to test and debug your script. Since 8th edition, the script editor has two tabs on its lower part: MEL and Python. Clicking Python part won't change anything but the interpreter the script editor will use to interpret your code. It provides simple ways to test it, but could be better if there was some syntax highlight;
- **Code outside Maya** One can write the whole code outside Maya using any editor he'd like. It's probably the best alternative for big plugins since most editors are a lot better than Maya Script Editor itself.

After learning that, and jumping on to the API to see what it was capable i was pleasantly surprised (since I've never spent much time studying MEL's API) to find out it was capable of many amazing things and they were all documented.

It's possible to access every attribute of any of the nodes on the scene, to create key frames and make complete animations through one script and to access many of the tools provided throughout the menus.

One quick example is how to create a bouncing ball animation.

```
import maya.cmds as cmds
cmds.polySphere( name = 'bouncing', radius = 2 )
g = -0.04
v0y = 0.8
```

```
v0x = 0.1
pos0y = 2
pos0x = 0
for itr in xrange(0,6):
for tx in xrange(0,42):
        posy = pos0y + v0y*(tx-1) + g*(tx-1)*(tx-1)/2
        posx = pos0x + v0x*((itr*42) + tx-1)
        cmds.setKeyframe( 'bouncing', attribute='translateY', value=posy, t=(itr*42) + tx )
        cmds.setKeyframe( 'bouncing', attribute='translateX', value=posx, t=(itr*42) + tx )
```

Line 1 imports all the Maya Comands to cmds and line 2 creates a polygonal sphere named "bouncing" with 2 units radius. Lines 3 to 7 set a few physical constants like gravity (g), initial speeds (v0x and v0y) and initial positions (pos0x and pos0y). Then comes the keyframe creation. The ball will jump 6 times, so we have a for from 0 to 6. Each bounce will take 42 frames what explains the second for from 0 to 42. Inside the second for and through some basic physical equations we set new positions for both x and y axis, and then create keyframes for both attributes.

I'm still trying to figure out a way to create less keyframes (through this method, I ended up creating 252 keyframes) even though I know this doesn't represent a real problem for Maya standards.

So, thanks for reading, hope you enjoyed it and, if you try this at home, please feel free to comment your experience. And don't forget to check our next episode: Python + Maya - Part 2!!

Powered by ScribeFire.

Possibly related posts: (automatically generated)

- **3D Modelling**

Posted in 3D, Maya, Python | 6 Comments

6 Responses

1. *on October 23, 2008 at 4:53 pm | Reply mr*

   This is my first attempt at python, so thanks for the tutorial.
   I had trouble running the script and the errors indicated that very specific indenting is necessary. In addition, I had to change the quote characters (not sure what you used) to a single quote character.
   I'm pasting the code below and hope the spacing will show up correctly - I followed Martin's comment above.
   Thanks again, you can certainly accomplish a lot w/a few lines of code!

```
import maya.cmds as cmds
cmds.polySphere(name='bouncing', radius=2)
g = -0.04
```

```
v0y = 0.8
v0x = 0.1
pos0y = 2
pos0x = 0
for itr in xrange(0,6):
for tx in xrange(0,42):
posy = pos0y + v0y*(tx-1) + g*(tx-1)*(tx-1)/2
posx = pos0x + v0x*((itr*42) + tx-1)
cmds.setKeyframe( 'bouncing', attribute='translateY', value=posy, t=(itr*42) + tx )
cmds.setKeyframe( 'bouncing', attribute='translateX', value=posx, t=(itr*42) + tx )
```