How are large applications embedding Python?

Peter Shinners

Pycon 2010

Conclusion

- No scary roadblocks or workarounds
- Be ready for heavy initial work
- Ongoing maintenance is light
- Have a "Python Guy"

Thanks for coming

Excellent Django, Cherrypy, EC3, and interpreter sessions are next door.

For everyone else...

Audience survey time.

Major Pieces

- Embedding
- Extending
- Distributing
- Graphical Interactive Interpreter

Major Pieces

- Embedding
- Extending
- Distributing
- Graphical Interactive Interpreter

```
#include <Python.h>
int main(int argc, char** argv) {
    Py_Initialize();
    Py_RunSimpleString("print 'Hello World'");
    Py_Finalize();
    return 0;
}
```

Large Applications

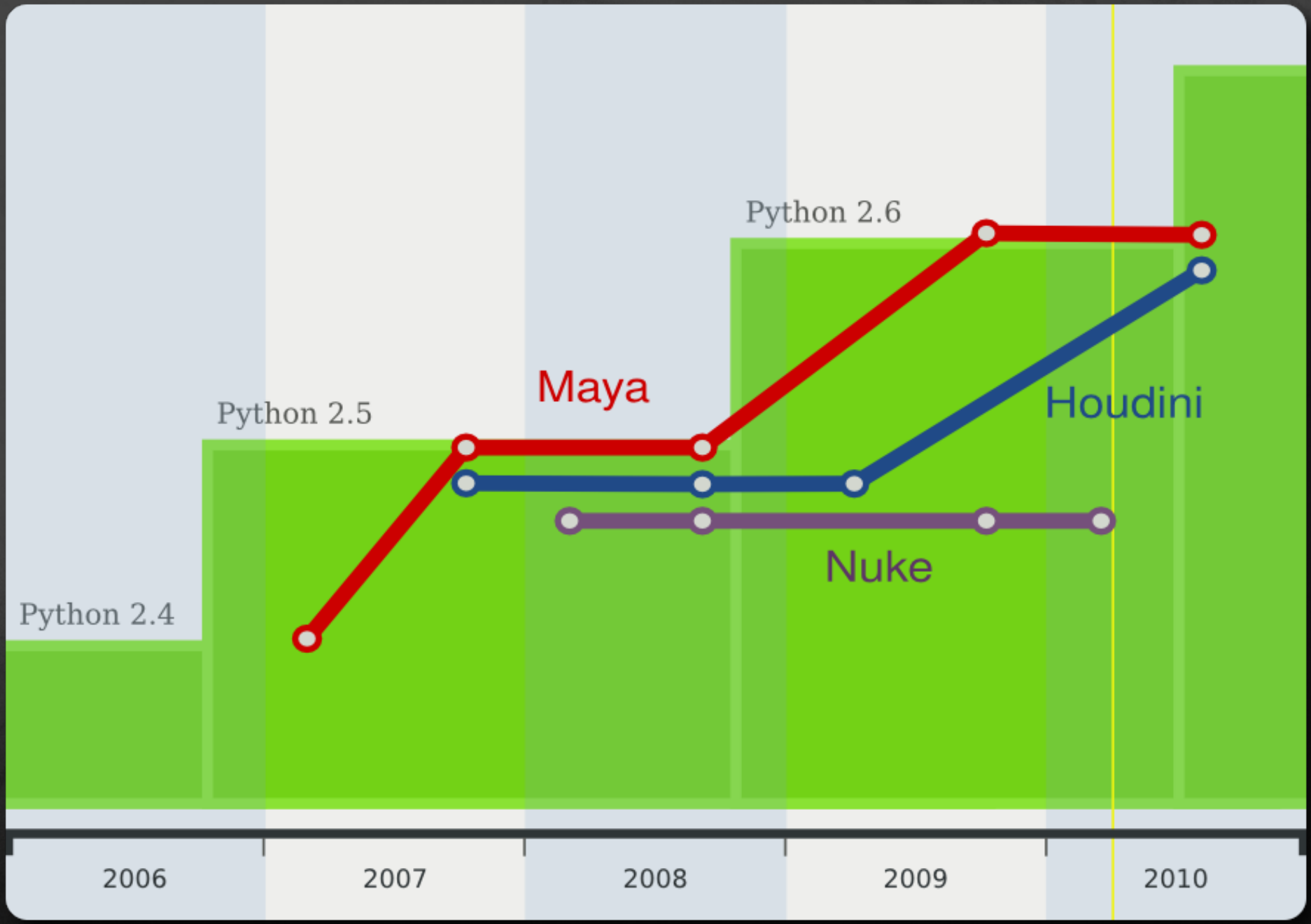
Maya
Autodesk

Nuke
The Foundry

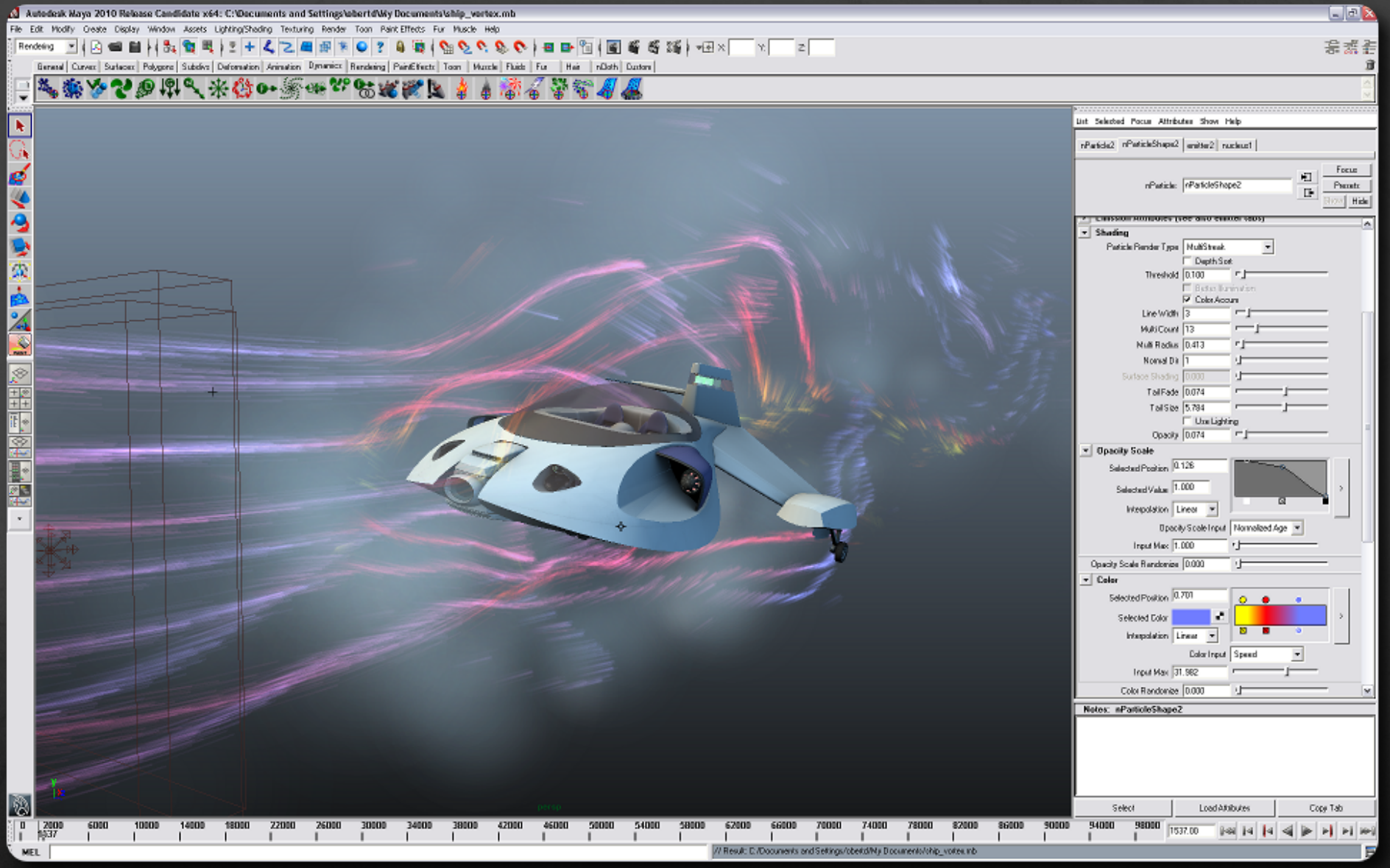
Houdini
Side Effect Software

Blender
Blender Foundation

Application Releases



Maya


Autodesk Maya 2010 Release Candidate x64: C:\Documents and Settings\oberti\My Documents\ship_vertex.mb
File Edit Modify Create Display Window Assets Lighting/Shading Texturing Render Toon Paint Effects Fur Muscle Help
Rendering
General Curves Surfaces Polygons Subdivs Deformation Animation Dynamics Rendering PaintEffects Toon Muscle Fluids Fur Hair nCloth Custom
List Selected Focus Attributes Show Help
nParticle2 nParticleShape2 emitter2 nucleus1
nParticle: nParticleShape2
Focus
Presets
Hide
Shading
Particle Render Type: MultiStreak
Depth Sort
Threshold: 0.100
Color Accum
Line Width: 3
Multi Count: 13
Multi Radius: 0.413
Normal Dir: 1
Surface Shading: 0.000
Tail Fade: 0.074
Tail Size: 5.784
Use Lighting
Opacity: 0.074
Opacity Scale
Selected Position: 0.126
Selected Value: 1.000
Interpolation: Linear
Opacity Scale Input: Normalized Age
Input Max: 1.000
Opacity Scale Randomize: 0.000
Color
Selected Position: 0.701
Selected Color
Interpolation: Linear
Color Input: Speed
Input Max: 31.982
Color Randomize: 0.000
Notes: nParticleShape2
Select
Load Attributes
Copy Tab
MEL

Maya Scripting

- Works alongside MEL
- Hundreds of functions
- Generated at runtime
- Use of named arguments

Maya Scripting

- Works alongside MEL
- Hundreds of functions
- Generated at runtime
- Use of named arguments

```
import maya.cmds as cmds
xforms = cmds.ls(type="transform")
meshes = cmds.listRelatives(xforms, type="mesh")
for mesh in meshes:
    if not cmds.isVisible(mesh):
        continue
    polys = cmds.polyInfo(mesh, numPolys=True)
```

Maya Plugins

- SWIG interface to C++
- Follows C++ coding style
- Maintaining SWIG interfaces not hard

Maya Plugins

- SWIG interface to C++
- Follows C++ coding style
- Maintaining SWIG interfaces not hard

```
from maya import OpenMaya, OpenMayaMP
class PythonNode(OpenMayaMPx.MPxNode):
    testAttr = OpenMaya.MObject()

length = OpenMaya.doublePtr()
edge.getLength(length, space)
length.value()
```

Pymel, Third Party API

- Luma Pictures, open source
- Pythonic interface, combines both APIs
- Great documentation

Pymel, Third Party API

- Luma Pictures, open source
- Pythonic interface, combines both APIs
- Great documentation

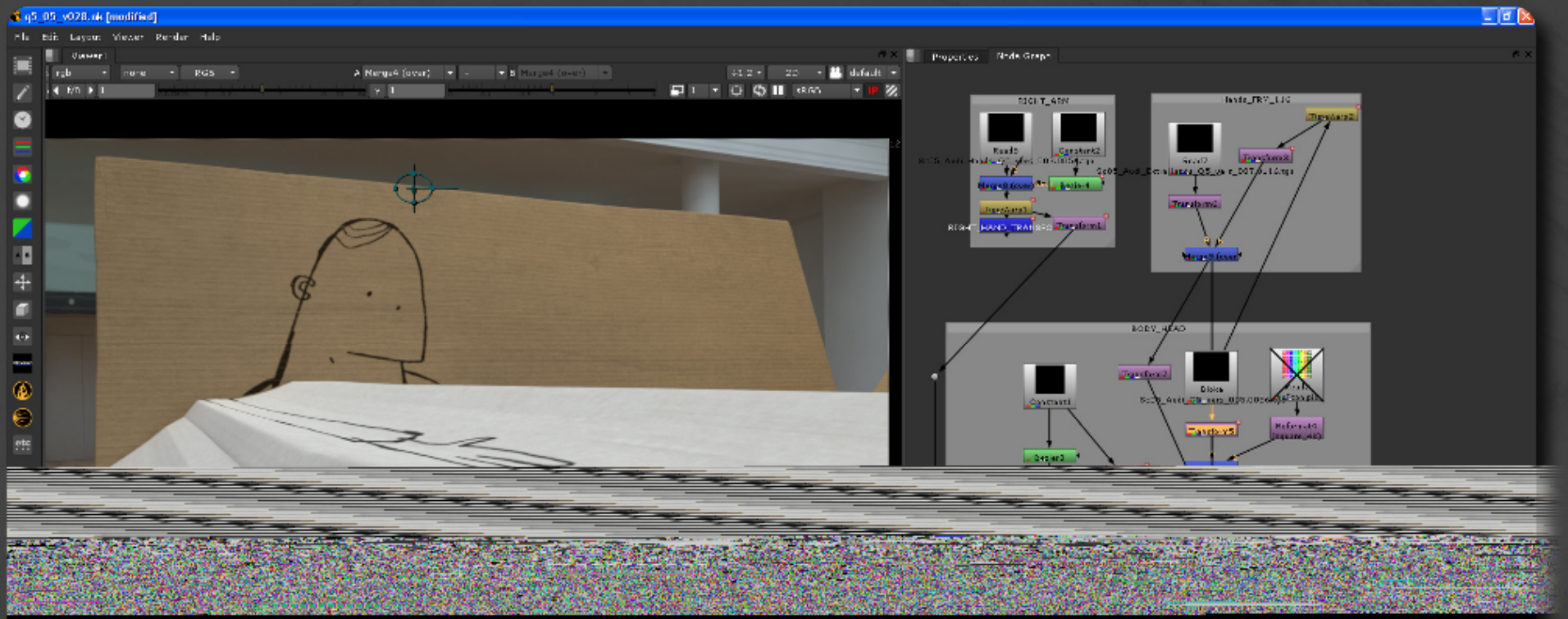
```
import pymel

for x in pymel.ls(type="transform"):
    history = x.getShape().history()
    x.setAttr("newAt", history, force=True)
    x.rotate.set((0, 0, 0))
    pymel.mel.myMelScript(x.type(), "Python")
```

Maya File Layout

```
bin/maya.exe
bin/mayapy.exe
bin/python26.dll
bin/python26.zip
include/python/
lib/python26.dll
python/
  DLLs/
  site-packages/
    maya/
```

```
bin/maya
bin/mayapy
bin/maya.bin
bin/python26.zip
include/python/
lib/python26.so
python/
  lib-dynload/
  site-packages/
    maya/
```

Nuke



Nuke Scripting

- Separate API from previous TCL
- Initially used Boost, switched to CPython
- Considering API version 2

Nuke Scripting

- Separate API from previous TCL
- Initially used Boost, switched to CPython
- Considering API version 2

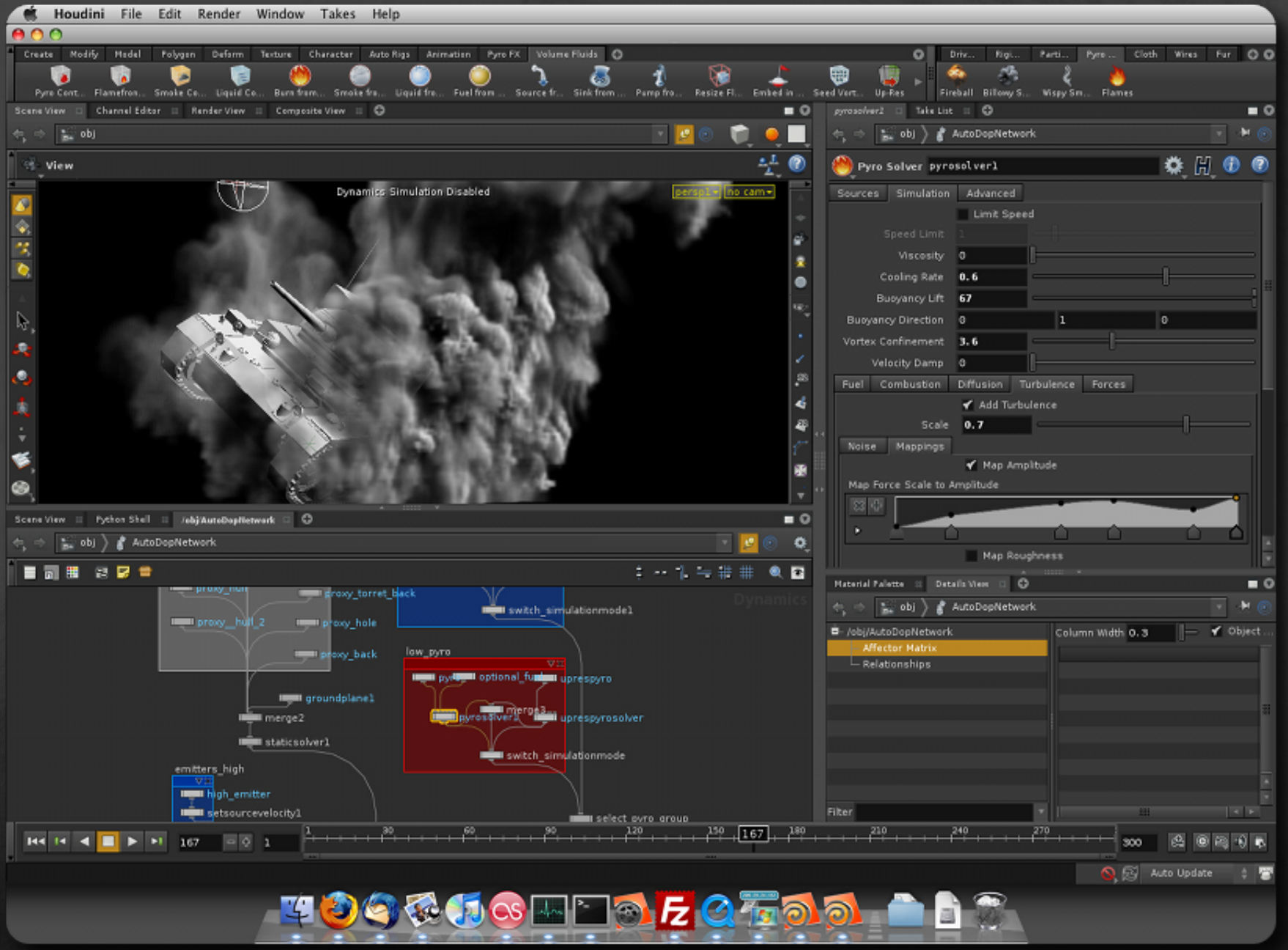
```
import nuke
def node_copypaste():
  nuke.nodeCopy(cut_paste_file())
  nodes = nuke.allNodes();
  for n in nodes:
    n.knob("selected").setValue(False)
  nuke.nodePaste(nukescripts.cut_paste_file())
```

Nuke File Layout

```
Nuke6.0.exe
python25.dll
plugins/
  nuke/
  nukescripts/
lib/
  python2.5/
```

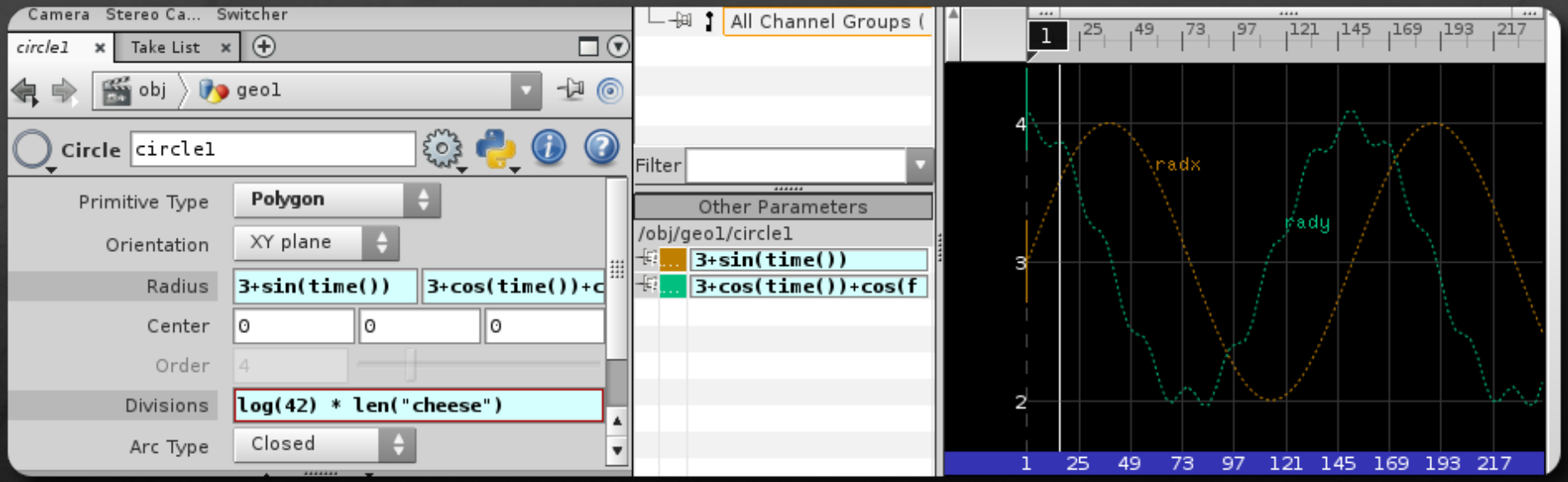
```
Nuke6.0
libpython2.5.so.1.0
plugins/
  nuke/
  nukescripts/
lib/
  python2.5/
```

Houdini


Houdini File Edit Render Window Takes Help
Dynamics Simulation Disabled
Pyro Solver pyrosolver1
Sources Simulation Advanced
Limit Speed
Viscosity
Cooling Rate
Buoyancy Lift
Buoyancy Direction
Vortex Confinement
Velocity Damp
Fuel Combustion Diffusion Turbulence Forces
Add Turbulence
Scale
Noise Mappings
Map Amplitude
Map Force Scale to Amplitude
Map Roughness
proxy_torret_back
proxy_hole
proxy_back
switch_simulationmode1
low_pyro
uprespyro
upprespyrosolver
groundplane1
merge2
staticsolver1
switch_simulationmode
emitters_high
high_emitter
setsourcevelocity1
select_pyro_group
/obj/AutoDopNetwork
Affector Matrix
Relationships
Column Width
Filter
Auto Update

Houdini Expressions

- Control values through Python expression
- Generate procedural animations
- Internally store compiled bytecode
- Escape key can interrupt execution



Houdini Scripting

- SWIG on for all Python bindings
  - Plugin extensions
  - Scripting
  - Expression
- Created new C++ for scripting

Houdini Scripting

- SWIG on for all Python bindings
  - Plugin extensions
  - Scripting
  - Expression
- Created new C++ for scripting

```
import hou

bbox = hou.BoundingBox()
geo = sop.geometry()
if geo and len(geo.points()):
    bbox = geo.boundingBox() * obj.
worldTransform()
```

Houdini File Layout

```
bin/houdini.exe
bin/hython.exe
bin/python26.dll
python/
  bin/python26.exe
  bin/msvc80.dll

  lib/python26/
houdini/scripts/
  python/
  pythonlibs/
```

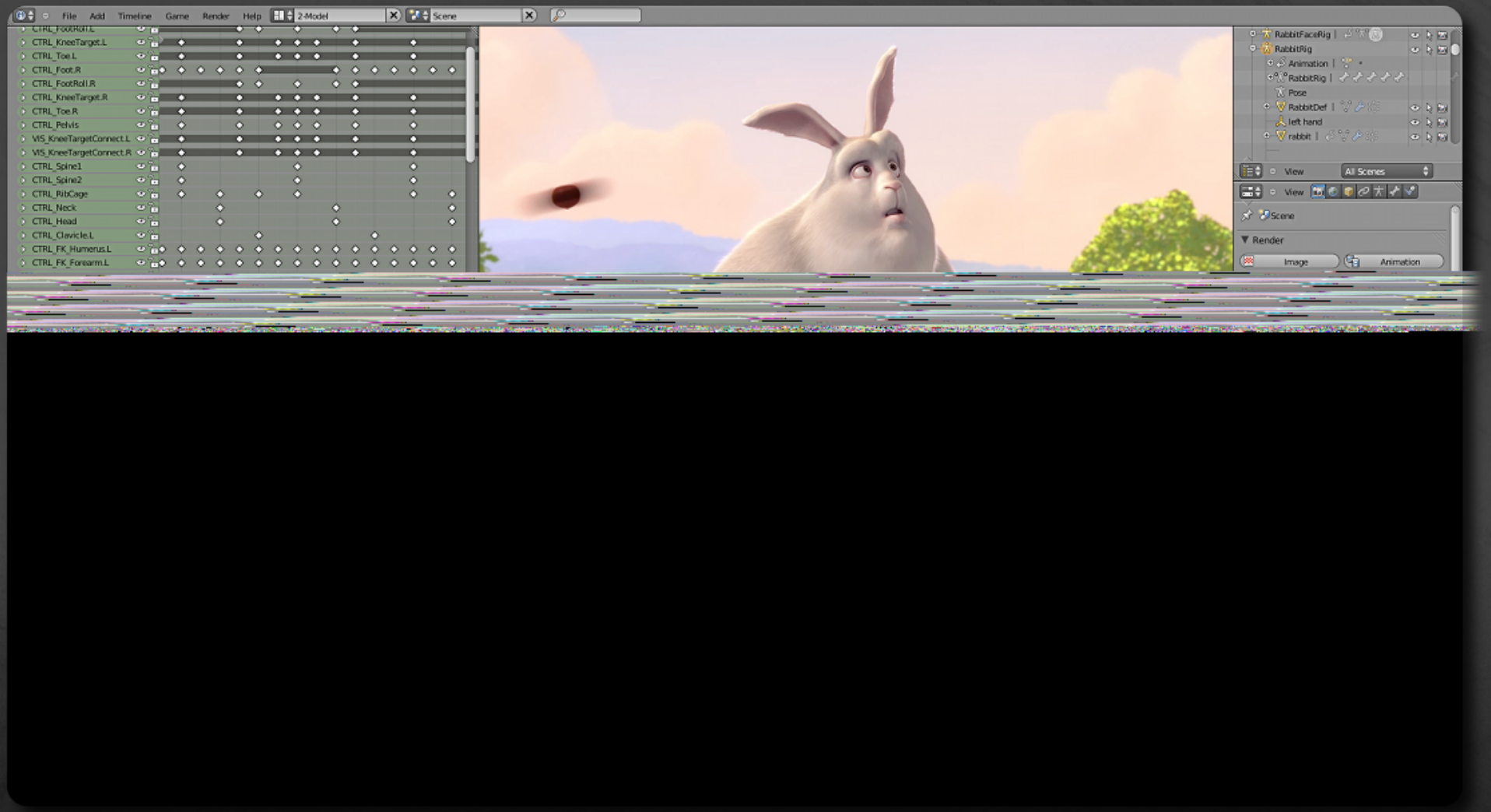
```
bin/houdini
bin/hython

python/
  bin/python26
  bin/python-bin
  lib/libpython26.so
  lib/python26
houdini/scripts/
  python/
  pythonlibs/
```

Houdini Linking

- Use external python shared library
- Internal PY library avoids linking
  - Macros for function pointers
  - Functions for constants
- Work with multiple Python versions

- Not used on Windows
  - Compiler runtime mismatch

Blender


File
Add
Timeline
Game
Render
Help
2-Model
Scene
CTRL_KneeTarget.L
CTRL_Toe.L
CTRL_Foot.R
CTRL_FootRoll.R
CTRL_KneeTarget.R
CTRL_Toe.R
CTRL_Pelvis
VIS_KneeTargetConnect.L
VIS_KneeTargetConnect.R
CTRL_Spine1
CTRL_Spine2
CTRL_RibCage
CTRL_Neck
CTRL_Head
CTRL_Clavicle.L
CTRL_FK_Humerus.L
CTRL_FK_Forearm.L
RabbitFaceRig
RabbitRig
Animation
RabbitRig
Pose
RabbitDef
left hand
rabbit
View
All Scenes
View
Scene
Render
Image
Animation

Blender File Layout

```
blender.exe
python31.dll
Python-license.txt
.blender/
  script/
    io/
    modules/
    op/
    ui/
  python/lib/
    python3.1
```

```
blender

Python-license.txt
.blender/
  script/
    io/
    modules/
    op/
    ui/
  python/lib/
    python3.1
```

Binary Compatibility

Win 

- MSVC compiler versions

Linux 

- Unicode size mismatch
  - UCS2 is Python default
  - UCS4 is most distributions

OSX 

- Stock mostly compatible Python.org
  - Distutils not "universal" friendly
  - Libraries and paths separated

Thank You

- *Maya*
  - Chris Grebeldinger
- *Nuke*
  - Matt Plec
  - Goncalo Carvalho
  - Vilya Harvey
- *Houdini*
  - Luke Moore
- *Blender*
  - Joseph Eagar
  - Willian Germano

Open Space Session

3D Graphics Session

Maya Developers Session

Reimport

http://code.google.com/p/reimport/

- Detect modules changed on disk
- Reload any module or package
- Propagate changes to runtime

```
import reimport

reimport.reimport(module)
changed = reimport.modified()
reimport.reimport(*changed)
```