mvp

Release 0.2.2

July 20, 2016

Contents

I really needed this...

This module exists to unify and pythonify the various commands and apis necessary to manipulate Maya's 3D Viewports. These include, hardwareRenderGlobal attributes, modelPanel and modelEditor commands, as well as some key features of OpenMayaUI's M3dView class.

```
from mvp import Viewport

view = Viewport.active()
view.camera = 'top'
view.background = 0.5, 0.5, 0.5
view.nurbsCurves = False
```

CHAPTER 1

Features

- Unified api for manipulating Maya Viewports
- Get or set every viewport attribute all at once. Making it easy to restore a Viewport to a previous state.
- Easily set focus and playblast Viewports. Much more consistent than using active view.
- Draw text in Viewports using QLabels.
- Show identifiers in viewports, making it easy to grab the correct viewport at a glance.

CHAPTER 2

Get MVP

2.1 PyPi

MVP is available through the python package index as **mvp**.

```
pip install mvp
```

2.2 Distutils/Setuptools

```
git clone git@github.com/danbradham/mvp.git
cd mvp
python setup.py install
```

CHAPTER 3

Table of Contents

3.1 Guide

This section will grow shortly.

3.1.1 Getting the Active Viewport

3.1.2 Setting the Active Viewport

Identifying Viewports

Getting an Inactive Viewport

Setting focus

3.1.3 Manipulating Viewports

Setting the camera and background

Toggling visibility of node types

3.1.4 Copy a Viewport

3.1.5 Changing RenderGlobals

Ambient Occlusion

Antialiasing

Motion Blur

Depth of Field

3.1.6 Playblasting

3.2 API Documentation

3.2.1 Viewport

class mvp.**Viewport**(*m3dview*)

A convenient api for manipulating Maya 3D Viewports. While you can manually construct a Viewport from an OpenMayaUI.M3dView instance, it is much easier to use the convenience methods Viewport.iter, Viewport.active and Viewport.get:

```
# Get the active view
v = Viewport.active()
assert v.focus == True

# Assuming we have a second modelPanel available
# Get an inactive view and make it the active view
v2 = Viewport.get(1)
v2.focus = True

assert v.focus == False
assert v2.focus == True
```

Viewport provides standard attribute lookup to all modelEditor properties:

```
    # Hide nurbsCurves and show polymeshes in the viewport
    v.nurbsCurves = False
    v.polymeshes = True
```

Parameters **m3dview** – OpenMayaUI.M3dView instance.

classmethod active()
Get the active Viewport.

background
Get the background color of the Viewport

camera
Get the short name of the active camera.

classmethod clear_identifiers()
Remove all the QLabels drawn by show_identifiers.

close()
Close this viewport

copy()
Tear off a copy of the viewport.

Returns A new torn off copy of Viewport

static count()
The number of 3D Viewports.

depthOfField
Get active camera depthOfField attribute

draw_identifier(*text*)
Draws an identifier in a Viewport.

float()
Tear off the panel.

focus
Check if current Viewport is the active Viewport.

classmethod get(*index*)
Get the Viewport at index.

get_state()
Get a state dictionary of all modelEditor properties.

classmethod identify(*delay=2000*)
Shows identifiers in all Viewports:

```
    Viewport.identify()
```

Parameters **delay** – Length of time in ms to leave up identifier

index
Returns the index of the viewport

classmethod iter()
Yield all Viewport objects.

usage:

```
        for view in Viewport.iter():
            print v.panel
```

panel
Returns a panel name for the Viewport.

playblast (*filename*, ***kwargs*)
Playblasting with reasonable default arguments. Automatically sets this viewport to the active view, ensuring that we playblast the correct view.

Parameters

- **filename** – Absolute path to output file
- **kwargs** – Same kwargs as `maya.cmds.playblast()`

properties
A list including all editor property names.

set_state (*state*)
Sets a dictionary of properties all at once.

Parameters **state** – Dictionary including property, value pairs

classmethod show_identifiers ()
Draws QLabels indexing each Viewport. These indices can be used to with **:method:'get'** to return a corresponding Viewport object.

widget
Returns a QWidget object for the viewport.

window
Returns a QWidget object for the viewports parent window

3.2.2 RenderGlobals

mvp.**RenderGlobals**
alias of `<Mock name='mock.cmds.getAttr()' id='140370482855376'>`