



Assignment - Maya Nodes Part 2

Topics Covered

- Dynamic attribute on MPxNode

Assignments

For every Plug-in, we will provide an “Exercise” folder and a “Solution” folder. Each includes a Visual Studio solution and corresponding files. Solution folder includes the complete code for you to finish the Plug-in, it is for you to reference when you are stuck at problems when adding code into “Exercise” folder. In the “Exercise” folder, all the code you need to finish is specified with comments “// - TODO:”, you need to search for all the “TODO” comments and add your code there.

transCircleNode Plug-in

- **Topics Covered**
 - Write a custom node “transCircleNode” with MPxNode class
 - Add compound attribute using MFnNumericAttribute
 - Implement functions of transCircleNode to achieve the functionality so that the output attribute’s value is the value of inputTranslate plus the value of a circular movement based on current time frame.
- **Overview**
 - In this exercise, we will implement a custom node transCircleNode, it takes in two input attributes and one output:
A compound input translate attribute “inputTranslate”, composes of three elements: translateX, translateY, and translateZ
An input attribute “input”: current time
An input attribute “frames”: rotating speed (frames per circle)
An input attribute “scale”: decides the radius of the circle
An output a translate attribute “outputTranslate”, the value of outputTranslate is the value of inputTranslate plus the value of a circular movement based on current time frame.
- **Exercises**

- 1. Double click on “transCircleNode.py” to open the Plug-in, the skeleton of the transCircleNode has already been provided.
- 2. Implement transCircleNode.py, declare output attributes
- 3. Implement transCircleNode.py
Relevant classes and methods:

```
MFnCompoundAttribute::create() MFnCompoundAttribute::addChild()
MDataBlock::inputValue(), MDataBlock::outputValue()
```
- 4. Put AETransCircleTemplate.mel to
C:\My Documents\maya\2009\prefs\scripts

- **Result:**

Open a new scene, execute the following script in Script Editor:

```
createNode transCircle -n circleNode1;
sphere -n sphere1 -r 1;
sphere -n sphere2 -r 2;
connectAttr sphere2.translate circleNode1.inputTranslate;
connectAttr circleNode1.outputTranslate sphere1.translate;
connectAttr time1.outTime circleNode1.input;
```

You will see two nurbs sphere created and once you hit the “play” button, one sphere is rotating around the bigger sphere; you can also set the radius of the circle, and rotating speed by setting the values in attribute editor of circleNode1.

dynNode Plug-in

- **Topics Covered**
 - Create dynamic attribute using MFnNumericAttribute and add it in
MPxNode::postConstructor()
 - Set up affecting relationship between dynamic attribute and general attribute
- **Overview**
 - In this exercise, custom node “dynNode” has two attributes: “input” and “output”. We will add a dynamic attribute “dynAttr” on this class, also set up the affecting relationship so that the value of “output” is the sum of “input” and “dynAttr”.
- **Exercises**
 - 1. Double click on “dynNode.py” to open the Plug-in, the skeleton of the dynNode has already been provided.
 - 2. Implement dynNode.py, add necessary function declaration
Relevant classes and methods:

```
MPxNode::postConstructor()
MPxNode::setDependentDirty()
```

- 3. Implement dynNode.py, create a dynamic attribute, set up affecting relationship between it and output attribute, also in compute(), set up so that $\text{output} = \text{input} + \text{dynAttr}$

Relevant classes and methods:

```
MFnNumericAttr::create()
MPxNode::thisMObject()
MFnDependencyNode::addAttribute()
MDataBlock::inputValue()
MDataHandle::asFloat()
```

- **Result**

In script editor, execute:

```
createNode dynNode;
```

Open “Attribute Editor”, you will see there are three attributes listed, “input”, “output” and “dynAttr”. If you change “dynAttr” value, and refresh the attribute editor, you will see “output” value will be the value of “input” plus the value of “dynAttr”.