**Day 2 Assignment**

**April 14**

**Topics Covered**

* Build Maya Custom Node with MPxNode

**Assignments**

For every project, 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 project, 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.

**simpleNode project**

* **Topics Covered**
  + Write a skeleton of a custom node “simpleNode” with MPxNode class from scratch
  + Add simple attribute using MFnNumericAttribute class
* **Overview**
  + In this exercise, we will implement a custom node simpleNode, it has two attributes: “input”, ”output”

Whenever the “input” attribute changes value, the “output” attribute will always be the “input” attribute value multiplied by 2.

* **Exercises**
  + 1.Assuming you already copied the whole project under C:\MayaAPITraining, go to C:\MayaAPITraining\simpleNode\Exercise\
  + 2. Double click on simpleNode.sln to open the project, the skeleton of the simpleNode has already been provided.
  + 3. In simpleNode.h, adding declaration of “output” attributes and also declare your unique node ID
  + 4. In simpleNode.cpp, implement functions that are declared in simpleNode.h.

Relevant classes and methods:

MFnNumericAttribute::create()

MPxNode::attributeAffects(), MPxNode::addAttribute()

MDataBlock::outputValue(), MDataBlock::setClean()

MDataHandle::set()

* + 5. In pluginMain.cpp, implement both initializePlugin() and uninitializePlugin() functions to handle registration and de-registration of the simpleNode node.

Relevant classes and methods:

MFnPlugin:: registerNode()

MFnPlugin:: deregisterNode()

* **Result**

In script editor, execute:

createNode simpleNode;

Open “Attribute Editor”, you will see there are two attributes listed, “input”, “output”. If you change “input” value, “output” value will be 2\* input.

**transCircleNode project**

* **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.sln” to open the project, the skeleton of the transCircleNode has already been provided.
  + 2. Implement transCircleNode.h, declare output attributes
  + 3. Implement transCircleNode.cpp

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.