Autodesk®

Developer Technical Services

Assignment - Maya Commands

Topics Covered

- Build Maya custom command with MPxCommand
- Parse command arguments with MSyntax and MArgDatabase
- Support undo/redo of a custom command
- Use Iterator class to traverse selected objects
- Retrieve plug information with MFnDependencyNode and manipulate compound plug values
- Retrieve instancing information and matrix information with MDagPath

nodeInfoCmd Plug-in

- Topics Covered
    - Write a custom command "nodeInfoCmd" with MPxCommand class from scratch
    - Implement flag argument for MPxCommand with MSyntax class
    - Retrieve node type information and plug information with MFnDependencyNode and MPlug
- Overview
    - In this exercise, we will implement a custom command nodeInfoCmd, with an option to specify a flag "-quiet". By default, for all the selected nodes in the scene, it will print out all the node types and connected plugs information and also the node which is connecting to this selected node as a source. If "-quiet" is provided by user, the command will only print out the selected node type.
- Exercises
    - 1. Double click on "nodeInfoCmd.py"
    - 2. Implement nodeInfoCmd.py, add necessary function declaration
      Relevant classes and methods:
      MPxCommand::doIt()
    - 3. Implement nodeInfoCmd.py,
      Relevant classes and methods:
      MSyntax::addFlag(), MArgDatabase::isFlagSet()
      MGlobal::getActiveSelectionList()
      MItSelectionList::getDependNode()

MFnDependencyNode::MFnDependencyNode()
MFnDependencyNode::getConnections()
MPlug::connectedTo()

- 4. In nodeInfoCmd.py, implement both initializePlugin() and uninitializePlugin() functions to handle registration and de-registration of the nodeInfoCmd command.
  Relevant classes and methods:
  MFnPlugin:: registerCommand()
  MFnPlugin:: deregisterCommand()

- Result
  - In any Maya scene, select a node that has connections and in script editor, execute: nodeInfoCmd, it will print out the node type and all the connected plugs information and also print out the node which is the source of the connection.
  - You can also create the transCircle node and set it up as in the transCircle Plug-in, select transCircle node, you will be able to print out its connected plug information.

dagInfo Project

- Topics Covered
  - Write a custom command "dagInfo" with MPxCommand class
  - Retrieve instancing information and dag path with MFnDagNode
  - Have a better understanding on DAG hierarchy by getting inclusive and exclusive matrix of dag node
- Overview
  - In this exercise, we will implement a custom command dagInfo. For all the selected DAG nodes in the scene, it will print out the instance information, dag path and also inclusive and exclusive matrix.
- Exercises
  - 1. Double click on "dagInfo.py"
  - 3. Implement dagInfo.py,
    Relevant classes and methods:
    MGlobal::getActiveSelectionList()
    MItSelectionList::getDependNode()
    MFnDagNode::instanceCount()
    MDagPathArray:: getAllPaths()
    MDagPath:: fullPathName()
    MDagPath:: exclusiveMatrix(), MDagPath:: inclusiveMatrix()
    MObject::hasFn()
    MFnTransform::transformation()

- 4. In dagInfo.py, implement both initializePlugin() and uninitializePlugin() functions to handle registration and de-registration of the dagInfo command.
  Relevant classes and methods:
  MFnPlugin::registerCommand()
  MFnPlugin::deregisterCommand()

- Result
  - Open a scene file with multiple instances, for example, multiInstance.ma, select the shape node which has multiple instances, and execute:
    dagInfo
    The number of instances, exclusive matrix and inclusive matrix of this shape node will be printed out.
    If you select a transform node, it will also print out the local transformation matrix of this transform node.
  - For a shape node: Me = Mi
    For a transform node: Mi = Me * Mlocal ,
    Where Me represents exclusive matrix, Mi represents inclusive matrix, Mlocal represents local transformation matrix.
    For a shape node, which is parented under a transform node,
    Mi = Me = Me(transform) * Mlocal(transform)