

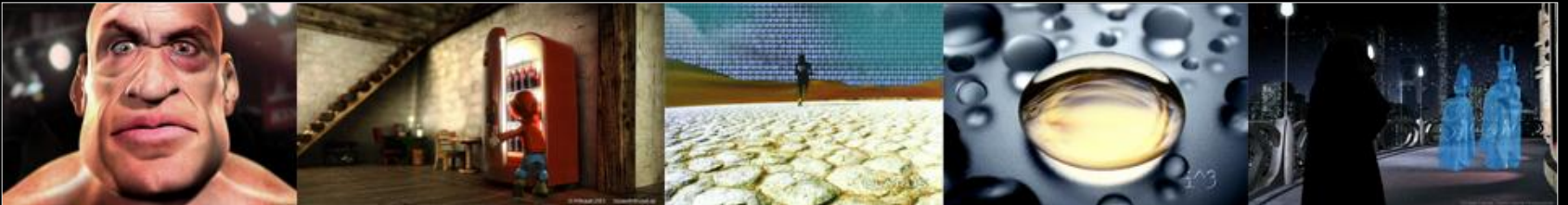
Maya Commands Arguments – Day 4

***Kristine Middlemiss**, Senior Developer Consultant
Autodesk Developer Network (ADN)*

Autodesk®

Agenda

- Adding Argument Syntax to Custom Commands



Adding Arguments



Command Arguments

- These flags and arguments are optional.
- Users can specify parameters to a Maya Command in two ways:
 - Command Flags and Flag Arguments
 - Command Arguments

Python Example: `cmds.headsUpDisplay("myHUD", s= 0, b=0, label="myHUD")`

MEL Example: `headsUpDisplay -s 0 -b 0 -label "myHUD" myHUD`

- Basic command structure:
 - `myCommand -myFlag <optionalFlagArg> <optionalCmdArg>`
- Can have multiple command flags and arguments but only one command arguments.

Custom Command

- The classes you need to work with when writing syntax objects are:
 - MSyntax
 - MArgDatabase
 - MArgParser
 - MArgList
- These classes are required for defining and handling command flag input.
- You must use the MSyntax and MArgParser classes to support arguments within a scripted MPxCommand.

Syntax Objects Classes

- MSyntax:
 - Used to specify flags and arguments passed to commands.
- MArgDatabase:
 - Used only to create an MArgDatabase object, then uses parent class functions (MArgParser) to work with the parsed objects.
- MArgParser:
 - Used to parse and store all of the flags, arguments and objects which are passed to a command.
- MArgList:
 - Used to create and hold the list of arguments that are passed to a command, this is where you can retrieve them from.

MSyntax

This newSyntax() method is used during the command registration into our plug-in:

- Syntax registration:

```
mplugin.registerCommand( kPluginCmdName, cmdCreator, syntaxCreator )
```

- Syntax definition:

```
cFlagShort = "-str"
```

```
cFlagLong = "-string"
```

```
def syntaxCreator():
```


```
    syntax = OpenMaya.MSyntax()
```


```
    syntax.addFlag( cFlagShort, cFlagLong )
```

```
    return syntax
```

MSyntax

How to retrieve arguments, that have been set by the user:

```
def dolt(self, args):  
    argData = OpenMaya.MArgDatabase(self.syntax(), args)  
    if( argData.isFlagSet( cFlagLong) ) :  
        argString = argData.flagArgumentString(cFlagLong, 0)  
    if (argData.isFlagSet(cFlagShort)):  
        argString = argData.flagArgumentString(cFlagShort, 0)
```

 MPxCommand::syntax()

Command with Arguments

- Retrieve arguments from MArgList

```
cmds.myCmd(q=True)
```

```
#- This is a method which will be used by our command dolt() method to
```

```
#- check arguments passed into our command call.
```

```
def parseArgs ( self, args ):
```

```
    argData = OpenMaya.MArgDatabase( self.syntax(), args)
```

```
    if argData.isFlagSet(kQuietFlag):
```

```
        self.quiet = 1
```

```
def dolt(self, args):
```

```
    self.parseArgs ( args )
```

Help on Custom Command

- Automatic Help

```
help myCmd;
```

- Adding your custom help

```
helpFlagShort = "-h"
```

```
helpFlagLong = "-help"
```

```
helpText = "\nThe myCmd command is used to ....."
```

```
def syntaxCreator():
```

```
    syntax = OpenMaya.MSyntax()
```

```
    syntax.addFlag(helpFlagShort , helpFlagLong)
```

```
def doIt(self, args):
```

```
    argData = OpenMaya.MArgDatabase(self.syntax(), args)
```

```
    if argData.isFlagSet(helpFlagShort) :
```

```
        self.setResult( helpText )
```

```
        return OpenMaya.MStatus.kSuccess
```

Autodesk