Autodesk MotionBuilder 2013

Programming in MotionBuilder || Focusing on Python

Naiqi Weng, Developer Consultant
Autodesk Developer Network

Module 3

MotionBuilder Architecture

Module 3

Module's Agenda

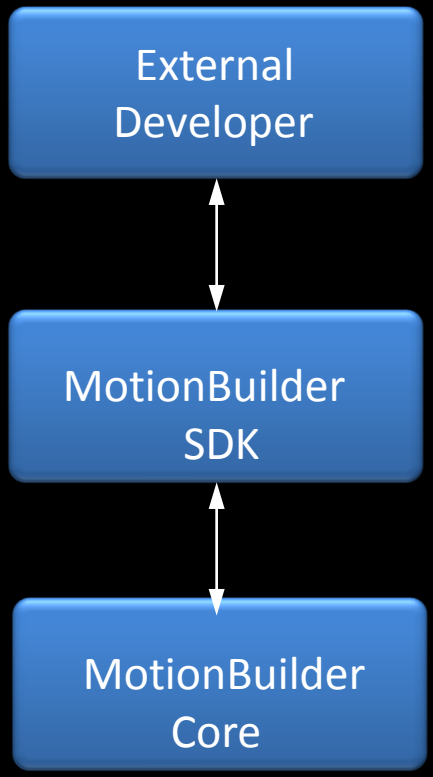
- MotionBuilder Architecture
- MotionBuilder Classes Introduction
- Reviewing the Python Documentation Closer
- Working with Callbacks
- More Control over Loading and Saving Scenes
- Assignment

MotionBuilder Architecture

- Real-time Engine
  - Real-time architecture dividing different functionality into different tasks with different priorities
  - Multi-threaded engine optimized for real-time performance
  - 'Don’t call us, we’ll call you' philosophy, any objects created get registered automatically

MotionBuilder SDK

- A tight wrapper around the internal architecture

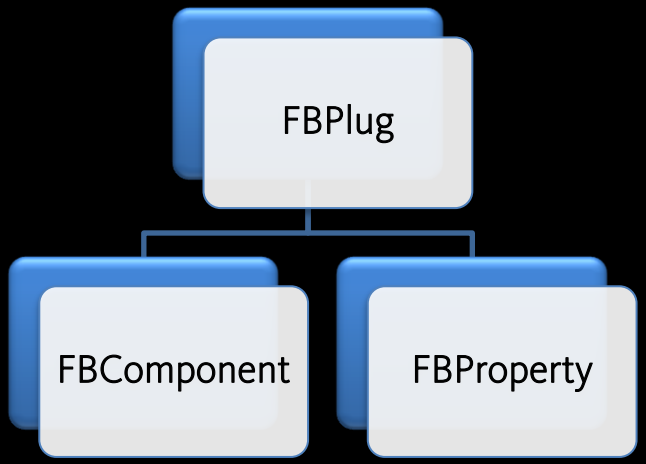


MotionBuilder SDK Classes

- These are main basic categories of classes:
  - Base Classes
  - Elements
  - Animation
  - User Interface
  - Plug-in Types
    - Boxes, Devices, Constraints, Tools, Manipulators,etc…
  - Utilities

Base Classes

- Fundamentals of the python SDK
- Define the basic structure



Base Classes : FBComponent

- FBComponent is the base entity class for most objects in the MotionBuilder SDK. It defines the basic memory management methods for object creation and destruction.

Base Class: FBProperty

- This class is the base property class. Most of the time, the state and behavior of an FBComponent object is defined by all of its properties, which are represented by various instances of FBProperty

Base Class: FBPlug

- It is the common base class for FBComponent and FBProperty. The FBPlug class is responsible for managing the connections among components and properties.

SceneElements

- Scene Elements are the basic building blocks of MotionBuilder. They consist of:
  - Models
  - Materials
  - Cameras
  - Lights
  - Characters
  - Etc.

Animation

- Takes(FBTake): containers for animation in a scene. A take stores data about animation for objects.
- Animation Node (FBAnimationNode): channels to input and output animation data on a Box object
- Fcurve (FBFCurve): Curve which records keying information for animation

User Interface

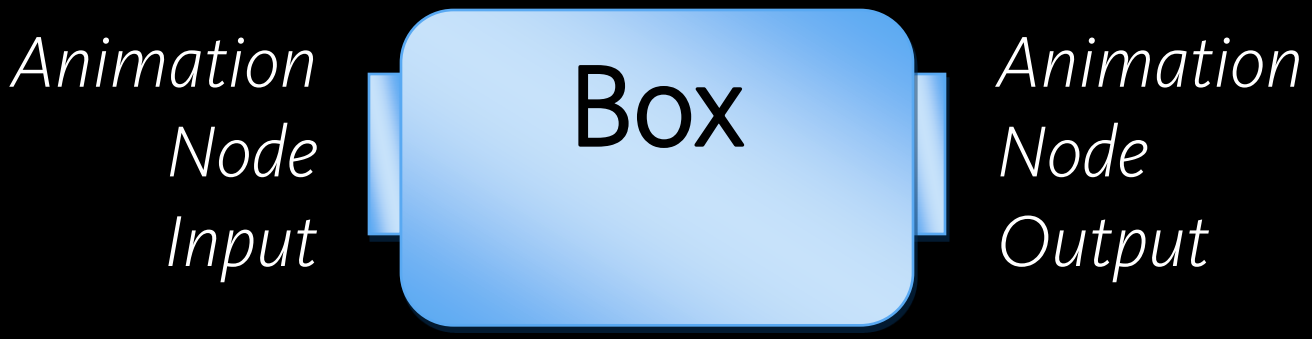
- All MotionBuilder UI controls are exposed to Python
- They behave mostly like their OpenReality C++ counterparts
- To help with UI building, helper classes are provided
- These helper classes are defined in the module pyfbsdk_additions

Plug-in Types: Boxes

- MotionBuilder is built on the concept of a ‘box’
- Most elements in MotionBuilder inherit the functionality of this item, which is to:
  - Read data from the input nodes.
  - Transform the data based on certain rules.
  - Send the data onto the output nodes.

Plug-in Types: Boxes

- These boxes appear in the Relations Constraint interface
- FBBox is the base class for anything that is Animatable



Plug-in Types : Devices

- Creating custom devices is not exposed in Python
- You can access and set data in existing devices (FBDevice)
- Devices fall into these categories:
  - Input
  - Optical
  - Output

Plug-in Types : Constraints

- Constraints are real-time evaluation conditions for elements in a scene
- They constrain the movement and behaviour of objects on input criteria
- They can be expressed in the form of formulae, positional information, hierarchal relationships, etc.
- Two forms: simple and complex

Plug-in Types : Tools

- A tool is a new user interface layout adding to the current functionality of the software
- Using the extensive user interface elements, the possibilities are limitless as to what tools can be added

Plug-in Types : Manipulators

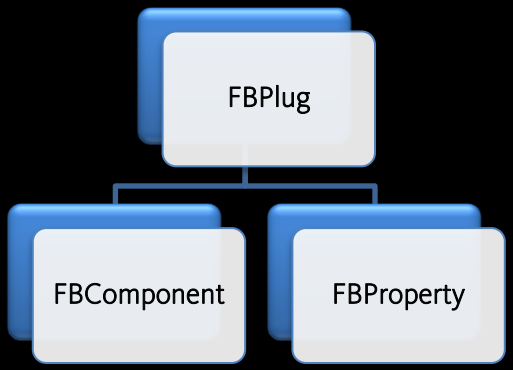
- Creating and accessing custom manipulators is currently not exposed in Python
- A manipulator is similar to a tool in usage, but bridges the gap between a tool's functionality and interaction with the 3D viewer
- Manipulators are found in the Transforms window in MotionBuilder (Transformation, Parenting, and so on)

Utility Classes

- The Python SDK comes with a series of utility classes
- They are to manipulate data or complete any of the normal functions that might be, such as:
  - Transport Control: FBPlayerControl
  - Media: FBVideo
  - System Interface: FBApplication, FBSystem
  - Etc.

Reviewing the Python Documentation Closer

- 45 standalone classes
- 1 enumeration parent class
- 1 base objects class that holds all the Python data types, tuple, list, str, etc
- Everything else is in the Parent class FBPlug



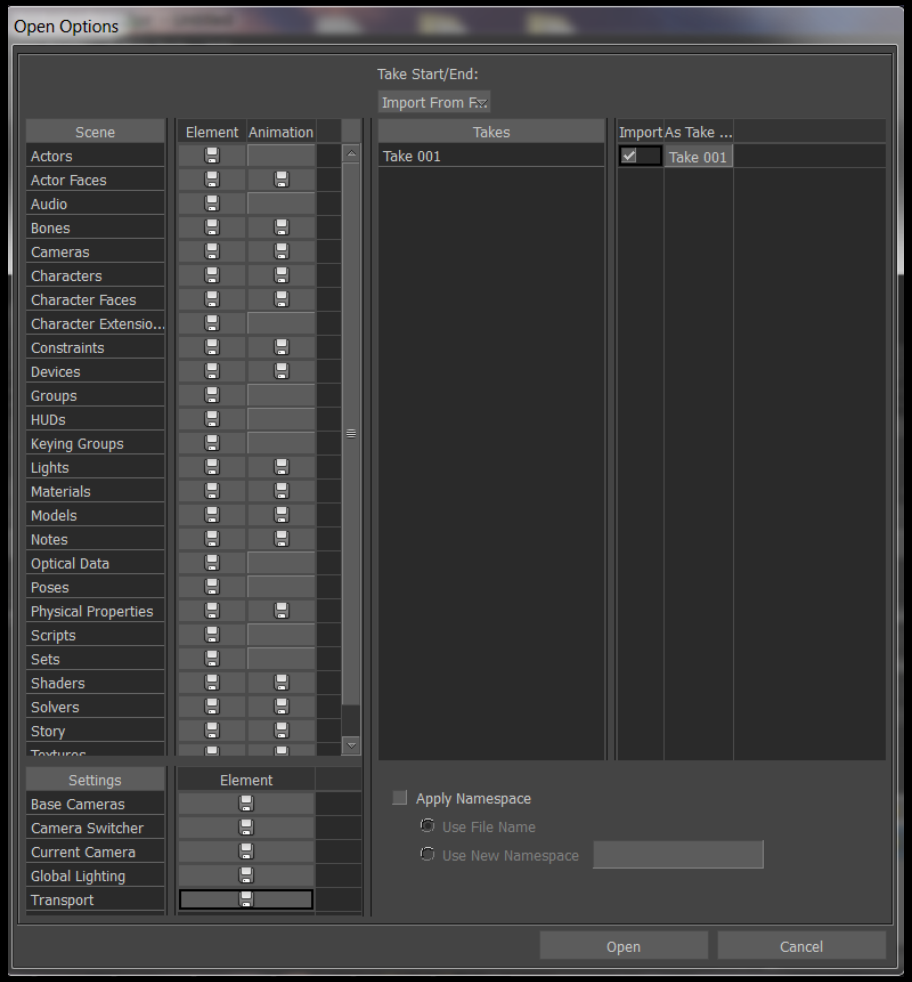
Working with Callbacks

- FBEvent: Base class for all system events that can be watched
- Key words to register callback and remove callback
  - add
  - Remove
- Callback signature: FunctionName(eventsource, event)

More Control over Loading and Saving Scenes

- FBFolderPopup
  - create a UI dialog for opening a folder
- FBFilePopup
  - creates a UI dialog box for opening/saving files.
- FBFilePopupStyle
  - kFBFilePopupOpen
  - kFBFilePopupSave

More Control over Loading and Saving Scenes



Autodesk®

More Control over Loading and Saving Scenes

- The ‘FBApplication’ Class
  - Deprecated FBFbxManager class
- FBFbxOptions
  - FBFbxOptions(True) - load scene data
  - FBFbxOptions(False)- save scene data
- FBElementAction determines what operations you want to do with scene elements.

Assignment Three

- Batch Loading Animation Script
- Description:
  - Allow the user to select a character file
  - Opened the file in a new scene
  - Allow the user to select a folder that contains animation
  - Load each FBX file in this folder into MotionBuilder one at a time, merging only the animation in these files into the current scene

Next Agenda

- Elements in the Scene
- MotionBuilder Data Types
- Custom Properties on Elements
- Groups and Sets
- Assignment