



Autodesk MotionBuilder 2013

Programming in MotionBuilder || Focusing on Python

Naiqi Weng, Developer Consultant
Autodesk Developer Network

Module 4



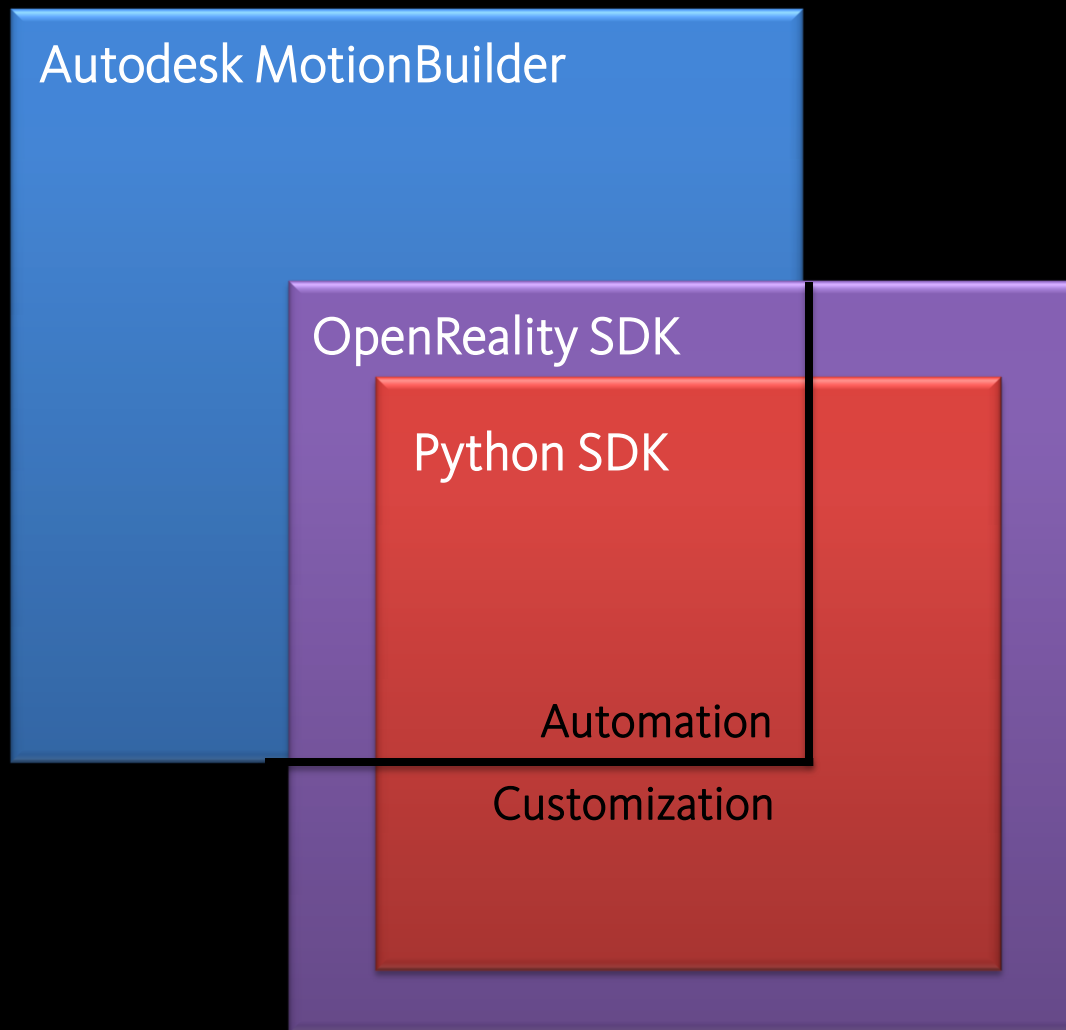
Elements and Properties in the Scene – Part 2

Module 4

Module's Agenda

- Last Week Review
- A Closer Look at Properties
- MotionBuilder Data Types
 - Python Built-in Data Type
 - MoBu Built-in Data Type
- Groups and Sets
- Assignment Five

Review



Review : File I/O Class

- FBApplication
 - File I/O and application utility class **providing** similar functionality to the "File" menu in MotionBuilder
- FBFilePopup, FBFolderPopup
 - create a UI dialog box for selecting files or directories/folders
- FBFbxOptions
 - customize file loading and saving options

Review: MotionBuilder Classes

- Base Classes : FBComponent, FBProperty, FBPlug
- Scene Elements
- Animation
- User Interface
- Plug-in Types
- Utilities
 - FBSystem, FBApplication

Review : Scene Elements

- Elements in the scene : FBModel
- Work with existing elements:
 - FBSystem().Scene.Cameras
 - FBSystem().Scene.Components
 - FBSystem().Scene.RootModel
 - Global functions:
 - FBFindObjectByFullName()
 - FBFindModelByLabelName()
 - Etc...
- Delete elements:
 - FBDeleteObjectsByName()
 - FBDelete, FBDestory

A Closer Look at Properties

- FBProperty defines the state and behavior of an FBComponent object
- FBProperty
 - “Public Attributes” on the class documentation

```
cube = FBModelCube('myCube')  
translationVector = cube.Translation
```

- FBComponent.PropertyList.Find()

```
cube = FBModelCube('myCube')  
translationProperty = cube.PropertyList.Find('Translation')  
translationVector = translationProperty.Data
```


A Closer Look at Properties

- Cannot Find Properties Based off the UI Names
- Setting a Property as Animatable or Keyable
- Custom Properties on Objects



Cannot Find Properties Based off the UI Names

- When you can't find the property based on the UI name, set the property in the UI to something distinct.
- Then save the scene as ASCII, using a text editor search the ASCII FBX file for the distinct value.
- If you can't find the property by using the UI name, generally it won't have any spaces in the property name so that will help when searching the ASCII FBX.



Setting a Property as Animatable or Keyable

- Animatable properties are properties which are represented in numeric values and can be recorded by a curve representing different values at different time.
- `FBProperty::IsAnimatable()`
- `FBPropertyAnimatable` is the base class of all animatable properties



Standard Python Built-in Data Types used in MoBu

- object
 - bool
 - float
 - int
 - list
 - long
 - str
 - tuple
- Enumeration

MoBu Built-in Data Types

- FBVector2d, FBVector3d, FBVector4d
- FBSVector
- FBColor
- FBColorAndAlpha
- FBMatrix
- FBModelList
- FBTime

Custom Properties on Objects

- `FBComponent::PropertyCreate()` is used for creating and adding custom properties to objects
- In the class `FBPropertyType`, we can see the types of custom properties
- `FBComponent::PropertyRemove()` is used to remove custom properties

Groups and Sets

- You can create groups and sets including models, elements, lights, materials, shaders, markers, nodes, textures, constraints and even other groups and sets.
- Groups and sets let you create custom associations between objects or assets
- This way, you can select a pre-defined collection of several objects every time you want to work with them
- Once stored in a group or a set, you can manipulate the group or set instead of each individual object
- 'FBGroup', 'FBSet' Classes

Groups vs. Sets

- Group not exclusive, meaning the same object can appear in many groups. Generally groups are used for the selection and organization of items in the scene
- Sets are exclusive, meaning an object can only appear in one set, so you cannot copy an object from one set to another, you need to move it from one set to another

Assignment Con'd

- Create a script for each of these character assets:
 - Actor Face
 - Actor
 - Character Extension
 - Character Face
 - Character
- Create custom property on those assets.