Duik

//TODO explications low level, mid level, high level des méthodes

Attributes

string *Duik.version*
float *Duik.versionNumber*
boolean *Duik.forceReload*

Name	Type	Description
version	string, read-only	Version string of libDuik
versionNumber	float, read-only	Version number of libDuik
forceReload	boolean, read-only during run time	When true, forces libDuik to be reloaded each time it is included in a script. When false, libDuik loads only on first run and then stays available until After Effects is shut down. This attribute should not be changed, unless you're editing libDuik.jsxinc itself and you need to do some testing and reload it without shutting down After Effects. To speed up launching of your scripts even at first launch, instead of including *libDuik.jsxinc*, you can copy it in the *Startup* subfolder of the *Scripts* folder of After Effects, and rename it to *libDuik.jsx*. This way, libDuik will be available to all scripts without *#include* macro. This attribute must be set by directly by editing libDuik.jsxinc to work.

Classes

Duik.uiString
Duik.settings
Duik.utils

Name	Description
uiStrings	Contains all string names used by effects created by Duik. You can set these strings to translate libDuik at runtime. Default values are English names.
settings	Access to settings used by Duik.
utils	Some useful tools

Methods

//TODO tri par level

Low-level methods are listed below (greyed) but they are not documented.
If you do not understand what low-level methods do by reading them in *libDuik.jsxinc*, you shouldn't need them.

Duik.addIK(controller, layer1, layer2, layer3, goal, clockWise, threeD, frontFacing)
Duik.addGoal(layer, controller)
Duik.addController(layer)
Duik.addControllers(layers)
Duik.addOneLayerIK(controller,layer)
Duik.addTwoLayerIK(threeD,controller,root,end,clockWise,frontFacing)
Duik.addWiggle(layer,property,all,x,y,z)
Duik.add3DWiggle(layer,property,x,y,z)
Duik.add2DWiggle(layer,property,x,y)
Duik.add1DWiggle(layer,property)
Duik.exposureControls(layer,property,adaptative,precision,exposureLimits)

Name	Description	Return	Level
addIK(controller, layer1, layer2, layer3, goal, clockWise, threeD, frontFacing)	Adds IK on the layers	true if successful, false if anything went wrong	mid
addGoal(layer, controller)	Adds a goal effect to the layer, which may be controlled by a controller	true if successful, false if anything went wrong	mid
addController(layer)	Creates a null object (controller) at layer position and named by layer.name	AVLayer; controller	mid
addControllers(layers)	For each layer, Creates a null object (controller) at layer position and named by layer.name	Array of AVLayer; controllers	High
addWiggle(layer, property, all, x, y, z)	Adds a wiggle effect to given property	true if successful, false if anything went wrong	Mid
exposureControls(layer, property, adaptative, precision, minExp, maxExp)	Adds exposure controles to given property	true if successful, false if anything went wrong	mid

Duik.addIK(controller, layer1, layer2, layer3, goal, clockWise, threeD, frontFacing)

mid-level method.

Adds IK on the layers

parameters:

controller | AVLayer
layer1 | AVLayer
layer2 | AVLayer or undefined
layer3 | AVLayer or undefined
goal | AVLayer or undefined
clockWise | boolean, used only with two-layer and three-layer IK, default: false
threeD | boolean, works only with two-layer IK, default: false
frontFacing | boolean, default: false

returns

true if successful, false if anything went wrong

Duik.addGoal(layer, controller)

mid-level method.

Adds a goal effect to the layer, which may be controlled by a controller

parameters:

layer | AVLayer
controller | AVLayer or undefined

returns

true if successful, false if anything went wrong

Duik.addController(layer)

mid-level method.

Creates a null object (controller) at layer position and named by layer.name

parameters

layer | AVLayer

returns

AVLayer controller

Duik.addControllers(layers)

high-level method.

For each layer,
Creates a null object (controller) at layer position and named by layer.name

parameters

layers | Array of AVLayer

returns

Array of AVLayer controllers

Duik.addWiggle(layer, property, all, x, y, z)

mid-level method.

Adds a wiggle effect to given property.

parameters

layer | AVLayer of the property
property | Property
all | boolean, true to apply the same wiggle to all dimensions, default: true
x | boolean, default: false
y | boolean, default: false
z | boolean, default: false

returns

true if successful, false if anything went wrong

Duik.exposureControls(layer, property, adaptative, precision, minExp, maxExp)

mid-level method.

Adds exposure controls to given property.

parameters

layer | AVLayer of the property
property | Property
adaptative | boolean, default: true
precision | float, default: 100
minExp | integer, default : 1, minimum exposure
maxExp | integer, default : 4, maximum exposure

returns

true if successful, false if anything went wrong

Duik.uiStrings

Contains all string names used by effects created by Duik.
You can set these strings to translate libDuik at runtime.
Default values are English names.

Attributes

Duik.uiStrings.ikFkBox
Duik.uiStrings.reverseBox
Duik.uiStrings.fkAngle
Duik.uiStrings.rootFkAngle
Duik.uiStrings.endFkAngle
Duik.uiStrings.goalFkAngle
Duik.uiStrings.weightSlider
Duik.uiStrings.xAmp = "X Amp.";
Duik.uiStrings.xFreq = "X Freq.";
Duik.uiStrings.yAmp = "Y Amp.";
Duik.uiStrings.yFreq = "Y Freq.";
Duik.uiStrings.zAmp = "Z Amp.";
Duik.uiStrings.zFreq = "Z Freq.";
Duik.uiStrings.amp = "Amp.";
Duik.uiStrings.freq = "Freq.";

Name	Type	Description
ikFkBox	string	*"IK / FK"*
reverseBox	string	*"Reverse"*
fkAngle	string	*"FK"*
rootFkAngle	string	*"Root FK"*
endFkAngle	string	*"End FK"*
goalFkAngle	string	*"Goal FK"*
weightSlider	string	"*LookAt Weight"*
xAmp	string	*"X Amp."*
xFreq	string	*"X Freq."*
yAmp	string	*"Y Amp."*
yFreq	string	*"Y Freq."*
zAmp	string	*"Z Amp."*
zFreq	string	*"Z Freq."*
amp	string	*"Amp."*
freq	string	*"Freq."*

Duik.settings

Access to settings used by Duik.

Attributes

Duik.settings.controllerSize
Duik.settings.controllerSizeAuto
Duik.settings.controllerSizeHint

Name	Type	Description	Default
controllerSize	integer	Size of controllers in pixels	100
controllerSizeAuto	boolean	If true, controller sizes will be automatically adapted to comp size, according to *Duik.settings.controllerSizeHint*	true
controllerSizeHint	integer	when controllerSizeAuto is true, 0 = small, 1 = medium, 2 = big	1

Duik.utils

Some useful methods.

Methods

Duik.utils.preparеProperty(property,isFX,index,depth,parentName)
Duik.utils.getPropertyDimensions(property)
Duik.utils.getLength(value1,value2)
Duik.utils.getAverageSpeed(layer,property)
Duik.utils.addPseudoEffect(layer,pseudoEffectName)

Name	Description	Return
prepareProperty(property, isFX, index, depth, parentName)	Prepares property to be rigged	true if property can set expression, false otherwise
getPropertyDimensions(property)	Gets the dimensions of the property (1, 2 or 3), taking care of 2D layer positions (reported as 3D by AFX, but to be considdered as 2D)	integer, number of dimensions
getLength(value1, value2)	Gets the length between the values, whichever dimensions they are	float, length between the values
getAverageSpeed(layer, property)	Gets the average speed of the animated property, between its first and last keyframe only	float, average speed of the property
addPseudoEffect(layer, pseudoEffectName)	Adds a Duik predefined pseudo effect to the layer	Property, the effect added

Duik.utils.preparеProperty(property,isFX,index,depth,parentName)

Prepare the given property to be rigged.
isFX, *index*, *depth*, *parentName* will be filled by the method with the values corresponding to this property.

parameters:

property | Property
isFX | boolean
index | integer
depth | integer
parentName | string

returns

true if property can set expression, false otherwise

Duik.utils.getPropertyDimensions(property)

Gets the dimensions of the property (1, 2 or 3), taking care of 2D layer positions (reported as

3D by AFX, but to be considdered as 2D)

parameters:

property | Property

returns

integer, number of dimensions

Duik.utils.getLength(value1, value2)

Gets the length between the values, whichever dimensions they are

parameters:

value1 | float or Array of float, first coordinates
value1 | float or Array of float, second coordinates

returns

float, length between the values

Duik.utils.getAverageSpeed(layer, property)

Gets the average speed of the animated property, between its first and last keyframe only.

parameters:

layer | AVLayer of the property
property | Property

returns

float, average speed of the property

Duik.utils.addPseudoEffect(layer, pseudoEffectName)

Adds a Duik predefined pseudo effect to the layer. The AFX preset file of the pseudo effect must be located in the ScriptUI Panels folder and called « Duik_ » + pseudoEffectName + « .ffx ». In the preset, the effect must be called pseudoEffectName.

parameters:

layer | AVLayer
pseudoEffectName | string, name of the pseudo effect

returns

Property, the effect added