PiXYZ SOFTWARE

STUDIO

CHANGELOG Python API

2018.2r30 to 2018.3r35

PiXYZ Software

Python API changelog 2018.2r30 to 2018.3r35

contact@pixyz-software.com

Table of content

A - NEW FUNCTIONS

Module: core

askString

askString(msg[, defaultValue]) -> text

Invite the user to enter a string

Parameters:

msg (String) : Message to display

defaultValue (String) : Message to display [optional] (default: "")

Return value:

text (String) : The string entered by the user

askYesNo

askYesNo(question[, defaultValue]) -> answer

Ask a question which need a Yes/No answer

Parameters:

question (String) : Question to display

defaultValue (Boolean) : Default value (if interfactive mode is disabled) [optional] (default: false)

Return value:

answer (Boolean) : True if the user say Yes, else False

choose

choose(message, values[, defaultValue]) -> choice

Invite the user to choose one value between multiple choice

Parameters:

message (String) : Message to display

values (StringList) : Possible values to choose

defaultValue (Int) : Default value index [optional] (default: 0)

Return value:

choice (Int) : Index of the choosen value in the values list

message

message(msg)

Display a message (or a MessageBox in GUI)

Parameters:

msg (String) : Message to display

setInteractiveMode

setInteractiveMode([interactive])

Switch between interactive mode and non-interactive mode, UI functions will no ask user on non-interactive mode and will return default values

Parameters:

interactive (Boolean) : True if you want to enter interactive mode, else False [optional] (default: true)

isInteractiveMode

isInteractiveMode() -> interactive

Returns True if the script is in interactive mode, else returns false

Return value:

interactive (Boolean) : True if interactive, else false

getColorFromIndex

getColorFromIndex(index) -> color

Returns a unique color associated with an index

Parameters:

index (Int) : Index of the color (index must be less than 2^24)

Return value:

color (Color) : The unique color associated to the given index

Module: geom

changeOfBasisMatrix

changeOfBasisMatrix(origin, x, y, z) -> changeOfBasis

Construct a Change of Basis Matrix (e.g multiplying the point [0,0,0] will result to the point origin)

Parameters:

origin (Point3) : Origin of the new basis

x (Vector3) : X axis of the new basis

y (Vector3) : Y axis of the new basis

z (Vector3) : Z axis of the new basis

Return value:

changeOfBasis (Matrix4) : The change of basis matrix

invertMatrix

invertMatrix(matrix) -> inverted

Invert a matrix

Parameters:

matrix (Matrix4) : The matrix to invert

Return value:

inverted (Matrix4) : The inverted matrix

multiplyMatrices

multiplyMatrices(left, right) -> result

Multiply two matrices, returns left*right

Parameters:

left (Matrix4) : Left side matrix

right (Matrix4) : Right side matrix

Return value:

result (Matrix4) : Result of the matrices multiplication

multiplyMatrixPoint

multiplyMatrixPoint(matrix, point) -> result

Multiply a point by a matrix (i.e apply the matrix to a point)

Parameters:

matrix (Matrix4) : The matrix to apply

point (Point3) : The point to multiply

Return value:

result (Point3) : The resulting point

multiplyMatrixVector

multiplyMatrixVector(matrix, vector) -> result

Multiply a vector by a matrix (i.e apply the matrix to a vector)

Parameters:

matrix (Matrix4) : The matrix to apply

vector (Vector3) : The vector to multiply

Return value:

result (Vector3) : The resulting point

Module: algo

decimateTarget

decimateTarget(scenePaths, targetTriangleCount[, removeSmallParts])

Experimental :reduce the polygon count by removing some vertices to obtain a target triangle count

Parameters:

scenePaths (ScenePathList) : Scene paths of components to process

targetTriangleCount (Int) : Target triangle count

removeSmallParts (Boolean) : If true, allows suppression of small parts [optional] (default: true)

replaceByConvexHull

replaceByConvexHull(scenePaths)

Experimental :Replace an object by its convex hull

Parameters:

scenePaths (ScenePathList) : ScenePath of components to replace

Module: material

createMaterialFromExtract

createMaterialFromExtract(extract[, images]) -> material

Create material from a material extract

Parameters:

extract (MaterialExtract) : The structure containing all the material informations

images (ImageDefinitionList) : Definition of images used by material textures, image must have been imported first [optional] (default:)

Return value:

material (Material) : The created material

getImageDefinition

getImageDefinition(image) -> definition

Returns the raw data of an image

Parameters:

image (Image) : The image

Return value:

definition (ImageDefinition) : Definition of the image

importImageDefinition

importImageDefinition(imageDef) -> image

Import an image from its raw data

Parameters:

imageDef (ImageDefinition) : The image definition

Return value:

image (Image) : The created image

Module: scene

getOBB

getOBB(paths) -> obb

Returns the Oriented Bounding Box of a list of scene paths (works only on meshes, fast method, not the Minimum Volume Box)

Parameters:

paths (ScenePathList) : List of paths to retrieve the AABB

Return value:

obb (OBB) : The oriented bounding box of all given paths

createSceneFromExtract

createSceneFromExtract(extract)

Create a scene from a scene extract (cf. getSceneExtract)

Parameters:

extract (SceneExtract) : The structure containing all the scene informations

getSceneExtract

getSceneExtract(root[, returnMeshes[, returnProperties[, returnImages[, matrixMode[, materialsMode[, visibilityMode]]]]]]) -> scene

Returns an extract of the whole scene in one call

Parameters:

root (Occurrence) : The root occurrence of the sub-scene wanted

returnMeshes (Boolean) : If true, meshes are returned in the SceneExtract [optional] (default: true)

returnProperties (Boolean) : If true, properties and instanceProperties are returned in the SceneExtract [optional] (default: true)

returnImages (Boolean) : If true, image of textures are returned in the SceneExtract [optional] (default: true)

matrixMode (MatrixExtractMode) : Define how matrices are returned, see doc of MatrixExtractMode [optional] (default: 1)

materialsMode (MaterialExtractMode) : Define how materials are returned, see doc of MaterialExtractMode [optional] (default: 1)

visibilityMode (VisibilityExtractMode) : Define how visibilities are returned, see doc of VisibilityExtractMode [optional] (default: 1)

Return value:

scene (SceneExtract) : The structure containing all wanted informations

moveOccurrence

moveOccurrence(occurrence, parent)

Move an occurrence under another occurrence (new parent)

Parameters:

occurrence (Occurrence) : Occurrence to move

parent (Occurrence) : The new parent occurrence (the new parent must be an assembly or an instance, if it is an instance, the old reference will not be instanciated anymore by the instance)

replaceMaterialsOnSelection

replaceMaterialsOnSelection(selection, oldMaterial, newMaterial)

Replace a material by another anywhere it is used in a selection

Parameters:

selection (ScenePathList) : The selection scenePaths list

oldMaterial (Material) : The material to replace

newMaterial (Material) : The new material to set

setInstanceRef

setInstanceRef(instance, reference)

Change the reference of an instance

Parameters:

instance (Instance) : The instance to modify

reference (Component) : The new component pointed to by the instance

B - FUNCTIONS MODIFIED

Module: algo

normalizeUV

normalizeUV(scenePaths, srcUVChannel[, dstUVChannel[, uniform[, sharedUVSpace[, **ignoreNullIslands**]]]])

Normalize UVs to fit in the [0-1] uv space

Parameters:

scenePaths (ScenePathList) : Scene paths of part to process

srcUVChannel (Int) : UV Channel to normalize

dstUVChannel (Int) : UV channel to store the normalized UV (if -1, srcUVChannel will be replaced) [optional] (default: -1)

uniform (Boolean) : If true, the scale will be uniform. Else UV can be deformed with a non-uniform scale [optional] (default: true)

sharedUVSpace (Boolean) : If true, all parts will be processed as if they were merged to avoid overlapping of their UV coordinates [optional] (default: true)

ignoreNullIslands (Boolean) : If true, islands with null height and width will be ignored and their UV coordinates will be set to [0,0] (Slower if enabled) [optional] (default: false)

repackUV

repackUV(scenePaths[, channel[, shareMap[, resolution[, padding[, uniformRatio[, iterations[, **removeOverlaps**]]]]]]]) -> **failedParts**

Pack existing UV (create atlas)

Parameters:

scenePaths (ScenePathList) : Scene paths of part to process

channel (Int) : The UV channel to repack [optional] (default: 0)

shareMap (Boolean) : If True, the UV of all given parts will be packed together [optional] (default: true)

resolution (Int) : Resolution wanted for the final map [optional] (default: 1024)

padding (Int) : Set the padding (in pixels) between UV islands [optional] (default: 2)

uniformRatio (Boolean) : If true, UV of different part will have the same ratio [optional] (default: false)

iterations (Int) : Fitting iterations [optional] (default: 3)

removeOverlaps (Boolean) : Remove overlaps to avoid multiple triangles UVs to share the same pixel [optional] (default: true)

Return value:

failedParts (PartList) : Parts which failed to be repacked at this resolution (try to increase the resolution or decrease the padding)

smartHiddenCreateVisibilityInformation

smartHiddenCreateVisibilityInformation(scenePaths, voxelSize, minimumCavityVolume, resolution[, **mode**])

Create visilibity information on parts viewed from a set of camera automatically generated

Parameters:

scenePaths (ScenePathList) : Scene paths of components to process

voxelSize (Distance) : Size of the voxels in mm (smaller it is, more viewpoints there are)

minimumCavityVolume (Volume) : Minimum volume of a cavity in cubic meter (smaller it is, more viewpoints there are)

resolution (Int) : Resolution of the visibility viewer

mode (SmartHiddenType) : Select where to place camera (all cavities, only outter or only inner cavities) [optional] (default: 0)

smartHiddenRemoval

smartHiddenRemoval(scenePaths, level, voxelSize, minimumCavityVolume, resolution[, **mode**])

Delete parts, patches or polygons not viewed from a set of camera automatically generated

Parameters:

scenePaths (ScenePathList) : Scene paths of components to process

level (SelectionLevel) : Level of parts to remove : Parts, Patches or Polygons

voxelSize (Distance) : Size of the voxels in mm (smaller it is, more viewpoints there are)

minimumCavityVolume (Volume) : Minimum volume of a cavity in cubic meter (smaller it is, more viewpoints there are)

resolution (Int) : Resolution of the visibility viewer

mode (SmartHiddenType) : Select where to place camera (all cavities, only outter or only inner cavities) [optional] (default: 0)

smartHiddenSelection

smartHiddenSelection(scenePaths, voxelSize, minimumCavityVolume, resolution[, **mode**])

Select parts not viewed from a set of camera automatically generated

Parameters:

scenePaths (ScenePathList) : Scene paths of components to process

voxelSize (Distance) : Size of the voxels in mm (smaller it is, more viewpoints there are)

minimumCavityVolume (Volume) : Minimum volume of a cavity in cubic meter (smaller it is, more viewpoints there are)

resolution (Int) : Resolution of the visibility viewer

mode (SmartHiddenType) : Select where to place camera (all cavities, only outter or only inner cavities) [optional] (default: 0)

Module: scene

findByActiveMaterial

findByActiveMaterial(material[, **roots**]) -> paths

Find all part occurrence with a given material as active material (i.e. as seen in the rendering)

Parameters:

material (Material) : A material

roots (ScenePathList) : If specified, restrict the search from the given roots [optional] (default:)

Return value:

paths (ScenePathList) : Scene paths or part with the given material as active material

findByAllProperty

findByAllProperty(property, regex[, **roots**]) -> paths

Returns all scene paths which a property value or an occurrence property matches the given regular expression (ECMAScript)

Parameters:

property (String) : Property name

regex (Regex) : Regular expression (ECMAScript)

roots (ScenePathList) : If specified, restrict the search from the given roots [optional] (default:)

Return value:

paths (ScenePathList) : Scene paths which matches the given regular expression

findByName

findByName(regex[, **roots**]) -> paths

Returns all scene paths which name matches the given regular expression (ECMAScript)

Parameters:

regex (Regex) : Regular expression (ECMAScript)

roots (ScenePathList) : If specified, restrict the search from the given roots [optional] (default:)

Return value:

paths (ScenePathList) : Scene paths which matches the given regular expression

findByOccurrenceProperty

findByOccurrenceProperty(property, regex[, **roots**]) -> paths

Returns all scene paths which an occurrence property value matches the given regular expression (ECMAScript)

Parameters:

property (String) : Occurrence property name

regex (Regex) : Regular expression (ECMAScript)

roots (ScenePathList) : If specified, restrict the search from the given roots [optional] (default:)

Return value:

paths (ScenePathList) : Scene paths which matches the given regular expression

findByProperty

findByProperty(property, regex[, **roots**]) -> paths

Returns all scene paths which a property value matches the given regular expression (ECMAScript)

Parameters:

property (String) : Property name

regex (Regex) : Regular expression (ECMAScript)

roots (ScenePathList) : If specified, restrict the search from the given roots [optional] (default:)

Return value:

paths (ScenePathList) : Scene paths which matches the given regular expression

getOccurrenceChildren

getOccurrenceChildren(occurrence[, **returnInstances**]) -> children

Returns the children of the given occurrence

Parameters:

occurrence (Occurrence) : The parent occurrence

returnInstances (Boolean) : If true, the instance node occurrences are returned, else their child are directly returned [optional] (default: false)

Return value:

children (OccurrenceList) : The children occurrences

getOccurrenceParent

getOccurrenceParent(occurrence[, **returnInstance**]) -> parent

Returns the parent of the given occurrence

Parameters:

occurrence (Occurrence) : The child occurrence

returnInstance (Boolean) : If the parent occurrence is an instance node: if returnInstances is true it will be returned, else the parent of the instance node will be returned (skip instance occurrence) [optional] (default: false)

Return value:

parent (Occurrence) : The parent occurrence