colladaRenderUtil Reference

PlayStation®Vita Programmer Tool Runtime Library Release 3.50

Table of Contents

Types

colladaRenderUtil::InstanceVisualScene

COLLADA scene instance

Definition

```
#include <colladarenderutil.h>
namespace colladaRenderUtil{
        class InstanceVisualScene
        {
              ..,
        };
)
```

Description

This is a COLLADA scene instance.

It retains scene-specific data such as node information and animation time. Multiple scene instances can be created from one `Collada`.

Keep the source `Collada` until all created scene instances are disposed of.

colladaRenderUtil::collada::VertexInput

Vertex information in vertex stream

Definition

```
#include <colladaRenderUtil.h>
namespace colladaRenderUtil {
        namespace collada {
              struct VertexInput
              {
                     VertexSemantic mSemantic;
                     int mInputSet;
                     int mOffsetInVertex;
              };
        }
}
```

Members

mSemantic	Semantic of the vertex data
mInputSet	Input set of the vertex data
mOffsetInVertex	Offset in the vertex data

Description

This indicates the data of one vertex attribute in the vertex data.

One of the following enters into *mSemantic*.

Macro	Value
SEMANTIC_BINORMAL	0
SEMANTIC_COLOR	1
SEMANTIC_CONTINUITY	2
SEMANTIC_IMAGE	3
SEMANTIC_INPUT	4
SEMANTIC_IN_TANGENT	5
SEMANTIC_INTERPOLATION	6
SEMANTIC_INV_BIND_MATRIX	7
SEMANTIC_JOINT	8
SEMANTIC_LINEAR_STEPS	9
SEMANTIC_MORPH_TARGETMORPH_WEIGHT	10
SEMANTIC_NORMAL	11
SEMANTIC_OUTPUT	12
SEMANTIC_OUT_TANGENT	13
SEMANTIC_POSITION	14
SEMANTIC_TANGENT	15
SEMANTIC_TEXBINORMAL	17
SEMANTIC_TEXCOORD	17
SEMANTIC_TEXTANGENT	18
SEMANTIC_UV	19
SEMANTIC_VERTEX	20
SEMANTIC_WEIGHT	21
SEMANTIC_INVALID	22

colladaRenderUtil::SimpleRenderer

colladaRenderUtil::SimpleRenderer

COLLADA simple renderer class

Definition

```
#include <colladarenderutil.h>
namespace colladaRenderUtil{
        class SimpleRenderer
        {
              ..,
        };
)
```

Internal Type

Type	Description
`SceneInfo`	Scene information during rendering

Members Functions

Function	Description
`initialize`	Initialize `SimpleRenderer`
`finalize`	Finalize `SimpleRenderer`
`loadCollada`	Initialize `Collada` class
`disposeCollada`	Dispose of `Collada`
`instantiateVisualScene`	Create a scene instance
`disposeVisualScene`	Dispose of a scene instance
`setTime`	Set the scene time
`render`	Render `InstanceVisualScene`

Description

This is a COLLADA simple renderer class. Use this with the following steps.

(1) Create this class instance and initialize it with the `initialize()` function.

(2) Use the `loadCollada()` function to load a COLLADA file.

(3) Use `instantiateVisualScene()` to create a scene instance.

(4) Use the `setTime()` function to set the time of the scene instance.

(5) Use the `render()` function to render the scene instance.

colladaRenderUtil::SimpleRenderer::SceneInfo

Scene information during rendering

Definition

```
#include <colladarenderutil.h>
namespace colladaRenderUtil{
        class SimpleRenderer
        {
           struct SceneInfo
           {
              colladaRenderUtil::InstanceVisualScene *instanceVisualScene;
              Vectormath::Aos::Matrix4 localToWorld;
           }
        };
)
```

Members Variables

Variable	Description
instanceVisualScene	Pointer to `colladaRenderUtil::InstanceVisualScene` to be rendered
localToWorld	Matrix for converting from `InstanceVisualScene` route coordinate system to world coordinate system

Description

This type is used with `colladaRenderUtil::SimpleRenderer::render()`.

Specify `InstanceVisualScene` to be rendered and the matrix for converting from the scene route coordinate system to world coordinate system.

colladaRenderUtil::SimpleRenderer::initialize

Initialize `SimpleRenderer`

Definition

```
#include <colladarenderutil.h>
int colladaRenderUtil::SimpleRenderer::initialize(
        uint32_t gpuHeapSize=32*1024*1024,
        ShaderParameterManager *uniformParamManager=NULL,
        bool verbose=false,
        SceGxmShaderPatcher* patcher=NULL
)
```

Arguments

gpuHeapSize	Size of memory area to be allocated within `SimpleRenderer` and be accessible from GPU
uniformParamManager	`ShaderParameterManager` used in a scene drawn with this `SimpleRenderer`
verbose	Flag that enables the redundant TTY output generated at the time of load processing
patcher	Shader patcher used when `SimpleRenderer` uses the shader. Shader patcher is generated automatically and internally in the case that NULL is specified.

Return Values

Returns `SCE_OK(0)` as the value of the function for success.

Returns a negative value for errors.

Description

Initializes `SimpleRenderer`.

colladaRenderUtil::SimpleRenderer::finalize

Finalize `SimpleRenderer`

Definition

```
#include <colladarenderutil.h>
int colladaRenderUtil::SimpleRenderer::finalize(void)
```

Arguments

None

Return Values

Returns `SCE_OK(0)` as the value of the function for success.

Returns a negative value for errors.

Description

Finalizes `SimpleRenderer`.
To perform proper finalization, this function must be executed after disposing of all instances of `Collada` and `InstanceVisualScene` created using `SimpleRenderer`.

colladaRenderUtil::SimpleRenderer::loadCollada

Initialize `Collada` class

Definition

```
#include <colladarenderutil.h>
int colladaRenderUtil::SimpleRenderer::loadCollada(
        colladaRenderUtil::Collada &collada,
        const char* daePath
)
```

Arguments

collada	Instance of `Collada` class for storing *collada* data
daePath	COLLADA file path

Return Values

Returns `SCE_OK(0)` as the value of the function for success.

Returns a negative value for errors.

Description

Loads the COLLADA file and stores the information in the `Collada` class.

colladaRenderUtil::SimpleRenderer::disposeCollada

Dispose of `Collada`

Definition

```
#include <colladarenderutil.h>
int colladaRenderUtil::SimpleRenderer::disposeCollada(
        colladaRenderUtil::Collada &collada
)
```

Arguments

collada	`Collada` class instance

Return Values

Returns `SCE_OK(0)` as the value of the function for success.

Returns a negative value for errors.

Description

Disposes of all data within the `Collada` class instance.

To dispose of the data properly, all instances of `InstanceVisualScene` created with this `Collada` must be disposed of.

colladaRenderUtil::SimpleRenderer::instantiateVisualScene

Create a scene instance

Definition

```
#include <colladarenderutil.h>
int colladaRenderUtil::SimpleRenderer::instantiateVisualScene(
        colladaRenderUtil::InstanceVisualScene &scene,
        const colladaRenderUtil::Collada &collada,
        ShaderParameterManager *instanceShaderParameterManager=NULL
)
```

Arguments

scene	Instance of `InstanceVisualScene` class where created scenes are stored
collada	*`collada`* used to create a scene
instanceShaderParameterManager	`ShaderParameterManager` used by this instance. This precedes the `ShaderParameterManager` registered in `SimpleRenderer` as for this instance

Return Values

Returns `SCE_OK(0)` as the value of the function for success.

Returns a negative value for errors.

Description

Creates an instance of a scene within *`collada`* and stores it in *`scene`*.

Keep the source `Collada` until this scene instance is disposed of.

colladaRenderUtil::SimpleRenderer::setTime

Set the scene time

Definition

```
#include <colladarenderutil.h>
int colladaRenderUtil::SimpleRenderer::setTime(
        colladaRenderUtil::InstanceVisualScene &scene,
        float time
)
```

Arguments

scene	Scene instance
time	Animation time

Return Values

Returns `SCE_OK(0)` as the value of the function for success.

Returns a negative value for errors.

Description

Specify the animation time of a scene. Specifying the time reproduces the scene at that time.

colladaRenderUtil::SimpleRenderer::disposeVisualScene

Dispose of a scene instance

Definition

```
#include <colladarenderutil.h>
int colladaRenderUtil::SimpleRenderer::disposeVisualScene(
        colladaRenderUtil::InstanceVisualScene &scene
)
```

Arguments

scene	Scene instance

Return Values

Returns `SCE_OK(0)` as the value of the function for success.

Returns a negative value for errors.

Description

Disposes of the scene instance.

colladaRenderUtil::SimpleRenderer::render

Render `InstanceVisualScene`

Definition

```
#include <colladarenderutil.h>
int colladaRenderUtil::SimpleRenderer::render(
        SceGxmContext *context,
        Vectormath::Aos::Matrix4_arg projectionMatrix,
        Vectormath::Aos::Matrix4_arg viewMatrix,
        Vectormath::Aos::Vector3_arg lightPosition,
        Vectormath::Aos::Vector3_arg lightColor,
        colladaRenderUtil::SimpleRenderer::SceneInfo *scenes,
        unsigned int numScenes
)
```

Arguments

context	Context of GXM
projectionMatrix	Projection matrix
viewMatrix	View conversion matrix
lightPosition	Point light position
lightColor	Point light color (X=red, Y=green, Z=blue) value range is 0.0 - 1.0
scenes	Array of scenes to be rendered
numScenes	Length of array of *scenes*

Return Values

Returns `SCE_OK(0)` as the value of the function for success.

Returns a negative value for errors.

Description

Renders `InstanceVisualScene` specified with *scenes*.

colladaRenderUtil::Collada

colladaRenderUtil::Collada::getLibraryGeometries

Get library geometries

Definition

```
#include <colladarenderutil.h>
colladaRenderUtil::collada::geometry::LibraryGeometries
colladaRenderUtil::Collada::getLibraryGeometries() const
```

Arguments

None

Return Values

Pointer to the library geometry structure

Description

Returns `LibraryGeometries` within `collada`.

colladaRenderUtil::Collada::getLibraryImages

Get the library images

Definition

```
#include <colladarenderutil.h>
colladaRenderUtil::collada::geometry::LibraryImages
colladaRenderUtil::Collada::getLibraryImages() const
```

Arguments

None

Return Values

Pointer to the library image structure

Description

Returns `LibraryImages` within `collada`.

colladaRenderUtil::collada::geometry::LibraryGeometries

colladaRenderUtil::collada::geometry::LibraryGeometries::getNumGeometries

Get the number of geometries

Definition

```
#include <colladarenderutil.h>
const uint32_t
colladaRenderUtil::collada::geometry::LibraryGeometries::getNumGeometries()
```

Arguments

None

Return Values

The number of geometries

Description

Returns the number of geometies within LibraryGeometries.

colladaRenderUtil::collada::geometry::LibraryGeometries::getGeometry

Get geometry with index specification

Definition

```
#include <colladarenderutil.h>
const Geometry*
colladaRenderUtil::collada::geometry::LibraryGeometries::getGeometry(
        int index
)
```

Arguments

index	Index of geometry

Return Values

Pointer to the Geometry structure

Description

Returns the geometry of the *index* number in LibraryGeometries.

colladaRenderUtil::collada::geometry:: Geometry

colladaRenderUtil::collada::geometry::Geometry::getMesh

Get mesh

Definition

```
#include <colladarenderutil.h>
Mesh*
colladaRenderUtil::collada::geometry::Geometry::getMesh()
```

Arguments

None

Return Values

Pointer to the Mesh structure

Description

Returns the mesh within the geometry.

colladaRenderUtil::collada::geometry::Mesh

colladaRenderUtil::collada::geometry::Mesh::getNumTriangless

Get the number of `Triangles`

Definition

```
#include <colladarenderutil.h>
uint32_t
colladaRenderUtil::collada::geometry::Mesh::getNumTriangless()
```

Arguments

None

Return Values

The number of `Triangles`

Description

Returns the number of triangular arrays in the mesh.

colladaRenderUtil::collada::geometry::Mesh::getTriangles

Get `Triangles`

Definition

```
#include <colladarenderutil.h>
const Triangles*
colladaRenderUtil::collada::geometry::Mesh::getTriangles(
        int index
)
```

Arguments

index	Index of `Triangles`

Return Values

Triangular array

Description

Returns the triangular array with an index specification.

colladaRenderUtil::collada::geometry::Triangles

colladaRenderUtil::collada::geometry::Triangles::getIndices

Get vertex index array

Definition

```
#include <colladarenderutil.h>
unsigned short*
colladaRenderUtil::collada::geometry::Triangles::getIndices()
```

Arguments

None

Return Values

Start pointer of the vertex index array

Description

Returns the vertex index array.

colladaRenderUtil::collada::geometry::Triangles::getNumIndices

Get the number of vertex indices

Definition

```
#include <colladarenderutil.h>
unsigned short
colladaRenderUtil::collada::geometry::Triangles::getNumIndices()
```

Arguments

None

Return Values

Number of vertex indices

Description

Returns the number of vertex indices.

colladaRenderUtil::collada::geometry::Triangles::getNumVertexInput

Get the number of attributes included in the vertex data

Definition

```
#include <colladarenderutil.h>
int
colladaRenderUtil::collada::geometry::Triangles::getNumVertexInput()
```

Arguments

None

Return Values

Number of attributes included in the vertex data

Description

Returns the number of attributes included in the vertex data.

colladaRenderUtil::collada::geometry::Triangles::getVertexInput

Get attribute information included in the vertex data

Definition

```
#include <colladarenderutil.h>
VertexInput
colladaRenderUtil::collada::geometry::Triangles::getVertexInput(
        int index
)
```

Arguments

index Index of `VertexInput`

Return Values

Attribute information included in the vertex data

Description

Returns the attribute information included in the vertex data.

colladaRenderUtil::collada::geometry::Triangles::getNumVertices

Get the number of vertices

Definition

```
#include <colladarenderutil.h>
unsigned short
colladaRenderUtil::collada::geometry::Triangles::getNumVertices()
```

Arguments

None

Return Values

Number of vertices

Description

Returns the number of vertices in the vertex data.

colladaRenderUtil::collada::geometry::Triangles::getStream

Get vertex data array

Definition

```
#include <colladarenderutil.h>
const float*
colladaRenderUtil::collada::geometry::Triangles::getStream()
```

Arguments

None

Return Values

Start pointer of the vertex data array

Description

Returns the vertex data array.

colladaRenderUtil::collada::geometry::Triangles::getStride

Get stride of the vertex data

Definition

```
#include <colladarenderutil.h>
unsigned int
colladaRenderUtil::collada::geometry::Triangles::getStride()
```

Arguments

None

Return Values

Stride of the vertex data

Description

Returns the stride of the vertex data.

colladaRenderUtil::collada::geometry::Triangles::getSemanticOffset

Get offset of semantic

Definition

```
#include <colladarenderutil.h>
int
colladaRenderUtil::collada::geometry::Triangles::getSemanticOffset(
        VertexSemantic semantic,
        int indexset=0
)
```

Arguments

semantic	Vertex semantic
indexset	Index set of the semantic

Return Values

Offset of semantic

Description

Returns the offset in the vertex stream of the specified semantic.

For *semantic*, specify one of the following.

Macro	Value
SEMANTIC_BINORMAL	0
SEMANTIC_COLOR	1
SEMANTIC_CONTINUITY	2
SEMANTIC_IMAGE	3
SEMANTIC_INPUT	4
SEMANTIC_IN_TANGENT	5
SEMANTIC_INTERPOLATION	6
SEMANTIC_INV_BIND_MATRIX	7
SEMANTIC_JOINT	8
SEMANTIC_LINEAR_STEPS	9
SEMANTIC_MORPH_TARGETMORPH_WEIGHT	10
SEMANTIC_NORMAL	11
SEMANTIC_OUTPUT	12
SEMANTIC_OUT_TANGENT	13
SEMANTIC_POSITION	14
SEMANTIC_TANGENT	15
SEMANTIC_TEXBINORMAL	17
SEMANTIC_TEXCOORD	17
SEMANTIC_TEXTANGENT	18
SEMANTIC_UV	19
SEMANTIC_VERTEX	20
SEMANTIC_WEIGHT	21
SEMANTIC_INVALID	22

colladaRenderUtil::collada::Image::LibraryImages

colladaRenderUtil::collada::Image::LibraryImages:: getNumImages

Get the number of images

Definition

```
#include <colladarenderutil.h>
uint32_t
colladaRenderUtil::collada::Image::LibraryImages::getNumImages()
```

Arguments

None

Return Values

The number of images

Description

Returns the number of images.

colladaRenderUtil::collada::Image::LibraryImages::getImage

Get `colladaRenderUtil::collada::Image`

Definition

```
#include <colladarenderutil.h>
const Image*
colladaRenderUtil::collada::Image::LibraryImages::getImage(
        int index
)
```

Arguments

Index Index

Return Values

Image

Description

Returns the image of the specified index.

colladaRenderUtil::ShaderParameterManager

colladaRenderUtil::ShaderParameterManager::getParameterId

Get the parameter ID

Definition

```
#include <colladarenderutil.h>
int
colladaRenderUtil::ShaderParameterManager::getParameterId(
        ShaderStage stage,
        const char *parameterName,
        uint32_t componentCount
)
```

Arguments

stage	Shader stage
parameterName	Parameter name
componentCount	Number of elements

Return Values

Parameter ID

Description

Set the ID to the parameter specified with the argument, and return that ID. The returned ID is used with `getSource()`.

If a negative value is returned, the default value is used for that parameter.

colladaRenderUtil::ShaderParameterManager::getSource

Get the value of the parameter

Definition

```
#include <colladarenderutil.h>
const float*
colladaRenderUtil::ShaderParameterManager::getSource(
        ShaderStage stage,
        int id
)
```

Arguments

stage	Shader stage
id	Parameter ID

Return Values

Value of the parameter

Description

Return the value to be set to the parameter corresponding to the parameter ID.

colladaRenderUtil::ShaderParameterManager::setNodeToSceneRootMatrix

Send notification of conversion matrix from node coordinate system to scene coordinate system

Definition

```
#include <colladarenderutil.h>
int
colladaRenderUtil::ShaderParameterManager::setNodeToSceneRootMatrix(
        sce::Vectormath::Simd::Aos::Matrix4_arg nodeToSceneRootMatrix
)
```

Arguments

nodeToSceneRootMatrix	Conversion matrix from node coordinate system to scene coordinate system

Return Values

If the value is set successfully, `SCE_OK` is returned. Otherwise, a negative value is returned.

Description

This is called when a renderer sends notification of a matrix for conversion from a node coordinate system to a scene coordinate system while rendering a node within a scene graph.

colladaRenderUtil::ShaderParameterManager::setSceneRootToWorldMatrix

Send notification of conversion matrix from scene coordinate system to world coordinate system

Definition

```
#include <colladarenderutil.h>
int
colladaRenderUtil::ShaderParameterManager::setSceneRootToWorldMatrix(
        sce::Vectormath::Simd::Aos::Matrix4_arg sceneRootToWorldMatrix
)
```

Arguments

sceneRootToWorldMatrix	Conversion matrix from scene coordinate system to world coordinate system

Return Values

If the value is set successfully, `SCE_OK` is returned. Otherwise, a negative value is returned.

Description

This is called when a renderer sends notification of a matrix for conversion from a scene coordinate system to a world coordinate system.

colladaRenderUtil::ShaderParameterManager::setViewProjectionMatrix

Send notification of view matrix and projection matrix

Definition

```
#include <colladarenderutil.h>
int
colladaRenderUtil::ShaderParameterManager::setViewProjectionMatrix(
        sce::Vectormath::Simd::Aos::Matrix4_arg viewMatrix,
        sce::Vectormath::Simd::Aos::Matrix4_arg projMatrix
)
```

Arguments

viewMatrix	View matrix
projMatrix	Projection matrix

Return Values

If the value is set successfully, `SCE_OK` is returned. Otherwise, a negative value is returned.

Description

This is called when a renderer sends notification of a view matrix and projection matrix.

colladaRenderUtil::ShaderParameterManager::getTextureParameterId

Get the texture parameter ID

Definition

```
#include <colladarenderutil.h>
int
colladaRenderUtil::ShaderParameterManager::getTextureParameterId(
        ShaderStage stage,
        const char *parameterName
)
```

Arguments

stage	Shader stage
parameterName	Parameter name

Return Values

Texture ID

Description

Set the ID to the texture parameter specified with the argument, and return that ID. The returned ID is used with `getSourceTexture()`.

If a negative value is returned, the default value is used for that parameter.

colladaRenderUtil::ShaderParameterManager::getSourceTexture

Get the texture

Definition

```
#include <colladarenderutil.h>
const SceGxmTexture*
colladaRenderUtil::ShaderParameterManager::getSourceTexture(
        ShaderStage stage,
        int id
)
```

Arguments

stage	Shader stage
id	Texture parameter ID

Return Values

Texture

Description

Return the value to be set to the texture corresponding to the texture parameter ID.