

Augmented Reality Scene Mapping on the PlayStation Vita

Generated by Doxygen 1.8.14

Contents

1	Deprecated List	1
2	Data Structure Index	3
2.1	Data Structures	3
3	File Index	5
3.1	File List	5
4	Data Structure Documentation	7
4.1	SmartTrackingState Struct Reference	7
5	File Documentation	11
5.1	D:/development/cmp404-2018/scene_mapping/ar_vita/sony_tracking.h File Reference	11
5.1.1	Function Documentation	12
5.1.1.1	sampleGetCameraImageTexture()	12
5.1.1.2	sampleGetLandmarkInfo()	12
5.1.1.3	sampleGetNumLandmarks()	13
5.1.1.4	sampleGetSceneMappingInitMode()	13
5.1.1.5	sampleGetSceneMappingState()	13
5.1.1.6	sampleGetTargetTransform()	13
5.1.1.7	sampleGetViewMatrix()	13
5.1.1.8	sampleIsCameraImageReady()	14
5.1.1.9	sampleRenderEnd()	14
5.1.1.10	sampleSceneMappingReset()	14
5.1.1.11	sampleSceneMappingSaveMap()	14
5.1.1.12	sampleTargetDetected()	14
	Index	17

Chapter 1

Deprecated List

Global `sampleRenderEnd ()`

This function no longer does anything. No need to call it.

Chapter 2

Data Structure Index

2.1 Data Structures

Here are the data structures with brief descriptions:

SmartTrackingState	7
--	---

Chapter 3

File Index

3.1 File List

Here is a list of all documented files with brief descriptions:

D:/development/cmp404-2018/scene_mapping/ar_vita/[sony_tracking.h](#) 11

Chapter 4

Data Structure Documentation

4.1 SmartTrackingState Struct Reference

Data Fields

- SInt **lost**
- SInt **detect**
- MyVector **pos**
- MyQuaternion **rot**
- struct {
 SceSmartSceneMappingResult **res**
 struct {
 Bool **enable**
 FilterState **state**
 } **filter**
 CameraBuffer * **image_queue** [DELAY_FRAME]
} **update**
- struct {
 UInt **lost**
 SInt **target**
 MyVector **pos**
 MyQuaternion **rot**
 SceUInt32 **timestamp**
 SceSmartTargetTrackingResult **res**
 struct {
 Bool **enable**
 SceInt32 **targetId**
 FilterState **state**
 } **filter**
} **tracking** [SMART_TRACKING_MAX]
- struct {
 UInt **num**
 SceSmartSceneMappingLandmarkInfo **info** [SMART_LANDMARK_MAX]
} **landmarks**

- ```

 struct {
 UInt num
 SceSmartSceneMappingInitializationPointInfo info [SMART_IP_MAX]
 } initial

```
- ```

      struct {
          SceInt32 id
          SceSmartTargetInfo info
      } target [SMART_TARGET_MAX]
      
```
- ```

 struct {
 volatile Bool exit
 MyThread thread
 MyEvent evt
 MySemaphore smp
 } coreThread

```
- ```

      struct {
          volatile Bool exit
          MyThread thread
          MyEvent evt
          SceSmartVector maskWork [3 *SMART_TRACKING_MAX *2]
          struct {
              SceSmartSceneMappingResult sm
          } res [SMART_RESULT_MAX]
          volatile UInt rpos
          volatile UInt wpos
          MyAtomic32 op
          volatile SceSmartSceneMappingInitMode mode
      } dispatchThread
      
```
- ```

 struct {
 volatile Bool exit
 MyThread thread [SMART_WORKER_THREAD_MAX]
 MyEventFlag flg
 MySemaphore smp
 } workerThread

```
- ```

      struct {
          volatile Bool exit
          volatile Bool run
          MyThread thread
          MyEvent evt
          MyMutex mtx
          SceSmartTargetTrackingResult res [SMART_TRACKING_MAX]
          SceUInt32 timestamp
          MyAtomic32 op
      } trackingThread
      
```
-

```
struct {  
    struct {  
        volatile float fps  
        volatile SceUInt32 pt  
        SceUInt32 t1  
    } mapping  
    struct {  
        SceUInt32 volatile pt  
    } worker [SMART_WORKER_THREAD_MAX]  
    struct {  
        volatile float fps  
        volatile SceUInt32 pt  
        SceUInt32 t1  
    } tracking  
} debug
```

- int **initStep**

The documentation for this struct was generated from the following file:

- D:/development/cmp404-2018/scene_mapping/ar_vita/smart_tracking.c

Chapter 5

File Documentation

5.1 D:/development/cmp404-2018/scene_mapping/ar_vita/sony_tracking.h File Reference

```
#include <gxm.h>
#include <libsmart.h>
#include <smart/scene_mapping.h>
```

Functions

- void [sampleInitialize](#) ()
Initialise the smart tracking library.
- void [sampleRelease](#) ()
Release resources used by the smart tracking library.
- void [sampleUpdate](#) ()
Release resources used by the smart tracking library. this once a frame with your application update.
- void [sampleRenderBegin](#) ()
Initialises smart tracking library for rendering the camera image. this once a frame in your application Render before drawing the camera feed.
- void [sampleRenderEnd](#) ()
Indicates the smart tracking library that rendering the camera image is finished.
- bool [sampleIsCameraImageReady](#) ()
Check if the camera image texture is ready to use.
- SceGxmTexture [sampleGetCameraImageTexture](#) ()
Get the texture containing the camera image.
- SceSmartSceneMappingState [sampleGetSceneMappingState](#) ()
Get the current state of the scene mapping process.
- void [sampleSceneMappingReset](#) ()
Reset the scene mapping process and start searching for landmarks.
- void [sampleSceneMappingChangeMode](#) ()
Change the scene mapping mode to the next mode.
- void [sampleSceneMappingSaveMap](#) ()
Saves the scene mapping data to file.
- int [sampleGetNumLandmarks](#) ()

- Get the number of landmarks found by the scene mapping process.*
- SceSmartSceneMappingLandmarkInfo [sampleGetLandmarkInfo](#) (int landmark_num)
Get information about a particular landmark.
- bool [sampleTargetDetected](#) (int idx)
Gets the tracking state of a particular marker.
- SceSmartSceneMappingInitMode [sampleGetSceneMappingInitMode](#) ()
Get the scene mapping initialisation mode from the smart tracking library.
- void [sampleGetViewMatrix](#) (gef::Matrix44 *mat)
Get the view matrix generated by the smart tracking library.
- void [sampleGetTargetTransform](#) (int idx, gef::Matrix44 *matrix)
Get the transformation matrix for a particular marker that is being tracked.

5.1.1 Function Documentation

5.1.1.1 [sampleGetCameraImageTexture\(\)](#)

```
SceGxmTexture sampleGetCameraImageTexture ( )
```

Get the texture containing the camera image.

Returns

the texture containing the camera image.

5.1.1.2 [sampleGetLandmarkInfo\(\)](#)

```
SceSmartSceneMappingLandmarkInfo sampleGetLandmarkInfo (
    int landmark_num )
```

Get information about a particular landmark.

Parameters

in	<i>landmark_num</i>	The landmark number.
----	---------------------	----------------------

Returns

the landmark data.

5.1.1.3 sampleGetNumLandmarks()

```
int sampleGetNumLandmarks ( )
```

Get the number of landmarks found by the scene mapping process.

Returns

the number of landmarks found by the scene mapping process.

5.1.1.4 sampleGetSceneMappingInitMode()

```
SceSmartSceneMappingInitMode sampleGetSceneMappingInitMode ( )
```

Get the scene mapping initialisation mode from the smart tracking library.

Returns

the scene mapping initialisation mode from the smart tracking library.

5.1.1.5 sampleGetSceneMappingState()

```
SceSmartSceneMappingState sampleGetSceneMappingState ( )
```

Get the current state of the scene mapping process.

Returns

the current state of the scene mapping process.

5.1.1.6 sampleGetTargetTransform()

```
void sampleGetTargetTransform (
    int idx,
    gef::Matrix44 * matrix )
```

Get the transformation matrix for a particular marker that is being tracked.

Note

This transformation matrix is in the world coordinate system

5.1.1.7 sampleGetViewMatrix()

```
void sampleGetViewMatrix (
    gef::Matrix44 * mat )
```

Get the view matrix generated by the smart tracking library.

Parameters

out	mat	The view matrix generated by the smart tracking library.
-----	-----	--

5.1.1.8 sampleIsCameraImageReady()

```
bool sampleIsCameraImageReady ( )
```

Check if the camera image texture is ready to use.

Returns

true if the camera image texture is ready to use, otherwise return false.

5.1.1.9 sampleRenderEnd()

```
void sampleRenderEnd ( )
```

Indicates the smart tracking library that rendering the camera image is finished.

Deprecated This function no longer does anything. No need to call it.

5.1.1.10 sampleSceneMappingReset()

```
void sampleSceneMappingReset ( )
```

Reset the scene mapping process and start searching for landmarks.

Note

Call this in the application update before [sampleUpdate\(\)](#).

5.1.1.11 sampleSceneMappingSaveMap()

```
void sampleSceneMappingSaveMap ( )
```

Saves the scene mapping data to file.

Note

NOT TESTED. USE AT OWN RISK.

5.1.1.12 sampleTargetDetected()

```
bool sampleTargetDetected (
    int idx )
```

Gets the tracking state of a particular marker.

Parameters

in	<i>idx</i>	The marker number [0..5]
----	------------	--------------------------

Returns

true if the marker is current being tracked, otherwise return false.

Index

D:/development/cmp404-2018/scene_mapping/ar_↵
vita/sony_tracking.h, 11

sampleGetCameraImageTexture

sony_tracking.h, 12

sampleGetLandmarkInfo

sony_tracking.h, 12

sampleGetNumLandmarks

sony_tracking.h, 12

sampleGetSceneMappingInitMode

sony_tracking.h, 13

sampleGetSceneMappingState

sony_tracking.h, 13

sampleGetTargetTransform

sony_tracking.h, 13

sampleGetViewMatrix

sony_tracking.h, 13

sampleIsCameraImageReady

sony_tracking.h, 14

sampleRenderEnd

sony_tracking.h, 14

sampleSceneMappingReset

sony_tracking.h, 14

sampleSceneMappingSaveMap

sony_tracking.h, 14

sampleTargetDetected

sony_tracking.h, 14

SmartTrackingState, 7

sony_tracking.h

sampleGetCameraImageTexture, 12

sampleGetLandmarkInfo, 12

sampleGetNumLandmarks, 12

sampleGetSceneMappingInitMode, 13

sampleGetSceneMappingState, 13

sampleGetTargetTransform, 13

sampleGetViewMatrix, 13

sampleIsCameraImageReady, 14

sampleRenderEnd, 14

sampleSceneMappingReset, 14

sampleSceneMappingSaveMap, 14

sampleTargetDetected, 14