

# Table of Contents

- Uralstech.UXR.QuestCamera ..... 2
  - CameraDevice ..... 3
  - CameraDevice.CaptureSessionObject ..... 8
  - CameraDevice.CaptureTemplate ..... 10
  - CameraDevice.ErrorCode ..... 11
  - CameraFrameForwarder ..... 12
  - CaptureSession ..... 14
  - NativeWrapperState ..... 18
  - UCameraManager ..... 19
  - YUVToRGBAConverter ..... 22

# Namespace Uralstech.UXR.QuestCamera

## Classes

### [CameraDevice](#)

A wrapper for a native Camera2 CameraDevice.

### [CameraDevice.CaptureSessionObject](#)

Simple class for grouping capture session related GameObjects.

### [CameraFrameForwarder](#)

Forwards frame callbacks from the native Kotlin plugin to Unity.

### [CaptureSession](#)

A wrapper for a native Camera2 CaptureSession and ImageReader.

### [UCameraManager](#)

Class for interfacing with the native Camera2 API on Android.

### [YUVToRGBAConverter](#)

The default YUV 4:2:0 to RGBA converter that uses a compute shader to convert the camera texture to RGBA.

## Enums

### [CameraDevice.CaptureTemplate](#)

Capture template to use when recording.

### [CameraDevice.ErrorCode](#)

Error codes that can be returned by the native CameraDevice wrapper.

### [NativeWrapperState](#)

The current assumed state of a native wrapper.

# Class CameraDevice

Namespace: [Uralstech.UXR.QuestCamera](#)

A wrapper for a native Camera2 CameraDevice.

```
public class CameraDevice : MonoBehaviour
```

## Inheritance

object ← CameraDevice

## Fields

### OnDeviceClosed

Invoked when the CameraDevice is closed.

```
public UnityEvent OnDeviceClosed
```

Field Value

UnityEvent

### OnDeviceDisconnected

Invoked when the CameraDevice is disconnected.

```
public UnityEvent OnDeviceDisconnected
```

Field Value

UnityEvent

### OnDeviceErred

Invoked when the CameraDevice encounters an error.

```
public UnityEvent<CameraDevice.ErrorCode> OnDeviceErred
```

Field Value

UnityEvent<[ErrorCode](#)>

## OnDeviceOpened

Invoked when the CameraDevice is opened.

```
public UnityEvent OnDeviceOpened
```

Field Value

UnityEvent

## Properties

### CameraId

The ID of the camera being wrapped.

```
public string CameraId { get; }
```

Property Value

string

### CurrentState

The current assumed state of the native CameraDevice wrapper.

```
public NativeWrapperState CurrentState { get; }
```

Property Value

[NativeWrapperState](#)

## IsActiveAndUsable

Is the native CameraDevice wrapper active and usable?

```
public bool IsActiveAndUsable { get; }
```

Property Value

bool

## Methods

### CreateCaptureSession(Resolution, CaptureTemplate, bool)

Creates a new capture session for use.

```
public CameraDevice.CaptureSessionObject CreateCaptureSession(Resolution resolution,  
CameraDevice.CaptureTemplate captureTemplate = CaptureTemplate.Preview, bool isContinuous  
= true)
```

Parameters

**resolution** Resolution

The resolution of the capture.

**captureTemplate** [CameraDevice.CaptureTemplate](#)

The capture template to use for the capture

**isContinuous** bool

Is this capture continuous (repeating) or for a single frame?

Returns

## [CameraDevice.CaptureSessionObject](#)

A new capture session wrapper. May be null if the current camera device is not usable.

### Remarks

Once you have finished using the capture session, either destroy its GameObject or call [Release\(\)](#) and [Release\(\)](#) to close the session and free up native and compute shader resources.

## OnDestroy()

```
protected void OnDestroy()
```

## Release()

Releases the CameraDevice's native resources, and makes it unusable.

```
public void Release()
```

## WaitForInitialization()

Waits until the CameraDevice is open or erred out.

```
public IEnumerator WaitForInitialization()
```

### Returns

IEnumerator

## \_onDeviceClosed(string)

```
public void _onDeviceClosed(string _)
```

### Parameters

`_` string

## `_onDeviceDisconnected(string)`

```
public void _onDeviceDisconnected(string _)
```

### Parameters

`_` string

## `_onDeviceErred(string)`

```
public void _onDeviceErred(string errorCodeStr)
```

### Parameters

`errorCodeStr` string

## `_onDeviceOpened(string)`

```
public void _onDeviceOpened(string _)
```

### Parameters

`_` string

# Class CameraDevice.CaptureSessionObject

Namespace: [Uralstech.UXR.QuestCamera](#)

Simple class for grouping capture session related GameObjects.

```
public class CameraDevice.CaptureSessionObject
```

## Inheritance

object ← CameraDevice.CaptureSessionObject

## Fields

### CaptureSession

The capture session wrapper.

```
public readonly CaptureSession CaptureSession
```

Field Value

[CaptureSession](#)

### GameObject

The GameObject containing the [CaptureSession](#) and [TextureConverter](#) components.

```
public readonly GameObject GameObject
```

Field Value

GameObject

### TextureConverter



The YUV to RGBA texture converter.

```
public readonly YUVToRGBAConverter TextureConverter
```

Field Value

[YUVToRGBAConverter](#)

## Methods

### Destroy()

Destroys the GameObject to release all native resources.

```
public void Destroy()
```

# Enum CameraDevice.CaptureTemplate

Namespace: [Uralstech.UXR.QuestCamera](#)

Capture template to use when recording.

```
public enum CameraDevice.CaptureTemplate
```

## Fields

**Default = 0**

Default value, do not use.

**Preview = 1**

Creates a request suitable for a camera preview window.

**Record = 3**

Creates a request suitable for video recording.

**StillCapture = 2**

Creates a request suitable for still image capture.

**VideoSnapshot = 4**

Creates a request suitable for still image capture while recording video.

**ZeroShutterLag = 5**

Creates a request suitable for zero shutter lag still capture.

# Enum CameraDevice.ErrorCode

Namespace: [Uralstech.UXR.QuestCamera](#)

Error codes that can be returned by the native CameraDevice wrapper.

```
public enum CameraDevice.ErrorCode
```

## Fields

**CameraAccessException = 1000**

The native code encountered a CameraAccessException.

**CameraDeviceError = 4**

The camera device has encountered a fatal error.

**CameraDisabled = 3**

The camera device could not be opened due to a device policy.

**CameraInUse = 1**

The camera device is in use already.

**CameraServiceError = 5**

The camera service has encountered a fatal error.

**MaxCamerasInUse = 2**

The camera device could not be opened because there are too many other open camera devices.

**SecurityException = 1001**

The native code encountered a SecurityException.

**Unknown = 0**

Unknown error.

# Class CameraFrameForwarder

Namespace: [Uralstech.UXR.QuestCamera](#)

Forwards frame callbacks from the native Kotlin plugin to Unity.

```
public class CameraFrameForwarder : AndroidJavaProxy
```

## Inheritance

object ← CameraFrameForwarder

## Constructors

### CameraFrameForwarder()

```
public CameraFrameForwarder()
```

## Fields

### OnFrameReady

Callback for processing the YUV 4:2:0 frame.

```
public Action<nint, nint, nint, int, int, int, int, int, int> OnFrameReady
```

### Field Value

Action<nint, nint, nint, int, int, int, int, int, int>

### Remarks

Parameters	
yBuffer (IntPtr)	Pointer to the buffer containing Y (luminance) data of the frame.
uBuffer (IntPtr)	Pointer to the buffer containing U (color) data of the frame.

Parameters	
<b>vBuffer (IntPtr)</b>	Pointer to the buffer containing V (color) data of the frame.
<b>ySize (int)</b>	The size of yBuffer.
<b>uSize (int)</b>	The size of uBuffer.
<b>vSize (int)</b>	The size of vBuffer.
<b>yRowStride (int)</b>	The size of each row of the image in yBuffer in bytes.
<b>uvRowStride (int)</b>	The size of each row of the image in uBuffer and vBuffer in bytes.
<b>uvPixelStride (int)</b>	The size of a pixel in a row of the image in uBuffer and vBuffer in bytes.

## Methods

### GetBufferPointer(AndroidJavaObject)

Gets the pointer to a native buffer from a Java ByteBuffer object.

```
protected static nint GetBufferPointer(AndroidJavaObject byteBuffer)
```

#### Parameters

**byteBuffer** AndroidJavaObject

The Java ByteBuffer object.

#### Returns

nint

A pointer to the native buffer.

# Class CaptureSession

Namespace: [Uralstech.UXR.QuestCamera](#)

A wrapper for a native Camera2 CaptureSession and ImageReader.

```
public class CaptureSession : MonoBehaviour
```

## Inheritance

object ← CaptureSession

## Fields

### OnSessionConfigurationFailed

Called when the session could not be configured.

```
public UnityEvent<string> OnSessionConfigurationFailed
```

Field Value

UnityEvent<string>

### OnSessionConfigured

Called when the session has been configured.

```
public UnityEvent OnSessionConfigured
```

Field Value

UnityEvent

### OnSessionRequestFailed

Called when the session request could not be set.

```
public UnityEvent<string> OnSessionRequestFailed
```

Field Value

UnityEvent<string>

## OnSessionRequestSet

Called when the session request has been set.

```
public UnityEvent OnSessionRequestSet
```

Field Value

UnityEvent

## Properties

### CurrentState

The current assumed state of the native CaptureSession wrapper.

```
public NativeWrapperState CurrentState { get; }
```

Property Value

[NativeWrapperState](#)

### IsActiveAndUsable

Is the native CaptureSession wrapper active and usable?

```
public bool IsActiveAndUsable { get; }
```

Property Value

bool

## Methods

### OnDestroy()

```
protected void OnDestroy()
```

### Release()

Releases the CaptureSession's native resources, and makes it unusable.

```
public void Release()
```

### WaitForInitialization()

Waits until the CaptureSession is open or erred out.

```
public IEnumerator WaitForInitialization()
```

Returns

IEnumerator

### \_onSessionConfigurationFailed(string)

```
public void _onSessionConfigurationFailed(string reason)
```

Parameters

**reason** string



## \_onSessionConfigured(string)

```
public void _onSessionConfigured(string _)
```

### Parameters

\_ string

## \_onSessionRequestFailed(string)

```
public void _onSessionRequestFailed(string reason)
```

### Parameters

reason string

## \_onSessionRequestSet(string)

```
public void _onSessionRequestSet(string _)
```

### Parameters

\_ string

# Enum NativeWrapperState

Namespace: [Uralstech.UXR.QuestCamera](#)

The current assumed state of a native wrapper.

```
public enum NativeWrapperState
```

## Fields

**Closed = 2**

The native wrapper failed with an error, was disconnected or was closed normally.

**Initializing = 0**

The native wrapper is still initializing.

**Opened = 1**

The native wrapper is open and ready.

# Class UCameraManager

Namespace: [Uralstech.UXR.QuestCamera](#)

Class for interfacing with the native Camera2 API on Android.

```
public class UCameraManager : DontCreateNewSingleton<UCameraManager>
```

## Inheritance

object ← UCameraManager

## Fields

### YUVToRGBAComputeShader

The compute shader to use to convert the camera's YUV 4:2:0 images to RGBA.

```
public ComputeShader YUVToRGBAComputeShader
```

Field Value

ComputeShader

## Properties

### CameraDevices

Gets the available camera devices. May be null.

```
public string[] CameraDevices { get; }
```

Property Value

string[]

# Methods

## Awake()

```
protected override void Awake()
```

## GetSupportedResolutions(string)

Gets the supported resolutions for the specified camera.

```
public Resolution[] GetSupportedResolutions(string camera)
```

### Parameters

**camera** string

The ID of the camera. You can get it from [CameraDevices](#).

### Returns

Resolution[]

## OnDestroy()

```
protected void OnDestroy()
```

## OpenCamera(string)

Opens a camera device for use.

```
public CameraDevice OpenCamera(string camera)
```

### Parameters

**camera** string

The ID of the camera. You can get it from [CameraDevices](#).

## Returns

[CameraDevice](#)

A new camera device wrapper. May be null if the current object is disposed/unusable.

## Remarks

Once you have finished using the camera, either destroy its GameObject or call [Release\(\)](#) to close the camera and free up native resources.

# Class YUVToRGBAConverter

Namespace: [Uralstech.UXR.QuestCamera](#)

The default YUV 4:2:0 to RGBA converter that uses a compute shader to convert the camera texture to RGBA.

```
public class YUVToRGBAConverter : MonoBehaviour
```

## Inheritance

object ← YUVToRGBAConverter

## Fields

### OnFrameProcessed

Called when a frame has been converted from YUV 4:2:0 to RGBA.

```
public UnityEvent<RenderTexture> OnFrameProcessed
```

## Field Value

UnityEvent<RenderTexture>

## Shader

The shader used to convert YUV 4:2:0 to an RGBA RenderTexture. Uses [YUVToRGBAComputeShader](#) if not specified here.

```
public ComputeShader Shader
```

## Field Value

ComputeShader

## `_uComputeBuffer`

Pointer to the buffer containing U (color) data of the frame being processed.

```
protected ComputeBuffer _uComputeBuffer
```

Field Value

ComputeBuffer

## `_vComputeBuffer`

Pointer to the buffer containing V (color) data of the frame being processed.

```
protected ComputeBuffer _vComputeBuffer
```

Field Value

ComputeBuffer

## `_yComputeBuffer`

Pointer to the buffer containing Y (luminance) data of the frame being processed.

```
protected ComputeBuffer _yComputeBuffer
```

Field Value

ComputeBuffer

## Properties

### CameraFrameForwarder

The native camera frame forwarder.

```
public CameraFrameForwarder CameraFrameForwarder { get; }
```

Property Value

[CameraFrameForwarder](#)

## FrameRenderTexture

The RenderTexture which will contain the RGBA camera frames.

```
public RenderTexture FrameRenderTexture { get; }
```

Property Value

RenderTexture

## Methods

### Awake()

```
protected void Awake()
```

### CopyNativeDataToComputeBuffer(ref ComputeBuffer, nint, int)

Copies native (unmanaged) byte data to a compute buffer.

```
protected static void CopyNativeDataToComputeBuffer(ref ComputeBuffer computeBuffer, nint  
nativeBufferPtr, int nativeBufferSize)
```

Parameters

**computeBuffer** ComputeBuffer

The buffer to copy to.



**nativeBufferPtr** nint

The memory to copy from.

**nativeBufferSize** int

The number of bytes to copy.

## OnDestroy()

```
protected void OnDestroy()
```

## OnFrameReady(nint, nint, nint, int, int, int, int, int, int)

Callback for [CameraFrameForwarder](#).

```
protected virtual void OnFrameReady(nint yBuffer, nint uBuffer, nint vBuffer, int ySize, int uSize, int vSize, int yRowStride, int uvRowStride, int uvPixelStride)
```

### Parameters

**yBuffer** nint

Pointer to the buffer containing Y (luminance) data of the frame.

**uBuffer** nint

Pointer to the buffer containing U (color) data of the frame.

**vBuffer** nint

Pointer to the buffer containing V (color) data of the frame.

**ySize** int

The size of **yBuffer**.

**uSize** int

The size of **uBuffer**.

**vSize** int

The size of **vBuffer**.

**yRowStride** int

The size of each row of the image in **yBuffer** in bytes.

**uvRowStride** int

The size of each row of the image in **uBuffer** and **vBuffer** in bytes.

**uvPixelStride** int

The size of a pixel in a row of the image in **uBuffer** and **vBuffer** in bytes.

## Release()

Releases the ComputeBuffers and RenderTextures associated with this converter.

```
public void Release()
```

## SendFrameToComputeBuffer(int, int, int)

Sends the camera frame stored in the compute buffers to the compute shader and dispatches it.

```
protected virtual void SendFrameToComputeBuffer(int yRowStride, int uvRowStride,  
int uvPixelStride)
```

### Parameters

**yRowStride** int

The size of each row of the image in [\\_yComputeBuffer](#) in bytes.

**uvRowStride** int

The size of each row of the image in [\\_uComputeBuffer](#) and [\\_vComputeBuffer](#) in bytes.

**uvPixelStride** int

The size of a pixel in a row of the image in [\\_uComputeBuffer](#) and [\\_vComputeBuffer](#) in bytes.