Class

LayerRenderer

A type that provides the Metal types and timing information you need to draw your content.

macOS 26.0+ | visionOS 1.0+

```
class LayerRenderer
```

Mentioned in

Drawing fully immersive content using Metal

Overview

A layer renderer type creates a bridge between a SwiftUI scene and the Metal code you use to draw fully immersive experiences. When you present an immersive space with `CompositorLayer` content, the system creates a `LayerRenderer` type and makes it available to the content's closure. Use the information in the layer renderer to set up your app's rendering loop, and to start drawing frames of content.

Each layer renderer has information that tells the system how to configure the Metal textures and data types your app needs. Compositor Services provides a default configuration for layers, but you can customize the configuration as needed. Specify your custom configuration details using the `CompositorLayerConfiguration` protocol and pass a type with those details to the initializer for your immersive space's content. Use the layer renderer's capability information to validate any configuration choices you make.

For information about how to create and configure a layer renderer and use it to run your rendering loop, see Drawing fully immersive content using Metal.

Topics

Configuring the layer renderer

`var configuration: LayerRenderer.Configuration`

The configuration details for the specified layer.

`struct Configuration`

A type that stores the texture formats, layout information, and other details you use to configure your rendering loop code.

`struct Capabilities`

The color formats, depth formats, and features that you can use to configure your rendering engine.

Getting the layer renderer properties

`var properties: LayerRenderer.Properties`

The configured properties of the layer renderer.

`struct Properties`

A type that describes the organization of the layer renderer's textures and the relationships between those textures and the views you use for drawing.

Getting the GPU device

`var device: any MTLDevice`

The GPU device that the layer renderer uses for drawing operations

Managing the rendering loop

`var state: LayerRenderer.State`

A value that indicates whether the layer renderer is currently visible and ready for you to draw content.

`func waitUntilRunning()`

Stops further execution of your code until the layer renderer leaves the paused state.

`enum State`

The states of the layer renderer, which tell you how to proceed with drawing operations.

`struct Clock`

A type that supports operations that require a precise time measurement.

Drawing a frame of content

`func queryNextFrame() -> LayerRenderer.Frame?`

Returns the next frame to use for drawing.

`struct Frame`

A type that provides access to the timing information and data types you need to render a single frame of content.

`struct Drawable`

A type that provides the textures and information you need to draw a frame of content.

Configuring the frame update rate

`var minimumFrameRepeatCount: Int32`

The number of additional frames for which the system displays the same content.

Defining quality level

`var renderQuality: LayerRenderer.RenderQuality`

Get the render quality to be used by the drawables.

`var defaultRenderQuality: LayerRenderer.RenderQuality`

The default render quality used on this platform.

`var maxRenderQuality: LayerRenderer.RenderQuality`

The max render quality the layer can use when drawing to the drawables.

Defining layer renderer quality

Declare the render quality of your textures to enable high-quality rendering.

Structures

`struct RenderQuality`

Render quality controls the quality which drawing happens at.

Instance Properties

```
var commandQueue: any MTL4CommandQueue
```

Returns the command queue that the layer uses for drawing operations.

```
var onSpatialEvent: (SpatialEventCollection) -> Void
```

A closure that receives the spatial events updates from the LayerRenderer

```
var onSpatialEvent: (SpatialEventCollection) -> Void
```

A closure that receives the spatial events updates from the LayerRenderer

---

Relationships

Inherits From

NSObject

Conforms To

CVarArg
CustomDebugStringConvertible
CustomStringConvertible
Equatable
Hashable
NSObjectProtocol

See Also

Render-loop setup

```
struct Frame
```

A type that provides access to the timing information and data types you need to render a single frame of content.