Protocol

MTLParallelRenderCommandEncoder

An instance that splits up a single render pass so that it can be simultaneously encoded from multiple threads.

iOS 8.0+ | iPadOS 8.0+ | Mac Catalyst 13.1+ | macOS 10.11+ | tvOS | visionOS 1.0+

```
protocol MTLParallelRenderCommandEncoder : MTLCommandEncoder
```

Mentioned in

Understanding the Metal 4 core API

Overview

Your app does not define classes that implement this protocol. To create an `MTLParallelRenderCommandEncoder` instance, call the `makeParallelRenderCommandEncoder(descriptor:)` method of the `MTLCommandBuffer` instance that you want to encode the rendering commands into. Then, call the renderCommandEncoder method on this `MTLParallelRenderCommandEncoder` instance to create one or more `MTLRenderCommandEncoder` instances. The subordinate `MTLRenderCommandEncoder` instances created encode their commands to the same command buffer and target the same `MTLRenderPassAttachmentDescriptor` instance. The `MTLParallelRenderCommandEncoder` instance ensures the attachment load and store actions only occur at the start and end of the entire rendering pass.

You can assign each `MTLRenderCommandEncoder` to its own thread and each can encode commands in parallel. You are responsible for any thread synchronization that is required. After all the subordinate encoders have finished encoding their commands, call `endEncoding()` to execute the commands. The rendering commands are executed in the order that the subordinate encoders were created.

Topics

Creating a render command encoder

`func makeRenderCommandEncoder() -> (any MTLRenderCommandEncoder)?`

Create an object that encodes commands that perform graphics rendering operations and may be assigned to a different thread.

Required

Setting render pass state

`func setColorStoreAction(MTLStoreAction, index: Int)`

Specifies a known store action to replace the initial `MTLStoreAction.unknown` value specified for a given color attachment.

Required

`func setColorStoreActionOptions(MTLStoreActionOptions, index: Int)`

Specifies known store action options for a given color attachment.

Required

`func setDepthStoreAction(MTLStoreAction)`

Specifies a known store action to replace the initial `MTLStoreAction.unknown` value specified for a given depth attachment.

Required

`func setDepthStoreActionOptions(MTLStoreActionOptions)`

Specifies known store action options for a given depth attachment.

Required

`func setStencilStoreAction(MTLStoreAction)`

Specifies a known store action to replace the initial `MTLStoreAction.unknown` value specified for a given stencil attachment.

Required

`func setStencilStoreActionOptions(MTLStoreActionOptions)`

Specifies known store action options for a given stencil attachment.

Required

Relationships

Inherits From

`MTLCommandEncoder`, `NSObjectProtocol`

See Also

Encoding a render pass in parallel

`enum MTLLoadAction`

Types of actions performed for an attachment at the start of a rendering pass.

`enum MTLStoreAction`

Types of actions performed for an attachment at the end of a rendering pass.

`struct MTLStoreActionOptions`

Options that modify a store action.