Sample Code

Customizing shaders using function pointers and stitching

Define custom shader behavior at runtime by creating functions from existing ones and preferentially linking to others in a dynamic library.

Download

iOS 15.0+ | iPadOS 15.0+ | macOS 12.0+ | Xcode 13.4+

Overview

Note

This sample code project is associated with WWDC2021 session 10229: Discover compilation workflows in Metal and WWDC2022 session 6596: Target and optimize GPU binaries with Metal 3.

See Also

Stitched function libraries

`class` `MTLStitchedLibraryDescriptor`

A description of a new library of procedurally generated functions.

`class` `MTLFunctionStitchingGraph`

A description of a new stitched function.

`class` `MTLFunctionStitchingInputNode`

A call graph node that describes an input to the call graph.

`class` `MTLFunctionStitchingFunctionNode`

A call graph node that describes a function call and its inputs.

`protocol` `MTLFunctionStitchingNode`

A protocol to identify call graph nodes.

`class` `MTLFunctionStitchingAttributeAlwaysInline`

An attribute to specify that Metal needs to inline all of the function calls when generating the stitched function.

`protocol` `MTLFunctionStitchingAttribute`

A protocol to identify types that customize how the Metal compiler stitches a function together.