Sample Code

Capturing Metal commands programmatically

Invoke a Metal frame capture from your app, then save the resulting GPU trace to a file or view it in Xcode.

Download

macOS 10.15+ | Xcode 11.0+

Overview

> Note
>
> This sample code project is associated with WWDC 2019 session 606: Delivering Optimized Metal Apps and Games.

Configure the sample code project

To run the app:

- Build the project with Xcode 11 or later.

See Also

Developer tools

Supporting Simulator in a Metal app

Configure alternative render paths in your Metal app to enable running your app in Simulator.

Logging shader debug messages

Print debugging messages that a shader generates using shader logging.

Developing Metal apps that run in Simulator

Prototype and test your Metal apps in Simulator.

Improving your game’s graphics performance and settings

Fix performance glitches and develop default settings for smooth experiences on Apple platforms using the powerful suite of Metal development tools.

Metal debugger

Debug and profile your Metal workload with a GPU trace.

Metal developer workflows

Locate and fix issues related to your app’s use of the Metal API and GPU functions.

GPU counters and counter sample buffers

Retrieve runtime data from a GPU device by sampling one or more of its counters.

Metal debugging types

Create capture managers and capture scopes, and review a GPU device’s log after it runs a command buffer.