API Collection

Blit passes

Encode a block information transfer pass to adjust and copy data to and from GPU resources, such as buffers and textures.

Overview

Your app can use a block information transfer (blit) pass to manage data within, and copy data between, buffers, textures, and other Metal resources. Start by creating a blit command encoder by calling an `MTLCommandBuffer` instance's `makeBlitCommandEncoder()` method. Then use the `MTLBlitCommandEncoder` instance's methods to encode individual commands of your blit pass.

You also have the option to customize a blit pass's runtime behavior, such as sampling GPU hardware data. To enable these options, configure an `MTLBlitPassDescriptor` instance and pass it to the command buffer's `makeBlitCommandEncoder(descriptor:)` method. For more information about sampling GPU hardware data in a blit pass, see the articles in GPU counters and counter sample buffers, including:

- Sampling GPU data into counter sample buffers
- Converting a GPU's counter data into a readable format

Topics

Encoding a blit pass

`protocol` `MTLBlitCommandEncoder`

An interface you can use to encode GPU commands that copy and modify the underlying memory of various Metal resources.

`struct MTLBlitOption`

The options that enable behavior for some blit operations.

Configuring a blit command encoder

`class MTLBlitPassDescriptor`

A configuration you create to customize a blit command encoder, which affects the runtime behavior of the blit pass you encode with it.

`class MTLBlitPassSampleBufferAttachmentDescriptor`

A configuration that instructs the GPU where to store counter data from the beginning and end of a blit pass.

`class MTLBlitPassSampleBufferAttachmentDescriptorArray`

A container that stores an array of sample buffer attachments for a blit pass.

See Also

Command encoders

- Render passes

  Encode a render pass to draw graphics into an image.

- Compute passes

  Encode a compute pass that runs computations in parallel on a thread grid, processing and manipulating Metal resource data on multiple cores of a GPU.

- Machine-learning passes

  Add machine-learning model inference to your Metal app's GPU workflow.

- Indirect command encoding

  Store draw commands in Metal buffers and run them at a later time on the GPU, either once or repeatedly.

- Ray tracing with acceleration structures

Build a representation of your scene's geometry using triangles and bounding volumes to quickly trace rays through the scene.