Choosing a SpriteKit Scene Renderer

Compare the different ways to display a SpriteKit scene.

Overview

To display SpriteKit content in a watch app, you use WKInterfaceSKScene, whereas on iOS, macOS, and tvOS, you use SKView to display a SpriteKit scene. If you need to mix SpriteKit with any custom Metal content, use SKRenderer.

> **Important:** SKRenderer is not available on watchOS.

On iOS, tvOS, and macOS, Draw Using a View or a Renderer

Choose from the options below to display SpriteKit content on iOS, tvOS, or macOS.

Draw Using a View

To draw SpriteKit content the most common way, set up an SKView and set its scene property. See Drawing SpriteKit Content in a View for an example of using SKView.

Draw Using a Renderer

When you want to mix SpriteKit content with your own Metal content, use SKRenderer to draw your SKScene. You might use SKRenderer to replace specific portions of a SpriteKit app with custom Metal code, or vice versa. SKRenderer combines the full control of Metal with the ease of use of SpriteKit and its library of useful objects. For example, to take full control of just your game's tiled background while using SpriteKit to draw the rest, use an SKRenderer and MTLView combination instead of SKView, and then write the tiled background code yourself in Metal.When you're ready to mix the SpriteKit content with the Metal content, you call SKRenderer's render(withViewport:commandBuffer:renderPassDescriptor:). Finally, call present(_:) on a Metal view when you're ready to present a frame.

On watchOS, Draw Using an Interface Scene

To display SpriteKit content on Apple Watch, you create a WKInterfaceSKScene and then set its sceneproperty. WKInterfaceSKScene functions much like SKView, but without the debugging visualization options.