

Loading and Using Textures

Learn the basics about using textures in SpriteKit.

Overview

Use the `init(imageNamed:)` method to create an `SKTexture`. When the texture is initialized using an image file, SpriteKit must perform two steps before it can use the texture to render sprites. First, it must load the texture data from the file. Second, it must prepare the texture data to be used by the graphics hardware. Both of these steps can be expensive compared to other SpriteKit operations. In particular, loading a texture from a file is very expensive. SpriteKit defers loading and preparing the texture as long as possible and provides methods you can use to control these steps.

The texture data is loaded when:

- >> The `size()` method on the texture object is called.
- >> Another method is called that requires the texture's size, such as creating a new `SKSpriteNode` object that uses the texture object.
- >> One of the preload methods is called

The texture data is prepared for rendering when a sprite or particle that uses the texture is part of a node tree that is being rendered. Once the `SKTexture` object is ready for rendering, it stays ready until all strong references to the texture object are removed.

Texture objects can be created from data generated at runtime. For example, you can create the texture from a Quartz 2D image, from raw pixel data, or by applying a Core Image filter to an existing texture. You can also call the `texture(from:)` method to render a node tree into a texture. In this case, preloading isn't necessary because the texture's image data is already in memory when the texture is created. However, the texture must still be prepared by SpriteKit before it can be used for rendering.

To create textures that can be updated at runtime, see the `SKMutableTexture` class.