Maximizing Texture Performance

Speed up image display and enable more images to be displayed at one time.

Overview

You use textures and atlases to hold your app’s image data, but need to follow recommended practices for the best performance. Because image data can consume much video memory, you should preload textures ahead of time and reuse them where possible. In addition, grouping related textures into an atlas enables them to draw faster. See SKTexture and SKTextureAtlas class references for more information.

Repeat Images with Textures

Textures, represented as SKTexture objects, are shared images used to render sprites. Always use textures whenever you need to apply the same image to multiple sprites. Usually you create textures by loading image files stored in your app bundle. However, SpriteKit can also create textures for you at runtime from other sources, including Core Graphics images, or even by rendering a node tree into a texture.

Preload Textures

SpriteKit simplifies texture management by handling the lower-level code required to load textures and make them available to the graphics hardware. Texture management is automatically managed by SpriteKit. However, if your game uses a large number of images, you can improve its performance by taking control of parts of the process. Primarily, you do this by telling SpriteKit explicitly to load a texture using SKTexture preload(_:withCompletionHandler:) or SKTextureAtlas preload(completionHandler:).

Group Similar Textures in an Atlas

A texture atlas is a group of related textures that are used together in your game. For example, you might use a texture atlas to store all of the textures needed to animate a character or all of the tiles needed to render the background of a gameplay level. SpriteKit uses texture atlases to improve rendering performance.