Drawing SpriteKit Content in a View

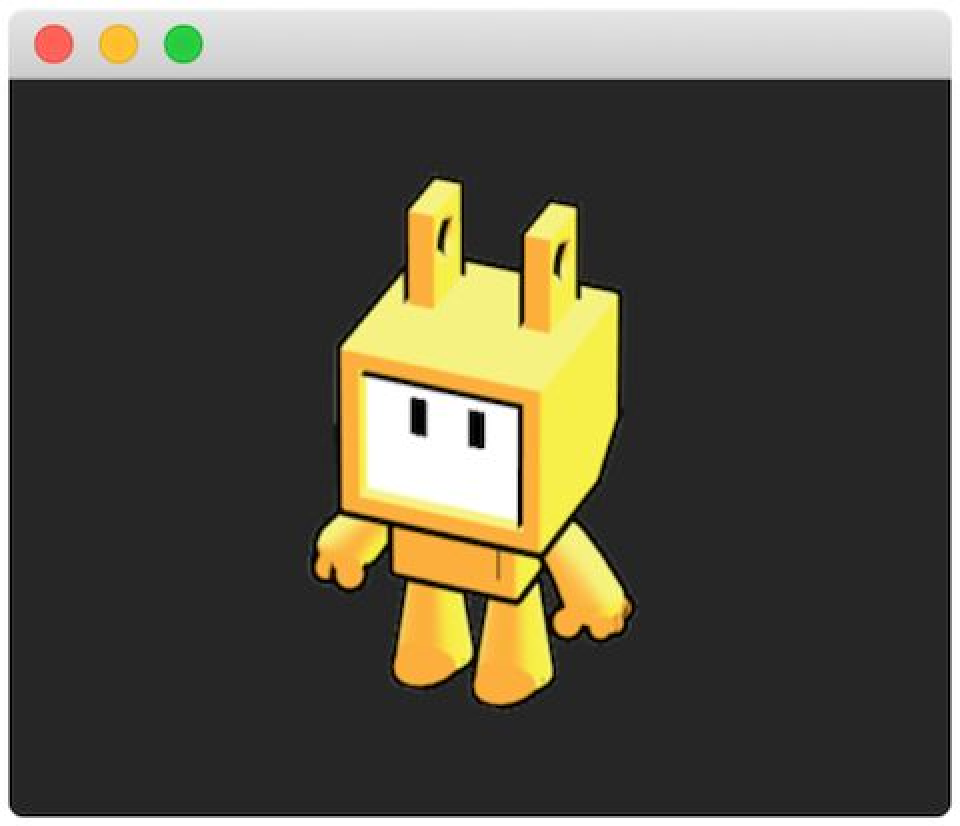
Display visual content using SpriteKit.

Overview

Display SpriteKit content on the screen by configuring a SpriteKit renderer, its scene, and the visual objects you lay out on top of the scene. SpriteKit provides objects that are designed specifically for various types of content, but for simplicity, this article displays an image in a view.

Note: *There are other ways to draw SpriteKit content besides using a view.*

Figure 1: An image displayed in a SpriteKit view



Because the code in this article sets up a view, you put the lines from each of the following code listings into a view controller's `viewDidLoad()` function.

Create the Scene

Everything displayed with SpriteKit is done through a scene object, which is an instance of `SKScene`. Use the following code to set up a basic scene:

```
let scene = SKScene(size: skView.bounds.size)

// Set the scene coordinates (0, 0) to the center of the screen.
scene.anchorPoint = CGPoint(x: 0.5, y: 0.5)
```

When you pass the size of the view's bounds to the scene initializer, you size the scene to the view's size. When you set the scene's `anchorPoint` to (0.5, 0.5), you determine that the coordinates (0, 0) map to the center of the view.
For further discussion about how setting the anchorPoint changes an object's position in the view.

Lay Out Visual Content on Top of the Scene

You use node objects to display graphics in a SpriteKit view. SpriteKit provides different nodes depending on the content. In this case, use an `SKSpriteNode` to display an image in the view:

```
let image = SKSpriteNode(imageNamed: "myImage.png")

// Add the image to the scene.
scene.addChild(image)
```

After you set up the scene, you display it in the view by calling `presentScene(_:)`:

```
if let skView = self.view as? SKView {
    skView.presentScene(scene)
}
```

Because the code in this article sets up a view, you add it to your view controller's `viewDidLoad()` function.