GAME 3004

SpriteKit - Week 3

Lesson 3

Expectation

Good understanding of how to use Swift within a Playground

Outcome

Introduction to SpriteKit within an Xcode Project

Key Concepts

SpriteKit Intro
SuperSpaceMan
Xcode Templates
Exploring Xcode IDE
GameViewController
SKView
SKScene
View Hierarchy
SKNode

SpriteKit

SpriteKit is an **animation and graphics** rendering framework

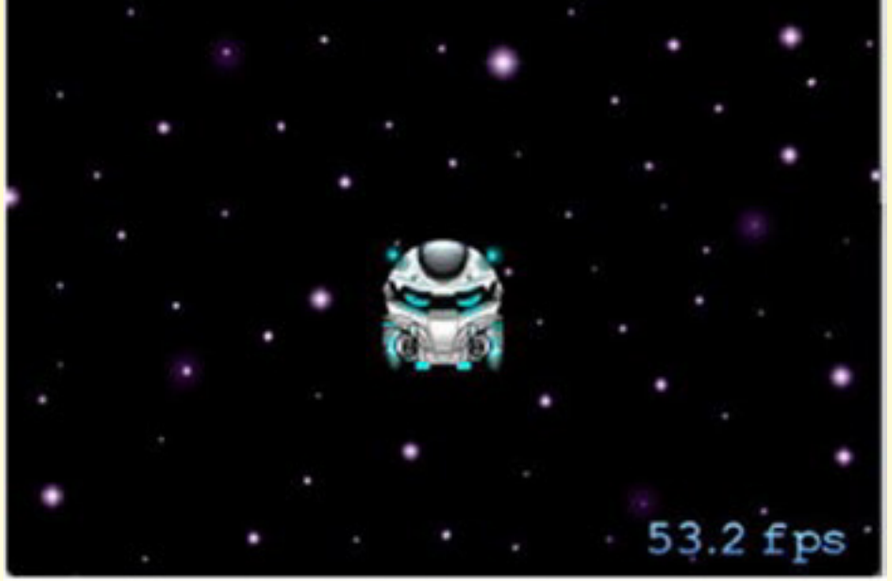
Easily **animate textured images**

Suitable for **apps or games** that require flexibility in how animations are handled

Render frames efficiently using **graphics hardware at 60 FPS**

Add **particle effects**

Includes integrated **physics library**



SUPERSPACEMAN

2D ENDLESS SHOOTER

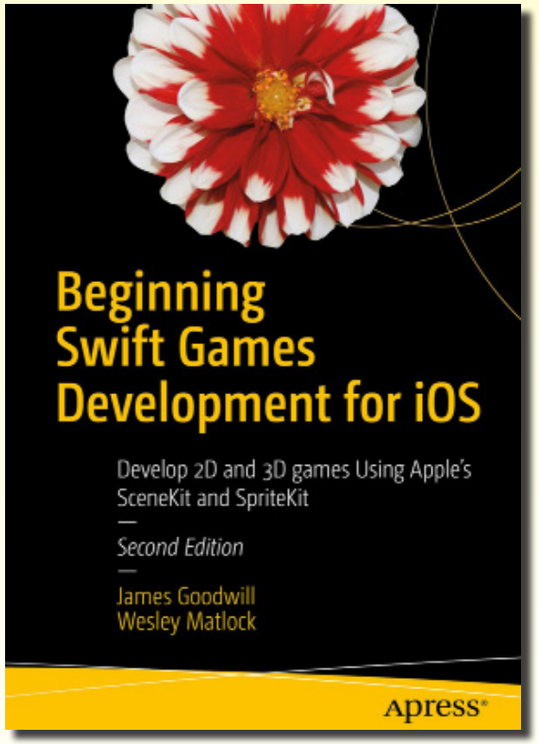
inspired by sonic jump fever

SuperSpaceMan

We're finally done using **Playgrounds** for now...

Lets create a brand new **SpriteKit** project in **Xcode**

File > **New** > **Project**



XCode Templates

- Augumented Reality Apps
- Document Based Apps
- Page-Based Apps
- Tabbed Apps
- iMessage Apps
- Single View Apps
- Game Apps

SuperSpaceMan

Product Name - Name of the product

Team - Licensed Developer for Product Distribution/Deployment

Organization Name - Name of Organization this product belongs to

Organization ID - Unique organization identifier for Bundle Id

Bundle ID - Organization Identifier + Product Name

Language - **Swift** or Objective-C

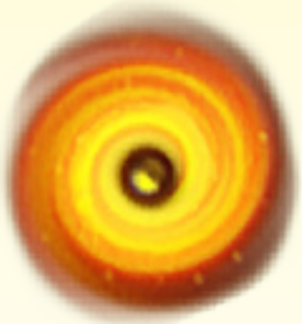
Game Technology - **SpriteKit**, SceneKit, Metal

SuperSpaceMan - Design

SuperSpaceMan

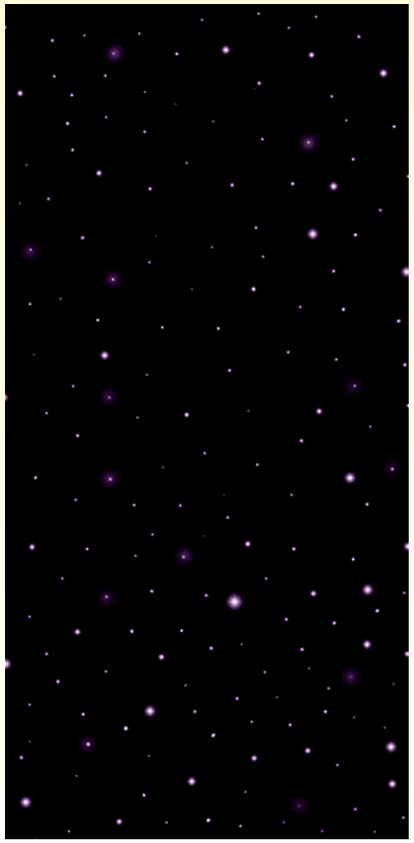
Power-Up

Black Holes

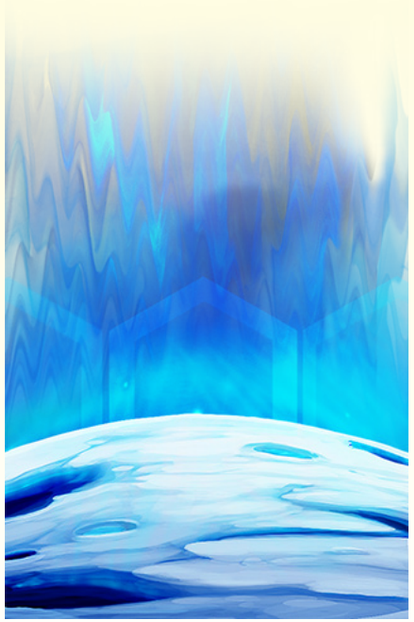


SuperSpaceMan - Design

Background



Planet



Stars

XCode IDE - Assets Catalog

Use **asset catalogs** to **organize** and **manage** the **different asset types** used by your **app**, such as **images**, **sprites**, **textures**, **stickers**, **and data**

An **asset catalog** can contain **multiple types of assets** and the **assets** can be **grouped into folders**

XCode IDE - Assets Catalog

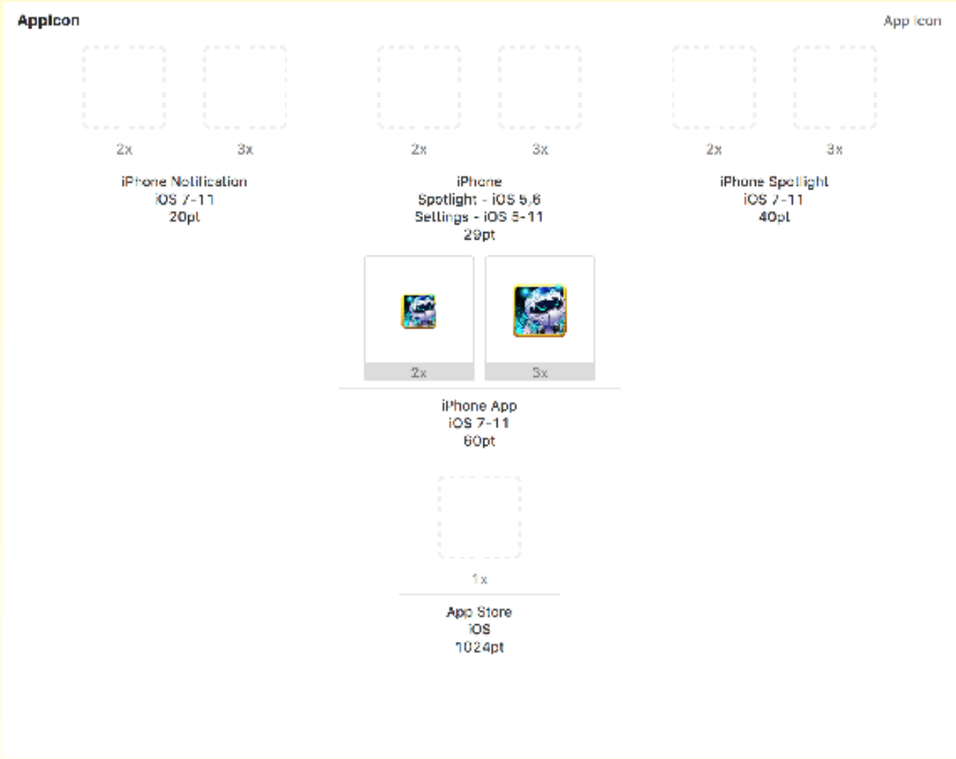
An **image set** can contain different files in any supported image format for the **2x and 3x images** used for an **iPhone and for an iPad.**

1x 2x 3x

XCode IDE - App Icons

You can use websites like **makeappicon.com**

to create every icon size needed for your **app**



XCode IDE - App Delegate

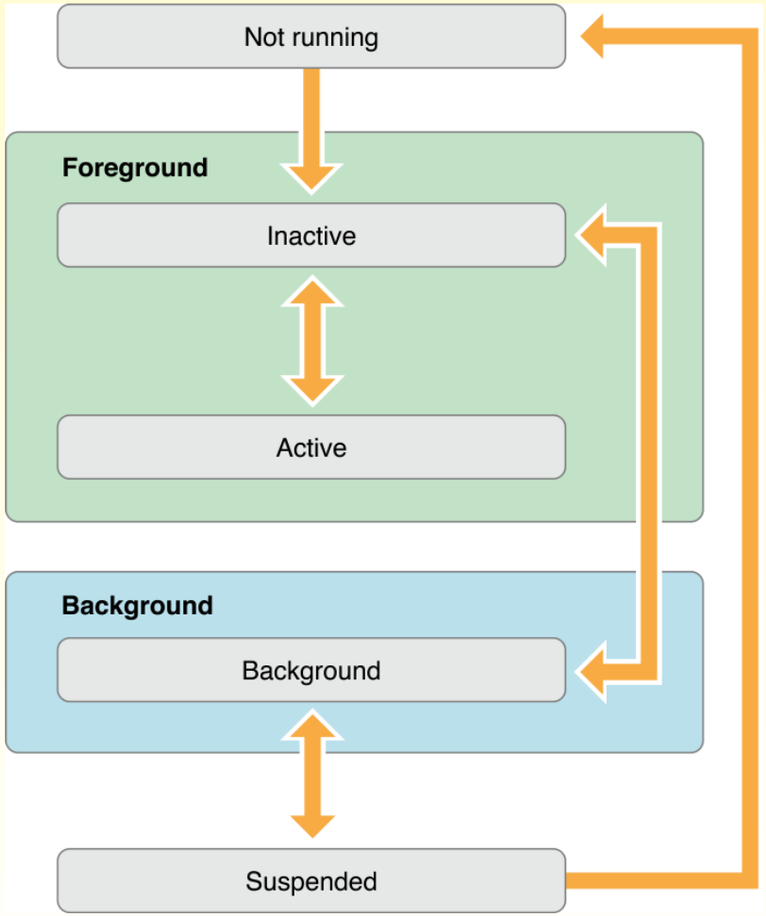
It contains your **App's startup code.**

This is the root of your **App's launch.**

It responds to when your App **transitions** from different states, like the **foreground to the background**.

It responds to **notifications** originating from outside the App, such as **low-memory warnings**, **download completion notifications**, and more.

XCode IDE - App Delegate



XCode IDE - App Delegate

Not running - App has **not** been **launched** or **terminated**

Inactive - App is **running** in the **foreground** but is **not receiving events**

Active - App is **running** in the **foreground** and **receiving events**

Background - App is **executing code** but is **not visible on screen**.

Suspending - App is in **memory** but is **not executing code**

XCode IDE - Info.plist

An information property list file is a **structured text file** that contains **essential configuration information**

The **root XML** node is a **dictionary**, whose contents are a **set of keys and values** describing different aspects of the bundle

The system uses these keys and values to **obtain information** about your **app** and how it is **configured**.

XCode IDE - Storyboard File

A **storyboard** is a **visual representation** of the user interface of an **iOS application**, showing screens of content and the connections between those screens

A storyboard is **composed of a sequence of scenes**, each of which represents a **view controller and its views**; scenes are **connected by segue objects**, which represent a transition **between two view controllers**.

GameViewController

Inherits from UIViewController class and is **apart** of the **UIKit framework**

Every **UIViewController** has an **UIView** embedded inside of it

The **UIViewController** will be notified once the **UIView** is **loaded** or has **appeared** or **disappeared**

GameViewController

viewDidLoad - The UIView has been **loaded** for the **first time**

viewWillAppear - The UIView is **about to appear** on the **foreground**

viewDidAppear - The UIView **has appeared** to the **foreground**

viewWillDisappear - The UIView is **about** to be **removed** from the **foreground**

viewDidDisappear - The UIView has **been removed** from the **foreground**

SKView

SKView builds on top of the **UIView** class, which is apart of the **UIKit Framework**

It provides the **view** for implementing your interface, the **event handling infra-structure** for delivering **touch events**

Use **presentScene** function to add an **SKScene** to the **view**

SKScene

A **SKScene** is **presented** by a **SKView**

The scene includes **properties** that define where the **scene's** origin is **positioned** and the **size** of the scene

Set the **SKSceneScaleMode enum** to set how the **scene scales** to fit the **SKView**

SKScene

SKSceneScaleMode.**fill** - Fill SKView with **no consideration** for ratio of width and height

SKSceneScaleMode.**aspectFill** - Fill SKView while **maintaining the aspect ratio** of the scene; **may be some cropping** if aspect ratio is different

SKSceneScaleMode.**aspectFit** - Fill SKView while **mantaining the aspect ratio** of the scene; **may be some letterboxing** if aspect ratio is different

SKSceneScaleMode.**resizeFill** - Fit SKView **exactly**

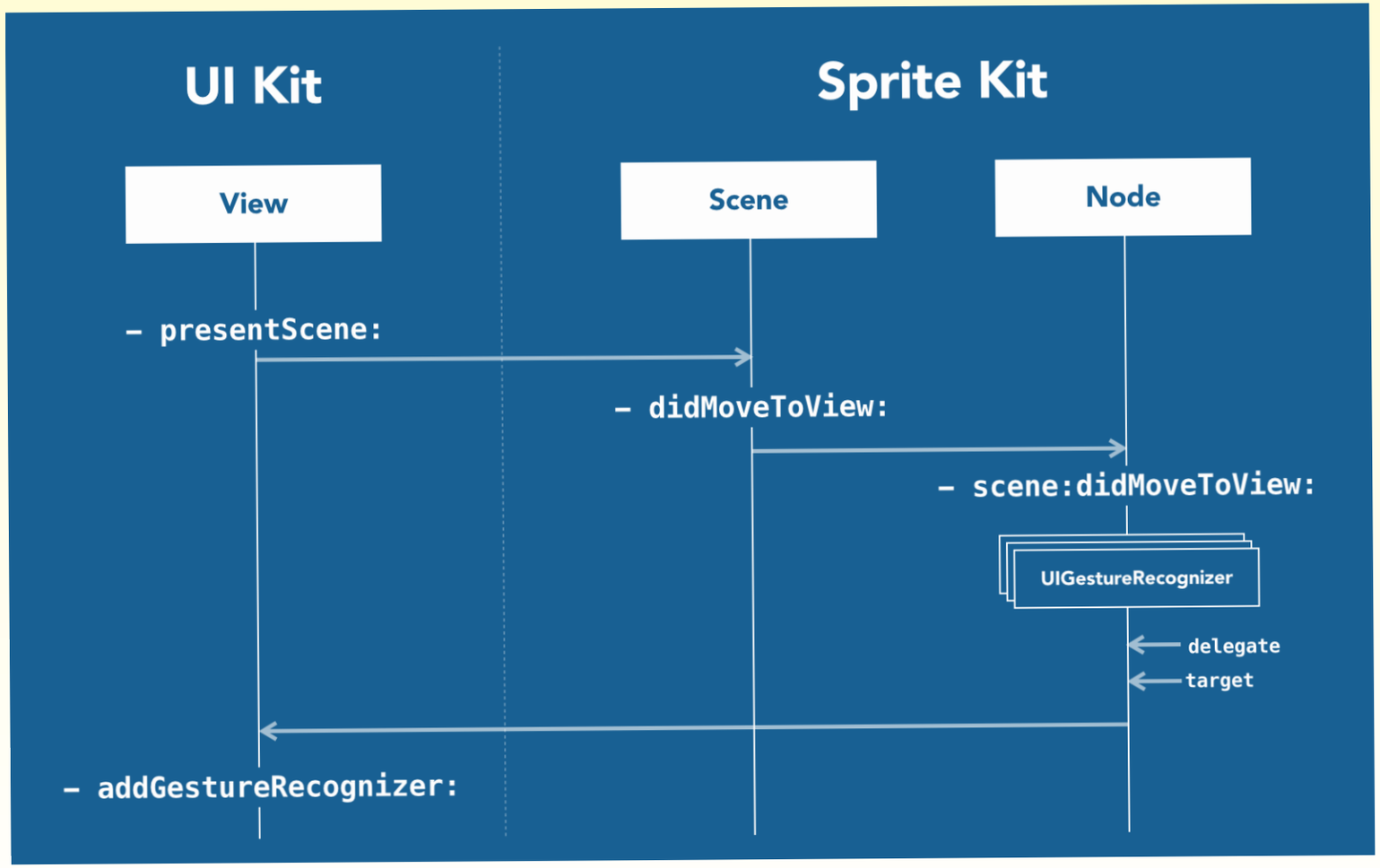
SKScene

sceneDidLoad() - Called after scene has been **initialized or declared**

willMove(from: SKView) - Called **before** scene is **removed from a view**

didMove(to: SKView) - Called **immediately after** a scene is **presented** by a view

View Hierarchy



SKScene (Game)

SKView (UIView)

GameViewController (UIView-Controller)

UIWindow

SKNode

The **SKNode** class doesn't draw any **visual content**

Its primary role is to **provide baseline behavior** that the other **node classes** use

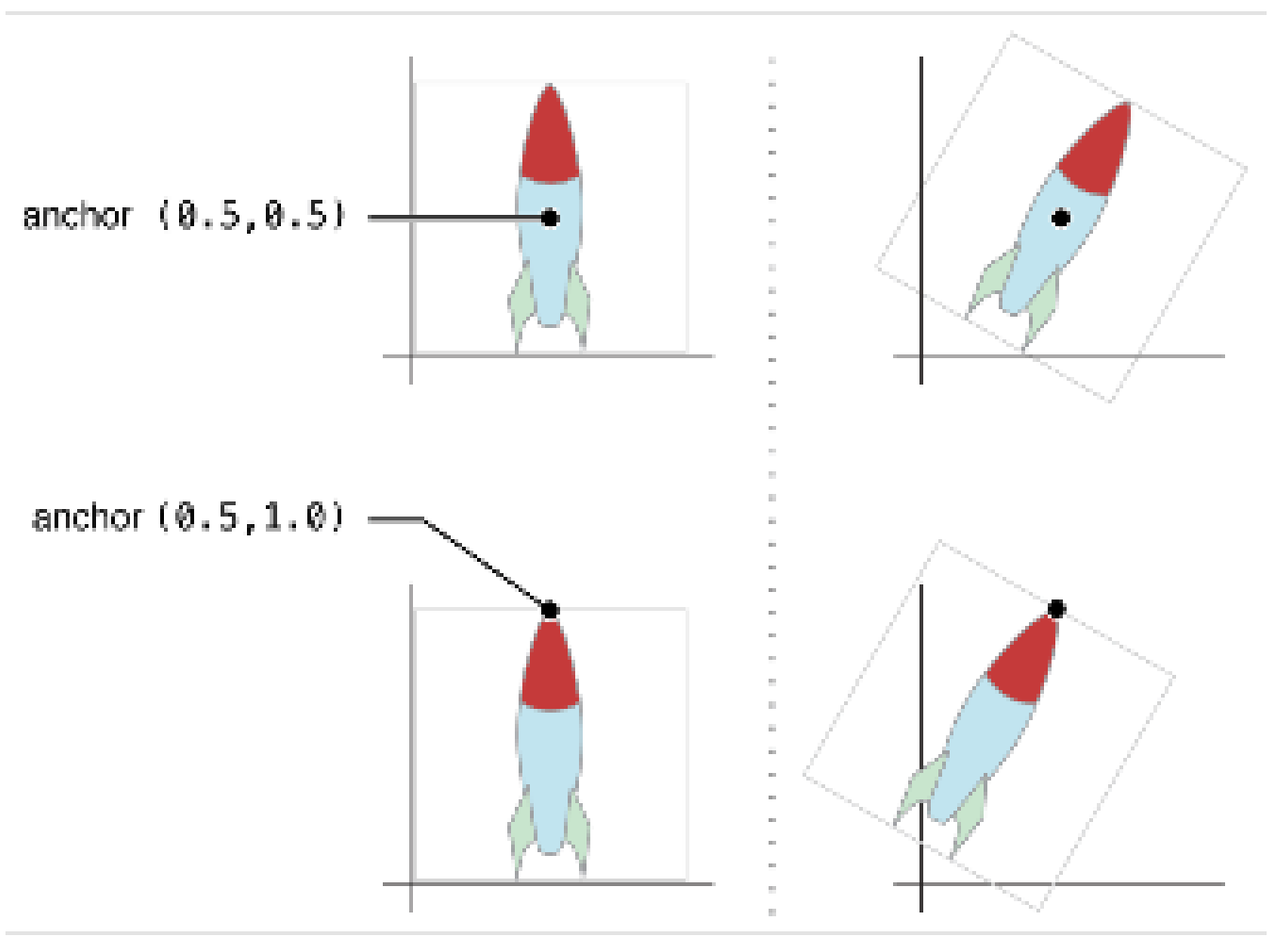
All **visual elements** in a SpriteKit-based game are **drawn** using predefined **SK-Node subclasses.**

SKSpriteNode

An SKSpriteNode is a **node** that **draws a texture** (optionally blended with a color), an **image, a colored square.**

By default, the SKSpriteNode's **anchor** is **(0.5, 0.5)**, which is the **center of the node.**

SKSpriteNode



SKSpriteNode

```
let spaceship = SKSpriteNode(imageNamed: “rocket.png”)
spaceship.position = CGPoint(x: 100, y: 100)
self.addChild(spaceship)
```

SKShapeNode

Shape nodes are also very useful for building and **displaying debugging information** on top of your game content

SKSpriteNode class offers **higher performance** than this class, so use **shape nodes sparingly.**

SKLabelNode

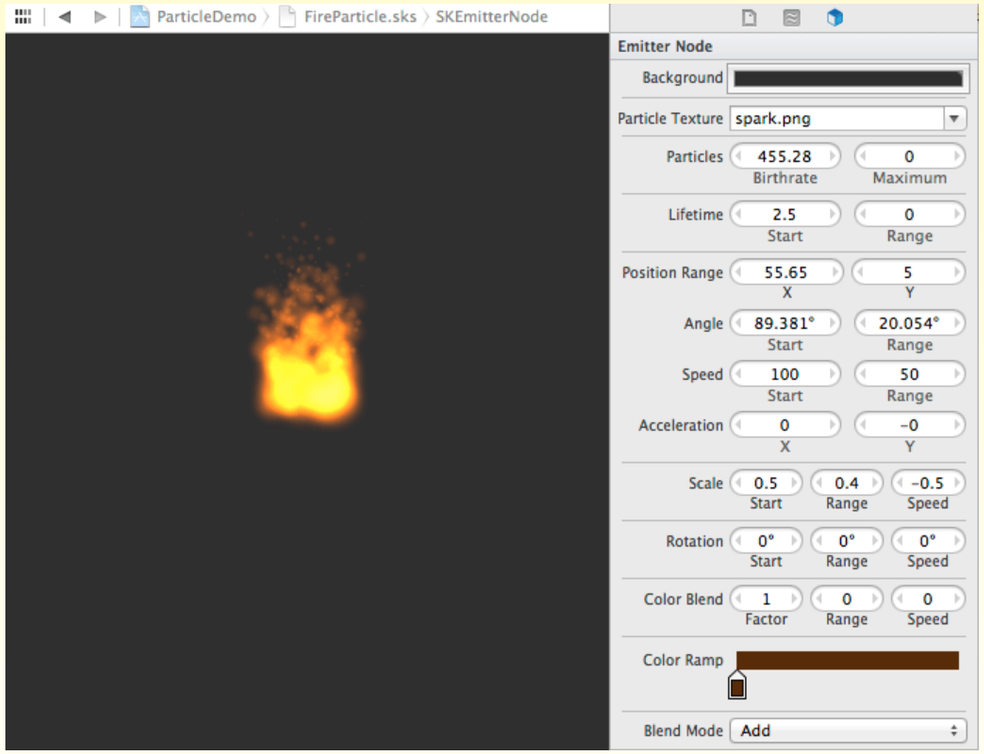
An **SKLabelNode** is a **performance friendly** way of **displaying text**

```
let winner = SKLabelNode(fontNamed: "Chalkduster")
winner.text = "You Win!"
winner.fontSize = 65
winner.fontColor = SKColor.green
```

SKEmitterNode

A **SKEmitterNode** object is a node that **automatically creates and renders small particle sprites**

Particles can also be created using the **Particle Emitter Editor** within **Xcode**



SKVideoNode

A node that displays **video content**.

SKVideoNode is built on top of the **AVPlayer class**, using **AVKit framework**

```
let sample = SKVideoNode(fileNamed: “sample.mov”)
sample.position = CGPoint(x: frame.midX, y: frame.midY)
addChild(sample)
sample.play()
```