颗粒动画

模拟瓦斯爆炸、暴雨和令人窒息的烟雾在屏幕上让我们惊讶这种动画。接下来创建自己的特效并使用粒子发射器。瀑布、火、烟和雨影响所有涉及大量的视觉项目-粒子有共同的物理特性,但可能仍有自己的独特的大小、方向、自旋和轨道。粒子很适合创建逼真的效果,每个粒子可以是随机的和不可预测的,就像对象是在自然界中。例如,每个雨滴在雷雨可能有一个独特的大小,形状,和速度。

创建发射层

可以使用Core Animation.的CAEmitterLayer这个类来创建粒子效果

let rect = CGRect(x: 0.0, y: 100.0, width: view.bounds.width, height: 50.0)

let emitter = CAEmitterLayer() emitter.frame = rect view.layer.addSublayer(emitter)

创建个emitter，设置层的Frame,占据整个屏幕的宽度和位置层屏幕的顶部附近。

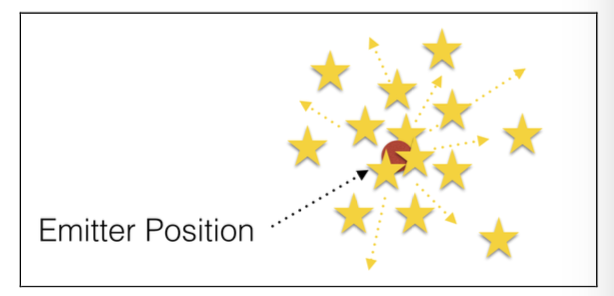
viewDidLoad():

emitter.emitterShape = kCAEmitterLayerRectangle

发射器的形状通常会影响粒子被创建的位置，当你创建3D粒子时也会影响坐标Z；

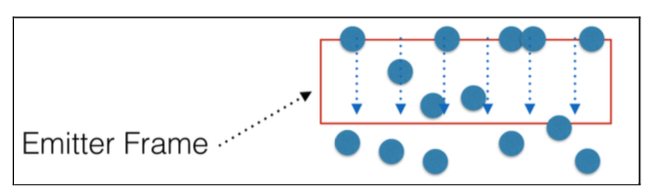
点状：

kCAEmitterLayerPoint ：发射器的形状设置为kCAEmitterLayerPoint会使所有的粒子从同样的点发出；



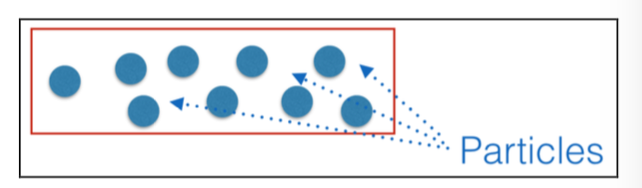
线状：

kCAEmitterLayerLine ：发射器的形状kCAEmitterLayerLine创建所有粒子发射器框架的顶部。这是发射器形状有用的瀑布效应;水粒子出现在顶部边缘的瀑布,瀑布一样:



矩形：

最后,kCAEmitterLayerRectangle创建粒子发射器形状随机通过给定的矩形区域:

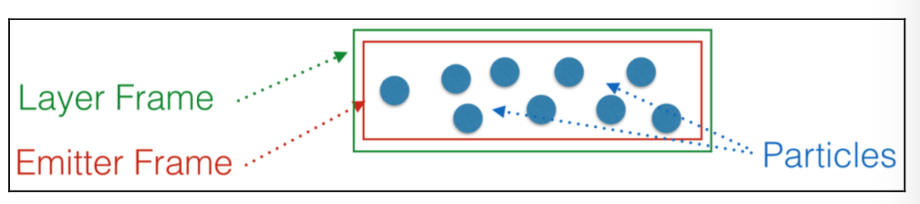


这种发射器形状很适合许多不同的效果,包括泡沫捧着爆米花碳酸饮料。

emitter.emitterPosition = CGPoint(x: rect.width/2, y: rect.height/2)

emitter.emitterSize = rect.size

结合emitterShape、emitterPosition、和emitterSize定义发射器的Frame，在此设置的尺寸和Layer层一样大，位置在Layer层的中间；



创建发射器的单位：

发射器单元代表一个粒子资源的数据模型，**emitter cell**. 继承自CAEmitterLayer，一个简单的发射层可以包含一个或多个单元；

let emitterCell = CAEmitterCell()

 emitterCell.contents = UIImage(named: "flake.png")!.CGImage

创建一个新的cell，并设置flake图片为他的内容，

emitterCell.birthRate = 20

emitterCell.lifetime = 3.5

emitter.emitterCells = [emitterCell]

每秒创建20个雪花，并保存3.5秒；意味有70个同时在屏幕上；emitterCells属性数组是所有发射器单元。记住,你可以有多个发射器单元；

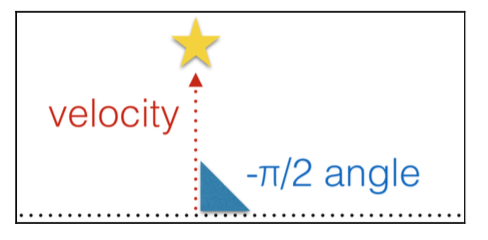
改变粒子的方向：

emitterCell.yAcceleration = 70.0

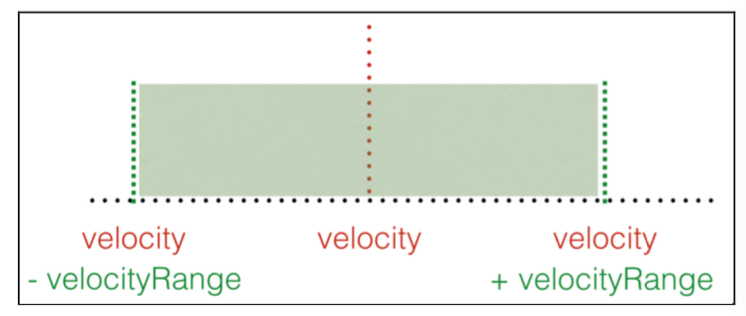
emitterCell.xAcceleration = 10.0

设置Y、X方向上的加速度，

emitterCell.velocity = 20.0 emitterCell.emissionLongitude = CGFloat(-M\_PI\_2)

粒子的发射经度（emissionLongitude）是初始角度,和速度参数（velocity）集的初始速度粒子就像这样:

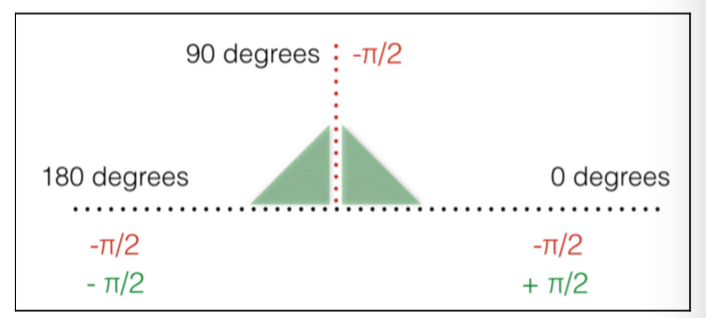
动画看起来更好点了。但这些粒子朝向一致。这是因为每个粒子有相同的初始角、速度和加速度。

更改随机速度，告诉发射器随机范围，因为每个粒子都有初始化速度20，所以最后的随机速度为-180（20 - 200） 到 220 （20 + 200）；

emitterCell.velocityRange = 200.0

更改发射角度，原理同随机速度一致

emitterCell.emissionRange = CGFloat(M\_PI\_2)



//随机色前提要设置初始值

emitterCell.color = UIColor(red: 0.9, green: 1.0, blue: 1.0, alpha: 1.0).CGColor

emitterCell.redRange = 0.1

emitterCell.greenRange = 0.1

emitterCell.blueRange = 0.1

随机出现的大小

emitterCell.scale = 0.8

emitterCell.scaleRange = 0.8

设置每秒相对于原始的尺寸粒子比例下降%15

emitterCell.scaleSpeed = -0.15

大颗粒会大幅缩小在消失之前,而小颗粒将完全消失在时间结束之前。

基本上效果不错；但是还可以改进，将雪花的透明度更改

emitterCell.alphaRange = 0.75

emitterCell.alphaSpeed = -0.15