

Blending a Sprite with Different Interpretations of Alpha

Reinterpret a sprite's alpha property to react differently to the objects below it.

Overview

The final stage of rendering is to blend the sprite's texture into its destination frame buffer. The default behavior uses the alpha values of the texture to blend the texture with the destination pixels. However, you can use other blend modes when you want to add other special effects to a scene.

You control the sprite's blending behavior using the `blendMode` property. For example, an additive blend mode is useful to combine multiple sprites together, for effects such as for fire or lighting. The following code shows how to position three overlapping sprite nodes in a circle to demonstrate the effect of different blend modes:

```
let blendMode = SKBlendMode.alpha
let imageNames = ["redCircle", "greenCircle", "blueCircle"]

for (index, imageName) in imageNames.enumerated() {
    let node = SKSpriteNode(imageNamed: imageName)

    node.alpha = 0.5
    node.blendMode = blendMode

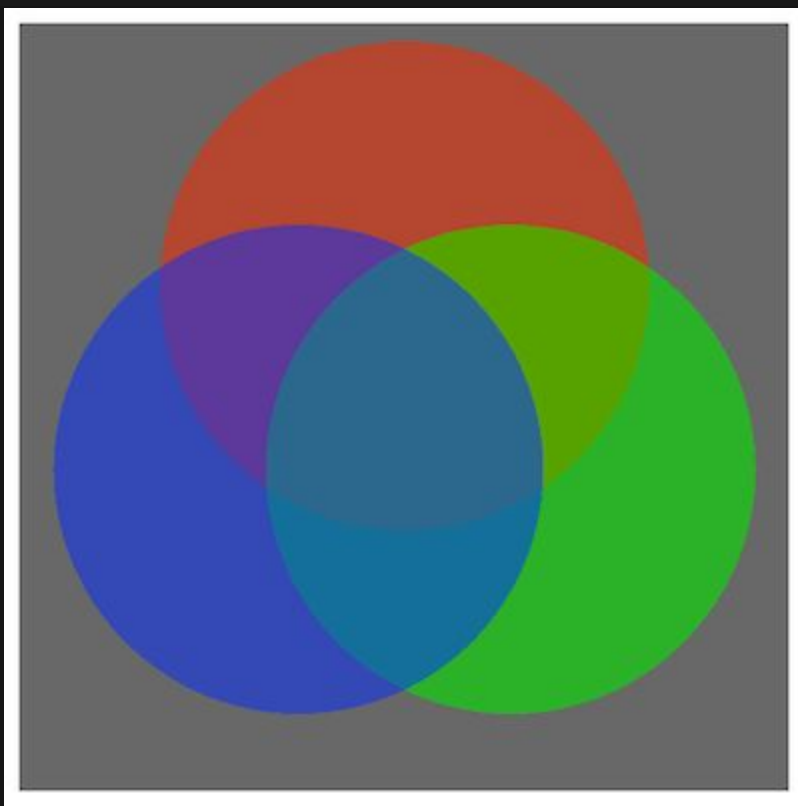
    let angle = (CGFloat.pi * 2) / CGFloat(colors.count) * CGFloat(index)

    let positionX = 320 + sin(angle) * radius / 2
    let positionY = 320 + cos(angle) * radius / 2

    node.position = CGPoint(x: positionX, y: positionY)

    scene.addChild(node)
}
```

With the default blend mode of `SKBlendMode.alpha`, the three circles look like:



However, with a blend mode of `SKBlendMode.add`, the color values are added together, creating a scene that looks like:

