

Getting Started with Shape Nodes

Create a filled or stroked shape from a path object.

Overview

A graphics path is a collection of straight lines and curves that can define either open or closed subpaths. You can specify separate rendering behavior for the filled and stroked portion of the path. Each part can be rendered using either a solid color or a texture; if you need to render more sophisticated effects, you can also use a custom shader.

Shape nodes are useful for content that cannot be easily decomposed into simple textured sprites. They're also useful for building and displaying debugging information on top of your game content. However, the `SKSpriteNode` class offers higher performance than this class, so use shape nodes sparingly.

Create a Shape Node

The following code demonstrates how to create a shape node. The example creates a circle with a blue interior and a white outline. The path is created and attached to the shape node's `path` property.

```
let path = CGMutablePath()
path.addArc(center: CGPoint.zero,
            radius: 15,
            startAngle: 0,
            endAngle: CGFloat.pi * 2,
            clockwise: true)
let ball = SKShapeNode(path: path)
ball.lineWidth = 1
ball.fillColor = .blue
ball.strokeColor = .white
ball.glowWidth = 0.5
```

You can see from the code that the shape has three essential elements:

>> The interior of the shape is filled. The `fillColor` property specifies the color used to fill the interior.

>> The outline of the shape is rendered as a line. The `strokeColor` and `lineWidth` properties define how the line is stroked.

>> A glow extends from the outline. The `glowWidth` and `strokeColor` properties define the glow.

You can disable any of these elements by setting its color to `[SKColor clearColor]`.

The shape node provides properties that let you control how the shape is blended into the framebuffer. You use these properties the same way as the properties of the `SKSpriteNode` class.