Animating a Sprite by Changing its Texture

Load a sequence of images and play them back at a rate you define, while optionally looping the resulting animation.

Overview

A sprite's texture property points to its current texture. You can change this property to point to a new texture. The next time the scene renders a new frame, it renders with the new texture. Whenever you change the texture, you may also need to change other sprite properties—such as size, anchorPoint, and centerRect—to be consistent with the new texture. Usually, it is better to ensure that all the artwork is consistent so that the same values can be used for all of the textures. That is, the textures should have a consistent size and anchor point placement so that your game does not need to update anything other than the texture.

Because animation is a common task, you can use actions to animate a series of textures on a sprite. The following code shows how to use an array of frames created to animate a sprite's texture.

```
let walkAnimation = SKAction.animate(with: monsterWalkTextures,
                                     timePerFrame: 0.1)

node.run(walkAnimation)
// insert other code here to move the monster.
```

SpriteKit provides you with the option of animating a sprite's texture. It doesn't impose a specific design on your animation system. This means you need to determine what kinds of animations a sprite may need and then design your own animation system to switch between those animations at runtime. For example, a monster might have walk, fight, idle, and death animation sequences—and it's up to you to decide when to switch between these sequences.