SKActions

Movement

class func moveBy(x: CGFloat, y: CGFloat, duration: TimeInterval)
Creates an action that moves a node relative to its current position.

class func move(by: CGVector, duration: TimeInterval)
Creates an action that moves a node relative to its current position.

class func move(to: CGPoint, duration: TimeInterval)
Creates an action that moves a node to a new position.

class func moveTo(x: CGFloat, duration: TimeInterval)
Creates an action that moves a node horizontally.

class func moveTo(y: CGFloat, duration: TimeInterval)
Creates an action that moves a node vertically.

class func follow(CGPath, duration: TimeInterval)
Creates an action that moves the node along a relative path, orienting the node to the path.

class func follow(CGPath, speed: CGFloat)
Creates an action that moves the node along a relative path at a specified speed, orienting the node to the path.

class func follow(CGPath, asOffset: Bool, orientToPath: Bool, duration: TimeInterval)
Creates an action that moves the node along a path.

class func follow(CGPath, asOffset: Bool, orientToPath: Bool, speed: CGFloat)
Creates an action that moves the node at a specified speed along a path.

Rotation

class func rotate(by Angle: CGFloat, duration: Time Interval)
Creates an action that rotates the node by a relative value.

class func rotate(to Angle: CGFloat, duration: Time Interval)
Creates an action that rotates the node counterclockwise to an absolute angle.

class func rotate(to Angle: CGFloat, duration: Time Interval, shortest Unit Arc: Bool)
Creates an action that rotates the node to an absolute value.

Animation Speed

class func speed(by: CGFloat, duration: Time Interval)
Creates an action that changes how fast the node executes actions by a relative value.

class func speed(to: CGFloat, duration: Time Interval)
Creates an action that changes how fast the node executes actions.

Scaling

class func scale(by: CGFloat, duration: Time Interval)
Creates an action that changes the x and y scale values of a node by a relative value.

class func scale(to: CGSize, duration: Time Interval)
Creates an action that changes the x and y scale values of a node to achieve

class func scale(to: CGFloat, duration: Time Interval)
Creates an action that changes the x and y scale values of a node.

class func scale X(by: CGFloat, y: CGFloat, duration: Time Interval)
Creates an action that adds relative values to the x and y scale values of a node.

class func scale X(to: CGFloat, y: CGFloat, duration: Time Interval)
Creates an action that changes the x and y scale values of a node.

class func scale X(to: CGFloat, duration: Time Interval)
Creates an action that changes the x scale value of a node to a new value.

class func scale Y(to: CGFloat, duration: Time Interval)
Creates an action that changes the y scale value of a node to a new value.

Display

class func unhide()
Creates an action that makes a node visible.

class func hide()
Creates an action that hides a node.

Transparency

class func fade In(with Duration: Time Interval)
Creates an action that changes the alpha value of the node to *1.0*.

class func fade Out(with Duration: Time Interval)
Creates an action that changes the alpha value of the node to *0.0*.

class func fade Alpha(by: CGFloat, duration: Time Interval)
Creates an action that adjusts the alpha value of a node by a relative value.

class func fade Alpha(to: CGFloat, duration: Time Interval)
Creates an action that adjusts the alpha value of a node to a new value.

Content

class func resize(by Width: CGFloat, height: CGFloat, duration: Time Interval)
Creates an action that adjusts the size of a sprite.

class func resize(to Height: CGFloat, duration: Time Interval)
Creates an action that changes the height of a sprite to a new absolute value.

class func resize(to Width: CGFloat, duration: Time Interval)
Creates an action that changes the width of a sprite to a new absolute value.

class func resize(to Width: CGFloat, height: CGFloat, duration: Time Interval)
Creates an action that changes the width and height of a sprite to a new absolute value.

class func set Texture(SKTexture)
Creates an action that changes a sprite's texture.

class func set Texture(SKTexture, resize: Bool)
Creates an action that changes a sprite's texture, possibly resizing the sprite.

class func animate(with: [SKTexture], time Per Frame: Time Interval)
Creates an action that animates changes to a sprite's texture.

class func animate(with: [SKTexture], time Per Frame: Time Interval, resize: Bool, restore: Bool)
Creates an action that animates changes to a sprite's texture, possibly resizing the sprite.

class func set Normal Texture(SKTexture)
Creates an action that changes a sprite's normal texture.

class func set Normal Texture(SKTexture, resize: Bool)
Creates an action that changes a sprite's normal texture, possibly resizing the sprite.

class func animate(with Normal Textures: [SKTexture], time Per Frame: Time Interval)
Creates an action that animates changes to a sprite's normal texture.

class func animate(with Normal Textures: [SKTexture], time Per Frame: Time Interval, resize: Bool, restore: Bool)
Creates an action that animates changes to a sprite's texture.

class func colorize(with: UIColor, color Blend Factor: CGFloat, duration: Time Interval)
Creates an animation that animates a sprite's color and blend factor.

class func colorize(with Color Blend Factor: CGFloat, duration: Time Interval)
Creates an action that animates a sprite's blend factor.

Physics

class func apply Force(CGVector, duration: Time Interval)
Creates an action that applies a force to the center of gravity of a node's physics body.

class func apply Torque(CGFloat, duration: Time Interval)
Creates an action that applies a torque to an node's physics body.

class func apply Force(CGVector, at: CGPoint, duration: Time Interval)
Creates an action that applies an force to a specific point on a node's physics body.

class func apply Impulse(CGVector, duration: Time Interval)
Creates an action that applies an impulse to the center of gravity of a physics body.

class func apply Angular Impulse(CGFloat, duration: Time Interval)
Creates an action that applies an angular impulse to a node's physics body.

class func apply Impulse(CGVector, at: CGPoint, duration: Time Interval)
Creates an action that applies an impulse to a specific point of a node's physics body.

class func change Charge(to: Float, duration: Time Interval)
Creates an action that changes the charge of a node's physics body to a new value.

class func change Charge(by: Float, duration: Time Interval)
Creates an action that changes the charge of a node's physics body by a relative value.

class func change Mass(to: Float, duration: Time Interval)
Creates an action that changes the mass of a node's physics body to a new value.

class func change Mass(by: Float, duration: Time Interval)
Creates an action that changes the mass of a node's physics body by a relative value.

class func strength(to: Float, duration: Time Interval)
Creates an action that animates a change of a physics field's strength.

class func strength(by: Float, duration: Time Interval)
Creates an action that animates a change of a physics field's strength to a value relative to the existing value.

class func falloff(to: Float, duration: Time Interval)
Creates an action that animates a change of a physics field's falloff.

class func falloff(by: Float, duration: Time Interval)
Creates an action that animates a change of a physics field's falloff to a value relative to the existing value.

Warping

class func animate(with Warps: [SKWarp Geometry], times: [NSNumber])
Creates an action to distort a node through a sequence of *SKWarp Geometry* objects.

class func animate(with Warps: [SKWarp Geometry], times: [NSNumber], restore: Bool)
Creates an action to distort a node through a sequence of *SKWarp Geometry* objects.

class func warp(to: SKWarp Geometry, duration: Time Interval)
Creates an action to distort a node based using an *SKWarp Geometry* object.

Removing Actions

class func remove From Parent()
Creates an action that removes the node from its parent.

Children

class func run(SKAction, on Child With Name: String)
Creates an action that runs an action on a named child object.

Combined Actions

class func group([SKAction])
Creates an action that runs a collection of actions in parallel.

class func sequence([SKAction])
Creates an action that runs a collection of actions sequentially.

class func `repeat`(SKAction, count: Int)
Creates an action that repeats another action a specified number of times.

class func repeat Forever(SKAction)
Creates an action that repeats another action forever.

Audio

class func play Sound File Named(String, wait For Completion: Bool)
Creates an action that plays a sound.

class func play()
Creates an action that tells an audio node to start playback.

class func pause()
Creates an action that tells an audio node to pause playback.

class func stop()
Creates an action that tells an audio node to stop playback.

class func change Playback Rate(to: Float, duration: Time Interval)
Creates an action that changes an audio node's playback rate to a new value.

class func change Playback Rate(by: Float, duration: Time Interval)
Creates an action that changes an audio node's playback rate by a relative amount.

class func change Volume(to: Float, duration: Time Interval)
Creates an action that changes an audio node's volume to a new value.

class func change Volume(by: Float, duration: Time Interval)
Creates an action that changes an audio node's volume by a relative value.

class func change Obstruction(to: Float, duration: Time Interval)
Creates an action that changes an audio node's obstruction to a new value.

class func change Obstruction(by: Float, duration: Time Interval)
Creates an action that changes an audio node's obstruction by a relative value.

class func change Occlusion(to: Float, duration: Time Interval)
Creates an action that changes an audio node's occlusion to a new value.

class func change Occlusion(by: Float, duration: Time Interval)
Creates an action that changes an audio node's occlusion by a relative value.

class func change Reverb(to: Float, duration: Time Interval)
Creates an action that changes an audio node's reverb to a new value.

class func change Reverb(by: Float, duration: Time Interval)
Creates an action that changes an audio node's reverb by a relative value.

class func stereo Pan(to: Float, duration: Time Interval)
Creates an action that changes an audio node's stereo panning to a new value.

class func stereo Pan(by: Float, duration: Time Interval)
Creates an action that changes an audio node's stereo panning by a relative value.

Custom Actions

init?(named: String)
Creates an action of the given name from an action file.

init?(named: String, duration: Time Interval)
Creates an action of the given name from an action file with a new duration.

init?(named: String, from: URL)
Creates an action of the given name from an action file.

init?(named: String, from: URL, duration: Time Interval)
Creates an action of the given name from an action file with a new duration.

class func custom Action(with Duration: Time Interval, action Block: (SKNode, CGFloat) -> Void)
Creates an action that executes a block over a duration.

class func perform(Selector, on Target: Any)
Creates an action that calls a method on an object.

class func run(() -> Void)
Creates an action that executes a block.

class func run(() -> Void, queue: Dispatch Queue)
Creates an action that executes a block on a specific dispatch queue.

Delay

class func wait(for Duration: Time Interval)
Creates an action that idles for a specified period of time.

class func wait(for Duration: Time Interval, with Range: Time Interval)
Creates an action that idles for a randomized period of time.

Kinematics

class func reach(to: CGPoint, root Node: SKNode, duration: Time Interval)
Creates an action that performs an inverse kinematic reach.

class func reach(to: CGPoint, root Node: SKNode, velocity: CGFloat)
Creates an action that performs an inverse kinematic reach.

class func reach(to: SKNode, root Node: SKNode, duration: Time Interval)
Creates an action that performs an inverse kinematic reach.

class func reach(to: SKNode, root Node: SKNode, velocity: CGFloat)
Creates an action that performs an inverse kinematic reach.

Reversing

func reversed()
Creates an action that reverses the behavior of another action.

Properties

var speed: CGFloat
A speed factor that modifies how fast an action runs.

var timing Mode: SKAction Timing Mode
The timing mode used to execute an action.

var timing Function: SKAction Timing Function
A block used to customize the timing function.

var duration: Time Interval
The duration required to complete an action.