SKPhysicsBody Properties

Initializing Volumes

init(circleOfRadius: CGFloat)
Creates a circular physics body centered on the owning node's origin.

init(circleOfRadius: CGFloat, center: CGPoint)
Creates a circular physics body centered on an arbitrary point.

init(rectangleOf: CGSize)
Creates a rectangular physics body centered on the owning node's origin.

init(rectangleOf: CGSize, center: CGPoint)
Creates a rectangular physics body centered on an arbitrary point.

init(bodies: [SKPhysicsBody])
Creates a physics body by performing a union of a group of volume-based physics bodies.

init(polygonFrom: CGPath)
Creates a polygon-shaped physics body.

init(texture: SKTexture, size: CGSize)
Creates a physics body from the contents of a texture.

init(texture: SKTexture, alphaThreshold: Float, size: CGSize)
Creates a physics body from the contents of a texture, capturing only the texels that exceed a specified transparency value.

Initializing Edges

init(edgeLoopFrom: CGRect)
Creates an edge loop from a rectangle.

init(edgeFrom: CGPoint, to: CGPoint)
Creates an edge between two points.

init(edgeLoopFrom: CGPath)
Creates an edge loop from a path.

init(edgeChainFrom: CGPath)
Creates an edge chain from a path.

Effects of Forces

var affectedByGravity: Bool
A Boolean value that indicates whether this physics body is affected by the physics world's gravity.

var allowsRotation: Bool
A Boolean value that indicates whether the physics body is affected by angular forces and impulses applied to it.

var isDynamic: Bool
A Boolean value that indicates whether the physics body is moved by the physics simulation.

Physical Properties

var mass: CGFloat
The mass of the body in kilograms.

var density: CGFloat
The density of the object in kilograms per square meter.

var area: CGFloat
The area covered by the body.

var friction: CGFloat
The roughness of the surface of the physics body.

var restitution: CGFloat
The bounciness of the physics body.

var linearDamping: CGFloat
A property that reduces the body's linear velocity.

var angularDamping: CGFloat
A property that reduces the body's rotational velocity.

Collisions & Contacts

var categoryBitMask: UInt32
A mask that defines which categories this physics body belongs to.

var collisionBitMask: UInt32
A mask that defines which categories of physics bodies can collide with this physics body.

var usesPreciseCollisionDetection: Bool
A Boolean value that determines whether the physics world uses a more precise collision detection algorithm.

var contactTestBitMask: UInt32
A mask that defines which categories of bodies cause intersection notifications with this physics body.

func allContactedBodies()
The physics bodies that this physics body is in contact with.

Applying Forces & Impulses

func applyForce(CGVector)
Applies a force to the center of gravity of a physics body.

func applyTorque(CGFloat)
Applies a torque to an object.

func applyForce(CGVector, at: CGPoint)
Applies a force to a specific point of a physics body.

func applyImpulse(CGVector)
Applies an impulse to the center of gravity of a physics body.

func applyAngularImpulse(CGFloat)
Applies an impulse that imparts angular momentum to an object.

func applyImpulse(CGVector, at: CGPoint)
Applies an impulse to a specific point of a physics body.

Inspecting the Body

var velocity: CGVector
The physics body's velocity vector, measured in meters per second.

var angularVelocity: CGFloat
The physics body's angular speed.

var isResting: Bool
A Boolean property that indicates whether the object is at rest within the physics simulation.