

Understanding Hit-Testing

Learn how to find child nodes at a given point by using hit-testing.

Overview

When SpriteKit processes touch or mouse events, it walks the scene to find the closest node that will accept the event. If the first node tested won't accept the event, SpriteKit checks the next closest node, and so on. This process is called hit-testing, and the order in which it's processed is essentially the reverse of drawing order.

For a node to be considered during hit-testing, its `isUserInteractionEnabled` property must be set to a true. The default value is a false for any node except a scene node. A node that will receive events needs to implement the appropriate responder methods from its parent class (`UIResponder` on iOS and `NSResponder` on macOS). This is one of the few places where you must implement platform-specific code in SpriteKit.

Sometimes, you also want to look for nodes directly, rather than relying on the standard event-handling mechanisms. In SpriteKit you can ask a node whether any of its descendants intersect a specific point in their coordinate system. Call the `atPoint(_:)` method to find the first descendant that intersects the point, or use the `nodes(at:)` method to receive an array of all of the nodes that intersect the point.