Class

GCStylus

An object that represents a physical stylus connected to the device.

visionOS 26.0+

```
class GCStylus
```

Mentioned in

Discovering and tracking spatial game controllers and styli

Overview

Use the `styli` property to get the currently connect stylus accessories when your application starts. Register for `GCStylusDidConnectNotification` and `GCStylusDidDisconnect Notification` to get notified when a stylus connects of disconnects while your application is running.

```
// Register for notifications
NotificationCenter.default.addObserver(self, selector: #selector(stylus(didConnect:)
NotificationCenter.default.addObserver(self, selector: #selector(stylus(didDisconnec

// Query current stylus devices
for stylus in GCStylus.styluses {
    ...
}

// Later, handle connection
@objc func stylus(didConnect notification: Notification) {
```

```
    guard let stylus = notification.object as? GCStylus else { return }
    ...
}
```

Check the `productCategory` to determine the type of stylus. A spatial stylus - capable of 6DoF tracking by Apple Vision Pro - has a `GCProductCategorySpatialStylus` category.

Use the `input` property to get the input profile of the stylus. A spatial stylus includes a pressure sensitive tip and an input cluster composed of two buttons.

- The primary button (`GCInputStylusPrimaryButton`) is the front button (closest to the stylus tip) in the input cluster of the stylus. This button is frequently used grab virtual objects.
- The secondary button (`GCInputStylusSecondaryButton`) is the middle button in the input cluster. It can measures pressure/force levels. It's intended to be used for controlling in-air drawing, selection, and generic interactions.
- The tip is also represented as a button (`GCInputStylusTip`).

```
guard let input = stylus.input else { return }
input.inputStateQueueDepth = 20
input.inputStateAvailableHandler = { input in
    // This block will be enqueued for execution when the state of
    // any stylus input changes.

    // Iterate through all input state changes since last execution of
    // the block.
    while let nextState = input.nextInputState() {
        // Use the value of `pressedInput.isPressed` for binary
        // interactions, such as object selection.
        let primaryButtonPressed = nextState.buttons[.stylusPrimaryButton]?.pressedI
        let secondaryButtonPressed = nextState.buttons[.stylusSecondaryButton]?.pres
        // Use the normalized press value for analog actions such as
        // controlling virtual ink flow.
        let secondaryButtonPressure = nextState.buttons[.stylusSecondaryButton]?.pre
        let tipPressure = nextState.buttons[.stylusTip]?.pressedInput.value

        ...
    }
}
```

Use the `haptics` property to get the haptics profile of the stylus. A spatial stylus may optionally support haptic feedback to a single locality - `GCHapticsLocalityDefault`.

Topics

Accessing the styli

`class var styli: [GCStylus]`

Get the collection of stylus accessories currently connected to the device.

Getting input values and haptics

`var input: (any GCDevicePhysicalInput)?`

Gets the input profile for the stylus.

`var haptics: GCDeviceHaptics?`

Gets the haptics profile for the stylus, if supported.

Retrieving the buttons

`var GCInputStylusPrimaryButton: String`

`var GCInputStylusSecondaryButton: String`

`var GCInputStylusTip: String`

Structures

`struct DidConnectMessage`

A message that posts after a stylus accessory connects to the device.

`struct DidDisconnectMessage`

A message that posts after a stylus accessory disconnects from the device.

Relationships

Inherits From

`NSObject`

Conforms To

CVarArg
CustomDebugStringConvertible
CustomStringConvertible
Equatable
GCDevice
Hashable
NSObjectProtocol

See Also

Game controllers

{} Supporting Game Controllers

Support a physical controller or add a virtual controller to enhance how people interact with your game through haptics, lighting, and motion sensing.

Letting players use their second-generation Siri Remote as a game controller

Support the second-generation Siri Remote as a game controller in your Apple TV game.

Discovering and tracking spatial game controllers and styli

Receive controller and stylus input to interact with content in your augmented reality app.

`protocol` GCDevice

A protocol that defines a common interface for game input devices.

`class` GCController

A representation of a real game controller, a virtual controller, or a snapshot of a controller.

`class` GCRacingWheel

An object that represents a physical racing wheel controller connected to a device.

`class` GCKeyboard

An object that represents a physical keyboard connected to a device.

`class` GCMouse

An object that represents a physical mouse connected to a device.