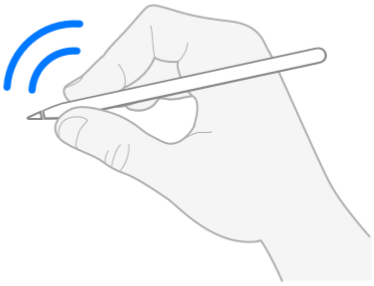
Article

Handling double taps from Apple Pencil

Detect and respond to double taps a person makes on Apple Pencil.

Overview

You can use Apple Pencil interactions to allow people to access functionality in your app quickly. Double-tapping Apple Pencil lets a person perform actions such as switching between drawing tools without moving the pencil to another location on the screen.



Register for a double tap

To respond to double taps from Apple Pencil in your app, you need to register your view to receive double-tap interactions.

SwiftUI UIKit

Add an `onPencilDoubleTap(perform:)` view modifier to your view.

```
MyView()
    .onPencilDoubleTap { value in
        // ...
    }
```

Check the preferred double-tap action

A person can choose which action they prefer to perform when they double-tap Apple Pencil. They choose this systemwide preference in Settings > Apple Pencil > Actions > Double Tap.

In your app, you can check the value of this preferred action for double tap.

SwiftUI UIKit

To check the preferred action, use the `preferredPencilDoubleTapAction` environment value. For possible values, see `PencilPreferredAction`.

```
@Environment(\.preferredPencilDoubleTapAction) private var preferredAction
```

Choose the action to perform

When possible, perform the preferred action to provide a consistent user experience across apps that support double taps. If the preferred action doesn't make sense in your app, consider giving people a way to choose a custom action that's suitable for your app. For design guidance, read Human Interface Guidelines > Apple Pencil and Scribble > Double tap.

The following code shows a snippet from a drawing app that provides custom drawing tools. This app allows a person to configure a custom action to quickly swap to their favorite custom drawing tool instead of using the systemwide preferred action for double taps. This app also supports the preferred actions to ignore double taps, switch to the previous tool, and switch to the eraser tool.

SwiftUI UIKit

```
enum Tool {
    case brush
    case lasso
    case eraser
    case magnifier
}
```

```
enum CustomAction: String {
    case switchLasso
    case switchMagnifier
}

@State private var currentTool: Tool? = .brush
@State private var previousTool: Tool?

@Environment(\.preferredPencilDoubleTapAction) private var preferredAction
@AppStorage("customPencilDoubleTapAction") private var customAction: CustomAction?

var body: some View {
    MyView()
        .onPencilDoubleTap { value in
            // Respect the systemwide preferred action to ignore double taps.
            guard preferredAction != .ignore else { return }

            // If the person chooses to override the systemwide
            // double-tap action to perform a custom action in this app,
            // check which custom action they prefer and perform that action.
            if let customAction {
                if customAction == .switchLasso, currentTool != .lasso {
                    (currentTool, previousTool) = (.lasso, currentTool)
                }
                else if customAction == .switchMagnifier, currentTool != .magnifier
                    (currentTool, previousTool) = (.magnifier, currentTool)
                }
            }

            // If the person prefers to use the systemwide double-tap action,
            // perform the actions that are appropriate in the context of this app:
            // switch to the previous tool, or switch to the eraser tool.
            else if preferredAction == .switchPrevious {
                (currentTool, previousTool) = (previousTool, currentTool)
            }
            else if preferredAction == .switchEraser, currentTool != .eraser {
                (currentTool, previousTool) = (.eraser, currentTool)
            }
        }
}
```

See Also

Related articles

Handling squeezes from Apple Pencil

Detect and respond to squeezes a person makes on Apple Pencil Pro.

Related reference in SwiftUI

```
nonisolated func onPencilDoubleTap(perform action: @escaping (PencilDoubleTapGestureValue) -> Void) -> some View
```

Adds an action to perform after the user double-taps their Apple Pencil.

```
struct PencilDoubleTapGestureValue
```

Describes the value of an Apple Pencil double-tap gesture.

```
struct PencilPreferredAction
```

An action that the user prefers to perform after double-tapping their Apple Pencil.

```
struct PencilHoverPose
```

A value describing the location and distance of an Apple Pencil hovering in the area above a view’s bounds.

Related reference in UIKit

```
@MainActor class UIPencilInteraction
```

An interaction that tells your app when a person double-taps or squeezes Apple Pencil.

```
@MainActor protocol UIPencilInteractionDelegate : NSObjectProtocol
```

The interface an object implements to handle double taps or squeezes a person makes on Apple Pencil.

```
@MainActor class Tap
```

An interaction that represents a double tap on Apple Pencil.

```
enum Phase
```

Constants that describe the phases of an interaction on Apple Pencil.

```
@MainActor class UIPencilHoverPose
```

An object that describes the hover pose of Apple Pencil during an interaction like double tap or squeeze.