

[UIKit](#) / [...](#) / [Handling touches in your view](#) / Getting high-fidelity input with coalesced touches

# Getting high-fidelity input with coalesced touches

Learn how to support high-precision touches in your app.

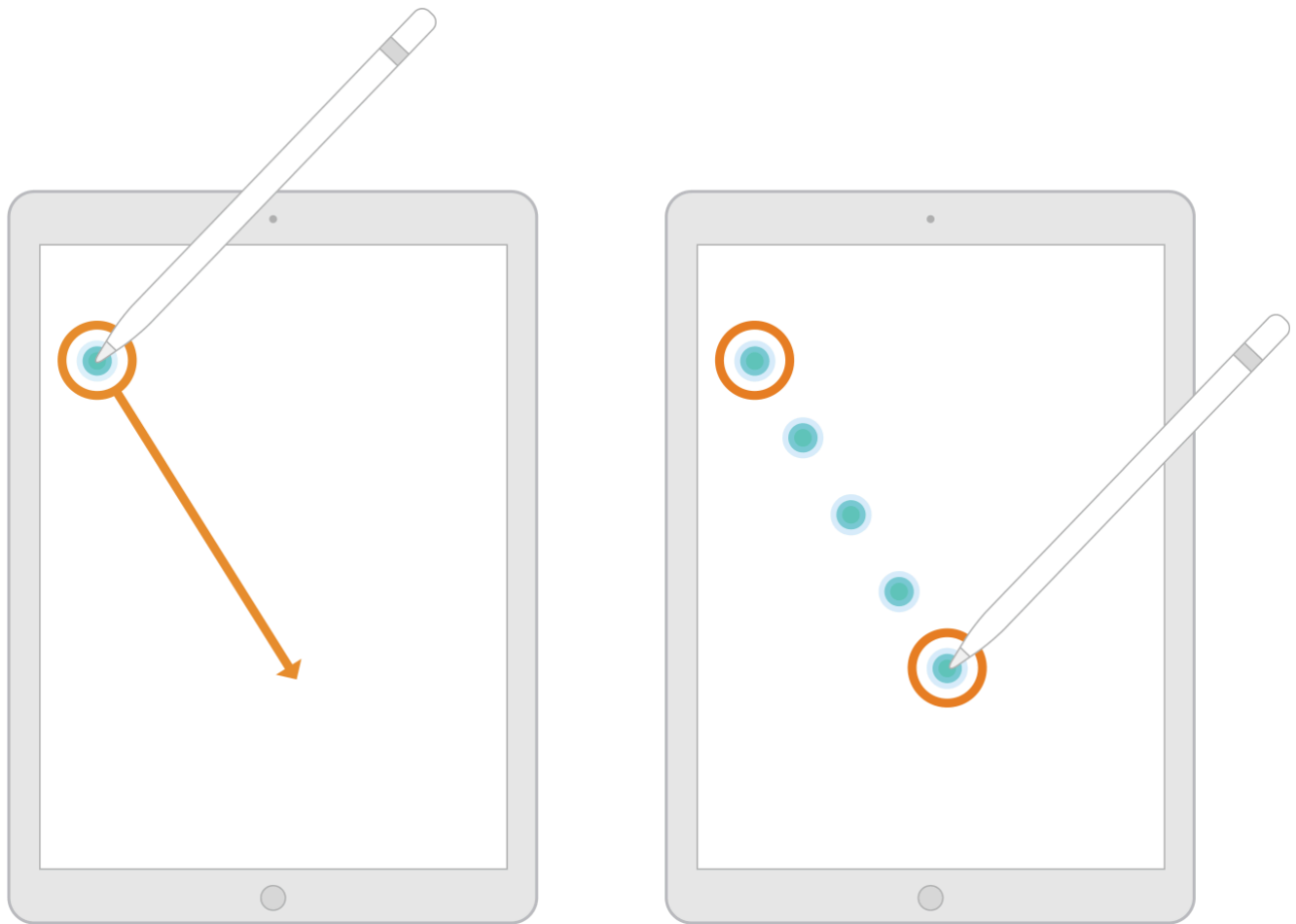
## Overview

UIKit usually delivers touches to your app at around 60 Hz, but some devices are capable of recording touch information at up to 240 Hz. On such devices, UIKit doesn't deliver the extra touch information automatically, in case the app doesn't need the extra data. Instead, it coalesces any extra touches into a single [UITouch](#) object, whose location reflects only the last recorded touch. However, apps that want the extra precision can retrieve and use the additional touch information.

### Important

Coalesced touches are intended for apps that need the extra precision and can handle the associated costs. Processing coalesced touches means gathering additional data and applying it to your content. If you don't need the extra precision, continue using the set of touch objects that UIKit passes to the methods of your views or gesture recognizers.

The following image illustrates what happens when the user drags Apple Pencil across the device. At the point where UIKit reports a touch event to the app, Apple Pencil has reported four touch positions, but UIKit reports only the last touch to the app by default. The remaining three touches are delivered as coalesced touches, and the app must retrieve them explicitly to use them.



To retrieve coalesced touches, call the `coalescedTouches(for:)` method of the `UIEvent` object containing the original `UITouch` object. That method returns the array of all touches since the last event, including the last `UITouch` object actually delivered to the app. You must retrieve coalesced touches immediately when handling an event. After handling the event, there's no guarantee that any coalesced touches will remain available.

---

## Topics

### Example

 Implementing coalesced touch support in an app

Learn how to create a simple app that handles coalesced touches.

---

## See Also

**Advanced touch handling**



## Implementing a Multi-Touch app

Learn how to create a simple app that handles multitouch input.



## Minimizing latency with predicted touches

Create a smooth and responsive drawing experience using UIKit's predictions for touch locations.