

[SwiftUI](#) / [SpatialEventGesture](#)

Structure

SpatialEventGesture

A gesture that provides information about ongoing spatial events like clicks and touches.

iOS 18.0+ | iPadOS 18.0+ | Mac Catalyst 18.0+ | macOS 15.0+ | visionOS 1.0+ | watchOS 11.0+

```
struct SpatialEventGesture
```

Overview

Use a gesture of this type to track multiple simultaneous spatial events and gain access to detailed information about each. For example, you can place a particle emitter at every location in a [Canvas](#) that has an ongoing spatial event:

```
struct ParticlePlayground: View {
    @State var model = ParticlesModel()

    var body: some View {
        Canvas { context, size in
            for particle in model.particles {
                context.fill(Path(ellipseIn: particle.frame),
                    with: .color(particle.color))
            }
        }
        .gesture(
            SpatialEventGesture()
                .onChanged { events in
                    for event in events {
                        if event.phase == .active {
```

```

        // Update particle emitters.
        model.emitters[event.id] = ParticlesModel.Emitter(
            location: event.location
        )
    } else {
        // Remove emitters when no longer active.
        model.emitters[event.id] = nil
    }
}
}
.onEnded { events in
    for event in events {
        // Remove emitters when no longer active.
        model.emitters[event.id] = nil
    }
}
}
)
}

```

The gesture provides a [SpatialEventCollection](#) structure when it detects changes. The collection contains [SpatialEventCollection.Event](#) values that represent ongoing spatial events. Each event contains a stable, unique identifier so that you can track how the event changes over time. The event also indicates its current location, a timestamp, the pose of the input device that creates it, and other useful information.

The phase of events in the collection can change to [SpatialEventCollection.Event.Phase.ended](#) or [SpatialEventCollection.Event.Phase.cancelled](#) while the gesture itself remains active. Individually track state for each event inside [onChanged\(_:\)](#) or [updating\(_:body:\)](#) and clean up all state in [onEnded\(_:\)](#).

Tip

Only use a spatial event gesture if you need to access low-level event information, like when you create a complex multi-touch experience. For most use cases, it's better to rely on gestures that recognize targeted interactions, like a [SpatialTapGesture](#), [MagnifyGesture](#), or [DragGesture](#).

Topics

Creating a spatial event gesture

```
init(coordinateSpace: any CoordinateSpaceProtocol)
```

Creates the gesture with a desired coordinate space.

Getting gesture properties

```
let coordinateSpace: CoordinateSpace
```

The coordinate space of the gesture.

Initializers

```
init(coordinateSpace3D: some CoordinateSpace3D)
```

Creates the gesture with a desired coordinate space 3D.

Relationships

Conforms To

Gesture

See Also

Recognizing spatial events

```
struct SpatialEventCollection
```

A collection of spatial input events that target a specific view.

```
enum Chirality
```

The chirality, or handedness, of a pose.