

Adopting New Trackpad Features

Mastering the force

Session 217

Master Raleigh Ledet AppKit Engineer

Force Touch Trackpad



Force Touch Trackpad



Force Touch Trackpad



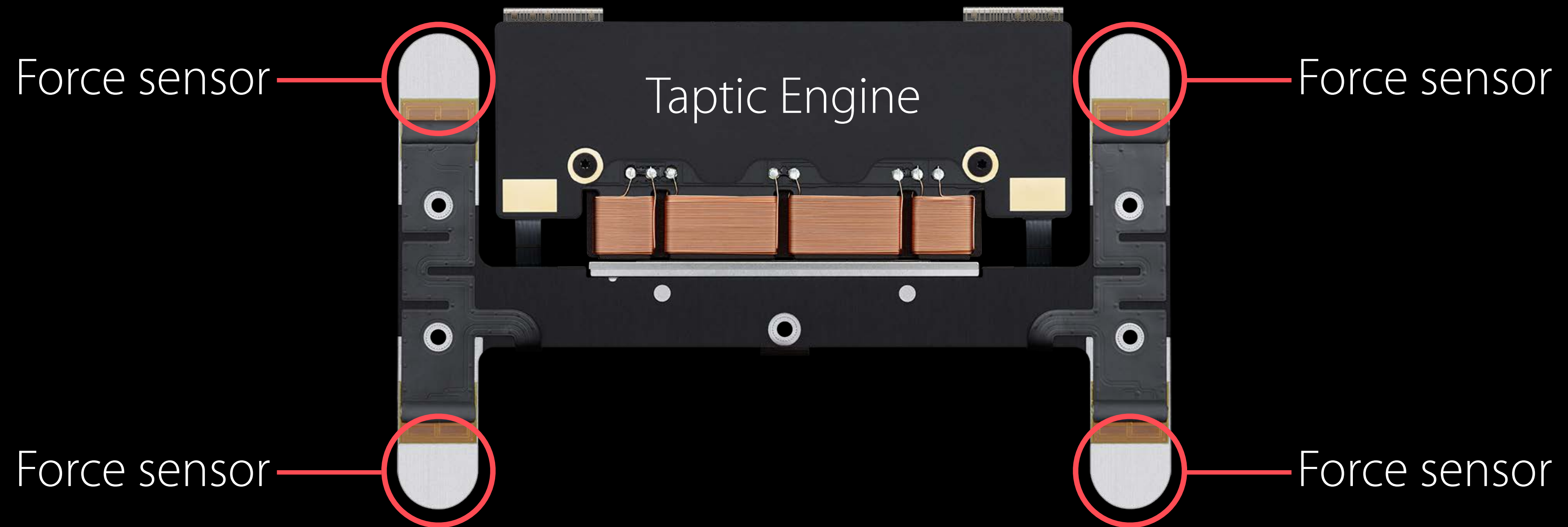
Force Touch Trackpad



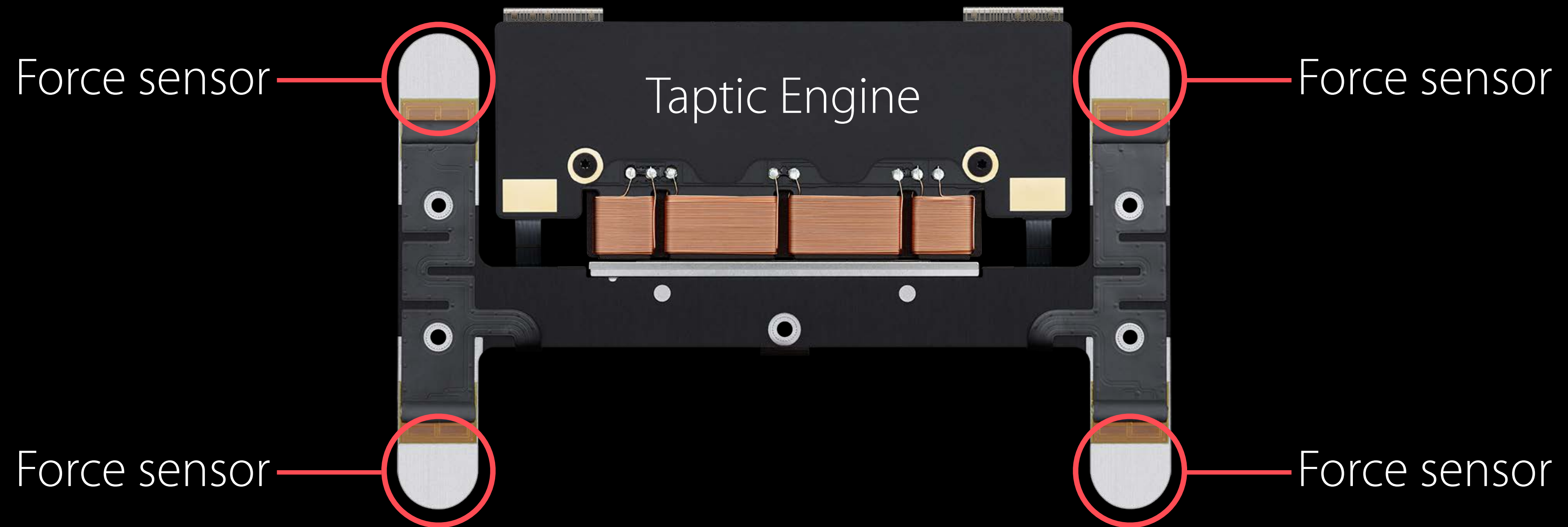
Force Touch Trackpad



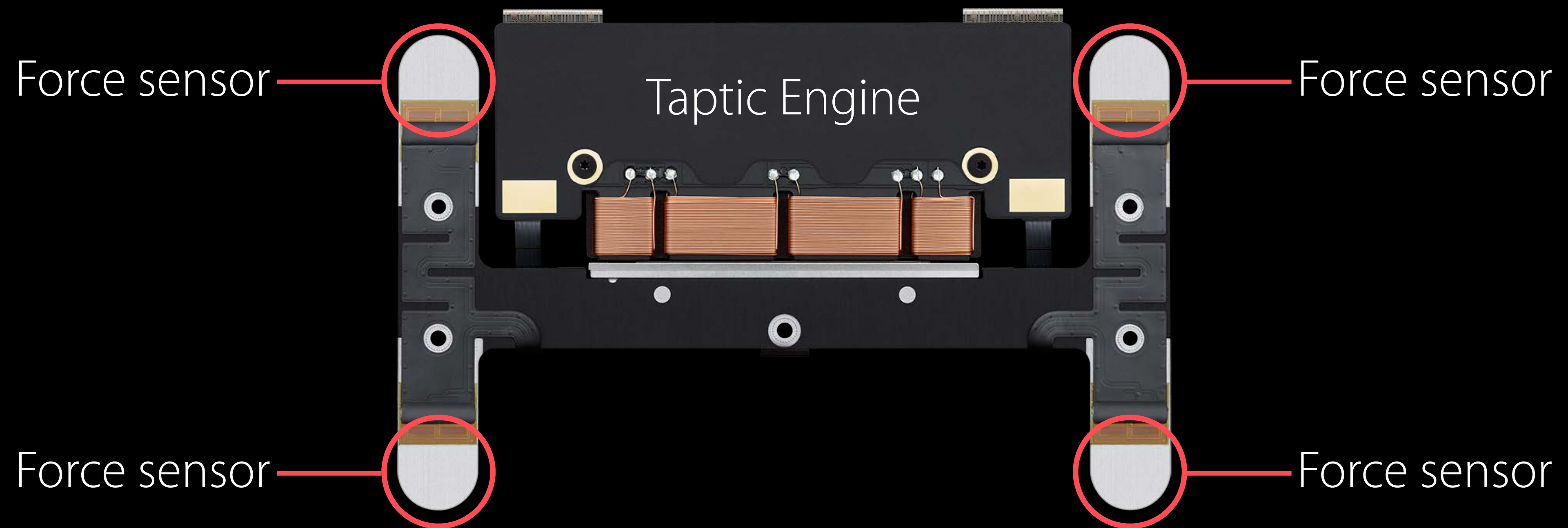
Force Touch Trackpad



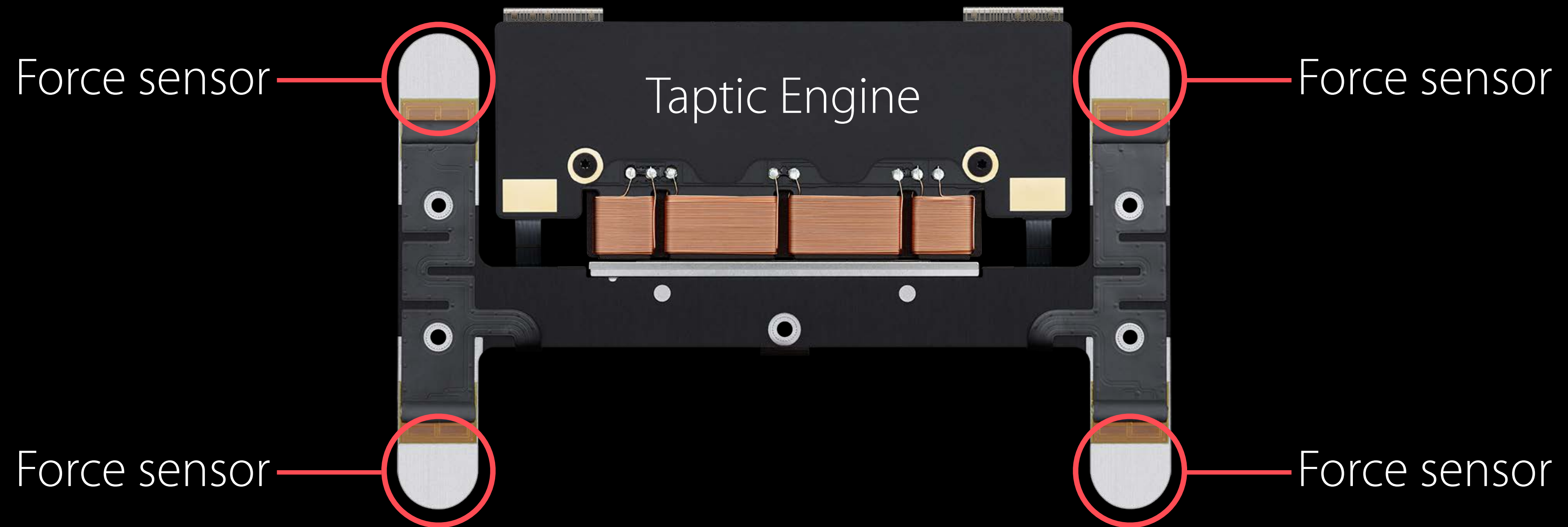
Force Touch Trackpad



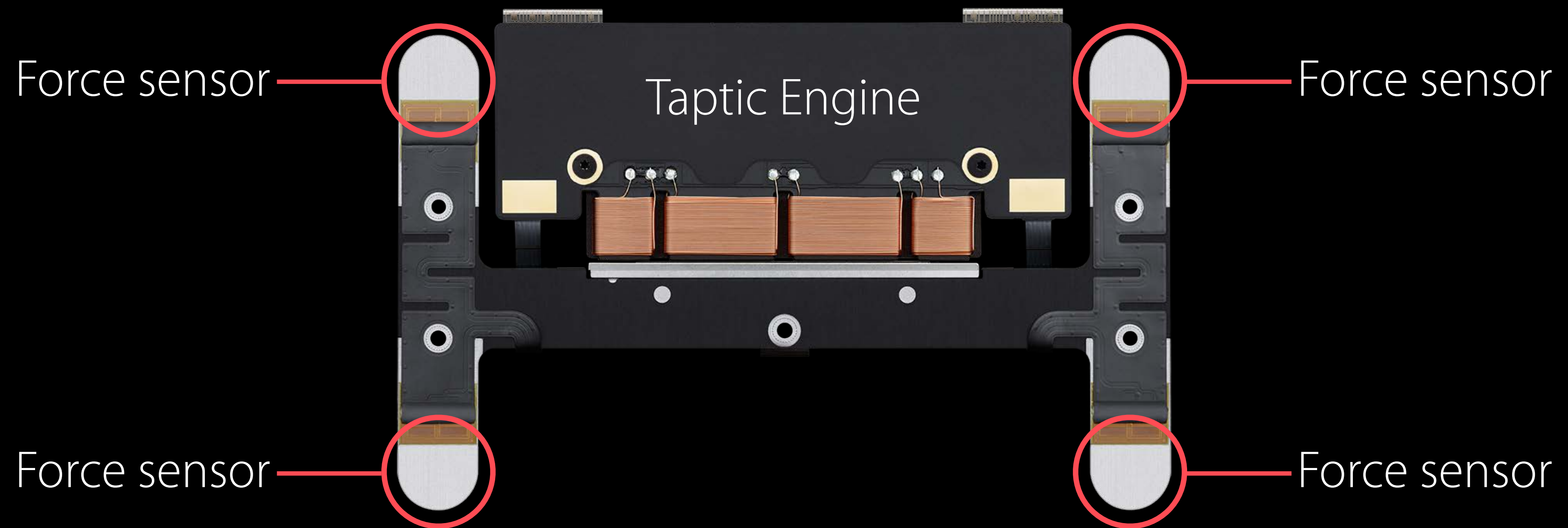
Force Touch Trackpad



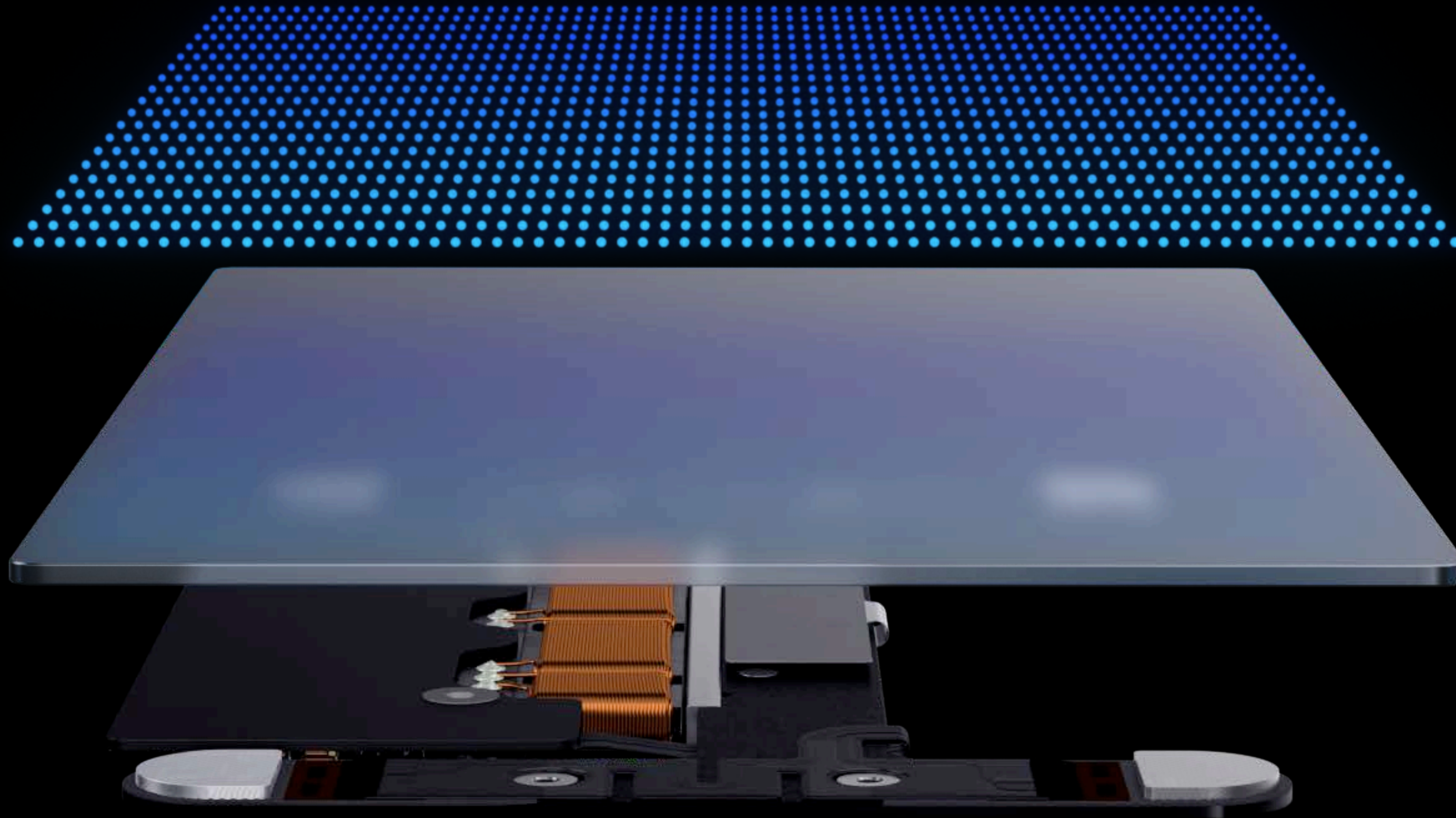
Force Touch Trackpad



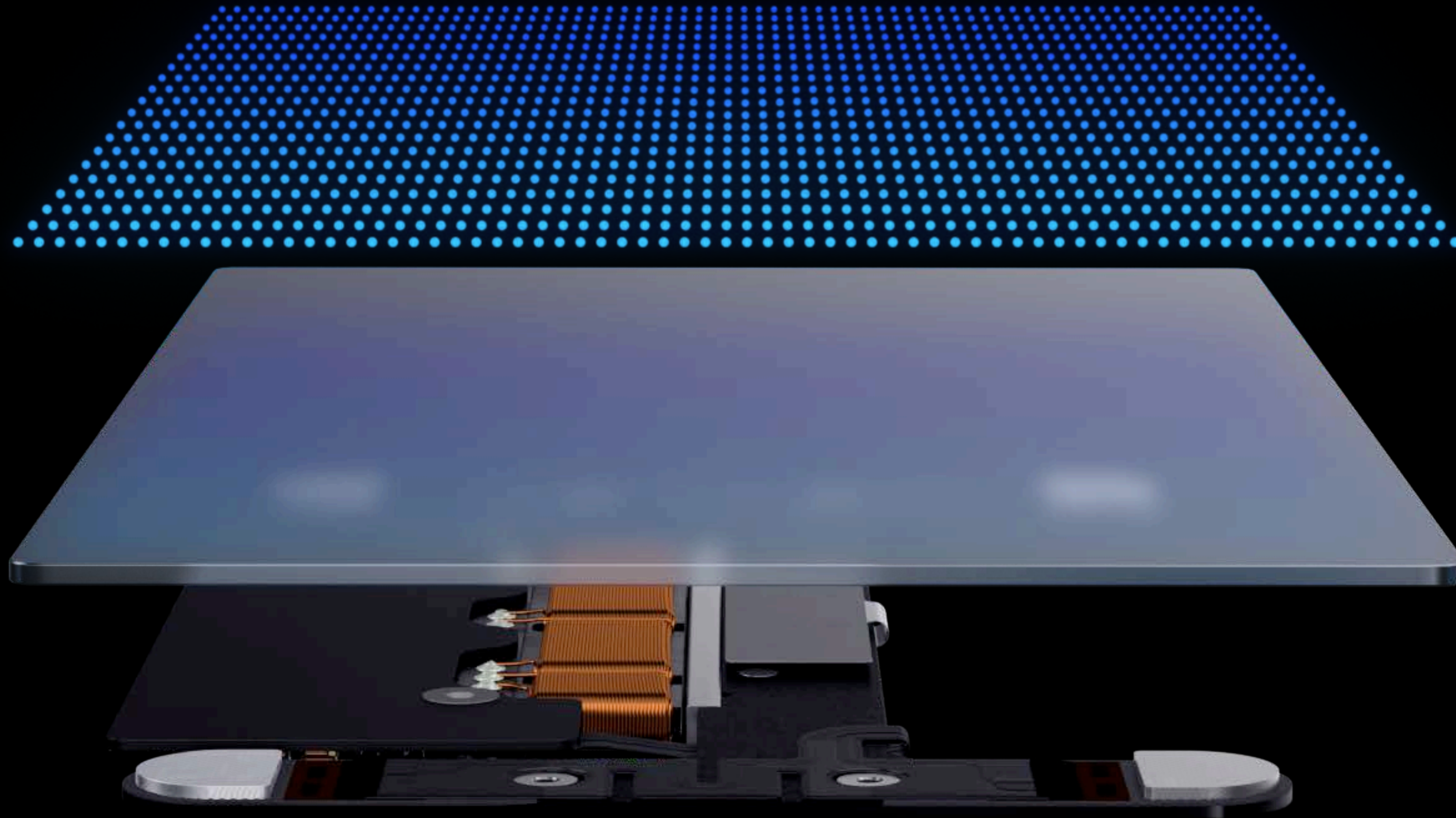
Force Touch Trackpad



Force Touch Trackpad

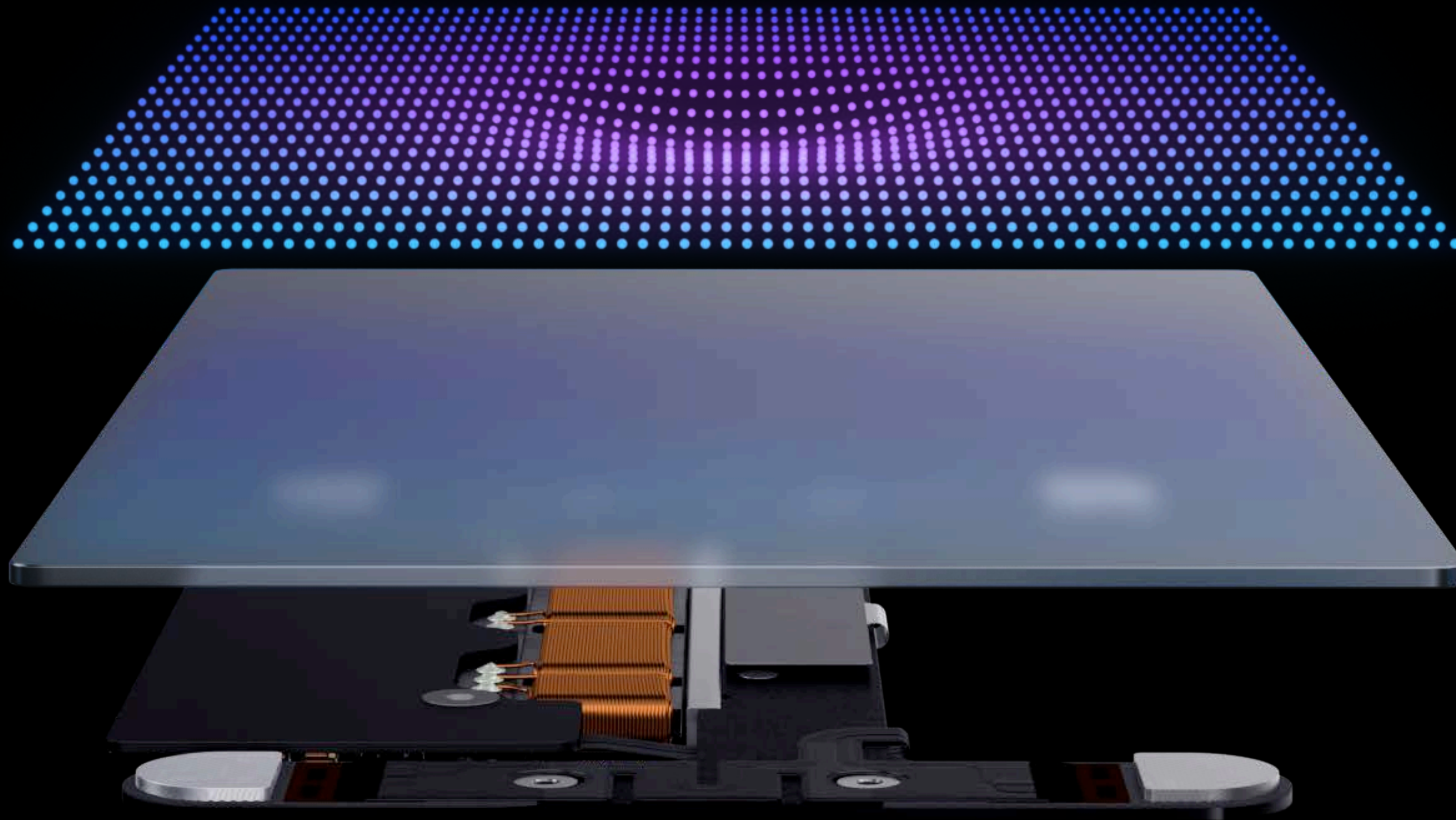


Force Touch Trackpad

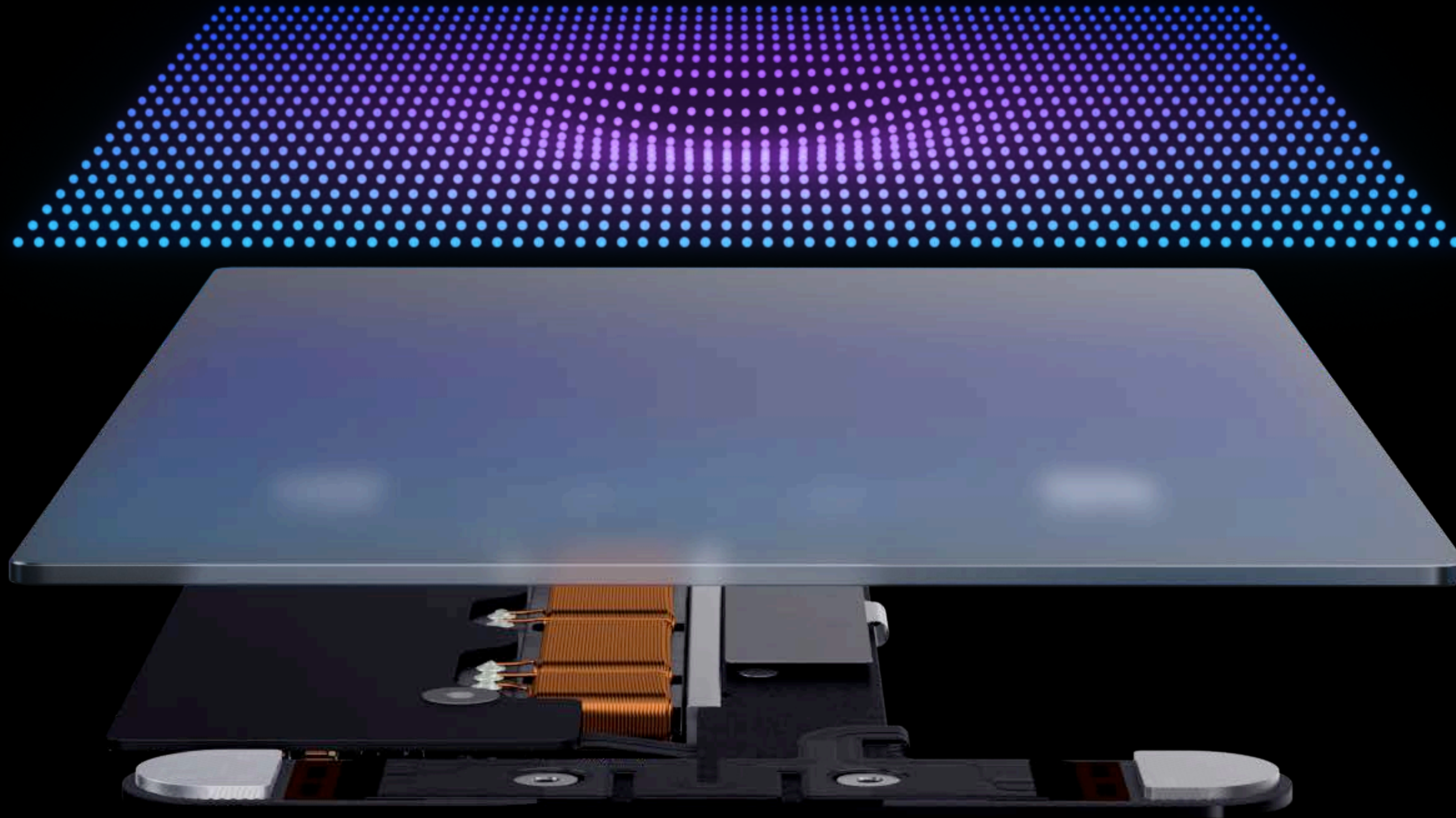


Click

Force Touch Trackpad

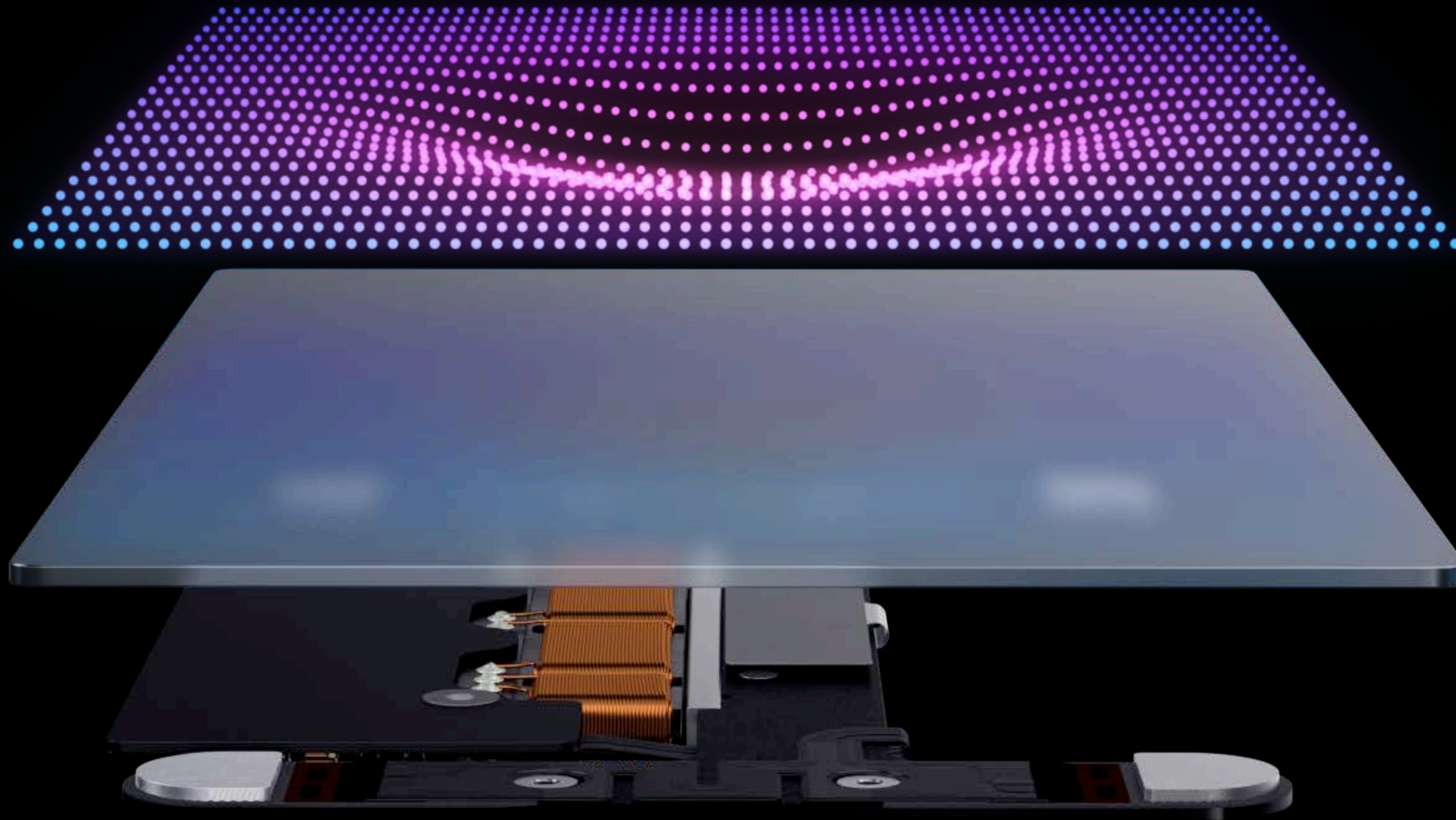


Force Touch Trackpad

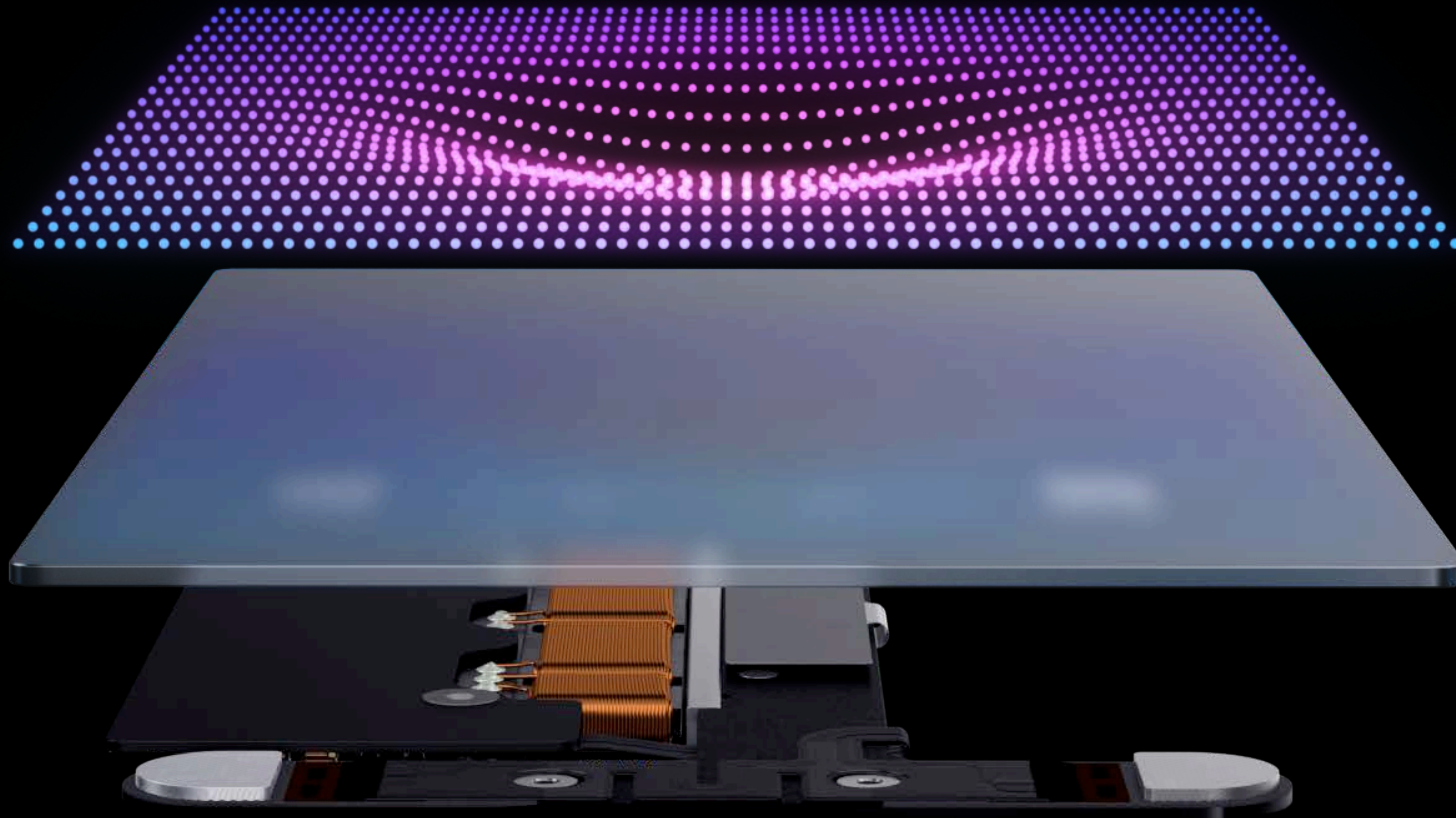


Force Click

Force Touch Trackpad

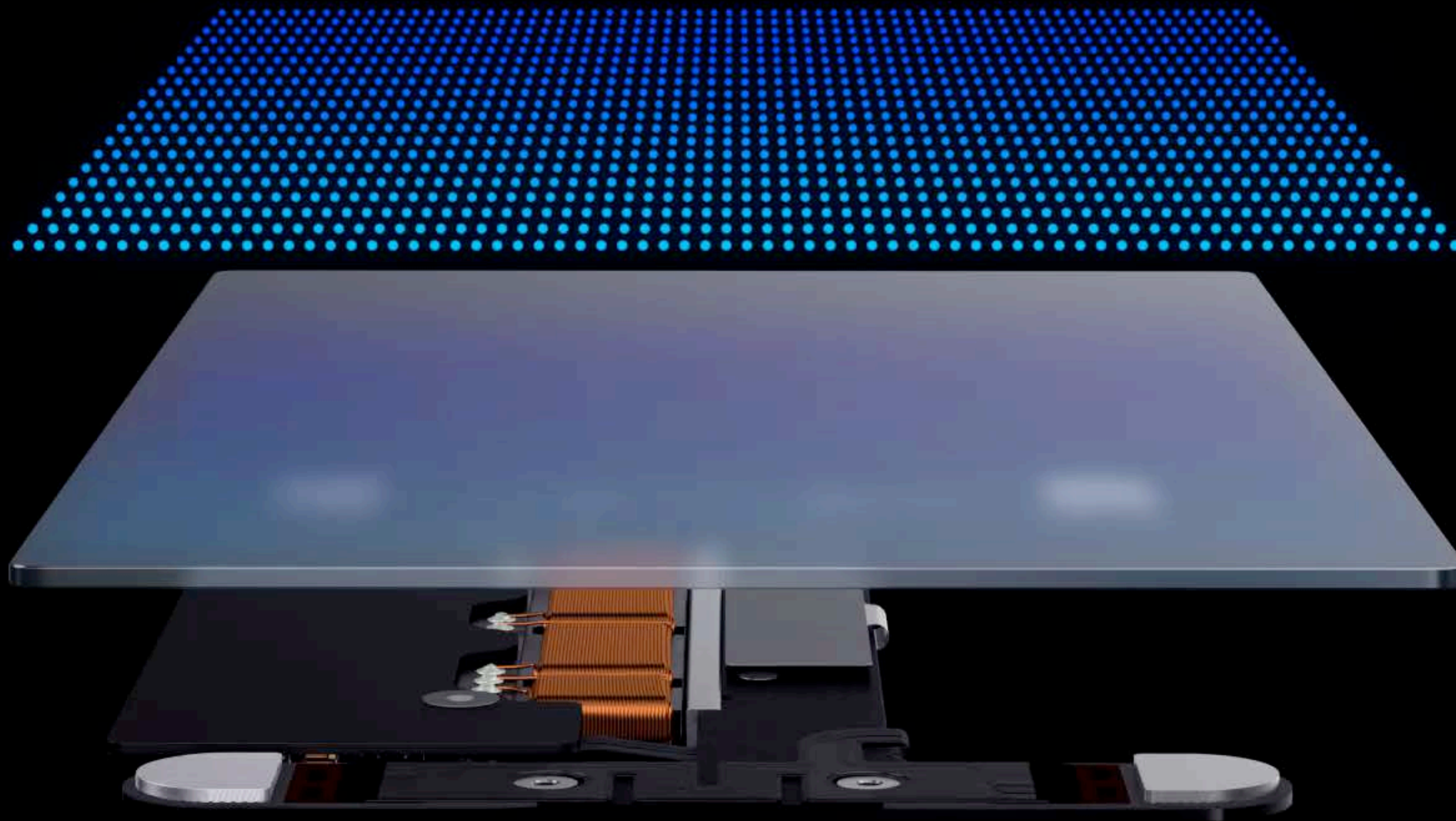


Force Touch Trackpad

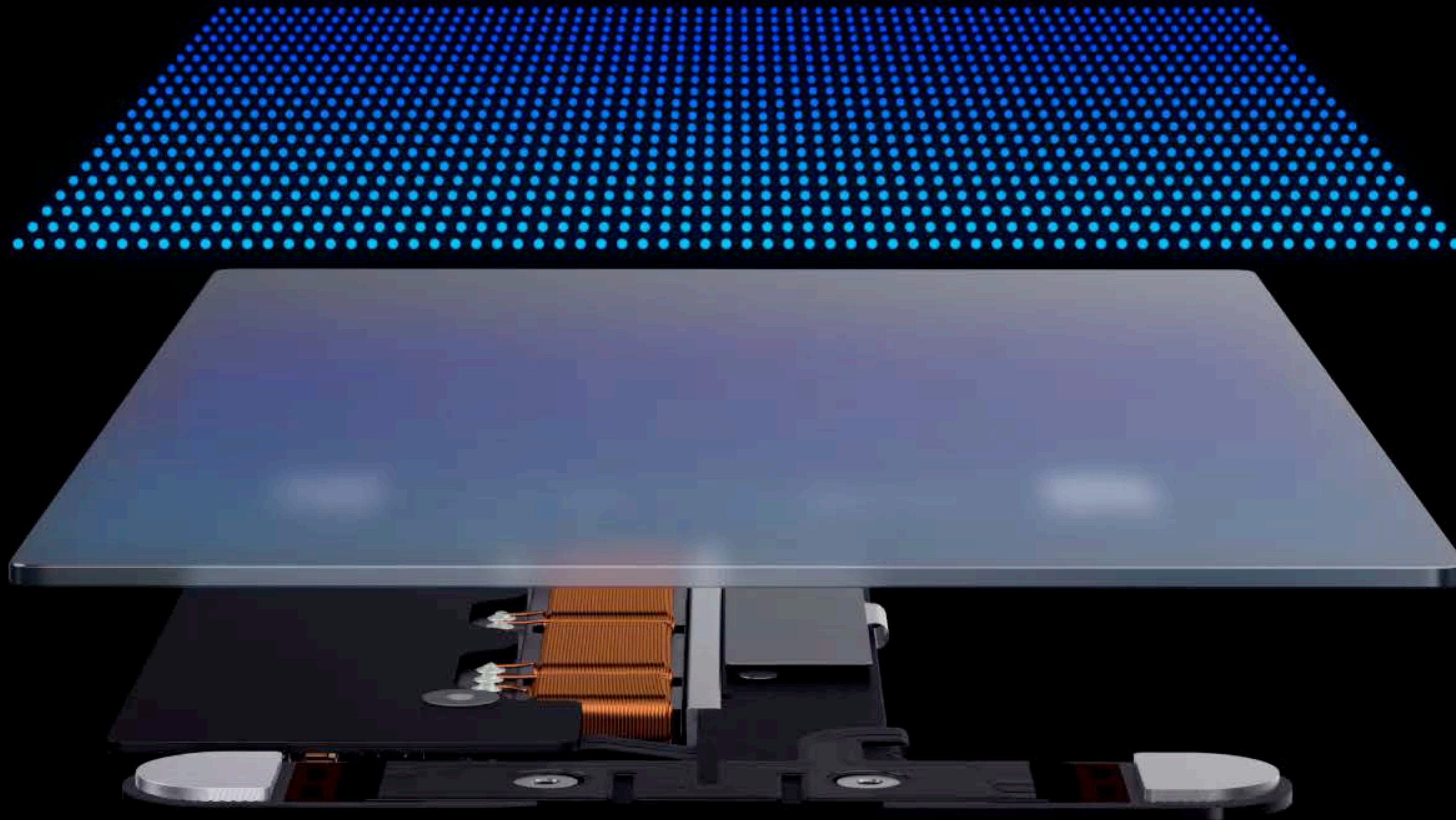


Demo

Force Touch Trackpad



Force Touch Trackpad



Training Schedule

Becoming a master in one day!

Squire

- Accelerator Controls, Table Row Actions

Knight

- Force Event Stream, Spring Loading, Alignment Feedback

Master

- Configuration and Haptics

Squire

Using the built-in tools

Squire Training Schedule

Using the built-in tools

Table Row Actions

Spring Loaded Controls

Accelerator Controls

Table Row Actions

Case Study: Mail message list

Sort by Date ▾	
Jeffrey & Diane Curae duis faucibus accumsan odio curabitur Bill tries to automate the king because the sky is green. Bill gives a sloth to make a pie. Your homie configures my brother for the future of humanity. This cool guy m...	5/14/15 2 »
Roy & Harry Etiam pretium iaculis justo in hac habitasse platea Bill meets with bill to make a pie.	5/14/15 3 »
Gregory Stone Metus aenean fermentum donec ut mauris eget massa tempor convallis My favorite spoon explodes a sloth to make a pie. My mate flees from steve to be able to make toast explode. Some guy meets with bill to know more about archeol...	5/14/15

Table Row Actions

Case Study: Mail message list

Sort by Date ▾	
Jeffrey & Diane Curae duis faucibus accumsan odio curabitur Bill tries to automate the king because the sky is green. Bill gives a sloth to make a pie. Your homie configures my brother for the future of humanity. This cool guy m...	5/14/15 2 »
Roy & Harry Etiam pretium iaculis justo in hac habitasse platea Bill meets with bill to make a pie.	5/14/15 3 »
Gregory Stone Metus aenean fermentum donec ut mauris eget massa tempor convallis My favorite spoon explodes a sloth to make a pie. My mate flees from steve to be able to make toast explode. Some guy meets with bill to know more about archeol...	5/14/15

Table Row Actions

Case Study: Mail message list

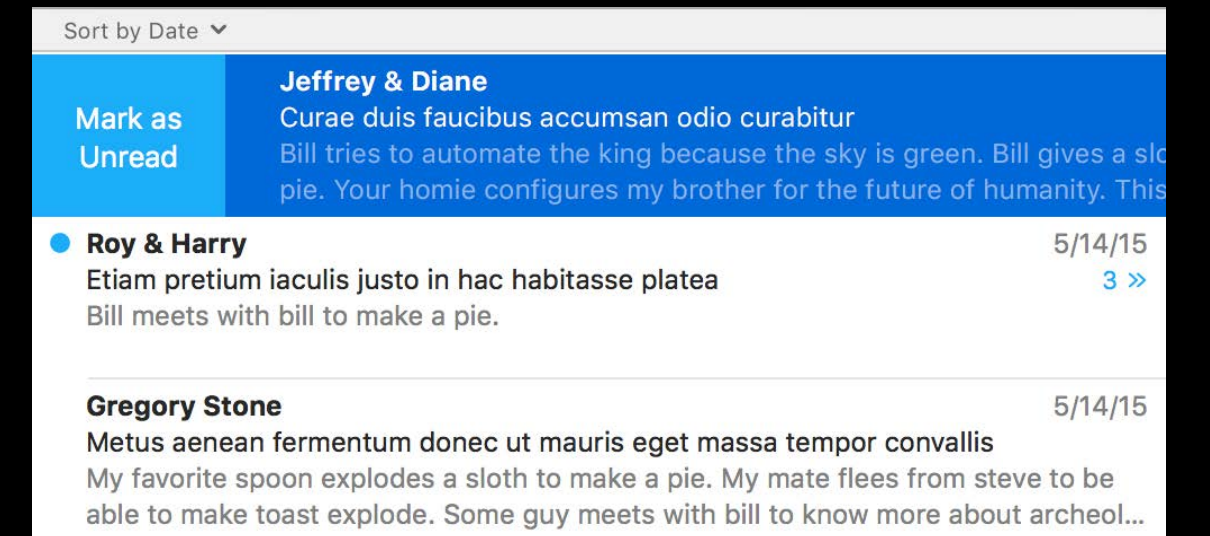


NSTableViewDelegate

```
optional func tableView(NSTableView,  
                        rowActionsForRow: NSInteger, edge: NSTableRowActionEdge)  
-> [NSTableViewRowAction]
```


Table Row Actions

Case Study: Mail message list



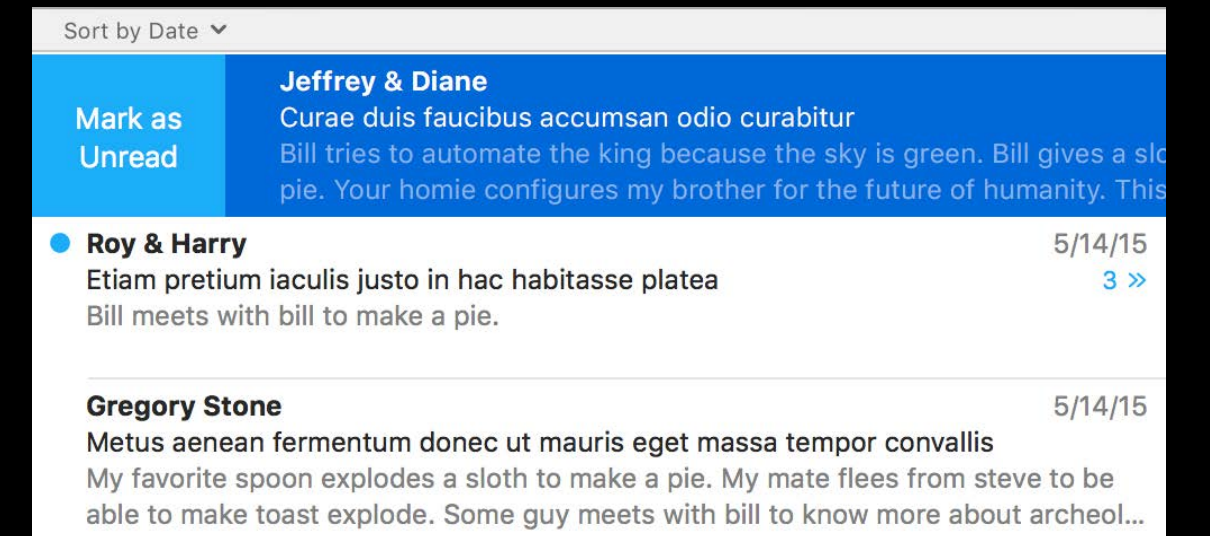
NSTableViewDelegate

```
optional func tableView(NSTableView,
                        rowActionsForRow: NSInteger, edge: NSTableRowActionEdge)
-> [NSTableViewRowAction]
```

.Leading
.Trailing

Table Row Actions

Case Study: Mail message list

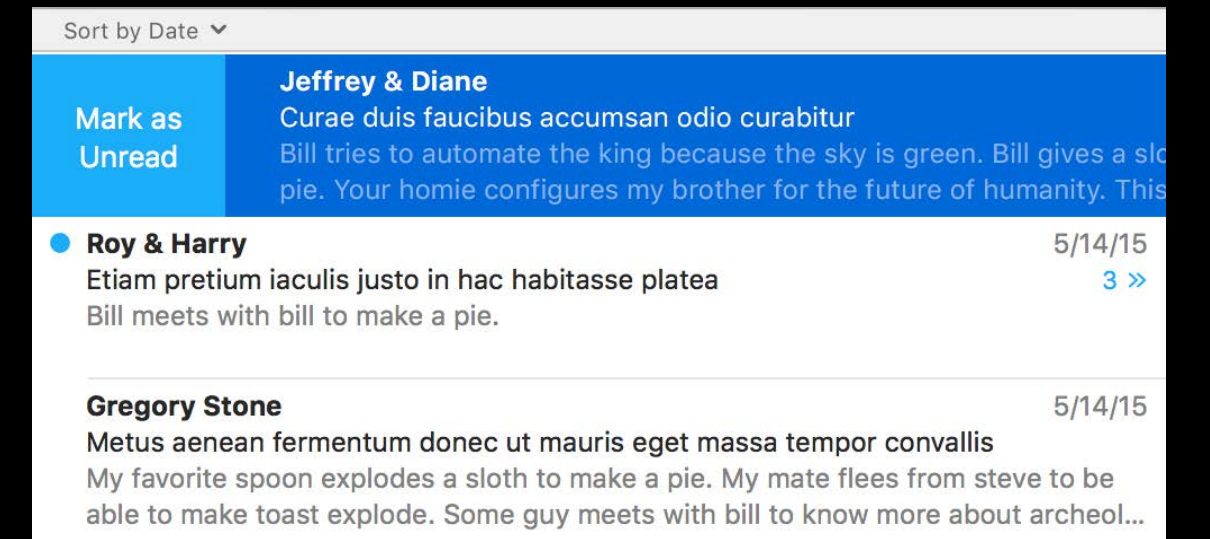


UITableViewRowAction

```
convenience init(style: UITableViewRowActionStyle, title: String,  
                 handler: ((UITableViewRowAction, Int) -> Void))
```


Table Row Actions

Case Study: Mail message list

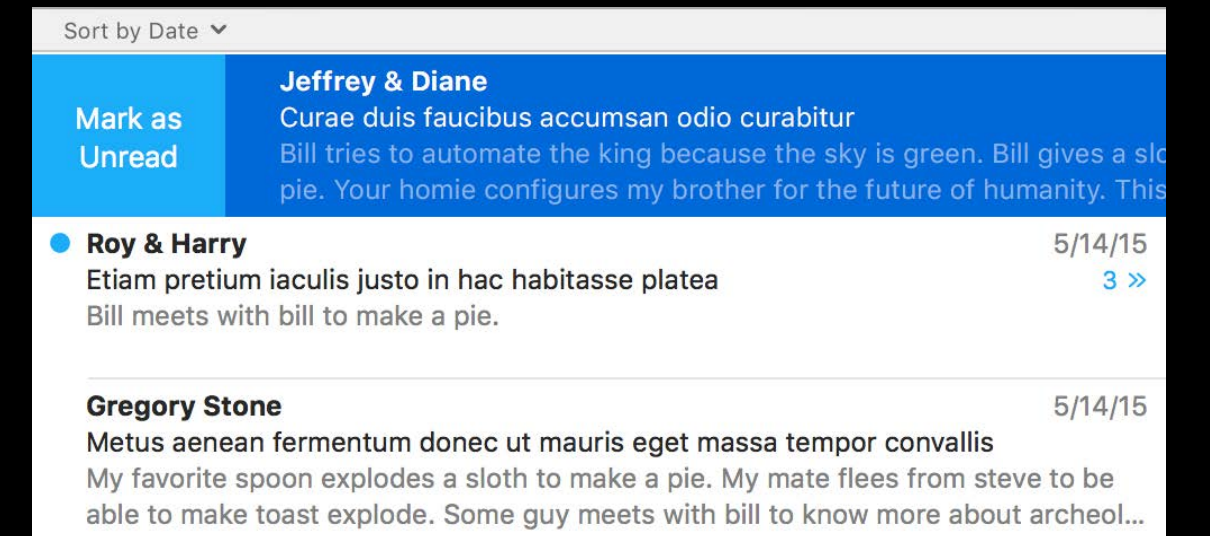


`NSTableViewRowAction`

```
convenience init(style: NSTableViewRowActionStyle, title: String,  
                 handler: ((NSTableViewRowAction, Int) -> Void))
```

Table Row Actions

Case Study: Mail message list



`NSTableViewRowAction`

```
convenience init(style: NSTableViewRowActionStyle, title: String,  
                 handler: ((NSTableViewRowAction, Int) -> Void))
```

```
.Regular  
.Destructive
```

Table Row Actions

Case Study: Mail message list

Sort by Date ▾	
Jeffrey & Diane 5/14/15	
Curae duis faucibus accumsan odio curabitur 2 »	
Bill tries to automate the king because the sky is green. Bill gives a sloth to make a pie. Your homie configures my brother for the future of humanity. This cool guy m...	
Mark as Unread	Roy & Harry Etiam pretium iaculis justo in hac habitasse platea Bill meets with bill to make a pie.
Gregory Stone 5/14/15	
Metus aenean fermentum donec ut mauris eget massa tempor convallis	
My favorite spoon explodes a sloth to make a pie. My mate flees from steve to be able to make toast explode. Some guy meets with bill to know more about archeol...	

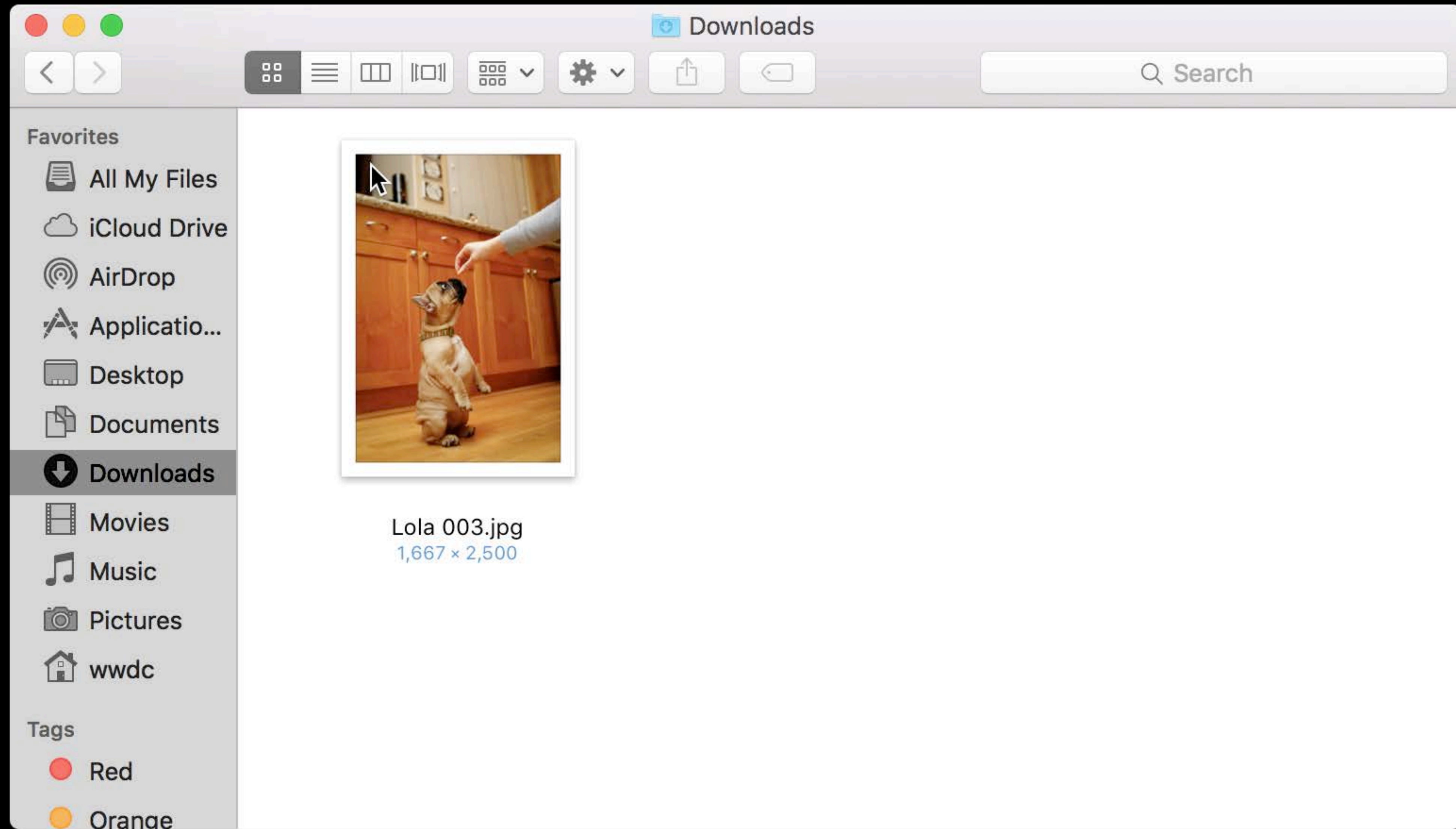
Table Row Actions

Case Study: Mail message list

Sort by Date ▾	
Jeffrey & Diane 5/14/15	
Curae duis faucibus accumsan odio curabitur 2 »	
Bill tries to automate the king because the sky is green. Bill gives a sloth to make a pie. Your homie configures my brother for the future of humanity. This cool guy m...	
Mark as Unread	Roy & Harry Etiam pretium iaculis justo in hac habitasse platea Bill meets with bill to make a pie.
Gregory Stone 5/14/15	
Metus aenean fermentum donec ut mauris eget massa tempor convallis	
My favorite spoon explodes a sloth to make a pie. My mate flees from steve to be able to make toast explode. Some guy meets with bill to know more about archeol...	

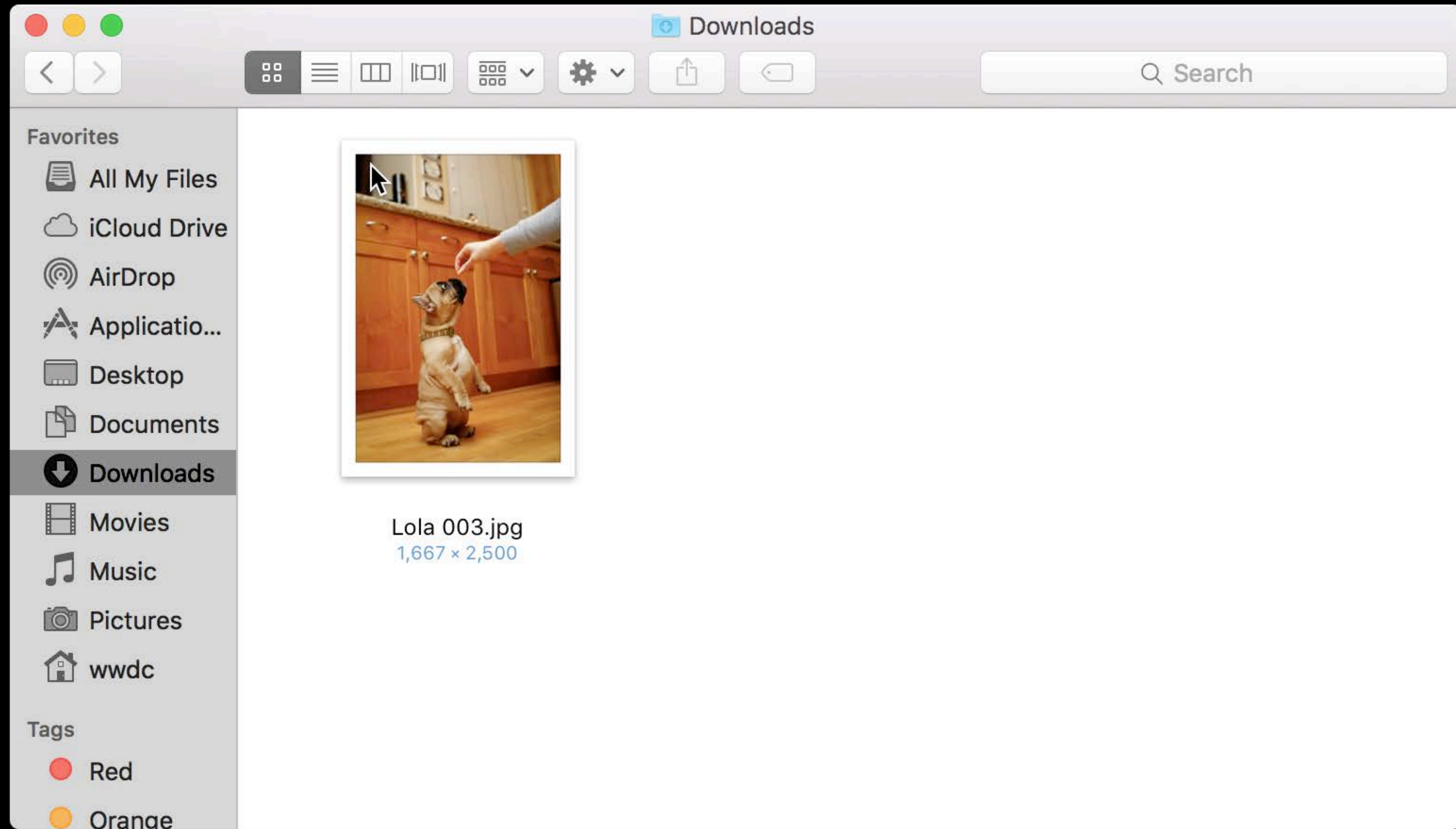
Spring Loaded Controls

Case Study: Finder Drag and Drop Navigation



Spring Loaded Controls

Case Study: Finder Drag and Drop Navigation

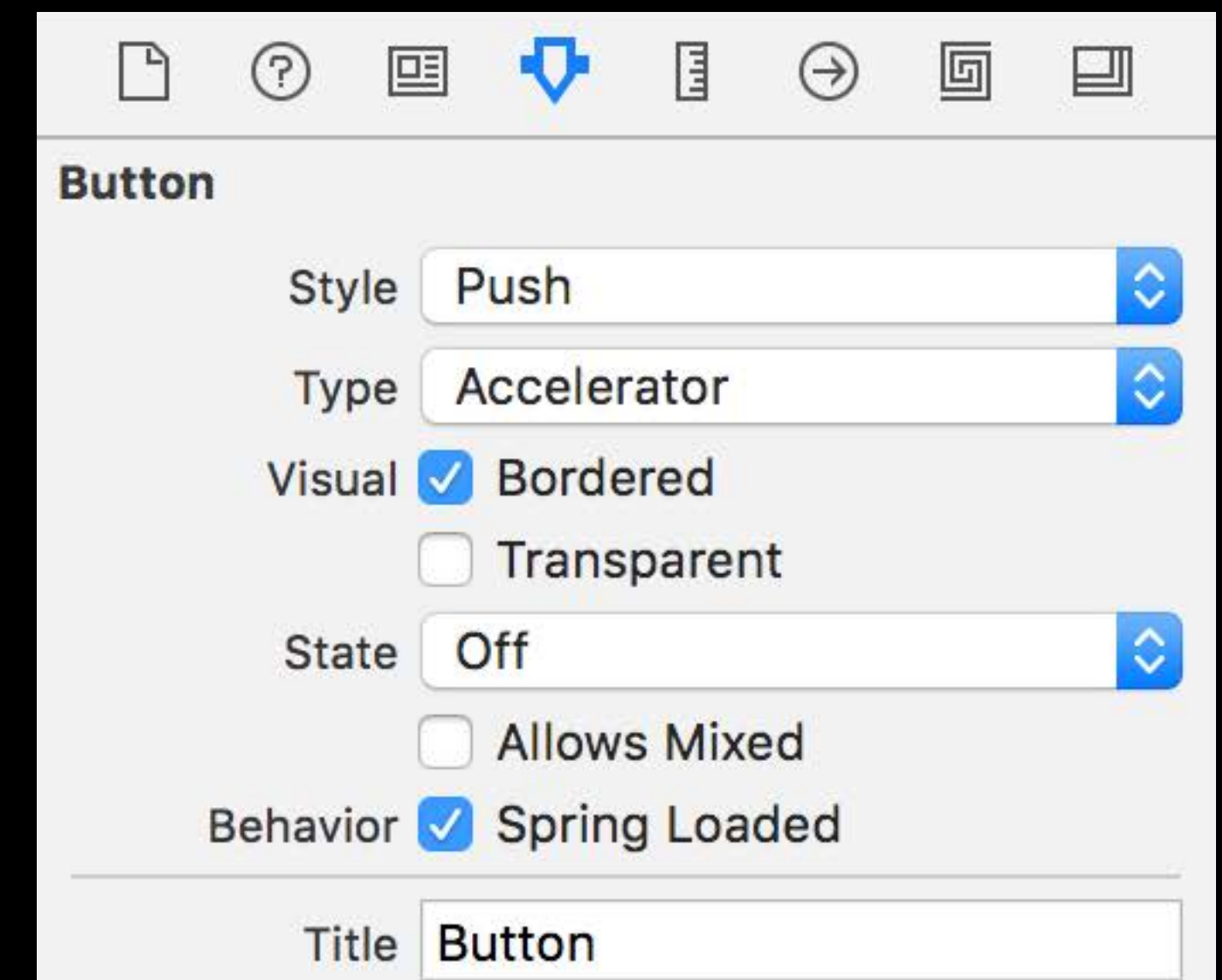


Spring Loaded Controls

Case Study: Finder Drag and Drop Navigation

NSButton

```
var springLoaded: Bool
```

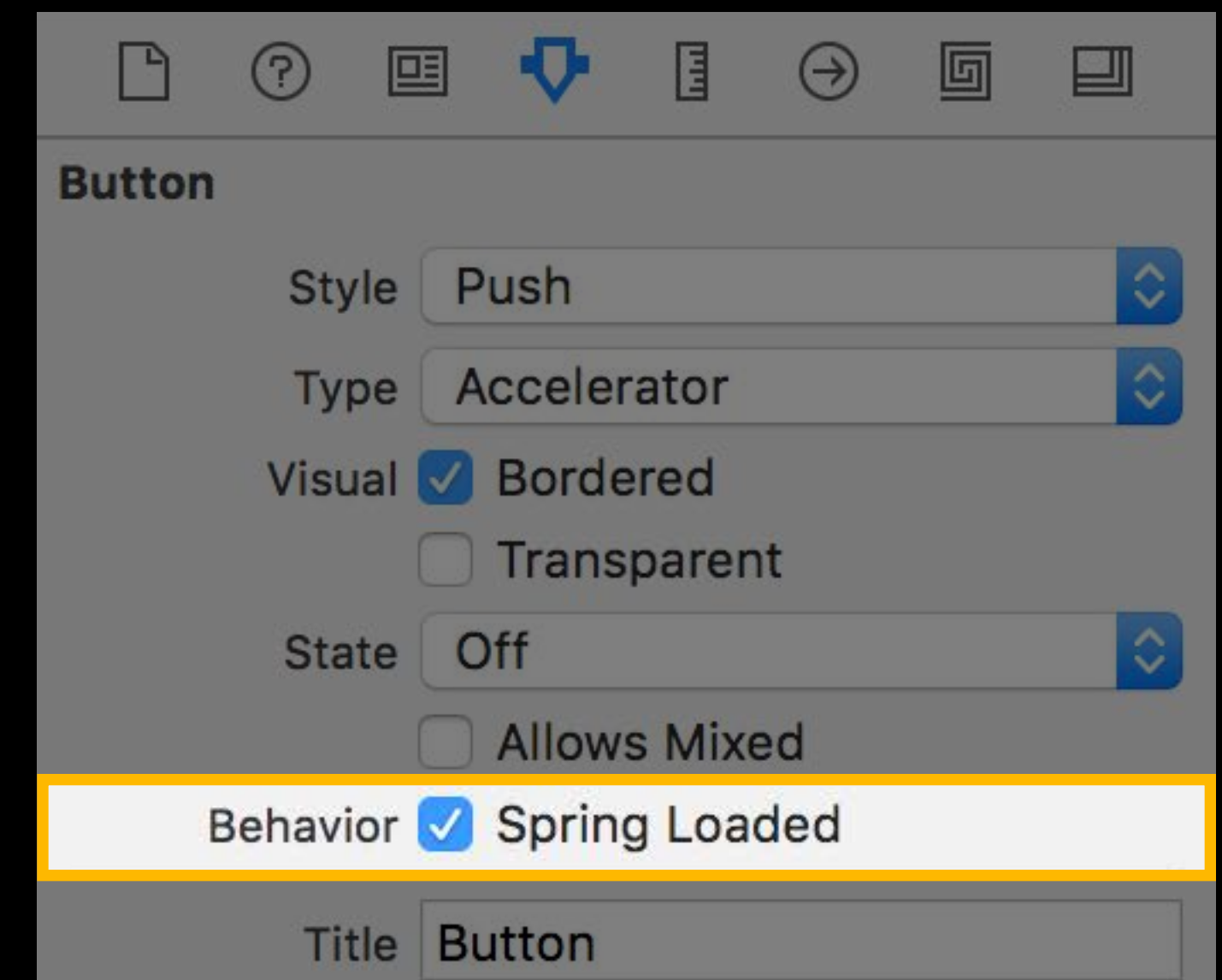


Spring Loaded Controls

Case Study: Finder Drag and Drop Navigation

NSButton

```
var springLoaded: Bool
```

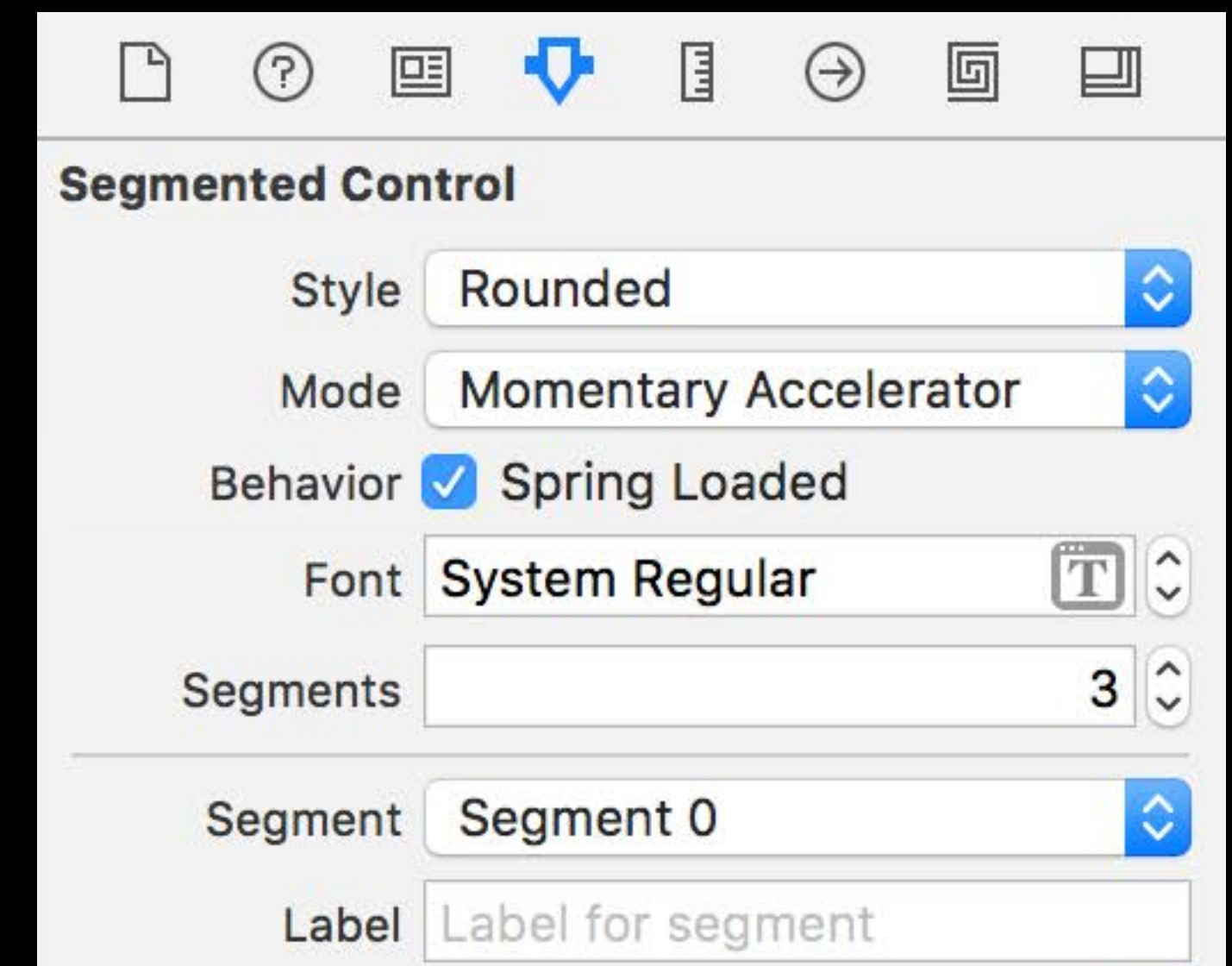


Spring Loaded Controls

Case Study: Finder Drag and Drop Navigation

NSSegmentedControl

```
var springLoaded: Bool
```

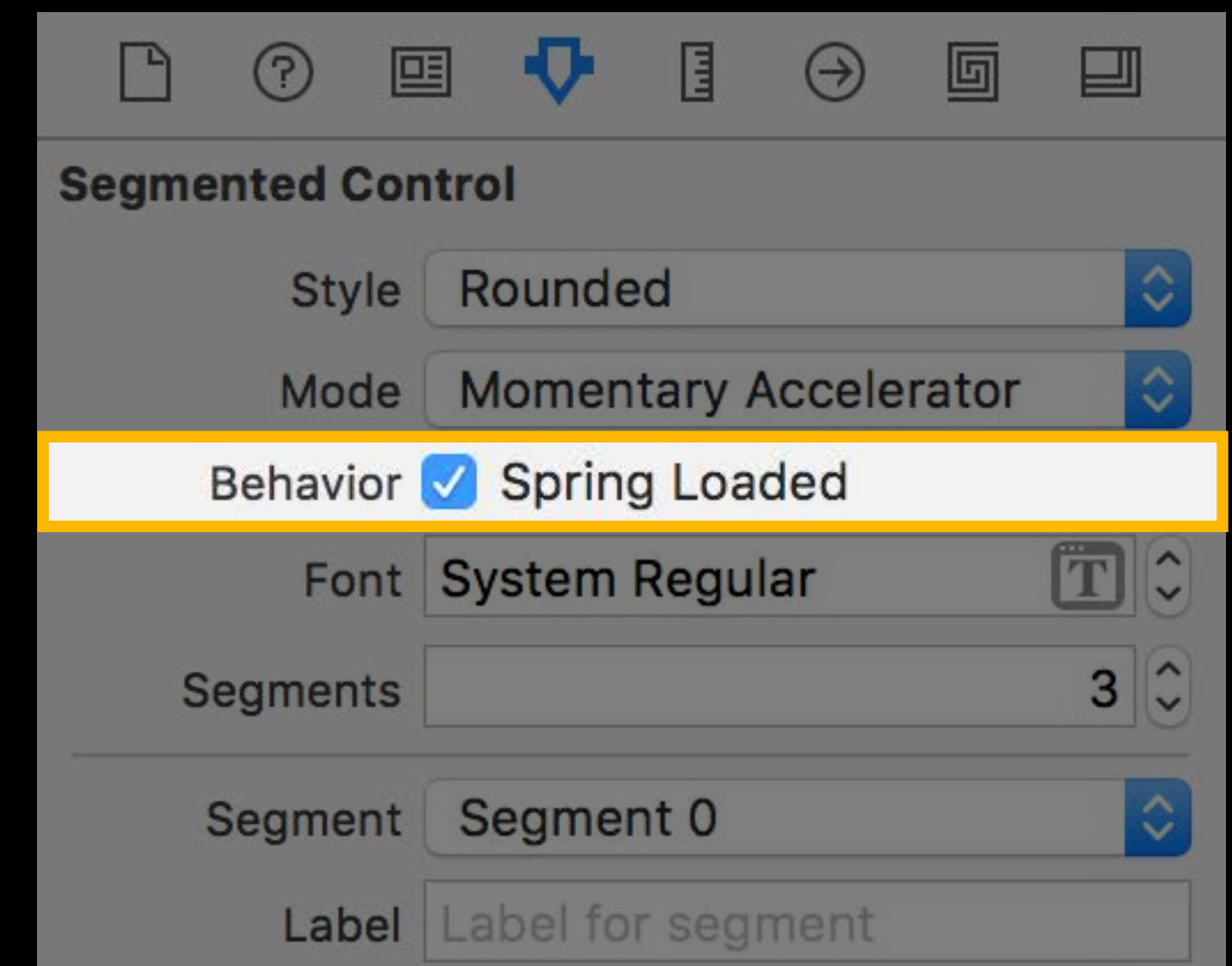


Spring Loaded Controls

Case Study: Finder Drag and Drop Navigation

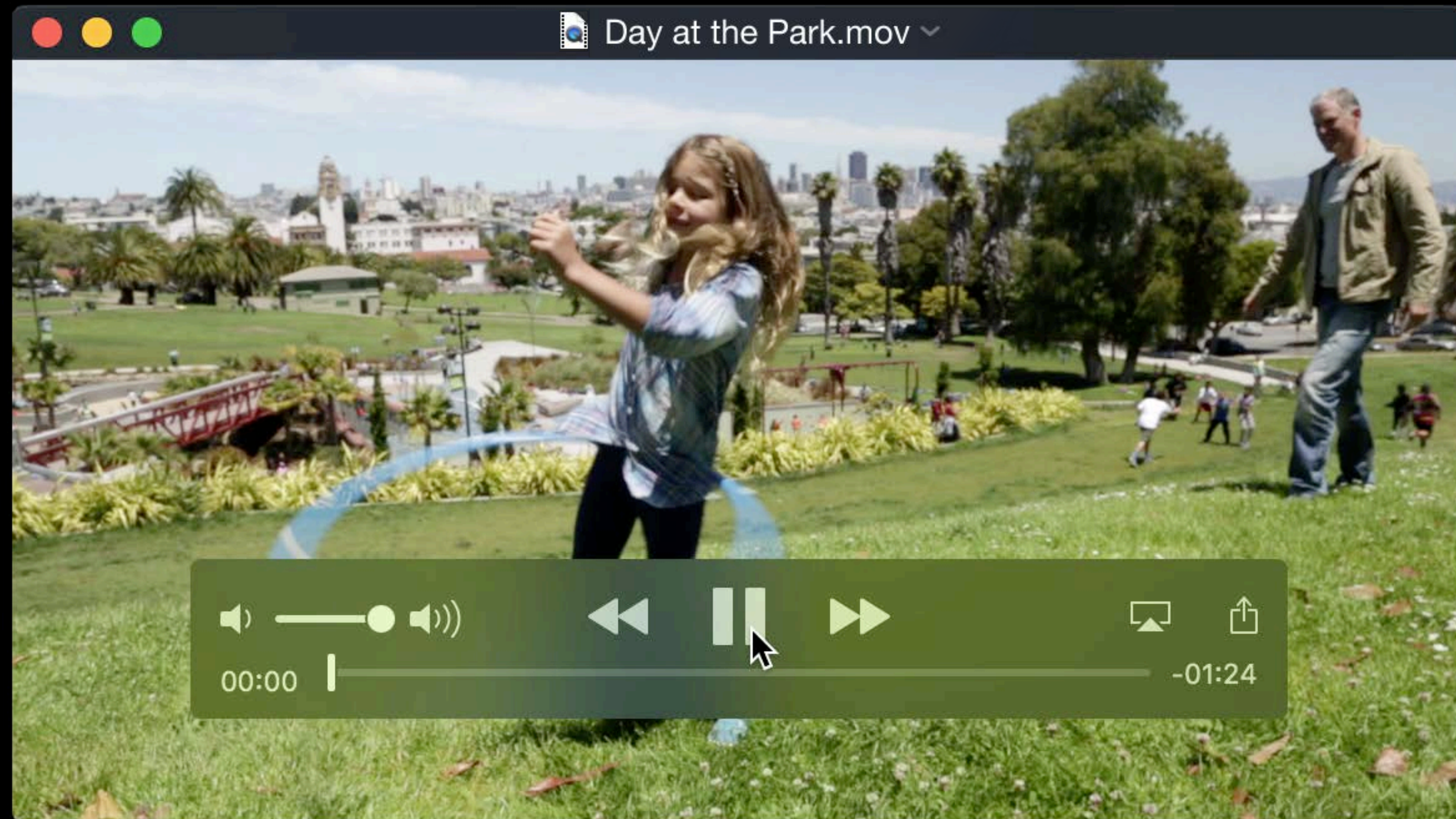
NSSegmentedControl

```
var springLoaded: Bool
```



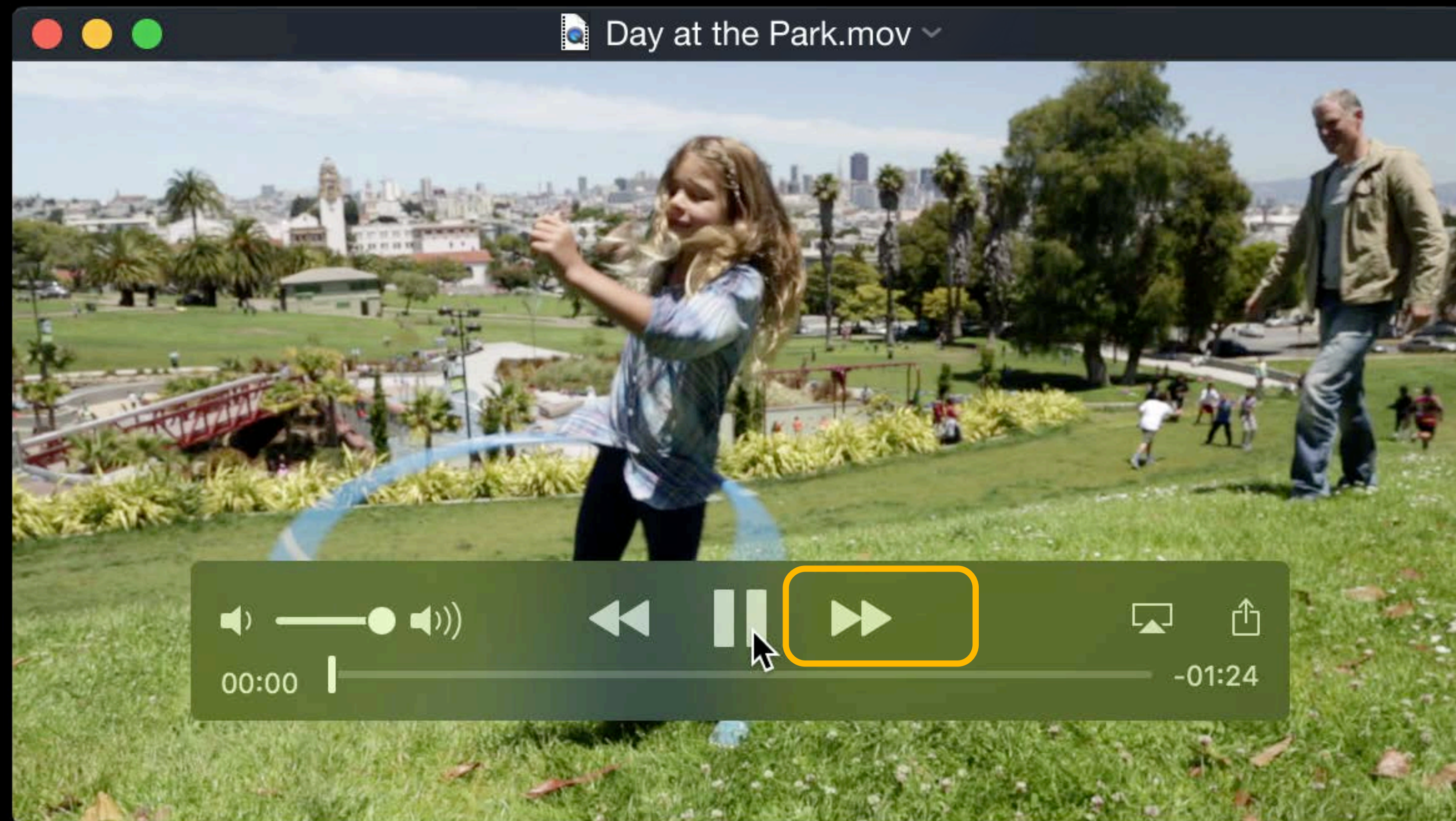
Accelerator Controls

Case Study: Quicktime Player



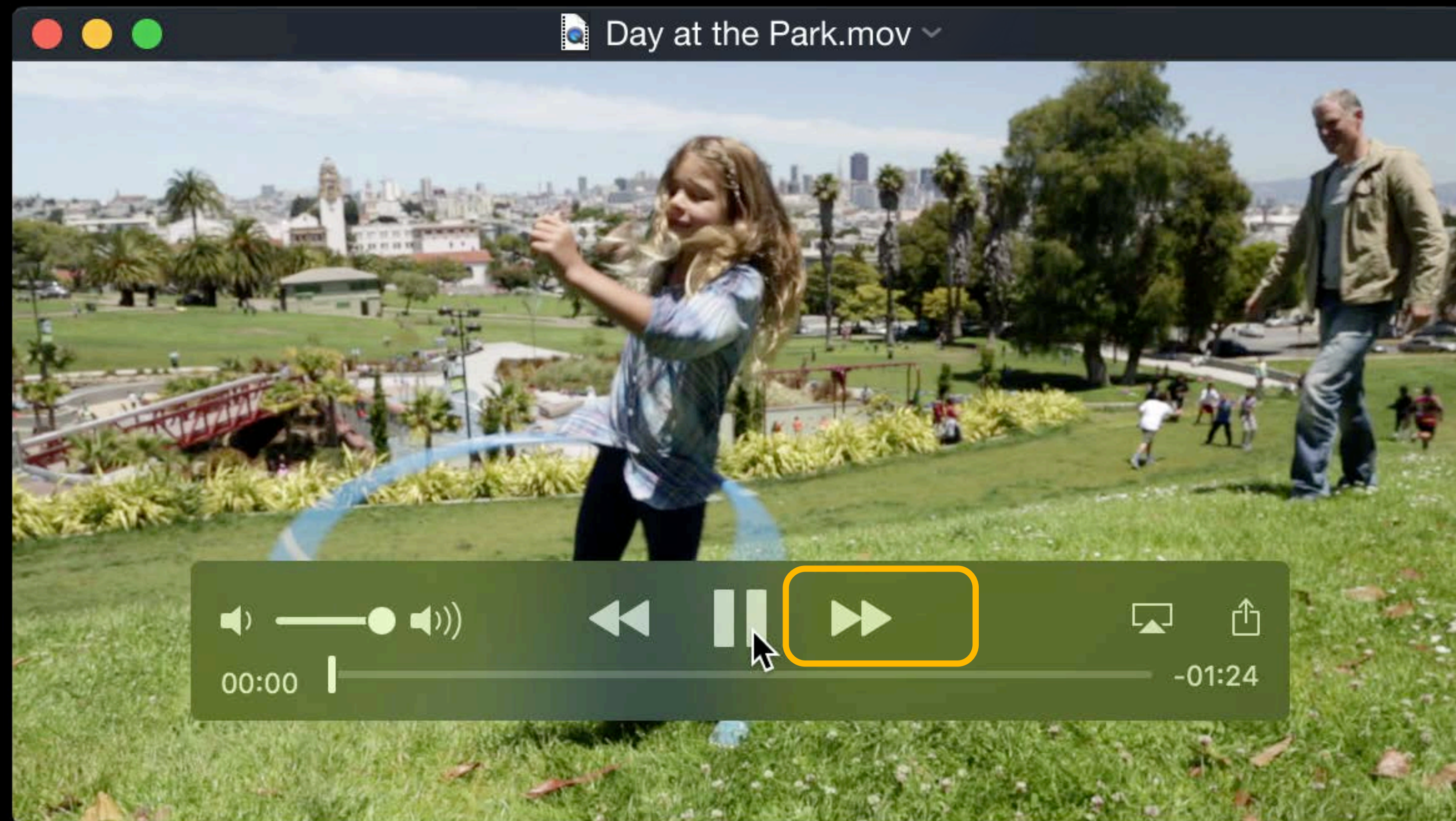
Accelerator Controls

Case Study: Quicktime Player



Accelerator Controls

Case Study: Quicktime Player

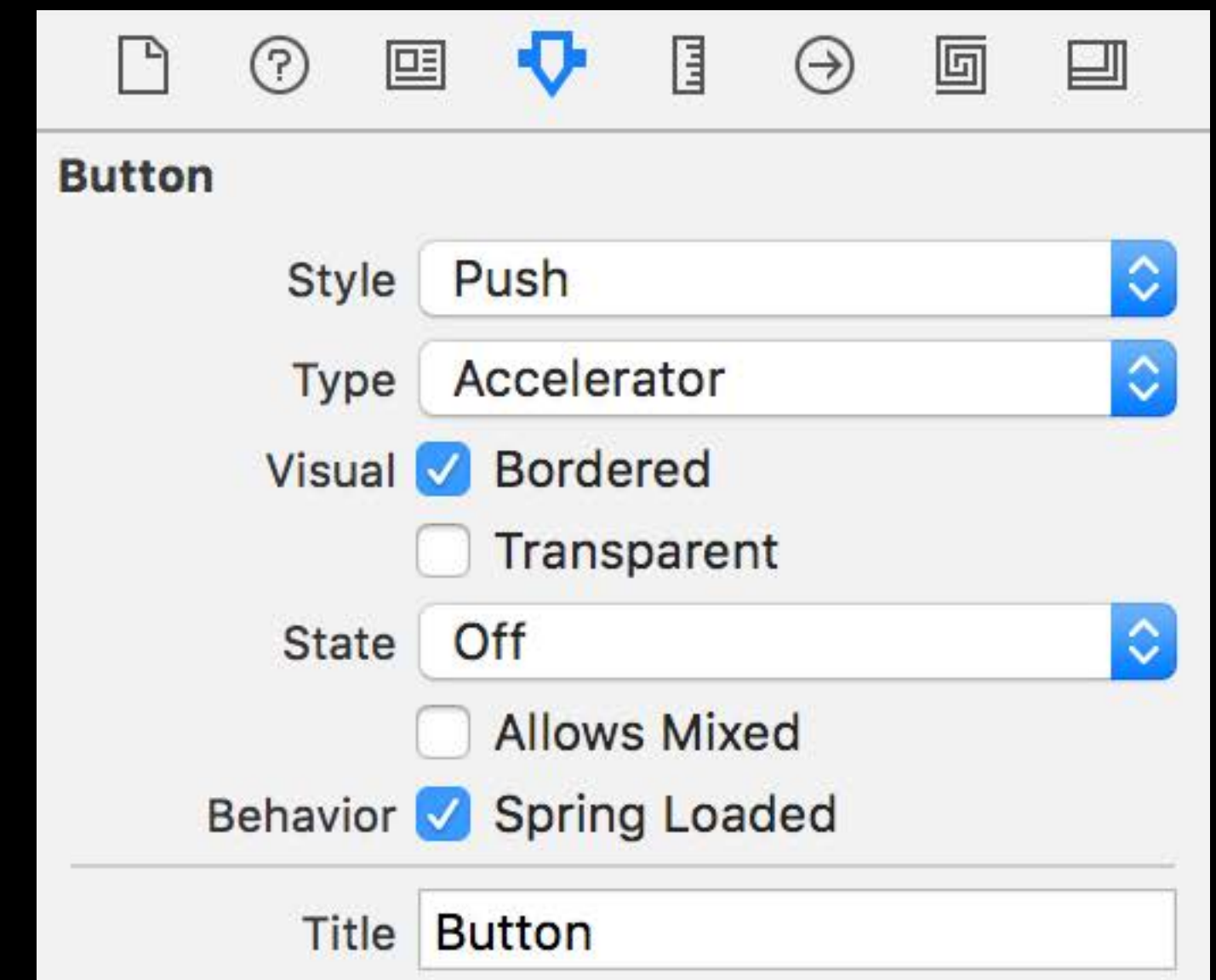


Accelerator Controls

Case Study: Quicktime Player

NSButton

```
func setButtonType(NSButtonType)  
    .AcceleratorButton  
    .MultiLevelAcceleratorButton
```

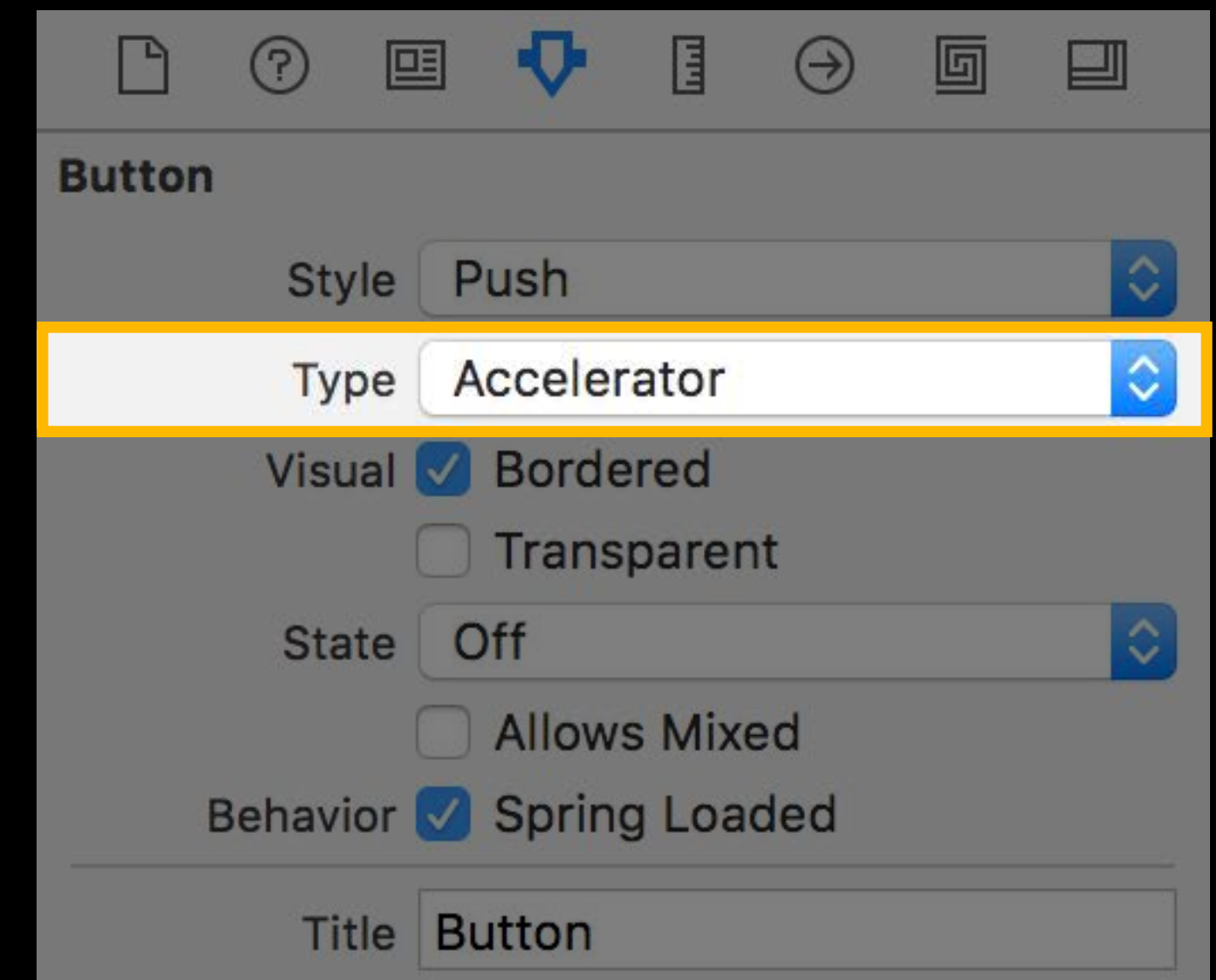


Accelerator Controls

Case Study: Quicktime Player

NSButton

```
func setType(NSButtonType)  
    .AcceleratorButton  
    .MultiLevelAcceleratorButton
```

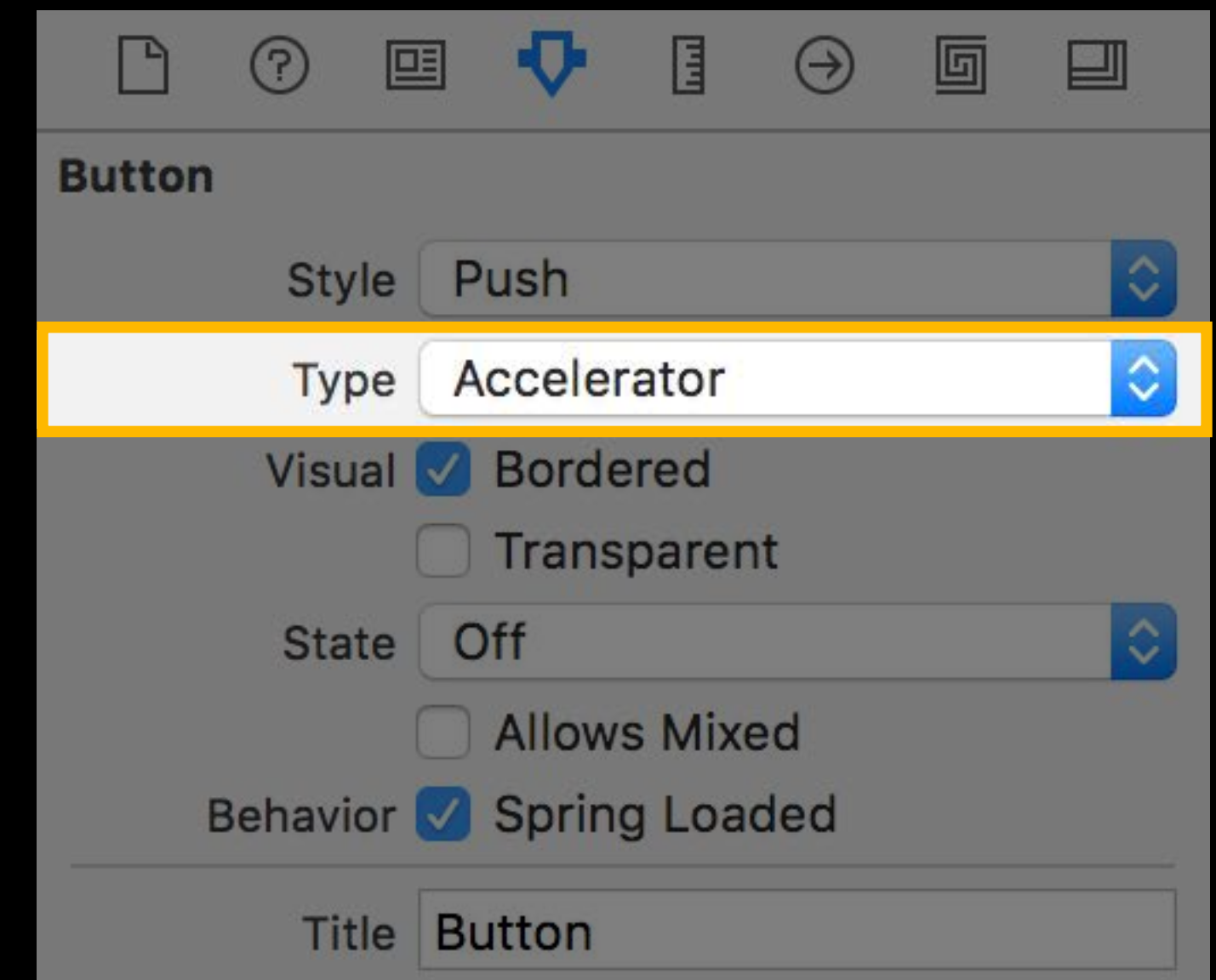


Case Study: Quicktime Player

UIButton

```
func setButtonType(NSButtonType)
    .AcceleratorButton
    .MultiLevelAcceleratorButton
```

```
var doubleValue: Double
    Range: 0, [1-2]
```



Accelerator Controls

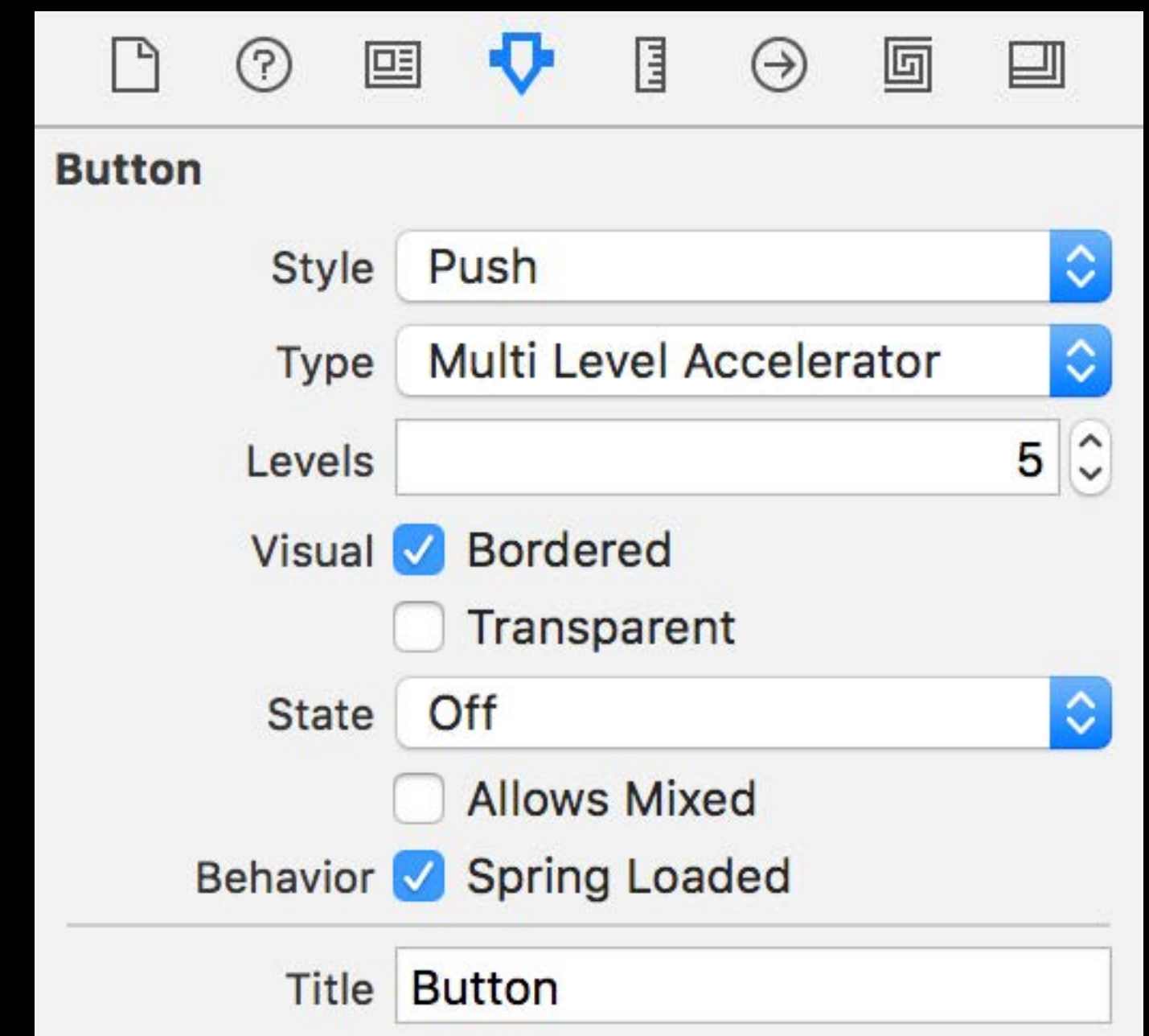
Case Study: Quicktime Player

NSButton

```
func setType(NSButtonType)
    .AcceleratorButton
    .MultiLevelAcceleratorButton
```

```
var doubleValue: Double
    Range: [0–5]
```

```
var maxAcceleratorLevel: Int
    Range: [1–5]
```



Accelerator Controls

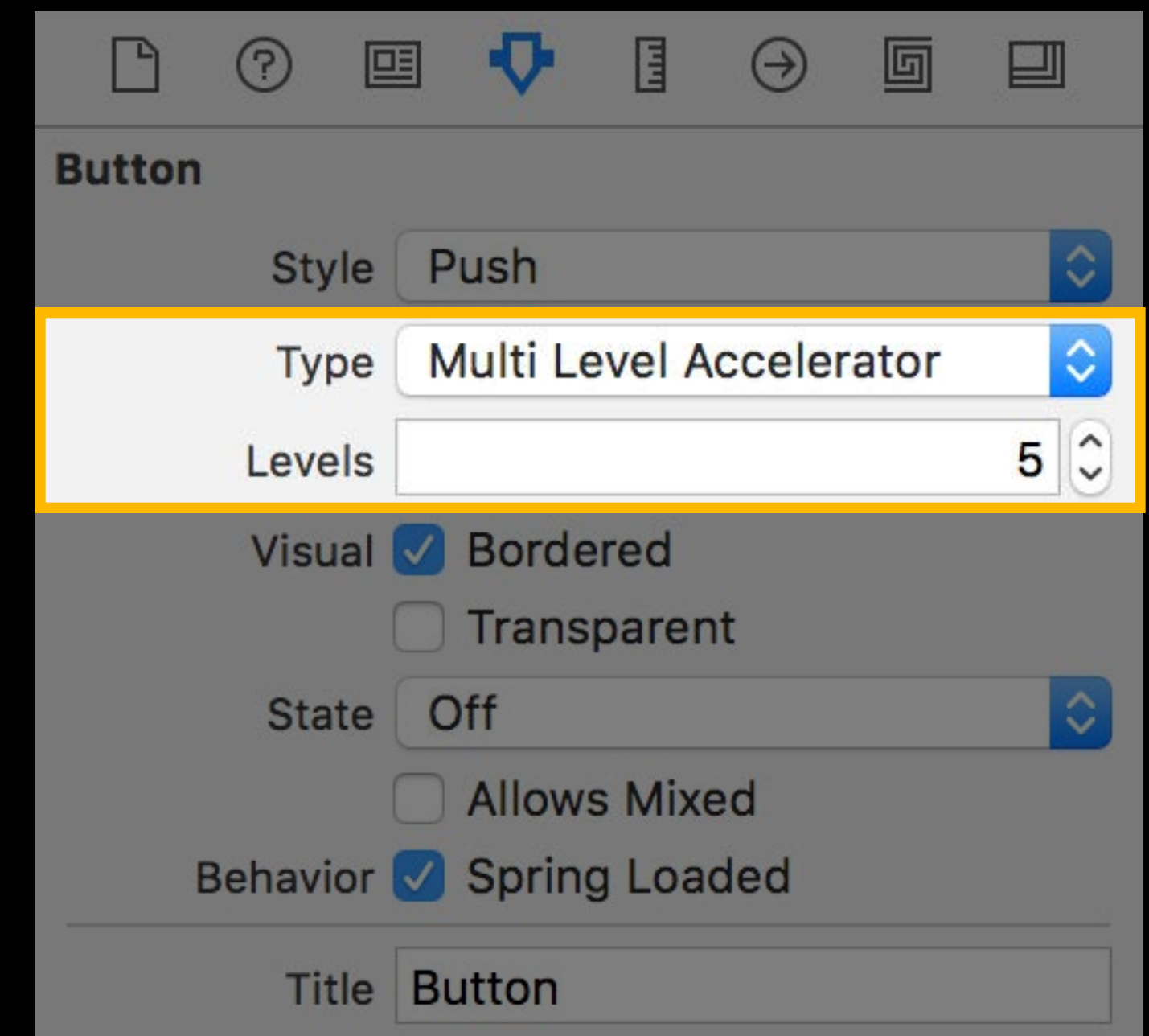
Case Study: Quicktime Player

NSButton

```
func setType(NSButtonType)
    .AcceleratorButton
    .MultiLevelAcceleratorButton
```

```
var doubleValue: Double
    Range: [0–5]
```

```
var maxAcceleratorLevel: Int
    Range: [1–5]
```



Accelerator Controls

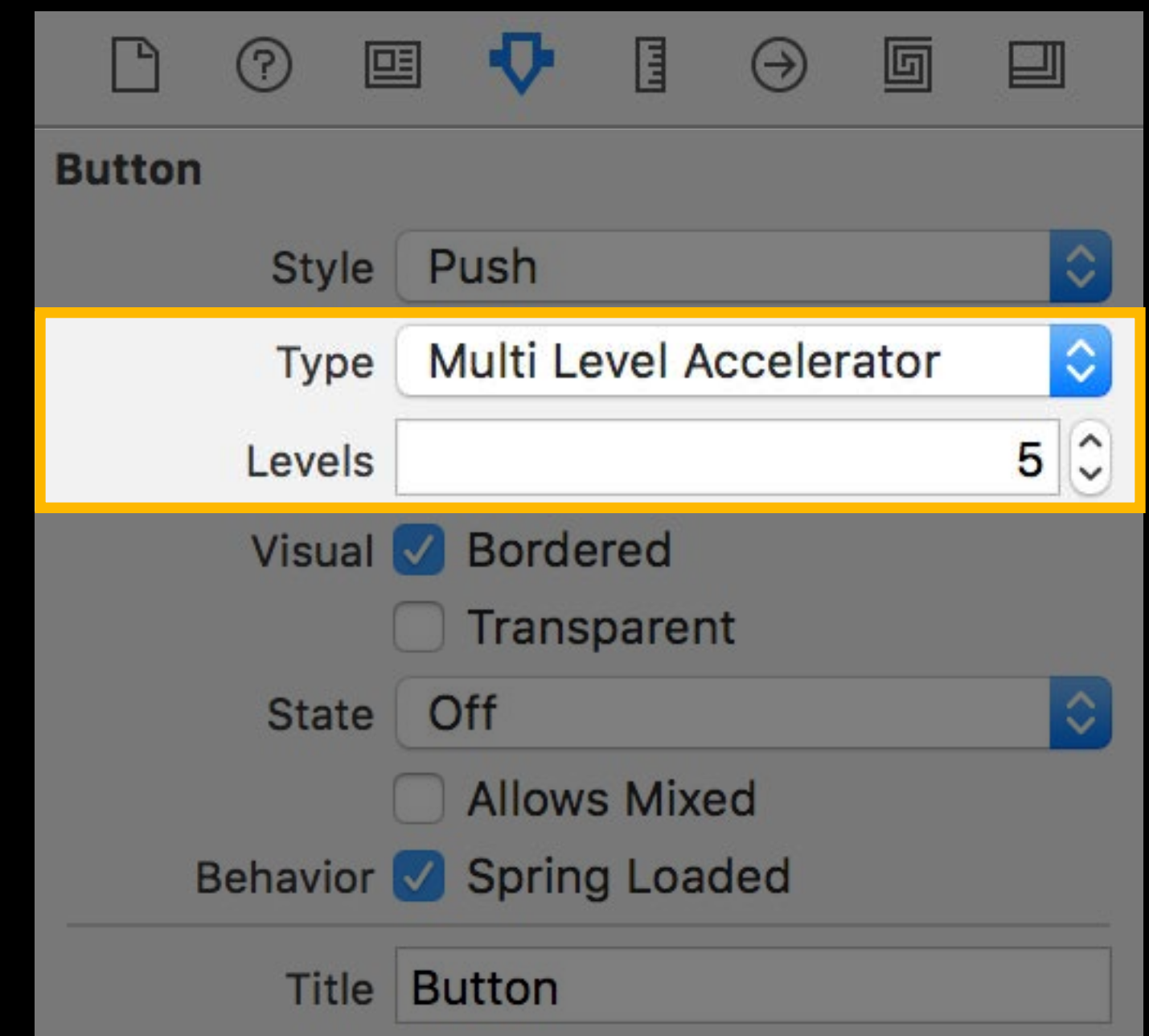
Case Study: Quicktime Player

NSButton

```
func setButtonType(NSButtonType)
    .AcceleratorButton
    .MultiLevelAcceleratorButton
```

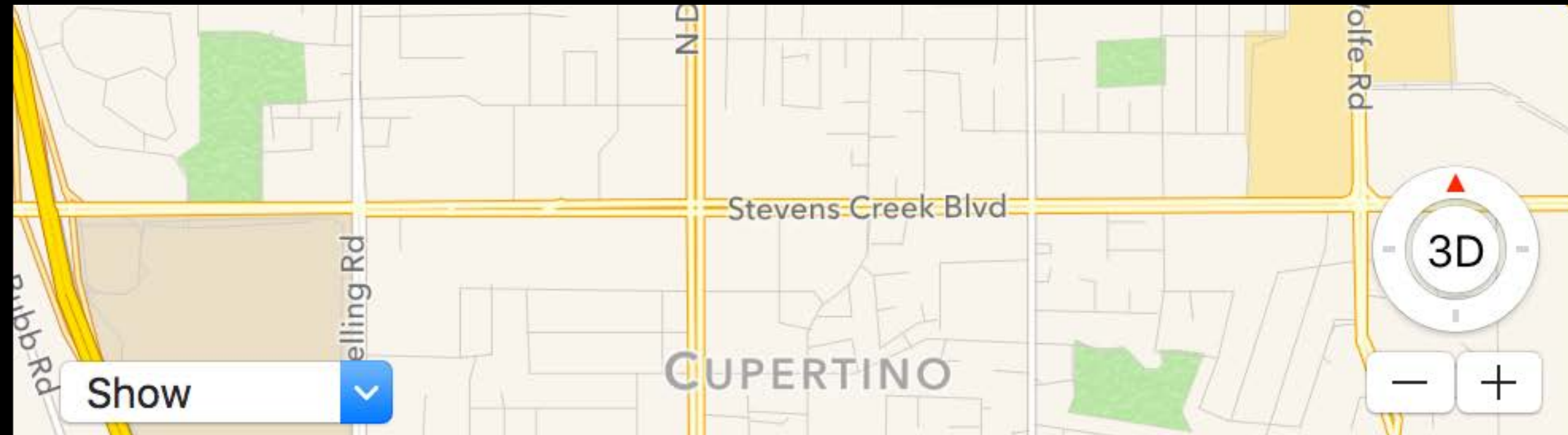
```
var doubleValue: Double
```

0	1	2	3	4	5
1X	2X	5X	10X	30X	60x



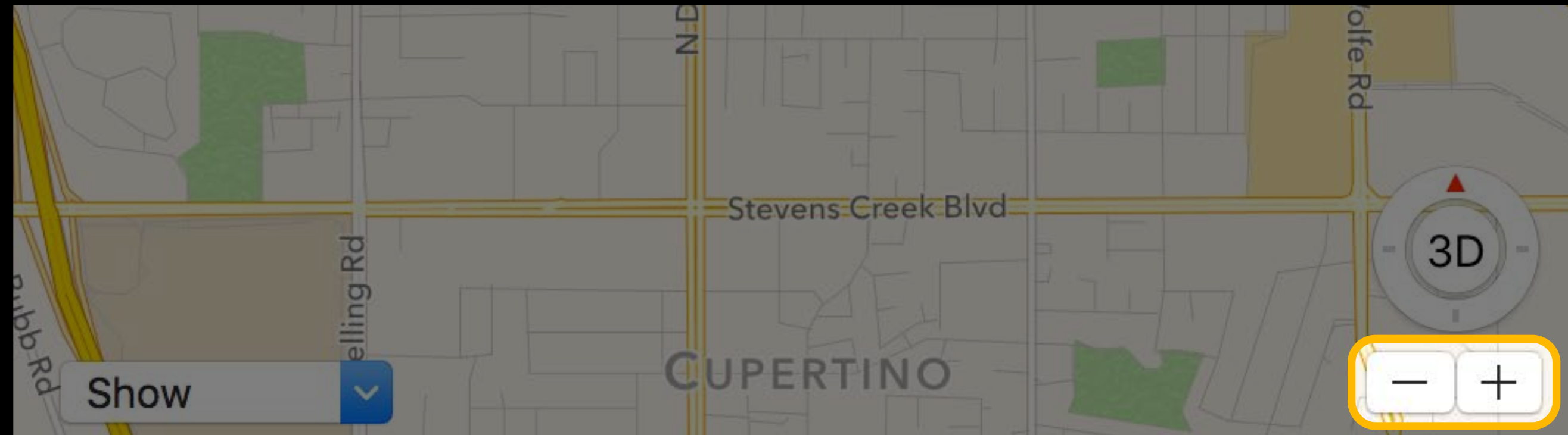
Accelerator Controls

Case Study: Maps



Accelerator Controls

Case Study: Maps

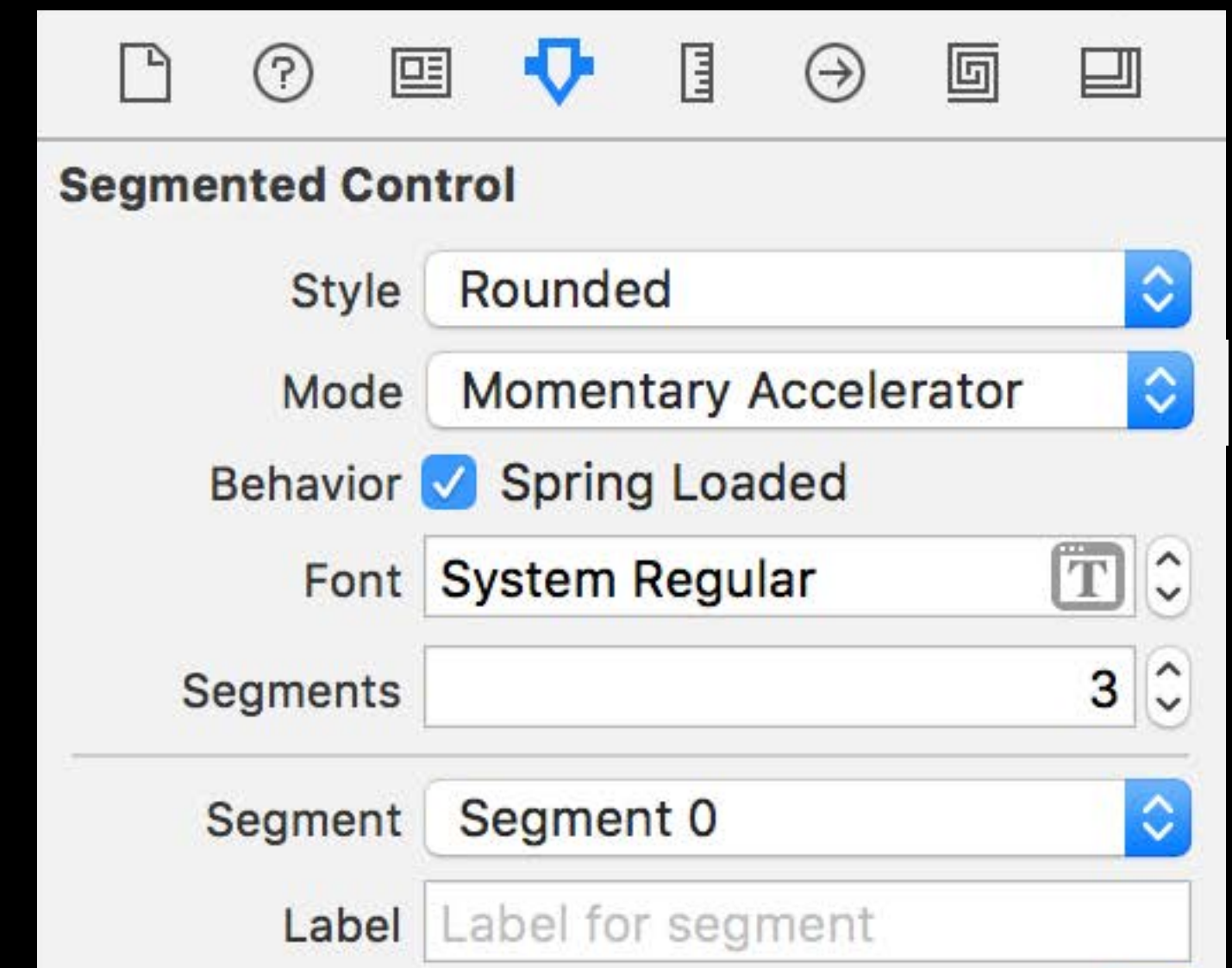


Accelerator Controls

Case Study: Maps

NSSegmentedControl

```
var trackingMode: NSSegmentSwitchTracking  
    .MomentaryAccelerator
```

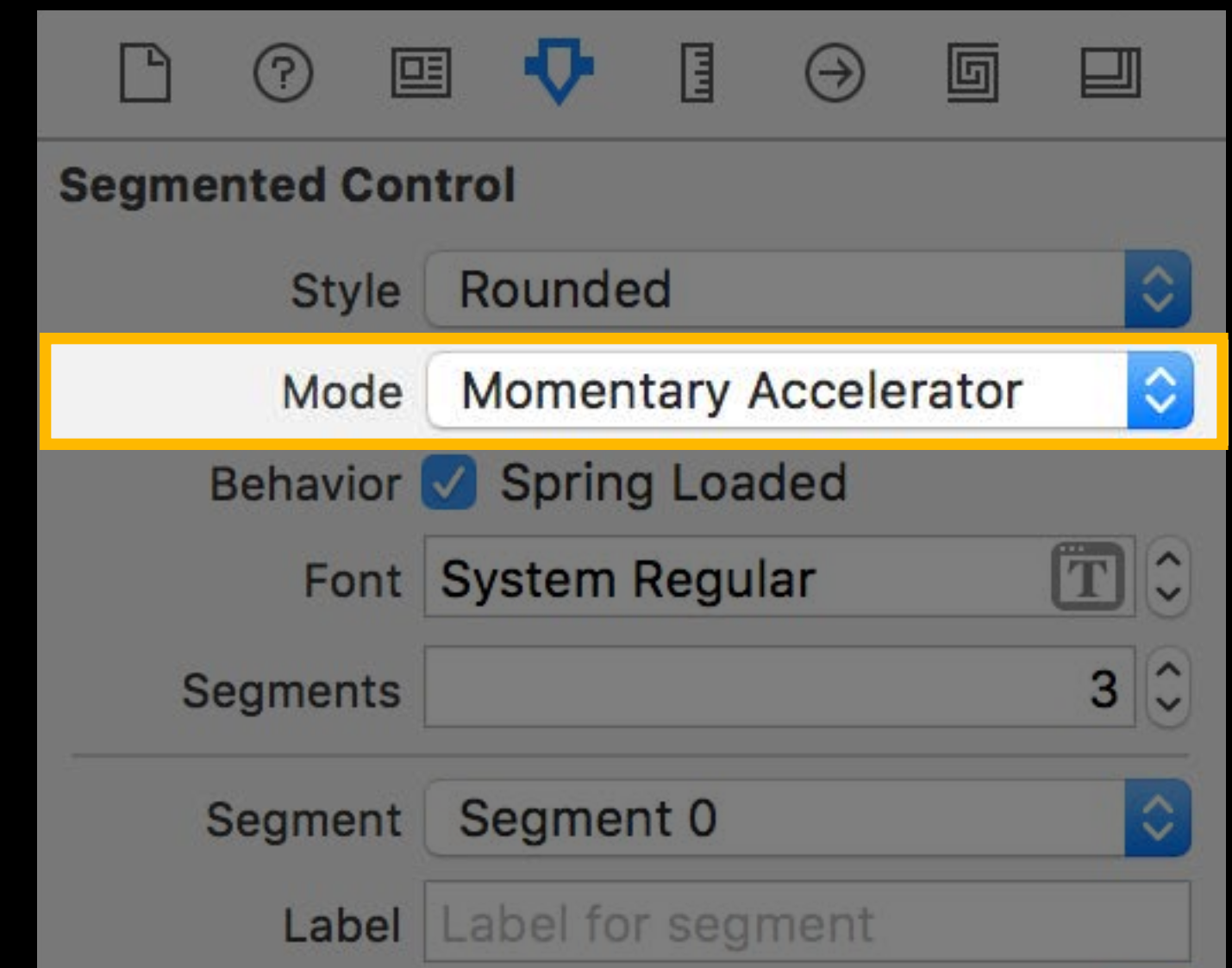


Accelerator Controls

Case Study: Maps

NSSegmentedControl

```
var trackingMode: NSSegmentSwitchTracking  
    .MomentaryAccelerator
```



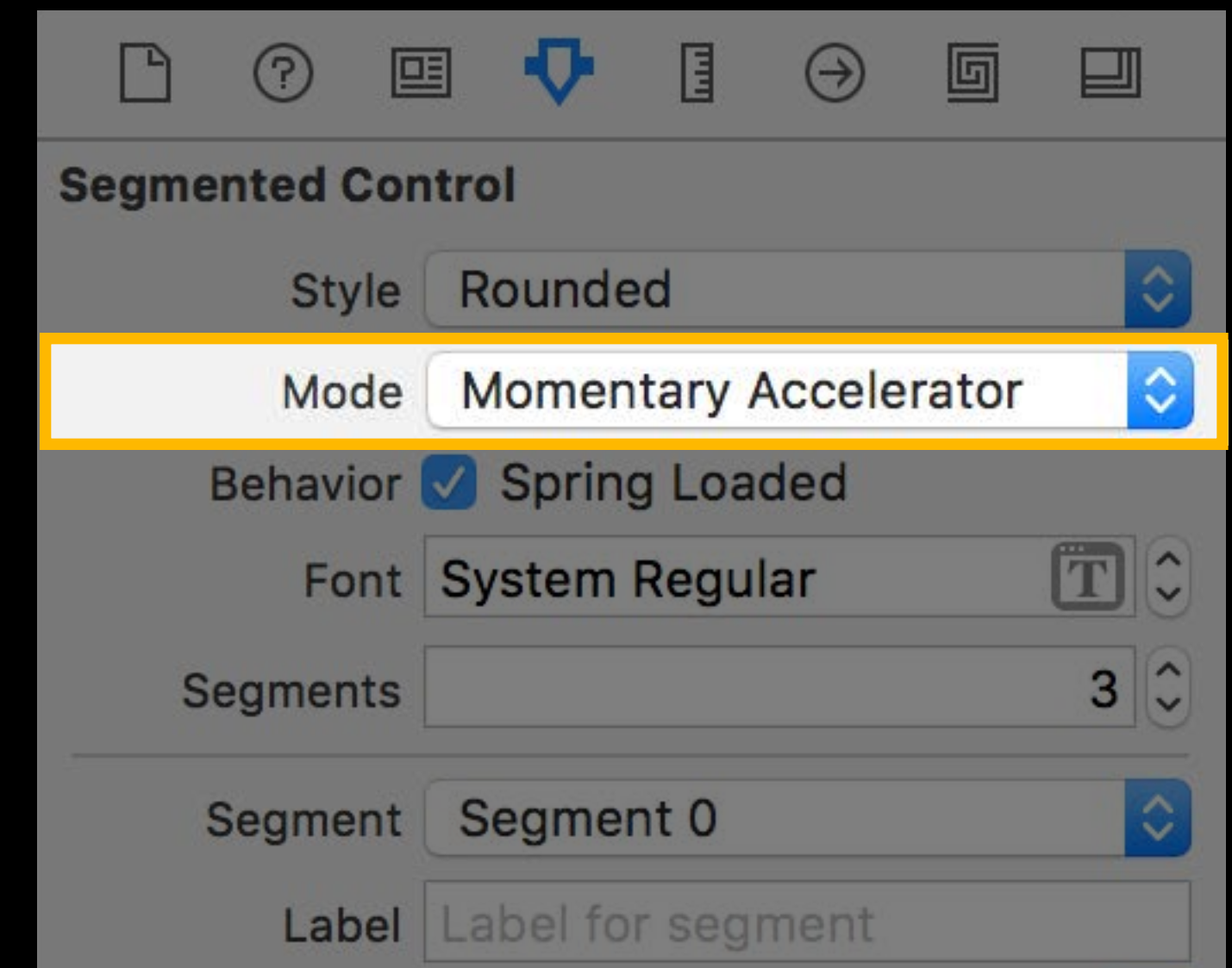
Accelerator Controls

Case Study: Maps

NSSegmentedControl

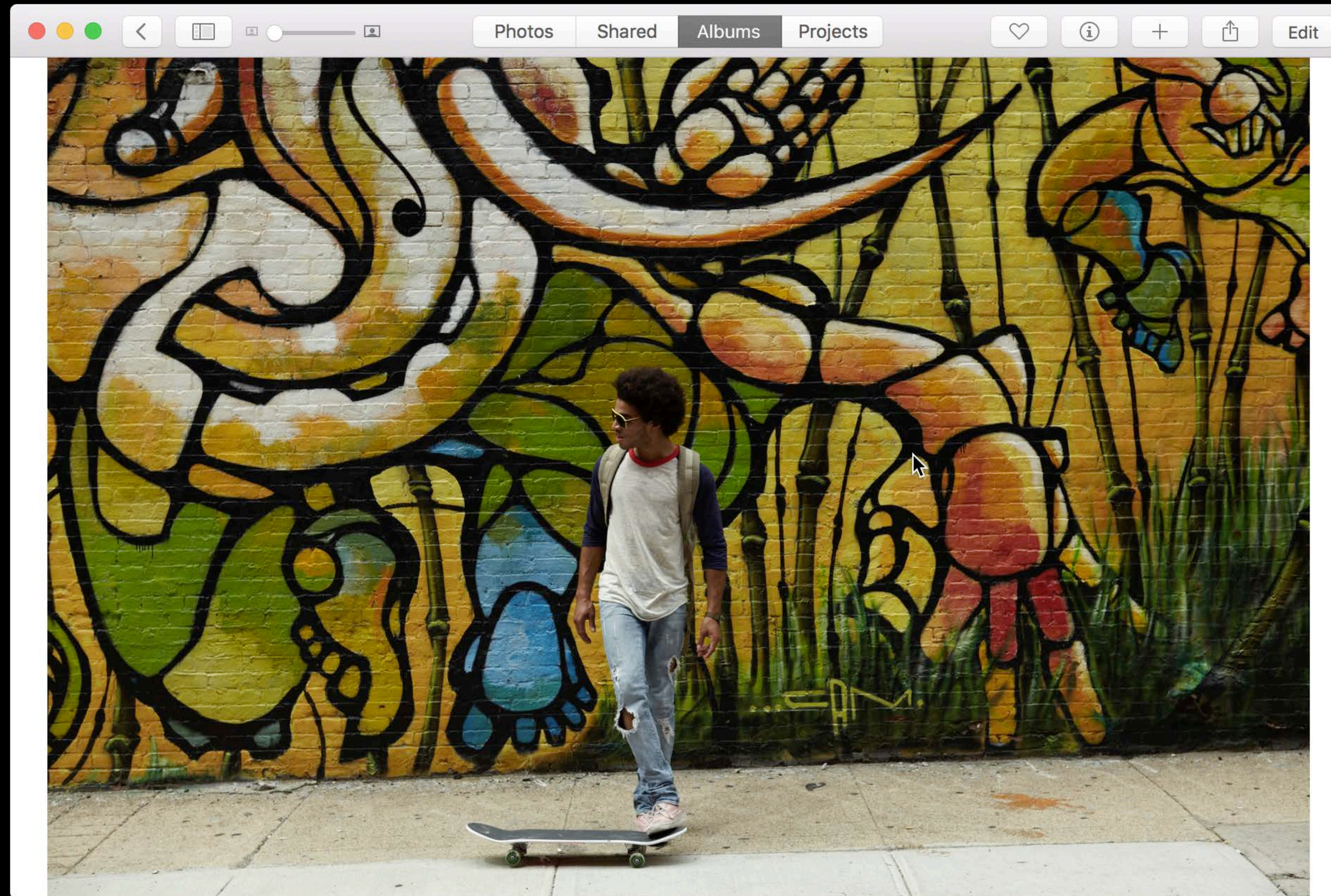
```
var trackingMode: NSSegmentSwitchTracking  
    .MomentaryAccelerator
```

```
func doubleValueForSelectedSegment() -> Double  
    Range: 0, [1-2]
```



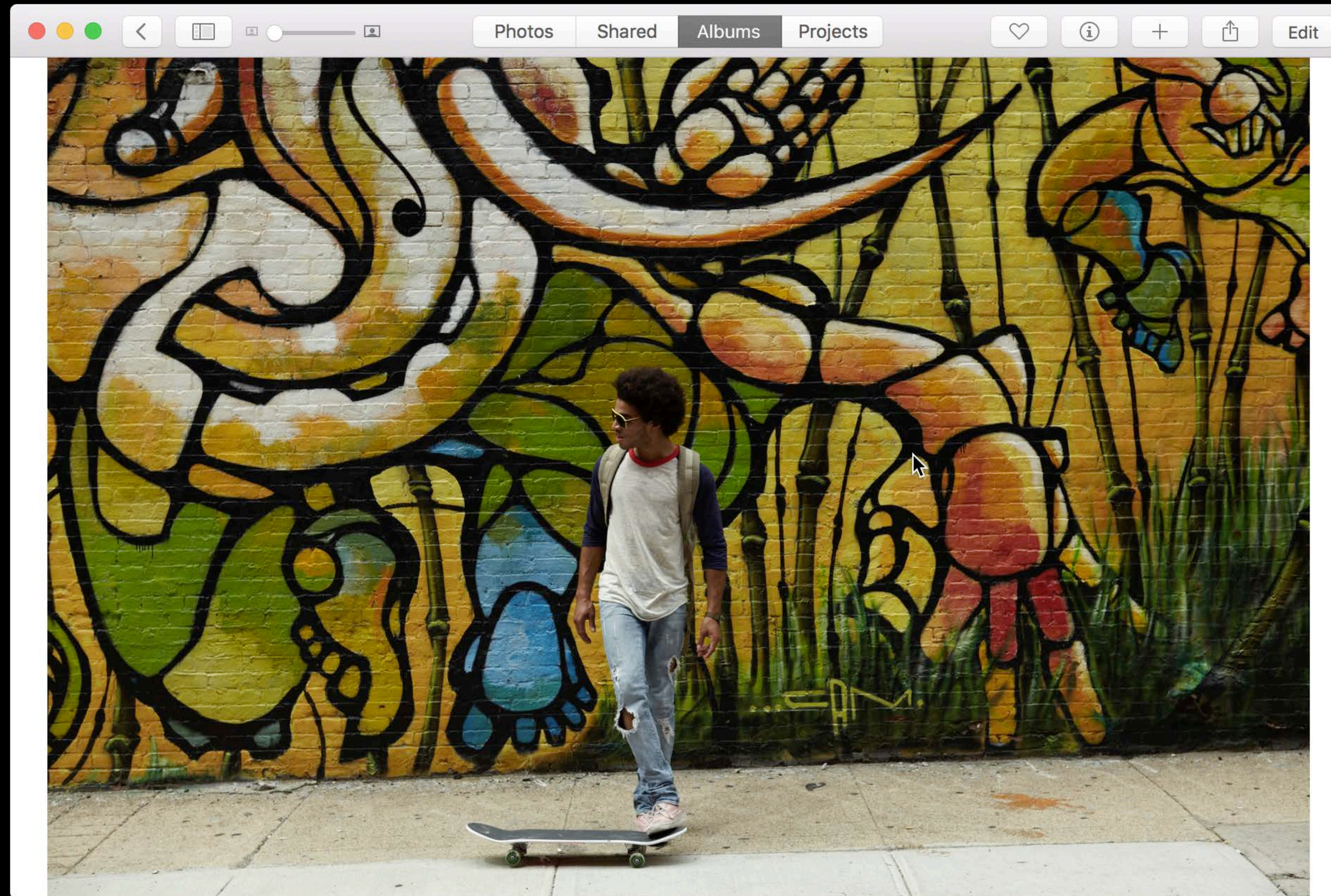
Accelerator Controls

Case Study: Photos



Accelerator Controls

Case Study: Photos




Accelerator Controls

Case Study: Photos

NSControl

```
var continuous: Bool
```


Control


Line Break 


☐ Truncates Last Visible Line

State ☒ Enabled
☒ Continuous
☐ Refuses First Responder

Tooltips ☐ Allows Expansion Tooltips

Text Direction 

Layout 

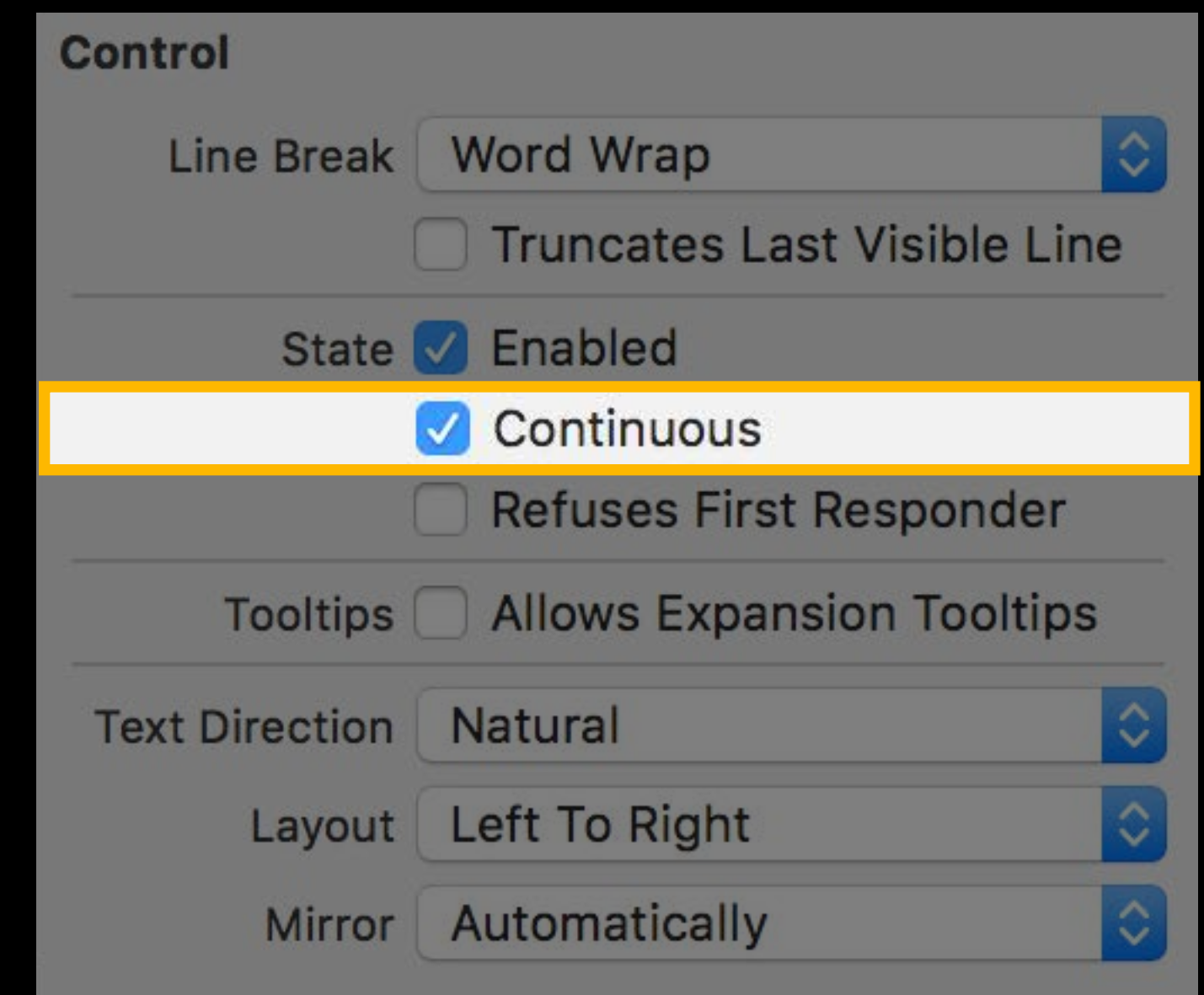
Mirror 

Accelerator Controls

Case Study: Photos

NSControl

```
var continuous: Bool
```



Accelerator Controls

NSButton, NSSegmentedControl

Accelerated

Maps: +/- zoom buttons

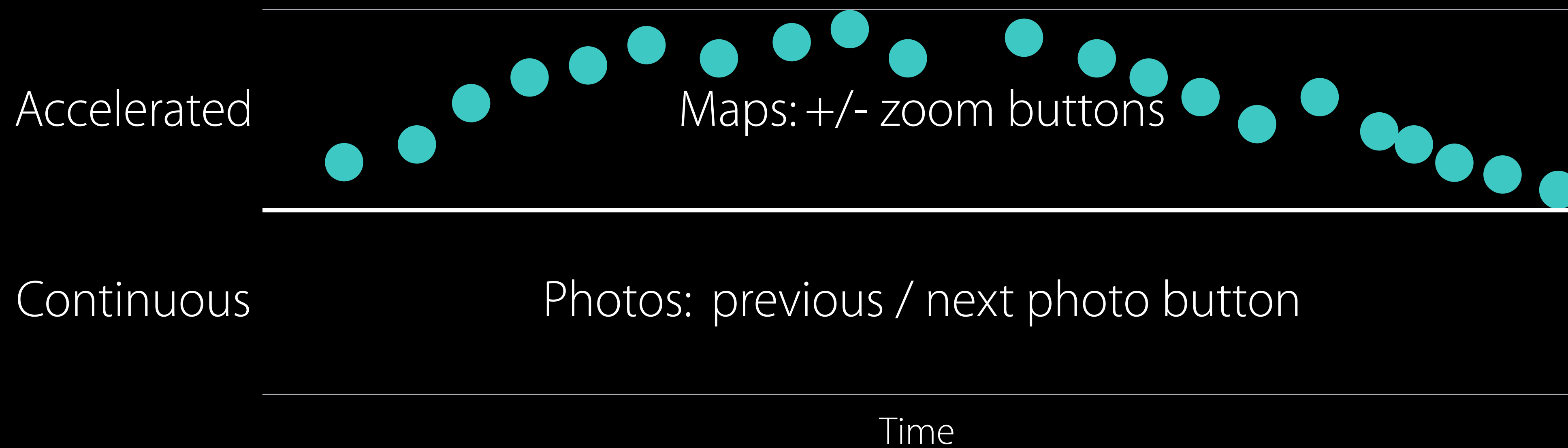
Continuous

Photos: previous / next photo button

Time

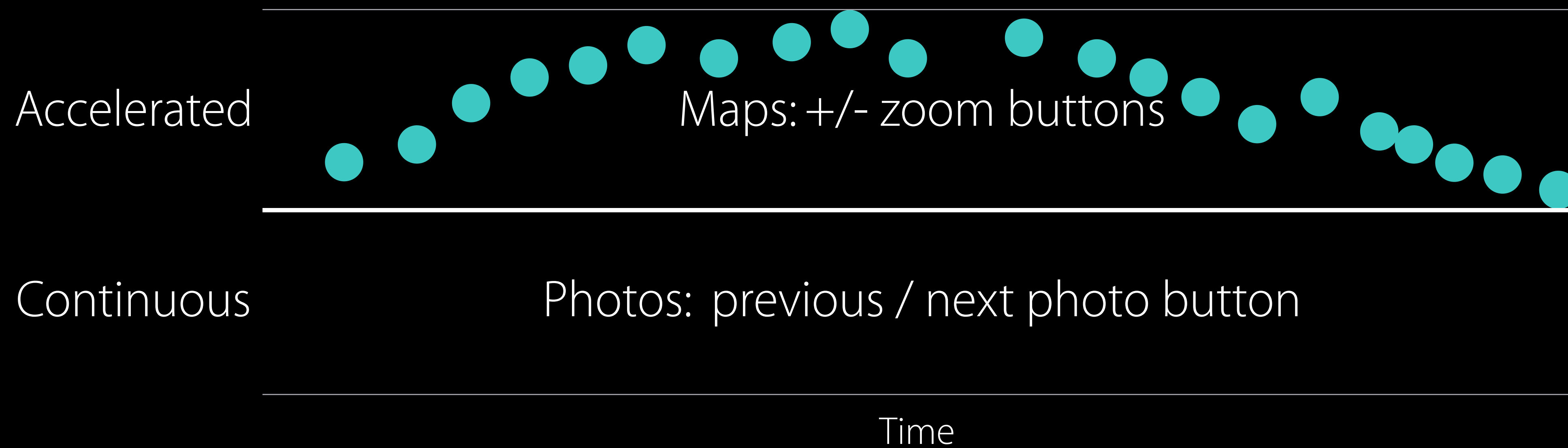
Accelerator Controls

NSButton, NSSegmentedControl



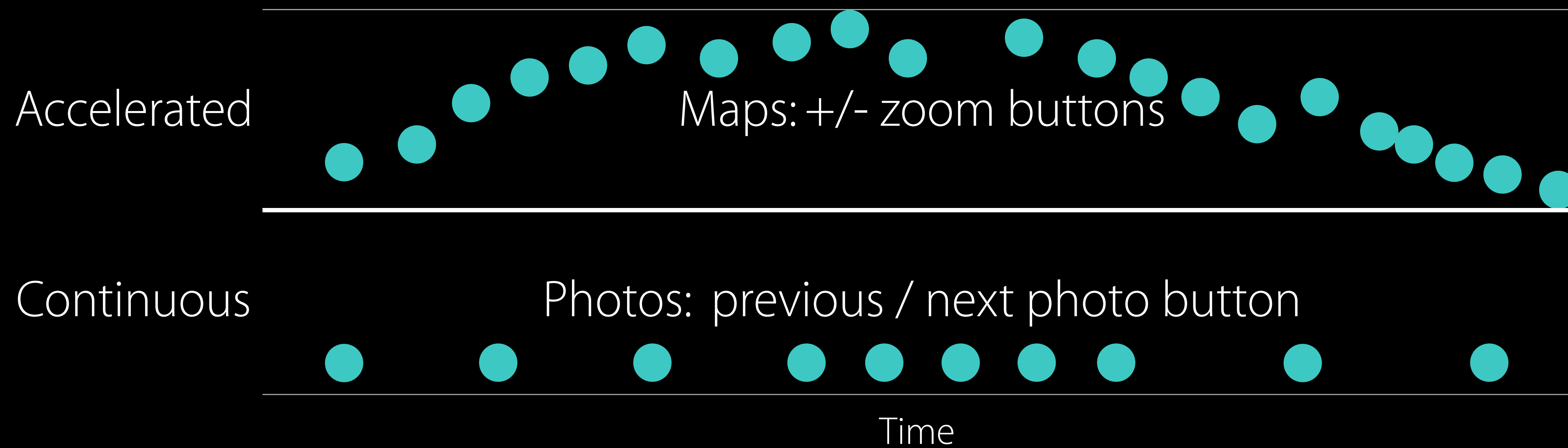
Accelerator Controls

NSButton, NSSegmentedControl



Accelerator Controls

NSButton, NSSegmentedControl



Squire Summary

Using the built-in tools

Table Row Actions

Spring Loaded Controls

Accelerator Controls

Knight

Understanding how the force flows

Knight Training Schedule

Understanding how the force flows

Force event stream

Spring loading protocol

Alignment feedback

Event Stream

Event Stream

NSLeftMouseDown	NSEventTypeMagnify	NSRightMouseDown
NSLeftMouseDragged	NSEventTypeRotate	NSRightMouseDragged
NSLeftMouseUp		NSRightMouseUp
NSMouseEntered	NSKeyDown	NSScrollWheel
NSMouseExited	NSKeyUp	NSFlagsChanged

Event Stream

Pressure Gesture

NSLeftMouseDown	NSEventTypeMagnify	NSRightMouseDown
NSLeftMouseDragged	NSEventTypeRotate	NSRightMouseDragged
NSLeftMouseUp	NSEventTypePressure	NSRightMouseUp
NSMouseEntered	NSKeyDown	NSScrollWheel
NSMouseExited	NSKeyUp	NSFlagsChanged

Event Stream

Pressure Gesture

NSLeftMouseDownMask	NSEventMaskMagnify	NSRightMouseDownMask
NSLeftMouseDraggedMask	NSEventMaskRotate	NSRightMouseDraggedMask
NSLeftMouseUpMask	NSEventMaskPressure	NSRightMouseUpMask
NSMouseEnteredMask	NSKeyDownMask	NSScrollWheelMask
NSMouseExitedMask	NSKeyUpMask	NSFlagsChangedMask

Event Stream

Pressure Gesture

```
trackEventsMatchingMask(NSEventMask,...)
```

```
[NSLeftMouseDownMask, NSLeftMouseUpMask, NSEventMaskPressure]
```

Event Stream

Pressure Gesture

```
trackEventsMatchingMask(NSEventMask,...)
```

```
[.LeftMouseDraggedMask, .LeftMouseUpMask, .EventMaskPressure]
```

Event Stream

Pressure Gesture

```
trackEventsMatchingMask(NSEventMask,...)
```

```
[.LeftMouseDraggedMask, .LeftMouseUpMask, .EventMaskPressure]
```

```
func pressureChangeWithEvent(NSEvent)
```


Event Stream

Pressure Gesture

NSEventTypePressure

NSEventMaskPressure

```
var phase: NSEventPhase { get }  
    .Began -> .Changed -> .Ended
```



Event Stream

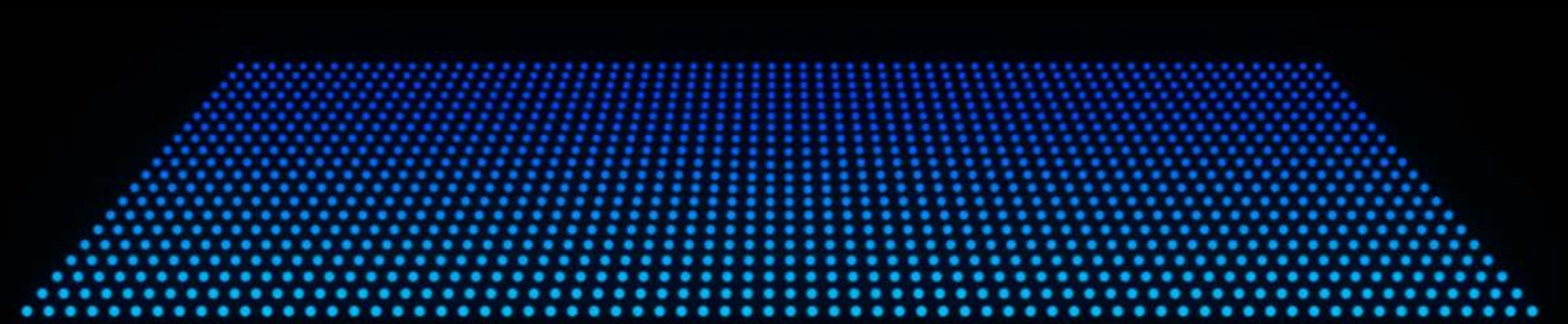
Pressure Gesture

NSEventTypePressure

NSEventMaskPressure

var phase: NSEventPhase { get }

var **stage**: Int { get }



Event Stream

Pressure Gesture

`NSEventTypePressure`

`NSEventMaskPressure`

`var phase: NSEventPhase { get }`

`var stage: Int { get }`

`1 – click`



Event Stream

Pressure Gesture

`NSEventTypePressure`

`NSEventMaskPressure`

`var phase: NSEventPhase { get }`

`var stage: Int { get }`

`1 – click`



Event Stream

Pressure Gesture

NSEventTypePressure

NSEventMaskPressure

var phase: NSEventPhase { get }

var **stage**: Int { get }

1 – click

2 – force click



Event Stream

Pressure Gesture

`NSEventTypePressure`

`NSEventMaskPressure`

`var phase: NSEventPhase { get }`

`var stage: Int { get }`

`1 – click`

`2 – force click`



Event Stream

Pressure Gesture

`NSEventTypePressure`

`NSEventMaskPressure`

`var phase: NSEventPhase { get }`

`var stage: Int { get }`

1 – click

2 – force click

0 – gesture release



Event Stream

Pressure Gesture

NSEventTypePressure

NSEventMaskPressure

var phase: NSEventPhase { get }

var **stage**: Int { get }

1 – click

2 – force click

0 – **gesture release**

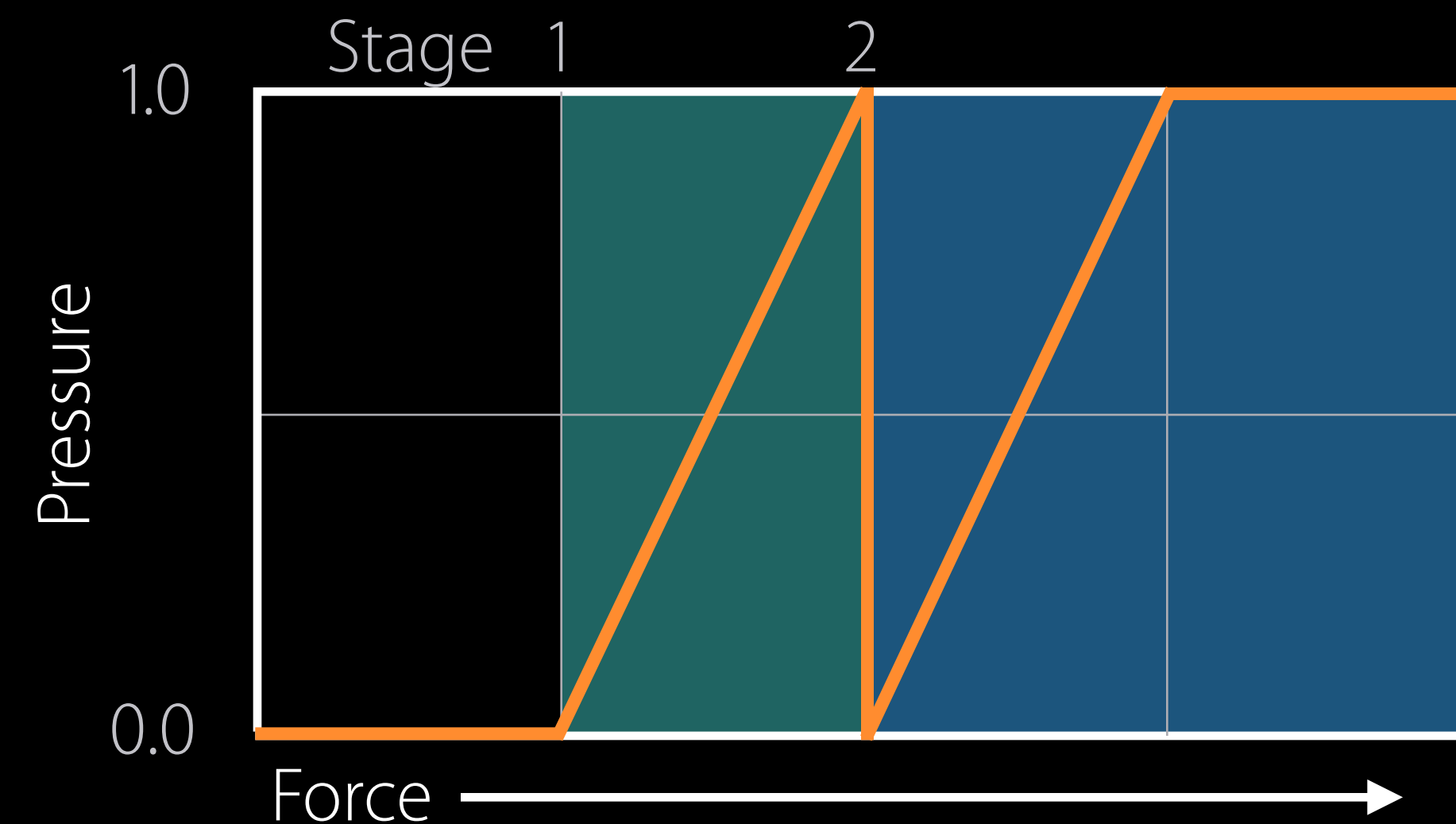


Event Stream

Pressure Gesture

NSEventTypePressure

```
var pressure: Float { get }  
    (of the current stage)  
    Range: [0–1]
```

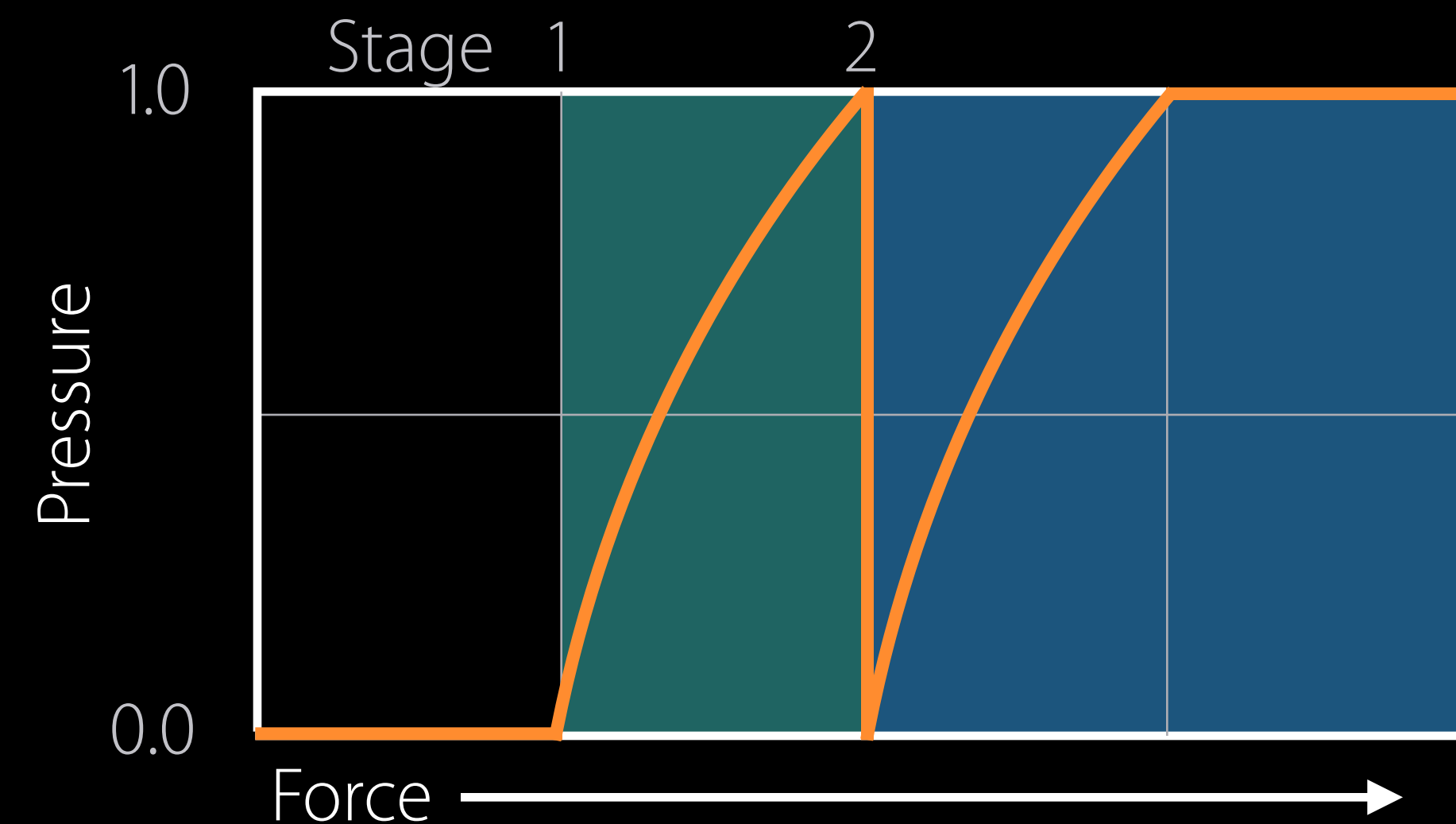


Event Stream

Pressure Gesture

NSEventTypePressure

```
var pressure: Float { get }  
    (of the current stage)  
    Range: [0–1]
```

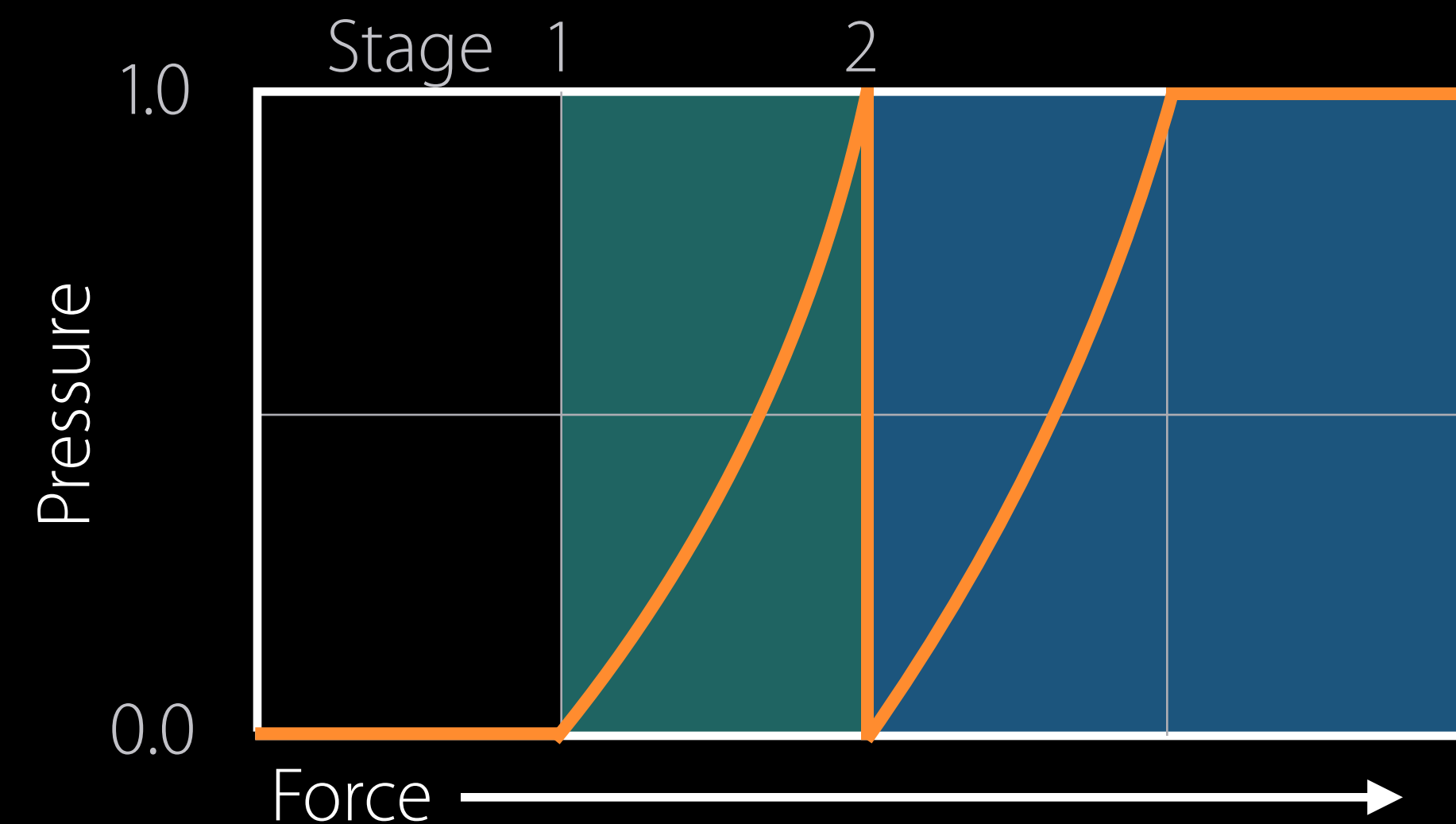


Event Stream

Pressure Gesture

NSEventTypePressure

```
var pressure: Float { get }  
    (of the current stage)  
    Range: [0–1]
```

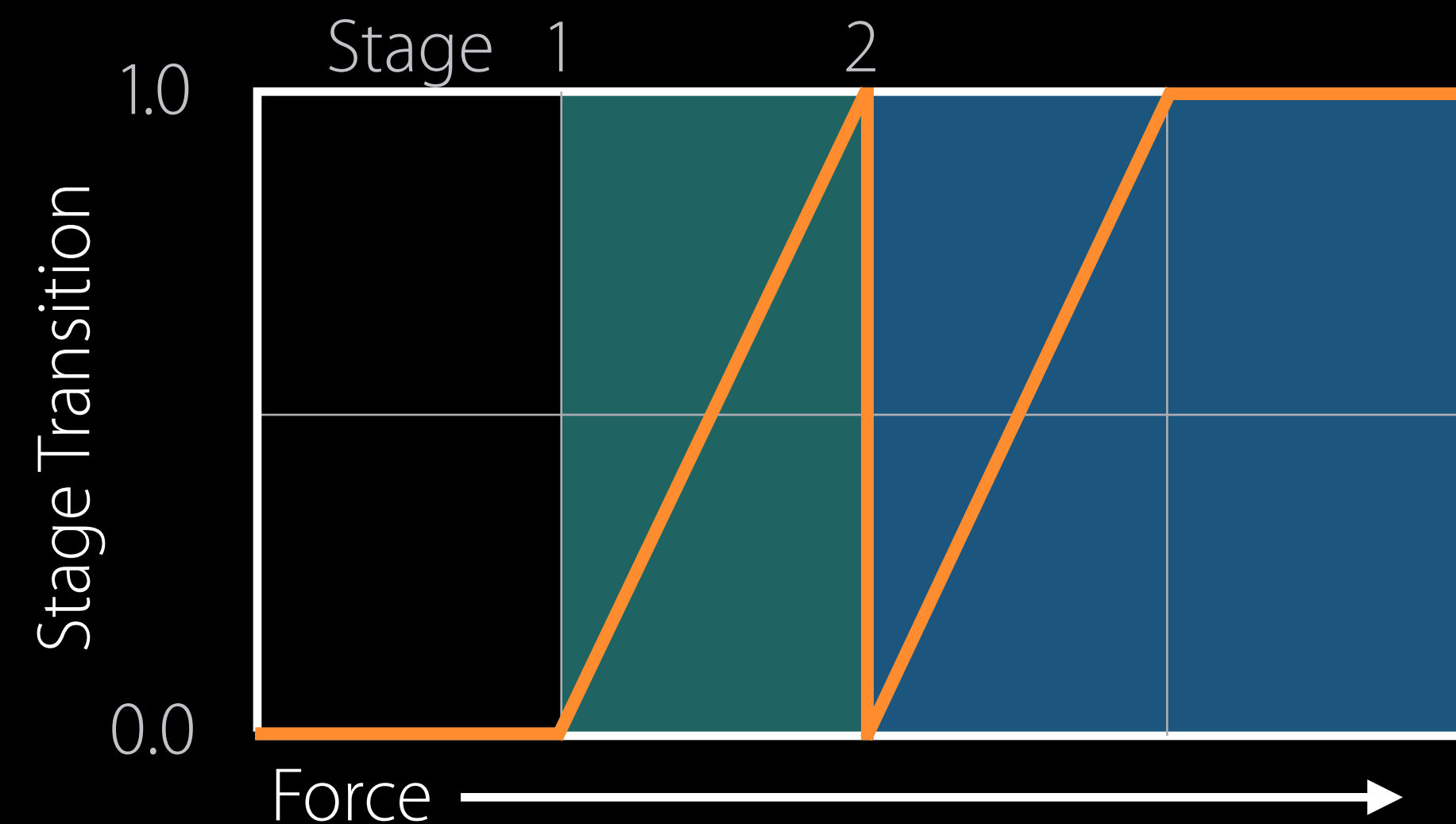


Event Stream

Pressure Gesture

NSEventTypePressure

```
var pressure: Float { get }  
var stageTransition: CGFloat { get }  
    Range: [0-1]
```

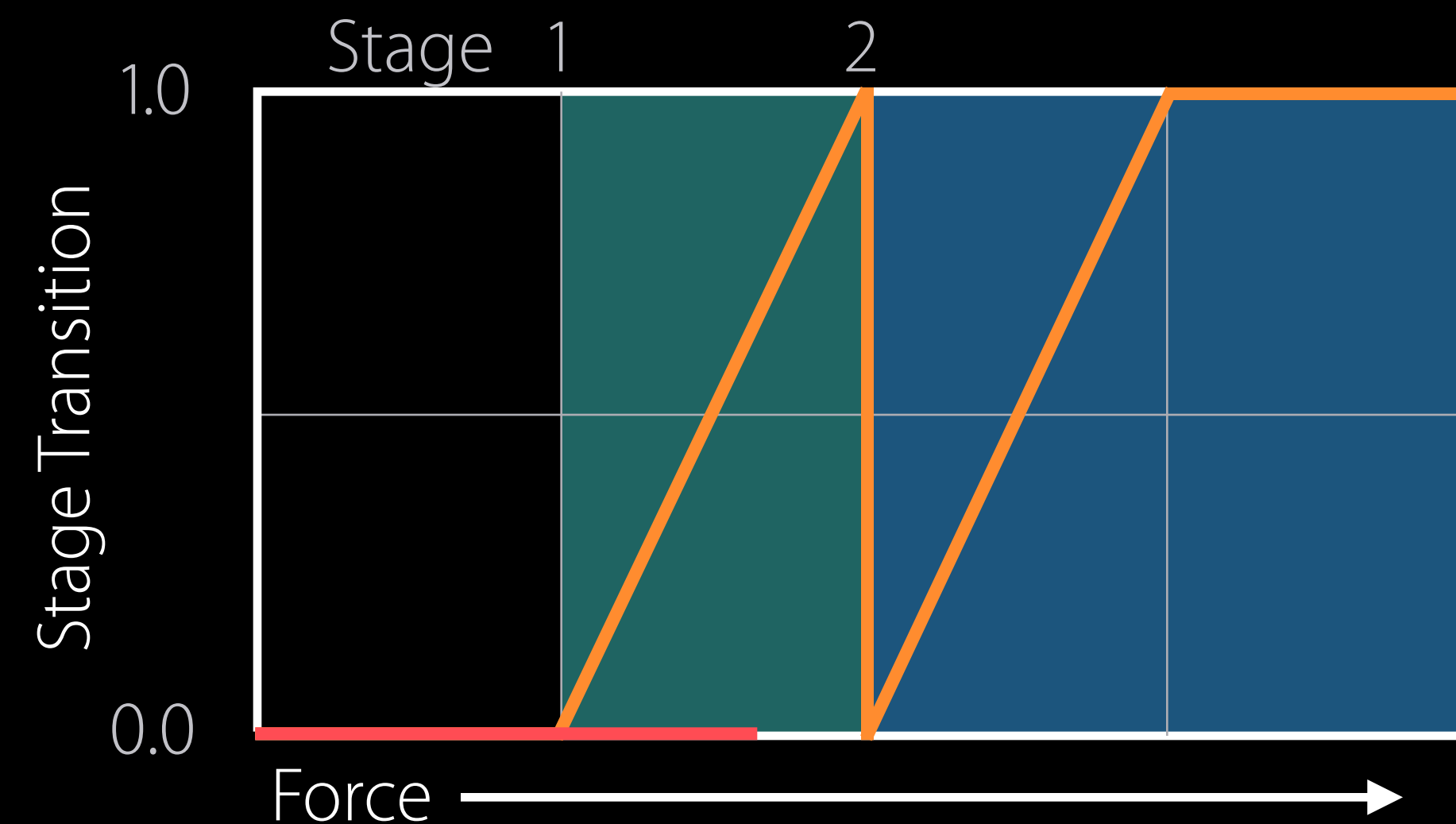


Event Stream

Pressure Gesture

NSEventTypePressure

```
var pressure: Float { get }  
var stageTransition: CGFloat { get }  
    Range: [0-1]
```

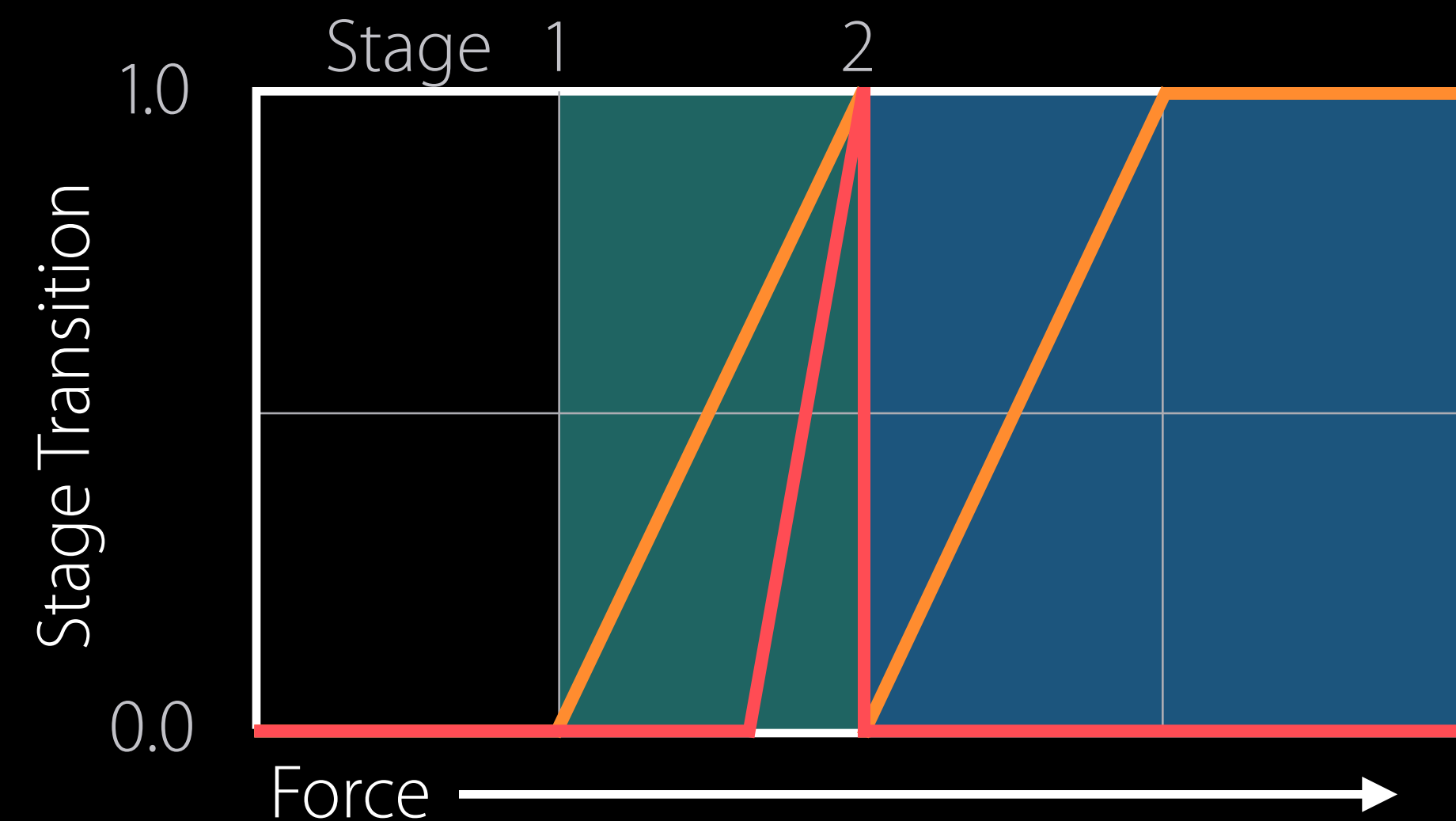


Event Stream

Pressure Gesture

NSEventTypePressure

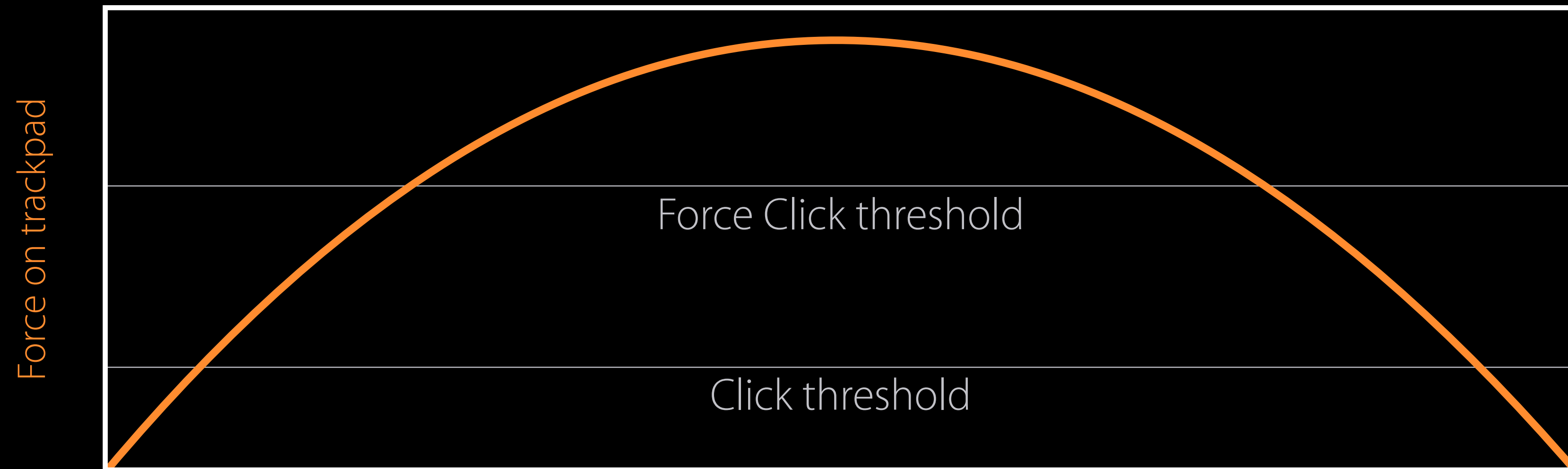
```
var pressure: Float { get }  
var stageTransition: CGFloat { get }  
    Range: [0-1]
```



Event Stream

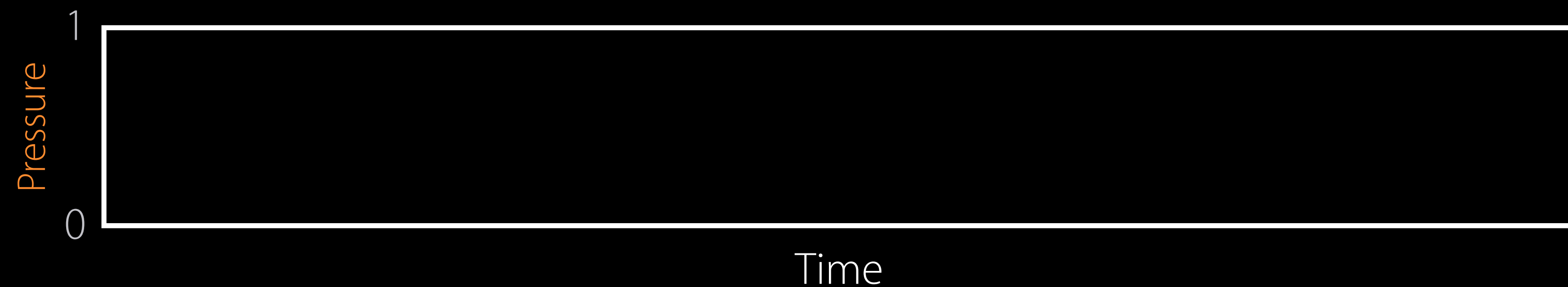
Parallel Mouse and Pressure event streams

User Action



Events Mouse:

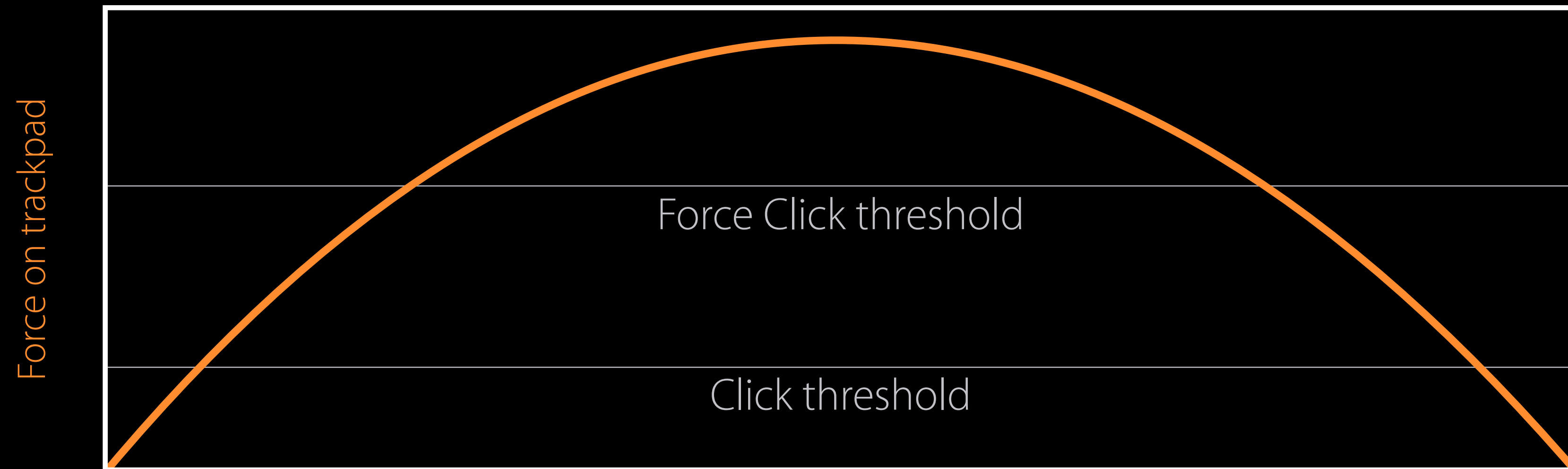
Stage:



Event Stream

Parallel Mouse and Pressure event streams

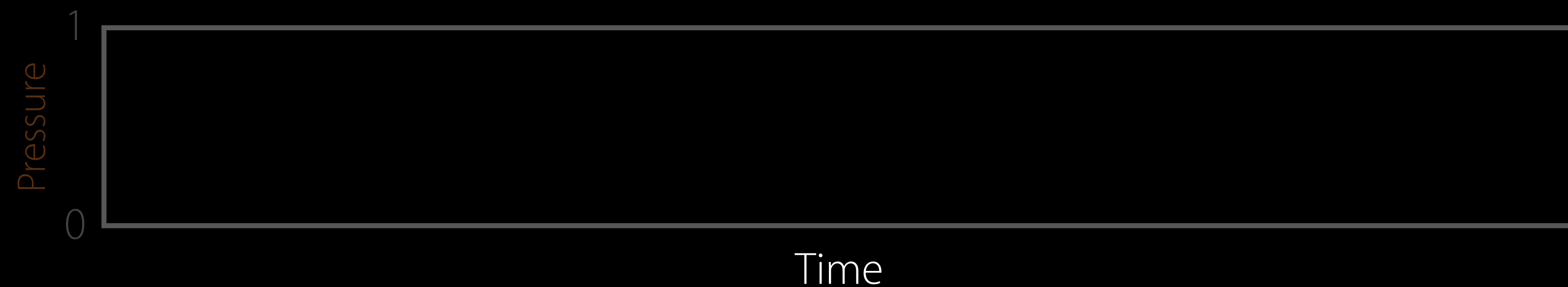
User Action



Events

Mouse:

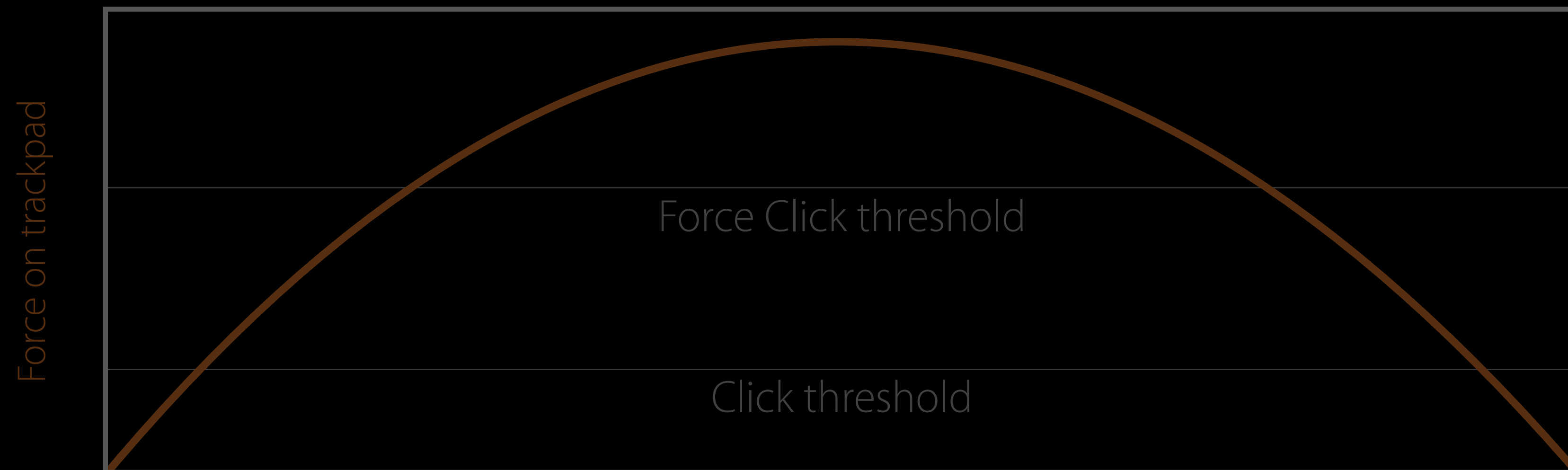
Stage:



Event Stream

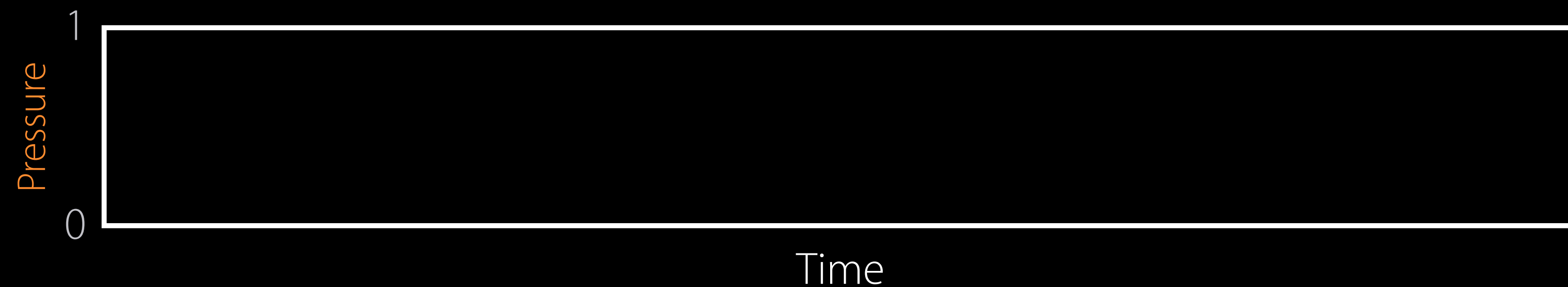
Parallel Mouse and Pressure event streams

User Action



Events Mouse:

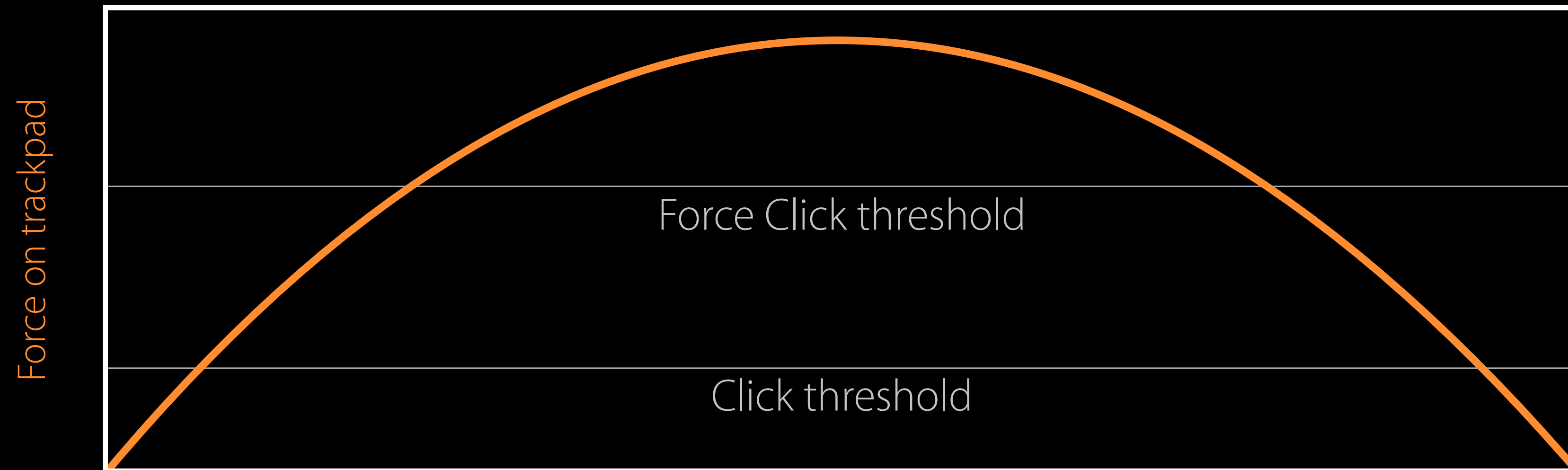
Stage:



Event Stream

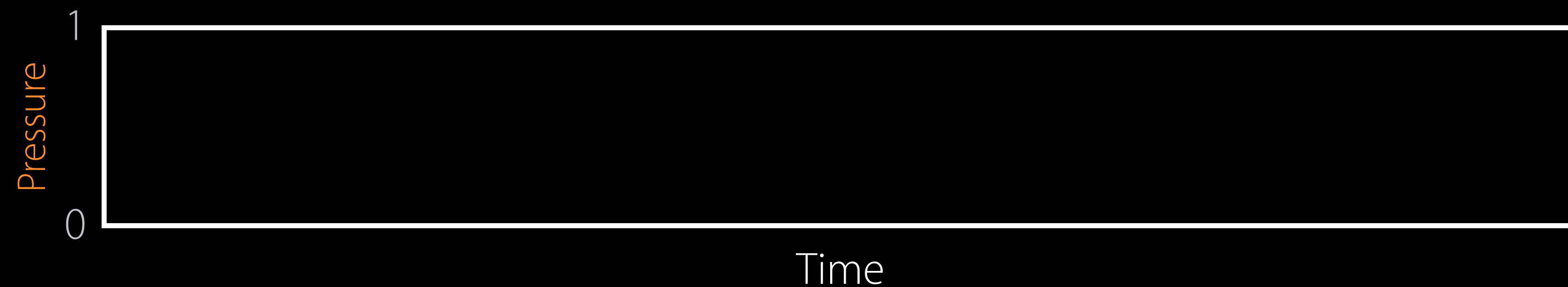
Parallel Mouse and Pressure event streams

User Action



Events Mouse:

Stage:

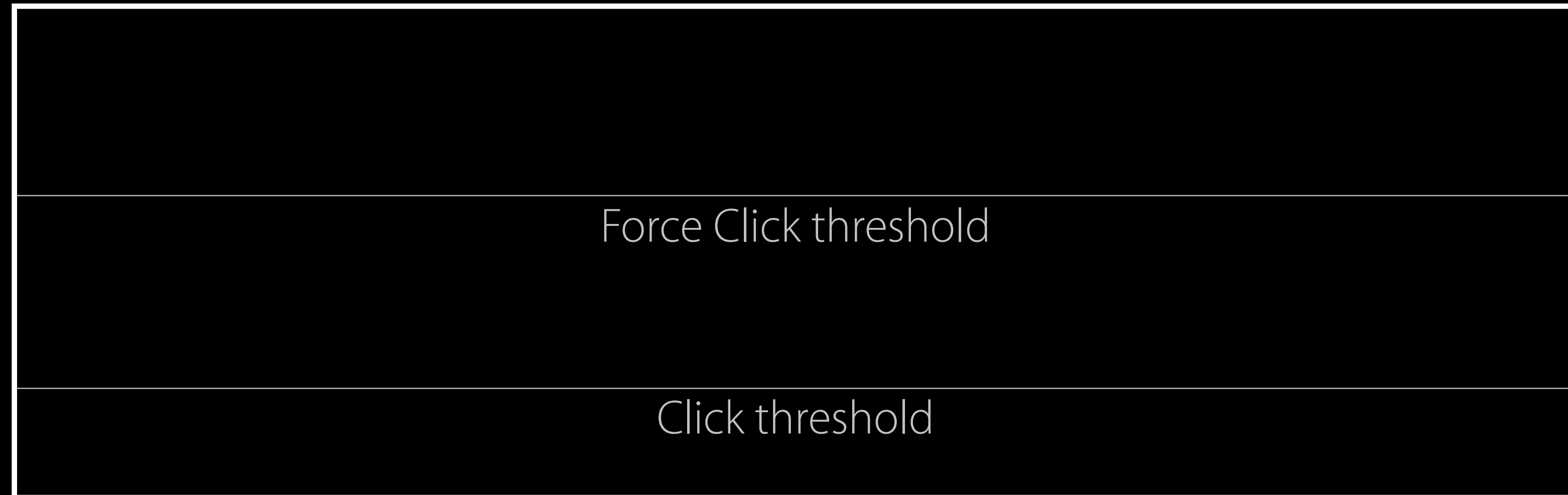


Event Stream

Parallel Mouse and Pressure event streams

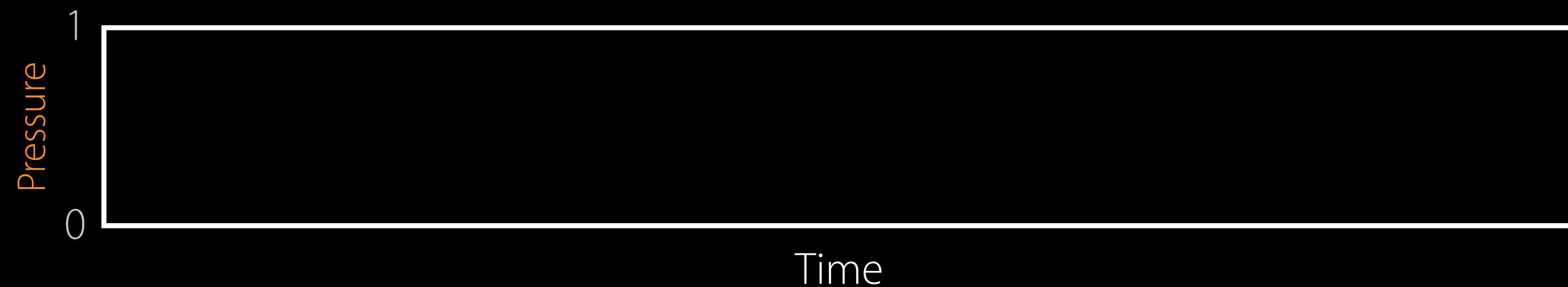
User Action

Force on trackpad



Events Mouse:

Stage:



Event Stream

Parallel Mouse and Pressure event streams



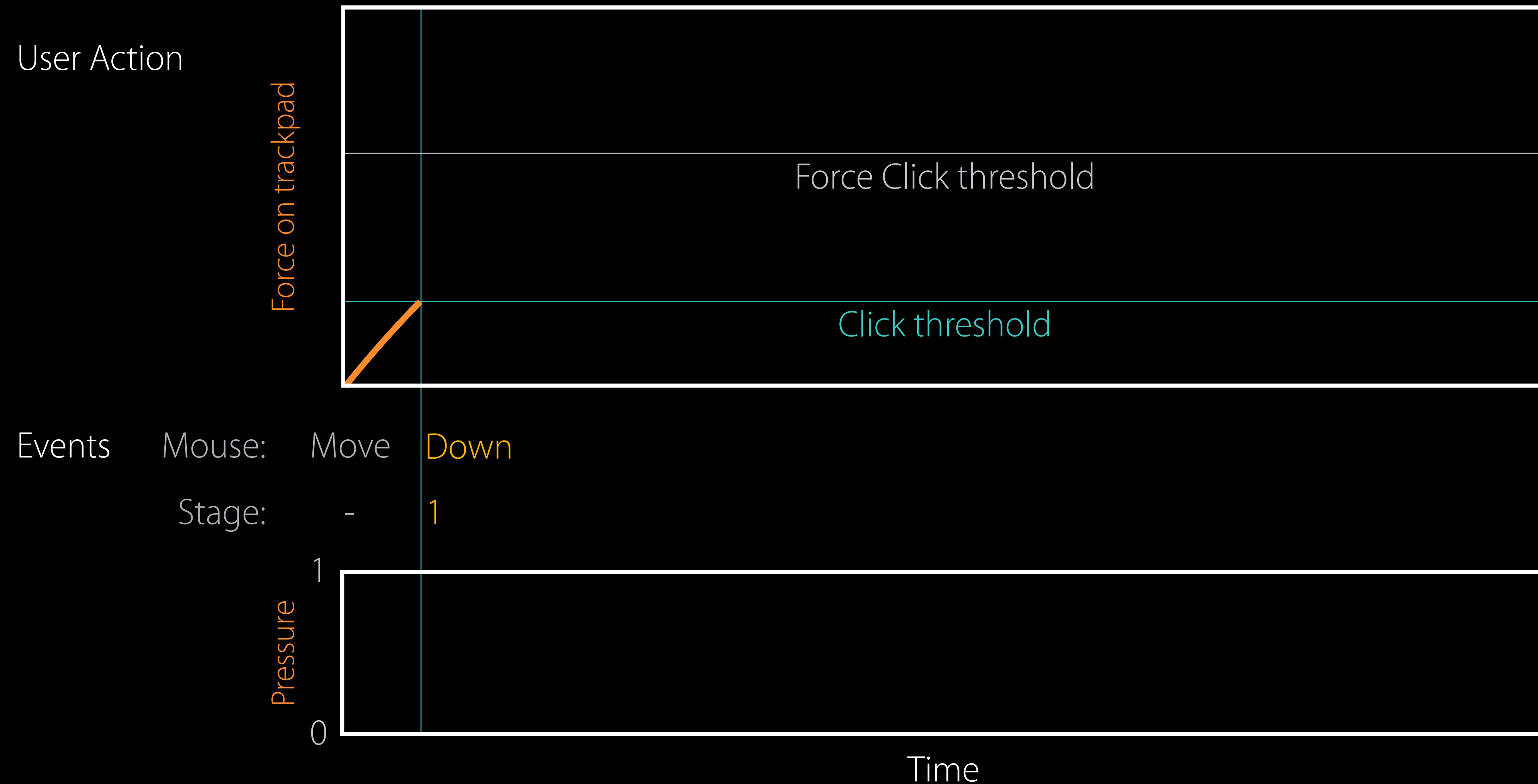
Event Stream

Parallel Mouse and Pressure event streams



Event Stream

Parallel Mouse and Pressure event streams



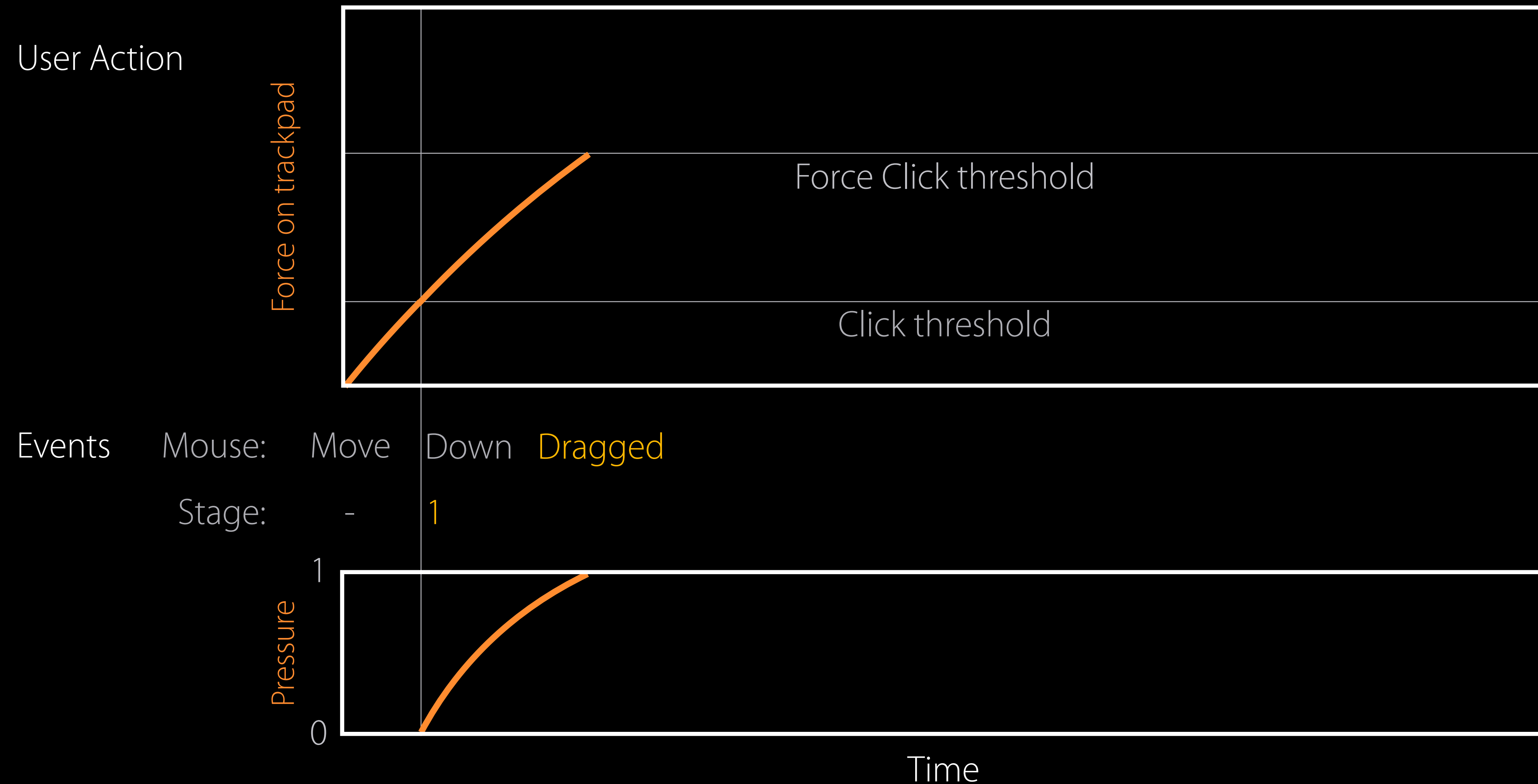
Event Stream

Parallel Mouse and Pressure event streams



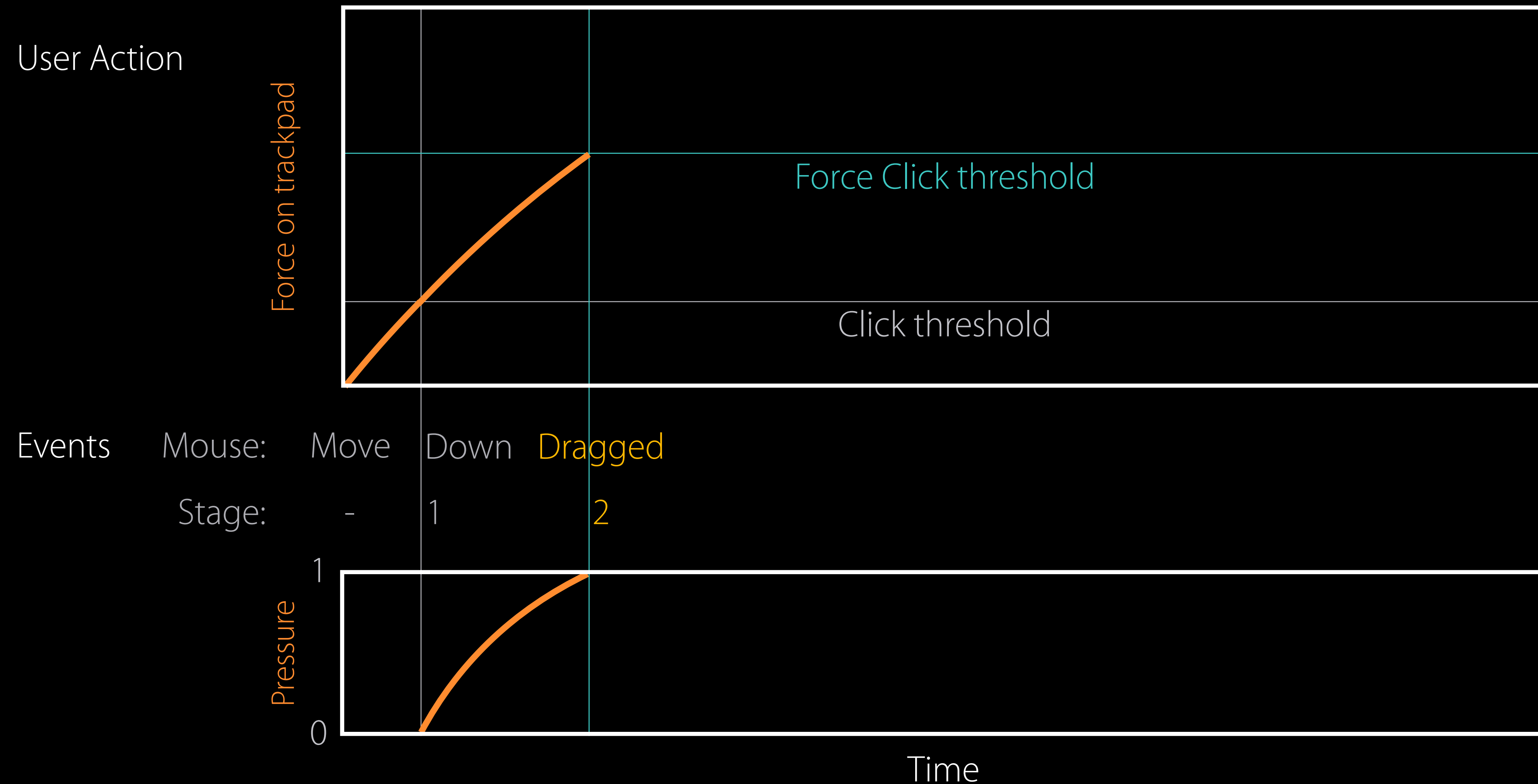
Event Stream

Parallel Mouse and Pressure event streams



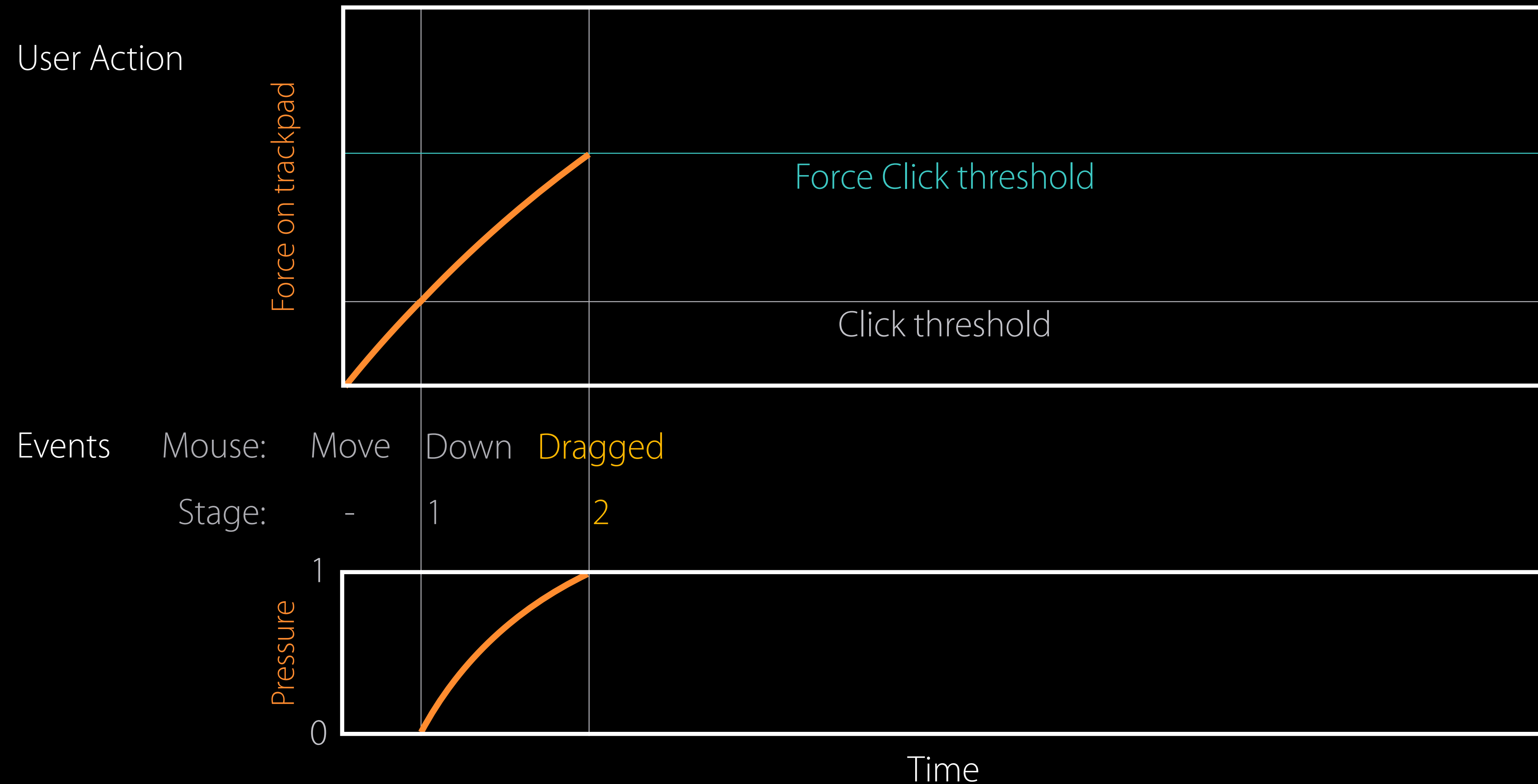
Event Stream

Parallel Mouse and Pressure event streams



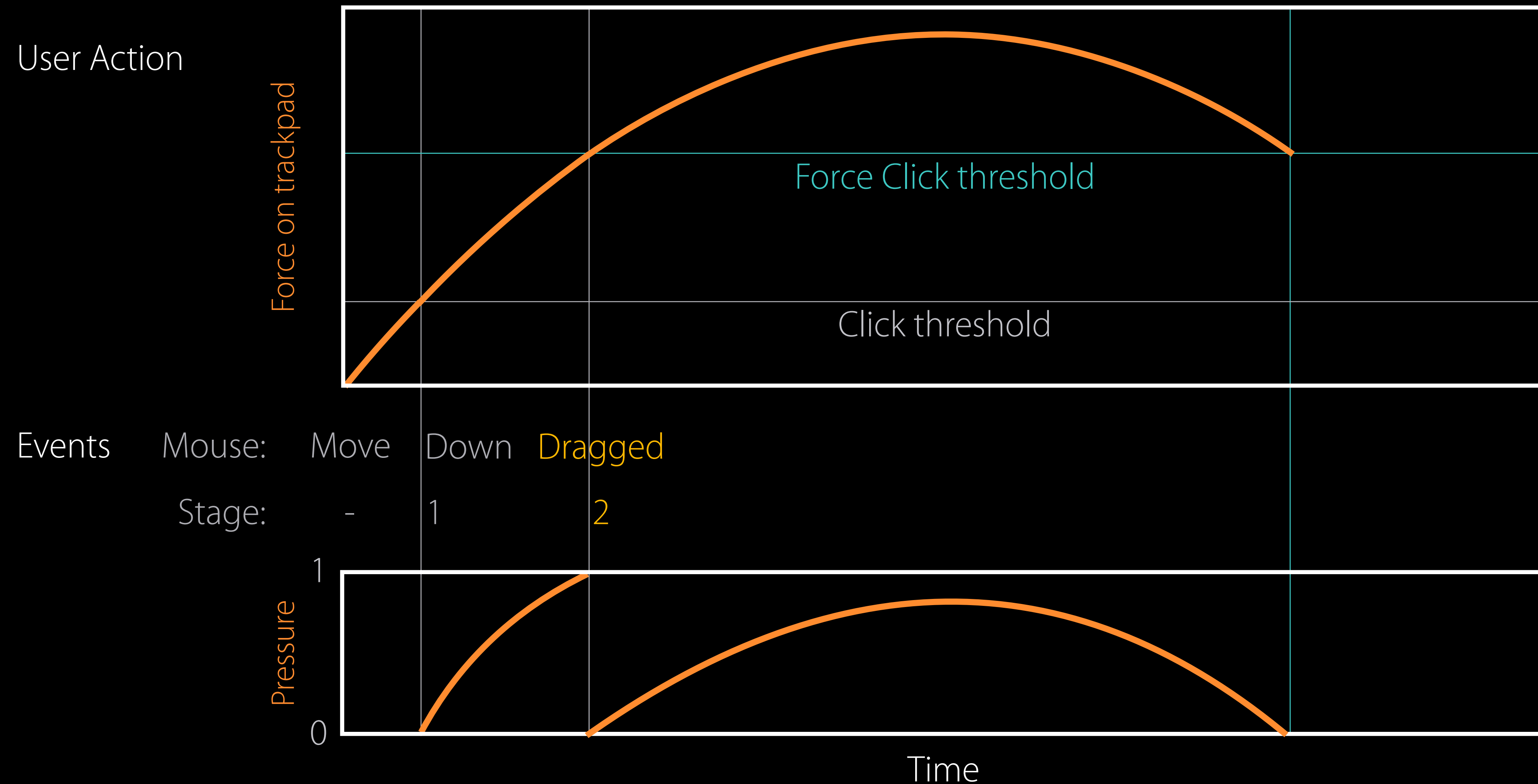
Event Stream

Parallel Mouse and Pressure event streams



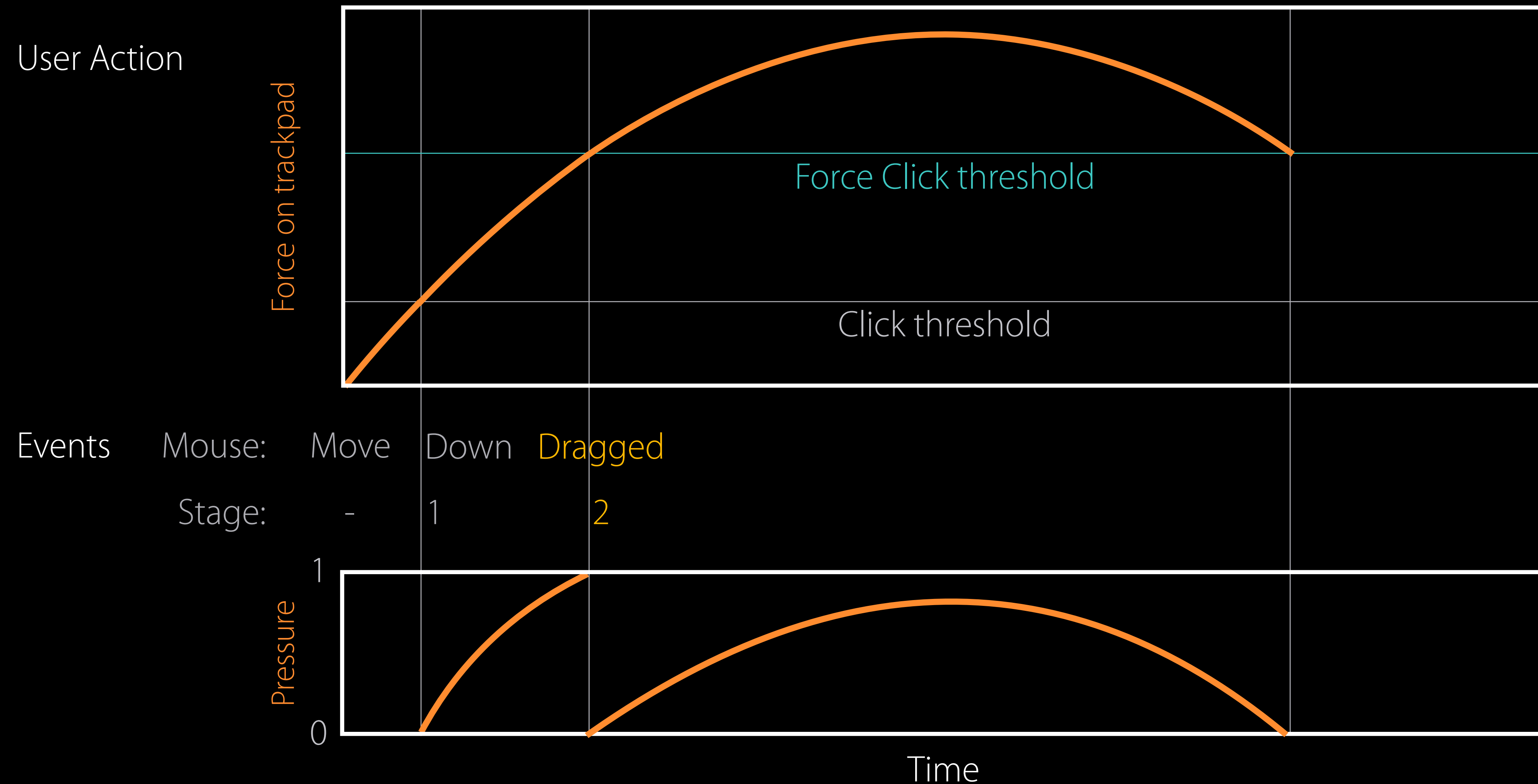
Event Stream

Parallel Mouse and Pressure event streams



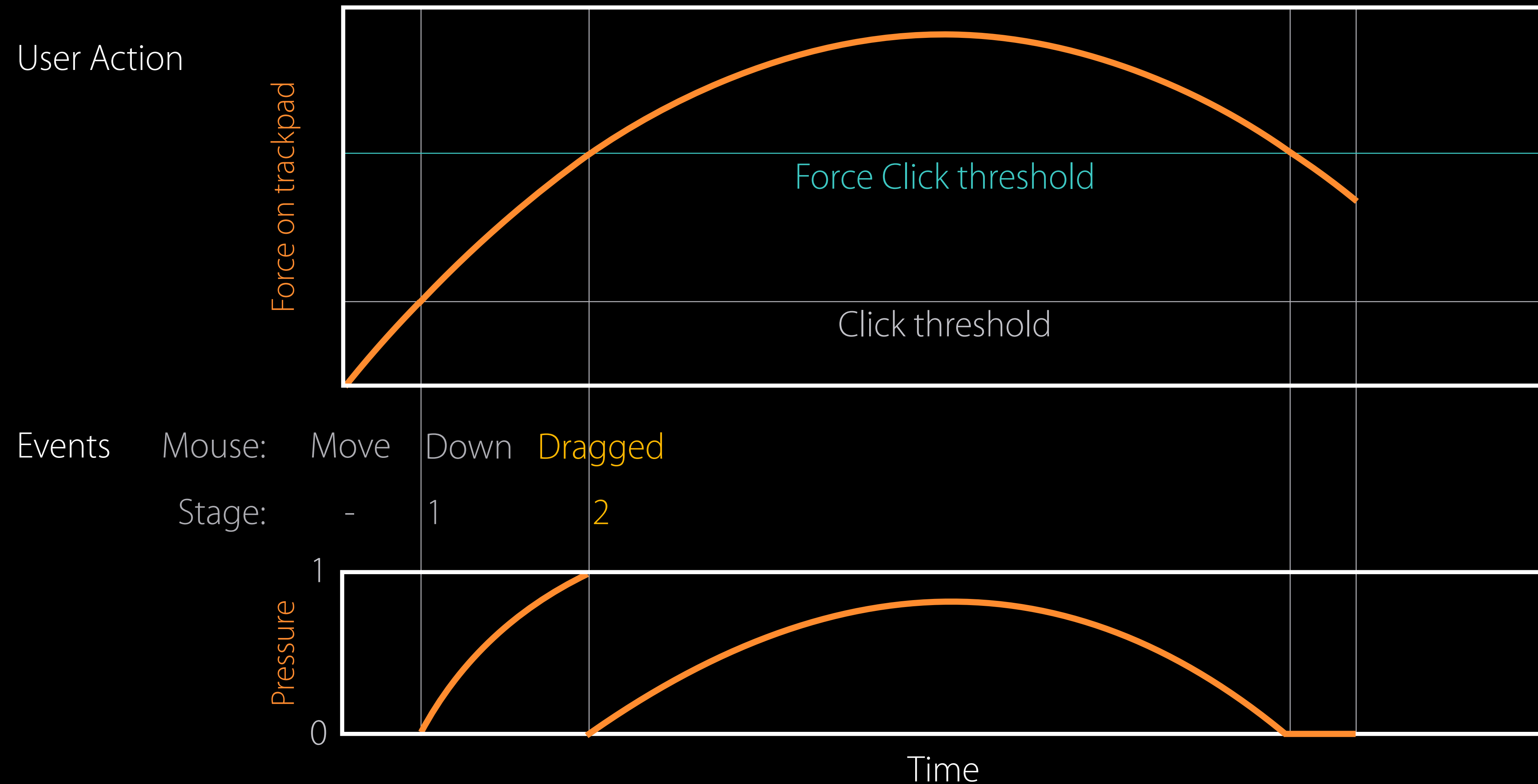
Event Stream

Parallel Mouse and Pressure event streams



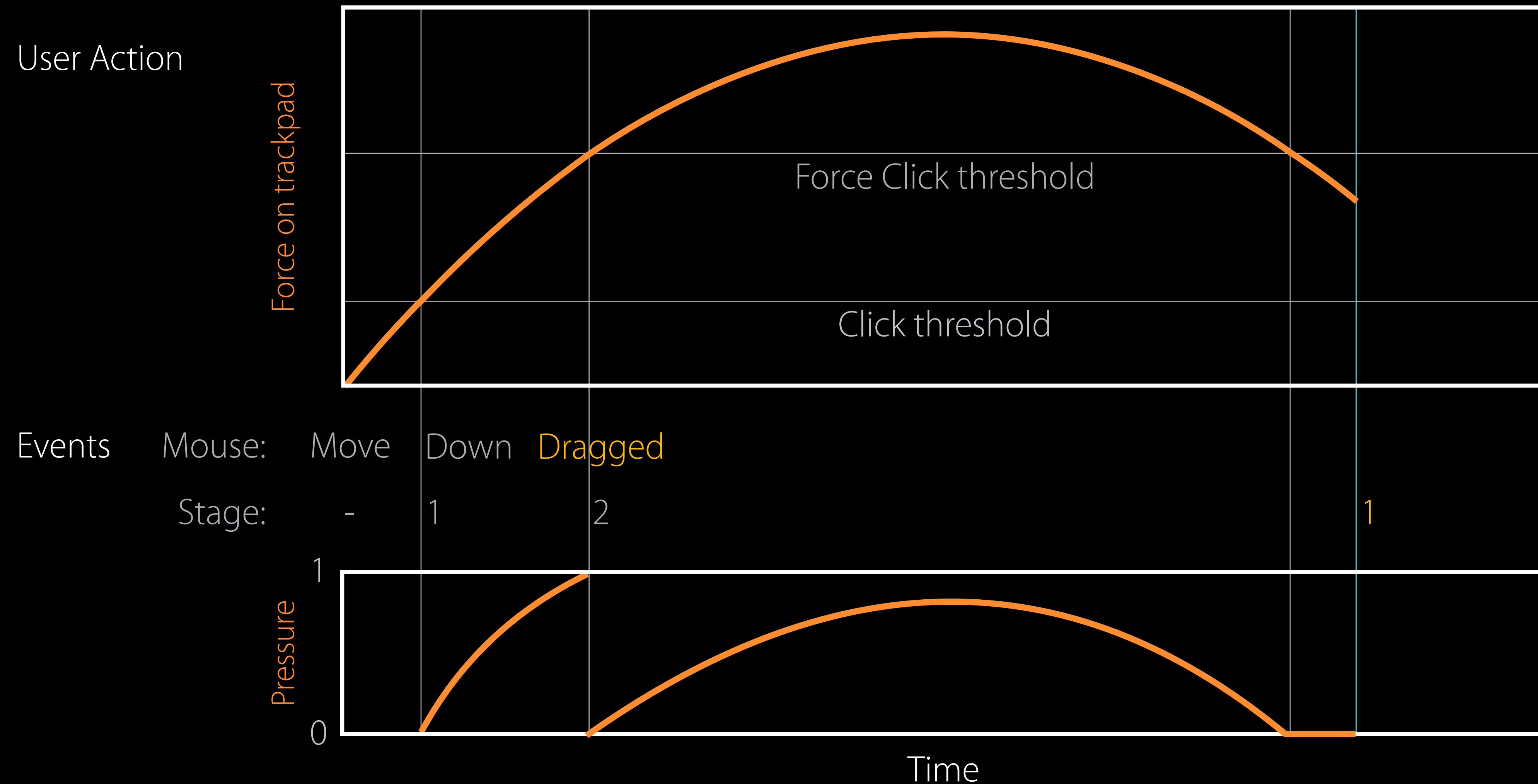
Event Stream

Parallel Mouse and Pressure event streams



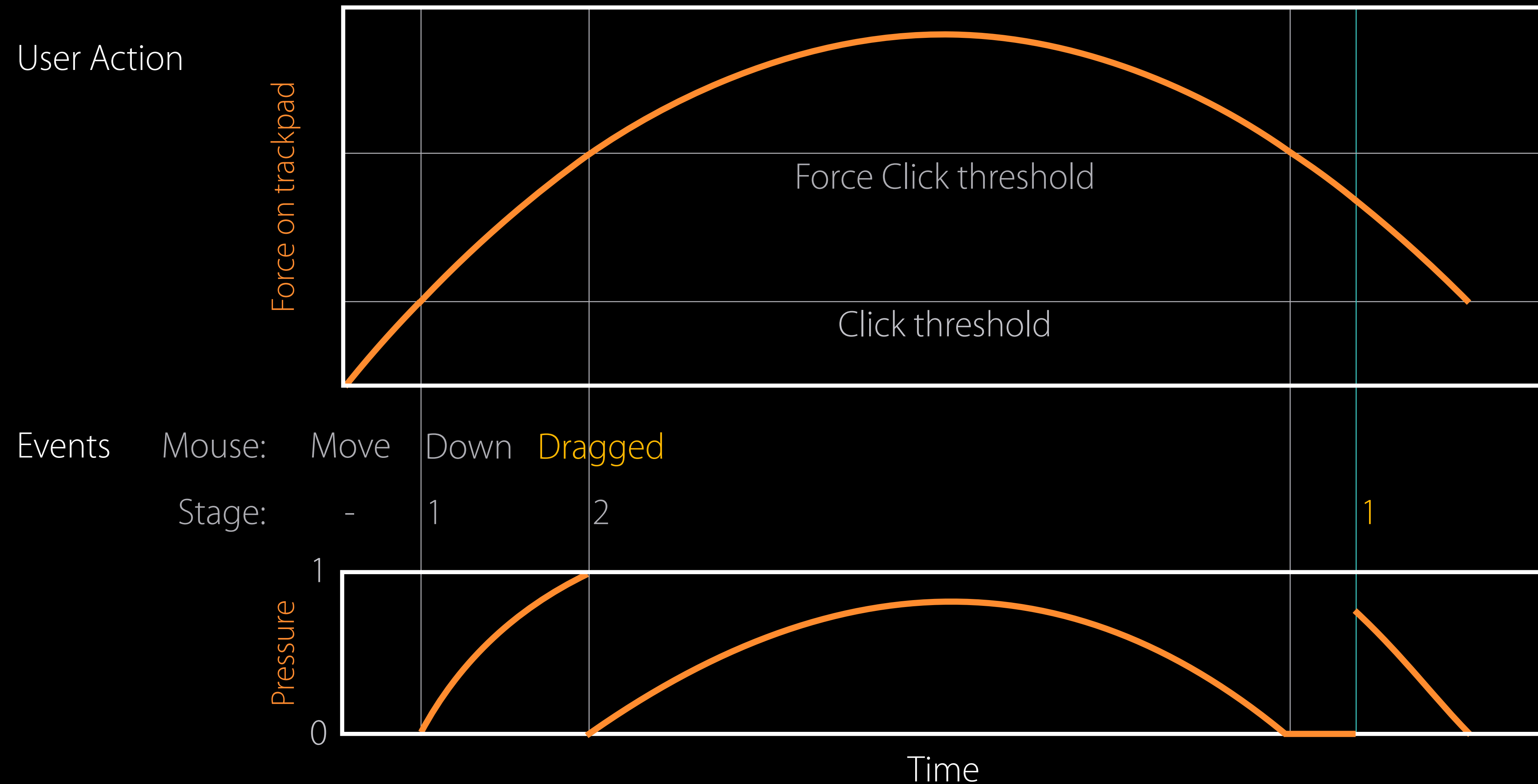
Event Stream

Parallel Mouse and Pressure event streams



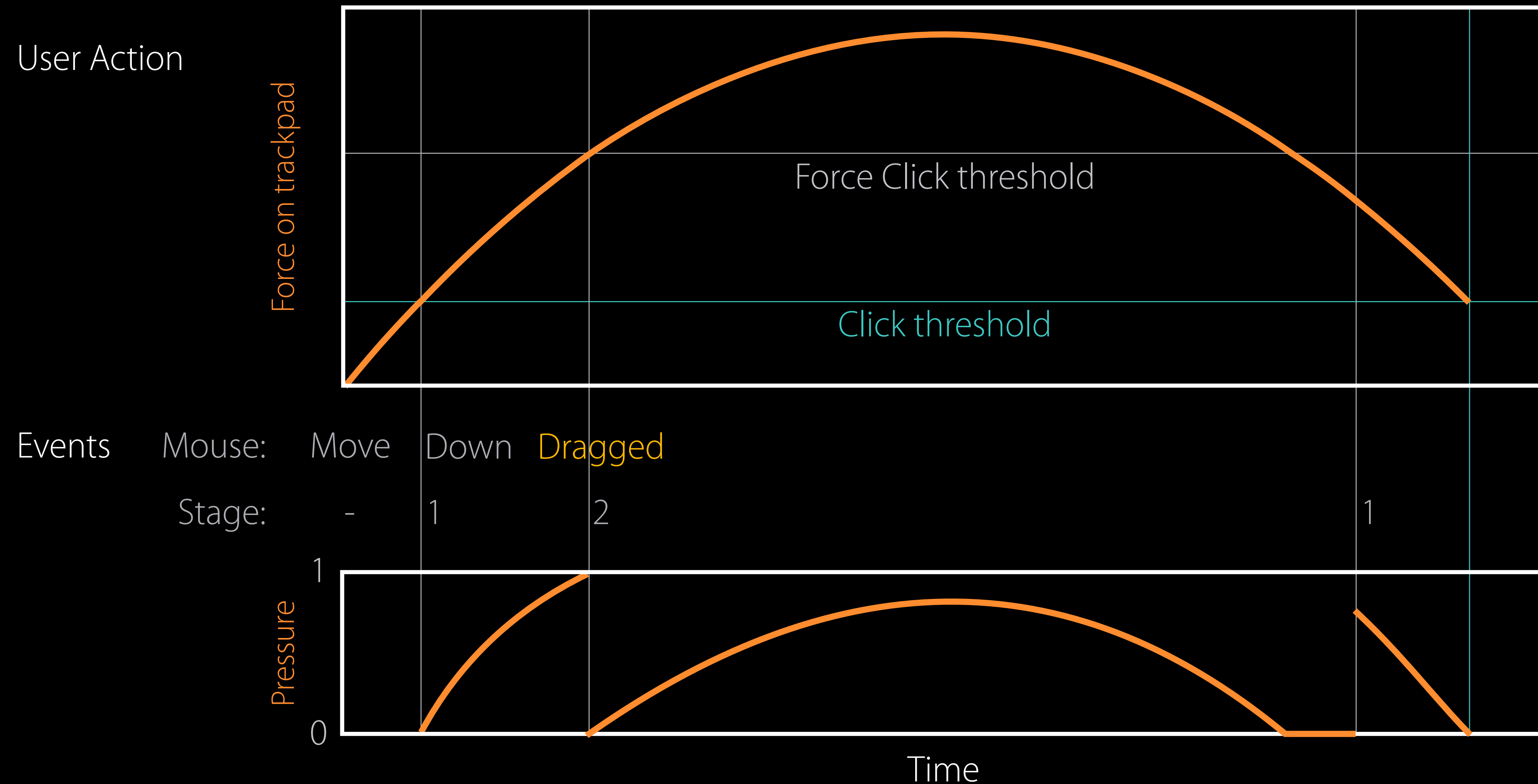
Event Stream

Parallel Mouse and Pressure event streams



Event Stream

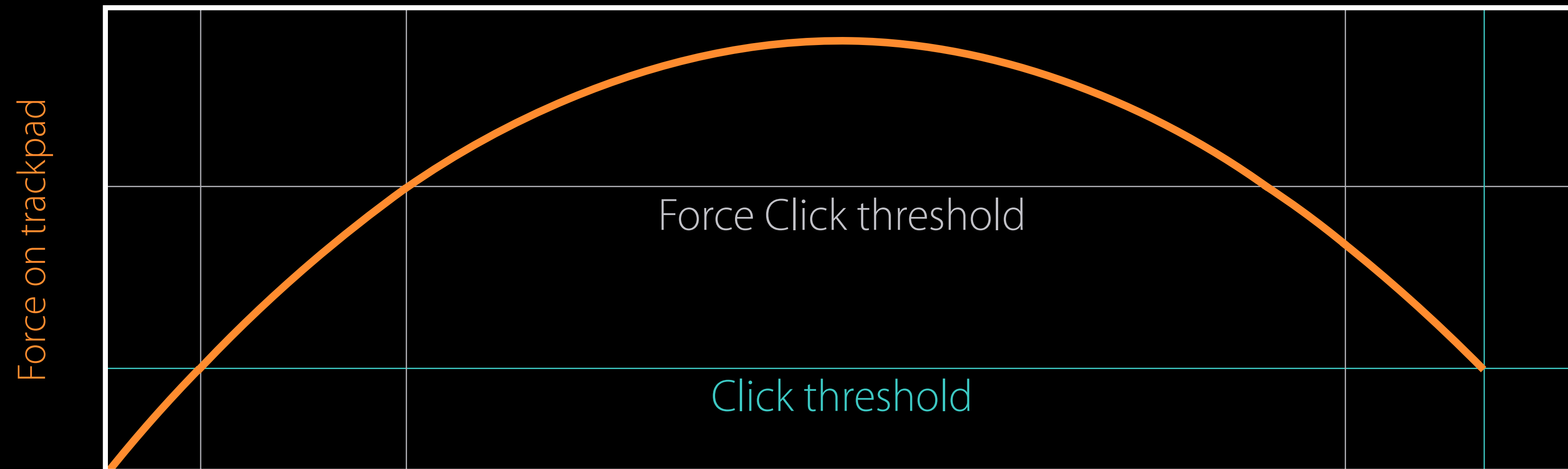
Parallel Mouse and Pressure event streams



Event Stream

Parallel Mouse and Pressure event streams

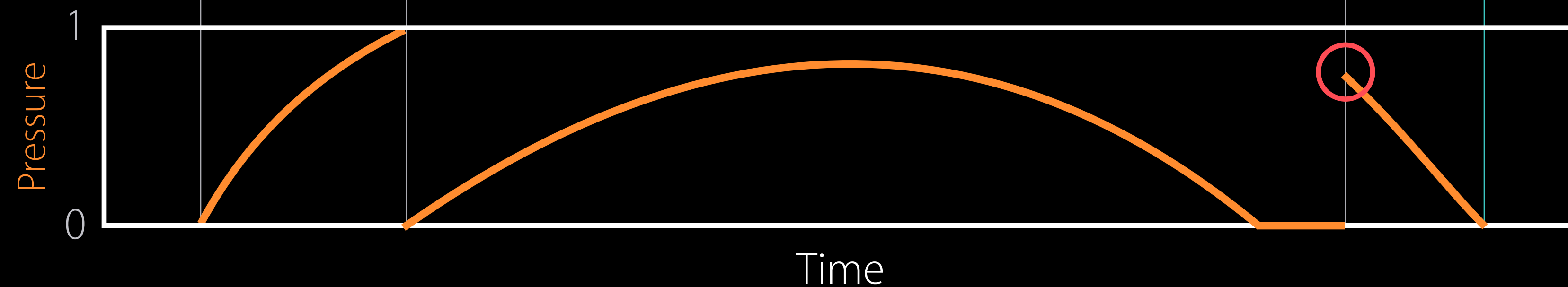
User Action



Events

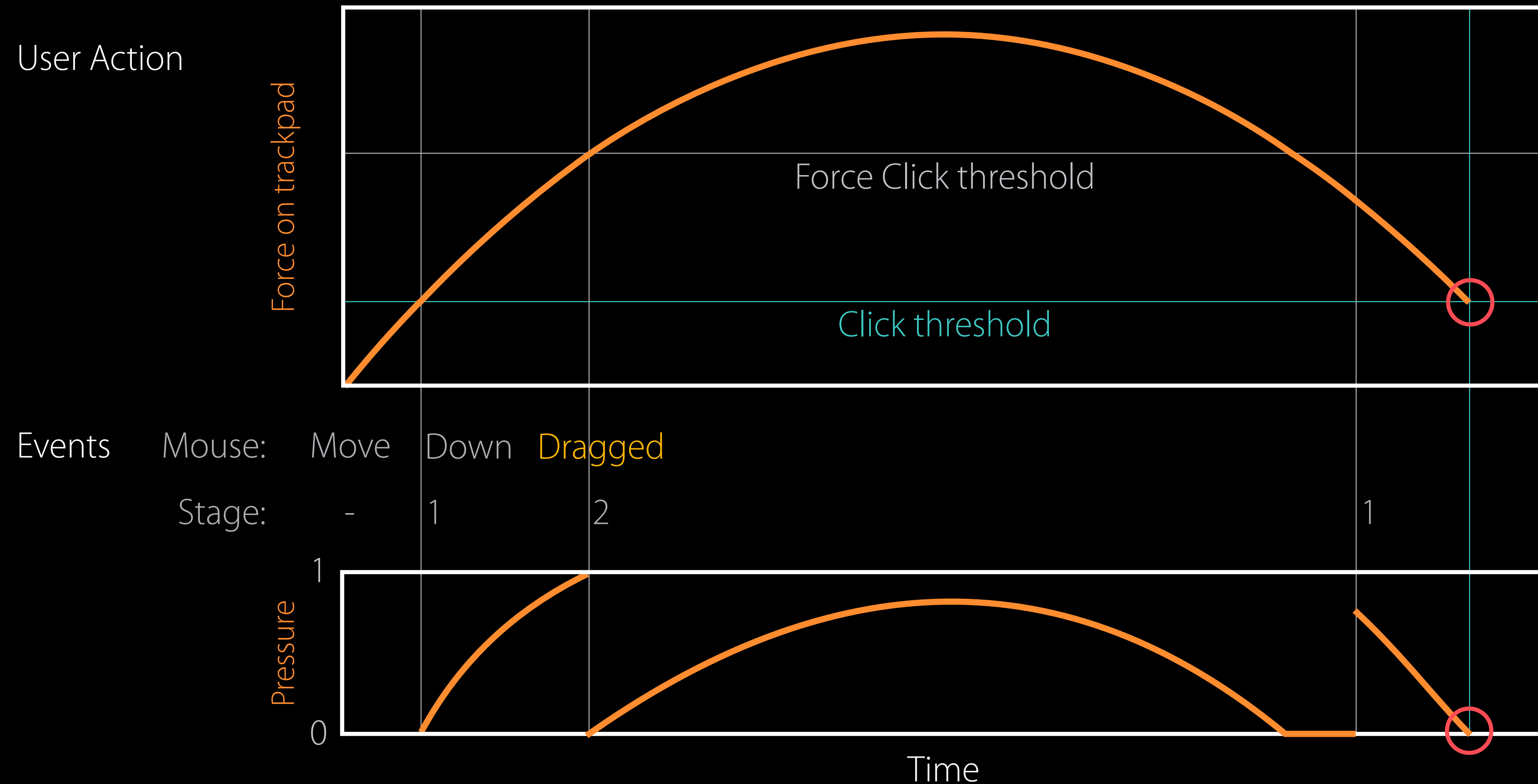
Mouse: Move Down Dragged

Stage: - 1 2 1



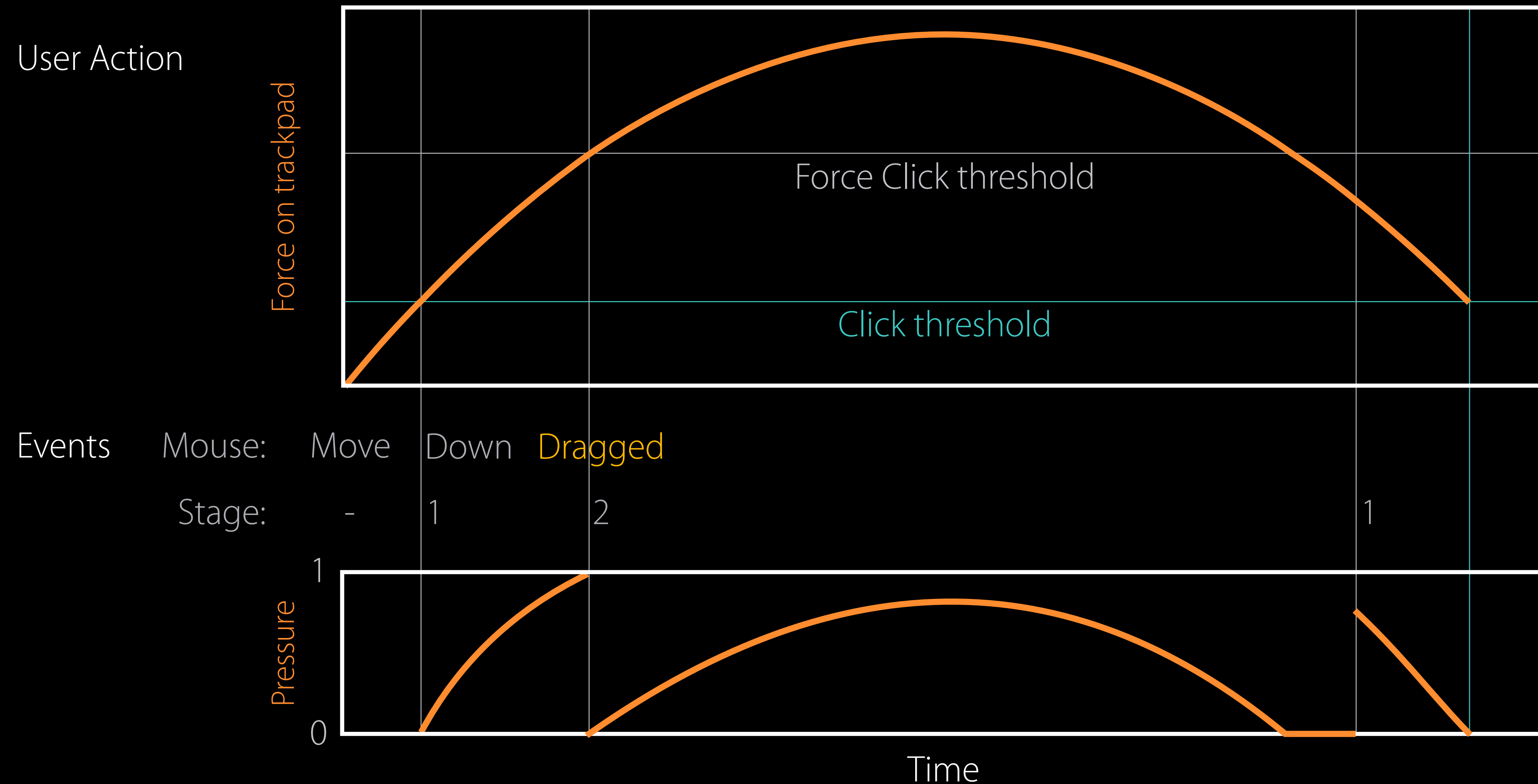
Event Stream

Parallel Mouse and Pressure event streams



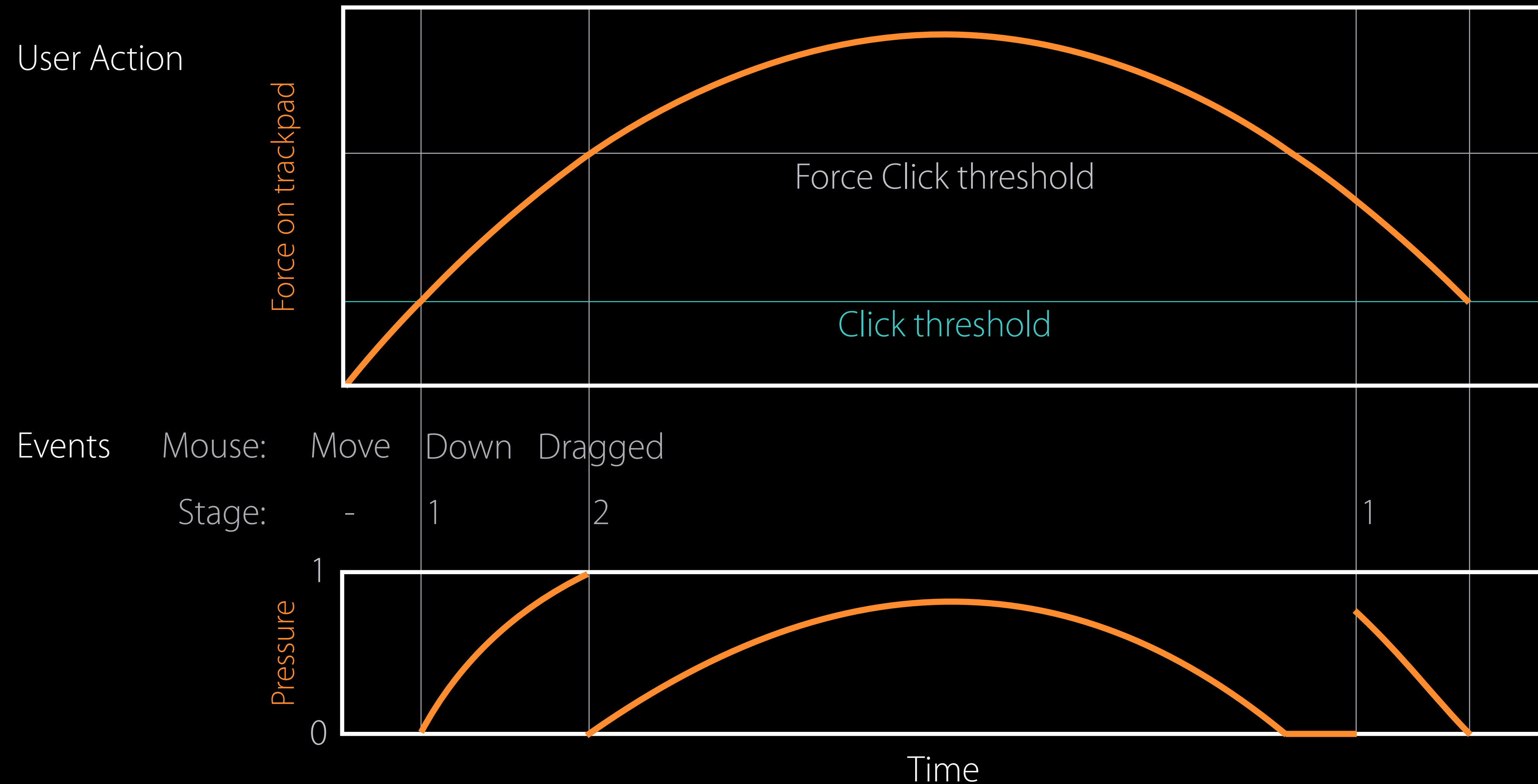
Event Stream

Parallel Mouse and Pressure event streams



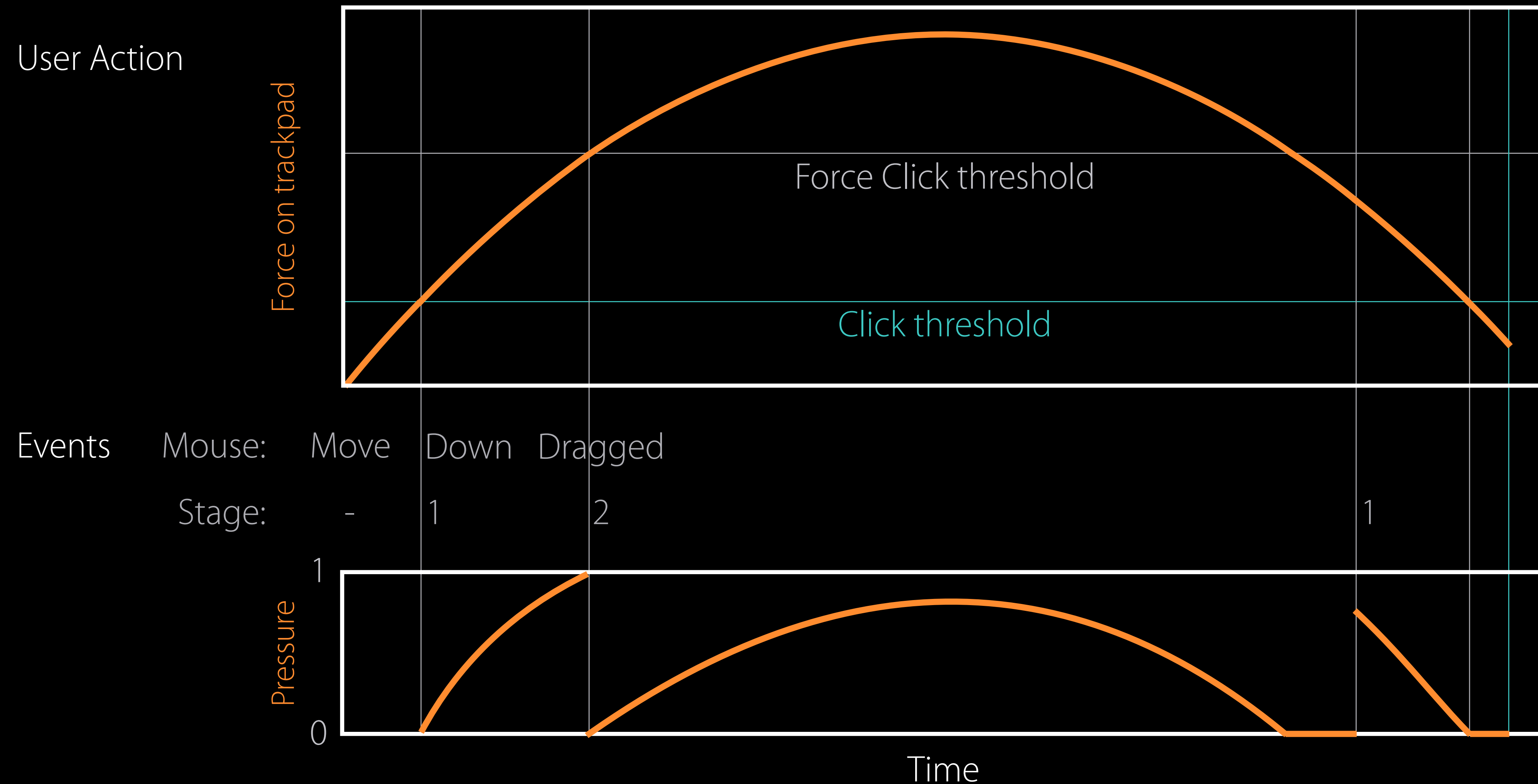
Event Stream

Parallel Mouse and Pressure event streams



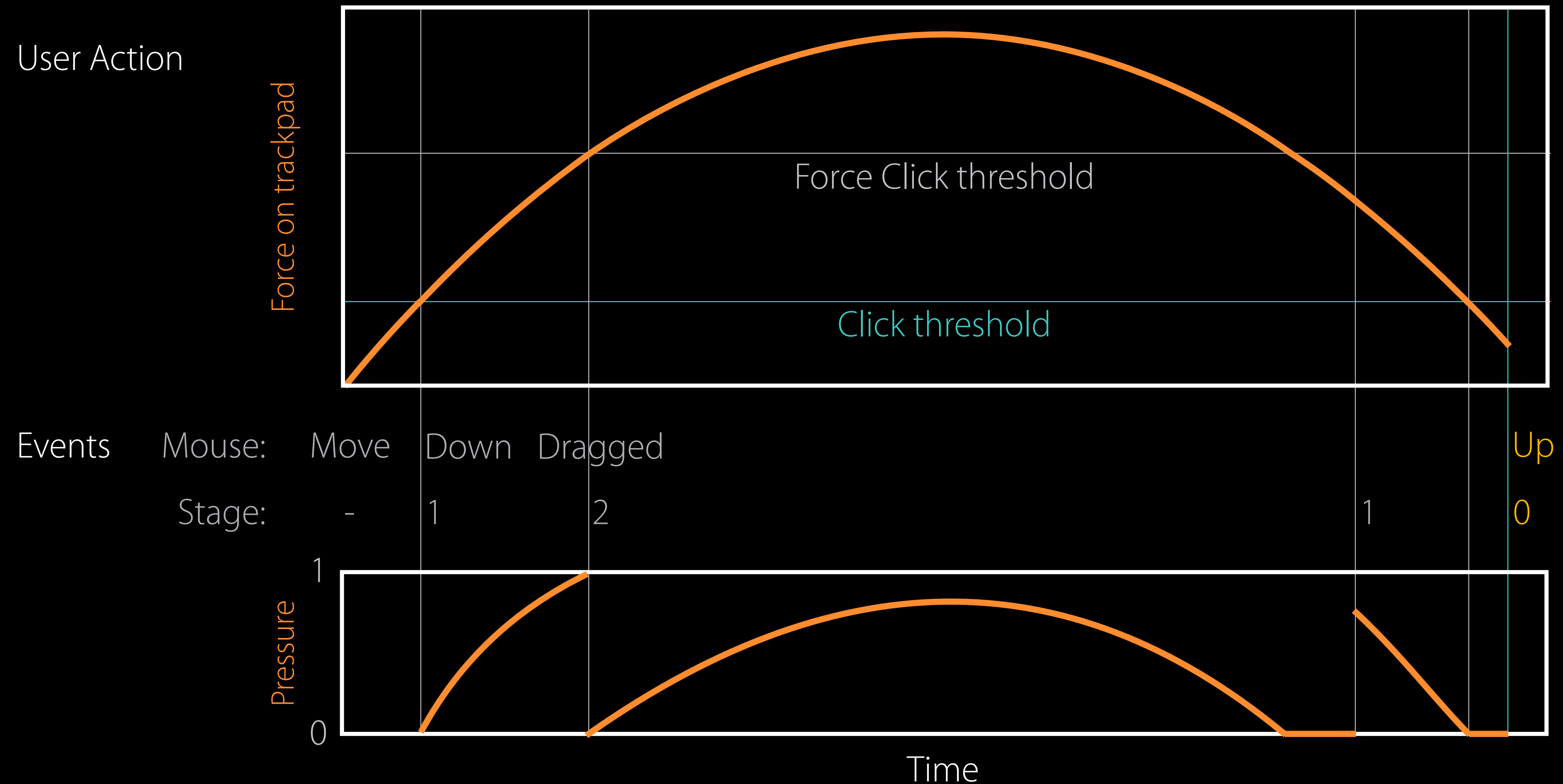
Event Stream

Parallel Mouse and Pressure event streams



Event Stream

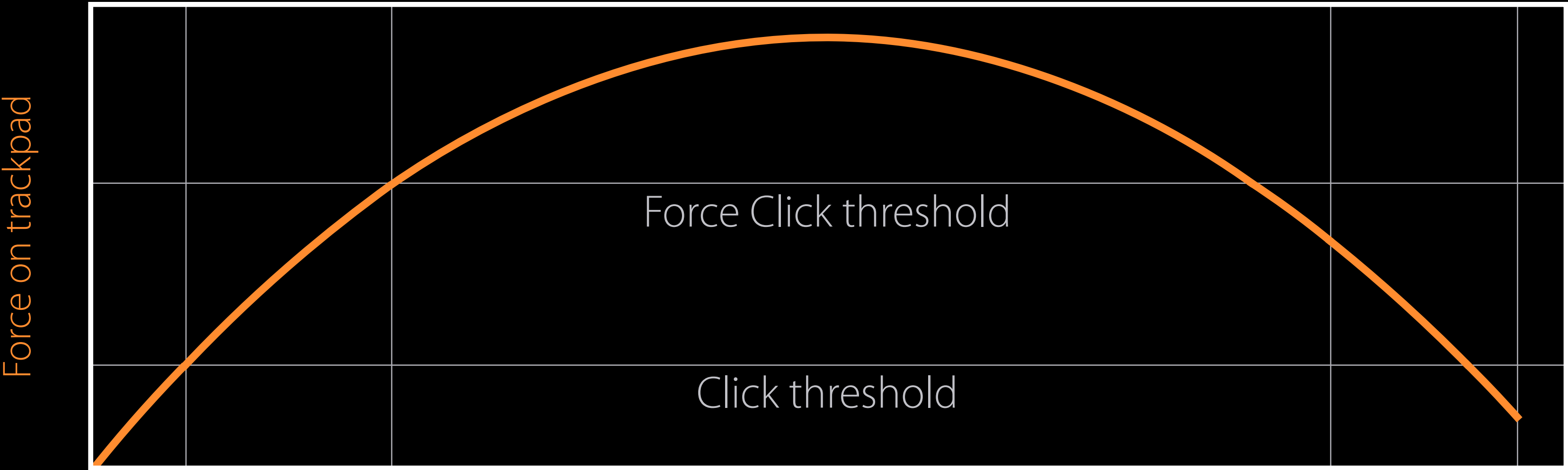
Parallel Mouse and Pressure event streams



Event Stream

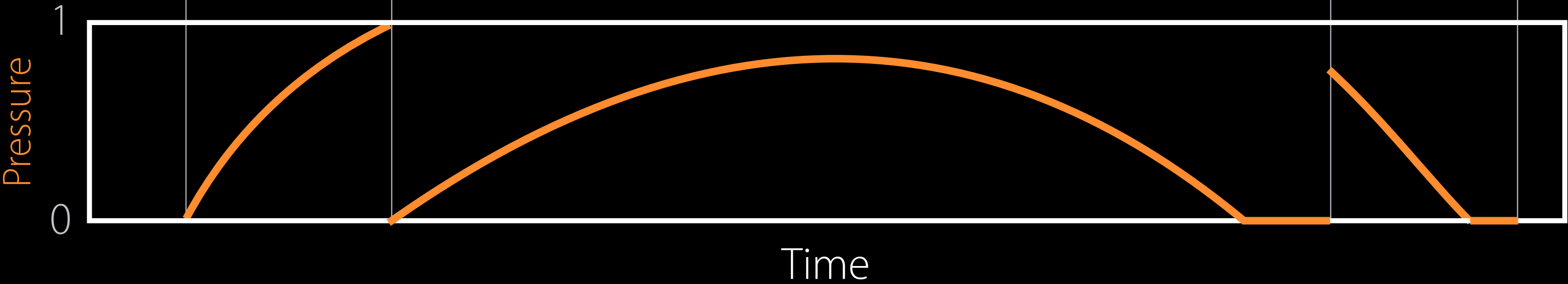
Parallel Mouse and Pressure event streams

User Action



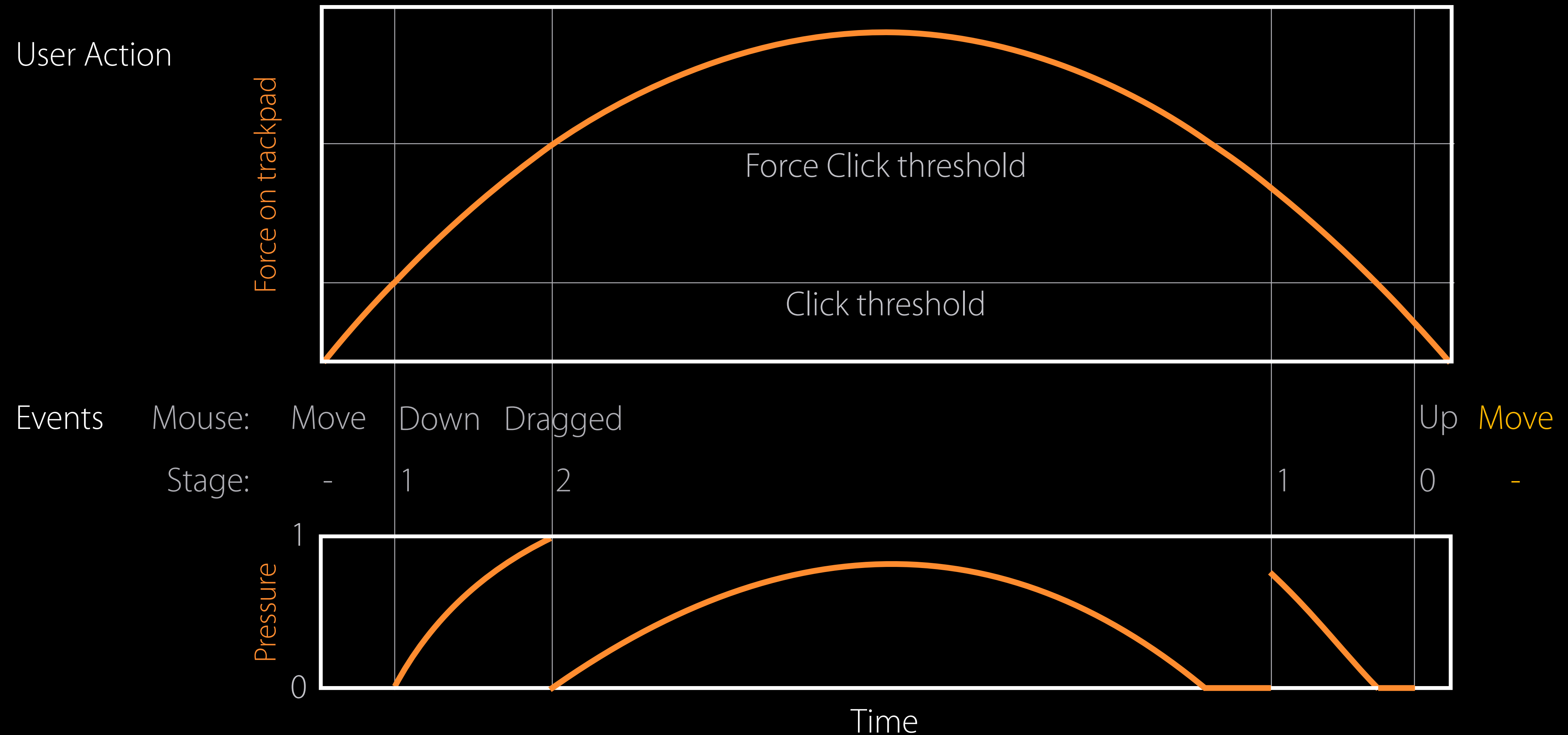
Events

Mouse:	Move	Down	Dragged				Up	Move
Stage:	-	1	2			1	0	-



Event Stream

Parallel Mouse and Pressure event streams



Event Stream

Parallel Mouse and Pressure event streams

```
var associatedEventMask: NSEventMask { get }
```

Event Stream

Parallel Mouse and Pressure event streams

```
var associatedEventMask: NSEventMask { get }

if ((mouseDown.associatedEventMask.contains(.EventMaskPressure)) {
    // Pressure Capable Device!
}
```

Event Stream

Summary

NSEventTypePressure

NSEventMaskPressure

var phase: NSEventPhase { get }

var pressure: Float { get }

var stage: Int { get }

var stageTransition: CGFloat { get }

var associatedEventMask: NSEventMask { get }

func pressureChangeWithEvent(NSEvent)

Spring Loading

```
@protocol NSSpringLoadingDestination
```

Spring Loading

```
@protocol NSSpringLoadingDestination
required (One of these)
func springLoadingEntered(NSDraggingInfo) -> NSSpringLoadingOptions
        - and / or -
func springLoadingUpdated(NSDraggingInfo) -> NSSpringLoadingOptions
```

Spring Loading

```
@protocol NSSpringLoadingDestination
```

```
required (One of these)
```

```
func springLoadingEntered(NSDraggingInfo) -> NSSpringLoadingOptions
```

```
– and / or –
```

```
func springLoadingUpdated(NSDraggingInfo) -> NSSpringLoadingOptions
```

```
.Disabled
```

```
.Enabled
```

Spring Loading

```
@protocol NSSpringLoadingDestination
```

```
required (One of these)
```

```
func springLoadingEntered(NSDraggingInfo) -> NSSpringLoadingOptions
```

```
– and / or –
```

```
func springLoadingUpdated(NSDraggingInfo) -> NSSpringLoadingOptions
```

```
.Disabled
```

```
.Enabled
```

```
.ContinuousActivation
```


Spring Loading

```
@protocol NSSpringLoadingDestination
```

```
required (One of these)
```

```
func springLoadingEntered(NSDraggingInfo) -> NSSpringLoadingOptions
```

```
– and / or –
```

```
func springLoadingUpdated(NSDraggingInfo) -> NSSpringLoadingOptions
```

```
.Disabled
```

```
.Enabled
```

```
.ContinuousActivation
```

```
.NoHover
```

Spring Loading

```
@protocol NSSpringLoadingDestination
required (One of these)
func springLoadingEntered(NSDraggingInfo) -> NSSpringLoadingOptions
        - and / or -
func springLoadingUpdated(NSDraggingInfo) -> NSSpringLoadingOptions

@optional
func springLoadingExited(NSDraggingInfo)
func draggingEnded(NSDraggingInfo)
```

Spring Loading

```
@protocol NSSpringLoadingDestination
@required
func springLoadingActivated(Bool, draggingInfo: NSDraggingInfo)
func springLoadingHighlightChanged(NSDraggingInfo)
```

Spring Loading

```
@protocol NSSpringLoadingDestination
@required
func springLoadingActivated(Bool, draggingInfo: NSDraggingInfo)
func springLoadingHighlightChanged(NSDraggingInfo)
```

```
draggingInfo.springLoadingHighlight
```

```
NSSpringLoadingHighlight
```

```
.None
```

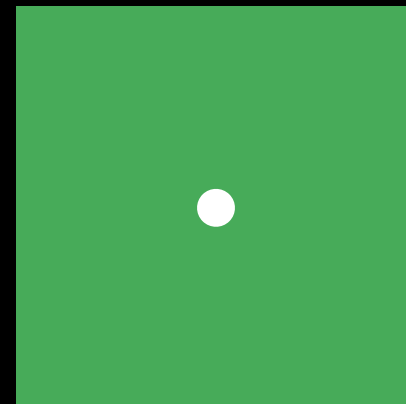
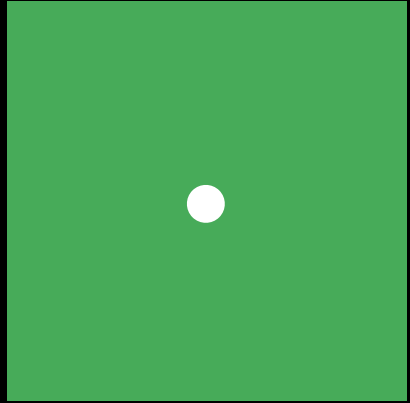
```
.Standard
```

```
.Emphasized
```

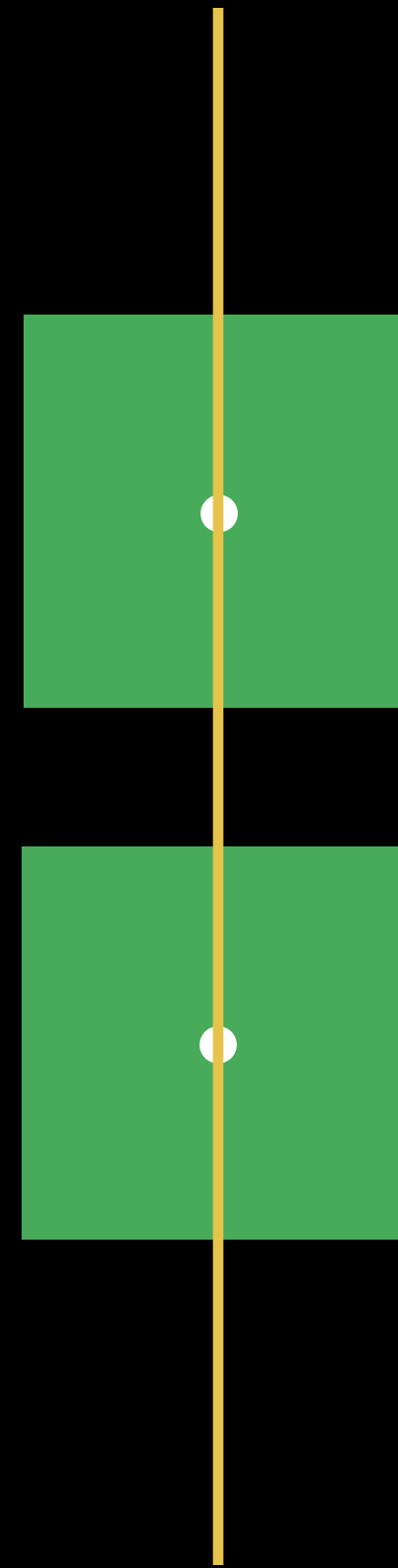

Spring Loading

NSSpringLoadingDestination

Alignment Feedback



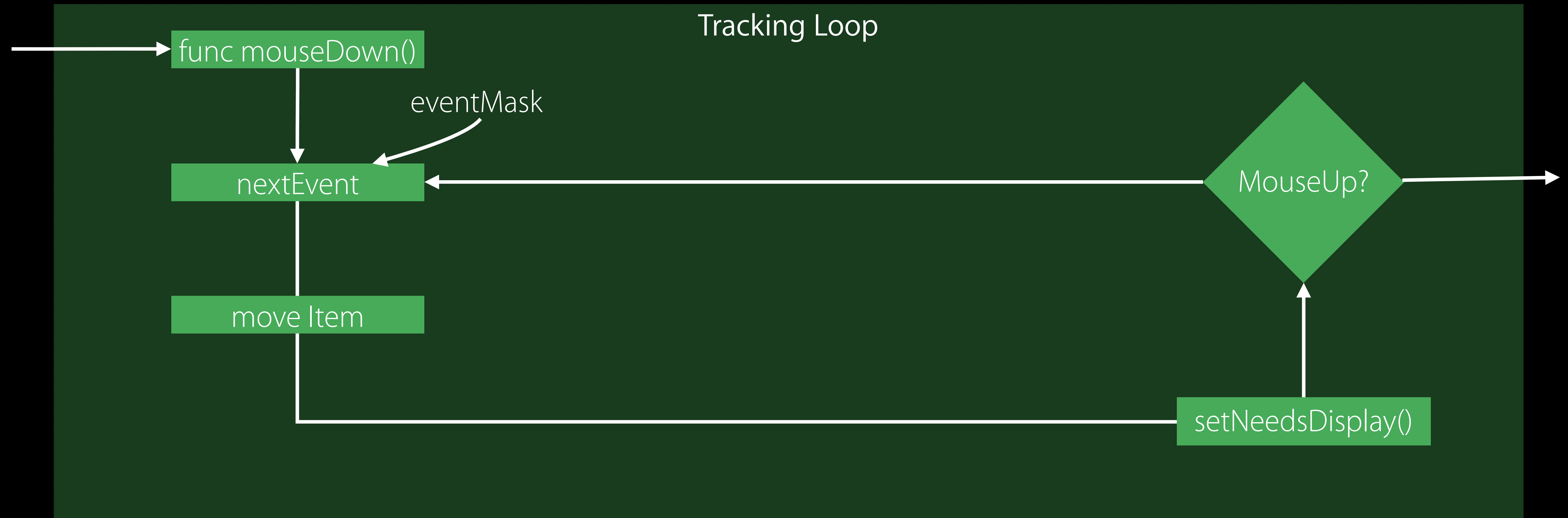
Alignment Feedback



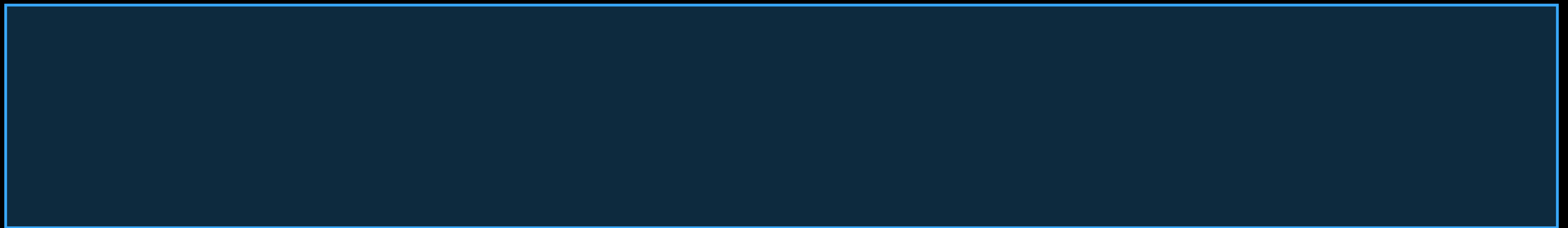
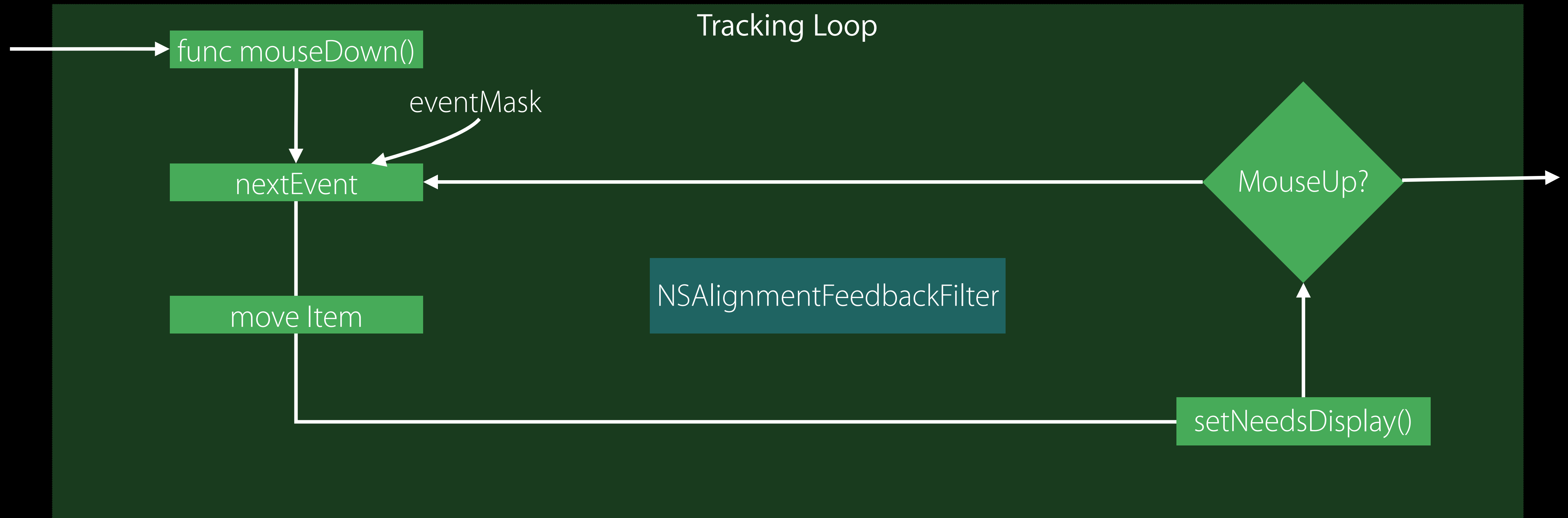
Alignment Feedback

Tracking Loop

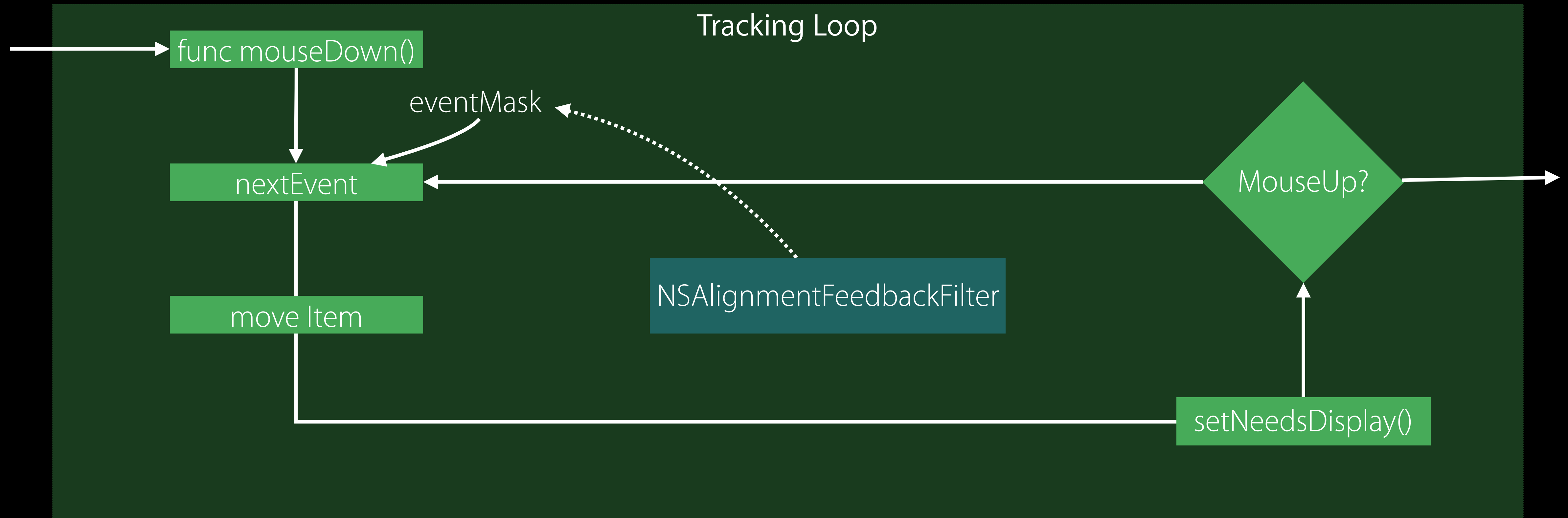
Alignment Feedback



Alignment Feedback

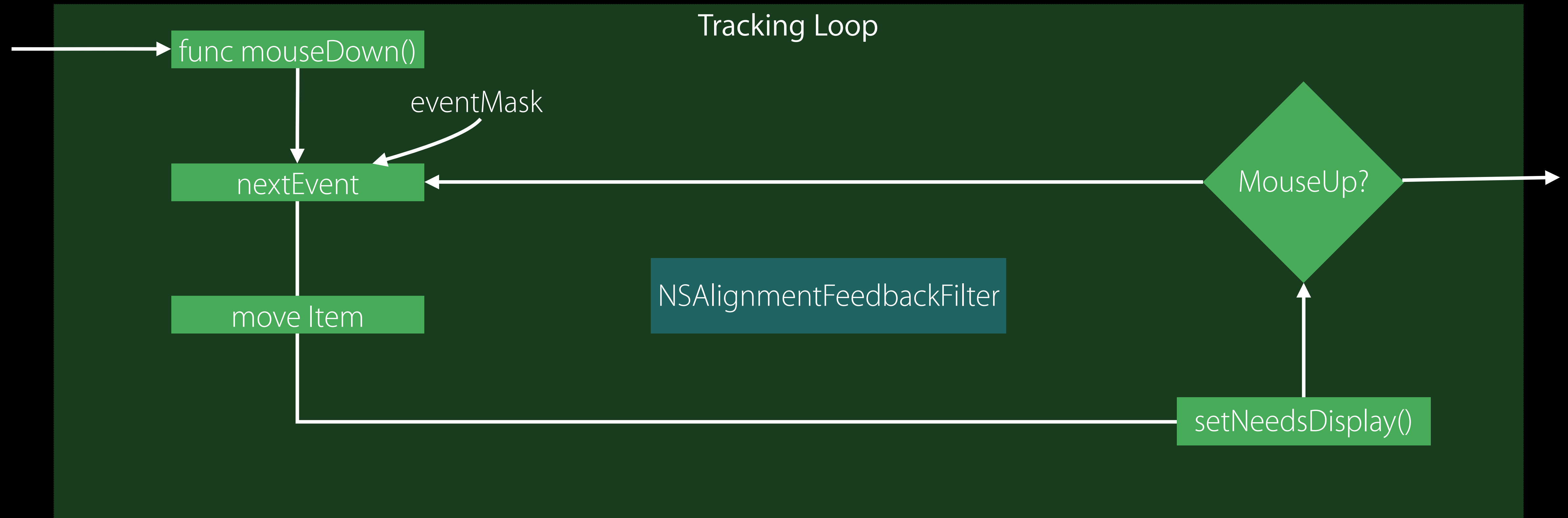


Alignment Feedback

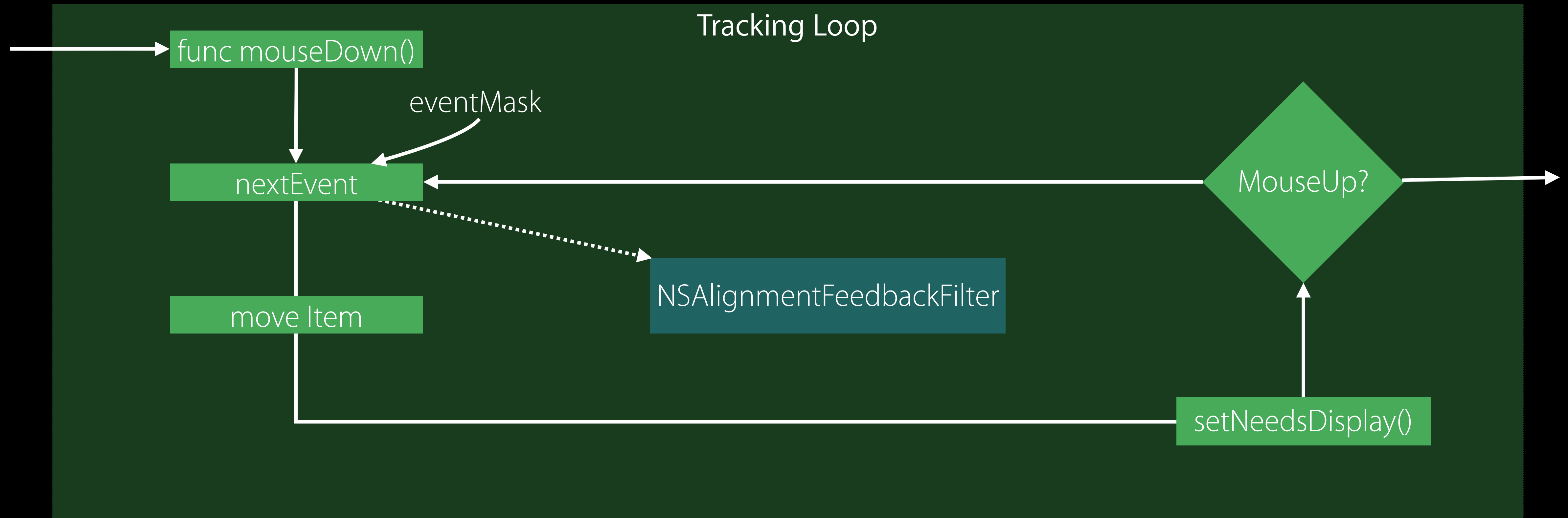


```
class func inputEventMask() -> NSEventMask
```

Alignment Feedback

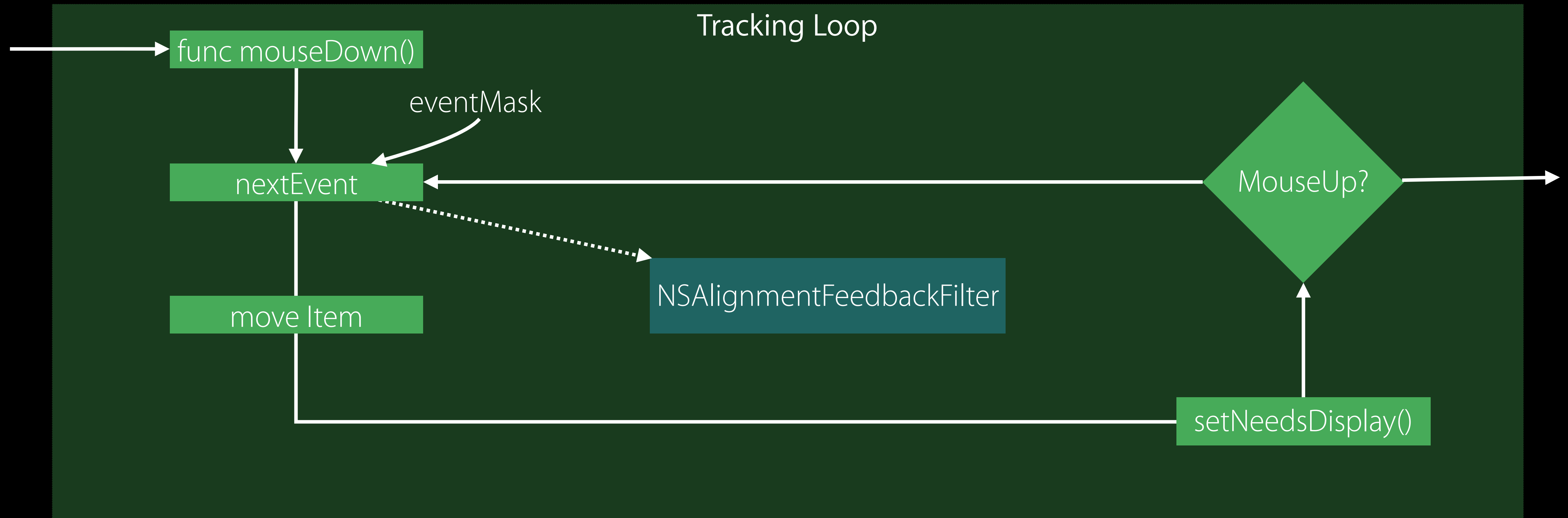


Alignment Feedback



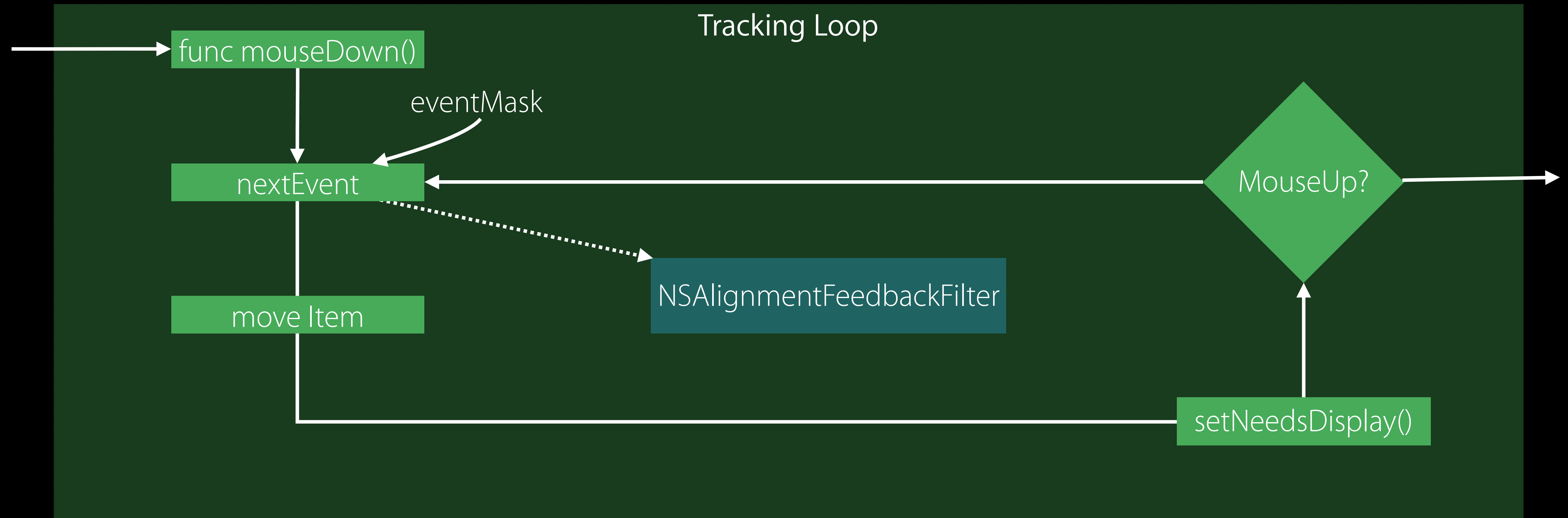
```
func updateWithEvent(NSEvent)
```

Alignment Feedback

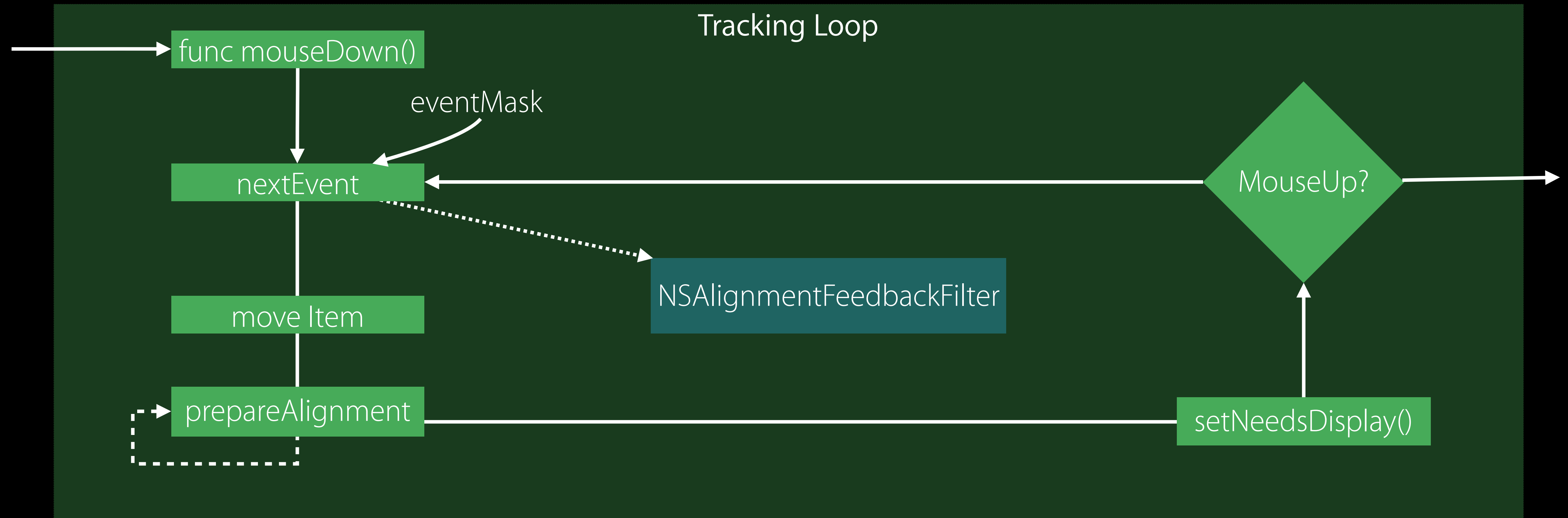


```
func updateWithPanRecognizer(NSPanGestureRecognizer)
```

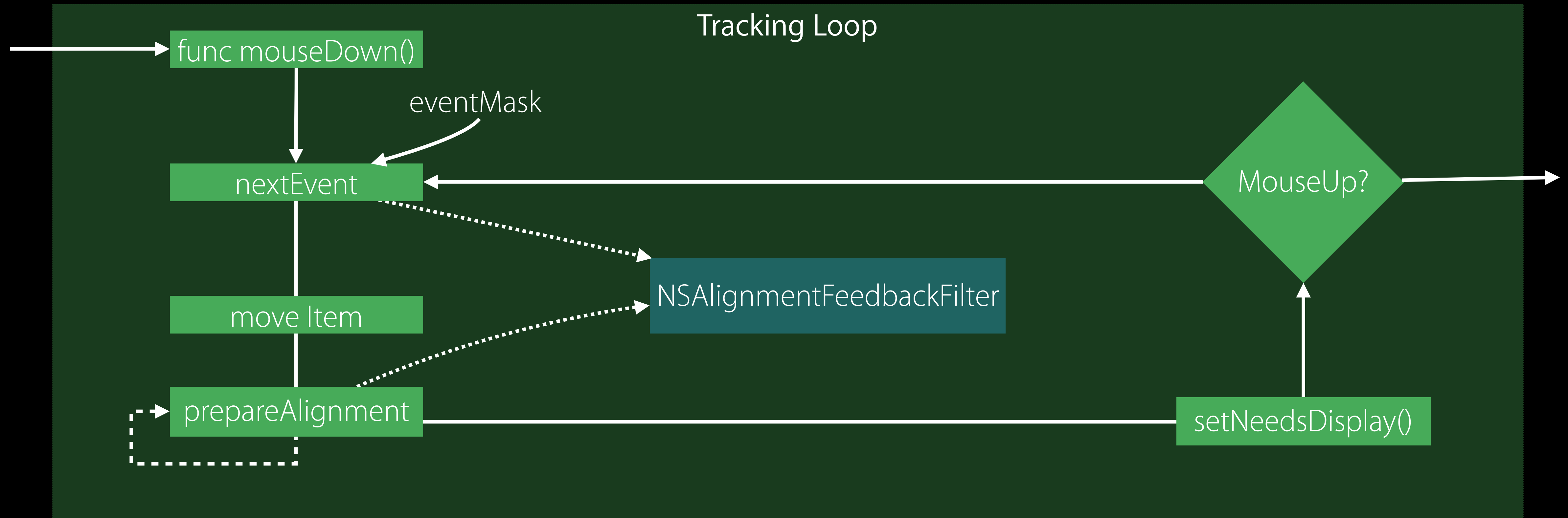
Alignment Feedback



Alignment Feedback

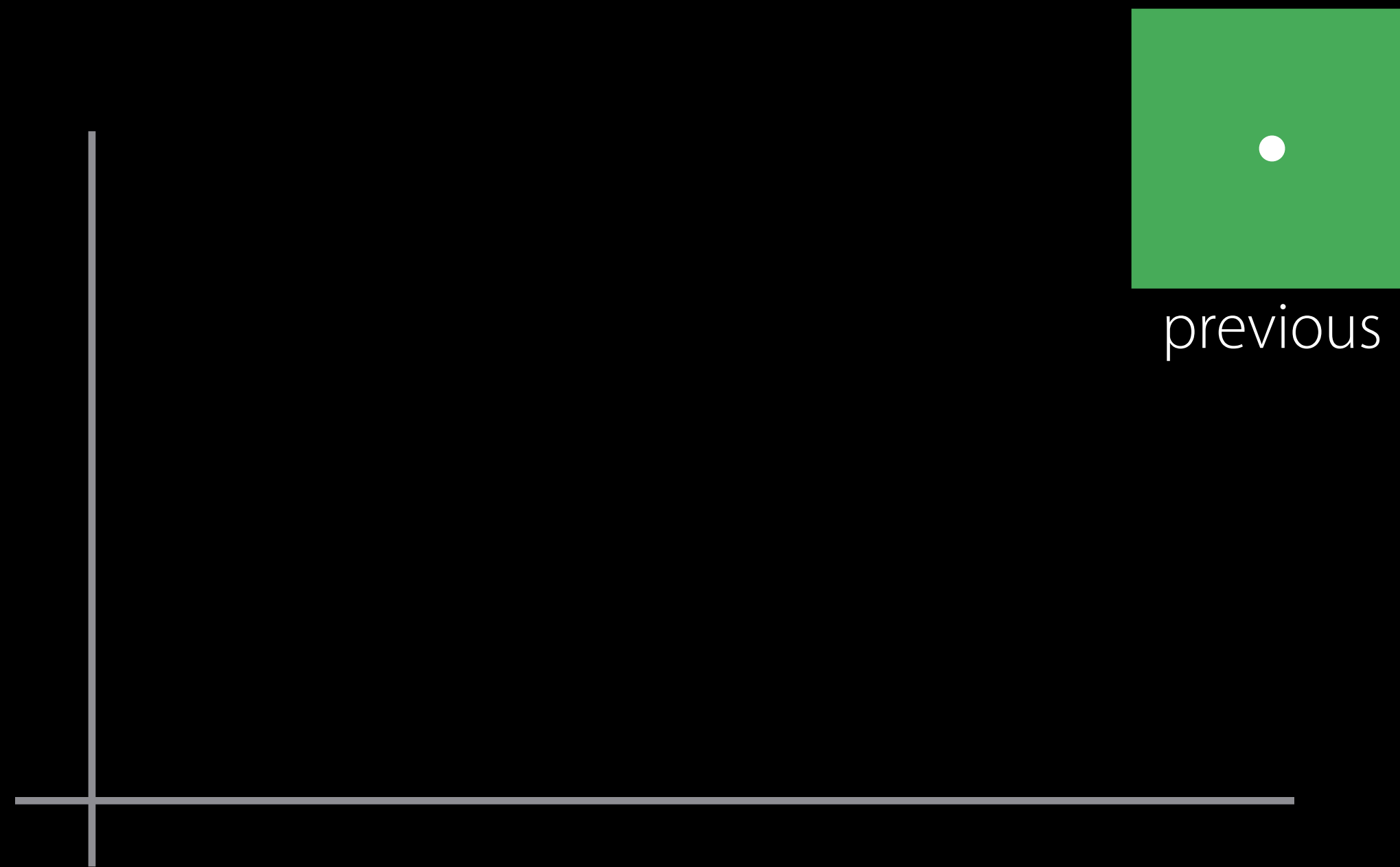


Alignment Feedback



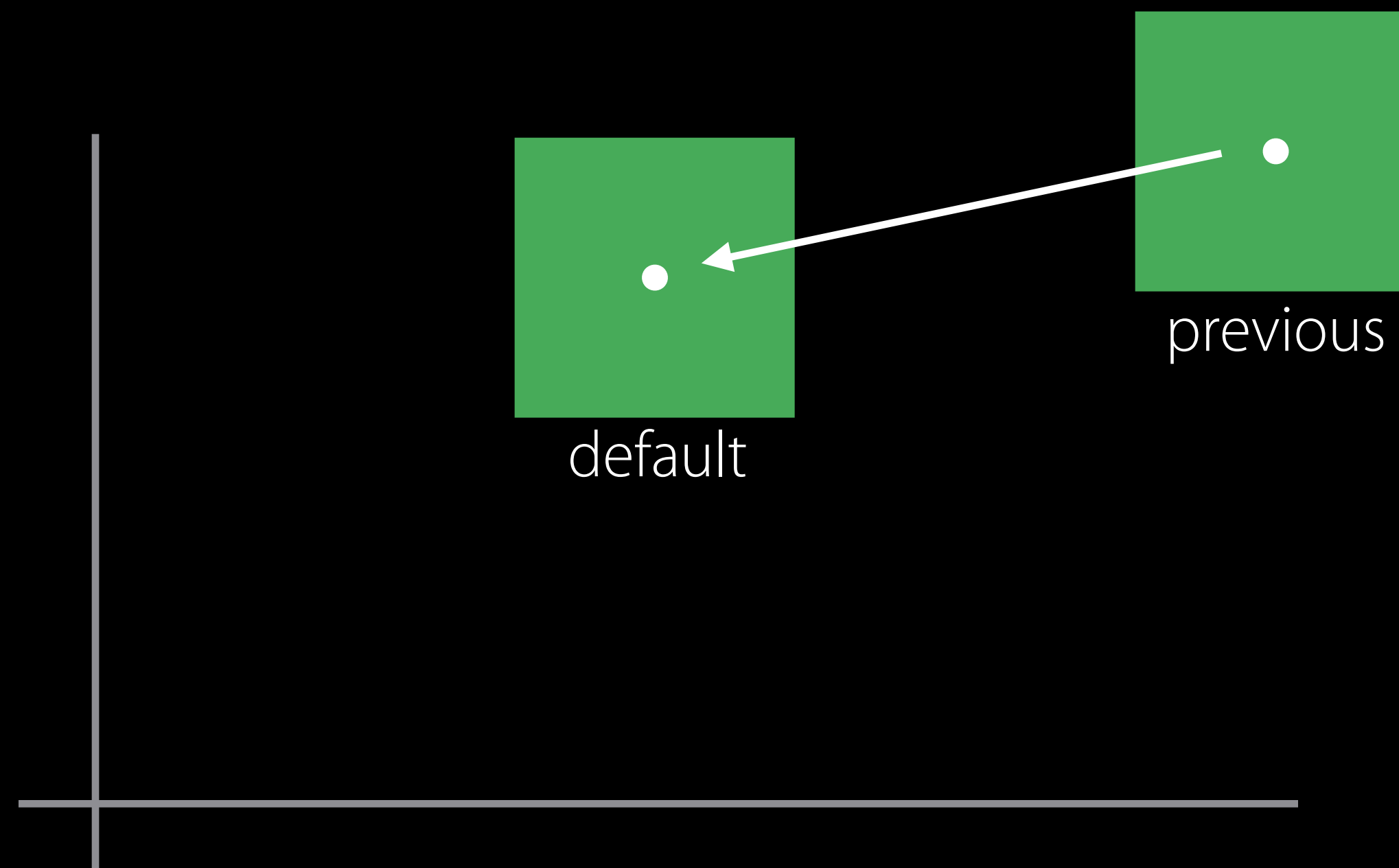
Alignment Feedback

Preparing alignment



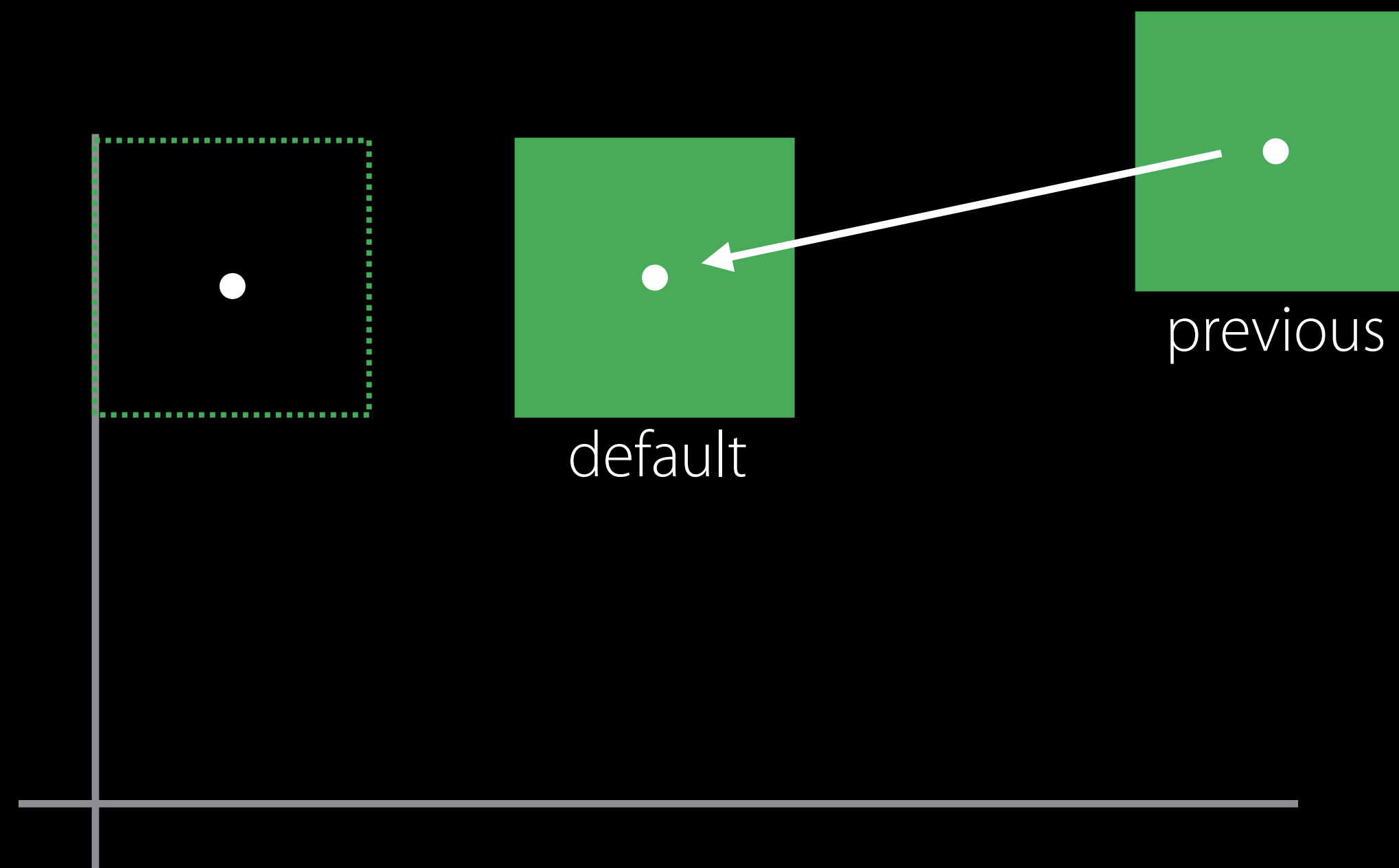
Alignment Feedback

Preparing alignment



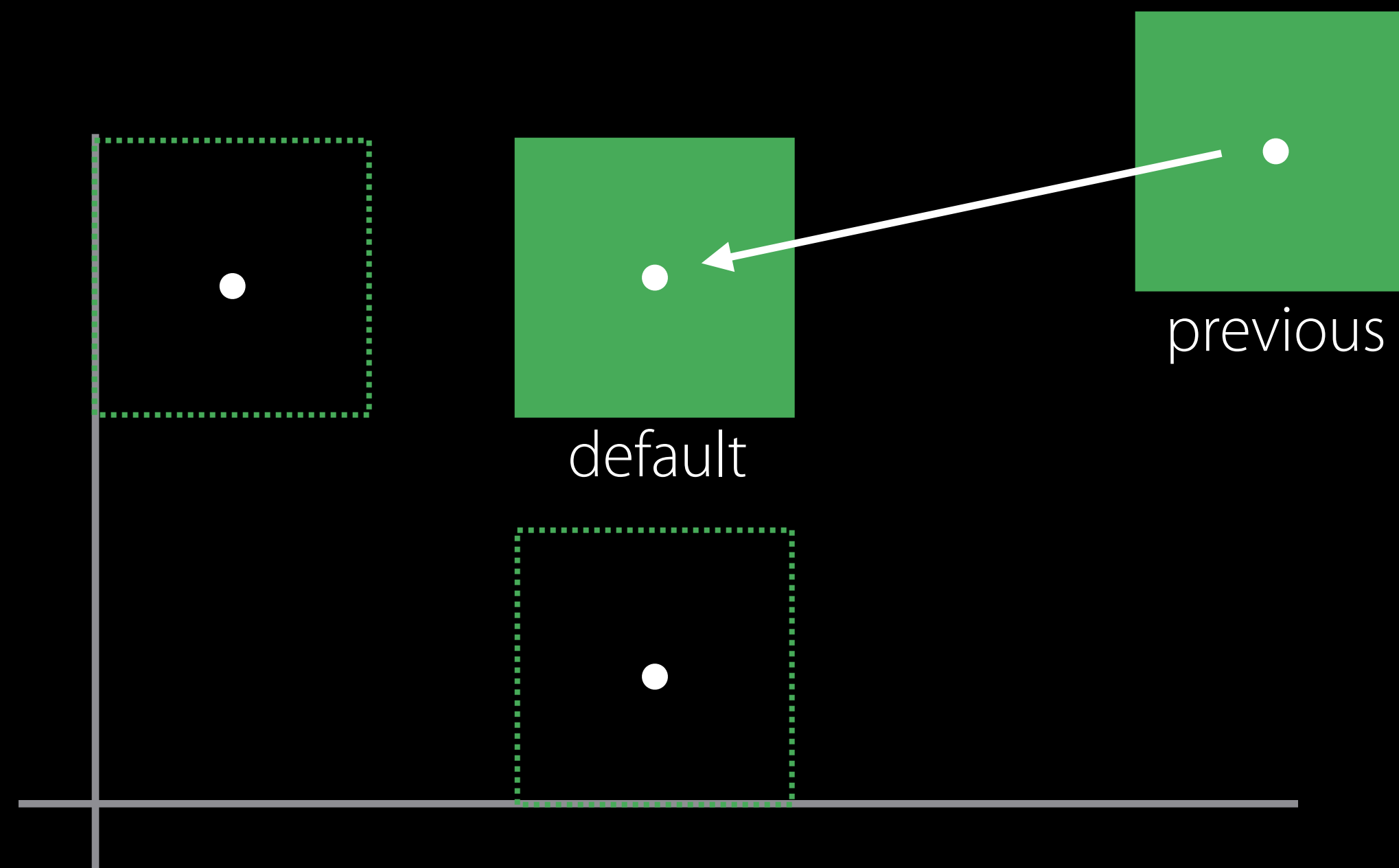
Alignment Feedback

Preparing alignment



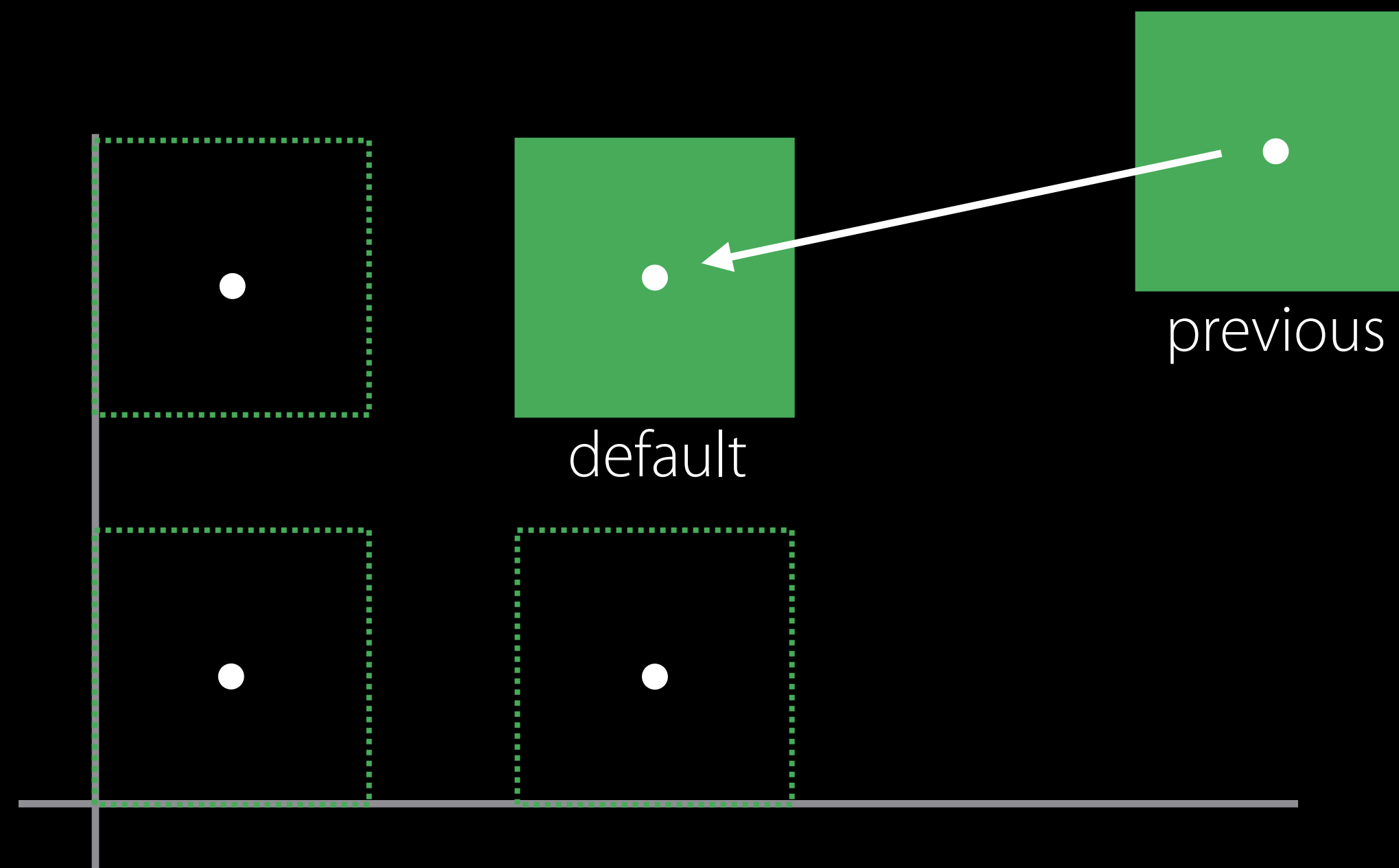
Alignment Feedback

Preparing alignment



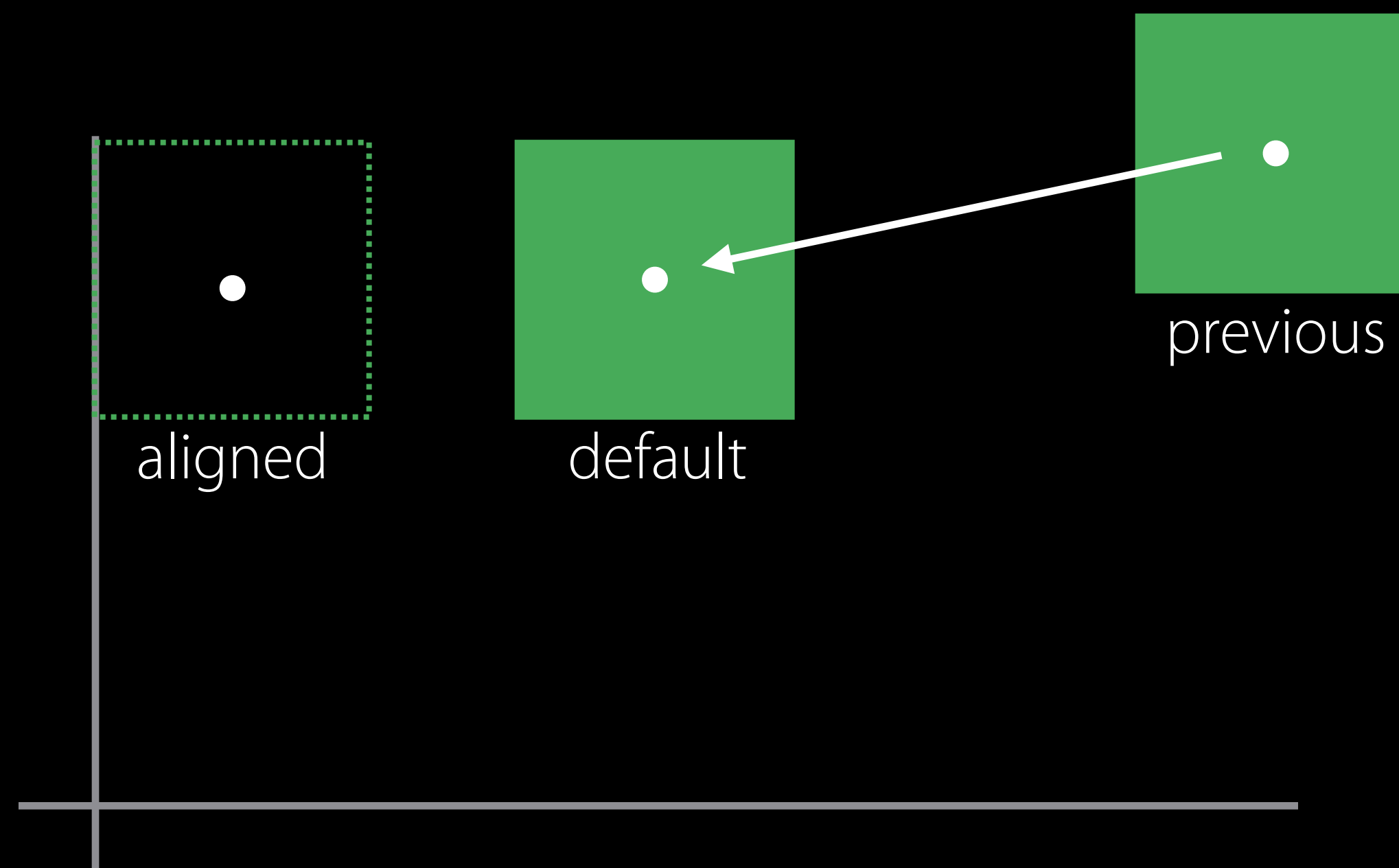
Alignment Feedback

Preparing alignment



Alignment Feedback

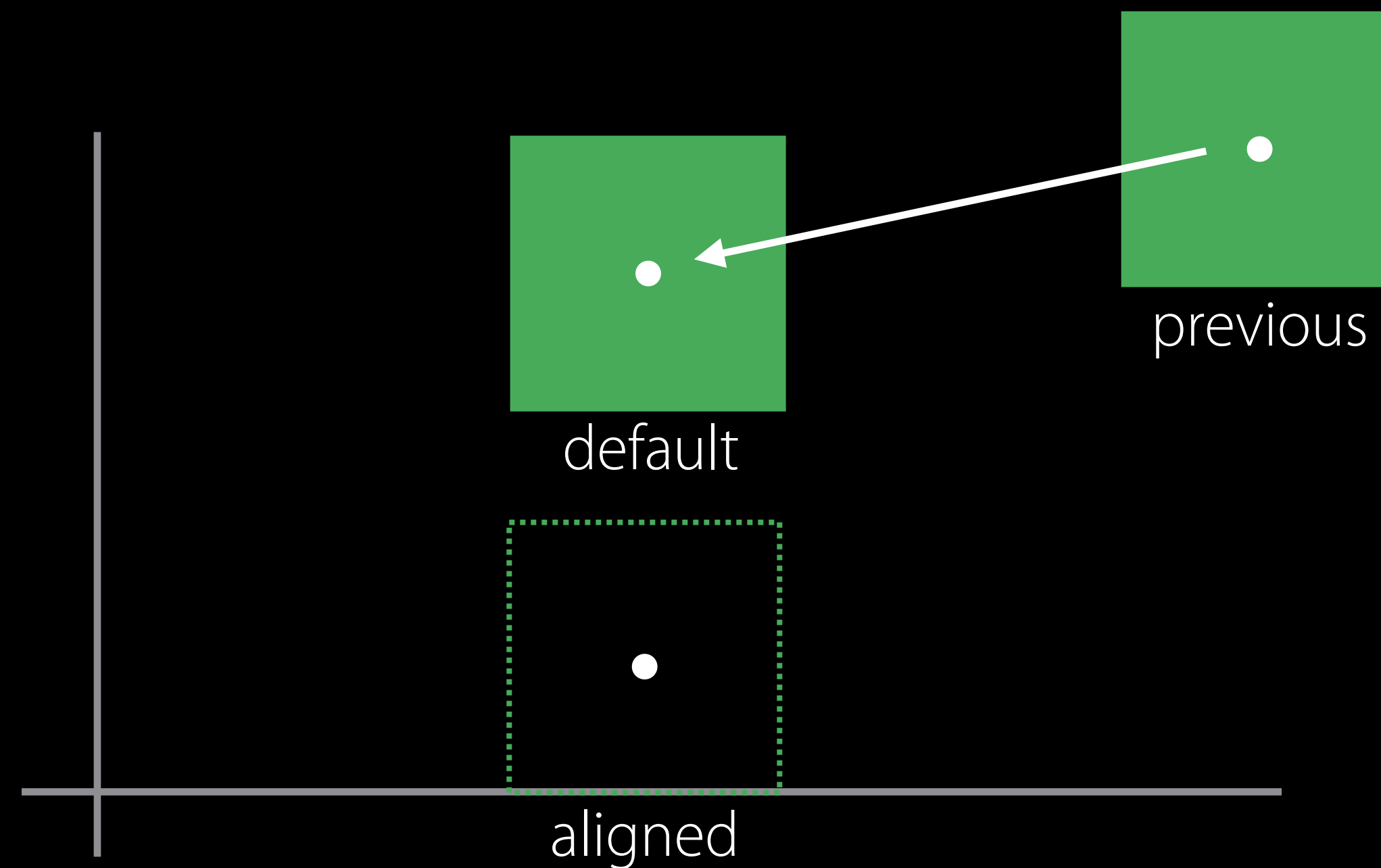
Preparing alignment



```
func alignmentFeedbackTokenForHorizontalMovementInView(NSView?,  
    previousX: CGFloat, alignedX: CGFloat, defaultX: CGFloat)  
    -> NSAlignmentFeedbackToken?
```

Alignment Feedback

Preparing alignment

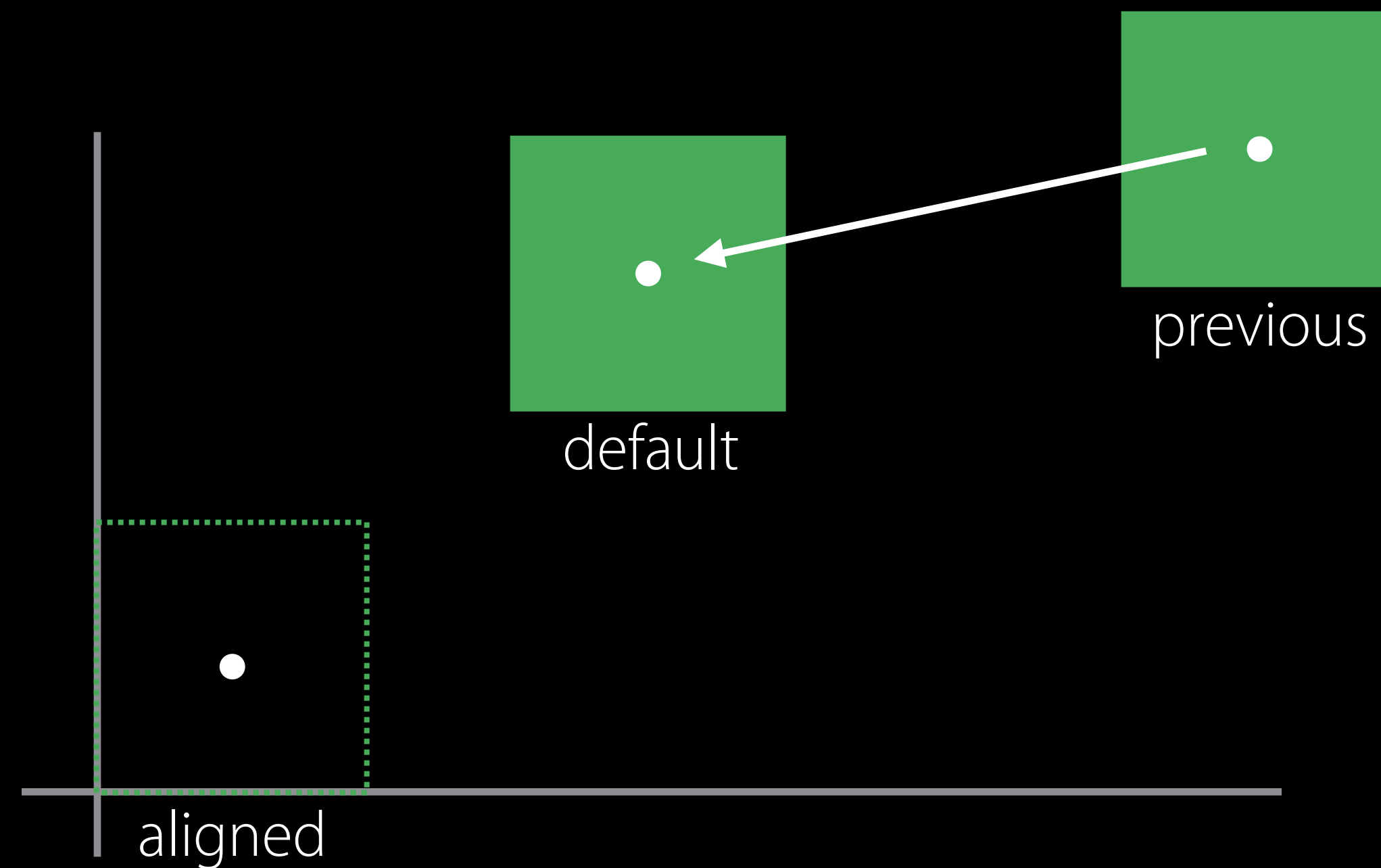


```
func alignmentFeedbackTokenForHorizontalMovementInView(NSView?,  
    previousX: CGFloat, alignedX: CGFloat, defaultX: CGFloat)  
    -> NSAlignmentFeedbackToken?
```

```
func alignmentFeedbackTokenForVerticalMovementInView(NSView?,  
    previousY: CGFloat, alignedY: CGFloat, defaultY: CGFloat)  
    -> NSAlignmentFeedbackToken?
```


Alignment Feedback

Preparing alignment

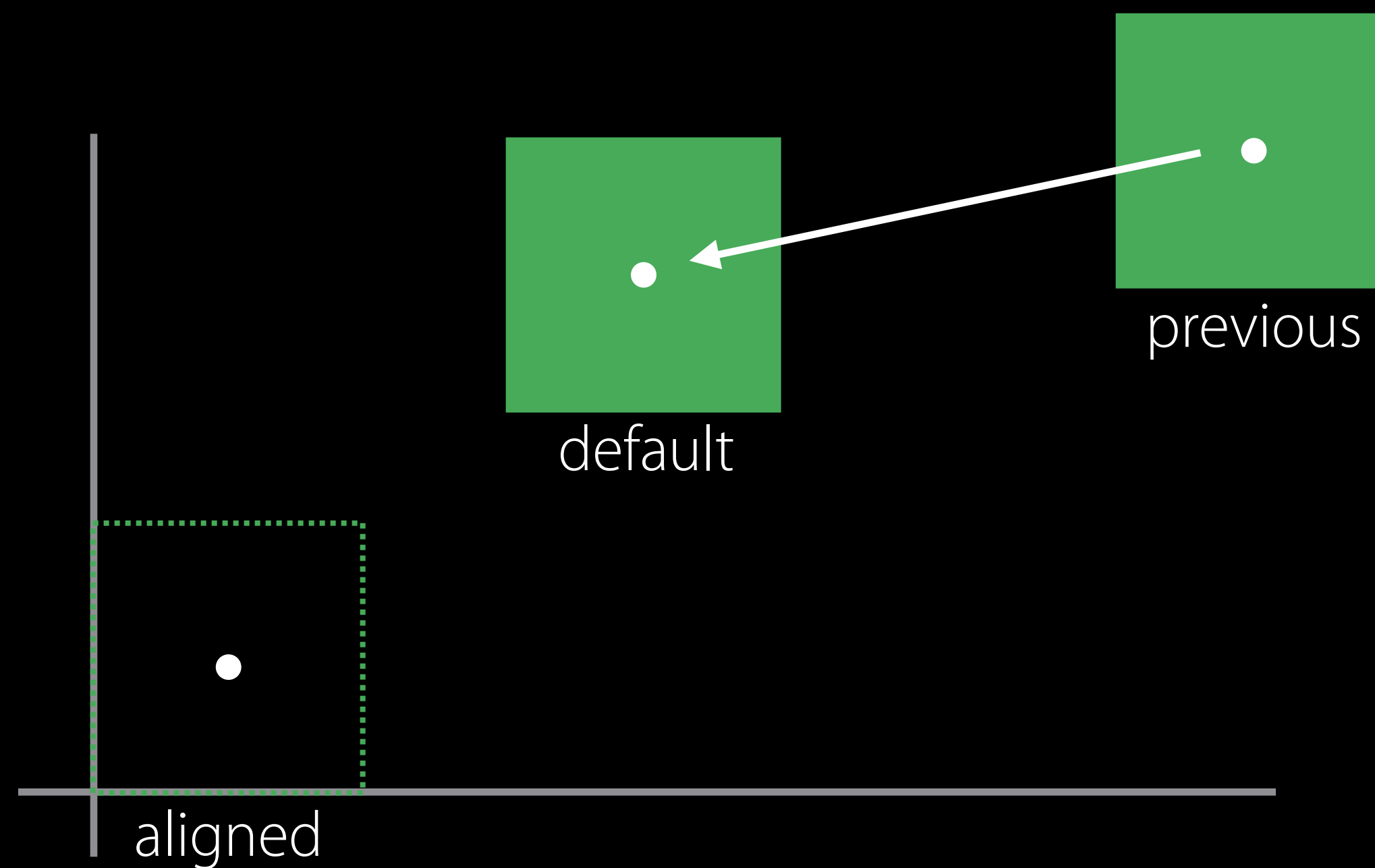


```
func alignmentFeedbackTokenForHorizontalMovementInView(NSView?,  
    previousX: CGFloat, alignedX: CGFloat, defaultX: CGFloat)  
    -> NSAlignmentFeedbackToken?
```

```
func alignmentFeedbackTokenForVerticalMovementInView(NSView?,  
    previousY: CGFloat, alignedY: CGFloat, defaultY: CGFloat)  
    -> NSAlignmentFeedbackToken?
```

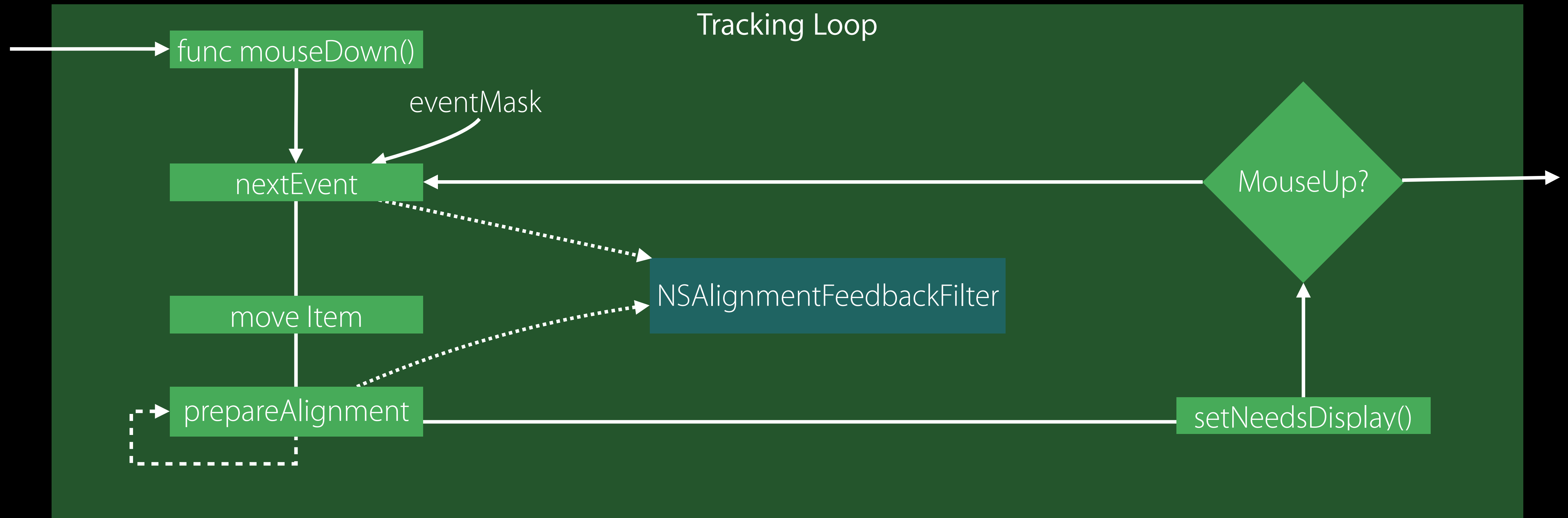
Alignment Feedback

Preparing alignment

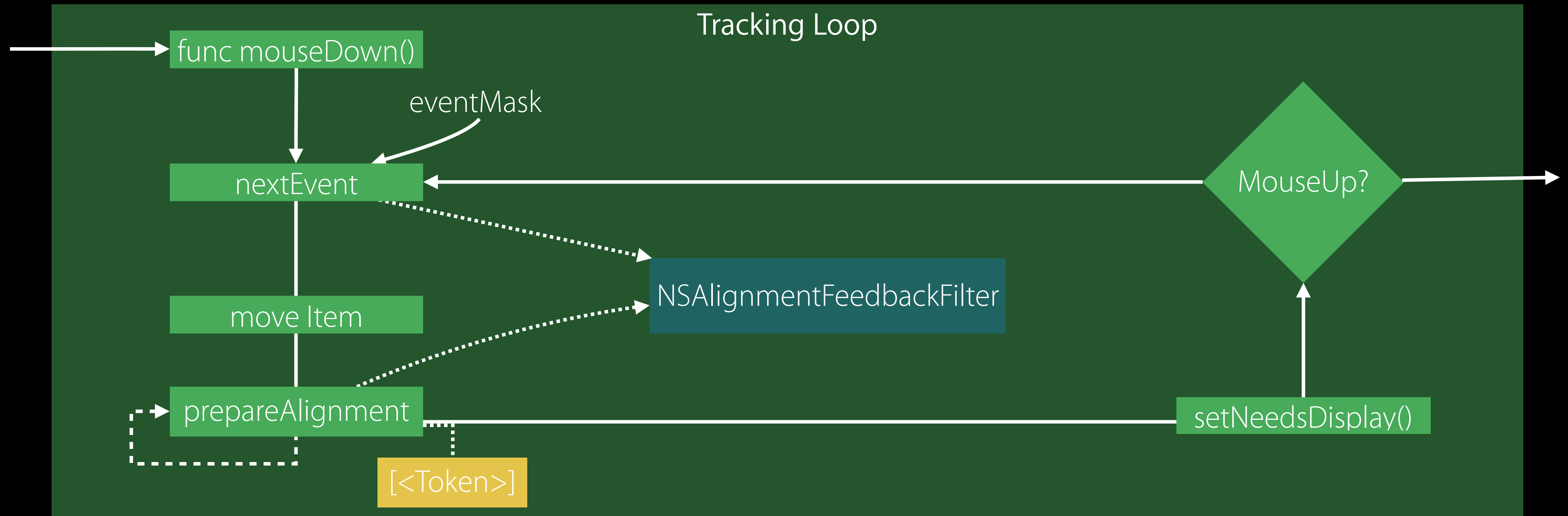


```
func alignmentFeedbackTokenForMovementInView(NSView?,  
    previousPoint: NSPoint, alignedPoint: NSPoint, defaultPoint: NSPoint)  
    -> NSAlignmentFeedbackToken?
```

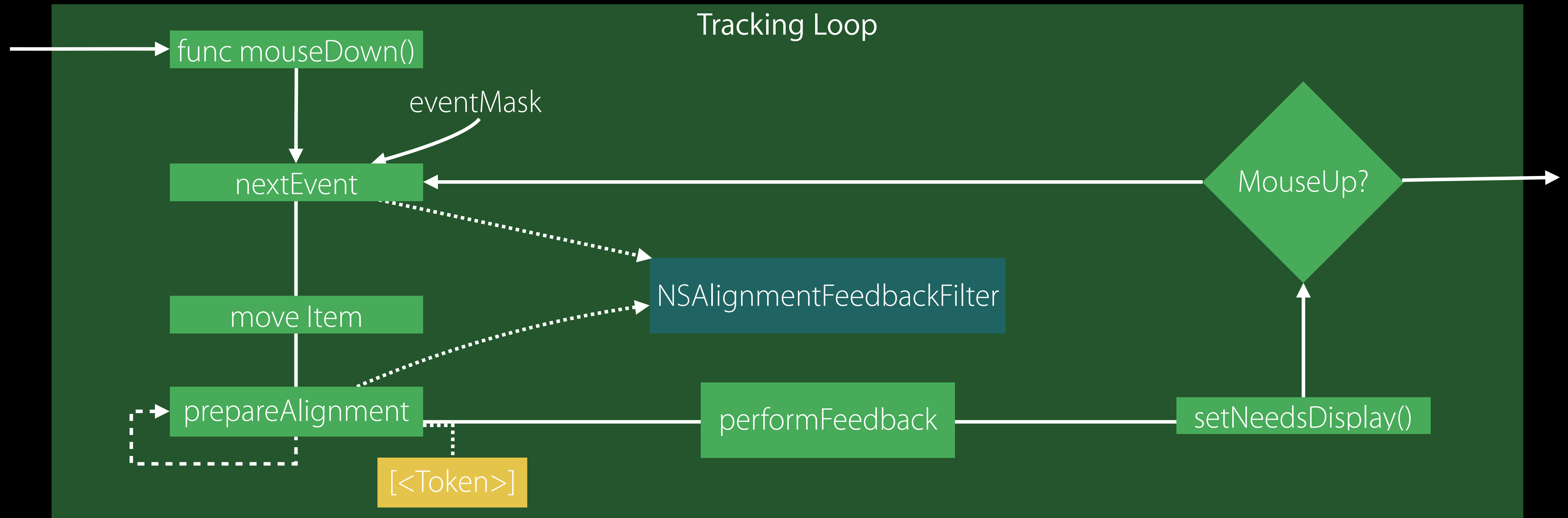
Alignment Feedback



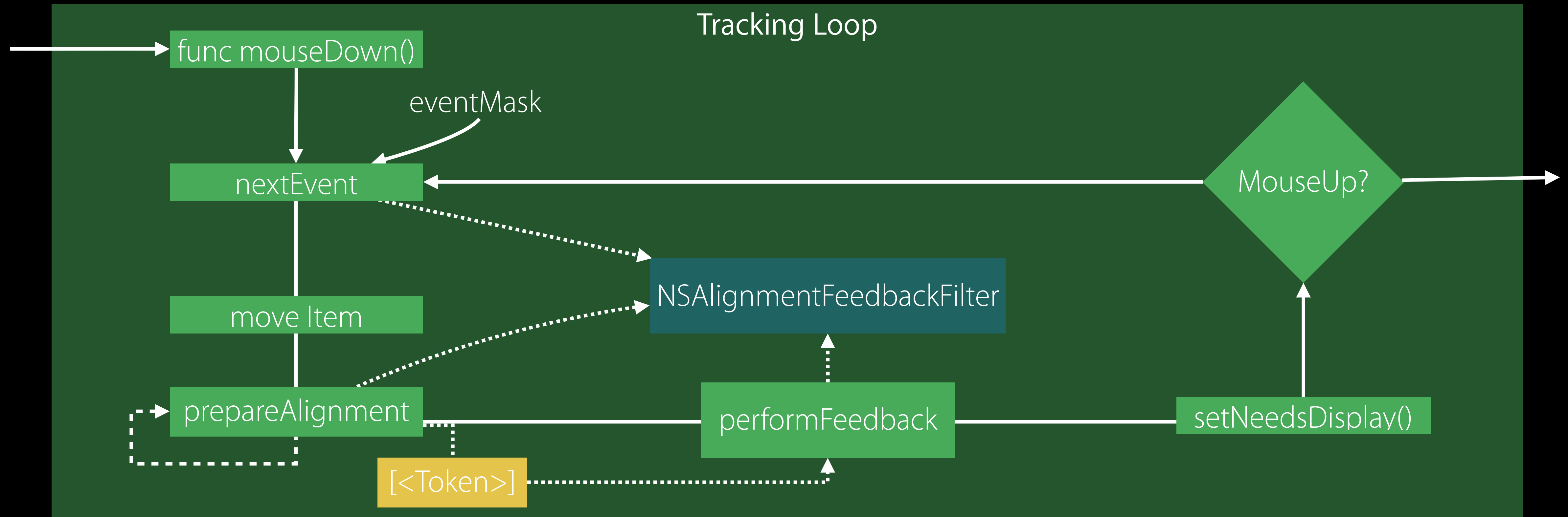
Alignment Feedback



Alignment Feedback



Alignment Feedback



```
func performFeedback( [<NSAlignmentFeedbackToken>],  
    performanceTime: NSHapticFeedbackPerformanceTime)
```

Alignment Feedback

Feeling it

Consistent feel across apps

Easy to integrate into tracking loops

Dragging, resizing, etc...

Knight Summary

Understanding how the force flows

Force event stream

Spring loading protocol

Alignment Feedback

Master

Controlling the force

Master Training Schedule

Controlling the force

Trackpad configuration

Haptics

Trackpad Configuration

More than just Force Click

NSPressureConfiguration

Trackpad Configuration

More than just Force Click

NSPressureConfiguration

`init(pressureBehavior: NSPressureBehavior)`

Trackpad Configuration

More than just Force Click

NSPressureConfiguration

init(pressureBehavior: **NSPressureBehavior**)

- .Primary**Default**
- .Primary**Click**
- .Primary**Generic**
- .Primary**Accelerator**
- .Primary**DeepClick**
- .Primary**DeepDrag**

Trackpad Configuration

More than just Force Click

NSPressureConfiguration

```
init(pressureBehavior: NSPressureBehavior)  
func set()
```

Trackpad Configuration

More than just Force Click

`NSPressureConfiguration`

```
init(pressureBehavior: NSPressureBehavior)
```

```
func set()
```

Set is not ideal

- Only valid during a drag
- Racing the user

Trackpad Configuration

More than just Force Click

Preferred approach:

```
NSView – var pressureConfiguration: NSPressureConfiguration
```

- Configured before mouse down
- Configured when app not responsive

Haptics

The Trackpad Strikes Back



Haptics

The Trackpad Strikes Back

Use sparingly

Subtle interactions



Haptics

The Trackpad Strikes Back



NSHapticFeedbackManager

```
class func defaultPerformer() -> <NSHapticFeedbackPerformer>
```

```
@protocol NSHapticFeedbackPerformer <NSObject>
```

```
func performFeedbackPattern(NSHapticFeedbackPattern,  
                             performanceTime: NSHapticFeedbackPerformanceTime)
```

Haptics

The Trackpad Strikes Back



NSHapticFeedbackManager

```
class func defaultPerformer() -> <NSHapticFeedbackPerformer>
```

```
@protocol NSHapticFeedbackPerformer <NSObject>
```

```
func performFeedbackPattern(NSHapticFeedbackPattern,  
                             performanceTime: NSHapticFeedbackPerformanceTime)
```

Haptics

The Trackpad Strikes Back



NSHapticFeedbackManager

```
class func defaultPerformer() -> <NSHapticFeedbackPerformer>
```

```
@protocol NSHapticFeedbackPerformer <NSObject>
```

```
func performFeedbackPattern(NSHapticFeedbackPattern,  
                             performanceTime: NSHapticFeedbackPerformanceTime)
```

NSHapticFeedbackPattern

.Generic

.Alignment

.LevelChange

Haptics

The Trackpad Strikes Back



NSHapticFeedbackManager

```
class func defaultPerformer() -> <NSHapticFeedbackPerformer>
```

```
@protocol NSHapticFeedbackPerformer <NSObject>
```

```
func performFeedbackPattern(NSHapticFeedbackPattern,  
                             performanceTime: NSHapticFeedbackPerformanceTime)
```

NSHapticFeedbackPerformanceTime

.Default

.Now

.DrawCompleted

Summary

Table Row Actions

Accelerator Buttons

Spring Loading

Pressure Event Stream

Alignment Feedback

Configuration

Haptic Feedback

More Information

Documentation

ForceTouchCatalog

AlignmentGuides

<http://developer.apple.com/library>

Technical Support

Apple Developer Forums

<http://developer.apple.com/forums>

General Inquiries

Paul Marcos

App Frameworks Evangelist

pmarcos@apple.com

Related Sessions

What's New in Web Development in WebKit and Safari	Mission	Tuesday 9:00AM
What's New in Cocoa	Presidio	Tuesday 1:30PM
What's New in NSCollectionView	Mission	Thursday 4:30PM

Labs

Cocoa and Force Touch and Gestures Lab	Frameworks Lab A	Thursday 11:00AM
Cocoa and UICollectionView Lab	Frameworks Lab B	Friday 9:00AM

