Animations for everyone

Amos Gyamfi & Stefan Blos

Stefan Blos

DX Engineer at Stream

Worked in mobile, web, cloud

𝕏 **@stefanjblos**

Amos Gyamfi

iOS Developer Advocate at Stream

Switched into DevRel from design

𝕏 **@amos_gyamfi**

Why are we here?

We like nice animations

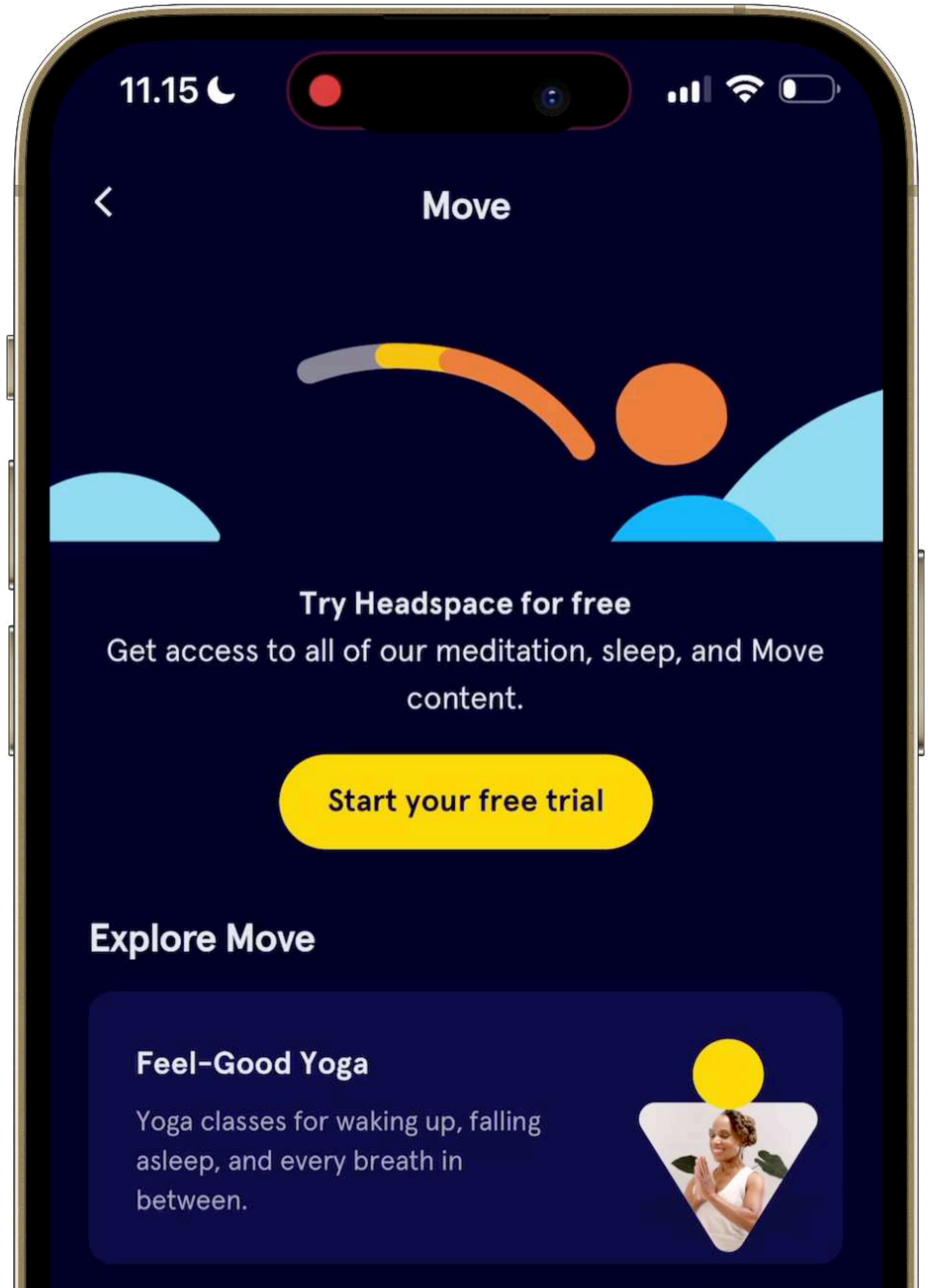
We want to give you best practices

We’ll talk about Accessibility

We’ll show you (a lot of) examples

Story time

From this…



11.15

Move

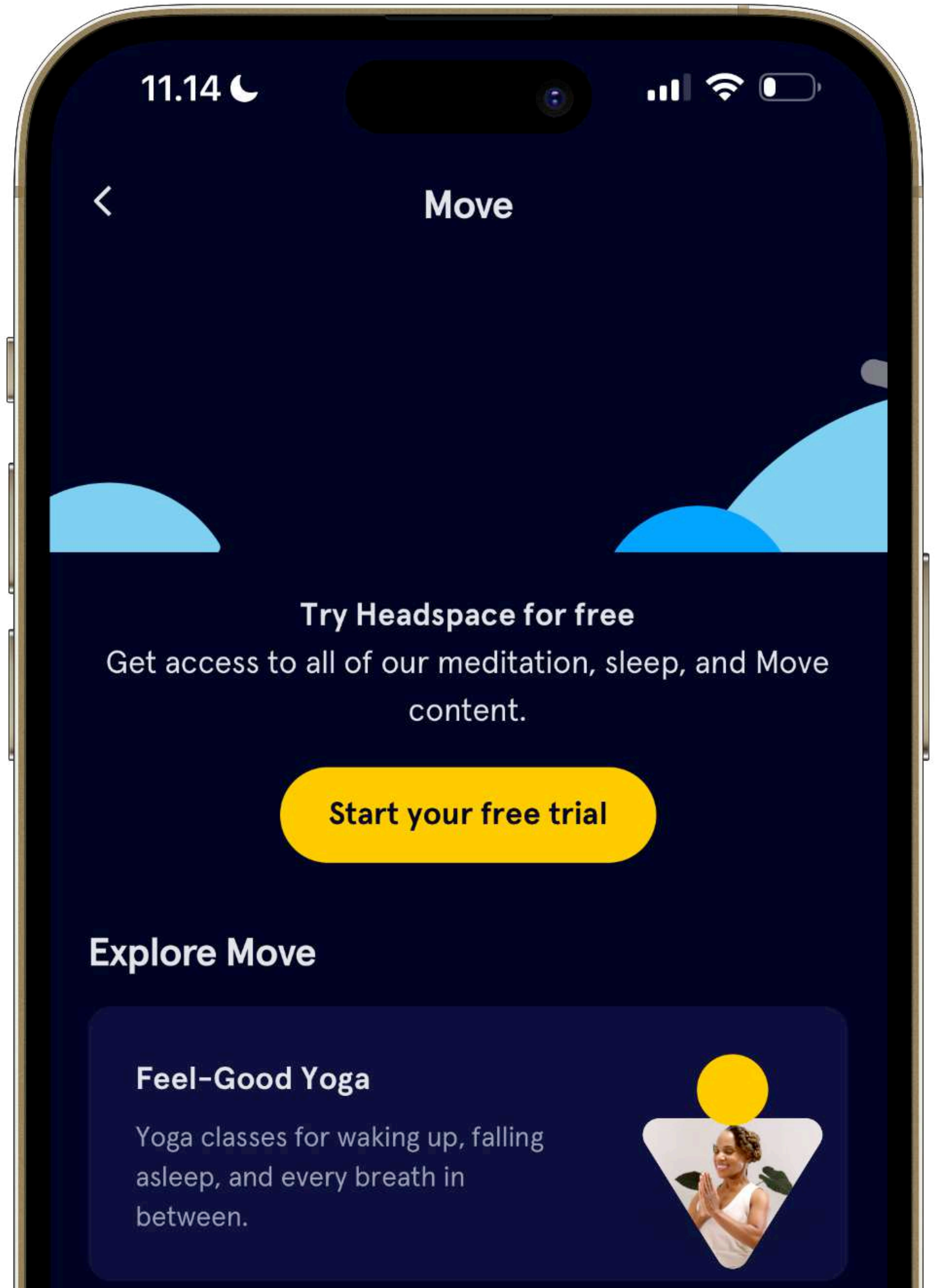
Try Headspace for free
Get access to all of our meditation, sleep, and Move content.

Start your free trial

Explore Move

Feel-Good Yoga
Yoga classes for waking up, falling asleep, and every breath in between.

…to this



11.14

Move

Try Headspace for free
Get access to all of our meditation, sleep, and Move content.

Start your free trial

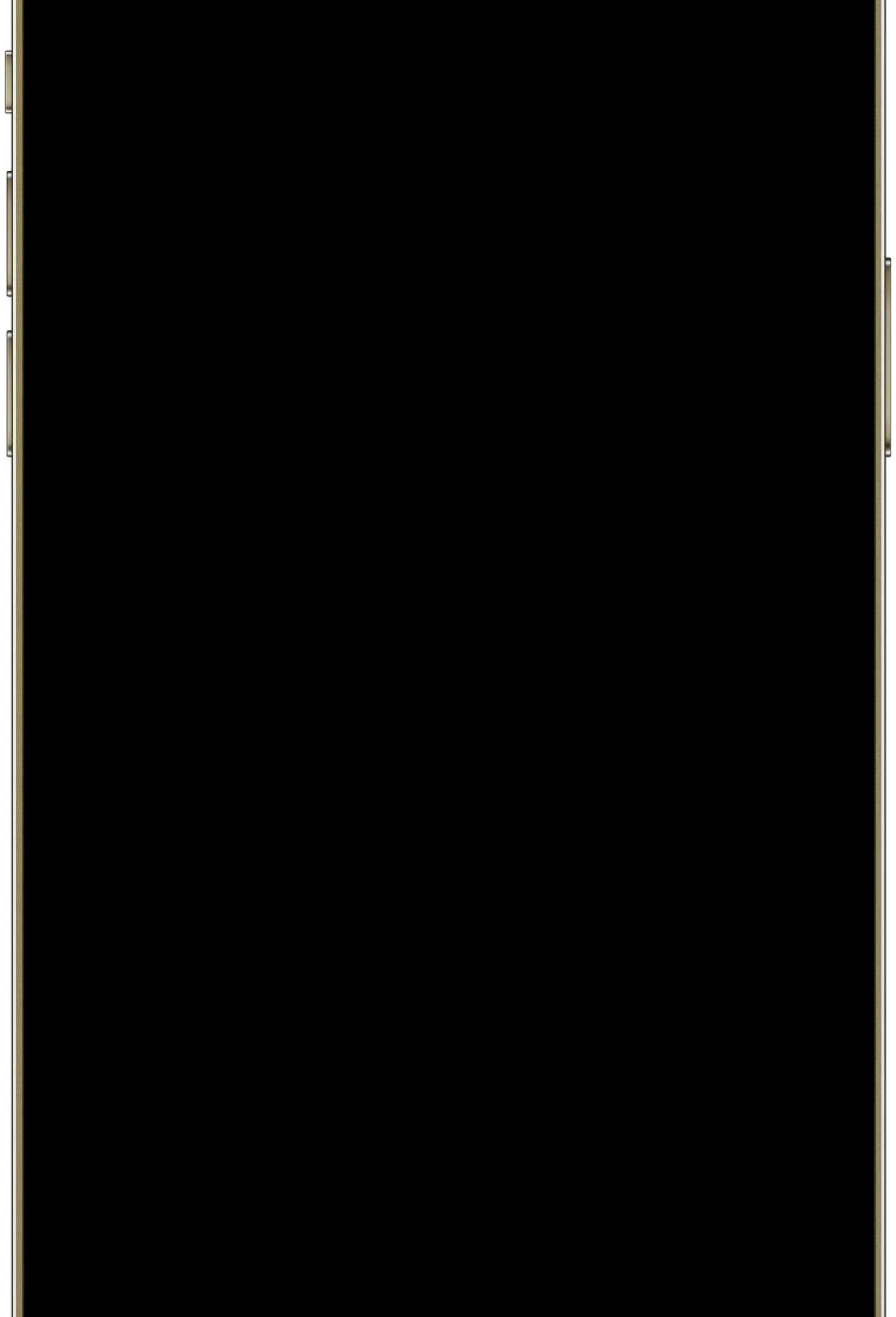
Explore Move

Feel-Good Yoga
Yoga classes for waking up, falling asleep, and every breath in between.

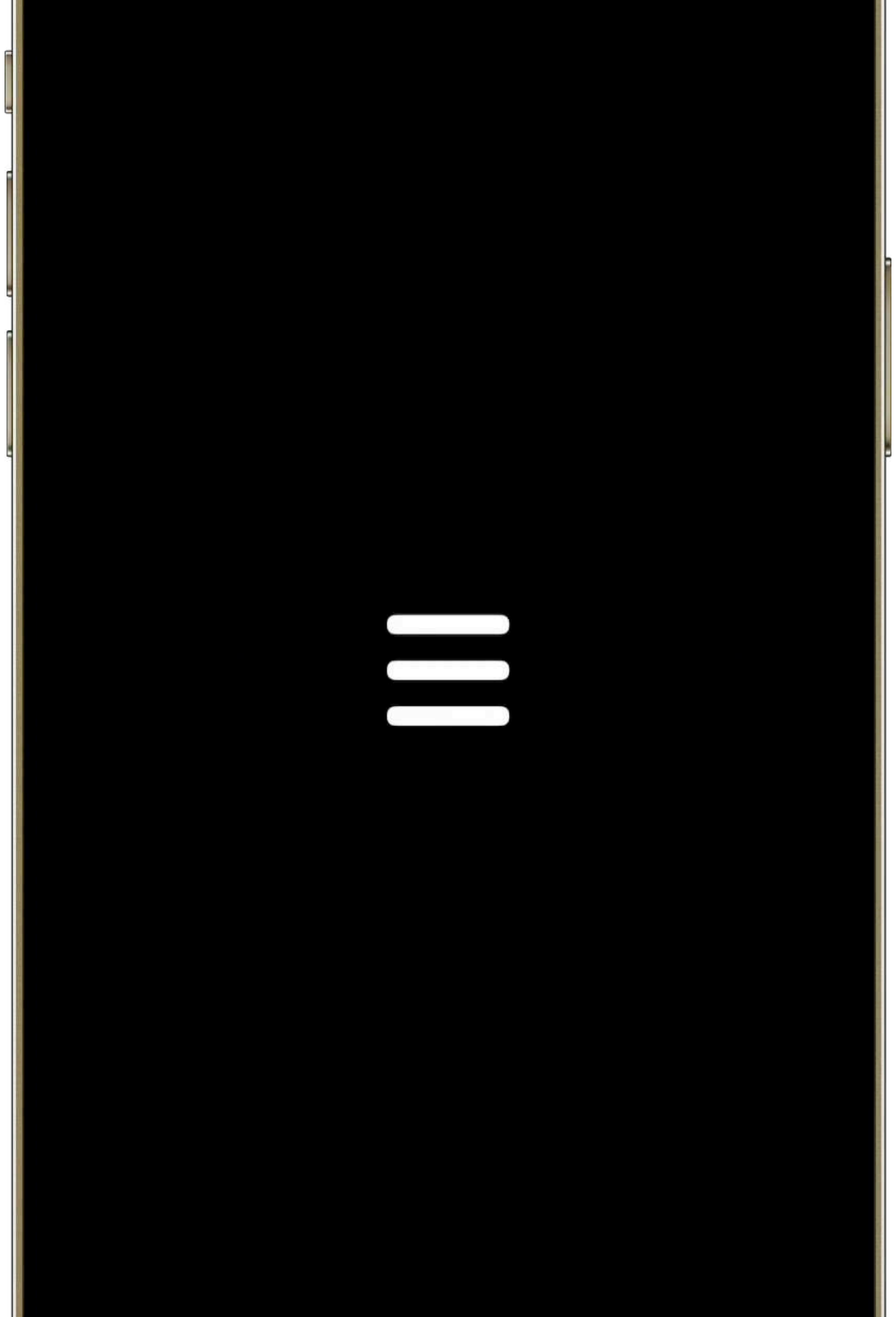
Why animations?

Delight

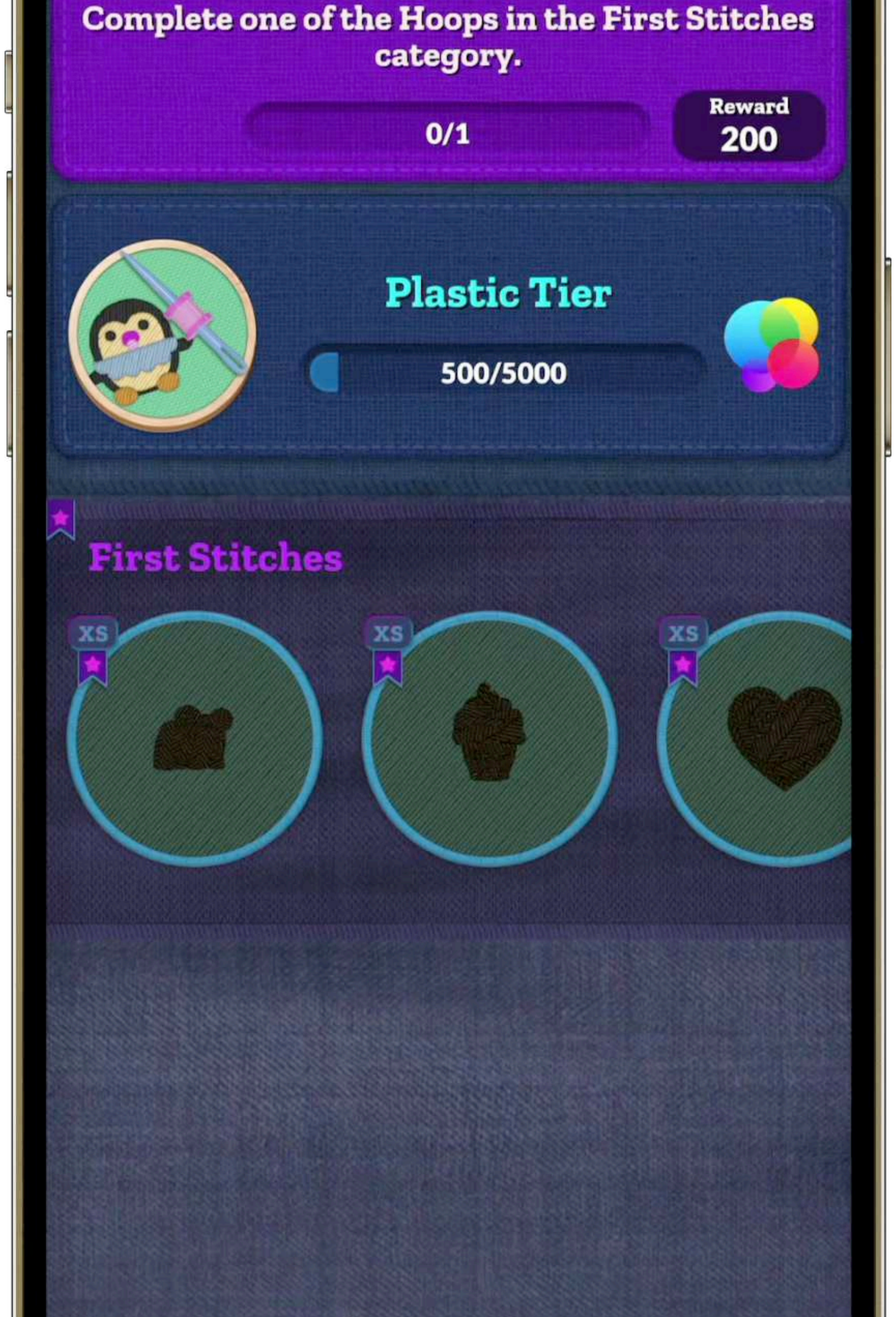
Playfulness



State change



Draw attention



Complete one of the Hoops in the First Stitches category.

0/1

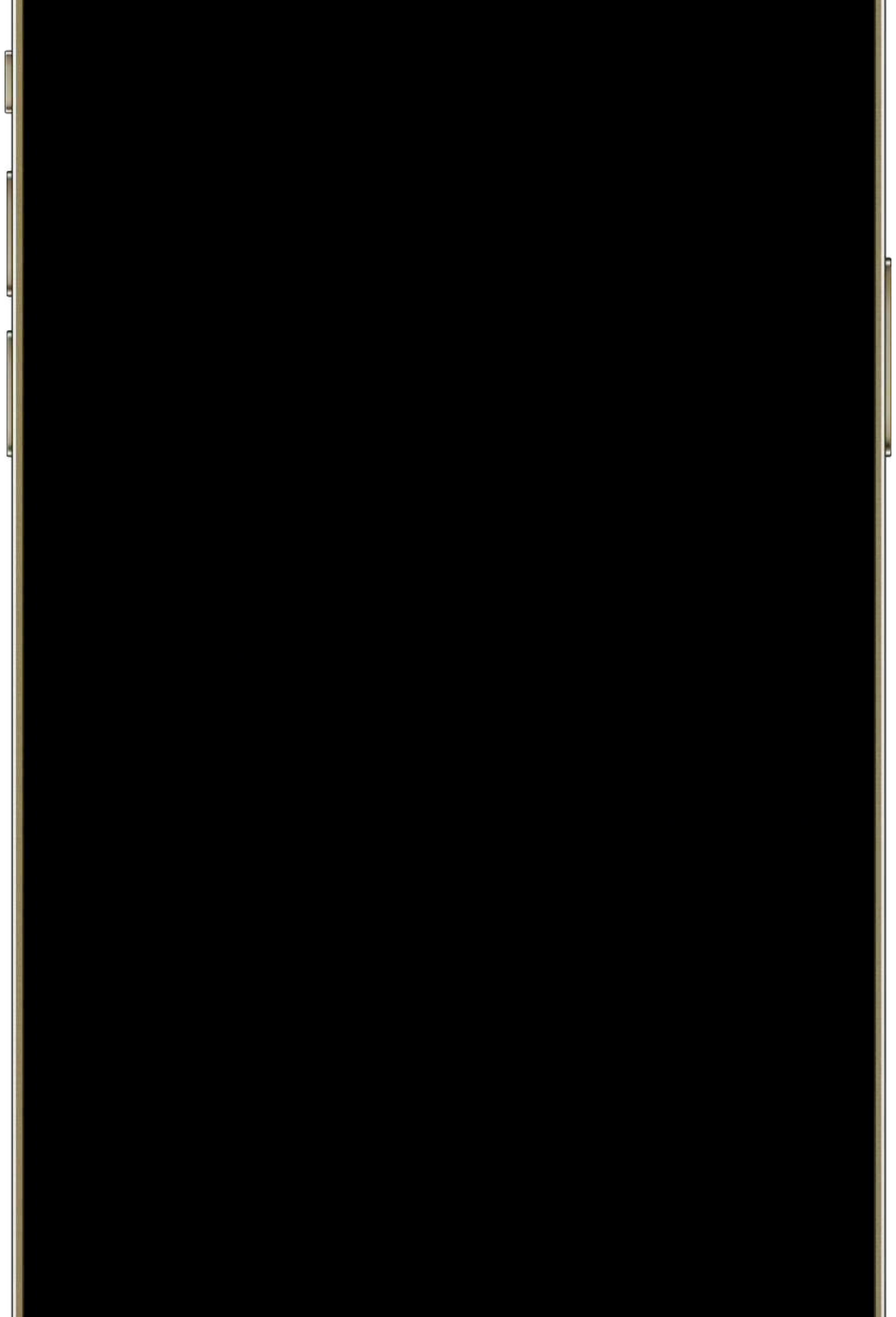
Reward 200

Plastic Tier

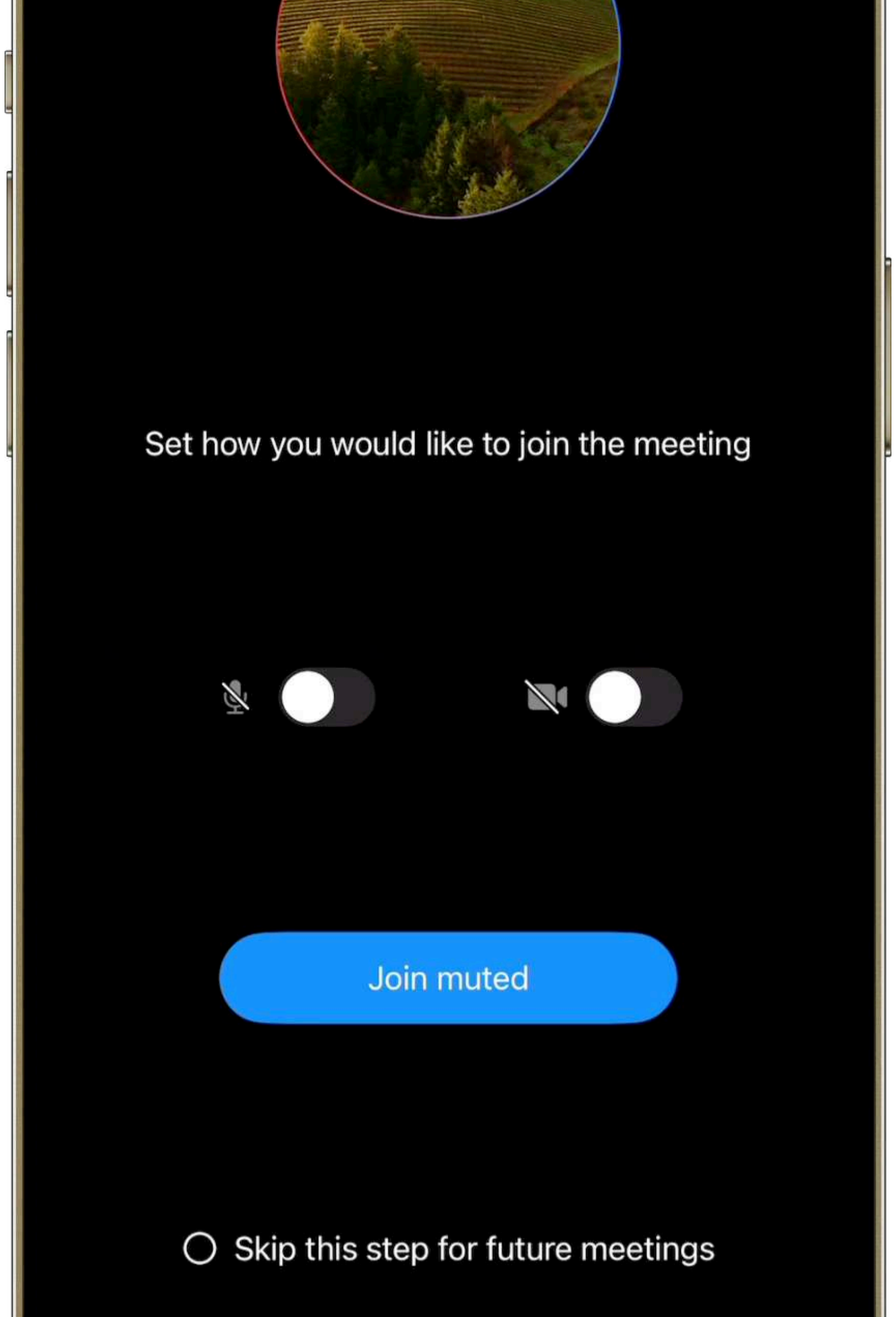
500/5000

First Stitches

Guidance



Animation types in SwiftUI?



Set how you would like to join the meeting

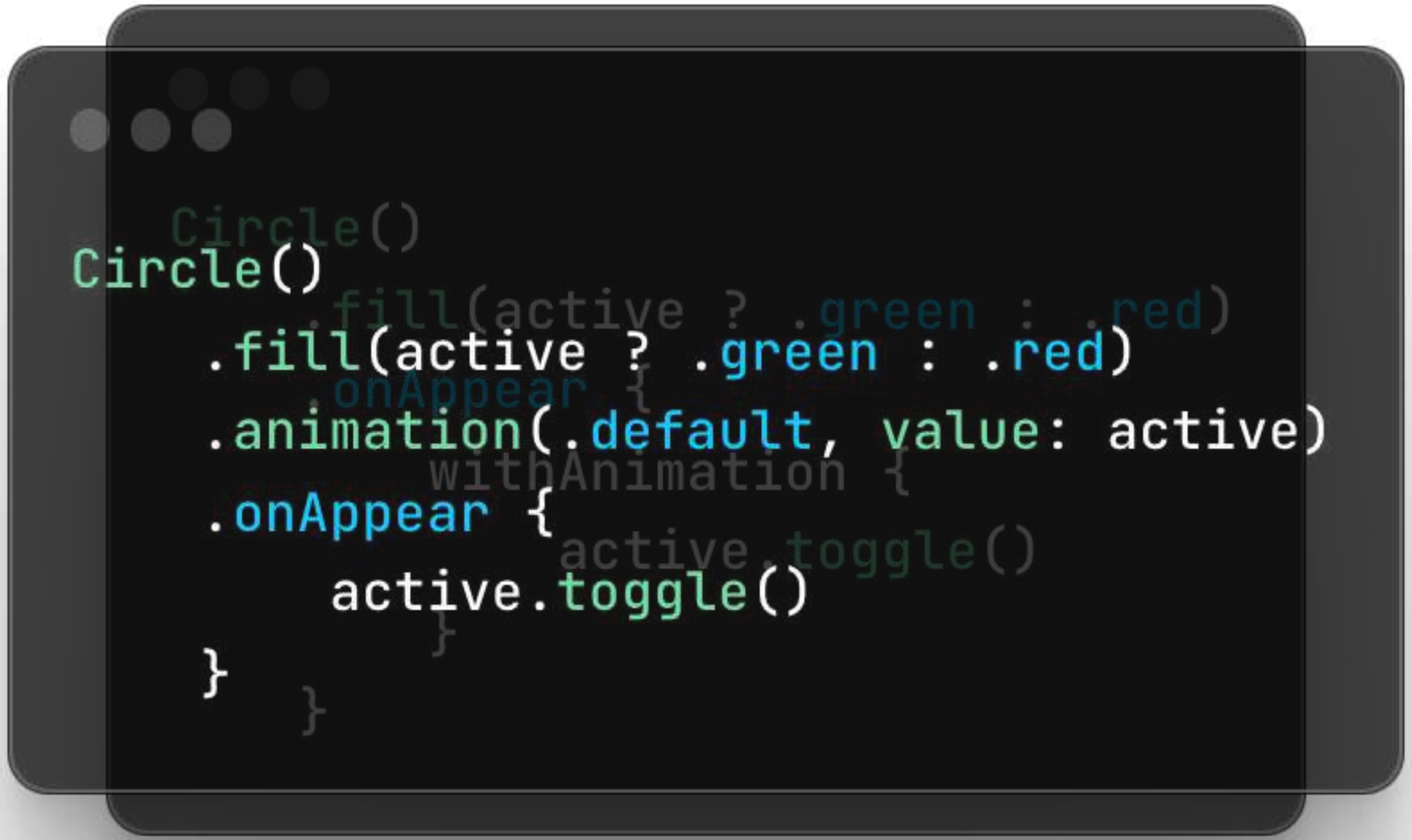
○ Skip this step for future meetings

User-initiated

Programmatical

Implicit

Explicit



```
Circle()
    .fill(active ? .green : .red)
    .animation(.default, value: active)
    .onAppear {
        active.toggle()
    }
```

```
Circle()
    .fill(active ? .green : .red)
    .onAppear {
        withAnimation {
            active.toggle()
        }
    }
```

- Symbols
- 17 PhaseAnimator
- 17 Keyframes
- 17 Springs

Symbol effects

Animate between symbol (states)

Content switching

Great for tab bars / toolbars

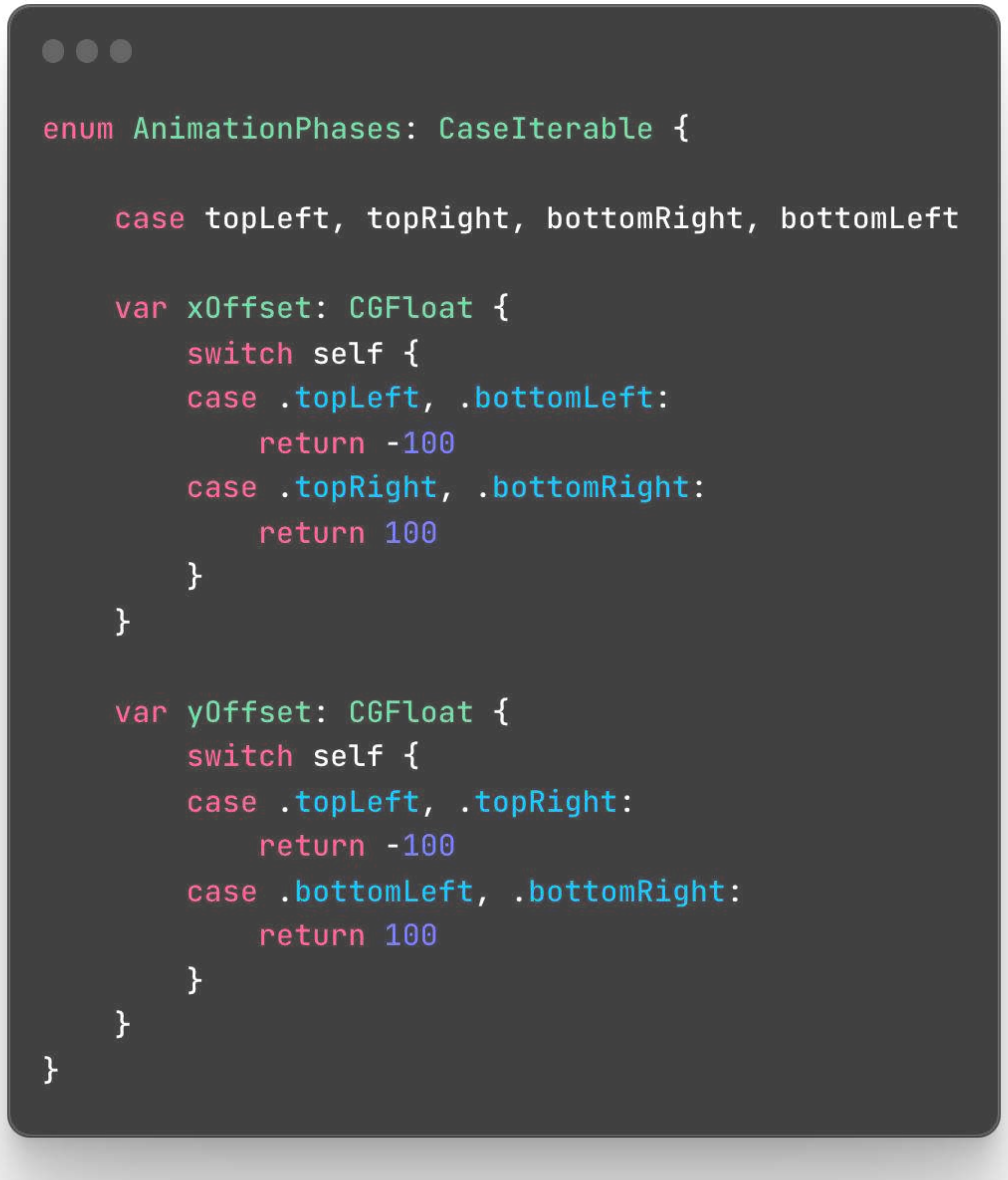
PhaseAnimator 17

Split animation in phases

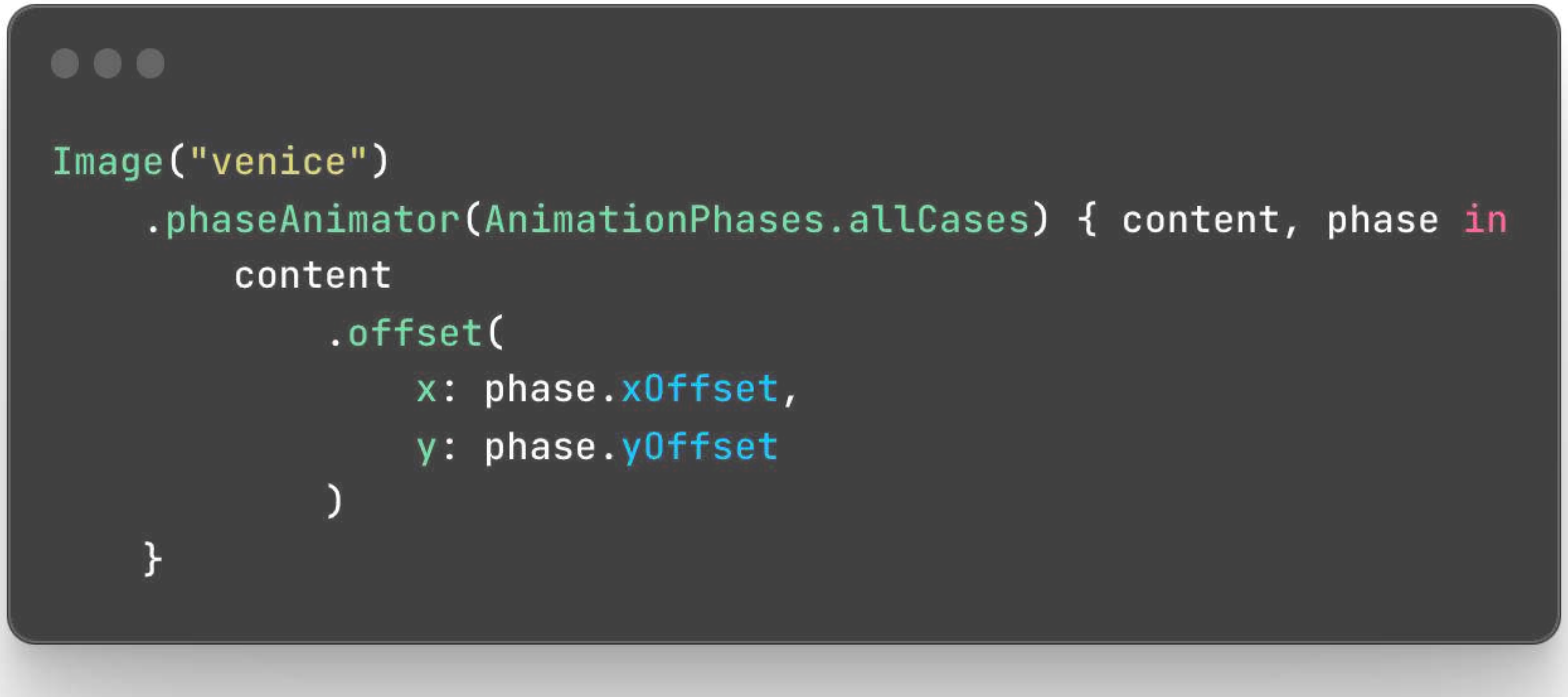
Animate multiple properties

All properties have same timings

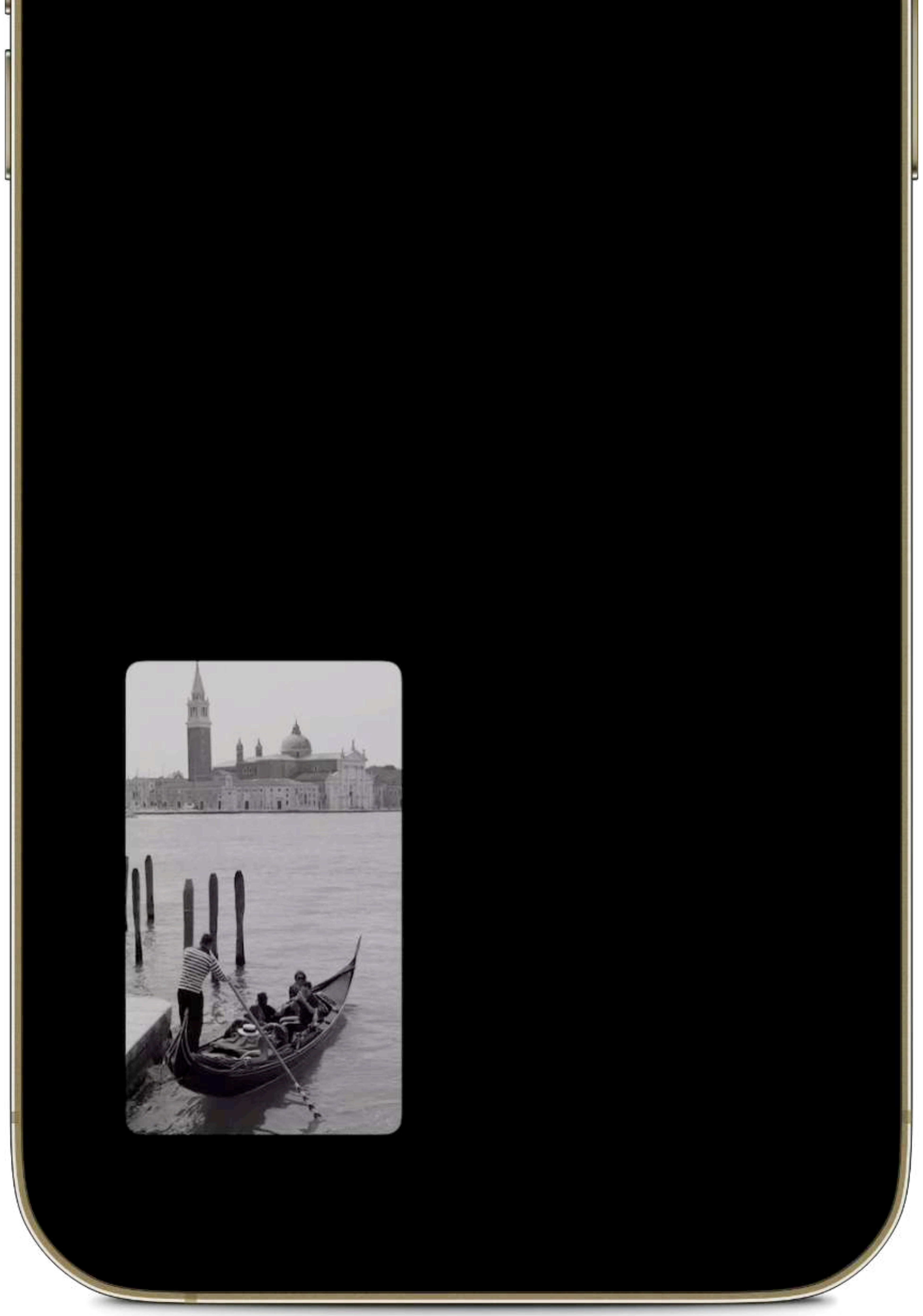
PhaseAnimator 17



```
enum AnimationPhases: CaseIterable {

    case topLeft, topRight, bottomRight, bottomLeft

    var xOffset: CGFloat {
        switch self {
        case .topLeft, .bottomLeft:
            return -100
        case .topRight, .bottomRight:
            return 100
        }
    }

    var yOffset: CGFloat {
        switch self {
        case .topLeft, .topRight:
            return -100
        case .bottomLeft, .bottomRight:
            return 100
        }
    }
}
```



```
Image("venice")
    .phaseAnimator(AnimationPhases.allCases) { content, phase in
        content
            .offset(
                x: phase.xOffset,
                y: phase.yOffset
            )
    }
```



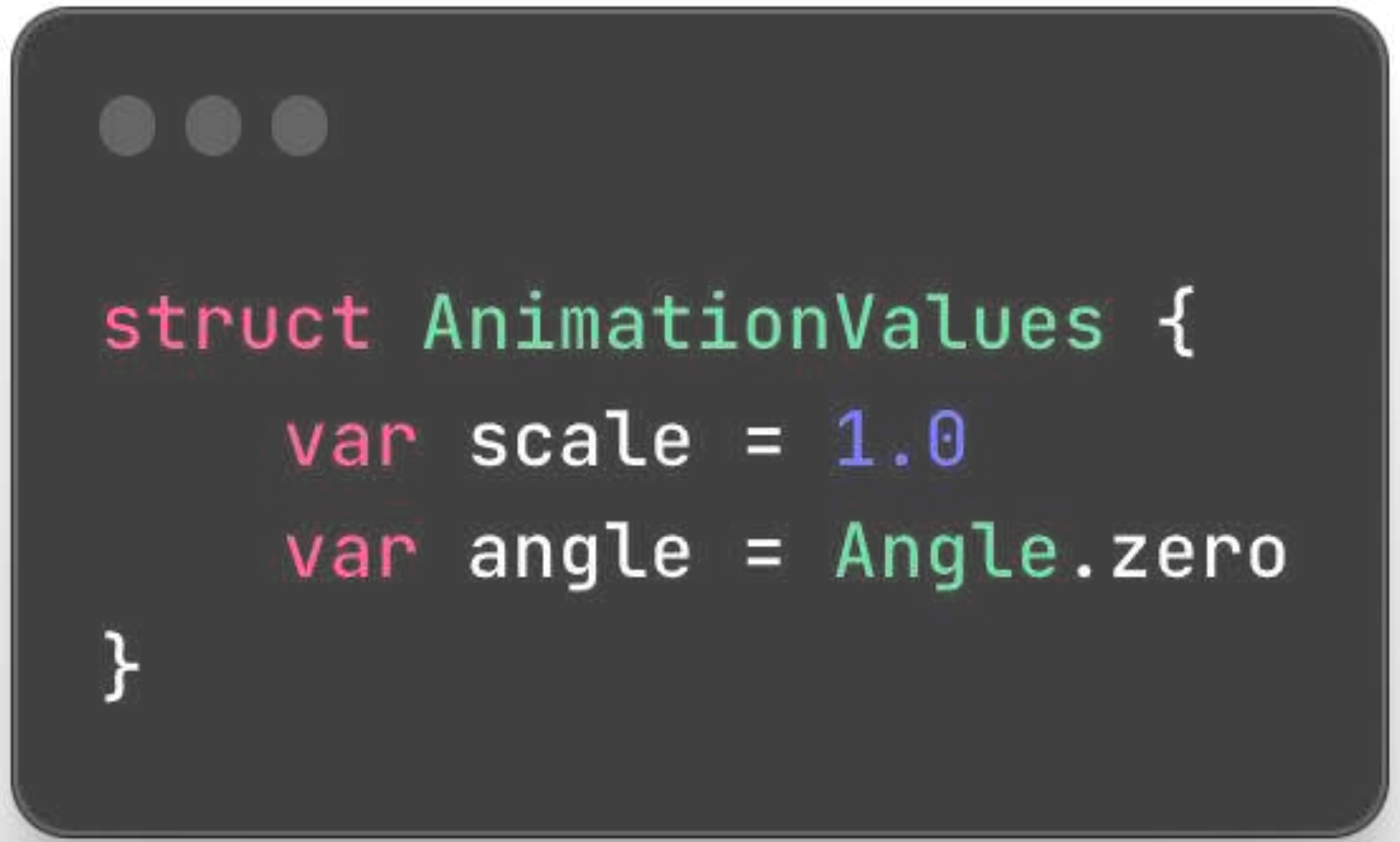
Keyframes 17

Different properties separately

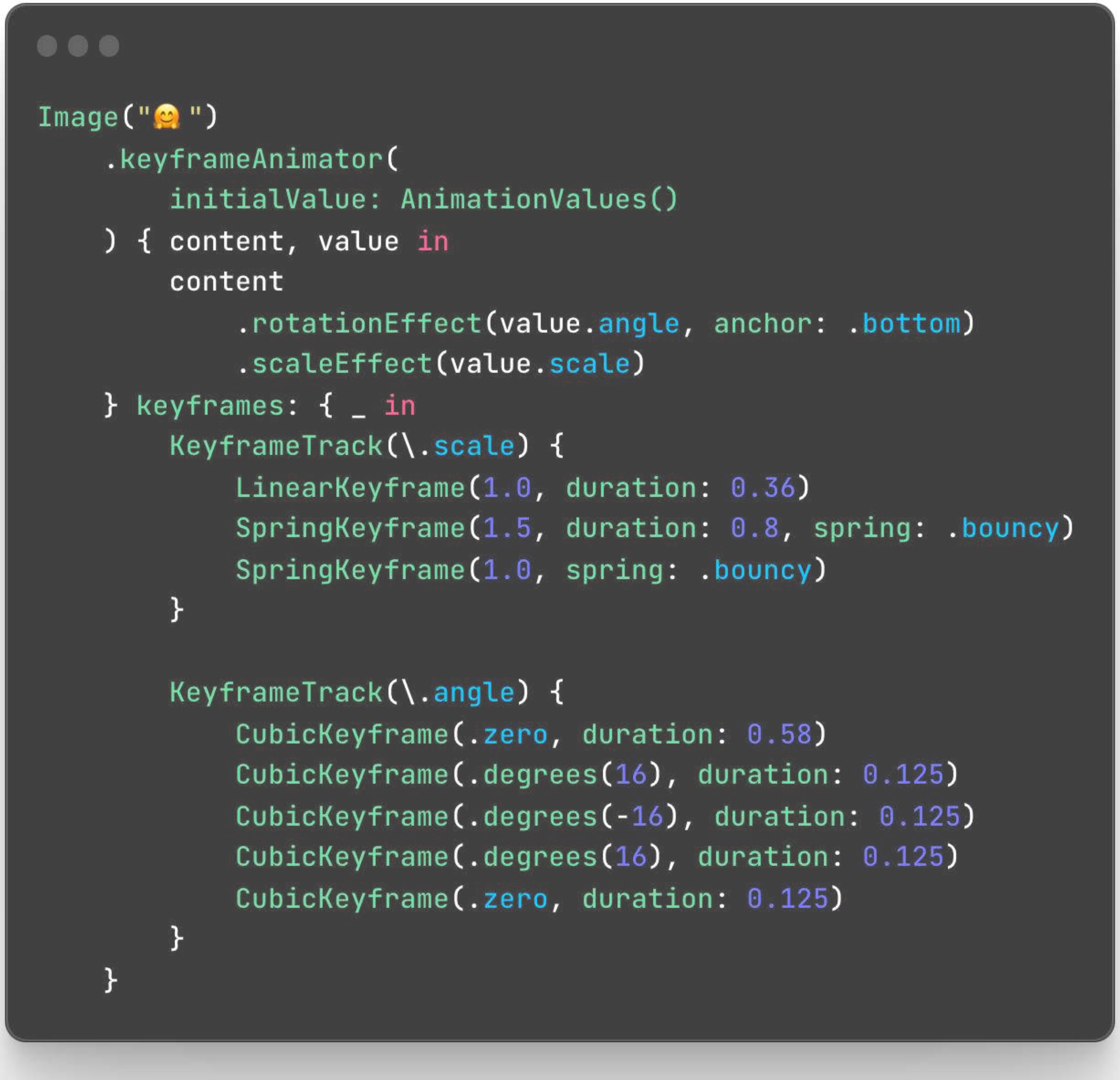
Full control of timings / durations

Most freedom, most complex

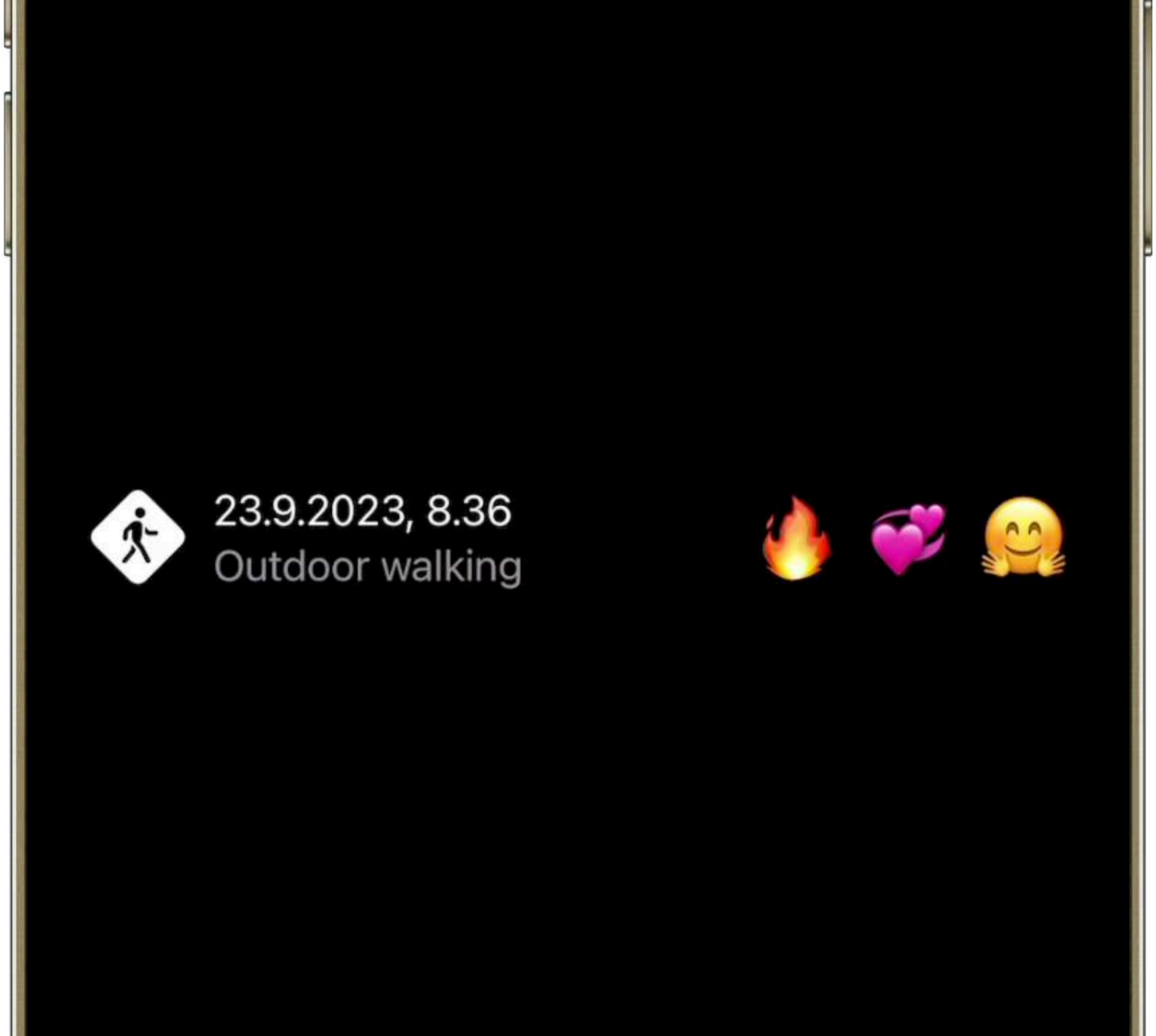
Keyframes 17



```swift
struct AnimationValues {
    var scale = 1.0
    var angle = Angle.zero
}
```



```swift
Image("🤗 ")
    .keyframeAnimator(
        initialValue: AnimationValues()
    ) { content, value in
        content
            .rotationEffect(value.angle, anchor: .bottom)
            .scaleEffect(value.scale)
    } keyframes: { _ in
        KeyframeTrack(\.scale) {
            LinearKeyframe(1.0, duration: 0.36)
            SpringKeyframe(1.5, duration: 0.8, spring: .bouncy)
            SpringKeyframe(1.0, spring: .bouncy)
        }

        KeyframeTrack(\.angle) {
            CubicKeyframe(.zero, duration: 0.58)
            CubicKeyframe(.degrees(16), duration: 0.125)
            CubicKeyframe(.degrees(-16), duration: 0.125)
            CubicKeyframe(.degrees(16), duration: 0.125)
            CubicKeyframe(.zero, duration: 0.125)
        }
    }
```



23.9.2023, 8.36

Outdoor walking

🔥 💞 🤗

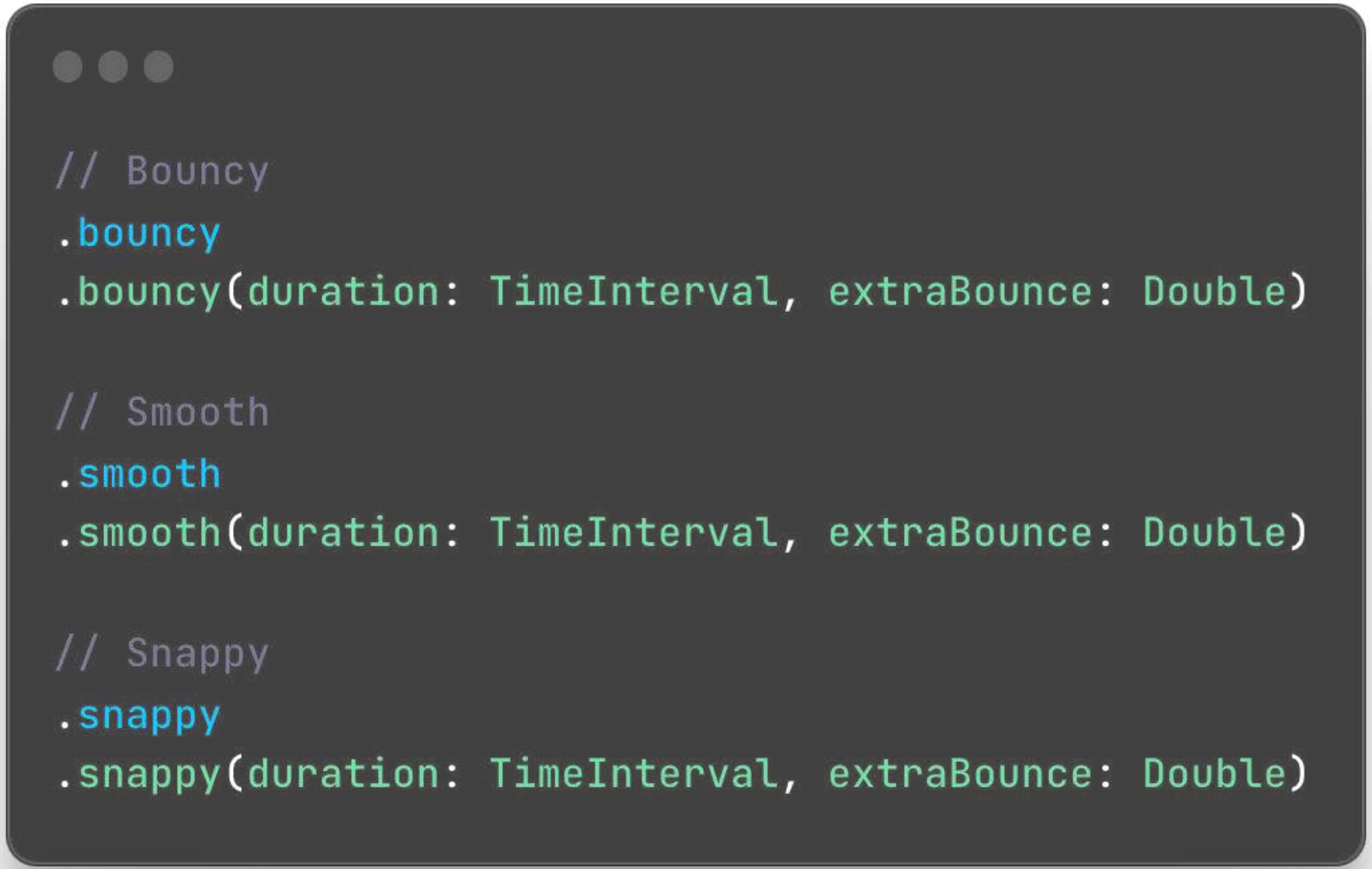
Springs 17

Object attached to a spring

Engaging, fun

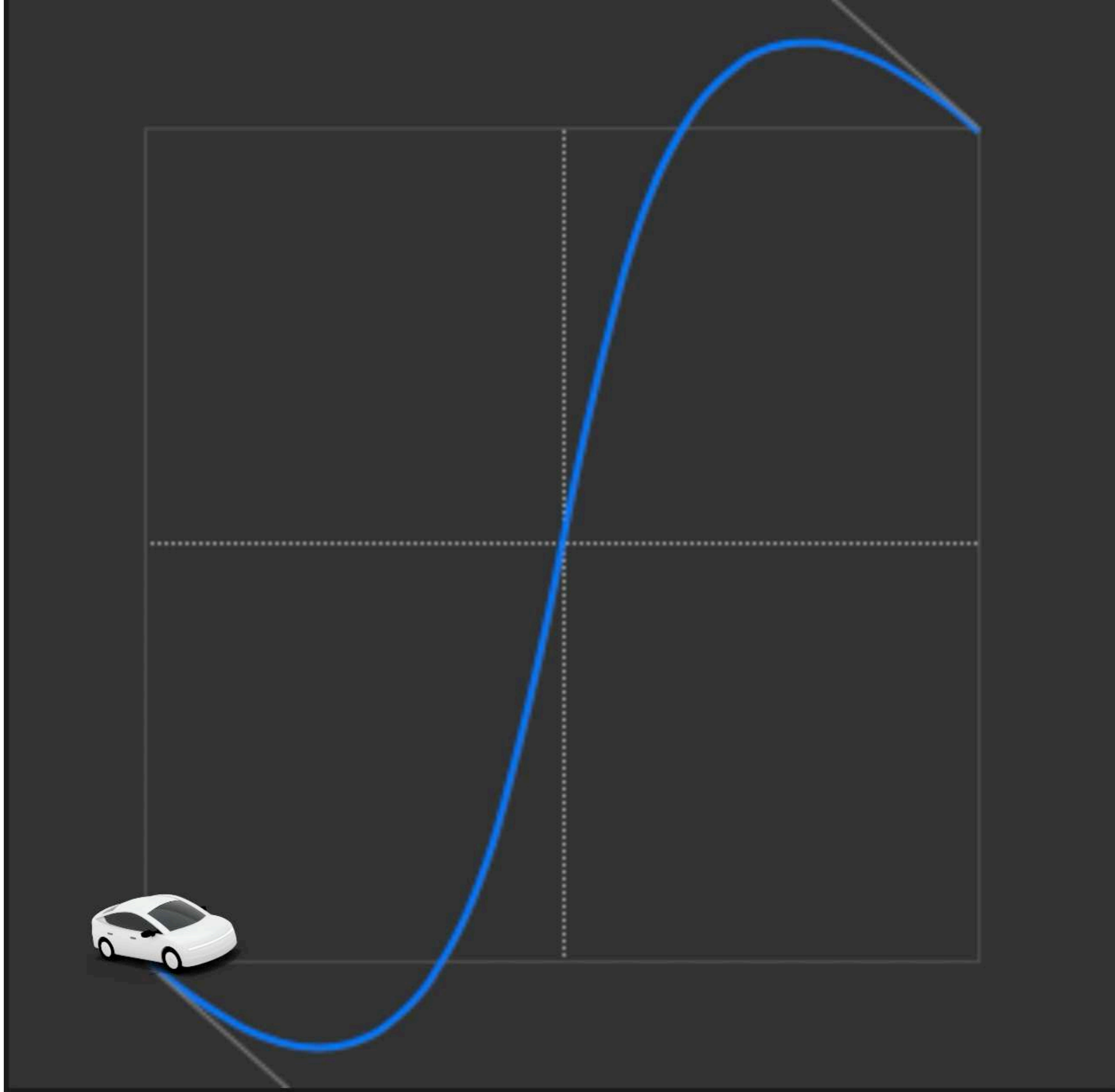
Used across the system

Springs 17



```
// Bouncy
.bouncy
.bouncy(duration: TimeInterval, extraBounce: Double)

// Smooth
.smooth
.smooth(duration: TimeInterval, extraBounce: Double)

// Snappy
.snappy
.snappy(duration: TimeInterval, extraBounce: Double)
```

Timing curves



Standard easings

Custom timing curves

Springs

Animations are cool, right?

Available on YouTube

21.06
zoom
Zoom
Spotify

HOME > TECH

Apple's iOS 7 Is Making People Sick

Alyson Shontell Sep 26, 2013, 5:28 PM GMT+2

The most recent complaint about Apple's new software, iOS 7, is that it makes users feel sick.

iOS 7 includes more animation than iOS 6. Apps fly open and closed, for example. Some people have said the excessive movement gives them motion sickness.

Note: This guy is car sick, he isn't an iOS 7 user. Flickr via ouvyt

An Apple message board spotted by The Verge lists a number of ...mptoms, including "severe vertigo" and the feeling of "car

...en users

App

A flurry
who upg
feeling v

in

e Apple users motion

LATEST TECHNOLOGY >

ENTERTAINMENT

Vampires, ad
broken heart a
metaphorical r
'El Paso, Elsew

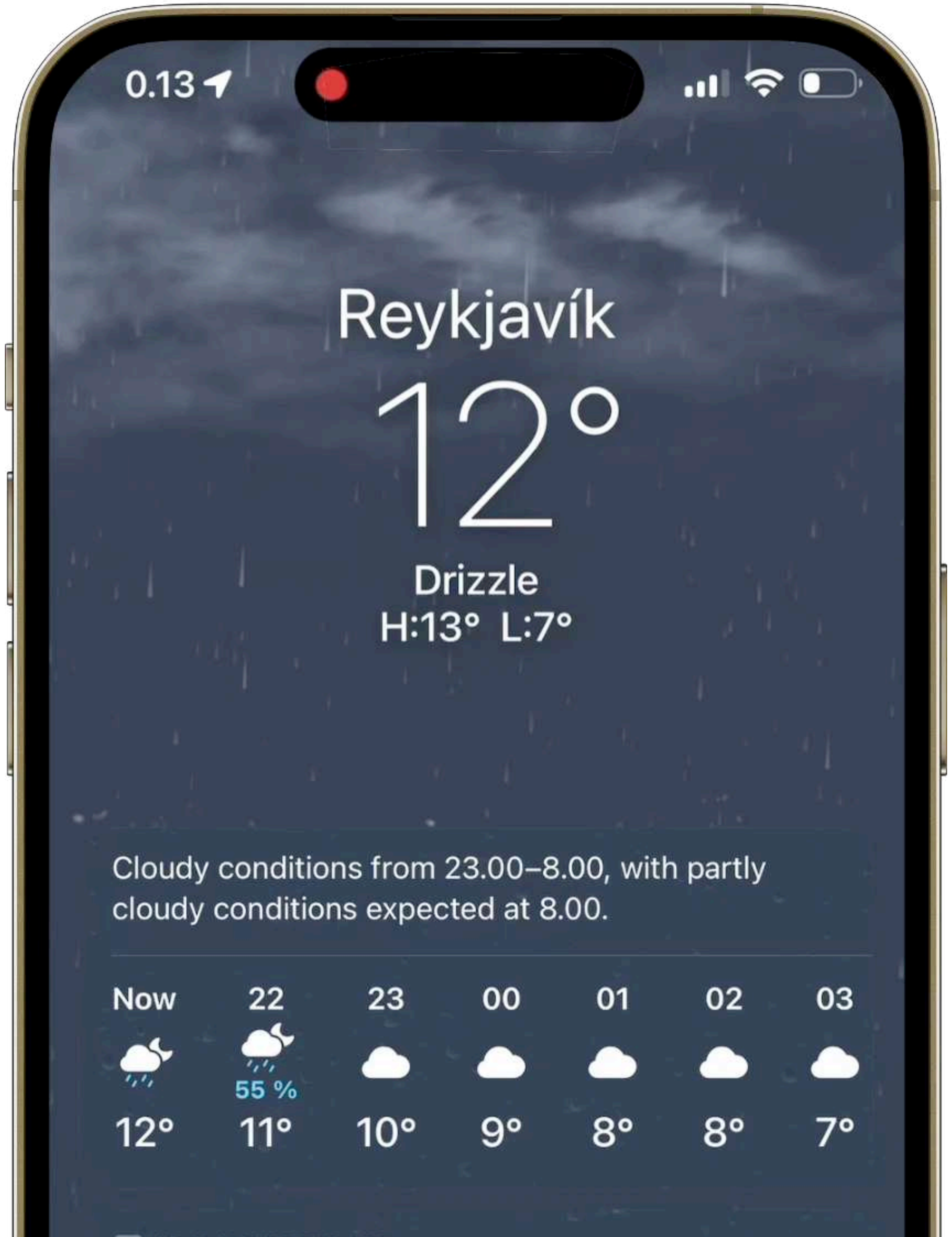
Sept. 27, 2023

TECHNOLOGY AND THE

Column: The wri
strike was the first
workplace battle b
humans and AI. Th
humans won

Which animations / motion can be distractive?

Frequent particle animations



Reykjavík

12°

Drizzle

H:13° L:7°

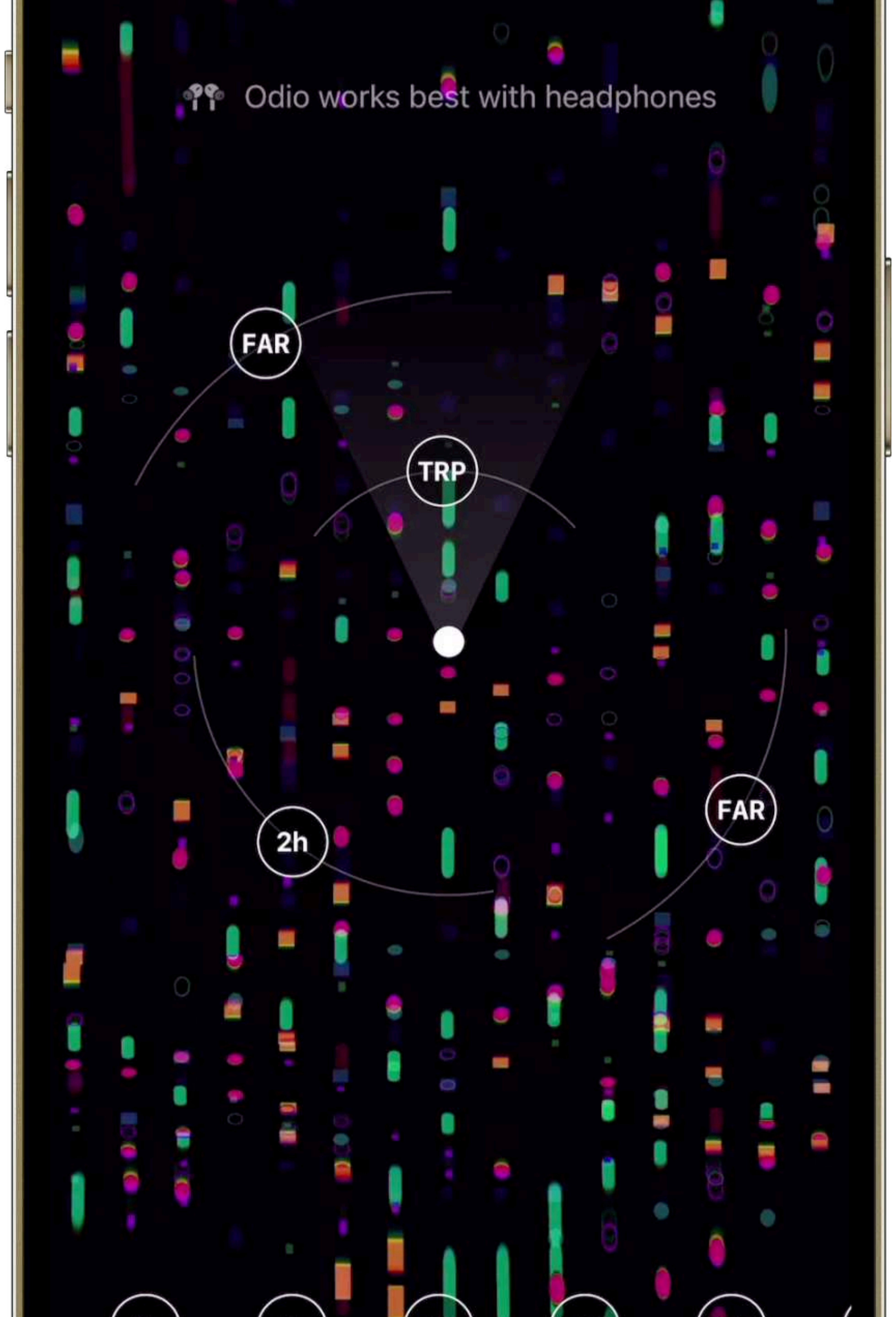
Cloudy conditions from 23.00–8.00, with partly cloudy conditions expected at 8.00.

Now	22	23	00	01	02	03
	55 %					
12°	11°	10°	9°	8°	8°	7°

Parallax

UIMotionEffect

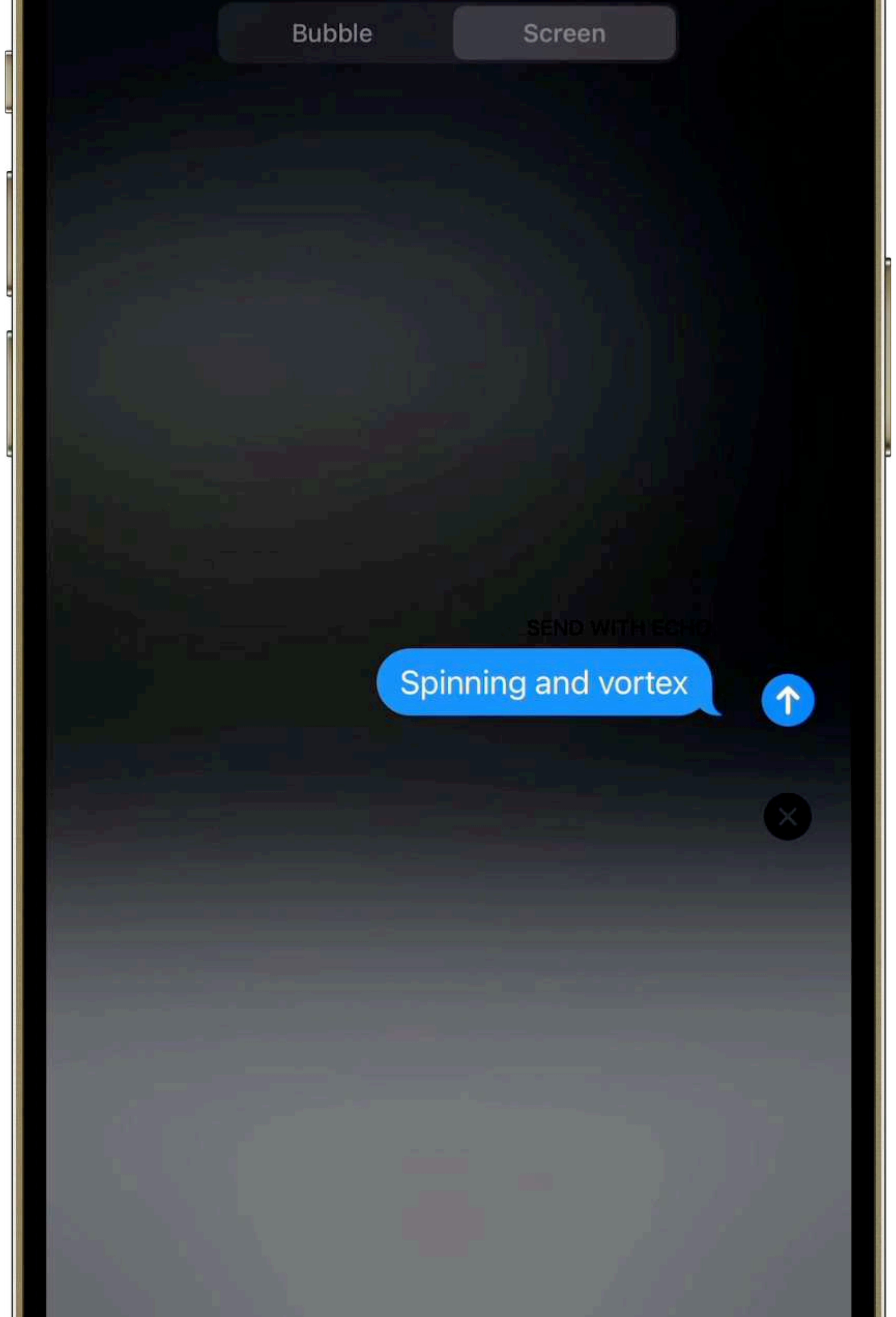
Background & Foreground effects


Odio works best with headphones

Spinning

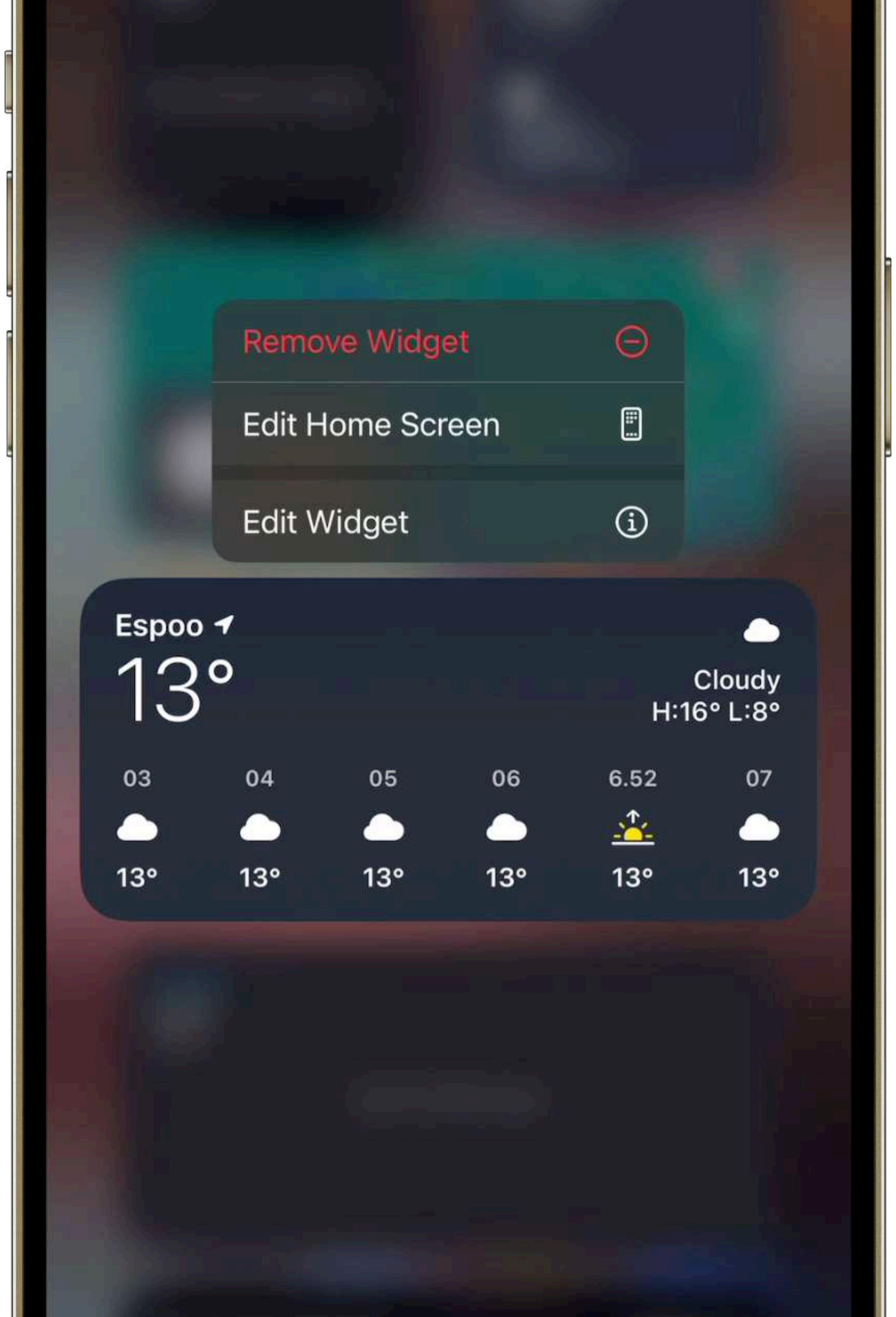
Rotating

Vortex



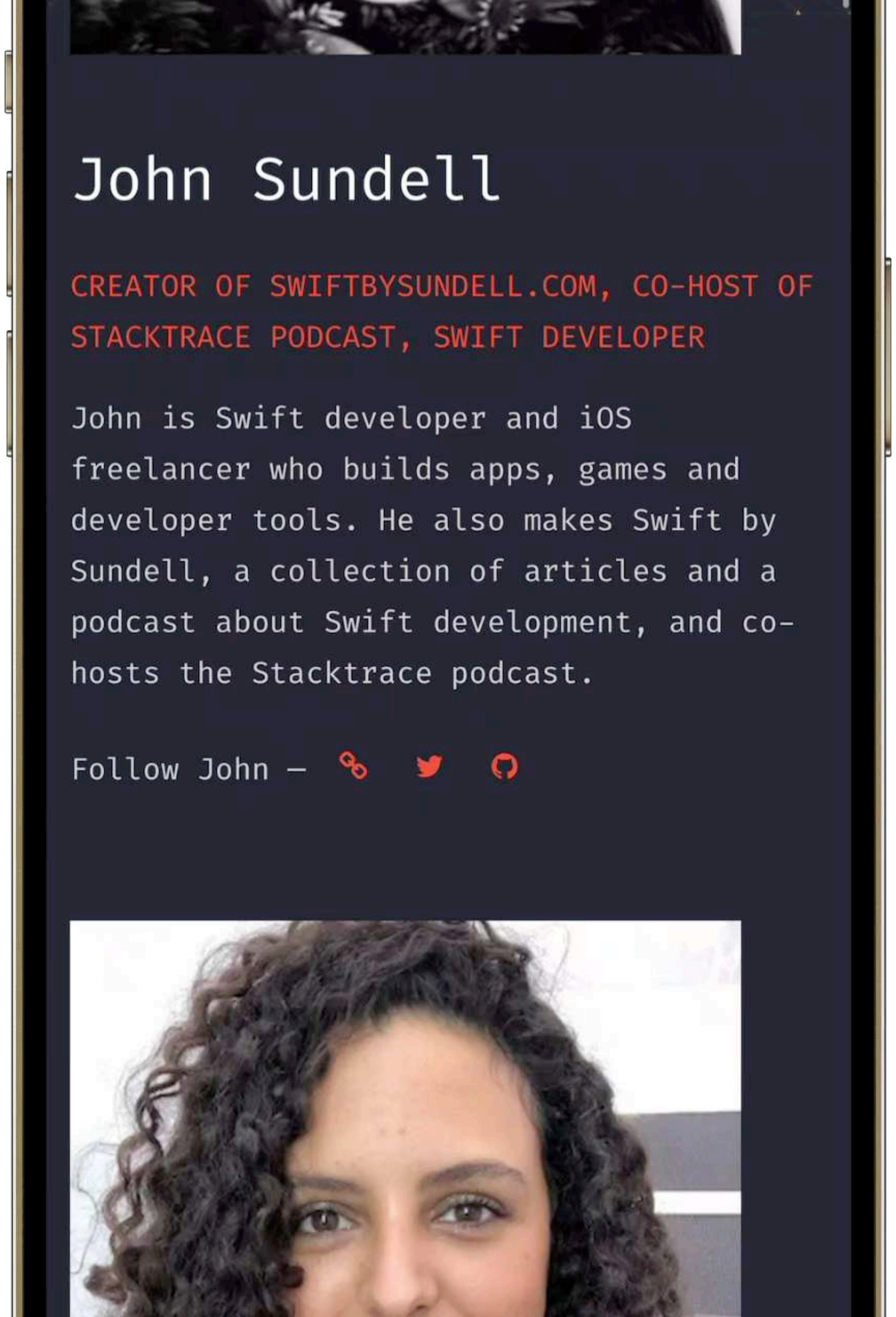
Spinning and vortex

Depth changes



Multi-sliding

Opposite direction to scroll



John Sundell

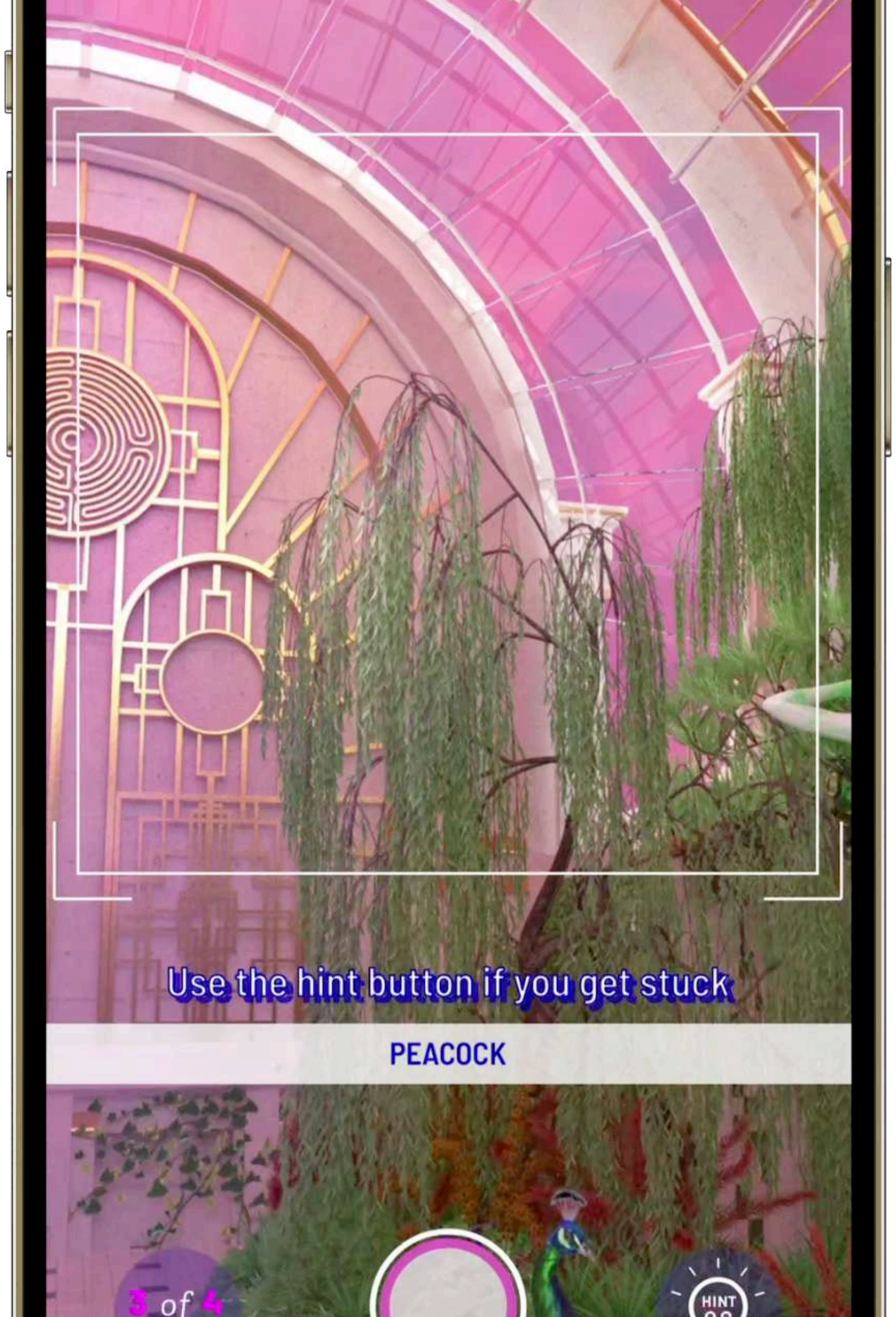
CREATOR OF SWIFTBYSUNDELL.COM, CO-HOST OF STACKTRACE PODCAST, SWIFT DEVELOPER

John is Swift developer and iOS freelancer who builds apps, games and developer tools. He also makes Swift by Sundell, a collection of articles and a podcast about Swift development, and co-hosts the Stacktrace podcast.

Follow John —

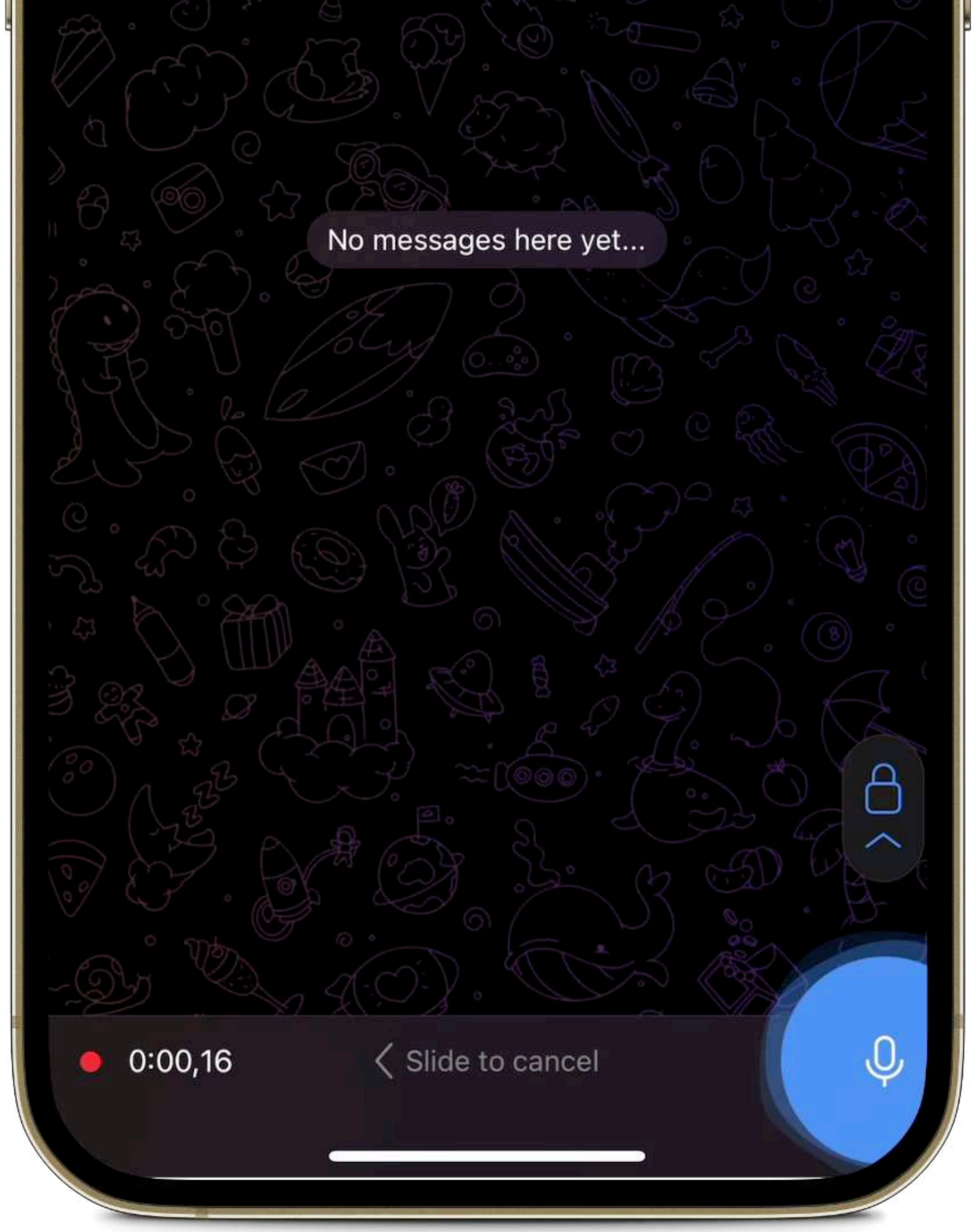
Intense Animations

Glitching and flicking



Use the hint button if you get stuck

Blinking animation



No messages here yet...

0:00,16

Slide to cancel

What can we do better?

Apple Developer Videos

Human Interface Guidelines (HIG)

Web Content Accessibility Guidelines (WCAG)

Apple Design Awards

3 easy-to-follow guides

1 Pause, Play, Hide

Autoplaying GIFs

Video previews

Animated illustration

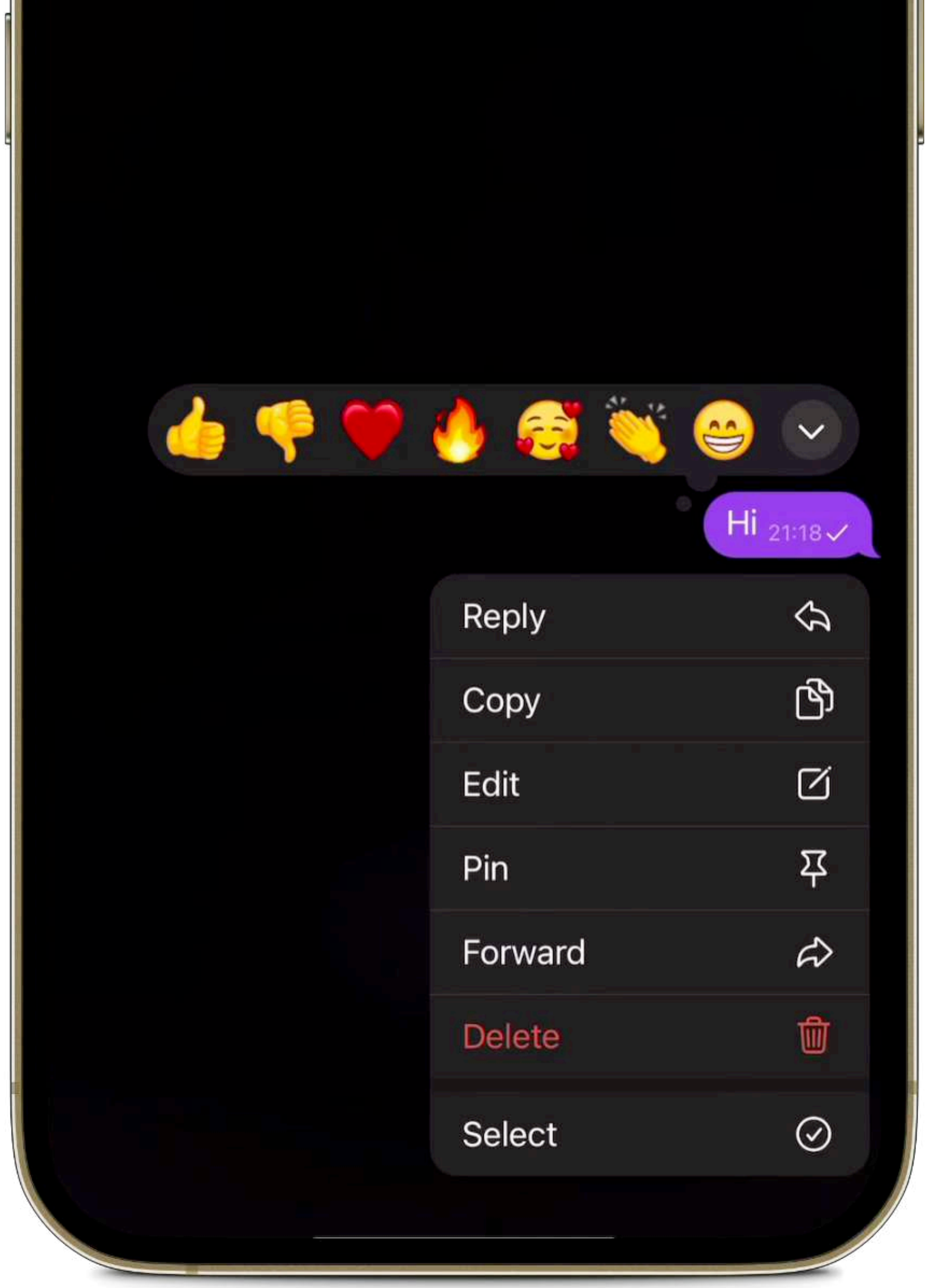
2 Don’t flash more than 3 times in 1 s

Distracting

Can cause seizures

Replace flashing

3 Provide a way to disable animations



Hi 21:18

User-initiated

Replace bouncy animation

We know what, but not how

Reduce motion

If you don’t want visual motion on your iPhone screen, you can stop or reduce the movement of some screen elements, such as parallax effects, screen transitions, animations, full-screen effects, and more.

Apple Support (modified)

Settings > Accessibility > Motion

Auto-Play Message Effects

Auto-Play Video Previews

Auto-Play Animated Images

Dim Flashing Lights

Limit Frame Rate

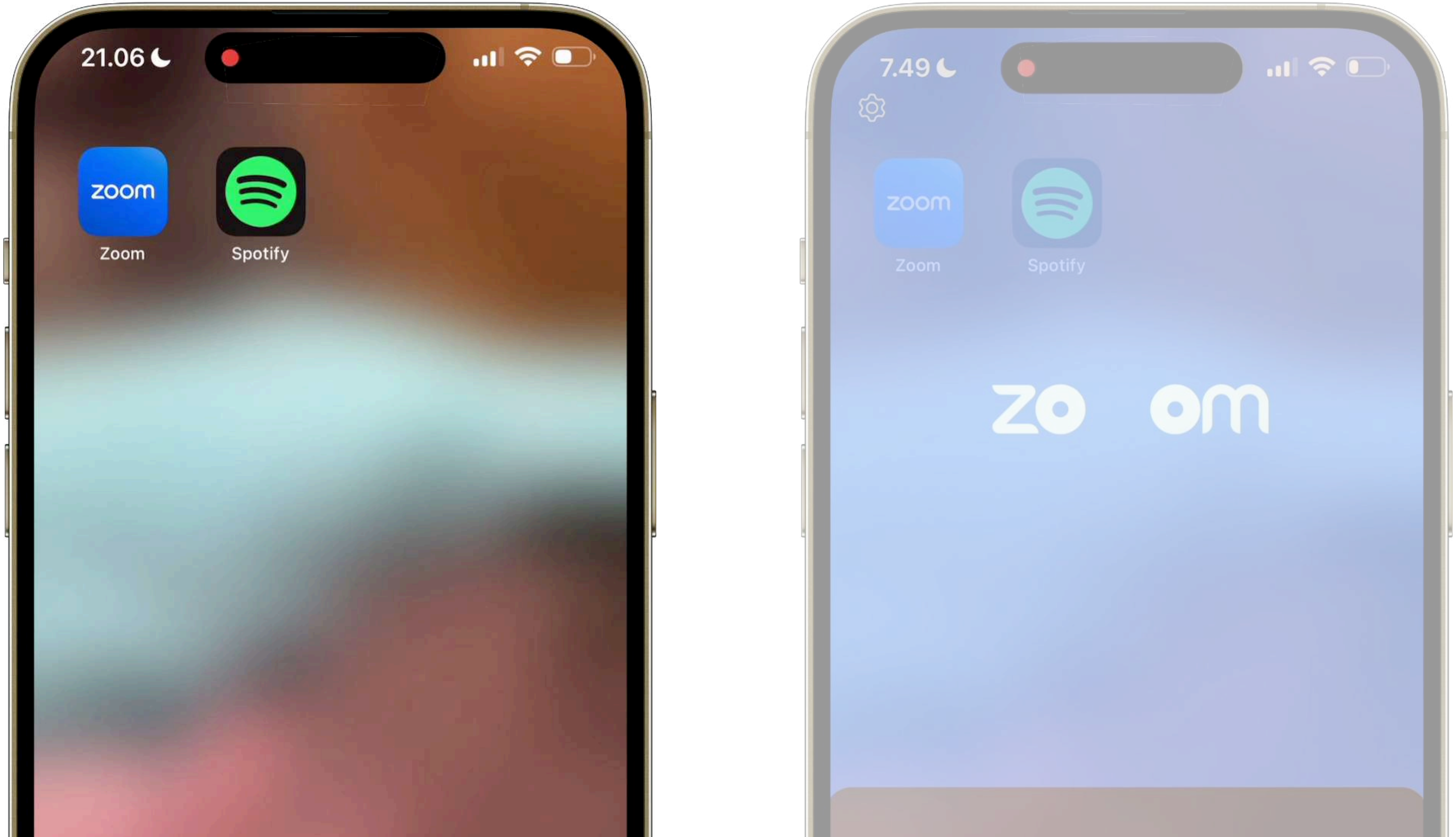
Reduce Motion

Doesn’t remove all animations

Provides a reduced behaviour

Possible to set on per-app basis

E.g. image-switching technique

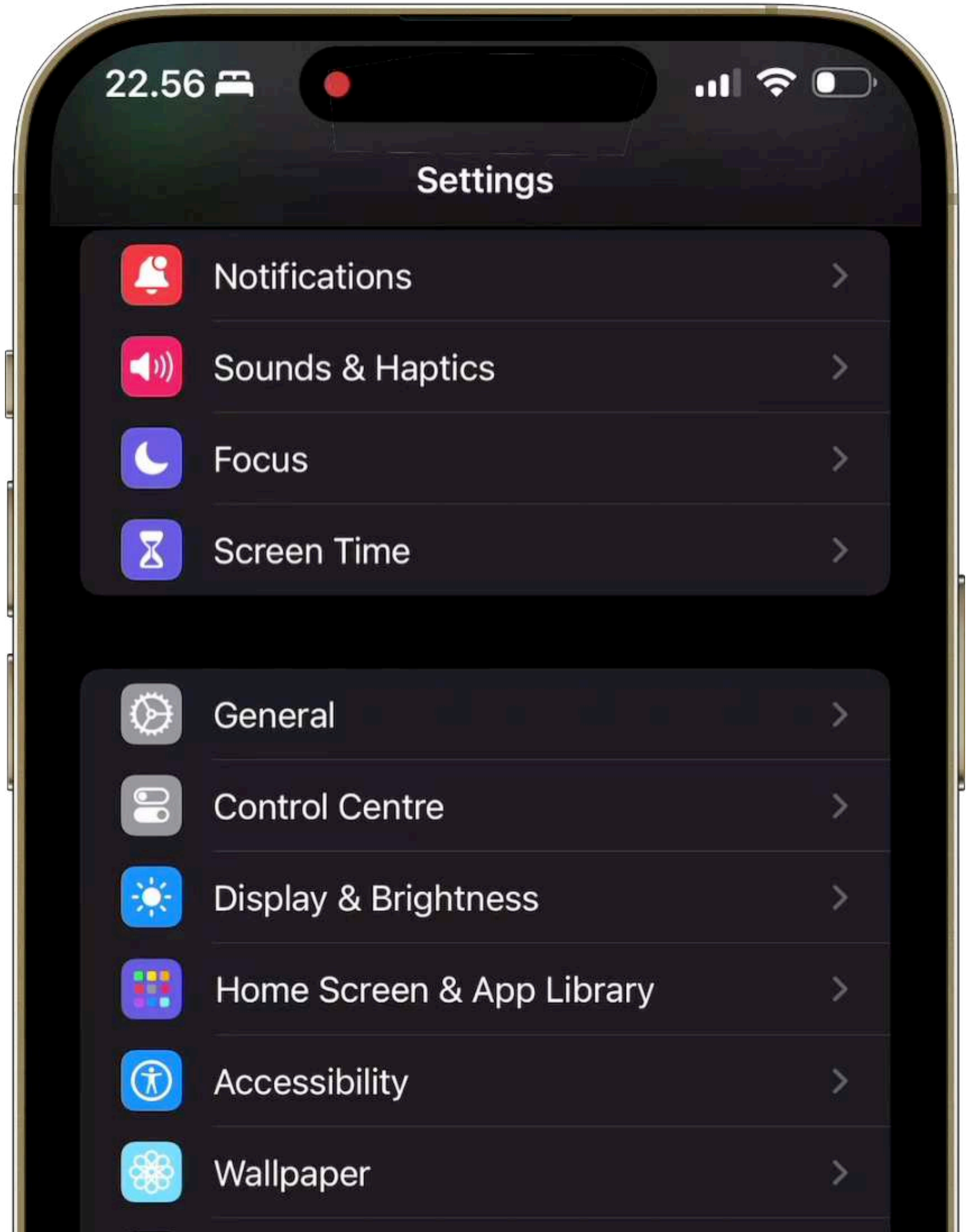

21.06
zoom
Zoom
Spotify
7.49
zoom
Zoom
Spotify
zo om

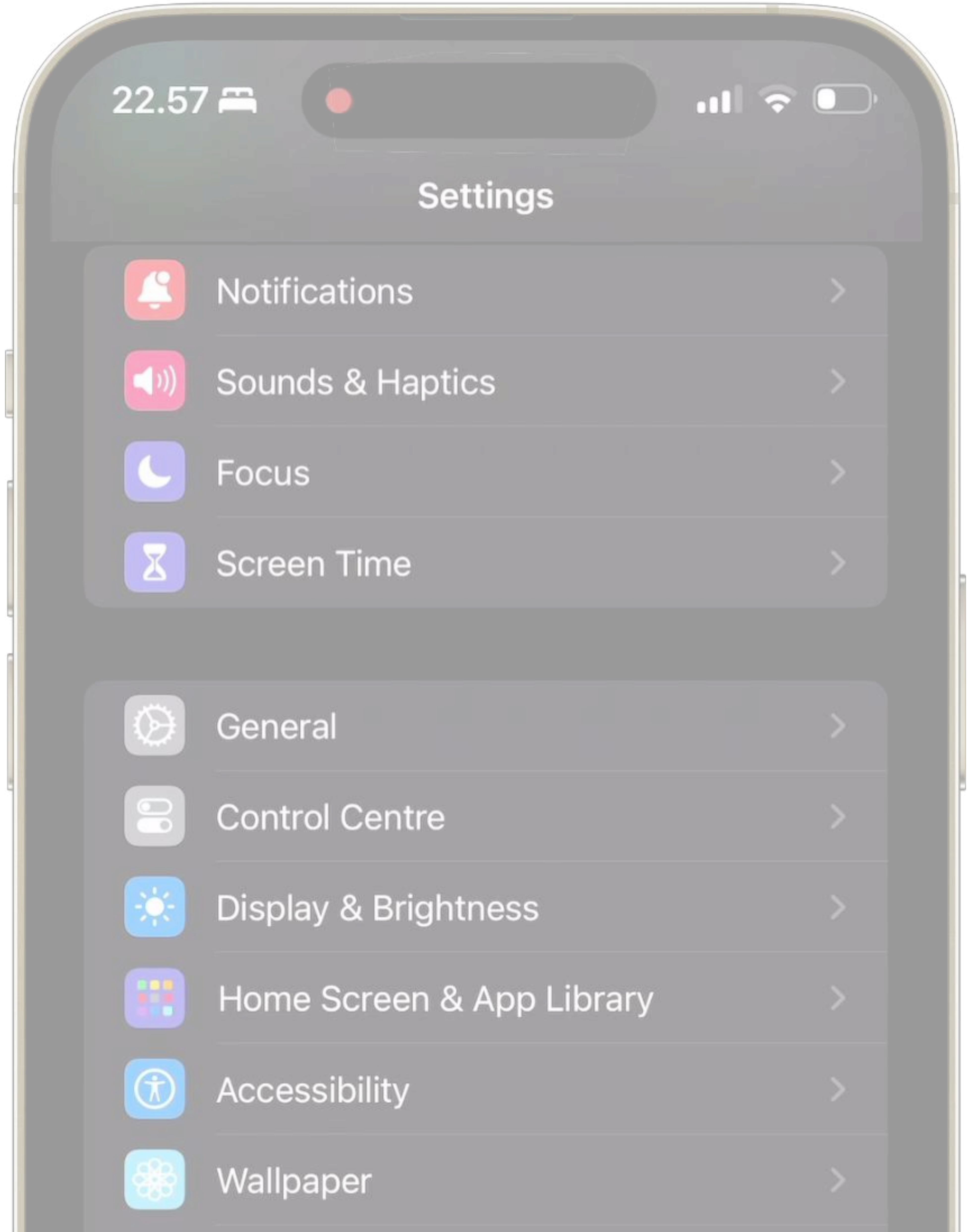
Prefer Cross-Fade Transitions

Replace slide-in / slide-out transitions

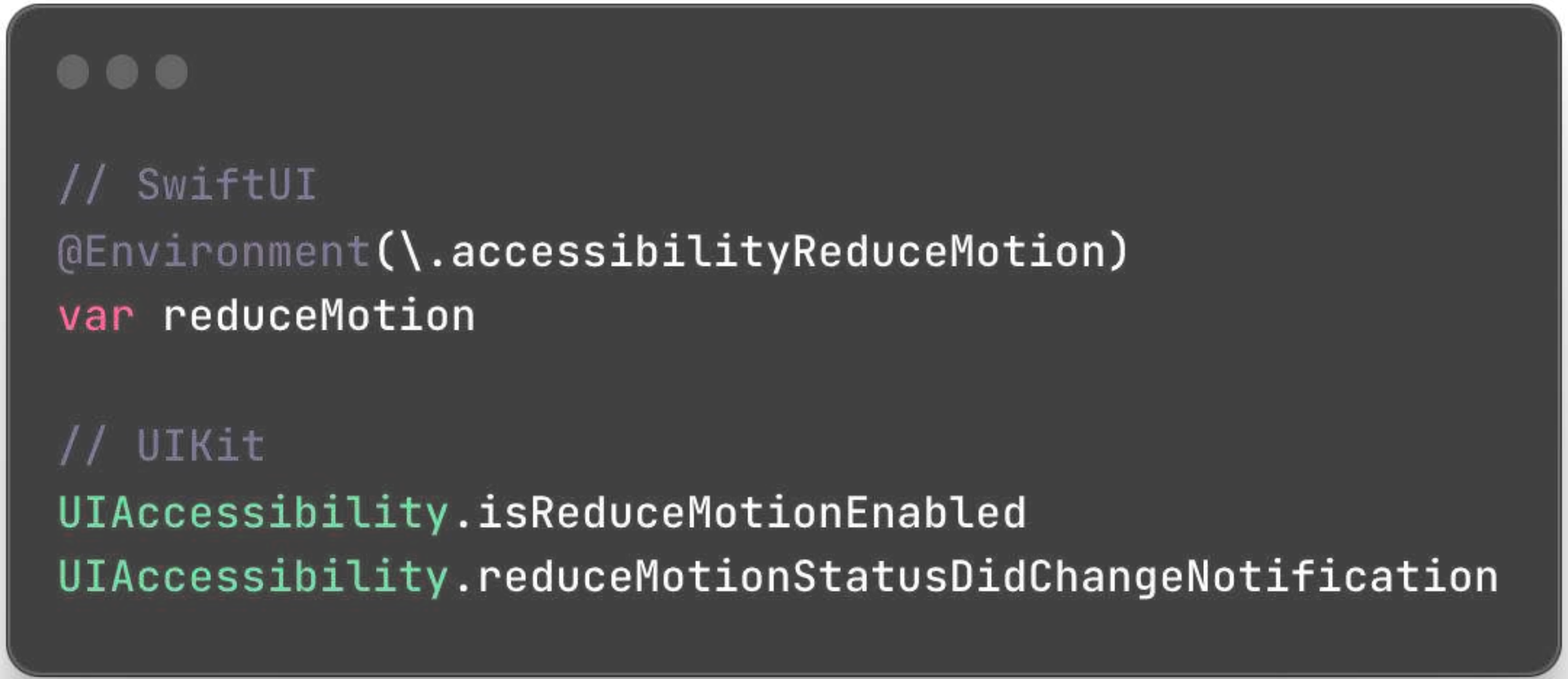
Use cross-fade

For free with NavigationLink

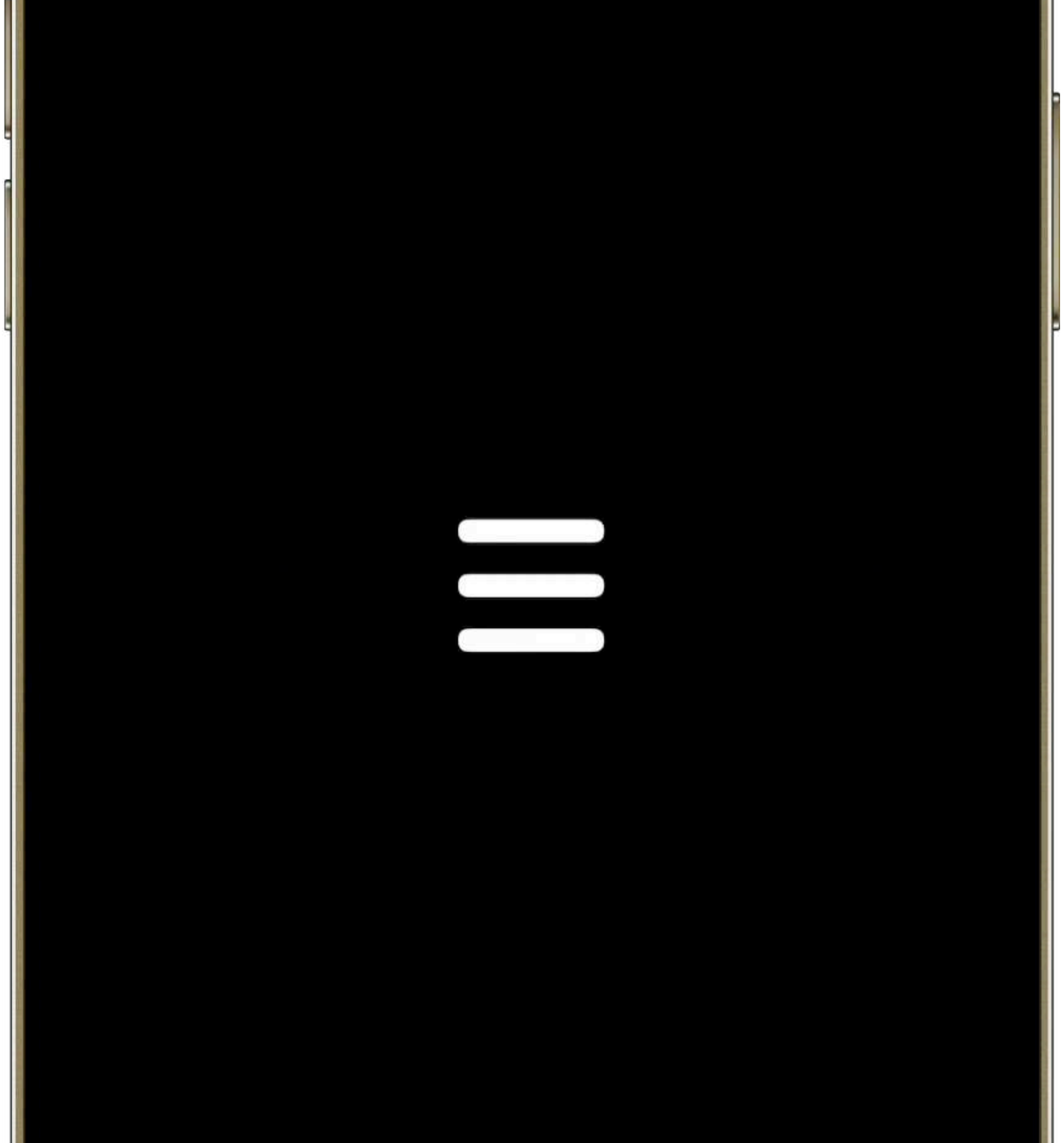

22.56
Settings
Notifications
Sounds & Haptics
Focus
Screen Time
General
Control Centre
Display & Brightness
Home Screen & App Library
Accessibility
Wallpaper


22.57
Settings
Notifications
Sounds & Haptics
Focus
Screen Time
General
Control Centre
Display & Brightness
Home Screen & App Library
Accessibility
Wallpaper

Checking Reduce Motion

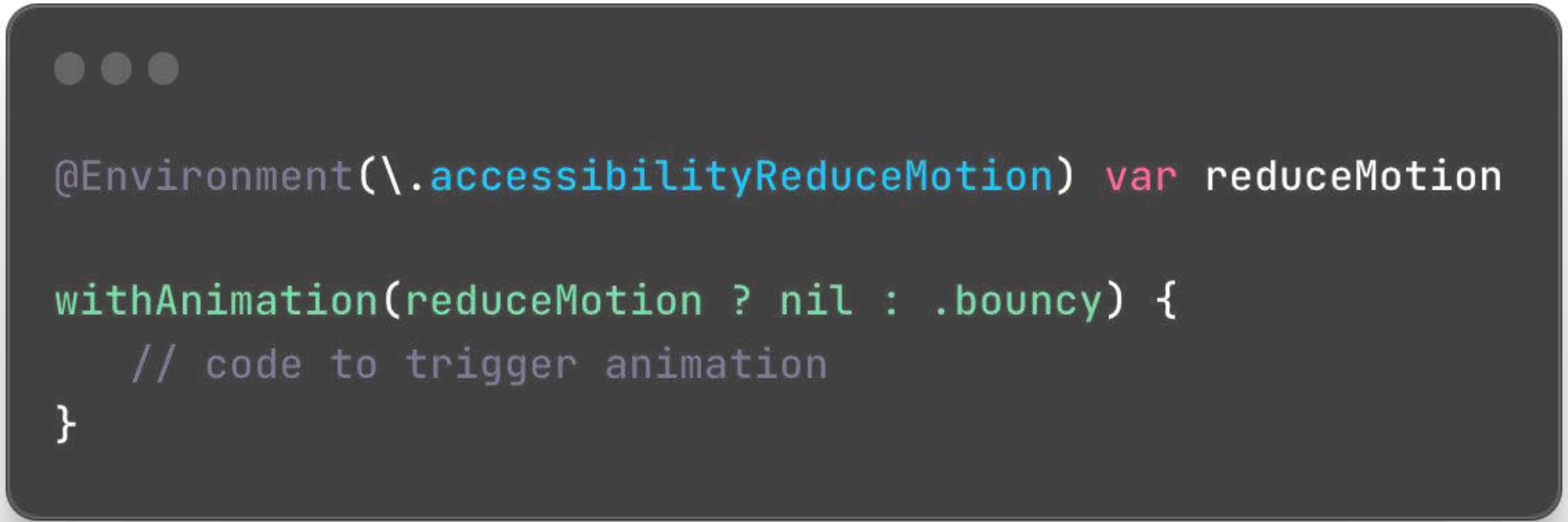


```
// SwiftUI
@Environment(\.accessibilityReduceMotion)
var reduceMotion

// UIKit
UIAccessibility.isReduceMotionEnabled
UIAccessibility.reduceMotionStatusDidChangeNotification
```



Adopting Reduce Motion

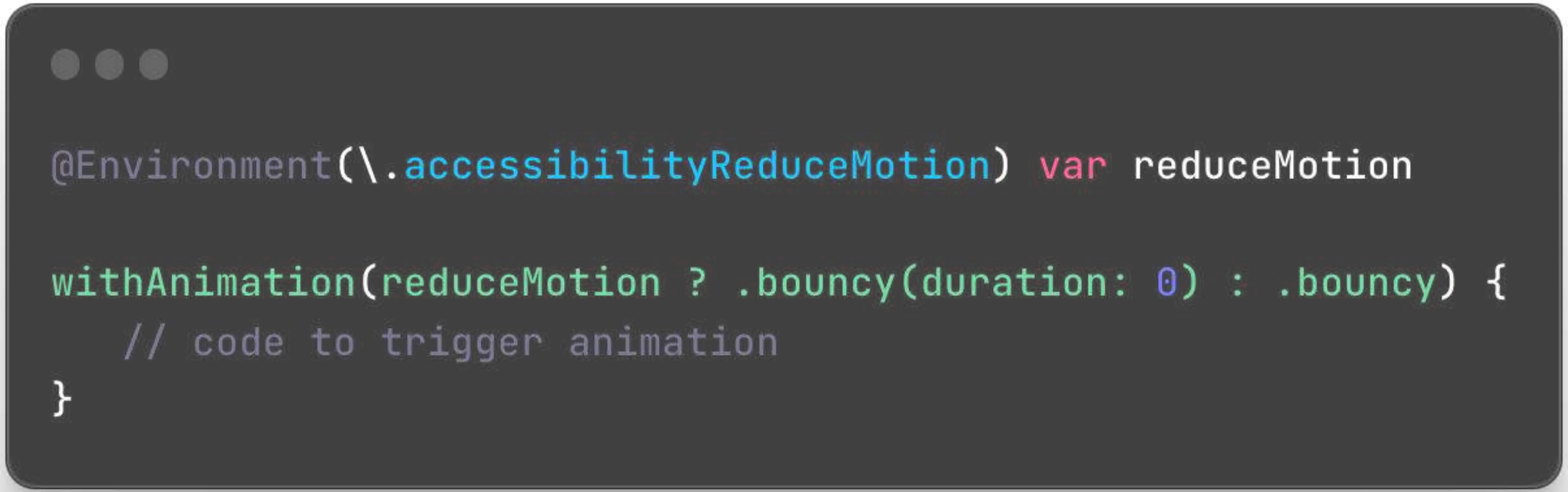
Remove animation



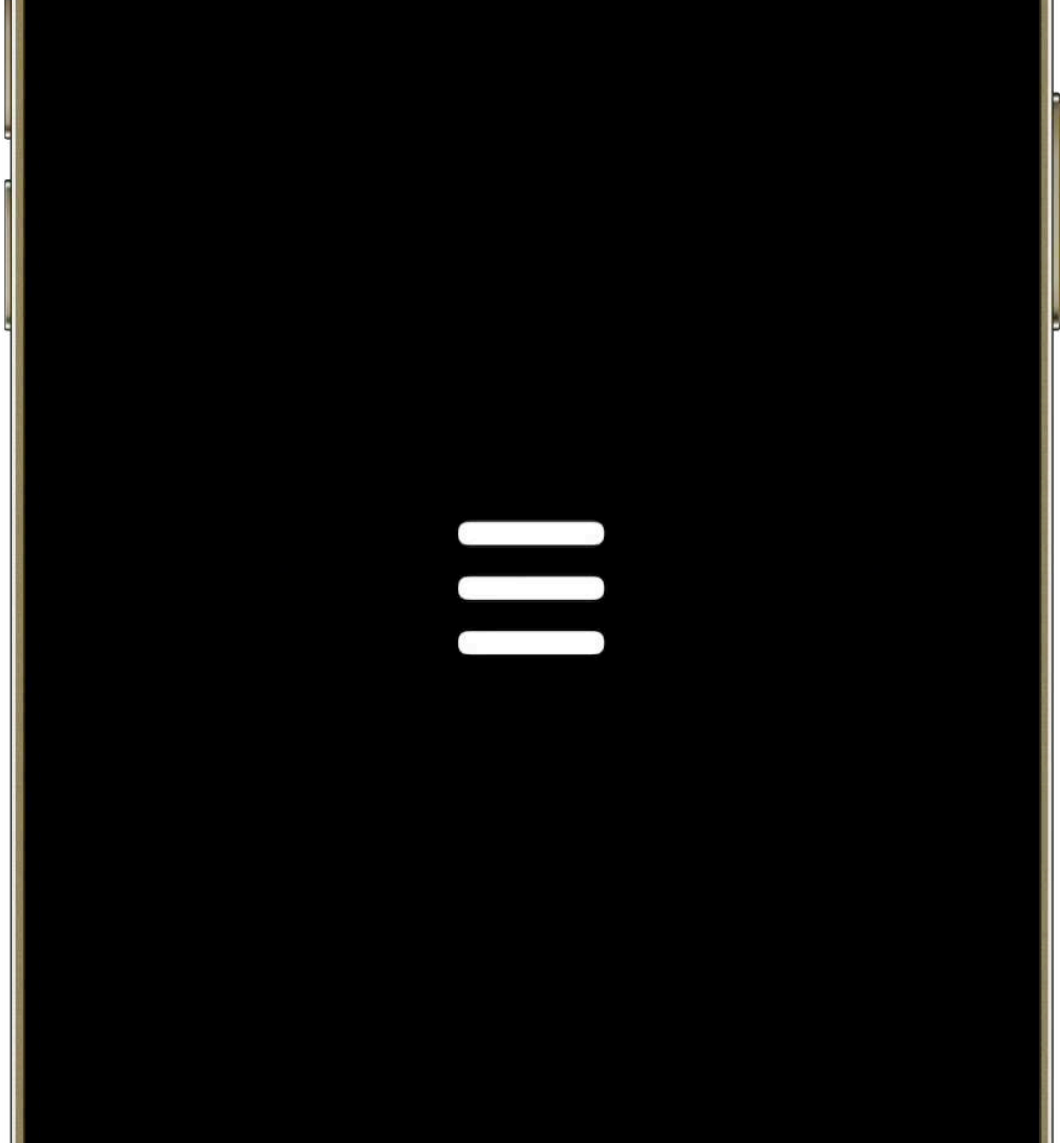
```
@Environment(\.accessibilityReduceMotion) var reduceMotion

withAnimation(reduceMotion ? nil : .bouncy) {
    // code to trigger animation
}
```

Adopting Reduce Motion

Reduce animation duration

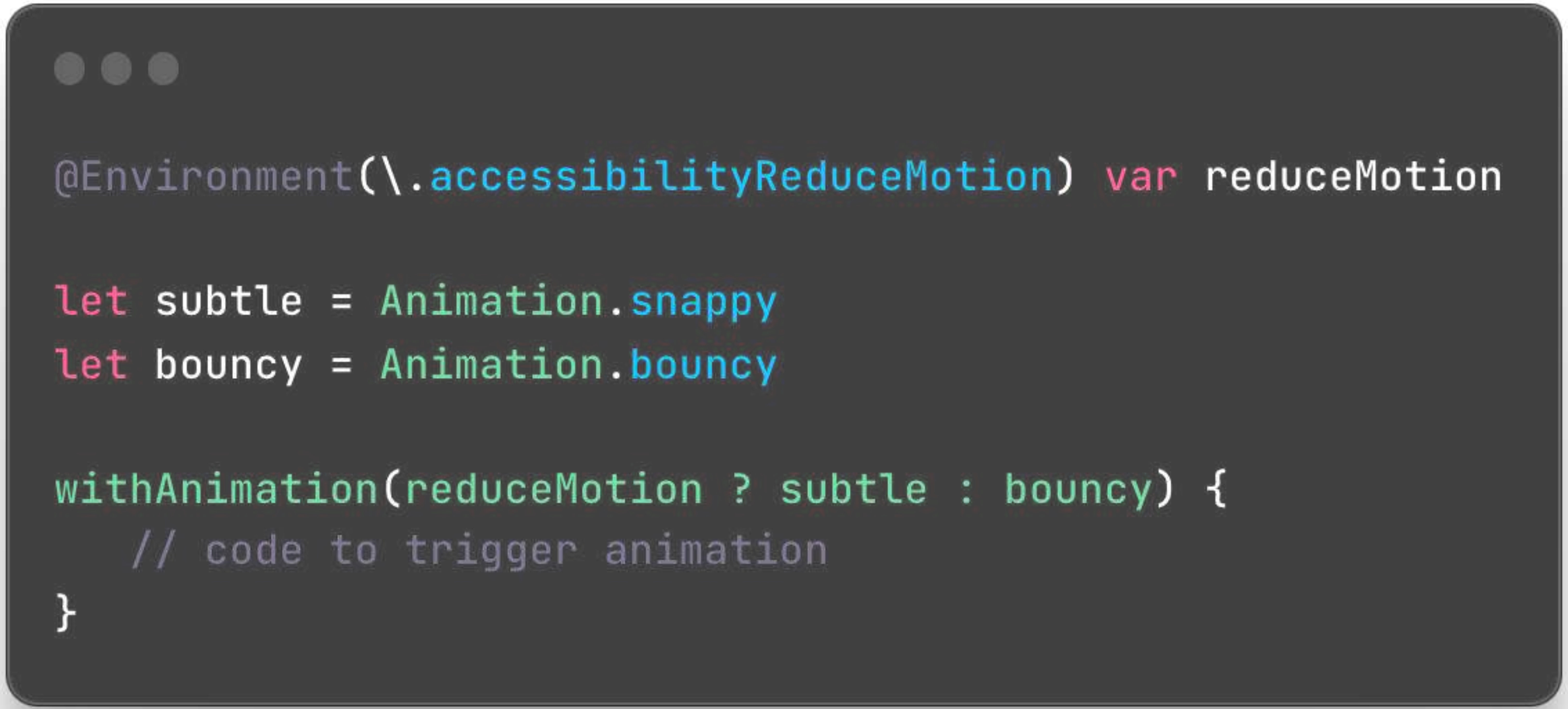


```swift
@Environment(\.accessibilityReduceMotion) var reduceMotion

withAnimation(reduceMotion ? .bouncy(duration: 0) : .bouncy) {
    // code to trigger animation
}
```

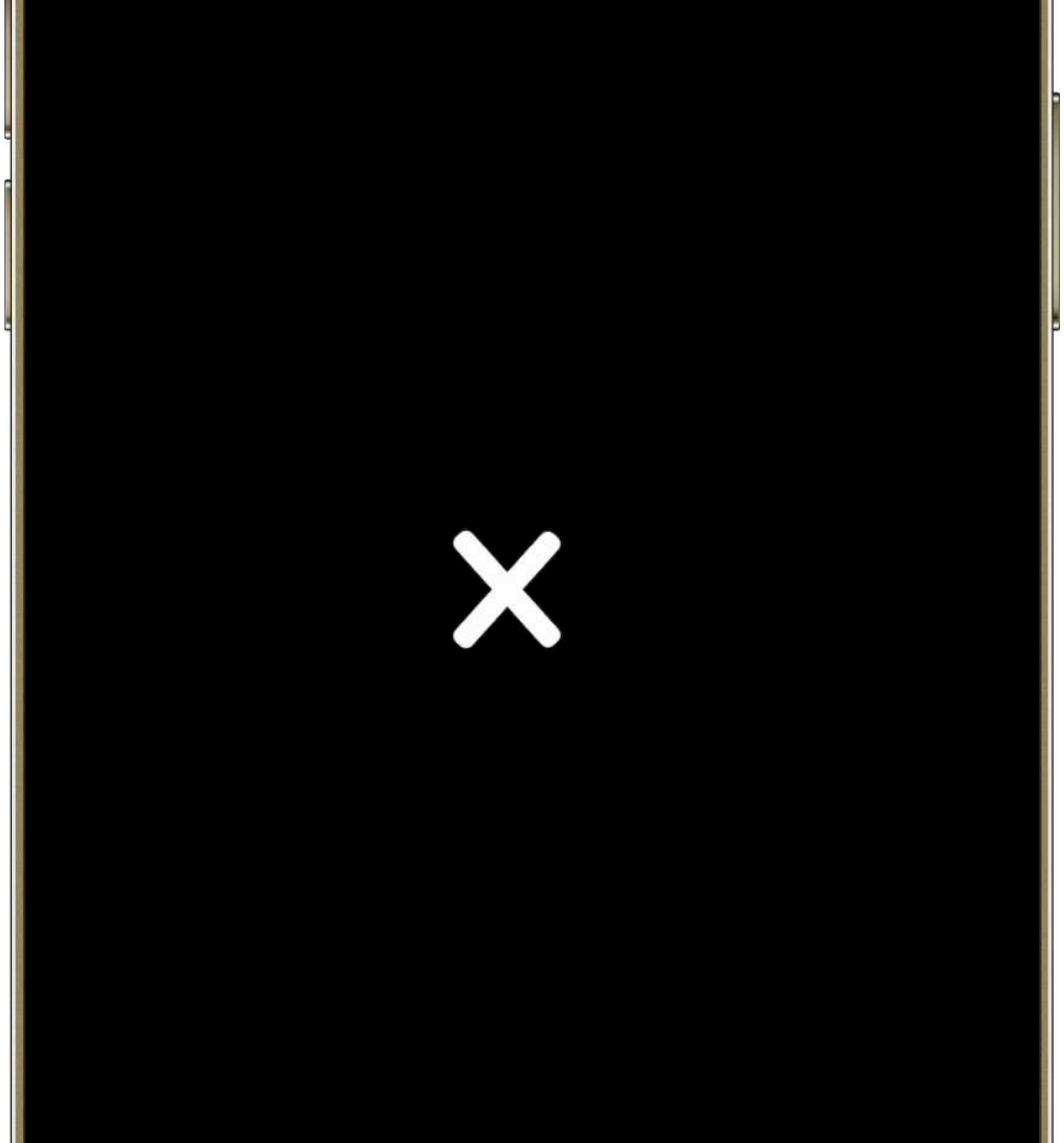


Adopting Reduce Motion

Replace animation



```
@Environment(\.accessibilityReduceMotion) var reduceMotion

let subtle = Animation.snappy
let bouncy = Animation.bouncy

withAnimation(reduceMotion ? subtle : bouncy) {
    // code to trigger animation
}
```



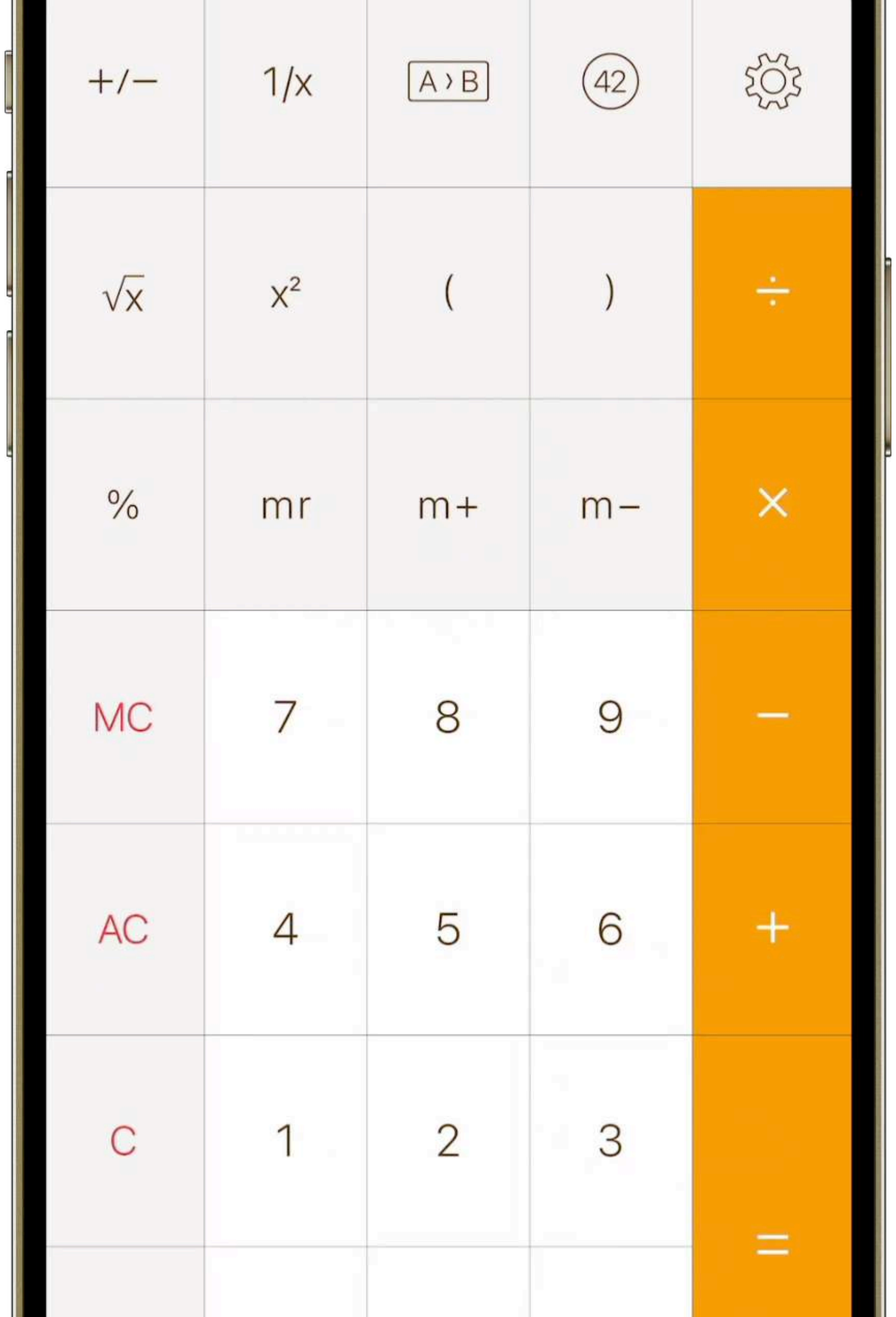
One more thing…

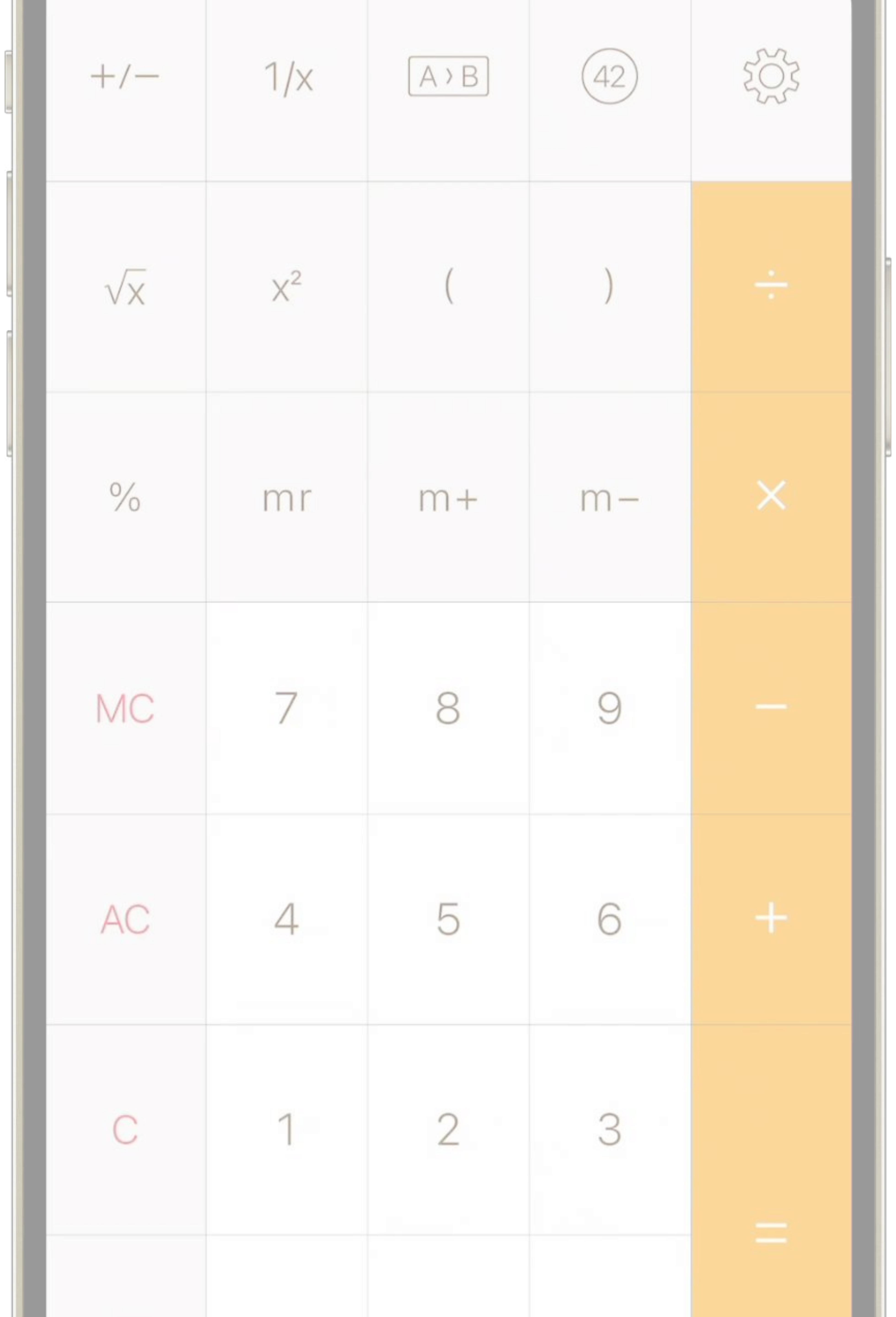
ReduceMotion doesn’t stop all movements

Custom In-App settings

Specific to your app

Reduce all sudden movements


+/−
1/x
A › B
42
√x
x²
(
)
÷
%
mr
m+
m−
×
MC
7
8
9
−
AC
4
5
6
+
C
1
2
3
=


+/−
1/x
A › B
42
√x
x²
(
)
÷
%
mr
m+
m−
×
MC
7
8
9
−
AC
4
5
6
+
C
1
2
3
=

VoiceOver

VoiceOver is a **gesture-based screen reader** that **enables people** to experience the interface on their devices **without having to see** the screen.

Apple Documentation

Settings > Accessibility > VoiceOver

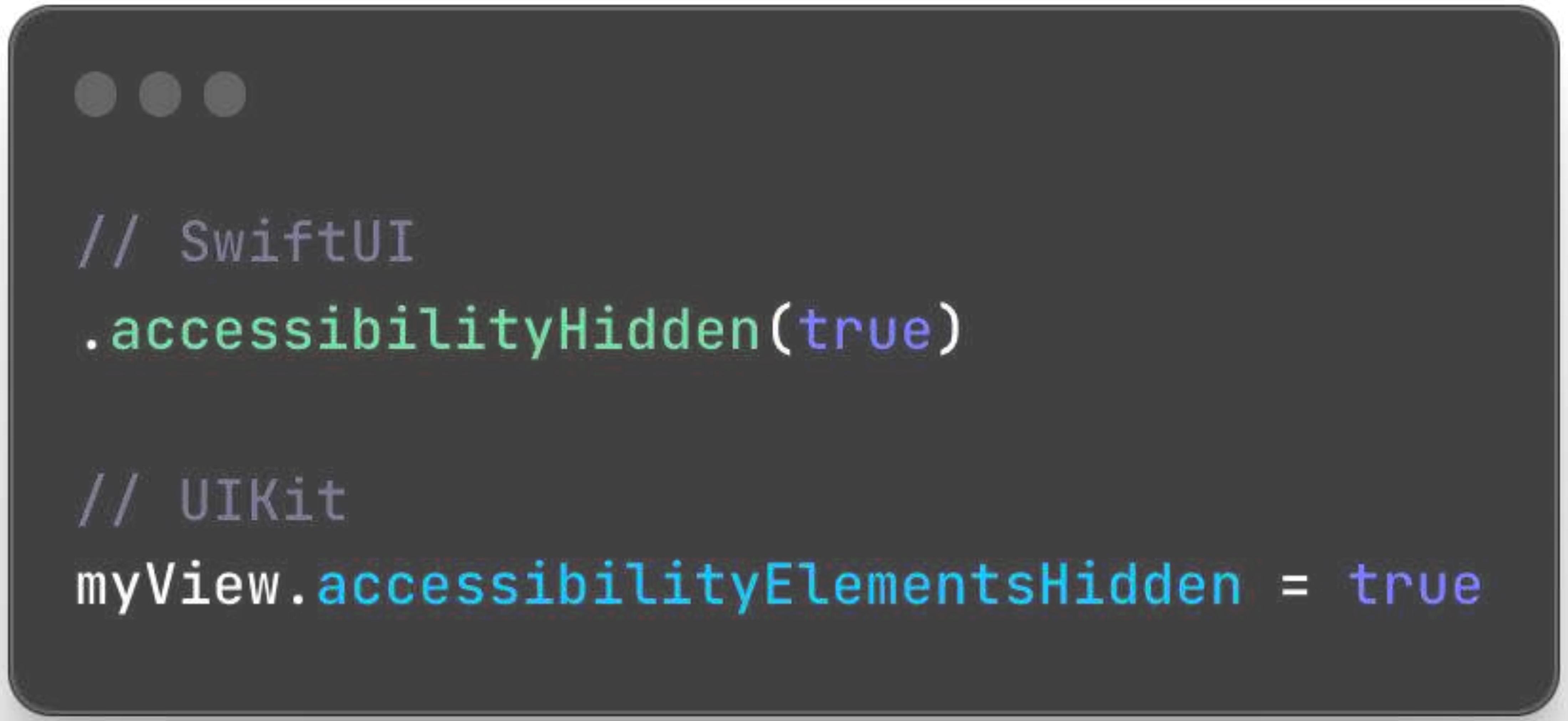
VoiceOver mentions all layers

Hide decorative animations

Add labels to meaningful animations

Adopting VoiceOver

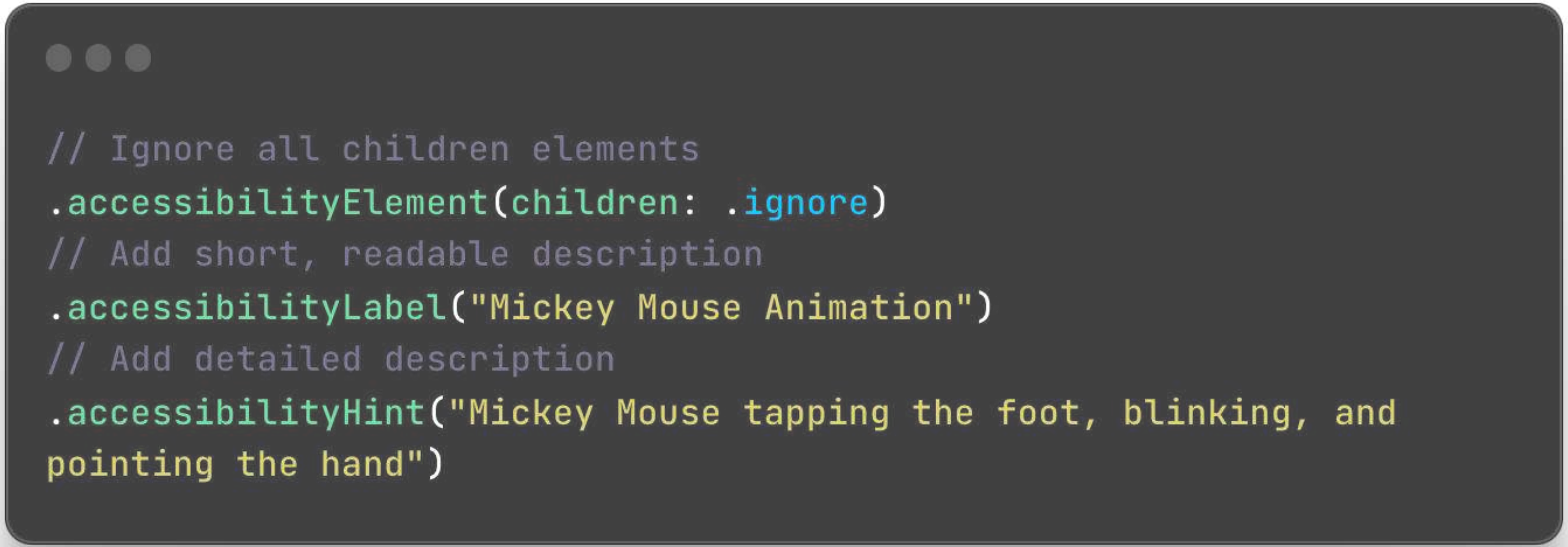
Hide decorative animation



```
// SwiftUI
.accessibilityHidden(true)

// UIKit
myView.accessibilityElementsHidden = true
```

Adopting VoiceOver

Add VoiceOver label and hint



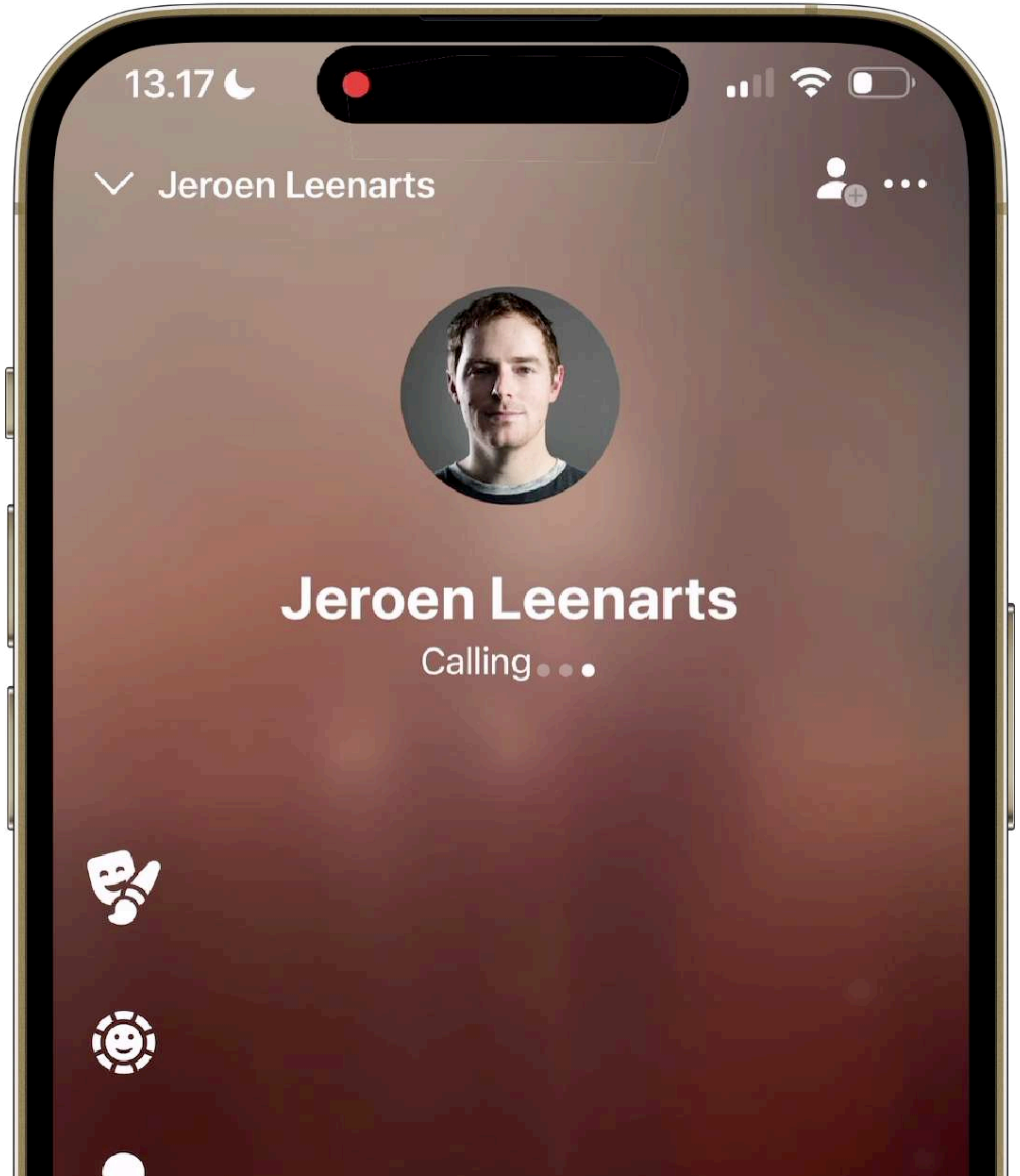
```
// Ignore all children elements
.accessibilityElement(children: .ignore)
// Add short, readable description
.accessibilityLabel("Mickey Mouse Animation")
// Add detailed description
.accessibilityHint("Mickey Mouse tapping the foot, blinking, and
pointing the hand")
```

Haptic Feedback

Mimick physical touch and drag

Haptic engine - vibrations

Emulate sound when phone is silent


13.17
Jeroen Leenarts
Jeroen Leenarts
Calling

Best practices

Follow basic accessibility guidelines

Screen flashing can cause seizures

Alternate effects for ReduceMotion

Excessive motion can cause discomfort

Examples: Parallax, sliding animations

There are 3 easy wins:

1. Pause, Play, Hide
2. Don’t flash more than 3 times in 1 s
3. Provide a way to disable animations

Practical tips

Be mindful of motion usage

Use NavigationLink (when possible)

Respect ReduceMotion

Think in VoiceOver

Ask yourself these questions

Can this animation cause discomfort?

Can people with motion sensitivities enjoy my app?

What if the user’s reduced motion setting is on?

Think
of accessibility
before you animate

Resources

Apple Design Awards: Inclusivity Winners

Apple Human Interface: Accessibility

Apple Developer Videos

Responsive Design For Motion

Playing Haptics

References

Human Interface Guidelines

Principles of motion

Writing Great Accessibility Labels

Supporting VoiceOver in your app

Motion sickness: Nausea

Creating usability with motion

Understanding the intuition of easing

Reduce screen motion on Apple devices

Responsive design for motion

Animations are nice, right?

Thank you

@amos_gyamfi

@stefanjblos