From UIKit

Veronica Ray

Senior Software Engine

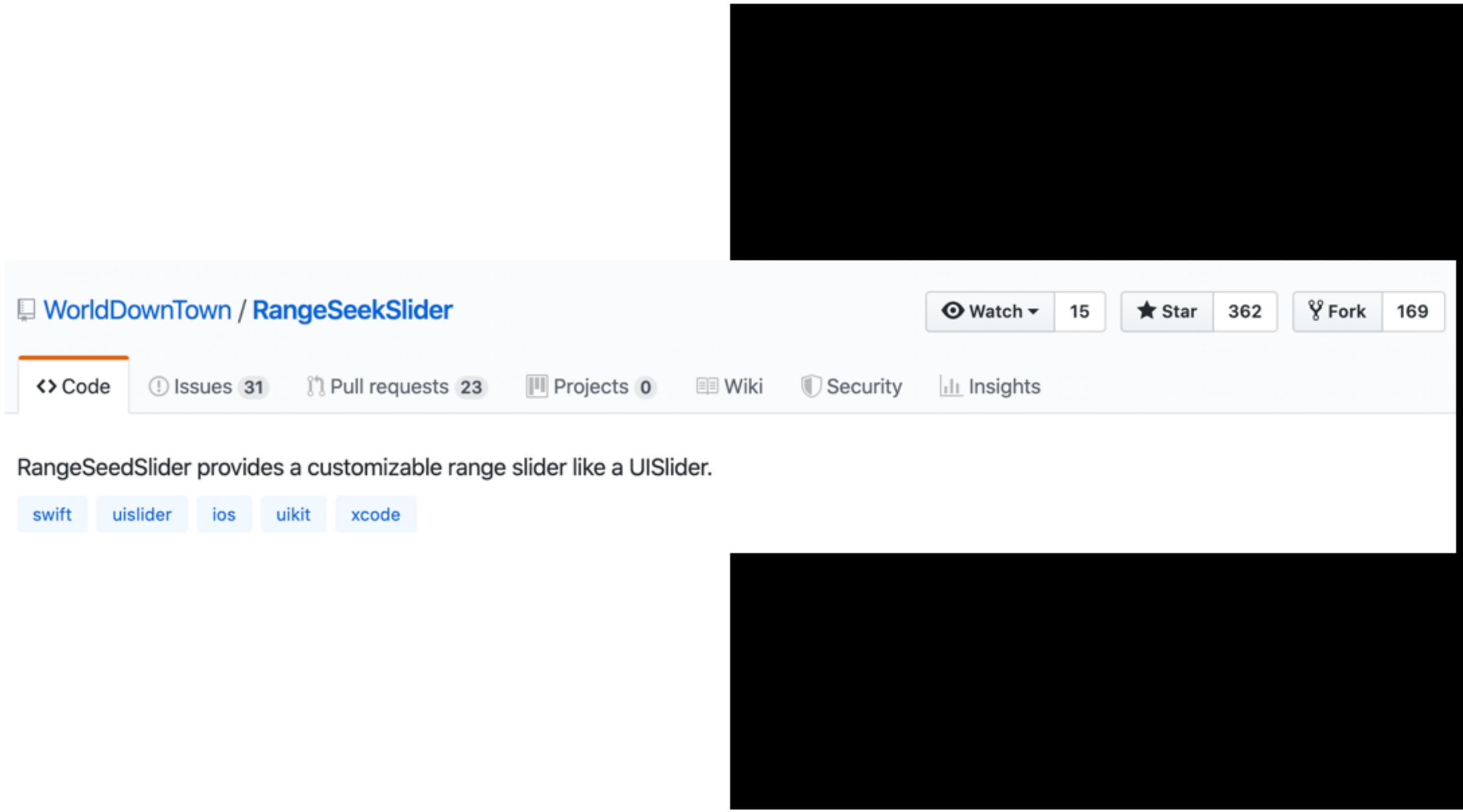
Goal

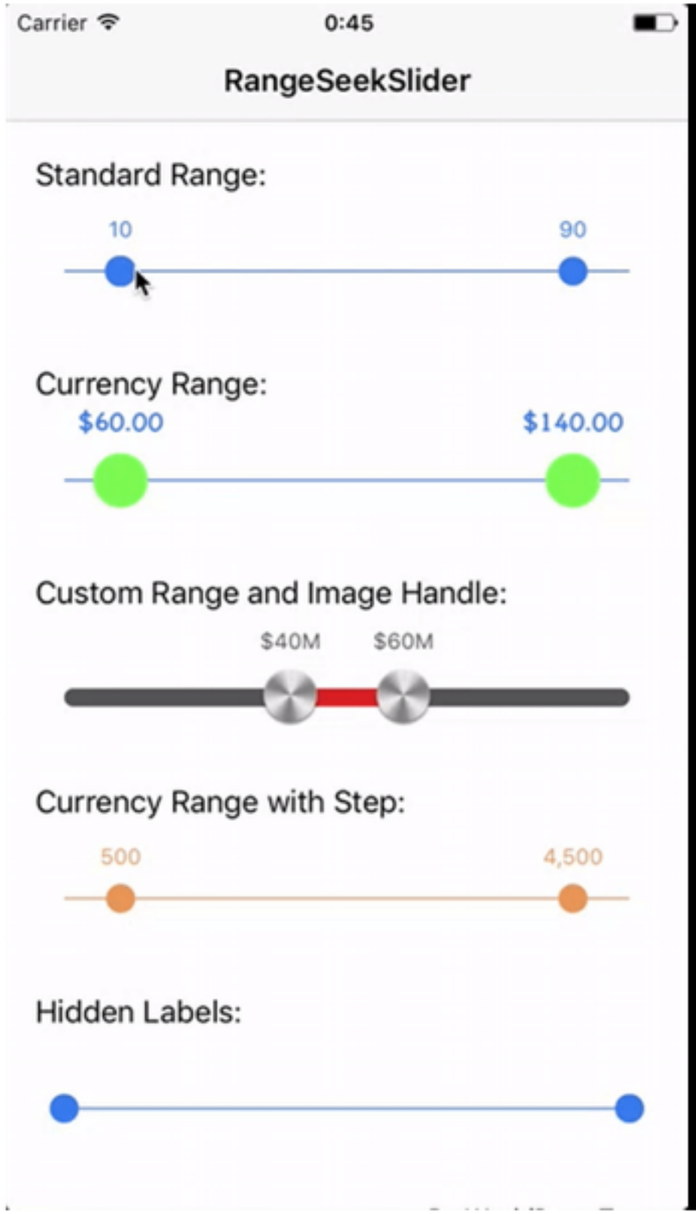
[redacted]

Rewrite a range slider from UIKit [redacted]

Compass RangeS

- Based on an existing open sou
- Heavily modified to remove fe
  follow our coding style


WorldDownTown / RangeSeekSlider
Watch 15
Star 362
Fork 169
Code
Issues 31
Pull requests 23
Projects 0
Wiki
Security
Insights
RangeSeedSlider provides a customizable range slider like a UISlider.
swift
uislider
ios
uikit
xcode


Carrier
0:45
RangeSeekSlider
Standard Range:
10
90
Currency Range:
$60.00
$140.00
Custom Range and Image Handle:
$40M
$60M
Currency Range with Step:
500
4,500
Hidden Labels:

RangeSeekSlider

RangeSeekSliderViewModel

Sha
Gest

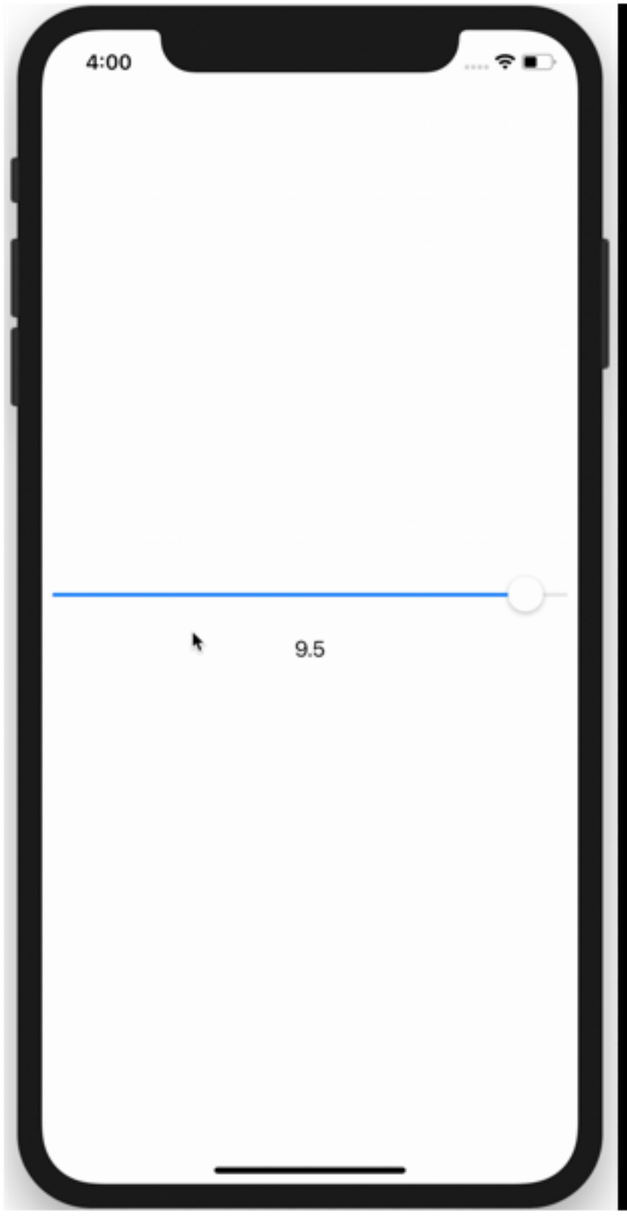
Start S
And G
Little Det

Nat
uis

Goals

[redacted]

- ☞ Correct padding on left and rig
- ☞ Formatted text that displays th


4:00
9.5

Dependenc
With

This Will Not Co

```
struct TestSlider : View {
    @State var selectedValue: CGFloat = 0.0
    @State var numberFormatter: NumberFormatter

    init(numberFormatter: NumberFormatter = NumberFormatter
        self.numberFormatter = numberFormat
    }

    var body: some View {
        // removed for brevity
    }
}
```

“@State variab
should not be in
data you pass do
initializer...”

—*Joe Groff, Senior Swift Compile*

“...since the mod
maintained outs
there is no guara
value will really

—Joe Groff, Senior Swift Compile

`@State` Is

Intend

Small-Sca

```
struct TestSlider: View {
    @State private var selectedValue: CGFloat   0.0
    private let numberFormatter: NumberFormatter
    private let minValue: CGFloat
    private let maxValue: CGFloat
    private let step: CGFloat

    init(rangeType: RangeType) {
        self.numberFormatter = rangeType.num
        self.minValue = rangeType.minValue
        self.maxValue = rangeType.maxValue
        self.step = rangeType.step
    }

    var body: some View {
        VStack {
            Slider(value: $selectedValue, fr
                .padding()
            Text(numberFormatter.string(from                    as NSNumber) ?? ""
        }
    }
}
```


1:24
$0
0

Why Not Use @B[redacted]

- ☞ numberFormatter, min, max a[redacted]
  once you initialize the slider
- ☞ Only one view needs access to[redacted]

CGFl

I thought we we
from tradition...

And leaving behi
that didn't serve

Nick Lockwood
@nicklockwood

I'm a bit disappointed that SwiftUI is still using CGPoint, CGRect, etc. The ergonomics of using CGFloat with Swift are horrible, can't we just have new geometric primitives based on Doubles?

11:29 AM · Jul 9, 2019 · Twitter Web App

15 Retweets **157** Likes

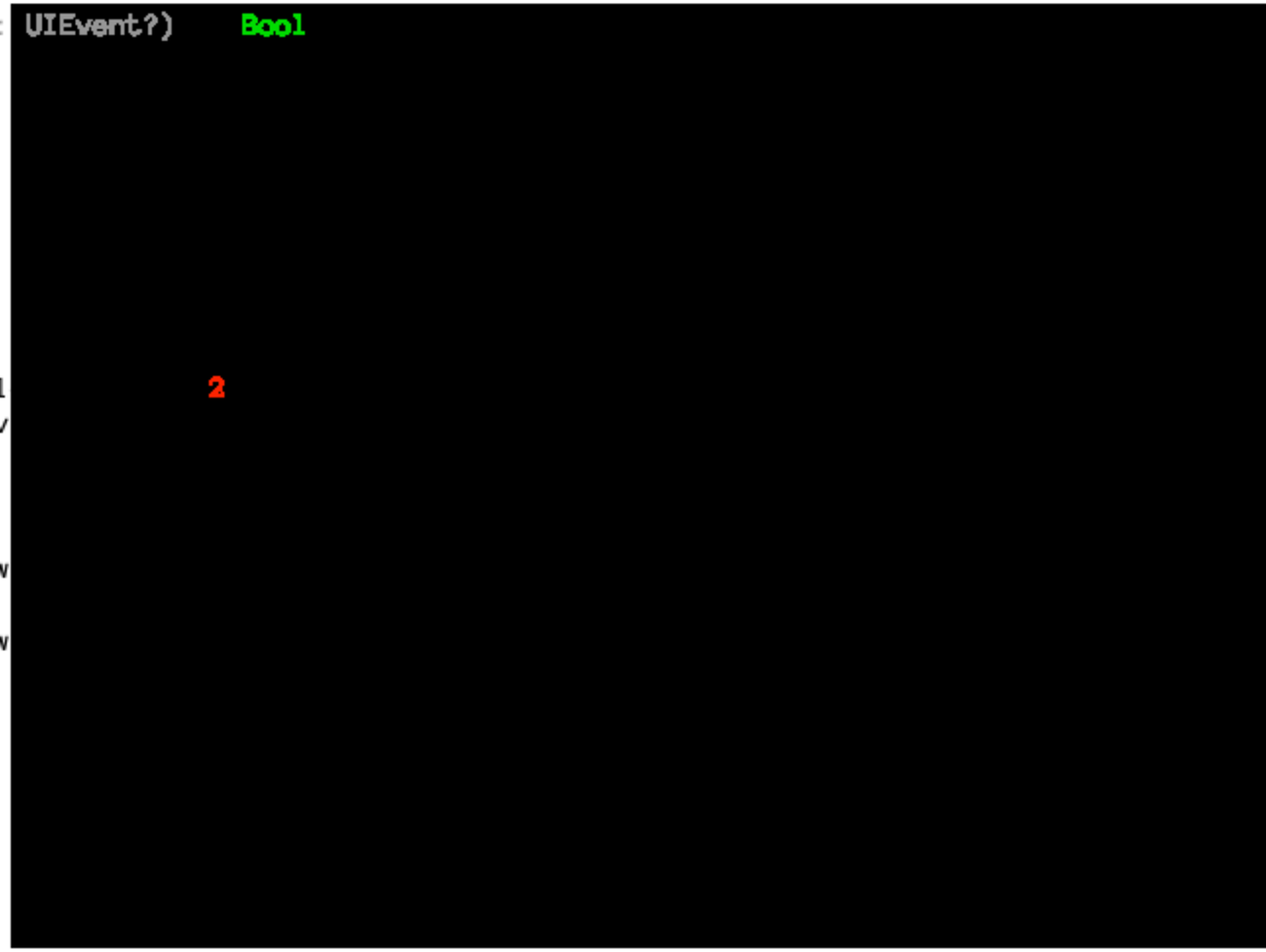
Joe Groff's Respo

- ☞ CGFloat is still single-precision
  for any retina display, or a non
  1024×768.
- ☞ It's too late for shipping ABIs.
- ☞ "change CGFloat.h so that CGF
  defined platforms" might be a

Sim
Gest

```
override func beginTracking(_ touch: UITouch, with event: UIEvent?)    Bool
    super.beginTracking(touch, with:
    // calculations

    guard isTouchingLeftHandle || isT                    else
        return false
    }
    // assign handleTracking to .left or .right

    return true
}
```

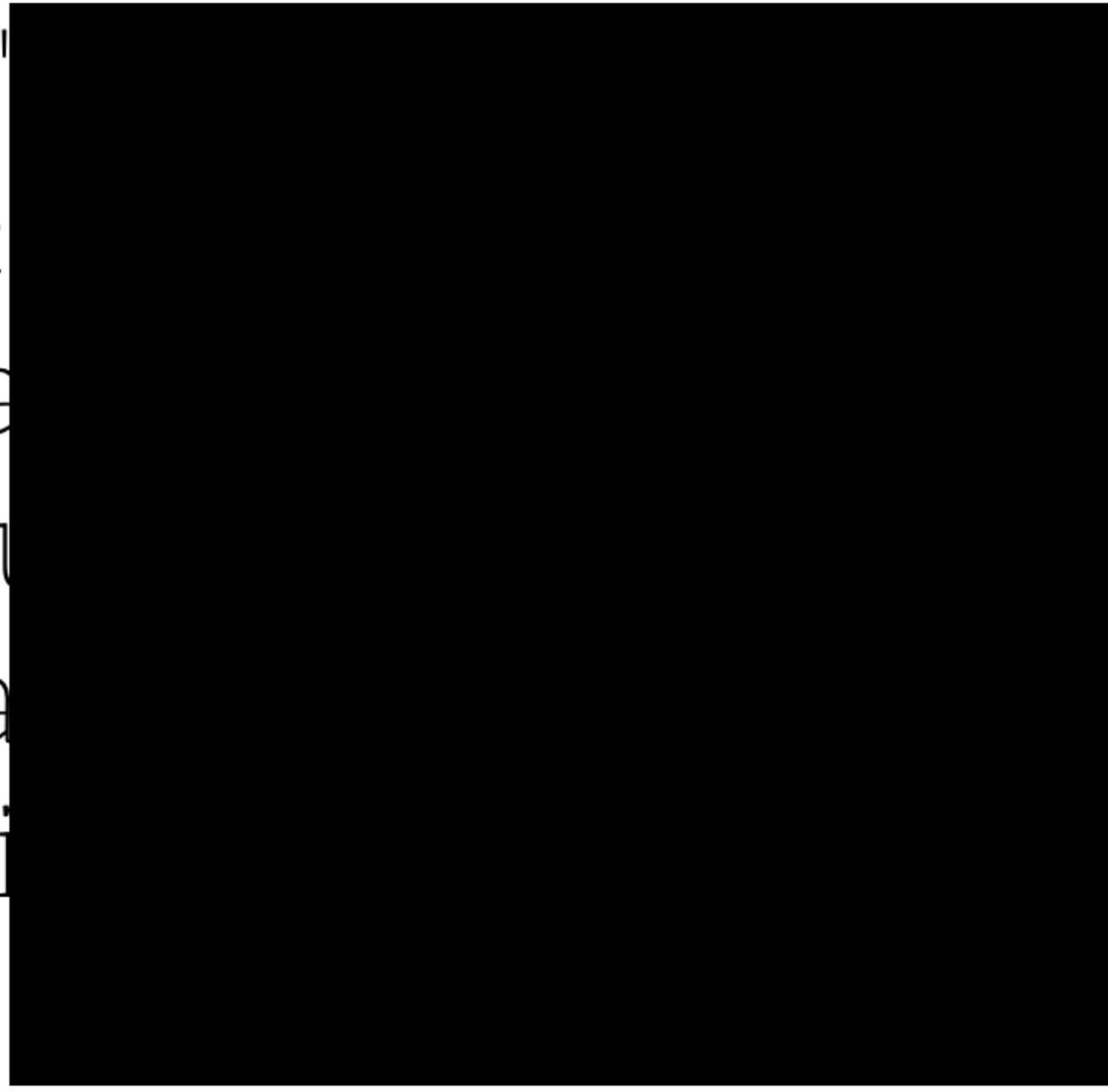
```swift
override func continueTracking(_ touch: UITouch, with event: UIEvent?) -> Bool {
    super.continueTracking(touch, with: event)
    guard handleTracking != .none else {
        return false
    }

    let location = touch.location(in: self)

    var percentage: CGFloat = 0
    var selectedValue: CGFloat = 0
    percentage = (location.x - sliderLine.frame.minX - handl
    selectedValue = max(percentage * (viewModel.maxValue - v

    switch handleTracking {
    case .left:
        viewModel.selectedMinValue = min(selectedValue, view
    case .right:
        viewModel.selectedMaxValue = max(selectedValue, view
    case .none:
        break
    }

    refresh()

    return true
}
```



```
override func endTracking(_ touch: UITouch?, with event: UIEvent?)
    super.endTracking(touch, wit
    handleTracking = .none
    initialTouchPoint = CGPoint.
    strokePhase = .notStarted
    trackedTouch = nil
}
```

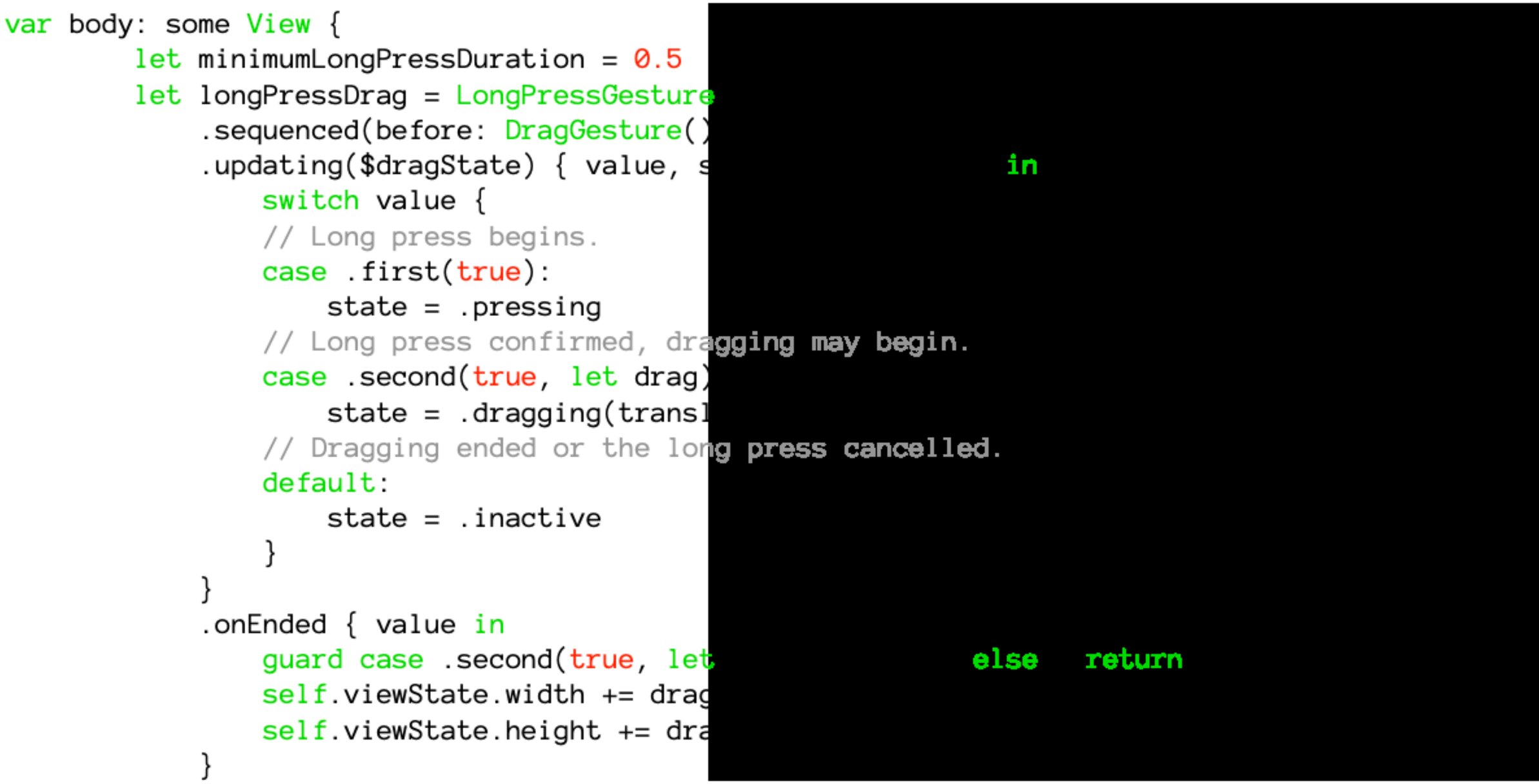
“SwiftUI doesn'
updating callbac
ends or cancels a
Instead, the gestu
property automa
state back to its i

—SwiftUI Documentation



“SwiftUI only in
`onEnded(_:)` ca
gesture succeeds

—SwiftUI Documentation


var body: some View {
let minimumLongPressDuration = 0.5
let longPressDrag = LongPressGesture
.sequenced(before: DragGesture()
.updating($dragState) { value, s
switch value {
// Long press begins.
case .first(true):
state = .pressing
// Long press confirmed, dragging may begin.
case .second(true, let drag)
state = .dragging(transl
// Dragging ended or the long press cancelled.
default:
state = .inactive
}
}
.onEnded { value in
guard case .second(true, let
self.viewState.width += drag
self.viewState.height += dra
}
in
else return

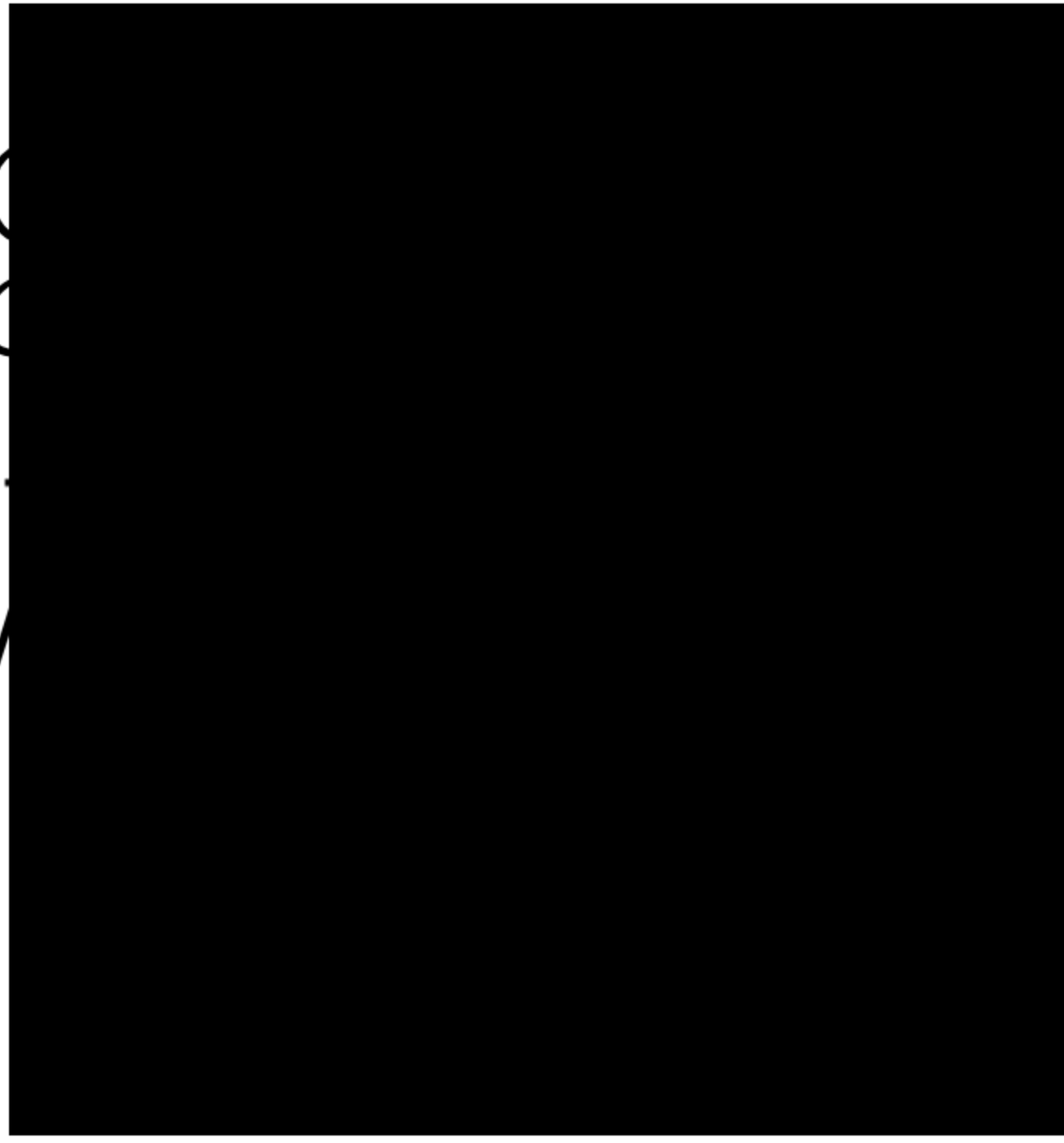
Matt Gallagher
Cocoa with Love

First impressions of SwiftUI

June 8, 2019 by Matt Gallagher
Tags: app-design, cocoa

A little over a month ago, I released CwlViews and then followed up with an article suggesting that Apple might be about to release their own declarative views library. At WWDC this week, they did just that, releasing SwiftUI.

Moving
"How
Do X



To

"What

New

Where is `UICont`[redacted]

Where are `begin`[redacted]

`continueTracki`[redacted]

`endTracking`? I [redacted]

them in my new[redacted]

There's
more
UIContr
Use a V

We now call **upd**

onChanged and **c**

on **Gesture** inste

overriding any fu

class.

RangeSe
Without

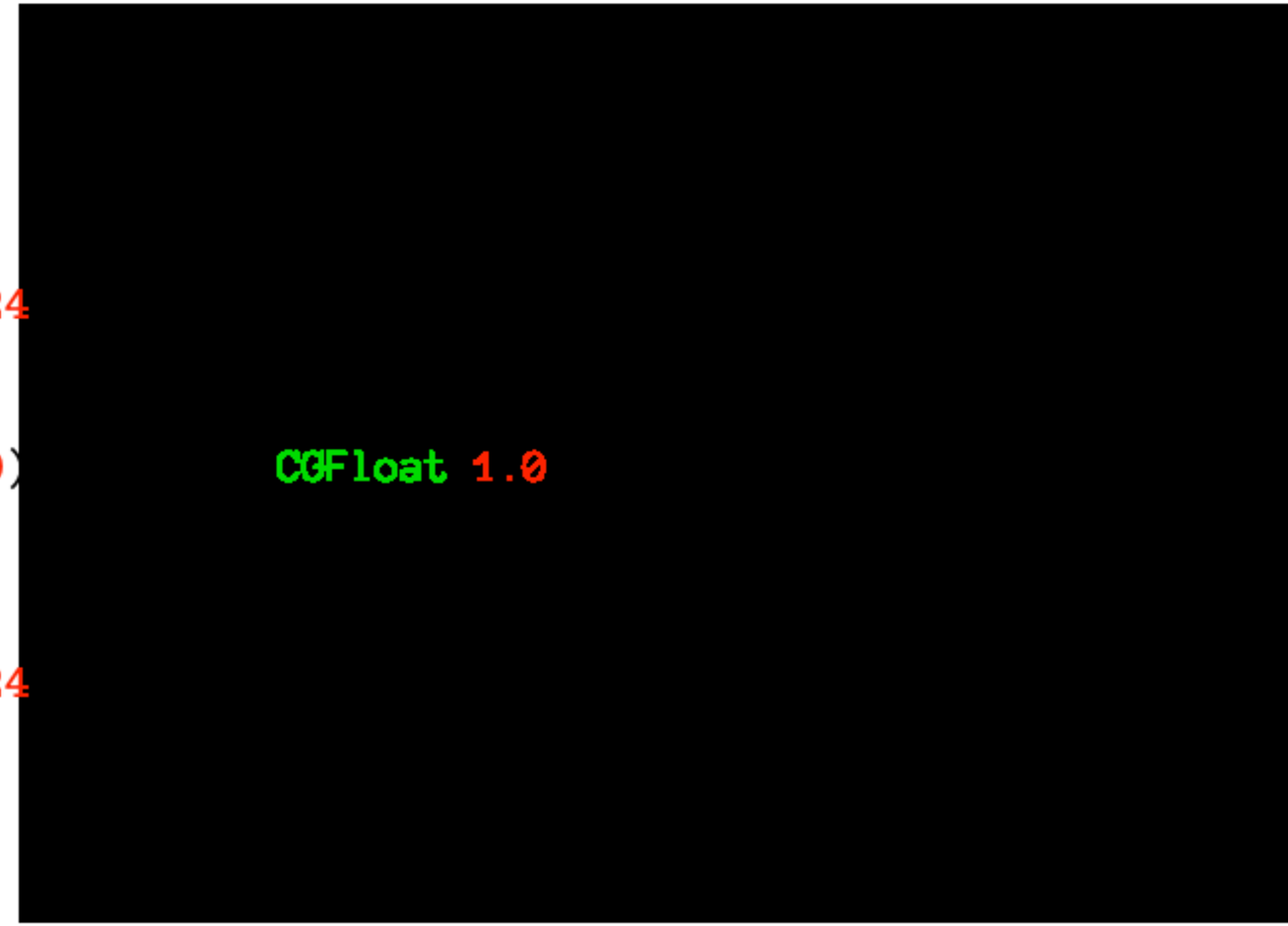
Building Custom Views in SwiftUI

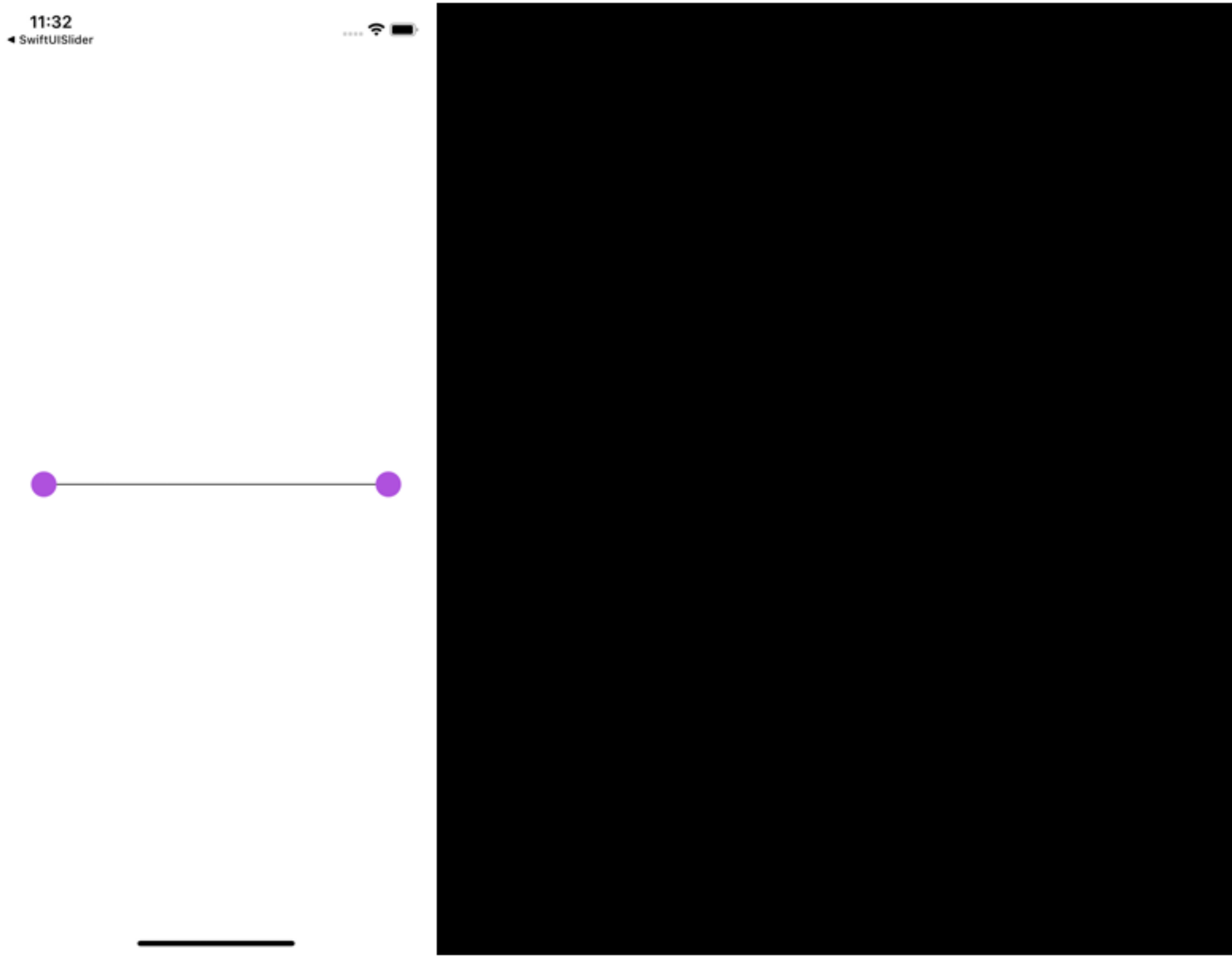
Graphic effects and layout

Dave Abrahams
John Harper

- ☞ We no longer have a `UIKit` an[redacted]
- ☞ Everything in SwiftUI is a `Vie`

```swift
struct ContentView : View {
    var body: some View {
        return HStack(spacing: 0) {
            Circle()
                .fill(Color.purple)
                .frame(width: 24, height: 24)
                .zIndex(1)
            Rectangle()
                .frame(width: CGFloat(300.0))
                .zIndex(0)
            Circle()
                .fill(Color.purple)
                .frame(width: 24, height: 24)
                .zIndex(1)
        }
    }
}
```




11:32
SwiftUISlider

RangeSe
With G

```swift
import SwiftUI

struct PriceContentView : View {
    @State private var selectedMinValu
    @State private var selectedMaxValu
    @State private var leftHandleViewS
    @State private var rightHandleView
    private let numberFormatter = Range
    private let minValue = RangeType.p
    private let maxValue = RangeType.p
    private let step = RangeType.price
    private let lineWidth: CGFloat = 300.0
    private let handleDiameter: Length
```

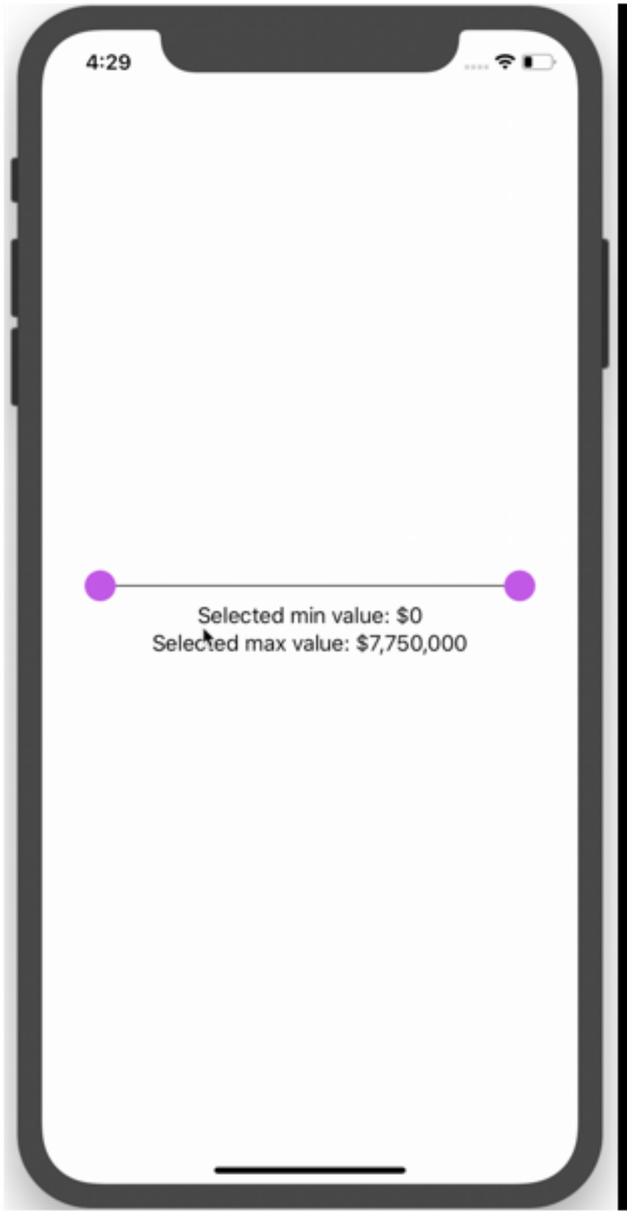
```
    CGFloat   RangeType
    CGFloat   RangeType
        CGSize
         CGSize
Type
                          24
```

```
var body: some View {
    let leftHandleDragGesture = DragGesture(minimumDista
        .onChanged { value in
            guard value.location.x >= 0, value.location.
                return
            }
            self.leftHandleViewState.width = value.locat
            let percentage = self.leftHandleViewState.wi
            self.selectedMinValue = max(percentage * (se
            self.selectedMinValue = CGFloat(roundf(Float
    }
```

```
    1

     self            self                else

    self            self
lf          self            self            self
self                self          self
```

```
let rightHandleDragGesture = DragGesture(minimumDist[redacted]
    .onChanged { value in
        guard value.location.x <= 0, value.location.[redacted]
            return
        }
        self.rightHandleViewState.width = value.loca[redacted]
        let percentage = 1 - abs(self.rightHandleVie[redacted]
        self.selectedMaxValue = max(percentage * (se[redacted]
        self.selectedMaxValue = CGFloat(roundf(Float[redacted]
}
```

```
        return
            VStack(spacing: 0) {
                HStack(spacing: 0) {
                    Circle()
                        .fill(Color.purple)
                        .frame(width: handleDiameter, he
                        .offset(x: leftHandleViewState.w          0
                        .gesture(leftHandleDragGesture)
                        .zIndex(1)
                    Rectangle()
                        .frame(width: lineWidth, height:  CGFloat 1.0
                        .zIndex(0)
                    Circle()
                        .fill(Color.purple)
                        .frame(width: handleDiameter, he
                        .offset(x: rightHandleViewState.          0
                        .gesture(rightHandleDragGesture)
                        .zIndex(1)
                }
                Text("Selected min value: \(numberFormatter.string(from: selectedMinValue as NSNumber) ?? "")"
                Text("Selected max value: \(numberFormatter.string(from: selectedMaxValue as NSNumber) ?? "")"
        }
    }
}
```


4:29
Selected min value: $0
Selected max value: $7,750,000

```
private func xPositionAlongLine(for value: CGFloat)    CGFloat
    let percentage = percentage          for
    let maxMinDif = sliderLine.
    let offset = percentage * m
    return sliderLine.frame.min
}

private func percentageAlongLine(for value: CGFloat)    CGFloat
    guard viewModel.minValue <                    else
        return 0
    }
    let maxMinDif = viewModel.m
    let valueSubtracted = value
    return valueSubtracted / ma
}
```

Lines Of Code	UI
Determine which handle was being dragged	2
Set the x position of the handle during the drag gesture	1

Continuously Re

- ☞ Change `minimumDistance` of g
- ☞ Guard against touch location t
  on the line
- ☞ Formatted text that displays th

Your Kn

Port


Dow Jones
28000
26000
24000
22000
20000
Punkte
J J A S O N D J F M A M
vom Vortag, Veränderung in Prozent zum Vortag
2874,4 -0,07%
16.431,0 -0,08%
2882,3 -2,48%
9659,1 -0,02%
3140,1 -0,21%
2867,2 -0,51%
-0,19%
USA
China

SwiftUI S
Open S

“...the nature of
declarative syste
API documentat
fill-in all the deta

—***"First impressions of SwiftUI"***

“...there is too m
that does not ma
the interface.”

—"First impressions of SwiftUI"

www.nerd

@nerd