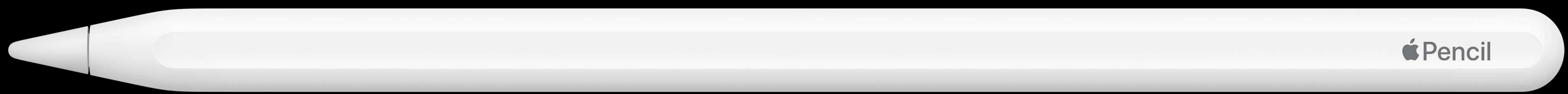


#WWDC19

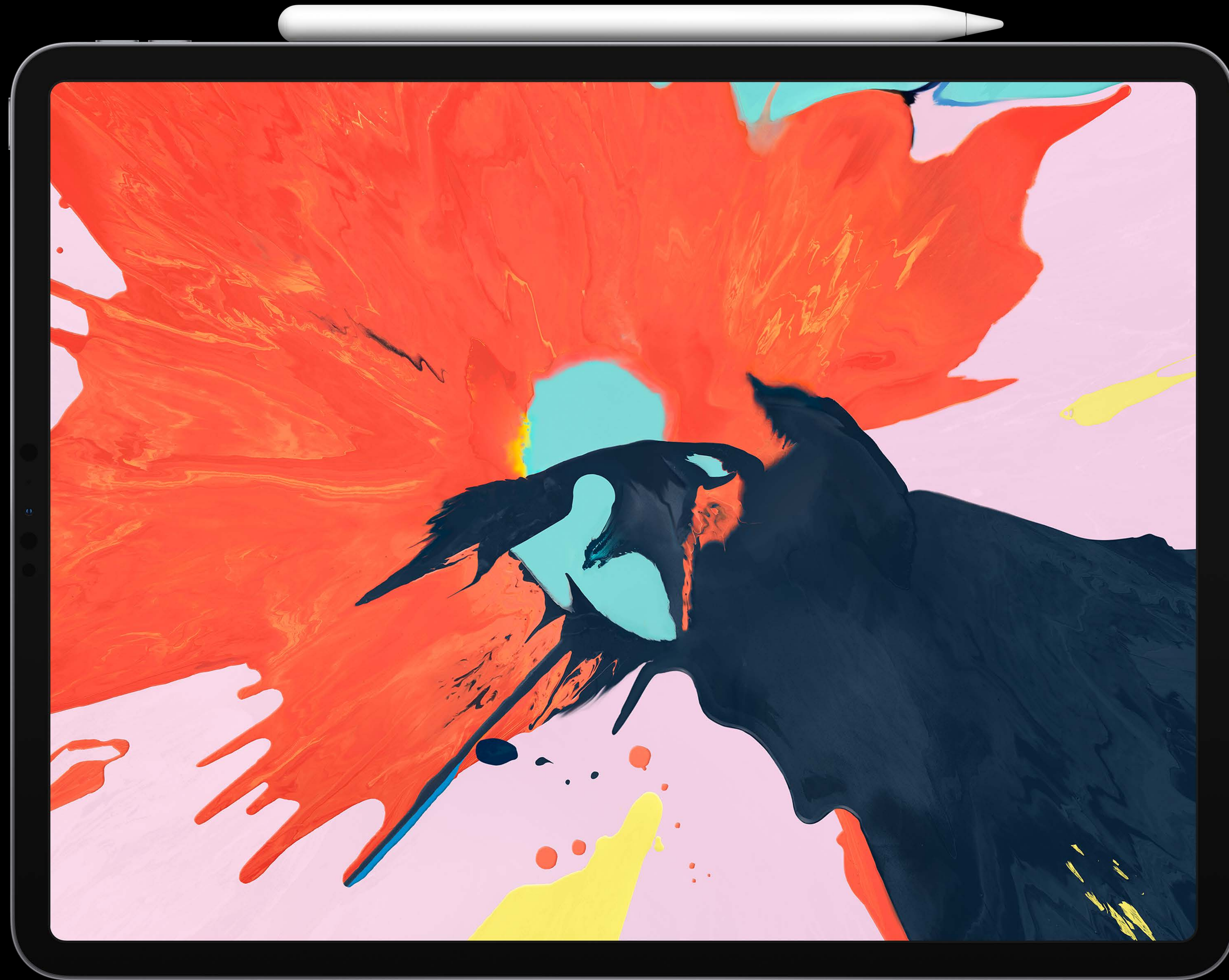
# Introducing PencilKit

Will Thimbleby, Pencil and Paper Engineer  
Jenny Chen, Pencil and Paper Engineer



Pencil is the most unique aspect of iPad

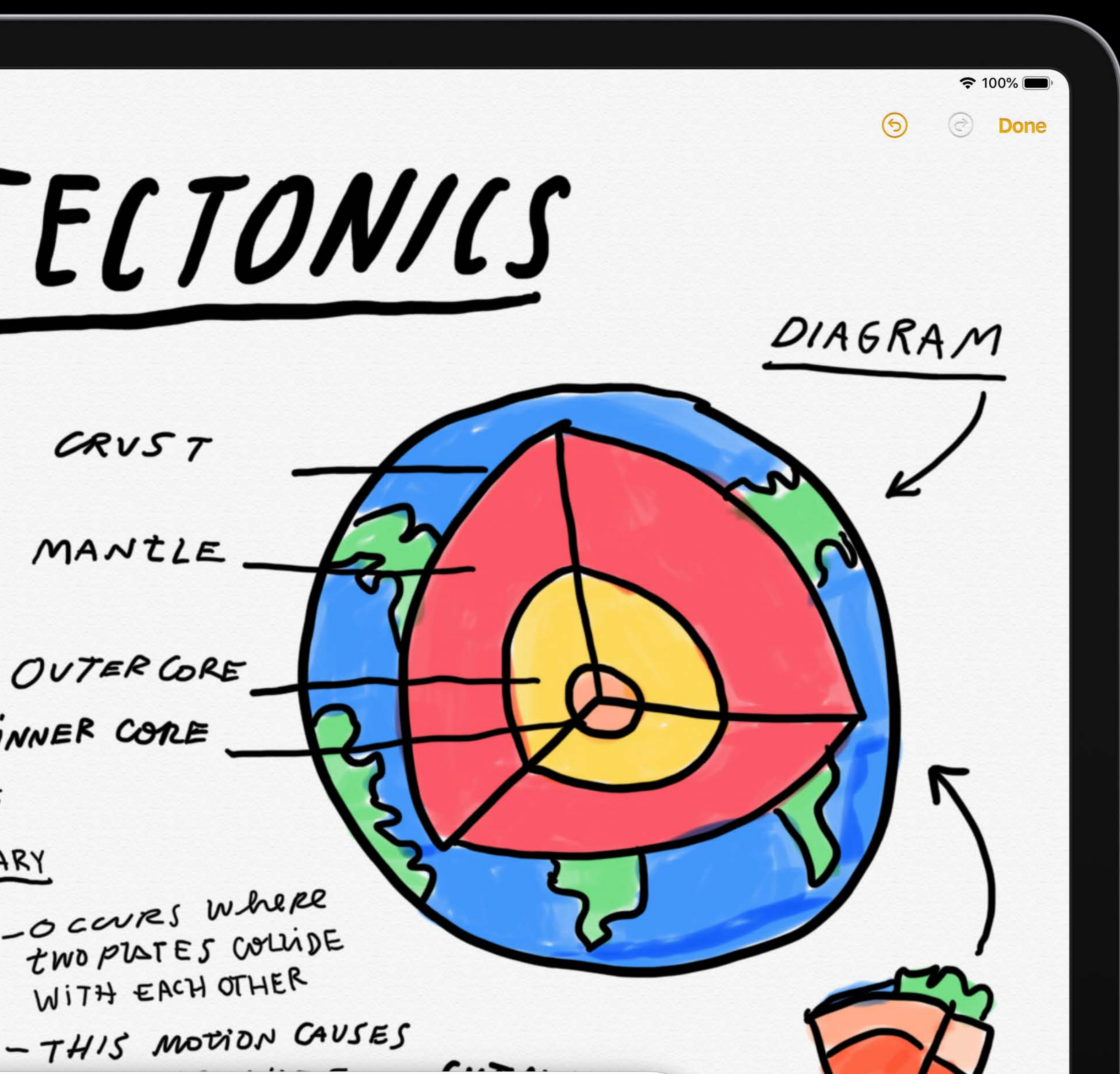
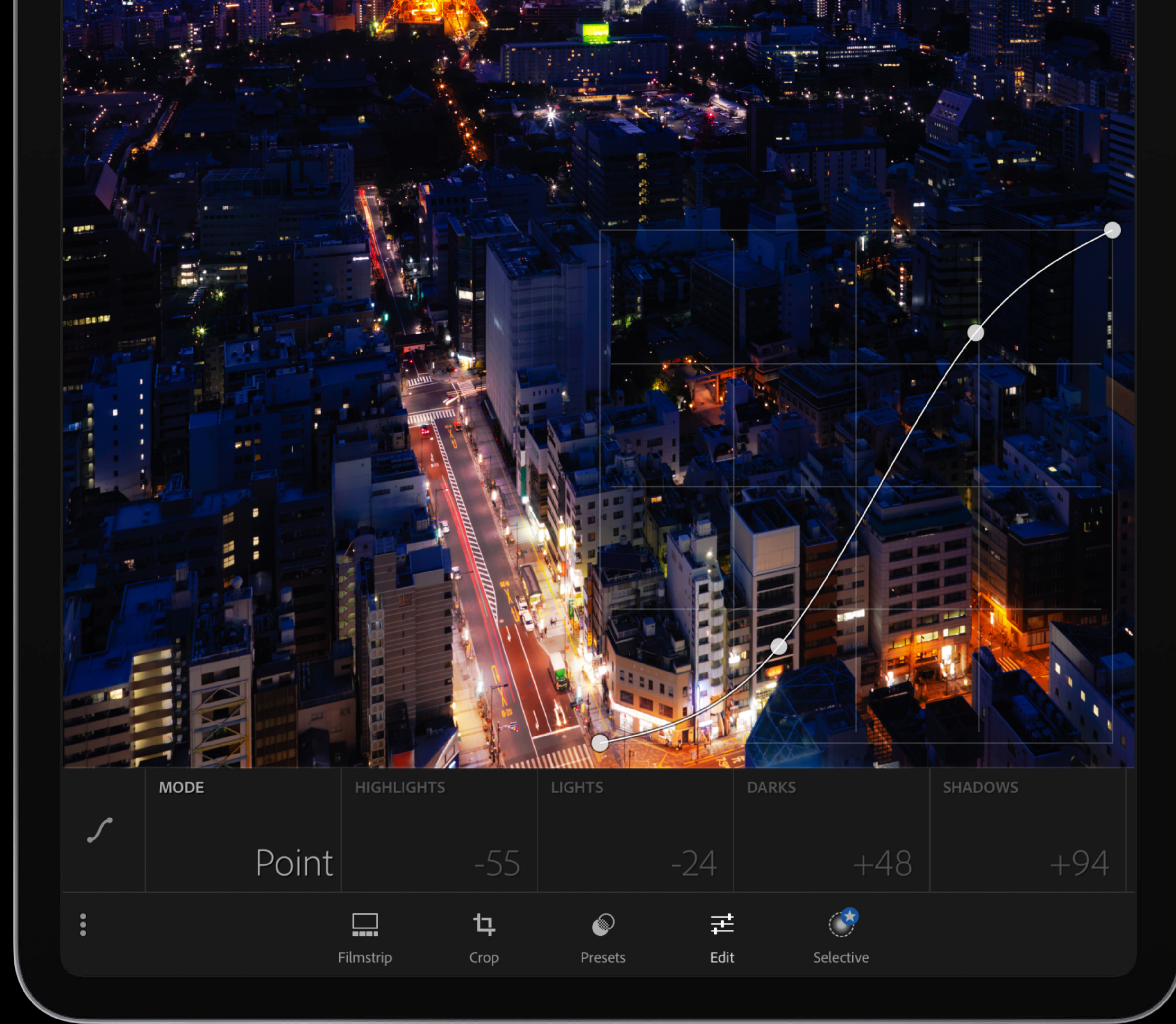
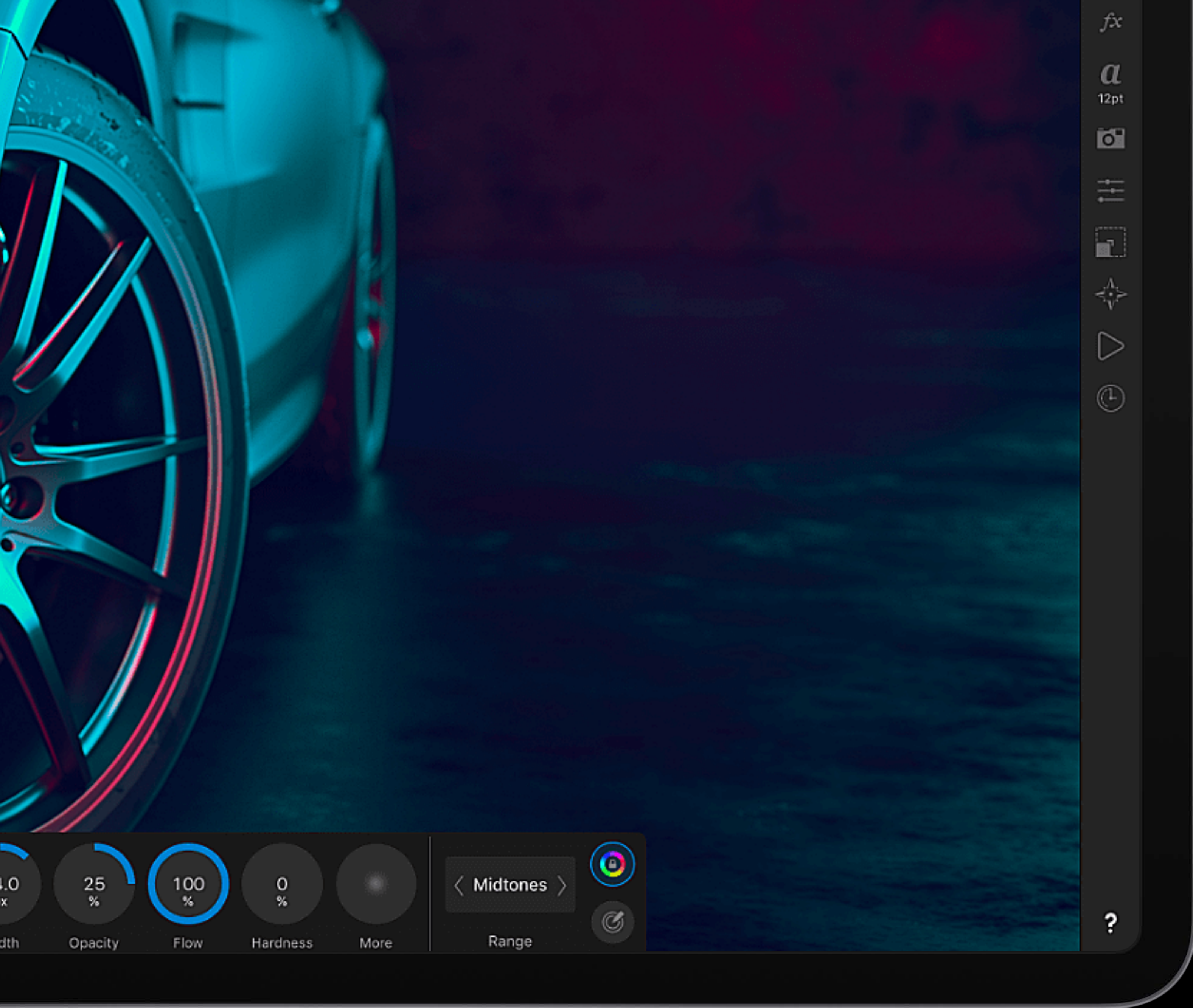














# New in iOS 13

Lower latency

New tool palette

PencilKit

Markup everywhere



Great pencil experiences

PencilKit

Markup everywhere



Great pencil experiences

Great pencil experiences  
fully use Pencil



# Fully Using Pencil

Precision enables new user experiences

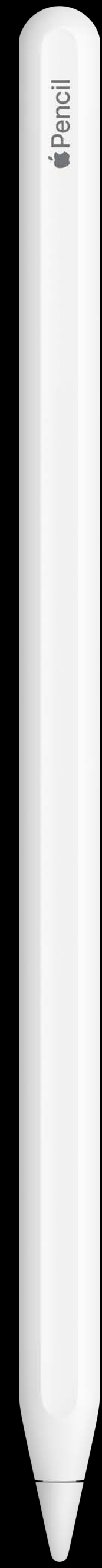
Force, azimuth, and altitude allow expressive marks

Pencil taps switch modes

# Three Types of Pencil



1st Gen



2nd Gen



Crayon



# Three Types of Pencil

1st Gen

2nd Gen

Crayon

Precision



Azimuth/Altitude



Force



Tap Gesture





# Building Great Pencil Experiences

Rich touch data from UIKit

Fast rendering in Metal

Async performance with GCD

---

Leveraging Touch Input on iOS

WWDC 2016

---

Metal Performance Optimization Techniques

WWDC 2015

---

Modernizing Grand Central Dispatch Usage

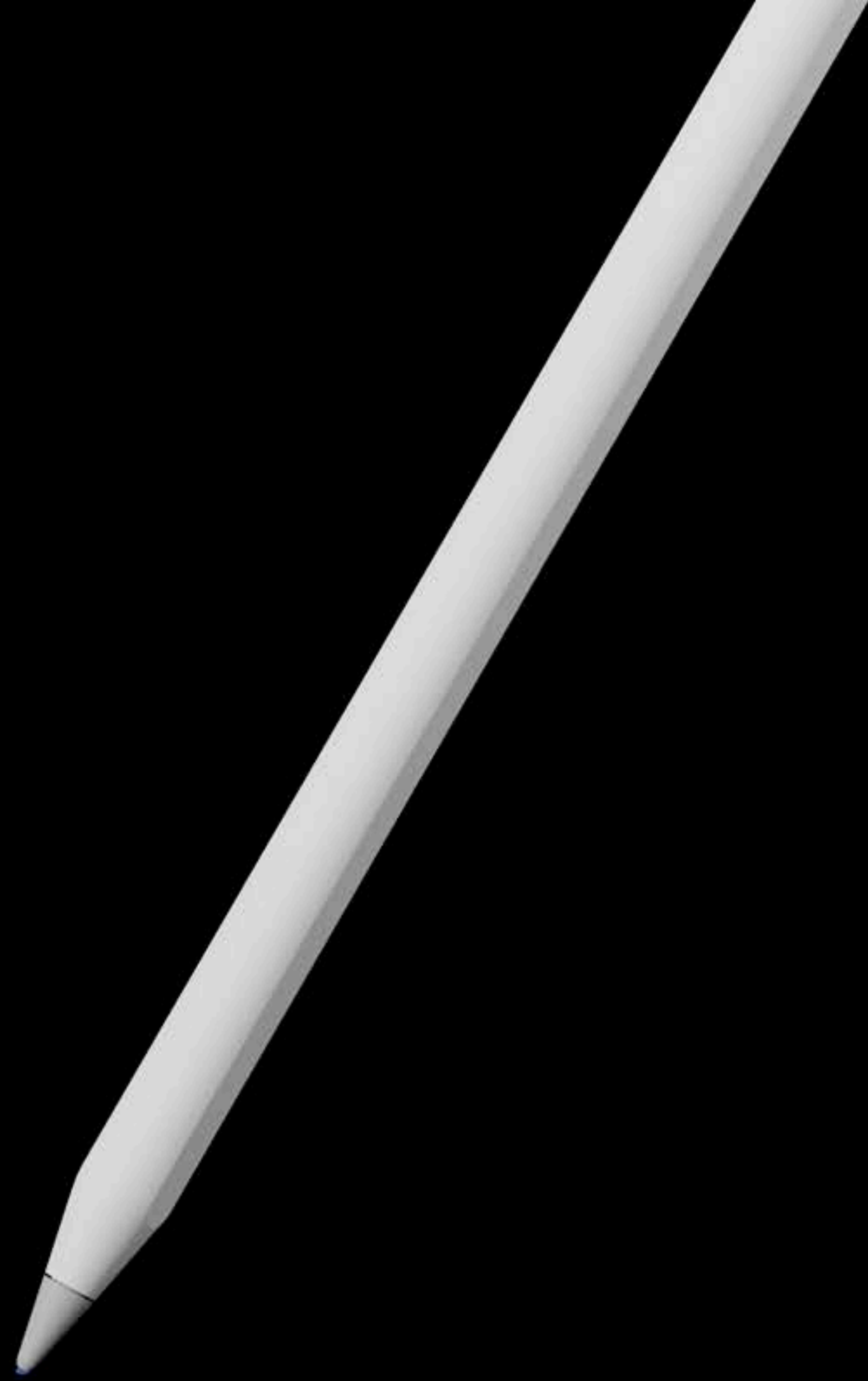
WWDC 2017

---

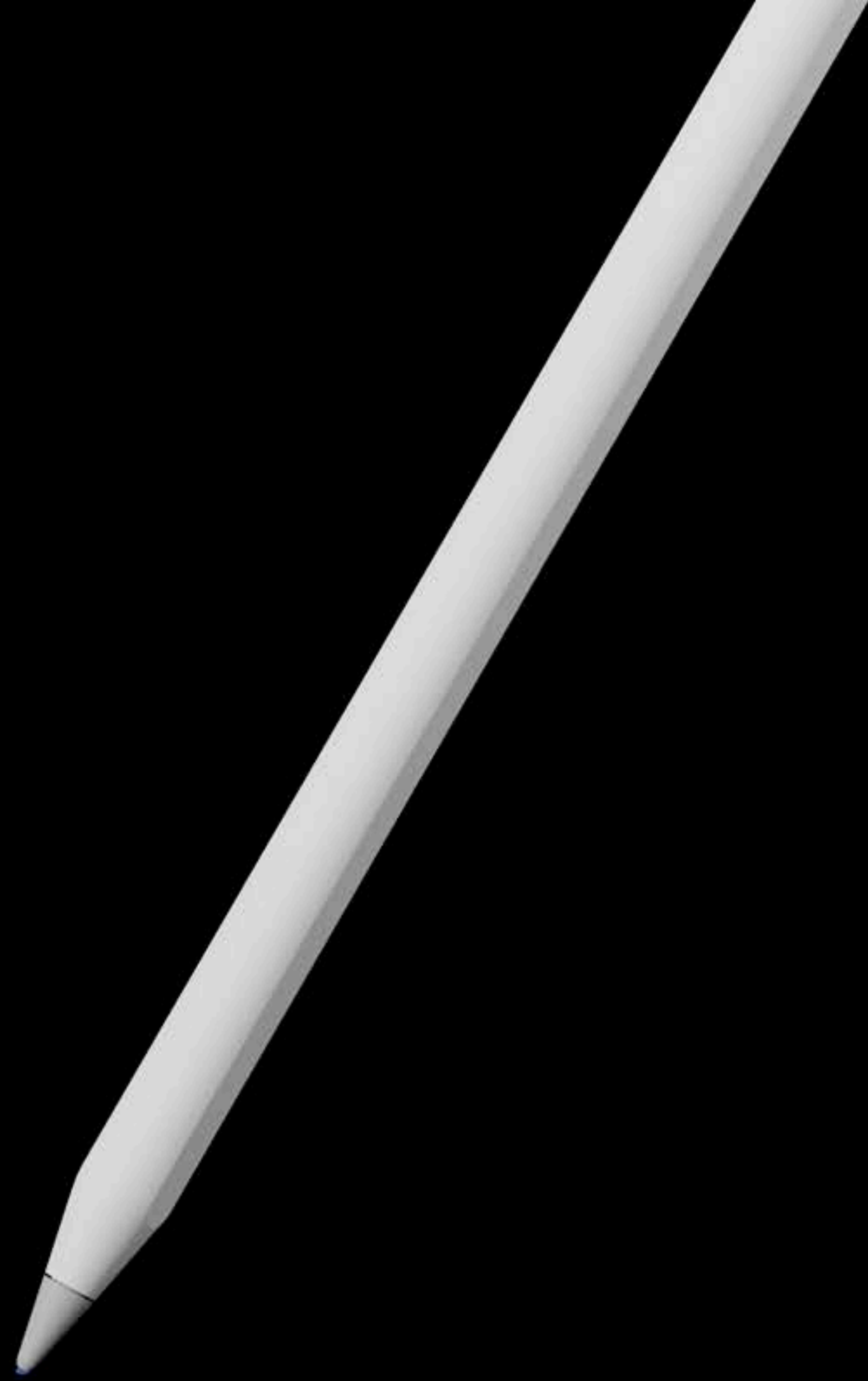


# How Pencil Works











# The Impact of How Pencil Works

Azimuth / altitude may be estimated

Azimuth is imprecise when pencil is perpendicular

Force data is delayed

# Back-fill Azimuth and Altitude



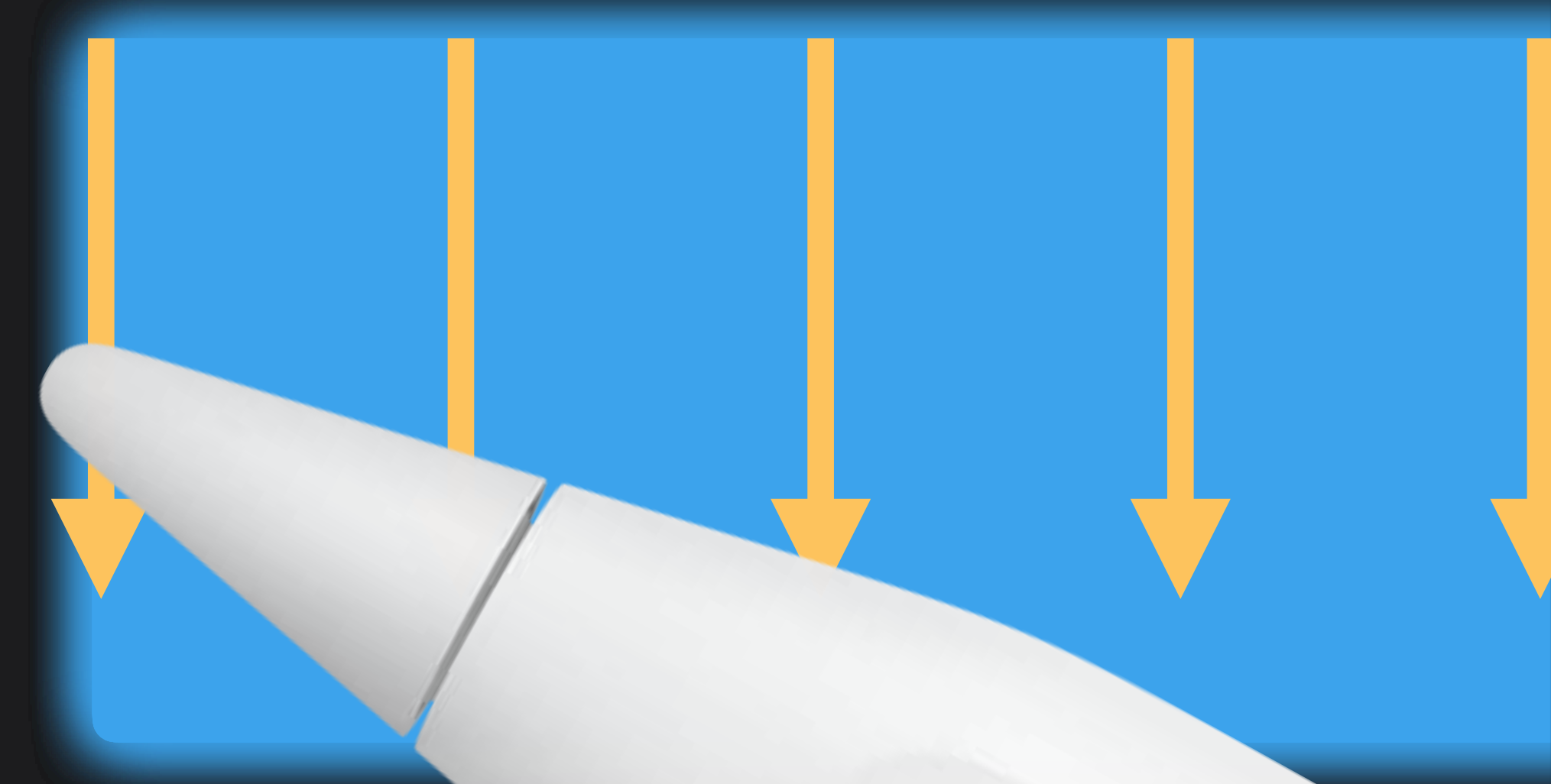


# Back-fill Azimuth and Altitude



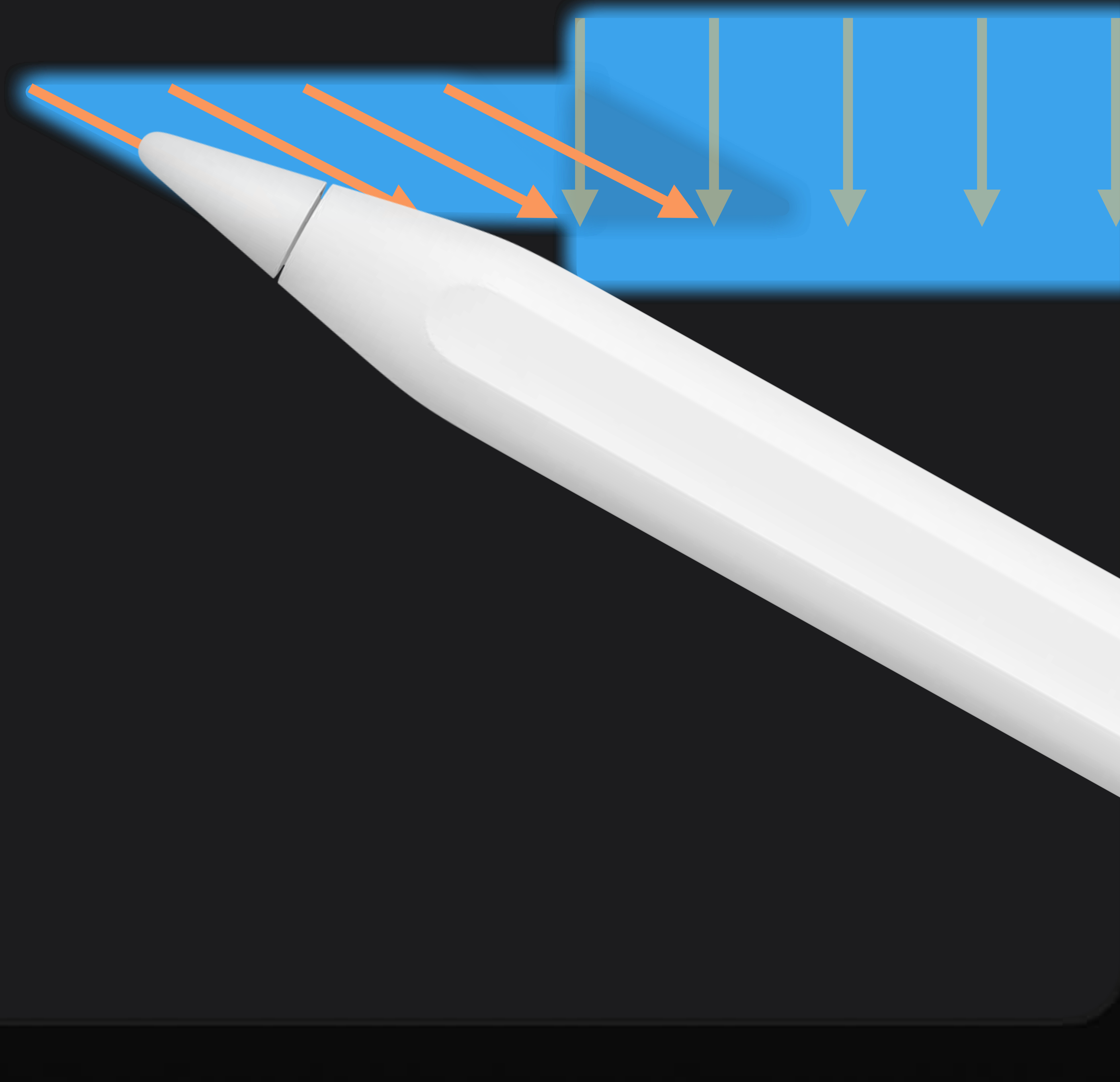


# Back-fill Azimuth and Altitude





# Back-fill Azimuth and Altitude





# Back-fill Azimuth and Altitude

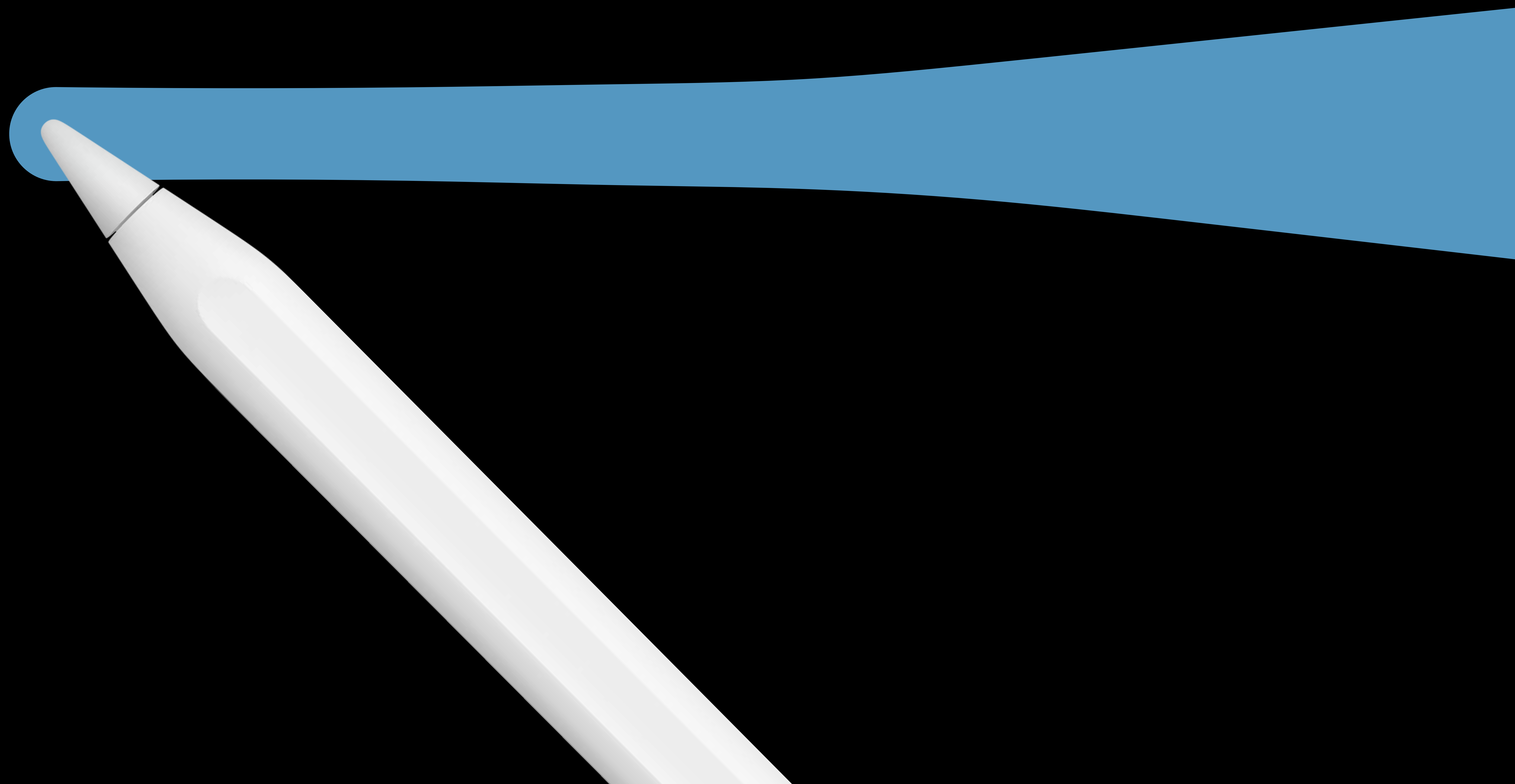




# Handle Force Updates After Touches Ended

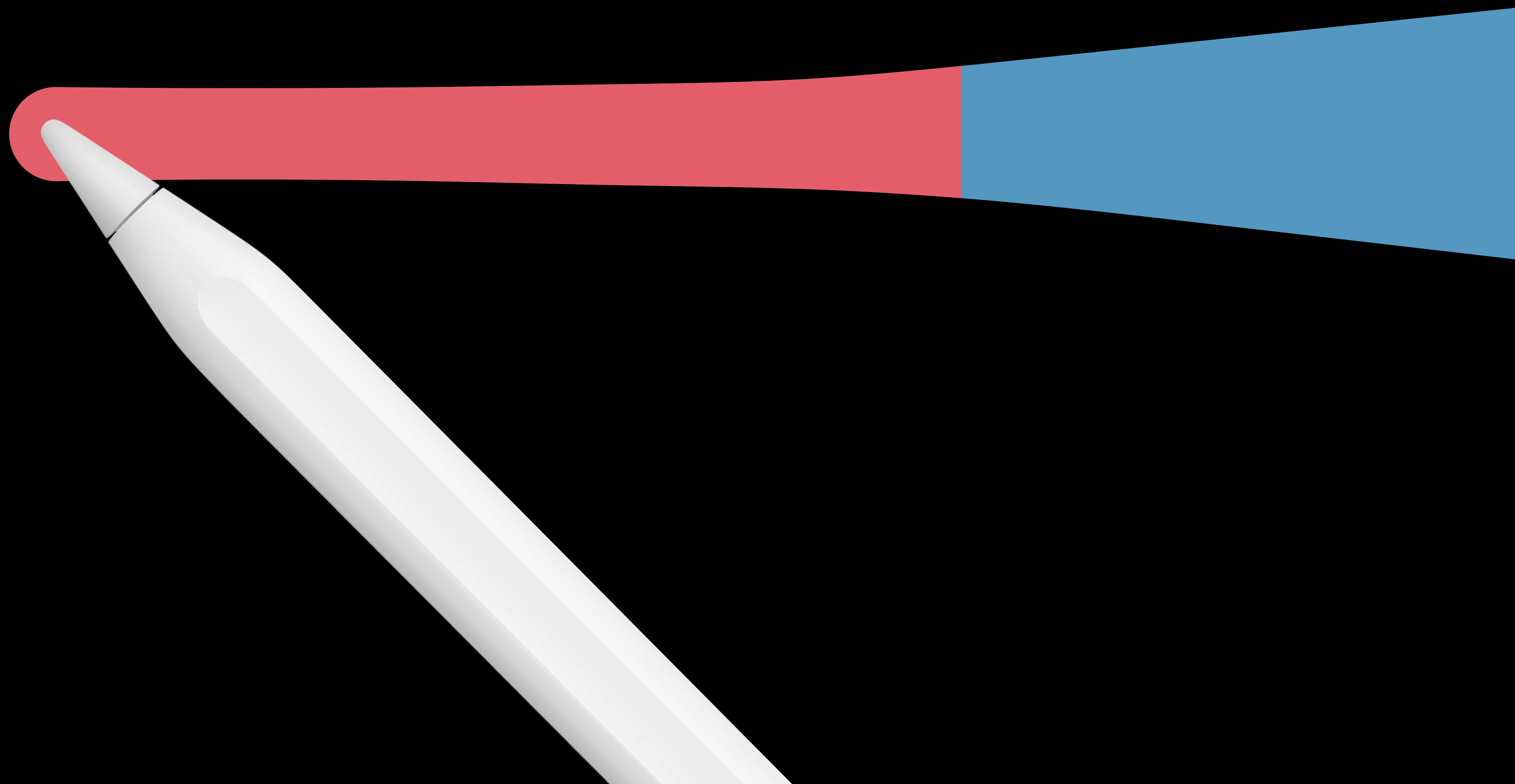


# Handle Force Updates After Touches Ended



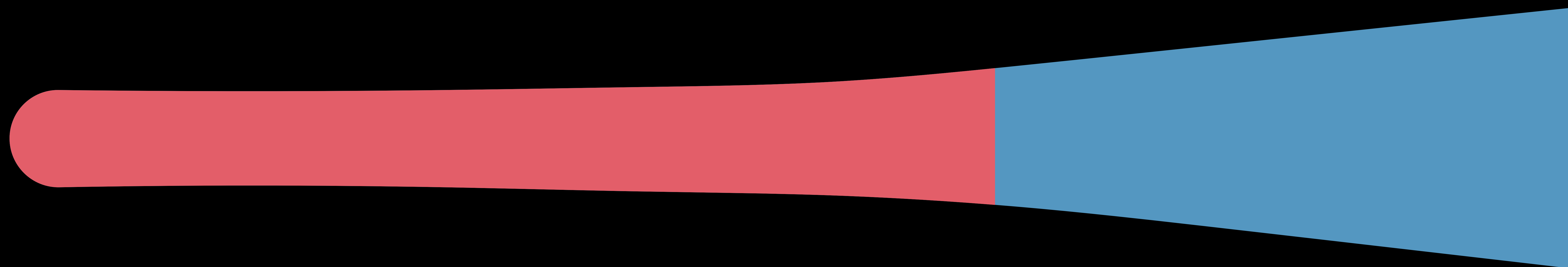


# Handle Force Updates After Touches Ended



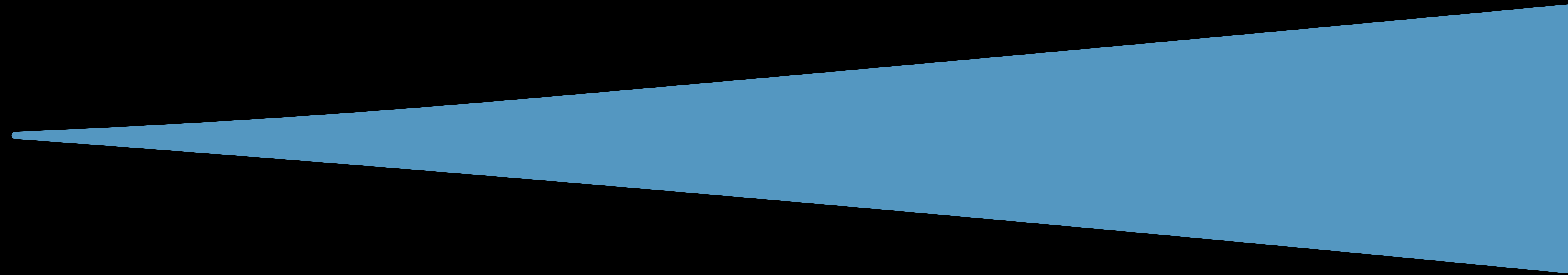


# Handle Force Updates After Touches Ended



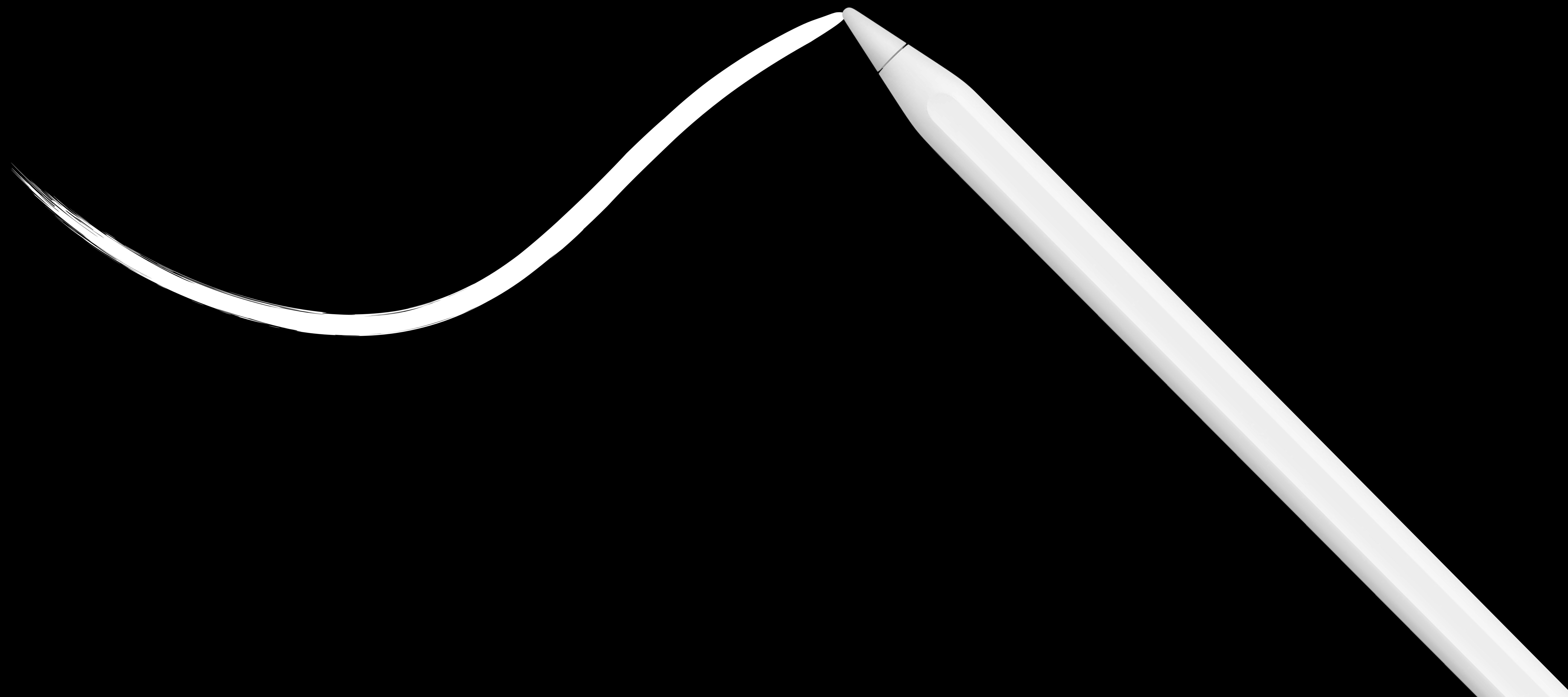


# Handle Force Updates After Touches Ended



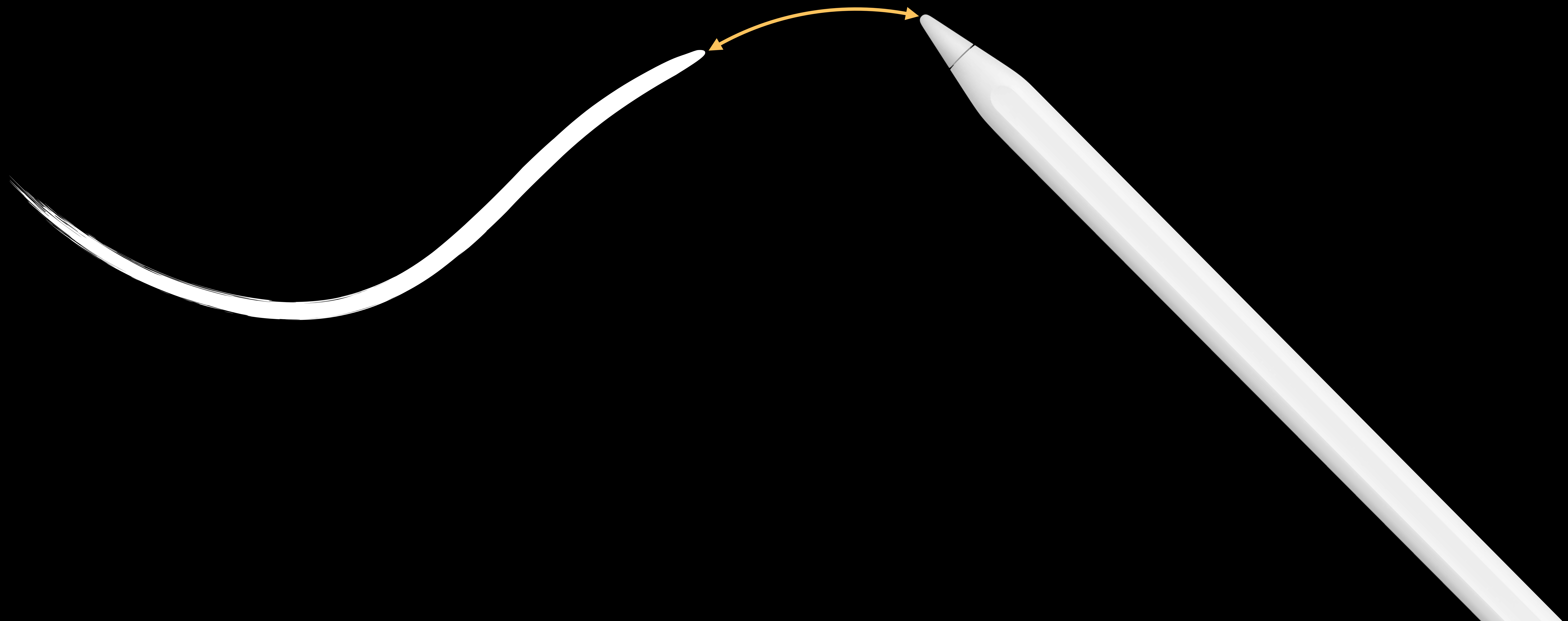


# Latency Is Critical





# Latency Is Critical







ISO: 17000 EXP: 1/1050 FPS: 800.00





ISO: 17000 EXP: 1/1050 FPS: 800.00





ISO: 17000 EXP: 1/1050 FPS: 800.00





ISO: 17000 EXP: 1/1050 FPS: 800.00



The image is a very blurry, low-resolution photograph. It appears to show a landscape with a body of water in the foreground and a distant, indistinct structure or building in the background. The colors are muted and the overall quality is poor, making it difficult to discern specific details. The text at the bottom provides technical camera data.

ISO: 17000 EXP: 1/1050 FPS: 800.00



ISO: 17000 EXP: 1/1050 FPS: 800.00



# For Best Latency



Render with Metal

Use predicted touches

# For Absolute Best Latency



Render with Metal

Use predicted touches



Transparent Metal layers

Overlay effects



CIE L\*a\*b\* space

↳ marginally better, still used in video

↳ superceded by

IE Lab color space ★

- based on opponent process color vision
- non-linear version of XYZ
- designed to be perceptually uniform
- $\Delta E$  3D Euclidean distance





CIE L\*a\*b\* space

↳ marginally better, still used in video

↳ superceded by

IE Lab color space ★

- based on opponent process color vision
- non-linear version of XYZ
- designed to be perceptually uniform
- $\Delta E$  3D Euclidean distance





# Support Pencil Taps

UIPencilInteraction for listening to Pencil taps

Respect the user's `preferredTapAction`

```
let interaction = UIPencilInteraction()
interaction.delegate = self
view.addInteraction(interaction)

func pencilInteractionDidTap(_ interaction: UIPencilInteraction) {
    switch UIPencilInteraction.preferredTapAction {
        // ...
    }
}
```



# Support Pencil Taps

UIPencilInteraction for listening to Pencil taps

Respect the user's `preferredTapAction`

```
let interaction = UIPencilInteraction()
interaction.delegate = self
view.addInteraction(interaction)

func pencilInteractionDidTap(_ interaction: UIPencilInteraction) {
    switch UIPencilInteraction.preferredTapAction {
        // ...
    }
}
```



# Support Pencil Taps

UIPencilInteraction for listening to Pencil taps

Respect the user's `preferredTapAction`

```
let interaction = UIPencilInteraction()
interaction.delegate = self
view.addInteraction(interaction)

func pencilInteractionDidTap(_ interaction: UIPencilInteraction) {
    switch UIPencilInteraction.preferredTapAction {
        // ...
    }
}
```



# Support Pencil Taps

UIPencilInteraction for listening to Pencil taps

Respect the user's `preferredTapAction`

```
let interaction = UIPencilInteraction()
interaction.delegate = self
view.addInteraction(interaction)

func pencilInteractionDidTap(_ interaction: UIPencilInteraction) {
    switch UIPencilInteraction.preferredTapAction {
        // ...
    }
}
```

Use all the coalesced touches

Handle estimated force

Ensure double

Design some beautiful inks

Dark mode

Create tool and color picker

Avoid over-rendering to save power

Estimated azimuth/altitude

Undo

and zoom

# Great Pencil Experiences

Tiling, scroll

render fast in Metal

Bitmap and vector eraser

Use predicted touches

Smooth the point data

Do most of this asynchronously

Build a data m

rayon

Achieve incredible latency

Pencil

Support pencil tap




**PencilKit**



### Anatomy of a Blue Emperor Butterfly

Identify and describe 7 different areas of the butterfly below

**PAPIL**



**COMPOUND EYE**

made of many hexagonal lenses just like other insect eyes.

**THORAX**

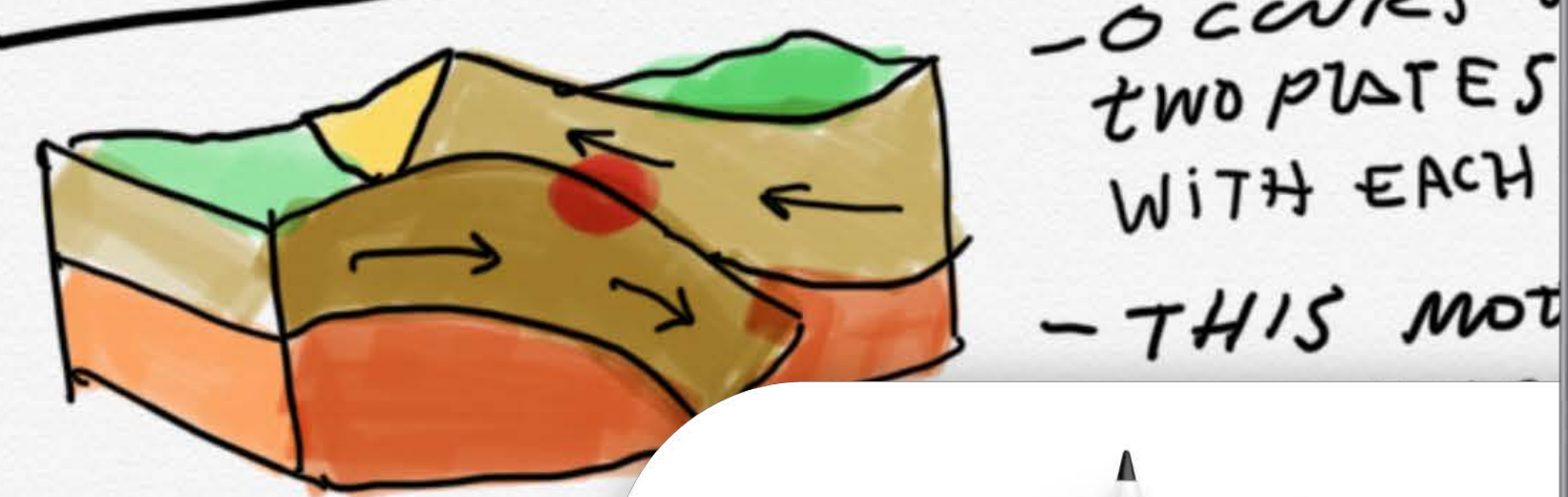
the section of the body where the wings and legs are attached.

**WINGS**

the connective end

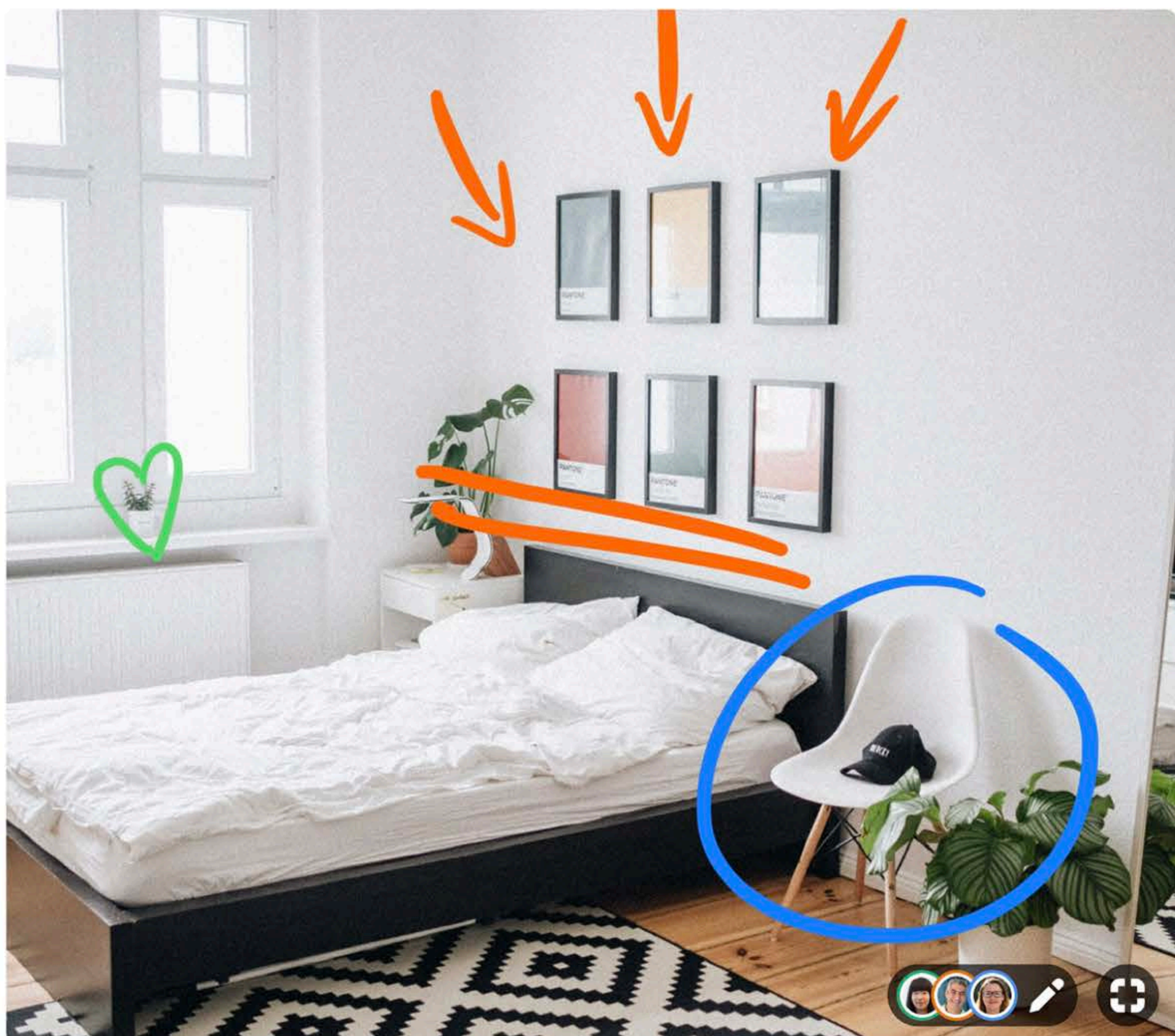
# PLATE TECTONICS

- FACTS**
- PROCESS BEGAN ON EARTH 3.5 BILLION YEARS AGO.
  - THE LITHOSPHERE IS COOLER AND MORE RIGID
  - THE ASTHENOSPHERE IS HOTTER AND HIGHLY VISCOUS
- CONVERGENT PLATE BOUNDARY**



- OCCURS BETWEEN TWO PLATES WITH EACH OTHER

- THIS MOVEMENT



### 10 Best Minimal Scandinavian Bedrooms

See our favorite examples of simple, bright bedrooms with Scandinavian-inspired furniture.

Neutral and simple can also become cozy and snug in a Scandinavian-style bedroom. Think lots of clean, white design accented with a variety of textures. Thick blankets, weathered wood, and linen wallpaper. A bedroom with a fireplace or wood stove can really feel like your very own cottage in a Nordic forest.

Visit

More like this





Three lines of code



# Three Lines of Code

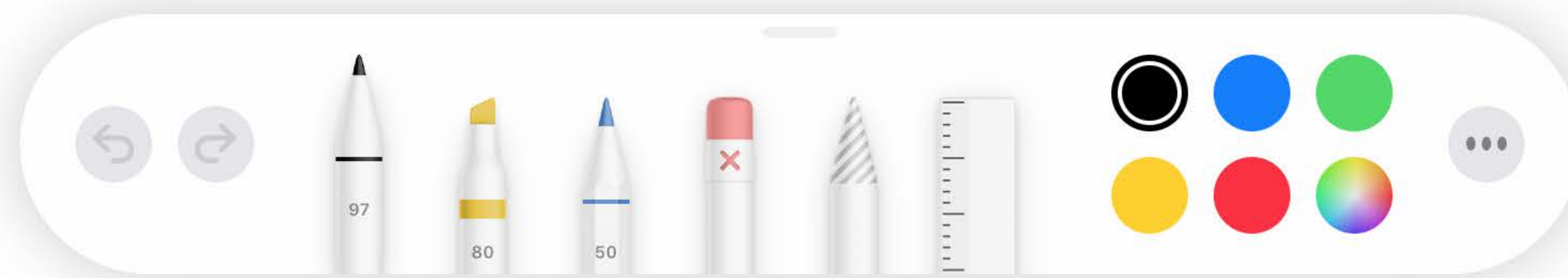
```
let canvas = PKCanvasView(frame: bounds)
view.addSubview(canvas)
canvas.tool = PKInkingTool(.pen, color: .black, width: 30)
```

hello.



***Demo***

hello.





hello.



PKCanvasView

hello.



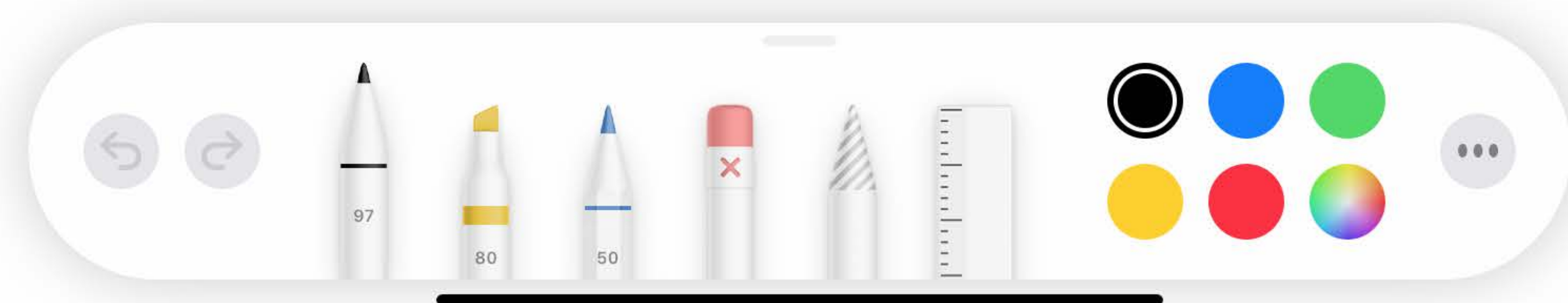
PKCanvasView



9:41 AM Mon Jun 3

100%

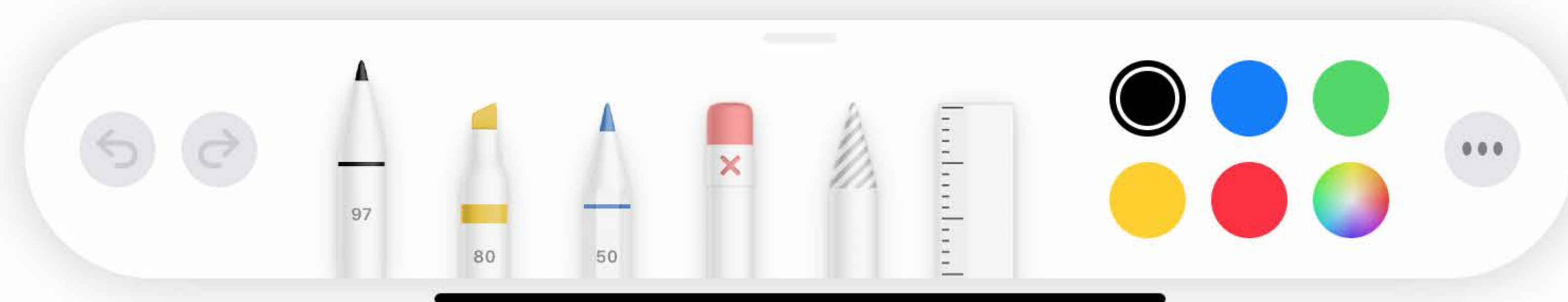
hello.



PKCanvasView

PKDrawing

hello.



PKCanvasView

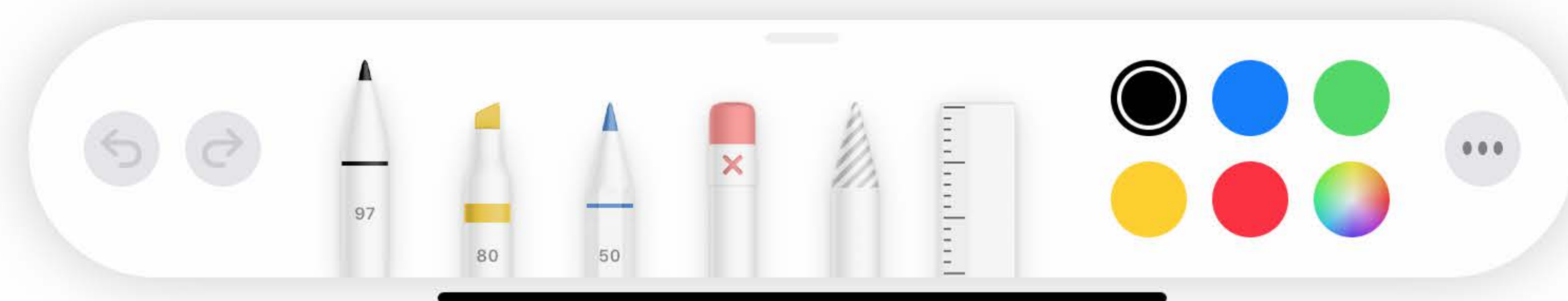
PKDrawing



9:41 AM Mon Jun 3

100%

hello.



PKCanvasView

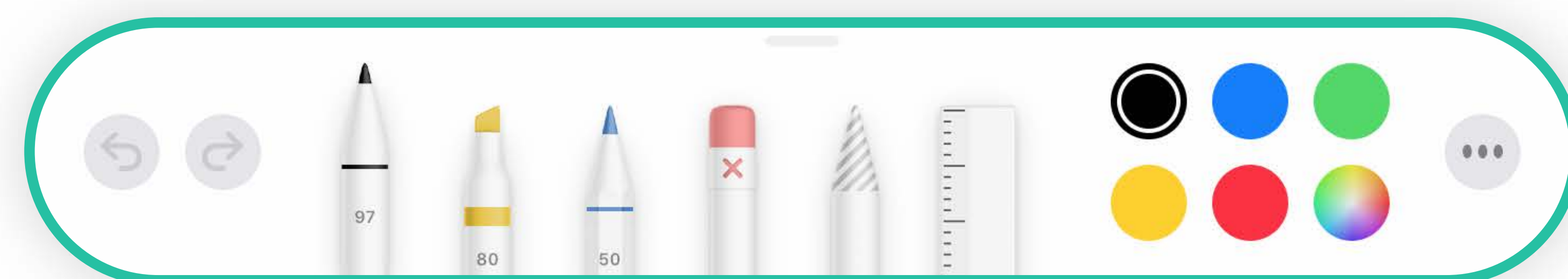
PKDrawing

PKToolPicker

9:41 AM Mon Jun 3

100%

hello.



PKCanvasView

PKDrawing

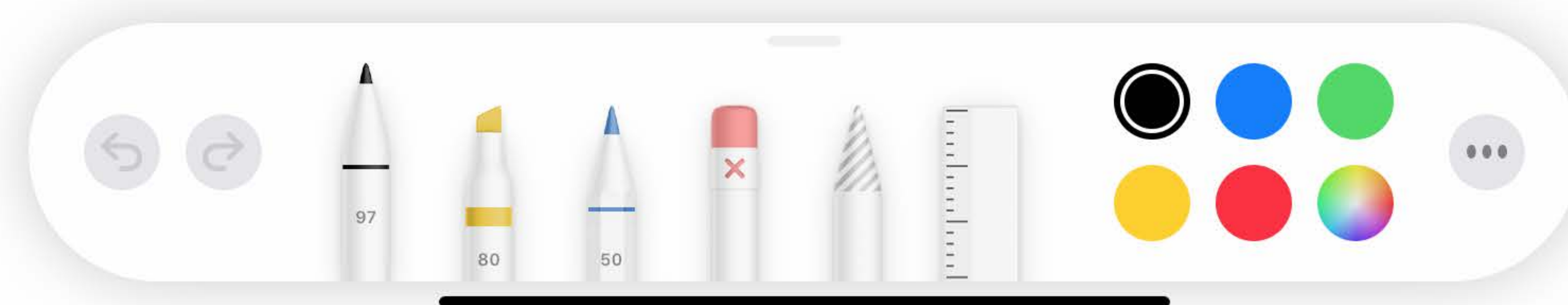
PKToolPicker



9:41 AM Mon Jun 3

100%

hello.



PKCanvasView

PKDrawing

PKToolPicker

PKTools

- PKInkingTool

- PKEraserTool

- PKLassoTool

9:41 AM Mon Jun 3

100%

hello.



PKCanvasView

PKDrawing

PKToolPicker

PKTools

- PKInkingTool
- PKEraserTool
- PKLassoTool



# PKCanvasView

UIScrollView for panning and zooming

Data model is available via `drawing`

Interaction mode is changed via `tool`

Delegate methods notify about changes

# PKDrawing

Serializable to data

Generates images

Can be appended or transformed

Available on macOS



# Thumbnail Generation

```
func thumbnail(drawing: PKDrawing, thumbnailRect: CGRect, traitCollection: UITraitCollection) {
    thumbnailQueue.async {
        traitCollection.performAsCurrent {
            let image = drawing.image(from: thumbnailRect, scale: 2.0)
            DispatchQueue.main.async {
                self.updateThumbnail(drawing, image: image)
            }
        }
    }
}
```

# Thumbnail Generation

```
func thumbnail(drawing: PKDrawing, thumbnailRect: CGRect, traitCollection: UITraitCollection) {
    thumbnailQueue.async {
        traitCollection.performAsCurrent {
            let image = drawing.image(from: thumbnailRect, scale: 2.0)
            DispatchQueue.main.async {
                self.updateThumbnail(drawing, image: image)
            }
        }
    }
}
```



# Thumbnail Generation

```
func thumbnail(drawing: PKDrawing, thumbnailRect: CGRect, traitCollection: UITraitCollection) {
    thumbnailQueue.async {
        traitCollection.performAsCurrent {
            let image = drawing.image(from: thumbnailRect, scale: 2.0)
            DispatchQueue.main.async {
                self.updateThumbnail(drawing, image: image)
            }
        }
    }
}
```

# Thumbnail Generation

```
func thumbnail(drawing: PKDrawing, thumbnailRect: CGRect, traitCollection: UITraitCollection) {
    thumbnailQueue.async {
        traitCollection.performAsCurrent {
            let image = drawing.image(from: thumbnailRect, scale: 2.0)
            DispatchQueue.main.async {
                self.updateThumbnail(drawing, image: image)
            }
        }
    }
}
```



# Thumbnail Generation

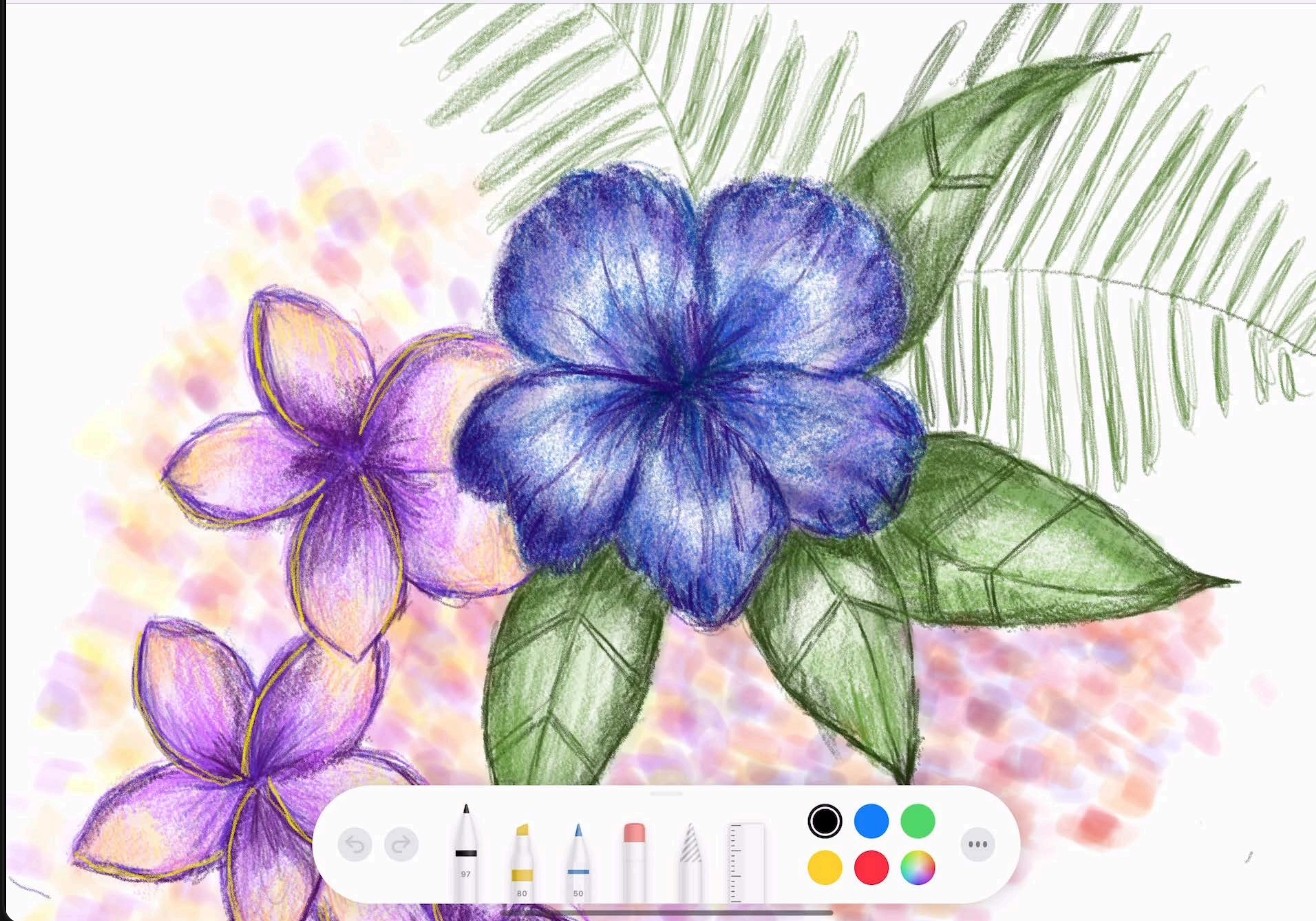
```
func thumbnail(drawing: PKDrawing, thumbnailRect: CGRect, traitCollection: UITraitCollection) {
    thumbnailQueue.async {
        traitCollection.performAsCurrent {
            let image = drawing.image(from: thumbnailRect, scale: 2.0)
            DispatchQueue.main.async {
                self.updateThumbnail(drawing, image: image)
            }
        }
    }
}
```





< Back

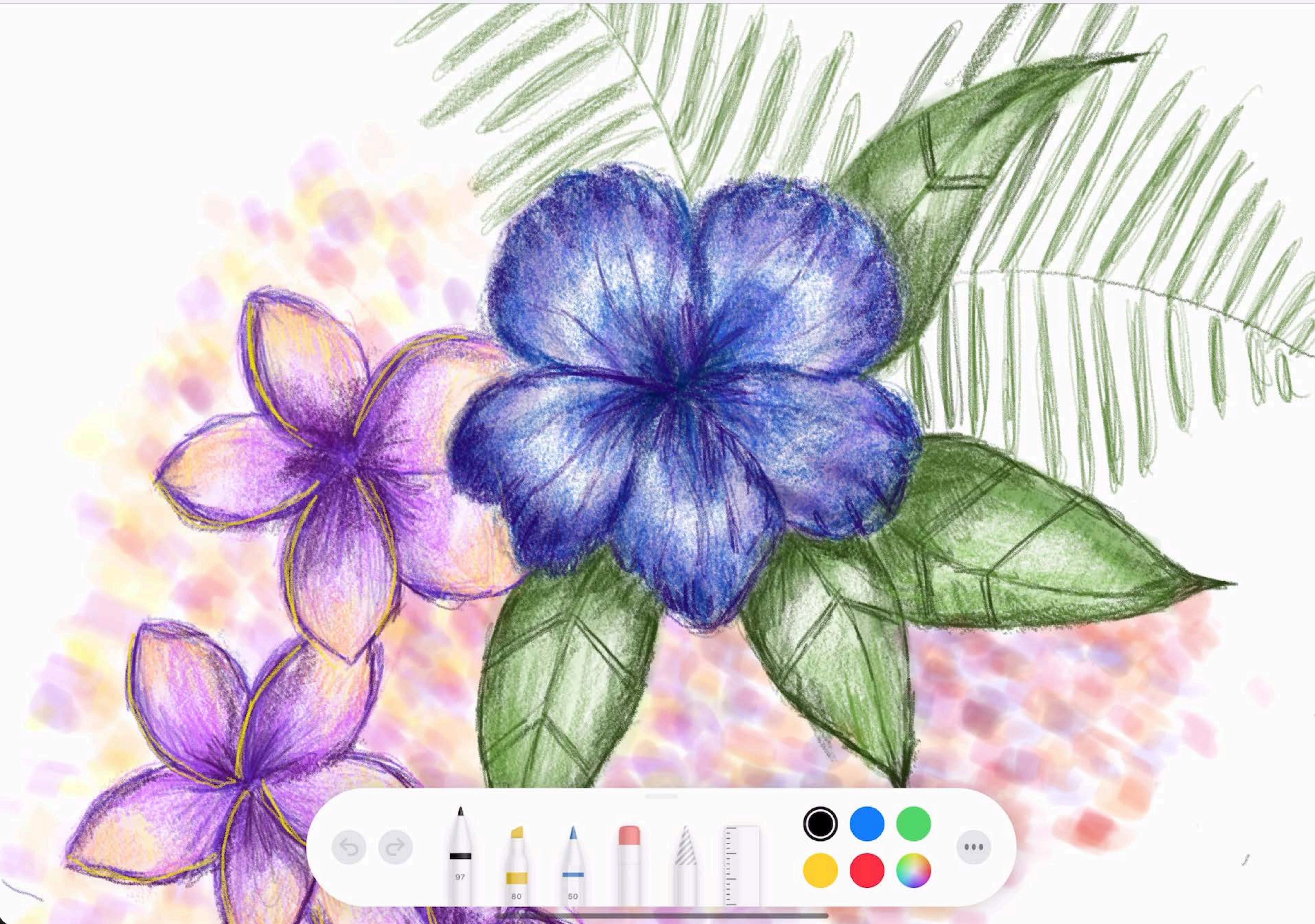
Pencil Signature





< Back

Pencil Signature





# Tools



PKInkingTool  
pen

# Tools



PKInkingTool  
pen



# Tools



PKInkingTool  
marker



# Tools



PKInkingTool  
marker



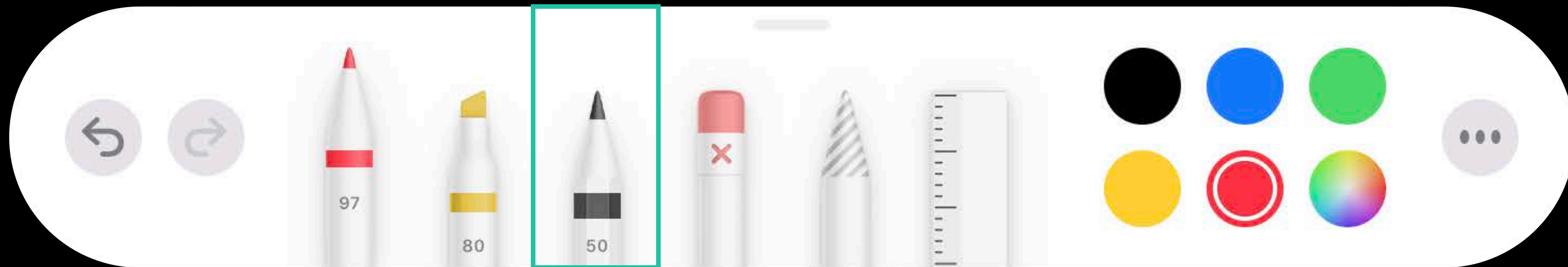
# Tools



PKInkingTool  
pencil



# Tools



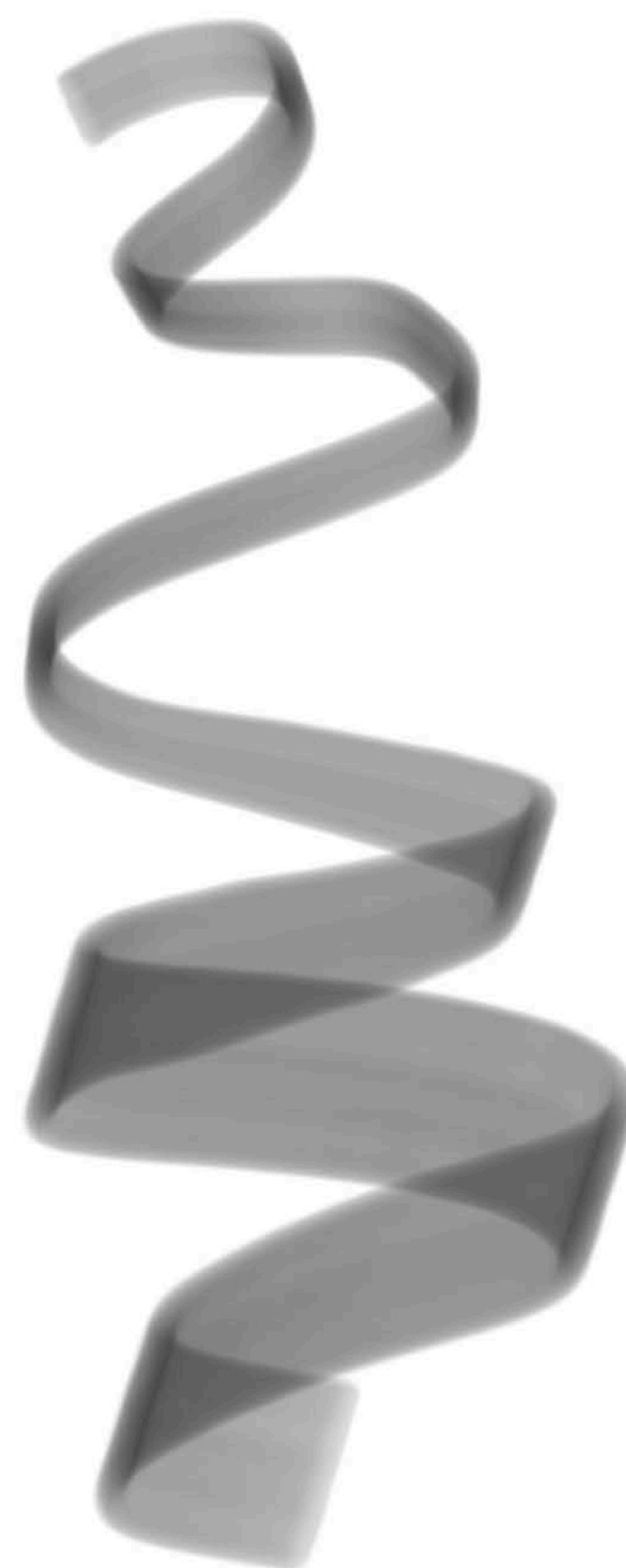
PKInkingTool  
pencil



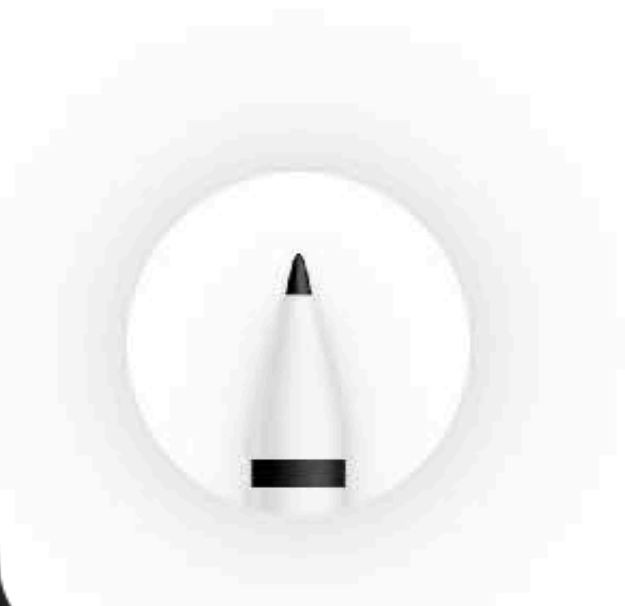
Pen



Marker



Pencil





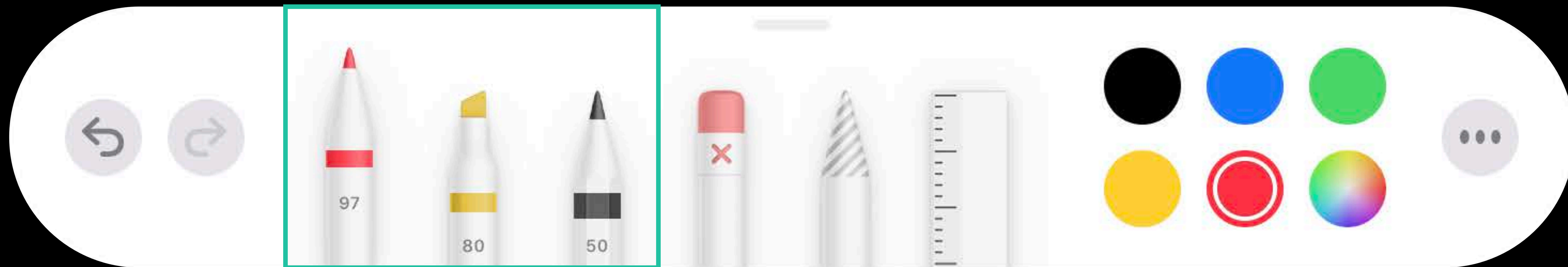
# Tools



```
canvasView.tool = PKInkingTool(.pen, color: .blue, width: 10)
```



# Tools

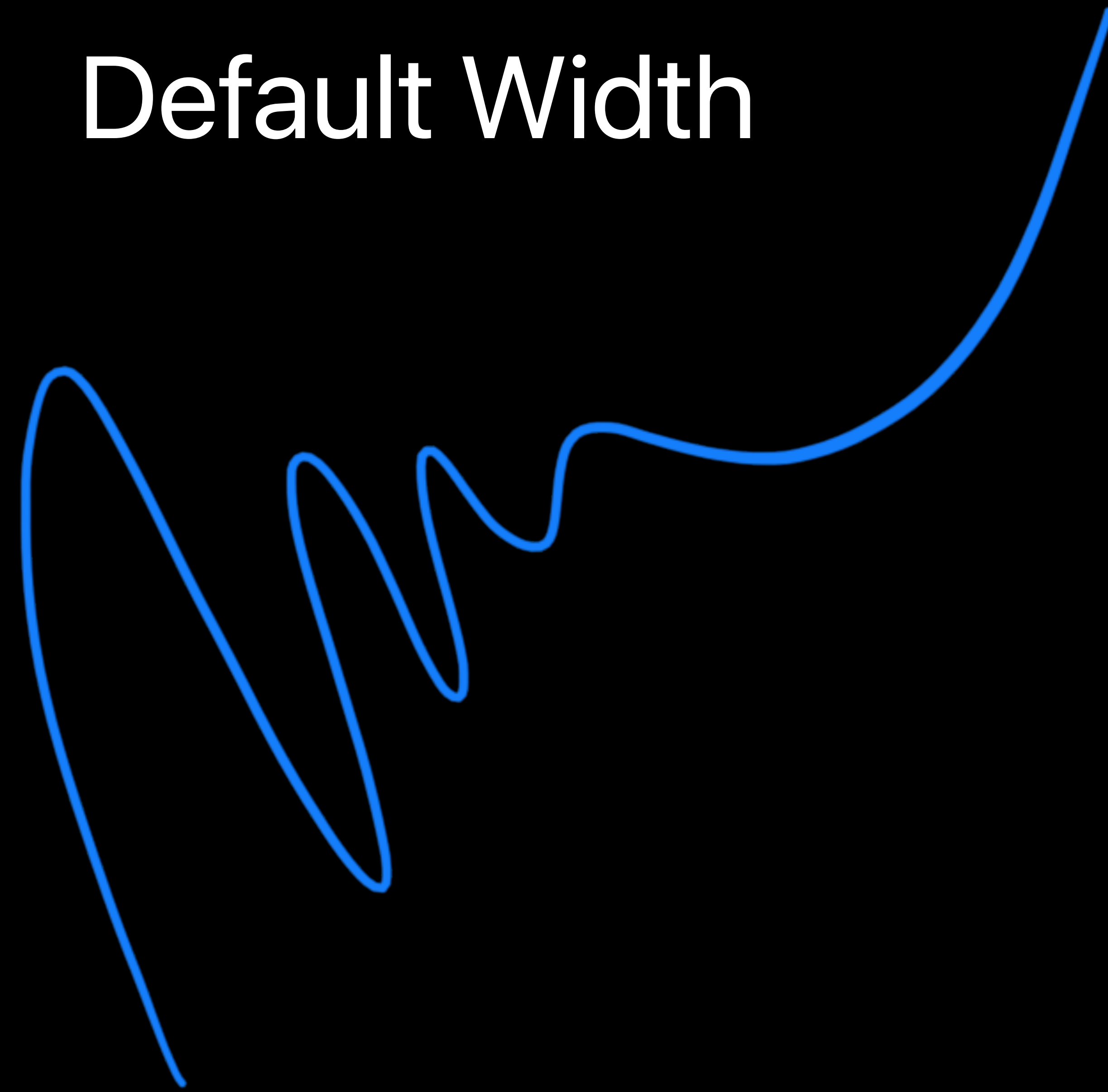


```
canvasView.tool = PKinkingTool(.pen, color: .blue, width: 10)
```

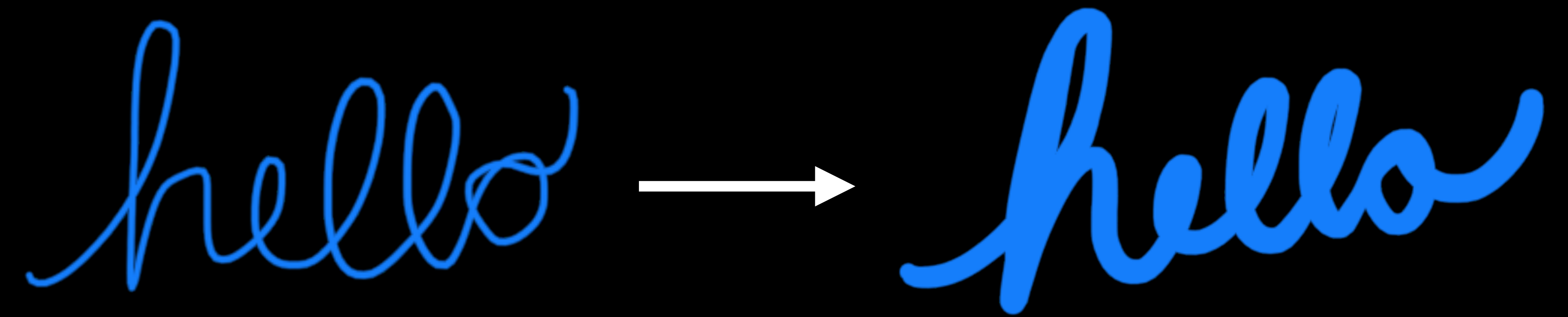


# Tools

Default Width



Valid Width Range













# Tools



## PKEraserTool



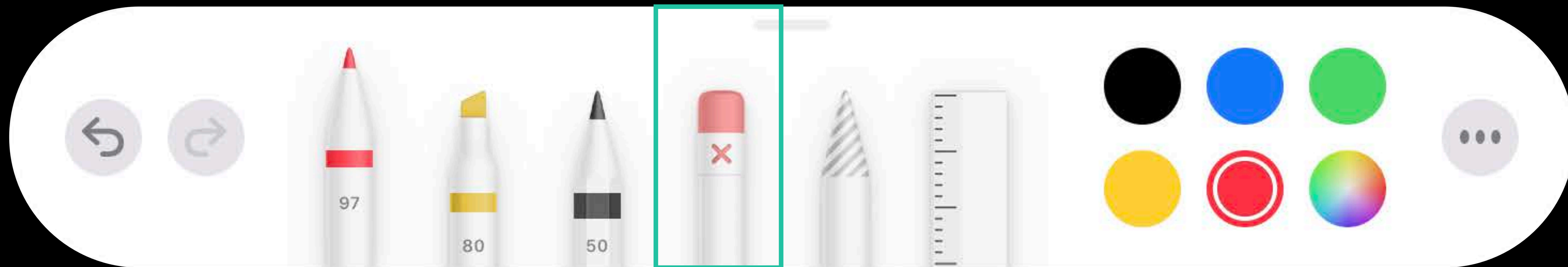
vector



bitmap



# Tools



PKEraserTool



vector

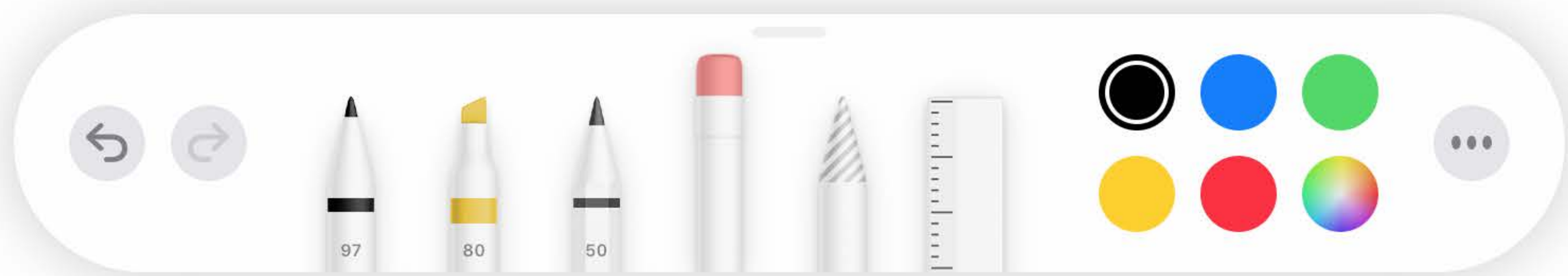


bitmap



< Back

Pencil Signature





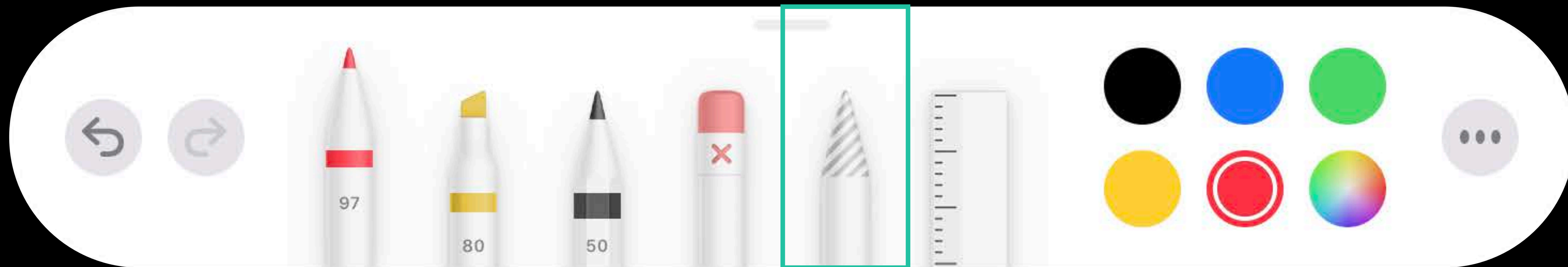
# Tools



PKLassoTool

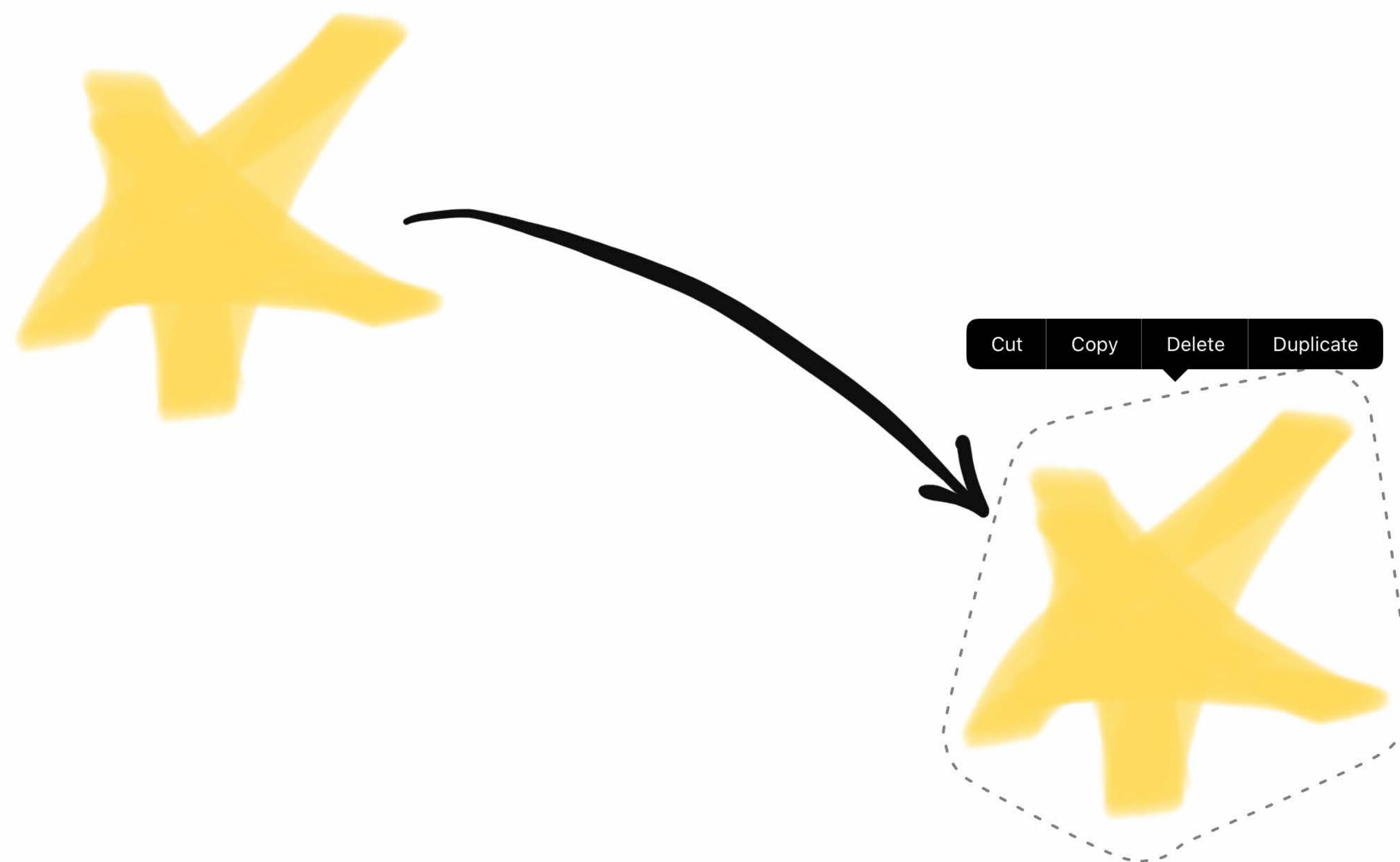


# Tools



PKLassoTool







# Tools

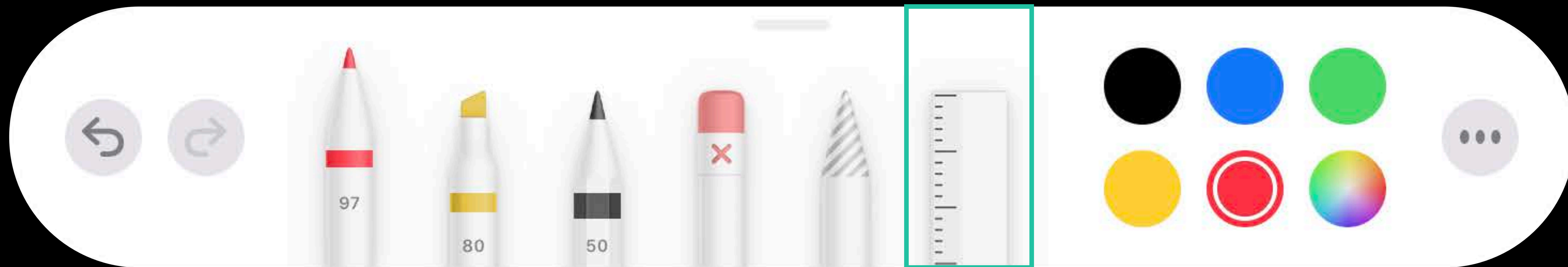


Ruler

```
canvasView.rulerActive = true
```



# Tools



Ruler

```
canvasView.rulerActive = true
```



A vertical toolbar on the left side of the screen containing various drawing tools: a pencil, two highlighters (purple and red), a brush, a ruler, and a color palette with yellow, black, red, blue, and green circles. There are also undo and redo icons at the top of the toolbar.

Welcome  
to  
San Francisco





**PKToolPicker**



# PKToolPicker

Responder-based visibility

Similar to the keyboard

Palette floats over everything

Visibility based on first responders



# PKToolPicker

## Responder-based visibility

```
let toolPicker = PKToolPicker.shared(for: window)
toolPicker.addObserver(canvasView)
toolPicker.setVisible(true, forFirstResponder: canvasView)
canvasView.becomeFirstResponder()
```



# PKToolPicker

## Responder-based visibility

```
let toolPicker = PKToolPicker.shared(for: window)
toolPicker.addObserver(canvasView)
toolPicker.setVisible(true, forFirstResponder: canvasView)
canvasView.becomeFirstResponder()
```



# PKToolPicker

## Responder-based visibility

```
let toolPicker = PKToolPicker.shared(for: window)
toolPicker.addObserver(canvasView)
toolPicker.setVisible(true, forFirstResponder: canvasView)
canvasView.becomeFirstResponder()
```



# PKToolPicker

## Responder-based visibility

```
let toolPicker = PKToolPicker.shared(for: window)
toolPicker.addObserver(canvasView)
toolPicker.setVisible(true, forFirstResponder: canvasView)
canvasView.becomeFirstResponder()
```



# PKToolPicker

## Responder-based visibility

```
let toolPicker = PKToolPicker.shared(for: window)
toolPicker.addObserver(canvasView)
toolPicker.setVisible(true, forFirstResponder: canvasView)
canvasView.becomeFirstResponder()
```



# PKToolPicker

## Responder-based visibility

```
let toolPicker = PKToolPicker.shared(for: window)
toolPicker.addObserver(canvasView)
toolPicker.setVisible(true, forFirstResponder: canvasView)
canvasView.becomeFirstResponder()
```



# PKToolPicker

## Responder-based visibility

```
let toolPicker = PKToolPicker.shared(for: window)
toolPicker.addObserver(canvasView)
toolPicker.setVisible(true, forFirstResponder: canvasView)
canvasView.becomeFirstResponder()
```

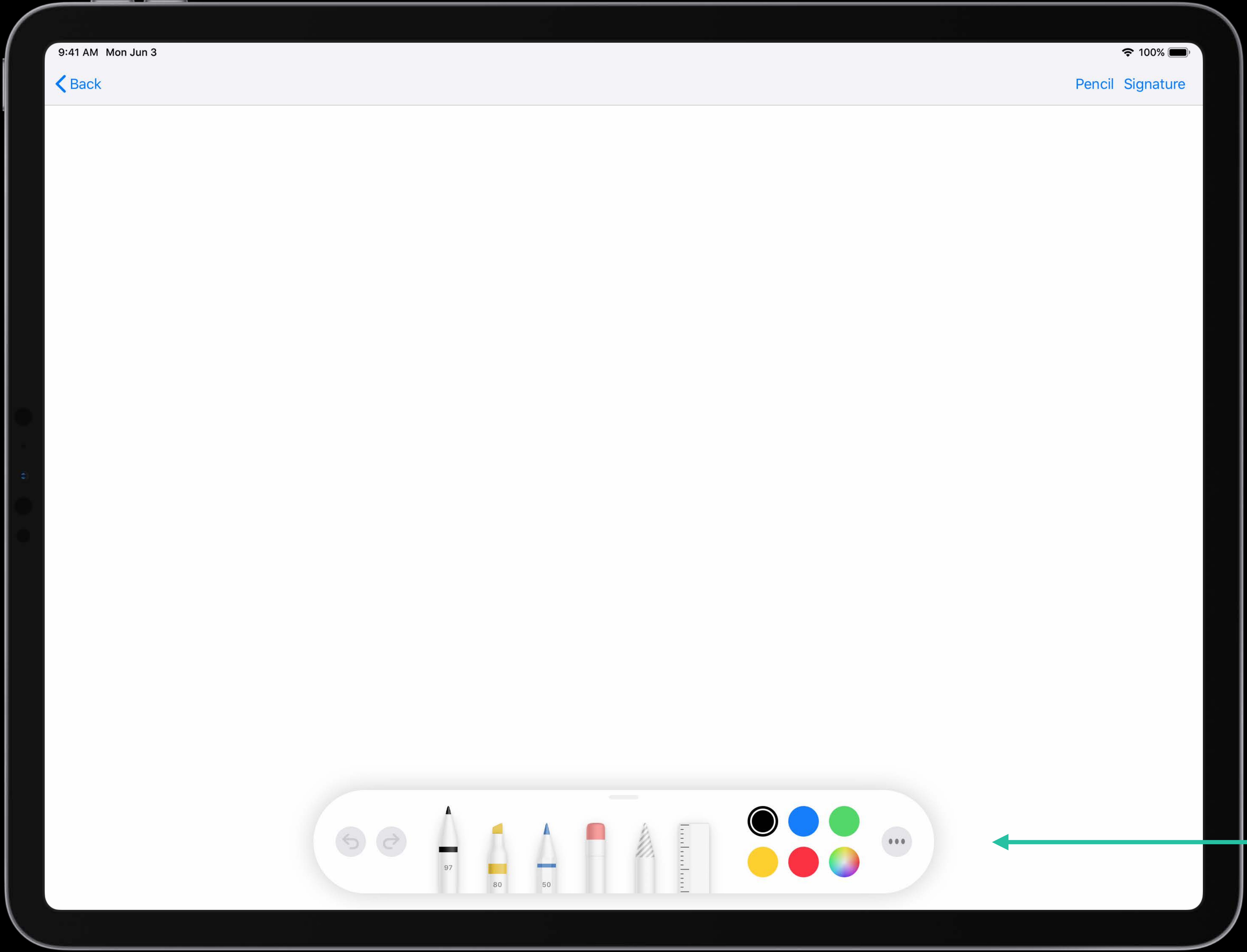


# PKToolPicker

## Responder-based visibility

```
let toolPicker = PKToolPicker.shared(for: window)
toolPicker.addObserver(canvasView)
toolPicker.setVisible(true, forFirstResponder: canvasView)
canvasView.becomeFirstResponder()
```





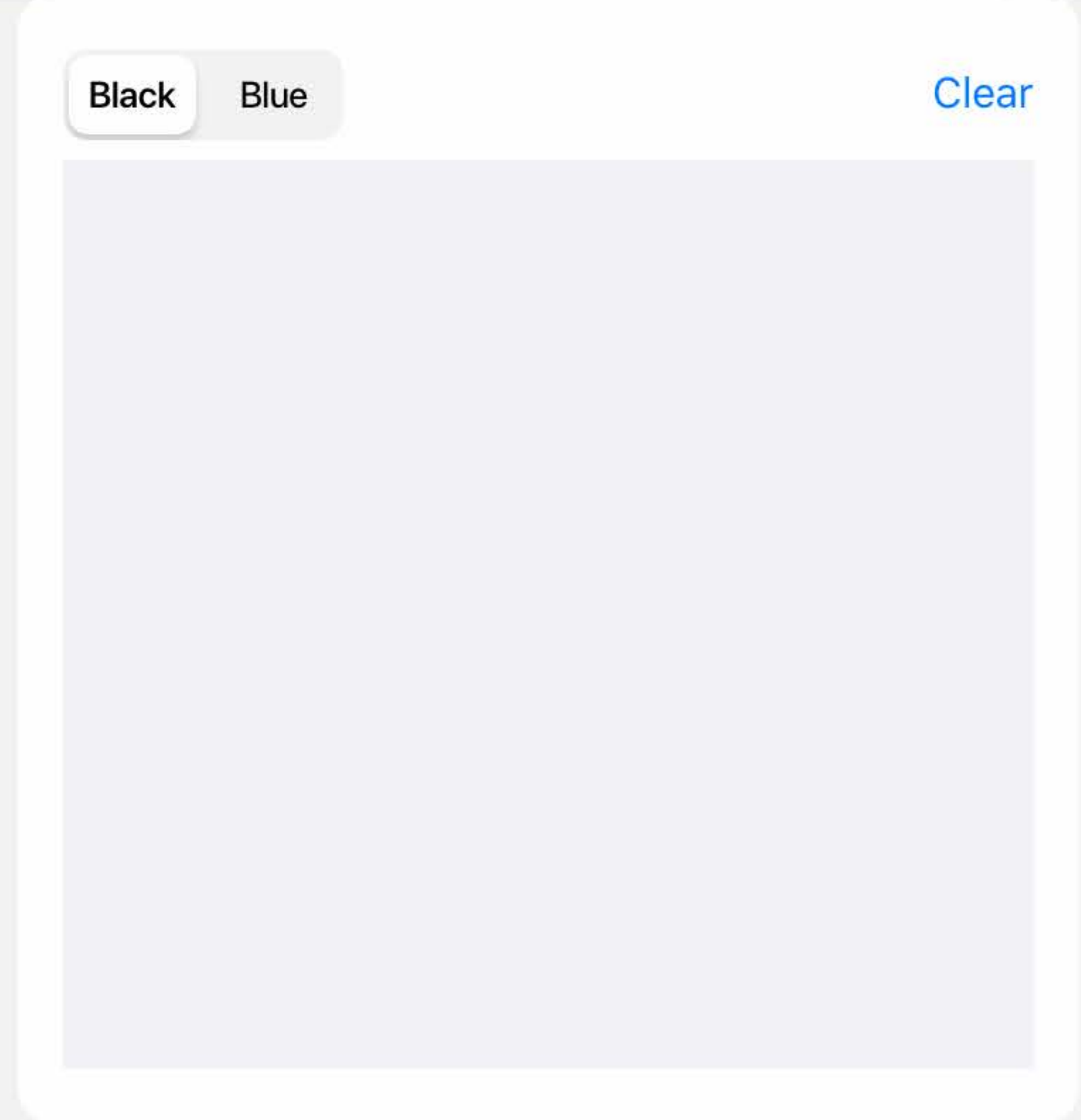
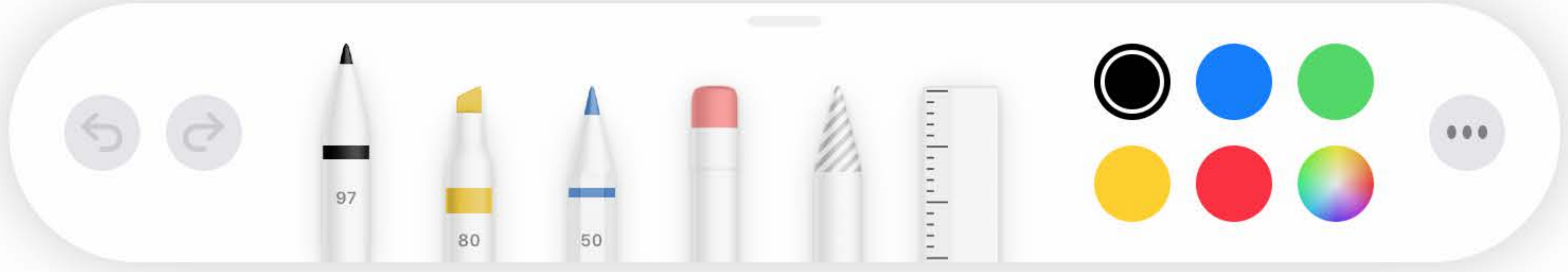
Canvas is first responder  
Tool picker is shown



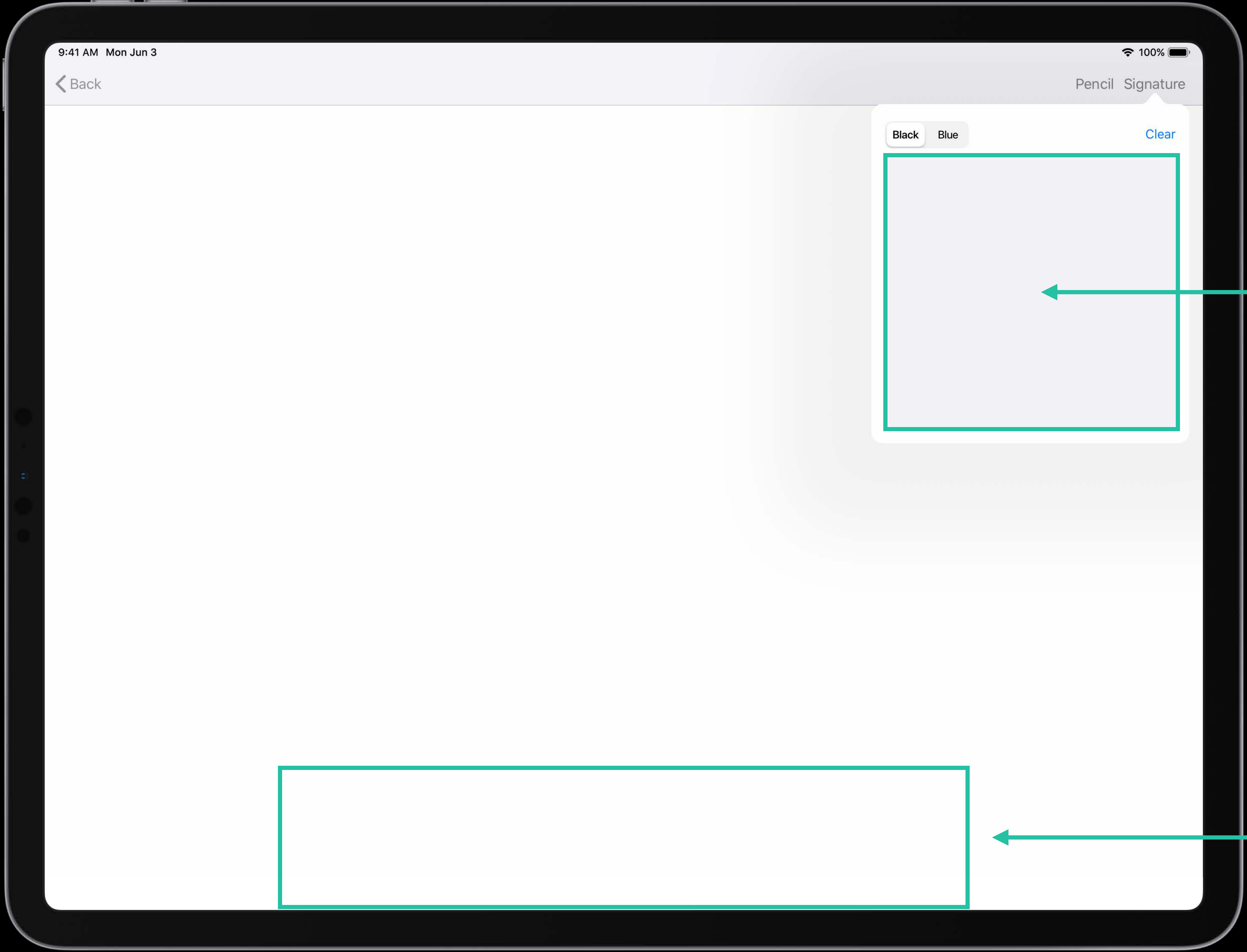
< Back

Pencil Signature

Black Blue Clear



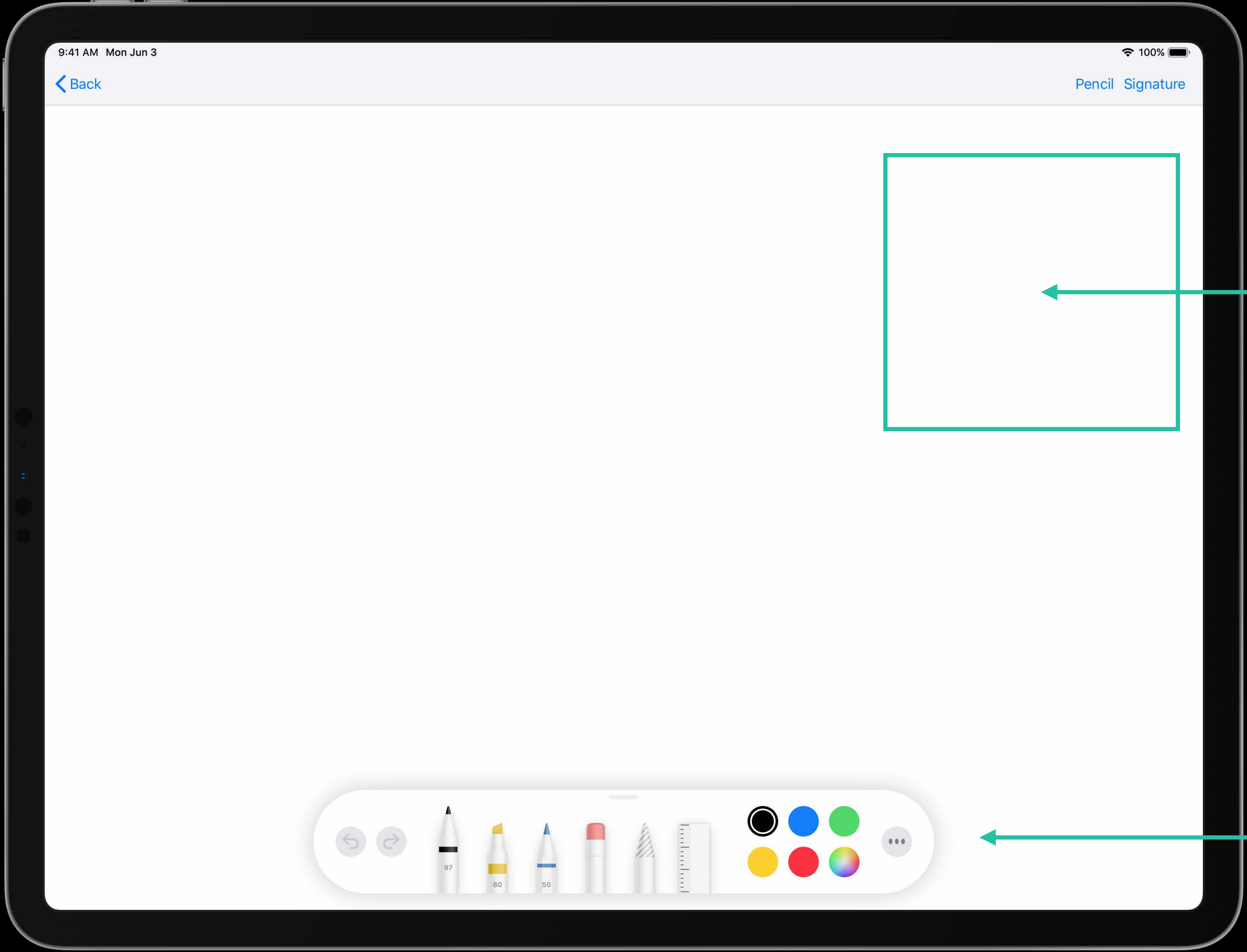




becomeFirst Responder

Tool picker goes away





Dismiss Signature View Controller  
Automatically resigns first responder

Tool picker shows again

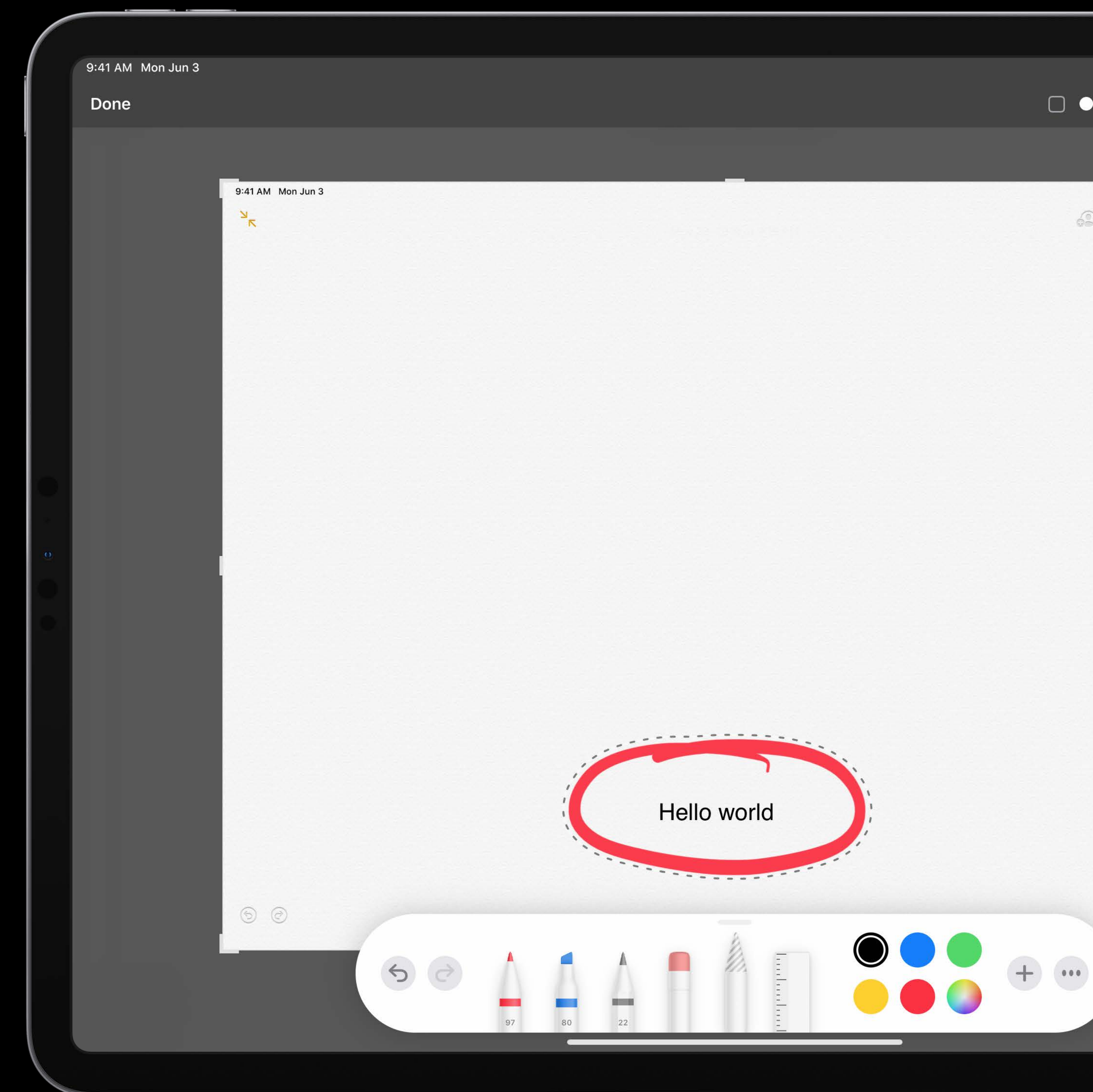


# PKToolPicker

Responder-based visibility

Keep palette visible with other responders

```
public func set(_ visible: Bool for responder: UIResponder)
```



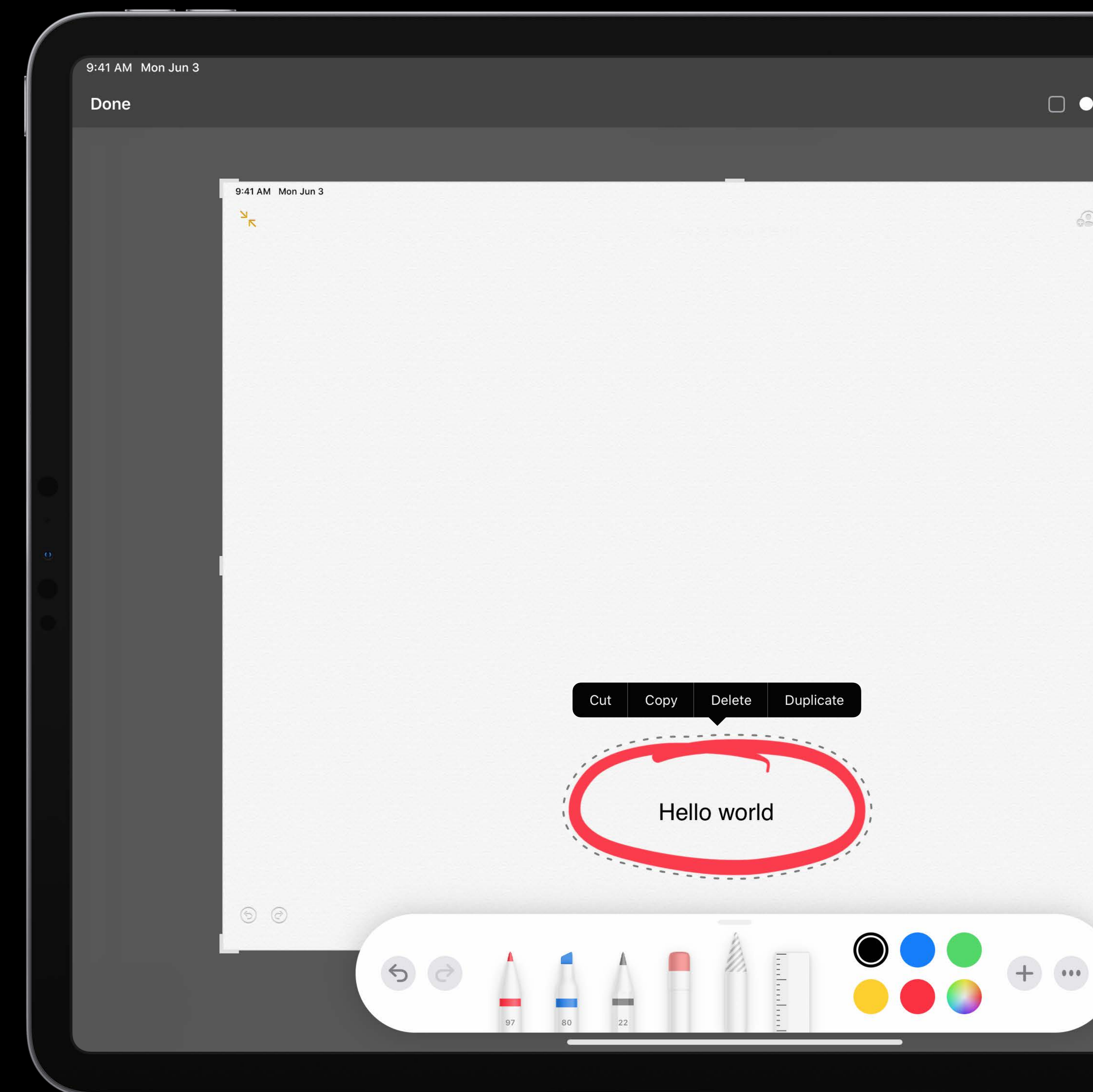


# PKToolPicker

Responder-based visibility

Keep palette visible with other responders

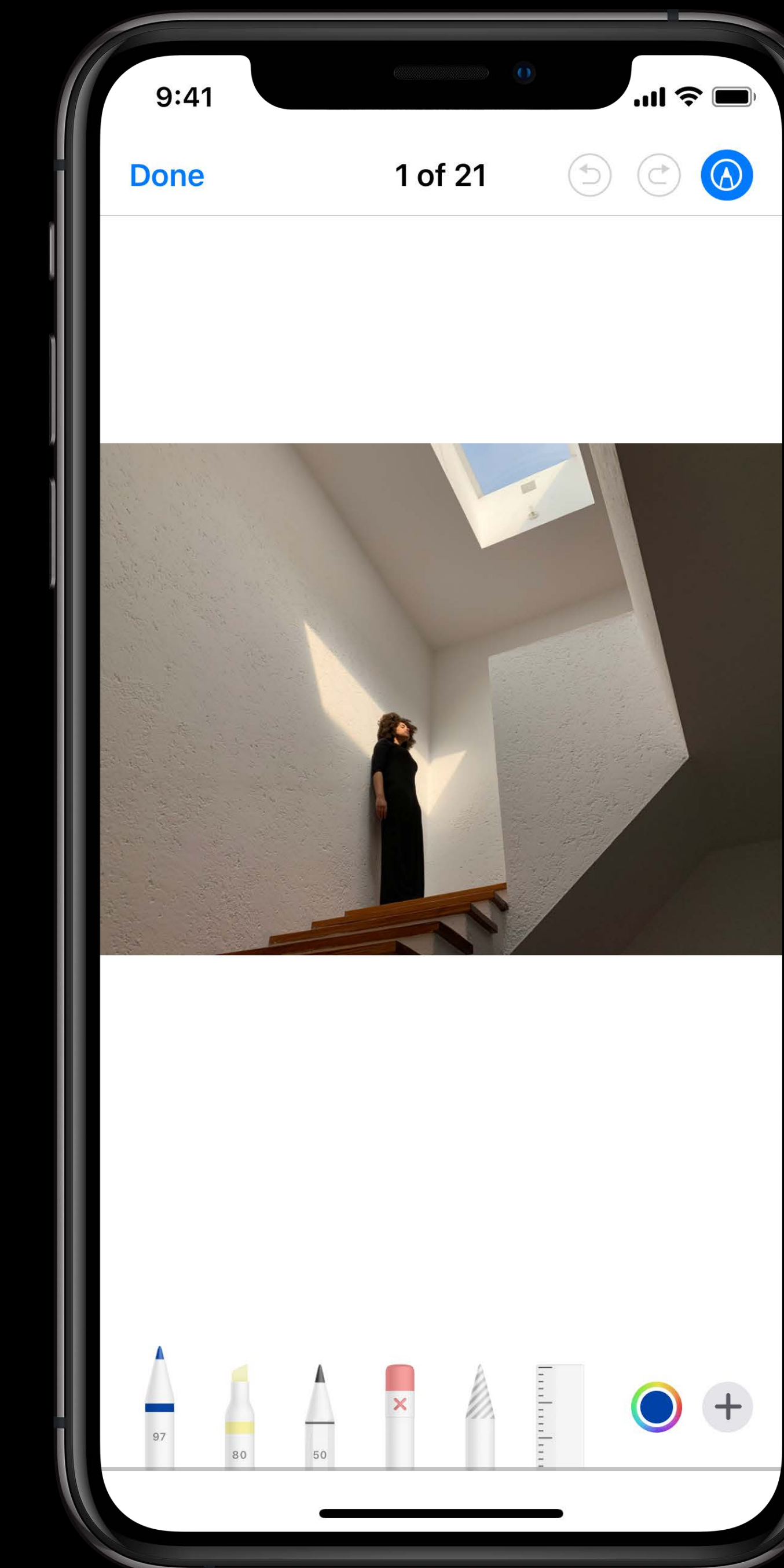
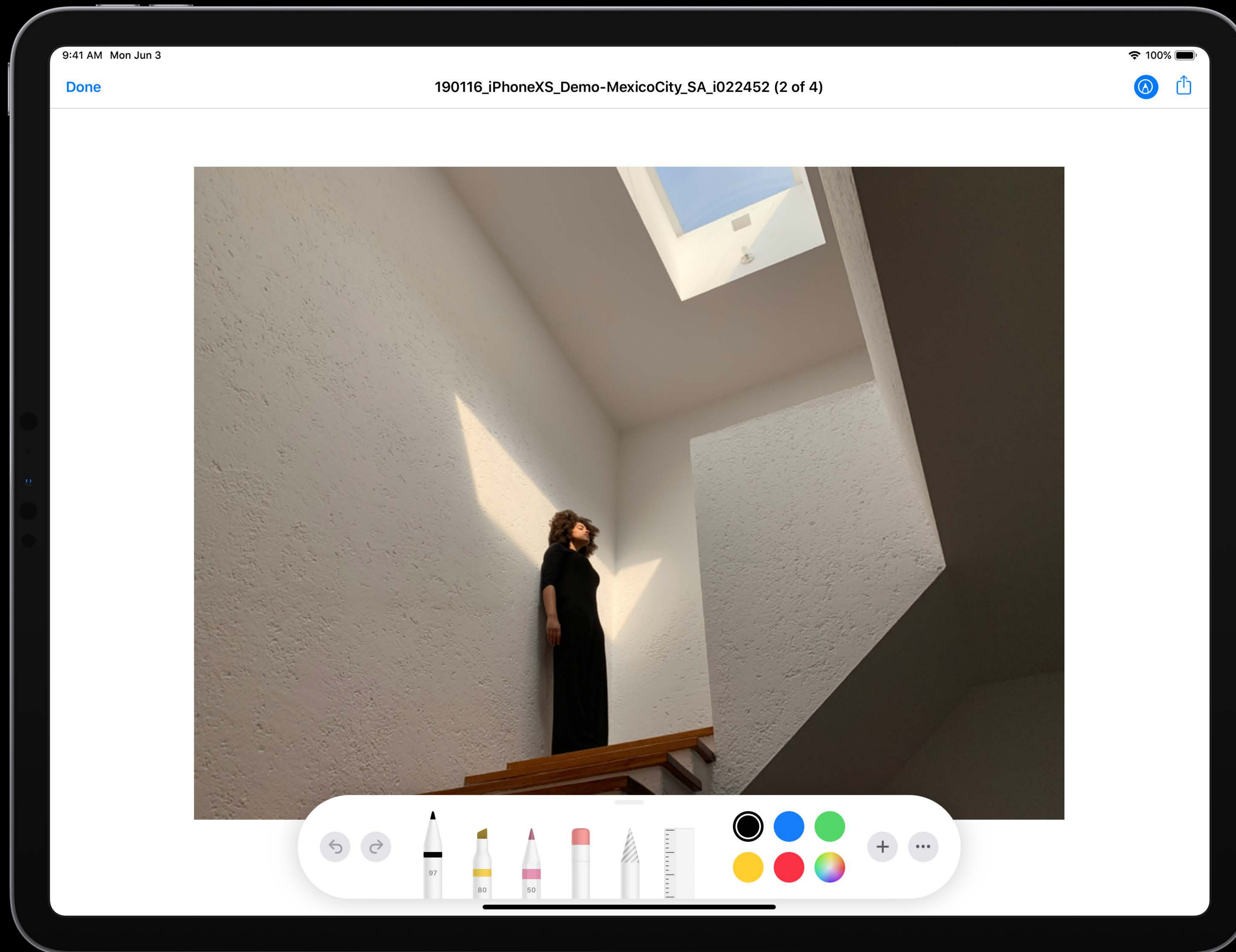
```
public func set(_ visible: Bool for responder: UIResponder)
```





# PKToolPicker

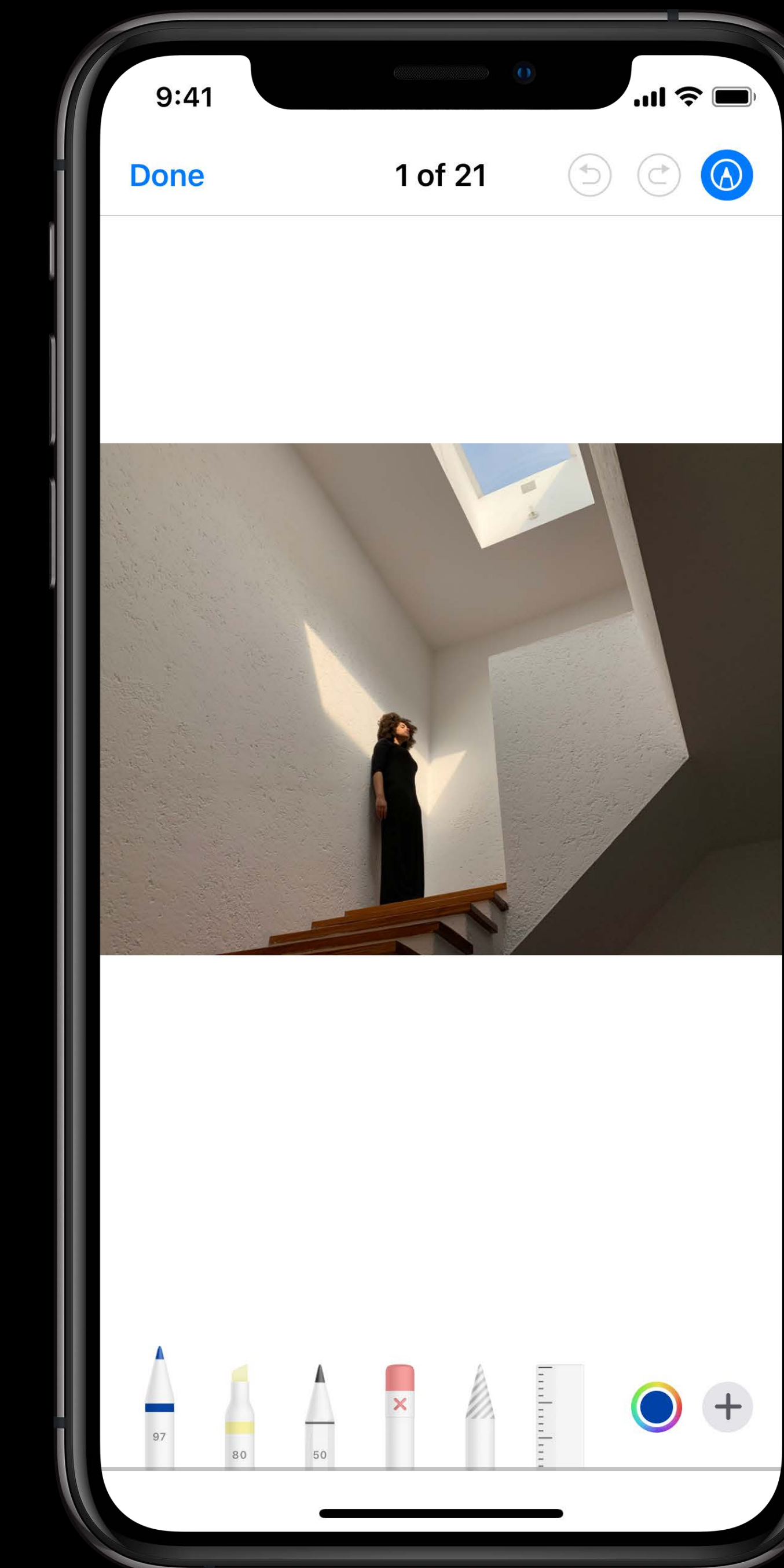
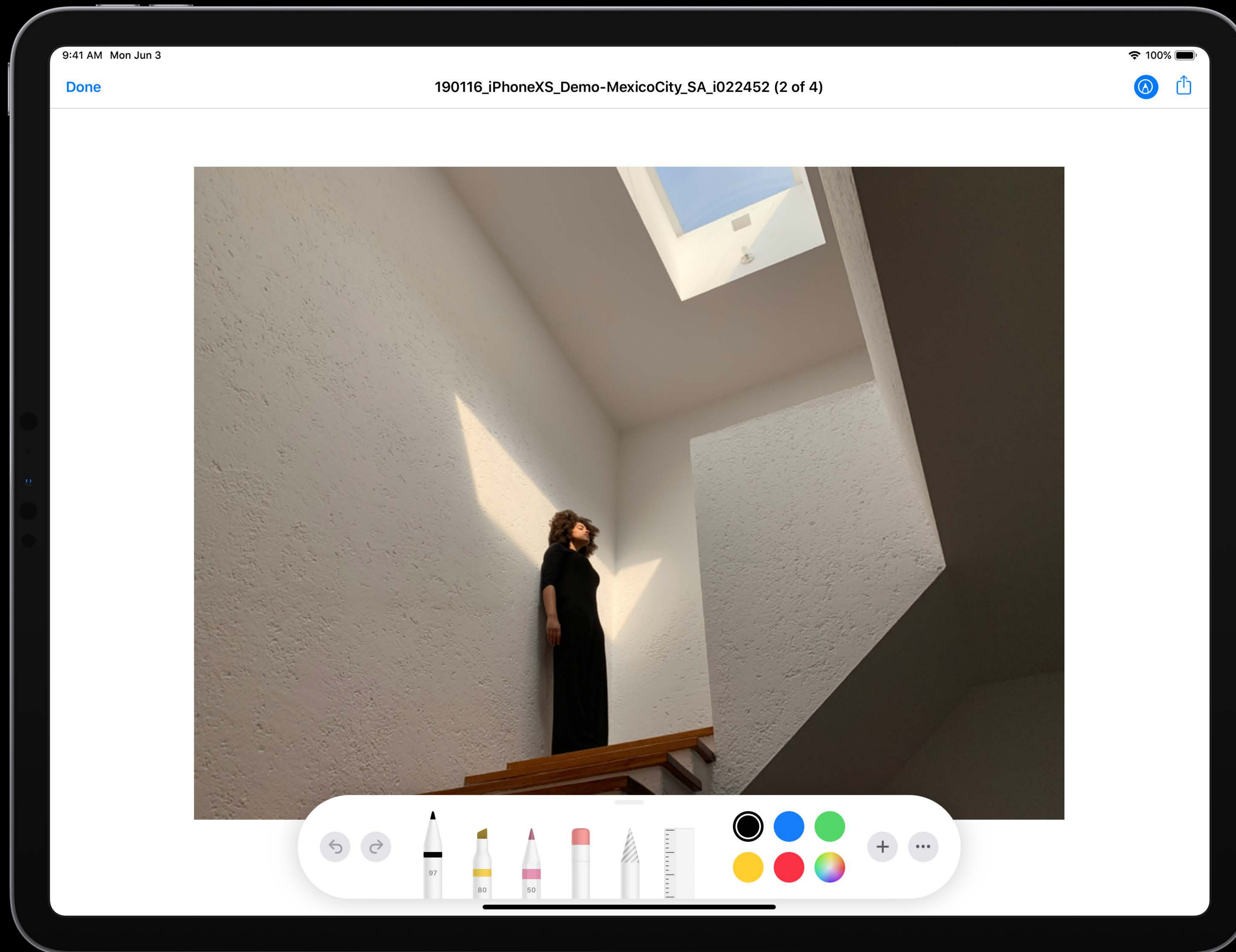
## Tool Picker UI — Compact layout





# PKToolPicker

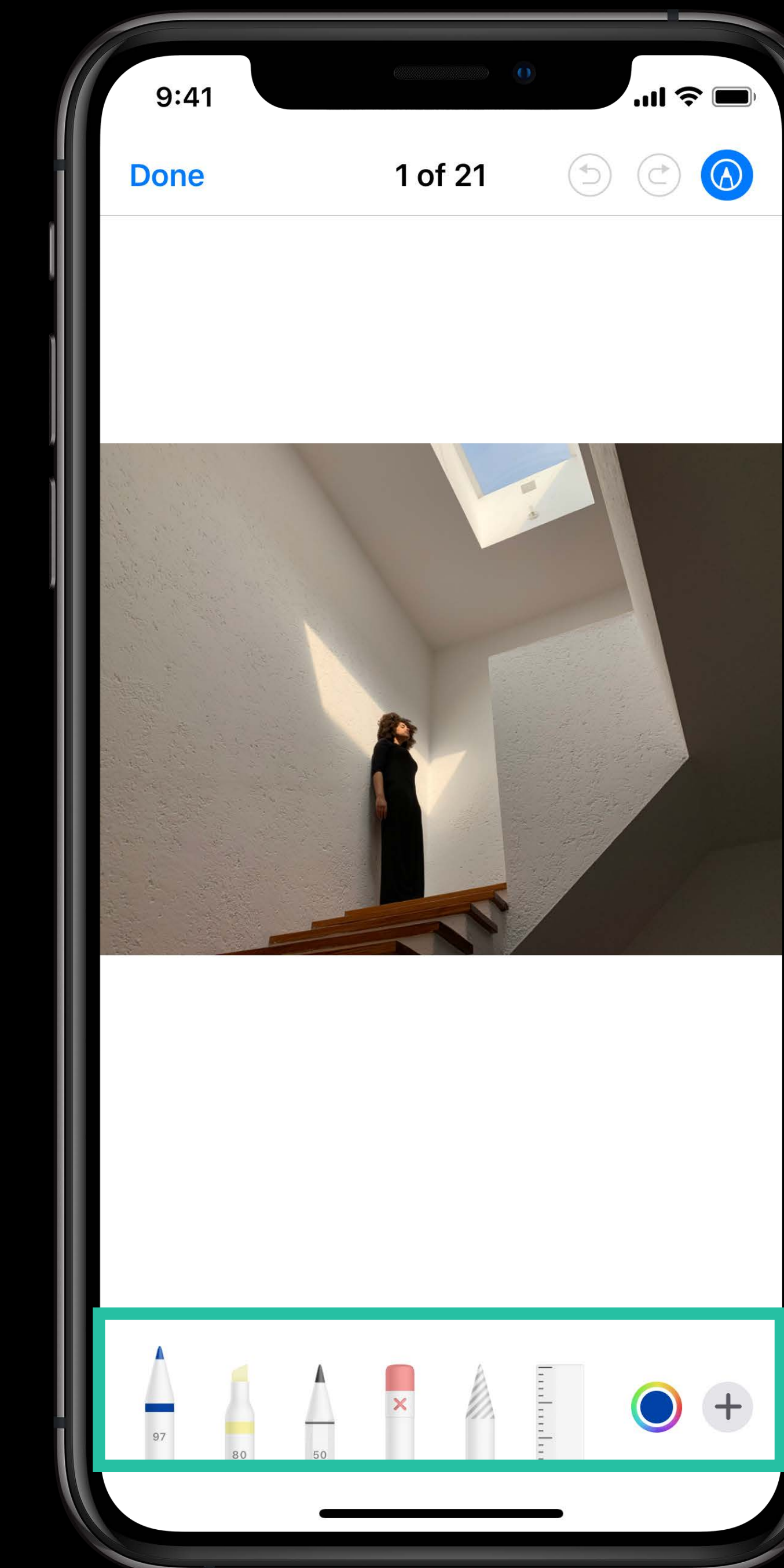
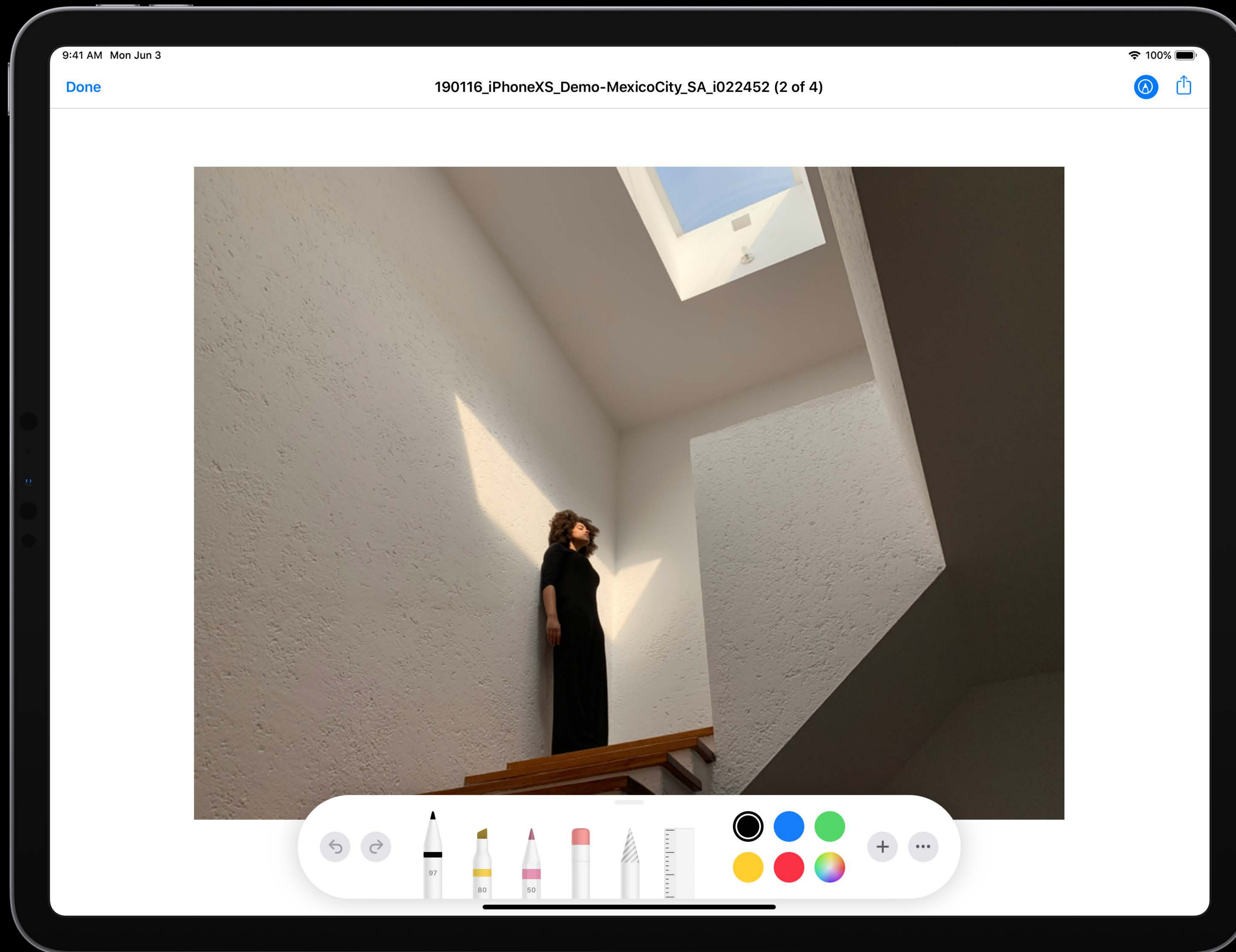
## Tool Picker UI — Compact layout





# PKToolPicker

## Tool Picker UI — Compact layout

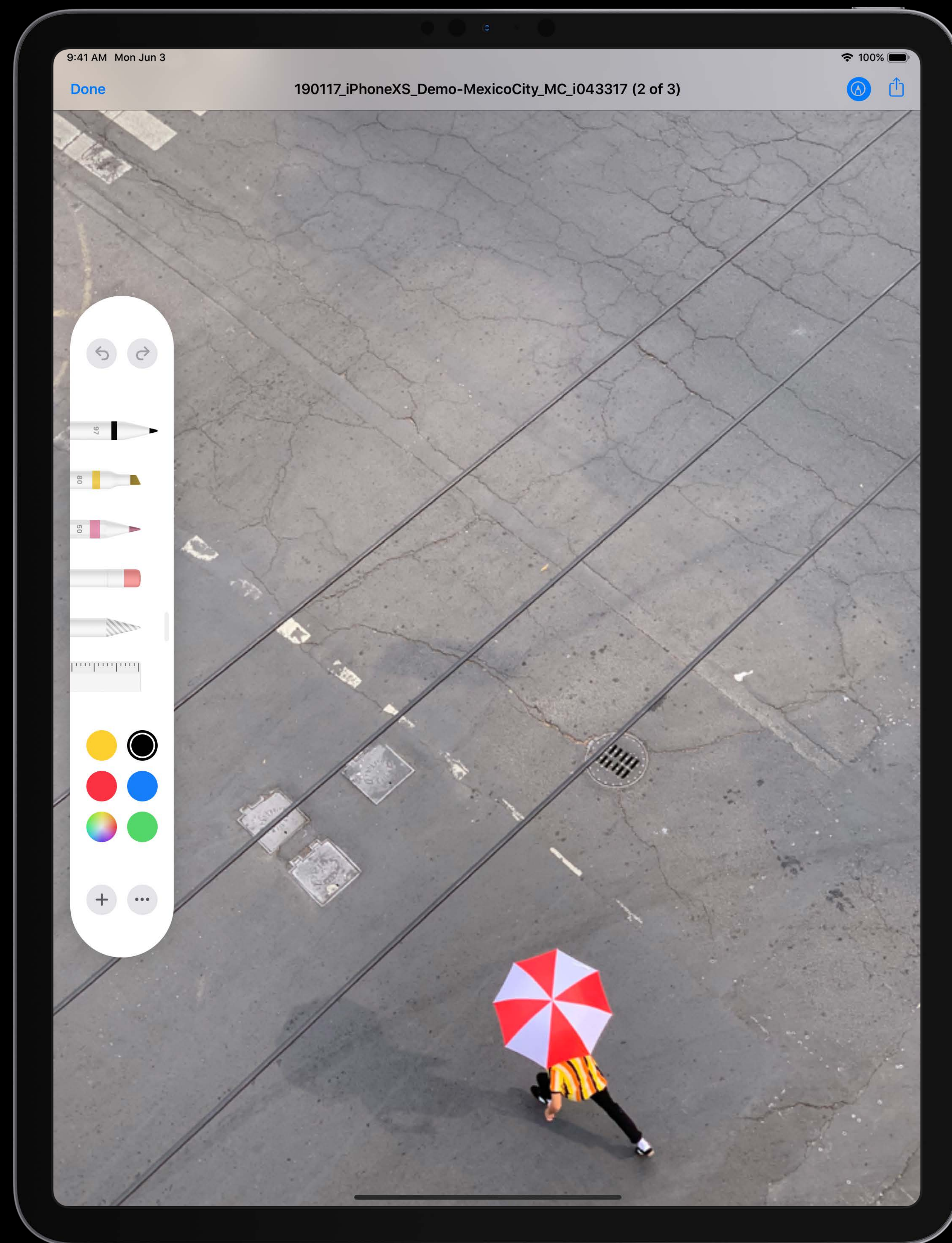


Fixed  
and docked  
to the bottom



# PKToolPicker

Tool Picker UI — Compact layout

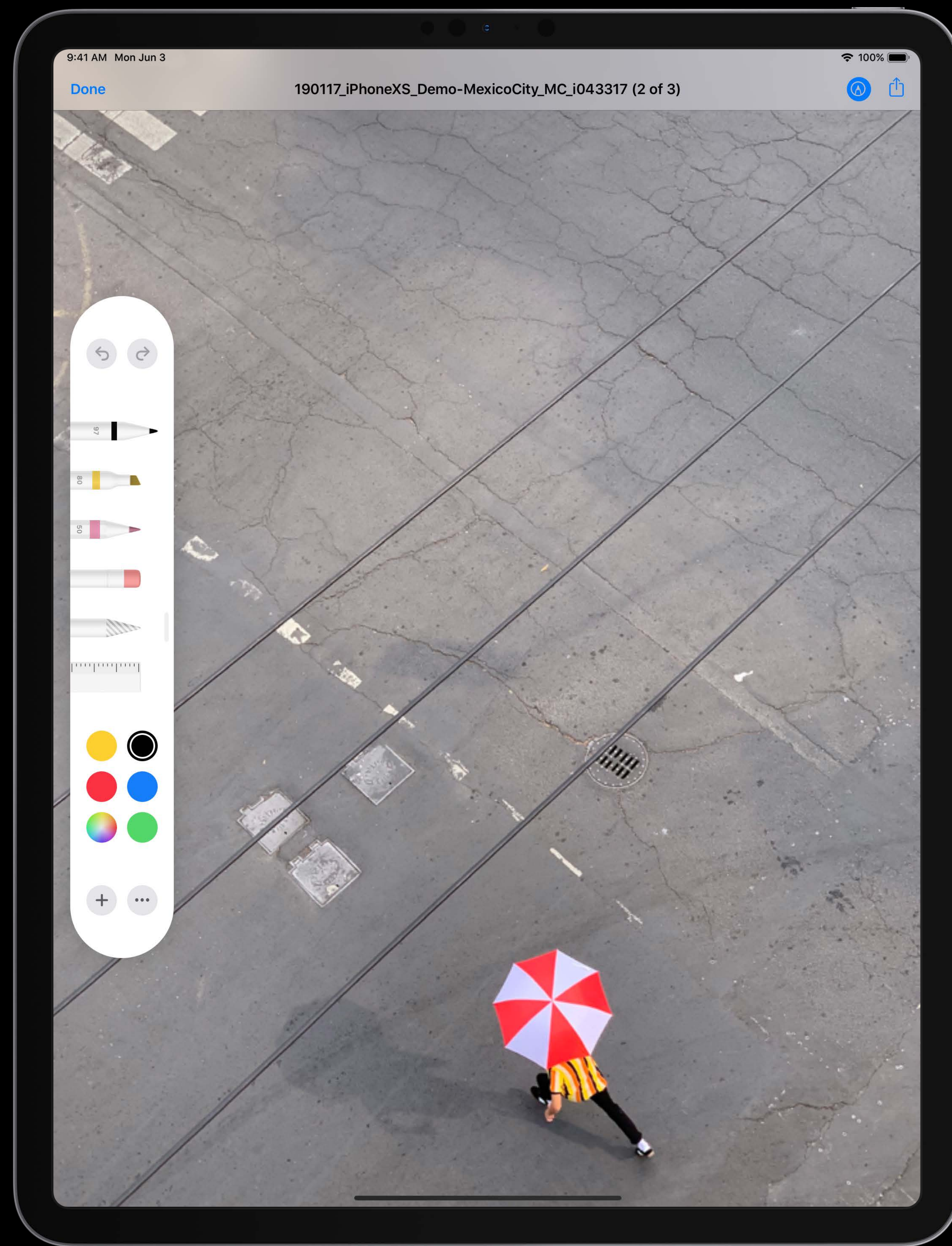


iPad



# PKToolPicker

## Tool Picker UI — Compact layout



iPad

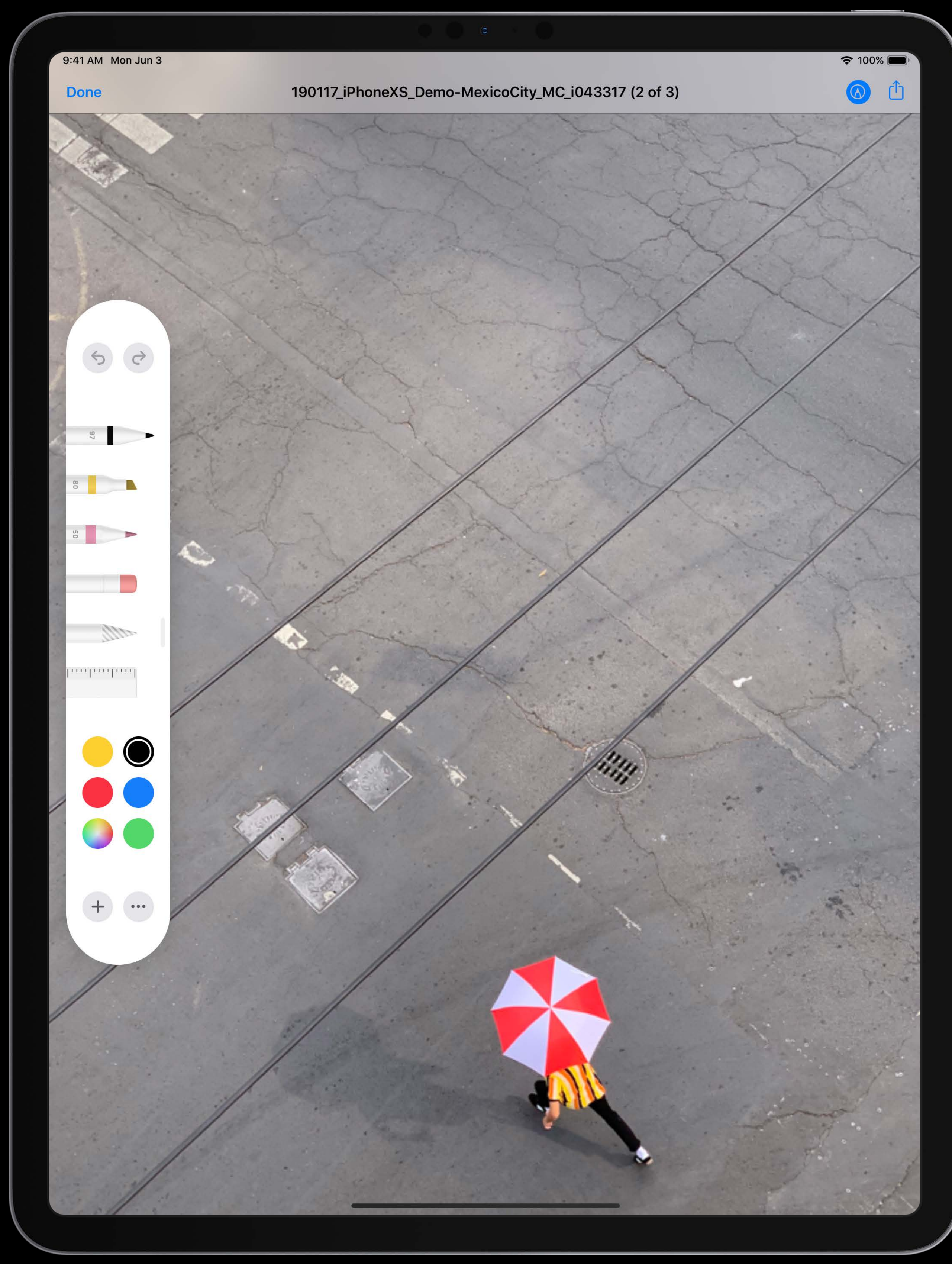


iPhone



# PKToolPicker

## Tool Picker UI — Compact layout



iPad



iPhone



iPhone

Obscured  
Frame



# PKToolPicker

Tool Picker UI — Compact layout

Listen for tool picker's frame changes

```
optional func toolPickerFramesObscuredDidChange(_ toolPicker: PKToolPicker)
```

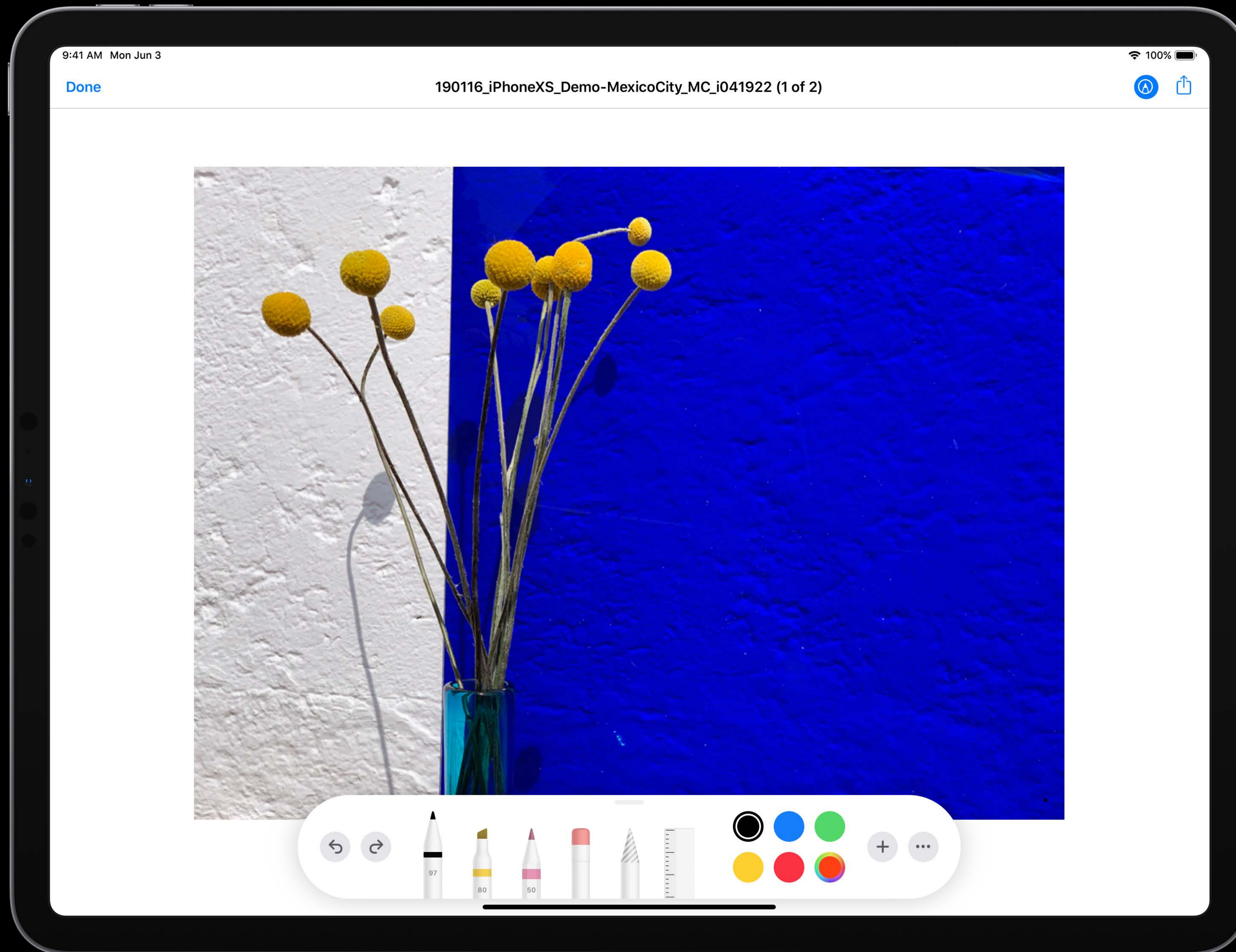
Adjust content accordingly

```
public func frameObscured(in view: UIView) -> CGRect
```



# Tool Picker

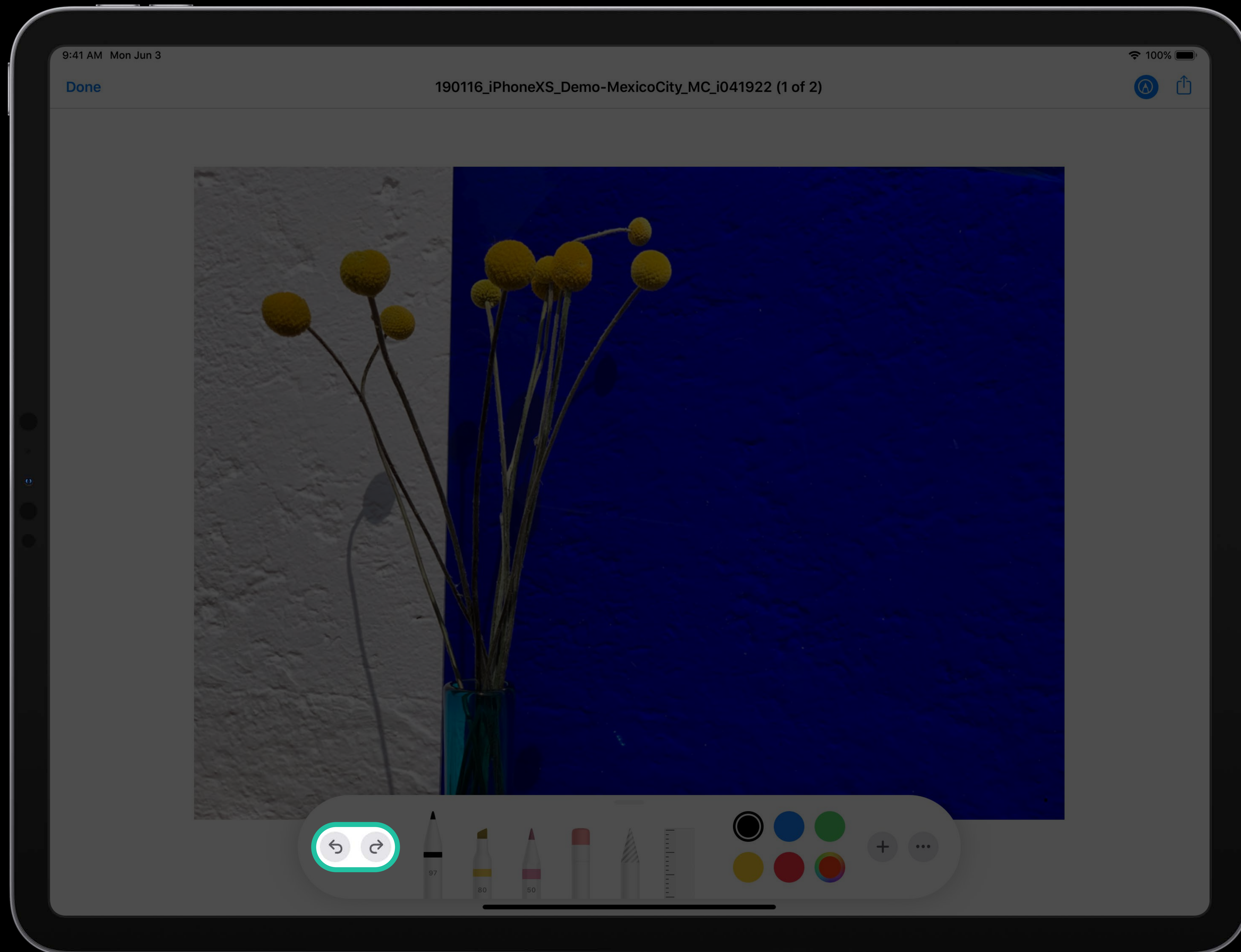
Tool Picker UI — Undo buttons





# Tool Picker

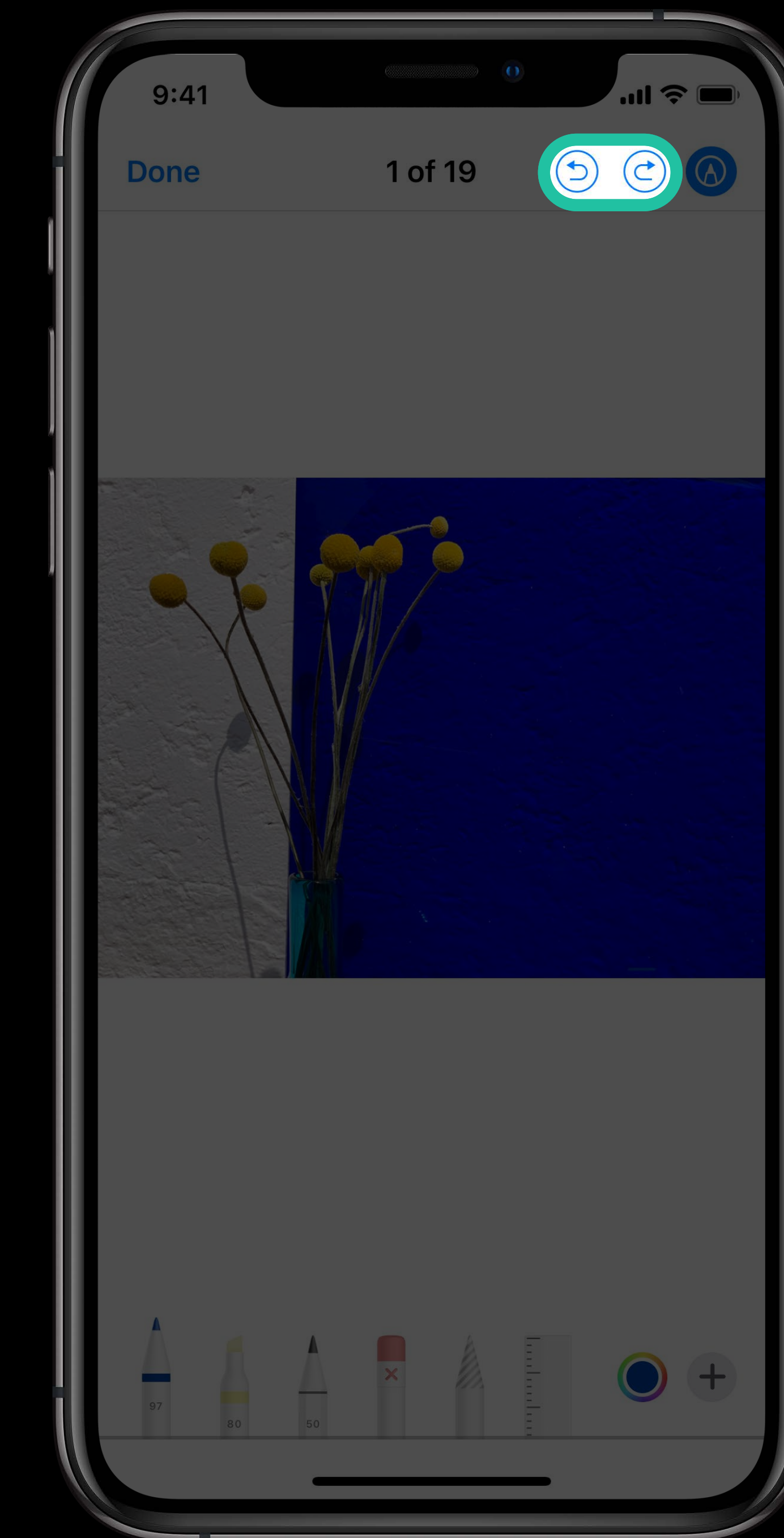
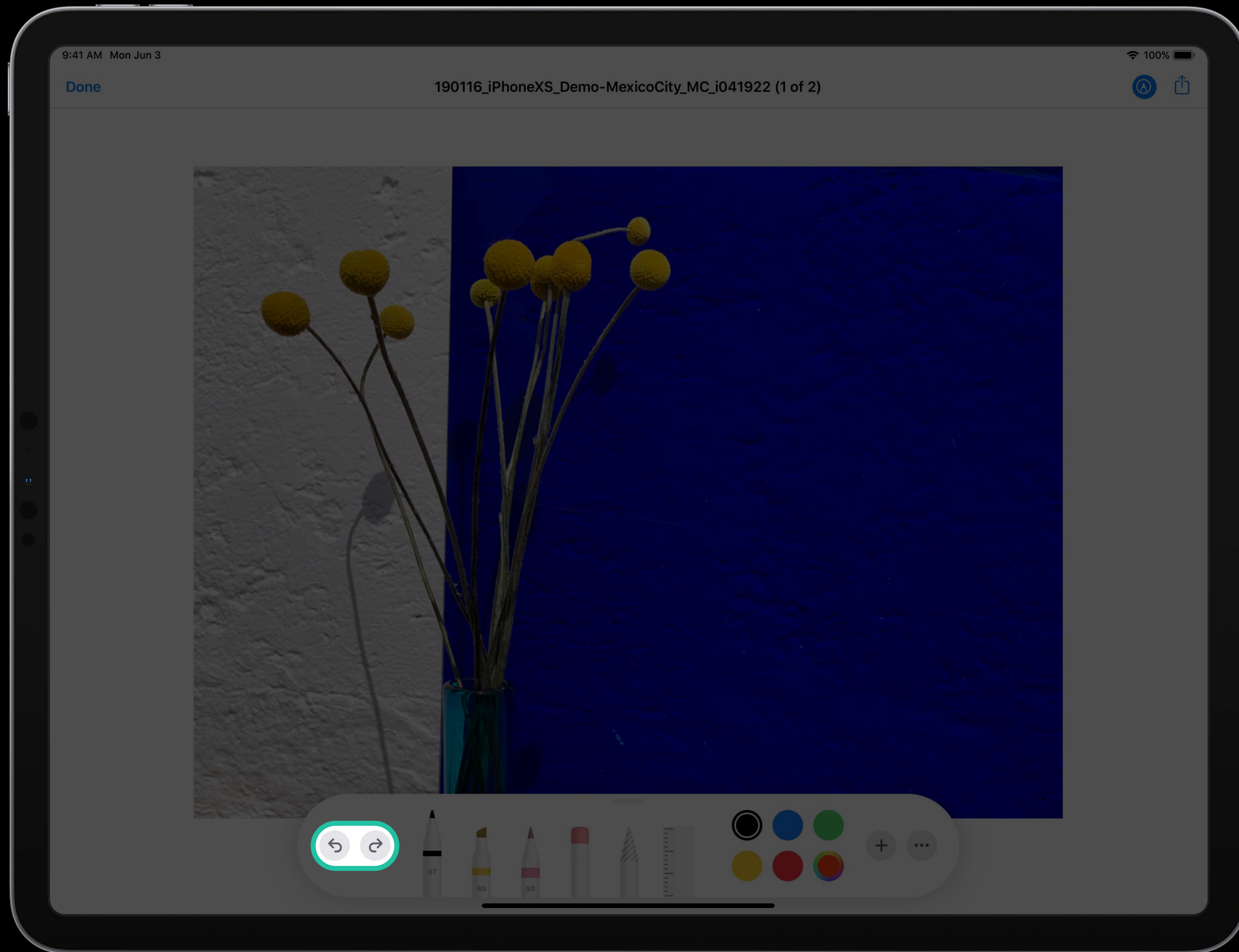
Tool Picker UI — Undo buttons





# Tool Picker

Tool Picker UI — Undo buttons





# Advanced Behaviors



# PKCanvasView

Delegate callbacks

```
graph TD; A[pencil/touch down] --> B[canvasViewDidBeginUsingTool];
```

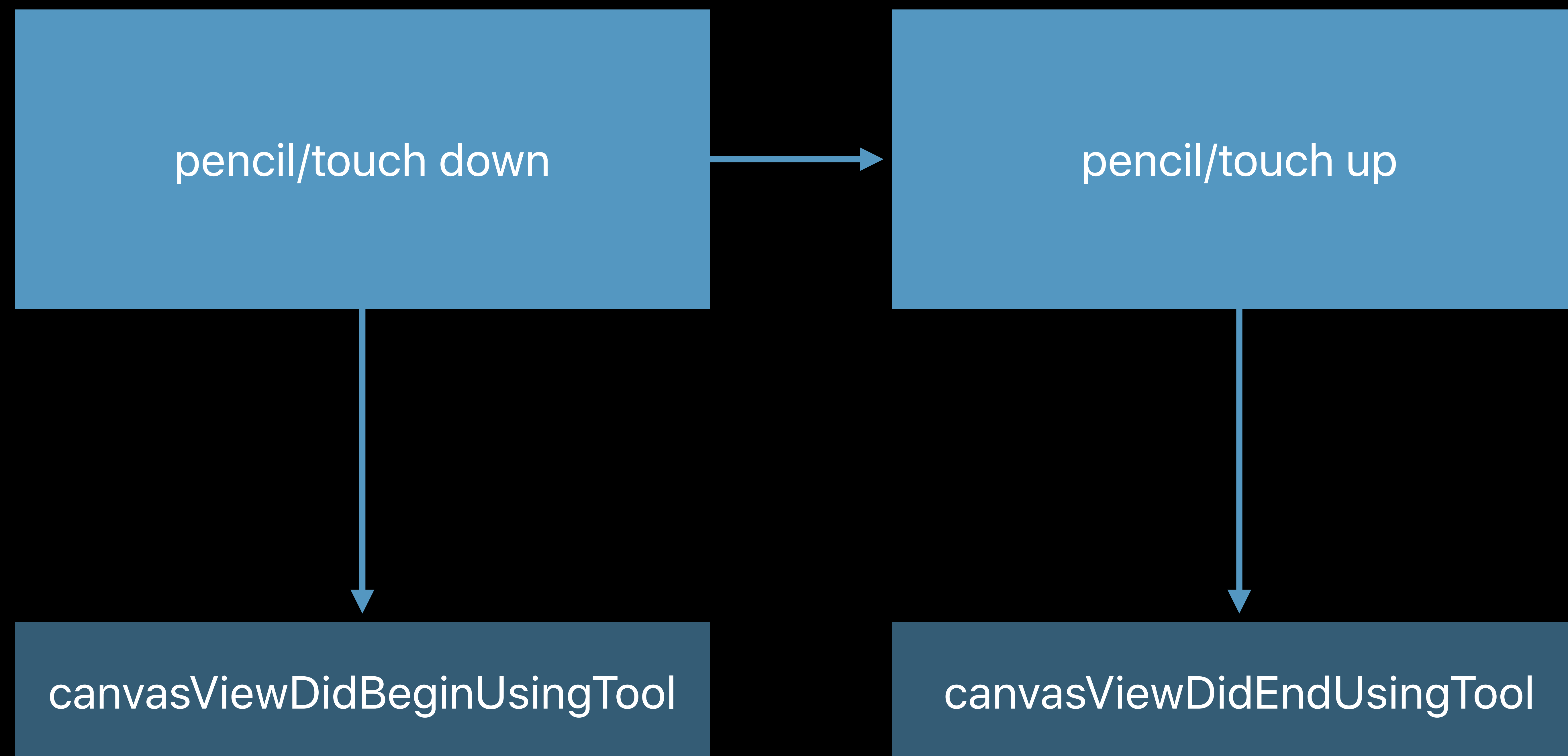
pencil/touch down

canvasViewDidBeginUsingTool



# PKCanvasView

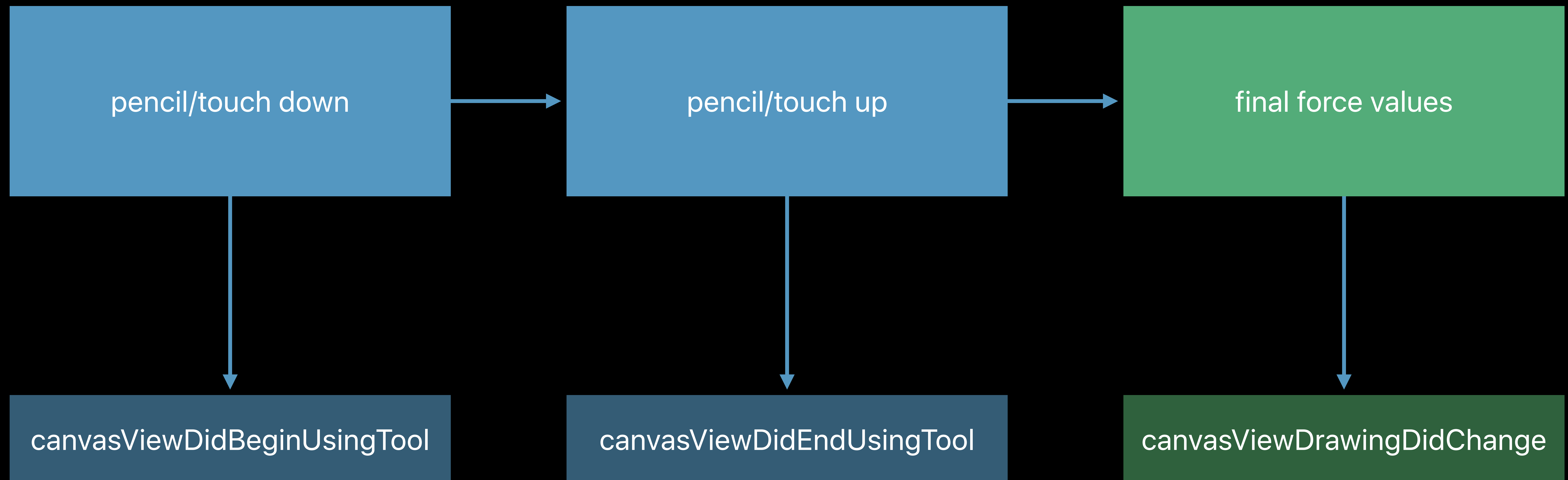
## Delegate callbacks





# PKCanvasView

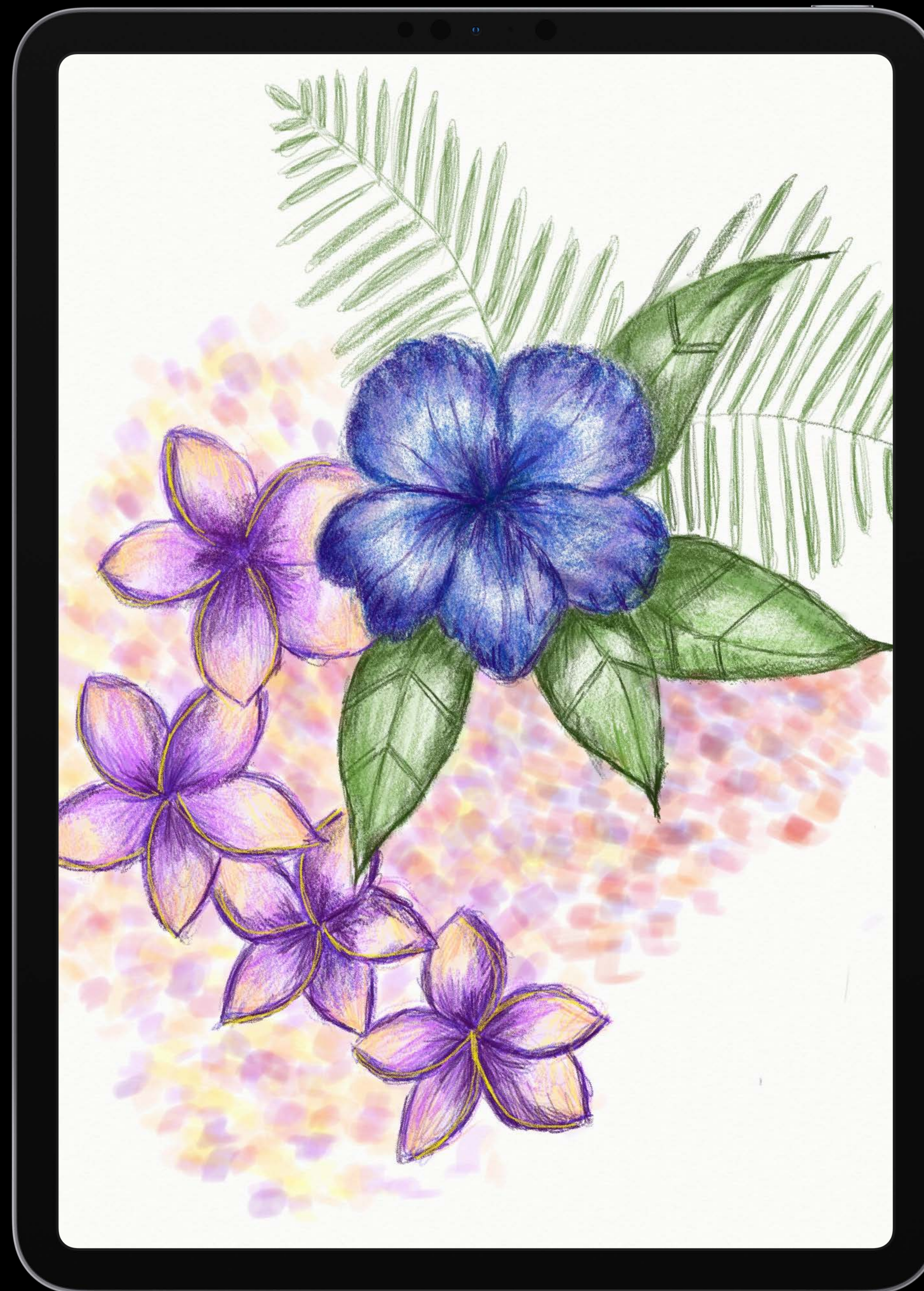
## Delegate callbacks





# PKCanvasView

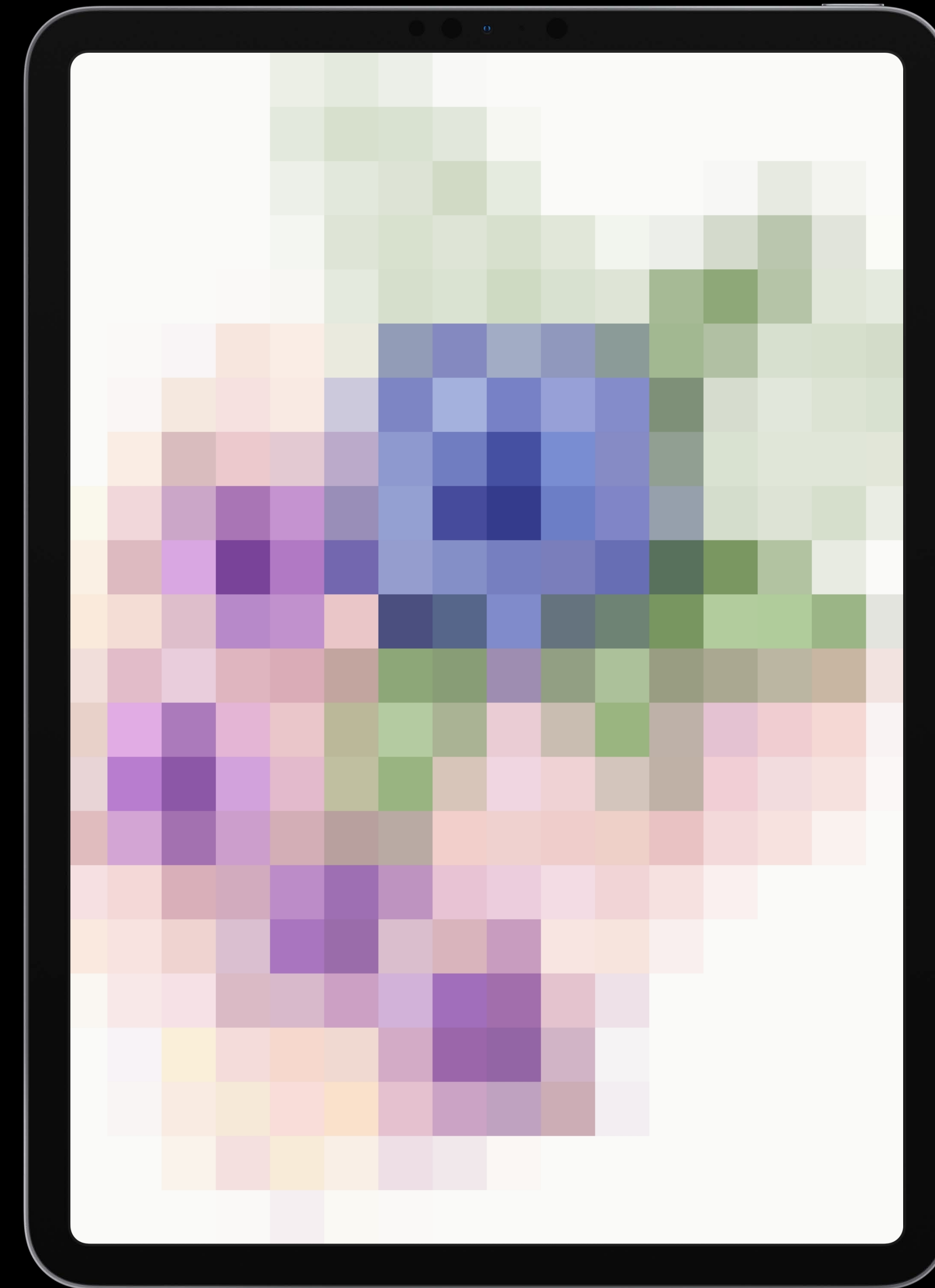
Delegate callbacks





# PKCanvasView

Delegate callbacks



PKCanvasView

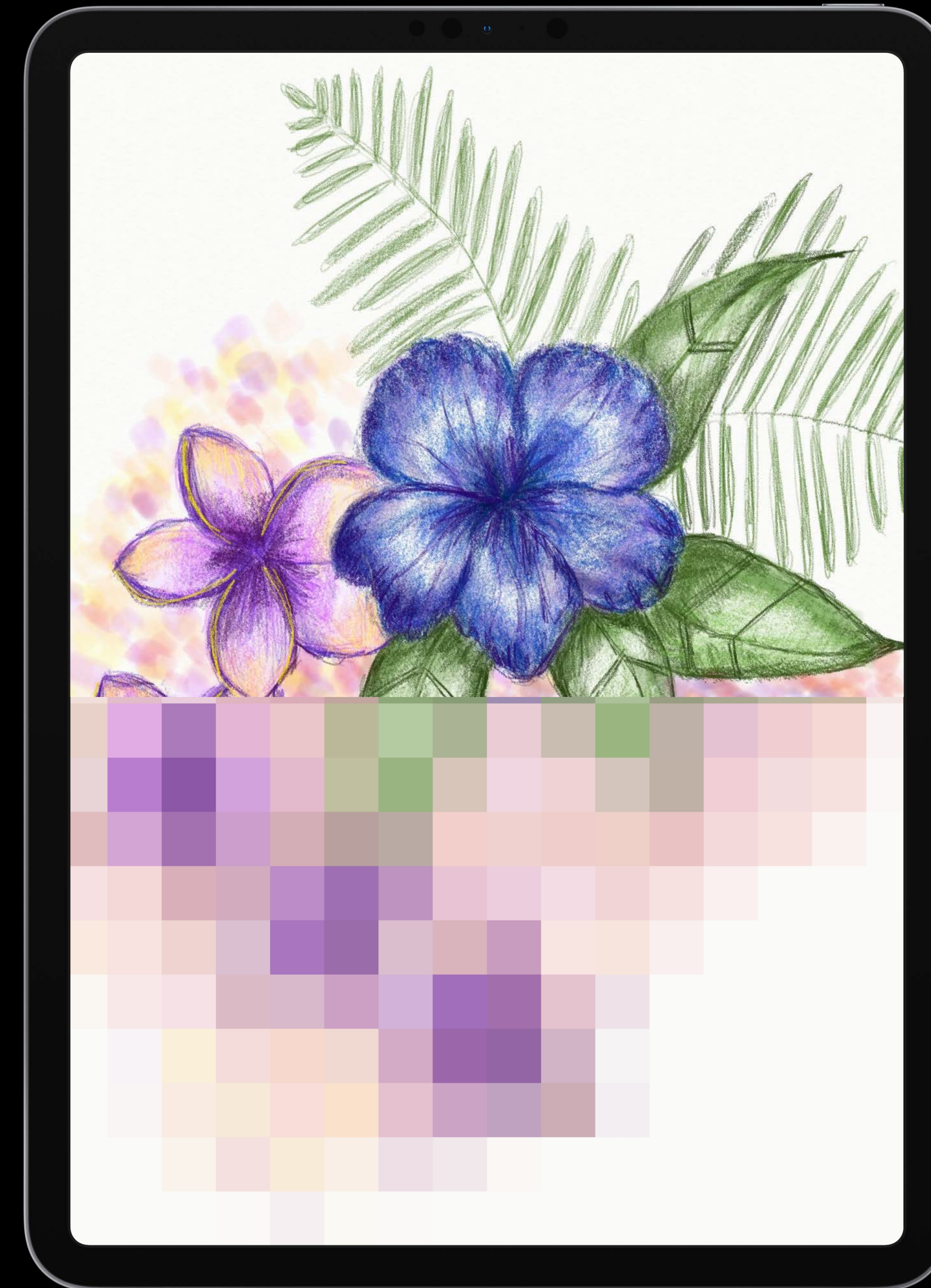
Set drawing





# PKCanvasView

Delegate callbacks



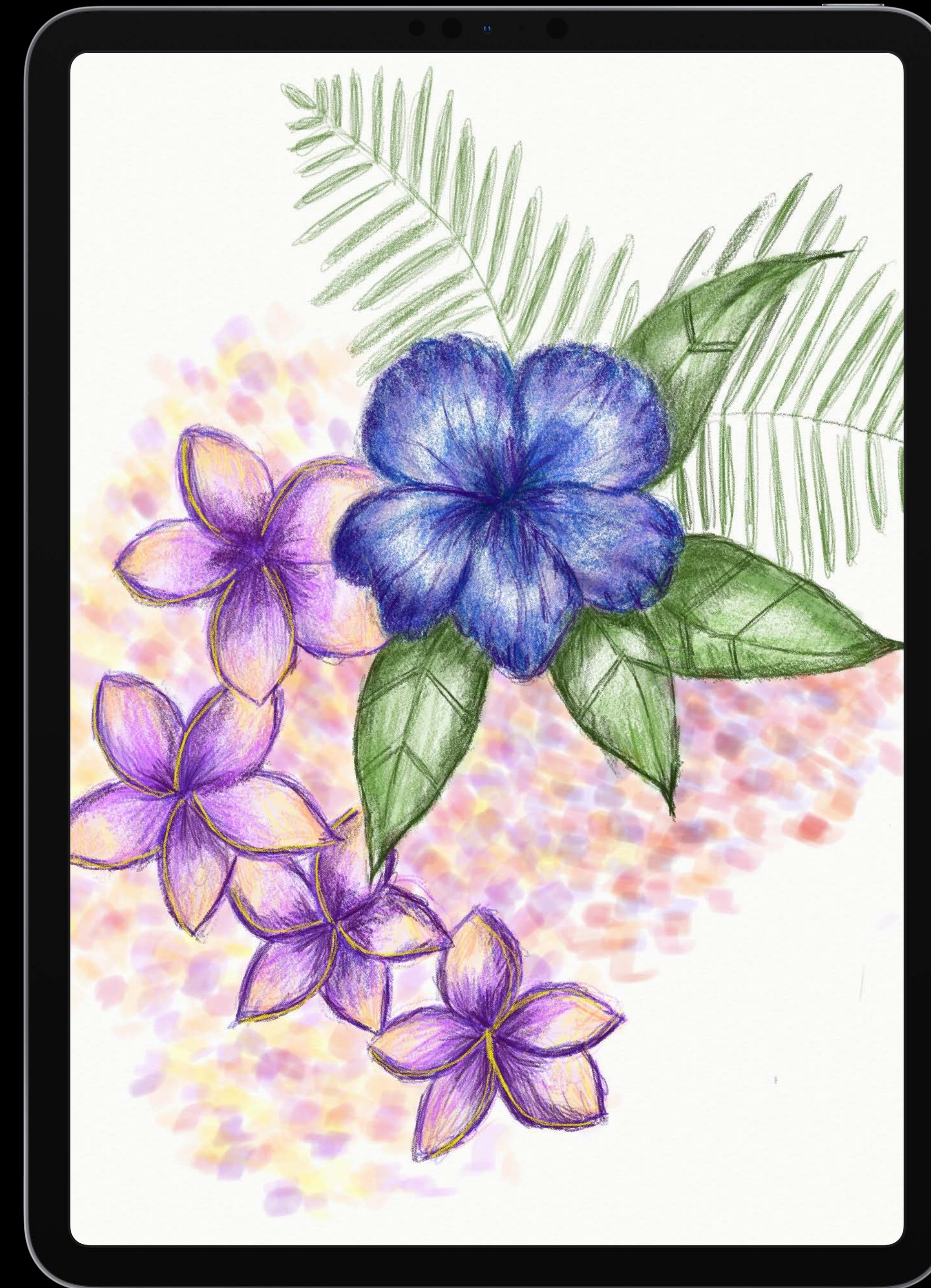
PKCanvasView

Set drawing



# PKCanvasView

Delegate callbacks



PKCanvasView

Set drawing

canvasViewDidFinishRendering













# Scrolling and Touch

## PKCanvasView

```
var allowsFingerDrawing: Bool { get set }
```

allowsFingerDrawing

Drawing

Scrolling

true

1 Finger and Pencil Draw

2 fingers scroll

false

Pencil Draws

1 finger scrolls



# Scrolling and Touch

## PKCanvasView

```
var allowsFingerDrawing: Bool { get set }
```

allowsFingerDrawing

Drawing

Scrolling

true

1 Finger and Pencil Draw

2 fingers scroll

false

Pencil Draws

1 finger scrolls



# Scrolling and Touch

## PKCanvasView

```
var allowsFingerDrawing: Bool { get set }
```

allowsFingerDrawing

Drawing

Scrolling

true

1 Finger and Pencil Draw

2 fingers scroll

false

Pencil Draws

1 finger scrolls



# Scrolling and Touch

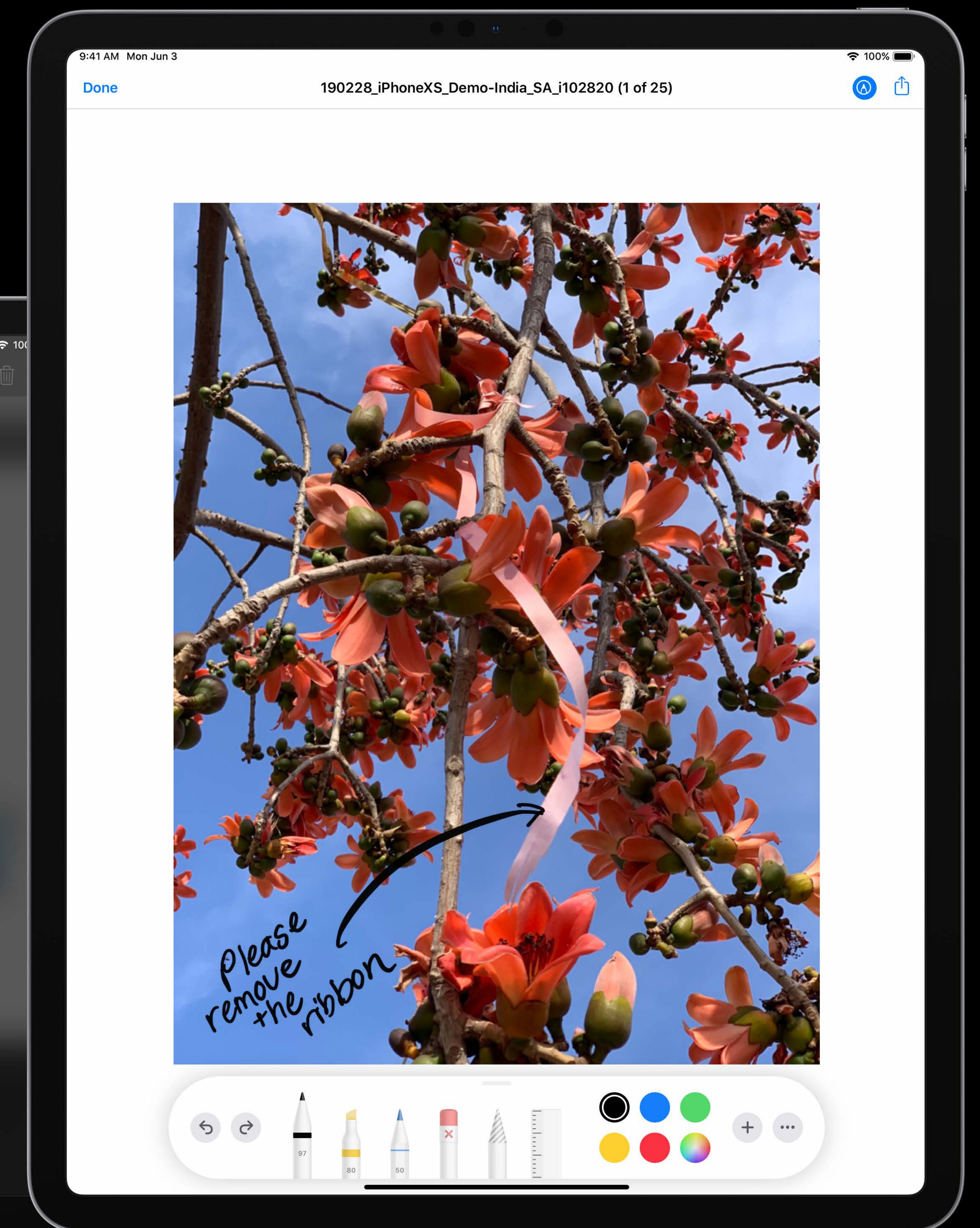
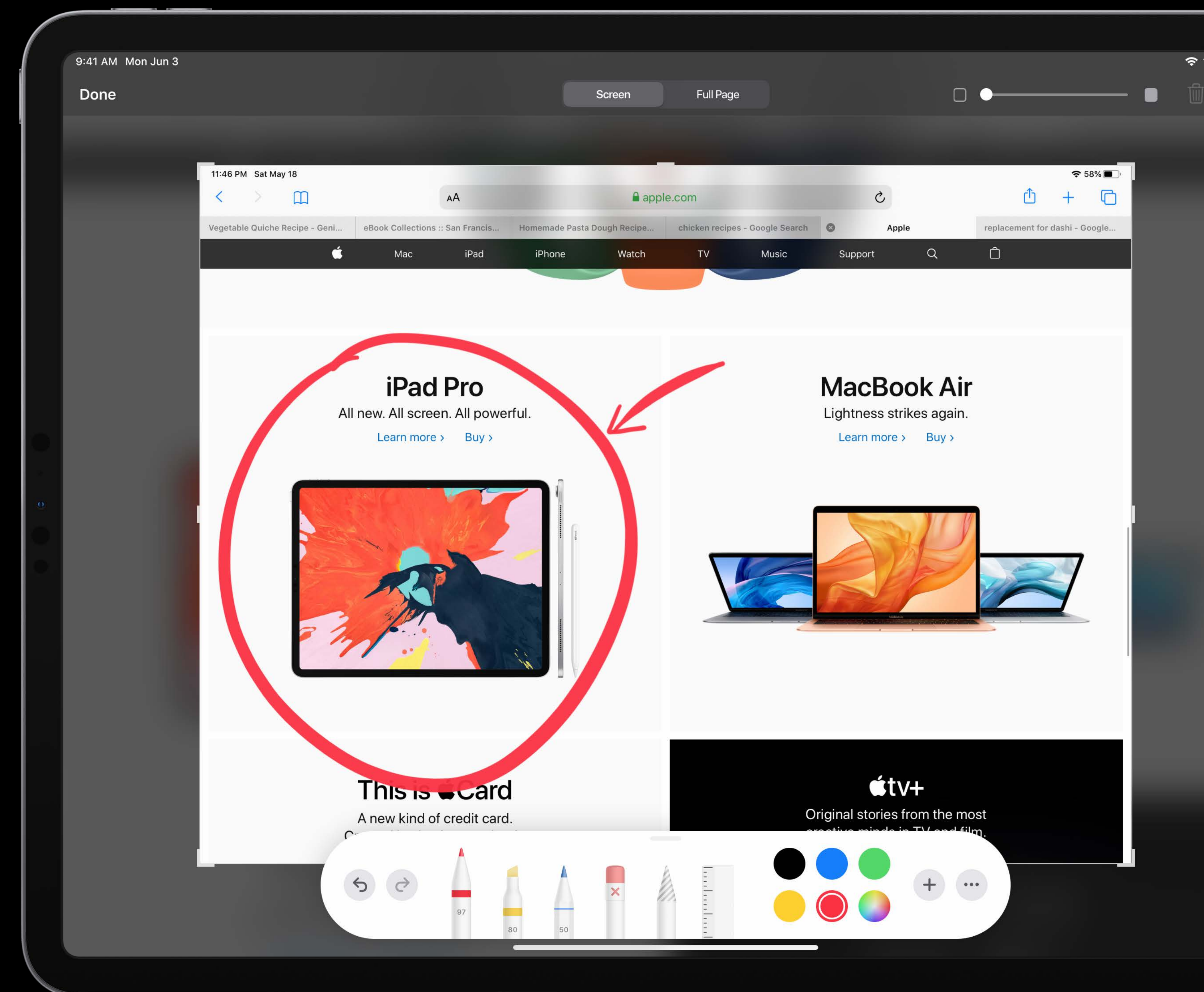
## PKCanvasView

```
var drawingGestureRecognizer: UIGestureRecognizer { get }
```



# Drawing Over Content

```
canvasView.opaque = false  
canvasView.backgroundColor = .clear
```



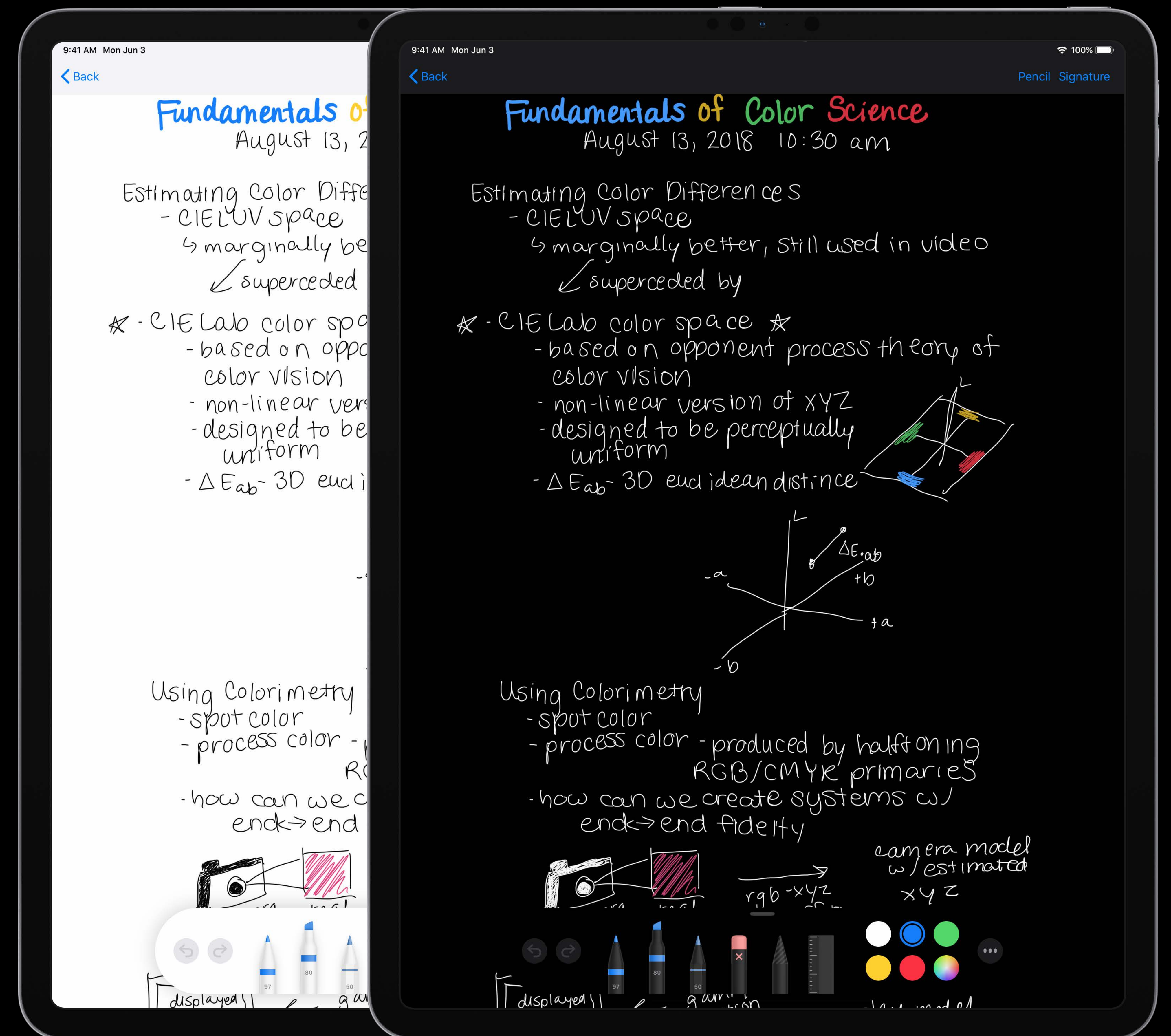






# Dark Mode

```
canvasView.overrideUserInterfaceStyle = .light
```





# Markup Everywhere

## Screenshots





Mac

iPad

iPhone

Watch

TV

Music

Support



## iPhone XR from \$19.99/mo. or \$479.

Two great ways to buy. Just trade in your current iPhone online or at an Apple Store.\*

[Buy >](#) [Learn more >](#)

## iPhone X<sup>®</sup>

All-screen design. Longest battery life ever in an iPhone.  
Fastest performance. Studio-quality photos.

[Learn more >](#) [Buy >](#)





## iPhone XR from \$19.99/mo. or \$479.

Two great ways to buy. Just trade in your current iPhone online or at an Apple Store.\*

[Buy >](#) [Learn more >](#)

## iPhone X<sup>®</sup>

All-screen design. Longest battery life ever in an iPhone.  
Fastest performance. Studio-quality photos.

[Learn more >](#) [Buy >](#)





# Fundamentals of Color Science

August 13, 2018 10:30 am

Estimating Color Differences

- CIE LUV space

↳ marginally better, still used in video

↳ superceded by

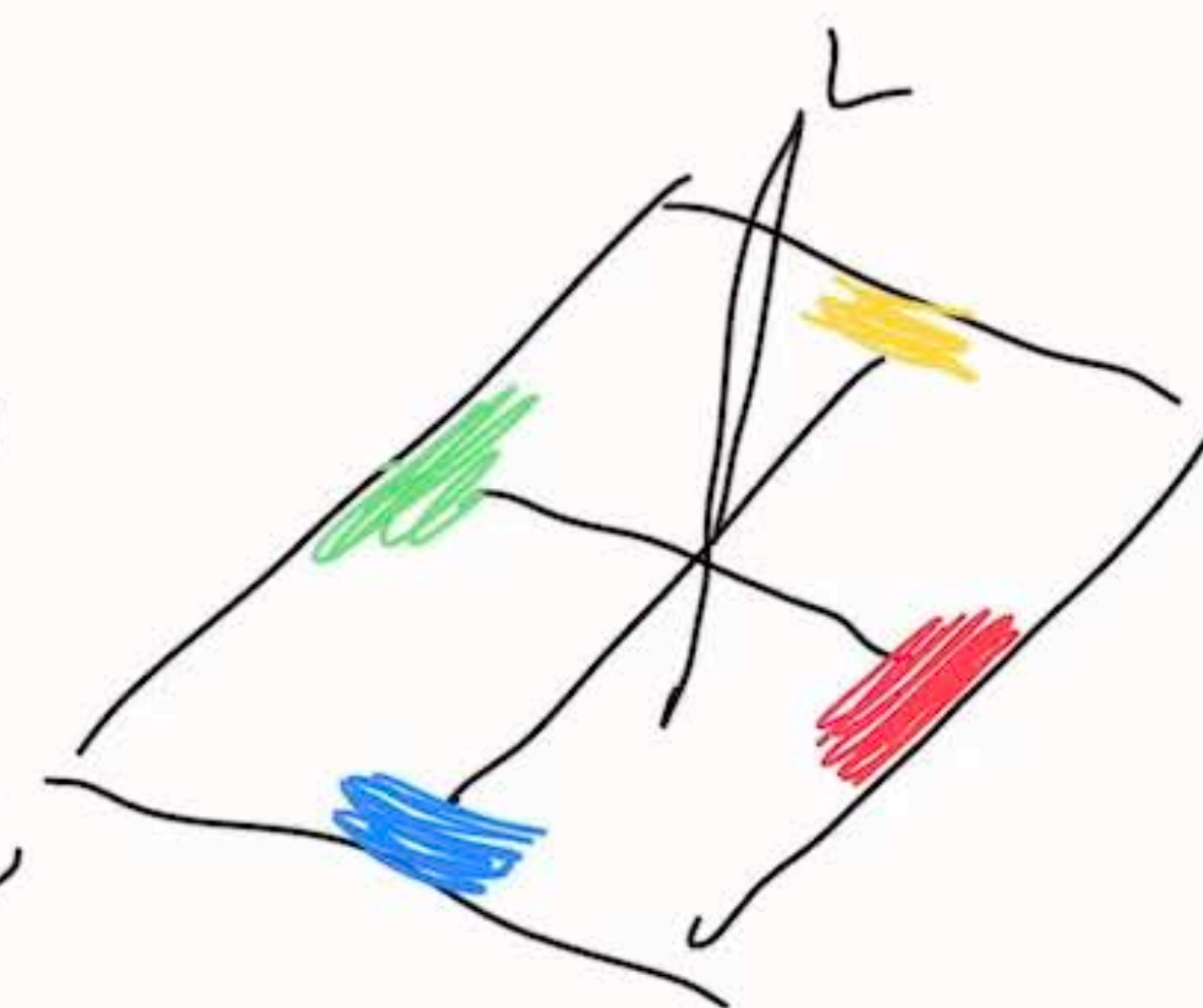
★ - CIE Lab color space ★

- based on opponent process theory of color vision

- non-linear version of XYZ

- designed to be perceptually uniform

-  $\Delta E_{ab}$  - 3D euclidean distance





# Fundamentals of Color Science

August 13, 2018 10:30 am

Estimating Color Differences

- CIE LUV space

↳ marginally better, still used in video

↳ superceded by

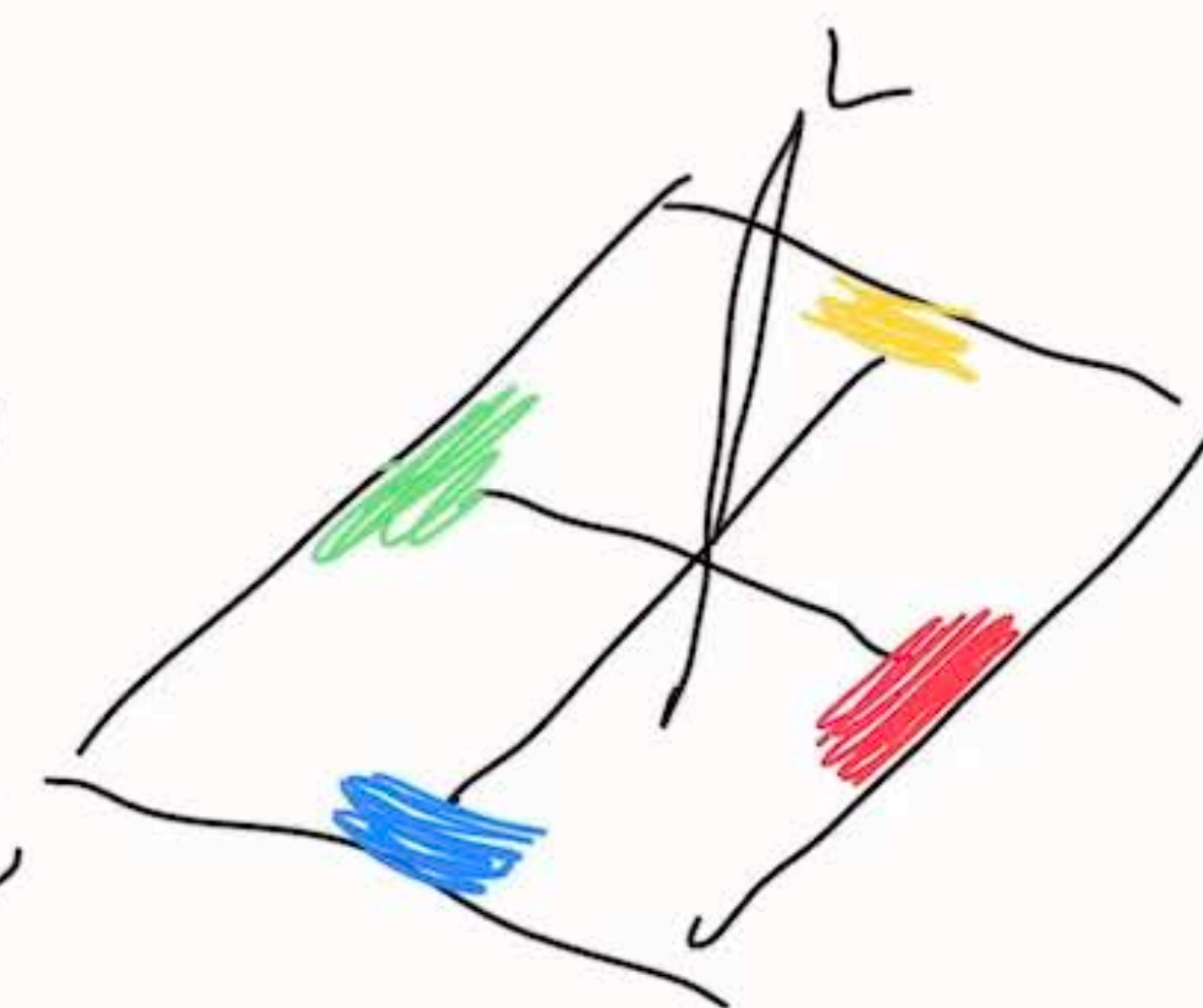
★ - CIE Lab color space ★

- based on opponent process theory of color vision

- non-linear version of XYZ

- designed to be perceptually uniform

-  $\Delta E_{ab}$  - 3D euclidean distance





9:41 AM Mon Jun 3

### Apple Park

Corporate Headquarters · 36 mi

**Directions**  
44 min drive



Photos on Wikipedia

[More Photos on WIKIPEDIA](#)



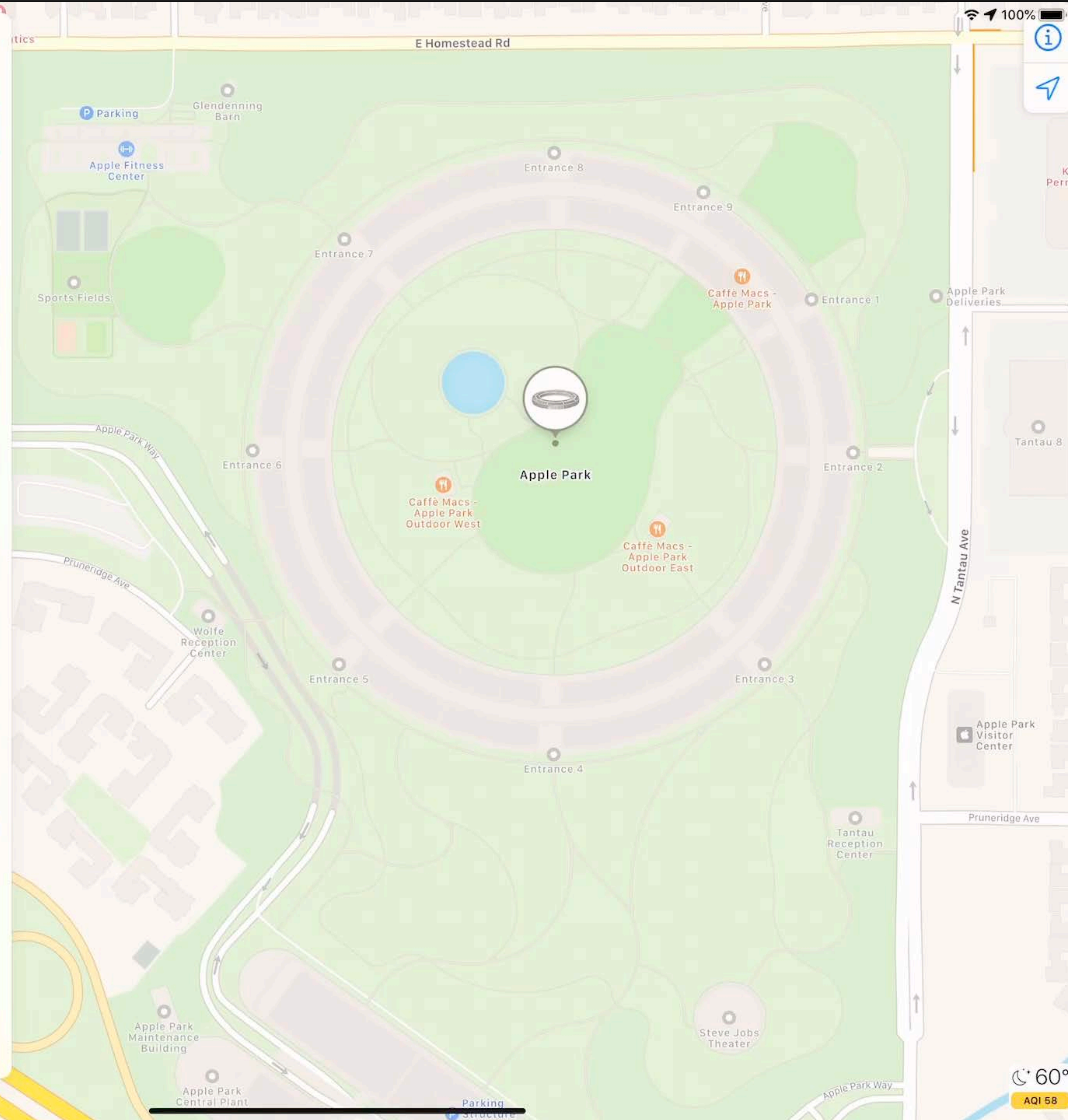
Address  
Apple Inc.  
One Apple Park Way  
Cupertino, CA 95014  
United States

Phone  
+1 (408) 996-1010

Website  
[apple.com](http://apple.com)

Wikipedia [Open Wikipedia](#)

Apple Park is the corporate headquarters of Apple Inc., located at 1 Apple Park Way in Cupertino, California, United States. It opened to employees in April 2017, while construction was still underway. Its research and development facilities are occupied with over 2,000 people. It replaced the original headquarters at 1 Infinite Loop, which opened in 1993.



100%

60°

AQI 58

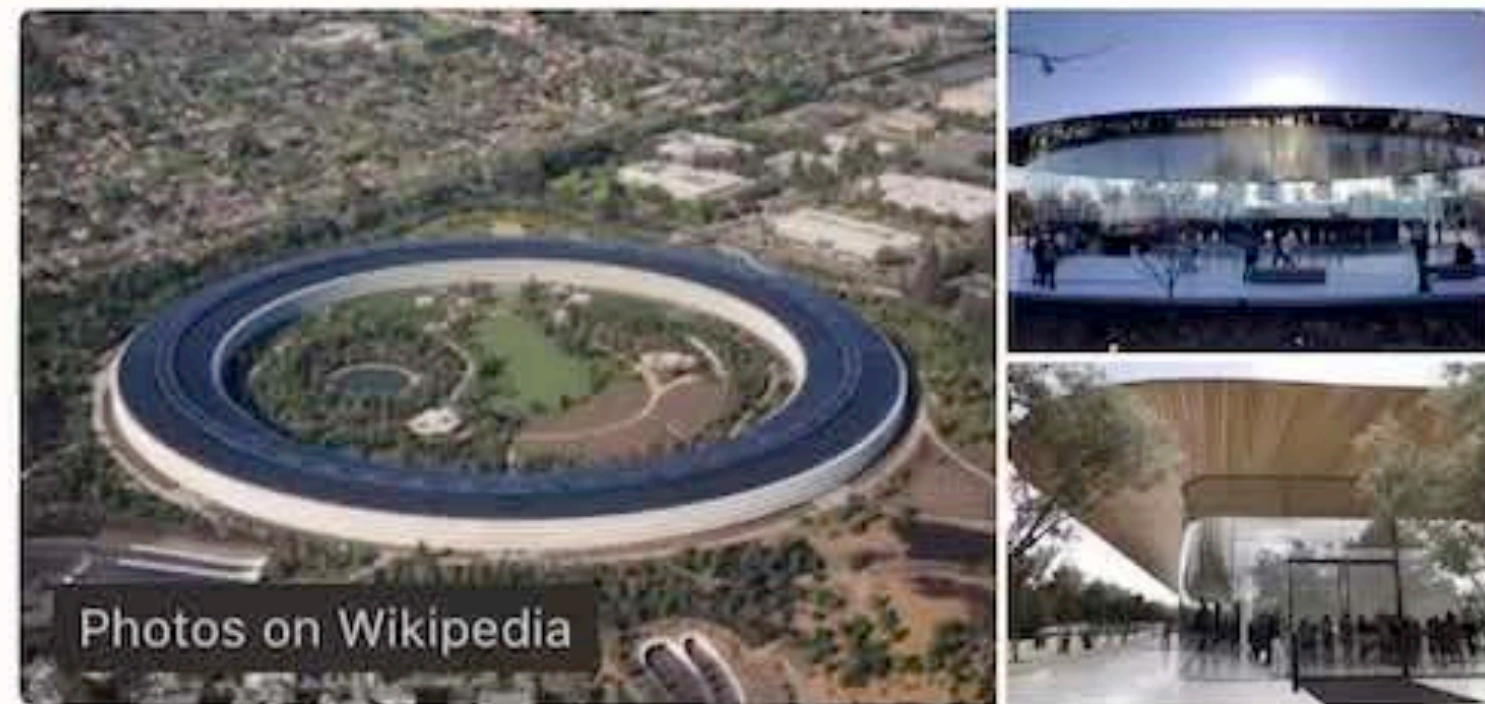


9:41 AM Mon Jun 3

### Apple Park

Corporate Headquarters · 36 mi

**Directions**  
44 min drive



Photos on Wikipedia

[More Photos on WIKIPEDIA](#)



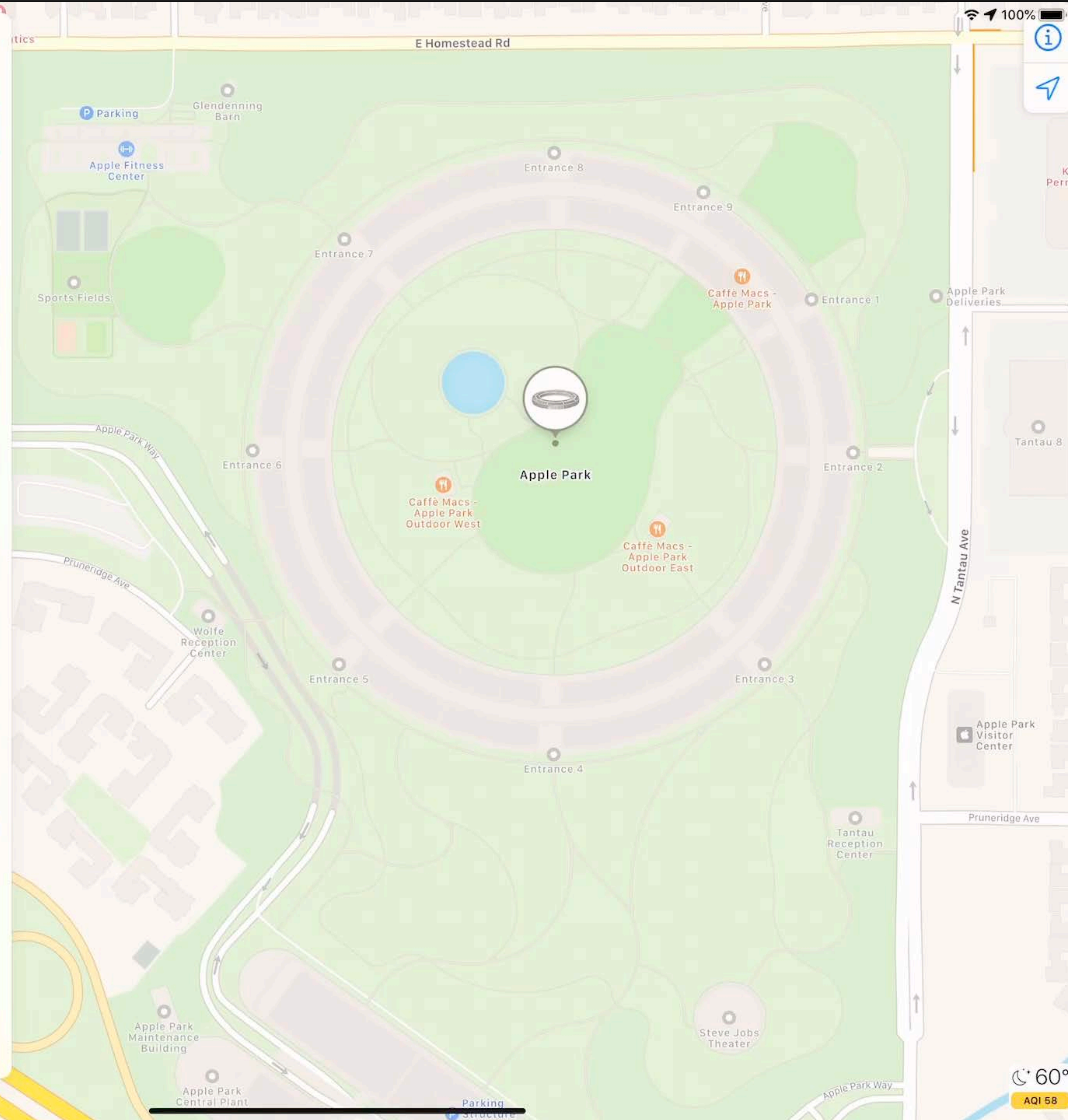
Address  
Apple Inc.  
One Apple Park Way  
Cupertino, CA 95014  
United States

Phone  
+1 (408) 996-1010

Website  
[apple.com](http://apple.com)

Wikipedia [Open Wikipedia](#)

Apple Park is the corporate headquarters of Apple Inc., located at 1 Apple Park Way in Cupertino, California, United States. It opened to employees in April 2017, while construction was still underway. Its research and development facilities are occupied with over 2,000 people. It replaced the original headquarters at 1 Infinite Loop, which opened in 1993.





# Screenshots

## UIScreenshotService on UIWindowScene

```
class DrawingViewController: UIScreenshotServiceDelegate {  
    override func viewWillAppear() {  
        super.viewWillAppear()  
        view.window.windowScene.screenshotService.delegate = self  
    }  
}
```



# Screenshots

Full content delivered as PDF data

```
func screenshotService(  
    UIScreenShotService,  
    generatePDFRepresentationWithCompletion: (Data?, Int, CGRect) -> Void)
```



# Screenshots

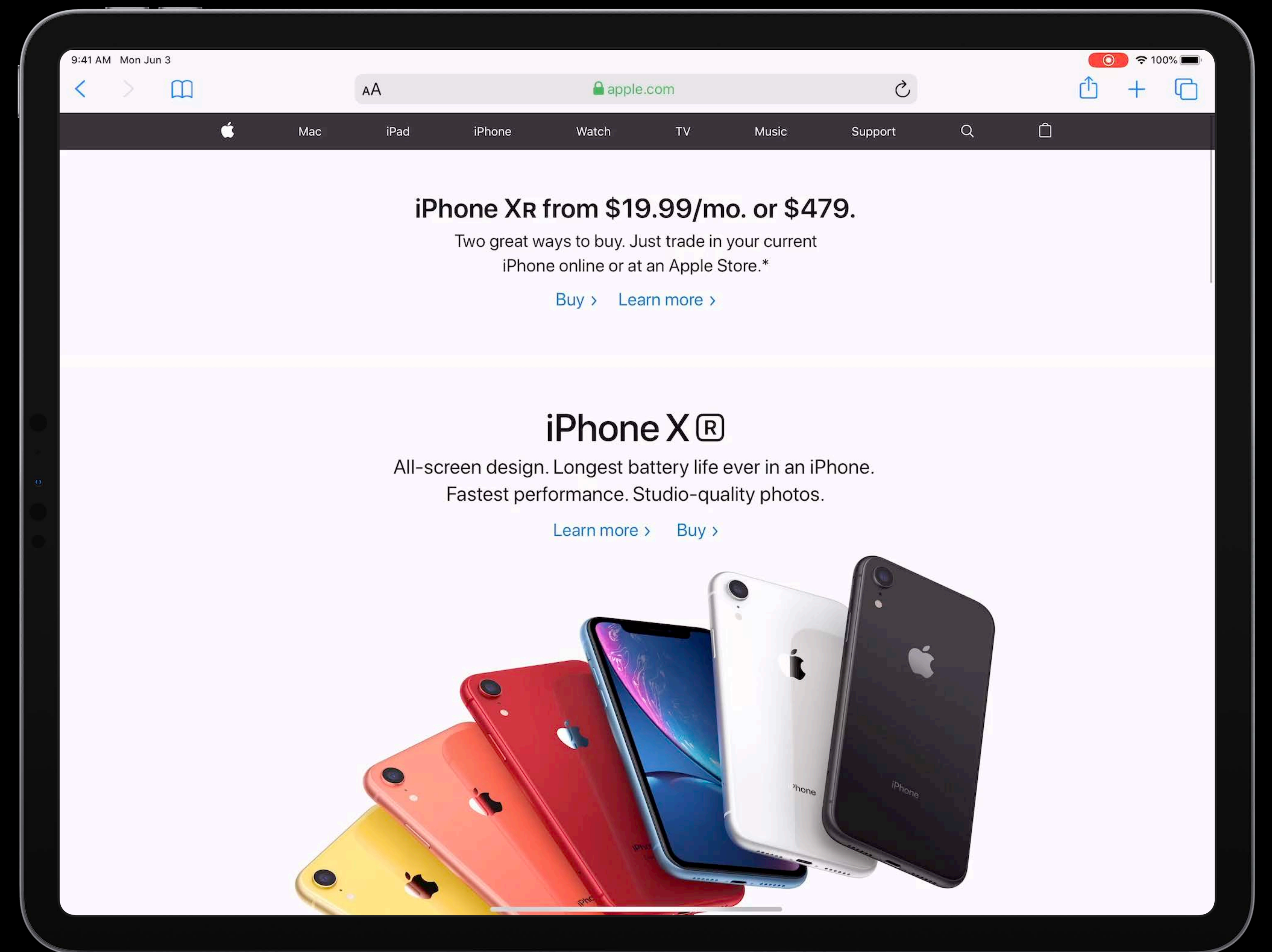
`indexOfCurrentPage`





# Screenshots

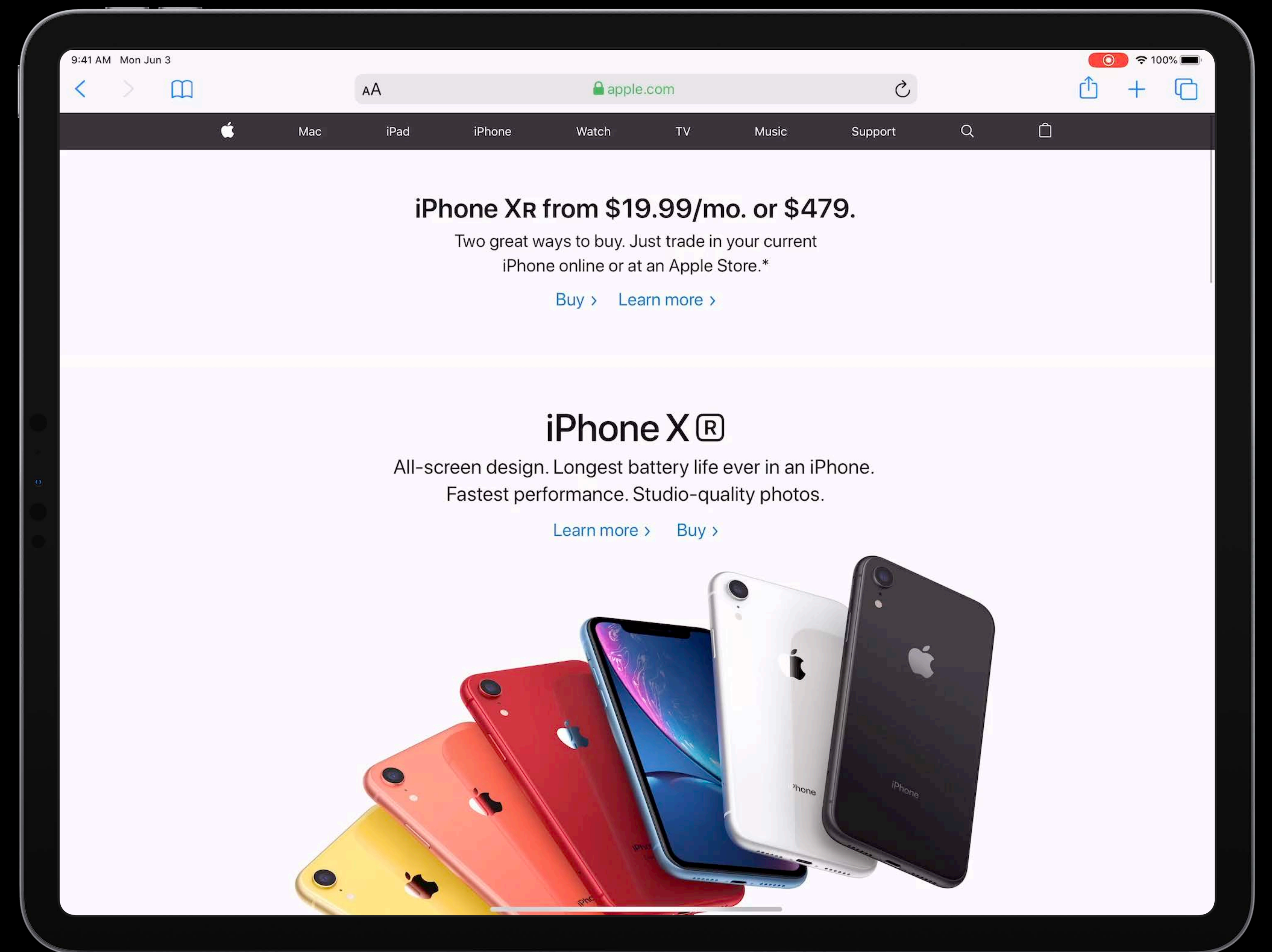
rectInCurrentPage





# Screenshots

rectInCurrentPage





# Screenshots

`rectInCurrentPage`

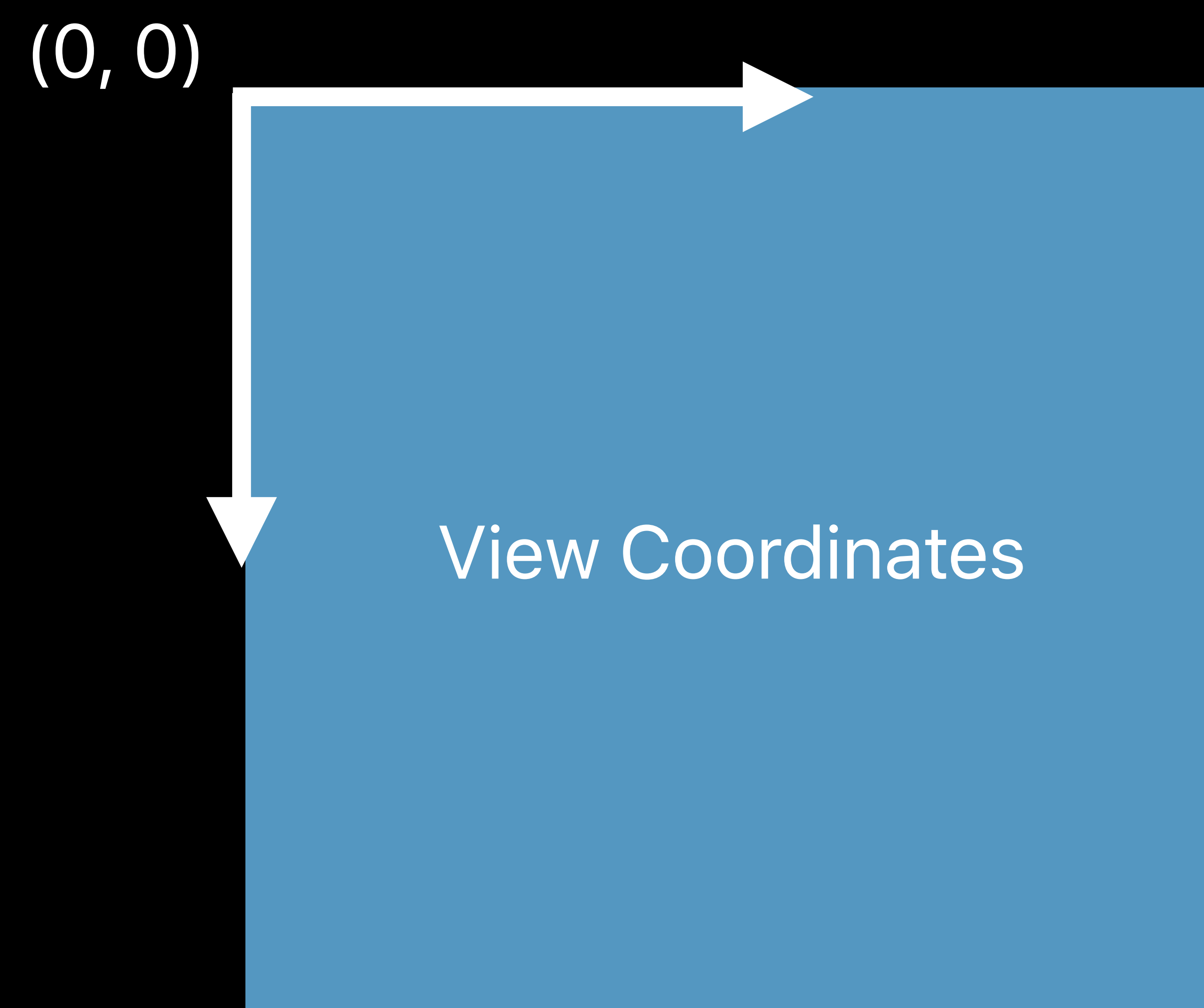
Expected in PDF coordinates



# Screenshots

`rectInCurrentPage`

Expected in PDF coordinates

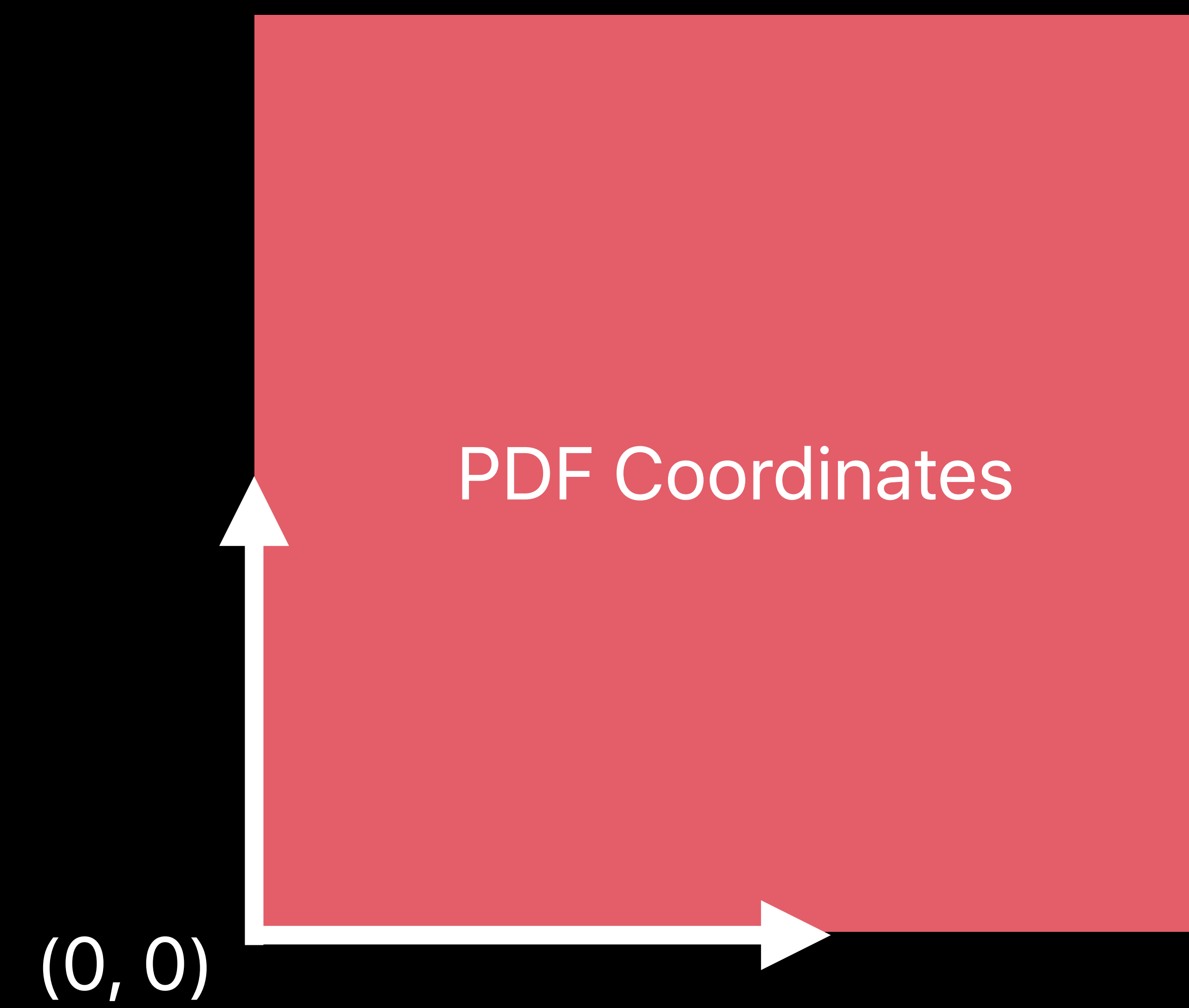
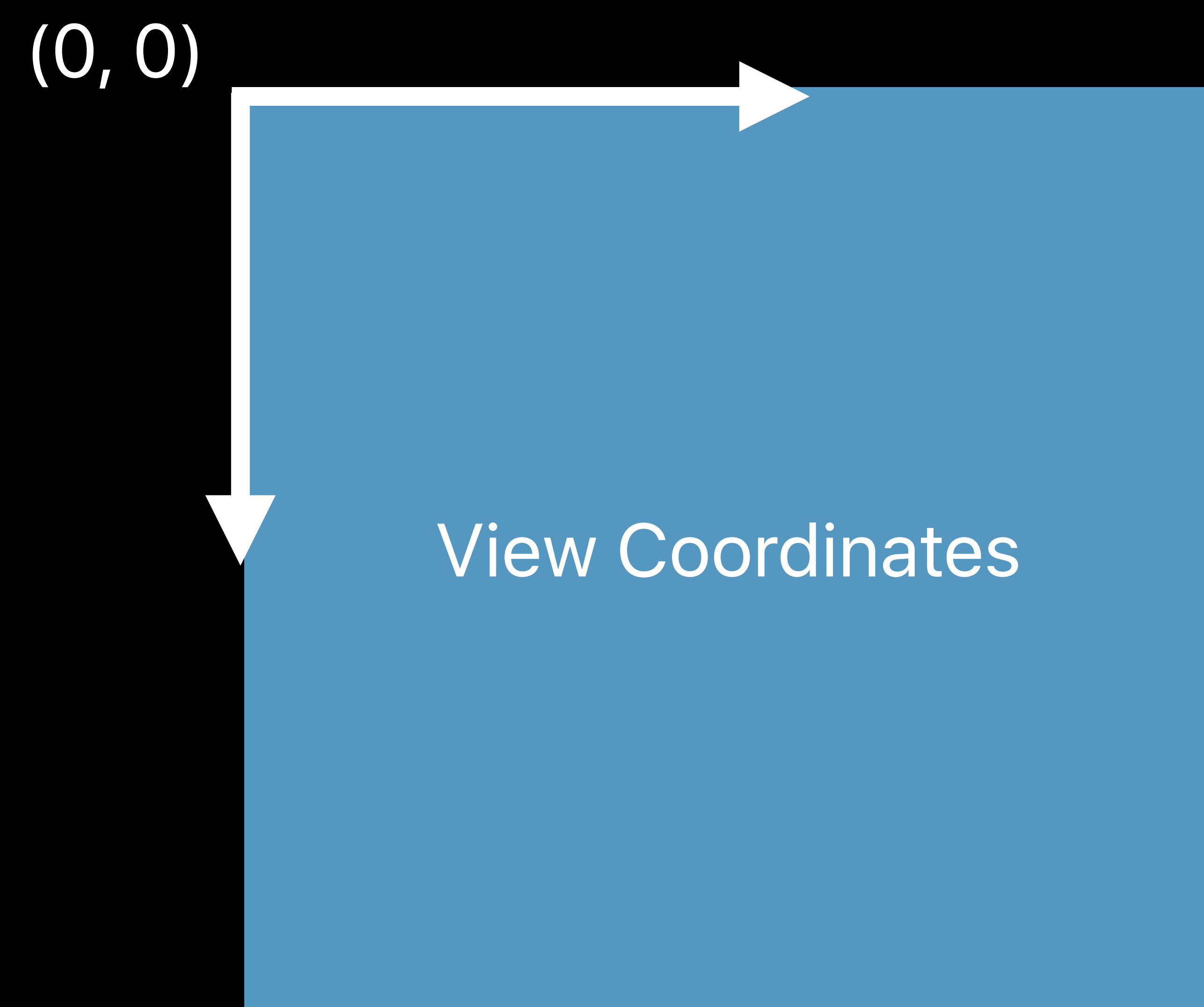




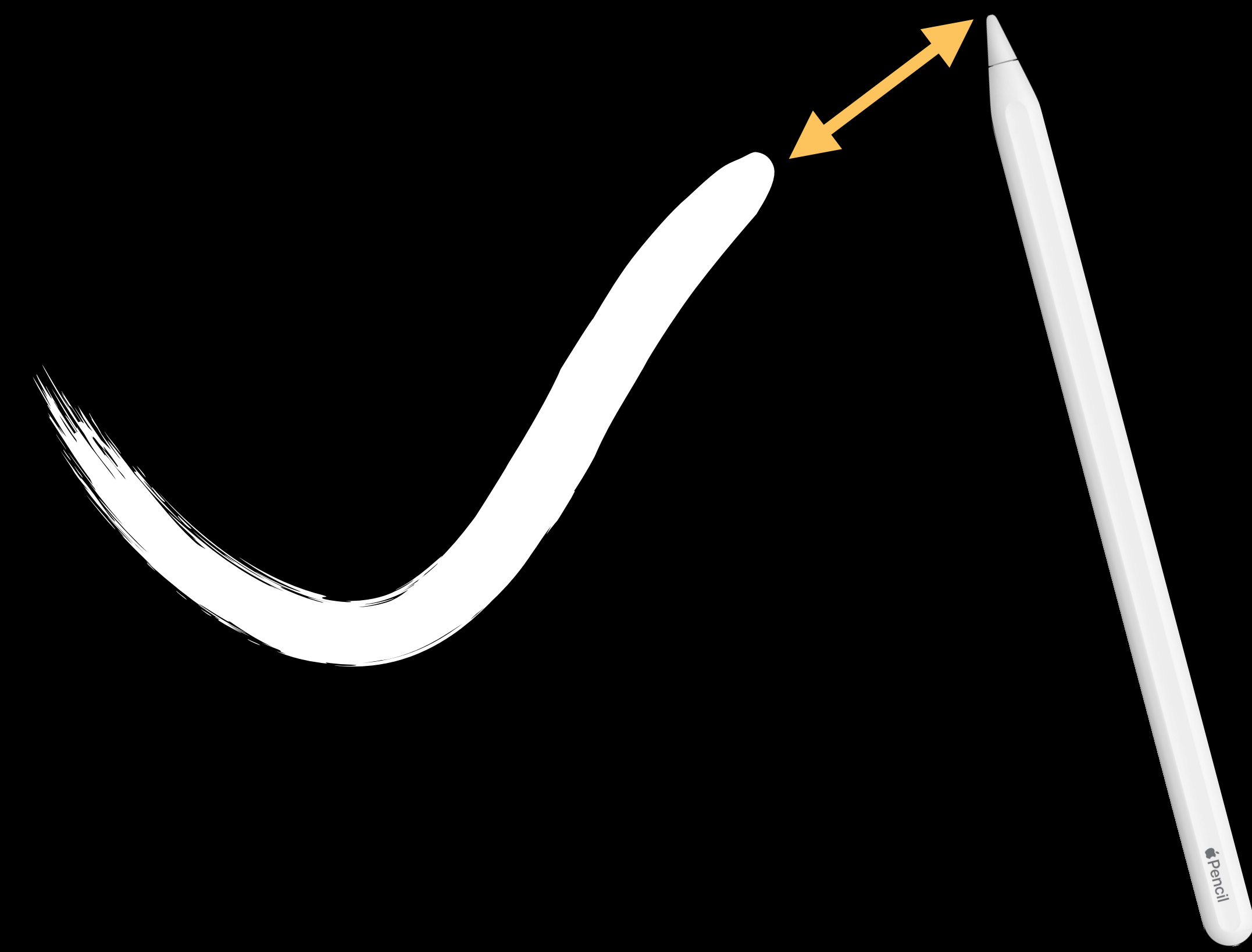
# Screenshots

`rectInCurrentPage`

Expected in PDF coordinates

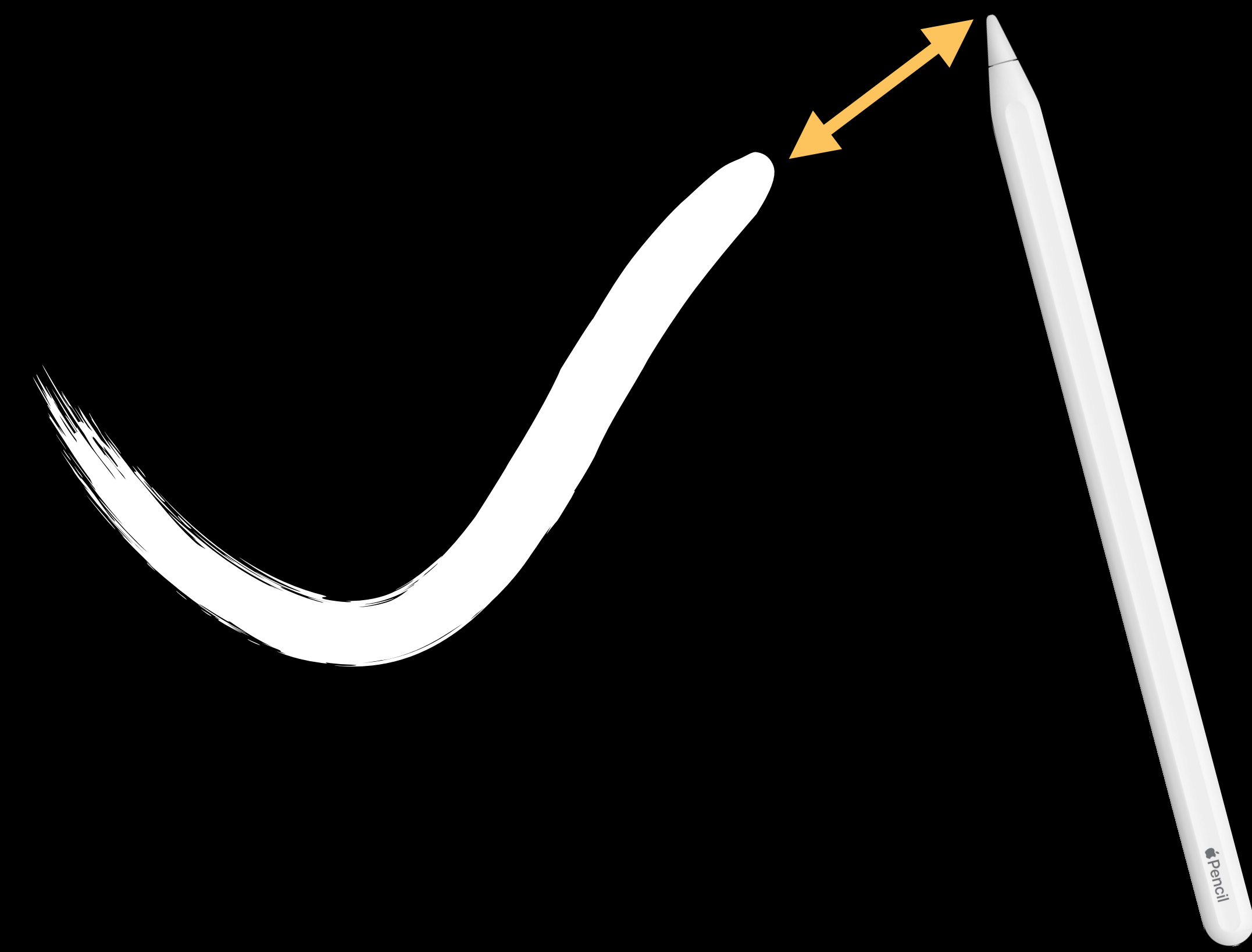






Build Your Own Engine



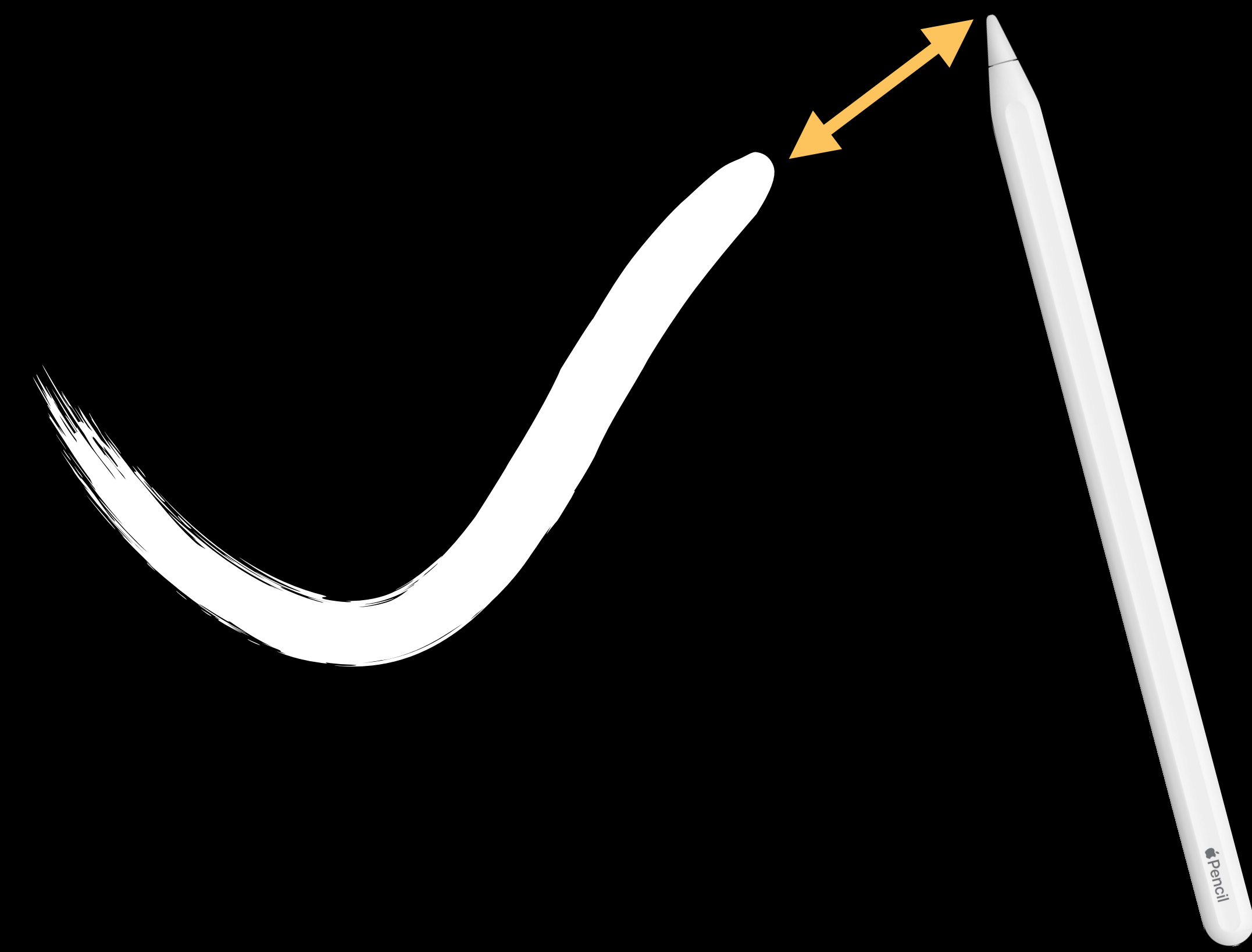


Build Your Own Engine



PencilKit

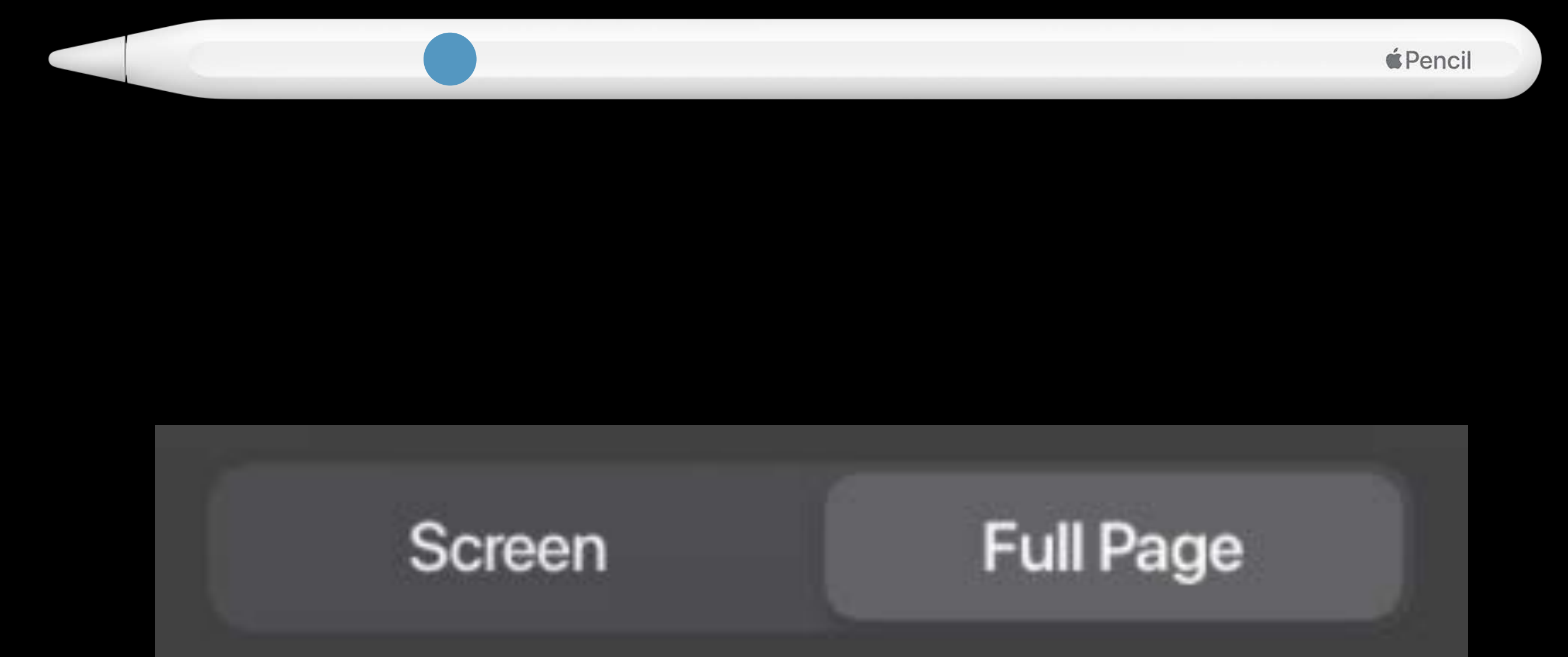




Build Your Own Engine



PencilKit



Other Great Pencil APIs



# More Information

[developer.apple.com/wwdc19/221](https://developer.apple.com/wwdc19/221)



 WWDC19