Advanced Scrollviews and Touch Handling Techniques

Session 235

Josh Shaffer
iOS Apps and Frameworks Engineer

Eliza Block
iOS Apps and Frameworks Engineer

Scrollview Touch Handling History

Scrollview Touch Handling History

2008	iPhone OS 2.0	UITouch

Scrollview Touch Handling History

2008	iPhone OS 2.0	UITouch
2009	iPhone OS 3.0	Nested scroll views

Scrollview Touch Handling History

2008	iPhone OS 2.0	UITouch
2009	iPhone OS 3.0	Nested scroll views
2009	iPhone OS 3.2	UIGestureRecognizer

Scrollview Touch Handling History

2008	iPhone OS 2.0	UITouch
2009	iPhone OS 3.0	Nested scroll views
2009	iPhone OS 3.2	UIGestureRecognizer
2010	iOS 4.0	Retina display support

Scrollview Touch Handling History

2008	iPhone OS 2.0	UITouch
2009	iPhone OS 3.0	Nested scroll views
2009	iPhone OS 3.2	UIGestureRecognizer
2010	iOS 4.0	Retina display support
2011	iOS 5.0	Gesture recognizer properties

Scrollview Touch Handling History

2008	iPhone OS 2.0	UITouch
2009	iPhone OS 3.0	Nested scroll views
2009	iPhone OS 3.2	UIGestureRecognizer
2010	iOS 4.0	Retina display support
2011	iOS 5.0	Gesture recognizer properties
2012	iOS 6.0	Resting touches

Scrollview Touch Handling History

2008	iPhone OS 2.0	UITouch
2009	iPhone OS 3.0	Nested scroll views
2009	iPhone OS 3.2	UIGestureRecognizer
2010	iOS 4.0	Retina display support
2011	iOS 5.0	Gesture recognizer properties
2012	iOS 6.0	Resting touches
2013	iOS 7.0	Keyboard dismissal

Hit Testing

Hit Testing

Gesture Recognizers

Hit Testing

Gesture Recognizers

Touch Delivery

Transparent Overlays

Transparent Overlays

Dragging While Scrolling

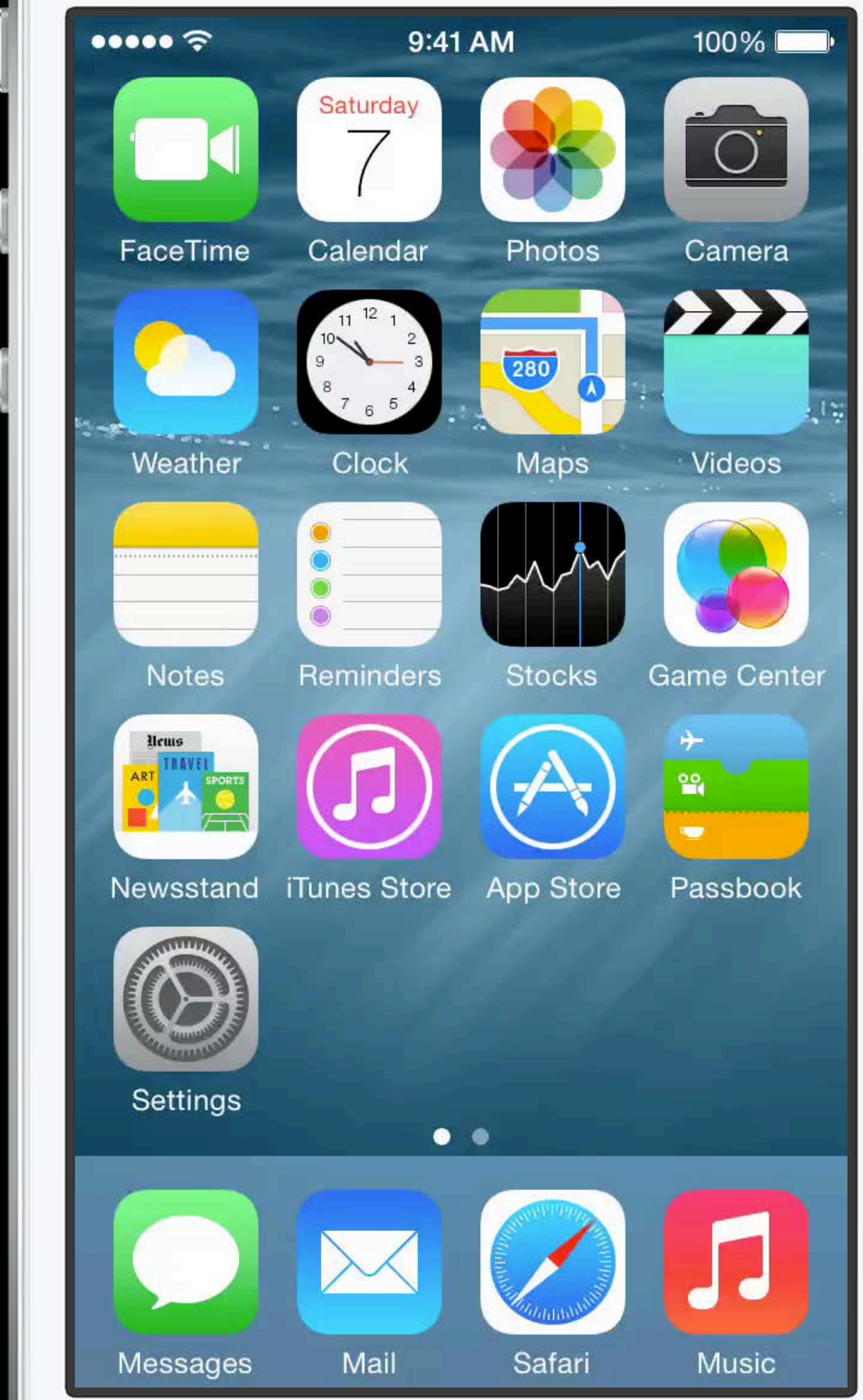
Transparent Overlays

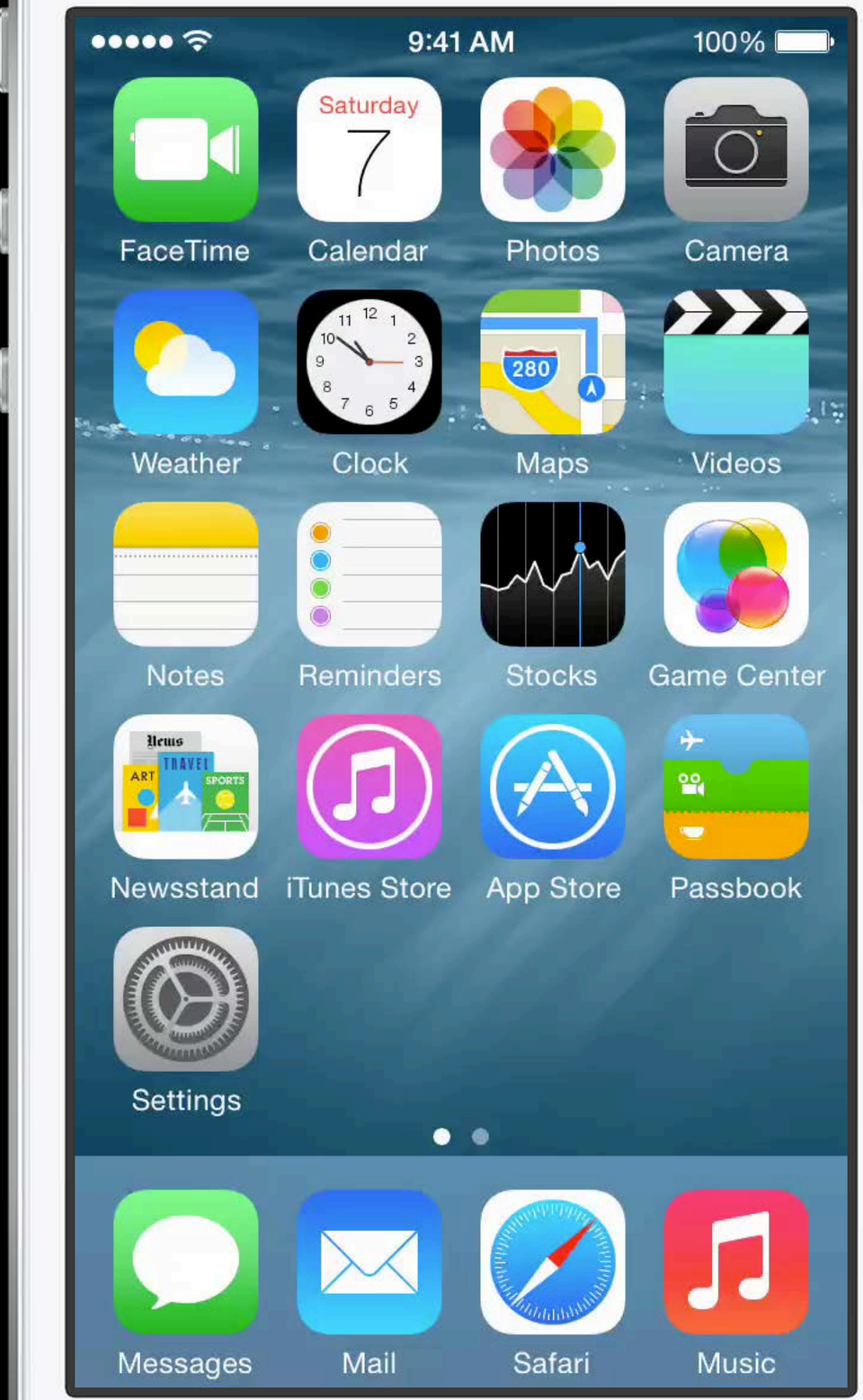
Dragging While Scrolling

Highlighting Objects

Transparent Overlays

Hit testing


9:41 AM
100%
FaceTime
Saturday
7
Calendar
Photos
Camera
Weather
Clock
280
Maps
Videos
Notes
Reminders
Stocks
Game Center
Newsstand
iTunes Store
App Store
Passbook
Settings
Messages
Mail
Safari
Music


9:41 AM
100%
FaceTime
Saturday
7
Calendar
Photos
Camera
Weather
Clock
280
Maps
Videos
Notes
Reminders
Stocks
Game Center
Newsstand
iTunes Store
App Store
Passbook
Settings
Messages
Mail
Safari
Music


9:41 AM
100%
Spotlight Search
Cancel
Q W E R T Y U I O P
A S D F G H J K L
Z X C V B N M
123
space
Search


9:41 AM
100%
Spotlight Search
Cancel
Q W E R T Y U I O P
A S D F G H J K L
Z X C V B N M
123
space
Search

Demo

Eliza Block

hitTest:withEvent:

```
func hitTest(point: CGPoint, withEvent event: UIEvent) -> UIView? {

}
```

```
func hitTest(point: CGPoint, withEvent event: UIEvent) -> UIView? {
    if /* point is in our bounds */ {

    }

    return nil
}
```

```
func hitTest(point: CGPoint, withEvent event: UIEvent) -> UIView? {
    if /* point is in our bounds */ {
        return self
    }

    return nil
}
```

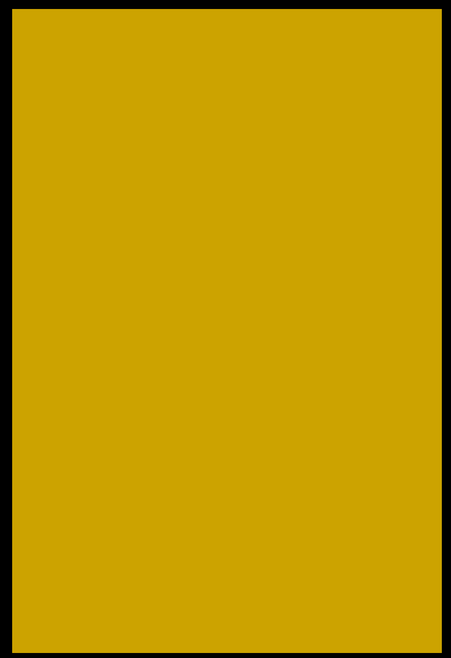
```
func hitTest(point: CGPoint, withEvent event: UIEvent) -> UIView? {
    if /* point is in our bounds */ {
        for /* each subview, in reverse order */ {
        }

        return self
    }

    return nil
}
```

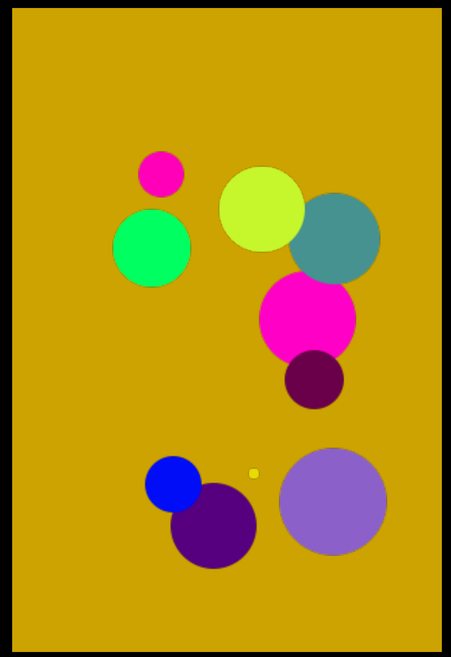
```
func hitTest(point: CGPoint, withEvent event: UIEvent) -> UIView? {
    if /* point is in our bounds */ {
        for /* each subview, in reverse order */ {
            let hitView = /* recursive call on subview */
        }

        return self
    }

    return nil
}
```

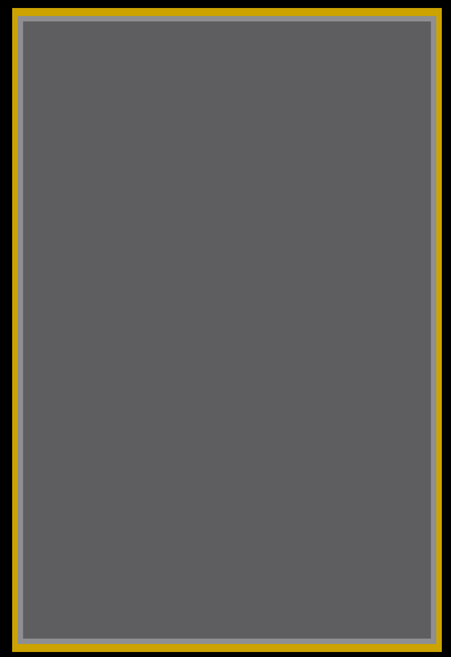
```
func hitTest(point: CGPoint, withEvent event: UIEvent) -> UIView? {
    if /* point is in our bounds */ {
        for /* each subview, in reverse order */ {
            let hitView = /* recursive call on subview */
            if hitView != nil {
                return hitView
            }
        }

        return self
    }

    return nil
}
```

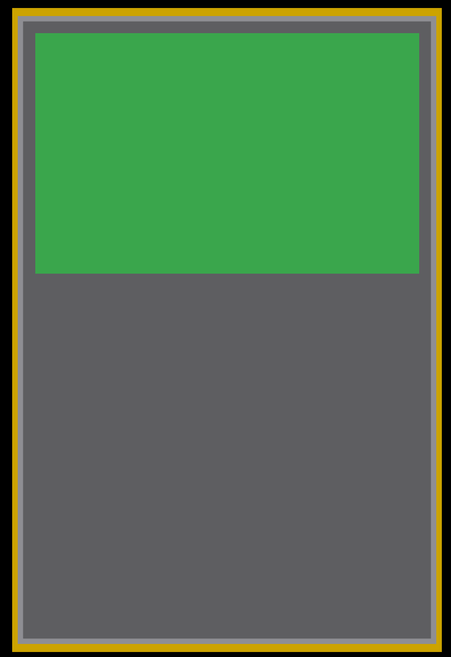
```
func hitTest(point: CGPoint, withEvent event: UIEvent) -> UIView? {
    if /* point is in our bounds */ {
        for /* each subview, in reverse order */ {
            let hitView = /* recursive call on subview */
            if hitView != nil {
                return hitView
            }
        }

        return self
    }

    return nil
}
```

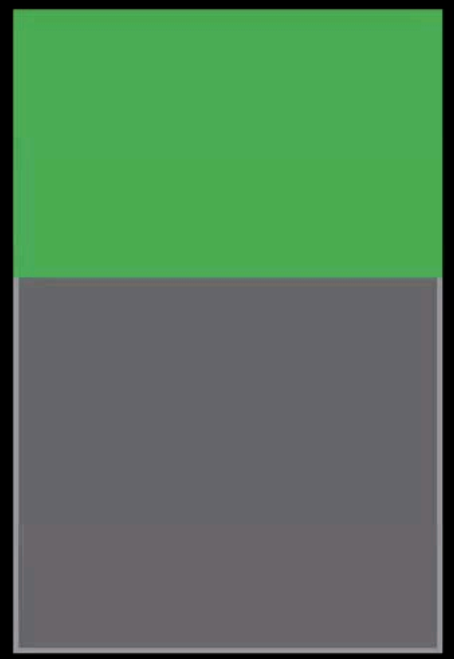
```
- (UIView *)hitTest:(CGPoint)point withEvent:(UIEvent *)event {
    if (/* point is in our bounds */) {
        for (/* each subview, in reverse order */) {
            UIView *hitView = /* recursive call on subview */
            if (hitView != nil) {
                return hitView;
            }
        }

        return self;
    }

    return nil;
}
```

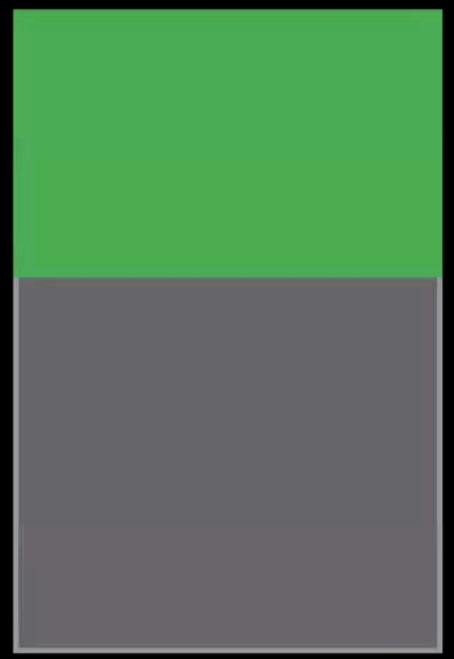


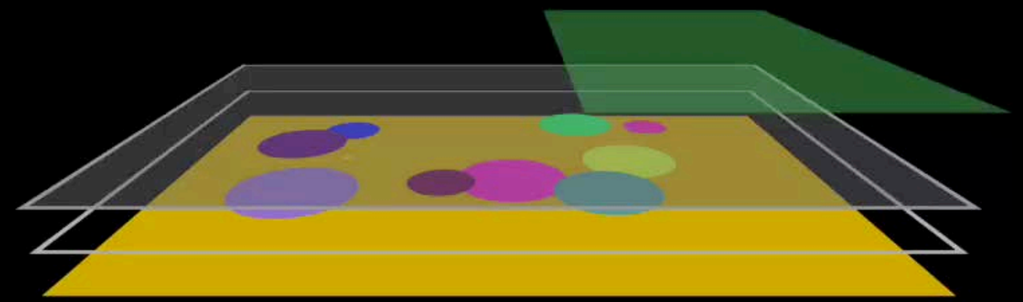


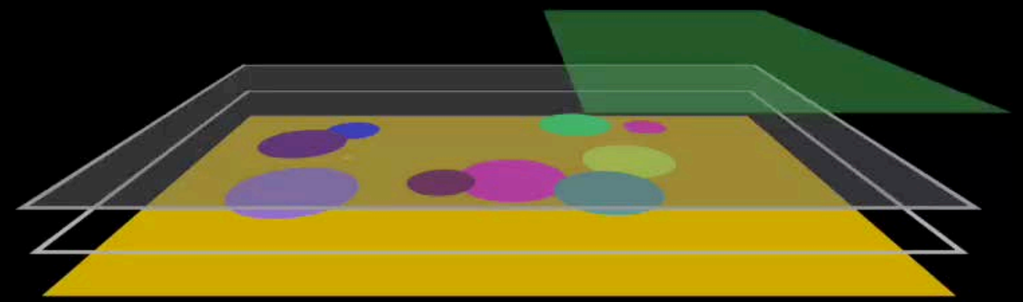


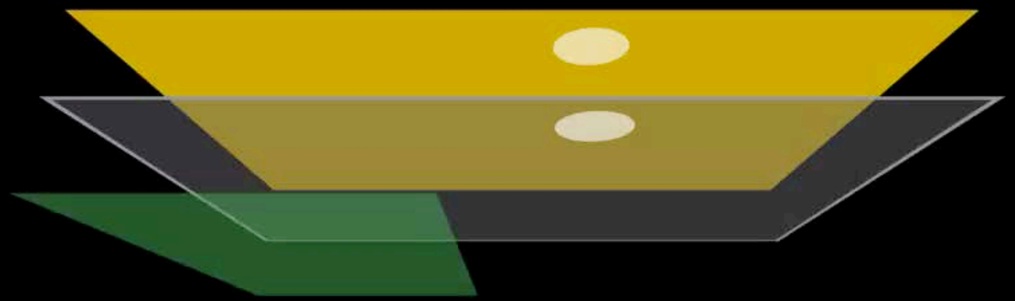


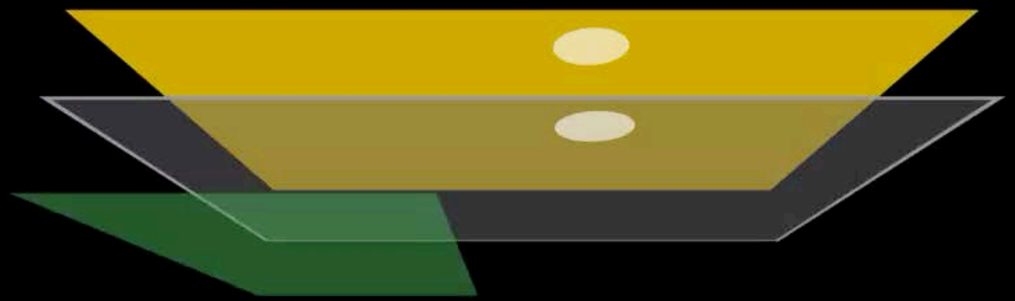












```
override func hitTest(point: CGPoint, withEvent event: UIEvent) -> UIView? {

}
```

```
override func hitTest(point: CGPoint, withEvent event: UIEvent) -> UIView? {
    let hitView = super.hitTest(point, withEvent: event)

    return hitView
}
```

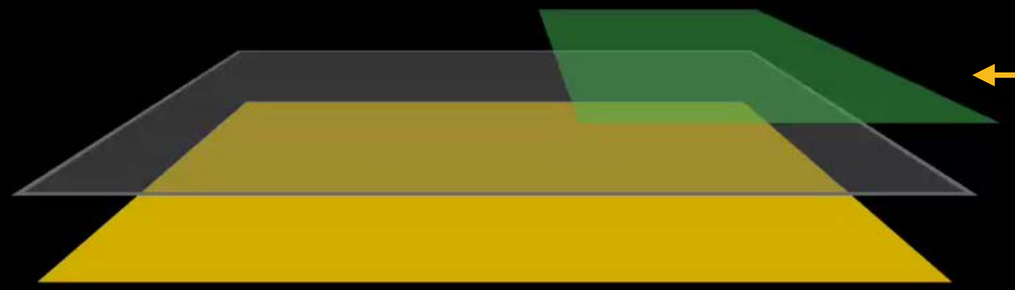
```swift
override func hitTest(point: CGPoint, withEvent event: UIEvent) -> UIView? {
    let hitView = super.hitTest(point, withEvent: event)

    if hitView == self {
        return nil
    }

    return hitView
}
```

Demo

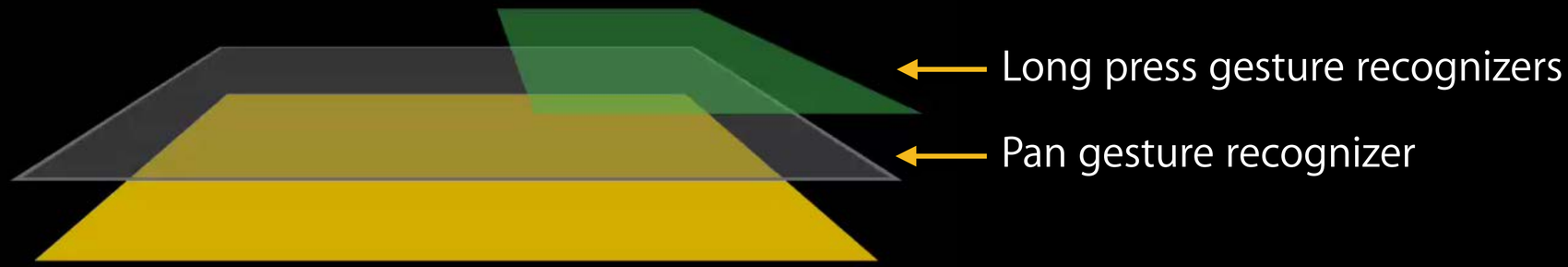
Eliza Block

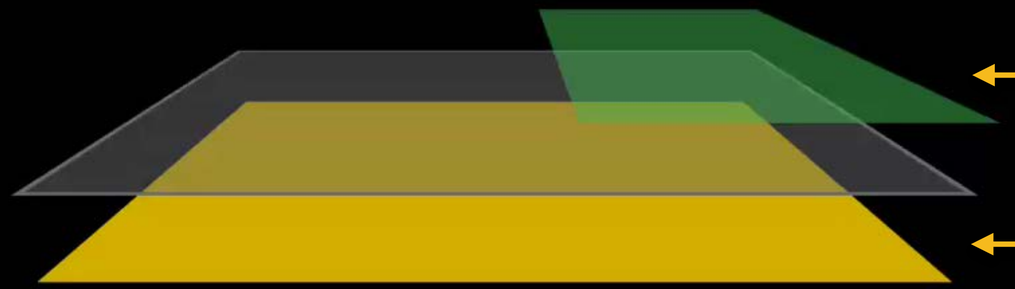
Dragging While Scrolling

Gesture recognizers



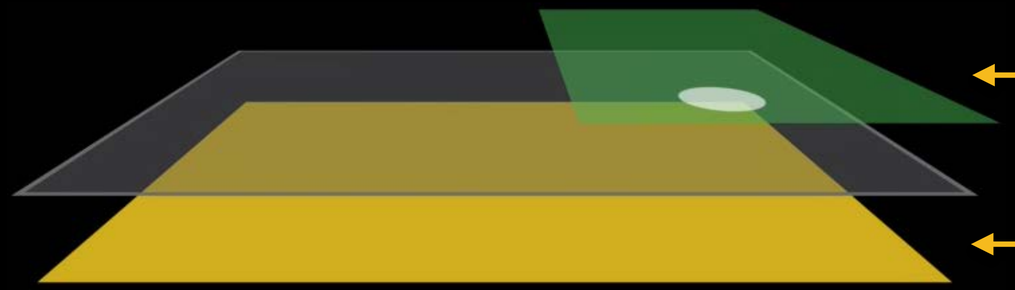
Long press gesture recognizers


Long press gesture recognizers
Pan gesture recognizer



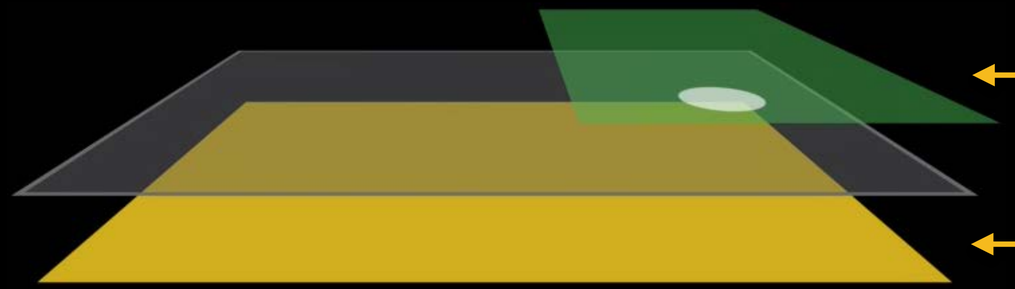
Long press gesture recognizers

Pan gesture recognizer



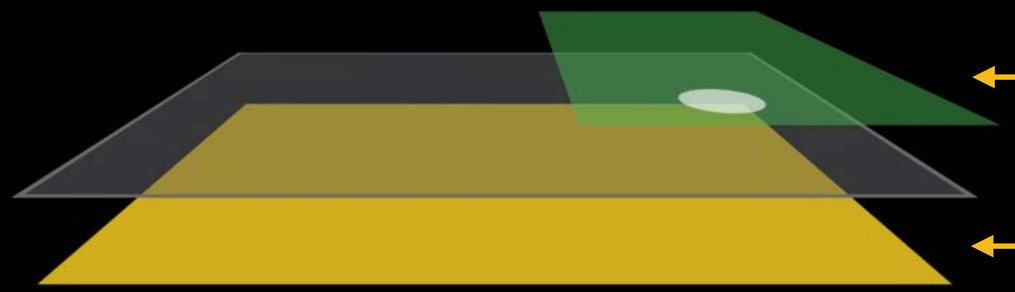
Long press gesture recognizers

Pan gesture recognizer



Long press gesture recognizers

Pan gesture recognizer



Long press gesture recognizers

Pan gesture recognizer

Stop the Pan

```
pan.enabled = false
```

```
pan.enabled = true
```

Demo

Eliza Block

Highlighting Objects

Touch delivery


9:41 AM
100%
June
Fri Jun 6
Sat Jun 14
all-day
Flag Day
Sun Jun 15
all-day
Father's Day
Fri Jul 4
all-day
Independence Day
Mon Sep 1
all-day
Labor Day
Mon Oct 13
all-day
Columbus Day
Fri Oct 31
Today
Calendars
Inbox


9:41 AM
100%
June
Fri Jun 6
Sat Jun 14
all-day
Flag Day
Sun Jun 15
all-day
Father's Day
Fri Jul 4
all-day
Independence Day
Mon Sep 1
all-day
Labor Day
Mon Oct 13
all-day
Columbus Day
Fri Oct 31
Today
Calendars
Inbox

9:41 AM 100%

June

Sat Jun 14

Sun Jun 15

all-day Father's Day

Fri Jul 4

all-day Independence Day

Mon Sep 1

all-day Labor Day

Mon Oct 13

all-day Columbus Day

Fri Oct 31

all-day Halloween

Sun Nov 2

Today Calendars Inbox

delaysContentTouches

Pan Gesture Recognizer

Pan Gesture Recognizer

Pinch Gesture Recognizer

Pan Gesture Recognizer

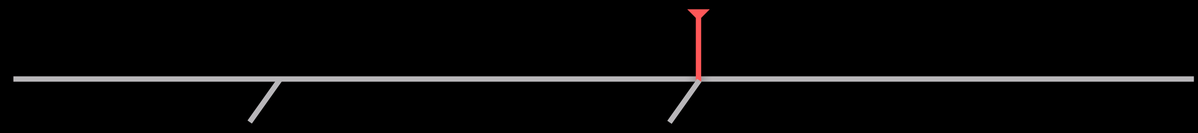
Pinch Gesture Recognizer

Touch Delay Gesture Recognizer

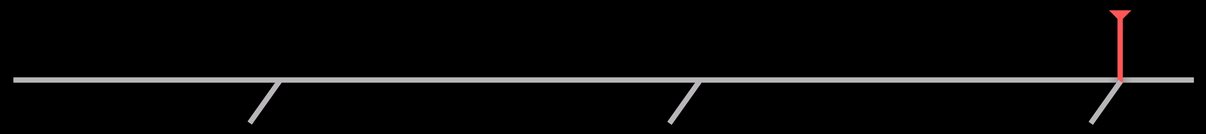
delaysTouchesBegan

Touch			
Pan Gesture			
Delay Gesture			
View			

Touch	down		
Pan Gesture			
Delay Gesture	timer start		
View			



Touch	down		
Pan Gesture			
Delay Gesture	timer start	fail	
View		touchesBegan	



Touch	down		move
Pan Gesture			recognize
Delay Gesture	timer start	fail	
View		touchesBegan	touchesCancelled

Touch		
Pan Gesture		
Delay Gesture		
View		

Touch	down	
Pan Gesture		
Delay Gesture	timer start	
View		

Touch	down	move
Pan Gesture		recognize
Delay Gesture	timer start	reset
View		

```
init(target: AnyObject!, action: Selector) {
}
```

```
    init(target: AnyObject!, action: Selector) {
        // set delaysTouchesBegan to true
    }
```

```
func touchesBegan(touches: NSSet!, withEvent event: UIEvent!) {

}

func touchesEnded(touches: NSSet!, withEvent event: UIEvent!) {

}

func touchesCancelled(touches: NSSet!, withEvent event: UIEvent!) {

}
```

```
func touchesBegan(touches: NSSet!, withEvent event: UIEvent!) {
    // start timer
}

func touchesEnded(touches: NSSet!, withEvent event: UIEvent!) {
}

func touchesCancelled(touches: NSSet!, withEvent event: UIEvent!) {

}
```

```
func touchesBegan(touches: NSSet!, withEvent event: UIEvent!) {
    // start timer
}

func touchesEnded(touches: NSSet!, withEvent event: UIEvent!) {
    // set state to .Failed
}

func touchesCancelled(touches: NSSet!, withEvent event: UIEvent!) {
    // set state to .Failed
}
```

```
func timerFired() {

}
```

```
func timerFired() {
    // set state to .Failed
}
```

```
func timerFired() {
    // set state to .Failed
}

func reset() {

}
```

```
func timerFired() {
    // set state to .Failed
}

func reset() {
    // clear and reset timer
}
```

Demo

Eliza Block

Touching Small Objects

Hit testing

Minimum Hit Target Size

```
func hitTest(point: CGPoint, withEvent event: UIEvent) -> UIView? {
    if /* point is in our bounds */ {
        for /* each subview, in reverse order */ {
            let hitView = /* recursive call on subview */
            if hitView != nil {
                return hitView
            }
        }

        return self
    }

    return nil
}
```

```
func hitTest(point: CGPoint, withEvent event: UIEvent) -> UIView? {
    if /* point is in our bounds */ {
        for /* each subview, in reverse order */ {
            let hitView = /* recursive call on subview */
            if hitView != nil {
                return hitView
            }
        }

        return self
    }

    return nil
}
```

```
func hitTest(point: CGPoint, withEvent event: UIEvent) -> UIView? {
    if pointInside(point, withEvent: event) {
        for /* each subview, in reverse order */ {
            let hitView = /* recursive call on subview */
            if hitView != nil {
                return hitView
            }
        }

        return self
    }

    return nil
}
```

```
func pointInside(point: CGPoint, withEvent event: UIEvent) -> Bool {
}
```

```
func pointInside(point: CGPoint, withEvent event: UIEvent) -> Bool {
    return CGRectContainsPoint(bounds, point)
}
```

Demo

Eliza Block

More Information

Jake Behrens
App Frameworks Evangelist
behrens@apple.com

Documentation
Scroll View Programming Guide for iOS
http://developer.apple.com/ios

Apple Developer Forums
http://devforums.apple.com

Related Sessions

• Building Interruptible and Responsive Interactions	Presidio	Friday 11:30AM

Labs

• Open Hours	Frameworks Lab A/B	Friday 2:00PM

WWDC14