

# CS193p

## Spring 2010



# Announcements

- If you haven't received an e-mail with the sentence "Approved." in it, then your project has not yet been approved!

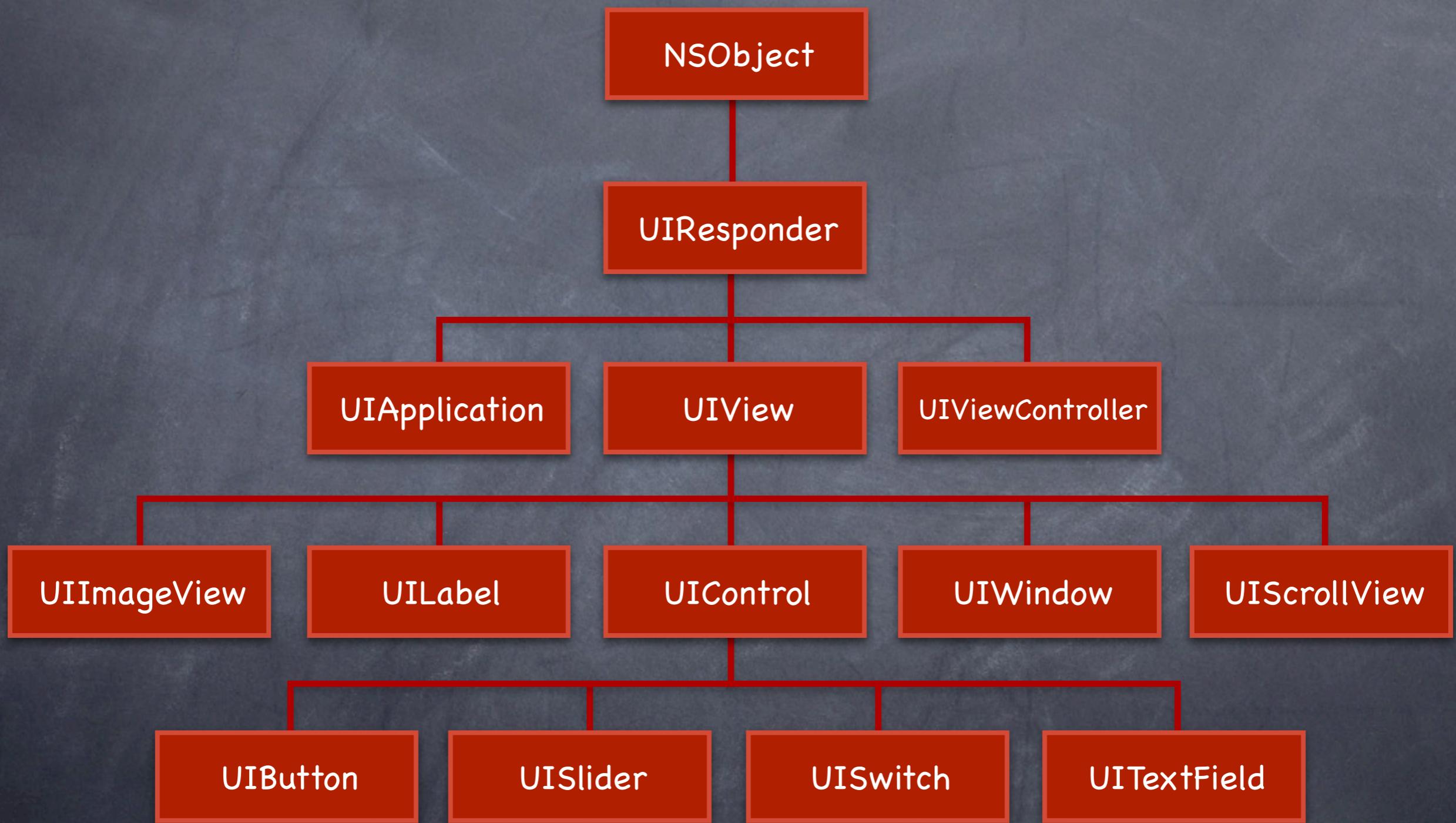
# Today's Topics

- ⌚ Multitouch without UIGestureRecognizer
- ⌚ UITextField
- ⌚ Modal View Controller

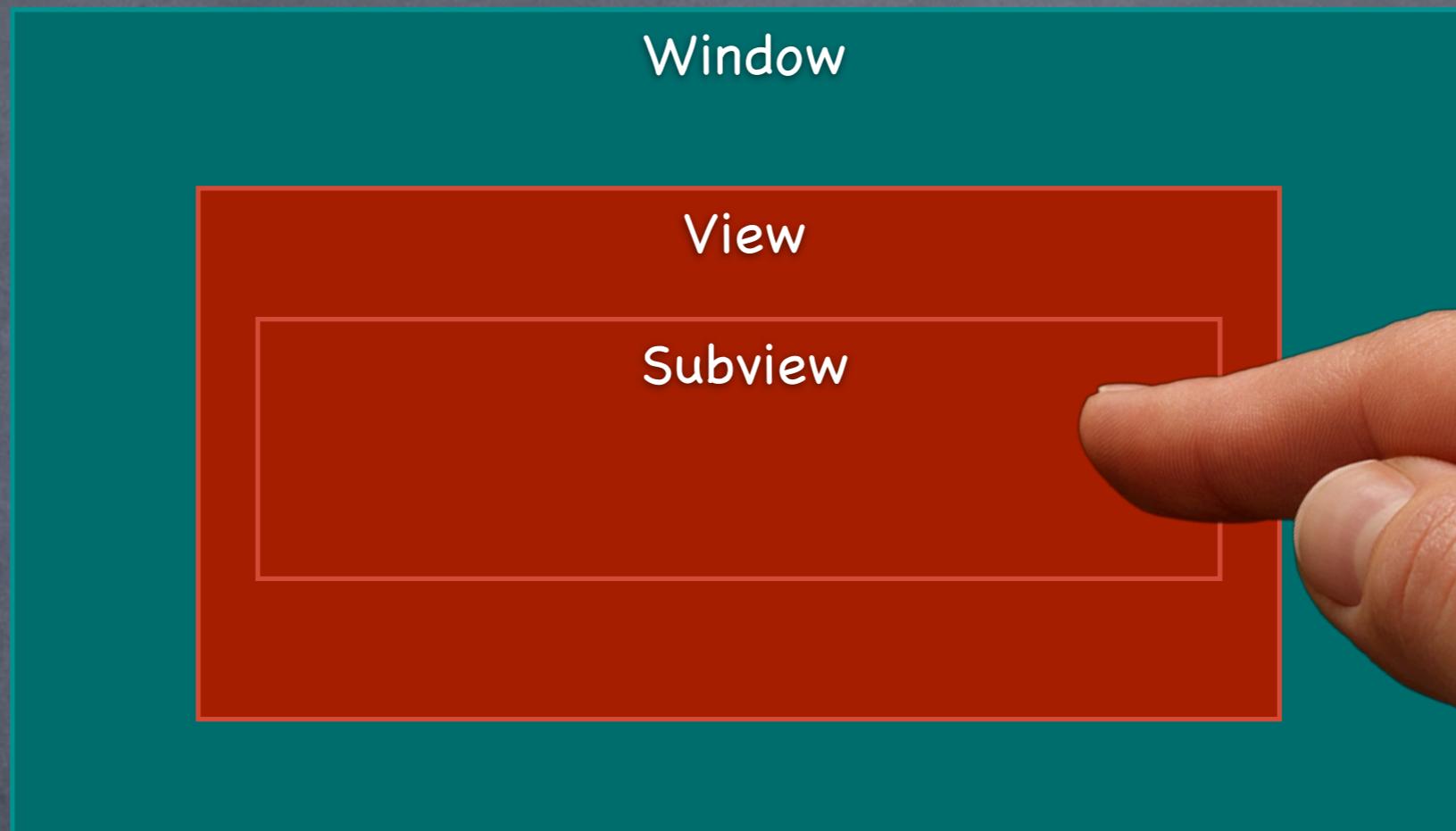
# Multitouch

- ⦿ UIKit manages a “responder chain” in each window
- ⦿ Consists of UIView objects and their controllers
- ⦿ Touch events run up the responder chain  
Looking for someone to handle the event
- ⦿ They start at “active” view under the touch
  - `userInteractionEnabled` is YES
  - `hidden` is NO
  - `alpha` is greater than or equal to 0.1
- ⦿ Then check its view controller, then superview
  - Object must be a subclass of UIResponder to participate
- ⦿ Stops when it finds implementor of “the methods”

# UIResponder



# Responder Chain

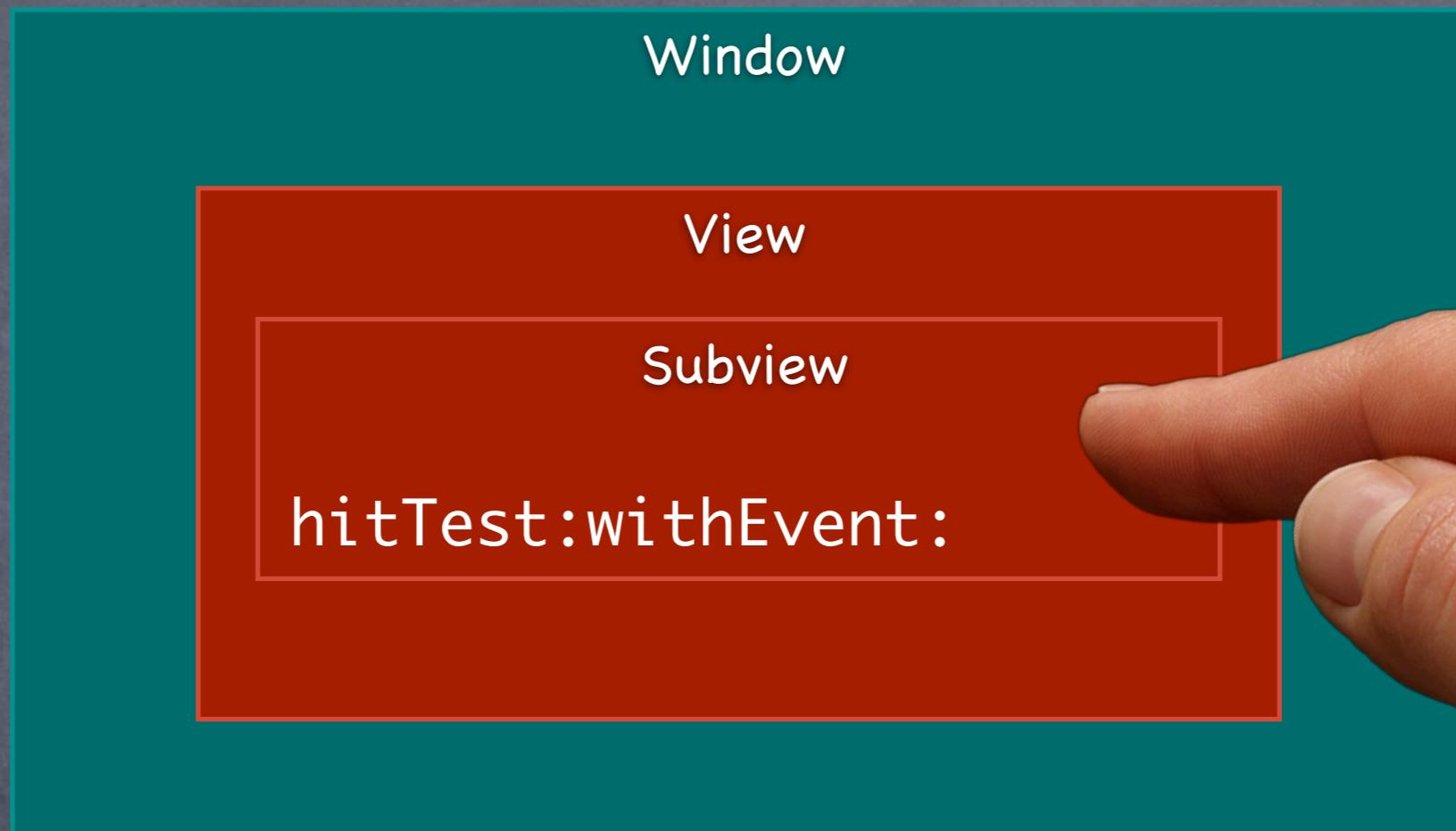


# Responder Chain



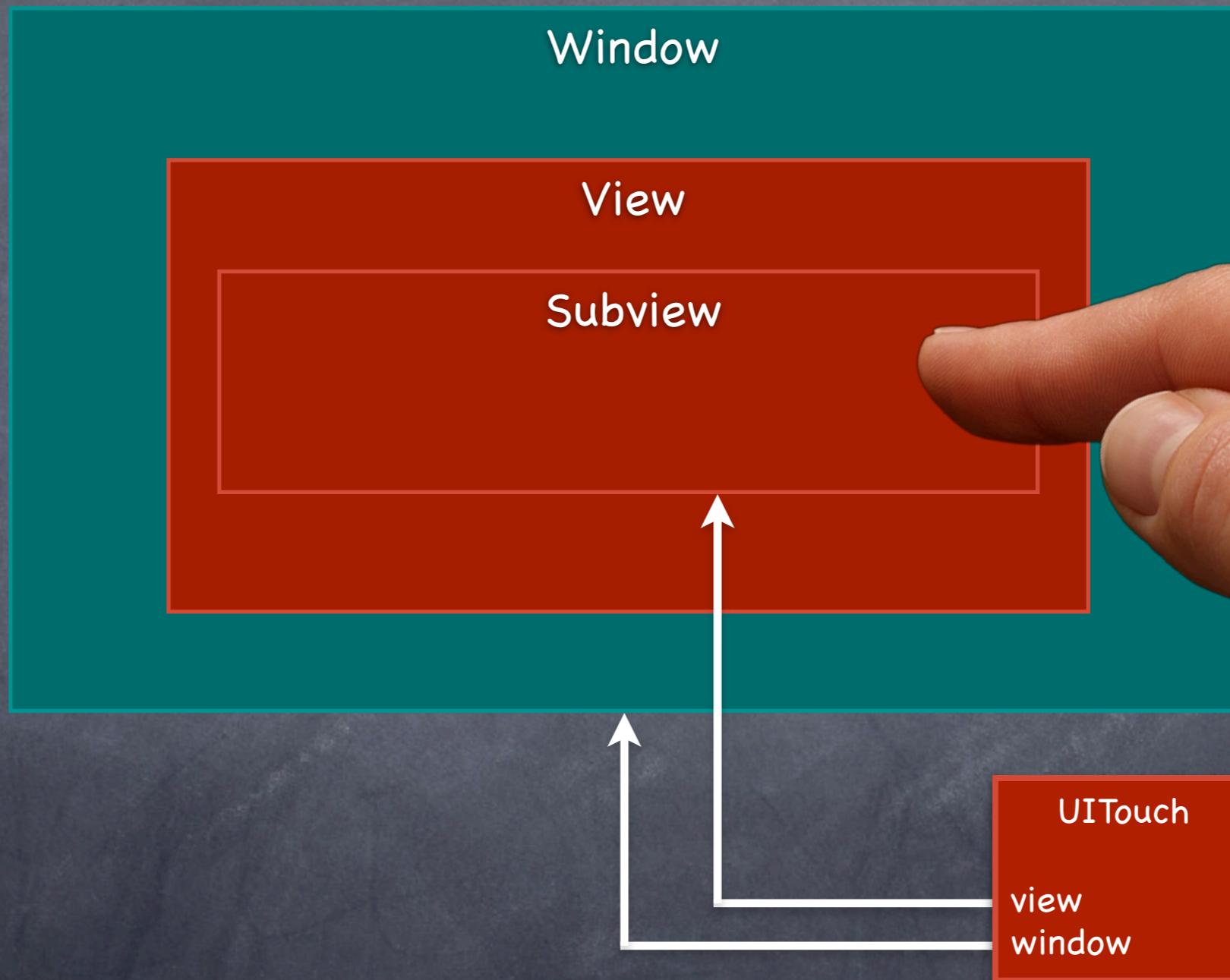
- `userInteractionEnabled`
- `hidden/alpha`
- `pointInside:withEvent:`

# Responder Chain

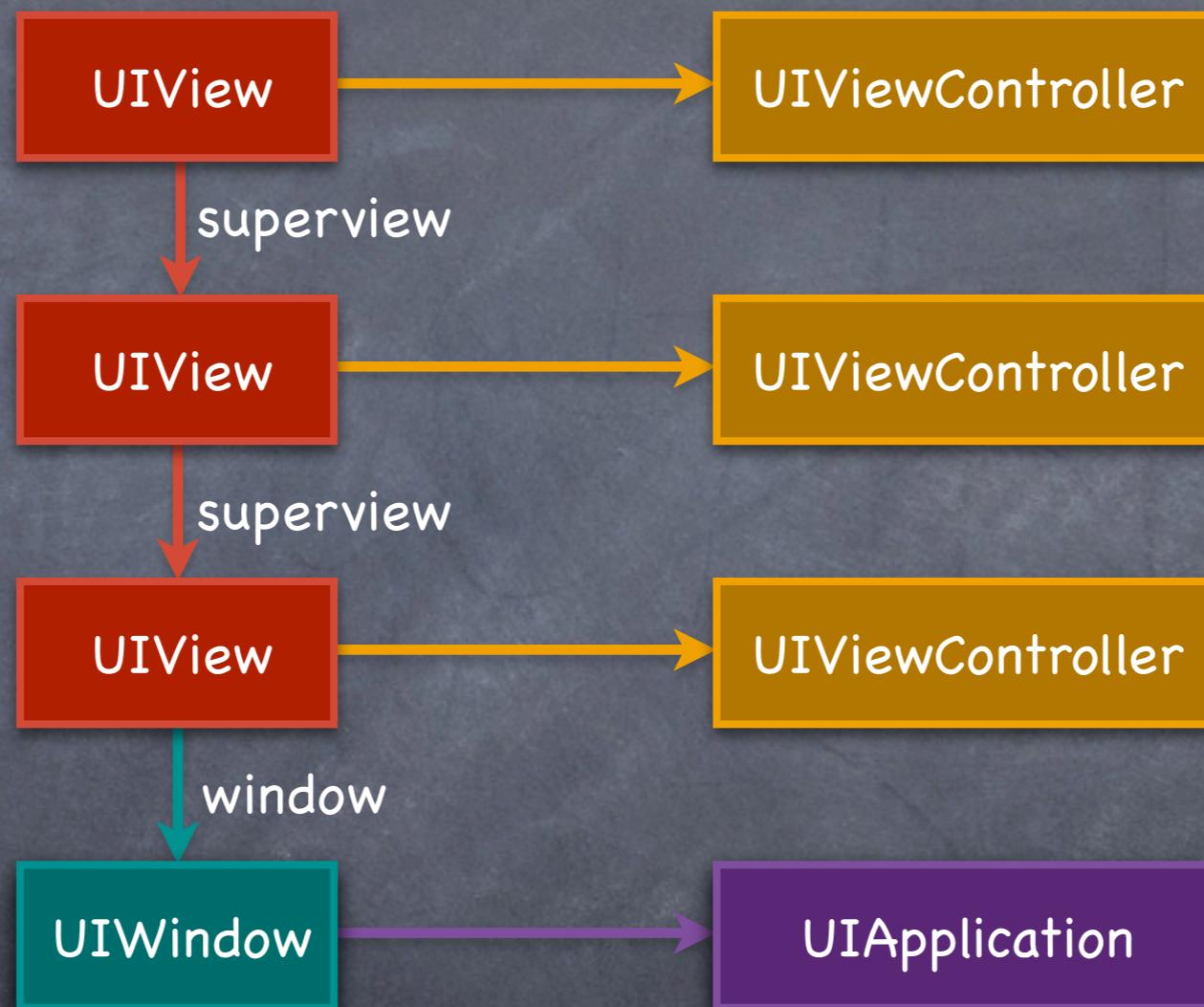


- `userInteractionEnabled`
- `hidden/alpha`
- `pointInside:withEvent:`

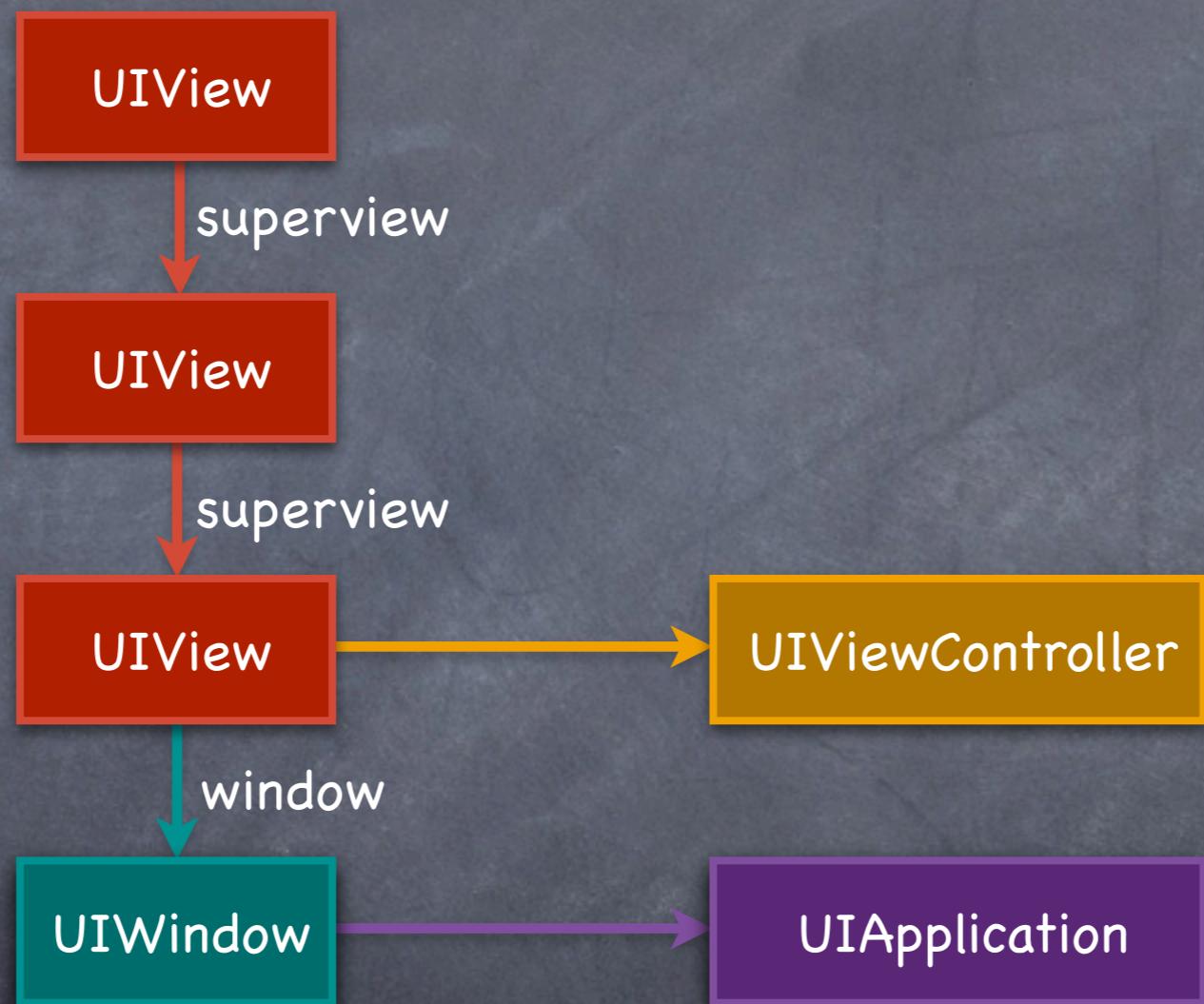
# Responder Chain



# Responder Chain



# Responder Chain



# Multitouch

## ⦿ The methods

- (void)touchesBegan:(NSSet \*)touches withEvent:(UIEvent \*)event;
- (void)touchesMoved:(NSSet \*)touches withEvent:(UIEvent \*)event;
- (void)touchesEnded:(NSSet \*)touches withEvent:(UIEvent \*)event;
- (void)touchesCancelled:(NSSet \*)touches withEvent:(UIEvent \*)event;

The NSSets above contain one or more UITouch objects (more on this later).

## ⦿ Implement ALL FOUR!

You don't one one responder implementing began and a different one implementing moved.

## ⦿ By default, a view will only receive "unitouches"

The NSSets above will only have one UITouch in them.

You must explicitly enable a UIView to receive multi-finger touches.

Set `multipleTouchEnabled` in a UIView to YES.

# Multitouch

- ⦿ The methods

- `(void)touches<Phase>:(NSSet *)touches withEvent:(UIEvent *)event;`

- ⦿ **UITouch objects (that arrive in above NSSets)**

- `@property UITouchPhase phase;`

- `UITouchPhase` is just `Began`, `Moved`, `Ended` or `Cancelled` (or `Stationary`)

- `(CGPoint)locationInView:(UIView *)aView;`

- `@property int tapCount;`

- `@property NSTimeInterval timestamp;`

- ⦿ The `UIEvent` also has interesting information

- `@property NSSet allTouches; // all fingers that are down, even if not multitouch`

- `(NSSet *)touchesForView:(UIView *)aView;`

- `(NSSet *)touchesForWindow:(UIWindow *)aWindow;`

# Multitouch

- ⦿ The methods

- `(void)touches<Phase>: (NSSet *)touches withEvent: (UIEvent *)event;`

- ⦿ If multitouch `enabled`, NSSet can contain multiple  
If they are touched down at the same time.

- Even if multitouch not `enabled`, the UIEvent might have multiple touches.

- ⦿ Note that the touch event is persistent throughout

- In other words, the pointer stays the same from began to moved to ended.

- So don't retain them or copy them.

- Copy any information you need out of the UITouch into other variables.

- Usually use static variables to track state throughout a touch sequence,  
not ivars (because once it has ended, it makes no sense to keep such state).



UITouch 0x123

Phase: Began  
Location: 160, 120

touchesBegan:withEvent:



touchesMoved:withEvent:



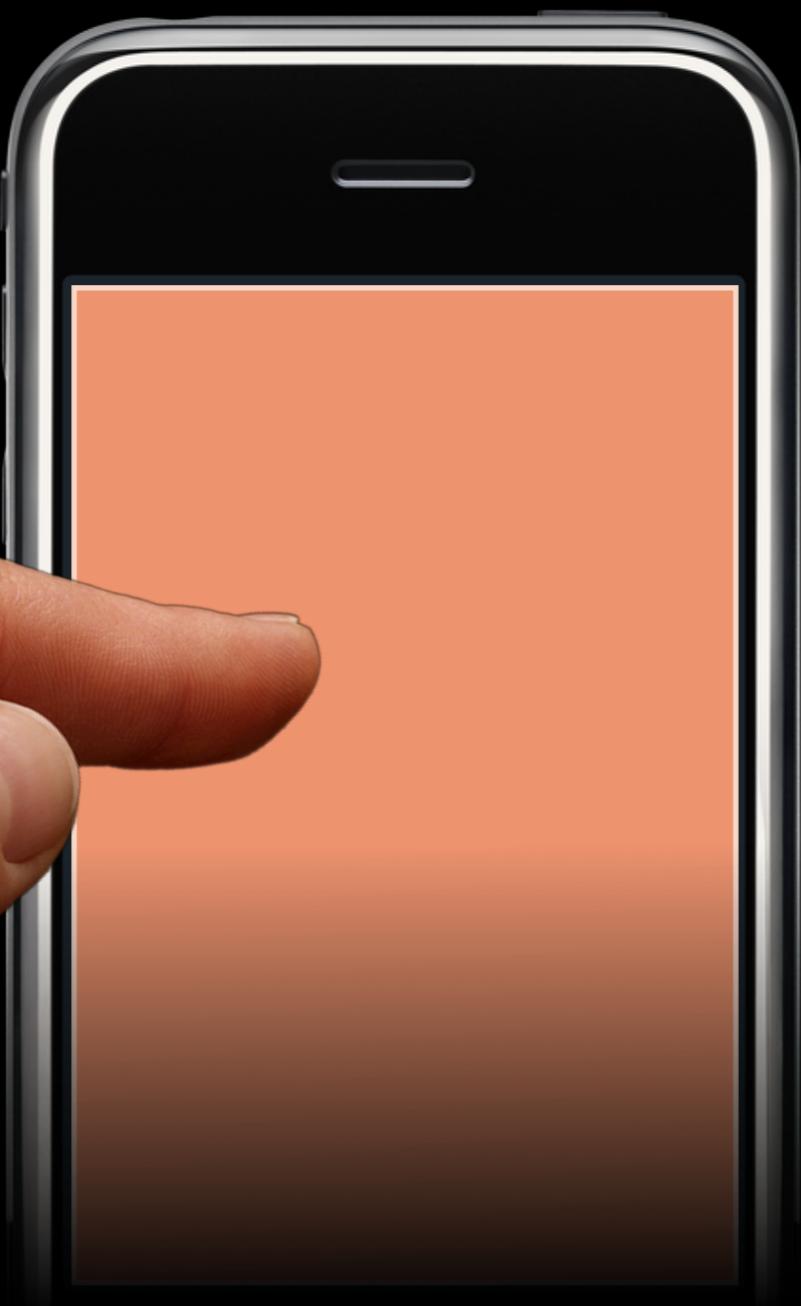
touchesMoved:withEvent:



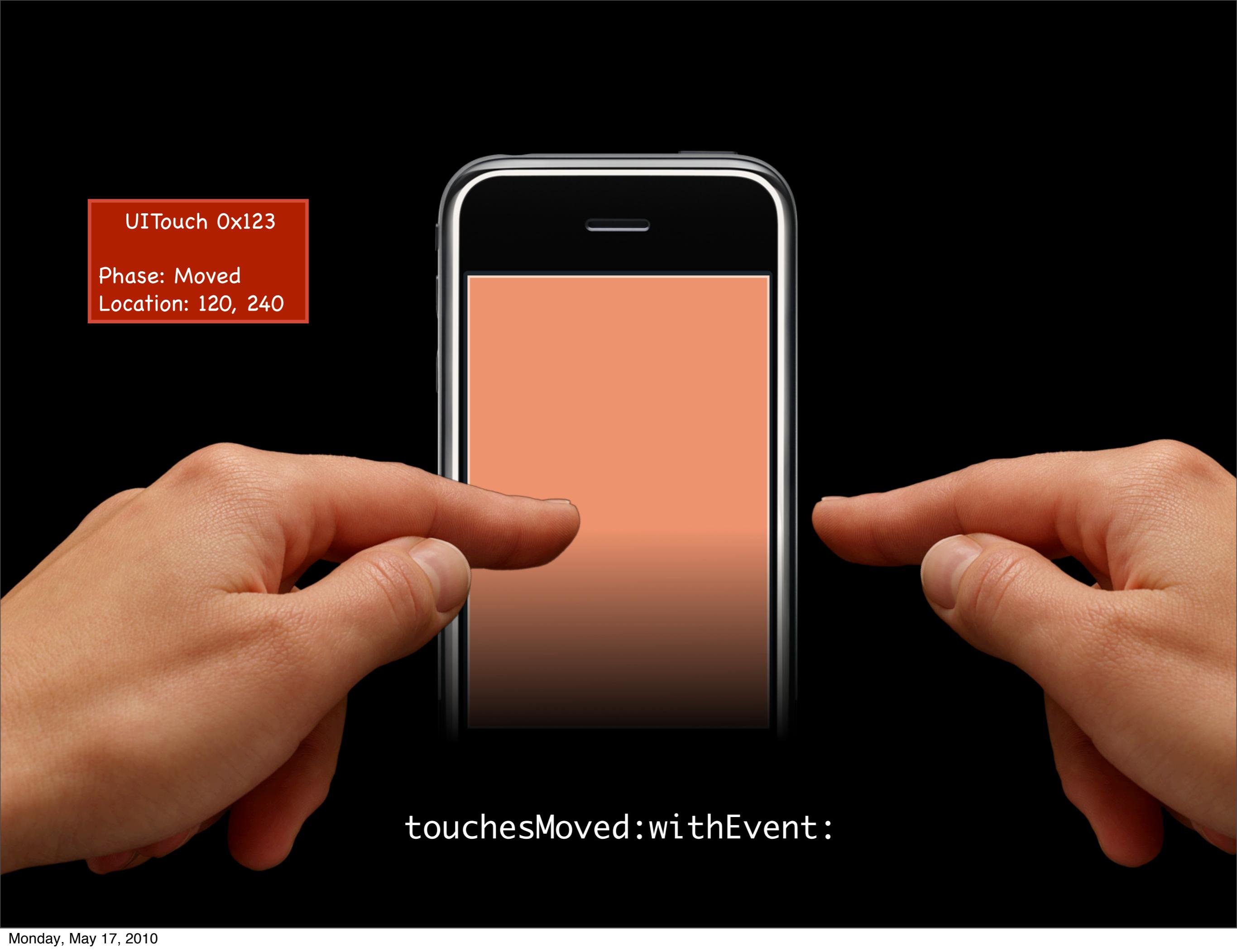
UITouch 0x123  
Phase: Ended  
Location: 160, 240

touchesEnded:withEvent:

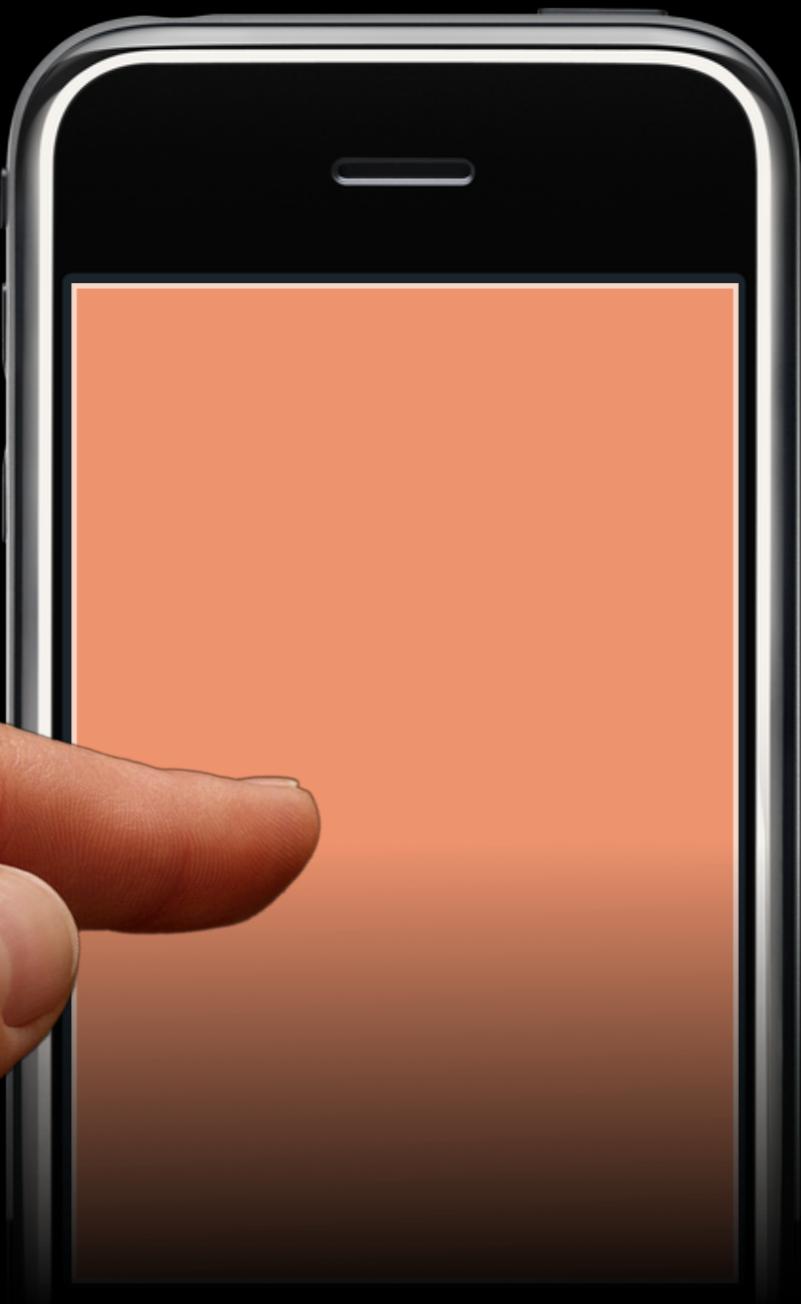
UITouch 0x123  
Phase: Began  
Location: 120, 200



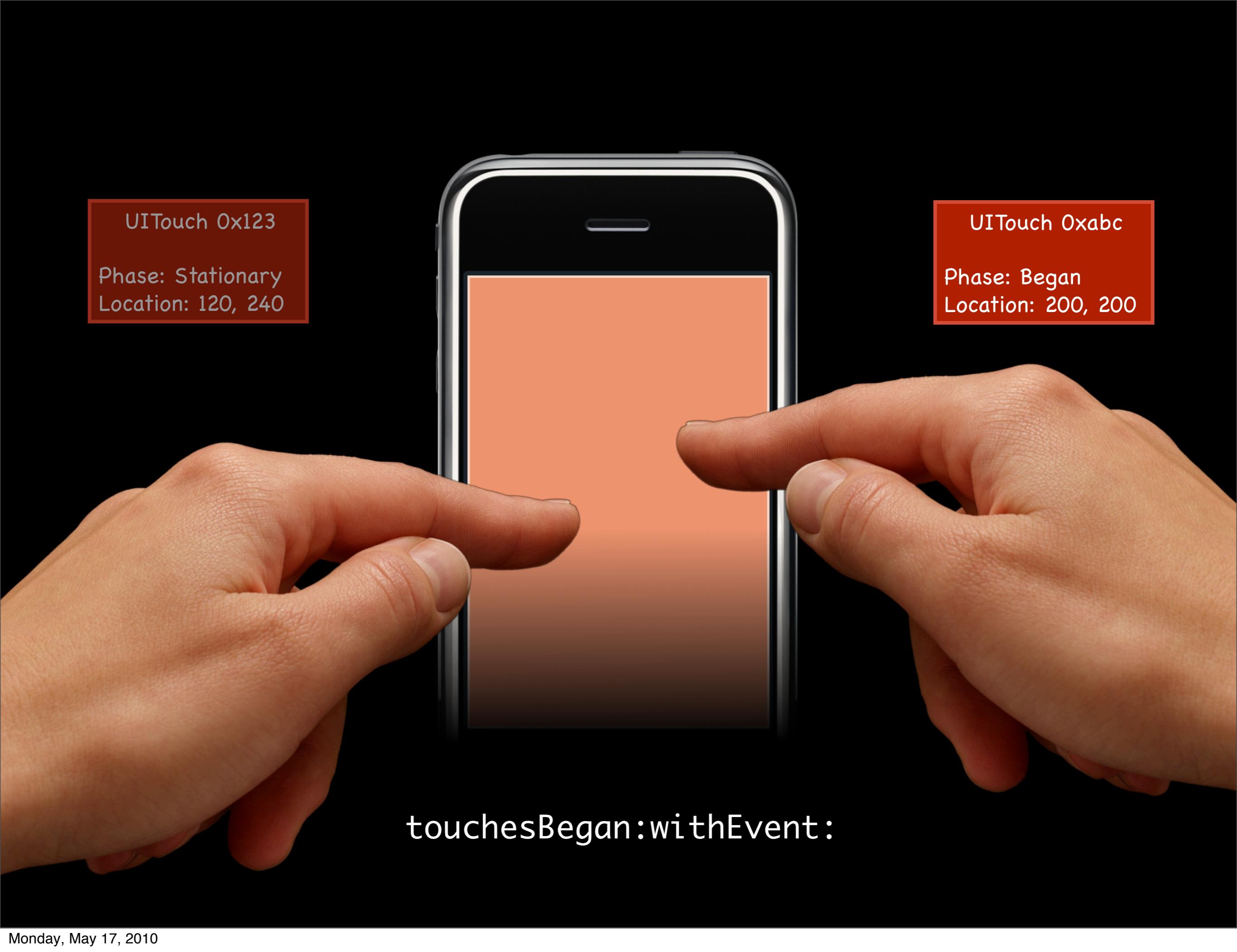
touchesBegan:withEvent:



UITouch 0x123  
Phase: Moved  
Location: 120, 240



touchesMoved:withEvent:



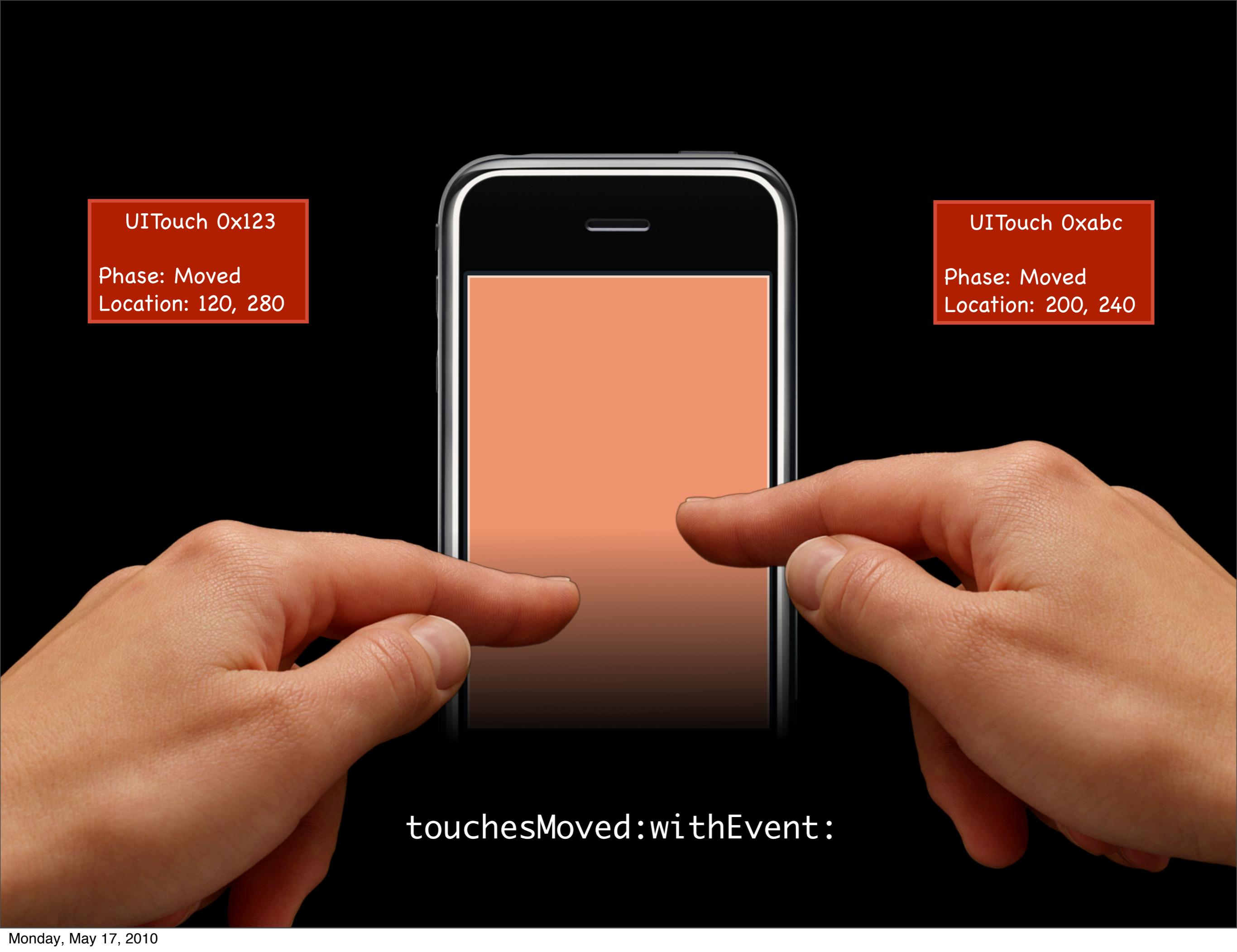
UITouch 0x123

Phase: Stationary  
Location: 120, 240

UITouch 0xabc

Phase: Began  
Location: 200, 200

touchesBegan:withEvent:



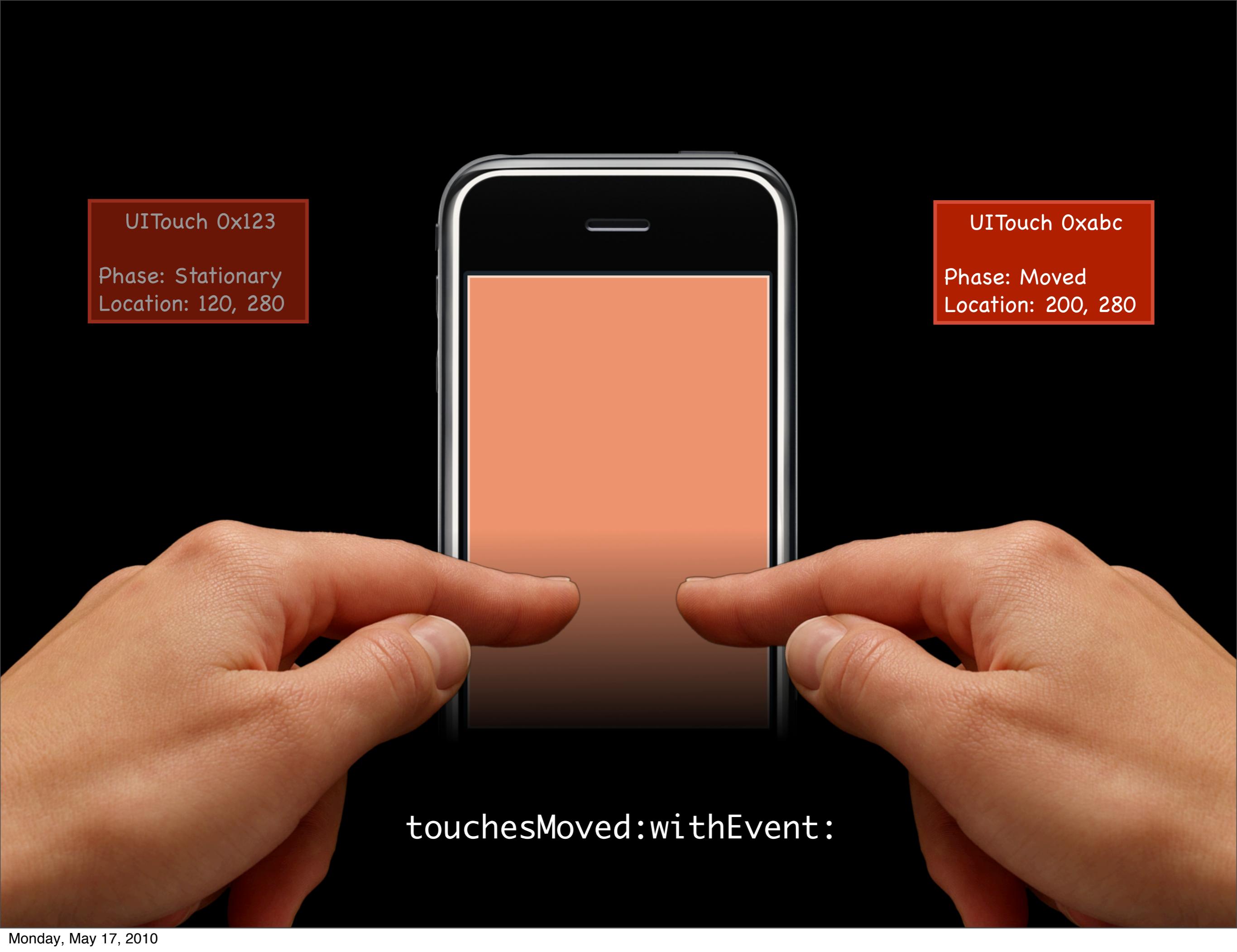
UITouch 0x123

Phase: Moved  
Location: 120, 280

UITouch 0xabc

Phase: Moved  
Location: 200, 240

touchesMoved:withEvent:



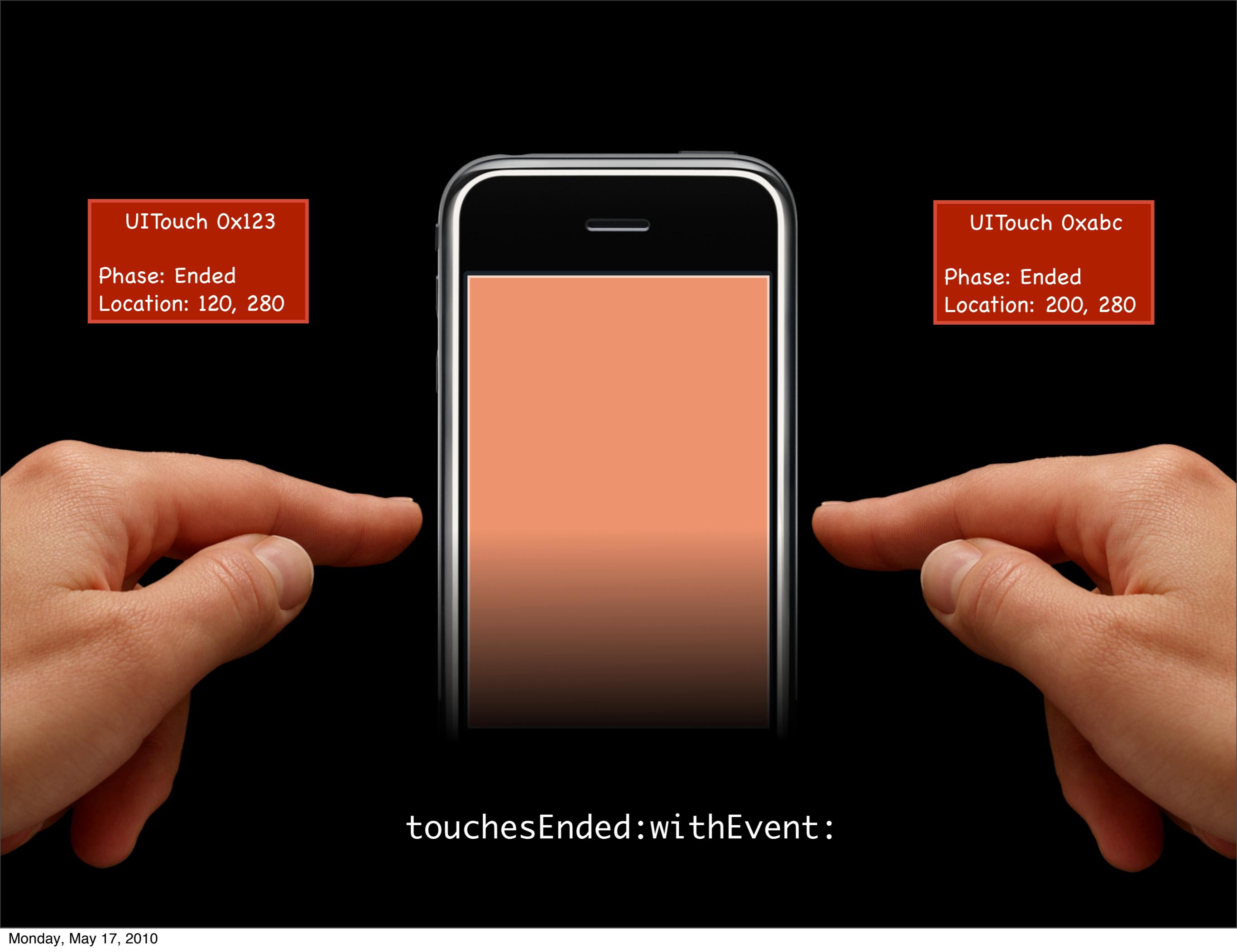
UITouch 0x123

Phase: Stationary  
Location: 120, 280

UITouch 0xabc

Phase: Moved  
Location: 200, 280

touchesMoved:withEvent:

A close-up photograph of a person's hands interacting with a white iPhone. The phone's screen displays a single orange square centered on the black background. Two hands are visible; the left hand's index finger is pointing towards the top-left corner of the orange square, while the right hand's index finger is pointing towards the top-right corner. The phone has a thin silver frame and a small speaker at the top edge.

UITouch 0x123

Phase: Ended  
Location: 120, 280

UITouch 0xabc

Phase: Ended  
Location: 200, 280

touchesEnded:withEvent:

UITouch 0x123

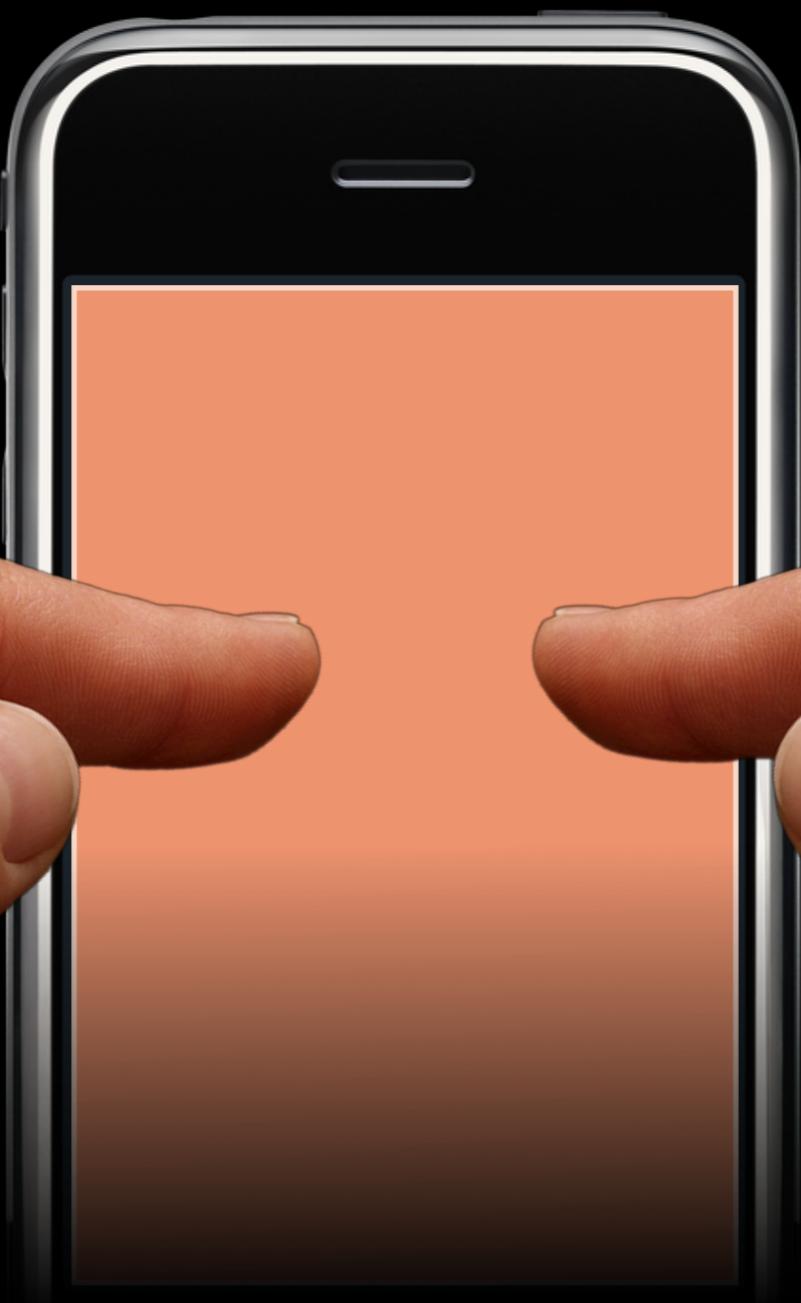
Phase: Began

Location: 120, 200

UITouch 0xabc

Phase: Began

Location: 200, 200



touchesBegan:withEvent:

UITouch 0x123

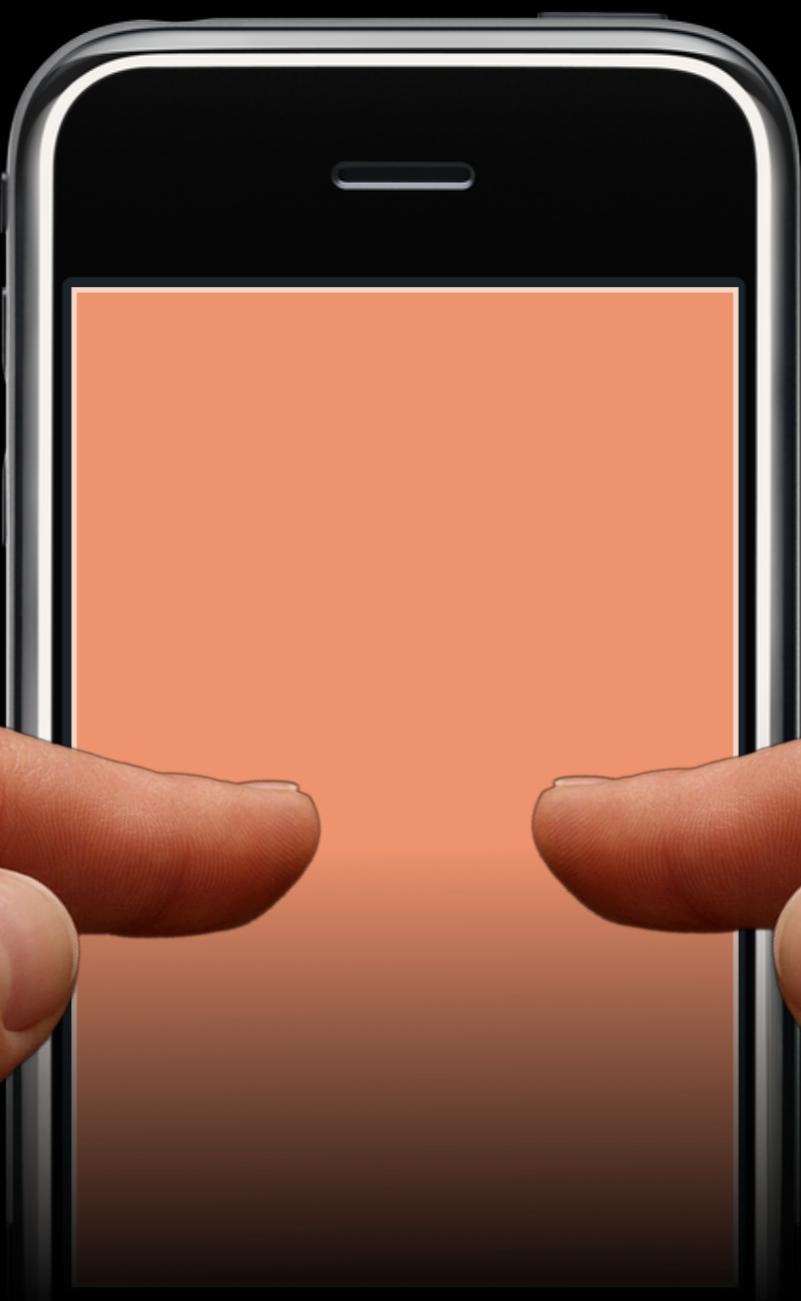
Phase: Moved

Location: 120, 240

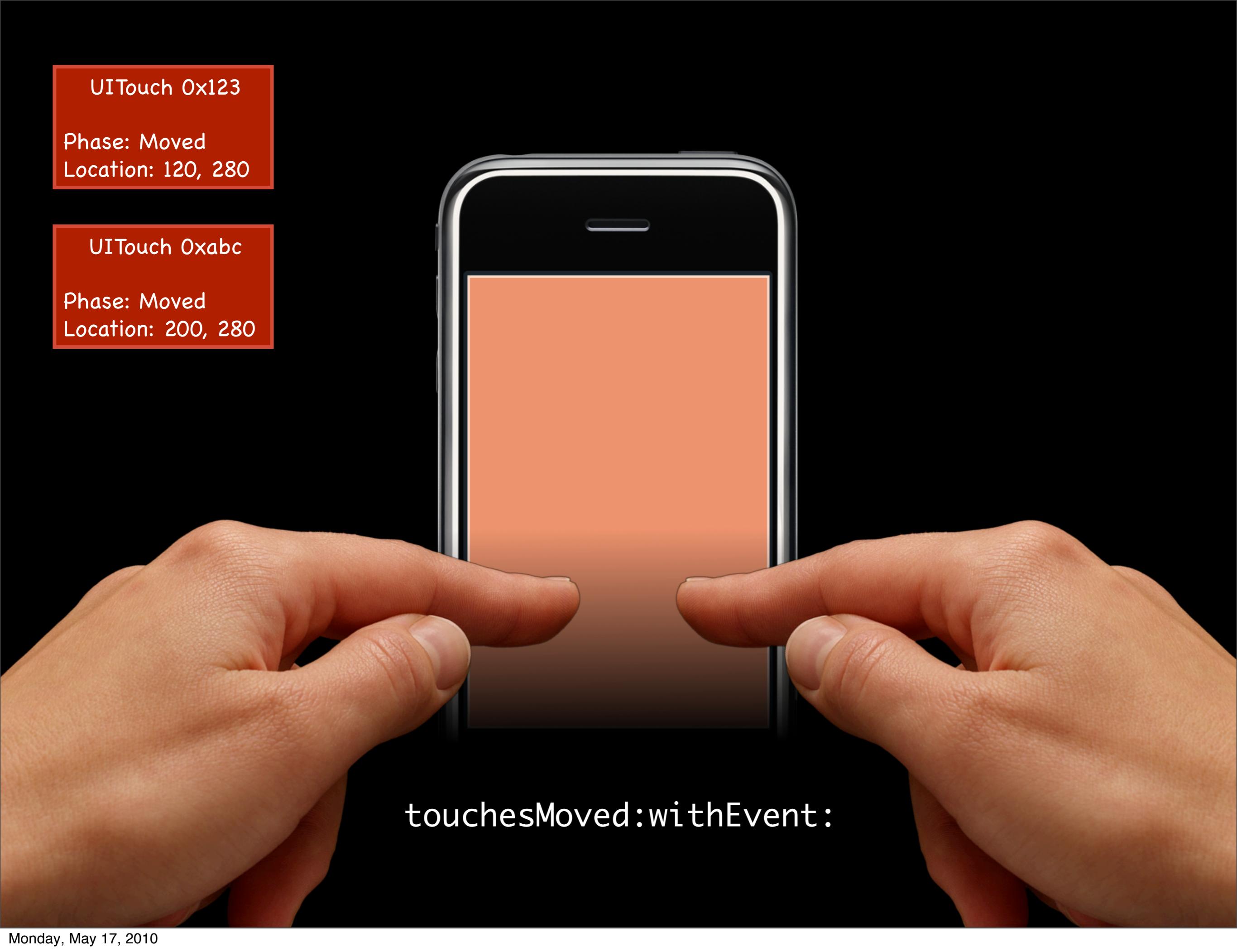
UITouch 0xabc

Phase: Moved

Location: 200, 240



touchesMoved:withEvent:



UITouch 0x123

Phase: Moved

Location: 120, 280

UITouch 0xabc

Phase: Moved

Location: 200, 280

touchesMoved:withEvent:

UITouch 0x123

Phase: Ended

Location: 120, 280

UITouch 0xabc

Phase: Ended

Location: 200, 280



touchesEnded:withEvent:

# Multitouch

- ➊ Demo

# UITextField

- ⦿ Like UILabel, but editable
- ⦿ Automatically brings up keyboard
- ⦿ In fact, only way to bring up keyboard  
Well ... UITextView and UIWebView too.  
But the point is that you cannot bring up keyboard yourself.
- ⦿ Keyboard appears when it becomes “first responder”  
It will do this automatically when user clicks on it.  
Or you can make it do it by sending it `becomeFirstResponder`.  
To make it go away, send `resignFirstResponder` to the UITextField.
- ⦿ Usually text is obtained using text field’s delegate

# UITextField

## ⌚ Delegate methods

Sent when return key pressed. Often send resignFirstResponder here.

- (BOOL)textFieldShouldReturn:(UITextField \*)tf;

Sent when the text field actually resigns being first responder.

Usually extract what was typed here (or at least notify someone to do it).

- (void)textFieldDidEndEditing:(UITextField \*)tf;

## ⌚ UITextInputTraits protocol

Text editing objects (like UITextField) implement it.

Let's you configure the keyboard that will appear when editing occurs.

```
@property UITextAutocapitalizationType autocapitalizationType;  
@property UIReturnKeyType returnType;  
@property BOOL secureTextEntry;  
@property UIKeyboardType keyboardType; // PhonePad, Email, URL, etc.
```

# UITextField

## ⌚ Notifications

When keyboard appears, oftentimes it shifts your UI out of the way. You can sign up to receive notifications (via NSNotificationCenter) about the showing and hiding of the keyboard, so you can react by moving your views around (if necessary).

These notifications are sent by the UIWindow.

UIKeyboardWillShowNotification  
UIKeyboardDidShowNotification  
UIKeyboardWillHideNotification  
UIKeyboardDidHideNotification

```
[ [NSNotificationCenter defaultCenter]
    addObserver:self
        selector:@selector(theKeyboardAppeared:)
        name:UIKeyboardDidShowNotification
        object:self.view.window];
```

The userInfo in the NSNotification you get will have details about changes.

# UITextField

- In 3.2, can add an accessory view to the keyboard  
Not going to cover that, but take a look in the documentation if interested.

# Modal Views

- One VC presents another VC's view modally

Use the following method in UIViewController

`- (void)presentModalViewController:(UIViewController *)modalVC animated:(BOOL);`

This will take over the receiving view controller's space on screen and prevent the user from doing anything but what the presented (modal) view controller wants until it is dismissed.

- When done, the original VC dismisses the modal VC

Note that the modal VC does not dismiss itself (the original VC dismisses it).

The modal VC should message other VC that's done its job (via delegation).

When the original VC gets that message, it then dismisses the modal VC using the following method on itself:

`-(void)dismissModalViewControllerAnimated:(BOOL);`

# Modal Views/UITextField

- Demo

# Wednesday

- ⌚ View Animation
- ⌚ Accelerometer
- ⌚ GameKit
- ⌚ Other if time permits