Announcements

- **Final Project descriptions are due on Friday**
  - If you are still in need of a partner, send me an email by midnight tonight
    - Include a potential app idea in the email
  - I will assist with the matching process and email out groups on Thursday

- **Final Project Proposals start on Monday**

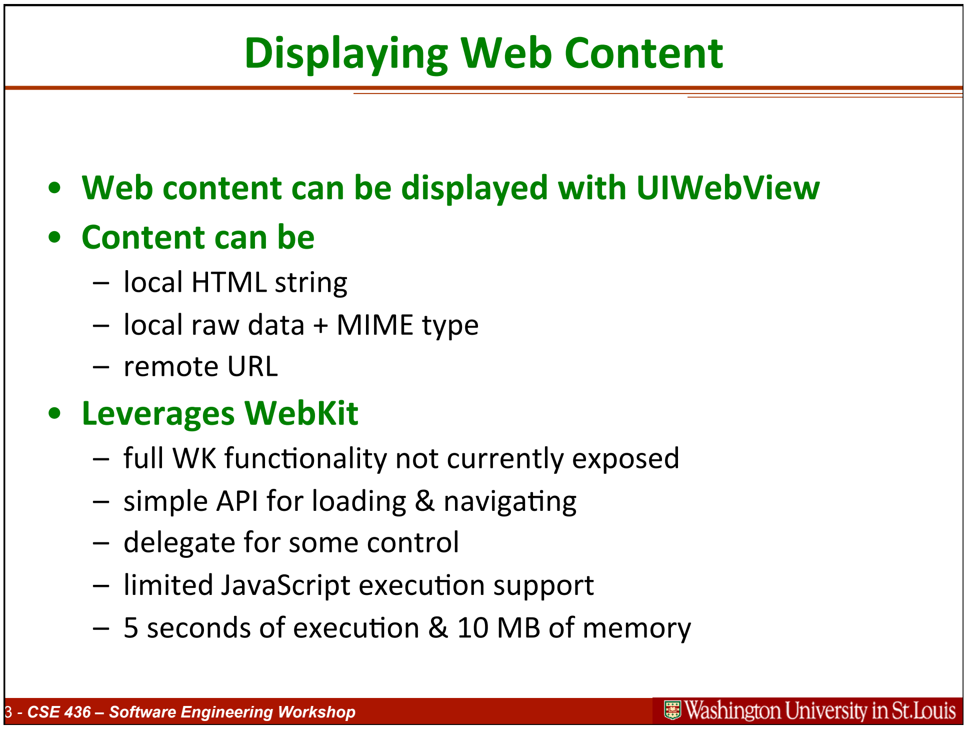
- **Lab 5 is due on Monday March 30th**

Today's Topics

- **Webviews**

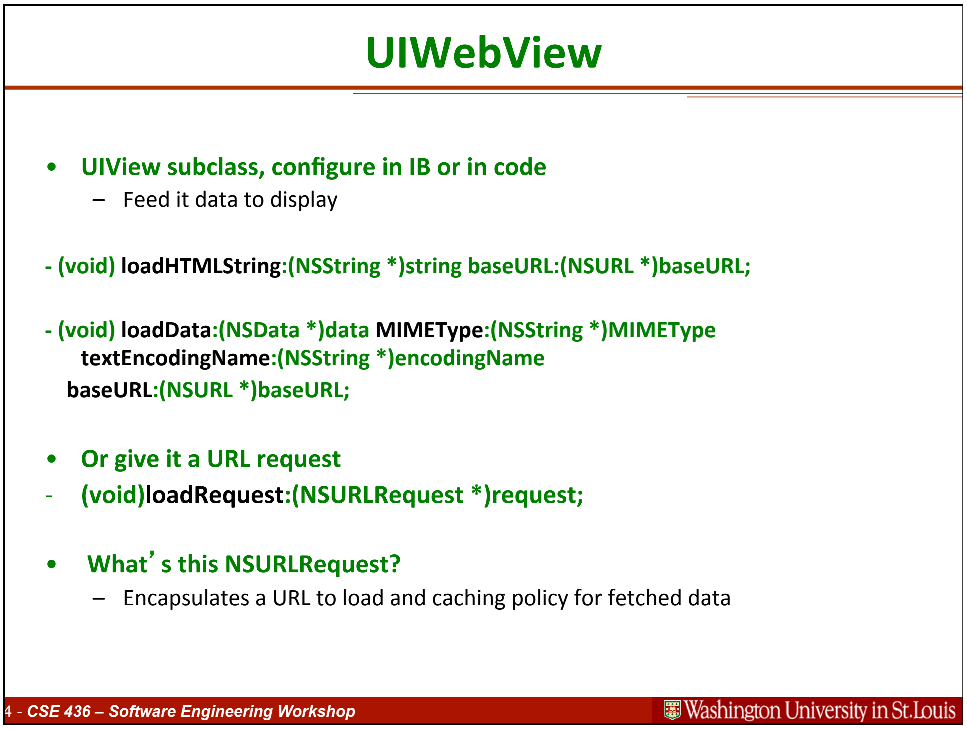
- **MapKit**

- **Touch Events**



Displaying Web Content

- **Web content can be displayed with UIWebView**
- **Content can be**
  - local HTML string
  - local raw data + MIME type
  - remote URL
- **Leverages WebKit**
  - full WK functionality not currently exposed
  - simple API for loading & navigating
  - delegate for some control
  - limited JavaScript execution support
  - 5 seconds of execution & 10 MB of memory



UIWebView

- **UIView subclass, configure in IB or in code**
  - Feed it data to display

```
- (void) loadHTMLString:(NSString *)string baseURL:(NSURL *)baseURL;

- (void) loadData:(NSData *)data MIMEType:(NSString *)MIMEType
      textEncodingName:(NSString *)encodingName
    baseURL:(NSURL *)baseURL;
```

- **Or give it a URL request**

```
-    (void)loadRequest:(NSURLRequest *)request;
```

- **What’s this NSURLRequest?**
  - Encapsulates a URL to load and caching policy for fetched data

UIWebView

- Properties and actions you’ d expect from a web view

```
@property BOOL loading;
@property BOOL canGoBack;
@property BOOL canGoForward;
- (void)reload;
- (void)stopLoading;
- (void)goBack;
- (void)goForward;
```

- A couple others are handy

```
@property BOOL scalesPageToFit;
@property BOOL detectsPhoneNumbers;
```

UIWebViewDelegate

- Callbacks for load progress

```
- (void)webViewDidStartLoad:(UIWebView *)webView;
- (void)webViewDidFinishLoad:(UIWebView *)webView;
```

- Error handling

```
- (void)webView:(UIWebView *)webView
  didFailLoadWithError:(NSError *)error;
```

- Navigation management

```
-(BOOL)webView:(UIWebView *)webView
  shouldStartLoadWithRequest:(NSURLRequest *)request
  navigationType:(UIWebViewNavigationType)navigationType;
```

  - navigationType specifies things like link clicked, reload, form submitted, back/forward, or other

Demo
UIWebView

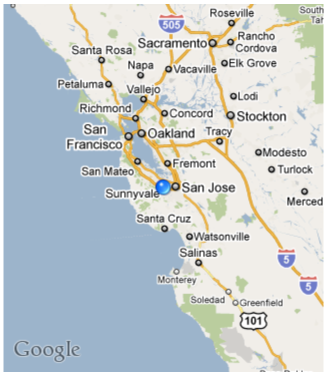
MapKit

What is MapKit?

- API to display Maps
- Classes to transalte between CLLocation and human-readable addresses
- Support for “annotations” (pins on a map)
- Reverse Geocoding

MKMapView

- Handles display of map
- “Map” & “Satellite” types
- Panning and Zooming
- Annotations
- Display User Location



Properties in MKMapView

```
@property MKCoordinateRegion region;

@property CLLocationCoordinate2D centerCoordinate;

@property MKMapType mapType;

@property NSArray *annotations;

@propertyMKUserLocation userLocation;

@property id <MKMapViewDelegate> delegate;
```

MKMapViewDelegate

- Callback methods about loading state:

```
- (void)mapViewWillStartLoadingMap :(MKMapView *)mapView;
- (void)mapViewDidFinishLoadingMap :(MKMapView *)mapView;
- (void)mapViewDidFailLoadingMap :(MKMapView *)mapView
  withError:(NSError *)error;
```

- Callback methods about region changes:

```
- (void)mapView:(MKMapView *)mapView
  regionWillChangeAnimated  :(BOOL)animated;

- (void)mapView:(MKMapView *)mapView
  regionDidChangeAnimated  :(BOOL)animated;
```

MKMapViewDelegate

- Callback methods to customize and interact with “annotations”:

```
- (MKAnnotationView *) mapView :(MKMapView *)mapView
  viewForAnnotation :(id <MKAnnotation>)annotation;

- (void) mapView :(MKMapView *)mapView
  didAddAnnotationViews :(NSArray *)views;

- (void) mapView :(MKMapView *)mapView
  annotationView :(MKAnnotationView *)view
  calloutAccessoryControlTapped :(UIControl *)control;
```

MKAnnotation

- A @protocol - not a @class

- Add to a MapView to plot pins

```
@property CLLocationCoordinate2D coordinate;

@property NSString *title;
@property NSString *subtitle;
```

MKPlacemark

- Conforms to MKAnnotation protocol
- Convenience for holding human-readable addresses alongside Coordinate

- (void)initWithCoordinate :(CLLocationCoordinate2D *)coordinate
addressDictionary :(NSDictionary *)dictionary;

- Easy to convert between AddressBook addresses and location:
  - thoroughfare, subThoroughfare, locality, subLocality, administrativeArea, subAdministrativeArea, postalCode, country, countryCode

MKUserLocation

- Special case of an MKAnnotation
- Represents device' s location only

@property BOOL updating (getter = isUpdating);
@property CLLocation *location;

@property NSString *title;
@property NSString *subtitle;

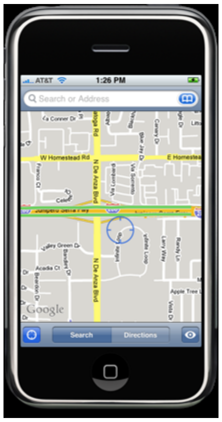
Demo
MKMapView and friends

Core Location

Core Location

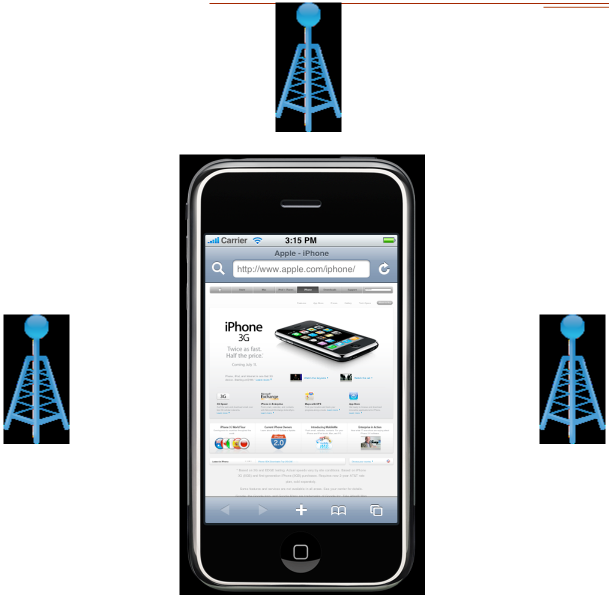
- **What is it?**
- **Core Location**

Activate service



Location ring

Core Location



Core Location

Carrier
3:15 PM
Apple - iPhone
http://www.apple.com/iphone/
iPhone
3G
Twice as fast.
Half the price.

Core Location

Carrier
3:15 PM
Apple - iPhone
http://www.apple.com/iphone/
iPhone
3G
Twice as fast.
Half the price.

Core Location

- **Location Technologies**

Bootstrap
Crosscheck
Complement

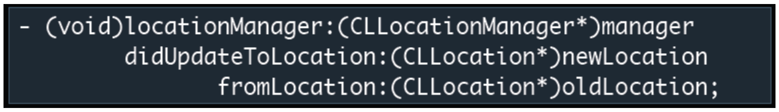
Core Location Framework

Core Location Framework

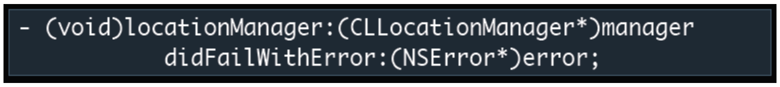
- **The core classes and protocols**
- **Classes**
  - CLLocation
    - Represents a point and vector in the real world
  - CLLocationManager
    - Allows you to get a CLLocation
- **Protocol**
  - CLLocationManagerDelegate

Core Location Framework

- **CLLocationManagerDelegate protocol**
- **Two optional methods**



```
- (void)locationManager:(CLLocationManager*)manager
        didUpdateToLocation:(CLLocation*)newLocation
               fromLocation:(CLLocation*)oldLocation;
```

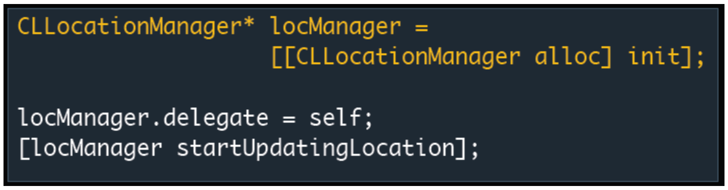


```
- (void)locationManager:(CLLocationManager*)manager
           didFailWithError:(NSError*)error;
```

- **Called asynchronously on main thread**
- **Issues movement-based updates**

Getting a Location

- **Starting the location service**



```
CLLocationManager* locManager =
                    [[CLLocationManager alloc] init];

locManager.delegate = self;
[locManager startUpdatingLocation];
```

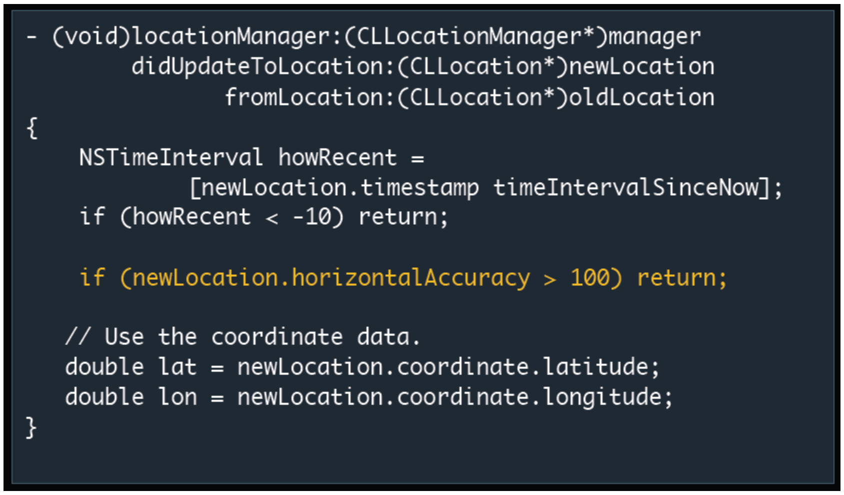
Core Location Framework new in iOS 8

- **iOS 8 introduced additional requirements to obtain your location**
  - Call the requestWhenInUseAuthorization method
  - Add an entry to your plist file to request location
    - NSLocationWhenInUseUsageDescription

- **See example code on course website for a working example**

Getting a Location

- **Using the event data**



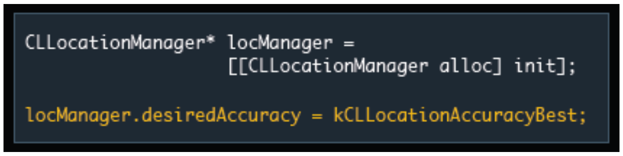
```
- (void)locationManager:(CLLocationManager*)manager
        didUpdateToLocation:(CLLocation*)newLocation
               fromLocation:(CLLocation*)oldLocation
{
    NSTimeInterval howRecent =
             [newLocation.timestamp timeIntervalSinceNow];
    if (howRecent < -10) return;

    if (newLocation.horizontalAccuracy > 100) return;

   // Use the coordinate data.
   double lat = newLocation.coordinate.latitude;
   double lon = newLocation.coordinate.longitude;
}
```

Desired Accuracy

- **Choosing an appropriate accuracy level**



```
CLLocationManager* locManager =
                   [[CLLocationManager alloc] init];

locManager.desiredAccuracy = kCLLocationAccuracyBest;
```

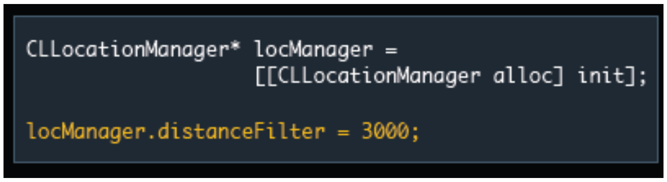
- **Choose an appropriate accuracy level**
  - Higher accuracy impacts power consumption
  - Lower accuracy is “good enough” in most cases

- **Can change accuracy setting later if needed**

- **Actual accuracy reported in CLLocation object**

Distance Filter

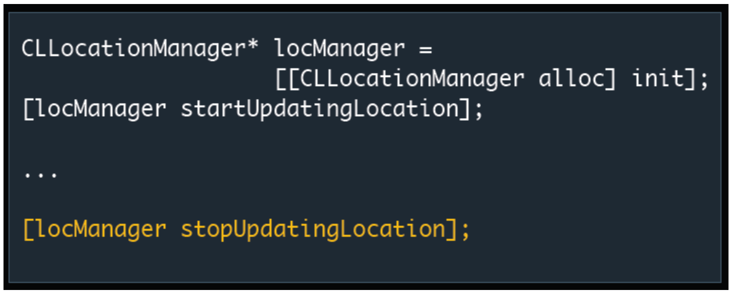
- Choosing an appropriate update threshold
- New events delivered when threshold exceeded



```
CLLocationManager* locManager =
                  [[CLLocationManager alloc] init];

locManager.distanceFilter = 3000;
```

Stopping the Service



```
CLLocationManager* locManager =
                  [[CLLocationManager alloc] init];
[locManager startUpdatingLocation];

...

[locManager stopUpdatingLocation];
```

- Restart the service later as needed

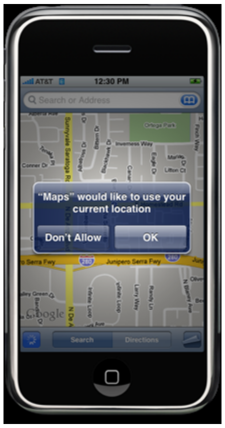
Responding to Errors

- User may deny use of the location service

- Results in a kCLErrorDenied error

- Protects user privacy

- Occurs on a per-application basis



“Maps” would like to use your current location

Responding to Errors

- Location may be unavailable

- Results in a kCLErrorLocationUnknown error

- Likely just temporary

- Scan continues in background

Demo
GPS Data

CLGeocoder

- **Given a location, what's the human-readable address? (Reverse Geocoding)**
- **Given an address which is my latitude and longitude? (Forward Geocoding)**

- **CLGeocoder uses a completion handler that is a "block" in Objective-C**
  - Blocks are similar to anonymous functions, we will discuss them in more detail in a later lecture

- **(void)reverseGeocodeLocation:(CLLocation *)location completionHandler:(CLGeocodeCompletionHandler)completionHandler**

```
CLGeocoder *geoCoder;

[self.geoCoder reverseGeocodeLocation: locationManager.location completionHandler:   ^(NSArray *placemarks, NSError *error) {

     //Get nearby address
    CLPlacemark *placemark = [placemarks objectAtIndex:0];

    //String to hold address
    NSString *locatedAt = [[placemark.addressDictionary valueForKey:@"FormattedAddressLines"] componentsJoinedByString:@", "];

    //Print the location to console
    NSLog(@"I am currently at %@",locatedAt);

  }];  //End of block code
```

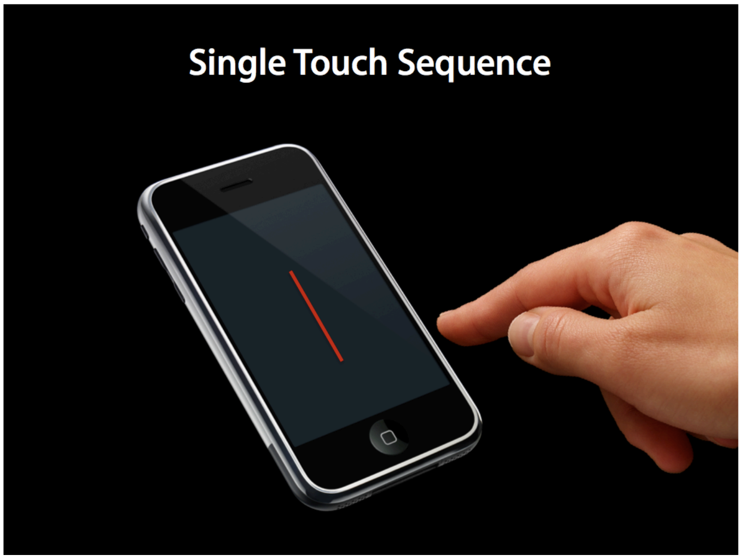
Demo
CLGeocoder Examples

UI Touch Events

Touch Events

- Touch Sequences
- Touch and Event Objects
- Touch Delivery
- Single Touch
- Multiple Touches

Touch Events (from CS193p) slides



Single Touch Sequence

UITouch

- Represents a single finger

```
@property(nonatomic,readonly) NSTimeInterval timestamp;
@property(nonatomic,readonly) UITouchPhase phase;
@property(nonatomic,readonly) NSUInteger tapCount;
@property(nonatomic,readonly,retain) UIWindow *window;
@property(nonatomic,readonly,retain) UIView *view;
- (CGPoint)locationInView:(UIView *)view;
- (CGPoint)previousLocationInView:(UIView *)view;
```

UIEvent

- A container for one or more touches

```
@property(nonatomic,readonly) NSTimeInterval timestamp;
- (NSSet *)allTouches;
- (NSSet *)touchesForWindow:(UIWindow *)window;
- (NSSet *)touchesForView:(UIView *)view;
```

Receiving Touches

- UIResponder

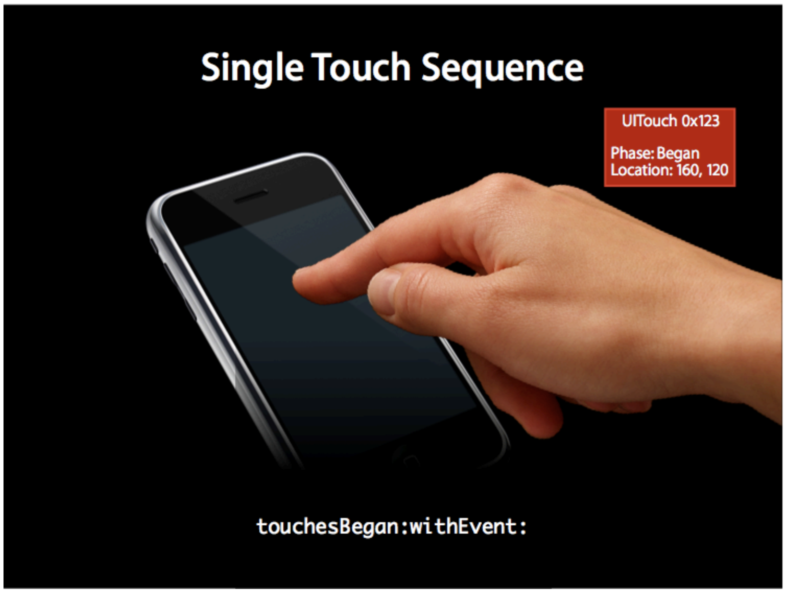
```
- (void)touchesBegan:(NSSet *)touches withEvent:(UIEvent *)event;

- (void)touchesMoved:(NSSet *)touches withEvent:(UIEvent *)event;

- (void)touchesEnded:(NSSet *)touches withEvent:(UIEvent *)event;

- (void)touchesCancelled:(NSSet *)touches withEvent:(UIEvent *)event
```

Touch Events (from CS193p) slides



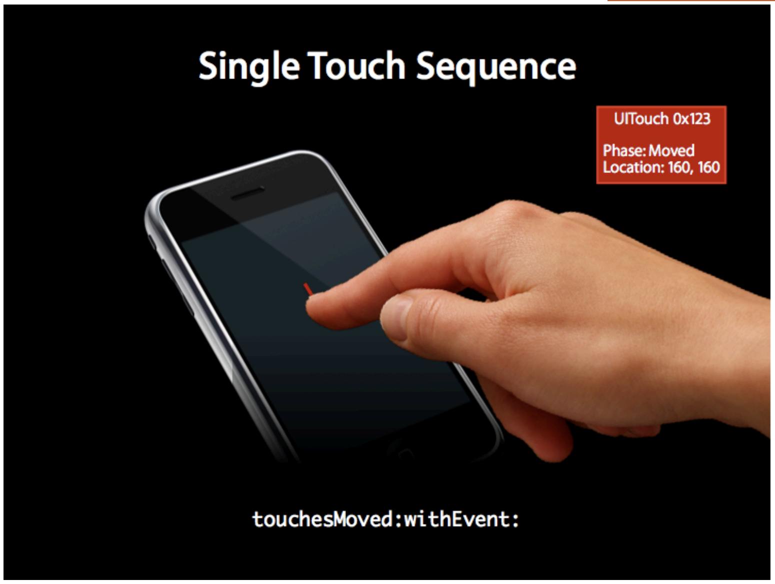
Single Touch Sequence

UITouch 0x123

Phase: Began
Location: 160, 120

`touchesBegan:withEvent:`

Touch Events (from CS193p) slides



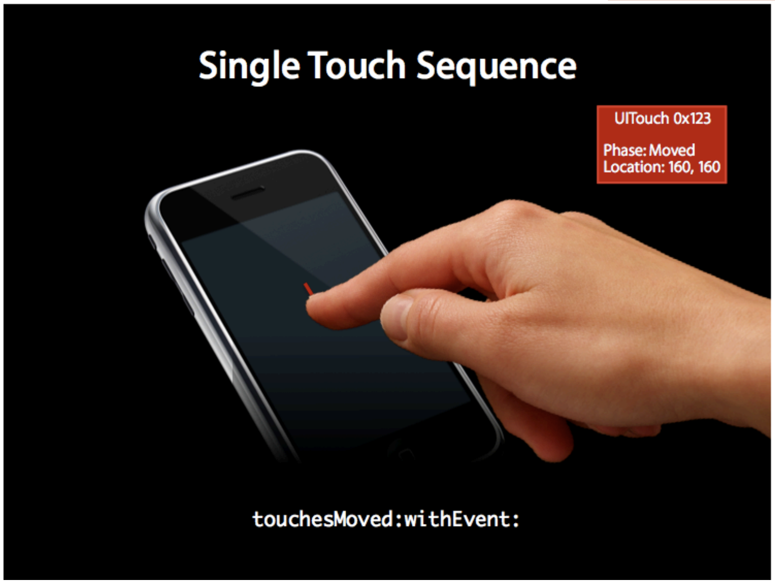
Single Touch Sequence

UITouch 0x123

Phase: Moved
Location: 160, 160

`touchesMoved:withEvent:`

Touch Events (from CS193p) slides



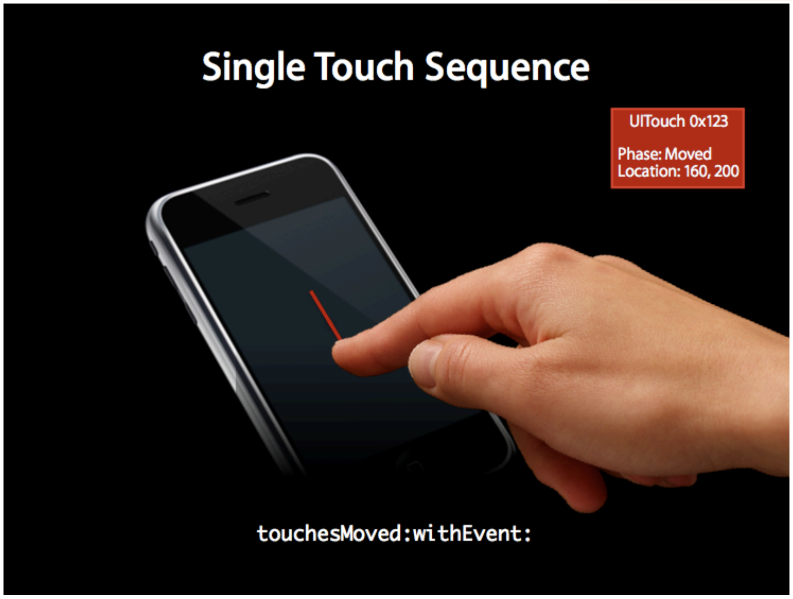
Single Touch Sequence

UITouch 0x123

Phase: Moved
Location: 160, 160

`touchesMoved:withEvent:`

Touch Events (from CS193p) slides



Single Touch Sequence

UITouch 0x123

Phase: Moved
Location: 160, 200

touchesMoved:withEvent:

Touch Events (from CS193p) slides

Single Touch Sequence

UITouch 0x123

Phase: Moved
Location: 160, 240

touchesMoved:withEvent:

Touch Events (from CS193p) slides

Single Touch Sequence

UITouch 0x123

Phase: Ended
Location: 160, 240

touchesEnded:withEvent:

Demo
Touches