iOS: MapKit and Movies

Tommy MacWilliam

tmacwilliam@cs.harvard.edu

April 4, 2012

Today

- Keepin' it Stylin'
- MapKit
- Annotations
- Core Location
- Movies

Style

iOS: MapKit and Movies

Tommy MacWilliam

Keepin' it Stylin'

MapKit

Core Location

Notification Center

Movies

- code style is serious business

Style

iOS: MapKit and Movies

Tommy MacWilliam

Keepin' it Stylin'

MapKit

Core Location

Notification Center

Movies

- seriously.

DO NOT WANT


```
public function getId()
{
    ...
}
function set_id($Id){
...
}
```

DO NOT WANT

DO NOT WANT


```
// this loads view called header
$this->load->view('header');
// this loads the view called footer
$this->load->view('footer');
```

DO NOT WANT

DO NOT WANT


```
$hereIsAReallyLongListOfItemsThatIMade =
    array(1, 2, 3);
$lstFItms = array(4, 5, 6);
```

DO NOT WANT

DO NOT WANT


```
/* set a to 1 */
$a = 1;
$b = 2; // set b to 2
```

DO NOT WANT

MapKit

- Google Maps for iOS!
- defined in `MapKit/MapKit.h`
- also need `MapKit.framework`

Defining Areas

- `CLLocationCoordinate2D`: struct containing `latitude` and `longitude`
- `MKCoordinateSpan`: struct containing `latitudeDelta` and `longitudeDelta`
- `MKCoordinateRegion`: struct containing `center` and `span`

Using Areas

- `MapCenterExample`

Annotations

- pushpins with popovers
- `MKPointAnnotation` defines model for pin
  - `coordinate`, `title`, `subtitle`
- `addAnnotation` to `MapView`

Annotations

- `MapAnnotationExample`

Custom Annotation Models

- create a new class implementing `MKAnnotation`
  - must expose `coordinate`, `title`, and `subtitle`

Custom Annotation Views

- same paradigm as `UITableViewCells`
- `MKMapViewDelegate` declares methods that create annotations
  - be sure to connect `MapKit` delegate

Custom Annotation Views

- `mapView:viewForAnnotation:` just like `tableView:cellForRowAtIndexPath:`
  - more caching with `dequeueReusableAnnotationViewWithIdentifier`

Custom Annotation Views

- `MKPinAnnotationView` replaces `UITableViewCell`
  - `pinColor`
  - `animatesDrop`
  - `canShowCallout`
  - `rightCalloutAccessoryView`

Delegate Methods

- `mapView:annotationView:calloutAccessoryControlTapped`
- `mapView:regionWillChangeAnimated:`
- `mapView:didDeselectAnnotationView:`

Custom Annotations

- `CustomAnnotationsExample`

Overlays

- define arbitrary lines and shapes
  - `CLLocationCoordinate2D` still defines points
- `MKPolyline, MKCircle, MKPolygon`

Overlays

- `OverlayExample`

Core Location

- event-based framework to get current location
- defined in `CoreLocation/CoreLocation.h`
- also need `CoreLocation.framework`

Configuration

- configure using `CLLocationManager`
  - `distanceFilter`: distance device needs to move before updating
  - `desiredAccuracy`: GPS data accuracy
  - `startUpdatingLocation`
- `stopUpdatingLocation` to conserve battery!

Delegate

- `CLLocationManagerDelegate` defines callback methods
- `locationManager:didUpdateToLocation: fromLocation:`

Core Location

- `CoreLocationExample`

Notification Center

- implementation of the observer design pattern
  - remember broadcasts in Scratch?

Notification Center

- objects register to listen to notifications
- another object broadcasts a notification
- all objects listening fire a callback

Notification Center

iOS: MapKit and Movies

Tommy MacWilliam

Keepin' it Stylin'

MapKit

Core Location

Notification Center

Movies

- `NotificationCenterExample`

MediaPlayer

- media playback framework
- defined in `MediaPlayer/MediaPlayer.h`
- also need `MediaPlayer.framework`

Movie Player

- `MPMoviePlayerController` manages video playback
- needs to have a `frame`, or can't view video

Player Notifications

- `MPMoviePlayerContentPreloadDidFinishNotification`
- `MPMoviePlayerPlaybackDidFinishNotification`
- `MPMoviePlayerPlaybackStateDidChangeNotification`
- `MPMoviePlayerScalingModeDidChangeNotification`

Movie Player

- `MoviePlayerExample`

Or If You're Lazy

- the web is amazing.

For the Lazy

- `WebViewExample`