

iOS Development @ Cohab

Brent Nycum
brentnycum@gmail.com

Developer Survey

- Survey to gauge interest in more classes at Cohab
- Ruby on Rails, jQuery, PHP, CSS, etc
- <http://bit.ly/pHykzE>

iShuttle Launch Party

- September 28th
- 4:30 - 7:00 pm
- Open Bar
- Drawing for concert tickets

Tech Rouler Launch Party

- September 29th
- 5:30 - 7:00 PM
- “Come here our answer to those who say there are no good jobs to be found in North Louisiana. We know better.”

Baby News



September 8th @ 1:57 A.M.

- Media
 - Camera
 - Movies
 - Images
 - Sounds

- Core Location
- Map Kit

Media

Modal View Controllers

- Child window that requires user interact with it before returning to the parent window

Modal View Controllers

- From any View Controller
 - `UIViewController *vc = ....;`
 - `[self presentViewController:vc animated:YES];`

Modal View Controllers

- Dismissing is easy as well, from your modal View Controller
 - `[self dismissModalViewControllerAnimated:YES];`

Camera

Camera

- UIImagePickerController
- Modal view to get media from camera or photo library

Camera

- To present the camera interface, you must first ensure that three things are in place

Camera

- 1 - The device your app is running on must have a camera
 - If taking pictures or movies is essential to your app, specify that by configuring the `UIRequiredDeviceCapabilities` in your `Info.plist` property file
 - If your app remains useful on a device without a camera, then your code should provide an alternate path

Camera

- 2 - The device's camera must be available to use
 - `isSourceTypeAvailable`: method

Camera

- Why wouldn't the source be available if the device has one?
 - Trying to pick from the User's Library when the Library is empty
 - If the camera is already in use

Camera

- 3 - You must have implemented a delegate object to respond to the user's interaction with the image picker controller

Camera

- UIImagePickerController Usage
 - Create with alloc / init
 - Set delegate
 - Configure source, kind of media
 - Present it
 - Respond to delegate method when user is done

Camera

- What the user can do depends
 - Some have camera
 - Some can record video
- You want to check what's available

Camera

```
UIImagePickerController *picker = [[UIImagePickerController alloc] init];  
picker.delegate = self;
```

Camera

- `+(BOOL)isSourceTypeAvailable:(UIImagePickerControllerSourceType) sourceType;`
- `UIImagePickerControllerSourceTypePhotoLibrary`
- `UIImagePickerControllerSourceTypeCamera`
- `UIImagePickerControllerSourceTypeSavedPhotosAlbum`

Camera

```
if ([UIImagePickerController  
isSourceTypeAvailable:UIImagePickerControllerSourceTypeCamera]) {  
    picker.sourceType = UIImagePickerControllerSourceTypeCamera;  
}
```

Camera

- Not every source type can give video
 - `+(NSArray *)availableMediaTypesForSourceType:(UIImagePickerControllerSourceType)sourceType;`
 - Returns an array of strings, but you can check against constants
 - `kUTTypeImage`
 - `kUTTypeMovie`

Camera

```
NSString *desired = (NSString *)kUTTypeImage;

if ([[UIImagePickerController availableMediaTypesForSourceType:picker.sourceType]
containsObject:desired]) {

    picker.mediaTypes = [NSArray arrayWithObjects:desired, nil];

}
```

Camera

- Checking for Front/Rear cameras
 - `+(BOOL)isCameraDeviceAvailable:(UIImagePickerControllerCameraDevice)cameraDevice;`
 - `UIImagePickerControllerCameraDeviceFront`
 - `UIImagePickerControllerCameraDeviceRear`

Camera

- Is there a flash available?
 - `+(BOOL)isFlashAvailableForCameraDevice:
(UIImagePickerControllerCameraDevice)device;`

Camera Delegate

- – `(void)imagePickerController:(UIImagePickerController *)picker
didFinishPickingMediaWithInfo:(NSDictionary *)info;`

Camera Delegate

- What's in this info dictionary?
 - UIImagePickerControllerMediaType // kUTTypeImage or kUTTypeMovie
 - UIImagePickerControllerOriginalImage // UIImage

Camera

- No way to simulate camera
 - Either rely on Library or test using an actual device

Camera

- Easy way to add photos to our Simulator
 - Drag pictures onto the simulator
 - Safari will open to picture
 - Tap and hold for the menu to save picture
- Or save pictures from the browser

Movies

Movies

- MPMoviePlayerViewController
- H.264 Baseline and MPEG-4

Movies

- `-initWithContentsOfURL:(NSURL *)url;`
- `NSString *path = [[NSBundle mainBundle] pathForResource:@"movie" ofType:@"m4v"];`
- `NSURL *url = [NSURL fileURLWithPath:path];`

Presenting

- Presented modally like Image Picker, but...
- `[self presentViewControllerAnimated:...];`
- `[self dismissMoviePlayerViewControllerAnimated];`

Sound

Sounds

- Core Audio, AVFoundation, Media Player, OpenAL, etc

Sounds

- For doing simple sounds there are 2 choices
 - System Sound API
 - AVAudioPlayer

Sounds

- Simplest sound playing API
 - Short, non-repeating sounds
 - AIFF or WAV uncompressed formats

Sounds

- Register your sound with the system
 - `SystemSoundID mySound;`
 - `NSString *soundFilePath [[NSBundle mainBundle] pathForResource:@"mySound" ofType:@"caf"];`
 - `NSURL *soundFileURL = [NSURL URLWithString:soundFilePath];`
 - `AudioServicesCreateSystemSoundID((CFURLRef)soundFileURL, &mySound);`

Sounds

- Play the sound
 - `AudioServicesPlaySystemSound(mySound);`

Sounds

- Free the sound when done
 - `AudioServicesDisposeSystemSound(mySound);`

AVAudioPlayer

- More sophisticated audio playing
 - Longer sounds
 - Looping
 - Object-Oriented API

AVAudioPlayer

- `NSString *soundFilePath = [[NSBundle mainBundle] pathForResource:@"mySound" ofType:@"mp3"];`
- `NSURL *soundFileURL = [NSURL URLWithString:soundFilePath];`
- `AVAudioPlayer *player = [[AVAudioPlayer alloc] initWithContentsOfURL:soundFileURL];`

AVAudioPlayer

- [player play];
- [player stop];

AVAudioPlayer

- Properties
 - currentTime
 - duration
 - volume

Core Location

CoreLocation

- Framework for managing location and heading

CoreLocation

- Simulator defaults to Cupertino
 - Test using device for now
- iOS 5 Simulator will change this (Fall)

CoreLocation

- CLLocationManager
- Use an instance of this class to establish the parameters that determine when location and heading events should be delivered
- To start and stop the actual delivery of those events

CoreLocation

- Always check to see whether the desired services are available before starting any services and abandon the operation if they are not

CLLocationManager

- + (BOOL)locationServicesEnabled;
- + (BOOL)headingAvailable;
- + (BOOL)significantLocationChangeMonitoringAvailable;
- + (BOOL)regionMonitoringAvailable;
- + (BOOL)regionMonitoringEnabled;

Core Location

- Create Instance of CLLocationManager
 - `[[CLLocationManager alloc] init];`
- Set delegate
 - `locationManager.delegate = self;`

Core Location

- Configure any additional properties
- `desiredAccuracy`
 - `kCLLocationAccuracyBestForNavigation`
 - `kCLLocationAccuracyBest`
 - `kCLLocationAccuracyNearestTenMeters`
 - `kCLLocationAccuracyHundredMeters`
 - `kCLLocationAccuracyKilometer`
 - `kCLLocationAccuracyThreeKilometers`

Core Location

- Start Monitoring for changes
- – `(void)startUpdatingLocation;`
- – `(void)stopUpdatingLocation;`

Core Location

- Listen for changes
 - – (void)locationManager:(CLLocationManager *)manager
didUpdateToLocation:(CLLocation *)newLocation
fromLocation:(CLLocation *)oldLocation;

CLLocation

- Properties
 - `coordinate`
 - `altitude`
 - `horizontalAccuracy` / `verticalAccuracy`
 - `timestamp`
 - `speed`
 - `course`

CLLocation

- speed
 - Instantaneous value (not averaged)
- course
 - Compass direction (not supported by older devices)

CLLocation

- timestamp
 - Location values can be delivered inconsistently

Maps

Map Kit

- Display a Map
- Maps can have annotations on them

Map Kit



Annotations

Annotations

- Adding
 - `-(void)addAnnotation:...`
 - `-(void)addAnnotations:(NSArray *)....;`
- Removing
 - `-(void)removeAnnotation:...`
 - `-(void)removeAnnotations:(NSArray *)...`

Annotations

- Limit the number of annotations
- Limit to visible ones

Annotations

- `-(void)mapView:(MKMapView *) regionDidChangeAnimated: (BOOL);`
- `@property (readonly) MKMapRect visibleRect;`

Annotations

- `MKMapPoint annotationPoint = MKMapPointForCoordinate(annotation.coordinate);`
- `if (MKMapRectContainsPoint(mapView.visibleRect, annotationPoint)) { ... }`

Sample App

- Lets build a clone of the Maps App

Sample App

- Going to use the Google Maps API
- <http://maps.google.com/maps/geo?q=Shreveport&output=csv>
- 200,4,32.5251516,-93.7501789

Sample App

- NSString *urlString = [[NSString stringWithFormat:@"http://maps.google.com/maps/geo?q=%@&output=csv", searchBar.text]

stringByAddingPercentEscapesUsingEncoding:NSUTF8StringEncoding];

Sample App

- `NSString *locationString = [NSString stringWithContentsOfURL:[NSURL URLWithString:urlString]];`
- Be sure to use the non-deprecated methods or actual `NSURLConnection` with delegates
- Using this method for brevity

Sample App

- Get Coordinates
- `NSArray *components = [locationString
componentsSeparatedByString:@","];`
- `double longitude = [[components objectAtIndex:2] doubleValue];`
- `double latitude = [[components objectAtIndex:3] doubleValue];`

Sample App

- How do we change the pin?

Sample App

```
- (MKAnnotationView *) mapView:(MKMapView *)mapView  
viewForAnnotation:(id <MKAnnotation>) annotation{  
    MKPinAnnotationView *annView=[[MKPinAnnotationView alloc]  
initWithAnnotation:annotation reuseIdentifier:@"currentloc"];  
  
    annView.pinColor = MKPinAnnotationColorGreen;  
  
    annView animatesDrop = YES;  
  
    return annView;  
}
```

Questions?