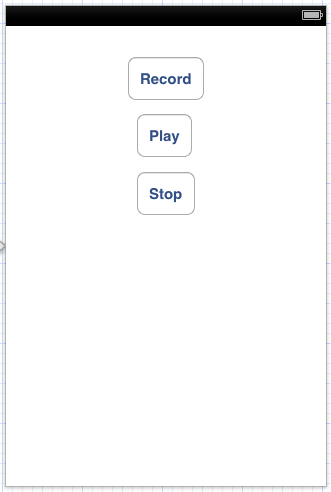
**Audio Recorder**

**Create a new iPhone iOS single view-based application named Record with a corresponding class prefix with Storyboards and Automatic Reference Counting enabled.**

**Since the iOS 7 AVAudioRecorder class is part of the AV Foundation framework it will be necessary to add the framework to the project.**

**Select the product target entry from the project navigator panel (the top item named Record) and clicking on the Build Phases tab in the main panel. In the Link Binary with Libraries section click on the ‘+’ button, select the AVFoundation.framework entry from the resulting panel and click on the Add button.**

**Create buttons that look like this on the single interface:**



**Now go into your Assistant Editor and prepare to make connections**

**You need to make an outlet and an action for all three buttons**

**Your h file should look like this:**

#import <UIKit/UIKit.h>

#import <AVFoundation/AVFoundation.h>

@interface RecordViewController : UIViewController

<AVAudioRecorderDelegate, AVAudioPlayerDelegate>

@property (strong, nonatomic) AVAudioRecorder \*audioRecorder;

@property (strong, nonatomic) AVAudioPlayer \*audioPlayer;

@property (strong, nonatomic) IBOutlet UIButton \*recordButton;

@property (strong, nonatomic) IBOutlet UIButton \*playButton;

@property (strong, nonatomic) IBOutlet UIButton \*stopButton;

- (IBAction)recordAudio:(id)sender;

- (IBAction)playAudio:(id)sender;

- (IBAction)stop:(id)sender;

@end

**Now we need to work on our implementation (m) file**

**When the application is first launched, an instance of the AVAudioRecorder class needs to be created. This will be initialized with the URL of a file into which the recorded audio is to be saved.**

**Also passed as an argument to the initialization method is an NSDictionary object indicating the settings for the recording such as bit rate, sample rate and audio quality. Well need to do that to the viewDidLoad method**

- (void)viewDidLoad {

[super viewDidLoad];

\_playButton.enabled = NO; **//may need synthesizing disabled because nothing has been recorded yet**

\_stopButton.enabled = NO; **//may need synthesizing disabled because nothing has been recorded yet**

NSArray \*dirPaths;

NSString \*docsDir;

**//makes a referenceable variable for the folder that will save the audio**

dirPaths = NSSearchPathForDirectoriesInDomains(

NSDocumentDirectory, NSUserDomainMask, YES);

docsDir = dirPaths[0];

**//makes a referenceable variable path to the file that will be named sound.caf**

NSString \*soundFilePath = [docsDir

stringByAppendingPathComponent:@"sound.caf"];

**//This will be the location of the file**

NSURL \*soundFileURL = [NSURL fileURLWithPath:soundFilePath];

**//Here’s the aforementioned dictionary item…I’d look up more on these specs in the ios docs…**

NSDictionary \*recordSettings = [NSDictionary

dictionaryWithObjectsAndKeys:

[NSNumber numberWithInt:AVAudioQualityMin],

AVEncoderAudioQualityKey,

[NSNumber numberWithInt:16],

AVEncoderBitRateKey,

[NSNumber numberWithInt: 2],

AVNumberOfChannelsKey,

[NSNumber numberWithFloat:44100.0],

AVSampleRateKey,

nil];

NSError \*error = nil;

**//initialize recorder**

\_audioRecorder = [[AVAudioRecorder alloc]

initWithURL:soundFileURL

settings:recordSettings

error:&error];

if (error)

{

NSLog(@"error: %@", [error localizedDescription]);

} else {

[\_audioRecorder prepareToRecord];

}

}

**Here are your actions…pretty straightforward**

- (IBAction)recordAudio:(id)sender {

if (!\_audioRecorder.recording)

{

\_playButton.enabled = NO;

\_stopButton.enabled = YES;

[\_audioRecorder record];

}

}

- (IBAction)playAudio:(id)sender {

if (!\_audioRecorder.recording)

{

\_stopButton.enabled = YES;

\_recordButton.enabled = NO;

NSError \*error;

\_audioPlayer = [[AVAudioPlayer alloc]

initWithContentsOfURL:\_audioRecorder.url

error:&error];

\_audioPlayer.delegate = self;

if (error)

NSLog(@"Error: %@",

[error localizedDescription]);

else

[\_audioPlayer play];

}

}

- (IBAction)stop:(id)sender {

\_stopButton.enabled = NO;

\_playButton.enabled = YES;

\_recordButton.enabled = YES;

if (\_audioRecorder.recording)

{

[\_audioRecorder stop];

} else if (\_audioPlayer.playing) {

[\_audioPlayer stop];

}

}

**In order to receive notification about the success or otherwise of recording or playback it is necessary to implement some delegate methods of the AVAudioPlayer. We need to implement the methods to indicate errors have occurred and also when playback finished. Once again, edit the recordViewController.m file and add these methods as follows:**

-(void)audioPlayerDidFinishPlaying:

(AVAudioPlayer \*)player successfully:(BOOL)flag

{

\_recordButton.enabled = YES;

\_stopButton.enabled = NO;

}

-(void)audioPlayerDecodeErrorDidOccur:

(AVAudioPlayer \*)player

error:(NSError \*)error

{

NSLog(@"Decode Error occurred");

}

-(void)audioRecorderDidFinishRecording:

(AVAudioRecorder \*)recorder

successfully:(BOOL)flag

{}

-(void)audioRecorderEncodeErrorDidOccur:

(AVAudioRecorder \*)recorder

error:(NSError \*)error

{

NSLog(@"Encode Error occurred");

}